A Digital Identity in the Hands of Swiss Citizens

Jean-Luc Beuchat[*] and Valon Rexhepi[†]

University of Applied Sciences and Arts Western Switzerland (HES-SO), Sierre, Switzerland

2024-04-15 — Version #2bd24a

Abstract

The Swiss law on electronic identity (LSIE) was rejected on March 7, 2021. Its opponents accused it of involving private companies which could thus collect citizens' data and store them centrally. Six motions with identical wording were tabled on March 10, 2021: they all ask the Swiss Federal Council to set up a state-run system allowing citizens to prove their identity online in complete confidence. They stipulate that only necessary information is collected and stored in a decentralized manner. The Swiss Federal Council has recommended to Parliament to approve these motions on May 26, 2021, and wishes to propose a new e-ID solution responding to citizens' concerns as soon as possible. The Federal Department of Justice and Police has been asked to draw up a first draft presenting several technical solutions and specifying their respective costs. Following the publication of a working document on September 2, 2021, a public consultation was opened. It ended on October 14, 2021, with a public debate organized at the Government House in Bern and broadcasted live on a virtual platform. Self-Sovereign Identity (SSI) is one of the solutions identified during this process. It gives the citizens control of their electronic identity: they hold credentials issued by public administrations and choose the data they wish to disclose when they authenticate with a service (they can for example prove that they are over 18 without specifying their exact date of birth).

We propose here a decentralized and user-centric e-ID system based on SSI principles. Our solution embraces an open-source philosophy, fostering transparency and community involvement. We employ blockchain technology as a design pattern to establish trust and ensure the immutability of identity-related data. By design, our solution ensures the right to be forgotten by exclusively storing the digests of verifiable credentials on the blockchain. To demonstrate the feasibility and effectiveness of our SSI solution, we have developed a proof of concept leveraging the Partisia blockchain.

1 Introduction

In May 2015, the Swiss Federal Office of Justice (FOJ) opened an informal consultation involving large companies and the cantons on a state-recognized electronic identity (e-ID). Such a tool is indeed essential to the proper functioning of an e-government. The results of the debates as well as the examination of similar initiatives launched in other countries indicated that an e-ID developed by a state implies higher IT costs than those of the private sector. In addition, a state does not have the flexibility required to adapt quickly to market needs and/or to changing technologies. The role of the Swiss Confederation would therefore be limited to defining the legal framework and verifying the identity of citizens. The FOJ then drafted a bill on e-ID services (Swiss law on electronic identity or LSIE) which was adopted by the Parliament during its autumn 2019 session.

The LSIE proposed a centralized architecture in which private companies played the role of Identity Providers (IdPs). In accordance with the results of the 2015 consultation, the development and operation of e-ID solutions are entrusted to private companies recognized by the Confederation. The latter intervenes:

- Before issuing a new e-ID: anyone who wants an e-ID applies to fedpol via an approved Identity Provider.

[*] jean-luc.beuchat@hevs.ch

[†] valon.rexhepi@hevs.ch

- When updating credentials managed by an Identity Provider. The frequency of this operation depends on the level of guarantee of the e-ID: annually for the low level, quarterly for the substantial level, and weekly for the high level.

The LSIE quickly had many detractors whose main criticisms were:

- Any private company acting as an IdP can centrally record every use of an e-ID. The IdP is also a single point of failure: an attacker could block the service and/or steal the data belonging to the citizens.
- The e-ID is a unique identifier for services that have nothing to do with each other. No one can guarantee zero risk in the face of credential loss or theft.
- The Confederation is merely a data provider.

The LSIE was rejected by the Swiss people on March 7, 2021. Six identically worded motions were tabled on March 10, 2021 [And21; Grü21; Mar21; Mäd21; Sta21; Gro21]. They ask the Federal Council to set up a state-run system allowing citizens to prove their identity online in complete confidence. Several criteria are defined:

- The issuance of e-IDs and the operation of the system are assumed by public services.
- Private companies can provide products or services necessary for the operation of the e-ID.
- Citizens can use their e-ID for interactions with the public and private sectors.
- Only the necessary information is collected and stored in a decentralized manner.

The Federal Council has recommended to Parliament to approve these motions on May 26, 2021, and wishes to propose a new e-ID solution responding to citizens' concerns in May 2022. The FOJ was asked to draw up a first document [Dép21] presenting several technical solutions and specifying their respective costs. Three approaches were proposed:

- The first solution, already suggested by opponents of the LSIE during public debates, is to give the Confederation the role of Identity Provider. In this scenario, the Confederation develops the solution and processes all authentication requests. It can thus learn the browsing habits of its citizens. Correlating authentications with e-banking services to tax data then makes it possible to unmask citizens who do not declare all of their accounts.
- The Federal Office of Information Technology, Systems and Telecommunication (FOITT) operates a Public Key Infrastructure (PKI) used by the Swiss authorities and several organizations close to the Confederation. Examples of application include the COVID certificate or electronic signature guaranteeing the integrity of the document as well as the identity of the signatory.

  This PKI could also manage electronic identities. The citizen generates an asymmetric key pair, then transmits their public key and the information needed to verify his identity to the identity to the Confederation. The e-ID is then issued in the form of an X.509 certificate. The citizen must carefully store their certificate and private key. In the event of loss or theft, they must promptly revoke the certificate.

  Every service provider trusts the Confederation and has a copy of its root certificate. It also keeps an up-to-date list of revoked certificates. To access an online service, the citizen presents their certificate. If the latter has not been revoked and the signature is valid, the citizen is authenticated and can access the service. Unlike the centralized solution described above, the Confederation has no knowledge of the transactions carried out by the owner of an e-ID. This system is decentralized, and the citizen is responsible for their electronic identity.
- With the solutions presented so far, the user transmits all the information linked to his e-ID at each authentication. According to the GDPR, a company must only process and store the data that is essential for its business. Thus, an online liquor store must ensure that a customer is over 18 and

does not need to know the exact date of birth. Self-Sovereign Identity (SSI) is a new decentralized e-ID model. C. Allen gives the following definition: “Self-sovereign identity is the next step beyond user-centric identity and that means it begins at the same place: the user must be central to the administration of identity. That requires not just the interoperability of a user’s identity across multiple locations, with the user’s consent, but also true user control of that digital identity, creating user autonomy. To accomplish this, a self-sovereign identity must be transportable; it can’t be locked down to one site or locale” [All16].

It is essential for us to preserve the private sphere of the citizens and the most convincing solution is SSI. It is also in line with the European General Data Protection Regulation (GDPR) and the New Federal Act on Data Protection (nFADP) which will come into force in September 2023. Our goal is to build a prototype that meets the expectations of Swiss citizens. Our system must:

- Guarantee the user the control of their data.
- Be auditable (i.e. open-source).
- Easily integrate into an existing portal (for example to replace a traditional identity provider). Our solution must therefore be compatible with OpenID Connect and/or SAML.

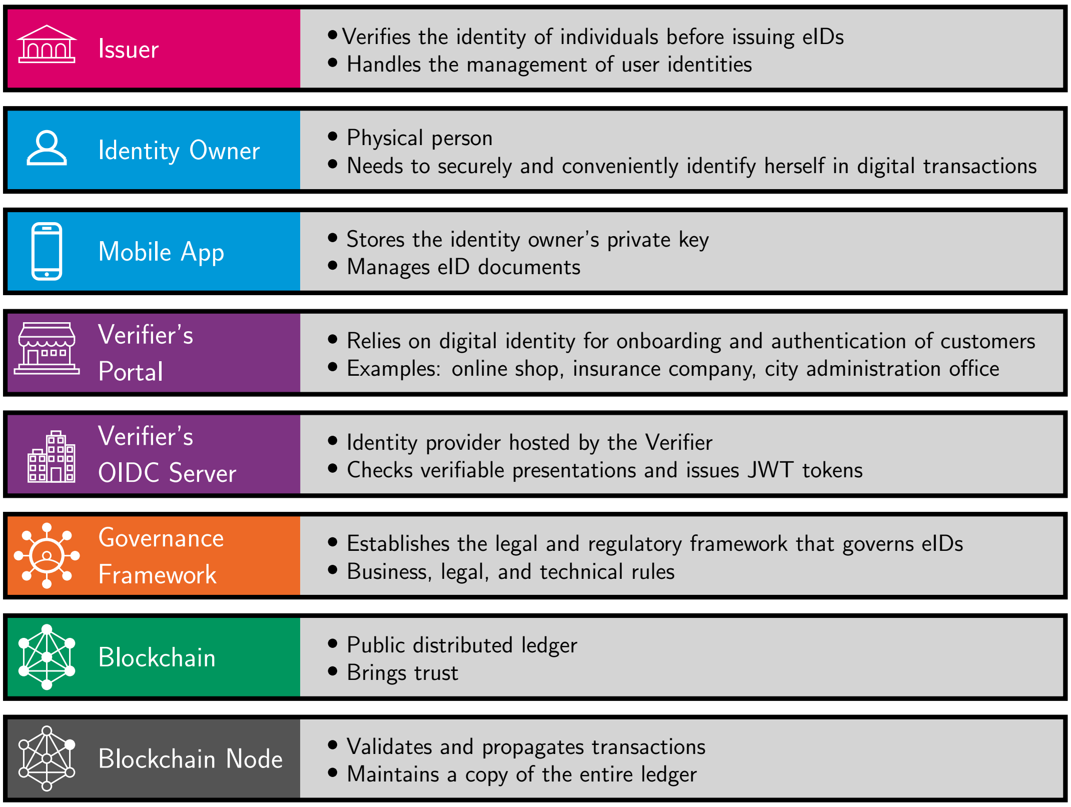
The rest of the article is organized as follows. First, we delve into the details of the trust triangle (Section 2), explaining the roles and interactions of the Governance Framework, blockchain, Identity Owner, Issuers, and Verifiers. Next, we explore the concept of verifiable credentials (VCs) (Section 3), discussing their structure and how they enable selective disclosure of identity attributes. We describe our smart contract, examining its role as a secure repository for storing digests of VCs. Then, we focus on the specific roles of the issuer and verifier, elucidating their responsibilities in the issuance and verification processes (Section 4). Following that, we provide a comprehensive overview of our proof of concept conducted using the Partisia blockchain (Section 5). Furthermore, we present a first analysis of the security of our solution (Section 6). Finally, we discuss the future developments of our solution (Section 7).

2 Trust Triangle

Figure 1 describes the actors of our system. Identity owners are individuals or entities who have control over their own digital identities. We assume that each Identity Owner has a mobile application to manage their electronic identities. Issuers are trusted entities responsible for establishing the trustworthiness of Identity Owner and generating their credentials. Verifiers are the parties relying on those credentials to make informed decisions about granting access or privileges. Web portals often use an identity provider (IdP) to authenticate users. Our solution integrates with the IdP. That’s why we model the Verifier using two entities, namely a Portal and an OIDC Server. To guarantee privacy, it is important that both entities are managed by the Verifier: a third party would learn the user’s connection history to the portal. A blockchain is a distributed digital ledger that records transactions across multiple computers or nodes [Nak09]. Each transaction is stored in a block that is cryptographically linked to previous blocks, creating a chronological chain of data. The decentralized nature of blockchain ensures that no single entity has control over the entire network, promoting trust, immutability, and tamper-resistance of the data. In this work, blockchain is used as a design pattern that brings trust between Issuers, Identity Owner’s, and Verifiers. Our solution is designed to be blockchain agnostic. The sole prerequisite is the support of Turing-complete smart contracts.

The trust triangle (Figure 2) is a foundational concept in the field of SSI that outlines the three essential entities involved in establishing trust within a decentralized identity system: Issuers, Identity Owners, and Verifiers. The trust triangle emphasizes the balanced relationship between these entities, ensuring individuals have control over their personal information, Issuers can reliably issue trusted credentials, and Verifiers can confidently verify those credentials.

To address the challenge of building trust between the Verifier and the Issuer, we introduce a Governance Framework that establishes the policies and procedures governing the behavior and responsibilities of the various entities involved (Figure 3). By having a universally trusted Governance Framework, all participants can rely on its authority and oversight to ensure the integrity and trustworthiness of the decentralized identity system.



Issuer	• Verifies the identity of individuals before issuing eIDs • Handles the management of user identities
Identity Owner	• Physical person • Needs to securely and conveniently identify herself in digital transactions
Mobile App	• Stores the identity owner's private key • Manages eID documents
Verifier's Portal	• Relies on digital identity for onboarding and authentication of customers • Examples: online shop, insurance company, city administration office
Verifier's OIDC Server	• Identity provider hosted by the Verifier • Checks verifiable presentations and issues JWT tokens
Governance Framework	• Establishes the legal and regulatory framework that governs eIDs • Business, legal, and technical rules
Blockchain	• Public distributed ledger • Brings trust
Blockchain Node	• Validates and propagates transactions • Maintains a copy of the entire ledger

Figure 1: Actors of our e-ID solution.

Let's explain how an Issuer earns the trust of the Governance Framework. The prospective Issuer initiates the process by reaching out to the Governance Framework and demonstrating its trustworthiness through a designated administrative procedure, which falls outside the scope of this article. Following this verification, the Governance Framework proceeds to deploy a smart contract on the blockchain as a confirmation of the established trust (Figure 4). The Governance Framework is the owner of the smart contract and has the ability to revoke the Issuer at any time. Trust is a transitive relationship. As the Issuer successfully establishes trust with the Governance Framework, this confidence is then transferred to Verifiers, allowing them to trust the Issuer as a reliable and authentic source of identity information.

A Verifier reads data stored in a smart contract on the blockchain for each authentication. Some blockchains, such as Partisia [Par21], allow you to fetch the state of the smart contract. A malicious actor observing the Verifier obtains no information specific to the Identity Owner being authenticated. Other blockchains, such as Ethereum [But13], use getters to access a specific field of the smart contract. In this case, the malicious actor collects information about the Identity Owner. To avoid this scenario, we assume that Verifiers run their own blockchain nodes and access only their local copy of the data.

3 Verifiable Credentials

A verifiable credential (VC) is a tamper-evident and cryptographically secure digital representation of a piece of information or attribute about an individual (e.g. driving license, identification document, educational degree, etc.). It is generated and digitally signed by a trusted Issuer, and can be stored and managed by the Identity Owners themselves. Verifiable credentials are designed to be easily shared with and verified by relying parties, allowing for trusted and privacy-preserving interactions. They enable individuals to selectively disclose relevant information while maintaining control over their personal data, thus fostering trust and facilitating secure digital interactions within SSI frameworks.

Selective disclosure refers to the ability of individuals to control which specific pieces of personal information or attributes they share with different relying parties during digital interactions. This allows for a more privacy-preserving approach, empowering individuals to share only the necessary information required

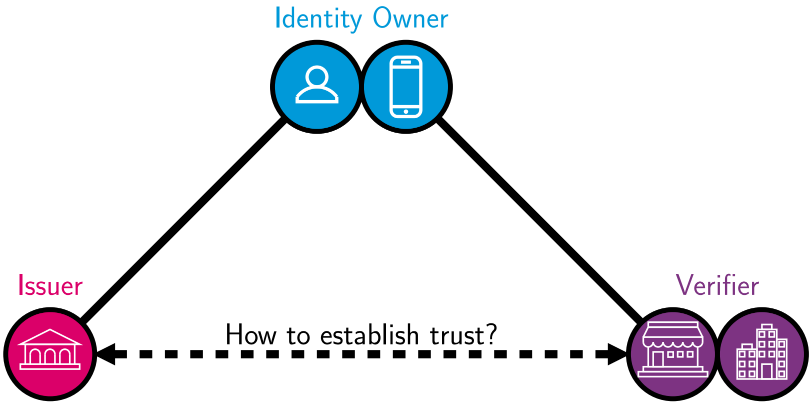


Figure 2: Trust triangle.

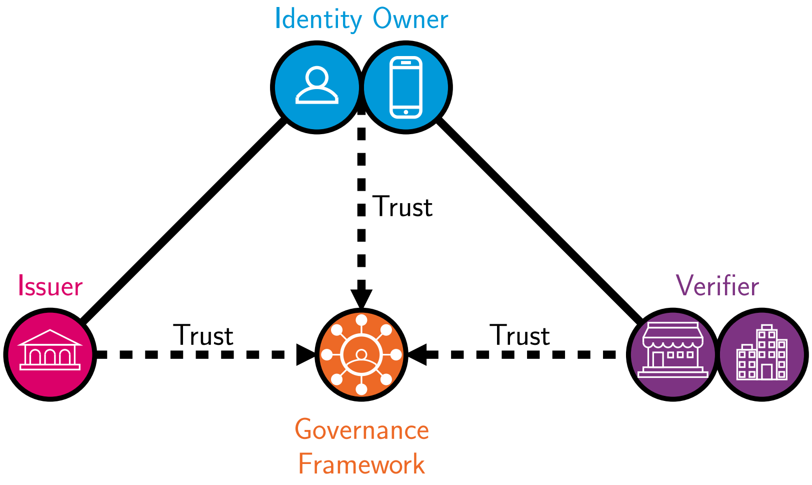


Figure 3: Trust triangle with the Governance Framework.

for a particular transaction. Assume that an individual wants to prove that they are over 18 years old to access a website. In a traditional approach, the individual would typically be required to share their complete date of birth. This method raises privacy concerns because the website now has access to sensitive personal information beyond what is necessary for age verification. Furthermore, the nFADP and the GDPR state that a company must only process and store the data that is essential for its business. With selective disclosure, the website's owner can confirm that the individual is indeed over 18 without needing access to their birthdate. Selective disclosure can be implemented through various cryptographic techniques:

- Zero knowledge proofs (ZKPs) enable an individual to prove the validity of a statement without revealing any additional information beyond the statement's truth. Although ZKPs are a powerful tool, we believe that they may not be well-suited for integration into SSI systems, except for specific attributes like date of birth. Currently, we have not identified any other attributes within the electronic identity context for which ZKPs prove to be useful.

- BBS+ (see for instance [TZ23]) is a signature scheme that supports selective disclosure. It allows the holder of a verifiable credential to generate a valid signature for specific attributes while keeping the rest private. Besides the date of birth, the Issuer also incorporates a Boolean attribute to validate that the individual is above 18 years old. Subsequently, the Identity Owner has the option to present either their full date of birth or only the Boolean value as needed.

- The main goal of the SD-JWT [FYC23] specification is to provide selective disclosure in the simplest way possible and with security-by-design principles in mind. SD-JWT relies on standard algorithms such as JSON Web Signatures and cryptographic hash functions.

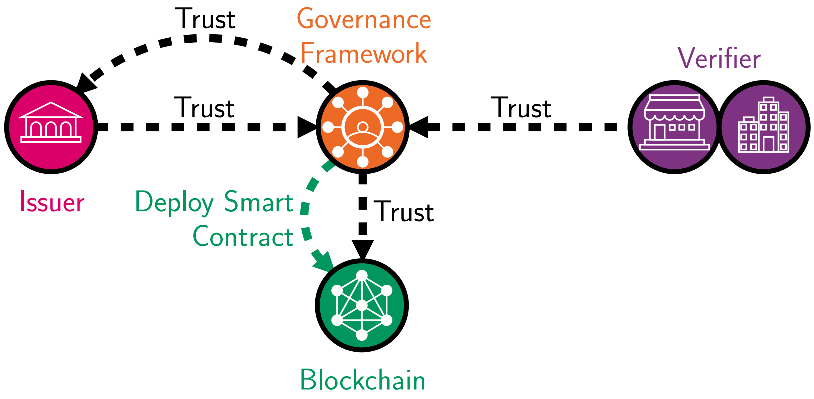


Figure 4: The role of blockchain in establishing trust between the Verifier and the Issuer.Trust flows between entities, forming a transitive relationship where trust in one entity can extend to another.

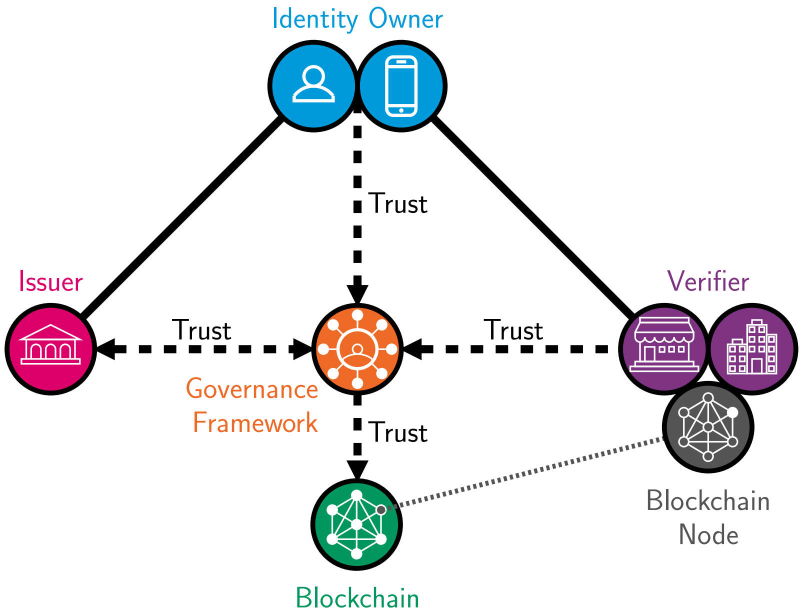


Figure 5: Trust triangle. The Verifier runs a blockchain node to guarantee privacy.

Verifiers have the ability to verify the integrity of the data contained within the Verifiable Credentials and, if desired, implement holder bindings. A holder binding involves requesting the Identity Owner to demonstrate their identity as the intended holder of the Verifiable Credential. This validation can be achieved by proving possession of the private key linked to the public key mentioned in the VC. Let's consider the example of an educational degree to show how the Issuer builds a verifiable credential. Our data are:

```
{
    "name": "John",
    "surname": "Doe",
    "birthdate": "04.05.2000",
    "school": "HES-SO Valais-Wallis",
    "diploma": "Bachelor's degree in Business Information Technology"
}
```

For every claim, a unique random salt is generated. A disclosure encompasses the salt, claim name, and claim value. The issuer then proceeds to encode each disclosure in base64url format and compute the SHA-256 digest of the resulting payload (Table 1). The digests are then arranged in a random order:

```
{
    "_sd": [
        "g54UdI2zQu9g0PXEQj9JfRb9fQGYxBoVnySp6uApsqY",
        "q9gYb3NWsRhQBl12jxCAEhe3XHpUNzVJFegZgr7PwvY",
        "qF9QSMWY3WbtVdRplKx-EviTZYmRnY_zmIN58gIP1sI",
        "wAuMX_zxH-I43u1dobqxaCE6dSJDlQliVQM1OzolKbA",
        "xTujqkR2uEuHezi4QMh0gQV7fsKu9WJKixCcviqftaU"
    ]
}
```

In the following, we refer to this JSON payload as SD-JSON. By applying a digital signature to the SD-JSON, the Issuer obtains a SD-JWT (Figure 6). The resulting SD-JWT, along with the accompanying disclosures, constitutes the VC. The Issuer can include fake disclosures, also known as decoys, to prevent the Verifier from determining the exact number of claims in a VC.

Suppose the Identity Owner aims to demonstrate possession of this educational degree from the school without disclosing their date of birth. The Identity Owner presents to the verifier the SD-JWT along with the four relevant disclosures concerning their first name, name, school, and diploma.

Although the SD-JWT specification is recent, the technologies that make up its various blocks are well understood and widely used. It's also worth noticing that there are plans to integrate SD-JWT technology into well-known SSI frameworks such as Hyperledger Aries.[1] Furthermore, SD-JWT might become a building block in the latest EUDI Wallet Architecture and Reference Framework [Eur23].

For this project, we have opted to utilize the SD-JSON format due to its simplicity. SD-JSON digests and their validity periods are recorded on the blockchain by the Issuer. Note that every blockchain transaction carries the signature of the issuer (Figure 7). By referring to the blockchain, a Verifier can validate the authenticity of a VC. As a result, there is no longer a need to generate a signed SD-JWT from the SD-JSON payload. In order to ensure consistency, each SD-JSON document must contain the following claims:

- A random unique identifier.
- Issuer information
    - Blockchain selection. The Governance Framework determines the blockchain on which the smart contracts will be deployed. Our solution remains blockchain agnostic, allowing the use of multiple

[1] https://github.com/hyperledger/aries-framework-go/releases/tag/v0.2.0

Table 1: SD-JWT — Disclosures and their digests.

Disclosure [salt, claim name, claim value]	Digest [base64url-encoded]
["mH3BjtD619KTdrl4l03osQ","school","HES-SO Valais-Wallis"]	g54UdI2zQu9g0PXEQj9JfRb9fQGYxBoVnySp6uApsqY
["BNA5h8VE3E8zbEKDAmA4Tw","birthdate", "04.05.2000"]	q9gYb3NWsRhQBll2jxCAEhe3XHpUNzVJFegZgr7PwvY
["myGen7hW3DTTouTgJzBsRA","diploma", "Bachelor"\u0027s degree in Business Information Technology"]	qF9QSMWY3WbtVdRplKx-EviTZYmRnY_zmIN58gIP1sI
["Dj5VyIPvDsfsMKWy4g6s0Q","name","John"]	wAuMX_zxH-I43u1dobqxaCE6dSJDlQliVQM1OzolKbA
["r-za0yzSFlyhsuCufB-7eQ","surname","Doe"]	xTujqkR2uEuHezi4QMh0gQV7fsKu9WJKixCcviqftaU

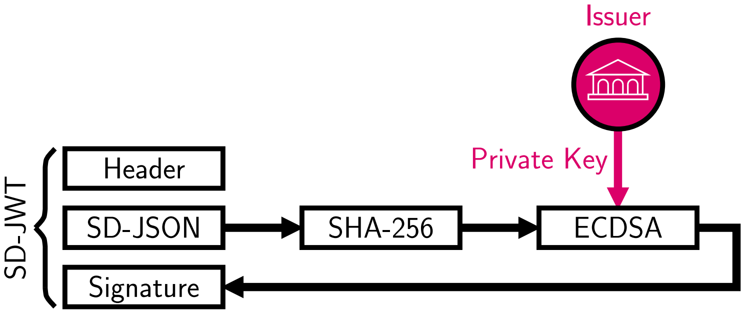


Figure 6: Signature of a SD-JWT token.

blockchains for issuing VCs. Additionally, our solution is capable of managing multiple Governance Frameworks, provided that all participants trust the wallet addresses associated with the Governance Frameworks.

  - Blockchain address of the Issuer.
  - Blockchain address of the smart contract deployed by the Governance Framework for this issuer.

- Identity Owner's public key.
- Type. Each VC is assigned a specific type to denote its purpose. For instance, if a VC is issued by a government entity, it may have a type of `eid`. Similarly, if a VC represents an educational degree, it may be assigned a type of `degree`. Verifiers have the capability to request a specific type of credential from the Identity Owner to fulfill their verification requirements. For instance, a Verifier may ask for a credential of the type `eid` to verify the Identity Owner's citizenship.
- Issuance date.
- A random decoy, serving as a salt.

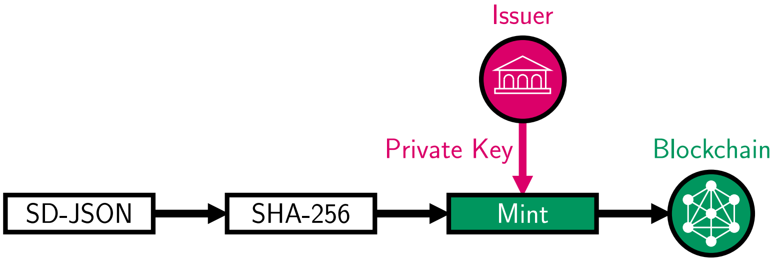


Figure 7: Signature of a blockchain transaction.

Within our smart contract, we maintain a mapping structure that stores the digests of VCs minted by an Issuer (Table 2). This mapping associates each VC digest with its corresponding validity period expressed in Unix time (number of milliseconds that have elapsed since January first, 1970, at 00:00:00 UTC). The entity responsible for deploying the smart contract assumes the role of the owner. When the smart contract is initially deployed, it is associated to the blockchain address of an Issuer.

The Issuer possesses two key capabilities (Table 3). Firstly, they can mint a new VC, which involves adding a new digest and its corresponding timestamp to the mapping structure within the smart contract. Secondly, the Issuer has the authority to revoke a VC by setting its validity period to zero. Once the smart contract has been deployed, the owner's authority is limited to a single action: disabling the contract. This action serves as a mechanism to safeguard the system from a rogue issuer.

Table 2: Smart contract — State variables.

Variable	Type	Description
`Status`	Boolean	Active if `true` (the Governance framework sets this variable to `false` to revoke the Issuer)
`Owner`	Address	Blockchain address of the contract owner (i.e. Governance Framework)
`Issuer`	Address	Blockchain address of the Issuer
`Tokens`	Map	Digest of a verifiable credential and its validity period

Table 3: Smart contract — Functions to modify the state variables.

Function	Role	Description
`Disable`	Contract owner	Revoke the contract (the Issuer loses their trusted status, rendering all the VCs they generated no longer valid)
`Mint`	Issuer	Add the digest of a VC and its validity to the `Tokens` map
`Revoke`	Issuer	Revoke a VC by setting its validity period to 0

4 Verifiable Credential Lifecycle

Upon deployment of the smart contract by the Governance Framework, the Issuer can generate VCs (Figure 8). Once a request from an Identity Owner is received, it undertakes verifications to ensure the identity of the requester. This procedure can be carried out by videoconference, or may require the Identity Owner to be physically present at the Issuer's premises. The definition of this process is defined by each Issuer and is beyond the scope of this article. The Identity Owner then transmits their public key to the issuer[2], which generates the SD-JSON and its disclosures, calculates the SD-JSON digest, and registers it on the blockchain. If the operation is successful, the Issuer returns the new VC to the Identity Owner.

An Identity Owner can now use their VC to authenticate themselves on a web portal (Figure 9). They are redirected to the Verifier's OIDC server, which generates a presentation request in the form of a QR code. The latter contains the list of claims to be provided (e.g. name, educational degree, address, etc.), as well as the URL to which the SD-JSON and necessary disclosures should be sent.[3] The mobile application of the Identity Owner then takes care of finding a set of VCs that collectively encompass all the claims requested by the Issuer. This involves solving an instance of the set-covering problem, which is known to be NP-complete [Cor+09]. A greedy algorithm exists for approximating the set-covering problem in polynomial time. This algorithm follows a specific rule: at each step, select the set that contains the highest number of elements that have not been covered yet. Given the relatively small number of VCs involved, finding the

[2] Similar to a Certification Signing Request (CSR) used in the context of X.509 certificates, the Issuer can initiate a signing process to verify the Identity Owner's control over the private key. This verification serves to establish the authenticity and ownership of the key pair.

[3] One can also envision a scenario where the Verifier specifies a type of VC for each of the claims. For example, the name and date of birth must strictly originate from a VC of type `eid`.

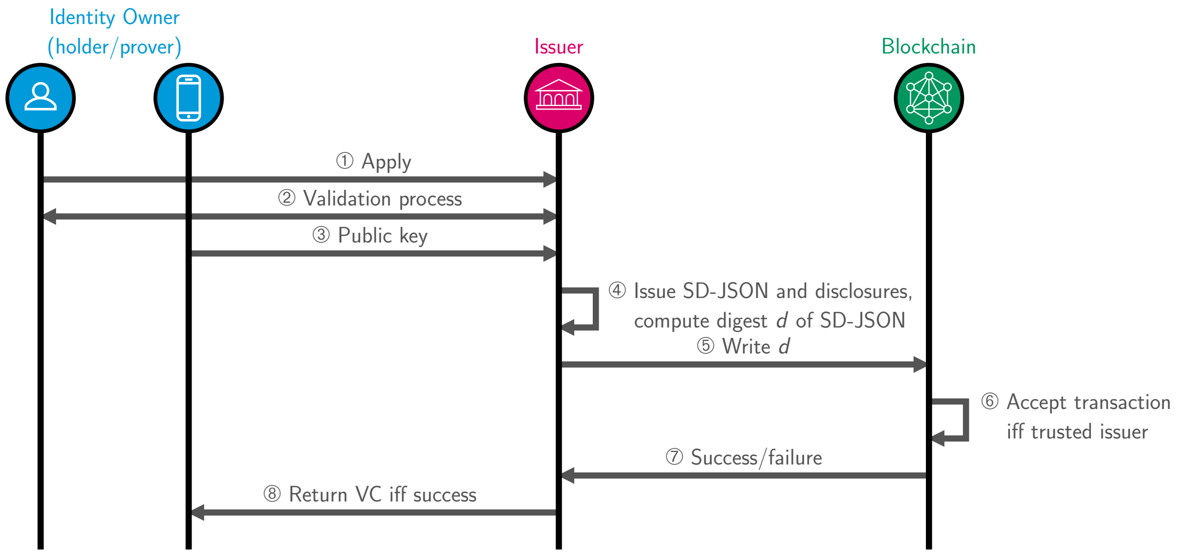

Identity Owner (holder/prover)
Issuer
Blockchain
① Apply
② Validation process
③ Public key
④ Issue SD-JSON and disclosures, compute digest d of SD-JSON
⑤ Write d
⑥ Accept transaction iff trusted issuer
⑦ Success/failure
⑧ Return VC iff success

Figure 8: Issuing a new VC to an Identity Owner.

optimal solution remains a viable choice. In the following, we denote by $\mathcal{S}$ the (optimal) solution to the set-covering problem.

To prove control over the private key(s) associated with the public key(s) recorded in the set of VCs $\mathcal{S}$ and prevent replay attacks, the Identity Owner constructs a binding that includes the following elements:

- Disclosures for the claims required by the Verifier.
- A timestamp.

The Identity Owner signs the binding for each distinct public key in $\mathcal{S}$. Then, the Identity Owner sends to the Verifier a verifiable presentation that consists of $\mathcal{S}$, the binding, and the signature(s). The Verifier proceeds as follows to validate the presentation:

- For each SD-JSON provided by the Identity Owner, the Verifier computes its digest and verifies its status in the smart contract of the Issuer. If a VC is revoked or the Governance Framework no longer trusts the Issuer, the Verifier denies access to the web portal.
- The Verifier now trusts the information contained in $\mathcal{S}$. The Verifier checks a digital signature of the binding for each distinct public key in $\mathcal{S}$.
- The Verifier validates the timestamp and check the disclosures.

If all the above steps are validated, the Verifier generates a JWT token for the Identity Owner, allowing them to access the web portal.

5 Software Implementation

We have developed an initial prototype of our solution for the Partisia blockchain [Par21]. We chose this blockchain for the following reasons:

- Governance is provided by a foundation based in Zug, Switzerland.
- Partisia chose an energy-efficient consensus “consuming only a fraction of a thousandth of the energy required to power legacy blockchains” [Par].
- Partisia relies on sharding to avoid congestion. The blockchain can be dynamically scaled to any number of users.

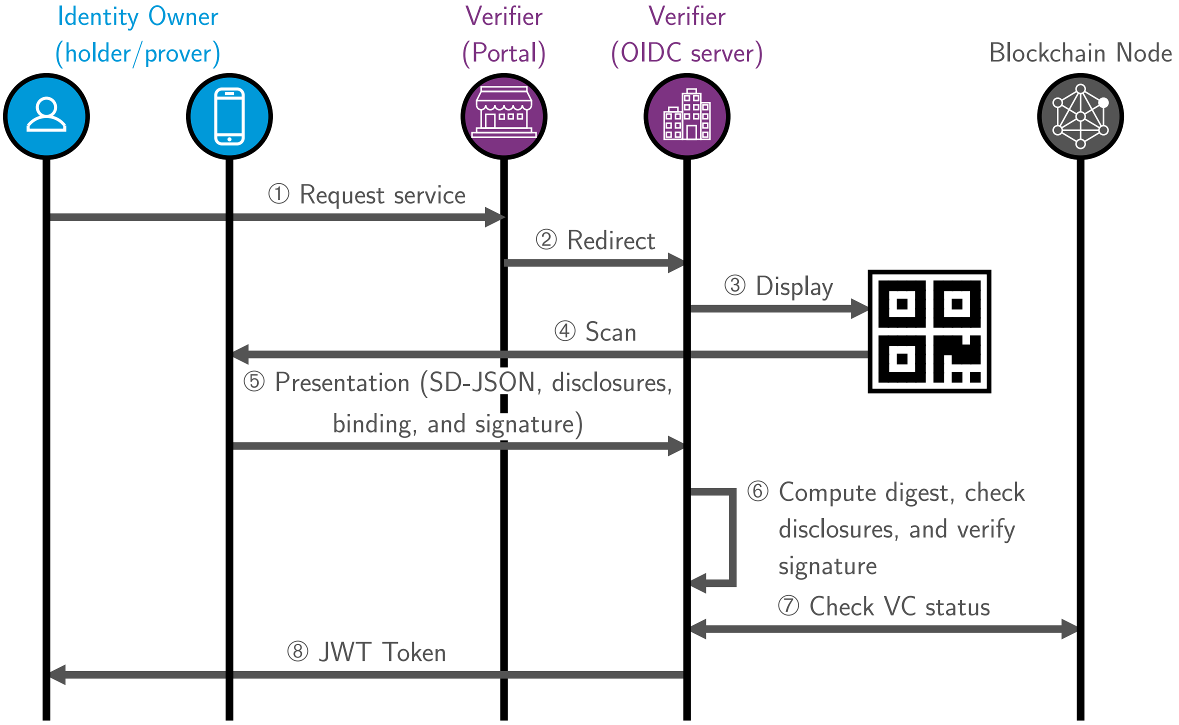


Figure 9: Validating a VC.

Figure 10 describes the architecture of our Verifier. Our main building block is a Spring Authorization Server that is itself build on top of Spring Security. It offers a convenient way to create an OpenID Connect provider. In order to showcase SSI as a second factor of authentication, we employ Keycloak as the primary Identity Provider (IdP) in our system.

1. Identity Owners wishing to access a protected resource are redirected to the Keycloak login page.

2. They to log in with the SSI Identity Provider.

3. They are redirected to the SSI Identity Provider login page that displays a QR Code. The latter contains the claims to be presented (e.g. name, date of birth, educational degree, etc.).

4. They scan the QR code with their mobile application[4] (Figure 11). The QR code stores a base64url-encoded JSON document. In our example, the payload is:

```
{
  "claims":["nationality","date_of_birth"],
  "csrf_parameter_name":"_csrf",
  "csrf_value":"6ujgvYOjGM6...aPmuW9tDZ",
  "endpoint_url":"http://localhost:9000/ssi-login",
  "session_cookie_name":"jsessionid",
  "session_id":"1EEE5C7888F65E3A4726286D82132527"
}
```

   The fields csrf_parameter_name, csrf_value, session_cookie_name, and session_id are specific to Spring Security, and their detailed explanation is beyond the scope of this article.

5. The mobile application builds a verifiable presentation consisting of the SD-JSON and the disclosures requested by the Verifier (field claims). In our example, the Verifier will check the nationality as well as the date of birth of the Identity Owner.

[4] At the moment, we simulate the mobile application with a Python script.

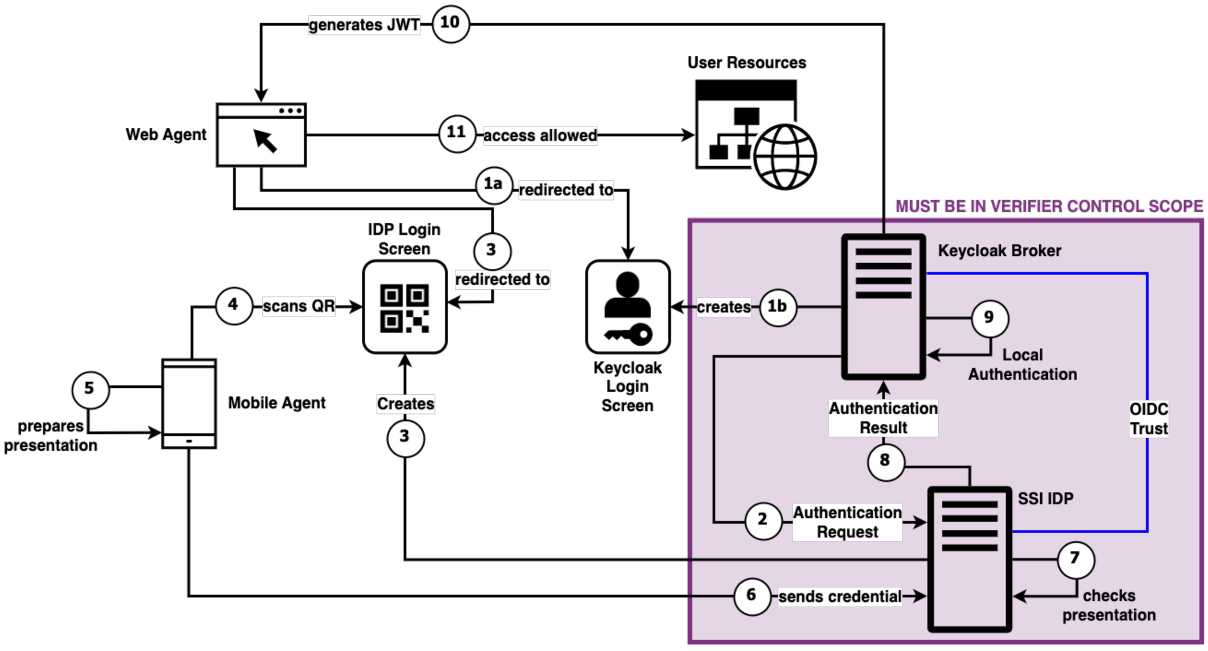


Figure 10: Self-Sovereign Identity PoC Architecture

6. The verifiable presentation is signed withe the Identity Owner's private key and sent to the SSI Identity Provider. The field `endpoint_url` in the QR code specifies the URL to which the request should be sent.
7. Our SSI Identity Provider validates the payload.
8. Our SSI Identity Provider then sends the authentication result to Keycloak.
9. Keycloak authenticates the Identity Owner.
10. Keycloak sends a JWT token to the Identity Owner's web agent.
11. The Identity Owner can now access the protected resource.

6 Preliminary Security Analysis

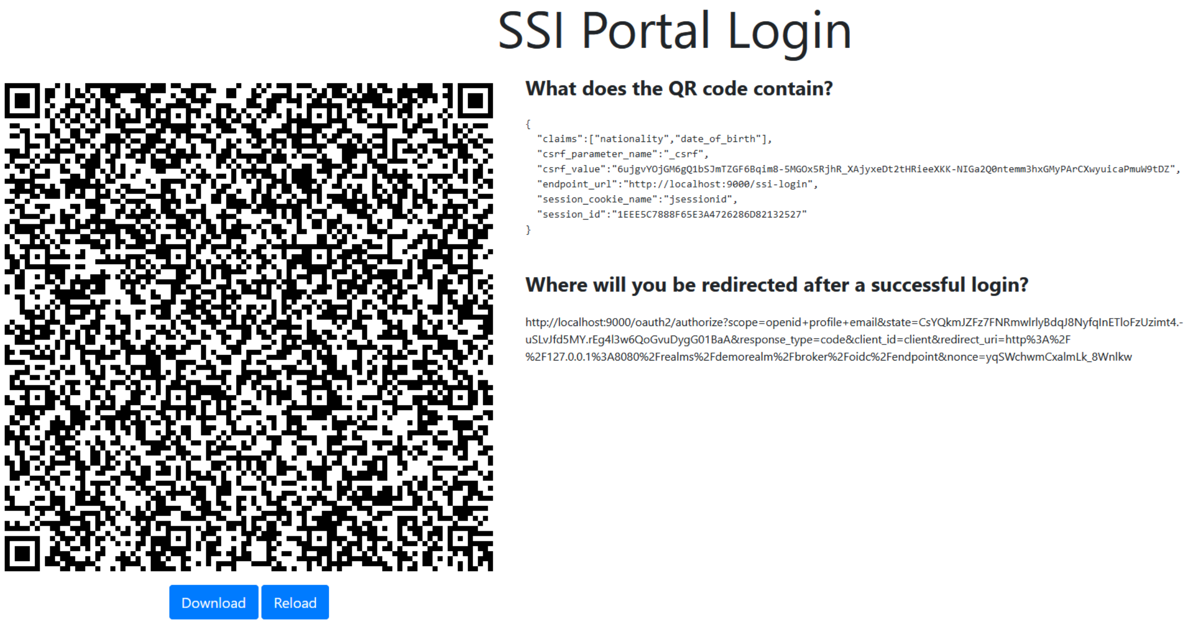
In this article, we present an initial security analysis of our solution. To delve deeper into the security aspects, we plan to develop a real-world use case involving multiple issuers, verifiers, and a large number of VCs.

Let's consider the scenario involving an honest-but-curious adversary whose objective is to discover a SD-JSON document by observing SHA-256 digests stored on the Issuer's smart contract. This implies an attempt to reverse the cryptographic hash function employed in the creation of the digest (first pre-image attack). Since our SD-JSON always incorporates a decoy random element serving as a salt, it is computationally infeasible to build rainbow tables for finding the SD-JSON payload. The time complexity of the attack is $\mathcal{O}(2^{256})$.

The second scenario involves a malicious adversary willing to impersonate an identity owner. The first step is the same as in the previous example: knowing a SHA-256 digest, the attacker must find a SD-JSON document. Then, the attacker must perform a pre-image attack for each hidden claim in the document. Here again, the time complexity is $\mathcal{O}(2^{256})$.

For the third scenario, we consider a malicious Identity Owner willing to add a claim to their VC. Let m and d denote a genuine SD-JSON document and its digest, respectively. The attacker

- Generates a new disclosure that consists of a random salt, a claim name, and a claim value.



SSI Portal Login

What does the QR code contain?

```
{
  "claims":["nationality","date_of_birth"],
  "csrf_parameter_name":"_csrf",
  "csrf_value":"6ujgvYOjGM6gQ1bSJmTZGF6Bqim8-5MGOx5RjhR_XAjyxeDt2tHRieeXKK-NIGa2Q0ntemm3hxGMyPArCXwyuicaPmuW9tDZ",
  "endpoint_url":"http://localhost:9000/ssi-login",
  "session_cookie_name":"jsessionid",
  "session_id":"1EEE5C7888F65E3A4726286D82132527"
}
```

Where will you be redirected after a successful login?

http://localhost:9000/oauth2/authorize?scope=openid+profile+email&state=CsYQkmJZFz7FNRmwlrlyBdqJ8NyfqInETloFzUzimt4.-uSLvJfd5MY.rEg4l3w6QoGvuDygG01BaA&response_type=code&client_id=client&redirect_uri=http%3A%2F%2F127.0.0.1%3A8080%2Frealms%2Fdemorealm%2Fbroker%2Foidc%2Fendpoint&nonce=yqSWchwmCxalmLk_8Wnlkw

Figure 11: QR code.

- Adds the digest of the new disclosure and a nonce to m. If the SHA-256 digest of the modified payload is equal to d, the attack succeeded. Otherwise, the attacker increments the nonce and repats the procedure.

The time complexity of this second pre-image attack is $\mathcal{O}(2^{256})$.

7 Conclusion

We have presented a self-sovereign identity solution that leverages blockchain technology to establish a trusted relationship between issuers and verifiers. To validate the correctness of our system, a proof of concept was conducted using the Partisia blockchain, showcasing the accuracy and reliability of our solution. Moving forward, our next steps encompass the following tasks:

- Wallet application. We will focus on building a user-friendly wallet application that allows individuals to manage and present their verifiable credentials effectively.

- Testing our solution with a large number of VCs. Rigorous testing will be conducted to evaluate the scalability and performance of our solution when handling a substantial volume of VCs. In order to achieve a representative self-sovereign identity solution for Switzerland, our goal is to mint a minimum of 8 million VCs. This will also contribute to a better understanding of the security of our SSI Solution.

- Interoperability. We aim to enhance the interoperability and versatility of our SSI solution by adapting our smart contract to work with different blockchain platforms. We will then conduct a comparative analysis of at least two blockchain networks, considering factors such as transaction fees, responsiveness, scalability, and consensus mechanisms. This evaluation will enable us to make informed decisions about the most suitable blockchain platforms for our SSI solution.

- We will conduct a thorough comparison of our self-sovereign identity solution with other notable alternatives in the market, such as MATTR VII [MAT23], Hyperledger Aries [Fou22], and reclaimID [SBS18].

- In the context of Self-Sovereign Identity (SSI), post-quantum resistance holds paramount importance due to the long-term validity of VCs such as educational degrees. The selection of SD-JWT is a strategic choice here, given its reliance on cryptographic hash functions. However, it's worth noting that blockchain transactions and bindings in our current implementation utilize the ECDSA signature algorithm, which is susceptible to being broken by a quantum computer.

To address the potential impact of quantum computing, we recognize the need to explore quantum-safe alternatives. We will compare various post-quantum signature schemes to sign our bindings (e.g. the algorithms selected by the NIST for post-quantum cryptography standardization [NIS]). Choosing a distributed ledger is somewhat more complicated. IOTA [Mul+22] used to rely on Witernitz one-time signatures [BBD09] and was the best-known quantum-resistant ledger. Unfortunately, IOTA announced the implementation of a more standard algorithm (EdDSA) [IOT23] and can no longer claim to be quantum-resistant. In the future, the Quantum Resistant Ledger (QRL) [MWL16] could potentially be considered as an option for our solution. However, it is important to note that, at the time of writing, smart contracts are not supported.

References

[All16] C. Allen. "The Path to Self-Sovereign Identity". In: *Life with Alacrity* (Apr. 2016). URL: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[And21] G. Andrey. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213124.

[BBD09] D. J. Bernsetin, J. Buchmann, and E. Dahmen, eds. *Post-Quantum Cryptography*. Springer, 2009.

[But13] V. Buterin. *Ethereum Whitepaper*. 2013. URL: https://ethereum.org/en/whitepaper.

[Cor+09] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2009.

[Dép21] Département fédéral de justice et police. *Document de travail concernant le projet d'identité électronique (e-ID)*. Aug. 2021.

[Eur23] European Commission. *The Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework*. Version 1.0.0. Jan. 2023. URL: https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework.

[Fou22] H. Foundation. *Hyperledger Aries*. 2022. URL: https://www.hyperledger.org/use/aries.

[FYC23] D. Fett, K. Yasuda, and B. Campbell. *Selective Disclosure for JWTs (SD-JWT)*. IETF Internet-Draft. June 2023. URL: https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/.

[Gro21] Groupe libéral-radical. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213129.

[Grü21] F. Grüter. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213125.

[IOT23] IOTA Wiki. *Path to Chrysalis*. June 2023. URL: https://wiki.iota.org/introduction/explanations/update/path_to_chrysalis.

[Mäd21] J. Mäder. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213127.

[Mar21] M. L. Marti. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213126.

[MAT23] MATTR. *MATTR VII — Overview*. 2023. URL: https://learn.mattr.global/docs/vii-platform/overview.

[Mul+22] S. Muller, A. Penzkofer, N. Polyanskii, J. Theis, W. Sanders, and H. Moog. "Tangle 2.0 Leaderless Nakamoto Consensus on the Heaviest DAG". In: *IEEE Access* 10 (2022), pp. 105807–105842. DOI: 10.1109/access.2022.3211422.

[MWL16] J. Matier, P. Waterland, and J. Lomas. *Quantum Resistant Ledger (QRL)*. Oct. 2016. URL: https://github.com/theQRL/Whitepaper/blob/master/QRL_whitepaper.pdf.

[Nak09] S. Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2009. URL: http://www.bitcoin.org/bitcoin.pdf.

[NIS] NIST. *Post-Quantum Cryptography Standardization*. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

[Par] Partisia Blockchain. *Solving the Blockchain Trilemma*. URL: https://partisiablockchain.com/.

[Par21] Partisia Blockchain. *Partisia Blockchain — A WEB 3.0 public blockchain built with MPC for trust, transparency, privacy and speed of light finalization*. Version 1.06. Mar. 2021. URL: https://partisia.com/wp-content/uploads/2021/03/Partisia_WhitePaper_1_06.pdf.

[SBS18] M. Schanzenbach, G. Bramm, and J. Schütte. *reclaimID: Secure, Self-Sovereign Identities using Name Systems and Attribute-Based Encryption*. abs/1805.06253. 2018. URL: https://arxiv.org/abs/1805.06253.

[Sta21] S. Stadler. *À l'État de mettre en place une identification électronique fiable*. Mar. 2021. URL: https://www.parlament.ch/fr/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20213128.

[TZ23] S. Tessaro and C. Zhu. *Revisiting BBS Signatures*. Cryptology ePrint Archive, Paper 2023/275. 2023. URL: https://eprint.iacr.org/2023/275.