

# A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems

Loïc Lesavre  
Priam Varin  
Peter Mell  
Michael Davidson  
James Shook  
*Computer Security Division  
Information Technology Laboratory*

This publication is available free of charge from:  
<https://doi.org/10.6028/NIST.CSWP.07092019-draft>

July 9, 2019

23

## Abstract

24 Identity management systems (IDMSs) are widely used to provision user identities while  
25 managing authentication, authorization, and data sharing both within organizations as well as on  
26 the Internet more broadly. Traditional identity systems typically suffer from single points of  
27 failure, lack of interoperability, and privacy issues such as encouraging mass data collection and  
28 user tracking. Blockchain technology has the potential to support novel data ownership and  
29 governance models with built-in control and consent mechanisms, which may benefit both users  
30 and businesses by alleviating these concerns; as a result, blockchain-based IDMSs are beginning  
31 to proliferate. This work categorizes these systems into a taxonomy based on differences in  
32 architecture, governance models, and other salient features. We provide context for the taxonomy  
33 by describing related terms, emerging standards, and use cases, while highlighting relevant  
34 security and privacy considerations.

35

## Keywords

36 blockchain; credential; data ownership; decentralized identifier; distributed ledger; identity  
37 management; public key infrastructure; self-sovereign identity; smart contract; user-controlled  
38 identity wallet; zero-knowledge proof.

39

## Disclaimer

40 Any mention of commercial products or reference to commercial organizations is for information  
41 only; it does not imply recommendation or endorsement by the National Institute of Standards and  
42 Technology (NIST), nor does it imply that the products mentioned are necessarily the best  
43 available for the purpose.

44

## Additional Information

45 For additional information on NIST's Cybersecurity programs, projects and publications, visit the  
46 [Computer Security Resource Center](#). Information on other efforts at [NIST](#) and in the [Information  
47 Technology Laboratory](#) (ITL) is also available.

48

49

## Public Comment Period: *July 9, 2019 through August 9, 2019*

50

National Institute of Standards and Technology

51

Attn: Computer Security Division, Information Technology Laboratory

52

100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

53

Email: [blockchain-idms-paper@nist.gov](mailto:blockchain-idms-paper@nist.gov)

54

All comments are subject to release under the Freedom of Information Act (FOIA).

55

56

## **Acknowledgements**

57 The authors wish to thank all contributors to this publication and their colleagues who reviewed  
58 drafts of this report, contributed technical and editorial additions, and supported this work. This  
59 includes NIST staff and associates Dylan Yaga, Frederic De Vault, Angela Robinson, and Katya  
60 Delak.

61

## **Audience**

62 This publication is designed for readers with some knowledge of blockchain technology who wish  
63 to understand at a high level how blockchain identity management systems work. It is not intended  
64 to be a technical guide; the discussion of the technology provides a conceptual understanding. Note  
65 that some examples, figures, and tables are simplified to fit the audience.

## 66 **Executive Summary**

67 Identity management systems allow one to provision identities to users, while managing  
68 authentication, authorization, and data sharing both within organizations as well as on the Internet.  
69 With traditional identity management, organizations usually store the credentials (e.g., a password)  
70 of each user they interact with, or with federated models, they use a third party to store this  
71 information. This creates interoperability, security, and privacy concerns due to the privileged  
72 position of the entity that controls the identity information.

73 A possible solution to these issues is found in the use of blockchain technologies for identity  
74 management: they can reduce, or even remove, the need for a third party. At a high-level, they can  
75 transform data governance and ownership models, and benefit both individual users and  
76 businesses. More specifically, it can enable users to control their data and share select personal  
77 information to relying parties. It can also enable businesses to streamline their operations by  
78 relying on verified user information without having to maintain the infrastructure themselves.

79 A large number of blockchain-based identity management approaches are currently being  
80 explored, implemented, and commercialized. Many of them use, or plan to use, smart contracts,  
81 the privacy capabilities gained from zero-knowledge protocols, and other scalability solutions atop  
82 the underlying blockchain. This is an emerging field and the features, capabilities, security, and  
83 privacy of these proposed systems are often unclear.

84 Identity is a far-reaching topic, and the systems being designed to support it can take architectural  
85 forms that are both on-chain and off-chain, and follow types of governance models spanning from  
86 top-down approaches to “self-sovereign” bottom-up ones. Each system has different control and  
87 delegation capabilities, as well as scalability constraints.

88 This work breaks down identifier and credential architectures, discusses their reliance to  
89 blockchains, and possible combination patterns. It looks at the levels at which on-chain registries  
90 are created, and who can control them. It investigates “bring-your-own” blockchain address  
91 schemes, along with credentials being issued as off-chain objects. It does not attempt to judge  
92 between the different architectures and models, but instead, highlights their differences.

93 We first offer a terminology for blockchain identity management as well as a list of associated  
94 standards and building blocks. We then provide a breakdown of distinguishing properties and  
95 architectures. Next, we discuss public registries, and then, system governance. Finally, we cover  
96 some of the security concerns that might affect these systems, along with additional considerations  
97 on core blockchain protocols, zero-knowledge proofs, presentation sharing and data mining, as  
98 well as ecosystem convergence. To make this discussion less abstract, we offer two use cases.

99 This will help the reader navigate how blockchain-based identity management systems work and  
100 what they can offer. It may be useful for the reader to better understand and build identity  
101 management systems, and can contribute towards designing efficient architectures. It will also  
102 enable the reader to be aware of what is possible, and thus, better able to distinguish between the  
103 many emerging systems.

104

## Table of Contents

105	<b>Executive Summary .....</b>	<b>iv</b>
106	<b>1 Introduction .....</b>	<b>1</b>
107	1.1 Background.....	1
108	1.2 Purpose and Scope .....	3
109	1.3 Disclaimers and Clarifications .....	3
110	1.4 Blockchain Identity Management Initiatives and Guidance .....	3
111	1.5 Document Structure .....	4
112	<b>2 Blockchains and Smart Contracts.....</b>	<b>5</b>
113	<b>3 Fundamentals of Blockchain Identity Management.....</b>	<b>6</b>
114	3.1 Terminology .....	6
115	3.2 Blockchain-based Identity Management Roles and Object Relationships .....	7
116	3.3 Emerging Standards .....	8
117	3.4 Building Blocks .....	11
118	3.5 Blockchain Identity Management Stack .....	14
119	<b>4 Blockchain Identity Management System Taxonomy.....</b>	<b>15</b>
120	4.1 Authority Model.....	15
121	4.1.1 Top-down vs Bottom-up Organizational Structures .....	15
122	4.1.2 Identifier Origination Schemes .....	15
123	4.1.3 Credential Issuance Schemes .....	16
124	4.2 Identifiers and Credentials Management .....	17
125	4.2.1 Lifecycle .....	17
126	4.2.2 Custody and Delegation .....	17
127	4.3 Presentation Disclosure .....	19
128	4.4 System Architecture Designs.....	20
129	4.4.1 Identifier Architectures.....	20
130	4.4.2 Credential Architectures .....	22
131	4.4.3 Combination Patterns .....	25
132	4.5 Public Registries and Reputation Management Implications .....	26
133	4.6 System Governance .....	27
134	<b>5 Security and Risk Management .....</b>	<b>31</b>

135 **6 Additional Considerations ..... 34**  
136 6.1 Underlying Blockchain Implications..... 34  
137 6.2 Introduction to Zero-Knowledge Protocols ..... 35  
138 6.3 Presentation Sharing and Data Mining ..... 37  
139 6.4 Ecosystem Convergence ..... 38  
140 **7 Use Cases ..... 40**  
141 **8 Conclusion ..... 42**

142  
143

**List of Appendices**

144 **References ..... 43**  
145 **Appendix A— Acronyms ..... 48**  
146 **Appendix B— Glossary ..... 50**

147

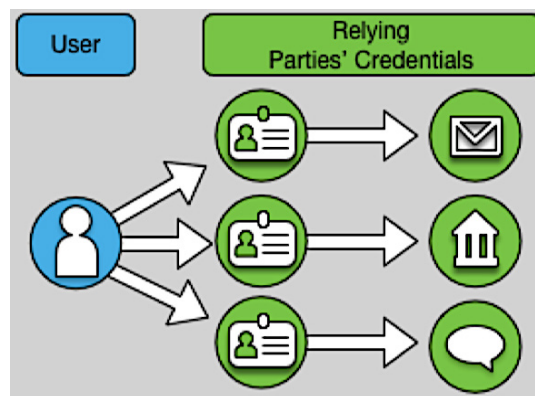
## 148 **1 Introduction**

149 A large number of blockchain-based identity management approaches are being explored,  
150 implemented, and commercialized. This is a new field, and the features, security, and privacy of  
151 these proposed systems are often unclear. While many of the approaches hold great promise, most  
152 projects rely on the prerequisite of using a blockchain platform that is reliable, secure, scalable,  
153 and, sometimes, publicly accessible. Thus, blockchain-based identity management systems are an  
154 emerging area that should be watched and carefully evaluated as a potential, but not guaranteed,  
155 breakthrough for digital identity and data ownership.

### 156 **1.1 Background**

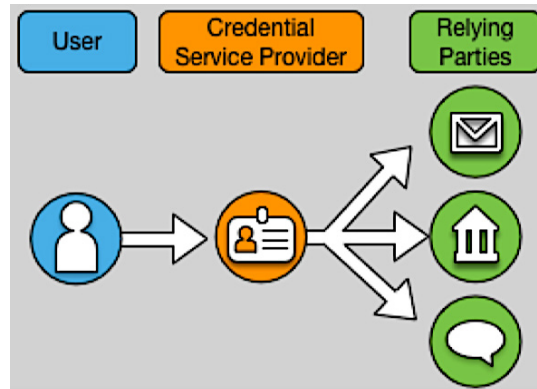
157 Identity management systems (IDMSs) are a foundational infrastructure for interactions between  
158 entities (individuals, organizations, or things) to support commerce, education, health care,  
159 government services, and many other aspects of society. An IDMS must allow entities to  
160 authenticate while at the same time distributing information about those entities to enable the  
161 granting of access privileges of differing levels or types.

162 With traditional identity management, businesses store credentials about each user with which they  
163 interact (e.g., a password). This enables a user to directly authenticate to the business (or more  
164 technically “relying party”) with which they need to interact, as shown in Figure 1. However, the  
165 user is burdened with needing to separately authenticate to each business using different  
166 credentials. In addition, businesses are not able to automatically obtain and evaluate verified  
167 identifying information about each user.



168 **Figure 1: Traditional Identity Management (copied from [1])**

169 More recently, federated identity management systems [1] enable credential service providers to  
170 maintain user credentials on behalf of various relying parties. This enables single-sign-on  
171 capabilities where a user utilizes a single set of credentials to access a large number of services, as  
172 shown in Figure 2. This frees up the user from having to maintain many passwords. However, it  
173 creates security and privacy concerns given the privileged position of the credential service  
174 provider between the user and relying parties. For example, it presents a single point of failure that  
175 could inhibit the user's access to the relying parties or, even worse, could enable the credential  
176 service provider to masquerade as a user.



177 **Figure 2: Federated Identity Management (copied from [1])**

178 A possible solution to these issues may be found in the use of blockchain technologies for identity  
179 management; blockchains can remove the need for traditional credential service providers and  
180 enable direct user to relying party interactions with verified information.

181 From the subject's perspective, blockchain-based IDMSs allow them to own their data, or control  
182 who owns it, while being able to share verified information with relying parties to facilitate a  
183 certain set of actions (e.g., business transactions). This architecture may enable the subjects to be  
184 in a better position to give their consent. For example, a subject can cryptographically approve a  
185 transaction prior to some relying party executing it on their behalf (e.g., a bank could not open an  
186 account for a user without their attested prior approval).

187 From the relying party's perspective, blockchain-based IDMSs allow them to verify that some user  
188 information needed to initiate a transaction is valid without having to store the personal  
189 information themselves. A key implication is that it lowers privacy and security burdens, and may  
190 facilitate bootstrapping new business activities as well as automating existing ones.

191 In summary, blockchain-based IDMSs have the potential to greatly enhance security and privacy,  
192 and grant built-in control and consent capabilities for both users and relying parties. However,  
193 there are tradeoffs to be made and it will be necessary to carefully evaluate the emerging solutions.



## 194 **1.2 Purpose and Scope**

195 This document provides an introduction to the different blockchain identity management  
196 approaches currently being explored and implemented. The purpose is not to review each solution  
197 individually, but to provide a taxonomic approach that will give the reader different viewpoints  
198 and methods by which blockchain-based identity management can be designed. In this way the  
199 document highlights the different features and characteristics that are possible while exploring the  
200 opportunities, challenges, and risks they represent.

201 As an emerging field, weaknesses may become evident that negate the apparent advantages or  
202 other data models may emerge with even greater benefit. It may take years for the proper  
203 blockchain infrastructure, the identity management platforms, and related user tools to mature and  
204 be deployed at scale. While the time for most readers to deploy these capabilities lies somewhere  
205 in the future, we argue that the capabilities and architecture designs discussed in this paper  
206 represent a major improvement over traditional identity management systems and thus, that this  
207 field deserves careful consideration and scrutiny today. We hope that this paper provides the  
208 foundational tools to enable such an ongoing evaluation.

## 209 **1.3 Disclaimers and Clarifications**

210 We will be referring to “blockchains” throughout this paper. However, this work may be extended  
211 to any kind of Distributed Ledger Technology (DLT). This paper refers to blockchain, smart  
212 contract capabilities, and related concepts without recommending or endorsing any particular  
213 protocols. Any products or protocols mentioned are for explanatory purposes only and do not  
214 imply any endorsement or suitability for use.

## 215 **1.4 Blockchain Identity Management Initiatives and Guidance**

216 Some organizations have already written guidance on blockchain in identity management. The  
217 European Union recently published *Blockchain for Government and Public Services* [2] and  
218 *Blockchain and Digital Identity* [3]. In the United States, there have been initiatives led at the state  
219 level such as the Illinois Blockchain Initiative [4]. The American Council for Technology and  
220 Industry Advisory Council (ACT-IAC) published a *Blockchain Primer* [5] to introduce how  
221 blockchain could impact the Federal Government as well as a *Blockchain Playbook* [6] to introduce  
222 how it could be applied to the U.S. Federal Government for different purposes, including identity  
223 management.

224 There are a handful of blockchain-based IDMS pilots as well. Some organizations have already  
225 adopted the use case of diploma and certificate issuance on the blockchain, such as the  
226 Massachusetts Institute of Technology with Blockcerts and Learning Machine [7]. Some  
227 jurisdictions are experimenting with blockchain-based IDMSs at different levels, such as Estonia  
228 (for electronic medical records) [8], the City of Zug in Switzerland using uPort (on the Ethereum  
229 blockchain) [9], and the Provinces of British Columbia and Ontario in Canada using the Verifiable  
230 Organizations Network (on the Sovrin blockchain) [10]. Note that many projects are currently  
231 under active development, characterizing the growing interest in blockchain-based identity  
232 management.

233 **1.5 Document Structure**

234 The rest of this document is organized as follows:

- 235 • **Section 2** introduces blockchain technology and smart contracts at a high-level.
- 236 • **Section 3** defines terminology and discusses emerging standards and building blocks for  
237 blockchain identity management.
- 238 • **Section 4** introduces and discusses a taxonomy in the form of distinguishing properties,  
239 which are then used to characterize different architecture designs.
- 240 • **Section 5** introduces some of the security concerns and their mitigation mechanisms.
- 241 • **Section 6** provides additional considerations.
- 242 • **Section 7** introduces potential use cases.
- 243 • **Section 8** is the conclusion.
- 244 • **References** lists the references used throughout the document.
- 245 • **Appendix A** provides a list of acronyms and abbreviations used in the document.
- 246 • **Appendix B** contains a glossary for selected terms defined in the document.
- 247

## 248 **2 Blockchains and Smart Contracts**

249 We invite the readers, who have little or no knowledge of blockchain technology and who wish to  
250 understand at a high level how it works, to read *Blockchain Technology Overview* NISTIR 8202  
251 [11]. It defines blockchain as “tamper evident and tamper resistant digital ledgers implemented in  
252 a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e.,  
253 a bank, company or government). At their basic level, they enable a community of users to record  
254 transactions in a shared ledger within that community, such that under normal operation of the  
255 blockchain network no transaction can be changed once published.”

256 The technology is called blockchain because the transactions are grouped and published separately  
257 in distinct data structures called blocks, which are cryptographically linked together, duplicated,  
258 and distributed in a peer-to-peer network to prevent tampering of previously published  
259 transactions. Blockchain accounts are based on asymmetric-key cryptography and allow  
260 participants to sign transactions. The transactions are added to blocks that must be validated by the  
261 nodes that are running the blockchain’s peer-to-peer node client. Consensus models determine  
262 which node gets the privilege of publishing the next block (see Section 4.6 on *Consensus*  
263 *Comparison Matrix* of NISTIR 8202).

264 The paper discusses two important categories that pertain to our investigation of identity  
265 management systems: “Blockchain networks can be categorized based on their permission model,  
266 which determines who can maintain them (e.g., publish blocks). If anyone can publish a new block,  
267 it is *permissionless*. If only particular users can publish blocks, it is *permissioned*. In simple terms,  
268 a permissioned blockchain network is like a corporate intranet that is controlled, while a  
269 permissionless blockchain network is like the public internet, where anyone can participate.  
270 Permissioned blockchain networks are often deployed for a group of organizations and individuals,  
271 typically referred to as a consortium.”

272 Some blockchains have highly expressive native smart contract capabilities which are often useful  
273 for blockchain identity management solutions. A smart contract is defined as: “a collection of code  
274 and data (sometimes referred to as functions and states) that is deployed using cryptographically  
275 signed transactions on the blockchain network (e.g., Ethereum’s smart contracts, Hyperledger  
276 Fabric’s chaincode). The smart contract is executed by nodes within the blockchain network; all  
277 nodes that execute the smart contract must derive the same results from the execution, and the  
278 results of execution are recorded on the blockchain. [...] The smart contract code can represent a  
279 multi-party transaction, typically in the context of a business process. In a multi-party scenario,  
280 the benefit is that this can provide attestable data and transparency that can foster trust, provide  
281 insight that can enable better business decisions, reduce costs from reconciliation that exists in  
282 traditional business to business applications, and reduce the time to complete a transaction. [...] Smart contracts must be deterministic, in that given an input they will always produce the same  
283 output based on that input.” Furthermore, a source of off-chain data that serves as input for a smart  
284 contract is referred to as an “oracle”.  
285

286 Note that the owner of a blockchain identity management system does not necessarily own the  
287 blockchain upon which this system is built. In fact, an entity can deploy an identity management  
288 system without having to build or maintain the underlying blockchain infrastructure that is being  
289 leveraged.

## 290 **3 Fundamentals of Blockchain Identity Management**

291 Prior to us introducing our taxonomy in the next section, this section details key terminology,  
292 common roles and objects, emerging supportive standards, essential building blocks, and a  
293 blockchain identity management communication stack. These terms, standards, and abstractions  
294 are used by most blockchain identity management systems.

### 295 **3.1 Terminology**

296 Specialized terminology is used for blockchain-based identity management schemes.  
297 Unfortunately, the terminology is not always consistent among the various projects and standards.  
298 Further complicating matters is that some domain-specific terms are related to identity  
299 management in general while others are specific to blockchain identity management.  
300 Understanding the following terms is necessary in order to understand the concepts discussed in  
301 this paper.

302 **Claim:** A characteristic or statement about a *subject* made by an *issuer* as part of a *credential*.

303 **Credential:** A set of one or more *claims* made by an *issuer*. A *credential* is associated with an  
304 *identifier*.

305 **Custodian:** An *entity* acting on behalf of another *entity* with respect to their identifiers and/or  
306 credentials.

307 **Entity:** A person, organization, or thing.

308 **Holder:** A *custodian* holding a *credential* on behalf of a *subject*.

309 **Identifier:** A blockchain address or other pseudonym that is associated to an *entity*.

310 **Issuer:** An *entity* that issues a *credential* about a *subject* on behalf of a *requester* and owns one or  
311 more *identifiers*.

312 **Presentation:** Information derived from one or more *credentials* that a *subject* discloses to a  
313 *verifier* (working on behalf of some *relying party*) to communicate some quality about a *subject*.

314 **Relying Party:** An *entity* that receives information about a subject from a *verifier*.

315 **Requester:** An *entity* that makes a request to an *issuer* to issue a *credential* about a *subject*.

316 **Subject:** An *entity* that acts as a regular participant in a given identity management system and  
317 owns one or more *identifiers*.

318 **System Owner:** An *entity* that owns a given identity management system.

319 **Verifier:** An *entity* that verifies the validity of a *presentation* on behalf of a *relying party*.

### 3.2 Blockchain-based Identity Management Roles and Object Relationships

With this terminology we can identify the common roles that occur in blockchain-based IDMSs and the relationships between these roles. We can also identify common objects found in these systems and the relationships between those objects.

Figure 3 provides a high-level overview of the identity management roles defined in Section 3.1.

- *Requesters, Issuers, and Subjects* are involved in credential issuance.
- *Subjects, Verifiers, and Relying Parties* are involved in presentation disclosure.
- *Requesters* ask for the issuance of a credential from *Issuers*. *Issuers* provide credentials to *Subjects*.
- *Subjects* reveal presentations to *Verifiers*.
- *Verifiers* verify presentations on behalf of *Relying Parties*.

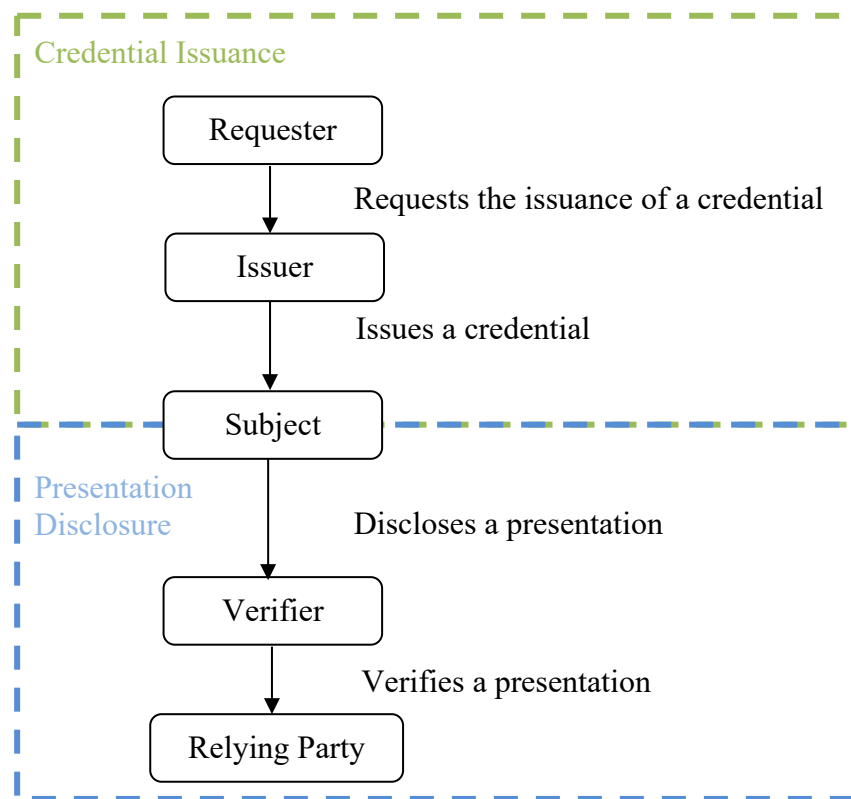


Figure 3: Identity Management Roles

Note that these roles are not exclusive. For instance, a subject and an issuer can both take the requester role or a subject and a verifier can both be a relying party. Depending on the IDMS, the approval of a subject may be required to issue a new credential to that subject.

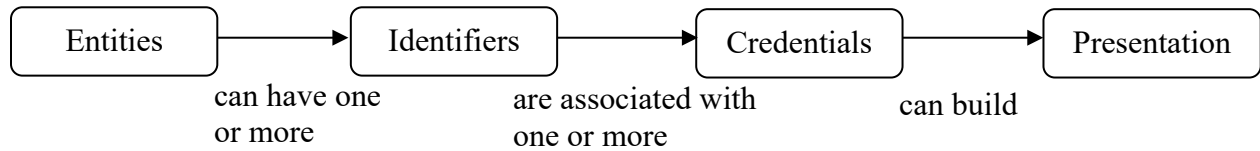


Figure 4: Hierarchy of IDMS Objects

334 Figure 4 provides a high-level overview of the objects that entities interact with in a blockchain  
335 IDMS. The figure shows that entities can have one or more identifiers, that identifiers are  
336 associated with one or more credentials, and that presentations are derived from credentials.

### 337 3.3 Emerging Standards

338 There is a set of emerging standards that support blockchain-based IDMSs including:

- 339 • Decentralized Identifiers and Verifiable Credentials, from the World Wide Web Consortia  
340 (W3C)
- 341 • Open Badges, from Mozilla and IMS Global
- 342 • Universal Resolver and Identity Hubs, from the Decentralized Identity Foundation (DIF)

343 In the subsequent sections of this paper, we will be using the terms identifiers, credentials, and  
344 presentations, but will not necessarily be referring to standards of this section.

345 In addition, we will refer to blockchain network specific standards such as Ethereum Request for  
346 Comments (ERCs) and Bitcoin Improvement Proposals (BIPs).

#### 347 Decentralized Identifiers – W3C:

348 Decentralized Identifiers (DIDs) [12] are identifiers whose purpose is to facilitate the creation of  
349 persistent encrypted private channels between entities without the need for any central registration  
350 mechanism. They can be used, for example, for credential exchanges and authentication. An entity  
351 can have multiple DIDs, even one or more per relationship with another entity (see *Pairwise-*  
352 *pseudonymous and Single-use Identifiers* in Section 4.3). When an entity has one DID per  
353 relationship with other entities, it is called a pairwise pseudonymous DID. Ownership of a DID is  
354 established by demonstrating possession of the private key associated with the public key bound  
355 to the DID.

356 A DID method is a public, standard set of schemes by which to create, resolve, update, and delete  
357 DIDs. These methods allow for DID registration, replacement, rotation, recovery, and expiration  
358 within an IDMS.

359 A DID has the following format:

360 “did:” + <did-method> + “:” + <method-specific-identifier>

361 As an example, a DID for a “NIST DID method” could look like: did:nist:0x1234abcd.

362 As part of a DID method, a DID resolver allows one to take a DID as input and to return the  
363 associated metadata, called a DID document and formatted as a JavaScript Object Notation for  
364 Linked Data (JSON-LD) object [13]. JSON-LD is a JSON-based format used to serialize linked  
365 data and build interoperable services. According to W3C’s primer [14], a DID document is  
366 comprised of the following standard elements:

- 367 • A DID that identifies the subject of the DID document
- 368 • A set of public keys used for authentication, authorization, and communication  
369 mechanisms
- 370 • A set of authentication methods used for the DID subject to prove ownership of the DID  
371 to another entity
- 372 • A set of authorization and delegation methods for allowing other entities to operate on  
373 behalf of the DID subject (i.e., custodians)
- 374 • A set of service endpoints to describe where and how to interact with the DID subject
- 375 • A Uniform Resource Identifier (URI) to uniquely identify terminology and protocols that  
376 allows parties to get a common understanding of the identifier
- 377 • A timestamp for auditing
- 378 • A signature for integrity

379 **Universal Resolver – DIF:**

380 While DID documents can be retrieved through using a DID resolver, there are advantages to  
381 having a more general resolver that can communicate with multiple decentralized identifier  
382 systems (including DID systems). The Universal Resolver [15] achieves this goal; it enables  
383 application code to be written to a single resolver interface that enables communication to multiple  
384 decentralized identifier systems. A DID-based blockchain IDMS that supports the Universal  
385 Resolver must define and implement a DID Driver that links the Universal Resolver to their  
386 system-specific DID Method for reading DID documents. This allows applications relying on the  
387 IDMS to query DIDs in a common interface so they do not have to deal with fetching the system-  
388 specific DID methods themselves. This takes place according to the steps shown in Figure 5.

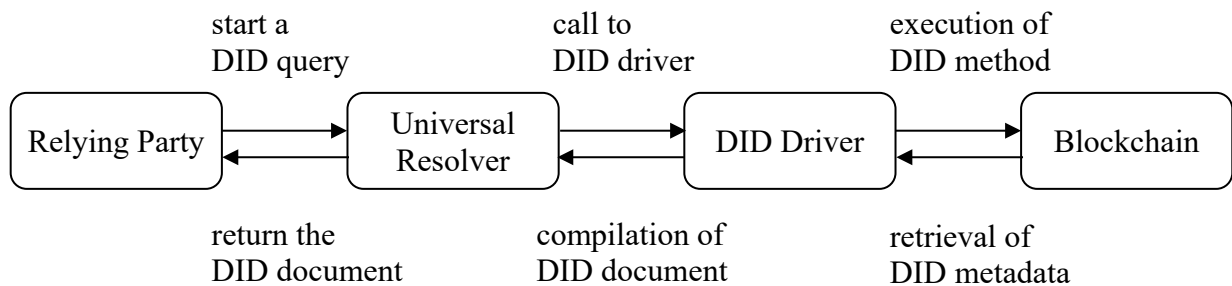


Figure 5: DID Lookup using the Universal Resolver



389 **Verifiable Credentials and Verifiable Presentations – W3C:**

390

391 The Verifiable Credentials specification [16] defines a format for credentials that can be exchanged  
392 between DIDs (using JSON-LD). A Verifiable Credential is a tamper-resistant credential that is  
393 cryptographically signed by its issuer.

394 A Verifiable Credential includes:

- 395 • DIDs for the subject and the issuer
- 396 • URI to uniquely identify the credential
- 397 • Claims data or metadata to access it
- 398 • URI to uniquely identify terminology and protocols that allows parties to get a common  
399 understanding of the identifier
- 400 • Expiration conditions
- 401 • Credential status (active, suspended, or revoked)
- 402 • Cryptographic signature of the issuer

403 The W3C specification also defines Verifiable Presentations. A Verifiable Presentation is a  
404 tamper-resistant presentation derived from a Verifiable Credential and cryptographically signed  
405 by the subject disclosing it.

406 A Verifiable Presentation includes:

- 407 • URI to uniquely identify contexts
- 408 • URI to identify the presentation
- 409 • One or more verifiable credentials, or data derived from them
- 410 • Cryptographic signature of the subject

411 **Open Badges – Mozilla and IMS Global:**

412 Open Badges [17] is another approach to credentials, which are referred to as badges. There are  
413 three core data classes used to instantiate a badge: Assertions, BadgeClasses, and Profiles. They  
414 have the following features:

- 415 • The “Assertion” class contains data about the entity that received the badge (the entity  
416 about which something is being asserted), the issuance timestamp, as well as instructions  
417 for verifying the information hosted in the badge. Additional properties can also be made  
418 available, such as a revocation status or an expiration date.
- 419 • The “BadgeClass” class adds context to the type of credential that is enclosed in the badge  
420 by listing its name and category, the criteria used to describe how to earn the credential, as  
421 well as a reference to the entity that issued the badge.
- 422 • The “Profile” class brings more information (e.g., name, email address, phone number,  
423 public keys) about the entities linked to the badge (e.g., the badge issuer, recipient, and  
424 endorser).

425 Just like DID documents and Verifiable Credentials, Badges take the form of JSON-LD documents  
426 that can be encoded into Quick Response (QR) codes, allowing easier integration into applications.



427 **Identity Hubs – DIF:**

428 Identity Hubs [18] are encrypted personal datastores connected together, using both edge devices  
429 (e.g., smartphones, personal computers) and cloud storage. They are used to securely store and  
430 share identity data when such sharing is approved by the owner.

431  
432 An Identity Hub is made of one or more Hub instances, which can run on a personal device or be  
433 hosted by a provider. Each Identity Hub is linked to a given DID and can be integrated with the  
434 Universal Resolver. The data attached to a DID is replicated and stored across a set of Hub  
435 instances. This architecture was designed to avoid single points of failure as well as to let a subject  
436 manage access permissions granularly.

437 **3.4 Building Blocks**

438 The building blocks of blockchain identity management systems vary, but at a high-level, they are  
439 commonly comprised of the following technical components:

440 **Blockchain:**

441 A blockchain can support the management of keys and identifiers by acting as a Decentralized  
442 Public Key Infrastructure (DPKI).<sup>1</sup> Note that the blockchain may be application-specific such as  
443 Hyperledger Indy [22] and/or may support a native smart contract platform. In most cases, the  
444 DPKI, sometimes augmented by separate protocols atop the blockchain, forms a decentralized  
445 identifier system (called DID method if it follows the DID specification). In addition to keys and  
446 identifiers, credentials may also rely on the blockchain.

447 **Second Layer Protocol:**

448 A decentralized identifier system may rely on both a blockchain and a separate protocol on top of  
449 it, often referred as “second layer” (off-chain) protocol. These protocols can be used to build  
450 scaling solutions by “off-loading” operations away from the blockchain layer. That way, smart  
451 contracts can be designed such that blockchain transactions (triggered by function calls) track a  
452 set of operations rather than a single one. For example, the SideTree protocol [23] (run by SideTree  
453 nodes that are separate from those of the underlying blockchain) allows one to bundle DID  
454 operations together before posting them onto a blockchain.<sup>2</sup>

---

<sup>1</sup> NIST Special Publication (SP) 800-32 [19] defines a Public Key Infrastructure (PKI) as follows: “[A PKI] binds public keys to entities, enables other entities to verify public key bindings, and provides the services needed for ongoing management of keys in a distributed system”. Note that the company Evernym was awarded a grant from the U.S. Department of Homeland Security in 2017 to develop a decentralized key management solution based on NIST SP 800-130, *A Framework for Designing Key Management Systems* [20]. This became the key management foundation of the Sovrin [21] IDMS; the Sovrin codebase was then added to the Hyperledger Foundation open-source projects under the name of Hyperledger Indy.

<sup>2</sup> The SideTree protocol (released as a DIF project) has been implemented to develop decentralized identifier systems (that follow the DID specification) by Microsoft on top of the Bitcoin protocol (the DID method is called Identity Overlay Networks [24] (ION)), and by Transmute Industries (with ConsenSys) on top of the Ethereum protocol (the DID method is called Element).

455 In addition to the scalability benefits, second layer protocols may have a different level of privacy  
456 than transactions in the underlying blockchain. Finally, second layer protocols do not function as  
457 standalone blockchains, rather they require one or more blockchains to operate. A key implication  
458 is that second layer protocols can help promote the development of interoperable, blockchain-  
459 agnostic systems by allowing for the integration of multiple blockchains without necessarily  
460 requiring any fundamental change to their codebase.

#### 461 **Smart Contracts:**

462 Blockchains may support smart contracts, which are vital to many blockchain-based IDMSs (some  
463 of them implementing all the logic in the form of smart contracts). The power of smart contracts  
464 is that they can act as a trusted third party given that the blockchain network guarantees the  
465 execution of their code. This enables blockchain-based IDMSs to use smart contracts to replace  
466 many functions formerly assumed by the traditional credential service provider in non-blockchain  
467 identity management solutions, and potentially increase trust in these systems. In particular, they  
468 are currently used to implement on-chain registries and governance structures.

#### 469 **Credential Storage Methods:**

470 A foundational architectural feature for blockchain IDMSs is the method (or methods) by which  
471 credentials are stored (see Section 4.4.2 on *Credential Architectures*). Some blockchain-based  
472 IDMSs allow for storage of credentials using a blockchain while others store the credentials off-  
473 chain. Off-chain credentials may be stored by a subject in a wallet application (explained in the  
474 *User-Controlled Identity Wallet* paragraph below) or by a third party custodian to whom the  
475 subject has delegated this role.

#### 476 **Data Exchange Models:**

477 To request, issue, disclose, and verify credentials and/or presentations (e.g., for authentications),  
478 blockchain-based IDMSs commonly leverage data exchange formats such as JSON Web Token  
479 (JWT), Security Assertion Markup Language (SAML), and eXtensible Data Interchange (XDI).

#### 480 **User-Controlled Identity Wallet:**

481 A user-controlled identity wallet is an application that primarily aims at allowing a subject to hold  
482 identifiers and corresponding private keys, as well as credentials. It also serves as an interface for  
483 entities to interact with one another. For example, the subjects can receive and approve credentials  
484 from the issuers, and disclose presentations to relying parties. Actions can be initiated  
485 automatically through Application Programming Interfaces (APIs) calls that may be triggered by  
486 a user through scanning QR codes. Depending on the system identifier architecture (see Section  
487 4.4.1), a subject may be able to generate an identifier on their own directly in a wallet (it may thus  
488 be done offline).

489 Identity wallets may be linked to cloud data custodians to benefit from various services such as  
490 data and/or private key storage, backup, and recovery mechanisms. Wallets that are proposed as a  
491 service by a third party that controls a user's private keys are called custodial wallets.

492 In addition, identity wallets may act as a control center as entities can approve requests for  
493 information, thereby giving their consent to perform some action. It may also be a gateway to  
494 access and use applications and services (e.g., a decentralized application store).

495 Identity wallets may take various forms such as dedicated hardware wallets or mobile applications  
496 (or even paper wallets, private keys being simply printed out and kept somewhere safe). They may  
497 also come natively in a browser, an operating system, or as extensions.

#### 498 **Application Libraries:**

499 There exist application libraries and APIs that facilitate the integration of applications supporting  
500 the various identity management roles (e.g., requester, issuer, relying party, and verifier roles).  
501 Note that Hyperledger Aries [26] is a framework released by the Hyperledger Foundation that  
502 offers several client-side components and wallet services integration to support interactions  
503 between participants in blockchain-based IDMSs.

504 **3.5 Blockchain Identity Management Stack**

505 The Decentralized Identity Foundation published the draft protocol stack [27] shown in Table 1.  
 506 It shows a breakdown of blockchain identity management layers with the aim of facilitating the  
 507 emergence of portable and interoperable solutions. Note that adjacent layers do not have to be built  
 508 as separate applications and can be grouped together if desired for simplicity, scalability, or to  
 509 more closely align with adopted standards. While DID-specific, the stack should be similar for  
 510 approaches using other decentralized identifier systems.

511 **Table 1: Proposed Identity Stack (from the Decentralized Identity Foundation [27])**

Layer	Description
Application	Application(s) that interact with a given identity management system through library integrations and API calls
Implementation	Libraries that integrate the system in third-party applications
Payload	Message format(s) - such as JWT - used to exchange data between participants
Encoding	Method(s) for encoding data at both the encryption and payload layers
Encryption	Method(s) for encrypting messages between participants as well as encrypting the data held by the identifier owner
DID Authentication	Method(s) to authenticate a participant using their DID
Transport	Transport protocol(s) used for sharing data between participants and devices, such as Hyper-Text Transfer Protocol (HTTP) or a QR code
DID Resolution	DID Resolver used to convert a DID into its corresponding DID document
DID Operation	Create, Read, Update, and Delete operations for a DID document
DID Storage	Method for storing DID Documents and DIDs
DID Anchor	Network that serves as medium for DIDs

512

## 513 **4 Blockchain Identity Management System Taxonomy**

514 This section discusses how blockchain identity management systems are constructed and what  
515 differentiates the various approaches. We examine system authority models, identifier origination  
516 schemes, and credential issuance schemes in Section 4.1. We then evaluate methods for identifier  
517 and credential management in Section 4.2, and presentation disclosure in Section 4.3. Section 4.4  
518 looks at different system architecture designs and Section 4.5 discusses the use of public registries  
519 and related implications. We conclude our taxonomic analysis with a higher level discussion of  
520 system governance options in Section 4.6.

### 521 **4.1 Authority Model**

522 This section discusses the different control models for blockchain IDMSs and the different ways  
523 for such systems to establish new identifiers for their users.

#### 524 **4.1.1 Top-down vs Bottom-up Organizational Structures**

525 The authority model of a system specifies how it is controlled. The two main approaches are top-  
526 down and bottom-up (with the latter being frequently associated with “self-sovereign” identity  
527 schemes). Note that they form a spectrum of authority models that can support various use cases  
528 and serve as a novel medium to represent different types of power structures (with appropriate  
529 power delegation mechanisms).

#### 530 **Top-down Approach:**

531 A system owner acts as a central authority that has control over identifier origination and/or  
532 credential issuance. Power may be delegated through roles to create a hierarchical structure. This  
533 model may be appropriate for organizations that want to explore distributing their processes and  
534 architectures to better meet their needs and provide enhanced control and privacy for the users  
535 while keeping ownership of the system and control of its governance, as discussed in Section 4.6.  
536 An example system using this approach is described in *Smart Contract Federated Identity*  
537 *Management without Third Party Authentication Services* [28].

#### 538 **Bottom-up Approach:**

539 No single entity acts as a central authority that has control over identifier origination and/or  
540 credential issuance. Participants manage their own identifiers, but must still follow the rules of the  
541 IDMS (often enforced through a set of smart contracts). This approach relies on a web-of-trust,  
542 since there is no central authority. Note that this does not exclude the possibility of some entities  
543 playing more significant roles than others in designing and maintaining the system architecture  
544 and incentives.

#### 545 **4.1.2 Identifier Origination Schemes**

546 There are many possible methods for creating new identifiers within blockchain IDMSs. The  
547 generation of blockchain addresses is achieved directly by the subjects (who control the associated  
548 private keys). Blockchain addresses alone, however, do not fully meet the need of identity  
549 management; there must be additional logic to use them as identifiers in a IDMS.

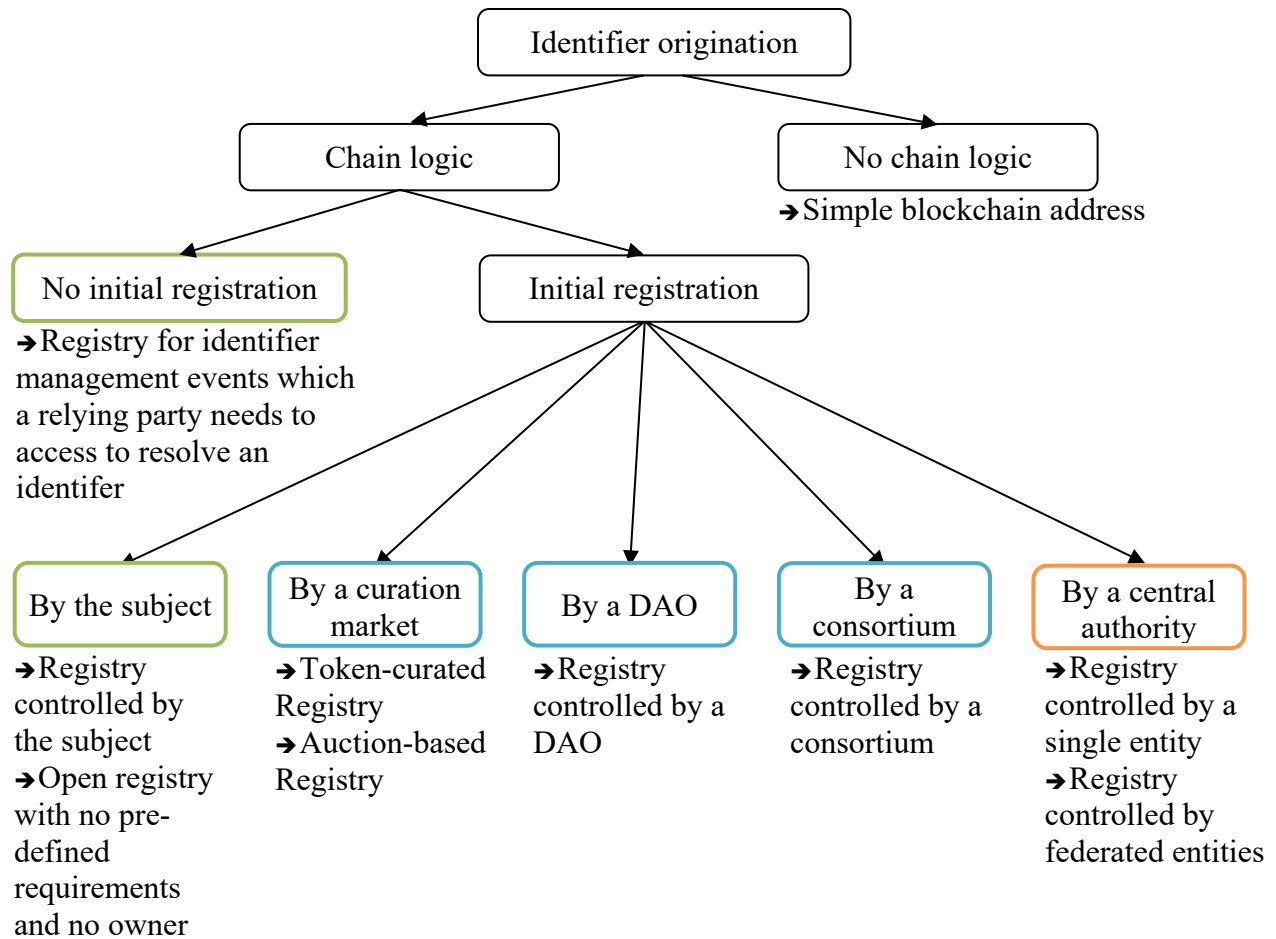


Figure 6: Identifier Origination Schemes

550 Figure 6 contains a diagram showing different methods that can be used by systems to originate  
 551 identifiers. Identifier origination based on a central authority with a top-down approach is shown  
 552 in the bottom right (in orange). Schemes involving no initial registration or a self-registration  
 553 following a bottom-up authority approach are on the left (in green). Finally, schemes involving a  
 554 curation market (see Section 4.5 on *Public Registries and Reputation Management Implications*),  
 555 a Decentralized Autonomous Organization (DAO), or a consortium can lean towards one side or  
 556 the other depending on how the permissions are implemented and controlled by the participants  
 557 (in the middle of the figure in blue). An example of DAO-controlled identifier registration for  
 558 Internet Protocol (IP) addresses can be found in [29]. Section 4.4.1 on *Identifier Architectures*  
 559 provides different approaches for implementing these identifier origination schemes.

560 **4.1.3 Credential Issuance Schemes**

561 A credential is issued to a subject by an issuer following a request by a requester. The approval of  
 562 the subject may be required and the issuer may be compensated for issuing the credential (e.g.,  
 563 through some marketplace mechanism built into the protocol).

564 With the top-down authority model, credential issuance may be controlled or regulated by a central  
 565 authority (see Section 4.1).

566 In the bottom-up authority model, any user can issue a credential to another user.<sup>3</sup>

567 A credential might also be self-issued by a subject. This would be used, for example, when a  
568 subject wants to publicly share information such as a public key, a service endpoint to make  
569 themselves reachable, or consent preferences to help other users know how to interact with them.

570 Note that a credential may be required to be issued according to a standardized nomenclature.

## 571 **4.2 Identifiers and Credentials Management**

572 This section discusses lifecycle and custody issues related to identifiers and credentials. This  
573 includes creation, issuance, discoverability, transferability, recovery, suspension, and revocation.

### 574 **4.2.1 Lifecycle**

#### 575 **Lifecycle Determination at Origination:**

576 The lifecycle of a given identifier or credential can be set at the time of origination such that there  
577 will be no need for outside intervention in the future (e.g., making it expire after a certain amount  
578 of time or making it irrevocable). This can enable an identifier or credential to take a lighter, self-  
579 supporting form in order to let the subject be more independent (see *Bring-your-own Blockchain*  
580 *Address* in Section 4.4.1.2 and *Off-chain Object* in Section 4.4.2.2). In the case where the identifier  
581 or credential is irrevocable, a relying party may not need to be actively connected to the identity  
582 management system in order to verify the credential or identifier. Alternatively, if an identifier or  
583 credential does not have its lifecycle fixed, entities need access to the blockchain to verify them.

#### 584 **Suspension and Revocation:**

585 An identifier or a credential may be suspended or revoked by the issuer, the holder, or when  
586 predefined conditions are met.<sup>4</sup> Furthermore, performing these actions may require approval from  
587 the participants involved.

### 588 **4.2.2 Custody and Delegation**

589 This section discusses the custody and delegation processes for identifiers and credentials,  
590 including ownership, storage, and transferability. Control over an identifier and/or a credential can  
591 be delegated to a custodian for a certain period of time. This can enable marketplaces to provide  
592 services while acting on behalf of the subject, such as storage, management of control and consent  
593 preferences and relationships with relying parties, recovery mechanisms in case of loss, and  
594 authenticated communication channels.

---

<sup>3</sup> There are additional advanced schemes to issue credentials anonymously and without relying on any trusted issuer by using the techniques in [30], but the claims for which these credentials are issued must be verifiable by anyone participating in that system. Another credential issuance scheme is using a threshold of mutually distrusting parties as in [31].

<sup>4</sup> Blockchains can help make the revocation process more transparent and secure; for instance, CertLedger [32] is a scheme that is comparable to Google's Certificate Transparency (CT), while preventing the "split-world" attack that is possible against CT.



595 The identifiers themselves may be stored publicly on a blockchain or may remain privately stored  
 596 and shared off-chain, depending on the IDMS (see Section 4.4.1 on *Identifier Architectures*). Users  
 597 may lose their private keys associated with an identifier, which may be recovered through a variety  
 598 of mechanisms: a custodian designated by the user, a list of user-appointed trustees (social  
 599 recovery), time delay mechanisms, and/or a central authority. Also, an identifier may be abandoned  
 600 and what is owned by the identifier transferred to another. This may be done for key rotation  
 601 purposes and not just when the private keys are lost. In Sovrin [21] for example, programs - called  
 602 “agents” – can act on behalf of an identifier and help them perform tasks such as interacting with  
 603 the ledger, transacting with other agents, or serving as backup datastores.

604 In general, credentials are not transferable from one subject to another. However, transferability  
 605 can be appropriate for specific use cases, such as representations of ownership (e.g., a certificate  
 606 proving ownership of a good that a subject may then be able to transfer on their own if and when  
 607 selling the good). Systems may implement this using some form of a non-fungible token (see  
 608 Section 4.4.2 on *Credential Architectures*).

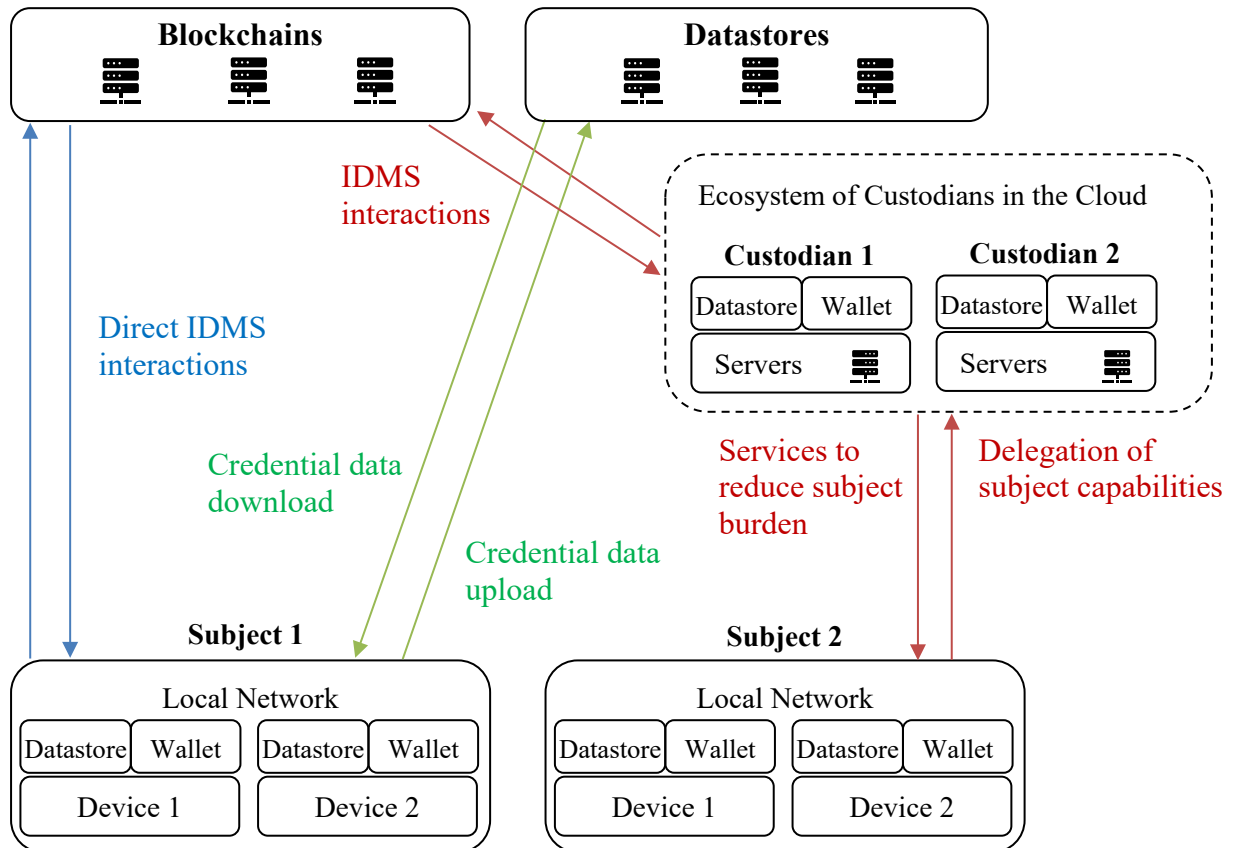


Figure 7: Interactions between subjects, custodians, and decentralized systems

609 Figure 7 is a diagram of different interactions between a subject and an identity management  
 610 system; these interactions are either direct or delegated through an identifier custodian.



### 611 **4.3 Presentation Disclosure**

612 A presentation is a quality derived from one or more credentials, which allow subjects to  
613 authenticate themselves and to share verified information with a relying party. This can reduce, or  
614 even remove, the need for a third party. The sharing of a presentation from a subject to a relying  
615 party is called presentation disclosure. This relationship comes with its own management, control,  
616 and consent considerations, which the following properties attempt to characterize.

617 Subjects can control the release of their data with relying parties (e.g., businesses, applications)  
618 and may do so at differing levels of granularity to limit information being released to the minimal  
619 necessary. Note that a subject may be compensated for presentation disclosures (e.g., with rewards  
620 and reputation systems built-in monetization schemes).

#### 621 **Selective Disclosure Mechanisms:**

622 A presentation disclosure may involve sharing an entire credential, one or more claims from a  
623 credential, or a quality derived from a credential. A presentation can include a minimal amount of  
624 information to interact with a relying party on a need-to-know basis, with a zero-knowledge proof  
625 to verify. Subjects may therefore have the ability to avoid oversharing information.

626 Zero-knowledge proofs are cryptographic schemes where a prover is able to convince a verifier  
627 that a statement is true, without providing any more information than that single bit (that is, that  
628 the statement is true rather than false).

629 Consider a patron who is stopped by the bouncer while attempting to enter a bar, because the  
630 bouncer must be convinced that the patron is at least 21 years old. The patron shows the bouncer  
631 their driver's license, the bouncer quickly looks for a birthday, and then the patron can enter if they  
632 are of age. In this scenario, the bouncer learns far more information about the patron than would  
633 be ideal, and a particularly malicious bouncer may be able to learn enough about the patron that  
634 they can commit identity theft. Contrast this example with one that employs a zero-knowledge  
635 proof scheme. The prover (the patron) proves to the verifier, the bouncer, the statement "I, the  
636 prover, am at least 21 years old". They are able to do so without revealing their birthday, driver's  
637 license number, or any other information. The patron then enters the bar with their identity and  
638 privacy secure, but a different, underage patron is unable to create a convincing proof.

639 Zero-knowledge protocols (those utilizing zero-knowledge proofs) are a major area of active  
640 research (see Section 6.2 for a high-level technical overview).

#### 641 **Pairwise-pseudonymous and Single-use Identifiers:**

642 Users may be able to maintain a set of special purpose identifiers that are not linked to the primary  
643 identifier, which enables users to maintain a level of anonymity. For example, users may use  
644 pairwise-pseudonymous identifiers, where they have a unique dedicated identifier for each  
645 relationship they have with a third-party.

646 Alternately, they may use single-use identifiers that are discarded after a particular interchange  
647 [33].

648 BIP-32 [34] can be used to create multiple unlinkable identifiers from a single master key. Note  
649 that identifier unlinkability schemes can be combined with selective disclosure mechanisms.<sup>5</sup>

#### 650 **Unicast, Multicast, and Broadcast Disclosure Modes:**

651 A presentation can be disclosed to a single relying party, a group of relying parties, or everyone.  
652 Public disclosure has reputation management implications (see Section 4.5) and is often used by  
653 relying parties who publicly disclose a presentation about themselves in order to prove who they  
654 are and to justify that they have a valid reason to request presentations and to receive personal  
655 information from participants.

#### 656 **Usability and Cost:**

657 A presentation may require on-chain processing at the time of disclosure by the subject.  
658 Alternatively, a self-contained presentation can be disclosed by the subject without interacting  
659 with the blockchain. The relying party receiving the self-contained presentation may still need  
660 blockchain access to process and verify it. These considerations result in solutions with varying  
661 usability and costs. Some actions can be achieved off-chain, quickly, and at no cost. Other ones  
662 may be free of cost, but require access to the published blocks. Finally, actions may be delayed by  
663 transaction processing time and induce costs for paying the blockchain miners to process a  
664 transaction.

665 In the case of smart contract based systems on top of permissionless blockchains, third-party  
666 entities may pay smart contract transaction fees on behalf of the users so that they do not have to  
667 deal with holding and spending the native digital currency of the blockchain themselves.

### 668 **4.4 System Architecture Designs**

669 This section focuses on the architectural design options that can be made when building a  
670 blockchain identity management system. Pieces of the system can be constructed as distinct  
671 modules or can be combined into monolithic architectures, although some designs are mutually  
672 exclusive. Note that some of them rely on on-chain registries and logic (generally implemented in  
673 the form of smart contracts on a blockchain that can be, depending on the purpose, permissionless  
674 or permissioned) that may be augmented by system-specific off-chain schemes.

675 We first discuss architectures for identifiers, then credentials, and finally, more complex  
676 combinations of architectures for identifiers, credentials, or both.

#### 677 **4.4.1 Identifier Architectures**

678 This section discusses the technical means to implement the identifier origination schemes  
679 introduced in Section 4.1.2.

---

<sup>5</sup> For example, [35] presents a system built atop Bitcoin that uses Brands' commitment scheme to let users selectively disclose their credentials via zero-knowledge proofs.

680 **4.4.1.1 On-chain Registry**

681 **Credentials Registry Acting as Identifier:**

682 For each identifier participating in the system a dedicated smart contract is deployed that can store  
683 credentials for that identifier. This architecture typically follows a bottom-up authority model  
684 approach, which is well-suited to permissionless blockchains in order to foster greater  
685 decentralization. The deployment of a new contract for every identifier allows participants to own  
686 their own contract, and thus, have control over their own identifiers. This comes at the expense of  
687 higher cost since many contracts must be deployed, more data must be posted on a blockchain,  
688 and there may be slower processing speeds due to the number of transactions on a blockchain.  
689 These aspects can hinder scalability and there are possible interoperability issues if different  
690 identity management contracts are deployed by different users (or simply different versions of the  
691 same contract). This may be mitigated by using standards such as *ERC-725 Proxy Account*, a  
692 proposed Ethereum standard that follows this architecture. It allows other smart contracts to take  
693 action based on verifiable identity information contained in *ERC-725* smart contracts. In addition,  
694 *ERC-734 Key Manager* can complement them by allowing subjects to delegate certain capabilities  
695 to custodians of their choice.

696 **Global Identifiers Registry:**

697 A single monolithic smart contract, or set of integrated contracts, is deployed that acts as a global  
698 registry for storing and managing all identifiers. It is logically centralized but physically  
699 decentralized to the extent that the blockchain nodes are distributed. This approach can follow  
700 either the top-down or bottom-up authority models, as the logical centralization does not imply  
701 control by a single entity. The writer of the smart contract can encode a variety of possible  
702 governance models. This can range from the entity deploying the contract having complete control  
703 of the system, having only limited control of it, or having no control of it. In the case of no control,  
704 the governance of the contract would be run by participating users (e.g., with a DAO). The registry  
705 can contain all the necessary logic and data to resolve identifiers to their metadata (e.g., DID  
706 documents when the DID specification is followed) or may contain only hashes which are mapped  
707 to the actual metadata stored elsewhere.

708 **Anchors Registry:**

709 A single monolithic smart contract is deployed that acts as a global registry that registers the hashes  
710 of identifier management operations that are grouped together into bundles, or “anchors”. The  
711 bundling (grouping) of identifier management operations is executed by a second layer protocol  
712 that sits on top of the blockchain to which the anchors registry is located. The protocol then adds  
713 the hashes of those anchors in the registry, and uses decentralized storage systems such as the  
714 Inter-Planetary File System (IPFS) [36] to store the anchors data (identifier management  
715 operations). The Element [25] decentralized identifier system based on the SideTree protocol  
716 (second layer protocol) on top of the Ethereum blockchain follows this architecture.

717 Note that an anchors registry (coupled with a second later protocol) may be used for any on-chain  
718 registry (e.g., one that supports credentials).

#### 719 **4.4.1.2 Bring-your-own Blockchain Address**

720 Any blockchain address is a valid identifier and can be immediately used without having to be  
721 registered beforehand. Identifier creation and storage is usually done locally in the identity wallet.  
722 This architecture follows a bottom-up authority model where the user is self-reliant; identifier  
723 creation takes place offline without any gatekeeper, and at no cost.

724 Identifier management (by the subjects) and use (by the verifiers), however, may require on chain  
725 capabilities. This differs though from the identifiers registry smart contract architecture because  
726 identifiers are initially not registered and stored on-chain, making them non-discoverable by  
727 default.

728 This architecture may help the system operate at scale since no blockchain transactions are needed  
729 for initial identifier creation. Users control their identifiers, as with the per-identifier smart contract  
730 architecture, and may gain privacy advantages as identifiers need not be publicly viewable.  
731 Moreover with identifier creation being cheap, users may utilize pairwise pseudonymous  
732 identifiers (or unique, one-time identifiers) to enhance their privacy when interacting with relying  
733 parties (see *Pairwise-pseudonymous and Single-use Identifiers* in Section 4.3). On-chain logic may  
734 be necessary to implement additional functionalities such as identifier management capabilities.  
735 For example, it will be needed to access the chain to resolve an identifier. The information  
736 necessary to do this must be stored on a blockchain and, likely, managed through some smart  
737 contract.

738 ERC-1056 *Lightweight Identity* is a proposed Ethereum standard that follows this architecture and  
739 that is used by uPort [37]. DID operations are stored in the form of Ethereum events. Resolving a  
740 DID to its DID document consists in iterating over the DID operations that may have been posted  
741 by the subject. Note that protocols that define and implement DID methods to build DID  
742 documents for bring-your-own blockchain identifiers may be further developed in a way to interact  
743 with multiple blockchains. Note that, in the case of Ethereum, blockchain log data cannot be  
744 queried from other smart contracts; however, an external method can be designed to access the  
745 chain and iterate over the logs to build a document (as in uPort).

#### 746 **4.4.2 Credential Architectures**

747 This section discusses architectural designs for storing and managing credentials. The choice of  
748 design may depend on how identifiers are managed. The credential can be stored on-chain or off-  
749 chain.

750 On-chain credentials often only require on-chain storage for the hashes of the credentials, with the  
751 non-hash data being stored on any data store a subject has access to, be it a designated custodian  
752 or a decentralized storage system such as IPFS [36].<sup>6</sup>

---

<sup>6</sup> In addition to using IPFS with an on-chain pointer, the research literature has demonstrated a number of designs for how to store credentials (and other data) off-chain securely. For example, [38] uses a blockchain for enforcing access control policies on an off-chain data store, where the off-chain data store is implemented as a distributed hash table (such as Kademlia). An alternative system, described in [39], uses centralized and decentralized databases linked together by a blockchain in order to

753 The integrity of the data may be checked by the receiving party by hashing the credential and  
754 comparing the hash with the one found on the blockchain. Note that the hashes are often stored  
755 either in the form of state variables or in the form of blockchain logs, the latter being sometimes  
756 cheaper than on-chain storage (e.g., Ethereum events).

757 Credentials can also be stored fully off-chain, either directly on the subject's device and/or by a  
758 designated custodian. There may still be, however, additional mechanisms to handle revocation.

759 There are usability, privacy, and security issues related to where credentials are stored and how  
760 they are managed.

#### 761 **4.4.2.1 On-chain Registry**

##### 762 **Per-identifier Credentials Registry:**

763 In this architecture, credentials are managed as entries in a per-identifier smart contract that acts  
764 as a container as defined in Section 4.4.1.1. This architecture can give the subject unilateral control  
765 over their credentials. As owner of the contract, a subject can remove any credential they want  
766 without the approval of the credential issuer. Also, their approval is required, in addition to the one  
767 of the credential issuers, for the issuance of a credential (see Section 4.1.3).

768 While subjects can manage their own on-chain credentials in this way, this architecture is heavily  
769 reliant on on-chain transactions. This can hinder system scalability due to blockchain transaction  
770 costs and the relatively slow processing speed for transactions. The architecture thus can make it  
771 expensive to use privacy features such as pairwise pseudonymous identifiers for every relationship  
772 (see *Pairwise-pseudonymous and Single-use Identifiers* in Section 4.3). ERC-735 *Claim Holder* is  
773 a proposed Ethereum standard that follows this architecture and can be utilized jointly with ERC-  
774 725 *Proxy Account*.

##### 775 **Global Credentials Registry:**

776 In this architecture, credentials are registered and managed as entries in a single smart contract. It  
777 is logically centralized for the entire system but physically decentralized to the extent that the  
778 blockchain nodes are distributed. Usually, the identifier that deployed the contract initially owns  
779 the system. However, that authority can be delegated, transferred, or limited depending on how  
780 the contract is coded. Thus, this architecture requires the initial owner to set up a governance model  
781 that establishes the rules and permissions for managing credentials. This may necessitate handling  
782 concepts such as reputation and negative credentials (see Section 4.5). Credential management  
783 involves on-chain transactions and access, which impacts the usability and cost of presentation  
784 disclosure as discussed in Section 4.3, as well as privacy as discussed in Section 5. This  
785 architecture can be used as a registry for revoking credentials. A relying party then is able to verify  
786 the validity of the off-chain credential.

---

allow users to exclude others from using their data, while still allowing the data to be searchable (which can be useful for areas such as medical research). Finally, Calypso [40] is a more advanced construction with auditable access control, which uses threshold cryptography to protect access to data.

787 Another use for this architecture is to allow a user to publish credentials about themselves and  
788 share information publicly such as a public key or a service endpoint.<sup>7</sup> ERC-780 *Ethereum Claims*  
789 *Registry* is a proposed Ethereum standard that follows this architecture. ERC-1056 (see Section  
790 4.4.1.2 on *Bring-your-own Blockchain Address*) also implements a credentials registry, although  
791 it is limited to self-issued credentials (and is based on blockchain logs).

## 792 **Non-fungible Token Registry:**

793 In this architecture, a credential takes the form of a Non-Fungible Token (NFT). An NFT is a  
794 unique, not interchangeable token that is owned and may be managed and traded. Minting and  
795 management of the tokens are performed through a NFT factory smart contract (that acts as a  
796 registry that manages the NFTs). NFT-based credentials primarily aim at fitting use cases that deal  
797 with digital ownership, especially, but not exclusively, when it is meant to be transferable (see  
798 Section 4.2.2 on *Custody and Delegation*). The minting of specific tokens can implement  
799 application-specific token formats, rules, and requirements and therefore provide token lifecycle  
800 management capabilities. In addition, this architecture can use interoperable token formats thus  
801 enabling a marketplace for transferable credentials. NFTs can either be issued individually or to a  
802 group (a distribution method also called “airdrop”). These capabilities come at the expense of the  
803 need for participants to issue blockchain transactions and have blockchain access (See *Usability*  
804 *and Cost* in Section 4.3).

805 The ERC-721 *Non-fungible Token Standard* is a proposed Ethereum standard that follows this  
806 architecture. As an example, 0xcert provides a framework for building decentralized applications  
807 that aim at creating and managing ERC-721-compliant NFT-based credentials [42].

## 808 **Entitlement to a User-mintable Non-fungible Token:**

809 In this architecture, a credential takes the form of an entitlement to let a user mint a pre-defined  
810 and pre-assigned NFT at a future date or condition.

811 This can be achieved through system-specific NFT factory smart contract designs. As an example,  
812 Centrifuge [43] allows one to turn credentials, of which the hashes are stored on-chain, into NFTs.  
813 The proof that one is entitled to mint a given NFT is verified through the Merkle root hash (stored  
814 on-chain) of some of the off-chain credential data.

815 This may also be achieved for a group of subjects through the use of a Merkle airdrop (see  
816 definition in Glossary in Appendix B), which allows group distribution of the entitlement to  
817 redeem an NFT. This scheme is highly scalable in that it requires only one transaction by the issuer  
818 and is independent of the size of the group. No management support is needed after the distribution  
819 as all of the activity comes from the subject side.

---

<sup>7</sup> Advanced cryptographic primitives, such as the hash-based accumulator employed in [41], can allow a registry to retain a constant-sized storage regardless of how many credentials are registered.



820 A credential is private by default, and a subject can redeem it only if they want to use or transfer  
821 it. However, the list of all the identifiers the Merkle airdrop was issued to must be available to the  
822 subjects to redeem their NFT (both the private key and the list of all the identifiers included in the  
823 Merkle airdrop are needed to build the Merkle proof and mint the NFT). Note that for a Merkle  
824 airdrop, the tokens must be “pulled” by the users, while for a traditional airdrop, the tokens are  
825 “pushed” to the user and even those who do not want to receive them.

#### 826 **4.4.2.2 Off-chain Object**

827 In this architecture, a credential takes the form of an off-chain object that acts as a self-contained  
828 vehicle for transmitting information directly between parties. This can go hand in hand with the  
829 *Bring-your-own Blockchain Address* architecture discussed in Section 4.4.1.2 to establish a  
830 lightweight identity management system that can operate at scale. It best matches use cases where  
831 the lifecycle of a credential is predetermined. However, verification of a credential (see *Lifecycle*  
832 *Determination at Origination* in Section 4.2.1) may require chain access (see *Usability and Cost*  
833 in Section 4.3). In particular, if revocability is permitted, on-chain artifacts are required for one to  
834 check if the credential was revoked, such as with a credential revocations registry (see *Off-chain*  
835 *Objects coupled with Global Credentials Registry* in Section 4.4.3).

836 It can provide a high level of user control as the subjects own their own credentials. It ensures  
837 privacy by default and need not be constrained to a specific blockchain. This architecture may use,  
838 for example, the JWT format (see Section 3.4 on *Building Blocks*), as in Blockstack [44].

#### 839 **4.4.3 Combination Patterns**

840 It is possible to combine the architectures for identifiers, credentials, or both. This section provides  
841 some examples of how this is being done, but is not exhaustive.

##### 842 **Global Identifiers Registry coupled with Per-identifier Credentials Registry:**

843 An IDMS can be designed so that identifiers are stored in a global registry, but each identifier has  
844 their own dedicated smart contract for storing and managing credentials.

845 The Smart ID project from Deloitte [45] follows this architecture. Note that the global identifiers  
846 registry may also serve as a smart contract factory to create and manage all of the per-identifier  
847 credentials registry smart contracts.

##### 848 **Global Registry for Both Identifiers and Credentials:**

849 A single smart contract can implement both an identifiers registry and a credentials registry as  
850 described in Section 4.4.1.1 and 4.4.2.1.

851 This approach is followed in *Smart Contract Federated Identity Management without Third Party*  
852 *Authentication Services* [28]. Another example of this approach is SCPKI (Smart Contract-based  
853 PKI) [46], which stores all identifiers and credentials on a single smart contract, and allows relying  
854 parties to use a web of trust to decide whether or not an identifier is authorized to perform some  
855 action. SCPKI can also be extended with blind signatures in order to provide privacy [47].

856 Another example is that of BlockPKI [48], which can generate one or more smart contracts per  
857 identifier in the system. These per-identifier contracts (called “certificates”) contain a set of  
858 credentials and are used to store signatures from certificate authorities; once enough signatures  
859 have been gathered in a contract, they are aggregated and then sent along with the certificate data  
860 to a global credentials registry contract. Relying parties can use this global credentials registry to  
861 verify signed certificates in the system.

#### 862 **Off-chain Objects coupled with Global Credentials Registry:**

863 Off-chain objects can be used as the primary way to issue and share credentials while relying on a  
864 central registry smart contract to publicly store the service endpoint URLs and public keys  
865 necessary for the participants to discover and authenticate one another.

866 A credentials registry can also be leveraged to act as a revocation registry for off-chain credentials.  
867 Such a registry is used in both uPort [37] (it is based on ERC-780 and deployed on the public  
868 Ethereum blockchain) and Hyperledger Indy [22]. In the latter, an issuer can control a revocation  
869 registry that relies on a cryptographic accumulator (protocol that allows one to prove a membership  
870 in a set; see Section 6.2 on zero-knowledge protocols) to let relying parties verify whether a given  
871 credential was revoked by the issuer or not without compromising the privacy of the registry.

#### 872 **Off-chain Objects coupled with Global Identifiers Registry for Issuers:**

873 Issuers have their identifiers stored on an on-chain registry. They can issue off-chain credentials  
874 directly to any blockchain addresses controlled by the subjects. Verifiers only need to verify that  
875 the signatures of the credentials issuers match those on the on-chain registry.

#### 876 **Non-fungible Tokens with Global Credentials Registry:**

877 Rules and permissions based on a central registry, which may be implemented in a smart contract,  
878 can be implemented to restrict the context in which transfers of NFT-based credentials take place  
879 (if they are allowed). This way, parties that trust each other can transact securely and according to  
880 the agreed-upon rules.

881 This can be leveraged, for example, to establish Know-Your-Customer (KYC) checks for  
882 exchanges of tokens as in the Transaction Permission Layer Protocol [49] with the ERC-1616  
883 *Attribute Registry* Ethereum standard proposal.

### 884 **4.5 Public Registries and Reputation Management Implications**

885 Some blockchain IDMS architectures rely on on-chain registries, and therefore, may have publicly  
886 readable data stored in a central location (e.g., smart contracts). This can be leveraged by subjects  
887 wanting to share public information about themselves (e.g., a service endpoint at which they can  
888 be reached if they wish to be discoverable). It can also be used by organizations wanting to build  
889 reputation systems such as public institutions (e.g., TheOrgBook project of the Government of  
890 British Columbia [50] running on the Verifiable Organization Network, “a public repository of  
891 verifiable claims about organizations”) and e-commerce platforms (e.g., product and seller  
892 ratings).



893 The public centralized architecture does not necessarily imply that the user privacy is violated or  
894 that users do not have control over their identity. Schemes may use hashing or encryption to protect  
895 publicly posted data and varying degrees of granularity can be implemented enabling users to  
896 manage their own credentials and associated reputation. One important design feature is whether  
897 or not user consent will be required prior to a credential being issued to that user; the user may  
898 view certain claims about themselves as being negative and not want them published. Some  
899 systems allow unilateral claim issuance while others require user approval. If the user can not stop  
900 the claim from being issued, they may then want to get a counter-claim issued. A reputation system  
901 may be used to track the reputation of issuers, which verifiers then can evaluate. Note that such  
902 systems must protect themselves from, and may be subject to, attacks designed to inappropriately  
903 alter user reputation.

#### 904 **Sybil Attacks and Structural Barriers:**

905 Reputation systems need to protect against Sybil attacks, where an attacker pretends to be many  
906 people at once, by imposing a structural barrier. For systems with access control (that may sit on  
907 top of either a permissionless or permissioned blockchain), it can take the form of identifier  
908 verification and the use of roles and permissions (e.g., TheOrgBook [50]). For open systems, the  
909 structural barrier can be made of a cost to register, exist in, and/or exit the system. This makes  
910 attacking the system disproportionately expensive compared to the benefits the attack would  
911 produce. While transaction fees act as a basic cost structure, more advanced ones relying on game  
912 theoretic concepts can be designed to achieve objectives such as disincentivizing participants from  
913 leaving an identity to regain newcomer status and ensuring participants do not get an advantage  
914 by issuing multiple identities.<sup>8</sup>

915 An example of such a cost structure are “token-curated registries”, which feature an incentivized  
916 voting game to let a community of participants decide whether an entry should be added or  
917 removed from the registry. These Sybil-resistance mechanisms can be based on staking funds (e.g,  
918 with collateral and/or escrow contracts), reputation, or work (committing a certain amount of  
919 resources for a certain period of time).

#### 920 **4.6 System Governance**

921 Blockchain-based IDMSs must have a governance structure that makes the system trustworthy to  
922 its participants. Approaches can vary significantly, and often involve a combination of both on-  
923 chain and off-chain organizational structures.

924 The on-chain structures can consist of smart contracts deployed on some underlying blockchain  
925 (either permissioned or permissionless); users are thus required to trust both the governance  
926 models of the smart contract-based system and the underlying blockchain. Alternatively, solutions  
927 exist where a blockchain is developed and deployed for the sole purpose of supporting an IDMS,  
928 called “application-specific” blockchains.

---

<sup>8</sup> [51] describes three other types of generic attacks against a reputation system - bad-mouthing, ballot-stuffing, and whitewashing, and proposes a blockchain-based solution to mitigate them. [52] is another blockchain-based reputation system designed for reputation in file-sharing networks or for e-commerce, while [53] aggregates social media reputation.

929 There may be security tradeoffs between these approaches. If the blockchain is not application-  
930 specific, governance of the blockchain itself is an important topic but not the focus of this paper,  
931 which examines the identity application specifically (a few applicable considerations are provided  
932 for reference in Section 6.1 on *Underlying Blockchain Considerations*).

933 A set of the higher level recurring governance traits are discussed below.

#### 934 **Ownership and Funding:**

935 A system can be owned by a for-profit organization (e.g., a company), a non-profit organization  
936 (e.g., a foundation), a consortium, a government agency, an open-source community, and/or a  
937 DAO.

938 It can be directly financed through traditional fundraising and monetized by the entities that  
939 administer it. It can also rely on crowdfunding, through an Initial Coin Offering (ICO) for example.  
940 Note that token-holders are not necessarily share-holders of the system in that the tokens may not  
941 give any piece of ownership of the system. Finally, the system may have no dedicated funding at  
942 all, and be maintained solely on a volunteer basis by the members of the community.

#### 943 **Operating Model:**

944 An IDMS can be designed and administered as a permissioned system to meet the internal needs  
945 of the members of an organization or a group of organizations. This means that only an approved  
946 set of users may access and maintain the system.<sup>9</sup> This permissioned system might be offered as a  
947 proprietary service to customers or it might be deployed internally. Access control takes place  
948 either at the smart contract level (that sit on top of an underlying blockchain) and/or at the  
949 blockchain protocol level (i.e., a permissioned blockchain).

950 Note that all permissioned blockchains require identity management systems to determine who the  
951 validators are (for example, in a proof of authority consensus model). This may take place off-  
952 chain (typically the validator nodes have a list of the other nodes that they want to connect with),  
953 or via smart contracts on-chain. Changes to the list of validators may then be administered through  
954 on-chain voting by administrators.

955 Alternatively, an IDMS may form an open protocol and/or ecosystem that can be used and  
956 integrated by anyone. It can be a general-purpose ecosystem, or an application-specific one (e.g.,  
957 credit scoring with Bloom Protocol [55]). Furthermore, an IDMS can involve users authenticating  
958 at the application level, or at the ecosystem level such as in Blockstack [44]. The latter differs from  
959 traditional “single sign-on” identity management in that identifier origination, credential issuance,  
960 and presentation disclosure are not necessarily controlled by a single entity.

---

<sup>9</sup> A second layer protocol can be used as an access control mechanism for permissioned blockchains. For example, the ChainAnchor scheme [54] offers this, while allowing users to transact pseudonymously and maintain transaction unlinkability: users can selectively disclose their transactions if asked to (e.g., for regulatory purposes) without revealing their other transactions. This scheme makes use of the “Enhanced Privacy ID” zero-knowledge protocol.

961 In some systems, tokens may be utilized to design an incentive structure and boost certain desired  
962 behaviors from the participants (e.g., through earning rewards) to facilitate ecosystem  
963 coordination, self-sustainability, and growth (it can be based on various game theory techniques).  
964 The incentive structure can be extended to built-in monetization schemes to buy and sell services.  
965 More specifically, they may be coded directly as part of the functions that implement actions such  
966 as credential issuance and presentation disclosure.

#### 967 **Internal Rules Management:**

968 Every system will have rules that dictate how participants interact with a given system. These rules  
969 are often implemented and enforced through smart contract code that is visible to all participants.  
970 Since the underlying blockchain enforces correct execution of the smart contract, users can trust  
971 that these rules will be executed correctly. These rules may also specify how changes to the rules  
972 themselves are managed (e.g., how the system is upgraded). Allowing such rule changes may  
973 prove beneficial - even necessary - for mitigating security issues or adding new features. However,  
974 allowing arbitrary changes can hinder user trust in the system, especially changes done without  
975 user consent. Thus, the upgradability of these systems can be treated carefully so that expectations  
976 regarding the immutability of contracts remain valid [56]. It may be important that there exist  
977 platforms to communicate and facilitate decision-making among stakeholders of a system (e.g., to  
978 raise awareness of the desired benefits and the associated risks of a certain proposal).

979 The modifications to the smart contracts can be actively governed by the system's users through a  
980 voting system (like Bloom's polling mechanism [55]) or through a Decentralized Autonomous  
981 Organization (DAO). The modifications may also be enforced with a time delay to let participants  
982 opt-out of the system if they are not satisfied with the rule changes. Lastly, it is possible that a  
983 system may have multiple versions live simultaneously (for example, both the upgraded and non-  
984 upgraded versions). This allows participants to opt into updating to the new version. Finally, note  
985 that time-stamped entries in an on-chain public registry (immutable and tamper-resistant) can  
986 facilitate accountability by serving as support for posting update proposals using accounts with  
987 identifiers registered in the system.

#### 988 **Software Management:**

989 The management of the software for a system is a vital governance issue as the software  
990 implements the rules and maintains the system, but also provides the users' portal into the system  
991 (e.g., in the form of decentralized applications, sometimes called "dapps"). Blockchains can  
992 provide significant security advantages for identity management systems, but if the user software  
993 is vulnerable, corrupted, or malicious these protections mean little.

994 The software can be managed by the developers as an open-source project shared publicly on a  
995 version control platform such as Github, or the software can be proprietary. Development patterns  
996 can be leveraged to enable smart contract upgradability (e.g., a registry contract that points to the  
997 latest version of the main contract of the system or an interface contract that is inherited by the  
998 system and defines a set of key functions and parameters). Periodic third-party audits, automated  
999 tests, and reports can also be performed and disclosed to help assess whether the rules are properly  
1000 enforced.

1001 **External Influences:**

1002 A given blockchain-based IDMS can be subject to external influences (that may depend on its  
1003 operating model) such as:

- 1004 • Regulatory compliance requirements (e.g., the European Union’s General Data Protection  
1005 Regulation), and law enforcement.
- 1006 • Industry alliances (e.g., the Ethereum Enterprise Alliance, Hyperledger Foundation,  
1007 Decentralized Identity Foundation, Trusted IoT Alliance) and standards bodies (e.g.,  
1008 International Organization for Standardization (ISO), Internet Engineering Task Force  
1009 (IETF)) that publish specifications, formats, protocols, and patterns.
- 1010 • Peer-reviewed research and bug bounty programs.
- 1011 • Social norms and user expectations.

1012 A key implication is that they introduce a certain framework of disclosure and transparency, which  
1013 might directly affect or even require certain protocol designs. This may help participants be aware  
1014 of, supportive of, and ideally, educated about, the rules of the platform. Community expectations  
1015 may play a significant role in holding the administrators of a system accountable (especially if the  
1016 community has the means to opt-out at a reasonable cost and to port their accounts to another  
1017 provider).

## 1018 **5 Security and Risk Management**

1019 Blockchains can provide security advantages to a variety of applications by removing or reducing  
1020 the need for trusted third parties. Second layer protocols can add more flexibility and may help  
1021 better scalability, privacy, and interoperability. These foundational building blocks can provide  
1022 enhanced integrity and resiliency. However, blockchains do not solve all security issues, and  
1023 careful examination of the risks and challenges of blockchain usage is needed.

1024 Some of these issues and associated mitigations are discussed below.

### 1025 **Private Data Leak:**

1026 When a user shares personal data with a relying party, the relying party may share that data outside  
1027 of the context of the IDMS. This is a significant problem for any identity management system  
1028 where user personal data is shared. However, this can be minimized by the use of minimal  
1029 presentation disclosure mechanisms. For example, zero-knowledge protocols may be utilized to  
1030 share presentations that contain only the necessary information for a given interaction to relying  
1031 parties rather than full credentials.

1032 Separately, architectures that put less data on-chain may in general be more privacy preserving,  
1033 but it depends on the exact architecture being used and how that data is being stored (e.g.,  
1034 unencrypted, encrypted, pointers to outside repositories, or hashes). Finally, vulnerabilities may  
1035 be found in the authentication and messaging protocols used by a given system to support peer-to-  
1036 peer data transmissions.

### 1037 **Metadata Tracing:**

1038 Pattern analysis techniques may be applied by attackers to on-chain metadata and possible  
1039 interceptions of messages between parties. They may look at, for example, the time that  
1040 transactions or credentials were submitted to the blockchain, which issuers signed them, or the IP  
1041 addresses that they were broadcast from. This information may be leveraged by attackers to  
1042 compromise the confidentiality of Personally Identifiable Information (PII). This correlation risk  
1043 can be minimized by decoupling users from a unique persistent identifier through the use of  
1044 pairwise pseudonymous identifier (or more advanced identifier unlinkability techniques). Zero-  
1045 knowledge proofs may also be used to obfuscate the details of blockchain transactions.

### 1046 **Replay Attacks and Impersonation:**

1047 A rogue relying party can attempt to collect user credentials and presentations in the aim of fooling  
1048 another relying party into believing that they are that user. This kind of man-in-the-middle attack  
1049 can be mitigated through relying parties using certain challenge response protocols and encrypted  
1050 tunnels such that the subjects must always prove their identity (that they know the private key for  
1051 the identifier associated with the transaction).

1052 **Private Key Compromise:**

1053 In most IDMSs, knowledge of a private key for an identifier is equivalent to owning the identifier.  
1054 Thus, preventing the compromise of private keys is essential. Keys can be compromised due to  
1055 errors in key generation, storage, or use, or can be stolen by malicious actors. Human errors can  
1056 be mitigated through well-designed tools for key management and secret sharing (typically that is  
1057 a user-friendly identity wallet); as discussed in [57], a system may be secure only if it is usable.  
1058 Once lost or stolen, identifier recovery mechanisms may be implemented to enable a subject to  
1059 regain control of an identifier (see Section 4.4.2.1). In general, architectures that provide more  
1060 privacy may reduce the risk of being targeted and having private keys stolen.

1061 **Data Withholding Attacks and Data Availability Issues:**

1062 When users manage their identifiers and credentials themselves, they benefit from a high-level of  
1063 autonomy and can ensure the availability of their data. An alternate approach is for users to choose  
1064 to rely on custodians to hold and manage their data for convenience. However, custodians can  
1065 misbehave, compromising the ability of the user to access their identifiers and credentials.  
1066 Although proper delegated control restrictions can help constrain such a rogue custodian, this does  
1067 not prevent data withholding attacks. Even a well-behaved custodian can experience temporary  
1068 service disruptions (or even go out of business), thus making user data unavailable.

1069 Therefore, it may be important for a subject to implement data redundancy by storing multiple  
1070 copies of identifier and credential data in locations that are either directly controlled by the user  
1071 (such as identity wallets across different personal devices) or delegated to custodians with proper  
1072 access and control permissions in place. This could involve identity hubs as mentioned in Section  
1073 3.3 on *Emerging Standards*. Note that these are issues with traditional IDMSs and that the use of  
1074 blockchain can be seen as a potential improvement.

1075 **Quantum Computers:**

1076 Blockchain networks depend on cryptography for their security, in particular, on public-key  
1077 cryptography. If a sufficiently powerful quantum computer is built in the future, the most widely  
1078 used public key cryptographic algorithms in blockchain systems will become insecure. This  
1079 represents a long term concern for identity data stored on a blockchain. Note that this concern  
1080 applies to the entire Internet; it is not just a concern for blockchain technology.

1081 **Smart Contract Flaws:**

1082 The smart contracts implemented to support the blockchain-based IDMS may have security flaws.  
1083 Such contracts are usually short and concise, but nonetheless there have been flaws discovered in  
1084 published smart contracts that enabled them to be compromised.

1085 Audits, tests, and the use of well-audited libraries can help mitigate this risk. Furthermore, data  
1086 integrity at the smart contract level may be achieved by establishing permissions to prevent  
1087 unauthorized participants from accessing and modifying user identifiers and credentials.

1088 **System Governance Design Flaws:**

1089 Some blockchain identity management system architectures (e.g., top-down authority models)  
1090 may incorporate logic that creates single points of failure. For example, they may provide a certain  
1091 type of participant a high level of privilege that could be improperly used.

1092 This can be mitigated against by instituting appropriate separation of authorities between  
1093 participants along with a security analysis of the system to identify single points of failure with  
1094 respect to bad actors in the system. Furthermore, governance architectures that rely on game  
1095 theoretic incentives have their own risks (e.g., see Section 4.5 on *Public Registries and Reputation*  
1096 *Management Implications*).

1097 **Oracles and Second Layer Protocol Compromise:**

1098 A blockchain IDMS may integrate off-chain data, logic, and processing in the form of oracles and  
1099 second layer protocols. Should they get compromised, the on-chain part of the system may not be  
1100 able to identify the threat adequately and cope with the compromised data, resulting in a “garbage  
1101 in, garbage out” situation. It is therefore important to ensure that necessary checks and balances  
1102 are in place.



## 1103 **6 Additional Considerations**

1104 This section provides additional considerations regarding some of the fundamental topics of  
1105 blockchain identity management discussed previously.

### 1106 **6.1 Underlying Blockchain Implications**

1107 Blockchains have unique properties and underlying governance implications that must be  
1108 considered while designing an identity management system or deciding on one to use. *Blockchain*  
1109 *Technology Overview* NIST-IR 8202 [11] in Section 7.2 *Users Involved in Blockchain Governance*  
1110 states: “the software developers, publishing nodes, and blockchain network users all play a part in  
1111 the blockchain network governance”. Below are some key considerations.

#### 1112 **Data Persistence and Privacy:**

1113 Any data added to a blockchain will be available permanently. This can have substantial  
1114 ramifications for privacy in multiple ways:

- 1115 • If personal information is encrypted and then stored on a blockchain, confidentiality for  
1116 that data will be lost if the encryption algorithm is broken.
- 1117 • Over time, as more and more individual metadata is shared with various relying parties and  
1118 credential issuers, it can be correlated with on-chain data in order to link users and their  
1119 activities (see *Metadata Tracing* in Section 5 on *Security and Risk Management*).

1120 While the effects of metadata tracking in these systems requires more study, the permanence of  
1121 blockchain data will affect anyone who uses a blockchain-based IDMS. However, note that there  
1122 are systems being developed and implemented into production that may allow the building of finer  
1123 privacy solutions.

#### 1124 **Consensus Algorithms, Time Delays, and Data Integrity:**

1125 Working with blockchains means that their operations rely on distributed consensus algorithms.  
1126 There are a wide variety of consensus algorithms – including both permissioned and  
1127 permissionless ones – and they have different properties that may be important to schemes built  
1128 on top of ledgers that use them. A consequence of this is that a scheme built on top of blockchain  
1129 A may have different security, integrity, and usability considerations than an otherwise identical  
1130 scheme built on blockchain B.

1131 The simplest example of this is the expected delay between broadcasting a transaction and having  
1132 it included in a block. Permissioned consensus algorithms tend to find blocks within seconds,  
1133 whereas the Bitcoin network, for example, experiences an approximately 10-minute delay between  
1134 finding new blocks. If an on-chain claim were issued on Bitcoin, it could take an hour or more  
1135 before it is recognized by relying parties. Verifiers often need access to this blockchain data to  
1136 compare revealed information against public hashes of that data or query an on-chain revocation  
1137 registry. The time delay for releasing blocks, or for reading and processing newly published ones,  
1138 can affect the view the application has of the current data.

1139



1140 **Blockchain Forks:**

1141 Another potential issue is that of chain splits, such as that which occurred between Ethereum and  
1142 Ethereum Classic. When some kinds of disputes arise between users or stakeholders in a  
1143 blockchain system, a single chain can split into two chains with a shared history up until the point  
1144 of the split. If a smart contract existed on the chain prior to the split, it will have its state, history,  
1145 and logic copied to both chains. This can cause confusion for users, especially during the time  
1146 around the split. It may present further issues, such as replay attacks, such that a transaction that  
1147 is valid on one chain is also valid on the other – even if the transaction is only intended for a single  
1148 chain. This may require relying parties and users to monitor both chains for some period of time.

1149 **Blockchain Resiliency:**

1150 As NIST-IR 8202 states, “Traditional centralized systems are created and taken down constantly,  
1151 and blockchain networks will likely not be different. However, because they are decentralized,  
1152 there is a chance that when a blockchain network “shuts down” it will never be fully shut down,  
1153 and that there may always be some lingering blockchain nodes running. A defunct blockchain  
1154 would not be suitable for a historical record, since without many publishing nodes, a malicious  
1155 user could easily overpower the few publishing nodes left and redo and replace any number of  
1156 blocks.”

1157 For an IDMS built on top of an underlying blockchain, it is important to carefully monitor the  
1158 validators’ activity and to establish security thresholds and metrics to ensure that the increased risk  
1159 of attacks on a declining blockchain are understood and considered acceptable. When a blockchain  
1160 is deemed insecure, an identity management system may be migrated to a more secure one.

1161 **6.2 Introduction to Zero-Knowledge Protocols**

1162 Zero-knowledge protocols (abbreviated ZK protocols, or ZKP) can play a fundamental role in  
1163 blockchain-based identity management systems for transaction confidentiality, user identification,  
1164 and presentation disclosure. Credentials can be taken as input to build presentations using zero-  
1165 knowledge proofs, which allow subjects to control the amount of information disclosed to relying  
1166 parties and the context the presentation takes place in (see Section 4.3 on *Presentation Disclosure*).

1167 The notion of zero-knowledge was first introduced in 1985 [58] and has since evolved into a class  
1168 of algorithms with several practical applications [59, 60]. This section presents a high-level  
1169 overview of zero-knowledge protocols and their role in identity management. We encourage the  
1170 reader to explore specialized publications such as [61] to gain a deeper understanding of zero-  
1171 knowledge protocols. Note that ZKProof.org is an initiative led by industry and academia to  
1172 standardize the use of zero knowledge proofs.

1173 **Definition and Properties:**

1174 There are at least two parties in a ZK protocol: a prover and a verifier. The prover aims to convince  
1175 the verifier that a statement is true without revealing any additional information. There are four  
1176 common statement types, though the following is not an exhaustive list:

- 1177 • An equality statement (the subject's bank account balance is equal to  $x$ ), or non-equality  
1178 statement.
- 1179 • An inequality statement (the subject's bank account balance exceeds  $x$ ).
- 1180 • A range statement (the subject's bank account balance is within interval  $[a, b]$ ), or out-of-  
1181 range statement.
- 1182 • A membership statement (the subject is on the client list of bank  $X$ ), or non-membership  
1183 statement.

1184 Generally, there are two kinds of ZK protocols: interactive and non-interactive. In an interactive  
1185 ZK protocol, the prover and verifier engage in at least three rounds of communication exchange.  
1186 Such protocols permit the verifier to submit challenges to the prover, whereby the prover replies  
1187 with responses that reinforce the validity of the prover's original statement. There is no challenge-  
1188 response interaction in non-interactive ZK protocols, though there is sometimes a common  
1189 reference string shared in advance by both parties.

1190 A ZK protocol produces a proof which is sent to the verifier. For statement  $S$ , prover  $P$ , and verifier  
1191  $V$ , the resulting proof  $\pi$  must satisfy the three following properties to be considered secure:

- 1192 • Completeness: If  $S$  is true, then  $\pi$  will convince  $V$  that  $S$  is true with overwhelming  
1193 probability.
- 1194 • Soundness: If  $S$  is false, then the probability that  $P$  can convince  $V$  that  $S$  is true is  
1195 negligible.
- 1196 • Zero-knowledge: If  $S$  is true, then  $V$  learns nothing from  $\pi$  besides the fact that  $S$  is true.

1197 The soundness property captures the inability for a prover  $P$  to convince the verifier  $V$  of a false  
1198 assertion. If, for example,  $P$  can cheat with probability  $1/3$ , then the ZK protocol may need to be  
1199 repeated  $n$  times to reduce the soundness error from  $1/3$  to  $1/3^n$ .

1200 The zero knowledge property can be statistical, or computational. If the verifier is assumed to have  
1201 unlimited computational resources but learns no additional information from the protocol, then the  
1202 protocol is considered to achieve statistical zero knowledge. If the zero knowledge property holds  
1203 by some assumption about the verifier's computational power, then the protocol achieves  
1204 computational zero knowledge.

1205 **Usability and Cost:**

1206 The scalability and cost of ZK protocols depend on the succinctness of the proof. It measures the  
1207 required storage size of the proof, the proving time, and the verification time; these considerations  
1208 are of special interest for blockchain-based ZKP schemes, with the blockchain having its own  
1209 limited storage and transaction speed.

1210 Note that the trusted setup phase that is required for some zero-knowledge protocols (e.g., the zk-  
1211 SNARKs protocol implemented in Zcash [60]) involves a significant initial cost, but then enables  
1212 verifications of the proof to require fewer resources (it allows a statement to be proven many times  
1213 by verifiers that have limited time and resources).

### 1214 **6.3 Presentation Sharing and Data Mining**

1215 This section discusses protocols – such as those based on zero-knowledge – to control the context  
1216 in which presentations may be used by relying parties for data mining and data exchanges with  
1217 third parties (i.e., other relying parties). Note that they represent advanced research topics, and  
1218 could trigger the emergence of novel data broker business models.

#### 1219 **Convincing Power:**

1220 When a subject discloses a presentation to a relying party (as discussed in Section 4.3), information  
1221 is revealed, that cannot be undone, and the relying party may share that information to other relying  
1222 parties. However, in some schemes such as interactive zero-knowledge protocols, relying parties  
1223 are, by design, unable to convince other relying parties that a statement (that a subject convinced  
1224 them was true beforehand) is true. An interactive proof typically only convinces a single verifier  
1225 that has established a direct and authenticated contact with the prover. In contrast, non-interactive  
1226 protocols may convince multiple verifiers simultaneously, and possibly at a later date.

1227 Schemes also exist where ZK protocols allow for privacy-preserving querying of credential  
1228 revocation registries (e.g., some cryptographic accumulator schemes).

#### 1229 **Benefits of Credential Properties:**

1230 Presentations can take the form of credentials to benefit from properties credentials have. For  
1231 instance, a presentation may have the ability to be accessed conditionally (see *Entitlement to a*  
1232 *User-mintable Non-fungible Token* in Section 4.4.2.1) and to be transferable (see *Non-fungible*  
1233 *Token Registry* in Section 4.4.2.1). Such presentations can also be used to derive a limited number  
1234 of presentations, like in the scheme described in [62].

#### 1235 **Presentation Encapsulation:**

1236 A relying party that receives presentations may be able to encapsulate them into another  
1237 presentation and disclose it to another relying party. In that case, the issuer of the encapsulated  
1238 presentation is not the issuer of the original presentation. However, it allows the relying party to  
1239 verify a snapshot of a presentation to another relying party (a timestamp and signature may be  
1240 added).<sup>10</sup>

---

<sup>10</sup> Non-interactive ZK protocols (NIZK protocols) being potentially transferrable, the original verifier could turn around and claim the original NIZK proof as their own while interacting with third party verifiers. [63] provides a way to tie NIZK proofs to the identity of the original prover, such that when the original verifier presents it to a third party, the third party will understand that it was the original prover, and not the original verifier, who issued the original proof.

## 1241 **6.4 Ecosystem Convergence**

1242 A key catalyzer for the development of the decentralized identity management ecosystem is the  
1243 development of standards, recommendations, and cross-ledger integrations. Contingent to this is  
1244 the identification of criteria, patterns, and best practices to understand which architecture designs  
1245 are relevant, depending on the use cases at stake, and how to assemble them into suitable solutions.  
1246 This will help inform decisions on how to use an existing system as a service, integrate one to a  
1247 given solution, or build and deploy a new one.

### 1248 **Universal Wallets:**

1249 Standards such as BIP-32 and ERC-20 facilitated the emergence of interoperable cryptocurrency  
1250 wallets. Additionally, the ecosystem of password managers (for storage and management of  
1251 traditional identifiers and credentials) can be seen as mature.

1252 In this context, the different architecture designs and components discussed in this paper, and  
1253 standards such as emerging Ethereum ones (e.g., ERC-1056, ERC-780, ERC-725, ERC-734, ERC-  
1254 735), may facilitate the emergence of interoperable user-controlled identity wallets, which  
1255 integrate identifiers and credentials, alongside cryptocurrencies and other digital assets.

1256 This can create a layer of abstraction for the users, who could access and manage all their services  
1257 and applications from a single identity management interface; these services may integrate and/or  
1258 rely on different identity management systems. This can concretely take the form of a software  
1259 suite with standalone applications and extensions for browsers and operating systems. It may serve  
1260 as a gateway to interact with third party marketplaces, applications, or stores of applications. It can  
1261 also integrate digital asset exchange platforms and identity management custodians to reduce the  
1262 burden for the users and provide additional services.

1263 As discussed in Section 3.4 on *Building Blocks*, custodial wallets are provided by a third party that  
1264 controls a user's private keys. Additional cryptographic schemes can be used to choose a trusted  
1265 third party that is not the custodian service provider itself. For example, ZenGo [64] has developed  
1266 a wallet that uses threshold signatures to create two secret shares that take the role of a user's  
1267 private key when they are combined (which controls assets and/or credentials). More shares may  
1268 be used to create schemes that require more than one trusted third party.

1269 Secret shares are also featured in the Horcrux protocol [65].<sup>11</sup> It uses the Biometric Open Protocol  
1270 Standard [66]<sup>12</sup> to power blockchain-based authentication with biometric information.

---

<sup>11</sup> In this protocol, biometric data is collected by a device owned by the user, then divided into multiple shares. One of these shares is sent to a dedicated server, which selects a blockchain, creates a DID for the biometric share and stores the resulting DID document using off-chain storage providers. The other biometric shares can similarly be assigned to other blockchains, creating more DIDs. As a result, the original biometric data can act as a junction between different identity management platforms. This can help create more robust, blockchain-agnostic solutions.

<sup>12</sup> The Biometric Open Protocol Standard (BOPS) was introduced by IEEE under reference 2410-2018. It provides a framework to support biometric authentication methods. This standard also offers guidance for identification, access control, and auditing capabilities. Dedicated Application Programming Interfaces (API) designs, device requirements, and security and privacy considerations are also introduced.

1271 **Cross-Ledger Integration:**

1272 There are several ways blockchain-based identity management systems can integrate with one  
1273 another and/or be part of a common larger structure:

- 1274 • Universal resolver: As mentioned in Section 3.3 on *Emerging Standards*, the blockchain  
1275 agnostic Universal DID Resolver maintained by the DIF allows the integration of any  
1276 identity management system, which can then be queried by the users through a common  
1277 interface.
- 1278 • Second layer protocols: As mentioned in Section 3.4 on *Building Blocks*, second layer  
1279 protocols such as the SideTree protocol [23] may also be used to interact with one or more  
1280 blockchains simultaneously and in a blockchain agnostic manner.
- 1281 • Bridges: The capabilities of a given system may be integrated in another system by  
1282 implementing the libraries provided by the former system in the form of on-chain logic  
1283 (e.g., smart contracts) in the latter system. For example, Cordentia [67] is a Corda smart  
1284 contract that integrates Hyperledger Indy capabilities in the Corda platform. Thus, Corda  
1285 ledger transactions can be contingent on credentials managed with a Hyperledger Indy-  
1286 based blockchain. In addition, SecureKey has explored integrating Hyperledger Indy  
1287 capabilities in Verified.me, its Hyperledger Fabric-based identity management system  
1288 [68].

## 1289 7 Use Cases

1290 There are many uses for blockchain identity management, which can be intended to be public  
1291 facing, privacy-preserving (to provide solutions for individual users), or both. They include  
1292 financial services, reusable identities to support Anti-Money Laundering (AML) and KYC laws,  
1293 verification of certificates, traceability of assets, and supply chain management.

1294 These uses can be relevant for applications in various areas such as:

- 1295 • Education: for the issuance of transcripts, diplomas, and certifications that can then serve  
1296 as verified credentials during job applications.
- 1297 • Healthcare: for the issuance of prescriptions, submission of claims to health insurance, and  
1298 sharing of health records.
- 1299 • Banking: for account opening, fraud prevention, proof of funds, credit risk evaluation, as  
1300 well as ownership, exchange, and trading of financial assets.
- 1301 • Government services: for the issuance of driver's licenses and birth certificates,  
1302 maintaining public registries of voters.
- 1303 • Public safety: for managing sets of equipment and reliable communication permissions.
- 1304 • Manufacturing: for representing ownership of 3D models.
- 1305 • Transportation: for the identification of autonomous vehicles.
- 1306 • Data brokerage: for exchanges of datasets.

1307 We provide below two use cases in the aim of further assisting the reader in their understanding  
1308 of blockchain identity management.

### 1309 Renting a Vehicle:

1310 In this use case, we consider an individual that proves to a car rental company that they meet all  
1311 the requested requirements without disclosing more information than what is strictly needed.

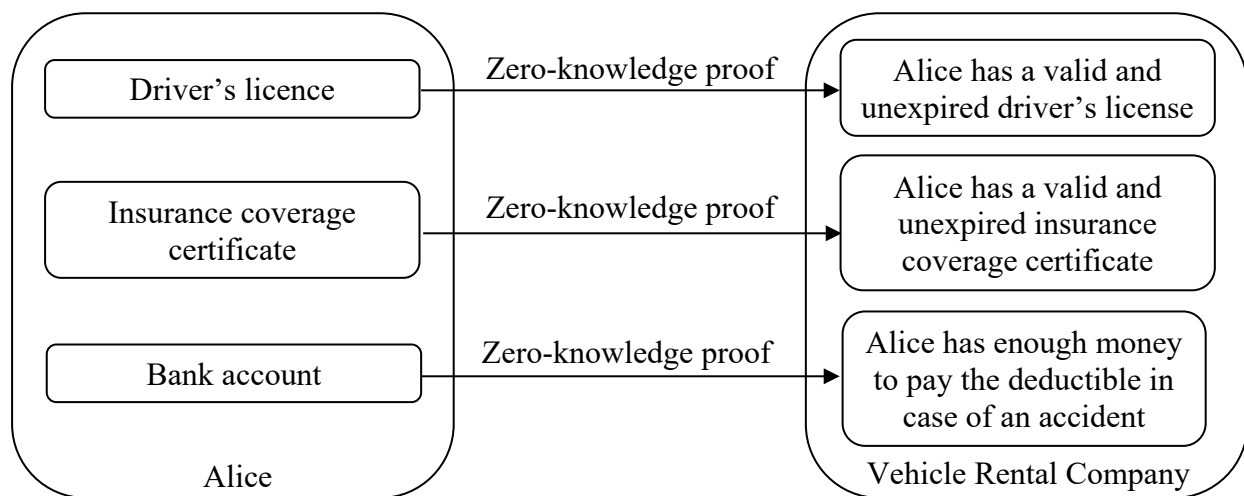


Figure 8: Minimal Disclosure to Rent a Car

1312 This takes place through a system that enables the individual to build and disclose proofs derived  
1313 from credentials that they own on a given IDMS. The credentials are: an unexpired and valid  
1314 driver's license, an insurance certificate (showing that the individual has sufficient coverage), and  
1315 a bank statement (showing that the individual has the means to pay the deductible in case of an  
1316 accident). Rather than sharing the credentials in their entirety to the rental company, the  
1317 presentation built by the system allows the individual to combine the derived information from  
1318 each credential (as shown in Figure 8) and proves that the individual meets all the requirements. It  
1319 may not even be necessary to disclose the full name of the individual.

1320 An alternative version of this scenario is that of an employee that rents a vehicle on behalf of the  
1321 company that they work for. In this case, the company can delegate access to some of its credentials  
1322 to the employee, so that information derived from these credentials can then be added to the  
1323 presentation the employee discloses to the car rental company.

#### 1324 **Exchanging Concert Tickets and Coupons:**

1325 In this use case, we consider a system controlled by a company that enables the issuance of tickets  
1326 and coupons for concerts, conferences, and other events, while allowing the users to sell or  
1327 exchange those tickets and coupons on their own.

1328 The system is owned by a ticketing company that controls initial identity proofing and user  
1329 registration. Once registered, event organizers can issue transferable tickets (in the form of non-  
1330 fungible tokens) to registered users. Although the initial registration is controlled by the system  
1331 owner, users can transfer tickets on their own (without any further approval being necessary from  
1332 the system owner). For instance, a ticket owner may be able to exchange it for one at another date,  
1333 give it to a friend, or even sell it. After attending a concert, an individual may keep the ticket as a  
1334 souvenir and add it on social media to connect with other attendees and artists.

1335 The system also implements a loyalty program to get rewards and attend other events. It  
1336 periodically distributes redeemable coupons (in the form of a Merkle airdrop of non-fungible  
1337 tokens) to the customer base that can be used to claim a discount to attend new events. While these  
1338 coupons have an expiration date and were issued to a certain group of individuals, they are  
1339 transferable. That way, an individual that receives a coupon can transfer it to a friend, thus allowing  
1340 the event organizers to reach a wider target audience.



1341 **8 Conclusion**

1342 Blockchain-based identity management is an emerging field that holds great promise in providing  
1343 improvements over the traditional and federated models currently in use. This paper provided the  
1344 reader with a general understanding of the benefits, challenges, and opportunities of such systems.  
1345 It discusses the foundational building blocks of blockchain identity management systems and the  
1346 current standardization efforts. It then identified different system properties that can be achieved  
1347 through different architectural designs using a taxonomic approach. The paper reviewed select  
1348 security and risk management issues as well as other considerations. It finished with some example  
1349 use cases highlighting the utility of these systems.

1350 Of special importance, the paper discussed the ability for blockchain identity management systems  
1351 to reduce, or even remove the need for a trusted third party in the authentication and credential  
1352 passing process with relying parties. Many other capabilities can be built into these systems and  
1353 this paper reviewed such improvements and the different architectures that can support them.  
1354 Critical to many of these benefits are the related technologies of smart contracts to act as trusted  
1355 third parties, the use of zero-knowledge proofs to avoid oversharing information, and second layer  
1356 protocols to build more scalable and private solutions.

1357 Despite having great promise, this field is still emerging and it is unclear if it will provide a usable,  
1358 secure, and scalable replacement for today's non-blockchain identity management systems. If or  
1359 when this happens, blockchain-based identity management systems would become a fundamental  
1360 architectural component of tomorrow's Internet.

1361

## References

- 1362 [1] Temoshok D, Abruzzi C (2018) Developing Trust Frameworks to Support Identity  
1363 Federations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST  
1364 Interagency or Internal Report (IR) 8149. <https://doi.org/10.6028/NIST.IR.8149>
- 1365 [2] Lyons T, Courcelas L, Timsit K (2018) Blockchain for Government and Public Services.  
1366 (European Union Blockchain Observatory & Forum).
- 1367 [3] Lyons T, Courcelas L, Timsit K (2019) Blockchain and Digital Identity. (European Union  
1368 Blockchain Observatory & Forum).
- 1369 [4] The Illinois Blockchain Initiative (2017) *Illinois Partners with Evernym to Launch Birth*  
1370 *Registration Pilot*. Available at [https://illinoisblockchain.tech/illinois-partners-with-](https://illinoisblockchain.tech/illinois-partners-with-evernym-to-launch-birth-registration-pilot-f2668664f67c)  
1371 [evernym-to-launch-birth-registration-pilot-f2668664f67c](https://illinoisblockchain.tech/illinois-partners-with-evernym-to-launch-birth-registration-pilot-f2668664f67c)
- 1372 [5] De Vault F, Hager T (2017) *Blockchain Primer: Enabling Blockchain Innovation in the*  
1373 *U.S. Federal Government*. (ACT-IAC Emerging Technology Community of Interest,  
1374 Blockchain Working Group). Available at [https://www.actiac.org/act-iac-white-paper-](https://www.actiac.org/act-iac-white-paper-enabling-blockchain-innovation-us-federal-government)  
1375 [enabling-blockchain-innovation-us-federal-government](https://www.actiac.org/act-iac-white-paper-enabling-blockchain-innovation-us-federal-government)
- 1376 [6] De Vault F, Hager T (2018) *Blockchain Playbook for Federal U.S. Government*. (ACT-  
1377 IAC Emerging Technology Community of Interest, Blockchain Working Group).  
1378 Available at [https://www.actiac.org/act-iac-white-paper-blockchain-playbook-us-federal-](https://www.actiac.org/act-iac-white-paper-blockchain-playbook-us-federal-government)  
1379 [government](https://www.actiac.org/act-iac-white-paper-blockchain-playbook-us-federal-government)
- 1380 [7] Durant E , Trachy A (2017) *Digital Diploma debuts at MIT*. (MIT News). Available at  
1381 [http://news.mit.edu/2017/mit-debuts-secure-digital-diploma-using-bitcoin-blockchain-](http://news.mit.edu/2017/mit-debuts-secure-digital-diploma-using-bitcoin-blockchain-technology-1017)  
1382 [technology-1017](http://news.mit.edu/2017/mit-debuts-secure-digital-diploma-using-bitcoin-blockchain-technology-1017)
- 1383 [8] e-Estonia (accessed 02/23/19) *FAQ Estonian Blockchain Technology*. Available at  
1384 <https://e-estonia.com/wp-content/uploads/faq-a4-v03-blockchain-1-1.pdf>
- 1385 [9] Offerman A (2018) *Swiss City of Zug issues Ethereum blockchain-based eIDs*. (Joinup).  
1386 Available at [https://joinup.ec.europa.eu/collection/egovernment/document/swiss-city-zug-](https://joinup.ec.europa.eu/collection/egovernment/document/swiss-city-zug-issues-ethereum-blockchain-based-eids)  
1387 [issues-ethereum-blockchain-based-eids](https://joinup.ec.europa.eu/collection/egovernment/document/swiss-city-zug-issues-ethereum-blockchain-based-eids)
- 1388 [10] VON (accessed 03/17/19) *Verifiable Organizations Network*. Available at <https://vonx.io>
- 1389 [11] Yaga D, Mell P, Roby N, Scarfone K (2018) Blockchain Technology Overview. (National  
1390 Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal  
1391 Report (IR) 8202. <https://doi.org/10.6028/NIST.IR.8202>
- 1392 [12] Reed D, Sporny M, Longley D, Allen C, Grant R, Sabadello M (2019) Decentralized  
1393 Identifiers (DIDs) v0.12 – Data Model and Syntaxes for Decentralized Identifiers. (W3C  
1394 Credentials Community Group)
- 1395 [13] Sporny M, Kellogg G, Lanthaler M (2014) JSON-LD 1.0 - A JSON-based serialization for

- 1396 linked data. (W3C)
- 1397 [14] Hughes A, Sporny M, Reed D (2019) A Primer for Decentralized Identifiers - An  
1398 introduction to self-administered identifiers for curious people. (W3C Credentials  
1399 Community Group)
- 1400 [15] Sabadello M (2017) *A universal resolver for self-sovereign identifiers*. (Medium -  
1401 Decentralized Identity Foundation). Available at [https://medium.com/decentralized-  
1402 identity/a-universal-resolver-for-self-sovereign-identifiers-48e6b4a5cc3c](https://medium.com/decentralized-identity/a-universal-resolver-for-self-sovereign-identifiers-48e6b4a5cc3c)
- 1403 [16] Sporny M, Longley D, Chadwick D (2019) Verifiable credentials data model 1.0 –  
1404 Expressing verifiable information on the Web. (W3C)
- 1405 [17] IMS Global (2018) Open Badges v2.0.
- 1406 [18] Decentralized Identity Foundation (accessed 03/17/19) *DIF Identity Hubs*. Available at  
1407 <https://github.com/decentralized-identity/identity-hub/blob/master/explainer.md>
- 1408 [19] Kuhn DR, Hu VC, Polk WT, Chang S-J (2001) Introduction to Public Key Technology and  
1409 the Federal PKI Infrastructure. (National Institute of Standards and Technology,  
1410 Gaithersburg, MD), NIST Special Publication (SP) 800-32.  
1411 <https://doi.org/10.6028/NIST.SP.800-32>
- 1412 [20] Barker E, Smid M, Branstad D, Chokhani S (2013) A Framework for Designing  
1413 Cryptographic Key Management Systems. (National Institute of Standards and  
1414 Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-130.  
1415 <https://doi.org/10.6028/NIST.SP.800-130>
- 1416 [21] Sovrin (accessed 02/22/19) *Sovrin*. Available at <https://sovrin.org>
- 1417 [22] Hyperledger (accessed 02/22/19) *Hyperledger Indy*. Available at  
1418 <https://www.hyperledger.org/projects/hyperledger-indy>
- 1419 [23] Decentralized Identity Foundation (accessed 05/13/19) *Sidetree Protocol Specification*.  
1420 Available at [https://github.com/decentralized-  
1421 identity/sidetree/blob/master/docs/protocol.md](https://github.com/decentralized-identity/sidetree/blob/master/docs/protocol.md)
- 1422 [24] Decentralized Identity Foundation (accessed 05/13/19) *ION*. Available at  
1423 <https://github.com/decentralized-identity/ion>
- 1424 [25] Decentralized Identity Foundation (accessed 05/20/19) *Element*. Available at  
1425 <https://github.com/decentralized-identity/element>
- 1426 [26] Hyperledger (accessed 05/29/19) *Hyperledger Aries Proposal*. Available at  
1427 <https://wiki.hyperledger.org/display/HYP/Hyperledger+Aries+Proposal>
- 1428 [27] Terbu O (2019) *The Self-sovereign Identity Stack*. (Medium - Decentralized Identity  
1429 Foundation). Available at <https://medium.com/decentralized-identity/the-self-sovereign->

- 1430 [identity-stack-8a2cc95f2d45](#)
- 1431 [28] Mell P, Dray J, Shook J (2019) Smart Contract Federated Identity Management without  
1432 Third Party Authentication Services. *Open Identity Summit 2019*.  
1433 [https://ws680.nist.gov/publication/get\\_pdf.cfm?pub\\_id=925957](https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=925957)
- 1434 [29] Angieri S, García-Martínez A, Liu B, Yan Z, Wang C, Bagnulo M (2018) An experiment  
1435 in distributed Internet address management using blockchains. *arXiv preprint*  
1436 *arXiv:1807.10528*.
- 1437 [30] Garman C, Green M, Miers I (2014) Decentralized Anonymous Credentials. *Network and*  
1438 *Distributed System Security Symposium 2014*. [https://www.ndss-symposium.org/wp-](https://www.ndss-symposium.org/wp-content/uploads/2017/09/07_3_1.pdf)  
1439 [content/uploads/2017/09/07\\_3\\_1.pdf](https://www.ndss-symposium.org/wp-content/uploads/2017/09/07_3_1.pdf)
- 1440 [31] Sonnino A, Al-Bassam M, Bano S, Meiklejohn S, Danezis G (2018) Coconut: Threshold  
1441 issuance selective disclosure credentials with applications to distributed ledgers. *arXiv*  
1442 *preprint*. <https://arxiv.org/abs/1802.07344>
- 1443 [32] Kubilay MY, Kiraz MS, Mantar HA (2018) CertLedger: A New PKI Model with  
1444 Certificate Transparency Based on Blockchain. *arXiv preprint*.  
1445 <https://arxiv.org/abs/1806.03914>
- 1446 [33] Goodell G , Aste T (2019) A Decentralised Digital Identity Architecture. *SSRN Electronic*  
1447 *Journal*. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3342238](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3342238)
- 1448 [34] Wuille P (2012) *BIP32: Hierarchical Deterministic Wallets*. Available at  
1449 <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>
- 1450 [35] Augot D, Chabanne H, Clémot O, George W (2017) Transforming face-to-face identity  
1451 proofing into anonymous digital identity using the bitcoin blockchain. *2017 15th Annual*  
1452 *Conference on Privacy, Security and Trust*, (IEEE), pp 25-2509.
- 1453 [36] IPFS (accessed 02/21/2019) *IPFS*. Available at <https://ipfs.io>
- 1454 [37] Uport (accessed 02/20/19) *Uport*. Available at <https://uport.me>
- 1455 [38] Zyskind G, Nathan O, Pentland A (2015) Decentralizing privacy: Using blockchain to  
1456 protect personal data. *2015 IEEE Security and Privacy Workshops*, (IEEE), pp 180-184.
- 1457 [39] Bertram S, Georg C-P (2018) A privacy-preserving system for data ownership using  
1458 blockchain and distributed databases. *arXiv preprint* <https://arxiv.org/abs/1810.11655>
- 1459 [40] Kokoris-Kogias E, Alp EC, Siby SD, Gailly N, Gasser L, Jovanovic P, Syta E, Ford B  
1460 (2018) Calypso: Auditable sharing of private data over blockchains. *Cryptology ePrint*  
1461 *Archive*. <https://eprint.iacr.org/2018/209/20180806:124914>
- 1462 [41] Patsonakis C, Samari K, Kiayias A, Roussopoulos M (2019) On the Practicality of Smart  
1463 Contract PKI. *arXiv preprint* <https://arxiv.org/abs/1902.00878>

- 1464 [42] Oxcert (accessed 02/22/19) *Oxcert*. Available at <https://0xcert.org>
- 1465 [43] Centrifuge (accessed 05/23/19) *Centrifuge*. Available at <https://centrifuge.io>
- 1466 [44] Blockstack (accessed 02/22/19) *Blockstack*. Available at <https://blockstack.org>
- 1467 [45] SmartIdentity (accessed 05/20/19) *Smart Identity*. Available at  
1468 <https://github.com/SmartIdentity/smartId-contracts>
- 1469 [46] Al-Bassam M (2017) SCPKI: a smart contract-based PKI and identity system. *Proceedings*  
1470 *of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, (ACM), pp 35-40.
- 1471 [47] Azouvi S, Al-Bassam M, Meiklejohn S (2017) Who am I? Secure identity registration on  
1472 distributed ledgers. *Data Privacy Management, Cryptocurrencies and Blockchain*  
1473 *Technology*, (Springer), pp 373-389.
- 1474 [48] Dykcik L, Chuat L, Szalachowski P, Perrig A (2018) BlockPKI: An Automated, Resilient,  
1475 and Transparent Public-Key Infrastructure. *2018 IEEE International Conference on Data*  
1476 *Mining Workshops*, (IEEE), pp 105-114.
- 1477 [49] TPL (accessed 04/23/19) *Transaction Permission Layer Protocol*. Available at  
1478 <https://tplprotocol.org>
- 1479 [50] Province of British Columbia (accessed 05/20/19) *TheOrgBook*. Available at  
1480 <https://github.com/bcgov/TheOrgBook>
- 1481 [51] Schaub A, Bazin R, Hasan O, Brunie L (2016) A trustless privacy-preserving reputation  
1482 system. *IFIP International Conference on ICT Systems Security and Privacy Protection*,  
1483 (Springer), pp 398-411.
- 1484 [52] Dennis R, Owen G (2015) Rep on the block: A next generation reputation system based on  
1485 the blockchain. *2015 10th International Conference for Internet Technology and Secured*  
1486 *Transactions*, (IEEE), pp 131-138.
- 1487 [53] Yasin A, Liu L (2016) An online identity and smart contract management system. *2016*  
1488 *IEEE 40th Annual Computer Software and Applications Conference*, (IEEE), pp 192-198.
- 1489 [54] Hardjono T, Pentland A (2019) Verifiable anonymous identities and access control in  
1490 permissioned blockchains. *arXiv preprint* <https://arxiv.org/abs/1903.04584>
- 1491 [55] Bloom (accessed 02/22/19) *Bloom*. Available at <https://bloom.co>
- 1492 [56] Marx S (2019) *Upgradeability Is a Bug*. (Medium - Consensys Diligence). Available at  
1493 <https://medium.com/consensys-diligence/upgradeability-is-a-bug-dba0203152ce>
- 1494 [57] Dunphy P, Petitcolas FA (2018) A first look at identity management schemes on the  
1495 blockchain. *IEEE Security & Privacy* 16(4):20-29.
- 1496 [58] Goldwasser S, Micali S, Rackoff C (1989) The knowledge complexity of interactive proof

- 1497 systems. *SIAM Journal on Computing* 18(1):186-208.
- 1498 [59] Del Pino R, Lyubashevsky V, Neven G, Seiler G (2017) Practical quantum-safe voting  
1499 from lattices. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and*  
1500 *Communications Security*, (ACM), pp 1565-1581.
- 1501 [60] Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M (2014) Zerocash:  
1502 Decentralized anonymous payments from bitcoin. *2014 IEEE Symposium on Security and*  
1503 *Privacy*, (IEEE), pp 459-474.
- 1504 [61] ZKProofs (accessed 05/13/19) *Zero-Knowledge Proofs*. Available at <https://zkp.science>
- 1505 [62] Augot D, Chabanne H, Chenevier T, George W, Lambert L (2017) A user-centric system  
1506 for verified identities on the bitcoin blockchain. *Data Privacy Management,*  
1507 *Cryptocurrencies and Blockchain Technology*, (Springer), pp 390-407.
- 1508 [63] Katz J, Ostrovsky R, Rabin MO (2004) Identity-based zero-knowledge. *International*  
1509 *Conference on Security in Communication Networks*, (Springer), pp 180-192.
- 1510 [64] KZen (accessed 06/07/19) *ZenGo*. Available at <https://zengo.com>
- 1511 [65] Othman A, Callahan J (2018) The Horcrux protocol: a method for decentralized biometric-  
1512 based self-sovereign identity. *2018 International Joint Conference on Neural Networks,*  
1513 (IEEE), pp 1-7.
- 1514 [66] IEEE (2017) *IEEE 2410-2017 - IEEE Standard for Biometric Open Protocol* (IEEE  
1515 Standards Association). <https://standards.ieee.org/standard/2410-2017.html>
- 1516 [67] Luxoft (accessed 05/29/19) *Cordentity*. Available at <https://github.com/Luxoft/cordentity>
- 1517 [68] Douglas S (2018) *SecureKey Technologies to explore interoperability between Verified.Me*  
1518 *and Hyperledger Indy*. (SecureKey) Available at  
1519 <https://securekey.com/press-releases/hyperledger-indy>
- 1520 [69] Jones M, Bradley J, Sakimura N (2015) RFC 7519: Json Web Token (JWT) (Internet  
1521 Engineering Task Force (IETF)).
- 1522 [70] Cheikes BA, Waltermire D, Scarfone K (2011) Common Platform Enumeration: Naming  
1523 Specification Version 2.3. (National Institute of Standards and Technology, Gaithersburg,  
1524 MD), NIST Interagency or Internal Report (IR) 7695.  
1525 <https://doi.org/10.6028/NIST.IR.7695>
- 1526

1527 **Appendix A—Acronyms**

1528 Selected acronyms and abbreviations used in this paper are defined below.

ACT-IAC	American Council for Technology and Industry Advisory Council
AML	Anti-Money Laundering
API	Application Programming Interface
BIP	Bitcoin Improvement Proposal
DAO	Decentralized Autonomous Organization
DID	Decentralized Identifier
DIF	Decentralized Identity Foundation
DPKI	Decentralized Public Key Infrastructure
DLT	Distributed Ledger Technology
DNS	Domain Name System
ERC	Ethereum Request for Comments
ETH	Ethereum
FIM	Federated Identity Management
HD	Hierarchical Deterministic
HTTP	Hyper-Text Transfer Protocol
ICO	Initial Coin Offering
IDMS	Identity Management System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPFS	Inter-Planetary File System
ISO	International Organization for Standardization
ITL	Information Technology Laboratory
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Token
KYC	Know Your Customer
NFT	Non-Fungible Token
NIST	National Institute of Standards and Technology



NIST-IR	National Institute of Standards and Technology Internal Report
NIST SP	National Institute of Standards and Technology Special Publication
PII	Personally-Identifiable Information
QR	Quick Response
RBFT	Redundant Byzantine Fault Tolerance
RFC	Request For Comments
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SSO	Single Sign-On
SSI	Self-Sovereign Identity
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XDI	eXtensible Data Interchange
ZK	Zero-Knowledge
ZKP	Zero-Knowledge Protocol

1530

## Appendix B—Glossary

Airdrop	A distribution of digital tokens to a list of blockchain addresses.
Asymmetric-Key Cryptography	A cryptographic system where users have a private key that is kept secret and used to generate a public key (which is freely provided to others). Users can digitally sign data with their private key and the resulting signature can be verified by anyone using the corresponding public key. Also known as Public-key cryptography. [11]
Authentication	Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
Consensus Model	<p>A process to achieve agreement within a distributed system on the valid state.</p> <p>Also known as a consensus algorithm, consensus mechanism, consensus method. [11]</p>
Curation Market	A token-based organization model that aims at incentivizing and coordinating market participants around the curation of some information. <i>Term introduced by Simon de la Rouviere.</i>
Cryptocurrency	<p>A digital asset/credit/unit within the system, which is cryptographically sent from one blockchain network user to another. In the case of cryptocurrency creation (such as the reward for mining), the publishing node includes a transaction sending the newly created cryptocurrency to one or more blockchain network users.</p> <p>These assets are transferred from one user to another by using digital signatures with asymmetric-key pairs. [11]</p>
Cryptographic Hash Function	<p>A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties:</p> <ol style="list-style-type: none"><li>1. (<i>Preimage resistant</i>) It is computationally infeasible to compute the correct input value given some output value (the hash function is “one way”).</li><li>2. (<i>Second preimage resistant</i>) One cannot find an input that hashes to a specific output.</li><li>3. (<i>Collision resistant</i>) It is computationally infeasible to find any two distinct inputs that map to the same output. [11]</li></ol>
Decentralized Application	An application with self-enforceable backend code running on a decentralized ledger rather than a centralized server (it can rely on a set of smart contracts). Also known as “dapp”.
Decentralized Autonomous	A system that is not controlled by a single entity or leader, and that,

Organization	instead, uses on-chain registries and logic to establish some form of self-sustainable organizational structure (e.g., through market incentives, network effects, and protocol designs).
Factory Smart Contract	A smart contract that creates, and sometimes, manages other smart contracts.
Hash	The output of a hash function (e.g., hash(data) = digest). Also known as a message digest, digest, hash digest, or hash value. [11]
JSON Web Token	A JSON Web Token (JWT) is a data exchanged format comprised of a header, a payload, and a signature where the header and the payload take the form of JSON objects. They are encoded and concatenated with the aggregate being signed to generate a signature. The standard was introduced by RFC 7519 from the IETF [69].
Linked Data	A method for interconnecting data structures to promote interpretability. <i>Term introduced by Tim Berners-Lee.</i>
Merkle Airdrop	A scheme to distribute the entitlement to redeem a digital token to a list of blockchain addresses in a single transaction rather than distributing the tokens themselves in a batch of transactions as in a standard airdrop. The list must be available to the participants so that they can build the proof needed to redeem the token (called Merkle proof, as it relies on a Merkle tree).
Merkle Tree	A data structure where the data is hashed and combined until there is a singular root hash that represents the entire structure. [11]
Mintable	Refers to the ability of a digital token to be created.
Node	An individual system within the blockchain network. [11]
Non-Fungible	Refers to something that is not replaceable or interchangeable.
Off-Chain	Refers to data that is stored, or a process that is implemented and executed, outside of any blockchain system.
On-Chain	Refers to data that is stored, or a process that is implemented and executed, within a blockchain system.
Token	A representation of a particular asset that relies on a blockchain.
Unlinkability	The extent to which a relying party is unable to link a given identifier to other ones a subject may own.
Uniform Resource Identifier	A compact sequence of characters that identifies an abstract or physical resource available on the Internet. [70]