Self-Sovereign Identity Management: A Comparative Study and Technical Enhancements

Noot A. Alissa Author[1†] and Waleed A. Alrodhan* Author[2††],
441022235@sm.imamu.edu.sa *warodhan@imamu.edu.sa*
Imam Mohammad bin Saud Islamic University, College of Computer and Information Sciences, Department of Computer Science, Riyadh, KSA

Abstract
Nowadays usage of different applications of identity management IDM demands prime attention to clarify which is more efficient regarding preserve privacy as well as security to perform different operations concerning digital identity. Those operations represent the available interactions with identity during its lifecycle in the digital world e.g., create, update, delete, verify and so on. With the rapid growth in technology, this field has been evolving with a number of IDM models being proposed to ensure that identity lifecycle and face some significant issues. However, the control and ownership of data remines in the hand of identity service providers for central and federated approaches unlike in the self-sovereign identity management SSIM approach. SSIM is the recent IDM model were introduced to solve the issue regarding ownership of identity and storing the associated data of it. Thus, SSIM aims to grant the individual's ability to govern their identities without intervening administrative authorities or approval of any authority. Recently, we noticed that numerous IDM solutions enable individuals to own and control their identities in order to adapt with SSIM model. Therefore, we intend to make comparative study as much of these solutions that have proper technical documentation, reports, or whitepapers as well as provide an overview of IDM models. We will point out the existing research gaps and how this study will bridge it. Finally, the study will propose a technical enhancement, everKEY solution, to address some significant drawbacks in current SSIM solutions.

Keywords:
Self-sovereign identity, blockchain-based identity management, decentralized digital identity, emerging identity solutions, identity management models.

1. Introduction

Digital identity has been a hot topic in academia in the past few years due to its various methods of and levels of preserving users' privacy and applicable security mechanisms. Those methods attempted to overcome associated struggles of managing identity considering a significant challenging issue in cyberspace by providing different solutions for storing and controlling individuals' identity information. So far, this field has been evolving with a number of identity management frameworks that define the whole lifecycle of digital identity from creation, storage, authentication, authorization to revocation and destruction. This study aims to provide an overview of the common frameworks that have their own method in managing identity, focusing on one of the recent ones; Self-Sovereign Identity Management SSIM. This model emerged in 2016 through the defined ten principles introduced by Christopher Allen. SSIM aims to grant the individuals complete control and management over their identities without the need for centralized authority or a trusted third party. However, the emerging number of proposed SSIM solutions has motivated us to gather and analyze them in one place, aiming to build a scientific reference for the researchers in the field. Towards this goal, we intend to present an academic investigation of the emerging SSIM solutions and bridge the gap between relevant recent studies.

Many proposed solutions have been introduced to be integrated with the SSIM model's requirements while keeping their own requirements. Each varies concerning the SSIM adaption model. We have noticed an evident lack of comprehensive study involving all solutions that adhered to the concept of Self-Sovereign Identity and investigated their architecture, design, strengths, and weaknesses. Thus, there is a need for a comparative study involving all solutions with proper technical documentation, reports, or whitepapers to reach an overall picture of these solutions to fill the gap in this area.

The study aims to accomplish three fundamental objectives that came from the need for more academic researches that revolves around the SSIM model.

- Provide an overview of the concept of Self-Sovereign identity model and the current proposed solutions.
- Clarify the difference between the Self-Sovereign identity model and other identity management models.
- Describe the shortcomings and differences of the current Self-Sovereign identity model solutions in order to fill the existing research gap.

Manuscript received December 5, 2023
Manuscript revised December 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.12.3

Below, we state our research questions (RQ), which contribute to filling the existing research gap and aim to avoid ambiguity about this model:
RQ1. What are the main differences between the traditional Identity Management models and the Self-Sovereign identity model?
RQ2. Do the current Self-Sovereign identity model solutions solve the existing Identity Management issues?
RQ3. What are the main shortcomings of the current Self-Sovereign identity model solutions? And how can they be addressed?

Answering the above questions would shed light on the SSIM model and current solutions that can be integrated with it. Besides, it enables us to obtain more elaborate insight into their differences, shortcomings, and digital identity issues that need to be addressed. This paper is organized as follows. The next section provides the base background that supports this study. Section 3 presents most common identity management models. Section 4 compares most available SSIM solutions. In section 5, we discuss the results and highlight the possible enhancements. Section 6 concludes the study by summarizing the results and gives some recommendations.

2. Background

2.1 Digital Identity Life Cycle

The term of identity is denoted to an expression which is referred to an entity in a given context that has recognizably distinct existence (e.g., person or organization). This expression shows a defied collection of associated information regarding a specific entity and connected to one or more attributes that consider a distinctive characteristic of a particular entity [3]. Also, these attributes could refer to be personally identifiable information (PII) in most cases.

In the context of identity, every entity might have more than one identity. Those identities called a partial identity, which is considered a subset of the whole identity that has a certain entity. An identifier is an attribute or set of attributes that can uniquely characterise an identity in terms of used to refer to an entity in a specific context (e.g., a health insurance card number together with a name of the insurance company) [3]. We could consider that identifier is a distinct attribute of an entity that must be unique in the context of use. Figure 1 clarifies the relationship between the entity, identities, attributes and identifiers [4].

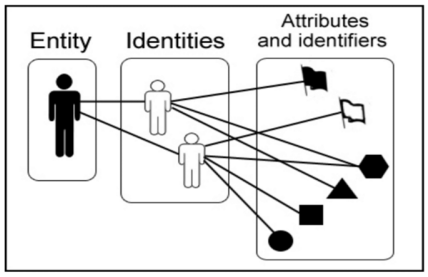

Fig 1. Relationship between entities, identities, attributes and identifiers

As shown in figure 1, attributes can be shared by identities of a particular entity, and those identities represent an expression of a subset of all possible attributes of a given entity.

According to ISO/IEC 24760 [3], five stages of identity lifecycle has been identified within the identity management system: Unknown, Established, Active, Suspended, and Archived. No information is present in the identity registry in the unknown stage, so that entity will also be unknown. When the identity's entity is enrolled to the established stage, the identity information will be generated and registered, e.g. reference identifier. Afterwards, identity information will be verified during the enrolment process by performing initial entity authentication, which is necessary for including it in the identity register. This stage allows the entity to be known within a particular domain of applicability. After activating the identity's entity in the active stage, the established identity can be used by authorized entities, allowing them to access the resources and interact with services provided by the target domain.

Five possible transitions can be applied for managing identity in the active stage, namely: identity adjustment, suspension, maintenance, archive, delete. Identity adjustment concerns updating associated information regarding the entity in the identity registry in which that new information gives rise to the modification of activation information (access rights). Suspension transaction indicating that some identity information for entity stored in identity register is unavailable to use as temporarily which leads to removing access rights expressed in that information. Maintenance transaction enables the identity information stored in the identity register to be updated through changing one or more attributes values of identity and without modifying activation information. Archive transaction is a partial removal of identity information from the identity register, and it couldn't be used for recognizing the entity except in case re-enrolment where the archived information helps to establish a new identity for the entity by including some of that information (restore). Delete transaction is the complete removal of stored identity information for a specific identity and the entity becomes unknown by referring to the first stage.

Suspended stage means that the entity cannot utilize the resources of a particular domain as a result of removing his access rights temporarily, which needs to be reactivated for some identity information to be available for use again. Finally, the identity can be archived in the last stage to completely delete or restore the archived information in the re-enrolment process to establish a new identity. Figure 2 shows all the above identity lifecycle in an identity management system by clarifying the previous stages with its transactions, giving that no information about an entity initially will be unknown [3].

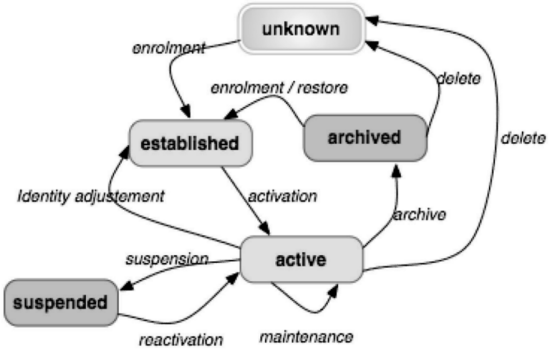

Fig 2. Identity Lifecycle

2.2 Identity Management

The widespread use of web applications has resulted in the importance of managing digital identity by facilitating different available operations that can be performed on the Identity through its lifecycle. Identity management, also known as Identity and access management (IAM), is designed to implement user access policies and rules. Given that Identity and access management are two significant parts for any web services and completing each other through determining and authenticating the identity of the user and authorizing him to access a particular service or source he wants after verifying his Identity, the restrictions associated with it, the level of access and the permissions he has; so, the IAM systems has three main tasks, namely: identify, authenticate, and authorize.

A recent draft of ISO/IEC 24760 [3] defines identity management as a set of processes, policies, standards, and technologies included in managing the lifecycle of identity information from initial enrolment to deletion and concerning to an entity that has known identity in a particular domain. Functions of identity information authority will be supported by identity management processes, policies, standards, and technologies that enable the interactions between the entity for which identity is managed and the identity information authority. An identity information authority is an entity related to a particular domain and concerned with making provable statements to the relying party about one or more attribute values of a given entity. A similar definition of identity management can be found in ITU-T X.1250 [5]; identity management is defined there as a " set of functions and capabilities (e.g., administration, management and maintenance, discovery, communication exchanges, correlation and binding, policy enforcement, authentication and assertions) used for: assurance of identity information (e.g., identifiers, credentials, attributes), assurance of the identity of an entity (e.g., users/subscribers, groups, user devices, organizations, network and service providers, network elements and objects, and virtual objects), and supporting business and security applications."

Since identity management is an essential part of many security services and therefore access management systems that authorize the legitimate user to access the services he wishes. This significance has taken place due to identity management processes that included managing identity lifecycle, identity information, entity authentication, and then authorization. The authentication process is concerned with verifying associated attributes of a specific entity's identity, and in case of failure, this process will reflect the validity of entering a particular system. Therefore, most authentication methods used depend on one or more of three factors: 1) something you know, e.g. Password. 2) something you have, e.g. One-time Password. 3) something you are, e.g. Biometrics [6]. To summarize that, this process aims to compare a set of attributes that refer to a certain entity for allowing it to be sufficiently recognized within a given context.

2.3 Self-Sovereign Identity Principles

It was proposed by Christopher Allen [7] to support the SSIM model which relies on a peer-to-peer network to securely exchange the information and grant the users full control over their identities. These principles represent a comprehensive spectrum of SSIM requirements, enabling some researchers in this field to use it as a guiding principle to evaluate the recent SSIM solutions. Parts of these principles have already been involved in laws of identity proposed by Kim Cameron [8]. However, Allen defines the concept of “Self-Sovereign identity” by introducing ten principles as follows:

- **Existence**: Users must have an independent existence, which can never exist only in digital form.
- **Control**: Users must have the entire control over their identities and be able to handle them. This means they should be decisive authorities regarding these identities.
- **Access**: Users must be able to access their own identities and be aware of all changes adopted to all the claims associated with their identities. This must happen without connecting to any third party.

- **Transparency**: Systems and algorithms used to administer and operate the network of identities should be transparent in how they function; they are managed and updated. Besides, the code of those systems should be open source; thereby, anyone can figure out the operating environment of it.
- **Persistence**: Identities must be long-lived forever or at least according to the user's wishes.
- **Portability**: Information about identity must be transportable and must not be held by a singular third-party, even those who have a high level of trustworthiness.
- **Interoperability**: Identities should be as widely usable as possible and ensure information about it is widely available, crossing boundaries while preserving user control.
- **Consent**: Users must agree regarding using their identities and sharing those data must only occur with their consent.
- **Minimalazition**: Disclosure of data must be minimized, which is necessary to accomplish the required task. Some techniques can support this principle, such as the zero-knowledge technique, which improves privacy as best as possible.
- **Protection**: Users' rights must be protected, and in case of an existing conflict between the needs of the network and the rights of users, priority should be given to the users. To ensure that, the authentication process must occur through independent algorithms run in a decentralized manner.
-

2.4 Web Services Standards

Web services standards are mainly supported by organizations that aim to find an appropriate method for transferring data across multiple web domains and mainly enable interoperability between different platforms of software applications [9]. Most of these are based on eXtensible Markup Language XML, which is designed to encode web documents in a format that would be readable for humans. In this study, we are concerned with World Wide Web Consortium W3C and Organization for the Advancement of Structured Information Standards OASIS2 that focused on the standards of web services and further used through the context of this study.

2.4.1 Organization for the Advancement of Structured Information OASIS standard

It was founded in 1993 as "Standard Generalized Markup Language SGML open" that aims to provide and adopt guidelines for interoperability between the products that support this language. It has been worked on the SGML until the XML appeared in 1998 then shifted its activity to establish qualified open standards regarding global information society, especially in web services utilized in various environments of software platforms2. One of the OASIS standards is SAML, which mainly supports exchanging the data of authentication/authorization between concerned parties, i.e., Identity Provider IDP and Service Provider SP. This standard is explained below in further detail.

2.4.2 Security Assertion Markup Language SAML Standard

It is an XML-based standard for exchanging the data of authentication/authorization between web applications entities. SAML2.0 is the latest version of the SAML standard, and it is incompatible with the previous version SAML 1.13. SAML standard defines three roles [10]: principle or user, identity provider IDP, and service provider SP. SP is a party that provides services to the user after receiving a security token issued by an identity provider. SP must declare its security policy to the user and contain the issuer information to determine SP accepted tokens, the credentials to be asserted, security tokens type, and identity proofing methods. IDP is a party that issues and signs the user's credentials and generator of security tokens after user authentication success by one of the authentication methods supported. User is a party that wants to obtain the services from the SP and must agree to the security policy of the SP before going to the IDP for obtaining the security token required from the SP.

SAML V2.0 defines a security token or so-called SAML assertion [11] which can contain three significant pieces of information about authentication for indicating whether or not that user has been authenticated; if so, it identifies the authentication method used along with its performed time. Attributes can also be included in SAML assertion, thereby containing information about the user, e.g. phone number, email and so on. the last aspect is related to an authorization decision, which is a recommended access control decision for whether or not that user is worth obtaining access to a given resource.

2.5 World Wide Web Consortium W3C standards

It was founded in 1994 as an international community concerning web services to standardize its technologies and, therefore, develop open standards that serve that. It is represented as the main international standards consortium

for the worldwide web1. However, W3C recently has adopted two significant standards to enable a decentralized or "self-sovereign" digital identity. These standards are explained below in further detail.

2.5.1 Decentralized Identifier DID

It is a JSON-based standard for having a unique identifier referring to its owner by using a string URL published universally to have anyone look for it while preserving the security and privacy considerations [12]. It seems helpful for public organizations to use such a permanent DID; alternatively, private sectors or individuals can use temporary DID to resist tracking attempts by intermediary parties and, therefore, hard to correlate their identity information. DID standard support individuals and organizations to generate as many as needed of DIDs for keeping their interactions from being tracked [12]. DID architecture relies on cryptographic mechanisms to establish an end-to-end encryption channel and ensure that any entity has the ability to prove DID that it controls. Figure 3 shows the structure of DID where the part of the scheme is a fixed prefix that refers to DID specification and could be generated by DID method (e.g., Bitcoin), and the last part indicates that unique identifier. DID is mainly registered in any decentralized network and controlled by an entity itself that establishes it with no dependency on any other party to perform that. DID document allows trustable interactions with a specific entity who established DID that basically indicates to its document on the decentralized network, which then express cryptographic mechanisms for proving the entity's control regarding DID with the respect that entity's private key is stored locally in such "a digital wallet".

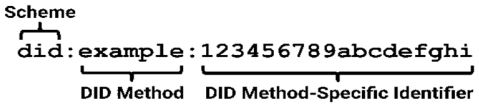

Fig 3. Decentralized Identifier Structure

The overall architecture of DID is shown in figure 4 and DID standard [12] defines them as the following: DID Subject refers to any entity (e.g. person, Institute or government organization) who established DID. Verifiable Data Registry (e.g., distributed ledger technology) concerns recording and storing DID and DID documents. DID URL and DID are explicitly different, where the DID is considered as Uniform Resource Identifier URI contains three parts shown in figure 3 and resolvable to DID document. On the other hand, DID URL extends DID's syntax to involve the path of a specific resource. DID document include the necessary information for asserting an identity's Subject such as DID, cryptographic public key, service endpoints for interacting with DID Subject, authentication mechanism, and other confirming data. A DID controller could be DID Subject mostly where the difference between them represents that DID Subject is the entity identified by DID. In contrast, a DID controller is an entity authorized to perform changes on the DID document and DID might have more than one DID controller. However, it's an optional property that can be selected in a DID document once created DID.

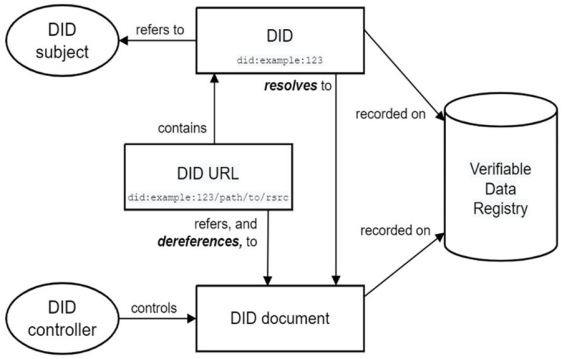

Fig 4. Decentralized Identifier Architecture

Figure 5 shows an example of a DID document encoded by JavaScript Object Notation JSON data-interchange format. Id stands for DID, which indicates a public key inserted in a DID document and shows authentication mechanisms and service endpoint that owned so-called Verifiable Credential VC. The last part represents the proof needed for ensuring DID's integrity and audit history.

```
{
 "@context": "https://w3id.org/did/v1",
 "id": "did:example:123456789abcdefghi",
 "publicKey": [{
     "id": "did:example:123456789abcdefghi#keys-1",
     "type": "RsaSigningKey2018",
     "owner": "did:example:123456789abcdefghi",
     "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
 }],
 "authentication": [{
     "type": "RsaSignatureAuthentication2018",
     "publicKey": "did:example:123456789abcdefghi#keys-1"
 }],
 "service": [{
     "type": "ExampleService",
     "serviceEndpoint": "https://example.com/endpoint/8377464"
 }],
 "created": "2002-10-10T17:00:00Z",
 "updated": "2016-10-17T02:41:00Z",
 "signature": {
     "type": "RsaSignature2016",
     "created": "2016-02-08T16:02:20Z",
     "creator": "did:sov:8uQhQMGzWxR8vw5P3UWH1j#key/1",
     "signatureValue":"IOmA4R7TfhkYTYW87z640O3GYFldw0
                       yqie9Wl1kZ5OBYNAKOwG5uOsPRK8/2
                       C4STOWF+83cMcbZ3CBMq2/gi25s="
 }
 }
}
```

Fig 5. Decentralized Identifier Document example

2.5.2 Verifiable Credentials VC

It is another standard provided by W3C for supporting such a decentralized identity framework, which is based on a JSON format for exchanging the data between web applications [13]. A recent draft of VC specification defines

the attributes associated with certain individuals and can be cryptographically verified as " Verifiable Credential". This VC is a piece of identity's information asserted by a party that attests this information to become digitally signed encoded by JSON Web Tokens (JWT). An Issuer party of this VC must be a trusted authority for issuing it only for the party requested that VC. The use cases presented in [13] shows that VC is a tamper proof credential established by Issuer and can be verifiable at specific parties. The VC in figure 6 contains metadata information about credential such as an public identifier, Issuer, expiry date and time. Besides, the required asserted claims and Issuer's proof in order to preserve the integrity of that VC from tampering.

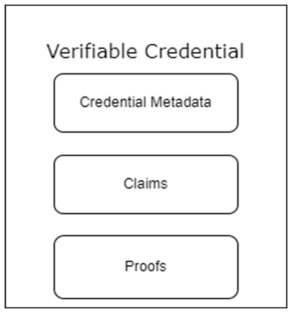

Fig 6. Basic Components of Verifiable Credential

After receiving this VC at the party requested that the so-called "Holder", it can be stored in his digital wallet for sharing later with the desired parties. The main goal of [13] is to enhance privacy so that once the Holder decides to share VC with a specific party, it will be shared with selective disclosure or minimal disclosure of information. This could occur through Verifiable Presentation VP, which assists the Holder in introducing his VC to other parties securely with his proof, as shown in Figure 7. Since the [13] mainly support the individuals to have full control over their VCs and to be stored locally in a so-called digital wallet and further decides whether or not sharing it with others. This led to removing the fundamental role of external authority or third parties to manage it on their behalf. Service providers, typically called "Verifier", can verify the VC provided by the individual or the Holder by using asymmetric cryptography. A Holder generates key pairs before sharing the VC with any party that demands it. One is used to prove VC's possession through a so-called digital signature by a cryptographic private key, and the corresponding key will be stored publicly on a blockchain/distributed ledger. By establishing a pairing of DID [12] and public key on the network, anyone of concerned parties can read DID and verify the VC given by the Holder, enabling the parties to compare the publicly available DID with that included in VC. Both Issuer and Verifier doesn't store any information about the Holder and remains their transactions only and only by blockchain/distributed ledger technology.

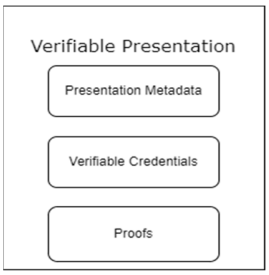

Fig 7. Basic Components of Verifiable Presentation

Due to the primary goal of W3C standards [12-13] that support user's privacy-preserving and grant the users to directly manage their identities, most data in decentralized identity framework can be stored privately or so-called "offChain storage" to have the users governed over their data or to be more closely control it, on the other hand, some of these data need to be securely stored on blockchain/distributed ledger or so-called "onChain storage" such as cryptographic public key and any other data that the user wants to reveal publicly.

2.6 Blockchain and Distributed Ledger Technology

Blockchain is a distributed and immutable ledger that can be used in different applications to remove the role of depending on a trusted authority. It is considered transaction-based that relies on a peer-to-peer P2P network for achieving such a decentralized transaction. A blockchain has been defined by ISO 22739 [14] as "synchronized distributed ledger technology storing information replicated and shared across multiple nodes using consensus mechanisms and represented as confirmed blocks organized in a sequential chain using cryptographic links". Satoshi Nakamoto in 2008 [15] introduced the first concept of blockchain by providing a technical foundation of a new type of electronic cash that is mainly relying on a P2P network. Although he doesn't refer to this technology explicitly, he mentioned its architecture as: "Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones behind it".

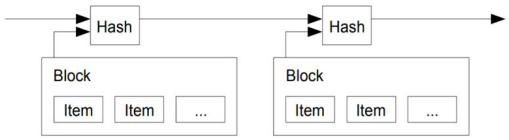

Fig 8. Nakamoto's Project Proposal

Blockchain allows the parties to perform data interchange over its P2P network by using cryptographic operation that the trust relationship builds upon with no intermediary between the parties. Transaction history

cannot be timber-with or manipulated depending upon blockchain nature, considering a decentralized P2P network that fundamentally replicates data storage in order to prevent any potential data loss. There are number of blockchain characteristics discussed in [16-20] that mainly summarized the significant blockchain features as decentralization, transparency, traceability, distrusting, anonymity, unforgeable, immutability, auditability, and validity.

2.6.1 Types of Blockchain

NIST technical report [21] classified blockchain into two types based on their architectures and permission model, which identified who can publish blocks on the network:

- **Permissionless blockchain networks:** It is an open decentralized platform where anyone of the participants can publish new blocks and participate in consensus. Although transparency features are supported in its transactions, there is some anonymity or pseudonym used by participants, which doesn't completely prevent privacy concerns. However, it is considered open-source software available to all for downloading locally. Participants can then utilize this for issuing transactions and read anything recorded into the blockchain. Since this type is open to all blockchain users, malicious participants could benefit from this for publishing blocks that harm the system. This kind of blockchain takes considers this issue by using such a multiparty agreement or 'consensus' system that adheres the participants to expend or maintain resources in case of publishing blocks. Proof of work (PoW) and proof of stake (PoS) methods are examples of such a consensus. The drawback of Permissionless blockchain is that it has lower performance for processing transactions due to the fixed size of block and nature of the huge network that affects the time of processing as well. Furthermore, focusing on preserving data integrity rather than confidentiality in most cases.
- **Permissioned blockchain networks:** it is specifically for authorized participants to publish blocks. This kind of blockchain could be relied on a completely decentralized network or centralized authority to decide whether or not participants can read the blockchain or issue transactions. Dispute restricting use to authorized participants; it can be instantiated or maintained by both opened or closed sources of software and could have the same traceability of digital assets, distributed, resilient, and redundant data storage system as in the other type of blockchain. Besides, consensus mechanisms can be used to legitimate transactions and reach an agreement between the nodes before publishing blocks. In contrast of Permissionless, it often doesn't require expending or maintaining resources for such doing so. This is because, in case, joining as a member of Permissioned blockchain requires the establishment of identity to preserve a level of trust between blockchain members and therefore could be authorized for publishing blocks or have revoked authorization in case misbehave. As a result of restricting the use of Permissioned blockchain to authorized participants, it's highly scalable, and consensus mechanisms could be faster and less computationally expensive than Permissionless blockchain. Permissioned blockchain might be a private network directed to the organizations based on their business needs where a single authority can control who can publish blocks on its blockchain. On the other hand, it can be directed to a group of organizations that are not fully trusted between them and that so-called consortium blockchain. They can be used as a shared distributed ledger to record their transactions, and they typically agree on one of the consensus mechanisms to be used.

For both blockchain types, a user plays a significant role in ensuring that his private key, which can be used for signing transactions, is saved and kept secret by storing it in his digital wallet. On the other hand, the public key seems to be an address on the blockchain and can be shared with others. When the user can't access, for whatever reason, his private key, blockchain typically does not support any method for key recovery in case of losing it. Therefore, the user will lose ownership of the information associated with that key [22]. Also, interoperability across blockchain applications could be another obstacle facing the users of this technology in which restrict sharing, transferring, and accessing between different type of blockchain and that refer to varying technological designs such as speed of transactions processing to be recorded into blockchain and this due to different size of blocks supported in these applications [23- 24]. Another obstacle is that they have varying degrees of permission-ability, mainly selecting the level of how the users can participate in the system [24].

2.6.2 Data Storage in Blockchain

There are two figures of the data storage in the blockchain: offChain and onChain. Each of them has benefits and drawbacks, as mentioned in [25-26]. OnChain storage method uses the resources of blockchain to store the data, which then be available to all the parties using the same network. Since this kind of storage is considered the simplest way of storing an amount of data in the chain itself, it could lead to scalability issues due to fixed block size. On the other hand, OffChain storage methods not necessary be a part of the blockchain network, and it introduced to enhance the privacy and scalability that mainly face the public blockchain network through achieving further increased storage by shifting this job into an external device or any various forms to perform that, besides, a verified data by using blockchain doesn't necessary to be available to the public. Although the benefits mentioned above, the link between the chain itself and external storage device location might be a challenge without using either smart contracts or distributed hash tables (DHT) [25].

2.6.3 Blockchain Structure

As shown in Figure 9, the structure of blockchain contains a chain of blocks with a header and body. Each block retains a hash value of the previous block header and timestamp, nonce, and hash of the current block. Timestamp is an auditing history for production of a particular block and nonce such a random number that can be used only once. Merkle root is a current hash of the block, which contains the root hash of a Merkle tree for all transactions stored in the block body. Ti denoted as such a particular transaction and further Hi denoted as a hash value of previous transactions Ti. Merkle root is useful to reduce the efforts for verifying included transactions in a block and that due to in case performing any changes in such as one of a transaction can show significantly varying Merkle root in which simplify helpful to check the value of Merkle root rather than checking all the transactions.

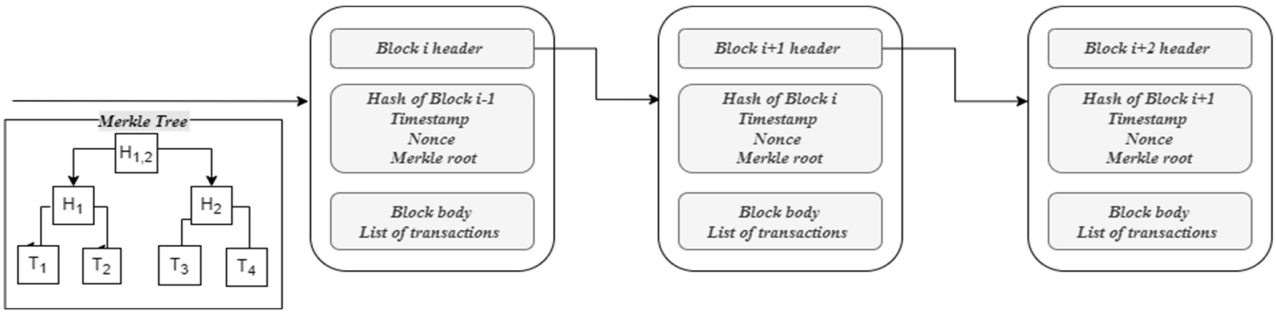

Fig 9. Basic Structure of Blockchain

2.6.4 Blockchain Applications

Numerous blockchain platforms play a fundamental role in blockchain development, especially for creating projects that aim to achieve a decentralized approach. They are briefly described below; specifically, that has commonly used.

- **Bitcoin:** It was the first earlier blockchain project that Satoshi Nakamoto mentioned in 2008 [15]. It is mainly helpful for creating decentralized payments applications built on a Permissionless blockchain in which anyone can download and verify a copy of the data whenever he wants. Therefore, users' identities will be recorded on such a public ledger; thereby, any recorded transaction will be immutable then seems hard to be deleted. Limited capacity and transactions costs are considered the main limitations in bitcoin, in which the size of a block is specifically with 1 MB, which leads to a slower transaction recording process that could be performed in minutes, and the result is a Scalability issue [27-28].
- **Ethereum:** It was introduced in 2013 as a blockchain-based software platform that aims to build so-called Decentralized Applications DApps interacting with various ledgers with support the smart contracts used for automating the execution of an agreement, thereby showing whether or not predefined conditions are satisfied [29]. Furthermore, Ethereum is developed as a public Permissionless blockchain and can be used across multiple domains. As in Bitcoin, processing a large number of transactions leads to slower speed in processing and increases the costs as well, which represents a challenging issue for scalability. [30].
- **Hyperledger:** One of the blockchain projects started in early 2016 to provide an open-source platform concerning blockchain development, built on a Linux foundation and used for both private and public blockchain [31]. The main goal of this kind of platform is to improve the efficiency and reliability of decentralized applications. This platform is hosted of five subprojects: Factory, Iroha, Burrow, Sawtooth, Indy.

- **Ripple** is another blockchain project for creating digital payments applications used by banks and other financial institutions to be built upon Permissioned blockchain [32]. Since it relies on distributed or shared ledger, it utilizes consumes mechanism to make sure transactions occur securely in a disturbed manner. Confirming transactions in Ripple use less computational power and occur in seconds with minimal cost; in contrast, confirming transactions in bitcoin that take a long processing time with higher costs.
- **Multichain:** It is an open-source platform for developing blockchain applications introduced in 2015 and designing private blockchain to support organizations within or between them to lunch and manage their custom blockchains in the financial industry [33]. This platform is derived from bitcoin in order to overcome a significant obstacle about utilizing blockchain in the financial sector by providing an opportunity for using a private blockchain, which led to enhance privacy through specifically visible blockchain's activity by chosen participants as well as resolve the issue related to scale that existed in bitcoin by enabling chain's participants of controlling the maximum block size [33].
- **Corda:** It is another open-source blockchain platform for building applications of distributed ledgers specialized in the financial sector, thereby helpful in recording and processing financial agreements. This platform supports the smart contract and could be used for a private and public blockchain. It seems similar to bitcoin in the immutable state, which clarifies that the value recorded into that ledger cannot be changed once it's created [34].
- **Quorum**: It is a developed version of the Ethereum framework that is based on the Go implementation of the Ethereum protocol (e.g. geth), which is considered as Command Line Interface CLI that uses Go language for running a node that aims to have the users ability to perform transactions as well as interact with smart contracts on that platform[35-36]. Quorum developed as a Permissioned blockchain, unlike the type of blockchain that the Ethereum used, and this enhances privacy by restricting smart contracts visibility at only the transacting parties and the peers defined in those contracts can join the network, which leads to achieving high performance for processing transactions eventually [37].

2.7 Related Works

Self-Sovereign identity has recently taken place in academia, with many research topics highlighting its concept with some of the digital identity issues it has managed to overcome. Therefore, it is considered a hot topic along with blockchain, which could be a fundamental key to its success, as it represents the underlying technology that this model depends on. In this section, we give the relevant studies that address this model in different ways. We point out the main contributions, objectives, and existing gaps for each of them and how the current study can help and support it, extending the study for SSIM solutions as well as providing a review of most identity management models. To organize this section, we start with a summary of each study, then shed light in Table 1 on existing gaps and main contributions for each of them to clarify the current study between the relevant studies in this field.

In [38], the authors provide a brief overview of the development of identity management models and then discuss two of the SSIM solutions, uPort and Sovirn. They mention three identity management models in the context of this study, namely: centralized, federated, and self-sovereign identity management models. Each of these models has been identified with its components and simple practicality. However, they point out challenging issues, like how some of those models work. One of them realizes a centralized identity management model in the organization concerned about storing the user's identities and having the ability to control it. Thus, required credentials in this model should be separated from each party that the user wishes to obtain his services. Later, this challenge was solved in the federated identity management model by introducing the Identity Provider IDP. Most contribution in this study is to propose a revised and extended SSIM specification of Kim Kameron's laws of identity [39] and Christopher Allen's Guiding Principles of SSIM [7]. The authors use those specifications to evaluate two different frameworks, uPort and Sovirn. The results show that most of the given specifications were supported for both, while a few other specifications have yet to accomplish and satisfy it and require more development and adaption of a set of common protocols and standards introduced by standards organizations. However, both solutions have a significant implementation issue represented in scalability due to the type of underlying technology that it depends on.

On the other hand, some relevant studies [40-41] relied directly on the existing literature, such as laws of identity [39] and principles of SSIM [7], to assess SSIM solutions. Those studies are almost similar in their methodology and vary based on their objectives and the total SSIM solutions covered.

In [40], the authors provide a brief overview of the concept of Self-Sovereign identity and mention some earlier contributions in this field. This study aimed to investigate the state-of-the-art developments adapted with SSIM through utilizing a decentralized identity framework that relies on blockchain technology. For the availability of technical details and scope of the study, the authors decided to select five SSIM solutions: Sovrin, uPort, ShoCard, Civic and Blockstack. Those solutions were evaluated depending on both [7] and [39]; thus, the results reveal that all the evaluated solutions have four significant shortcomings: 1) different levels of decentralization and incorporate blockchain. 2) Identity revocation. 3) Human integration. 4) Economic barriers. With regard to identity revocation and human integration, it is hard to revoke the user's associated cryptographic keys and rely on the user to keep his data securely in his smartphone or PC, which could lead to a significant issue in case of failure or loss of that device, although security features are provided by this method. However, existing SSIM solutions focused on the underlying technology, not the user interaction, e.g., Usable interface, key management, and privacy implications for users. Furthermore, they should design SSIM systems to solve the challenges faced by end-users. Then, the authors mentioned the economic barriers, which could form a challenge to using SSIM solutions. Specifically, if those solutions rely on Bitcoin and Ethereum platforms to build their systems, miners who are responsible for linking the blocks to each other will be required to reach consensus in the network by so-called Proof of Work (PoW). This mechanism helps prevent certain kinds of economic attacks. It relies on having sizable computational power, and as long as this is the fundamental technology that those solutions are based on, there must be a cost associated with usage.

The authors in [41] provide a brief overview of the evolution of digital identity models that shed light on the SSIM model and blockchain technology. They define a typical blockchain system by basically containing a set of components: Peer-to-peer network, storage, validation, consensus, and cryptography. Each of those components ensures that exchanging information and storing associated data of the user occurs in a secure manner, forming an integral part of any blockchain-based system. Besides, they briefly introduce types of blockchain in order to categorize SSIM solutions later easily. Based on the scope of the study, three SSIM solutions have been selected: uPort, Sovrin, and ShoCard. The studies mentioned its components and features, along with the operating environment on which it relies. To evaluate those current solutions, the authors utilize Kim Kameron's laws of identity [39] to better understand the differences. The results show that all solutions grant the users more control over their identities and serve the decentralizing identity approach and easier verification to multiple entities. Despite those features, the study points to a noticeable lack of contextual understanding regarding the user experience, which may form a difficult to those solutions to deliver their goals. Thus, this confirms an existing shortcoming in human integration mentioned in the previous study [40].

The authors in [42] contribute to analyze three distributed ledger-based identity management solutions: uPort, Sovrin, and shoCard. Afterwards, they use Kim Kameron's laws of identity [39] to evaluate the given identity management schemes. Each of those solutions has been described with its design, architecture, and functionality. Given that distributed ledger technology DLT plays a fundamental role in building some identity management systems, it provides significant features to such systems: decentralized, user control, transparency, etc. Based on the followed approach in this study, they found that DLT-based identity management schemes fell into two categories: Self-Sovereign Identity SSIM and Decentralized Trusted Identity DTI. Solutions that rely on SSIM, e.g. Sovrin and uPort, enable the users to own and control their identities without relying on any external administrative authority. In contrast, solutions that rely on DTI, e.g., shoCard, depend on centralizing servers to provide identity and perform identity proofing of users based upon existing trusted credentials, e.g., passport, and records identity attestations on a DLT for later validation by third parties. Since most DLT-based identity management systems aim to remove the central authority, this may not be a realistic goal because most of these systems need trust and have different levels of decentralization [40]. For example, in shoCard, trusted third parties can certify attributes of certain identifiers while uPort and Sovrin support both self-attestation of attributes and those assigned by other entities. The authors lastly point out other challenges represented by a lack of contextual understanding relating to the user experience and relying on the user to keep his data in a smartphone or PC; this could raise the risk that those users will be unable to recover resources attached to lost keys when something goes wrong [40-41]. This concept is called "key recovery" and was proposed in some DLT-based identity management systems such as Sovrin and uPort.

Although the studies of [41-42] seem to close in used framework and SSIM solutions, they vary in methodology and objectives for each. In [41], the study focuses on the most significant underlying identity management and blockchain technology concepts. In [42], the study proposed a scheme for identity management and focused on analyzing solutions that rely on the DLT to enhance decentralization, transparency, and user control.

In [43], the authors introduced a review of the essential aspects of blockchain and the SSIM model to further understand the comparative blockchain-based SSIM solutions. They decide to select Sovrin, uPort, EverID, LifeID, Sora, and SelfKey. Each of those solutions has

explained and described most features and drawbacks. However, the comparison shows that most current solutions are still evolving and need more exploration to understand their functionalities to address the current challenges. The authors have mentioned those challenges with given possible enhancements; for example, in Sora, there are basically two significant issues representing storing key pairs on a centralized server. It is not fully decentralized as mentioned in [40][42]. Furthermore, to encrypt the user's identity information, the user selects an 8-digit password, which must be a set of small letters, capital letters, and digit, forming a master key to decrypt that information. However, this could be vulnerable to various attacks due to the security of the master password. On the other hand, some of those solutions don't fulfill the minimization principle of SSIM, such as uPort and EverID, which require disclosing full information for claim verification. Besides, portability forms another requirement that needs to be enhanced in the future at Sovrin and uPort. Based on the comparative survey, they also selected other challenges, like transparency in EverID and scalability in LifeID. Although the above solutions adopted SSIM differently and obtained some inherited properties from blockchain technology, there is a need to address these challenges to build trustworthy, scalable, and provable identity systems.

In [44], the authors compared centralized and decentralized identity management systems, depending on their trust model and data storage schemes. Given that trust is an essential aim to any identity management system, they point out two types that need to be implemented in most use cases of identity management first, whether any identity owner can trust the identity management system to store the data and not perform any changes on those data without his permission. Second, whether users can trust each other based on the verification data, shows that operation-level differences between those two types of systems stem from different storage schemes. For example, in decentralized identity management systems, the data can be verified without trust to any centralized authority for both types of trust, forming a basic component in the other systems. Also, the storage scheme can vary for both, which means that storing the data can be implemented through transactions recorded in DLT in decentralized identity management. In contrast, centralized identity management systems depend on the centralized database to do so. Thus, decentralized identity management is considered more trustless than centralized identity management in its trust model. This trustlessness could be obtained at the cost of difficulties of certain identity operations such as 'deletion,' which would be more difficult in a decentralized identity system.

In [45], the study provides a brief overview of most identity management approaches and points out most features offered by blockchain in managing identity. A decentralized identity that is blockchain-powered is considered a secure way to preserve users' control over their data. The authors briefly explain the standard method of identity proof and attribute verification, which could happen when the verifier of credentials checks out that the signature of trusted authorities of the given credentials is valid and the same as what they claim. On the other hand, managing user's data are performed by keeping relevant claims offline, whereas his public identifier is kept on blockchain. Finally, they analyze some existing blockchain-based solutions and highlight their design and implementation in light of SSIM architecture. In conclusion, some blockchain-based identity management challenges have been discussed, such as elimination of Intermediaries [40][42][43], scalability, the trust required, portability, and user experience [40][41][42].

In [46], the authors conducted a comprehensive taxonomy of SSIM, categorized as taxonomies of foundation property, controllability property, sustainability property, security property and flexibility property. However, a taxonomy of controllability represents essential property forming the concept of SSIM, which permits users' consent, disclosure, and control regarding their identities. They intend to utilize those taxonomies to evaluate SSIM systems: uPort, Jolocom, Soverin, and blockcerts. The result shows that most of those systems, except blockcerts, satisfy several properties. And due to its reliance on blockchain, each of these systems is transparent, and the cost for creating transactions or storing data in their blockchain platforms might form a barrier for any wide-scale adoption [40]. Also, none of the above systems fulfills the requirements of a flexible digital identity.

Appendix 1 shows a summary of related works by mainly their contributions and existing gaps found there. Most of these studies have a common research gap in their need to extend those studies to cover more SSIM solutions. The current study will help support these studies though covering most SSIM solutions with proper technical documentation, reports, or whitepapers. Along with that common research gap, there's an absence of the main differences between identity management models in most of these studies and how these models can operate in their environment, highlighting their challenging issues. One of the current study's objectives is to bridge the gap between relevant studies by providing an academic investigation about common identity management models and discussing their main drawbacks, features, and how these models can differ. Finally, the current study aims to present most identity management issues to make SSIM solutions more understandable and clarify their features and drawbacks. Blockchain will also take place in this study, considering the underlying technology that supports most SSIM solutions and classifying these solutions to a full or partial SSIM system based on the criteria discussed later.

3. Overview of Identity Management Models

3.1 Silo Centralized Identity Management Model

The Isolated User Identity centralized model (i.e. Silo model) forms the simplest and the oldest identity management frameworks, where the individual's identity is stored and controlled by the organization and credentials data should be separate for each organization that the user wishes to access the service from. Since that organization considering as both service and identity provider, it will be controlled by all mechanisms for the authentication in which allow to establishing confidence in an identity claims' truth, and also has the ability to control the level of access and permissions ' authorization ' through deciding what the user should be allowed to do in order to obtain their services. Thus, the design of centralized identity management frameworks primarily benefited the service providers 'organization' rather than the end users and that due to separate credentials that the user should be managed during utilizing these systems and resulting from that have numerous partial identities with corresponding identifiers, which form a difficult to manage them manually.

3.1.1 Identity Proofing

Since the user adheres to keeping distinct identifiers for each SPs he communicates with, he should manage a lot of these identifiers aiming to make sure that doesn't repeat for any trustworthy relations with concerning parties. This model supports a single factor authentication method; typically, it uses shared secrets represented in username and password. SP authenticates the user's identity through utilizing that application layer technique; in contrast, the user agent authenticates the SP through lower layer technique, e.g. SSL/TLS.

This method doesn't require any proof of identity during the registration process, whereas the options of shared secrets permitted vary in the authentication process, including username, password, birthday, etc. In general, this model doesn't support using any cryptographic methods to prove the assertions of identity at both registration and authentication processes.

3.1.2 Use Case

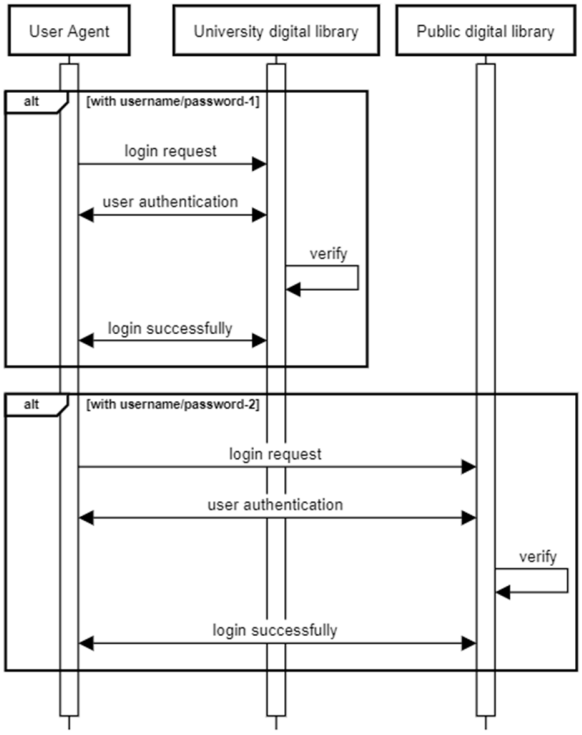

Fig 10. Silo model

Suppose that the user wants to access the resources of his university digital library, so-called SP1/IDP1, with a credential issued by it. Once the user agent provides a login request, SP1/IDP1 demands his credential, which has been stored locally in the SP1/IDP1 server. After responding by the required credential, SP1/IDP1 perform a verification process for whether or not to allow access. When the user needs to extend his search by using public digital library SP2/IDP2, he should utilize that credential which is defined and stored at SP2/IDP2. In this case, the user adheres to performing an independent authentication process once he wishes to obtain access to the resources of these relying parties. These examples are sketched in Figure 10.

3.1.3 Advantages

It primarily helps the organizations to preserve a high level of compliance and useability by keeping and managing the data in-house and directly controlling all users' associated data rather than relying on a third party. Also, ensure that credentials don't be reused when the user needs to communicate with other service providers, which leads to enhancing the security aspects regarding dealing with different credentials for different trust relationships.

3.1.4 Disadvantages

It's hard for the user to deal with each trust relationship with different credentials, which leads to having a lot of it that needs to be memorized and then affecting the user experience through shifting the burden of credentials management on the user. In this case, he adheres to satisfying different password requirements given by varying SPs. Also, this model supports authentication in session-based rather than persistent and from the perspective of security and privacy for storing the individual's data in such local central servers; this could raise a major concern regarding the risk of misuse of private data and might be vulnerable to various attacks. Finally, portability considers the fundamental issue in this model since it doesn't support the cooperation between the SPs to share the attributes about a single user. This refers to being trusted by only itself, which leads to playing two roles simultaneously: SP and IDP.

3.2 Federated Identity Management Model

With the rapid technology change, Federated Identity Management systems take place for processing the relevant identity data in a new method aiming to separate the process of managing the credentials from the service providers to a third party (e.g. IDP). The fundamental role of this party is storing and issuing the identity's attributes to the service providers SPs that utilize these issued attributes to allow a certain user to access their services. An IDP must be trusted by all SPs who are defined in such a circle of trust that typically contain a single IDP and multiple SPs. However, a trust relationship is built depending on their signing policies and agreements, which explain the requirements and responsibilities for communicating between them. Since those agreements ensure that a particular identity is issued in a given circle of trust, all SPs included will recognize it as long as it has an IDP-issued identity.

This model supports the Single Sign On 'SSO' feature, which improves the user experience by allowing the user to access multiple domains after being authenticated with a trusted single IDP during a session. Therefore, most federated identity management systems that support this feature also support the 'Single Sign Off' feature, allowing the user to just sign off once. This process reflects automatically to all accessed SPs. However, managing numerous partial identities could be easier than before, resulting in shifting the registration and authentication of certain identities to be the responsibility of the IDP.

The essential property in this model is mainly the support identity federation process, aiming to authenticate the users by IDP and transfer the asserted credentials required to authorize their access at SPs. To ensure that, the SP will discover whether or not the user has the necessary assertion given by IDP, which has asserted his identity throughout using the 'common domain cookies' technique [47]. Since the identity federation needs to have a unique identifier to point out a specific user during communication between IDP and SPs, anonymity is supported in most of the systems built on this model to provide unlikability of certain user's identity through utilizing a pseudonym. Both SPs and IDP must have a shared agreement regarding using that pseudonym as a reference to a particular user in the federation process.

3.2.1 Identity Proofing

Once the user is authenticated with the IDP during a single working session, there's no need to repeat this process at the SPs that have the same circle of trust. Moreover, SP can verify that the process is valid and performed with a trusted IDP by transferring the user agent, e.g. Web browser, to that IDP after the user demands access to its resources. If the user has already valid authentication with that IDP during its existing working session, in this case, IDP redirects the user agent back to the SP with the security credentials, which is a result of this action and proof certain user has been successfully authenticated with the IDP.

Most federated identity management systems support three methods for identity proofing in which the user can prove his identity to any party that requires it. These methods are listed below, along with a brief description of each [47-48].

- **Bearer**: It is one of the identity proofing methods that any party can use to prove his identity due to no cryptographic evidence provided to SP during forward the security credentials. When selecting this method based on the agreement between SP and IDP, every entity conveying such identity is valid. In this case, SP adheres to accept that any user forwards the security credential is that rightful owner. This increases the risk of stealing security credentials from the attacker to gain access to the resources of SP.
- **Holder-of-key**: is another method for identity proofing that utilizes cryptographic evidence to prove the rightful possession of specific assertion given by the user and intended to access the resources of SP. The assertion will include a cryptographic key as a reference of that user and encrypted by a method ensuring that anyone can't decrypt it except concerning that SP. Two methods could be used to achieve this: symmetric and asymmetric approaches. If a symmetric approach is used, the symmetric key must be included inside the requested assertion provided by IDP and must be signed before sharing it with any party who demands it. Afterwards, the user proves the

rightful possession of that assertion though introducing that key to SP. On the other hand, in the asymmetric approach, the user's cryptographic public key must be included in the signed assertion; thereby, the user can utilize the corresponding private key to prove the rightful possession of the key referenced in that assertion to SP. For both cases of symmetric and asymmetric, usually, it's more difficult for the attacker to use stolen assertion proofed by the holder-of-key method given that he would need to steal the cryptographic key as well.

- **Sender-Vouches**: this method can be used if the holder of a certain assertion has permission to present that assertion to SP on its behalf of the subject's assertion. Even though this assertion is already asserted by IDP whereby the holder can present it to another SP it must be signed by the holder itself before forwarding it to a specific SP.

The recent draft of NIST [48] provides varying levels of federation assurance which benefit to ensure that authentication occurs correctly and is used by validated parties in a federated environment. This draft assists the organizations to select the most appropriate requirements for proving the user's identity based on their needs and aims to provide technical guidelines for performing attributes sharing and parties communicating about certain identities securely. Therefore, these levels describe the requirements of how assertions for a given transaction can be secured depending on the request of SP or be required due to sharing configuration between SP and IDP. However, federation assurance levels are categorized into three levels:

1. Allows the user to gain the SP to receive a bearer assertion that IDP must sign using approved cryptography.
2. Adds a further requirement that the assertion must be encrypted through it and utilizing approved cryptography such that SP is the only party that can decrypt it.
3. Requires the user to introduce proof of possession of the cryptography key referenced in certain assertion. The assertion itself must be signed by IDP using approved cryptography and encrypted to SP.

3.2.2 Use Case

As in Figure 11, suppose that the user is a fresh employee and needs to access the Ministry of Justice SP1 resources to start his job. After he sends the login request, SP1 already redirects the user agent to IDP, the so-called "Nafath" with the same circle of trust for performing the authentication process. The user can also visit the website of IDP first for doing this process, then pick which SP wishes to access its resources during the current working session. Based on the above scenario, the user provides his credentials to IDP for authentication purposes. In case of success, IDP redirects the user agent back to SP1 with a SAML assertion response that holds the identity proofing data and is digitally signed by IDP. Both IDP and SPs can identify which trusted parties the user visited or came from through the 'common domain cookies' technique [47]. This technique is a reliable means for discovering the identity of

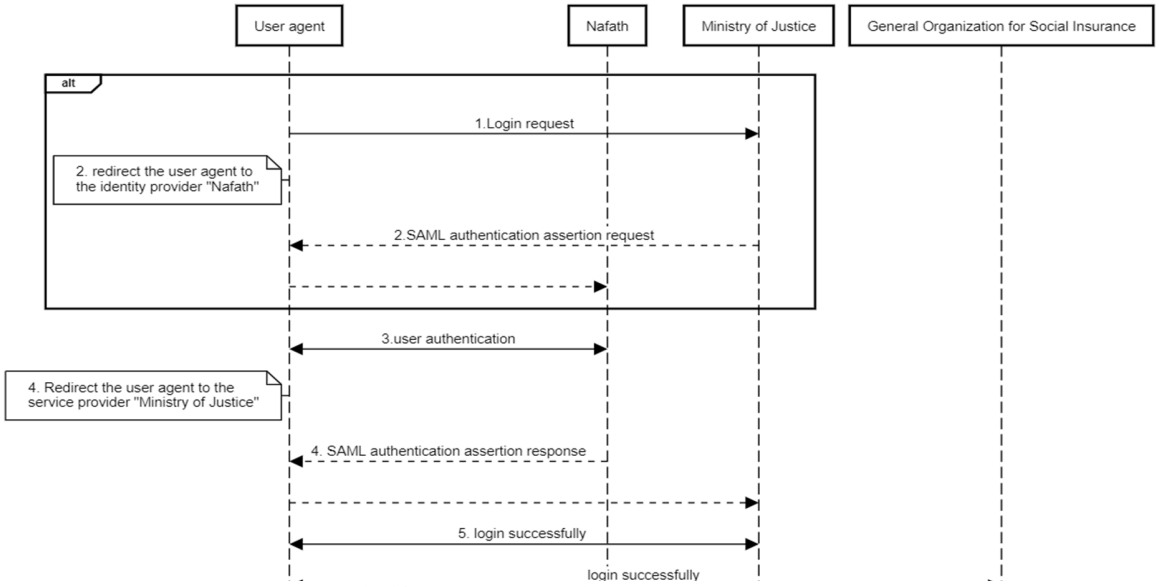

Fig 11. Federated Identity Management model

concerning parties involved in a specific circle of trust where any of them can read cookies written by any other party. Afterwards, SP1 will check the information included within the SAML assertion and verify the IDP signature to decide whether or not to accept this assertion. If so, a fresh employee will be logged in to the resources of the Ministry of Justice, and in case he needs further information provided by the general organization for social insurance regarding his job, it will be allowed to obtain its services during this working session without needing for another authentication process and that due to SSO feature supported in this model. Figures 12 and 13 show examples of SAML assertion request and response massages.

```
<samlp:AuthnRequest
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="identifier_1"
 Version="2.0"
 IssueInstant="2004-12-05T09:21:59Z"
 AssertionConsumerServiceIndex="1">
 <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
 <samlp:NameIDPolicy
 AllowCreate="true"
 Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>

</samlp:AuthnRequest>
```

Fig 12. SAML assertion authentication request example

```
<samlp:Response
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="identifier_2"
 InResponseTo="identifier_1"
 Version="2.0"
 IssueInstant="2004-12-05T09:22:05Z"
 Destination="https://sp.example.com/SAML2/SSO/POST">

 <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
 <samlp:Status>
 <samlp:StatusCode
 Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </samlp:Status>
 <saml:Assertion
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="identifier_3"
 Version="2.0"
 IssueInstant="2004-12-05T09:22:05Z">
 <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
 <!-- a POSTed assertion MUST be signed -->
 <ds:Signature
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
 <saml:Subject>
 <saml:NameID
 Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
 3f7b3dcf-1674-4ecd-92c8-1544f346baf8
 </saml:NameID>

<saml:SubjectConfirmation
 Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
 <saml:SubjectConfirmationData
 InResponseTo="identifier_1"
 Recipient="https://sp.example.com/SAML2/SSO/POST"
 NotOnOrAfter="2004-12-05T09:27:05Z"/>
 </saml:SubjectConfirmation>
 </saml:Subject>

<saml:Conditions
 NotBefore="2004-12-05T09:17:05Z"
 NotOnOrAfter="2004-12-05T09:27:05Z">
 <saml:AudienceRestriction>
 <saml:Audience>https://sp.example.com/SAML2</saml:Audience>
 </saml:AudienceRestriction>
 </saml:Conditions>
 <saml:AuthnStatement
 AuthnInstant="2004-12-05T09:22:00Z"
 SessionIndex="identifier_3">
 <saml:AuthnContext>
 <saml:AuthnContextClassRef>
 urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
 </saml:AuthnContextClassRef>
 </saml:AuthnContext>
 </saml:AuthnStatement>
 </saml:Assertion>
 </samlp:Response>
```

Fig 13. SAML assertion response example

3.2.3 Advantages

This model solves the portability issue existing in the previous model by allowing sharing the identity's attributes of a single user between SPs that have a circle of trust. Also, the identity federation feature improves the user experience by allowing the utilization of certain identities in multiple domains. Given that identity is authenticated with IDP during a session only once, users can access one or more SPs that have trusted relationships through enabling SSO. This feature further aims to support useability and anonymity during the identity proofing and authentication process, which doesn't support global public identifiers in most cases. Most federated identity management systems support the feature of minimal disclosure of information by allowing the SP to identify which identity's attributes need to be asserted by the IDP without revealing any attribute that isn't necessary to confirm the user identity. The required attributes will be included in the SP's request to that IDP, which is involved in the same circle of trust.

3.2.4 Disadvantages

Dispute the provided features in this model such as useability, security features represented in support different methods of authentications and ensure that user's data cannot be shared between service providers without referring to the identity provider, possible tracking the users' activities by identity provider is still a concern along with that misuse their private data. Besides, in most federated identity management systems, IDP discovery performs on the SP server, leading to significant risk regarding stealing the security credentials due to redirecting the user to a fake IDP website. Also, in most cases, the users have limited or maybe no control over their identities throughout exchanging the relevant attributes between SPs and IDP. Additionally, due to the centralization approach which the systems following this model relaying it, there's a risk for potential attack representing in compromise a trusted single issuer 'IDP' where affecting directly to concerning parties that are dealing with. A further disadvantage is limiting the data protection role to a single point of trust and control 'IDP' [49]

3.3 Self-Sovereign Identity Management Model

In recent years with rapid growth in technology, a new identity management model has appeared called a 'Self-Sovereign Identity' model. Christopher Allen [7] defined this model by introducing ten principles which have been mentioned in section 2.3 previously. The main goal was to give the users all the rights to manage their identities without depending on any party. User's privacy plays a crucial role in proposing this model where everything that refers to a certain user is considered sensitive data and must

be shared in a secure manner with concerned parties. This model is based on a decentralized approach for managing identity, thereby, no direct communication between SP and IDP for confirming user's credentials and shifting this job to blockchain/distributed ledger technology [50].

The decentralized approach improved the exchanging information technique in the federation identity management model by performing the peer-to-peer method. This allocates information across multiple nodes, alternatively concentrating it on a single server [51]. In centralized approaches, updating and storing information remains the responsibility of a single party, whether it is the IDP in the federated identity management model or the SP in the Silo centralization identity management model and, therefore, the information repository placed in single central servers for both these models. Having an identity in such the provider-centric models means that individuals can't manage their identities and, therefore, have no autonomy and ownership on that.

Most features provided by the SSIM model are caused by supporting the blockchain/distributed ledger technology. Depending on this, this technology's drawbacks will also be reflected in this model and, therefore, essentially affect it. However, SSIM architecture adheres to participation parties to have a kind of trust that is mainly based on blockchain/distributed ledger to communicate with each other before requesting the verification of certain identity's attributes already stored by the user in his digital wallet. Thus, once the user intends to obtain a particular online service from the SP named a Verifier in the SSIM model, he has to forward the required credentials to obtain access authorization of the selected service to the Issuer, which is the IDP aiming to confirm these credentials by its digital signature. The user can then prove its possession of conveyed credentials to the Verifier by utilizing an appropriate cryptographic mechanism. Despite the trust relationship between the Issuer and Verifier, they shouldn't reveal their identities to each other and interacting between them remains to utilize blockchain/distributed ledger.

3.3.1 Identity Proofing

Based on the adopted standards by W3C supporting this model [12-13], there are various ways to prove a certain user's identity. These standards proposed a new method for achieving that decentralized identity framework, which aims to exchange information about a specific entity with confirming that it is genuine. This can occur due to the complex mechanisms of cryptography which assist to proof authentic claiming a user about himself, thereby, no need for disclosing his data actually to the desired party SP simultaneously it remains kept secret through employing these methods. Depending on that agreed standards [12-13], there are two methods for identity proofing described below:

- **Public-Key Cryptography (PKI):** also known as asymmetric encryption that uses a pair of keys, e.g. public and private, used for authentication purposes in order to proof an identity in a specific domain. Generally, the required data that need to perform such a method to access a resource of a certain party must be encrypted by its public key before sending it, and therefore, the target party can successfully decrypt it by corresponding private key. In SSIM, the method of public-key cryptography is mainly beneficial to ensure that authenticity of the holder of identity in which the relevant data of it must be digitally signed by the private key and another key can be publicly available on the blockchain/distributed ledger network for allowing the Verifier or SP to ensure the truth of certain identity. This could be clarified by using the DID document, which contains metadata for describing and interacting with an entity. Metadata can include a public key, authentication mechanism, and service endpoints.
- **Zero-Knowledge Proofs (ZKP):** is a method of identity proofing that allows one party, e.g. Prover, to prove knowledge of a secret to another party, e.g. Verifier, without revealing the knowledge that it has [13]. In SSIM, the identity holder can prove its possession of his credentials to the Verifier using Zero-Knowledge proof without disclosing the actual value. This could be achieved by verifiable presentation if the holder obtained a verifiable credential from the issuer included with his signature. Afterwards, supporting selective disclosure of information enhances privacy by enabling the holder to prove the validity of signature without revealing actual values that they signed for.

3.3.2 Use Case

The study simulated the previous scenario in section 3.2.2 to be reflected in the use case of SSIM, as shown in Figure 14. The current scenario adheres the user to have a Verifiable Credentials VC issued by a trusted party, an "Nafath" that will be responsible for making sure the user's claims about himself before granting any VCs. First, the user will present his credential (e.g. national identification number) that needs to be asserted by Nafath. Afterwards, if that credential has been successfully asserted by Nafath, a user will get all the VCs that he needs to obtain an access to ministry of justice, for example, criminal record, valid National ID, traffic violations record, etc. Considering an Nafath is a recognized party to keep all this information in Saudi Arabia, it will perform an authentication process with that user before awarding the VCs that the user needs. Once it obtains a successful authentication with the user, the Issuer establishes a DID for the VCs that the user requested before and registers it on the blockchain through the DID document, which contains that DID, a public key for the verification, service endpoint for interacting with that entity or so-called DID subject, a timestamp for audit history, a signature for integrity, and authentication mechanism. Afterwards, the Issuer can send the VCs to the user, which is mainly signed by the Issuer's private key and then the user can store these VCs in his digital wallet for sharing later with any SP demand so. However, in case the user decides to share some or all his issued VCs with a specific party, for example, the Ministry of justice, which is the Verifier of the provided information, he doesn't share the actual values of his issued credentials and instead of that using so-called Verifiable Presentation VP to perform so. A VP can help achieve "zero-knowledge proofs", which fundamentally preserve the user's privacy by proving the VCs' possession without revealing its actual values. Before sharing the VCs with any party, the user creates another DID specifically for those relationships to be registered in the blockchain. A VP contains one or more VCs that the user picks to share with a Verifier and might be issued by multiple Issuers. Besides, VP contains proof given by the user that utilizes his private key to sign VP and therefore prove that the user who is sharing this VP is also the credential subject that the Issuer issued these VCs for. The main difference between VC and VP is that the VC is signed by the Issuer, whereas the user will sign the VP in order to prove the right of possession of the VC given by that Issuer. For verifying that, DID that included in both VC and VP could be an address of public

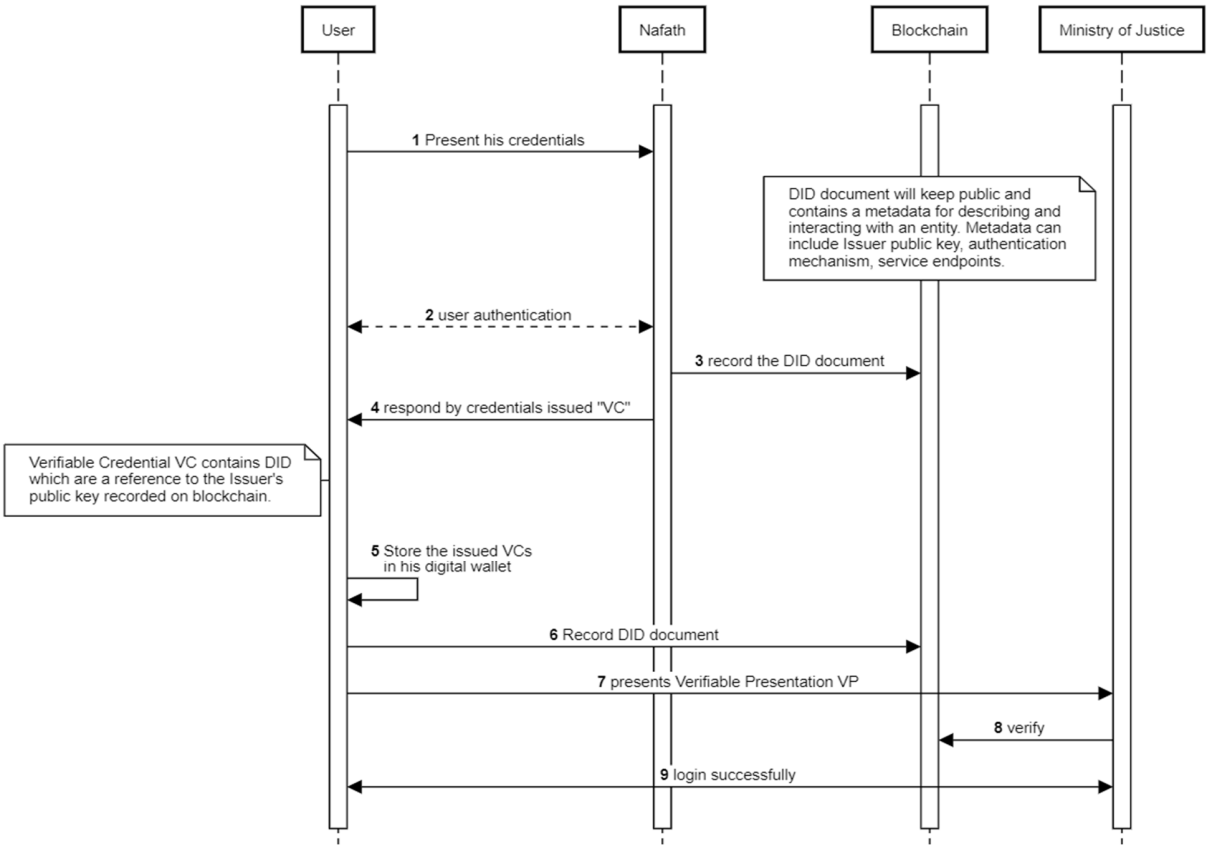

Fig 14. Self-Sovereign Identity management model

key upon a blockchain that allows the Verifier to use it to find out who issued the VC, a fact of issued this VC to the user itself, and if the required credential has tamper-with or changed by the user and have still valid upon receipt. Once the Verifier or Ministry of Justice confirms that user's credential, the user can obtain access to its resources after exchanging the required credential using the SSIM model. Figures 15 and 16 show examples of a VC and a VP.


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.com/credentials/4643",
  "type": ["VerifiableCredential"],
  "issuer": "https://example.com/issuers/14"
  "issuanceDate": "2018-02-24T05:28:04Z",
  "expirationDate": "2018-03-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:abcdef1234567",
    "name": "Ahmad",
    "NationalIdStatus": "valid",
    "criminal record: "null",
    "traffic violations record": "null"
  },
  "proof": {  }
}
```


Fig 15. Verifiable Credential VC example


```
{  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ], "type": "VerifiablePresentation",
  "verifiableCredential": [ {  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ], "id": "http://example.com/credentials/4643",
  "type": ["VerifiableCredential"],
  "issuer": "https://example.com/issuers/14"
  "issuanceDate": "2018-02-24T05:28:04Z",
  "expirationDate": "2018-03-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:abcdef1234567",
    "name": "Ahmad",
    "NationalIdStatus": "valid",
    "criminal record: "null",
    "traffic violations record": "null"
  }, "proof": {
    "type": "RsaSignature2018",
    "created": "2018-02-24T05:28:04Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://example.com/jdoe/keys/1",
    "jws": "eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0I119
      ..DJBMvvFAIC00nSGB6Tn0XKbbF9XrsaJZREWvR2aONYTQQxnyXirtXnlewJMB
      Bn2h9hfcGZrvnC1b6PgWmukzFJ1I1H1dWgnDIS81BH-IxXnPkbuYDeySorc4
      QU9MJxdVkY5EL4HYbcIfwKj6X4LBQ2_ZHZIu1jdqLcRZqHcsDF5KKylKc1TH
      n5VRWy5WhYg_gBnyWny8E6Qkrze53MR7OuAmmNJ1m1nN8SxDrG6a08L78J0-
      Fbas5OjAQz3c17GY8mVuDPOBIOVjMEgh8lgl3nOi1ysxbRGhHLEK4s0KKbeR
      ogZdgt1DkQxDFxxn41QWDw_mmMCjs9qxg0zcZzqEJw"
  } }], "proof": {
    "type": "RsaSignature2018",
    "created": "2018-09-14T21:19:10Z",
    "proofPurpose": "authentication",
    "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#keys-1",
    "challenge": "1f44d55f-f161-4938-a659-f8026467f126",
    "domain": "4jt78h47fh47",
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0I119..kTCYt5
      XsITJX1CxPCT8yAV-TVIw5WEuts01mq-pQy7UJiN5mgREEMGlv50aqzpqh4Qq_PbChOMqs
      LfRoPsnsgxD-WUcX16dUOqV0G_zS245-kronKb78cPktb3rk-BuQy72IFLN25DYuNzVBAh
      4vGHSrQyHUGlcTwLtjPAnKb78"
  }}
```


Fig 16. Verifiable Presentation VP example

3.3.3 Advantages

Since the previous models have concerns about availability, trust, and privacy, SSIM was proposed in order to overcome. Most SSIM features revolve around its decentralization approach, which grants the users a crucial priority to manage their identities without depending on the intermediaries to do so. This mainly improved the method of exchanging information existing in previous models, which focus on the providers rather than users to govern an update and store their identities. One of further SSIM features is supporting the verifiable credentials and decentralized identifiers used for exchanging information, which basically performs on blockchain/distributed ledger environment to ensure cryptographically provable identity verification that makes tampering mathematically infeasible.

The nature of distributed data control shows no single point of trust nor single point of failure, as seems in centralized approaches that rely on external authority to organize their transactions or data. Therefore, enhancing user's privacy and providing security during identity management was one of the main motivations for designing this model, where it mainly supports significant properties such as selective disclosure of attributes, portability, controllability, autonomous and further attempting to achieve the balance between transparency and anonymity. Based on standards supporting SSIM, usage cryptographic proof is enabled in case of interacting with the desired parties where these parties can refer to a shared platform (e.g., blockchain/distributed ledger) for verifying an identity and utilizing a challenge-response mechanism.

3.3.4 Disadvantages

Since the responsibility of managing identity has to be in the hands of users by allowing them to keep their partial identities in their digital wallet, there's no clear mechanism for dealing with the challenge of lost, stolen, or broken digital identity [40][41][42]. However, most SSIM features are due to the underlying technology used (e.g. blockchain/distributed ledger). Therefore, privacy and data protection safeguards remain in technology resulting in that any drawbacks will be reflected on SSIM as well. For example, the architecture of blockchain offers an immutable nature of transactions and data, which could have a significant concern regarding the removal of data when the user wishes to perform that [52]. Scalability is another challenge in blockchain/distributed ledger technology, such as increased latency of transactions causing an unacceptable performance that will slow down the overall process [53]. Besides, hard interoperability between SSIM systems due to varying the kind of blockchain framework used in these systems. This can affect the adoption of SSIM to be widely used.

4. Comparison of Identity Management Models

Most identity management models' differences have been clarified in Table 1 in attempting to define the main comparison points used between them. Furthermore, Table

2 explains some possible risks that could occur on such models, describing each one in the brief figure.

Table 1. Comparison of Identity Management Models

Comparison Point	SILO	FIM	SSIM
Scalability	Supported	Supported	Typically not supported and depends on the kind of blockchain
Selective disclosure	Not supported	Typically not supported	Supported
Recover lost encryption keys	Not supported	Not supported	Typically not supported
Anonymity	Not supported	Supported	Supported
Discovery of IDP	Not supported	Typically performed on SP server	Performed on decentralized network e.g. blockchain
SSO feature	Not supported	Supported	Not supported
Trust between parties	Mandatory between the organization and the user.	Mandatory among the partners inside a single circle of trust.	Not mandatory.
Token	-	SAML assertion token	JSON Web Token (JWT)
Approach	Isolated centralized approach	server-centric centralized approach	User- centric decentralized approach
Identity Proofing methods	Username /password	Holder of key – bearer- Sender- Vouches	Public key cryptography- Zero knowledge proofs
Main feature	Low cost for the SPs	User convenience	User control and high security

Table 2. Possible Risks of Identity Management Models

Risk		Description
SILO	**Identity theft**	When the user uses the same credentials for multiple SPs, and the attacker has control of one of these SP's servers, it will be able to steal the credentials of that user and obtain access to all SPs as well.
	Single point of failure	Occur If SP's central server used for storing the data is being compromised, all the data can be managed by the attacker in this case.
	Weak authentication mechanism	Due to relying on single factor authentication, e.g. username/password, that facilitates the way forward an attacker to use different ways to break the authentication process, e.g. brute force attack.
FIM	**Malicious server**	In most FIM cases, IDP discovery occurs on the SP server,, which leads to increased security concerns in case malicious SP redirects the user agent toa a fake IDP website.
	Single point of failure	As in any centralized approach, if the IDP server is being compromised, the stored data will be at threat of being stolen or misused, such as accessing all members in its circle of trust.
	Identity theft	In case using the bearer method for identity proofing, the SP will accept any assertion given by its holder, considering as has the right to hold this assertion which could raise significant concerns in case using a stolen assertion.
SSIM	**User misuse**	Due to having the user's full control regarding his identity and enabling him to store it off-chain in such as digital wallet, this could raise the risk in case this user has misused or lost the required proofs relevant to his data, e.g. private key which considered as a guarantee of user ownership of the data. So far, there's no way to address this issue and nearly impossible to recover it.
	High latency	It occurs when the system of SSIM relies on one of the blockchain permissionless platforms that are basically open for the public and thus have high latency.

5. Comparison of Self-Sovereign Identity Management Solutions

In this section, the study built the comparison criteria that are used as a documenting guide in classifying SSIM solutions. These are based on the feedback of the previous chapter regarding the blockchain and main points of comparison of the identity management models, which discussed the differences for dealing with an identity in clear insight and then clarifying the related issues that directly affect using those models. All these have been taken into account during creating our criteria; additionally, some other criteria were inherited from the literature reviews besides proposing part of them. Each criterion has to be described along with expected results for using it through analysing SSIM solutions. The second section of chapter three is to clarify the search process of SSIM solutions and further examines them based on the introduced criteria.

5.1 Comparison Criteria

Since the SSIM model turns around the ownership of data and how the users can own their identities and manage them independently, the study in this section will define the comparison criteria that would be useful to investigate the claim of current SSIM solutions about the users' identities. The study aims during this comparison to assess three primary aspects of SSIM solutions. First, the claim of having the users' full control and ownership over their identities. Second, the claim of removing the role of such a third party to have or control the data associated with users' identities. Third, the claim of having such a decentralized identity framework by utilizing blockchain in those solutions and the possibility of making the identity portable over other identity frameworks. The above aspects in SSIM solutions motivated the study to identify the below criteria.

- **Fully/Partially SSIM Solution:** This criterion will classify current SSIM solutions into two categories: full or partial SSIM solutions, aiming to investigate whether or not the user has the ability to control his identity by himself. This criterion has been inherited from the [44], and the current study intends to utilize it to evaluate the user's control over each identity core operation and whether or not he can perform each of those by himself. Besides, this criterion will assess the ability of the identity owner to grant permission to other users by himself as well as the ability of a third party to store a user's data considering that it is a trusted authority. These will be helpful in classifying SSIM solutions into full or partial solutions depending on the SSIM model's goal which is having complete control to the users to manage their identity.
- **Blockchain Application:** Since the blockchain is considered an underlying technology for the SSIM solutions, the study mentioned in sections 2.6.1 and 2.6.4 various types of blockchain and applications that might have differences functionally. The study presents some blockchain issues that fundamentally differ for each in those sections. However, since current solutions rely on blockchain to obtain such a decentralized environment, the study sees that those issues would also be inherited in those solutions. Therefore, this criterion aims to figure out the blockchain application used for each solution and highlight whether or not it could affect the user experience.
- **Identity Proofing approach:** Most current SSIM solutions use one or all identity proofing methods supported in the SSIM model. Those methods are powered by the proposed standards of the SSIM model [12][13], which mainly support the users' control over their identities. This criterion will assess the user's ability to proof himself independently by utilizing such those methods and figure out if current solutions utilize any further methods during the identity proofing process.
- **Support Identity Core Operations:** The identity core operations involve creating, updating, and deleting part or all identity data in identity management systems. However, making these core operations available will provide value for any identity management system. Unlike that, it will directly affect the useability of a system since the change that requires by a user regarding his identity cannot occur completely form. This criterion will assess the ability of current solutions to support the identity core operations mentioned above.
- **OffChain/OnChain Storage:** Due to relying on blockchain technology as a decentralization network in most SSIM solutions, there are two methods for storing the user's data represented in "onChain" and "offChain". If the data is stored on the blockchain, that means a system utilizes "onChain" storage to keep those data available for a user. On the other hand, if the system relies on external physical storage without including the blockchain to perform that, this is called "offChain" storage [25]. According to the SSIM model, once a user keeps his data stored locally in his digital wallet,, it will help to govern the place where the data would be stored inside it. This criterion will assess the user's ability to control data stored in SSIM solutions.

- **Full/Partial Decentralization:** Despite relying on the blockchain in most current SSIM solutions to overcome the issue of a single point of failure and control, some of those solutions have a different level of decentralization due to not completely removing the role of the third party, as mentioned in [40]. The study in this criterion aims to investigate the claim of removing the role of central authority that might perform some activities on behalf of the users and preserve some of the user's private data. However, the study under this criterion will classify SSIM solutions into full or partial decentralization based on completely removing the role of a third party and achieving a complete decentralized identity framework that the SSIM model aims to accomplish.
- **Portability:** Although most current SSIM solutions rely on W3C standards: Verifiable Credential (VC), and Decentralized Identifier (DID) [12][13], to facilitate the portability among different environments that utilizes the same infrastructure of SSIM. The study aims by utilizing this criterion to investigate whether or not a user's identity might be portable over other environments that are not consistent with [12] and [13] and utilizes a previous version of identity management models. Accordingly, the study believes that a solution that can satisfy this criterion would be more attractive for using than the other solutions that cannot accomplish that for a user.
- **Cost:** Since current SSIM solutions rely on a blockchain environment, there is a cost for publishing each transaction on the network. Various kinds of blockchain motivate the study to assess if there is a further cost of utilizing those solutions along with the transaction cost fee, which might make the users rethink about using such those solutions.

5.2 Search Process

5.2.1 Data collection

Three methods are used to collect the data for the current study that aims to figure out most of the SSIM solutions published recently to satisfy the requirements of self-sovereign identity. Those methods facilitate selecting the SSIM solutions that the study needs to investigate how those solutions employ its infrastructure to shift the identity control from the central authorities into the holder itself and how it can be integrated with SSIM. Data collection methods involve: Searching through databases (IEEE, Springer, ResearchGate, SDL, Google scholar), searching by using the keywords which were determined by (self-sovereign identity, blockchain-based identity management, decentralized digital identity, emerging identity solutions), and inspection of the reference lists of the related studies in section 2.7.

5.2.2 Selection process

The search process resulting from the data collection is 100, and each result has been evaluated to determine whether or not it can be included in section 5.3. The result of the search process can be reduced by going through two stages to select the SSIM solutions.

1) Initial selection based on related topics "self-sovereign identity platforms", "blockchain-based identity management solutions", "decentralized identity management systems".
2) Elimination of some SSIM solutions depending on the filtering criteria.

We state our filtering criteria to exclude SSIM solutions that are neither related nor have enough references to rely on through studying these solutions. Filtering criteria aim to make this study applicable and assist in accomplishing its objectives. The filtering criteria involve 1) the relevant solutions mainly provide a proper platform to manage the identity by using the blockchain and promises the users to have a complete control over their identities. 2) the availability of references such as technical documentation, whitepapers, or technical reports that are publicly published. Some private organizations produced their technical solutions that are fundamentally adapted to SSIM, but using them is limited inside these organizations, which makes their solutions not publicly available and then not available enough references to clarify the main architectures or how these solutions work. 3) Selection of the launch date of these solutions to be around 2016 – 2022 to shed light on the solutions that were introduced after the concept of self-sovereign identity appeared in 2016 [7].

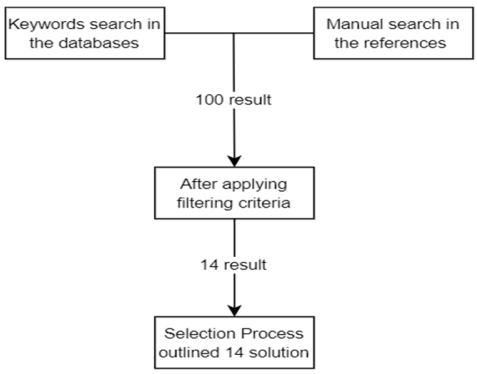

Fig 17. Results of selection process

The study in figure 17 presents the result of a selection process that begins with 100 search results throughout the

related topics in stage one and it has been highlighted in Appendix 2. After excluding some of these solutions based on filtering criteria, we obtained 14 solutions at stage two. Table 3 clarifies those solutions outlined from the selection process as well as the launch date for each.

Table 3. SSIM solutions

Project		Launch date	Project		Launch date
	Sovrin	2017		Identity.com	2019
	uPort	2017		Blockstack	2018
	EverID	2018		ShoCard	2017
	LifeID	2019		Jolocom	2022
	Sora	2019		Dock	2021
	Selfkey	2018		Sphere Identity	2017
	Civic	2017		NuID	2017

5.3 Self-Sovereign Identity Management Solutions

5.3.1 Sovrin

Sovrin [54] is a decentralized public permissioned network that aims to provide a new approach for managing digital identities and overcoming most of their online issues. This solution has been developed by the Evernym company [55] primarily to meet the emerging SSIM model through the Sovrin foundation, which is non-profit. Afterwards, the codebase has been upgraded into a Linux foundation to become the Hyperledger Indy project [56]. It supports decentralized identifiers DID and exchanging the verifiable credentials VC to empower the users to have the needed control about their identities. Sovrin architecture is fundamentally custom development that developed its trust framework to achieve governance between all involved parties and transactions that might occur on the network. Nodes in the Sovrin network are designed to be either validator nodes expressly to accept write transactions or observer nodes to process read requests. Sovrin trust framework ensures Spreading trust among validator and observer nodes that are mainly operated by the Sovrin stewards "organizations" that decided to use the Sovrin framework as the underlying network to their systems. Accordingly, there is an additional layer for human governance regarding the trust, which fundamentally depends on both people and code. For each relationship the identity owner creates, he can use a different DID; even though that DID of a specific relationship has been compromised, he could create another one without any impact on this relationship. Sovrin supports privacy by design through employing some cryptographic techniques to ensure that interactions occur securely and minimizing the unwanted correlations between data and identifiers by using pairwise-pseudonymous identifiers "DID". Besides, Zero-knowledge proof is typically used for the selective disclosure purpose of personal data and supports VC. According to the Sovrin technical report, All the VCs will be stored off-ledger by the identity owner, as they mentioned that " no private information is ever stored on the ledger, in any form" [54].

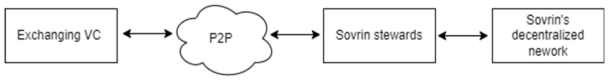

Fig 18. Sovrin layers

Figure 18 shows that the Sovrin layers from the initial layer “exchange VCs” that the users can receive and send the credentials stored in their digital wallet. Developers can develop at the P2P layer a software that fundamentally relies on the Sovrin network to provide such decentralized solutions for managing the identity. Sovrin’s deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Although the users can perform their identity operations by themselves and give the needed consent over it, it is a partial SSIM solution since they cannot give a permission to another user without referring to Sovrin stewards to give the needed trust.
- **Blockchain application:** Hyperledger Indy public blockchain
- **Identity Proofing approach**: ZKP and PKI
- **Support Identity Core Operations:** The users can create as much as they want of identity along with the possibility update it, but the obstacle here is that the deletion of an old ID when the user establishes new communication with a certain party might be difficult due to the nature of blockchain used, and this corresponds to the study [44] that it mentioned it before.
- **offChain/onChain Storage:** offChain storage.
- **Full/Partial Decentralization:** Full decentralization, there's no data stored outside the user's control and no cloud storage or central server might control the identity management process.
- **Portability:** Even though this solution supports using open standards, e.g. DID & VC, to make it portable, the main architecture limits the portability since the Sovrin stewards have the ability to decide whether the transactions might be trusted or not to make the validator & observer nodes handle with.
- **Cost**: It is open source project and cost free.

5.3.2 Uport

Uport [57] is an open source framework built on a public permissionless blockchain Ethereum to enable building decentralized applications to manage the identity and utilize a smart contract to observe and implement the agreement concerning a digital representation of a certain entity. A smart contract is a piece of code stored on the Ethereum blockchain with a cryptographic identifier. It can be deployed by any user aiming to validate the assertions during interacting with another contract [58]. In uPort, this cryptographic identifier is mentioned as uPort identifier, which is globally unique and handled as the address of Ethereum smart contracts. uPort architecturally relies on four components to handle the user transactions [58]: Controller contract, Proxy contract, Recovery Quorum contract and application contract. Figure 19 shows the architecture overview of the uPort system. The controller contract aims to maintain the access control features over the proxy contract at receiving a transaction; the users can then authenticate themselves using their cryptographic private key. The fundamental advantage of this solution is that it resolves the unability of recovering a lost private key by allowing so-called "recovery delegates" in the recovery quorum contract. Recovery delegates can present a new user's address instead of that lost to the controller contract. This process occurs with the needed confirmation provided by the user so that he can add/remove recovery delegates. Users can inform the recovery delegates about their new address; these delegates can be closer friends or family members, which must be defined in the controller contract. Once the recovery delegates confirm the new address and provide it to the controller contract, the new user address will be updated at the controller contract. Afterwards, if the users want to interact with another application smart contract, they can send the transactions to the proxy contract after obtaining the right from the controller contract. A proxy contract forwards these transactions to the target smart contract on the Ethereum blockchain.

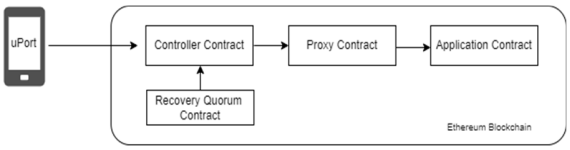

Fig 19. uPort Components

However, there are some major obstacles in the design of uPort regarding privacy and useability [58]. The current design enables the recovery delegates of the user to be publicly available on the blockchain, which could lead to having a concern about if an attacker compromises the user's delegates identities, he can then affect directly on the user's identity that will be disclosed to him. Besides that, the uPort mobile app only holds one identity for the user. In case he cannot access it, for example, by getting a new phone, he can restore it by connecting with the recovery delegates. uPort's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Fully SSIM solution since the users have full control over their identity operations, and they can grant other users the needed permissions by themselves to help for recovering the lost identity.
- **Blockchain application:** Ethereum public permissionless blockchain
- **Identity Proofing approach**: Since this solution relies on a public platform where anyone can read the data stored on it, uPort uses PKI to limit the data disclosure and also uses the Diffie-Hellman method to securely exchange the data.
- **Support Identity Core Operations:** The users can create and update their identities through the user's device; thereby, all the data linked to their identity will be held locally. The deletion process can also be done easily for the attributes the user has, but for the immutability nature of blockchain, deleting the old ID when the user updates his address is difficult [44].
- **offChain/onChain Storage:** offChain storage.
- **Full/Partial Decentralization** No central server stores the data or third parties between the user and the system. A trust can be selected by the users themselves concerning pick the delegates and they are handled with a full decentralized system.
- **Portability:** Not supported, causing it is not feasible to other uPort users to become delegates besides that it is not allowed to other uPort identities to attest.
- **Cost**: It supports DID, VCs and it is an open source project, but there are cost fees on the Ethereum transactions.

5.3.3 EverID

EverID is another decentralized identity management platform built on the Ethereum permissioned blockchain to achieve independent identity ownership, secure value transfer and third-party integrations [59]. The fundamental goal was for the users to have a complete control over their identities and secure access to the identitie's associated data and never be transferred between members of the network without the user's consent. EverID uses various authentication mechanisms to protect the identity, such as biometrics, password, and Personal Identification Number (PIN), along with using PKI to secure exchanging the data over the blockchain. However, the EverID platform, architecturally, is divided into six software components:

EverID Datagram, EverID Decentralized Application (DApps) [60], EverID Application Programming Interface (API), Ethereum Private Blockchain, EverID Core Smart-contracts, and EverID Supernodes [59]. EverID Datagram is a storage array consisting of the information associated with a user's identity to be stored at the user's mobile device with a backup copy in EverID supernodes. All the data stored in EverID Datagram is locked by the user's biometric ID and restricted by an additional layer of authentication mechanisms. Biometric ID ensures that the user cannot create more than one identity account while using this platform. Once a user needs to delete his EverID, some data will still be valid such as the anonymous identifier of the user's biometric. Therefore, EverID DApp or agent DApp is an EverID-enabled application that can create EverID Datagram for the users, which leads to managing, controlling, and storing their EverID data efficiently. More precisely, EverID DApp is an application installed on the user's device that enables the EverID solution. EverID agent DApp is an agent with a device running EverID agent DApp for the users that do not have their own technology to register into EverID. This might be a feature since the users can access their data without the need to have their own device.

EverID API plays a fundamental role in facilitating integration with other applications that it doesn't rely on EverID. Besides that, the EverID API is mainly secured by HMAC (hash-based message authentication code). A smart-contract framework of Ethereum blockchain is divided for the EverID into five fundamental parts which facilitate handling of all transactions performed on the network and those are namely: EverID Creation and Management, Validation, Transaction, Remote Management, and Organizational EverID. Those smart contracts allow the users to have complete ownership of their data whereby they control each transaction recorded on the blockchain whom this data will be shared with (through their public key), the length of availability, copy of the information shared itself, user's public key, biometric sample, and defined authentication mechanisms. Since the EverID cannot support selective disclosure, users must disclose the full actual data to verify. EverID Supernodes are the host of the blockchain used for coordination and bootstrapping of the EverID Identity Network. EverID's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** The user's data cannot be transferred or accessed without the consent of the user and, therefore, users can perform all identity core operations by themselves, and they grant other parties the needed permission to access it whenever they want. However, a user can access to his data through EverID agent DApp which provides a remote access to data and actually that user does not have a control over the place of stored data so it is not fully SSIM solution.
- **Blockchain application:** Private Ethereum blockchain
- **Identity Proofing approach**: Although It uses a private network of Ethereum, it relies on a method of biometrics verification and uses the PKI method to prove the validity of identity claims. However, selective disclosure is not supported since the users must disclose the actual data when they decide to share it with any party to verify.
- **Support Identity Core Operations:** The users can create, update, and delete the identity claims easily. However, they cannot do that for the biometric identifier since it will be created during the registration process on EverID platform. Then users cannot make it updated or deleted. Also, the immutability nature of blockchain doesn't ensure that completely deletion occur for all records, especially regarding old IDs. [44]
- **offChain/onChain Storage:** The user's data can be stored in both offChain and onChain.
- **Full/Partial Decentralization** Since the users have the ability to keep their data on the cloud or so-called "EverID agent DApp" there is a kind of centralization implicitly whereby the data can be stored in such a central server. And then, it has a partial decentralization.
- **Portability:** Not supported, since EverID customized the user's activities inside its platform or with the applications that already integrated with EverID platform.
- **Cost**: It is not an open source project; there is a cost for using this platform for managing the identity along with transaction cost fees on the network.

5.3.4 LifeID

LifeID [61] is an open source identity management platform built on a decentralized smart contract permissionless blockchain that has been released to satisfy SSIM requirements. This platform aims to empower the users to have an independent identity managed without the need to rely on any other parties to do that on behalf of the users themselves. Besides that, sharing some or all identity data cannot occur without obtaining the needed consent from its owner. LifeID offers a software development kit (SDK) and smartphone application to facilitate such portability throughout the partner's platforms [62]. LifeID application contains a friendly user interface to manage the identity and a wallet to keep users' credentials stored offchain. Besides that, SDK is considered an open source

tool that LifeID provides to the developers or any other parties for the possibility of integration with LifeID or creating such identity solutions that rely on the LifeID software layer. Information about the user's identity can be resealed in the form of a verifiable credential VC which any party requires. LifeID utilizes the PKI and ZKP methods to minimize the data disclosed. Once the party needs to verify the user's credentials, he will ask him to provide proof confirmed by a trusted entity so that the party can verify those credentials without exposing the actual values of it throughout the blockchain.

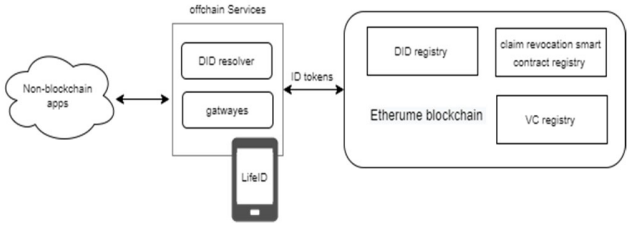

Fig 20. LifeID Architecture

Figure 20 shows the LifeID architecture, which fundamentally presents two main offchain services capable of LifeID application: DID resolver and gateways [62]. Since LifeID supports DID [12] and VC [13] standards, offChain service of DID resolver facilitates the interaction between LifeID application and the blockchain environment. Once the user creates a new transaction on the network, it will establish a decentralized identifier specific for that transaction, which will be recorded in a DID document containing the needed information to verify that user's identity. DID resolver handled as a pointer to lookup a DID and retrieve its corresponding DID document. On the other hand, the gateways are software that bridge existing identity protocols such as OpenID connect to interact with the LifeID platform. Besides DID documents recorded on the blockchain, VC can also be recorded on the blockchain and claims revocation smart contract registry to check whether or not the authenticity of a certain claim is still valid. However, LifeID uses an "ID token" to store and exchange the data that offChain services support.

LifeID offers three different identity recovery options: self backup, backup using a trusted group of family or friends, and backup using a trusted organization. For the self backup, LifeID uses a 12 or 24 word seed to build a wallet that the user can utilize to recover his identity. Using a trusted group of friends to recover the identity can be performed by a predefined list of trusted members with which the user can connect through the LifeID application. Also, a trusted organization can be used to recover the identity in which it must be included in a predefined list created by the user [62]. Using such a trusted group above might lead to major concerns regarding the user's privacy since the other parties selected might be using that copy of data for their own purposes. The guarantee for that is only the user's trust in these parties. LifeID's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Fully SSIM solution since the users have full control over their identity operations, and they can grant other users or organizations the needed permissions by themselves to help for recovering the lost identities.
- **Blockchain application:** Public Ethereum blockchain
- **Identity Proofing approach**: LifeID uses biometric verification capable by the phone besides that it uses PKI and ZKP to minimize the data disclosed in case of exchanging it between the concerned parties.
- **Support Identity Core Operations:** All the identity core operations seem to be supported, users can create their identities on LifeID, and they also have the options of updating and deleting whatever they need. The core issue here, as in the previous solutions, is completely deleting the updated information recorded on the public blockchain [44].
- **offChain/onChain Storage:** All the user's data is stored offChain.
- **Full/Partial Decentralization** It is a full decentralization solution. No data is stored away from the user's control, so all these are stored privately in the user's wallet. Also, it relied on a decentralized public blockchain to interact with this platform.
- **Portability:** Supported , since that LifeID offers some offchain services that fundamentally facilitate interacting with LifeID platform through by external platforms built on OpenID connect.
- **Cost**: There is no cost for using this platform, but since it utilizes ethereum blockchain there is a transaction cost fee.

5.3.5 Sora

Sora [63] is a platform that facilitates managing the identity in a decentralized permissioned blockchain. It is built on Hyperledger Iroha [64], and only the preselected participations can interact and access the system. This platform utilizes the W3C standards [12] and [13] so that every Sora user has a unique decentralized identifier DID recorded on the blockchain with its corresponding DID document. Those are managed through a DID resolver which allows the users to specify their cryptographic public key in a DID document and then uses a private keypair to create transactions for Iroha [65]. In Sora, a user can create

multiple identities with different DIDs for each to avoid correlating the identity attributes of that user. However, user's data can be stored in both offChain and onChain. Private data and every keypair can be stored on a remote central server in an encrypted format so that the user has to select an 8-digit password that must contain capital letter(s), small letter(s), and digits for encryption [65]. Since this password is a master key for decrypting user's encrypted data, it might be vulnerable to well-known dictionary attacks. On the other hand, Sora application provides a simple user interface that allows the users to enter their data and upload it to the blockchain in case the data is public; otherwise, it can be stored on a remote server instead of storing it locally in so-called "user wallet". This implicitly does not empower the users to have the needed control over their private data to be stored locally through their devices.

A user can obtain VC from any concerned party "Issuer", which often contains one or more claims signed by him. Once the user creates a transaction and broadcasts it to the Iroha blockchain [65], it includes salted hashes of the claims themselves, a digital signature, and information about the issuer. A user can then pick any part of his identity to be shared with the Verifier, who is the party that the user wishes to access its services. Afterwards, a Verifier can lookup into the blockchain in order to verify the data that the user shared with it so that the corresponding public data stored on the blockchain will be available. Sora's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Partially SSIM solution, since the users haven't control over the data stored remotely in such a central server so that it implicitly performs this action on behalf of the users themselves, although they can access it by using the password that they set before.
- **Blockchain application:** Hyperledger Iroha public blockchain.
- **Identity Proofing approach**: Sora uses PKI and ZKP which capabled by DID and VC standards [12][13].
- **Support Identity Core Operations:** Users in Sora platform have the ability to create as many as they want identities with a unique DID for each identity created. Besides that, they can update it easily with the feature capable of creating multiple identities. Completely deleting the data from the blockchain as previous solutions seems to be hard since the revoked data and such an old ID will still be recorded [44].
- **offChain/onChain Storage:** Both offChain and onChain.
- **Full/Partial Decentralization:** Partial decentralization, since the private data of the user and keypairs are stored in a central server and, therefore, cannot become a completely decentralized solution.
- **Portability:** Not Supported.
- **Cost**: It is an open source project, there is no cost for using this platform.

5.3.6 SelfKey

Selfkey is an open source platform for managing the identity in a decentralization manner whereby it is proposed to meet SSIM principles and empower the users to have complete ownership and control over their identities [66]. Selfkey runs on Ethereum permissionless blockchain, and the users can store identity associated data locally in the so-called "Selfkey wallet application" installed on their devices. However, Selfkey splits into three main components: Selfkey identity wallet, Selfkey marketplace, and the tokens [67]. In the Selfkey marketplace, the users can access various Selfkey products and services provided by the relying parties that use this system. The tokens assist in enabling the trust and exchange of the data between the identity owners and the relying parties in the Selfkey marketplace. Besides, the users adhere to prove their ownership of a token for the transactions with Selfkey services and products by using a trusted verification. A security and privacy transaction in Selfkey is designed to be compatible with W3C standards [12][13], whereby the users can utilize DID and VC capabilities for their transactions with the network. Accordingly, the Selfkey foundation enables the users to disclose a little amount of data to any entity that needs to verify the truth of the user's claims, which is accomplished by ZKP technology [67].

Selfkey identity authentication occurs through independent algorithms such as censorship resistant and force-resilient. Those algorithms are run in a decentralized manner in order to preserve the user's privacy and make tracking the user's activities hard. Since the storage in Selfkey is designed to be offChain, and the users have full control over that, the Selfkey foundation tackles the issue of losing the user's private key through the uPort key recovery mechanism [58]. This mechanism allows the user to delegate one or more trusted users to recover the lost keys. All the drawbacks of using this mechanism will be inherited for Selfkey, specifically concerning misusing the information stored by those trusted members. Furthermore, if they become a target for a certain attack, the user's Selfkey identity will be disclosed. Selfkey's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Fully SSIM solution, since the users have all the needed control

over their identities, the storage mechanism, and the transactions that they are creating. Besides that, users themselves can decide which data is allowed to be shared with other parties as well as grant permissions to other members to recover the key if it is lost.

- **Blockchain application:** Ethereum public blockchain.
- **Identity Proofing approach**: Selfkey uses PKI capable of DID and VC standards [12][13], and ZKP to minimize the data disclosure between the user and the entities that need to verify the claims introduced by him.
- **Support Identity Core Operations:** All the Identity core operations in this solution are supported except the deletion. Although the user has the ability to update his claims as well as the issuer can revoke the claims issued by him, the immutability nature of blockchain records all of that in the logs list [44].
- **offChain/onChain Storage:** This solution empowers the users to store their data and the tokens offChain.
- **Full/Partial Decentralization** It is a full decentralization, since all the user's interactions occur in the decentralized framework enabled by the public blockchain capabilities.
- **Portability:** Not Supported because of that, the system relies on its owned marketplace, allowing users to only interact with partners that utilize Selfkey products and services. Although they mentioned that interoperability and portability with other identity systems are possible [67].
- **Cost**: It is an open source project, but there is a transaction cost fee.

5.3.7 Civic

Civic [68] is an Ethereum-based decentralized identity management solution that aims to support the users' complete control and ownership over their identities. It is designed to facilitate low-cost access to identity verification services on blockchain. Civic token (CVC) plays a fundamental role in transacting those services, exchanging the data between Civic users and the blockchain, and accessing Civic products and services. Architecturally, Civic relies on its own application, the "Civic secure identity app", to be installed on the user's device whereby Civic users can create, verify, and store their identity through using biometrics feature capabled by mobile devices such as fingerprint ID [69]. The Civic application empowers the users to share and manage their claims and record the relevant attestations on the blockchain to allow any Civic identity requester to verify the claim's authenticity. In the Civic application, the identity partner seems to be the identity authenticator, that the user needs to prove his identity for him in order to provide the needed claims and store the user's authentication on the blockchain. However, claims requests will be shown as QR codes whereby the user can scan those codes and review the requested claims, which allows the user to accept or deny sharing those claims with a specific identity requester. Figure 21 shows the Civic architecture and the prominent roles of each party [69].

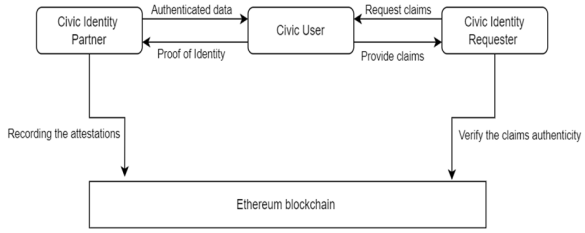

Fig 21. Civic Architecture

Besides the ability of Civic identity partner to record the identity data attestations, he can issue a new specific transaction to revoke the recorded data and inform any identity requester that those data are not valid through using the revocation registry on the blockchain. The recorded data onChain includes the public identifier, the hashed of claims, and the validity of using the claims. Other than that, data will be stored locally on the user's device, such as all the private data, verified claims, and cryptographic keys. The users have the opportunity to back up their data on the cloud through using a personal account or distributed storage platform [69]. This could lead to some security concerns about whether these data might be a target for a certain attack or be lost due to being stored in such a centralized server. Civic's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** the users can manage their identities by themselves and give the needed consent over sharing those data once a party requests that and no access is granted without obtaining the user's permissions. However, a user's data can be stored in cloud storage to provide a copy of data whenever a user needs to recover it, so it is not a fully SSIM solution since a user does not have a control over the storage place.
- **Blockchain application:** Ethereum public permissionless blockchain.
- **Identity Proofing approach**: Civic uses the cryptographic keys PKI and the biometrics verification method for the identity proofing.
- **Support Identity Core Operations:** The users can create their identities easily after registering

and verifying them by Civic. Civic utilizes the biometrics verification method capabled by the mobile device to recognize the user's identity. The users can also update the identity claims as per the concerned parties' needs. However, the deletion process is possible by adding a new transaction specific for the desired data part but cannot be fully deleted due to the previous transaction containing those data be still valid [44].

- **offChain/onChain Storage:** Despite the Civic technical report mentioning that all the actual private data will be stored offChain [69], this data also will be stored onChain in encrypted format.
- **Full/Partial Decentralization** It is a partial decentralization, since the private data can be backed up into the cloud in such a personal account.
- **Portability:** Not Supported because of that, the solution relies on its own products and services provided more specific for the Civic platform; besides, validating the Civic user's identity needs to be verified at a certain Civic identity partner so that it is not allowed for the external partners to perform that.
- **Cost**: As any Ethereum based identity platform, there is an inherited cost for all the transactions using this network. Moreover, this solution is not yet open source.

5.3.8 Identity.com

Identity.com [70] is an open source project that aims to give individuals secure access to the decentralized identity capabled by Ethereum permissionless blockchain and for "on-demand, secure identity verification". Due to the increasing Civic participants and network growth, this platform is designed to support the Civic identity system so that "Civic will stop being the only player in the ecosystem". However, identity.com utilizes CVC tokens; thereby, it is divided into three main components: smart contracts, open source libraries and applications. Identity.com recently joined the W3C consortium as a member, which resulted in supporting two primary standards DID and VC [12][13], that fundamentally empower the SSIM model. Also, the Identity.com Gateway Protocol empowers any decentralized identity applications by adding a permissioning layer which relies on the prespecified rules [70].

Once a user requests to attest his credentials with a particular validator and thus attestations will be created for those credentials; smart contracts ensure that exchanging CVC tokens occur correctly in a way that preserves the integrity of data; besides that, a validator can enforce his rules regarding how should his attestation be used, or is it still valid or revoked. For each attestation created at the validator and utilized by the identity requester, the CVC token will be changed at the validator. A user can upload the authenticated data on his device unless its corresponding attestations are revoked on the blockchain [70]. Due to reliance Identity.com on [12] and [13], it mainly supports data minimum disclosed during interaction with the relying parties, e.g. Zero Knowledge Proof. Identity.com's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Fully SSIM solution; no data about the user will be shared until he grants explicit permission over that. Besides, the users have a complete control regarding how their data will be used, stored, and by whom. The users can also perform all the identity core operations by themselves.
- **Blockchain application:** Ethereum permissionless blockchain.
- **Identity Proofing approach**: Identity.com uses the cryptographic keys PKI, ZKP capabaled by DID and VC standards and biometrics verification.
- **Support Identity Core Operations:** All the identity core operations are supported except the deletion [44]. The user can create and register his identity and then share it with the validator to verify and issue the needed attestations for it. Once the user needs to update a piece or all of his data, he can perform that by sending it to the validator again for authentication of the updated data. The validator can then update the attestation as per the authenticated data. For the deletion process, it is hard to achieve a complete deletion for the recorded data on the blockchain due to the immutable feature of this environment.
- **offChain/onChain Storage:** offChain storage. DID and VC enables the user's private data to be stored in their Credentials Wallets; no data will be stored outside the user's control.
- **Full/Partial Decentralization** It is full decentralization; no data will be stored in a central server as well as it is not allowed to store it outside the user control.
- **Portability:** Creating multiple identities for a single user and interacting with an external partner is not supported; identity.com relies on its own products and services marketplace
- **Cost**: It is an open source project, no cost for using this platform except the cost of adding the transaction on the Ethereum.

5.3.9 Blockstack

Blockstack [71] is an open source decentralized computing platform built on permissionless stacks blockchain, which are mainly an enhanced version of bitcoin to be compatible with smart contracts functionality and building decentralized applications. Proof of Transfer (PoT) has been developed as a novel consensus algorithm to connect the bitcoin and stacks blockchains and facilitate creating the transactions between them [72]. However, blockstack aims to put the users in control over their identities and where they can be stored. Besides, it is designed to support obtaining the user consent for every transaction that needs to share his identity or even store it somewhere. Although this platform implicitly mentioned that the users have to manage their identities by themselves without the intervention of any central authority to own a self-sovereign identity, it does not support W3C open source standards [12][13] enabling the SSIM model.

A user can create as many as he wants of the identities. Each identity created will be associated with a unique identifier (e.g. Blockstack ID) and secret key enabled by the PKI mechanism. A user can interact with any application built on this platform, thereby utilizing his identity to log in. Blockstack developed a Stacks token to assist the users to exchange their data and execute smart contracts for that. This platform has three fundamental layers: blockchain, peer network, and decentralized storage [73]. For the kind of blockchain used [72], it is implemented by a so-called virtualchains logical layer which facilitates the binding process of the digital property (e.g. domain names) to public keys and then allows any nodes to independently verify all the data bindings for a certain public key. Further, it has improved the processing time of transactions by introducing the concept of microblocks that fundamentally give initial confirmation in seconds. Those two points were handled as features to resolve the performance and scalability obstacles in bitcoin by packaging multiple virtualchain transactions into a single blockchain transaction and getting the initial response for the created transaction in seconds. A peer network facilitates discovering the resources by utilizing so-called zone files that act as a pointer to the storage locations in a decentralized manner. Since the zone files store the routing information leading to the storage layer, blockstack designed the peer network layer to preserve the integrity of the recorded data that is already linked to its hash in the blockchain layer. The actual data will be stored in the storage layer with the needed signature by the owner key defined in the blockchain layer before. The storage layer acts as decentralized storage that uses a Gaia system to benefit the capabilities of cloud service providers to store the data in such a way that user-controlled private data lockers and the stored data will be encrypted and signed by him [73]. Users can either store their data locally or use decentralized storage hosting by cloud storage providers.

Blockstack designed its blockchain Name System to provide a naming service that binds such a human-readable name to the user blockstack ID and makes it globally visible, which could lead to some privacy concerns about the user's data [73]. Blockstack does not support key recovery methods in case the user needs to restore the data that he lost for any reason. Blockstack's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Although user consent is needed whereby the users can pick the data allowed to be shared and by whom, and a user decides whether his data can be stored locally or by using a cloud service provider, the design of the blockchain name system allows anyone on the network to read some data about that user, and therefore he cannot control that; it is available for all. So, it is not a fully SSIM solution.
- **Blockchain application:** Permissionless stacks blockchain
- **Identity Proofing approach**: The users utilize the PKI capabilities to proof their ownership of data.
- **Support Identity Core Operations:** The users can create their identities by themselves, and they can update and revoke them easily. A complete deletion from the blockchain cannot happen once a user performs a transaction for that. This is due to the nature of permissionless blockchains that preserve some data that might still be valid on the network.
- **offChain/onChain Storage:** Even though all the actual data values will be stored offChain, some data will also be stored onChain in encrypted format e.g. data hashes.
- **Full/Partial Decentralization** It is partial decentralization due to implicitly utilizing a remote server for storage even though the way of storing the data is controlled by the user in such private data lockers, but still have an inherited risk about using the central cloud server, such as the possibility of a single point of failure which could directly effect on data stored.
- **Portability:** Not supported. Since the current design of blockstack is not allowed to be compatible with other platforms to extend the user experience thereby, the user adheres to utilize one of the dApps built on the blockstack platform even though they can create multiple identities on it.
- **Cost**: Although it is an open source project, there is a transaction cost fee.
-

5.3.10 ShoCard

ShoCard (now PingID) [74], is an identity management platform that supports the utilization of multiple blockchain applications at the same time and changing to new ones if needed. This fundamentally benefits extending the user experience over those blockchains and forms a solid base that not only depends on one kind of blockchain to be used; it is fixable regarding the underlying technology. On the other hand, this main feature could directly affect the computational power of processing the transactions. ShoCard whitepaper [75] mentioned that the expected processing time could take around 30 minutes to verify five million records on the blockchain. However, ShoCard aims to give the users complete control over their identities, where they can be securely stored, and who they can be shared with. Nevertheless, the ShoCard infrastructure does not provide a proper environment to be compatible with W3C standards [12][13], whereby the exchange of data will be done by including the actual value of it; hence, it does not support minimal disclosure for data. A user utilizes the ShoCard application to interact with the platform by scanning the QR codes for all sending and responding actions.

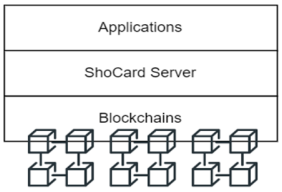

Fig 22. ShoCard Architecture

Figure 22 shows ShoCard architecture generally that contains three main components [75]: blockchain, ShoCard server, and Applications. In the applications, a user has to provide real identity credentials such as a picture of himself and the passport to get an asymmetric key pair from the ShoCard application and thus get ShoCardID. Those credentials would be securely stored locally in a user device while the hash of this data will be recorded in the blockchain for verification purposes. This process prevents the user from creating more than one identity while utilizing the ShoCard platform. Next, the ShoCard server acts as a secure communication pipeline to facilitate exchanging the data between various parties. Specifically, it handles the user information certified and signed by the identity provider via a digital secured envelope. ShoCard server, after storing the enveloped data will create EnvelopeID for it to make it available for access if needed. Once a user needs to verify his identity to another party, he will generate a QR code specific to that EnvelopeID and include his public key to send to that party. A party, in turn, will validate the provided information from the user by scanning the QR code and downloading the enveloped data from the ShoCard server by EnvelopeID included in that QR code. Afterwards, comparing the data he got from the server by the data recorded in blockchain to grant the access to that user.

Since the ShoCard is designed to allow the user data to be stored in the ShoCard server in an encrypted format and it cannot extract any information from it, it is implicitly not a fully decentralized identity platform. Besides, it raises the risk of service interruption since all the exchanged data between the parties cannot be done without referencing the ShoCard server, which would be useless if it was down. Further, ShoCard does not support any key recovery methods, and hence there is no possible way to recover the lost user's identity, and in this case, he can get a new asymmetric key pair with a new ShoCardID after re-registering himself again. ShoCard's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Although the user has control over his data by himself and grants the needed consent to share it with other parties, he cannot control the data stored in ShoCard server since this seems to be mandatory to make the user authentication be done on this platform. So it is a partial SSIM solution.
- **Blockchain application:** Multiple types of blockchains.
- **Identity Proofing approach**: ShoCard utilizes the biometric verification method capabled by the mobile phone, such as a selfie picture, along with the PKI mechanism to allow the users to provide the needed proof of the ownership of their identities.
- **Support Identity Core Operations:** A user can create his identity easily by introducing such real identity credentials to get ShoCardID. Updating his identity seems to be hard since those credentials are real and associated with asymmetric keys generated by ShoCard as well as ShoCardID specific to his identity. In case he cannot access his data, he can reregister himself again, but all his data associated with the old identity would be unavailable. Deletion, as in all solutions based on blockchain, cannot occur completely [44].
- **offChain/onChain Storage:** Both since the users can store their private data offChain by using their devices and, at the same time, it can be available onChain in an encrypted format.
- **Full/Partial Decentralization** Partial decentralization. Since it depends on the ShoCard server to validate all the data before exchanging it with others.

- **Portability:** Not supported, since a user identity cannot be used outside the ShoCard platform.
- **Cost**: It is not an open source project and it has a transaction cost fee.

5.3.11 Jolocom

Jolocom [76], is an open source decentralized identity solution built on the public permissionless Ethereum blockchain aiming to empower the users to manage their identities by themselves instead of any entity that could perform that on behalf of the users. Jolocom is designed to be compatible with W3C standards [12][13] and thus facilitates verifying the identity in a minimal amount of data disclosed by embedded metadata that does not carry the actual values of that data while exchanging it between the concerned parties. Further, individual users have been empowered in Jolocom to manage multiple personas by using such hierarchical deterministic (HD) keys [77]. A user utilizes the Jolocom SmartWallet application that serves as a user interface to create and control his identity locally on the smartphone. The HD keys allow the users to generate multiple child keys from the parent key so that the user can attach them to his multiple identities. The individual users significantly govern those keys, which would be generated by a "known seed" that facilitates recovery of those keys by introducing them again if needed. Jolocom, in its design, not just utilizes the HD keys to handle the user identities; it combines those keys with the DIDs that the user owns to avoid usability issues in HD keys [77].

The exchanging of data between a user and other parties occurs in the form of QR codes so that the user can receive and request the VCs by scanning those codes and taking action, either by rejecting or accepting. Since the Jolocom platform is designed to be compatible with all SSIM requirements and [12][13] standards, it will rely on DID and VC for all the interactions between the involved parties, which leads to preserving the user's privacy, thereby, no private data would be recorded on blockchain except the public keys used for authentication and such a public profile that can the user create it if he want. All that information would be stored in the DID document, allowing anyone on the network to verify the authenticity of the provided VC in a cryptographically verifiable method. This method allows a relying party to compare the private keys included in the VC with those recorded publicly on the blockchain without disclosing the actual values of claims.

Jolocom aims to use a different method for key recovery that does not rely on other people or the central server to backup the user's data. A user has to utilize a known phrase seed and securely store it somewhere, whereby he cannot recover the lost data again without introducing the same seed. Jolocom library facilitates extending the user experience to interact with other SSIM solutions that are designed to be compatible with W3C standards [12][13] and the SSIM requirements. Even though Jolocom utilizes the public Ethereum blockchain, the whitepaper [77] mentioned that Jolocom can support in future the utilization of technology agnostic as in ShoCard [74] that, allows using multiple types of blockchains at the same time aiming to ensure that integrity of the user's identity will persist beyond specific network environment. Jolocom's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Full SSIM solution. Since the Jolocom empowered the users to decide whether or not their identities can be "resolvable" or "knowable" or should no longer "exist" within the system [77]; besides, all the activities about their identities should be associated with a given consent by them, so that the users can grant to other people needed permission to do something. They can also perform all the core identity operations by themselves without requiring another party to perform that on their behalf.
- **Blockchain application:** Public permissionless Ethereum blockchain.
- **Identity Proofing approach**: The users can utilize PKI mechanisms to prove their identities' ownership in the form of pairwise keys and utilize zero-knowledge proof enabled by DID and VC standards [12][13].
- **Support Identity Core Operations:** Even though the user can create a Jolocom identity easily with DID, he can update his public key recorded on the blockchain in place of the old one if he wants to update the identity with new data. Also, a user can delete his identity, but as in all SSIM solutions that utilize blockchain, this deletion cannot occur completely; some data would still be valid on the network [44].
- **offChain/onChain Storage:** offChain storage. No private data would be stored away from the users; the users store their private data locally. Also, no private data would be stored on the blockchain, even if it was in an encrypted format.
- **Full/Partial Decentralization:** Full decentralization solution, all the interactions between the parties occur in a decentralized manner; there is no central authority or even central servers employed in Jolocom to perform specific actions.
- **Portability:** Not supported, since the current design of Jolocom customized the user experience inside its platform or with the partners applications which were already integrated with Jolocom through its SDK.

- **Cost**: It is an open source project. No cost of utilizing this platform except the inherited cost of transaction fees on Ethereum.

5.3.12 Dock

Dock [79], is a decentralized data exchange and identity management platform powered by public permissionless Ethereum blockchain, aiming to remove the obstacles in forward owning the user's complete control over their data and make it portable between the applications in the same context [80]. The Dock is designed to integrate with W3C standards [12][13] and the underlying technology, blockchain, to facilitate the creation of tamper-proof VCs, DIDs, and decentralized applications with the same infrastructure to support the decentralized identity framework. The Dock utility token (DOCK) is fundamental in facilitating data exchange between all the Dock network's participants. Further, it is a crucial base for executing transactions, creating DID, issuing and attaching VC, validation, and all other network functions. Figure 23 clarifies the Dock architecture in a general insight [79]. Since this platform is compatible with W3C standards [12][13], it has developed its infrastructure with three core entities: Issuer, Holder, and Verifier. An Issuer is a certified and trusted entity that grants such verifiable information VCs to a specific entity; thereby, other concerned parties can prove the authenticity of those VCs in a cryptographical manner. A Holder is the party that receives the credential from other certified entities, Issuers, and keeps all the VCs that he obtains in his digital wallet; Dock wallet application, to share it later if needed. A verifier is a party that needs to verify the authenticity of the provided claims by looking into the blockchain and comparing the signature included in VC with that stored publicly on the blockchain. Also, he can verify if the VCs are still valid or have been revoked by their Issuer. However, all the VCs can be accessed and verified by using QR codes. These codes can be created in the Dock certs platform. All the participants on the Dock can utilize this platform to manage, create, and present the VCs by using their own accounts on Dock certs. They have to create their own DID before dealing with VCs. DID would be associated with a unique cryptographic keypair whereby a party can prove his ownership by signing any VC he issued, if he was an Issuer, or he obtained from other parties, if he was a Holder. Afterwards, Dock certs will publish all the transactions into the blockchain.

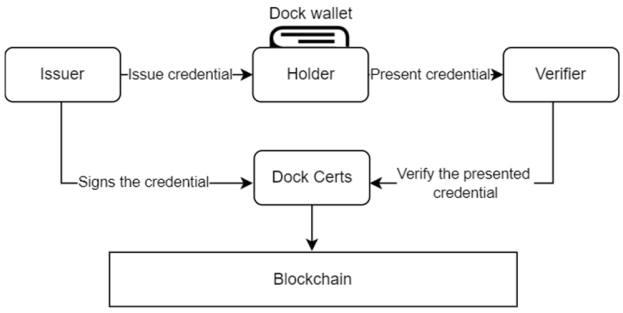

Fig 23. Dock Architecture

Dock provides an SDK toolkit to enable businesses and developers to incorporate the underlying technology and the Dock infrastructure to create decentralized applications that it can interact with Dock. This benefits the user by making his identity portable across all those applications powered by the Dock network. On the other hand, Dock empowers the users to keep their data stored offChain in their wallet; thereby, no data would be stored onChain except the data that the user wants to be publicly available such as the public profile. This shifts the responsibility to the user regarding which data he decides to share with the public and, in case wrongly configured the client to publish sensitive data, it will be available for all, and thus, he cannot retrieve it again to keep it secret. Dock provides an option to the users to store their private data in the Dock cloud server to facilitate restoring a backup of those data if needed. Besides the significant feature this method provides to the users, it has some security concerns. If that server were down for any reason, the data stored there would also be unavailable. Dock's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** A user has the right to create his identity, update, and delete it by himself without relying on other parties. Also, he can selectively grant a specific party a permission to read some data if needed. However, some of the user's data can be stored in such a cloud server and thus it is a partial SSIM solution since the data would be stored at a third party.
- **Blockchain application:** Public permissionless Ethereum blockchain.
- **Identity Proofing approach**: The user utilizes the PKI mechanism to proof the authenticity of the credential provided by him as well as utilizes the zero knowledge proof during exchanging the data that can be verifiable in a cryptographical manner.
- **Support Identity Core Operations:** A user can introduce himself to the platform by creating his identity that can be connected with a certain DID. he can also update the information associated with

his identity and delete it if needed. There is an obstacle in forward achieving complete deletion on the network, represented in the deleted part, let's say DID, would still be accessible and valid on the network even though the issued VCs by using that DID would be revoked once it was deleted. Keeping DID alive means the DID document is also alive, which may contain the public profile of the party that he owns that DID [44].

- **offChain/onChain Storage:** All the private data would be stored offChain in the Dock wallet app. No private data will be shared publicly on blockchain.
- **Full/Partial Decentralization:** Partial decentralization. Dock utilizes such a central server, cloud storage, to backup the user's private data. So it is not a completely decentralized identity solution.
- **Portability:** Not supported, since the user's identity can only be portable with all the applications powered by the Dock SDK toolkit. Interaction with other external platforms is not possible.
- **Cost**: It is not an open source project. There is a cost of utilizing this platform as well as transaction cost fees.

5.3.13 Sphere Identity

Sphere Identity [81], is a platform that aims to introduce a blockchain-based distributed storage solution to manage self-sovereign identity for businesses and individuals. The fundamental values of Sphere Identity are security and privacy by design, aiming to enhance the user's and business's experience in handling with secure global identity platform. Sphere Identity promises the users to keep their data safe and have complete control over where those data can be stored and who can be shared with. Besides, businesses can integrate Sphere Identity with their systems and facilitate onboarding customers by utilizing the Sphere Identity sign-up button enabled by the integration. Sphere Identity is divided into three fundamental components: business application, Sphere Identity platform, and personal application [82]. For the business customers, they can be integrated with Sphere Identity after choosing the suitable subscription plan that this platform provides for businesses, whereby they will obtain an API SDK toolkit that allows such a web application to interact with the Sphere Identity platform. Therefore, the users of those business applications can be onboarded through a simple QR code scan that has been enabled by Sphere Identity. They can utilize their personal accounts, previously configured, with all the business partners that decided to integrate with this platform. The personal application is specific to the individual users, aiming to preserve their identity by empowering them to have such a local storage and full rights to grant other parties needed permission to share certain of their data. The Sphere Identity platform handles both business and personal application interactions securely, ensuring the integrity of the shared data and providing identity management services for both.

Sphere Identity does not support any key recovery methods except the randomly generated security keywords that allow the user to keep his data accessible if he lost it for any reason. If the user cannot remember the generated phrase provided by Sphere Identity, he, therefore, creates another account and has to register himself again and note that he will lose access to the associated data kept in his old account [82]. Further, Sphere Identity adheres the users who wish to utilize its personal application to provide approval over the usage policy that mentioned, "I understand that Sphere Identity is collecting personally identifiable information (data) which might include sensitive data" to continue the registration process [81]. This is explicitly inconsistent with the self-sovereign identity that they aim to comply with. Sphere Identity's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Partially SSIM solution. Even though Sphere Identity empowers the users to have control over the identity core operations and the ability to grant other parties such needed consent over their data, Sphere Identity implicitly mentioned that in the usage policy of the personal application, it can collect, terminate the user account, and transfer its rights to another organization if needed without obtaining the user consent.
- **Blockchain application:** Public Ethereum blockchain.
- **Identity Proofing approach**: Sphere Identity relies on a personal identification number PIN, generating 12 random security keywords, "phrase", and biometric features during the registration process. For the data exchange, Sphere Identity uses such a combination of private/public key pairs and a symmetric key, thereby ensuring no access to the transmissioned data except the concerned party whom a user has accepted to share his data with.
- **Support Identity Core Operations:** The user can create his identity on the Sphere personal application by uploading and scanning his documents easily. A user also can update his data whenever he wants but deleting cannot occur completely; some data would still be valid at the

business partners even though the user has deleted from the application [44].

- **offChain/onChain Storage:** Sphere Identity utilizes both onChain and offChain storage. A user can keep his data away from the network by storing it locally in the mobile device, and also some data will be stored onChain in an encrypted format.
- **Full/Partial Decentralization:** Even though no direct central authority or central server controls the user's data, Sphere Identity serves some data in its server, and it can provide usage reporting if needed; partial decentralization network.
- **Portability:** Not supported. Since the user can only interact with the business applications that utilize Sphere Identity; Sphere Identity does not support interaction with the external platforms
- **Cost**: Even though there is no cost of utilizing the personal application, there is a cost of utilization for any business partner who wants to utilize Sphere Identity for its customers, besides a transaction cost fee on the network.

5.3.14 NuID

NuID [83], is a trustless authentication solution that has been designed to introduce a decentralized identity framework and Self-Sovereign Identity for websites and applications. This solution aims to replace the current hash and store model of authentication with the NuID model that employs the Ethereum distributed ledger and zero knowledge cryptography to empower the users to prove themselves to other parties without the need to share, store, or see the authentication data, in a purpose of achieving so-called "trustless authentication". A goal of this authentication was to avoid the risk of breaking the credential data that would be stored in the traditional siloed databases. A NuID ensures that the user can prove the ownership of his own credential in a completely decentralized framework, thereby cannot anybody knows the actual value of the authentication data that a user utilizes, even the NuID itself. However, the NuID authentication service utilizes lightweight client libraries to convert the user's credentials into the zero knowledge proof parameters. The NuID provides independent authentication that does not rely on a separate application to complete the trustless method of authentication, whereby it is just a service that provides a verifiable decentralized identity solution for the business's websites and applications [84]. NuID allows businesses to leverage its authentication service into their systems by incorporating the client libraries (SDK toolkit) with their service and connecting them to the NuID REST API that they can integrate with, allowing the user's public parameters to be sent via the POST request. Figure 24 shows a user's registration and authentication flow on NuID [84].

Fig 24. NuID Registration and Authentication Flow

1. A user can define new credentials, a username and a secret (e.g. password) during the registration process. 2. NuID client libraries installed in the user's frontend would take the secret introduced by the user to generate zero knowledge proof parameters. A username and zero knowledge proof parameters will be sent to the user's backend, which is a relying party server, and the parameters will be forwarded to the NuID API. 3. After defining the username and Zero knowledge proof parameters at the user's backend and NuID API, it would be posted to the ledger as the user's public parameters whereby the authentication flow would be more accessible by looking for those public data. 4. Once the ledger stores the data coming from the registration process, it returns an address (e.g. unique, persistent identifier) that is located to those data. 5. A username is associated with that address where the zero knowledge proof parameters are posted. Once a user inputs his credentials again and clicks "login", 6. A username would be sent to the user's backend. 7. User's backend, in turn, sends the associated address that meets the provided username to the NuID API to request a cryptographic challenge. 8. NuID API retrieves the zero knowledge proof parameters from the ledger, aiming to generate a one-time cryptographic challenge derived from the user's zero knowledge proof parameters to be forwarded to the user's frontend in step 9. NuID client libraries, in step 10, utilizes the secret and the received challenge to generate another one-time cryptographic challenge that would be forwarded to the NuID API for verification purpose in step 11. Finally, NuID API, in turn, returns a success or failure response to the user's backend based on the result of verification, step 12.

NuID solution [83][84] does not mention any key recovery methods once a user needs to recover his lost identity. However, it empowers the users to keep their credentials stored locally on their devices, in their mind, or both. Suppose the user's identity cannot be accessible. In that case, the user adheres to create a new identity with mentioning that the old authentication identity credential would be available for any access attempt in future, as well as its public key on the ledger. NuID's deployment of each criterion is as follows.

- **Fully/Partially SSIM Solution:** Fully SSIM solution, since the user has all the right to manage

his identity by himself thereby, neither the relying party nor the NuID can perform that on behalf of the user or even share identity associated data without obtaining the user's approval.

- **Blockchain application:** Public Ethereum blockchain.
- **Identity Proofing approach**: NuID supports a zero knowledge proof method to verify the ownership of identity and the username and secret the user has to provide once he wants to interact with any relying party. Those credentials can be protected under device-local biometrics.
- **Support Identity Core Operations:** NuID enables users to create and update their identities easily through the SDK libraries integrated with the business applications. Further, deletion of the identity is possible, but the immutability nature of the blockchain may allow the old authentication credential to be available; complete deletion cannot occur [44].
- **offChain/onChain Storage:** NuID architecture clarifies that the user's private data can never be stored or transmitted away from his device except for the data that can be publicly viewed, so NuID utilizes offChain storage for the private data.
- **Full/Partial Decentralization:** Full decentralization, because no central authority controls the user's credentials except the user himself. NuID infrastructure is completely compatible with a decentralized framework that empowers the user to control his credential without intermediaries to perform that on his behalf.
- **Portability:** Not supported. The current infrastructure of NuID does not support interacting with the external identity platforms, so the users cannot utilize their NuID identity globally except for the business applications integrated with the NuID platform.
- **Cost**: It is open source solution, but there is a cost of utilizing the distributed ledger storage, e.g. Ethereum, for any transaction submitted on it.

6. Results and Discussion

6.1 The Main Differences Between the Traditional Identity Management Models and the Self-Sovereign Identity Management Model

During the investigation of common identity management models presented in chapter two, the study notices that there are different goals for each identity model striving to be satisfied with different identity management requirements of a specific period of time. For example, the Silo model is the first identity model among the others. It shows the simplest form of authentication mechanism (e.g. username/password) that the user can use to prove his identity to another party aiming to gain the required permission to access its services. The fundamental approach in the Silo model is to isolate each login activity from the other logins, meaning that for each trust relationship a user establishes with other parties, the credential must be separated, which requires the user to redefine his credential for each. Even though this method can lower the cost at the SP than remembering all the login activities that the user performs, it directly affects the usability of this model from the user's perspective.

On the other hand, a federated identity model or single sign on model restricts the user's activities inside such a single circle of trust. For each single working session, a user can access all the SPs included inside his circle of trust and benefit from their services. This method is enabled by a single sign on feature that improves the user experience in terms of allowing the user to have authorized permission to access multiple SPs after being authenticated with a trusted authority (e.g. IDP). Most federated identity systems that support this feature also support single sign off, allowing users to log out all the accessed SPs simultaneously. A circle of trust typically contains a single IDP and multiple SPs where all must trust the IDP. Cooperation among all concerned parties in a federated identity model is possible, unlike other identity models, which makes the delegation process between them possible. The fundamental role of IDP is shifting the process of managing the credentials from the isolated SPs into a central authority trusted party that all the SPs can refer to for verifying the introduced assertion presented by the user. The Federated identity model enables the IDPs to have the ability to store and share the user's credentials, if needed, with any party involved in its circle of trust, unlike other models that mainly turn around the user to be responsible for who is the party that can his data share with and how can he manage his data to be kept secret. Although in the Silo model, SP can store the user's credentials in the local server, there is no cooperation between it and other parties, so these credentials can be shared with them if needed. In a federated identity model, a user can prove the rightful possession of his identity by one of three identity proofing methods, unlike the Silo model, which relies on the user's predefined username/password setting. Federated identity might be authenticated by utilizing the bearer, holder-of-key, or sender-vouches method, and the study clarified each in section 3.2.1 in further detail. The anonymity in the federated identity model is supported by aiming to make the linkability of the user's identity and tracking his activities difficult and comes in two forms long term or short-term pseudonyms depending on the agreement between the IDP and SPs to facilitate such pointing to the particular user during the communication between them. This feature is not supported

in the Silo model since the user utilizes his explicit credentials that might contain a part of his identity during the communication with another party which facilitate mentioning that user if needed. Despite all the features introduced by the federated identity model, a user can't control his data stored at IDP, considering that it is a central authority responsible for managing the user's identities on their behalf.

Self-sovereign identity model removes the central authority role aiming to enhance the users' privacy by empowering them to manage their identities by themselves. This model is designed to be such a decentralized approach capabaled by the blockchain technology that turns around the user-centric regarding all the activities referring to the user himself, unlike the federated identity and Silo models that are centralized approach which allow another party to have the ability to control the user's data and tracking the user's activities considering that those models have built to be around server-centric. Since the blockchain plays a crucial role in the self-sovereign identity model in establishing indirect communication between the IDP and SP, ensuring that a user's activities cannot be tracked by any party and simultaneously preserving his privacy. A user in SSIM can control all the data that refer to his identity in such a way that ensures that data cannot be fully disclosed during exchanging it with another party and shifting the job of verifying the introduced data to the blockchain. Instead of storing the user's data in such a local central server at IDP as the other models, SSIM supports storing those data locally at the user's device (e.g. offChain) or storing the data across multiple nodes on the blockchain (e.g. onChain). Also, SSIM empowers the users to have independent identities so that any party on the network cannot manage those identities except the users themselves; where they would be responsible for where their data should be stored and who can share it with as well as they have complete control and ownership on that. Trust has taken place as a significant difference between all the identity models considering that a user in the Silo model, for example, needs to have a trust relationship with SP before exchanging the data between them. On the other hand, the federated identity model restricted that inside such a circle of trust, allowing a user to exchange his data with the involved parties inside this circle. SSIM removed this restriction by allowing the user to build a trust relationship with any party he wants to connect with since exchanging and verifying the data between the parties occurs in such a decentralization manner (e.g. blockchain) which benefits in facilitating those activities without revealing their identities to each other.

However, both federated identity and Self Sovereign identity models support utilizing cryptographic mechanisms. For example, in a federated identity model, identity may be verified by utilizing symmetric or asymmetric encryption approaches in the Holder-of-Key identity proofing method. This might be selected in the agreement policy previously configured between IDP and SP, where the SAML assertion introduced by a user contains a value of confirmation method that he performed before obtaining the authorized permission of access to the SP services. On the other hand, Self Sovereign identity model utilizes only the asymmetric encryption approach to prove the rightful possession of identity and the zero-knowledge proofs enabled by W3C standards [12][13]. Section 2.3 contains a comparison that summarizes the overall differences between the identity management models.

6.2 Self-Sovereign Identity Management Solutions and Current Comparison Criteria

The study noticed, in chapter three, that most SSIM solutions shared the main goal, which is focusing on the user's control that must come back to the identity owner himself, aiming to allow him to have such a self-sovereign identity that removes the role of an intermediary central third party. Besides, the study noticed that not all SSIM solutions support the W3C standards [12][13] that have fundamentally enabled the concept of self-sovereign identity since the first time this model was introduced in 2017. Each solution has adapted to the SSIM model in such a consistent method considering that the solution's goals itself and the model's goals. Some of those solutions utilize a public blockchain where anyone on the network is able to read the data which is already recorded on the blockchain, and, on the other hand, some solutions utilize such a consortium kind of blockchain or private blockchain aiming to restrict using the network to only authorized participants. Before comparing SSIM solutions, the study defined comparison criteria in section 5.1 to facilitate extracting most differences from them and pointing out their limitations and possible enhancements later. The study compared fourteen available SSIM solutions in chapter three and highlighted their architectural design and goals. This section aims to discuss the relevant results extracted from SSIM solutions in further detail.

- **Fully/Partially SSIM Solution**

The study classified SSIM solutions into two categories depending on this criterion that aims to shed light on the truth of the user's ability to easily control all the data that refer to his identity without relying on any central authority to perform that on his behalf. The study assesses this criterion based on three perspectives: the ability of the identity owner to create, update and delete his data by himself, if he can grant the needed permission to other people so that they can share, read, or write the data, and if he has control over the place of storing the data where this job should not be shifted to any third party to do it on the user's behalf. The study in table 4.1 shows a classification of SSIM solutions depending on whether or not can classify

it into full or partial SSIM solutions with mentioning the reason if it was a partial solution.

Table 4. SSIM solutions classification

SSIM solution	Fully SSIM solution	Partially SSIM solution	Reason if any
Sovrin		✓	There is a kind of restriction if the user wants to grant permission to another user where he has to refer to Sovrin stewards to give the trust first.
uPort	✓		-
EverID		✓	EverID agent DApp provides remote access to data to its users, which shifts the ownership of the place of storage to a third party. A user can access that place from any device where does not necessarily own that device.
LifeID	✓		-
Sora		✓	A user's data can be stored at a third party's central server so that this party has control regarding the data he owns.
Selfkey	✓		-
Civic		✓	Storage place of the user's data might be available in both a user and cloud server for backed-up purposes.
Identity.com	✓		-
Blockstack		✓	The architectural design of blockstack allows anyone on the network to read some data that refers to a particular user without having permission from that user.
ShoCard		✓	All the exchanged data between the parties must be passed through the ShoCard server, considering that it's a trusted third party, and thus some of the user's data must be stored in this server for verification purposes.
Jolocom	✓		-
Dock		✓	User's private data can be backed up in such cloud storage in case a user needs to recover his lost data. This fundamentally provides control to a third party over those data since it owns the storage place.
Sphere Identity		✓	User's data can be collected by Sphere Identity without obtaining a user permission
NuID	✓		-

- **Blockchain Application**

Since varying blockchain applications powered SSIM solutions and the difference for each kind, especially in transaction cost, processing time, and scalability, reflect directly on the SSIM solution considering that it is a fundamental technology that provides such a decentralization environment for those solutions. The outcome of this criterion shows that all the SSIM solutions utilize either a single blockchain or multiple blockchains. The study in chapter three notices a solution utilizing multiple blockchains, namely ShoCard. The main purpose of multiple blockchains is to add some features of utilizing their environments; for example, ShoCard supports this design aiming to be an independent solution that does not rely on one blockchain to process the transactions; however, multiple blockchains can be used. Besides the features provided by merging blockchains, there is a network scalability issue since the computational power of processing the user's transactions will be longer at the solutions that are designed to resolve the difficulties of communicating among different kinds of blockchains and this fundamentally back to different transaction processing time and transaction cost for each blockchain used.

On the other hand, Blockstack is built on permissionless stacks blockchain, which is fundamentally

considered an enhanced version of bitcoin, and its goal is to be compatible with smart contracts functionality and resolve the bitcoin's scalability issue by adding a virtualchains logical layer.

Further, the study noticed that some SSIM solutions utilize public Hyperledger blockchain such as Sovrin and Sora. This kind of blockchain is built on the Linux foundation, which is open-source for utilization and is cost-free. The main purpose of this blockchain is to be more scalable, cost-effective, and improve the reliability of DApps, considering that it is designed to be compatible with the requirements of decentralized identity and to accommodate the size of participants considering it is already built on an open source environment.

All remaining SSIM solutions utilize the Ethereum blockchain allowing them to build DApps that use the same blockchain environment. Even though there is a cost fee for any published transactions on the network, most current SSIM solutions utilize this kind of blockchain, especially for the public. Only one solution utilizes the private Ethereum blockchain, namely EverID. Although adopting this blockchain environment at most current SSIM solutions, there is a scalability issue that might simultaneously affect the user experience. The main feature is facilitating data portability among the DApps built on this blockchain.

- **Identity Proofing approach**

Some SSIM solutions are designed to be compatible with W3C standards [12][13] since it has been developed to enable such a completely independent identity empowered by DID and VC. Thus, all SSIM solutions that support those standards already provide two fundamental identity-proofing approaches, namely PKI and ZKP. Those solutions are represented in Sovrin, uPort, LifeID, Sora, SelfKey, Identity.com, Jolocom, and Dock. Some of those solutions adopted other approaches along with PKI and ZKP. For example, LifeID and Identity.com utilize biometric verification during the registration process for their users. On the other hand, the study noticed that some other SSIM solutions, such as EverID, Civic, and ShoCard, rely on biometric verification and PKI. In contrast, the Blockstack solution relies on only PKI. Unlike most SSIM solutions, Sphere Identity utilizes Personal Identification Number PIN, biometric verification, and asymmetric/symmetric encryption. NuID solution utilizes username/password, and ZKP identity proofing approaches.

- **Support Identity Core Operations**

Although most SSIM solutions are designed to satisfy the need for identity management independently and support identity core operations, the study noted that all those solutions does not support complete deletion when a user wants to delete something, and that refers to the immutability nature of blockchain that all these solutions rely on. This point is also previously mentioned in [44] as a drawback that allows a piece of data to be alive and available to access from the participants on the same network since any data recorded there is mainly public to read. Otherwise, a user can easily delete any data defined in his wallet or personal application since those data is still not published on the blockchain. Some solutions that utilize biometric verification, such as EverID, LiveID, Civic, Identity.com, ShoCard, Sphere Identity, and NuID does not support creating multiple identities for a single user, and that refers to biometric attributes that must be unique. Since those solutions enable the users to have one identity already linked with biometric attributes, updating the user's identities cannot be supported even though they can update their credentials during exchange data with other parties by making it revoked and filled with new values if needed. An uPort solution utilizes a trusted approach to update the user's identity, where a user can communicate with close people to confirm the updating process. The remaining SSIM solutions support updating both identities and credentials if needed. Creating identity and new credentials are supported in all SSIM solutions as well.

- **offChain/onChain Storage**

Since the underlying technology behind all SSIM solutions and decentralization environments is blockchain, data storage becomes either onChain or offChain storage. The primary purpose of this criterion is to clarify the place of sensitive data or user's private data where it should be stored in those solutions. The study noted that some SSIM solutions store the private data offChain and provide a copy of it onChain in an encrypted format, such as Sphere Identity, EverID, Sora, Civic, Blockstack, and ShoCard. On the other hand, the solutions that store the user's private data offChain and never make it available onChain have various options about the place of storing those data for the user, such as the possibility of storing it locally in his device or storing it in the solution's personal application (e.g. digital wallet) or store it at a third party (e.g. cloud storage). The solutions utilizing offChain storage regarding private data are Sovrin, uPort, LifeID, SelfKey, Identity.com, Jolocom, Dock, and NuID. Both these two methods have features and drawbacks; for example, onChain storage is helpful to public data such as a public profile that a user accepts to be published on the network, and anyone on the same network can read those data or even verify the validity of introduced credentials by comparing the private digital signature of a specific user and Issuer authority included inside those credentials with the corresponds public key recorded on the network. However, making private data available onChain, even if it is in an encrypted format, means making it available to all the participants that use the same network,

and it is going to take away the user's control that these solutions claim to provide it to a user. An offChain storage has the same obstacle if a solution shifts the job of storing data to a third party, and even though private data cannot be available onChain in that solution, however, it shifts the user's control over those data into a specific party, which becomes inconsistent with its main goal represented in a user should have all the right to manage his identity by himself with removing the role of a third party to manage those data on his behalf.

- **Full/Partial Decentralization**

The study under this criterion classified SSIM solutions into full or partial decentralized solutions based on their architecture and design that provide an environment to manage the identity. Some SSIM solutions such as NuID, Jolocom, Identity.com, SelfKey, LifeID, uPort, and Sovrin are designed to be fully decentralized solutions where all the activities of identity management occur without relying on a central authority, cloud storage, and central servers. Those methods would remove the concept of a complete decentralization environment in SSIM solutions, even if they were built on blockchain to achieve that purpose. The study noticed that most SSIM solutions utilize partial decentralization either to provide a backup of the user's data in cloud storage such as Dock, Blockstack, Civic, Sora, and EverID or to be part of the identity management process through its central servers such as Sphere Identity and ShoCard.

- **Portability**

The study aims under this criterion to shed light on the possibility of extending the user experience in different environments by making the user's identity portable and allowing interaction with external parties. Even though most current SSIM solutions aim to be compatible with W3C open standards [12][13], the architecture of those solutions is different for each, and most of them customize the user experience to be included inside its platform. For example, the solutions such as Sovrin, uPort, Sora, Selfkey, Civic, Identity.com, blockstack, Dock, Sphere identity, EverID, ShoCard, Jolocom and NuID have customized the user experience inside their platforms or with the partners who integrated with their platforms and therefore a user cannot utilize his identity outside on different platforms. The study noticed that LifeID is a solution among all SSIM solutions that bridge the older identity technologies with SSIM platforms through its offChain services. This connectivity allows non-blockchain identity platforms that utilize such an OpenID connect protocol to interact with the LifeID platform in a function of reads only, which will look up into a DID document of a specific party on blockchain to take some information if needed. This will be facilitated by gateway software provided by LifeID, which fundamentally assists in extending the user experience on different platforms and allow the identity in LifeID to be accessible and portable throughout blockchain and non-blockchain identity management solutions.

- **Cost**

The study noted that not all SSIM solutions are open source, even though most support two open standards [12] and [13]. For example, all the solutions that utilize Ethereum blockchain have a further cost regarding pushing new transactions on the network, called a transaction fee. The solutions such as uPort, LifeID, SelfKey, Identity.com, Jolocom, and NuID are designed to be open-source projects; however, there is a transaction fee since it utilizes the Ethereum network. Besides, the solutions such as EverID, Civic, Dock, and Sphere Identity are not open-source projects; there is a cost of utilizing such solutions and publishing a transaction on the Ethereum network. On the other hand, Sovrin and Sora are built on a hyperledger network, and both are open-source projects. Blockstack also is open source, but there is a transaction fee for all the transactions published on the stacks network. Since a ShoCard utilizes multiple blockchains, there is a different cost of transactions; besides, it is not an open-source project.

6.3 Self-Sovereign Identity Management Solutions and Solving Current Identity Management Issues

A study mentioned in chapter two some of the identity management issues that exists in the most widely used systems that utilize a federated identity model. Those issues represented the possibility of tracking the user's activities by a central authority, the possibility of losing the data stored at IDP, which is considered as crucial part threatened by a single point of failure, the possibility of collecting and sharing the data of specific user between IDP and SP without obtaining the needed consent by a user over that, customize a user interaction and sharing identity attributes with the concerned parties previously defined inside a circle of trust, the possibility of accessing all the user's logged in website once stole his federated identity credential, a mandatory of existing a trust before selecting the specific party for connectivity, and the last issue was regarding a single point of control where all interactions among the parties should be passed through a central authority causing the high latency and cost as well as its job to control the user's data on their behalf.

The results of this study show that some of the above issues still exist in SSIM solutions which are hard to overcome yet. For example, the main goal of all SSIM solutions was to give a user all the right to manage his identity as he wants and have complete control over where data should be stored and who a party can share those data with. A user can use his digital wallet to keep his data privately stored. Once that user cannot access his wallet,

that means permanently losing the data stored there. Some SSIM solutions does not provide any option to resolve that and tend to put the responsibility on the user to make his wallet accessible. Other SSIM solutions provide one or more options among three to allow retrieving the user's data on his wallet. First, a possibility of retrieving it through a trusted relationship with others, either they were individuals or organizations. Second, a possibility of retrieving it from a cloud service provider. Third, the possibility of retrieving it through a recovery phrase. However, the study noticed that the identity recovery options introduced by current SSIM solutions does not remove the reliability of building a circle of trust with specific parties. Therefore, the possibility of losing the data and mandatorily building a trust with selected parties still exists in SSIM solutions.

In contrast, current SSIM solutions allow sharing identity attributes with anyone without restricting that with a specific group of parties. Also, despite the possibility of building a circle of trust in some SSIM solutions to recover the lost data associated with the user's identity, those solutions enable a user to control his circle of trust instead of a third party. And regarding the issue of the ability of a third party to control each circle of trust and identify which SP a user can connect with, SSIM solutions that utilize this method have reordered this role by assigning the user himself in place of a third party where a user can select which the parties can include inside his circle. Besides, SSIM solutions built their systems to let the user's interactions occur with complete reliance on the blockchain to help check whether or not a specific party deserves to be trustworthy. This method assists in removing the possibility of direct connectivity between IDP and SP once they need to connect to verify the validity of a user credential, and even though that verifying will be done indirectly, there is no opportunity to track a user's activities since the identities of IDP and SP should not be revealed to each other during the connectivity except their public profiles recorded onChain. There is an exceptional case in SSIM solutions represented in the ShoCard architecture. This solution has mainly relied on its server to complete such a data exchange procedure among all ShoCard's parties. It is handled as a party responsible for giving other parties the validation over the introduced credential and storing some data locally to utilize it during that validation. Since this server acts as an intermediary between the ShoCard user and SP, it might be able to track data exchange procedures among all parties and associate the parties' identities if needed. Besides, it is not a completely decentralized system and this back to the centralized controlling point that ShoCard provides, which contains a single point of control and failure because ShoCard users cannot prove themselves without referring to that server. On the other hand, all current SSIM solutions are not threatened by the possibility of accessing all the user's logged-in websites once an attacker steals his credential since it does not rely on such a federated identity. Each website accessed by a user in SSIM solutions should have isolated connectivity than the other websites, which removes any attempt to govern the other connections, even if a credential has been stolen.

6.4 Limitations of Self-Sovereign Identity Management Solutions

The results of the comparison criteria mentioned previously in section 6.2 have assisted in figuring out some of the shortcomings in current SSIM solutions. Some of those shortcomings were found in the design of the solution itself, and some were inherited from blockchain, considering that all SSIM solutions rely on the fundamental technology to provide such a decentralization. Below the study mention those solutions' limitations in further detail.

- **Deletion in SSIM solutions**

Data deletion is a crucial issue in SSIM solutions since the procedure of this identity operation cannot occur in a completed way. Blockchain is an underlying technology for all those solutions; thus, this limitation does not directly refer to the SSIM solution; instead, it refers to the nature of blockchain enabling such a decentralization environment for these solutions. This obstacle was previously mentioned in the studies [40][44], and the current study confirmed that by analyzing SSIM solutions in chapter three. However, any data recorded on the blockchain can only partially be removed once a user wants to perform that. This backs to the immutability nature and consensus mechanisms that formed such fundamental features of blockchain architecture [85][86].

- **Trust in SSIM solutions**

Although the SSIM solutions aim to follow a trustlessness approach among the participants on the network, only some of those solutions have complied with this approach [42][45]. A study noticed that all SSIM solutions that have a partial decentralization have a trusted relationship with a specific third party. This party is responsible for providing a service to other parties for data storage or facilitating data exchange. On the other hand, some SSIM solutions empower a user to build a trusted relationship with a group of people to retrieve his identity once it is lost. Accordingly, a user experience might be incompatible with SSIM goals, specifically in achieving trustlessness among all parties [41]. Besides, blockchain technology has a community that shares the decisions and votes of different transactions and utilizes the trust to accomplish such a consensus mechanism. The aim behind this mechanism is to ensure blockchain records are authentic and not fake. The existence of such communities raises many questions about the 'zero-trust' claim by many

blockchain service providers. This issue should be examined and studied in further detail in future work.

- **Decentralization in SSIM solutions**

There are different levels of decentralization in SSIM solutions, as mentioned previously in [40][42][43][45]. The current study confirmed this limitation during analyzing SSIM solutions in chapter three. Even though the main goal of those solutions was to remove a third-party role enabled by centralized environments, some SSIM solutions bring that role into their environments for different purposes. The first category utilized a third-party service to provide backup storage regarding the user's private data. The second category utilized its own servers to manage data exchange processes among network participants, where they could know each connectivity that occurred among the parties and collect the data about those activities if needed.

- **Security and privacy in SSIM solutions**

Since the fundamental goal of all current SSIM solutions is to enable the user to manage and control his data without any other party's intervention, the study noticed that most solutions that are designed to be fully decentralized solutions are more secure rather than the solutions that have partial decentralization. This is because the solutions built on partial decentralization have taken advantage of third-party roles to obtain certain services. For example, backup services in SSIM solutions are either through a cloud service provider or the central server of the solution itself, which might control the accessibility and availability of the user's data and further collect some data regarding a user's activities if needed. Moreover, this is inconsistent with SSIM goals regarding owning a user to his data and being able to manage those data securely [41]. On the other hand, some solutions decided to provide an option to recover the lost identity of a user through connecting with other parties, which leads to a significant security concern if those parties are malicious or a specific attack has previously compromised even their identities.

- **Usability and Scalability in SSIM solutions**

Scalability is a significant point that directly affects any system's usability and wide-scale adoption. Even though current SSIM solutions strive to provide such a user-centred design that brings most of its features around a user and works to achieve effective user and business partner experience, there are obstacles inherited from the blockchain such as scalability and useability [38][40][43][45]. A scalability issue is more notable in the solutions that utilize multiple blockchains and the public Ethereum blockchain. Hence, if a network can process many transactions per second through sizable computational power to perform that, it is considered a scalable network. In contrast, SSIM solutions have various levels of computational power and consensus protocol utilized to validate transactions. Blockchain-powered SSIM solutions have different nodes that must be involved and participate in the consensus process. Those nodes play a fundamental role in keeping the network updated by relaying the transactions to other nodes. However, useability is associated with a scalability issue; once a solution cannot be scalable, it is reflected in the usability of that system which affects making it widely used.

6.5 Enhancement Solutions

A study in this section provides a possible enhancement that could address or mitigate the impact of existing limitations in SSIM solutions. The study would clarify that some of the above limitations, such as the difficulty of performing a complete data deletion, are considered an obstacle inherited from blockchain, and so far, no references mention such a proper method to overcome that. For the solutions that relied on cloud storage to provide such a backup to the user's data, the study presents below an method that can manage the user's keys and thus ensure that data stored locally inside such a user wallet can be accessible again even if the user lost his mobile phone. Also, a user does not need to build any trust relationship with any party to have the opportunity of restoring his lost key or even handling with a third-party service to keep a backup of his data. Regarding the scalability and useability limitations in SSIM solutions, the current study mentioned that the usability limitation resulted from scalability limitation where the pushed transaction into the blockchain can take a long processing time to be done and thus restrict a user experience in interacting with those solutions regarding accomplishing each transaction. Due to this association, the study would mention that the study [87] presents a comprehensive survey about all the solutions proposed to address scalability issue in various blockchains and performs a taxonomy to clarify the effective solutions. However, the study sheds light in this section on enhancing the method of cryptographic key management since it has a relation with a number of limitations mentioned above.

Datarella [88] introduced a solution, namely everKEY, to achieve an independent key recovery mechanism that fundamentally removes the need to keep valid connectivity with cloud agents or even a circle of trusted people for the purpose of recovering the lost private key associated with the user's data in a digital wallet. An everKEY has been proposed to tackle the obstacle facing various SSIM solutions that utilize the PKI mechanism in case of losing the private key. Moreover, everKEY allows a user to authenticate himself independently and obtain full control over his data thereby no party can preserve such a piece of the user's private data in a purpose of assisting him to recover his data once it becomes inaccessible. Figure 25 shows how everKEY works.

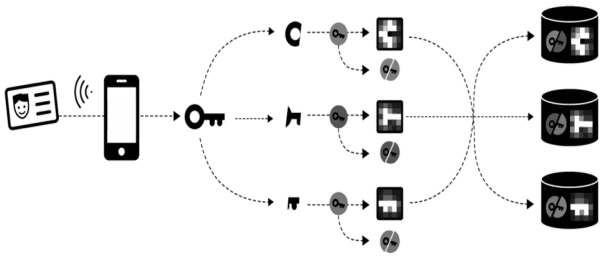

Fig 25. everKEY solution

First, this solution allows a user to verify himself by using his legal identity during the authentication process and once a user accomplishes that successfully he can recover cryptographic keys associated with his identity. Therefore, the procedures of storing those keys once it was generated goes through three stages. First, everKEY performs a key sharding for the key associated with the user's authentication. Second, for the key pieces resulting from the first sharding process, everKEY will handle each piece as a new key, and further, it will split those keys into two parts: the first part will perform the second round of the sharding process, and the other part will be a hash value of that key. Third, everKEY performs an exchange between each piece previously sharding in the second stage and thus stores them along with the hash values in such decentralized secure storage across multiple locations on the blockchain. However, everKEY relies on a complex method of guessing the value of cryptographic keys associated with the user's data stored on the digital wallet. In general, everKEY is consistent with SSIM goals that turn around a user owning and managing his data and the associated cryptographic keys of those data by himself without relying on an intermediary to perform that on the user's behalf. Even though the practical part of everKEY [88] needs to be clarified more since there is a lack of references about this emerging solution yet, it will resolve various limitations that already exist in most SSIM solutions. Below the study mentions SSIM limitations that can be resolved by everKEY.

- **Trust**

Since some SSIM solutions prefer utilizing trusted people as one of the key recovery methods, everKEY removes the need to have valid connectivity with them or even build a trust relationship with others to make user data accessible once he cannot access it. An everKEY allows a user to backup and recover his data in such an independent method empowering a user to have control over his data which is the primary goal of the SSIM model, and further ensures that no party could preserve those data except a user himself.

- **Security and privacy**

An everKEY can mitigate the security and privacy concerns regarding owning such a part of a user's private data by others (e.g. cloud service providers or trusted people) and further remove the concept of a single point of control and single point of failure that can be applied once a third party can own a data associated with a specific user and thus a user can rely on that party to manage his data as well. Also, everKEY removes the possibility of relying on another party to recover the lost identity that perhaps that party may be malicious or even misuse the preserved data at him.

- **Decentralization**

Since most SSIM solutions have a different level of decentralization due to the nature of how those solutions work, everKEY assists in achieving a decentralized approach that removes the role of central authority that manages the users' identities on their behalf. However, there is a fundamental advantage obtained by utilizing everKEY: there is no need to backup the user's data through a central cloud server and, further, no need to grant third parties the right to store some data. Therefore, everKEY theoretically is consistent with SSIM goals in granting the user the right to manage his data by himself and encouraging SSIM solutions to achieve decentralized environments that are compatible with the SSIM model.

7. Conclusion

This study aimed to introduce an overview of the emerging identity management model, self-sovereign identity. A study placed three main questions to shed light on that model from different aspects. First, a place of self-sovereign identity model among previous identity management models. The outline of this aspect was to study the most well-knowing identity management models and present the use cases relevant to those models aiming to clarify the advantages and disadvantages for each and providing a comparison that highlights most differences between them. Second, the main goal of question two was to analyze most identity management systems that have been introduced to adapt with the emerging SSIM model and investigate whether or not those systems are able to overcome the identity management issues mentioned in chapter two. Third, the outlines of question three shed light on the limitations and the possible enhancements of SSIM solutions.

This study built the research questions to bridge the current gap regarding the emerging SSIM model and the proposed solutions that attempted to integrate with it. This study obtained its motivation from the need to perform a systematic comparison covering the most available SSIM solutions. The current study noticed a need for more

academic research that turned around the term self-sovereign identity. Furthermore, aiming to remove the ambiguity of this emerging concept, the current study presented a comprehensive overview of the SSIM model and the associated solutions of self-sovereign identity. A study extends the work of related studies to provide a comparison review of most SSIM solutions and figure out their goals, limitations, and architectures. Some results have previously been mentioned in the related studies, and the current study confirmed those results after investigating SSIM solutions. Since this model aims to grant such a full control to the users over their identities, the current study sheds light on the truth of having the users the required ability to manage their identities by themselves through defining some fundamental criteria inherited from various aspects that support user experience such in a decentralized approach.

The results of this study show that there are different levels of controlling the data granted to a user in SSIM solutions. Each solution's level of control differs based on the prerequisites of design, goals, and underlying technologies supported. Some solutions, such as ShoCard and Sphere Identity, promise the users to have full control over their data. On the other hand, that control allows that solution to perform some central authority practices, as in the Silo and Federated identity management models. Besides, other solutions such as EverID, Sora, Civic, and Dock also promised the users to own their data, and in fact, those data can be stored at a third party, such as cloud service providers. Accordingly, there is an explicit conflict between the actual working environment of SSIM solutions and the goals mentioned previously.

The study noticed that some SSIM issues, such as scalability, useability, and deletion, were inherited from blockchain since it is the powerful feature of decentralization to SSIM solutions. These issues affect the adoption of these solutions by some business partners, making them unattractive to be widely used. Some other issues, such as trust, decentralization, security and privacy, were related to the design of SSIM solutions and the way these solutions work.

A study concluded by mentioning the available enhancements that could address the trust and decentralization limitations in SSIM solutions. Regarding the trust point, the study focused on the purpose behind using a trust: restoring the user's lost keys through trusted parties. Thus, this study mentioned another option to utilize without building a trusting relationship with others. Also, this option can be helpful to own the data by their owners without relying on such cloud storage to provide a backup of those data.

8. Recommendations

After investigating the emerging SSIM model and going through the available solutions, a study achieved its objectives by answering each research question predefined in chapter one. The results motivate the study to propose the following recommendations:

- Some SSIM solutions have utilized the term self-sovereign identity to follow the market trends, which has led to conflicts between its goals that they claim to be compatible with SSIM and the actual working environment. This study recommends that redefine those solutions to be blockchain-based solutions instead of self-sovereign identity solutions to address that conflict.
- The current study aimed to figure out the emerging SSIM model by introducing three fundamental questions that motivated the study to follow the comparison method to find proper answers. The scope of the study was limited to those questions, and therefore, the study recommends extending the current scope by proposing suitable methods to measure the performance of SSIM solutions and identify the industry field that it follows.
- The possibility of Interoperability between DID and VC standards-based solutions needs to be explored, and the study recommends bridging the gap in the future.
- The study recommends further investigation about the communities associated with different kinds of blockchains and their impact on the infrastructure of blockchain’s trustlessness, as it is the underlying technology on which the SSIM model is based.

9. Future Directions

The study intends in future work to build a comprehensive taxonomy covering most SSIM solutions to measure their performance experimentally. This taxonomy will assist in classifying those solutions based on simulating the user experience across their environments. A classification will divide the solutions into the solutions that have high performance, those that have moderate performance, and those that have low performance. The solution’s performance will be measured by performing the identity core operations predefined in this study and measuring the processing time for each. Also, the study intends to investigate the everKEY solution in further detail, which is proposed to overcome most challenges in key recovery methods in SSIM solutions and perform a use case of one of those solutions and the possibility of applying that solution in it.

References

[1] Quintana, L. and Hermida, J., 2019. El método hermenéutico y la investigación en Ciencias Sociales. Aportes al Derecho, 1(3), pp.1-16.

[2] Azarian, R., 2011. Potentials and limitations of comparative method in social science. International Journal of Humanities and Social Science, 1(4), pp.113-125.

[3] ISO/IEC 24760-1:2019. IT Security and Privacy — A framework for identity management — Part 1: Terminology and concepts. May. 2019.

[4] Alrodhan, W.A. and Mitchell, C.J., 2010, May. Enhancing user authentication in claim-based identity management. In 2010 International Symposium on Collaborative Technologies and Systems (pp. 75-83). IEEE.

[5] ITU-T X.1250: Baseline capabilities for enhanced global identity management and interoperability. Framework. Sep.2009.

[6] Coskun, B. and Herley, C., 2008, September. Can "something you know" be saved?. In International Conference on Information Security (pp. 421-440). Springer, Berlin, Heidelberg.

[7] Allen, C., 2016. The path to self-sovereign identity, Apr. 2016.

[8] Cameron, K., 2009. The laws of identity. 2005. Microsoft Corporation.

[9] Ferris, C., 2004. Web services architecture. Standard, W3C World, p.10.

[10] Hughes, J. and Maler, E., 2005. Security assertion markup language (saml) v2. 0 technical overview. OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08, 13.

[11] Cantor, S., Kemp, J., Philpott, R. and Maler, E. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2. Tech. Rep, 2009.

[12] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, "Decentralized Identifiers v1.0 Core architecture, data model, and representations," W3C Technical report, 2021. [online]. Available: https://www.w3.org/TR/did-core/

[13] M. Sporny, D. Longley, and D. Chadwick, "Verifiable Credentials Data Model 1.1" W3C Technical report, 2021. [online]. Available: https://www.w3.org/TR/vc-data-model/

[14] International Standardization Organization ISO 22739, Blockchain and Distributed Ledger Technologies—Vocabulary,2020(En).[online].Available:https://www.iso.org/obp/ui#iso:std:iso:22739:ed-1:v1:en

[15] S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[16] Bernabe, J.B., Canovas, J.L., Hernandez-Ramos, J.L., Moreno, R.T. and Skarmeta, A., 2019. Privacy-preserving solutions for blockchain: Review and challenges. IEEE Access, 7, pp.164908-164940.

[17] Fraga-Lamas, P. and Fernández-Caramés, T.M., 2019. A review on blockchain technologies for an advanced and cyber-resilient automotive industry. IEEE access, 7, pp.17578-17598.

[18] Zheng, Z., Xie, S., Dai, H., Chen, X. and Wang, H., 2017, June. An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress) (pp. 557-564). IEEE.

[19] Lu, Y., 2019. The blockchain: State-of-the-art and research challenges. Journal of Industrial Information Integration, 15, pp.80-90.

[20] Viriyasitavat, W. and Hoonsopon, D., 2019. Blockchain characteristics and consensus in modern business processes. Journal of Industrial Information Integration, 13, pp.32-39.

[21] D. Yaga, P. Mell, N. Roby, and K. Scarfone, Draft NISTIR 8202: Blockchain Technology Overview. NIST, 2018 [online]. Available: https://csrc.nist.gov/publications/detail/nistir/8202/draft

[22] Aydar, M., Cetin, S.C., Ayvaz, S. and Aygun, B., 2019. Private key encryption and recovery in blockchain. arXiv preprint arXiv:1907.04156.

[23] Akram, S.V., Malik, P.K., Singh, R., Anita, G. and Tanwar, S., 2020. Adoption of blockchain technology in various realms: Opportunities and challenges. Security and Privacy, 3(5), p.e109.

[24] Hardjono, T., Lipton, A. and Pentland, A., 2019. Toward an interoperability architecture for blockchain autonomous systems. IEEE Transactions on Engineering Management, 67(4), pp.1298-1309.

[25] Hepp, T., Sharinghousen, M., Ehret, P., Schoenhals, A. and Gipp, B., 2018. On-chain vs. off-chain storage for supply-and blockchain integration. it-Information Technology, 60(5-6), pp.283-291.

[26] Eberhardt, J. and Heiss, J., 2018, December. Off-chaining models and approaches to off-chain computations. In Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers (pp. 7-12).

[27] Kim, S.T., 2020. Bitcoin dilemma: Is popularity destroying value? Finance Research Letters, 33, p.101228.

[28] Greenspan, G., 2015. Multichain private blockchain-white paper. URl: http://www. multichain. com/download/MultiChain-White-Paper. pdf, pp.57-60.

[29] Buterin, V., 2013. Ethereum white paper. GitHub repository, 1, pp.22-23.

[30] Dhulavvagol, P.M., Bhajantri, V.H. and Totad, S.G., 2020. Blockchain ethereum clients performance analysis considering E-voting application. Procedia Computer Science, 167, pp.2506-2515.

[31] HYPERLEDGER, "An Overview of Hyperledger Foundation," 2021, [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2021/11/HL_Paper_HyperledgerOverview_102721.pdf

[32] Schwartz, D., Youngs, N. and Britto, A., 2014. The ripple protocol consensus algorithm. Ripple Labs Inc White Paper, 5(8), p.151.

[33] Gürcan, Ö., Del Pozzo, A. and Tucci-Piergiovanni, S., 2017, October. On the bitcoin limitations to deliver fairness to users. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 589-606). Springer, Cham.

[34] Brown, R.G., Carlyle, J., Grigg, I. and Hearn, M., 2016. Corda: an introduction. R3 CEV, August, 1(15), p.14.

[35] Morgan, J.P., 2016. Quorum whitepaper. New York: JP Morgan Chase.

[36] Buterin, V., 2014. A next-generation smart contract and decentralized application platform. white paper, 3(37), pp.2-1.

[37] Baliga, A., Subhod, I., Kamat, P. and Chatterjee, S., 2018. Performance evaluation of the quorum blockchain platform. arXiv preprint arXiv:1809.03421.

[38] Naik, N. and Jenkins, P., 2020, April. Self-Sovereign Identity Specifications: Govern your identity through your digital wallet using blockchain technology. In 2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) (pp. 90-95). IEEE.

[39] Cameron, K., 2009. The laws of identity. 2005. Microsoft Corporation. - duplicated

[40] Satybaldy, A., Nowostawski, M. and Ellingsen, J., 2019, August. Self-Sovereign Identity Systems. In IFIP International Summer School on Privacy and Identity Management (pp. 447-461). Springer, Cham.

[41] El Haddouti, S. and El Kettani, M.D.E.C., 2019, April. Analysis of identity management systems using blockchain technology. In 2019 International Conference on Advanced Communication Technologies and Networking (CommNet) (pp. 1-7). IEEE.

[42] Dunphy, P. and Petitcolas, F.A., 2018. A first look at identity management schemes on the blockchain. IEEE security & privacy, 16(4), pp.20-29.

[43] Kaneriya, J. and Patel, H., 2020, December. A Comparative Survey on Blockchain Based Self Sovereign Identity System. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (pp. 1150-1155). IEEE.

[44] Liu, J., Hodges, A., Clay, L. and Monarch, J., 2020, September. An analysis of digital identity management systems-a two-mapping view. In 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) (pp. 92-96). IEEE.

[45] Gilani, K., Bertin, E., Hatin, J. and Crespi, N., 2020, September. A survey on blockchain-based identity management and decentralized privacy for personal data. In 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) (pp. 97-101). IEEE.

[46] Ferdous, M.S., Chowdhury, F. and Alassafi, M.O., 2019. In search of self-sovereign identity leveraging blockchain technology. IEEE Access, 7, pp.103059-103079.

[47] Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R. and Maler, E., 2005. Profiles for the oasis security assertion markup language (saml) v2. 0. OASIS standard.

[48] Grassi, P., Garcia, M. and Fenton, J., 2020. Digital identity guidelines (No. NIST Special Publication (SP) 800-63-3). National Institute of Standards and Technology.

[49] Li, C. and Palanisamy, B., 2018, October. Decentralized release of self-emerging data using smart contracts. In 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS) (pp. 213-220). IEEE.

[50] Nofer, M., Gomber, P., Hinz, O. and Schiereck, D., 2017. Blockchain. Business & Information Systems Engineering, 59(3), pp.183-187.

[51] Parameswaran, M., Susarla, A. and Whinston, A.B., 2001. P2P networking: an information sharing alternative. Computer, 34(7), pp.31-38.

[52] Conti, M., Kumar, E.S., Lal, C. and Ruj, S., 2018. A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials, 20(4), pp.3416-3452.

[53] Upadhyay, N., 2020. Demystifying blockchain: A critical analysis of challenges, applications and opportunities. International Journal of Information Management, 54, p.102120

[54] P. Windley and D. Reed, "Sovrin: A protocol and token for self sovereign identity and decentralized trust,"

Sovrin Foundation whitepaper, 2018. [online]. Available: https://sovrin.org/wp-content/uploads/Sovrin-Protocol-and-Token-White-Paper.pdf

[55] Evernym. [Online]. Available: https://www.evernym.com/

[56] Hyperledger Indy. [Online]. Available: https://www.hyperledger.org/projects/hyperledger-indy

[57] uport.me. [Online]. Available: https://developer.uport.me/

[58] Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z. and Sena, M., uPort: A platform for self-sovereign identity. white paper, 2017. [online]. Available: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf.

[59] Reid, B., Witteman, B. and Brad, W., 2018. Everid whitepaper. EverID, techreport, May.

[60] Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C. and Leung, V.C., 2018. Decentralized applications: The blockchain-empowered software system. IEEE Access, 6, pp.53019-53033.

[61] LifeID. [online]. Available: https://lifeid.io/

[62] LifeID, "An open-source, blockchain-based platform for self-sovereign identity," LifeID, Tech. Rep.[Online]. Available: https://lifeid.io/whitepaper.pdf

[63] Soramitsu. [Online]. Available: https://soramitsu.co.jp

[64] "Hyperledger Iroha", Available on: https://github.com/hyperledger/iroha

[65] Takemiya, M. and Vanieiev, B., 2018, July. Sora identity: Secure, digital identity on the blockchain. In 2018 ieee 42nd annual computer software and applications conference (compsac) (Vol. 2, pp. 582-587). IEEE.

[66] SelfKey. [online]. Available: https://selfkey.org/

[67] SelfKey, "Selfkey," The SelfKey Foundation, Tech. Rep., Sep. 2017. [Online]. Available: https://selfkey.org/wp-content/uploads/2017/11/selfkey-whitepaper-en.pdf

[68] Civic Secure Identity Ecosystem—Decentralized Identity & Reusable KYC. [Online]. Available: https://www.civic.com.

[69] Lingham, V. and Smith, J., Civic white paper, 2018. [online]. Available: https://tokensale. civic. com. CivicTokenSaleWhitePaper. Pdf.

[70] Identity.com, [online]. Available: https://www.identity. com/

[71] Blockstack. [Online]. Available: https://blockstack.org

[72] Ali, M. Stacks 2.0 Apps and Smart Contracts for Bitcoin. Stacks whitepaper, 2020. [online]. Available: https://coinpaprika.com/storage/cdn/whitepapers/10650531.pdf

[73] Ali, M., Shea, R., Nelson, J. and Freedman, M.J., 2017. Blockstack Technical Whitepaper. 2017. [online]. Available: http://nil.lcs.mit.edu/6.824/2020/papers/blockstack-2017.pdf

[74] Shocard .[online]. Available: https://shocard.com

[75] ShoCard, S.I.T.A., 2016. Travel Identity of the Future–White Paper.

[76] Jolocom. [Online]. Available: http://jolocom.io.

[77] Robles, K. and Appelcline, S., 2016. Hierarchical Deterministic Keys for Bootstrapping a Self-Sovereign Identity. Retrieved April, 28, p.2019.

[78] Jolocom, J., a decentralized, open source solution for digital identity and access management, Jolocom white paper, 2019. URL https://github. com/jolocom/jolocom-lib/wiki/Jolocom-Whitepaper.

[79] Dock.[online]. Available: https://www.dock.io/

[80] Dock, decentralized data exchange powered by Ethereum, Whitepaper Dock Protocol V0.5. Mar. 2018.

[81] Sphere Identity.[online]. Available: https://sphereidentity.com/en/

[82] Sphere Identity, Sphere Identity Whitepaper V1.1, May. 2019.

[83] NuID. [online]. Available: https://nuid.io/

[84] NuID: A Model for Trustless, Decentralized Authentication and Self-Sovereign Identity, whitepaper, NuID, 2017. [online]. Available: https://nuid.io/pdf/nuid-white-paper.pdf

[85] Summers, A., 2022. Understanding Blockchain and Cryptocurrencies: A Primer for Implementing and Developing Blockchain Projects. CRC Press.

[86] Bashir, I., 2022. Blockchain Consensus: An Introduction to Classical, Blockchain, and Quantum Consensus Protocols

[87] Hafid, A., Hafid, A.S. and Samih, M., 2020. Scaling blockchains: A comprehensive survey. IEEE Access, 8, pp.125244-125262.

[88] Datarella. [online]. Available: https://datarella.com/everid/

Figure References

Figure 1. Alrodhan, W.A. and Mitchell, C.J., 2010, May. Enhancing user authentication in claim-based identity management. In 2010 International Symposium on Collaborative Technologies and Systems (pp. 75-83). IEEE.

Figure 2. ISO/IEC 24760-1:2019. IT Security and Privacy — A framework for identity management — Part 1: Terminology and concepts. May. 2019.

Figure 3. DID standard. [online]. Available: https://www.w3.org/TR/did-core/#dfn-did-resolvers

Figure 4. DID standard. [online]. Available: https://www.w3.org/TR/did-core/#dfn-did-resolvers

Figure 5. DID standard. [online]. Available: https://www.w3.org/TR/did-core/#dfn-did-resolvers

Figure 6. VC standard. [online]. Available: https://www.w3.org/TR/vc-data-model/

Figure 7. VC standard. [online]. Available: https://www.w3.org/TR/vc-data-model/

Figure 8. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf

Figure 12. Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P. and Scavo, T., 2008. Security assertion markup language (saml) v2. 0 technical overview. OASIS Committee Draft, 2.: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

Figure 13. Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P. and Scavo, T., 2008. Security assertion markup language (saml) v2. 0 technical overview. OASIS Committee Draft, 2.: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

Figure 25. Datarella. [online]. Available: https://datarella.com/everid/

NOOT ALISSA received the B.Sc. degree in computer science from Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia, in 2018 and the M.Sc. degree in information security from Imam Mohammad bin Saud Islamic University, Riyadh, Saudi Arabia, in 2023. She is currently working as a software developer in Ministry of Justice, Saudi Arabia, and her research interests includes identity management systems, software development techniques securely.

WALEED ALRODHAN has received his B.Sc. degree in Computer Sciences from King Saud University, his M.Sc. degree (with distinction) and Ph.D. degree in Information Security from Royal Holloway, University of London. Currently, he is an Information Security Associate Professor and a member of the university's Scientific Council. Before that, he was the Dean of the College of Computer and Information Sciences at Imam Muhammed ibn Saud University for five years, and the Head of the Cyber Defense Centre at a confidential governmental organization. His research interests include privacy, identity management, federated identity, single sign-on, and secure web-based protocols.

APPENDIX 1

Ref	Number of solutions	Contribution	Existing gap
38	two	- Proposing SSIM specifications that are extended from Kim Kameron [39] and Christopher Allen [7]. - briefly explained some challenging issues in identity management models.	- There's a need to extend the scope of study to cover more SSIM solutions. - the study provides a brief overview of most of the identity management models without entering into the technical details or discussing the security aspects for each.
40	five	- investigating state-of-the-art developments adapted with SSIM through utilizing existing frameworks. - point out some significant shortcomings in the existing solutions.	- the scope of study, which included five SSIM solutions. - there's a need to provide a review of evolution of the identity management models and how these models differ, instead of focusing on the SSIM model to reach a better understanding about it.
41	three	- provide a brief overview of blockchain technology. - analyze some SSIM solutions by utilizing existing frameworks and investigating its features, components, and operating environment. - discuss some barriers that need to develop.	- the scope of study needs to cover more SSIM solutions. - the study briefly presented an overview of identity management models, then focused on the SSIM without pointing out the main differences between them.
42	three	- point out most features of applying DLT to identity management. - classifying DLT-based solutions into two categories: Self-Sovereign identity and decentralized trusted identity. - analyzing some solutions by utilizing existing frameworks and pointing out their challenges.	- the scope of study needs to cover more SSIM solutions - providing a review of the SSIM concept and how it varies from other identity management models is missing.
43	six	- discuss fundamental principles concerning SSIM along with its architectural components. - discuss features and drawbacks of some SSIM solutions	- the scope of study needs to extend the investigation to cover more solutions. - the study clarifies most significant aspects of SSIM without reviewing previous identity management models and presents the differences between them.

44	-	- Provides two propositions: the core operations and trust model built on the mapping data. - conduct a comparison between centralized and decentralized identity management systems. - presents some challenges in decentralized identity management.	- the study conducts a comparison between two types of identity management systems, centralized and decentralized, with no mention of examples of such systems. - there's a need to explain the SSIM concept in more detail and point out its basic aspects. - it doesn't mention the blockchain and its components considering the fundamental part of decentralized identity management systems.
45	eight	- presents an overview of the identity management approach and briefly explains the impact of managing identity using blockchain. - compared some SSIM solutions based on the technical evaluation. - discussed some corresponding challenges towards building complete identity management systems.	- the scope of study needs to be extended, involving more solutions. - the study provides a brief overview of identity management models with no discuss the security aspects, besides barriers it could face.
46	four	- presented & analyzed the existing SSIM definitions. - conducted a comprehensive taxonomy of SSIM for utilizing it later in the assessment of SSIM solutions.	- The scope of study needs to extend, involving more solutions. - the study introduced a review of e most identity management models without discussing the challenges for each of them.

APPENDIX 2

No.	SSIM solution	Official website	Release date	Relevant	Reference availability
1.	Sovrin	https://sovrin.org/	2017	Related	available
2.	Uport	https://www.uport.me	2017	Related	available
3.	EverID	https://everest.org.	2018	Related	available
4.	LifeID	https://lifeid.io	2019	Related	available
5.	Sora	https://www.soraid.com/	2019	Related	available
6.	SelfKey	https://selfkey.org.	2018	Related	available
7.	Civic	https://www.civic.com/	2017	related	available
8.	Identity.com	https://www.identity.com/	2019	related	available
9.	ShoCard	https://shocard.com.	2017	Related	available
10.	Blockstack	:https://blockstack.org	2018	Related	available
11.	Jolocom	https://jolocom.io/	2022	related	available
12.	Dock	https://www.dock.io/	2021	related	available

13.	Sphere Identity	https://sphereidentity.com/	2017	related	available
14.	NuID	https://nuid.io/	2017	related	available
15.	Verity	https://www.evernym.com/verity	2018	Related	not available enough references
16.	Connect.me	https://connect.me	2018	Related	not available enough references
17.	Bitnation	https://bitnation.co	2014	Not related	-
18.	Credits	https://credits.com	2018	Not related	-
19.	Cordentity	https://www.corda.net/blog/meet-cordentity	2018	Related	not available enough references
20.	BitMark	https://bitmark.com	2016	Related	not available enough references
21.	BLOCKDENTITY	http://blockdentity.com	2018	Related	not available enough references
22.	1Kosmos BlockID	https://www.1kosmos.com	2018	not related	-
23.	Cove Identity	https://www.coveidentity.com	2017	Related	not available enough references
24.	tron	https://tron.network	2018	not Related	-
25.	meeco	https://www.meeco.me	2018	Related	not available enough references
26.	Ontology	https://ont.io	2021	Related	not available enough references
27.	Sudo Platform	https://sudoplatform.com	2016	Related	not available enough references
28.	trinsic	https://trinsic.id/	2020	Related	not available enough references
29.	IOTA (MIOTA)	https://www.iota.org	2017	Related	not available enough references
30.	TrustNet PK	https://trust.net.pk	2020	Related	not available enough references
31.	MintHealth	https://www.minthealth.io	2018	Related	not available enough references
32.	Blockpass	https://www.blockpass.org	2017	Related	not available enough references

33.	Bloom	https://bloom.co	2017	Related	not available enough references
34.	Colendi	https://www.colendi.com	2018	not related	-
35.	Datum	https://datum.org	2017	Related	not available enough references
36.	idento.one	https://www.idento.one	2016	Related	not available enough references
37.	persona	https://persona.im	2017	Related	not available enough references
38.	Pillar	https://www.pillar.fi	2017	not Related	-
39.	Rate3	https://www.rate3.network	2018	Related	not available enough references
40.	VETRI	https://vetri.global	2017	Related	not available enough references
41.	Procivis SSI+	https://www.procivis.ch/en/procivis-ssi	2022	Related	not available enough references
42.	Tierion Network	https://tierion.com	2017	Related	not available enough references
43.	idunion	https://idunion.org	2020	related	not available enough references
44.	WeIdentity	https://weidentity.readthedocs.io	2019	related	not available enough references
45.	TENZ-ID	https://www.tenzorum.org/tenz_id	2018	related	not available enough references
46.	taqanu	https://www.taqanu.com	2016	not related	-
47.	SpidChain	http://www.spidchain.com	2015	not related	-
48.	EDDITS	https://eddits.io	2018	related	not available enough references
49.	dominode	http://www.dominode.com	2016	related	not available enough references
50.	Mattr	https://mattr.global/	2019	related	not available enough references
51.	ORGiD	https://windingtree.com/	2020	related	not available enough references

52.	WWPass Electronic Identity	https://www.wwpass.com/electronic-identity	2018	related	not available enough references
53.	VidChain	https://www.validatedid.com/vidchain	2017	related	not available enough references
54.	Peer Mountain	https://peermountain.com	2017	related	not available enough references
55.	Indicio TestNet	https://indicio.tech/indicio-testnet	2020	related	not available enough references
56.	idRamp	https://idramp.com	2016	related	not available enough references
57.	HearRo	https://www.hearro.com	2016	related	not available enough references
58.	finema	https://finema.co	2017	related	not available enough references
59.	Elliptic	https://www.elliptic.co	2015	not related	-
60.	Edufied	https://edufied.network	2018	related	not available enough references
61.	smart-id	https://www.smart-id.com	2017	not related	-
62.	Credify	https://www.credify.one	2019	related	not available enough references
63.	CREDEBL	https://www.credebl.id	2019	related	not available enough references
64.	Blockster	http://www.blockster.global	2019	related	not available enough references
65.	AyanWorks	https://www.ayanworks.com	2015	not related	-
66.	Polygon Technology	https://polygon.technology	2017	not related	-
67.	cognitohq	https://cognitohq.com	2014	not related	-
68.	BlockCypher	https://www.blockcypher.com/	2014	not related	-
69.	BizSecure	http://www.bizsecure.com	2020	not related	-
70.	humbl	https://www.humblpay.com	2020	related	not available enough references
71.	bedrockdb	https://bedrockdb.com	2007	not related	-
72.	Authentiq	https://www.authentiq.com	2015	not related	-

73.	Affinidi	https://www.affinidi.com	2020	related	not available enough references
74.	Accredify	https://accredify.io	2019	related	not available enough references
75.	C-LOG	https://c-log.io	2017	related	not available enough references
76.	Namecoin	https://www.namecoin.org	2011	not related	-
77.	NameID	https://nameid.org	2013	not related	-
78.	DecentID	https://www.lynk-me.com/decentid	2018	related	not available enough references
79.	Blockcerts	https://www.blockcerts.org	2016	not related	-
80.	TrustChain	https://www.trustchain.com	2017	not related	-
81.	Truechain	https://www.truechain.network	2019	related	not available enough references
82.	Neo	https://neo.org	2014	not related	-
83.	THEKEY	http://www.thekey.vip	2017	related	not available enough references
84.	IDchainZ	https://www.chainzy.com/products/idchainz	2018	related	not available enough references
85.	UniquID	https://uniquid.com	2021	related	not available enough references
86.	Netki Platform	https://www.netki.com	2014	not related	-
87.	TransactID	https://www.netki.com/transactid	2016	not related	-
88.	KYC-Chain	https://kyc-chain.com	2016	related	not available enough references
89.	Open Rights Exchange (ORE)	https://ore.network	2017	related	not available enough references
90.	Identifi	https://www.identifi.com	2021	not related	-
91.	HYPR	https://www.hypr.com	2014	not related	-
92.	Guardtime	https://guardtime.com	2008	not related	-
93.	ExistenceID	http://www.existenceid.com	2017	not related	-
94.	CredyCo	http://www.crebaco.org	2018	not related	-

95.	BlockVerify	http://blockverify.io/	2015	not related	-
96.	Cambridge Blockchain	https://cambridgeblockchain.org	2018	not related	-
97.	SimpleID	https://simpleid.xyz	2019	related	not available enough references
98.	athena	https://www.athena-co.io	2021	not related	-
99.	ID2020	https://id2020.org	2016	related	not available enough references
100.	IDHub	https://sath.com/idhub	2005	not related	-