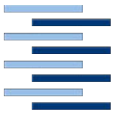

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterthesis

Maximilian Stahl

Entwicklung und Implementierung einer blockchainbasierten Sachbilanz für die Ökobilanzierung von Waren und Dienstleistungen

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Maximilian Stahl

Entwicklung und Implementierung einer blockchainbasierten Sachbilanz für die Ökobilanzierung von Waren und Dienstleistungen

Masterthesis eingereicht im Rahmen Masterprüfung
im Studiengang Informations- und Kommunikationstechnik (M.Sc.)
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüferin: Prof. Dr. Heike Neumann
Zweitgutachterin: Dr.-Ing. Ilka Gehrke
Abgegeben am 16.09.2020

Maximilian Stahl

Thema der Masterthesis

Entwicklung und Implementierung einer blockchainbasierten Sachbilanz für die Ökobilanzierung von Waren und Dienstleistungen

Stichworte

Sachbilanz, Ökobilanz, Distributed Ledger Technology, Blockchain, Ethereum, Interplanetary Filesystem

Kurzzusammenfassung

Die Sammlung von Daten für die Berechnung einer Sachbilanz, welche dazu verwendet wird die Ökobilanz oder den ökologischen Fußabdruck eines Produkts oder einer Dienstleistung zu berechnen, weist Schwächen auf. Darunter zählen das verwenden einer Sachbilanzdatenbank für Prozesse, die von Zulieferern abgewickelt werden und für die die Firmen keine Daten vorliegen haben. Diese Datenbanken haben oft nur veraltete und durchschnittliche Werte vorliegen. Außerdem berechnet jede Firma die Ökobilanz ihrer Produkte in einer anderen Weise. Somit entstehen Ungenauigkeiten und die Ökobilanz zweier ähnlicher Produkte ist nicht ohne Voreingenommenheit zu berechnen. Um dieses Problem zu lösen wird ein Proof of Concept zur blockchainbasierten Datensammlung für die Erstellung einer Sachbilanz, welcher höhere Transparenz und Reproduzierbarkeit bringen soll, vorgeschlagen, implementiert, jedoch nicht vollautomatisiert und dessen Stärken und Schwächen diskutiert. Es wurde zuerst ein Modell erstellt. Anschließend wurde zur Umsetzung die Smart Contract Plattform Ethereum und das verteilte Peer-to-Peer (P2P) Dateispeichersystem Interplanetary Filesystem (IPFS) verwendet. Die Tests zeigen, wie eine blockchainbasierte Datensammlung für eine Sachbilanz umgesetzt werden könnte, jedoch gibt es Limitierungen, an die das Modell stößt. Durch die momentanen Skalierungsprobleme von Blockchains, zu denen es jedoch einige Lösungen gibt, lohnt es sich wegen den hohen Ausführungsgebühren im Moment nicht Rechnungen direkt auf der Blockchain auszuführen. Außerdem würde dies auch zu einem Problem der Privatsphäre und Geheimhaltung führen, zu welchen es jedoch schon Ansätze wie Homomorphe Verschlüsselung für Smart Contracts gibt. Die Zwischenspeicherung der Daten von der Blockchain in eine Datenbank macht die Daten jedoch wieder manipulierbar, wenn diese nicht durch zusätzliche Sicherheitsmaßnahmen gesichert sind. Darüber hinaus konnte die Implementierung nicht mit realen Daten und in einer realen Umgebung getestet werden. In diesem Fall würden sich Fragen zur Internetverfügbarkeit, Sensorausstattung entlang der Lieferkette, Kommunikation zwischen Herstellern und Lieferanden, Geheimhaltung, zu Anreizen, zur zirkulären Wirtschaft und gesetzlichen Regulierungen stellen. Dies zeigt, dass sich dieses Thema am Übergang von der Wissenschaft in die Technik befindet und noch weitere Studien durchgeführt werden müssten, um einen Ansatz zu finden, der für alle Teilnehmenden zufriedenstellend ist.

Maximilian Stahl

Title of the paper

Development and implementation of a blockchain-based life cycle inventory for the life cycle assessment of goods and services

Keywords

Life Cycle Inventory, Life Cycle Assessment, Distributed Ledger Technology, Blockchain, Ethereum, Interplanetary Filesystem

Abstract

The collection of data for the calculation of a Life Cycle Inventory (LCI), which is used to calculate the Life Cycle Assessment (LCA) or ecological footprint of a product or service, has its weaknesses. These include the use of a LCI database for processes that are carried out by suppliers and for which the companies do not have data available. These databases often only have outdated and average values. In addition, each company calculates the LCA of its products in a different way. This creates inaccuracies and the LCA of two similar products cannot be calculated without bias. In order to solve this problem, a proof of concept for blockchain-based data collection for the preparation of a LCI, which should bring higher transparency and reproducibility, is proposed, implemented, however not fully automated and its strengths and weaknesses are discussed. First a model was created. Subsequently, the smart contract platform Ethereum and the distributed peer-to-peer file storage system Interplanetary Filesystem (IPFS) were used for implementation. The tests show how a blockchain-based data collection for a LCI could be implemented, but there are limitations that the model encounters. Due to the current scaling problems of blockchains, for which there are some solutions, it is currently not worthwhile to execute calculations directly on the blockchain due to the high execution fees. This would also lead to a problem of privacy and secrecy, for which there are already approaches like homomorphic encryption for smart contracts. However, the intermediate storage of data from the blockchain into a database makes it possible to manipulate the data again if the database is not secured by additional security measures. Furthermore, the implementation could not be tested with real data and in a real environment. In this case, questions would arise about Internet availability, sensor equipment along the supply chain, communication between manufacturers and suppliers, confidentiality, incentives, circular economy, and legal regulations. This shows that this topic is at the transition from science to technology and that further studies would need to be conducted to find an approach that is satisfactory for all participants.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In den letzten Jahren wurden die Auswirkungen des menschenverursachten Klimawandels immer spürbarer, da die Häufigkeit von Dürreperioden und extremen Wetterverhältnissen zugenommen hat. Viele junge Menschen gehen auf die Straßen, um die Politik auf diese Problematik aufmerksam zu machen, da ihre Zukunft auf dem Spiel steht. Seit der Zeit der Industrialisierung verbraucht die Menschheit immer mehr Ressourcen und es wird oft nicht nachhaltig mit ihnen umgegangen (Bruno Oberle et al. 2019) .

Supply Chain Management (SCM) beschäftigt sich mit dem Fluss von Dienstleistungen und Materialien, die zur Herstellung eines bestimmten Produkts benötigt werden. Dies umfasst üblicherweise Zwischenlagerungen und Produktionszyklen bis zur Lieferung zum Endverbraucher. Da Lieferketten heutzutage oftmals eine globale Skalierung haben und viele verschiedene Firmen miteinander interagieren, ist das Management sehr komplex, wobei die Kosten für die Verwaltung des Inventars und Prozess- und Fehlererkennung besonders hoch sind (Wüst und Gervais 2018).

Bei der Herstellung und Lieferung von Produkten werden viel Wasser und Energie verbraucht und CO_2 ausgestoßen. Das Einsparpotential durch die Optimierung der Lieferkette kann deshalb einen erheblichen Beitrag zum Klimaschutz leisten (Kashmanian und Moore 2014). Hierbei kommt die Ökobilanz (LCA: Life Cycle Analysis) ins Spiel, die Umweltemissionen und Verbrauch natürlicher Ressourcen, welche entlang ganzer Lieferketten entstehen, analysiert. Richtig angewandt, erlaubt die Ökobilanz einen Vergleich des ökologischen Fußabdrucks von unterschiedlichen Produkten, die dieselbe Funktion haben und ist deshalb ein hilfreiches Werkzeug im Zusammenhang mit nachhaltigem Konsum und Produktion (Wolf et al. 2012).

Nach Hellweg und Milà i Canals (2014) besteht die Ökobilanz aus vier Schritten, die iterativ durchgeführt werden. Zuerst wird das Ziel gesetzt, das festlegt, welche Fragen

mit dem Aufstellen der LCA beantwortet werden sollen (1). Danach wird die Sachbilanz - im englischen Life Cycle Inventory (LCI) - aufgestellt, welche eine Datenbank darstellt und den gesamten Energie- und Rohstoffverbrauch, den ein Produkt oder eine Dienstleistung in seiner Lebenszeit hat, aufzeichnet (2). Dies dient zur Grundlage des dritten Schrittes, welcher die Auswirkungen des Produktes oder der Dienstleistung auf die Umwelt berechnet (3). Der vierte Schritt ist die Interpretation und Reiteration (4). Es werden unterschiedliche Schritte durchgeführt, die z.B. die Konsistenz und Vollständigkeit des gewählten Modells prüfen. Außerdem werden verschiedene Analysen veranlasst, die z.B. die Sensitivität einiger Parameter untersuchen. Das bedeutet, dass festgestellt wird, wie groß die Auswirkungen bestimmter Eingangsparameter auf das Ergebnis sind. Zuletzt wird noch die Unsicherheit der gewählten Parameter analysiert. Abhängig vom Ergebnis des vierten Schrittes werden Änderungen im Ziel vorgenommen, worauf der Prozess wieder von vorne beginnt.

Durch die Möglichkeit den Ressourcenfluss entlang der Lieferkette zu erfassen wird die Ökobilanz als ein fundamentales, unterstützendes Element bei der Einführung von kreislaufwirtschaftlichen Richtlinien betrachtet (Haupt und Zschokke 2017). In der Kreislaufwirtschaft wird im Gegensatz zur Linearwirtschaft, die den „Wiege zur Urne“ Ansatz verfolgt, von einem „Wiege zur Wiege“ Ansatz gesprochen. Der Unterschied der Ansätze beläuft sich darauf, dass die Kreislaufwirtschaft versucht den Lebenszyklus eines Produkts so lange wie möglich zu verlängern. Produkte werden hergestellt, dann benutzt, danach so oft wie möglich repariert; anschließend werden die Rohstoffe für das Renovieren eines anderen Produkts verwendet und zum Schluss, wenn die Rohstoffe nicht mehr zur Wiederverwendung geeignet sind, z.B. durch Verbrennung in Energie umgewandelt (Stahel 2016).

Nach Pizzol et al. (2015) könnte das Ergebnis der Ökobilanz mit Ansätzen der Umweltökonomie kombiniert werden, um dadurch die Umweltkosten in die Produkte mit einzuberechnen, wie z.B. eine CO_2-Steuer. Um dies jedoch durchzusetzen, müsste die Ökobilanz eines Produkts sehr präzise sein, da die Auswirkungen der Steuern enorm sein können. Allerdings ist die Qualität der Ökobilanz auf die der Datensätze für die Sachbilanz angewiesen, welche oft limitiert ist (Edelen und Ingwersen 2018). Eine weitere mögliche Verwendung der Ökobilanz ist eine Kennzeichnung von Produkten,

die somit die Wahl von umweltbewussten Verbrauchern beeinflusst (Kareiva et al. 2015) oder als Grundlage für ein System dient, welches Produkte, die nicht zirkulär sind, höher besteuern könnte (OECD 2018).

Sachbilanzen enthalten eine Liste der Material- und Energieflüsse, die in einem bestimmten Produkt entstehen und liefern den Eingangswert für den Schritt der Wirkungsabschätzung. Derzeit werden Daten aus Sachbilanz-Datenbanken, z.B. EcoInvent, oftmals aus Sekundärquellen bezogen, welche aus Studien oder Literatur stammen, was zu Qualitätsproblemen führen kann, wenn diese Daten gemittelt und nicht häufig aktualisiert werden (Kuczenski et al. 2018). Dagegen ist es wünschenswert vor allem Primärdaten zu verwenden, die direkt von Firmen gesammelt werden. Fehlende Transparenz in den Sachbilanzdatenbanken kann dazu führen, dass dasselbe Produktsystem von unterschiedlichen Ökobilanz-Praktikern anders modelliert wird, was zu einer Inkonsistenz führen und somit den Vergleich von unterschiedlichen Ökobilanzstudien verhindern kann (Kuczenski et al. 2018). Sachbilanzen tendieren daher dazu durchschnittliche Werte für Material- und Energieflüsse für Produktklassen, anstatt Werte für spezifische Produkte, anzugeben, wie in Abbildung 1 zu sehen ist.

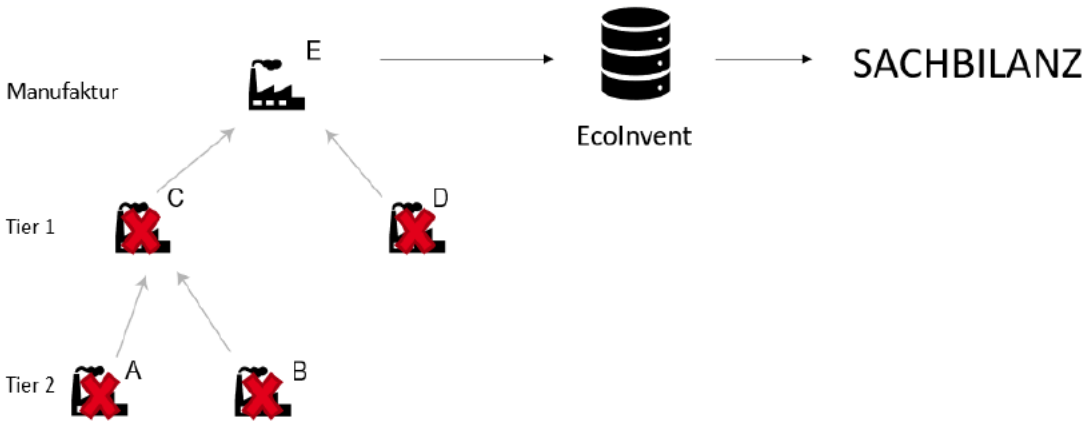

Abbildung 1: Aktuelles Modell zur Sammlung von Daten für eine Sachbilanz. Firma E möchte die Ökobilanz eines ihrer Produkte berechnen. Dazu werden die Primärdaten gesammelt, die sie selbst in der Firma erheben können, um eine Sachbilanz zu erstellen, jedoch stehen ihnen die Daten von ihren Zulieferern (Firmen A, B, C, D) nicht zur Verfügung. Deshalb müssen diese Daten aus einer Datenbank wie EcoInvent entnommen werden. Diese Daten stellen den Industriedurchschnitt einer Kategorie dar und sind demnach oft ungenau und können durch unregelmäßige

Aktualisierung veraltet sein. Zusätzlich werden Lizenzgebühren für den Zugriff auf die Datenbank verlangt.

Eine Methode die Sachbilanzdaten zu sammeln, könnte auf Basis der Blockchain sein, welche eine Datenstruktur innerhalb eines dezentralen Systems ist. Diese erlaubt es Daten in einer vertrauenswürdigen, transparenten und sicheren Weise zu erfassen. Laut Leible et al. (2019) ist die Technologie erst ca. 10 Jahre in Verwendung und entwickelt sich, stetig fort. Seitdem vor ca. 5 Jahren Plattformen, die Smart Contract-fähig sind veröffentlicht wurden, passiert vor allem in den Sektoren Finanzwesen, Lieferkettenmanagement, Versicherungen u.ä. sehr viel. Firmen wie Ernst & Young, Microsoft und andere bilden Zusammenschlüsse mit dem Ziel, die Adaption der Distibuted Ledger Technology (DLT), welche in Kapitel 2.3 detailliert erklärt wird, zu beschleunigen (Samuel Haig 2020). Die Blockchain wird sogar verwendet, um bei der Situation des neuartigen COVID-19 Virus die Verteilung von Medikamenten an die örtliche Apotheke oder vor die Haustüre der Patienten zu gewährleisten und bei Krankheit oder Todesfällen Versicherungsansprüche geltend zu machen (Ting et al. 2020).

Ein weiteres Anwendungsgebiet ist das Verwenden der DLT in der Lieferkette, wie es z.B. Daimler oder IBM und Walmart vormachen (Mercedes-Benz 2020; Kamath 2018). Daimler fokusiert sich darauf die Transparenz beim CO_2-Ausstoß in der Kobalt-Lieferkette zu erhöhen und bei Walmart im Zusammenschuss mit IBM wird es dazu verwendet für eine höhere Lebensmittelsicherheit bei Lieferketten zu sorgen. Ein weiterer Vorteil ist die Ermöglichung einer lückenlosen Aufzeichnung von Material- und Geldflüssen, die sich aus der Sicht des finanziellen Aspekts des zugrundeliegenden Konsensalgorithmus nicht lohnen würde, zu manipulieren. Dies bringt die Möglichkeit zu einer umfassenden Erstellung einer Sachbilanz, welche die Genauigkeit der Ökobilanz positiv beeinflusst.

Zielstellung

Basierend darauf ist das Ziel der Thesis einen konzeptionellen Ansatz zu finden, der die Eigenschaften von Blockchains nutzt, um den Prozess der Datensammlung für Sachbilanzen und damit die Unterscheidung von Produkten mithilfe von Ökobilanzen zu verbessern.

Es wird die Annahme getroffen, dass ein Großteil der Daten, die zur Aufstellung einer Sachbilanz eines Produktes benötigt werden, aus Datenflüssen wie Lagerscheinen oder Lieferscheinen entnommen werden können. Dies ist sicherlich bei Wirtschaftsströmen der Fall, welche die Flüsse der Güter und Dienstleistungen sind, die von verschiedenen Prozessen in einem Produktsystem ausgetauscht werden. Dabei ergibt sich die Problematik, dass diese Daten über mehrere Knoten in einer Lieferkette aufgeteilt sind. Dies führt dazu, dass die meisten dieser Daten nicht zugänglich sind und diese während der Erstellung einer Ökobilanz hergeleitet werden müssen, was zu Qualitätsproblemen führt (Hellweg und Milà i Canals 2014). Was wäre, wenn diese Daten verfügbar wären?

Um diese Frage zu beantworten, wird von einem Szenario ausgegangen, in dem sich alle Vorteile der blockchainbasierten Unternehmensplanung in den Alltag eingefunden haben. Materialflüsse werden durch Transaktionen aufgenommen, in denen Informationen über die Güter oder die Dienstleistung erfasst werden. Es soll beantwortet werden, ob es möglich ist, einen blockchainbasierten Ansatz für die Datensammlung zur Erstellung einer Sachbilanz zu verwenden und was die praktischen Umsetzungsprobleme eines solchen Systems sind.
Um die Hypothese zu veri- oder falsifizieren wird

i. ein Modell zur blockchainbasierten Datensammlung erstellt.
ii. das Modell mithilfe der Smart Contract Plattform Ethereum umgesetzt.
iii. das umgesetzte Modell auf einem Testnetzwerk überprüft.

Dies wird es erlauben zu testen, ob ein blockchainbasierter Ansatz verwendet werden kann, um die Datensammlung einer Sachbilanz zu optimieren. Im Fall, dass dies gelingt, können Vor- und Nachteile, sowie eventuelle technische Probleme debattiert

werden. Im Fall, dass es nicht gelingt oder nur teilweise funktioniert, können die Probleme oder die Schritte, die nicht funktioniert haben, erläutert und auf mögliche Lösungen für die Zukunft spekuliert werden.

Die Arbeit wurde in Kooperation mit dem Fraunhofer UMSICHT (Umwelt-, Sicherheits- und Energietechnik) in Oberhausen und der Hochschule für Angewandte Wissenschaften Hamburg geschrieben.

2 Technische Grundlagen

Zuerst wird eine Einführung in die Ökobilanz gegeben und danach werden Methoden zur Berechnung einer Sachbilanz vorgestellt. Eine vereinfachte, nicht vollautomatisierte Berechnung der Sachbilanz wurde nur zu Zwecken der Veranschaulichung durchgeführt. Anschließend wird auf die Distributed Ledger Technologie, Blockchain und das verwendete System Ethereum und die Software eingegangen, die verwendet wurde.

2.1 Ökobilanz ISO 14044

Nach Hauschild et al. (2018) zeigt die Ökobilanz die Auswirkungen eines Produkts oder einer Dienstleistung auf, welche in dessen Lebenszeit auf die Umwelt ausgeübt werden. Die Erstellung stellt einen iterativen Prozess dar, welcher nach ISO 14040-14044 in vier Schritte unterteilt wird, wie es in Abbildung 2 veranschaulicht wird.

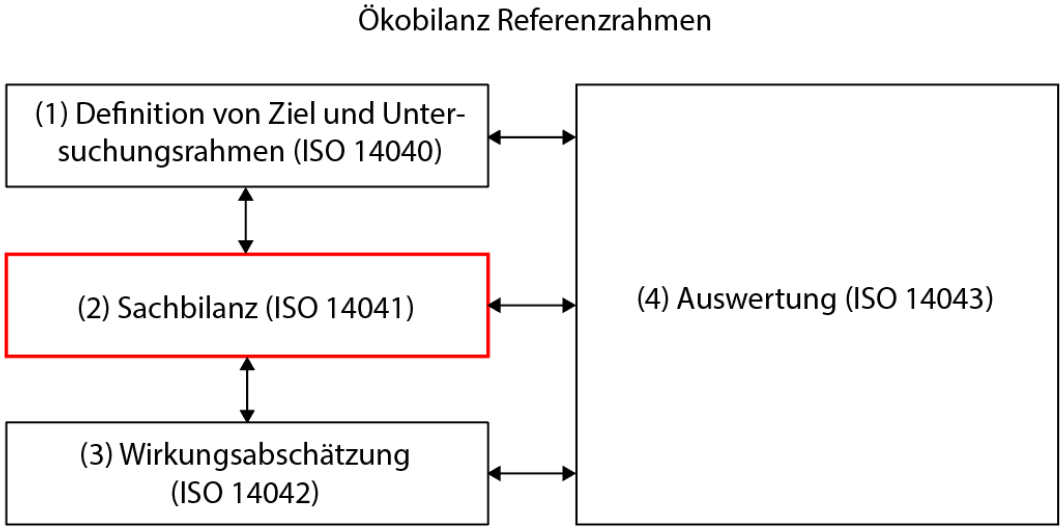

Abbildung 2: Referenzrahmen der Ökobilanz. Die Ökobilanz teilt sich in die vier Schritte (1) der Definition von Ziel und Untersuchungsrahmen; (2) des Aufstellens der Sachbilanz; (3) der Wirkungsabschätzung und (4) der Auswertung zusammen. Die Ökobilanz stellt einen iterativen Prozess dar, der dieselbe immer wieder versucht zu verbessern.

Definition von Ziel und Untersuchungsrahmen (ISO 14040)

Der erste Schritt in der Erstellung einer Ökobilanz ist das Bestimmen der Ziele, welche mit dem Aufstellen der Ökobilanz erreicht werden sollen und in welchem Umfang dies stattfinden soll. Es beginnt mit der Bestimmung der Funktionellen Einheit, welche eine quantitative Beschreibung der Dienstleistung oder Funktion ist, für die die Bewertung ausgeübt wird (Hauschild et al. 2018). Außerdem ist dies die Grundlage für das Festlegen des Referenzflusses für das Produkt, welcher aussagt, wie viel des Produkts benötigt wird, um die Funktionellen Einheit zu realisieren. Überdies muss der Umfang des Produktsystems festgelegt werden, in dem entschieden wird, welche Vorgänge und Prozesse zu dem Lebenszyklus des untersuchten Produkts gehören. Zudem spielt bei der Interpretation der Endergebnisse das anfänglich festgelegte Ziel und der Umfang eine wichtige Rolle, da diese die Systemmodellierung und die Art und Weise wie die Daten erhoben werden, beeinflussen.

Sachbilanz (ISO 14041)

Der Prozess des Erstellens der Sachbilanz benötigt die meisten zeitlichen und monetären Mittel, wobei das Sammeln der Daten zu ihrer Erstellung der Schwerpunkt dieser Thesis ist, wie Abbildung 3 zeigt. Die Hauptaufgabe ist das Sammeln und Kompilieren von Daten aus Elementarflüssen, welche Rohstoffe sind, die von der Umwelt in das Produktsystem und wieder zurückfließen, aber auch Zwischenflüssen, welche von einem Unit Prozess zum anderen stattfindet. Das Ergebnis aus dieser Phase der Ökobilanz wird als Grundlage zur Wirkungsabschätzung herangezogen (Hauschild et al. 2018). Die kleinste Einheit von Eingangsdaten in einer Sachbilanz, welche in Materialien, Energie und Ressourcen unterschieden werden, sowie die Ausgangsdaten, welche in Produkte, zu behandelnder Abfall und Emissionen eingeteilt werden, stellen Unit Prozesse dar (Hauschild et al. 2018). Die Datenqualität wird in zwei Arten unterschieden: Primärdaten, welche produktspezifisch sind, da die Daten selbst aus Sensoren erhoben werden und Sekundärdaten, welche den Industriedurchschnitt darstellen, weil sie aus Studien und Datenbanken entnommen werden (Hauschild et al. 2018).

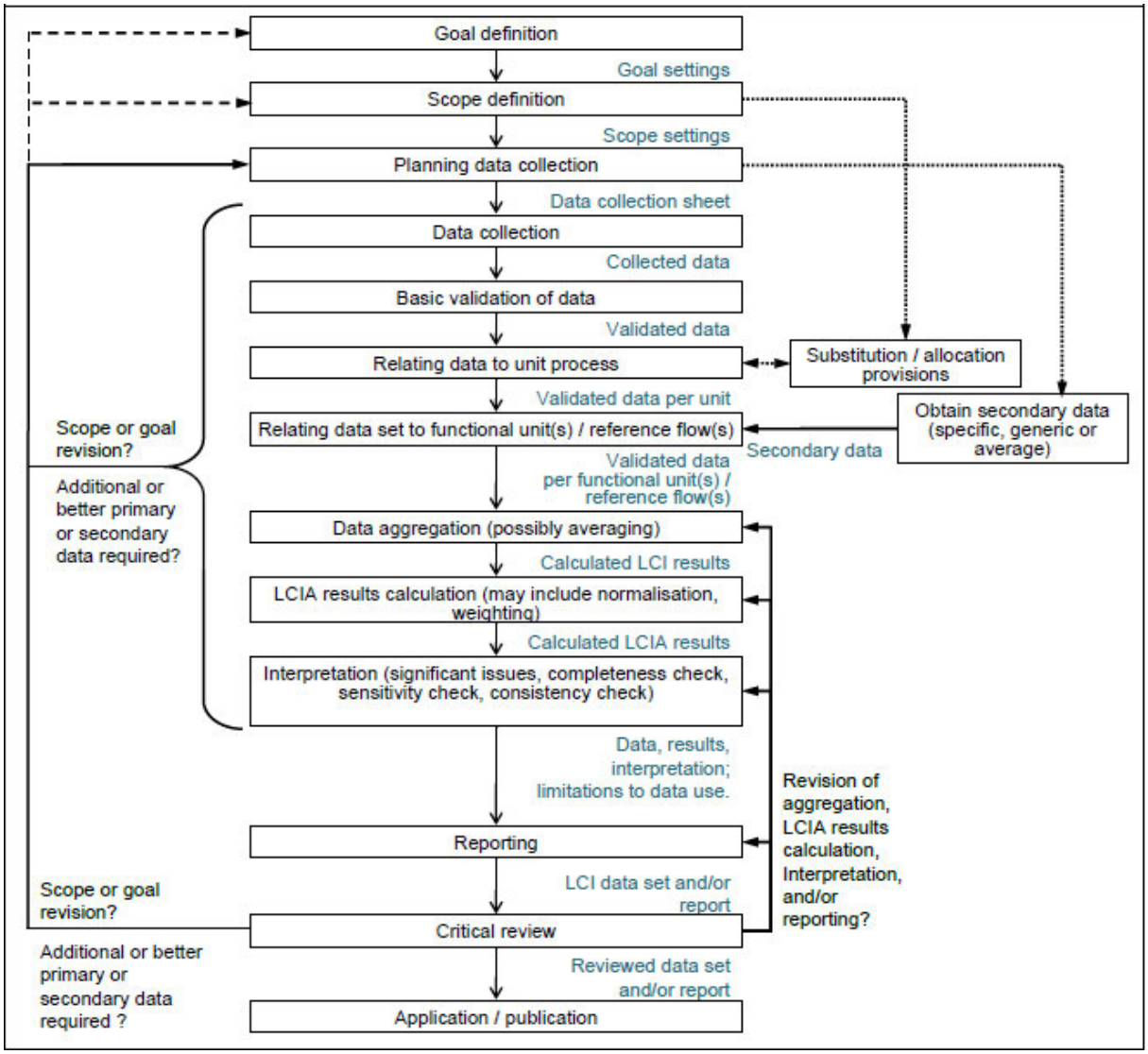

Abbildung 3: Auf diesem Bild wird der komplette Ablauf des Erstellens einer Ökobilanz und der dafür benötigten Sachbilanz dargestellt. Der Teil mit dem sich die Thesis beschäftigt ist nur der der Datensammlung (Data Collection), wobei von den Schritten vorher (Planning data collection) und nachher (Basic Validation) angenommen wird, dass diese von anderen Parteien übernommen werden und nicht im Fokus der Thesis ist (Joint Research Center (JRC) 2010)**.**

Wirkungsabschätzung (ISO 14042)

Im nächsten Schritt wird die Wirkungsabschätzung festgestellt. Nach Hauschild et al. (2018) wird diese im Allgemeinen in fünf Schritte aufgeteilt, wobei nur die ersten drei namentlich die Auswahl der Methode, die Klassifizierung und Charakterisierung verpflichtend sind.

Auswertung (ISO 14043)

In dieser Phase werden die Ergebnisse der LCA durch unterschiedliche Prüfungen interpretiert, um damit eine Verbesserung des Modells zu erreichen. Es wird in Verfahrensschritte, welche zeigen, wie vollständig und konsistent die Ökobilanz ist und in numerische Schritte, die angeben wie robust die Ökobilanz ist, unterteilt.

2.2 Methoden zur Berechnung einer Sachbilanz

Nach Lenzen und Crawford (2009) gibt es verschiedene Methoden die Sachbilanz einer Ökobilanz zu berechnen, welche sich in der Genauigkeit und der Vollständigkeit in der Systembetrachtung unterscheiden. Die drei Hauptmethoden sind eine prozessbasierte Sachbilanz, welche genau ist, der jedoch die Vollständigkeit fehlt, eine Ein- und Ausgangssachbilanz, welche Vollständigkeit aufweist, in der jedoch die Daten ungenau sein können und eine hybrid Sachbilanz, die genau und vollständig ist, jedoch Daten doppelt gezählt werden können und eine höhere mathematische Komplexität aufzeigt. Um den passenden Ansatz für das jeweilige Vorhaben zu wählen, müssen der Zweck, der Umfang, die verfügbare Zeit, die auszuführende Arbeit und die finanziellen Mittel, die zur Verfügung stehen, betrachtet werden. Die Ein- und Ausgangssachbilanz arbeitet nach dem Leontiefschen Modell, in dem die Verhältnisse unterschiedlicher Sektoren zueinander aufgezeigt werden. Dies ergibt eine relativ ungenaue Sachbilanz und die Aktualisierung der Datensätze kann mehrere Jahre dauern. Für das Modell, welches in dieser Arbeit entwickelt wurde, würde sich eine prozessbasierte Sachbilanz anbieten, da es mit ihr möglich ist, mehrere Ein- und Ausgänge eines Prozesses zu betrachten und dies benötigt wird.

Die folgenden Berechnungen, die Matrizen verwenden, basieren auf Suh und Huppes (2010) und Islam et al. (2016). Die Matrix $\boldsymbol{A}$ ist als $n \times n$ Matrix definiert, bei der ein Element a_{ij} die Zu-und Abflusse von Ware i des Prozesses j während der Betriebszeit aufzeigt, wobei Zuflüsse positive und Abflüsse negative Werte besitzen.

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad (1)$$

Die Matrix ***B*** ist als *p*×*n* Matrix definiert, bei der ein Element b_i die Menge an Schad- oder Rohstoffen i, die vom Prozess j während der Betriebszeit ausgestoßen oder verbraucht wird, aufzeigt.

$$B = [b_1 \quad \cdots \quad b_n] \tag{2}$$

Der Vektor ***k*** stellt die Funktionseinheit des Systems dar, welche in dem später gewählten Beispiel die Menge an CO_2 ist, welche beim Herstellen von einem Toaster entsteht.

$$k = \begin{bmatrix} k_1 \\ \vdots \\ k_n \end{bmatrix} \tag{3}$$

M stellt die gesamte direkte und indirekte Umweltinterventionsmatrix dar. Gleichzeitig ist dies die Gleichung, die verwendet wird, um das Ergebnis einer Sachbilanz eines bestimmten Produkts mit einer bestimmten Funktionseinheit zu berechnen.

$$M = BA^{-1}k \tag{4}$$

2.3 Distributed Ledger Systeme & Blockchain

Das Problem der byzantischen Generäle wurde von Lamport et al. (1982) beschrieben und befasst sich damit, wie mehrere Parteien in einem dezentralen Netzwerk zu einem Konsens kommen können, bevor eine Entscheidung getroffen wird. Das Szenario stellt dar, dass Generäle mit ihren jeweiligen Soldaten, um eine Stadt, die sie einnehmen wollen, mit einem Abstand von mehreren Kilometern verteilt sind. Eine koordinierte Attacke von allen Armeen zur gleichen Zeit führt zu einem Sieg. Wenn jedoch die Armeen in unkoordinierter Weise angreifen, werden diese von der Armee der Stadt geschlagen. Die Voraussetzung ist, dass die Generäle nur mit Boten kommunizieren dürfen und sich dadurch auf einen gemeinsamen Plan einigen. Wenn ein General z.B. einen Boten an alle anderen Generäle schicken würde, kann der General weder sicher sein, dass alle anderen Generäle mit guten Absichten handeln, keine Verräter sind und/oder den Boten töten würden, noch kann er nicht sicher sein, ob er überhaupt Antwort zurückerhalten wird. Wie können die anderen Generäle sicher sein, dass der Bote, der zu ihnen kommt, die Wahrheit spricht? Es wird angenommen, dass sich alle loyalen Generäle für denselben Plan entscheiden und ein Verräter die loyalen Generäle nicht überzeugen kann seinem Plan zu folgen. Zur Lösung des Problems kommt die Distributed Ledger Technologie mit einem Konsensmechanismus ins Spiel.

Alle Belagerer aus dem vorherigen Szenario werden als Computer in einem Netzwerk dargestellt, wobei die Generäle eine Kopie eines Computers abbilden, der ein Logbuch führt (Distributed Ledger). Bevor ein neuer Eintrag in diesem Logbuch eingetragen werden kann, müssen alle Teilnehmer im Netzwerk zu einem Konsens kommen, erst dann kann ihr Logbuch aktualisiert werden. Damit dies in einer Reihenfolge stattfindet, die nicht verändert werden kann, wird eine Blockchain verwendet. Dadurch, dass die Teilnehmer sich nicht vertrauen können und sie trotzdem in gemeinschaftlicher Weise, dezentral miteinander arbeiten und einen Konsens erreichen wollen, müssen sie ein Konsensprotokoll anwenden, von denen in dem Unterkapitel des Referenzmodells eines genannt wird (Conte de Leon et al. 2017). Es ist möglich das Problem nur durch das Verwenden von Boten zu lösen, solange zwei Drittel der Geräle ehrlich handeln. Zunächst wird jedoch noch auf die Eigenschaften einer Blockchain eingegangen.

Eine Blockchain ist ein Mechanismus, mit dem digitale Daten in Form eines Logbuches gespeichert werden können. Ein Block besteht aus einer Anzahl von Daten, wie z.B. Transaktionen. Dieser Block wird als Eingangswert in eine Hashfunktion gegeben und bildet somit den Blockhash. Innerhalb des Blocks befindet sich der Blockhash des vorherigen Blocks, wie es in Abbildung 5 dargestellt wird (Conte de Leon et al. 2017). Es gibt außerdem verschiedene Arten von Blockchains, die hauptsächlich in öffentliche (permissionless) und private (permissioned) unterschieden werden.

Permissionless Blockchains

Diese Art von Blockchain ist offen und dezentralisiert. Jeder kann dem Netzwerk als Lesender oder Schreibender beitreten oder es verlassen, wann er möchte. Außerdem existiert keine zentrale Einheit, die Teilnehmer aus dem Netzwerk, die Inhalte unrechtmäßig schreiben oder lesen, ausschließen könnte. Dies bedeutet, dass alle geschriebenen Daten auf der Blockchain für jeden lesbar sind. Beispiele für diese Art von Blockchain sind Ethereum und Bitcoin (Wüst und Gervais 2018).

Permissioned Blockchains

Es werden nur einer limitierten Anzahl von Lesenden und Schreibenden der Zugang zu der Blockchain gewährt. Im Gegensatz zur permissionless Blockchain gibt es eine zentrale Einheit, die entscheidet, wer schreiben und lesen darf. Ein Beispiel dafür ist Hyperledger Fabric (Wüst und Gervais 2018).

2.3.1 Referenzmodell der Blockchain Technologie

Laut Yuan und Wang (2018) wird für die Blockchain Technologie das folgende Referenzmodell, Abbildung 4 , vorgeschlagen, aus dem Bestandteile der einzelnen Ebenen genannt werden.

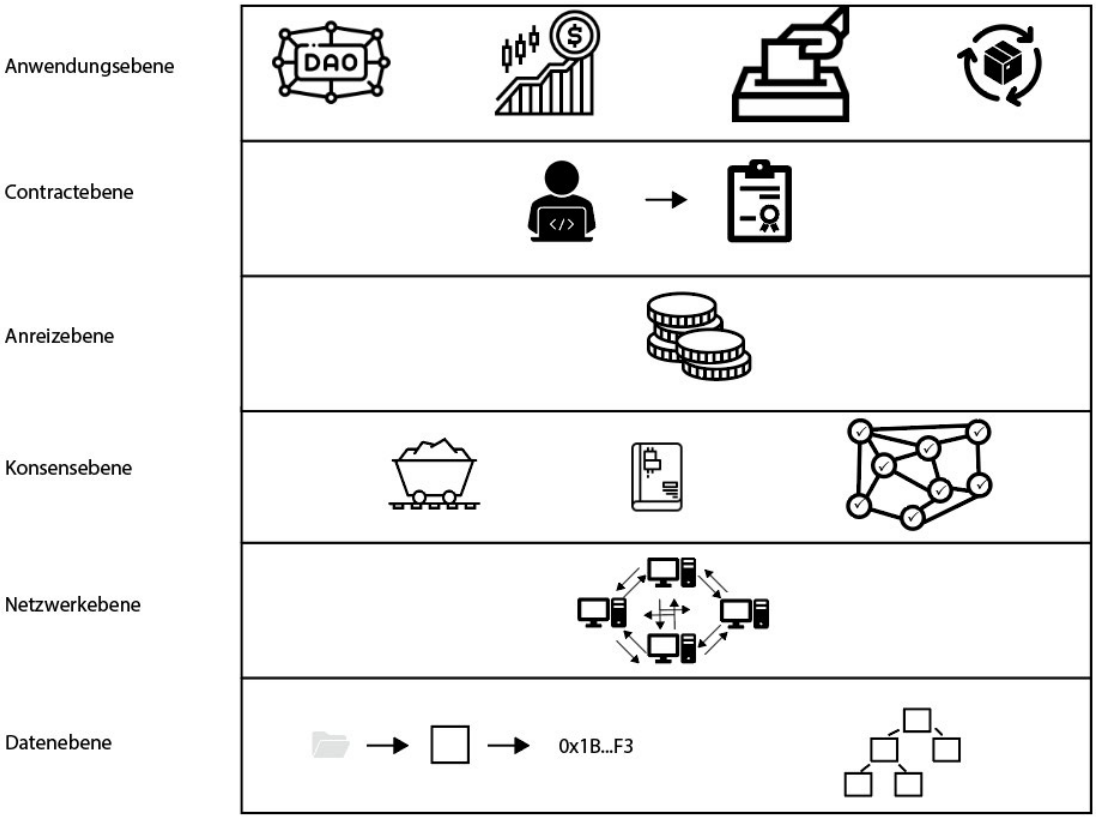

Abbildung 4: Ein Referenzmodell für die Blockchain Technologie eingeteilt in sechs Ebenen. In der Datenebene befinden sich alle Werkzeuge, wie z.B. die Hashfunktion und der Merkle Tree, die zur Realisierung einer Blockchain benötigt werden. In der Netzwerkebene befindet sich das P2P Netzwerk, in dem alle Teilnehmenden gleichgestellt sind. In der Konsensebene wird durch Konsensmechanismen wie Proof of Work sichergestellt, dass unter den Teilnehmern im Netzwerk ein Konsens herrscht und so das Logbuch für alle gleich ist. In der Anreizebene wird durch Belohnungen ein Anreiz geschaffen sich ehrlich zu verhalten. In der Contractebene können sogenannte Smart Contracts programmiert werden, die bei Interaktion mit deren Funktionen den festgelegten Code ausführen. In der Anwendungsebene können verschiedene Anwendungen ihren Nutzen finden. Beispiele sind eine dezentrale autonome Organisation, decentralized Finance, Governance und Supply Chain.

Datenebene

In dieser Ebene werden alle Werkzeuge verwendet, die zur Realisierung des Blockchainmechanismus benötigt werden.

Hashfunktion

Es gibt viele unterschiedliche Hashfunktionen, jedoch wird im Blockchainbereich vor allem der Secure Hash Algorithm (SHA-256) verwendet, der laut Selvakumar und Ganadhas (2009) wie alle Hashfunktionen folgende Eigenschaften aufweisen muss:

- Determiniertheit, was bedeutet, dass die Funktion für einen Eingabewert immer denselben Ausgabewert ausgibt.
- Es ist rechnerisch nicht realisierbar den Eingangswert herauszufinden.
- Kollisionsresistenz, da es genauso rechnerisch nicht realisierbar ist zwei Eingangswerte zu finden, die denselben Ausgangswert ergeben.
- Bei der Änderung von nur einem Bit muss sich der Ausgangswert signifikant ändern, der sog. Lawineneffekt.

Merkle Tree

Laut Merkle (1988) ist ein Merkle Tree eine Datenstruktur aus Hashes, die aus Blättern, Ästen und einer Wurzel besteht. Wie in Abbildung 5 zu sehen ist, fängt dieser Baum bei den Blättern an, welche die Ausgangswerte von Daten, die in eine Hashfunktion übergeben werden, darstellen. Zwei der Blätter werden anschließend durch hashen in einen Ast zusammengefasst. Dies kann sich mehrere Male fortsetzen bis zur letzten Ebene, in der die Äste zu einer Wurzel durch hashen zusammengefasst werden. Diese Wurzel wird dann in den Block mit der Blockhash des vorherigen Blocks und anderen Werten wiederum gehasht und in den darauffolgenden Block übergeben.

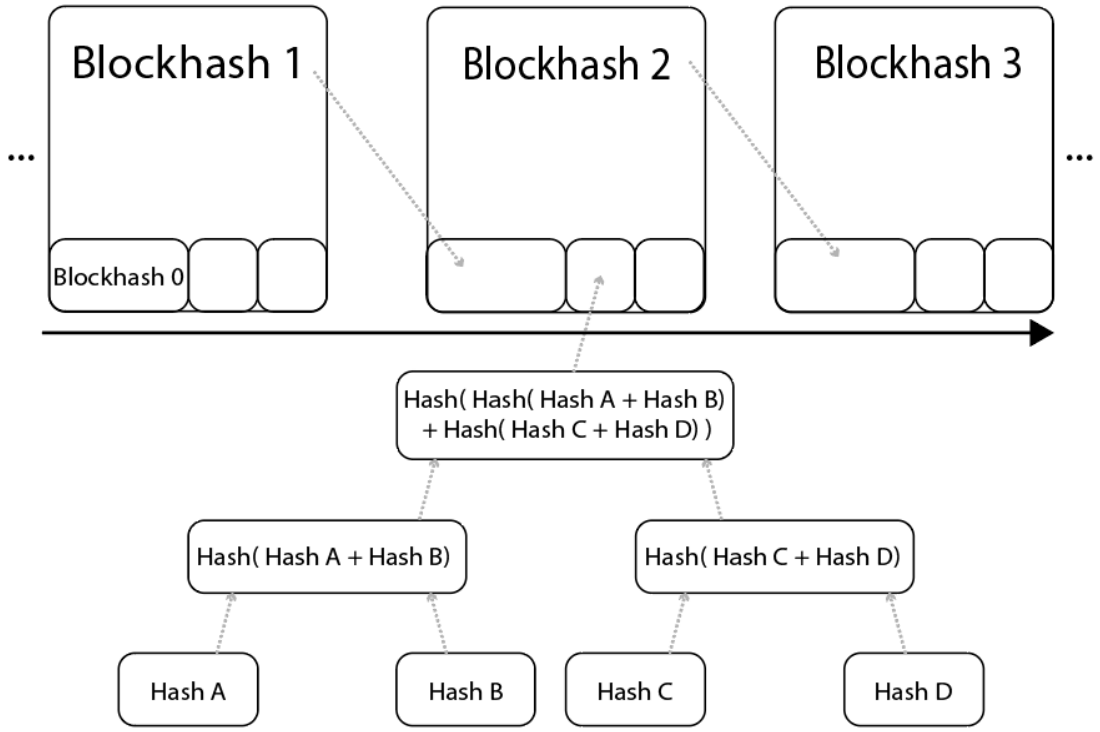

Abbildung 5: Merkle Tree und Blockchainmechanismus. Aus den untersten Hashes (Hash A - D), die Transaktionen darstellen könnten, wird wieder aus zweien ein Hash gebildet. Dies setzt sich weiter bis zur Merkle Root fort, welche hier in den Block 2 gespeichert wird. Um den Blockhash 2 zu erhalten, wird der Blockhash des vorherigen Blocks, die Merkle Root und andere Parameter wie einem Zeitstempel verwendet. Somit verändern sich bei einer Modifizierung eines beliebigen vorherigen Blocks alle Hashwerte der folgenden Blöcke.

Netzwerkebene

Diese Ebene spezifiziert die dezentrale Natur der Distributed Ledger Technologie. Dies bedeutet, dass das Netzwerk aus einer großen Anzahl an Geräten besteht, die miteinander als P2P verbunden sind, bei denen alle Teilnehmer gleiche Berechtigungen haben.

Konsensebene

Ein Konsensmechanismus besteht aus mehreren Schritten. Zuerst werden im Block Proposal neue Blöcke erstellt und ein Beweis für die Erstellung des Blocks angehängt, um Sybil-Attacken zu verhindern, welche eine Attacke speziell bei P2P-Netzwerken

darstellt und die darauf abzielt falsche Identitäten zu erstellen, mit dem Ziel, die Mehrheit der Stimmen im Netzwerk zu erreichen (Xiao et al. 2020).

Im Schritt der Informationspropagation wird ausgehend vom Erzeuger des Blocks, der neue Block mit den enthaltenen Transaktionen z.B. durch den sogenannten Gossiping Mechanismus, welcher in Abbildung 6 dargestellt wird, im Netzwerk verbreitet (Xiao et al. 2020). Jedoch kann es auch passieren, dass zwei gültige Blöcke von zwei Erzeugern zur gleichen Zeit gefunden werden. Wegen diesem Fall existiert der GHOST-Mechanismus, der entscheidet welcher dieser Blöcke gültig ist.

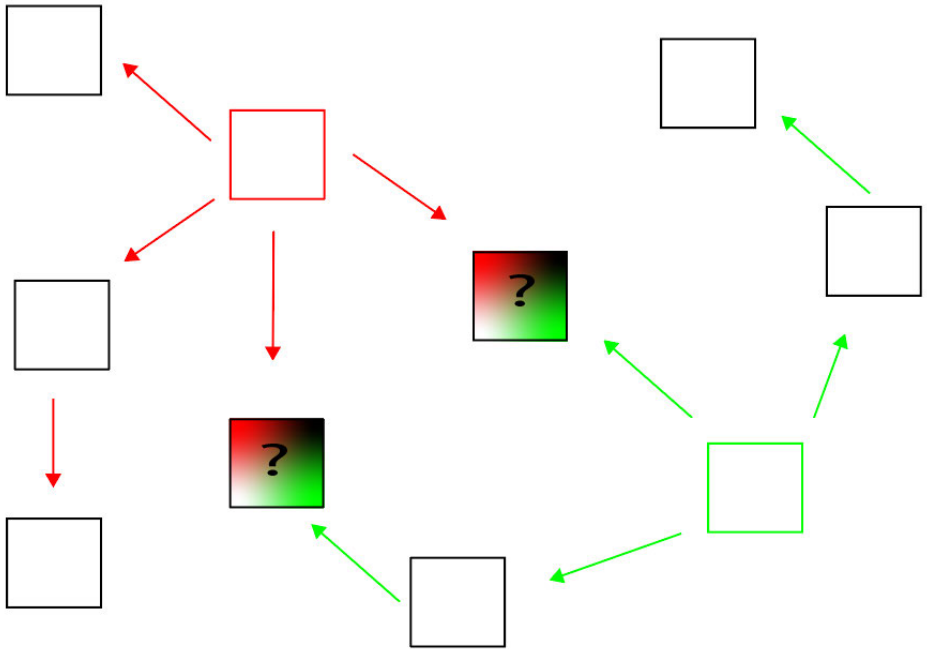

Abbildung 6: Der Gossiping Mechanismus. Wenn ein Miner, welcher einen Betreiber eines Knotens in dem Netzwerk darstellt, einen neuen Block vorgeschlagen hat, dann wird diese Information von ihm an die nächsten Knoten im Netzwerk verschickt, worauf diese die Information auch wieder weitergeben. Dabei kann es auch dazu kommen, dass zwei Miner (Rot und Grün) zur selben Zeit einen gültigen Block vorgeschlagen haben. Um diese Situation zu klären wird der GHOST-Mechanismus angewandt.

In der Blockvalidierung wird der Beweis für die Erstellung des Blocks und die Gültigkeit der enthaltenen Transaktionen überprüft. In dem Schritt der Blockfinalisierung wird darüber abgestimmt, ob die validierten Blöcke akzeptiert werden. Dies geschieht mit einem von vielen Mechanismen. Einer davon ist der sogenannte GHOST (Greedy Heaviest Observed Subtree)-Mechanismus, der in Abbildung 7 veranschaulicht wird,

bei dem die gültige Blockchain diejenige ist, welche in der Summe die höchste aufgewandte Arbeit vollbracht hat (Xiao et al. 2020).

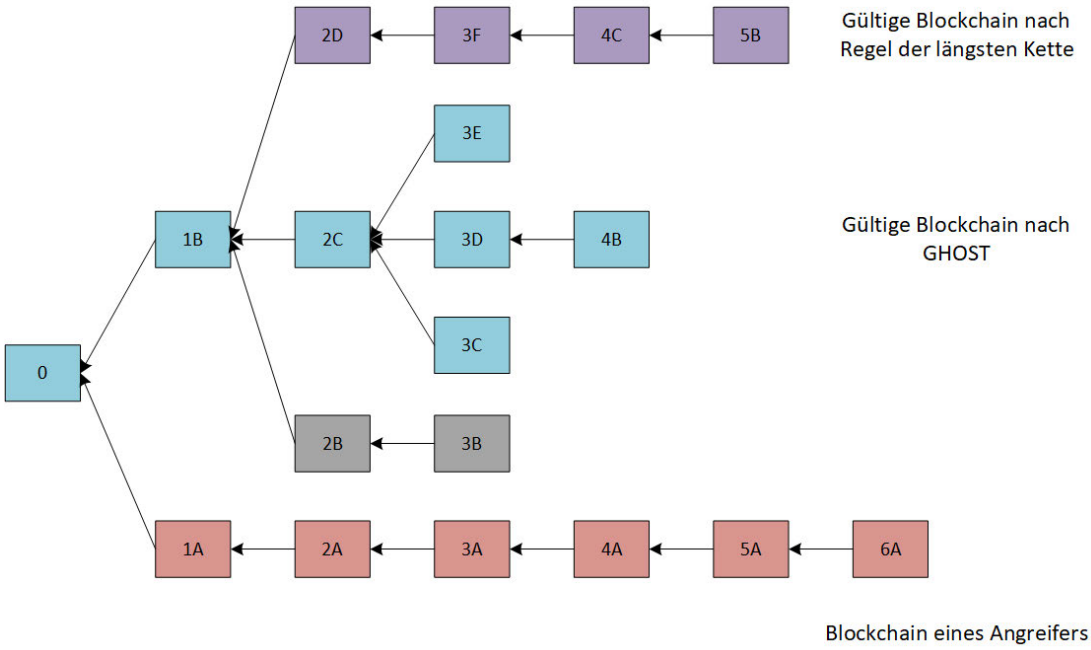

Abbildung 7: Darstellung verschiedener Finalisierungsmechanismen. Es werden der Mechanismus der längsten Blockchain, welcher durch einen Angreifer, mit einer längeren Kette ausgehebelt werden kann und der GHOST-Mechanismus, der alle Blöcke, die das Gewicht aller Blöcke die auf dem Weg zum neuen Block produziert wurden, mit einberechnet (Sompolinsky und Zohar 2015).

Nakamoto Consensus

Ein Beispiel für ein Konsensprotokoll ist der Nakamoto Konsens, der im Prinzip nach den fünf Hauptkomponenten des Gossiping, der Valdierung, der Blockfinalisierung mit der Regel der längsten Kette und dem Anreiz durch Block Belohnungen abläuft, jedoch spezifisch für den Vorgang des Block proposals den Proof of Work Mechanismus, verwendet. Dieser setzt voraus, dass für das Erstellen des nächsten Blocks das Ergebnis einer Hashfunktion gefunden werden muss, welches einen bestimmten Schwierigkeitsgrad aufweist. Dafür muss eine bestimmte Anzahl von Eingängen, die einen Ausgang einer Hashfunktion, der eine bestimmte Anzahl von führenden Nullen produziert, gefunden werden. Somit beweist aufgrund der Eigenschaften der Hashfunktion, dass eine computerbasierte Arbeit (Proof of Work) durch Ausprobieren verrichtet wurde (Xiao et al. 2020).

Anreizebene

Der Anreizmechanismus regt Teilnehmende dazu an, aufrichtig zu handeln, da sie durch das Erstellen von neuen Blöcken plus den Gebühren der enthaltenen Transaktionen in Form einer digitalen Währung entlohnt werden (Xiao et al. 2020).

Contract-Ebene

In dieser Ebene können durch Smart Contracts Vereinbarungen programmiert werden, die zuvor von mehreren Parteien festgelegt wurden. Deren Funktionen werden bei Interaktion automatisch ausgeführt und sind nicht veränderbar, nachdem sie in das Netzwerk hochgeladen wurden.

Anwendungsebene

In dieser Ebene können die tatsächlichen Anwendungen, sog. dApps (decentralized Applications) ausgeführt werden, welche in unterschiedlichen Branchen ihren Nutzen finden können, wie z.B. DeFi (Decentraliced Finance), Supply Chain oder im Gesundheitswesen.

2.3.2 Ethereum

Aufgrund von den Kriterien von Wüst und Gervais (2018) und noch detailierter in Yaga et al. (2018), die in Abbildung 8 gezeigt werden, dass in dem System, das entwickelt werden soll, ein Status gespeichert werden muss, es mehrere Schreibende gibt, kein vertrauenswürdiger Drittanbieter verwendet werden kann, der immer online ist und nicht alle Schreibenden bekannt sind, fiel die Entscheidung auf die permissionless Blockchain Ethereum. Jedoch muss dazu erwähnt werden, dass der Punkt, ob eine dritte vertrauenswürdige Partei verwendet werden kann nicht ganz eindeutig ist, da es sicherlich Cloud Computing Plattformen, wie Amazon AWS oder Google Cloud gibt, die herangezogen werden könnten. Diese entsprechen hingegen nicht den geforderten Kriterien einer transparenten Sammlung der Daten. Außerdem könnte es sein, dass eine Firma einen Anbieter dem Anderen vorzieht, jedoch das Ziel war ein Rahmenwerk

zu erstellen, dass für alle Teilnehmenden in einer Lieferkette übergreifend gilt. Zuletzt könnte es noch passieren, dass ein Server nicht erreichbar ist.

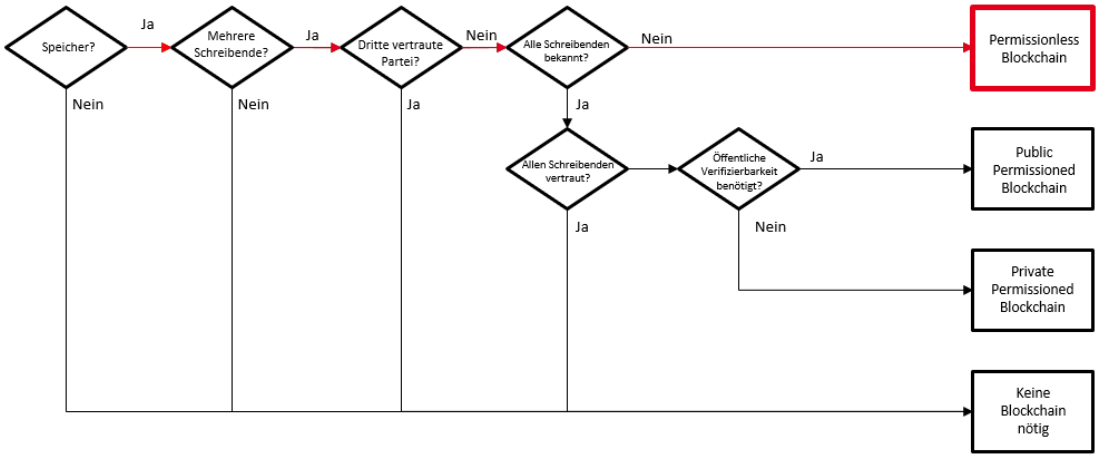

Abbildung 8: Entscheidungsgraph zur Verwendung einer Blockchain (Wüst und Gervais 2018; Yaga et al. 2018) **.**

Nach Dr. Gavin Wood (2019) ist Ethereum ein globaler Computer, eine virtuelle Maschine (VM), die auf der Ethereum Blockchain ausgeführt wird. Dieser Computer befindet sich in einem World State, welcher Adressen auf den Kontostatus abbildet. Innerhalb eines Kontostatus befinden sich Informationen über das Guthaben, den Speicher und eventuellen Code, sogenannte Smart Contracts. Wenn eine Funktion eines Smart Contracts von einer externen Adresse oder einem anderen Smart Contract aufgerufen wird, dann wird dies in einer Transaktion dokumentiert und verändert somit den World State. Da jede Transaktion von jedem Knoten im Netzwerk ausgeführt werden muss und daher die Skalierbarkeit gegenüber einem zentralisierten System stark reduziert ist, sind die Kosten um die Programme auszuführen hoch. Die resultierenden Transaktionen werden in neue Blöcke der Blockchain auf jedem Knoten im Netzwerk abzugespeichert. Dafür muss für jede Transaktion Gas in Form des ERC20-Tokens Ether bezahlt werden.

Ethereum Virtual Machine

Der virtuelle Zustandsautomat, der Ethereum Virtual Machine (EVM) genannt wird, ist eine Quasi-Turing-vollständige Maschine. Eine Turing Maschine wurde das erste Mal von Alan Turing definiert und beschreibt eine Maschine, die aus einem unendlich

langen Magnetband und einem Schreibe- und Lesekopf besteht, der auf jedes Feld des Magnetbands schreiben oder dieses Löschen kann. Diese Maschine kann eine endliche Zahl an Zuständen einnehmen. Um die Bedingungen für eine Turing-vollständige Maschine zu erfüllen, muss diese alles können was eine Turing Maschine kann. Dies wäre zum einen die Möglichkeit bedingte Verzweigungen zu erstellen wie „if und else“ Strukturen und zweitens, dass eine willkürliche Menge von Speicher zur Verfügung steht (Church und Turing 1937). Die Programmiersprache der EVM Solidity ist in der Lage bedingte Verzweigungen zu verwenden, jedoch wird durch die Rechenleistung, die die EVM verwenden kann, durch die Ausführungsgebühren (Gas) eingeschränkt und ist deshalb nur Quasi-Turing-vollständig. Hinter dem Befehlssatz, der dafür sorgt, dass der Computer bestimmte Aufgaben ausführen kann, stecken OPCODES. Diese haben eine Wortgröße von einem Byte, also können maximal 2^8=256 verschiedene Befehle ausgeführt werden. Die Programmiersprache Solidity wurde dafür entwickelt, um diese Befehle für Menschen verständlicher darzustellen, bevor der Code in den Maschinencode umgewandelt wird. Mit dieser Sprache, die sich an JavaScript, Python und C++ orientiert, können Smart Contracts geschrieben werden (Dr. Gavin Wood 2019).

Mehr zu der Syntax der Programmiersprache kann unter https://solidity.readthedocs.io/en/v0.6.6/ entnommen werden. Dort können auch die verschiedenen Versionen eingesehen werden.

2.3.3 JavaScript Module

Um mit der Blockchain zu interagieren, Daten außerhalb der Blockchain abzuspeichern und eine graphische Oberfläche zu erstellen, werden bestimmte JavaScript Module benötigt.

InterPlanetary File System (IPFS)

Das IPFS ist ein P2P Protokoll, das anstatt der standardmäßigen ortsbasierenden Adressierung zum Abrufen von gespeicherten Medien im Internet eine inhaltsbasierende Adressierung verwendet. Da das Speichern von Daten auf der Blockchain hohe Kosten verursacht, kann mithilfe von IPFS anstatt dem Inhalt selbst

nur der Link des Inhaltes, die sogenannte Content-ID (CID), auf der Blockchain gespeichert werden (Benet 2014).

IPFS verwendet eine Distributed Hash Table (DHT), welche eine Datenstruktur ist, in der Einträge, die mit einem Schlüssel verbunden sind, gespeichert werden können. Der Schlüssel ist dabei der Ausgang der Hashfunktion eines willkürlichen Datensatzes, der gespeichert werden soll. Dies findet verteilt statt, sodass jeder Knoten des P2P Netwerks einen Teil der Hash Table speichert. Es gibt außerdem ein Mapping dazu, wer welchen Teil der DHT speichert. Diese bietet den Vorteil, dass nicht jeder Knoten die ganze Hash Table speichern muss. Wenn nach einer bestimmten Datei gesucht wird, werden zuerst die Knoten gesucht, die die Datei gespeichert haben und danach wird die Datei abgerufen. Auf der Konferenz Proceeding of 18th International Conference on Computer Communications and Networks wurde ein Artikel von Nguyen et al. (2009) veröffentlicht, der zeigt, dass ein P2P Netzwerk für Video Streaming die verwendete Bandbreite um bis zu 60% verringern könnte. IPFS weist dieselbe Eigenschaft auf, da beim Herunterladen einer angeforderten Datei nicht nur ein einzelner Knoten angefragt werden muss. Überdies bietet es zusätzlich den Vorteil, dass nicht alle im Netzwerk ihren Knoten aktualisieren müssen, wenn neue Einträge hinzukommen (Benet 2014). Da IPFS mit der CID einen Hash des Inhalts angibt, ist es auch nicht möglich Duplikate von derselben Datei im Netzwerk zu erstellen. Wenn zwei gleiche Dateien hochgeladen werden, wird immer derselbe Hash zurückgegeben. Außerdem kann jeder Hash, hinter dem jeweils Daten stecken, mit einem lesbaren Namen für Menschen mithilfe des Interplanetary Name Service (IPNS) gespeichert werden.

Eine ausführliche Beschreibung von IPFS kann unter Benet (2014) entnommen werden. Zudem wird in der Abbildung 9 die Funktionsweise von IPFS stark vereinfacht dargestellt.

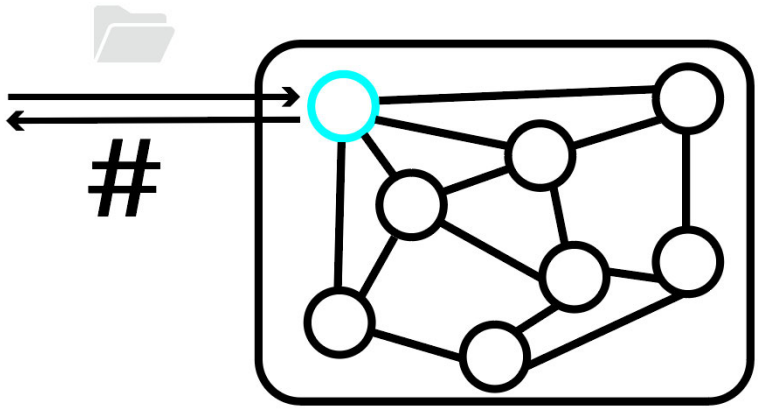

Abbildung 9: Verwendung des IPFS. Es werden Daten in das IPFS hochgeladen, und es wird eine Content-ID zurückgegeben, unter der die Daten abgerufen werden können.

Web3 Provider

Das Ethereum Netzwerk besteht aus vielen Nodes, die alle eine Kopie der Blockchain haben. Der Web3 Provider legt fest mit welchem Node kommuniziert wird. Es wäre möglich einen eigenen Node bereitzustellen, jedoch benötigt dieser je nach Art mehrere 100GB Speicher. Deshalb gibt es dafür Dienstleister, wie z. B. Infura, die sich um das aufsetzen des Nodes kümmern, von dem Nutzen gemacht wurde.

Metamask

Metamask ist eine Browsererweiterung, ein Wallet, welcher mit dem Ethereum Netzwerk kommuniziert. Dort werden die Ethereum Accounts und Privatekeys von Benutzern sicher verwaltet und es wird außerdem möglich gemacht, dass diese Accounts mit Smart Contracts interagieren können, die hinter den graphischen Oberflächen auf Webseiten oder Apps stehen.

Truffle Suite/ Buidler

Die Truffle Suite ist eine Entwicklungs- und Testumgebung zum Schreiben von Smart Contracts auf Ethereum. Es besteht aus drei Hauptkomponenten. Der Entwicklungsumgebung Truffle, in der Smart Contracts kompiliert, in das Netzwerk hochgeladen und Unit Tests geschrieben werden können. Außerdem der persönlichen Blockchain Ganache und Drizzle, um eine graphische Oberfläche darzustellen.

Buidler ist eine neuere Entwicklungs- und Testumgebung, in der mit verschiedenen Entwicklungswerkzeugen gearbeitet werden kann und auf einfache und schnelle Weise Prototypen programmiert werden können.

Web3 Connector

Web3.js sowie ethers.js sind zwei von mehreren JavaScript Bibliotheken, die dafür sorgen, dass es möglich ist mit der Ethereum Blockchain zu kommunizieren. Sie teilen

- die Adresse des Smart Contracts,
- die Funktion, die aufgerufen werden soll,
- und die Variablen, die in die Funktion übergeben werden soll mit.

Contract

Web3.js braucht zwei Parameter, um mit dem Contract zu kommunizieren. Dessen Adresse und Application Binary Interface (ABI).

Addresse

Nachdem der Smart Contract geschrieben wurde, wird er kompiliert und bereitgestellt (deployed). Nachdem der Contract deployed wurde, bekommt er eine fixe Adresse, auf der er für immer stehen wird. Diese Adresse wird ausgegeben und von nun an kann sie verwendet werden, um mit dem Contract zu kommunizieren.

Application Binary Interface (ABI)

Das ABI ist eine Darstellung der Methoden des Contracts im JSON Format und übermittelt Web3.js, wie Funktionsaufrufe zu formatieren sind, sodass der Contract dies versteht. Das ABI muss, nachdem der Contract hochgeladen wurde, auch gespeichert werden. Dies wird in Form einer separaten Datei mit einem Namen wie contractname_abi.js gespeichert.

Contract Funktionen aufrufen

In Web3.js gibt es die zwei Methoden *call()* und *send()*, um eine Funktionen des Contracts aufzurufen. Call() wird für *view* und *pure* Funktionen verwendet, welche nur auf dem lokalen Ethereum Node bzw. Infura ausgeführt werden. Es wird gelesen, aber keine Transaktion auf der Blockchain ausgeführt.
Mit send() wird eine Transaktion und eine Veränderung auf der Blockchain ausgelöst. Alle Funktionen, die nicht die Modifier *view* oder *pure* haben, müssen diese verwenden und für das Ausführen der Funktion, welche in einer Transaktion resultiert Gas zahlen. In Metamask wird sich ein Fenster öffnen, welches auffordert die Transaktion zu bestätigen.

3 Aufgabenstellung

In dieser Arbeit soll gezeigt werden, wie ein blockchainbasierter Ansatz für die Datensammlung einer Sachbilanz umgesetzt werden könnte. Das Problem der Sachbilanz heutzutage ist die Ungenauigkeit der zugrundeliegenden Daten. Jede Firma hat einen Sachbilanzersteller, von denen jeder die benötigten Ein- und Ausgangsparameter für ein Produkt anders definiert. Dadurch kommt es zu einer Inkonsistenz, wenn die Sachbilanz für dasselbe Produkt von zwei verschiedenen Sachbilanzerstellern erhoben wird. Ein weiteres Problem ist, dass Daten, die den Sachbilanzerstellern nicht zur Verfügung stehen aus Datenbanken wie EcoInvent bezogen werden, welche jedoch oft nur alte und durchschnittliche Daten für einen bestimmten Produktprozess haben. Ein möglicher Ansatz ist demnach die Verwendung der DLT/Blockchain, um die Datensammlung für Produkte zu spezifizieren. Außerdem wäre die Datensammlung konsistent und unveränderbar und würde somit in einer genaueren Sachbilanz für Produkte resultieren.

Die Anforderungen für die blockchainbasierte Datensammlung für eine Sachbilanz sind das Hochladen von Daten in IPFS, einen Smart Contract zu schreiben, der dazu in der Lage ist, Daten von unterschiedlichen Lieferanten in einer Lieferkette zu empfangen und diese Informationen durch das Auslesen eines Events zurückzuerhalten. Eine Limitierung ist unter anderem, dass kein realer Datensatz zur Verfügung steht und deshalb ein sehr vereinfachter auf Grundlage von der Herstellung eines Toasters aus (Suh und Huppes 2010) erstellt wurde. Das detaillierte Modell wird im Kapitel 4 - Konzept vorgestellt. Im Kapitel 6 - Diskussion wird noch weiter auf Limitierungen und mögliche zukünftige Lösungsansätze eingegangen. Die Tabelle 1 zeigt das Vorgehen, wie so eine Datensammlung umgesetzt werden könnte.

Tabelle 1: Beschreibung von den Aufgaben, der Umsetzung, den Details und in welchem Kapitel sich genauere Informationen dazu finden lassen.

Aufgabe	Umsetzung	Details	Kapitel
Erstellen eines Modells	Recherche von unterschiedlichen Tools und Software, um Probleme zu lösen, auf die andere Forscher gestoßen sind.	Lieferkette DLT/Blockchain Ethereum IPFS (Homomorphe Verschlüsselung)	2
	Theoretische Kombination der Tools, um ein Modell mit Adobe Illustrator zu visualisieren.	Adobe Illustrator 2020	4
Implementierung und testen des Modells	Hardware	Lenovo L450 privater Laptop mit Windows 10.	5
	Virtuelle Maschine einrichten.	Oracle VM Ubuntu 18.04 LTS	5
	Benötigte Pakete installieren	Node.js Truffle Suite/(Buidler) Web3 IPFS React MongoDB TheGraph	5
	Testdaten erstellen	JSON Herstellung eines Toasters (Energie, Metall, …)	5

	Testdaten zu IPFS senden	Zum Testen die JSON Dateien aller Teilnehmenden der Lieferkette an IPFS senden.	5
	Smart Contract schreiben	Mithilfe von Truffle Suite und Buidler Smart Conctract schreiben und zuerst auf lokale Blockchain ausführen. Danach auf öffentliches Testnetzwerk Rinkeby hochladen.	5
	Smart Contract testen	Test mit Buidler durchgeführt.	5
	Smart Contract auf Etherscan verifizieren	Dazu müssen der Code und die ABI hochgeladen werden. Somit lassen sich genauere Spezifizierungen einsehen, wie welche Funktion aufgerufen und welches Event ausgeführt wurde.	5
	Datensatz für Lieferkette erstellen	Lieferkettennummer, Chargennummer und Adressen	5
	Web3-Skript für Interaktion mit Smart Contract Funktion schreiben	Die Web3.js Bibliothek wird verwendet, um mit den Funktionen zu interagieren.	5
	Web3-Skript testen	Parameter in Funktion eingeben. Skript ausführen	5
	Automatischer Test Web3-Skript mit Smart Contract	Es wurde ein Skript mit web3 cronjob und shelljs geschrieben, welches zufällig Transaktionen zwischen 10 Lieferanden ausführt.	5

	Graphische Oberfläche zur Interaktion mit Smart Contract erstellen	Es wurde eine graphische Oberfläche mit HTML, CSS und React.js erstellt, welche es einfacher macht die JSON Testdaten in IPFS hochzuladen und danach die Funktion des Smart Contracts auszuführen.	5
	Web3-Skript, um Event von Funktion zurückzuerhalten	Wenn die Transaktion ausgeführt wurde, dann wird das Event zurückgegeben.	5
	TheGraph einrichten	Github Account mit TheGraph verbinden. Subgraph ausführen.	5
	Events in TheGraph einsehen	GraphQL Skript für TheGraph in Online Explorer erstellen und ausführen	5
	Parameter in mongoDB abspeichern	Alle notwendigen Parameter (SupplychainID (SCID), BatchID (BID), Adressen und der IPFS-Hash) abspeichern.	5
	IPFS Daten anfordern	IPFS Daten wurden heruntergeladen, jedoch kann dies je nach dem auf wie vielen Knoten sich die Daten befinden ein paar Minuten dauern.	5
	Matrix befüllen und Sachbilanz berechnen	In diesem Schritt wurde die Sachbilanz mit einer sehr vereinfachten Lieferkette aufgestellt und anschließend berechnet.	5

4 Konzept

Um die Fragestellung zu beantworten wurden zwei Konzepte entwickelt. Ein Ansatz mit einem non-fungible Token (NFT), nach dem ERC721 Standard, an das Informationen über das Produkt oder den Rohstoff angehängt werden kann. Der Hersteller müsste diese Informationen jedoch wieder auf IPFS hochladen, da der Inhalt dort im Gegensatz zu herkömmlichen URLs nicht veränderbar ist. Danach könnte mit diesen gehandelt werden und jedes Mal, wenn ein Produkt hergestellt wird, die Token, die als Rohstoffeingänge dienen für das neue Produkt „verbrannt“ werden, um ein neues Token herzustellen, welches nun die Rohstoffe miteinander kombiniert darstellt. Jedoch wurde schnell klar, dass es schwierig werden würde diese Abläufe zu verfolgen und dies nur hohe Gaskosten verursachen würde, da jedes Mal wenn ein Token erstellt wird für den Speicher, welches dieses auf der Blockchain belegt, bezahlt werden muss und dies auch keinerlei Vorteil zum Sammeln der Daten bringen würde. Deshalb wurde im zweiten Ansatz ein System entwickelt, welches es unter Verwendung eines einfachen Smart Contracts auf der Ethereum Blockchain in Kombination mit IPFS möglich macht, alle Transaktionen entlang einer Lieferkette in einer transparenten Weise, die reproduzierbar ist, darzustellen und diese zu einer Einheit, die eine unabhängige Prüfstelle für die Ökobilanz ist, sendet.

Wie Abbildung 10 zeigt wurde ein komplettes Modell erstellt, wie das System umgesetzt werden könnte. Um dieses Modell umzusetzen wurde zunächst eine Testlieferkette definiert, welche von A nach E führt. Als nächstes wurde ein Smart Contract geschrieben, der es ermöglicht eine einzigartige Identifikationsnummer je Lieferkette zu übergeben. Außerdem wird eine Quelle und ein Ziel der Transkation, eine Identifikationsnummer für jede Charge und der einzigartige Hashwert der hochgeladenen Daten von IPFS benötigt.

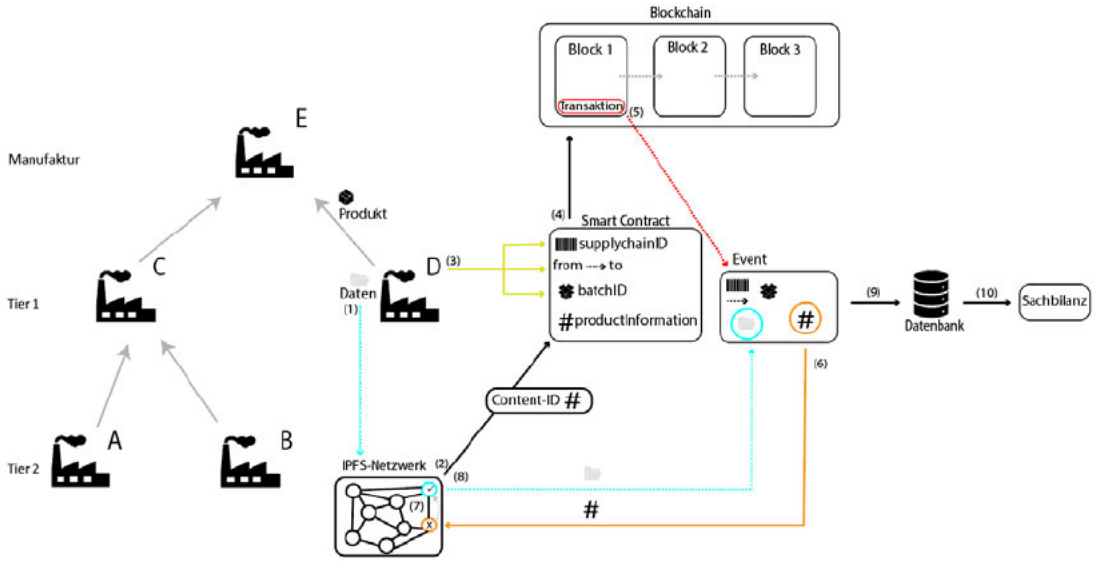

Abbildung 10: Modell blockchainbasierten Datensammlung eine Sachbilanz entlang einer vorgegebenen Lieferkette. Dieser Vorgang gilt für jeden Teilnehmer in der Lieferkette, jedoch wird dies zur Vereinfachung der Veranschaulichung nur mit Firma D aufgezeigt. (1) Umweltdaten werden von Firma D gesammelt und ins IPFS-Netzwerk hochgeladen; (2) eine Content-ID, welche die Produktinformation (#) darstellt wird an die Funktion des Smart Contracts zurückgegeben; (3) die Firma trägt alle benötigten Daten ein (supplychainID, from, to, batchID); (4) die Funktion des Smart Contracts wird ausgeführt, damit eine Transaktion in das Blockchain Netzwerk gesendet und der Block wird validiert; (5) ein Eventlistener bekommt Bescheid, wenn die Funktion des Smart Contracts aufgerufen wird und erhält alle Parameter, die in die Funktion übergeben worden sind; (6) die Content-ID wird im IPFS-Netzwerk gesucht; (7) wenn der Knoten, zu dem eine Verbindung besteht die Information nicht vorweisen kann wird der Weg zum richtige Knoten gezeigt; (8) Daten werden heruntergeladen; (9) alle Informationen können sortiert in eine unveränderbare Datenbank gespeichert werden; (10) die Sachbilanz kann berechnet werden. Nun kann E die Sachbilanz verwenden, um die Ökobilanz eines ihrer Produkte zu berechnen.

5 Implementierung & Tests

Als Umgebung wurde eine virtuelle Maschine auf einem privaten Windows 10 Laptop erstellt, welche Ubuntu 18.04 LTS simuliert. Zur Projektverwaltung wurde ein Projekt auf einem Gitlab-Server des Fraunhofer UMSICHT angelegt. Zeitgleich wurde ein Wiki-Eintrag auf dem gleichen Server angelegt, um den Projektverlauf zu dokumentieren. Alle Programme mit jeweiliger README.md mit Anweisungen zur Ausführung befinden sich auf der beigelegten DVD-R.

Das JavaScript Paket node.js wurde installiert, um Skripte außerhalb des Browsers ausführen zu können:

```
$ sudo apt install nodejs
```

```
$ npm install -g npx
```

Die Entwicklungsumgebung Truffle Suite wurde installiert, um Smart Contracts mit Solidity zu schreiben und mithilfe von Web3 mit denselben zu interagieren.

```
$ npm install -g truffle
```

Zusätzlich wurde eine Installierung von Mocha und Chai vorgenommen, um Tests mit dem Smart Contracts durchführen zu können.

```
$ npm install mocha
```

```
$ npm install chai
```

Außerdem wurde IPFS installiert.

```
$ npm install ipfs -g
```

Zudem wurde Cronjob heruntergeladen, welches erlaubt in Abständen von Minuten, Stunden, Tagen, Woche, Jahren Skripte auszuführen. Zusätzlich wurde shelljs installiert, dass es erlaubt Terminalbefehle in Skripten auszugeben.

```
$ npm install cron
```

```
$ npm install shelljs
```

Anschließend wird ein Wallet erstellt, wofür Metamask als Browsererweiterung heruntergeladen werden muss. In diesem Fall wurden Google Chrome und Brave als Browser verwendet, jedoch kann dies auch für andere Browser oder Betriebssysteme heruntergeladen werden, wie unter https://metamask.io/download.html zu sehen ist. Es muss ein Password gewählt und ein Mnemonic Seedphrase aufgeschrieben werden, aus dem die Addresse des Wallets generiert wird. Dieser besteht meistens aus 12 oder mehr Wörtern.

Da auf dem Rinkeby Testnetzwerk gearbeitet wird, müssen Testether von einer Faucet wie https://faucet.rinkeby.io/ angefordert werden, um Transaktionsgebühren bezahlen zu können. Es wird ein Truffle Projekt angelegt. Hierbei kann ein schon vorgefertigtes Projekt verwendet werden, welches schon mehrere Pakete, die benötigt werden, installiert hat. Dies kann mit dem folgenden Befehl erreicht werden, der in dem Ordner ausgeführt werden muss, in dem das Projekt abgelegt werden soll:

```
$ npx truffle unbox react
```

Dieses Projekt besteht aus mehreren Ordnern mit folgender Struktur:

- client
- contracts
- migrations
- test
- zusätzliche Dateien wie truffle-config.js und README.md

Im client Ordner befindet sich das Front-end, das mit react.js gebildet wird. In dem contracts Ordner befinden sich die Smart Contracts. Der migrations Ordner beinhaltet Dateien, die es möglich machen die Smart Contracts auf die Blockchain hochzuladen.

Im test Ordner steht Code mit chai/mocha, der dazu dient die Smart Contracts zu testen. Die truffle-config.js dient dazu den Wallet einzustellen, außerdem den Infura Node auszuwählen, mit dem der Wallet verbunden sein soll und auf welches Netzwerk der Contract hochgeladen werden soll.

Um mit dem Ethereum Netzwerk zu interagieren muss zuerst ein Konto bei infura.io eröffnet werden. Danach muss ein neues Projekt angelegt werden. Anschließend können die Endpunkte zum Datenaustausch mit dem Ethereum Node ausgewählt werden. Da dies auf dem Rinkeby Testnetzwerk stattfindet wird Rinkeby ausgewählt. Es gibt zwei unterschiedliche URLs.

- https://rinkeby.infura.io/v3/25f83305af074f0884d3c8873fa23fcd um Anfragen an den Node zu schicken und
- wss://rinkeby.infura.io/ws/v3/25f83305af074f0884d3c8873fa23fcd um Events zu lesen.

Dann kann ein Smart Contract geschrieben und mithilfe von

```
$ truffle compile
```

kompiliert, dann getestet

```
$ truffle test test_file
```

und auf das Rinkeby Testnetzwerk hochgeladen werden.

```
$ truffle migrate –network rinkeby
```

Der Smart Contract (SC) besitzt nun eine Adresse, mit welcher von nun an interagiert werden kann. Dieser SC ist unveränderbar. Wenn sich ein Fehler in dem Code befindet, kann dieser nicht geändert werden, jedoch kann der SC gelöscht werden, wenn vorher eine Funktion zur Selbstzerstörung implementiert wurde.

Durch die Verwendung von web.js kann nun mit den Funktionen des Smart Contracts interagiert werden. Außerdem geben die Events, die ausgelöst werden, wenn eine Funktion ausgeführt wird, falls sie in dem Smart Contract programmiert sind, ausgewählte Daten von der Blockchain zurück.

Eine andere Art die Events sichtbar zu machen ist unter Verwendung von TheGraph möglich. Dies kann mit den folgenden Befehlen, die im Terminal eingegeben werden müssen, installiert werden:

```
$ npm install -g @graphprotocol/graph-cli

$ graph init \

 --from-contract <CONTRACT_ADDRESS> \

 [--network <ETHEREUM_NETWORK>] \

 [--abi <FILE>] \

 <GITHUB_USER>/<SUBGRAPH_NAME> [<DIRECTORY>]
```

Für <CONTRACT_ADDRESS> wird die Adresse des Smart Contracts eingegeben. Für <ETHEREUM_NETWORK> wird der Netzwerkname des Ethereum Netzwerks eingegeben. <FILE> ist eine optionale Eingabe für die ABI des Smart Contracts. Für <GITHUB_USERNAME> wird der Benutzername von Github, und für <SUBGRAPH_NAME> der Name des Subgraphen eingegeben. Die Angabe von <DIRECTORY> ist optional, wobei bei nicht Angabe der Name des Subgraphs übernommen wird.

Zusätzlich wird noch eine Datenbank, MongoDB, verwendet, um die Daten von der Blockchain zwischenzuspeichern und danach weiterverarbeiten zu können. Um die Datenbank zu installieren müssen folgende Befehle in das Terminal eingegeben werden.

```
$ npm install mongodb
```

Danach können Skripte angelegt werden, in denen die Datenbank angelegt und beschrieben werden kann.

Zum Schluss kann die Sachbilanz eines Produkts berechnet werden, vorausgesetzt alle Daten liegen vor. Um die Berechnung durchzuführen kann ein JavaScript Skript geschrieben werden, für das das Paket math.js benötigt wird.

Somit wurden die benötigte Software und Pakete installiert und alle Einstellungen vorgenommen. Als nächstes kann der Smart Contract geschrieben und anschließend getestet werden. Danach muss der Rückempfang der Daten von der Blockchain getestet werden. Anschließend werden diese Daten zur Zwischenspeicherung in eine MongoDB geschrieben. Zuletzt kann die Sachbilanz berechnet werden. Jedoch muss beachtet werden, dass die Berechnung selbst nicht der Hauptteil der Thesis ist, dies deshalb nur stark vereinfacht und nicht vollautomatisiert durchgeführt wurde und nur um die generelle Durchführbarkeit zu demonstrieren. Die benötigten Schritte vorher (Planning Data Collection) und nachher (Data Validation etc.) wurden nicht ausgeführt. Der Grund dafür ist, dass der komplette Ablauf aller Schritte von der Sammlung der Daten bis zur Berechnung der Sachbilanz ein interdisziplinärer Vorgang ist. Dieser enthält Fragestellungen um Qualitätskontrollen, fehlende Daten bis zu rechtlichen Limitierungen und Geheimhaltung der Daten und übersteigt deshalb den Rahmen dieser Thesis.

Der Smart Contract hat lediglich zwei Funktionen. Eine (setNewTransaction) die die Lieferketten-ID, Chargen-ID, Ziel und die Produktinformation als Eingabeparameter verlangt und diese abspeichert und im Falle des Aufrufens ein Event (getNewTransaction) triggert. Die andere Funktion (getData) dient zum Auslesen der Produktinformationen, wenn die Lieferketten-ID, Quelle, Ziel und Chargen-ID übergeben werden. Zusätzlich gibt es noch ein Event, welches ausgeführt wird, wenn die erste Funktion aufgerufen wird. Zuerst wird der Smart Contract kompilliert, auf einem lokalen Testnetzwerk hochgeladen und mithilfe von Mocha und Chai getestet, was auf Abbildung 11 zu sehen ist.

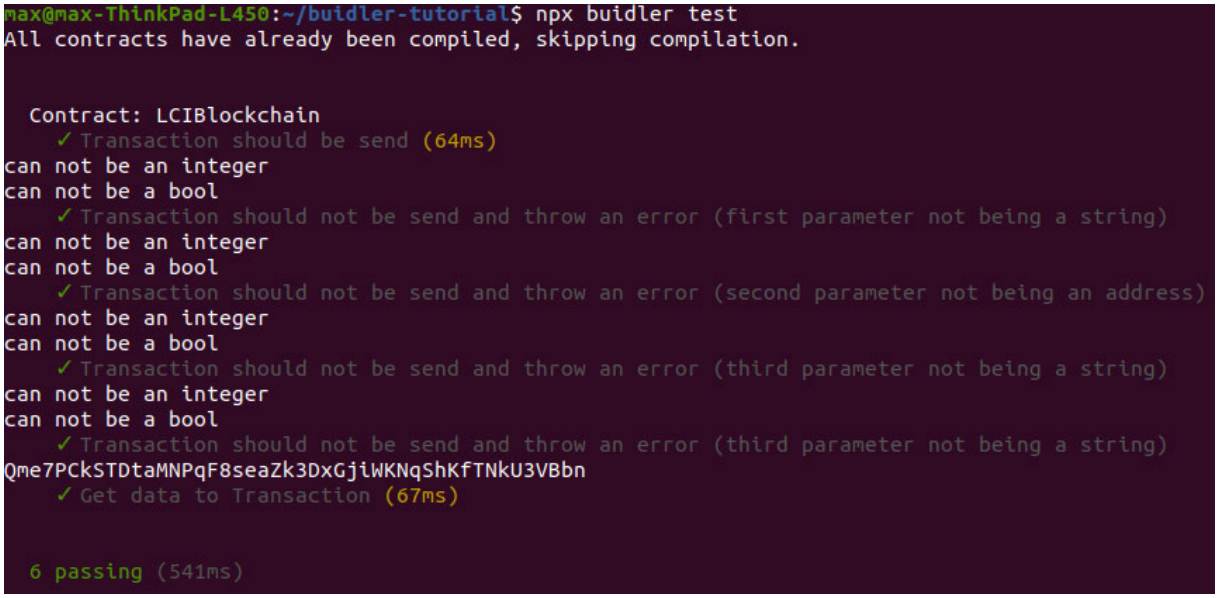

Abbildung 11: Smart Contract Tests. Alle Tests verlaufen ohne Fehler.

Anschließend wird der Smart Contract auf das Rinkeby Testnetzwerk hochgeladen, welches ein öffentliches Testnetzwerk ist. Bei der Erstellung des Smart Contracts bekommt dieser eine Adresse, 0x3d9f4199ca3960f74749ed697edfb2e79ca2206b, unter der er jetzt für immer unter *https://rinkeby.etherscan.io/* zu finden ist. Hierzu muss nur die Adresse in die Suche eingegeben werden. Es werden alle Interaktionen des Smart Contracts sichtbar, wie die Transaktionen und die ausgeführten Events. Um den Code des öffentlich sichtbar zu machen, die Events zu sehen sodass mit dem Contract interagiert werden kann, muss der Smart Contract unter https://rinkeby.etherscan.io/verifyContract verifiiziert werden. Dazu müssen die Adresse des Smart Contracts, der Compiler Typ, die Compiler Version und der Lizenztyp gewählt werden und auf „continue“ geklickt werden. Anschließend müssen der Quellcode eingefügt, für Optimization „Yes“ ausgewählt und das ABI angegeben, eine reCAPTCHA gelöst und auf „verfify and publish“ geklickt werden.

Für die Produktinformation wird eine JSON Datei angelegt, welche den wie in Abbildung 12 gezeigten Aufbau hat. SCMembers listet alle Teilnehmenden dieser Lieferkette auf. Input und Output zeigen den Ein- und Ausgang des Prozesses auf. Danach wird die Datei, die auf Abbildung 12 zu sehen ist auf IPFS hochgeladen und

der Hash QmdgYg8jJE7xE3bTje2LG63p3e5VZTioXHpQPKSUCKrXKN wird zurückgegeben.

```
{
    "SCMembers": ["0x9690e4564DE716d2c0C5ec4C66bD304ca2b9D933",
                  "0x4280c6Ef5851D6c2AA00f2344929a3AF21c05EEF",
                  "0x62851803D86dE4BE12651750604044Fc9B2A9589",
                  "0x293B3F0285F45D86E7F03dc81fc22031Ee10C838",
                  "0xBdD949173df8E7F472f51cd0cf450AB7F075A78c"
    ],
    "Input": [{
        "Dampf": -0.5,
        "SI": "MJ"
    }],
    "Output": [{
            "Stahl": 1,
            "SI": "kg"
        },
        {
            "CO2": 1,
            "SI": "kg/kg Stahl"
        }
    ]
}
```

Abbildung 12: Darstellung einer JSON-Datei, die als Produktinformation zu IPFS hochgeladen wird. Diese besteht aus den Teilnehmenden der Lieferkette, und den Eingang und Ausgang eines bestimmten Prozesses einer bestimmten Firma.

Danach wird das automatische Senden von Transaktionen von Zulieferern an Manufakteure im Abstand von 1 Minute getestet. Dazu wurde ein JavaScript Programm mit den Bibliotheken web3.js, shell.js und cron.js geschrieben. Dies Ausgabe des Skripts ist in Abbildung 13 zu sehen. Es ergibt sich ein Fehler, da die Nonce, welche Angibt wie oft mit dem Contract interagiert wurde und sogenannte Replay-Attacken verhindert, jedes Mal um eins inkrementiert wird, zu niedrig war, da beim Senden in zu kurzen Zeitabständen die vorherige Transaktion noch nicht bestätigt war.

Abbildung 13: Automatisierter Test von Smart Contract Funktion auf Rinkeby Testnetzwerk. Einige Transaktionen gelingen, aber bei anderen entsteht durch die zu niedrige Nonce ein Fehler.

Eine Replay-Attacke kann zustande kommen, wenn Alice ihre Identität gegenüber Bob beweisen möchte, in dem Fall, wenn beide Alices geheimes Password kennen. Dabei sendet Alice den Hash ihres Passwords zu Bob, welcher es daraufhin mit dem Hash vergleichen kann, den er berechnet hat. Dabei kann es vorkommen, dass Carl, der mitgehört hat, diesen Hash abfängt und sich dann bei Bob als Alice ausgeben kann. Durch das Verwenden einer Nonce kann Bob das Password verändern und an Alice zurückschicken, welche dies gegenprüft. Wenn Carl nun wieder probieren sollte die Identität von Alice vorzutäuschen, wird dies fehlschlagen, da er die Nonce nicht besitzt und sein Hash nicht mit dem von Bob übereinstimmt.

Es wird zu einem Test mit graphischer Oberfläche übergegangen, wofür ein react.js Skript erstellt wird. Obwohl es auch möglich ist direkt auf www.rinkeby.etherscan.io mit dem Smart Contract zu interagieren, ist es mit der selbst erstellten graphischen Oberfläche einfacher eine JSON Datei, welche Produktinformationen enthält, wie in Abbildung 12 dargestellt wird, einzulesen und diese danach auf IPFS hochzuladen. Es wird ein Hashwert (Content-ID) zurückgeliefert, unter dem der Inhalt ab jetzt abgerufen werden kann. Dieser setzt sich aus https://ipfs.io/ipfs/“Hashwert“ zusammen. Der Link zusammen mit Informationen wie der Lieferkettennummer (SupplychainID), dem Ziel (To) und der Chargennummer (BatchID) wird in die graphische Oberfläche eingegeben und abgesendet wie in Abbildung 14 zu sehen ist.

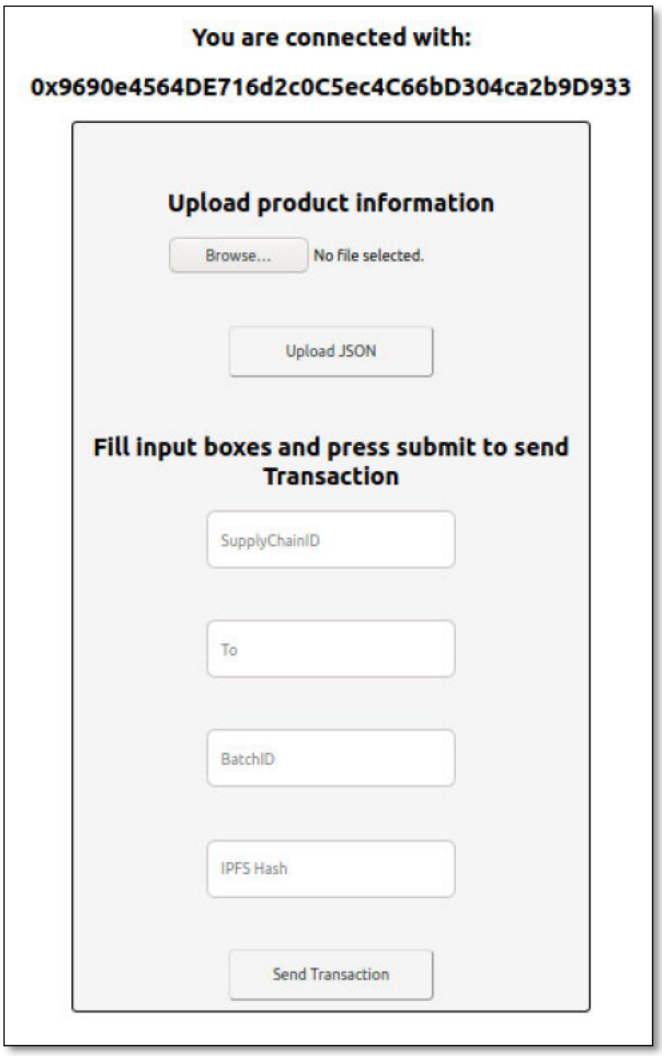

Abbildung 14: Graphische Oberfläche. Diese Benutzeroberfläche wurde erstellt, um es in einer einfachen Weise möglich zu machen, Daten auf IPFS hochzuladen und mit dem Smart Contract zu interagieren. Dazu muss zuerst mit dem *Browse* Knopf eine JSON Datei ausgewählt werden. Dann kann mit dem *Upload JSON* Knopf die Produktinformation im JSON-Format auf IPFS hochgeladen werden und es wird automatisch der Hashwert in das Feld des *IPFS Hash*/Produktinformation eingetragen. Die SupplychainID, To und BatchID werden danach eingetragen. Zum Schluss kann die Transaktion mit dem *Send Transaction* Knopf abgeschickt werden.

Nachdem die Transaktion abgeschickt wird, poppt ein Fenster von Metamask auf, in dem die Transaktion bestätigt und eine Transaktionsgebühr gezahlt werden muss wie auf Abbildung 15 zu sehen ist.

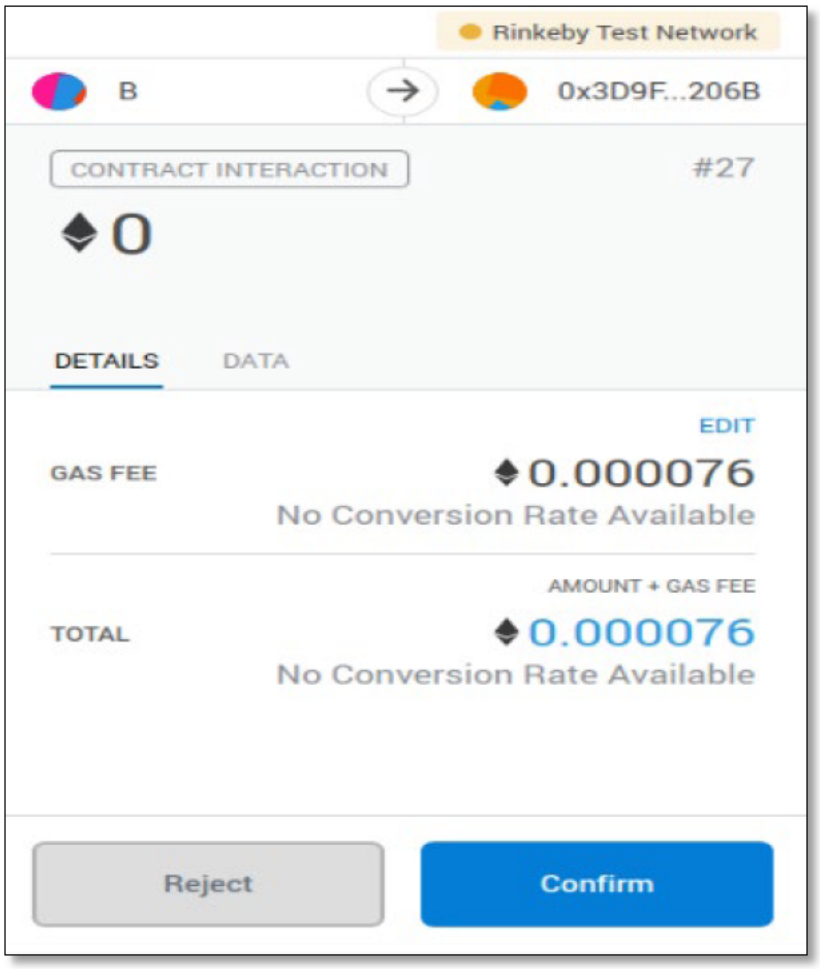

Abbildung 15: Bestätigung einer Transaktion auf dem Rinkeby Testnetzwerk. Nachdem die Transaktion bestätigt wurde, kann diese, je nachdem wie viel Gas bezahlt wurde und wie voll das Netzwerk ist, kurz danach auf rinkeby.etherscan.io angesehen werden wie bei Abbildung 16 und Abbildung 17 zu sehen ist.

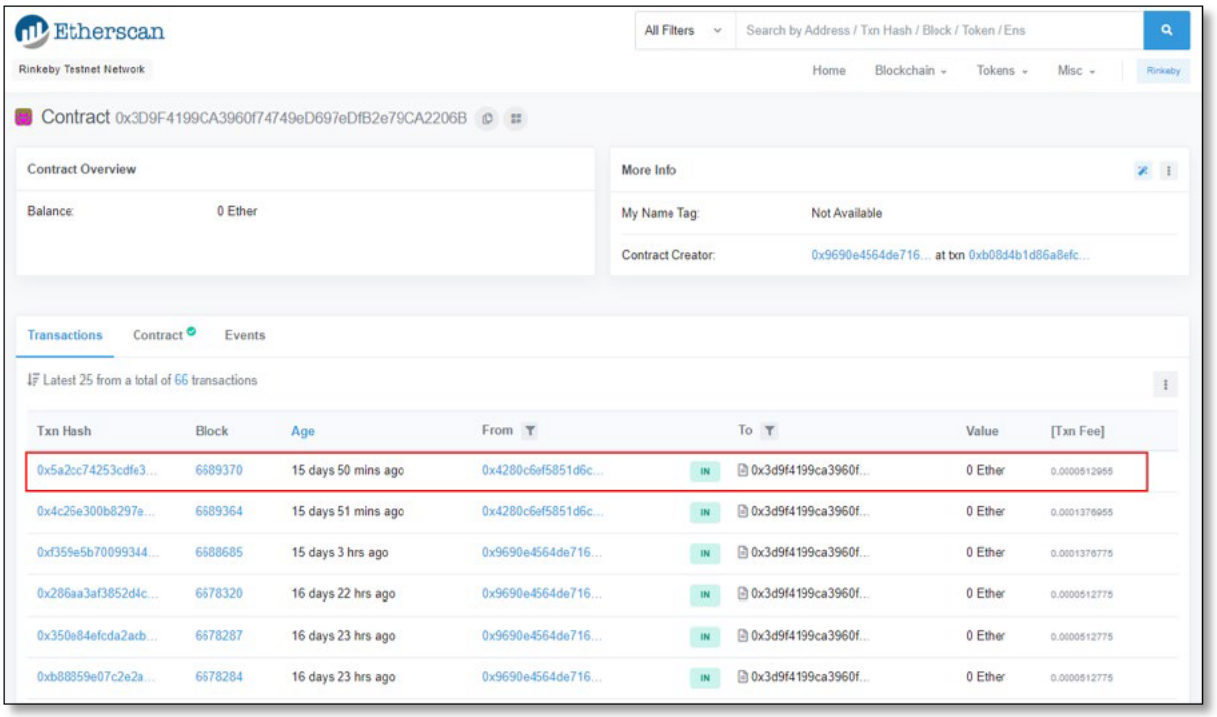

Abbildung 16: Übersicht aller Transaktionen, die mit diesem Smart Contract interagieren. Auf den die rot markierte Transaktion wird in der nächsten Abbildung weiter eingegangen.

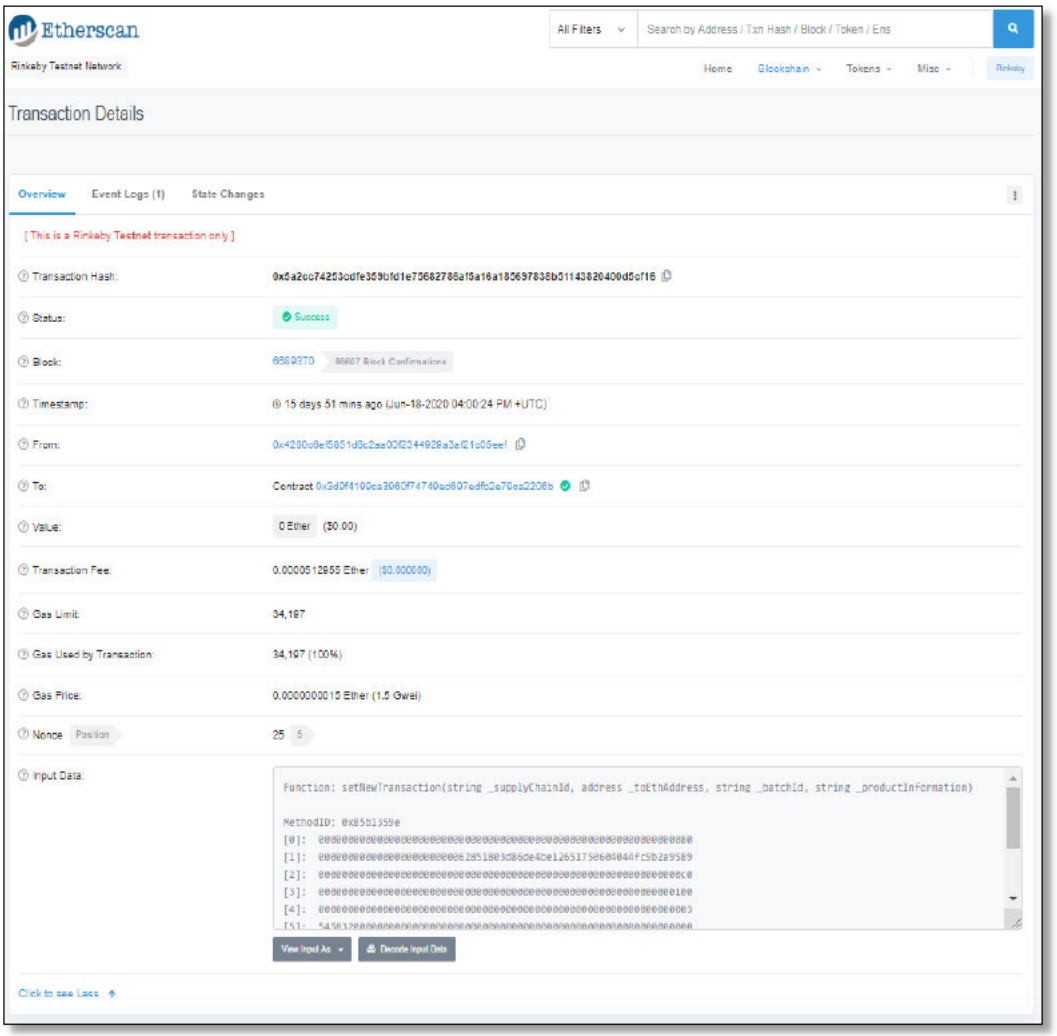

Abbildung 17: Transaktionsdetails. Dort können detailliertere Informationen über die Transaktion entnommen werden, wie z.B. welche Funktion des Smart Contracts aufgerufen und welche Events ausgeführt wurden.

Nun kann entweder web3.js mit Events oder Thegraph verwendet werden, um auf die Informationen auf der Blockchain zugreifen zu können. Es muss darauf geachtet werden, dass bei der Verwendung von der kostenlosen Version von Thegraph nur auf eine bestimmte Anzahl von vorherigen Blöcken ausgehend des neusten Blocks der Blockchain zugegriffen werden kann. Es kann danach gefiltert werden nach welcher Lieferkettennummer und welcher Chargennummer gesucht wird und ob diese bereits vollständig vorliegen. Dies kann zunächst im Online Explorer dargestellt werden, wie Abbildung 18 zeigt. Eine Anleitung wie ein Subgraph von TheGraph eingerichtet werden kann ist unter https://thegraph.com/docs/quick-start zu entnehmen.

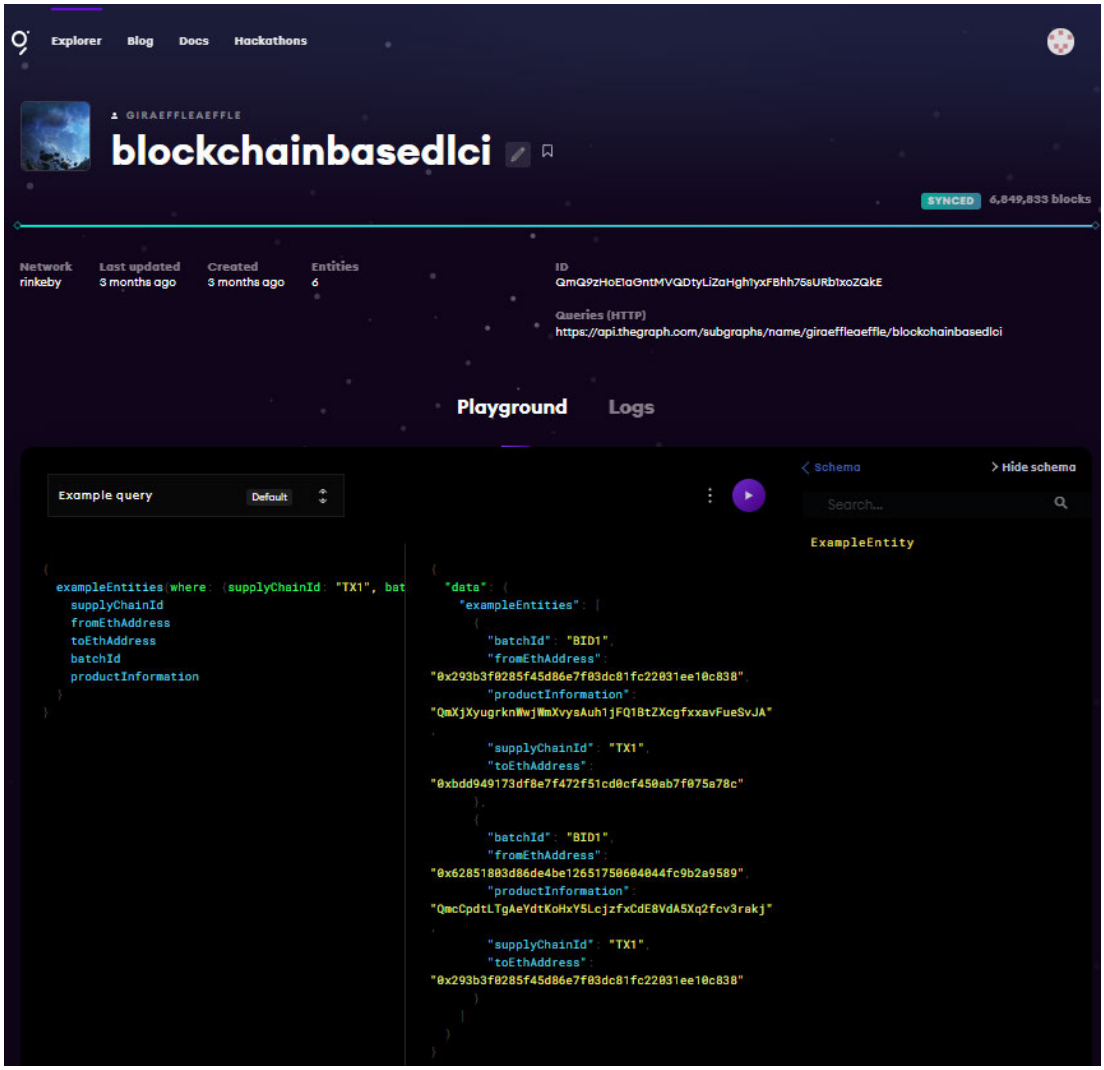

Abbildung 18: Online Explorer des Subgraphs, der mithilfe von TheGraph hochgeladen wurde. Es lässt sich auf der linken Seite, in die die Abfrage eingegeben werden kann, ein Filter erstellen, welcher die Daten ausgibt, die von Interesse sind. Die Syntax ist basierend auf der Programmiersprache GraphQL. Die Anfrage kann unter der URL https://thegraph.com/explorer/subgraph/giraeffleaeffle/blockchainbasedlci eingesehen und je nach Bedarf angepasst werden. Danach kann die Abfrage mit Strg+Eingabetaste oder durch das Drücken des violetten Knopfs ausgeführt werden. Das Ergebnis ist danach in der zweiten Spalte zu sehen.

Jedoch müsste, um diese Daten verwenden zu können, ein lokales Skript mit TypeScript geschrieben werden. Deshalb wurden die Daten auf der Blockchain zunächst mithilfe von Events, die mit web3.js durch unterschiedliche Funktionen erlangt werden können, zurückgegriffen. Dazu wird wieder ein JavaScript Programm geschrieben, welches aus einer Asynchronen Funktion besteht, die bei jeder

Transaktion, die das Event aufruft, ausgelöst wird. Diese Daten werden direkt in eine MongoDB Datenbank abgespeichert.

Nach einem festgelegten Zeitintervall kann nun immer wieder überprüft werden, ob alle Transaktionen für eine bestimmte Charge einer bestimmten Lieferketten-ID vorhanden sind. Die Produktinformation (IPFS Hashes) wurden von IPFS zurückerlangt, jedoch kann dies, wenn sich die Daten nur auf wenigen Knoten befinden, mehrere Minuten dauern. Danach wurden die jeweiligen JSON zu jeder Transaktion durchsucht und alle benötigten Werte direkt in die Matrizen A, B und k eingetragen. Anschließend konnte die Sachbilanz berechnet werden. Zum Rest des Skripts wurden Kommentare hinzugefügt, welche Schritte noch notwendig wären, um diesen Vorgang vollautomatisch durchzuführen.

In der Abbildung 19 A und B werden alle nötigen Bausteine zur Umsetzung des Modells aufgezeigt. Zuerst wird bei Teil A eine Funktion aufgezeigt, welche jedes Mal ausfgeführt wird wenn ein Event von einer Funktion des Smart Contracts ausgeführt wird und die Transaktion im Netzwerk bestätigt wird. Die eingegebenen Parameter werden direkt in einer MongoDB Datenbank abgespeichert. Danach kann die Datenbank wieder ausgelesen werden, was von B dargestellt wird. Dies kann in einem bestimmten Zeitintervall mit einem cronjob-Skript, z.B. alle 12 Stunden, geschehen. Zunächst werden alle Transaktionsdaten aus der MongoDB gelesen und die Produktinformationen von IPFS heruntergeladen. Der rot umrahmte Teil des Struktogramms wurde nicht umgesetzt, würde jedoch benötigt werden, um die Berechnung komplett zu automatisieren. Dort werden alle einzigartigen Lieferketten- und Chargennummern (SID, BID) gesucht und abgespeichert. Wenn dies abgeschlossen ist, wird der Array, in dem die SID und BID stehen, verwendet, um nach diesen Datenpaaren in der MongoDB zu suchen. Danach werden deren Daten aus IPFS angefordert und mit den Teilnehmern der Supplychain verglichen. Falls noch nicht alle Teilnehmenden dieser bestimmten Lieferkette einen Eintrag vorgenommen haben wird sich vorgemerkt, dass die Kombination der SID und BID noch unvollständig ist und danach wird die nächste SID- und BID-Kombination aus dem Array abgefragt. Wenn die Kombinationen vollständig sind kann die Sachbilanz berechnet werden,

wenn die Schritte nach der „Data collection“ ignoriert werden. Dazu wird zunächst der Array, der die Teilnehmer der Lieferkette beinhaltet durchlaufen. Bei jedem Eintrag werden Daten des Teilnehmenden aus IPFS angefordert und alle Ein- und Ausgabedaten ausgelesen. Aus den Daten werden die Technologiematrix (1) und die Umweltinterventionsmatrix (2) erstellt. Die funktionale Einheit (3), welche bestimmt wird bevor die Daten gesammelt werden, müsste die Firma, die die Daten Anfordert noch eine Transaktion senden, welche die Information über die Wahl der Funktionseinheit enthält. Anschließend kann die gesamte direkte und indirekte Umweltinterventionsmatrix (4) berechnet werden. Nun kann die Firma, die die Ökobilanz eines ihrer Produkte berechnen möchte, dies tun, ohne auf zusätzliche Durchschnittswerte einer Datenbank angewiesen zu sein.

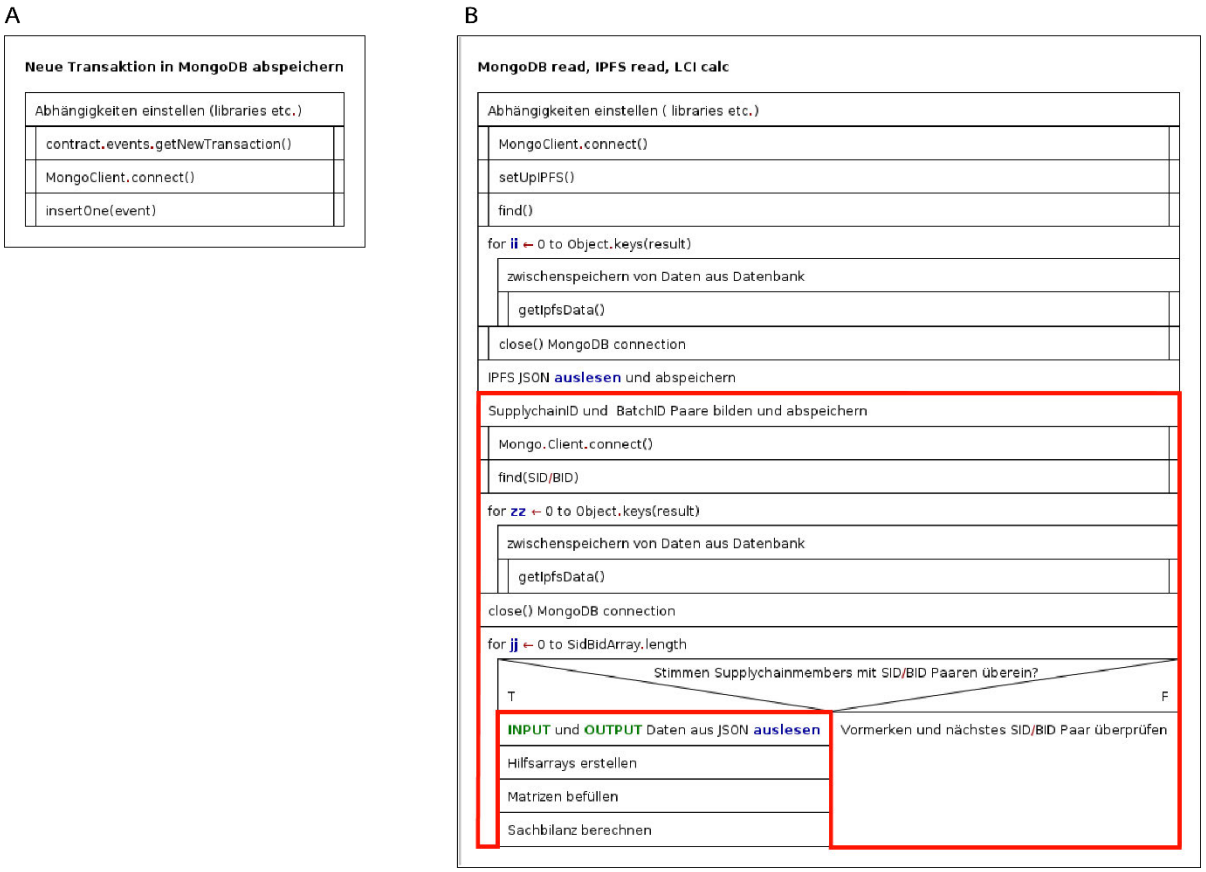

Abbildung 19: Struktogramme für die Umsetzung des Modells. Der Teil A speichert neue Transaktionen in MongoDB ab. In B werden die Transaktionsdaten aus MongoDB gelesen, die Produktinformationen von IPFS heruntergeladen, sortiert und die Sachbilanz berechnet. Der rot markierte Teil wurde nicht umgesetzt, aber könnte für eine Vollautomatisierung umgesetzt werden.

6 Diskussion

Es wurde die Hypothese, dass eine blockchainbasierte Datensammlung für eine Sachbilanz erstellt werden kann, verifiziert. Ein Modell wurde erstellt, welches umgesetzt, jedoch nicht vollautomatisiert wurde. Außerdem wurden die Schritte, die vor und nach der Datensammlung für die Erstellung einer Ökobilanz durchgeführt werden müssen nicht ausgeführt. Es wurde eine fiktive Lieferkette entworfen und ein Datensatz dafür angelegt, da wegen hoher Komplexität und Geheimhaltung kein realer Datensatz verwendet werden konnte. Ein Smart Contract wurde programmiert und die Umweltdaten auf das IPFS hochgeladen. Dazu wurde ein Front-End angefertigt, um zuerst die Datensätze auf IPFS hochzuladen und den zurückgegeben Hash in die setTransaction() Funktion des Smart Contract, zusammen mit den anderen Parametern zu übergeben. Außerdem wurde ein Skript geschrieben, dass die Daten des Events in eine MongoDB abspeichert, wenn eine Funktion des Smart Contracts ausgeführt wird. Anschließend wurden die Daten wieder aus der MongoDB entnommen, die Daten von IPFS angefordert und eine Sachbilanz berechnet.

Im Allgemeinen hat dieser Proof of Concept demonstriert, wie die DLT/Blockchain eingesetzt werden könnte, um den Prozess der Datensammlung einer Sachbilanz durchzuführen. Jedoch muss beachtet werden, dass, sobald die Daten die Blockchain verlassen und in eine Datenbank gespeichert werden, diese wieder manipulierbar sind, sofern die Datenbank nicht durch weitere Sicherheitsmaßnahmen geschützt ist. Außerdem kann das Anfragen von Daten aus IPFS, je nachdem auf wie viel Knoten diese gespeichert sind, manchmal mehrere Minuten dauern. Das Berechnen der Sachbilanz selbst müsste demnach optimalerweise direkt auf der Blockchain erfolgen, jedoch würde dies in hohen Gasgebühren resultieren und die Ergebnisse der Berechnung wären für jeden einsehbar. Es gibt allerdings schon vereinzelte Projekte, wie Hellhound von Consensys, die es für Smart Contracts möglich machen sollen Fully Homomorphic Encryption zu unterstützen (Abdelhamid Bakhta 2019). Es ist zu

erkennen, dass das Thema noch sehr neu ist und noch nicht viel in diesem Forschungsgebiet getan wurde, denn vor kurzem wurde das erste Paper mit einem Referenzrahmen für eine blockchainbasierte Ökobilanz veröffentlicht (Zhang et al. 2020). Dort wird konzeptionell aufgezeigt, in welchen Bereichen diese Technologie Vorteile bieten könnte. Es wird außerdem erwähnt, dass vor allem große Firmen eine wichtige Rolle in der Adaptierung der Technologie spielen könnten, da diese für ihre Lieferanten einen Anreiz schaffen könnten, wenn sie die Technologie implementieren. Dies könnte dazu führen, dass die Kosten für das Erstellen einer Ökobilanz verringert werden und dies außerdem mit einer höheren Genauigkeit und Geschwindigkeit durchgeführt werden kann, sowie Entscheidungen besser getroffen werden können. Allerdings ist dieses Paper nur rein theoretisch und schlägt weder ein Proof of Concept vor, noch ein konkretes Modell, wie dies praktisch umgesetzt werden könnte.

Eines der ersten Umsetzungsprobleme, welches in der realen Welt auftritt, das noch gelöst werden müsste, bevor eine blockchainbasierte Datensammlung für eine Sachbilanz integriert werden kann, ist die Erreichbarkeit. Das Projekt Starlink, welches es möglich machen soll die ganze Erde mit Satelliten zu vernetzen und Internetzugang vor allem zu Menschen zu bringen, die in ländlichen Regionen leben, ist sicherlich ein Schritt in die richtige Richtung (Tang et al. 2020). Eine gute Internetverbindung ist die Voraussetzung, um das Sammeln der Umweltdaten mit smarten IoT Sensoren, die vorher angebracht werden müssten, möglich zu machen. Diese Sensoren könnten z.B. redundant den CO_2-Ausstoß erfassen, der durch die Pumpe bei einer Ölförderung generiert wird. Anschließend können die Daten in einem Oracle Netzwerk wie Chainlink, auf das später noch detailierter eingangen wird, ausgewertet werden, der Median berechnet und danach an einen Smart Contract geschickt werden. Laut Zhang et al. (2020) wäre IoT ein großer Faktor, um genauere Daten in Realzeit bereitstellen zu können. Außerdem soll eine Analyse der gesammelten Daten in der Lieferkette mithilfe von maschinellem Lernen und künstlicher Intelligenz zu genaueren Einsichten führen und es möglich machen, schnellere und bessere Entscheidungen fällen zu können. Dies könnte dazu führen, dass Precision Farming, welches die Agrarwirtschaft mit der Informationstechnik kombiniert und daher zu einer höheren Präzision der eingesetzten Rohstoffe führt (Finger et al. 2019), auch in weniger entwickelten

Regionen der Erde realisiert werden könnte und somit das Erstellen von genaueren Datensätzen ermöglichen würde. Wenn das Anbringen der Sensoren entlang der Lieferketten funktioniert, stellt sich eine weitere Herausforderung.

In dem Modell, das vorgeschlagen wurde, wird eine einzigartige Identifikationsnummer für jede Lieferkette (supplychainID) benötigt, damit die unabhängige Institution die Daten, die sie von den Teilnehmern in einer spezifischen Lieferkette sammelt, einordnen kann. Jedoch kennen die Manufakteure laut Hellweg und Milà i Canals (2014) in der Realität meistens nur die direkten Lieferanten, das heißt Tier 1 und nicht weiter hinaus. Dies stellt ein großes Problem dar, da sich alle Teilnehmer einer Lieferkette einigen müssten, ohne sich direkt zu kennen. Es müsste ein Mechanismus entwickelt werden, der dafür sorgt, dass sich keiner der Teilnehmer im Netzwerk kennen müsste und die Daten trotzdem richtig eingeordnet werden könnten. Angenommen dieser Mechanismus würde schon existieren, gibt es eine weitere Schwachstelle, die behoben werden müsste.

Da die Umweltdaten den Strom- und Wasserverbrauch und Ausstoß von Schadstoffen, wie z.B. CO2 darstellen, könnte von diesen auf Herstellungsprozesse der Firma zurückgeschlossen werden, was diese Daten vertraulich macht. Keine Firma hat Interesse daran einem Wettbewerber einen Vorteil zu bringen, wenn dieser diese Daten kennt. Das Ethereum Netzwerk ist eine öffentliche Blockchain und somit könnte jeder auf diese Daten zugreifen. Daher wäre es ratsam diese Daten vor dem Hochladen auf IFPS in einer Art und Weise zu verschlüsseln, die es möglich macht, dass keiner, nicht einmal die unabhängige Institution, die Daten kennt. Eine Möglichkeit dies umzusetzen, ist die Anwendung von homomorpher Verschlüsselung, mit der Berechnungen auf verschlüsselte Daten durchgeführt werden können. Jedoch gibt es bisher nur homomorphe Verschlüsselung auf einer single-key Basis, die von Microsoft SEAL implementiert wurde, was bedeutet, dass jeder Teilnehmer, denselben private Key verwenden müsste (Chen et al. 2017). Dies wäre nicht wünschenswert, da dann das gleiche Problem wie zuvor auftritt, da jede Firma die Daten von anderen Firmen kennen würde. Eine potentielle Lösung dazu ist eine multi-key homomorphe Verschlüsselung, die es mehreren Teilnehmern möglich macht zusammen ein

Ergebnis einer Funktion zu berechnen, ohne ihren individuellen Eingangswerten enthüllen zu müssen an der Chen et al. (2019) mit einem Proof of Concept arbeitet.

Wie könnte jedoch erreicht werden, dass alle Teilnehmer mit gutem Motiv handeln und die Daten nicht manipuliert werden, da es nicht möglich wäre diese zu überprüfen? Es müsste ein Anreiz geschaffen werden, dass die Firmen ehrlich handeln, wie z.B. das Vergeben von steuerlicheren Vorteilen und von einem grünen Produktlabel. Außerdem wurde im März 2020 das Baseline Protocol zusammen von mehreren größeren Firmen veröffentlicht, welches dafür sorgen soll, dass Firmen auf einem gemeinsamen Referenzrahmen, wie dem Ethereum Netzwerk, das immer online ist, in einer Weise arbeiten zu können, dass sie ihre persönlichen Daten nicht preisgeben müssen (OASIS 2020). Ihr Ziel ist es, dass dieses Protokoll zum neuen technologischen Standard wird, mit dem Firmen Geschäftsprozesse miteinander abwickeln. Damit die Daten, die die Firmen miteinander austauschen vertraulich bleiben, werden diese nicht direkt auf die Blockchain, sondern nur ein sogenannter Zero Knowledge Proof hochgeladen. Dieser macht es möglich Anderen etwas zu beweisen, ohne die vertraulichen Daten selbst bekannt geben zu müssen (Quisquater et al. 1990). Darüber hinaus schafft Unibright es, Firmen die keine Erfahrung mit der DLT haben, einen Einstieg in die Branche zu bieten. Unibright ist eine eine deutsche Softwareentwicklungsfirma, die in Blockchain Anwendungen spezialisiert ist und deren Geschäftsführer über 20 Jahre Erfahrung mit Geschäftsprozessen und derer Integrierung haben. Der Service der Firma ist vergleichbar wie ein Browser zum Internet, der zur Interaktion mit dem Baseline Protocol dient. Im August 2020 wurde eine Pressemittteilung von Unibright und den Unternehmen Provide und Chainlink bekanntgegeben, dass diese ein Projekt mit Coca-Cola haben, dass die Verbesserung der Lieferkette derer Flaschen in Nordamerika zum Ziel hat (Unibright 03.08.2020; Provide 03.08.2020).

Angenommen, dass das Problem der Geheimhaltung auch gelöst worden wäre, gibt es noch einen weiteren Punkt, in den tiefer eingegangen werden sollte. Um dieses System auf die zirkuläre Wirtschaft anzuwenden, müssten alle Produkte, die von Menschen gekauft wurden, mit einem Mechanismus ausgestattet werden, der es

ermöglicht diese Produkte auch nach dem Zeitpunkt ihrer Herstellung zu verfolgen. Da das Ziel der zirkulären Wirtschaft die maximale Verlängerung der Lebenszeit eines Produkts ist, müsste ein Procedere entwickelt werden, dass z.B. der Vorgang einer Reparatur eines Toasters, der kaputt gegangen ist, mit in die Sachbilanz einberechnet werden könnte. Nachdem der Toaster nicht mehr zu reparieren ist, kann dieser erneuert oder recycelt und zu einem anderen Gegenstand umfunktioniert werden. Ab diesem Zeitpunkt wird es sehr schwierig noch nachzuvollziehen zu können, ab wann die Berechnung noch zu dem alten Toaster gehört oder schon zu dem Rohstoffeingang des neuen Produkts gehört.

Ein weiteres Hindernis ist, dass viele Firmen nicht bereit sind oder es für diese nicht erlaubt ist Kryptowährungen zu halten, welche benötigt werden, um die Gebühren im Ethereum Netzwerk zu bezahlen. Jedoch gibt es schon Projekte die daran arbeiten, dass sie für den Service, den der Kunde haben möchte, bezahlen und diese auch keine Kryptowährungen halten müssen, da es möglich ist, in Fiatgeld zu bezahlen. Die Regulierungen für das Halten von Krytowährungen lockern sich jedoch stetig, da es laut BGBI Artikel 2 §64y seit 1. Januar 2020 deutschen Banken erlaubt ist Kryptowerte zu verwahren (Bundestag und Bundesrat 2019). Kryptowährungen und die Standards, aus denen sie bestehen sog. Token allgemein sehr interessant für verschiedene Bereiche und können alles auf einer Blockchain repräsentieren. Von Aktien einer Firma bis hin zu Geld und Zeit. Es gibt verschiedene Token Standards, die für unterschiedliche Aufgaben besser geeignet sind. Token werden in fungible und non-fungible unterschieden. Fungible Token sind identisch zu anderen Tokens von diesem Typ. Daher sind sie gegen andere Token, die denselben Wert haben austauschbar. Außerdem können diese Token in sehr kleine Einheiten aufgeteilt werden. Ein Beispiel dafür ist der ERC20 Token Standard (VOSHMGIR 2019). Non-Fungible Token oder auch NFTs genannt sind im Gegensatz zu den fungible Token einzigartig und unterscheidet sich von anderen Token desselben Typs, da sie einzigartige Informationen oder Attribute aufweisen, wie das geben von Zugangsrechten oder ein Zertifikat. Außerdem können NFTs wegen diesen Eigenschaften nicht gegen andere Token desselben Typs ausgetauscht werden. Des Weiteren sind diese Token auch nicht teilbar, da sie direkt an eine Identität gebunden sind und es deshalb nicht sinnvoll

wäre einen Teil eines Zugangsrechts oder eines Zertifikats zu haben (VOSHMGIR 2019).

In dem Paper von Zhang et al. (2020) wird außerdem davon gesprochen, dass es zu gewollter oder ungewollter Manipulation der Daten kommen kann, was zu Unstimmigkeiten der Daten führen könnte. Ein Lösungsansatz dazu wäre ein sogenanntes dezentralisiertes Oracle-Netzwerk wie Chainlink. Ein Oracle wird benötigt, um es einem Smart Contract zu ermöglichen mit Daten aus der realen Welt zu interagieren, wenn es eine Statusänderung im Smart Contract gibt. Zum Beispiel könnte es eine Dateneinspeisung zu den $kgCO_2$/kWh, €/$kgCO_2$ oder €/kWh Paaren geben, die sich minütlich durch Sensoren und verschiedene Parameter ändert und auf der der Smart Contract angewiesen ist. Dazu wird ein Netzwerk von unterschiedlichen Teilnehmern, die die Datenflüsse zur Verfügung stellen aufgebaut. Dies können am Beispiel von smarten Sensoren entlang der Lieferkette Sensoren von zwei Parteien sein, die direkte Zulieferer sind. Dies gewährt Redundanz in der Datensammlung, um somit einen Single Point of Failure ausschließen zu können.

Das Problem, welches Blockchains mit der Speicherung von zu vielen Daten haben, das in Zhang et al. (2020) erwähnt wurde, könnte komplett durch die Verwendung des IPFS, wie es in der Arbeit gezeigt wurde, gelöst werden. Ein großer Nachteil ist jedoch, dass dies eventuell nicht konform mit dem General Data Protection Regulation (GDPR) ist, da aufgrund dieses Gesetzes angefordert werden kann private Daten zu löschen, da laut Art. 17 GDPR jeder das Recht hat vergessen zu werden, jedoch die Daten in dem IPFS Netzwerk und auf der Ethereum Blockchain nicht löschbar sind (Voigt und Bussche 2017).

Durch die Eigenschaften der Blockchain könnte sich das Supply Chain Management in das Demand Chain Management (DCM) umwandeln, bei der die Interessen des Kunden im Mittelpunkt stehen. Die Kosten werden reduziert, ein leistungsfähiger Kundenservice ist essenziell und eine schnellere Markteinführung einer Idee oder eines minimal überlebensfähigen Produkts wünschenswert. Damit dies funktioniert, haben alle Akteure die Möglichkeit in Echtzeit auf die Anforderungen der Kunden

zugreifen zu können. Hierbei ist es wichtig, dass alle Knotenpunkte der Lieferkette eng miteinander verbunden sind. Beim Supply Chain Management ist es dagegen wichtig, dass der Warenfluss immer garantiert ist. Dies bringt jedoch mit sich, dass Markteinschätzungen oft auf ungenauen Daten beruhen können, wobei das DCM einen genauen Überblick über den Markt generieren soll, damit optimale Entscheidungen, die die Produktion betreffen, getroffen werden können (Wüst und Gervais 2018). Jedoch ist noch nicht klar, ob die Blockchain die richtige Lösung für die Verbesserungen des Supply Chain Management ist. Ein Problem ist, dass die meisten Teilnehmer nicht nur in einer Lieferkette eine Rolle spielen. Deshalb würde eine einzige Blockchain für alle Lieferketten, in der der Akteur involviert ist, benötigt werden, was die Leistungsfähigkeit durch die geringe Transaktionsrate verschlechtern würde, woran jedoch schon gearbeitet wird. Ein weiteres Problem ist, dass die Interaktion von der digitalen mit der analogen Welt immer auf Vertrauen basiert. Wenn z.B. ein LKW mit Tiefkühlwaren einen Sensor im Laderaum angebracht hat, der in den jeweiligen Zeitabständen immer wieder den Status des gekühlten Guts an die Blockchain sendet, könnte der Sensor einfach in eine separat gekühlte Box gelegt werden und der Rest des Laderaums nicht oder nur weniger als die zugelassene Temperatur gekühlt werden, um Kosten zu sparen. Dem nächsten Glied in der Lieferkette müsste genauso vertraut werden, da dort normalerweise geprüft werden muss, ob das Gut im Lager angekommen ist und die Qualität keine Einbußen hatte (Wüst und Gervais 2018).

Trotz all der Probleme, die noch in Bezug auf eine blockchainbasierte Sachbilanz auftreten, erwähnt Saberi et al. (2019) vor allem, wie die Blockchain einen Beitrag im Bereich des Umweltschutzes leisten könnte. Es werden unter anderem die Bereiche Supply Chain Management, Umweltabkommen, Recycling, der Energiesektor und Steuern basierend auf dem CO2-Fußabdruck der Firmen mit jeweiligen Projekten dazu erwähnt (Future Thinkers 2017). Um ein Beispiel im Bereich Recycling zu nennen, gibt es das Projekt plasticbank, das Menschen in Indonesien, in dem Plastikmüll ein großes Problem darstellt, dazu animiert Plastik zu sammeln und gegen Kryptowährungen einzutauschen. Menschen, die gerne grüne Produkte kaufen, den Lieferweg des Produkts jedoch nicht kennen, könnten durch den Einsatz der Blockchain profitieren,

da dies zu besser informierten Verbraucherentscheidungen führen könnte, die stärker mit ihren nachhaltigen Werten übereinstimmen.

7 Zusammenfassung und Ausblick

Die Hypothese, ob eine blockchainbasierte Datensammlung für eine Sachbilanz erstellt werden könnte, wurde verifiziert. Es bietet den Vorteil, dass dies in einer transparenten und sicheren Weise durchgeführt werden kann. Dazu wurde zuerst ein Modell angefertigt. Danach wurde dieser Ansatz umgesetzt, einige Tests durchgeführt und eine halb-automatisierte Sachbilanz berechnet. Außerdem konnten z.B. der herkömmliche und der blockchainbasierte Ansatz nicht verglichen werden, da hierfür eine größere Studie mit echten Datensätzen, mit unterschiedlichen Testszenarien und zusätzlichen Lieferketten benötigt wären. Außerdem gibt es einige Limitierungen des Ansatzes, da dieser nur einen kleinen Teil der kompletten Ökobilanz abdeckt. Des Weiteren kann gesagt werden, dass das Thema der Verwendung der Blockchain in der Lieferkette und in der Ökobilanz noch relativ neu ist. Die verwendete Smart Contract Plattform Ethereum feierte am 30.07.20 erst ihr fünfjähriges Bestehen. Jedoch ist dieser Bereich sehr innovativ und wie schon vorher erwähnt wurde, möchten viele Firmen Expertise in der DLT aufbauen oder zumindest testen, ob sie davon profitieren könnten. Eines der ersten Paper zu dem Thema Blockchain in der Ökobilanz wurde Anfang dieses Jahres veröffentlicht und stellt dort einen ersten Referenzrahmen, jedoch noch keinen PoC oder eine Implementierung vor. Dies zeigt, wie jung dieses Forschungsgebiet ist. Weitere Studien müssten in den Bereichen der Internetverfügbarkeit, Sensorausstattung entlang der Lieferkette, Kommunikation zwischen Herstellern und Lieferanden, Geheimhaltung, Anreize für Teilnehmende, zirkuläre Wirtschaft und gesetzlichen Regulierungen und andere Technologien, die noch nicht ihren benötigten Reifegrad erreicht haben durchgeführt werden, um einen Schritt in eine Zukunft zu gehen, in der Ökobilanzen noch höhere Genauigkeiten aufweisen können.

8 Literaturverzeichnis

Abdelhamid Bakhta, Amira Bouguera (2019): Hellhound Red Paper. Online verfügbar unter https://github.com/ConsenSys/hellhound/blob/master/hellhound-red-paper.pdf.

Benet, Juan (2014): IPFS - Content Addressed, Versioned, P2P File System. Online verfügbar unter http://arxiv.org/pdf/1407.3561v1.

Bruno Oberle et al. (2019): Global Resources Outlook 2019. Natural resources for the future we want. Online verfügbar unter https://wedocs.unep.org/bitstream/handle/20.500.11822/27517/GRO_2019.pdf, zuletzt geprüft am 12.09.2020.

Bundestag und Bundesrat (Hg.) (2019): Gesetz zur Umsetzung der Änderungsrichtlinie zur Vierten EU-Geldwäscherichtlinie. Online verfügbar unter https://www.bgbl.de/xaver/bgbl/text.xav?SID=&tf=xaver.component.Text_0&tocf=&qmf=&hlf=xaver.component.Hitlist_0&bk=bgbl&start=%2F%2F*%5B%40node_id%3D%27632876%27%5D&skin=pdf&tlevel=-2&nohist=1, zuletzt geprüft am 12.09.2020.

Chen, Hao; Chillotti, Ilaria; Song, Yongsoo (2019): Multi-Key Homomorphic Encryption from TFHE. In: Steven D. Galbraith und Shiho Moriai (Hg.): Advances in Cryptology – ASIACRYPT 2019, Bd. 11922. Cham: Springer International Publishing (Lecture Notes in Computer Science), S. 446–472.

Chen, Hao; Laine, Kim; Player, Rachel (2017): Simple Encrypted Arithmetic Library - SEAL v2.1. In: Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A Ryan, Vanessa Teague et al. (Hg.): Financial Cryptography and Data Security, Bd. 10323. Cham: Springer International Publishing (Lecture Notes in Computer Science), S. 3–18.

Church, Alonzo; Turing, A. M. (1937): On Computable Numbers, with an Application to the Entscheidungsproblem. In: *The Journal of Symbolic Logic* 2 (1), S. 42. DOI: 10.2307/2268810.

Conte de Leon, Daniel; Stalick, Antonius Q.; Jillepalli, Ananth A.; Haney, Michael A.; Sheldon, Frederick T. (2017): Blockchain: properties and misconceptions. In: *Asia Pcfc Jrnl Innvtn & Entrprnshp* 11 (3), S. 286–300. DOI: 10.1108/APJIE-12-2017-034.

Dr. Gavin Wood (2019): ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER. Online verfügbar unter https://ethereum.github.io/yellowpaper/paper.pdf, zuletzt aktualisiert am 20.10.2019, zuletzt geprüft am 15.04.2020.

Edelen, Ashley; Ingwersen, Wesley W. (2018): The creation, management, and use of data quality information for life cycle assessment. In: *Int J Life Cycle Assess* 23 (4), S. 759–772. DOI: 10.1007/s11367-017-1348-1.

Finger, Robert; Swinton, Scott M.; El Benni, Nadja; Walter, Achim (2019): Precision Farming at the Nexus of Agricultural Production and the Environment. In: *Annu. Rev. Resour. Econ.* 11 (1), S. 313–335. DOI: 10.1146/annurev-resource-100518-093929.

Future Thinkers (2017): 7 Ways The Blockchain Can Save The Environment and Stop Climate Change. Online verfügbar unter https://futurethinkers.org/blockchain-environment-climate-change/, zuletzt geprüft am 24.05.2020.

Haupt, Melanie; Zschokke, Mischa (2017): How can LCA support the circular economy?—63rd discussion forum on life cycle assessment, Zurich, Switzerland, November 30, 2016. In: *The international journal of life cycle assessment* 22 (5), S. 832–837. DOI: 10.1007/s11367-017-1267-1.

Hauschild, Michael Z.; Rosenbaum, Ralph K.; Olsen, Stig Irving (2018): Life Cycle Assessment. Cham: Springer International Publishing.

Hellweg, Stefanie; Milà i Canals, Llorenç (2014): Emerging approaches, challenges and opportunities in life cycle assessment. In: *Science (New York, N.Y.)* 344 (6188), S. 1109–1113. DOI: 10.1126/science.1248361.

Islam, Samantha; Ponnambalam, S. G.; Lam, Hon Loong (2016): Review on life cycle inventory: methods, examples and applications. In: *Journal of Cleaner Production* 136, S. 266–278. DOI: 10.1016/j.jclepro.2016.05.144.

Joint Research Center (JRC) (2010): International Reference Life Cycle Data System (ILCD) Handbook - Specific guide for Life Cycle Inventory (LCI) data sets. Luxembourg: Publications Office (EUR (Luxembourg), 24709).

Kamath, Reshma (2018): Food Traceability on Blockchain: Walmart's Pork and Mango Pilots with IBM. In: *The JBBA* 1 (1), S. 1–12. DOI: 10.31585/jbba-1-1-(10)2018.

Kareiva, Peter M.; McNally, Brynn W.; McCormick, Steve; Miller, Tom; Ruckelshaus, Mary (2015): Improving global environmental management with standard corporate reporting. In: *Proceedings of the National Academy of Sciences of the United States of America* 112 (24), S. 7375–7382. DOI: 10.1073/pnas.1408120111.

Kashmanian, Richard M.; Moore, Justin R. (2014): Building Greater Sustainability in Supply Chains. In: *Environmental Quality Management* 23 (4), S. 13–37. DOI: 10.1002/tqem.21376.

Kuczenski, Brandon; Marvuglia, Antonino; Astudillo, Miguel F.; Ingwersen, Wesley W.; Satterfield, M. Barclay; Evers, David P. et al. (2018): LCA capability roadmap—product system model description and revision. In: *Int J Life Cycle Assess* 23 (8), S. 1685–1692. DOI: 10.1007/s11367-018-1446-8.

Lamport, Leslie; Shostak, Robert; Pease, Marshall (1982): The Byzantine Generals Problem. In: *ACM Trans. Program. Lang. Syst.* 4 (3), S. 382–401. DOI: 10.1145/357172.357176.

Leible, Stephan; Schlager, Steffen; Schubotz, Moritz; Gipp, Bela (2019): A Review on Blockchain Technology and Blockchain Projects Fostering Open Science. In: *Front. Blockchain* 2, S. 1. DOI: 10.3389/fbloc.2019.00016.

Lenzen, Manfred; Crawford, Robert (2009): The path exchange method for hybrid LCA. In: *Environ. Sci. Technol.* 43 (21), S. 8251–8256. DOI: 10.1021/es902090z.

Mercedes-Benz (2020): Mercedes-Benz Cars drives "Ambition2039" in the supply chain: blockchain pilot project provides transparency on CO2 emissions. Online verfügbar unter https://media.daimler.com/marsMediaSite/en/instance/ko/Mercedes-Benz-Cars-drives-Ambition2039-in-the-supply-chain-blockchain-pilot-project-provides-transparency-on-CO2-emissions.xhtml?oid=45528015, zuletzt geprüft am 03.04.2020.

Merkle, Ralph C. (1988): A Digital Signature Based on a Conventional Encryption Function. In: G. Goos, J. Hartmanis, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries et al. (Hg.): Advances in Cryptology — CRYPTO ’87, Bd. 293. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 369–378.

Nguyen, Kien; Nguyen, Thinh; Kovchegov, Yevgeniy (2009 - 2009): A P2P Video Delivery Network (P2P-VDN). In: 2009 Proceedings of 18th International Conference on Computer Communications and Networks. 2009 Proceedings of 18th International Conference on Computer Communications and Networks - ICCCN 2009. San Francisco, CA, USA, 03.08.2009 - 06.08.2009: IEEE, S. 1–7.

OASIS (2020): Baseline Protocol. Paul Brody, Yorke Rhodes, John Wolpert et al. Online verfügbar unter https://docs.baseline-protocol.org/.

OECD (2018): The Macroeconomics of the Circular Economy Transition. A Critical Review of Modelling Approaches. Paris: OECD Publishing ((Keine Angabe), no. 130).

Pizzol, Massimo; Weidema, Bo; Brandão, Miguel; Osset, Philippe (2015): Monetary valuation in Life Cycle Assessment: a review. In: *Journal of Cleaner Production* 86, S. 170–179. DOI: 10.1016/j.jclepro.2014.08.007.

Provide (03.08.2020): Provide and Chainlink Collaborate to Make a New Easy Button for Enterprise Smart Contract Oracles. Online verfügbar unter https://provide.services/provide-and-chainlink-collaborate-to-make-a-new-easy-button-for-enterprise-smart-contract-oracles/, zuletzt geprüft am 18.08.2020.

Quisquater, Jean-Jacques; Quisquater, Myriam; Quisquater, Muriel; Quisquater, Michaël; Guillou, Louis; Guillou, Marie Annick et al. (1990): How to Explain Zero-Knowledge Protocols to Your Children. In: Gilles Brassard (Hg.): Advances in Cryptology — CRYPTO’ 89 Proceedings, Bd. 435. New York, NY: Springer New York (Lecture Notes in Computer Science), S. 628–631.

Saberi, Sara; Kouhizadeh, Mahtab; Sarkis, Joseph; Shen, Lejia (2019): Blockchain technology and its relationships to sustainable supply chain management. In: *International Journal of Production Research* 57 (7), S. 2117–2135. DOI: 10.1080/00207543.2018.1533261.

Samuel Haig (2020): EY, Microsoft and ConsenSys Launch Enterprise Platform on Ethereum Mainnet. Online verfügbar unter https://cointelegraph.com/news/ey-microsoft-and-consensys-launch-enterprise-platform-on-ethereum-mainnet, zuletzt geprüft am 03.04.2020.

Selvakumar, Arul Lawrence; Ganadhas, C. Suresh (2009 - 2009): The Evaluation Report of SHA-256 Crypt Analysis Hash Function. In: 2009 International Conference on Communication Software and Networks. 2009 International Conference on

Communication Software and Networks. Chengdu Sichuan, China, 27.02.2009 - 28.02.2009: IEEE, S. 588–592.

Sompolinsky, Yonatan; Zohar, Aviv (2015): Secure High-Rate Transaction Processing in Bitcoin. In: Rainer Böhme und Tatsuaki Okamoto (Hg.): Financial Cryptography and Data Security, Bd. 8975. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 507–527.

Stahel, Walter R. (2016): The circular economy. In: *Nature* 531 (7595), S. 435–438. DOI: 10.1038/531435a.

Suh, Sangwon; Huppes, Gjalt (2010): Methods in the Life Cycle Inventory of a Product. In: Sangwon Suh (Hg.): Handbook of input-output economics in industrial ecology, Bd. 23. [Nachdr.]. Dordrecht: Springer (Eco-Efficiency in Industry and Science, 23), S. 263–282.

Tang, Zhu; Li, Sudan; Deng, Wenping; Wang, Yongzhi; Yu, Wanrong (2020): Optimization of Satellite-Ground Coverage for Space-Ground Integrated Networks Based on Discrete Global Grids. In: Quan Yu (Hg.): Space Information Networks, Bd. 1169. Singapore: Springer Singapore (Communications in Computer and Information Science), S. 132–144.

Ting, Daniel Shu Wei; Carin, Lawrence; Dzau, Victor; Wong, Tien Y. (2020): Digital technology and COVID-19. In: *Nat Med* 542, S. 125. DOI: 10.1038/s41591-020-0824-5.

Unibright (03.08.2020): Baselining the North America Coca-Cola Bottling Supply Chain. Online verfügbar unter https://medium.com/unibrightio/baselining-the-north-america-coca-cola-bottling-supply-chain-f87539220269, zuletzt geprüft am 18.08.2020.

Voigt, Paul; Bussche, Axel von dem (2017): The EU General Data Protection Regulation (GDPR). A practical guide. Cham, Switzerland: Springer. Online verfügbar unter http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&local_base=BVB01&doc_number=029688307&sequence=000002&line_number=0002&func_code=DB_RECORDS&service_type=MEDIA.

VOSHMGIR, SHERMIN (2019): TOKEN ECONOMY: How blockchains and smart contracts revolutionize the economy. [S.l.]: SHERMIN VOSHMGIR.

Wolf, Marc-Andree; Pant, Rana; Chomkhamsri, Kirana; Sala, Serenella; Pennington, David (2012): The International reference Life Cycle Data system (ILCD) handbook. Towards more sustainable production and consumption for a resource-efficient Europe. Luxembourg: Publications Office (EUR. Scientific and technical research series, 24982).

Wüst, Karl; Gervais, Arthur (2018): Do you Need a Blockchain?, S. 45–54. DOI: 10.1109/CVCBT.2018.00011.

Xiao, Yang; Zhang, Ning; Lou, Wenjing; Hou, Y. Thomas (2020): A Survey of Distributed Consensus Protocols for Blockchain Networks. In: *IEEE Commun. Surv. Tutorials*, S. 1. DOI: 10.1109/COMST.2020.2969706.

Yaga, Dylan; Mell, Peter; Roby, Nik; Scarfone, Karen (2018): Blockchain technology overview. Gaithersburg, MD.

Yuan, Yong; Wang, Fei-Yue (2018): Blockchain and Cryptocurrencies: Model, Techniques, and Applications. In: *IEEE Trans. Syst. Man Cybern, Syst.* 48 (9), S. 1421–1428. DOI: 10.1109/TSMC.2018.2854904.

Zhang, Abraham; Zhong, Ray Y.; Farooque, Muhammad; Kang, Kai; Venkatesh, V. G. (2020): Blockchain-based life cycle assessment: An implementation framework and system architecture. In: *Resources, Conservation and Recycling* 152, S. 104512. DOI: 10.1016/j.resconrec.2019.104512.

9 Anhang

Der Anhang der Masterthesis befindet sich auf einer DVD-R, die zusammen mit der physischen Version der Thesis abgegeben wurde.

BESCHREIBUNG	SPEICHERORT
MASTERTHESIS ALS PDF	Masterthesis_Maximilian_Stahl.pdf
CODE: IN DEN ORDNERN BEFINDEN SICH NOCH ZUSÄTZLICHE README.MD, DIE BESCHREIBEN WAS GETUN WERDEN MUSS, FALLS DER CODE GETESTET WERDEN MÖCHTE	Code_Masterthesis/... Code_Masterthesis/LCI_UI Code_Masterthesis/buidler
LITERATUR + CITAVI	Literatur/...
SELBST ERSTELLTE ABBILDUNGEN	Bilder/...
HINWEISE ZUR AUSFÜHRUNG DER PROGRAMME	README.md

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.
Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Stahl

Vorname: Maximilian

dass ich die vorliegende Masterarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Entwicklung und Implementierung einer blockchainbasierten Sachbilanz für die Ökobilanzierung von Waren und Dienstleistungen

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Ort Datum Unterschrift im Original