Eemsgolaan 3
9727 DW Groningen
P.O. Box 1416
9701 BK Groningen
The Netherlands

www.tno.nl

T +31 88 866 70 00
F +31 88 866 77 57

TNO report

Technical Report Studybits

Date	March 13, 2018
Author(s)	H.J. Jongsma, H.J.M. Joosten
Number of pages	27 (incl. appendices)
Number of appendices	<number of appendices>
Sponsor	Quintor
Project name	Studybits
Project number	<project number>

In case this report was drafted on instructions, the rights and obligations of contracting parties are subject to either the General Terms and Conditions for commissions to TNO, or the relevant agreement concluded between the contracting parties. Submitting the report for inspection to parties who have a direct interest is permitted.

Contents

1 Introduction

This technical document is part of the Studybits SBIR feasibility study for using blockchain related technology to support the Erasmus+ Exchange process.

1.1 Context: Erasmus+

Erasmus+ is the EU-funded program to support education, training, youth and sport in Europe. With a budget of EUR 14.7 billion, it aims to provide opportunities to over 4 million European citizens until 2020. One of the key actions of the program is to encourage the mobility of students, staff, trainees, etc., both within and outside of the European Union. The Erasmus+ Exchange program encourages the international mobility of students by providing the framework that enables students of European universities to take part of their studies abroad and subsidizing the related costs.

The exchange program currently depends on mutual trust, knowledge of the educational program of exchange partners, and formal agreements and contracts between the participating higher education institutions. Institutions that exchange a large number of students often have a close relationship and employees are well-informed about the programs at the partnering institutions. Fraud regarding the qualifications of the individual exchange students is virtually inexistent. However, this does not mean that there is no way to improve the exchange process. Student information is exchanged manually by the exchange officers through email in the form of text documents and scans of physical forms. Selecting the students that are allowed to partake in the exchange program is also a manual process aided by factsheets and prior knowledge about the courses offered at the receiving institution.

1.2 Pilot

The Studybits pilot targets a group of 20 students of the University of Groningen participating in the Erasmus+ Exchange program. As part of their exchange, these students will take courses at foreign institutions: the universities of Ghent, Uppsala and Göttingen. Together with the University of Groningen they form the U4 Network. The pilot will use a to-be-developed Proof of Concept for digitalizing the exchange of student information between the institutions using blockchain related technologies.

1.3 Related projects

Studybits is not the only project which aims to improve the exchange process of students within the Erasmus+ Exchange program by digitizing the data-exchange between higher education institutions. However, it is the first project that fully commits to doing so in a self-sovereign and distributed fashion. Some projects with both centralized and decentralized approaches are:

EMREX

EMREX is an online platform that facilitates the exchange of student records between higher education institutions in Europe. Students can initiate the transfer of data located at one institution to another one, in a secure and verifiable fashion. The platform is connected to the Student Information System of each participating institution, or in some cases a centralized national version of such a system. The system is decentralized in the sense that data is fetched from individual systems, aggregated by country, called National Contact Points (NCP).

A student can initiate a transfer from any webpage that implements a client for EMREX called a Student Mobility Plugin (SMP). The transfer than occurs as follows:

1. Student initiates a data transfer from the website of institution A
2. The SMP contact the centralized EMREX directory to obtain a list of available NCPs.
3. The student selects the NCP from which the data needs to be transferred.
4. The SMP redirects the student to the authentication page of the chosen NCP. After authenticating with the NCP, the student selects which data to share. This data can originate from multiple institutions connected to and serviced by the NCP.
5. The selected data is digitally signed by the NCP and send to institution A.
6. The SMP verifies the integrity of the received data using the public key of the sending NCP, to make sure it has not been tempered with.
7. If the data is deemed valid, it is imported into the systems of institution A.

Given that different institutions use different terminology and storage methods for student records, to ensure interoperability between institutions within the EMREX platform, data is exchanged according to the ELMO XML standard. The standard prescribes how to present information regarding students, institutions, courses, grading schemes, and results among other things.

European Student Card
The goal of the ESC project is to issue one single student card that can be used by all students of European higher education institutions. This is made possible by giving each student a unique digital identity and collecting part of the student information that are currently present at the various educational institutions in a central registry. With the help of the physical student card, parties can retrieve information about the student from the central registry.

Erasmus Without Papers
The aim of Erasmus Without Papers is to create a network of higher education institutions in which all information regarding the Erasmus+ exchanges is exchanged digitally between the different institutions. The participating institutions share student data, learning agreements and study results from their own information systems using a shared API framework.

1.4 Goal of this document

The first phase of the Studybits project investigates the feasibility of using blockchain-related technologies for supporting the exchange of students within the

Erasmus+ Exchange program. One of topics that is part of the investigation is to determine the requirements that are needed of a technical solution and to select the technologies on which to base this solution. To this aim we investigated Blockcerts and Sovrin: two blockchain and self-sovereign identity related projects. This document gives an overview of both technologies and argues which of the two projects seems to be the best fit for developing the Studybits Proof of Concept.

2 Project description

In the part of the project that has been assigned to TNO, two deliverables will be created.

D1: Modification of the customer journeys that were supplied by Quintor. These customer journeys describe the current manual process of producing and accepting student records and related information by the participating universities for the Erasmus+ exchanges. TNO will adapt the customer journeys according to the principles of self-sovereign identity and blockchain. The original and modified journeys can be found in the appendix of this document. The main differences between the different versions is described in chapter 5.

D2: Technical feasibility study. This is the current document. The feasibility study will serve as input for the SBIR phase two proposal for the Studybits project.

3 References

[The Known Traveller] World Economic Forum, Accenture. The Known Traveller, januari 2018, http://www3.weforum.org/docs/WEF_The_Known_Traveller_Digital_Identity_Concept.pdf

[GDPR en Blockchain] Finck, Michèle, Blockchains and Data Protection in the European Union (November 30, 2017). Max Planck Institute for Innovation & Competition Research Paper No. 18-01. Available at SSRN: https://ssrn.com/abstract=3080322 or http://dx.doi.org/10.2139/ssrn.3080322

[Erasmus+ Programme Guide] Europese Commissie, Erasmus+ Programme Guide, januari 2017, https://ec.europa.eu/programmes/erasmus-plus/sites/erasmusplus/files/files/resources/erasmus-plus-programme-guide_en.pdf

[What Goes On The Ledger] Andrew Tobin, Evernym, What Goes On The Ledger, april 2017, https://www.evernym.com/wp-content/uploads/2017/07/What-Goes-On-The-Ledger.pdf

[Blockcerts Critique] Jaap-Henk Hoepman, Blockcerts: Using blokchain for identity management is (mostly) ridiculous, September 2017, https://blog.xot.nl/2017/09/06/blockcerts-using-blokchain-for-identity-management-is-mostly-ridiculous/

[Blockcerts FAQ] Blockcerts, FAQ, https://www.blockcerts.org/guide/faq.html

[How Sovrin Works] Phillip J. Windley, How Sovrin Works, oktober 2016, https://sovrin.org/wp-content/uploads/2017/04/How-Sovrin-Works.pdf

[Technical Foundations] Drummond Reed, Jason Law & Daniel Hardman, Technical Foundations of Sovrin, september 2016, https://sovrin.org/wp-content/uploads/2017/04/The-Technical-Foundations-of-Sovrin.pdf

[Path to SSI] Allen, C., The Path to Self-Sovereign Identity, april 2016, https://www.coindesk.com/path-self-sovereign-identity/

4 Blockchain and Self-Sovereign Identity

4.1 Blockchain

The rising popularity of blockchain and distributed ledger technology has had an disruptive effect in many fields, especially within the financial sector, where cryptocurrencies have seen a sharp rise in popularity.

A blockchain is a ledger where every time a new set of entries (a 'block') is added to the ledger, this block is cryptographically linked to the last block. This creates a chain of blocks - a 'blockchain'. A distributed ledger is a collection of (identical) ledgers – usually blockchains. Each individual ledger is stored on a node (computer) in a peer-to-peer network. The nodes are constructed in such a way that all additions to the ledger that are on one node are synchronized, stored and replicated to the other nodes, using a consensus algorithm that does not need a central administrative authority or centralized data storage. As the number of (different!) parties managing the nodes / ledgers increases, it becomes increasingly difficult to change the content of the ledgers as soon as there is consensus in the network about such content. A well-managed distributed ledger therefore has the property that both the content and the order in which such content is added are preserved over time (fraud-proof). Note that since in principle any content can be basically recorded in a ledger, this does not guarantee that the content is actually true.

There are several distributed ledger types. A 'permissionless' ledger is one where everyone can run a node and take part in the network. You can only participate in a 'permissioned' ledger as a node if you have received permission from the party that manages the network. There are also 'public' ledgers - ledgers where everyone can add entries (items, transactions), and 'private' ledgers, for which you must have permission to do so.

4.2 Self-Sovereign Identity

The ability of distributed ledger technology to solve problems that traditionally required a (single) trusted party, and the high degree in which they are censorship- and tamperproof, make it an interesting technology for creating and supporting solutions for digital identities. In recent years, numerous identity-related projects using distributed ledger technology have been created.

4.2.1 *Goal*

The basic purpose of having digital identities is to facilitate business transactions in the electronic domain. Here, we understand a ‘business transaction’ to be the execution of some agreement that (implicitly, explicitly or by reference) specifies obligations and expectations of participants to one another, and that is (implicitly or explicitly) committed to by every such participant. Examples include a sale (e.g. in a web-shop), the registration of a mortgage, a student exchange, etc.

Currently, setting up such agreements and committing to them often requires the participation of human actors, e.g. for verifying identity documents, and for deciding whether or not the business that they represent should commit to such agreements. Enabling the construction of such agreements by electronic actors (computers) that the business party that they represent would commit to, would provide significant benefits, particularly in terms of efficiency gains. For example, waiting times for collecting, transferring and showing identity information to a validating actor, as well as the validation itself, will be reduced from what could be days to months, to a few seconds. Also, the validity of such information can be ascertained with significantly more confidence than when relying on human actors. We will go deeper into this in chapter 6.

4.2.2 *Context*

A party will commit to any transaction agreement when the projected net result of the transaction execution is positive, and the risks involved are acceptable. We can readily observe this when we consider how we ourselves engage in real life business transactions e.g. with shops, banks, government agencies, our work, etc.

Real-world parties (individuals) will assess the projected net result and the risk that is involved in their heads – i.e.: in their “information context”. We assume that each real-world party has an information context, which is its knowledge of what exists, how to classify the entities that it knows to exist, and how to reason with that. All this is quite intangible, but there are models (e.g. in the field of semiotics) that allow us to mentally envisage such information-contexts.

In order for a party to delegate the creation of transaction agreements and the making of commitment decisions to a computer, it needs several things.

First, it needs a mapping between its (real-world) information-context and the electronic world, where data (bits, bytes) represent this information. We can limit this mapping to

- the information that is needed for constructing the transaction agreement and reaching the commitment decision, and
- the business logic that evaluates the information to reach the commitment decision.

The result of this mapping is a set of statements that represent the information need, and a formalized set of business rules that a computer can evaluate.

Second, the party needs each argument, i.e. the actual application of the business rules to a set of actual data, to be valid. For the data, this means that its meaning (semantics) must be known and sufficiently correct, and the data must represent information that is actually true. Similarly, the business logic must be valid.

The use of semantic web technologies (RDF, RDFS, OWL, etc.) provides sufficient (technical) possibilities to come to grips with the semantics.

The truth of statements is a subjective matter: every party gets to decide on this for itself. This is a fact of life that is readily observed – e.g. I may find myself trustworthy, while others disagree: there is no single truth here. This does not preclude that parties may agree on what they think is true. In fact, one party may trust the judgement of another party with respect to the truth of specific statements (e.g. the trustworthiness of others). If a party decides to trust another parties judgement with respect to some statement type(s), verification of such statements can become very easy: all that is needed is an attestation to the truth by that other party.

The validity of the business logic of a party is also a subjective matter. We can leave this matter to the individual parties, because whenever a business party decides a business logic to be valid where this should not be the case, it is the party itself that will suffer the consequences.

We can summarize this as follows. In order for a business party to delegate business transactions to the electronic domain, it must formalize its business logic and deduce what kinds of statements it needs from that. Also, it must specify how the computer can assess the validity of such statements; preferably, this means that the party decides what other parties to trust for asserting the truth of statements of some kind.

4.2.3 Self-Sovereign Identities (SSI)

With Self-Sovereign Identities (SSI), the individuals themselves manage statements that say something about themselves – they are the subjects of these statements. They can obtain such statements from any organization that is willing and capable to issue them, and attest to their truth. An individual will have an electronic component, which we call a wallet (which might be an app on a mobile phone), that manages such statements and attestations, which consists of:

- obtaining such attestations;
- securely storing them;
- using them when another party, with whom the individual wants to conduct a business transaction, requires it in order to create the transaction agreement or to make its commitment decision.

Of course, in a SSI framework, we also have parties that apply an electronic actor to provide an (electronic) service to such individuals. Whenever an individual requests this electronic actor to provide its service, the electronic actor will

- specify the kinds of statements it needs in order to create and commit to a transaction agreement;
- specify which of these statements need to be attested to and if so, which parties it trusts for that;
- receive statements and attestations, upon which it will decide whether or not to commit to the transaction, and if so, provide the requested service.

Because this electronic actor relies on data (statements and attestations) that are provided, we also refer to this as a ‘relying party’ or RP. Please note that a RP represents a business party, but is not a business party itself – it is an electronic actor. A special kind of RP is that whose service is to issue attestations. We refer to such specializations as ‘Attestation provider’ or AP.

Thus, the contribution of an SSI framework to the automation of business transactions is that it facilitates the exchange and validation of data that is needed to create transaction agreements and commitment decisions – under the assumptions that the information needs and ways of reasoning with that can be made sufficiently explicit.

Another contribution of such an SSI framework is that it satisfies many of the commonly accepted privacy principles [Privacy Principles], and that it allows for a relatively simple alignment with EU-privacy regulations (see [GDPR and Blockchain]).

4.3 Advantages for students and higher educational institutions

Having a self-sovereign digital identity offers students the opportunity to manage their own student records, and to share these (in a reliable way) with other parties if they want to. For example, they can share their diplomas, study results and other activities in, for example, an application procedure or applying for subsidies, as is the case within Erasmus+.

In the context of life-long learning, a student with a (self-sovereign) digital identity can also maintain his knowledge development outside regular educational institutions by recording certificates of courses followed, as well as the development of his qualifications on the work floor, attested by his employer, by storing them in his own wallet (that is the app or device in which he stores his digital identity).

If (educational) institutions provide and use self-sovereign digital identities, they will not only help students, but it will also be possible for them to further automate processes that are still partly / mainly manual. For example, by digitally recording the requirements for taking a subject and accepting the self-sovereign digital identities of students, computers may be able to determine the suitability of the student to follow that course. The student will then be able to easily compile his own educational program.

Dutch universities want to attract foreign students. Admitting students without the proper qualifications, however, entails a major risks for a university. In addition, the validation of foreign diplomas can be a costly and time-consuming manual process, sometimes requiring the help of external agencies such as DUO and Nuffic. Using digital statements about the training of the student by foreign educational institutions, and statements from DUO (or foreign versions) about the accreditation of these programs, this process can be automated and be made faster and cheaper.

Educational institutions prove their students a great service by 'kick-starting' the digital identity of a student. Universities are large institutions with a lot of social trust. A statement from a university about personal information from a student is very valuable for this person.

4.4 Other (self-sovereign) identity projects

As mentioned earlier, there are several projects in the field of digital identity that use blockchain. A small selection of these projects is briefly described here. These projects have not been selected as candidates for application in the Studybits pilot.

Uport - https://www.uport.me/
Uport is an open source identity project that uses the Ethereum blockchain. The aim of the project is to create a single digital identity for users that can be used both on the Ethereum blockchain and beyond. The identity of the user is managed by various smart contracts, which makes things like key recovery possible. Information about Uport identities is not on the blockchain, but is stored outside of it, for example on the IPFS.

Civic - https://www.civic.com/
Civic is a commercial party that offers various apps and api's for storing, sharing and verifying personal data. Organizations from a closed network (Civic and its partners) confirm and certify the validity of different personal data of the user and embed this data on the blockchain. The user can then share this data with third parties who check this data using the blockchain.

OWLChain / BOSCoin - https://boscoin.io/
The OWLChain is a permissioned blockchain in which semantic data can be stored. OWLChain also supports smart contracts that can work with this semantic data. Although this technology does not specifically focus on identity, the storage of semantic data, and the linking of semantic data from different contexts and automatic processing and reasoning with this data is very interesting for digital identities.

5 Customer Journeys

Two customer journeys have been added to this as an appendix. These journeys describe the process that is followed by the exchange of a Dutch student at the University of Groningen, who participates in an Erasmus+ exchange with Ghent University. The first journey describes the process as it is currently taking place, the second journey describes what the process might look like when using a self-sovereign digital identity for the student. In both cases the student passes through roughly the steps in Figure 1. In this chapter we summarize the most important differences.

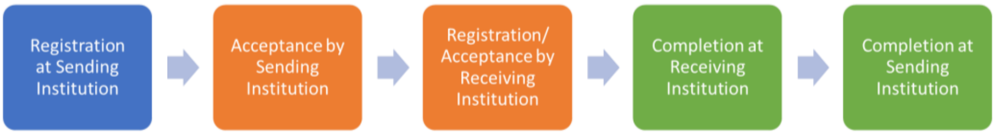

Registration at Sending Institution
Acceptance by Sending Institution
Registration/ Acceptance by Receiving Institution
Completion at Receiving Institution
Completion at Sending Institution

Figure 1: Steps in the exchange process

Within this process, the exchange of data (Learning Agreement, Transcript of Records, etc.) currently takes place mainly through emails between the exchange officers working at the two universities. This data are mostly in the form of scans of paper document and must therefore be entered manually in different systems of the receiving party. Each party trust the correctness of this information because they trust the exchange officer from the other institution. In addition, determining the suitability of students for a particular exchange location is a manual process. See, Figure 2.

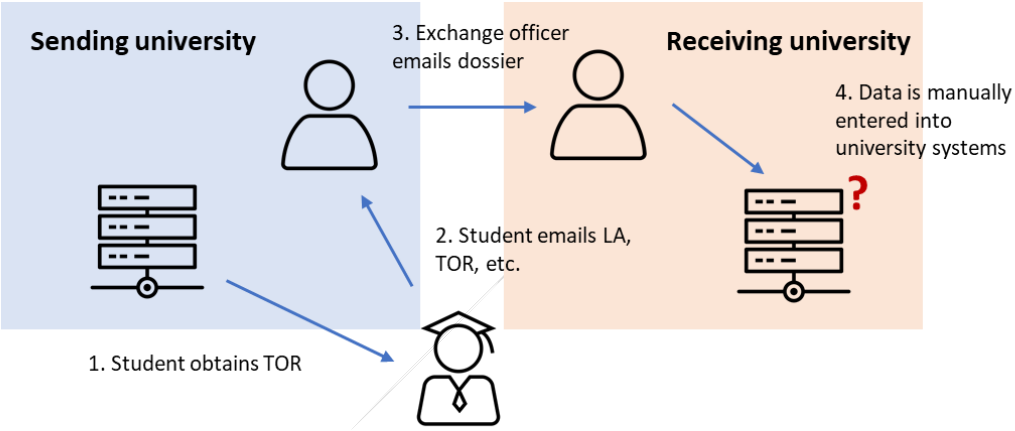

Sending university
3. Exchange officer emails dossier
Receiving university
4. Data is manually entered into university systems
?
2. Student emails LA, TOR, etc.
1. Student obtains TOR

Figure 2: Manual information exchange

The important difference with the desired situation is that the exchange of data mainly takes place electronically, and where possible the decision making is automated. This is possible because clear rules have been laid down in the Erasmus+ Exchange program concerning the information requirement for different decision moments [Erasmus + Program Guide]. This means that data can be shared directly by the student, and the correctness of this information can be verified electronically. See Figure 3.

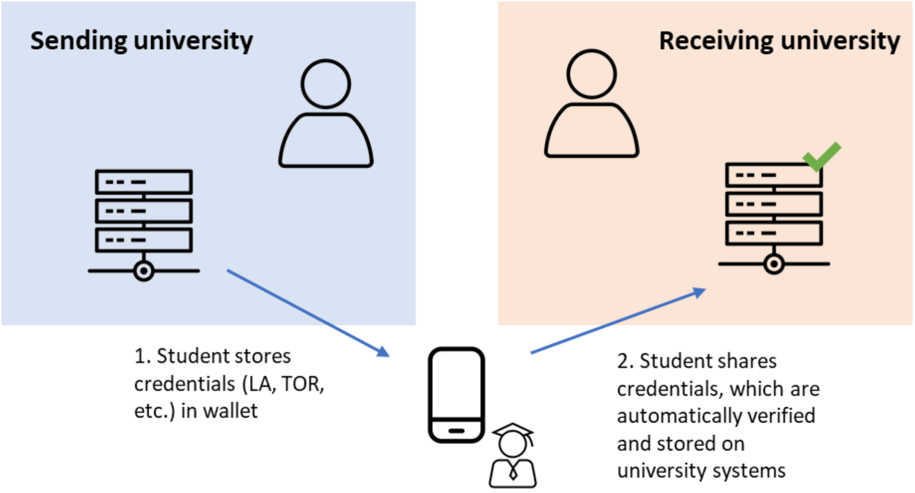

Sending university
Receiving university
1. Student stores credentials (LA, TOR, etc.) in wallet
2. Student shares credentials, which are automatically verified and stored on university systems

Figure 3: Electronic data exchange

6 Requirements

In this chapter we mention the most important requirements that we believe should be satisfied by the Studybits solution. We note that this solution consists of an infrastructure, and that a complete solution, which after all must also offer process-supporting functionality, will also have different (extra) requirements. We also assume that the 'normal' security requirements apply, such as communication over networks is encrypted (SSL, TLS), applications are neatly designed (no vulnerabilities on the OWASP list), and so on.

Our requirements are about providing, storing and using data and making decisions based on reasoning with those data. We do not set explicit privacy requirements in this document; we consider these to be properties of the chosen technologies, and will to investigate these properties in the relevant chapters.

The Studybits exchange system is a distributed system, components of which are organized, managed and used by (or on behalf of) the participating educational institutions. The purpose of the electronic component(s) of an institution is to support the exchange process as far as it is the responsibility of that institution. An institution can have various responsibilities, such as making providing exchange spot for a visiting student, or facilitating their own student who wants to take a part of his study elsewhere.

The purpose of each component of this exchange system is that it helps the educational institution to which it belongs, to make decisions that are necessary for the institution to address their responsibility, and to make the data necessary for that decision available in an efficient manner.

A decision is the result of some reasoning about the data, and is taken by or on behalf of a party. It is up to that party themselves to determine which reasoning they consider valid – this is a subjective matter. For example, the reasoning whether a student meets the qualifications for a certain exchange spot at a certain university will be determined by that university. And a student themselves – with their own logic – determines whether they accept the offer to make use of an available exchange place or not. In order to be able to make the components that can support the decision making of different parties, it is therefore necessary to know which parties are participating, and what kind of decisions they have to take.

Human actors reason differently than computers. Because of their human characteristics (flexibility, being able to make mistakes and recover from them, reflecting, etc.) it is not always necessary to be very explicit or precise about the type of data that is needed or the logic that they have to follow. Humans are often able to work it out. That's why it's good to know:

- which decision(s) of which parties are to be taken by people,
- what line of reasoning is to be followed, and
- what kind of data (sets) are needed for this.

For different decisions regarding Erasmus+ exchanges this is specified for by the [Erasmus+ Program Guide].

This is not sufficient for letting computers make decisions: they have to be told very precisely what information they need and how to correlate this information in order to arrive at the required decision. Here it is necessary to be *very precise* in the type of data that is needed and the reasoning steps must also be very precisely defined.

Moreover, in such cases it is necessary to check whether reasoning that is held according to the specified logic is actually valid. That is why it is necessary that:

- the logic that must be followed by the machine is considered valid by the party on whose behalf the machine is reasoning. Reasoning is, after all, subjective.
- the type of data required for this reasoning is determined by that same party. The party must determine
    - what kind of data is needed for the reasoning;
    - that this data (to a sufficient degree) means what the party thinks it means, and
    - how to determined that received data is really truthful.

  After all, reasoning with data that has an incorrect meaning, as well as a reasoning of with data that is wrong or not truthful, is not a valid reasoning and therefore does not always lead to a correct decision.

Only when all participating parties have divided the work (and decisions) between man and machine, and have sufficiently specified the 'business logic' and the corresponding information (and data) requirement (at least for the machines), the exchange system can function properly.

Thus, the requirements for the exchange system can be summarized as follows:

1. There is a list of decisions to be taken in the exchange process; the name of a role is linked to each decision; we use this name to refer to actors who make such decisions. Example: the decision whether or not to honor a student's application for an exchange place to the institution is taken by the Evaluator. Each institution can then assign this role to people or machines that may take this decision on behalf of the institution.
2. Each participating institution selects the decisions from this list that they can and want to take. For each of these decisions
    a. they determine whether or not this must be taken manually or automatically.
    b. in case it could / should be taken automatically
        i. according to what logic this decision must be made;
        ii. what kind of data is needed;
        iii. how to determine the correct meaning (of instances) of these data (semantics);
        iv. how the truthfulness (of instances) of this data should be determined.

Note that points iii and iv both imply decisions, whereby 'decision recursion' occurs. We want to terminate this recursion as soon as possible. For iii this can be done by referring to one or more semantic predicates from schemas for every bit of data. Although it is not necessary, it is useful (and conceivable) that (over time) a consensus can arise between participants – this could then be codified in a semantic 'standard'.

For iv, a party can stop the recursion by determining which organizations they trust to issue valid data. This kind of 'policies' are also subjective: each party will have to decide for themselves. Such lists, especially if they are not too complex, can easily be used by computers to determine whether received data is 'truthful' and is therefore valid for basing decisions on.

In order to facilitate other participants in determining their data needs for certain decisions, a party may publish the types of data for which it is able and willing to issue attestations. This enables such participants to be certain that if they need data from that party (which they must necessarily trust – otherwise it does not make sense), it can actually be obtained from that party. If each party does this, an information ecosystem will develop over time, in which the exchange of students (and many other processes) will become increasingly efficient.

7 Investigated technologies

In the first phase of the Studybits project it was decided to look at the suitability of two blockchain related technologies for use in the Proof of Concept: Blockcerts and Sovrin to be developed. It has been decided to focus on these two projects for the following reasons:

- The feasibility study of the Studybits SBIR project should be completed within a limited time frame. As a result, there is no time to examine the many existing blockchain related digital identity projects in-depth.
- Blockcerts and Open Badges have been developed for sharing academic qualifications and are already awarded to students by various (educational) institutions. This makes it a logical choice for application in the Erasmus+ exchange project, where academic qualifications and study results are also shared.
- Sovrin offers a complete digital identity solution with many privacy-friendly features. The student data that is shared during the exchange process also has value for the student outside of this process. A complete solution for digital identity has added value compared to a specific application.
- The principles of Blockcerts and Sovrin are very similar to those of Self-Sovereign Identity. See for example the following comparison with respect to other (blockchain) identity solutions [The Known Traveler, p21]:

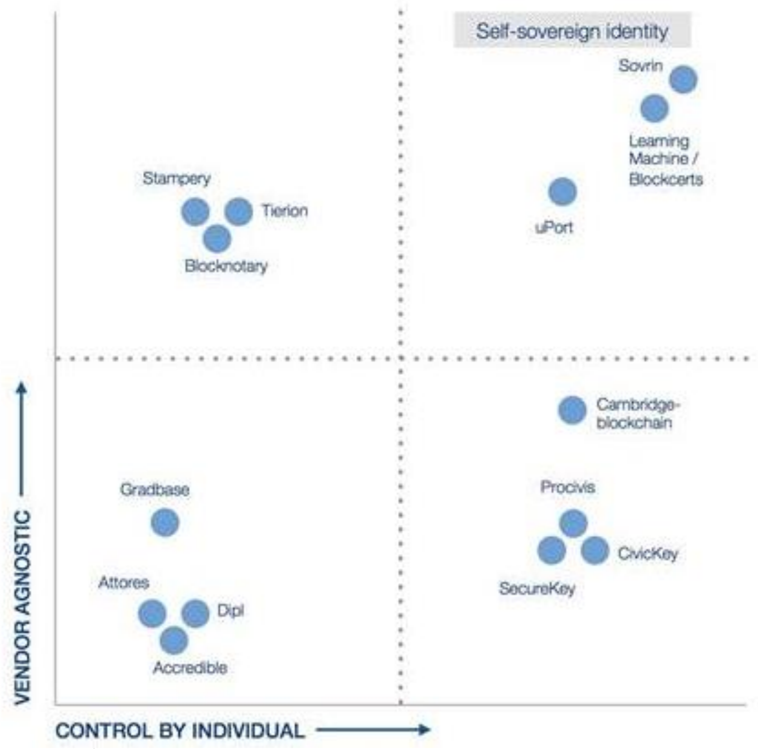


Figure 4: Non-exhaustive overview of Digital Identity Initiatives

7.1 Blockcerts

The Blockcerts project originated from the Media Lab of the Massachusetts Institute of Technology and the American company Learning Machine. The technology behind Blockcerts can be used to issue and validate digital certificates of, for example, study results, diplomas, but also professional and personal data. The digital certificates used in Blockcerts are based on version 2 of the Open Badges standard developed by the Mozilla Foundation. Blockcerts expands the capabilities of Open Badges with distributed ledger technology.

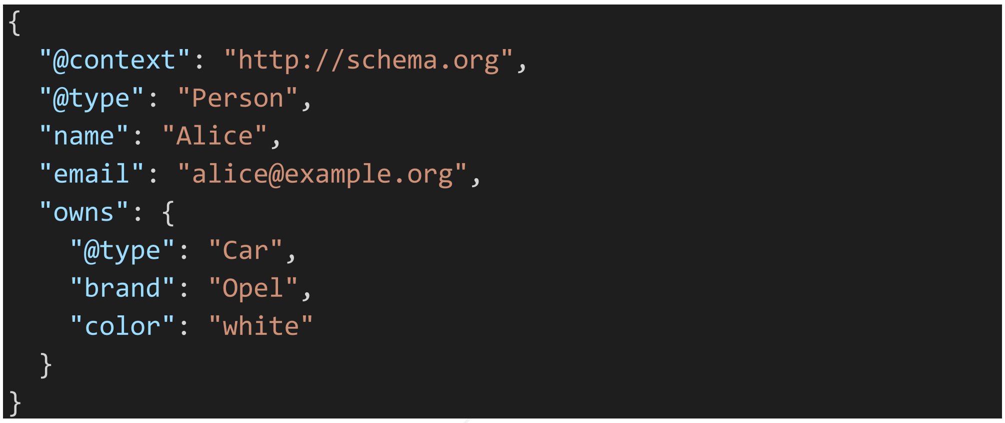
7.1.1 Semantics

Open Badges, and therefore also Blockcerts, is a semantic standard. It is prescribed what information should be present in each certificate and how this information should be represented. The Open Badges standard builds on JSON-LD. This is a W3C Recommendation for presenting RDF data in the form of JSON, a widely used data format.

Each JSON-LD document consists of a graph of semantic data, in which, besides attributes of different objects from multiple contexts, the relations between these objects can also be described. Such a document is semantic in the sense that it is unambiguously recorded what the *meaning* is of all available data. All this is done in a way that is readable for both people and machines.

Example

The following JSON-LD document contains information about a person, Alice, and about a car. Various attributes are given for both the car and Alice, namely the email address of Alice, and the color and brand of the car. In addition, there is a relationship between Alice and the car: Alice is the owner of this vehicle.



```
{
  "@context": "http://schema.org",
  "@type": "Person",
  "name": "Alice",
  "email": "alice@example.org",
  "owns": {
    "@type": "Car",
    "brand": "Opel",
    "color": "white"
  }
}
```

Example JSON-LD Document

Bovenstaande JSON-LD document vertaalt naar de onderstaande RDF statements. Een RDF statement is een triple van de vorm (subject, predicaat, object). De exacte betekenis van elk predicaat is vastgelegd. Hier zijn *_alice* en *_car* willekeurig gekozen identifiers om de twee objecten in het JSON-LD document aan te duiden.

The above JSON-LD document translates to the following RDF statements. An RDF statement is a triple of the form (subject, predicate, object). The exact meaning of each predicate has been established. Here, *_alice* and *_car* are arbitrarily chosen identifiers to indicate the two objects in the JSON-LD document.

Subject	Predicate	Object
_alice	http://schema.org/email	alice@example.org
_alice	http://schema.org/name	Alice
_alice	http://schema.org/owns	_car
_alice	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Person
_car	http://schema.org/brand	Opel
_car	http://schema.org/color	white
_car	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Car

7.1.2 *Verifiable claims*

Each Open Badge certificate is a JSON-LD document containing statements about different objects. Each certificate contains statements about:

- The subject; the person to whom the certificate has been issued.
- The issuer; the organization that issued the certificate.
- Meaning of the certificate itself and criteria for obtaining the certificate.
- The way in which the validity of the certificate can be checked.

There are several ways in which the validity of the certificate can be checked:

- the certificate is digitally signed (and thus a verifiable claim). The issuer / signer hereby declares that the content (according to them) is correct and published by them. The properties of digital signatures ensure that the content of the certificate cannot be changed. Open Badges calls this signed badges.
- the certificate can be obtained at a fixed location (URL). In this case, the user must be confident that the manager of that location has not compromised the integrity of the badge. Open Badges calls this hosted badges.
- Blockcerts adds a new method to this, by registering the (secure) hash of the badge on a blockchain. A user checks the integrity of the certificate by checking whether the hash is indeed registered there, and its validity by checking whether it finds the party (issuer) that has done this registration sufficiently reliable.

There is room in the badge itself to indicate where information about a possible revocation can be found (for example via a certificate revocation list (CRL), or via the online certificate status protocol (OCSP)).

7.1.3 Identity

A Blockcert certificate is linked to the subject (recipient) in two ways. The certificate contains both a reference to the public key and the email address of the person to whom the certificate was issued. When a subject sends a certificate to a third party, the third party can determine that the certificate actually belongs to that person by sending a verification email to the inserted e-mail address and / or by going through a challenge-response in which the user demonstrates they possess the private key associated with the public key included in the certificate. This key material is stored in the wallet of the recipient. This is a (mobile) application in which all the certificates of the recipient are stored in addition to their key material. The wallet is used to receive new certificates and to share existing certificates with third parties.

7.1.4 Privacy

Each Blockcert certificate is unique and contains both the email address and the public key of the subject of the certificate. As a result, someone who shows his Blockcert certificates is easy to follow by parties who compare their received certificates. It is also trivial to link different certificates that an organization has received from the same person on multiple occasions.

The issuer of Blockcert certificates can also follow the recipient, for example by assigning a unique revocation list to each certificate. This enables him to keep track of how often (and perhaps even with which parties) the certificate is shared.

A publisher does not even have to go that far, because the standard libraries for validating a Blockcert certificate give the publisher a notification each time a certificate issued by them is shared [Blockchain Critique].

7.1.5 Scalability

Blockcerts uses existing general purpose blockchains to store the hash of each certificate. These transactions can be accompanied by (high!) costs, for example if they are stored on the Bitcoin blockchain, or on Ethereum. Because Blockcerts makes use of public permissionless blockchains by default, there is usually no party that can intervene if the transaction price becomes excessively high – issuers of Blockcert certificates are then completely dependent on the market.

A recent development in the area of scalability is supporting Merkle Tree-proofs in Blockcerts. This allows the hashes of an (unlimited) number of certificates to be included in one blockchain transaction, thus reducing the costs per certificate.

7.1.6 Blockcerts and SSI

For SSI it is necessary that all kinds of statements about an individual, of different kinds and meanings, can be included in a Blockcert, and that a party that uses these statements can their meaning, and can decide whether they are sufficient truthful/valid for their purpose.

The Blockcert technology can meet these requirements. Whether a specific Blockcert certificate is usable depends only on the statements contained within it, and the extent to which those matches match what a user needs.

7.2 Sovrin

Sovrin is a decentralized, public permissioned (blockchain) network for supporting digital identity. The project was originally developed by the American company Evernym. The project code was later transferred to the Hyperledger Foundation, an open source partnership for the development and improvement of various distributed ledger technology related projects, as an open source project. It is currently available under the name 'Hyperledger Indy'.

7.2.1 Semantics

Sovrin works with collections of claims (credentials) according to the W3C standard that is currently being developed for verifiable claims. A credential can contain any conceivable collection of attributes. Before a issuer can issue a Sovrin credential with a certain set of attributes, its structure must first be specified in a schema which is stored on the Sovrin ledger [How Sovrin Works]. Each issuer can then use this schema by referring to it in the credentials that it issues – thereby registering the meaning of the claims and the way in which they sign these claims. Each consumer of such credentials can establish their integrity and the meaning of the attributes/statements contained therein. This stimulates (semantic) interoperability between different parties.

7.2.2 Identity

A Sovrin credential is linked to a 'master secret' – a secret code that is stored in the Sovrin wallet. The idea is that the wallet can only be used by one person – Sovrin assumes that this is the case. The idea is that credentials linked to the same master secret are issued to the same person.

The wallet can make its own identities – so-called DIDs – that can be used by relying parties to identify the wallet. In fact, they can set up a secure 2-way channel with the wallet. Under the aforementioned assumption, that is a secure 2-way channel between the individual and the relying party.

Sovrin provides a mechanism [What Goes On The Ledger] for the safe registration of such 2-way channels, where both parties can update the data relevant for maintaining that channel, such as their own key material.

7.2.3 *Privacy*

Credentials in Sovrin build on Idemix (Identity Mixer) principles, as developed by IBM Research. Idemix is a cryptographic system for issuing and validating anonymous credentials. This technology offers various possibilities with very privacy-friendly features. The Idemix technology has previously been successfully applied in the IRMA project, developed by Radboud University in collaboration with, among others, TNO.

Credentials in Sovrin have very good privacy properties and support, among other things, selective disclosure, zero-knowledge proofs and unlinkability. In selective disclosure, the user may choose to disclose only part of all attributes in a claim to a third party. With the aid of zero-knowledge proofs, a user can prove the truth of a statement, without the third party obtaining extra information (zero-knowledge); this way a user can prove that they are over 18 years old without revealing the exact date of their birth (which is in the credential).

There are two types of unlinkability of attributes:

- a relying party can only determine that certain attributes belong to one and the same person if they are obtained in a single session (session unlinkability);
- two different relying parties cannot determine whether they had to deal with a single person in their own sessions with that person (RP unlinkability).

The use of DIDs in 2-way channels means that the parties that maintain this channel can identify and authenticate each other, while the identifiers used (the DIDs) do not have an identifying value for other parties. This provides a strong RP unlinkability.

A possible point of attention is that using such a 2-way channel, a relying party can link attributes that are provided to him in different sessions to each other, because they were sent through the same channel. Hence, there is no session unlinkability if these persistent 2-way channels are used.

7.2.4 *Scalability*

The Sovrin blockchain is a permission ledger that is only used for Sovrin transactions, without any transaction costs. It is specifically designed to be used for this purpose, with high availability and low latency when used on a worldwide scale.

No blockchain transactions are required for the issuance of new credentials, but revocations (which are less frequent) do require transactions.

7.2.5 *Sovrin and SSI*

For SSI it is necessary that all sorts of statements about an individual, of different kind and significance, can be included in a credential, and that a party using these statements can know what their meaning is, and can decide whether they are sufficiently truthful for their purpose.

As with Blockcerts, Sovrin's technology can meet these requirements. Again, the suitability of a given credential for use in a decision process of a certain party completely depends on the statements contained within in.

8 Comparison of technologies

8.1 Comparison

In this chapter we compare the two chosen technologies on the basis of the requirements drawn up in chapter 6 and their technical possibilities. The main differences between the technologies are listed in the following table:

	Blockcerts	Sovrin
Purpose	Issue Open Badges certificates and validate them with blockchain.	Self-Sovereign Identity platform based on blockchain and Attribute Based Credentials.
Type of blockchain/ledger	'Public permissionless' general-purpose blockchains like Bitcoin and Ethereum	'Public permissioned' Sovrin ledger, for digital identity
Governance blockchain/ledger	Not applicable	• Network managed by Sovrin Foundation • Nodes managed by 'Stewards': organisations trusted by network
Usage of blockchain/ledger	• Register the issuance of certificates by storing hashes in blockchain transactions	• Store DIDs and related data for PKI • Store schemas and corresponding claim definitions • Store public claims • Store consent receipts • Store revocations
Claims transport (type of messages)	JSON-LD document (Open Badge)	Anonymous credentials using ideas from Idemix
Claim storage	In wallet, back-ups not in scope.	Public claims on ledger, private claims in wallet. Back-ups not in scope.
Privacy properties	• No plain text or encrypted personal information on blockchain • Unique certificates makes tracking possible • No session unlinkability	• No plain text or encrypted personal information on blockchain • Only end-user info stored on blockchain are pseudonymous

	• No RP unlinkability • Default libraries notify issuer when certificates are shared	DIDs. • Unlinkability using Attribute based credentials, but some linking through use of persistent channels possible, resulting in strong RP unlinkability, with (reduced) session unlinkability.
Scalability	• Uses (public, unpermissioned) general-purpose blockchains. • Transaction costs and time dependent on chosen blockchain. • Possible to issue multiple certificates in a single transaction.	• Uses dedicated (public, permissioned) Sovrin blockchain • No transaction costs for usage of blockchain. • Claims can be issued without modifying the ledger.
Semantics and interoperability	• Uses JSON-LD and related semantic technologies. • Existing standard about what goes/can go into a certificate	• No build-in semantic standard • For every kind of credential, a schema can/needs to be stored on the ledger • Schemas are reusable by multiple parties, improving interoperability.
Revocation	Stored in a online list maintained by issuer	Stored on blockchain by issuer
Mutual party identification	Out of scope, parties need to find out their own way of determining which keys are used by whom	Using DIDs parties can authenticate and identify themselves over 2-way secure channels.
Linkage of credentials from different issuers	Certificates are linked to the email and public key (and name) of the recipient, correlation can occur on those attributes	Certificates are linked to a (blinded) master secret, different credentials can be cryptographically linked together by the recipient
Robustness against unexpected events	Problems occur when a recipient changes email or loses their private key (certificates need to be reissued)	With DIDs, loss/compromise of private keys can be mitigated. Loss/compromise of master secret leads to problems
Ease of implementation and support	Open source libraries available. Because of narrow scope implementation can be straightforward. For	Open source (low level) SDKs available. Because of broad scope implementation is more difficult. Buildingblocks for complete

	complete solution, extra infrastructure will need to be build. Support obtainable from commercial party (Learning Machine)	solution are there. Support by Sovrin Foundation or Evernym.

8.2 Advice

The above comparison shows that Sovrin offers many advantages over Blockcerts to serve as a basis for a solution for a complete self-sovereign digital identity. It has very strong privacy-friendly features and provides the infrastructure and protocols for the safe and reliable issuing and sharing of digital credentials. Since a reliable digital identity is such a useful thing to have for students (and of course other people), the Studybit Proof of Concept seems like an excellent opportunity to investigate if this can also be used to streamline existing processes at educational institutions.

9 Architecture

In this chapter we briefly describe what is required for the development of the Studybits Proof of Concept. We describe the architecture required for the data exchange between the student and the educational institution. It also describes which parts of the exchange process are within the scope of the Studybits Proof of Concept that will be used during the pilot. We then provide an overview of the various (software) components that have to be developed for PoC and which existing packages can be used for this purpose. Finally, we give a number of external factors that influence the success of the pilot and the application of the chosen technology (outside of this pilot) at universities and other institutions.

9.1 Interaction students and institutions

A (generic) interaction between a student and an educational institution always follows the same pattern; after all, a student wants something from the educational institution, and then the institution has to decide whether to go along with the wish of the student. To make this happen in the electronic domain, we postulate the existence of:

- an **attestation provider**, i.e. a component (connected to the backend systems of an institution) that can issue the attestations/credentials that the institution wants to be able to issue;
- an **info-shop**, i.e. a component on which an attestation provider informs the other participants what type of attestations are issued (and also what the semantics are, how the truth is established, etc.) for relying parties, and where (on which URL) they can be obtained by a recipient;
- a **relying party**, i.e. a component (connected to the backend systems of an institution) that can decide to grant or reject requests from students for a

certain service (such as being registered as an exchange student, reserving a place, etc.)[1]
- a **wallet**, i.e. a component (mobile app) that can collect attestations/credentials from attestation providers on behalf of an individual, and can use these attestations in obtaining a service from a (different) relying party.

An individual can obtain a service as follows. He browses to the relying party, and requests (to be admitted to) a certain service. The relying party now has to decide whether that is possible or not. The policy set by the relevant party specifies what type of data is required for this, and by which attestation provider(s) that data must have been issued (and that there is certainty that this provider also issues such attestations), and how exactly the machine must reason with this data to come to that decision. The relying party can thus send a list of requested data to the wallet, and identify the attestation providers that must have issues that data. The wallet searches in its own storage to see if it already has that data. Missing data can be retrieved 'on the fly' by the user-selected (and for the RP acceptable) attestation providers (if available). The wallet then sends the collected credentials to the RP, which checks whether they have been revoked and if that is not the case, reasons with them and decides to provide the service or not. This process is summarized in the following image:

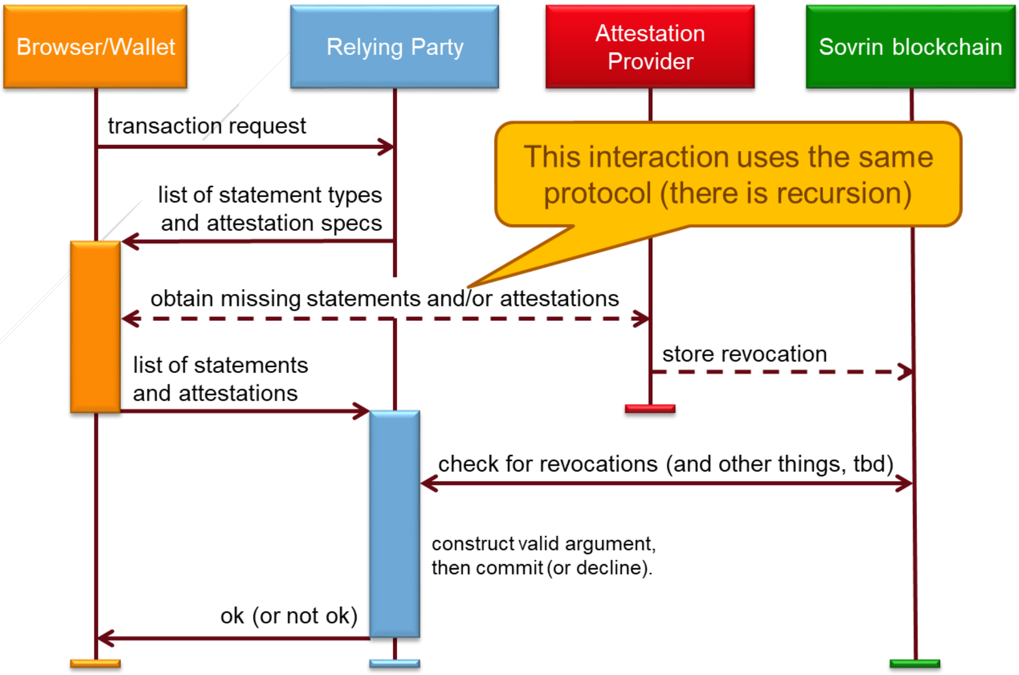

Browser/Wallet
Relying Party
Attestation Provider
Sovrin blockchain
transaction request
list of statement types and attestation specs
This interaction uses the same protocol (there is recursion)
obtain missing statements and/or attestations
store revocation
list of statements and attestations
check for revocations (and other things, tbd)
construct valid argument, then commit (or decline).
ok (or not ok)

Figure 5: Service/transaction request process

Within the Studybits Proof of Concept, the high level interaction structure between the different (human and machine) actors and systems (at a single university) is displayed in the following image. At the other (sending or receiving) institution, the exact same interaction structure is present, showing the symmetry of the process

[1] A attestation provider is thus a specific instance of a relying party

where both institutions act as relying party (when verifying credentials) and attestation provider (when issuing new credentials).

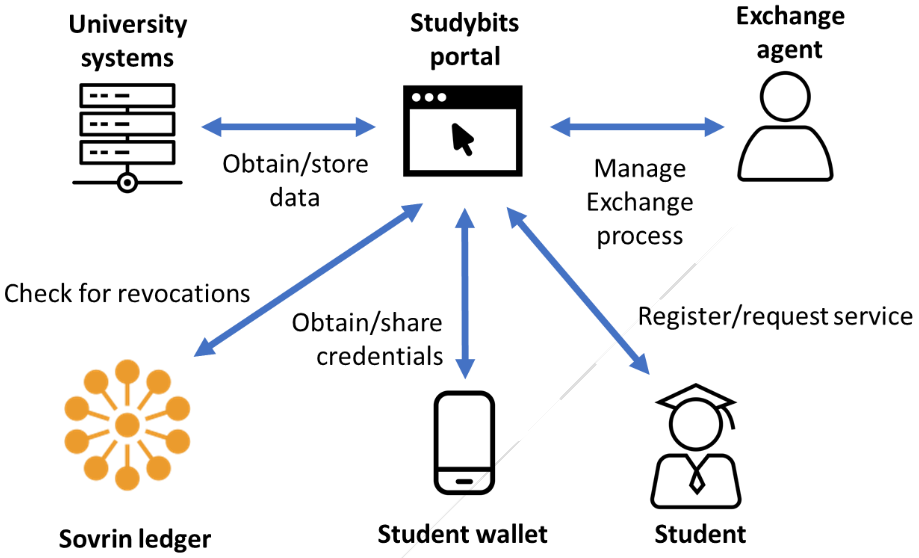


Figure 6: High level interaction structure

9.2 Proof of Concept

The Studybits Proof of Concept will support part of the process as described in the self-sovereign customer journey. The PoC will support:

1. Drafting and publishing a digital profile of requirements for each of the available exchange places at the participating universities.
2. The onboarding of the digital identity of the student by issuing a digital student card / registration certificate.
3. Applying for an exchange location (both at the sending and receiving institutions) with corresponding data exchange (Grant Agreement, Learning Agreement, Transcript of Records, results of the language test, etc.) and acceptance of the student by the sending and receiving university.
4. The completion of the exchange process, whereby the study results obtained by the student during the exchange are received and accepted by the sending university.

The following parts as described in the customer journey will not be part of the Proof of Concept:

1. Drafting a digital profile of requirements for the different subjects that can be followed during the exchange: this is too much work during a pilot and very complex compared to the profile of requirements for an exchange site.
2. Applying for and processing the subsidy application: this does not add much to the above-mentioned components in order to demonstrate the success of the pilot.

9.3 Available technological solutions

For the pilot, a separate Sovrin test network will be set up, where the nodes will be managed by different participants in the Studybits project. For this purpose, the open source Hyperledger Indy Node code is used with an adapted genesis transaction. As a result, only the nodes of the Studybits participants, which act as Stewards, can participate in this test network. The universities participating in the pilot will be included in the blockchain as Trust Anchor, allowing them to put new schemas, claim definitions and DIDs on the blockchain.

For the interaction with the/a Sovrin blockchain, Software Development Kits (SDK) are available for different programming languages. One of these SDKs will be used in a developing agent (electronic actor) that will be run at participating universities in order to receive student credentials and issue new credentials. These agents support the dashboards/applications that will be used by the exchange officers. These dashboards provide the exchange officers with insight into the status of the exchange process of a given student and which credentials of the student have been received and processed. Using this dashboard, exchange officers can also give permission to issue new credentials to the student (such as a signed Learning Agreement and Transcript of Records). Part of the issued data comes from existing student information systems at the universities (including the Transcript of Records and registration data). The agent will have to be able to access this information by some method, such as on the fly or via some import mechanism.

Finally, there is the wallet in which the students save their credentials and from which they can share these credentials. There is a Sovrin SDK for iOS and one for Java. These can be used to develop their own wallet for both iOS and Android devices. In addition, Evernym has its own wallet for use with the Sovrin blockchain. It is possible that they are willing to share the source code with the Studybits project.

To record the semantics of the exchanged data, different schemas will have to be specified and placed on the Sovrin blockchain. A digital schema will have to be drawn up for the following documents:

- Grant Agreement
- Learning Agreement
- Transcript of Records (or perhaps individual study results)
- Results of language test

In the EMREX project, an XML scheme called ELMO has already been created for sharing study results. Open Badges and Blockcerts also have different RDF predicates for the description of academic qualifications. Aimed at possible interoperability in the future, it may be a good idea to adopt one of these standards.

9.4 External factors (technical) feasibility

There are a number of external factors that influence the technical feasibility of the Studybits project. For example, it is necessary that a link is created between the (different) student information systems that are present at each university and the agent running at that institution. The possibility to extract digital information from these systems or to write to them is strongly dependent on the individual systems at the universities. Designing, developing and maintaining these links can also involve (high) costs. In addition, the required expertise for setting up or maintaining a Sovrin agent might not be present at a university.

The pilot will also show whether it is possible to ‘hide’ the advanced cryptography and complex concepts such as Trust Anchors, DIDs and anonymous credentials from the end-user or to present them in a user-friendly package (GUI), so that it is not necessary for the end-user to know about this type of technology: *it just works*.