2 | Problem Description

The goal of this thesis project is to realize practical verification of an actor's qualifications on a scalable Self-Sovereign Identity infrastructure. We can break this down into two major problems:

> **Problem 1** Verify the qualifications of an actor in a self-sovereign manner.
>
> **Problem 2** Facilitate a multitude of use cases on a self-sovereign identity infrastructure.

Our second problem is about solving multiple instances of the first problem with a network of actors. To understand the requirements of such infrastructure, we will start with breaking down the first problem. We will use the following scenario:

> Alice and Bob engage in some transaction which involves risk to Bob. To minimize his risk, Bob must have confidence that Alice *qualifies* for the intended transaction. Therefore, Bob relies on *claims* about Alice made by another actor, Chris.
>
> If Chris's claims may somehow be *false*, Bob is again at risk. So Bob must either *trust* Chris with this, or gain confidence that Chris himself qualifies for making such claims. For this, Bob relies on claims by yet another actor, making the problem recursive.

Figure 2.1 illustrates this scenario with Alice as SUBJECT, Bob as VERIFIER of CLAIMS made by Chris, the ISSUER. In the second part of the scenario, Chris plays the role of SUBJECT [1].

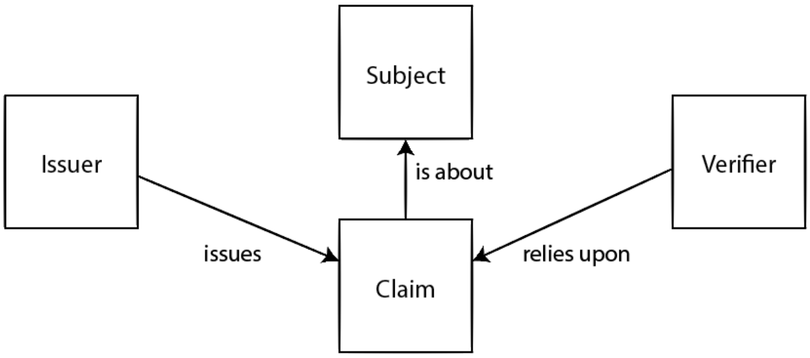

Figure 2.1: Semantic Model of Claim Based Identity

[1] Some related work [1] [2] considers a slightly different scenario where Alice herself makes Claims, which are then *attested to* by an other actor, Chris.

2.1 Cryptographic and Semantic Layers

Existing Self-Sovereign Identity infrastructures – such as IRMA, Sovrin, uPort and Trustchain – realize the *cryptographic trust* in this scenario. They ensure that Alice can forward claims from Chris to Bob whilst preserving integrity and allow for selective disclosure claim contents or even proving something about the claim without disclosing it at all (known as Zero-Knowledge Proof of Knowledge); e.g. instead of disclosing a date of birth, simply prove that it was at least 18 years ago. The use of digital signatures ensures that claims cannot be tampered with (integrity), and the issuer cannot deny making the claim (non-repudiation). In different ways, these solutions also provide means for *revocation*; i.e. withdrawing the validity of a claim before it expires.

A single *use case* is a specification of this model describing (a) *what* qualifies Alice for their intended transaction (e.g. a minimum age, an authorization, mastery of a skill), and (b) which claims and which sources Chris can rely upon to gain confidence in that qualification (e.g. a passport issued by Alice's government). Alice, Bob and Chris are real world entities such as humans or their collective organisations, institutions and governments. So between a use case agnostic cryptographic layer and any *meaningful* application we must add precisely that: meaning. This calls for another layer in which actors can coordinate *what* to exchange and *whom* they trust. This layer must perform the mapping between cryptographic elements – such as pseudonyms, claims and signatures – and real world entities, social relations and business logic. We refer to this layer as the *semantic layer.*

This semantic layer could, in principle, be fully realized by applications running on a purely cryptographic infrastructure. However, we will argue in the next section why it is better to handle at least part of the semantics in the shared infrastructure.

2.2 The need for a Common Semantics Layer

As argued in the introduction, the key to a shared identity infrastructure is to find the right point of decoupling; i.e. the correct *separation of concerns.* Note that not all semantics or business logic can be fully facilitated by a shared infrastructure (as Cameron has argued), hence applications must at least *complete* this semantic layer. Henceforth, we will distinguish between the Common Semantics Layer (CSL) and the Application Layer (AL). We will argue why such Common Semantics Layer is necessary.

2.2.1 Reuse of claims

First of all, Allen's principle of *interoperability* states that a subject should be able to reuse her claims with as many parties as may need them. In other words, issuers should not need to reissue or redesign their data for every new use case or application.

2.2.2 User Control: a Commons in need of Semantics

Furthermore, Alice's data should be protected at all times. No party should be able to come between her and her data. If applications were to handle the entire semantic layer, they would be able to tamper with Alice's *understanding* and *choices*. For example, consider the popup on many European websites asking the user's consent to store cookies[2]. The content of this agreement is fully designed by the website provider, and there is no easy way for the user to be sure that her choice is actually respected: she cannot prove that she clicked *deny*.

Therefore Alice must have an application that makes transparent all that that she has and what happens to it. This application should be independent, hence free of conflicts of interest, of any other actor. We can imagine several applications being created for different user's needs, but all must be open source, transparent and deal with a multitude of use cases an a way that still provides the relevant meaning to Alice. In line with this thought, IRMA, Sovrin and uPort offer such applications called Wallets or Agents which provide a single user interface to the subject. Trustchain limits its concerns to the *cryptographic layer*.

2.3 Building on top of Trustchain

In contrast to IRMA, Sovrin and uPort, Trustchain offers full peer-to-peer anonymity by using the TOR protocol for communication. Furthermore, it uses an alternative distributed ledger technology that does not rely on global consensus, making it a scalable infrastructure. It has already been used in trials with the Dutch government and offers security at the level of the Dutch passport. This makes it perfect for studying in the context of this thesis.

As opposed to the commercial and academic alternatives, Trustchain does not yet have a common semantic layer in place. Several implementations of Wallets and other applications have been made by students, but these do not yet provide the desired semantic functionality. This section dives deeper into the technology of Trustchain. Section 2.4 describes in more detail the desired properties of the semantic layer.

Stokkink and Pouwelse presented a claim model for Blockchain-Based Self-Sovereign Identity [2] that meets passport-level use requirements by facilitating legally valid signatures. Their model consists of five claim metadata fields: name, timestamp, validity term, proof format and proof link. They also present three models for using claims, to satisfy different requirements: a passive, an intent-based and an active model. In the passive model the issuer and subject together sign an attestation. Any verifier can simply look up the claim metadata and verify the accompanying signatures to check its validity. However, if revocation is required, the subject may have the ability to withhold the revocation information to a verifier. The intent-based model makes this form of identity fraud evident by adding an intent block to the chain pointing to the attestation that was verified. Any auditor may now verify that no revocation

[2] EU directive `https://eur-lex.europa.eu/legal-content/NL/TXT/?uri=LEGISSUM:l24120`, Dutch law `https://wetten.overheid.nl/BWBR0009950/2020-03-01`

was done before the verification. Identity fraud may now become evident, but it is still possible. The third model basically requests a new attestation from the issuer using a unique challenge to prevent a replay attack. In this way, revocation is basically facilitated by actively refreshing the credential. Note that this does not include any schemes.

2.4 Common Semantic Layer Properties

This section breaks down the concerns of the Common Semantic Layer.

2.4.1 Meaning and Value of Claims

The meaning of *trust* depends on the context, or in this infrastructure: on the use case. We never have *generic trust* in a party: we may trust a bank with our money but not with our darkest secrets. Hence, when we speak of trust, we actually mean *confidence* or a *belief* in a certain aspect of a party: its name, its honesty, its competence, its authority, etc. It is therefore essential that issuers, verifiers but also subjects agree on *what* is being claimed or attested to; i.e. the MEANING of the claim.

Secondly, they must understand the VALUE of the claim. We can think of this value as the risk in relying upon that claim; the probability of it being false and the magnitude of the consequences. As the goal of the initial exercise was to minimize the risk, we must understand this risk. We consider the following aspects:

ISSUER ACCOUNTABILITY. The Verifier may mitigate this risk by deferring it to the Issuer, either through warranty, liability or compensation. Since this puts the Issuer at risk, he must have incentive to issue the claim anyway.

ISSUER QUALIFICATION. If accountability is not applicable, or insufficient for the Verifier, the risk of incorrect claims may be reduced by understanding how the claim came to be. This problem is similar to our original problem, *Subject Qualification*, only now we have the additional constraint that evidence must be passed through the original *Subject*. The type of qualification again strongly depends on the use case; it may be honesty or competence in many forms.

ACTUALITY. When claims are based on values that can change (such as a home address or salary), an important aspect of the claim's value is its ACTUALITY. Note that the Verifier may wish to have the most current information about the Subject, so it may be risky to act on an outdated claim. We must also consider historical assertions, i.e. what was the Subject's income at the start of this year.

Standardization of meaning and value of information has been common practice for a long time. The latest development in respect to Self-Sovereign Identity is the new Recommendation from the World Wide Web Consortium (W3C): Verifiable Credentials.

2.4.2 Extending Trust

The value of claims is established by the issuer. If this process is somehow corrupted, the Verifier relying on that claim is at risk. The Verifiable Credentials scheme assumes that the Verifier trusts the Issuer directly to deliver correct claims. However, in many practical scenarios, this assumption of direct trust is too restrictive. Consider the following problem:

Alice is an employee of a company called Dave's. She wishes to sign a contract with a supplier Bob, in name of Dave's. Bob needs confidence that Alice is authorized to sign such contracts, so he wishes to Verify her authority. Alice provides claims signed by her boss Chris, the owner of Dave's. Bob does not know Chris, so how can he trust his claims?

The rest of this thesis will focus on problems of this kind, described in more detail in the next chapter. There are several ways for Bob to know that Chris owns Dave's and hence has the required authority. For example, a public record could list Chris as the owner. However, this approach sacrifices privacy for security. So we add the following constraint: any intermediate untrusted issuers should be able to remain anonymous so long as identification is not necessary for the use case.

2.4.3 User Convenience

The adoption of Self-Sovereign Identity will offer more and cheaper assurance in identity transactions to relying parties, and more control to subjects. However, with this control comes the responsibility to manage the collection and distribution of all these claims. This likely becomes too cumbersome for Alice, which may slow down the adoption of SSI or force an overwhelmed Alice to make poor choices that could harm her privacy.

It is often Alice's responsibility to deliver and hence fetch the necessary claims. Especially in a multi-issuer use case such as with diploma's, only Alice knows her issuer, i.e. her school. The most common approach seen in many recent SSI use cases to date is what we call *portal based.* The issuer provides a web portal where Alice can log in, using SSI claims or other authentication mechanisms. There, the relevant claims can be requested by scanning some QR code.

Many approaches seem to assume that the bulk of Alice's claims are gathered beforehand, that is before they are requested by a Verifier. Whilst we feel this is the safest approach for Alice, as she can then protect her claims for later use, we imagine the common individual will not go through the trouble of building their identity unless necessary. In other words, we believe that most issuings will happen *ad-hoc*, on a request-basis. In that scenario, the portal based approach is not the most efficient as it requires Alice to do a lot of work and still does not offer her a consistent user experience (Cameron's seventh law).

2.4.4 App/System Integration

As discussed in the introduction, this infrastructure will be operated primarily on user's smart phones. Issuing and Verifiying organisations will likely want to manage their processed in an automated fashion on cloud infrastructure. To make this semantic layer suitable for both cases, we assume the following model as our technical context:

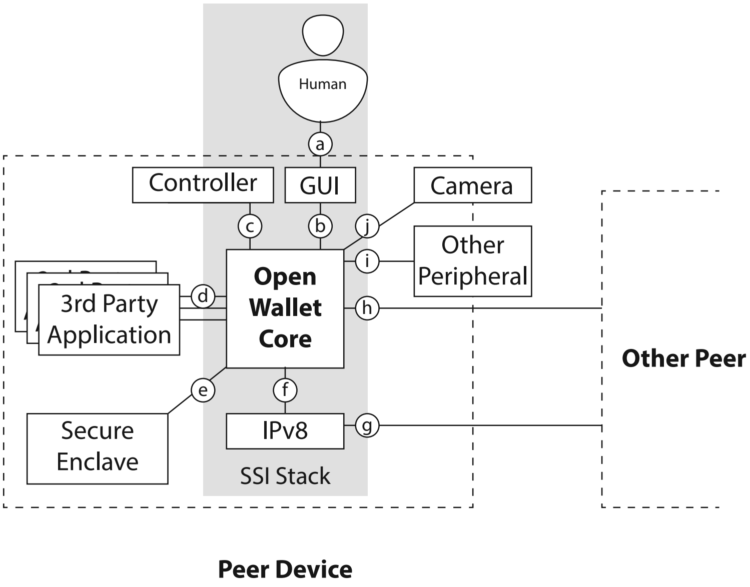

Figure 2.2: Technical Context of the Semantic Layer

Figure 2.2 shows the technical context in which the Agent operates. The subject that owns the Agent can either control it manually via a Graphical User Interface (a), or in an automated fashion using software (c).

3rd party applications running on the device may interact with the user's sovereign identity through the Open Wallet Core (d). These differ from the programmed controller (c) in terms of authority: the Controller is assumed be a trusted representative of the Subject and hence has full control, whereas 3rd party applications who need a programmable interface are not immediately trusted so their control is limited.

The Open Wallet Core assumes availability of a Secure Enclave (e) for storing keys and private data and on mobile devices it makes use of a camera (j) for scanning QR codes or other peripherals (i) such as Near-Field Communication (NFC).

It makes use of the Trustchain library as introduced before (f). The stack is designed in such a way that this library, responsible for the core identity operations such as signing, is only operated by the Open Wallet Core. Finally, each Peer can communicate with other peers running this stack through the low level Trustchain protocol or a higher level Open Wallet protocol.

3 Example Problem: Extended Power of Attorney

The wide variety of use cases that could fit the model in Figure 2.1 makes it difficult to design for. Hence, within this thesis we will focus on a subset of problems, whilst tackling one of the more complex issues in identity systems: *delegation*. We consider the following problem:

> **Problem:** verify that a person authorized to act on behalf of some organisation.

In collaboration with the Dutch Chamber of Commerce, we develop an application that provides high legal assurance, but also convenience, in verifying such authority. This is a complex case that heavily depends on the ability for individuals to digitally identify and sign, making it an appropriate test for our system. We will limit our scope to Dutch legal entities, natural persons and legislation.

Our goal is to answer the following question in an automated fashion:

> **Assertion:** $Q(P, L, A)$ = person P is authorized to perform action A on behalf of legal entity L.

In general, i.e. for any combination of P, L and A in any legal system, answering this is extremely hard, if not impossible. Hence, for high risk transactions, a notary is called in to perform various checks, thereby consulting several registers. He then produces a *legal opinion*, which is an official statement of his findings.

For many day-to-day activities, however, such rigor is unnecessary and business can be conducted by mutual trust and the laws of *Power of Attorney* (volmacht). In this section, we derive a simplified reasoning model based on Dutch legal texts and expert interviews. We will limit our scope to those actions A for which the question $Q(P, L, A)$ can be safely answered with this model. We consider the following elements:

1. Registered directors (*functionarissen*) who control a legal entity.
2. Registered Power of Attorney (*volmachten*) granting a natural person full or partial power over a legal entity.
3. Unregistered Power of Attorney based on a shared platform.

3.1 Executive Power over Legal Entities

Depending on its form, a legal entity is owned and controlled by one or more natural or legal persons, called *directors*. This relationship is registered in the trade register (or *Handelsregister*), managed by the Chamber of Commerce.

Human Director. The role of director may be assumed by another legal entity, which can repeat itself several times. At the top of such a chain, one or more natural persons are always in control.

Signing capacity. If a legal entity has more than one director, it may be the case that directors cannot individually sign contracts but instead must do so together[1]. We will however constrain our problem to the case where a director has full signing capacity over a legal entity.

The natural person forms the root of a tree of legal entities. The laws, and possibly other conditions, determine the extent of power this person has over each of the legal entities in the tree. We will abstract over these complexities by assuming the following:

> **Assumption 3.1.** For each combination of legal entity L and natural person P, the Chamber of Commerce has the ability and authority to state whether P has full control over L.

Next, we consider how power can be extended to other legal and natural persons.

3.2 Extending Power over Legal Entities

Whereas the human directors act as the root of the command hierarchy, enjoying unlimited control[2], they may extend their power to other legal or natural persons by means of *Power of Attorney* (or *volmacht*). The Dutch law states that Power of Attorney can be made *explicitly* or *implicitly.*[3].

3.2.1 Explicit Power of Attorney

The Chamber of Commerce explicitly registers Power of Attorney using a form (*Formulier 13*[4], see appendix). It allows to either grant the subject full authority, or restricted by options shown in Table 3.1. This also allows to fill in a restriction in natural language. As this is inconvenient for automation purposes, the Chamber of Commerce is developing a semantic model that can replace this free-form text.

The Explicit Power of Attorney method has two processes:

1. **Issuing.** To issue a Power of Attorney, the *grantor* (or an authorized representative) must fill out the form and visit one of the offices of the Chamber of Commerce together with the person being granted.

2. **Verifying.** To verify an explicit Power of Attorney, one can retrieve an excerpt (or *uittreksel*) at the Chamber of Commerce website[5] at the cost of € 2,30. The verifier must then compare the full name and date of birth of the attorney, as stated on the excerpt, with some form of legal identification.

[1] https://www.kvk.nl/advies-en-informatie/fraude/tekenbevoegdheid-per-rechtsvorm/

[2] As by our assumption

[3] Burgelijk Wetboek 3 Artikel 61:1

[4] Formulier 13 Inschrijving Gevolmachtigde

[5] https://www.kvk.nl/producten-bestellen/bedrijfsproducten-bestellen/uittreksels/

By Financial Amount	Maximum amount in Euros
By Act	One or more of the following: *Requesting changes in the Trade Register, Issuing Quotations, Access RDW license plate services.*
By Contract Type	One or more of the following: *Purchase, Sales, Warranty, Lease (Rental), Financing, Software, Maintenance* and/or *a custom description*
By Establishment	Entire legal entity, or specified to a specific establishment (by address).

Table 3.1: Restriction options for registered Power of Attorney

3.2.2 Implicit Power of Attorney

The Power of Attorney method is a tedious process both when filing it and when checking it. Suppose a customer checks out at the local grocery store. Before handing over the money, he must check if the cashier actually has Power of Attorney to receive that money. As this situation is far from practical, Dutch law provides the concept of *implicit Power of Attorney*:

> Is een rechtshandeling in naam van een ander verricht, dan kan tegen de wederpartij, indien zij op grond van een verklaring of gedraging van die ander heeft aangenomen en onder de gegeven omstandigheden redelijkerwijze mocht aannemen dat een toereikende volmacht was verleend, op de onjuistheid van deze veronderstelling geen beroep worden gedaan. - Burgerlijk Wetboek 3 Artikel 61:2[6]

Freely translated, this means that if a counter party may reasonably assume that power of attorney was granted to a particular person, he may act upon that assumption. The assumption must be based on a *statement* or *behaviour* by the *grantor* and take *circumstances* into account.

In the cashier example, the customer (being the counter party) may reasonably assume that the cashier (the representative, or attorney) has Power of Attorney to receive the cash in name of the grocery store (the grantor). This can be based on the circumstances, i.e. the cashier wearing company clothing and sitting behind the cash register, and the behaviour of the grantor by (its other personnel) not removing that person.

3.2.3 Sharing Authority Information

The aforementioned methods of explicit and implicit Power of Attorney does not suit all transactions. Whereas the physical space allows customers to easily (yet superficially) assess the identity (face) and attributes (company clothing and placement) to support a *reasonable assumption*, the digital space is by default much less transparent.

This has lead to alternative approaches, such as the Dutch E-Herkenning

[6] https://wetten.overheid.nl/BWBR0005291/2020-01-01/#Boek3_Titeldeel3_Artikel61_Lid2

system[7], a cross-organisational identity and access management platform. The solution roughly works like this:

1. Digital service providers to list their online products and services in a central catalog, along with the *level of assurance* they require.
2. Consumers register an E-Herkenning account for each participating employee, at an annual cost of around 5 to 30 euros, depending on the level of assurance.
3. The directors of the consuming legal entity are given full access and must subsequently grant access to specific products and services to individuals, as they see fit. They may also grant other individuals the authority to manage access.
4. As an individual wishes to use a service in name of a legal entity, they must log in with their E-Herkenning account at the right level of assurance. The account must be authorized to use the service in question.

A major drawback of this approach is that the catalog of products and services quickly becomes to large to comprehend. It also requires managers to foresee exactly which products and services their people will need in the future. When access is too restricted, employees are stalled in their work and have to bother their superiors for granting extra access. This quickly leads to dangerous practices such as lending the credentials of a colleague, or simply granting individuals with full access. ▷ Verify these issues

The aim of this design project is to provide a simple, yet effective way for employees to share authorizations when it suits them. Note that the E-Herkenning system separates the right to access services from the right to grant access to those services. In this design iteration however, we will simplify the problem by choosing these to be equal.

3.3 Actions versus Jurisdictions

In the previous two sections, we have outlined the basic forms of control natural persons can have over legal entities. The extent of this control is usually described in general terms (e.g. *sales*, or *purchases up to 5mln*). We will refer to this as the *jurisdiction* of a person.

Before we can answer the question $Q(P, L, A)$ which considers a specific action A, we must find a way to map actions to jurisdictions. The E-Herkenning system solves this by simply expressing jurisdictions in terms of actions, i.e. the mapping is one-to-one. This solution is however not practical for many other cases, as it would require ticking a box for each possible allowed product or service.

For our current model we define a jurisdiction J as a subset of the set of all actions. We say that if and only if $A \in J$, then the action A is allowed under

[7] https://www.eherkenning.nl/

the jurisdiction J.

TODO: *Who defines these semantics?*

3.4 Qualification Logic

From the previous sections we can derive the following predicates:

Predicate	**Meaning**	**Trusted Issuer**
$Full(P, L)$	states a person P has full control over a legal entity L.	Chamber of Commerce
$PoA(P, L, J)$	states that a person P has registered Power of Attorney over L restricted to some jurisdiction J.	Chamber of Commerce
$Auth(P, L, J)$	states that a person P is authorized to act in name of L within the boundaries of some jurisdiction J.	Any qualified person

Table 3.2: Predicates for determining authority

Note that we cannot assume truth, so we depend on claims by various parties. We mark the issuer of a claim by a subscript; $Full(P, L)_{KVK}$ means that the Chamber of Commerce (KVK) states that P has full control over L.

$$Full(P, L)_{KVK} \rightarrow Q(P, L, A) \tag{3.1}$$

$$PoA(P, L, J)_{KVK} \textbf{ and } A \in J \rightarrow Q(P, L, A) \tag{3.2}$$

$$Auth(P, L, J)_P \textbf{ and } A \in J \textbf{ and } Q(P', L, J') \textbf{ and } J \leq J' \rightarrow Q(P, L, A) \tag{3.3}$$

$$\triangleright \text{ We can also simplify by using } Q(P, L, J)$$

$$\triangleright\ Q(P, L, J) and J \geq J' \rightarrow Q(P, L, J')$$

TODO: *Check formal logic notation*

[8]

Note that claims $Full(P, L)$ and $PoA(P, L, J)$ are only trusted when issued by the Chamber of Commerce. In contrast, a claim $Auth(P, L, J)$ can be made by any party P', but then requires that P' has sufficient jurisdiction; i.e. $Q(P', L, J')$ where $J' \geq J$.

As rule 3.3 is recursive, one may have a chain of claims that end with the subject P. For this chain to be valid however, it must start with a person P' for which one of the base cases hold; i.e. $Full(P', L)$ or $PoA(P', L, J)$.

Furthermore, we observe that in all cases the *grantor* reserves the right to revoke its claim.

[8] https://en.wikipedia.org/wiki/Propositional_calculus https://en.wikipedia.org/wiki/First-order_logic#Deductive_s

3.5 Usage Scenarios (or Context)

Answering the question $Q(P, L, A)$ is primarily relevant when parties are about to perform action A. Hence, our application must be able to provide this answer at that moment.

We first distinguish two usage scenarios: virtual and physical verification. In the virtual scenario, the subject P attempts to use an online service. Before proceeding, the service provider needs to know who P is, which legal entity L is involved and whether P is allowed to do A on behalf of L.

The physical scenario works rather similar. The counter party is now a human being asking the subject P for the same information. In this scenario we have the added benefit that the counter party can visually identify P.

In both scenarios we leave it to the counter party to map the desired action A to a jurisdiction J as defined in the system.

TODO: *Describe Scenarios. May fit better in design section.*

TODO: *Add identifiers of Legal and Human entities*