A-SIT

Secure Information Technology Center – Austria

The Role of Verifiable Data Registries in the Verifiable Credential Ecosystem

The Role of Verifiable Data Registries in the Verifiable Credential Ecosystem

Autor:
Edona Fasllija
Mail:edona.fasllija@iaik.tugraz.at

Verifiable data registries are defined in the W3C-VC data model and manage all data related to the creation and verification of verifiable credentials, such as trusted credential schemes, public keys of trusted issuers, revocation registries, etc. One of the advantages of this definition is that it is comprehensive and flexible enough to enable tailored verifiable data registries to manage Verifiable Credential data and to accommodate a whole range of different options; from trusted centralized databases to distributed databases and ledgers, such as blockchains. As part of this project, we explore the different trust registry options for managing Verifiable Credential data and identify the benefits and limitations of the alternative storage systems for different trust frameworks.

1. Introduction

In the quest for improved security, ease of use, and privacy, the development of digital identity solutions has undergone a notable transformation. From the conventional username/password approach to Federated Identity systems [1], and now to user-centric models like Verifiable Credentials, there has been a shift towards empowering users with more control, privacy, and flexibility over their identity information.

The Verifiable Credentials approach allows individuals to manage and control their digital attestations of identity attributes. A Verifiable Credential (VC) is a tamper-evident credential with cryptographically verifiable authorship [2]. This model is typically implemented through a personal and secure "wallet" application [3] that receives, stores, presents, and manages Verifiable Credentials and key material of the user. This wallet can be conveniently installed on personal devices, like smartphones, thus providing easy access for the individual.

A VC ecosystem involves three distinct entities:

- The *User*, i.e. the entity seeking to authenticate their identity claims, sometimes also referred to as the Identity Holder or Credential Subject.
- The *Verifier*, i.e. the entity requesting the user to prove specific identity claims (also known as the Service Provider or Relying Party).
- The *Issuer,* i.e. the entity providing the user with proof of the validity of their claims (also known as the Identity Provider).

Specifically, the Issuer provides the User with a *Verifiable Credential (VC)*. This digital credential contains a set of claims or attributes about the User and *cryptographic proof of integrity* in the form of the Issuer's digital signature. The user can then present this Verifiable Credential via a *Verifiable Presentation* (VP) to any Verifier that requests specific attributes listed in one or more of the user's credentials. Verifiable Presentations are derived from the Verifiable Credential in a way that discloses the minimum amount of information required to meet a purpose through *selective disclosure* mechanisms.

Apart from verifying a statement about the subject of the Verifiable Credential (VC), the verifier should ascertain the following from the provided VC to decide to accept those credentials:

(i) the Issuer of the credential,

(ii) the integrity of the VC since its issuance, and

(iii) the validity of the VC, ensuring it has not expired or been revoked.

The Verifier will then use *publicly accessible information* stored in *Verifiable Data Registries* [2] *(VDRs)* to confirm the validity of the proof of the Verifiable Credential without needing to contact the Issuer. Upon successful verification, the Verifier can confirm the accuracy of the claims within the Verifiable Credential.

The Verifiable Credentials Data Model v2.0 [2] represents an open standard for digital credentials formatting, guaranteeing cryptographic security, privacy preservation, and machine verifiability. However, it is important to note that verifiable credentials extend beyond the W3C VC Data model and encompass credentials expressed via other data models as well, such as the Mobile Driver's License (mDL) [4].

User-centric identity paradigms heavily rely on the concept of Verifiable Data Registries. They serve as a robust and reliable source of data, which offers an element of trust in the system. These registries facilitate the creation and validation process of credentials in the VC ecosystem by providing access to a system of records that helps one party decide that an entity is trustworthy. This process allows for more secure and trustworthy information exchanges, which is a fundamental aspect of the VC identity paradigm.

1.1. Trust Model

The architecture of the Verifiable Credentials ecosystem represents a significant paradigm shift, placing the User at the center of interactions between the Verifier and the Credential Issuer. This paradigm shift is also giving rise to new trust frameworks.

It offers enhanced privacy, portability, and control for Users by allowing them to:

i. Present credentials to Verifiers without requiring the Verifiers to contact the credential issuer.

ii. Use Identifiers that are not assigned by a specific third-party provider.

iii. Manage trust relationships with Verifiers independently from Credential Issuers [5].

It is crucial to recognize that realizing the full benefits of verifiable credentials hinges on effectively establishing *trust* within the ecosystem. Merely verifying the Credential Issuer's signature on a credential might not suffice for verifiers to accept the credential and provide access to services for users. The Verifier needs to ascertain that the Issuer of the Credential is trusted to issue the specific credential type in that particular domain. The user (or the wallet application) needs to determine whether the Verifier is trusted to request the particular attributes during the Verifiable Presentation process. Another question of trust is whether the User can utilize any Wallet application to manage their verifiable credentials. Trust Frameworks provide mechanisms that facilitate these trust decisions based on the trust relationship between the User and the Verifier (or Relying Party) and between Verifiers and Wallets.

Establishing these trust relationships entails regulatory or contractual agreements in addition to technical interoperability. Verifiable Data Registries serve the purpose of capturing and serving these agreements.

Alternatively called *Trust Registries*, VDRs offer a secure and reliable foundation for the entities within the system to establish trust and engage with one another. They primarily store the trust anchors for the relevant entities of the ecosystem. The trust anchor is generally the entity identifier and associated public key used to verify signatures created by that entry. Moreover, they store data about credential templates and verification policies, and a list of entities authorized for issuing and verifying credentials within the ecosystem. Furthermore, these registries may also include records specifying what Issuers are authorized to issue what types of Verifiable Credentials, but also what Verifiers are authorized to request what types of Verifiable Presentations.

The W3C VC Data Model describes the trust model of a VC ecosystem based on a trust triangle (depicted in Figure 1) that describes the interactions between the Issuer, the User, and the Verifier. In this model, the Verifier trusts the Issuer to issue the credential that it received, which is established through a proof that the Issuer generated the credential. The User and Verifier trust the Issuer to issue true credentials about the subject and revoke them quickly when necessary. *All entities* trust the Verifiable Data Registry to be *tamper-evident* and to represent a *correct* record of which data is controlled by which entities [2].

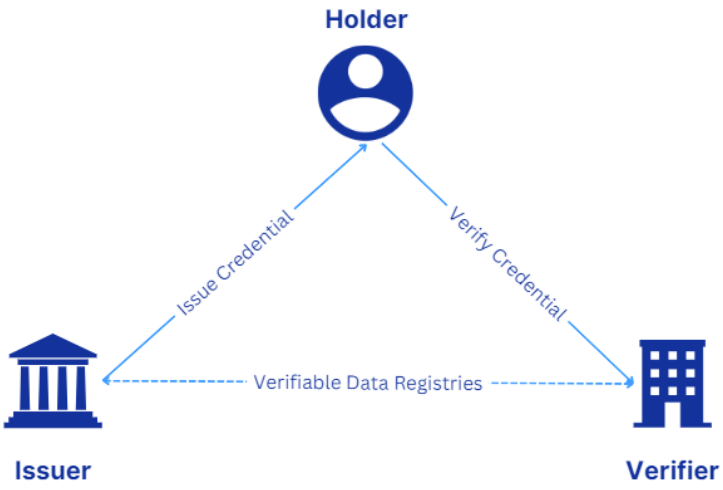

Figure 1 Trust Triangle in the Verifiable Credential Ecosystem

VDRs enable the technical trust required for the interactions among the entities. For the Verifiers to ascertain the origin of the Verifiable Credential, they need to check whether the Issuer of the VC is recognized as trusted within the domain, along with the source of this recognition. In this context, the verifier has an indirect trust relationship with the issuer. Instead of direct communication between the issuer and the verifier, the issuer maintains information about its public keys in the Trust Registries readily accessible by the Verifier. During the verification process, the proof within the Verifiable Credential can be authenticated using the public key stored in the registries. As the VDR is tamper-resistant, the verifier places trust in the integrity of the stored information belonging to the issuer and ensures the public key remains unaltered, thereby completing the trust triangle as shown in Figure 1.

1.2. Trust Frameworks

In the VC ecosystem, Credential Issuers, Users, and Verifiers require identifiers. Typically, these identifiers are also used to access a cryptographic public key, which is necessary for verifying the credential signature. Verifiable Credentials provide more control for the User by enabling them to utilize identifiers that are not necessarily assigned a specific third-party authority (Identity Provider). The concept of decentralizing identifiers, often associated with W3C Decentralized Identifiers (DIDs), is prominent, although alternative identifier types besides DIDs are also applicable. Available options include HTTPS URLs, JWKs, X.509 certificates, and more.

Verifying the validity of a Verifiable Credential entails understanding the trust framework on which it is based. The trust framework is needed to address the trustworthiness of various parties involved in the VC Ecosystem. Based on the trust model and the entity identifier type they adopt; verifiable credential ecosystems can be categorized into two main groups: decentralized or centralized.

Systems in the former category utilize a decentralized public key infrastructure (DPKI) [6] and may employ technologies like distributed ledgers or blockchains to generate and handle decentralized identifiers (DIDs) in a decentralized fashion. In these systems, the identifiers are globally unique and anchored to a Verifiable Data Registry which supports the discovery and the verification process. DIDs are designed to operate independently of centralized identity providers and certificate authorities. The owner of the DID can therefore demonstrate control over it without needing authorization from any other entity.

Conversely, centralized systems rely on a centralized public key infrastructure (PKI), depending on a hierarchy of trusted Certification Authorities (CAs) for validation. This involves a trust root issuing certificates to certification authorities, who then issue certificates to users, creating a trust chain from the root to the end certificates.

To enable Users to receive credentials directly from Issuers and present them *directly* to Verifiers, it's crucial to establish a mechanism for Verifiers to access the public keys managed by the Issuers. This is exactly where VDRs come into play. The Verifiers can access the public Keys of the credential Issuers in a trustworthy manner via the VDRs.

1.3. The Role of VDRs

Verifiable data registries play a crucial role in verifiable credential ecosystems for several reasons:

i. Trusted Mediator: A VDR functions as a trusted intermediary or mediator within the VC ecosystem. It is a repository or database that stores and provides access to critical information related to VCs. This includes storing cryptographic public keys, service endpoints, authentication parameters, timestamps, and metadata. The VDR ensures this information is available and accessible to support resolution and verification processes.

ii. Lifecycle Management: the VDR manages and maintains the lifecycle of VCs. It supports the creation, registration, and revocation of VCs by serving as a centralized control or coordination point. For instance, an entity issuing a new VC can use the VDR to ensure proper registration and management of the related information. Likewise, the VDR handles updates or revocations of VCs, reflecting these changes in the system.

iii. Interoperability: The VDRs also enhance interoperability and standardization within the ecosystem. It enforces consistent data formats, validation rules, and data-sharing protocols, ensuring that the stored information adheres to established standards and specifications. This promotes compatibility and seamless integration across different VC issuers, and verifiers.

iv. Auditability: VDRs help make the interactions among the ecosystem entities more transparent and auditable. They support compliance with regulations by enabling the auditability of credential issuance and verification processes.

2. Analysis of Verifiable Data Registries

2.1. Requirements

The requirements for storage solutions of Verifiable Data Registries in the Verifiable Credential ecosystem primarily revolve around security, transparency, and the specific needs of the use case at hand. The requirements are derived from a variety of sources, including standards and guidelines, technical specifications, regulatory requirements, best practices, and use-case requirements. Below we list a non-exhaustive list of some key requirements:

1. Security: The storage solution must be secure to prevent unauthorized access or tampering. This is crucial when storing sensitive data such as identity credentials.
2. Transparency: The solution should be transparent, allowing for easy verification of data. This is particularly important for distributed ledgers or blockchains, where data must be publicly verifiable.
3. Availability: The storage solution should be highly available, ensuring that data can be accessed whenever needed.
4. Suitability for Use Case: The choice between a trusted database, a centralized ledger, or a distributed ledger largely depends on the specific requirements of the use case. Factors to consider include the need for centralization or decentralization, the required speed and efficiency, and the level of trust placed in the operator of the registry.
5. Scalability: As the volume of identity data grows, the storage solution should be able to scale efficiently to accommodate this increase.
6. Data Integrity: The solution should ensure data integrity so that the data remains accurate and consistent over its entire lifecycle.
7. Tamper-evident: The storage solution should be tamper-evident, meaning that any alteration of the data should be easily detectable.
8. Immutability: For some use-cases, it may be necessary that the stored data is tamper-proof. This is a key feature of ledgers, ensuring that once a transaction is recorded, it cannot be altered. This feature is fundamental to enable the secure sharing of information such as DID Documents.
9. Decentralization: In some cases, a decentralized solution may be necessary to avoid a single point of failure and to ensure that no single entity has control over all the data.

10. Interoperability: The solution should be interoperable with other systems in the VC ecosystem. This includes compatibility with various credential formats, issuer and verifier systems, and other data registries.

It is important to note that these requirements are not exhaustive or mandatory and should be tailored to fit the specific needs of the application.

2.2. Functionality

Verifiable Credential Ecosystems can differ in terms of openness to participation. Depending on the ecosystem, specific permissions or certifications may be needed for entities such as wallet application providers, credential issuers, and verifiers to join. In contrast, other ecosystems might be entirely open to all entities to participate. To encapsulate these certifications, Verifiable Data Registries can take on the following functionalities:

Trusted Issuer Registry: A Trusted Issuer Registry is a type of Verifiable Data Registry that holds the public keys of Issuers. It serves as a reliable source to verify the authenticity of credentials issued by these entities. The registry ensures that the issuer is recognized and trusted within the ecosystem, contributing to the safety and integrity of the VC framework. Examples of Trusted Issuer Registries are present in the European Blockchain Service Infrastructure (EBSI) [7] or Trust-service Status List (TSL) following ETSI TS 119 612 format that can be exposed via a well-known URL or Domain Name System PTR Records (DNS PTR Records), like in the TRAIN [8] model.

Trusted Verifier Registry: Verifiers play a central role in the VC ecosystem as they are responsible for verifying the claims made in a credential. A Trusted Verifier Registry is a type of Verifiable Data Registry that contains the public keys of verifiers. This allows for the verification of a verifier's identity during the credential verification process, ensuring that the entity verifying the credentials is also trustworthy and recognized within the ecosystem.

Trusted Wallet Provider Registry: This registry maintains a list of verified and trustworthy wallet providers. The registry ensures that these wallet providers adhere to specific standards and protocols for security, privacy, and interoperability.

Credential Schema Registry: The Credential Schema Registry is a type of Verifiable Data Registry that stores the various schemas for verifiable credentials. These schemas define the structure and format of the credentials, including the types of claims that can be made and the data types of these claims. The Credential Schema Registry is essential for standardization across the VC ecosystem, enabling different parties to issue, verify, and understand credentials consistently.

Revocation Registry: The Revocation Registry is a type of Verifiable Data Registry that tracks which credentials have been revoked by the Issuer. This is crucial because it allows verifiers to check whether a presented credential is still valid or if it has been revoked by the issuer. This ensures the currency and relevance of the credentials, contributing to the overall trust in the VC ecosystem.

Trusted Accreditation Organization Registry: This Registry stores the list of organizations that are authorized to accredit the Trusted Issuers and the Trusted Verifiers of the ecosystem.

Public Keys Registry: Registry that stores and manages the public keys of entities in Verifiable Credential Systems. In decentralized user-centric identity systems. For example, the Issuer Public Key Registry allows the Verifiers to check the authenticity of the credential by comparing the signature on the credential with the stored public key of the Issuer.

Each time a User presents a credential to a Verifier, the Verifier consults these Trust Registries. Depending on the application-specific requirements, for the Credential to be considered successfully validated, the following checks can be executed [9] :

- The format of the credential complies with a schema that is listed in the *Credential Schema Registry*.
- The Credential is issued by an Issuer present in the *Trusted Issuers Registry*.
- The signature of the Issuer present in the Credential can be verified with the Issuer's corresponding public key listed in the *Public Keys Registry*.
- The Credential has not been revoked and is listed as valid in the *Revocation Registry*.
- The credential is stored in a certified and secure wallet, complying with necessary security standards.
- The Issuer is authorized by an Accreditation Organization to issue credentials using the corresponding schema.

2.3. Storage Implementation Solutions

The Verifiable Credentials standards do not provide explicit specifications on how the VDR should be implemented. Most of the proposed methods rely on Distributed Ledger Technologies (DLTs) due to their inherent design that ensures tamper-proofing and traceability. However, alternative approaches have also been deployed. Storage implementation solutions for VDRs can be generally categorized into centralized and decentralized solutions. Below we present some examples for each category:

2.3.1. Trusted Databases

Trusted databases, with their strong security protocols and authentication mechanisms, are a classic option for storing sensitive data such as identity credentials. They can be centrally managed, meaning that one or more organizations control how data on the registry is accessed, maintained, and distributed. However, trusted databases can also be maintained across multiple systems or even run on a cloud. The trust model for verifiable credentials only requires the verifiable data registry to be tamper-evident and to be a correct record of which data is controlled by which entities. Therefore, if a secure, centralized, or distributed database meets these criteria, it can serve as a verifiable data registry in the VC ecosystem.

Trusted databases typically offer fast data retrieval speeds and can handle large amounts of data, making them highly scalable. They use strong security protocols to maintain data integrity. However, due to their centralized nature, they can be vulnerable to attacks targeting their central point of failure. Additionally, they may not offer the same level of transparency as some decentralized options, which can be a consideration for some use cases.

2.3.2. Centralized Ledgers

Centralized ledgers, such as Amazon QLDB [10], are managed and operated by a central authority. Users of these ledgers trust that the contents haven't been tampered with by trusting the operator of the ledger. These ledgers have some critical qualities that make them more appropriate for enterprise-grade verifiable credentials services. For instance, centralized ledgers like QLDB don't require a consensus mechanism like distributed ledgers do, which makes them faster, more efficient, and often cheaper to use as a verifiable data registry.

2.3.3. Distributed Ledgers

Distributed ledgers, on the other hand, contain data that is replicated, shared, and synchronized across multiple systems or entities. Blockchain is a well-known example of a distributed ledger. The contents of

the ledger are verified collectively by all entities running a node, making it challenging for data to be tampered with. Depending on the distributed ledger technology used, entries can be publicly verified, making it easy for anyone to check if a credential has been altered. This feature is particularly important for the secure exchange of information related to DIDs [11].

2.3.4. Distributed Hash Tables (DHT)

DHT is a decentralized distributed system that provides a lookup service similar to a hash table. Examples include the BitTorrent protocol or the Distributed Hash Table network used by the IPFS (InterPlanetary File System).
DHTs are decentralized systems that are efficient in handling large amounts of data and can quickly locate the data across the network. However, they can be slower than centralized databases due to the need to search across multiple nodes. This can be a trade-off for the increased resilience to faults and failures that these distributed systems offer.

2.3.5. Distributed File Systems:

Distributed file systems such as Hadoop Distributed File System (HDFS) [12] are designed to be highly scalable and reliable. They distribute data across multiple nodes, which can increase data redundancy and fault tolerance. However, due to the distributed nature of these systems, data retrieval can be slower than on centralized systems.

2.3.6. Content Delivery Networks (CDNs):

CDNs [13] such as Akamai or Cloudflare can handle large volumes of data and provide fast delivery of content. Their global network of servers works together to provide high availability and performance. However, they can also be prone to centralized attacks and may not provide the same level of control over data as some other options.

In conclusion, while distributed ledgers and blockchains offer robust security and transparency, they can be less efficient due to the consensus process required to verify data. Therefore, they are typically chosen for use cases where data must be publicly available, and a decentralized authority is a requirement.

The choice between a centralized or a decentralized storage solution as a trust registry in the VC ecosystem depends largely on the specific requirements of the use case at hand. Each has its unique strengths and potential drawbacks, and the decision should be made based on factors such as the need for centralization or decentralization, the required speed and efficiency, and the level of trust placed in the operator of the registry.

3. Revocation of Verifiable Credentials

Despite the numerous advantages that VCs offer, several technical challenges remain to be tackled. One challenge is the design of revocation mechanisms for VCs. Revocation is an essential process that allows issuers to invalidate a credential that they had previously issued. This might be necessary in certain instances, where an individual's identity information contained in a credential is no longer accurate or if it has been deemed to have been mistakenly or fraudulently issued.

Verifiers need to confirm both the *status* and *validity* of the VCs to uphold trust among the Issuer, User, and Verifier. Without this revocation framework, the likelihood of fraudulent behavior rises, undermining trust and credibility across the VC ecosystem. Despite the increasing interest in VC, efficient revocation methods are still in the early stages of development, as also indicated by the absence of a specific W3C standard.

In the W3C specification, the revocation status of a credential is expressed in the Credential Status property of a VC. During the credential issuance process, the Issuer can include a reference to an external (publicly available) revocation registry.

However, revocation of verifiable credentials can pose certain privacy threats. These include traceability, linkability, exposure of personal information, and unauthorized revocations. To mitigate these potential threats, it is crucial to design revocation processes that prioritize privacy. This can be achieved by designing a Revocation Registry that allows the Credential Issuers to manage the credentials they issue, while also preserving the anonymity of the VC holders.

3.1. Design Considerations for Revocation of Verifiable Credentials

Revocation of verifiable credentials, while essential for maintaining the integrity and trustworthiness of a system, can also pose certain privacy threats if not managed carefully. Potential privacy threats include:

Traceability: If the revocation process involves checking a credential against a publicly accessible revocation list, it can potentially allow tracking of the credential holder's activity, thereby infringing upon their privacy. Querying or accessing the revocation status records on the Revocation Registries must not allow other entities to track the use of the VC.

Linkability: If a unique identifier is used to revoke a credential, it could provide a way for different parties to link together separate activities or transactions of the credential holder. The revocation status record stored on the Revocation Registries must not serve as a globally unique identifier or correlator of the User.

Exposure of Personal Information: In some revocation methods, personal information may be revealed to the verifier or to other parties involved in the verification process. Accessing the revocation or status record of a VC must not reveal any other information about the User.

Unauthorized Revocations: In user-centric models where the credential holder has the power to revoke their own credentials, there is a risk of unauthorized or fraudulent revocations if robust security measures are not in place.

Performance considerations are also of uttermost importance when designing a revocation mechanism. One key factor is the bandwidth and processing load it imposes on both the client side that retrieved revocation information and the server side.

3.2. Existing Revocation Mechanisms

Revocation mechanisms can be categorized as centralized or decentralized depending on whether a central entity that manages the Revocation Registries is involved or not.

Centralized revocation: This approach involves a central entity managing the revocation process by maintaining a list of invalidated credentials. Verifiers check this list before accepting a credential to ensure it hasn't been revoked. An example of this method is Let's Revoke [14]. However, these solutions may raise privacy concerns due to unique credential identifiers, which can allow Verifiers to link the Holder to the specific credential.

Decentralized revocation: Decentralized revocation mechanisms often use distributed ledger technologies (DLTs) or smart contracts, typically storing a revocation list on a blockchain. This category also includes Peer-to-peer (P2P) networks. Given that ledgers are append-only, means that it's always it is possible to validate the credential status at a specific point in time. Contrary to a centralized revocation list, it is difficult to backdate a status change. However, these approaches can also raise privacy concerns.

The complexity of revocation for VCs means that a one-size-fits-all solution is hard to find. As with the storage implementation solutions, the revocation strategy has to be chosen in accordance with the use case requirements. Below we list alternative implementations of revocation mechanisms for VCs.

3.2.1. List-Based Revocation

List-based revocation methods use simple allow or blocklists. These lists are publicly accessible or can be queried without restrictions via an interface. An example is the Certificate Revocation List (CRL) used in the Web PKI.

Other list-based revocation mechanisms hide the allow or blocklists. A trusted party manages the list and controls access. Group signatures can be employed, with a group manager overseeing the list and ensuring anonymity.

Another option for List-based Revocation Registries is compressed lists. Compressed lists reduce the size of the information, making storage, downloading, and querying more efficient. Techniques such as Bloom filters or bit-arrays can be used to implement these methods. Bitstring Status List 1.0 [15] from W3C (World Wide Web Consortium) is a specification for managing and communicating the status of entities in a compact and efficient manner using bitstrings.

Since these lists contain revocation statutes for many credentials, credential subjects can achieve "herd privacy" with regards to the Revocation Registry maintaining entity. However, some use cases may require revocation mechanisms with an increased level of privacy for credential subjects.

3.2.2. Accumulator-based Revocation

Some revocation mechanisms leverage cryptographic techniques such as accumulator-based revocation or zero-knowledge proofs to prove the non-revocation status of a credential without revealing the specific credential itself. This enables efficient revocation checking without compromising privacy.

An accumulator is a one-way function that aggregates a large set of items into a single value. Accumulators can be symmetric or asymmetric. Asymmetric accumulators require additional information (known as the witness) for membership verification. Examples include RSA-based, Bilinear-Pairing-based and Merkle tree-based accumulators. Symmetric accumulators like Bloom Filters do not require witness information.

Moreover, accumulators can be categorized based on the proofs they support. Positive accumulators support membership proofs, demonstrating that an item is included in the set. Negative accumulators support non-membership proofs, showing that an item is not included, Finally, Universal accumulators support both membership and non-membership proofs.

4. Conclusion

In summary, Verifiable Data Registries are a cornerstone of the Verifiable Credential ecosystem, facilitating secure and trustworthy information exchanges. They offer a reliable method for verifying the accuracy of identity claims, contributing to the overall integrity and trust in the system.

One of the prominent advantages of the W3C definition of verifiable data registries is its comprehensive nature and flexibility. It is designed in a way that allows the creation of tailored verifiable data registries to handle Verifiable Credential data effectively. Furthermore, it is adaptable enough to accommodate a wide array of different options for data storage. These options can vary from trusted centralized databases, which are managed by a single authority, to distributed databases that distribute the data

across multiple locations for enhanced security and reliability. Other options include ledgers, or blockchains, which use cryptographic methods to secure data transactions and prevent unauthorized changes.

Furthermore, the importance of well-implemented revocation mechanisms cannot be overstated. These mechanisms ensure that the integrity and trustworthiness of the VC system are maintained, even when a VC needs to be invalidated for various reasons. However, the implementation of these mechanisms should be done with care to avoid potential privacy threats such as traceability, linkability, and exposure of personal information.

Ultimately, the goal of a VC system is to provide a secure, efficient, and user-centric means of managing and verifying credentials. Therefore, the choice of the trust model and revocation mechanisms should be guided by these principles.

As part of this project, we delved into the exploration of different trust registry options suitable for managing VC data. This exploration allows us to identify the benefits and limitations of each of these alternative storage systems within different trust frameworks. Understanding these aspects is crucial in the decision-making process when choosing an appropriate storage system for verifiable credentials depending on the specific requirements and constraints of different trust frameworks.

References

[1] J. Camenisch and B. Pfitzmann, "Federated Identity Management," *Security, Privacy, and Trust in Modern Data Management*, pp. 213–238, Jun. 2007, doi: 10.1007/978-3-540-69861-6_15.

[2] "Verifiable Credentials Data Model v2.0." Accessed: Apr. 02, 2024. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/

[3] B. Podgorelec, L. Alber, and T. Zefferer, "What is a (Digital) Identity Wallet? A Systematic Literature Review", doi: 10.1109/COMPSAC54236.2022.00131.

[4] "ISO/IEC 18013-5:2021 - Personal identification — ISO-compliant driving licence — Part 5: Mobile driving licence (mDL) application." Accessed: Apr. 22, 2024. [Online]. Available: https://www.iso.org/standard/69084.html

[5] K. Yasuda, T. Lodderstedt, D. Chadwick, K. Nakamura, and J. Vercammen, "OpenID for Verifiable Credentials," 2022, Accessed: May 27, 2024. [Online]. Available: https://www.iso.org/committee/45144.html

[6] A. Papageorgiou, A. Mygiakis, K. Loupos, and T. Krousarlis, "DPKI: A Blockchain-Based Decentralized Public Key Infrastructure System," *GIoTS 2020 - Global Internet of Things Summit, Proceedings*, Jun. 2020, doi: 10.1109/GIOTS49054.2020.9119673.

[7] "Home - EBSI -." Accessed: May 27, 2024. [Online]. Available: https://ec.europa.eu/digital-building-blocks/sites/display/EBSI/Home

[8] “A novel approach to establish trust in verifiable credential issuers in Self-sovereign identity ecosystems using TRAIN.” Accessed: May 27, 2024. [Online]. Available: https://dl.gi.de/items/ae2dd3f9-d34b-48a1-9d67-2ee5db2e8df3

[9] “The Role of Trust Registries in an SSI Ecosystem | Gataca.” Accessed: Apr. 02, 2024. [Online]. Available: https://gataca.io/blog/the-role-of-trust-registries-in-an-ssi-ecosystem/

[10] “CRUD Database - Amazon Quantum Ledger Database (QLDB) - AWS.” Accessed: Apr. 02, 2024. [Online]. Available: https://aws.amazon.com/qldb/

[11] P. Dunphy and F. A. P. Petitcolas, “A First Look at Identity Management Schemes on the Blockchain *,” 2018.

[12] “HDFS Architecture Guide.” Accessed: May 27, 2024. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[13] “What Is a CDN (Content Delivery Network)? | How Do CDNs Work? | Akamai.” Accessed: May 27, 2024. [Online]. Available: https://www.akamai.com/glossary/what-is-a-cdn

[14] “Let’s Revoke: Scalable Global Certificate Revocation - NDSS Symposium.” Accessed: May 27, 2024. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/lets-revoke-scalable-global-certificate-revocation/

[15] “Bitstring Status List v1.0.” Accessed: May 27, 2024. [Online]. Available: https://www.w3.org/TR/vc-bitstring-status-list/