# LACCHAIN ID FRAMEWORK

A SET OF RECOMMENDATIONS FOR BLOCKCHAIN-BASED INTEROPERABLE, PRIVACY-PRESERVING, REGULATORY COMPLIANT, SECURE, AND STANDARDIZED DIGITAL IDENTIFIERS, CREDENTIALS, AND WALLETS

IDB IDB | LAB LACCHAIN

# LACCHAIN ID FRAMEWORK

## A SET OF RECOMMENDATIONS FOR BLOCKCHAIN-BASED INTEROPERABLE, PRIVACY-PRESERVING, REGULATORY COMPLIANT, SECURE, AND STANDARDIZED DIGITAL IDENTIFIERS, CREDENTIALS, AND WALLETS


**Authors:**

Marcos Allende, Technical Leader of LACChain and IT Specialist in Blockchain, SSI, and Quantum Technologies at IDB, USA

Sergio Cerón, Blockchain Lead Architect, LACChain, Mexico.

**Supervisor:**

Alejandro Pardo, Leader of LACChain and Principal Specialist at IDB, USA.

Marcelo da Silva, IT Principal Specialist at IDB, USA



**Design:**

.Puntoaparte Editores


IDB IDB | LAB LACCHAIN

This paper has been elaborated by the LACChain ID working group, which includes international experts in blockchain, digital identity, and SSI that participate in the working group in representation of member entities of the LACChain Global Alliance (https://www.lacchain.net/alliance). All the collaborators are acknowleged at the end of the document. This aim of this framework is not to enforce any of the recommendations made over the document, but rather to serve as a tool for those entities that might find it useful for their implementations of SSI solutions.

# TABLE OF CONTENTS

# ACRONYMS

| | |
|---|---|
| **CA** | Certificate Authority |
| **CL** | Circular Layout |
| **CRL** | Certificate Revocation List |
| **DOS** | Denial of Service |
| **DID** | Decentralized Identifier |
| **DLT** | Distributed Ledger Technology |
| **DNS** | Domain Name System |
| **eIDAS** | Electronic IDentification, Authentication and Trust Services (European Union) |
| **EBSI** | European Blockchain Services Infrastructure |
| **ECDSA** | Elliptic Curve Digital Signature Algorithm |
| **EIP-1812** | Ethereum Verifiable Claims |
| **EIP-712** | Ethereum Typed Structured Data Hashing and Signing |
| **ESSIF** | European Self-Sovereign Identity Framework |
| **GDPR** | General Data Protection Regulation (European Union) |
| **GPG** | Command line tool with features for easy integration with other applications |
| **IANA** | Internet Assigned Numbers Authority |
| **ICAO** | Civil Aviation Organization |
| **IPFS** | InterPlanetary File System |
| **JWS** | JSON Web Signature |
| **LOA** | Level Of Assurance |
| **MRA** | Mutual Recognition Agreements |
| **NGOs** | Non-governmental Organization |
| **NIST** | National Institute of Standards and Technology |
| **OIDC** | OpenID Connect |
| **OSCP** | Offensive Security Certified Professional |
| **OCSP** | Online Certificate Status Protocol |
| **PII** | Personally Identifiable Information |
| **PKD** | Public Key Directory |
| **PKI** | Public Key Infrastructure |
| **RIR** | Regional Internet Registries |
| **RSA** | Rivest–Shamir–Adleman is a public-key algorithm |
| **SHA-256** | Secure Hash Algorithm 256 bits SSH - Secure Shell Protocol |
| **SSI** | Self-Sovereign Identity |
| **TL** | Marcos please review and add it |
| **Uint** | U - unsigned (meaning this type can only represent positive integers, not positive and negative integers) INT - integer |
| **URI** | Uniform Resource Identifier |
| **UUID** | Universally Unique Identifier |
| **VC** | Verifiable Credential |
| **W3C** | World Wide Web Consortium |
| **WHO** | World Health Organization |
| **ZKP** | Zero-knowledge Proof |

An identity can be attributed to a person, an organization, a thing, or a process, and refers to anything that characterizes them. For individuals, some elements of identity may include physical features, gender, biometric information, experiences, belongings, and titles. The way individuals can be uniquely identified and prove, or authenticate, their identity to others has evolved significantly over the past few decades. For example, photo IDs began being broadly adopted during the 20th century, and the shape and size of ID cards were standardized in 1985 by ISO/IEC 7810. Today, people in most countries can obtain a government-issued ID card or passport for national international, and sometimes regional identification and authentication.

However, there are at least two big challenges related to identification of individuals that have not yet been successfully addressed. First, more that 1 billion individuals around the world (approximately 14% of the population) lack any form of proof of identity. Second, globally, there is a lack of a robust means for providing electronic ID authentication in order to access and receive digital services, which has become especially urgent in light of the COVID-19 pandemic. Although some countries have already enabled means for digital identification and authentication, such as chip cards with PIN, the usability, user experience, level of assurance, and availability of e-services are currently far from ideal.

Addressing these challenges is not something that a single person, institution, or even government can achieve on their own. A joint effort is required. New standards to be developed by organizations of standards with an end-user approach would be necessary. Further, governments could adopt those standards and leverage emerging technologies to provide better means for digital identification and authentication. Finally, private entities could enable user-friendly mechanisms for individuals' digital identification and authentication to e-services. These steps have already been set in motion.

The traditional schemes for digital identity such as centralized and third-party providers, and federations, have been very successful in authenticating individuals for Internet websites and platforms, and corporate sites. However, when it comes to electronic services provided by administrations such as e-health, e-justice, e-law, e-tax, e-voting, e-police, e-land-registry, among others, as well as services provided by private institutions that require high levels of assurance in the identification and authentication of individuals, such as e-banking, these solutions have not proliferated nor become broadly adopted.

Collaborative international efforts have been put in place to design and develop new solutions that aspire to be more successful in enabling means for the identification and authentication of individuals across the globe that are more user centric as well

as inclusive, protective, and scalable. A set of standards, protocols, technologies, and ideas that are gaining traction have been taxonomized under the name Self-Sovereign Identity (SSI). Some people have claimed that the aim of SSI is to erase governments and authorities and enable self-issuance of identity credentials. Thus, many people have reacted against SSI schemes, despite not fully understanding the technology and the potential behind them. On the contrary, we believe SSI actually has the potential to be very useful in helping governments and authorities address the two challenges we mentioned at previously: the 1 billion people without means of ID authentication and the lack of robust solutions for digital identity. Given these challenges, SSI has tremendous potential for social, economic, and financial inclusion.

SSI does not currently have an agreed upon definition. To us, SSI has, in its core, two main standards from the W3C, which include the verifiable credentials (VCs) and the decentralized identifiers (DIDs). The VC standard consists of a data model that is particularly suitable for certifying certain attributes of a person, organization, thing, or process, enabling real time digital verification against several sources, varying from an OSCP to a smart contract. They also allow for generation of verifiable presentations from one or several credentials in which the subject chooses information to disclose. VCs are also designed to enable every type of delegation.

The DID standard proposes a new type of identifier which is a perfect complement to the VC standard. Also, the DID standard is designed to associate these new identifiers with different types of verification methods that include cryptography and biometrics. It enables key rotation, delegation, and recovery. Among many other benefits, DIDs are ideal for erasing undesired associations between various electronic interactions in the same or different contexts. Some outstanding use cases of VCs and DIDs are digital diplomas, passports, government-issued IDs, and vaccination certificates.

Standards for data models such as the VC and the DID are very important, but a digital identity scheme requires several other components as well, including technological tools and trust frameworks. There are two technological tools that fit perfectly with the DID and VC standards: digital wallets and blockchain networks. Rather than requiring individuals to carry a physical card with a chip, remember a password, and use a device to connect to a computer to authenticate to electronic services, it is more natural, user-friendly, and safe to allow individuals to manage their digital identifiers and digital certificates with an application in a personal device connected to the internet, such as a smartphone. This more efficient system is referred to as a digital wallet in the context of SSI. Regarding blockchain, we explained previously that VCs can be verified against

centralized registries and that DIDs can be resolved against centralized databases. However, the potential to use decentralized, public, and reliable ledgers to store the proofs of VCs and resolve DIDs opens a wider range of possibilities.

Naturally, public administrations should not stop maintaining and enabling access to centralized databases just because smart contracts can be used in decentralized registries. Furthermore, governments could even decide to refuse the use of blockchain networks at all. However, governments willing to explore the benefits of blockchain could decide to enable and manage a smart contract deployed in a blockchain network that allows to verify government-issued digital certificates against any decentralized blockchain node, instead against to a central registry maintained by the government. Other entities such as NGOs, multilaterals, universities, schools, hospitals, pharmacies, supermarkets, and a long list of other institutions already issue or have the potential to issue certificates or credentials. However, they might not even want or be able to maintain centralized infrastructures to allow the verification of digital credentials issued by them. Therefore, being able to just register the proofs in a decentralized registry -such as a blockchain network- that they do not have the responsibility to maintain and protect can be of much value to them as issuers. With blockchain ledgers, digital certificates scalability can be more efficient and secure.

There is one more element, beyond standards and technologies, which plays an essential when addressing an identity scheme: trust frameworks. Trust frameworks are agreements that enable recognition, enforceability, regulatory compliance, and non-repudiation of digital IDs and digital certificates between entities, and can be public, private, or a combination of both. Typically, through national frameworks, governments regulate the use of electronic signatures and certificates and then define the entities responsible for the issuance of the government-based IDs and certificates. Regional frameworks, such as the eIDAS regulation in the European Union, allow recognition and non-repudiation at an international level. SSI is not intended to replace regulation of government-based IDs and certificates. Instead, the SSI scheme encompasses standards and technologies that are suitable to the new strategies that national governments will use to modernize identity systems in order to turn them into safe and scalable  user-friendly solutions.

In 2020, we published a book on the topic of SSI, which reviewed various digital identity management systems, analyzed the emerging standards and technologies, outlined steps for implementation, and reflected on predicted challenges and opportunities. We also presented the feasibility of implementing these solutions in Latin America and the Caribbean specifically based on current regulatory frameworks[1]. In this paper, we aim to complement the technological analysis from previous SSI book by proposing a framework consisting of a set of recommendations for the development of SSI-based solutions,

including digital diplomas issued by academic institutions to vaccination credentials and government-issued IDs. This framework, called the LACChain ID Framework, has been developed by the identity working-group of LACChain, which is the global alliance for the development of the blockchain ecosystem in Latin America and the Caribbean. The LACChain ID Framework is already being used by various entities within the LACChain ecosystem. Additionally, in this paper, we will also address matters related to regulation of trust frameworks, which is essential for actualization of the outlined technological proposals.

## Requirements Notation and Conventions

The terms "SHALL" and "SHALL NOT" indicate requirements that must be followed strictly in order to conform to the publication and from which no deviation is permitted.

The terms "SHOULD" and "SHOULD NOT" indicate that among several possibilities, one is recommended as particularly suitable, without excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited.

The terms "MAY" and "NEED NOT" indicate a course of action permissible within the limits of the publication.

The terms "CAN" and "CANNOT" indicate a possibility and capability, whether material, physical or causal or, in the negative, the absence of that possibility or capability.

LACCHAIN ID FRAMEWORK

# 1 DECENTRALIZED IDENTIFIERS

IDB IDB | LAB LACCHAIN

A working group with the World Wide Web Consortium (W3C) is currently developing the Decentralized Identifiers (DIDs) standard. According to the W3C, a DID is "a new type of identifier that enables verifiable and decentralized digital identities. A DID identifies any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) that the controller of the DID decides it identifies". A DID is a URI that associates a DID subject with a DID document allowing trustable interactions associated with that subject, which contains information about the authentication methods to prove ownership, endpoints, and other attributes of that DID. A person is expected to manage multiple DIDs in order to eliminate undesired associations.

## 1.1. DID Documents

DID documents SHALL be comprised of the following standard elements[1].

- A Uniform Resource Identifier (URI) to uniquely identify terminology and protocols that allow parties to read the DID document
- A DID that identifies the subject of the DID document
- A set of authenticators (i.e., public keys) used for authentication, authorization, and communication mechanisms
- A set of authentication methods used for the DID subject to prove ownership of the DID to another entity
- A set of authorization and delegation methods for allowing other entities to operate on behalf of the DID subject (i.e., holders different from the subject)
- A set of service endpoints to describe where and how to interact with the DID subject

A DID document CAN have the following elements:

- A timestamp for when the document was created
- A timestamp for when the document was last updated
- Cryptographic proof of identity (e.g., digital signature)

Additionally, DID documents SHOULD:

- Contain an element that indicates the DID status (active, suspended, or revoked)

1 In alignment with NIST at https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.01142020.pdf.

```
{
  “@context”: [“https://w3id.org/did/v1”, “https://id.lacchain.net/did/v1”],                    1
  “id”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”,                               2
  “controller”: [“did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”],                     3
  “verificationMethod”: [                                                                       4
     {
       “id”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1”,                     a
       “type”: “Secp256k1VerificationKey2018”,                                                  b
       “controller”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”,                  c
       “publicKeyHex”: “0xadf1702b76419f428014d1386af487b2d8145f83”                              d
     },
     …
     {
       “id”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-3”,
       “type”: “X25519KeyAgreementKey2019”,
       “controller”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”,
       “publicKeyBase58”: “0417372efd731a942f769712d13573f…9d7e47cb5de7b0453341
       dc18472f875fa999a7e93”
     }
  ],
  “authentication”: [                                                                           5
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1”,
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-2”
  ],
  “capabilityInvocation”: [                                                                     6
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1”
  ],
  “capabilityDelegation”: [                                                                     7
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1”
  ],
  “assertionMethod”: [                                                                          8
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1”
  ],
  “keyAgreement”: [                                                                             9
     “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-3”
  ],
  “service”: [                                                                                  10
     {
        “id”:”did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#mailbox”,
        “type”: “VerifiableCredentialService”,
        “serviceEndpoint”: “https://mailbox.lacchain.net”
     }
  ],
  “proof”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...joxNTE2MjM5MDIyfQ.                          11
  SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQss5”,
  “created”: “2021-01-14T16:22:00.028Z”,                                                        12
  “updated”: “2021-01-14T16:22:00.028Z”,                                                        13
  “status”: “active”                                                                            14
}
```

*Figure 1. Example of a DID Document compliant with the LACChain ID Framework.*

And DID documents MAY:

- Live in blockchain networks, which allows creation of public and decentralized DID registries so that DIDs can easily be resolved against an accessible, trusted, and decentralized ledger

The DID document in Figure 1 contains attributes within the W3C standard DID definition and is compliant with the LACChain ID Framework. The attributes contain the following elements:

1. The "@context" - an array of URIs that describes common elements within the DID, as well as the definition of the data type. In figure 1, the "@context"s are the W3C schema of DID v1.0 and the LACChain extension to include the "proof" field described below.
2. The "id" - a unique DID identifier with the following nomenclature: "did:<method>:<network>:<unique_id>". In figure 1, "<method>" is "lac", which represents the LACChain DID method, and "<unique_id>" is an Ethereum address.
3. The "controller" – a collection of DIDs that can make modifications within the DID document. In principle, the "controller" is the same as the "id" of the DID document itself. If this field is omitted, the controller is assumed to be the DID, which is indicated in the field, "id" (see Element 2). If only one controller is specified, the array brackets can be avoided, leaving the "controller" id as a string. This can be used so that an entity different from the subject generating the DID can carry out a permanent delegation by changing the controller to the subject DID, which is different from the DID of the document (see Section 3.3.2.3).
4. The "verificationMethod" field - a set of verification methods that can be used for different purposes. Each verification method must be composed of:

    a. an identifier "id" according to the following nomenclature, "did#vm-<number>", where <number> is an autoincrement integer starting with 1.
    b. the cryptographic algorithm "type", which can be:

        - JsonWebKey2020: JSON Web Signature of 2020 https://w3c-ccg.github.io/lds-jws2020/
        - EcdsaSecp256k1VerificationKey2019: The elliptic curve Secp256k1 https://w3c-ccg.github.io/lds-ecdsa-secp256k1-2019/
        - Ed25519VerificationKey2018: The elliptic curve Curve25519 used commonly by Hyperledger Indy https://w3c-ccg.github.io/lds-ed25519-2018/
        - GpgVerificationKey2020: The GPG Public Key https://gpg.jsld.org/contexts/#GpgVerificationKey2020

- RsaVerificationKey2018: The RSA Signature Suite https://w3c-ccg.github.io/lds-rsa2018/
- X25519KeyAgreementKey2019: The X25519 algorithm used to encrypt information
- EcdsaSecp256k1RecoveryMethod2020: The ECDSA over secp256k1 with encoded recovery bit. https://identity.foundation/EcdsaSecp256k1RecoverySignature2020/#ES256K-R

**c.** the DID of the public key "controller" (in Figure 1, this is the same as the DID, but is not necessary)

**d.** the public key associated with the verification method, which can be:

- publicKeyHex: if the public key is encoded in hexadecimal
- publicKeyBase64: if the public key is encoded in base 64
- publicKeyBase58: if the public key is encoded in base 58
- publicKeyJwk: if the public key is a JSON Web Key that conforms with the RFC7517 (https://www.w3.org/TR/did-core/#bib-rfc7517)
- blockchainAccountId: if the public key is a blockchain account, such as a bitcoin or ethereum address

This subfield "controller" is different from the DID document "controller" (see Element 3). It must be a string and only represents the controller of that specific verification method. This field is not optional because the verification method controller cannot be inferred from the DID document.

**5.** The "authentication" - a set of identifiers, "id", from the "verificationMethod" section, representing those that can be used as authentication methods to prove ownership of the DID and to perform authentication to a service.

**6.** The "capabilityInvocation" - a set of identifiers, "id", from the "verificationMethod" section, representing those that can be used for cryptographic proposes, such as authorization to access a specific service. A specific service could request that the public access key is within this section in addition to being an authentication method.

**7.** The "capabilityDelegation" - a set of identifiers, "id", from the "verificationMethod" section, indicating those that can be used to delegate authority to a third party representing the DID subject in certain processes, such as accessing a specific service or signing a document.

**8.** The "assertionMethod" - a set of identifiers, "id", from the "verificationMethod" section, indicating those that can be used within specific cryptographic processes such as verifying claims in a Verifiable Credential (VC).

9. The "keyAgreement" - a set of identifiers, "id", from the "verificationMethod" section, indicating those that can be used to specify how to encrypt DID subject information, including a VC or creating a secure communication channel with a SSH public key.
10. The "service" - a list of endpoints where the DID can be reached. For example, these endpoints might include verifiable credentials and presentations. The service has three main components:

    a. "id": a unique identifier to the service (in this case a DID)
    b. "type": the type of service
    c. "serviceEndpoint": the URL of the service endpoint

11. The "proof" - a cryptographic proof of the DID document itself in JSON Web Signature (JWS) format.
12. The "created" – used for the timestamp of the DID creation (in Figure 1, this refers to the time it was registered within the smart contract) in ISO string format (YYYY-MM-DDTHH:mm:ssZ).
13. The "updated" - the last modification to the DID in ISO string format (YYYY-MM-DDTHH:mm:ssZ). The possible events that can update this field are:

    a. change the DID ownership
    b. add a verification method
    c. add a new verification relationship (authentication, assertionMethod, capabilityDelegation, invocationDelegation or keyAgreement)
    d. add a new service
    e. change the controller

14. The "status" - the current status of the DID document (active, suspended, revoked, expired)

    Each event is generated when there is a call to the identity registry smart contract that generates a transaction in the blockchain.

## 1.2. DID Methods

The different manifestations of the DID standard are referred to as DID methods. DID methods SHALL comply with the following requirements:

- Contain all elements listed in Section 1.1, including the status of the DID document.

- Have more than one authentication method (i.e., RSA, EC, post-quantum keys, and biometrics)
- Not disclose any personal data or information in the DID documents
- Guarantee privacy and pseudonymity in the use of the DIDs
- If the DID was generated from a private key, avoid the use of the associated public key for authentication, encryption, or signature. Therefore, when possible, destroy the seed of the DID

Additionally, DID methods SHOULD comply with the following requirements:

- Be scalable enough to economically afford the generation of the required amount of DIDs for the specific use case in the chosen network
- Set different functionalities for the different keys, so that some primary keys can be used for authentication, some secondary keys can be used for temporary delegation, and some tertiary keys can be used for retrieving primary and secondary keys
- Register the DIDs in a smart contract with a well-defined governance (an on-chain DID registry) so that issuers or verifiers that intend to resolve a DID can easily find it in a public ledger
- Use quantum-safe cryptography for authentication, encryption, and signature, and include a variety of cryptographic algorithms

Also, DID methods MAY:

- Allow -responsible- use of biometrics (by wallets and applications used by individuals to manage their digital identity information)

We encourage the use of DID Documents in JSON format for document representation and JWS for the generation of on-chain and off-chain signatures.

The LACChain Alliance has developed the "lac" DID method[2], which is based on the ERC-1056 standard and can be used in Ethereum-based networks. This DID method has been developed by the LACChain ID Working Group, and it is a modification of the "ethr" DID method[3] to allow for compliance with the latest version of the W3C standard for DID documents (v. 20210212)[4] and this LACChain ID specification. Specifically:

- The "proof" - a digital signature of the DID Document (in JSON format) by the DID following the JWS standard that guarantees the integrity of the DID
- The identity revocation list. This is not a field in the DID document, but rather an additional smart contract used to register the status of the DID Document so the DID can be revoked entirely by the subject.

The "lac" DID method allows for fully on-chain management of DIDs, using smart contracts (DID Registries[5]) to store the DIDs and the information associated with them. This is a simple way to create and maintain on-chain DID registries where each entity is represented by a generic Ethereum address with the ability to sign any type of data, such as VCs (see Section 2), and has different public keys and endpoints associated.

In comparison with the "ethr" that inspired the "lac" method, "lac" also implements mechanisms for automatic key rotation, delegate assignment, and key recovery. Each one of the contract functions generates events that are later used to reconstruct the document associated with the DID. One of the main advantages of this method is that any blockchain account is valid as a DID without need for a prior registration in the on-chain DID Registry. The DID Registry smart contract only needs to be called on to add a verification method, add an authentication method, or change the controller.

The LACChain Alliance has also made available the LACChain DID Resolver[6] that allows for querying and resolving various DID methods compatible with the different LACChain Networks. The DID Resolver is available for public use at https://resolver.lacchain.net. It is also integrated with the LACChain Mailbox[7], a tool enabled by the LACChain Alliance for the exchange of messages and credentials.

LACCHAIN ID FRAMEWORK

# 2

# VERIFIABLE CREDENTIALS AND PRESENTATIONS

IDB IDB | LAB LACCHAIN

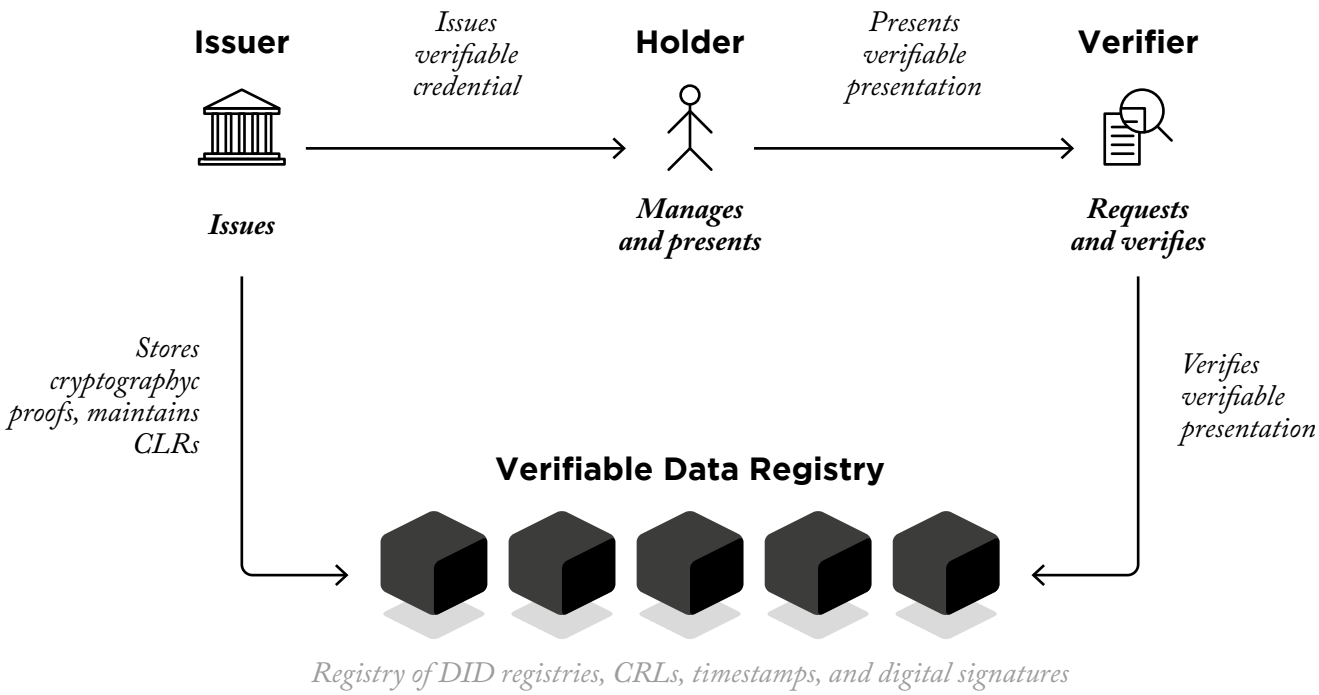


*Figure 2. Verifiable data registry.*

A verifiable credential is a digital file that contains one or more key-value claims (e.g., birth date, name, qualifications, gender, citizenships, etc.) about an entity (the subject), issued by another entity (the issuer), and is verifiable by any entity (the verifier). A Working Group with the W3C has developed the Verifiable Credentials (VC) standard[8]. A VC is structured into claims, metadata, and proofs. Proofs are what make the credential verifiable.

## 2.1. Structure

A VC SHALL contain the following standard elements[2]:

- URI to uniquely identify the credential and/or the subject of the credential (e.g., DIDs)
- URI to identify the issuer (e.g., a DID)
- URI to identify the credential type
- URI to identify terminology and protocols that allow parties to read the credential

2 As sorted by NIST https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.01142020.pdf.

- Cryptographic proof of the issuer
- Claims data or metadata
- Issuance date
- Expiration conditions
- Location of the credential status (e.g., a smart contract in a blockchain network)

Sensitive claims, data, or metadata from the credential SHALL NOT be registered in any blockchain or public registry.

Additionally, a VC SHOULD comply with the following:

- The subject's and issuer's identifiers (e.g., DIDs) can be found and resolved in a blockchain
- Expiration conditions can be automatically obtained from and verified in the VC
- The status of the credential can be verified against a smart contract living in the blockchain. This eliminates the need for external and/or centralized CRL or OCSP
- The issuer enables a protocol for the subject to request revocation

The VC presented in Figure 3 contains the attributes within the W3C standard DID definition and is compliant with the LACChain ID Framework. It contains:

1. The "@context" - an array of URIs to identify terminology and protocols that allow parties to read the VC. The fields in the VC are complementary and mutually exclusive with each other and the definition in the W3C repository. This means that each type of VC specified must have a link to the context defining the standard (in this case the W3C Verifiable Credential Data Model 1.0) and to the context defining the fields that extend the VC, which is usually in the "credentialSubject" field (in this case a LACChain template https://id.lacchain.net/vc/library/{type}/{hash}/{version}) where:

    - {type} is the credential type name
    - {hash} is the SHA-256 of the first version of the credential type
    - {version} is the current version of credential type, in format v1, v2, and so on

2. The identifier "id" of the credential - a Universally Unique Identifier (UUID) that guarantees no duplication, regardless of the system or platform where the VCs are generated or stored. The UUID v4 is a 16-byte number: 32 hexadecimal digits divided into five groups separated by hyphens of the form XX-YY-ZZ-AA-BB which gives a total of 36 characters (32 digits and 4 hyphens).
3. The "type" - an array of credential types, where each entry in this array should be associated with an entry in the "@context" field, in the same order.

```
{
   “@context”: [                                                                    1
      “https://www.w3.org/2018/credentials/v1”,
      “https://www.lacchain.net/credentials/library/
      education/4e6c312cd8e6b18116fe3fd2e9b6e5df810afe0a716c1c511ef6c19cb8554578/v1”
   ],
   “id”: “28dffe06-097a-46a2-bd76-396c233c602a”,                                    2
   “type”: [                                                                        3
      “VerifiableCredential”,
      “Certificate”
   ],
   “issuer”: “did:ethr:lacchain:0xadf1702b76419f428014d1386af487b2d8145f83”,        4
   “issuanceDate”: “2020-10-10T20:06:00.033Z”,                                      5
   “expirationDate”: “2025-10-10T20:06:00.033Z”,                                    6
   “credentialSubject”: {                                                           7
      “id”: “did:ethr:lacchain:0x48007072061dc756e5a2ecf15cf2c2bcc091de52”,
      “givenName”: “Juan”,
      “familyName”: “Perez”,
      “email”: “juan.perez@gmail.com”
   },
   “evidence”: true,                                                                8
   “credentialStatus”: {
      “id”: “0x4185Dab0662ccDa3D3F35779578a4242bb89Db37”,
      “type”: “SmartContract”
   },
   “proof”: [
      {                                                                             9
         “id”: “did:ethr:lacchain:0xadf1702b76419f428014d1386af487b2d8145f83”,
         “type”: “EcdsaSecp256k1Signature2019”,
         “proofPurpose”: “assertionMethod”,
         “verificationMethod”: “0x7a746D34754C14EB3eb1F214BD0EA23a1A18Be7A”,
         “proofValue”: “0x867147db1b65930f20e1f8bcef67f4869426081522b3cd5c351da8ca
         8b8b82d42f2a63c6c19e28ce668a1f85b51522761a71f28dda5f70c5510f841518f2836e1c”
      }
   ]
}
```

*Figure 3. Example of a VC compliant with the LACChain ID Framework.*

4. The "issuer" - the issuer identifier, which in this case, is their DID.
5. The "issuanceDate" - the VC creation timestamp in ISO string format (YYYY-MM-DDTHH:mm:ssZ).
6. The "expirationDate" – the VC expiration timestamp in ISO string format (YYYY-MM-DDTHH:mm:ssZ).
7. The "credentialSubject" - a field that contains all claims of the credential. The format of this field is not pre-defined; it can be any simple or composite object with all fields included, and its structure must be defined in the schema specified in the "@context" field and the credential "type".
8. The "credentialStatus" - points to a smart contract that plays the role of a certificate revocation list (CRL). This field has two parameters:

    - "id": the URI of the resource where the status of the credential can be consulted, which, in this case, is the smart contract address.
    - "type": the type of resource where the status of the credential can be consulted, which, in this case, is a Smart Contract

    The LACChain ID Framework encourages the expiration conditions of the VC to be validated on-chain using a smart contract that performs a function similar to a CRL, where the unique VC identifier is stored together with the original expiration date. In this way, the revocation status can be changed by calling on the smart contract indicated in the "id" field of "credentialStatus".
9. The "proof" - an array of cryptographic signatures associated with an issuer or a signer that guarantees the credential's integrity. Each item in this section has the following fields:

    - "id": a unique proof identifier, which, in this case, is the issuer DID
    - "type": the digital signature algorithm
    - "proofPurpose": the purpose of the signature. In the case of Figure 2, is to verify the integrity of the credential content. The possible values are: *assertionMethod* (to verify the content integrity) and *authentication* (to verify the issuer identity)
    - "verificationMethod": the Verification Method used to validate the signature. In the case of Figure 2, is the #vm-1 which corresponds to the first VM DID fragment
    - "domain": the endpoint used to verify the signature. In the case of Figure 2, is the smart contract address where the issuer registered the VC hash and signature
    - "proofValue": the digital signature in hexadecimal format

In order to keep VC verification fully on-chain and compliant with EIP 1812, the issuer's hash claims VC signature can be stored in a smart contract using the *secp256k1* algorithm.

The signature that is sent to the smart contract is displayed in the "proofValue" field and the actual address of the contract is specified in the "domain" field of this section. Anyone can query the smart contract to verify the VC (see Section 2.5).[3]

```
{
   "@context": [{
      "@version": 1.1
   },"https://www.w3.org/ns/odrl.jsonld", {
      "la": "https://lacchain.net/credentials/library/education",
      "schema": "http://schema.org/",
      "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
      "givenName": "la:givenName",
      "familyName": "la:familyName",
      "email": "la:email",
      "holds": "la:holds",
      "category": "la:category",
      "industry": "la:industry",
      "skillset": "la:skillset",
      "course": "la:course",
      "description": "la:description",
      "duration": "la: duration",
      "modality": "la:modality",
      "location": "la: location"
}] "url": "la:url" }
```

*Figure 4. Example of an extension of a VC "credentialSubject" developed by LACChain for diplomas issued by LACChain Academy.*

3 LACChain has defined a collection of schemas in JSONLD format to extend the W3C VC 1.0 standard to add claims to the credential in the field "credentialSubject". LACChain maintains a public library of these extensions (see Section 2.2).

## 2.2. Credential Registries and Classification

Only off-chain registries SHALL be used as credential registries because they are the only scalable approach for transactions and storage, as well as the only option that meets data protection requirements. The blockchain is called upon when the VCs are issued in order to register the cryptographic proofs and are queried when the VCs are presented to others to verify those proofs. In most blockchain networks, these queries do not generate transactions. Therefore, they do not leave any tracks nor consume any blockchain resource. In the SSI model, he issuer and the subject are the entities that should have copies of the credentials. Issuers typically integrate their existing databases and issuing systems with the blockchain, while subjects are encouraged to use reliable, secure, portable, and user-friendly digital wallets (see Section 3).

The LACChain Alliance has developed and maintains a public library of VCs that aims to incorporate VCs used in real use cases across Latin America and the Caribbean in areas such as education, health, energy, public administration services, and land registry, among others. This library is in the domain https://credentials-library.lacchain.net and the VCs are also stored in the LACChain Github[9] and the LACChain IPFS nodes (see the LACChain Framework for Permissioned Public Blockchain Networks[10]).

## 2.3. Presentations

In this context, the exchange of credentials can be understood as a presentation. The W3C introduces the concept of verifiable presentations in the VC specification. As stated in the W3C specification, "a verifiable presentation expresses data from one or more VCs and is packaged in such a way that the authorship of the data is verifiable. If VCs are presented directly, they become verifiable presentations"[11]. Verifiable presentations SHALL contain the following elements:

- URI to uniquely identify the presentation
- URI to uniquely identify the type of object
- One or more verifiable credentials or claims

- URI to identify the entity generating the presentation (e.g., a DID)
- Cryptographic proof of the subject (e.g., a digital signature)

As the verifiable presentations contain one or several VCs, they SHALL also comply with the requirements presented in Section 2.1. One of the main challenges when dealing with different electronic services that issue, manage, present, and verify verifiable credentials is determining how to exchange or present these services in a consensual way. Unfortunately, there is not yet a single standard that dictates how to do so. Some DID methods specify mechanisms and protocols for the exchange of credentials, which is not ideal because it limits interoperability with other DID methods. DID method specifications SHOULD remain at the DID layer and protocols for the exchange of credentials SHOULD be established at the credential level.

The LACChain Alliance has developed the LACChain Mailbox, a solution that allows for various electronic services that follow different DID methods to be able to exchange VCs. The LACChain Mailbox is a secure and private system for exchanging messages, VCs, and verifiable presentations. It is a controlled by a centralized orchestration entity that allows identified clients using DIDs to send and receive messages that are stored and encrypted in a secure database. Different algorithms are recognized for authentication, including post-quantum cryptography. This is an open-source tool available for identity services on top of the LACChain Networks.

## 2.4. Revocation

Clear revocation rules SHALL be defined for each credential so that it is clear who and under which conditions someone can modify the status. Some examples of these rules are:

- Status is automatically set as active when the credential is issued by the issuer
- Issuers can change the status to "revoked" when the subject ceases meeting the claims attested in the credential
- Issuers can change the status to "suspended" when the subject reports that the credential, authenticators, or associated proofs have been compromised

- Issuers can change the status to "revoked" when the user reports that they do not want to use the credential anymore
- Subject or holder[4] can change the status to "suspended" when the credential or their keys have been compromised
- Subject or holder can change the status to "revoked" when they no longer want to use the credential

The status of the credential is automatically changed to "revoked" after the expiration date where the revocation rules are defined as follows:

- issuer_all: the issuer can revoke the credential at any time
- subject_before_expires: the subject can request to revoke the credential before it expires

```
"credentialStatus": {
   "id": "0x032423480238498203948028342 83",
   "type": "SmartContract",
   "revocationRulesList": ["issuer_all", "subject_before_expires"]
}
```

*Figure 5. Example of the definition of the revocation rules in a VC or verifiable presentation.*

This form of revocation uses the hash of "credentialSubject" as the VS identifier in the Revocation List, following the EIP 712[12]. Issuers SHALL be able to change revocation status by invocating the smart contract revocation function, as long as the credential is within a validity period.

Based on the EIP 1812[13], which also relies on EIP 712 for on-chain data signing, it is possible to maintain an on-chain revocation list of VCs. Both the issuer and the subject CAN be allowed to revoke the VC unilaterally by defining the revocation rules in the smart contract or by simply changing the status of the credential by calling the "revokeCredential" method directly.

It is also possible to define a set of rules that specify under which conditions a revocation can be executed before the VC expires inside the VC.

4 In the terminology of the W3C standard, holders are authorized users in possession of VCs not being necessarily the subjects of the VCs.

## 2.5. Verification Process

The process for the verification of digital credentials is not standardized and is generally not rigorous enough. The LACChain ID Framework comes with the LACChain Verification Process presented in this section and allows any verifier entity to accomplish diligent electronic verifications of digital credentials that holders present to the verifiers. The LACChain ID Verification Process is as follows:

### 2.5.1. Verification of the Validity of the Credential

The verifier SHALL verify that the structure, format, and context are correct. All of this information is contained within the credential and can be automatically verified by a verification service. Standardization of structure, format, and context will enable worldwide VC recognition, including digital passports, diplomas, and property titles.

### 2.5.2. Verification of the Status of the Credential

The verifier SHALL verify that the VC has an active status. As mentioned before, we encourage the use of smart contracts to play the role of CRLs maintained by the credential issuers. In this case, VCs contain a field that indicates the address of the smart contract where the identifier of the VC is associated with a dynamic status which can be changed by the issuer between "active", "suspended", and "revoked". Step 3 of the Verification Process SHALL only be considered successful when the status is active.

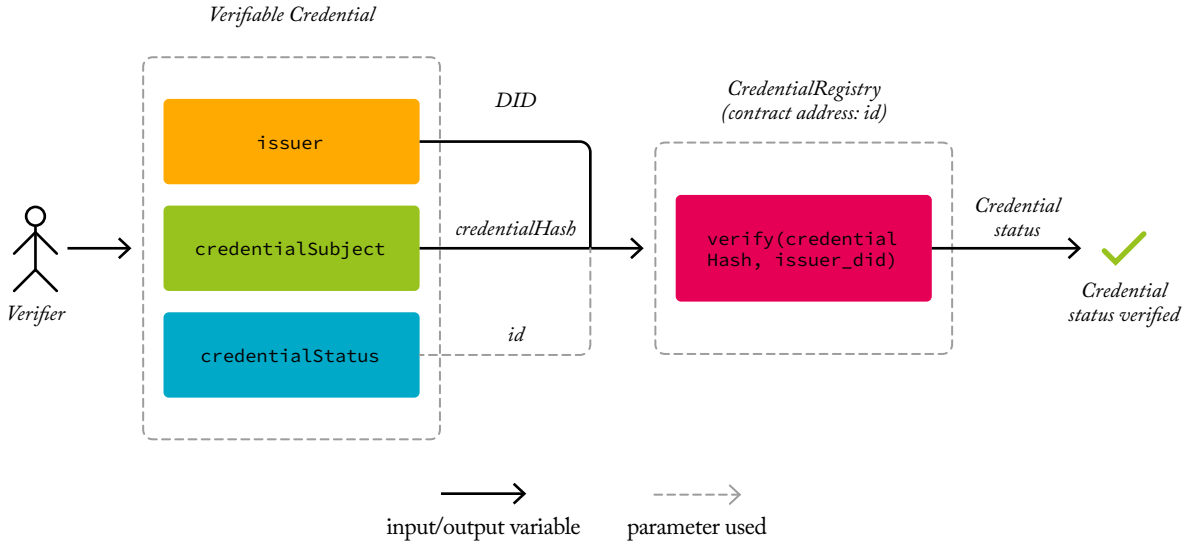


*Figure 6. On-chain verification of the status of using smart contracts.*

Following the EIP 1812, the verification of VC status can be based on a smart contract, which contains and allows for status modification based on simple rules (see Section 2.4). This is illustrated in Figure 6. The diagram shows the "verify" smart contract method, which address is indicated in the "id" field of the "credentialStatus" of the VC. The "credentialHash" of the "credentialSubject" and the "issuer" DID of the VC are passed as parameters.

### 2.5.3. Verification of the Issuer

When required, the verifier SHALL verify that the issuer signed the VC, and that the issuer is trusted or recognized. In order for the verifier to trust the issuer, the verifier might need to know the issuer's real identity. In general, the credential presented by the presenter only contains the DID and digital signature of the issuer, but not additional information about the issuer's identity. Therefore, the DID and digital signature of the issuer, even if valid, might be unknown or untrusted by the verifier. If this is the case, in order to verify the issuer's real identity, the verifier might require knowing that the issuer's DID has been certified by an entity (a certificate authority (CA) or trusted issuer) that is trusted by the verifier.

In order to allow this to be verified, the presenter could present not only their VC, but also a chain of VCs that contains verifiable claims about the root of trusted issuers that go from this issuer to a trusted issuer or CA, as an analogy of what happens with X.509 certificates today (where browsers verify these roots of trust). However, these chains of VCs can be optimized because CAs and trusted issuers MAY maintain on-chain public key directories and trusted lists so there is no need for the presenter to present a chain of VCs in order for the verifier to verify the identity of the issuer (see Section 4.2.1 and 4.2.2). A public key directory contains information that associates the issuer's DID with information about the issuer's real identity.

Figure 6 illustrates the verification of the issuer, which consists of two important steps:

1. Verification of the issuer's digital signature: requires verification of the issuer's cryptographic proof of the VC. In this case, the smart contract which address is indicated in the "id" field of the "proof" section is called, together with the hash of the "credentialSubject" signed by the "issuer" DID.
2. Verification that the issuer is authorized for the issuance: requires verification that the entity that issued and signed the VC is trusted by the verifier. To achieve this objective, a trust framework might be necessary to define under which identifiers (e.g., public keys or DIDs) a trusted authority is authorizing an entity as an issuer (e.g., a government authorizing a medical center to issue vaccination credentials). There are different ways of establishing roots of trust discussed in Section 4.2.2.

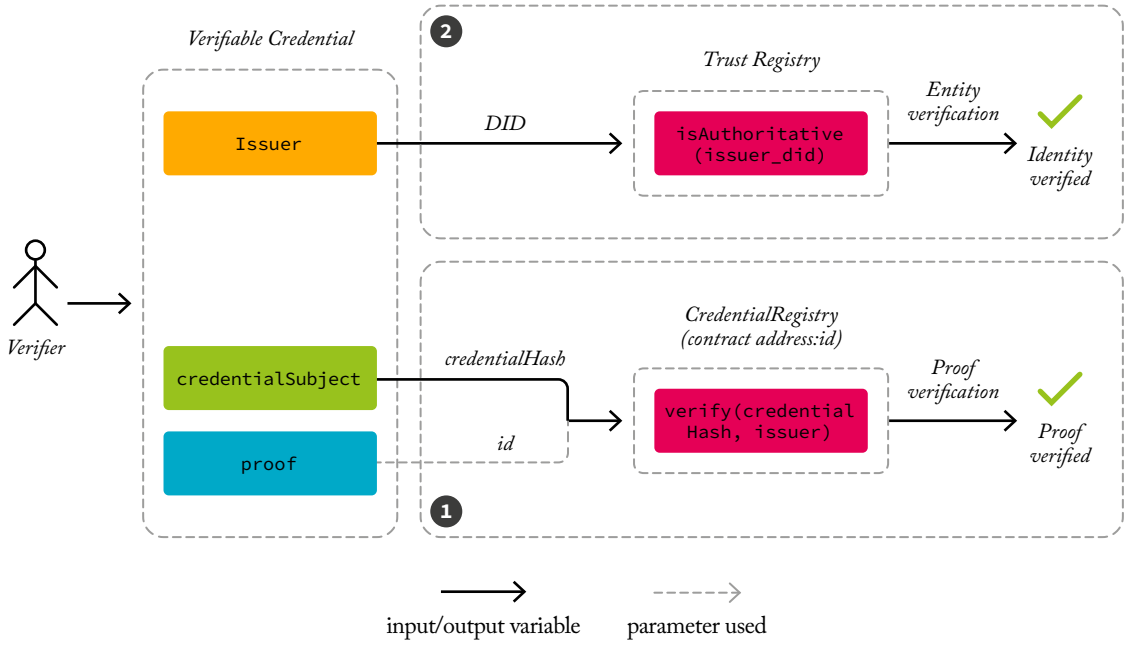


*Figure 7. On-chain verification of the Issuer of a VC compliant with the LACChain ID Framework.*

### 2.5.4. Verification of the Presenter

The verifier SHALL verify that the presenter is authorized to present the VC, either because the presenter is the subject or because the presenter has been authorized by the subject to present the VC. The verifier indicates the level of assurance they require for the verification of the presenter. When the presentation is electronic (i.e., there is not an in-person presentation that allows for in-person visual verification of biometric characteristics from a photo ID), the verifier CAN require different types of verification, including the following two:

- Control of the private keys: The presenter will have to sign the presentation with a key associated with the subject's DID that appears in the VC.
- Control of the private keys + persistent identity proofing over the VC lifetime: The presenter will have to sign the presentation with a key associated with the subject's DID that appears in the VC and will also have to prove somehow that the person who was in control of the DID at the time of the issuance of the VC by the issuer is the same person who is in control of the keys at the moment of presentation of the VC to the issuer. This way, the verifier relies on the identity proofing to the subject carried out by the issuer.

### Control of the Private Keys

The presenter is required to prove that they are in control of the DID by solving a challenge associated with one of the authentication methods. If the presenter is not the subject but is someone authorized by the subject, there are at least two options. Option one is that the presenter is given one of the private keys required for the authentication of that DID. Option two is that the VC specifies that a DID different from the subject can present the credential, which would be the presenter DID.

In any case, the presenter needs to sign the VC to guarantee its integrity, by following the next process:

1. The presenter generates the of the hash of the "credentialSubject" field in the VC using the SHA-256 algorithm
2. The presenter selects a verification method from their DID, specifically from the "assertionMethod" section, and signs the hash of the "credentialSubject" with the private key associated with the selected Verification Method public key.
3. The presenter includes the signature in the VC's "proof" section

### Persistent Identity Proofing (Control of the Private Keys + Proof of Identity over the VC Lifetime)

In any electronic interaction, there may be needed two types of verifications: 1) verification of the electronic information exchanged between parties or presented from one party to the other and 2) verification of the physical entities behind the digital personas involved in the digital interaction. In the SSI model, individuals store, manage, and present credentials using digital wallets.

In general, when a holder electronically presents a credential to a verifier, the holder first needs to establish a communication channel between the holder's digital wallet and the verifier's digital service (e.g., https). When the verifier receives the credential, they are capable of verifying all the electronic information (i.e., validity of the credential, status, issuer, presenter, and claims) against different trust registries. However, there is not an easy way for the verifier to verify that the person in control of the device that is presenting the digital credential (the presenter) is truly the digital subject or someone authorized by the digital subject. This issue is known as the identity binding problem.

Some discussions around the topic of the identity binding problem assume that that if the issuer of a specific credential accomplishes identity proofing (including verifying

that the subject is in control of a specific wallet) it is sufficient enough for any verifier's trust. However, even if the verifier can trust that the claimed subject was in control of the wallet at the time of issuing the credential, it does not prove that the subject is still in control of the wallet at the time of presentation of the credential to the verifier. What if someone stole the wallet and is impersonating the subject? What if the subject lent the wallet to someone else?

As we discussed previously, the verifier will define the level of assurance required for the presenter's verification, which in some cases will require the presenter to prove that they are the same person who was in control of the DID when the VC was issued to that DID. For electronic presentations, one alternative to accomplishing identity proofing is requiring a third-party verification. Another alternative consists of trusting the wallet to certify that the user accessing the wallet at the time of presenting the VC is the same user that was using it at the time of receiving the VC (e.g., by doing a local biometric timestamp). In the case of physical presentations, it is always possible to show both the digital presentation and a photo-ID.

We believe that it is essential for digital wallets to become trust services according to regulatory frameworks in order be trusted for user authentication by verifiers at the time of use for digital interactions, such as presenting VCs and verifiable presentations. The recognition and certification of digital wallets as trust services is essential for the scalability of SSI solutions.

In conclusion, when the technology and the regulatory frameworks become advanced enough, digital wallet providers SHOULD be able to guarantee that a credential presenter is authenticated to the wallet with a certain level of assurance at the time of presentation. This suggests the digital wallet is able to guarantee that the holder presenting a credential is the same person that was in control of the wallet when the credential was issued to it with the level of assurance required by a verifier. The digital wallet provider SHOULD assume liability for this assurance.

### 2.5.5. Verification of the Claims

If all the previous steps are successful, the verifier SHALL verify the information attested in the VCs or presentation, according to the verification mechanism indicated in the VC (see Section 2.1). Figure 7 illustrates how to verify the claims of a VC using a smart contract. The verifier takes the "proofValue" field from the "proof" section of the VC to check the signature of the claims by the "issuer"'s DID. The address of the contract is taken from the "id" field of the same "proof" section.

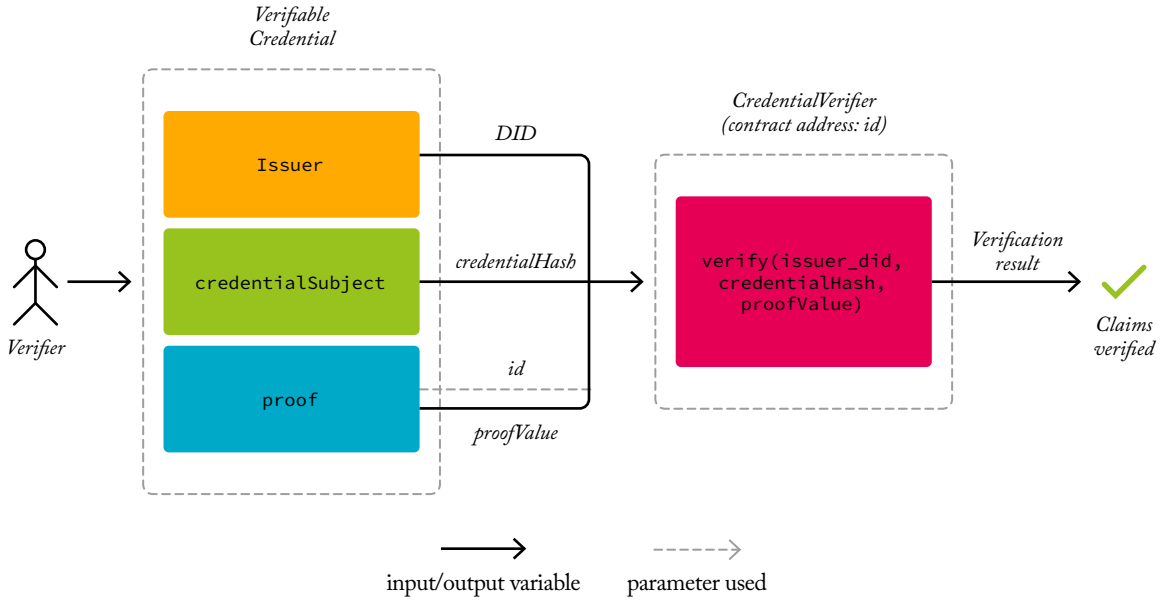


*Figure 8. Verification of the claims of a VC.*

## 2.6. Selective Disclosure Mechanisms and ZKP

When working with elements such as blockchain registries and digital signatures, there is a more sophisticated way of proving that something is true without the need to reveal any information about the fact itself. These claims are known as zero-knowledge proofs (ZKP). According to NIST, the main types of ZKPs are the following[14].

- Equality: the value of a magnitude is equal or non-equal to a given value.
- Inequality: the value of a magnitude is higher or lower than a given value.
- Membership: a subject is in a list.
- Range: the value of a magnitude is within a given interval [a, b] or not.

In the context of a prover (the entity trying to prove the veracity of a claim) and a verifier (the entity trying to verify whether a claim is true), ZKPs must satisfy the following three properties:

- Completeness: if the claim is true, an honest verifier will be convinced.
- Soundness: if the claim is false, a cheating prover will not be able to convince the verifier.
- Zero-knowledge: if the claim is true, the verifier learns nothing other than the fact that it is true.

ZKPs in the context of VCs allows for sharing of information in a verifiable presentation format without disclosing other information or data that the VC may contain. By selecting only some VC claims issued by a trusted issuer, it is possible with digital signatures to verify those claims and their issuance by the trusted issuer without the need to present the full VC issued by the trusted issuer. To achieve the proposed objective, it is necessary to use digital signature algorithms that allow the generation of dynamic signatures from a previously issued VC, but without the issuer itself having to re-certify each of the issued claims. For an example of an implementation of a ZKP algorithm using VCs and verifiable presentations, see Appendix I.

## 2.7. Traceability and Monitoring

VCs and verifiable presentations SHALL be off-chain files. The only information about the VCs and verifiable presentations that CAN be registered in the blockchain is the registration, modification, and revocation of cryptographic information associated with it. Presentation and verification SHOULD not leave any track in the blockchain. In order to track the registration, modification, and revocation of VCs, the following events CAN be generated by the smart contracts used to register the hash and status of the VCs:

- event CredentialRegistered (bytes32 indexed credentialHash, address by, address id, uint iat). This event is generated when a VC is issued. It contains the VC hash, addresses associated with the sender and recipient's DIDs, and the VC expiration date.
- event CredentialRevoked (bytes32 indexed credentialHash, address by, uint256 date). This event is generated when a VC is revoked. It contains the VC hash, issuer DID, and block timestamp.
- event CredentialStatusChange (bytes32 indexed credentialHash, address by, uint status, uint256 date). This event is generated when a VC changes its status. It contains the VC hash, issuer DID, VC status, and block timestamp.

LACCHAIN ID FRAMEWORK

# 3 DIGITAL WALLETS

IDB IDB | LAB LACCHAIN

# 3.1. Definition

In the context of self-sovereign identity, a digital wallet is a private repository that allows its owner to store, manage, and present both keys and identity credentials. Digital wallets SHALL comply with the following requirements:

- Provide secure access to the holder, by guaranteeing that only authorized entities can access it.
- Ensure security and strong data encryption.
- Provide recovery of keys and credentials.
- Do not have access to subject's private keys.
- Be connected to the ledgers where the DID registries, trusted lists, and cryptographic proofs of the DID documents and credentials are stored.
- Provide mechanisms for subjects and issuers to change the status of their credentials.
- Provide mechanisms for the owner to erase all the data associated with them (including revoking DIDs and VCs).

Digital wallets SHOULD comply with the following requirements.

- Provide dashboards of activity.
- Provide mechanisms for reducing PII of the entities' activities by combining the use of different DIDs for different interactions.
- Become certified and/or audited to be acknowledged as trusted services.

Also, digital wallets CAN:

- Keep transactional information about the subjects, if authorized to it.

There are different types of digital wallets, including the following.

- Desktop wallets (installed onto a particular computer)
- Browser wallets (browser extensions installed in a particular computer)
- Hardware wallets (physical devices such as a hard drive or USB)
- Cloud wallets (based in cloud-storage)
- Mobile wallets (mobile applications)

## 3.2. Key Recovery

Private keys allow individuals to prove ownership of their identifiers and credentials. Therefore, it is essential that digital wallets guarantee key recovery mechanisms in case of loss or theft of the keys themselves or the digital wallets as key management systems. There are different ways of enabling key recovery, among which we encourage four: local back-ups, cloud back-ups, deterministic key generation, and decentralized key storage. Digital wallets SHOULD allow both local back-ups and cloud back-ups complementarily, as well as other recovery mechanisms.

### 3.2.1. Local Back-Ups of Primary Keys

Local back-ups consist of exporting copies of the keys or seeds of the keys to a local storage, such as a hard drive. This way, they can be used to retrieve the identity using a new digital wallet when the old one is lost or compromised. Digital wallets SHALL enable the back-up of the keys and seeds and the reconstruction of an existing identity by importing those keys.

### 3.2.2. Cloud Back-Ups of Primary Keys

Similar to the local back-ups, digital wallets SHALL enable the use of private cloud instances to store private keys and seeds. In these cases, the digital wallets need to enable secure access to these back-ups with specific authentication mechanisms.

### 3.2.3. Identity Recovery Using Multiple Keys

It is also possible to establish a hierarchy of keys than can be used to recover the identity when the primary keys are compromised. For this purpose, there are two strategies worth mentioning: deterministic generation and social recovery based on interpolation algorithms.

### 3.2.4. Recovery of Credentials

Similar to key storage and management, credentials can be backed-up locally, in the cloud, or in decentralized registries, such us the LACChain IPFS nodes.[5] Digital wallets SHALL enable different back-up mechanism according to the privacy conditions required by individuals.

---

5 For more information read the *LACChain Framework for Permissioned Public Blockchain Networks*[10].

# 3.3. Identity Proofing, Authentication, and Authorization

Identity proofing, authentication, and authorization are present in every provision of an electronic service by a service provider to a requester. Identity proofing consists of verifying that the requester is who they claim to be. Authentication consists in making sure that the electronic service is provided and consumed safely. Authorization consists of verifying that the requester is authorized to consume the service and allowing them to do so. Implementations SHALL be mapped to the ISO/IEC 29115 standard to ensure interoperability.

## 3.3.1. Identity Proofing

The identity proofing flow in SSI can be described as follows.

1. The requester entity applies for identity VCs.
2. The issuer verifies the real identity of the subject and/or its DIDs.
3. The issuer issues the identity VCs and sends them to the subject/holder.
4. The subject/holder stores the identity VCs in a digital wallet.

Any entity can become an issuer of identity VCs in the SSI scheme. However, that does not mean that any third party will recognize those identity credentials when presented to them. When a subject presents an identity VC to a third party that plays the role of a verifier (see Section 2), it is up to this third party to accept or deny the validity of that VC. The roots of trust behind the certification of each identity issuer and the level of assurance required for the identity credentials (see Section 4) are established by trust frameworks (see Section 6).

## 3.3.2. Authentication

Authentication is always based on three types of factors:

- Something you know (i.e., a password).
- Something you have (i.e., a mobile phone, ID credential received after accomplishing identity proofing, or a cryptographic key).
- Something you are (i.e., a fingerprint or other biometric data).

## Authentication to Digital Wallets

In the SSI model, individuals store their keys and credentials in personal and private repositories that are self-managed by them with no need of any third party. As discussed repeatedly in this document, these repositories are known as digital wallets. Preventing non-authorized entities from accessing someone else's digital wallets is critical. If a non-authorized entity has access to another person's digital wallet, they could control that person's identity credentials, digital money, cryptocurrencies, digital diplomas, digital property titles, and any other data stored there, allowing the entity to impersonate that person.

In order to have reliable and secure SSI solutions, the security of authentication to digital wallets must be guaranteed. Wallet providers must develop solutions that do not require any level of technical or technological skills for the user to ensure their protection.

Wallet providers SHALL develop different mechanisms to authenticate individuals:

- At the time of signing/setting up
- At the time of signing in

Wallet providers SHOULD develop different mechanisms to authenticate individuals:

- At the time of accessing particularly sensitive assets
- At the time of exchanging assets or presenting digital credentials to others

These stages can be broken down as follows.

1. **Setting up the digital wallet:** Digital wallets SHALL require a minimum set of authenticator factors to the user. Once the user accomplishes the sign-up process, they can start creating DIDs, generating and receiving VCs, and presenting information to others (e.g., in order to access digital or physical services).
2. **Logging in to the digital wallet:** Digital wallets SHALL ensure that authentication factors allow verification of the user's identity with a high level of assurance, combining factors that the user knows, has, and is.
3. **Accessing assets and digital credentials:** Once the user is logged in, digital wallets SHALL restrict access to particularly sensitive information and request additional real-time verifications, such as biometrics or security questions, when intending to access this information.
4. **Present assets and digital credentials from the digital wallet to others:** Digital wallets SHALL also require users to accomplish additional verifications before sharing

sensitive information to others or when using digital wallets to access electronic services. In some cases, the third party requesting a credential could indicate to the wallet that it requires a specific assurance in the subject verification (see Section 2.5.4). An airline requiring authentication to book a flight is an example of this scenario.

### Authentication to Services

In order to access a digital service, the holder presents a credential from their digital wallet to the service provider. When receiving the VC, the service provider has to be able to accomplish the steps in the Verification Process described in Section 2.5. This includes the verification of the 1) structure, format, and context of the credential, 2) status of the credential, 3) issuer, 4) presenter, and 5) claims.

If the service provider cannot verify all of the previous items, the authentication process will fail and the individual will not be authorized to access the service. The service provider will not be able to accept a credential if they do not trust the digital wallet from which the credential is presented. The service provider will also not be able to accept a credential if they do not recognize the syntax of the credential or if they do not trust the issuer. In a self-sovereign ecosystem completely aligned with regulatory policies, non-discrimination of certain authentications coming from qualified digital wallets and standardized credentials issued by qualified and trusted issuers (e.g., a digital passport issued by a government) can be enforced.

This framework encourages the use of authentication methods based on the use of a DID as an identity verification mechanism, using the public keys of a DID document to prove the identity of the subject in control of that DID.

DID Connect (originally proposed by KayTrust DID Connect)[15] is an extension of OIDC for use of DIDs to achieve electronic authentication. DID Connect introduces the use of DIDs and VCs for a decentralized mechanism that allows a client to verify a user's identity. DID Connect makes authentication mechanisms and keys associated with the DID possible and also allows VCs to include information about the user in the sign-up process.[16]

### 3.3.3. Authorization

As introduced in the LACChain ID Verification Process presented in Section 2.5, when a digital credential is presented with the purpose of being granted access to a service (digital or physical), the service provider verifies that the digital wallet presenting the credential (if applicable) can be trusted, that the credential is valid, the issuer is known, and the presenter is authorized to be presenting that credential. Digital wallets SHOULD enable mechanisms for subjects to authorize the use of some of their VCs and verifiable presentations in a responsible way, aligned with rules established by the issuers. There are at least two types of authorizations that can be checked when a VC is presented.

**Authorization of the Presenter**

When a credential is presented to a verifier, in step 4 of the LACChain ID Verification Process (see Section 2.5.4) the verifier verifies that the presenter is authorized to present that credential, either because they are a legitimate subject or because they have been authorized to present that credential, which implies a delegation.

**Authorization of the Subject**

If the presenter is the subject and there is no delegation, the authentication proceeds as usual.

- The presenter must be able to prove that they are in control of the subject's DID by solving a challenge to one of the verification mechanisms listed the DID Document, which requires being in control of the private keys (see Section 3.3.2.2)

**Authorization of Delegated Holders/Presenters**

There are at least two ways of delegating the right to present a credential:

- In the DID document, by indicating different types of verification mechanisms (e.g., authentication methods) and indicating some mechanisms that are in the delegate´s control
- In the VC or the verifiable presentation, by indicating authorized presenters that are different from the subject.

In Figure 9, we illustrate how to delegate control of the DID by changing the owner. As noted, the "id" of the DID Document is not the same as that of the "controller". That is because the control of the DID is delegated to another DID in the "controller" field, which grants the ability to make modifications to the DID.

In Figure 10, we illustrate how to delegate control of the DID by adding a verification method in the "capabilityDelegation" section. In Figure 11 we illustrate how to authorize a holder to present a VC or verifiable presentation, indicating the "holder" in the VC or verifiable presentation itself.

```
{
   "@context": "https://w3id.org/did/v1",
   "id": "did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83",
   "controller": "did:lac:main:0x094abde76419f428014d1386ae3e6184d423324",
   "verificationMethod": [
      ...
   ]
}
```

*Figure 9. Delegation in the DID document by changing the controller of the DID.*

```
{
   "@context": "https://w3id.org/did/v1",
   "id": "did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83",
   "verificationMethod": [
      {
         "id": "did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1",
         "type": "Secp256k1SignatureAuthentication2018",
         "controller": "did:lac:main:0x094abde76419f428014d1386ae3e6184d423324",
         "publicKeyHex": "0417372efd731a942f769712d13573f...9d7e47cb5de7b04533
         41dc18472f875fa999a7e93"
      }
   ],
   "capabilityDelegation": [
      "did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83#vm-1"
   ]
}
```

*Figure 10. Delegation in the DID document by adding a delegate capability.*

```
{
   “@context”:[
      “https://www.w3.org/2018/credentials/v1”,
      “https://www.lacchain.net/credentials/library/
      education/4e6c312cd8e611ef6c19cb8554578/v1”
   ],
   “type”:[
      “VerifiableCredential”,
      “Certificate”
   ],
   “issuer”: “did:lac:main:0x094e5df810afe0a716c1c511ef6c19cb8554578”,
   “holder”: “did:lac:main:0x094abde76419f428014d1386ae3e6184d423324”,
   “credentialSubject”:{
      “id”: “did:lac:main:0x094e5df810afe0a716c1c511ef6c19cb8554578”,
      “degreeType”:”BachelorDegree”,
      “degreeSchool”:”College of Engineering”
   }
}
```

*Figure 11. Delegation in the VC or verifiable presentation by introducing an authorized “holder”.*

### Authorization of the Purpose

Based on the W3C VC standard, it is possible to specify the purpose of a VC in the field “proofPurpose” inside the “proof” section. In Figure 13 the “proofPurpose” is assertionMethod.

```
{
   “@context”: [
      “https://www.w3.org/2018/credentials/v1”,
      “https://www.lacchain.net/credentials/library/
      education/4e6c312cd8e611ef6c19cb8554578/v1”
   ],
   “type”: [“VerifiableCredential”, “Certificate”],
   “issuer”: “did:lac:main:0x094e5df810afe0a716c1c511ef6c19cb8554578”,
   “credentialSubject”: {
      “id”: “did:lac:main:0x094abde76419f428014d1386ae3e6184d423324”
   },
   “proof”: {
      “type”: “RsaSignature2018”,
      “created”: “2017-06-18T21:19:10Z”,
      “proofPurpose”: “assertionMethod”,
      “verificationMethod”: “SmartContract”,
      “jws”: “eyJhbGciOiJSUzI1hbGciOihbG...NiIsImI2NCI6ZmFsc2U2NCI6ImI”
   }
}
```

*Figure 12. Authorization of the purpose in a VC.*

LACCHAIN ID FRAMEWORK

# 4

# PKDS, DNSS, CRLS, OSCPS, TLS, AND ROOTS OF TRUST, AND TRUST FRAMEWORKS

IDB IDB | LAB LACCHAIN

# 4.1. Traditional PKI

It is essential to understand that an SSI scheme based on DIDs, VCs, digital wallets, and decentralized registries is not an alternative to a PKI infrastructure. Rather, it is PKI infrastructure based on a new generation of protocols, standards, and technologies. In order to understand what the scenarios in which an SSI-based PKI scheme is beneficial, it is important to understand how traditional PKI systems work today.

In a digital identity scheme based on PKI, identity credentials -particular types of electronic certificates- are issued by certificate authorities (CAs). These identity credentials allow individuals and entities to electronically identify and authenticate to others. In government-based solutions, the government designates the root CAs. In non-government-based solutions, different types of entities (e.g., banks, universities, or NGOs) can become trusted CAs in certain contexts.

CAs maintain lists of certificates that have been revoked, called certificate revocation lists (CRL). When a certificate is presented to a verifier by the subject, the verifier can verify its status against the CRL, which is usually a URL indicated in the certificate. Another method used to check revoked certificates is the online certificate status protocol (OCSP), in which the browser requests the status of a particular certificate from the issuing CA's revocation server instead of dealing with a full CRL. When passing the certificate's serial number to the CA's, it replies with a digitally signed response containing the certificate status, which can be "good", "revoked", or "unknown".

Not all electronic certificates issued directly by CAs are recognized by all verifiers or browser agents. Sometimes a root CA generates a first certificate, and then a chain of linked certificates is built from that first certificate. This scenario happens often within corporations or for subdomains. In order to verify the final certificates of the chain, the verifier -generally a browser agent- resolves the entire root of trust and verifies it up to the root CA.

The way in which domain names are remembered[6] is another interesting component of the way the Internet works with regard to traditional PKIs. Every device connected to a computer network that uses the Internet for communication has an internet protocol

6 An example of an IP address is using the Internet Protocol version 4 (IPv4) is 35.245.15.167. The domain system for this address is explorer.lacchain.net. The IPv4 is being replaced by the IPv6, which IP addresses are way harder to remember, as they can be sequences of 30 digits (letters and numbers).

(IP) address -assigned by the network administrator- which serves as a network interface identification and is used to address the device's location. The Internet Assigned Numbers Authority (IANA) and five regional Internet registries (RIRs) globally manage the IP address space. However, IP addresses are not easy to remember. This is why there is a domain name system (DNS), which is essentially a list that keeps track of the domain name of each IP (if it has been registered). The DNS is maintained and can be resolved against root servers operated by 12 trusted entities[7].

Similar to how DNS allows IPs to associate with known domain names, public key directories (PKDs) can associate public keys with entities. For example, the European Commission maintains a PKD with public keys associated with all the Country Members, and the World Health Organization is developing a global PKD to enable the verification of digital vaccination certificates issued by different countries. The infrastructure and the entity responsible for these PKDs are usually centralized, and when there is not a trusted central entity or infrastructure, it is not easy to create these PKDs. This happens in Latin America and the Caribbean, for example, because, unlike Europe, there are not regional entities representing common interests and maintaining common infrastructures.

In essence, the Internet works fine today. It serves the purpose of enabling companies and servers to authenticate to one another and establish trusted connections. However, it is not suitable for the generation of digital certificates for people, which have become extremely demanded in light of the COVID-19 pandemic. The verification of current digital certificates rely on centralized PKDs, CRLs, and OSCPs and do not allow universal verification (i.e., access is generally restricted to authenticated and authorized entities). Additionally, there are not secure and portable personal repositories to manage our credentials, and issuance and verification processes are generally not compatible nor interoperable between different entities or coutries. The new generation of VCs, DIDs, digital wallets, and decentralized ledgers can help with these issues. Of course, it would not make sense to redo the entire PKI infrastructure. The use of these emerging tools must take place in a progressive manner and under proper frameworks -such as this paper- that allow compatibility with legacy systems.

---

7 The 12 trusted entities maintaining DNS root servers are VeriSign (two of them), the University of Southern California, Cogent Communications, the University of Maryland, NASA Ames Research Center, the Internet Systems Consortium, the Defense Information Systems Agency, Netnod, RIPE NCC, ICANN, the WIDE Project, and the US Army Research Lab.

## 4.2. SSI-Based PKI

In the SSI scheme, CAs are known as issuers and digital certificates follow the VC standard, not the X.509. The idea of having just one public key per identity, which does not allow rights, such as anonymity and the right to be forgotten, nor facilitates key rotation, recovery, or delegation, is replaced by the use of DIDs. In the SSI scheme, each identity can have an unlimited number of unique identifiers, or DIDs, and each DID can have an unlimited amount of various public keys and authentication mechanisms that are associated.

It is possible to use smart contracts for PKDs and DNS to link certain DIDs to particular entities or people. This MUST only be done with user´s consent, understanding that this link between a DID and an identity is being made public and will be persistently registered in the ledger where the smart contract lives. Some examples of when this can be interesting are a government or bank that wants their DID to be recognized in order to enable the verification of VCs they issue to individuals, or a journalist who wants their DID to be publicly associated with themselves because they have signed their articles with that DID and want it to be recognized as a seal of quality. These PKDs and DNS do not need to rely on centralized admins or infrastructure as they can be accessed in a decentralized way. In public permissioned networks, we encourage the Permissioning Committee (see the LACChain Framework for Permissioned Public Blockchain Networks[10]) to maintain a PKD with DIDs associated to every node operator.

Currently, in order for a certificate to be verified when a subject presents it to a verifier (see Section 2), it is necessary to go against a CRL or an OSCP to verify that the certificate was indeed issued and has not been revoked. These are usually maintained in a centralized way by the issuers and in order to protect themselves from denial of service (DOS) and other attacks, the issuers usually only allow for authenticated and authorized access. This is a strong limitation, because if, for example, a government issues a digital ID or vaccination credential to a person and then restricts access to the verification system to only a few entities (e.g., banks and insurance firms), the person will not be able to use their ID for daily tasks such as entering a restaurant which requires ID. Further, many credential issuers do not have the capacity to maintain centralized infrastructures for verification of digital certificates, such as universities, schools, NGOs, multilaterals, hospitals, pharmacies, supermarkets, stores, libraries, or corporations. Having the possibility to register the proofs of a VC in smart contract allows for on-chain PKDs and CRLs and enables anyone to verify the digital credential against any node connected to the network.

Step 3 of the LACChain ID Verification Process (see Section 2.5.3) consists of the verification of the issuer or issuers of a VC. As we mentioned previously, verifying an issuer might involve at least two steps. First, there is a cryptographic verification of the content signature to ensure that the signature was indeed issued by the entity acting as the credential issuer. Second, in all cases that require some level of assurance, it is necessary to verify that the issuer is an entity that is authorized to issue that credential. This verification process is in line with the concept of trust frameworks, as trust frameworks determine whether or not a particular verifier trusts a specific issuer.

In the case of digital vaccination certificates, entities that designate the authorized vaccination centers are, in general, Ministries of Health. The Ministries of Health designate these centers by maintaining trusted lists (TLs). If a verifier receives a verifiable presentation of a digital vaccination certificate, they will attempt to verify whether the issuer is indeed an entity that has been authorized by a recognized ministry of health. This situation is similar in the case of digital diplomas, digital IDs, driver licenses, and many other digital certificates and credentials.

Trust frameworks determine authorized issuers, and verifiers CAN decide whether they recognize an issuer or not, unless it is enforced by regulation. Trust frameworks lead to the proliferation of roots of trust that can resolve the chain from the issuer to trusted authority in order for the certificate to be trusted by a verifier. Both the rules of the trust frameworks and the roots of trust can be maintained both off-chain or on-chain using smart contracts.

Figure 13 illustrates some different combinations of off-chain and on-chain possibilities for the different elements of a digital identity scheme that we have been discussing. This framework endorses the use of DIDs as identifiers, VCs for digital certificates, and smart contracts for the PKD, TLS, roots of trust, and rules of the trust frameworks. But it also encourages the combination of the SSI standards with the use of X.509 certificates to be issued by institutional entities to issue VCs to individuals. Open-source implementations compliant with the LACChain ID Framework have already been made available by the LACChain Alliance.

| PKD | | Certificate standard | Credential verification | Identifier | TLs, Roots of Trust, and Trust Frameworks (rules) | |
|---|---|---|---|---|---|---|
| Off-chain | On-chain (smart contract) | X.509 | Off-chain CLR/ OSCP (http) | public key | Off-chain | |
| Off-chain | On-chain (smart contract) | X.509 | On-chain CRL/ OSCP (smart contract) | public key | Off-chain | On-chain (smart contract) |
| Off-chain | On-chain (smart contract) | VC | On-chain CRL/ OSCP (smart contract) | DID | Off-chain | On-chain (smart contract) |
| Off-chain | On-chain (smart contract) | VC | On-chain CRL/ OSCP (smart contract) | DID | Off-chain | On-chain (smart contract) |

*Figure 13. On-chain and off-chain possibilities of the different elements of a digital identity scheme.*

### 4.2.1. DNS and PKD

On-chain DNS or PKDs can serve as decentralized and public lists of entities behind identifiers, specifically DIDs, and there are different ways of accomplishing realizing them in a blockchain network. This SHALL only be done with public entities that want to establish a public disclosure of their DIDs. We would like to present three:

- **Trusted authority**: A trusted authority accomplishes identity proofing (i.e., verifies that a specific entity or person is indeed in control of a certain DID) and registers the specific entity or person's DID and identifiable information in the smart contract that represents the DNS or PKD.
- **Oracle:** An off-chain oracle automatically accomplishes verification after checking different types of documents provided by the entity or person (e.g., traditional X.509, biometric, etc.) and registers it in the smart contract that represents the DNS or PKD.
- **Code-based:** A set of smart contracts accomplish the same tasks on-chain that an oracle would accomplish off-chain. The advantage of on-chain accomplishment is that unlike the oracle, the smart contract does not need to be managed or operated by anyone. It can be a decentralized and totally transparent process. On the other hand, the computational capacity required for this process will be extremely high (probably too much to run on current blockchain networks) and unsupervised processes always leave use cases out (i.e., some entities and individuals with particular circumstances and characteristics not included in the smart contract would not be verified).

### 4.2.2. Roots of Trust

When using an SSI scheme, there are different options for resolving a root of trust that allows for verification that an issuer is authorized by a trusted authority to issue a specific VC. The two most relevant options that this framework encourages are the following:

- **Smart-contract-based:** If the DIDs of the trusted authorities authorized for the issuance of a particular VC are registered in smart contract-based PKDs and TL, it is possible to resolve against that smart the entire root of trust. As an illustrative example, when thinking about vaccination certificates, for each issuer it would be possible to call the PKD and TL smart contracts to verify if the issuers's DID is in the TL of a Ministry of Health, and the DID of the Ministry of Hhealth could also be checked against a PKD maintained for instance by the WHO.
- **Chain of VCs:** An alternative to using smart contracts is requiring each entity in the root of trust to send a VC that contains claims to proof its own identity and role to the entity below in the root of trust. Therefore, if the final issuer has 4 entities above to reach the root-CA or top trusted issuer, this final issuer would send to the subject 4 credentials (one for each issuer in the root of trust) plus the VC that that contains whatever attributes the final issuer is certifying to the subject. Then, each time the subject generates a verifiable presentation to the verifier, the subject needs to present all the identity credentials of the issuers -in this case 4- plus the VC that contains those attributes or claims that the subject is intended to proof to the verifier. In this case, the verifier does not go against a smart contract to resolve and verify the root of trust. Conversely, it does it off-chain using all the linked VCs. This process is similar to the way in which chains of X.509 certificates allow for creation of roots of trust today.

There are already implementations that establish a root of trust and a governance framework through the use of VCs.

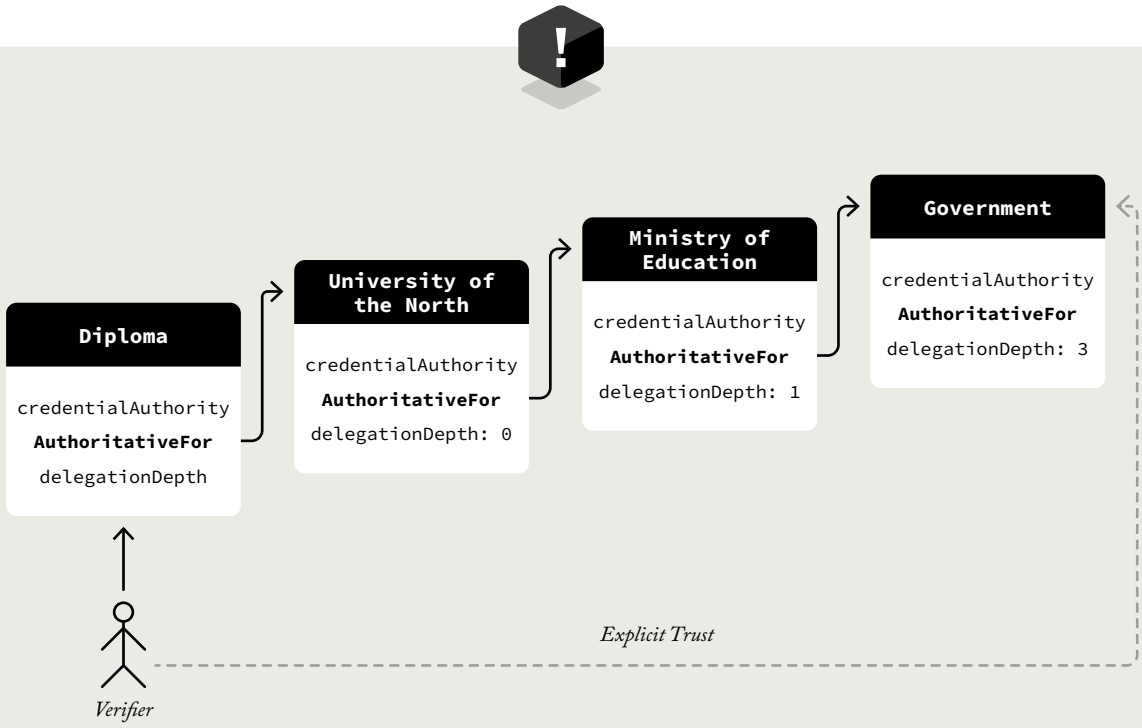


*Figure 14. KayTrust Root-of-Trust flow with Trusted Credentials.*

Figure 14 illustrates a root of trust solution using VCs, which not only verifies an entity or a subject, but also has authority to perform certain actions within a trust framework. This represents KayTrust Trusted Credentials, a unique and specific model which is presented within the same definition of VC to establish rules of «who» can be issued «what» credential type until reaching a trusted entity's point. Each VC specifies the entity authority to issue a certain type of credential within different levels.

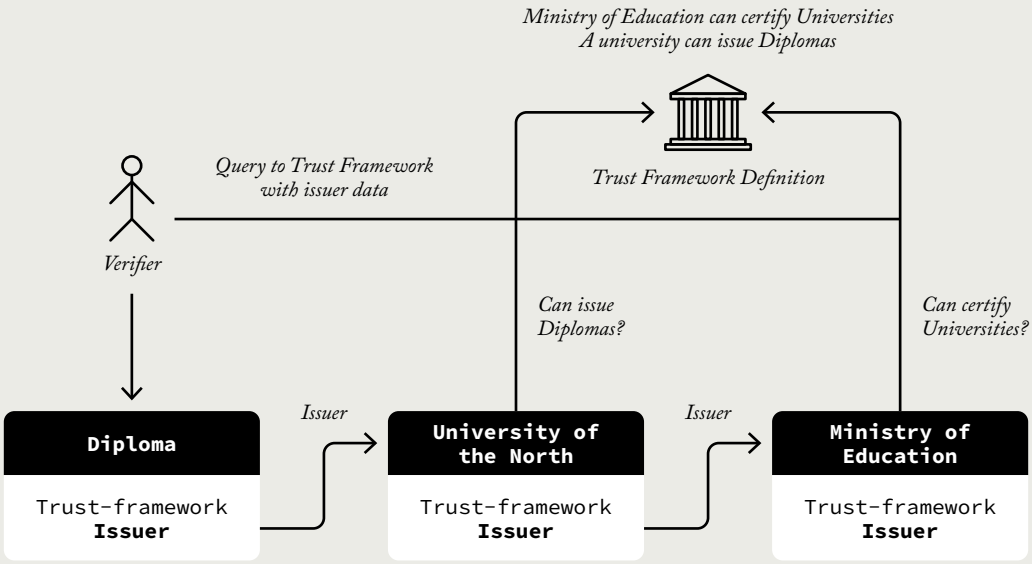


*Figure 15. Aries RFC 0430: Machine-Readable Governance Frameworks.*

Figure 15 illustrates a set of rules that defines a trust framework that allows all the necessary artifacts to verify if an issuer is endorsed by a specific trust framework. This is the case with Aries RFC 0430: Machine-Readable Governance Frameworks, where a new scheme is presented in JSON format that serves as a goal structure influencing the behavior of VCs. The set of rules that can be defined depends on the technology used (smart contracts, VCs, etc.), allowing artifacts related to SSI to operate in multiple trust frameworks. Some regions of Canada have implemented a ToIP-compatible VC registry service, such as Verifiable Organizations Network[17] and OrgBook BC[18].

LACCHAIN ID FRAMEWORK

# 5
# REGULATION

IDB IDB | LAB LACCHAIN

Within the regulation layer, regulations on electronic elements such as signatures, transactions, certificates, timestamps, and documents, and regulations on data privacy and protection are considered.

## 5.1. Regulation on Electronic Transactions, Signatures, Documents, and Timestamps

The SSI model relies on the cryptography of immutable and decentralized ledgers, digital signatures of transactions and digital credentials, and timestamps. Fortunately, from a regulatory perspective, these topics are not new. However, the proliferation of SSI depends on the recognition of the legal value of elements such as blockchain networks, DIDs, VCs, and digital wallets. The necessary steps to move from current regulations on electronic identification and authentication to enhanced versions that recognize the new elements introduced by SSI need to be outlined for each regulation individually. Countries that currently lack regulatory policies still have the opportunity to catch up with those that do. SSI solutions SHALL be designed and implemented in compliance with regulations on electronic transactions, signatures, documents, and timestamps.

The European Union has the most advanced and globally recognized regional regulation on electronic transactions, signatures, and documents to date. Adopted on the July 23, 2014,[19] the regulation 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS) "provides a predictable regulatory environment to enable secure and seamless electronic interactions between businesses, citizens and public authorities.[20]

The European Commission has launched a regional blockchain network named EBSI managed by Country Members and intended to serve to provide services to citizens. As an essential piece of this initiative, there is an ongoing project called the "eIDAS Bridge" component that is now being developed in the context of the European Self-Sovereign Identity Framework (ESSIF). The eIDAS ESSIF Bridge is aimed to "make eIDAS available as a trust framework in the SSI ecosystem".[21][22]

On June 3rd 2021, the European Commission released a proposal amendment to the eiDAS regulation arguing that "What is emerging In the market is a new environment where the focus has shifted from the provision and use of rigid identities to the provision and reliance on specific attributes related to those identities. […] The eIDAS Regulation revealed that the current Regulation falls short of addressing new market demands, mostly due its inherent limitations to the public sector, the limited possibilities and the complexity for online private providers to connect to the system, its insufficient availability of notified eID solutions in all Member States and its lack of flexibility to support a variety of use cases".[23]

This amendment also states that "electronic ledgers provide users with proof and an immutable audit tail for the sequencing of transactions and data records, safeguarding data integrity. While this trust service was not part of the impact assessment, it builds upon existing trust services as it combines time stamping of data and their sequencing with certainty about the data originator, which is similar to e-signature. This trust service is necessary to prevent fragmentation of the internal market, by defining a single pan-European framework that enables the cross-border recognition of trust services supporting the operation of qualified electronic ledgers. Data integrity, in turn, is very important for the pooling of data from decentralized sources, for self-sovereign identity solutions, for attributing ownership to digital assets, for recording business processes to audit compliance with sustainability criteria and for various use cases in capital markets".

## 5.2. Regulation on Data Protection and Privacy

In an increasingly digital world, it is essential to protect people's data and privacy. The best way to enforce these protections is through regulations. Unfortunately, many countries either lack or have outdated data protection regulations. We believe that if SSI solutions are implemented in compliance with the LACChain ID Framework, they can be completely compliant with the most advanced data protection regulations in the

world. To further make this point evident, we will focus on the following six converging areas between advanced data protection regulations and SSI.

- **Consent:** Implementations SHALL enable consent at all times by requesting the subject to authorize any use of their data in compliance with data protection regulations.
- **Data portability:** Data portability is provided by digital wallets, where an individual can store their keys, credentials, and data. SSI solutions CAN enable data portability through the use of mobile wallets.
- **Data protection by design and by default:** All aspects of the SSI model presented in this framework, including DIDs, VCs, verifiable presentations, identification, authentication and authorization, digital repositories and wallets, and a decentralized registry, SHALL be designed to protect data by default.
- **Pseudonymization:** DID registries and DID methods SHALL guarantee pseudonymization (see Section 1). These allow for subject or holder to manage as many pseudonymous identifiers as desired so that they can interact with various services securely. They SHALL authenticate without revealing more data or PII than what is consented by subjects. Functionalities such as selective disclosure mechanisms and ZKP SHOULD be enabled (see Section 2.6).
- **Records of processing activities:** Digital wallets CAN keep a private record of processing activities. Public and decentralized blockchain registries allow for more pseudonymous traceable data; it SHALL NOT be possible to associate identifiers with the real subjects behind. developed. In all cases, data privacy SHALL be preserved, including the PII that could be derived from exchanges and verifications.
- **Right to erasure (right to be forgotten):** users SHOULD (i) know exactly where the data is, (ii) be able to authenticate themselves to those who own their data so they can ask them to erase it, and (iii) not have personal data in immutable and decentralized registries.

Arguably the most advanced international regulation on data privacy is the General Data Privacy Regulation (GDPR). This is a European egulation, in effect as of May 25th, 2018, to all states that are members of the European Union, put in place in order to "harmonize data privacy laws across Europe"[24].

LACCHAIN ID FRAMEWORK

# 6 TRUST FRAMEWORKS

IDB IDB | LAB LACCHAIN

A trust framework is a "generic term often used to describe a legally enforceable set of specifications, rules, and agreements that govern a multi-party system established for a common purpose, designed for conducting specific types of transactions among a community of participants, and bound by a common set of requirements. [...] They are referred to as operating regulations, scheme rules, or operating policies in contexts different from digital identity".[25]

The scope of a trust framework[8] spans from recognition within single organizations or groups of entities, to regional, sectorial, and international agreements. An example of a national trust framework is a national ID, which establishes government sovereignty for the issuance of identity credentials. An example of a regional framework is the international recognition of national passports that follow standards dictated by the International Civil Aviation Organization (ICAO). Examples of sectorial frameworks include the mutual recognition agreements (MRA) between customs, the settlement networks between financial institutions, and the recognition of certifications between universities.

In a digital identity ecosystem, self-sovereign or not, a trust framework defines the governance model, the certificate authorities[9], the identity providers (i.e., issuers in the SSI context), the levels of assurance, and the communication channels, among others. This allows for the establishment of all necessary elements of trust that we discussed in Section 4.

## 6.1. Levels of Assurance

Trust frameworks usually define different levels of assurance (LOAs) depending on who issued electronic certificates and how they were issued. One of the most reputable frameworks for levels of digital identity assurance comes from the International Organization for Standardization (ISO), presented in Figure 17. In order to evaluate and recognize the new standards, protocols, and technologies brought together under the SSI umbrella, the LOAs defined by current trust frameworks may require to be review. SSI Implementations SHALL indicate their levels of assurance and be compliant with the ISO/IEC 29115[26] standard to ensure compatibility and interoperability".

8 There is also a good reference by NIST that can be retrieved from https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8149.pdf

9 The concept of certificate authority is equivalent to the concept of credential service provider.

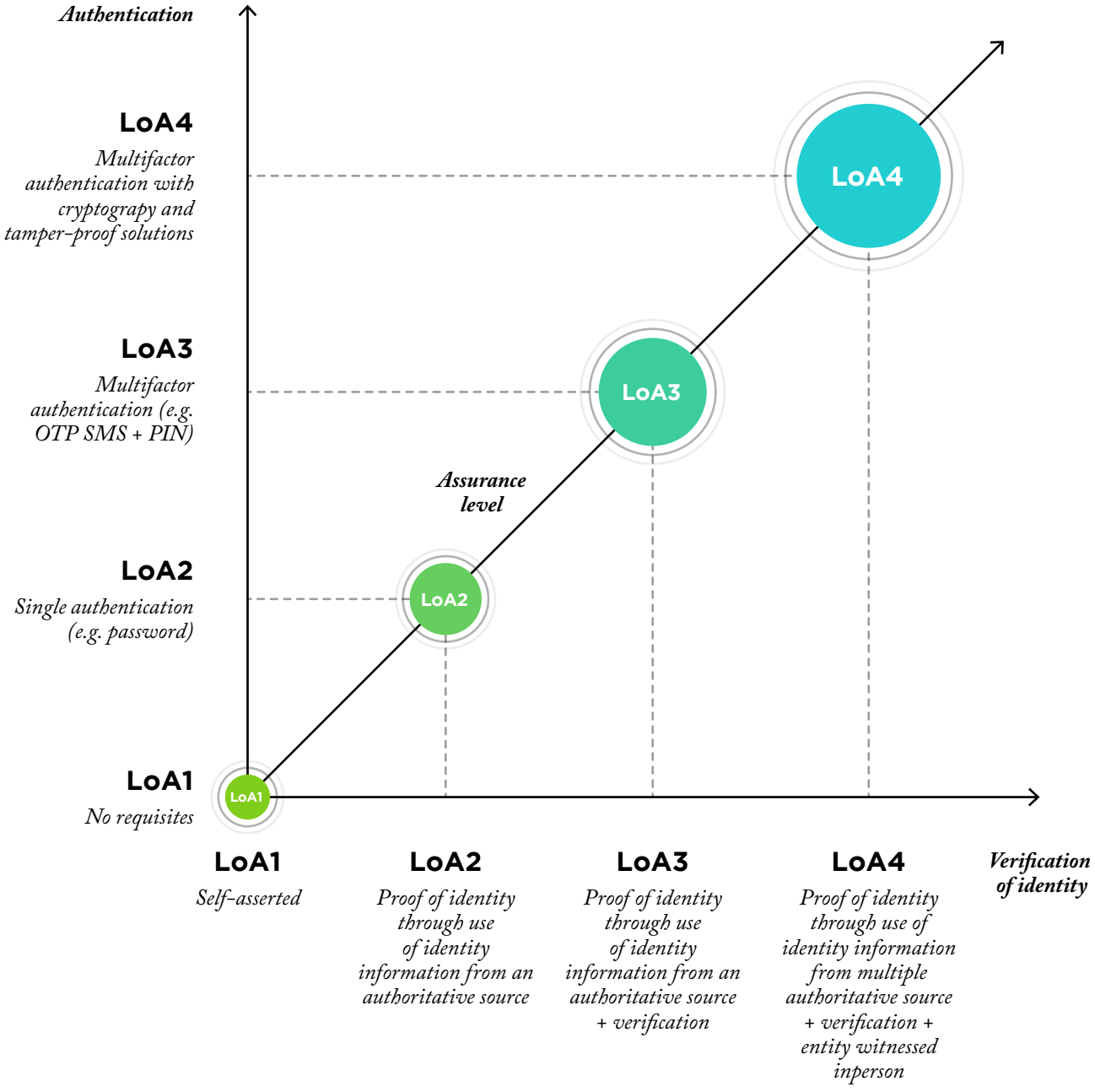


*Figure 17. Levels of assurance of ISO/IEC DIS 29115*[26].

## 6.2. Independent Elements of Governance

A governance model establishes principles, policies, terminology, standards, and responsibilities. Frameworks establish agreements and "rules of the game", and governance models define the roles and responsibilities of the "game's players". In current implementations of digital identity schemes, a governance model usually establishes who the CAs (i.e., issuers) are and where the TLs, CRLs, and PKDs can be found. In SSI the governance model CAN define the same things, but under this identity scheme there are differences in the technological tools that MAY introduce new elements of governance, such as decentralized ledgers where the proofs of VCs are stored and the TLs and PKDs are maintained. The LACChain ID Framework proposes the division of independent elements of governance into five categories, as presented in Figure 18, assuming that the trusted registries are blockchain networks: blockchain networks (access/permissioning); blockchain networks (block generation); TLs, PKDs, and roots of trust; DID registries; management of keys and credentials.

**Blockchain networks - access/permissioning:** In permissionless networks there are no permissioning requirements, and therefore governance does not apply. In permissioned networks, both private and public, there is a process to access the network that encompasses certain requirements. When permissioned blockchain networks are as trusted registries for SSI solutions, permissioning requirements to access these networks SHALL be defined in a way that maximizes accessibility, security, and interoperability for end users.

**Blockchain networks - block generation:** In both permissioned and permissionless networks, blocks are generated according to certain rules that are known as the consensus protocol. The consensus protocol in blockchain networks that are used as trusted registries for SSI solutions SHALL NOT anyone to rewrite history without the consent of all the entities that have identity-related information stored in the network.

**TLs, PKDs, and roots of trust:** The set of elements that allow to verify that a VC has been issued by an authorized issuer, including TLs, PKDs, and roots of trust, MAY be stored in smart contracts living in blockchain networks. These smart contracts SHALL establish management rules that allow only authorized entities to register and update information.

**DID registries:** DID registries MAY be maintained in smart contracts living in blockchain networks. DID Registries SHALL NOT allow unauthorized entities to modify information about a DID; only the subject or a delegated SHALL be able to modify DID-related information in the DID registry. DID registries SHALL be designed in a way that allow individuals to erase or request issuers to erase their DID

related information. DID registries SHALL respect data privacy be compliant with data protection regulations. Only when a person or entity explicitly decides to reveal the identity behind a DID, understanding the consequences related to it, DID registries MAY allow to expose real identities behind DID identifiers.

**Keys and credentials:** Keys and credentials related to individuals and entities SHALL be managed by them or by an entity they have given consent to. Individuals SHOULD use digital wallets to manage keys and credentials. Digital wallets SHALL comply with data privacy and data protection regulations.

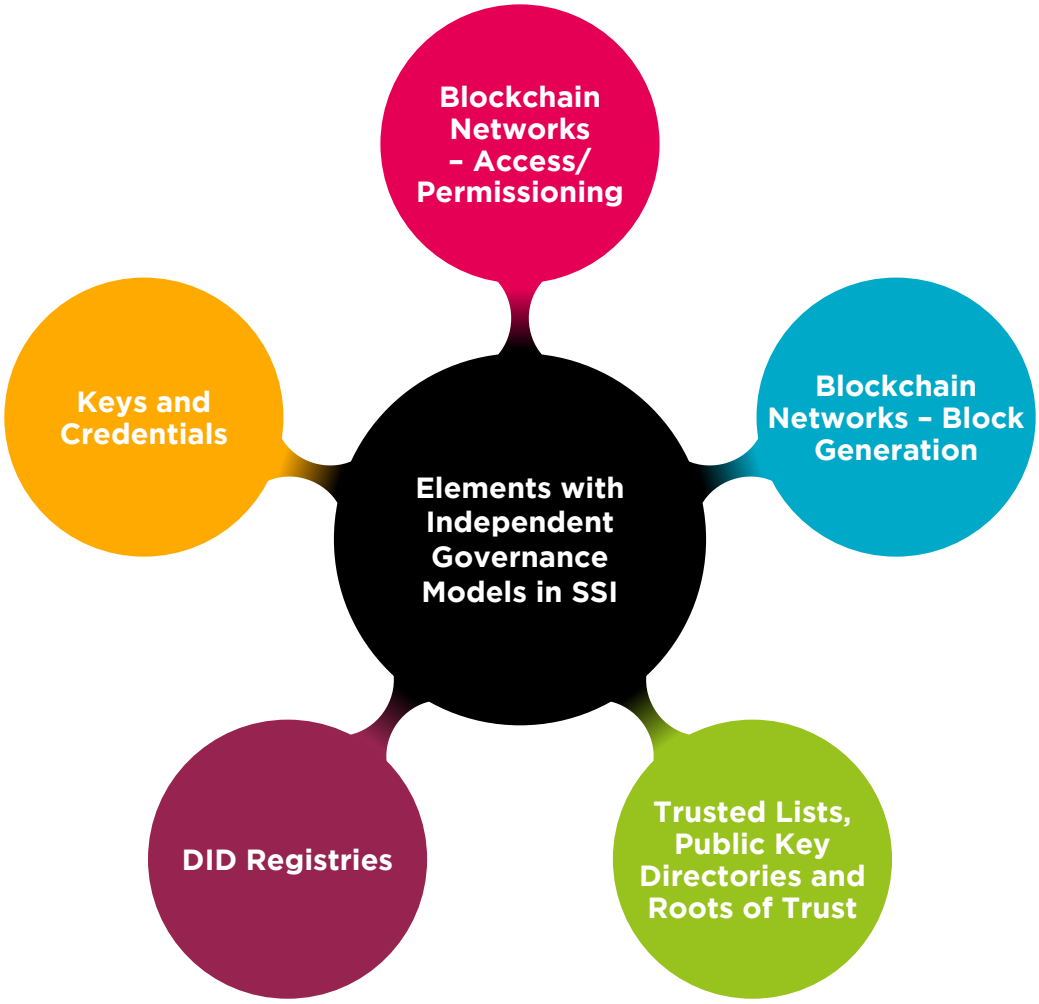


*Figure 18. Elements with independent governance models in SSI implementations.*

# APPENDIX. Example of the Camenisch-Lysyanskaya ZKP Algorithm Using Verifiable Credentials and Presentations

In Figures 19 and 20, we have presented a verifiable credential (VC) and a verifiable presentation, respectively, generated in compatibility with the Camenisch-Lysyanskaya (CL)[27] ZKP algorithm. This example illustrates the issuance of a LACChain diploma in the format of a VC that contains personal information about the subject ("givenName" and "familyName") and the certified degree ("holds"). In the verifiable presentation, only claims related to the certified degree are disclosed, and not personal information about the subject. In order for this to take place, when a VC is issued, according to the CL algorithm, it is necessary to specify the "credentialSchema" field, which points to the definition of fields described in the "credentialSubject", so both the issuer and the verifier can verify the data type of each field and thereby execute the necessary ZKP algorithms. The "credentialSchema" field extends the definition of the credential in the "proof" section. Following this strategy, the verifier can verify not only the authenticity of the presentation itself, but also the issuer's wrapped claims. This eliminates the need for the original issuer to reissue each claim separately and allows the subject to generate as many verifiable presentations as desired containing only subsets of verifiable claims.

```
{
   “@context”: [
      “https://www.w3.org/2018/credentials/v1”,
      “https://www.w3.org/2018/credentials/examples/v1”
   ],
   “type”: [“VerifiableCredential”, “Certificate”],
   “credentialSchema”: {
      “id”: “did:lac:main:0x094abde76419f428014d1386ae3e6184d423324”,
      “type”: “LACChainAcademyCertificate”
   },
   “issuer”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”,
   “credentialSubject”: {
      “givenName”: “Juan”,
      “familyName”: “Perez”,
      “holds”: {
         “type”: “LACChainAcademyCertificate”,
         “name”: “Introduction to LACChain”,
         “college”: “LACChain Academy”
      }
   },
   “proof”: {
      “type”: “CLSignature2019”,
      “issuerData”: “5NQ4TgzNfSQxoLzf2d5AV3JNiCdMaTgm...
      BXiX5UggB381QU7ZCgqWivUmy4D”,
      “attributes”: “pPYmqDvwwWBDPNykXVrBtKdsJDeZUGFA...
      tTERiLqsZ5oxCoCSodPQaggkDJy”,
      “signature”: “8eGWSiTiWtEA8WnBwX4T259STpxpRKuk...
      kpFnikqqSP3GMW7mVxC4chxFhVs”,
      “signatureCorrectnessProof”: “SNQbW3u1QV5q89qhxyVqFa6jCrKwv...
      dsRpyuGGK3RhhBUvH1tPELH”
   }
}
```


*Figure 19. Example of a VC representing a LACChain diploma compatible with the Camenisch-Lysyanskaya ZKP algorithm.*

```
{
   “@context”: [
      “https://www.w3.org/2018/credentials/v1”,
      “https://www.w3.org/2018/credentials/examples/v1”
   ],
   “type”: “VerifiablePresentation”,
   “verifiableCredential”: [
      {
         “@context”: [
            “https://www.w3.org/2018/credentials/v1”,
            “https://www.w3.org/2018/credentials/examples/v1”
      ],
      “type”: [“VerifiableCredential”, “Certificate”],
      “credentialSchema”: {
         “id”: “did:lac:main:0x094abde76419f428014d1386ae3e6184d423324”,
         “type”: “did:lac:main:0x108e206f6ec5b7124102a14f6addc0b9300630ee”
      },
      “issuer”: “did:lac:main:0xadf1702b76419f428014d1386af487b2d8145f83”,
      “credentialSubject”: {
         “type”: “LACChainAcademyCertificate”,
         “college”: “LACChain Academy”
      },
      “proof”: {
         “type”: “AnonCredDerivedCredentialv1”,
         “primaryProof”: “cg7wLNSi48K5qNyAVMwdYqVHSMv1Ur8i...
         Fg2ZvWF6zGvcSAsym2sgSk737”,
         “nonRevocationProof”: “mu6fg24MfJPU1HvSXsf3ybzKARib4WxG...
         RSce53M6UwQCxYshCuS3d2h”
      }
   }],
   “proof”: {
      “type”: “AnonCredPresentationProofv1”,
      “proofValue”: “DgYdYMUYHURJLD7xdnWRinqWCEY5u5fK...
      j915Lt3hMzLHoPiPQ9sSVfRrs1D”
   }
}
```


*Figure 20. Example of a ZKP verifiable presentation from a LACChain diploma compatible with the Camenisch-Lysyanskaya ZKP algorithm.*

# REFERENCES


1 M. Allende. (2020) Self-sovereign identity: The future of identity: Self-Sovereign Identity, Digital Wallets, and Blockchain. Inter-American Development Bank (IDB). DOI: http://dx.doi.org/10.18235/0002635

2 LACChain DID Developer Portal. Retrieved from https://dev.lacchain.net/ssi/lac-did

3 Ethr-DID. Uport-project. Github repository. Retrieved from https://github.com/uport-project/ethr-did

4 M. Sporny et all. Decentralized Identifiers (DIDs) v1.0. World Wide Web Consortium (W3c). Retrieved from https://www.w3.org/TR/did-core/

5 LACChain-did-registry. LACChain. Github repository. Retrieved from https://github.com/lacchain/lacchain-did-registry

6 Universal-resolver. LACChain. Github repository. Retrieved from https://github.com/lacchain/universal-resolver

7 M. Sporny et all. Verifiable Credentials Data Model 1.0. World Wide Web Consortium (W3C). Retrieved from https://www.w3.org/TR/vc-data-model/

8 Mailbox. LACChain. Github repository. Retrieved from https://github.com/lacchain/mailbox

9 LACChain ID Credentials Schema Repository: https://credentials.library.lacchain.net/

10 M. Allende. (2021) LACChain Framework for Permissioned Public Blockchain Networks. Inter-American Development Bank. Retrieved from https://publications.iadb.org/en/lacchain_blockchain_framework

11 M. Sporny et all. Verifiable Credentials Data Model 1.0 – Verifiable Presentations. World Wide Web Consortium (W3C). Retrieved from https://www.w3.org/TR/vc-data-model/#dfn-verifiable-presentations

12 EIPs. Ethereum. Github repository. Retrieved from https://github.com/ethereum/EIPs/blob/master/EIPS/eip-712.md

13 EIP-812. Ethereum. Github repository. Retrieved from https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1812.md

14 L. Lesavre et all. (2020) A taxonomic approach to understanding emerging blockchain identity management systems. National Institute for Standards in Technology (NIST). Retrieved from https://csrc.nist.gov/publications/detail/white-paper/2020/01/14/a-taxonomic-approach-to-understanding-emerging-blockchain-idms/final

15 LACChain Developer Portal. Authentication. Retrieved from https://dev.lacchain.net/en/working-groups/ssi/authentication

16 OpenID Connect Core 1.0. Specification. Retrieved from https://openid.net/specs/openid-connect-core-1_0.html

17 https://www.vonx.io/

18 www.orgbook.org.bc.ca

19 Agencia Estatal Boletín Oficial del Estado. Gobierno de España. https://www.boe.es/

20 eIDAS Regulation. European Commission. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation

21 About SSI eIDAS Bridge. European Commission. Retrieved from https://joinup.ec.europa.eu/collection/ssi-eidas-bridge/about

22 N. Alamillo. (2020). SSI eIDAS Legal Report. European Commision. Retrieved from https://www.blockchain4europe.eu/wp-content/uploads/2021/05/SSI_eIDAS_legal_report_final_0.pdf

23 Proposal for a regulation of the European parliament and of the council amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity. European Commission. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0281

24 General Data Protection Regulation. European Commission. Retrieved from https://gdpr-info.eu/

25 https://openidentityexchange.org

26 Peer DID Method Specification. World Economic Forum (W3C). Retrieved from https://identity.foundation/peer-did-method-spec/

27 J. Camanisch and A. Lysyanskaya. An efficient system for non-traceable anonymous credentials with optional anonymity revocation. Retrieved from https://www.iacr.org/archive/eurocrypt2001/20450093.pdf

# ACKNOWLEDGEMENTS


Adrian Pareja, Blockchain Lead Architect, LACChain, Perú.

Albi Rodríguez, Head of Ecosystems and Communities, LACChain, Spain.

Alecs Garza, CTO & Co-founder, OS City, Mexico.

Anantharaman Iyer, Lead - Strategy Partnerships and Alliances, Tata Consultancy Services, India.

Andres Gomez Ramirez, Blockchain Security Researcher, EOS, Costa Rica.

Andres Junge,Co-Founder and CTO, Notabene, Chile.

Annamalai (Anbu) Anbukkarasu, Head, Digital & Emerging Technology, Banking Financial Services and Insurance, Tata Consultancy Services, India.

Antonio Leal Batista, Head of Project Pipeline, LACChain, Perú.

Blanca Sandoval, Communication Manager, LACChain, Mexico.

Bob Trojan, President & CEO, Token Insights / Financial Services Insights / DC Insights, LLC, USA.

Brian Desiderio, Technical Leader, Extrimian, Argentina.

Carolina Maldonado, Global Business Relations Manager, VU Security, Argentina.

Chris Fergus, Analytics, Metrika, USA

Daniel Zárate, Communication associate, LACChain, Mexico.

David Ammouial, Blockchain & Digital Identity Expert Architect, Everis-NTTData, France.

Diego López León, Blockchain Architect, LACChain, Argentina.

Drummond Reed, Chief Trust Officer, Evernym, USA.

Edgard Espinoza, Lead Software Architect Backend, Everis-NTTData, Perú.

Eduardo Marchena, DevOps, LACChain, Perú.

Eduardo Lemp, Operations Advisor, LACChain, Chile.

Erick Pacheco Pedraza, Blockchain Architect, Everis-NTTData, Perú.

Gene Vayngrib, CEO & Co-founder, Tradle, USA.

Guillermo Villanueva, Founder and CEO, Extrimian, Argentina.

Ilán Meléndez Lugo, Regional Lead, LACChain, Costa Rica.

Ismael Arribas, Owner, Kunfud, Spain.

Itzel Nava, Executive Coordinator of LACChain, LACChain, Mexico.

Iván Castelán, Software Development Engineer, OS City, Mexico.

Jaime Alberto Centellas, Chief Technology Officer, World Data Inc., USA.

Jesús Cepeda, CEO & Co-founder, OS City Mexico.

Juan José Miranda, Director Digital Technology Innovation and Labs Blockchain Everis-NTTData, Perú.

Kennedy Roman, Commercial Director Caribbean and Central America Region, VU Security, Argentina.

Lucas Jolias, CSO & Co-founder, OS City, Argentina.

Lucía Latorre, Marketing and Commercial Advisor, LACChain, USA.

Mahesh Chandrahas Karajgi, Senior IT Officer, ITS Technology & Innovation, World Bank Group, USA

Marcela Ribeiro Gonçalves, Diretora de Desenvolvimento Empresarial e Sócia, Multiledgers, Brazil.

Martin Hargreaves, Chief Product Officer, Quant Network, UK.

Matías Oveja Smith, Tech Lead, Proyecto DIDI, Argentina.

Máximo García Martínez, Blockchain Developer, Inetum, Spain.

Miguel Gómez Carpena, Blockchain Consultant, Izertis, Spain.

Moisés Menéndez, Principal Advisor in Legal and Economics, LACChain, Spain.

Oscar Bazoberry, CEO & Founder, World Data Inc., USA.

Pablo Luna, Devops, Extrimian, Argentina.

Pablo Mosquella, Chief of Staff, Extrimian, Argentina.

Pabo Raices, Regional Lead for the Southern Cone, LACChain Uruguay.

Pallavi Kumari, Tata Consultancy Services, India.

Pamela Martínez Licona, Full Stack Developer, OS City, Mexico.

Pedro Perrotta, Co-founder and Chairman, Grupo Sabra, Argentina.

Rocío Murillo Mora, IT Coordinator, LACChain, Costa Rica.

Raunak Mittal, IT Officer, ITS Technology & Innovation Lab, World Bank Group, USA

Saurabh Nagesh,Delivery Manager - Cloud, Tata Consultancy Services, India.

Sergio Bazoberry, Head of Operations & Co-founder, World Data Inc., USA.

Simon Wilkinson, Operations Director, Tradle, USA.

Sooin Yoon, Product, Metrika, USA

Sosu Alex, Associate Consultant - Blockchain Architect, Tata Consultancy Services, India.

Suzana Maranhão Moreno, Co-founder, BNDES Digital Innovation Lab, Brazil.

Tamara Bagdassarian, Coordinadora IT, Proyecto DIDI, Argentina.

Urko Larrañaga, Lead of Blockchain, Izertis, Spain.

Venu Reddy, Blockchain Consultant, Digital Solutions, BFSI US East, Tata Consultancy Services, USA.

Veronica Massera, Technical Leader, Grupo Sabra, Argentina.

Yusuf Karacaoglu, Chief Technology Advisor ITSVP, Director of Technology and Innovation, World Bank, USA.

# LACCHAIN ID FRAMEWORK

IDB IDB | LAB LACCHAIN