A Survey on Decentralized Identifier Methods for Self Sovereign Identity

Stefano Bistarelli[1], Francesco Micheli[1] and Francesco Santini[1,*]

[1]Dipartimento di Matematica e Informatica, University of Perugia, Italy

Abstract

We survey different approaches to Decentralized Identifiers (DIDs) for managing Self-Sovereign Identity (SSI). This kind of identification is used to give full control of identity to end-users by means of a distributed ledger, without the use of a centralized authority that manages the identity of users. We first describe SSI and then DIDs, by summarizing their characteristics. We list the different format implementations of these digital identities by reporting the type of verifiable data registry on which they are based on. We also report the document status od DIDs, the link to their description, and date of creation and update, with the purpose of understanding how much these projects are popular. Finally, we show which and how many of these DIDs are based on public or non-public ledgers.

Keywords

Decentralized identifiers, Self-Sovereign identity, Identity management

1. Introduction

The last few years featured a sudden explosion in remote work due to the pandemic, which together with an increase in cloud-based services adoption and recent data privacy requirements (e.g., the *European General Data Protection Regulation*, or *GDPR* for short) caused *Identity and Access Management* (*IAM*) systems to face new challenges and threats. Moreover, such a scenario correlated with a rise of malware performing phishing, ransomware attacks, and data exfiltration. The increased opportunity for a larger amount of information to be accessed inappropriately thus requires public and private organizations to focus on the protection of both their own and customer information to ensure service sustainability and to retain user trust. Hence, additional data protection needs to be offered also through increasingly sophisticated IAM schemes and tools.

In existing *Identity Management* (*IdM*), central authorities are in charge of managing identities, and users have little to no control over data sharing and privacy. One distinct digital identity that is created, managed, and controlled by the identity owner is the result of current efforts to abolish the central service providers. *Self-Sovereign Identity* is a type of identity that guarantees user-centric data ownership.

User authentication is developed in a number of central solutions as part of IdM systems. For example, a lot of organizations have built their authentication process around *Opend ID Connect*[1] and *OAuth*[2] standards, but the central authority aspect has not changed. Digital identity authentication verifies that users are who they say they are on the internet. The IdM's primary determinant of trustworthiness is the subject's verification and the preservation of sensitive data. Access control, which differs from

ITASEC 2023: The Italian Conference on CyberSecurity, May 03–05, 2023, Bari, Italy

*Corresponding author.

stefano.bistarelli@unipg.it (S. Bistarelli); francesco.micheli@studenti.unipg.it (F. Micheli); francesco.santini@unipg.it (F. Santini)

[1]OIDC: https://openid.net.

[2]OAuth: https://oauth.net.

IdM and authentication, is the process of approving or rejecting a subject's request for access to an object (i.e., someone or something that wants to utilize a resource) (i.e., resources that a subject wants to use like network, data, application, service, etc.). Access control, in other terms, is a security method that limits who or what can perform a given activity (such as use, read, write, execute, or view) on a particular resource in a computing environment. Most often, this step is finished following successful authentication.

A new class of identifiers called a *Decentralized Identifier* (*DID*) allows for the creation of a verified, decentralized digital identity. A DID can be any topic that the DID's controller chooses (e.g., a person, group, object, data model, abstract entity, etc.).[3] DIDs have been created to be independent of centralized registries, identity providers, and certificate authorities, in contrast to conventional, federated identities. DIDs are URIs that associate a DID subject with a DID document allowing trustable interactions associated with that subject. Each DID document may contain cryptographic information, verification techniques, or other services that offer a variety of ways for a DID controller to demonstrate control over the DID. Services allow for secure communication involving DID subjects. If the DID subject is an information resource, such as a data model, then a DID might offer the capability to return the DID subject itself.

In this paper, we present the many formats in which these digital identities have been implemented together with information on their status as documents, links to their descriptions, and dates of creation and updates. Lastly, we demonstrate which DIDs are based on *public* or *non-public* ledgers and how many there are of each. Anyone can join the network of a *public* blockchain infrastructure without needing to ask for permission; additionally, all network users have access to the shared ledger and can participate in consensus by validating transactions (some examples are the blockchains of *Bitcoin* and *Ethereum*). *Private* networks are only accessible by invitation, which indicates that a centralized authority manages who is permitted access to the network. Additionally, this central entity has the power to assign participants roles, such as granting them the possibility to perform transactions and mining privileges. The existing transactions on the chain can be edited, deleted, or overridden by the same entity (some examples are the *Morpheus Network*[4] and *Corda*[5]). Finally, a *permissioned* blockchain (also *consortium* blockchain) needs the operator's consent in order to join and carry out certain operations. They have an extra layer of access control as a security mechanism, limiting specific on-chain actions to identifiable participants only. While permissioned blockchains allow any node to operate once the operator grants permission, private blockchains only permit known nodes to operate (some examples are *Ripple* and *IBM Food Trust*[6]). For non-public blockchains, we consider all the proposals that do not meet the features of public blockchains, hence private and permissioned. Some of the proposals are *ledger-agnostic*: they are given without specifying the ledger, while some other proposals are based on a registry which is not a blockchain.

The paper has the following structure: after this introduction motivating the use of DIDs (Section 1), we present *Self-Sovereign Identity* in Section 2 and then Section 3 elaborates on DIDs by presenting their structure and features. Section 4 summarizes the different DID methods, while Section 5 overviews the related work. Finally, we wrap up the paper with final conclusions and future work.

[3] W3C recommendation on Decentralized Identifiers (DIDs) v1.0: https://www.w3.org/TR/did-core/#dfn-decentralized-identifiers.
[4] Morpheus Network: https://morpheus.network/.
[5] Corda: https://www.r3.com/corda-platform/.
[6] IBM Food Trust: https://www.ibm.com/products/supply-chain-intelligence-suite/food-trust.

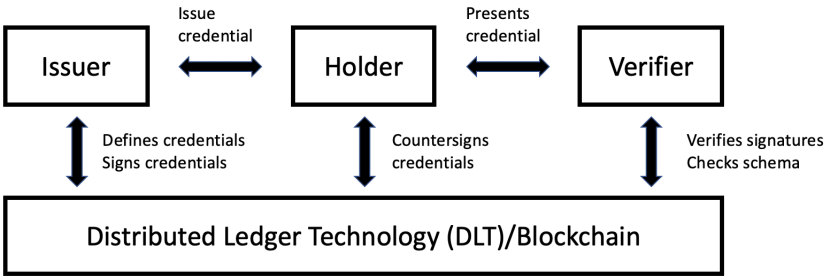


Figure 1: An example of SSI ecosystem [2].

2. Self-Sovereign Identity

The term "identity" is in general considered as a collection of attributes associated with a specific person or entity, given a particular context. A non-human identity may be related to a piece of software or hardware, for example. An identity includes at least one *identifier* (or more) and may also indeed include further attributes associated with that person or entity. For example, human identities may include attributes such as name, age, address, phone number, and job title. On the other hand, non-human identities collect attributes such as an owner, IP address, and perhaps a model or version number. This set of attributes can be used for *authentication* and *authorization*, but also to provide information about the related identity to applications.

A *Digital Identity* [1] is a means for people to prove electronically that they are who they say they are. The history of online identities started with *centralized identities*, then *federated*, and finally *user-centric* identities nowadays. Unfortunately, giving centralized authorities the power to govern a person's digital identity has many of the same drawbacks as giving state authorities control over people's physical identities: users are tied to a single authority who can confirm a false identity, or even deny their own. Power is inherently transferred from the users to centralized entities through centralization. One of the first federated systems was *Microsoft's Passport* (year 1999). It envisioned federated identification, which would enable users to use a single identity across several websites. Nonetheless, it made the federation virtually as centralized as conventional government by placing Microsoft at its heart. Sun Microsystem established the *Liberty Alliance* in response (year 2001). They opposed the concept of centralized power and established a "genuine" federation instead. The outcome, however, was an oligarchy because the power of centralized authority was shared among different organizations (e.g., Intel, Oracle, British Telecom).

Self Sovereign Identity (SSI) is a sovereign, enduring, and portable identity for any person, organization, or body, that allows its owner to access all relevant digital services by utilizing credentials linked to the identity in a privacy-preserving manner [2]. Unlike previous identity management systems where the service provider was at the center of the model, SSI is *user-centric*. In this system, the claim issuer releases the identity by attesting some attributes of the user, in order for him to be authenticated by a relying party. The latter can request claims provided by claim issuers, there must be a relationship of trust between the relying party and the claim issuer. So the SSI ecosystem is composed of three main roles *Issuer*, *Holder* and *Verifier*: **Issuer:** creates and issues credentials to a holder; **Holder:** receives credentials from an issuer, retains it and when it is required, it shares them with a verifier; **Verifier:** receives and verifies credentials presented by a holder. This workflow, which is further detailed in the following of this section, is represented in Figure 1.

The basis of the SSI architecture is the distributed ledger of a blockchain. The blockchain acts as a replacement for the registration authority in classic identity management systems (i.e., IAM) and works

as an *identifier registry*. A *verifiable data registry* (*VDR*),[7] in the context of decentralized identity, is a place where DIDs can be anchored to. Everyone who reads the blockchain can verify the identifier stored in it by posing a challenge to the user or a delegate. Public-private key pairs are the most common authentication method used in SSI solutions, in fact, they are called *Decentralized Public Key Infrastructures*. However, some of the proposals are not based on a blockchain (as we will see in Figure 2). The pairing of *identification* and *authentication* is maintained. The identifier as well as the *verifiable claims* are directly managed by the user. The actual identity claim is stored in user-controlled storage, typically off-chain for privacy considerations.

The SSI architecture relies on mapping an identifier to a specific authentication method that is recorded on a registry [3]. Identifiers can be grouped into three different categories:

- First, identifiers based on random number generation rely on probabilities to avoid collisions.
- Then, centralized identifiers utilize a registration authority in order to assign identifiers and prevent collisions.
- Finally, blockchain technology can help merge the best aspects of both previous approaches, and authentication is typically done with the use of a public/private key pair, where the public key is stored as the value of the identifier on the blockchain [3].

The underlying architecture of the SSI is the blockchain for the registration authority in classic identity management systems. In fact, it is also called *identifier registry*.

In defining the key characteristics of SSI, ten principles underlying this model were defined: these principles have been grouped into three sections in [4], as shown in Table 1. *Security* can be summed up as the protection of personal user data and the limiting of data exposure to the minimum required to fulfill a function. A persistent identity was named as a security requirement, but this should not contradict a "right to be forgotten" as stated by Allen [5]. The *Controllability* category as both *control* and *consent* should extend to the removal of the identity, not only the creation and access. The *Portability* of the Identity is essential so that the user can use their identity wherever they want and be independent of any particular identity provider.

The ten principles consist in **Existence**: A user must have an independent existence, this means that an SSI identity can never be "decoupled" from its physical entity and therefore cannot exist exclusively in the digital world. **Control**: a user must control his own identity. Note that this does not preclude other entities, such as users, companies, or institutions, from making assertions about the user and defining certain characteristics or properties. **Access**: the user must have access to his own data. **Transparency**: Systems and algorithms must be transparent. **Persistence**: Identities must be long-lived. At the same time, the user should be able to have an identity if they wish and claims should be modified or removed, this is referred to as the "*right to be forgotten*". **Portability**: Identity information and services must be portable, i.e. not linked to a digital entity, for example, a social network, nor to a specific jurisdiction, such as a State. **Interoperability**: Identities should be as widely usable as possible, they have to be used globally and are not limited to certain businesses or industries. **Consent**: Users must agree to the use of their identity. **Minimization**: Disclosure of claims must be minimized. **Protection**: User rights can be protected and they have priority over the network needs to support the SSI model.

There are three technical elements at the basis of an SSI system:

- **Decentralized Identifiers (DIDs)**: an alphanumeric string that uniquely identifies an entity. For example, some identifiers for an individual may be the name and surname or the tax code, or codes that allow this individual to be recognized unambiguously. Similarly, in an SSI model, *DIDs are alphanumeric codes based on a double cryptographic key system*, stored on the blockchain, which allows an entity to be uniquely identified online.

[7] Verifiable data registry: https://www.w3.org/TR/did-core/#dfn-verifiable-data-registry.

Security	Controllability	Portability
Protection	Existence	Interoperability
Persistance	Control	Transparency
Minimisation	Consent	Access
		Portability

Table 1
Ten Principles of Self-Sovereign Identity [3].

- **Verifiable Claims (VCs)**: any type of attribute associated to an entity. Some equivalents of a VC in the real world are a driving license or a university degree. However, in the SSI model, the VCs are *digital, immutable, and independently verifiable* in each interaction where that specific attribute is required.
- **DID Documents**: a set of data describing the DID subject, including mechanisms, such as cryptographic public keys, that the DID subject or a DID delegate can use to authenticate itself and prove its association with the DID [6].

In this paper, we focus on the first component, i.e., DIDs, which will be detailed in the following sections.

3. Decentralized Identifier (DID)

A *Decentralized Identifier* (*DID*) is a new type of globally unique identifier. DIDs are the core component of a new layer of decentralized digital identity and public key infrastructure (PKI) for the Internet. W3C decentralized identifiers [6], can be seen as a high-level naming scheme, such as *Uniform Resource Name* (*URN*). DIDs are composed of: *i)* a *scheme*, *ii)* a *method*, and *iii)* a *method specific identifier*, which are separated by colons: for example, "id:example:12345". The scheme is the DID keyword, the method part describes the DID method used to store the identity data, and the method-specific identifier is different for each method, and it describes the way to generate the identifier.

DIDs are a new type of identifier that enables verifiable, decentralized digital identity [6]. DIDs design was studied to be separated from centralized registries, identity providers, and certificate authorities. Moreover, the DID design makes it possible for the controller of a DID to prove control over it without the permission of any other party. DIDs are, in the end, *Universal Resource Identifiers* (*URI*s) that associate a DID subject with a DID document allowing trustable interactions associated with that specific subject [6]. In this context, a DID document records cryptographic material, verification methods, and services that enable a DID controller to prove control over the DID while services make it possible to have trusted interactions associated with the DID subject.

The *Decentralized Identity Foundation* (*DIF*) developed a resolver for these paths [7]; a specific driver is developed and maintained for each individual method. It uses the *method type* to decide which driver to use and uses the *method specific identifier* to resolve the DID document stored on the specified blockchain. The DID document is the key part of the decentralized identity [3]. In it, the authentication method is defined to bind the specified identifier to an identity that is in control of a secret key or other data used in the authentication.

DIDs are only the base layer of decentralized identity infrastructure. The higher layer is *Verifiable credentials*, the technical term for a digitally signed electronic credential [8]. DIDs can be used to identify various entities in the Verifiable Credentials ecosystem such as issuers, holders, subjects, and verifiers. DIDs can be used as identifiers for people, devices, and organizations [8].

3.1. DID Document

The DID infrastructure can be thought of as a global *key-value database* in which the database is all the DID-compatible blockchains, distributed ledgers, or decentralized networks where the key is a DID and the value is a DID document. A DID document is a valid *JSON-LD* (*JSON for Linking Data*)[8] object that uses the DID context defined in the DID specification [8]. Developers can use any other representation method, such as *XML* or *YAML*, that is capable of expressing the data model.[9] The DID document includes six optional components:

1. **DID**, so that the DID document is fully self-describing.
2. **Set of cryptographic material**, such as public keys, that can be used for authentication or interaction with the DID subject.
3. **Set of cryptographic protocols** for interacting with the DID subject, such as authentication and capability delegation.
4. **Set of service endpoints** that describe where and how to interact with the DID subject.
5. **Timestamps** for auditing.
6. **JSON-LD signature** if needed to verify the integrity of the DID document.

Listing 1: An example of DID Document.

```
11
    {"@context": [
            "https://www.w3.org/ns/did/v1", "https://w3id.org/security/suites/ed25519-2020/v1
                "],
            "id": "did:example:123",
            "authentication": [{
                    "id": "did:example:123#z6MkecaLyHuYWkayBDLw5ihndj3T1m6zKTGqau3A51G7RBf3",
                    "type": "Ed25519VerificationKey2020", // external (property value)
                    "controller": "did:example:123",
                    "publicKeyMultibase": "zAKJP3f7BD6W4iWEQ9jwndVTCBq8ua2Utt8EEjJ6Vxsf"}],
            "capabilityInvocation": [{
                    "id": "did:example:123#z6MkhdmzFu659ZJ4XKj31vtEDmjvsi5yDZG5L7Caz63oP39k",
                    "type": "Ed25519VerificationKey2020", // external (property value)
                    "controller": "did:example:123",
                    "publicKeyMultibase": "z4BWwfeqdp1obQptLLMvPNgBw48p7og1ie6Hf9p5nTpNN"}],
            "capabilityDelegation": [{
                    "id": "did:example:123#z6Mkw94ByR26zMSkNdCUi6FNRsWnc2DFEeDXyBGJ5KTzSWyi",
                    "type": "Ed25519VerificationKey2020", // external (property value)
                    "controller": "did:example:123",
                    "publicKeyMultibase": "zHgo9PAmfeoxHG8Mn2XHXamxnnSwPpkyBHAMNF3VyXJCL"}],
            "assertionMethod": [{
                    "id": "did:example:123#z6MkiukuAuQAE8ozxvmahnQGzApvtW7KT5XXKfojjwbdEomY",
                    "type": "Ed25519VerificationKey2020", // external (property value)
                    "controller": "did:example:123",
                    "publicKeyMultibase": "z5TVraf9itbKXrRvt2DSS95Gw4vqU3CHAdetoufdcKazA"}]
    }
```

The DID document uses JSON-LD format and is composed of various components, here is a list of them:

- The *context* on the JSON-LD format can be seen as "the context of the conversation" when people communicate with one another on a specific subject. Context is used to map terms, a short word

[8] JSON-LD: https://json-ld.org.
[9] DID representation formats: https://www.w3.org/TR/did-core/#representations.

that may be expanded to IRIs (Internationalized Resource Identifiers). An IRI is built on top of a URI and is defined as a sequence of characters, not as a sequence of octets. The DID Document must have exactly one top-level context statement. The key for this property is *@context* and the value for this key is the URL for the generic DID context *https://w3id.org/did/v1*.

- *id* represents the actual identifier.
- The *publicKey* lists all the public keys whose corresponding private keys are controlled by the entity identified by the DID. The value of the *publicKey* property is an array of public keys. It includes also id and type properties and the value property can be PublicKeyPem, PublicKeyHex, PublicKeyBase58, EthereumAddress, or similar, depending on the format and encoding of the public key.
- The *DID Authentication* is the mechanism by which an entity can cryptographically prove that they are associated with a DID. The value of the *authentication* property is an array of the proof mechanisms, each one includes the type property and embeds or references a public key.
- The DID Service Endpoint includes a *serviceEndpoint* property that is a JSON array of service endpoints. *serviceEndpoint* describes the network address at which a service operates on behalf of an entity. DID Service Endpoints include discovery services, social networks, file storage services, and verifiable claim repository services.
- The last component of a DID Document is the *Proof*. *Proof* contains a *created* property which represent a creation timestamp and an *updated* property that represent the update timpestamp. Both of the timestamps are valid XML DateTime values normalized to UTC 00:00.

4. The DID Method Specification and a Survey

DIDs and DID Documents can be adapted to any modern blockchain, distributed ledger, or other decentralized networks capable of resolving a unique key into a unique value. Defining how a DID and DID document are created, resolved, and managed on a specific blockchain or "target system" is the role of a DID method specification. DID method specifications define the following operations for a particular target system:

1. **Create.** Some DID methods may generate a DID directly from a cryptographic key pair.
2. **Read.** Some DID methods use blockchains that can store DID Documents directly on the blockchain. Others may instruct DID resolvers to construct them dynamically based on attributes of a blockchain record. Others may store a pointer on the blockchain to a DID document stored in one or more parts on other decentralized storage networks such IPFS [9].
3. **Update.** The update operation is the most critical from a security standpoint because control of a DID document represents control of the public keys or proofs necessary to authenticate an entity. Since verification of DID document update permissions can only be enforced by the target blockchain, the DID method specification must define precisely how authentication and authorization are performed for any update operation.
4. **Delete.** DID entries on a blockchain are by definition immutable, so they can never be "deleted" in the conventional database sense. However, they can be revoked in the cryptographic sense. A DID method specification must define how this termination is performed.

4.1. A List of DID Methods

In Table 2 we report all the 168 DID methods that follow the W3C standard [10], which have been collected up to May 2023. We enrich this list by providing the network/DLT on which the proposal is used. Some of these proposals use two different registries, as the *grn* DID, for example. For each

registry we report the fundamental characteristics: if the registry is a *Public ledger* (*Pb*) *Private ledger* (*Pr*), *Permissioned ledger* (*Pd*), *Permissionless ledger* (*Pl*), *Non-Ledger* (*Nl*), and finally *Ledger-agnostic* (*La*).

Figure 2a shows that most of the proposals in Table 2 adopt non-public blockchains, while Figure 2b highlights the fact that Ethereum is the most frequent public blockchain used for DIDs.

Table 3 shows the DID proposals and their status concerning some non-public blockchains for which the last update was in 2022, while Table 4 does the same for all the DID proposals on public blockchains; the proposals updated in 2022 are highlighted in bold. Table 3 and Table 4 provide a Web link to the documentation of the related method: if the DID method is tagged with "app" it means that an implementation is also available, while with "draft" only the draft of a potential DID specification is given. We can see that, despite a large number of proposals, only a few of them has been updated in 2022. A large majority of the proposals, more than 70%, use non-public blockchains, and in the case of public ledgers, Ethereum is definitely the first choice with one DID method out of two adopting this choice (around 51%).

Table 2: All the available DID methods. Question marks point to information that we were not able to find. The acronyms in the third column can be read by using the following legend: *Public ledger* (*Pb*) *Private ledger* (*Pr*), *Permissioned ledger* (*Pd*), *Permissionless ledger* (*Pl*), *Non-Ledger* (*Nl*), *Ledger-agnostic* (*La*). This table is updated to May 2023.

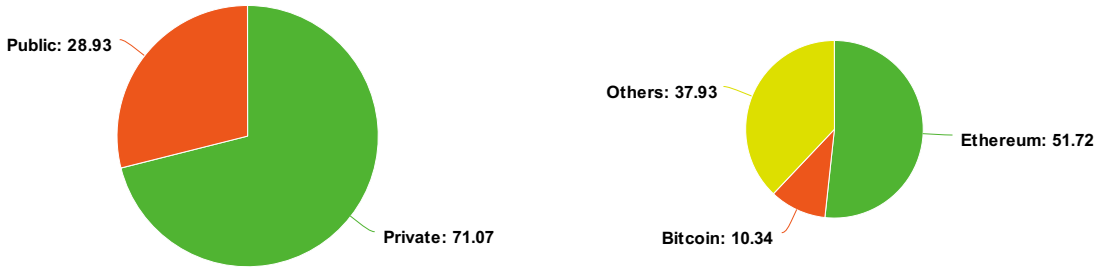
DID Method	**Registry**	**Type**
3	Ceramic Network	Pl Pb
abt	ABT Network	Pd Pb/Pr
aergo	Aergo	Pl Pb
ala	Alastria	Pd Pb
amo	AMO blockchain mainnet	Pd Pr
antelope	Antelope	Pl Pb
art	Artwork ID Method	Nl
asset	Ledger-independent generative DID method based on CAIP-19 identifiers	La
bba	Ardor	Pd Pb
bee	Ledger agnostic	La
bid	bif	Pd Pb
bluetoqueagent	Trusted Digital Web	La
bluetoquedeed	Trusted Digital Web	La
bluetoquenfe	Trusted Digital Web	La
bluetoqueproc	Trusted Digital Web	La
bnb	Binance Smart Chain	Pl Pb
bryk	bryk	Pd Pr
btcr	Bitcoin	Pl Pb
ccf	Confidential Consortium Framework (CCF)	Nl
ccp	Quorum	Pd Pr
celo	Celo	Pl Pb
cheqd	cheqd	Pl Pb
com	commercio.network	Pd Pr
corda	Corda	Pd Pr
cosmos	Cosmos application chains	Pl Pb/ Pd Pr
cot	CoTChain	?
cr	Hyperledger Fabric	Pd Pr
did	Decentralized Identifiers	?
dns	Domain Name System (DNS)	Nl
dock	Dock	Pd Pr
dom	Ethereum	Pl Pb
dsrv	NaN	?
dual	Ethereum	Pl Pb
dxd	fabric.data-alliace.com	Pd Pr
dyne	Dyne.org Foundation (Ethereum)	Pl Pb
echo	Echo	Pl Pb
elastos	Elastos ID Sidechain	Pl Pb
elem	Element DID	Pl Pb
emtrust	Hyperledger Fabric	Pd Pr
ens	Ethereum	Pl Pb

eosio	EOSIO	Pd Pb
erc725	Ethereum	Pl Pb
etho	Ethereum	Pl Pb
ethr	Ethereum	Pl Pb
ev	Any Ethereum or EVM-compatible ledger	Pl Pb
evan	evan.network	Pl Consortium
everscale	Everscale blockchain	Pl Pb
example	DID Specification	?
factom	Factom	Pl Pb
fairx	FairX Node	Nl
future	Netease Chain	Pd Pr
gatc	Ethereum, Hyperledger Fabric, Hyperledger Besu, Alastria	Pl Pb/ Pd Pr
gns	GNU Name System	Nl
grg	GrgChain (EVM)	Pl Pb
grn	Any CosmWasm-compatible ledger	Pl Pb/ Pd Pr
health	DID Health	?
hedera	Hedera Hashgraph	Pl Pb
holo	Holochain	Pl Pr
hpass	Hyperledger Fabric	Pd Pr
hsk	PlatON	Pl Pb
iamx	undefined	?
ibmdc	Hyperledger Fabric	Pd Pr
icon	ICON	Pl Pb / Pr
id	ID Service	?
iid	Inspur Chain	Pd Pr
indy	Any Hyperledger Indy Ledger	Pd Pr
infra	InfraBlockchain	Pl Pb/ Pd Pr
io	IoTeX	Pl Pb/ Pd Pr
ion	Bitcoin	Pl Pb
iota	IOTA	Pl Pb
ipid	IPFS	Nl
is	Blockcore	Pl Pb
iscc	Public Blockchains	?
iwt	InfoWallet	?
jlinc	JLINC Protocol	Pd Pr
jnctn	Jnctn Network	Nl
jolo	Ethereum	Pl Pb
jwk	Ledger agnostic	La
keri	Ledger agnostic	La
key	Ledger-independent DID method based on public/private key pairs	La
kilt	KILT Blockchain	Pl Pb
klay	Klaytn	Pl Pb
kr	Korea Mobile Identity System	Nl
kscirc	KSChain Blockchain	?
lac	LACChain Network	Pd Pb
life	RChain	Pl Pb
lit	LEDGIS	Pl Pb
meme	IPFS & DNS & HTTP	Nl
mesh	Trusted Digital Web	La
meta	Metadium	Pl Pb
moac	MOAC	Pd Pr
monid	Ethereum	Pl Pb
morpheus	Hydra	Pl Pb
mydata	iGrant.io	Pd Pr
near	NEAR	Pl Pb
next	Nextme DIDs Network	Nl
nft	Ceramic Network	Pl Pb
nuggets	Nuggets Network	Nl
nuts	Nuts network	Nl
object	Trusted Digital Web	La
ockam	Ockam	Nl
omn	OmniOne	Pd Pr
onion	Ledger agnostic	La
ont	Ontology	Pl Pb
op	Ocean Protocol	Pl Pb
orb	Ledger agnostic	La
oyd	Ledger agnostic	La
panacea	Panacea	Pd Pr

peaq	peaq Blockchain	Pl Pb
peer	peer	La
pid	ProofID Blockchain	La
pistis	Ethereum	Pl Pb
pkh	Ledger-independent generative DID method based on CAIP-10 keypair expressions	La
pml	PML Chain	La
polygon	Polygon (Previously MATIC)	Pl Pb
polygonid	EVM compatible chains. Primary on Polygon	Pl Pb
prism	The Cardano blockchain	Pl Pb
psi	LEDGIS	Pl Pb
psqr	Public Square	Pd Pb
ptn	PalletOne	La
qes	QES	Nl
qui	Qui	Nl
ray	Ethereum	Pl Pb
real	Ethereum	Pl Pb
rm	Real-world Asset Tokenization DID Method	La
safe	Gnosis Safe	Pl Pb
san	SAN Cloudchain	La
schema	Multiple storage networks, currently public IPFS and evan.network IPFS	Nl
scid	StraitsChain	?
self	Ledger agnostic	La
selfkey	Ethereum	Pl Pb
sideos	Ledger agnostic	La
signor	Ethereum, Hedera Hashgraph, Quorum, Hyperledger Besu	Pl Pb/ Pd Pr
sirius	ProximaX Sirius Chain	Pd Pb
snail	Penpal network	?
snplab	SNPLab MyD Network	La
sol	Solana	Pl Pb
sov	Sovrin	Pd Pb
ssb	Secure Scuttlebutt	Nl
ssw	Initial Network	Pl Pb/ Pd Pr
stack	Bitcoin	Pl Pb
tangle	IOTA Tangle	Pd Pr
tdid	FISCO BCOS	Pl Pb/ Pd Pr
ti	TiChain	Pl Pb
tls	Ethereum	Pl Pb
trust	TrustChain	Pd Pr
trustbloc	Hyperledger Fabric	Pd Pr
trx	TRON	Pl Pb
ttm	TMChain	Pl Pb
twit	Twit	Nl
tyron	Zilliqa	Pl Pb
tys	DID Specification	?
tz	Tezos	Pl Pb
unik	uns.network	Pl Pb
unisot	Bitcoin SV	Pl Pb
uns	uns.network	Pl Pb
uport	Ethereum	Pl Pb
v1	Veres One DLT	Pd Pr
vaa	bif	Pd Pb
vaultie	Ethereum	Pl Pb
vertu	VERTU	Pl Pb
vid	VP	Nl
vivid	NEO2, NEO3, Zilliqa	Pl Pb
vtid	NaN	?
vvo	Vivvo	Pd Pr
web	Web	Nl
wlk	Weelink Network	Pl Pb
work	Hyperledger Fabric	Pd Pr
zk	Arweave	Pl Pb

Method	State	Created	Update
did:ala	app https://alastria.io/en/id-alastria/	2020	2022
did:bba	draft https://wubco.blobaa.dev/	2020	2022
did:dns	draft https://danubetech.github.io/did-method-dns/	2020	2022
did:key	draft https://w3c-ccg.github.io/did-method-key/	2020	2022
did:orb	draft https://trustbloc.github.io/did-method-orb/	2021	2022
did:panacea	draft https://github.com/medibloc/panacea-core	2020	2022
did:sov	draft https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html	2019	2022
did:ssb	draft https://viewer.scuttlebot.io/web/%25B79jjfhYHrr%2BCPSDpr2S5vRGWUJ5vwyfzbYCp0OyV9k%3D.sha256	2022	2022
did:trust	draft https://github.com/trustcerts/did-trust-method	2021	2022

Table 3
DID methods which take advantage of non-public blockchains, updated in 2022.



(a) Of all DID methods, the 28.93% of them uses public blockchains.

(b) The public blockchains used in DIDs are mainly based on Bitcoin (10.34%) and Ethereum (51.72%).

Figure 2: Partitioning of methods between public and non-public ledgers and the most used public blockchains.

5. Related Work

In this section, we present and describe some of the works in the related literature that survey blockchain-based approaches to SSI and applications concerning SSI and blockchain-based systems.

In [11] the authors provide an overview of IAM solutions based on their basic components, including identity management, authentication, and access control; related to the identity concept, they discuss self-sovereign identity. A taxonomy based on their features is then proposed to categorize these proposals. Finally, the existing methods are compared by using the proposed taxonomy.

The work in [3] provides an overview of the Self-Sovereign Identity (SSI) concept, focusing on four different components that are considered essential to the architecture. Self-Sovereign Identity is enabled by the new development of blockchain technology. The authors give a simple overview of blockchain-based SSI, introducing an architecture overview as well as relevant actors in such a system. Then they discuss identifiers in such systems by presenting some related approaches in SSI. Most central to the concept of an SSI is the verifiable claims that are presented to relying parties.

[12] provides a review of existing blockchain-based identity management papers and patents published between May 2017 and January 2020, which allow the user to take over control of his/her own identity. The work in [13] examines the properties that a self-sovereign identity should have and explores the impact of SSI on the laws of identity. It also describes the essential life cycles of an identity management system and inter-relates how the notion of SSI can be applied in these life cycles.

Method	Network	State	Created	Update
did:bnb	Binance Smart Chain	draft https://github.com/ontology-tech/DID-method-specs/blob/master/did-bnb/DID-Method-bnb.md	2020	2020
did:btcr	Bitcoin	draft https://github.com/dcdpr/btcr-DID-method	2018	2021
did:dual	Ethereum	draft https://github.com/Smart-ID-Card/Dual-DID/blob/main/docs/dual-did-method.md	2020	2021
did:ens	Ethereum	draft https://github.com/veramolabs/did-ens-spec	2021	2021
did:eosio	Eosio	app https://www.gimly.io/eosio-identity	2021	2021
did:erc725	Ethereum	discontinued	2018	2018
did:etho	Ethereum	draft https://github.com/ontology-tech/DID-method-specs/blob/master/did-etho/DID-Method-etho.md	2020	2020
did:ethr	Ethereum	draft https://github.com/decentralized-identity/ethr-did-resolver/blob/master/doc/did-method-spec.md	2019	2021
did:gatc	Ethereum and others	draft https://github.com/gataca-io/gataca-did-method	2020	2020
did:ion	Bitcoin	app https://identity.foundation/ion/	2020	**2022**
did:ipid	IPFS	draft https://did-ipid.github.io/ipid-did-method/	2018	2019
did:jolo	Ethereum	app https://jolocom.io/	2020	2021
did:monid	Ethereum	app https://www.monid.online/	2020	2021
did:pistis	Ethereum	draft https://github.com/uino95/ssi/blob/consensys/dashboard/server/pistis/pistis-did-resolver/README.md	2019	2020
did:pkh	Ethereum	app https://spruceid.dev/docs/didkit/	2021	2021
did:polygon	Polygon, Ethereum	draft https://github.com/ayanworks/polygon-did-method-spec	2021	2021
did:scheme	IPFS	draft https://github.com/51nodes/schema-registry-did-method/blob/master/README.md	2020	2020
did:selfkey	Ethereum	app https://selfkey.org/download/	2019	2019
did:signor	Ethereum e altro	? https://github.com/cryptonicsconsulting/signor-did-contracts/blob/master/did-method-spec.md	2020	2021
did:sol	Solana	app https://www.identity.com/	2021	**2022**
did:stack	Bitcoin	draft https://github.com/stacks-network/stacks-blockchain/blob/stacks-1.0/docs/blockstack-did-spec.md	2021	2021
did:tangle	IOTA Tangle	app https://tangleid.github.io/#/	2019	**2022**
did:tls	Ethereum	draft https://github.com/digitalcredentials/tls-did/blob/master/doc/did-method-spec.md	2020	2021
did:trx	Tron	draft https://github.com/ontology-tech/DID-method-specs/blob/master/did-trx/DID-Method-trx.md	2020	2020
did:tyron	Zilliqa	app https://www.ssiprotocol.com/#/	?	?
did:tz	Tezos	draft https://did-tezos.spruceid.com/	2020	**2022**
did:unisot	Bitcoin SV	draft https://gitlab.com/unisot-did/unisot-did-method-specification	2021	2021
did:vaultie	Ethereum	app https://vaultie.io/about/	2019	2020
did:vivid	Zilliqa, Neo	app https://moonlight.io/	2021	2021

Table 4
DID methods on public blockchain systems. We highlight in bold the methods updated in 2022.

Moreover, the paper illustrates several possible information flows involving a self-sovereign identity leveraging blockchain technology covering different aspects of an identity management system.

The paper in [14] presents a blockchain-based digital identity solution. Without depending upon a single trusted third party; the proposed framework achieves passport-level legally valid identity. This solution for making identities Self-Sovereign builds on a generic provable claim model for which attestations of truth from third parties need to be collected. Four different implementations are shown to offer sub-second performance for claim creation and claim verification. In [15] the authors present the Sora identity system, which is a mobile app that takes advantage of blockchain technology to create a secure protocol for storing encrypted personal information, as well as sharing verifiable claims about personal information.

The work in [16] provides a criteria-driven survey of the solutions and technologies for identity-managing blockchain-based systems and architectures in the context of verified claims and self-sovereign identities. the authors consider an extensive set of requirements covering ecosystem aspects, end-user functionality, mobility and overhead aspects, compliance/liability, EU regulations, standardization, and integration.

In [17] the state-of-the-art in Blockchain (BC)-based self-sovereignty and patient data records in healthcare is reviewed, by also proposing to provide an analysis of the design trade-offs. The motivation is to investigate the potential of blockchain technology for use in patient data and identity management. As a distributed decentralized technology, blockchains can be very beneficial, giving patients control over their own data and self-sovereign identity. The work in [18] first validates nine properties of self-sovereignty proposed by credible sources, then it proposes five new ones, and finally, it reasons about and validates these properties by proposing an architecture to enforce them. In [19] the authors implement a PoC of a decentralized OpenID Connect Provider by matching it with SSI, in order to give users the freedom to choose from a large pool of identity providers instead of just a select few corporations.

6. Conclusion

With the rapid growth of digital ecosystems, individuals are increasingly sharing large amounts of personal data through low-security digital interactions, sacrificing their privacy and security. Digital ID is a way of proving who we are and generating opportunities to carry out interactions in a simple and safe way in the digital world. Efficiency, cost reduction, fraud prevention, and less bureaucracy are some of the benefits of using a secure digital identity.

Blockchains (public from our point of view) can support the transparent use of DID for real-world applications. However, using this approach to IdM is still challenging: Many IdM features, including identity recovery, lookup services, backup of cryptographic keys, etc., may be jeopardized by the complete removal of the central authority. As a result of their familiarity with the various warnings, it has been demonstrated in practice that user agreement frequently results in the disclosure of the maximum information. Moreover, in order to provide pseudonymity while retaining the necessary levels of secrecy, integrity, authenticity, non-repudiation, and robustness, a user-controlled identity requires a transparent flow of data.

In this paper, we have collected and reviewed 107 different approaches to represent a DID, all following W3C standards and reference documents [6, 8, 10]. The goal is to understand where possible real-world applications based on SSI are directed to, and the general adoption of this approach.

Acknowledgments

S. Bistarelli and F. Santini are members of INdAM GNCS and Consorzio CINI. This work has been partially supported by: GNCS-INdAM, CUP E55F22000270001; Project FICO, Ricerca di Base 2021, University of Perugia; Project BLOCKCHAIN4FOODCHAIN, Ricerca di Base 2020, University of Perugia; Project GIUSTIZIA AGILE, CUP J89J22000900005.

References

[1] P. A. Grassi, J. L. Fenton, M. E. Garcia, Digital identity guidelines [including updates as of 12-01-2017], 2017. doi:https://doi.org/10.6028/NIST.SP.800-63-3.

[2] N. Naik, P. Jenkins, uport open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain, in: 2020 IEEE International Symposium on Systems Engineering (ISSE), IEEE, 2020, pp. 1–7.

[3] A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel, A survey on essential components of a self-sovereign identity, Comput. Sci. Rev. 30 (2018) 80–86.

[4] A. Tobin, D. Reed, The inevitable rise of self-sovereign identity, The Sovrin Foundation 29 (2016) 18.

[5] C. Allen, The path to self-sovereign identity, 2016. URL: https://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[6] W3C, Decentralized identifiers (dids) v1.0, 2021. URL: https://www.w3.org/TR/did-core/.

[7] D. I. F. (DIF), Universal resolver, 2022. URL: https://dev.uniresolver.io/.

[8] W3C, A primer for decentralized identifiers, 2021. URL: https://w3c-ccg.github.io/did-primer/.

[9] TranSendX, did:ipid method specification, 2018. URL: https://did-ipid.github.io/ipid-did-method/.

[10] W3C, List of did methods, 2023. URL: https://www.w3.org/TR/did-spec-registries/#did-methods.

[11] F. Ghaffari, K. Gilani, E. Bertin, N. Crespi, Identity and access management using distributed ledger technology: A survey, International Journal of Network Management 32 (2021). doi:10.1002/nem.2180.

[12] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, K.-K. Raymond Choo, Blockchain-based identity management systems: A review, Journal of Network and Computer Applications 166 (2020) 102731. doi:https://doi.org/10.1016/j.jnca.2020.102731.

[13] M. S. Ferdous, F. Chowdhury, M. O. Alassafi, In search of self-sovereign identity leveraging blockchain technology, IEEE Access 7 (2019) 103059–103079. doi:10.1109/ACCESS.2019.2931173.

[14] Q. Stokkink, J. Pouwelse, Deployment of a blockchain-based self-sovereign identity, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 1336–1342. doi:10.1109/Cybermatics_2018.2018.00230.

[15] M. Takemiya, B. Vanieiev, Sora identity: Secure, digital identity on the blockchain, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), volume 02, 2018, pp. 582–587. doi:10.1109/COMPSAC.2018.10299.

[16] M. Kuperberg, Blockchain-based identity management: A survey from the enterprise and ecosystem perspective, IEEE Transactions on Engineering Management 67 (2020) 1008–1027. doi:10.1109/TEM.2019.2926471.

[17] B. Houtan, A. S. Hafid, D. Makrakis, A survey on blockchain-based self-sovereign patient identity in healthcare, IEEE Access 8 (2020) 90478–90494. doi:10.1109/ACCESS.2020.2994090.

[18] K. C. Toth, A. Anderson-Priddy, Self-sovereign digital identity: A paradigm shift for identity, IEEE Security & Privacy 17 (2019) 17–27. doi:10.1109/MSEC.2018.2888782.

[19] Z. A. Lux, D. Thatmann, S. Zickau, F. Beierle, Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials, in: 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2020, pp. 71–78. doi:10.1109/BRAINS49436.2020.9223292.