VERY LARGE
BUSINESS APPLICATIONS
Carl von Ossietzky Universität Oldenburg

Chargenrückverfolgung in der Fleischwarenindustrie - Konzeption und prototypische Implementierung einer Blockchain Lösung

Masterarbeit

Themensteller:	Prof. Dr.-Ing. Jorge Marx Gómez
Betreuer:	Stefan Wunderlich (M.Sc.)
Vorgelegt von:	Nils Lutz Erlenweg 5 26129 Oldenburg +49 173 25 28 407 nils.lutz@uni-oldenburg.de
Abgabetermin:	30. April 2017

Inhaltsverzeichnis

Akronyme

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1. Einleitung

1.1. Motivation

> *„Weltweit ist die Fleischerzeugung zwischen 2002 und 2012 um 23% und in Deutschland um 29% gestiegen. Die globalen Fleischexporte erhöhten sich im gleichen Zeitraum um 60%, in Deutschland sogar um 124%. Deutschland zählt sowohl beim Import als auch beim Export von Fleisch- und Fleischprodukten zu den bedeutendsten Handelsnationen weltweit.“*
>
> Efken et al. (2015)

Lebensmittelsicherheit ist strategisch für die Volksgesundheit und das Wohlbefinden einer Gesellschaft. Der öffentliche Druck auf Hersteller für eine ausreichende Kennzeichnung von Produkten und ihre Bestandteile wird stetig größer. Jeder Teil der Lieferkette ist in der Verpflichtung im Falle von Kontamination schnellstmöglich reagieren zu können. (Europa Parlament und Europäischer Rat, 2002).

Vom Rohstofflieferanten bis zum Endkunden gibt es allein in Deutschland ein Netz von Marktteilnehmern mit erheblicher Größe. Knapp 150.000 Betriebe für die Rinder Mast und Milchproduktion, etwa 30.000 Betriebe im Bereich der Schweinehaltung und rund 60.000 Unternehmen für die Geflügelhaltung (Efken et al., 2015). Dabei existiert kein Standardverfahren zwischen diesen Marktteilnehmern zum Informationsaustausch für die Chargenrückverfolgung. In der Fleischwarenindustrie beispielsweise existieren weit über 140 unterschiedliche Austauschformate zwischen den Teilnehmern einzelner Lieferketten.

Zum jetzigen Zeitpunkt (Stand 2019) findet eine Chargenrückverfolgung daher fast ausschließlich durch einen Datei-Austausch bzw. eine zentrale Datenbank je Teilnehmer der Lieferkette statt. Dabei müssen Informationen für einen mehrstufigen Produktionsprozess bereitgestellt und verarbeitet werden (Siepermann et al., 2015).

Aus der geringen Umsatzrendite von -1% bis +1,5% und den dadurch entstehnden Druck am Markt bestehen zu bleiben resultieren immer häufiger Unregelmäßigkeiten innerhalb der Lieferkette. Nur Betriebe in Österreich und Spanien können eine langfristige Rentabilität innerhalb des europäischen Marktes aufweisen (Efken et al., 2015). Ein Beispiel für die genannten Unregelmäßigkeiten ist der „Pferdefleisch

Skandal“ aus dem Jahr 2013, bei dem Fleischprodukte nachträglich neu etikettiert und dadurch in Produkten wie Lasagne oder Hamburger Patties weiterverarbeitet wurden (Die Grünen, 2013).

Informationen der Lieferkette und einzelner Chargen werden zentral je Hersteller oder Transportunternehmen gepflegt und sind dadurch nicht ausreichend vor Manipulation geschützt innerhalb der gesamten Lieferkette. Die *Blockchain-Technologie* ermöglicht das manipulationssichere ablegen von solchen Informationen und könnte daher eine Lösung für dieses Problem darstellen. Bereits heute gibt es Anwendungen der *Blockchain*, um beispielsweise den Kilometerstand eines Fahrzeugs täglich „in die *Blockchain*“ zu schreiben. Die inhärenten Eigenschaften der *Blockchain* ermöglichen es sehr einfach festzustellen, ob ein Kilometerstand nachträglich durch Fremdeinwirkung manipuliert wurde. Ebenfalls ist keine zentrale „Clearing Stelle“ mehr nötig, um die Echtheit des hinterlegten Wertes sicherzustellen (carVertical, 2017).

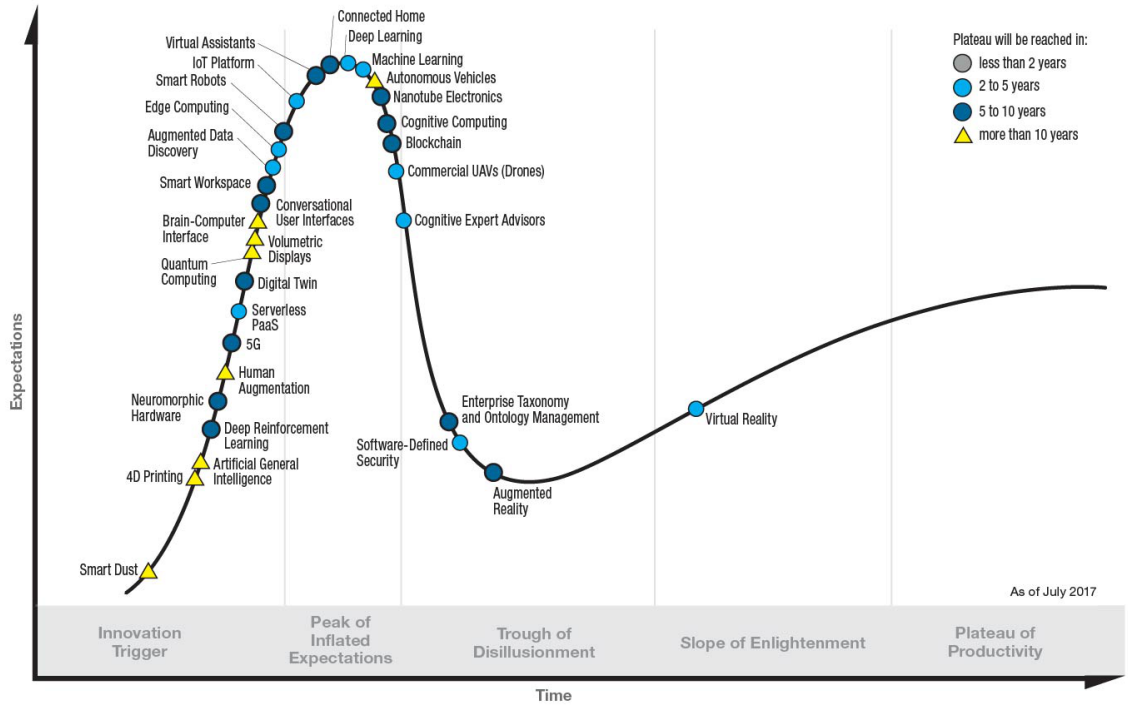

Abbildung 1: Emerging Technologies Hype Cycle 2017(Panetta, 2017)

Aktuell ist die *Blockchain* jedoch noch kein industrieller Standard oder verbreitet im Einsatz. Bemessen am jährlich erscheinenden Hype Cycle des Marktforschungsinstituts Gartner, Inc. (*Abb.* 1) hat die Technologie noch fünf bis zehn Jahre Entwicklungszeit vor sich. Erst dann wird sie nach aktueller Einschätzung im produktiven Einsatz sein.

„Es ist davon auszugehen, dass wir in ein bis zwei Jahrzehnten wirtschaftlich über Mechanismen miteinander interagieren werden, für die wir bislang weder Konzepte noch Begriffe haben“ (Platzer, 2014, S. 92). Auch die Deutsche Bundesregierung ist an der *Blockchain-Technologie* interessiert und erwägt den Einsatz in Zukunft für die unterschiedlichsten Services. In einer der jüngsten Pressemitteilungen hat der *Blockchain* Bundesverband mitgeteilt, dass die Regierung eine umfassende Strategie zum Umgang und Einsatz der Technologie erarbeiten will (Florian Glatz, 2018).

1.2. Problemstellung

Um eine formal korrekte Identitätskette vom Erzeuger bis zum Groß- und Einzelhandel aufzubauen, wird eine verlässliche Basis, grade auch dann, wenn Futtermittel- und Logistik-Informationen unter allen Marktteilnehmern ausgetauscht werden müssen, benötigt. Grundlage dafür ist die EU-Verordnung 178/02 (insbesondere Artikel 18 und 19), welche die Notwendigkeit beschreibt, dass jeder Akteur der Lieferkette dafür verantwortlich ist, nachzuweisen von wem er seine Waren bezogen und an wen er seine Waren geliefert hat (Europa Parlament und Europäischer Rat, 2002).

Als konkretes Beispiel wird beim Praxispartner Westfleisch SCE mbH zur Realisierung einer Chargenrückverfolgung die Software Global Batch Traceability (GBT) vom Hersteller SAP eingesetzt. Mithilfe dieser Software werden die Stammdatenobjekte *Charge*, *Produkt* und *Geschäftspartner* verwaltet und mit dem Enterprise Resource Planning (ERP) System integriert. GBT ist dabei als zentrales System konzipiert, welches über eine Schnittstelle von Akteuren der Lieferkette mit Informationen zu einer *Charge* beliefert werden kann. Diese Schnittstelle verwendet *IDoc*[1] bzw. *XML*[2] als Austauschformat. Der eigentliche Austausch erfolgt dabei entweder manu-

[1] Ein Intermediate Document (IDoc) ist ein Container für den Datenaustausch zwischen SAP und Nicht-SAP-Systemen (SAP SE, 2019).

[2] Die Extensible Markup Language (XML) ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei (Yergeau et al., 2008).

ell über einen Dateiimport/-export Mechanismus oder über das Internet mittels des *HTTP*[3] Protokolls. Bei diesem Austausch besteht grundsätzlich die Möglichkeit, dass Datensätze vor dem Austausch oder nachträglich verändert werden können - ohne das Teilnehmer der Lieferkette hiervon etwas mitbekommen würden.

Aus den beschriebenen Sachverhalten ergibt sich für eine zeitnahe und transparente Rückverfolgung von *Chargen* über den gesamten Verlauf der Wertschöpfungskette in Produktionsnetzwerken mittels *Blockchain-Technologie* folgende Forschungsfrage:

FF1 Wie kann die Rückverfolgbarkeit von *Chargen* in der Fleischwarenindustrie entlang der gesamten Lieferkette mithilfe von *Blockchain-Technologie* realisiert werden?

- FF1.1 Welche Anforderungen an ein System zur Rückverfolgbarkeit von *Chargen* werden seitens der Fleischwarenindustrie gestellt?
- FF1.2 Welche Daten müssen in einer *Blockchain* persistiert werden, um eine Rückverfolgbarkeit zu ermöglichen?
- FF1.3 Welche *Blockchain-Technologie* kommt in Frage um FF1 zu realisieren und den spezifischen Anforderungen der Fleischwarenindustrie gerecht zu werden?
- FF1.4 Welche Systemarchitektur erfüllt die Anforderungen der Fleischwarenindustrie, um eine Chargenrückverfolgung zu realisieren?

1.3. Vorgehen / Methodik

Die in Abschnitt 1.2 beschriebenen Probleme und Herausforderungen sollen gelöst werden mittels der Design Science Methode nach Hevner (2007); Hevner et al. (2004). Dabei konzentriert sich Design Science auf die Entwicklung von (entworfenen) Artefakten mit der Absicht, die funktionale Leistung des Artefakts zu verbessern. Design Science wird in der Regel für Artefakte aus den Kategorien Algorithmen, Mensch-Computer-Schnittstellen und Prozessmodellen verwendet (Kuechler and Vaishnavi, 2008; Peffers et al., 2012). Abbildung 2 stellt die drei Design Science Zyklen nach Hevner (2010) dar.

[3] Hypertext Transfer Protocol (HTTP)

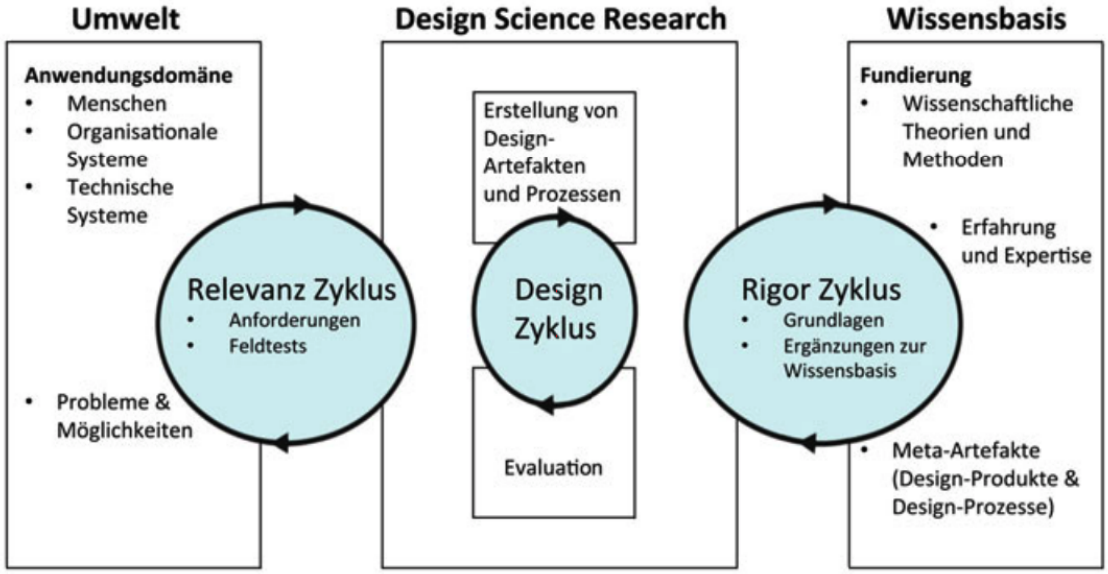

Abbildung 2: Die drei Design Science Zyklen nach Hevner (2010) (Trepper, 2015)

Im Sinne des Relevanz Zyklus (siehe auch Simon, 1996) soll eine Betrachtung der bisherigen Supply Chain Systeme und der Wertschöpfungskette inklusive ihrer einzelnen Geschäftsprozesse aus technischer Sicht erfolgen. Als Ergebnis dieser Betrachtung sollen Anforderungen an das Artefakt identifiziert werden. Anschließend wird durch den Rigor Zyklus eine wisschenschaftliche Basis erarbeitet, um bereits vorhandene Erkenntnisse in die Arbeit einfließen zu lassen. Durch den Rigor Zyklus soll sichergestellt werden, dass das Artefakt eine Innovation darstellt und nicht bereits erforschte Resultate repliziert werden (Hevner, 2010). Innerhalb des Design Zyklus soll ein möglicher Systementwurf zur Lösung der Probleme aus Abschnitt 1.2 erarbeitet werden. Dieser Systementwurf wird als Prototyp implementiert und anschließend einer Evaluation durch Experteninterviews (siehe auch Wilde and Hess, 2007) unterzogen.

1.4. Ziele

Der Einsatz von *Blockchain-Technologie* könnte - für die in Kapitel 1.2 beschriebene Problemstellung - eine Lösung darstellen. Eine *Blockchain* ist ein dezentrales System zur manipulationssicheren Speicherung von Informationen in sog. *Blöcken*

die untereinander durch kryptographische Methoden verkettet sind - daher auch der Name *Blockchain*. Eine *Blockchain* verwendet verschiedenste Verfahren zur Konsensbildung innerhalb des Netzwerks, um sicherzustellen das neue *Blöcke* und die darin enthaltenen Transaktionen vom gesamten Netzwerk validiert und verifizert werden bevor der *Block* in die *Blockchain* geschrieben wird (siehe auch Buterin, 2014; Cardano, 2017; carVertical, 2017; Nakamoto, 2009).

Außerdem kann eine *Blockchain* durch den Einsatz einer kryptographischen *Hashfunktion*[4] zur Bildung einer Prüfsumme für jeden *Block* innerhalb der *Blockchain* sicherstellen, dass bereits persistierte Informationen nicht ohne weiteres manipuliert werden können. Im Idealfall ist eine *Blockchain* dezentral konzipiert, was bedeutet, das jeder Teilnehmer eines *Blockchain* Netzwerks eine exakte Kopie des Datenbestands lokal vorhält. Hierdurch soll sichergestellt werden, das auch bei einem Ausfall oder einer Kompromittierung einzelner Teilnehmer das Gesamtsystem weiterhin in seiner Funktion stabil bleibt (Drescher, 2017; Tribis et al., 2018).

Ziel dieser Arbeit ist es, durch Entwicklung und Evaluation eines Prototyps die Möglichkeiten und Grenzen der *Blockchain-Technologie* im Kontext der Chargenrückverfolgung in der Fleischwarenindustrie zu überprüfen. Dabei sollen die dafür nötigen Daten und Informationen ermittelt und in einen Systementwurf eingearbeitet werden. Außerdem ist angestrebt, aus der Vielzahl von unterschiedlichen Implementierungen einer *Blockchain* genau die Ausprägung zu identifizieren, welche für die spezifischen Anforderungen der Fleischwarenindustrie ideal erscheint.

Konkret lassen sich hieraus folgende Ziele und erwartete Ergebnisstypen zu den jeweiligen Forschungsfragen aus Kapitel 1.2 ableiten:

- Identifikation verwandter Arbeiten aus Wissenschaft und Praxis für FF1.1
- Anforderungserhebung und -analyse mit dem Praxispartner für FF1.1
  - Funktional
  - Qualitativ

[4]Spezielle Form einer Hashfunktion, welche kollisionsresistent ist. Es ist praktisch nicht möglich, zwei unterschiedliche Eingabewerte zu finden, die einen identischen Hashwert ergeben (Menezes, 1997).

  - Rahmenbedingungen

- Prozessaufnahme und -analyse für FF1.2
  - Schwachstellenanalyse des *Ist*-Prozess
  - Modellierung eines *Soll*-Prozess bei Einsatz von *Blockchain-Technologie*
- SWOT-Analyse als Vorbereitung für eine Nutzwertanalyse zur Klärung von FF1.3
- Ableitung eines Systementwurfs mittels Design Science Research für FF1.4
- Entwicklung eines Prototyps anhand der Ergebnisse von FF1.1-4 für FF1
- Evaluation des Prototyps durch Experteninterview für FF1

Der enstandene Prototyp soll beim Praxispartner Westfleisch SCE mbH in Münster/Coesfeld als Entscheidungshilfe für eine zukünftige Innovationsstrategie zur Optimierung der Lieferkette dienen.

1.5. Struktur der Arbeit

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2. Verwandte Arbeiten

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.1. Thunfisch Traceability

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.2. Halal Food Chain

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.3. Fruchthändler

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est

Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3. Grundlagen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

3.1. Chargenrückverfolgung

Notwendigkeit einer Charge erläutern auf Grund der Gruppierung von vielen Einzelprodukten eben zu einer Charge.

3.1.1. Definition Charge

Eine *Charge* bezeichnet eine Ansammlung eines Produkts, welche unter gleichen Bedingungen produziert wurde. Bei dem Produkt kann es sich beispielsweise um Werkstoffe, Bauteile, Baugruppen oder Endprodukte handeln. Die Begriffe *Los* oder *Partie* werden oft als Synonym für *Charge* verwendet. Einige Branchen sind bei der Produktion auf die Erzeugung definierter *Chargen* zugeschnitten. Diese Chargenproduktion, die auch diskontinuierliche Produktion genannt wird, zeichnet sich durch einen zeitlich unterbrochenen Materialfluss aus. So kann ein Produktionsgefäß mit unterschiedlichen Rohstoffen befüllt und anschließend verarbeitet werden. In der diskontinuierlichen Produktion versteht man daher unter einer *Charge* eine Menge eines Erzeugnisses, welche in einem Produktionsgang gefertigt worden ist und identische Kennzeichen in Bezug auf Materialzusammensetzung, Fertigungsprozess und Produktqualität aufweist. Beispiele hierfür finden sich in der Stahlproduktion, der pharmazeutischen und chemischen sowie in der Lebensmittelindustrie (Günther and Tempelmeier, 2012).

Inzwischen wird der Begriff der *Charge* aber auch in der kontinuierlichen Produktion verwendet. Die *Charge* wird dabei durch die Berücksichtigung einer oder mehrerer der folgenden Eigenschaften charakterisiert:

- Herstellung auf einer Fertigungslinie,
- einheitliche Zulieferteile,

- homogene Qualität,
- gleichbleibende Prozesskette,
- identisches Produktionsdatum.

Es bleibt festzuhalten, dass die Parameter in der kontinuierlichen Produktion nicht so eindeutig abgrenzbar sind wie in der diskontinuierlichen Produktion. Zudem können in der kontinuierlichen Produktion Schwankungen durch dynamische Prozesse wie Abnutzung von Werkzeugen auftreten, die innerhalb einer definierten *Charge* zu deutlichen Qualitätsunterschieden führen können und so die Praxistauglichkeit der Chargenverfolgung in Frage stellen.

In der für die Lebensmittelindustrie wichtigen Los-Kennzeichnungs-Verordnung (LKV) wird unter einem *Los* „die Gesamtheit von Verkaufseinheiten eines Lebensmittels verstanden, das unter praktisch gleichen Bedingungen erzeugt, hergestellt oder verpackt wurde.“ (Bundesregierung, 1993). Dagegen bezeichnen laut Code of Federal Regulation *Los* oder *Charge* „ein oder mehrere Bauteile oder fertige Geräte eines einzigen Typs, Version, Klasse, Größe, Zusammensetzung oder Software Version, welche im wesentlichen unter gleichen Bedingungen hergestellt werden und die innerhalb spezifizierter Grenzen einheitliche Eigenschaften und Qualität haben sollen.“ (Food and Drug Administration, 1996). Somit können auch einzelne Produkte eine *Charge* oder ein *Los* bilden. Im Hinblick auf eine möglichst genaue Eingrenzung bestimmter Produkte beispielsweise bei einer Rückrufaktion sollte eine kleinstmögliche Chargengröße gewählt werden, die im Idealfall nur ein einzelnes Produkt umfasst.

3.1.2. Einordnung in die Wertschöpfungskette

Die Chargenverfolgung wird innerhalb des Produktionsprozesses für das Upstream Tracing und in dem Distributionsprozess für das Downstream Tracing eingesetzt. Bei einer gut organisierten Chargenverfolgung im Downstream Prozess behält der Hersteller den Überblick, wo seine Produkte wann gelagert, verkauft und eingesetzt werden und ist so in der Lage, gezielt Rückrufe durchzuführen. Durch die Chargenverfolgung im Upstream Prozess können eventuelle Qualitätsprobleme bis zum

Vorlieferanten nachverfolgt werden. Abbildung 3 zeigt schematisch die Wertschöpfungskette in der Lebensmittelindustrie. Bei einem optimal eingerichtetem Up- und Downstream Tracing behalten die Hersteller und Konsumenten während der ganzen Wertschöpfung einen Überblick wo sich die Waren aktuell im Einsatz befinden.

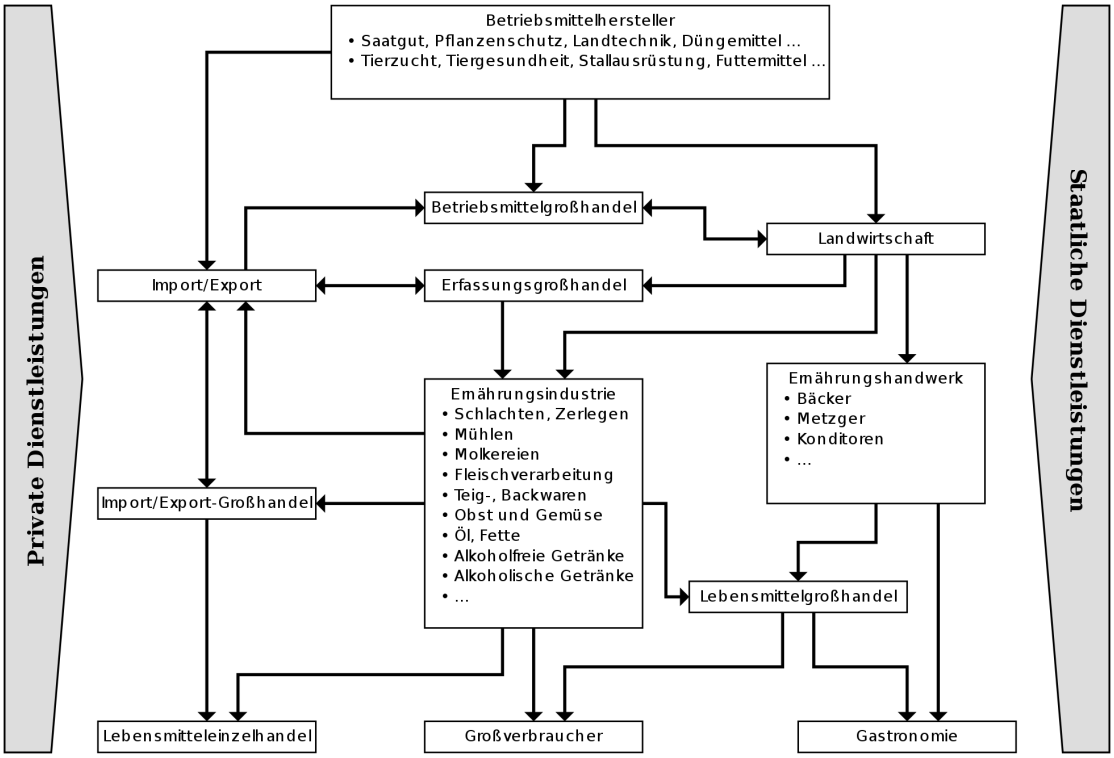

Abbildung 3: Wertschöpfungskette: Lebensmittelindustrie QUELLE

Downstream Tracing (Abwärts-Rückverfolgbarkeit)

Als Downstream Tracing wird die Rückverfolgbarkeit ausgehend vom Erzeuger zum Endprodukt bezeichnet. Gegenstand der Rückverfolgung ist typischerweise ein *Los* (*Charge*) oder eine einzelne Einheit eines Produkts. Abhängig vom Grad der Integration innerhalb der Lieferkette lässt sich die Rückverfolgung bis zum Einzelhandel bzw. auch bis zum Endverbraucher durchführen. Zum Einsatz kommt das Downstream Tracing wenn Probleme in Waren zu einem späten Zeitpunkt festgestellt wurden und geprüft werden muss in welchen Endproduktchargen sich hierdurch weitere Probleme ergeben könnten (Trienekens and Beulens, 2001; Zailani et al., 2010).

Wegner-Hambloch (2004) beschreibt Downstream Tracing als „Ortsbestimmung von bereits hergestellten Produkten zwecks nachträglichen Rückrufs von gesundheitsgefährdenden Produkten“.

Upstream Tracing (Aufwärts-Rückverfolgbarkeit)

Unter Upstream Tracing versteht man die Rückverfolgbarkeit vom Endverbraucher in Richtung des Erzeugers. Tritt ein Problem bei Lebensmittelprodukten auf wird das Upstream Tracing zur Ursachenforschung eingesetzt. So lassen sich Probleme die beispielsweise vom Konsumenten beim Endprodukt oder bei einer Qualitätskontrolle von Teilprodukten festgestellt wurden zurückverfolgen bis zum Urerzeuger (Trienekens and Beulens, 2001; Zailani et al., 2010). Nach Wegner-Hambloch (2004) ist Upstream Tracing „die Bestimmung der Produktgeschichte vom Endprodukt [...] bis zu den Futtermitteln.“

3.1.3. Zentrale vs. dezentrale Ansätze

Unterschied zwischen zentraler Informationssysteme (F-Trace) und dezentraler logischer Systeme (Zugriff auf F-Trace). Letzteres sind nur dem Anschein nach dezentral. Ihre zugrunde liegende Infrastruktur der Informationssysteme ist zentral und wird von einem Intermediär verwaltet und betrieben. Angriffspunkte für Manipulation und Kontrolle eines einzelnen rausarbeiten. (allgemeine fleischer zeitung, 2011; Steins, 2015) Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

3.1.4. Dokumentationspflichten

Für landwirtschaftliche Waren und daraus hergestellte Nahrungsmittel existieren eine Vielzahl von gesetzlichen Regelungen aus denen Bedingungen und Anforderungen zum Thema Rückverfolgbarkeit abgeleitet werden können. Die VO (EG) Nr. 178/02 (Europa Parlament und Europäischer Rat, 2002) wird in diesem Kontext als Basisverordnung gesehen. Darüber hinaus sind die horizontale Lebensmittelhygieneverordnung sowie die vertikalen Hygieneverordnungen für Fleisch und Fleischerzeugnise, Milch- und Milcherzeugnisse, Fisch und Fischerzeugnisse mit der Vorgabe zur Umsetzung betrieblicher Eigenkontrollen oder Einrichtung eines HACCP-Systems[5] elementare Bestandteile eines wirkungsvollen, innerbetrieblichen Rückverfolgungssystems in Lebensmittelbetrieben. Eine verbindliche fünfjährige Speicherung von Daten der Transaktionen bezüglich der Lieferanten und Abnehmer ist ebenfalls festgelegt.

Weitere Regelungen zur Rückverfolgbarkeit für die EU:

- Rindfleischetikettierungs-VO (EWG) Nr. 1760/2000
- EU-Öko-VO (EWG) 2092/91
- EU-Verordnung über amtliche Futter- und Lebensmittelkontrollen (Vorschlag vom 5. Februar 2003)
- Vermarktungsnormen für Eier 1907/90/EWG

Nationale Regelungen für Deutschland:

- Lebensmittelkennzeichnungsverordnung (LMKV)
- Los-Kennzeichnungs-Verordnung (LKV)
- verschiedene Fleisch- und Geflügelfleisch-Hygienevorschriften
- Weingesetz und Weinwirtschaftsgesetz

[5] Englisch für *Hazard Analysis and Critical Control Points (HACCP)*. Beschreibt ein Qualitätskontrollsystem für den sicheren Umgang mit Lebensmitteln durch strukturierte und präventive Maßnahmen zur Verhinderung von Erkrankungen und Verletzungen des Konsumenten.(Europa Parlament und Europäischer Rat, 2004)

- Handelsklassenrecht
- Lebensmittel- und Bedarfsgegenständegesetz (LMBG)

Über die gesetzlichen Regelungen hinaus gelten verbindliche Standards der Handelsseite, die übergreifend von der Global Food Safety Initiative (GFSI) vorgegeben werden. Der in Deutschland meist gefragte International Food Standard (IFS), der Standard des British Retail Consortium (BRC) für Lieferanten nach England und diverse andere Standards definieren das detaillierte Anforderungsniveau transparenter Warenströme aus Handelssicht für den Hersteller.

3.1.5. ???Besonderheiten der Fleischwarenindustrie???

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2. *Blockchain-Technologie*

Beginnend mit einer allgemeinen Definition der Technologie wird in diesem Kapitel ein Grundverständnis des Aufbaus und der Funktionalität gegeben. Weiter werden die verschiedenen Begrifflichkeiten aus dem Umfeld der Distributed Ledger Technology (DLT) vorgestellt und untereinander abgegrenzt. Da konkrete Blockchain Systeme auf verschiedene Arten implementiert und umgesetzt werden soll eine Erläuterung der Kategorien klarheit schaffen. Abschließend wird ausführlich auf den technologischen Hintergrund der Technologie eingegangen.

3.2.1. Definition

Eine *Blockchain* als Ganzes betrachtet, ist ein System zur Transaktionsabwicklung mit besonderen Eigenschaften. Als erstes beschrieben wurde die *Blockchain* im Paper von Nakamoto (2009) zur Realisierung der digitalen Währung Bitcoin (BTC). Aus technischer Sicht gehört die *Blockchain-Technologie* zum Bereich der verteilten Datenbanken. Ein *Block* in einer *Blockchain* repräsentiert eine Menge von Datensätzen die in der *Blockchain* (Datenbank) vorgehalten werden. Jeder *Block* (Datensatz) widerrum besitzt genau einen Vorgänger und einen Nachfolger. Allerdings werden diese Blöcke nicht wie in klassischen relationalen Datenbanksystemen in Tabellenstrukturen abgelegt und verwaltet. Durch die im Block enthaltene Information des Vorgängerblocks wird jeder neue Datensatz immer an den letzten Datensatz angehangen. Daraus bildet sich eine Kette von Blöcken - daher der Name *Blockchain* (dt. Blockkette).

Ein *Block* innerhalb der Kette kann definiert werden als verschlüsseltes Stück Information. Er beinhaltet neben den Transaktionen noch einen Zeitstempel und zwei kryptographische Hashwerte. Der erste Hashwert wird aus dem *Block* selbst gebildet und der zweite Hashwert ist die Verknüpfung zum Vorgänger (Tschorsch and Scheuermann, 2016). Wird nachträglich ein Wert einer Transaktion verändert oder ein ganzer *Block* aus der Kette entfernt passt der jeweilige Hashwert des Vorgängers nicht mehr und durch den linearen Aufbau der *Blockchain* würde diese Manipulation jederzeit unmittelbar bemerkt werden bei der Validierung von neuen Transaktionen. Die Daten in der *Blockchain* sind somit vor unbefugter Veränderung geschützt. Als dezentrale Datenbank wird auf jedem Knoten des sich aufspannenden Netzwerks

aus Teilnehmern der *Blockchain* eine exakte Kopie[6] des Datenbestands vorgehalten. Diese dezentrale Struktur bedeutet, dass ein *Blockchain* Netzwerk nicht unter der Kontrolle oder Regulierung einer einzelnen Entität steht. Jeder Teilnehmer kann eigenständig im Netzwerk agieren und es ist kein Zwischenhändler nötig (Drescher, 2017; Meier and Stormer, 2018).

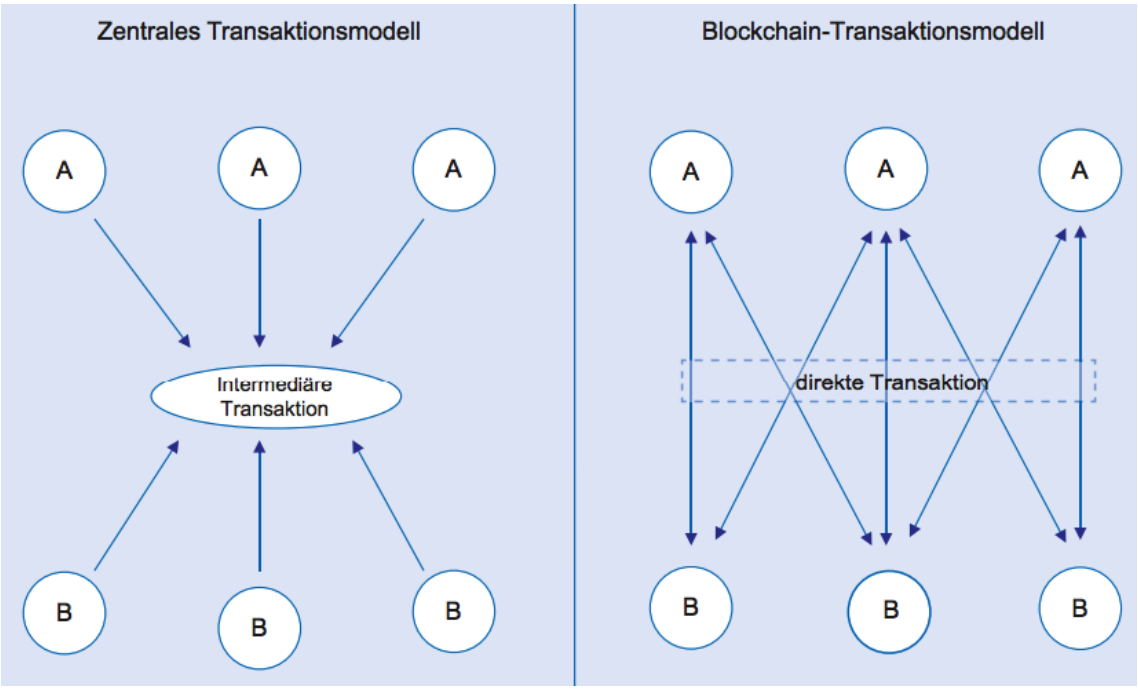

Abbildung 4: Transaktionsmodell Blockchain QUELLE

Wird von einem der Teilnehmer eine Transaktion ausgelöst, wird diese nicht durch einen Intermediär sondern durch das Netzwerk erfasst und verarbeitet (Abbildung 4). Ein neuer *Block* wird erschaffen und validiert wie es durch das Konsensprotokoll festgelegt wird. Dabei können solche *Blockchain* Systeme unterschiedlich ausgeprägt sein. Dies zeigt sich zb. an der Art des Zugriffs, also wer darf Transaktionen lesen, wer darf sie schreiben. Außerdem kann der Mechanismus zur Konsensfindung je System anders sein.

[6]Es gibt Ausprägungen von DLT Systemen bei denen sog. Light Nodes nur einen zeitlichen Abschnitt der Datensätze vorhalten, um neue Transaktionen validieren zu können. In der generellen Definition wird von sog. Full Nodes ausgegangen in denen stets alle Datensätze vorgehalten werden.

3.2.2. Begriffliche Abgrenzung

Die am häufigsten verwendeten Begriffe werden im Folgenden anhand eines Schichtenmodells (Abbildung 5) erklärt und voneinander abgegrenzt. Jede Schicht wird in der Abbildung durch einen Balken dargestellt und ist unabhängig von den darüber liegenden Schichten. Von oben nach unten gelesen stehen die Schichten in einer „ist enthalten in“ Beziehung zueinander. Entsprechend verlaufen die Schichten von einer konkreten Ausprägung zu einem abstrakten technologischen Konzept. Nachfolgend werden die einzelnen Schichten genauer erklärt.

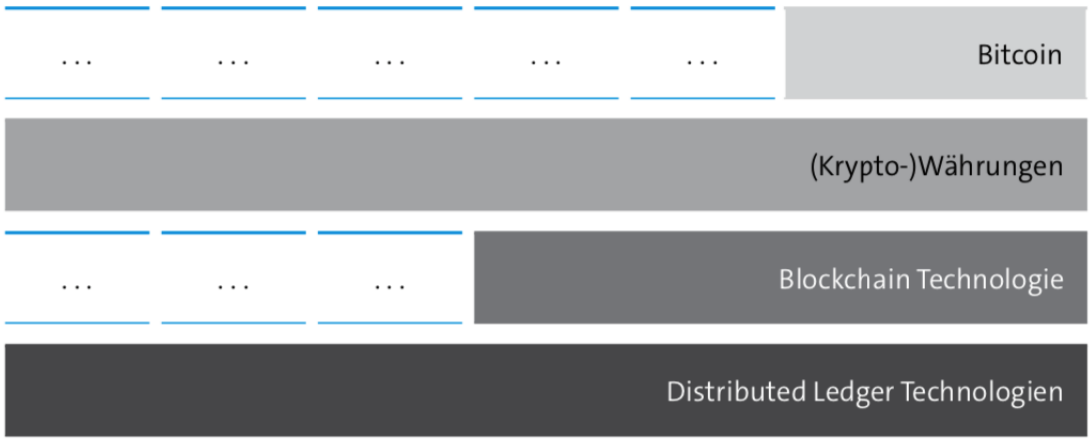

Abbildung 5: Schichtenmodell *Blockchain* Begriffe QUELLE

Distributed Ledger

Der *Distributed Ledger* bildet die Basis des Schichtenmodells. Er ist im Grunde genommen ein klassisches Betandsbuch, das über einen Mechanismus verfügt, es auf alle teilnehmenden Parteien zu verteilen. *Distributed Ledger* existieren bereits seit längerer Zeit und sind meist auf der technischen Basis einer verteilten Datenbank mit einer Logik auf Programm- oder Datenbankseite versehen, die aus der reinen Datenbank ein Bestandsbuch macht.

Distributed Ledger Technologie wird zunehmend synonym zum bisherigen Gebrauch von *Blockchain* genutzt, um die Entwicklungen nach dem Bitcoin und den Kryptowährungen von eben diesen begrifflich abzugrenzen.

Blockchain-Technologie

Die *Blockchain* ist eine Form, einen *Distributed Ledger* zu organisieren und zu implementieren. Auf die technische Implementierung der *Blockchain* wird in den folgenden Kapiteln näher eingegangen; zur Begriffsbestimmung seien hier die grundlegenden Eigenschaften aufgezählt, die der *Blockchain* in den letzten Jahren die steigende Aufmerksamkeit ermöglich haben:

- Dezentralisiert
- Peer-to-Peer
- Transparenz und Anonymität
- Vertrauen

Blockchain gehört zu den bekanntesten Distributed-Ledger-Technologien. Aus diesem Grund wird die Bezeichnung Blockchain-Technologie in dieser Arbeit synonym für Distributed-Ledger-Technologien benutzt. Auf die technischen Eigenschaften von weiteren Ausprägungen der Distributed-Ledger-Technologien wird in dieser Arbeit daher nicht eingegangen.

Kryptowährungen

Mit der *Blockchain* als Basistechnologie lassen sich darauf aufbauende komplexe Systeme, wie z.B. Währungen abbilden. Wie in Kapitel 3.2.1 erwähnt wurde die Blockchain-Technologie als erstes im Zusammenhang mit einer Kryptowährungen, dem Bitcoin, beschrieben. Die *Blockchain* ist somit ein Nebenprodukt einer technischen Plattform, die eine kryptographische Währung erschuf und gleichzeitig ein System implementierte, um diese Währung zu nutzen und zu handeln.

Neben dem Bitcoin existiert eine Reihe weiterer Kryptowährungen, die sich zum Teil der dem Bitcoin zugrunde liegenden öffentlichen *Blockchain* bedienen. Genannt seien hier z.B. Litecoin oder Dogecoin. Es existieren darüber hinaus Kryptowährungen, die eigene Blockchains zur Basis haben - zum Teil auf einer komplett eigenen technischen Implementierung. Vertreter hierfür sind z.B. Ethereum, Ripple oder Iota (siehe auch Buterin, 2014; carVertical, 2017; J.P.Morgan, 2017).

Bitcoin

Der Bitcoin ist die Kryptowährung, die auf der ursprünglichen *Blockchain* gehandelt wird. Im Rahmen dieser Arbeit wird der Bitcoin und andere Kryptowährungen nicht weiter betrachtet.

3.2.3. Arten von *Blockchain*

Bei der Auswahl der Art einer *Blockchain* trifft man auf zwei Widersprüche die nachfolgend kurz erläutert sind. Darauf folgt eine Betrachtung der Konfliktursachen und die sich daraus ableitenden Kategorien in die sich ein Blockchain System einordnen lässt.

Transparenz vs. Vertraulichkeit

Verwendet man eine *Blockchain* werden Besitzverhältnisse durch die Transaktionshistorie ermittelt. Dabei lässt sie eine *Blockchain* mit einem öffentlichen Register vergleichen. Im Sinne der Übertragung von Eigentum sind Offenheit und Transparenz zwei wesentliche Eigenschaften der Blockchain. Durch diese Offenheit ist jeder Teilnehmer in der Lage alle Transaktionen einzusehen und auf Manipulationen zu prüfen.

Dieses Vorgehen steht im Gegensatz zur Vertraulichkeit, die in bestimmten Bereichen unabdingbar ist. Durch Vertraulichkeit werden Informationen wie die Transaktionsdaten oder deren Details (beteiligte Konten oder transferierte Menge) vor unbefugter Einsicht geschützt. Hierdurch entsteht der Widerspruch zwischen Transparenz auf der einen Seite und Anforderungen an die Vertraulichkeit auf der anderen Seite (Drescher, 2017).

Sicherheit vs. Geschwindigkeit

Die Datenstruktur einer *Blockchain* sichert die Transaktionshistorie vor Manipulationen und Fälschungen. Jeder neue *Block* der in der *Blockchain* gespeichert werden soll muss vom Netzwerk durch das Lösen einer kryptographischen Aufgabe erzeugt und der Datenstruktur hinzugefügt werden. Dadurch ist es ziemlich aufwendig die Transaktionshistorie nachträglich zu manipulieren oder zu fälschen. Durch diesen Sicherheitsmechanismus sinkt die Geschwindigkeit mit der ein *Blockchain* Netzwerk

neue Transaktionen verarbeiten kann. Moderne Applikationen erfordern Geschwindigkeit und Skalierbarkeit was im direkten Kontrast zum erwähnten Sicherheitskonzept einer *Blockchain* steht (Drescher, 2017).

Ursachen der Konflikte

Zwei grundlegende Operationen eines *Blockchain* Netzwerks sind Ursache für die beiden beschriebenen Widersprüche - Schreiben und Lesen von Transaktionsdaten. Der Konflikt zwischen Transparenz und Vertraulichkeit ist auf die Lese-Operationen einer *Blockchain* zurückzuführen. Je offener die Leseberechtigungen einer *Blockchain* sind, desto höher ist die Transparenz und desto niedriger ist die Vertraulichkeit der Transaktionsdaten. Die Schreib-Operationen sind für den Widerspruch zwischen Sicherheit und Geschwindigkeit verantwortlich. Je restriktiver die Berechtigungen zum Schreiben innerhalb des *Blockchain* Netzwerks sind, desto höher ist die Geschwindigkeit mit der Transaktionen verarbeitet werden können. In Tabelle 1 werden die technischen Beschränkungen, der Widerspruch und die Operation innerhalb der *Blockchain* zusammengefasst (Drescher, 2017).

Beschränkung	**Widerspruch**	**Blockchain Operation**
Keine Vertraulichkeit	Transparenz vs. Vertraulichkeit	Transaktionshistorie lesen
Skalierbarkeit	Sicherheit vs. Geschwindigkeit	Transaktionen schreiben

Tabelle 1: Technische Beschränkungen der *Blockchain* und ihre Ursachen

Public vs. Private

Betrachtet man die Berechtigungen zum Lesen innerhalb eines *Blockchain* Netzwerks in der einfachsten Form muss das System zwischen Transparenz und Vertraulichkeit entscheiden. Entweder es werden allen Teilnehmern Leseberechtigungen zugeteilt oder nur einer ausgewählten Gruppe von Teilnehmern. Anhand des Kriterium, welcher Teilnehmer im Netzwerk neue Transaktionen erstellen und die Historie lesen kann, lässt sich eine *Blockchain* als öffentliche oder private *Blockchain* charakterisieren (Drescher, 2017).

Permissioned vs. Permissionless

Die Schreibrechte bestimmen für ein *Blockchain* Netzwerk den Grad der Skalierbarkeit. Werden Schreibrechte in ihrer einfachsten Form zugeteilt und alle Teilnehmer sind berechtigt Schreib-Operationen auszuführen, erhöht sich der Arbeitsaufwand je Teilnehmer der zur Berechnung nötigt wird. Dies ist für die Sicherheit des Netzwerk positiv, wirkt sich aber negativ auf die Geschwindigkeit aus. Durch die Geschwindigkeit wird das Netzwerk in der Skalierbarkeit beschränkt. Teilt man hingegen nur einer Gruppe von Teilnehmern Schreibrechte zu, ist der Arbeitsaufwand im Vergleich niedrig. Hierdurch kann das Netzwerk Transaktionen vergleichsweise schnell verarbeiten und ist dadurch selbst skalierbarer (Drescher, 2017).

	Permissionless	**Permissioned**
Public	Bitcoin, Ethereum, IOTA	Ethereum 2.0
	Jeder kann validieren	Ausgewählte Gruppe kann validieren
	Jeder kann teilnehmen	Jeder kann teilnehmen
Consortium/Private	Interplanetary Database(IPDB)	Hyperledger, Quorum
	Jeder kann validieren	Ausgewählte Gruppe kann validieren
	Ausgewählte Gruppe kann teilnehmen	Ausgewählte Gruppe kann teilnehmen

Tabelle 2: Arten von Blockchain Netzwerken (eigene Darstellung)

Alle zuvor beschriebenen Eigenschaft einer Blockchain ermöglichen es eine Matrix mit zwei Dimensionen zu modellieren in die sich nahezu sämtliche Blockchain Lösungen einordnen lassen. Ausgenommen sind etwaige Mischformen, die für sehr spezielle Anwendungsfälle konzipiert wurden und sich beispielsweise aus einer Kombination einer öffentlichen und konsortialen Blockchain zusammensetzen. Tabelle 2 zeigt diese Matrix. Die vertikale Achse beschreibt in diesem Fall die Anonymität der Teilnehmer. Diese reicht von vollständiger Anonymität[7] bis zur Offenlegung und direkten Verknüpfung zwischen einem Teilnehmer des Netzwerks und einer Entität (Person, Maschine oder Unternehmen) in der realen Welt. Auf der horizontalen Achse wird das Vertrauen in die Validatoren abgebildet. Konkret können entweder alle Teilneh-

[7] Anonymität meint hier eine Pseudo-Anonymität, da aus technischer Sicht mit einigem Aufwand der Teilnehmen klar identifiziert werden kann.

mer auch als Validatoren auftreten (Permissionless) oder es wird eine Gruppe von Teilnehmern zum validieren von Transaktionen gebildet, die definierte Anforderungen erfüllen (Permissioned). An den Schnittpunkten der Zeilen und Spalten wurden Beispiele für Implementationen der jeweiligen Kombination eingefügt.

3.2.4. Peer-to-Peer Netzwerke

Ein *Peer-to-Peer* Netzwerk ist der Gegensatz zum klassischen *Client-Server-Modell*, bei dem ein *Server* einen Dienst zur Verfügung stellt und ein oder mehrere *Clients* diesen Dienst abrufen und nutzen. Bei einem *Peer-to-Peer* Netz sind alle Teilnehmer, die sog. *Peers*, gleichberechtigt und können Dienste anbieten und auch konsumieren. *Peer-to-Peer* Netzwerke operieren als *Overlay-Netze*[8] auf dem Internet. Einige der häufigsten Eigenschaften von *Peer-to-Peer* Netzwerken sind nach Steinmetz and Wehrle (2005):

- Heterogenität zwischen den *Peers* in Bezug auf Bandbreite, Rechenkraft und Speichergröße
- Qualität einzelner *Peers* in Form von Verfügbarkeit und Verbindungsstärke lässt sich nicht voraussetzen
- Client-Server-Funktionalität wird für *Peers* ermöglicht, um Dienste und Ressourcen anzubieten und zu konsumieren
- Austausch von Diensten und Ressourcen unter allen *Peers* gewährleistet
- Bereitstellung von Such-Funktionen durch ein zusätzliches *Overlay-Netz*
- Autonomie der *Peers* in punkto Ressourcenbereitstellung
- Das *Peer-to-Peer* Netzwerk organisiert sich selbst und nicht durch Dritte

[8] Ein *Overlay-Netz* baut auf ein bestehendes Netz (*Underlay Netz*) auf. Es kann mit eigenen Protokollen arbeiten und selbst als *Underlay Netz* fungieren.(Andersen et al., 2001)

3.2.5. Kryptografisches Hashing

Kryptografisches Hashing gehört zu einem der wichtigsten Instrumente der Kryptografie und bildet einen eigenen Teilbereich der Kryptografie. Mit einer kryptografischen Hash-Funktion lässt sich aus einem beliebig langen Wort (oder Datensatz) eine Zeichenkette mit fixer Stellenanzahl generieren. Die jeweilige Ausgabelänge wird in Bit angegeben. Formal ist eine Hash-Funktion definiert als

$$f : \{0,1\}^* \mapsto \{0,1\}^n \tag{1}$$

Das Ergebnis wird als digitaler Fingerabdruck bezeichnet. Die Generierung des Hash-Werts ist nicht zwingend kryptografisch, denn nicht jede Hash-Funktion erfüllt alle Anforderungen einer kryptografischen Hash-Funktion (Diffie, 1976; Menezes, 1997). Dabei gilt, eine kryptografische Hash-Funktion muss folgende Kriterien erfüllen:

- Eindeutigkeit
- Reversibilität
- Kollisionsresistenz

Mit der Eindeutigkeit ist gegeben, dass ein bestimmter Eingabewert immer zum selben Ausgabewert führt. Reversibilität beschreibt die Eigenschaft einer Hash-Funktion, das der Ausgabewert nicht in den ursprünglichen Eingabewert zurückberechnet werden kann. Die Kollisionsresistenz sorgt dafür, dass zwei unterschiedliche Eingabewerte nicht den gleichen Ausgabewert erzeugen. Abbildung 6 zeigt schematisch die Funktionsweise einer kryptografischen Hash-Funktion. Der Eingabewert, hier Urbild, wird durch die kryptografische Hash-Funktion in einen Ausgabewert (Hash-Wert) fester Länge transformiert.

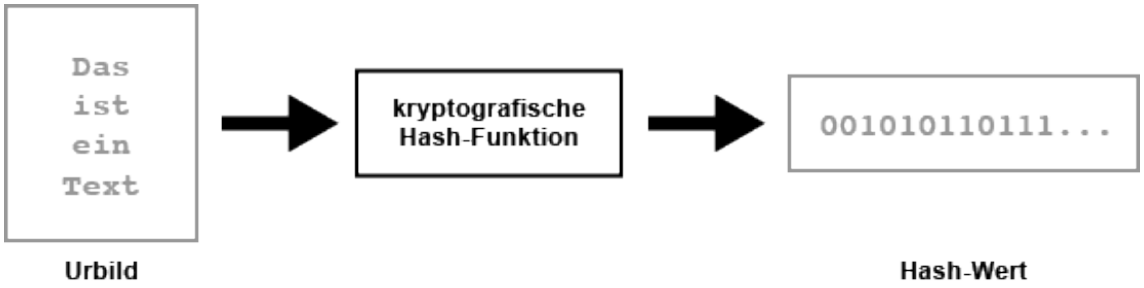

Abbildung 6: Funktionsweise einer kryptografischen Hash-Funktion QUELLE Elektronik Kompendium

3.2.6. Signierte Transaktionen durch Public-Key-Infrastruktur

Wird eine Transaktion von einem Teilnehmer erstellt und soll durch das Netzwerk validiert werden kommen digitale Signaturen zum Einsatz. Digitale Signaturen gehören zur asymmetrischen Kryptographie und werden dazu verwendet die Urheberschaft und Integrität einer Nachricht oder, im Falle der Blockchain, einer Transaktion zu prüfen.(Beutelspacher et al., 2010; Menezes, 1997)

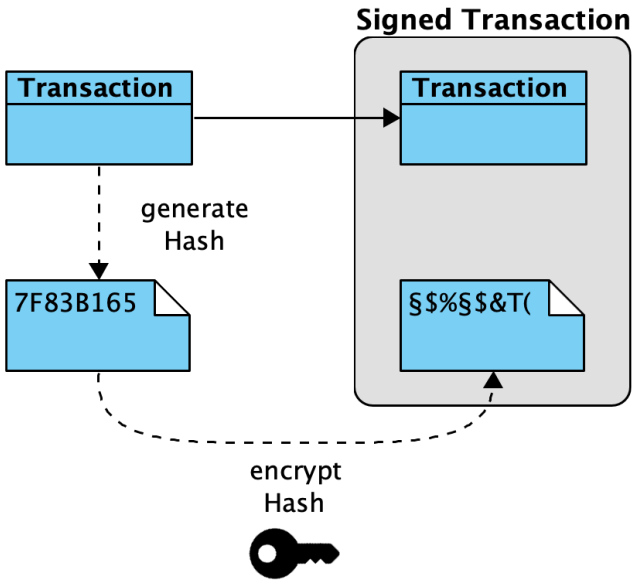

Abbildung 7: Schematische Darstellung für das Erstellen einer digitalen Signatur (in Anlehnung an Drescher (2017))

In Abbildung 7 wird das digitale Signieren einer Transaktion verdeutlicht. Der Prozess startet oben links in der Abbildung mit einer Transaktion. Durch Anwendung einer kryptographischen Hashfunktion wird ein Hash gebildet. Dieser Hashwert wird anschließend mit dem privaten Schlüssel des Erstellers verschlüsselt. Dieser verschlüsselte Hashwert ist die digitale Signatur und zusammen mit der Transaktion bilden sie die digital signierte Transaktion. Durch die Verwendung der Public-Key-Infrastructure (PKI) ist die digitale Signatur auf zwei Arten einzigartig. Zum einen kann der Ersteller der Signatur eindeutig zugeordnet werden und zum anderen wird die Integrität der Transaktion sichergestellt (Drescher, 2017).

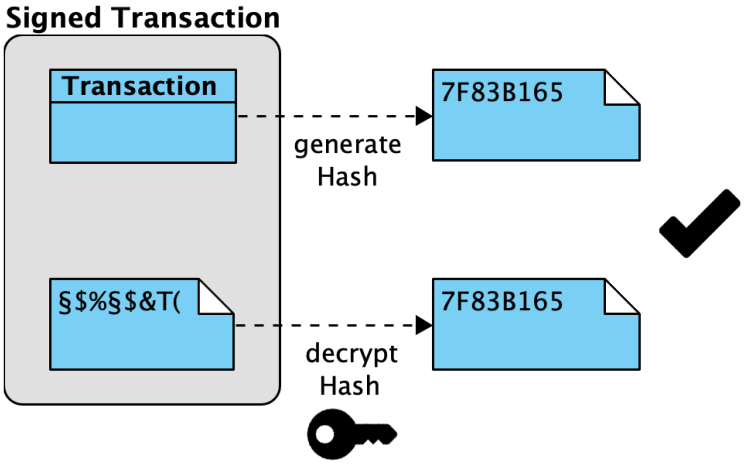

Abbildung 8: Erfolgreiche Prüfung einer digitalen Signatur (in Anlehnung an Drescher (2017))

Soll eine signierte Transaktion vom Netzwerk verarbeitet und erfolgreich verbucht werden müssen zwei Eigenschaften erfüllt sein. Die Urheberschaft muss eindeutig zuzuordnen sein und die Integrität der Transaktion darf nicht verletzt worden sein. Dazu wird wie in Abbildung 8 zuerst mit dem öffentlichen Schlüssel des Absenders die digitale Signatur entschlüsselt. Gelingt dies, ist sichergestellt das der Ersteller der digitalen Signatur eindeutig über die PKI zugeordnet werden kann. Im zweiten Schritt wird aus der Transaktion der Hashwert gebildet und mit der entschlüsselten digitalen Signatur verglichen. Sind beide Werte gleich ist garantiert, dass die Transaktion auf dem Weg der Übermittlung nicht manipuliert wurde (Drescher, 2017).

Stellt sich bei der Überprüfung der signierten Transaktion heraus, dass die Has-

hwerte nicht übereinstimmen können zwei Gründe dafür verantwortlich sein. Entweder wurde die eigentliche Transaktion während der Übermittlung von einem Angreifer manipuliert oder die Transaktion wurde nicht vom vermeintlichen Teilnehmer des Netzwerks autorisiert (Drescher, 2017). Abbildung 9 zeigt schematisch diese Situation.

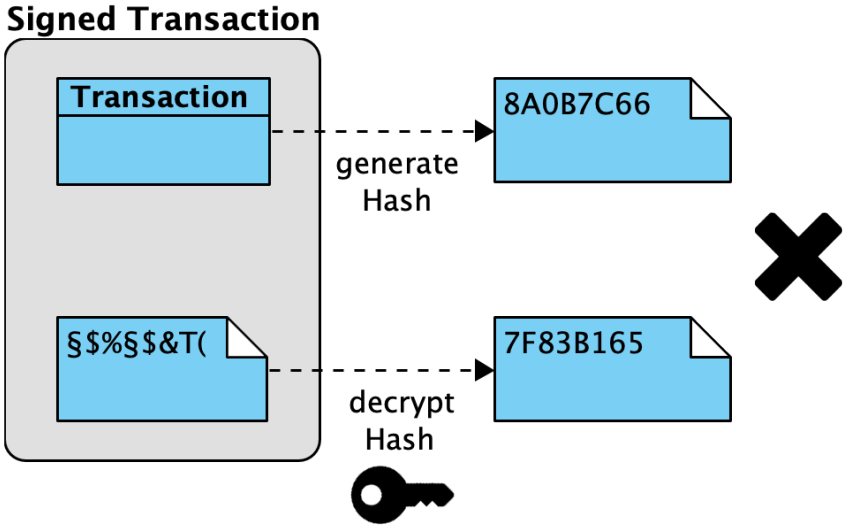

Abbildung 9: Erkennung von Manipulation anhand der digitalen Signatur (in Anlehnung an Drescher (2017))

3.2.7. Konsensmechanismen

Es gibt hauptsächlich zwei Kategorien von Konsensmechanismen:

- Lotterie-basiert
- Byzantinische Fehlervereinbarung

Die erste Kategorie wird auch Nakamoto-Konsens genannt nach dem Pseudonym des Bitcoin Erfinders Satoshi Nakamoto. Der Konsensmechanismus wählt den Prüfer, d.h. den Knoten, der entscheidet, welcher der nächste Block ist, der an die Blockchain angehangen wird. Dabei ist die Wahl eine Lotterieziehung. Der Gewinner ist der Validierer. Jeder neue Block erfordert auch eine neue Ziehung eines Validierers.

Die Auswahl durch eine Lotterie reduziert die Wahrscheinlichkeit, dass ein kompromittierter Knoten einen gefälschten Block validiert. Hierbei folgt die Lotterie keiner gleichwertigen Verteilung. Jeder Mechanismus definiert seine eigene Wahrscheinlichkeitsverteilung anhand eine bestimmte Eigenschaft des Gewinners bevorzugt wird. So besitzt jeder Lotterie-basierte Konsensmechanismus ein anderes Vertrauensmodell. Bitcoin beispielsweise verwendet den bekanntesten Mechanismus - Proof-of-Work (PoW). Daneben gibt es wie beschrieben noch einige andere Mechanismen wie Proof-of-Stake (PoS), Proof-of-Space (PoSp) oder Proof-of-Elapsed-Time (PoET).

Byzantine Fault Tolerant (BFT)-Systeme bilden die Basis für Mechanismen der zweiten Kategorie. BFT-Systeme sind so konzipiert, dass sie auch bei Ausfall einiger Teilnehmer des Netzwerks weiterhin funktionieren. Dabei kann der Ausfall unfreiwillig (z.B. ein teilnehmender Knoten ist außer Betrieb) oder freiwillig (z.B. ein Angreifer kontrolliert den fehlerhaften Knoten) sein. BFT-Systeme verwenden Abstimmungsmechanismen, um einen Konsens herstellen zu können. Der verwendete Mechanismus legt das Vertrauensmodell fest. Der Practical Byzantine Fault Tolerant (pBFT) Mechanismus ist der bekannteste Mechanismus dieser Kategorie. Außerdem sind hybride Konsensmechanismen möglich die eine Mischung aus Lotterie und BFT darstellen. Nachfolgend sollen die beiden meist verwendeten Konsensmechanismen kurz erläutert werden.

Proof-of-Work

Das Konzept des Proof-of-Work (PoW) existierte schon vor der ersten Blockchain Applikation (Bitcoin). Die erste moderne Anwendung wurde 1996 von Adam Back unter dem Namen „Hashcash“ eingereicht. Diese Anwendung hat auf Grundlage des SHA265-Algorithmus einen PoW Mechanismus eingesetzt um E-Mail Spam zu verhindern (Back, 2002).

Der Mechanismus des PoW kann relativ simpel beschrieben werden. Es ist die Tatsache, dass ein Teilnehmer des Netzwerks allen anderen Teilnehmern das Ergebnis der von ihm durchgeführten Berechnungen vorlegt. Die durchzuführenden Operationen sind an sich nicht kompliziert, allerdings müssen sie so oft durchgeführt werden, dass der Teilnemer eine erhebliche Rechenleistung dafür aufbringen muss. Daher spricht man von „Proof-of-Work“, da der Teilnehmer mit einem korrekten Ergebnis einen Nachweis seiner geleisteten Arbeit gibt. Konkret muss der Teilnehmer ein

Ergebnis finden, das mit einer bestimmten Anzahl an führenden Nullen beginnt. Je größer die Anzahl der führenden Nullen ist, desto schwierieger ist es für den Teilnehmer ein valides Ergebnis zu finden. Die Anzahl der Nullen bzw. die Schwierigkeit wird an die Anzahl der Teilnehmer und ihrer Rechenleistung im Netzwerk angepasst, sodass ein neues Ergebnis in festen Intervallen gefunden werden kann.[9] Für die Berechnung des Ergebnisses fügt der Teilnehmer zu den eigentlichen Transaktionsdaten eine sogenannte „Nonce“ hinzu. Aus diesen Daten versucht der Teilnehmer das Ergebnis zu berechnen mit der entsprechenden Anzahl an führenden Nullen. Bei jeder Runde wird die „Nonce“ verändert. Dies wird solange durchgeführt bist das Ergebnis zur aktuellen Schwierigkeit im Netzwerk passt.

Practical Byzantine Fault Tolerance

Das pBFT-Modell konzentriert sich in erster Linie auf die Bereitstellung einer Zustandsmaschine, die byzantinische Fehler (kompromittierte Knoten oder Netzwerkteilnehmer) toleriert. Dies geschiet durch die Annahme, dass es unabhängige Knotenausfälle und manipulierte Nachrichten gibt. Der Algorithmus wurde für den Einsatz in asynchronen Systemen konzipiert und optimiert auf hohe Performance. Im Wesentlichen sind alle Knoten im pBFT-Modell in Reihe angeordnet, wobei ein Knoten als Primärknoten und die restlichen Knoten als Backupknoten bezeichnet werden. Alle Knoten innerhalb des Systems kommunizieren untereinander mit dem Ziel einen einheitlichen Zustand des Systems zu finden. Die Knoten müssen dabei nachweisen, das eine Nachricht von ihnen stammt und das diese Nachricht während der Übertragung nicht manipuliert wurde (Dinh et al., 2017).

Damit das pBFT-Modell funktioniert, wird davon ausgegangen, dass die Anzahl der kompromittierten Knoten im Netzwerk nicht größer oder gleich 1/3 der Gesamtanzahl an Knoten im Netzwerk ist. Je mehr Knoten das Netzwerk bilden, desto mathematisch unwahrscheinlicher ist es, dass eine Anzahl von Knoten die sich 1/3 der Gesamtknotenanzahl nähert kompromittiert ist.

Jede Runde des pBFT-Konsens, genannt *Views*, besteht aus 4 Phasen. Das Modell folgt dabei eher dem Format „Kommandant und Offiziere“ durch die Anwesenheit des Primärknotens. Beim byzantinischen Generalsproblem sind alle Generäle gleich-

[9] Im Bitcoin Blockchain Netzwerk wird die Schwierigkeit dauerhaft so angepasst, dass nur alle 10 Minuten ein neuer Block berechnet werden kann.

wertig, was hier nicht der Fall ist. Die Phasen des pBFT-Konsens sehen wie folgt aus.

1. Ein Client sendet eine Anfrage an den Primärknoten, um eine Serviceoperation durchzuführen.
2. Der Primärknoten sendet die Anfrage an alle Backupknoten.
3. Die Knoten führen die Anfrage aus und senden eine Antwort an den Client.
4. Der Client erwartet $3f + 1$ Antworten von verschiedenen Knoten mit dem gleichen Ergebnis.[10] Das Ergebnis ist das Ergebnis der Serviceoperation.

Alle Knoten müssen die Anforderung erfüllen deterministisch zu operieren und im gleichen Zustand mit der Operation zu beginnen. Das Endergebnis ist, das sich alle nicht-kompromittierten Knoten auf die Reihenfolge der Datensätze einigen und dies geschlossen akzeptieren oder ablehnen.

Der Primärknoten wird in jeder *View* nach dem Round-Robin Verfahren ausgewählt und kann auch ausgetauscht wurde durch eine Erweiterung des Modells. Ein Austausch kann durchgeführt werden, wenn der Primärknoten die Anfrage nicht innerhalb eines bestimmten Zeitlimits an die Backupknoten weiterleitet.

Quellen ergänzen!

[10] Mit f ist die Anzahl an tollerierbaren kompromittierten Knoten gemeint.

4. Lösungskonzept

Dieses Kapitel soll aufzeigen mit welcher konkreten Ausprägung der Blockchain Techologie der gewählte Use-Case realisiert werden kann. Dazu wird im ersten Schritt eine SWOT-Analyse zur Blockchain Technologie allgemein durchgeführt und die Ergebnise beschrieben. In Schritt zwei kommt eine Nutzwertanalyse zum Einsatz anhand welcher ermittelt wird welche Ausprägung der Technologie sich zur Umsetzung bestmöglichst eignet.

4.1. SWOT-Analyse der *Blockchain-Technologie*

Durch die Vielzahl an unterschiedlichen Use-Cases die mittels der Blockchain Technologie umgesetzt werden ist es nötig für den spezifischen Use-Case der Chargenrückverfolgung die Technologie einer SWOT-Analyse zu unterziehen. Hierdurch wird gewährleistet, dass die Technologie für den Use-Case überhaupt geeignet ist. Im folgenden werden daher aus interner Sicht die Stärken und Schwächen gegenübergestellt, sowie die dadurch möglichen externen Chancen und Risiken diskutiert. Abbildung 10 zeigt eine schematische Sicht der SWOT-Analyse.

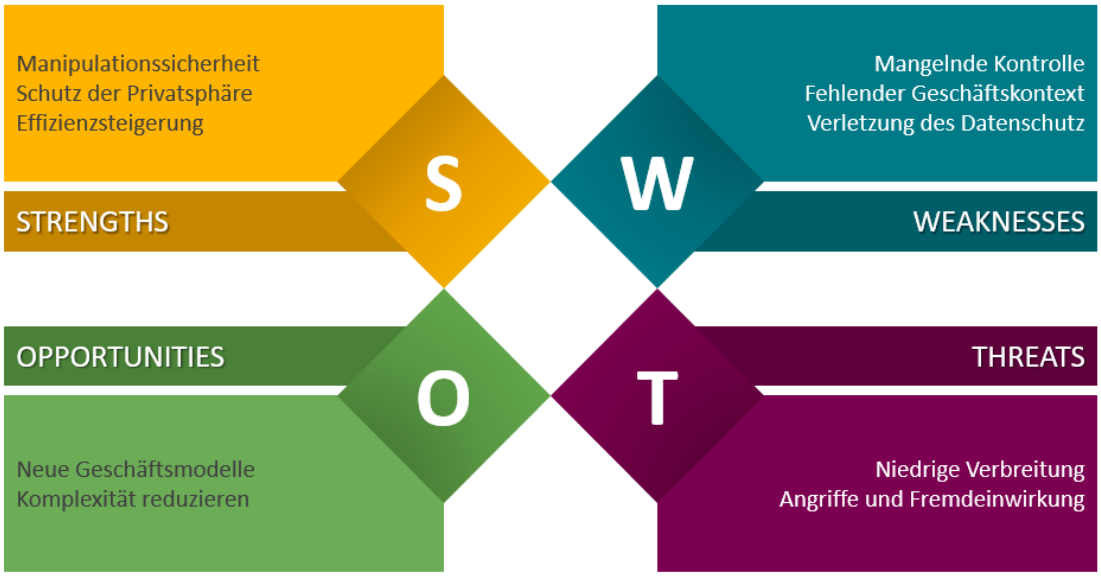

Abbildung 10: Blockchain Technologie SWOT Analyse (eigene Darstellung)

4.1.1. Stärken

Manipulationsicherheit Eine der Schlüsselstärken der Technologie ist, dass sie eine Manipulation von Datensätzen direkt erkennbar macht durch die Art und Weise wie Transaktionen gespeichert und verknüpft werden.

Schutz der Privatsphäre Durch eine Implementierung eines Berechtigungskonzepts können Teilnehmer des Netzwerks eigenständig definieren wer auf die Daten zugreifen kann, für welchen Zweck und für welchen Zeitraum. Diese Regeln werden in Smart Contracts programmatisch abgebildet und bei jeder Ausführung geprüft. So lassen sich beispielsweise komplexe Berechtigungsstrukturen direkt innerhalb des Netzwerks abbilden ohne dazu eine zusätzliche Abstraktionsebene einführen zu müssen.

Effizienzsteigerung Zusätzlich zur Manipulationsicherheit und dem Schutz der Privatsphäre bietet die Blockchain Technologie die Möglichkeit der Effizienzsteigerung für Geschäftsprozesse. Durch den Einsatz von Kryptographie können zwei Parteien vertrauensvoll miteinander interagieren. Eine gesonderte Prüfung der Transaktionen entfällt hierbei, da sie durch Smart Contracts bereits geprüft wurde. Hierdurch entsteht ein Einsparungspotential bzw. eine Effizienzsteigerung.

4.1.2. Schwächen

Mangelnde Kontrolle In der Theorie sind Blockchain Lösungen dezentralisiert und selbstverwaltend (siehe auch Nakamoto, 2009) in der Praxis zeigt sich jedoch, dass der Betrieb eines solchen Systems maßgeblich unter der Kontrolle einer Gruppe von Entwicklern bzw. einer eigens dafür gegründeten Organisation steht.

Fehlender Kontext Eine weitere Schwäche ist das Fehlen eines Mechanismus, um Datensätze in der Kette zurück in den Geschäftskontext ihrer Erstellung zu verknüpfen. Dies kann es schwierig machen, sich auf Blockchain-Datensätze als Nachweis für Geschäftsvorgänge zu verlassen.

Betrachten Sie eine Blockchain-Lösung, wie sie in Schweden und Brasilien erprobt wurde, bei der Landtransfers und Millionen anderer Transaktionen auf einer

öffentlichen Blockchain erfasst wurden. Wie wäre es möglich, den auf der Blockchain aufgezeichneten Hash abzurufen, der einem bestimmten Landtitel zugeordnet ist, wenn es keine Möglichkeit gibt, die Transaktion mit ihrem Geschäftskontext zu verknüpfen? Wie wäre es möglich, E-Discovery-Aufträge zu erfüllen?

Verletzung des Datenschutzes Gesetze zur Datenlokalisierung können sich aus Gesetzen und Vorschriften ergeben, die die Aufbewahrung von Dokumenten in einem Geschäftsgebäude vorschreiben, oder aus Gesetzen, die sich mit Datenschutz und Privatsphäre in Bezug auf Technologie befassen. Im europäischen Kontext ist ein Beispiel die Datenschutz-Grundverordnung (DSGVO), die Anforderungen an die Verarbeitung personenbezogener Daten stellt. Für Länder, die sich auf die Speicherung von Elementen ihrer öffentlichen Aufzeichnungen in einer Blockchain verlassen, die nicht vollständig in ihrer Hoheitsgewalt operiert, ist es notwendig zu prüfen, ob das System den Gesetzen und Vorschriften zur Datenlokalisierung und zum Datenschutz entspricht.

4.1.3. Chancen

Neue Geschäftsmodelle Überall dort wo zur Zeit noch Intermediäre eingesetzt werden zur Abwicklung von Transaktionen zwischen zwei oder mehreren Parteien kann die Blockchain Technologie eingesetzt werden. Mit dem Einsatz von Smart Contracts können Verträge auf der Blockchain abgebildet und mit Hilfe von Algorithmen dezentral über das Netzwerk ausgeführt werden. Es ist nicht notwendig das Intermediäre für die Ausführung und Gestaltung der Verträge von den Vertragsparteien beauftragt werden. Die Erfüllung des Vertrags wird ebenfalls vollständig über die Blockchain kontrolliert und automatisch verwaltet. Unternehmen die als einziges Geschäftsmodell die Vermittlung und Bereitstellung einer Plattform für Anbieter und Kunde haben, also rein zur Abwicklung von Transaktionen dienen, können durch den Einsatz einer Blockchain obsolet werden. Das selbe Prinzip lässt sich auch auf das Lieferketten Management anwenden.

Komplexität reduzieren Der Nachweis einer Charge eines beliebigen Lebensmittels vom Hersteller bis zum Urerzeuger aller verwendeter Bestandteile kann weit

über 200 Papierdokumente von allen beteiligten Teilnehmern der Lieferkette erzeugen. Zahlreiche Amtstellen benötigen diese Dokumente für Nachweispflichten in Bezug auf Hygiene- und Gesundheitsvorschriften. Streckt sich die Lieferkette über mehrere Länder oder sogar Kontinente aus müssen in den meisten Fällen für Zollbehörden ebenfalls Originaldokumente zum Herkunftsnachweis gefordert. Kleinste Mängel an den Dokumente können zu Verzögerungen führen und Chargen die sich im Transit befinden verderben lassen oder die Zahlungen verlangsamen. Mit einer Blockchain kann hier die Komplexität des Prozesses vermindert werden. Jedesmal wenn ein Dokument mehreren Teilnehmern zur Verfügung stehen muss, ermöglicht die Blockchain durch das Hinzufügen eines Datensatzes das sämtliche Aktualisierungen des Dokuments in Echtzeit bereitstehen und die Gültigkeit und Integrität durch das Netzwerk abgesichert sind. Dies kann zu Zeit- und Kosteneinsparungen führen.

4.1.4. Risiken

Niedrige Verbreitung Im Lieferkettenmanagement sind alle Teilnehmer in einem Netzwerk organisiert. Je optimierter dieses Netzwerk ist desto besser kann es in seiner Gesamtheit performen. Entscheiden sich einige Teilnehmer dafür die Blockchain Technologie einzusetzen und einige Teilnehmer nicht so entsteht ein klassischer Systembruch wodurch in diesem Fall die Effizienz der Blockchain sinkt. Wenn beispielsweise die Urerzeuger nicht an dem Blockchain Netzwerk teilnehmen, kann ein vollständiger Nachweis allein über die Blockchain vom Hersteller nicht erbracht werden. Die Vertrauenskette endet an dem Punkt an dem ein virtuelles Gut, was in der Blockchain abgebildet ist, Produktionschritte durchläuft die nicht über die Blockchain abgewickelt werden.

Angriffe und Fremdeinwirkung Wie auch andere IT Landschaften ist ein Blockchain Netzwerk nicht vollkommen vor Angriffen von außen oder innen geschützt. Sicherheitslücken in der verwendeten Plattform der Blockchain oder logische Fehlkonstrukture in Smart Contracts können ein Netzwerk beschädigen oder es sogar komplett stilllegen. Ein möglicher realer Wertverlust für die Teilnehmer des Netzwerk ist in einem solchen Fall kaum zu umgehen. Ebenfalls kann die Vertrauenskette

komprimitiert werden durch bewusste Falscheingabe von Informationen und Metadaten.

4.2. Nutzwertanalyse

Die Nutzwertanalyse unterstützt die Auswahl einer Alternative. Sie wird in diesem Kontext eingesetzt um verschiedene Entscheidungsvarianten miteinander vergleichen zu können. Neben den Entscheidungsvarianten werden Bewertungskriterien definiert und mit dem paarweisen Vergleich priorisiert. Nachdem die Varianten bewertet worden sind kann ein Ergebnis aus der Analysetabelle gelesen werden.

4.2.1. Entscheidungsvarianten

Als Entscheidungsvarianten wurden im Rahmen dieser Arbeit vier potentielle Kandidaten ausgewählt - Ethereum, Hyperledger Fabric, IOTA sowie Quorum. Nachfolgend soll eine kurze Beschreibung dazu dienen alle Kandidaten im Kontext der Blockchain Technologie vorzustellen.

Ethereum Ethereum war die erste Ausprägung der Blockchain Technologie in der Smart Contracts realisiert wurden. Aus diesem Grund wurde Ethereum als erste Option zur Umsetzung einer Supply Chain Lösung in betracht gezogen, denn ohne die Möglichkeit der programmatischen Ausführung von Geschäftslogik lassen sich moderne IT-gestützte Geschäftsprozesse gar nicht erst mit einem Blockchain System abbilden. Die Bitcoin Blockchain besitzt in ihrer ursprünglichen Form beispielsweise keine Unterstützung für Smart Contracts und wurde daher auch direkt als möglicher Kandidat ausgeschlossen. Ethereum ist eine Open Source Lösung. Das Ethereum Netzwerk hat keine Zulassungsbeschränkungen und ist öffentlich, d.h. jeder kann am Netzwerk teilnehmen und auch selber Transaktionen anderer Teilnehmer validieren. Hierdurch ist ein hoher grad an Dezentralisierung und Transparenz gegeben, da keine einzelne Entität das Netzwerk und den Validierungsprozess kontrolliert. Ebenso unterstützt diese Offenheit die Ausfallsicherheit des gesamten Netzwerk sowieso einen gewissen Schutz vor Angriffen aus dem Netzwerk selbst. Ethereum verwendet zur Programmierung von Smart Contracts die Sprache Solidity.

Hyperledger Fabric Hyperledger Fabric ist, wie Ethereum, eine Open Source Lösung. Die Implementierung der Blockchain Technologie wurde Ursprünglich von IBM entwickelt und dann an die Linux Foundation übergeben, welche es dann der Öffentlichkeit frei zur Verfügung stellte. Hyperledger Fabric ist kein fertiges Blockchain Netzwerk welches für einen bestimmten Anwendungsfall konzipiert wurde. Es ist ein Framework um Business Netzwerke und deren Transaktionen in einer einheitlichen Modellierungssprache zu erfassen und umzusetzen. Mit Hyperledger Fabric modellierte Netzwerke sind permissioned und private bzw. in konsortial Form aufgesetzt. Das bedeutet nur ein ausgewählter Kreis an Parteien darf an dem Netzwerk teilnehmen und die Validierung von Transaktionen wird von einer ausgewählten Gruppe von Teilnehmern durchgeführt. Hierdurch weisen Hyperledger Fabric Blockchain Netzwerke eine wesentlich höhere Durchsatzrate für Transaktionen auf als Ethereum, außerdem skaliert ein solches Netzwerk besser, da die Validierungsdauer nicht zwingend mit der Anzahl der Netzwerkteilnehmer ansteigt.

IOTA IOTA wurde entwickelt für eine sichere Kommunikation und Zahlungen im Machine-to-Machine Bereich und dem Internet der Dinge. Das IOTA Netzwerk ist ähnlich wie Ethereum permissionless und public. Im Gegensatz zu Lösungen wie Ethereum oder Hyperledger Fabric verwendet IOTA keine Blockchain als Datenstrukur sondern den sogenannten „Tangle“. Der Tangle ist ein gerichteter azyklischer Graph[11]. Dabei gibt es keine Blöcke wie in der Blockchain sondern die einzelnen Transaktionen im Netzwerk bilden die Knoten des Graphen. Da es sich bei IOTA um ein öffentliches Netzwerk handelt, kann auch jeder Teilnehmer Transaktionen validierer bzw. schreibt der Konsensalgorithmus von IOTA sogar vor, dass jede neue Transaktion zwei vorhandene nicht validierte Transaktionen validieren muss bevor das Netzwerk die neue Transaktion entgegen nimmt. Hieraus ergibt sich der Umstand, das das IOTA Netzwerk mit wachsender Nutzerzahl performanter wird. Zum aktuellen Zeitpunkt kann IOTA nicht als dezentrales System bezeichnet werden, da im IOTA Netzwerk noch ein sog. Coordinator zentral betrieben wird, welcher in regelmäßigen Abständen Snapshots des Netzwerks und der darin enthaltenen Transaktionen veröffentlicht. Alle Transaktionen innerhalb des Snapshots werden als sicher validiert eingestuft.

[11] HIER VERWEIS zu directed acyclic graph

Quorum Quorum ist ein auf Ethereum basierendes Distributed-Ledger-Protokoll, das von JPMorgan Chase entwickelt wurde, um der Finanzdienstleistungsbranche eine Implementierung von Ethereum bereitzustellen die allerdings zulassungsbeschränkt und nicht öffentlich ist. Mit Quorum sollen die Transaktions- und Vertragsdaten geschützt werden, anders als bei Ethereum wo jeder die Transaktionen und Verträge öffentlich einsehen kann. Die Hauptmerkmale von Quorum lassen sich als Erweiterung von Ethereum verstehen und lauten wie folgt:

- Transaktions- und Vertragsdatenschutz
- mehrere abstimmungsbasierte Konsensmechanismen
- Netzwerk/Peer-Berechtigungssystem
- Höhere Leistung in Form eines größeren Transaktionsdurchsatzes

Auch wenn Quorum mit Blick auf die Anwendungsfälle von Finanzdienstleistungen entwickelt wurde, ist die Implementierung nicht spezifisch für Finanzdienstleistungen und daher für andere Branchen geeignet, die an der Nutzung von Ethereum interessiert sind, aber die oben genannten primären Funktionen benötigen.

4.2.2. Analyse Methode

Die vorgestellten Entscheidungsvarianten werden in der Nutzwertanalyse anhand von festgelegten Kriterien bewertet um einen objektiven Vergleich zu schaffen. Dabei ist es wichtig die Kriterien untereinander zu priorisieren, damit das Ergebnis der Analyse möglichst genau für den Use-Case zugeschnitten ist. Um jetzt Kriterien zu priorisieren, existieren die verschiedensten Ansätze, für diese Arbeit wurde der Ansatz des paarweisen Vergleich herangezogen.

Was ist der Paarweise Vergleich? Beim paarweisen Vergleich werden jeweils zwei Kriterien miteinander verglichen und festgelegt welches Kriterium wichtiger ist. Diesen Vergleich führt man mit jedem möglichen Paar aus Kriterien durch und erhält so eine Rangfolge für alle Kriterien.

Wann kann man diese Methode einsetzen? Sind die gewählten Kriterien nicht eindeutig messbar bietet sich der Paarweise Vergleich an. Hierdurch werden alle Kriterien systematisch gegenübergestellt und es wird möglich eine objektive Entscheidung bei der Gewichtung der Kriterien zu erhalten.

Wie funktioniert der Paarweise Vergleich? Alle Kriterien der Nutzwertanalyse werden in eine sog. Präferenzmatrix eingetragen. Die Schnittpunkte zwischen Zeilen und Spalten stellen den eigentlichen Vergleich dar. Je Kriterium wird der Zeilenwert mit allen Spaltenwerten paarweise Verglichen. Das Ergebnis des Vergleichs kann drei Ausprägungen annehmen.

- Der Zeilenwert ist weniger wichtig.
- Der Zeilenwert ist gleich wichtig.
- Der Zeilenwert ist wichtiger.

Zuletzt wird der Gesamtnutzwert einer Entscheidungsvariante berechnet. Dazu multipliziert man das Gewicht des Kriteriums mit dem Teilnutzenwert einer Entscheidungsvariante. Das Ergebnis entspricht dem gewichteten oder relativen Teilnutzenwert. Anschließend werden die gewichteten Teilnutzenwerte addiert. Das Resultat ist der Gesamtnutzwert der Entscheidungsvariante.

$$GN_i = \sum_{j=1}^{n} g_j \times TN_{ij} \tag{2}$$

Mit:

- GN_i als Gesamtnutzwert der Entscheidungsvariante i
- g_j als Gewicht des Bewertungskriteriums j
- n als Anzahl der Bewertungskriterien
- TN_{ij} als Teilnutzen der Entscheidungsvariante i in Bezug auf das Kriterium j

Aus diesen Summen der Zeilenwerte ergibt sich eine Rangfolge bzw. eine Gewichtung für die Kriterien. Mit diesen gewichteten Kriterien lassen sich dann die Entscheidungsvarianten in der eigentlichen Nutzwertanalyse bewerten.

4.2.3. Kriterien

Aus den Ergebnis der SWOT-Analyse in Kapitel 4.1 wurden die folgenden Kriterien der Nutzwertanalyse abgeleitet. Eine kurze Erläuterung der Kriterien soll einen Überblick bieten.

- Konsensmechanismus
- Skalierbarkeit
- Interoperabilität
- Reifegrad
- Vertrauen in Tx-Validierer
- Anonymität der Tx-Validierer
- Supply Chain Suitability
- Governance

Konsensmechanismus Das Kriterium Konsensmechanismus soll zum Einen die Möglichkeit eines austauschbaren Algorithmus und zum Anderen generell die Entscheidungsvariante bezüglich des eingesetzten Algorithmus bewerten. Dabei kommt es darauf an wie leistungsintensiv der eingesetzte Algorithmus und die möglichen Alternativen sind. Der Konsensmechanismus der für ein Blockchain Netzwerk verwendet wird hat Auswirkungen auf die Performance und Effizienz.

Skalierbarkeit Die Skalierbarkeit einer Blockchain Technologie kann unter anderem von der benötigten Speichergröße oder einer bestimmten minimalen Transferrate innerhalb des Netzwerks rein technisch begrenzt werden. Ebenfalls sind nichttechnische Begrenzungen denkbar wie beispielsweise gesetzlich definierte maximal oder minimal Werte für bestimmte Eigenschaften des Netzwerks oder einzelner Netzwerkkomponenten.

Interoperabilität Unter Interoperabilität ist die Konnektivität der Blockchain Netzwerke zu anderen Systemen gemeint. Dazu zählen vorhandene Schnittstellen oder Dienste durch die Smart Contracts Informationen und Daten bei der Ausführung beziehen können.

Reifegrad Mit dem Reifegrad einer Variante wird einerseits die Softwarereife und andererseits die Zeit seit Gründung/Entwicklung bzw. Präsenz am Markt bewertet. Auf Grund der hohen Geschwindigkeit in der Weiterentwicklung der einzelnen Technologie „Stacks“ können Angebote von Software Frameworks relativ schnell wieder vom Markt verschwinden. Dies muss bei der Konzeption eines zukünftigen Blockchain Netzwerks zwingend beachtet werden um eine Migration möglichst zu verhindern.

Vertrauen in Validatoren Ein Blockchain Netzwerk benötigt zwingend einzelne Teilnehmer oder eine Gruppe von Teilnehmern, welche neue Transaktionen im Netzwerk auf ihre Integrität hin validieren. Blockchain Netzwerke werden wie in Kapitel 3.2.3 beschrieben eingeteilt in permissioned und permissionless Netzwerke. Über dieses Kriterium lässt sich also bewerten in wie weit das Netzwerk Vertrauen in die Validierer benötigt. In einem permissionless Netzwerk kann jeder Teilnehmer als Validator auftreten, in einem permissioned Netzwerk muss jeder Validator bestimmte Anforderungen erfüllen um Transaktionen validieren zu können.

Anonymität der Validatoren Aus Kapitel 3.2.3 geht hervor, dass Blockchain Netzwerke auf zwei Arten den Zugang zum Netzwerk regeln. Ein sog. public Netzwerk ist vollständig öffentlich zugänglich, es bestehen demnach keine Zugangsbeschränkungen außer von technischer Seite. Für ein private bzw. consortium Netzwerk gelten definierte Zugangsbeschränkungen, sodass jeder Teilnehmer in der Regel durch eine neutrale Entität für den Zugang zum Netzwerk authorisiert wird. Je nach Art der gewählten Zugangsbedingungen wird entsprechend die Anonymität der Teilnehmer bestimmt.

Supply Chain Suitability Supply Chain Suitability beschreibt die allgemeine Nutzbarkeit der Entscheidungsvariante für Anwendungsfälle im Bereich des Supply Chain

Management. Technische Grenzen oder Designentscheidungen können den Einsatz einer bestimmten Blockchain Technologie erschweren oder sogar gänzlich unmöglich machen.

Governance Die Governance beschreibt die Hoheitsrechte an der Technologie. Nicht alle Ausprägungen der Blockchain Technologie sind vollständig als Open Source Software entwickelt und konzipiert worden. Proprietäre Blockchain Lösungen weisen per Definition weniger Transparenz auf können aber präziser auf einen bestimmten Use-Case zugeschnitten sein, da solche Lösungen in der Regel nicht als generisches System für mehr als einen Use-Case konzipiert werden. Im Gegensatz dazu sind proprietäre Lösungen weniger flexibel bei der Adaption von neuen Technologien oder Anpassungen auf Grund von Änderungen im Prozess.

Die beschriebenen Kriterien lassen sich in einer Präferenzmatrix (Tabelle 3) erfassen und dann mit der Methode des Paarweisen Vergleichs priorisieren. Die priorisierten Bewertungskriterien werden in die Nutzerwertanalyse übertragen, um die einzelnen Entscheidungsvarianten bewerten zu können.

Kriterium	**Nr.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Punkte**	**Gewichtung**
Konsensmechanismus	1		1	1	4	5	1	7	8	3	10,7
Skalierbarkeit	2			2	4	5	6	2	2	3	10,7
Interoperabilität	3				3	5	6	3	3	3	10,7
Reifegrad	4					5	6	4	8	3	10,7
Vertrauen	5						5	5	5	7	25,0
Anonymität	6							7	6	4	14,3
Supply Chain Suitability	7								7	3	10,7
Governance	8									2	7,1
										Total	**100,0**

Tabelle 3: Präferenzmatrix der Bewertungskriterien der Nutzwertanalyse

4.2.4. Ergebnis

Das Grundgerüst der Nutzwertanalyse ergibt sich aus dem Zusammenschluss der Komponenten. Die bewerteten Entscheidungsvarianten lassen sich an den Spalten der Tabelle 4 ablesen. Anhand der Besonderheiten sollen die Entscheidungsvarianten diskutiert werden.

Ethereum, als erste Entscheidungsvariante, lässt sich durch den Aufbau des Systems für den Einsatz einer Chargenrückverfolgung in der Fleischwarenindustrie nur bedingt einsetzen. Dies lässt sich begründen mit der Art und Weise wie innerhalb des Ethereum Netzwerks neue Transaktionen validiert werden (permissionless). Ein entscheidender Faktor warum Ethereum am schlechtesten bei der Analyse abschneidet, ist die fehlende Möglichkeit Geschäftsdaten ausreichend vor ungewollter Einsicht schützen zu können. Ebenfalls bietet Ethereum keine native Möglichkeit Daten oder Informationen aus Drittsystemen zu beziehen und für die Ausführung der Geschäftslogik (Smart Contracts) zu nutzen.

IOTA, nach Ethereum die nächst höher bewertete Entscheidungsvariante, ist von Grund auf als DLT für den Einsatz im Internet of Things (IoT) konzipiert worden und bietet daher einige Vorteile gegenüber Ethereum. Die Kriterien Interoperabilität und Konsensmechanismus erfüllt IOTA mehr als Ethereum. Konkret bietet IOTA einen Konsensmechanismus der zukunftssicher sein soll und einen erheblich niederigen Energieverbrauch verursacht als klassische Konsensmechanismen wie beispielsweise PoW. Im Gegenzug steht IOTA noch relativ am Anfang was den Reifegrad des Gesamtsystems betrifft. So ist das IOTA Netzwerk zum aktuellen Zeitpunkt nicht dezentralisiert. Es wird zur Koordination der Transaktionen noch ein sogenannter „Coordinator“ eingesetzt.Quelle Coordinator

Quorum realisiert einige Aspekte des Blockchain Netzwerks grundlegend besser als Ethereum im Kontext des Use-Cases. Quorum ist als permissioned private Netzwerk konzipiert was dem Gegenteil von Ethereum entspricht. Aus diesem Grund setzt Quorum nicht auf einen PoW Konsensmechanismus, sondern bietet verschiedene BFT-basierte Mechanismen an. Ebenfalls bietet Quorum eine Möglichkeit Transaktionen mit Zugriffsbeschränkungen zu nutzen. Dadurch sind nur Teilnehmer berechtigt den Inhalt der Transaktion zu sehen, die im Vorfeld für diese Transaktion bestimmt

wurden. Dennoch wird solch eine Transaktion vom Gesamtnetzwerk verarbeitet und validiert. Quorum wurde von JPMorgan Chase entwickelt und entsprechend ist die Lösung nicht quelloffen. Eine Herstellabhängigkeit kann nicht ausgeschlossen werden.

Hyperledger bietet anhand den Ergebnissen der Analyse die besten Möglichkeiten um eine Chargenrückverfolgung über eine Blockchain zu realisieren. Dies beruht darauf, dass wichtige Kriterien wie Konsensmechanismus, Interoperabilität und allgemeine Eignung für den Einsatz im Supply Chain Umfeld von Hyperledger im Vergleich zu den drei anderen Entscheidungsvarianten am meisten erfüllt werden. So lassen sich in einem Hyperledger Netzwerk die unterschiedlichsten Konsensmechanismen nutzen. Je nach Einsatzzweck des Netzwerks kann der Konsensmechanismus nahezu frei gewählt werden. Außerdem bietet Hyperledger eine native Möglichkeit um Smart Contracts mit Informationen aus Drittsystemen zu versorgen. So lassen sich Hyperledger Netzwerk nahtlos in vorhandene Systemlandschaften implementieren.

So lässt sich aus Tabelle 4 entnehmen, das Hyperledger die Kriterien mit Abstand am besten erfüllt. Aus diesem Grund wird für die Konzeption und prototypische Implementierung einer Chargenrückverfolgung für die Fleischwarenindustrie die Hyperledger Plattform verwendet.

			Ethereum		**Hyperledger**		**IOTA**		**Quorum**	
Nr.	Kriterium	Gewichtung	Score	Result	Score	Result	Score	Result	Score	Result
1	Konsensmechanismus	10,7	5	54	9	96	7	75	6	64
2	Skalierbarkeit	10,7	5	54	9	96	8	86	8	86
3	Interoperabilität	10,7	5	54	9	96	5	54	7	75
4	Reifegrad	10,7	7	75	7	75	5	54	8	86
5	Vertrauen	25,0								
6	Anonymität	14,3								
7	Supply Chain Suitability	10,7	4	43	8	86	6	64	7	75
8	Governance	7,1	8	57	7	50	6	43	5	36
	Total	100,00		336		500		375		421

Tabelle 4: Tabellarische Darstellung der Nutzwertanalyse

4.3. Zusammenfassung Lösungskonzept

Ausgehend von einer SWOT-Analyse, mit welcher die Potentiale und Probleme der Blockchain Technologie allgemein aufgezeigt wurden, erfolgte keine Bewertung der identifizierten Potentiale bzw. Probleme. In diesem ersten Schritt wurden noch keine konkreten Ausprägungen der Blockchain Technologie untersucht, sondern die Technologie als Ganzes. Im nächsten Schritt, der Nutzwertanalyse, wurden dann vier Entscheidungsvarianten zur Konzeption und prototypischen Implementierung einer Chargenrückverfolgung für die Fleischwarenindustrie ausgewählt und kurz vorgestellt. Eine Präferenzmatrix wurde erstellt und dokumentiert, um Kriterien für die Nutzwertanalyse untereinander priorisieren zu können. Mit diesen priorisierten Kriterien konnten dann die vier Varianten innerhalb der Nutzwertanalyse bewertet werden. Als Ergebnis der Analyse hat sich herausgestellt, dass die Hyperledger Blockchain Lösung am besten geeignet ist zur Umsetzung des Use-Cases. Im nächsten Kapitel wird dann ein Systementwurf modelliert und dokumentiert.

5. Systementwurf

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

5.1. Vorgehensweise Anforderungserhebung

Die Anforderungen für ein zu konzipierende System wurden vor dem Hintergrund der Evaluation des Geschäftsprozesses erhoben. Außerdem wurde bei der Erfassung der Anforderung darauf geachtet, das der Prototyp beim Praxispartner als Unterstützung für zukünftige Innovationsfragen herangezogen werden kann. Wie von Dick et al. (2017); Hull (2011) beschrieben, kann das Prototyping selbst bereits als Anforderungsanalyse angesehen werden, jedoch wurde für die prototypische Implementierung des Systementwurfs eine gesonderte Anforderungsanalyse durchgeführt. Ziel dieser Vorgehensweise ist eine präzise Definition und Eingrenzung der Anforderungsbeschreibung während des Konzeptions- und Implementierungsprozesses.

Im Zuge der Anforderungserhebung wurden die Anforderungen in Zusammenarbeit mit dem Praxispartner entwickelt. Die Anforderungen wurden dabei in textueller Form nach einem festen Muster in Anlehnung an Pohl and Pohl (2015) definiert. Ergänzt wurden die textuellen Anforderungen um Prozessdiagramme und Mockups der Nutzeroberflächen. Der Fokus des konzipierten System liegt dabei allerdings auf eigentlichen Blockchain Netzwerk und weniger auf der Benutzungsoberfläche.

Die Anforderungen wurden untergliedert in funktionale Anforderungen, Rahmenbedingungen und Qualitätsanforderungen. Ebenso wurden die Anforderungen hierarchisch strukturiert und um eine Quelle ergänzt nach Koelsch (2016). Dies soll die Nachverfolgbarkeit der Anforderungen während der Evaluation unterstützen.

5.2. Das Ziel: Chargenrückverfolgung innerhalb der Fleischwarenindustrie

Das System soll unter experimentellen, abstrahierten Bedingungen die Chargenrückverfolgung von Schweinen innerhalb der Produktions- und Wertschöpfungskette realisieren. Dafür muss das System den Prozess vom Erzeuger bis zum Groß- und Einzelhandel unterstützen. Konkret sollen Erzeuger neue Tiere im Blockchain Netzwerk registrieren und einer Charge zuordnen können. Bereits registrierte Tiere sollen zur Weiterverarbeitung freigegeben werden können und ein Eigentumswechsel muss durch das System abbildbar sein. Die Gesamtheit der Transaktionen zwischen den Teilnehmern der Wertschöpfungskette kann als Graph angesehen werden. Anhand dieses Graphen soll eine Rückverfolgbarkeit einer Charge gewährleistet werden. Über eine Benutzungsoberfläche sollen die Teilnehmer jederzeit in der Lage sein den Graphen einsehen zu können. Für die technische Umsetzung des System spielt die Benutzungsoberfläche jedoch eine nachgelagerte Priorität. Hauptaugenmerk des Systementwurfs liegt auf dem technologischen Aufbau des Blockchain Netzwerk und den Schnittstellen für etwaige Drittsysteme zur automatischen Erfassung von Tieren. Eine automatische Erfassung von neuen Tieren kann beispielsweise über IoT-Sensoren in Schlachthaken erfolgen. Ebenso würde sich ein Eigentumswechsel, wenn Tiere vom Erzeuger an den Schlachthof verkauft werden, über RFID-Chips und entsprechende Lesegeräte, welche per Schnittstelle mit dem Blockchain Netzwerk verbunden sind, abwickeln lassen (Dorri et al., 2017; Samaniego and Deters, 2016).

5.3. Die Wertschöpfungskette im Detail

Nachfolgend soll eine kurze Erläuterung der in Kapitel 5.2 erwähnten Wertschöpfungskette dazu dienen, die Daten- und Warenströme zwischen den Teilnehmern klar zu trennen und die für diesen Systementwurf wichtigen Informationen herauszuarbeiten. Da eine Chargenrückverfolgung nur gewährleistet werden kann, wenn in den vorgelagerten Prozessen die nötigen Informationen in einem System bereitgestellt wurden, soll auf die Teilschritte vom Erzeuger zum Endverbraucher eingegangen werden.

Die Fleischwirtschaft hat in den letzten Jahren einen Strukturwandel vollzogen,

welcher auch Auswirkungen auf die eigentliche Tätigkeit sowie die Lieferanten- und Abnehmerbeziehungen zwischen den Unternehmen hat (Nolte, 2006). Als eine der zentralen Ursachen für den Strukturwandel wird die Konzentrierung der Schlachtunternehmen gezählt. Inzwischen werden deutlich mehr als 50% aller Schweine in Deutschland von drei Unternehmen geschlachtet - Tönnies, Vion und Westfleisch. Unter Beachtung anderer Wirtschaftszweige wie beispielsweise der Geflügelschlachtung, die noch wesentlich stärker konzentriert ist, und dem Hintergrund das in Ländern wie Dänemark die Schlachtung nur noch von zwei Unternehmen durchgeführt wird, wird deutlich das der Konzentrationsprozess in Deutschland auf der Schlachtstufe noch nicht abgeschlossen ist. Im Gegensatz dazu ist der Viehandel und die Landwirtschaft weniger stark konzentriert, weshalb sie sich in einer schwachen Verhandlungsposition befinden. Um dieser schwachen Verhandlungsposition entgegenzuwirken sind Unternehmen des Viehandels dazu gezwungen immer größere Mengen an Schlachttieren zu einer Charge zu bündeln. Ebenfalls sind zahlreiche unternehmensübergreifende Kooperationen im Viehandel zu beobachten (Voss et al., 2010).

Vom Erzeuger bis zum Endverbraucher ist die Wertschöpfungskette in Deutschland sehr vielfältig ausgeprägt (Freund, 1997). Der Hauptabsatzweg für Schweinemäster läuft entweder über eine direkt Vermarktung an Schlachtbetriebe (einstufige Vermarktung) oder indirekt über den privaten Viehandel, Viehvermarktungsgenossenschaften oder Erzeugergemeinschaften (zweistufige Vermarktung). Die Schlachtstufe lässt sich daher als Flaschenhals der Wertschöpfungskette aus Sicht der Schweinemäster betrachten. Um klar bestimmen zu können welche Informationen und virtuellen Assets in dem Blockchain Netzwerk abgebildet werden müssen, werden der Waren- und Datenstrom nachfolgend einzeln betrachtet.

5.3.1. Betrachtung des Warenstroms

Die Wertschöpfungskette vom Erzeuger bis zum Fleischwarenproduzenten gliedert sich grob in vier Produktionsschritte, welche nachfolgend kurz beschrieben und in Abbildung 11 schematisch dargestellt werden. Dabei sind sieben Parteien direkt in den Gesamtprozess bis zum Verbraucher involviert und eine achte Partei wirkt indirekt als Vermittler zwischen den anderen Parteien mit.

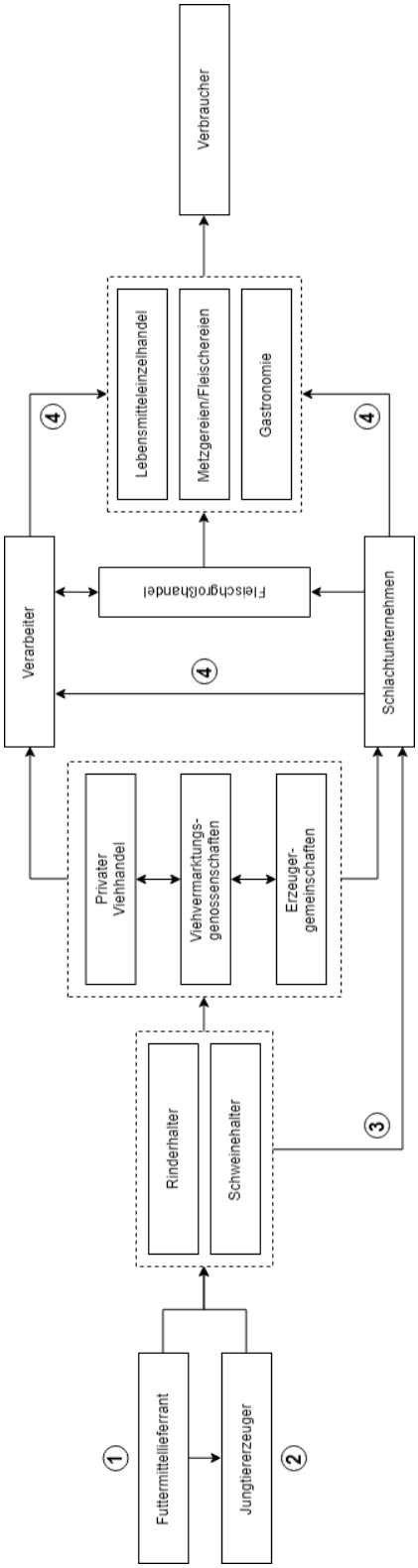

Abbildung 11: Struktur der Wertschöpfungskette der Fleischwirtschaft nach Beck et al. (2008); Petersen et al. (2010); Voss et al. (2010)

Der Warenstrom beginnt mit (1) der Futtermittellieferung an die Jungtiererzeuger und Viehhalter. Jeder Betrieb wird dabei über die Internationale Lokationsnummer (ILN) global eindeutig identifiziert. (2) Nach der Aufzucht der Jungtiere werden diese durch Transportunternehmen zu den Viehhaltern transportiert. In den Mästbetrieben bleiben die Tiere dann bis zur Schlachtreife. (3) Im Auftrag der Schlacht- und Zerlegebetriebe werden die schlachtreifen Tiere von den Mästbetrieben angeliefert. Nach der Verarbeitung der Tiere in den Schlacht- und Zerlegebetrieben werden diese (4) an die verschiedenen Abnehmer geliefert, um letztendlich zu Produkten für den Verbraucher weiterverarbeitet zu werden. Hieraus ergibt sich, dass mindestens an den erwähnten vier Punkten der Wertschöpfungskette eine Prozessschnittstelle vom Blockchain Netzwerk bedient werden können muss.

5.3.2. Informationswege in der Fleischindustrie

Abbildung 12 zeigt den nachfolgend beschriebenen Datenstrom zwischen den einzelnen Produktionsstufen der Fleischindustrie. Referenz zu Anforderungen (1) Jungtiererzeuger und Viehhalter senden jeweils eine Futtermittelbestellung an den Futtermittellieferanten. (2) Nach erfolgreicher Lieferung informiert der Futtermittellieferant den privaten Viehhandel bzw. die Viehvermarktungsgenossenschaften respektive Erzeugergemeinschaften. Die Viehhalter melden einerseits (3) die Aufnahme der Jungtiere und andererseits (4) die schlachtreife von Tieren an die Viehvermarktungsgenossenschaften zur Weitervermittlung and die Schlacht- und Zerlegebetriebe. (5) Bei der Weitervermittlung werden die Informationen über die Tiere an Schlacht- und Zerlegebetriebe übermittelt. (6) Mit dem Lieferauftrag initiiert das Schlachtunternehmen die Bestellung und den Transport der schlachtreifen Tiere. (7) Die Viehvermarktungsgenossenschaften bestätigen den Lieferauftrag mit einer elektronischen Ankündigung der Schlachtviehlieferung. Bei der Anlieferung der Tiere gleicht das Schlachtunternehmen die tatsächliche angelieferte Anzahl mit der bestellten Menge ab und meldet die Werte an die Viehvermarktungsgenossenschaften zurück. Mit dieser Wareneingangsmeldung kann die Viehvermarktungsgenossenschaft den Bestand und die aktuellen Standorte der Tiere aktualisieren. (9) Im Schlachtunternehmen werden dann weitere Informationen zu den Stammdaten der Tiere erfasst. Dazu zählen die Vieh-Verkehrs-Verordnung (VVVO)-Nummern der Landwirte, eine Verga-

be Partie-Nummer je Lkw und eine fortlaufende Schlachtnummer. (10)Anschließend werden die Informationen wieder an die Viehvermarktungsgenossenschaft zurück gemeldet. (11) Letztendlich bedienen die Schlacht- und Zerlegebetriebe die Bestellungen der Fleischwerke, Lebensmitteleinzelhandel, Metzgereien und die Gastronomie. (12) Hier werden dann auch die letzten Stammdaten zu den Produkten erfasst und verknüpft wie beispielsweise Artikelbezeichnung, Stückzahl, Schlachtdatum und Schlacht-Nummer. (13) Mit der Zuordnung der zuverarbeitenden Fleischerzeugnisse zum Lieferschein in einem ERP-System enden die betrachteten Informationswege in der Fleischindustrie.

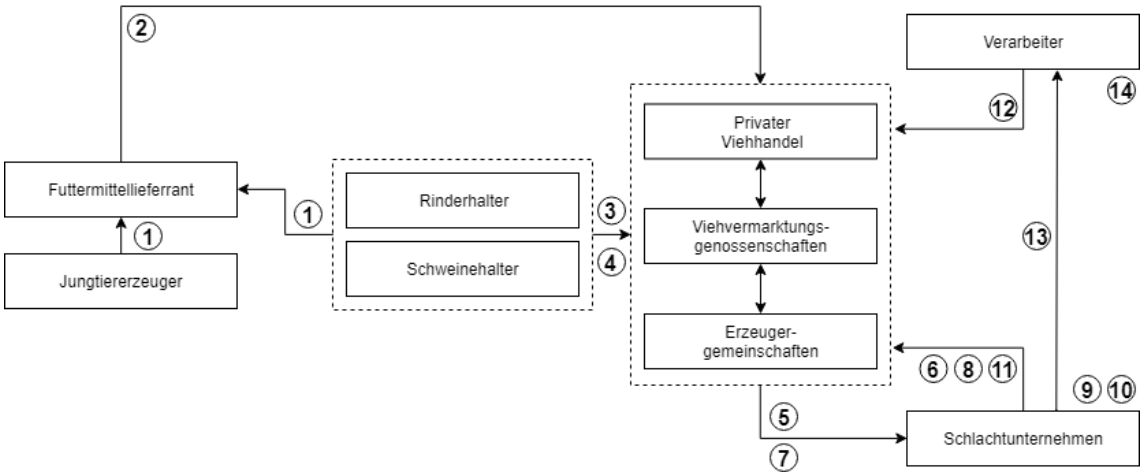

Abbildung 12: Datenströme innerhalb der Wertschöpfungskette QUELLE

5.4. Geschäftsprozess Chargenrückverfolgung

Die vorrangegangene Betrachtung der Waren- und Datenströme macht deutlich an welchen Schnittpunkten der Wertschöpfungskette Informationen gesammelt und zentral über die Viehvermarktungsgenossenschaften verwaltet werden. Dies ist wichtig für den Geschäftsprozess der Chargenrückverfolgung, da eine lückenlose Rückverfolgbarkeit nur dann gewährleistet ist wenn vom Erzeuger bis zum Endverbraucher alle Informationen konsistent und transparent zur Verfügung stehen. Dabei spielt es keine Rolle von welcher Seite der Wertschöpfungskette eine Rückverfolgung durchgeführt wird im Sinne des Down- und Uptracing.

Der Vergleich zwischen dem Prozess der Rückverfolgung wie er aktuell durch-

geführt wird (Ist-Prozess) und wie er mit dem Einsatz eines Blockchain Systems aussehen kann (Soll-Prozess) dient dazu die funktionalen Anforderungen ableiten zu können.

Ist-Prozess Der Ist-Prozess (Abbildung 13) durchläuft die Schritte von der Verbrauchermeldung bis zur Information der anderen Teilnehmer in der Wertschöpfungskette. Dabei wird anhand der Produktkennung und Verbrauchermeldung ermittelt zu welcher Produktcharge die Meldung gehört. Hierfür wird eine vielzahl an Software und Datenbeständen benötigt. Dazu zählt die Office Suite von Microsoft und ein ERP-System in Kombination mit einer Lieferantenmanagement- (SAP SRM) und Vertriebslösung (SAP CRM). Nach der Zuordnung der Verbrauchermeldung zur Produktcharge wird im Sinne des Uptracing die Charge bis zum Erzeuger zurückverfolgt, um zu prüfen in welchem Produktionsschritt das gemeldete Problem entstanden ist. Hierdurch können Maßnahmen zum Abstellen des Problem erarbeitet werden, die an alle Teilnehmer übermittelt werden. Die Chargeninformationen werden bereitsgestellt von einer zentralen Instanz, der Viehvermarktungsgenossenschaft. Dies bedeutet, liegen der Viehvermarktungsgenossenschaft lückenhafte bzw. manipulierte Datensätze vor besteht die Gefahr eine Rückverfolgung nicht vollständig durchführen zu können. Ebenfalls muss der Verbraucher der Viehvermarktungsgenossenschaft vertrauen für vollständig- und korrektheit der bereitgestellten Informationen. Nachdem alle Teilnehmer informiert sind ist der Prozess der Rückverfolgung abgeschlossen. Entsprechende folge Prozesse für einen eventuellen Rückruf von Produkten werden beim Abschluss der Rückverfolgung teils automatisch teils manuell ausgelöst.

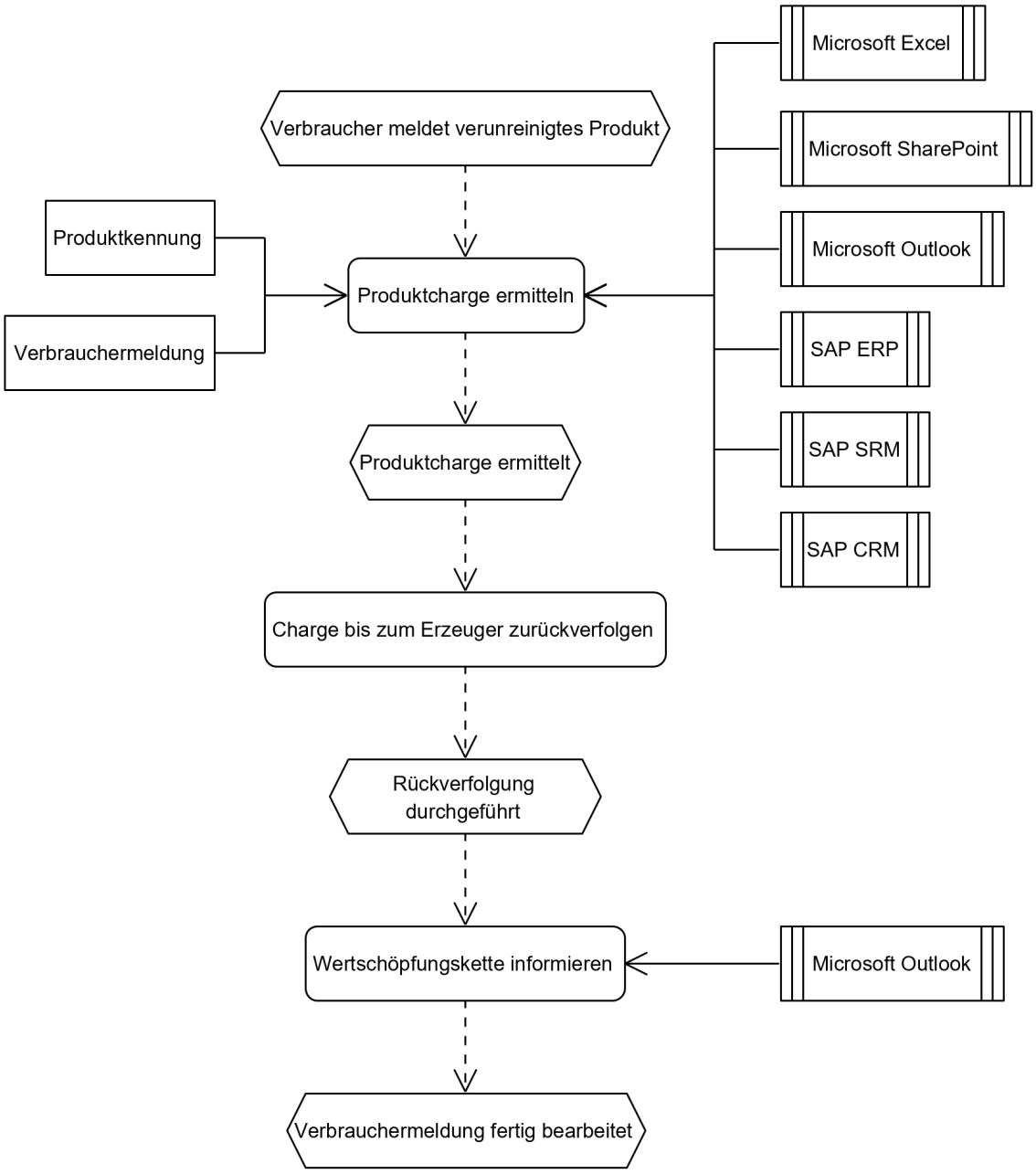

Abbildung 13: Darstellung des Geschäftsprozess Chargenrückverfolgung in eEPK Notation

Soll-Prozess Während im Ist-Prozess (Abbildung 13) viele verschiedene IT-Systeme zum Einsatz kommen um alle Chargeninformationen zusammenzutragen, wird im

Soll-Prozess das Blockchain Netzwerk und darauf aufsetzende dezentrale Applikationen genutzt. Betrachtet man die einzelnen Prozessschritte so ändert sich bei dem Einsatz einer Blockchain oberflächlich nichts, bei näherer Betrachtung wird dann allerdings deutlich, dass sämtliche Informationen zur Rückverfolgung der Charge vom Blockchain Netzwerk zur Verfügung gestellt werden und nicht in einzelnen Datensilos liegen wie im Ist-Prozess. So dient die Blockchain als gemeinsame Datenbasis für sämtliche Informationen die während der Produktion vom Erzeuger bis zum Lebensmitteleinzelhandel erhoben werden. Änderungen werden transparent in der Blockchain erfasst und sind durch den Konsensmechanismus vor nachträglicher Manipulation geschützt.

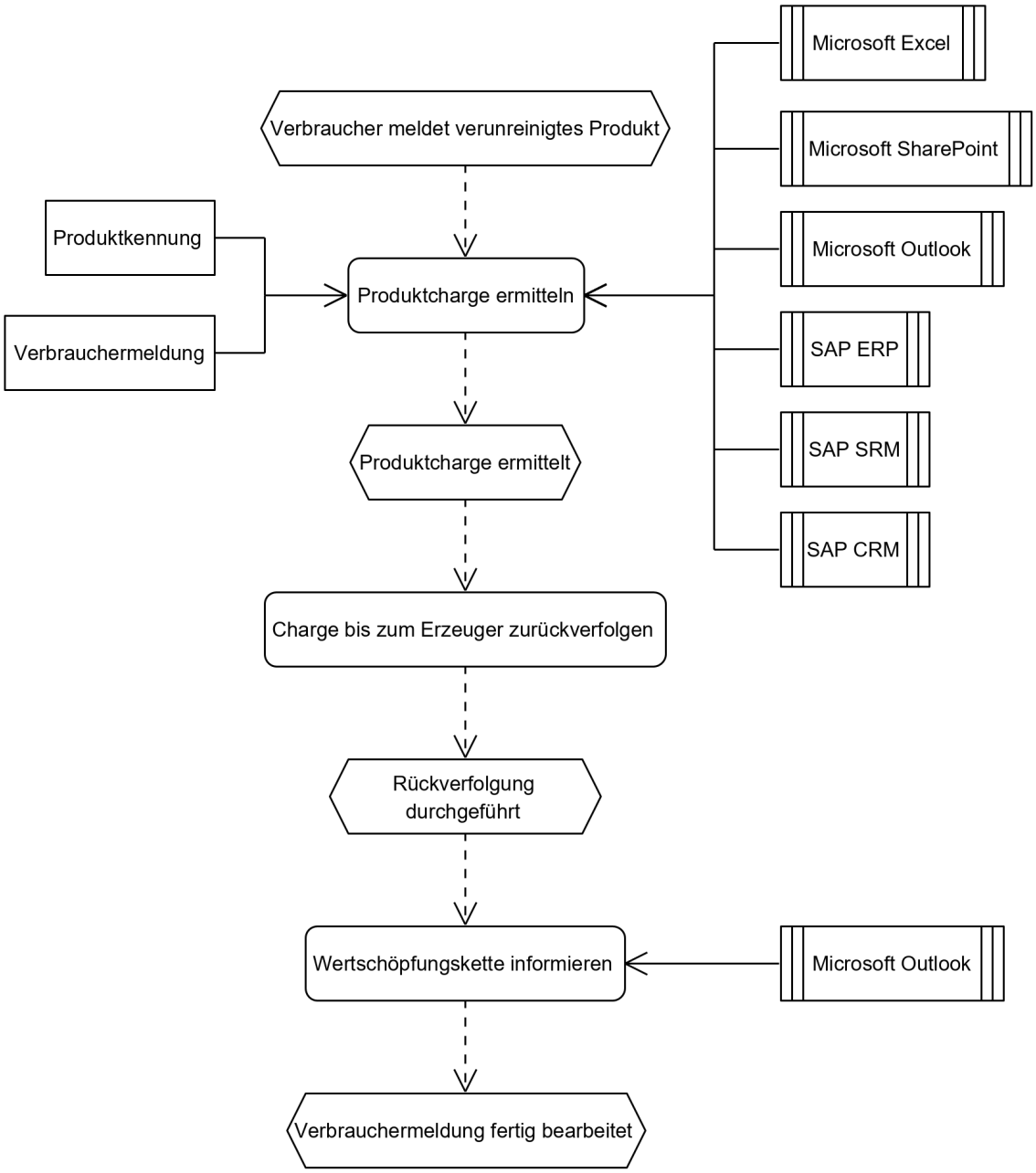

Abbildung 14: Darstellung des Geschäftsprozess Chargenrückverfolgung in eEPK Notation Zu SOLL anpassen

5.5. Systementwurf gemäß Architekturkonzept

Unter berücksichtigung der Resultate aus Kapitel 4 im Kontext des Anwendungsfalls ergibt sich die Grobarchitektur für das System wie in Abbildung 15 dargestellt.

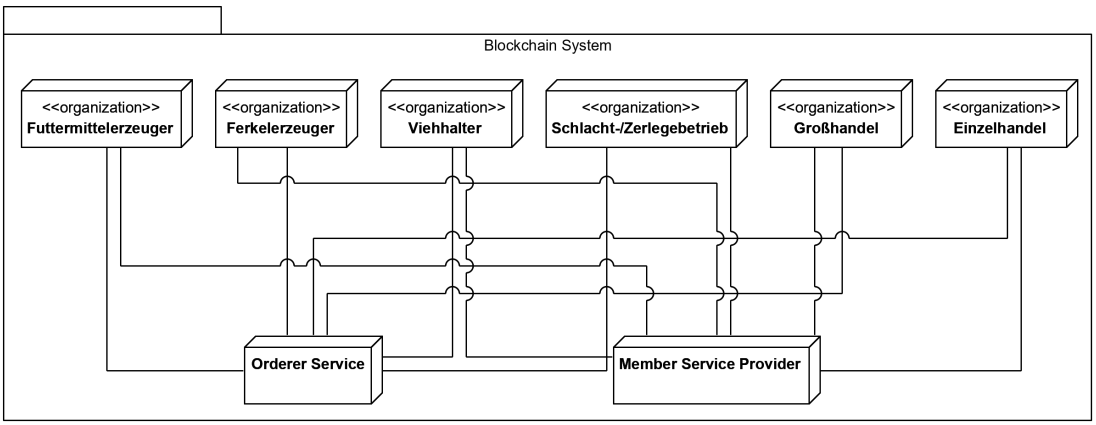

Abbildung 15: Blockchain System Architektur

Abbildung 16 zeigt einen Knoten vom Typ *Organization* im Detail. Dem nach besteht eine *Organization* aus logischer Sicht aus dem *Ledger*, einer *Zustandsdatenbank*, den *Smart Contracts* (Chaincode), dem *Konsensmechanismus*, den einzelnen *Teilnehmern* und dem *User Interface*. *Ledger*, *Zustandsdatenbank* und *Smart Contracts* werden zusammen als *Peer* bezeichnet. Wobei die Ausführung der Smart Contracts in einer isolierten Umgebung erfolgt. Zusätzlich gibt es noch eine Sicherheitsstrategie (Certificate Authority (CA)) zum Schutz der einzelnen Komponenten. Jeder Teilnehmer des Systems muss mindestens einen *Peer* betreiben, um Transaktionen im Netzwerk erstellen und validieren zu können. Mit jedem zusätzlichen Peer wird die individuelle Ausfallsicherheit der *Organization* erhöht. Nachfolgend werden die einzelnen Komponenten näher beschrieben.

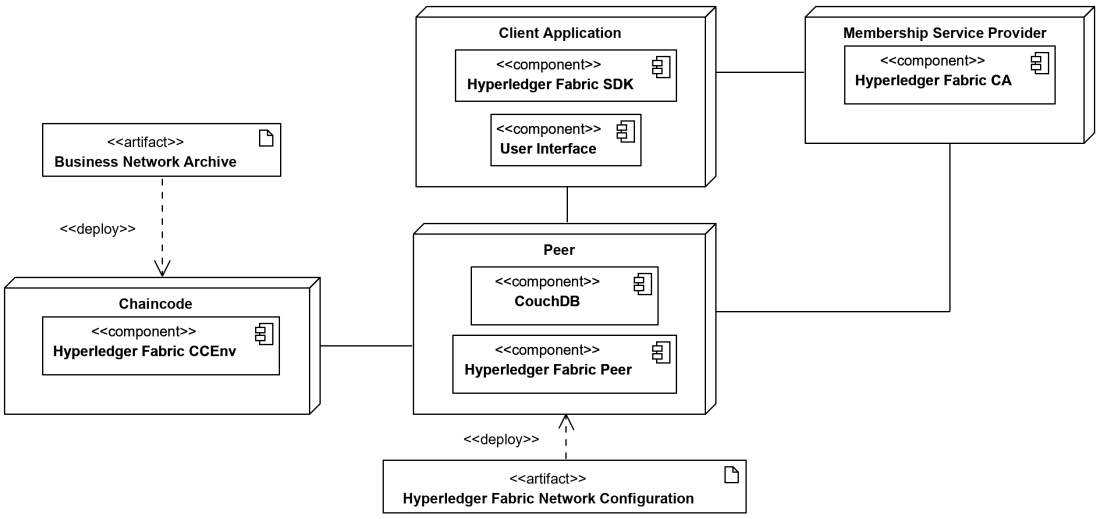

Abbildung 16: Organisation Komponenten Diagramm

5.5.1. Ledger/Konsens

Das sogenannte Ledger besteht aus der verketteten Liste der Transaktionen (Hyperledger Fabric Peer), einer Zustandsdatenbank (CouchDB) und dem Orderer Service. Zustandsveränderungen sind Veränderungen aufgrund von Smart Contract Ausführungen, welche durch Teilnehmer oder Smart Contracts ausgelöst werden. Jede Transaktion beschreibt eine Menge von Schlüsselwertpaaren zugehörig zu einem Asset. Assets und die darauf aufbauende Business Netzwerk Definition wird in Kapitel 5.5.2 näher erläutert. Damit ein Teilnehmer sich gegenüber dem Ledger authentifizieren kann verwendet das Hyperledger Fabric Framework eine Public-Key-Infrastructure (PKI). Diese PKI wird realisiert durch einen Member Ship Provider (MSP) genannten Service. Dieser Service kümmert sich um die Vergabe und den Abgleich von digitalen Identitäten mit denen sich User gegenüber dem Ledger authentifizieren können. Durch diese Designentscheidung wird das Blockchain Netzwerk ein private permissioned Netzwerk und realisiert damit Anforderung *A2.4*. Eine Anonymität der Teilnehmer ist innerhalb der Lieferkette ohnehin kaum gegeben und nur indirekt über mehrere Produktionsschritte erreichbar. Ein Ziel des Systems ist es transparenz für die Nutzer des Netzwerks zu bieten, aus diesem Grund wurde der Aspekt Anonymität außer acht gelassen.

Neue Transaktionen im Netzwerk werden über ein User Interface durch einen Teilnehmer ausgelöst. Jede Transaktion durchläuft dann einen dreistufigen Prozess bis sie schlussendlich dem Ledger hinzugefügt wird (Abbildung 17). Die einzelnen Stufen sind

- Endorsement,
- Ordering,
- Validation.

Das Endorsement beginnt mit der Übermittlung der Transaktion zu einem Peer. Dieser verteilt die Transaktion im Netzwerk. Jeder Peer simuliert und prüft nun die Transaktion anhand der Geschäftslogik (Smart Contract). Nach erfolgreicher Prüfung erhält der Transaktionssteller eine Endorsement Signatur, welche an den Orderer Service weitergeleitet wird (Ordering). An dieser Stelle wird die Konsensmechanik durchlaufen und wenn ein Konsens über das Ergebnis der Transaktion hergestellt wurde (Validation) gibt der Orderer Service die Transaktion für das Ledger frei.

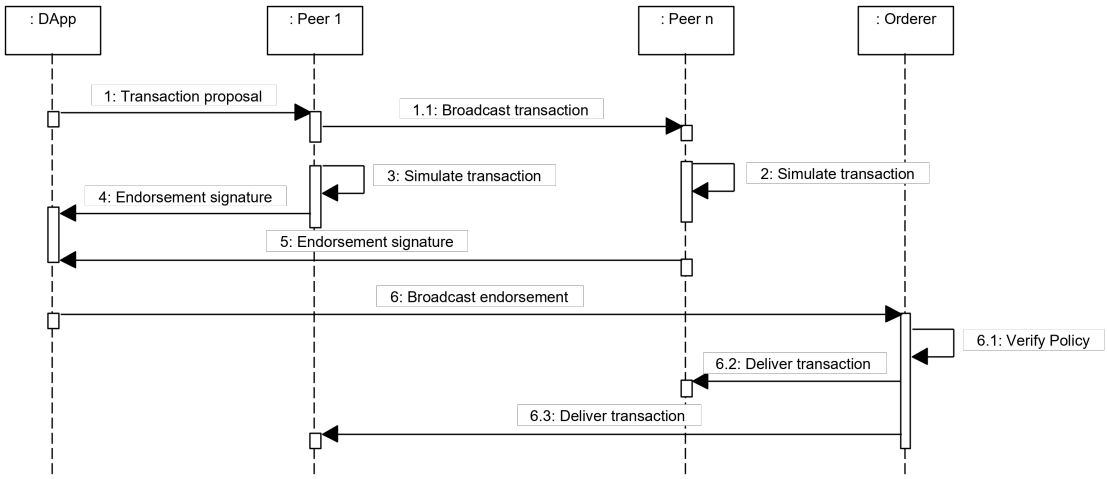

Abbildung 17: Transaction Flow in Anlehnung an (Choudhury et al., 2018)

5.5.2. Smart Contracts / Business Netzwerk Modell

Smart Contracts sind abgebildet als *Transaction Processor Functions*. Zusätzlich werden für Hyperledger Fabric noch Definitionen zu *Participants*, *Assets* und *Queries* erfasst. Diese Komponenten bilden zusammen das *Business Network Archive (BNA)* und stellen die Geschäftslogik dar. Anforderung *A1.1* wird mit dem *Business Network Archive* realisiert. Hyperledger Fabric bringt eine eigene Modellierungssprache mit. Mit dieser Sprache werden *Assets*, *Participants* und *Transactions* modelliert. Die Sprache unterstützt Vererbung, Templates und abstrakte Klassen, ähnlich der objektorientierten Programmierung.

Damit Transaktionen im Netzwerk verarbeitet werden können, müssen zugehörige *Assets* modelliert und später im System angelegt werden. Für die transparente, lückenlose Rückverfolgung von Chargen sind folgende *Assets* modelliert worden:

- *Material*
- *Batch*
- *BatchNetwork*

Mit einem *Asset Material* werden die Tiere bzw. Erzeugnisse der Produktionsbetriebe abgebildet. Sie werden identifiziert über eine global eindeutige Nummer. Eine Charge ist definiert als *Batch* und ebenfalls wie ein *Material* global eindeutig identifizierbar. Zur darstellung eines Chargengraphs dient die Entität *BatchNetwork*. Darüber hinaus werden noch *Enumerations* und *Concepts* verwendet, um das modellierte System möglichst modular halten zu können. So ist eine nachträgliche Erweiterung bzw. Anpassung ohne großen Aufwand realisierbar. Es wurden folgende Enumerations und Concepts modelliert:

- *MaterialType*
- *MaterialQuality*
- *TransportLog*
- *Location*

- *SensorData*
- *Status*

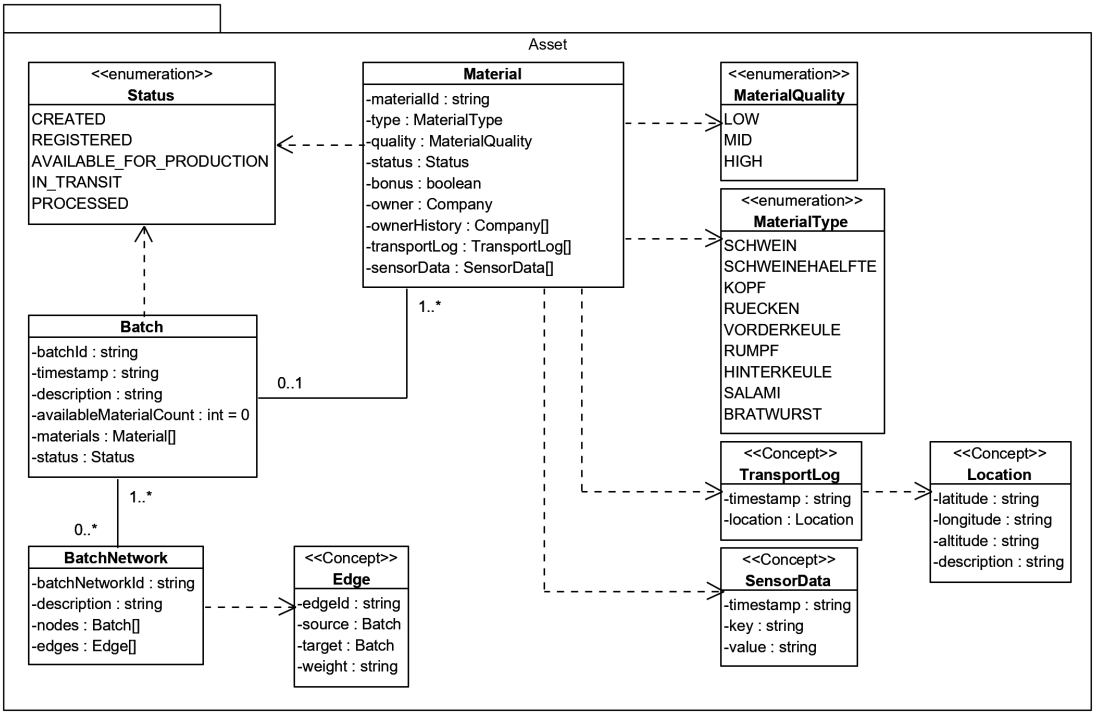

Abbildung 18: Klassendiagramm Blockchain Netzwerk *Assets*

Abbildung 18 stellt die Beziehungen zwischen den *Assets*, *Enumerations* und *Concepts* in Form eines UML Klassendiagramms dar.

Die Modellierung der *Participants* ist relativ simpel gehalten. Es gibt eine abstrakte Entität *Company* von der sich jeweils eine Teilnehmerkategorie des Blockchain Netzwerk spezialisiert. Eine *Company* wird identifiziert durch die Global Location Number (GLN)[12]. Außerdem wurde noch ein komplexer Datentyp in Form eines *Concepts* verwendet. Mit dem *Address Concept* wird eine reguläre Geschäftsadresse des Unternehmens abgebildet und als eigenes Attribut der *Company* Entität verwendet.

[12]Die GLN ist eine zentral vergebene Identifikationsnummer von der GS1-Organisation zur eindeutigen Identifikation von Betriebsstätten.

Anforderung *A2.2* wird mit dieser Datenstruktur erfüllt. In Abbildung 19 wird das beschriebene Konstrukt als Klassendiagramm dargestellt.

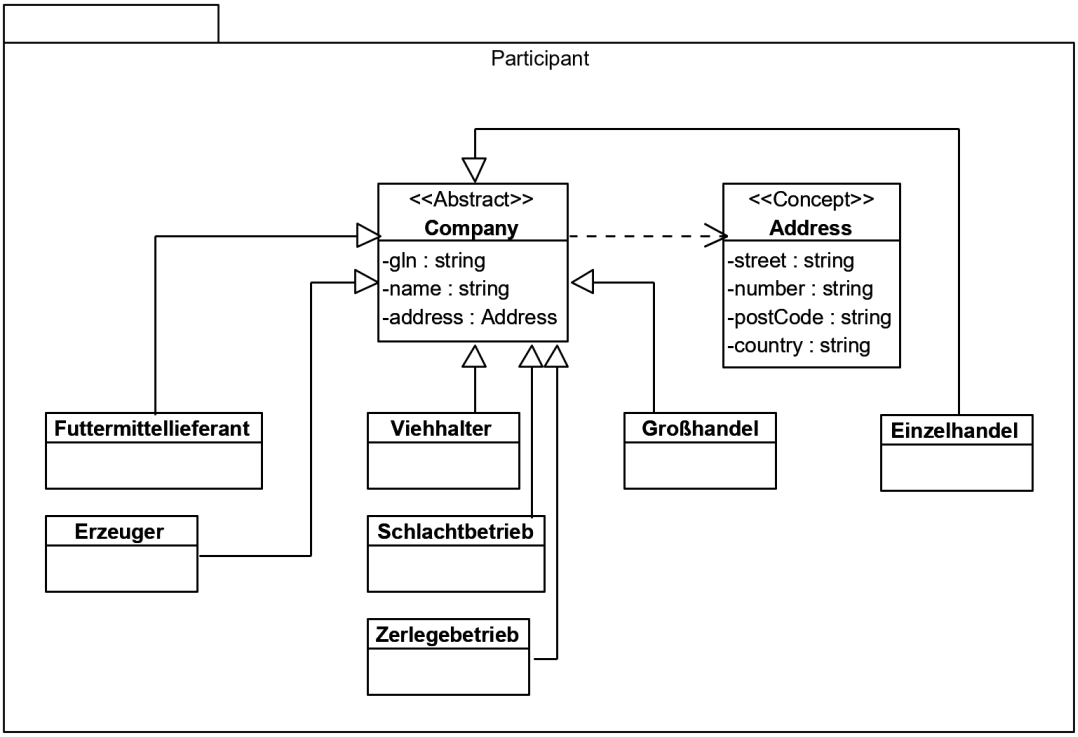

Abbildung 19: Klassendiagramm Blockchain Netzwerk *Participants*

Die dritte Komponente des *BNA* ist die Menge an *Transactions* (Abbildung 20). *Transactions* werden von *Participants* ausgelöst und sie verändern oder erzeugen *Assets*. Entsprechend wurden die Geschäftsvorgänge abgebildet die nötig sind um eine Chargenrückverfolgung zu ermöglichen (siehe Kapitel 5.2). Es wurde eine *Transaction* modelliert zum erzeugen von neuem Material - *produceMaterial*. Diese *Transaction* verlangt mehrere Parameter. Bis auf den Parameter *newMaterial* sind alle weiteren Parameter optional. *newMaterial* enthält alle Daten für ein neues Tier, das im Netzwerk registriert wird. Die optionalen Parameter werden verwendet, wenn in späteren Produktionsschritten vorhandene Erzeugnisse zu Zwischenprodukten weiterverarbeitet werden. Des weiteren sind *Transactions* modelliert mit denen die Eigentumsverhältnisse eines *Assets* verändert werden können. Außerdem lassen sich

Chargen angelegen und mit registrierten Tieren verknüpfen.

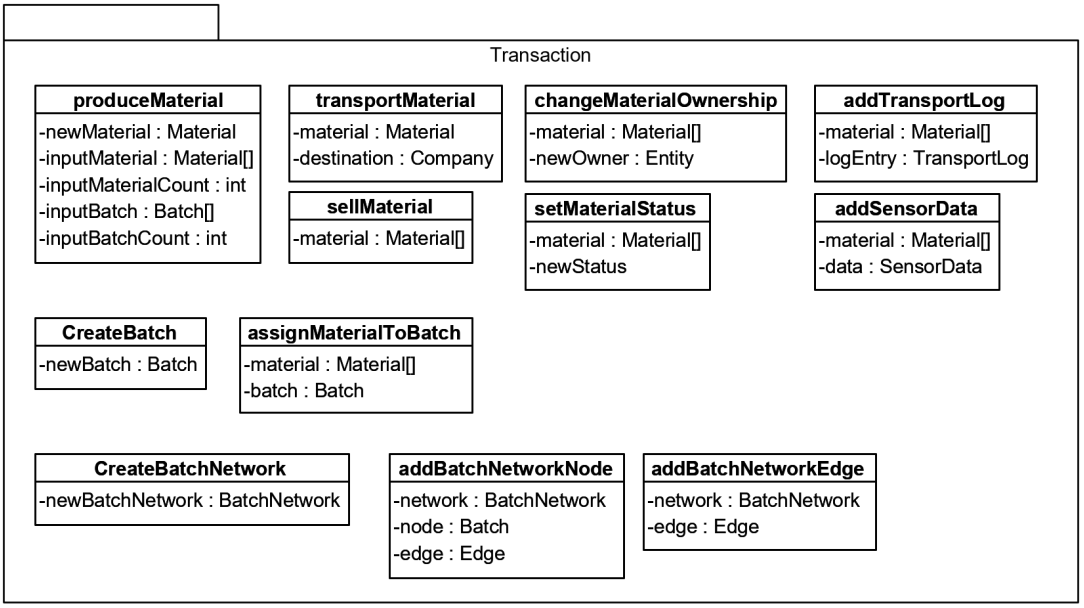

Abbildung 20: Klassendiagramm Blockchain Netzwerk *Transactions*

5.5.3. Identity Management

Administration und Interaktion mit dem System wird über ein Identity Management organisiert. Da es sich bei dem System um ein private permissioned Ledger handelt, sind per Definition (Kapitel 3.2.3) alle Teilnehmer untereinander vollständig bekannt und es gibt keine Anonymität. Dies wird durch den *Membership Service Provider* realisiert. Die verwendete Public-Key-Infrastructure (PKI) besteht dabei aus

- einer Registrierungsstelle (RA), die die Identität von Instanzen überprüft, die ihre digitalen Zertifikate in der CA speichern möchten,
- einer Zertifizierungsstelle (CA), die die digitalen Zertifikate speichert, ausstellt und signiert,
- einem zentralen Verzeichnis, d. h. einer sicheren Datenbank zum Speichern und für das Indexieren von Schlüsseln,

- einem Zertifikatsverwaltungssystem, das beispielsweise den Zugriff auf gespeicherte Zertifikate oder die Zustellung der auszugebenden Zertifikate verwaltet,
- einer Zertifikatsrichtlinie mit den Anforderungen der PKI für ihre Verfahren. Außenstehende können damit die Vertrauenswürdigkeit der PKI analysieren.

Am Beispiel der Zertifikatsausstellung soll die Funktionsweise des *Membership Service Providers* dargestellt werden. Zur Veranschaulichung des Ablaufs dient Abbildung 21.

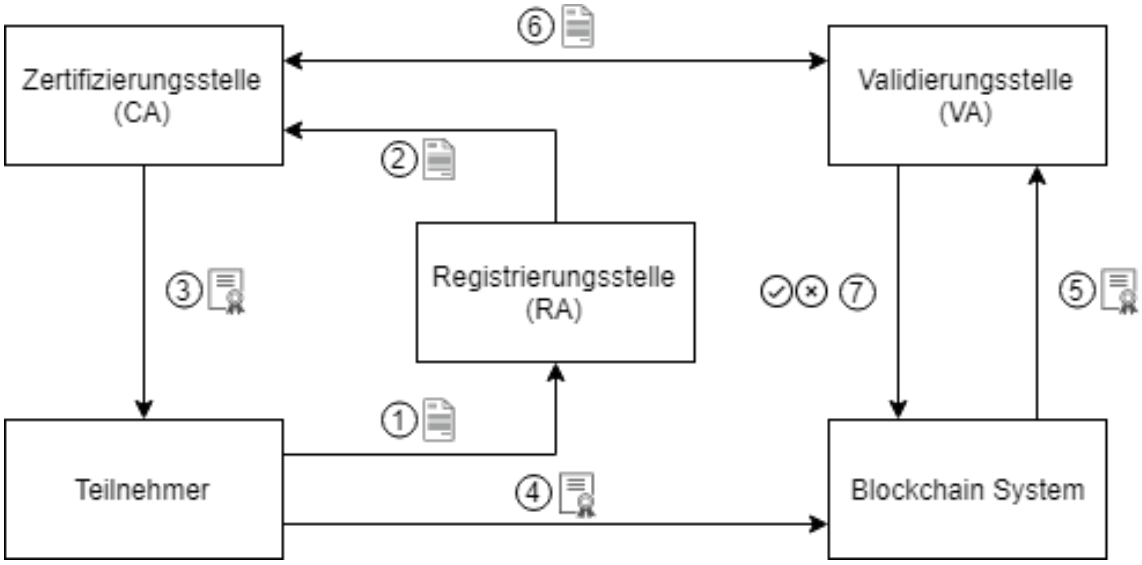

Abbildung 21: Ausstellen einer digitalen Identität für einen Teilnehmer der Blockchain

Bevor eine Transaktion ins Netzwerk zur Verarbeitung eingebracht werden kann, muss sich ein Teilnehmer gegenüber dem System authentifizieren. Hierfür ist ein gültige Ausweisdokument (in diesem Beispiel die digitale Identität) notwendig. Um diese ausgehändigt zu bekommen, werden folgende Schritte durchlaufen: Der entsprechende Teilnehmer meldet sich bei der zuständigen Stelle zur Registration (siehe Abbildung: Registrierungsstelle RA) und beantragt ein Ausweisdokument (1). Damit das Ausweisdokument eindeutig zugeordnet werden kann, ist dieses mit den für den Teilnehmer notwendigen und spezifischen Informationen ausgestattet. Die

Registrierungsstelle überprüft die hinterlegten Informationen und bestätigt (2) diese gegenüber der Zertifizierungsstelle (CA), welche im nachfolgenden Schritt das entsprechende Zertifikat (digitale Identität) ausstellt (3). Mit Hilfe dieses Zertifikats kann sich der Teilnehmer dann gegenüber dem System authentifizieren (4). Zur Überprüfung der Gültigkeit und Integrität des Dokuments wird die digitale Identität im abschließenden Schritt gegenüber einer Validierungsstelle geprüft (5). Diese gleicht alle hinterlegten Informationen der CA (6) mit dem vorliegenden Dokument ab und bestätigt im Idealfall zum einen die Echtheit der Person und zum anderen auch die Echtheit und den Inhalt des Zertifikats (7). Der Teilnehmer beweist mithilfe der digitalen Identität, dass es sich wirklich um diesen Teilnehmer handelt.

Der *Membership Service Provider* ist so konzipiert, dass er bei Bedarf auch extern bereitgestellt werden könnte. So ist den Teilnehmern freigestellt, ob sie den Dienst selber betreiben oder das gesamte Netzwerk beispielsweise durch eine externe Zertifikatstelle die Identitätsvergabe regelt.

5.5.4. User Interface / ĐApps

Endanwender sollen mit dem Gesamtsystem über GUI-Applikationen interagieren. Dazu wurden Mockups für die einzelnen Oberflächen designt. Der Einstieg erfolgt über eine sogenannte Launchpad Seite. Alle weiteren Applikationen lassen sich vom Launchpad aus erreichen. Das Launchpad dient dem Endanwender als zentrale Anlaufstelle um alle Geschäftsvorgänge abzuwickeln. Applikationen werden als Kachel in unterschiedlichen Gruppen angezeigt. Dabei wurde jeweils eine Gruppe für Asset Operationen und Chargen Operationen modelliert (siehe Abbildung 22). Die Kacheln sind nach dem Ablauf des Lebenszyklus angeordnet. Beginnend mit der Anmeldung eines neuen Tieres im Netzwerk (*Register Asset*).

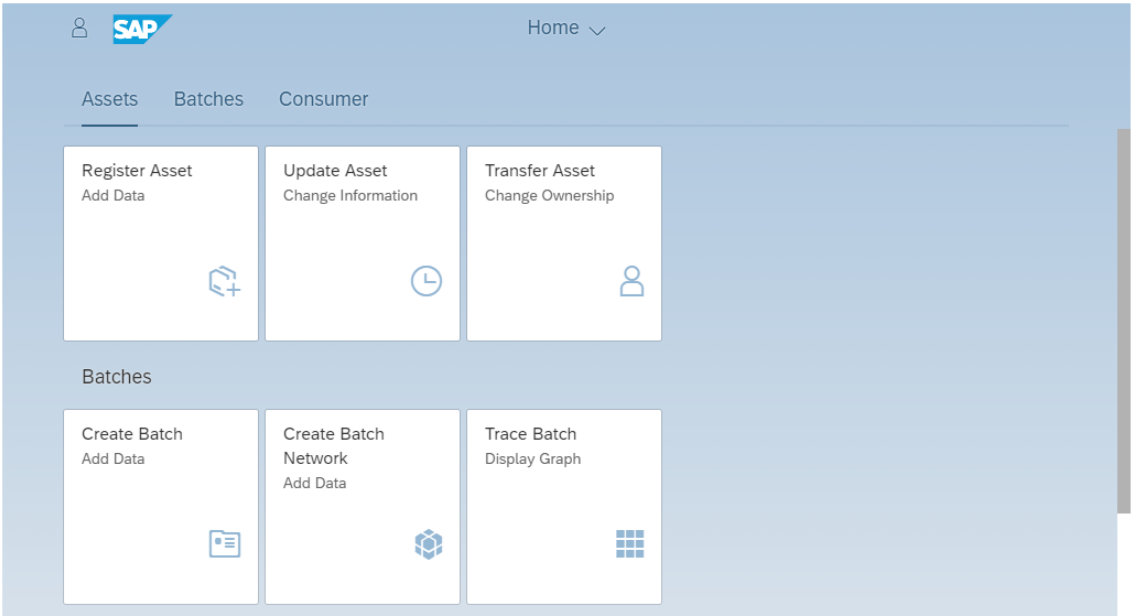

Abbildung 22: Mockup: Einstiegsseite Endanwender

Über diese Oberfläche kann ein Anwender die Eigenschaften des zu registrierenden Tieres erfassen. Zwingend nötige Informationen sind mit einem Stern am Beginn der Formularzeile markiert (Abbildung 23). So sind bei einem Ferkel beispielsweise keine Tiere weiterverarbeitet worden (Formularfeld *Processed Assets*) sondern es stellt den Anfang des Warenstroms dar. Wenn ein Ferkel zum Mastbetriebt transportiert wurde und der Mastbetrieb dann ein schlachtreifes Schwein erfassen will hat er die Möglichkeit das Ferkel bei der Registrierung des Schweins mitanzugeben. Innerhalb der Transaktionslogik kann dann auf diese Information reagiert werden. Im einfachsten Fall erfährt das verarbeitete Asset eine Statusänderung.

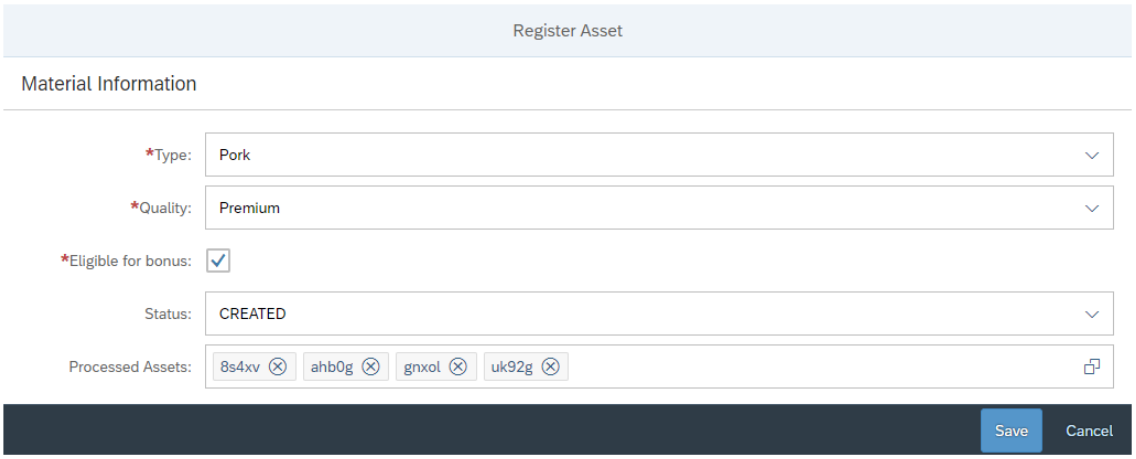

Abbildung 23: Mockup: Asset Registrierung

Des weiteren wurde eine Applikation modelliert mit der bereits erfasste Assets gepflegt werden können. Wie in Abbildung 24 durch die ausgegrauten Eingabefelder dargestellt, können bei einem vorhandenen Asset nicht alle Informationen nachträglich verändert werden. Schlüsselmerkmale wie die Identifikationsnummer werden vom System automatisch generiert beim Erfassen eines Assets und können daher nicht vom Anwender angepasst werden.

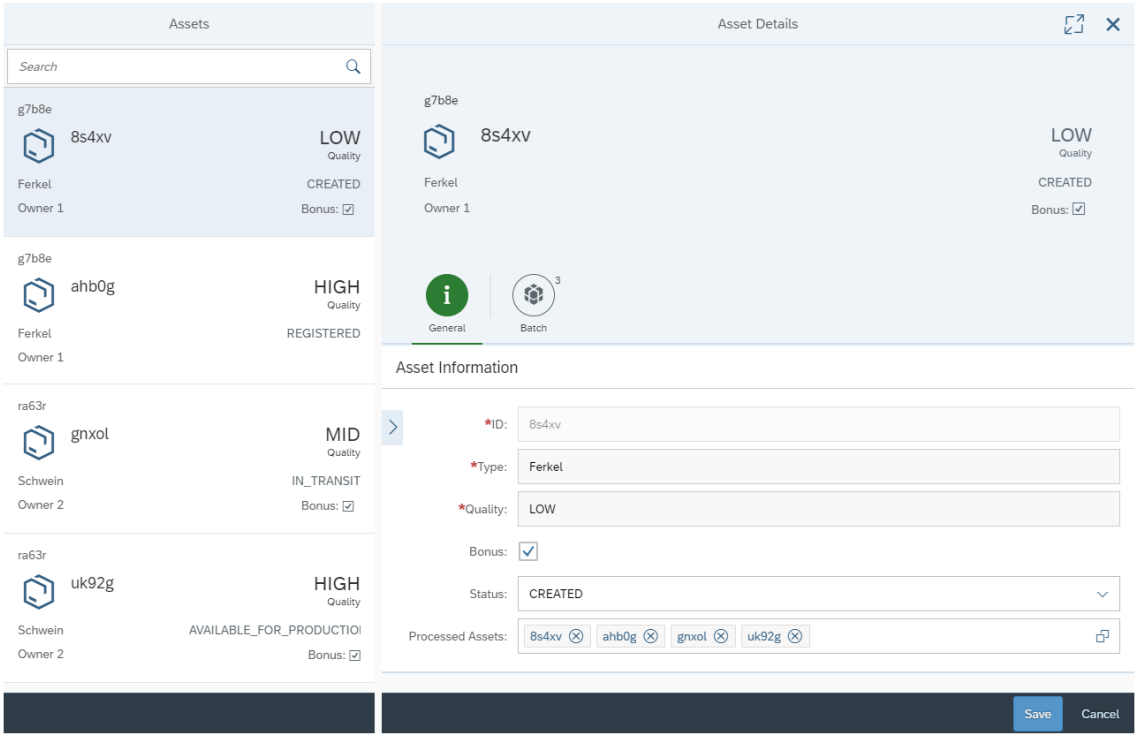

Abbildung 24: Mockup: Asset Update

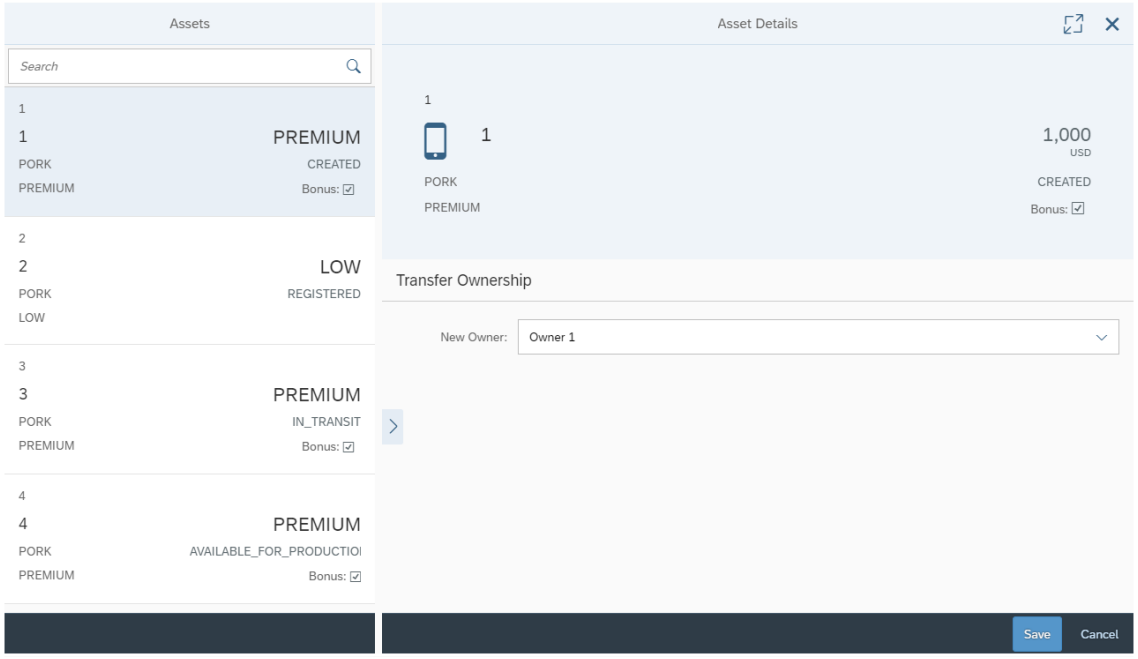

Abbildung 25: Mockup: Asset Transfer

Äquivalent zu den Assets wurden ebenfalls Applikationen zum Anlegen und Pflegen von Chargen modelliert.

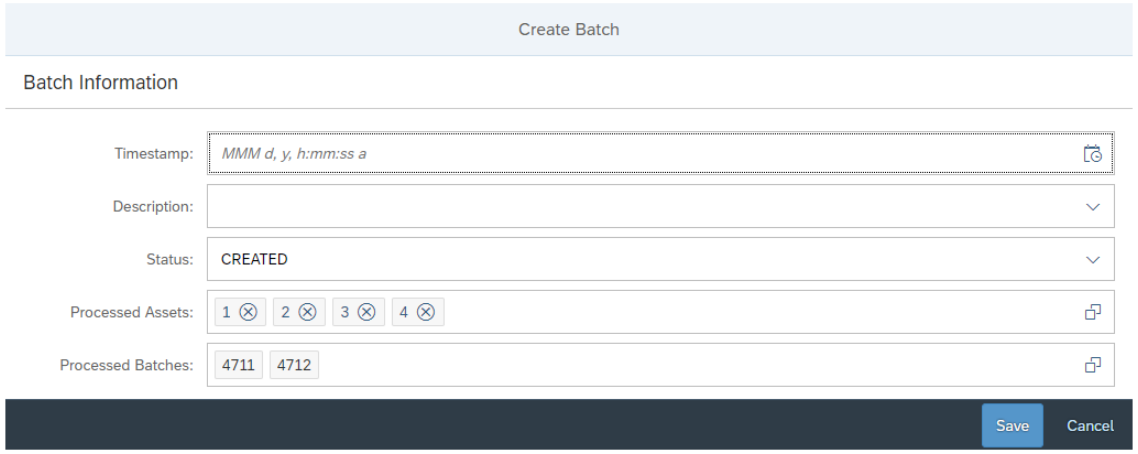

Abbildung 26: Mockup: Batch Create

5.6. Zusammenfassung Systementwurf

Mit dem Kapitel Systementwurf wurde die Methode der Anforderungserhebung beschrieben und eine Zieldefinition gegeben. Daneben sollte eine Betrachtung der Wertschöpfungskette und des Geschäftsprozess in Ist- und Soll-Variante aufzeigen an welchen Punkten ein Blockchain System im Prozess eingesetzt werden sollte um den Gesamtprozess der Chargenrückverfolgung zu unterstützen bzw. überhaupt erst möglich zu machen. Abschließend wurde der Systementwurf beschrieben unterteilt in Ledger/Konsens, Smart Contracts, Identity Management und dem User Interface. Im nächsten Kapitel wird dann die technische Umsetzung des Systementwurf für den Prototyp detailiert beschrieben.

6. Technische Umsetzung

In diesem Kapitel wird die Umsetzung des modellierten Systementwurfs als prototypische Implementierung im Detail beschrieben. Es gibt einen Einblick in den Prozess der Konfiguration eines Blockchain Netzwerks, das mehrere Unternehmen umfasst. Dabei wird Eingangs auf die zugrunde liegende Architektur des Business Netzwerks bezug genommen. Aufbauend auf dem Fundament des Business Netzwerks werden Geschäftslogik und Berechtigungssystem erläutert.

6.1. Business Netzwerk

Als Basis des Blockchain Systems dient ein Hyperledger Fabric Netzwerk. Alle Dienste des Netzwerks werden in einer virtualisierten Umgebung bereitstellt. Dazu wird die Container Technologie von Docker[13] verwendet. Zum einen bietet sich die Container Technologie zur Umsetzung eines Prototyps an, da sie sehr viel flexibler und leichtgewichtiger ist als die konventionelle Virtualisierung über Virtuelle Maschinen (Ahmed and Pierre, 2018). Zum anderen sind die Basis Komponenten zum aufspannen eines Hyperledger Fabric Netzwerks bereits von der Linux Foundation als Container Abbild bereitgestellt, was die Realisierung des Prototypen signifikant beschleunigt. Im folgenden wird die technische Umsetzung eines Peer Knotens mittels Docker beispielhaft beschrieben. Die anderen Systemkomponenten (siehe Kapitel 5.5) verhalten sich vom Aufbau her äquivalent zu einem Peer Knoten, sie sind lediglich unterschiedlich konfiguriert um verschiedene Aufgaben auszuführen. Die Netzwerke beider Unternehmen werden in diesem Fall auf der selben Maschine betrieben. In einem produktiven Umfeld würde jedes Unternehmen seine eigene Umgebung bereitstellen.

Das Prototyp Netzwerk umfasst zwei Organisationen: *Org1* und *Org2*. Das Unternehmen *Org1* verwendet den Domänennamen *org1.example.com*. Der Member Ship Provider (MSP) für *Org1* wird als *Org1MSP* bezeichnet. Das Unternehmen *Org2* verwendet den Domänennamen *org2.example.com*. Der Member Ship Provider (MSP) für *Org2* heißt *Org2MSP*.

[13] Docker basiert auf Linux Techniken wie *Cgroups* und *Namespaces*, um isolierte Umgebungen innerhalb eines Hostsystems bereitzustellen (Bengel et al., 2008; Öggl, 2019).

Netzwerk Komponenten

Das Hyperledger Fabric Netzwerk besteht insgesamt aus den folgenden Komponenten und Schnittstellen:

- Zwei Peer Knoten für *Org1*
  - *peer0.org1.example.com*
  - *peer1.org1.example.com*
- Eine CA für *Org1* (*ca.org1.example.com*)
- Zwei Peer Knoten für *Org2*
  - *peer0.org2.example.com*
  - *peer1.org2.example.com*
- Eine CA für *Org2* (*ca.org2.example.com*)
- Ein einzelner Orderer Peer (*orderer.example.com*)

Jede dieser Komponenten stellt einen Docker Container dar und ist auf Netzwerkebene über seinen Hostnamen ansprechbar. Die gesamte Netzwerkkommunikation ist über das Transport Layer Security (TLS)-Protokoll[14] abgesichert. Aus diesem Grund müssen alle Zertifikate der CA auf dem Hostsystem zur Verfügung stehen, damit eine Kommunikation mit dem Netzwerk stattfinden kann. Für Organisation *Org1* ist ein Administrator User angelegt mit Namen *Admin@org1.example.com*. Ebenfalls ist für Organisation *Org2* ein Administrator User angelegt der *Admin@org2.example.com* heißt. Zusätzlich zu den Administrator Usern der Organisationen ist die CA mit einem Standard User konfiguriert. Der CA User besitzt im gegensatz zu den Administrator Usern keine Berechtigungen, um Smart Contracts (Chaincode) auf Peers des Netzwerk zu installieren. Damit die Peer Administatoren sich mit dem Netzwerk verbinden können wird ein Verbindungsprofil benötigt. In diesem Verbindungsprofil werden alle Komponenten des Netzwerks definiert und die zugehörigen TLS Zertifikate hinterlegt (siehe Anhang 6). Verbindungsprofil und die digitale Identität des

[14] TLS ist ein hybrides Verschlüsselungsprotokoll, um Datenübertragungen vor Angriffen zu schützen (Dierks and Rescorla, 2008).

Administrator Users, bestehend aus Zertifikat und privatem Schlüssel, bilden zusammen die sogenannte *Business Network Card*. Hiermit kann sich der Administrator User über eine Hyperledger Fabric Command Line Interface (CLI) mit dem Netzwerk verbinden und Befehle absetzen.

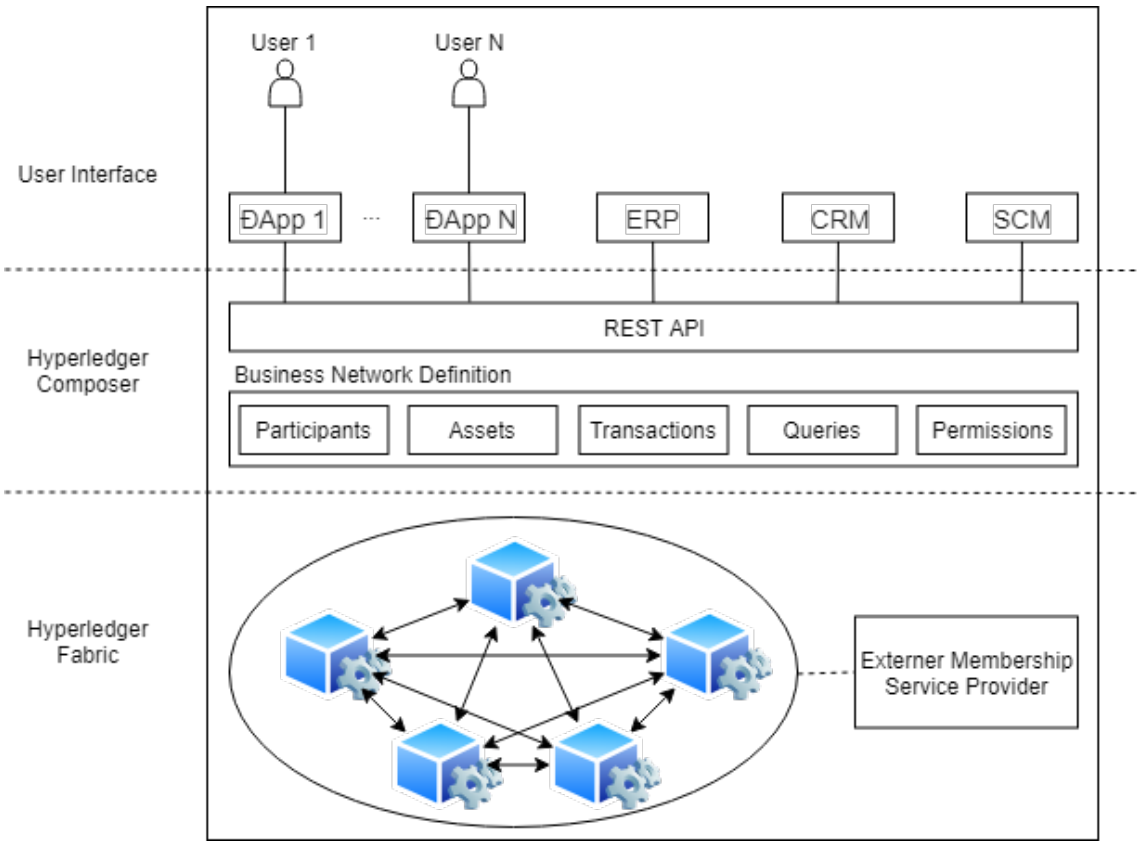

Abbildung 27: Gesamtsystem Prototyp

Nachdem starten der Docker Container lässt sich ein einfacher Smoke Test[15] durchführen, um sicherzustellen das das Netzwerk ordnungsgemäß hochgefahren wurde und alle Knoten arbeiten. Docker bietet zum Mangement der Container ein CLI an. Hiermit lässt sich der Smoke Test mit einem Einzeiler auf dem Terminal ausführen. Damit ist die Basiskonfiguration des Systems abgeschlossen und das Peer Netzwerk ist aufgespannt (siehe Abbildung 27 Abschnitt *Hyperledger Fabric*). Im

[15]Mit einem Smoke Test sollen grundlegende Probleme bei einer Software oder einem System offengelegt werden, bevor die Entwicklung von folge Komponenten begonnen wird (Everett, 2007).

aktuellen Zustand kann das Netzwerk noch keine Transaktionen erzeugen oder verarbeiten. Dazu muss erst noch die im nächsten Kapitel beschriebene Geschäftslogik durch einen Administrator User auf einem Peer Knoten des Unternehmens installiert und instantiiert werden.

6.2. Smart Contracts

Smart Contracts heißen im Hyperledger Model *Chaincode*. Sie setzen sich aus vier Elementen zusammen. Model, Logik, Zugriffskontrolle und Abfragedefinition bilden das sog. *Business Network Archive* (*BNA*). Das *BNA* lässt sich in jedes mit Hyperledger Fabric aufgespannte Blockchain Netzwerk deployen. Die Funktionsweise eines Smart Contracts soll hier am Beispiel des Eigentumswechsels eines Materials näher erläutert werden. Dazu wird auf jedes der vier Elemente eines *BNA* eingegangen, um den strukturellen Aufbau zu zeigen. Im Sinne des Models werden ein *Participant*, ein *Asset* und eine *Transaction* definiert wie in Listing 1. Die eigentliche Verarbeitungslogik wird gesondert von der Datenstruktur definiert (Listing 2). In diesem Fall wurde die Logik in der Programmiersprache JavaScript implementiert.

```
1  namespace io.dev.foodchain
2
3  abstract participant Company identified by gln {
4      o String gln
5      o String name
6  }
7
8  participant Farmer extends Company {}
9
10 asset Material identified by materialId {
11     o String materialId
12     --> Company owner optional
13 }
14
15 transaction changeMaterialOwnership {
16     --> Material material
17     --> Company newOwner
18 }
```

Listing 1: Model Example Definition

Zeile 1 in Listing 1 definiert einen Namensraum für das gesamte Model. In einem produktiven Scenario würde ein Model erheblich größer sein, als das im Prototyp verwendete vereinfachte Model. Damit bei steigender Komplexität des abzubildenden Models Überblick und Wartbarkeit erhalten bleiben lässt sich das Model über mehrere Dateien abbilden und über den Namensraum auf logischer Ebene miteinander verknüpfen. Zeile 3 bis einschließlich Zeile 6 zeigt die Definition der abstrakten Klasse *Company* vom Typ *Participant*. Diese Definition wird in Zeile 10 konkret ausgeprägt durch die Klasse *Farmer*. Äquivalent dazu werden auch alle anderen Teilnehmer der Wertschöpfungskette implementiert. Zeile 10 bis Zeile 14 zeigt die Implementierung des Assets *Material*. Über die Eigenschaft *owner* (Zeile 12) wird ein *Material* später immer einem eindeutigen Besitzer zugeordnet. Die Eigenschaft *owner* wurde dabei als direkte Ressourcenverknüpfung implementiert. Eine Ressourcenverknüpfung im Hyperledger Model lässt sich mit einer Fremdschlüsselbeziehung in einem relationalen Datenbankschema vergleichen. Die Eigenschaft kann in diesem Fall nur Werte annehmen die eine gültige Ausprägung der abstrakten Klasse *Company* sind und damit auch allen konkreten Ausprägungen dieser Klasse. Um das Beispiel einfach und verständlich zu halten wurden die Definitionen der *Participants* und *Assets* in verkürzter Form abgebildet. Das vollständige Prototypen Model befindet sich im Anhang.

```
1 /**
2 * Change material ownership transaction
3 * @param {io.dev.foodchain.changeMaterialOwnership} tx
4 * @transaction
5 */
6 async function changeMaterialOwnership(tx) {
7     // input parameter
8     const oMaterial = tx.material;
9     const oNewOwner = tx.newOwner;
10    const oActualOwner = tx.material.owner;
11
12    // checks
13    //// check if proposed input material exists
14    const oMaterialRegistry = await getAssetRegistry(NS + '.Material');
15    const bMaterialExists = await oMaterialRegistry.exists(oMaterial.
      getIdentifier());
```

```
16      if(!bMaterialExists) {
17          throw new Error('Input material "' + oMaterial.getFullyQualifiedName() +
        '" does not exist.');
18      }
19      //// AND belong to tx issuer (should be checked by permissions file)
20      if (oMaterial.owner !== getCurrentParticipant()) {
21          throw new Error('You are not allowed to change asset: "' + oMaterial.
        getFullyQualifiedName() + '".');
22      }
23      //// check if proposed new owner exists
24      const oParticipantRegistry = await getParticipantRegistry(oNewOwner.
        getNamespace());
25      const bNewOwnerExists = await oParticipantRegistry.exists(oNewOwner.
        getIdentifier());
26      if(!bNewOwnerExists) {
27          throw new Error('New owner "' + oNewOwner.getFullyQualifiedName() + '"
        does not exist.');
28      }
29
30      // business logic
31      //// put actual owner to the history stack
32      oMaterial.ownerHistory.push(oActualOwner);
33      //// set new owner for asset
34      oMaterial.owner = oNewOwner;
35      //// update asset registry
36      await oMaterialRegistry.update(oMaterial);
37
38      // event emitting
39      //// create and emit default tx event
40
41
42      // set return value
43
44  }
```

Listing 2: Transaction Processor Function *changeMaterialOwnership(tx)*

```
1  rule OwnerHasFullAccessToTheirAssets {
2      description: "Allow all participants full access to their assets"
3      participant(p): "io.dev.foodchain.*"
```

```
4     operation: ALL
5     resource(r): "io.dev.foodchain.*"
6     condition: (r.owner.getIdentifier() === p.getIdentifier())
7     action: ALLOW
8 }
```

Listing 3: Berechtigungsdefinition

```
1 query selectMaterialsWithBonusCondition {
2   description: "Select materials based on bonus condition"
3   statement:
4       SELECT io.dev.foodchain.Material
5           WHERE (bonus == true)
6 }
```

Listing 4: Abfragedefinition

6.3. User Interface

1. SAP Build
2. SAPUI5

6.4. Zusammenfassung technische Umsetzung

7. Evaluation

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

7.1. Experten Interviews

7.2. Kennzahlen

Transaktionskosten

Transaktionsgeschwindigkeit

Datenverfügbarkeit

Innovationskraft

Transaktional

Geschwindigkeit

Transparenz

Vertrauen

Unveränderlichkeit

Geschäftsregeln

PKI

Platform

8. Abschlussbetrachtung

Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt.

8.1. Zusammenfassung

Falsches Üben von Xylophonmusik quält jeden größeren Zwerg. Heizölrückstoßabdämpfung. Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim.
Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark. „Fix, Schwyz!“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf Boxkämpfer quer über den großen Sylter Deich. Falsches Üben von Xylophonmusik quält jeden größeren Zwerg. Heizölrückstoßabdämpfung. Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim. Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark. „Fix, Schwyz!“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf

8.2. Reflexion

Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim. Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark.
„Fix, Schwyz“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf Boxkämpfer quer über den großen Sylter Deich. Falsches Üben von Xylophonmusik quält jeden größeren Zwerg. Heizölrückstoßabdämpfung. Zwei flinke Boxer jagen die quirlige Eva und

ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern.
Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim. Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark. „Fix, Schwyz“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf Boxkämpfer quer über den großen Sylter Deich.

8.3. Ausblick

Falsches Üben von Xylophonmusik quält jeden größeren Zwerg. Heizölrückstoßabdämpfung. Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim.
Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark. „Fix, Schwyz“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf Boxkämpfer quer über den großen Sylter Deich. Falsches Üben von Xylophonmusik quält jeden größeren Zwerg. Heizölrückstoßabdämpfung. Zwei flinke Boxer jagen die quirlige Eva und ihren Mops durch Sylt. Franz jagt im komplett verwahrlosten Taxi quer durch Bayern. Zwölf Boxkämpfer jagen Viktor quer über den großen Sylter Deich. Vogel Quax zwickt Johnys Pferd Bim. Sylvia wagt quick den Jux bei Pforzheim. Polyfon zwitschernd aßen Mäxchens Vögel Rüben, Joghurt und Quark. „Fix, Schwyz“ quäkt Jürgen blöd vom Paß. Victor jagt zwölf

A. Anhang

Funktionale Anforderungen

ID	Anforderung	Quelle
A1.1	Das Gesamtsystem muss fähig sein den Lebenszyklus eines Tieres vom Erzeuger bis zum Lebensmitteleinzelhandel abzubilden.	*Wissensch. Kontext*
A1.1.1	Das Gesamtsystem muss fähig sein Tiere anzulegen/registrieren.	
A1.1.2	Das Gesamtsystem muss fähig sein Tiere und Chargen einander zuzuordnen.	
A1.1.3	Das Gesamtsystem muss fähig sein Tiere zwischen Teilnehmern zu transferieren im Sinne eines Eigentumswechsel.	
A1.2	Das Gesamtsystem muss eine generische Schnittstelle zur Kommunikation mit dem Ledger anbieten.	*Partner*
A1.3	Das Gesamtsystem muss fähig sein Transaktionsdaten manipulationssicher speichern zu können.	*Partner*
A1.4	Das Gesamtsystem muss fähig sein den Lebenszyklus einer Charge abzubilden.	*Partner*
A1.4.1	Das Gesamtsystem muss fähig sein Chargen anzulegen.	
A1.4.2	Das Gesamtsystem muss fähig sein Chargen und Tiere einander zuzuordnen.	

Tabelle 5: Funktionale Anforderungen

A.1. Rahmenbedingungen

ID	Anforderung	Quelle
A2.1	Der Prototyp muss mit der Hyperledger Fabric Blockchain Technologie konzipiert und implementiert werden.	*Partner*
A2.2	Der Prototyp bildet die Teilnehmer der Wirtschöpfungskette vom Erzeuger bis zum Lebensmitteleinzelhandel ab.	*Partner*
A2.3	Der Prototyp fokussiert sich bei der Transaktionsabwicklung auf die Tierart Schwein. (Verminderte Komplexität)	*Partner*
A2.4	Das Gesamtsystem muss in einer abgeschlossenen Umgebung gehosted und vor pseudonymen Zugriff geschützt sein.	*Partner*

Tabelle 6: Funktionale Anforderungen

A.2. Qualitätsanforderungen

ID	Anforderung	Quelle
A3.1	Die Architektur des Systems muss eine nachträgliche Erweiterung ermöglichen, um weitere Geschäftszweige abbilden zu können.	*Wissensch. Kontext*
A3.2	Die Architektur des Systems muss mindestens eine konstante Performance bei steigender Teilnehmerzahl.	*Partner*
A3.3	Das System muss auch bei Ausfall oder Komprimitierung eines oder mehrerer Teilnehmer konsistent und stabil weiter arbeiten.	*Partner*

Tabelle 7: Funktionale Anforderungen

A.3. Listings

A.3.1. Listing A

```
1  FROM golang:1.11.5
2
3  ENV DEBIAN_FRONTEND noninteractive
4  ENV FABRIC_ROOT=$GOPATH/src/github.com/hyperledger/fabric
5  ENV CHAINTOOL_RELEASE=1.1.2
6
7  # Architecture of the node
8  ENV ARCH=amd64
9  # version for the base images (baseos, baseimage, ccenv, etc.), used in core.yaml
       as BaseVersion
10 ENV BASEIMAGE_RELEASE=0.4.14
11 # BASE_VERSION is required in core.yaml for the runtime fabric-baseos
12 ENV BASE_VERSION=1.4.0
13 # version for the peer/orderer binaries, the community version tracks the hash
      value like 1.0.0-snapshot-51b7e85
14 # PROJECT_VERSION is required in core.yaml to build image for cc container
15 ENV PROJECT_VERSION=1.4.0
16 # generic golang cc builder environment (core.yaml): builder: $(DOCKER_NS)/fabric
      -ccenv:$(ARCH)-$(PROJECT_VERSION)
17 ENV DOCKER_NS=hyperledger
18 # for golang or car's baseos for cc runtime: $(BASE_DOCKER_NS)/fabric-baseos:$(
      ARCH)-$(BASEIMAGE_RELEASE)
19 ENV BASE_DOCKER_NS=hyperledger
20 ENV LD_FLAGS="-X github.com/hyperledger/fabric/common/metadata.Version=${
      BASE_VERSION} \
21     -X github.com/hyperledger/fabric/common/metadata.BaseVersion=${
      BASEIMAGE_RELEASE} \
22     -X github.com/hyperledger/fabric/common/metadata.BaseDockerLabel=org.
      hyperledger.fabric \
23     -X github.com/hyperledger/fabric/common/metadata.DockerNamespace=hyperledger
      \
24     -X github.com/hyperledger/fabric/common/metadata.BaseDockerNamespace=
      hyperledger \
25     -X github.com/hyperledger/fabric/common/metadata.Experimental=true \
26     -linkmode external -extldflags '-static -lpthread'"
27
```

```
28 # Peer config path
29 ENV FABRIC_CFG_PATH=/etc/hyperledger/fabric
30 RUN mkdir -p /var/hyperledger/db \
31     /var/hyperledger/production \
32     $GOPATH/src/github.com/hyperledger \
33     $FABRIC_CFG_PATH \
34     /chaincode/input \
35     /chaincode/output
36
37 # Install development dependencies
38 RUN apt-get update \
39         && apt-get install -y apt-utils python-dev \
40         && apt-get install -y libsnappy-dev zlib1g-dev libbz2-dev libyaml-dev
       libltdl-dev libtool \
41         && apt-get install -y python-pip \
42         && apt-get install -y tree jq unzip\
43         && rm -rf /var/cache/apt
44
45 # install chaintool
46 #RUN curl -L https://github.com/hyperledger/fabric-chaintool/releases/download/v0
       .10.3/chaintool > /usr/local/bin/chaintool \
47 RUN curl -fL https://nexus.hyperledger.org/content/repositories/releases/org/
       hyperledger/fabric/hyperledger-fabric/chaintool-${CHAINTOOL_RELEASE}/
       hyperledger-fabric-chaintool-${CHAINTOOL_RELEASE}.jar > /usr/local/bin/
       chaintool \
48     && chmod a+x /usr/local/bin/chaintool
49
50 # install gotools
51 RUN go get github.com/golang/protobuf/protoc-gen-go \
52     && go get github.com/maxbrunsfeld/counterfeiter \
53     && go get github.com/axw/gocov/... \
54     && go get github.com/AlekSi/gocov-xml \
55     && go get golang.org/x/tools/cmd/goimports \
56     && go get golang.org/x/lint/golint \
57     && go get github.com/estesp/manifest-tool \
58     && go get github.com/client9/misspell/cmd/misspell \
59     && go get github.com/estesp/manifest-tool \
60     && go get github.com/onsi/ginkgo/ginkgo
61
```

```
62 # Clone the Hyperledger Fabric code and cp sample config files
63 RUN cd $GOPATH/src/github.com/hyperledger \
64     && git clone --single-branch -b release-1.4 --depth 1 http://gerrit.
       hyperledger.org/r/fabric \
65     && cp $FABRIC_ROOT/devenv/limits.conf /etc/security/limits.conf \
66     && cp -r $FABRIC_ROOT/sampleconfig/* $FABRIC_CFG_PATH/ \
67     && cp $FABRIC_ROOT/examples/cluster/config/configtx.yaml $FABRIC_CFG_PATH/ \
68     && cp $FABRIC_ROOT/examples/cluster/config/cryptogen.yaml $FABRIC_CFG_PATH/
69
70 # install configtxgen, cryptogen and configtxlator
71 RUN cd $FABRIC_ROOT/ \
72     && go install -tags "experimental" -ldflags "${LD_FLAGS}" github.com/
       hyperledger/fabric/common/tools/configtxgen \
73     && go install -tags "experimental" -ldflags "${LD_FLAGS}" github.com/
       hyperledger/fabric/common/tools/cryptogen \
74     && go install -tags "experimental" -ldflags "${LD_FLAGS}" github.com/
       hyperledger/fabric/common/tools/configtxlator
75
76 # Install eventsclient
77 RUN cd $FABRIC_ROOT/examples/events/eventsclient \
78     && go install \
79     && go clean
80
81 # Install discover cmd
82 RUN CGO_CFLAGS=" " go install -tags "experimental" -ldflags "-X github.com/
       hyperledger/fabric/cmd/discover/metadata.Version=${BASE_VERSION}" github.com/
       hyperledger/fabric/cmd/discover
83
84 # The data and config dir, can map external one with -v
85 VOLUME /var/hyperledger
86 #VOLUME /etc/hyperledger/fabric
87
88 # temporarily fix the `go list` complain problem, which is required in chaincode
       packaging, see core/chaincode/platforms/golang/platform.go#
       GetDepoymentPayload
89 ENV GOROOT=/usr/local/go
90
91 WORKDIR $FABRIC_ROOT
92
```

```
93 # This is only a workaround for current hard-coded problem when using as fabric-
       baseimage.
94 RUN ln -s $GOPATH /opt/gopath
95 LABEL org.hyperledger.fabric.version=${PROJECT_VERSION} \
96     org.hyperledger.fabric.base.version=${BASEIMAGE_RELEASE}
```

Listing 5: Hyperledger Fabric Peer *Dockerfile*

Listing B

```
1     {
2         "name": "hlfv1",
3         "x-type": "hlfv1",
4         "x-commitTimeout": 300,
5         "version": "1.0.0",
6         "client": {
7             "organization": "Org1",
8             "connection": {
9                 "timeout": {
10                    "peer": {
11                        "endorser": "300",
12                        "eventHub": "300",
13                        "eventReg": "300"
14                    },
15                    "orderer": "300"
16                }
17            }
18        },
19        "channels": {
20            "composerchannel": {
21                "orderers": [
22                    "orderer.example.com"
23                ],
24                "peers": {
25                    "peer0.org1.example.com": {
26                        "endorsingPeer": true,
27                        "chaincodeQuery": true,
28                        "ledgerQuery": true,
29                        "eventSource": true
30                    }
31                }
```

```
32              }
33          },
34          "organizations": {
35              "Org1": {
36                  "mspid": "Org1MSP",
37                  "peers": [
38                      "peer0.org1.example.com"
39                  ],
40                  "certificateAuthorities": [
41                      "ca.org1.example.com"
42                  ]
43              }
44          },
45          "orderers": {
46              "orderer.example.com": {
47                  "url": "grpc://orderer.example.com:7050"
48              }
49          },
50          "peers": {
51              "peer0.org1.example.com": {
52                  "url": "grpc://peer0.org1.example.com:7051"
53              }
54          },
55          "certificateAuthorities": {
56              "ca.org1.example.com": {
57                  "url": "http://ca.org1.example.com:7054",
58                  "caName": "ca.org1.example.com"
59              }
60          }
61      }
```

Listing 6: Hyperledger Fabric Network *Connection Profile*

B. Literaturverzeichnis

Ahmed, A. and Pierre, G. (2018). Docker Container Deployment in Fog Computing Infrastructures. In *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE.

allgemeine fleischer zeitung (2011). Weg von der Insellösung - Tönnies will GS1-Standard in F-Trace einbinden. *afz - allgemeine fleischer zeitung*, (33).

Andersen, D., Balakrishnan, H., Kaashoek, F., and Morris, R. (2001). Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*. ACM Press.

Back, A. (2002). Hashcash - A Denial of Service Counter-Measure. http://www.hashcash.org/papers/hashcash.pdf. abgerufen am 15.08.2019.

Beck, M. et al. (2008). ZMP-Marktbilanz, Vieh und Fleisch 2008. *Bonn. ZMP Zentrale Markt-und Preisberichtstelle GmbH.*

Bengel, G., Baun, C., Kunze, M., and Stucky, K.-U. (2008). Virtualisierungstechniken. *Masterkurs Parallele und Verteilte Systeme: Grundlagen und Programmierung von Multicoreprozessoren, Multiprozessoren, Cluster und Grid*, pages 395–414.

Beutelspacher, A., Neumann, H. B., and Schwarzpaul, T. (2010). *Digitale Signaturen*, pages 167–171. Vieweg+Teubner, Wiesbaden.

Bundesregierung (1993). Los-Kennzeichnungs-Verordnung.

Buterin, V. (2014). White Paper. http://bit.ly/2KOC6mK. abgerufen am 23.05.2018.

Cardano (2017). Why we are building Cardano. https://goo.gl/4xcTW1. aufgerufen am 05.04.2018.

carVertical (2017). Whitepaper. https://www.carvertical.com/carvertical-whitepaper.pdf?updated=20171224. aufgerufen am 05.04.2018.

Choudhury, O., Sarker, H., Rudolph, N., Foreman, M., Fay, N., Dhuliawala, M., Sylla, I., Fairoza, N., and Das, A. (2018). Enforcing Human Subject Regulations using Blockchain and Smart Contracts. *Blockchain in Healthcare Today*.

Dick, J., Hull, E., and Jackson, K. (2017). *Requirements Engineering*. Springer International Publishing.

Die Grünen (2013). PFERDEFLEISCHSKANDAL: WO BLEIBEN DIE GESETZE?! http://bit.ly/2Do1Lkj. aufgerufen am 09.02.2019.

Dierks, T. and Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, RFC Editor. `http://www.rfc-editor.org/rfc/rfc5246.txt`.

Diffie, W. ; Hellman, M. E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., and Wang, J. (2017). Untangling Blockchain: A Data Processing View of Blockchain Systems. *CoRR*, abs/1708.05665.

Dorri, A., Kanhere, S. S., and Jurdak, R. (2017). Towards an optimized blockchain for IoT. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 173–178. ACM.

Drescher, D. (2017). *Blockchain Grundlagen : Eine Einführung in die elementaren Konzepte in 25 Schritten*. mitp, Frechen, 1. auflage. edition.

Efken, J., Deblitz, C., Kreins, P., Krug, O., Kueest, S., Peter, G., and Hass, M. (2015). Stellungnahme zur aktuellen Situation der Fleischerzeugung und Fleischwirtschaft in Deutschland.

Europa Parlament und Europäischer Rat (2002). Verordnung (EG) Nr. 178/2002 des Europäischen Parlaments und des Rates. https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32002R0178. abgerufen am 07.02.2019.

Europa Parlament und Europäischer Rat (2004). Verordnung (EG) Nr. 852/2004 des Europäischen Parlaments und des Rates. https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex:32004R0852. abgerufen am 30.03.2019.

Everett, G. D. (2007). *Software testing : testing across the entire software development life cycle*. Wiley-Interscience, Piscataway, NJ] Hoboken, N.J.

Florian Glatz, Friederike Ernst, J. L. (2018). Deutsche Regierung setzt auf Blockchain. https://goo.gl/qzFfhE. abgerufen am 05.04.2018.

Food and Drug Administration (1996). Quality System Regulation, Code of Federal Regulations 21 CFR Part 820, Verordnung zur Einführung von guten Herstellungspraktiken (Good Manufacturing Practice) für die Herstellung, Entwicklung, Validierung, Verpackung, Lagerung und Installation von Medizingeraten.

Freund, U. (1997). Die optimalen Betriebsgrößen und Standorte der Schlachthöfe in Bayern. *Fleischwirtschaft*, 77(5):404–408.

Günther, H.-O. and Tempelmeier, H. (2012). *Produktion und Logistik*.

Hevner, A. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19.

Hevner, A. (2010). Design research in information systems : theory and practice.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hull, E. (2011). Requirements engineering.

J.P.Morgan, I. (2017). Blockchain. https://goo.gl/pQ23Fb. abgerufen am 05.04.2018.

Koelsch, G. (2016). *Requirements Writing for System Engineering*. Apress.

Kuechler, B. and Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5):489–504.

Meier, A. and Stormer, H. (2018). Blockchain = distributed ledger + consensus. *HMD Praxis der Wirtschaftsinformatik*, 55(6):1139–1154.

Menezes, A. J. (1997). Handbook of applied cryptography.

Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. http://bit.ly/2KL3zWM. abgerufen am 23.05.2018.

Nolte, B. (2006). *Auswirkungen des Strukturwandels auf die Personalentwicklung in Sparkassen*. Springer.

Panetta, K. (2017). Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017. https://goo.gl/acfrrr. abgerufen am 05.04.2018.

Peffers, K., Rothenberger, M., and Kuechler, B., editors (2012). *Design Science Research in Information Systems. Advances in Theory and Practice*. Springer Berlin Heidelberg.

Petersen, B., Spiller, A., and Theuvsen, L. (2010). Vom Viehvermarkter zum Dienstleistungsprofi.

Platzer, J. (2014). *Bitcoin : kurz & gut*. O'Reilly Verlag, Köln.

Pohl, K. V. and Pohl, K. (2015). Basiswissen requirements engineering : Aus- und weiterbildung zum certified professional for requirements engineeringffoundation level nach ireb-standard.

Samaniego, M. and Deters, R. (2016). Blockchain as a service for iot. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 433–436. IEEE.

SAP SE (2019). IDocs (SAP Library. http://bit.ly/2tUpZhD. abgerufen am 06.03.2019.

Siepermann, C., Vahrenkamp, R., Siepermann, M., and Amann, M. (2015). Risikomanagement in Supply Chains : Gefahren abwehren, Chancen nutzen, Erfolg generieren.

Simon, H. A. (1996). *The sciences of the artificial*. MIT Press, 3 edition.

Steinmetz, R. and Wehrle, K. (2005). *2. What Is This "Peer-to-Peer" About?*, pages 9–16. Springer Berlin Heidelberg, Berlin, Heidelberg.

Steins, M. O. (2015). Nur eine Schnittstelle für alle Kunden und Lieferanten - Pilotprojekt zu Traceability in der O+G-Branche - GS1 Standards als einheitliche Grundlage. *Lebensmittel Zeitung*, (5).

Trepper, T. (2015). *Fundierung der Konstruktion agiler Methoden : Anpassung, Instanziierung und Evaluation der Methode PiK-AS*. Springer Fachmedien Wiesbaden, Wiesbaden s.l.

Tribis, Y., Bouchti, A. E., and Bouayad, H. (2018). Supply chain management based on blockchain: A systematic mapping study. *MATEC Web of Conferences*, 200:00020.

Trienekens, J. and Beulens, A. (2001). The implications of EU food safety legislation and consumer demands on supply chain information systems. In *11th Annual world food and agribusiness forum, Sydney*.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123.

Voss, A., Frentrup, M., and Theuvsen, L. (2010). Geschäftsmodelle in kleinen und mittelständischen Unternehmen: Empirische Ergebnisse zu Strategien im Agribusiness. *Strategien von kleinen und mittleren Unternehmen. Lohmar*, pages 117–142.

Wegner-Hambloch, S. (2004). *Rückverfolgbarkeit in der Praxis: Artikel 18 und 19 der VO (EG) Nr. 178/2002 schnell und einfach umgesetzt*. Behr's Verlag DE.

Wilde, T. and Hess, T. (2007). Forschungsmethoden der Wirtschaftsinformatik : Eine empirische Untersuchung. *Wirtschaftsinformatik*, 49(4).

Yergeau, F., Sperberg-McQueen, M., Maler, E., Paoli, J., and Bray, T. (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C recommendation, W3C. http://www.w3.org/TR/2008/REC-xml-20081126/.

Zailani, S., Arrifin, Z., Abd Wahid, N., Othman, R., and Fernando, Y. (2010). Halal traceability and halal tracking systems in strengthening halal food supply chain for food industry in Malaysia (a review). *Journal of food Technology*, 8(3):74–81.

Öggl, B. (2019). Docker : das Praxisbuch für Entwickler und DevOps-Teams.

Abschließende Erklärung

Ich versichere hiermit, dass ich meine Masterarbeit selbständig und ohne fremde Hilfe angefertigt habe, und dass ich alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlegenden Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

Oldenburg, den 8. September 2019 Nils Lutz