SERAPH ID

A Self-Sovereign Identity solution designed on NEO

July 8, 2019

Contents

1 Introduction **2**
- 1.1 Identity history . . . 2
- 1.2 Introduction to privacy & shifting behaviour . . . 2
- 1.3 Challenges . . . 4

2 Self-Sovereign Identity in brief **4**
- 2.1 Standards . . . 5

3 Seraph ID **5**
- 3.1 Why NEO? . . . 5
- 3.2 Seraph ID actors . . . 6

4 SSI network and ROT-Managers **7**
- 4.1 Goals: A few examples of what is possible to do with Seraph ID 9
- 4.2 Seraph ID Example Use Case: Smart Shared Accommodation . . 9

5 Seraph ID characteristics **11**
- 5.1 Identity uniqueness . . . 11
- 5.2 Authentication . . . 11
- 5.3 Encryption . . . 11
- 5.4 Off-chain vs On-chain project . . . 12
- 5.5 Zero-Knowledge proof . . . 12
- 5.6 Credential Schemas . . . 12
- 5.7 Seraph ID components overview . . . 12

6 Interactions **14**
- 6.1 Roles . . . 14
- 6.2 Issuing a claim . . . 14
- 6.3 Verifying a claim . . . 15
- 6.4 Verifying a claim with a ROT-Manager . . . 17

7 Seraph ID technical details **17**

8 Roadmap **18**

9 Terminology **18**

Abstract

In the recent past, peoples' sensibility regarding the privacy of their data has increased worldwide. Especially when it comes to digital information, our data is spread and exchanged around the globe without explicit control from the direct data owner, and besides the lack of privacy, other consequences such as safety and security issues could arise.

Given these circumstances and considering the current need for a user-centric approach, Self-Sovereign Identity (SSI) represents a potential solution that will allow users to administer their data and identify themselves online, just as it is typically done in the real world. Public blockchain technologies, decentralized by design, can empower the realization of a truly Self-Sovereign Identity solution. Currently, the lack of standards is leading to difficulties in interoperability, which represents one of the biggest challenges that is slowing down the adoption of blockchain technologies. Luckily, in the Self-Sovereign Identity space, there are several World Wide Web Consortium (W3C) working groups putting effort into creating standards. These groups standardize what digital credentials should look like and how third parties can verify their authenticity.

A few implementations using the Self-Sovereign Identity principle are already available, and almost all of them are following standards defined by W3C working groups. These implementations showcase a first step towards the interoperability of different identity solutions. In fact, at a later stage, it does not matter on what system the credentials are issued as the various solutions are following the same standards and therefore also support the same validation procedures.

NEO, as an open source project driven by its community, aims to provide a set of toolboxes that can be used to create distributed applications for a 'Smart Economy." NeoID is a decentralized identity solution that seeks to empower users and organizations to have better control of their identities and deliver a higher degree of trust and security in the smart economy. To increase the adoption of Self-Sovereign Identities, users not only need proper tools to manage their credentials, but it is also of utmost importance that the SSI functionalities are used and more widely implemented in different dApps. That is the vision behind Seraph ID, which is positioned as a critical component of the NeoID project, leveraging part of its functionalities.

For that reason, Seraph ID is providing tools to empower Self-Sovereign Identity usage, not just to the end-users, but also to entities that are building applications on top of the NEO ecosystem.

1 Introduction

1.1 Identity history

The first legal identity document was passed into law by King Henry of England in 1414. It was a very early version of what we know nowadays as a passport. For the next 500 years, up until the first World War, most people did not have or need an ID. A photographic identity document was required for the first time in 1915 by the British government.

The identity document was created, and it would soon prove crucial as a simple authentication tool during the demographic explosion.

Simultaneously with the digitalization of industries, identity documents have also followed the same path, becoming electronic and digital. Every new emerging technology has influenced the state of identification and identification tools. As an example, we would like to mention documents with biometric data such as fingerprints or retina measurements.

In the past decade, people have experienced different types of identity. The most recent, private-by-design solution to date is Self-Sovereign Identity, based on blockchain technology.

1.2 Introduction to privacy & shifting behaviour

Around 40% of the world population[1] has an internet connection today and this number is growing every year. With an increasing number of users comes the simultaneously increase in the amount of data uploaded and shared online. Individuals' identity is by far the most sensitive data category and is often stored in data centers around the world, which can be subject to cyber-attacks.

According to the Breach Level Index[2], 269,435 data records are lost or stolen online every hour. To put things into perspective, that means 4,491 data records are lost or stolen every minute and 75 every second. These figures are very alarming, and they will keep rising with the increase of internet dependence. According to a Juniper research[3] forecast, the global cost of cybercrimes in 2019 will exceed US $2 trillion, and in 2020, the average cost of a data breach will exceed US $150 million.

With the increasing dependence on the internet and its features, users are forced to store and share more and more personal data online. Most users dislike this fact but do not have any choice, since access to services is otherwise

[1] https://www.internetlivestats.com/internet-users/
[2] https://www.breachlevelindex.com
[3] https://www.juniperresearch.com/home

restricted. Most users are also concerned about their online data being stolen or sold, or experiencing identity theft.

According to Statista[4] from August 2017, 73 percent of the respondents stated they would feel most concerned about hackers gaining access to their personal banking information. In second place came credit card numbers, which can be used maliciously. Third and fourth-ranked were social security numbers and then residency addresses. All this data is personal data that is stored on the web by multiple, diverse agents with very different security and privacy policies.

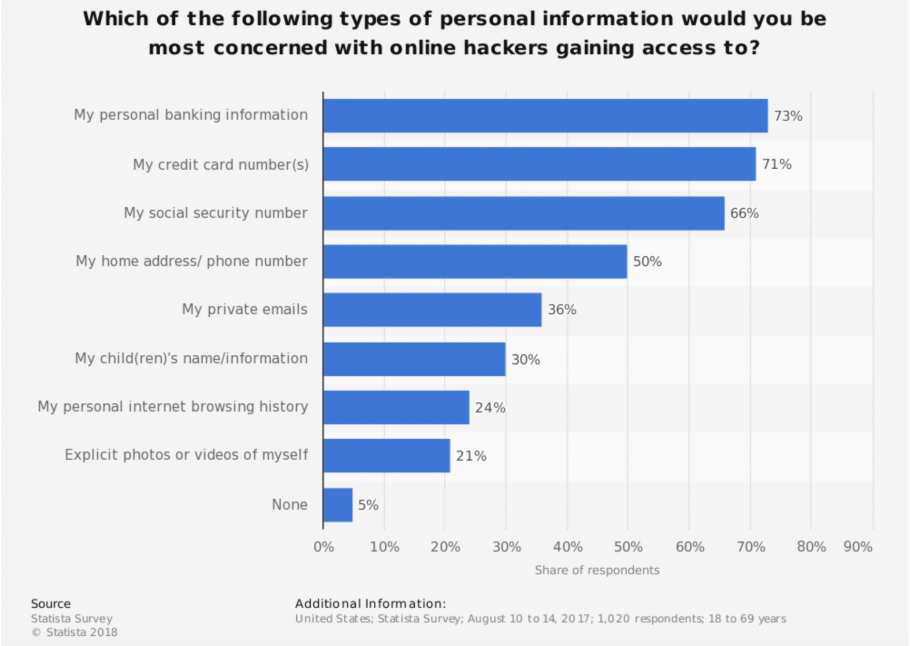


Figure 1: Statista survey on personal information concerns

Banking information, home addresses, personal pictures, private emails, plus much more. All this information makes up what we call "identity." Any of this data being stolen could then potentially negatively affect the integrity of a user's identity in their daily life. It can have lasting effects and dire consequences, for example, if a crime is committed using their stolen identity or they are harmed themselves. According to an IdentityForce report from 2017, there were 16.7 million victims of identity theft worldwide. One million of them, which equals almost 6%, are children.

User behavior is now shifting to more secured and decentralized (personally

[4] `https://www.statista.com`

owned) methods of storing both personal and private data.

1.3 Challenges

Since nearly all companies nowadays are managing data, there is one key questions to answer: how can identity be managed on the internet safely?

It becomes crucial for companies to find an alternative to traditional login or digital identity. Self-Sovereign Identity can be a sustainable solution for governments and companies to redefine the way people are identified on the internet and in their daily lives.

2 Self-Sovereign Identity in brief

In the physical world, we carry around a set of credentials in our pockets every day: driver licenses, credit cards, gym membership cards, national ID, etc. When we need to prove certain facts about ourselves to a third party (e.g., age) or some claim (e.g., "I am a citizen of that country") it is enough to show a physical credential. Usually, there is a manual check that is done by simply looking at the provided credentials. This way, the authenticity can be validated, and the reported information can be trusted.

When we show our passport to authorities at the entry or exit of an airport, the immigration officer is trusting the document and the information stated within it. This is possible because the passport itself is recognized internationally, as it has been standardized and there is trust between the verifier (immigration officer) and the issuer (governments capable of issuing a valid passport).

In the digital world, such capability is not openly available to each entity who would spontaneously need to verify some identity or claim. Federated identity solutions could allow a specific verifier to validate claims issued by a specific issuer, but that would require an integration project between the two parties. Those integration projects may be use-case specific (B2B) or based on standards such as OAuth2.0 or OpenID Connect (e.g., social logins). This federated model is centralizing the storage of identities and claims on external identity providers, increasing privacy risks, and creating a single point of failure.

Self-Sovereign Identity aims to solve this issue by avoiding the centralization of identities and allowing the owner to store and manage their identity. The end user is then equipped to decide if, when, and what data to share with a third party that is requesting access to specific information about the user. This third party (verifier), once in possession of the piece of digital information shared by the user, should then be capable of validating it. The validation process involves verifying that the claim is still valid (it has not been revoked), that it has not been tampered with (it is authentic), and who the issuer was. This can be

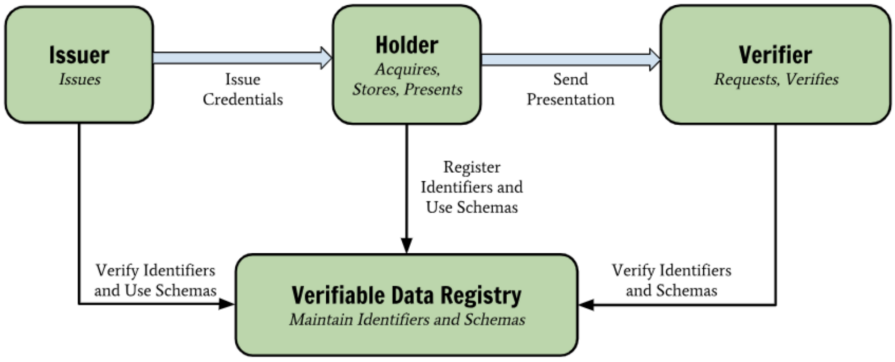


Figure 2: The roles and information flows forming the basis for this specification.[6]

compared to a passport; it is issued to the owner by a valid institution, who is then able to share it on demand with an immigration officer, who is, in turn, able to verify it. By using blockchain technology, digital credentials can go through the same process online.

2.1 Standards

The first step necessary to bring the advantages of the physical world into the digital space, and to allow third parties to easily validate many different types of credentials (even without previous preparation), is the adoption of shared standards. As mentioned, working groups of W3C are drafting the first two primary standards. One standard defines how we can represent and generate a digital credential,[7] and another that defines how digital credentials can be validated.[8]

3 Seraph ID

3.1 Why NEO?

The vision of NEO is to create a Smart Economy, which consists of three elements: digital assets, digital identity, and smart contracts. We believe in SSI as an important element of NEO's decentralized identity protocol, NeoID. Self-Sovereign Identity can help to empower the NEO network and provide further features in the NEO ecosystem to facilitate the achievement of a Smart Economy. Seraph ID can become a fundamental part of NEO's vision, and acts as an enabler for future projects.

[6] https://www.w3.org/TR/verifiable-claims-data-model/
[7] https://w3c-ccg.github.io/did-spec/
[8] https://www.w3.org/TR/verifiable-claims-data-model/

Swisscom Blockchain AG believes in Self-Sovereign Identity as an empowering tool for individuals. It is a way for people to regain control over their data and privacy, by protecting themselves from being hacked, having their data stolen, or having their identity lost. We think Self-Sovereign Identity should not be only available to a specific group or user base, but instead, it should be widely available to the public. Currently, there are a lot of different entities building Self-Sovereign Identity capabilities on top of existing blockchain protocols.

NEO is an established and recognized partner in the blockchain space, and fulfills all requirements that are needed to build a Self-Sovereign Identity solution (e.g., smart contracts, public blockchain). Since we are already working closely in collaboration with the NEO team, we have decided to use NEO as our underlying platform.

In the first version of the Seraph ID, we will explore the current state of Self-Sovereign Identity and how we can leverage the NEO platform's capabilities and functionalities to design and implement a working product. As an integral part of NeoID, we will align with the technical features and design functionalities of NeoID through close collaboration with the team from NEO Global Development. However, this project is meant to be open source, so we encourage and value community contributors to join and help advance the project once the initial stages of development are more clearly defined.

3.2 Seraph ID actors

Within the Seraph ID model there are four roles available:

Issuer A role an entity performs by asserting claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder.

Verifier A role an entity performs by receiving one or more verifiable presentations for processing. Other specifications might refer to this concept as a *relying party*.

Holder A role an entity performs by possessing one or more verifiable credentials. A holder is usually, but not always, the subject of the verifiable credentials they are holding. Holders store their credentials in credential repositories.

ROT-Manager (Root-of-trust) A role an entity performs by governing an SSI-network. A ROT-manager defines a set of business, legal, and technical rules that any participant should adhere to in order to be onboarded and participate in the specific SSI-network.

4 SSI network and ROT-Managers

Although Seraph ID aims to introduce generic SSI capabilities on the NEO blockchain, it should also allow other entities to spin-up their own specific business case. This need for individual use cases is the reason we have decided to introduce the SSI-network concept in Seraph ID. Technically speaking, anyone could perform the role of an issuer (as we will see later, it is enough to deploy an issuer-type smart contract). However, this means that it is necessary to trust a specific issuer and therefore, the claims he generates for various subjects. This trust is essential during the validation process. It is not just about digital verification, but about a real-world agreement and trust between entities.

This process should happen through a set of business and legal rules between all entities involved. Why do we trust passports? Because we have trust in the entity that has issued the passport itself (national government). So now let us assume that, before starting to deal with smart contracts and credentials, a couple of entities sit together around a table and agree to trust each other, while embracing a specific role within the SSI-network that they will hold. They could even decide to sign a business document, that contains the terms of their agreement. Once this step is done, a claim issued by a trusted issuer can then be considered meaningful, as behind the digital validation process, there is an agreement in place. Once a set of entities agree on their role, they have formed an SSI-network: a specific group of entities that are trusting each other to issue and validate claims. How an offline agreement can be reached is not within the scope of this document, but should be decided by the involved parties based on their particular business requirements.

Even though everyone is free to spin-up their own distributed "SSI-network" and decide the rules that are governing it, it makes sense to grow and leverage the network of trust by implementing ROT-managers. They can onboard other parties, and since the network trusts the ROT-manager, it also trusts the parties it is onboarding.

Once an entity takes on the role of ROT-manager, it represents a repository of whitelisted entities that can be indirectly trusted by verifiers. A ROT-manager has the goal to onboard other entities offline and through the business agreements that govern the SSI-network. As a verifier, once you trust a ROT-Manager, you then can benefit immediately from the subsequent trust of all the issuers that the ROT-Manager whitelisted within the SSI-network. Having a ROT-Manager in an SSI-network is optional and can be decided based on the specific business case.

To have a clearer understanding of the possible trust distribution model provided by Seraph ID, we want to give an example. Let us imagine a LinkedIn-like application, where every time a user wants to add previous job experience or education to their profile, they need to share valid proof to confirm the authen-

ticity of the information. The issuer itself can decide the form that this proof takes, but what matters is that LinkedIn trusts the third party. In this example, we assume that every company is capable of issuing a credential with a meaning such as "valid employee or ex-employee' to their employees, while every university is capable of issuing a "graduation certificate" to their students.

In the picture below, LinkedIn is trusting Microsoft and Apple. So, once a user presents a claim from those companies, LinkedIn will be able to validate it and will consider it as proof, whereas this will not be possible for credentials issued by Samsung (as in the example, LinkedIn does not have any trust relationship with it). For university credentials, the trust distribution model is different; LinkedIn is relying on an entity that acts as a ROT-Manager: the university consortium. Each university, enrolled in the consortium, is trusted by the consortium and therefore, indirectly, LinkedIn will trust all the participants.

Trust distribution model with Seraph ID:

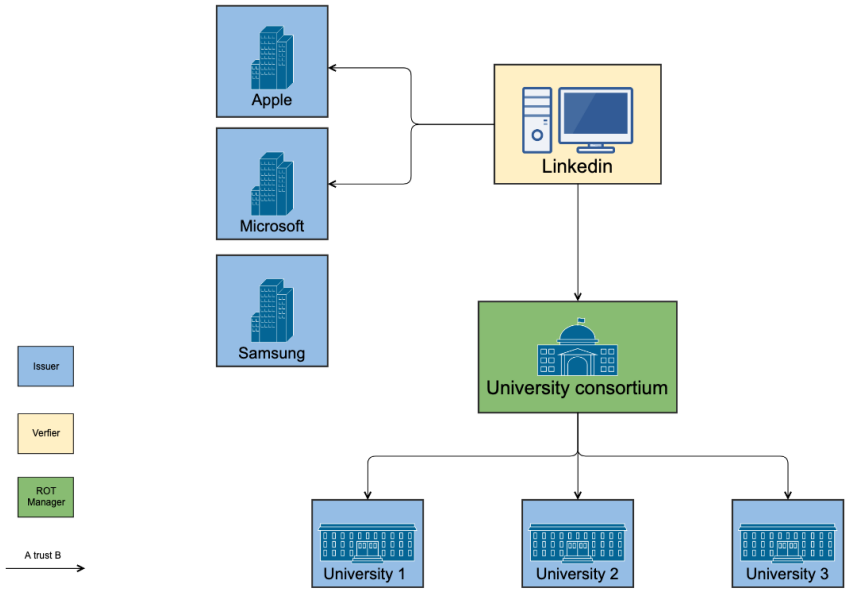


Figure 3: "LinkedIn-like" use case

As described, all dApp developers or consortia can then decide on their trust model and eventually leverage one or more ROT-Managers. This ongoing distribution of trust is not something that an identity owner would need to consider, as they will only hold or share claims.

4.1 Goals: A few examples of what is possible to do with Seraph ID

The main goal of Seraph ID is to provide a set of tools for the NEO development community that would allow all dApps to leverage and implement SSI capabilities easily. End users will have the ability to interact within the NEO ecosystem without a central authority. They will instead directly manage their identity and stay in control of the data they want to share with third parties.

Let us have a look at a few simple examples:

For issuers, it adds to your dApps:

- the ability to issue a claim to users that have completed an onboarding process
- the ability to issue a claim to your users that have completed a training module

For verifiers, it adds to your dApps:

- the ability to verify if a user is older than 18
- the ability to verify if a user has subscribed to a partner-service

For identity holders, it allows you to:

- store your data and claims issued to you
- decide what data to share with whom

4.2 Seraph ID Example Use Case: Smart Shared Accommodation

Every day, thousands of tourists search for accommodations to stay with at their respective destinations. Typically, online accommodation agencies such as Airbnb look for hosts and landlords that can offer accommodation to customers. The online agency then acts as an intermediary between the landlord and the customer, through a verification of the tourist's identity and the issuance of an access key to the tourist to access the accommodation. This is achieved by means of forwarding the contact details to the landlord or providing a unique code that can be used to gain entrance to the accommodation.

Oliver is one of these tourists, and he is looking for accommodation in Barcelona, Spain. He has discovered a new smart shared accommodation dApp built on top of the NEO blockchain. Oliver decides to use it and starts searching through the dApp for his desired accommodation in Barcelona. The dApp offers a list of all available accommodation in Barcelona, allowing the users to filter by dates and number of people. Oliver then selects the most suitable option

and applies for it. To make the reservation, the dApp requires Oliver to identify himself as an individual.

Fortunately for Oliver, his government recently decided to implement a digital version of the national ID with the help of the NEO blockchain. Some months ago, Oliver applied for a digital ID by presenting his physical ID to a local authority. In the process, the government authority generated a digital credential, containing Oliver's personal information, and a hash of that credential ID was uploaded to the NEO blockchain.

Now Oliver can identify himself to the dApp with his digital identity. To do so, the dApp has integrated a tool that interacts with the wallet of the user, where the digital credential is stored. Through the application, the user is requested to share a specific set of credentials with the dApp such as name, address, and if he is over 18 years old. Oliver then accepts to share his personal information and the dApp then verifies the authenticity of the credential through the blockchain.

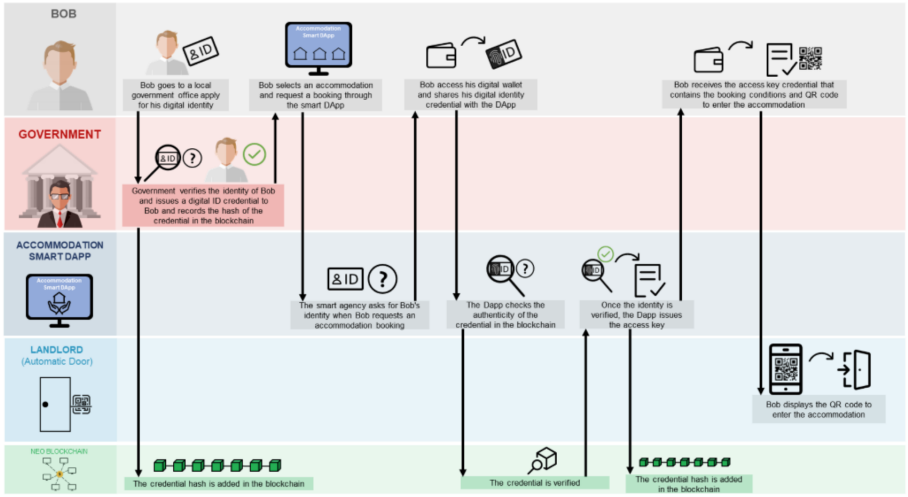


Figure 4: Demo workflow: roles and interactions.

Once the identity is verified, the dApp issues an access key credential to Oliver with information to access the accommodation such as the address, room, date from, stay duration, and a code that gives access to an automatic door to enter the accommodation.

Oliver has arrived in Barcelona today. Once he arrives at his accommodation, he displays the booking confirmation to the landlord, or in this case, to an automatic door featuring an IoT device with the ability to verify Oliver's claim.

Once the booking is verified, Oliver is now able to enter the accommodation without sharing any unnecessary additional personal data.

5 Seraph ID characteristics

5.1 Identity uniqueness

As with every other SSI project, NEO needs to have its own decentralized identifier (DID). The Neo DID definition[9] describes how to enforce uniqueness.

5.2 Authentication

"Authentication is the mechanism by which a DID subject can cryptographically prove that they are associated with a DID."[10] It is possible to define a correspondent section within the DID definition, which would allow it to support different methods for authentication. Different authentication methods are also necessary, based on the type of agent (edge or cloud) that the actor is using to interact with his own identity. Nevertheless, as NEO is currently working for providing a native X.509 PKI infrastructure (NeoID), an obvious option could be to leverage that solution.

5.3 Encryption

Seraph ID is leveraging NEO native primitives:

1. Encryption algorithm
    - ECC: secp256r1
2. Signing algorithm
    - ECDSA
3. Hash algorithm
    - RIPEMD160
    - SHA256
    - Murmur3
    - Scrypt

[9] `https://github.com/swisscom-blockchain/seraph-id-sdk/blob/master/DID.md`
[10] `https://w3c-ccg.github.io/did-spec/`, Authentication section.

5.4 Off-chain vs On-chain project

While designing Seraph ID, we have been trying to focus on the privacy-by-design principle. That is why most of the components provided within the Seraph ID ecosystem are meant to be handled off-chain. As we will see later in this document, the amount of data stored in the ledger is quite small, related either to public information (issuer identity, revocation registry) or private information that is hashed (hash of credential ID).

5.5 Zero-Knowledge proof

Despite our intention to support zero-knowledge proofs in the future, this capability is not taken into consideration in the current design of Seraph ID. Nevertheless, we designed the current version considering that, even if a claim takes the form of a set of attributes, the verifier should be able to request to read just a single specific attribute, providing a more precise authorization request to the subject.

5.6 Credential Schemas

Each issuer can define and store its own list of credential schemas. A schema is a generic definition of the credential's structure itself, a key-value pair of attribute-name/attribute-type. Every credential schema is stored in the issuer's smart contract storage, and it can be queried by third parties, for example by a verifier. A verifier would need to retrieve credential schemas for two different purposes:

Claim validation A claim shared by a specific subject should reflect the schema defined by the issuer. If not, it could represent the first warning that the claim has been manipulated.

Selective disclosure A verifier could need to request just a specific attribute of a schema (e.g., not the entire passport, but only the date of birth of a subject). To do that, the verifier would need to know what the schema looks like.

5.7 Seraph ID components overview

In the following table, we summarize the list of components that are available for the NEO developer community. Those components can be used to enable SSI capabilities in NEO dApps (based on the role your dApp needs to cover).

Actor	**Component**	**Usage**	**Features**	**Usage example**
Issuer	Library	Front-end	Generate claims offline.	Add this library to your front-end and generate a claim based on a specific event.
	Smart contract	On-chain	Store claim ID (ID generated by the issuer). Revoke claims. Manage schemas. Manage issuer public information.	Extend your smart contract with the following method to enable it to support SSI or modify and deploy it.
ROT-Manager (Root-of-Trust)	Smart contract	On-chain	Manage issuer whitelist.	Deploy this smart contract and act as a ROT-Manager.
Verifier	Library	Front-end	First validation of claims (offline). Enhanced validation of claims (online).	Add this library in your front-end to interact with an identity owner and validate its claim.
Identity holder	Library	Front-end	Managing DID. Storing claims. Authorize sharing.	Add this library to your mobile app to increase support for SSI.

Table 1: SeraphID available components.

6 Interactions

The following section describes the main interactions between actors and components. For more information regarding the methods listed in the sequence diagrams, please refer to Seraph ID GitHub.

6.1 Roles

Holder Holder is interacting with the system through a wallet. Different types of wallets can be built using the *"identity holder library"*.

Issuer Issuer is a publicly available identity registered on the blockchain through the process of deploying the *"issuer smart contract"*. The user interface provided by the issuer needs to embed the *"issuer library"*.

Verifier Verifier is providing a specific online front-end that has been built using the *"verifier library"*. The front-end itself is also built based on the specific use-case of the verifier. E.g., A dApp embedding the verifier library can easily request to each user a specific claim.

ROT-Manager A public entity that is represented online by deploying the *"ROT-Manager smart contract"*. The ROT-Manager will be able to list and delist trusted issuers.

6.2 Issuing a claim

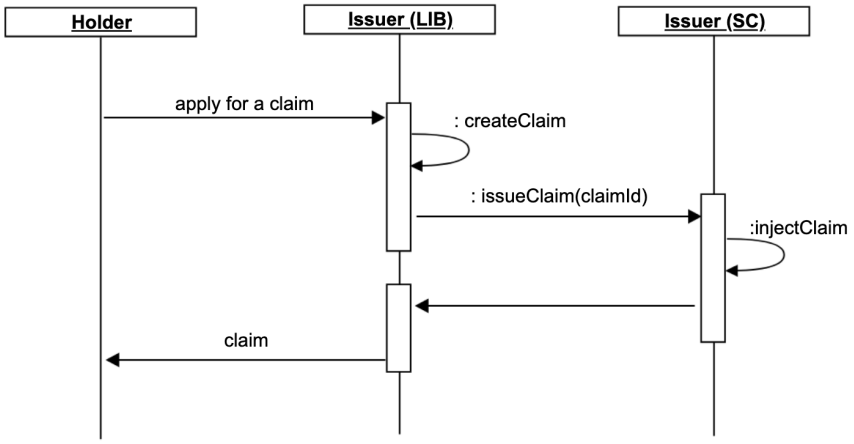


Figure 5: Issuing a claim

Precondition Identity holder has generated locally, in their wallet, their own DID.

Process Once a request is received by an issuer for generating a claim the issuer will check the content of the request itself. If the request is considered valid from the issuer, the claim will be generated. Secondly the issuer will store the claim ID on the blockchain (useful information for the revocation registry). Finally the claim will be sent to the holder who initiated the request.

6.3 Verifying a claim

Precondition Verifier trusts a specific issuer and verifier has got the claim from the identity owner.

Process Once the verifier has the claim from the identity owner, he must ask the issuer the schema details on which the claim was generated from. This will be useful to check if the claim has been modified (it does not reflect the schema structure). Then the verifier can execute a specific verification process, offline (not connected to the blockchain or online). Seraph ID offers three different levels of verification. Each developer can choose which one to use based on his use case and the performance he wants to achieve.

Method	**Description**	**Type of validation**	**Reference**
verifyOffline	It will validate the claim just using the contained information. "Offline" because it does not connect to the blockchain.	Lightweight validation which helps to verify if the claim has been signed correctly by the issuer at the time of generation.	Figure 6
verify	It will connect to the blockchain in order to validate the current public identity of the issuer.	This type of validation connects to the blockchain in order to check the identity of the issuer in real time.	Figure 7
validateClaim	It will connect to the blockchain, check the issuer public identity but also the revocation registry.	This validation is the most complete one as integrates the previous and checks the revocation registry.	Figure 8

Table 2: Possible verification methods.

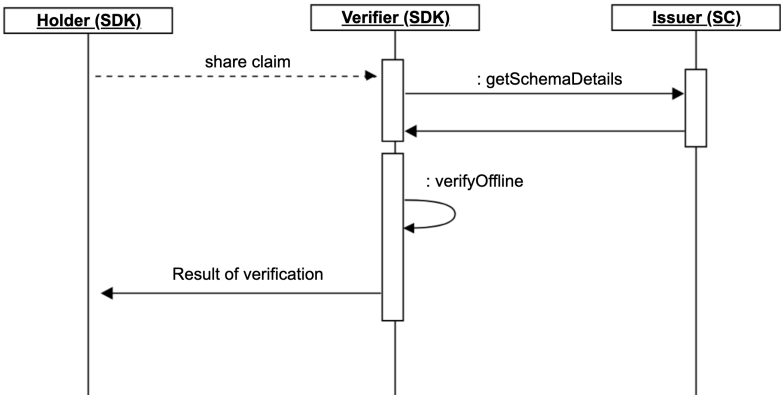


Figure 6: Verifying claim offline

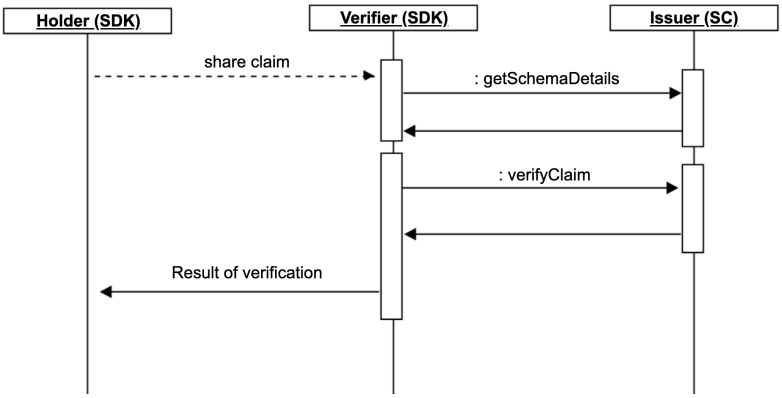


Figure 7: Verifying claim

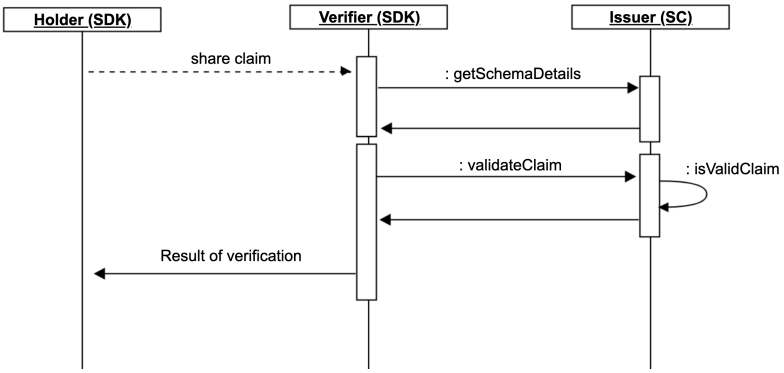


Figure 8: Validating claim

6.4 Verifying a claim with a ROT-Manager

Precondition Verifier trusts a specific ROT-Manager.

Process Once a verifier has a claim, issued by an unknown issuer, he could ask his ROT-manager by querying the issuer whitelist to ensure if he should trust the issuer or not. If the issuer is included in the whitelist, then the verifier can move forward with the normal verification process (explained in the previous chapter).

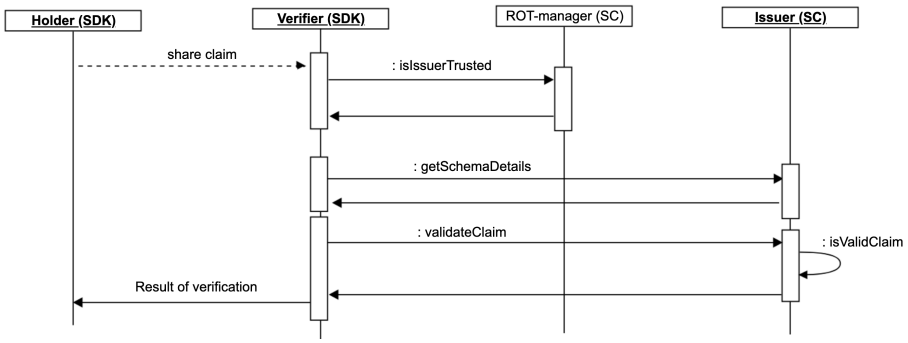


Figure 9: Verifying a claim with a ROT-Manager

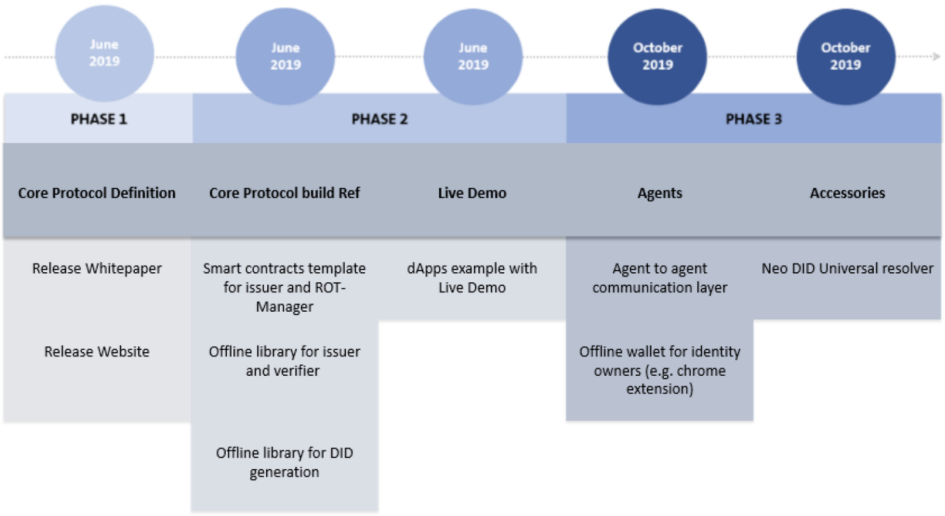
7 Seraph ID technical details

Please refer to Seraph ID GitHub:

- Smart contacts [11]
- SDK [12]
- Demo [13]

[11] https://github.com/swisscom-blockchain/seraph-id-smart-contracts
[12] https://github.com/swisscom-blockchain/seraph-id-sdk
[13] https://github.com/swisscom-blockchain/seraph-id-demo

8 Roadmap



9 Terminology

In the following section, we introduce the terminology that every reader should be familiar with. For terms related to the general SSI concept, we refer to the official terminology defined in W3C Terminology.[14]

The following list has been updated according to the Seraph ID scope, removing some terms not related to it and adding others.

Claim An assertion made about a subject.

Credential A set of one or more claims made by an issuer. A *verifiable presentations* is a tamper-evident credential, that has authorship, which can be cryptographically verified. Verifiable credentials can be used to build verifiable presentations, which can also be cryptographically verified. The claims in a credential can be about different subjects.

Decentralized identifier Portable URL-based identifier, also known as a *DID*, associated with an entity. These identifiers are most often used in a credential and are associated with subjects, such that a credential itself can be easily ported from one repository to another without the need to reissue the credential. An example of a DID is did:example:123456abcdef.

[14] `https://w3c.github.io/vc-data-model/`, see terminology section.

Decentralized identifier document A document that is accessible using a verifiable data registry and contains information related to a specific decentralized identifier, such as the associated repository and public key information.

Digital signature A mathematical schema for demonstrating the authenticity of a digital message.

Entity A thing with distinct and independent existence, such as a person, organization, concept, or device. An entity can perform one or more roles in the ecosystem.

Graph A network of information, consisting of subjects and their relationship to other subjects or data.

Holder A role, an entity can perform by possessing one or more verifiable credentials. A holder is usually, but not always, the subject of the verifiable credentials they are holding. Holders store their credentials in credential repositories.

Identity The means for keeping track of entities across contexts. Digital identities enable tracking and customization of entity interactions across digital contexts, typically using identifiers and attributes. Unintended distribution or usage of identity information can compromise privacy. Collection and use of such information should follow the principle of data minimization.

Identity provider An identity provider, sometimes abbreviated as *idP*, is a system for creating, maintaining, and managing identity information for holders, while providing authentication services to relying party applications within a federation or distributed network. In this case, the holder is always the subject. Even if the credentials are bearer credentials, it is assumed the credentials remain with the subject, and if they are not, they were stolen by an attacker. This specification does not use this term, unless comparing or mapping the concepts in this document to other specifications. This specification decouples the identity provider concept into two distinct concepts: the issuer and the holder.

Issuer A role an entity might perform by asserting claims about one or more subjects, creating a verifiable credential from there claims, and transmitting the verifiable credential to a holder.

Repository A program, such as a storage vault or personal verifiable credential wallet, that stores and protects access to holder credentials.

Selective disclosure The ability for a holder to decide what information to share with fine-grained precision.

SSI-network (Seraph related) A group of entities, that spontaneously decide to agree on a set of business, legal, and technical rules to enable the process of issuing and validating credentials. Each entity within the SSI-network is then considered a trusted party within the network itself.

Subject An entity about which claims are made.

Root-of-Trust manager (Seraph related) A role an entity can perform by governing an SSI-network. A ROT-Manager defines a set of business, legal, and technical rules that any participant should adhere to in order to be onboarded and to participate in the specific SSI-network.

User agent A program, such as a browser or other Web client, that mediates the communication between holders, issuers, and verifiers.

Validation The assurance that a verifiable credential or a verifiable presentation meets the needs of a verifier and other dependent stakeholders. This specification is constrained to verifying verifiable credentials and verifiable presentations, regardless of their usage. Validating verifiable credentials or verifiable presentations is outside the scope of this specification.

Verifiable data registry A role a system might perform by mediating the creation and verification of subject identifiers, verifiable credential schemas, revocation registries, and issuer public keys. Some registries, such as ones for UUIDs and public keys, act as namespaces for identifiers.

Verification The evaluation of whether or not a verifiable credential or verifiable presentation complies with a given specification.

Verifier A role an entity might perform by receiving one or more verifiable presentations for processing. Other specifications might refer to this concept as a *relying party*.

URI An identifier as defined by [RFC3986].