MASTERARBEIT

EINSATZ VON BLOCKCHAIN-TECHNOLOGIEN IN UNTERNEHMEN ZUR DOKUMENTATION VON VERTRÄGEN.

ausgeführt an der

am Studiengang
Informationstechnologien und Wirtschaftsinformatik

Von: Michael Kochauf, BSc.
Personenkennzeichen: 51842831

Graz, am …….. .....

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

An dieser Stelle möchte ich mich bei allen Menschen bedanken, die mich bei der Erstellung dieser Masterarbeit unterstützt haben. Zunächst gilt mein Dank an meinem Betreuer, Herr DI Markus Petelinc, BSc, der mir als Betreuer das benötigte Feedback geben konnte, damit ich diese Masterarbeit in dieser Form fertigstellen konnte.

Ein besonderer Dank geht an meinen Bruder Stephan Kochauf, der mich immer mit wertvollem Feedback und großartigen Ideen unterstützt hat. Außerdem möchte ich mich auch bei meiner Familie und ganz besonders bei meinen Eltern Monika und Engelbert Kochauf, sowie meinen Freunden bedanken, die meine Höhen und Tiefen mit mir geteilt haben und mir in allen Situationen immer wieder Zuversicht geschenkt haben.

KURZFASSUNG

Blockchains wurden besonders in den letzten Jahren sehr populär und der Zugang um diese zu Nutzen wurde auch sehr einfach. Aufgrund dieser Entwicklungen wurden viele neue Anwendungsgebiete geschaffen. Die meisten Menschen assoziieren Blockchains mit Kryptowährungen, allerdings gibt es noch weitere Bereiche, die diese Technologie nutzen. Beispiele hierfür sind der Gesundheits- oder der Energiesektor. Immer mehr Unternehmen überprüfen, wie sie diese Technologie ebenfalls nutzen können, um davon zu profitieren und Prozesse zu verbessern. Ein Anwendungsfall, der in dieser Arbeit behandelt wird, ist das Dokumentieren von Dokumenten, Verträgen und Vereinbarungen zwischen Mitarbeiterinnen oder Mitarbeitern und dem Unternehmen. Das Ziel dieser Masterarbeit ist es, zu evaluieren, wie Unternehmen Blockchain-Technologien nutzen können und wie genau Verträge dokumentiert werden können. Es wurde dazu ein Prototyp entwickelt mit dem Verträge in der Blockchain dokumentiert werden können, indem ein Smart Contract verwendet wird. Die Ergebnisse zeigen, dass diese Technologie das Potenzial hat, für diesen Anwendungsfall genutzt werden zu können. Der größte Vorteil dabei ist die Transparenz. Das bedeutet, dass jede Person, die Zugang zur Blockchain hat, nachprüfen kann, wer an diesem Vertrag beteiligt war und wann er abgeschlossen wurde. Allerdings wurden auch einige Probleme festgestellt. Zum Beispiel wird, um Daten in die Blockchain zu schreiben noch mehr Zeit benötigt im Vergleich zu konventioneller Software. Weiters sind bei der Nutzung des entwickelten Prototypen auch öfter Fehler passiert als bei der herkömmlichen Software. Zusammenfassend lässt sich sagen, dass die Verwendung von Blockchain-Technologien zur Dokumentation von Verträgen verwendet werden könnte und es dem Unternehmen einen Mehrwert bieten kann, allerdings ist die Technologie aufgrund der Probleme noch nicht ausgereift genug.

ABSTRACT

Blockchain accessibility is rapidly improving, and many new use cases are emerging. Blockchains are most commonly associated with cryptocurrencies, but there are many applications such as in the health or energy sectors, or as a documentation tool in business. Companies are evaluating how to benefit from this technology in their processes. This thesis explores how companies can use blockchain to document their contracts and agreements between themselves and their employees. A prototype is developed that documents contracts in the blockchain by executing a smart contract. Findings indicate that this technology can be used successfully for this task. The greatest advantage is transparency. Therefore, every user with blockchain access can verify contract participants and the agreement date. However, some problems have been identified. For example, when writing data onto the blockchain, it typically takes longer than utilizing traditional software. Additionally, the use of the prototype resulted in a higher frequency of errors compared to the conventional software. In conclusion, using blockchain technologies for documenting contracts has potential to bring value to companies, but it is not ready yet.

INHALTSVERZEICHNIS

1 EINLEITUNG

Blockchains wurden in den vergangenen Jahren immer bekannter und der Zugang, diese Technologie zu verwenden, wurde für viele Menschen erleichtert. Oft wird dieses Thema mit digitalen Währungen wie etwa dem Bitcoin assoziiert, allerdings gibt es noch weitere Anwendungsbereiche wie beispielsweise das Gesundheitswesen oder der Energiesektor. (W. Yang, Garg, Raza, Herbert, & Kang, 2018)

Da die Verwendung dieser Technologie vergleichsweise einfach wurde und auch die Anzahl an Experten in diesem Gebiet in den letzten Jahren sehr stark gestiegen ist, evaluieren immer mehr Firmen wie sie Blockchains für sich nutzen können, um Prozesse im eigenen Unternehmen zu unterstützen und zu verbessern. (Pinto-Gutiérrez, Gaitán, Jaramillo, & Velasquez, 2022)

Ein Anwendungsbereich wäre das Ablegen oder das Hinterlegen von Dokumenten in der Blockchain. Die Herausforderung dabei ist es, den Besitzer der Dokumente eindeutig mit der realen Person zu verknüpfen und die Authentizität zu gewährleisten. Wenn dies geschafft ist, kann die Historie des Besitzes und die Echtheit von Dokumenten sichergestellt werden.

1.1 Aufgabe

Ziel dieser Masterarbeit ist es, eine sowohl theoretische als auch technische Grundlage zu bilden und folgende Forschungsfrage zu beantworten:

Wie werden Blockchain-Technologien genutzt, um die Gültigkeit von Verträgen zwischen Unternehmen und Mitarbeiterinnen oder Mitarbeitern zu gewährleisten?

Zur Beantwortung dieser Frage wurde ein Prototyp implementiert, um Verträge in der Blockchain zu dokumentieren. Anschließend wurde dieser Prototyp evaluiert und mit dem Wissen, den Erfahrungswerten und der theoretischen Grundlage erarbeitet, ob diese Form der Nutzung in Unternehmen eingesetzt werden kann.

1.2 Aufbau

Der Inhalt dieser Arbeit ist in 6 größere Kapitel gegliedert. Zu Beginn ist die theoretische Grundlage. Diese stellt eine allgemeine Einführung in das Thema Blockchain und deren Verwendungszwecke dar. Dabei wird erläutert, wie diese Technologie von Unternehmen genutzt werden kann und welche Möglichkeiten es gibt, um Verträge sicher in der Blockchain abzulegen und zu dokumentieren. Im nächsten Schritt wird der umgesetzte Prototyp genauer beschrieben und aufgezeigt, wie dieser entwickelt wurde. Die Ergebnisse spiegeln das Resultat der Evaluation dar. Im Zuge der Auswertung der Ergebnisse werden die Probleme, die diese Technologie mit sich bringt, diskutiert. Abschließend folgt eine Zusammenfassung, welche die Forschungsfrage beantwortet.

2 THEORETISCHE GRUNDLAGE

Die Basis dieser wissenschaftlichen Arbeit stellt die theoretische Grundlage aus der Literatur dar. Mithilfe dieser wird die Beantwortung der Forschungsfrage erst möglich. Zu Beginn werden allgemeine Grundlagen zur Blockchain, Smart Contracts und NFTs beschrieben. Anschließend wird noch erläutert, wie Blockchain-Technologien entwickelt werden und wie diese in Unternehmen eingesetzt werden können, um Verträge abzulegen. Eine besonders große Herausforderung dabei ist die Identifikation der realen Person. Abschließend wird noch behandelt, wie Verträge zwischen Unternehmen und Mitarbeitern entstehen.

2.1 Blockchain

In der digitalen Welt der Geschäftsprozesse sind Verlässlichkeit und Vertrauen, dass die Daten richtig sind ein wesentlicher Bestandteil. Um Transaktionen und Daten abzubilden, werden meist traditionelle Datenbanken verwendet. Dabei wird ein zentralisierter Ansatz verfolgt. Das bedeutet, dass eine Autorität die Daten und Transaktionen verwaltet und speichert. Dieser Ansatz hat Risiken wie zum Beispiel Ausfallsicherheit, Authentizität oder Angriffe auf die Integrität. Bei Blockchains wird jedoch ein dezentralisiertes Netzwerk verwendet, welches die Korrektheit von Transaktionen gewährleistet. (Neugebauer, 2018)

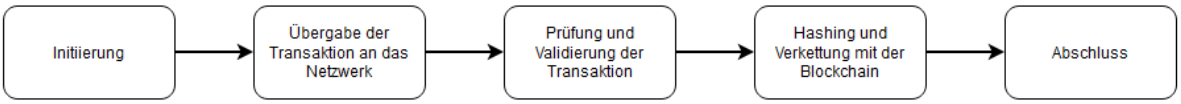
Im Jahr 1991 wurde das erste Mal von einer kryptographisch abgesicherten Verkettung einzelner Blöcke beschrieben. Später im Jahr 2008, wurde diese Idee erneut aufgegriffen und die erste Implementierung einer digitalen Währung, dem Bitcoin, wurde erschaffen. (Mihus, 2022)

Besonders in den letzten Jahren hat diese Technologie sehr viel Aufmerksamkeit von der Öffentlichkeit bekommen und es gehört nun zu den beliebtesten Technologien am Markt. Vorwiegend durch digitale Währungen wie den Bitcoin ist diese Technologie sehr bekannt geworden. (Risius & Spohrer, 2017)

2.1.1 Konzept

Eine Blockchain ist eine erweiterbare Liste von Datensätzen, die in Blöcke aufgeteilt wird. Diese Informationen werden in vollständig verteilten Systemen gespeichert und verifiziert. Das Ziel dabei ist es, konsistente und unveränderbare Daten zu haben, wo man auch eine Historie über den Transaktionsverlauf hat. In solch einem Netzwerk werden somit die Datensätze transparent gespeichert und die Validität der Daten kann systemweit gewährleistet werden. (Risius & Spohrer, 2017)

Um eine Transaktion in der Blockchain abzulegen, werden folgende Schritte durchlaufen:



2-1: Schritte einer Transaktion

Bei der Initiierung wird jede Transaktion von der Erstellerin oder dem Ersteller an das Netzwerk gesendet und digital signiert. Im nächsten Schritt wird diese Transaktion an das Netzwerk übergeben und an alle Knoten verteilt. Anschließend überprüft jeder Knoten die Gültigkeit der Transaktion und es wird versucht, ein Konsens zu finden. Dafür können folgende Ansätze verwendet werden (Neugebauer, 2018):

- Proof-of-Work: Hierbei wird in der Regel eine sehr hohe Rechenleistung benötigt. Im Blockchain-Kontext ist es hierbei das Ziel, dass mit jedem neuen Block, der an die Blockchain angehängt werden soll, eine Zahl (bzw. ein Hashwert) ausgerechnet werden soll, der eine Bedingung erfüllt. Jeder Knoten im Netzwerk rechnet diese Zahl aus und der schnellste ist der Gewinner. (Neugebauer, 2018)
- Proof-of-Stake: Das ist ein Mechanismus, um einen verteilten Konsens zu erreichen. Dabei validieren nicht mehrere Knoten einen einzelnen Block, sondern nur ein Knoten, der anhand gewisser Kriterien oder zufällig automatisch vom Netzwerk ausgewählt wurde. (Neugebauer, 2018)
- Proof-of-Authority: Diese Alternative ist ähnlich zu Proof-of-Stake. Der Unterschied ist, dass bereits im Voraus definiert wurde, welche Knoten Blöcke validieren dürfen und welche nicht. (Kirli et al., 2022)
- Proof-of-Elapsed-Time: Dabei wird der Knoten im Netzwerk, der den Block validieren soll, zufällig nach einem Lotterie-System ausgewählt. (Kumar, Radhesyam, & SrinivasaRao, 2019)

Im vorletzten Schritt wird somit die gültige Transaktion gespeichert und durch Hashfunktionen in ein standardisiertes Format gebracht. (Neugebauer, 2018)

Eine Eigenschaft der Hashfunktion ist, dass diese nur in eine Richtung funktioniert. Das bedeutet, aus einem Input wird man immer denselben Hash bekommen, aber aus diesem Hash kann nicht der Originaldatensatz wiederhergestellt werden. (Chen et al., 2003)

In der Blockchain wird jeder Block mit dem Hashwert des vorherigen Blocks verknüpft und anschließend wird daraus ein neuer Hash generiert. Ein Block lässt sich somit eindeutig repräsentieren. Diese Methode ist auch manipulationssicher, da eine Änderung in der Transaktion den Hashwert verändern würde und somit die Blockchain nicht mehr konsistent wäre. (Neugebauer, 2018)

2.1.2 Anwendungsgebiete

Blockchains ermöglichen es, digital Transaktionen abzuwickeln, ohne Instanz in der Mitte. Das bedeutet, dass zwei Parteien diese transparent tätigen können. Das hat das Potenzial, viele konventionelle Geschäftsprozesse zu verändern. Besonders von dieser Veränderung betroffen sind Banken, Versicherungen und Instanzen, die Informationen authentifizieren und lizenzieren. (Abou Jaoude & George Saade, 2019)

Beispiele für Anwendungsbereiche sind(Abou Jaoude & George Saade, 2019):

- IoT: Das Internet of things ist ein ganz großer Profiteur dieser Technologie. Besonders wegen der hohen Datensicherheit und Nachvollziehbarkeit bei Geräten die miteinander Daten austauschen. (Abou Jaoude & George Saade, 2019)
- Energiesektor: Ähnlich wie bei IoT können Energielieferungen zwischen Verbraucherinnen und Verbrauchern und Erzeugerinnen und Erzeugern sicher und eindeutig dokumentiert werden. (Andoni et al., 2019)
- Gesundheitswesen: Hierbei könnten medizinische Daten fälschungssicher archiviert werden und dass, ohne dafür eigene Systeme aufzubauen. (Hölbl, Kompara, Kamišalić, & Nemec Zlatolas, 2018)
- Finanzwirtschaft: Besonders auf die Finanzindustrie hat das Thema Blockchain enormen Einfluss genommen. Geldtransaktionen können, wie es bei digitalen Währungen wie den Bitcoin bereits der Fall ist, transparent in der Blockchain abgebildet werden. (Treleaven, Gendal Brown, & Yang, 2017)
- Smart Government: Dadurch werden mit Hilfe der Blockchain virtuelle Notardienstleistungen beim Handel von Immobilien oder Unternehmen denkbar. Durch die Transparenz kann somit auch der Steuerhinterziehung entgegengewirkt werden, da beispielsweise Steuererklärungen durch intelligente Apps automatisch gemacht werden. (Brühl, 2017)

Es gibt allerdings noch viele weitere Anwendungsbereiche für die Blockchain und es werden täglich auch weitere entdeckt. Besonders in Bereichen, wo zwei Parteien etwas transparent und sicher Dokumentieren müssen, kann diese Technologie eingesetzt werden. (Abou Jaoude & George Saade, 2019)

2.1.3 Arten von Blockchains

Bei Blockchains gibt es verschiedene Arten und Weisen, wie diese funktionieren und umgesetzt sind. Grundsätzlich kann man sie in drei Kategorien einteilen:

1. Public Blockchain:

   Das ist die bekannteste Form einer Blockchain. Sie ist dezentral und auf vielen Servern, auch Nodes genannt, verteilt. Ziel dabei ist es, dass es keinen einzelnen Besitzer dieser Datenbank gibt und sie somit niemanden allein gehört. Auf jedem Node im öffentlichen Netzwerk sind alle Transaktionen gespeichert und auch jeder kann diese Transaktionen einsehen. Weiters kann auch jeder mit seinem eigenen Computer an diesem Netzwerk teilnehmen und seine Ressourcen zur Verfügung stellen. Ein Beispiel für so eine Blockchain ist Bitcoin oder Ethereum. (Düring & Fisbeck, 2017)

2. Consortium Blockchain: Bei dieser Art von Blockchain wird die Konsistenz durch definierte Knoten validiert. Dabei benötigt man weniger Nodes als bei einer Public Blockchain. Ein Vorteil davon ist, dass weniger Daten im Netzwerk verschickt werden müssen. Allerdings sind die Transaktionen nicht mehr von jedem öffentlich einsehbar. Deshalb ergibt sich der Nachteil, dass diese Art nicht so viel Transparenz erlaubt im Vergleich zur Public Blockchain. (Dib, Brousmiche, Durand, Thea, & Hamida, 2018)

3. Private Blockchain:

   Private Blockchains sind wie es der Name bereits vermuten lässt, nicht für die Öffentlichkeit zugänglich. Diese Art wird meistens von jemanden verwaltet. Das hat den Vorteil, dass auch nur den Akteuren Zugriff gegeben werden kann, die aus auch benötigen. Weiters gibt es nur eine bestimme Anzahl an Nodes, die für das Netzwerk zugelassen sind, um die Transaktionen zu validieren. Trotzdem ist es das Ziel, dass die Knoten im Netzwerk verteilt über mehrere Standorte sind. Der Vorteil hierbei ist es, dass es aufgrund der geringeren Anzahl an Knoten viel schneller zu einem Konsens kommt. Das bedeutet, diese Art von Blockchain kann darauf ausgelegt werden, sehr viele Transaktionen zu prozessieren. (R. Yang et al., 2020)

   Beispielsweise bietet Amazon an, sich eine private Blockchain zu erstellen. Dabei wird durch AWS die Infrastruktur bereitgestellt und die Herausgeberin oder der Herausgeber kann die gewünschten Parameter der Blockchain auf die geforderten Anforderungen konfigurieren. (Beaumier & Kalomeni, 2021)

2.2 NFT

Seit 2020 ist die Beliebtheit von NFTs sehr stark gestiegen und heute gehört es zu den bekanntesten Teilbereichen von Blockchains und Kryptowährungen. Bei einem NFT (Non-Fungible Token) handelt es sich um einen kryptografisch eindeutigen, unteilbaren und unersetzbaren Token, der einen Gegenstand, egal ob physisch oder digital in der Blockchain repräsentiert. Bekannt wurden NFTs in den letzten Jahren besonders durch digitale Kunstwerke, wobei die Originalität und der Besitz dieser eindeutig zugewiesen werden konnte. (Pinto-Gutiérrez et al., 2022)

Diese Technologie macht es möglich, dass der Besitz eines digitalen Objektes nachgewiesen werden kann. Weiters können diese digitalen Güter transparent weitergehandelt werden. Besonders für Künstlerinnen und Künstler hat sich durch diese Technologie eine neue Einnahmequelle offenbart. Für sie ist es möglich, ihre Kunstwerke online zu verkaufen und den Käuferinnen und Käufern die Authentizität des Werkes zu gewährleisten. (Valeonti et al., 2021)

Der Großteil der derzeitigen NFTs sind mit der Ethereum-Blockchain entwickelt. Dazu benötigt es einen Smart Contract. Auf Smart Contracts wird im Kapitel 2.3 noch genauer eingegangen. (Valeonti et al., 2021)

2.2.1 Struktur eines NFTs:

Jeder Token ist ein Key-Value-Pair. Das bedeutet, der Key ist die Identifikation des Tokens. Diese wird benötigt, um den Token eindeutig im Netzwerk zu identifizieren. In der Value werden alle Informationen und Daten gespeichert, die zu dem NFT gehören. (Karandikar, Chakravorty, & Rong, 2021)

Das sind beispielsweise folgende Daten:

- Die Besitzerin der Besitzer
- Notizen
- Link zum Dokument
- …

Die Value muss sich nicht nur auf diese Daten beschränken, sondern dies kann je nach NFT angepasst werden, um die benötigten Informationen zum gegebenen Anwendungsfall zu speichern.

2.2.2 Arten von NFTs

Grundsätzlich ist ein NFT immer ein digitales Objekt, dass von einer Person besessen werden kann und dieser Besitz ist immer in der Blockchain dokumentiert. Der Großteil der bestehenden NFTs kann in folgende sechs Gruppen unterteilt werden (Bao & Roubaud, 2022):

- Art
- Collectibles
- Games
- Metaverse
- Other
- Utility

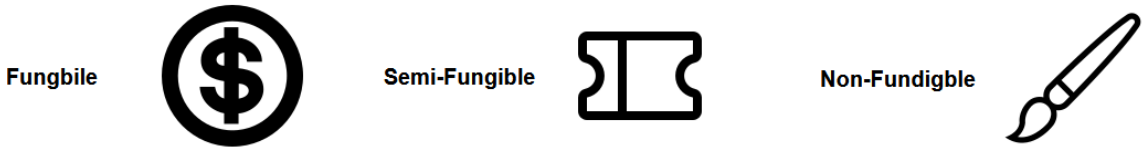
Je nach Einsatzzweck oder Art des Gegenstands kann man ein NFT in eine dieser Kategorien klassifizieren. Es ist allerdings noch möglich, dass sich die Anzahl der Typen in der Zukunft noch erhöht und die Anzahl der Typen unendlich groß ist. (Bao & Roubaud, 2022)

2.2.3 Unterschied zu Fungible Tokens

In der Welt der Blockchains und Kryptowährungen gibt es einige verschiedene Standards. Je nach Standard gibt es Fungible Tokens, Non-Fungible Tokens und Semi-Fungible Tokens.

Fungible Tokens sind homogen austauschbar. In der realen Welt könnte man beispielsweise Geldnoten oder Edelmetalle als Fungible sehen. Auch ein Bitcoin ist ein Fungible Token. Ein Coin hat dieselben Eigenschaften und ist genau gleich viel Wert wie ein anderer Coin. Es ist zwar möglich, einzelne Bitcoins als unterscheidbar zu markieren, aber dieser Coin hat trotzdem dieselben Eigenschaften wie ein anderer. Wenn man einen Bitcoin als „Distinguishable“ markiert, nennt man ihn auch Colored Coin. Die Funktionalität einzelne Coins zu markieren, war im Laufe der Zeit allerdings zu wenig und somit wurde im Jahr 2017 das erste Ethereum-based NFT erschaffen. Wenn etwas Non-Fungible ist, dann kann es zwar auf den ersten Blick gleich aussehen, hat aber im Hintergrund einzigartige Informationen und Eigenschaften. Es war hiermit möglich, eine gewisse Seltenheit eines digitalen Gutes zu generieren. (Bamakan, Nezhadsistani, Bodaghi, & Qu, 2022)

Semi-Fungible Tokens funktionieren zu Beginn ähnlich wie Fungible Tokens. Das bedeutet, sie können beliebig oft ausgeschüttet werden. Zu einem bestimmten Zeitpunkt werden die Bedingungen eines Smart Contracts erfüllt und ab diesem Moment kann der Token nicht mehr weiter erstellt werden. Die bestehenden Tokens haben somit eine gewisse Seltenheit. Ein Beispiel für Semi-Fungible Tokens in der realen Welt wären Konzert-Tickets. Durch ein Ticket kann jede Person zum selben Konzert gehen, allerdings ist diese Karte an einem anderen Zeitpunkt nicht mehr gültig. (Singh & Singh, 2021)



2-2: Token-Unterschiede

2.2.4 Krypto-Standards

Um die einzelnen Kryptowährungen, Blockchains und deren Funktionalitäten und Eigenschaften zu klassifizieren, wurden für die Ethereum-Blockchain unterschiedliche Standards eingeführt. Folgende drei sind die wichtigsten, in denen die meisten Kryptowährungen sind (Bamakan et al., 2022):

- ERC-20:

  In der Ethereum-Blockchain implementieren einige Smart Contracts eigene Tokens. Diese Tokens kann man theoretisch als Sub-Währung sehen. Mit diesem Standard werden die fungiblen Tokens abgebildet. Diese Tokens können einen monetären Wert haben. Dabei ist es für Entwicklerinnen und Entwickler sehr einfach, neue Tokens zu erstellen. Der größte Vorteil dieses Tokens liegt darin, dass er sich auch mit anderen Tokens aus der Ethereum-Blockchain austauschen lässt. (Bauer, 2022)

- ERC-721:

  Bei diesem Standard handelt es sich um nicht-fungible Tokens. Das bedeutet, sie sind wie im vorherigen Kapitel beschrieben, nicht austauschbar. Dies unterscheidet diesen Standard vom ERC-20. Denn hier ist jeder Token einzigartig in der Blockchain. Mithilfe dieses Standards können NFTs erschaffen werden. (Bauer, 2022)

- ERC-1151:

  Dieser Standard kombiniert die beiden Standards ERC-20 und ERC-721. Hierbei ist es möglich, dass man sowohl austauschbare als auch nicht austauschbare Tokens entwickelt. Weiters ist es mit diesem Standard auch möglich, die im vorherigen Kapitel beschriebenen Semi-fungible Tokens abzubilden. Zusätzlich ist es bei diesem Standard auch möglich, Tokens zu gruppieren. Dies hat den großen Vorteil, dass man gleich mehrere Tokens mit einer einzigen Transaktion verschieben kann. Hierbei wird somit das Netzwerk der Blockchain nicht so stark belastet. (Ethereum, 2022)

ERC steht dabei für Ethereum Request for Comments. Hiermit werden gewisse Regeln und Eigenschaften der Tokens definiert und standardisiert.

2.2.5 Vorteile

Der größte Vorteil eines NFT ist, dass der Besitz eines digitalen Gegenstandes in der Blockchain protokolliert ist und dieser Eintrag nicht verändert, werden kann. Es ist somit möglich, nachzuweisen, wer diese Datei erstellt, in der Vergangenheit besessen hat oder derzeit besitzt. Dadurch ist die gesamte Historie des Besitzes transparent nachvollziehbar. (Valeonti et al., 2021)

Wie bereits beschrieben, werden Smart Contracts benötigt, um NFTs zu erstellen und in der Blockchain zu verwalten. Diese bieten nun auch die Möglichkeit, die Erstellerin oder den Ersteller immer zu benachrichtigen, wenn das NFT die Besitzerin oder den Besitzer wechselt. Somit hat der Herausgeber immer die Übersicht, wer das digitale Objekt gerade besitzt. (Kugler, 2021)

2.2.6 Nachteile

Wenn bei einem NFT ein digitaler Gegenstand verlinkt wird, dann muss dieser auch irgendwo abgespeichert werden. Festplatten und andere Speichermedien können kaputt werden und wenn man die Datei auf Cloud Servern ablegt, gibt es auch das Problem, dass die Server nach einigen Jahren abgeschaltet oder der Speicherort verschoben werden können. In solchen Fällen würde dann der Link zum NFT in der Blockchain nicht mehr übereinstimmen und das NFT hätte keine Datei hinterlegt. In diesem Fall wäre dieser Token wertlos. (Valeonti et al., 2021)

Daraus geht das Risiko, dass wenn die Dateien von NFTs zentral auf Servern abgelegt werden, die Besitzer davon nicht sicherstellen können, dass sie auf diese Dateien für immer zugreifen können. Betreiber von Storage-Servern haben somit die Möglichkeit, über die Dateien der NFT-Besitzerinnen und Besitzer zu verfügen. (Valeonti et al., 2021)

Weiters ist es derzeit auch nicht rentabel, die Dokumente oder Dateien direkt in der Blockchain abzulegen, da in der Blockchain selbst der Speicherplatz sehr teuer ist. Im Jahr 2020 hat 1 Megabyte in der Ethereum-Blockchain mehr als 13.000 USD gekostet. (Valeonti et al., 2021)

Ein weiterer Nachteil entsteht beim Handel von NFTs, wenn einer der Beteiligten für die Transaktionskosten aufkommen muss, um in die Blockchain zu schreiben. Dieser Betrag setzt sich in den meisten Fällen aus der derzeitigen Menge an Transaktion, die gerade im Netzwerk erfolgen, zusammen. Wenn im derzeitigen Moment gerade viele Transaktionen verarbeitet, werden müssen, ist der Preis höher, als zu einem Zeitpunkt, an dem gerade wenige Transaktionen zu prozessieren sind. Dabei muss verhandelt werden, wer von den Beteiligten diese Kosten übernimmt. (Valeonti et al., 2021)

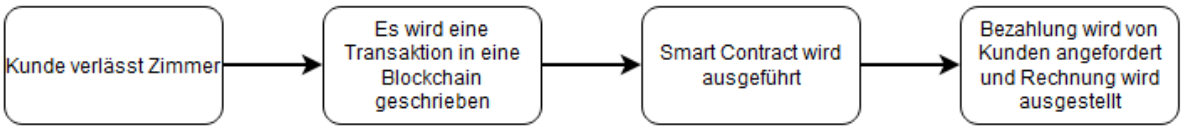
2.3 Smart Contracts

Ein Smart Contract ist eine Software, die von den Computern in der Blockchain ausgeführt wird. Sie können dazu genutzt werden, um Verträge in der Blockchain abzubilden und gewisse Parameter zu überprüfen. Wenn ein Smart Contract ausgeführt beziehungsweise ausgelöst wird, können weitere definierte Handlungen initiiert und durchgeführt werden. Ein Beispiel ist, dass Zahlungen unter gewissen Bedingungen automatisch veranlasst werden. (Meitinger, 2017)

Der Begriff Smart Contract wurde allerdings bereits lange vor Blockchains erfunden. Im Jahr 1994 wurde er definiert als ein digitales Transaktionsprotokoll, dass vertragliche Bedingungen wie zum Beispiel Zahlungsbedingungen, Vertraulichkeit oder auch Vollstreckungsbedingungen beinhaltet. Ziel dabei ist es, die bisherigen vertrauten Zwischenparteien wie zum Beispiel Notare zu ersetzen. Später im Jahr 1997 wurde definiert, dass Smart Contracts eine Kombination von Protokollen mit Nutzeroberflächen sind, um formale und Sicherheitsbeziehungen zwischen unterschiedlichen Netzwerken festzulegen. (Ante, 2021)

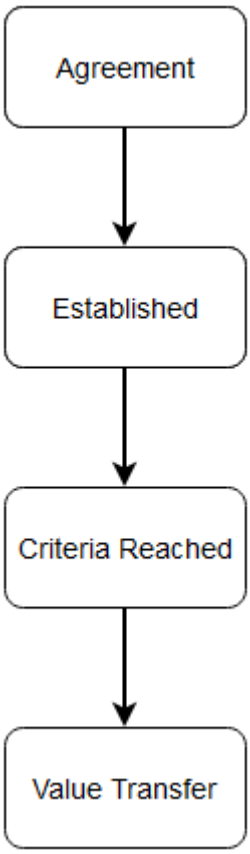
Der Begriff Smart und Contract sind etwas irreführend, da es eigentlich nur Computer-Code ist, der keine rechtlichen Bedingungen erfüllt. Allerdings kann dieser Code dazu genutzt werden, um rechtliche Bedingungen von Verträgen automatisiert sicherzustellen. (Ante, 2021)

Ein Beispiel, wie ein Smart Contract genutzt werden kann, ist ein Hotel Management System. Wenn der Gast das Zimmer verlässt, wird automatisch eine Transaktion in die Blockchain geschrieben. Diese löst anschließend den Smart Contract aus. Diese Software sorgt dafür, dass das Hotel bezahlt wird und eine Rechnung an die Kundin oder den Kunden ausgestellt wird. (Ante, 2021)



2-3: Hotelzimmer Beispiel

In anderen Worten besteht ein Smart Contract oder der Prozess davon aus folgenden vier Schritten (Kirli et al., 2022):



2-4: Smart Contract Prozess

Im ersten Schritt müssen sich alle Parteien dazu entscheiden, die Transaktion zu schreiben und den Smart Contract auszulösen. Im zweiten Schritt wird für das Schreiben der Transaktion alles vorbereitet. Somit sind alle Vorbedingungen erfüllt, um den Code auszuführen. Im nächsten Schritt wird im Smart Contract überprüft, ob alle Kriterien erfüllt werden, um die Transaktion und die definierten Schritte erfolgreich auszuführen. Wenn dieser Schritt erfolgreich erledigt wurde, wird im letzten Schritt die Transaktion in die Blockchain geschrieben und der Smart Contract ausgeführt. Folgendermaßen werden auch alle im Smart Contract definierten Befehle ausgeführt. (Kirli et al., 2022)

2.3.1 Vorteile

Der größte Vorteil von Smart Contracts ist die Dokumentation aller Ereignisse in der Blockchain. Keine Transaktion und auch kein anderer Schritt, der durchgeführt wurde, um dem Smart Contract auszuführen, kann verloren gehen. Jeder in der Blockchain ist im Anschluss dazu fähig, nachzuvollziehen, ob alle Schritte richtig abgelaufen sind. (Negara, Hidayanto, Andryani, & Syaputra, 2021)

Durch die Transparenz der Blockchain ergibt sich auch automatisch der nächste Vorteil. Jede Person, die Zugriff auf das Netzwerk hat, kann auch auf die geschriebenen Transaktionen des Smart Contracts zugreifen. Somit ist für jede und jeden klar, welchen Inhalt der Vertrag hat. (Negara et al., 2021)

Ein weiterer Vorteil ist, dass der Code von einem Smart Contract nur sehr schwer verloren gehen kann, da er auf jedem Knoten im Netzwerk der Blockchain vorhanden sein muss. Weiters kann dadurch auch die Integrität des Codes sichergestellt werden. Es ist somit nicht möglich, dass der Code durch Dritte verändert wurde. (Negara et al., 2021)

Die Geschwindigkeit und Zuverlässigkeit ist auch ein ausschlaggebender Vorteil. Ein Smart Contract wird in dem Moment ausgeführt, in dem die definierten Bedingungen erfüllt werden. Dadurch, dass der Code auf allen Nodes im Netzwerk gespeichert ist, kann auch eine schnelle Abwicklung gewährleistet werden. Ebenso, da es sich um ein verteiltes System handelt, ist die Ausfallsicherheit sehr hoch. Wenn ein Knoten im Netzwerk ausfällt, gibt es immer noch weitere, die die Arbeit auch übernehmen. (Kirli et al., 2022)

2.3.2 Nachteile

Ein besonders großer Nachteil ist, dass das Abspeichern des Codes in der Blockchain nicht effizient ist. Das bedeutet, jeder Node im Netzwerk muss eine Kopie der Software gespeichert haben, damit dieser auch ausgeführt werden kann. Da es besonders schwierig ist, in einer Blockchain etwas zu löschen, existiert auch das Problem, dass veraltete Smart Contracts nicht gelöscht werden können. Das bedeutet, im Laufe der Zeit wird es sehr viel obsoleten Code geben, der noch immer auf den Computern im Netzwerk gespeichert bleibt. (Ante, 2021)

Um bei dem Nachteil mit dem Löschen anzuschließen. Bei Smart Contracts werden oft zusätzliche Daten der Parteien, die die Transaktion ausgelöst haben, gespeichert. Wenn nun eine Person möchte, dass die Daten gelöscht werden, ist das ebenso schwer möglich. Dadurch können in vielen Fällen rechtliche Anforderungen nicht eingehalten werden. (Ante, 2021)

Aufgrund dessen, dass Smart Contracts in der Blockchain abgespeichert werden, können sie wie bereits beschrieben nicht mehr verändert werden. Falls nun Sicherheitslücken oder andere Fehler auftauchen, können bereits abgewickelte Transaktionen nicht mehr auf den neuesten Stand gebracht werden. Auch der bestehende Code kann nicht verändert werden, um die Lücke zu schließen. Somit muss in solchen Fällen ein neuer Smart Contract in die Blockchain aufgenommen werden. (Kirli et al., 2022)

2.3.3 Entwicklung eines Smart Contracts

Die beliebteste Art und Weise, um einen Smart Contract zu implementieren, ist über eine private distributed ledger die auf Ethereum basiert. Dabei wird der Smart Contract auf die Blockchain geladen und darin ausgeführt. Um einen Smart Contract zu erstellen, müssen folgende Schritte durchgeführt werden:

1. Konfiguration der lokalen Blockchain mit Nodes und dem Account
2. Entwickeln des Codes des Smart Contracts. Bei der Ethereum-Blockchain werden hierfür die Programmiersprachen Solidity oder Vyper verwendet.
3. Den Smart Contract kompilieren
4. Den kompilierten Code in die Blockchain laden.
5. Den Smart Contract ausführen.

Wie im vorherigen Kapitel beschrieben, ist es nur sehr schwer möglich, Fehler von Smart Contracts zu beheben. Um von Beginn an möglichst guten Code für die Smart Contracts zu entwickeln, gibt es bereits gewisse Vorlagen, die von mehreren Instanzen auf deren Fehler geprüft wurden. Somit ist es ratsam, die Anforderungen, die man umsetzten, möchte, genauestens zu prüfen und zu evaluieren, ob es bereits eine Vorlage gibt, die gewissen Teile der Anforderungen abdeckt. Dadurch kann die Wahrscheinlichkeit verringert werden, dass schlussendlich Fehler im Code sind. (“2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER),” 2021 - 2021)

Eine Plattform, die diese Vorlagen bietet, ist Openzeppelin. Diese wird von einer Community von Entwicklerinnen und Entwicklern verwaltet, die sich zum Ziel gesetzt haben, die Sicherheit und Effizienz von Smart Contracts zu verbessern. Es wurde untersucht, dass in Smart Contracts regelmäßig duplizierter Code verwendet wird. Dabei wurde meistens nicht die Codebasis von Openzeppelin verwendet. Diese Untersuchung hat gezeigt, dass 79,1% der Smart Contracts Codeteile wiederverwendeten, aber nur 18,4% von diesem Code von Openzepplin stammt. Dabei kann die Blockchain davon sehr stark profitieren, wenn bei dupliziertem Code jener verwendet wird, der bereits von mehreren Entwicklerinnen und Entwicklern begutachtet wurde. (“2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER),” 2021 - 2021)

2.4 Entwicklungsprozess

Der Prozess der Entwicklung eines Smart Contracts oder eines NFTs läuft ähnlich ab wie der Prozess, um eine Software zu entwickeln. Das bedeutet, man kann gewisse Modelle der Software-Entwicklung auf die Entwicklung von Blockchain-Technologien übertragen und nutzen. In typischen Software-Projekten gibt es folgende sechs Phasen:

1. **Requirements**

   Am Anfang jedes Projektes müssen die Anforderungen an die Software definiert werden. Das bedeutet, was genau soll umgesetzt werden und welches Ziel beziehungsweise welches Problem soll mit der Software gelöst werden. Weiters stellt man sich auch die Frage, welcher Nutzen für die Anwenderinnen und Anwender generiert wird. Diese Phase ist besonders wichtig, da die zukünftige Software anhand dieser Beschreibung erstellt wird. Spätere Änderungen könnten nur mehr schwerer umsetzbar sein. (K.Pandey & Batra, 2013)

2. **Design**

   In dieser Phase wird die Grundstruktur der Software definiert. Hierbei wird die Technologie sowie auch die Architektur festgelegt, in der entwickelt wird. Dabei unterscheidet man zwischen Grob- und Feinentwurf. Beim Feinentwurf wird dann ein detailliertes Konzept erstellt, wie die Software umgesetzt werden soll. (Crnkovic, 2001)

3. **Programming/Unit Testing**

   In diesem Schritt werden die definierten Anforderungen in einzelnen Komponenten umgesetzt. Die einzelnen Komponenten, die dabei entstehen, werden anschließend von den Entwicklerinnen und Entwicklern getestet. (Ammann & Offutt, 2017)

4. **Integration Testing**

   In diesem Schritt werden alle entwickelten Komponenten zu einer Software zusammengesetzt. Weiters wird anhand gewisser Techniken getestet und geprüft, ob die Software, Fehler aufweist. Falls Fehler entdeckt werden, sollten diese direkt behoben werden. (Ammann & Offutt, 2017)

5. **System Testing**

   Hierbei wird überprüft, ob die Software allen Anforderungen und den Design-Ansprüchen aus den ersten beiden Schritten entspricht. Weiters wird das gesamte Verhalten und die Benutzerfreundlichkeit der Software überprüft. Falls hier noch größere Fehler auftauchen, ist dies sehr schlecht für den weiteren Verlauf des Software-Projektes. Fehler in dieser Phase bedeuten oft größere Anpassungen in der Software. (Ammann & Offutt, 2017)

6. **Deployment and Operation**

   Im letzten Schritt wird die Software ausgeliefert. Das bedeutet, sie kann anschließend von Kundinnen und Kunden genutzt werden. Weiters beinhaltet dieser Schritt die Wartung, denn Software muss im Laufe der Zeit immer wieder gewartet und auf den neuesten Stand gebracht werden. (Ruparelia, 2010)

Für die Entwicklung von Blockchain-Technologien können ebenfalls diese sechs Phasen angewendet werden. Das bedeutet im Kapitel Beschreibung des Prototyps wird genauer darauf eingegangen, wie diese Phasen im Zuge dieser Arbeit abgelaufen sind.

2.5 Einsatz von Smart Contracts und NFTs zum Abbilden von Verträgen

Wie in Kapitel 2.1.2 beschrieben, gibt es einige Einsatzgebiete für Blockchains und deren unterschiedlichen Funktionalitäten. Auch Unternehmen haben mittlerweile begonnen, diese Technologien für sich zu nutzen und versuchen damit gewisse Prozesse zu verbessern. Der Einsatzzweck, der bei dieser Arbeit genauer beleuchtet wird, ist das Dokumentieren beziehungsweise das Abbilden von digitalen Verträgen in der Blockchain.

2.5.1 Rechtliche Einordnung

Smart Contracts müssen nicht notwendigerweise einen Vertrag abbilden. Ein Vertrag setzt sich in der Regel aus zwei korrespondierenden Willenserklärungen zusammen. Weiters müssen schriftliche Verträge gewisse Informationen enthalten. Bei einem Kaufvertrag sind das beispielsweise Informationen zu den Vertragspartnern. Ein Smart Contract ist wie in den vorherigen Kapiteln beschrieben, ein Protokoll, welches unter gewissen Bedingungen gewisse Aktionen automatisiert ausführt. (Hoffmann & Skwarek, 2019)

Die Aktionen, die vom Smart Contract automatisch ausgeführt werden, können allerdings eine rechtliche Bedeutung haben. Wenn zum Beispiel der Vorgang, der ausgeführt wird, die Pflichten des protokollierten Vertrages erfüllt. Dieses Konzept ist allerdings nicht neu. Es wird bereits bei Warenautomaten angewendet. Das bedeutet, nach Einwurf einer Münze wird die Ware ausgegeben. Der Unterschied zwischen einem Warenautomaten und einem Smart Contract ist der Erfüllungsort. Das bedeutet, dass die Daten, die übertragen werden, an einem anderen Platz auf der Welt liegen. (Hoffmann & Skwarek, 2019)

Aufgrund der Dezentralität und Asynchronität stellt sich hierbei die Frage, wer bei einem Smart Contract die Handlungsgewalt und wer für die ausgeführten Aktionen die Verantwortung hat. Dafür könnten folgende Akteure in Betracht gezogen werden (Hoffmann & Skwarek, 2019):

- Die Person, die den Smart Contract entwickelt hat
- Die Person, die die Transaktion initiiert hat.
- Die Transaktion selbst
- Der Node bzw. die Besitzerin oder der Besitzer des Nodes.

Weiters gibt es Herausforderungen, falls Leistungsstörungen ins Spiel kommen. Da hierbei in der realen Welt Verträge und Vertragsbedingungen geändert werden können. Bei Blockchain-Technologien ist es allerdings nicht mehr möglich, bereits geschriebene Datensätze zu bearbeiten. Somit müsse in solchen Fällen ein neuer Vertrag beziehungsweise eine neue Transaktion geschrieben werden. (Hoffmann & Skwarek, 2019)

2.5.2 Blockchain und die DSGVO

In der Blockchain werden nicht direkt personenbezogene Daten gespeichert. Vielmehr sind es Hashes der Daten. Zur Verknüpfung mit der natürlichen Person gibt es öffentliche Schlüssel, mit denen man anhand der Daten in der Blockchain auf die Person Rückschlüsse machen könnte. Bei permissioned bzw. zulassungsbeschränkten Blockchains ist es sehr einfach, einen Bezug zur Person herzustellen. Denn jede Person, die Zugriff auf diese Blockchain hat, hat einen eindeutigen Schlüssel. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

Bei öffentlichen Blockchains sieht das anders aus. Hier ist es nur mit verschiedenen Mitteln möglich, die Person hinter dem Schlüssel zu identifizieren. Beispielsweise über diverse online Märkte. Hierbei muss der Markt die Person und den Schlüssel miteinander verknüpfen und somit werden in der Blockchain indirekt personenbezogene Daten gespeichert. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

Somit ist es bei Blockchains möglich, die natürliche Person zu identifizieren und dadurch unterliegt der Betrieb einer Blockchain den Datenverarbeitungsgrundlagen der DSGVO. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

Der erste Punkt, der bei Blockchain-Technologien unklar ist, ist die datenschutzrechtliche Verantwortlichkeit. Diese Technologie basiert auf dem Prinzip der dezentralen Speicherung und Bearbeitung. Dies erschwert die Identifikation der verantwortlichen Personen enorm. Als Verantwortliche können mehrere Akteure in Betracht gezogen werden (Europäisches Parlament & Panel for the Future of Science and Technology, 2019):

- Die Entwicklerin oder der Entwickler
- Die Stelle die die Blockchain initiiert hat
- Das Mitglied des Netzwerkes, welches eine Transaktion erstellt
- Jeder der einen Node in der Blockchain betreibt
- Der Miner, die neuen Blöcke generiert.

Es ist auch eine gemeinsame Verantwortlichkeit denkbar. Das bedeutet, dass mehrere Akteure diese Rolle einnehmen können. Die DSGVO verlangt eine Stelle, die über die Verarbeitung der personenbezogenen Daten entscheidet. Dies muss, um rechtlich einwandfrei zu arbeiten, geklärt werden. (Art.26 - Gemeinsam für die Verarbeitung Verantwortliche, 2016)

Was dieses Problem noch schwieriger zu lösen lässt ist, dass die Akteure in der Blockchain oft in Ländern verteilt sind, in denen die DSGVO nicht mehr gilt.

Ein weiterer Punkt und auch das größte Problem ist das Recht auf Löschung beziehungsweise Vergessenwerden. Denn wie bereits bekannt, sind bestehende Transaktionen nicht mehr rückgängig zu machen und können auch nicht gelöscht werden. Somit ist die Frage noch offen, wie dieses Recht geltend gemacht werden kann. Es gibt gewisse Wege, mit denen es möglich ist, dieses Problem zu lösen. Bei privaten Blockchains gibt es sogar gewisse Technologien, mit denen man einzelne Transaktionen entfernen kann. Wie dies allerdings bei öffentlichen Blockchains gemacht werden kann, ist nach wie vor unklar. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

Eine Möglichkeit, dies zu lösen, wäre es, die personenbezogenen Daten nie in der Blockchain gespeichert werden, sondern lediglich eine Verlinkung zu einem Speicherort, wo diese zugreifbar sind. In diesem Fall könnten die verlinkten Daten gelöscht oder verändert werden. Nachdem hier die Daten gelöscht wurden, verbleibt lediglich der Hash und eine leere Verlinkung in der Blockchain. Hierbei spricht man von Off-Chain-Speicherung. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

Falls diese Methode angewendet wird, würde das aber gegen das Konzept von Blockchains sprechen. Denn die Idee ist, dass die Daten nicht verändert werden können. Das Recht auf Löschung ist somit eine der größten Herausforderungen, wenn es um die Rechtslage von Blockchains geht. (Wirth & Kolain, 2018)

Ein anderer Ansatz, um dieses Ziel zu erreichen, könnte sein, dass die Daten, die auf die Blockchain geschrieben werden, verschlüsselt werden. Falls das Recht auf Löschung nun geltend gemacht wird, könnte man die Schlüssel löschen und somit können die Transaktionen von niemanden mehr rückverfolgt werden. Diese Methode wurde auch von der französischen Datensicherheitsbehörde CNIL als mögliche Lösung vorgeschlagen. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019)

2.5.3 Ablegen von Verträgen in der Blockchain

Um Verträge in der Blockchain abzulegen, kann man verschiedene Ansätze wählen. Einerseits ist es denkbar, Verträge direkt mittels Smart Contracts abzubilden. Das bedeutet, die Vertragspartner schließen den Vertrag ab und die Gegebenheiten werden in dem Smart Contract abgespeichert. Weiters können damit auch noch automatisierte Aktionen erstellt werden. Somit könnte man beispielsweise automatisiert überprüfen, ob die Bedingungen des Vertrages eingehalten wurden und je nachdem die Akteure benachrichtigen. Es wäre sogar möglich, direkt Mahnungen und Ausgleichszahlungen zu beantragen. (Fries & Paal, 2019)

Ein anderer Ansatz ist mittels NFT. Da würde man den Vertrag mittels Dokuments erstellen und dieses erstelle Dokument dann beim NFT hinterlegen. Hierbei ist das Problem, dass die meisten NFTs nur von einer Person gehalten werden können. Es gibt aber bereits Möglichkeiten, mit denen mehrere Personen ein NFT besitzen können.

Eine Lösung wäre Fraktionale NFTs. Hierbei wird das NFT in mehrere Teile aufgeteilt und jede Besitzerin oder Besitzer bekommt einen prozentualen Anteil am gesamten Token. Dieser Prozess wurde ursprünglich dafür eingeführt, dass der Besitz an hochpreisigen Kunstwerken geteilt werden kann und die Sammlerstücke somit bezahlbar für mehrere Personen werden. (Mazur, 2021)

Wie viele Teile es von NFTs geben kann, muss dabei von der Erstellerin oder dem Ersteller des NFTs festgelegt werden. Jeder dieser Anteile des NFTs ist dabei allerdings Fungible. Das bedeutet, diese Teile können miteinander getauscht werden und auch jedes Prozent des Anteils hat denselben Wert. Diese Form von Besitz ist vergleichbar mit den Anteilen von Firmen. (Mazur, 2021)

2.6 Identifikation der realen Person

Eine große Herausforderung ist es, die Person in der realen Welt mit dem Datensatz in der Blockchain zu verknüpfen. Sowohl bei NFTs als auch bei Smart Contracts muss diese Zuweisung eindeutig sein, um die Richtigkeit und Gültigkeit von Verträgen zu gewährleisten.

Grundsätzlich hat jede Akteurin und jeder Akteur in der Blockchain seine eigene Identifikation. Auch genannt Private Key beziehungsweise Public Key. Diese Identifikationen sind sozusagen Pseudonyme der Personen und Firmen, die Transaktionen ausführen. Diese Adressen zeigen allerdings nicht, wer die wirklichen Personen hinter dieser ID sind. Nur durch diese Identifikationen ist es somit nicht möglich, die reale Person eindeutig zu identifizieren. (Ra, Seo, Bhuiyan, & Lee, 2020)

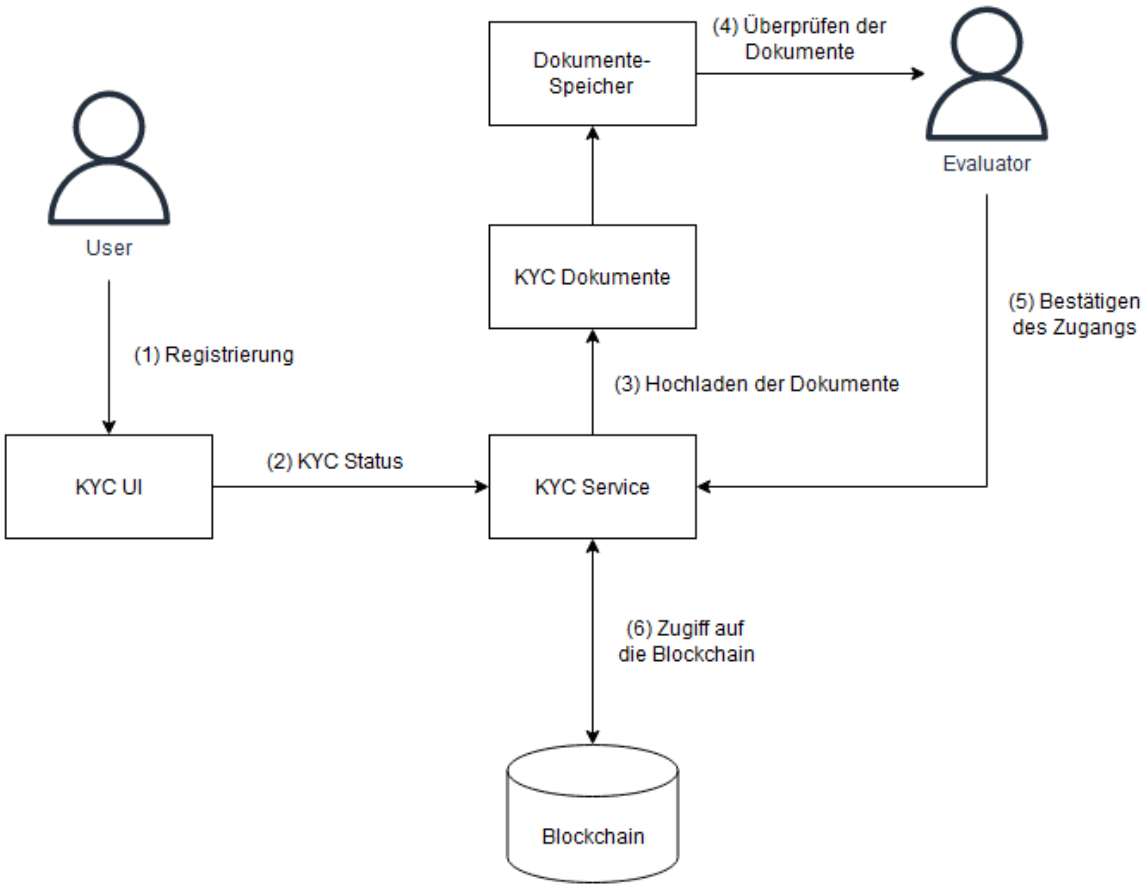
2.6.1 Know your customer (KYC)

Bei diesem Verfahren muss sich die Person vor dem Blockchain-Betreiber ausweisen und verifizieren. Bereits sehr viele Regierungen setzen auf dieses Verfahren und Verlangen, dass es angewendet wird, wenn jemand eine neue Blockchain erstellen will.

Dieses Verfahren kann sowohl bei öffentlichen als auch bei privaten Blockchains angewendet werden. Bei einer public Blockchain wird dafür ein Smart Contract benötigt. Dabei registriert sich die Nutzerin oder Nutzer bei dem KYC-Anbieter. Dabei muss diese Person die benötigten Daten zu ihrer Person auf einer Plattform hochladen und ein Administrator dieser Plattform prüft diese und gewährt gegebenenfalls den Zugang zur Blockchain. Wenn nun eine Transaktion in die Blockchain geschrieben werden soll, dann wird zuerst der KYC-Smart-Contract ausgeführt. Dieser überprüft bei der Plattform, ob die Nutzerin oder der Nutzer Zugang und die geforderten Berechtigungen hat. Falls dies nicht der Fall ist, wird die Transaktion abgebrochen. (Kapsoulis et al., 2020)

Bei privaten Blockchains kann das KYC-Verfahren einfacher durchgesetzt werden, da die Person nur Zugriff auf die Blockchain bekommt, wenn dieser von der Betreiberin oder dem Betreiber freigegeben wurde. Im Schritt der Freigabe für den Zugriff kann dabei die Identität der Nutzerin oder des Nutzers sichergestellt werden und der KYC-Prozess durchgeführt werden. (Kapsoulis et al., 2020)

In der folgenden Abbildung wird der Prozess dargestellt, wie das KYC-Verfahren funktionieren könnte. Im ersten Schritt (1) registriert sich dir Person an der KYC UI. Danach (2) wird geprüft, ob es die Benutzerin oder den Benutzer bereits im System gibt. Falls nicht wird muss diese Person Dokumente hochladen, um ihre Identität zu verifizieren (3). Diese Dokumente werden anschließend von einem Evaluator überprüft (4) und wenn alles richtig ist, wird der Zugang bestätigt (5). Ab diesem Moment hat die Person CRUD Zugriff auf die Blockchain (6). Wenn es sich um eine öffentliche Blockchain handelt, dann fragt der Smart Contract beim Schreiben von Transaktionen noch beim KYC Service ab, ob die Person den gewünschten Zugriff hat. Falls die Person bereits registriert war, werden die Schritte 3 – 5 übersprungen und sie hat direkt Zugriff. (Kapsoulis et al., 2020)



2-5 KYC-Prozess (Kapsoulis et al., 2020)

2.6.2 eIDAS-Verordnung

Die eIDAS-Verodnung oder auch electronic Identification, Authentication and trust Services Verordnung regelt europaweit den Einsatz von Diensten zur elektronischen Identifizierung. Sie regelt folgende elektronische Transaktionen:

- elektronische Identifizierung
- Vertrauensdienste
- Elektronische Signaturen
- Elektronische Siegel
- Validierung und Bewahrung von elektronischen Siegeln und Signaturen
- Zeitstempel
- Zertifikate für Websites
- Dienste für elektronische Einschreiben

Bereits im Jahr 1999 hat die europäische Union den Rechtsrahmen geschaffen, dass Fernsignaturen genauso rechtlich gültig sind wie handschriftliche. Besonders seit 2013 ist die Nutzung der digitalen Signatur sehr stark gestiegen. Der besonders große Vorteil dieser Richtlinie ist, dass sie innerhalb der EU grenzüberschreitend eingesetzt werden kann. (Leitold & Konrad, 2018)

Genau durch diese Richtlinie wurden auch den Zertifizierungsstellen genaue rechtliche Rahmen vorgelegt. Somit wurde die Art und Weise, wie digitale Signaturen zertifiziert werden konnten, standardisiert und vereinheitlicht. Diese Stellen, die diese Signaturen zertifizieren, müssen sich somit auch vor Prüfungsstellen akkreditieren und werden auch laufend über die Einhaltung der Standards überprüft. (Leitold & Konrad, 2018)

Bei der eIDAS wird zwischen drei Arten unterschieden, wie Signaturen durchgeführt werden können.

- Einfache Signatur: Hierbei handelt es sich um Daten in elektronischer Form, die anderen Daten beigefügt wurden und logisch miteinander verbunden sind und von der Person, die die Dokumente unterzeichnet hat, verwendet werden. Beispiele hierfür sind eine Namensnennung in einer E-Mail (Heinze & Ojea, 2018)
- Fortgeschrittene Signatur: Diese Signatur muss bestimmte Anforderungen erfüllen. Dabei muss die Unterzeichnerin oder der Unterzeichner eindeutig der Unterschrift zugeordnet werden können. Dabei muss diese Person auch identifiziert werden können und die Unterschrift muss auch von ihr selbst durchgeführt worden sein. Auch eine nachträgliche Veränderung darf nicht mehr möglich sein. (Heinze & Ojea, 2018)

- Qualifizierte Signatur: Diese Art hat in den EU-Mitgliedsstaaten einen besonderen Rechtsstatus. Sie kann mit einer handschriftlichen Signatur gleichgesetzt werden und muss mit einer Signaturerstellungseinheit erzeugt worden sein. Diese Signaturerstellungseinheit ist eine Software oder Hardware, die die Vertraulichkeit der Daten während des Signaturvorganges sicherstellt. Weiters sorgt sie durch gewisse Algorithmen für einen guten Schutz vor Fälschungen und Fremdzugriff. (Heinze & Ojea, 2018)

2.7 Verträge zwischen Unternehmen und deren Mitarbeiterinnen und Mitarbeiter

In Österreich ist zumindest ein Vertrag zwischen Mitarbeiterinnen und Mitarbeitern Pflicht, um ein gültiges Arbeitsverhältnis zu schaffen. Durch diesen Vertrag werden wesentliche Merkmale des Verhältnisses definiert. Diese Informationen können folgende sein (oesterreich.gv.at-Redaktion, 2022):

- Einordnung, wie die Arbeitnehmerin oder der Arbeitnehmer in die Organisation eingegliedert ist.
- Arbeitszeit
- Arbeitsort
- Arbeitsabfolge
- Bindungen
- Laufende Kontrolle durch die Arbeitgeberin oder den Arbeitgeber

Weiters gibt es noch viele weitere Vertragsarten, zwischen denen Unterschieden werden kann, die das Unternehmen und die oder der Angestellte abschließen können. Alle diese Verträge müssen sorgfältig vom Unternehmen dokumentiert werden, um bei rechtlichen Prüfungen und auch bei Auseinandersetzungen die Richtigkeit des Handelns nachweisen zu können.

Schon beim Arbeitsvertrag gibt es je nach Tätigkeit und Anstellungsverhältnis unterschiedliche Arten:

- Kollektivvertrag: Das ist eine der wichtigsten Vertragsarten. Das ist eine allgemeine, gesetzlich festgelegte und vertragliche Vereinbarung zwischen Arbeitnehmer und Arbeitgeber. Darin werden alle Rahmenbedingungen des Arbeitsverhältnisses festgelegt. Wie zum Beispiel: Mindestgehalt, Gehaltszulagen, Feiertag, …. Dieser Vertrag wird immer für eine gesamte Branche bestimmt. Das dient dazu, faire und gleiche Arbeitsbedingungen für alle Arbeitnehmer zu schaffen. (Gerlach, Gruber-Risak, Martin, 1969-, Höfle, & Schrank, Franz, 1945-, 2019)

- Werkvertrag: Bei dieser Vertragsart wird nicht festgelegt, wie und wann gearbeitet wird, sondern man stellt ein Werk für jemanden her. Dieses Ergebnis muss nur bis zum vereinbarten Zeitpunkt fertiggestellt sein. Sobald das Wert erbracht wurde, endet das Dienstverhältnis auch automatisch. (Gerlach et al., 2019)
- Freier Dienstvertag: Diese Vertragsart ist ähnlich zum Werksvertrag. Das bedeutet man ist zeittechnisch und ortstechnisch vom Unternehmen unabhängig. Der Unterschied besteht darin, dass in den meisten Fällen die Arbeitsmittel vom Unternehmen gestellt werden. Weiters hat man hier auch Anspruch auf eine Vertreterin oder einen Vertreter, dieselbe fachliche Ausbildung hat, wie man selbst. Wenn das Ergebnis nicht den Vorstellungen des Unternehmers entspricht, hat man trotzdem Anspruch auf das vereinbarte Entgelt. (Gerlach et al., 2019)

Weiters gibt es noch zusätzlich Betriebsvereinbarungen, die zwischen Unternehmen und Arbeiterinnen und Arbeitern getroffen werden können. Diese können Beispielsweise folgende sein (Gerlach et al., 2019):

- Änderungen des Arbeitsvertrages: Beispielsweise bei Gehaltserhöhungen oder Änderung der Arbeitszeit
- Zusatzvereinbarungen über eine Dienstfahrzeug
- Geheimhaltungsverträge
- Weiterbildungskosten
- Reisekosten
- Private Internetnutzung am Arbeitsplatz

All diese Vertrage müssen entsprechend lang aufbewahrt werden, um bei etwaigen Prüfungen vorgewiesen werden zu können. Darüber hinaus enthalten diese sensible und persönliche Daten, die unbedingt geschützt werden müssen. Daher müssen sie seitens der Firma sicher abgelegt werden und nur autorisierten Personen darf Zugriff auf diese Dokumente gegeben werden. (WKO, 2019)

Wenn diese Dokumente digitalisiert wurden, dann müssen sie auch dementsprechend abgespeichert werden, sodass sie den Bestimmungen der Datenschutzgrundverordnung entsprechen. Viele Unternehmen setzen bei diesem Thema auf Contract-Management-Tools. Mithilfe dieser Software können Verträge verwaltet werden. (Waldegge, 2018)

Bei diesen Tools oder generell, wenn sensible Dokumente digital abgespeichert werden, ist es wichtig, dass diese Software die Daten sicher speichert. Das bedeutet, dass folgende Kriterien erfüllt sein müssen (Dieluweit, 2017):

- Ausfallsicherheit: Auf die Dokumente sollte immer zugegriffen werden können. Falls es Ausfälle gibt, sollte es Möglichkeiten geben, den Zustand vor dem Ausfall wiederherzustellen.
- Datenschutz: Die Daten, die auf den Servern gespeichert werden, müssen vor unautorisierten Zugriffen geschützt werden.
- Datensicherheit: Falls es an dem Standort, wo die Server stehen, zu Katastrophen wie Feuer oder Überschwemmungen kommt, sollten die Daten trotzdem noch an einem anderen Ort als Backup gespeichert sein.
- Skalierbar: Falls das Unternehmen rapide wächst und die bestehende Infrastruktur nicht mehr ausreicht, sollte es mit einfachen Mitteln möglich sein, diese zu vergrößern.

2.7.1 Vorteile von digital abgewickelten Verträgen

In der heutigen Zeit ist es nach wie vor so, dass in den meisten Unternehmen die Verträge auf Papier unterzeichnet werden. Diese Papiere werden anschließend digitalisiert und auch archiviert. Dabei wäre es bereits möglich, diese Prozesse vollständig digital abzubilden. (Hameurlain, Küng, Wagner, Dang, & Thoai, 2017)

Verträge digital abzuhandeln hätte auch folgende Vorteile:

- Zeitersparnis: Die Vertragspartner können sich direkt die Verträge zuschicken und selbst unterzeichnen. Ein persönliches Treffen ist nicht mehr notwendig. (Hameurlain et al., 2017)
- Kopien: Von Verträgen müssen eventuell mehrere Kopien angefertigt werden. Bei digitalen Verträgen ist das Vervielfachen leichter und Ressourcensparender als bei physischen. (Hameurlain et al., 2017)
- Umweltschutz: Für physische Verträge wird viel Papier verwendet. Dieses Papier muss auch bedruckt werden. Dieser Schritt kann bei elektronischen Verträgen vermieden werden. (Hameurlain et al., 2017)
- Suche nach Verträgen: Falls Dokumente zu einem späteren Zeitpunkt wieder benötigt werden, muss bei Verträgen in Papierform das Archiv durchsucht werden. Bei digitalen Versionen kann mithilfe von Suchalgorithmen schneller der gewünschte Vertrag gefunden werden. (Hameurlain et al., 2017)

2.7.2 Nachteile von digital abgewickelten Verträgen

Trotz der Vorteile, die digitale Verträge bieten, werden sie noch nicht von jedem Unternehmen eingesetzt. Besonders psychologische Barrieren halten die Menschen noch auf, Dokumente auf diese Art und Weise zu unterzeichnen. Im inneren der Menschheit steckt immer etwas Skepsis, wenn Technologie Prozesse verändert, die seit sehr vielen Jahren gleich ablaufen. Bei Unterschriften ist es die physische Präsenz der Person, die ein Dokument handschriftlich unterschreibt. (Chou, 2015)

Eine Unterschrift repräsentiert immer die Identität der Unterzeichnerin oder des Unterzeichners. Die Unterschrift selbst hat dabei nicht sehr viel Wert, denn es ist eher die Leistung, die dazu verknüpft ist. Beispielsweise hat eine Autogrammkarte eines Stars einen höheren Wert als eine Kopie davon und dass nur, weil diese Person diese Karte in physischer Präsenz eigenhändig unterzeichnet hat. Bei digitaler Kommunikation sowie auch bei Signaturen, die digital erfolgen, findet diese physische Präsenz nicht zwingend statt. (Chou, 2015)

In Studien wurde gezeigt, dass genau diese fehlende physische Präsenz dafür sorgt, dass Verträge, die mit digitalen Signaturen unterzeichnet sind, also nicht so vertrauensvoll angesehen werden. (Chou, 2015)

Weiters müssen die Akteure, die ein Dokument digital signieren wollen, auch das technische Wissen haben, um dies korrekt durchführen zu können. Eine weitere Voraussetzung für digitale Signaturen ist oft die Verwendung einer Mobilen-ID oder einer Zwei-Faktor-Authentifizierung. Diese Verifizierungsmethoden müssen vorerst eingerichtet sein, was auch eine Hürde darstellt. (Hameurlain et al., 2017)

3 VORGEHEN UND METHODE

Zur Beantwortung der Forschungsfrage wurde ein Prototyp entwickelt, mit dem Verträge in der Blockchain festgehalten werden können. Aus der Erfahrung aus den vorherigen Kapiteln wurden die Rahmenbedingungen für den Prototypen geschaffen. Zu Beginn musste die richtige Blockchain, die für die Anforderungen geeignet ist, ausgewählt werden. Dabei handelt es sich um reine Grundvoraussetzungen und Funktionalitäten, die von der Technologie unterstützt werden mussten.

Im Anschluss wurde der Entwicklungsprozess aus dem Kapitel 2.4 durchgegangen. Das bedeutet, zu Beginn mussten die Anforderungen definiert werden, die der Prototyp erfüllen muss. Gefolgt von den weiteren Schritten, die notwendig sind, um eine Software zu entwickeln. Ein besonderer Punkt dabei ist die Entwicklung der Software. Diese wurde anhand der definierten Anforderungen entwickelt und anschließend getestet und überprüft. Abschließend konnte der fertige Prototyp verwendet und evaluiert werden.

Dieser Prototyp wird anhand von Metriken geprüft, um herauszufinden, ob diese Technologie in Unternehmen eingesetzt werden kann.

3.1 Evaluation anhand von Metriken

Für die Evaluation wurden Metriken definiert, mit denen die Implementation in der Blockchain und eine konventionelle Art und Weise Verträge zu dokumentieren verglichen wird. Diese Metriken sollen Aufschluss darüber geben, wie gut Blockchain-Technologien insbesondere der Prototyp in der Praxis eingesetzt werden kann. Zu Beginn wird überprüft, wie gut das Ablegen der Verträge in der Blockchain und wie gut die implementierten Funktionen funktionieren. Da das direkte Speichern von Verträgen in der Blockchain sehr ineffizient und teuer ist, wird wie Kapitel 4 beschrieben, nur eine Verlinkung zum Dokument darin gespeichert. Somit wird der Fokus bei der Evaluation das Dokumentieren von Verträgen sein und wie sinnvoll diese Art der Dokumentation ist.

Um die tatsächliche Einsatztauglichkeit des Prototyps widerzuspiegeln, wurden unterschiedliche Metriken herangezogen. Diese behandeln einerseits nur die Blockchain und den entwickelten Smart Contract und andererseits wird der Prototyp mit einer konventionellen Methode verglichen, die in einer Vielzahl an Unternehmen Einsatz findet.

Alle Werte, die erhoben werden, können in eine der drei Kategorien klassifiziert werden (Fenton & Pfleeger, 1998):

- **Prozess:** Bei prozessbezogenen Metriken werden Daten erhoben, die direkt Bezug auf die Softwareentwicklung haben. Beispielsweise möchte man herausfinden, wie lange es dauert einen Prozess fertigzustellen und wie viel es kostet.
- **Produkt:** Hierbei werden Artefakte, Leistungen und Dokumente analysiert, die bei einer Prozessaktivität entstanden sind. Diese Werte müssen nicht immer unbedingt fertiggestellte Software-Produkte analysieren. Es können auch Prototypen und Teile von Software analysiert werden. Beispiele hierfür sind die Nutzerfreundlichkeit, Integrität, Effizienz oder die Testbarkeit.
- **Ressourcen:** Das sind Werte, die erhoben werden, die während der Prozessaktivität benötigt werden. Das bedeutet, hier werden die Inputs analysiert, die für die Softwareproduktion verwendet werden. Beispiele hierfür sind: Personal, Material oder Tools

Im weiteren Verlauf wird die Metriken noch einmal klassifiziert. Hierbei gibt es die beiden Kategorien (Fenton & Pfleeger, 1998):

- **Internes Attribut**: Diese Art von Metrik ist einfach zu messen. Hierbei werden strukturelle Werte des Produktes herangezogen, wie beispielsweise die Anzahl der Zeilen an Code.
- **Externes Attribut**: Externe Attribute hängen sowohl am Verhalten der Software als auch an der Umgebung von dessen ab. Hierbei kann beispielsweise die Zuverlässigkeit von Software analysiert werden. Dazu müsste sowohl die Hardware als auch die Art der Verwendung untersucht werden.

3.1.1 Auswahl der Metriken

Um die gewünschten Metriken auszuwählen, wurde das Goal-Question-Metric paradigm angewendet. Alle Daten zu messen ist oft nicht zielführend, da sie für die Ergebnisse keinen sinnvollen Beitrag leisten. Weiters ist auch die Zeit, die für das Messen benötigt wird, auch ein wesentlicher Faktor, der bei Software-Projekten beachtet werden muss. Mit dieser Methode werden die wichtigsten und aussagekräftigsten Metriken definiert, um einen effizienten Evaluierungsprozess zu gewährleisten. (Fenton & Pfleeger, 1998)

Zu Beginn dieser Methode werden die Ziele definiert, die erreicht werden sollen. Für diese Arbeit sind diese:

1. Die Qualität des Smart Contracts evaluieren
2. Mit einem konventionellen Softwaresystem vergleichen

Aus den definierten Zielen werden anschließend Fragen abgeleitet. Die Fragen selbst beziehen sich immer auf das im vorherigen Schritt definierte Ziel. Für jedes Ziel werden mindestens eine oder mehrere Fragen definiert. Man sollte beachten, dass es zur Beantwortung der Fragen im weiteren Verlauf auch sein kann, dass mehrere Metriken definiert werden, um diese zu beantworten. (Fenton & Pfleeger, 1998)

In der folgenden Aufzählung werden die Fragen für die definierten Ziele behandelt:

Ziel 1: a) Was bedeutet Qualität?

Ziel 2: a) Welches System ist effizienter?

b) Wie kann man die Sicherheit von Daten gewährleisten?

c) Können das System mehrere Personen nutzen?

d) Können Verträge in der Vergangenheit nachvollzogen werden?

e) Welches System ist in der Verwendung günstiger?

f) Kann die Software beliebig angepasst werden?

Aus diesen Fragen können anschließend die Metriken abgeleitet werden, die zur Messung der Software herangezogen werden. In den folgenden zwei Kapiteln werden die Metriken aus den Fragen detailliert beschrieben und klassifiziert.

3.1.2 Metriken des Smart Contracts

In diesem Kapitel wird das erste definierte Ziel und die Frage, was Qualität im Zusammenhang mit einem Smart Contract bedeutet, behandelt. Um grundsätzlich herauszufinden, wie gut der Smart Contract implementiert wurde und ob dieser sowohl performant als auch kostengünstig ausgeführt werden kann, wird dieser zuerst anhand folgender Metriken evaluiert:

1. **Gas Price:**

   Beschreibung: Die erste Grundlage, die herangezogen wird, um den Smart Contract zu überprüfen, ist der Gas Price. Das ist der Preis, der bezahlt werden muss, wenn der Code ausgeführt werden soll. Mit diesem Geld sollen die Kosten gedeckt werden, die das Netzwerk hat, um die Transaktion zu validieren. Ziel dabei ist es den Preis möglichst gering zu halten, um einerseits die Personen zu entlasten, die den Smart Contract ausführen wollen und andererseits weniger Leistung des Netzwerkes zu beanspruchen. (Vacca, Di Sorbo, Visaggio, & Canfora, 2021)

   Kategorie: Prozess

   Art: Externes Attribut

   Skala: Intervall

2. **Codequalität:**

   Beschreibung: Im nächsten Schritt wird die Codequalität des Smart Contracts überprüft. Dafür gibt es verschiedene Tools, mit denen das gemacht werden kann. Das Tool, das verwendet wird, ist Smartcheck. Diese Software wandelt den Code in eine XML-basierte Form und vergleicht diesen mit anderen Smart Contracts von der Platform Etherscan.io. Dabei können Sicherheitslücken, Funktionalitätslücken oder auch andere Probleme, die bei der Entwicklung entstanden sind, entdeckt werden. (Vacca et al., 2021)

   Kategorie: Prozess

   Art: Internes Attribut

   Skala: Ordinal: Schlecht, Mittel, Gut

3.1.3 Metriken zum Vergleich mit einem konventionellen System

Um schlussendlich auch Aussagen darüber treffen zu können, wie gut diese Technologie beziehungsweise der Prototyp in der Praxis in Unternehmen eingesetzt werden könnte, werden noch weitere Metriken herangezogen, mit denen das bewiesen werden können soll. Dazu wurden folgende Metriken verwendet:

1. **Effizienz:** Diese Metrik beschreibt, wie schnell und zuverlässig die Aufgaben ausgeführt werden können. Hierbei werden folgende Werte erhoben:

   - **Wie lange dauert es, bis die gewünschte Aufgabe erledigt wurde? [E1]**

     Beschreibung: Hier wird die Zeit gemessen, die benötigt wird, um den Testfall auszuführen. Desto niedriger die Zeit ist, desto eher deutet es darauf hin, dass die Software effizient genutzt werden kann.

     Kategorie: Prozess

     Art: Externes Attribut

     Skala: Intervall

   - **Wie häufig treten Fehler bei der Ausführung der Aufgaben auf? [E2]**

     Beschreibung: Hier wird gezählt, wie oft es zu Fehlern und zu Komplikationen gekommen ist. Weiters wird dokumentiert, um welchen Fehler es sich handelt. Desto weniger Fehler bei er Verwendung auftreten, desto besser ist es.

     Kategorie: Prozess

     Art: Externes Attribut

     Skala: Intervall

2. **Sicherheit:** Hier wird überprüft, wie sicher die Daten sind, die gespeichert werden. Das bedeutet auch, es wird überprüft, wie einfach es ist, die Daten nachträglich zu manipulieren. Hier werden folgende Werte erhoben:

    - **Wie gut sind die Daten gegen unbefugten Zugriff gesichert? [S1]**

      Beschreibung: Hier wird die Art und Weise gemessen, wie einfach sich die Verträge gegen fremden Zugriff schützen lassen. Das bedeutet, folgende Punkte werden überprüft:

        1. Wie lassen sich die Dokumente vor unbefugten Zugriff schützen?
        2. Wie lässt sich die Dokumentation selbst vor unbefugten Zugriff schützen?

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Ordinal: Schlecht, Mittel, Gut

    - **Wie sind die Daten vor Änderungen geschützt? [S2]**

      Beschreibung: Hier wird überprüft, wie einfach sich die Daten nachträglich verändern lassen. Folgende Daten werden geprüft:

        1. Können Daten nachträglich geändert werden?
        2. Sind Änderungen nachvollziehbar und versioniert?

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Ordinal: Schlecht, Mittel, Gut

3. **Skalierbarkeit:** Wie gut kann die Software mit steigendem Datenaufkommen oder mehreren gleichzeitigen Nutzerinnen und Nutzern umgehen?

    - **Wie viele Personen können gleichzeitig Verträge dokumentieren? [Sk1]**

      Beschreibung: Hier wird versucht, über mehrere Instanzen der Software einen Vertrag gleichzeitig zu Dokumentieren. Dabei wird überprüft, ob es zu Konflikten oder zu längeren Wartezeiten kommt.

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Intervall

4. **Transparenz:** Wie können dokumentierte Verträge nachvollzogen werden?

    - **Wie gut kann nachverfolgt werden, welche Aktionen durchgeführt wurden? [T1]**

      Beschreibung: Hier wird überprüft, wie gut in der Vergangenheit dokumentierte Verträge nachvollzogen werden können. Folgende Daten werden geprüft:

        1. Kann nachvollzogen werden, wann welcher Vertrag dokumentiert wurde?
        2. Kann nachgewiesen werden, wann Änderungen passiert sind?

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Ordinal: Schlecht, Mittel, Gut

5. **Kosten**: Diese Metrik vergleicht, welche Kosten durch die Verwendung der konventionellen Software und dem Smart Contract entstehen.

    - **Welche Kosten fallen beim Dokumentieren eines Vertrages an? [K1]**

      Beschreibung: Hier werden die Kosten überprüft, welche pro Vertrag anfallen.

      Kategorie: Ressourcen

      Art: Internes Attribut

      Skala: Intervall

    - **Welche Kosten fallen pro Jahr an, um die Software nutzen zu können? [K2]**

      Beschreibung: Hier werden die Kosten überprüft, welche jährlich anfallen. Zum Beispiel Lizenzen.

      Kategorie: Ressourcen

      Art: Internes Attribut

      Skala: Intervall

6. **Flexibilität:** Hierbei wird überprüft, wie einfach es ist, die Software an die sich ändernden Anforderungen anzupassen. Weiters wird hier auch überprüft, wie einfach sich bestehende Funktionen anpassen lassen können.

    - **Wie einfach ist es, die Anwendung anzupassen? [F1]**

      Beschreibung: Hier wird überprüft, wie und ob man die Software anpassen kann, falls sich die Anforderungen an diese geändert haben. Hier werden folgende Punkte überprüft:

        1. Kann der Prototyp/das Tool an die Anforderungen angepasst werden?
        2. Kann der Prototyp/das Tool, nachdem bereits Verträge dokumentiert wurden, angepasst werden?

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Ordinal: Schlecht, Mittel, Gut

    - **Welche Möglichkeiten zur Integration zu anderen Systemen gibt es? [F2]**

      Beschreibung: Hier wird überprüft, ob es Möglichkeiten gibt, die Software mit anderen Systemen zu integrieren. Hier werden folgende Punkte überprüft:

        1. Gibt es Schnittstellen zur Integration mit anderen Systemen?
        2. Gibt es andere Möglichkeiten zur Integration mit anderen Systemen?

      Kategorie: Prozess

      Art: Externes Attribut

      Skala: Ordinal: Schlecht, Mittel, Gut

    - **Wie oft können die Funktionen angepasst werden? [F3]**

      Beschreibung: Hier wird überprüft, wie oft man die Software an die geänderten Anforderungen anpassen kann.

      Kategorie: Prozess

      Art: Internes Attribut

      Skala: Intervall

Bei den Metriken, die ordinal skaliert sind, gibt es jeweils zwei Sub-Fragen. Können beide Fragen beantwortet werden, bekommt die Metrik die Bewertung Gut. Wenn nur eine der Fragen mit Ja beantwortet werden kann, wird die Bewertung Mittel vergeben und für den Fall, dass keine der beiden Fragen beantwortet werden oder mit Nein beantwortet wird, wird die Bewertung Schlecht vergeben.

3.1.4 Vor- und Nachteile

Der große Vorteil dieser Methode ist, dass sie sehr zeiteffizient und kostengünstig ist. Das bedeutet, innerhalb kurzer Zeit ist es möglich, Ergebnisse zu sammeln und die Metriken auszuwerten. Weiters werden mithilfe der Metriken Aussagen getroffen werden, wie gut sich diese Technologie im Vergleich zur konventionellen Software einsetzen lässt. Der Wert dieser Aussagen ist meist auch besser im Vergleich zu rein statistischen Auswertungen. (Fenton & Pfleeger, 1998)

Ein großer Nachteil der Evaluierung anhand von Metriken ist, dass die Software wirklich nur anhand der gewählten Metriken beleuchtet wird. Somit könnten essenzielle Teile vergessen werden und die getroffenen Aussagen könnten die Realität nur bedingt widerspiegeln. Hier kann allerdings entgegengewirkt werden, indem in der Planungsphase und bei der Definition der Ziele besonders Acht gegeben wird, dass die gewünschten Ziele untersucht werden. (Solingen & Berghout, 1999)

3.1.5 Ablauf der Evaluation

Für die Evaluation wurden Testfälle ausgearbeitet, die die Verwendung dieser Software in der Praxis widerspiegeln sollen. Diese Fälle werden von der Evaluatorin oder dem Evaluator durchgeführt. Dabei wird der Prototyp und die Software untersucht und anhand der Metriken bewertet. Abschließend werden die Ergebnisse der Evaluation dokumentiert. Um einen Vergleich zwischen den Prototypen und der konventionellen Software ziehen zu können, wurde eine Bewertungsmatrix angefertigt.

In dieser Bewertungsmatrix werden die ausgewerteten Ergebnisse der Evaluation eingetragen. Mithilfe dieser Matrix ist es möglich, die zwei Programme miteinander zu vergleichen und Aussagen darüber treffen zu können, welche besser ist.

Weiters sollen die Ergebnisse aus der Evaluation analysiert werden, um herauszufinden, wie sinnvoll der Einsatz von Blockchain-Technologien in der Praxis sein könnte und wie gut sich der Prototyp bereits einsetzen lassen würde.

3.2 Testszenarien

In diesem Kapitel werden zwei Testszenarien abgebildet, mit denen anschließend der Prototyp und Microsoft Excel evaluiert werden. Diese Testfälle sollen dabei helfen, eine klare Struktur vorzugeben und der Person, die die Evaluation durchführt, Schritte vorzugeben, welche abgearbeitet werden sollen. Dadurch kann gewährleistet werden, dass bei beiden Tools dieselbe Art der Anwendung überprüft wird. Weiters ist es dadurch auch leichter nachzuvollziehen, wie die Ergebnisse zustande gekommen sind, da der Test anhand der definierten Testfälle klar vorgegeben wurde.

3.2.1 Erster Testfall

Als Erstes sollen zehn Verträge zwischen dem Unternehmen und jeweils einer Mitarbeiterin oder einem Mitarbeiter festgehalten werden. Welche Art von Vertrag es ist, spielt dabei keine Rolle. Es handelt sich somit um einen Vertrag zwischen zwei Akteuren. Somit werden auch zwei Adressen eines Wallets in der Blockchain benötigt. Die Adresse des Unternehmens wird in diesem Fall als Haupthalter hinterlegt und die andere Partei wird bei den zusätzlichen Adressen im Smart Contract abgespeichert.

Die Vorbedingungen für diesen Testfall sind, dass der Vertrag bereits an einem Ort digital abgespeichert wurde und der Pfad beziehungsweise der Link zu diesem Dokument bekannt ist. Dieser Link soll anschließend in der Dokumentation hinterlegt werden.

In der folgenden Tabelle werden die Daten der zehn Verträge genau beschrieben. Die Namen und Daten wurden dabei zufällig generiert:

#	Unternehmen	Mitarbeiterin oder Mitarbeiter	Datum
1	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Markus Cole (0xf132eed8126AF848Acad92d548710f0d1eAfaeC2)	14.02.2023
2	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Ralf Sanger (0xa5c8901657182CB5725834414CC0836d311B01e2)	09.02.2023
3	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Marco Bader (0xf66B18b22152A1d058ef06e29EcD5bdeCC1d14f3)	07.02.2023
4	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Sophia Koch (0xEBDf7204706060B1D81e155867F33055591B6503)	15.02.2023

5	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Franziska Lang (0x7F127572607c9dB3AA25F51F80C9A89dE49E94C2)	16.02.2023
6	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Nicole Papst (0x7F27A11911CB62ef9954a0B9BdB44dC15025cF6A)	01.02.2023
7	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Manuela Gärtner (0x1E4946B93A60e60728877A951E662451D930CD3a)	04.02.2023
8	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Katja Bader (0xE275CDFA5a906b19495131DEdfa253c33b633280)	07.02.2023
9	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Christian Schuh (0xbb71264397C2f493E7E3a3AEef22b43e618Dc360)	09.02.2023
10	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Kristin Beike (0xcFDBa40dd5A5ABa695A17c6988718165FeEF0a13)	03.02.2023

Nachdem die Verträge dokumentiert wurden, sollen einige Daten abgefragt werden. Zu Beginn werden Basisdaten wie die Adressen der Beteiligten, Anzahl der Beteiligten und der Pfad zu einzelnen Verträgen selbst abgefragt. In der folgenden Tabelle wird aufgeschlüsselt, welche Daten von welchem Vertrag abgefragt werden sollen:

#	Vertrag Nr	Daten	Erwartetes Ergebnis
1	2	Anzahl Beteiligten	2
2	5	Pfad der Dokumente	Pfad des Dokuments
3	8	Beteiligte Personen	Adressen/Namen der Personen
4	3	Anzahl der Beteiligten; Pfad der Dokumente	2, Pfad des Dokuments
5	10	Beteiligten Personen; Pfad der Dokumente	Adressen/Namen der Personen; Pfad des Dokuments

Nachdem die Basisdaten einiger Verträge abgefragt wurden, sollen auch noch weitere Funktionen getestet werden. Dabei soll jeweils herausgefunden werden, bei welchen Verträgen die folgenden Akteure beteiligt sind:

#	Akteur	Erwartetes Ergebnis
6	Manuela Gärtner	Nr/Identifikation des einen Vertrages
7	Unternehmen A	Nr/Identifikation der 10 Verträge

3.2.2 Zweiter Testfall

Bei diesem Testfall sollen fünf Verträge zwischen dem Unternehmen und mehreren Mitarbeiterinnen und Mitarbeitern geschlossen werden. Das bedeutet, es handelt sich um Verträge mit mehr als zwei Akteuren. Somit werden auch die Adressen der Wallets der Beteiligten benötigt. Wie im ersten Testfall wird die Adresse des Unternehmens als Haupthalter in der Blockchain hinterlegt. Alle anderen Parteien werden bei den zusätzlichen Adressen abgespeichert.

Um weitere Funktionalitäten zu testen, besteht dieser Vertrag aus mehr als einem Dokument. Das bedeutet, es wird eine Liste an Links zu den Verträgen gespeichert.

In der folgenden Tabelle werden die Daten der zehn Verträge genau beschrieben. Die Namen und Daten wurden dabei zufällig generiert:

#	Unternehmen	Mitarbeiterin oder Mitarbeiter	Datum
1	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Markus Cole (0xf132eed8126AF848Acad92d548710f0d1eAfaeC2) Ralf Sanger (0xa5c8901657182CB5725834414CC0836d311B01e2)	14.02.2023
2	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Marco Bader (0xf66B18b22152A1d058ef06e29EcD5bdeCC1d14f3) Sophia Koch (0xEBDf7204706060B1D81e155867F33055591B6503) Franziska Lang (0x7F127572607c9dB3AA25F51F80C9A89dE49E94C2)	09.02.2023

3	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Nicole Papst (0x7F27A11911CB62ef9954a0B9BdB44dC15025cF6A) Manuela Gärtner (0x1E4946B93A60e60728877A951E662451D930CD3a)	07.02.2023
4	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Katja Bader (0xE275CDFA5a906b19495131DEdfa253c33b633280) Christian Schuh (0xbb71264397C2f493E7E3a3AEef22b43e618Dc360)	15.02.2023
5	Unternehmen A (0x7318541BfA160fA11f0D1cf2C4958757e5947372)	Kristin Beike (0xcFDBa40dd5A5ABa695A17c6988718165FeEF0a13) Franziska Lang (0x7F127572607c9dB3AA25F51F80C9A89dE49E94C2) Markus Cole (0xf132eed8126AF848Acad92d548710f0d1eAfaeC2)	16.02.2023

Nachdem die Verträge dokumentiert wurden, sollen einige Daten abgefragt werden. Zu Beginn werden Basisdaten wie die Adressen der Beteiligten, Anzahl der Beteiligten und der Pfad zu einzelnen Verträgen selbst abgefragt. In der folgenden Tabelle wird aufgeschlüsselt, welche Daten von welchem Vertrag abgefragt werden sollen:

#	Vertrag Nr	Daten	Erwartetes Ergebnis
1	2	Anzahl Beteiligten	4
2	3	Pfad der Dokumente	Liste der Pfade der Dokumente
3	5	Beteiligten Personen; Pfad der Dokumente	Adressen/Namen der Personen; Liste der Pfade der Dokumente

Nachdem die Basisdaten einiger Verträge abgefragt wurden, sollen auch noch weitere Funktionen getestet werden. Dabei soll jeweils herausgefunden werden, bei welchen Verträgen die folgenden Akteure beteiligt sind:

#	Akteur	Erwartetes Ergebnis
4	Markus Cole	Nr/Identifikationen der Verträge
5	Unternehmen A	Nr/Identifikation der 5 Verträge

3.3 Konventionelle Software

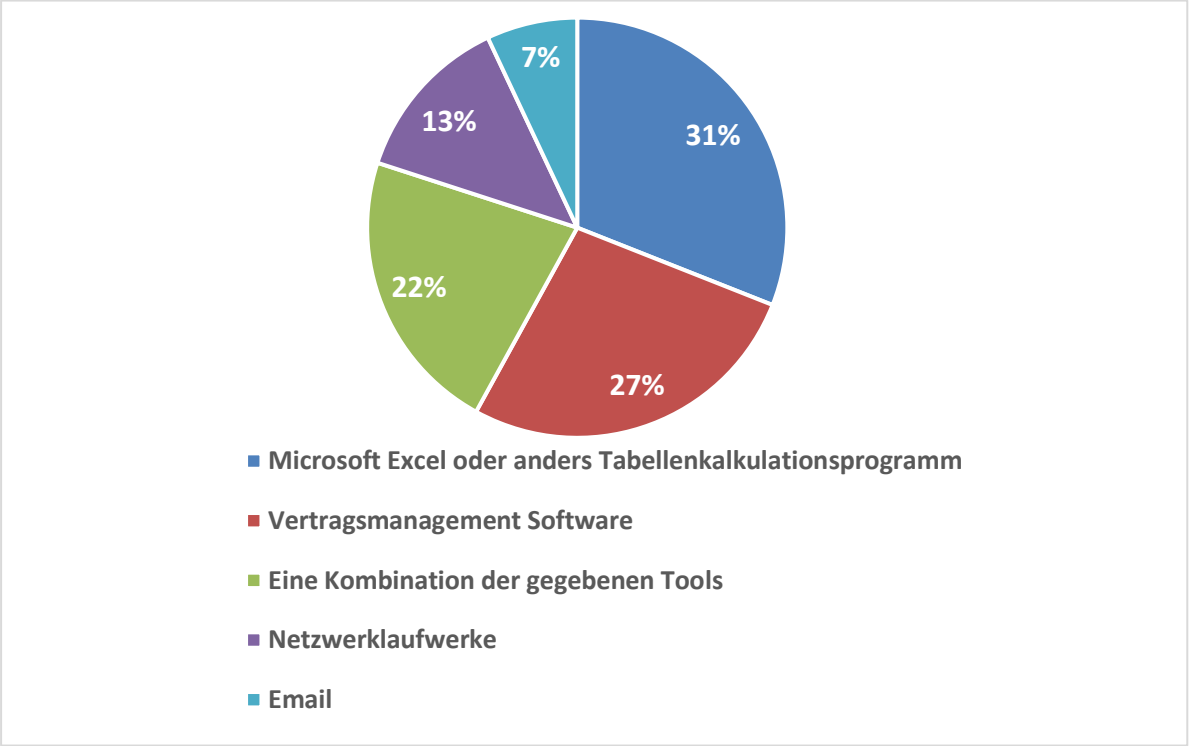
Als konventionelle Software, mit der der Prototyp verglichen wird, wurde Microsoft Excel gewählt. Da diese Software in vielen Unternehmen Einsatz findet und auch das Wissen und Fähigkeiten, diese zu verwenden, bei sehr vielen Personen vorhanden ist. (Formby, Medlin, & Ellington, 2017)

Microsoft Excel ist ein Tabellenkalkulationsprogramm. Es ermöglicht Berechnungen mit Formeln und Funktionen. Weiters können die Benutzerinnen und Benutzer dabei Daten organisieren, analysieren und auch grafisch darstellen. Der Vorteil dieser Software ist, dass sie von sehr vielen Personen verwendet wurde. Das bedeutet, dass Personen, die neu im Unternehmen sind, dieses Tool oftmals bereits kennen und somit der Aufwand für die Einschulung geringer ausfällt. Weiters ist es durch die starke Flexibilität für eine Vielzahl an Anwendungsbereichen geeignet, da die Tabellen und Oberflächen auf den spezifischen Anwendungsfall angepasst werden können. (Vonhoegen, 2019)

Diese Software war im zweiten Quartal 2022 eine der am meisten heruntergeladenen Programmen der Firma Microsoft weltweit. Was darauf schließen lässt, dass sie auch von einer Vielzahl an Personen sowie auch in vielen Unternehmen eingesetzt wird. (AppMagic, 2022)

Des Weiteren gibt es für diverse Anwendungsfälle im Internet Vorlagen zu finden, die häufig zum Einsatz kommen. Ebenso gibt es diese für den Anwendungsfall, Verträge zu dokumentieren und zu verwalten. Hierbei gibt es eine große Menge an Vorlagen, auf die zurückgegriffen werden, kann. (Quintela-del-Río & Francisco-Fernández, 2017)

Auch für den gegebenen Anwendungsfall wird diese Software in verschiedenen Unternehmen verwendet. Eine Umfrage aus 2019 zeigt, dass 31 Prozent der Unternehmen Microsoft Excel oder ein anderes Tabellenkalkulationsprogramm als Vertragsmanagementsoftware verwenden. Mit diesem Ergebnis ist diese Software das am meisten verwendete Tool für den gegebenen Anwendungsfall. (DocuSign, 2019)



3-1 Softwarenutzung für Vertragsmanagement (DocuSign, 2019)

Der Nachteil dieser Software ist allerdings, dass sie nicht automatisch mit anderen Personen in der Firma synchronisiert ist. Somit muss sich die Nutzerin oder der Nutzer um die Verteilung innerhalb der Firma selbst kümmern. Das hat zur Folge, dass gegebenenfalls Änderungen in dem Dokument nicht von allen Stakeholdern wahrgenommen werden können. Abhilfe dafür kann der Cloud-Service von Microsoft schaffen. Dadurch wird das Dokument an einem zentralen Ort gespeichert. Weiters ist es dadurch möglich, dass auch mehrere Personen gleichzeitig an einem Dokument arbeiten. (EnviroInfo, 2014)

3.3.1 Beschreibung der Microsoft Excel Vorlage

Um einen Vertrag ordnungsgemäß zu dokumentieren, müssen gewisse Daten erhoben werden. Diese werden in der Vorlage in Form von Spalten dargestellt. Für den in dieser Arbeit beschriebenen Anwendungsfall wurde die Vorlage von der Plattform https://excel-template.net verwendet und angepasst. In der Version, die zum Vergleich zum Smart Contract genutzt wird, wurden folgende Spalten definiert:

- **Vertragstyp**: Die Art des Vertrages, die der Vertrag annehmen kann.
- **Teilnehmer**: Hier werden alle Personen aufgelistet, die am Vertrag beteiligt sind. Die Personen werden anhand von Beistrichen getrennt.
- **Datum**: Das Datum gesetzt, an dem der Vertrag zustande gekommen ist.
- **Dokumentenpfad**: Hier werden die Pfade hinterlegt, welche zu den digitalisierten Dokumenten führen sollen. Falls es beim Vertrag mehrere Dokumente gibt, werden die Pfade mit einem Beistrich getrennt.
- **Bemerkungen**: In dieser Spalte werden weitere Details des Vertrages festgehalten, die nicht in den anderen Spalten bereits hinterlegt wurden.

In die Funktionalität des Suchens zu ermöglichen wurde das Filtern der Spalten aktiviert. Diese Funktion wurde für alle Spalten eingeschaltet. Dadurch wird es ermöglicht, dass die Nutzerin oder der Nutzer die gewünschten Daten einfacher finden kann. In der Kopfzeile erscheint nun ein Pfeil, der beim Klick ein Dropdown öffnet. In diesem Dropdown können Werte, die in der Spalte vorkommen, deselektiert werden, wodurch die Anzahl der angezeigten Zeilen verringert werden kann.

4 BESCHEIBUNG DES PROTOTYPEN

Ziel dieser Arbeit ist das Erstellen eines Prototyps, mit welchem Verträge zwischen Mitarbeiterinnen oder Mitarbeitern und deren Dienstgeber festgehalten werden können. Dies soll mithilfe von Blockchain-Technologien ermöglich werden.

4.1 Auswahl der Blockchain

Die unterschiedlichen Blockchains, die es gibt, sind für unterschiedliche Einsatzzwecke geeignet. Nicht von jeder Blockchain werden alle Funktionen unterstützt und somit ist es essenziell, die Richtige im Vorhinein auszuwählen, um die Anforderungen zu erfüllen. Folgende Funktionen müssen unterstützt werden, um der Intention gerecht zu werden:

- Ausführen von Smart Contracts
- Erstellen von eigenen Smart Contracts
- Erstellen von NFTs

Weiters stellt sich auch die Frage, ob es sich um eine Public oder Private Blockchain handelt. Der große Vorteil von öffentlichen Netzwerken ist, dass die gesamte Infrastruktur bereits existiert. Das bedeutet, dass keine verteilten Nodes mehr installiert werden müssen. Weiters ist es besonders am Anfang günstiger, die bestehende Infrastruktur zu nutzen. (Himmer, 2019)

Auf der anderen Seite haben private Blockchains oft eine bessere Performance, da sie nicht so viele Personen nutzen. Zusätzlich kann hier das Problem mit der Identifikation der Akteure leichter gelöst werden. Abschließend ist es auch möglich, die Regularien der DSGVO einzuhalten. Nachteile sind, dass diese Blockchain und das Netzwerk erst erstellt werden muss und dafür hohe Ressourcen notwendig sind. (Europäisches Parlament & Panel for the Future of Science and Technology, 2019; R. Yang et al., 2020)

Es gibt auch Plattformen, die hybrid sind. Das heißt sowohl öffentlich als auch privat. Hierbei werden die Vorteile beider Blockchain-Arten vereint. Hierbei gibt es drei dominante Blockchains:

- Ethereum – Quorum and Hyperledger Besu
- Hyperledger Fabric
- Corda

Um Smart Contracts zu testen, gibt es auch Simulatoren und Test-Netzwerke, worin der entwickelte Code ausprobiert werden kann. Das bedeutet, es können kostengünstig und in einer sicheren Umgebung die Funktionen des Smart Contracts überprüft werden. In dieser Test-Umgebung wird nicht in die tatsächliche Blockchain geschrieben. Das bedeutet, dass dafür auch weniger Ressourcen benötigt werden. Auch ist es möglich, bei Fehlern oder Problemen vergangene Transaktionen rückgängig zu machen. Dies erlaubt ein ausgiebiges Testen, bevor der Smart Contract in die Blockchain übertragen wird, wo er dann nicht mehr gelöscht werden kann. (Wohrer & Zdun, 2021)

4.2 Anforderungen an das System

Da mit der Software, Verträge zwischen den Mitarbeiterinnen und Mitarbeitern festgehalten werden können sollen, muss das gesamte System auch gewissen Anforderungen erfüllen. Diese Anforderungen können direkt von den Kriterien eines CM-Tools abgeleitet werden. Das bedeutet, es sollte ein besonderes Augenmerk auf folgende Kriterien gelegt werden:

- Ausfallsicherheit
- Datenschutz
- Datensicherheit
- Skalierbarkeit

Weiters muss auch sichergestellt werden, dass die gewählte Technologie langlebig genug ist. Es hat sich gezeigt, dass gewisse Features der Blockchain nur ein Hype waren, der später wieder abgeklungen ist und man wieder zurück zu konventionellen Technologien gegriffen hat. (Meinel, Gayvoronskaya, & Schnjakin, 2018)

Da Verträge wie in Kapitel 2.7 beschrieben, entsprechend lange dokumentiert und aufbewahrt werden müssen, muss diese Technologie und die Unterstützung der verwendeten Features für diese Zeitspanne gewährleistet werden. Würde das nicht passieren, müsste im Falle des Abschaltens der Technologie eine Migration der Verträge auf ein anderes System passieren. Dies würde für das Unternehmen einerseits bedeuten, dass zu diesem Zeitpunkt Ressourcen aufgewendet werden müssen. Andererseits würde dies bereits zu Beginn eine Barriere bedeuten, dieses System einzuführen.

4.3 Anforderungen an den Prototypen

Der Prototyp soll eine Möglichkeit schaffen, Verträge in der Blockchain abzulegen. Die Anwenderin oder der Anwender füllt dazu die benötigten Daten in ein Formular ein und speichert diese anschließend. Im Hintergrund werden die eingegebenen Daten in der Blockchain entweder als Smart Contract oder als NFT abgelegt. Dabei soll sichergestellt werden, dass die Akteure, die den Vertrag geschlossen haben, auch eindeutig identifiziert werden können.

Dies kann beispielsweise über die ID der Wallets geschehen. Da jede Person, die mit der Blockchain agieren möchte, auch ein Wallet besitzen muss, ist es somit möglich, auch die Person eindeutig zu identifizieren. Im Kontext auf das Dokumentieren von Verträgen zwischen Unternehmen und Mitarbeiterinnen und Mitarbeitern könnte das Unternehmen in diesem Schritt die Identifikationen der Wallets ihrer Mitarbeiterinnen und Mitarbeiter erheben. Somit wären die Personen in der Blockchain anonym und das Unternehmen selbst kann die Akteure der Verträge intern den betroffenen Personen zuordnen.

Weiters soll es die Möglichkeit geben, Verträge auch geheim halten zu können. Da es sich um Dokumente zwischen dem Unternehmen und deren Mitarbeiterinnen und Mitarbeitern handelt, sollten diese nicht von der Öffentlichkeit einsehbar sein. In diesem Zuge sollte auch die Option bestehen, dass Verträge auch gelöscht werden, falls Mitarbeiterinnen oder Mitarbeiter das Unternehmen verlassen und infolgedessen das Recht auf Löschung aus der DSGVO beanspruchen.

Wenn es bereits abgelegte Verträge gibt, sollte es auch die Möglichkeit geben, diese abzufragen. Das bedeutet, dass die ID der Tokens an denen eine Person beteiligt ist abgefragt werden kann und im weiteren Sinne auch die Details zu diesen Verträgen abgerufen werden können Das bedeutet folgende Details der Verträge sollten mindestens abgerufen werden können:

- Identifikationen der beteiligten Personen
- Anzahl der beteiligten Personen
- Pfad der Dokumente
- Weitere Details

Es sollte, um die Nutzung zu vereinfachen, auch möglich sein, dass man die Identifikation einer Person angibt und alle Verträge zurückbekommt, an denen sie beteiligt ist. Somit könnte leichter nach bereits erstellen Verträgen gesucht werden.

Falls Fehler auftreten oder es andere Komplikationen gibt, sollte es auch die Möglichkeit geben, den Betrieb der Blockchain-Applikation zu stoppen. Für den Fall, dass sich diese Komplikationen gelöst haben und es sich nicht um einen Fehler im Code handelt, sollte man den Betrieb auch wiederaufnehmen können.

4.4 Design

Um die Anforderungen an das System abzudecken, ist bereits die richtige Auswahl der Blockchain sehr wichtig. Falls man sich für eine private Blockchain entscheidet, muss man für die Ausfallsicherheit, Skalierbarkeit und Datensicherheit geeignete Strategien entwickeln. Wohingegen diese drei Punkte bei einer öffentlichen Blockchain bereits gewährleistet sind. Andererseits ist es bei einer privaten Blockchain einfacher, den Datenschutz zu gewährleisten, da die Zugriffsrechte leichter beschränkt und Daten leichter gelöscht werden können. Diese Anforderungen können allerdings auch durch die richtige Implementierung eines Smart Contracts in einer öffentlichen Blockchain sichergestellt werden. (R. Yang et al., 2020)

Für diese Arbeit wurde die Ethereum-Blockchain ausgewählt, da sie besonders für Smart Contracts und NFTs am meisten verbreitet ist. Darüber hinaus gibt es für diese Blockchain bereits mehrere Simulatoren und Test-Netzwerke, was das Entwickeln des Prototyps vereinfacht und günstiger macht. (Benahmed et al., 2019)

Zusätzlich gibt es, um die Entwicklung des Smart Contracts zu vereinfachen, auch die Plattform OpenZeppelin. Dadurch kann auf Vorlagen zurückgegriffen werden und auch die Sicherheit erhöht, beziehungsweise eventuelle Fehler bereits im Voraus vermieden werden.

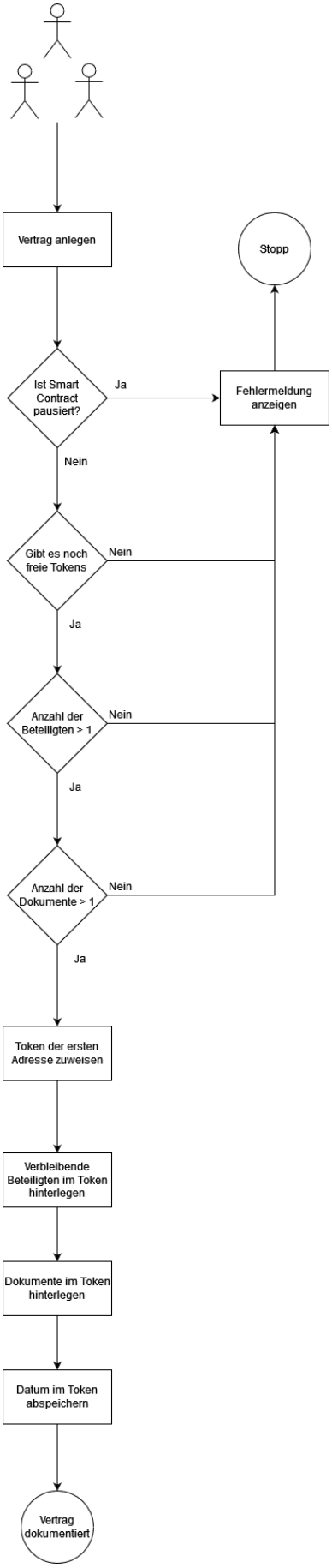
Damit das Ablegen der Verträge die geforderten Anforderungen erfüllt und da das Speichern von Dokumenten direkt in der Blockchain sehr ressourcenaufwendig ist, werden die Dokumente als Pfad in der Blockchain abgelegt. Somit können auch die Anforderungen der DSGVO eingehalten werden, da die Dokumente auch gelöscht werden können. Dabei wird die Integrität des Dokumentes nicht gewährleistet. Dies könnte allerdings durch einen Hashcode des Dokumentes und der Ablage dessen in der Blockchain ermöglicht werden.

Da es auch die Möglichkeit gibt, dass ein Vertrag aus mehreren Dokumenten besteht, sollte auch diese Funktionalität unterstützt werden. Das bedeutet, innerhalb eines Tokens kann eine Liste an Dokumenten hinterlegt werden. Dadurch müssen, falls der Vertrag aus mehreren Dokumenten besteht, diese nicht in ein Dokument zusammengefasst werden, sondern können, wie sie entstanden sind, in der Blockchain dokumentiert werden.

In der folgenden Aufzählung werden die einzelnen Funktionen beschrieben, die von der Software abgedeckt werden sollten:

1. **Das Erstellen beziehungsweise Dokumentieren eines Vertrages**

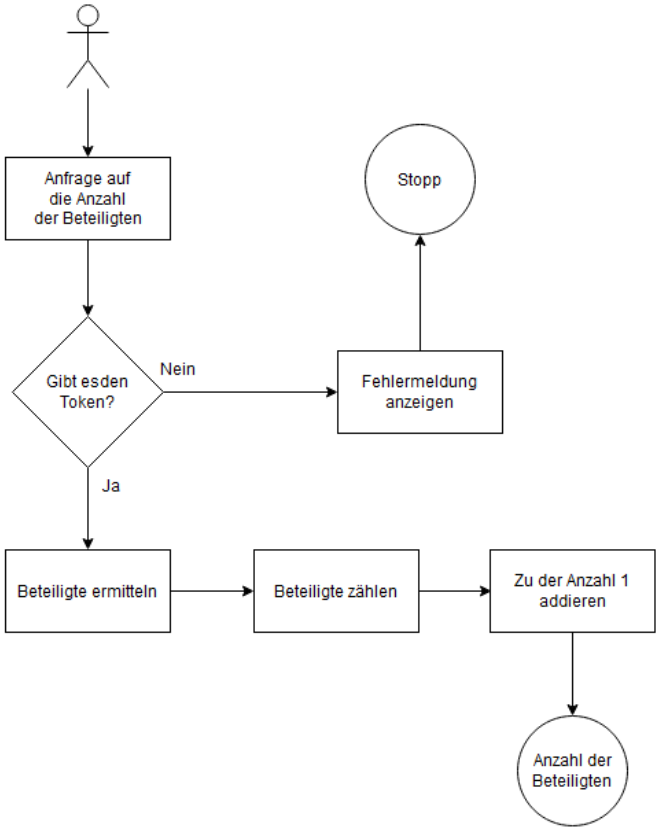
Hier muss im ersten Schritt das Dokumentieren des Vertrages initiiert werden. Nach diesem Schritt werden einige Validierungen durchgeführt. Das bedeutet, es wird beispielsweise überprüft, ob der Smart Contract nicht pausiert wurde, es noch ausreichend Kontingent an Tokens gibt oder die Anzahl der Beteiligten und Anzahl der Dokumente stimmt. Falls eine dieser Regeln gebrochen wird, wird die Ausführung des Smart Contracts abgebrochen und eine Fehlermeldung zurückgegeben. Falls die Validierungen erfolgreich waren, wird im Abschluss der Token der Haupthalterin oder dem Haupthalter zugewiesen und die restliche Information in der Blockchain abgespeichert und die ID des Tokens zurückgegeben.



4-1 Vertrag erstellen

2. Anzahl der Beteiligten abfragen

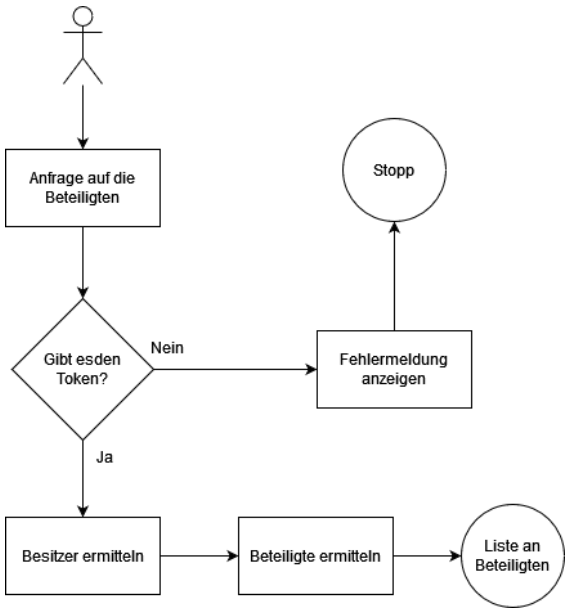
Um die Anzahl der beteiligten Akteure abzufragen, muss zu Beginn die Identifikation des Tokens übergeben werden. Anschließend wird überprüft, ob der Token existiert. Falls er nicht existiert, wird eine Fehlermeldung zurückgegeben. Wenn der Token existiert, werden die beteiligten Personen ermittelt und gezählt, wie viele es sind. Weiters wird zu dieser Anzahl der Wert eins addiert, um die Besitzerin oder den Besitzer miteinzuschließen. Diese Zahl wird abschließend zurückgegeben.



4-2 Anzahl der Beteiligten abfragen

3. Abfragen der Beteiligten

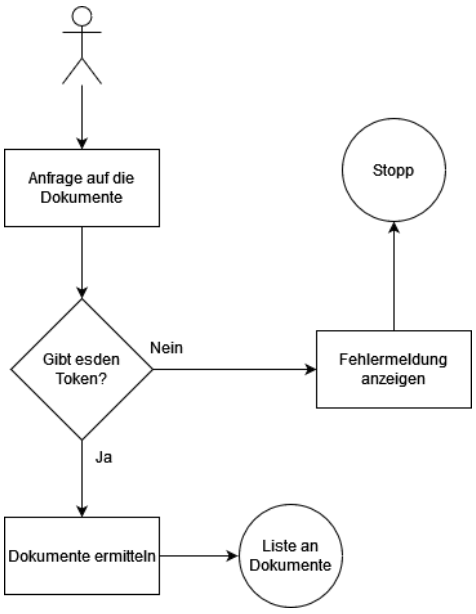
Hierbei wird die Identifikation des Tokens übergeben. Anschließend wird in der Blockchain geprüft, ob es diesen Token gibt. Falls nicht, wird eine Fehlermeldung zurückgegeben. Wenn es den Token gibt, wird zuerst die Besitzerin oder der Besitzer des Tokens ermittelt. Anschließend werden die restlichen Beteiligten ermittelt und eine Liste alle Personen zurückgegeben.



4-3 Abfragen der Beteiligten

4. Abfragen der Dokumente

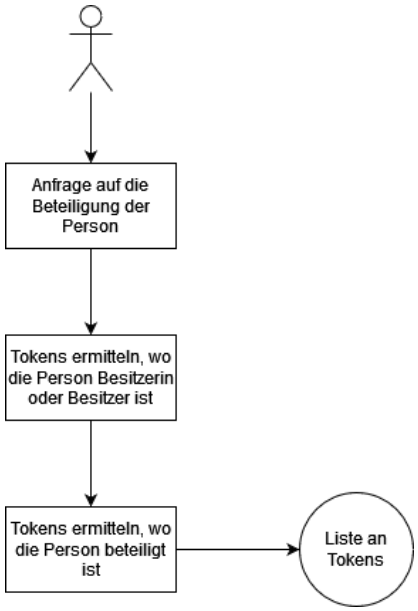
Hier muss ebenfalls die Identifikation des Tokens angegeben werden. Anschließend wird, wie bei der vorherigen Funktion überprüft, ob der Token existiert und wenn nicht eine Fehlermeldung zurückgegeben. Im anderen Fall werden die Dokumente für die ID des Tokens abgerufen und zurückgegeben.



4-4 Abfragen der Dokumente

5. Herausfinden der Verträge, an denen eine Person beteiligt ist

Bei dieser Funktion wird die Adresse einer Person übergeben. Es wird anschließend nach den Tokens gesucht, wo diese Person die Haupthalterin oder Haupthalter ist. Im nächsten Schritt werden alle Tokens gesucht, wo diese Person beteiligt ist. Im Abschluss wird eine Liste an Token-Identifikationen zurückgegeben. Falls die Person bei keinem Vertrag beteiligt ist, wird eine leere Liste retourniert.



4-5 Beteiligung der Person

4.4.1 Zielgruppe

In erster Linie sind Personen, die Erfahrung mit der Verwendung von Computern und ein technisches Grundverständnis besitzen, vorerst die Hauptzielgruppe dieser Software. Um den Vertrag zu Dokumentieren und den Datensatz in der Blockchain abzulegen ist technisches Wissen notwendig.

Der Besitz eines Wallets in der Ethereum-Blockchain ist Grundvoraussetzung, um diesen Prototyp verwenden zu können. Dieses Wallet kann allerdings kostenlos erstellt werden. Das Unternehmen, dass die Verträge über diesen Weg dokumentieren möchte, muss dann die Adressen der Mitarbeiterinnen und Mitarbeiter verwalten, damit die Personen eindeutig den Verträgen zugeordnet werden.

Das bedeutet auch im Unternehmen ist ein Grundverständnis von Blockchains und eine Möglichkeit, die Wallets zu verwalten und mit den Mitarbeiterinnen und Mitarbeitern zu verlinken, gefordert.

Da es möglich sein soll, dass Verträge geheim gehalten werden sollen und diese auch im späteren Verlauf gegebenenfalls gelöscht werden können, sollte ein Speicherplatz bestehen, auf denen die Dokumente abgelegt werden können. Auf diesen Speicherort sollte auch der Zugriff beschränkbar sein, damit nur die Personen Zugriff auf deren Dokumente bekommen, für die sie auch die Berechtigung haben.

4.5 Entwicklung

Im ersten Schritt wurde die Blockchain eingerichtet. Das bedeutet, man muss ein sogenanntes Wallet erstellen. Auf diesem Wallet wird der private Key gespeichert um die Einheiten der Kryptowährung abzufragen. Diese Wallet wurde mithilfe der Software MetaMask erstellt. Hierbei ist lediglich eine Registrierung notwendig. Weiters musste dieses Tool auf das richtige Netzwerk konfiguriert werden, da für diese Arbeit hauptsächlich ein Test-Netzwerk der Ethereum-Blockchain verwendet wird. In diesem Fall wurde für das Goerli-Netzwerk entschieden. Dieses Netzwerk ermöglicht es, alle Funktionalitäten zu testen, ohne dass dabei der Smart Contract in die produktive Blockchain geschrieben wird. Weiters werden durch die Verwendung des Test-Netzwerkes auch Kosten gespart, da die Verwendung von diesem völlig kostenlos ist.

Im nächsten Schritt wurde ein API-Zugang angelegt, um mit dem Netzwerk zu kommunizieren. Mithilfe dieses Zugangs kann der Smart Contract auf die Blockchain geladen werden. Sobald der Code in der Blockchain ist, können mithilfe dieses Zugangs auch die entwickelten Funktionen aufgerufen werden.

Um den API-Zugang anzulegen, wurde die Plattform Alchemy verwendet. Diese bietet Tools an, um mit Blockchains zu kommunizieren. Eine davon ist eine API-Schnittstelle für die Ethereum-Blockchain und die Test-Netzwerke. Diese Plattform wurde ausgewählt, da sie in der Web3-Industrie der Marktführer ist. Über 100 Milliarden Transaktionen wurden bereits mithilfe der Tools, die diese Plattform anbietet, erstellt. Ein weiterer Vorteil ist, dass die benötigten Werkzeuge und API-Schnittstellen, die für diese Arbeit benötigt wurden, kostenlos genutzt werden konnten. (Alchemy Insights, 2023)

In dieser Plattform musste der API-Zugriff zuerst konfiguriert werden. Dazu wurde eine Applikation angelegt und für diese das gewünschte Netzwerk ausgewählt werden. Nach erfolgreicher Erstellung der App kann man die Zugangsschlüssel exportieren, die in einem späteren Schritt benötigt werden. Auch bekommt man auf dieser Plattform eine Übersicht und Statistiken, wie viele Anfragen auf die konfigurierte Schnittstelle gemacht wurden. Weiters ist es dabei möglich. die letzten Anfragen genauer zu inspizieren, um gegebenenfalls auch Fehler leichter finden zu können.

Anschließend wird der Code entwickelt. Dafür wird die JavaScript Runtime Node.js verwendet. Node.js wird verwendet, um skalierbare Netzwerk-Applikationen mit der Programmiersprache JavaScript zu entwickeln. Mit Node.js wird auch der Package-Manager npm installiert. Damit ist es möglich, Open-Source-JavaScript-Bibliotheken für Projekte zu installieren. Mithilfe dieses Package-Manager wurde die Library Hardhat installiert. Diese wird benötigt, um den entwickelten Code in Solidity zu kompilieren, testen und debuggen.

Nachdem nun alle Libraries installiert und die Entwicklungsumgebung eingerichtet wurden, wurde der Code und die dazugehörigen Funktionen entwickelt. Folgende Funktionen werden von dem Smart Contract implementiert:

- **Mint**: Mithilfe dieser Funktion wird der Token der Besitzerin oder dem Besitzer zugeordnet. Bei dieser Methode wird eine Liste an Adressen übergeben und der Pfad, an dem das Dokument abgespeichert ist. Die Adressen sind die der beteiligten Personen. Das bedeutet, dass jede Person, die an dem Vertrag beteiligt ist, auch ein Wallet benötigt. Dabei wird die erste Adresse in der Liste die Hauptbesitzerin oder der Hauptbesitzer des Tokens. Die anderen Adressen werden anschließend auch im Smart Contract hinterlegt, um nachzuvollziehen, wer an diesem Vertrag beteiligt war. Der Pfad, an dem das Dokument gespeichert ist, muss auch nicht öffentlich zugreifbar sein, da es Verträge gibt, die nicht öffentlich einsehbar sein sollten. Da der Pfad für jeden Vertrag einzeln festgelegt werden kann, ist es auch möglich, diesen so zu wählen, dass für gewisse Verträge eine Authentifizierung gefordert wird. Weiters ist es durch diese Variante auch möglich, Verträge nachträglich zu löschen.
- **GetParticipantsOfToken**: Mithilfe dieser Methode können alle Beteiligten des Vertrages abgerufen werden. Dabei wird die ID des Tokens übergeben. Als Rückgabe liefert diese Funktion eine Liste der Akteure. Auch die Haupthalterin oder der Haupthalter ist in dieser Liste inkludiert.
- **GetCountParticipantsOfToken**: Mit dieser Funktion kann die Anzahl an beteiligten Personen abgerufen werden. Dabei ist die Haupthalterin oder der Haupthalter inkludiert.
- **GetTokensWhereAddressParticipates**: Bei dieser Methode wird die Adresse einer Person übergeben. Der Smart Contract sucht anschließend in allen bereits abgeschlossenen Verträgen noch der Identifikation dieser Person. Als Rückgabe liefert diese Funktion eine Liste der Identifikation aller Tokens, bei denen die übergebene Adresse entweder als Besitzerin oder als Besitzer oder als Beteiligte oder Beteiligter hinterlegt ist.
- **GetTokensOfOwner**: Bei dieser Funktion wird ebenfalls die Adresse eines Wallets übergeben. Diese Methode liefert anschließend eine Liste an Identifikationen von Tokens zurück, bei denen diese Person die Haupthalterin oder der Haupthalter ist.
- **Pause**: Falls die Erstellerin oder der Ersteller des Smart Contracts möchte, dass keine neuen Verträge mehr angelegt werden können, kann man mithilfe der Pause-Funktion die Nutzung des Smart Contracts stoppen. Alle bisherigen abgeschlossenen Verträge bleiben weiterhin bestehen, lediglich das Hinzufügen von neuen ist nicht mehr möglich. Es soll auch machbar sein, dass mit dieser Funktion der Smart Contract auch wieder aktiviert werden kann. Das bedeutet, man könnte die Nutzung auch nur temporär stoppen.
- **GetDocumentPath**: Bei dieser Funktion kann man die Dokumente eines Tokens abfragen. Dazu wird lediglich die ID des Tokens benötigt. Anschließend wird genau dieselbe Liste an Pfaden zurückgegeben, die bereits bei der Mint-Methode übergeben wurde.

Weiters gibt es von den Bibliotheken von OpenZeppelin weitere Standard-Funktionen ausgeführt werden können:

- **tokenOfOwnerByIndex**: Falls eine Besitzerin oder ein Besitzer mehrere Tokens hat, dann kann man mithilfe dieser Funktion den gewünschten Token zurückbekommen. Dabei wird die Adresse der Halterin oder des Halters und der Index angegeben.
- **totalSupply**: Jeder Smart Contract, der einen Non-Fungible-Token implementiert, muss eine maximale Anzahl an Tokens enthalten. Jedes Mal, wenn ein Token einer Besitzerin oder einem Besitzer zugeordnet wird, schrumpft diese Zahl. Mit dieser Methode kann man die noch verbleibende Anzahl an Tokens bekommen.
- **ownerOf**: Fall man nur die Id eines Tokens hat und man herausfinden möchte, wem dieser Token gehört, kann man diese Methode aufrufen. Sie gibt die Adresse der Besitzerin oder des Besitzers zurück.
- **name**: Wenn ein Smart Contract erstellt wird, bekommt er von der Herausgeberin oder dem Herausgeber einen Namen. Mit dieser Funktion kann der Name des Smart Contracts abgerufen werden
- **symbol**: Ähnlich wie beim Namen bekommt der Smart Contract beim Erstellen auch ein Symbol. Dieses ist maximal vier Zeichen lang. Mit dieser Funktion kann das Symbol abgerufen werden.
- **transferFrom**: Bei dieser Funktion kann man die Besitzerin oder den Besitzer eines Tokens ändern. Dabei werden die Adressen von wem zu wen und die Id des Tokens übergeben.

Abschließend, bevor der Smart Contract ausgeführt werden kann, muss der Code noch in die Blockchain beziehungsweise in das Test-Netzwerk geladen werden. Dafür muss zuerst die lokale Umgebung mit dem API-Zugang und der Wallet-ID verbunden werden. Weiters wird, um mit der API leichter zu interagieren, eine weitere Library benötigt. Diese Bibliothek nennt sich Ether.js. Nachdem nun alles miteinander verknüpft wurde, muss der Code noch kompiliert werden. Abschließend wird der Code in das Test-Netzwerk deployed. Wenn dies erfolgreich geschehen ist, sollte der Smart Contract als eine eigene Transaktionszeile auf der Plattform Etherscan.io zu sehen sein.

Etherscan.io ist ein Block-Explorer, mit den Transaktionen und Blöcke innerhalb der Ethereum-Blockchain und deren Netzwerken nachverfolgt werden können. Weiters können auf dieser Plattform diverse Statistiken zu Preisverläufen und der Anzahl an Transaktionen eingesehen werden. (Lee, 2019)

Um nun zu überprüfen, ob der entwickelte Code wie gewünscht funktioniert, kann man wieder mithilfe von EtherJs mit dem Smart Contract interagieren. Dazu wird eine der implementierten Funktionen ausgeführt. Nachdem die Funktion des Smart Contracts ausgeführt wurde, sollte man ebenfalls auf der Plattform Etherscan.io eine neue Transaktion sehen. Weiters kann mithilfe einer weiteren Funktion auch überprüft werden, ob der Code auch das gewünschte Ergebnis erzielt hat.

Ein großer Vorteil des Test-Netzwerks ist auch die Vorberechnung der Transaktionskosten. Diese nennt sich bei der Ethereum-Blockchain Gas. Das soll ein Synonym sein für die Energie, die aufgewendet werden musste, um den Smart Contract und die Transaktion zu verarbeiten.

4.6 Integration Testing

Nachdem die Software fertig entwickelt wurde und in das Testnetzwerk von Ethereum deployed wurde, kann der Smart Contract getestet werden. Dafür werden zuerst mindestens 2 Wallets für dieses Netzwerk benötigt. Diese Wallets werden mit der Software MetaMask erstellt.

MetaMask ist ein Tool, um Accounts in verschiedenen Ethereum Netzwerken zu erstellen und zu verwalten. Darin ist es möglich, mehrere Accounts zu besitzen und diese auch bei Bedarf zu wechseln. Weiters ist es auch möglich, einzelne Transaktionen, also Ethers zwischen Accounts zu transferieren. (Lee, 2019)

Mit diesem Tool wurden somit zwei Accounts für das Test-Netzwerk erstellt und Ethers aufgeladen. Im nächsten Schritt konnte der Smart Contract auch schon getestet werden. Dafür wurden mehrere Scripts geschrieben, dass die Funktionen des Smart Contracts aufruft. Dadurch kann geprüft werden, ob der Code des Smart Contracts erfolgreich ausgeführt wurde. Folgende Testfälle abgedeckt:

1. **Mint**: Hier wird ein Vertrag dokumentiert. Dafür wurden beide der erstellten Identifikationen der Wallets übergeben
2. **GetCountParticipantsOfToken**: Hierbei wird die ID des im vorherigen Test erstellten Tokens übergeben und die Anzahl, die vom Blockchain-Netzwerk zurückgeliefert wird, ausgegeben.
3. **GetDocumentPath**: Bei diesem Test soll der Pfad des übergebenen Dokumentes eines bestehenden Vertrags herausgefunden werden. Dazu wird ebenfalls die Identifikation des Tokens aus dem ersten Test übergeben und der Pfad ausgegeben, der im Smart Contract in der Blockchain hinterlegt wird, in der Konsole ausgegeben.
4. **GetParticipantsOfToken**: Dieser Test läuft ähnlich ab, wie die Tests zuvor. Dazu wird die ID des Tokens aus Test 1 übergeben und eine Liste der beteiligten Wallet Identifikationen in der Konsole ausgegeben.
5. **GetTokensOfOwner**: Bei diesem Test wird eine Wallet ID übergeben. Anschließend wird eine Liste ausgegeben mit den Token-IDs, bei denen dieses Wallet als Besitzer hinterlegt ist.
6. **GetTokensWhereAddressParticipates**: Hier wird ebenfalls eine Wallet ID übergeben. Danach wird eine Liste an Identifikationen der Tokens ausgegeben, bei denen das angegebene Wallet, entweder als Besitzer oder auch als Beteiligter hinterlegt ist.

Um nun zu überprüfen, ob der Token und die Transaktionen wirklich in das Blockchain-Netzwerk geschrieben wurden, kann man auf der Seite Etherscan.io die Transaktionen überprüfen. Auf dieser Plattform kann der Smart Contract und die Transaktionen, die mit diesem erstellt wurden, überprüft werden. Weiters kann man auch mit der Adresse der Wallets alle Transaktionen sehen, die damit getätigt wurden. (Lee, 2019)

Nachdem nun das Script ausgeführt wurde, kann auf dieser Plattform die Adresse des Smart Contracts sowie die Adressen der Wallets eingegeben werden. Hier konnte nachvollzogen werden, dass alle Transaktionen erfolgreich in die Blockchain geschrieben wurden.

Um die auch die anderen Funktionen zu testen, wurde ein weiteres Script entwickelt, welches alle Funktionen aufruft. Dabei wurde überprüft, ob immer die Daten, die zurückgeliefert wurden, korrekt waren. Somit kann auch sichergestellt werden, dass alle Methoden wie gewünscht funktionieren.

4.7 System Testing

Hier werden nun die Anforderungen aus Schritt 4.2 und das festgelegte Design aus Kapitel 4.3 mit den tatsächlichen Prototypen verglichen. Dabei wurden die Anforderungen nochmals als Liste angefertigt, um leichter sicherstellen zu können, dass alle Punkte erfüllt worden sind.

- Erstellen eines Tokens
- Festhalten eines Vertrages
- Festhalten der Akteure
- Hinterlegen eines oder mehrere Dokumente
- Schreiben der Transaktion in die Blockchain
- Abfragen der abgelegten Verträge einer Person
- Abfragen der Beteiligten eines Vertrages
- Abfragen der Dokumente eines Vertrages
- Abfragen der weiteren Details eines Vertrages
- Pausieren des Smart Contracts
- Wiederaufnehmen des Smart Contracts

Diese Liste wurde mit den Prototypen abgeglichen und sichergestellt, dass alle Punkte erfüllt wurden. Falls ein Punkt noch nicht erfüllt wurde, musste die Software geändert werden und nach dieser Änderung wurde diese Aufzählung der Anforderungen nochmals überprüft. Dieser Prozess wurde so lange wiederholt, bis der Prototyp allen definierten Anforderungen entsprochen hat.

4.8 Deployment and Operation

Hier wurde nun sichergestellt, dass der Smart Contract allen Anforderungen entspricht und dieser auch fehlerfrei funktioniert. Dabei ist es wichtig, auf die Qualität des Codes zu achten, da sobald der Smart Contract einmal in der Blockchain ist, nicht mehr verändert werden kann. Würden nach diesem Schritt noch Fehler entdeckt werden, wäre dies Fatal und es müsste ein neuer Smart Contract in die Blockchain deployed werden. Bereits bestehende Transaktionen und Tokens könnten dabei nicht rückgängig oder ausgebessert werden.

Das Deployment in die Blockchain funktioniert dabei genau gleich wie in das Test-Netzwerk. Dazu wird genau dieselbe API aufgerufen wie im Entwicklungsprozess. Lediglich wurden die Zugangsdaten des Goerli-Netzwerkes, mit denen des Ethereum-Netzwerkes ausgetauscht.

Nachdem das Deployment abgeschlossen ist, ist der Smart Contract in Betrieb und kann genutzt werden. Das bedeutet, jede Person, die ein Wallet besitzt, kann die Mint-Funktion aufrufen, um sich einen Token zuzuweisen. Hierbei ist zu beachten, dass es sich nicht mehr um das Test-Netzwerk handelt. Das bedeutet, es werden Transaktionskosten für das Ausführen des Smart-Contracts anfallen, die von der Person bezahlt werden müssen, die den Mint-Vorgang gestartet hat.

5 ERGEBNISSE

In diesem Kapitel werden die Ergebnisse der durchgeführten Evaluierung anhand der beschriebenen Metriken dargestellt.

5.1 Evaluation des Smart Contracts

In diesem Teil werden die beschriebenen Metriken aus Kapitel 3.1.2 ausgewertet. Bei diesen Metriken liegt das Augenmerk direkt auf den Smart Contract und dessen Qualität selbst. Dabei wurde er anhand der beschriebenen Tools überprüft.

5.1.1 Gas Price

Ein ausschlaggebender Punkt, der die Verwendbarkeit von Smart Contracts beeinflusst, ist der Gas Price. Das ist der Preis, der bezahlt werden muss, um eine Transaktion in die Blockchain zu schreiben oder eine Funktion auszuführen.

Grundsätzlich muss für jede Transaktion, die in der Blockchain gemacht wird, der Gas Price bezahlt werden. In der folgenden Tabelle sind alle Funktionen die der Smart Contract unterstützt mit ihrem Gas Price aufgelistet:

Funktion	**Gas Price**
Deployment	8 Gwei
Mint	3.042412774 Gwei
GetParticipantsOfToken	0 Gwei
GetCountParticipantsOfToken	0 Gwei
GetTokensWhereAddressParticipates	0 Gwei
GetTokensOfOwner	0 Gwei
pause	1.500152429 Gwei
GetDocumentPath	0 Gwei
tokenOfOwnerByIndex	0 Gwei
totalSupply	0 Gwei
ownerOf	0 Gwei
name	0 Gwei
symbol	0 Gwei
transferFrom	1.500117264 Gwei

Der Gas Price wird in Gwei angegeben. Bei der Ethereum-Blockchain gibt es ähnlich wie beim Euro den Cent, auch kleinere Konfession. Diese nennt sich Wei. Gwei bedeutet dabei Giga wei. Das bedeutet ein Gwei sind 1.000.000.000 Wei. Wenn alle Funktionen einmal ausgeführt werden, dann erhält man insgesamt 14,04 Gwei. Laut derzeitigem Stand des Ethers (13.02.2023) wären somit die Kosten bei 0,000019 Euro, um alle Funktionen auszuführen. Möchte man nur einen neuen Vertrag hinzufügen, sind nur die Kosten der Mint-Methode zu bezahlen. Das bedeutet, dass die Kosten pro neuen Vertrag auf 0,0000041 Euro betragen.

5.1.2 Codequalität

Hierbei wurde die Codequalität mittels automatisierter Software überprüft. Dazu gibt es dafür speziell entwickelte Programme, die die Codequalität von Smart Contracts überprüfen können. Das erste Tool, das verwendet wurde, ist SmartCheck.

Bei der Ausführung der Analyse durch diese Software wurden fünf potenzielle Probleme gefunden, die zu Fehlern führen könnten. Diese Probleme traten im gesamten Smart Contract insgesamt elf Mal auf. In der Abbildung 5-1 ist ersichtlich, wie oft welches Problem gefunden wurde. In der folgenden Aufzählung wird erläutert, was welches Problem bedeutet und warum dies aufgetreten ist:

- **SOLIDITY_VISIBILITY:**

  Dieses Problem bedeutet, dass eine Funktion im Smart Contract keine Sichtbarkeit definiert hat. Standardmäßig ist die Sichtbarkeit bei Funktionen public, was bedeutet, dass sie von jedem aufgerufen werden kann.

  In diesem Fall wurde dieses Problem beim Konstruktor des Smart Contracts identifiziert. Für den Konstruktor ist das Definieren der Sichtbarkeit allerdings nicht notwendig, denn dieser ist keine Funktion, sondern wird nur für das Initialisieren des Smart Contracts verwendet.

- **SOLIDITY_LOCKED_MONEY:**

  Dieses Problem wird dann identifiziert, wenn es in diesem Smart Contract keine Möglichkeit gibt, die aufgebuchten Coins einzulösen und auf ein anderes Konto zu überweisen. Diese Funktion wird besonders bei NFTs genutzt, da dadurch das Guthaben an die Besitzerin oder den Besitzer ausgeschüttet werden kann. Für den Smart Contract in dieser Masterarbeit ist diese Meldung allerdings irrelevant, da es keine Möglichkeit gibt, auf den Vertrag selbst Guthaben abzulegen. Infolgedessen ist es auch nicht notwendig, das Guthaben auszahlen lassen zu können.

- **SOLIDITY_PRIVATE_MODIFIER_DONT_HIDE_DATA:**

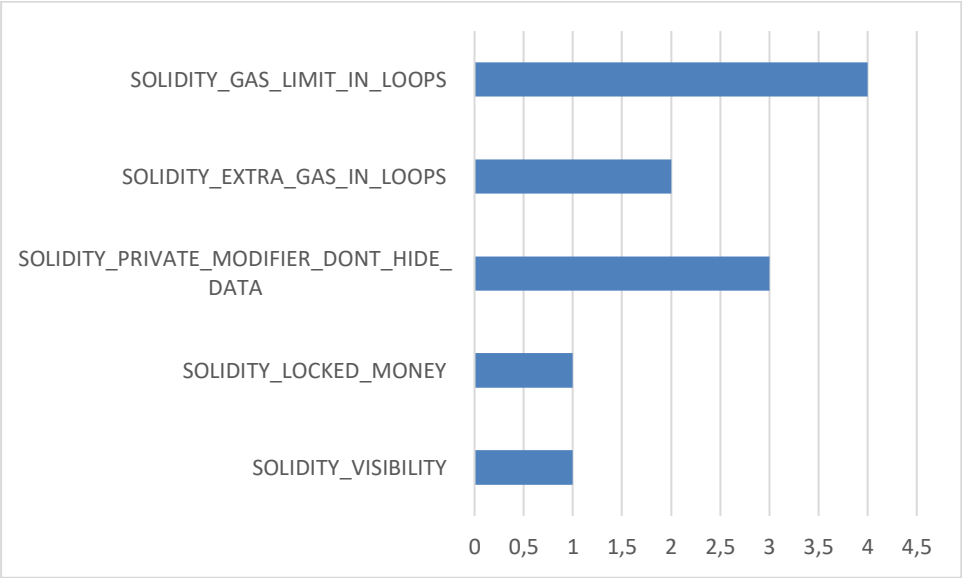
  Dieses identifizierte Problem ist lediglich eine Warnung. Da ein Smart Contract in die Blockchain deployed wird, ist es auch möglich für jede Person, die Zugriff auf die Blockchain hat, den Code zu sehen. Diese Meldung warnt die Entwicklerin oder den Entwickler davor, dass Funktionen oder Variablen, die mit der Sichtbarkeit „Privat“ gekennzeichnet wurden, trotzdem für jeden sichtbar sind. Diese sind lediglich nicht für jeden ausführbar.

- **SOLIDITY_EXTRA_GAS_IN_LOOPS:**

  Das Abfragen der Länge eines Feldes verursacht einen höheren Gas Preis. Deshalb wird es empfohlen, diesen Wert in eine Variable zu speichern, um die Kosten zu senken. In den beiden Fällen, wo dieses Problem identifiziert wurde, wurde die Länge eines Feldes einmal, und zwar in der Schleife abgefragt. Deshalb ist diese Warnung ebenfalls irrelevant.

- **SOLIDITY_GAS_LIMIT_IN_LOOPS:**

  Wenn der Gas Preis bei Funktionen zu hoch ist, kann es sein, dass das Blockchain-Netzwerk diese nicht oder nur sehr zeitverzögert ausführt, da Funktionen mit niedrigen Kosten bevorzugt werden. Diese Meldung soll vor dem überschreiten eines Limits warnen, was vorkommen kann, wenn rechenintensive Funktionen ausgeführt werden. Dies könnte zur Folge haben, dass sie nicht wie gewünscht ausgeführt werden.



5-1 SmartCheck - potenzielle Fehler

Bereits die Auswertung der Ergebnisse des ersten Tools zeigt, dass der Code noch einige Probleme hat. Einige der identifizierten Probleme können allerdings ignoriert werden, da sie für diesen Anwendungsfall keinerlei Relevanz haben. Besonders Fokus sollte auf die Codeteile geworfen werden, die die Meldung SOLIDITY_GAS_LIMIT_IN_LOOPS produzieren. Diese zu beheben könnte allerdings eine Herausforderung darstellen, da aufgrund der Suchfunktionen gewisse Operationen durchgeführt werden müssen, die in der Blockchain sehr kostspielig sind.

5.1.3 Vergleich mit Microsoft Excel

In diesem Kapitel werden die Metriken erhoben, um den Smart Contract und Microsoft Excel zu vergleichen. Um die Werte zu erheben, wurden die beiden beschriebenen Testfälle durchgeführt und die Daten erhoben. In der folgenden Aufzählung werden die Daten zu deinen einzelnen Kategorien und Metriken aufgeschlüsselt:

- **Effizienz:**
    - *Dauer, um Verträge zu dokumentieren:*

      Um bei der Dauer aussagekräftige Ergebnisse zu bekommen, wurden die beiden definierten Testfälle durchgeführt und dabei die Zeit die für das durchführen der Testfälle benötigt wurde gestoppt. Dabei kam es zu folgenden Ergebnissen:

      Zeiten für das Dokumentieren der Verträge aus Testfall 1:

      Microsoft Excel:

      Gesamtdauer: 330,03 Sekunden

      Durchschnittsdauer pro Vertrag: 33 Sekunden

      Smart Contract:

      Gesamtdauer: 327,87 Sekunden

      Durchschnittdauer pro Vertrag: 32,78 Sekunden

      Zeiten für das Abfragen der Daten:

      Microsoft Excel:

      Gesamtdauer: 60,95 Sekunden

      Durchschnittsdauer pro Abfrage: 8,70 Sekunden

      Smart Contract:

      Gesamtdauer: 85,04 Sekunden

      Durchschnittsdauer pro Abfrage: 12,14 Sekunden

      Zeiten für das Dokumentieren der Verträge aus Testfall 2:

      Microsoft Excel:

      Gesamtdauer: 162,33 Sekunden

      Durchschnittsdauer pro Vertrag: 32,46 Sekunden

      Smart Contract:

      Gesamtdauer: 216,38 Sekunden

Durchschnittdauer pro Vertrag: 43,27 Sekunden

Zeiten für das Abfragen der Daten:

Microsoft Excel:

Gesamtdauer: 47,09 Sekunden

Durchschnittsdauer pro Abfrage: 9,41Sekunden

Smart Contract:

Gesamtdauer: 48,16 Sekunden

Durchschnittsdauer pro Abfrage: 9,63 Sekunden

Bei Testfall 1 kann erkannt werden, dass beim Durchführen der Testfälle der Unterschied der benötigten Zeit zwischen Microsoft Excel und dem Prototyp nicht groß ist.

Allerdings sind die Abfragen der Daten in Excel schneller.

Beim Testfall 2 gibt es einen größeren Unterschied. Dabei war die Zeit, die benötigt wurde, um Verträge über Microsoft Excel zu dokumentieren, kürzer. Bei den Abfragen der Daten gab es hier jedoch im Vergleich zu Testfall eins wenige Unterschiede.

- *Fehlerhäufigkeit:*

  Während der Durchführung der Testfälle wurde auch mitdokumentiert, wie oft ein Fehler passiert ist. Dabei kam es zu folgenden Ergebnissen:

  Anzahl der Fehler, die beim Dokumentieren der Verträge bei Testfall 1 aufgetreten sind:

  Microsoft Excel: 1

  Smart Contract: 1

  Anzahl der Fehler, die bei den Abfragen bei Testfall 1 aufgetreten sind:

  Microsoft Excel: 0

  Smart Contract: 0

Anzahl der Fehler, die beim Dokumentieren der Verträge bei Testfall 1 aufgetreten sind:

Microsoft Excel: 0

Smart Contract: 2

Anzahl der Fehler, die bei den Abfragen bei Testfall 1 aufgetreten sind:

Microsoft Excel: 0

Smart Contract: 0

Insgesamt passierten bei beiden Testfällen beim Smart Contract mehr Fehler als bei Microsoft Excel. Dies lässt darauf schließen, dass besonders bei komplexeren Verträgen das Dokumentieren dieser auch komplizierter wird.

- **Sicherheit:**
  - *Schutz gegen unbefugten Zugriff:*

    Microsoft Excel: Grundsätzlich können Excel Tabellen mit einem Passwort geschützt werden. Die Verträge selbst werden in der Tabelle nur als Pfad hinterlegt. Somit können die Dokumente an einem gesicherten Ort abgelegt werden.

    Ergebnis: Gut

    Smart Contract: Jede Person, die auf die Blockchain Zugriff hat, kann auch die Daten des dokumentieren Vertrages abfragen. Gleich wie bei Microsoft Excel wird nur der Pfad zum Dokument in der Blockchain gespeichert, wodurch der Zugriff verwaltet werden kann.

    Ergebnis: Mittel

    Dadurch, dass beim Tabellenkalkulationsprogramm der Zugriff der Dokumentation auch besser geschützt werden kann, ist hier die Bewertung von Microsoft Excel besser.

  - *Schutz vor Änderung:*

    Microsoft Excel: Die Daten können im späteren Verlauf immer wieder geändert werden. Das bedeutet, dass auch andere Akteure oder Pfade zu anderen Dokumenten hinterlegt werden können.

    Bewertung: Schlecht

    Smart Contract: Eine nachträgliche Änderung eines bereits dokumentierten Vertrages ist nicht möglich. Alle Daten, die in die Blockchain geschrieben wurden, werden auch für immer dort gespeichert bleiben.

    Bewertung: Gut

- **Skalierbarkeit:**
    - *Anzahl Personen, die gleichzeitig Verträge dokumentieren können:*

      Microsoft Excel: An einem Dokument, dass auf einem Computer wie in diesem Szenario abgelegt wurde, kann eine Person gleichzeitig arbeiten.

      Smart Contract: Da der Smart Contract und die Daten vom Blockchain-Netzwerk abgearbeitet werden, gibt es keine maximale Zahl an Personen, die gleichzeitig einen Vertrag dokumentieren können.

- **Transparenz:**
    - *Nachverfolgbarkeit der Aktionen:*

      Microsoft Excel: Da das Dokument auf einem Computer gespeichert wurde, wird dieses auch immer wieder neu überschrieben. Das bedeutet, es kann nicht nachvollzogen werden, wann genau welche Aktion passiert ist. Lediglich wann die letzte Änderung passiert ist, kann herausgefunden werden.

      Bewertung: Schlecht

      Smart Contract: Da jede Transaktion in die Blockchain geschrieben wird, kann auch von jeder Person, die Zugriff auf das Netzwerk hat, nachvollzogen werden, wann welcher Vertrag erstellt wurde.

      Bewertung: Gut

- **Kosten:**
    - *Kosten pro Vertrag:*

      Microsoft Excel: Pro Vertrag fallen bei dieser Software keine Kosten an.

      Ergebnis: 0,00 €

      Smart Contract: Pro Transaktion muss der Gas Price bezahlt werden dieser beläuft sich auf 3**.**042412774 Gwei.

      Ergebnis: Stand 13.02.2023: 0,0000046 €
    - *Kosten pro Jahr:*

      Microsoft Excel: Bei Excel muss jährlich eine Lizenz pro User gekauft werden. Alternativ könnte es auch einmalig für 150€ gekauft werden. Für diesen Fall wird allerdings die Jährliche Lizenz in Betracht gezogen, da der Funktionsumfang größer ist. (Microsoft, 2023)

      Ergebnis: 69,00 €

      Smart Contract: Nachdem der Smart Contract einmal deployed wurden, fallen keine jährlichen Kosten an.

      Ergebnis: 0,00 €

- **Flexibilität:**
  - *Einfachheit, Anwendung anzupassen:*

    Microsoft Excel: Die Tabelle und die Vorlage können beliebig oft angepasst werden.

    Bewertung: Gut

    Smart Contract: Das Anpassen des Codes ist möglich, allerdings muss anschließend ein neuer Smart Contract in die Blockchain deployed werden. Das bedeutet, dass der laufende Smart Contract nicht bearbeitet werden kann.

    Bewertung: Mittel

  - *Möglichkeiten zur Integration:*

    Microsoft Excel: Grundsätzlich gibt es für Microsoft Excel direkt keine Schnittstellen, die die Daten direkt von der lokalen Datei auslesen können. Es gibt allerdings Möglichkeiten, bei diversen Tools mit Excel erstellte Dokumente zu importieren und die importierten Daten weiter zu nutzen.

    Bewertung: Mittel

    Smart Contract: Da die Ethereum-Blockchain öffentlich ist, gibt es auch Schnittstellen, mit denen man davon Daten auslesen und schreiben kann. Weiters ist es ebenso möglich direkt beim Smart Contract Schnittstellen zu integrieren, auf diese zugegriffen werden kann.

    Bewertung: Gut

  - *Anzahl, wie oft die Funktionen angepasst werden können:*

    Microsoft Excel: Da es sich um ein editierbares Dokument handelt, kann das Dokument auch beliebt oft an die sich ändernden Anforderungen angepasst werden.

    Ergebnis: unendlich

    Smart Contract: Der Code kann ebenfalls beliebig oft angepasst werden, allerdings muss der Smart Contract neu deployed werden.

    Ergebnis: 0

5.2 Zusammenfassung der Ergebnisse

Die Ergebnisse zeigen, dass die Blockchain-Technologie in einigen Bereichen Vorteile bringen kann. Ein großer Punkt dabei ist die Transparenz und Datensicherheit. Dabei können die bestehenden Verträge und Transaktionen lückenlos in der Blockchain nachvollzogen werden. Ein weiterer Vorteil ist auch in den Kosten erkennbar. Denn einen einzelnen Vertrag zu dokumentieren, kostet weniger als einen Cent, während bei der Nutzung von Microsoft Excel jährlich Lizenzkosten anfallen.

Es konnten aber auch einige Nachteile ermittelt werden. Einerseits weist der Code des Prototyps noch einige Probleme auf. Diese sind zwar nicht so ausschlaggebend, jedoch sollten sie nicht außer Acht gelassen werden. Weiters ergeben sich im Vergleich mit Microsoft Excel auch weitere Nachteile. Beispielsweise ist die Zeit, die benötigt wird, um einen Vertrag zu dokumentieren oder Daten abzufragen, oft länger als mit der konventionellen Software. Dies kann besonders im Arbeitsalltag zu Hindernissen führen. Ein weiterer großer Nachteil des Prototyps ist auch, dass er Fehleranfälliger ist als Microsoft Excel. Dies liegt besonders daran, dass der Prototyp nur über Konsolen-Befehle verwendet werden kann. Was zur Folge hat, dass die Nutzerfreundlichkeit geschmälert wird. Weiters gibt es auch kein direktes Behandeln und Reagieren auf Fehler. Das bedeutet, dass die Nutzerin oder der Nutzer selbst auf Fehler reagieren und diese beheben muss.

5.3 Ausblick

Die vorliegende Arbeit setzt sich aus der Literatur, der Erfahrung aus der Implementierung des Prototyps und der Evaluierung zusammen. Als nächster Schritt sollte ein weiterer Prototyp entwickelt oder der bestehende Prototyp erweitert werden, in der die nachgewiesenen Probleme im Code behandelt werden. Diese könnten in der zukünftigen Arbeit vollständig beseitigt werden, um gegebenenfalls in einem Produktivbetrieb Sicherheitslücken und Fehler zu vermeiden.

Auf Basis dieses Prototyps könnte auch in weiteren Schritten eine grafische Oberfläche entwickelt werden, wodurch Verträge leichter dokumentiert werden können. Des Weiteren sollten in dieser grafischen Oberfläche ebenfalls alle Funktionen, um Daten in der Blockchain abzufragen, implementiert sein, um den vollen Funktionsumfang zu gewährleisten. Durch die Entwicklung dieser Software wäre es auch möglich, dass diese Technologie für Personen ohne technisches Wissen zugänglich wird und die Fehler, die in der Evaluierung passiert sind, könnten eventuell vermieden werden.

Weiters könnte das Einrichten und Verwenden einer privaten Blockchain für diesen Verwendungszweck evaluiert werden. Der Vorteil davor wäre, dass die dokumentierten Daten nicht mehr öffentlich einsehbar wären. Weiters wäre auch möglich, Daten, die in der Vergangenheit gespeichert wurden, wieder zu löschen. Besonders diese Funktion ermöglich das Einhalten der DSGVO-Richtlinien einfacher. Somit hätte diese Art der Technologie das Potenzial, ohne Umwege rechtskonform zu sein.

6 ZUSAMMENFASSUNG

Übergeordnetes Ziel dieser Masterarbeit ist es, zu überprüfen, wie Unternehmen Blockchain-Technologien für ihren Vorteil nutzen können. Dabei wurde der Anwendungsfall des Dokumentierens von Verträgen zwischen Mitarbeiterinnen und Mitarbeitern genauer betrachtet. Dementsprechend wurde zuerst eine theoretische Grundlage mithilfe einer Literaturrecherche geschaffen. Dabei wurden bereits bestehende Anwendungsfälle erläutert und analysiert und welche Arten es gibt, um Verträge in der Blockchain zu dokumentieren. Anschließend wurde beschrieben, wie der Entwicklungsprozess einer Applikation in der Blockchain ablaufen kann. Im Anschluss wurde ein Prototyp implementiert, mit dem Verträge in der Blockchain dokumentiert werden können. Nachfolgend wurden Metriken erhoben, mit denen die entwickelte Software auf deren Einsatztauglichkeit überprüft werden kann. Mithilfe der Ergebnisse hat sich herausgestellt, dass Blockchain-Technologien das Potenzial haben, bestehende Prozesse zu verändern und auch Verträge damit dokumentiert werden können und somit auch die Gültigkeit eines Vertrages zwischen Mitarbeiterinnen oder Mitarbeitern und dem Unternehmen nachgewiesen werden kann. Allerdings gibt es derzeit auch noch Nachteile, die nicht vernachlässigt werden sollten.

Zusammenfassend ist die Dokumentation von Verträgen zwischen Mitarbeiterinnen und Mitarbeitern in Unternehmen ein wichtiger Bestandteil, der ernst genommen werden sollte. Um alle rechtlichen Anforderungen und auch den Überblick über die Verträge zu behalten, kann eine Software für diesen Anwendungsfall verwendet werden. Der Großteil der Unternehmen verwendet als Tool nach wie vor Microsoft Excel, da viele Personen bereits Kenntnisse mit dieser Software haben und sie auch bereits auf vielen Geräten installiert ist. Blockchain-Technologien wurden in den letzten Jahren immer bekannter und deren Anwendungsgebiete wachsen stetig. Deren großer Vorteil ist, dass die gespeicherten Daten miteinander verkettet sind und sie über ein Netzwerk an Rechnern verifiziert werden. Dadurch ist eine Manipulation nicht möglich. Durch diesen Vorteil kann ebenso eine lückenlose und ausfallsichere Dokumentation der Verträge in einem Unternehmen gewährleistet werden.

Der entwickelte Prototyp und die Ergebnisse der Evaluierung haben gezeigt, dass die Technologie das Potenzial hat, um für den beschriebenen Einsatzzweck verwendet zu werden. Allerdings gibt es noch Probleme, die dafür sorgen, dass zum jetzigen Zeitpunkt der Prototyp in Unternehmen nicht eingesetzt werden sollte. Da die Funktionen, die der Smart Contract implementiert, nur anhand von Konsolen-Befehlen ausgeführt werden können, ist technisches Wissen notwendig. Weiters gibt es auch einige Punkte, in denen die konventionelle Software besser ist, wodurch sich die Schwelle, diese Technologie zu Nutzen weiter erhöht. Da diese Technologie allerdings noch neu ist, kann davon ausgegangen werden, dass es auch in der Zukunft noch weitere Entwicklungen geben wird, was einige der Probleme lösen könnte. Ab dann könnte sie auch in einem Unternehmen zum Dokumentieren von Verträgen genutzt werden.

ANHANG

> Gegenüberstellung Microsoft Excel und Smart Contract

Metrik	Excel	SC
E1	600,4	677,45
E2	1	3
S1	Gut	Mittel
S2	Schlecht	Gut
Sk1	1	999
T1	Schlecht	Gut
K1	0	0,0000046
K2	69	0
F1	Gut	Mittel
F2	Mittel	Gut

> Zusammenfassung ordinaler Metriken

Excel:

Skala	Anzahl
Schlecht	2
Mittel	1
Gut	2

Smart Contract:

Skala	Anzahl
Schlecht	0
Mittel	2
Gut	3

> E1_E2_Testfall1

Dokumentation:

Vertrag	**Excel**	**Fehler**	**SC**	**Fehler**
1	35,37	0	32,16	0
2	28,76	0	29,7	0
3	31,17	0	33,24	0
4	43,17	1	38,2	0
5	30,22	0	37,69	0
6	31,86	0	32,99	1
7	34,49	0	27,26	0
8	30,86	0	30,59	0
9	33,21	0	30,97	0
10	30,92	0	35,07	0
Summe	330,03	1	327,87	1
AVG	33,003	0,1	32,787	0,1

Abfragen:

Test	Excel	Fehler	SC	Fehler
1	7,1	0	7,38	0
2	5,22	0	9,51	0
3	8,43	0	12,67	0
4	11,17	0	19,5	0
5	10,18	0	15,37	0
6	8,86	0	12,28	0
7	9,99	0	8,33	0
Summe	60,95	0	85,04	0
AVG	8,70714286	0	12,1485714	0

> E1_E2_Testfall2

Dokumentation:

Vertrag	Excel	Fehler	SC	Fehler
1	34,39	0	49,6	1
2	30,88	0	53,6	1
3	35,86	0	33,36	0
4	28,91	0	31,35	0
5	32,29	0	48,47	0
Summe	162,33	0	216,38	2
AVG	32,466	0	43,276	0,4

Abfragen:

Test	Excel	Fehler	SC	Fehler
1	9,02	0	9,26	0
2	6,99	0	6,66	0
3	10,73	0	12,92	0
4	11,25	0	6,82	0
5	9,1	0	12,5	0
Summe	47,09	0	48,16	0
AVG	9,418	0	9,632	0

> S1

	Excel	SC
Wie lassen sich die Dokumente vor unbefugten Zugriff schützen?	Zugriffsverwaltung auf Link	Zugriffsverwaltung auf Link
Wie lässt sich die Dokumentation selbst vor unbefugten Zugriff schützen?	Excel mit Passwort schützen	-
Bewertung	Gut	Mittel
Bewertung in Punkten	2	1

> S2

	Excel	SC
Können Daten nachträglich geändert werden?	Ja	Nein
Sind Änderungen nachvollziehbar und versioniert?	Nein	Ja
Bewertung	Schlecht	Gut
Bewertung in Punkten	0	2

> Sk1

	Excel	SC
Anzahl an Personen, die gleichzeitig Editieren können	1	unendlich

> T1

	Excel	SC
Kann nachvollzogen werden, wann welcher Vertrag dokumentiert wurde?	Nein	Ja
Kann nachgewiesen werden, wann Änderungen passiert sind?	Nein	Ja
Bewertung	Schlecht	Gut
Bewertung in Punkten	0	2

> K1

	Excel	SC
Kosten pro Vertrag	0	0,0000046 €

> K2

	Excel	SC
Kosten pro Jahr	69 €	0

> F1

	Excel	SC
Kann der Prototyp/das Tool an die Anforderungen angepasst werden?	Ja	Ja
Kann der Prototyp/das Tool nachdem bereits Verträge dokumentiert wurden, angepasst werden?	Ja	Nein
Bewertung	Gut	Mittel
Bewertung in Punkten	2	1

> F2

	Excel	SC
Gibt es Schnittstellen zur Integration mit anderen Systemen?	Nein	Ja
Gibt es andere Möglichkeiten zur Integration mit anderen Systemen?	Ja	JA
Bewertung	Mittel	Gut
Bewertung in Punkten	1	2

ABKÜRZUNGSVERZEICHNIS

API ..... *Application Programming Interface*
AWS ..... *Amazon Web Services*
bzw ..... *beziehungsweise*
CRUD ..... *Create, Read, Update, Delete*
DSGVO ..... *Datenschutz-Grundverordnung*
eIDAS ..... *electronic IDentification, Authentication and trust Services*
ERC ..... *Ethereum Request for Comments*
EU ..... *Europäische Union*
ID ..... *Identifikation*
IoT ..... *Internet of things*
KYC ..... *Know your customer*
NFT ..... *Non-Fungible Token*
SC ..... *Smart Contract*
UI ..... User Interface
USD ..... *US-Dollar*
XML ..... *Extensible Markup Language*

ABBILDUNGSVERZEICHNIS

7 LITERATURVERZEICHNIS

(2021, March - 2021, March). *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE: IEEE.

Abou Jaoude, J., & George Saade, R. (2019). Blockchain Applications – Usage in Different Domains. *IEEE Access*, *7*, 45360–45381. https://doi.org/10.1109/ACCESS.2019.2902501

Alchemy Insights, I. (2023). Alchemy. Retrieved from https://www.alchemy.com/

Ammann, P., & Offutt, J. (2017). *Introduction to software testing* (Second edition). Cambridge, United Kingdom, New York, NY, Melbourne, Australia, New Delhi, India, Singapore: Cambridge University Press.

Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., . . . Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, *100*, 143–174. https://doi.org/10.1016/j.rser.2018.10.014

Ante, L. (2021). Smart contracts on the blockchain – A bibliometric analysis and review. *Telematics and Informatics*, *57*, 101519. https://doi.org/10.1016/j.tele.2020.101519

AppMagic (2022). Most popular Microsoft apps worldwide in 2nd quarter 2022, by downloads (in millions). Retrieved from https://www.statista.com/statistics/1268166/most-downloaded-microsoft-apps-worldwide/

Bamakan, S. M. H., Nezhadsistani, N., Bodaghi, O., & Qu, Q. (2022). Patents and intellectual property assets as non-fungible tokens; key technologies and challenges. *Scientific Reports*, *12*(1), 2178. https://doi.org/10.1038/s41598-022-05920-6

Bao, H., & Roubaud, D. (2022). Non-Fungible Token: A Systematic Review and Research Agenda. *Journal of Risk and Financial Management*, *15*(5), 215. https://doi.org/10.3390/jrfm15050215

Bauer, D. P. (2022). *Getting Started with Ethereum: A Step-by-Step Guide to Becoming a Blockchain Developer* (1st ed. 2022). *Springer eBook Collection*. Berkeley, CA: Apress; Imprint Apress. https://doi.org/10.1007/978-1-4842-8045-4

Beaumier, G., & Kalomeni, K. (2021). Ruling through technology: politicizing blockchain services. *Review of International Political Economy*, 1–24. https://doi.org/10.1080/09692290.2021.1959377

Benahmed, S., Pidikseev, I., Hussain, R., Lee, J., Kazmi, S. A., Oracevic, A., & Hussain, F. (2019, September 8–11). A Comparative Analysis of Distributed Ledger Technologies for Smart Contract Development. In *2019 IEEE 30th Annual International Symposium on*

Personal, Indoor and Mobile Radio Communications (PIMRC) (pp. 1–6). IEEE. https://doi.org/10.1109/PIMRC.2019.8904256

Brühl, V. (2017). Bitcoins, Blockchain und Distributed Ledgers. *Wirtschaftsdienst*, *97*(2), 135–142. https://doi.org/10.1007/s10273-017-2096-3

Chen, Y., Venkatesan, R., Cary, M., Pang, R., Sinha, S., & Jakubowski, M. H. (2003). Oblivious Hashing: A Stealthy Software Integrity Verification Primitive. In G. Goos, J. Hartmanis, J. van Leeuwen, & F. A. P. Petitcolas (Eds.), *Lecture Notes in Computer Science. Information Hiding* (Vol. 2578, pp. 400–414). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36415-3_26

Chou, E. Y. (2015). Paperless and Soulless. *Social Psychological and Personality Science*, *6*(3), 343–351. https://doi.org/10.1177/1948550614558841

Crnkovic, I. (2001). Component-based software engineering ? new challenges in software development. *Software Focus*, *2*(4), 127–133. https://doi.org/10.1002/swf.45

Dib, O., Brousmiche, K.-L., Durand, A., Thea, E., & Hamida, E. (2018). Consortium Blockchains: Overview, Applications and Challenges.

Dieluweit, M. H. (2017). Lean Management am Legal Operations-Arbeitsplatz. In R. P. Falta & C. Dueblin (Eds.), *Praxishandbuch Legal Operations Management* (pp. 605–618). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-50506-9_43

DocuSign (2019). *State of Contract Management 2019.*

Düring, T., & Fisbeck, H. (2017). Einsatz der Blockchain-Technologie für eine transparente Wertschöpfungskette. In A. Hildebrandt & W. Landhäußer (Eds.), *Management-Reihe Corporate Social Responsibility. CSR und Digitalisierung* (pp. 449–464). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53202-7_33

Information and communication technology for energy efficiency: Conference proceedings ; EnviroInfo 2014 - 28th International Conference on Informatics for Environmental Protection ; 10th - 12th September, 2014, Oldenburg, Germany (2014). Oldenburg: BIS-Verl. der Carl-von-Ossietzky-Univ. Retrieved from http://nbn-resolving.de/urn:nbn:de:gbv:715-oops-20009

Ethereum (2022). ERC-1155 Multi-Token Standard. Retrieved from https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/

Art.26 - Gemeinsam für die Verarbeitung Verantwortliche (2016).

Europäisches Parlament; Panel for the Future of Science and Technology (2019). *Blockchain and the general data protection regulation: Can distributed ledgers be squared with European data protection law? : Study*. Brussels: European Union. https://doi.org/10.2861/535

Fenton, N. E., & Pfleeger, S. L. (1998). *Software metrics: A rigorous and practical approach* (2. ed., rev. print). Boston: PWS Publ.

Formby, S. K., Medlin, B. D., & Ellington, V. (2017). Microsoft Excel®: Is It An Important Job Skill for College Graduates? *Information Systems Education Journal (ISEDJ)*, 55–63.

Fries, M., & Paal, B. P. (Eds.) (2019). *Smart Contracts*. Mohr Siebeck GmbH and Co. KG.

Gerlach, R., Gruber-Risak, Martin, 1969-, Höfle, W., & Schrank, Franz, 1945- (2019). *Praxishandbuch Arbeitsvertragsgestaltung* (2. Auflage). [Wien] : Linde Verlag Ges.m.b.H. Retrieved from https://permalink.obvsg.at/fwg/AC15474385

Hameurlain, A., Küng, J., Wagner, R., Dang, T. K., & Thoai, N. (Eds.) (2017). *Lecture Notes in Computer Science*. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXVI*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-56266-6

Heinze, C., & Ojea, G. P. (2018). Der Beweis mit privaten elektronischen Dokumenten nach ZPO und eIDAS-VO. *Computer Und Recht*, *34*(1), 37–44. https://doi.org/10.9785/cr-2018-0111

Himmer, K. (2019). *Blockchain-basiertes Fundraising als innovative Alternative der Unternehmensfinanzierung*. Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-26816-9

Hoffmann, T., & Skwarek, V. (2019). Blockchain, Smart Contracts und Recht. *Informatik-Spektrum*, *42*(3), 197–204. https://doi.org/10.1007/s00287-019-01180-3

Hölbl, M., Kompara, M., Kamišalić, A., & Nemec Zlatolas, L. (2018). A Systematic Review of the Use of Blockchain in Healthcare. *Symmetry*, *10*(10), 470. https://doi.org/10.3390/sym10100470

K.Pandey, S., & Batra, M. (2013). Formal Methods in Requirements Phase of SDLC. *International Journal of Computer Applications*, *70*(13), 7–14. https://doi.org/10.5120/12020-8017

Kapsoulis, N., Psychas, A., Palaiokrassas, G., Marinakis, A., Litke, A., & Varvarigou, T. (2020). Know Your Customer (KYC) Implementation with Smart Contracts on a Privacy-Oriented Decentralized Architecture. *Future Internet*, *12*(2), 41. https://doi.org/10.3390/fi12020041

Karandikar, N., Chakravorty, A., & Rong, C. (2021). Blockchain Based Transaction System with Fungible and Non-Fungible Tokens for a Community-Based Energy Infrastructure. *Sensors (Basel, Switzerland)*, *21*(11). https://doi.org/10.3390/s21113822

Kirli, D., Couraud, B., Robu, V., Salgado-Bravo, M., Norbu, S., Andoni, M., . . . Kiprakis, A. (2022). Smart contracts in energy systems: A systematic review of fundamental approaches

and implementations. *Renewable and Sustainable Energy Reviews*, *158*, 112013. https://doi.org/10.1016/j.rser.2021.112013

Kugler, L. (2021). Non-fungible tokens and the future of art. *Communications of the ACM*, *64*(9), 19–20. https://doi.org/10.1145/3474355

Kumar, M. A., Radhesyam, V., & SrinivasaRao, B. (2019, January 10–11). Front-End IoT Application for the Bitcoin based on Proof of Elapsed Time (PoET). In *2019 Third International Conference on Inventive Systems and Control (ICISC)* (pp. 646–649). IEEE. https://doi.org/10.1109/ICISC44355.2019.9036391

Lee, W.-M. (2019). *Beginning Ethereum smart contracts programming: With examples in Python, Solidity, and JavaScript*. New York: Apress. Retrieved from https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5892707

Leitold, H., & Konrad, D. (2018). Erfahrungen aus eIDAS Zertifizierung. *Datenschutz Und Datensicherheit - DuD*, *42*(7), 429–433. https://doi.org/10.1007/s11623-018-0973-6

Mazur, M. (2021). Non-Fungible Tokens (NFT). The Analysis of Risk and Return. *SSRN Electronic Journal.* Advance online publication. https://doi.org/10.2139/ssrn.3953535

Meinel, C., Gayvoronskaya, T., & Schnjakin, M. (2018). *Blockchain: Hype oder Innovation*. *Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam: Vol. 113*. Potsdam: Universitätsverlag Potsdam.

Meitinger, T. H. (2017). Smart Contracts. *Informatik-Spektrum*, *40*(4), 371–375. https://doi.org/10.1007/s00287-017-1045-2

Microsoft (2023). Office Produktvergleich. Retrieved from https://www.microsoft.com/de-at/microsoft-365/buy/compare-all-microsoft-365-products

Mihus, I. (2022). Evolution of practical use of blockchain technologies by companies. *ECONOMICS, FINANCE and MANAGEMENT REVIEW*. (1), 42–50. https://doi.org/10.36690/2674-5208-2022-1-42

Negara, E., Hidayanto, A., Andryani, R., & Syaputra, R. (2021). Survey of Smart Contract Framework and Its Application. *Information*, *12*(7), 257. https://doi.org/10.3390/info12070257

Neugebauer, R. (Ed.) (2018). *Digitalisierung* (1. Aufl. 2018). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1614614

Oesterreich.gv.at-Redaktion (2022, April 6). Arbeitsverhältnis. Retrieved from https://www.oesterreich.gv.at/lexicon/A/Seite.991389.html

Pinto-Gutiérrez, C., Gaitán, S., Jaramillo, D., & Velasquez, S. (2022). The NFT Hype: What Draws Attention to Non-Fungible Tokens? *Mathematics*, *10*(3), 335. https://doi.org/10.3390/math10030335

Quintela-del-Río, A., & Francisco-Fernández, M. (2017). Excel Templates: A Helpful Tool for Teaching Statistics. *The American Statistician*, *71*(4), 317–325. https://doi.org/10.1080/00031305.2016.1186115

Ra, G., Seo, D., Bhuiyan, M. Z. A., & Lee, I. (2020). An Anonymous Protocol with User Identification and Linking Capabilities for User Privacy in a Permissioned Blockchain. *Electronics*, *9*(8), 1183. https://doi.org/10.3390/electronics9081183

Risius, M., & Spohrer, K. (2017). A Blockchain Research Framework. *Business & Information Systems Engineering*, *59*(6), 385–409. https://doi.org/10.1007/s12599-017-0506-0

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8–13. https://doi.org/10.1145/1764810.1764814

Singh, J., & Singh, P. (2021). Distributed Ownership Model for Non-Fungible Tokens. In N. Gupta, P. Chatterjee, & T. Choudhury (Eds.), *Smart and Sustainable Intelligent Systems* (pp. 307–321). Wiley. https://doi.org/10.1002/9781119752134.ch22

Solingen, R., & Berghout, E. (1999). The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development.

Treleaven, P., Gendal Brown, R., & Yang, D. (2017). Blockchain Technology in Finance. *Computer*, *50*(9), 14–17. https://doi.org/10.1109/MC.2017.3571047

Vacca, A., Di Sorbo, A., Visaggio, C. A., & Canfora, G. (2021). A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges. *Journal of Systems and Software*, *174*, 110891. https://doi.org/10.1016/j.jss.2020.110891

Valeonti, F., Bikakis, A., Terras, M., Speed, C., Hudson-Smith, A., & Chalkias, K. (2021). Crypto Collectibles, Museum Funding and OpenGLAM: Challenges, Opportunities and the Potential of Non-Fungible Tokens (NFTs). *Applied Sciences*, *11*(21), 9931. https://doi.org/10.3390/app11219931

Vonhoegen, H. (2019). *Excel 2019: Der umfassende Ratgeber* (1st ed.). Bonn: Rheinwerk Verlag. Retrieved from https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6383093

Waldegge, S. H. von (2018). Steigerung der Effizienz im Vertragsmanagement. In A. Khare, D. Kessler, & J. Wirsam (Eds.), *Marktorientiertes Produkt- und Produktionsmanagement in digitalen Umwelten* (pp. 85–100). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-21637-5_7

Wirth, C., & Kolain, M. (2018). *Privacy by BlockChain Design: A BlockChain-enabled GDPR-compliant Approach for Handling Personal Data*. European Society for Socially Embedded Technologies (EUSSET). https://doi.org/10.18420/blockchain2018_03

WKO (2019, November 21). Personal-Unterlagen: rechtskonform aufbewahren und Fristen kennen. Retrieved from https://www.wko.at/branchen/vbg/information-consulting/unternehmensberatung-buchhaltung-informationstechnologie/personal-unterlagen.html

Wohrer, M., & Zdun, U. (2021, December 6–8). DevOps for Ethereum Blockchain Smart Contracts. In *2021 IEEE International Conference on Blockchain (Blockchain)* (pp. 244–251). IEEE. https://doi.org/10.1109/Blockchain53845.2021.00040

Yang, R., Wakefield, R., Lyu, S., Jayasuriya, S., Han, F., Yi, X., . . . Chen, S. (2020). Public and private blockchain in construction business process and information integration. *Automation in Construction*, *118*, 103276. https://doi.org/10.1016/j.autcon.2020.103276

Yang, W., Garg, S., Raza, A., Herbert, D., & Kang, B. (2018). Blockchain: Trends and Future. In K. Yoshida & M. Lee (Eds.), *Lecture Notes in Computer Science. Knowledge Management and Acquisition for Intelligent Systems* (Vol. 11016, pp. 201–210). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-97289-3_15