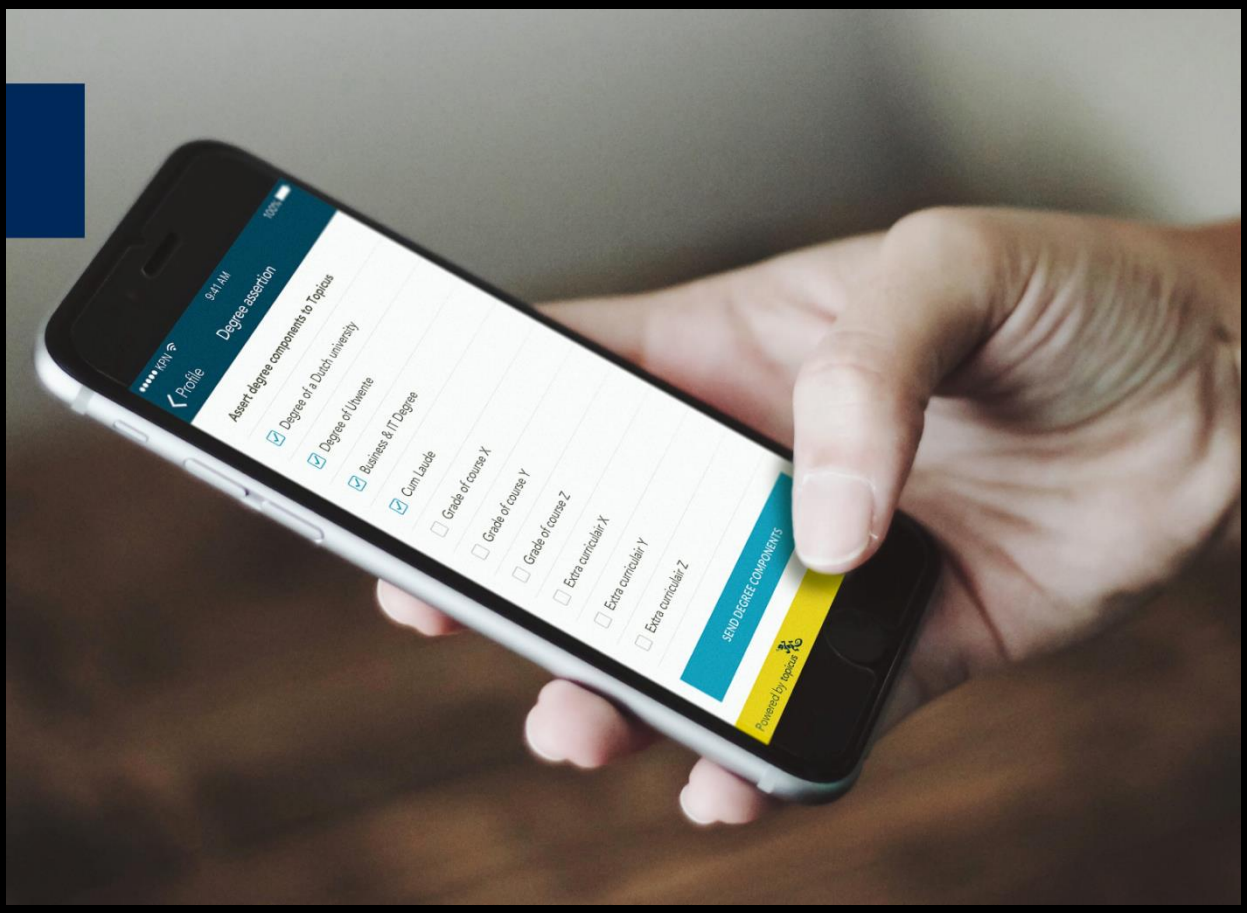


Decentralized credential publication and verification

A method for issuing and verifying academic degrees with smart contracts

Frank Brinkkemper
UNIVERSITY OF TWENTE Drienerlolaan 5, Enschede

1 Abstract

In 2011 it was estimated that more Ph.D.'s were illegally purchased (50,000) in the United States every year, than were actually given out (45,000) (John Bear, 2012). This troublesome statistic is an expensive problem for both higher educational institutes, and employers, as the verification of these credentials is currently a manual process and therefore inefficient and time-consuming. Although a paper degree is often a beautifully crafted memory of one's education, it should not be used as a formal verification method anymore.

In this design research we propose a decentralized system for digital degree issuing and verification. We start by studying the currently available methods for degree issuance and verification in the Netherlands. These methods are checked against the user needs in a universal system for verifiable claims, which constitute the broader use case of verified credentials. The current issue and verification systems fail to comply with all the user needs, as the verification relies heavily on a centralized party, and the data ownership is not transferred to the recipient of the degree.

In order to tackle these problems in the form of a new design, we focus on a decentralized solution by utilizing blockchain and smart contract technology. Several blockchain based credential issuing methods have been created thus far, of which Blockcerts is the most notable (Blockcerts, 2018a). However, these current solutions fail to comply with all user needs. In the proposed design, a combination of the Blockcerts standard and an Ethereum claims registry make it possible to comply to the defined user needs (Joel Torstensson, 2017).

In our design proposal we use the public Ethereum blockchain to our advantage for high data availability, the possibility of verification without the explicit cooperation of the original issuer, and the smart contract functionality. Even in the case an issuer goes out of business, the public blockchain continues to maintain an incentive to host your proof data.

The proposal has been validated through interviews with multiple experts from both higher educational, and the blockchain field. Presentations for the ICT cooperation for education and research (SURF), and distributed ledger experts from the Ethereum Foundation (Nick Johnson), Rabobank, and TNO provided valuable feedback on the design.

Finally, we show how these concepts are now applied at the biggest Dutch mortgage software creator in order to reduce manual document verification for mortgage applications. A proof of concept has been built for issuing a verifiable employer's statement to employees. The employer statement is currently refused 90 percent of the time on the first assertion by mortgage suppliers (Olivier Tardieu, 2017).

Although there are still some challenges, like (1) identification and identity management of the issuer and receiver, (2) complying fully with the right to be forgotten, the design offers enough ground for further research. The most important aspect to tackle next is to improve the end-user usability, and integration with existing issuing software to make world-wide adoption possible.

1 Acknowledgements

This thesis was created as final assignment for the master Business & IT at the University of Twente. I would like to extend a lot of gratitude to my supervisors Jos van Hillegersberg and Jaco van de Pol for their patience, and helping me shape this research to this final product. Their insights, and contacts were of tremendous value for this thesis.

The Dutch fintech company Topicus.Finance gave me the opportunity to perform my research at their company. A very special thanks to Maarten Schopman and Michiel Schipper for the opportunity, the many hours of feedback despite your busy schedules, and the critical mind-set throughout this project.

I would also like to thank all the people I have interviewed and/or presented to for this thesis. A special thank you to: Nick Johnson from the Ethereum foundation, Maarten Everts from TNO/University of Twente, Pascal van Eck from Ethereum DEV NL, Kim Hamilton Duffy from Blockcerts/Learning Machine, Djuri Baars and Jarl Nieuwenhuizen from the Rabobank, and Frans Ward and Alexander Blanc from Surf.

Lastly, I would like to deeply thank my friends and family for pushing me through this journey. Marijke, Sjaak, Lineke, Martin, your kind and helpful words were a major motivation. Most importantly, the loving force of Kim is probably the only reason you are able to read this thesis right now. Thank you for motivating me throughout this process, and especially in the last few months in our home.

Enjoy life, be happy.

Frank.

Contents

1 Introduction

After years of studying and overall hard work, students are rewarded with a proof of their academic achievement in the form of a paper degree. This document, together with a method to verify it, allows them to enter the job market for the higher educated.

The digital transformation of the Western world is in a far stage and the job application market is not left untouched. Online job vacancies, and aggregators thereof, make it easier for the jobless to find appropriate work. Finding candidates for a job offer is also easier than ever with professional social networks like LinkedIn. Thanks to these technological advancements, physical distance between employer and potential employee has become less important.

The distance factor is even less important in the decision for a higher educational institute. In the Netherlands, universities are actively marketing towards international students. This approach is working as the number of international students in the Netherlands has steadily doubled to over 80 thousand since 2006 (Huberts, 2016).

1.1 Degree verification

These students provide increased financial security for universities overall. However, the application offices of the universities need to be able to handle all the proofs of intellectual achievement. The verification of the previous degrees is an enormous process. For some universities this amounts to thousands of applicants from all over the world, each with a custom degree verification method.

In 2016 over 76,000 people in the Netherlands achieved an academic degree, both bachelor and master (DUO, 2017a). Verifying the legitimacy of the degrees is an important task in both the case of a job application with a higher educational requirement, and for the application to a universities master's programme. According a study by Kroll, a New York City risk consultancy firm, 22 percent of the resumes the firm verified in 2007 for technology companies contained misrepresentations of academic credentials. This percentage is since not expected to drop (Patel, 2009). A study in 2009 by the American Automatic Data Processing HR firm, about data gathered in the year prior, found that upwards of 46% of employment, education, and/or reference checks turned up discrepancies (ADP, 2009).

The current methods for degree verification rely on the availability of the issuing party, or a trusted centralised party. These also require some manual effort by both the person looking to verify a degree and the original issuer.

1.2 Blockchain and smart contract technologies

In recent years blockchain technology achieved mainstream attention, mostly through Bitcoin. However, more applications of the technology are built and experimented with on a daily basis. Blockchain offers a decentralized alternative to siloed centralized databases. This is achieved by having many nodes, in the case of Bitcoin several thousand, to keep a synchronized record of all historic transactions. These nodes monitor the incoming transactions, and verify the validity before finalizing them in a new block.

The first *killer-app* of blockchain technology is this possibility to create a new monetary system in the form of cryptocurrency without needing a centralized trusted party, e.g. a bank, to keep a record of transactions. Now, many people across the world are trying to find the next *killer-app*. Since 2015 this has become significantly easier, as Ethereum launched a platform for anyone to create their decentralized applications (DApps) by publishing a piece of code on their blockchain (Buterin, 2014).

1.3 Problem statement

Verifying degrees is an important process to combat fraud, and build trust in the applicant. Yet, the verification of degrees has not kept up with the advancements in technology, and is often a cumbersome manual process. Consequently, many organisations skip the verification, and trust the applicant on their word, creating a breeding ground for fraud.

It should be trivial for any organisation to verify the claimed degree of an applicant. As long as it is not trivial to do so, there are inefficiencies in the job-, and higher educational-application process. Below, some of the main problems in these cases are highlighted.

First, from the perspective of the company offering a job. Companies often receive more applications than there are job offerings available. Sifting through all these applications is a tedious process. In some of the supplied documents for the applications there might be omissions, exaggerations or even flat out lies. Doing a simple degree verification upfront makes sure that no unnecessary time is wasted on applicants who do not meet the correct educational requirement.

Universities have a similar problem. The admission offices of universities get flooded with applications. In popular migration countries like Canada, the United States of America, and the Netherlands, people are abusing university applications in order to falsely receive a student visa (Merola, 2016; Pinxteren, 2004). The task of the admission office is to filter the legitimate students from the unqualified and frauds. Fast verification of the degrees could greatly improve the efficiency and thoroughness of this process.

It is also to the best interest of alumni, degree holders, to support fast verification of degrees. This inefficiency in the job application process causes longer waiting times for the applicants. Besides, the requirement of a physical copy of the degree is redundant in this digital age. So, a solution towards automatic verification will be beneficial for the issuer, holder and verifier of the degree.

The main hindrance towards automatic verification is the absence of an electronical equivalent of an educational degree. The transformation from a paper degree to an electronic one might seem straightforward. However, transferring the authenticity features of a degree, i.e. a signature, is difficult to do without opening up the possibility to create fake degrees.

Some efforts have been made to create a solution for this problem using blockchain technology. Several projects created tools to notarize the proof of credentials like an academic degree on the Bitcoin blockchain (Benjamin Boeser, 2017; Blockcerts, 2018a; Manuel Araoz, 2018; UNIC, 2017). However, these systems do not abide all the user needs for all stakeholders in this case.

At the moment of writing this research there aren't any widely adopted projects that use Ethereum or another smart contract platform to tackle the aforementioned problems. In this research these technologies are studied, and used to design a decentralized degree issuance and verification system.

1.4 Research questions

The current solutions for decentralized digital degree issuance and verification do not provide an answer to all the problems they were set to solve. This research aims to create a workable standard for all stakeholders to trustworthily issue and verify electronical degrees for higher education.

The main research question is:

How can we design a decentralised degree issue & verification method in order to combat fraud and increase efficiency in the job application process, for both the employee and employer?

To reach the goal of this design study the following research questions, and sub-questions, have been drawn up:

1. What are the benefits and limitations of current verification methods as a universal solution to digital verifiable degrees?
    a. Why are digital verifiable degrees needed in the job application process?
    b. What is the current method to verify educational degrees?
    c. What are current methods to create verifiable certificates for MOOC's, workshops, and job experience?

   d. Why are the current methods not sufficient?
2. What are the benefits and limitations of blockchain and smart contract technology as a method for verifiable degrees?
   a. What is the state of the art of blockchain technology?
   b. What is the state of the art of smart contract technology?
   c. What are the characteristics of these technologies that make it a contender for solving the problems identified?
3. How can the current efforts be improved with smart contract technology to a functioning electronic degree verification method?
   a. What are the current efforts of degree issuance on blockchain technology?
   b. Why are these methods lacking as a functional verifiable degree?
4. How is the design of the degree publication and verification method structured in order to fulfil the identified user needs?
   a. How does the method satisfy the user needs?
   b. What are the steps for the degree issuing, attesting and verification processes?
5. Is the created method a valid solution to the identified problems?
   a. What are the possible attack vectors, and how can they be mitigated?
   b. Is the method valuable in other document verification areas?
   c. Is there an alternative design without blockchain?

The scope of this research focusses on Dutch higher educational degrees, and their verification methods. However, as this is a world-wide problem, it is applicable for the entire higher educational industry. Although the data is gathered in the Netherlands, and the validation is done with mostly Dutch stakeholders, the issuing and verification processes are similar in the rest of the world. In the validation chapter it is discussed how well the design can be used in similar problem areas in other contexts.

1.5 Research method

"If I have seen further than others, it is by standing upon the shoulders of giants.", Sir Isaac Newton.

Before answering the research questions, we first discuss the method used to find answers to the research questions. The purpose of design science research is to find innovative, technology-based, solutions for relevant business problems. Hevner et. al. popularised design science research methodology for information systems (Hevner, March, Park, Ram, & Ram, 2004). Their essay introduced an information systems research framework, which has been adjusted to a three cycle view (Hevner, 2007). In Figure 1 this three cycle view is applied to this research. As the main research question is a design question, this widely accepted research method is a natural fit.

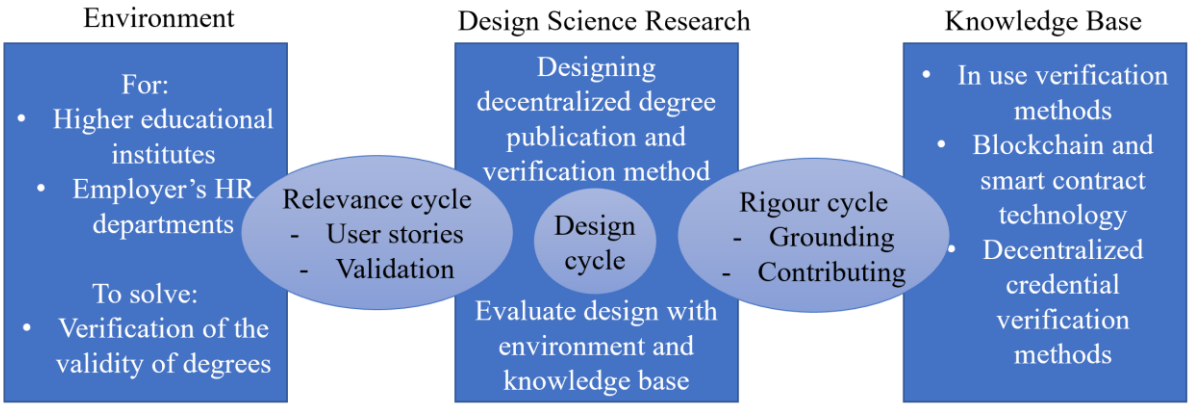


Figure 1: Application of the design science research in information systems to this thesis. (Hevner, 2007)

These three cycles describe the various research processes that form together a qualitative design science research study. The relevance cycle is carried out in chapters 2, in which the most important user stories are obtained, and 6 where the design is validated. The results of the design cycle are discussed in chapter 5. In chapters 3 and 4 the research is grounded to the current relevant knowledge base.

This design science model has been applied in this research to form the following research model shown in Figure 2. This research is performed by first acquiring knowledge through desk research. This consisted of (1) reading through the latest material on blockchain technology, smart contracts, and decentralized verification of credentials (2) studying the case, interviewing universities about the current verification processes. Then the current understanding was combined into a design.

This design is then validated with experts, where the design is presented, and feedback noted. With the useful feedback under the sleeve, open questions were researched, and used for a new iteration of the design. After a few iterations, the design for the verificfation method started to form, and validation interviews were held.

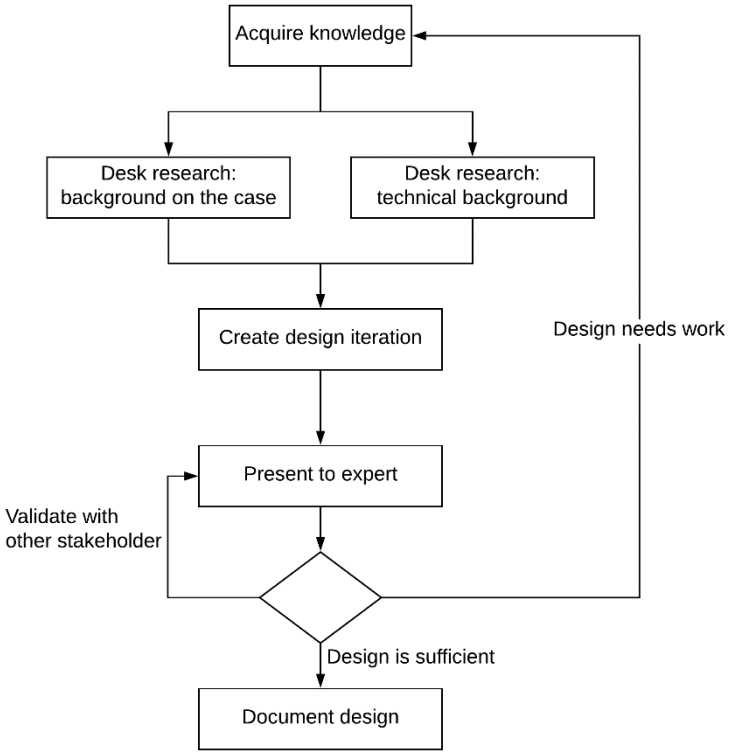


Figure 2: Research model

The feedback from all those final sources has led to the here presented design. These interviews are carried out by first asking some general questions, then presenting the method, and finally requesting feedback. The full transcript of the interview with Nick Johnson is provided in appendix A. The most important remarks by the other experts can be found in the validation chapter.

1.5.1 Risks of the research methodology

Creating a system for something as universal as academic degree verification is not something that can be done left alone. Therefore, this research has presented the final design to many experts, with at least one from each relevant stakeholder, within the limited timeframe. However, there are still a few risks that have to be taken into account with this research methodology.

- Most of the experts have an interest in blockchain technology. Therefore these interviews were all quite positive on the developed method. If multiple traditional institutions without blockchain experience would get together and find a solution to the problem, they might end up with a different design.
- During the interviews, the method was always presented in slide format, with a short live demo. However, the interviewees did not get the opportunity to use the method themselves to issue degrees. Therefore, this research does not make strong claims on the end-user usability of the method.

While these risks might have influenced the result of the study, the resulting design is still validated to be a proper system for degree publication and verification. In order to reach universal acceptability of the method, this research has focussed on combining existing worldwide standards.

2 Background on the case

In this section the problem domain is delved into. The respective domain is the degree verification process for job applications and master's degree applications. Furthermore, the generalised problem that this case belongs to is discussed. The goal of this part is to answer the first research question:

1. *What are the benefits and limitations of current verification methods as a universal solution to digital verifiable degrees?*

In order to find an answer to this research question, the following sub questions are explored:

a. Why are digital verifiable degrees needed in the job application process?
b. What is the current method to verify educational degrees?
c. What are current methods to create verifiable certificates for MOOC's, workshops, and job experience?
d. Why are the current methods not sufficient?

This chapter first delves into the job application process to further explain the motivation for easing the verification process of educational degrees.

2.1 Job application process

The job application process is an established process throughout the world. For vacancies with strong prerequisites, most firms follow similar steps in this process. In this section a typical job application process is illustrated, and the reasoning for the steps is explained.

In a typical job application process the recruiting company filters candidates through either two or three steps. First the candidates are shortlisted based on essential criteria. For most, especially complex, vacancies, a higher educational degree is part of the minimum qualifications (Cook, 2016).

The best candidates are then invited for an interview, and/or assessment. Optionally the recruiting company performs a background check on the remaining candidates. In a background check the company can verify the achieved degrees, contact references, and request a statement of conduct.

The complexity of higher educational jobs is increasing, and so the task of recruiting, and selecting will also become more complex (Rumsey, Walker, & Harris, 2013). While the complexity of the job application process may increase, the underlying selecting factor remains the same. Companies often claim to search for the most fitting employee. However, the main selecting factor is the trust factor (Interaction Associates and Human Capital Institute, 2013). The job application process has evolved into the current process for the sole purpose of building trust as quickly as possible. Each step in the process is devised to gain trust in the capabilities of the potential new employee. The recruiting company wants high confidence that an applicant will be able to perform the assigned tasks. In figure 2.1 the typical application process has been visualised.

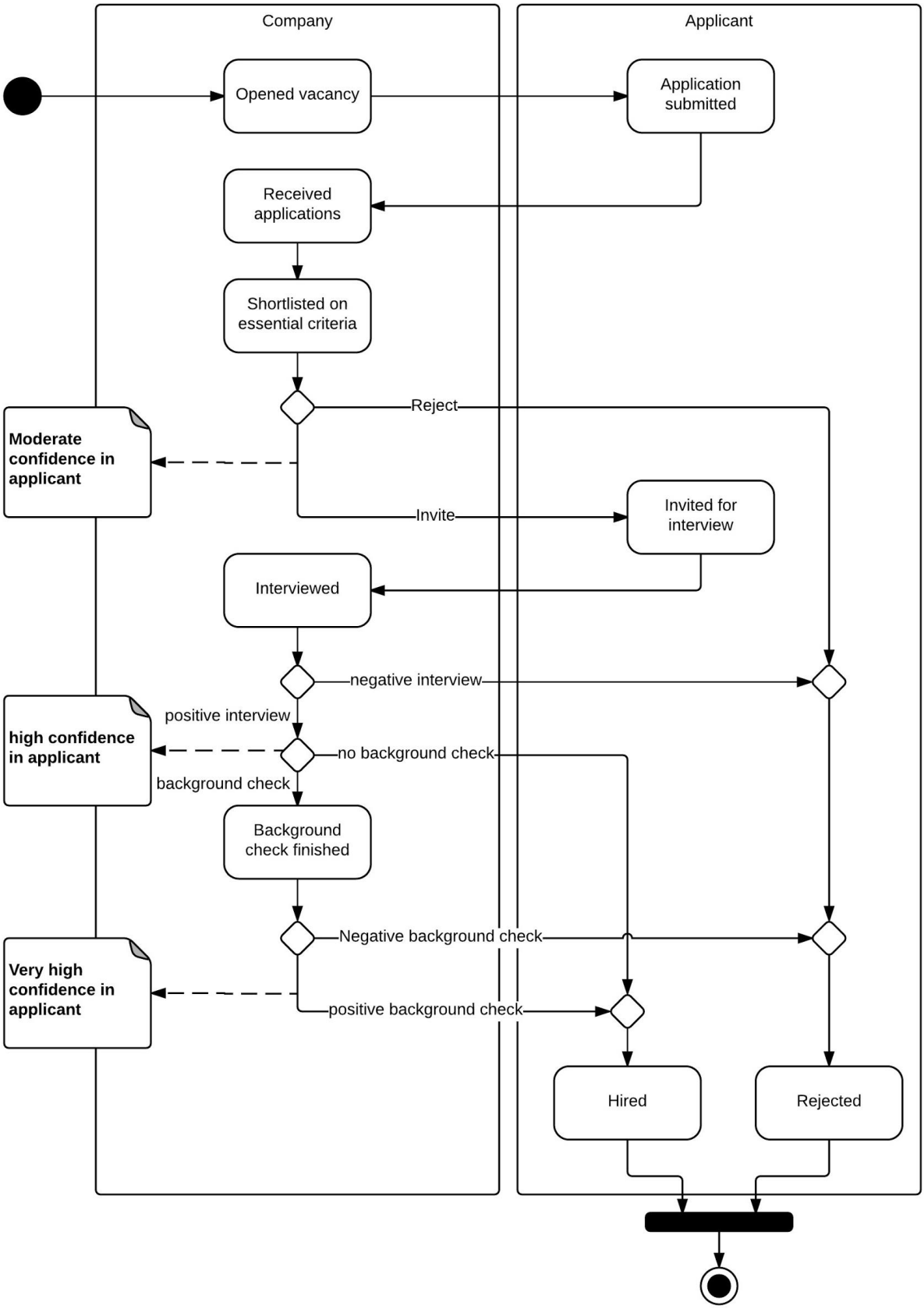


Figure 3: State diagram application process

2.1.1 Background check

Essentially a background check is a verification of the claims made by the applicant. Some companies are satisfied with the level of trust that an interview offers, and therefore do not perform background checks. Companies perform background checks to elevate the level of trust to the maximum reachable before hire. Reasons for a background check in practice:

- The company itself wants a higher level of trust in the potential new employee.
- Clients of the company require that its employees have certain verified degrees.
- Government regulation requires certain verification of a background. In the Netherlands there are over 100 regulated job titles (Directive 2005/36/EC, 2005)

2.1.2 Motivation

Degree verification is an important part of the background check process. A degree is often part of the minimum qualifications for a job, so verifying the applicant is truthful in their claimed degree is essential in building trust between the employer and potential employee.

As can be seen in the state diagram, Figure 3, the background check and therefore the verification of educational degrees is now done in a late stage of the job application process. The methods for verification will be covered in the next section. However, without knowing the cost and difficulty of these methods, enabling employers to do the verification earlier in the process will increase efficiency by detecting fraud early in the process.

As such, given that valid educational degrees are a deciding factor in the job application process, and time spent on job applicants with invalid educational degrees is a waste of time, making it trivial to verify a degree early in the job application process will increase the overall efficiency

2.2 Methods for educational achievement verification

In this part the current methods to verify the validity of educational achievement are discussed. A distinction is made between formally regulated educational degrees and informal unregulated educational achievement. The former are issued by higher educational institutes for the completion of a diploma granting study, while the latter can be issued by any institution for the completion of any form of education.

To understand the methods for educational achievement verification, we must consider the current issue process of degrees. A simplified model of this process is shown in Figure 4. As the degrees are issued to the alumni, they are not only stored locally at the higher educational institutes, but also in a central database hosted by the Dutch ministry for education (DUO).

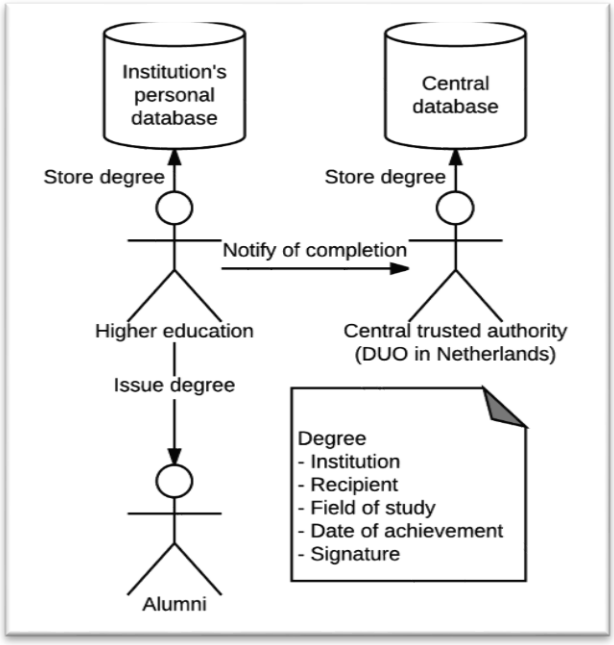


Figure 4: Current degree issue process in the Netherlands

2.3 Formal educational achievement verification

In formal higher education the degree is traditionally handed out in paper form. Although this is often a pleasantly formatted paper, the proof of validity is given merely by a pen signature and/or issuer stamp. Consequently, the person requiring verification of the degree has to trust the holder of the degree that the document was not forged.

With the advancements and availability of illustrator software, forging a degree is easier than ever (Rowley, 2012). Simply trusting the piece of paper, or a digital scan, is not good enough as verification of a degree.

Currently there are two options for verification:

1. Verify the degree(s) directly with the institution(s)
2. Verify the degree(s) indirectly via a central trusted authority/authorities

2.3.1 Verify directly

The employer, or other entity aiming to verify a degree, can do so by directly contacting the source of the issued degree. In the Netherlands a mere handful higher educational institutes explicitly state on their websites to provide verification for potential employers.

Universiteit Leiden, Saxion, Universiteit Utrecht (UU), and the Erasmus University Rotterdam (EUR) are as of these writings the only ones to openly state to have a process in place for degree verification (EUR, 2016b; Saxion, 2017; Universiteit Leiden, 2017; Universiteit Utrecht, 2017). All offer manual verification, which means an administrative officer of the higher educational institute will personally verify the requested degree with their personal systems. The UU and EUR both pass on the cost of labour for this process to the entities requesting verification by requiring a payment of 25 euros. The other institutes perform the manual degree verification for free.

The rest of the higher educational institutes do not explicitly offer degree verification. This does not imply they do not receive requests for verification nor that they ignore the requests. A telephonic survey conducted for this research with employees from the administrative offices of some of the higher educational institutes in the Netherlands yielded the results in Table 1.

Academic institute	Estimated verification requests per month	Unique remark
TU Delft	60	Yearly, around two of these requests return an invalid degree
Erasmus University Rotterdam	45	Despite the required payment of 25 euros, and offering a free direct alternative
Hogeschool Windesheim	20	Only one in five requests come from a Dutch entity.
Maastricht University	60	1% of the requests return an invalid degree
Saxion Deventer & Enschede	15	Most requests come from foreign head-hunters
Universiteit Leiden	40	Requests are quite evenly spread across studies
Vrije Universiteit Amsterdam	60	About ten minutes spent per request

Table 1: Survey about degree verification with various academic institutes in the Netherlands.

Although the data is limited a safe conclusion is that universities in the Netherlands receive 1-2 verification requests per day. The institutions who do not openly state their verification method receive a similar amount of verification request as the institutes that do.

The survey participants estimated that the time spent per request by the higher educational institute ranges from a few minutes to about a quarter of an hour. However, the time from request until response through the manual verification method can take anywhere from a few hours to several days depending on the other priorities of the administration offices.

2.3.2 Online registry

The EUR is the only higher educational institute in the Netherlands that offers a direct online verification option (EUR, 2016a). All the students of the EUR were asked during (re)-enrolment to be entered into the diploma register. Anyone can query the registry with just a few personal details about an alumnus from the university to find their degree(s), see Figure 5.

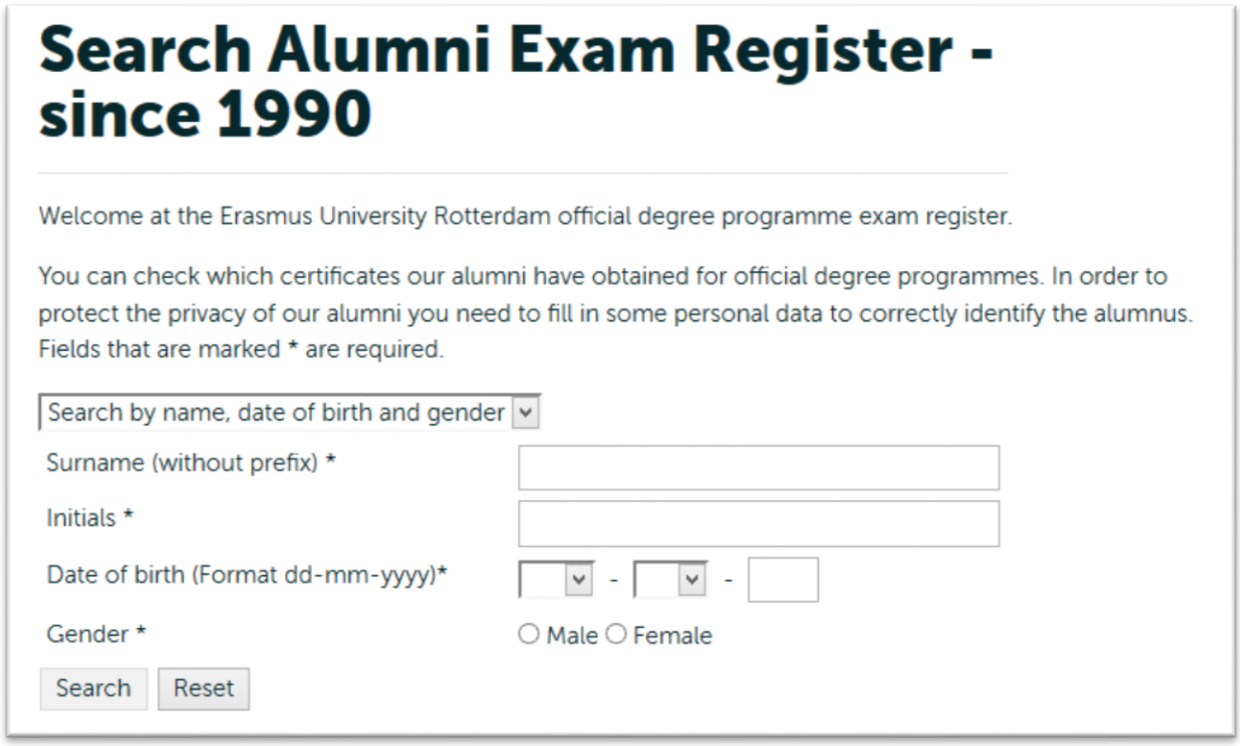


Figure 5: Diploma registry from the Erasmus University Rotterdam (EUR, 2016).

The registry provides a direct response. Interestingly, some employers still use the paid manual service at the EUR. The paid manual service does not supply extra information to the employer. According to the responsible department at the EUR, there are about 50-60 page views to the diploma registry page per day. It is unknown how many registry queries are done per day.

2.3.3 Verify indirectly with central authority

Verifying the claim indirectly via a central trusted authority is the other possibility. The Dutch ministry for education has a central database that stores the educational records for all students since 1996, and has been in operation since 2012 (DUO, 2012). The motivation for this register is to combat fraud, by having one central place to verify any Dutch diploma(Delta, 2011).

The data in this database can only be directly accessed by the data owners, so the students that obtained a degree. The degrees are stored as a pdf, and are digitally signed by DUO. The person doing the verification can then compare the digital signature on the pdf with signatures supplied by DUO to be sure of the validity of the document (DUO, 2017).

DUO also offers an integration service for employers and other degree verifiers. This integration automates the service, while still requiring the consent of the alumni. The cost for this service is 12 euros per request (DUO, 2017b). Although this can be a helpful service to companies who hire a lot of Dutch students, this method is not an adequate solution as is further explained in paragraph 3.1.4.

2.4 Informal educational achievement verification

In the previous paragraph the current verification methods of formal degrees have been explained. In this section some verification methods of the less formal forms of education are showcased, in order to get a broader view of the current situation.

2.4.1 MOOC certificate verification

Education is not a monopoly by the higher educational institutes anymore. Massive Online Open Courses have become a very popular way to share and obtain knowledge. These courses have

professionalized recently in the sense that for some of the courses even college credit can be obtained as recognition for your achievement (figure 2.4).

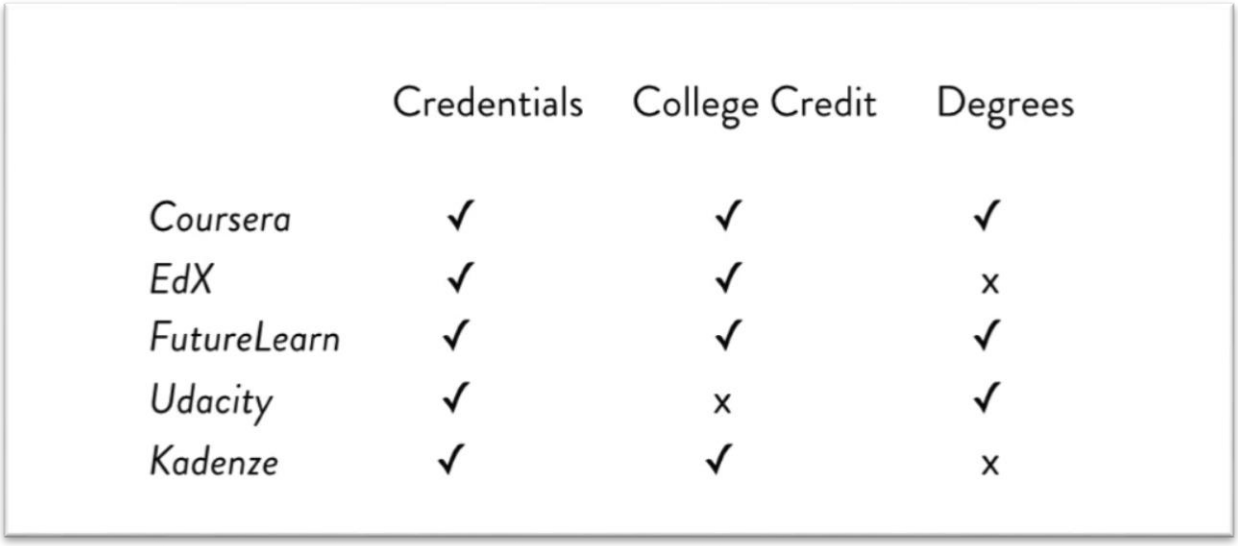


Figure 6: MOOC providers and their credential, credit and degree possibilities (Class Central, 2016).

The business model of these MOOC providers is quite similar. Most content is free to use, but to receive a certificate a payment is required (Coursera, 2017; edX, 2017; FutureLearn, 2017; Kadenze, 2017; Thrun, 2014).

These certificates can be used across the web, and can often also be showcased on professional social networking sites like LinkedIn. The verification of these can be done by using the unique identifier of the certificate and following it to the MOOC provider. This entails that the ultimate trust for the validity of these certificates is dependent on the administration of the MOOC website.

2.4.2 Open Badges

In 2011 The Mozilla Foundation created the Open Badges standard. This standard introduced a method to digitally recognise achievement outside formal educational institutions. These badges motivate the student by allowing them to showcase their learning experiences (Goligoski, 2012).

Mozilla has handed over the rights to govern the standard to IMS global, whose goal it is to enable better digital credentialing. IMS has noticed a trend in educational models that focus on the result of the educational process in the form of digital credentials (IMS GLOBAL, 2017a).

The Open Badges standard allows anyone to issue credentials. These are issued in the form of a digital badge with a JSON linked data (JSON-LD) structure in the metadata of the image. The open standard enables the receiver to hold all of their badges in a single place, referred to as the badge wallet. These can then be displayed in a CV like manner.

There are two standard supported methods for verification: hosted verification and signed verification. The issuer decides the method by supporting the right format in the JSON-LD. The hosted verification is similar to the verification of the MOOC certificates. A URI points to the issuer hosted website that contains the same certificate.

A certificate with the signed verification method contains a digital signature of the certificate. This digital signature ensures the integrity of the certificate if the key is a valid key from the issuer. The issuer will need to host their valid keys so verifiers can compare them to the ones in the signature (IMS GLOBAL, 2017b).

The introduction of this standard for issuing verifiable credentials is a leap forward from the traditional methods as explained in 2.3. However, the standard is not used for the publication and verification of formal degrees. The current verification method in this standard relies heavily on the original issuer's availability. This standard does provide a great starting point, but will need to be made more future proof. In chapter four we discuss how this standard is enhanced with this requirement in mind.

2.5 Generalised use case: Verifiable claims

The necessity of verifying documents is not exclusive to degrees in the context of a job/study application. There are many sectors where a verifiable credential from a person or organisation is required. These can be generalised as verifiable claims. In this section some examples of these claims are discussed, their user needs, and the application of this generalisation to the problem scope of this research.

All over the world companies are digitally transforming their businesses, and business models. Even though predictions for a completely paperless society still need to come true, every day fewer businesses are dependent on paper or face to face agreements (Dykstra et al., 2009). Making a digital agreement with another person across the world is easier than ever. However, current widespread techniques have drawbacks like the ability to fake one's online identity, and ease of creating false agreements. This makes it difficult to attain the same level of trust as one would in real life.

Therefore, making digitally verifiable claims to each other helps in building a trustworthy agreement and thereby relationship. The W3C credentials community acknowledged this need, and is actively developing requirements for a standard in verifiable claims (Andrieu, Lee, & Otto, 2017). This group defines a verifiable claim as:

> "A *verifiable claim* is a qualification, achievement, quality, or piece of information about an entity's background." ~*W3C credentials community*

The uses for these claims are broad. For example most of the interaction with a bank or mortgage provider requires verifiable claims. These industries are highly regulated regarding know your customer, and anti-money laundering laws. Customers should be able to provide verifiable proof of origin of their money. Doing these claims in a machine-readable verifiable way increases the efficiency and ease of compliance in this sector.

Many more uses are identified by the credentials community on a few focus domains. These are summarized in Figure 7.

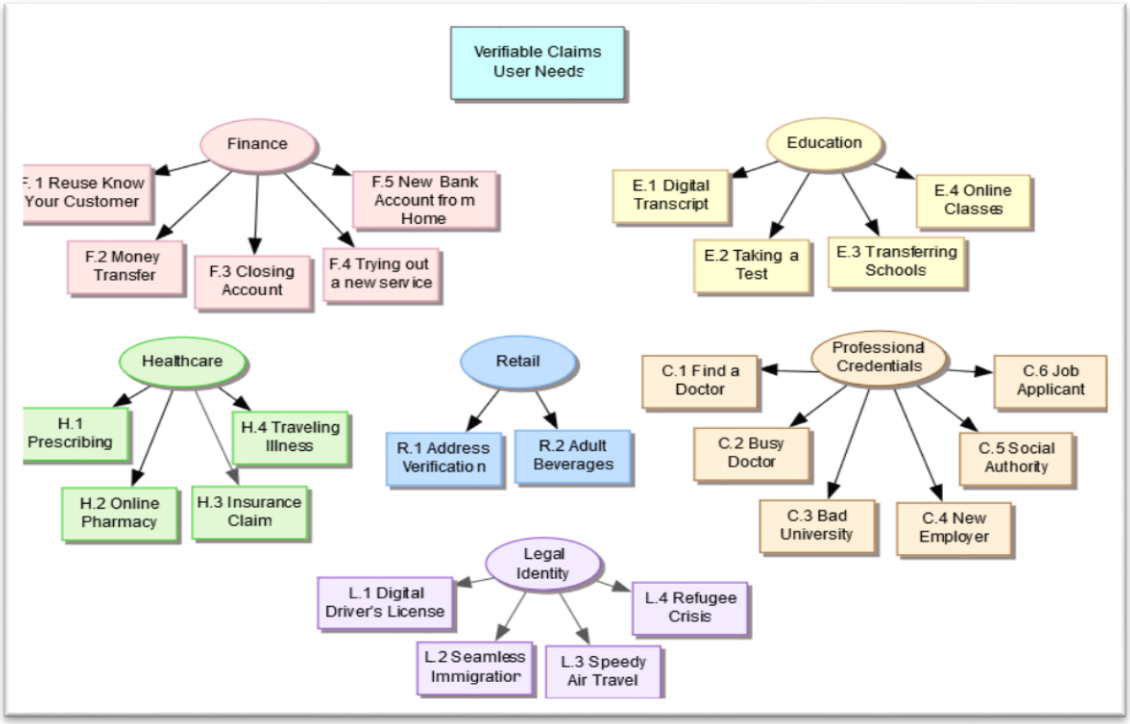


Figure 7: Verifiable claims uses in a few key domains (Andrieu et al., 2017).

In the provided definition, there is no mention of a necessity for the claims to be done online. For the application to a university, a prospective student will need to send a high school degree. The degree is verifiable by contacting the respective high school. However, in practice this is rarely done as it is a time intensive task. The purpose of the creation of a verifiable claims standard is to make claims machine-readable. So the verification of the claim becomes a trivial task.

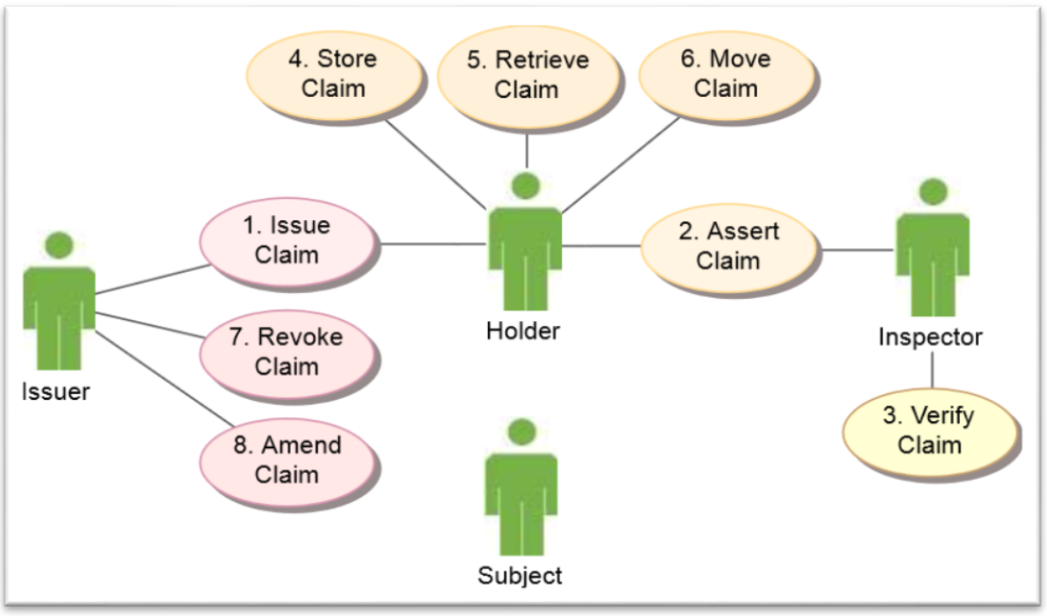


Figure 8: User tasks for verifiable claims (Andrieu et al., 2017).

The verifiable claims working group have also identified the user tasks in the context of verifiable claims, see Figure 8. The four roles in the verification process are the issuer, holder, subject, and inspector. Their role and user tasks are further summarized in this section. Note that in the perspective of the W3C credentials community the needs are stated as an ideal situation. Some needs are in reality carried out by a different role.

Issuer: Any entity can become an issuer by issuing a claim to a particular holder.

Needs:

- **1: Issue Claim:** The entity must be able to issue a claim to a holder. The claim is a statement about themselves that can be inherently trusted.
- **7: Revoke Claim:** The issuer must be able to revoke claims made earlier. The holder of the claim should not be able to pass verification when asserting parts of the revoked claim.
- **8: Amend Claim:** The issuer must be able to amend previous made claims. Some particular claims might need a yearly update that should be stored with the original claim.

Holder/subject: Any entity can become a holder by receiving a claim from an issuer. The subject is the entity that the claim is about, often the same entity as the holder. However, in some scenarios, for example a parent receiving claims about vaccinations for their kids, the holder and subject are different entities. The needs do not differ between the two entities.

Needs:

- **2: Assert claim:** The holder must be able to hand over a claim to an inspector for them to verify it. The holder must be able to choose to share parts of the claim, if not the entire claim is required.
- **4: Store claim:** The claim must be storable by the holder in a fitting repository.
- **5: Retrieve claim:** The claim must be retrievable from storage by the holder to send it to the inspector.
- **6: Move claim:** The holder is responsible for the storage location, and must be able to move their claim to another repository if requested. Factors for relocation can for example be employer requirements, privacy concerns, and accessibility.

Inspector: Any entity that requires a verifiable claim from any other entity.

Needs:

- **3: Verify claim:** The inspector must be able to verify the claim with the issuer. This can only be done with claims received by the holder.

2.6 Application to the case

The user needs as described by the W3 Credentials Community for verifiable claims, provide a basis for the requirements for a degree issuing and verification method. Here we look closer at these needs and define user stories according to a standard template. These stories are combined with the stories that arise from the technical background, and are implemented in the design chapter. For each user need, the stories are formed below according to the "As a __, I want to __, So that __" (Mike Cohn, 2008).

Claim Issuer: Higher educational institute('s student administrator)

Epic 1: As a higher educational institute, I want to issue standardized digital degrees to my students, So that I can get rid of the paper back-up, give students ownership over their data, and reduce the need for manual verification.

- *User Story (US) 1: As an educational institute's student administrator, I want to create and issue digital degrees from the student administration software, so that I minimize manual work and mistakes.*
- *US2: As an educational institute's student administrator, I want to revoke earlier published digital degrees, so that I can mitigate fraud like plagiarism.*
- *US3: As an educational institute's student administrator, I want to amend earlier published digital degrees, so that I can include missed credentials in an issued degree.*

Claim Holder/Subject: Alumnus

Epic 2: As an alumnus, I want to have full control over my degree data, so that I can have my credentials verified without endorsement of my former educational institute.

- *US4: As an alumnus, I want to be able to assert any subset of my degree data, so that I can choose what to share about myself.*
- *US5: As an alumnus, I want to choose where to store my degree data, so that I can have full control over my degree data.*
- *US6: As an alumnus, I want to be able to retrieve or move the degree data at any time, so that I don't lose the data.*

Claim inspector: e.g. Employer

Epic 3: As an employer, I want to quickly verify the employee's asserted digital degree, so that I can verify the credentials earlier in the job application and know it isn't fraudulent.

- *US7: As an employer, I want to be able to accept digital degree assertions, so that I can verify these with less manual work.*
- *US8: As an employer, I want to instantly verify that the asserted digital degree belongs to the employee, so that I know the employee gave his own credential.*
- *US9: As an employer, I want to instantly verify that the asserted digital degree has not been altered or otherwise tampered with, so that I know I verified the original document.*
- *US10: As an employer, I want to instantly verify that the asserted digital degree was originally issued by the stated educational institute, so that I know which institute the degree is from.*

The verification stories for the inspector have been split up into three, as these are individual distinguishable steps in the verification process. However, all these steps need to return the correct result before the inspector should regard the degree as verified.

The stories of the inspector are written from the employer persona. However, these same stories apply to other inspectors. In a validation interview of these user stories, a higher educational institute commented that the institutes themselves inspect a lot of degrees for the acceptance of Master's students. Therefore, the design takes into account that any party can perform the verification.

In Table 2 the user stories are mapped to the current available methods for verification. Both current methods for formal degree verification do not suffice all user stories, so improvements should be made to overcome this gap. The current verification methods focus on the higher educational institutions, instead of on the actual owner of the data, the alumnus. The design should be beneficial for all stakeholders and should fulfil all the user stories.

	Direct verification (at institutions)	Indirect verification (at DUO)
US1	Implemented	Implemented
US2	Implemented	Implemented
US3	Implemented	Implemented
US4	Not implemented	Not implemented

US5	Not implemented	Not implemented
US6	Not implemented	Not implemented
US7	Not implemented	Not implemented
US8	Implemented by some institutions	Implemented
US9	Implemented by some institutions	Implemented
US10	Implemented by some institutions	Not Implemented

Table 2: Mapping the user stories to the current degree verification methods

2.7 Conclusion

In this chapter we have motivated the research by showing there is an active problem in the inefficiency in the degree verification and therefore need for a universal digital standard. Moreover, we explored the generalized applicability of such a standard, and distilled some functional user stories from the user needs for verifiable claims. We have concluded that both currently available methods to perform degree verification do not satisfy these user stories. Thus, we continue with studying alternatives.

3 Technological background: Blockchain & smart contracts

This chapter covers the technological background for the creation of a digitally verifiable degree on a public blockchain. Both the state of the art of blockchain and smart contract technology are delved into. The overhanging research question for this part is:

What are the benefits and limitations of blockchain and smart contract technology as a method for verifiable degrees?

a. What is the state of the art of blockchain technology?
b. What is the state of the art of smart contract technology?
c. What are the characteristics of these technologies that make it a contender for solving the problems identified?

This chapter also introduces technologically related work, i.e. other efforts to create verifiable certificates on blockchain technology. Most importantly the standards that are used in the design chapter.

3.1 Blockchain technology

The original Bitcoin paper introduced the world to blockchain technology in 2009 (Nakamoto, 2008). Initially blockchain was regarded simply as the technology that powered Bitcoin. Even until quite recently, as proves the widely cited book on the topic from 2015: "Blockchain: Blueprint for a new economy":

"The blockchain is the public ledger of all Bitcoin transactions that have ever been executed" (Melanie Swan, 2015).

In the past three years many have realised that the technology is applicable to many more uses than purely an electronic peer 2 peer cash system. Hal Finney, so-called cyberpunk and the first person to receive a Bitcoin transaction after Satoshi, explained the innovation as follows on a cryptography mailing list:

"One thing I might mention is that in many ways bitcoin is two independent ideas: a way of solving the problems of creating a globally consistent but decentralized database; and then using it for a system similar to Wei Dai's b-money (which is referenced in the paper) but transaction/coin based rather than account based. Solving the global, massively decentralized database problem is arguably the harder part, as James emphasizes. The use of proof-of-work as a tool for this purpose is a novel idea well worth further review IMO." (Finney & Nakamoto, 2009).

So, blockchain is the first of these two independent ideas. A globally consistent decentralized database. In practise this database is append only, and thus in essence immutable. The technology can keep globally consistent, because of the consensus algorithm. For Bitcoin, and most other public blockchains this algorithm type is Proof of Work, but some coins are working on different consensus types. Explaining the technicalities behind these consensus algorithms is out of the scope of this research, and regarded as common knowledge of the readers.

Blockchain has become a tool which can be applied to any problem space, like it has been for cash in the case of Bitcoin. Many companies are actively researching the implications and possibilities that this technology brings. However, only a few have achieved to put an application that uses a blockchain in production.

3.1.1 Public, permissioned, and private blockchains

Blockchains are generally divided in three different types: public, permissioned, and private. The type of blockchain is dependent on the access rights to the blockchain. In Table 3 the access properties of for each of these types are noted.

Who can…	… become validator of transactions?	… write transactions?	… read the transaction input/output?

Public	Anyone	Anyone	Anyone
Permissioned	Decided by current validators	Depends, often current validators	Depends, often anyone
Private	Decided by current validators	Only current validators	Only current validators.

Table 3: Public, permissioned, private blockchains and their access rights

Public blockchains are most permissive, as anyone can become a validator and read/write transactions to the chain. However, this comes as a cost in regards of performance. A network of stakeholders that are setting up a blockchain can choose much higher technical requirements in terms of GPU power and network speed than a general household has. Public blockchains strive for decentralization in the form that anyone can become a validator, which means that the requirements to join the network have to be relatively low.

Another limitation of a public blockchain versus permissioned and private blockchains is the fact that all data is public. One of the main reasons to use a less permissive blockchain is in order to keep certain transaction information private. In a public blockchain setting, all the transaction data is open for everyone.

3.1.2 Scalability

The current available public blockchains do not scale very well. Bitcoin can handle up to about 7 transactions per second, and Ethereum can handle about 15 per second. Both of these blockchains regularly reach their cap. To increase the possible adoption, the underlying protocols need scalability upgrades in order to handle a higher transaction throughput.

According to research by the Ethereum foundation, the difficulty of this problem lies in the following scalability trilemma (Buterin Vitalik, 2018):

- ***Decentralization*** *(defined as the system being able to run in a scenario where each participant only has access to O(c) resources, i.e. a regular laptop)*
- ***Scalability*** *(defined as being able to process O(n) > O(c) transactions)*
- ***Security*** *(defined as being secure against attackers with up to O(n) resources)*

Here **c** refers to the size of computational resources, and **n** refers to the size of the ecosystem in transaction load and state size. This scalability trilemma makes that only two of the three properties can be achieved. There might be solutions to the trilemma in the form of sharding, but this is still actively researched and not yet developed. In order to make a currently working application on a public blockchain, the scalability issues should be noted.

While a public blockchain is available for everyone to join, and perform transactions, the draw-backs are clear. Lower transaction throughput, and less privacy options when compared to permissioned blockchain options.

Therefore, I argue that when a certain method is proven to work in a public blockchain setting, the same method will definitely also suffice in a permissioned blockchain network. As is clear from the drawbacks, this argument currently does not hold the other way around.

3.1.3 Merkle trees

It is not for nothing that cryptocurrencies has “crypto” in its name. Various types of hashing algorithms are used throughout all blockchain implementations. Merkle trees improve the speed of verifying the validity of various hashes by bundling them together and creating a tree out of it.

Merkle trees an invention by Ralph C. Merkle from twenty years before the Bitcoin paper (Merkle,

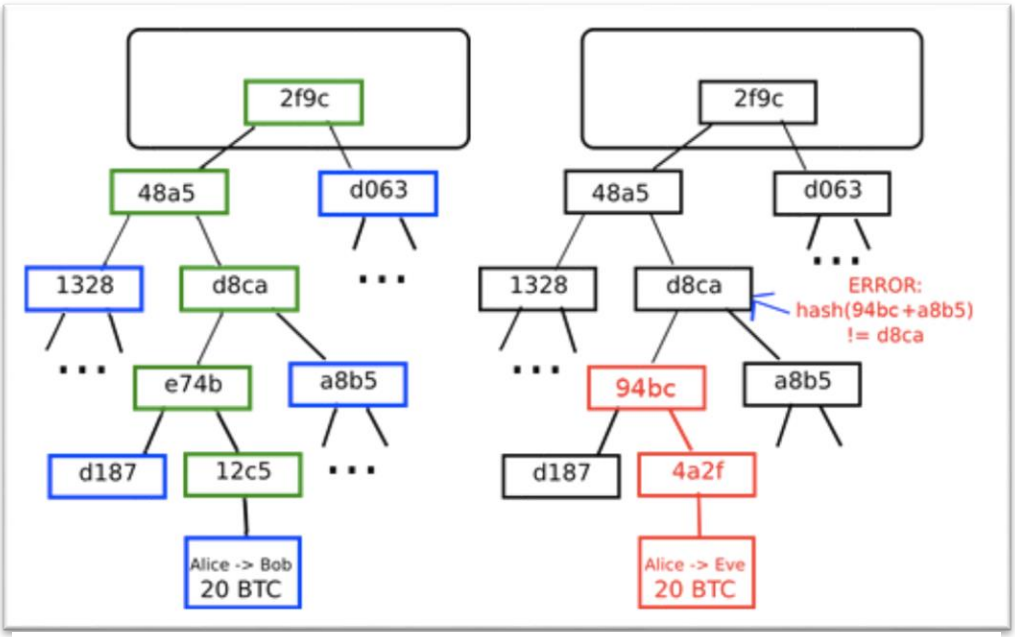


Figure 8: Left: only small amount of nodes required to calculate node proof. Right: Any change to any node in the Merkle tree leads to an invalid tree (Buterin, 2014).

1988). These trees enable very fast verification of the underlying data. In Figure 8, the left Merkle tree shows that in this hash tree structure, only several nodes out of a large set are required to prove that the transaction of 20 BTC from Alice → Bob is indeed part of the underlying data of the Merkle root (i.e. 2f9c). If an attacker would later try to change the receiver of the earlier made transaction, then the Merkle tree would become invalid. This is the advantage of using Merkle trees for blockchain implementations: fast verification of the incoming data by other participants, and only having to verify a small part for each incoming transaction.

Figure 10: Poll for the leader of the free world. Ends up slightly in favour of Merkle!

These Merkle trees can also be used to minimize the required data that is stored on the blockchain. Various projects have used this technique in their attempt to notarize arbitrary data. These projects are further discussed in the related work part of the next section.

3.1.4 Blockchain vs. current systems with traditional databases

In the previous section we have covered a few of the advantages and disadvantages of blockchain technology. Here, these are expanded on a little further, and compared to setting up a traditional database.

3.1.4.1 (De)centralization

The most important difference between a traditional database and a blockchain is the decentralization aspect. However, this characteristic is often misunderstood. The creator of Ethereum defined the meaning of decentralization on three dimensions: politically, physically, and logically (Buterin, 2017a).

- **Politically (de)centralized:** depends on the amount of people or organisations that ultimately decide on the faith of the system in the case of changes, upgrades, or other governance questions.
- **Physically (de)centralized**: depends on the number of computers a system is made up of. The more random outages the system can handle without being fully unavailable, the more physically decentralized it is.
- **Logically (de)centralized:** depends on whether the system is viewed as one single entity, or is comprised of many different entities.

One can now rank systems according to these three dimensions. In Table 4 a public blockchain system and the currently available degree verification methods at Dutch universities are compared according to these three dimensions of decentralization. The two verification methods refer to the methods explained in the previous chapter. Below the table, each cell is further explained.

	Politically	Physically	Logically
Public blockchain	(1) Decentralized	(2) Decentralized	(3) Centralized
Direct verification system (at institutions)	(4) Decentralized	(5) Centralized	(6) Decentralized
Indirect verification system (at DUO)	(7) Centralized	(8) Centralized	(9) Centralized

Table 4: (De)centralization of a public blockchain vs. the current available degree verification methods

Public blockchain (Buterin, 2017a)

1. Public blockchains are politically decentralized, as every node owner can decide for themselves which implementation of software to run, and thereby choose to upgrade (or choose not to).
2. These systems are also physically decentralized. The Bitcoin network is currently maintained by over ten thousand nodes, and Ethereum by over fifteen thousand (Flippening, 2018).
3. Lastly, they are logically centralized. A public blockchain has one commonly agreed upon state, and it behaves like a single entity. One can address the system to any node in the network, and expect the same behaviour.

Direct verification system (at institutions)

4. This system is politically decentralized, as each institution decides individually how the verification requests are currently handled at their institution, and decide themselves if this workflow is ever changed.
5. Physically this system is often very centralized compared to a blockchain system. There is one place per institution where the degrees are stored with perhaps a single to a few back-ups. At each institution only the degrees issued by that institution are stored.
6. This verification system is logically decentralized, as there is no standard for verification requests of degrees that is supported by multiple institutions.

Indirect verification system (at DUO)

7. Politically, this system is centralized. DUO ultimately decides on the faith of this system. They might have the best intentions to follow the needs of the users and issuing institutions, but they do not have to.

8. This system is also physically centralized. It is not publicly known how DUO stores the data in their degree registry. However, while the data is hopefully physically at least in a few places, it is far less than a public blockchain system.
9. The system is also logically centralized. This is a good property, as there is a single method of interacting with the system to get a degree verified.

In Buterin's article on the meaning of decentralization, several advantages are noted. The most important are: fault tolerance, attack resistance, and collusion resistance (Buterin, 2017a).

In the design of systems that cross organisations, institutions, and borders, the ideal (de)centralization properties to capitulate on these advantages are: *politically decentralized* (so each participating party has a say in the governance decisions), *physically decentralized* (so the system is resilient to catastrophic failures), and *logically centralized* (so the system can be easier optimized and automated).

In conclusion, both current verification systems do not meet the identified ideal properties of the decentralization dimensions. A public blockchain system does meet these requirements. Therefore, researching the design of a degree verification system on a blockchain is valuable for solving this problem space.

3.2 Smart contract technology

Bitcoin is a great showcase of the capabilities of blockchain technology, and made it clear that it could possibly be applied to many use cases. The moment Bitcoin gained popularity, many clones and slightly altered forks were deployed. Creating a clone is not much effort, as source code is open and without license restrictions. Most clones were fairly low-effort, but a few of the early ones are still quite popular, like Litecoin and arguably even Dogecoin (Dogecoin, 2018; Litecoin, 2018). Others, like Namecoin, and Peercoin, have a good use case, but are not actively improved(Namecoin, 2018; Peercoin, 2018). The most important reason for their demise is that these blockchains target a very niche use case which has too little business value to sustain the infrastructure requirements needed for a blockchain. This problem was the motivation for Ethereum, making sure any application can benefit from the characteristics of a blockchain.

Ethereum introduced a programming language used to create small programs that are executed on every participating node of the Ethereum blockchain (Buterin, 2014). This programming language on a blockchain enables "rich statefulness", a notion of state more powerful than the traditional model used in blockchains like Bitcoin (Buterin, 2017b).

3.2.1 Characteristics

3.2.1.1 Rich statefulness

Bitcoin uses the unspent transaction output (UTXO) model, which checks for every transaction whether the input coins have been spend yet (Bitcoin.org, 2018). In Ethereum the state of self-conceived variables can be stored, and the smart contracts can create complex rules by which the state of the variables may be altered. This system can for example be used to create decentralized insurance contracts, provably fair online gambling, and an advanced decentralized naming system (Ethereum Name Service, 2018; Etherisc, 2018; FunFair, 2018) .

Since Ethereum, various other smart contract platforms have been created. Projects like NEO, EOS, Cardano, TRON, and NEM, all try to become the most advanced public smart contract platform (Cardano, 2018; EOSio, 2018; NEM, 2018; NEO, 2018; TRON, 2018). Most of these platforms try to solve the scalability issue of Ethereum, but do so at the cost of decentralisation. By far the most developers, and development is currently happening on Ethereum thanks to the first mover advantage, but also because many enterprise are contributing (EEA, 2018). Therefore the rest of this thesis will focus on Ethereum as public smart contract platform.

3.2.1.2 Non-Upgradability

Upgrading a smart contract on a public blockchain is not trivial. Once a smart contract has been deployed, there is no built in upgrade function. The publisher of the contract can make the contract upgradable by building this function themselves. However, this puts trust in the creator, as the code within the functions of the original smart contract can then be upgraded to anything. There are several patterns for upgradability in smart contracts, but none have been used very widespread, yet (Calvanese, 2018).

3.2.1.3 Performance

Also, smart contracts are not optimized to be the most efficient. The virtual machine on which the contracts run should be executable on relatively low-end computers, as the goal of a public blockchain is to stay decentralized. In the public Ethereum blockchain every single node runs every single transaction, and thereby every smart contract execution. Thus, only the strictly required operations should be included in the smart contracts to keep it optimized. Any code which doesn't require the decentralized execution, should be run outside the blockchain.

Vitalik Buterin estimates that the public Ethereum chain is currently around a million times less efficient to run code on compared to centralized solutions, thus the decentralization aspect of the blockchain should be worth it for the use case (Buterin, 2018). Even when all the scaling solutions are implemented, an estimated 1000 time inefficiency loss is expected when using a blockchain for computation. Therefore, the benefit per user/beneficiary of blockchain is most useful in high value transfers, like in cryptocurrencies (Buterin, 2015).

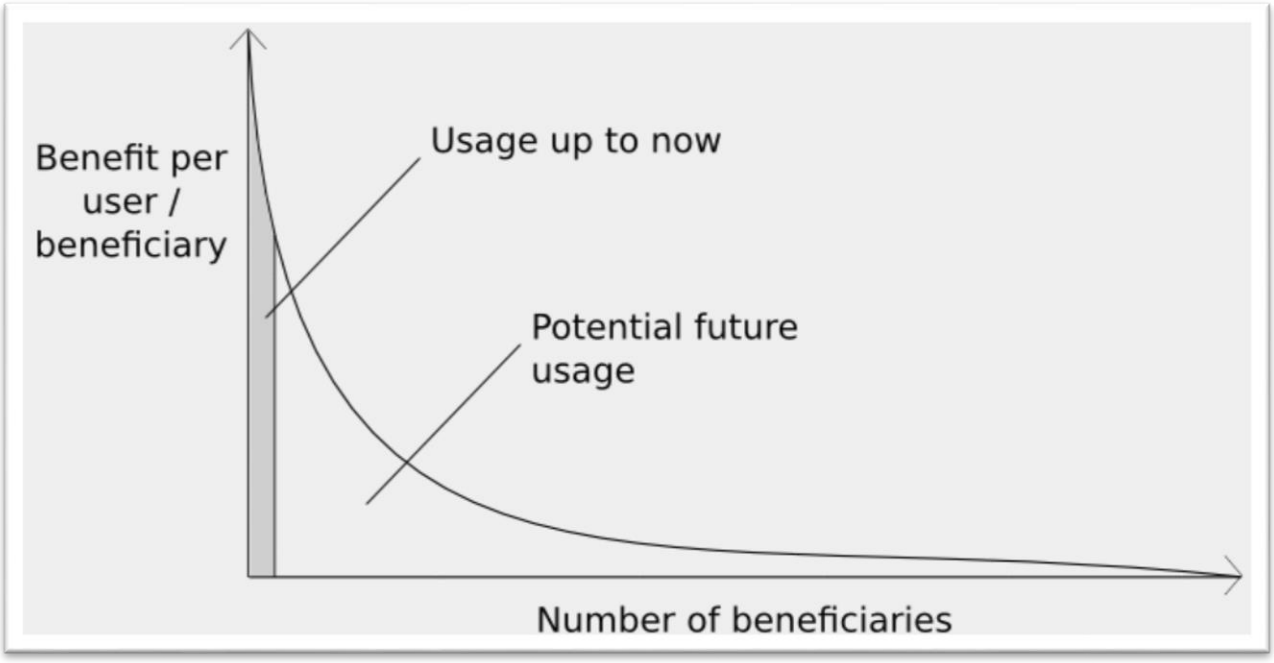


Figure 11: Usefulness of blockchain technology (Buterin, 2015).

In 2015, before the Ethereum public chain was finished, the creator explained what use cases Ethereum would be most suitable for. This explanation is visualised in Figure 11. According to the creator, the strength of Ethereum lies in enabling the "long tail" for potential usage of blockchain technology, by supplying a decentralized infrastructure for even the smallest use cases (Buterin, 2015).

3.2.1.4 Data

Another element to keep in mind is that any data stored within a smart contract on a public chain is publicly readable. This means that it should only be used for data types of which the owner does not

mind that the data is out, forever. Although, one could encrypt data, or create Merkle Trees and just publish the tree root on the blockchain. In that last pattern, it should be noted that once the data on the leaves of the tree is publicized, anyone can validate it was part of the Merkle root, forever.

3.2.1.5 Privacy and law in blockchain identity solutions

Using an immutable ledger for private and privacy sensitive information is easy to do, but ethically, and even legally questionable. Therefore, creating an identity solution that abide the privacy needs is easier said than done.

Suppose for example a smart contract that is used to store a single item of personal information. Even if the contract contains a function to delete the information, the original input transactions will live as long as the particular blockchain does.

Since the introduction of the European GDPR law, an active legal discussion is being held on the implications on blockchain technology, see: Blockchain Technology and the GDPR - How to Reconcile Privacy and Distributed Ledgers.

While an overall identity solution is out of the scope of this research, the design should make sure no personal information is stored in any way on the blockchain.

3.3 Blockchain and identity

Proving one's identity digitally has always been a difficult problem. Blockchain technology nor smart contracts provide an immediate solution for that. Proving an identity always relies on a trust anchor, i.e. you either have to trust the identification based on the person itself, or based on another entity like the government.

The founder of Aragon, a project building tools for decentralized autonomous organisations, held a Twitter poll at the start of this year that asked what is most needed in the Ethereum ecosystem (Aragon, 2018; Cuende, 2018). As can be seen in Figure 12, the majority of the votes agreed that an identity solution is most needed.

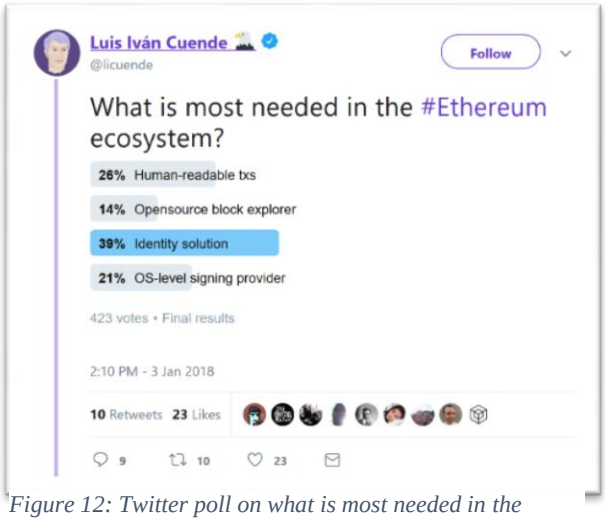


Figure 12: Twitter poll on what is most needed in the Ethereum ecosystem (Cuende, 2018).

Although blockchains contain built-in decentralized public key infrastructure, which enables users to create public and private key pairs, with which they can identify themselves. *Ownership* over a public key ultimately comes down to knowing the private key, as there is no centralized authority who determine ownership like in most current traditional systems.

A worldwide pseudonymous system like Ethereum, but also other systems that rely on decentralized public key infrastructure, have the following inherent identity problems:

1. A public key can be owned by multiple identities. On top of that, one identity can own multiple public keys. As visualized in Figure 13: there is no inherent 1 to 1 relation, which an identity solution requires.

2. If a 1 to 1 relation between a person's identity and a public key is established, the identity should not be lost or even stolen in the case of a private key loss or compromise.

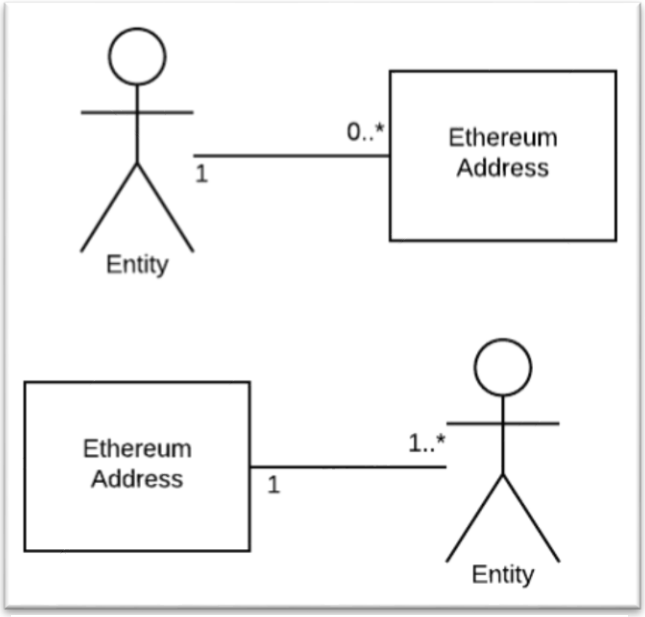


Figure 13: Identity problem for public keys.

3.3.1 Self-sovereign identity

Creating a blockchain based identity management solution is out of the scope of this research, thus in the proposed design we only give the requirements for the identity management system. The proposed design focusses on making a digital claim in the context of degrees verifiable with certain identity assumptions.

For the interested readers, there is a lot of work being undertaken by many parties to create a decentralized identity management system. Their motivation is to enable self-sovereign identity, a world where the user is central to the administration of identity (Allan, 2016).

3.3.1.1 ERC780 + 1056

The creators of uPort are standardizing the interfaces of a lightweight on-chain identity contract (ERC1056) and an on-chain claims registry (ERC780). Especially the on-chain claims registry is very tiny, as its implementation consists of just 40 lines of Solidity. It is made for general verifiable claims use cases, and is therefore also applied in our design proposal.

3.4 Usability vs. security trade-off

When designing methods for contexts that handle anything of value, there is a trade-off relation between usability and security. The most ideal solution from the usability perspective is never the most secure solution to the problem. And, the other way around, the most secure solution, is never the most usable. Therefore, a compromise should be found, where both the minimum security requirements and the minimum usability requirements are met. See Figure 14 for a visualization of this relationship (Braz, Seffah, & Raihi, 2007).

In a blockchain context, the compromise that is chosen as trade-off for usability and security is the unique selling point of an application. For example: comparing centralized exchanges (like Binance http://binance.com/) to decentralized exchanges (like Etherdelta http://etherdelta.github.io/), the centralized ones can offer greater usability. Their fast centralized servers, and a forget-password feature, are superior in usability. However, as the 0x project described it in their whitepaper: "decentralized exchanges can provide stronger security guarantees to end users since there is no longer a central party which can be hacked, run away with customer funds or be subjected to government regulations."

(Warren & Bandeali, 2017). However, users should realise that in a decentralized exchange setting the security of their funds is their own responsibility.

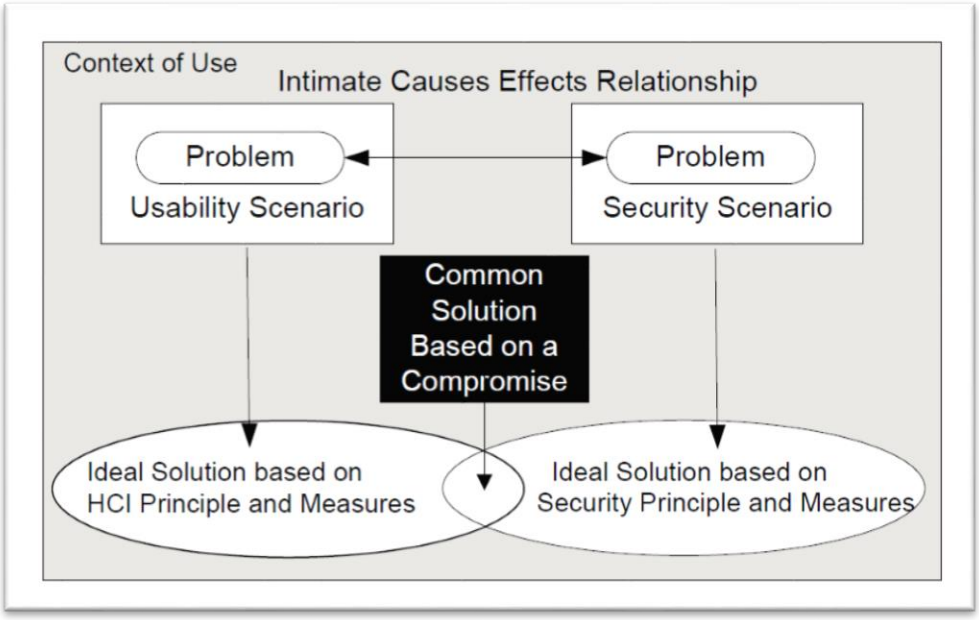


Figure 14: Usability and Security trade-off: A common solution based on a compromise.

In the case of utilizing smart contracts as a degree registry there is also a security versus usability trade-off. The following problems are competing against each other:

Security problem: Both the issuing party and the receiving party want full control over their issued, and received degrees. With blockchain technology there is no central party that can be called if something happens what is not in line with your intentions. If the issuer's private key solely control which degrees get to be issued, then the private key may never be stolen, deleted, or forgotten. The same holds for the receiver. Therefore extra security measures are required to keep issuers and receivers in control.

Usability problem: A receiver would like their degree as fast as possible after completion, and attesting it should be as easy as sending a message. A higher educational institute wants to be able to issue degrees just as quickly as they can do today, and don't want to take a lot of extra steps. Therefore the steps for both the issuer and the receiver to complete their needs, have to be as few as possible.

3.5 Security and usability user stories

In the proposed design a compromise has to be found for the above explained security versus usability trade-off. The design goal is to implement the following user stories:

- *US11: As an institution I want to be able to update my issuing key so that I don't lose issue rights if I accidentally lose my private key.*
- *US12: As an institution I want to be notified if my private key is compromised, so that I can update my key to make sure others can't fraudulently use it.*
- *US13: As a receiver I want to be able to update my receiving keys so that I can control my data in the case that my keys are lost or stolen.*

3.6 Conclusion

The application building language allows a smart contract platform to enable more complex types of transactions than traditional public blockchains could. The rich state that a smart contract can alter comes with some general trade-offs when comparing a publicly run smart contract to centrally run code. Although smart contracts lack easy upgradability, are far from efficient, and should only store public data, the technology enables decentralisation of the smallest possible use cases one can think of.

By applying this technology to this use case one needs to keep the identity problem in mind. An entire decentralized identity model is a big challenge in itself, but luckily being worked on by multiple companies and researchers. In our proposed design we leave the full identity problem out of the scope, but we take a few user stories regarding the identity problem into account, so the design is secure and usable.

4 Current efforts of credential notarization on blockchain technology

As both blockchain and smart contract technology have been introduced, it's time to move onto the current efforts of using these technologies for notarizing achievements and degrees. In this part the most significant efforts are discussed to show how the field has evolved. We tackle the following research question:

How can the current efforts be improved with smart contract technology to a functioning electronic degree verification method?

a. What are the current efforts of degree issuance on blockchain technology?
b. Why are these methods lacking as a functional verifiable degree?

We start by chronologically discussing the existing solutions in this area. Then the most promising existing solution is discussed and we comment how it can be further improved.

4.1 Existing solutions for blockchain based document notarization

4.1.1 Proof of Existence

The Proof of Existence service was launched in 2013 by Manuel Araoz and Esteban Ordano to allow anyone to notarize any document on the Bitcoin blockchain (Manuel Araoz, 2018). This was achieved by hashing the original document, and adding this hash to the OP_RETURN opcode (Bitcoin Core, 2018). This script marks the transaction output as invalid, and as such not spendable. As the Bitcoin protocol is based on the UTXO model, these unspendable transactions are part of the Bitcoin state forever.

This mechanism allows anyone to proof that a certain document existed at the time the hash was published to the Bitcoin blockchain. While there may have been some degrees notarized through this service, it did not focus on this use-case and was just open sourced as a proof of concept.

4.1.2 University of Nicosia

The Proof of Existence experiment inspired the University of Nicosia to publish the proof of the certificates for their digital currency MOOC on the Bitcoin blockchain as well (UNIC, 2017). As far as is known publicly, they were the first to publish proof of actual academic credentials to a public blockchain. In a later iteration of their certification method, they used Merkle trees to bundle all the certificates of one semester into one transaction for optimization and improved validation speeds.

4.1.3 MIT Media Lab

The MIT Media Lab built the first issuer agnostic degree notarization service on the Bitcoin Blockchain. Their high level concept has been visualised in Figure 15. Their proof of concept improved upon the proof of existence methods by standardizing the certificate information format according to the already existing Open Badges standard (MIT Media lab, 2016).

Their method also added revocation options. As a certificate is published, the respective Bitcoin transaction also contains a small spendable output to the issuing public key. The experiment eventually resulted in the creation of the Blockcerts open standard. Blockcerts are delved into more in that piece.

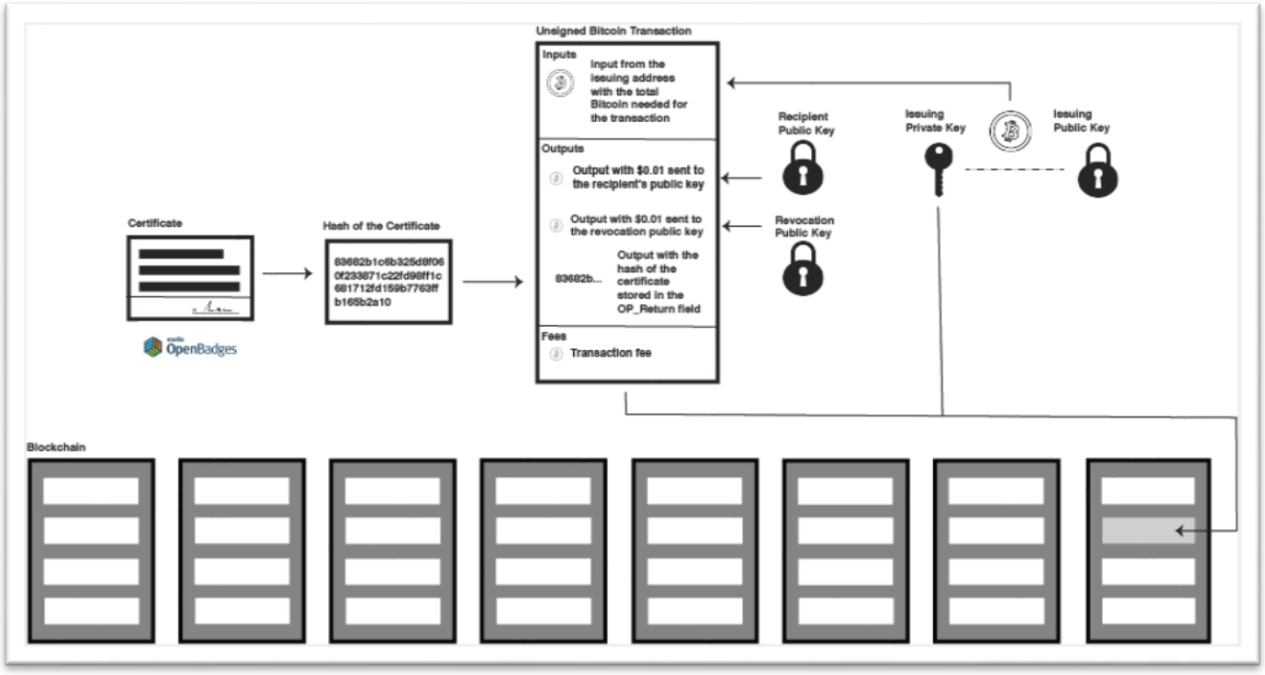


Figure 15: Architecture of MIT Media Lab blockchain certification PoC (MIT Media lab, 2016).

4.1.4 TrueRec

People studying blockchain technology at SAP also came up with the notarizing certificates use case. Not much information on the inner workings has been made public aside from that it uses the public Ethereum network for the anchoring transactions, instead of the Bitcoin blockchain. SAP also included an app to increase the user experience (Benjamin Boeser, 2017). However, the actual method has not been made public, thus a vendor lock-in arises when this method is used.

4.1.5 Initial Coin Offerings

There have also been several Initial Coin Offerings (ICO's) which aim to create a token economy around the issuance and verification of academic degrees on a blockchain. For example Skillchain and BCDiploma try to leverage a token in their plan to make degrees verifiable on a blockchain (BCDiploma, 2018; SkillChain, 2018). These projects are both creating something far from an open standard, and thus are not considered further in this research.

There are also several other companies, universities, and other institutes that have announced to be working on a blockchain notarized degree, but the aforementioned were the current most notable. Recently a project

4.1.6 Blockcerts

Blockcerts is currently the best known open standard for issuing credentials on the blockchain. The MIT Media Lab project evolved into the Blockcerts open standard after their proof of concept was finished (Blockcerts, 2018a). This standard is based on, and kept compatible to the IMSGlobal Open Badges standard (IMS GLOBAL, 2017a). Blockcerts is available for anyone, and has open source tools on Github to issue, assert and verify credentials according to the standard (Blockcerts, 2018b). An example certificate issued by the University of Twente according to the Blockcerts standard has been included in Appendix A.

Learning Machine is commercializing Blockcerts by building enterprise ready software for institutions that want to issue Blockcerts. In Figure 16 various current efforts are drawn on two axis: recipient ownership, and vendor independence. The aforementioned ICO's do give the recipient ownership over their data, but their systems require specific tokens on top of the already paid network fees thus are

dependent on the vendor of those tokens. Although this figure has been drawn by the CEO of Learning Machine, we do agree that of the current efforts Blockcerts have the most vendor independent standard, and lays the data ownership at the recipient.

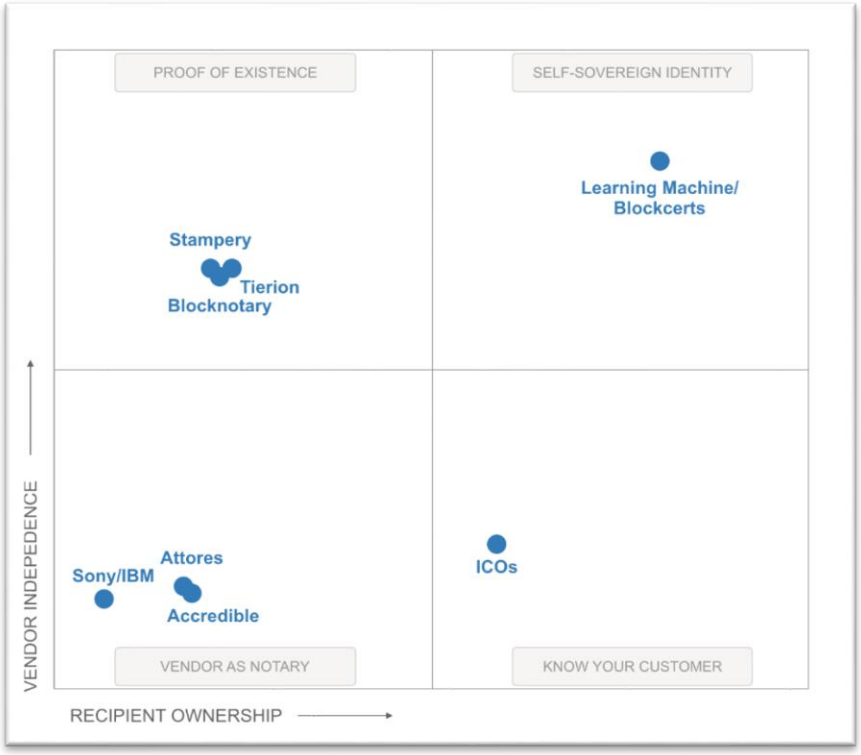


Figure 16: Current efforts for credential issuing, drawn by the CEO of Learning Machine (Jagers, 2018)

The current technical architecture of the Blockcerts modules is visualised in Figure 17. To issue a certificate, one first needs to create the certificates according to the standard. To do so, one can use the Cert-Tools and a CSV file of the recipients, together with some parameters to give details on the type of certificate. After the unsigned certificates have been created, they are signed and published in a transaction to a blockchain. Then the corresponding transaction hash is included in the Blockcert, and it is send to the receiver as it is now verifiable. The receiver can assert the certificate at any time to an inspector, who can view a formatted version with the Cert Viewer, and then verify it.

Blockcerts has been set-up to make it blockchain agnostic. This entails that the proof in the form of the merkle root, can be published to any blockchain. However, the current implementation could only issue to the Bitcoin blockchain. Since that time, we have helped with the port towards Ethereum. Now, one can issue the proof of the blockchain certificates to both the Bitcoin blockchain and the Ethereum blockchain. This is since then being used on a day to day basis, as can be seen on the hardcoded address where the trials are issued to on the Ropsten Ethereum testnet: https://ropsten.etherscan.io/address/0xdeaddeaddeaddeaddeaddeaddeaddeaddeaddead.

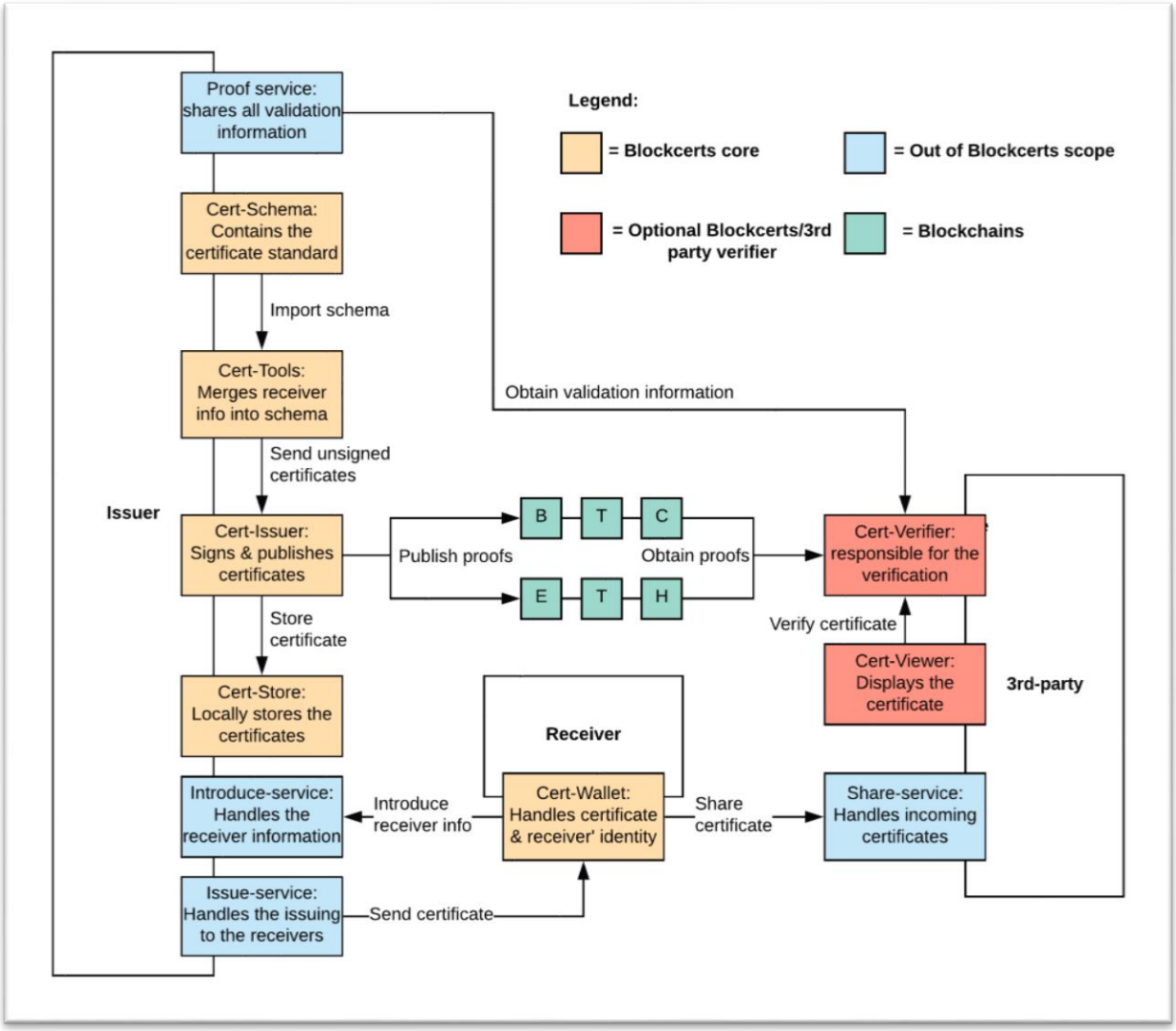


Figure 17: Technical architecture of the Blockcerts system.

4.2 Benefits and improvement opportunities to Blockcerts

Blockcerts already achieved quite a lot in the creation of a verifiable, decentralized, digital degree. Most importantly, Blockcerts introduced an open standard for the claims. This allows for any organisation to issue certificates while tools only have to create and accept a single verification method. Also, this enables developers to create tools for the entire standardized ecosystem.

Another benefit of Blockcerts is that their developers are actively working with and contributing to the Verifiable claims working group of W3C. They are making sure that the standard is compliant, and work together towards a generalizable solution. Creating a standard that others are willing to use is not something that can be done alone. Therefore, it makes sense to improve upon the Blockcerts standard in order to not reinvent the wheel.

There are three main improvement opportunities to the full Blockcerts system:

1. Blockcerts have no on chain sense of state, which has the main downfall that revoking is cumbersome. This is mainly because the current system uses Bitcoin for anchoring the proof, which is too expensive for a use case like this.
2. Blockcerts have limited protection against private key losses or compromises.

3. Blockcerts promote issuance on a degree basis, which makes finer selective disclosure not possible for the alumnus.

The above problems mean Blockcerts do not suffice for the user stories: US2 (revoking), US4 (selective disclosure), and US12 (private key security). In the design chapter we go into more detail regarding these three improvement opportunities and explain how the proposed design improves on these points.

4.3 Conclusion

There aren't currently any widely adopted technologies to issue higher educational degrees on a blockchain. However, there are some promising technologies, of which Blockcerts is the most notable. Their standardization efforts have not gone unnoticed, but their design is lacking to be fully functional for our use case. In our design proposal we cover how these issues can be tackled, while remaining Blockcerts compatible.

5 Verifiable degrees using smart contracts

In this chapter a proposal is presented for the system to issue verifiable degrees using smart contracts. We start by providing a brief overview of the design, and then the details of how the design covers each of the user needs are explained. We answer the following research question in this chapter:

How is the design of the degree publication and verification method structured in order to fulfil the identified user needs?

a. How does the method satisfy the user needs?
b. What are the steps for the degree issuing, attesting and verification processes?

This proposal is a visionary architecture of the issuing, attesting and verification methods for degrees at higher educational institutes. The design choices are highlighted throughout the chapter. It was designed in multiple iterations where each iteration was presented to experts. Here, only the final design choices are presented, and in the validation chapter we cover how some of the earlier design choices have changed due to expert interviews.

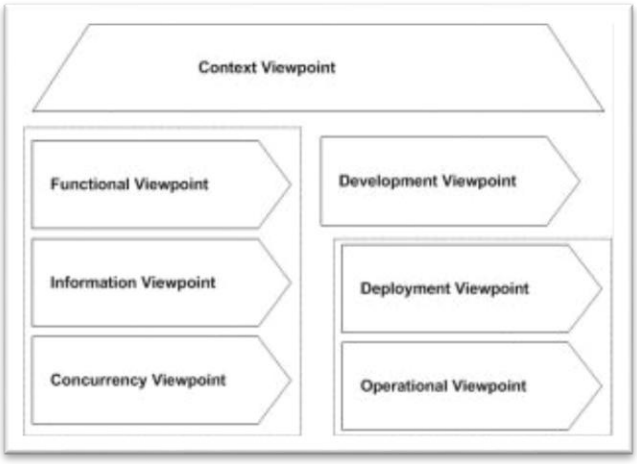


Figure 18: Software architecture viewpoints (Rozanski & Woods, 2012)

In order to provide a complete explanation of the proposed design, we acknowledge the widely accepted software architecture viewpoints model as shown in Figure 18. Throughout this chapter we cover the most important viewpoints of the proposed architecture. The context viewpoint has already been covered in the previous chapters, and will therefore not be delved into.

a. User stories – functional viewpoint
b. General overview – deployment & operational viewpoint
c. Detailed design of three key processes – information viewpoint
d. Design process of the on-chain degree registry – information viewpoint & operational viewpoint

5.1 User stories

In Table 5: User stories, with their origin and design location.Table 5 we restate all the user stories, and show where they originate and in which paragraph they are designed.

Table 5: User stories, with their origin and design location.

#	User story	Introduced in	Designed in
US:1	*As an educational institute's student administrator, I want to create and issue digital degrees from the student administration software, so that I minimize manual work and mistakes.*	2.6	5.2.1, 5.3.1

US:2	*As an educational institute's student administrator, I want to revoke earlier published digital degrees, so that I can mitigate fraud like plagiarism.*	2.6	5.2.1, 5.4.1
US:3	*As an educational institute's student administrator, I want to amend earlier published digital degrees, so that I can include missed credentials in an issued degree.*	2.6	5.3.1
US:4	*As an alumnus, I want to be able to assert any subset of my degree data, so that I can choose what to share about myself.*	2.6	5.3.2
US:5	*As an alumnus, I want to choose where to store my degree data, so that I can have full control over my degree data.*	2.6	5.3.1
US:6	*As an alumnus, I want to be able to retrieve or move the degree data at any time, so that I don't lose the data.*	2.6	5.3.1
US:7	*As an employer, I want to be able to accept digital degree assertions, so that I can verify these with less manual work.*	2.6	5.2.3
US:8	*As an employer, I want to instantly verify that the asserted digital degree belongs to the employee, so that I know the employee gave his own credential.*	2.6	5.3.2
US:9	*As an employer, I want to instantly verify that the asserted digital degree has not been altered or otherwise tampered with, so that I know I verified the original document.*	2.6	5.3.2
US:10	*As an employer, I want to instantly verify that the asserted digital degree was originally issued by the stated educational institute, so that I know which institute the employee is from.*	2.6	5.3.2
US:11	*As an institution I want to be able to update my issuing key so that I don't lose issue rights if I accidentally lose my private key.*	3.5	5.4.1
US:12	*As an institution I want to be notified if my private key is compromised, so that I can update my key to make sure others can't fraudulently use it.*	3.5	5.4.1
US:13	*As a receiver I want to be able to update my receiving keys so that I can control my data in the case that my keys are lost or stolen.*	3.5	5.4.1

These user stories are mapped to the user needs as identified by W3C credentials community group in Figure 19. Every user need is covered by at least one user story. The design proposal implements all these user stories and therefore indicates the completeness of the design. The user stories 11-13 do not implement an identified user need, but rather protect the issuer's and receiver's data in this decentralized solution.

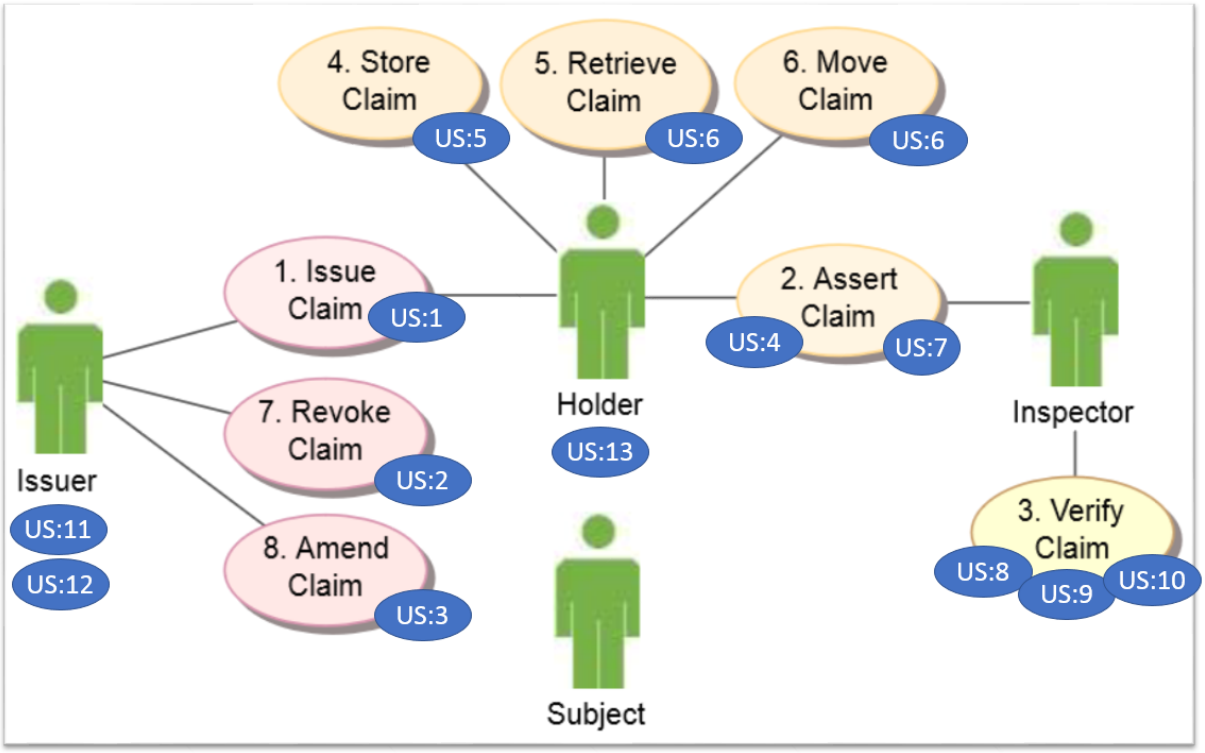


Figure 19: Mapping the user stories to the W3C user needs.

5.2 General overview of the design

In Figure 20 an abstract overview of the deployed architecture is visualized. The system consists of three applications, one for the higher educational institutes, one for the alumnus, and one for the employer. These applications can all communicate with the *public Ethereum Blockchain.*

Design choice 1: Use the public Ethereum blockchain as smart contract platform.

We have chosen to build with smart contracts on top of the public Ethereum blockchain. This choice was made as Ethereum has by far the biggest developer following of all the existing smart contract platforms on the planet right now. Therefore it is the most well documented public chain, and is also supported with many tailored developer tools.

We also promote to use the public Ethereum main network instead of setting up a blockchain by the issuing parties. In doing so, the designed method will also be usable on a permissioned ledger, as permissioned chains can be configured to higher performance needs than a public blockchain is capable of. The public Ethereum blockchain also has inherent incentive to keep the platform running for an indefinite period, which is much harder to achieve for a permissioned blockchain. This has the advantage that any issuer can join and leave at any time, while their degrees will be verifiable for as long as Ethereum exists.

5.2.1 Electronic Degree Issuer application

The higher educational institute already own a study information- administration system where all the grades for all their students are stored. From those applications, the degrees are currently exported to a printable format. In our proposal this export should lead to the *Electronic Degree Issuer* application. This application contains all the functionality for the degree issuer: it transforms the degrees into a universal standard, publishes proof on the Ethereum blockchain, sends the degrees to the alumnus, and has the ability to amend and revoke earlier issued degrees.

5.2.2 Electronic Degree Viewer application

The alumnus can visualize the received degrees in an *Electronic Degree Viewer* application. This application reads the standardized degree format, and verifies the validity through the published proof on the blockchain. The alumnus also has the ability to revoke the proof on the blockchain if they wish to do so. Finally the alumnus have the ability to attest their degrees to employers.

5.2.3 Electronic Degree Verifier application

Medium sized employers, and enterprises can have an applicant tracking system to help their recruitment department. That system is complemented by an *Electronic Degree Verifier* which can accept the attested degrees of their future employees. This application verifies the attestations by using the provided information to find the corresponding proof on the blockchain. There is no need for a built in link to each issuing higher educational institute required in this way, as the blockchain is used for building a proof registry.

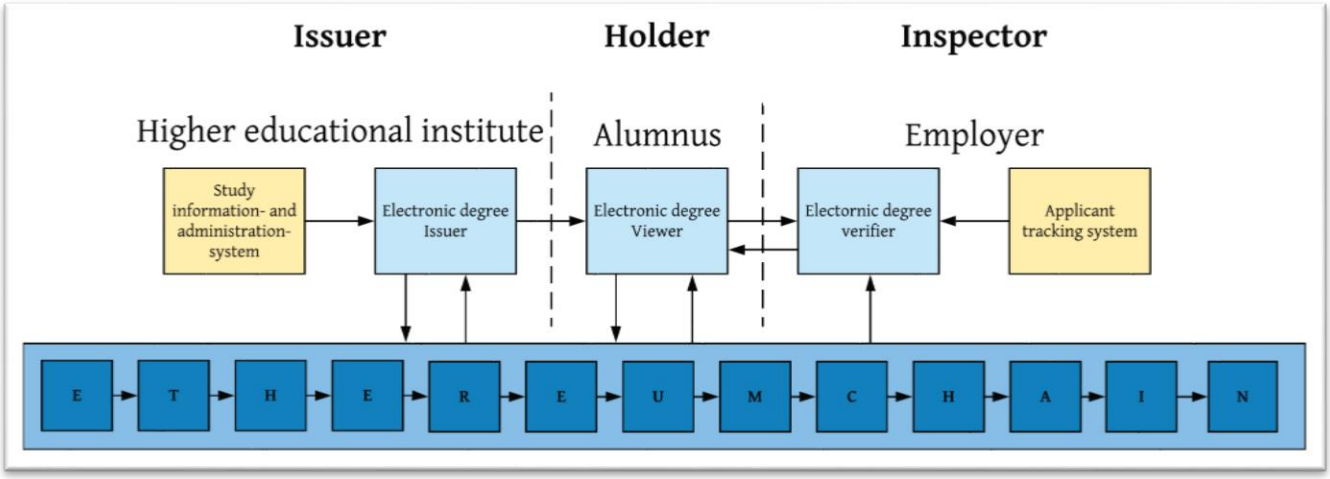


Figure 20: Deployment of the applications

5.3 Zooming in on the issuance, attestation and verification

Next, we cover the issuance, attestation and verification in more detail. The steps that the applications make for issuance, attestation, and verification are visualized in

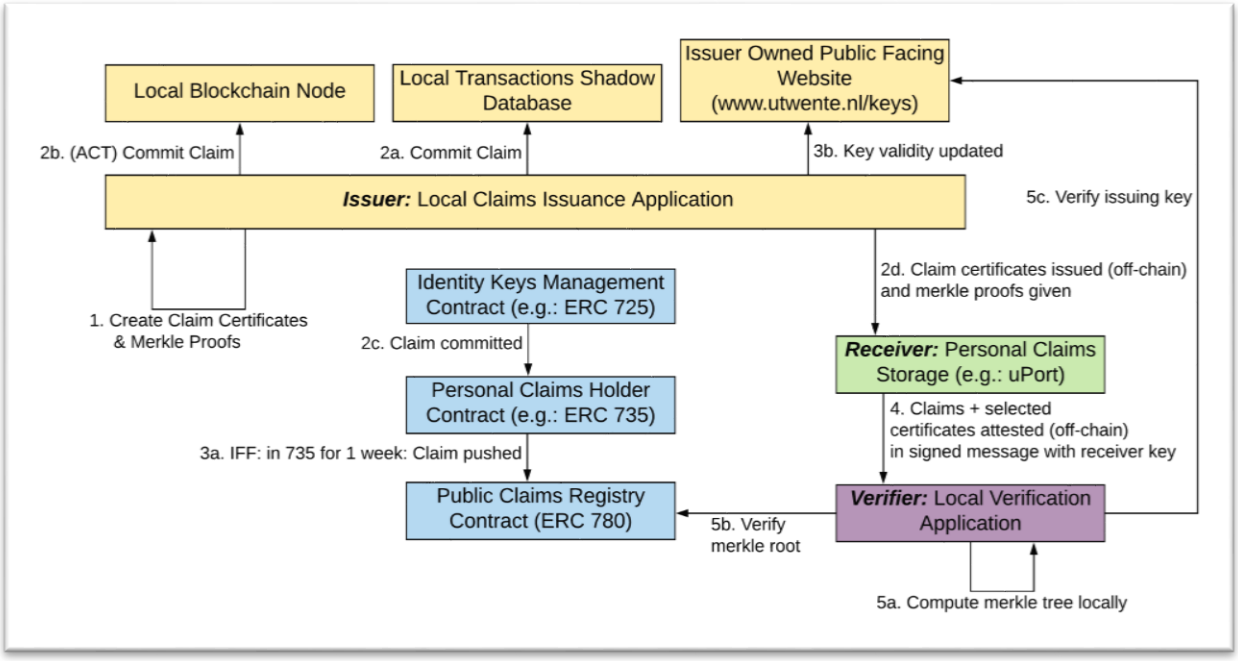


Figure 21.

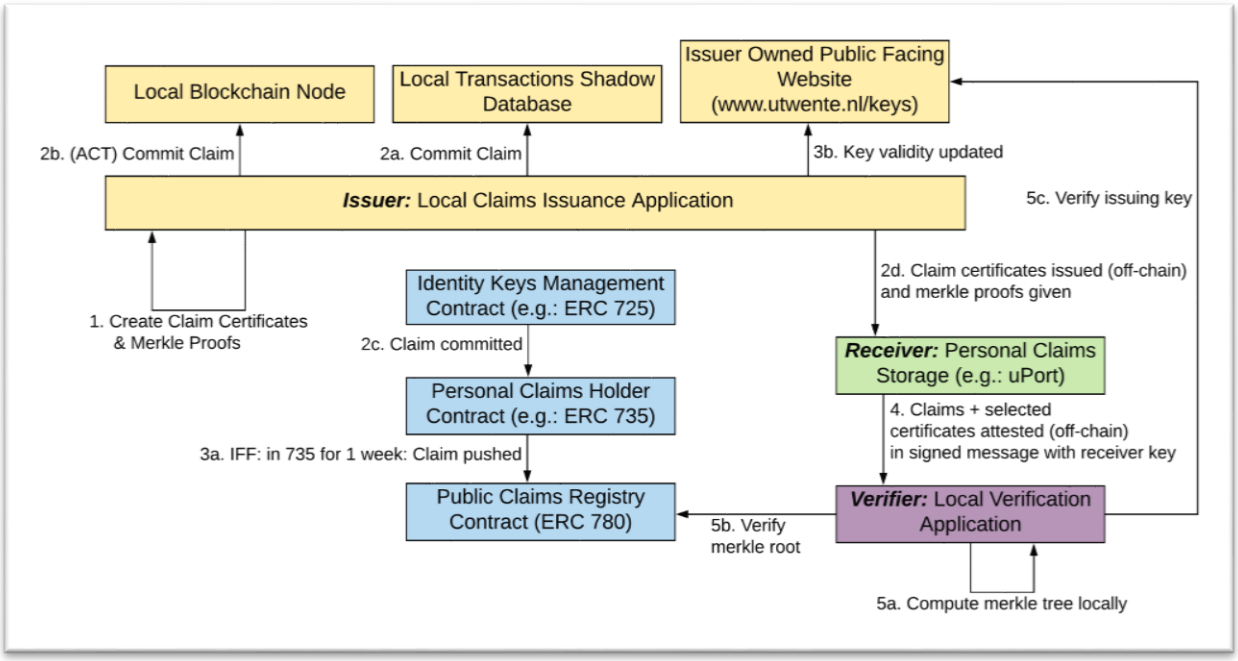


Figure 21: Steps to issue, attest, and verify degrees

5.3.1 Certificate creation, and issuance

The first step of the issuance process is to create the certificates and Merkle proofs.

Design choice 2: Use the Blockcerts standard, but create one certificate for every course and other credited event. Combine the Blockcerts into a Merkle tree. The degree is the entire combination of all these Blockcerts.

We have chosen to continue with Blockcerts, as we believe their open standard route is an inviting path for higher educational institutes. The Blockcerts standard also allows for pictures to be included in the certificates, which provides the institutions with branding opportunity.

However, we do promote to slice a degree in all the accredited bits it consists of, so each course is an individual certificate. These are then combined in a Merkle tree, which minimizes the required proof to be publicized to a single hash as can be seen in Figure 22.

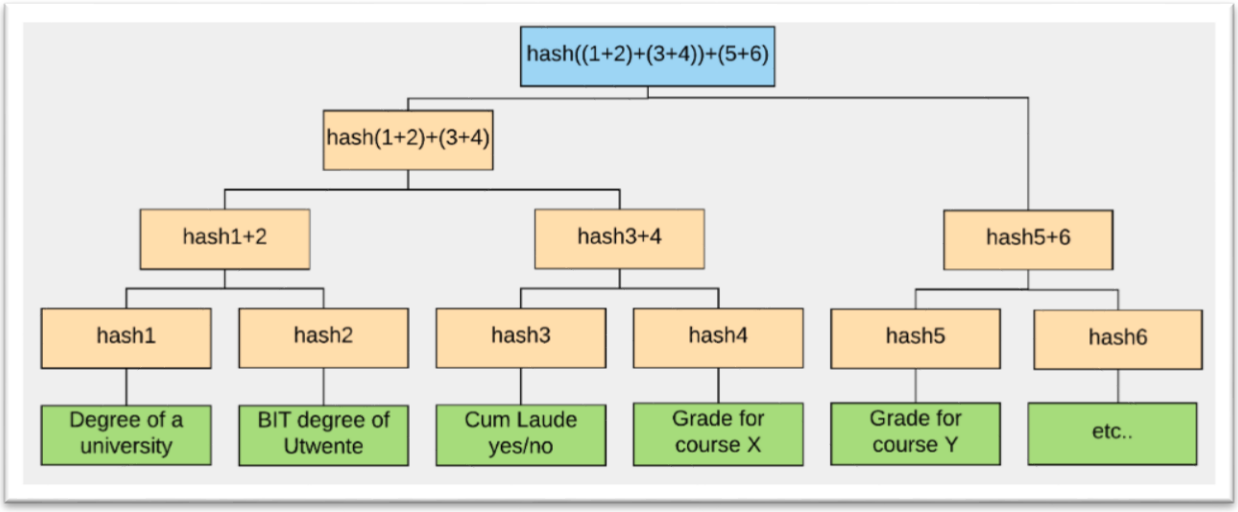


Figure 22: Merkle tree of Blockcerts for a single alumnus

The splitting of the degree in its parts allows an alumnus to selectively disclose their achievements. One might not be satisfied with the grade of course X, but may find pride in course Y, and as such can choose to attest only the grade of course Y. A proposed view of the alumnus application with the selective disclosure functionality is shown in Figure 23.

The standardized Blockcerts allows developers to create clients themselves to view, attest, or revoke the degree parts. The storage location of the Blockcerts may differ per application. Some applications will store the degrees locally, while others may store them in a cloud solution. By taking advantage of the universality of the standard, the end-users can move to whatever application provides the best usability for them.

Figure 23: Proposed view of the alumnus application

The next step in the issuing process is to commit the degree. The degree is first send to a local shadow database, and then send via a local blockchain node to the Ethereum degrees registry contract.

Design choice 3: Use an implementation of ERC780 as finalized degree registry. Use a personal in between registry for security.

We propose to use an implementation of the ERC780 smart contract as finalized degree registry. The reason is that it is kept very simple, while being scalable to hundreds of millions of registrations. By using a standard for the interface of the smart contract, anyone can create applications that can easily read from the same registries. So groups of universities could chose to work with different registries, while being all compatible to a single interface for verifiers.

The personal registry is used to store the committed degree for one week, before being pushed to the finalized degree registry. In the attacks & mitigations section we will further explain the reason for this design choice.

Design choice 4: The only data stored on the chain for about the degree is: (1) the public key of the issuer, (2) the public key of the receiver, (3) the degree title given by the issuer, and (4) the Merkle root of the underlying Blockcerts.

To issue a degree a transaction is being committed according to Figure 24. The transaction is send *from* the Ethereum public key of the issuer, *to* the location of the degree registry contract, with the *addDegree* call. This call commits the *Merkle root* of the degree, as explained above, with the *title of the degree* and the *public key of the alumnus.*

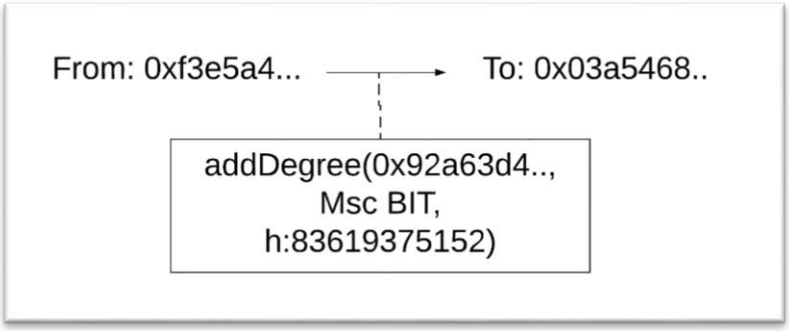


Figure 24: Issuance transaction to the Ethereum degrees registry

As the Merkle root is transacted to the blockchain, the Blockcerts are send off-chain to the alumnus. This can be done through e-mail, or for example by handing out a USB-stick at the degree ceremony. The higher educational institute could even host a service where alumni can log-in and download the Blockcerts.

Amending the degree can be done in one of two ways. Option one is to use the updateDegree function of the contract, with which the Merkle root can be updated of a degree receiver to include newly created Blockcerts. However, this entails that the previous issued Blockcerts all need to be updated to include the new Merkle root. The other option is to only issue the Merkle root of the amended Blockcerts through the standard issuing route. Doing so creates an extra entry in the on-chain degree registry. The advantage of this route is that the previously issued Blockcerts can still be verified. This design proposal does not restrict one of either options, as higher educational institutes have different preferences on how to deal with this type of situation.

After the proof has been a week in the personal claims registry, the degree can be finalized towards the final registry. At the moment the degree is finalized, the validity of the issuer keys are updated to reflect the newly issued degrees. This is put in a JSON-LD that follows the issuer specifications by Blockcerts, and hosted by the issuer on an external facing website. This is further explained in the verifying process.

5.3.2 Attesting and verifying processes

Now, the alumnus is full owner of their data, and can start selectively disclosing their information. For example to an employer. To disclose the information, an attestation according to the lines of Figure 25 is send to the employer.

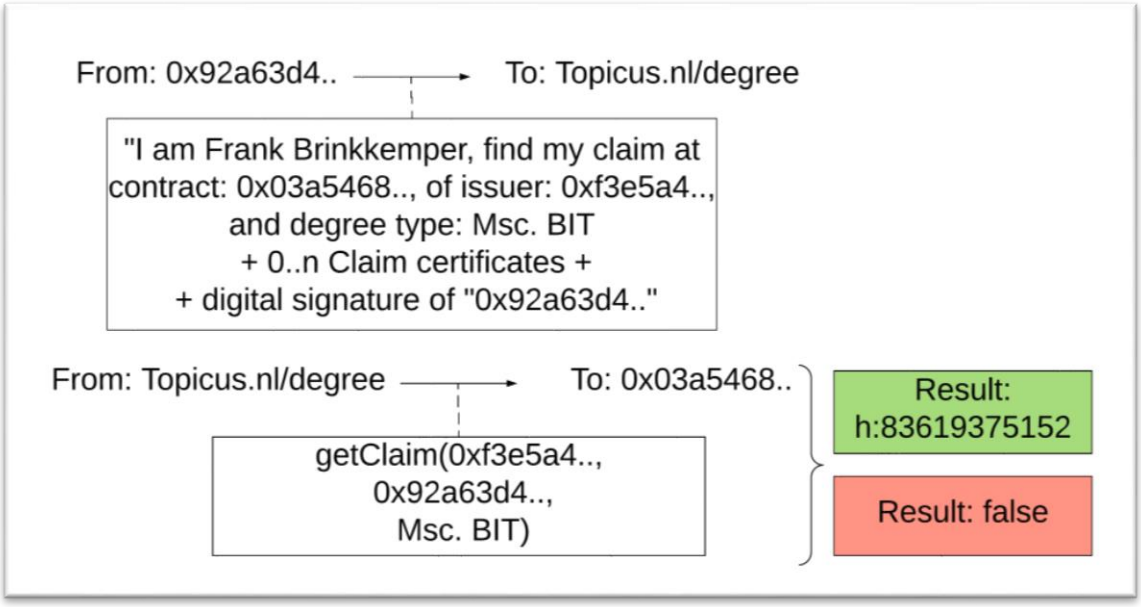


Figure 25: Attestation to an employer (in this case Topicus), and verification.

Design choice 5: An attestation contains: the name of the degree owner, the location of the Ethereum degree registry, the issuer's public key, the degree type, and 0..n Blockcerts including the routes to the Merkle root. The attestation is digitally signed by the alumnus before it is send to the inspector.

The inspector needs this information to perform the verification process. The verification performs the following checks:

- Does the supplied information lead to a returned Merkle root from the Ethereum registry?
- Does the issuing key match to a valid key on the issuer hosted website?
- Does the receiving name and public key in the Blockcerts match the provided name and public key?
- Can the Blockcerts be hashed to form the Merkle root included in the Blockcert?
- Do the included Blockcerts' Merkle roots match the Merkle root from the registry?

Finally the inspector needs to do two manual verifications:

- Is the name of the person I am verifying the same as the name provided by the alumnus?
- Is the issuer an issuer I know/trust?

Then the inspector is certain that the provided Blockcerts are issued by the party they know, and are indeed owned by the person they are verifying.

5.4 Step-by-step design process of the on-chain registry

In this part we go over the design process for the on-chain degree registry. As the proof in the smart contract ultimately declares the validity of the degree, the proof on the blockchain should at all times reflect the actual state of the degree. To achieve this, we developed the method step by step, adding functionality in each step. In this section we visualize each step, and explain the reasoning behind the design.

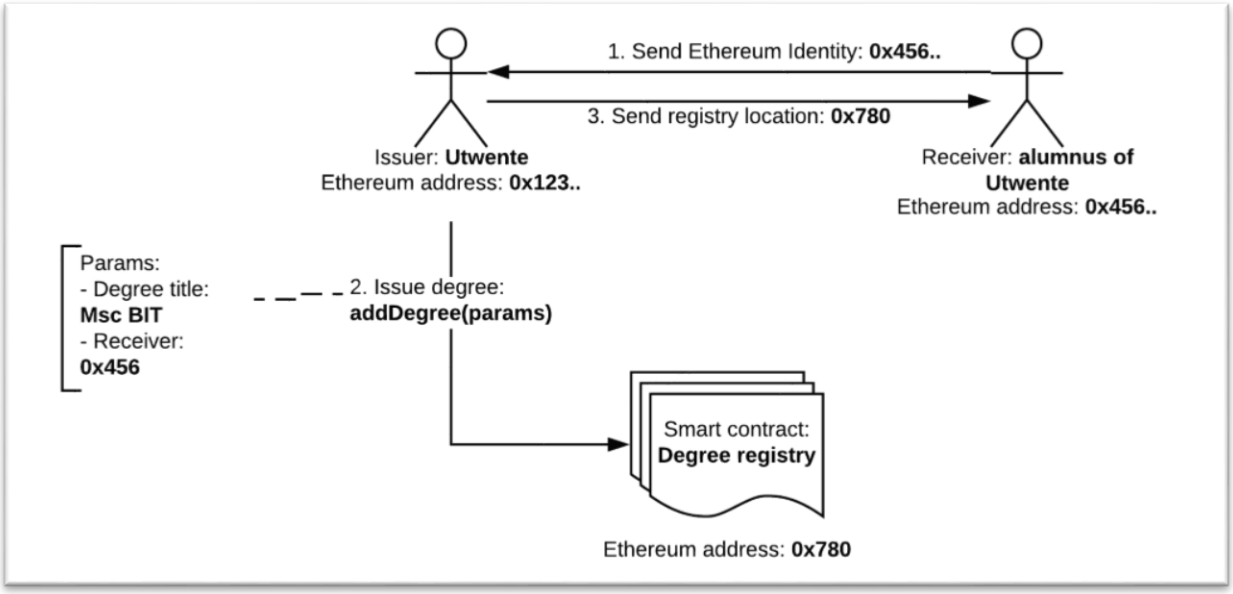


Figure 26: Issue a degree to the on-chain degree registry

The first step is visualized in Figure 26. In this first step we provide the issuer with a place to store the degree by publishing the Ethereum claims registry (Joel Torstensson, 2017). The issuer first requests the Ethereum address to the receiver, after which the issuer publishes the claim of the achieved degree to the registry, linking it to the receiver's public key. The issuer then sends the location of the registry to the alumnus, so that they can verify the publication.

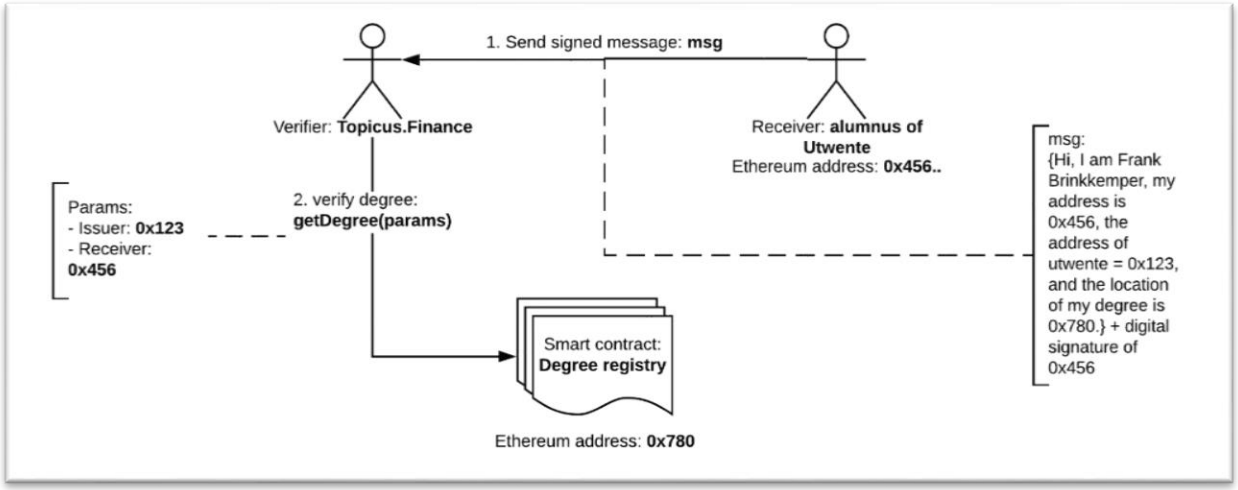


Figure 27: Attest the degree for verification.

The next step is to enable the receiver to verify the degree to others, shown in Figure 27. The receiving alumnus uses the information received from the issuer to attest their degree to a verifier. To do so, the receiver needs to digitally sign the information with the same keys they received the degree to. The verifier uses the retrieving functionality to verify the received information with the proof in the smart contract.

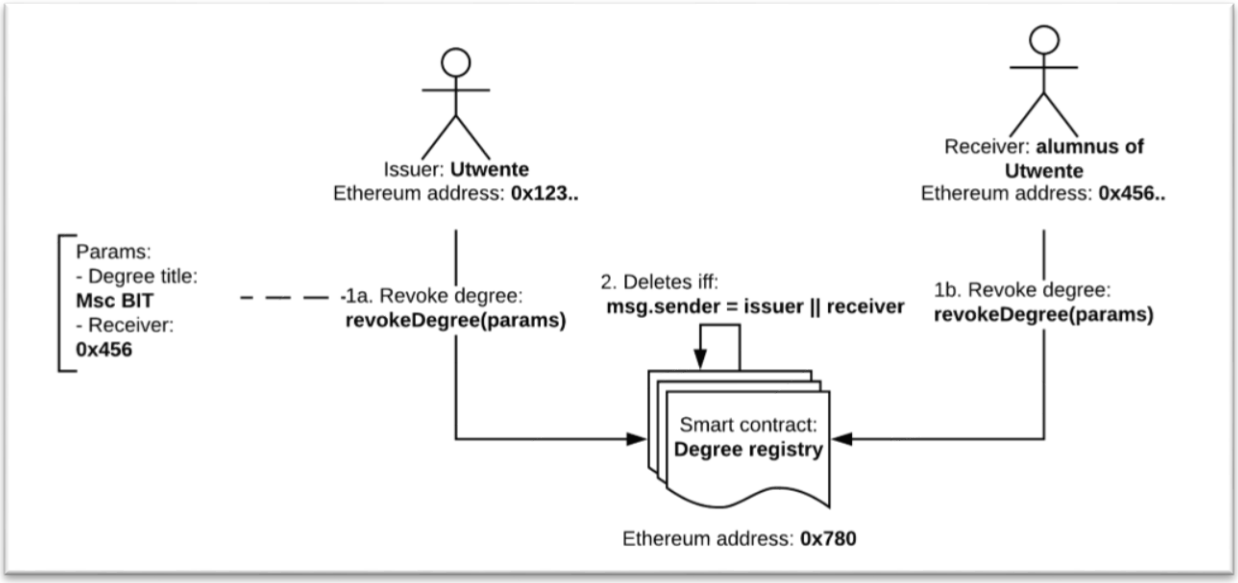


Figure 28: Revoke the validity of the degree.

As stated in the introduction of this part, the validity of the degree depends on the validity inside the smart contract. Therefore, the validity of the degree should be revocable by the issuer if they question the truthfulness of the achievement. The receiver may also revoke the validity of the degree, if they no longer want to be recognized for it. These revoking transactions are only executed by the smart contract if their origin is either the issuer or the receiver.

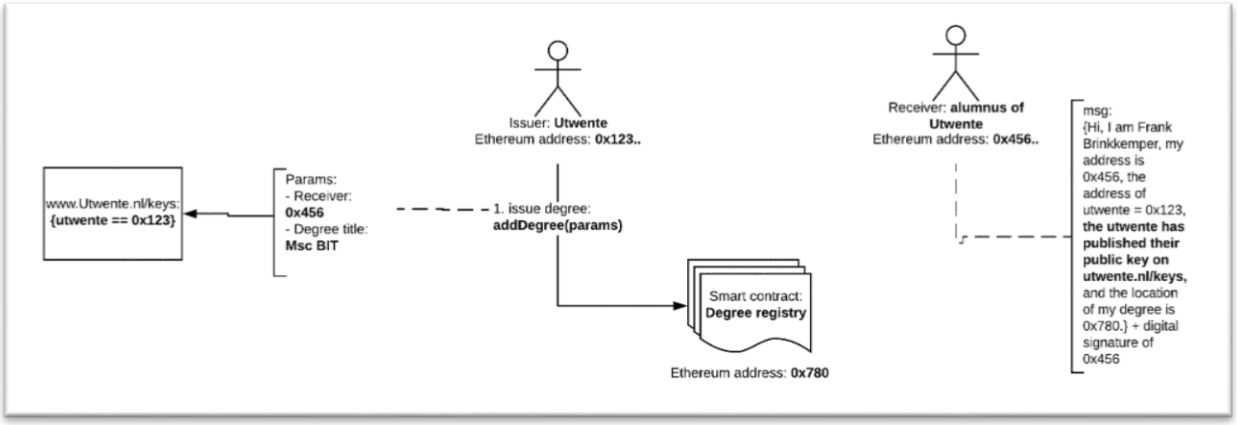


Figure 29: Issuer identity should be verifiable.

Although the receiver could attest they received a specific degree title from an Ethereum account, the verifier needs a method to verify the identity of this issuing Ethereum account. As the public Ethereum network is used, any entity can issue degrees. To enable the verification of the identity of the issuer, we propose to follow the Blockcerts standard method: utilizing the issuer hosted website, where the public keys are linked to the higher educational institute (Figure 29). This makes the issuer the ultimate trust anchor. Other solutions, by delegating the identity verification to another party like a certificate organisation, reduce the decentralization characteristic of the solution.

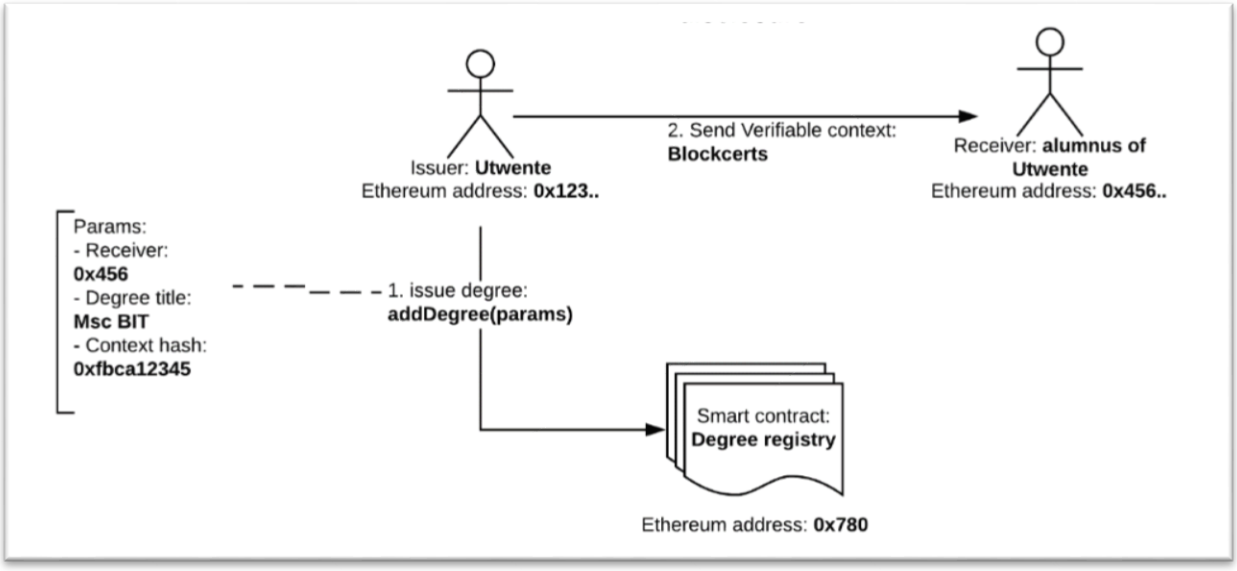


Figure 30: Supply verifiable context to the receiver.

In the previous steps, all the information stored on chain about the degree was merely the degree title. Often, this might be the only piece of information one might want to attest about themselves. Also, it is the only information that is public about someone. However, in other cases, the verifier might request more information regarding the degree, like followed courses or grades. Therefore, we standardize this information in Blockcerts as explained in paragraph 5.3.1. The Merkle root of these Blockcerts is then published with the degree title to the registry. Now the Receiver can verify any subset they please of the supplied Blockcerts by attesting these to verifiers.

5.4.1 Ensuring the security of the on-chain registry entries.

The aforementioned design proposal suffices for the normal circumstances. But to be production ready, there are some extra requirements to the design as introduced by user stories 11-13. These user stories try to tackle some solution specific problematic scenarios. In the current existing solutions, the publication of valid claims is only secured by the private key of the issuer. Thus the assumption is

made that this private key is never compromised. In this section we cover the rest of the design proposal which is made to mitigate this issue.

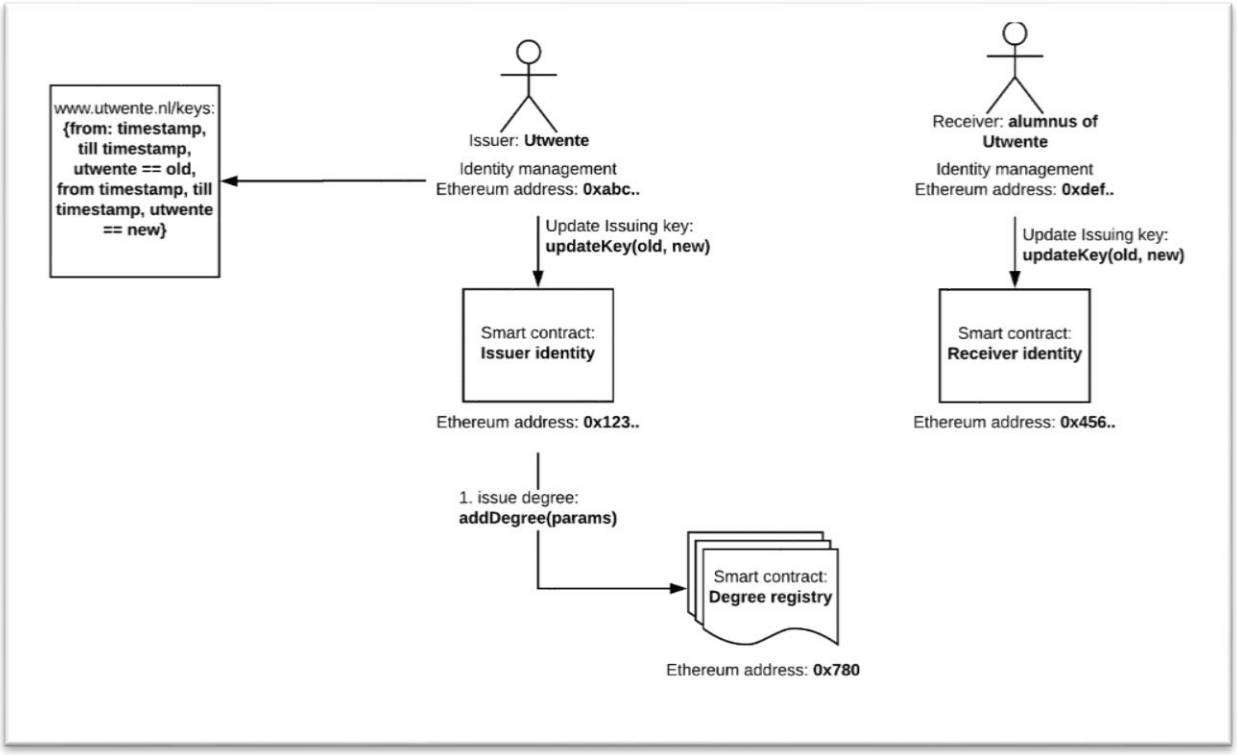


Figure 31: Enable updating the issuer and receiver identities.

The first addition to the design is the introduction of a new smart contract for the issuing and receiving parties that control their identities. The exact implementation of these contracts is out of the scope of this research, and we refer the reader to the discussions on ERC725/735 and ERC1056 (Braendgaard & Torstensson, 2018; Vogelsteller, 2017a, 2017b). However, we do supply a few requirements for these identity contracts.

We introduce the identity contracts in order to remove the reliance of the issuer identities on a single private key. This smart contract should at least have two types of keys: *issuing keys*, and *management keys*. The issuing keys are used in the day-to-day business to publicize degrees. Only in the case of a key loss, or (expected) compromise, the management key is used. With this key the issuing key can be updated, as in Figure 31.

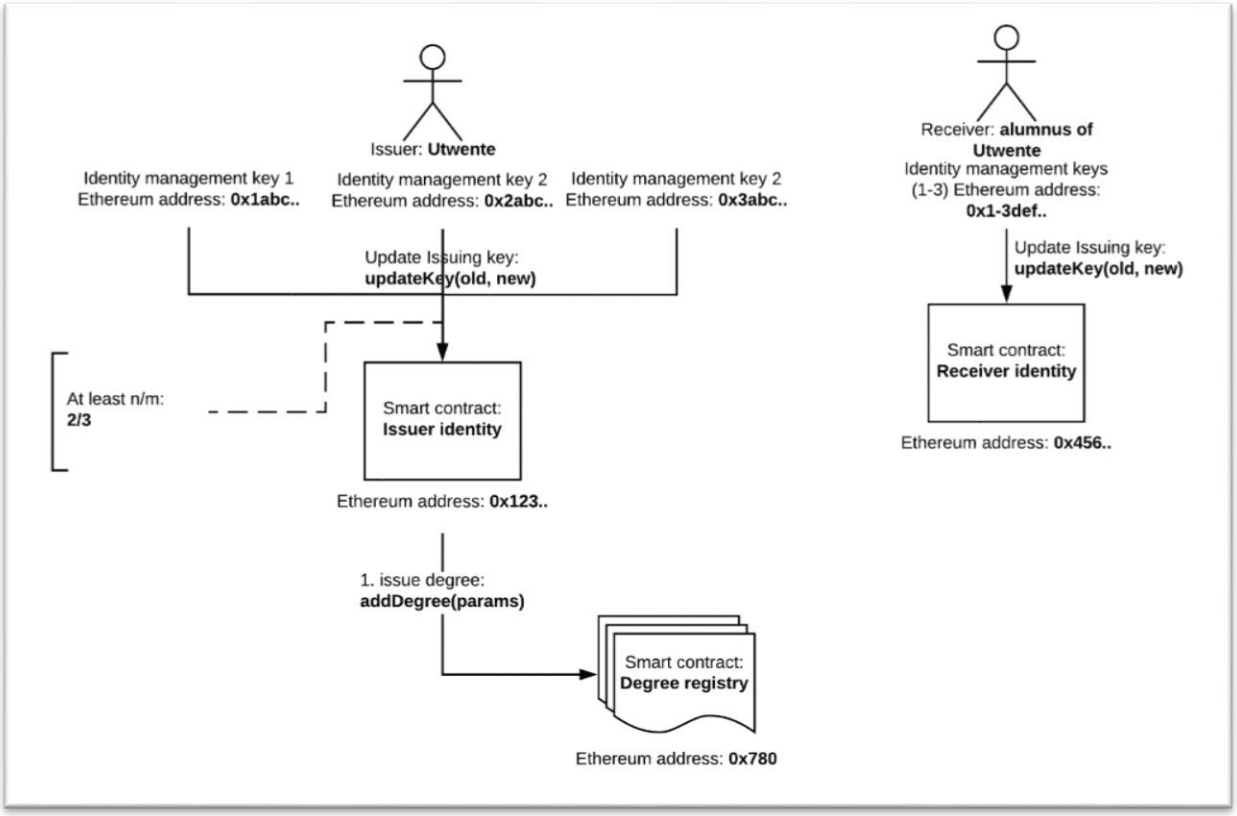


Figure 32: Making the management keys multi signature.

The splitting of the key purposes is a step in the right direction to make the design production ready. Nevertheless, the design now still relies on a single key, but this management key is used for far less operations, decreasing its key compromise chances. For many higher educational institutes the ultimate reliance on a single key might still not be secure enough. Therefore we propose the usage of a multi-signature management key structure in the identity smart contract, as visualized in Figure 32. With this mechanism, every action with a management key requires a chosen amount of *n* out of *m* signatures. The corresponding private keys should preferably be handled by separate people of the higher educational institute to mitigate the risk of multiple management keys being compromised simultaneously.

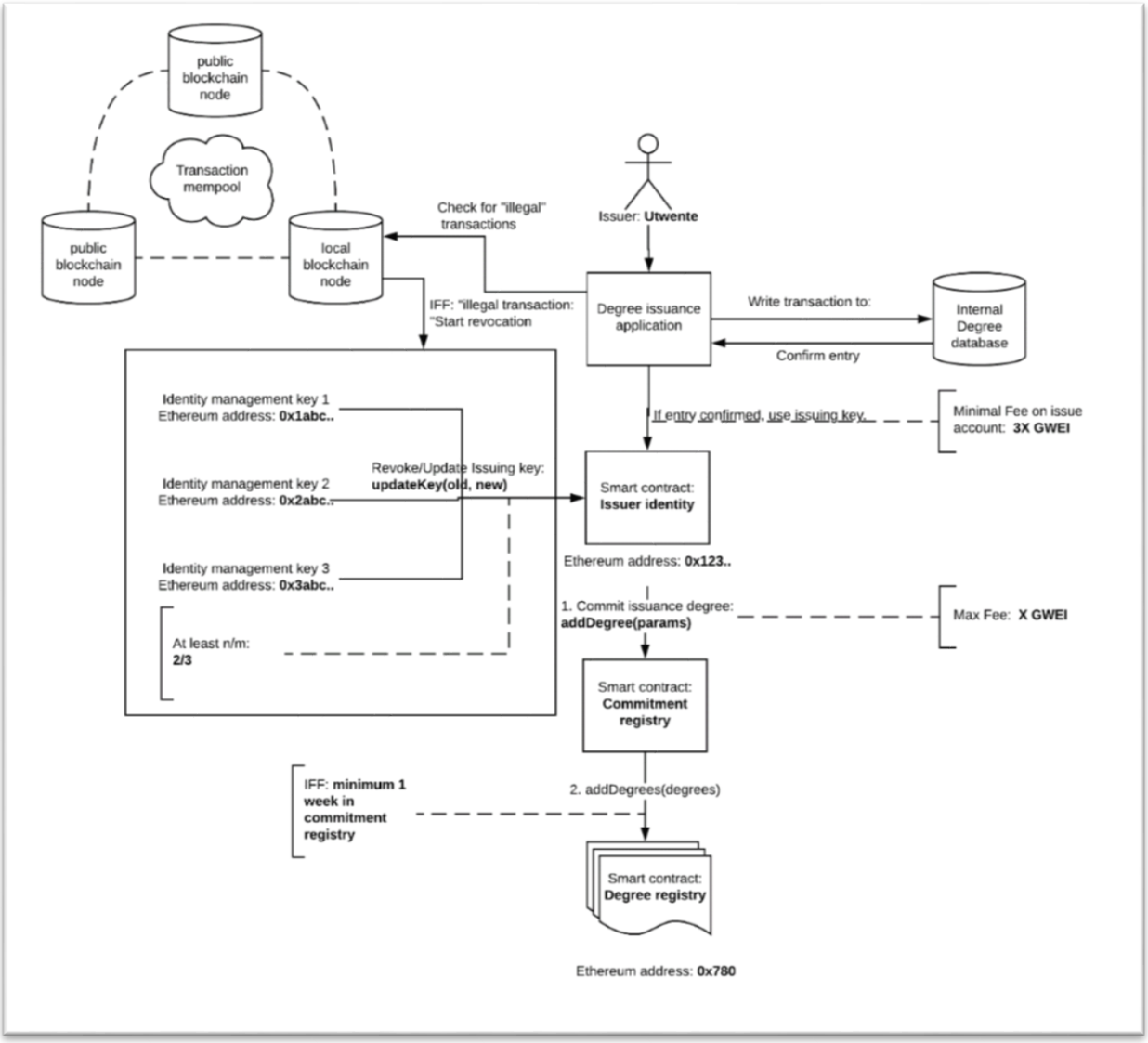


Figure 33: Securing the finalized degrees with a shadow database and an intermediate registry

Lastly, the higher educational institutes need to be notified if their issuing private key is used without their knowledge, so that they can revoke and update the key (US:12). To enable this, we propose the following additions to the design:

- Create an internal degree database to which every issuance, amend, and revoke action is first published.
- Use a commitment registry, to which the degrees are first published. Only if the publication has been committed for a week, the degree can be published to the finalized registry.
- Let the degree application continuously listen to an Ethereum node, preferably local, to check if any of the incoming transactions is an unapproved transaction from the issuing key.
- When there are unauthorized transactions of which the corresponding action is not first committed to the internal database, the revocation process of the issuing key is started.
- To minimize the losses in the case of an issuing key compromise, a maximum fee is set on the issuing function in the corresponding smart contract. The issuing key should never contain more than a few times this maximum transaction cost, to prevent many unwanted actions in the intermediary registry in the case that an issuing key is compromised.

With these security mechanisms in place institutions can create production grade applications without the need to fear the consequences of a key compromise.

5.5 Conclusion

Starting with 13 user stories we have shown that these have all been fulfilled in the design proposal. We present a design for the applications of the higher educational institutional issuer, the alumnus, and the verifier that together enable verifiable degrees. We have zoomed in on the processes of issuing, attesting and verifying, in order to provide more details regarding the precise transactions that are to be performed both on the blockchain to the smart contract, and off-chain between the involved parties. We finalized this chapter by showing the step-by-step creation of the design process, in which the additional security measures are proposed. The complete design will be validated in the next chapter.

6 Validation

In this chapter we validate the design proposal. We do so by answering the following research question:

Is the created method a valid solution to the identified problems?

a. What are the possible attack vectors, and how can they be mitigated?
b. Is the method valuable in other document verification areas?
c. Is there an alternative design without blockchain?

6.1 Introduction

There are several strategies for validation in design science research. In this research we used the framework for evaluation in design science research (FEDS) to choose a fitting strategy. (Venable, Pries-Heje, & Baskerville, 2016). The most important risk in the design proposal is the correctness of the techniques used to reach their intended goals. Testing the complete proposed design in a real environment was not deemed possible due to the limited resources available. Based on these circumstances the most fitting strategy according to the FEDS is the technical risk & efficacy strategy.

The goal of applying this strategy is to reduce the technical difficulties before the actual implementation. To reach this goal, we have analysed the design proposal with several experts. Both from a technical perspective, and from the higher educational field. With these experts some earlier design choices were presented. The feedback from these interviews has been used to improve the design, and eventually validate it.

Throughout this chapter we answer the following three questions for validation of design science (Wieringa, 2014):

1. Does the designed artefact satisfy in the defined context? (see 6.2)
2. Would another design satisfy in this context? (see 6.3)
3. Would the designed artefact work in different contexts? (see 6.4)

For the first we analyse it by (1) looking at possible attacks and mitigation strategies to these attacks, and (2) by expert interviews. The second question is solely discussed in an expert interview, and the final question is answered by building a prototype of the design proposal in a different context, namely the verifiability of the employer's statement.

For the validation the design has been presented to:

1. Nick Johnson, lead developer of the Ethereum Name Service (ENS) (https://twitter.com/nicksdjohnson)
2. The Open Badges PoC group at Surf, the collaborative ICT organisation for Dutch education and research (https://www.surf.nl/en/about-surf)
3. Tech lead of the blockchain team at Rabobank (www.rabobank.nl)
4. TNO researcher, and teacher of a blockchain course at the University of Twente
5. The blockchain specialists at Topicus (www.topicus.nl).
6. The Ethereum community at EthCC (www.ethcc.io) and on the Ethereum improvement proposal for the Ethereum claims registry (see appendix B) (Joel Torstensson, 2017).

6.2 Artefact in context

We start by analysing the artefact in the presented context. We have already illustrated that the proposed design implements all the user needs in Table 5. In this part we further analyse the design proposal in the context by looking at possible illicit behaviour in the system, and discussing the versions leading up to the design proposal with various domain experts.

6.2.1 Possible illicit behaviour and mitigations

Thanks to the increased security measures, explained in paragraph 5.4, higher educational institutions do not need to fear the consequences of key losses or key compromises. Here, we discuss four possible attack methods how an ill-willing entity could try to issue valid degrees with keys from others, and how these are mitigated in the proposed design. These attacks and their mitigations are based on four assumptions.

Assumptions:

1. Keys to issuer's internal degree database are not compromised.
2. The issuer's public facing website is not compromised.
3. At least n/m (2/3) of the issuer's management keys are not compromised.
4. There are no bugs in the used smart contracts.

With these assumptions in mind, we have analysed the below three possible scenarios where attackers try to issue illegitimate degrees.

Possible attacks:

1. Issue key is unknowingly compromised by a naive attacker.
2. Issue key is unknowingly compromised by a mining attacker.
3. m-n management keys are compromised.

Mitigation strategy to attack 1:

We define the "*naive*" attacker, as a non-mining attacker who tries to publish a false degree. We have visualized the mitigation to this type of attack in Figure 34. The internal issuing application of this higher educational institute continuously monitors the public Ethereum mempool for incoming transactions that use the issuing key of the institute.

If such a transaction is found, it is checked against the internal degree database. The internal issuing application will conclude that the transaction originated from an attacker if the corresponding action is absent from the internal degree database. The issuing application then starts the revocation of the issuing key with a management key. Moreover, the issuing application publishes a transaction with the same nonce as the attacker, but with a higher fee. If done quickly enough, mining nodes will prioritize the issuer's transaction over the attacker's transaction, as they prioritize on fees. In this way, the transaction gets overruled by the issuer (MyEtherWallet Knowledge Base, 2017).

Even if the attacker's transaction does get through, the issuer's key will get updated, and the issuer has a week to use management keys to remove the faulty degree from the commitment registry, before the wrong degree can get validated.

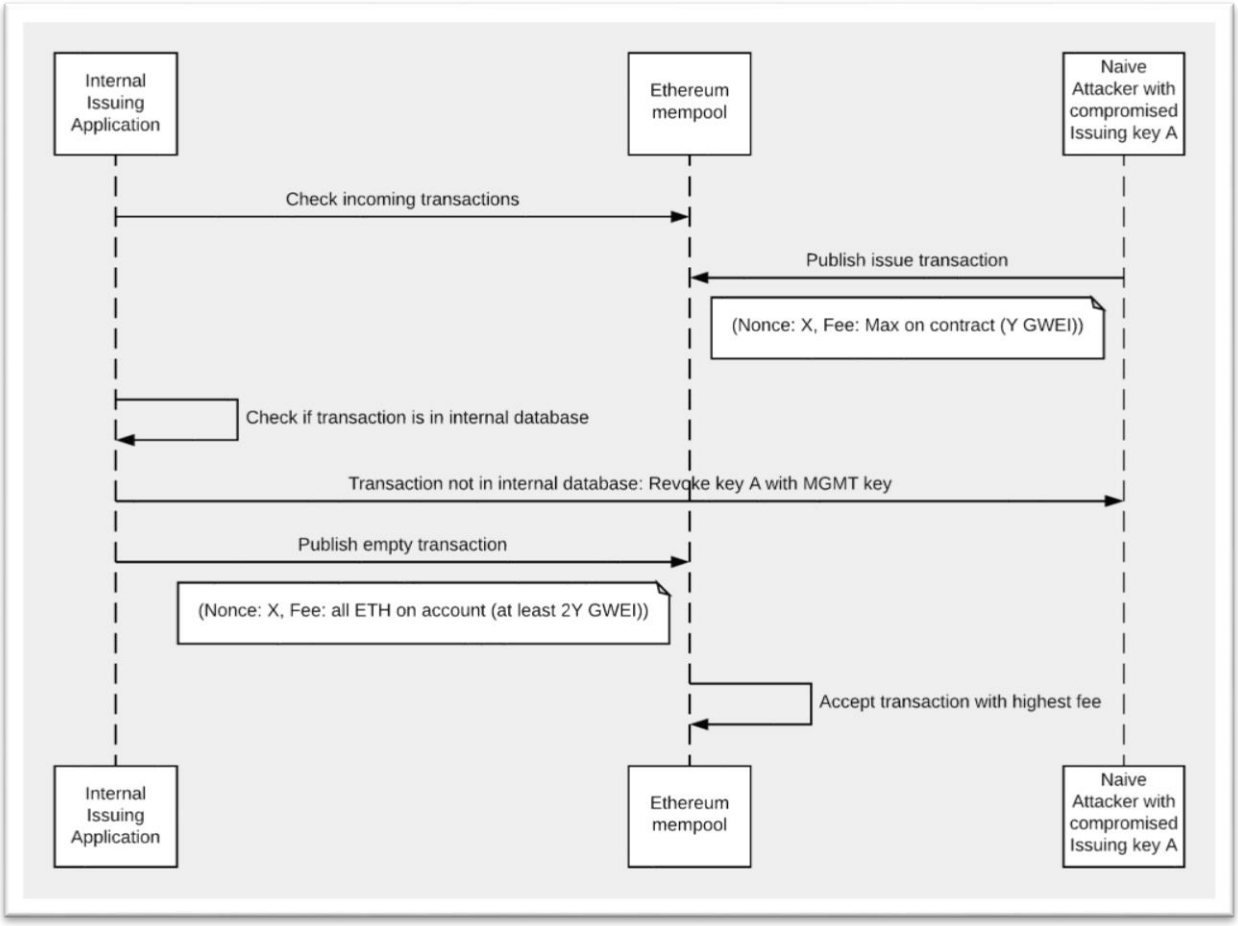


Figure 34: Sequence diagram of an attack by naive attacker with the issuing key of a higher educational institute.

Mitigation to attack 2:

In the case that a mining attacker has compromised the issuing key of a higher educational institute, the miner can include the faulty degree issuance transaction in a block without first making it public by propagating it in the Ethereum mempool.

Therefore, mining attackers can use the compromised issuing key to publish a faulty degree to the smart contract as visualized in Figure 35. The internal issuing application can immediately go ahead with damage control, by starting to revoke the issuing key with a management key, and emptying the issuing key. Finally, the commitment needs to be undone with the use of the management keys, so that the faulty degree does not end up in the final registry.

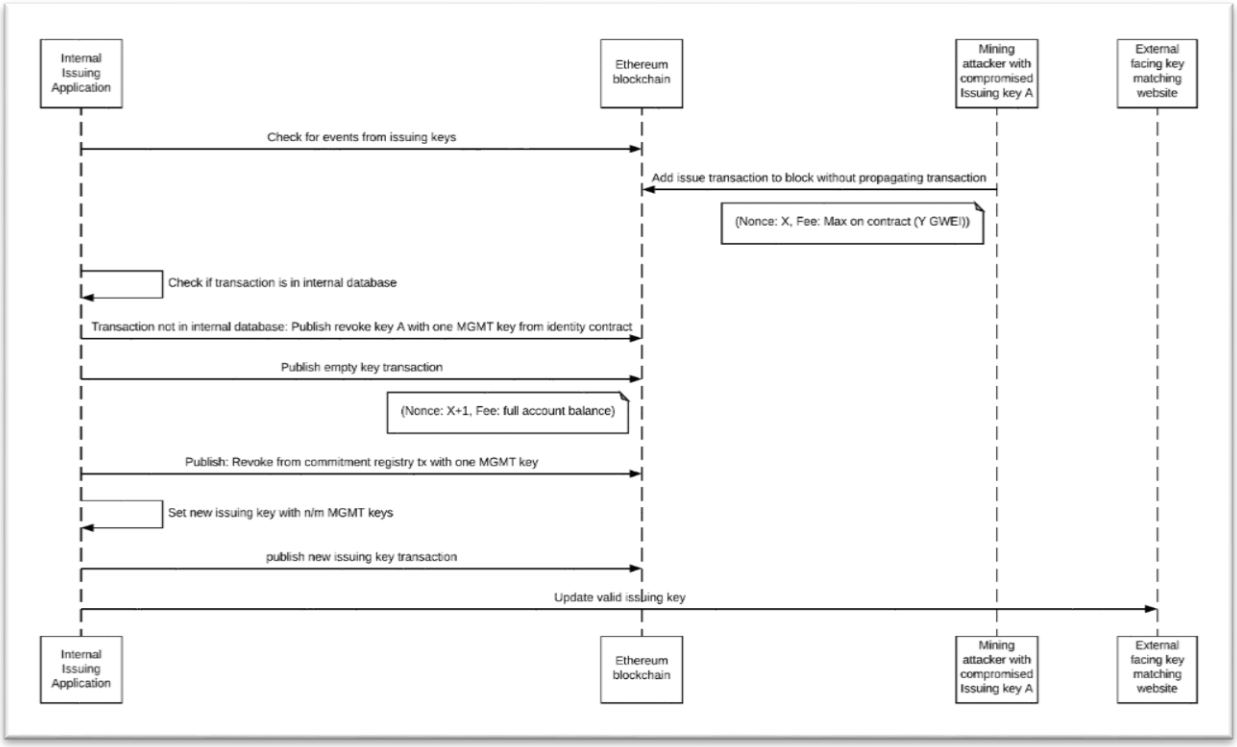


Figure 35: Sequence diagram of the scenario where a mining attacker compromised the issuing key.

Mitigation to attack 3:

If it is detected or suspected that a management key is compromised, the key should be immediately updated in the identity smart contract. With the single management key, the attacker won't be able to create a new issuer key, nor revoke degrees from the finalized contract. After revoking the validity of the management key with the other management keys, the higher educational institute can undo the minor actions that could have been done by the compromised key.

Remarks to these attacks and mitigation strategies

As long as the higher educational institute keeps at least the majority of the management keys private, no attacker can create verifiable degrees by that educational institute. In the scenario that the majority does get compromised the higher educational institute will need to set-up a new identity smart contract, from which the degrees need to be re-issued. The usage of these management keys is minimized to make sure these are far less likely compromised.

These mitigation strategies were evaluated with Nick Johnson, who commented:

"Internally, you can have this mechanism, which can make sure that the attacks can be mitigated with less overhead and hassle than that would be the case if all the claims need to be re-issued." And: *"I think the mechanism outlined there should be pretty effective, because the management keys are kept offline."*

However, he did note that this is indeed a mitigation strategy, and not integral part of the security model, as far as an external user is concerned:

"From the outside users' point of view: the external facing website is the trust-root. So whether or not you have this mitigation strategy for mitigating key compromises, the most important thing for a user is whether you keep in control of the external facing website. So even if you didn't have this

mechanism. If the attacker has got your keys, you could also issue new claims to everyone, and say on the external facing website: "this is our new address".”

We do agree that this is indeed the case, however the security of the higher educational institute’s own website should be their highest priority.

6.2.2 Expert Interviews

During the project I had several in depth discussion on my design proposal with several domain experts. Throughout these discussions there were multiple suggestions that we happily accommodated in the design proposal. We will here present the overview of the accepted suggestions for each individual expert.

Blockchain expert and educator: This expert commented that we should add an extra security measure in the Merkle tree of the Blockcerts. Neighbouring leaves in a Merkle tree that were kept private can be guessed as the schema is standardized, and the information could be brute forced. Therefore each leaf should get a neighbour with nothing but a salt hash. Then the selective disclosure method can succeed in its goal of keeping parts of the tree private. Furthermore he sharpened the added value of blockchain technology. What is, and what is not possible only thanks to blockchain and smart contracts.

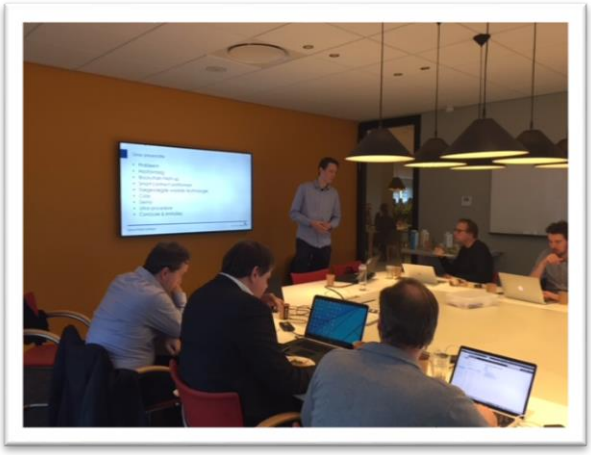
Nick Johnson, lead architect of the Ethereum Naming service: He gave multiple useful suggestions: In order to be able to retrieve the degrees-types with a standardized way, we changed the “key” value in the ERC780 registry to standardize the degree types. Before this discussion, we used the issuer hosted website as “key” value. He commented that we should remove this from the information inside the registry as it should not be hardcoded for every degree entry. Next to this, he noted that the mitigations are something that the system can do automatically. The user does not need to get involved. He also gave some interesting comments on the added value of blockchain technology and smart contracts for this use case, which can be found in paragraph 6.4.

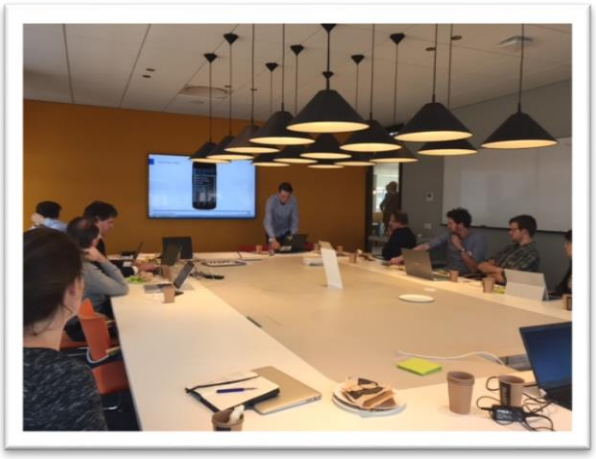
Blockchain expert at fintech company 1: In a first iteration, we did not include any information about the degree aside from the Merkle root of the Blockcerts inside the degree registry. This expert suggested to reduce the need for off-chain information. The claim itself should be enough to prove one has a degree, as that is already often public information.

Blockchain expert at fintech company 2: This expert suggested to issue to a new public key for every new claim someone receives. However, another expert who had tried this method commented that this could lead to public key explosion. Although this could be mitigated by using a mnemonic phrase to generate the keys. We decided not to include this in our design proposal for scoping reasons.

6.2.3 Validation with higher educational institutions

The proposed design has been presented at SURF, the institute for Dutch higher educational ICT collaboration. This institute is performing a proof of concept to use Open Badges at eight higher educational institutes in the Netherlands. We had the honour to present the proposed design for this group.





The group was generally enthusiastic about the use case. In their current proof of concept with open badges, the group questioned the ability to verify the badges for a period longer then the experiment, as the required proof is all stored locally. This in mitigated in our proposal, by storing the proof in the public Ethereum blockchain, which contains built-in incentives to keep running for an indefinite amount of time.

Another interesting comment was made by one of the representatives of the University of Maastricht. Their large increase in interest from international students has created an enormous strain on the admission office. The representative commented: *"universities themselves are probably the biggest group of inspectors verifying higher educational credentials".* They agreed a decentralized credential publication system like the one presented would massively benefit their admission process for international students.

There was one argument that the current design does not handle properly. The proposal links the proofs to a public key of the receiver, to give them the ability to revoke a credential about themselves. The argument was that this does not enable the participant with a true right-to-be-forgotten, as the original issuing and revoking transactions are still retrievable from the blockchain. We do not believe this to be a major problem, as the information stored on-chain is public information currently as well. Future work could delve into this problem, and try to add a right-to-be-forgotten mechanism with for example ring signatures or other advanced cryptography.

The SURF proof of concept group appreciated the proposal, and they are researching the possibilities for an advanced pilot that integrates this proposal. At the time of writing this was still under discussion.

6.3 Satisfaction of another design in this context

We have to answer whether all of this can be achieved without blockchain technology, but with merely decentralized public key infrastructure. Although a similar system could be created, this cannot be done with equal fraud protection and data availability. The smart contracts add a sense of state to the degree. In the proposed architecture, the degrees are only valid if the proof in the form of the Merkle root is in the not revoked state in the contract. This creates the advantage that an issuer can continuously, and automatically, check whether there are no fraudulent degrees issued with their keys.

In a similar architecture based on decentralized public key infrastructure without a blockchain an issuer could unknowingly have their private key compromised. This compromised key can then be used by a degree mill to issue degrees to anyone.

We also discussed the advantage of blockchain technology and smart contracts to this use case with Nick Johnson. He commented the following on the added value of blockchain:

"So, I mean, you can proof things with crypto, for example private keys, and merkle roots. What you can't do is revocation. And you can't prevent the equivalent here of double spending, where the issuer would sign one degree, and then later sign a different degree, and thereby altering your degree. And also that you can provide data availability here. Once you have provided the alumnus with the location in the blockchain and the merkle tree of documents, then they can always prove it independently without the cooperation of the university. Also when it goes out of business, or stuff like that. So I would say that the advantages here are non-revocation and the availability issue + the fact that it can be verified independently without the cooperation of the issuer."

And on the added value of smart contracts:

"It allows you to build more flexible ways of doing issuance and revocation. So regular public key cryptography, combined with merkle proofs, would not allow you to do such things as the failover mechanism against the attack, and they would make it more difficult to make nuanced permission and update structures."

For the remainder of this interview see Appendix B.

6.4 Application in different contexts: Verifiable Employer's statement

6.4.1 Explanation of the use case

As explained in paragraph 2.5, the degrees use case is an application of the more general use case of verifiable claims. Therefore, the design can easily be translated to other contexts. We have explored one of these other contexts at Topicus.Finance in a proof of concept. In this part the proof of concept is explained, and the benefits are discussed.

At Topicus.Finance, one of the focus areas is providing software for the mortgage industry. The software helps automate the application process for mortgages. However, the current process still takes weeks to fully complete. One of the major time-consuming steps in the mortgage process is the verification of the required documents.

To prove to the mortgage supplier that an applicant will be able to pay-off their monthly charges, an employer's statement is requested. This document needs to be filled in by the applicant's employer and contains information on the salary, and the intention to keep the applicant as employee. The document is then signed by the employer, and handed to the employee. The employee then attests the statement to the mortgage supplier.

The employer statement is currently refused 90 percent of the time on the first assertion by mortgage suppliers (Olivier Tardieu, 2017). This enormous percentage is in a significant part due to the dated verification method of the document. The mortgage supplier currently verifies the employer's statement by comparing the colour of the ink of the signature to that of the filled in salary. Verifying the claim by manually contacting the employer directly is too time-consuming and therefore too expensive for the mortgage supplier. Thus, applicants are asked to request an improved employer's statement that do pass the arbitrary security measures like the ink comparison.

6.4.2 Applying the design to the use case

The mortgage suppliers want certainty about the data issued in the employer's statement, without the need for direct integration with the employer. The decentralized architecture of the proposed design, provides benefits for all the stakeholders in this use case.

- Employer (Issuer): Remains in full control over the issued statements. As the valid statements are always visible in the blockchain, the employer can control that no false statements can be validated. Also, as the statements are standardized and digitized, it is much easier to fill in according to the demands of the mortgage supplier. Therefore it should take less time on average to help an employee get through their mortgage application.

- Employee (Holder): Remains in full control over the data ownership. They can attest to as many parties as they like, and with selective amounts of information. One issued employer's statement can therefore be used to get access to more services than just a mortgage, like a private car lease.
- Mortgage supplier (Inspector): Can audit the authenticity of the data, and verify its origin in an automated process. They do not need to rely on the arbitrary security that ink signatures provide.

Standardizing this process utilizing the design proposal will generally lead to less human errors in the verification process.

6.4.3 The proof of concept

A small proof of concept was created in which part of the proposed design implemented. The proof of concept for the issuance and verification of the employer's statement was created to get a better understanding of the benefits, limitations and the general implications of the implementation in practise in the mortgage industry.

In Figure 36 and Figure 37 the created front-end application for the employers is shown. For now, the employer only needs to fill in the employee's name, its Ethereum public key, and the yearly salary. This statement can then be signed by the employer, after which the Merkle root is notarized in the registry contract. This front-end could be integrated in existing human resources software, so that the statements can be issued in a single click.

Then the employee can digitally attest the information to the mortgage supplier, who can verify the information as shown in Figure 38 and Figure 39. This view would normally be integrated in the mortgage application software. As the correct information has been attested to the mortgage supplier, the attestation is verified by locally computing towards the same Merkle root.

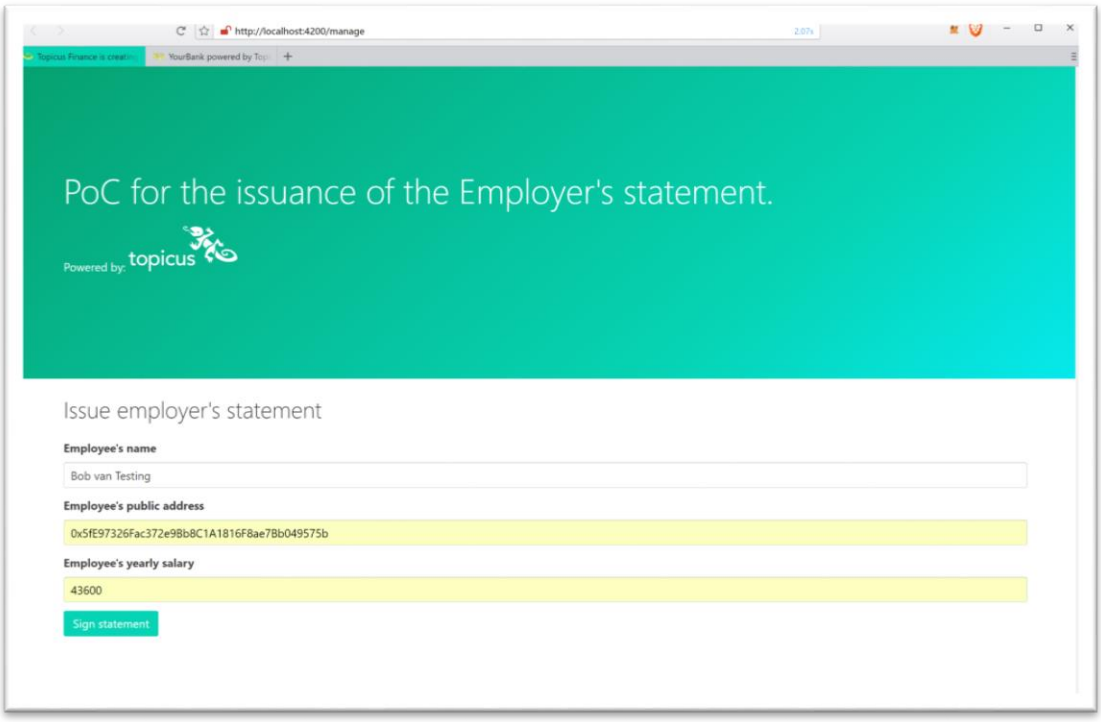


Figure 36: Front-end for employers to issue employer's statements

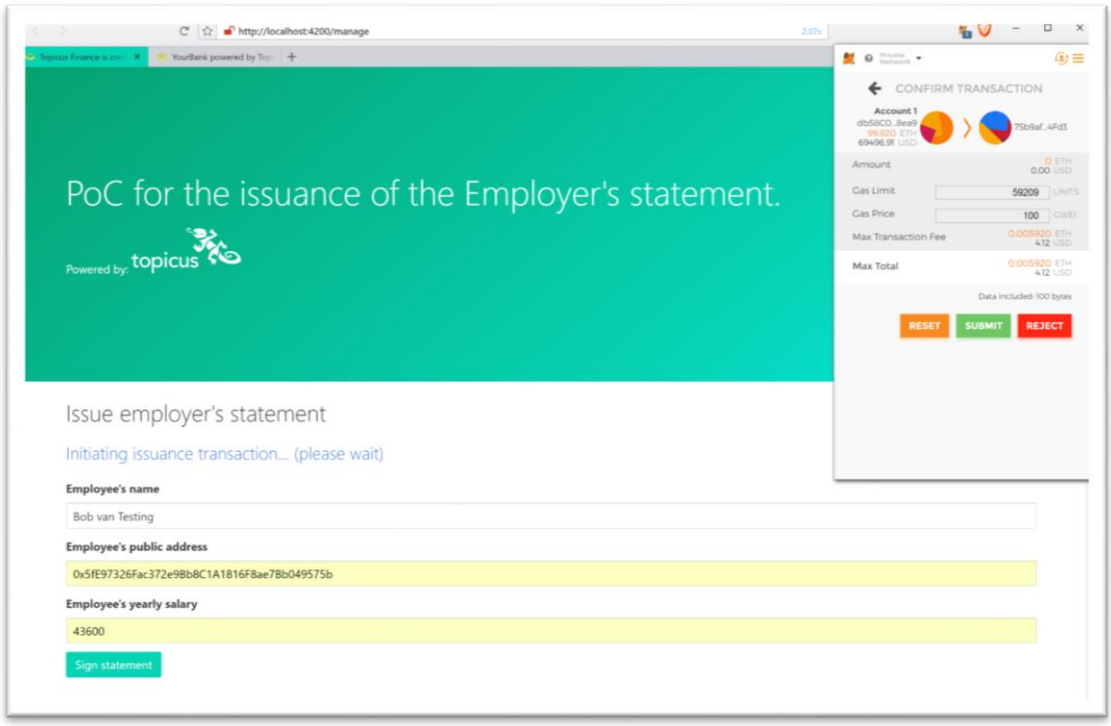


Figure 37: Initiating the blockchain transaction to issue the employer's statement

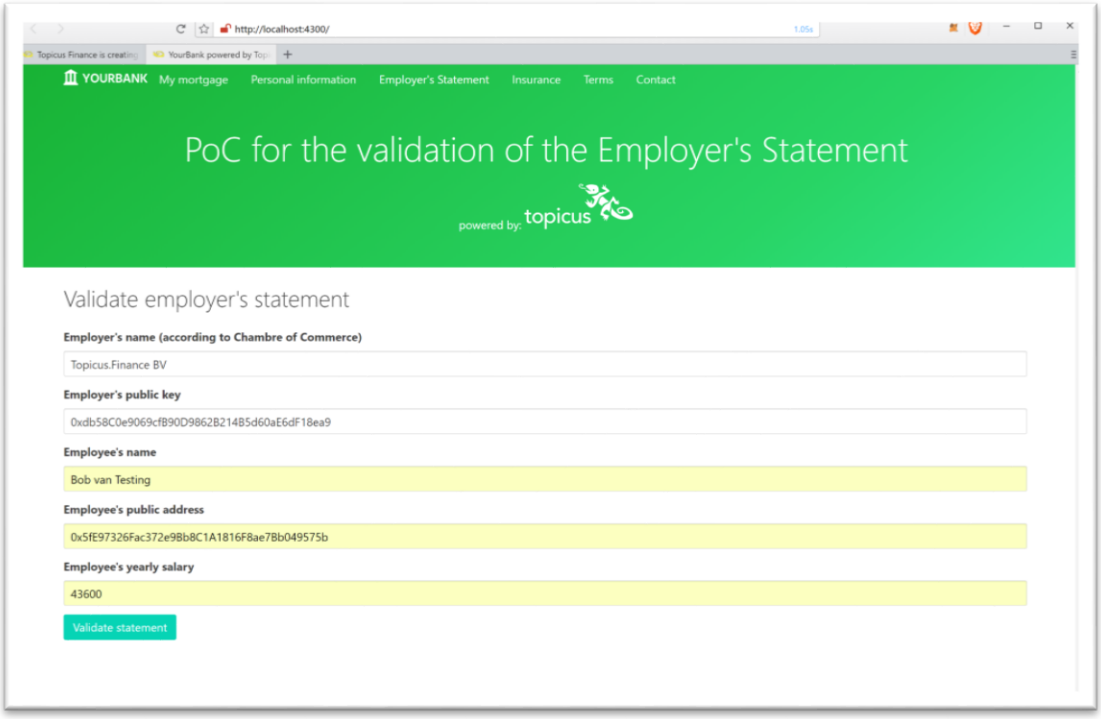


Figure 38: Validation page for the mortgage supplier's side.

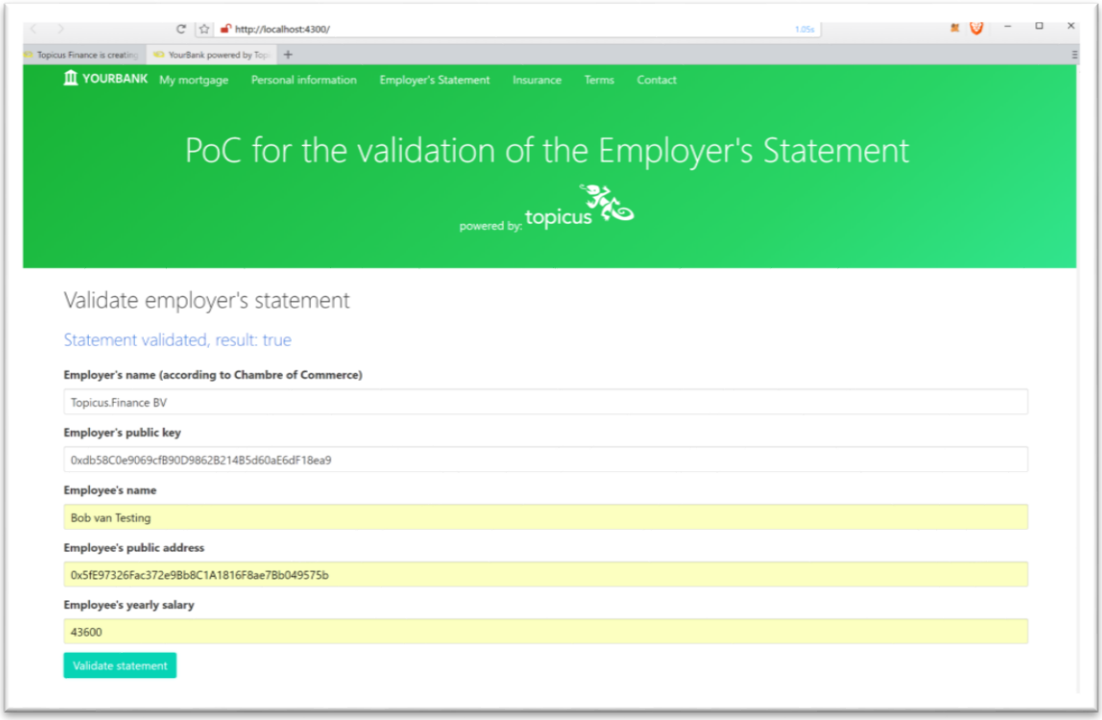


Figure 39: The correct information is supplied by the mortgage applicant, so the employer's statement validates

6.4.4 Validation

Although not the full design proposal was implemented in this proof of concept, it gave a good view of how the proposed design can be used in other contexts. The proposed design could fairly easily be fully implemented for this use case. However, the proof of concept helped uncover the following considerations for a full implementation:

- *Standardizing the data fields is an absolute necessity if the verification should be fully automated*. Custom fields need to be added to the Blockcerts in order to easily share verifiable salary information.
- *The identity of the employer needs a more reliable verification method*. The proposed design requires an issuer hosted website to host the public keys of the issuer. However, not even all employers have a company website. This could be solved by adding the public key field to the Chamber of Commerce hosted employer's registry.
- *Integrating the design with existing software applications is the key to adoption*. Even if the data fields are standardized, employers will only benefit from the system if the fields are automatically filled in from their own administration.

We are currently further investigating the product roadmap for the employer's statement application for the mortgage market.

6.5 Conclusion

According to Wieringa the validation of design artefacts is to be performed by answering the three questions as stated in 6.1 (Wieringa, 2014). We have explored the attacks and mitigations possibilities in the designed proposal, and reflected on the design with several experts. Furthermore, we discussed with blockchain experts the added value of our blockchain solution versus a solution using decentralized public keys. Lastly, we created a proof of concept that utilizes the design elements from our proposal to create a verifiable employer's statement for mortgage applications. Hereby we conclude that we have sufficiently validated the design proposal in multiple contexts.

7 Discussion and conclusion

The educational sector walks slow, and rightfully so. Especially when it comes down to digital improvements they tend to not be an early adopter. However, as we are getting bombarded nowadays with CV-liars, sifting fact from fiction is a time-consuming endeavour. Just recently, the CEO of the hundred year old company Samsonite resigned under false degree allegations (Zhong, 2018). To overcome this universal problem it is time for the higher educational industry to start experimenting with the creation of a universal decentralized digital degree. In this chapter we reflect on the research, discuss the generalizability of the proposed design, and go over a few limitations and future work.

7.1 Answer to the research questions and contributions

Our main research question for the research is:

How can we design a decentralised degree issue & verification method in order to combat fraud and increase efficiency in the job application process, for both the employee and employer?

The main contributions of this research are two-fold:

- Proposed a protocol by combining and applying existing standards to create a workable digital degree issuance and verification system.
- Proposed a method to ensure safe institutional issuance on a public blockchain with mitigations to private key compromises.

We concluded that the current institutional solutions do not suffice the user needs in this world-wide system of degree issuance and verification for employment. The verification methods are mostly manual, and non-standardized. Therefore it costs employers, and admission office employees tremendous amount of work to verify all the incoming job and study applications. With our design proposal, higher educational institutes benefit from the existing Ethereum infrastructure to host the degree proofs.

The method uses the existing Blockcerts open standard to create digital degrees, and improves it by relying on the smart contract for the revocation of the degree. We propose the usage of the ERC780 standard for the degree registry. The integration of these two standards is the core of the design proposal. An alumnus who has received their degree digitally according to our design, can selectively attest solely their degree title without off-chain Blockcert data requirements, or pick a subset of their Blockcerts to add to the attestation.

The benefits for using a public blockchain in this proposal are the high data availability, and the possibility of verification without the explicit cooperation of the original issuer. Even in the case an issuer goes out of business, the public blockchain continues to maintain an incentive to host your proof data. Although in this case a new trust anchor would need to be found for the identity verification of the issuer.

The added value of the smart contract technology is to contribute a sense of state to the data stored on the blockchain. We propose a relatively simple mechanism for the smart contract in which degrees can be revoked according to the ERC780 standard, but this otherwise called rich statefullness can be applied in much more intricate ways. With verifiable degrees on the public ledger, anyone could create a new smart contract that interacts with this attribute.

If our proposal would be implemented in the existing software it would be trivial to verify one's academic achievements. Therefore, employers and higher educational institutes can combat fraudulent applicants much easier. As the current job and study application processes are slowed by the manual verification of the credentials, we are convinced that the universal usage of this system will benefit the speed and trust in this workflow.

7.2 Limitations

Performing research implies setting restrictions in scope and ambition levels. This also applies for our research. Blockchain and smart contracting are a fast developing information technology, so it is important to understand the following limitations of this work.

- *Openness of certain data:* Our proposal uses a public blockchain. One of the consequences of this is that everyone will be able to see the amount of certificates registered by any entity. However, in the Netherlands this is already public information, so we do not regard this as a problem. With the usage of this system, the alumnus should only share their privately held Blockcerts with institutions they trust. The limiting consideration is: if the personal certificates of the alumnus are leaked, they can be eternally verified with the usage of the on-chain data.
- *Questionable eternal existence*: Although the current track-record for universal availability of the Bitcoin network for the last ten years, and for the public Ethereum network the last three, this does not necessarily mean this will continue forever. These networks have built in incentives to continue running, and have a huge security against networking attacks. Therefore we are convinced that the public networks have the highest probability of being around for a generation. If a consortium chain is set-up, there needs to be willingness by the stakeholders to maintain the blockchain for an indefinite amount of time, including governance over the chain. In that sense it is easier to use an existing public blockchain.
- *Adoption of the degree verification system by educational institutions:* Only if a critical mass is achieved, will the design proposal save time for the verifying parties, otherwise it is just an extra system that verifiers needs to get familiar to.
- *Continuity of the issuer institution*: The proposal still relies on the availability of the issuer for the identification of the keys. Over time, it may happen that this trust anchor can be moved to another place, and might even be included on the blockchain, but right now we propose the usage of existing websites as trust anchor.

7.3 Generalizability

A higher education degree is just one instance of a verifiable claim. Therefore, it is obvious that this research is generalizable to many more applications, as we have showcased through the application of the generic claim standard to two key societal practises: educational degrees and the employer's statement (6.4). There are many, many, more thinkable, verifiable claims use cases (Andrieu, Lee, & Otto, 2017). This proposed design is in essence fully generalizable in the sense that it does not lean much on the user needs specific to verifiable degrees. Although other use cases may find the meta-data with Blockcerts too little or too much, so it should be considered on a case by case basis what kind of meta-data standard is used.

We do strongly suggest to set up a meta-data standard for the verifiable data, as it can improve adoption tremendously as the system is then logically centralized, while being politically and physically decentralized. This enables anyone to create clients for the verification of the standardized documents.

7.3.1 Public blockchain vs other decentralized public key infrastructure solutions

By providing a design that makes degree verification possible on a public smart contract platform, we argue that the same would also be possible on a permissioned, or even private blockchain. As one can set the rules themselves in a permissioned, or private, system, the performance of the platform would only increase

A similar system could even be created without a smart contract platform entirely, by only using public key infrastructure, but then the solution would lose the rich statefulness characteristic. This complicates the revocation aspect and could also lead to unknowing misuse of compromised private keys.

7.4 Future work

7.4.1 Adoption and deployment

In this work we have presented a design proposal for the publication of digitally verifiable degrees. We have not completely implemented the proposal in a production ready environment, nor integrated it with any existing software at higher educational institutes or employers. Future research should focus on piloting the software with these parties, to be able to more accurately measure the efficiency benefit of a full roll-out of the system.

Furthermore, the research can focus on setting up an ecosystem with the intended users of the system. We suggest to collaborate with the active Blockcerts community, to find worldwide enthusiastic supporters.

7.4.2 Privacy by design

Currently, only a Merkle root of the certificates is put on the blockchain. But, it does show that the public key that you might use in other scenarios has received a degree from the higher educational institute that you went to. This is often public information, especially for universities, but in the traditional systems one has the "right to be forgotten". This is currently not possible in the proposed design. While the degree will not verify anymore if it is revoked by the user, the transaction that entered the degree in the registry is still part of the blockchain.

If a verifier leaks the Blockcerts you've handed over, then that personal information can be verifiably coupled to you, forever. In the case of a higher educational degree, I do not regard this as a huge risk.

It would be worth the effort to research the possibility to improve this proposed design into a privacy by design system. This might be achievable with advanced cryptography like ring signatures, or other zero-knowledge proof technology. However, we argue that the current relative simplicity of the proposed design will be beneficial in the adoption of the technology.

7.4.3 Educational Pathways

Educational pathways provide students with information on how degrees can be achieved through various routes of similar courses at multiple institutions. With the adoption of verifiable degrees, it would become easier for students to apply to other higher educational institutions as the application procedure can be simplified. Where it is too much hassle currently for a student to follow a single course at another university, the full deployment of this design proposal should reduce this hassle. Therefore students might follow courses at many more institutions than they currently do for a degree. Future research should concern how higher educational institutes can accommodate this even further to enable these educational pathways.

7.4.4 Reputation systems

The publication of the verifiable degree could be seen as the first element to a universal reputation system about you. The same design proposal can be used by issuers of other credentials to publish more verifiable records about someone. In a paper by Allan Third of the Open University UK he suggests that the current available technologies would already be sufficient to create the irrefutable history of you (Third & Domingue, 2017). Future research can be done into the positive applications of such decentralized systems.

8 Bibliography

ADP. (2009). *"Should employers consider outsourcing background checks?"* United States of America.

Allan, C. (2016). The Path to Self-Sovereign Identity. Retrieved May 14, 2018, from http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

Andrieu, J., Lee, S., & Otto, N. (2017). *Verifiable Claims Use Cases 1.0. W3c final Community report*. Retrieved from https://www.w3.org/2017/05/vc-use-cases/CGFR/2017-05-01/

Aragon. (2018). Aragon - Unstoppable Organizations. Retrieved May 14, 2018, from https://aragon.one/

BCDiploma. (2018). BCDiploma - Blockchain Certified Data - Certified Diploma on Ethereum. Retrieved May 14, 2018, from https://www.bcdiploma.com/index.html

Benjamin Boeser. (2017). Meet TrueRec by SAP: Trusted Digital Credentials Powered by Blockchain | SAP News Center. Retrieved from https://news.sap.com/meet-truerec-by-sap-trusted-digital-credentials-powered-by-blockchain/

Bitcoin.org. (2018). Unspent Transaction Output, UTXO - Bitcoin Glossary. Retrieved May 14, 2018, from https://bitcoin.org/en/glossary/unspent-transaction-output

Bitcoin Core. (2018). OP_RETURN - Bitcoin Wiki. Retrieved May 14, 2018, from https://en.bitcoin.it/wiki/OP_RETURN

Blockcerts. (2018a). Blockcerts : The Open Standard for Blockchain Credentialsredentials. Retrieved May 14, 2018, from https://www.blockcerts.org/

Blockcerts. (2018b). Blockcerts · GitHub. Retrieved May 14, 2018, from https://github.com/blockchain-certificates

Braendgaard, P., & Torstensson, J. (2018). ERC: Lightweight Identity · Issue #1056 · ethereum/EIPs · GitHub. Retrieved June 26, 2018, from https://github.com/ethereum/EIPs/issues/1056

Braz, C., Seffah, A., & Raihi, D. M. (2007). Designing a Trade-Off Between Usability and Security: A Metrics Based-Model. *Lecture Notes in Computer Science*, *4663*, 114–126. https://doi.org/10.1007/978-3-540-74800-7_9

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *Etherum*, (January), 1–36. https://doi.org/10.5663/aps.v1i1.10138

Buterin, V. (2015). Visions, Part 1: The Value of Blockchain Technology - Ethereum Blog. Retrieved May 14, 2018, from https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/

Buterin, V. (2017a). The Meaning of Decentralization. Retrieved May 14, 2018, from https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274

Buterin, V. (2017b). Vitalik Buterin on Twitter: "Turing completeness is and always was a red herring. What you *do* need for (practical) provably fair gambling is rich statefulness.… https://t.co/ItSuhg4eFh" Retrieved May 14, 2018, from https://twitter.com/vitalikbuterin/status/854271590804140033

Buterin, V. (2018). Vitalik Buterin speech in Deconomy 2018. Retrieved from https://www.youtube.com/watch?v=7WL9hr445uo

Buterin Vitalik. (2018). Sharding FAQ · ethereum/wiki Wiki · GitHub. Retrieved May 14, 2018, from https://github.com/ethereum/wiki/wiki/Sharding-FAQ

Calvanese, M. (2018). Flexible Upgradability for Smart Contracts. Retrieved May 14, 2018, from

https://medium.com/level-k/flexible-upgradability-for-smart-contracts-9778d80d1638

Cardano. (2018). Cardano - Home of the Ada cryptocurrency and technological platform. Retrieved May 14, 2018, from https://www.cardano.org/en/home/

Cook, M. (2016). *Personnel selection : adding value through people--a changing picture* (6th ed.). Retrieved from https://books.google.nl/books?hl=nl&lr=&id=g4sgCwAAQBAJ&oi=fnd&pg=PP7&ots=KtiHmz0hWd&sig=gYKITi_t3EcVmEeGMkI1yqiX4Xg#v=onepage&q&f=false

Coursera. (2017). Payments on Coursera – Coursera Help Center. Retrieved November 8, 2017, from https://learner.coursera.help/hc/en-us/articles/209818963-Payments-on-Coursera

Cuende, L. I. (2018). Luis Iván Cuende on Twitter: "What is most needed in the #Ethereum ecosystem?" Retrieved May 14, 2018, from https://twitter.com/licuende/status/948677941092904965

Delta, T. D. (2011, January 17). Geen cijferlijst in diplomaregister. *Delta*. Retrieved from https://www.delta.tudelft.nl/article/geen-cijferlijst-diplomaregister

Directive 2005/36/EC. (2005). Regulated professions by country, with competent authorities. Retrieved November 8, 2017, from http://ec.europa.eu/growth/tools-databases/regprof/index.cfm

Dogecoin. (2018). Dogecoin. Retrieved May 14, 2018, from http://dogecoin.com/

DUO. (2012). Alles over het diplomaregister - DUO. Retrieved November 8, 2017, from https://duo.nl/particulier/diplomas/over-mijn-diploma/diplomaregister.jsp

DUO. (2017a). *Aantal wo gediplomeerden - Open Onderwijsdata - DUO*. Retrieved from https://duo.nl/open_onderwijsdata/databestanden/ho/ingeschreven/wo-ingeschr/ingeschrevenen-wo5.jsp

DUO. (2017b). Diplomacheck doen - DUO Zakelijk. Retrieved November 8, 2017, from https://duo.nl/zakelijk/diploma/diplomacheck-doen.jsp

DUO. (2017c). Handleiding controle echtheidskenmerken. duo.nl.

Dykstra, R. H., Ash, J. S., Campbell, E., Sittig, D. F., Guappone, K., Carpenter, J., … McMullen, C. (2009). Persistent paper: the myth of “going paperless”. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, *2009*, 158–162. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2815440&tool=pmcentrez&rendertype=abstract

edX. (2017). Verified Certificate | edX. Retrieved November 8, 2017, from https://www.edx.org/verified-certificate

EEA. (2018). Enterprise Ethereum Alliance. Retrieved May 14, 2018, from https://entethalliance.org/

EOSio. (2018). eosio | Blockchain software architecture. Retrieved May 14, 2018, from https://eos.io/

Ethereum Name Service. (2018). Ethereum Name Service. Retrieved May 14, 2018, from https://ens.domains/

Etherisc. (2018). Etherisc - Decentralized Insurance. Retrieved May 14, 2018, from https://etherisc.com/#hero

EUR. (2016a). Examen register. Retrieved November 8, 2017, from https://www.eur.nl/essc/studentenadministratie/verificatie_diploma_en_andere_opleidingsgegevens/examen_register/

EUR. (2016b). Handmatige diploma verificatie. Retrieved November 8, 2017, from https://www.eur.nl/essc/studentenadministratie/verificatie_diploma_en_andere_opleidingsgegeve

ns/handmatige_diploma_verificatie/

Finney, H., & Nakamoto, S. (2009). Dai/Nakamoto emails. Retrieved May 14, 2018, from https://www.gwern.net/docs/bitcoin/2008-nakamoto

Flippening. (2018). Flippening Watch. Retrieved May 14, 2018, from http://www.flippening.watch/

FunFair. (2018). Game-changing blockchain casino technology - FunFair. Retrieved May 14, 2018, from https://funfair.io/

FutureLearn. (2017). Introducing Certificates of Achievement - FutureLearn. Retrieved November 8, 2017, from https://about.futurelearn.com/blog/introducing-certificates-of-achievement

Goligoski, E. (2012). Motivating the Learner : Mozilla ' s Open Badges Program. *Access to Knowledge*, *4*(1), 1–8.

Hevner, A. R. (2007). A Three Cycle View of Design Science Research A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems , 19*(192), 87–92. https://doi.org/http://aisel.aisnet.org/sjis/vol19/iss2/4

Hevner, A. R., March, S. T., Park, J., Ram, S., & Ram, S. (2004). Research Essay Design Science in Information. *MIS Quarterly*, *28*(1), 75–105.

Huberts, D. (2016). Update : Incoming student mobility in Dutch higher education 2016-17, 1–18.

IMS GLOBAL. (2017a). Enabling Better Digital Credentialing | IMS Global Learning Consortium. Retrieved November 8, 2017, from https://www.imsglobal.org/initiative/enabling-better-digital-credentialing

IMS GLOBAL. (2017b). Open Badges v2.0 specification. Retrieved November 9, 2017, from https://www.imsglobal.org/sites/default/files/Badges/OBv2p0/index.html#badge-verification

Interaction Associates and Human Capital Institute. (2013). Building Trust 2013, (June), 26.

Jagers, C. (2018). The Blockchain in Education `. Retrieved May 14, 2018, from https://medium.com/learning-machine-blog/the-blockchain-in-education-5a322fe9fe86

Joel Torstensson. (2017). ERC: Ethereum Claims Registry · Issue #780 · ethereum/EIPs · GitHub. Retrieved May 14, 2018, from https://github.com/ethereum/EIPs/issues/780

John Bear. (2012). *Degree Mills: The Billion-Dollar Industry That Has Sold Over a Million Fake Diplomas*. Retrieved from https://books.google.nl/books?id=DRgwAQAAQBAJ&printsec=frontcover&hl=nl#v=onepage&q&f=false

Kadenze. (2017). Pricing and Membership | Kadenze. Retrieved November 8, 2017, from https://www.kadenze.com/pricing_and_membership

Litecoin. (2018). Litecoin - Open source P2P digitale valuta. Retrieved May 14, 2018, from https://litecoin.org/nl/

Manuel Araoz. (2018). Proof of Existence. Retrieved May 14, 2018, from https://proofofexistence.com/

Melanie Swan. (2015). *Blockchain: Blueprint for a New Economy*. Retrieved from https://books.google.nl/books?hl=nl&lr=&id=RHJmBgAAQBAJ&oi=fnd&pg=PR3&ots=XQsGD1-Wk6&sig=8qSjJVaqkeoVNWUPKBXhTfeySVA#v=onepage&q&f=false

Merkle, R. C. (1988). A Digital Signature Based on a Conventional Encryption Function (pp. 369–378). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48184-2_32

Merola, R. (2016). *Combatting Student Visa Fraud: Top Destination Countries Employ Diverse*

Strategies. Retrieved from http://www.obhe.ac.uk/documents/view_details?id=976

Mike Cohn. (2008). User Story Template Advantages. Retrieved May 14, 2018, from https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template

MIT Media lab. (2016). What we learned from designing an academic certificates system on the blockchain. Retrieved May 14, 2018, from https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-blockchain-34ba5874f196

MyEtherWallet Knowledge Base. (2017). Checking or Replacing a TX After it's Been Sent · Transactions | MyEtherWallet Help & Support. Retrieved June 26, 2018, from https://kb.myetherwallet.com/transactions/check-status-of-ethereum-transaction.html

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, 9. https://doi.org/10.1007/s10838-008-9062-0

Namecoin. (2018). Namecoin. Retrieved May 14, 2018, from https://namecoin.org/

NEM. (2018). NEM – Distributed Ledger Technology (Blockchain). Retrieved May 14, 2018, from https://nem.io/

NEO. (2018). NEO Smart Economy. Retrieved May 14, 2018, from https://neo.org/

Olivier Tardieu. (2017). Werkgeversverklaring voor een hypotheek gaat vaak fout. Retrieved May 14, 2018, from https://www.peoplecompany.nl/werkgeversverklaring-voor-een-hypotheek-gaat-vaak-fout/

Patel, P. (2009). Experts Expect R??sum?? Fraud to Rise: In an economic downturn, the temptation to pad CVs is strong. *IEEE Spectrum*, *46*(6), 24. https://doi.org/10.1109/MSPEC.2009.4977601

Peercoin. (2018). Peercoin - Secure & Sustainable Cryptocoin. Retrieved May 14, 2018, from https://peercoin.net/

Pinxteren, G. van. (2004, January 22). Meer eisen voor Chinese student. *NRC*, p. 1. Retrieved from https://www.nrc.nl/nieuws/2004/01/22/meer-eisen-voor-chinese-student-7670701-a669211

Rowley, J. (2012, July 18). Degree fraud: detection as deterrent. *The Guardian*. Retrieved from https://www.theguardian.com/higher-education-network/blog/2012/jul/18/degree-fraud-hedd-checking-service

Rozanski, N., & Woods, E. (2012). *Software systems architecture : working with stakeholders using viewpoints and perspectives*. Addison-Wesley. Retrieved from https://books.google.nl/books?hl=nl&lr=&id=ka4QO9kXQFUC&oi=fnd&pg=PR7&dq=rozanski+woods&ots=ytOnX_VIWa&sig=4pjCBO7CIFWrSWKNlH-1mbvNCns#v=onepage&q=rozanski woods&f=false

Rumsey, M., Walker, C., & Harris, J. (2013). *Personnel Selection and Classification* (2nd ed.). Psychology press. Retrieved from https://books.google.nl/books?hl=en&lr=&id=9cVbus0Rv0cC&oi=fnd&pg=PP1&dq=Rumsey,+M.+G.,+Walker,+C.+B.,+%26+Harris,+J.+H.+(Eds.).+(2013).+Personnel+selection+and+classification.+Psychology+Press.&ots=3_leiFRLsS&sig=Im9OdtrFsvWnV8UngztfehhIK5E#v=onepage&q&f

Saxion. (2017). Diplomaverificatie - Saxion. Retrieved November 8, 2017, from https://www.saxion.nl/peno/samenwerken/diplomaverificatie

SkillChain. (2018). Skillchain - Your Certified Skills on Blockchain. Retrieved May 14, 2018, from https://www.skillchain.io/

Third, A., & Domingue, J. (2017). The irrefutable history of you: Distributed ledgers and semantics

for ubiquitous personal ratings. *CEUR Workshop Proceedings*, *1939*.

Thrun, S. (2014). Phasing out certificates of free courseware completion | Udacity. Retrieved November 8, 2017, from https://blog.udacity.com/2014/04/phasing-out-certificates-of-free16.html

TRON. (2018). TRON. Retrieved May 14, 2018, from https://tron.network/enindex.html

UNIC. (2017). Self-Verifiable Certificates on the Bitcoin Blockchain – UNIC Blockchain Initiative. Retrieved May 14, 2018, from https://digitalcurrency.unic.ac.cy/free-introductory-mooc/self-verifiable-certificates-on-the-bitcoin-blockchain/

Universiteit Leiden. (2017). Diplomaverificatie - Universiteit Leiden. Retrieved November 8, 2017, from https://www.universiteitleiden.nl/alumni/diploma-cijferlijst/diplomaverificatie

Universiteit Utrecht. (2017). Verklaringen - Studenten | Universiteit Utrecht. Retrieved November 8, 2017, from https://students.uu.nl/praktische-zaken/afstuderen/verklaringen

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Vogelsteller, F. (2017a). ERC: Claim Holder · Issue #735 · ethereum/EIPs · GitHub. Retrieved June 26, 2018, from https://github.com/ethereum/EIPs/issues/735

Vogelsteller, F. (2017b). ERC: Identity · Issue #725 · ethereum/EIPs · GitHub. Retrieved June 26, 2018, from https://github.com/ethereum/EIPs/issues/725

Warren, W., & Bandeali, A. (2017). 0x : An open protocol for decentralized exchange on the Ethereum blockchain, 1–16.

Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. *Springer Berlin Heidelberg*. https://doi.org/10.1145/1810295.1810446

Zhong, R. (2018). Samsonite C.E.O. Resigns After Accusation of Résumé Fraud - The New York Times. Retrieved June 26, 2018, from https://www.nytimes.com/2018/06/01/business/samsonite-chief-resign.html

9 Appendix

Appendix A: Example Blockcert


```
{"@context": [
    "https://w3id.org/openbadges/v2",
    "https://w3id.org/blockcerts/v2"
  ],
  "type": "Assertion",
  "id": "urn:uuid:fba193-2918-423f-d1c9-af7251dd731c",
  "badge": {
    "id": "urn:uuid:82a4c9f2-3588-457b-80ea-da6951b8fc",
    "type": "BadgeClass",
    "name": "Example Msc. Degree",
    "image": "...",
    "description": "This is an example blockchain certificate according to the Blockcerts
standard",
    "criteria": {
      "narrative": "To achieve this, you at least need to understand the inner workings of the
Blockcerts architecture."
    },
    "issuer": {
      "id": "https://www.utwente.nl/degree-issuance.json",
      "type": "Profile",
      "name": "University of Twente",
      "url": "https://www.utwente.nl",
      "email": "degrees@utwente.nl",
      "revocationList": "https://www.utwente.nl/degree-revocations.json",
      "image": "data:image/png;..."
    }
  },
  "recipient": {
    "hashed": false,
    "identity": "f.l.brinkkemper@student.utwente.nl",
    "type": "email"
  },
  "recipientProfile": {
    "type": [
      "RecipientProfile",
      "Extension"
    ],
    "publicKey": "ecdsa-koblitz-pubkey: 0xdeaDDeADDEaDdeaDdEAddEADDEAdDeadDEADDEaD ",
    "name": "Frank Brinkkemper"
  },
  "issuedOn": "2018-04-21T11:23:44.111425+00:00",
  "verification": {
    "publicKey": "ecdsa-koblitz-pubkey:0xfC522b943D068116074c1C36839515Fac0aa224E",
    "type": [
      "MerkleProofVerification2017",
      "Extension"
    ]
  },
  "signature": {
    "type": [
      "MerkleProof2017",
      "Extension"
    ],
    "targetHash": "4c15a6a12637ecaf680519a17ade09f5b424a32fa4b7b569e9ea48eb0e6f5ad0",
    "merkleRoot": "0b970f6de35344f029b45bb1a7b1fb73cccd16177b4c037acbc2541c7fc27078",
    "anchors": [
      {
        "sourceId": "b84e987267cfa4ffa7a532d75b7a5bdb3d5244b753e6a2ed49ad3848be1d82f8",
        "type": "ETH_mainnet"
      }
    ],
    "proof": [
      {
        "right": "74e220fe74111de9e2a357e4143407d1107123f2948eecd1fc6b946fbfd7e3e3"
      }
    ]
  }
}
```

Appendix B: Interview with Nick Johnson

Frank Brinkkemper (FB): I would ask if you could introduce yourself in a few sentences including your expertise with identity related use cases on a blockchain.

Nick Johnson (NJ): sure I am Nick Johnson I am a core developer for the Ethereum foundation. I work on the Go-Ethereum team, and primarily I am the lead developer and originator of the ENS, the Ethereum Name Service. Which is a distributed naming system for the Ethereum blockchain and for wider applications. It aims to solve the problem of allocating human usable identifiers to resources, and as such has uses in the identity field as well as general naming.

FB: For ENS, do you think it has uses besides having human readable names for public keys?

NJ: you can use it to assign human readable name to a variety of resources. For instance we see uses in DNS records, but more importantly it can be used to add arbitrary meta-data to both names and addresses. So you can associate a cryptographic claim or certificate to a name or address.

FB: Alright, and as the creator of the ENS what do you think is holding back traditional institutions wanting to use ENS?

NJ: Probably unfamiliarity is a significant factor, it works in tandem with the adoption of blockchain and Ethereum in particular. In ENS our main focus is first to get adoption for Ethereum based uses. Later we can get adoption for other things like DNS. So mostly thus far we are making sure it's getting used in Ethereum based projects, before its really used by external ones.

--- Frank shows presentation (slides see other appendix) ---

<< shows the general method>>

FB: Do you have any comments on the general method thus far?

NJ: How do you do the underlying verification? 'Cause in this demo you need the location for the keys. How does Topicus in this case know that these are the keys from a recognized organisation?

FB: That is still needed outside of the blockchain. The key location, in this case the information on utwente.nl/keys has to be trusted to believe that this is an actual degree from this university.

NJ: Right. Why not just rely on the sender of the request, the issuer in this case? And then you can look up claims by that issuer. You don't need the URI embedded in the blockchain, when it can be claimed on the issuer webpage.

FB: Good comment, I haven't fully thought through if that would indeed be possible. I will redo my argumentation there.

NJ: One of the open questions with ERC780 is what format the "keys" [of the open key, value pair] should take. One of the suggestions is that there should be space for standardized keys. So, for example that all the degrees are issued with a key "degree-claim", or something. The other issue with adding the URI to the blockchain is that it can never be changed. Regarding the value: how long is the merkle root of the Blockcerts?

FB: It should fit in a bytes32, that is the requirement. I currently have it together with the abbreviation of the degree title in the "value" parameter. This is not fixed in stone yet.

NJ: The merkle root should at least be 16 bytes long to be certain that it cannot be brute forced. Therefore the abbreviation of the degree title could better be put as "key" value. Or otherwise maybe two corresponding claims of which one references the degree title, and the other is done for the merkle root.

FB: Or perhaps a DID that points to a degree title, indeed something that I need to rethink.

One of the problems with the current explained method is that whenever the private key is compromised, whether knowingly or unknowingly, that could lead to degrees being issued by attackers. That is of course unwanted.

NJ: Right.

FB: So, I have also thought about a safe degree issuing application for the institutions.

<<FB explains the corresponding slides from appendix X>>

NJ: This model assumes that the attacker can't compromise the management keys, because they are stored offline somewhere I assume?

FB: That's correct. These are rarely used, only when necessary. Some actions require multiple management keys. For example setting new issuing keys requires multiple management keys. But not revoking an issuing key, which requires only 1 management key. This indeed assumes that at least 2 of the management keys, are kept uncompromised in the case of 3 total management keys.

NJ: Indeed, and as these management keys are not used in the day to day operation, they are a lot easier to keep uncompromised than the issuing keys. One suggestion is that it might be easy to make a distinction between the security models as a user sees it, and how you maintain security on a practical basis. From the outside users' point of view: the external facing website is the trust-root. So whether or not you have this mitigation strategy for mitigating key compromises, the most important thing for a user is whether you keep in control of the external facing website. So even if you didn't have this mechanism. If the attacker has got your keys, you could also issue new claims to everyone, and say on the external facing website: "this is our new address".

Internally, you can still have this mechanism, which can make sure that the attacks can be mitigated with less overhead and hassle than that would be the case if all the claims need to be re-issued. This is more a mitigation strategy, than an integral part of the security model, as far as an external user is concerned.

FB: Agreed.

NJ: That would make sense to call out, so it is understood that the users don't need to know about this mechanism. If the website hosts the correct keys, then it is authoritative.

FB: Would you say that this is sufficient for most institutions to mitigate these kind of key compromises by miners?

NJ: Yes, I think the mechanism outlined there should be pretty effective, because the management keys are kept offline. The question I would remain with, as a external user: "if the external facing website is the trust-root, why not store the merkle root there of all the degrees issued?"

FB: Good question. Right now I believe that, that would certainly be a possibility. Hopefully in the future the external facing website is not even needed. The system can rely on systems that put identity on the blockchain. So, for example ERC725 or ENS.

NJ: The other argument I'd make is that if you'd have to contact the website to verify your degree, then if the website is offline or uncooperative, there is no way to verify the merkle root is correct. Whereas on the blockchain you have universal availability.

FB: Agreed. Some final questions. What would you say is the added value of blockchain for this use case?

NJ: So, I mean, you can proof things with crypto, for example private keys, and merkle roots. What you can't do is revocation. And you can't prevent the equivalent here of double spending, where the issuer would sign one degree, and then later sign a different degree, and thereby altering your degree. And also that you can provide data availability here. Once you have provided the alumnus with the location in the blockchain and the merkle tree of documents, then they can always prove it independently without the cooperation of the university. Also when it goes out of business, or stuff like that. So I would say that the advantages here are non-revocation and the availability issue + the fact that it can be verified independently without the cooperation of the issuer.

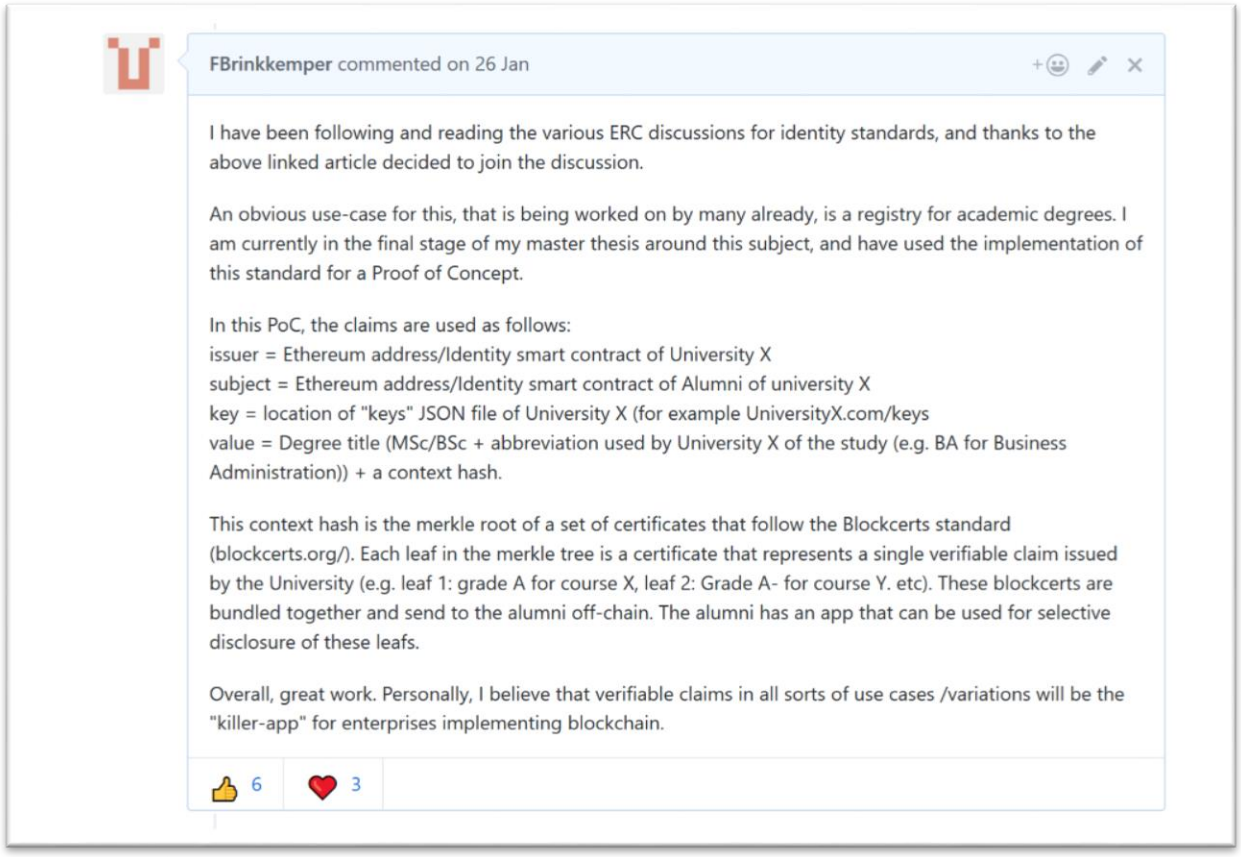
FB: Alright. Finally: what would you say is the added value of smart contracts here?

NJ: It allows you to build more flexible ways of doing issuance and revocation. So regular public key cryptography, combined with merkle proofs, would not allow you to do such things as the failover

mechanism against the attack. and they would make it more difficult to make nuanced permission and update structures.

FB: Great. Thanks a lot for your time, and answers!

Appendix C: Github comment on ERC780.



FBrinkkemper commented on 26 Jan

I have been following and reading the various ERC discussions for identity standards, and thanks to the above linked article decided to join the discussion.

An obvious use-case for this, that is being worked on by many already, is a registry for academic degrees. I am currently in the final stage of my master thesis around this subject, and have used the implementation of this standard for a Proof of Concept.

In this PoC, the claims are used as follows:
issuer = Ethereum address/Identity smart contract of University X
subject = Ethereum address/Identity smart contract of Alumni of university X
key = location of "keys" JSON file of University X (for example UniversityX.com/keys
value = Degree title (MSc/BSc + abbreviation used by University X of the study (e.g. BA for Business Administration)) + a context hash.

This context hash is the merkle root of a set of certificates that follow the Blockcerts standard (blockcerts.org/). Each leaf in the merkle tree is a certificate that represents a single verifiable claim issued by the University (e.g. leaf 1: grade A for course X, leaf 2: Grade A- for course Y. etc). These blockcerts are bundled together and send to the alumni off-chain. The alumni has an app that can be used for selective disclosure of these leafs.

Overall, great work. Personally, I believe that verifiable claims in all sorts of use cases /variations will be the "killer-app" for enterprises implementing blockchain.

6 3

https://github.com/ethereum/EIPs/issues/780#issuecomment-360825613