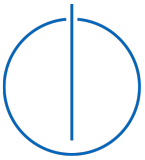
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Analysis of the State of the Art and the Practice of Digital Credentialing

Dominik Gerbershagen



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Analysis of the State of the Art and the Practice of Digital Credentialing

Analyse des aktuellen Stands und der praktischen Umsetzung digitaler Zertifizierungen

Author:	Dominik Gerbershagen
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Ulrich Gallersdörfer, M.Sc.
Submission Date:	March 15th, 2020

I confirm that this master’s thesis in information systems is my own work and I have documented all sources and material used.

Munich, March 15th, 2020 Dominik Gerbershagen

Acknowledgments

With this master's thesis, my studies at the Technical University of Munich come to an end. Therefore, I want to first and foremost thank my family for supporting me throughout this period. I could not have made it without you. Equally, I want to thank Julia who supported and motivated me to strive for excellence and keeping the spirits high.
Furthermore, I would like to extend my gratitude to the Chair of Software Engineering for Business Information Systems, namely Prof. Dr. Florian Matthes and Ulrich Gallersdörfer for accepting my thesis and letting me explore the domain of digital credentialing. Especially Ulrich was a great team player during the course of this thesis. I have not experienced an advisory that provided this level of feedback and inspiration I have received from Ulrich.
Lastly, a big thank you to Prof. Dr. Thomas Wüstrich, Mario Weiherer, Lars-Tristan Nißsen, Johannes Hollinka, Kevin Bischof and Patrick Gerbershagen for proof-reading the thesis. Although it is written on my own, the hints and advice you gave me improved the quality significantly.

Abstract

Within the educational sector, credentials are the most sensitive and valuable documents that one can obtain. Receiving a credential is a milestone for each person and can be of significant importance for his or her career. Contrary to current digitization efforts, credentials are still issued in printed formats. To support the paradigm of lifelong learning and equally enable people to share and store credentials safely, these credentials have to be digitized. Yet, digitization is prone to errors and counterfeit. Therefore, the domain of digital credentialing has emerged.

With the rise of the blockchain-based technology, tamper-proof and immutable storage systems have emerged. What first seemed to be a solution for currency systems parallel to the governmental ones has more and more developed into an architecture for protecting and verifying data. Thus, new business models have been created around the concept of immutability, transparency and distribution. One of these concepts is digital credentialing. This thesis investigates the current state of the art and practice in terms of research and standardization for this domain.

Prior to the investigation of the state of the art and practice, an overview of current blockchain systems and identification methods is provided. Both subjects are relevant for current digital credentialing systems. Blockchain-based systems feature verification and immutability mechanisms by design and is a widely adopted technology in the investigated sample. Identification methods are relevant to create a relationship between virtual entities and analogue counterparts.

Afterwards, specifications such as IMS OpenBadges and the Verifiable Credentials one by the World Wide Web Consortium are investigated. Based on a technical and functional analysis, a framework is established. This framework serves as a foundation to investigate the current state of the practice. Eighteen practitioners and ten research projects are investigated by applying the framework to them. The gathered data is then aggregated and analyzed to demonstrate similarities and differences within and across the peer-groups. Conclusively, the data is evaluated to form an impression of the current state of the practice.

Kurzfassung

Abschlusszeugnisse bilden im Bildungsbereich die sensibelsten und wertvollsten Dokumente, die eine Person erlangen kann. Der Erhalt eines solchen Dokuments stellt einen Meilenstein für jeden Teilnehmer dar und kann von bedeutender Wichtigkeit für den weiteren Verlauf der Karriere sein. Im Gegensatz zum aktuellen Trend der Digitalisierung, jedoch, werden Abschlusszeugnisse immer noch in Papierform ausgestellt. Um das Paradigma des lebenslangen Lernens zu unterstützen und gleichzeitig ein sicheres Speichern und Teilen der Zertifikate zu ermöglichen, müssen diese digitalisiert werden. Doch eine einfache Digitalisierung alleine ist anfällig für Fehler und Fälschung. Aus diesem Grund hat sich der Bereich der digitalen Zertifizierung entwickelt.
Mit dem Aufkommen von Blockhain-basierter Technologie haben sich fälschungssichere und unveränderbare Speichersysteme entwickelt. Was anfangs nach einer Lösung für parallele Währungssysteme aussah, hat sich über die vergangenen Jahre mehr und mehr zu einer Architektur für das Verifizieren und Schützen von Daten entwickelt. Eines dieser sich daraus neu entwickelten Geschäftsmodelle ist die digitale Zertifizierung. Diese Thesis befasst sich mit dem aktuellen Stand und der praktischen Umsetzung dieser Thematik.
Vor der Untersuchung wird jedoch noch ein Überblick über Blockhain-basierte Systeme und verschiedene Identifizierungsmethoden gegeben. Beide Themen sind relevant für aktuelle digitale Zertifizierungssysteme. Die Blockchain-Architektur beinhaltet baulich bedingt schon die Möglichkeit des Verifizierens und der Unveränderbarkeit, weswegen sie von vielen untersuchten Marktteilnehmern und Forschungsprojekten verwendet wird. Die Identifizierungsmethoden sind notwendig, um einen Bezug zwischen virtuellen Subjekten und ihren analogen Pendants herzustellen.
Nachfolgend werden Spezifikationen wie der IMS OpenBadges Standard und der Verifiable Credentials Entwuf vom World Wide Web Consortium untersucht. Basierend auf einer technischen und funktionalen Analyse wird ein Rahmenwerk entwickelt, welches als Grundlage für die Untersuchung der aktuellen Umsetzung von digitaler Zertifizierung dient. Achtzehn Marktteilnehmer und zehn Forschungsprojekte werden untersucht und das Rahmenwerk jeweils darauf angewandt. Die gesammelten Daten werden im Anschluss aggregiert und analysiert mit dem Ziel, Gemeinsamkeiten und Unterschiede sowohl in der jeweiligen Referenzgruppe, als auch gruppenübergreifend aufzuzeigen. Abschließend werden die Daten ausgewertet, um einen Überblick über den aktuellen Stand der Umsetzung zu gewinnen.

Contents

Acknowledgments **iii**

Abstract **iv**

Kurzfassung **v**

1 Introduction **1**
1.1 Problem Statement . . . 1
1.2 Research Questions . . . 2
1.3 Research Approach . . . 3
1.4 Outline . . . 4

2 Fundamentals **5**
2.1 Definition of Digital Identity . . . 5
2.2 Definition of Credential . . . 6
2.3 Differentiating Between Macro-Credentials and Micro-Credentials . . . 6
2.4 Introduction to Blockchain-Based Systems . . . 7
2.4.1 Overview of the Bitcoin Network . . . 8
2.4.2 Overview of the Ethereum Network . . . 9

3 Related Work **11**
3.1 Related Work on Macro-Credentials . . . 11
3.2 Related Work on Micro-Credentials . . . 14

4 Analysis of the State of the Art **16**
4.1 Stakeholders in the Domain of Digital Credentialing . . . 16
4.2 Requirements and Characteristics of a Digital Credentialing System . . . 17
4.3 Actions and Processes in a Digital Credentialing System . . . 21
4.3.1 Issue . . . 21
4.3.2 Store . . . 21
4.3.3 Refresh . . . 21
4.3.4 Revoke . . . 22
4.3.5 Receive . . . 22
4.3.6 Assemble . . . 22
4.3.7 Interact . . . 23
4.3.8 Verify . . . 23
4.3.9 Surroundings . . . 24
4.4 Creating a Framework for System Comparison . . . 24

5 Technical Examination of Digital Credentialing Specifications and Identification Methods **28**
5.1 Investigating the W3C Verifiable Credentials Data Model 28
5.2 Analysis of the OpenCerts Specification 36
5.3 Examining the IMS Mozilla OpenBadges Specification 40
5.4 Establishing Identification Between Virtual and Analogue Entities 46
5.4.1 Introduction to eIDAS 46
5.4.2 Introduction to Decentralized Identifiers 49
5.4.3 Creating a Link Between DID and eIDAS 51

6 Analysis of the State of the practice **52**
6.1 Applying the Framework to Practitioners 52
6.1.1 Accredible 52
6.1.2 APPII 55
6.1.3 BCDiploma 57
6.1.4 BlockCo 59
6.1.5 BlockCerts 61
6.1.6 Blockeducate 63
6.1.7 CHESICC 65
6.1.8 Credly 67
6.1.9 CVTrust 69
6.1.10 Edgecoin 71
6.1.11 Gradbase 73
6.1.12 Keeex 75
6.1.13 Parchment 77
6.1.14 SAP TrueRec 79
6.1.15 Sony Global Education 81
6.1.16 Sproof 83
6.1.17 Stampery 86
6.1.18 Vottun 88
6.2 Applying the Framework to Research Projects 90
6.2.1 Blockchain and Smart Contracts for Digital Certificate 90
6.2.2 Blockchain Education Platform 92
6.2.3 Blockchain-Based Education Records 94
6.2.4 Blockchain-Based Educational Record Repository 97
6.2.5 Blueprint for Learning Trace Repositories 99
6.2.6 Certificate Verifying Support System 102
6.2.7 CredenceLedger 104
6.2.8 Distributed Credit Transfer 106
6.2.9 Educational Certificate Blockchain 108
6.2.10 QualiChain 110
6.3 Evaluation of the State of the Practice 113
6.3.1 Practitioners 113

6.3.2 Research Projects 116

7 Conclusion and Future Work **119**

List of Figures **122**

List of Tables **123**

Acronyms **125**

Bibliography **128**

1 Introduction

Successful graduation form an educational program comes with the issuance of a credential. With the rise of smaller educational programs originating from the professional education sector, the credential types have changed as well. Apart from traditional ones such as the diploma or a master's degree, *badges* have been entered the credentialing market. Different to a diploma, a badge can be credited upon successful completion of a much smaller program, for instance participating in a Massive Open Online Course (MOOC). Lately, these courses have gained popularity. As a result, large companies such as IBM, Oracle, Microsoft and more have created their own educational platforms to further refine their staff's skill set. This is a trend, not only employees can benefit from. Universities, rooted in the traditional issuance of *macro-credentials*, have developed online courses as well. Apart from their regular schedule for students, smaller courses deal with certain aspects of a study program that is credentialed with badges instead of diplomas.
Although the educational sector is developing new strategies to credit students, the way they are issued, handled and stored has not changed yet. The educational still sector lacks a standardization and compliant systems that support the principle of *lifelong learning* [1]. However, with current technological trends, various solutions are setup to tackle this problem.
The most prominent trend regarding storage and verification is the blockchain technology. Beginning with Bitcoin that has been used as a completely digital currency, subsequent systems have changed the focus from currency to distributed machinery. Especially Ethereum offers the possibility to create a distributed network of components that cooperate as a an application. Based on this evolution, the domain of digital credentialing is evolving as well. The first player on the market has been IMS OpenBadges with a specification for digital badges. Using traditional storage systems such as central databases or servers, the implementation of the OpenBadges specification has rather been separated from blockchain trends. However, this is currently changing.
This thesis investigates the current trends of digital credentialing. Based on standardization efforts, a Framework is conducted to compare current market participants and research projects within the domain of digital credentialing.

1.1 Problem Statement

Credentials are highly sensitive documents containing information that is utmost relevant for the earner's career. Therefore, it has to be handled deliberately. Circumventing the document is an implicit model of trust. An employer, for instance, who receives the

document trusts that it has been issued by the institution that is mentioned in the document. Furthermore, the employer has to trust the institution that the document contains the correct content. A printed document does not feature verification process. The only way for an employer, or anyone else, to verify the data is to contact the institution and ask them if the document is valid. This becomes even more important when the document is scanned, because scanned documents can be manipulated easily. As a reference, the German Federal Criminal Police Office lists 76,176 cases of certificate fraud for 2018 [2]. The number shows that manipulating credentials occurs, but it does not show how many cases have not been discovered.
One problem with digital credentialing is that scanning and digitization alone is not enough to prove the authenticity of a document. The document has to be enhanced with mechanisms providing a proof for third-parties that the it has not been manipulated. Secondly, the authenticity of the issuer has to be proven as well. An authentic document does not only contain valid claims about the subject, it also comes from a valid source. Thirdly, the whole education sector is diversified. Although there is standardization in place such as the Bachelor's / Master's program, the diploma ultimately does not follow a unified schema every institution is using. This becomes even more scattered when taking professional education provided by companies instead of universities into account.
For the above mentioned problems, research and marketeers have already developed systems, specifications and projects. Yet, the overall maturity of them is still in its early stages.

1.2 Research Questions

From the problem statement above arise the following research questions (RQ) that are answered throughout the course of this thesis:

RQ1: How is the current state in terms of standardization for digital credentialing?
The educational sector is comprised of a broad variety of stakeholders. Among others, companies, learners, institutions and more are participating in this domain. To harmonize the way these stakeholders communicate, operate and educate, standardization has always been necessary. The same holds for translating traditional credentialing into the digital realm. With the urge for digitization in this sector, new standards have already been specified or are on the verge of publication. With this research question, specifications dealing with digital credentialing are investigated.

RQ2: What requirements and processes have to be in place to create a digital credentialing system?
With the findings of RQ1, a first outlook of a digital credentialing system can be provided. Since the goal of standardization is to create a common basement for creation, handling and communication of data, several requirements and processes have to be in place. Theoretically, each system can have a different implementation based on the same

requirements. To compare these, a framework is generated based on current specifications.

RQ3: Which companies and research projects are already participating in the market for digital credentialing?
With a framework in place, the market and research area of digital credentialing is traversed to investigate the current state of the practice. The goal of this research question is to find peers and apply the framework to the systems they offer. As a result, data is generated that can analyzed for similarities and differences. Furthermore, it shows current challenges and the degree of standardization within the state of practice.

RQ4: What are the differences and similarities of these companies and projects?
Lastly, the generated data from RQ3 is analyzed and an evaluation is provided. An overview of the state of practice is given that shows how companies are tackling the concept of lifelong learning. Equally, this research questions deals with how companies or research projects resemble and where they differ from each other.

1.3 Research Approach

Since this thesis provides an overview both of the current research and the current market situation in terms of digital credentialing, a dual approach has been taken.
First, a literature review is conducted that begins with a list of relevant work and research that encompasses the domain of digital credentialing. From there on, a backward propagation has been conducted for the original sources. Furthermore, a list of keywords has been aggregated that was used for literature search as well.
Within the literature, several specifications have been mentioned that are explained afterwards. Namely the IMS OpenBadges and the World Wide Web Consortium Verifiable Credentials specifications are examined in detail. Derived from these, requirements and processes are assembled and incorporated into the framework. Additionally, it contains two sections about system design and business relevant aspects. Research regarding these is inspired by the *Grounded Theory Method* [3]. Originating from social studies, the concept of iteratively generating and analyzing empirical data to get and derive theories from that has been useful for the mentioned categories. While doing research on current marketeers and research projects, these categories evolved from the data gathered.
The selection of companies working in the domain of digital credentialing was mostly based on the two factors:

1. Is the company dealing with macro-credentials?
2. Is there public information available about the company?

After creating a list of companies and research projects, the framework has been applied to create a data set for each one. In the same chapter, an evaluation aggregates these data sets into one and lists commonalities as well as differences for each domain.

1.4 Outline

Chapter 2 aims to create a common understanding of core technology and terminology for reading this thesis. In this chapter, credentials, digital credentials, identity and blockchain systems are defined.
In the next chapter, related work is presented that also deals with digital credentialing. However, these texts have either a different approach or a different scope.
Chapter 4.3 provides an overview of the state of the art and assembles a framework based on that. With this framework, companies and research projects are investigated and compared in Chapter 6. Preceding the application of the framework, Chapter 5 provides a technical examination of the state of the art. Additionally, it demonstrates how digital credentialing can be connected with two major identification methods. Eventually, the thesis ends with the conclusion and a section about future work in Chapter 7.

2 Fundamentals

Throughout the following chapters, various terms and technologies will be explained and put in relationship with each other. Some of them, especially *identity* and *credential*, can be very similar. To understand the differences and similarities between the terms and technologies, this chapter provides background information and serves as a foundation for later references.

2.1 Definition of Digital Identity

The domain of identity and its translation into the virtual realm itself are both broad and complex topics. Therefore, only a brief overview is provided to differentiate between digital identity and digital credentialing.
When taking a look at digital identity and the various existing definitions, it becomes clear that *identity* itself can only be seen as a part of a larger context. The minimum context to make identity work is the triad of *identity*, *attributes* and *entities* [4]. An attribute is the "fundamental element" [4] to describe a person, an institution, a car or anything else that has to be described. Attributes can be either temporary or persistent [5] and are usually in the form of <key, value>. A set of attributes forms an identity which can be used to *identify* a certain entity uniquely. To achieve a correct identification, either a unique attribute or a unique set of attributes has to be assigned to an entity. For instance, a human being cannot be uniquely identified by using the eye color attribute. A car with a serial number, however, can be uniquely identified since the attribute is sufficiently unique.
Entities can be described as the "(...) overall profile of a person or an organization" [4]. Consequently, digital identity is always comprised of these three elements: attributes, identities and entities.
Another triad comes into play when an identity is used to access a service or data. As L. Jean Camp describes in his *Digital Identity* article, "Identification requires authentication of identity; access to data must often be preceded by authentication" [5]. Therefore, the second triad according to Camp is the following:

- *Identification*: "Identification is the association of a personal identifier with an individual presenting attributes, e.g., 'You are John Doe.'"
- *Authentication*: "Authentication is proof of an attribute."
- *Authorization*: "Authorization is a decision to allow a particular action based on an identifier or attribute."

These six terms are all incorporated in the most common form of identification elements we currently have: the passport or personal identification card. This paper-bound document features the first triad entirely as it describes an entity with attributes, therefore forming an identity. The entity is in this case a human being. Furthermore, this passport can then be used for identification, authentication and authorization. It is the ultimate tool in the domain of traditional identity. But how is this translated into the digital realm?
In the European Union (EU), the translation of the second triad (identification, authentication and authorization) is regulated by the EU Regulation No. 910/2014 dealing with "electronic identification and trust services for electronic transactions in the internal market" [6]. In this regulation, the European Parliament sets the requirements for identification, authorization and authentication in the digital realm. In short, this regulation is called *eIDAS* for *Electronic Identification, Authentication and trust Services (eIDAS)*. The exact mechanism behind eIDAS is explained in Chapter 5.4.1.

2.2 Definition of Credential

A credential, according to the online version of Merriam-Webster, is "something that gives a title to credit or confidence" [7]. Following this definition, a credential can be tied to many use cases such as a driver's license indicating a person is permitted to drive a car. Or a passport that shows a person's citizenship. From a technical point of view, Herzberg and Mass define a credential as "(...) an assertion by an *issuer* of some *attributes* of the *subject* of the credential" [8]. This definition already shows how similar the two terms identity and credential are. Both rely on attributes that are assigned to either an entity or a subject, although the output is different for identity and credential. A credential relies on an identity, whereas an identity does not rely on a credential.
Furthermore, credentials are not only tied to positive aspects. A conviction report can equally be seen as a credential as e.g. a coding skill [8]. Consequently, the domain of digital credentialing is not bound to education, but has the potential to also impact the public sector and governmental agencies.
Throughout the past years, a trend has emerged to differentiate between two sorts of credentials in the educational sector: macro- and micro-credentials. As explained in the following section, both types have a unique scope and case of application.

2.3 Differentiating Between Macro-Credentials and Micro-Credentials

Traditionally, a credential in the educational sector is bound to degrees. A degree is usually achieved "(...) over extended periods on successful completion of a course (or program) (...)" [9]. Furthermore it is comprised of several marks and grades that are earned throughout this extended period of time. All this forms a macro-credential which is crediting a certain amount of successfully completed educational programs. Within the European

Union, a credential is always issued with a *transcript of records* that shows mandatory and voluntary courses [10]. Both issue credits that have to be accumulated in order to successfully earn a degree and ultimately the academic title the student aims to achieve.
For decades, the general understanding was that a credential is necessary to enter a career in certain jobs. This perception is changing. The consulting firm Ernst & Young followed a motion "to scrap a requirement of at least a 2:1 from its graduate application process" [11]. Instead only taking degrees into account, the company aims at more holistic assessments to check their applicants' abilities. It shows that the traditional perception of degrees is changing and moving towards other forms of qualifications. One of these forms are micro-credentials.
With the rise of online courses and platforms such as Coursera or Pluralsight, learning possibilities have diversified. Instead of participating in a lengthy program for e.g. computer science, learners can enroll in a Massive Open Online Course (MOOC) that teach interactively parts of the computer science study program. Each of these programs grant the learner a micro-credential in the form of a PDF, an image or a so called *Digital Badge*. Digital Badges are the common format for micro-credentialing. Originating from military and scout badges, the digital badge is "(...) a clickable graphic that contains an online record of 1) an achievement, 2) the work required for the achievement, 3) evidence of such work, and 4) information about the organization, individual, or entity that issued the badge" [12]. With a digital badge, a learner can demonstrate achievements that are tailored to specific interests or needs and show third-parties that certain requirements for e.g. a job is met. Contrary to the current state of macro-credentials, digital badges can be verified through platforms such as the Mozilla OpenBadges Infrastructure (OBI).
Although the trend of micro-credentialing has emerged and learners benefit from the flexibility of MOOCs, Lemoine and Richardson state that "non-traditional credentials have not carried the same trust in all areas of the world (...)" [12]. Macro-credentials such as diplomas are still perceived as the go-to solution for entering the job-market. This undermines the necessity of standardizing and digitizing macro-credentialing the same way micro-credentialing has been [9].

2.4 Introduction to Blockchain-Based Systems

Throughout the course of this thesis, the term *blockchain* will appear on several occasions. To create a common understanding about blockchain technology, what the current major systems are and how they separate from each other, this section describes both the Bitcoin and Ethereum network.
Blockchain systems operate differently from centralized servers with common client-server architectures. In these scenarios, a server hosts the data and application logic where numerous clients can connect to each with their own session. The result is one single trusted entity accountable for both data and procedures that are offered. Contrary to a centralized architecture, the blockchain is fully distributed among all participating peers (also referred to as nodes). Each (full) peer holds all the information of the state of the

blockchain and therefore serves as a replica of the entire system making it very reliable. The architecture giving the blockchain its name is similar to a linked list. Whenever new information is about to enter the network, a block with a header and a trunk with data in it is formed. The information in both parts varies from system to system, as explained below. Once a block is formed, it gets chosen by miners and validated by the network. The block is then mined and appended to the longest chain. Appending a block is achieved through hashing the whole block and adding this hash into the next block. As a result, a list is created that always points to the previous block in the chain. One of the effects this architecture provides is the effort it takes to forge a random block inside the chain: Each appended block only contains valid transactions inside the blockchain [13]. Therefore, when an already appended block has to be altered, each subsequent block has to be mined again to form the longest chain. Only then an attacker has successfully rewritten a block and formed a new valid chain.
Contrary to a central authority which is trusted, the blockchain concept is operating trust-less [14]. Trust-less means, that there is no single point that guarantees a user can trust a third-party. However, trust has to be established. Without it, any output is worthless since a user cannot expect it to be valid. Therefore, trust is achieved through various cryptographic mechanisms such as the consensus algorithm, a search puzzle and public-key infrastructure. The following subsections describe the most common blockchain-based systems and their implementation of the above mentioned mechanisms.

2.4.1 Overview of the Bitcoin Network

Chronologically, the Bitcoin system was the first on the market and serves as a digital currency. Bitcoin also is the largest one of the so called cryptocurrencies. On the day of writing this subsection, the trading volume consists of 19.6 Billion Euro [15]. Compared to the number two in the market, Ethereum (ETH), the Bitcoin (BTC) trading volume is more than twice as large [16].
As with every blockchain-based system, Bitcoin uses the core techniques of hashing and chaining a block so that immutability is achieved. Therefore, a timestamp server assures that an order of blocks is created. The server takes a block with information that is already hashed, adds the current time to it as well as pointer to the previously appended block. This creates the chain of blocks.
Inside these blocks, information about transactions, the hash of the previous block and a *nonce* (number only used once) are stored. To prevent attackers from forging and appending invalid information, the proof-of-work mechanism is implemented in the Bitcoin system. Each block that is supposed to be appended to the chain has to be mined by participating nodes, the so called miners. Each miner tries to solve an equation by finding the correct amount of zero bits in the nonce [14]. Once a miner solves the puzzle, the solution is broadcast throughout the network and participating nodes verify all included transactions, meaning none is already spent. Afterwards, the block is appended to the chain and the miner gets a reward for her work, in this case a coin. The protocol targets an interval time of around ten minutes with one megabyte block-size per block

[17]. In comparison to the below explained Ethereum network with an block-interval of approximately ten to twenty seconds, the Bitcoin blockchain operates significantly slower. With this mechanism, the integrity of the blockchain is asserted. Since nodes follow the longest chain, altering a node in the chain is highly expensive. The altered node and all subsequent ones would have to be mined and appended to another chain so that a new longest chain is created [14].
Contrary to traditional banking systems that are based on accounts, the Bitcoin system operates on a transaction-based ledger. Each transaction contains an input and output field that is signed with the owner's and receiver's private key. The amount of unspent coins is stored in a list of unspent transaction outputs instead of a bank account. Each participant is required to use a software called *wallet* that enables her to store and issue transactions [18]. Therefore, the participant creates a public-private key pair that is used to sign transactions. Each issued transaction is signed with a private key which can be validated with the corresponding public key by third party. Once a transaction is issued, miners include it in the next block and perform the above mentioned proof-of-work algorithm.

2.4.2 Overview of the Ethereum Network

While Bitcoin focuses on providing a transaction-based centralized application for cryptocurrency, the Ethereum system is set up to create an environment for applications on the blockchain. A main distinction that makes this possible is the usage of an account-based instead of a transaction-based ledger [19]. An Ethereum account consists of a nonce, its ether balance, "(...) the account's contract code (...) and its storage which is empty by default [19]. These accounts are either owned by a contract code or a private key. Similar to the Bitcoin wallet, an account owned by a private key is bound to an entity such as a person. The contract code account allows to create accounts that operate automatically: The so-called *smart contracts*. A main design principle in Ethereum is called *universality*, meaning that everyone who has the capacity and intention to create an application is able to do so on the Ethereum network [19]. Therefore, Ethereum features its own programming language called *Solidity* and the Ethereum Virtual Machine. Both technologies support the development of third-party Decentralized Applicatons (DApps). Based on smart contracts, DApps serve as containers for business logic. An account can call these smart contracts to get certain values or products in return. However, smart contracts have to be designed compliant to two underlying principles: First of all, each function has to be deterministic so that the blockchain network is able to verify the function (this excludes functions based on random numbers). Secondly, a smart contract cannot access any service outside the blockchain (e.g. a HTTP Service). Again, this is due to forcing deterministic behavior inside the Ethereum network. For intercommunication between the Ethereum network and different ones such as the internet, trusted third-parties are required serving as an entry point to the network. In the Ethereum terminology, these access providers are called *Oracles* [20]. With this mechanism, the deterministic constraint is met and the blockchain's integrity assured.
To avoid flooding the network and simultaneously reward participating nodes delivering

computational power to the network, Ethereum uses a fee for each computational step called *Gas*. With every transaction or message issued, the field `STARTGAS` represents "(...) the maximum number of computational steps the transaction execution is allowed to take (...)" and `GASPRICE` the amount the sender is willing to pay per computational step [19].

3 Related Work

This thesis serves as an overview of both the state of the practice and the art of digital credentialing. Both approaches have already been discussed in related work, which is presented here. Simultaneously, the following paragraphs separate this thesis from related work.

3.1 Related Work on Macro-Credentials

Alammary, Alhazmi, Almasri, and Gillani used an holistic approach to create an overview of the domain's current state of the research. Based on a multiple-step-methodology, the authors gathered, reviewed and classified papers published in the scope of their three research questions. The leading question here is "What applications have been developed with blockchain technology for educational purposes?" [21]. Starting with more than 2000 articles, the researchers applied their methodology and extracted 47 relevant articles for a full-text reading review. The resulting data shows that the domain of the blockchain-based applications for the educational sector is gaining momentum since the year 2016 and shows a significant increase in publications throughout the years 2017 to 2018. Additionally, the most mentioned term is *certificate management* (43%). Certificate management represents a broader scope of "(...) issuing, storing, and sharing students' academic certificate" [21].
The following two categories are "competencies and learning outcomes management" and "evaluating students' professional ability" [21]. The former category (29% of the articles) explicitly discusses blockchain-based solutions for the educational sector. This indicates that the evolution of blockchain towards "blockchain 3.0" [22] is on its way and the educational sector appears to be a promising one for the first application.
The third category deals with the issues of how an employer can distinguish between students that have good grades and students that are really matching the profile. Current academic transcripts the TUM issues only contain a headline and the grade of attended courses. Employers are not able to see the actual contents the student has learned in this lecture, neither does this indicate if a student is matching a job profile. The research in this area focuses on how enterprises can evaluate the achievements the student has made throughout her career [21].
Jirgensons and Kapenieks start their research from a different point. They investigate the potential application of blockchain technology to the education sector. Similar to Alammary, Alhazmi, Almasri, and Gillani, their approach is to get an overview of the current status of tool implementation. However, the focus of their paper is more on the future of the

educational sector. Remarking that "(...) in our information age where there appears a continuous stream of innovations every day (...)", it seems to be counter-intuitive that the lifelong learning is still bound to printed certificates [1]. Indeed, this is the starting point for various research papers and one of the major reasons for an evolution in the domain of certificates.

Focusing on the blockchain technology, the authors mention major challenges that have yet to be overcome. The first is scalability: Due to the technological nature of the blockchain, each node has to be mined to be appended to the chain. This mining process uses a high amount of energy and time, increasing simultaneously with the increase of puzzle-solving difficulty. Currently, the Bitcoin network alone is assumed to have a lower-bound energy consumption of 2.55 Gigawatts (GW) [23]. For a better understanding: To produce this amount of energy, 7,96 million photo-voltaic panels are necessary according to the United States Department of Energy [24]. Consequently, scaling a blockchain system means scaling energy consumption as well.

The second major challenge is privacy. Since the information in a blockchain network is transparent and accessing the network is unrestricted, each person with e.g. the hash of a another person can see his or her certificates.

The last challenge is storage. Although information is stored decentralized, storage capabilities have to be created in order to append blocks to the chain. Contrary to e.g. cloud computing data bases, the blockchain systems are not administered by a single company in a data center, but nodes can access and store the chain flexibly on their hard drives. With an increasing network size, the size of storage has to grow as well.

Concluding their research, Jirgensons and Kapenieks mention that there are currently two credible players on the market of educational blockchain: The MIT Media Lab in collaboration with Learning Machine and their Blockcerts technology based on the Bitcoin system. And the United Kingdom's (UK) Open University (OU) smart contracts based on the Ethereum Blockchain for storing micro-credentials. Both parties will be examined in Chapter 6.

Contrary to the aforementioned paper, G. Chen, B. Xu, Lu, and N.-S. Chen focus on the advantages a blockchain system bears. The authors mention four features that are standing out compared to traditional systems: "Decentralization, traceability, immutability, and currency properties" [25]. From these features, the main advantages of a blockchain system can be derived: Security, trust, efficiency and reliability [25]. At a first glance, the advantages remarkably resemble the challenges mentioned before by Jirgensons and Kapenieks. Due to the current state of the blockchain systems, mostly referred to Blockchain 2.0 by e.g. G. Chen, B. Xu, Lu, and N.-S. Chen and Alammary, Alhazmi, Almasri, and Gillani, these advantages yet have challenges that need to be overcome.

The four mentioned advantages base on a common foundation: decentralization. Consisting of many participating nodes, the blockchain network operates so that trust is not solely established by one single entity, but through and underlying mechanism such as Proof-of-work [26]. The same holds for reliability: If a connected node fails, the whole data set is still distributed on all remaining nodes, making the network more robust than a centralized

database. The chain is systematically backed-up on all connected peers [25].
Security and efficiency, however, are more related to the process of validation and issuing certificates to the system. A rigid procedure such as uploading the certificate to the blockchain, which then has to be mined and announced to the connected peers serves as an efficient automation that saves costs and labor [25]. Additionally, the announced certificate information is hashed. The hash value is unique for this document and once anchored in the blockchain, it can be distinguished between false copies and the original document.
From a conceptual point, the authors propose that smart contracts as implemented by the Ethereum blockchain can improve fairness and executive power [25]. To achieve this, teachers and students could submit their work to a public blockchain which tracks their academic contribution throughout the school year and provides automatic evaluation. This can then be rewarded by a custom currency based on the principle of "learning is earning" by Sharples and Domingue.They push the idea of an academic-used blockchain further to the extent of uploading the evaluation intellectual property to a public blockchain. Apart from validating certificates, the authors propose to use it as a "proof of intellectual work" [27]. A user could upload all sorts of intellectual property such as patents, poems, or an artwork which could be tested for authenticity and immediately claimed for that person. This way, a user would not need to issue a patent itself, for instance, since the patenting process could be achieved through smart contracts. Additionally, the authors propose an intellectual currency system based on the property's usage. Similar to H-Index for academic publications, the blockchain would store credits for the use in the author's wallet, thus granting more reputation [27].

Chakroun and Keevy form an overall perspective of the digital credentialing domain. From the very beginning, the authors state the urgency of a standardization and digitization to "(...) reach a common approach where all aspects of a person's learning are electronically documented, authenticated and can be accessed at any time and anywhere, shared and amended by the owner or by an authorized party" [28]. Not only is this relevant for degrees, diplomas and related certificates, summarized as *macro-credentials*, but also for minor achievements, summarized as *micro-credentials*, which can be obtained from single modules in online learning courses. The principle of life-long learning is here mentioned as well, which requires in infrastructure that serves as a life-long passport for educational achievements.
Furthermore, the authors describe the underlying digitization technologies. Firstly, current document types are mentioned, specifically Extensible Markup Language (XML), JavaScript Object Notation (JSON) and Portable Document Format (PDF). Secondly, database structures and repositories are explained of which some have been existent over a longer period of time. Yet, they are impacted by the current wave of digitization [28]. Two major areas are blockchain and Artificial Intelligence (AI). Both have high potential of impacting the way credentialing might work in the future, where the blockchain serves as the underlying technology for storage, verification and management of certificates and artificial intelligence as a way to gain more insights in the domain of credentialing. Specifically, artificial

intelligence could be implemented to grade students for their assessments [28]. However, AI is not assumed to make teachers redundant. Rather, the authors state that "(...) We need to think more carefully about what AI does well and what humans do well" [28]. This way, artificial intelligence can be used to automate repetitive tasks and letting teachers focus on the ones a human can do better.
In the following chapters, the authors describe previously mentioned concepts of a *digital credentials economy*. The very first blockchain system, Bitcoin, was initially designed as a cryptocurrency. The most recent blockchain systems such as Ethereum, Ripple and others also embody this trait, however offering a larger variety of services. In the educational sector, cryptocurrency can be used as a motivational system to gain credits whenever an educational milestone has been achieved. Investigating this concept is out of scope for this thesis but could be interesting as a part of future work.
In conclusion, the authors provide seven recommendations of which the first states "Ubiquity and interoperability should be based on agreed standards" [28]. This underlines the importance of a common standard for issuing and managing digital credentials.

3.2 Related Work on Micro-Credentials

The term *micro-credentials* forms the counter part to the aforementioned macro-credentials. Following the paradigm of life-long learning, a learner will eventually achieve multiple certificates that are not directly connected to a university degree. Mostly, these so called badges are earned through MOOCs. During these trainings, a learner can practically and theoretically gain knowledge about various topics such as programming languages, physics, social studies and many more. The difference to a university degree is the time it takes and the scope of such an online course. However, universities such as the Hasso-Plattner-Institut in Potsdam, Germany, offer these courses. This means, a university, that is a traditional macro-credentialing institute, uses micro-credentialing courses [29] as an additional way of education. This trend has been emerging since 2010 according to Gibson, Ostashewski, Flintoff, et al. One year later, in 2011, the Mozilla Open Badges initiative launched, representing a major approach to standardize the field of micro-credentialing [30]. But even before the uprise of educational badges, different platforms and communities used this technique to increase engagement and show achievements. Gibson, Ostashewski, Flintoff, et al. mention Foursquare and Microsoft's Xbox game service, but also companies such as Google use badges for their Google Maps service to promote user interaction with locations [31].

Muilenburg and Berge provide an overview of the micro-credentialing approach both in the educational and technical environment. The authors cluster various projects in standardizing and proprietary ones. Basically, the only standardizing technology is the Mozilla Open Badges Initiative which had been newly released as of the time of publishing the paper. Exemplary proprietary ones are the WordPress Simple Badges plugin allowing WordPress users to integrate online badges in their own websites, as well as Accredible, a

now hybrid solution for micro- and macro-credentialing. However, the paper was published and 2013 meaning that some of the mentioned solutions have either changed or been discarded. Nevertheless, the paper shows that even in 2013 the Mozilla Open Badge initiative had been seen as the major force to standardize the micro-credentialing domain.

4 Analysis of the State of the Art

Before investigating current approaches for digital credentialing systems, this chapter shows what they are comprised of. Starting with entities participating in these networks and which objects are relevant, requirements are stated that should be fulfilled in a digital credentialing system. Secondly, actions are investigated that are conducted in such a system. Use cases are established that are applied to the various systems in order to check them for differences and similarities. Thirdly, a framework that contains features and attributes to lay out a basement for comparing the systems from a property-perspective is established. With this set of tools a comparison can be conducted in the following chapters to investigate the current state of the practice in digital credentialing.

4.1 Stakeholders in the Domain of Digital Credentialing

When we investigate digital credentialing from an educational point, it becomes clear that there have to be three entities involved in the system:

1. The learner who is achieving something that can be credited,
2. The institution which is issuing the credential in line with the learner's performance,
3. And a relying party that receives the credential and depends on its correctness.

Besides the educational sector, digital credentialing can occur in different industries as well. A supply chain company could use digital credentials to assert the correctness of documents throughout its chain. Human resource departments could use credentials to store the career path of an employee. Consequently, the roles have to be suited not only to the educational sector as we intuitively stated above. A definition applicable to several industries and stakeholders has to be used.
The World Wide Web Consortium (W3C) has formed a specification for *Verifiable Credentials* [1] that defines these roles without binding them to the educational sector [33]. Instead of using the term *stakeholders*, the W3C employs the term *roles*. This allows to separate physical entities such as stakeholders by their purpose within the system. Furthermore, each entity can have one or more roles. The definition grants more flexibility and disconnects a digital credentialing system from the boundaries of the educational sector. The roles the W3C has created are the following:

[1] In the following chapters also referenced as the W3C VC draft.

- *Holder*: The holder is an entity that owns several "(...) *verifiable credentials* and generates *verifiable presentations* from them." A verifiable presentation is "(...) a tamper-evident presentation encoded in such a way that authorship of the data can be trusted after a process of cryptographic verification."
- *Issuer*: An entity that creates a verifiable credential out of claims that can be assigned to a subject. A claim serves as an assertion such as "I have completed the JavaScript introduction course" and can be verified through a verification process. Additionally, the Issuer sends the verifiable credential to the holder.
- *Subject*: Instead of tying the credential only to human beings such as learners, the W3C ties credentials to subjects. A subject can have any form of entity that has to be credentialed. Therefore, the credentialing system compliant to the W3C VC draft supports generalized usage beyond the educational sector. Furthermore, the standard itself states that "(...) in many cases the holder of a verifiable credential is the subject (...)."
- *Verifier*: What is defined as relying party above is named verifier in the data model: Someone who receives the credential and processes it. This can be an employer or an institution such as a university.
- *Verifiable Data Registry*: In the W3C specification, the system performs its own role. It can create and verify "(...) identifier, keys and other relevant data, such as verifiable credential schemas, revocation registries, Issuer public keys, and so on (...)". Furthermore, the document provides examples such as trusted databases or distributed ledgers.

Especially the roles *Issuer*, *holder* and *verifier* are relevant throughout the course of this thesis. As we will see in the subsequent sections, the *Verifiable Credentials* draft not only serves as a foundation for roles, but also for actions and requirements. Therefore, relevant parts will be extracted for creating a framework after a prior analysis of the draft.

4.2 Requirements and Characteristics of a Digital Credentialing System

Now that it has become clear who is participating in a digital credentialing system, it is relevant to see which requirements each role has. The World Wide Web Consortium has recently published a document created by a working group that summarizes this task [34]. The group conducted research on users' needs in several industries such as health care, retail, education and finance. These needs were then aggregated into larger meta-tasks that each domain has to implement without focusing on their special needs. Again, we adopt these requirements because it fits the role model defined above and allows us to use them not only for the educational sector, but also for other domains. The following quotes are cited from [34]:

- Requirement 1: Issue claim. "It MUST be possible for any entity to issue a verifiable credential."
- Requirement 2: Assert claim. "It MUST be possible for the holder of a verifiable credential to restrict the amount of information exposed in a credential they choose to share. It also MUST be possible for the holder to limit the duration for which that information is shared."
- Requirement 3: Verify claim. "It MUST be possible for a verifier to verify that the credential is an authentic statement of an Issuer's claims about the subject. The verifying entity must have the capability to connect the Issuer's identity to its credential identifier and the subject's identity to their identifier as indicated in the credential. The Issuer's verification information, such as its public key, must be discoverable from the credential record and verifiably linked to the Issuer. It MUST be possible to do this in an automated fashion."
- Requirement 4: Store claim. "It MUST be possible for the holder of a claim to store that claim in one or more credential repositories."
- Requirement 5: "It MUST also be possible for the holder to move a claim among credential repositories, and to do so without requesting a new claim from the claim Issuer."
- Requirement 6: Retrieve claim. "It MUST be possible for a holder to select if and which appropriate credential should be sent to a verifier."
- Requirement 7: Revoke claim. "It MUST be possible for the Issuer of a claim to revoke it, after which it will no longer satisfy verification procedures."

Using these requirements we can create a first outlook on how a digital credentialing system operates. Since *Requirement 1* states that verifiable credentials can be issued by "any entity" [34], an entity can be simultaneously a holder and an Issuer or vice versa. Especially the first example demonstrates the necessity of two components: A strong verification mechanism that uncovers that a holder has issued certificates himself. Secondly, an identification mechanism for at least the Issuer so that credibility is ensured. An issuing institution that self-issues credentials is in theory not a problem. Only if holders claim to be a credible institution and self-issue credentials, the system is undermined.
Furthermore, the requirements state that a system has to be able to retrieve data from different repositories (*Requirement 4*). The idea of using several credential repositories fits *Requirement 3* in [35]. The author describes that there has to be data portability to an extent that allows to move credentials from one system to another. Since the market of digital credentialing systems is emerging, the competition might consolidate into several larger players eliminating smaller ones. To save from getting lost in the consolidation process, moving it to another one has to be implemented. Furthermore, this requirement allows to store data in save environments an institution could provide or even storing the

data locally on a small machine connected to the network.
Another challenge that is incorporated in the requirements is privacy. D. Tapscott and A. Tapscott describes that several universities such as Ohio State or the University of Wisconsin-Milwaukee have been successfully attacked and credentialing data had leaked. It undermines that credentials contain highly sensitive and valuable information about their holders. Consequently, *Requirement 5* and *Requirement 2* restrict the amount of information that is shared with relying parties. A system that implements these requirements has to be able to let the holder decide for each claim if it may be shared with different users and select a certain time-limit until the sharing option is valid. Furthermore, it implies that information stored in repositories and processed in this network has to be disclosed and secured so that data cannot be accessed by someone who hasn't been granted access.
Revocation is described in *Requirement 6* that allows Issuers to invalidate credentials. The verification mechanism recognizes this revocation and marks the credential as invalid or unverifiable. Since the credential is not deleted, a history of formerly valid credentials can be created, indicating that a person e.g. held several credentials such as Microsoft Expert certifications that are only valid for a defined amount of time.
Lastly, identification has to be implemented so that each holder can be uniquely identified with the information contained in the credential the entity is holding (*Requirement 3*). Without the correct association between certificate and holder, the document becomes worthless. It also has to be done in an automated manner so that identification can be done by systems. The requirement doesn't state that an entity has to be identified. Only the connection between the holding entity and the credential has to be authentic. An open question arises at this point: How is the connection between an entity's ID in the system and the actual ID of in the real world (e.g. a passport) established. Assuming the verifier sees a credential with only the entity's ID in the system, it cannot correlate if the two entities are authentically linked.

Deriving from the above mentioned requirements, the authors of the W3C draft have proposed several "desirable characteristics" [33] a system should implement. With an assigned role and ID, these characteristics can be found in Table 4.1. Different to requirements, characteristics are not mandatory for implementation according to the consortium. They rather serve as a guideline how a credentialing system could operate. Deriving from this guideline, the next section creates a set of actions that a system could feature. These actions will allow us to compare existing digital credentialing systems based on functionalities.
Concluding this section we have defined requirements that have to be fulfilled and a set of characteristics that we adopt for clustering them into actions. Both items are based on the W3C VC draft since the scope of this document suits the research goals of this thesis.

ID	Role	Description
CHol1		"Holders assemble collections of verifiable credentials from different issuers into a single artifact, a verifiable presentation."
CHol2		"Holders can receive verifiable credentials from anyone."
CHol3	Holder	"Holders can interact with any issuer and any verifier through any user agent."
CHol4		"Holders can share verifiable presentations, which can then be verified without revealing the identity of the verifier to the issuer."
CHol5		"Holders can store verifiable credentials in any location, without affecting their verifiability and without the issuer knowing anything about where they are stored or when they are accessed."
CHol6		"Holders can present verifiable presentations to any verifier without affecting authenticity of the claims and without revealing that action to the issuer."
CHol7		"If a single verifiable credential supports selective disclosure, then holders can present proofs of claims without revealing the entire verifiable credential."
CIss1		"Issuers can issue verifiable credentials about any subject."
CIss2		"The specification must provide a means for issuers to issue verifiable credentials that support selective disclosure, without requiring all conformant software to support that
	Issuer	feature."
CIss3		"Issuers can issue verifiable credentials that support selective disclosure."
CIss4		"Issuers can issue revocable verifiable credentials."
CIss5		"Issuers can provide a service for refreshing a verifiable credential."
CIss6		"Issuers revoking verifiable credentials should distinguish between revocation for cryptographic integrity (for example, the signing key is compromised) versus revocation for a status change (for example, the driver's license is suspended)."
CIss7		"Issuers can provide a service for refreshing a verifiable credential."
CVer1	Verifier	"A a key has been compromised verify verifiable presentations from any holder, containing proofs of claims from any issuer."
CSys1		"Acting as issuer, holder, or verifier requires neither registration nor approval by any authority, as the trust involved is bilateral between parties."
CSys2		"Verifiable presentations allow any verifier to verify the authenticity of verifiable credentials from any issuer."
CSys3	System	"Verification should not depend on direct interactions between issuers and verifiers."
CSys4		"Verification should not reveal the identity of the verifier to any issuer."
CSys5		"The data model and serialization must be extendable with minimal coordination."
CSys6		"Verifiable credentials represent statements made by an issuer in a tamper-evident and privacy-respecting manner."
CSys7		"Verifiable presentations can either disclose the attributes of a verifiable credential, or satisfy derived predicates requested by the verifier. Derived predicates are Boolean conditions, such as greater than, less than, equal to, is in set, and so on."
CSys8		"Verifiable credentials and verifiable presentations have to be serializable in one or more machine-readable data formats. The process of serialization and/or de-serialization has to be deterministic, bi-directional, and lossless. Any serialization of a verifiable credential or verifiable presentation needs to be transformable to the generic data model defined in this document in a deterministic process such that the resulting verifiable credential can be processed in an interoperable fashion. The serialized form also needs to be able to be generated from the data model without loss of data or content."
CSys9		"Revocation by the issuer should not reveal any identifying information about the subject, the holder, the specific verifiable credential, or the verifier."

Table 4.1: Desirable characteristics a credentialing system should feature. Adopted and cited from [33]. The items are assigned to categories and numbered by the author of the thesis.

4.3 Actions and Processes in a Digital Credentialing System

The goal of this section is to cluster the characteristics defined by the W3C VC draft and shown in Table 4.1. From there on, we can create a set of actions based on clustering these characteristics. As already described in the previous sections, these characteristics are not mandatory for implementation. Consequently, neither are the actions and use cases that result in clustering the characteristics. The purpose of clustering and describing the actions is to better understand how a system could operate. When applying the framework, the goal is not to see how the actual processes cover the characteristics. It is rather interesting to see if and how companies or researchers implement them and to analyze if there are major differences or similarities between the systems. As a first step, the following subsections describe how these actions are specified based on the W3C VC draft.

4.3.1 Issue

Issuance of credentials is the starting point for a credentialing system's workflow. The W3C characteristics have several statements dealing with this topic. *CIss1* states that any *Issuer* can create verifiable credentials about any subject. In the W3C terminology, a *subject* is "An entity about which claims are made" [33]. Therefore, anything can be credentialed, but there has to be the role of a *Holder* that assembles these credentials (see Subsection 4.3.6).
These credentials should be issued and created in a manner that is tamper-proof and privacy-respecting (*CSys6*). A credential contains several statements a *Verifier* can verify to prove a *Holder's* abilities. For further support and to enhance privacy, credentials that support selective disclosure can be issued (*CIss3*). The system has to support this mechanism as well and provide *Issuers* a way to publish their credentials with that feature (*CIss2*). Lastly, credentials have to be serializable and machine-readable in a "(...) deterministic, bi-directional, and lossless" way [33]. The data has to be serialized compliant to the W3C data model.

4.3.2 Store

Once a credential is issued, it has to be stored in the system. Table 4.1 only shows one characteristic that has to be implemented: *CHol5*. It states that the *Holder* has to be enabled to store the credential in any location she wants without inflicting verifiability and the *Issuer* knowing the access and location of the credential. Once the credential is issued, only the *Holder* has the right to move them.

4.3.3 Refresh

Certain credentials will have an expiry date such as International Organization for Standardization (ISO) or Deutsches Institut für Normung (DIN) certifications. Companies can be compliant to e.g. ISO 27001 for a certain time and then lose their status due to an audit.

Furthermore, these credentials can have an expiry date as well that can be postponed with another certification process. Consequently, a refreshing mechanism has to be in place. An *Issuer* should be able to refresh its issued credentials by providing a service for that (*CIss7*).

4.3.4 Revoke

Following the example in Subsection 4.3.3, a company loses its compliance status to ISO 27001 or it turns out that there were errors during the audit. An *Issuer* should have the ability to revoke its credentials, therefore making them invalid. To enable revocation, the *Issuer* has to be able to issue revocable verifiable credentials (*CIss4*). Furthermore, it has to be distinguished between revocation because of invalidity (e.g. a company has lost its ISO 27001 compliance) or due to cryptographic reasons, e.g key compromisation (*CIss6*). Distinguishing between these two types is important for the integrity of the credential. If a revocation happens because there has been an attack on the *Holder*, the *Holder's* reputation is not inflicted. If the *Holder* has invalidated its credential on a subject-specific basis, it inflicts her reputation.
Furthermore, the reason for revocation can be disclosed by *Issuers* (*CIss5*) but revocation should not leak any identifying information about the *Holder*, *Issuer* or *Verifier* that might be involved in this process (*CSys9*).

4.3.5 Receive

Once the *Issuer* has issued a credential and it is stored in the *System*, the *Holder* has to receive it. According to the characteristics, the *Holder* is able to receive a credential from anyone (*CHol2*). This also means that anyone in the *System* should have the ability to become an *Issuer*. Issuing and holding credentials are decoupled from proving that a claim is actually valid. It is the *Verifier's* responsibility to prove that the content of a credential is valid.

4.3.6 Assemble

Once a credential is received by the *Holder*, she can assemble collections of credentials. Since the *Holder* is able to receive credentials by any *Issuer*, there can be clusters of credentials that belong together. There might be credentials that are issued by a university or by a company that certifies e.g. a successful study program or further education. These credentials should then be bound together "into a single Artifact" [33], a so called *verifiable presentation* (*CHol1*). A verifiable presentation can be seen as a piece of information that is aggregated and shared by the *Holder*. The main characteristic at this point is that the verifiable presentation is both tamper-proof and authorship of the data can be trusted. [33].

4.3.7 Interact

With all set up use cases in place, interaction is the next topic that has to be dealt with. Intuitively, the first interaction a *Holder* should be able to do is sharing her verifiable credentials. However, even before sharing items in the *System*, a *Holder* should be able to interact both with any *Issuer* and *Verifier* through any user agent (*CHol3*). This assures that *Holders* can both receive and share items from and with every other role in the *System*. By sharing verifiable credentials, the *System* has to make sure that privacy is not infringed. Although a *Holder* might be identified, the verifying party might want to remain unidentified (*CHol4*). This is important when it comes to verification by various parties whom the *Holder* might have a relationship with. It assures that the process of verification remains integer and both *Holder* and *Verifier* cannot interact with each other through different channels. Another privacy related issue is sharing verifiable credentials with selective disclosure (*CHol7*). Selective disclosure allows the *Holder* to share certain parts of a credential, e.g. a single claim, with a *Verifier*. Furthermore, the amount of information can be reduced to single boolean values such as *true* and *false*. In the end, the result is the same as with a manual check by the *Verifier*: If the credential suffices the requirements, the value is true or greater than. Else, the result is false or lesser than (*CSys7*). With selective disclosure, data protection and privacy are drastically enhanced and *Holders* are enabled to share only the information that is required.
Another rather intuitive characteristic is *CHol6*: "Holders can present verifiable presentations to any verifier without affecting authenticity of the claims and without revealing that action to the Issuer." [33]. Therefore, verifiable credentials will not be consumed by the action and *Issuers* stay out of the interaction between *Holders* and *Verifiers*.

4.3.8 Verify

Verification is a crucial part in the digital credentialing process. Only verification proves that claims are valid and benefits can be granted to requesting *Holders*. *CVer1* states that every "(...) Verifier can verify verifiable presentations from any Holder, containing proofs of claims from any Issuer" [33]. Consequently, there is no restriction in the system that hinders *Verifiers* to only process claims from certain sources (*Issuers*) or of a certain type. Each credential that contains a claim can be checked for validity. Additionally, each *Verifier* is allowed to check the authenticity of each verifiable presentation (*CSys2*). Different from *CVer1*, this characteristic aims at checking if cryptographic and interaction elements are correct, not the actual credential itself. Authentic credentials are ones that are not tampered or forged in any way.
Lastly, the verification process has to be independent of the interaction process (*CSys3*). Verifiable presentations should be designed in a way that allows verification without further interaction. Proofs should be conducted without further addressing the *Holder* or *Issuer*.

4.3.9 Surroundings

Apart from the above mentioned use cases that are all related to "do something" with or within the *System*, there are surroundings that address the environment rather than the actual processes. *CSys5* states that both data model and serialization have to be extendable with "minimal coordination" [33]. Furthermore, there is no registration process in place for any role (*Issuer*, *Verifier*, *Holder*). Every user can act as every role without approval by authorities (*CSys1*). Since credentialing systems involve a high amount of trust, this is an interesting topic to further investigate in the following sections. Identification and trust have to be established to process the above mentioned use cases. Regarding trust establishment, *CSys1* says that "(...) trust involved is bilateral between parties" [33], meaning that interacting parties have to trust each other rather than trusting authorities that bail for registered users.

4.4 Creating a Framework for System Comparison

In the previous sections, requirements and actions were identified by using the W3C *Verifiable Credentials* draft. Naturally, these two categories have to be incorporated into the framework, since they represent the logic that is implemented in such a system. Table 4.2 shows the two categories "Actions" and "Requirements" on top, followed by "System" and "Business".
Apart from the business logic layer (category *Business*), the framework aims to compare the verifiable credential implementation on a more technical layer as well. Therefore, the *System* category focuses on exactly these items. The most important item in this category is the data model. Here, the framework allows the user to denote if the implementation uses e.g. a JSON data model, PDF or XML files. Depending on the data model, the solutions will differ immensely from each other. For instance, a single PDF file does not have the extensibility of a JSON model. A PDF file neither provides the same machine-readability as other document formats.
Following the data model, permission and data storage will be examined. Both items correlate with each other. A blockchain-based system might not have a permission-model in place due to the public-key-infrastructure and cryptographic identification methods. However, there are permissioned blockchain systems such as IBM’s Hyperledger. With a permissioned blockchain, the system is locked behind a barrier where users have to get permission to enter the system. This could be useful for internal, non-public systems. The data storage model is important as it tells the reader if the system takes a centralized (e.g. central administered database) or a decentralized approach. Both have their advantages and disadvantages in terms of administration, data protection and longevity.
The following three items describe if any reference or compatibility is provided. References namely aims at the W3C draft or the OpenCerts reference architecture (see next chapter) but is framed openly to support different approaches which were not mentioned in this thesis. "Macro- / Micro-Credential Compatibility" states if the system supports either both or one of these domains. Lastly, the GDPR compliance flag indicates if the developers aim

Category	Key	Value	Example
Requirements	Issue Claim	Boolean	Yes
	Assert Claim	Boolean	No
	Verify Claim	Boolean	Yes
	Store Claim	Boolean	Yes
	Move Claim	Boolean	No
	Retrieve Claim	Boolean	Yes
	Revoke Claim	Boolean	Yes
Actions	Issue	String	Process description
	Store / Move Claim	String	Process description
	Refresh	String	Process description
	Revoke	String	Process description
	Receive	String	Process description
	Assemble	String	Process description
	Interact	String	Process description
	Verify	String	Process description
	Surroundings	String	Process description
System	Data model	String	JSON Data Model for credentials
	Permission	String	Permissionless
	Data Storage Model	String	Decentralized Blockchain
	References	String	W3C Draft compliant
	Macro- / Micro-Credential Compatibility	String	Macro-credentials only
	GDPR Compliance	Boolean	Yes
	API Available	Boolean	Yes
	Meta Data	String	Can be stored in Data Model
	Identification Method	String	Biometrical Identification
	Trust Model	String	PKI Infrastructure, peer-to-peer trust model
Business	Business Model / Pricing	String	Subscription Model
	Usage KPIs	String	500 issued credentials total
	Cooperations / Partners	String	Deployed at University of Nicosia
	Maturity	Maturity Model	Initial
	Target Industry	String	Educational Sector

Table 4.2: Framework for comparison of existing digital credentialing solutions.

to be or already are compliant to the General Data Protection Regulation of the European Union. Since privacy and data protection has become a more prominent factor in software development in recent years, this cannot be neglected in a framework.
The last four items in the system section deal with the presence of an aApplication Programming Interface (API), meta data, the identification method as a connection to "real" identities (e.g. biometric identification) and the trust model. The trust model also strongly correlates to the identification method, permission and GDPR compliance, since the trust model explains how peers can trust each other. There can be an authority that generates trust by issuing certificates for e.g. universities. However, trust can be generated by trusting peers directly without implementing an authority.
In the last category, "Business", the framework shows how the investigated solutions monetize their software and in which stage of development the system is. For monetization, a look at the pricing model is inevitable: Who is charged how much? Is there a monthly fee or a fixed price for the amount of issuances? This information will be provided through the pricing / business model field.
For identifying the state of the system, Key Performance Indicators (KPI) will be investigated. Usually, KPIs provided by the developing company are used for marketing, which is why the maturity cannot be defined by only looking at these numbers. That's where cooperations and partners come into play: By checking who is investing either financially or technologically into the platform, the reader gets an opinion about how much effort is put into the system. For instance: A system backed by several universities might be more mature and market-relevant than a system that is built by an independent developer on his or her own.

To further address this issue, the "maturity" flag serves as an indicator of how far the development process has come to the date of writing this thesis. According to [37], there are five levels of maturity:

1. *Initial*. In the first phase, there are few processes in place and the overall workflow is described as "ad hoc" and "chaotic" [37]. Yet, teams usually succeed in releasing a product during that phase by over-commitment and exceeding budget limitations. In this stage, processes and success are not repeatable due to the lack of fixed procedures.

2. *Managed*. Stage-two teams have established processes and can adhere to them, even in stressful situations. Stakeholders are involved in the development process and monitor the team's progress. Furthermore, the team is comprised of "skilled people" and it is able to allocate resources in a sensible manner to generate "controlled outputs" [37].

3. *Defined*. Similar to stage two, processes represent the key factor in the third one: They are "well characterized" and fully understood by the team. Furthermore, processes "(...) are described in standards, procedures, tools and methods". The main distinction between stage two and three is the way processes are customized from the

Area of Effect	Level				
	1. Initial	2. Managed	3. Defined	4.Quantitatively Managed	5. Optimizing
Technology	• Ad hoc, chaotic • Emerging • Lack of understanding	• Methodology establishment • Controlled and coordinated • Reactive	• Standardized and documented • Proactive	• Quality metrics establishment • Consolidated and reliable	• Continuous improvement • Share of knowledge and information
Market	• Focus on function • High cost	• Focus on reliability • Transactional customers • Broad no-target promotion Regulation	• Focus on assured delivery of services • Prices settle down • Requirements are measured	• Standard services • Price with incentives and outcome metrics • Customers are grouped with profiles • Promotion is targeted	• Empathy in dealing with emerging business needs • Create the product special influents in industry
Regulation	• Less supervision • Competition is forbidden	• Rules have been borrowed from related domains	• Regulation rules and laws are defined	• Measurements on regulation is set up • Competition is encouraged under supervision	• Free competition • Market based on well-established legal system

Table 4.3: CMMI Levels of maturity taken from [38]. The headlines for each level have been adapted to the CMMI definition. In [38], the author uses different headlines.

standard repertoire a company has. By taking standard processes and customizing them to the team's workflow, the team achieves more consistency, whereas stage two processes can be individually for each and every team without the guiding standards a company offers in stage three. Additionally, processes are described as more robust and have to be more obeyed than in stage two [37].

4. *Quantitatively managed*. As processes are standardized, tailored and in place, quantitative measurement comes into play. For further refinement and understanding where bottlenecks are, processes are enhanced with quantitative objects that "(...) are based on the needs of the customer, end users, organization, and process implementers." These objects are relevant throughout the project's lifetime. With quantitative measurement in place, processes can be controlled for performance and their runtime becomes predictable. This predictability distinguishes stage three from stage four [37].

5. *Optimizing*. With quantitative measures in place and understanding the processes' "variation" as well as their "causes of (...) outcomes", companies optimize themselves continually by means of technological and innovative advancement. This requires constant monitoring and revision of how processes match "performance objectives". Dissimilar to stage four, where the main goal was to understand processes from a measurement point, stage five aims at optimization efforts throughout the whole company involving several projects. Each process is part of leveraging the company's overall performance, whereas stage four aimed at understanding and measuring single processes and their dependencies [37].

Table 4.3 summarizes the above described levels of maturity and adds information about the corresponding market situation as well as regulations that are in place at certain stages.

5 Technical Examination of Digital Credentialing Specifications and Identification Methods

In the previous chapter we created the framework that can be used as a basis to compare existing solutions. The framework itself is based on principles that are adopted and derived from the Verifiable Credentials specification [33]. Yet, there has not been any technical documentation on how these principles are implemented or how data models linked to the specification are defined. Therefore, this chapter presents the technical foundations for both the Verifiable Credentials draft and the OpenBadges specification. Another purpose of this chapter is to create an understanding why the VC draft suits better as a foundation for the framework than the OpenBadges verification. This will be explained and demonstrated from a technical point of view, since the idea behind both specifications are partly similar. Additionally, a governmental approach from Singapore is presented. Different from OpenBadges or Verifiable Credentials, OpenCerts specification is tightly related to an ongoing developing process.
Lastly, this chapter includes a section about how identification and authentication can be provided. The W3C specification explicitly excludes this component and leaves this layer to the companies developing the systems. Consequently, the identification methods differ depending on the provider. For that reason, two concepts are explained that deal with the identification process: eIDAS, which has been mentioned already in Chapter 2, and the concept of *Decentralized Identifiers* that will be mentioned in Section 5.4 as well.

5.1 Investigating the W3C Verifiable Credentials Data Model

Listing 5.1 depicts an example implementation of a macro-credential that could have been issued by the Technical University of Munich (TUM). The example is an adaption and aggregation of several listings taken from [33] to demonstrate how a macro-credential such as an university degree could look like.
At the beginning of each macro-credential compliant with the W3C VC draft there is a section called `context`. Similar to XML files, this section establishes a common understanding for machines of what to expect in the following lines. Behind the URL of the `context` field is a schema which defines data types, thus lets systems understand the actual *context* of the document. By design, these values have to beUnified Resource Identifier (URI). In this example, the context indicates that the following data is about credentials (line 2) and lists an example credential (line 3).

Next, there is an identifier flag (`id`) that serves as a unique identification for the whole data set. In this case, the identifier is URI that points to the credential in the TUM credential repository. Identifiers can either be a URI as demonstrated in line 6, but they can also be a Decentralized Identifier (DID) which will be explained in a later section. Identifiers can be used various times throughout the document, whenever there is the necessity of identifying an involved party.
In line 7, the `type` declares what the document is actually about. Similar to `context`, the `type` key allows multiple values and therefore uses an array. According to the definition by the W3C, the credential type always has to state a broad and a narrow definition [33]. In this example, the broad type definition is "VerifiableCredential" indicating that this document is a credential and can be verified. The narrower definition is "UniversityDegree", indicating that this credential was issued by a university such as the TUM. Since the `type` key is also used at different levels in the document, it always has to have a specific value according to the level where it is used. For instance: At the first level (line 7), the type declares that the whole document is a "VerifiableCredential". In line 26 however, the type value is "RsaSignature2018" and stands for the type of proof that has to be executed to verify the credential. Without the `type` key at the very first level, the whole verifiable credential loses its verifiability and therefore its credibility.
In lines eight to twelve, there is a syntactic error embedded in the document for demonstration purposes. Technically, there can only be one single entry for an issuer. Yet, there is an `issuer2` key present that is wrong. The reason behind this is that the issuer can be denoted in two ways: First, as a URI such as the `https://tum.de` address. Secondly, as an object with a name and an id. Both methods are valid and identify the issuer correctly. It is up to the issuer how it wants to present itself in the document.
With the `issuanceDate` in line 13, the document states a point in time (usually in the future) from which the credential is valid. As of writing this thesis, the W3C VC draft's version is the January 15, 2020. In future versions, the `issuanceDate` will be replaced by a `validFrom` and `issued` key-set that explains the thought behind the issuance date better. Intuitively, the issuance date would indicate when the certificate has been issued rather than from what point in time it has been valid. However, all three values have to be compliant to the RFC-3339 standardization of describing dates so that technical device is able read to the correct date.
Below the `issuanceDate` there is the `credentialSubject`. As described in earlier sections, the credential subject is the actual person, organization or object about who or which the claims in the credential have been made. In this example, the credential subject is the author of this thesis as the `id` value shows. Below the identifier, the claims made about the subject are present. Although only one claim is made, there can be several more attached to this document. Even more, there can be various claims about several subjects stated in the credential subject object. An example made in the draft is that a governmental office issues a credential about a wedding that shows two persons who are now spouses. The narrow credential type is a "RelationshipCredential" and the subject contains both spouses who have each other's *Decentralized Identifier* (DID) as a key-value pair written

in the credential. The `credentialSubject` serves as the core part of the document as it contains the claims that are made about the entity and has to be included in each document. Otherwise, the credential becomes invalid.
Another feature included in the example is the `credentialStatus` object. Since credentials can become invalid or their issuance might be erroneous, issuers can attach a reference to their own status list. The `credentialStatus` is comprised of an identity and a type key. Both are mandatory inside the status object for the reason of uniquely identifying the credential's status. By design, the type references an endpoint where a list of statuses is stored. The ID points to an exact object in this list. Therefore, the correct value can be retrieved and the status read correctly.
Line 25 to 35 show an example of the so called "advanced concepts" of the W3C draft [33]. The `termsOfUse` object regulates the credential's usage when it is shared with relying parties such as verifiers. Only issuers and holders are allowed to declare terms of use and both are restricted to a different type document: The issuer is allowed to include terms of use in verifiable credentials, whereas the holder may only include terms of use in verifiable presentations. Again, verifiable presentations are a collection of verifiable credentials that the holder can aggregate and share with a verifier, whereas a credential is a single document that is issued by an issuer. Furthermore, the terms of use can state three different actions: "an obligation", "a permission" or "a prohibition" [33]. The obligation tells verifiers what they have to do when receiving the credential. The permission tells the verifiers what they are allowed to do and the prohibition includes actions that are forbidden to do with the document. As stated in the W3C draft, this section has no executive power such as destroying the document upon violating the terms of use. However, violation could lead to legal inquiries due to the terms of use's legal binding character.
The terms of use object is comprised of a `type` that specifies a policy (e.g. `IssuerPolicy`). Furthermore, it has an `id` that states the path to the policy. This, however, is optional. The `profile` key neither is mandatory, it only serves as a further source of information for machines for navigating to the policy. Below this, there is the `prohibition` object. In this, the `assigner` is the subject which states the terms of use (here the TUM). The assigner has an `assignee` field which specifies who has to be compliant to the terms of use (here: `allVerifiers`). The `target` defines an object the terms of use deal with (the credential in this document) and the action shows that *archival* is forbidden. In other words: The terms of use policy, issued by the TUM, prohibits every verifier to archive this exact credential. Another example could be that the holder forbids to do third-party correlation with the shared credential [33].
Lastly, this example shows a `proof` object containing data for the verifier about how to verify the document. Each credential has to provide at least one cryptographic proof that can either be *embedded* or *external* [33]. The external proof "(...) wraps an expression of this data model, such as a JSON Web Token (...)" [33]. The embedded one has to specify the correct method for the verifier to prove the document. Listing 5.1 shows an embedded proof containing the data relevant for execution by the verifier. As the `type` indicates, the contained proof is a RSA Signature that was `created` in 2019. The `proofPurpose` states

the reason why this document should be proved and the `verificationMethod` directs the verifier to a repository where the exact information about the to-be-chosen method is located. Furthermore, this proof contains a JSON Web Signature attached to the document that ensures the integrity of the document. In this exemplary case, the verifier would re-sign the document and check if the signatures are matching. If so, the document is valid, else it would be corrupted.
The proof object can vary depending on the chosen proof mechanism. According to the W3C VC draft, there is no standardization desired regarding the proof mechanisms. However, issuers have to provide enough information for verifiers to execute the mechanism and certify validity. Otherwise, the document would not be valid.

Listing 5.2 shows another example of how the W3C VC draft can be used. Instead of a university degree, the credential certifies that a student is currently enrolled at the given university. Therefore, the credential type has changed to `UniversityEnrollment`. The `credentialSubject` contains claims about what the student is currently pursuing and how far she has progressed. In Germany, it is common that students have to re-enroll for each term by paying a certain fee for public transportation and student welfare. Once the payment is received by the university, a student's enrollment validity is extended until the end of the upcoming term. Listing 5.2 demonstrates this exact use case. As already described above, the student's information is contained in the `credentialSubject`. Recently, she has paid the semester fee and has received a new expiration date which is denoted in the same key (`expirationDate`). With the usage of the expiration date key, the issuer has the possibility to attach a refresh service that extends the expiration date once the credential is either expired or about to expire. Here, the `refreshService` points to a URI that allows the issuer to refresh the credential. Furthermore, it states that the refresh service's type is `StudentIdRefreshService`. With the implementation of a refresh service come certain considerations. First, refreshing a credential should only be necessary if there is no status attached to it. Secondly, only the issuer is allowed to attach a refresh service. If the refresh service is attached to the verifiable credential, both the holder and the verifier can refresh it. If it is only attached to the verifiable presentation, the holder solely has the ability to refresh the document [33]. Lastly, refreshing a credential creates a connection between the issuer and the verifier, which is not intended by the W3C. It provides a mean to bypass the holder who should actually be in the position to choose if the credential is shared (again) with a verifier. Therefore, the implementers should prioritize the status mechanism over the refresh service [33].
Another use case for this document is to prove the regional public transportation service in Munich that the subject is enrolled at the given university. However, the public transportation office needs further documents besides the enrollment. It needs a physical student identification card and the invoice the university has given the student for her payment. Both items can be provided through the `evidence` object beginning in line 28. Evidences are different from cryptographic proofs. Although the enrollment can be verified by the attached proof, the transportation provider could still reject it due to the lack of further documents. Evidences are meant to fill this gap. Independent from the type of

Listing 5.1: Example Verifiable Credential adapted from [33].

```
{
      "@context": [
            "https://www.w3.org/2018/credentials/v1",
            "https://www.w3.org/2018/credentials/examples/v1"
      ],
      "id": "http://tum.de/credentials/3732",
      "type": ["VerifiableCredential", "UniversityDegree"],
      "issuer": "https://tum.de",
      "issuer2": {
            "id": "did:tum:76e12ec712ebc6f1c221ebfeb1f",
            "name": "Technical University of Munich"
      },
      "issuanceDate": "2020-01-02T10:09:59Z",
      "credentialSubject": {
            "id": "did:gerbershagen:abcd1f712ebc6f1c276e12ec21",
            "degree": {
                  "type": "MasterDegree",
                  "name": "Master of Science in Information Systems"
            }
      },
      "credentialStatus": {
            "id": "https://tum.de/credentialStatusList/01",
            "type": "CredentialStatusList2019"
      },
      "termsOfUse": [{
            "type": "IssuerPolicy",
            "id": "http://tum.de/policies/credential/4",
            "profile": "http://tum.de/profiles/credential",
            "prohibition": [{
                  "assigner": "did:tum:76e12ec712ebc6f1c221ebfeb1f",
                  "assignee": "AllVerifiers",
                  "target": "http://tum.de/credentials/3732",
                  "action": ["Archival"]
            }]
      }],
      "proof": {
            "type": "RsaSignature2018",
            "created": "2019-06-10T10:09:59Z",
            "proofPurpose": "assertionMethod",
            "verificationMethod": "https://tum.de/credAssertion/keys/1",
            "jws": "eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOls"
      }
}
```

evidence (here: `DocumentVerification` and `Invoice`), the provided information has to be enough that it meets the "(...) confidence requirements for relying on the credential" [33]. Furthermore, evidences can only be gathered and attached by the issuer. The subject can pass the information to the issuer from her to attach it to the credential. In this example, the public transportation would see the invoice attached to the credential and perform a manual check by one of its employees resulting in either granting the transportation ticket or not.

Listing 5.2: Evidence, refresh service and expiration date example adapted from [33].

```
{
      "@context": [
            "https://www.w3.org/2018/credentials/v1",
            "https://www.w3.org/2018/credentials/examples/v1"
      ],
      "id": "http://tum.de/credentials/3732",
      "type": ["VerifiableCredential", "UniversityEnrollment"],
      "issuer": "https://tum.de",
      "issuanceDate": "2020-04-01T10:09:59Z",
      "expirationDate": "2020-09-30T23:59:59Z",
      "refreshService": {
            "id": "https://tum.de/refresh/3732",
            "type": "StudentIdRefreshService"
      },
      "credentialSubject": {
            "id": "did:tum:ebfeb1f712ebc6f1c276e12ec21",
            "studentEnrollment": {
                  "id": "did:gerbershagen:abcd1f712ebc6f1c276e12ec21",
                  "name": "Dominik Gerbershagen",
                  "studyProgram": "Master of Science Information Systems",
                  "semester": 6
            }
      },
      "credentialStatus": {
            "id": "https://tum.de/credentialStatusList/01",
            "type": "CredentialStatusList2019"
      },
      "evidence": [{
            "id": "https://tum.de/evidence/f2aeec97-fc0d-42bf-8ca7-0548192d
               4231",
            "type": ["DocumentVerification"],
            "verifier": "https://tum.de/issuers/14",
            "evidenceDocument": "StudentID",
```

```
            "subjectPresence": "Physical",
            "documentPresence": "Physical"
      },{
            "id": "https://tum.de/evidence/f2aeec97-fc0d-42bf-8ca7-0548192
               dxyzab",
            "type": ["Invoice"],
            "verifier": "https://tum.de/issuers/14",
            "evidenceDocument": "SemesterFeePayment",
            "subjectPresence": "Digital",
            "documentPresence": "Digital"
      }],
      "proof": {
            "type": "RsaSignature2018",
            "created": "2019-06-10T10:09:59Z",
            "proofPurpose": "assertionMethod",
            "verificationMethod": "https://tum.de/credAssertion/keys/1",
            "jws": "eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOls"
      }
}
```

Both listings have not included the `holder` key in their credentials. The reason behind this is an implicit subject-holder relationship that is shown in Figure 5.1. The most common use case, as presented in the graph, is that the holder and the subject are both one entity. According to [33], the verifiable presentation containing the credentials is signed by the holder. The Verifier sees that all subjects in the verifiable presentation are also about the holder who signed the verifiable presentation. Therefore, neither the `holder` key nor the verifiable presentation are not modeled in the credential. However, the standard suits not only university credential use cases, but also so that contain subjects detached from the holder. As an exemplary case, the W3C draft mentions parents and their children. An underage child is not able to act for itself, wherefore the parents have to act for it. In this case, the child is the subject of the credential, but not the holder. Listing 5.3 shows a snippet of a child's credential that demonstrates how relationships between parents and children can be established. The first id is assigned to the child with another key that states that it is under 16 years old. Therefore, the parent object is included which depicts the relationship between another subject identified by its own DID. Furthermore, the type of relationship is described as `mother`.

Traversing the graph in Figure 5.1 it can be seen that there are many more use cases where the holder and the subject are different from each other. However, in most cases the actual `holder` key has not to be modeled. An example when it is necessary to include the key is when the "Issuer Independently Authorises [the] Holder" [33]. Here, the issuer (e.g. a governmental office) passes information about a subject to e.g. law enforcement which then executes an order such as visiting the subject at the given address. The subject is

completely independent from the holder, which is why the holder key has to be included in the credential.

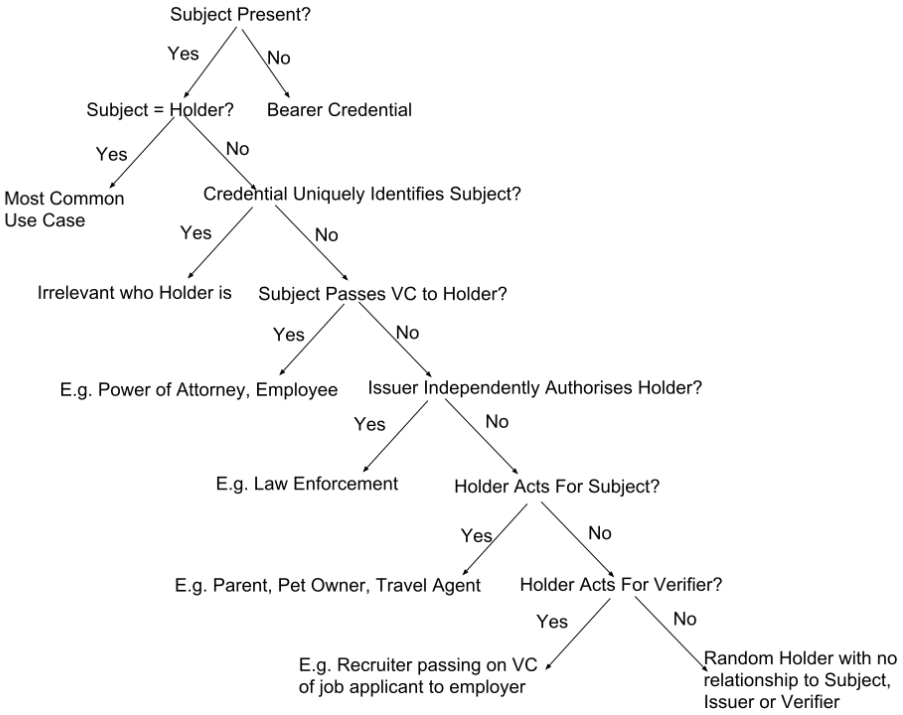


Figure 5.1: Subject-Holder relationship in the W3C VC Draft [33].

Listing 5.3: Child and parents relationship in credential subject. Example adapted from [33].

```
{
      "credentialSubject": {
            "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
            "ageUnder": 16,
            "parent": {
                  "id": "did:example:ebfeb1c276e12ec211f712ebc6f",
                  "type": "Mother"
            }
      }
}
```

Given all the above mentioned and further available components of the W3C VC draft, a trust model is formed. Figure 5.2 shows how the in Chapter 4 explained roles trust each other. Each arrow represents the direction of how parties establish trust. For instance: The arrows denoted with the number *2* indicates that the holder, issuer and the verifier all together trust the verifiable data registry. Since there is no arrow in the opposite direction,

trust is established only one-directional.
A part that has not been mentioned yet is the `verifiable data registry`. The registry is a storage system that contains metadata such as the certificate schemes, revocation reasons or issuer keys. Practically, the holder is registering her identifier in the verifiable registry so that the issuer can retrieve her id there. Once the issuer is about to issue a certificate, she retrieves the key and issues the certificate including the holder's ID [1]. Additionally, the issuer retrieves credential schemes such as the one in the `context` field provided in Listing 5.1. Overall, the verifiable data registry functions as a common foundation of understand that is trusted by each party: the holder, issuer and the verifier. In Figure 5.2, this line of trust is represented by the arrow denoted with *2*.
Arrow *1* shows that the verifier has to trust that the issuer is the one who issued the credential. There are two ways to fortify this bond of trust:

- The issuer includes a proof for the verifier so that she is able to not only trust but prove that she has issued the credential.
- The verifier trusts that the credential is stored and transmitted tamper-resistant and the issuer is making it very clear that she is the one who issued it. Compared to option one, this approach requires a higher level of trust since there is no proof attached for verification.

Furthermore, both the holder and the verifier trust the issuer to publish correct credentials. More specifically, the issuer is trusted to only publish a credential about a subject that is true and furthermore is able and willing to revoke it in case the credential becomes false (c.f. arrow 3 in Fig. 5.2).
Lastly, the holder trusts the repository (e.g. a wallet or vault) that credentials are securely stored, not unwillingly published and the credentials will not become corrupted.
Given this trust model, two implications can be made:

- The issuer and the verifier do not have to trust the repository. Only the holder has to.
- The issuer does not need to know or trust the verifier.

[33]

5.2 Analysis of the OpenCerts Specification

Different from the W3C VC draft, OpenCerts represents a framework for implementation. Where the W3C VC draft offers a broad variety of concepts and unties these concepts from technological implementation, the OpenCerts framework defines a foundation for developers to bootstrap a digital credentialing software. A first indicator for this is the mentioned compliance to the OpenAttestation specification [39] [40]. OpenAttestation, according to the GitHub repository, is a "(...) notary framework for any document types on

[1] Note that the holder can be different or equal to the subject as explained earlier in the this section.

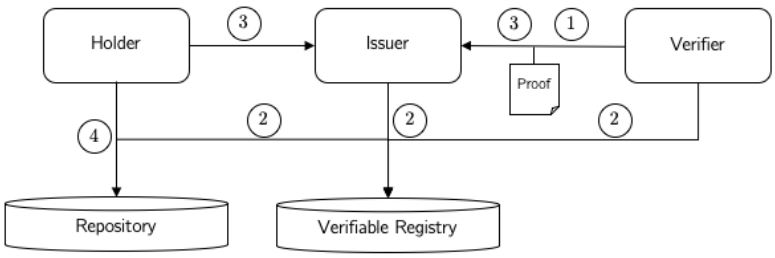


Figure 5.2: Trust model in the W3C VC Draft [33].

the blockchain" [40]. By being compliant to this specification, the underlying technology has to be the blockchain. Furthermore, the author states in the first chapter that certificates are stored in blockchain as well [39].

Listing 5.4 shows an example of a digital credential compliant to the OpenCerts specification. Similar to the listings in section 5.1, the OpenCerts data model includes an `id` field that can be the same as a serial number on a printed credential. Alternatively, the id can be a Universalliy Unique Identifier (UUID) that has been generated randomly.
Next, there is the `name` of the certificate and its issuance date. `Name` has to be a string and the `issuedOn` is a timestamp compliant to RFC-3339. Afterwards, the `issuer` forms the first object inside the credential. Similar to the W3C VC draft, the issuer contains several keys that identify the entity. `Name`, `url`, `email` and `phone` intuitively describe information that can be provided by the issuer for either identifying or contacting the entity. However, there is no `id` key such as the W3C VC draft offers. Instead of the `id` key, the OpenCerts specification offers the option to include a DID to uniquely identify the issuing entity. Related to the issuer is a `certificateStore` that references a smart contract address which can be obtained by (...) deploying an instance of [it]" [39]. In the certificate store, digital credentials are stored after being issued.
Since the issuer field is an array of objects, there can be more than one issuers included in a certificate. This allows, for instance, to include both the Technical University of Munich and the Faculty of Informatics. However, both objects are required to provide a certificate store and a name. How exactly a certificate can be issued by two entities has not been further explained as of writing this thesis.
Following the issuer, the `recipient` object defines the credential's subject. Again, `name`, `phone` and `email` intuitively describe information about the recipient and offer the possibility to contact that entity. Optionally, a DID can be attached to the recipient for further

Listing 5.4: Digital Credential data model in OpenCerts specification. Example adapted from [39]

```
{
  "id": "2018091259",
  "name": "Master of Information Systems",
  "issuedOn": "2020-03-15T23:59:32+08:00",
  "issuers": [{
    "name": "Technical University of Munich",
    "url": "https://tum.de",
    "certificateStore": "0x1989a05B320186f5fAc590fFf64730FC9099Bc7b",
    "did": "did:tum:21234567890",
    "email": "certificates@tum.de",
    "phone": "+4908912345678"
  }],
  "recipient": {
    "name": "Dominik Gerbershagen",
    "email": "dominik@mail.com",
    "phone": "+4908965431",
    "did": "did:gerbershagen:123456789"
  },
  "transcript": [{
      "name": "Master Thesis Digital Credentialing",
      "grade": "undefined",
      "courseCredit": 30,
      "courseCode": "MA-DC",
      "url": "https://in.tum.de/masterthesis",
      "description": "State of the art and practice of digital credentialing
          .",
      "score": 120
    },
    {
      "name": "Advanced Seminar",
      "courseCode": "ADSEM",
      "url": "https://wwwmatthes.in.tum.de/pages/rwzby1tlqhn9/Advanced-
         Seminar",
      "description": "Participating and communicating."
    }
  ],
  "additionalData": {
    "signature": "data:image/jpeg;base64...."
  }
}
```

identification. Dissimilar to the W3C VC draft, the recipient object does not contain claims. The OpenCerts specification differentiates between the subject and the claims that are made about it.
Below the `recipient` is the `transcript` object which is an array again. In this array, several transcripts can be stored that each contain a claim, e.g. line 20: "Master Thesis Digital Credentialing". For each transcript, the `name` property is mandatory to tell the verifier what claim is made in the transcript. The specification also defines additional keys such as `grade`, `courseCredit`, `courseCode`, `url`, `description` and `score`. With these additional keys, the transcript can be described in more detail. Even establishing connection to e.g. a university repository through the `url` field is possible. Furthermore, there can be additional data added to the `transcript` object that has not been defined by the OpenCerts designers. Since there is no information provided if different types of claims such as the under-age certificate mentioned in Listing 5.3 or the child-parent relationship can be created, the application of the specification to such use cases has to be conducted by future work.
Lastly, there is the `additionalData` object that allows to include for instance a handwritten signature as an image file. The specification also lists examples such as testimonials, awards and activities that have not found their way into the `transcript` object [39].
Although the overall data model resembles the W3C VC one, a major concept is not included: the `proof` object. From the provided information on the website and the documentation, the OpenCerts specification does not provide a cryptographic proof inside the certificate. Rather, the mechanism relies on the implementation on the Ethereum blockchain and optionally a trusted registry. For "(...) institutions that require a higher level of identity assurance" [39], the specification allows to create a connection between the issuer and her Domain Name System (DNS) domain. Usually, the described institutions have their domain names certified by a Certification Authority (CA) and therefore can be trusted when accessing their web page, for example. In these certificates, a `TXT` field can be inserted under the root domain (e.g. inside the tum.de domain) that contains a link to the TUM's document store (see Listing 5.5). The document store is an Ethereum smart contract address where the credentials are issued to [41]. Inside the credential, the `issuer` object can then contain another object called `identityProof` that references the root domain (see Listing 5.6 line 5-7) and the DNS type. Upon verification, the system queries the given domain and retrieves the certificate that contains the document store address and compares this with the credential's document store address. If they are the same, the identity is proven.

Listing 5.5: TXT entry in the DNS certificate. Example taken from [41].

```
1 TXT openatts net=ethereum netId=1 addr=0x007d40224f6562461633ccfbaffd359ebb
    2fc9ba
```

Apart from the data model, the OpenCerts framework incorporates several methods that facilitates digital credentialing. One of these methods is a document renderer that helps issuers to generate templates and deploy them on custom domains. With these templates, the data model in Listing 5.4 is used and can be customized to the needs of the institution

Listing 5.6: Identity proof inside the OpenCerts credential. Example taken from [41].

```
"issuers": [
{
   "network": "ETHEREUM",
   "documentStore": "0x9178F546D3FF57D7A6352bD61B80cCCD46199C2d",
   "identityProof": {
      "type": "DNS-TXT",
      "location": "openattestation.com"
   }
}
]
```

and create a corporate-wide conform certificate. Furthermore, a complete guideline and framework is provided that helps to create and issue the certificates onto the Ethereum blockchain. However, the specification is yet in version 2.0 and does not feature status updates or revocation. Additionally, there is no trust model provided as in the W3C VC draft. For digitizing printed credentials (e.g. diplomas) and storing them online for verification, the OpenCerts specification is a good starting point. Still, to create an ecosystem as [28] proposes, there has to be a more refined model of trust and a concept for revocation and status updates.

5.3 Examining the IMS Mozilla OpenBadges Specification

In the previous two sections, two specifications for macro-credentials have been presented. The IMS Mozilla OpenBadges specification deals with micro-credentials which are smaller achievements and usually presented in a way that resembles a military or boy scout badge. As explained in previous chapters, badges have become more and more popular over the past ten years and play a large role in MOOCs and educational sectors within the industry. From the very beginning of this trend, IMS and Mozilla have been in the market with their OpenBadges specification, which is the reason why it is further explained in this section.

Listing 5.7: Assertion compliant to the OpenBadges specification. Example adapted from [42]

```
{
      "@context": "https://w3id.org/openbadges/v2",
      "id": "https://tum.de/assertions/241010",
      "type": "Assertion",
      "recipient": {
            "type": "email",
            "identity": "dominik@emailaddress.com",
            "hashed": false
      },
      "issuedOn": "2020-03-15T23:59:59+00:00",
      "verification": {
            "type": "hosted"
      },
      "badge": {
            "type": "BadgeClass",
            "id": "https://tum.de/badges/255",
            "name": "Blockchain BootCamp",
            "description": "This badge is awarded for participating in the
               Blockchain BootCamp",
            "image": "https://tum.de/badges/255/image",
            "criteria": {
                  "narrative": "Students learn the technical foundations
                     about Blockchain Networks."
            },
            "issuer": {
                  "id": "https://tum.de/issuer",
                  "type": "Profile",
                  "name": "Technical University of Munich",
                  "url": "https://tum.de",
                  "telephone": "+49089111222",
                  "email": "contact@tum.de",
                  "description": "TUM is one of Europe's leading technical
                     universities and strives for excellence.",
                  "publicKey": "SHA256:xJrFkhNs9pwibJFZZB5LvcrltWxfAIovk/
                     UjKAXkIW4",
                  "verification": {
                        "allowedOrigins": "tum.de"
                  }
            }
      },
```

```
37      "evidence": {
38            "id": "https://example.org/dominiks-blockchain-network.html",
39            "name": "The DomChain",
40            "description": "Link to store and retrieve data on Dominik's
                 blockchain system.",
41            "narrative": "Dominik implemented his own blockchain system
                 that can be accessed through a web browser.",
42            "genre": "Blockchain Systems",
43            "audience": "Developers, Recruiters, Researchers"
44      }
45 }
```

Listing 5.7 demonstrates an example for an *assertion*. In the OpenBadges terminology, badges are called assertions since they represent a claim that has been made and can be verified by relying parties [42]. The document format is, as with the other examples in the previous section, written in JSON. Similar to the W3C draft, the example starts with a `context` to reference a schema so that machines can better communicate. Here, the OpenBadges v2 schema is referenced (cf. line 2). Below, the `id` field contains the unique identifier for the assertion. In this specific example, the `id` contains an Hypertext Transfer Protocol Secure (HTTPS) address since the `verification.type` is "hosted" (see line 4). There are two types of assertions: `hosted` ones that are verified by retrieving the badge from the provided URI. And `signed` badges that are wrapped in a JSON Web Signature [42]. Whenever badges are issued as `SignedBadges`, the `verification` object should contain a `creator` flag storing the issuer's public key for better identification.
In line 4 of Listing 5.7, the `type` of the document tells the verifier that it is an assertion. Since types are related to the context, the specification allows to create and issue assertions of other types as well. However, to be machine-readable, the types should always be referenced in the context. This allows for extensions as well as using the specification for several different aspects, similar to the W3C VC draft.
After the type, the `recipient` is described. The `type` inside the recipient object specifies what identification method to expect. Currently, according to the specification, most systems use `email` to identify their recipients. Technically, however, the identification method could also be a DID or UUID. As with the `type` property in line 4, this had to to be declared in the provided context. For the identity flag in the recipient object, the specification also offers the possibility to hash the value. Therefore, a `hashed` flag can indicate if the information was hashed and a `salt` key can store the hash's salt if any was used.
The `badge` object contains the assertion that has been made about the recipient. Depending on the `context` again, the type `BadgeClass` could either be changed completely or enhanced with other types for extending the document. The `BadgeClass` is stated as go-to model for creating assertions in this form. Each badge contains an `id` that contains a link to the hosted badge. Most systems can only interpret HTTP addresses at the time of writing this thesis and therefore access badges via web queries [42]. The properties `name`, `description` and `image` describe what the earner has achieved and optionally provide an image for displaying the badge. The `criteria` object contains either a narrative describing what the requirements were to fulfill earning this badge. Or it contains an `id` that points to an HTTP address where the criteria is stored (e.g. a MOOC course description).
The issuer is also located inside the `badge` object. Several fields are describing the issuer such as `name`, `url`, `telephone` and `description`. Although an `id` flag is present, the issuer is identified by the `email` value in most systems [42]. Therefore, providing an email address, although not required by the specification, is mandatory when using certain compliant systems. Additionally, there are two mechanisms included for better identification: First, the `publicKey` value of the `verification.type` is `SignedBadge`. Secondly, the `verification` object only allows IDs that contain the issuer's domain. For instance: If an assertion's id is `https://myUni.de/assertions/241010` but the allowed origin states `tum.de`, the

assertion would be considered invalid.
Lastly, an already known element from the W3C VC draft can be found in the OpenBadges specification as well: evidences. Identical to the evidences mentioned in section 5.1, the object can be used to further prove that the assertion is valid. Inside the `evidence` object is an `id` that points to a website containing e.g. a deployed blockchain system (see line 38, Listing 5.7). Below that, the evidence is described in more detail by the properties `name`, `description`, `narrative` and `genre`. The `audience` property provides the opportunity to define for which segment of people this evidence could be interesting (here: developers, researchers and recruiters). Contrary to the W3C VC's evidences, no attachments can be made to the assertion. The evidence is only described and referenced for verifiers.
Complementing assertions, the OpenBadges specification introduces *endorsements* to create credibility among peers. Endorsements have an own `type` and can be used for several use cases. In Listing 5.8, the Technical University of Munich endorses the provided information about another university in Munich, the Ludwig-Maximilians-Universität München (LMU). Within the `claim` object, the LMU is referenced with its issuer id. To endorse the information below correctly, it has to be the same as stated in the claim object's `id` flag. Third-parties can then see the endorsement that is referenced e.g. on the issuer's page or even create services that do automatic checks between the issuer's email and the email address that has been endorsed by others. Furthermore, the concept of endorsement allows to generate a "like" system. Similar to Facebook posts, badge classes could be endorsed if they provide a good framework for certain achievements. Due to the extensibility of the specification, badge classes can exist that are tied to a certain type of achievement which might be reused by other issuers. Therefore, endorsements provide a way to state that a certain class is better than others [42].

As the term *assertion* already states, verification and validation has to be conducted to prove the information contained in a badge. Depending on the verification type (hosted or signed), there are different verification methods:

General tests for both types
The first verification step is to validate all included date. This involves checking if each mandatory field contains a value and if the overall structure is a valid JSON document. Furthermore, every linked address is queried to see if the data is available by receiving an HTTP 200 code ("ok"). Afterwards, the verification starts by checking if all badge objects were created by the referenced issuer. Following, the recipient id is verified. Since an email address is mostly used for recipient identification, the issuer might store the email addresses in a separate registry declaring them as "known addresses" [42]. In the next step, the "assertion issuer" [42] is checked if she is authorized to issue the claims made in the badge class. This is typically true if the same issuer is mentioned in the badge class. Lastly, the expiration date is checked and revocation status is queried to see if both are still valid.

Listing 5.8: TUM endorses the issuer information about LMU. Example adapted from [42]

```
{
      "@context": "https://w3id.org/openbadges/v2",
      "type": "Endorsement",
      "id": "https://tum.de/endorsements/lmu-end2020.json",
      "issuer": "https://tum.de/issuer",
      "issuedOn": "2020-01-01T23:59:59Z",
      "claim": {
            "id": "https://lmu.de/issuer",
            "email": "lmu@emailaddress.com",
            "name": "Ludwig-Maximilans-Universitaet Muenchen",
            "phone": "+4989123123123"
      },
      "verification": {
            "type": "hosted"
      }
}
```

Verification of hosted assertions

Hosted badges are stored either on the issuer's server or on servers that contain several assertions from different issuers. In the former case, the check between the issuer's domain (e.g. tum.de) and the `allowedOrigin` is rather simple. In the latter case, however, the `verification` object should include a `startsWith` field that specifies the root domain of the host where the assertions are stored. Additionally to the URL checks, the above described general tests can be conducted by the verifier as well.

Verification cannot be separated from revocation. If an issuer decides to revoke an assertion due to any reason, the hosted badge mechanism provides an easy way for that. As explained before, verifiers have to retrieve the assertion via a HTTP GET request. Each HTTP request returns a response code, commonly 200 for "ok" or 404 for "not found". The OpenBadges specification uses the 410 "gone" code for revoking hosted badges. Once an assertion has been revoked, the 410 code will be returned along with a `revocationReason` inside the response body.

Verification of signed badges

Different from hosted badges, signed ones are wrapped into a JSON Web Signature (JWS). Upon issuing, the badge is signed by the issuer's private key. To check if the signature is matching with the issuer's public key, it has to be stated in the `publicKey` field inside the `issuer` object.

At the beginning of the verification process, the received assertion is decoded using the Base64 algorithm which results in a string. The string has then to be parsed into a JSON

document. If this fails, the assertion will be declared invalid. Afterwards, data validation has to be performed similar to the hosted badge. Again, if any mandatory field are missing, the assertion has to be declared invalid.
Now, the `publicKey` flag inside the `issuer` field comes into play. The signature has to be checked against the contained public key id of the issuer. If either this check fails or the issuer has not provided a key, the assertion should not be trusted, according to the specification [42]. If a key is contained, a HTTP GET should be executed to see if the key can be retrieved. This is another step to ensure credibility and trust in the Public Key Infrastructure (PKI) system. With the public key validated, a JWS verification has to be performed. Lastly, a new object comes into play that is only required for signed badges: the revocation list. Each issuer has to maintain a revocation list that contains at least the ids of revoked assertions. Verifiers have to retrieve this list (also a JSON document) and check if the to be verified assertion is on this list. If so, the assertion has been invalidated by the issuer. Additionally to the id, the issuer is able to provide a reason for revocation similar to the W3C VC draft's revocation mechanism [33] [42]. After having passed all checks, the assertion may be treated as valid.

5.4 Establishing Identification Between Virtual and Analogue Entities

Following the reference technologies explained in the three preceding sections, this one deals with two major identification approaches. The first one deals with governmental, therefore centralized identification using personal identification cards. The second approach is the often mentioned decentralized identification (DID) method. Both approaches have an extensive technological and conceptual background that cannot be addressed entirely in this section. Rather, the overall functionality and usage within the domain of digital credentialing is explained.

5.4.1 Introduction to eIDAS

In Chapter 1, the *electronic identification and trust services for electronic transactions in the internal market* regulation has been broached. As with all EU regulations, each member has to implement it in their own way and guarantee that the regulation has been fulfilled. Therefore, the eIDAS regulation does not state anything about technical implementation, it rather defines requirements that have to be met by the member countries. In Germany, the Buundesamt für Sicherheit in der Informationstechnik (BSI) is in charge for the technical implementation of the eIDAS guideline.
In the analogue world, showing the card is enough to identify as a citizen and prove that personal data is correct. The verifying party will see if the data submitted matches the data stated in the id card. Additional information is provided by querying connected systems such as crime databases. However, the main part is comparing the data on a sheet with the one on the card. In the end, the person either passes or fails the check. Contrary to e.g.

a *login*, the passport check does not involve a token so that a person could walk in and out of an airport security zone for a limited time. The check has to be re-done every time the person wants to enter the security zone. That paradigm has been used as a template for creating the eIDAS compliant online authentication system in Germany: Showing a relying party the passport but not leaving a "(...) permanent proof of identification (...)" [43].
At the beginning of the online authentication process, each citizen has to have an identification card that contains a chip inside. This chip stores most of the data including all the relevant data such as name, date of birth, address expiry date, but also the photograph and information such as stage name and indicators if the holder has reached a certain age or the place of residence matches the place stated [43]. Furthermore, a unique revocation token is stored on the card. It is used to check in a revocation list whether the card has been invalidated. If so, the authentication check will fail immediately.
With the personal identification card comes a personal identification number (PIN). Both elements form a two-factor authentication method by holding the card and knowing the PIN. In order to use the governmental identification for authentication to online services, the card, the PIN and a card reader device are necessary. The card itself contains a chip that can be read by either a certified terminal or an app developed by the BSI only.
Once the holder has these three components at hand, she is able to authenticate himself to web services. The authentication process is designed similar to the paradigm explained above: A person would unlikely show her personal data to an entity that she does not trust. Therefore, the authentication method implemented is called *mutual* trust. As the name already states, both parties have to be trusted to succeed the authentication process. On one the hand, the holder is trusted by holding the card and knowing the PIN. On the other hand, the verifier has to be trusted as well. Therefore, each verifier who wants to use authentication compliant to eIDAS has to receive a certificate issued by the Issuing Office for Authorisation Certificates [43]. Once a verifier has received the certificate, she is authorized to establish a connection to the holder's personal identification card chip.

In Figure 5.3, the infrastructure involved in the authentication processes is depicted. Beginning on the left side, the user "(...) requests a web service that requires authentication" ~[43]. The service provider receives the request and passes it to the eID-server that "(...)" activates the eID client via the user's application" [43]. The eID client and the eID server then communicate with each other. The user is able to see the data the service provider wants to retrieve and can select or deselect the parts she want to share. After that, the user gives her consent to authenticate himself by entering her PIN. Once entered, the *General Authentication Procedure* takes place involving a check if the session certificate is matching the verifier's authorization certificate. Upon successful verification, the data is securely transmitted from the chip on the passport to the eID server. The eID server then sends a response to the service provider containing the requested data (or the portions that have been shared by the user). The eID client on the user's side redirects to the web site and the service provider decides whether to grant the user access to the service or not. The authentication process does not involve any third-party between the chip on the id card

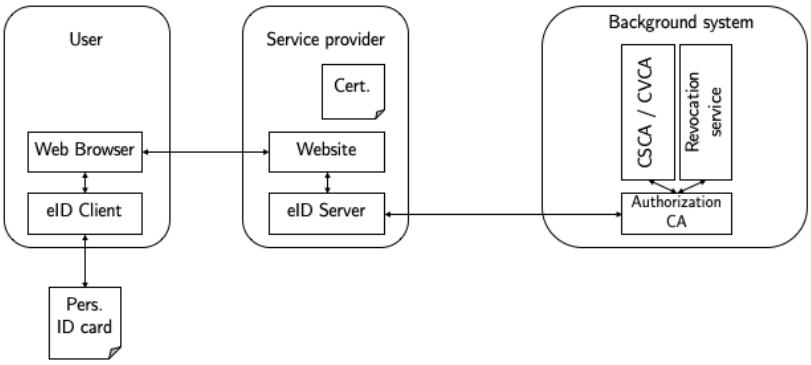


Figure 5.3: eIDAS infrastructure involved in the authentication process [43]

and the eID server on the verifier's side. Both parties can establish a direct and secured connection while performing the authentication process. Even the background system is not involved in this process. The service provider has to connect to the background system regularly to retrieve new invocation lists and refresh the authorization certificate. Apart from that, there is no further involvement in the process.
Inside the background system bracket in Figure 5.3 there are three components that have not been mentioned yet. The *Authorization CA*, or *Authorization Certificate Authority* (ACA) in full length, is responsible for creating, verifying and distributing the certificates service provider use in the authentication process. Since certificate validity is limited to one day, the authority has to contact each service provider on a very frequent basis and re-validate or re-distribute the certificate. Above the ACA, there are two components: the *Country Signing Certification Authority* (CSCA), *Country Verifying Certification Authority* (CVCA) along with the *Revocation Service*. According to [44], the CSCA forms a single point of trust nationally and certifies institutions responsible for creating electronic identification objects (e.g. ePassport or eID). These institutions are then called *Document Signer* (DS).
The CVCA is the highest institution in a country to generate root certificates for retrieving information from electronic identities. Hierarchically below the CVCA are the Document Verifier (DV). A document verifier has the permission to retrieve the information and can be in form of a control authority (e.g. the police or border patrol). [45]
Overall, the eIDAS authentication process forms a strongly secured but inflexible way to create a connection between virtual and analogue identities. It is compliant to privacy and anti-tracking regulations such as the GDPR and creates a bond of trust by using certificates on both the user and the relying party side. The downside of this algorithm is that there is no identifier creation. In the above mentioned specifications, most parties are

trusted by stating either a unique identifier or a public-key id. Furthermore, the electronic passport has no way to store data as it is in Estonia [46]. Here, citizens are able to store their medical credentials on their personal identification card. This enhances security and portability, but also makes the citizen transparent to central authorities.

5.4.2 Introduction to Decentralized Identifiers

Contrary to the centralized authentication and identification methods stated in the previous section, this one deals with a *decentralized* method. DIDs are still in development by the W3C. Similar to the Verifiable Credentials Draft [33], the W3C aims at bringing identification to the control of each individual user and distributing trust among a peer model. In the specification, several "primary design goals" [47] are stated: control, privacy, security, proof-based interoperability and more. Different from e.g. the eIDAS method, DIDs serve the purpose of being cryptographically provable instead of trusting a centralized authority. In this section, a brief overview of how decentralized identification is designed to be working.

Listing 5.9: DID Example taken from [47].

```
did:example:123456789abcdefghi
```

Listing 5.10: Corresponding DID Document to Listing 5.9 from [47].

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123456789abcdefghi",
  "authentication": [{

    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "service": [{

    "id":"did:example:123456789abcdefghi#vcs",
    "type": "VerifiableCredentialService",
    "serviceEndpoint": "https://example.com/vc/"
  }]
}
```

Each DID consists of two parts. The identifier, listed in 5.9 and the DID document it refers to (see Listing 5.10). The first and smaller part of the DID data model is the identifier. Syntactic, it is comprised of three parts that can bee seen in Figure 5.4: The first part is

the scheme which has to be set to `did` since it indicates that the system is dealing with a DID. Secondly, the method is specified. By design, the DIDs are based on distributed ledger technology. Depending on the underlying technology, the methods define the generation, readability, creation and deletion of DIDs [48]. Furthermore, the specifier behind the method is influenced as well. Applied to the Ethereum blockchain, a DID would look like the one in Figure 5.4 shows. Here, the method name is `ETHR` and the method-specific identifier references an address that is bound to a wallet participating in the Ethereum network. Similarly, Bitcoin address have `btcr` as method name and contain the transaction and block address of the sender [48].

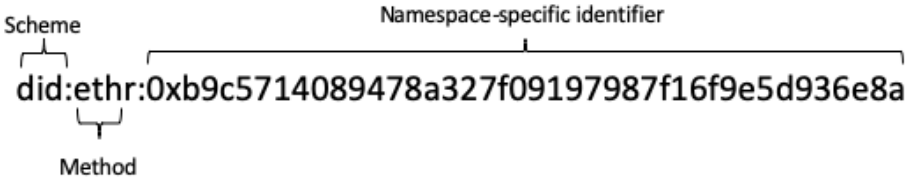


Figure 5.4: DID with Ethereum method and namespace-specific identifier. Adopted from [48].

Independent from the method and method-specific identifier, each DID has to reference a document that can be seen in Listing 5.10. The DID document resembles the verifiable credential data model stated in previous sections. Again, the `context` element creates a machine-readable environment by informing participating peers what to expect in the following and define the syntax of the document. The `id` states the DID subject. The subject, similar to the VC draft, is the entity that is identified by this document. In both specifications, the DID is used for identification purposes. In the following, an `authentication` object is included that contains several elements. The first one is the `id` which references the public key below that. The `type` states the verification method, e.g. Rivest–Shamir–Adleman Cryptosystem (RSA). The `controller` is "the entity, or a group of entities, in control of a DID or DID document" [47]. Here, the controller is the same as the subject stated in the first `id` element in line 3, Listing 5.10. Afterwards, the public key is denoted. In this case, it is stored as a *Privacy-Enhanced Mail* (PEM) property according to RFC-1421. Different properties such as `publicKeyJwk` or `ethereumAddress` are also allowed. The list of fitting properties here is "non-exhaustive" [47].
The `service` object is located below the `authentication`. Service endpoints are a way to allow further discovery of the identification. The DID subject may specify several service endpoints similar to the one stated in the example. In Listing 5.10, the service endpoint establishes a connection between the DID document and a verifiable credential that can be retrieved from the given `serviceEndpoint`. Apart from credentials, authentication services could be connected that contain a permission model or a message service that provides a way to communicate with the DID subject. Not mentioned in the example but relevant when it comes to authenticity of the document is the `proof` object. It can be contained to make the DID document cryptographically verifiable, but it does not create a proven

link between the DID and the document. However, the proof here is optional since the underlying technology has to be a distributed ledger which itself provides cryptographical verification.

Different from the eIDAS authentication in subsection 5.4.1, the DID document itself cannot create a binding between an entity in the virtual realm and its part in the real world. However, there are two ways to verify the authenticity of the document. The first one is indicating who is in control of the DID document. Signing the document alone with the controller's private key is not enough to say that the private key owner is the controller of the document. The DID has to be resolved so that it points to the DID document according to the specified method. Furthermore, the id of the document has to resolve the DID from the beginning of this process. Only if both ways show a match, the controller can be assumed as correct [47].

At this point, the DID and the DID document can be considered to be under the control of the provided public key inside the `authentication` object. In a next step, the public key has to be verified. Here, signing the document with the controller's private key would be one way to indicate that the public key inside the authentication object is valid. The other way would be sending a `challenge`. A challenge consists of a " (...)public key description from the DID document and a nonce (...)" that is sent to a service endpoint contained in the document [47]. Afterwards, the signature of the response would be verified against the public key description [47].

5.4.3 Creating a Link Between DID and eIDAS

In the previous subsections, both the centralized and decentralized methods have been explained. However, no direct linkage between the two methods has been stated. Derived from both methods, an example case is created in this subsection that explains how both methods could be used to create a virtual identity based on a state-issued identity.
As already explained, the eIDAS technology was designed to mimic the "Passports please" situation at the airport. It does not create a token that can be re-used. Rather, it shows once that a user is the one she claims to be. Contrary, DIDs provide a way to create a static but extensible identity in the virtual world that has no real connection to the analogue world. For that reason, public keys that are generated once. They are also unique for each generation. Each DID should have at least one unique public key attached to it. Therefore, the generation of either a wallet (acting as Ethereum address) or a private-public-key pair could involve the eIDAS process. eIDAS could be used as a gateway to create keys.

6 Analysis of the State of the practice

In this chapter, the state of the practice of digital credentialing is presented. More precisely, it shows the application of the framework defined in Chapter 4 to systems developed by companies (practitioners) and by researchers. Since being a researcher and contributing to a company does not exclude itself, there are some cases where both domains overlap. As an example, the company Sproof (see Subsection 6.1.16) has roots in the research environment. Furthermore, each subsection is comprised of two parts: the framework with the gathered data and a description about either the company or the research project. Since the information has been aggregated using public information, some field my be flagged as *N/A*, meaning *not available*. In these cases, information was not provided by the company or research project.

6.1 Applying the Framework to Practitioners

In this section, companies are investigated who are actively participating and competing in the market for digital credentialing systems. Most of the companies were either found as mentions in research publications or using public search engines. The domain of digital credentialing is vast and this chapter can only represent some of the existing companies. Furthermore, the main requirement for a company to be mentioned in this chapter is to approach macro-credentials. Since digital badges have been around for approximately ten years, this chapter investigates the relatively new domain of macro-credentialing.

6.1.1 Accredible

Accredible offers a web-based platform that allows both issuing and receiving badges, certificates and digital credentials. With an available API, integration into other systems is possible but limited to services. Accredible cooperates with, e.g. Brightside or Canvas. For universities that issue more than 10,000 credentials per year, a special contract will be negotiated that might include customized endpoints to the internal university system. However, at first sight this seems to be not the case for minor systems.
Accredible claims to be compliant with the OpenBadges specification but not with the W3C VC draft. Although several aspects are included in the data model that resemble the VC draft (such as the evidence list), major aspects have been left out. There is no concept of cryptographic proof available, nor any identification method for issuer or holder apart from email addresses. In a small test, the author was able to register as an issuer under the name of *TUMTEST* and use the URL "https://tum.de" for identification. It was not necessary to identify as a university representative or state any membership at all. This

impacts the peer-to-peer trust model that is implied by the platform. Since receivers and issuers are identified by either an `id` or their `email` address, peers would trust the author that she is a representative of the TUM although she is not. Here, real-world identification or at least certified identifiers should be in place to further improve the trust model.
Taking a look at the framework data presented in Table 6.1 underlines the non-conformance with the VC draft. From the information available during the time of writing this thesis, three requirements have not been met: asserting and moving claims as well as revocation. Although updating the entire credential data is possible for the issuer, there is no option to revoke the certificate once issued. Furthermore, the data is kept inside the Accredible system. When an issuer decides to cancel the contract, Accredible promises to keep certificates and credentials alive as long as they do not expire. However, there is no option provided to move credentials from the Accredible system to another one.
Regarding the processes, everything takes place inside the Accredible platform. Issuers are presented a dashboard where they can enter *courses* and assign recipients on course lists which are then attached to a credential. The credential can subsequently be batch-issued using a Comma-separated Value (CSV) and issuers receive an email with a link to the document. Once clicked, the receiver is invited to register on the Accredible platform to manage her credentials, e.g. setting the visibility to private. Verification takes also place inside the Accredible suite. From the information available, there are two stages of verification: Stage one indicates if the credential was issued by a registered issuer. Stage two tells verifiers if the credentials has been tampered or not. The second stage is only available if the issuer selected the blockchain option during the publishing process. Otherwise, only the stage one verification would take place. As already stated, there is no cryptographic proof attached to the data model nor any indication of PKI. Therefore, the verification quality is rather minor compared to the mechanism proposed in the W3C VC draft.
Overall, the platform looks mature and the business model indicates that the venture has reached a certain maturity level. Since there is no insight to the company available, the maturity level can only be assumed which will be the case for all the following companies as well. Here, the assumption is that the maturity level is *Quantitatively Managed* due the extend of how the service is working and how many customers the platform claims to have. Apart from the mentioned customers, no usage data is available. Table 6.1 shows the framework data for APPII [49].

Accredible		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Issuing via dashboard platform.
	Store / Move Claim	Credentials are stored in a blockchain that also serves as a verification tool.
	Refresh	The API offers an UPDATE call that allows issuers to update all data except the ID.
	Revoke	N/A
	Receive	Holders receive their credentials via email. A link leads to the platform where they can see the document and register for a wallet.
	Assemble	N/A
	Interact	Holders can share their credentials via regular Facebook, LinkedIn etc. sharing-services, unless the certificate is set to private.
	Verify	Verification can only be done if the credential is also issued to the blockchain. Which blockchain that is has not been stated.
	Surroundings	Every user can issue, verify and hold certificates using this system. Furthermore, the data model can accommodate custom fields that are represented in the credential.
System	Data model	JSON with PDF display possible.
	Permission	Password and username upon registration. Receiving and verification don't require permission.
	Data Storage Model	Blockchain
	References	OpenBadges conform, several aspects adopted from W3C VC Draft.
	Macro- / Micro-Credential Compatibility	Macro + Micro
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	Yes
	Identification Method	Username and Password. No authentication necessary for issuers.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Subscription packages based on recipients per year available.
	Usage KPIs	N/A
	Cooperations / Partners	Various customers including HootSuite, docker, Rosetta Stone, IT College Buenos Aires
	Maturity	Quantitatively Managed
	Target Industry	Educational Sector

Table 6.1: Framework data for Accredible.

6.1.2 APPII

APPII is a web- and mobile-based platform for creating verified Curriculum Vitae (CV). Different from issuing macro-credentials such as diplomas, institutions and organizations can verify a holder's entry in the digital CV. Therefore, an issuer has to register and undergo a manual check to be a registered and verified entity in the system. Each holder has to identify via biometrical checks during the registration process so that online and offline identity are matching. Once each entity is verified in the system, holders can issue claims about themselves that are compared either with claims an institution can make or simply verified by these entities. Upon successful assertion, the entry in the holder's CV is flagged as verified.

At its core, the platform serves the purpose of creating verified profiles of people by combining holders, companies and institutions. To generate a revenue from this, employers can join the platform as well and get data insights or direct access to a pool of people that is recommended by the system. To further improve this, interaction mechanisms such as sharing and communication is implemented as well.

Each claim in the CV is stored on the blockchain to prevent tampering and allow third-party verification apart from the verifiers that originally proved the claim. Consequently, a peer-to-peer trust model is created where verifiers rely on the decentralized and architectural nature of the blockchain to trust in the information provided by the holder.

As of writing this thesis, 744 holders are registered on the platform along with around 16,000 organizations. However, the quality of data in terms of how many verifications have been done could not be determined. Furthermore, information about revocation or moving claims from the APPII platform to another is not provided. Based on the usage KPIs and missing partners, the maturity is assumed to be *initial*. Table 6.2 shows the framework data for APPII [50].

		APPII
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	No
	Revoke Claim	No
Actions	Issue	Issuing via dashboard platform.
	Store / Move Claim	Credentials are stored on a blockchain. Moving claims is not possible since they are bound to the system.
	Refresh	The credential itself can be updated by the holder, but there is not verification whether the issuer can refresh the credential as well.
	Revoke	N/A
	Receive	Holders don't receive their credentials. They create them themselves and request verification from partners.
	Assemble	The core feature of APPII is to create a verified CV that itself is an assembly of various (un-) verified claims.
	Interact	Holders can share their assembled CVs via regular Facebook, linkedIn etc. sharing-services. There is no limitation regarding privacy or time.
	Verify	Partners verify the credentials via Blockchain mechanics.
	Surroundings	APPII defines specific roles for holders and verifiers. A verifier has to be an organization and cannot be another holder. Furthermore, the system is strictly designed to fulfill the CV use case.
System	Data model	Web page that can be exported into a PDF file.
	Permission	Password and username upon registration. Receiving and verification don't require permission.
	Data Storage Model	Blockchain
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro + Micro
	GDPR Compliance	No
	API Available	No
	Meta Data	Yes
	Identification Method	Biometrical identification required for holders. Verifiers have to register in custom process.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Verifiers gain rewards by verifying claims. Additionally, employers can access specific features for recruitment.
	Usage KPIs	744 registered holders (called "people") and 16,848 organizations in the registry. Data quality cannot be determined.
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Human Resources Sector / Employers

Table 6.2: Framework data for APPII.

6.1.3 BCDiploma

BCDiploma's goal is to enter the educational sector by serving an application that automates and "dematerializes" the issuance of macro-credentials [51]. Therefore, the company offers a web-based platform that allows issuing on certificates on the Ethereum blockchain. Using the blockchain allows the company to offer verification to third-parties in a similar fashion as already described previously: Employers, for instance, can check if the certificate is valid and if it has been tampered by accessing the received link to the credential on the blockchain. Furthermore, the software allows to store meta-data such as grades, personal information and what requirements had to be met to achieve the credential. Additionally, the BCDiploma offers a service to customize the appearance of the document so that institutions can design them according to their corporate identity.
In terms of privacy, each diploma is encrypted with three keys: One for the institution, one for the network and the last one for the holder. This ensures that only these entities can read the diploma.
BCDiploma is based on a framework called *EvidenZ*. EvidenZ provides the logic for registering and certifying institutions, but is also used for the issuing process. In a decentralized manner, institutions create a transaction containing their identification to the Ethereum blockchain, which is certified by so called *validators*. The institutions' identification transaction is stored in a smart contract that communicates with several other smart contracts serving as validators. Each validator checks the identity and address of the institution and sends back a transaction containing the validation (if so) to the smart contract. In the end, each institution has been checked and approved by numerous smart contracts to assure integrity and trust in the system. The framework is open-source, whereas the BCDiploma software serves as the business model [52].
Currently, over 80 institutions are registered in the issuers list and 927 transactions are visible in the EvidenZ Ethereum smart contract. Prices and business model are not described on the website, neither are any partners stated apart from the Microsoft for Startups partnership. The overall maturity is therefore assumed to be *managed* [51]. Table 6.3 summarizes the described data.

BCDiploma		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Institutions issue credentials through an app that deploys the credential onto the Ethereum blockchain in a smart contract.
	Store / Move Claim	Once issued, the credential is stored on the Ethereum chain.
	Refresh	N/A
	Revoke	Data can be made indecipherable by deleting the associated persistence key.
	Receive	Holders receive a link to the credential stored on the chain.
	Assemble	N/A
	Interact	Holders can share the URL to thrid-parties which are able to use a web-reader to see the credentials
	Verify	Verification is done via smart contracts on the Ethereum Blockchain. Furthermore, the issuer is certified by validators.
	Surroundings	Holders can be verifiers and issuers as well. However, validation has to be conducted to serve as issuers.
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Ethereum Blockchain
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	yes
	API Available	Yes (in the future)
	Meta Data	Grades and requirements along with personal information.
	Identification Method	Organizations are validated through certificates by "validators". Validation includes checking registries, physical address and banking information.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Pay per issuance in tokens
	Usage KPIs	80 registered institutions and 927 transactions in the EvidenZ ETH Smart contract
	Cooperations / Partners	Backed by Microsoft for Startups
	Maturity	Managed
	Target Industry	Educational Sector

Table 6.3: Framework data for BCDiploma.

6.1.4 BlockCo

BlockCo is a product "powered by" the University of Nicosia [53].The University of Nicosia is one of the leading institutions for issuing macro-credentials completely digital. BlockCo relies on the Bitcoin network and stores the data in Bitcoin transactions. Since the amount of data is restricted to 80 Bytes per transaction on the Bitcoin network, PDF files cannot be simply stored in a transaction. During the issuing process, the PDF document is hashed and the resulting fingerprint is included in a transaction. Once mined into the blockchain, the transaction details are added back to the document's metadata before shipping them to the holders. Holders can then send the credential to verifiers such as employers. To verify a document, third-parties can use either the BlockCo website or partnering websites. There, the document is uploaded and immediately verified. In a similar fashion, revocation takes place: Once an issuer wants to revoke a credential, another transaction containing the metadata of the credential is issued. Validators will see this transaction stored in a Merkle tree and render the certificate invalid during the check.
Similar to the OpenCerts framework, BlockCo uses DNS identification for asserting an institution's identity. Holders do not have to undergo these checks since they receive their credentials independent from the software. Documents are deployed on the blockchain and then sent to the holder using standard formats such as email or internal services.
Apart form the education sector, the BlockCo aims to serve industries such as suppliers, governmental sectors, professional trainings and insurances. The software is open-source and deployed at the University of Nicosia as well as the British University in Dubai. Clients are not listed on the website. The maturity is rated *managed* and the framework data can be found in Table 6.4 [53].

		BlockCo
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	PDF files is optionally attached with meta data. Fingerprint of the document is included in a BTC TX.
	Store / Move Claim	Data is stored as PDF files that are distributed to holders.
	Refresh	N/A
	Revoke	Another BTC transaction is issued that invalidates the previous record.
	Receive	Holders receive PDF once the BTC TX is complete and the TX data is attached to the document.
	Assemble	N/A
	Interact	Holders can share the PDF file independent from any system.
	Verify	Verification is done by uploading the PDF file at certain validators. The attached meta data contains the BTC TX which is checked against the network. If valid, the validator returns true.
	Surroundings	Everyone can be holder and issuer. Verifier have to be a validator as well to serve this role. This is tied to issuers, which complicates the process.
System	Data model	PDF file
	Permission	Permissionless
	Data Storage Model	Bitcoin Blockchain
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	Attached to PDF through BTC TX
	Identification Method	Issuers are identified by owning the domain and manual validation through the company.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	University of Nicosia: 87 TX
	Cooperations / Partners	University of Nicosia, British University in Dubai
	Maturity	Managed
	Target Industry	Educational sector

Table 6.4: Framework data for BlockCo.

6.1.5 BlockCerts

BlockCerts was initially designed and prototyped by the Massachusetts Institute for Technology (MIT) Media Lab and has been transformed into an open-source project. Currently, the ongoing development is separated form the MIT into a community project with its code base on GitHub.
OpenCerts aligns with several standards, most importantly for this thesis with the OpenBadges and W3C VC Draft. This determines both the data model and features that the platform offers. Credentials are issued in JSON format and stored on the Bitcoin blockchain. During the issuing process, institutions send invitation links to receive a blockchain credential. The holder accepts the invitation by sending her blockchain address. Afterwards, the issuer hashed the credential onto the blockchain and sends the credential to the recipient who gets it in her wallet (a mobile app). The holder is then able to pass the credential to a verifier who queries the blockchain to verify it. Although BlockCerts uses the Bitcoin network to store credentials, the GitHub repository states that Ethereum can also be used to anchor the data there.
BlockCerts explicitly excludes identification processes from their platform. The reason behind this is that basically everyone can obtain a valid Bitcoin address and peers have to simply trust this address. Rather, BlockCerts endorses to build "curated profiles" around Bitcoin address to gain credibility [54]. Another option would be to use claims as proposed in the DID specification described in chapter 5. Each institution could be endorsed by claims verified by peers.
Apart from a first deployment at the MIT and issuing a small amount of credentials there, the BlockCerts website does not provide information about usage. Furthermore, no partners have been listed. The project itself is open-source and therefore open for collaboration, making it a joint effort to build the credentialing platform. The community has comprised a road-map stating to expand to e.g. the Ethereum blockchain and refine the revocation system. In the future, decentralized identifiers should be included as well. Based on this, the maturity is assumed to be *managed*. Table 6.5 shows the framework data gathered about this project [54] [55].

BlockCerts		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	Yes
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Issuers hash credentials onto the blockchain and send the credential to holders.
	Store / Move Claim	Credentials are stored on the blockchain but can be held in a wallet by the holder.
	Refresh	Refreshing is not possible. Only re-issuing.
	Revoke	Same mechanism as issuer-hosted badges in OpenBadges specification.
	Receive	Holders have to register their blockchain address at the issuer. Afterwards, they receive the credential through a transaction.
	Assemble	Assembly can be done inside the wallet by gathering certain credentials into a verifiable presentation.
	Interact	Holders can send their verifiable presentation through an app to verifiers.
	Verify	Verifiers can use either an open-source system provided by BlockCerts or use the verifier on their website. The credential is checked against the blockchain and verified if an entry was found.
	Surroundings	Data model is extendable. Everyone can be verifier, issuer and holder at the same time.
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Blockchain (Bitcoin and Ethereum)
	References	W3C VC and OpenBadges compliance
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	Yes
	Identification Method	Claims-oriented identification method as proposed in the W3C DID draft.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Open-source
	Usage KPIs	N/A
	Cooperations / Partners	Massachusetts Institute of Technology Media Lab
	Maturity	Managed
	Target Industry	Educational sector

Table 6.5: Framework data for BlockCerts.

6.1.6 Blockeducate

Blockeducate is a similar product as Accredible and BlockCo. The key business model behind this digital credentialing platform is to enable institutions and companies to create customized credentials that are stored on the blockchain. Therefore, the company utilizes the Ethereum blockchain and several DApps that connect with it. Blockeducate offers both macro- and micro-credentials. Both types are issued using the platform and encrypted via smart contracts. Once issued, the smart contract pushes the credential to the holder's wallet who is then able to distribute it to verifiers. Third-parties receive the credential via standardized sharing mechanisms such as social media or email. Once received, the credential offers two ways to be verified: either via QR code or clicking a link. Both query the blockchain and state if the document is valid.
Institutions and enterprises can use the API to further automate and integrate issuing certificates. Both have the opportunity to create a customized design for their credentials. However, some of these additional features have to be unlocked by subscribing to one of Blockeducate's service tiers.
Blockeducate is part of the Vottun network. Similar to Acclaim (described Subsection 6.1.18) or BCDiploma and EvidenZ, Vottun offers frameworks for creating blockchain applications. Blockeducate is one manifestation of how companies generate a business model from the underlying Vottun technology.
Blockeducate is compliant to the OpenBadges specification but not to the W3C VC draft. Actions such as refreshing, assembly of credentials or revocation are not stated. Based on the business model and Vottun as the technological partner, the maturity is assumed to be quantitatively managed. Table 6.6 shows the framework data for Blockeducate [56].

Blockeducate		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Credentials are issued to the Ethereum blockchain using a web-based user interface. Once the credential is uploaded, it is sent to the holder's wallet.
	Store / Move Claim	The credential's hash is stored on the Ethereum blockchain, the credential itself is stored in the holder's wallet.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive their credentials in their wallet. The credential is transferred through the Blockeducate system.
	Assemble	N/A
	Interact	Holders can share their credentials with verifiers through the wallet. Furthermore, social media sharing is possible via URLs.
	Verify	Verification is based on blockchain mechanisms such as checking the hash and the issuer address.
	Surroundings	Holders, issuers and verifiers are not tied to their roles. However, issuers have to register and validate via third-part processes.
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Ethereum Blockchain
	References	OpenBadges compliance
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	N/A
	Identification Method	Issuers have to identify themselves in a process that involves third-party identification mechanisms. Afterwards, issuers are certified.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Features are behind certain Tier-packages. There is a free trial, a self-serve and an all-inclusive one available.
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Quantitatively managed
	Target Industry	Schools, Universities and similar insinuations

Table 6.6: Framework data for Blockeducate.

6.1.7 CHESICC

The Chinese Higher Education Student Information and Career Center (CHESICC) has developed an online verification platform for credentials. Backed by the Chinese Ministry of Education, CHESICC serves as a central authority for credential verification. Consequently, the system works completely different from the previous demonstrated ones. The first difference is that CHESICC has a database where each institution and each student is registered. The database contains information such as certificates, enrollment status, student photos and test results. According to their website, the database contains over a billion registered records [57]. The second difference is derived from the centralization aspect: Since each institution is connected to the database, the issuing process is streamlined for every institution. More specifically: every issuance is registered in this database. Therefore, verification is a rather simple process. Each credential contains a verification code that has to be entered on the CHESICC website. Once entered, the tool queries the database for the code and shows a report of the credential.
Holders have to register and identify on the platform in order to download the credential their. Each credential can be received in a print, PDF or HTML format. After reception, the credential can be distributed by holders to verifiers by email or social media. Apart from educational records, the system can also be used for immigration and visa issuing. Table 6.7 shows the framework data for the CHESICC platform [57].

CHESICC		
Category	Key	Value
Requirements	Issue Claim	No
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	The system itself does not offer an issuing service. However, the database is nationally deployed, whereas the verification tool has access to these credentials.
	Store / Move Claim	Storage is provided through the connected database. Moving is not possible.
	Refresh	The system offers a refreshment for expired credentials. The process itself is not described.
	Revoke	N/A
	Receive	Holders receive their credentials from the CHSI.
	Assemble	N/A
	Interact	Credentials can be shared via email since they are either in PDF or HTML format.
	Verify	The QR code can be scanned either via an app or the code behind it can be entered on the website. A database call checks whether the credential is valid or not.
	Surroundings	N/A
System	Data model	Print and online Report (HTML format) that can be exported to PDF. Contains a QR Code for verification.
	Permission	Permissioned. Central Authority grants access.
	Data Storage Model	Centralized Database (Chinese Higher Education Student Information)
	References	N/A
	Macro- / Micro-Credential	Macro-credential
	Compatibility	
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	Holders have to register with real name and citizen ID
	Trust Model	Central Authority
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	Chinese Ministry of Education
	Maturity	Optimizing
	Target Industry	Chinese educational sector and employers

Table 6.7: Framework data for CHESICC.

6.1.8 Credly

Credly's business model focuses on professional education. Deriving from this focus, the platform serves micro- rather than macro-credentials. Their customers list is comprised of large companies such as IBM, Dell, Oracle and Pearson, which indicates that the company has reached a high level maturity. Furthermore, it is compliant to the OpenBadges specification and implements Open Authenticate Version 2 (OAth 2) for identiyfing participants.
Customers connect their Learning Management System (LMS) to the Credly platform by using either an API or direct integration, in example the Moodle integration. Since the platform is compliant to OBI, the underlying data model is extendable and in JSON format. Requirements such as issuing claims, verification, retrieval and revocation are met by the company. Using the API, companies can issue badges by posting the data to the API. In return, Credly sends a 201 "created" response back and triggers a notification to holders who are able to receive the badge. Credentials are stored and hosted in a central database for each customer, which enables revocation for hosted badges as described in Chapter 5.3. If an issuer decides to invalidate a credential, an API call is sent containing the badge id as well as a revocation reason. In return, the API sends a 410 "gone" code back to the issuer and each verifier who subsequently queries this credential. In a similar way, verification is done via an API call: Verifiers query the badge URL and receive either a 200 "ok" code containing the information about the credential in the response, or they receive a 410 "gone" response upon invalidity.
Credly uses the Acclaim framework as underlying technology and builds a user interface as well as business logic on top. Acclaim handles verification, creation and distribution of data as well as authentication using an OAuth 2 interface. On top, Credly offers the platform that can be integrated into company processes. Furthermore, data can be analyzed and used for marketing purposes to extend the company's reach. Table 6.8 shows the framework data for Credly [58].

		Credly
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	An external platform can issue a badge by posting the data to the API via http and receiving a 201 "created" response. Then, the earner is notified via email that a new credential is available.
	Store / Move Claim	Credentials are stored in a central database.
	Refresh	Refreshing is provided by the API. Data can be changed using the badge ID and a HTTP PUT request.
	Revoke	Revocation is similar to the OBI hosted badge method. The badge is invalidated and returns a 410 "gone" response when queried.
	Receive	Holders recieve their badge via email and can access it with an URL.
	Assemble	N/A
	Interact	Badges can be shared in any way, either via email or embedded in emails / html.
	Verify	Verification is done by querying the given URL. Upon invalidity, the response code is 410.
	Surroundings	Data model is extendable, anyone can become an issuer and verification can be done by any party.
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Central Database
	References	OpenBadges compliant
	Macro- / Micro-Credential Compatibility	Micro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	Data model is capable of adding meta data
	Identification Method	OAuth 2
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	IBM, Oracle, Pearson, dell, Adobe (among others)
	Maturity	Optimizing
	Target Industry	Professional education

Table 6.8: Framework data for Credly.

6.1.9 CVTrust

CVTrust offers an application called *SmartCertificates*. SmarCertificates is a web-based suite for institutions that allows to issue customized credentials and store the data on the blockchain. According to available information, SmartCertificates is able to issue, verify, store and retrieve claims. However, information about the processes has not been available as of writing this chapter. A request for more information has not been answered. Consequently, the "Actions" section in table 6.9 could not be answered.
A short video introduction on their website shows that certificates can be issued in several formats such as JSON, XML or PDF and that verification is based on using either an URL or a QR code that probably leads to their platform. From its appearance, the overall processes are assumed to be similar to what has been described for other companies. Exemplary, the revocation is achieved by losing the ability to decipher issued credentials on the blockchain. CVTrust is backed by the Horizon 2020 program of the European Union and has strategic partners such as Capgemini or LinkedIn. The target audience for their product is the educational sector such as universities or professional education. The business model comprises three subscription packages. Each package unlocks certain features such as customization or API availability. Common for each package is the maximum amount of issuances included, which is set to 500 per year. Once reached, each issuance is charged individually. Table 6.9 shows the framework data for CVTrust. The maturity is rated as *Quantitatively Managed* [59].

CV Trust		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	N/A
	Store / Move Claim	N/A
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	N/A
	Interact	N/A
	Verify	N/A
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Blockchain
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	N/A
	Identification Method	Issuers have to undergo a manual verification process conducted by the company. Holders have to register to the platform.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Three separate subscription tiers that offer more features the higher the tier is. The number of included issuances it set to 500 for each.
	Usage KPIs	N/A
	Cooperations / Partners	Capgemini, LinkedIn, EFMD Global Network (among others), funded by European Union Horizon 2020 program
	Maturity	Quantitatively Managed
	Target Industry	Educational sector

Table 6.9: Framework data for CV Trust.

6.1.10 Edgecoin

Edgecoin calls its business model Blockchain-as-a-service. The company offers the service of storing data both on the Ethereum blockchain and a connected database, called Inter-Planetary File System (IPFS). The blockchain serves as a permanent, immutable record for the credentials and allows verification by traversing a Merkle tree. The IPFS stores meta data that is too large to store on-chain, for instance photos or attachments. The platform neither provides information about possible actions, nor if the requirements "revocation" or "retrieve claim" are met. On the website, neither KPIs nor partners were listed. Customers are able to enter the pilot phase of the program and extend this period with a "pay as you go" pricing model. Consequently, the maturity is assumed to be *managed*. Table 6.10 shows the available data for this company [60].

Edgecoin		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	No
	Revoke Claim	No
Actions	Issue	N/A
	Store / Move Claim	N/A
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	N/A
	Interact	N/A
	Verify	N/A
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Ethereum blockchain, meta data is stored in IPFS that is connected to the blockchain.
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Fix price for entering the pilot program and afterwards a contract is negotiated.
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Managed
	Target Industry	Educational sector

Table 6.10: Framework data for Edgecoin.

6.1.11 Gradbase

Gradbase is a web-based platform that has two core concepts. The first concept is issuing customizing credentials to the need of the institution. Secondly, Gradbase creates a "portfolio" for every Holder where she is able to access and share her credentials. Parts of the credential are stored on the blockchain, including grades, name, surname and faculty. Other parts such as photo or holder ID are stored in a centralized database that is connected to the portfolio.
Organizations can issue credentials either via excel files containing the necessary data or via user interface. The excel issuance allows to create multiple credentials simultaneously and subsequently issue them. Afterwards, holders receive an email containing a QR code and the credential which can be shared with third-parties. The QR code leads verifiers to the platform showing the credential and its status. Verification is automatically done and does not have to be triggered by the verifier. Based on the blockchain mechanisms, revocation is done by issuing a revocation statement that is visible when traversing a Merkle tree of the blockchain. Moving and refreshing claims is not supported.
Gradbase uses the Bitcoin blockchain for storing credentials. Once an issuer wants to issue credentials using the platform, a manual verification process will be conducted to assert the institution's credibility. Otherwise, the trust model is peer-to-peer.
Gradbase offers the possibility to customize the data model to customers' needs as a premium service. The business model itself is called *Freemium*, meaning that issuers and holders do not pay for issuances. However, premium features are available such as customization, analytical insights and marketing. Verifiers are able to pay a fee for data insights and portfolio recommendations.
As of writing this thesis, Gradbase issues 150-200 credentials per month and has partnered with the London Imperial College as well as the University College London (UCL) Centre for Blockchain Technologies. Upon request, the company responded that an API and an assertion mechanism is in development. Based on this, the overall maturity is assumed to be *managed*. Table 6.11 shows the framework data for Gradbase [61].

Gradbase		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Credentials are issued either via excel upload or using a form in the web interface.
	Store / Move Claim	Credentials are stored completely on the Bitcoin Blockchain.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive their credential through the platform. The platform serves as a repository for credentials.
	Assemble	Several credentials can be grouped together to a CV that can be sent to verifiers.
	Interact	Holders send their CVs to verifiers which contain a QR code. The Code leads to the platform that shows the digital credentials. Sharing can be done via email or social media.
	Verify	Verification is done by the platform. Whenever a credential has been issued to a holder, it is flagged as verified if no revocation is present.
	Surroundings	N/A
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Bitcoin Blockchain
	References	None
	Macro- / Micro-Credential	Macro-credentials
	Compatibility	
	GDPR Compliance	Yes
	API Available	No
	Meta Data	N/A
	Identification Method	Issuers have to undergo a manual verification process.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Freemium model. The base service is free for holders and issuers. Verifiers can pay a fee to see more data by default and get insights to the data.
	Usage KPIs	150-200 credentials issued per month.
	Cooperations / Partners	Imperial College London, Open HR, UCL Centre for Blockchain Technologies
	Maturity	Managed
	Target Industry	Human Resources

Table 6.11: Framework data for Gradbase.

6.1.12 Keeex

As a service provider, Keeex has created an interface that connects to existing architecture in companies to provide hashing and signing documents. Customers are able to send several data types, e.g. PDFs, videos, JPEGs, to Keeex which is in return storing the hash on the blockchain. In this case, there is no issuer-holder relationship. The company or institution that sends the document is assumed to be the holder and issuer in one person. Furthermore, there is no information available about distribution, reception, revocation and refreshing. Keeex offers a verification service on its website where files can be uploaded and a verification process takes place. Each document contains the Bitcoin transaction id where the corresponding hash is stored in its meta data section. It can be assumed that upon verification, the hash of the document and the hash stored in the blockchain will be compared and if both match, the file will be declared valid. However, the exact verification process has not been explained.
Keeex target suppliers and the supply chain industry for their product. Key partners, among other, are the french railway company SNCF, Capgemini and Wiko. Regarding usage, the website offers a timestamp explorer where each Bitcoin transaction is listed that has been issued by Keeex. The overall amount of transactions in February 2020 was 3800. Therefore, the maturity of the model can be assumed as *quantitatively managed*. Table 6.12 shows the data for Keeex [62].

		Keeex
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Keeex offers an interface that takes existing files and signs them. The signature is stored on the blockchain and the blockchain address is stored in the added meta data of the files.
	Store / Move Claim	Hashes are stored on-chain. Moving from one chain to another is not provided.
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	Assembly can be done outside the platform boundaries. Since hashes are stored on the chain, the corresponding transactions can be assembled.
	Interact	N/A
	Verify	Keeex offers a verification process on its website.
	Surroundings	N/A
System	Data model	N/A
	Permission	N/A
	Data Storage Model	Data is stored on the Bitcoin blockchain.
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	Yes
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	The company offers several products that can be embedded into existing processes in other companies.
	Usage KPIs	Approx. 3800 Bitcoin transactions listed.
	Cooperations / Partners	Capgemini, Ledger, Wiko, SNCF (among others)
	Maturity	Managed
	Target Industry	Supply chain industry industry.

Table 6.12: Framework data for Keeex.

6.1.13 Parchment

Based in the United States of America, Parchment claims to be one of the largest and longest existing companies for digital credentialing. With over 8000 customers, Parchment is one of the leading companies compared to the others mentioned in this thesis. Furthermore, the suite Parchment offers is divided into several products. The first one is *SEND*, which allows customers to outsource printing and sending certificates. Furthermore, surcharges can be generated which can be used to fund using this product.
With *Award*, the company offers a product for digital credentialing. It allows customers to issue digital and printed diplomas, as well as verification of these. Whenever a new credential is available, the holder will be notified via text messages. Unfortunately, there is no information publicly available about how the software works, what the data model looks like and where the data is stored. A request for further information has not been answered as of writing this chapter.
Similar to *Award*, Parchment offers *Receive* and *Credential Profile*. Both products handle acquiring, storing and sharing documents. Receive aims at institutions who receive many documents from other institutions, especially from abroad. Therefore, the product can be integrated in institutional processes and is compliant to e.g. CHESICC for document transfer. With *Credential Profile*, students can create a profile for requesting digital diplomas. Each student can add the institution she has attended and use Parchment for ordering the credentials. Additionally, the platform shows institutions which have accepted similar students as recommendations.
Lastly, Parchment offers services such as *Scan and Index* for digitizing records or *Registrar* which is managing online ordering, tracking and delivery of documents.
As Table 6.13 shows, no process documentation or explanation about the software could be retrieved. Yet, the software is rated as *optimizing* due to the amount of customers and the separation of the products.

Parchment		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	N/A
	Store / Move Claim	N/A
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	N/A
	Interact	Credentials can be shared using the Credential Profile product.
	Verify	N/A
	Surroundings	N/A
System	Data model	PDF, printed, customizable
	Permission	Permissioned
	Data Storage Model	N/A
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Digital credentialing suite is split into products such as send, award, receive and analyze.
	Usage KPIs	8000 customers using the Parchment platform
	Cooperations / Partners	BridgeU, CHESICC, CIS, Common App (among others)
	Maturity	Optimizing
	Target Industry	Educational sector and industry education

Table 6.13: Framework data for Parchment.

6.1.14 SAP TrueRec

SAP TrueRec is one of the first blockchain applications the German software company has created. Currently in use at SAP's own MOOC platform, TrueRec allows to issue certificates, hash and store them on the Ethereum blockchain. Serving as an extension of the SAP Leonardo suite, issuing credentials is embedded in the overall SAP workflow. Once an issuer wants to create a credential, she can do so by triggering an automated workflow inside the suite and store the credential in a TRU file. This TRU file is sent to the holder who is solely responsible for storing it. TRU files can only be read with an SAP app. Verifiers are able to open these files inside the specific app, but it also serves as a wallet where Holders can store their credentials and share them with third-parties.
Verification is based on the blockchain mechanisms that above mentioned companies mostly use as well. The hash of the document, which is visible in the app, is compared to the hash on the Ethereum blockchain. Using a smart contract that stores the hash values and revocations makes it easier to navigate through the blockchain platform.
SAP is compliant to the OpenBadges specification and allows the issuance of macro-credentials as well as micro-credentials. The SAP TrueRec data structure is based on JSON linked data and allows to be extended. Currently, the system is still under development and the latest news about SAP TrueRec has been issued in 2017 [63]. For this reason, there are no usage KPIs available nor any pricing model. However, the TrueRec app serves as an extension to the SAP suite and is coupled to the overall SAP business model. It does not serve as a standalone application. The overall maturity of the project is assumed to be *defined* and Table 6.14 shows the data for SAP TrueRec [64].

SAP TrueRec		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Issuers send a TRU file to holders along with the hash of the credential that is stored on the Ethereum blockchain.
	Store / Move Claim	Only the hash is stored, the document is sent to the holder in TRU format.
	Refresh	N/A but in development
	Revoke	Revocation can be done by issuer using a transaction that is stored in a smart contract containing all transactions
	Receive	Holders receive their credentials in their wallets
	Assemble	N/A
	Interact	Holder can share documents via the wallet.
	Verify	Verifiers receive TRU files and can verify them online
	Surroundings	N/A
System	Data model	JSON-LD
	Permission	Permissionless
	Data Storage Model	Ethereum Blockchain for hashes, documents are stored off-chain
	References	OpenBadges
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	Yes
	Identification Method	Holders have to be identified biometrically
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Included in SAP ecosystem, no standalone service
	Usage KPIs	None due to development
	Cooperations / Partners	SAP
	Maturity	Defined
	Target Industry	SAP customers, mainly education

Table 6.14: Framework data for SAP TrueRec

6.1.15 Sony Global Education

Sony Global Education (GED)'s mission is to achieve a certain standardization in the domain of education. The company offers a variety of services and events that are related to online education. One of the services is the Sony GED's blockchain application which allows to issue and store credentials digitally on the blockchain. Different from the approaches other companies use, Sony GED aims to create an own network based on the Hyperledger Fabric framework [65]. The main difference between a Hyperledger-based blockchain network and public blockchains such as Ethereum is that permissions are introduced. A permissioned network cannot be accessed by third-parties without the consent of network stakeholders. The reason why Sony GED chose this approach is not stated, nor is the business model described on the website [66].
The approach comprises also the digitization of credentials. In an info graphic, Sony GED states that original documents have to be submitted to create a digital copy of it. It is neither described how this is done exactly and what the output of the digitization is going to be.
The blockchain application has been deployed at the Global Math Challenge which is a world-wide competition hosted by the company. During this challenge, all credentials were digitally stored on the blockchain network using an API and distributed to the holders. Certificates for this event can still be verified on the Sony GED website.
As there is only little information available about the project, the maturity is assumed to be *defined*. Table 6.15 shows the framework data for this project [66].

Sony Global Education Platform		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Printed / Original credentials have to be sent to the platform for digitization and confirmation. Then, the data is stored on the blockchain.
	Store / Move Claim	Credentials are digitized and stored on the blockchain.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive their credentials in a wallet.
	Assemble	N/A
	Interact	Data can be shared by holders in their wallet.
	Verify	Data is verified on the blockchain.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	Own blockchain network using Hyperledger Fabric
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	Yes
	Meta Data	Yes
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	System was used in the Global Math Challenge.
	Cooperations / Partners	N/A
	Maturity	Defined
	Target Industry	Educational sector

Table 6.15: Framework data for Sony Global Education platform.

6.1.16 Sproof

Sproof is based on a scientific paper by Brunner, Knirsch, and Engel [67]. In this paper, the authors explain in detail the architecture and algorithms of the platform. The company created an application that can be integrated into the existing architecture of institutions and allows to issue documents fully automatically. Via a docker container, developers can attach their system to the API Sproof offers. During the issuing process, Sproof hashes and signs documents. The hash is stored on the blockchain whereas the document resides in an IPFS that is connected to the blockchain. Holders can use pseudonyms to be identified by the system that serves as an address to issue the credential to. Once issued, credentials cannot be changed anymore. Each credential stored on the Ethereum blockchain is tamper-resistant and immutable. However, data stored in the IPFS can be changed. Similar to the above described systems, revocations implemented by issuing a transaction that contains the ID of the credential and a revocation reason.
Holders receive their credential by sending the issuer an address, e.g. a public key or derived public key, and receive a notification once a credential or document is available. Then, holders can distribute a link to the document to share it with verifiers.
Sproof's data model is based on JSON format and can be extended using `schemas` [68]. Listing 6.1 shows an example call. In line two, the schema is stated that is used for the data structure of the call. Furthermore, parts of Sproof are open-source and can be downloaded on their GitHub page [69].
Regarding the business model, Sproof offers three subscription models based on the included amount of transactions. Each tier differentiates between events, such as updates or revocations, and transactions that are responsible for issuing data to the blockchain. The smallest tier includes 200 events per month and one daily transaction. Transactions can group several issuances together in a batch.
The Sproof app is available without registration and grants an insight about the platform usage. At the time writing this chapter, daily usage for the past several days has been visible. The exact amount of documents issued so far could not be determined. However, it can be assumed that the overall maturity based on usage, business model and the scientific background is *defined*. Table 6.16 shows the framework data for Sproof.

		Sproof
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Credentials are hashed and stored on a public blockchain. The holder submits a pseudonym to the issuer serving as an address for the credential.
	Store / Move Claim	Hashes are stored on the blockchain and meta data is stored in an IPFS.
	Refresh	The API offers an update call to edit issuer data, but credentials cannot be updated.
	Revoke	Revoke event is triggered that stores a hash of the document and a revocation reason.
	Receive	The holder receives her credential by sending the issuer an address (e.g. a public key or derived public key).
	Assemble	N/A
	Interact	Holders can send the blockchain entry to verifiers.
	Verify	Sproof offers a database that can be downloaded by verifiers to see the current state of Sproof. There, each event is stored. The verifier is then able to see every credential with the corresponding issuer and revocation events.
	Surroundings	Anyone can become an issuer, holder or verifier. Data model is extendable.
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Blockchain for document hashes and IPFS for documents.
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credential
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	Yes
	Identification Method	Identity evidence (e.g. DNS identification) and PKI for issuers, PKI for verifiers and holders
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Subscription packages based on volume of annual events.
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Defined
	Target Industry	Mainly educational sector but open to other industries as well.

Table 6.16: Framework data for Sproof.

Listing 6.1: Update call showing the Sproof data structure. Adopted from [68].

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "title": "Update Profile",
  "description": "Update profile sproof event",
  "type": "object",
  properties :{
    "eventType" :{
      "type" :"string",
      "enum" :["PROFILE_UPDATE"]
    },
    data: $ProfileSchema
  },
  "required" :['eventType', 'data']
}
```

6.1.17 Stampery

Similar to SAP TrueRec (Subsection 6.1.14), Stampery aims at securing documents instead of issuing educational records. The business model behind this startup is to create an API that connects to a distributed cluster of computers, the "Stampery BTA" (BTA stands for Blockchain Timestamp Architecture), where documents are hashed and anchored to multiple blockchain networks at the same time. Therefore, Stampery has created an own algorithm that queries e.g. the Ethereum and Bitcoin blockchain and creates a Merkle tree of both. Since both networks have different lifetimes of the Merkle tree (Bitcoin approximately 10 minutes and Ethereum 1 Minute), the algorithm rebuilds and Merkle tree of the Ethereum network for each minute until the ten minute mark is reached. Then, a leaf is appended containing the hash of the documents. This assures that both blockchains contain the document's hash at the same level.
The reason behind this parallel anchorage is based on two assumptions: The Bitcoin network has a large hashing power with the downside of latency and speed when it comes to confirmation. Contrary to that, Ethereum has a "lower hashing power" [70] but is faster in terms of confirmation. Anchoring the data in the Bitcoin network as well as in the Ethereum one benefits from the security of BTC and the availability of ETH.
Stampery aims at the governmental sector and enterprises that need auditing of their documents. A first deployment of their software has been done for the Estonian government where residents are able to register of an electronic residence ID. Furthermore, the startup is backed by investors such as boostVC and Blockchain Capital. A pricing model along with customers apart from the Estonian government are not provided. The overall maturity of the company is assumed to be *Defined*. The framework data for Stampery is provided in Table 6.17 [71].

Stampery		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	No
	Revoke Claim	No
Actions	Issue	N/A
	Store / Move Claim	N/A
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	N/A
	Interact	N/A
	Verify	Verification is based on generated cryptographic proofs.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Anchorage in several blockchains such as Bitcoin and Ethereum
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	Yes
	Meta Data	N/A
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	Estonian Government, Blockchain Capital, boostVC
	Maturity	Defined
	Target Industry	Governmental and educational sector

Table 6.17: Framework data for Stampery.

6.1.18 Vottun

As already mentioned in subsection 6.1.6, Vottun is a company that offers a framework to build blockchain solutions. One of them is Blockeducate that has been described earlier. The Vottun protocol is comprised of several layers. At the very top, companies that use the underlying technology and build a business model with Vottun are located. The second layer consists of an API that interacts between the companies and the Vottun back-end. In the bottom layer, Vottun connects with several blockchains such as Ethereum, Bitcoin and Hyperledger and pushes the data received via smart contracts (if available) to the blockchain.
Vottun also serves products that target several industries such as suppliers or educational institutions. For the latter, Vottun has a digital credentialing platform that is compliant to the OpenBadges specification and allows integration to learning management systems of various companies. Here, Vottun uses its protocol architecture and deploys an own business model along with user interfaces on top. The product enables institutions to issue credentials within the system and store the data on a blockchain at their will. Furthermore, data can automatically be refreshed when it expires or is reactivated. Users are able to assemble micro-credentials into larger macro-credentials by setting requirements and clustering the achievements internally. Therefore, holders receive their credentials in a wallet where they can see and stack their achievements. Verification of achievements is provided via cryptographic proofs.
Vottun's list of partners features large companies such as ATOS and PricewaterhouseCoopers. Overall, the maturity of the software is assumed to be *optimizing*. Vottun offers a solution for a variety of industries, has an own portfolio of products and mentions customers such as Santander or Naturgy which are large companies. Table 6.18 shows the framework for Vottun [72] [73].

Vottun Credentials		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Credentials are issued within the system or via an integration in the existing enterprise architecture.
	Store / Move Claim	Data is stored on the blockchain.
	Refresh	Realized via smart contracts that keep track of expiration dates.
	Revoke	N/A
	Receive	Holders have a digital wallet where they receive their credentials.
	Assemble	Holders can assemble their credentials on the platform and stack badges to create a full diploma.
	Interact	Holders can share their credentials from their wallets via social media or email.
	Verify	Verification is based on cryptographic proofs.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Data is stored on the blockchain.
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	N/A
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Platform will be integrated in and customized for existing architecture.
	Usage KPIs	N/A
	Cooperations / Partners	ATOS, PwC, BiT, RS
	Maturity	Optimizing
	Target Industry	Educational sector, Supply Chain Management, Banking, Government, Insurance

Table 6.18: Framework data for Vottun.

6.2 Applying the Framework to Research Projects

The purpose of this section is to investigate what researchers are working on in the domain of digital credentialing. During the literature aggregation and research, 18 publications were selected for applying the framework to. In the process of writing this chapter and analyzing the project, three were eliminated and another five were grouped together since they concerned the same project. Selecting the papers was based on two factors:

1. Does the paper present a complete or partial aspect of a digital credentialing platform?

2. Does the platform deal with macro-credentials primarily?

Furthermore, this chapter only provides an excerpt of the research for digital credentialing.

6.2.1 Blockchain and Smart Contracts for Digital Certificate

In the report *Blockchain and Smart Contract for Digital Certificate*, the authors describe a simple platform for issuing digital credentials and a verification process. At the core of this platform is the Ethereum blockchain that serves as a database for issued credentials. The platform has three stakeholders: *certification units*, *students* and the *service provider*. Certification units such as schools are responsible for issuing certificates. Therefore, the platform has a web-based front-end where the institution can enter data about a student. The system automatically assigns the student's serial number to the credential. This way, it links the document to her entity. Before issuing, the system runs a data verification check. Afterwards, the data is signed and stored on the blockchain with its serial number. Students are then notified via email that a new credential has been issued and can access it via the platform. Each credential contains a QR code and a serial number that can both be shared with third-parties. They use the QR code to check if the credential can be verified and receive a valid or invalid response. The service provider is responsible for maintaining the system and keeping the it running. Further information about this role has not been stated.
Students can register to the platform by entering basic information and setting a password. How issuers connect to the platform and how their identity is assured has not been stated. Neither is information provided about how smart contracts are involved in the architecture and which role they play. The system has been prototyped, but no usage KPIs are available. The overall maturity of the system is assumed to be *initial*. Table 6.19 shows the framework data for this project [74].

Blockchain and Smart Contract for Digital Certificates		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Institutions fill in a form containing the credential data and the holder's data. System generates a serial number.
	Store / Move Claim	The entered data is stored along with the serial number on the Ethereum blockchain forming a credential.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive a notification email with when a new credential has been issued. The credentials can be accessed on the platform.
	Assemble	N/A
	Interact	Holders pass the serial number or a QR code to verifiers.
	Verify	Verifiers look up the serial number in the system and get "valid" or "invalid" in return. The verification process is based on Merkle tree traversal.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Data is completely stored on Ethereum blockchain.
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	Holders register on the platform using their real names and password. No information regarding issuers available. Verifiers do not have to identify.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.19: Framework Data for *Blockchain and Smart Contracts for Digital Certificates*.

6.2.2 Blockchain Education Platform

Developed by researchers of the Fraunhofer Institut in Germany, the Blockchain Education Platform (BEP) is a prototype to fulfill the requirement of digitizing and automating issuance and management of digital credentials. Based on a decentralized approach using a blockchain, smart contracts and an IPFS, the researchers have created a prototype enabling institutions to do that.
The BEP meets most of the requirements stated in the framework as Table 6.20 shows. Assertion and moving of credentials is not stated, which results in the "not available" (N/A) notation. Furthermore, the system is compatible to the OpenBadges specification using a data model that can be extended with schemas and allows to add meta data.
As with other systems already described in this chapter, the BEP has three stakeholders who are participating in the system: *certification authorities*, *learners* and *employers* [75]. These translate to the issuer, holder and verifier as defined for this thesis. Each role has a unique set of features such as issuing certificates and managing these for the issuer. Holders are able to assemble them into portfolios and share these portfolios with verifiers. Verifiers are able to run validation and verification checks on either a single credential or the whole portfolio.
Different from the common approach is that the BEP uses smart contracts for access and certificate management. With the first smart contract the system checks the identity and rights a role has. For example, the issuer has to register as a *certification authority* to the smart contract that grants the right to issue certificates in the system. The latter smart contract is responsible to manage the lifecycle of issued credentials on the blockchain. A common use case for such a contract would be the automated revocation once a credential has reached its expiry date.
Peers in the system can remain anonymously. However, for certification authorities this is counterproductive. To gain credibility towards verifiers, a certification authority should create a profile that is stored in an IPFS and linked to the Ethereum address so that verifiers are able to check if the issued credential stems from a valid institution. The main advantage to store these profiles in an IPFS and not on the blockchain itself is that data can be mutated and held private at the same time. According to the authors, profiles contain sensitive information such as who is working for the institution and who is personally in charge for issuing certificates at the moment. To be compliant to the GDPR, this data has to be handled with caution and consequently should not be stored on the blockchain.
On top of the architecture is a web-based user interface that allows to issue, share and manage credentials. Different from other platforms, holders will be notified whenever a credential is processed by third-parties, e.g. a verification takes place. Furthermore, shared credentials can be un-shared again and holders are able to upload certificates that have already achieved.
The overall maturity of the project is rated as *initial* since it is in prototype state and not for public usage. Table 6.20 shows the framework data for the BEP [75].

Blockchain Education Platform		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Issuer collects all the necessary data for the credential, signs it and issues the fingerprint to the blockchain.
	Store / Move Claim	Document is signed and stored in a central database. Fingerprint is stored on-chain.
	Refresh	N/A
	Revoke	Available, but process is not described.
	Receive	Holders receive their credentials on the platform and are notified when a new credential is issued.
	Assemble	Holders are able to create a portfolio with the help of the document management system.
	Interact	Assembled portfolios can be shared with verifiers. Each interaction sends a notification to the holder, e.g. a verification.
	Verify	Blockchain-based mechanism. Either a single credential or whole portfolio can be verified at once.
	Surroundings	Data model can be extended using schemas.
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Blockchain for fingerprints, central database for documents
	References	OpenBadges compatibility
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	Yes
	API Available	No
	Meta Data	Can be added via schemas.
	Identification Method	Smart contracts control access management based on Ethereum addresses. Verifiers have to provide an IPFS profile to avoid anonymity.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	Only prototype implemented
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.20: Framework data for Blockchain Education Platform.

6.2.3 Blockchain-Based Education Records

In their work, the authors approach the implementation of blockchain-based system for education from another angle. They describe an architecture that can be used instead of a full system that is deployed on top of the blockchain layer. Figure 6.1 shows this architectural approach and denotes the workflow of how credentials are issued from A to G. In the center of this operation are three peers: the *provider node*, *individual node* and the *miner*. At the beginning of an issuing process is the record (or credential) that is processed by the provider node. Here, the record enters the *Education Record Manager* which is responsible for managing the data and has access to an off-chain database. The credential is processed until it is ready for deployment on the blockchain (denoted as B). Each deployment (or issuance) is included in a block that has to be mined by the miner. The credential is attached with a signature, hash value, index and a resource URL. Therefore, only the provider and the individual who is owning the credential can access the data. Afterwards, the credential can be accessed by individual nodes according to the rule set a provider has defined for its credentials. As Figure 6.1 depicts, the individual node mirrors the provider node and has the ability to update the credential and retrieve its status. The holder is then able to query the Education Record Manager at the individual node to obtain her credentials. Both the individual node and the provider node are able to communicate and synchronize.

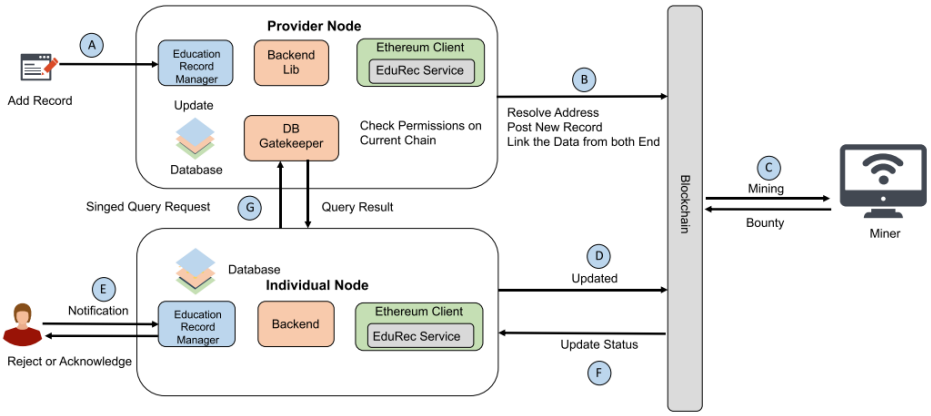


Figure 6.1: Blockchain-based architecture with three participating peers: Provider node, individual node and miner. Adopted from [76].

Smart contracts provide the ability to further define roles and relationships on-chain. An example for smart contract usage would be the relationship that an individual node and a provider node have within the blockchain network. The provider node could assign *stewards* and *owners* having different access rights to operations. Another possibility would grant viewership rights to certain individuals.
Although the Ethereum client is featured in the architecture, the authors do not reference this blockchain as their go-to solution. What they state is a blockchain that consists of

trusted peers, making the proof-of-work concept obsolete. In this system, each participating peer is authorized (therefore needs a permission to access the network) and can provide proof. Consequently, the system does not have to offer a proof on the data set level as it is the case with public blockchains.
The overall maturity of the system is rated *initial* since there is no data available that goes beyond the architectural design of the system. Furthermore, no partners or prototypes have been mentioned. Table 6.21 shows the framework data for this approach [76].

A Novel Blockchain-Based Education Records Verification Solution		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store / Move Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Credentials are issued by provider nodes who assemble the data, make it blockchain-ready and send it to the network.
	Store / Move Claim	Data is stored on-chain as a hash and in a database at the provider and individual node.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive a notification once the credential is issued and can access it through individual nodes.
	Assemble	N/A
	Interact	Interaction is provided by smart contracts and individual nodes.
	Verify	Verification is based on trusted peers rather than trusted certificates. Additionally, documents are always hashed which can be used for verification.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	Blockchain
	References	N/A
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	Hierarchy of identification. Peers have to be authorized by identification provider nodes.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.21: Framework data for *Blockchain-Based Education Records*.

6.2.4 Blockchain-Based Educational Record Repository

The blockchain-based educational record repository, in short BcER2, is a proof-of-concept using the Hyperledger Composer framework for creating blockchain applications. Here, the authors propose a consortium blockchain that is permissioned for manipulating the data but can be verified by anyone. The implementation is based on a "business model" that provides an overview of rules and mechanisms that are featured by this application. One example is the issuance of credentials: A university creates an asset that contains data such as student, institution, record type and more along with a selection of nodes that are responsible to bring the data into the network. Once this process is triggered, the selection of nodes perform a consensus mechanism and mine the data in a block. The specialty of the business model is that it defines engines that are responsible for access control as well as how transactions are defined. With these components, the system can be customized and adapted in a modular way.
The system has undergone a first testing phase at the University of Salvador for a proof-of-concept. As future work, the authors recommend to test the scalability of the system and extend the deployment, as well as add features that enable interaction with stakeholders. The system is rated as *initial* and Table 6.22 shows the data for this project [77].

		BcER2
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Credential is created and linked to student identifier. Afterwards, the document is time-stamped on the blockchain and validated by nodes in the network. Then it is added and dated to the blockchain.
	Store / Move Claim	Data is stored on the blockchain once it has reached consensus.
	Refresh	N/A
	Revoke	N/A
	Receive	N/A
	Assemble	N/A
	Interact	N/A
	Verify	Educational records can accessed using ID cards.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	Consortium Blockchain
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	Yes
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	Deployment at Salvador University (UNIFACS) for proof-of-concept
	Cooperations / Partners	UNIFACS
	Maturity	Initial
	Target Industry	Educational sector, professional training, workforce development

Table 6.22: Framework data for BcER2.

6.2.5 Blueprint for Learning Trace Repositories

Reading and analyzing the previous subsection it becomes clear that certain requirements and parts of the framework such as assembly and assertion have never been met by the creators. This is different for the proposed system here. The authors state a blueprint which is based on requirements that resemble the ones in the framework described in Chapter 4. Exemplary, the authors state the requirement "Data Ownership and Access" [35] that is approach by designing "pluggable" repositories. This means, that each repository can be moved and consequently the stored data inside of it as well. Another requirement is that data has to be aggregatable. Therefore, the data inside of a repository can be aggregated into a larger construct, such as a macro-credential.
At the beginning of the system is the integration into the daily routine of educational institutions. Learners are able to enroll in one or more learning activities that generate learning traces. Each learning trace describes an interaction such as taking an exam or conducting a thesis. At the end of each learning activity, a learning block is created that contains meta-data about the learning activity, the learner, resources and more. Once the block is formed, the learner signs it with her private key and optionally send it to different peers who can sign it as well. In the next step, the block is sent to peers who are required to sign it as well, such as a supervisor of the learning environment. Afterwards, the block is hashed and its hash is stored on the blockchain. The block itself is stored in a repository, e.g. a centralized database. Who is owning these repositories can be defined in the enrollment process. Owners can either be the issuer, one or more learners or a third-party.
Each block has to be verifiable by third-parties. Therefore, a permission system is in place that allows to grant access to third-parties. Responsible for that is the owner of the repository. Each repository represents an own entity and can be configured locally so that no one has to approve granting access to anyone apart from the owner. Once the access is granted, verifiers can generate the hash of a learner block and compare it to the one stored on-chain.
Overall, the system is divided into three layers: an "application layer", a "blockchain layer" and a "communication layer" [35]. The application layer contains the components that are responsible for learning activities, generating learning traces and storing these in a block. The blockchain layer picks up these blocks and creates a connection between the application layer and the blockchain. Furthermore, the blockchain layer is responsible for the enrollment process (storing public keys and "network addresses" of the repositories [35]), access control, data aggregation and emission of events. All of this is realized by using smart contracts. Lastly, the communication layer serves as a bond between the application and blockchain layer, but also as a gateway to the system. By design, the API allows the system to be modular and independent from e.g. a blockchain architecture. Furthermore, it enforces the rules defined by both layers, including access controls or distributing messages to end users.
As a result, the trust model differentiates as well from previously described systems. Here, the authors state that the system, especially the communication and blockchain layers,

have to be trusted. In example, a malicious communication layer would be able to grant access to attackers or register faulty records to the blockchain. Regarding the blockchain layer, each network has to provide immutability and formal verification. If this is not provided, the network cannot be trusted. On the contrary, the application layer containing the repositories and learning blocks must not be trusted.
As this is a blueprint for creating such systems, there has not been a deployment yet. The authors have surveyed 25 teachers regarding the requirements and confirmed that the ones stated in their paper are relevant for the educational sector. The project's maturity thus is rated *initial*. Table 6.23 contains the framework data for this project [35].

Blueprint for Learning Trace Repositories		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	Yes
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	Yes
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	Learning activities are formed into blocks that are hashed. The hash is issued to the blockchain.
	Store / Move Claim	Blocks are stored in a repository (off-chain). The hash is stored on chain.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive their claims inside of a repository. This can be owned either by the institution, the holder or a third-party.
	Assemble	Macro-credentials can formed out of several blocks.
	Interact	Smart contracts and the communication layer provide tools for interaction.
	Verify	Based on comparing the hash value of a block with the one stored on-chain.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless
	Data Storage Model	Blockchain for hashes, repositories are stored off-chain.
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	Yes
	Meta Data	N/A
	Identification Method	Holders have to enroll once where the repository and public keys are generated and stored.
	Trust Model	Communication and blockchain layers have to be trusted.
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.23: Framework data for *Blueprint for learning trace repositories.*

6.2.6 Certificate Verifying Support System

Certificate Verifying Support System (CVSS) describes a "blockchainized" solution for credential management. The system is comprised similar to other solutions already described: It features roles for issuer, holders and verifiers as well as the possibility to issue, revoke, store and share credentials on the Ethereum blockchain. Furthermore, the authors have implemented a known approach for identification called the *Know-Your-Customer* principle (KYC). Issuers have to register to the CVSS and provide information such as tax code, physical address and an email address. Once this check is completed by CVSS, it sends an email to the given address and an issuer is activated within the system. This ensures the system's credibility and trustworthiness.
Issuing certificates is kept simple and involves entering the credentialing data, authorization from the issuer's side and send the document to the blockchain where the issuer's smart contract stores the hash. Simultaneously, it is distributed to the holder's smart contract where she can share the credential with verifiers. Apart from the blockchain-process, issuers distribute copies of the credential either in print or electronic from. Both contain a Quick Response (QR) code which leads the verifier to the hash on the blockchain. During the verification, the hash value of the document is compared with the value stored on-chain. If the values match, the certificate is valid.
Revocation is also supported. Issuers can trigger a transaction that contains the information about a credential and store this in their smart contract. When the information is retrieved, the credential is flagged as invalid. A new credential can be issued in case the revocation reason was faulty information.
The system has been deployed for test reasons at the Ho Chi Minh City (HCMC) University of Technology using the Ethereum test network. During the deployment, the university issued certificates for two blockchain-related courses. Apart from this test, there is no usage stated. The project's maturity is rated *initial* since only a prototype has been deployed, no business model was stated and partners were not listed. Table 6.24 shows the framework data for CVSS [78].

		CVSS
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	No
	Revoke Claim	Yes
Actions	Issue	Issuer logs into the system and enters certificate data. This is hashed and stored in a .cvss file. After two-factor authorization, data is issued on the blockchain.
	Store / Move Claim	The hash of the certificate is stored on the blockchain and assigned to a student smart contract.
	Refresh	N/A
	Revoke	Revocation is based on transactions that mark the certificate as invalid.
	Receive	Holders receive their credential via email and have a link in their blockchain wallet.
	Assemble	N/A
	Interact	Holders can share their credentials to verifiers from their wallet. Verifiers receive either a link or a document with a QR code.
	Verify	Verification is based on comparing the hash of the document with the one stored on-chain.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissionless but identification process in place.
	Data Storage Model	Ethereum Blockchain
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	Name, identity, proof of affirmation, date of identification
	Identification Method	Know-your-customer principle for issuers. Holders and verifiers do not have to register.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	Test deployment at the HCMC University of Technology with small numbers of issuances.
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.24: Framework data for CVSS

6.2.7 CredenceLedger

CredenceLedger is a digital credentialing platform based on a permissioned blockchain. Different from public ones, the permissioned blockchain requires an invitation to join the network and is not publicly accessible. The software is based on MultiChain which is a framework for setting up blockchain-based enterprise networks [79]. MultiChain has a feature called "data streams". It uses the blockchain as an "append-only database" [79]. Different from other systems based public blockchains where each transaction costs a fee, streams are referenced in transactions and do not utilize cryptocurrency when used.
The system is comprised of a mobile app where holders can receive and share their credentials. An interface for third-parties such as employers or educational institutions has not been defined in the proposal. The overall maturity of the project can be defined as *initial* since it rather serves as an idea instead of an ongoing implementation. The proposal shows that permissioned blockchains have the potential to further secure access to sensitive data such as credentials. Table 6.25 shows the framework data for CredenceLedger [80].

CredenceLedger		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	N/A
	Store / Move Claim	Hash is stored on-chain.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive credentials in a mobile app and printed.
	Assemble	N/A
	Interact	Credentials can be shared through the mobile app.
	Verify	Based on blockchain mechanics (comparing the hash).
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	Blockchain
	References	None
	Macro- / Micro-Credential Compatibility	Macro-credentials
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	Issuers will be identified upon invitation.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational Sector

Table 6.25: Framework data for Credence Ledger

6.2.8 Distributed Credit Transfer

Credentialing platforms usually deal with documents that are digitized or digitally created, stored and distributed to stakeholders. Inside the university, however, macro-credentials are comprised of several sub-achievements that each grant credits. In Europe, this credit system is called European Credit Transfer System. Commonly, each course grants a certain amount of credits that the student receives in her account. Depending on the amount of credits, the macro-credential (e.g. a diploma) is issued when all courses have been passed and the required amount of credits is achieved. This subsection deals with an approach that takes the existing credit system and sets it up in a distributed context.
Srivastava, Bhattacharya, Singh, et al. describe a blockchain-based platform that is permissioned and consists of universities, students and employers. Universities serve as the gate keeper of the system and form nodes that participate in the custom consensus protocol. To invite a new node, each peer has to agree on the decision, resulting in an invitation link to a new peer. When the node accepts the invitation, a transaction will be issued that puts the information about a new participating node inside a block and appends it to the network. Using the consensus protocol, each node will be globally informed that a new peer has joined the network.
Students are added to the network by universities. Each student has a wallet that contains a public and private key. Along with personal information such as the enrollment number, the address of the wallet is also mined into the system. Then, the entity is known to participating peers. Once this is set up, credits can be transferred to students by spending tokens. Whenever a student has successfully completed a course, credits are transferred to her wallet via transactions that are issued by the university. Therefore, a multisignature system is in place so that only the university the student is currently enrolled at can transfer credits. Furthermore, the multisignature system allows to create a new block that is signed by the university's private and the student's public key. Via a private channel, a verifier asks the student to sign this block with her private key in order to verify the data contained inside. Once this is done, a verifier is able to see that the data is signed by both parties and knows that it is correct.
The major use case for this system is rather transferring credits from university A to student 1 or sending student 1's credits from university A to university B. However, employers are more interested in meta-data that is detached from the internal credit system universities are using. With the current approach, this data is not provided by the system. Since this is a proposal for a customized blockchain system for universities and has not been deployed yet for larger tests, the maturity is rated as *initial*. Table 6.26 shows the framework data for this architecture [81].

Distributed Credit Transfer		
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	Yes
	Retrieve Claim	Yes
	Revoke Claim	No
Actions	Issue	Credits are issued by universities when courses have been passed by students.
	Store / Move Claim	Credits are stored in a wallet that is assigned to a student.
	Refresh	N/A
	Revoke	N/A
	Receive	Students receive credits at the end of each semester inside their wallet.
	Assemble	Credits can be assembled easily by creating the sum of all credits.
	Interact	Students can transfer their wallet from one university to another by creating a new multisignature.
	Verify	The university signs a block that contains the credit-data with its private key and the student's public key. The student has to sign it with her private key to verify it to third-parties.
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	Blockchain
	References	None
	Macro- / Micro-Credential Compatibility	None
	GDPR Compliance	No
	API Available	No
	Meta Data	N/A
	Identification Method	Custom consensus protocol for joining the network.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.26: Framework data for *Distributed Credit Transfer*.

6.2.9 Educational Certificate Blockchain

The Educational Certificate Blockchain (ECBC) presented in this paper targets educational institutions such as schools and universities. This is important to know since the blockchain system proposed here depends on a permission-system for becoming a part of the network. Each institution needs an invitation to become a *peer* in the network. Peers, or issuers, are members that participate in the consensus algorithm and are responsible for the block creation. Based on this, the authors describe a new consensus algorithm that replaces the proof-of-work one used by Bitcoin and Ethereum, called *cooperation consensus*.
The very first assumption for the cooperation consensus is that the byzantine threshold, meaning nodes with the intention to gain control over the network, is much lower. Only one third of the total amount of peers is necessary to run into the byzantine problem and therefore reach the overall power over the network. Consequently, the permission system has to provide strong checks to keep the network operating and avoid the byzantine problem. Furthermore, the algorithm involves three steps to create a block. In the first step, the selected nodes for the quorum reach consensus about a block's link value. The link value of a new block is computed by sending each peer a random number, taking the previous block's link value and combine all of this with the Merkle root of transactions in this block. This data is then time stamped and hashed. As this value is comprised of several, partly randomized data, the chance to falsify this block can be minimized for the case that an attacker wants to re-create a chain and changing the data in e.g. one block.
Next, the quorum selects a primary node who is in charge for the block creation. This is based on the random numbers distributed to the selection of peers. Since randomization is not based on any rules, an attacker could use this step to infiltrate the system and become the primary peer, rendering the algorithm useless. Therefore, each node requests the random numbers from every other node and files a list. Each peer then compares the list with the list of other peers. If a list is not matched, the node who generated the random numbers will be eliminated. After that, the block with the number closest to the average of all random numbers will become the primary node for the block creation.
In the third step, the block is created and each peer validates the block structure. If a peer agrees that a block has been created correctly, she votes for appending it to the blockchain. If a block gets the majority of votes, which is *half of the amount of the quorum + 1*, then the block is attached to the chain.
Apart from this algorithm, the ECBC features a new structure which is a mix of a Merkle tree and a Patricia tree. In public blockchains such as Ethereum and Bitcoin, Merkle trees are used. Due to the size of the blockchain, querying data can become an intensive task since the tree has a large spread. Using a mix of Patricia and Merkle tree, in short MPT, makes querying the data faster and more efficient without loosing the benefit of data integrity. In a test with over 1.6 million transactions, the authors showed that their approach led to fast results, even when using more complicated queries.
Regarding the framework data in Table 6.27 it can be shown that research focuses more on the infrastructure than how the system would meet the requirements and processes of a digital credentialing system. Therefore, the maturity is rated *initial* [82].

		ECBC
Category	Key	Value
Requirements	Issue Claim	Yes
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	No
	Revoke Claim	Yes
Actions	Issue	N/A
	Store / Move Claim	N/A
	Refresh	N/A
	Revoke	A transaction of type "revoke" can be used to invalidate an issued credential.
	Receive	N/A
	Assemble	N/A
	Interact	N/A
	Verify	N/A
	Surroundings	N/A
System	Data model	N/A
	Permission	Permissioned
	Data Storage Model	MPT-Blockchain
	References	OpenBadges
	Macro- / Micro-Credential Compatibility	N/A
	GDPR Compliance	No
	API Available	No
	Meta Data	Type of certificate, HolderID, IssuerID, type of operation
	Identification Method	Issuers have to get a permission to join the network. Holders can remain anonymously.
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	N/A
	Usage KPIs	N/A
	Cooperations / Partners	N/A
	Maturity	Initial
	Target Industry	Educational sector

Table 6.27: Framework data for ECBC.

6.2.10 QualiChain

QualiChain is a project funded by the European Union's Horizon 2020 program and executed of a consortium lead by the Open University of the United Kingdom (OU UK) including several industry and research partners such as *ATOS*, *Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung* and *Technische Informationsbibliothek Germany*. The project goal is to evaluate the impact of new technologies such as blockchain and decentralization in combination with "(...) algorithmic techniques and computational intelligence (...)" [83].
Similar to the roles defined for this thesis, the QualiChain platform has stakeholders separated into three categories: Seekers, Validators and Providers. A seeker can be compared with the role of a holder. Seekers receive and hold credentials that are issued by providers such as educational institutions. Validators receive an assembly of credentials in the form of a CV. Each role is divided into several sub-roles which have access to certain data models and services. The seeker, for instance, can be distinguished into a job seeker, a lifelong learner or a student. Each sub-role has different requirements to the tool based on the use case. For example, the student has the requirement to be notified whenever a new credential is issued for her. A job seeker, however, should rather be notified for new job vacancies instead of the issuance of credentials. Lastly, the lifelong learner has to notified whenever new courses are available for her [84].
Different from products described in the previous section, the state of the project is completely transparent including services, interfaces and the overall system architecture. A closer look at this project would be out of scope of this thesis since the currently available deliverables comprise approximately more than 400 pages and the complexity is high enough for an own thesis. In short, the project features a modular architecture that is separated in engines. There are three main engines that deal with the features the project comprises [85]:

- Validation and Verification Engine: As the name indicates, this module is responsible to audit credentials, translate them and search for semantic equivalences.
- Recruitment and Competency Management Engine: Recruiters can set up profiles, qualifications are screened and matched to job postings, a decision support system helps finding the right candidate and insights are generated.
- Profiling and Career Management Engine: In this component, verified credentials are stored in intelligent profiles that interact with a career advisor module. Additionally, verification requests are handled by this engine.

Each engine has interfaces to different parts of the application as well as a connection to the blockchain-based "registry of verified qualifications" [85]. Figure 6.2 shows the described architecture model that is adopted from [85].
In conclusion, this project is the most ambitious and complex approach rooting in a research environment. The set of features and use cases exceed the ones described in the previous section and has not been met by any company examined. A further investigation of the

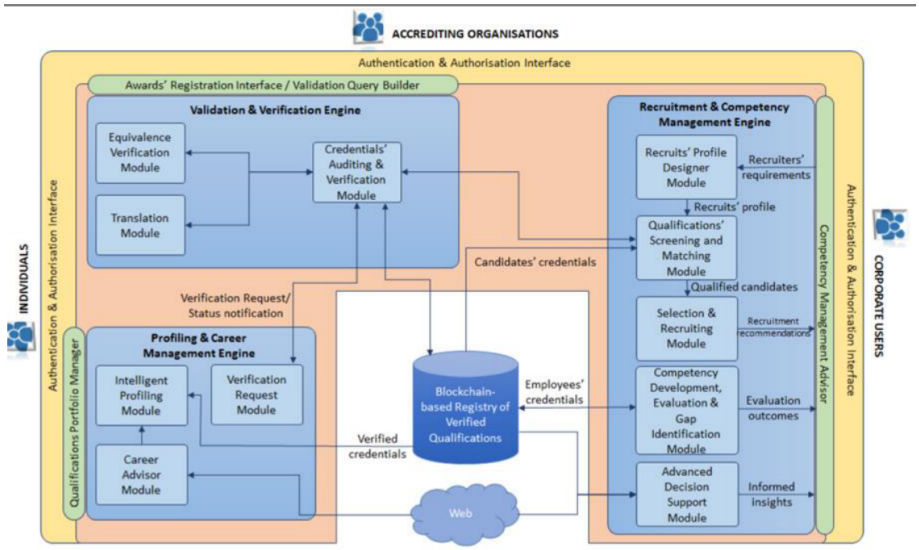


Figure 6.2: QualiChain architecture separated into several engines. Adopted from [85].

project that is funded by the EU until 2022 can be a part of future work. As the framework states in Table 6.28, some actions could not be described because of missing information.

QualiChain		
Category	Key	Value
Requirements	Issue Claim	No
	Assert Claim	No
	Verify Claim	Yes
	Store Claim	Yes
	Move Claim	No
	Retrieve Claim	Yes
	Revoke Claim	Yes
Actions	Issue	N/A
	Store / Move Claim	Verifiable credentials are stored on a blockchain.
	Refresh	N/A
	Revoke	N/A
	Receive	Holders receive credentials inside the platform that is called "intelligent profile".
	Assemble	N/A
	Interact	Interaction can be done inside the platform and is also supported by algorithmic computation.
	Verify	Verification is based on hashing documents and comparing the hash with data stored on-chain.
	Surroundings	N/A
System	Data model	JSON
	Permission	Permissionless
	Data Storage Model	Blockchain
	References	OpenBadges
	Macro- / Micro-Credential Compatibility	Macro- and Micro-credentials
	GDPR Compliance	Yes
	API Available	Yes
	Meta Data	Yes
	Identification Method	N/A
	Trust Model	Peer-to-peer
Business	Business Model / Pricing	Public Domain
	Usage KPIs	N/A
	Cooperations / Partners	Funded by the European Union’s Horizon 2020 program, ATOS, Fraunhofer Institute, OU UK
	Maturity	Initial
	Target Industry	Public and educational sector

Table 6.28: Framework data for QualiChain.

6.3 Evaluation of the State of the Practice

In the previous subsections, each company and research project has been investigated in terms of requirements, process descriptions, system architecture and business related indicators. The purpose of this section is to provide an overview of the aggregated data. In the previous chapter it became clear that that certain aspects of the framework such as the *data storage model* or the *issuing* process manifest similarities among companies. Therefore, a data analysis is shown along with some insights investigating the whole sample.

6.3.1 Practitioners

Beginning with the requirements, it can be seen that all companies provide the possibility to *issue*, *verify* and *store* credentials. The only company that does not provide any information about issuance is CHESICC, which itself is an outlier in the set of companies due to its centralized nature. The rest of the requirements shows more diversification. Figure 6.3 demonstrates how each requirement has been met overall by companies. Where 83% of the companies allow retrieval of claims, only about 33% offer the possibility of revoking one. Even lower is the *assertion* requirement with only one company meeting it. Assertion is a complex mechanism that allows to share only certain claims and restricting the time how long it is shared. Currently, only BlockCerts covers this requirement. Different from all other companies, BlockCerts is a MIT spin-off and has its roots in an academic environment. It also incorporates an open-source business model and has no platform attached to it like Parchment or Gradbase. It has a unique position in the market due to the compliance to the VC draft. Its open-source business model meeting most of the requirements stated in the framework.
Regarding the other companies, most of them have a common approach to the domain of digital credentialing. There are two major trends that can be seen regarding the product: At the core of each company is either the concept of creating a CV or storing credentials in a PDF format attached with a link to the blockchain. Surrounding the core product there is often a layer of customization, e.g. creating corporate-identity compliant templates for credentials or layouting the CV. Furthermore, the business models are very similar. Most companies charge a subscription fee based on multilevel tiers. Most tiers unlock a set of features or increases a certain amount of issuances per month, as an example.

Regarding the process descriptions it can be said that most companies use the same issuing, revocation and verification mechanics. For issuing, the companies use either the method of batch-creating credentials via Excel spreadsheets or creating a single credential with a user interface. Then, the credential is hashed and stored on a blockchain - mostly Ethereum or Bitcoin ones. 72% of the mentioned companies use a permissionless system and deploy credentials onto public blockchains. Only 28% have a different approach where the access to the network is permissioned. Since each venture apart CHESICC uses the blockchain and a peer-to-peer trust model, verification relies on the built-in mechanism

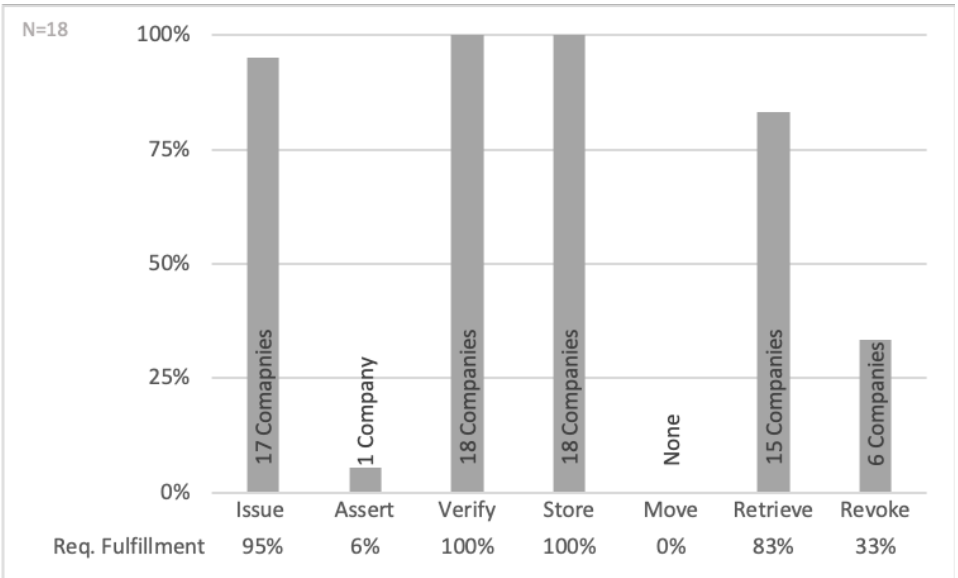


Figure 6.3: Requirements fulfillment aggregated over 18 investigated practitioners.

the blockchain architecture provides: comparing the hash of a document with the one on the blockchain. In practice, verification is either stated directly on the platform (e.g. Gradbase or APPII), or users have to manually upload the document to a verifier who hashes and compares the document stored on-chain (BlockCerts, Blockeducate). Lastly, revocation is based on either implementing the OpenBadges revocation method where retrieving a credential ends in the http 410 "gone" code or another transaction containing the credential ID, stating that the credential has been revoked.

Figure 6.4 shows three more data points that are interesting to look at. The first one is the GDPR compliance. By design, public blockchains are transparent, immutable and data cannot be deleted from them. This contradicts the General Data Protection Regulation issued by the European Union, specifically the "right to be forgotten". It states that data must be erased upon request by the user. The way how companies tackle this problem, and this seems to be a common approach according to what companies state, is by making the data undecipherable. Consequently, the key to decipher the data will be deleted and it cannot be read any longer. However, this does not delete the data itself as it is stated in the regulation. The question of how compliant this approach is to the GDPR is out of scope of this thesis but could be a topic for future work.
Next, the API availability is stated in the graph. It can be seen that 55,56% of the companies already have an API to connect to their platforms. Since the overall maturity of the companies is in the lower regions (see Figure 6.5), not all companies have developed an API for their systems. It can be assumed that this quote will increase since an API can be used for improved automation and integration to institutional systems.
Lastly, the graph shows how many companies target the educational sector with their

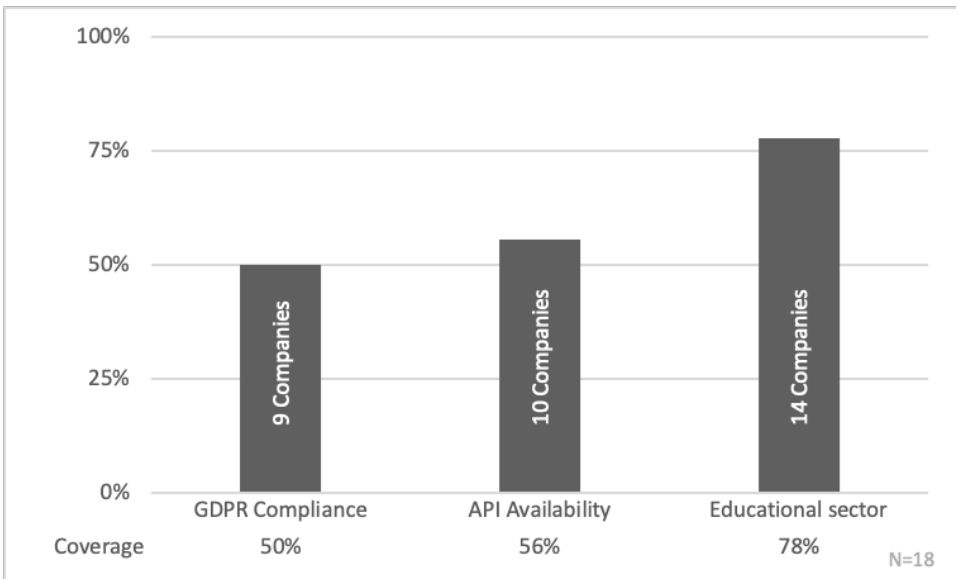


Figure 6.4: GDPR compliance, API availability and target industry set to educational sector on average based on 18 investigated practitioners.

product. According to this it can be assumed that a high demand is existing for the educational sector to digitize the way credentials are handled, issued and stored. This includes professional and institutional education. Both sides of the market can be covered with the credential management systems. Furthermore, some of the companies target different sectors and industries with the same product as well. Vottun is an example that deploys the same product both to the educational sector and to the supply chain market.

Lastly, Figure 6.5 shows the overall maturity distribution among the 18 investigated companies. It must be stated that this is only an assumption based on the framework data and what the companies provide in terms of information. The maturity measurement defined in Chapter 4 is based on facts about internal processes in the company and how these are defined, documented and executed. It was not possible to gain that insight during the conduction of this thesis. Yet, it can be stated that the overall maturity is low and the market is in its early stages. Taking business models, API availability and the similarities of the companies into account, it is possible that a market adjustment will occur once the companies reach a higher level of maturity. Another aspect that speaks for low maturity is the fulfillment of requirements and compliance to e.g. the W3C VC specification. The data shows that only five out of eighteen (28%) companies are compliant to the OpenBadges specification, which itself is not a specification for macro-credentials. It rather defines micro-credentials, whose market has been developing for approximately ten years now. The development of compliance to the W3C VC specification can be a topic for future work as well, especially meeting the requirements *move* and *assert*. Both of them are technically demanding and the former one has impact on the business model as well. Right now, no company fulfills the *move* requirement. One reason for this could be that companies do not

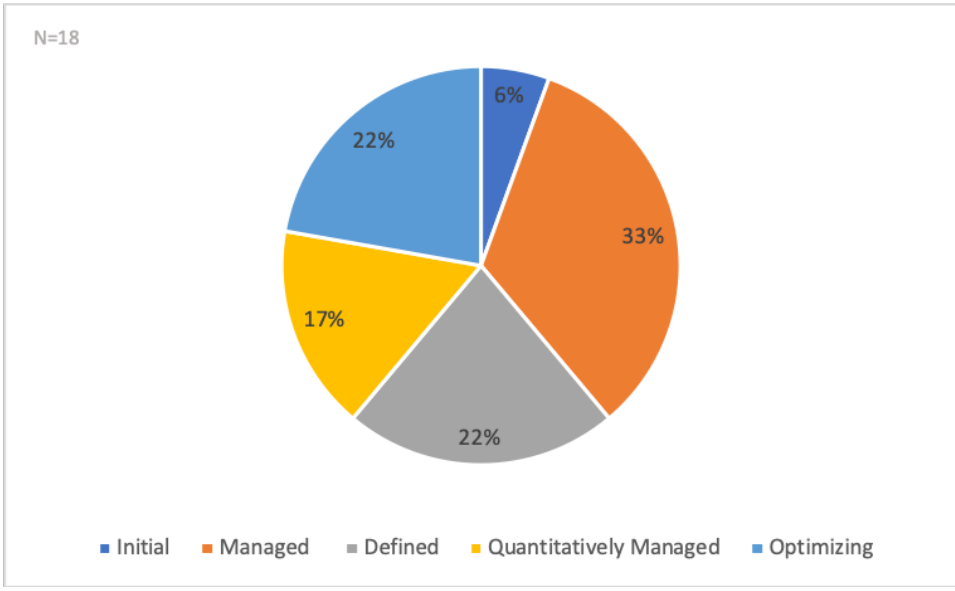


Figure 6.5: Distribution of maturity levels aggregated over 18 practitioners.

want customers to switch from one platform to another, impacting the companies' overall marketing and growth potential.

6.3.2 Research Projects

In terms of research projects, the framework data looks very different. Similar to the companies, a first look is taken at the requirements shown in Figure 6.6. Identical to the companies evaluation, the requirements *issue* (90% coverage), *store* (100%) and *verify* (100%) have been met by most projects along with *retrieve* (80% coverage). On the contrary, more research projects are also meeting the *move* (20% of research projects versus 0% of the companies) and *assert* (10% of research projects versus 5.56% of the companies) requirement. Especially the former one has not been fulfilled by any company investigated. Although the actual percentage of having met the *assert* requirement by research projects (10%) is higher compared to companies (5.56%), it can be said that fulfilling this requirement is not as important as others and at the same time a complex one. Only the blueprint project described in Subsection 6.2.5 deals with asserting claims. Within the process description section there is a large variation noticeable. Some projects define the processes relevant for the framework completely, some do not even mention one aspect of it. This is derived from the overall nature of the research papers. Many papers deal with a certain aspect of a credentialing system, especially the architectural part or the consensus algorithm (cf. [35], [82], [81]). Interestingly, none of the projects dealt with refreshing certificates and only one project described that the data model can be extended. In contrast to the companies where 72% used a permissionless blockchain model, re-

searchers are more interested in permissioned ones. 50% of the papers deal with permissioned systems and build their architecture or even consensus algorithms around this principle. Permissioned blockchains have the advantage that peers can only join by invitation, whereas public blockchains are open for everyone to join.

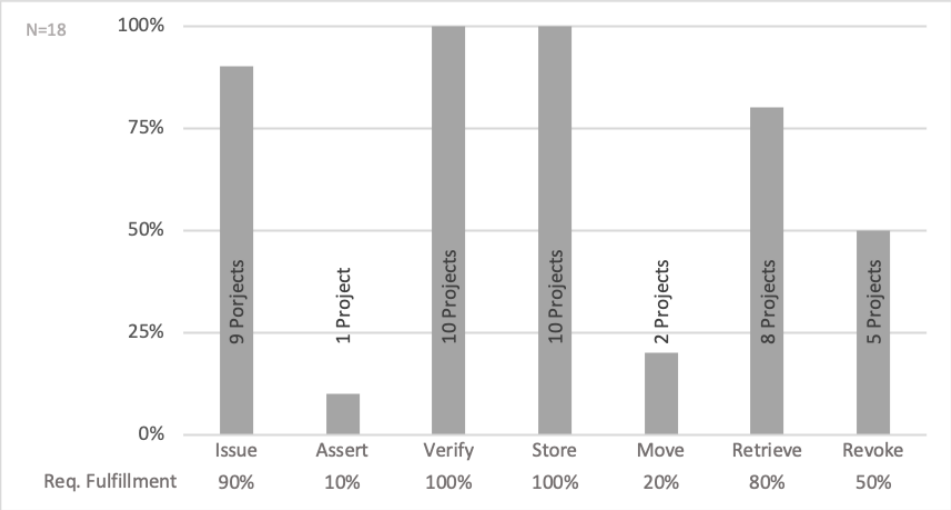


Figure 6.6: Requirements fulfillment aggregated over ten investigated research projects.

Figure 6.7 shows how many projects state compliance to the GDPR, have an available API and target the educational sector. Strikingly, only two out of ten (20%) have both an API and are compliant to the GDPR. Since most of the times the output of these papers is a proof-of-concept or a prototype, the GDPR compliance is not as important as it is for operating companies. The same holds for APIs. Both aspects become more relevant when a system becomes ready for a productive deployment. Yet, all research projects are in an early stage of development and have no enterprise structure behind them. Additionally, the graph shows that each project targets the educational sector, namely schools and universities. This is interesting because research is mostly conducted in an institutional environment. The data underlines the demand for credential management systems and that a market is emerging.

Similar to the companies, none of the research projects are compliant with the W3C VC draft. 30% are compliant to the OpenBadges implementation or refer to it in regards of data model, architecture or processes.

In conclusion it can be said that the overall market for digital credentialing is relatively immature and there is room for innovation. The research projects have shown a way how current approaches by companies can be modified or even new blockchain systems can arise. However, the practicability of the mentioned research projects has still to be proven.

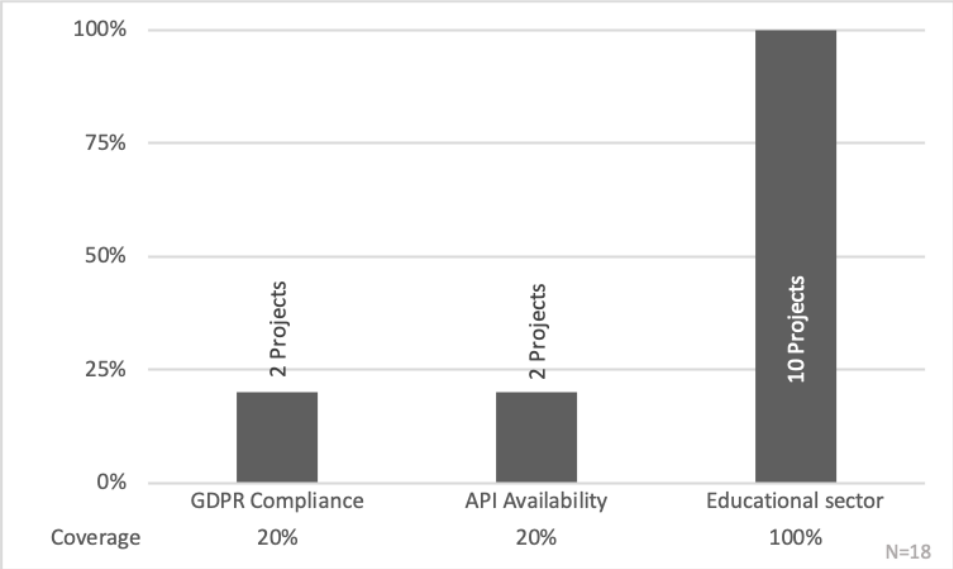


Figure 6.7: GDPR compliance, API availability and target industry set to educational sector on average based on ten investigated research projects.

7 Conclusion and Future Work

Before the master's thesis is concluded in this chapter, returning to the problem statement in Chapter 1 is necessary. In this chapter, the problem statement is provided along with its research questions. In the problem statement, the thesis states that "scanning and digitization alone is not enough to prove the authenticity of a document" and that there have to be mechanisms that provide a reliable proof of these characteristics. As the thesis has shown, several companies and researchers have already tackled this problem and created systems that deal with digital credentialing. Before the investigation of the mentioned companies could have started, an investigation of current standardization efforts was necessary. Therefore, the OpenBadges and W3C Verifiable Credentials draft have been examined and demonstrated. Both specifications are relevant for the industry of digital credentialing. Especially OpenBadges has a significant adoption rate, whereas the W3C VC draft is used only by BlockCerts. The reason for this can be derived from two aspects. Firstly, the market of digital credentialing has been focused on micro-credentials for the last ten years. Secondly, the Verifiable Credentials draft is comprised of technically complex principles such as evidences, zero-knowledge proofs and assertion of credentials. Compared to the OpenBadges specification, developers face a much greater challenge in creating a VC-compliant system than a OpenBadges-compliant one. Additionally, the Verifiable Credential draft itself is still in development. Following this thesis, a review of the current state of digital credentialing could be conducted in the future to investigate the adoption rate and implementation of the Verifiable Credential draft. Furthermore, investigating if and how the developers are connecting the Verifiable Credentials draft with the OpenBadges one would be interesting.

Comparing the companies has shown that a tendency to combine aspects of the verifiable credentials draft such as evidences with parts of the OpenBadges specification. Some companies rely on the hosted badge revocation process because it is simpler instead of providing a cryptographic proof.

For creating a comparison of several independent companies and research projects, a foundation had to be built. The approach in this thesis was to create a framework based on standards. Since the OpenBadges specification serves as a standard for micro-credentials and the Verifiable Credentials draft aims to standardize the macro-credential domain, the latter has been chosen to create the framework. Extracted from the data model and additional resources provided by the W3C, requirements and characteristics were shown. Since the requirements were already provided in an acceptable format, they were adopted for the framework. The processes could be derived form a larger set of characteristics that is also provided in the W3C VC draft. However, they could not be adopted in the same way the requirements have been. Consequently, the characteristics were clustered

and aggregated into processes. As a result, nine processes were formed and included in the framework. The requirements section has the purpose to see how much of the processes are implemented, whereas the section below describes the implementation of them. This way, a data could be accumulated to see how many companies are implementing the requirements on the one hand. On the other hand, the framework shows and describes how they are implemented. This dual approach enables to get an insight in a specific company, but also create an overview of the whole sample. As an example, it has been demonstrated that each company features the issuing process and implement it in a rather similar way. Apart from these two sections, the framework features system and business indicators that both are helpful to determine how the company is monetizing the system and also how the system operates. Especially the system section differs for each company. Where all companies use the blockchain as a data storage model, only few of them show the actual data model or offer an API for automated processing of the credentials.
When it comes to research projects, the framework has not been as applicable as it has been for practitioners. The chosen approach showed to be suitable for systems that are already existent. Furthermore, the system has to be completely implemented to generate the correct data. The main difference between the companies section and the research projects one in Chapter 6 is that most papers deal with parts of a system. Mostly, architectural parts of mechanisms such as the proof-of-work have been investigated by the researchers. For future work, two improvements can be proposed here:

1. Investigation of researcher projects should either be undertaken at a later point in time when the projects are in a prototype state.

2. The framework should be adapted so that it is suitable not only for whole systems, but also for parts of them.

During the investigation of research projects it became clear that most of them have no business section attached. Research is more focused on how to enable digital credentialing in an innovative way. Most of the systems do not show any business related indicators. The few that do state to be open-source. Although the framework is as good applying to research projects as it is applied to companies, it showed that research is focused on completely different aspects than the industry is. Especially the architectural parts and system design differ significantly from practitioners. As an example for that is the usage of permissioned networks instead of public ones. Here, future work could investigate if this is becoming a trend and how permissioned blockchains help to limit the amount of computational work that is required within such a network.
Future research could comprise the implementation of a prototype based either on the companies' commonalities or on aspects that the verifiable credentials draft features. Especially the assertion and evidence ones have still to be proven in terms of feasibility. Assuming that research topics are mostly open-source, implementing a repository system for credentials could also be interesting. None of the companies offers the option to move credentials from their system to another. As with most immature markets, only a few players will survive the early stages and become larger ones at the end. Therefore, a

repository that offers the ability for interaction with several other systems would provide a way for users to store their credentials safely and independently from companies.

In conclusion, the thesis has contributed to the topic of digital credentialing by investigating and comparing current standardization effort. Furthermore, an overview of the current state in the market and the research area has been given. The research has shown that the current market situation features a significant degree of similarity and that the Verifiable Credentials draft has not been adopted yet. However, the transformation of analogue credentialing into a fully digital process has started. As a first sector, institutional and professional education will benefit most from this. Additionally, the thesis has shown that tendencies to extend digital credentialing systems into different sectors can be beneficial as well. Especially with the final release of the Verifiable Credentials draft and further development of blockchain technology, digital credentialing has the potential to become a standard for industries and institutions equally.

List of Figures

5.1 Subject-Holder relationship in the W3C VC Draft [33]. 35
5.2 Trust model in the W3C VC Draft [33]. 37
5.3 eIDAS infrastructure involved in the authentication process [43] 48
5.4 DID with Ethereum method and namespace-specific identifier. Adopted from [48]. 50

6.1 Blockchain-based architecture with three participating peers: Provider node, individual node and miner. Adopted from [76]. 94
6.2 QualiChain architecture separated into several engines. Adopted from [85]. 111
6.3 Requirements fulfillment aggregated over 18 investigated practitioners. . . 114
6.4 GDPR compliance, API availability and target industry set to educational sector on average based on 18 investigated practitioners. 115
6.5 Distribution of maturity levels aggregated over 18 practitioners. 116
6.6 Requirements fulfillment aggregated over ten investigated research projects.117
6.7 GDPR compliance, API availability and target industry set to educational sector on average based on ten investigated research projects. 118

List of Tables

4.1 Desirable characteristics a credentialing system should feature. Adopted and cited from [33]. The items are assigned to categories and numbered by the author of the thesis. . . . 20
4.2 Framework for comparison of existing digital credentialing solutions. . . . 25
4.3 CMMI Levels of maturity taken from [38]. The headlines for each level have been adapted to the CMMI definition. In [38], the author uses different headlines. . . . 27

6.1 Framework data for Accredible. . . . 54
6.2 Framework data for APPII. . . . 56
6.3 Framework data for BCDiploma. . . . 58
6.4 Framework data for BlockCo. . . . 60
6.5 Framework data for BlockCerts. . . . 62
6.6 Framework data for Blockeducate. . . . 64
6.7 Framework data for CHESICC. . . . 66
6.8 Framework data for Credly. . . . 68
6.9 Framework data for CV Trust. . . . 70
6.10 Framework data for Edgecoin. . . . 72
6.11 Framework data for Gradbase. . . . 74
6.12 Framework data for Keeex. . . . 76
6.13 Framework data for Parchment. . . . 78
6.14 Framework data for SAP TrueRec . . . 80
6.15 Framework data for Sony Global Education platform. . . . 82
6.16 Framework data for Sproof. . . . 84
6.17 Framework data for Stampery. . . . 87
6.18 Framework data for Vottun. . . . 89
6.19 Framework Data for *Blockchain and Smart Contracts for Digital Certificates*. 91
6.20 Framework data for Blockchain Education Platform. . . . 93
6.21 Framework data for *Blockchain-Based Education Records*. . . . 96
6.22 Framework data for BcER2. . . . 98
6.23 Framework data for *Blueprint for learning trace repositories*. . . . 101
6.24 Framework data for CVSS . . . 103
6.25 Framework data for Credence Ledger . . . 105
6.26 Framework data for *Distributed Credit Transfer*. . . . 107
6.27 Framework data for ECBC. . . . 109

6.28 Framework data for QualiChain. 112

Acronyms

ACA Authorization Certificate Authority. 48

AI Artificial Intelligence. 13

API Application Programming Interface. 26

BcER2 Blockchain-based Educational Record Repository. 97

BEP Blockchain Education Platform. 92

BSI Buundesamt für Sicherheit in der Informationstechnik. 46, 47

BTA Blockchain Timestamp Architecture. 86

BTC Bitcoin. 8, 60, 86

CHESICC Chinese Higher Education Student Information and Career Center. 65, 77, 78, 113

CHSI China Higher Education Student Information. 66

CMMI Capability Maturity Model Integration. 27, 123

CSCA Country Signing Certification Authority. 48

CSV Comma-separated Value. 53

CV Curriculum Vitae. 55

CVCA Country Verifying Certification Authority. 48

CVSS Certificate Verifying Support System. 102

DApps Decentralized Applicatons. 9, 63

DID Decentralized Identifier. 29

DIN Deutsches Institut für Normung. 21

DNS Domain Name System. 84

DS Document Signer. 48

DV Document Verifier. 48

ECBC Educational Certificate Blockchain. 108

eIDAS Electronic Identification, Authentication and trust Services. 6, 28, 46, 47

ETH Ethereum. 8, 58, 86

EU European Union. 6

GDPR General Data Protection Regulation. 24

GED Sony Global Education. 81

GW Gigawatts. 12

HCMC Ho Chi Minh City. 102, 103

HTML Hypertext Markup Language. 66

HTTP Hypertext Transfer Protocol. 9

HTTPS Hypertext Transfer Protocol Secure. 43

IPFS InterPlanetary File System. 71, 92

ISO International Organization for Standardization. 21

JPEG Joint Photographic Experts Group File Interchange Format. 75

JSON JavaScript Object Notation. 13

JWS JSON Web Signature. 45

KPI Key Performance Indicators. 26

KYC Know Your Customer. 102

LMS Learning Management System. 67

LMU Ludwig-Maximilians-Universität München. 44

MIT Massachusetts Institute for Technology. 61

MOOC Massive Open Online Course. 1, 7, 14

MPT Merkle-Patricia Tree. 108

nonce Number Only Used Once. 8

OAth 2 Open Authenticate Version 2. 67

OBI OpenBadges Infrastructure. 7, 67

OU UK Open University of the United Kingdom. 110

PDF Portable Document Format. 13

PKI Public Key Infrastructure. 46

QR Quick Response. 90, 102

RQ Research Question. 2

RSA Rivest–Shamir–Adleman Cryptosystem. 30, 50

TUM Technical University of Munich. 11

TX Transaction. 60

UCL University College London. 73

URI Unified Resource Identifier. 28, 29

UUID Universalliy Unique Identifier. 37

VC Verifiable Credential. 16

W3C World Wide Web Consortium. 16

XML Extensible Markup Language. 13

Bibliography

[1] M. Jirgensons and J. Kapenieks. “Blockchain and the Future of Digital Learning Credential Assessment and Management”. In: *Journal of Teacher Education for Sustainability* 20.1 (2018), pp. 145–156. ISSN: 16915534. DOI: 10.2478/jtes-2018-0009.

[2] Bundeskriminalamt. “Polizeiliche Kriminalstatistik 2018”. In: (2018), pp. 1–109. ISSN: 1866-4911.

[3] S. Salzborn. *Klassiker der Sozialwissenschaften*. Second edi. Springer Verlag, 2014. ISBN: 9783658132125. DOI: 10.1007/978-3-658-13213-2. URL: http://dnb.d-nb.de.

[4] Z. P. Jin, J. Xu, M. Xu, and N. Zheng. “An attribute-oriented model for identity management”. In: *2010 International Conference on e-Education, e-Business, e-Management and e-Learning*. IEEE. 2010, pp. 440–444.

[5] L. J. Camp. “Digital identity”. In: *IEEE Technology and Society Magazine* 23.3 (2004), pp. 34–41. ISSN: 02780097. DOI: 10.1109/MTAS.2004.1337889.

[6] Official Journal of the European Union. *REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL*. 2014. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014R0910&from=DE#d1e1233-73-1.

[7] Merriam-Webster. *Credentials | Definition of Credentials by Merriam-Webster*. URL: https://www.merriam-webster.com/dictionary/credentials.

[8] A. Herzberg and Y. Mass. “Relying party credentials framework”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2020 (2001), pp. 328–343. ISSN: 16113349.

[9] B. Oliver. “Better 21C Credentials: Evaluating the Promise, Perils and Disruptive Potential of Digital Credentials”. In: January 2016 (2016). URL: http://www.assuringgraduatecapabilities.com/21c-credentials-olt-project.html.

[10] Bologna Follow-Up Group. “Ects Users ’ Guide 2015”. In: January (2015), p. 108. DOI: 10.2766/87592. URL: http://ec.europa.eu/education/library/publications/2015/ects-users-guide_en.pdf.

[11] L.-H. Liang. *Feeling depressed about your 2:2 degree? Get over it, employers have | Guardian Careers | The Guardian*. 2015. URL: https://www.theguardian.com/careers/2015/sep/01/graduate-jobs-employers-degree-result.

[12] P. A. Lemoine and M. D. Richardson. “Micro-Credentials, Nano Degrees, and Digital Badges”. In: *International Journal of Technology and Educational Marketing* 5.1 (2015), pp. 36–49. ISSN: 2155-5605. DOI: 10.4018/ijtem.2015010104.

[13] U. Gallersdörfer, P. Holl, and F. Matthes. *Blockchain-based Systems Engineering – Lecture Slides*. Oct. 2019. URL: https://github.com/sebischair/bbse.

[14] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Tech. rep. 2019, p. 9. DOI: 10.2139/ssrn.3440802. URL: www.bitcoin.org.

[15] Coinbase. *Bitcoin Price Chart (BTC) | Coinbase*. URL: https://www.coinbase.com/price/bitcoin?locale=en.

[16] Coinbase. *Ethereum Price Chart (ETH) | Coinbase*. URL: https://www.coinbase.com/price/ethereum?locale=en.

[17] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer. “Decentralization in Bitcoin and Ethereum Networks”. In: ().

[18] V. Brühl. “Bitcoins, Blockchain und Distributed Ledgers: Funktionsweise, Marktentwicklungen und Zukunftsperspektiven”. In: *Wirtschaftsdienst* 97.2 (2017), pp. 135–142. ISSN: 1613978X. DOI: 10.1007/s10273-017-2096-3.

[19] V. Buterin. “A next-generation smart contract and decentralized application platform”. In: *Etherum* January (2014), pp. 1–36. URL: https://github.com/ethereum/wiki/wiki/White-Paper.

[20] A. Gall. *Building your first Ethereum Oracle - Tales From the Crypto - Medium*. URL: https://medium.com/decentlabs/building-your-first-ethereum-oracle-1ab4cccf0b31.

[21] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani. “Blockchain-based applications in education: A systematic review”. In: *Applied Sciences (Switzerland)* 9.12 (2019). ISSN: 20763417. DOI: 10.3390/app9122400.

[22] D. Efanov and P. Roschin. “The all-pervasiveness of the blockchain technology”. In: *Procedia Computer Science*. Vol. 123. Elsevier B.V., Jan. 2018, pp. 116–121. DOI: 10.1016/j.procs.2018.01.019.

[23] A. de Vries. “Bitcoin’s Growing Energy Problem”. In: *Joule* 2.5 (2018), pp. 801–805. ISSN: 25424351. DOI: 10.1016/j.joule.2018.04.016. URL: https://doi.org/10.1016/j.joule.2018.04.016.

[24] M. Mueller and M. Rumph. *How Much Power is 1 Gigawatt?* 2019.

[25] G. Chen, B. Xu, M. Lu, and N.-S. Chen. “Exploring blockchain technology and its potential applications for education”. In: *Smart Learning Environments* 5.1 (2018), pp. 1–10. DOI: 10.1186/s40561-017-0050-x.

[26] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz. “Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities”. In: *IEEE Access* 7 (2019), pp. 85727–85745. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2925010.

[27] M. Sharples and J. Domingue. “The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward”. In: *Adaptive and Adaptable Learning. EC-TEL 2016. Lecture Notes in Computer Science, vol 9891. Springer, Cham* 2 (2016), pp. 490–496. DOI: 10.1007/978-3-319-45153-4.

[28] B. Chakroun and J. Keevy. “Digital Credentialing”. In: (2018), p. 43.

[29] Hasso Plattner Institut. *Overview*. URL: https://hpi.de/en/open-campus/overview.html.

[30] D. Gibson, N. Ostashewski, K. Flintoff, S. Grant, and E. Knight. “Digital badges in education”. In: *Education and Information Technologies* 20.2 (2015), pp. 403–410. ISSN: 15737608. DOI: 10.1007/s10639-013-9291-7.

[31] Google. *Add to Google Maps and earn badges - Google Maps Help*. URL: https://support.google.com/maps/answer/9197204?hl=en.

[32] L. Muilenburg and Z. Berge. “Digital badges in eduction”. In: *Education and Information Technologies* 20.2 (2013), pp. 403–410. DOI: 10.1007/s10639-013-9291-7.

[33] M. Sporny, D. Longley, and D. Chadwick. *Verifiable Credentials Data Model 1.0*. Tech. rep. W3C, 2019, pp. 1–115. URL: https://w3c.github.io/vc-data-model/%20https://www.w3.org/TR/vc-data-model/.

[34] S. Otto, S. Lee, B. Sletten, D. Burnett, M. Sporny, and K. Ebert. *Verifiable Credentials Use Cases*. Tech. rep. W3C, 2019. URL: https://www.w3.org/TR/vc-use-cases/.

[35] J. C. Farah, A. Vozniuk, M. J. Rodriguez-Triana, and D. Gillet. “A blueprint for a blockchain-based architecture to power a distributed network of tamper-evident learning trace repositories”. In: *Proceedings - IEEE 18th International Conference on Advanced Learning Technologies, ICALT 2018*. Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 218–222. ISBN: 9781538660492. DOI: 10.1109/ICALT.2018.00059.

[36] D. Tapscott and A. Tapscott. *The Blockchain Revolution and Higher Education*. Tech. rep. 2017. URL: http://www.blockcerts.org/.

[37] SEI. *CMMI® for Development, Version 1.3 CMMI-DEV, V1.3 - Improving processes for developing better products and services*. Tech. rep. 2010, p. 482. URL: http://www.sei.cmu.edu.

[38] H. Wang, K. Chen, and D. Xu. “A maturity model for blockchain adoption”. In: *Financial Innovation* 2.1 (2016). ISSN: 21994730. DOI: 10.1186/s40854-016-0031-z. URL: http://dx.doi.org/10.1186/s40854-016-0031-z.

[39] Government Technology Agency. *Documentations for opencerts*. 2020. URL: https://docs.opencerts.io/.

[40] *Open Attestation: Meta framework for notary service on the blockchain.* 2020. URL: https://github.com/Open-Attestation/open-attestation.

[41] C. Ruijie and R. Yeh. *Decentralized Issuer's Identity Proof using DNS-TXT*. 2019. URL: https://github.com/Open-Attestation/adr/blob/master/decentralized_identity_proof_DNS-TXT.md.

[42] J. Bohrer, T. F. Cook, M. Esquela, S. Gance, J. Goodell, M. Gylling, V. Haag, A. Hripak, K. Lemoie, M. Leuba, R. Macdonald, N. Otto, J. Pitcher, S. Ravet, A. Reis, J. Schmidt, and A. Szabo-Nagy. *Open Badges v2.0*. 2018. URL: https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html.

[43] F. Office for Information Security. *Overview of the German eID system*. Tech. rep. 2017. URL: https://www.bsi.bund.de.

[44] ICAO. *Country Signing Certification Authority (CSCA)*. URL: https://www.icao.int/Security/FAL/PKD/BVRT/Pages/CSCA.aspx.

[45] *BSI - Country Verifying Certificate Authority*. URL: https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/ElektronischeIdentitaeten/sicherPKI/sicherCVCA/cvca_node.html.

[46] E-Estonia. *e-Identity — e-Estonia*. 2018. URL: https://e-estonia.com/solutions/e-identity/.

[47] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, and M. Sabadello. *Decentralized Identifiers (DIDs) v1.0*. 2019. URL: https://www.w3.org/TR/did-core/.

[48] M. Schäffner. "Analysis and Evaluation of Blockchain-based Self-Sovereign Identity Systems". PhD thesis. Technical University of Munich, 2020, p. 126.

[49] Accredible. *Accredible Credential API · Apiary*. URL: https://accrediblecredentialapi.docs.apiary.io/#.

[50] APPII. *World's first blockchain career verification platform | APPII*. 2018. URL: https://appii.io/.

[51] BCDiploma. *BCDiploma White Paper v. 2.2*. Tech. rep. BCDiploma, 2018. URL: https://www.evidenz.io/img/pdf/BCD-WhitePaper_last.pdf.

[52] BCDiploma. *EvidenZ, the ultimate certification technology on the blockchain*. URL: https://www.evidenz.io/framework.html.

[53] Blockco. *Block.co*. URL: https://block.co/.

[54] Blockcerts. *Introduction - Blockcerts : The Open Standard for Blockchain Credentials*. URL: https://www.blockcerts.org/%20https://www.blockcerts.org/guide/.

[55] BlockCerts. *Blockcerts · GitHub*. URL: https://github.com/blockchain-certificates.

[56] Blockeducate. *Blockchain For Education, Blockchain Academic Certificate*. URL: https://blockeducate.com/services/blockchain-for-education/.

[57] CHISECC. *Brief Introduction to Online Verification Report_CHESICC*. URL: https://www.chsi.com.cn/xlcx/en/brief.jsp.

[58] Credly. *How Credly Works*. URL: https://info.credly.com/how-credly-works.

[59] CVTrust. *Smart Certificate for the education world*. URL: https://www.cvtrust.com/default.aspx.

[60] Edgecoin. *Edgecoin.io | Fraud-proof, Smart Education on the Blockchain*. URL: https://www.edgecoin.io/.

[61] Gradbase. *Gradbase - Instantly Verify Qualifications*. URL: https://gradba.se/en/%20https://www.gradba.se/en/.

[62] Keeex. *Solutions - KeeeX - the Universal Probative Value*. URL: https://keeex.me/solutions/.

[63] SAP. *TrueRec: digitale Brieftasche mittels Blockchain*. URL: https://news.sap.com/germany/2017/09/truerec-blockchain/.

[64] C. Guiterrez and A. Khzhiniak. *SAP Verifies Academic Credentials Using Blockchain and Cloud Foundry | Altoros*. 2017. URL: https://www.altoros.com/blog/sap-stores-academic-credentials-using-blockchain-and-cloud-foundry/.

[65] Hyperledger Fabric. *Introduction — hyperledger-fabricdocs master documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html#hyperledger-fabric.

[66] Sony Global Education. *SGE Education Blockchain*. URL: https://blockchain.sonyged.com/.

[67] C. Brunner, F. Knirsch, and D. Engel. *SPROOF: A platform for issuing and verifying documents in a public blockchain*. Tech. rep. URL: https://digitalcurrency.unic.ac.cy/free-introductory-.

[68] Sproof. *Schema — sproof 1.0 documentation*. URL: https://sproof-docs.readthedocs.io/en/latest/schema.html#document.

[69] Sproof. *sproof · GitHub*. URL: https://github.com/sproof.

[70] A. Sánchez De Pedro, C. Stampery, L. Ivan, and C. García. *Stampery Blockchain Timestamping Architecture (BTA)*. Tech. rep. 2016.

[71] Stampery. *Stampery Features | Stampery*. URL: https://stampery.com/features/#existence.

[72] Vottun. *Credentials - Vottun*. URL: https://vottun.com/services/digital-credentials/.

[73] Vottun. *Vottun Protocol - Vottun*. URL: https://vottun.com/vottun-protocol/.

[74] J.-C. Cheng, N.-Y. Lee, C. Chi, and Y.-H. Chen. "Blockchain and smart contract for digital certificate". In: *2018 IEEE international conference on applied system invention (ICASI)*. IEEE. 2018, pp. 1046–1051.

[75] W. Gräther, S. Kolvenbach, R. Ruland, J. Schütte, C. F. Torres, and F. Wendland. "Blockchain for Education: Lifelong Learning Passport". In: *Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies*. 10. European Society for Socially Embedded Technologies (EUSSET). 2018, pp. 1–8. DOI: 10.18420/blockchain2018. URL: https://dl.eusset.eu/bitstream/20.500.12015/3157/1/blockchain2018_10.pdf%0Ahttps://www.fim-rc.de/Paperbibliothek/Veroeffentlicht/756/wi-756.pdf.

[76] M. Han, D. Wu, Z. Li, Y. Xie, J. S. He, and A. Baba. "A novel blockchain-based education records verification solution". In: *SIGITE 2018 - Proceedings of the 19th Annual SIG Conference on Information Technology Education*. Vol. 18. ACM, 2018, pp. 178–183. ISBN: 9781450359542. DOI: 10.1145/3241815.3241870. URL: https://doi.org/10.1145/3241815.3241870.

[77] E. Bessa and J. Martins. *A Blockchain-based Educational Record Repository To cite this version : HAL Id : hal-02085749 A Blockchain-based Educational Record Repository*. Tech. rep. 2019, pp. 1–11. URL: https://hal.archives-ouvertes.fr/hal-02085749.

[78] D. H. Nguyen, D. N. Nguyen-Duc, N. Huynh-Tuong, and H. A. Pham. "CVSS: A blockchainized certificate verifying support system". In: *ACM International Conference Proceeding Series*. 2018, pp. 436–442. ISBN: 9781450365390. DOI: 10.1145/3287921.3287968. URL: https://doi.org/10.1145/3287921..

[79] MultiChain. *MultiChain data streams | MultiChain*. URL: https://www.multichain.com/developers/data-streams/.

[80] R. Arenas and P. Fernandez. "CredenceLedger: A Permissioned Blockchain for Verifiable Academic Credentials". In: *2018 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., Aug. 2018. ISBN: 9781538614693. DOI: 10.1109/ICE.2018.8436324.

[81] A. Srivastava, P. Bhattacharya, A. Singh, A. Mathur, O. Prakash, and R. Pradhan. "A Distributed Credit Transfer Educational Framework based on Blockchain". In: *Proceedings - 2018 2nd International Conference on Advances in Computing, Control and Communication Technology, IAC3T 2018*. Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 54–59. ISBN: 9781538641460. DOI: 10.1109/IAC3T.2018.8674023.

[82] Y. Xu, S. Zhao, L. Kong, Y. Zheng, S. Zhang, and Q. Li. "ECBC: A high performance educational certificate blockchain with efficient query". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10580 LNCS. Springer Verlag, 2017, pp. 288–304. ISBN: 9783319677286. DOI: 10.1007/978-3-319-67729-3{_}17.

[83] QualiChain. *QualiChain*. URL: https://qualichain-project.eu/.

[84] KBZ. *D2.2 QualiChain Stakeholders' Requirements and Use Cases*. Tech. rep. 2019.

[85] N. Chowdhury, A. Third, A. Mehrbod, V. Karakolis, C. Kontzinos, C. Botsikas, S. Scerri, I. Keck, N. Politou, and Miguel Correia. *D5.1 QualiChain Integrated Architecture*. Tech. rep. 2019.