MASTERARBEIT

SELBSTBESTIMMTE IDENTITÄT DURCH BLOCKCHAIN UND NON-FUNGIBLE-TOKEN

ausgeführt am

Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Peter Murer

Personenkennzeichen: 2010320032

Graz, am 21. Juni 2022 .....

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

Mit diesen Zeilen möchte ich mich bei all jenen Personen – welche mich im Rahmen dieser Arbeit unterstützt haben – bedanken.

Ein besonderer Dank gilt meinem Betreuer, Herrn Dr. Christian Hofer, Bakk., BSc, MSc, welcher mir in den letzten Monaten jederzeit mit konstruktivem Feedback, hilfreichen Tipps und fachlichem Wissen zur Seite gestanden ist.

Des Weiteren möchte ich mich bei Herrn Mag. Max Waltl für das Korrekturlesen dieser Arbeit recht herzlich bedanken.

Auch gilt ein großer Dank meiner Familie, welche im Laufe meines Studiums immer ein offenes Ohr für mich hatte und mich auch in schwierigen Phasen stets mit motivierenden Worten unterstützt hat.

Vielen Dank!

KURZFASSUNG

Aufgrund der fortschreitenden Digitalisierung werden mehr und mehr physische Produkte in der digitalen Welt abgebildet. Vor allem in Hinsicht auf digitale, identitätsbezogene Bescheinigungen ist es notwendig, dass diese nicht betrügerisch genutzt werden können, denn die jährliche Anzahl an Identitätsdiebstählen ist enorm. Hierfür braucht es eine Möglichkeit, diese digitalen Bescheinigungen so abzubilden, dass sie einerseits als fälschungssicher angesehen werden können und andererseits ein Identitätsdiebstahl unmissverständlich auffallen würde.

Ziel dieser Arbeit war es, die gestellte Forschungsfrage „Welche Herausforderungen birgt die Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen?“ zu beantworten. Um diese Forschungsfrage beantworten zu können, wurde neben der Erarbeitung von theoretischen Grundlagen ein Prototyp geschaffen, welcher als Grundlage für die Ermittlung der möglichen Herausforderung dient. Dabei wurde einerseits ein Smart-Contract entwickelt, welcher die notwendigen Funktionalitäten auf Blockchain-Basis zur Verfügung stellt und andererseits wurde eine Webanwendung kreiert, welche die Schnittstelle zwischen Smart-Contract und Anwender/Anwenderin darstellt.

Anhand der erhaltenen Ergebnisse des Prototyps konnten schließlich Herausforderungen identifiziert und mögliche Lösungsvarianten ermittelt werden. Als Hürden gelten dabei hauptsächlich Limitierungen im Sinne der Möglichkeiten und Struktur eines Smart-Contracts, welche der Definition einer Selbstbestimmtheit entsprechen und der Identifikation von seriösen Ausstellern der Nachweise. Sowohl die Aktualisierung des Gültigkeitsstatus anhand des Gültigkeitsdatums eines Nachweises als auch die Sicherstellung des Datenschutzes in Hinsicht auf sensible Daten stellen weitere Herausforderungen dar, welche es zu lösen gilt.

Die empfohlenen Lösungsvarianten dienen als Anhaltspunkt für weitere Forschungsarbeiten. Hierfür wurden abschließend offene Fragestellungen definiert, welche für dieses Vorhaben herangezogen werden können.

ABSTRACT

Due to increasing digitization, more and more physical products are being transferred into the digital world. Regarding digital, identity-related certificates, it is necessary that they cannot be used fraudulently because the annual number of identity thefts is enormous. Therefore, a method is needed to represent these digital certificates such that they can be considered forgery-proof and identity theft would be obvious.

The goal of this paper was to answer the research question, "What are the challenges of implementing a blockchain application to verify identity-related certificates?" To answer this research question, in addition to the development of theoretical foundations, a prototype was created, which provides the basis for determining the potential challenges. Thus, a smart contract was developed that provides the necessary functionalities on a blockchain basis, and a web application was created that represents the interface between the smart contract and the user.

Based on the results obtained from the prototype, it was possible to identify challenges and possible solutions. Challenges are mainly limitations in the sense of the possibilities and the structure of a smart contract, which correspond to the definition of self-determination and the identification of reputable issuers. Updating the validity status based on the validity date of a certificate as well as the safeguarding of data protection with regard to sensitive data represent further challenges that need to be solved.

The recommended solution variants serve as a starting point for further research work. Finally, open questions were defined that can be used for this purpose.

INHALTSVERZEICHNIS

1 EINLEITUNG

Der Begriff „Blockchain“ hat sich in den letzten Jahren zunehmend in der breiten Öffentlichkeit etabliert, wobei sich diese Technologie aufgrund ihrer Eigenschaften in Hinsicht auf Transparenz und Integrität von unterschiedlichsten Transaktionen auch vermehrt in Projekten von bereits bestehenden Unternehmen wiederfindet. Vor allem auch aufgrund des enorm schwankenden Preis-Kurses der momentan bekanntesten Blockchain-Anwendung „Bitcoin“ findet sich diese Technologie immer öfter in diversen Medien wieder und wird zudem von der breiten Masse nicht selten als reines Spekulationsobjekt angesehen. Doch hinter diesen Technologien steckt weit mehr als Bitcoin und die damit verbundenen Wetten auf steigende oder fallende Kurse, denn der wahre Kern der Blockchain besteht darin, dass es sich bei dieser Technologie um ein fälschungssicheres und unveränderliches Transaktionsregister handelt (Lewin, Dogan, Schwarz & Fay, 2019). Unter der Verwendung dieses unveränderlichen Transaktionsregisters bietet sich auch eine – in dieser Arbeit behandelte – Umsetzung zur fälschungssicheren und dezentralen Abbildung von identitätsbezogenen Nachweisen an, sodass jede Person die Originalität jener Nachweise belegen kann, die ihr von Behörden oder Unternehmen ausgestellt wurden und für sich selbst bestimmen kann, an wen diese Bescheinigungen weitergegeben werden.

1.1 Ausgangssituation

Bereits im Jahr 2008 wurde die Blockchain von – bis zum heutigen Zeitpunkt – unbekannten Entwicklern/Entwicklerinnen mit dem Pseudonym „Satoshi Nakamoto“ durch die Veröffentlichung eines White-Papers mit dem Titel „Bitcoin: A Peer-to-Peer Electronic Cash System“ (Nakamoto, 2008) ins Leben gerufen. In der Anfangsphase eher unbekannt, herrscht in den letzten Jahren ein wahrer Wettlauf um die innovativsten Anwendungen mit der Blockchain als technologische Grundlage. Neben der im Jahr 2008 entstandenen Idee für das dezentrale Zahlungssystem Bitcoin, gibt es heute bereits über 7.500 (Statista, 2021b) weitere, auf der Blockchain basierende Anwendungen, welche sich diese Technologie zu Nutze machen, um sich in der breiten Masse zu etablieren.

Der Begriff „Non-Fungible-Token“(NFT) hat im Jahr 2021 in der Blockchain-Community bereits enorm an Interesse gewonnen. NFT sind digitale, einzigartige Token, welche heutzutage üblicherweise für die transparente Darstellung vom Besitz eines digitalen Gutes herangezogen werden und unter anderem die Eigenschaft der Blockchain für sich nutzen, um diesen Besitz fälschungssicher verifizieren zu können (Chevet, 2018). Beispielsweise kann ein NFT kreiert und mit einem selbst erstellten Foto verknüpft werden, sodass der Besitzer/die Besitzerin anhand des fälschungssicheren Tokens seinen/ihren Besitz des Fotos zweifellos beweisen kann. NFT ermöglichen den Nutzern/Nutzerinnen den eindeutigen Nachweis, wer welches digitale Gut zu welchem Zeitpunkt besessen hat. Aufgrund der Einzigartigkeit eines jeden Token werden die

damit verknüpften, digitalen Güter größtenteils als Sammelstücke angesehen in der Hoffnung, dass diese Sammelstücke eines Tages an Wert zunehmen werden. Beispielsweise hat der Verkauf eines digitalen NFT-Kunstwerks mit dem Namen „Everydays: The First 5000 Days“ des Künstlers „Beeple“ – um einen Preis von 69 Mio. USD – zu einem wahren Hype rund um diese Blockchain-Anwendung geführt (New York Times Company, 2021).

Aufgrund der Tatsache, dass sowohl NFT als auch die Identität von Menschen einzigartig sind, könnten die jeweiligen Identitätsmerkmale mittels verknüpftem NFT fälschungssicher in der Blockchain abgebildet werden. Durch die Möglichkeit zur Einsicht der kompletten Historie eines jeden NFT wäre man in der Lage, originale Identitätsnachweise von Fälschungen abzugrenzen. Durch die eindeutige Verifizierung einer Person und mittels weiterer NFT wäre es zusätzlich möglich, Nachweise von Behörden/Unternehmen mit der jeweiligen Person zu verknüpfen und dadurch ein digitales, fälschungssicheres Abbild der eigenen Identität zu schaffen.

1.2 Zielsetzung und Hypothesen

Ziel dieser Arbeit ist es, die Forschungsfrage „Welche Herausforderungen birgt die Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen?“ zu beantworten. Diese Arbeit setzt den Fokus auf ein weiteres Anwendungsgebiet für NFT, welches nicht (wie bisher) auf den Handel von Besitztümern ausgerichtet ist, sondern auf die Sicherstellung der eigenen, digitalen Identität und der damit in Zusammenhang stehenden Reduzierung der digitalen Identitätsdiebstähle. Um die definierte Forschungsfrage beantworten zu können, stützt sich diese Arbeit neben wissenschaftlichen Artikeln und Fachliteratur auf das Kreieren eines Prototyps, welcher schlussendlich die wesentlichen Herausforderungen aufzeigt, die sich bei der Integration der Blockchain zum Nachweis von identitätsbezogenen Bescheinigungen mittels NFT zeigen. Auch soll in dieser Arbeit eine Evaluierung dahingehend stattfinden, ob und in welchem Ausmaß die für den Prototyp definierten Wertekriterien Leistungsstandards erfüllen, welche für den Nachweis von identitätsbezogenen Bescheinigungen notwendig sind.

Anhand der definierten Forschungsfrage wurden im Vorhinein zwei wesentliche Hypothesen getroffen, welche im weiteren Verlauf dieser Arbeit bestätigt oder widerlegt werden:

Hypothese 1:

Eine Anwendung zur Sicherstellung einer selbstbestimmten, digitalen Identität soll gewährleisten, dass jede Person für sich selbst bestimmen kann, an wen ihre Daten weitergegeben werden. Es soll also für dritte Parteien nicht möglich sein, identitätsbezogene Nachweise anderer einsehen zu können, ohne das Einverständnis jener Person, welcher diese Nachweise zugeordnet sind. In diesem Zusammenhang definiert sich die erste Hypothese wie folgt: Die Einhaltung der Selbstbestimmung über die eigenen Daten stellt eine Herausforderung dar, weil nicht gewährleistet werden kann, dass die Daten von dritten Parteien direkt aus der Blockchain ausgelesen werden können.

Hypothese 2:

Als weitere Herausforderung könnte sich die Unterscheidung zwischen seriösen Behörden/Unternehmen und unseriösen Ausstellern behördlicher Nachweise und/oder Identitäten darstellen. Eine typische, dezentrale Blockchain zeichnet sich dadurch aus, dass sich sämtliche Teilnehmer/Teilnehmerinnen in diesem System lediglich durch ihr Schlüsselpaar, bestehend aus privatem und öffentlichem Schlüssel, unterscheiden und sich der Besitz eines einzigartigen, privaten Schlüssels durch eine sogenannte „digitale Signatur“ beweisen lässt (siehe Kapitel 2.1.1). Einzig der Beweis, dass man im Besitz eines bestimmten privaten Schlüssels ist, verifiziert jedoch nicht, dass es sich dabei auch um eine seriöse Behörde/ein seriöses Unternehmen handelt, denn eine Verifikation der Seriosität eines Teilnehmers/einer Teilnehmerin dieses Netzwerkes wird bis dato in den Grundzügen der gängigen Blockchains nicht berücksichtigt. Die zweite Hypothese wird demnach folgendermaßen definiert: Die Abgrenzung zwischen seriösen und unseriösen Parteien ist innerhalb einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen nicht möglich, da sämtliche Teilnehmer/Teilnehmerinnen als gleichwertige Akteure im Blockchain-Ökosystem agieren können.

1.3 Aufbau dieser Arbeit

Um die definierte Forschungsfrage beantworten zu können, ist es zunächst notwendig, ein Verständnis für die wesentlichen Bestandteile des Prototyps zu schaffen. Hierbei werden die für diese Thematik benötigten, theoretischen Grundlagen in Kapitel 2 aufgearbeitet. Dabei wird das Grundgerüst der Blockchain-Technologie beschrieben, es wird an die Idee und die Funktionsweise von Smart-Contracts herangeführt und darauf aufbauend werden die Bestandteile und Funktionen von NFT aufgezeigt. Folglich wird der Begriff „selbstbestimmte (digitale) Identität“ in seinen Einzelheiten erläutert, um ein Verständnis für diese Begrifflichkeit zu vermitteln und dadurch auf den Mehrwert dieser Thematik aufmerksam zu machen. In Kapitel 3 wird einerseits die in dieser Arbeit angewandte Methodik des „Prototyping“ näher beschrieben und es wird darauf eingegangen, wie in dieser Arbeit konkret vorgegangen wurde, um das Prototyping hinsichtlich dieser Thematik durchzuführen. Auch wird in diesem Kapitel ein theoretisches Konstrukt einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Nachweisen aufgezeigt und – daraus abgeleitet – werden abschließend Requirements – welche sich aus den theoretischen Grundlagen ableiten lassen – definiert. Das Kapitel 4 stellt jenen Teil dieser Arbeit dar, welcher die Ergebnisse des Prototyps schriftlich und grafisch gestützt aufzeigt. In Kapitel 5 erfolgt eine Diskussion hinsichtlich der festgestellten Herausforderungen in Bezug auf die bereits genannte Evaluierung und die Umsetzbarkeit auf Basis der vorhandenen, technologischen Rahmenbedingungen. Zum Abschluss erfolgt in Kapitel 6 ein Fazit über die gewonnenen Erkenntnisse und ein Ausblick über weitere, in dieser Arbeit nicht behandelte Fragen, die in weiteren Forschungsarbeiten behandelt werden können.

2 STAND DER FORSCHUNG

Die erstmals im Jahr 2008 konzipierte Blockchain-Technologie hat seit ihrer Veröffentlichung in den letzten 13 Jahren eine alternative Umsetzmöglichkeit in Hinsicht auf die Verwaltung von Daten in zentraler Form zu einer dezentralen Alternative geschaffen. Auch wenn diese Variante zur dezentralen Verwaltung eines Transaktionsregisters (auch bekannt als Distributed-Ledger-Technology (Lewin et al., 2019)) nicht für jedes Geschäftsmodell einen ersichtlichen Vorteil mit sich bringt, hat die Blockchain aufgrund ihrer Eigenschaften in Bezug auf Datensicherheit nicht nur eine Welle an neuen Start-Ups und Projekten ausgelöst, sondern auch die Politik beschäftigt sich mittlerweile mit dieser Technologie. Im österreichischen Regierungsprogramm 2020-2024 wurde dahingehend folgender Punkt festgehalten:

> *Schaffung einer vorausschauenden österreichischen Positionierung zur Förderung, Anwendung und Regulierung der Blockchain-Technologie und ihrer unterschiedlichen Anwendungen (z. B. Kryptowährungen). Unter Miteinbeziehung relevanter Stakeholder in Politik (z. B. Finanzministerium, Wirtschaftsministerium, Infrastruktur- und Technologieressort) und Forschung. Einsatz auf EU-Ebene, um Österreichs Beitrag zu Europas Blockchain-Strategie sicherzustellen (in Anwendung und Regulierung) (Bundeskanzleramt Österreich, 2020)*

Spricht man im Zusammenhang mit Blockchain über „Anwendungen“, so handelt es sich hierbei – neben den unterschiedlichen Kryptowährungen – zumeist um eine dezentrale Applikation (Dapp), welche mittels sogenannten Smart-Contracts umgesetzt wird (siehe Kapitel 2.2). Die Blockchain stellt hierbei die wesentlichen, kryptographischen Grundbausteine für die Sicherheit des Netzwerkes bereit, wobei unterschiedliche Anwendungen/Projekte auch unterschiedliche Blockchain-Realisierungen als Basis heranziehen können. Grundsätzlich unterscheiden sich diese verschiedenen Realisierungen der Blockchain-Technologie in der Definition und der Umsetzung eines geeigneten Konsensus-Algorithmus, welche im Wesentlichen zum Schutz vor Attacken auf das Netzwerk eingesetzt werden (siehe Kapitel 2.1.2 und Kapitel 2.1.3).

In den folgenden Unterkapiteln wird der Fokus der Fokus neben den einleitenden, generellen Grundzügen der Blockchain auf Ethereum gelegt, denn hierbei handelt es sich um das bisher größte Projekt, welches die technologischen Grundlagen für dieses Vorhaben bereits zur Verfügung stellt. Grundsätzlich gibt es ein großes Angebot an Projekten, welche sich bei der Entwicklung ihrer eigenen Blockchain an jenen Aspekten orientieren, welche für die Beantwortung der in dieser Arbeit gestellten Forschungsfrage benötigt werden. Da sich jedoch Ethereum bereits stark am Markt etabliert hat und sich die benötigten Tools in der großen Community praktisch bewährt haben, werden detaillierte Vergleiche zwischen anderen Projekten in dieser Arbeit nicht behandelt.

2.1 Blockchain

Bei der Blockchain-Technologie handelt es sich um ein Verfahren der dezentralen Datenspeicherung, welches – mithilfe ihrer kryptographischen Bestandteile – die Integrität der in ihr verarbeiteten Daten wahrt, sodass Änderungen vergangener Transaktionen nahezu unmöglich sind. Das zugrundeliegende Netzwerk ist als verteiltes System organisiert, d.h. dass jeder Teilnehmer/jede Teilnehmerin als sogenannter „Node“ in diesem System partizipieren kann. Die einzelnen Nodes sind hierbei für den Informationsgewinn und Informationserhalt zuständig. Wie in Abbildung 1 ersichtlich, kann jeder Node Informationen an einen anderen Node weiterleiten und auch Informationen von anderen Nodes erhalten. Ziel dabei ist es, dass alle Nodes denselben Informationsgehalt aufweisen, sodass es für den Enduser keine Rolle spielt, von welchem Node er die Informationen erhält bzw. an welchen Node er Informationen übermittelt. (Bashir, 2017)

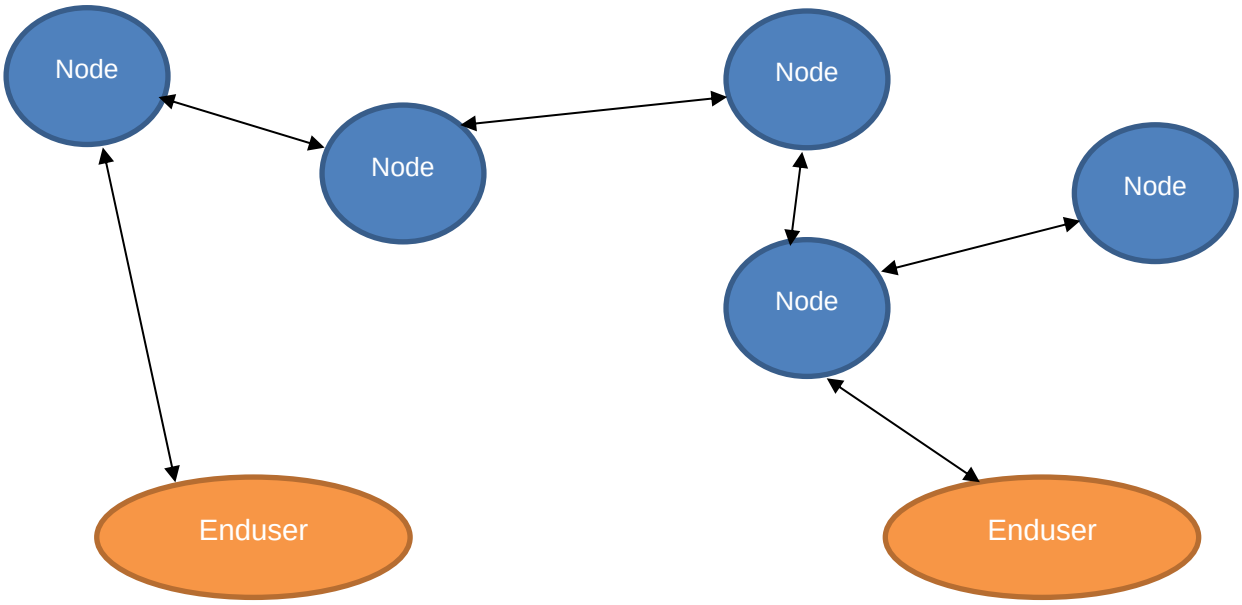

Abbildung 1: Verteiltes System (Quelle: eigene Darstellung)

Die Blockchain stellt hierbei die Informationen bereit, welche von einzelnen Nodes gesichert und an andere Nodes weitergleitet werden.

2.1.1 Kernelemente der Blockchain

Eine Blockchain besteht grundsätzlich aus zwei wesentlichen Bestandteilen. Den Blöcken, welche neben den Metadaten die gewünschten Informationen beinhalten, die es zu speichern gilt, sowie den Verbindungen zwischen den einzelnen Blöcken, sodass eine chronologische Reihenfolge der Blöcke gewährleistet ist. Jeder Block verweist durch diese Verbindungen eindeutig auf einen einzelnen, vorhergehenden Block, sodass eine Kette bis zurück zum sogenannten „Genesis-Block“ entsteht. Als Genesis-Block wird der erste Block bezeichnet, welcher bei der Kreierung der Blockchain „hardcoded“ implementiert wurde und von welchem ausgehend alle weiteren Blöcke angekettet wurden. (Bashir, 2017) Als visuelle Unterstützung wird der Aufbau einer Blockchain in nachfolgender Abbildung grafisch und vereinfacht dargestellt.

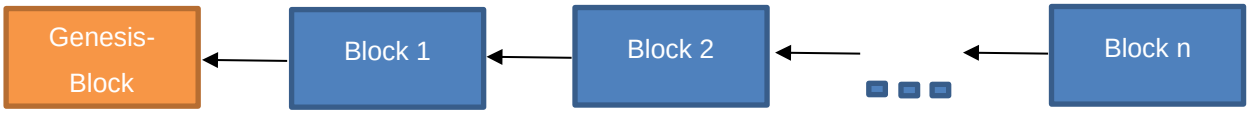

Abbildung 2: Aufbau einer Blockchain (Quelle: eigene Darstellung)

Um den Aufbau sowie die Kreierung eines einzelnen Blocks verständlich darstellen zu können, ist es zunächst wichtig, auf einige wichtige kryptographische Grundlagen einzugehen. Diese werden folglich theoretisch aufgearbeitet und dahingehend beschrieben, welche Rolle sie im Zusammenhang mit der Blockchain-Technologie spielen.

Die Hashfunktion

Eine Hashfunktion wird in der Kryptographie typischerweise dafür verwendet, um Informationen auf ihre Integrität zu prüfen, d.h. dass man eindeutig nachvollziehen kann, dass eine Nachricht am Übertragungsweg nicht verändert wurde oder Informationen verloren gegangen sind. Eine Hashfunktion sollte somit folgende Merkmale mit sich bringen (Wätjen, 2018):

- Der Output einer Hashfunktion resultiert – unabhängig von der Länge des Input – immer in einer vordefinierten Bitlänge.
- Ist der Input einer Hashfunktion bekannt, so ist der zugehörige Output einfach zu berechnen.
- Ist der Output einer Hashfunktion bekannt, so ist der zugehörige Input annähernd unmöglich zu berechnen.

Typischerweise ist es für eine robuste Hashfunktion notwendig, dass sie kollisionsresistent ist. Die Wahrscheinlichkeit dafür, dass zwei unterschiedliche Inputs denselben Output aufweisen, soll demnach möglichst gering sein, um die Integrität von Informationen zuverlässig prüfen zu können. (Wätjen, 2018)

In einer Blockchain wird die Hashfunktion einerseits für die Verkettung zum vorhergehenden Block eingesetzt und andererseits, beim sogenannten „Proof-of-Work“, um das Netzwerk vor Angriffen zu schützen (siehe dazu Kapitel 2.1.2). Jeder gehashte Block resultiert – aufgrund der unterschiedlichen, enthaltenen Informationen – in einem anderen *z*, sodass die Information in einem Block über den Hashwert des vorherigen Blocks die Verkettung eindeutig macht und somit die Historie der Blockchain eindeutig nachvollzogen werden kann (siehe Abbildung 3).

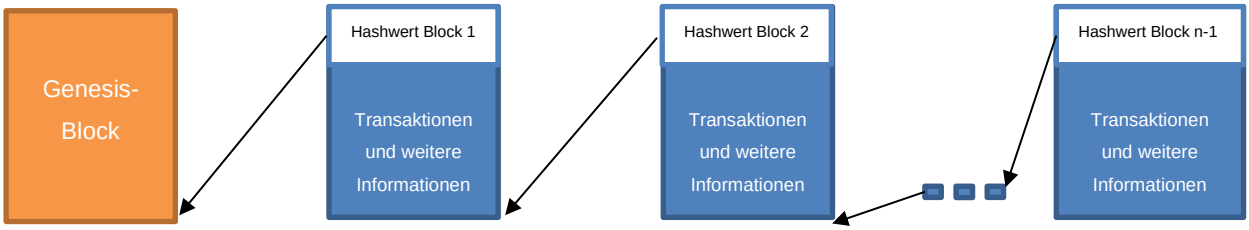

Abbildung 3: Blockchain-Verkettung mittels Hashwerte (Quelle: eigene Darstellung)

Unter „weitere Informationen“ werden in der oben angeführten Abbildung Metadaten verstanden, welche je nach Konsensusalgorithmus voneinander abweichen. Diese Algorithmen und Metadaten werden in Kapitel 2.1.2 genauer erläutert, sodass der gesamte Aufbau eines Blocks schrittweise nähergebracht werden kann.

Asymmetrisches Schlüsselpaar

In der Kryptographie unterscheidet man im Wesentlichen zwischen zwei unterschiedlichen Verschlüsselungsverfahren. Sie werden eingesetzt, um Informationen am Übertragungsweg für Dritte unleserlich darzustellen und sichern somit die Vertraulichkeit von Informationen ab. Bei der symmetrischen Verschlüsselung erfolgt das Verfahren zur Verschlüsselung von Informationen mit demselben Schlüssel wie zur Entschlüsselung, wohingegen beim asymmetrischen Verfahren die Ver- und Entschlüsselung einer Nachricht mittels Schlüsselpaar – bestehend aus privatem und öffentlichem Schlüssel – erfolgt. Hierbei verschlüsselt der Versender/die Versenderin die Nachricht mit dem privaten Schlüssel und der Empfänger/die Empfängerin kann diese Nachricht wiederum mit dem zugehörigen, öffentlichen Schlüssel entschlüsseln. (Buchmann, 2008)

Das asymmetrische Schlüsselpaar dient in einer Blockchain unter anderem der Erstellung von Wallet-Adressen – also jenen Adressen, an welche Transaktionen versendet werden können, welche wiederum den Besitz von digitalen Vermögenswerten widerspiegeln. Für eine Ethereum Wallet-Adresse wird der aus dem privaten Schlüssel – über eine elliptische Kurve (ECDSA) – berechnete öffentliche Schlüssel herangezogen und mittels der Hashfunktion Keccak-256 gehashed, sodass eine hexadezimale Zeichenkette von 32 Byte entsteht. Die letzten 20 Bytes ergeben dabei in Kombination mit dem Präfix 0x0 die – vom öffentlichen Schlüssel abgeleitete – Ethereum Wallet-Adresse, an welche alle auf Ethereum basierenden Token versendet werden können. (Bergmann, 2017)

Digitale Signatur

Ein weiteres Merkmal des asymmetrischen Verschlüsselungsverfahrens ist die Möglichkeit zur digitalen Signatur. Die digitale Signatur stellt die Authentizität von Informationen sicher, d.h. dass die Identität des Versenders/der Versenderin einer Nachricht eindeutig verifiziert werden kann. Hierfür wird neben der verschlüsselten Nachricht vom Versender/von der Versenderin ein Hashwert der versendeten Nachricht berechnet und mittels eigenem privaten Schlüssel verschlüsselt (signiert). Der Empfänger/Die Empfängerin dieser Nachricht ist nun einerseits in der

Lage, die verschlüsselte Nachricht mit dem eigenen privaten Schlüssel zu entschlüsseln, und andererseits den erhaltenen Hashwert – durch Anwendung desselben Hashverfahrens und der Entschlüsselung des signierten Hashwertes mit dem öffentlichen Schlüssel des Versenders/der Versenderin – mit dem selbst berechneten Hashwert abzugleichen. (Buchmann, 2008)

In der Blockchain spielt vor allem die digitale Signatur eine tragende Rolle, denn bei jeder Transaktion muss diese vom Versender/von der Versenderin auch signiert werden, sodass jeder Teilnehmer/jede Teilnehmerin diese Signatur mit dem zugehörigen öffentlichen Schlüssel verifizieren kann. (Bashir, 2017) Aufgrund der Tatsache, dass diese digitale Signatur lediglich von jener Person durchgeführt werden kann, die auch im Besitz des privaten Schlüssels ist, kann die Transaktion als valide angesehen werden und die Transaktion wird in einem neuen Block aufgenommen.

2.1.2 Konsensusalgorithmen

Grundsätzlich unterscheidet man heutzutage in Bezug auf die Konzipierung einer Blockchain im Wesentlichen zwischen zwei Konsensusalgorithmen – „Proof-of-Work“ (PoW) und „Proof-of-Stake“ (PoS). Ein Konsensusalgorithmus kommt in einer Blockchain zur Anwendung, um einerseits neue Blöcke an die Blockchain zu knüpfen und andererseits, um das Netzwerk vor ungewünschten Angriffen zu schützen. In einem dezentralen Netzwerk gibt es (per Definition) keine zentrale Partei, welche die durchgeführten Transaktionen auf ihre Gültigkeit kontrolliert und sicherstellt, dass dieselben Transaktionen nicht doppelt durchgeführt werden. Betrachtet man das Blockchain-Netzwerk, in welchem jeder Teilnehmer/jede Teilnehmerin seine/ihre eigenen Transaktionen signieren und damit die Authentizität einer Transaktion sicherstellen kann, braucht es ein Verfahren, um sicherzustellen, dass eine Transaktion nicht doppelt durchgeführt wurde. Ohne ein solches Verfahren könnte (wie in Abbildung 4 ersichtlich) ein Enduser A *x* digitale Währungseinheiten sowohl an Enduser B als auch an Enduser C transferieren bzw. als Eintrag in der Blockchain vermerken, obwohl Enduser A lediglich über die genannte Anzahl *x* an digitalen Währungseinheiten verfügt, was auch als „Double-Spending-Problem“ bezeichnet wird. (Bashir, 2017)

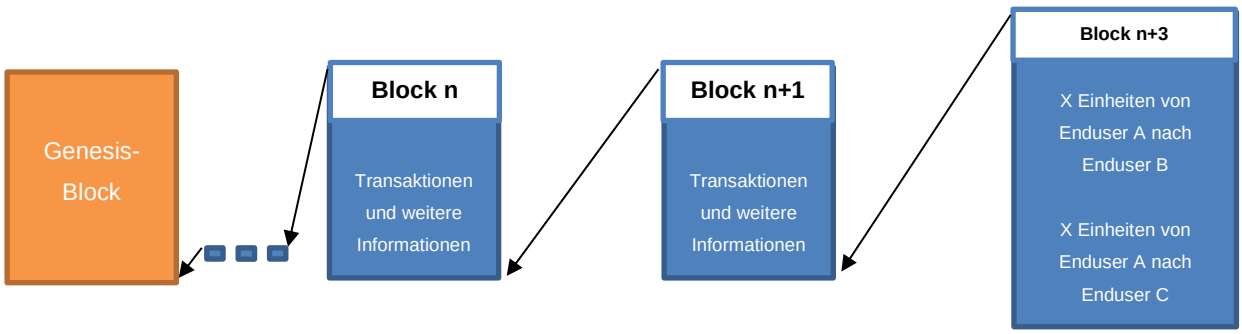

Abbildung 4: Double-Spending-Problem (Quelle: eigene Darstellung)

Um dieser Problematik entgegenwirken zu können, braucht es Akteure, welche die Gültigkeit der Transaktionen prüfen, sie in einem Block zusammenfassen und diesen mit dem vorherigen Block verketten. Da in einem typischen dezentralen Netzwerk wie der Blockchain jede Person gleichberechtigt ist, kann auch jede Person einen Block mit einzigartigen Transaktionen kreieren, weshalb die Notwendigkeit besteht, sich innerhalb des Netzwerks zu einigen (daher Konsensus), welcher Block für die nächste Verkettung ausgewählt wird. Hierfür ist vorgesehen, dass man für die Verkettung zum vorherigen Block eine gewichtete Wahrscheinlichkeit anhand unterschiedlicher Ressourcen (je nach Konsensusalgorithmus) heranzieht (Bashir, 2017).

Proof-of-Work

Beim PoW werden zum Schutz des Netzwerks vor Double-Spending-Attacken und zur Kreierung von Konsens Kosten in Form von Rechenleistung herangezogen. Hierfür verifizieren sogenannte „Miner“ unterschiedliche Transaktionen auf ihre Gültigkeit und bündeln diese Transaktionen in einem neuen Block. Um diesen Block nun an die Blockchain anhängen zu können, muss der kreierte Block mittels Hashfunktion einen vordefinierten Hashwert mit einer vordefinierten Bitlänge ergeben. Wie bereits erläutert, zeichnet sich eine zuverlässige Hashfunktion dadurch aus, dass unterschiedliche Inputs in unterschiedlichen Outputs resultieren bzw. dieselben Inputs dieselben Outputs ergeben. Würde man also für einen Block ohne weitere Parameter zwei Mal dieselbe Hashfunktion anwenden, würde sich der Output dieser zwei Berechnungen nicht unterscheiden. Hierfür kommt eine sogenannte „Nonce“ im Blockheader zum Einsatz, welche von jedem Miner frei bestimmt werden kann, um durch die Anpassung dieser Nonce bei derselben Hashfunktion unterschiedliche Outputs erzeugen zu können. In den gängigsten Proof-of-Work-Blockchains wurde definiert, dass ein Block nur dann an den vorherigen Block gekettet werden darf, sobald man mittels dieser Hashberechnung einen Binär-Output erzeugt, welcher eine definierte Anzahl an führenden aufeinanderfolgenden Nullen aufweist (Bashir, 2017). Beispielsweise hätte man dadurch bei einer Vorgabe von einer führenden Null eine Wahrscheinlichkeit von 1:2, um einen solchen Output bei der ersten Berechnung zu erzeugen, bei zwei führenden Nullen eine Wahrscheinlichkeit von 1:4, etc. Die Anzahl an durchschnittlichen Versuchen verdoppelt sich also mit jeder zusätzlichen führenden Null, womit auch die benötigte Rechenleistung steigen muss, sofern man den benötigten Output schneller erzeugt haben möchte als alle anderen Miner im Netzwerk. Die Definition der Anzahl an führenden Nullen wird je nach vorhandener Rechenleistung im Netzwerk dynamisch korrigiert, sodass die durchschnittliche Dauer für die Berechnung eines neuen (validen) Blocks – durch Erhöhung oder Reduzierung der Anzahl an führenden Nullen unter Berücksichtigung der Rechenleistung im gesamten Netzwerk – nahezu konstant bleibt (Bashir, 2017). Ein Miner müsste also im Mittel über 50% der gesamten Rechenleistung besitzen, um einen betrügerischen Block an die Blockchain anketten zu dürfen, was mit enormen Kosten in Form von Rechenleistung und dem dafür benötigten Stromverbrauch einhergeht. Belohnt wird dieser Rechenaufwand und damit die Absicherung des Netzwerks in Form von Block-Rewards, d.h. dass jener Miner, welcher die Berechnung am schnellsten durchführen konnte, sich selbst eine Transaktion in Form von Coins der jeweiligen Blockchain in definierter Höhe zuschreiben darf.

Proof-of-Stake

Im Gegensatz zum Proof-of-Work, bei welchem die Sicherheit des Netzwerks an 50% der Rechenleistung im Netzwerk gekoppelt ist, kommen beim Proof-of-Stake Kosten in Form von Vermögenswerten zum Einsatz. Hierbei wird ein Block nicht mittels vorhergehender Berechnung eines Hashwertes an die Blockchain gekettet, sondern die Wahrscheinlichkeit, dass man einen Block an den vorherigen Block anhängen darf, steigt unter anderem mit der Anzahl an bestimmten Coins einer bestimmten Kryptowährung, welche man als sogenanntes „Staking" in der Blockchain hinterlegt. Grundsätzlich gilt also, dass die Wahrscheinlichkeit, einen Block an die Blockchain ketten zu dürfen, steigt, je höher das eingesetzte Kapital ist. Jener Node, welcher die Blockchain mit seinem Block erweitern darf, darf sich selbst – wie auch beim Proof-of-Work – eine Transaktion in Form von Coins der jeweiligen Blockchain in definierter Höhe zuschreiben. Da ein Node mit einem hohen Kapitaleinsatz auch eine höhere Wahrscheinlichkeit hat, sich diesen Reward zuschreiben zu dürfen, steigt dessen Kapitaleinsatz auch konstant im Vergleich zu jenen Nodes, welche einen geringeren Kapitaleinsatz zur Verfügung stellen. Dies resultiert wiederum darin, dass durch diesen Reward der Kapitaleinsatz und damit die Wahrscheinlichkeit dieses Nodes für den nächsten Block steigt. Um diese Problematik zu umgehen, wurden verschiedene Methoden entwickelt, wobei folgend die zwei gängigsten Methoden kurz erklärt werden. (Antonopoulos & Wood, 2018)

Bei der „Randomized-Block-Selection" wird jedem Node ein gewisser Hashwert zugewiesen. Die Auswahl eines Nodes, welcher den nächsten Block an die Blockchain ketten darf, erfolgt hierbei durch die Kombination aus niedrigstem Hashwert und höchstem Kapitaleinsatz. (Hazari & Mahmoud, 2019)

Im Gegensatz dazu wird bei der „Coin-Age-Based-Selection" ein Node anhand des Kapitaleinsatzes und der Anzahl an Tagen, an denen die entsprechenden Coins bereits gestaked sind, ausgewählt. Je länger die Coins also bereits gestaked sind und je höher der Kapitaleinsatz ist, desto höher ist auch die Wahrscheinlichkeit, dass ein Node den Block an die Blockchain anketten und sich damit den Reward zuschreiben darf. (Hazari & Mahmoud, 2019)

2.1.3 Kritische Aspekte

Wie in den vorhergehenden Unterkapiteln aufgezeigt, tragen unterschiedliche technologische Bestandteile der Blockchain dazu bei, dass durchgeführte Transaktionen als authentisch und die Informationen in der Blockchain als integer angesehen werden können. Je mehr Personen am jeweiligen Konsensusalgorithmus teilnehmen, desto sicherer wird das Netzwerk gegenüber unerwünschten Attacken, da die Kosten – um eine Mehrheit in Form von Ressourcen abzubilden – mit jedem weiteren Teilnehmer/jeder weiteren Teilnehmerin steigen. In diesem Kapitel werden nun einige kritische Aspekte betrachtet, welche bereits in der Vergangenheit zu Problemen geführt haben und möglicherweise auch in Zukunft einer Massenadaptierung im Weg stehen könnten.

51%-Attacke

Als 51%-Attacke wird ein Angriff auf eine dezentrale PoW-Blockchain betrachtet, bei welchem mittels sogenanntem „Hardfork“ versucht wird, eine Abspaltung (ab einem gewissen Block) zu schaffen und an diese neue Verkettung schneller neue Blöcke anzuketten als in der ursprünglichen Verkettung. Generell wird davon ausgegangen, dass die Mehrheit der Miner zu Gunsten des gesamten Netzwerks agieren und sich nicht durch betrügerische Absichten selbst bereichern wollen, sodass diese Miner nach einem möglichen kurzen Split der Kette wieder an der längsten Kette weiterarbeiten. Sollte sich eine Gruppe von Minern (mit einer hohen Rechenleistung) dazu entscheiden, einen Teil der Blockchain rückgängig zu machen (siehe dazu Abbildung 5), so könnten diese an einem Block *n* einen zweiten Block *n+1b* verketten und an dieser neuen Kette so lange weiterarbeiten, bis sie die ursprüngliche Kette in der Anzahl der Blöcke *n+1a* eingeholt hat. Sollte dieses Szenario eintreten, so würden alle Miner wieder auf die neue Kette mit einer höheren Anzahl an Blöcken *n+2b* überspringen, welche jedoch keine der durchgeführten Transaktionen der anderen Kette (seit dem Split) beinhalten würde. (Herschel & Adobatti, o.J.)

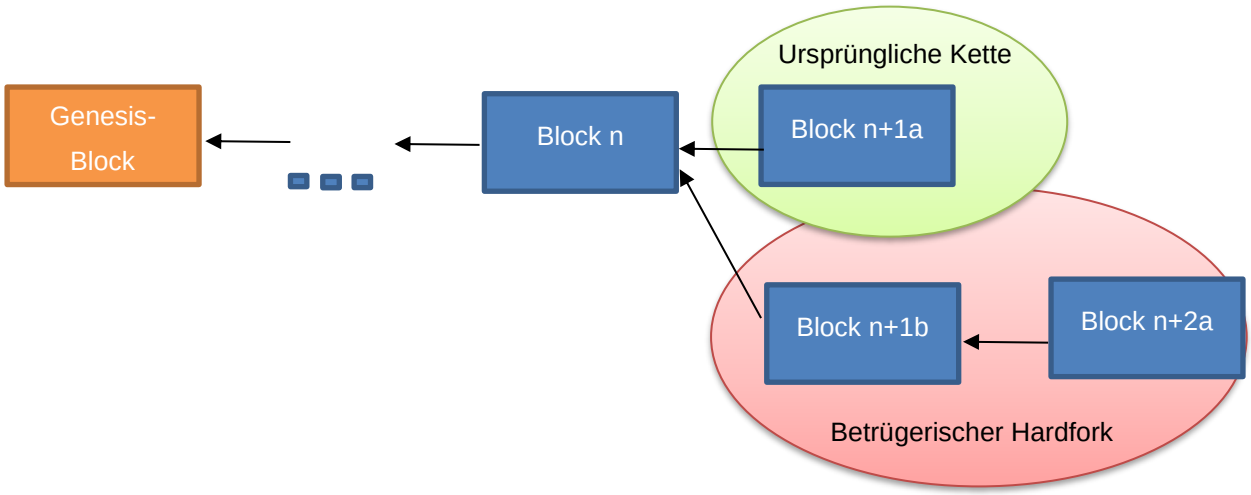

Abbildung 5: 51%-Attacke (Quelle: eigene Darstellung)

Theoretisch wäre es einem Angreifer dadurch möglich, eine Transaktion durchzuführen und – sobald diese Transaktion mittels neuen Blocks an die Blockchain angehängt wurde – einen Hardfork vor der Transaktion durchzuführen. Sobald diese Kette länger ist als jene, welche die durchgeführte Transaktion beinhaltet, würden die Miner auf die längere Kette überspringen und die durchgeführte Transaktion wäre obsolet. So könnte, vereinfacht ausgedrückt, eine Gegenleistung für eine Bezahlung bereits erbracht worden sein und diese Bezahlung im Nachhinein rückgängig gemacht werden.

Eine solche Attacke ist zwar aufgrund der enormen Kosten, die dadurch entstehen, unwahrscheinlich, jedoch findet sich diese Problematik auch im täglichen Miningprozess wieder, denn es kann auch vorkommen, dass aufgrund von Latenzen zwei Blöcke nahezu zeitglich an die Blockchain gehängt werden. In diesem Fall wird kurzzeitig an beiden Ketten weitergearbeitet und sobald ein darauffolgender Block schneller an den vorherigen Block angekettet wird, springen

die Miner wiederum auf die längere Kette über. Auch hierbei würden die Transaktionen aus der langsameren Kette verloren gehen bzw. in der längeren Kette nicht berücksichtigt werden.

Coding-Fehler

Die Vergangenheit hat gezeigt, dass Coding-Fehler auch in der Blockchain ein Problem darstellen können. In den ersten Jahren von Ethereum konnte ein Ethereum-User einen Coding-Fehler im Blockchain-Source-Code entdecken, welcher es dem User erlaubte, Vermögenswerte einer dezentralen autonomen Organisation auf seine eigene Krypto-Wallet zu transferieren (Madnick, 2019). Dieser Umstand zeigt deutlich, dass auch in großen Open-Source-Projekten kein fehlerfreier Code gewährleistet werden kann und gerade aufgrund der öffentlichen Einsicht auch Coding-Fehler für Außenstehende einfacher zu erkennen sind.

Stromverbrauch

Einer der größten Kritikpunkte von PoW-Blockchains ist der enorme Stromverbrauch, welcher bei den Hashwertberechnungen anfällt. Schätzungen zufolge hat die größte aller Blockchain-Anwendungen – Bitcoin – einen Energieverbrauch von 135 Terawattstunden pro Jahr, was einen höheren Energieverbrauch als vergleichsweise der gesamte Staat Norwegen darstellt (Cambridge Centre for Alternative Finance, 2021). PoS ist dahingehend wesentlich energieschonender, da hierbei der Konsensus aufgrund von eingesetztem Kapital anstatt von Rechenleistung kreiert wird. Ob und in welcher Form dieser Energiebedarf in Zukunft für alternative Anwendungsbereiche verwendet werden kann, wird sich zeigen, jedoch sollte dieser Aspekt in Hinsicht auf die globale Erderwärmung kritisch angesehen werden.

Selbstverantwortung

Wie bereits in Kapitel 2.1.1 dargestellt, wird für eine Transaktion in der Blockchain eine digitale Signatur benötigt, welche lediglich mit dem privaten Schlüssel durchgeführt werden kann. Sollte dieser private Schlüssel verloren gehen, können Transaktionen, welche diesen verlorenen Schlüssel benötigen, nicht mehr stattfinden. Innerhalb eines zentralen Netzwerkes kann ein verlorengegangenes Passwort über einen zentralen Support wiederhergestellt werden. Dies ist in einem dezentralen Netzwerk nicht möglich, weshalb hierbei die Enduser zur Gänze in der Eigenverantwortung stehen, ihren privaten Schlüssel nicht zu verlieren. Alternativ könnten hierbei unterschiedliche Service-Anbieter eine zentrale Verwaltung der privaten Schlüssel zur Verfügung stellen, was jedoch ein Vertrauen in diese Anbieter voraussetzt.

Große Mining- und Staking-Pools

In Kapitel 2.1.2 wurde bereits beschrieben, dass Teilnehmer/Teilnehmerinnen einer Blockchain je nach Konsensusalgorithmus an gewissen Rewards profitieren können, sollten sie eine gewisse Wahrscheinlichkeit erreichen, einen neuen Block an die Blockchain anketten zu dürfen. Dies

machen sich auch viele Menschen mittels sogenannten „Mining- und Staking-Pools“ zu Nutze. Hierbei findet sich eine große Anzahl an Menschen in einem Pool wieder, sodass sich diese einerseits die Kosten für das Mining oder für den Kapitaleinsatz beim Staking teilen können und andererseits erlangen sie dadurch eine höhere Wahrscheinlichkeit, die Rewards für sich zu gewinnen und unter sich aufzuteilen. Solche Miningpools sind zwar in den gängigen Blockchain-Netzwerken typisch und auch legitim, jedoch würden wenige (zu große) Mining- und/oder Staking-Pools zu einer gewissen Zentralisierung führen. Betrachtet man die Verteilung der Rechenleistung im Ethereum-Netzwerk, so ist ersichtlich, dass die drei größten Miningpools mehr als die Hälfte der Rechenleistung des gesamten Netzwerkes zur Verfügung stellen (Bitfly, 2021). Sollte der Fall eintreten, dass diese Miningpools sich auf eine Attacke auf das Ethereum-Netzwerk einigen, so könnten sie sich mittels 51% entweder selbst bereichern oder anderen Teilnehmern/Teilnehmerinnen schaden.

2.2 Smart-Contracts

Obwohl das erste Whitepaper zur Blockchain-Technologie erst im Jahre 2008 beschrieben und veröffentlicht wurde, wurde jene Funktion von computergestützten, automatisierten Verträgen, welche heutzutage über die Blockchain-Technologie abgewickelt werden, bereits im Jahre 1996 theoretisch beschrieben. Die Vision hinter Nick Szabo’s Idee war, Computerprogramme zu entwickeln, welche sowohl von Menschen als auch von Computern gelesen werden können und darüber hinaus sollte es nicht möglich sein, diese smarten Verträge von einer einzelnen Partei einseitig zu kündigen. Unter dem Begriff „Smart-Contracts“ werden heutzutage Blockchain-Anwendungen bezeichnet, welche Verträge zwischen zwei oder mehreren Vertragsparteien ohne zentralen Mittelsmann abwickeln. Grundsätzlich handelt es sich bei Smart-Contracts um computergestützte Wenn-Dann-Beziehungen, welche sich von klassischen zentralen Anwendungen darin unterscheiden, dass sie aufgrund der Eigenschaften einer Blockchain als manipulationssicher angesehen werden können. Dies bedeutet, dass (sofern dies im darunterliegenden Code nicht bewusst oder unbewusst durch eventuelle Coding-Fehler erlaubt wird) weder eine der involvierten Vertragsparteien noch eine dritte, unabhängige Partei im Nachhinein den Vertrag rückgängig machen kann. (Fries & Paal, 2019)

2.2.1 Technologische Definition und Funktionsweise

Die ursprüngliche, theoretische Idee von Smart-Contracts wurde knapp 20 Jahre später von Ethereum-Mitgründer Vitalik Buterin aufgegriffen und praktisch umgesetzt. Vitalik Buterin stellte damals fest, dass die Blockchain-Technologie weit mehr Möglichkeiten bieten könne, als in der bislang größten Anwendung Bitcoin angeboten werden. Er machte darauf aufmerksam, dass jene Programmiersprache – welche für Bitcoin verwendet wird – zu wenig Möglichkeiten für die Softwareentwicklung bietet und entwickelte daraufhin die Ethereum-Blockchain, welche es – basierend auf der verwendeten Programmiersprache Solidity – erlaubt, weitere Anwendungsgebiete der Blockchain-Technologie zu entwickeln und zu nutzen. (Fries & Paal,

2019) Peter Kenning und Jörn Lamla definierten den Begriff Smart-Contracts in ihrem Buch „Entgrenzungen des Konsums“ folgendermaßen:

> *„Als „Smart Contract“ wird ein Programmcode bezeichnet, der auf einer Blockchain läuft und dort digitale Assets oder Repräsentationen körperlicher Gegenstände bzw. der daran bestehenden Rechte auf der Grundlage von anderen (externen) Daten, die zum Zeitpunkt der Programmierung des Codes noch nicht feststanden, zwischen zwei oder mehreren Parteien in Form von Transaktionen neu zuordnet.“ (Kenning & Lamla, 2018)*

Mittels Smart-Contracts können also nicht nur Vermögenswerte nach einer festgelegten Bedingung den Besitzer/die Besitzerin wechseln, sondern auch Rechte über physische und digitale Assets übertragen werden. Um diese Definition und das grundsätzliche Verhalten eines Smart-Contracts einfacher darzulegen, wird nun der Kauf/Verkauf eines – über einen Smart-Contract abgewickelten – digitalen Kunstwerks als Beispiel betrachtet und mittels Abbildung 6 grafisch begleitet. In diesem Beispiel wird vorerst nicht auf eine mögliche Einzigartigkeit und mögliche Kopien des Kunstwerks eingegangen, sondern lediglich auf den – im Code hinterlegten, vertraglich festgelegten – Austausch von digitalen Vermögenswerten und digitalem Gut. Eine ausführliche Beschreibung zur Feststellung der Einzigartigkeit eines digitalen Assets erfolgt in Kapitel 2.3.

Vertragspartei A möchte von Vertragspartei B ein digitales Kunstwerk erwerben. Beide Vertragsparteien einigen sich darauf, diesen Kauf/Verkauf über einen Smart-Contract der Ethereum-Blockchain abzuwickeln. Hierfür verweist Vertragspartei B auf einen Smart-Contract, in welchem festgehalten ist, dass die Rechte des zu verkaufenden Objekts dann an Vertragspartei A übertragen wird, sobald er/sie Vermögenswerte in einer definierten Höhe an Vertragspartei B überweist. Dieser smarte Vertrag wird nun in jenem Moment ausgeführt, in welchem Vertragspartei A die festgelegten Vermögenswerte überweist, sodass dieses Vermögen und auch das Recht am digitalen Kunstwerk ihren Besitzer/ihre Besitzerin wechseln.

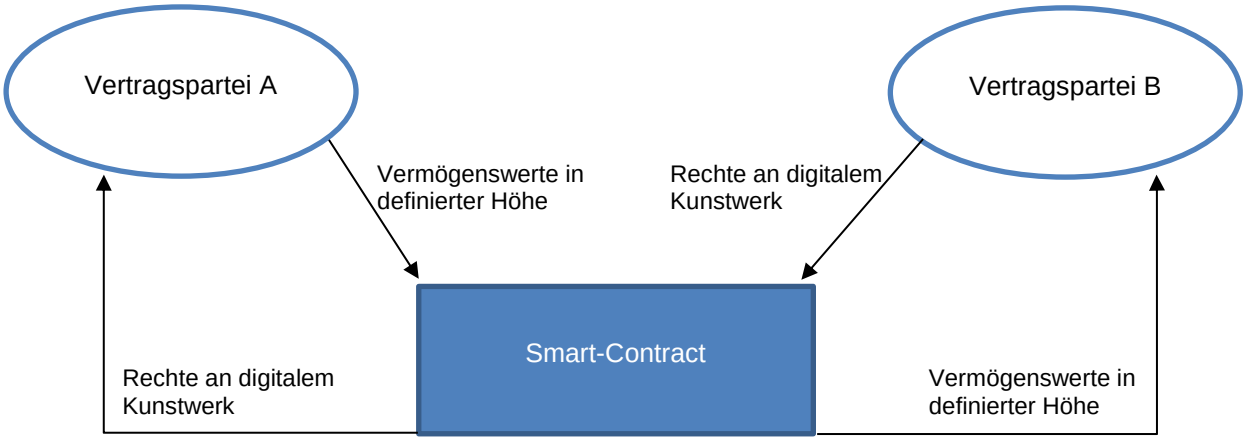

Abbildung 6: grafische Darstellung eines Smart-Contracts (Quelle: eigene Darstellung)

Wie in oben angeführter Abbildung ersichtlich interagieren beide Vertragsparteien nicht direkt miteinander, sondern lediglich über den entsprechenden Smart-Contract, sodass dieser Vertragsabschluss weder ein Vertrauen der beiden Vertragsparteien zueinander noch ein Vertrauen zu einer dritten Person voraussetzt. Vertraut werden muss hierbei lediglich dem – im Smart-Contract – definierten Programmcode, welcher als Open-Source-Code von jeder Partei eingesehen werden kann.

2.2.2 Rechtliche Definition

Betrachtet man Smart-Contracts aus technologieneutraler Sicht, so kann man erkennen, dass die Definition des Begriffes nicht erst seit der Einführung von Blockchains aus rechtlicher Perspektive untersucht wird. Hierbei erfolgt jene Definition von Smart-Contracts ohne Berücksichtigung der Blockchain als ausführende Instanz und ein solcher smarter Vertrag muss eine Kombination aus „(i) einem digital prüfbaren Ereignis; (ii) einem Programmcode, welcher das Ereignis verarbeitet; und (iii) einer rechtlich relevanten Handlung, welche auf Grundlage des Ereignisses ausgeführt wird“ aufweisen (Fries & Paal, 2019). Anhand dieser Definition geht hervor, dass einem Smart-Contract nicht zwingenderweise ein dezentrales Netzwerk zu Grunde liegen muss, sondern jegliche computergestützte Wenn-Dann-Funktion, welche ein prüfbares Ereignis auf Basis einer rechtlich relevanten Handlung liefert, ausreicht. Somit kann auch ein Dauerauftrag bei einer Bank – welcher nur dann ausgeführt wird, wenn der Kontoinhaber/die Kontoinhaberin die Willenserklärung dahingehend mitteilt – als Smart-Contract angesehen werden, der sich von Smart-Contracts im Sinne der Blockchain-Technologie in folgenden Punkten im Wesentlichen unterscheidet:

- Der Programmcode eines Dauerauftrages wird zentral und proprietär verwaltet
- Die Transaktion kann unter bestimmten Bedingungen rückgängig gemacht werden
- Die Ausführung eines Smart-Contracts kann unterbunden werden

Eine Unterscheidung zwischen dezentralen und zentralen Smart-Contract-Realisierungen ist daher zwar grundsätzlich gegeben, jedoch finden sich beide Formen in jenen Eigenschaften wieder, als dass es sich dabei um selbstauslösende Verträge handelt „die den Abschluss und die Vollziehung von Rechtsgeschäften vollständig, autark und unmittelbar vornehmen können“ (Fries & Paal, 2019).

2.2.3 Einbindung in die Blockchain-Technologie

Ein Smart-Contract wird innerhalb eines dezentralen Netzwerks als eigenständige Teilnehmer angesehen, welcher sich von den herkömmlichen Nutzern/Nutzerinnen der Blockchain darin unterscheidet, dass er lediglich anhand seiner festgelegten Regeln Transaktionen durchführt. Wie bereits in Abbildung 6 grafisch dargestellt, interagieren die User des Netzwerkes bei der Ausführung eines Smart-Contracts nicht mit dem Vertragspartner/der Vertragspartnerin direkt, sondern über den jeweiligen Smart-Contract, welcher den Vertrag zwischen den beiden Parteien schlussendlich abwickelt und absichert. Dahingehend wird in der Ethereum-Blockchain zwischen

sogenannten „Contract-Accounts“ und den „User-Accounts“ unterschieden, wobei User ihre Transaktionen mittels privatem Schlüssel signieren und Contract-Accounts Transaktionen anhand einer vordefinierten Regel ausführen. Wie auch in Abbildung 7 grafisch dargestellt, ist es den Usern dadurch möglich, sowohl mit anderen User-Accounts als auch mit Contract-Accounts zu kommunizieren. Für Contract-Accounts existieren keine privaten Schlüssel, sodass diese Smart-Contracts im Nachhinein nicht korrigiert bzw. manipuliert werden können und daher eine Korrektur eines Smart-Contracs selbst nur möglich ist, indem ein neuer Smart-Contract kreiert wird (siehe auch Abbildung 7). (Wilkens, 2019)

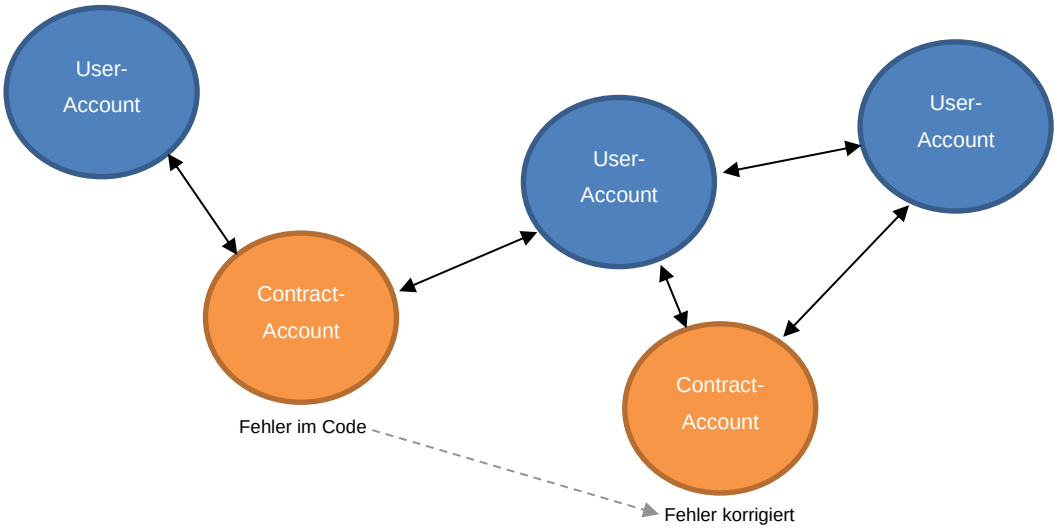

Abbildung 7: Kommunikation zwischen User- und Contract-Accounts (Quelle: eigene Darstellung)

Betrachtet man diese Netzwerkkommunikation unter Berücksichtigung, dass Fehler in Contract-Accounts nicht direkt behoben werden können, so ist ersichtlich, dass hierbei das Risiko besteht, dass User-Accounts mit fehlerhaften Contract-Accounts interagieren. Hierfür wurde eine Lösung zur indirekten Modifikation mittels sogenannter „Dispatcher“ geschaffen. Hierbei erfolgt eine Verlinkung von einem Contract-Account zum Dispatcher, welcher wiederum zu unterschiedlichen modifizierbaren Libraries verweist (siehe Abbildung *8*). Durch diesen Aufbau ist gewährleistet, dass Modifikationen zwar nicht auf einem einzelnen Smart-Contract durchgeführt werden können, aber jeder Smart-Contract über einen Dispatcher auf Libraries verweist, welche im Nachhinein Updates erhalten können. (Wilkens, 2019)

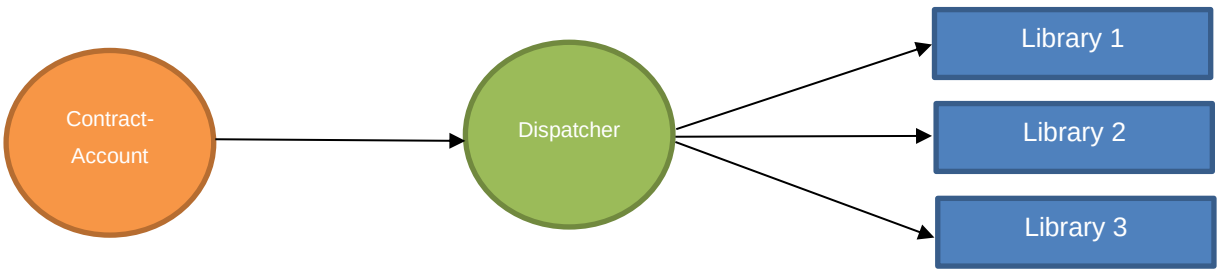

Abbildung 8: indirekte Modifikation von Contract-Accounts (in Anlehnung an Wilkens, 2019)

Eine neue Kreierung eines Update-Contract-Accounts ist durch diese Architektur nicht mehr notwendig und das Risiko einer Kommunikation zwischen einem User und einem fehlerhaft-identifizierten Contract-Account kann dadurch minimiert werden.

2.3 Non-Fungible-Token

NFT bieten ein breites Spektrum an Anwendungsmöglichkeiten, welchen die Möglichkeit zur Identifizierung von Einzigartigkeiten zu Grunde liegen. In den folgenden Unterkapiteln wird eine Definition für NFT erarbeitet, auf deren Aufbau und Funktionsweise im Zusammenhang mit der Blockchain-Technologie eingegangen und es werden bereits existierende Projekte betrachtet, sodass die Vielfalt der möglichen Anwendungsbereiche vermittelt werden können.

2.3.1 Definition

Im Duden finden sich zu dem Begriff „fungibel“ Synonyme wie „austauschbar“ oder „auswechselbar“ (Dudenredaktion, o.J.). Bargeld wird demnach als fungibel bezeichnet, denn jeder Bargeldschein ist in einen gleichwertigen Bargeldschein oder mehrere Bargeldscheine – welche in Summe denselben Wert widerspiegeln – austauschbar. Im Sinne der Kaufkraft macht es also keinen Unterschied, ob man einen 100 Euro-Schein gegen einen anderen 100 Euro-Schein oder zwei 50 Euro-Scheine tauscht. Auch die momentan bekannteste Kryptowährung Bitcoin kann demnach als fungibel angesehen werden, denn jeder dieser digitalen Coins hat denselben inneren Wert (Hoeren & Prinz, 2021). Im Gegensatz dazu kann schlussgefolgert werden, dass es sich bei nicht fungiblen Gütern um Werte handelt, die nicht 1:1 austauschbar sind. Hierbei handelt es sich zumeist um seltene Güter, für welche der Wert anhand ihrer Originalität bestimmt wird. Als Beispiel der analogen Welt können hier einzigartige Kunstwerke genannt werden. Ein Gemälde eines Künstlers/einer Künstlerin kann zwar in verschiedenster Art und Weise vervielfältigt werden, jedoch bleibt das zugrundeliegende, originale Kunstwerk einzigartig und kann dadurch nicht gegen eine Kopie (mit demselben Wertverhältnis) ausgetauscht werden.

In der digitalen Welt stellt die Replikation von digitalen Gütern und deren Nutzung für eigene Zwecke ein Problem in Hinsicht auf das Urheberrecht dar, für welches der Einsatz von NFT Abhilfe schaffen könnte. NFT stellen per Definition nicht austauschbare Token dar, mit deren Hilfe für digitale Güter eine sogenannte „Exklusivität“ kreiert werden kann. Anhand der Einzigartigkeit eines jeden NFT kann ein digitales Gut eindeutig einem Besitzer/einer Besitzerin zugeordnet werden. Betrachtet man wiederum ein Beispiel aus der Kunst, so kann der erschaffende Künstler/die erschaffende Künstlerin sein/ihr Kunstwerk mit einem NFT verknüpfen. Sollte der Künstler/die Künstlerin nun das digitale Kunstwerk verkaufen, so kann mittels Übertragung dieses NFT an den Käufer/die Käuferin auch der Besitz des Kunstwerkes digital übergeben werden. (Hoeren & Prinz, 2021) Die eindeutige Nachvollziehbarkeit der Rechteübertragung wird in Kombination mit der Blockchain-Technologie gewährleistet und wird nun im folgenden Unterkapitel erläutert.

2.3.2 Funktionsweise

Es wurde bereits auf die Sinnhaftigkeit und die Funktionsweise von Smart-Contracts eingegangen. In diesen smarten Verträgen wird auch der Besitz sowie der Handel von NFT abgebildet. Dabei speichert der Smart-Contract (wie auch in Abbildung 9 ersichtlich) einerseits die Zuordnung eines NFT zu einem Teilnehmer/einer Teilnehmerin des dezentralen Netzwerkes und andererseits einen Hashwert, welcher eindeutig auf die wichtigsten Metadaten eines digitalen Gutes in einem verteilten Dateisystem verweist. Diese Metadaten enthalten wiederum jene Informationen, wo dieses digitale Gut gespeichert ist. (Hoeren & Prinz, 2021)

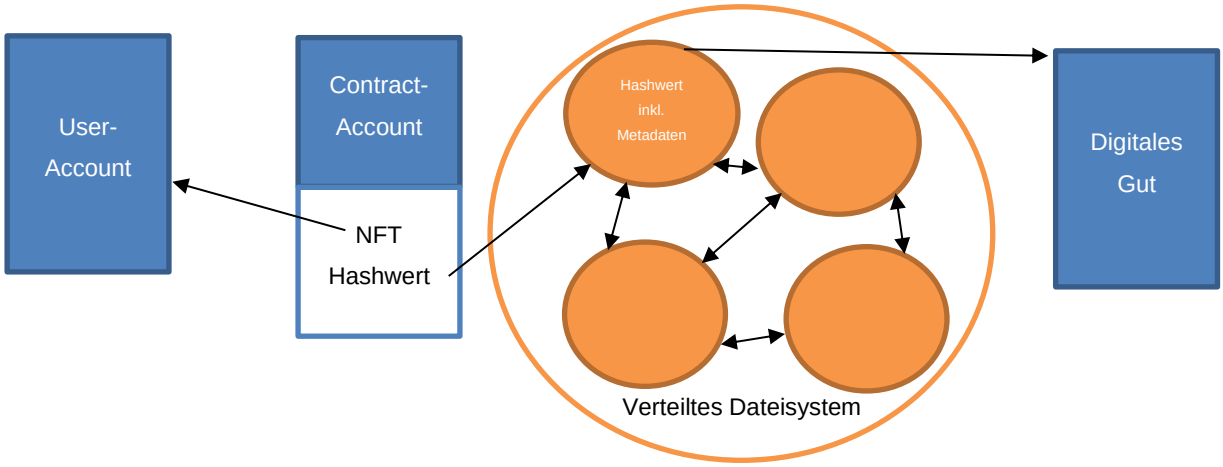

Abbildung 9: Funktionsweise von NFT (Quelle: eigene Darstellung)

Erfolgt nun ein Handel des digitalen Gutes, so wird lediglich das Recht am NFT mit den verknüpften Metadaten an den Käufer/die Käuferin übertragen, sodass der Smart-Contract schlussendlich den NFT dem neuen Besitzer/der neuen Besitzerin zuordnet (Hoeren & Prinz, 2021).

Eine solche Transaktion wird wiederum in der Blockchain dokumentiert und das gesamte Netzwerk kann unmissverständlich verifizieren, dass der Käufer/die Käuferin nun im Besitz dieses NFT ist. Da es anhand der Blockchain möglich ist, die gesamte Transaktionshistorie – bis zum Genesisblock – nachzuvollziehen, kann auch jeder Teilnehmer/jede Teilnehmerin des Netzwerkes einsehen, wer welchen NFT zu welchem Zeitpunkt besessen hat. Aufgrund dieser Eigenschaft kann auch eindeutig festgestellt werden, wer einen NFT erstmals kreiert hat, womit Kopien des digitalen Gutes eindeutig vom Original abgegrenzt werden können. Betrachtet man wiederum das Beispiel eines digitalen Kunstwerks, so kann dieses zwar immer noch repliziert werden, jedoch kann das Original anhand des Erstellers/der Erstellerin erkannt sowie jeder darauffolgende Handel des zugrundeliegenden NFT festgestellt und nachvollzogen werden.

2.3.3 Gegenwärtige Anwendungsbereiche

Ein paar Jahre nach der Entwicklung der Ethereum-Blockchain wurde das Potential in Hinsicht auf die digitale Darstellung von Exklusivität erkannt und es folgten zahlreiche Projekte, die sich mit dieser Thematik auseinandersetzten und damit der Blockchain-Community erstmals die Möglichkeit anboten, digitale, nicht-austauschbare Güter zu kreieren und zu handeln. Dabei wurden Ideen generiert, welche nicht nur neue Geschäftsfelder bedienen können, sondern auch neue Einkommensstrategien für bereits bestehende Märkte generierten. Folglich werden unterschiedliche Märkte anhand eines bestehenden Projektes oder einer Idee als Beispiel für das breite Spektrum an Anwendungsmöglichkeiten näher betrachtet und dahingehend analysiert, inwiefern eine solche – auf der Blockchain basierte – Einkommensstrategie einen Mehrwert für die Partizipanten schaffen kann.

Digitale Sammelstücke

Inspiriert von der Londoner Punk-Szene wurde im Jahr 2017 von Matt Hall und John Watkinson erstmals die Möglichkeit geschaffen, digitale Sammlerstücke in Form von 8-Bit-Pixelbildern mit dem Titel „CryptoPunks“ zu erwerben. Das Besondere an diesen Pixelbildern war, dass erstmals eine begrenzte Anzahl von 10.000 Bildern eindeutig auf der Blockchain nachvollzogen werden konnte, denn jedem CryptoPunk liegt ein NFT zu Grunde. Des Weiteren unterscheidet sich jeder CryptoPunk von allen anderen in seiner Optik, sodass ein Sammlerwert in Hinsicht auf die Seltenheit der unterschiedlichen Merkmale entstanden ist. Sowohl der Aspekt, dass jede Person einsehen kann, wer im Besitz eines einzelnen CryptoPunk ist und dieser Besitz nicht erschlichen werden kann, als auch die Tatsache, dass es sich dabei um die ersten offiziellen NFT-Sammelstücke handelt, hat einen wahren Hype für diese Pixelbilder ausgelöst, sodass diese CryptoPunks innerhalb eines Jahres (April 2020 – April 2021) im Durchschnitt zu einem Marktpreis von 30.400 USD gehandelt wurden. (Christie’s, 2021)

Digitale Kunst

Wie bereits im Einleitungskapitel erwähnt, schaffte es das um 69 Millionen USD verkaufte NFT-Kunstwerk mit dem Titel „Everydays: The First 5000 Days“ des Künstlers „Beeple“ in die weltweiten Schlagzeilen. Dieses Beispiel zeigt, dass es mittels NFT auch möglich ist, Einkünfte durch die eigenen künstlerischen Fähigkeiten digital zu erzielen. Da der zu Grunde liegende Code eines jeden Smart-Contracts beliebig definiert werden kann, bieten diese smarten Verträge im Zusammenhang mit NFT für digitale Kunst attraktive Ertragsmodelle. So kann beispielsweise ein NFT so gestaltet werden, dass der Ersteller/die Erstellerin eines NFT bei jedem weiteren Handel am erzielten Gewinn profitieren kann (CV Publishing AG, 2021). Ein Musiker/eine Musikerin könnte demnach den Besitz eines Musikstücks anhand eines NFT abbilden und damit nicht nur am eigenen (Erst-) Verkauf profitieren, sondern auch an allen darauffolgenden Verkäufen. Dadurch ist es einerseits den Käufern/Käuferinnen erstmals möglich, Musikstücke ihrer Idole für sich zu beanspruchen und andererseits können die Musiker/Musikerinnen eine zusätzliche Einkommensquelle generieren.

Gaming

In Anlehnung zu den bereits beschriebenen digitalen Sammelstücken, können diese auch in beliebten Videospielen eine tragende Rolle spielen. In herkömmlichen Videospielen werden jegliche Gegenstände, Charaktere, etc. zwar von den Spielern/Spielerinnen gefunden und/oder erworben, jedoch gab es bisher keine Möglichkeit, deren Einzigartigkeit und einen möglichen Sammlerwert zu bestimmen. Mittels NFT könnten hierbei die erworbenen Gegenstände und Charaktere eindeutig den Spielern/Spielerinnen zugeordnet werden und somit dem Spiel einerseits ein gewisses Maß an Selbstbestimmung in Hinsicht auf die Nutzung dieser Items zugeschrieben werden und andererseits könnten die Spieler/Spielerinnen Handel mit den erworbenen NFT betreiben. Bedenkt man, dass der weltweite Umsatz des Gaming-Markts im Jahr 2021 auf ca. 128 Milliarden EUR Umsatz geschätzt und ein jährliches Wachstum von etwa 8,6 % erwartet wird (Statista, 2021a), so kann auch erahnt werden, dass ein Handel von Gaming-NFT ein attraktives Ertragskonzept darstellen könnte.

Metaversum

Nicht zuletzt hat das Rebranding von Facebook in „Meta“ großes Aufsehen für das Metaversum gesorgt. Als Metaversum versteht man ein virtuelles Universum, welches nicht an die reale Welt gekoppelt ist. Die Vision eines solchen digitalen Universums ist es, eine virtuelle Umgebung für die Menschheit mit eigenem Wirtschaftssystem und eigener Währung zu schaffen. Jede Person soll sich schließlich innerhalb des Metaversums digital frei bewegen können und sich somit eine zweite digitale Existenz aufbauen können. (Eisenbrand, 2020) Grundsätzlich mag eine solche „Zwischenwelt“ zwar utopisch klingen, doch einige Projekte arbeiten bereits an der Umsetzung dieser Vision. NFT fungieren dabei als Sicherstellung von Besitztümern wie digitalen Grundstücken, Gebäuden oder Gegenständen, wobei diese Besitztümer wiederum öffentlich gehandelt werden können (Digital World LTD, 2021).

Die eben genannten Beispiele dienen zur Veranschaulichung der bereits umgesetzten Möglichkeiten von NFT. Grundsätzlich kann anhand dieser Beispiele angenommen werden, dass NFT heutzutage noch primär als Handelsprodukt angesehen werden, jedoch bieten NFT in Kombination mit Smart-Contracts und der Blockchain-Technologie – aufgrund ihrer Eigenschaft zur eindeutigen Identifikation von Fälschungen – noch weitere Anwendungsgebiete, welche nicht auf den reinen Handel dieser nicht-austauschbaren Token ausgerichtet sind. Um jenen Anwendungsbereich dieser Arbeit, nämlich die Schaffung und den Nachweis von identitätsbezogenen Bescheinigungen, konzipieren und prototypisch umsetzen zu können, bedarf es vorerst noch einer Aufarbeitung dahingehend, was der Begriff „selbstbestimmte Identität“ bedeutet und was darunter konkret im Zusammenhang mit der fortschreitenden Digitalisierung verstanden wird.

2.4 Selbstbestimmte, digitale Identität

In diesem Kapitel werden die Begriffe „Identität“, „digitale Identität“ sowie „Selbstbestimmtheit“ im thematischen Zusammenhang aufgearbeitet, um einerseits ein Verständnis dahingehend aufzubauen, was unter diesen Begrifflichkeiten verstanden wird und andererseits, um die Relevanz dieser Thematik – vor allem in Sinne des Schutzes der eigenen Identität – in den Vordergrund zu rücken.

2.4.1 Identität im allgemeinen Sinne

Grundsätzlich handelt es sich bei einer Identität um eine Eigenschaft, die eine eindeutige Identifizierung einer Person gewährleistet. Diese Eigenschaft ist einerseits für außenstehende, dritte Personen erkennbar und andererseits ist es auch für eine einzelne Person möglich, sich selbst von anderen Menschen abzugrenzen, unabhängig davon, ob sich äußere Einflussfaktoren wie beispielsweise die Zeit oder die Situation, in welcher man sich befindet, verändert. Doch die Identität eines Menschen wird nicht als absolut und statisch angesehen, denn hinter dieser Begrifflichkeit werden auch alle Eigenschaften und Fähigkeiten verstanden, die man sich im Laufe des Lebens angeeignet hat, womit sie sich in diesem Sinne auch stets weiterentwickelt und dadurch als lebenslanger Prozess betrachtet werden kann. Die eigene Identität spiegelt sich wiederum in Teil-Identitäten wider, welche sich dadurch auszeichnen, dass sich jede Teil-Identität einer Person nicht unbedingt von Teil-Identitäten ihrer Mitmenschen unterscheidet, jedoch die Kombination der einzelnen Teil-Identitäten eine eindeutige Abgrenzung zu anderen Identitäten ermöglicht, was sich schlussendlich in der Individualität eines jeden Menschen manifestiert. Solche Teil-Identitäten können sowohl körperliche Eigenschaften wie beispielsweise die Hautfarbe oder die Größe als auch Fähigkeiten und Gefühle einer jeweiligen Person sein. Um die Identitäten aus staatlicher Perspektive feststellen und unterscheiden zu können, kommen hierfür Ausweis-Identitäten zum Einsatz. Diese dokumentieren die verschiedenen körperlichen Attribute einer Person und ermöglichen dadurch eine Unterscheidung zwischen verschiedenen Individuen. (Kunze, 2003)

2.4.2 Digitale Identität im digitalen Zeitalter

In der digitalen Welt kann eine Identität, wie sie im vorherigen Unterkapitel beschrieben wurde, nicht 1:1 abgebildet werden, denn wie auch bei Ausweis-Identitäten ist es auch hier nicht möglich, alle Attribute, Fähigkeiten sowie Gefühle einer Person lückenlos festzuhalten. Ziel dabei ist auch hier, dass ein möglichst genauer Nachweis zur Identifikation der Identität gewährleistet wird. Dieser Nachweis kann dabei anhand der Überprüfung von drei unterschiedlichen Merkmalen gewährleistet werden (Hornung, 2005):

- Besitz; beispielsweise Personalausweis oder Chipkarte
- Wissen; beispielsweise ein Passwort oder PIN-Code
- Sein; beispielsweise ein Fingerabdruck

Im Sinne einer digitalen Identität wird keine alleinstehende Identität verstanden, sondern eine Abbildung unterschiedlicher Informationen, welche sich einem Individuum möglichst eindeutig zuordnen und anhand einer oder mehrerer der genannten Überprüfungsmöglichkeiten nachweisen lassen. Es werden also Teil-Identitäten abgebildet, wobei die Anzahl an unterschiedlichen Teil-Identitäten je nach Bedarf variieren kann (Hansen & Meints, 2006). In einem Unternehmen kann beispielsweise die digitale Identität anhand einer Personenkennzahl erfolgen, welche eindeutig einer einzelnen Person zugeordnet werden kann. Der Nachweis erfolgt dabei durch die Überprüfung des Wissens mittels Passwort beim Login am Firmen-Rechner.

Eine digitale Identität weist einen Lebenszyklus auf, welcher folgende Phasen durchläuft (Stephan et al., 2018):

- Als erstes werden die digitalen Identitäten erzeugt. Dabei werden einerseits die Teil-Identitäten von einer Person oder Institution geprüft bzw. validiert und andererseits werden diese Informationen eindeutig einem Individuum zugeordnet.
- Im täglichen Gebrauch wird die digitale Identität verwaltet und gepflegt, sodass Änderungen der zugrundeliegenden Identität auch digital abgebildet werden.
- Wird eine digitale Identität nicht mehr benötigt, so wird diese schlussendlich gelöscht oder archiviert, unabhängig davon, ob die zugrundeliegende Identität noch existiert oder nicht.

2.4.3 Selbstbestimmung über die eigene digitale Identität

Unter dem Begriff „Selbstbestimmung“ versteht man die Freiheit über die eigene Identität sowie die daraus resultierende Eigenverantwortung für sein eigenes Tun und Handeln (Mai, 2021). Jeder Mensch unterliegt somit gewissen Rechten und Pflichten, die ein gemeinsames Miteinander gewährleisten und damit zur Entfaltung der eigenen Identität beitragen. Bezieht man diese Definition der Selbstbestimmung auf das eigene Tun und Handeln in der digitalen Welt, so kann auch hierbei die Eigenverantwortung für sich selbst Anwendung finden, denn auch im Internet gibt es gewisse Regeln, an die sich jeder Teilnehmer/jede Teilnehmerin halten muss. Da die Selbstbestimmung mit der eigenen Identität einhergeht und sich in den Teil-Identitäten wiederum körperliche Eigenschaften sowie Fähigkeiten und Gefühle der einzelnen Individuen wiederfinden, kann daraus auch geschlussfolgert werden, dass jedes Individuum ein Recht auf die Verwaltung dieser identitätsbezogenen Informationen im Sinne der Selbstbestimmung hat. Demnach soll es also nicht möglich sein, dass fremde Parteien – ohne vorheriger Einwilligungserklärung – die Möglichkeit haben, die identitätsbezogenen Informationen ihrer Mitmenschen weiterzugeben oder anderwärtig zu verwenden.

Selbstbestimmte digitale Identität kann demnach so verstanden werden, dass jedes Individuum nicht nur ein Recht darauf hat, die eigene digitale Identität zu schaffen, sondern auch, dass man die uneingeschränkte Kontrolle über die in ihr enthaltenen Informationen haben sollte und für sich selbst entscheiden kann, an wen diese Informationen weitergegeben werden.

2.4.4 Kriminalität

Laut Statista wurden bereits 14 Prozent aller Österreicher/Österreicherinnen Opfer eines Identitätsdiebstahls (Statista, 2020). Der tatsächliche Wert dürfte jedoch höher sein, da ein Identitätsdiebstahl erst dann bemerkt werden kann, wenn dieser auch missbräuchlich verwendet wird und/oder diese Person auf den Diebstahl aufmerksam wird. Angreifer – die sich diese Kriminalität zu Nutze machen – können sich so als eine andere Person ausgeben und sich dadurch in ihrer eigenen Anonymität in Sicherheit wähnen.

Es wurde bereits darauf eingegangen, dass eine digitale Identität anhand unterschiedlicher Merkmale nachgewiesen wird. Problematisch wird dies, sobald diese Nachweisinformationen in fremde Hände gelangen, denn auf diese Art und Weise kann sich eine Person beispielsweise mittels Nachweis eines fremden Passworts auch als diese fremde Person ausgeben. Vor allem aufgrund der fortschreitenden, weltweiten Vernetzung und der dabei verwendeten Anmeldemechanismen einiger weniger Social-Media-Anbieter kann dies ein attraktives Angriffsziel für Identitätsmissbrauch darstellen. Betrachtet man beispielsweise die Tatsache, dass es nahezu auf jeder Plattform die Möglichkeit gibt, sich über seinen Facebook-Account anzumelden, so kann durch das Wissen eines einzigen Passworts – nämlich jenes für die Facebook-Anmeldung – der Identitätsnachweis für alle Plattformen erbracht werden, welchen die jeweiligen Facebook-Zugangsdaten zu Grunde liegen.

2.5 Bestehende Projekte

Das wohl bekannteste Projekt in Hinsicht auf die Sicherstellung digitaler Identitäten ist das „Identity Overlay Network“ (ION) von Microsoft. Auch hier sieht man die Problematik, dass mit fortschreitender Digitalisierung die Notwendigkeit für eine fälschungssichere, digitale Abbildung der eigenen Identität immer mehr an Bedeutung gewinnt. ION stellt dabei eine zweite Ebene auf der Bitcoin-Blockchain zur Verfügung, welche es erlaubt, digitale Identifikatoren zu kreieren und einer Person eindeutig zuzuweisen. Dabei werden die Eigenschaften zur Dezentralisierung anhand der Bitcoin-Blockchain als erste Ebene herangezogen, indem die durchgeführten Transaktionen im ION laufend an die Blockchain-Historie der Bitcoin-Blockchain verankert werden (Buchner, 2019). Der Hauptfokus liegt hierbei auf der Verifizierung von Zugangsdaten, sodass sich Menschen nicht mittels herkömmlichen Login-Verfahren identifizieren (und damit ihre persönlichen Daten bekannt geben) müssen, sondern die Verifizierung erfolgt anhand dezentraler Identifikatoren, welche den Nachweis über die Identität des Users liefert (Brutkasten Media GmbH, 2021). Beispielsweise wäre es dadurch für einen Serviceanbieter möglich, einem User die Zugangsdaten als dezentralen Identifikator zu übermitteln, welcher über das ION und die Blockchain eindeutig dem User zugeordnet ist, sodass sich der User folglich nur mehr mit diesem dezentralen Identifikator identifizieren muss. Der in dieser Arbeit ausgearbeitete Ansatz zur Sicherstellung von selbstbestimmten, digitalen Identitäten, legt den Fokus (neben der fälschungssicheren Verifizierung dieser Identitäten) auf die Zuordnung von identitätsbezogenen Nachweisen von Behörden oder Unternehmen und der notwendigen Nachweisbarkeit der Originalität dieser Nachweise.

Auch die Europäische Kommission arbeitet bereits an einer dezentralen Infrastruktur, welche unter anderem zur Sicherstellung der eignen digitalen Identität dienen soll. Dabei stellt die European-Blockchain-Services-Infrastructure (EBSI) ein Netzwerk von verteilten Nodes zur Verfügung, welche für ausgewählte Anwendungsgebiete zum Einsatz kommen soll. (Europäische Kommission, 2022)

> *EBSI is a joint initiative from the European Commission and the European Blockchain Partnership. The vision is to leverage blockchain to accelerate the creation of cross-border services for public administrations and their ecosystems to verify information and to make services more trustworthy. (Europäische Kommission, o. J.b)*

Für die Umsetzung dieses Vorhabens bedient sich die EBSI der Konsensfindung anhand eines Autoritätsnachweises. Dabei werden die durchgeführten Transaktionen nicht (wie bei PoW oder PoS) von einer Vielzahl von Minern oder Stakern verifiziert, sondern dieser Vorgang wird von einer kleinen Anzahl von berechtigten Akteuren durchgeführt (Neue Tageskrypto, 2020). Im Falle der EBSI entsprechen diese Akteure je einem Node pro EU-Mitgliedsstaat, was die Konsensfindung einerseits zwar zentraler gestaltet als bei PoW und PoS, jedoch entstehen dadurch geringere Kosten und der Stromverbrauch kann gesenkt werden. Wie auch beim ION-Projekt können innerhalb der EBSI digitale Identifikatoren erstellt und einer Person eindeutig zugewiesen werden. (Doerk, 2020) Die selbstbestimmte Identität wird dabei sowohl für natürliche als auch für juristische Personen sichergestellt und dabei von den Identitäten selbst kontrolliert und verwaltet. Das zugrundeliegende Framework soll auch eine Digitalisierung von Diplomen und Genehmigungen – wie beispielsweise Führerscheine – ermöglichen und somit zu einer fortschreitenden Digitalisierung der eigenen Identität führen. (Europäische Kommission, o. J.a)

3 ANGEWANDTE METHODIK UND VERWENDETE TOOLS

In diesem Kapitel wird auf die für diese Arbeit gewählte Methodik zur Beantwortung der definierten Forschungsfrage eingegangen. Dabei wird zuerst die Entscheidungsgrundlage für die angewandte, empirische Forschungsmethodik erläutert, sodass der grundlegende Zweck des Prototyps für diese Arbeit in den Vordergrund gestellt wird. Darauf folgt eine theoretische Einführung in die wichtigsten Aspekte dieser Forschungsmethode sowie eine kurze Erläuterung der Grundzüge von Web-Applikationen. Folglich werden in diesem Kapitel die für den Prototyp verwendeten Frameworks, Bibliotheken und Tools näher betrachtet, sodass sich ein mögliches Setup für den Verwendungszweck der Umsetzung von identitätsbezogenen Nachweisen mittels Blockchain als technologische Grundlage erkennen lässt. Mit dem theoretischen Konstrukt zum Aufbau und zur Funktionsweise des zu entwickelnden Prototyps wird das Kapitel abgeschlossen.

3.1 Entscheidungsgrundlage für angewandte Methodik

Die angewandte Methodik dient der Beantwortung der in Kapitel 1.2 gestellten Forschungsfrage "Welche Herausforderungen birgt die Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen?“ und stellt somit den empirischen Teil dieser Arbeit dar. Durch die bereits beschriebenen theoretischen Grundlagen in Hinsicht auf die Blockchain-Technologie, Smart-Contracts, NFT und der Definition einer selbstbestimmten digitalen Identität, wurde bereits das notwendige Know-How generiert, um Anforderungen an eine solche Webanwendung sowie den zugrundeliegenden Smart-Contract zu modellieren. Diese Anforderungen stützen sich somit auf Erkenntnisse unterschiedlicher Quellen in Form von – an die Thematik gerichtete – Literatur, wobei im empirischen Teil nicht nur die Möglichkeit zur praktischen Umsetzung dieser Anforderungen bewiesen werden soll, sondern es sollen vor allem jene Herausforderungen erhoben werden, welche sich durch reine Literaturrecherche – aufgrund der speziellen Anwendungsform der Blockchain-Technologie in Kombination mit NFT im Rahmen dieser wissenschaftlichen Arbeit – nicht oder nur schwierig erkennen lassen. Es sollen dabei qualitative Aussagen über mögliche, noch nicht gelöste Probleme dieser Anwendungsform der Blockchain-Technologie untersucht werden, sodass sich hierfür quantitative Forschungsmethoden als nicht praktikabel herausstellen. Wie auch in der Forschungsfrage definiert, richten sich die erwarteten Herausforderungen an die praktische Umsetzung einer Webanwendung. Diese Herausforderungen einer praktischen Umsetzung können nun entweder durch Interviews mit Experten/Expertinnen in diesem Fachgebiet erhoben oder anhand eigener Erfahrungen dargelegt werden. Da sich die Thematik dieser wissenschaftlichen Arbeit auf ein neuartiges Anwendungsgebiet der Blockchain-Technologie bezieht, wurde entschieden, die Herausforderungen dieser konkreten Idee der Umsetzung einer Blockchain-Anwendung zum Nachweis von Bescheinigungen anhand eines Prototyps zu ermitteln und Lösungsvorschläge anhand der gewonnenen Erkenntnisse zu definieren.

3.2 Das Prototyping

Es wurde bereits geklärt, warum für die Beantwortung der definierten Forschungsfrage die Methodik des Prototypings angewandt wird. Folglich wird – neben der allgemeinen Definition eines Prototyps – der für diese Arbeit entwickelte Prototyp anhand seiner Art und seines Musters eingegliedert und es wird darauf eingegangen, wie der Prototyp im weiteren Verlauf dieser Arbeit evaluiert werden soll.

3.2.1 Definition

Prototyping kann per se als Prozess angesehen werden, welcher sich entweder als Bestandteil eines Softwareentwicklungsprozesses darstellt oder als Ansatz, der den Lebenszyklus dieses Softwareentwicklungsprozess beeinflusst. Dabei dient das Prototyping dazu, komplexe Anforderungen in kleine Anforderungspakete zusammenzufassen und somit die anfängliche Komplexität der Gesamtanforderung strukturell zu minimieren. Durch das Prototyping lassen sich also Lösungsansätze für ein Problem in Teilprobleme untergliedern, sodass man sich der Umsetzung der definierten Anforderungen in mehreren Zyklen nähern kann. (Carr & Verner, 1997) Des Weiteren dienen Prototypen einer Konsensfindung zwischen Nutzern/Nutzerinnen und den Entwicklern/Entwicklerinnen, denn ein Prototyp lässt in einem frühen Stadium des Softwareentwicklungsprozesses erkennen, ob und in welcher Form die gestellten Anforderungen umsetzbar sind (Kuhrmann, 2012). Im Rahmen des Prototypings „wird eine Vorabversion eines Anwendungssystems entwickelt und evaluiert. Beide Schritte können neue Erkenntnisse generieren“ (Wilde & Hess, 2007), sodass für den weiteren Softwareentwicklungsprozess die notwendigen Informationen in Hinsicht auf die Umsetzbarkeit gesammelt werden können. Ein Prototyp kann somit als Mittel zum Zweck gesehen werden, um innerhalb eines definierten Systems anhand erster Realisierungen lernen zu können (Schork, 2020).

3.2.2 Eingliederung des Prototyps nach Art und Muster

Grundsätzlich werden Prototypen in drei gebräuchliche Arten und zwei unterschiedliche Muster untergliedert (Kuhrmann, 2012):

Arten

- Demonstratoren: Geben einen Anhaltspunkt, in welche Richtung sich eine Software entwickeln wird. Sie werden in einer frühen Phase des Softwareentwicklungsprozesses genutzt und sind daher noch weit vom tatsächlichen Endprodukt entfernt.
- Labormuster: Geben Auskunft über technische Fragestellungen und die damit verbundene Umsetzungsmöglichkeit einer vorab definierten Architektur. Anwender/Anwenderinnen werden in dieser Art des Prototypings grundsätzlich nicht miteingebunden.
- Pilotsysteme: Bilden eine Form der Software ab, welche sich bereits in einer späten Entwicklungsphase befinden. Dabei werden Anwender/Anwenderinnen in die Tests

miteinbezogen, sodass Feedback aus der Sicht der Kunden/Kundinnen generiert werden kann.

Muster

- Horizontaler Prototyp: Dabei wird lediglich ein definierter Bereich einer Software umgesetzt. Zumeist handelt es sich dabei um das GUI, sodass den Anwendern/Anwenderinnen bereits eine grafische Darstellung der umzusetzenden Software angeboten werden kann.
- Vertikaler Prototyp: Dabei werden ausgewählte Funktionalitäten in ihrer Gesamtheit umgesetzt. Es werden also neben dem GUI auch sämtliche Schichten zur Gänze implementiert, sodass definierte Aspekte von der Datenspeicherung bis zur grafischen Benutzeroberfläche präsentiert werden können.

Betrachtet man diese theoretische Eingliederung von Prototypen genauer und wendet diese auf den in dieser Arbeit definierten Zweck – einer Ableitung von Herausforderungen anhand eines praktisch-umgesetzten Prototyps – an, so wird ersichtlich, dass es sich dabei um einen vertikalen Labormuster-Prototyp handelt. Einerseits sollen anhand des resultierenden Softwarestücks technische Fragestellungen in Hinsicht auf die möglichen Herausforderungen bei der Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Nachweisen beantwortet werden und andererseits sollen für diesen Zweck sämtliche Aspekte zur Umsetzung einer solchen Webanwendung betrachtet werden. Wie in Abbildung 10 grafisch begleitet, soll der Prototyp neben dem Graphical User Interface (GUI) zur grafischen Darstellung der Benutzeroberfläche und Interaktion der Anwender/Anwenderinnen, auch den benötigten Smart-Contract zur Implementierung und Abgrenzung der Funktionalitäten sowie eine Schnittstelle zwischen (Test-) Blockchain und Smart-Contract aufweisen.

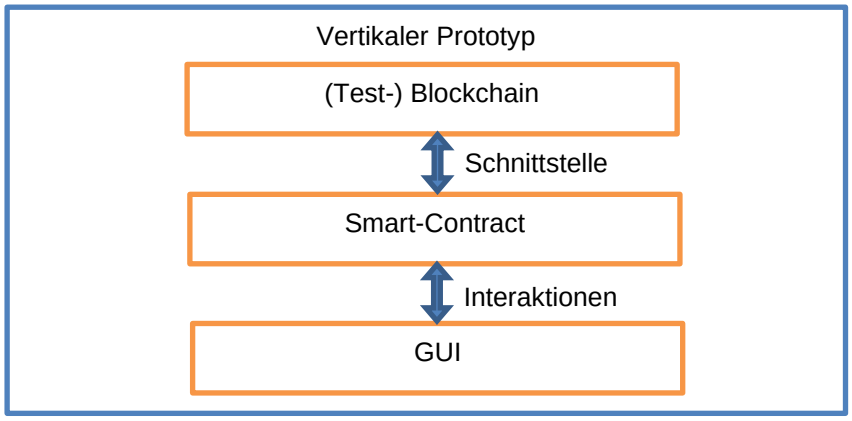

Abbildung 10: Vertikaler Prototyp (Quelle: eigene Darstellung)

3.2.3 Evaluierung von Prototypen

Im Rahmen dieser Arbeit soll nicht nur ein Prototyp zum dezentralen Nachweis identitätsbezogener Bescheinigungen entwickelt werden, sondern dieser soll in der anschließenden Diskussion auch evaluiert werden. Der Begriff Evaluation kann interpretiert werden als „ein systematisches Sammeln, Auswerten und Interpretieren von Daten, um eine reliable und valide Bewertung der Benutzungsschnittstelle zu ermöglichen. Dabei wird aus den Ergebnissen der Evaluation abgeleitet, ob ein vorab definiertes Designziel erreicht ist bzw. ob und wo weitere Verbesserungsmöglichkeiten ausgeschöpft werden können." (Hegner, 2003) In Hinsicht auf diese Arbeit bedeutet dies, dass der Prototyp dahingehend evaluiert wird, ob die definierten User-Stories umgesetzt werden konnten und daraus werden jene Herausforderungen abgeleitet, die zur Beantwortung der gestellten Forschungsfrage dienen. Dabei kann grundsätzlich zwischen zwei verschiedenen Arten von Evaluationen unterschieden werden (Hegner, 2003):

- Die formative Evaluation beurteilt die zu entwickelnde Software im Rahmen ihres Entwicklungsprozesses. Das Ziel ist hierbei auf die Usability ausgerichtet, sodass die Qualität der einfachen Handhabung bereits im Entwicklungsprozess laufend verbessert werden kann. In der Regel werden dafür sogenannte Usability-Tests wie beispielsweise die Think-Aloud-Methode angewandt.
- Die summative Evaluation legt den Fokus auf die vorab definierten Evaluationskriterien, wobei die Einhaltung dieser überprüft wird. Aus dieser Evaluation kann anschließend ein Vergleich zu alternativen Umsetzungsvarianten erfolgen. Diese Art der Evaluation überprüft, ob theoretische Annahmen auch tatsächlich so umgesetzt werden können.

Die Evaluierung des in dieser Arbeit entwickelten Prototyps kann somit der summativen Evaluation zugeordnet werden. Die User-Stories dienen als Evaluationskriterien, für welche die Einhaltung/Umsetzung im Rahmen der Diskussion überprüft wird. Aus den daraus abgeleiteten Umsetzungs-Herausforderungen können Lösungsvarianten geschaffen werden, welche in weiteren Forschungsarbeiten analysiert werden können.

Evaluiert wird der Prototyp schlussendlich anhand folgender vier Schritte. Im ersten Schritt werden Wertekriterien definiert, welche eine Aussage darüber geben, welche Kriterien der Prototyp erfüllen muss, um der Definition einer selbstbestimmten, digitalen Identität gerecht zu werden. Anschließend werden Leistungsstandards definiert, welche eine Beurteilung dahingehend ermöglichen, ob die definierten Wertekriterien erfüllt wurden. Der Prototyp wird anschließend anhand der Wertekriterien auf die Einhaltung der Leistungsstandards gemessen und als letzter Schritt wird ein abschließendes Werturteil gefällt. (Hegner, 2003)

Die Resultate dieser Evaluierung geben schlussendlich Auskunft darüber, welche Wertekriterien anhand der definierten Leistungsstandards nicht ausreichend erfüllt werden konnten und zeigen damit erste Herausforderungen auf, die zur Beantwortung der Forschungsfrage dienen.

3.3 Die Web-Applikation

In dieser Arbeit wird ein Blockchain-basierter Prototyp zum Nachweis identitätsbezogener Nachweise entwickelt. Die Blockchain dient dabei als dezentrale Datenbank, welche einerseits die benötigten Informationen über die Nachweise selbst speichert und andererseits Auskunft darüber gibt, welche Nachweise welchem Benutzer/welcher Benutzerin zugeordnet sind. Da die Blockchain öffentlich einsehbar ist, kann auch jeder Teilnehmer/jede Teilnehmerin diese Informationen direkt von der Blockchain abrufen. Für eine vereinfachte Usability wird hierfür jedoch eine Webanwendung zur Verfügung gestellt, welche diese Informationen auf die User angepasst aufbereitet und grafisch im Webbrowser anzeigt. In diesem Unterkapitel wird eine Definition von Webanwendungen erarbeitet und es erfolgt eine Eingliederung des Prototyps im Sinne der zugrundeliegenden Webanwendung.

3.3.1 Definition

Webanwendungen (oder auch Web-Applikationen) sind Software, welche unabhängig vom Betriebssystem über jeden Browser aufgerufen werden können (Weyers, 2021). Es bedarf daher keiner lokalen Installation der Software am eigenen Rechner, sondern man benötigt lediglich einen Browser, der das GUI und die bereitgestellten Funktionen über eine Internetanbindung abruft.

> *Eine Webanwendung ist eine Website, deren Seiteninhalt noch nicht oder nur zum Teil festgelegt ist. Der endgültige Inhalt einer Seite wird erst festgelegt, wenn der Besucher die Seite vom Webserver anfordert. Da sich der endgültige Inhalt der Seite je nach den Aktionen des Besuchers von Anforderung zu Anforderung ändern kann, wird eine solche Seite als dynamische Seite bezeichnet. (Adobe Inc., 2021)*

Eine Webanwendung kann daher individuell auf deren Anwender/Anwenderinnen zugeschnitten und dynamisch an deren Anforderungen angepasst werden. Anders als bei statischen Webseiten, wo der zugrundeliegende Hypertext-Markup-Language(HTML)-Code für die Struktur der Inhalte fix hinterlegt ist und sich während der Verwendung nicht mehr verändert, kann sich die Struktur von Webanwendungen mit der Interaktion der Anwender/Anwenderinnen laufend verändern.

3.3.2 Eingliederung des Prototyps

Bei klassischen Webseiten wird seitens des Webbrowsers lediglich ein statischer Inhalt von einem Webserver angefordert und der Webserver retourniert schließlich diese Inhalte ohne vorherige, individuelle Anpassungen an den Webbrowser. Bei dynamischen Webanwendungen hingegen reagiert ein Anwendungsserver auf die Anforderung des Webbrowsers, sodass dieser die übermittelten Daten verarbeiten kann und einen auf die Anforderung angepassten Inhalt an den Webserver retourniert. Der Webserver retourniert diese Inhalte wiederum an den Webbrowser, wobei sich diese Inhalte aus der Sicht der Anwender/Anwenderinnen von jenen Inhalten statischer Webseiten nicht unterscheiden. (Adobe Inc., 2021)

Diese Unterscheidung berücksichtigend, kann dies in Hinsicht auf den zu entwickelnden Prototyp anhand eines Beispiels verdeutlicht werden. Der Prototyp sollte zwischen Privatpersonen und seriösen Unternehmen/Behörden unterscheiden und demnach – anhand der übermittelten Informationen in Form einer Ethereum Wallet-Adresse (Login-Informationen) – unterschiedliche Inhalte zur Verfügung stellen. Als seriöses Unternehmen/seriöse Behörde ist es notwendig, identitätsbezogene Nachweise erstellen zu können, wohingegen eine Privatperson lediglich ihre eigenen Nachweise grafisch dargestellt bekommen sollte. Durch die Verwendung einer dynamischen Webapplikation werden die Webseite und die Login-Informationen vom Webserver verarbeitet und dem Anwender/der Anwenderin wird die gewünschte Webseite mit deren individuellen Inhalten retourniert.

Bei einer Multi-Page-Applikationen (MPA) werden Benutzer/Benutzerinnen nach speziellen Interaktionen auf weitere Unterseiten der Webanwendung weitergeleitet, was nicht nur den Vorteil bietet, dass die Webapplikation aufgrund ihrer Erweiterung über zusätzliche Unterseiten einfach skalierbar ist, sondern die Navigation durch die Webapplikation auch nachvollziehbar in der Uniform-Resource-Locator (URL) dargestellt wird. Eine Single-Page-Applikation (SPA) zeichnet sich dadurch aus, dass sämtliche Inhalte einer Webanwendung auf nur einer Seite angezeigt werden. Diese Inhalte unterscheiden sich zwar anhand der User-Interaktionen, benötigen jedoch keine Navigation durch weitere Unterseiten, sodass die URL nach jeder Interaktion dieselbe bleibt. Diese SPA bietet sich vor allem für jene Webanwendungen an, welche eine geringe Anzahl an Funktionen anbietet und gilt durch die Darstellung auf nur einer Seite performanter als MPA. (Lvivity LLC., 2020)

Da der Prototyp lediglich jene Funktionen anbieten soll, die der Smart-Contract beinhaltet und die Herausforderungen aufzeigen soll, die bei einer Umsetzung einer Blockchain-Anwendung zum Nachweis identitätsbezogener Bescheinigungen auftreten können, wird hierfür die Form einer SPA herangezogen. Die Webanwendung verfügt somit lediglich über eine Seite (und somit eine einzige URL), welche die angezeigten Informationen anhand der Login-Informationen der Anwender/Anwenderinnen bereitstellt.

3.4 Verwendete Tools und Programmiersprachen

In diesem Unterkapitel werden jene Frameworks, Bibliotheken und Tools aufgezeigt, die im Rahmen des Entwicklungsprozesses verwendet werden, um nicht nur die benötigten Funktionalitäten implementieren zu können, sondern auch im GUI eine strukturierte Erstversion zu realisieren. Der Fokus bei der Entwicklung des Prototyps liegt jedoch hauptsächlich auf der Implementierung der technischen Funktionalitäten, da es sich – wie bereits in Kapitel 3.2.2 aufgezeigt – um ein Labormuster handelt und dabei noch keine Anwender/Anwenderinnen für etwaige Usability-Tests miteinbezogen werden. Für weitere Forschungsarbeiten zu dieser Thematik wird angemerkt, dass die nachfolgend angeführten verwendeten Hilfsmittel lediglich Möglichkeiten darstellen, wie eine bestimmte Funktionalität umgesetzt werden kann, jedoch für bestimmte Aspekte auch alternative Tools zur Verfügung stehen.

3.4.1 Bootstrap

Bootstrap wurde im Jahr 2011 von Twitter ins Leben gerufen und veröffentlicht. Grundsätzlich handelt es sich dabei um ein Framework, welches HTML und Cascading-Style-Sheets(CSS)-Konventionen zur Verfügung stellt, um bereits optisch-vordefinierte Designmuster ressourcenschonend in die GUI-Umgebung integrieren zu können. Bootstrap basiert auf Less – eine Stylesheet-Sprache – welche durch einen flexiblen Präprozessoperator mehr Flexibilität und Leistung bietet als herkömmliches CSS. Das Kompilieren von Less kann dabei unter anderem mittels Node.js erfolgen. Das Framework ist dabei auf sieben Less-Dateien aufgeteilt, welche jeweils ein Designmuster für bestimmte Komponenten bereitstellen. Wurden diese Less-Dateien einmal kompiliert, so wird für die weitere Verwendung lediglich eine einzige CSS-Datei benötigt. (Otto, 2011) Als wesentliche Vorteile von Bootstrap gelten einerseits das Responsive-Design, welches Bootstrap bereits integriert und damit die Kompatibilität mit Smartphones, Tablets, etc. sicherstellt und andererseits die Tatsache, dass Bootstrap alle modernen Webbrowser von unterschiedlichsten Anbietern unterstützt, da es auf den meistverwendeten Markup-Languages JavaScript und CSS basiert (hackr.io, 2022).

3.4.2 React

React ist eine JavaScript-Bibliothek, welche bei der Gestaltung von interaktiven User-Interfaces zur Anwendung kommt. Dabei können unterschiedliche Komponenten logisch voneinander getrennt werden, jedoch über eine Hauptkomponente wieder gemeinsam zur Darstellung gebracht werden. Jede React-Klasse kann somit HTML-Strukturen definieren und in einer anderen Klasse als sogenannte JSX-Tags wieder abgerufen werden. (Meta Platforms Inc., o.J.) Beispielsweise kann durch React eine Sidebar-Komponente mit der Bezeichnung „Sidebar“ erstellt werden und in jede beliebige Komponente – welche diese Sidebar benötigt – mittels JSX-Tag <Sidebar /> aufgerufen werden. Beim Rendern werden die einzelnen Komponenten in das Domain-Object-Model (DOM) übersetzt. Dabei handelt es sich um eine Baum-Struktur aller HTML-Elemente. React greift dabei auf das DOM zu und kreiert eine Repräsentation des Userinterfaces im Zwischenspeicher mittels virtuellem DOM. Werden nun Änderungen im Code vorgenommen, so ist es nicht notwendig, den gesamten Code neu zu rendern, sondern es wird lediglich der Zwischenspeicher mit den gewünschten Codeänderungen verglichen und Änderungen im DOM werden nur für jene Bereiche vorgenommen, welche auch tatsächlich geändert wurden. (Kovac, 2021)

3.4.3 Metamask

Metamask ist eine Open-Source Ethereum-Wallet, welche sämtliche Token (wie beispielsweise NFT) unterstützt, die von Ethereum angeboten werden. Die Wallet erlaubt dabei nicht nur das Senden und Empfangen von Ethereum-basierten Token, sondern bietet auch die Möglichkeit, als Webbrowser-Extension mit einer Webanwendung zu interagieren. Dabei werden lediglich jene Informationen über die Wallet-Adresse weitergegeben, sodass die Webanwendung über diese Adresse die zugeordneten Token von der Blockchain abrufen kann. Bei jeder Verbindung zu einer

neuen Webanwendung oder bei jeder Transaktion muss dies über die Browser-Extension mit dem privaten Schlüssel signiert werden. (Binance Academy, 2020)

Diese Browser-Extension dient dem Prototyp dabei, eine Verbindung mit dem privaten Schlüssel der Anwender/Anwenderinnen herzustellen, sodass jede Transaktion signiert werden kann. Sie stellt also in diesem System die Login-Informationen in Form der Wallet-Adresse zur Verfügung und wickelt sämtliche Transaktionsvorgänge über die digitale Signatur ab.

3.4.4 Web3(.js)

Web3 ist ein Begriff, welcher eine Dezentralisierung des digitalen Ökosystems bezeichnet. Nach Web1 – welches jenen digitalen Zeitrahmen beschreibt, in welchem erstmals dezentralisierte, offene Protokolle zur Kommunikation mit klassischen, statischen Webseiten zur Anwendung gekommen sind – und Web2, das die gegenwärtige Ära des Internets beschreibt, in welcher sowohl Kommunikation als auch Handel über zentrale Plattformen durchgeführt werden, zeigt Web3 die Möglichkeit zur dezentralen Organisation sämtlicher Internetaktivitäten auf. (Edelman, 2021) Web3.js dient dabei als Tool, um diese genannte Philosophie von Web3 zu ermöglichen. Es handelt sich dabei um eine Ethereum-JavaScript-Programmierschnittstelle, welche eine Vielzahl an Bibliotheken bereitstellt, um über eine Webapplikation mit einem Ethereum-Blockchain-Node zu interagieren. (Ethereum, o.J.a) Beispielsweise können mittels Web3.js Transaktionen durchgeführt, Vermögenswerte abgerufen oder Interaktionen mit Smart-Contracts durchgeführt werden. (McCubbin, 2022)

In Kombination mit der bereits vorgestellten Webbrowser-Extension Metamask stellt Web3.js dem Prototyp die Funktionalität zur Interaktion mit der Blockchain bzw. dem in ihr integrierten Smart-Contract zur Verfügung und Metamask dient zur Transaktionsabwicklung mittels digitaler Signatur. Beide Tools benötigen dabei eine Anbindung an dieselbe Blockchain, erfüllen jedoch ihren eigenen, voneinander unabhängigen Zweck.

3.4.5 Ganache und Truffle

Den Test des Prototyps anhand des Hauptnetzwerkes der Ethereum-Blockchain durchzuführen, würde enorme Kosten mit sich bringen, denn pro Transaktion werden zum jetzigen Zeitpunkt in etwa 13 USD an Transaktionsgebühren verrechnet (BitInfoCharts, o.J.). Diese Kosten würden somit für jede einzelne Transaktion anfallen, bei welcher neue Informationen in die Blockchain geschrieben werden.

Aus diesem Grund wird für die Tests eine lokale Ethereum-Test-Blockchain herangezogen, bei welcher die Transaktionsgebühren mit keinen realen Kosten verbunden sind, sondern lediglich symbolisch verrechnet werden. Ganache stellt eine Test-Blockchain für Ethereum zur Verfügung, welche lokal am eigenen Rechner installiert werden kann. Mit diesem Tool können Smart-Contracts in eine Testumgebung veröffentlicht und von ihr auch abgerufen und getestet werden (Lee, 2019). Ganache bietet eine Desktop-Version mit einem User-Interface an sowie die Test-Blockchain, die über die Kommandozeile der jeweiligen Entwicklungsumgebung initialisiert wird.

Truffle hingegen bietet sämtliche Möglichkeiten an, um mit der Test-Blockchain von Ganache zu kommunizieren. Das Tool wird im Projekt initialisiert und steht anschließend für die Kompilierung, die Veröffentlichung und für das Auslesen der veröffentlichten Smart-Contracts innerhalb der Kommandozeile zur Verfügung. (Trufflesuite, o.J.)

Trotz der Tatsache, dass Ganache lokal und damit zentral installiert wird, bildet dieses Tool alle Funktionalitäten ab, die eine reale Blockchain anbietet. Es können Smart-Contracts veröffentlicht und jede Transaktion kann in einem Block festgehalten werden. Die einzelnen Transaktionen und Blöcke sind einsehbar und es werden zahlreiche Wallet-Adressen, bestehend aus öffentlichem und privatem Schlüssel, zur Verfügung gestellt, über welche die jeweiligen Transaktionen abgewickelt werden können. Der Unterschied zu einer öffentlichen, realen Blockchain liegt lediglich darin, dass die Blockchain nicht verteilt – über mehrere Nodes – abgespeichert wird. Dies stellt jedoch für die Entwicklung des Prototyps kein Problem dar, da die Funktionalität der dezentralen Speicherung bereits ausreichend getestet ist und die Umsetzbarkeit einer Blockchain-Anwendung für den Nachweis identitätsbezogener Bescheinigungen auch in einer simulierten, zentralen Blockchain-Umgebung geprüft werden kann.

Im Sinne des Prototyps ergibt sich somit folgende Vorgangsweise zur Bereitstellung und Anwendung aller Funktionalitäten:

- Der Smart-Contract wird mittels Truffle in die lokale Test-Blockchain Ganache veröffentlicht.
- Die Test-Blockchain speichert somit alle Funktionalitäten, welche für die Erstellung, den Transfer sowie den Abruf der individuellen Nachweise in Form von NFT notwendig sind.
- Über Truffle können diese Funktionalitäten über die Kommandozeile abgerufen und getestet werden.
- Sowohl die Webanwendung als auch Metamask werden mit der lokalen Test-Blockchain verbunden.
- Die Webanwendung ruft über Web3.js die in Ganache veröffentlichen Funktionen des Smart-Contracts ab und bildet diese im GUI ab, sodass nach einer Interaktion eines Anwenders/einer Anwenderin die gewünschte Transaktion angestoßen wird.
- Durch diese Interaktion werden die Informationen über die Transaktion an Metamask weitergegeben, wo abschließend die Einwilligung für die Transaktion digital signiert wird.

3.4.6 Solidity und ERC721

Wie bereits beschrieben, wird der Smart-Contract des Prototyps – aufgrund der Bekanntheit und der bereits bestehenden Tools zur Kreierung von NFT – in eine Ethereum-Test-Blockchain veröffentlicht. Das Projekt Ethereum bedient sich hierbei der Programmiersprache Solidity. Solidity ist eine objektorientierte Programmiersprache, welche sich in Hinsicht auf ihre Syntax an der Sprache C++, Python und JavaScript orientiert (Ethereum, o.J.b). Solidity bietet neben der Möglichkeit zur Vererbung von Attributen und Methoden unterschiedlicher Klassen auch die

Unterstützung von Bibliotheken zur Integration von bereits bestehendem Funktionalitäten. Dabei wird die Version von Solidity im Projekt mittels „pragma solidity^<Version>“ initialisiert, sodass der Compiler alle notwendigen Informationen darüber hat, wie er den Source-Code beim kompilieren behandeln soll. (Ethereum, o.J.c)

Ein ERC721-Token bildet in Solidity den Token-Standard für NFT ab. Es handelt sich dabei um einen eigenständigen Smart-Contract, welcher bereits alle Methoden zur Verfügung stellt, welche für die Implementierung und Verwaltung von NFT benötigt werden. Dabei nimmt jeder über ERC721 erstellte NFT eine uint256-Variable in Form einer Token-ID entgegen. Diese „unique Integer“-Variable mit einer maximalen Länge von 256 Bit dient der Einzigartigkeit jedes einzelnen NFT und ist für jeden ERC721-Token zwingend notwendig. (Smith, 2022)

3.5 Mittels Blockchain zur selbstbestimmten, digitalen Identität

In diesem Unterkapitel folgt nun eine Verknüpfung der bereits erläuterten theoretischen Grundlagen. Die bereits gewonnenen Erkenntnisse werden thematisch zusammengeführt und es wird ein theoretisches Konzept einer Webanwendung aufgezeigt, welche den eindeutigen Nachweis von identitätsbezogenen Bescheinigungen erlaubt. Dabei wird zum einen auf die Notwendigkeit der technologischen Grundbausteine eingegangen, zum anderen wird erläutert, welche Aspekte für die Sicherstellung der Originalität der identitätsbezogenen Nachweise zwingend notwendig sind. Anschließend werden – aus der Sicht der Anwender/Anwenderinnen sowie Behörden und Unternehmen – Requirements in Form von User-Stories definiert, um nicht nur eine angemessene Usability zu gewährleisten, sondern auch auf jene Aspekte zu achten, welche für die Akzeptanz als digitale Nachweise erforderlich sind. Abschließend werden die für die Evaluierung benötigten Wertekriterien und Leistungsstandards definiert, sodass diese in der abschließenden Evaluierung gemessen werden können.

3.5.1 Gewährleistung der Fälschungssicherheit und Interpretation der Gültigkeit

In den theoretischen, technologischen Grundlagen wurde bereits des Öfteren auf die Eigenschaft der Unveränderlichkeit einer Blockchain eingegangen. Diese Eigenschaft ermöglicht es in einer Blockchain, jegliche Veränderungen anhand von stattgefundenen Transaktionen zu erkennen und bietet damit einer Webanwendung – zum Nachweis von identitätsbezogenen Bescheinigungen – einerseits die Möglichkeit, die Transaktionshistorie dieser Nachweise einzusehen und andererseits Veränderungen/Korrekturen dieser ausgestellten Nachweise zu erkennen. Vor allem die Einsicht der Transaktionshistorie stellt ein wesentliches Sicherheitsmerkmal dieser Webanwendung dar, denn dadurch kann sowohl ein (möglicher) Diebstahl als auch eine nicht-erlaubte Weitergabe dieser Bescheinigungen erkenntlich gemacht werden. Stellt also beispielsweise eine Behörde eine Bescheinigung an Person A aus und wird diese von Person A an Person B weitergeleitet, so kann diese Bescheinigung (sofern dies nicht ausdrücklich erlaubt ist) schließlich als ungültig interpretiert werden. Wie auch in Abbildung 11 ersichtlich, kann diese Gültigkeit anhand eines Transaktions-Zählers (folglich in Abbildung 11 als „transactionCount“ angegeben) – welcher den jeweiligen Nachweisen beigefügt wird – in der

Webanwendung dargestellt werden. Weist der Zähler einen Wert von 0 auf, bedeutet dies, dass die Behörde/das Unternehmen einen bestimmten Nachweis bereits erstellt hat, jedoch der zugehörigen Person A noch nicht transferiert hat. Der Nachweis ist daher noch nicht gültig. Wurde der Nachweis an Person A schließlich transferiert, so erhöht sich der Zähler auf 1 und der Nachweis wird als gültig angesehen. Tritt der Fall ein, dass der Nachweis ein weiteres Mal transferiert wird und der Zähler damit einen höheren Wert als 1 hat, so wird der Nachweis in der Webanwendung wiederum als ungültig gekennzeichnet.

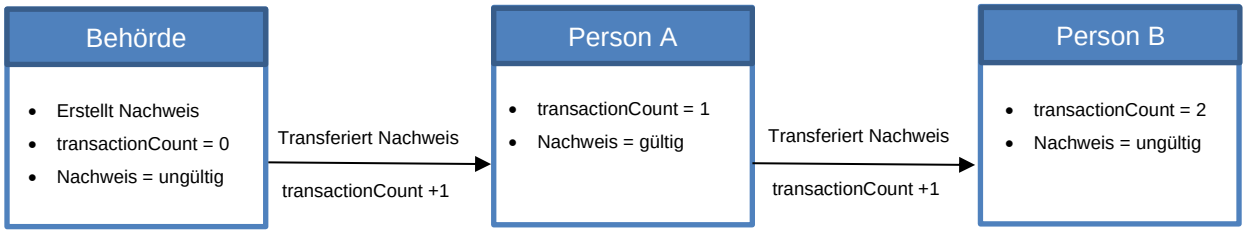

Abbildung 11: Gültigkeitsinterpretation anhand eines Transaktions-Zählers (Quelle: eigene Darstellung)

Jene Logik – welche unter anderem die beschriebene Erhöhung des Transaktions-Zählers realisiert – wird in einem Smart-Contract definiert. Der Smart-Contract dient also der Webanwendung dabei (anhand vordefinierter Daten), mögliche Diebstähle und Weitergaben von identitätsbezogenen Nachweisen (sofern dies mittels Transaktion erfolgt ist) zu verifizieren. Auch wird in diesem Smart-Contract der Aussteller der jeweiligen Bescheinigungen festgehalten, sodass in der Webanwendung ersichtlich dargestellt werden kann, von wem eine Person diese Bescheinigung erhalten hat. Diese Darstellung des Ausstellers erleichtert das Erkennen von Fälschungen, denn obwohl in einer öffentlichen Blockchain alle Teilnehmer/Teilnehmerinnen gleichberechtigt sind, sind jedoch auch alle Teilnehmer/Teilnehmerinnen aufgrund ihres kryptographischen Schlüsselpaares einzigartig. Durch die Definition des Ausstellers im Smart-Contract kann sodann jeder Teilnehmer/jede Teilnehmerin erkennen, ob es sich dabei um ein Original oder eine Fälschung handelt.

3.5.2 Definition von Funktionen

Obwohl alle Parteien in einer Blockchain gleichberechtigt sind, können die unterschiedlichen Teilnehmer/Teilnehmerinnen anhand der Information über deren Wallet-Adresse voneinander abgegrenzt werden. Durch die Identifizierung der Wallet-Adresse nach erfolgreicher Anmeldung mittels privatem Schlüssel können gewissen Anwendern/Anwenderinnen zusätzliche – für deren Funktion benötigte – Funktionalitäten zur Verfügung gestellt werden, was sich einerseits in einer besseren Usability widerspiegelt und andererseits unseriösen Parteien das Erstellen von gefälschten Nachweisen erschwert. Folgende Funktionalitäten werden demnach im Smart-Contract definiert und abhängig von der Wallet-Adresse bereitgestellt:

Funktionalitäten für seriöse Behörden und Unternehmen

- Kreierung von Nachweisen: Beim Erstellen eines Nachweises werden alle definierten Variablen an eine Funktion übergeben, die Identifikationsnummer des neu erstellten NFT wird mit dem definierten Nachweis verknüpft, der Transaktions-Zähler wird auf 0 gesetzt und die ausstellende Behörde wird als Besitzer und Aussteller abgebildet.
- Transfer von Nachweisen: Dabei wird eine Funktion aufgerufen, um einen erzeugten Nachweis an eine bestimmte Privatperson zu transferieren. Hierbei wird die Identifikationsnummer des NFT abgerufen, der Empfänger/die Empfängerin wird als neuer Besitzer/neue Besitzerin definiert und der Transaktions-Zähler wird um den Wert 1 erhöht. Sofern der Transaktions-Zähler durch diesen Transfer einen Wert von 1 annimmt, wird der Nachweis als gültig markiert. Sollte dieser Zähler jedoch einen Wert größer als 1 annehmen, wird der Gültigkeitsstatus als ungültig definiert.

Rollenunabhänige Funktionalität:

- Löschen von Nachweisen: Diese Funktion wird sowohl Privatpersonen als auch Behörden und Unternehmen jederzeit zur Verfügung gestellt, damit einerseits Privatpersonen ungültige Nachweise aus deren Nachweisbestand entfernen können und andererseits Behörden und Unternehmen die Möglichkeit haben, bei einer fehlerhaften Dateneingabe den noch nicht transferierten Nachweis wieder zu löschen.
- Gültigkeitsabruf von Nachweisen: Diese Funktion nimmt die Identifikationsnummer eines NFT entgegen und retourniert direkt von der Blockchain, welche jene Information, ob ein Nachweis gültig ist oder nicht, beinhaltet. Bevor die Gültigkeit des Nachweises übergeben wird, wird jedoch der Gültigkeitsstatus anhand des Transaktions-Zählers erneuert. Dies dient einer profilunabhängigen Validierung der Nachweise, sodass eine kontrollierende Partei anhand der ID eines NFT die Gültigkeit des zugehörigen Nachweises direkt von der Blockchain abrufen kann.

3.5.3 Verknüpfung mittels NFT und Definition von Variablen

Die zugrundeliegenden Nachweise werden mittels NFT verknüpft. Wie bereits erörtert, besitzen NFT die Eigenschaft, dass jeder dieser digitalen Token einzigartig ist. Wird also ein Nachweis neu erstellt, so wird dieser mit all seinen Daten in der Blockchain als NFT dargestellt. Hierfür wird dem erstellten Nachweis die einzigartige Identifikationsnummer des NFT zugewiesen. Die Eindeutigkeit dieser Token stellt wiederum sicher, dass einerseits Nachweise nicht dupliziert werden können und andererseits, dass jeder NFT einen individuellen Wert darstellen kann. Der Aufbau ist zwar bei jedem NFT ident, jedoch unterscheiden sich die einzelnen NFT anhand der in ihm gespeicherten Werte. Dabei soll jeder NFT folgende Werte aufweisen:

- TokenID: Diese Variable wird bei jedem ausgestellten Nachweis um den Wert 1 erhöht und stellt somit die Einzigartigkeit eines jeden NFT sicher.

- TransactionCount: Wie bereits beschrieben, wird für die Nachvollziehbarkeit der Gültigkeit eines Nachweises in der Webanwendung ein Zähler verwendet, wobei dieser bei der Erstellung mit dem Wert 0 initialisiert wird und bei jeder darauffolgenden Transaktion um den Wert 1 erhöht wird. Die Gültigkeit ist nur dann gegeben ist, sofern dieser Zähler den Wert 1 aufweist.
- Description: Diese Variable gibt an, um welche Art von Nachweis es sich handelt (z.B. Reisepass, Führerschein, Diplom, etc.)
- IssuerAddress: Speichert die Wallet-Adresse des Ausstellers des Nachweises, welcher von der Webanwendung erkenntlich dargestellt werden kann.
- IssuerName: Speichert den übergebenen Namen jener Behörde/jenes Unternehmens, welche den Nachweis ausgestellt hat. Der Name der Einrichtung wird dabei manuell beim Erstellen der Bescheinigung erfasst.
- Owner: In dieser Variable wird festgehalten, wer im Moment im Besitz des Nachweises ist, sodass dieser auch eindeutig einem Besitzer/einer Besitzerin zugeordnet und in der Blockchain dargestellt werden kann.
- Name: Speichert den Namen jener Person, für welche der Nachweis bestimmt ist.
- DateOfBirth: Speichert das Geburtsdatum jener Person, für welche der Nachweis bestimmt ist.
- ValidityDate: Gibt an, wie lange ein bestimmter Nachweis gültig ist, sodass dieser nach einer gewissen Zeit erneuert werden muss.
- Validity: Diese Variable soll die Variablen „transactionCount" und „validityDate" zu einer Gültigkeitsüberprüfung kombinieren. Sie kann lediglich die Werte „true" oder „false" annehmen, sodass die Gültigkeit anhand dieser beiden Werte abgefragt werden kann.
- URL: Verlinkt auf die Webseite des Ausstellers, auf welcher weitere Informationen zum Nachweis bereitgestellt werden können.
- [weitere Variablen]: Je nachdem, um welche Art von Nachweis (Variable „Description) es sich handelt, können weitere, individuelle Felder befüllt werden. All jene Variablen, die für einen bestimmten Nachweis nicht benötigt werden, werden dabei mit einem Default-Wert belegt..

3.5.4 Webanwendung zur Bereitstellung dezentraler Funktionen

Grundsätzlich kann ein Smart-Contract – welcher in einer Blockchain veröffentlicht wurde – von jeder Person ohne Weboberfläche über eine beliebige Kommandozeile genutzt werden. Webanwendungen dienen hierbei einerseits zur Bereitstellung der im Smart-Contract definierten Funktionen und andererseits sind sie für Anwender/Anwenderinnen eine vereinfachte Möglichkeit, diese Funktionen zu nutzen. Der veröffentliche Smart-Contract kann somit auch von jeder Person in eine eigene Webanwendung integriert werden. Wie in Abbildung 12 grafisch

dargestellt, benutzen sie somit die Blockchain als einheitliche, dezentrale Datenbank und definieren dadurch die Schnittstelle zwischen Blockchain und Endanwender/Endanwenderin.

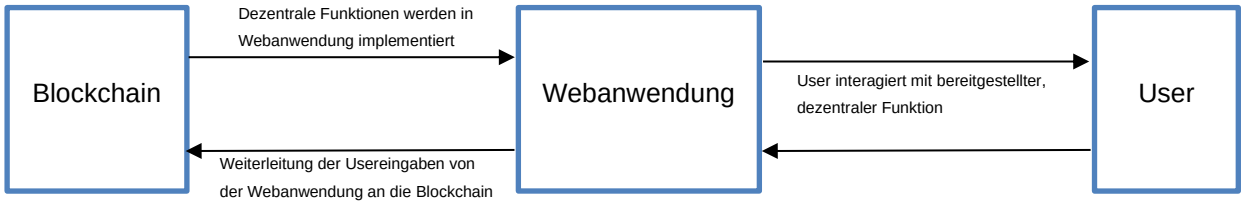

Abbildung 12: Webanwendung als Schnittstelle zw. Blockchain und User (Quelle: eigene Darstellung)

Da in dieser Architektur eine dezentrale Datenbank bereitgestellt wird und diese anhand der dezentralen Schlüsselpaare Informationen zuordnet, ist auch kein zentrales Login-Verfahren notwendig. Das Login soll lediglich über den privaten Schlüssel der User funktionieren, denn dieser ermöglicht jedem Teilnehmer/jeder Teilnehmerin den Zugriff auf die ihm/ihr zugeordneten NFT. Dieser Aspekt führt zu einer Selbstbestimmung über die eigenen, identitätsbezogenen Nachweise, denn die Webanwendung kann keinem User den Zugriff auf die eigenen NFT verwehren, da diese lediglich die Informationen aus der Blockchain grafisch darstellen. Jeder Nutzer/jede Nutzerin hat somit die alleinige Kontrolle über seine/ihre NFT.

3.5.5 Definition von User-Stories

In diesem Unterkapitel wird die Webanwendung sowie deren Funktionalität aus der Perspektive der Anwender/Anwenderinnen betrachtet. Dies ermöglicht eine Erörterung der Anforderungen seitens der User und bildet somit die Mindestanforderungen für eine entsprechende Usability ab. Alle User-Stories beantworten jene Fragestellungen, welche in der agilen Softwareentwicklung typischerweise zur Anwendung kommen (Raharjana, Siahaan & Fatichah, 2021):

- „Wer“ ist der Akteur im System?
- „Was“ ist die Anforderung bzw. der Wunsch dieses Akteurs?
- „Warum“ trägt diese Anforderung zur Zufriedenheit des Akteurs bei?

Die daraus resultierenden User-Stories dienen schlussendlich dazu, Funktionalitäten an die Wünsche unterschiedlicher Akteure anzupassen und um den Erledigungsstatus anhand der User-Kriterien zu messen. Bei der Definition der User-Stories wird – speziell aus der Perspektive der Privatpersonen – möglichst darauf geachtet, dass die Gründe für eine Anforderung an den in dieser Arbeit untersuchten Aspekt der Selbstbestimmung ausgerichtet sind, wobei jede User-Storie dieselbe Struktur aufweist:

Als <Akteur>(Wer?) möchte ich <Funktionalität>(Was?), damit <Begründung>(Warum?).

Dabei werden drei unterschiedliche Gruppen von Akteuren (Behörden/Unternehmen, kontrollierendes Organ, Privatperson) näher betrachtet, sodass die daraus resultierende Webanwendung die Wünsche sämtlicher Teilnehmer/Teilnehmerinnen individuell abdeckt und zu einer möglichst großen Akzeptanz führt. Die User-Stories richten sich weniger an die bereits definierten Funktionen des Smart-Contracts als mehr an die Usability der Webanwendung, welche schlussendlich mittels der Funktionen durch die Interaktion der Anwender/Anwenderinnen in einem GUI erfüllt werden soll.

User-Story 1:

Sowohl als Privatperson als auch als Unternehmen/Behörde möchte ich mich mit meinem privaten Schlüssel unkompliziert anmelden können, damit ich sämtliche – mir zugeordnete – Bescheinigungen vollumfänglich verwalten kann.

User-Story 2:

Als Unternehmen/Behörde möchte ich Eingabefelder abhängig von der Art des zu erstellenden Nachweises angeboten bekommen, damit ich den Prozess der Erstellung schnell durchlaufen kann.

User-Story 3:

Als Unternehmen/Behörde möchte ich einen Ausweis nach Eingabe aller Daten unkompliziert über einen Button erstellen und anschließend in meiner Ansicht angezeigt bekommen, damit ich den ausgestellten Ausweis nochmals auf die korrekte Dateneingabe kontrollieren kann.

User-Story 4:

Als Privatperson möchte ich, dass die für mich bestimmten, identitätsbezogenen Nachweise nach der Erstellung an mich transferiert werden können, damit meine eigenen Nachweise auch in meiner Kontrolle sind.

User-Story 5:

Als Privatperson möchte ich alle Nachweise gesammelt und alphabethisch nach der Art des Nachweises sortiert auf einer Seite angezeigt bekommen, damit ich bei einer möglichen Kontrolle nicht meinen gesamten Nachweisbestand durchsuchen muss.

User-Story 6:

Als kontrollierendes Organ möchte ich durch die Eingabe der Identifikationsnummer eines Nachweises selbst aus der Blockchain dessen Gültigkeit sowie die Adresse der ausstellenden Behörde/des ausstellenden Unternehmens abrufen können, sodass mögliche, grafische Manipulationen seitens der sich ausweisenden Person ausgeschlossen werden können.

User-Story 7:

Als Privatperson möchte ich grafisch dargestellt bekommen, wie lange und ob ein Nachweis noch gültig ist, damit ich mich ehestmöglich um eine neue Ausstellung des betreffenden Nachweises kümmern kann.

User-Story 8:

Als Behörde/Unternehmen möchte ich fehlerhafte Nachweise im Zeitraum zwischen Erstellung und Transfer wieder löschen können, damit diese in Zukunft nicht irrtümlicherweise an eine Privatperson versendet werden.

User-Story 9:

Als Privatperson möchte ich ungültige Nachweise löschen können, damit mir diese in meinem Profil nicht mehr angezeigt werden.

3.5.6 Definition von Wertekriterien und Leistungsstandards

In Kapitel 3.2.3 wurde dargestellt, dass für eine Evaluierung eines Prototyps gewisse Wertekriterien und Leistungsstandards benötigt werden, um eine messbare Aussage über die praktische Umsetzung zu erhalten und dahingehend ein Werturteil fällen zu können. Diese Wertekriterien und Leistungsstandards werden nun definiert und in Kapitel 5.1 gemessen und bewertet.

Wertekriterium 1: Kontrolle über die eigenen, identitätsbezogenen Nachweise

Leistungsstandard 1: Damit dieses Wertekriterium erfüllt werden kann, ist es wesentlich, dass jene identitätsbezogenen Nachweise, welche einer bestimmten Partei zugeordnet sind, auch im Besitz dieser Person sind und unabhängig von einem GUI verfügbar sind. Die Nachweise und Bescheinigungen dürfen von keiner zentralen Partei abhängig sein, sodass die Verfügbarkeit dieser NFT an die Verfügbarkeit von zentralen Webapplikationen gerichtet ist. Ein Serviceanbieter soll lediglich in der Lage sein, seinen Anwendern/Anwenderinnen die benötigten Funktionalitäten zur Kontrolle der eigenen NFT in einer einfacheren, grafischen Form bereitzustellen.

Wertekriterium 2: Uneingeschränkte Nutzung aller Funktionalitäten des Smart-Contracts

Leistungsstandard 2: Unter Berücksichtigung des Wertekriteriums 1 soll es allen NFT-Besitzern auch möglich sein, sämtliche Funktionen des Smart-Contracts nutzen zu können. Es soll in Hinsicht auf eine vollumfänglichen Verfügbarkeit über die eigenen Bescheinigungen auch jeder Person möglich sein, Funktionalitäten nutzen zu können, welche abseits von bereitgestellten Funktionen der Webanwendungen zur Verfügung stehen, um so eine Unabhängigkeit von zentralen Anbietern zu schaffen.

Wertekriterium 3: Verifizierung von seriösen Parteien/Behörden

Leistungsstandard 3: Da sämtliche Funktionalitäten nicht von einer bestimmten Webanwendung abhängig sein dürfen, sollte es auch jedem Nutzer/jeder Nutzerin von sich aus möglich sein, zu beweisen, dass ein erhaltener Nachweis von einer seriösen Behörde/einem seriösen Unternehmen stammt. Die Seriosität darf dabei von keiner zentralen Partei definiert sein, denn dies würde wiederum zu einer zentralen Abhängigkeit führen und damit auch die Verifizierung von einzelnen Teilnehmern/Teilnehmerinnen abhängig machen.

Wertekriterium 4: Sicherstellung von nachträglichen Manipulationen

Leistungsstandard 4: Auch in diesem Wertekriterium spielt die Kontrolle über die eigenen Nachweise eine wichtige Rolle. Es soll nach der Ausstellung eines NFT nicht mehr möglich sein, dass dieser NFT von einer dritten Partei im Nachhinein manipuliert und/oder entzogen wird.

Wertekriterium 5: Aktualisierung der Gültigkeit eines Nachweises

Leistungsstandard 5: Wie bereits innerhalb der erforderlichen Funktionalitäten beschrieben, wird die Gültigkeit eines Nachweises über jene Variable definiert, welche die Anzahl der durchgeführten Transferierungen eines NFT zählt. Es sollte jedoch auch möglich sein, den Gültigkeitsstatus anhand des definierten Gültigkeitsdatums zu aktualisieren, sodass ein Nachweis mit Ablauf dieses Datums als ungültig dargestellt wird und dadurch nicht manuell kontrolliert werden muss.

Wertekriterium 6: Sicherstellung des Datenschutzes

Leistungsstandard 6: Da die digitalisierten Nachweise sensible Daten beinhalten, ist es notwendig, dass diese von dritten Parteien nicht ohne die Erlaubnis des Besitzers/der Besitzerin eingesehen werden können. Dennoch sollte es möglich sein, dass der Besitzer/die Besitzerin die Einsicht auf seine/ihre eigenen Daten – für eine mögliche Kontrolle – temporär gewährt.

4 ERGEBNISSE

In diesem Kapitel werden die Ergebnisse des Prototyps sachlich aufgezeigt und mittels Screenshots grafisch abgebildet. Die Darstellung der Ergebnisse wird an die definierten User-Stories gerichtet, sodass die Umsetzbarkeit der benötigten Funktionalitäten aus Sicht der Anwender/Anwenderinnen dargestellt werden kann. Als Namensgeber für den Prototyp wurde „Digity“ definiert, es steht für den englischen Begriff der digitalen Identität – „digital identity“.

Bevor die Umsetzung der User-Stories präsentiert wird, werden vorerst die Unterschiede der SPA dahingehend dargestellt, ob es sich bei der nutzendenden Person um eine Privatperson oder um eine seriöse Einrichtung handelt. Die Unterschiede liegen dabei im benötigten Funktionsumfang, welcher der nutzenden Partei bereitgestellt werden soll. Als Beispiel zur Unterscheidung der rollenbezogenen Nutzer stellt in folgenden Abbildungen die Wallet-Adresse „0xEbAD1AA4B5E267d0C86fCeaD6942f8C945047E39“ ein Beispielunternehmen dar, während die Wallet-Adresse „0x515df25aff9aF517D378F3624431E8A5beBC5ccD“ eine Privatperson abbildet.

Wie in Abbildung 13 ersichtlich, unterscheiden sich die GUI im Wesentlichen lediglich in der Möglichkeit, ob eine neue Bescheinigung ausgestellt werden kann oder nicht. Während seriösen Behörden/Unternehmen diese Funktionalität zur Verfügung gestellt wird, soll es allen anderen Anwendern/Anwenderinnen nicht möglich sein, über die Webapplikation neue, identitätsbezogene Nachweise zu erstellen, was einen ersten Aspekt im Sinne der Fälschungssicherheit darstellt.

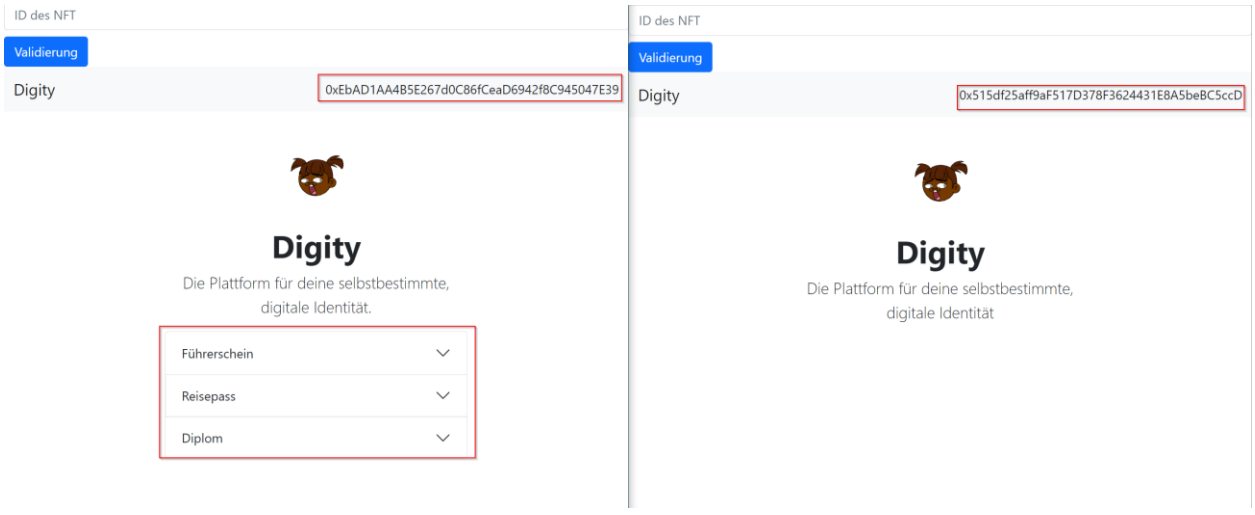

Abbildung 13: Digity - Rollenabhängige Unterscheidungen der SPA (Quelle: eigene Darstellung)

Weitere GUI-Unterscheidungen werden in jenen User-Stories dargestellt, in welchen eine Unterscheidung seitens der Webanwendung notwendig ist, um mögliche Fälschungen bereits auf GUI-Basis zu erschweren.

4.1 User-Story 1

Sowohl als Privatperson als auch als Unternehmen/Behörde möchte ich mich mit meinem privaten Schlüssel unkompliziert anmelden können, damit ich sämtliche – mir zugeordnete – Bescheinigungen vollumfänglich verwalten kann.

Wie bereits in Kapitel 3.4.3 beschrieben, werden sämtliche Transaktionen über die Webbrowser-Extension Metamask abgewickelt. Sie stellt die Schnittstelle zwischen Blockchain und Webanwendung dar. Da die Webanwendung lediglich jene Informationen und Funktionalitäten als GUI zur Verfügung stellt, welche im Smart-Contract definiert wurden, wird auch für die Anmeldung kein separates Login-Verfahren benötigt, denn anhand der jeweiligen Wallet-Adresse des Anwenders/der Anwenderin können bereits alle notwendigen Informationen aus der Blockchain ausgelesen werden. Für das Login ist somit lediglich der Import des eigenen, privaten Schlüssels in die Browser-Extension von Metamask notwendig, wodurch schlussendlich – wie auch in Abbildung 14 dargestellt – die Anmeldung über die daraus generierte Wallet-Adresse automatisch bei der nächsten Aktualisierung der Webanwendung angestoßen wird.

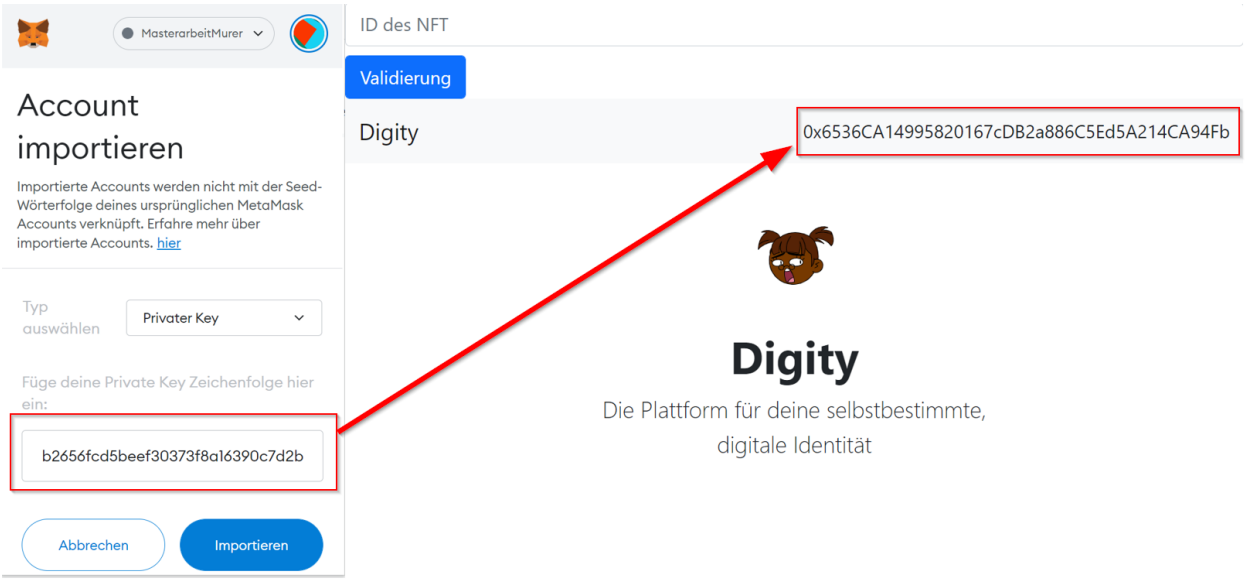

Abbildung 14: Digity - Login anhand privatem Schlüssel (Quelle: eigene Darstellung)

Anhand der generierten Wallet-Adresse wird wiederum festgestellt, ob es sich um eine Privatperson oder eine seriöse Einrichtung handelt und der Anwender/die Anwenderin bekommt anhand dieser Feststellung wiederum die für ihn/sie notwendigen Funktionalitäten zur Verfügung gestellt.

4.2 User-Story 2

Als Unternehmen/Behörde möchte ich Eingabefelder abhängig von der Art des zu erstellenden Nachweises angeboten bekommen, damit den Prozess der Erstellung schnell durchlaufen kann.

Sofern es sich bei der erkannten Wallet-Adresse um eine seriöse Einrichtung handelt, wird dem Anwender/der Anwenderin die Möglichkeit zur Erstellung unterschiedlicher Nachweise eingeblendet. Zur Veranschaulichung der Umsetzungsmöglichkeit wurden – wie in Abbildung 15 ersichtlich – die Nachweis-Arten Führerschein, Reisepass und Diplom definiert.

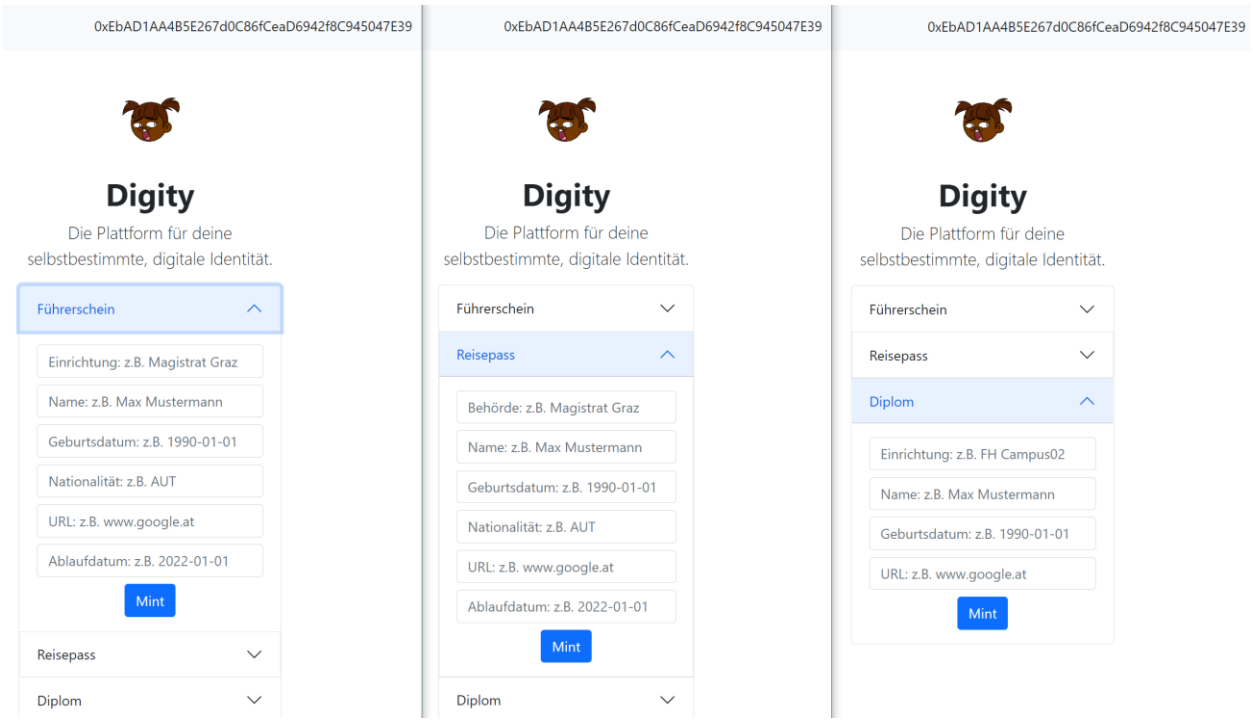

Abbildung 15: Digity - unterschiedliche Formulare für unterschiedliche Nachweis-Arten (Quelle: eigene Darstellung)

Anhand der Art des auszustellenden Nachweises können dabei unterschiedliche Eingabefelder angeboten werden. All jene Eingabefelder, die für einen entsprechenden Nachweis nicht benötigt werden, werden dabei ausgeblendet. Beispielsweise wurde im Prototyp definiert, dass ein Diplom kein Ablaufdatum haben sollte und auch die Nationalität des Absolventen/der Absolventin keine – für diese Bescheinigung – notwendige Information darstellt. Die Art des Nachweises muss bei der Erstellung nicht manuell erfasst werden, sondern wird automatisiert durch den Klick auf das jeweilige Drop-Down-Formular in der Webanwendung als Nachweisart übernommen.

4.3 User-Story 3

Als Unternehmen/Behörde möchte ich einen Ausweis nach Eingabe aller Daten unkompliziert über einen Button erstellen und anschließend in meiner Ansicht angezeigt bekommen, damit ich den ausgestellten Ausweis nochmals auf die korrekte Dateneingabe kontrollieren kann.

Jede seriöse Einrichtung hat die Möglichkeit, einen Nachweis über das Formular der jeweiligen Nachweisart zu erstellen. Für die Erstellung steht ein Button „Mint“ zur Verfügung, welcher eine Blockchain-Transaktion auslöst und über Metamask bestätigt werden muss. Ein Beispiel dieses Erstellungsprozesses wird in Abbildung 16 anhand der Kreierung eines neuen Führerscheins grafisch dargestellt.

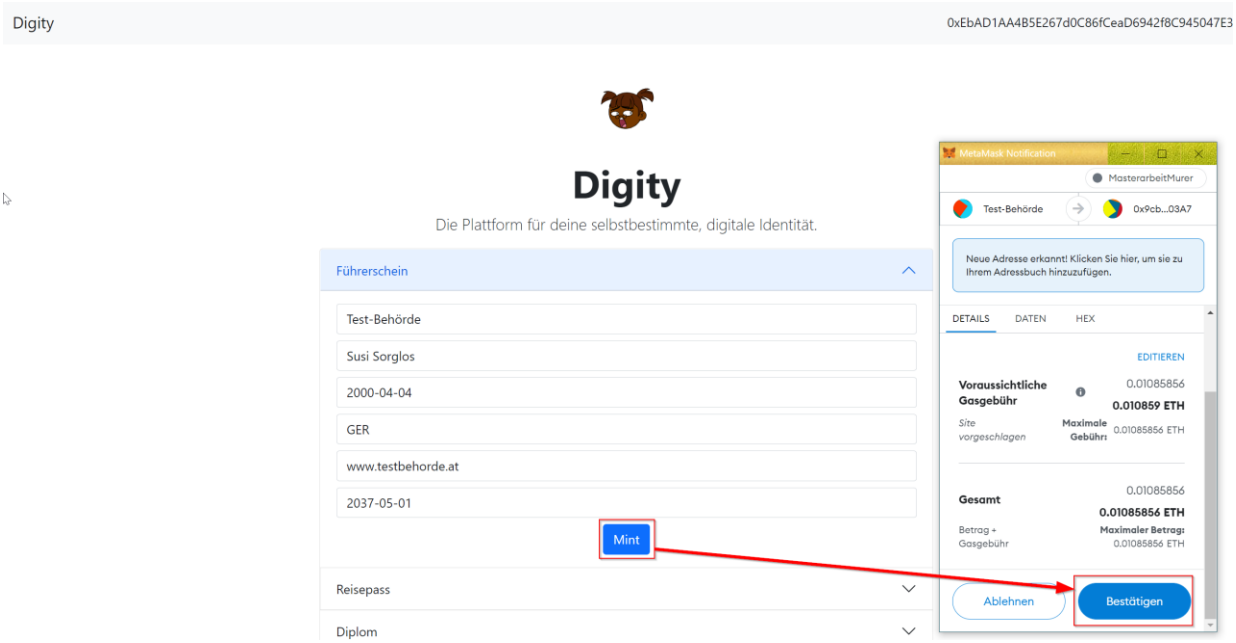

Abbildung 16: Digity - Prozess der Erstellung eines Führerscheins (Quelle: eigene Darstellung)

Sobald die Transaktion und damit die Einwilligung zur Erstellung des neuen Führerscheins bestätigt wurde, lässt sich die Transaktion mit den notwendigen Informationen innerhalb der Blockchain auslesen (siehe dazu nachfolgenden Auszug aus der Ganache Test-Blockchain).

Abbildung 17: Darstellung der Führerscheinerstellung innerhalb einer Blockchain-Transaktion (Quelle: eigene Darstellung)

Anzumerken ist dabei, dass jene Daten für das Geburts- und Gültigkeitsdatum in der Blockchain als Unix-Format dargestellt werden, da diese Informationen im Smart-Contract als Integer-Datentyp initialisiert wurden, um in weiterer Folge weitere Berechnungen mit diesen Daten durchführen zu können.

Der erstellte Führerschein wird – sobald die Transaktion in einem neuen Block aufgenommen und dieser an die Blockchain gehängt wurde – in der Übersicht der ausstellenden Behörde angezeigt (siehe Abbildung 18 in User-Story 4).

4.4 User-Story 4

Als Privatperson möchte ich, dass die für mich bestimmten, identitätsbezogenen Nachweise nach der Erstellung an mich transferiert werden können, damit meine eigenen NFT auch in meiner Kontrolle sind.

Sobald der Führerschein in der Ansicht jener Behörde angezeigt wird, welche ihn ausgestellt hat, steht ein weiterer Button sowie ein Eingabefeld zum Transfer des Führerscheins zur Verfügung. In Abbildung 18 wird der Prozess zum Transfer des Führerscheins abgebildet und in weiterer Folge textlich begleitet.

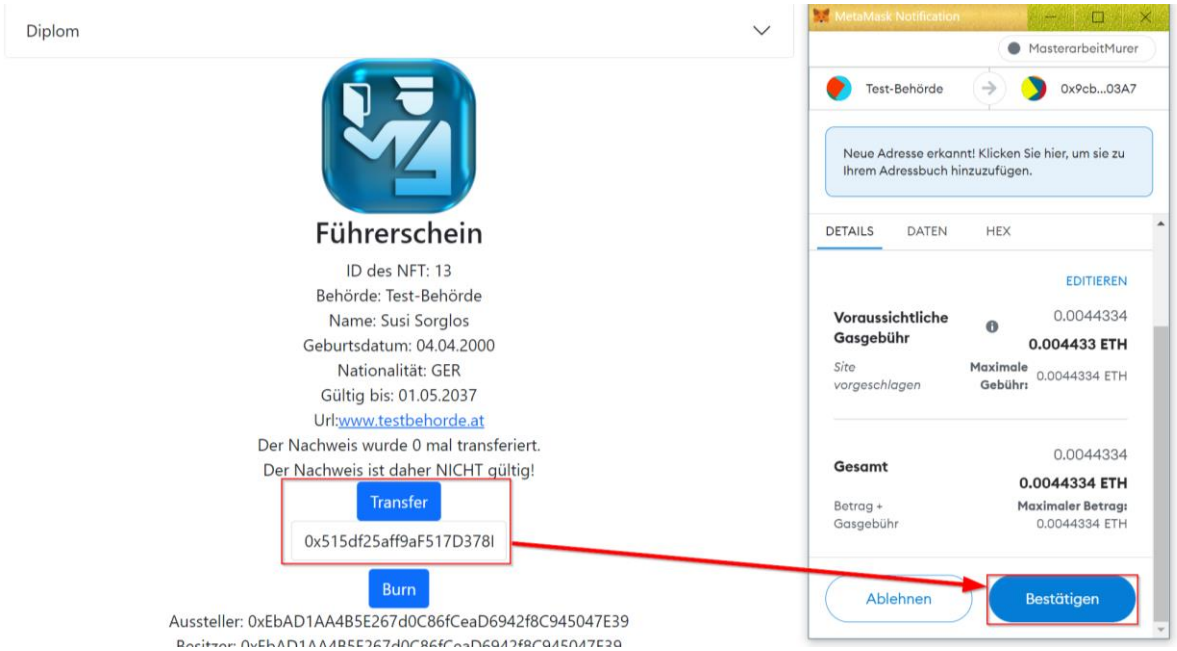

Abbildung 18: Digity - Transferprozess eines Führerscheins (Quelle: eigene Darstellung)

Im ersten Schritt wird die Wallet-Adresse jener Person im Eingabefeld erfasst, für welche dieser Führerschein gedacht ist. Mittels Transfer-Button wird eine Blockchain-Transaktion ausgeführt, für welche wiederum über Metamask die Einwilligung bestätigt werden muss. Auch ist in der oben angeführten Abbildung bereits ersichtlich, dass der ausgestellte Führerschein vor dem Transfer noch nicht gültig ist, da er noch nicht transferiert wurde. Wie in Kapitel 3.5.3 formuliert, wurde im Smart-Contract definiert, dass die Variable „validity“ eines NFT nur dann denn Wert „true“ annimmt, sofern die Variable „transactionCount“ den Wert 1 aufweist. Diese Zähler-Variable wird – wie in Kapitel 3.5.2 beschrieben – bei der Erstellung eines Nachweises mit dem Wert 0 vorbelegt und bei jeder weiteren Transaktion um den Wert 1 erhöht. Nach erfolgter Transaktion kann diese in der Blockchain wiederum eingesehen werden. Aus Abbildung 19 geht hervor, dass die Funktion „transfer()“ ausgeführt wurde, welcher die zu transferierende Nachweis-ID (tokenID) sowie jene Adresse des Empfängers/der Empfängerin übergeben wurde.

Abbildung 19: Darstellung des Führerschein-Transfers innerhalb einer Blockchain-Transaktion (Quelle: eigene Darstellung)

Die Transaktion wurde also durchgeführt, sodass – wie nachfolgend abgebildet – der Besitz des Nachweises an die erfasste Wallet-Adresse übertragen wurde und dieser Nachweis dem neuen Besitzer/der neuen Besitzerin in seiner/ihrer Übersicht angezeigt wird.

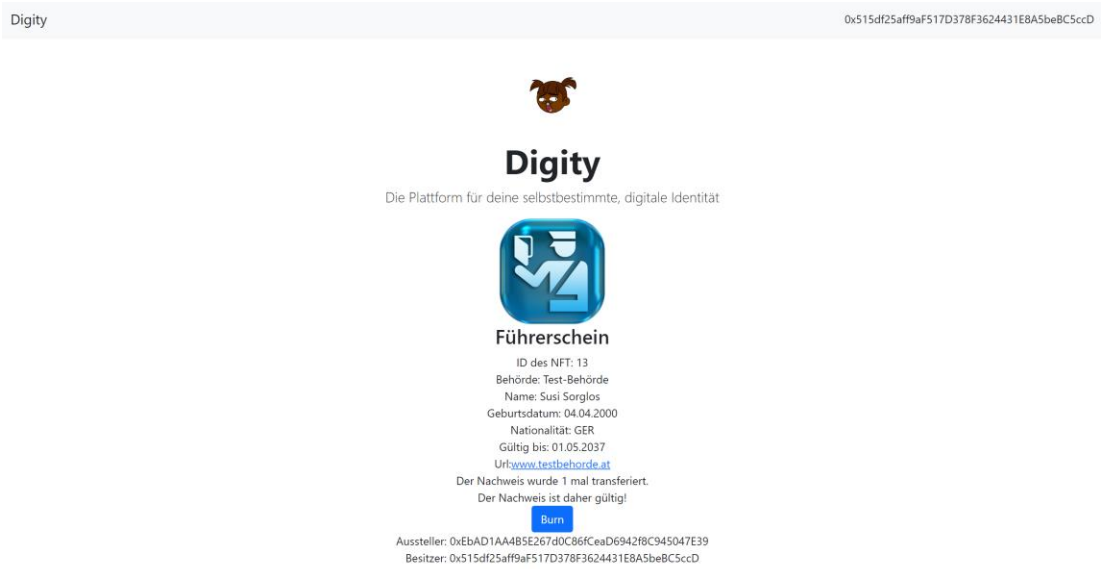

Abbildung 20: Digity - Transferierter Führerschein im Besitz des Empfängers/der Empfängerin (Quelle: eigene Darstellung)

Durch den Transfer wurde auch der Transaktions-Zähler um den Wert 1 erhöht und wird somit als „gültig“ identifiziert. Auch ist die Wallet-Adresse der ausstellenden Behörde weiterhin ersichtlich, sodass jederzeit nachgewiesen werden kann, wer den Nachweis kreiert und an den Empfänger/die Empfängerin übermittelt hat.

4.5 User-Story 5

Als Privatperson möchte ich alle Nachweise gesammelt und alphabethisch nach der Art des Nachweises sortiert auf einer Seite angezeigt bekommen, damit ich bei einer möglichen Kontrolle nicht meinen gesamten Nachweisbestand durchsuchen muss.

In Abbildung 21 ist zu erkennen, dass einem Anwender/einer Anwenderin jene ausgestellten Nachweise alphabetisch nach der Art des Nachweises grafisch dargestellt werden, welche ihm/ihr transferiert wurden. Wichtig zu erwähnen ist dabei, dass nicht nur aufgrund der Usability, sondern auch aus Datenschutz-Gründen lediglich jene Nachweise angezeigt werden, welche dieser Privatperson auch zugeordnet sind.

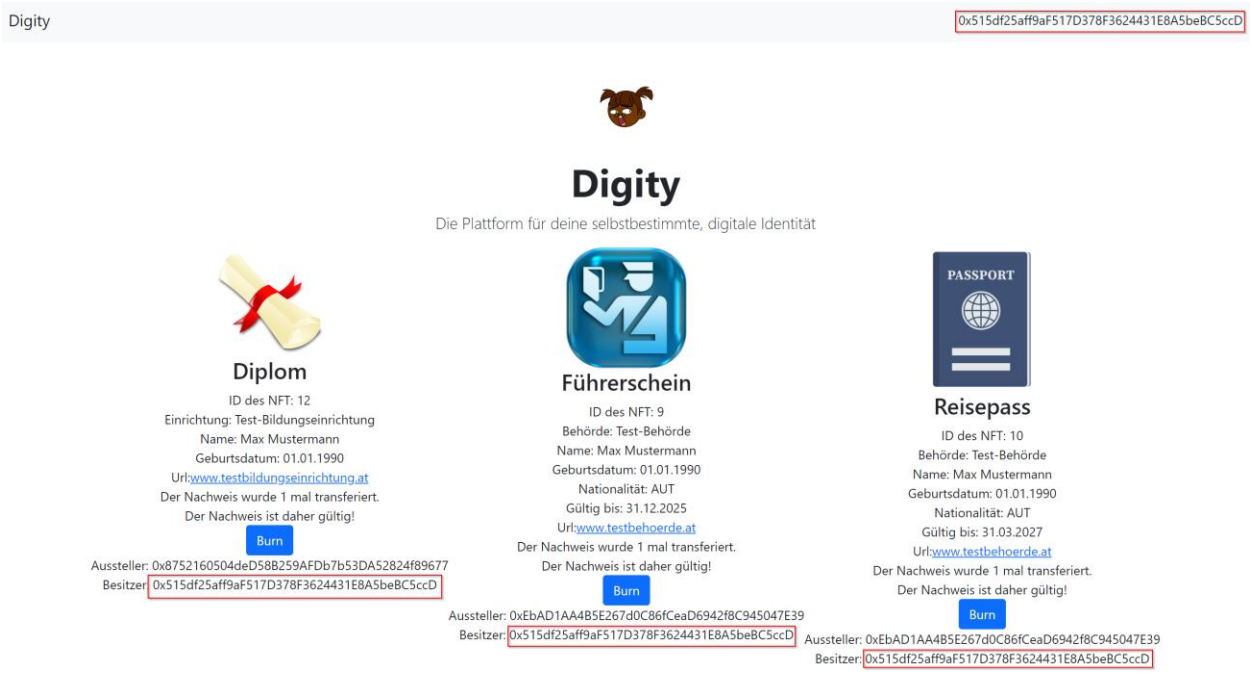

Abbildung 21: Digity - grafische Darstellung der eigenen Nachweise (Quelle: eigene Darstellung)

Auch ist zu erkennen, dass jene Informationen, welche für eine mögliche Kontrolle oder einen Nachweis notwendig sind, unter jeder Bescheinigung angeführt werden. Jene Informationen, welche bereits bei der Erstellung der Bescheinigung keine Relevanz hatten, werden auch in der Übersicht der Nachweise ausgeblendet.

4.6 User-Story 6

Als kontrollierendes Organ möchte ich durch die Eingabe einer bestimmten Identifikationsnummer eines Nachweises selbst aus der Blockchain dessen Gültigkeit sowie die Adresse der ausstellenden Behörde/des ausstellenden Unternehmens abrufen können, sodass mögliche, grafische Manipulationen seitens der sich ausweisenden Person ausgeschlossen werden können.

Für diesen Zweck wurde am oberen Bildausschnitt ein Eingabefeld für die Abfrage der Gültigkeit eines Nachweises mit einer bestimmten ID geschaffen. Wie in Abbildung 22 ersichtlich, wird anhand der erfassten Nachweis-ID angezeigt, wer den Nachweis ausgestellt hat und ob dieser Nachweis gültig (validity = true) ist. Dabei spielt es keine Rolle, ob die abfragende Partei der Besitzer/die Besitzerin des Nachweises ist. Wird nach einer nicht-vorhandene Nachweis-ID gesucht, so wird der suchenden Person auch angezeigt, dass es zu dieser ID keinen Nachweis gibt.

Abbildung 22: Digity – Gültigkeitsvalidierung eines Nachweises (Quelle: eigene Darstellung)

Diese Funktionalität stellt einen weiteren Sicherheitsaspekt in Hinsicht auf mögliche Manipulationen auf dem Bildschirm der sich ausweisenden Person dar, denn – wie bereits erwähnt – muss die abfragende Person für die Gültigkeitsvalidierung nicht im Besitz des Nachweises sein, um diese Informationen abfragen zu können. Die benötigten Daten werden dabei direkt aus der Blockchain abgerufen, welche aufgrund ihrer Fälschungssicherheit als valide angesehen werden können. Die Fälschungssicherheit kann auch deswegen angenommen werden, da im Smart-Contract keine Funktion angeboten wird, die eine Manipulation der Nachweis-Informationen erlaubt.

4.7 User-Story 7

Als Privatperson möchte ich grafisch dargestellt bekommen, wie lange und ob ein NFT noch gültig ist, damit ich mich ehestmöglich um eine neue Ausstellung des betreffenden Nachweises kümmern kann.

Wie bereits in Abbildung 20 dargestellt, wird jedem Besitzer/jeder Besitzerin innerhalb seines/ihres digitalen Nachweises angezeigt, ob ein Nachweis gültig ist. Auch wird jenes Datum ausgegeben, welches seitens der ausstellenden Behörde als Ablaufdatum definiert wurde. Jene Anzeige, ob es sich um einen gültigen oder ungültigen Nachweis handelt, wird dabei lediglich anhand des Transaktionszählers bestimmt ohne Berücksichtigung des definierten Gültigkeitsdatums, da dies eine technologische Herausforderung darstellt, welche in der abschließenden Diskussion näher erläutert wird.

4.8 User-Story 8

Als Behörde/Unternehmen möchte ich fehlerhafte Nachweise im Zeitraum zwischen Erstellung und Transfer wieder löschen können, damit mir diese in Zukunft nicht irrtümlicherweise an eine Privatperson versendet werden.

In Abbildung 18 ist erkenntlich, dass ein Nachweis bereits nach der Erstellung über einen Burn-Button verfügt. Dieser Button dient dem Löschen eines Nachweises. In nachfolgender Abbildung wird wiederum der Löschprozess grafisch festgehalten und anschließend textlich beschrieben.

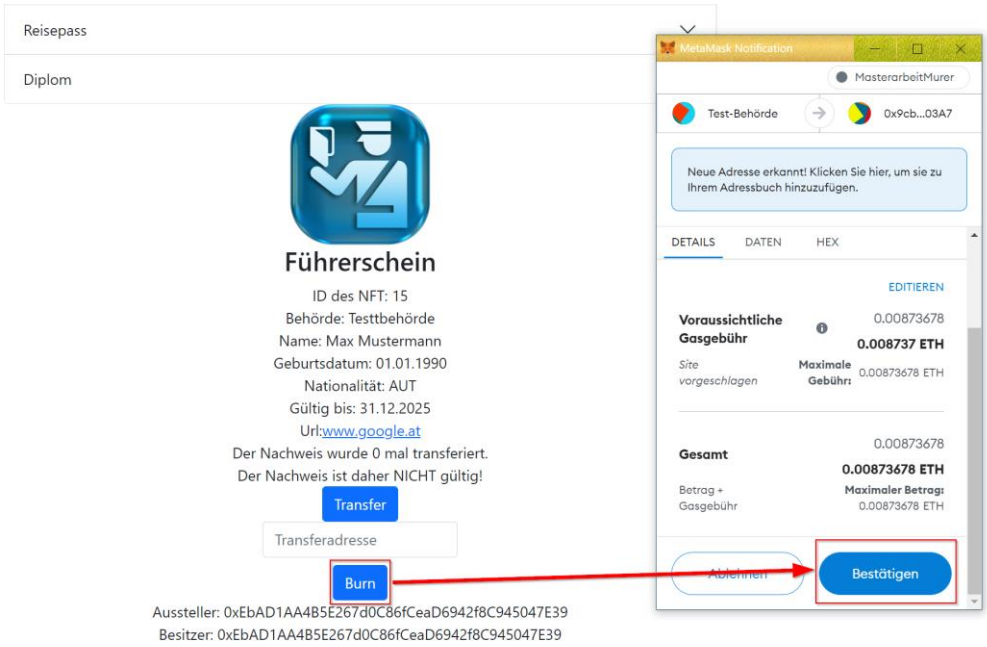

Abbildung 23: Digity - Löschprozess eines Führerscheins (Quelle: eigene Darstellung)

Wird der Löschprozess mittels Burn-Button angestoßen, wird eine Blockchain-Transaktion an Metamask signalisiert und die Einwilligung – wie auch alle anderen Transaktionen – mittels Bestätigung erklärt. Innerhalb der Blockchain-Transaktion kann die ausgeführte Funktion wiederum eingesehen werden. (siehe nachfolgende Abbildung)

Abbildung 24: Darstellung des Löschens eines Nachweises innerhalb einer Blockchain-Transaktion (Quelle: eigene Darstellung)

Dabei wird die im Smart-Contract definierte burn()-Funktion aufgerufen, welcher die Nachweis-ID als tokenID übergeben wird. Nach abgeschlossener Transaktion wird der Nachweis in der Übersicht nicht mehr angeführt und kann – wie in folgender Abbildung dargestellt – auch über die Validierung nicht mehr gefunden werden.

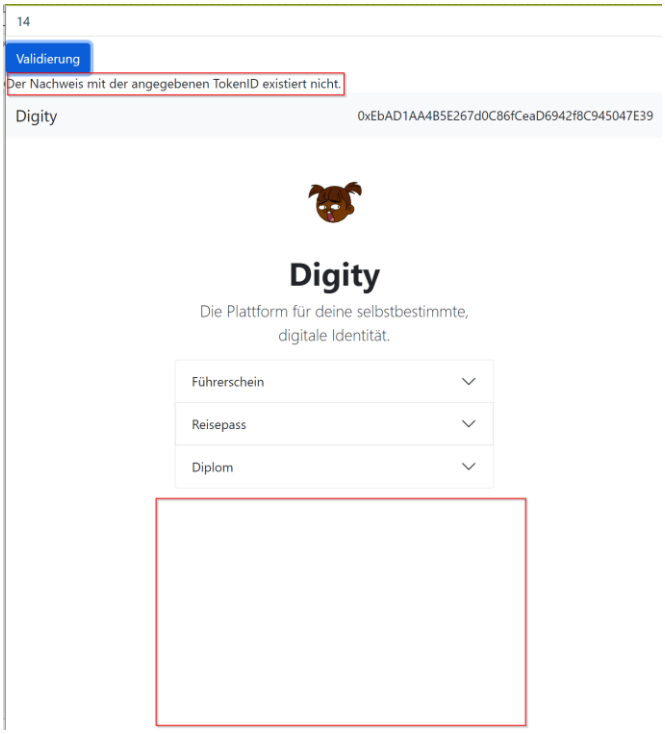

Abbildung 25: Nachweisübersicht nach Löschprozess inkl. Gültigkeitsüberprüfung (Quelle: eigene Darstellung)

4.9 User-Story 9

Als Privatperson möchte ich ungültige Nachweise löschen können, damit mir diese in meinem Profil nicht mehr angezeigt werden.

Der Löschvorgang für Privatpersonen funktioniert analog zu jenem von seriösen Behörden/Unternehmen. Während der Entwicklung des Prototyps wurde entschieden, dass der Burn-Button Privatpersonen jederzeit zur Verfügung gestellt werden soll, da dies die Möglichkeit bietet, dass Nachweise, welche als gültig gekennzeichnet sind, jedoch Fehler in den darin festgehaltenen Daten enthalten, auch gelöscht werden können. Ansonsten wäre ein Löschen von Nachweisen mit Rechtschreibfehlern oder im Falle einer Namensänderung innerhalb des GUI nicht mehr möglich. Die Funktionalität zur Anzeige des Burn-Buttons in Abhängigkeit zur Gültigkeit ist jedoch technisch umsetzbar.

5 EVALUIERUNG UND DISKUSSION

In diesem Kapitel werden die anhand des entwickelten Prototyps dargestellten Ergebnisse dahingehend evaluiert, inwiefern die definierten User-Stories umgesetzt werden konnten, sodass sich daraus eine selbstsouveräne Identität innerhalb der digitalen Welt ergibt. Anhand der Evaluierung werden in weiterer Folge die daraus interpretierten Herausforderungen in der technischen Umsetzung aufgezeigt und es werden diesbezüglich anhand des bereits gewonnenen, theoretischen Wissens, mögliche Lösungsansätze zur Bewältigung dieser Herausforderungen aufgezeigt.

5.1 Evaluierung des Prototyps

Wie bereits in Kapitel 3.2.3 beschrieben, handelt es sich um einen summativen Prototyp, welcher die Einhaltung von definierten Evaluationskriterien und somit die Umsetzbarkeit von theoretischen Annahmen überprüft. Dabei erfolgt die Evaluierung des Prototyps mittels der vier Schritte „Definition von Wertekriterien“, „Definition von Leistungsstandards“, „Messen der Leistungsstandards“ und „Erschließung eines Werturteils“. Die Definition von Wertekriterien und Leistungsstandards erfolgte hierfür bereits in Kapitel 3.5.6. In folgenden Unterkapiteln werden nun diese Leistungsstandards gemessen und es wird ein abschließendes Werturteil in Hinsicht auf die Umsetzung der Leistungsstandards gefällt, wobei in dieser Phase noch nicht auf die damit verbundenen Herausforderungen eingegangen wird. Diese werden anschließend in einem gesonderten Unterkapitel diskutiert.

5.1.1 Messen der Leistungsstandards

Leistungsstandard 1

Wie in Listing 1 ersichtlich, wird innerhalb des Smart-Contracts bei einem Transfer eines NFT der Empfänger/die Empfängerin als neuer „Owner“ angegeben. Auch ist in jeder Funktion definiert (siehe auch Beispiel anhand Listing 1), dass diese nur dann ausgeführt werden kann, wenn es sich bei der aufrufenden Adresse um jene Adresse des Besitzers/der Besitzerin handelt. Die Definition des Besitzers/der Besitzerin innerhalb eines jeden NFT stellt sicher, dass nur dieser/diese in der vollen Kontrolle seiner/ihrer Nachweise ist und damit sämtliche Versuche einer Transaktion mittels abweichender Adresse abgelehnt werden. Ein solcher Sicherheitsaspekt sollte in weiterer Folge auch von einem unabhängigen Auditor garantiert werden, sodass es – durch die Identifizierung von möglichen Fehlern im Sourcecode – zu keiner Zeit zu einem Transfer eines NFT seitens einer unberechtigten Partei kommen kann.

```
function transfer(uint256 tokenID, address to) public {

    require(proofs[tokenID].owner == msg.sender);

    _transfer(msg.sender, to, tokenID);

    proofs[tokenID].owner = to;

    proofs[tokenID].transactionCount++;

    if (proofs[tokenID].transactionCount == 1) {
        proofs[tokenID].validity = true;
    } else {
        proofs[tokenID].validity = false;
    }
}
```

Listing 1: Transfer eines Nachweises (Quelle: eigene Darstellung)

Da dieser Besitz direkt im NFT definiert ist, besteht also keine Abhängigkeit zu zentralen Webapplikationen. Die Kontrolle der eigenen Nachweise ist somit direkt innerhalb der Blockchain festgehalten und kann von jeder beliebigen Webanwendung (welche die notwendigen Funktionalitäten zur Verfügung stellt) und/oder direkt von der Kommandozeile abgerufen werden.

Leistungsstandard 2

Innerhalb der entwickelten Webanwendung ist definiert, dass lediglich jene Parteien, welche sich als seriös qualifizieren, NFT erstellen und transferieren können. Das stellt auf Front-End-Basis bereits sicher, dass unseriöse Parteien (innerhalb des GUI) keine Möglichkeit haben sollten, Nachweise auszustellen und an eine weitere Person zu übermitteln. Dennoch sollte es im Sinne der Selbstbestimmtheit jeder Person möglich sein, die volle Kontrolle über ihre NFT zu haben und damit auch sämtliche Funktionen nutzen zu können, welche im Smart-Contract angeboten werden. Wie in Listing 2 ersichtlich, kann eine Transaktion auch über die Truffle-Konsole ausgeführt werden und somit steht jedem Nutzer/jeder Nutzerin jede – im Smart-Contract definierte – Funktionalität unabhängig von einer Webanwendung zur Verfügung.

```
truffle(ganache)> await digityContract.transfer(2, accounts[2])
{
  tx: '0x85fab810d9518f136c080a33096d41b60a2e30077aec79c14aacb46d96376aa0',
  receipt: {
    transactionHash: '0x85fab810d9518f136c080a33096d41b60a2e30077aec79c14aacb46d96376aa0',
    transactionIndex: 0,
    blockHash: '0xbf09aaf13952f4513fb7d98e9ece1bb3871a32075536a183325d80934ecbc344',
    blockNumber: 519,
    from: '0xebad1aa4b5e267d0c86fcead6942f8c945047e39',
    to: '0xf7922b104a22bca8ecd111822fe714d9e2c7c74f',
    gasUsed: 113580,
    cumulativeGasUsed: 113580,
    contractAddress: null,
    logs: [ [Object], [Object] ],
    status: true,
    logsBloom:
'0x040000000000000000000000000000000800000000000000000001000000000000000000000000
00000020000000000000000000000000000000000000020000000000000000000000000000800000000
00000000000000000000000000000200000000002000000000000000000800000000000000000000
00000100000004000000000000000000000000000000000000000000000000000000000000000002
00000000000000000001000000000000000000000000000000000000000000000000200010000000
00000000000001000000000000000000000000000200021000000000000001000000000000000000
00000080000000000000000000',
    rawLogs: [ [Object], [Object] ]
  }
```

Listing 2: Transaktion über die Truffle-Konsole (Quelle: eigene Darstellung)

In oben angeführtem Beispiel wird ein NFT mit der ID 2 an eine definierte Adresse transferiert. Eine Transaktion über die Truffle-Konsole kann jedoch für jede – im Smart-Contract definierte – Funktion durchgeführt werden.

Leistungsstandard 3

Der entwickelte Smart-Contract beinhaltet bereits eine Funktion, welche einen bestimmten Nachweis mit allen Variablen retourniert (siehe Listing 3). Dabei wird der Funktion eine ID eines Nachweises übergeben und die Gültigkeit wird anhand des transactionCount aktualisiert. Retourniert wird dabei unter anderem auch die Variable „issuer“, welche den Aussteller des betreffenden NFT darstellt. Diese Adresse kann mit jener Adresse des seriösen Unternehmens/der Behörde verglichen und dadurch verifiziert werden.

```
function showValidity(uint256 tokenID) public returns (Proof memory) {

    if (proofs[tokenID].transactionCount == 1) {
        proofs[tokenID].validity = true;
    } else {
        proofs[tokenID].validity = false;
    }

    return proofs[tokenID];
}
```

Listing 3: Abfrage der Gültigkeit eines Nachweises (Quelle: eigene Darstellung)

Diese Information kann seitens der GUI zusätzlich genutzt werden, um die erhaltene Adresse des Ausstellers mit einer Liste von verifizierten Adressen abzugleichen, um eine manuelle Kontrolle der Adresse einzusparen.

Leistungsstandard 4

Wie bereits im Leistungsstandard 1 beschrieben, können sämtliche Transaktionen lediglich von jenem Anwender/jener Anwenderin durchgeführt werden, welcher/welche innerhalb eines NFT als „owner“ angeführt ist. Damit ist es für dritte Parteien auch nicht möglich, einen beliebigen NFT zu löschen, denn auch die Burn-Funktion kann lediglich vom Besitzer/von der Besitzerin des Nachweises für den jeweiligen NFT aufgerufen werden. Dieser Aspekt wird innerhalb des Smart-Contracts mittels folgender Abfrage sichergestellt:

```
require(proofs[tokenID].owner == msg.sender);
```

Listing 4: Abfrage des Besitzers bei Funktionsaufruf (Quelle: eigene Darstellung)

Für das Löschen eines NFT ist somit die Einwilligung des Besitzers/der Besitzerin notwendig, denn auch wenn ein Löschen eines NFT gerechtfertigt ist, sollte lediglich der Besitzer/die Besitzerin dazu in der Lage sein, diese Transaktion durchzuführen.

Leistungsstandard 5

Wie in Listing 5 ersichtlich, wurde im Smart-Contract der Aspekt berücksichtigt, dass ein Nachweis nur dann als gültig interpretiert wird, sofern dieser erst einmal transferiert wurde, denn nur dann ist sichergestellt, dass dieser Nachweis bereits von der ausstellenden Behörde/vom ausstellenden Unternehmen an die betroffene Person übermittelt und daraufhin nicht an eine andere Person weitergeleitet wurde.

```
if (proofs[tokenID].transactionCount == 1) {
     proofs[tokenID].validity = true;
 } else {
     proofs[tokenID].validity = false;
 }
```

Listing 5: Aktualisierung der Gültigkeit eines Nachweises (Quelle: eigene Darstellung)

Die Implementierung der Funktionalität des Aktualisierens der Gültigkeit anhand des Gültigkeitsdatums ist zwar theoretisch möglich, jedoch birgt dies wiederum eine Herausforderung im Sinne der dafür benötigten Transaktionskosten und der Entscheidung darüber, wer diese Transaktionskosten zu tragen hat (siehe Kapitel 5.2.2).

Leistungsstandard 6

Bei einer öffentlichen Blockchain handelt es sich um eine öffentlich-einsehbare, dezentrale Datenbank und sie kann dadurch – per Definition – auch von jedem Teilnehmer/jeder Teilnehmerin eingesehen werden. Bei der Erstellung eines Nachweises werden alle erfassten Informationen an die dafür vorgesehene Funktion übergeben und in einem neuen Block gespeichert. Die darin gespeicherten Daten sind (wie bereits in den Ergebnissen zu User-Story 3 in Abbildung 17 dargestellt) innerhalb der Blockchain im Klartext gespeichert. Das Problem ist hierbei, dass diese (teilweise) sensiblen Informationen nicht nur für jene Person abrufbar sind, welcher der Nachweis zugeordnet ist, sondern auch von dritten Personen aus der Blockchain gelesen werden können. Der Prototyp verfügt also noch nicht über die notwendige Funktionalität, diese sensiblen Informationen vor der Einsicht Dritter zu schützen. Eine mögliche Lösungsvariante hierfür wird in Kapitel 5.2.5 erörtert.

5.1.2 Erschließung eines Werturteils

Wie in der Bemessung der Leistungsstandards zu erkennen ist, wurden bei der Entwicklung die notwendigen Aspekte zur Sicherstellung einer selbstbestimmten, digitalen Identität berücksichtigt. Es wurde darauf geachtet, dass die Besitzer/Besitzerinnen über ihre eigenen NFT vollumfänglich verfügen und alle Transaktionen (welche im Smart-Contract definiert wurden) ausgeführt werden können. Die entwickelte GUI dient dabei lediglich der einfachen Interaktion mit dem Smart-Contract und stellt den Usern/Userinnen die wichtigsten Funktionalitäten zur Verfügung. Die Webanwendung stellt dabei jedoch keine zentrale Instanz dar, die einen Einfluss auf die Kontrolle ihrer Anwender/Anwenderinnen auf ihre eigenen Bescheinigungen hat. Aus dieser notwendigen Selbstbestimmtheit ergeben sich in weiterer Folge jedoch auch zusätzliche Herausforderungen, welche im gegenständlichen Prototyp noch nicht gelöst werden konnten. Doch auch abseits dieses Aspekts ergeben sich bei der Umsetzung eines Prototyps zum Nachweis von identitätsbezogenen Nachweisen Herausforderungen, welche es zu lösen gilt. Durch diese Evaluierung wird das Werturteil getroffen, dass die definierten Leistungsstandards

aus technologischer Sicht und in Hinsicht auf eine unabhängige Identität gut umgesetzt werden können, es jedoch für gewisse Herausforderungen noch Lösungen braucht, um eine – in diesem Sinne – vollumfängliche Identität mittels Blockchain und NFT unter der Berücksichtigung von datenschutzrechtlichen Aspekten zu realisieren. Diese Herausforderungen werden nun in folgendem Unterkapitel genauer dargestellt.

5.2 Herausforderungen und mögliche Lösungsansätze

Sowohl im Rahmen der Entwicklung des Prototyps als auch anhand der anschließenden Evaluierung wurden Herausforderungen auffällig, welche einerseits eine vollständige Eigenverwaltung von identitätsbezogenen Nachweisen erschweren und andererseits dem Konzept der vollständigen Dezentralität widersprechen. Auf diese Herausforderungen wird nun genauer eingegangen und dahingehend diskutiert, ob diesen Problemen technologisch und/oder organisatorisch mittels Lösungsansätzen entgegengewirkt werden kann oder ob eine Behebung von bestimmten Herausforderungen überhaupt notwendig ist, um die Idee einer freien Verwaltung von identitätsbezogenen Nachweisen zu realisieren.

5.2.1 Der Burnmechanismus

Wie bereits in den Ergebnissen und der Evaluierung aufgezeigt, können NFT lediglich von jenen Personen gelöscht werden, welche im Besitz des jeweiligen Nachweises sind. Dies ist eine gewünschte Funktionalität, denn die Besitzer/Besitzerinnen sollte eine uneingeschränkte, alleinige Kontrolle über ihre Nachweise haben. Dieser Aspekt birgt jedoch das Problem, dass diese Bescheinigungen nicht ohne das Zutun der jeweiligen Person entzogen werden können. Sollte es beispielsweise bei einer Führerscheinkontrolle zur Abnahme eines Führerscheins kommen, so muss die Burn-Transaktion vom Besitzer/von der Besitzerin durchgeführt werden und kann nicht von der Exekutive vorgenommen werden. Die Herausforderung besteht dabei also darin, dass eine Möglichkeit gefunden werden soll, dass eine (dazu berechtigte) kontrollierende Person innerhalb des Smart-Contracts die Befugnis dazu hat, den Gültigkeitsstatus eines NFT zu verändern.

Ob diese Herausforderung auf technischer Seite lösbar ist, ist fraglich, denn hierfür müsste das kontrollierende Organ eine Transaktion für einen ihm nicht zugeordneten NFT vornehmen, was wiederum der Definition der Selbstbestimmtheit widerspricht. Besser geeignet wäre hierfür eine Lösung auf organisatorischer Ebene. Wie in Kapitel 4.6 beschrieben, hat jeder Teilnehmer/jede Teilnehmerin des Netzwerks die Möglichkeit, die Gültigkeit eines NFT abzufragen. Sollte ein NFT mit einer bestimmten ID nicht existieren, so wird dies bei der Abfrage auch textlich signalisiert. Kommt es also beispielsweise zu einer Führerscheinabnahme, so kann sich das kontrollierende Organ die ID des Führerscheins vermerken und eine Geldstrafe anordnen, sofern dieser Führerschein nicht gelöscht wird. Das kontrollierende Organ kann dabei zu jeder Zeit den Existenzstatus des Führerscheins mittels zugehöriger ID prüfen und ggf. weitere strafrechtliche Schritte einleiten. Eine weitere mögliche Lösungsmethode wäre, den Gültigkeitszeitraum gewisser Bescheinigungen auf eine Woche zu beschränken, sodass diese wöchentlich erneuert

werden müssen, um eine weitere Gültigkeit vorweisen zu können. Diese Erneuerung müsste in weiterer Folge automatisiert erfolgen, um einen zu hohen Verwaltungsaufwand seitens der Behörden zu vermeiden. Durch diese Lösungsvariante wäre es seitens der Behörde bei einer möglichen Konfiszierung eines Nachweis möglich, diesen innerhalb eines gewissen Zeitraums nicht mehr zu erneuern.

5.2.2 Aktualisierung des Gültigkeitsstatus anhand des Gültigkeitsdatums

Im Rahmen der Entwicklung wurde keine praktikable Möglichkeit gefunden, den Gültigkeitsstatus eines Nachweises an das Gültigkeitsdatum zu binden, denn für jede Statusänderung eines NFT braucht es eine Blockchain-Transaktion, welche wiederum mit Kosten in Form von Transaktionsgebühren verbunden ist. Auf Basis des GUI ist eine manipulierte Ausgabe der Gültigkeit zwar umsetzbar, jedoch behält ein Nachweis innerhalb der Blockchain weiterhin seine Gültigkeit, was zu einer Abhängigkeit von der jeweiligen Webanwendung führt. Wie in Kapitel 3.4.5 ausgeführt, belaufen sich die derzeitigen Ethereum-Transaktionskosten auf über 10 USD, was eine Aktualisierung des Gültigkeitsstatus mit enormen Kosten verbinden würde.

Eine Lösung dieser Herausforderung ist aufgrund der damit verbundenen Kosten nicht zwingend notwendig, da für ein kontrollierendes Organ auch die Möglichkeit besteht, das Gültigkeitsdatum (wie auch bei physischen Nachweisen) manuell zu kontrollieren. Der Gültigkeitsstatus dient hierbei als Absicherung dahingehend, dass ein Nachweis nicht öfter als einmal transferiert wurde und damit einen möglichen Identitätsdiebstahl erschwert. Auch sollte in diesem Zusammenhang angemerkt werden, dass an der Problematik der hohen Transaktionsgebühren seitens Ethereum bereits gearbeitet wird. Mit der bis zum Jahr 2023 geplanten Umstellung von PoW auf PoS soll nicht nur die Anzahl an Transaktionen pro Sekunde steigen, sondern auch die Transaktionsgebühren pro Transaktion wesentlich sinken (Kelly, Millman & Graves, 2022).

5.2.3 Verifikation von seriösen Unternehmen und Behörden

Die wohl größte Herausforderung bei der Umsetzung einer Blockchainanwendung zum Nachweis von identitätsbezogenen Bescheinigungen liegt in der Verifikation der seriösen Unternehmen/Behörden. Wie in Listing 6 ersichtlich, wurden im Rahmen der Prototyp-Entwicklung die Adressen dieser seriösen Unternehmen/Behörden lediglich fest im Sourcecode hinterlegt.

```
const verifiedUsers = [
  "0x8752160504deD58B259AFDb7b53DA52824f89677", //Magistrat Graz
  "0xEbAD1AA4B5E267d0C86fCeaD6942f8C945047E39"]; //Campus02
```

Listing 6: Feste Definition von seriösen Unternehmen/Behörden (Quelle: eigene Darstellung)

Diese Verifikation ist damit zentral an den Anbieter der Webanwendung gebunden und kann damit einerseits von Anbieter zu Anbieter variieren und andererseits stellt dieser Aspekt auch einen zentralen Angriffspunkt für Hacker dar, sodass diese beliebige Wallet-Adressen als „seriös“

einschleusen könnten. Die Herausforderung liegt dabei in der technologischen Realisierung zur Bestimmung von seriösen Unternehmen/Behörden auf Blockchain-Basis. Wie bereits im Theorieteil beschrieben, hat in einem dezentralen Netzwerk wie der Blockchain jeder Teilnehmer/jede Teilnehmerin dieselben Rechte, dennoch können die unterschiedlichen Teilnehmer/Teilnehmerinnen anhand ihrer Wallet-Adressen unterschieden werden. Um den Gültigkeitsstatus eines Nachweises nun an die Seriosität des Ausstellers zu binden, müsste der Smart-Contract auf eine Liste dieser verifizierten Wallet-Adressen zugreifen und den Gültigkeitsstatus abhängig von dieser Liste entsprechend setzen.

Hierbei stellt sich jedoch die Frage, wer in der Lage sein sollte, diese Liste zu befüllen und wie dieses Recht innerhalb des Blockchain-Netzwerks realisiert werden könnte, ohne dass dieses Recht missbräuchlich verwendet werden würde. Eine mögliche Lösung wäre, dass verifizierte Adressen weitere Adressen verifizieren können. So würde das Recht der Verifikation lediglich von seriösen Unternehmen/Behörden ausgehen. Sollte es hierbei jedoch zu einem Irrtum kommen, so könnte dies das gesamte Netzwerk gefährden. Tritt beispielsweise der Fall ein, dass ein verifizierter Teilnehmer/eine verifizierte Teilnehmerin eine Adresse irrtümlich als seriös anerkennt, obwohl es sich dabei möglicherweise um eine Person mit betrügerischen Absichten handelt, so hat schlussendlich auch diese betrügerische Person das Recht, die vermeintliche Seriosität weiterer Wallet-Adressen zu verifizieren, wodurch wiederum der Gültigkeitsstatus eines Nachweises in Frage gestellt werden müsste. Eine einfachere Lösung wäre es, in der URL-Variable des NFT auf die eigene Webseite zu verweisen, welche wiederum die jeweilige Wallet-Adresse aufweist. So kann die Wallet-Adresse des Ausstellers des NFT mit der Wallet-Adresse auf dessen Webseite verglichen werden und dadurch die Gültigkeit eines Nachweises bestimmt werden.

5.2.4 NFT- und Smart-Contract-Struktur

Betrachtet man Listing 7, so kann man erkennen, dass jeder Nachweis dieselbe Struktur hat. Dabei macht es keinen Unterschied, ob es sich um einen Führerschein, einen Reisepass oder ein Diplom handelt.

```
struct Proof {
      uint256 tokenID;
      string description;
      address issuerAddress;
      string issuerName;
      uint256 transactionCount;
      address owner;
      string name;
      uint256 dateOfBirth;
      string nationality;
      string url;
      bool validity;
      uint256 validityDate;
}
```

Listing 7: NFT-Struktur (Quelle: eigene Darstellung)

Bei der Erstellung eines neuen Nachweises muss daher jedes Feld, welches für eine bestimmte Nachweisart (Führerschein, Reisepass, Diplom, usw.) nicht benötigt wird, mit einem Default-Wert (z.B. leerer String) angegeben werden, weil dem Funktionsaufruf ansonsten die benötigten Informationen zur Verarbeitung fehlen. Beispielsweise wird bei einem Reisepass die Nationalität des Besitzers/der Besitzerin benötigt, wohingegen diese Information bei einem Diplom möglicherweise keine Rolle spielt. Für die Erstellung beider Nachweise wird jedoch dieselbe Funktion verwendet, sodass für die Nationalität beim Diplom ein leerer String angegeben werden muss.

Eine mögliche Lösung hierfür wäre, unterschiedliche Strukturen für unterschiedliche Nachweisarten zu definieren (z.B. struct Passport, struct Diploma, usw.). Da ein Smart-Contract im Nachhinein jedoch nicht mehr korrigiert werden kann, wäre es hierbei auch nicht möglich, im Nachhinein weitere Nachweisarten hinzuzufügen. Die unterschiedlichen Nachweisarten müssten somit bei der Veröffentlichung des Smart-Contracts bereits fest definiert sein, was in der Praxis nicht praktikabel ist, denn sollte eine Nachweisart bei der Veröffentlichung vergessen werden, so kann diese im selben Smart-Contract nicht mehr realisiert werden. Eine weitere Lösungsvariante wäre das Auslagern von spezifischen Nachweisinformationen auf die jeweilige Behörde/das jeweilige Unternehmen über die – im NFT angeführte – URL. Dadurch könnte man eine einheitliche Struktur eines jeden NFT schaffen, welcher die notwendigen Informationen für den Gültigkeitsstatus sowie für die Identifikation eines Nachweises beinhaltet. All jene Informationen, welche von dieser Struktur abweichen, können dabei über die verlinkte Webseite der ausstellenden Partei eingesehen werden.

5.2.5 Datenschutz

Wie bereits in Kapitel 4.4 beschrieben und in der darin angeführten Abbildung 17 ersichtlich, werden bei jedem Erstellungsprozess eines Nachweises die Daten in die Blockchain geschrieben. Da die Blockchain für jeden Teilnehmer/jede Teilnehmerin zugänglich ist, können auch sensible Daten wie beispielsweise das Geburtsdatum einer Person von Dritten eingesehen werden. Dies stellt ein Problem im Sinne des Datenschutzes dar, denn diese Informationen sollten lediglich von jener Person gesehen werden können, für welche ein Nachweis auch zugeordnet ist. Auch wurde bereits eine Definition der Selbstbestimmtheit erarbeitet, welche besagt, dass jede Person für sich entscheiden können sollte, an wen sie welche Daten weitergeben möchte, weshalb dieses Problem auch gegen diesen Aspekt verstoßen würde.

Als Lösungsvorschlag könnte hier das kryptografische Schlüsselpaar, bestehend aus privatem und öffentlichen Schlüssel, genannt werden. Damit könnten bei der Erstellung eines Nachweises die in ihm enthaltenen Informationen seitens der ausstellenden Behörde mit dem öffentlichen Schlüssel der betreffenden Person verschlüsselt werden. Die Informationen würden dadurch lediglich in verschlüsselter Form in der Blockchain gespeichert werden und könnten wiederum nur von jener Person entschlüsselt werden, welche im Besitz des zugehörigen privaten Schlüssels ist. Die Realisierung dieses Lösungsvorschlages und inwiefern dies zu weiteren Folgeherausforderungen führt, müsste in einer fortführenden Forschungsarbeit analysiert

werden, jedoch sind die kryptografischen Grundvoraussetzungen für eine solche Umsetzung bereits gegeben.

6 CONCLUSIO

Die Blockchain-Technologie hat in den letzten Jahren einen Aufschwung in Hinsicht auf die dezentrale Speicherung von Informationen erlebt. Es wurden zahlreiche, neue Projekte ins Leben gerufen, welche sich diese Technologie zu Nutze machen wollen, um sich der Realisierung von Web3 – einer digitalen Welt ohne zentrale Instanz – anzunähern. Die Blockchain dient dabei als dezentrale Datenbank, welche aufgrund der kryptografischen Verkettung der einzelnen Blöcke und der Verteilung der Informationen auf zahlreichen Nodes als fälschungssicher angesehen werden kann. Durch die Anwendung von Smart-Contracts und NFT lassen sich einzigartige Besitztümer fälschungssicher und nachvollziehbar in der Blockchain darstellen. Anhand dieser technologischen Grundlagen konnte im Rahmen dieser Arbeit ein Prototyp entwickelt werden, welcher identitätsbezogene Nachweise digital und im Sinne einer selbstbestimmten Identität abbildet. Es konnte somit eine weitere Anwendungsmöglichkeit für NFT in Verbindung mit der Blockchain-Technologie aufgezeigt werden. Es wurden Requirements in Form von User-Stories sowie Wertekriterien und Leistungsstandards definiert, sodass diese anhand einer Evaluierung jene Herausforderungen aufgezeigt haben, welche bei der Umsetzung dieses Prototyps entstanden sind. In Anbetracht der definierten Forschungsfrage „Welche Herausforderungen birgt die Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen?“ können somit folgende 5 Herausforderungen festgehalten werden:

Der Mechanismus zum Löschen einer Bescheinigung sollte nur von jener Person durchgeführt werden können, welcher die jeweilige Bescheinigung zugeordnet ist. Dies birgt die Herausforderung, dass bei einer möglichen Abnahme dieser Bescheinigung das aktive Zutun des Besitzers/der Besitzerin notwendig ist. Es kann innerhalb des Smart-Contracts zwar definiert werden, dass auch andere Parteien dazu befugt sind, gewisse Bescheinigungen zu löschen, jedoch würde dies wiederum der Definition einer Selbstbestimmung widersprechen.

Des Weiteren wurde im Rahmen dieser Arbeit keine Möglichkeit gefunden, den Gültigkeitsstatus eines Nachweises anhand dessen Gültigkeitsdatums zu aktualisieren, ohne dass dabei Transaktionskosten anfallen. Ein Nachweis kann innerhalb eines Smart-Contracts nur dann einen anderen Wert annehmen, wenn eine definierte Transaktion ausgeführt wird, was wiederum mit Transaktionskosten verbunden ist. Dadurch kann das Erneuern des Gültigkeitsstatus eines Nachweises nicht kosteneffizient abgebildet werden.

In einem dezentralen Netzwerk sind sämtliche Teilnehmer/Teilnehmerinnen gleichberechtigt, sodass sämtliche Funktionalitäten des Smart-Contracts auch von jeder Person abgerufen werden können. Dieser Aspekt führt zu dem Problem, dass auch jede Person einen Nachweis ausstellen und an eine beliebige Person transferieren kann. Die Herausforderung liegt dabei in der Identifikation von seriösen Behörden/Unternehmen, sodass originale Bescheinigungen von Fälschungen abgegrenzt werden können. Es besteht zwar die Möglichkeit, anhand der Webanwendung seriöse Wallet-Adressen zu definieren, von welchen die ausgestellten Nachweise als Original angesehen werden können, jedoch führt dies wiederum zu einer zentralen Abhängigkeit und setzt Vertrauen in den Betreiber der Webanwendung voraus. Eine Möglichkeit zur dezentralen Identifikation von seriösen Unternehmen und Behörden konnte in dieser Arbeit

nicht ermittelt werden. In Kapitel 1.2 wurde die Hypothese 2 aufgestellt, dass die Abgrenzung zwischen seriösen und unseriösen Parteien innerhalb einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen nicht möglich ist, da sämtliche Teilnehmer/Teilnehmerinnen als gleichwertige Akteure im Blockchain-Ökosystem agieren können. Diese Hypothese stellt sich demnach als bestätigt heraus.

Da ein NFT innerhalb eines Smart-Contracts eine fixe Struktur aufweist, führt dies je nach Nachweisart zur unnötigen Speicherung von definierten Default-Werten, denn unterschiedliche Nachweise unterscheiden sich auch anhand der unterschiedlichen Informationen, die sie benötigen. Hierbei gilt als Herausforderung, dass eine Möglichkeit gefunden werden sollte, spezifische Informationen vom Smart-Contract auszulagern, unter der Berücksichtigung, dass für diese Informationen trotzdem eine Fälschungssicherheit garantiert werden kann.

Auch konnte eine Herausforderung im Sinne des Datenschutzes erkannt werden. Sämtliche Informationen, die in einer öffentlichen Blockchain abgespeichert werden, sind auch von jeder Person öffentlich einsehbar. Dadurch entsteht das Problem, dass sensible Daten von Dritten in der Blockchain abgerufen werden können. Es muss in diesem Sinne eine Möglichkeit geschaffen werden, diese Informationen innerhalb der Blockchain für dritte Parteien unkenntlich zu machen, sodass lediglich jene Person die Informationen im Klartext angeboten bekommt, welche auch im Besitz des betreffenden Nachweises ist. Diese Herausforderung bestätigt die in Kapitel 1.2 definierte Hypothese 1, welche die Vermutung nahegelegt hat, dass die Einhaltung der Selbstbestimmung über die eigenen Daten eine Herausforderung darstellt, weil nicht gewährleistet werden kann, dass die Daten nicht von dritten Parteien direkt aus der Blockchain ausgelesen werden können.

Für sämtliche Herausforderungen wurden in der abschließenden Diskussion technologische und/oder organisatorische Lösungsmöglichkeiten angegeben, welche aufzeigen sollen, dass eine Umsetzung einer Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen mittels Smart-Contracts und NFT trotz der erkannten Herausforderungen möglich ist. Aus den entstandenen Herausforderungen und Lösungsmöglichkeiten würden sich folgende, noch zu klärende Fragen ergeben:

- Welche rechtlichen Voraussetzungen müssen gegeben sein, um strafrechtliche Schritte einzuleiten, sollte ein digitaler Nachweis bei der Abnahme von einer Privatperson nicht gelöscht werden?
- Welche Möglichkeiten können auf Smart-Contract-Basis geschaffen werden, um den Gültigkeitsstatus eines Nachweises anhand dessen Gültigkeitsdatums zu aktualisieren, ohne dass dabei Transaktionskosten entstehen?
- Wie können in dieser dezentralen Architektur seriöse Behörden/Unternehmen innerhalb eines Smart-Contracts identifiziert werden, sodass der Gültigkeitsstatus eines Nachweises auch an die Seriosität der ausstellenden Partei gekoppelt werden kann?
- Wie können spezifische Nachweisinformationen vom Smart-Contract ausgelagert werden, unter Berücksichtigung, dass auch für diese Informationen eine Fälschungssicherheit gewährleistet werden kann?

- Welche Herausforderungen birgt die kryptografische Verschlüsselung von Informationen innerhalb der Blockchain, sodass diese lediglich von jener Person abgerufen werden können, welche im Besitz des zugehörigen, privaten Schlüssels ist?

Abschließend kann festgehalten werden, dass eine Blockchain-Anwendung zum Nachweis von identitätsbezogenen Bescheinigungen mittels NFT umsetzbar ist, es jedoch noch Herausforderungen zu lösen gilt, welche im Sinne der Selbstbestimmtheit, der Fälschungssicherheit und des Datenschutzes unabdingbar sind.

ABKÜRZUNGSVERZEICHNIS

CSS	Cascading-Style-Sheets
Dapp	Dezentrale Applikation
DOM	Domain-Object-Model
EBSI	European-Blockchain-Services-Infrastructure
GUI	Graphical User Interface
HTML	Hypertext-Markup-Language
ION	Identity Overlay Network
IPFS	InterPlanetary File System
MPA	Multi-Page-Applikationen
NFT	Non-Fungible-Token
PoS	Proof-of-Stake
PoW	Proof-of-Work
SPA	Single-Page-Applikationen
URL	Uniform-Resource-Locator

ABBILDUNGSVERZEICHNIS

LISTINGS

LITERATURVERZEICHNIS

Adobe Inc. (2021). *Grundlegendes zu Webanwendungen*. Zugriff am 27.04.2022. Verfügbar unter: https://helpx.adobe.com/de/dreamweaver/using/web-applications.html

Antonopoulos, A. M. & Wood, G. (2018). *Mastering Ethereum*. Sebastopol: O'Reilly Media.

Bashir, I. (2017). *Mastering Blockchain*. Birmingham: Packt Publishing.

Bergmann, C. (2017). *Adressen bei Kryptowährungen: eine Einführung*. Zugriff am 03.05.2022. Verfügbar unter: https://bitcoinblog.de/2017/06/12/adressen-bei-kryptowaehrungen-eine-einfuehrung/?amp=1

Binance Academy. (2020). *How to Use MetaMask*. Zugriff am 29.04.2022. Verfügbar unter: https://academy.binance.com/en/articles/how-to-use-metamask

Bitfly. (2021). *Miner Statistics*. Zugriff am 27.11.2021. Verfügbar unter: https://etherchain.org/miner

BitInfoCharts. (o.J.). *Ethereum - Transaktionsgebühr*. Zugriff am 27.04.2022. Verfügbar unter: https://bitinfocharts.com/de/comparison/ethereum-transactionfees.html#3y

Brutkasten Media GmbH. (2021). *So nutzt Microsoft die Bitcoin-Blockchain, um Passwörter überflüssig zu machen. Das dezentrale und offene Identitätsnetzwerk ION von Microsoft ist am Bitcoin-Mainnet gestartet.* Zugriff am 21.01.2022. Verfügbar unter: https://brutkasten.com/so-nutzt-microsoft-die-bitcoin-blockchain-um-passworter-uberflussig-zu-machen/

Buchmann, J. (2008). *Einführung in die Kryptographie*. Berlin Heidelberg: Springer Berlin Heidelberg.

Buchner, D. (2019). *Toward scalable decentralized identifier systems*. Zugriff am 20.01.2022. Verfügbar unter: https://techcommunity.microsoft.com/t5/azure-active-directory-identity/toward-scalable-decentralized-identifier-systems/ba-p/560168#

Bundeskanzleramt Österreich. (2020). *Aus Verantwortung für Österreich. Regierungsprogramm 2020–2024*. Zugriff am 08.08.2021. Verfügbar unter: https://www.dieneuevolkspartei.at/Download/Regierungsprogramm_2020.pdf?fbclid=IwAR21w_rl0ktnwWY7LBENj6RDrhq3ep8ybLEp0ivFXDSEhWKRrKS7bw3U3SQ

Cambridge Centre for Alternative Finance. (2021). *Bitcoin Electricity Consumption Index. Country Ranking*. Zugriff am 17.01.2022. Verfügbar unter: https://ccaf.io/cbeci/index/comparisons

Carr, M. & Verner, J. (1997). Prototyping and software development approaches. Zugriff am 12.06.2022. Verfügbar unter: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.1166&rep=rep1&type=pdf

Chevet, S. (2018). Blockchain Technology and Non-Fungible Tokens: Reshaping Value Chains in Creative Industries. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3212662

Christie's. (2021). 10 things to know about CryptoPunks, the original NFTs. Zugriff am 31.12.2021. Verfügbar unter: https://www.christies.com/features/10-things-to-know-about-CryptoPunks-11569-1.aspx

CV Publishing AG. (2021). Das Potenzial von Musik-NFTs. Zugriff am 31.12.2021. Verfügbar unter: https://cvj.ch/fokus/hintergrund/das-potenzial-von-musik-nfts/

Digital World LTD. (2021). *Was ist Decentraland?* Zugriff am 31.12.2021. Verfügbar unter: https://coin-ratgeber.de/was-ist-decentraland/

Doerk, A. (2020). *ESSIF: The European self-sovereign identity framework*. Zugriff am 12.06.2022. Verfügbar unter: https://ssi-ambassador.medium.com/essif-the-european-self-sovereign-identity-framework-4572f6875e12

Dudenredaktion. (o.J.). *fungibel,* Bibliographisches Institut GmbH. Zugriff am 29.12.2021. Verfügbar unter: https://www.duden.de/rechtschreibung/fungibel

Edelman, G. (2021). *The Father of Web3 Wants You to Trust Less*. Zugriff am 29.04.2022. Verfügbar unter: https://www.wired.com/story/web3-gavin-wood-interview/

Eisenbrand, R. (2020). *Metaverse – Das ist gerade eines der heißesten Buzzwords in der globalen Tech-Elite*. Zugriff am 31.12.2021. Verfügbar unter: https://omr.com/de/metaverse-fortnite-facebook-tencent/

Ethereum. (o.J.b). *Language Influences*. Zugriff am 28.04.2022. Verfügbar unter: https://docs.soliditylang.org/en/v0.8.13/language-influences.html

Ethereum. (o.J.c). *Solidity*. Zugriff am 28.04.2022. Verfügbar unter: https://docs.soliditylang.org/en/v0.8.13/

Ethereum. (o.J.a). *web3.js - Ethereum JavaScript API*. Zugriff am 29.04.2022. Verfügbar unter: https://web3js.readthedocs.io/en/v1.7.3/

Europäische Kommission. (o. J.a). *High-level scope (ESSIF)*. Zugriff am 12.06.2022. Verfügbar unter: https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=379913698

Europäische Kommission. (o. J.b). *What is EBSI?* Zugriff am 12.06.2022. Verfügbar unter: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/What+is+ebsi

Europäische Kommission. (2022). *European Blockchain Partnership*. Zugriff am 12.06.2022. Verfügbar unter: https://digital-strategy.ec.europa.eu/en/policies/blockchain-partnership

Fries, M. & Paal, B. P. (2019). *Smart Contracts*. Tübingen: Mohr Siebeck.

Hackr.io. (2022). *What is Bootstrap? Pros and Cons Of This Framework*. Zugriff am 05.06.2022. Verfügbar unter: https://hackr.io/blog/what-is-bootstrap-framework

Hansen, M. & Meints, M. (2006). *Digitale Identitäten – Überblick und aktuelle Trends. Identity-Lifecycle, Authentisierung und Identitätsmanagement*. Zugriff am 08.01.2022. Verfügbar unter: http://www.fidis.net/fileadmin/fidis/publications/2006/DuD09_2006_543.pdf

Hazari, S. S. & Mahmoud, Q. H. (2019). Comparative evaluation of consensus mechanisms in cryptocurrencies. *Internet Technology Letters*. https://doi.org/10.1002/itl2.100

Hegner, M. (2003). Methoden zur Evaluation von Software. Verfügbar unter: https://www.ssoar.info/ssoar/handle/document/50730

Herschel, M. & Adobatti, V. (o.J.). *Bitcoin Sicherheit,* Hochschule Hannover. Zugriff am 27.11.2021. Verfügbar unter: https://herschel.io/assets/pub/Bitcoin_Sicherheit.pdf

Hoeren, T. & Prinz, W. (2021). Das Kunstwerk im Zeitalter der technischen Reproduzierbarkeit – NFTs (Non-Fungible Tokens) in rechtlicher Hinsicht. Was Blockchain-Anwendungen für den digitalen Kunstmarkt bewirken können. *Computer und Recht*. https://doi.org/10.9785/cr-2021-370816

Hornung, G. (2005). *Die digitale Identität*. Baden-Baden: Nomos.

Kelly, L. J., Millman, R. & Graves, S. (2022). *What Is Ethereum 2.0? Ethereum's Consensus Layer and Merge Explained*. Zugriff am 06.06.2022. Verfügbar unter: https://decrypt.co/resources/what-is-ethereum-2-0

Kenning, P. & Lamla, J. (Hrsg.). (2018). *Entgrenzungen des Konsums. Dokumentation der Jahreskonferenz des Netzwerks Verbraucherforschung*. Wiesbaden: Springer Fachmedien Wiesbaden.

Kovac, A. (2021). *React Tutorial für Einsteiger*. Zugriff am 29.04.2022. Verfügbar unter: https://reactjs.de/artikel/react-tutorial-deutsch/#warum-nicht-ganz-ohne-bibliotheken-libraries-und-frameworks

Kuhrmann, M. (2012). *Prototyping,* Enzyklopädie der Wirtschaftsinformatik. Zugriff am 22.04.2022. Verfügbar unter: https://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Prototyping

Kunze, C. P. (2003). Digitale Identität und Identitäts-Management. Zugriff am 08.01.2022. Verfügbar unter: https://vsis-www.informatik.uni-hamburg.de/getDoc.php/thesis/130/identitaet.pdf

Lee, W. M. (2019). *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*. New York City: Apress.

Lewin, M., Dogan, A., Schwarz, J. & Fay, A. (2019). Distributed-Ledger-Technologien und Industrie 4.0. *Informatik Spektrum*. https://doi.org/10.1007/s00287-019-01176-z

Lvivity LLC. (2020). *Single-Page Application vs Multi-Page Application: Pros, Cons, and Which is Better?* Zugriff am 29.04.2022. Verfügbar unter: https://lvivity.com/single-page-app-vs-multi-page-app

Madnick, S. E. (2019). Blockchain Isn't as Unbreakable as You Think. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3542542

Mai, J. (2021). *Selbstbestimmung: Was ist ein selbstbestimmtes Leben?* Zugriff am 08.01.2022. Verfügbar unter: https://karrierebibel.de/selbstbestimmung/

McCubbin, G. (2022). *Intro to Web3.js - Ethereum Blockchain Developer Crash Course*. Zugriff am 29.04.2022. Verfügbar unter: https://www.dappuniversity.com/articles/web3-js-intro

Meta Platforms Inc. (o.J.). *React*. Zugriff am 29.04.2022. Verfügbar unter: https://reactjs.org/

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Zugriff am 04.08.2021. Verfügbar unter: https://bitcoin.org/bitcoin.pdf

Neue Tageskrypto. (2020). *Was ist ein Autoritätsnachweis?* Zugriff am 12.06.2022. Verfügbar unter: https://newdaycrypto.com/de/what-is-a-proof-of-authority/

New York Times Company. (2021). *JPG File Sells for $69 Million, as 'NFT Mania' Gathers Pace*. Zugriff am 04.08.2021. Verfügbar unter: https://www.nytimes.com/2021/03/11/arts/design/nft-auction-christies-beeple.html

Otto, M. (2011). *Bootstrap from Twitter*. Zugriff am 29.04.2022. Verfügbar unter: https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter

Raharjana, I. K., Siahaan, D. & Fatichah, C. (2021). User Stories and Natural Language Processing: A Systematic Literature Review. https://doi.org/10.1109/ACCESS.2021.3070606

Schork, S. (2020). *Methodische Entwicklung aussagekräftiger Prototypen durch Analyse der Produktkonfigurationen*. Darmstadt. https://doi.org/10.25534/tuprints-00014202

Smith, C. (2022). *ERC-721 Non-Fungible Token Standard*. Zugriff am 28.04.2022. Verfügbar unter: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/

Statista. (2020). *Wie häufig haben Sie einen Identitätsdiebstahl im Internet erlebt oder sind Opfer davon geworden?* Zugriff am 04.08.2021. Verfügbar unter:

https://de.statista.com/statistik/daten/studie/541806/umfrage/umfrage-zum-identitaetsdiebstahl-in-oesterreich/

Statista. (2021a). *Videospiele*. Zugriff am 31.12.2021. Verfügbar unter: https://de.statista.com/outlook/dmo/digitale-medien/videospiele/weltweit

Statista. (2021b). *Number of cryptocurrencies worldwide from 2013 to November 2021*. Zugriff am 06.11.2021. Verfügbar unter: https://www.statista.com/statistics/863917/number-crypto-coins-tokens/

Stephan, M., Weisgerber, S., Jacumeit, V., Helfritz, B., Müller, S., Seipel, C. et al. (2018). *Projektbericht Sichere Digitale Identitäten (SDI)*. Zugriff am 08.01.2022. Verfügbar unter: https://www.din.de/resource/blob/306552/1e281ee0a725f5569469af8285ff0183/din-dke-projektbericht-data.pdf

Trufflesuite. (o.J.). *Truffle*. Zugriff am 28.04.2022. Verfügbar unter: https://trufflesuite.com/docs/truffle/

Wätjen, D. (2018). *Kryptographie: Grundlagen, Algorithmen, Protokolle*. Wiesbaden: Springer Vieweg.

Weyers, J. (2021). *Web App – Die interaktive Webanwendung verständlich erklärt*. Zugriff am 27.04.2022. Verfügbar unter: https://www.exovia.de/journal/web-app/

Wilde, T. & Hess, T. (2007). Forschungsmethoden der Wirtschaftsinformatik. https://doi.org/10.1007/s11576-007-0064-z

Wilkens, R. (2019). *Smart Contracts*. Wiesbaden: Springer Vieweg.