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:	22. Oktober 2019

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).

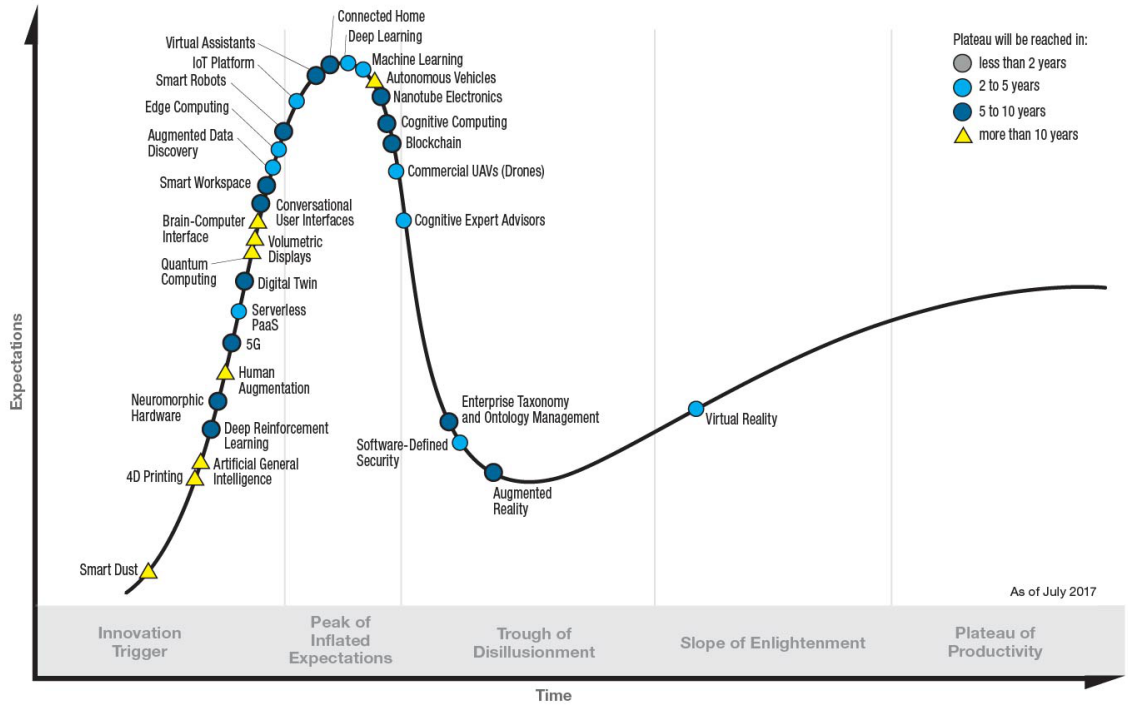

Abbildung 1: Emerging Technologies Hype Cycle 2017(Panetta, 2017)

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 und Vaishnavi, 2008; Peffers et al., 2012). Abbildung 2 stellt die drei *Design Science* Zyklen nach Hevner (2010) dar.

[3] *Hypertext Transfer Protocol (HTTP)*

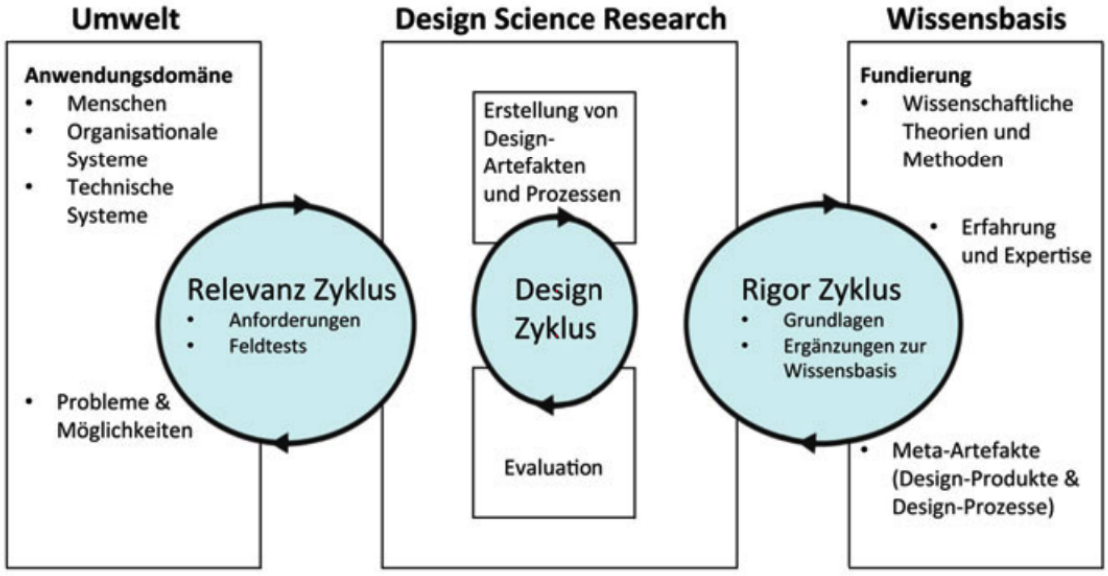

Abbildung 2: Die drei *Design Science* Zyklen nach Hevner (2010) (Trepper, 2015)

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 mittels eines Experteninterviews (siehe auch Wilde und 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

Einleitend werden in Kapitel 2 *Verwandte Arbeiten* zwei relevante Projekte aus dem selben thematischen Umfeld dieser Arbeit diskutiert. Anschließend erfolgt eine Beschreibung und Definition der grundlegenden Themenfelder *Chargenrückverfolgung* und *Blockchain-Technologie*. Dabei wird die *Chargenrückverfolgung* als solche definiert und in die Wertschöpfungskette eingeordnet. In Kapitel 4 *Lösungskonzept* soll durch zwei Analyseverfahren gezeigt werden, ob sich ein System zur *Chargenrückverfolgung* generell mittels der *Blockchain-Technologie* realisieren lässt und mit welcher konkreten Ausprägung der Technologie ein Prototyp technisch umgesetzt werden kann. Darauf folgt die Erläuterung eines möglichen Systementwurfs mit Augenmerk auf die Zieldefinition, Integration in Wertschöpfungskette und Geschäftsprozess, sowie dem eigentlichen Systementwurf zur Umsetzung eines Prototypen. In den darauffolgenden beiden Kapiteln werden die technische Umsetzung des implementierten Prototypen und eine Evaluation durch ein Experteninterview ausführlich beschrieben. Im letzten Teil der Arbeit wird eine Zusammenfassung des modellierten Systems und realisierten Prototypen gegeben, gefolgt von einem Ausblick welche

Erweiterungsmöglichkeiten System und Prototyp bieten sowie einer abschließenden Reflexion.

2. Verwandte Arbeiten

In diesem Kapitel soll ein kurzer Überblick über zwei vorhandene Lösungen im Bereich der Lebensmittelsicherheit und Supply Chain gegeben werden. Im Fokus der Betrachtung liegt die jeweils verwendete *Blockchain-Technologie* für den spezifischen *Use-Case*, sowie die tatsächliche Umsetzung und der Stand der Lösung.

2.1. Thunfisch Traceability

Der World Wildlife Fund (WWF) in Australien, Fidschi und Neuseeland hat in Zusammenarbeit mit dem US-amerikanischen Technologie-Innovator ConsenSys, dem Technologie-Implementierer TraSeable und dem Thunfischfang- und -verarbeitungsunternehmen Sea Quest Fiji Ltd., ein Pilotprojekt in der Thunfischindustrie der Pazifikinseln gestartet, das mit Hilfe der *Blockchain-Technologie* den Weg des Thunfisches vom „Köder auf den Teller“ verfolgen wird. Ziel ist es, dazu beizutragen, illegale, nicht gemeldete und unregulierte Fischerei und Menschenrechtsverletzungen in der Thunfischindustrie zu stoppen. Dazu gehören Berichte über Korruption, illegalen Handel und menschliche Sklaverei auf Thunfischfängern.

Das WWF-Pilotprojekt wird eine Kombination aus *Radiofrequenz-Identifikation (RFID)-Tags*, *Quick Response (QR)-Code-Tags* und Lesegeräten verwenden, um Informationen über die Reise eines Thunfisches an verschiedenen Punkten der Lieferkette zu sammeln. Während dieser Technologieeinsatz für das *Supply-Chain-Tracking* nicht neu ist, ist der innovative Teil, dass die gesammelten Informationen dann mit Hilfe der *Blockchain-Technologie* aufgezeichnet werden. Die Ortung beginnt, sobald der Thunfisch gefangen wird. Sobald ein Fisch gelandet ist, wird er mit einem wiederverwendbaren *RFID-Tag* auf dem Schiff befestigt. Geräte, die auf dem Schiff, am Dock und in der Verarbeitungsfabrik angebracht sind, erkennen dann die *Tags* und laden automatisch Informationen in die *Blockchain* hoch.

Nach der Verarbeitung des Fisches wird der wiederverwendbare *RFID-Tag* gegen einen kostengünstigeren *QR-Code* ausgetauscht, der an der Produktverpackung ange-

bracht wird. Der eindeutige *QR-Code* wird mit dem *Blockchain*-Datensatz verknüpft, der dem jeweiligen Fisch und seinem ursprünglichen *RFID-Tag* zugeordnet ist. Der *QR-Code* wird verwendet, um den Rest der Reise des Fisches zum Verbraucher zu verfolgen. Im Moment ist die Verknüpfung von *Tags* nicht schwierig, da sich das Projekt auf den gesamten Export konzentriert - also den gesamten frischen Fisch abzüglich Kopf, Kiemen und Eingeweiden. Etwas komplizierter wird es, wenn der Fisch in Lenden, Steaks, Würfel und Dosen zerlegt wird, aber das Projektteam ist nun in der Lage, die *QR-Code-Tags* auf den Verpackungen des verarbeiteten Fisches mit dem Datensatz des Originalfisches auf der *Blockchain* zu verknüpfen. Auch wenn es möglich sein könnte, *RFID-Tags* während des gesamten Prozesses zu verwenden, könnten die Kosten dieser *Tags* kleineren Unternehmen in der Fischwirtschaft die Teilnahme an dem System verbieten, wenn es sich ausweitet. Es besteht auch das Potenzial, in Zukunft mit *Near Field Communication (NFC)* die Fische bis zum Verbraucher zu verfolgen (McEntire und Kennedy, 2019; Visser und Hanich, 2017).

2.2. Halal Food Chain

Da Lebensmittel zwischen den verschiedenen Akteuren der Lieferkette verstreut sind, wächst die Sorge um die Gewährleistung der Lebensmittelsicherheit durch die Einführung vieler Internet- und Sachtechnologien. Neben der Unsicherheit ist die Möglichkeit, dass Halal-Lebensmittel nicht Halal sind, aufgrund der Fahrstrecke, die viele Handhabungspunkte einschließt, und des anhaltenden Risikos einer Kreuzkontamination mit Nicht-Halal-Materialien größer. Um die Fragen im Zusammenhang mit der strengen Einhaltung des Scharia-Rechts durch Halal-Produkte zu klären, gibt es einige Vorveröffentlichungen in Studien über die Rückverfolgbarkeit von Halal-Fleischprodukten (Mohammed et al., 2016). Als Lösung dafür schlug Mohamad et al. (2016) eine Methode vor, um zu bestimmen, ob das Geflügel nach islamischer Art und Weise unter Verwendung einer Untersuchung der Fleischfarbe geschlachtet wird. Junaini und Abdullah (2008) beschreiben eine mobile Unterstützungsanwendung für Muslime zur Identifizierung des Halal-Status. Kassim et al. (2012) führten ein System ein, um die Informationen von Produkten zu verifizieren und zu erkennen und damit ihren Halal-Status in Echtzeit von einem Echtzeit-Zugriff auf ihre Datenbank zu bestätigen. Bahrudin et al. (2011) schlugen

eine umfassende und geeignete Tracking & Tracing-Technologie mit *RFID* vor, um die Integrität des Halal-Produkts aufrechtzuerhalten und die gesamte Lieferkette des Halal-Produktprozesses zu unterstützen. Tan et al. (2012) fanden heraus, dass Technologien wie *Transportation Management System (TMS)*, *Wharehouse Management System (WMS)*, *Electronic Data Interchange (EDI)* und *Global Positioning System (GPS)* bei Halal-Logistikdienstleistern weit verbreitet sind. Außerdem betonte Tan et al. (2012) die Kompatibilität der *Tracking & Tracing*-Eigenschaft von *RFID* mit der Halal-Transportrichtlinie. Mohammed et al. (2016) stellen einen Rahmen für die Entwicklung eines *RFID*-fähigen *Halal Meat Supply Chain (HMSC)*-Netzwerks zur Verbesserung der Rückverfolgbarkeit der Halal-Fleischintegrität in der gesamten Lieferkette vor. Alle zuvor genannten Forschungsarbeiten sind die Idee der Verwendung eines zentralen Systems, das letztlich der einzig denkbare Weg war, um Informationstransparenz entlang der Lieferketten zu erreichen (Tian, 2017). Es gibt jedoch nicht genügend Beweise für die Richtigkeit und Vertrauenswürdigkeit der gemeinsamen Informationen im Rückverfolgbarkeitssystem und zwischen den Akteuren der Halal-Fleischlieferkette. Dies führt zu einem undurchsichtigen System, Informationsasymmetrie und vielen anderen Problemen. Das *RFID*-fähige Rückverfolgbarkeitssystem reicht nicht aus, um die Halal-Integrität von Fleisch zu gewährleisten. Das manuelle Abrufen und Speichern von Informationen in der zentralen Datenbank bringt viele Möglichkeiten der Irreführung und Verfälschung mit sich. In ähnlicher Weise ist es problematisch, sicherzustellen, dass die so genannten Halal-Fleischprodukte den islamischen Ernährungsvorschriften entsprechen und frei von irreführenden Herkunftsgeschichten sind.

3. Grundlagen

In diesem Kapitel sollen zuerst die Grundlagen zur *Chargenrückverfolgung* selbst dargelegt werden. Dies erfolgt über eine allgemeine Definition einer *Charge* einer anschließenden Einordnung von *Chargen* in die Wertschöpfungskette der Fleischwarenindustrie sowie den besonderen Dokumentationspflichten für *Chargen* in Deutschland. Darüber hinaus wird in Kapitel 3.2 die *Blockchain-Technologie* erörtert. Hier wird ebenfalls eine grundsätzliche Definition der Technologie gegeben, sowie eine Abgrenzung zwischen den verschiedenen Begrifflichkeiten vorgenommen. Außerdem wird im Detail auf die einzelnen technischen Konzepte und Komponenten eingegangen, aus denen sich eine *Blockchain* bildet. So soll ein grundlegendes Verständnis für die beiden Thematiken *Chargenrückverfolgung* und *Blockchain-Technologie* aufgebaut werden als Unterstützung für das darauf folgende Lösungskonzept und den Systementwurf.

3.1. Chargenrückverfolgung

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 und 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 *Chargenrückverfolgung* 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.

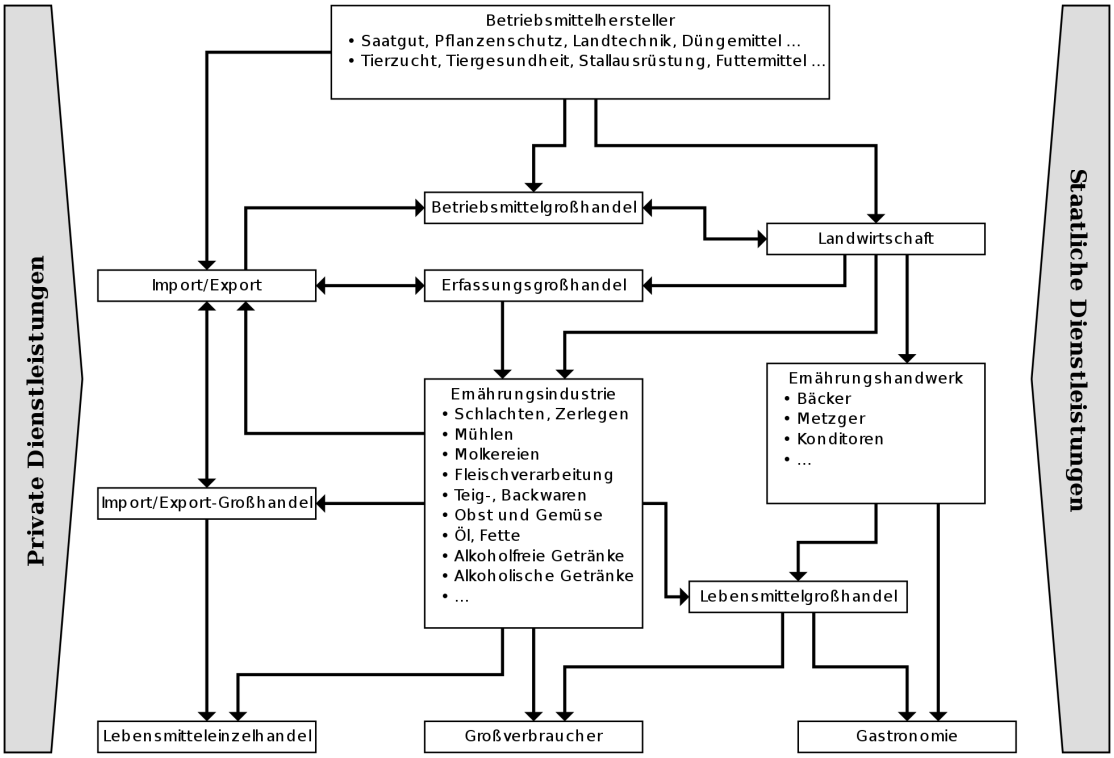

Abbildung 3: Wertschöpfungskette: Lebensmittelindustrie (Strecker, 2010)

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 und 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 und 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. 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
- 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.

[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)

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 und 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 und Stormer, 2018).

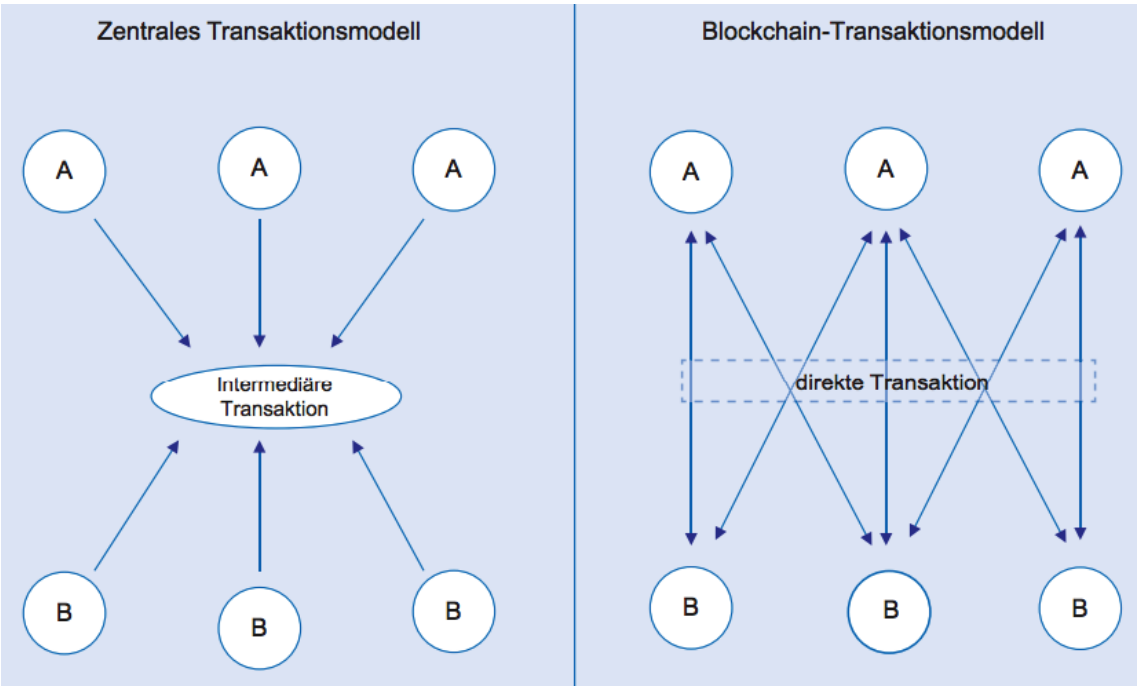

Abbildung 4: Transaktionsmodell *Blockchain* (Wald, 2017)

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.

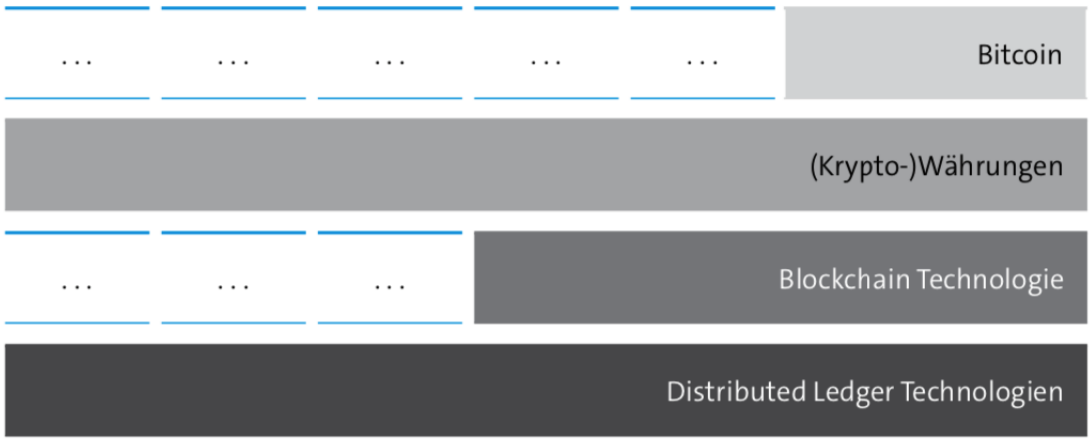

Abbildung 5: Schichtenmodell *Blockchain* Begriffe (Eigene Darstellung)

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 Technology (DLT) 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 eine eigene *Blockchain* 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 Geschwin-

digkeit 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 Skalierbar-

keit. 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 Eigenschaften 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 Teilnehmer auch als Validatoren auftreten (Permissionless) oder es wird eine Gruppe von Teilnehmern zum validieren von Transaktionen gebildet, die definierte Anfor-

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

derungen 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 und 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

3.2.5. Kryptographisches Hashing

Kryptographisches Hashing gehört zu einem der wichtigsten Instrumente der Kryptographie und bildet einen eigenen Teilbereich der Kryptographie. Mit einer *kryptographischen Hashfunktion* lässt sich aus einem beliebig langen Wort (oder Datensatz)

[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)

eine Zeichenkette mit fixer Stellenanzahl generieren. Die jeweilige Ausgabelänge wird in *Bit* angegeben. Formal ist eine *Hashfunktion* definiert als

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

Das Ergebnis wird als *digitaler Fingerabdruck* bezeichnet. Die Generierung des *Hashwerts* ist nicht zwingend kryptographisch, denn nicht jede *Hashfunktion* erfüllt alle Anforderungen einer *kryptographischen Hashfunktion* (Diffie, 1976; Menezes, 1997). Dabei gilt, eine *kryptographische Hashfunktion* 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 *Hashfunktion*, 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 *kryptographischen Hashfunktion*. Der Eingabewert, hier Urbild, wird durch die *kryptographische Hashfunktion* in einen Ausgabewert (*Hashwert*) fester Länge transformiert.

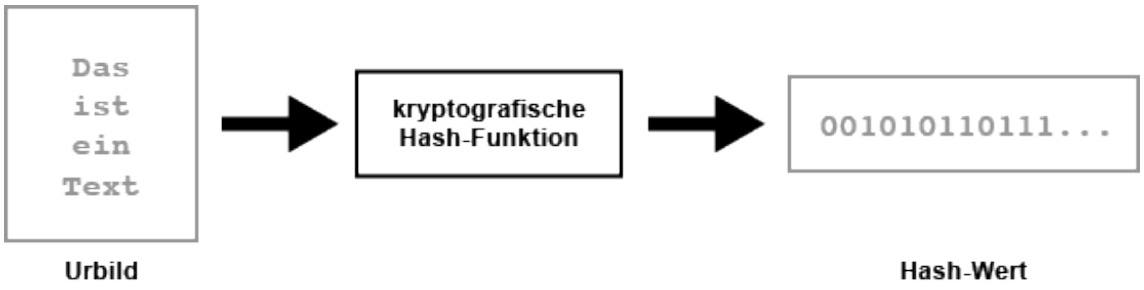

Abbildung 6: Funktionsweise einer *kryptographischen Hashfunktion* (Schärer, 2019)

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)

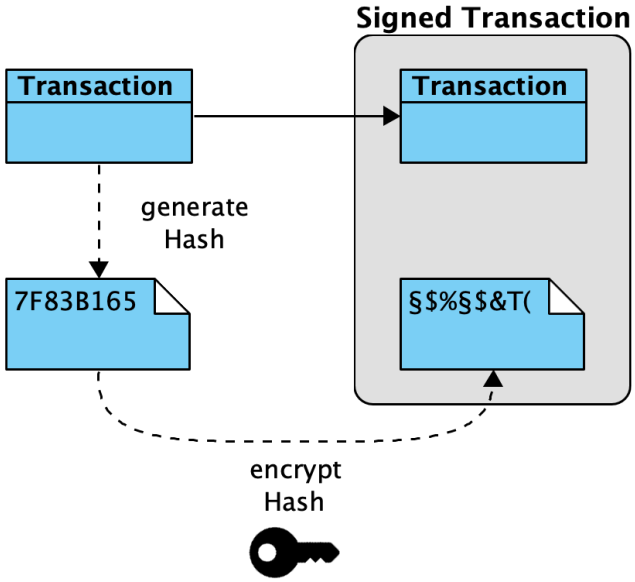

Abbildung 7: Schematische Darstellung für das Erstellen einer *digitalen Signatur* (in Anlehnung an Drescher (2017))

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 durch sie sichergestellt (Drescher, 2017).

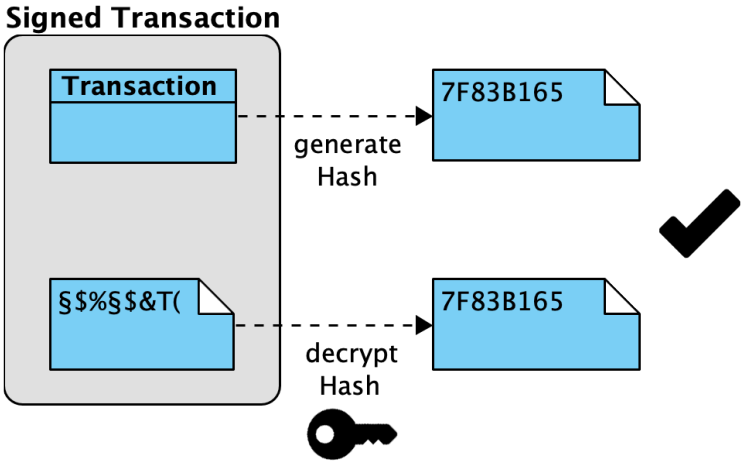

Abbildung 8: Erfolgreiche Prüfung einer *digitalen Signatur* (in Anlehnung an Drescher (2017))

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 *Hashwerte* 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.

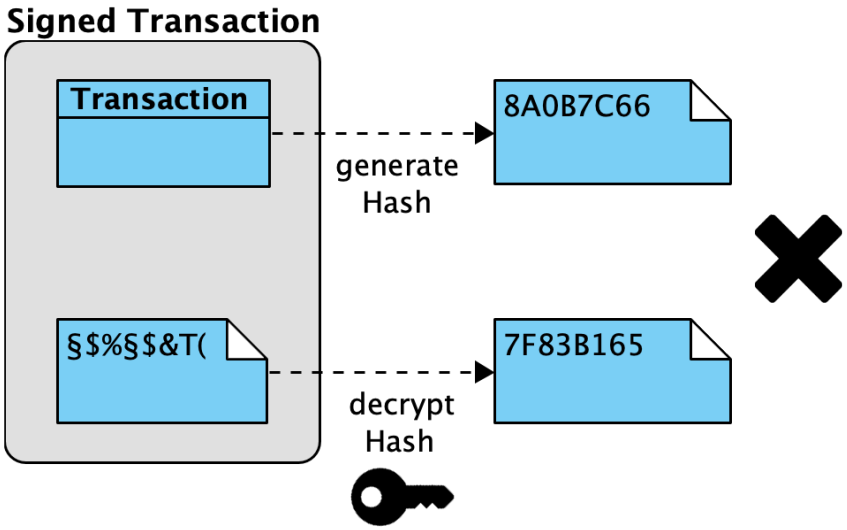

Abbildung 9: Erkennung von Manipulation anhand der *digitalen Signatur* (in Anlehnung an Drescher (2017))

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 gleichwertig, 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.

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

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 werden durch eine Erweiterung des Modells. Ein Austausch kann durchgeführt werden, wenn der Primärknoten die Anfrage nicht innerhalb eines bestimmten Zeitlimits an die Backupknoten weiterleitet (Castro et al., 1999).

[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*.

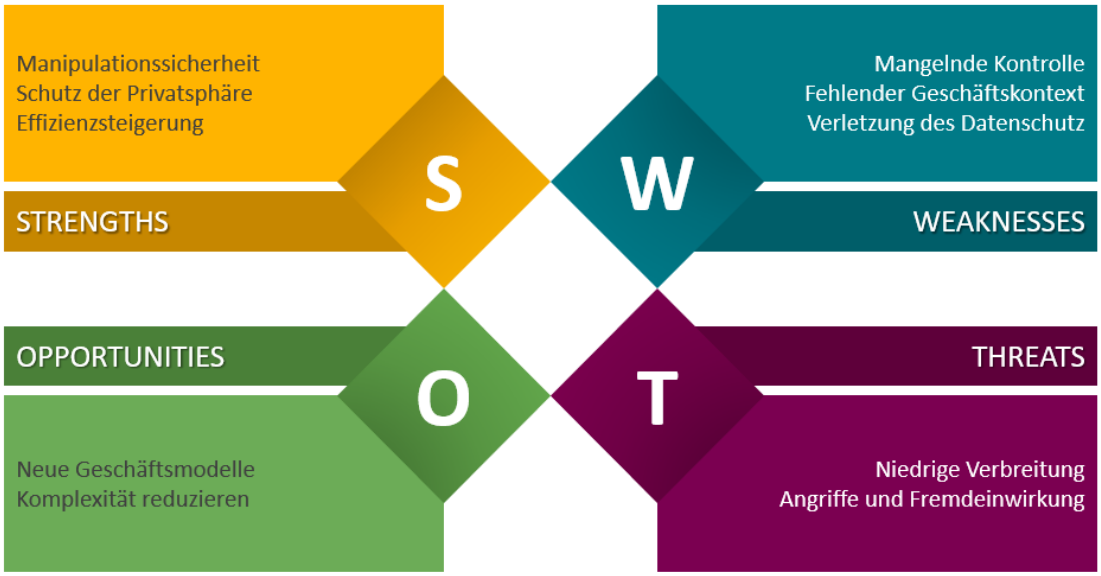

Abbildung 10: *Blockchain-Technologie SWOT-Analyse* (eigene Darstellung)

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 man 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?

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*. Ausgewählt wurden die Kandidaten nach der sog. *Grounded Theory* Methode (Strübing, 2002). Nachfolgend soll eine kurze Beschreibung dazu dienen alle Kandidaten im Kontext der *Blockchain-Technologie* vorzustellen.

Ethereum [11] 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

[11] Buterin u a. (2013)

gewissen Schutz vor Angriffen aus dem Netzwerk selbst. *Ethereum* verwendet zur Programmierung von *Smart Contracts* die Sprache Solidity.

Hyperledger Fabric [12] 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 [13] wurde entwickelt für eine sichere Kommunikation und Zahlungen im *Machine-to-Machine* Bereich und dem Internet of Things (IoT). 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 (Ferraro et al., 2018). 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

[12] Valenta und Sandner (2017)
[13] Popov (2018)

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.

Quorum [14] 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.

[14] Chase (2016)

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 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. Jedoch sind nicht alle Ausprägungen der *Blockchain-Technologie* 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 (Schiener, 2017).

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

Beginnend mit einer Erläuterung der Anforderungserhebung soll in diesem Kapitel der Systementwurf dokumentiert werden. Dazu ist eine Zieldefinition gegeben anhand welcher das spätere System und seine Funktionalitäten skizziert werden. Neben der reinen Zieldefinition dient eine Beschreibung der Wertschöpfungskette im Allgemeinen und eine Betrachtung des Waren- und Datenstrom im genaueren dazu den Kontext für den Systementwurf herzustellen. Anschließend wird der eigentliche Geschäftsprozess der Chargenrückverfolgung im Ist- und Soll-Zustand dargestellt, um die Unterschiede bei einem Einsatz einer *Blockchain* Lösung herauszuarbeiten. Der Systementwurf mit einer Erklärung aller Einzelkomponenten schließt dieses Kapitel ab.

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 und 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 hier-

archisch strukturiert und um eine Quelle ergänzt nach Koelsch (2016). Dies soll die Nachverfolgbarkeit der Anforderungen während der Evaluation unterstützen.

5.2. Das Ziel: Chargenrückverfolgung

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-Tags* und entsprechende Lesegeräte, welche per Schnittstelle mit dem *Blockchain* Netzwerk verbunden sind, abwickeln lassen (Dorri et al., 2017; Samaniego und 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.

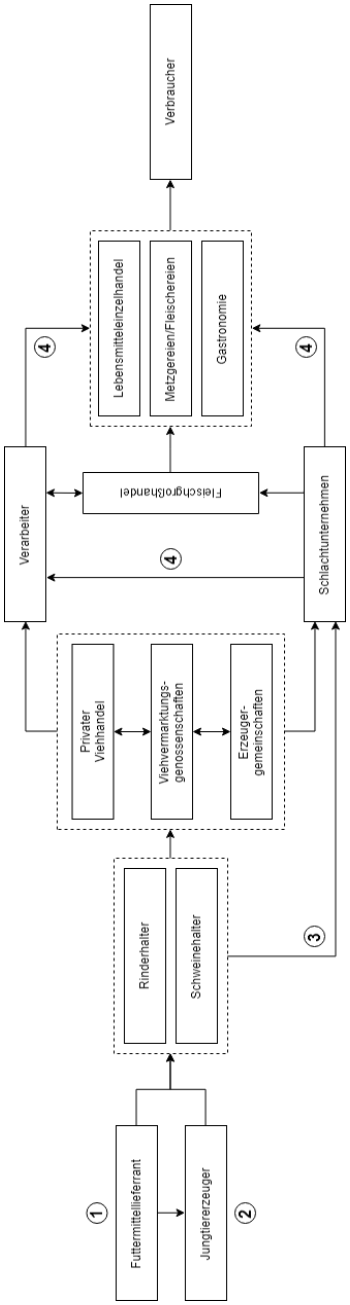

Abbildung 11: Struktur der Wertschöpfungskette der Fleischwirtschaft nach Beck (2008); Petersen et al. (2010); Voss et al. (2010)

Der Warenstrom beginnt mit (1) der Futtermittellieferung an die Jungtiererzeuger und Viehhalter. Jeder Betrieb wird dabei über die Internationale Lokationsnum-

mer (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. (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 Vergabe 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.

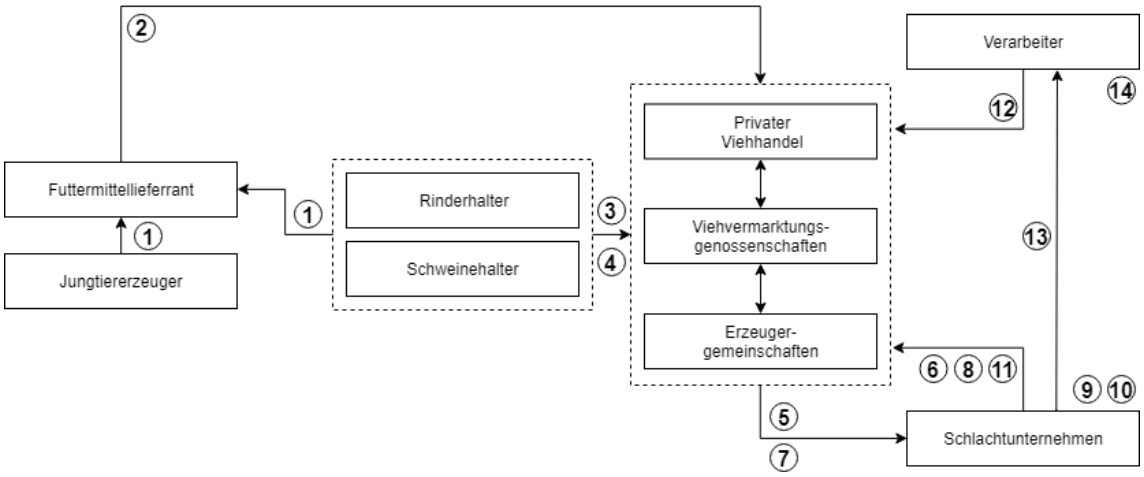

Abbildung 12: Datenströme innerhalb der Wertschöpfungskette nach Beck (2008); Petersen et al. (2010); Voss et al. (2010)

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 durchgeführt 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 von einer zentralen Instanz bereitsgestellt, 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.

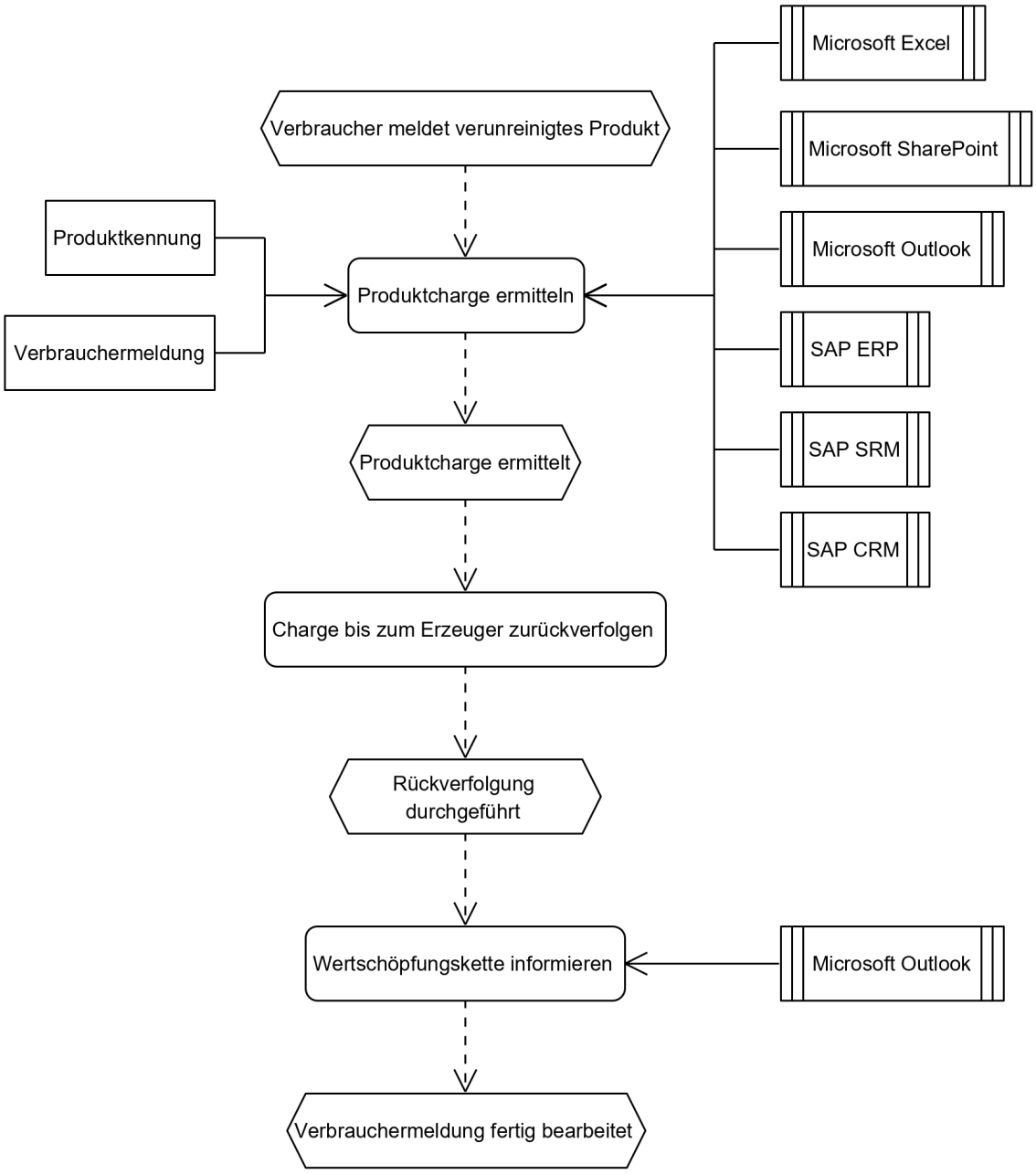

Abbildung 13: Ist-Geschäftsprozess Chargenrückverfolgung in eEPK Notation

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 Ap-

plikationen 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.

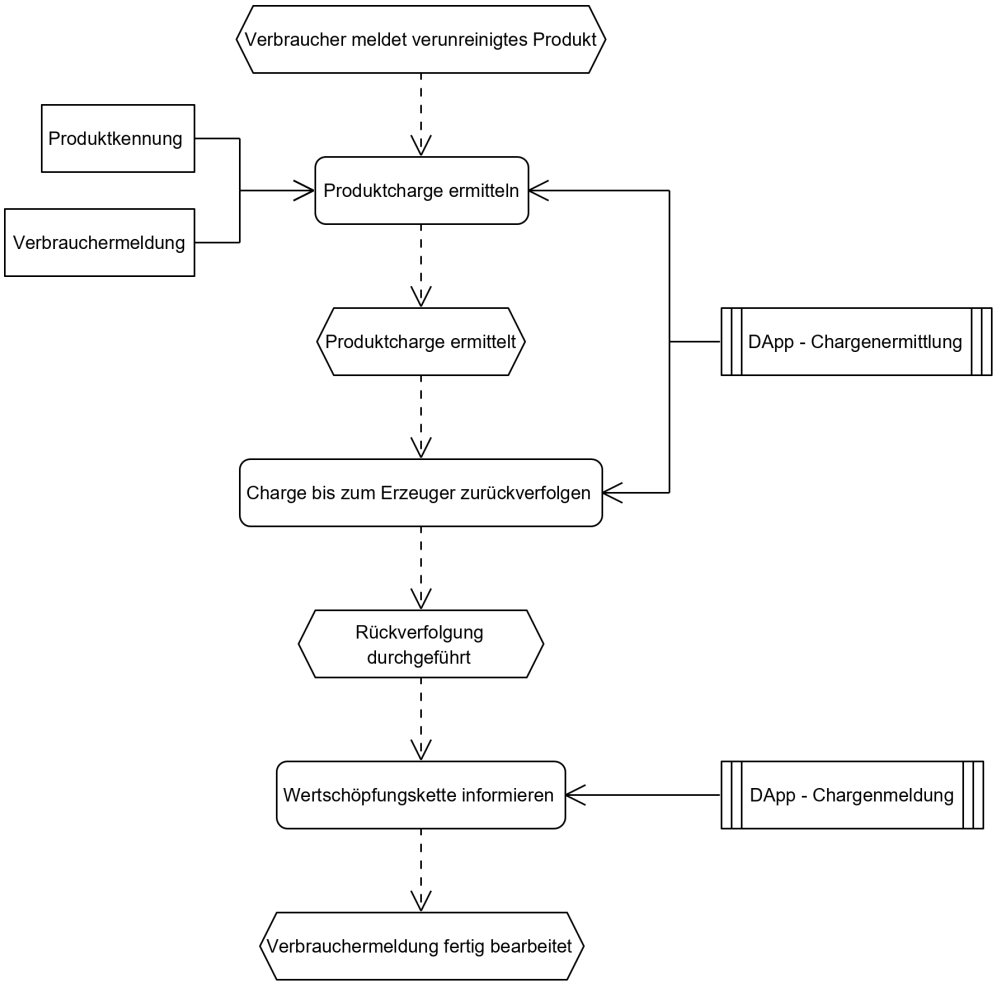

Abbildung 14: *Soll*-Geschäftsprozess Chargenrückverfolgung in eEPK Notation

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. Außerdem wird ein einzelner Knoten der Gesamtarchitektur im Detail beschrieben.

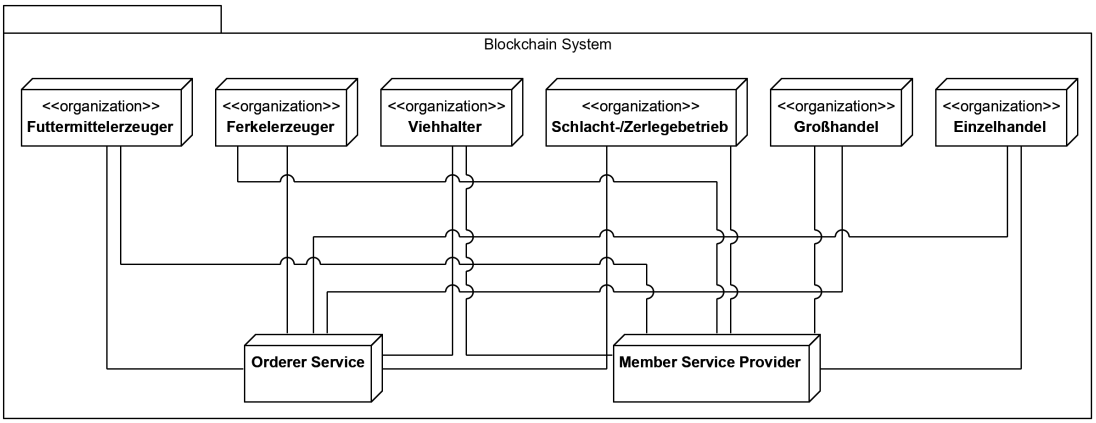

Abbildung 15: *Blockchain* System Architektur

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.

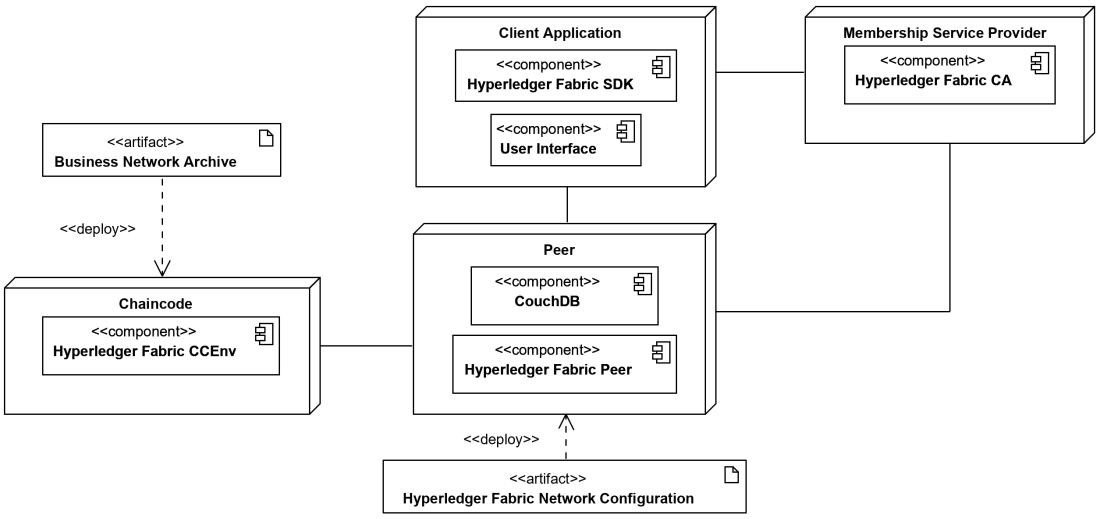

Abbildung 16: Organisation Komponenten Diagramm

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.

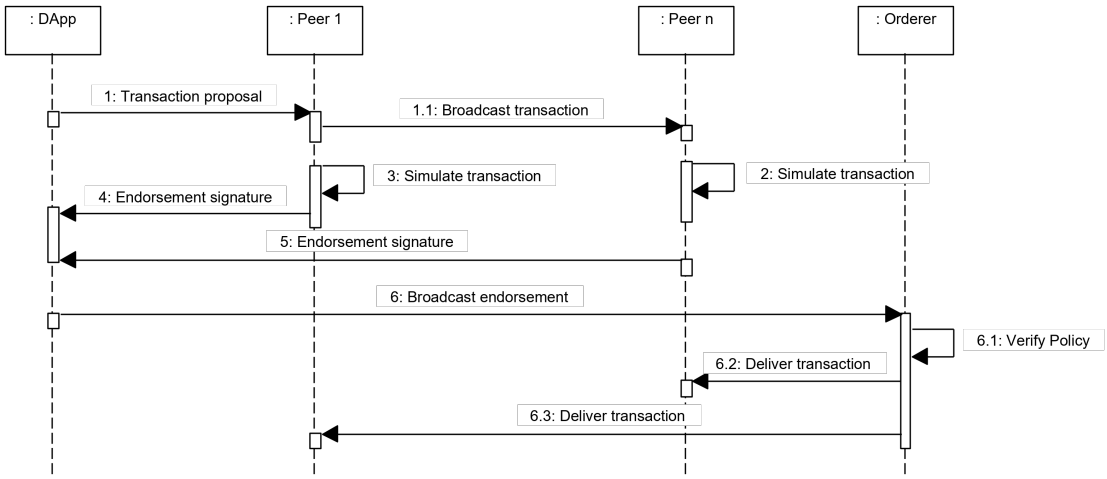

Abbildung 17: *Transaction Flow* in Anlehnung an (Choudhury et al., 2018)

5.5.2. Smart Contracts

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*

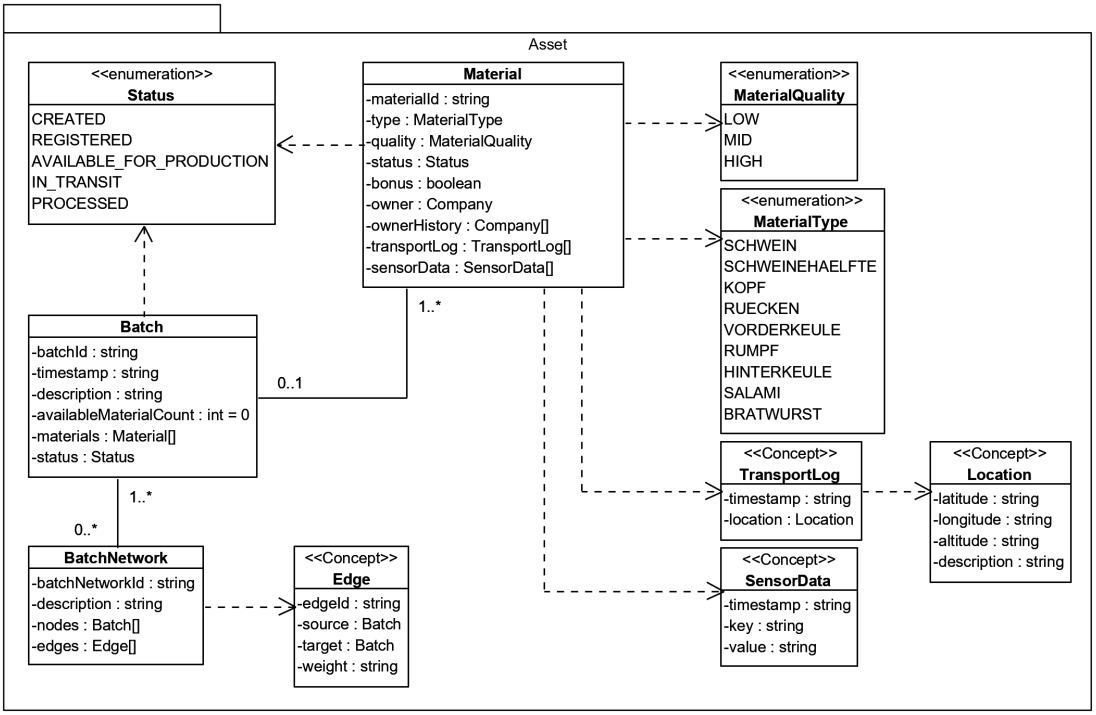

Abbildung 18: Klassendiagramm *Blockchain* Netzwerk *Assets*

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)[15]. 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. Anforderung *A2.2* wird mit dieser

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

Datenstruktur erfüllt. In Abbildung 19 wird das beschriebene Konstrukt als Klassendiagramm dargestellt.

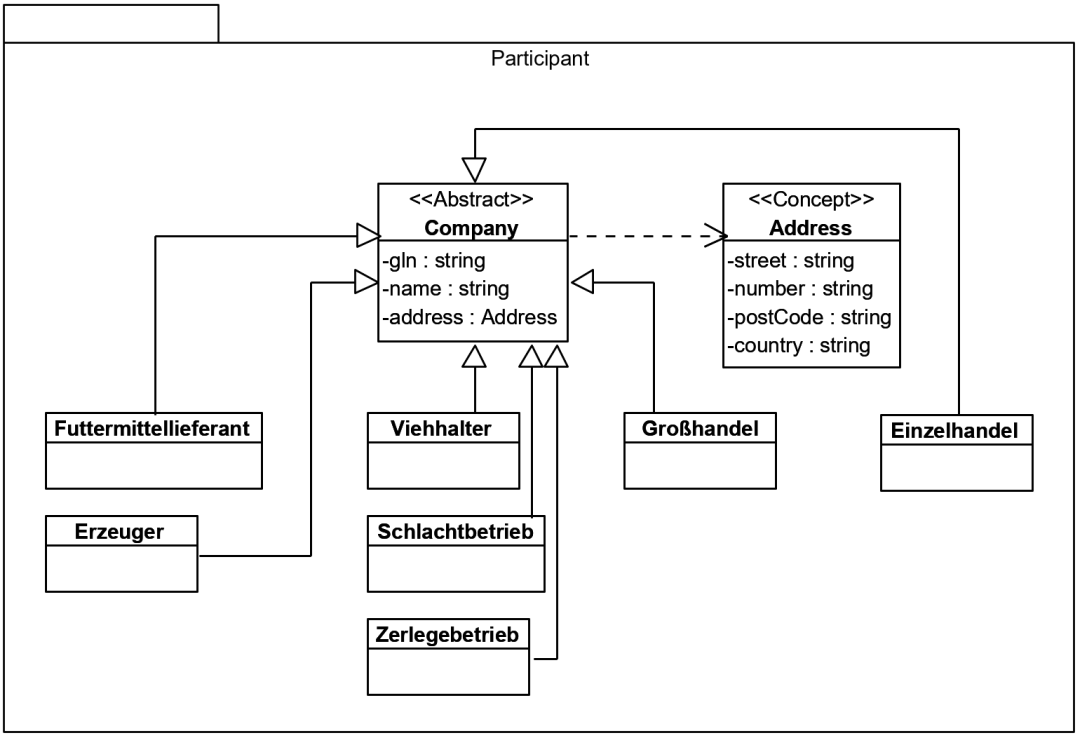

Abbildung 19: Klassendiagramm *Blockchain* Netzwerk *Participants*

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 Ei-

gentumsverhältnisse eines *Assets* verändert werden können. Außerdem lassen sich *Chargen* anlegen und mit registrierten Tieren verknüpfen.

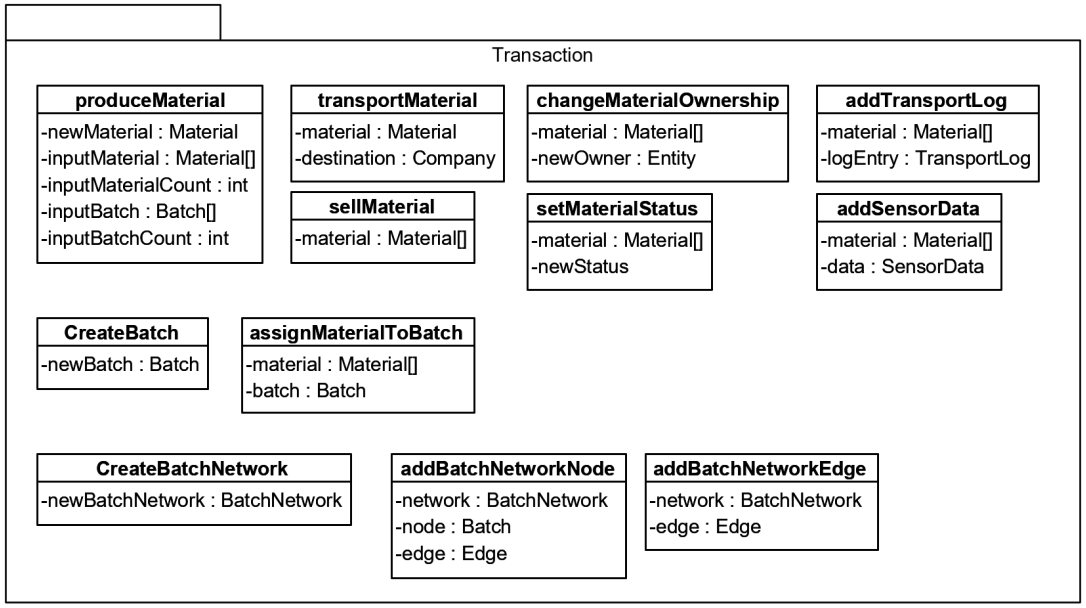

Abbildung 20: Klassendiagramm *Blockchain* Netzwerk *Transactions*

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 *Member Ship Provider (MSP)* 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.

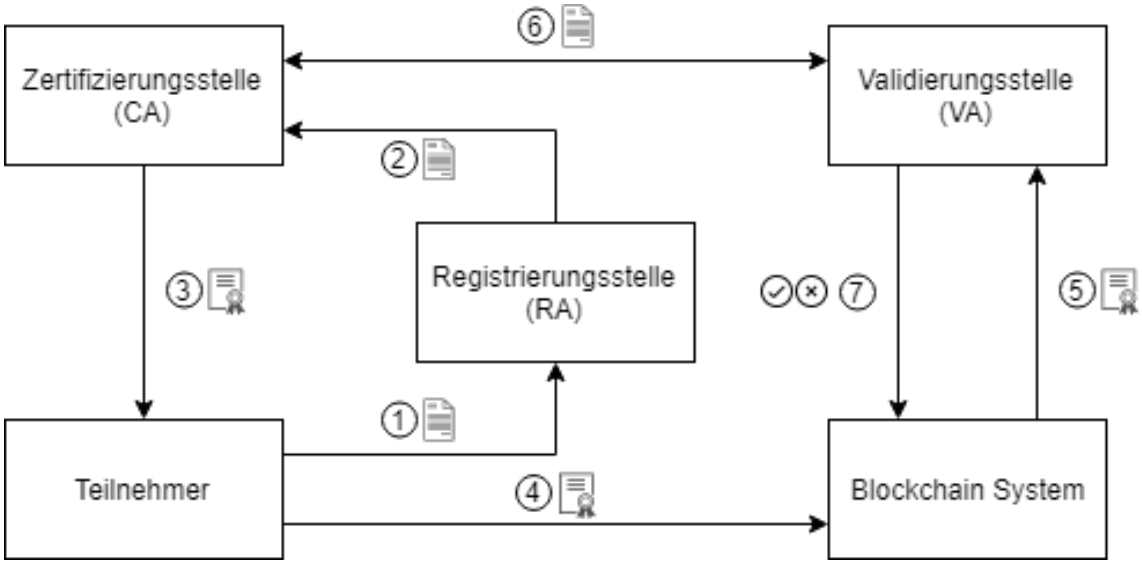

Abbildung 21: Ausstellen einer digitalen Identität für einen Teilnehmer der Blockchain

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*).

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.

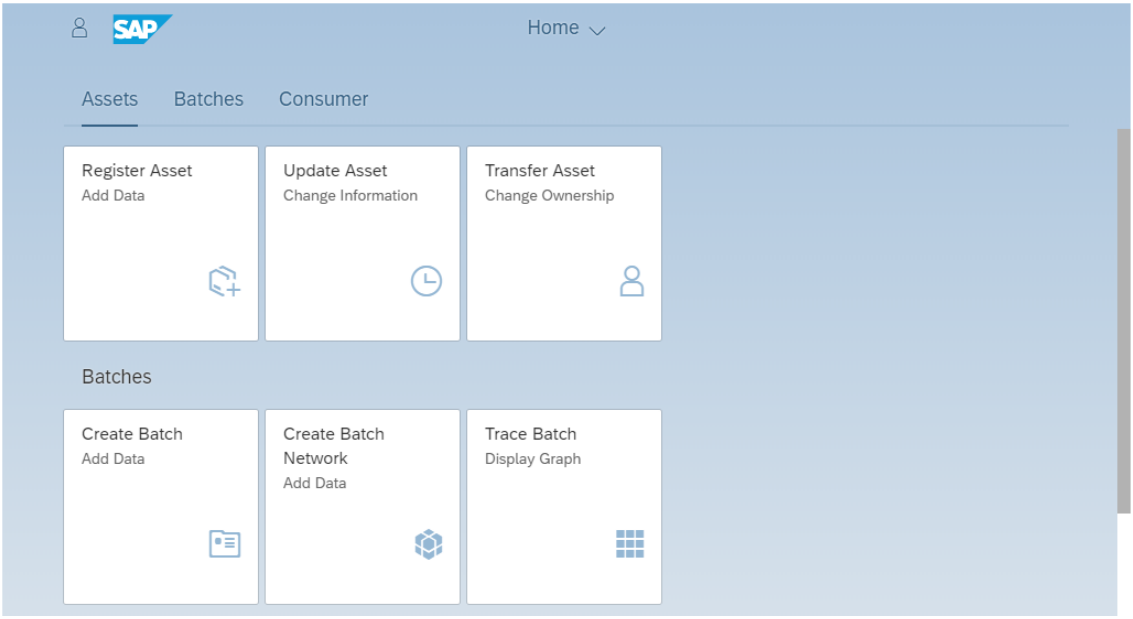

Abbildung 22: Mockup: Einstiegsseite Endanwender

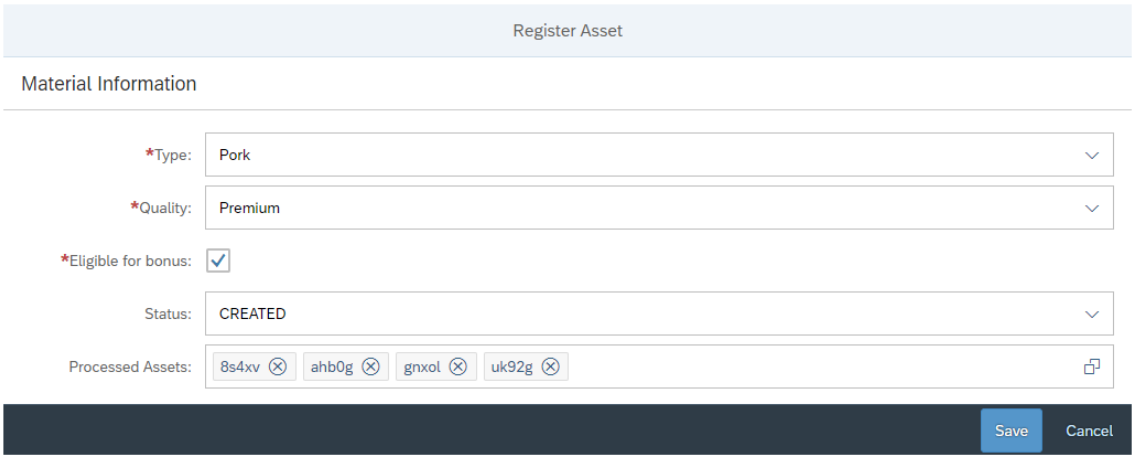

Abbildung 23: Mockup: Asset Registrierung

Des weiteren wurde eine Applikation modelliert mit der bereits erfasste *Assets* gepflegt werden können. Wie in Abbildung 24 durch die ausgegrauten Eingabe-

felder dargestellt, können bei einem vorhandenen *Asset* nicht alle Informationen nachträglich verändert werden. Schlüsselmerkmale wie die Identifikationsnummer werden beim Erfassen eines *Assets* automatisch vom System generiert und können daher nicht durch den Anwender angepasst werden.

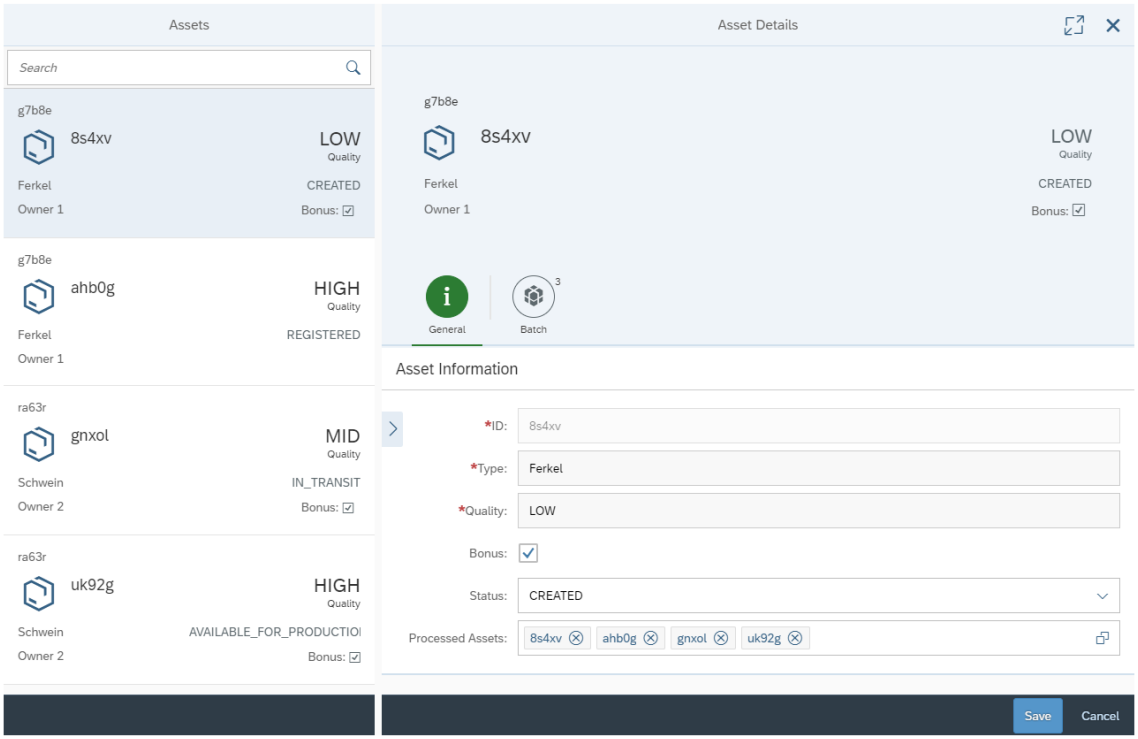

Abbildung 24: Mockup: Asset Update

Da während der Produktionskette ein *Asset* mehrfach zwischen den Teilnehmern ausgetauscht werden kann, muss der Anwender eine Möglichkeit haben diesen Eigentumswechsel erfassen zu können. Diese Möglichkeit wird in Abbildung 25 dargestellt. Der Anwender sieht in der linken Liste alle *Assets*, welche aktuell im Besitz der Organisation sind. Nach der Auswahl eines *Assets* hat der Anwender die Möglichkeit in der rechten Detailansicht einen neuen Eigentümer auszuwählen. Alle Teilnehmer des Netzwerk werden in der Dropdown-Liste aufgeführt. Speichert der Anwender das *Asset*, erfolgt im Hintergrund die Ausführung der Transaktion im *Blockchain* Netzwerk. Die Benutzungsoberfläche reagiert auf das Ergebnis der Transaktion mit

einer Aktualisierung der *Asset* Liste, sodass wieder nur die *Assets* angezeigt werden die auch tatsächlich im Besitz der Organisation sind.

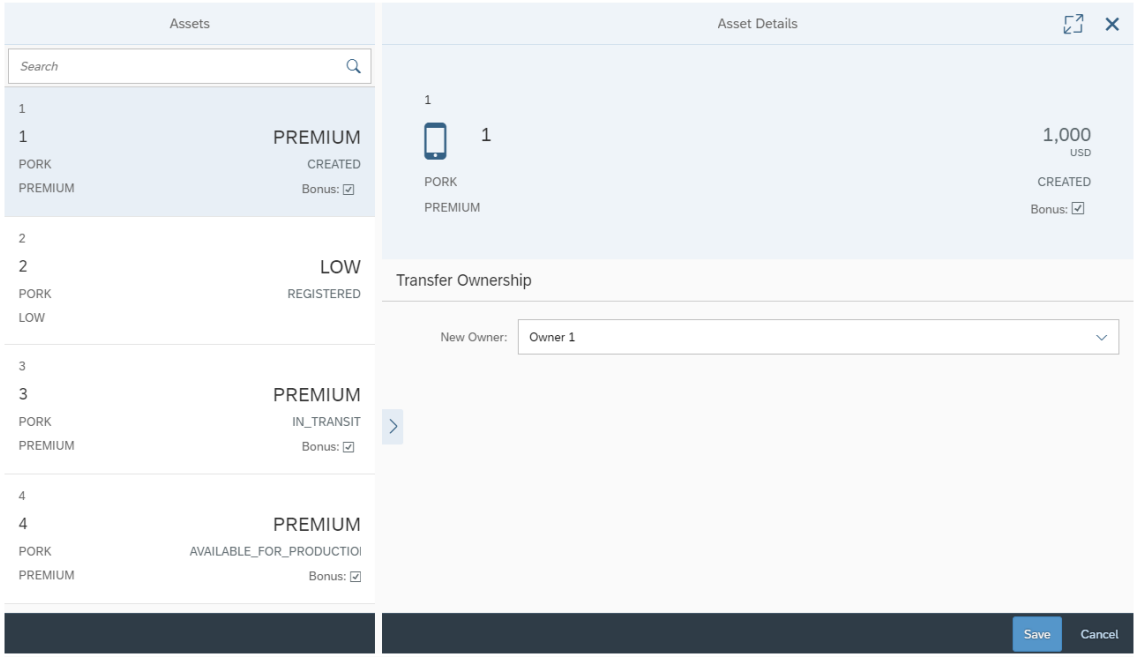

Abbildung 25: Mockup: Asset Transfer

Äquivalent zu den *Assets* wurden ebenfalls Applikationen zum Anlegen und Pflegen von *Chargen* modelliert. Wie in Kapitel 5 modelliert, wird eine *Charge* eindeutig über eine Identifikationsnummer identifiziert. Zusätzlich muss vom Anwender ein Zeitstempel erfasst werden, um eine *Charge* nachträglich einfacher finden zu können. Der Zeitstempel wird standardmäßig mit dem heutigen Datum vorbelegt, da beim Praxispartner mit Tageschargen gearbeitet wird. Damit im Nachgang ermittelt werden kann, welche *Assets* bzw. *Chargen* in die neue *Charge* eingeflossen sind lassen sich über zwei Eingabefelder im Netzwerk registrierte *Chargen* und *Assets* auswählen.

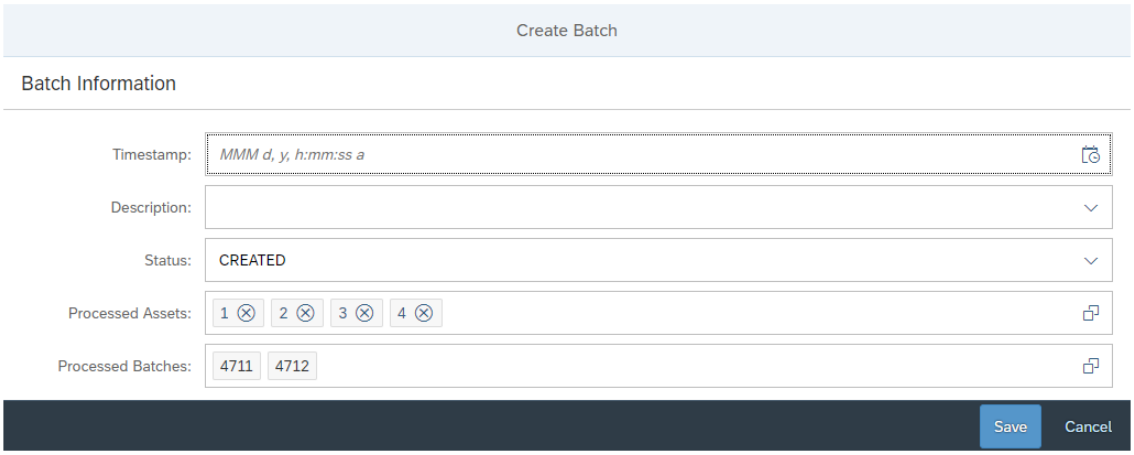

Abbildung 26: Mockup: Batch Create

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. Dabei hat sich gezeigt, dass durch den Einsatz einer *Blockchain* eine Vielzahl unterschiedlicher Software und Datensenken innerhalb des Prozesses der Chargenrückverfolgung durch die *Blockchain* abgelöst werden können. 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[16] 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 und 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 Prototypen 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*.

[16] 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[17] 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 B.2). Verbindungsprofil und die digitale Identität

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

des 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.

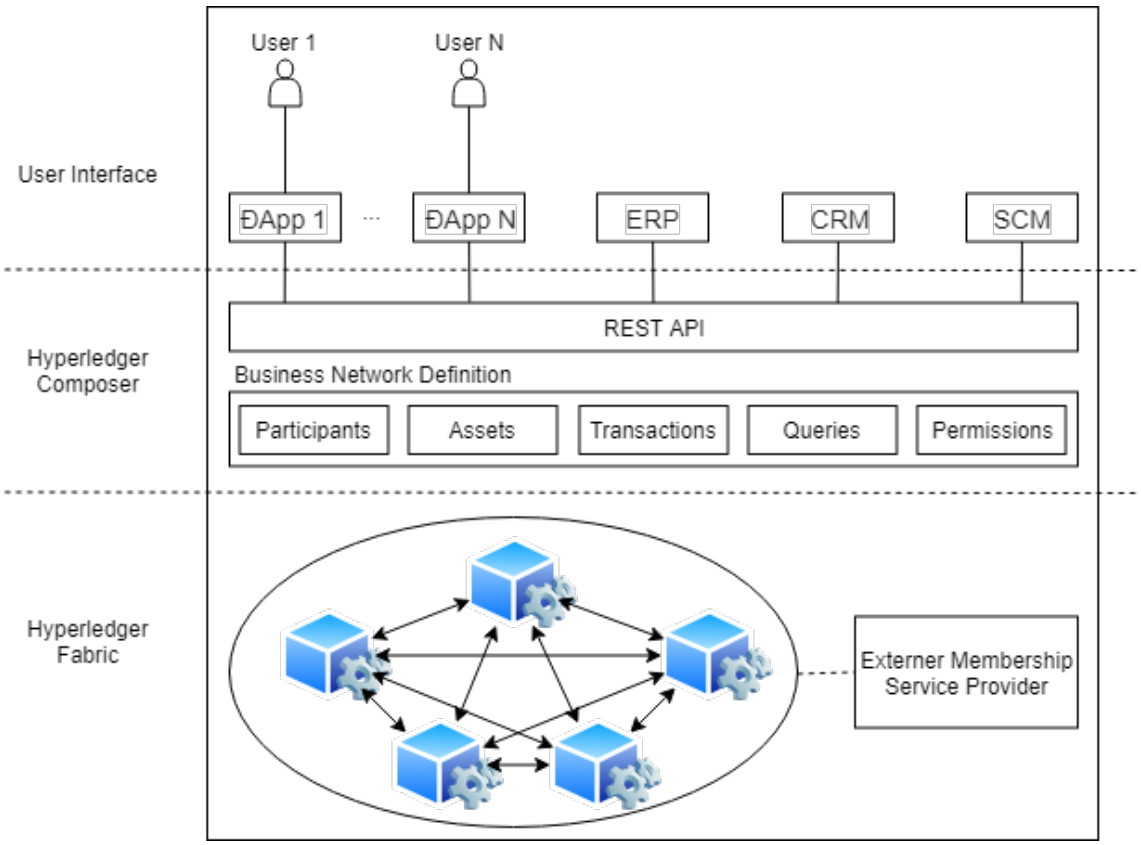

Abbildung 27: Gesamtsystem Prototyp (Eigene Darstellung)

Nachdem starten der Docker Container lässt sich ein einfacher *Smoke Test*[18] 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

[18]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* mit einem *Event* 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 }
```

```
19
20 event notification {
21     --> Material changedMaterial
22 }
```

Listing 1: Model Example Definition

Zeile 1 in Listing 1 definiert einen Namensraum für das gesamte Model. In einem produktiven Szenario 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* darstellen 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 B.3. Eine *Transaction* mit zugehörigem *Event* ist in Zeile 15 bis 22 dargestellt. Die *Transaction* definiert dabei zwei Parameter als Ressourcenverknüpfung. Es werden das Asset *material* sowie der neue Eigentümer *newOwner* benötigt. Das *Event* definiert nur einen Parameter und zwar eine Ressourcenverknüpfung zum angepassten Asset *changedMaterial*. Wird das Event emittiert kann der Empfänger über die Ressource alle Informationen des Vorgangs nachvollziehen. In angebundenen User Interface (UI) Applikationen kann dann auf das Event entsprechend reagiert werden bzw. können Drittsysteme beispielsweise Workflowprozesse auslösen.

```
async function changeMaterialOwnership(tx) {
    const oMaterial = tx.material;
    const oNewOwner = tx.newOwner;
    const oActualOwner = tx.material.owner;

    const oMaterialReg = await getAssetRegistry(NS + '.Material');
    const bMaterialExists = await oMaterialReg.exists(oMaterial.getIdentifier());
    if(!bMaterialExists) {
        throw new Error('Input material does not exist.');
    }
    if (oMaterial.owner !== getCurrentParticipant()) {
        throw new Error('You are not allowed to change asset.');
    }
    const oParticipantRegistry = await getParticipantRegistry(oNewOwner.
    getNamespace());
    const bNewOwnerExists = await oParticipantRegistry.exists(oNewOwner.
    getIdentifier());
    if(!bNewOwnerExists) {
        throw new Error('New owner does not exist.');
    }

    oMaterial.ownerHistory.push(oActualOwner);
    oMaterial.owner = oNewOwner;
    await oMaterialRegistry.update(oMaterial);

    const oNotification = getFactory().newEvent('io.dev.foodchain', 'notification
    ');
    oNotification.changedMaterial = oMaterial.getIdentifier();
    emit(oNotification);
}
```

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

Damit ein *Participant* Funktionen auf einem *Asset* ausführen kann wurden im vorherigen Abschnitt *Transactions* modelliert. Zu jeder *Transaction* Definition im Modell gehört eine Logikimplementierung. Verknüpft wird die Modelldefinition mit der Implementierung über die Annotation *@transaction*. Eine *Transaction* Funktion hat als einzigen Parameter das *Transaction* Objekt. Über dieses Objekt kann innerhalb der Funktion auf alle Werte der Transaktion zurückgegriffen werden. Für das Beispiel

des Eigentumswechsel wurde eine Transaktion definiert, die zum einen das *Material* beinhaltet und zum anderen eine Referenz auf den neuen Eigentümer (siehe Listing 1 Zeile 16 f.). Der Aufbau einer *Transaction* Funktion folgt stets dem Muster - Initialiseren der Eingabeparameter, Plausibilitätsprüfungen, Geschäftslogik und abschließend die optionale *Event* Emittierung. Das Initialisieren der Eingabewerte wurde von Zeile 8 bis Zeile 10 implementiert. Es werden alle benötigten Werte der Transaktion zu lokalen Variablen zugewiesen. Zeile 14 bis Zeile 28 deckt die Plausibilitätsprüfung ab, hier wird geprüft ob im Falle des Eigentumswechsels

- das *Material* im Netzwerk vorhanden ist,
- der Transaktionsemittent auch Besitzer des *Materials* ist und
- ob der neue Eigentümer als *Participant* im Netzwerk vorliegt.

Die eigentliche Geschäftslogik ist relativ simpel und von Zeile 31 bis 33 implementiert. Für eine spätere Rückverfolgung der Eigentumsverhältnisse wird der aktuelle Eigentümer zur Eigentümerhistorie (Eigenschaft *ownerHistory*) hinzugefügt und der neue Eigentümer wird gesetzt. Danach müssen die Assetänderungen noch an das Systemregister übermittelt werden. Das Schlüsselwort *await* wird verwendet, da es sich hier um einen asynchronen Aufruf handelt und in der Logik so eine Haltemarke gesetzt wird sodass auf das Ergebnis des Aufrufs gewartet wird bevor mit der weiteren Verarbeitung der Funktion fortgefahren wird. Sollten bis zu diesem Zeitpunkt keine Fehler in der Verarbeitung aufgetreten sein, wird ein *Event* erzeugt, mit Daten gefüllt und emittiert. Einfache Berechtigungsprüfungen wie in der Transaktionslogik lassen sich auch über die Zugriffskontrolle regeln. Hyperledger unterscheidet zwischen der Zugriffskontrolle für Ressourcen innerhalb des Netzwerks und der Zugriffskontrolle für Änderungen seitens der Netzwerkadministration. Um den Zugriff auf eine Ressource zu steuern wird eine Regel definiert wie in Listing 3. Diese Regel sagt aus, dass sie für jeden *Participant* und bei jeder Operation (Lesen, Anlegen, Ändern, Löschen) angewandt wird (Zeile 3/4). Sie gilt für alle Ressourcen aus dem Namensraum *io.dev.foodchain.** und als Bedingung wurde definiert, dass der Besitzer *r.owner.getIdentifier()* der Ressource gleich dem aktuellen *Participant* ist (Zeile 5/6). Ist diese Regel erfüllt wird die Operation erlaubt bzw. bei nicht erfüllen der Zugriff auf die Ressource verweigert. Es ist anzumerken, dass die Regeln in der

Reihenfolge ausgewertet werden in der sie definiert sind und die erste Regel deren Bedingung erfüllt ist, bestimmt ob der Zugang gewährt oder verweigert wird. Sofern keine Regel angewandt werden kann wird der Zugriff standardmäßig verweigert.

```
rule OwnerHasFullAccessToTheirAssets {
    description: "Allow all participants full access to their assets"
    participant(p): "io.dev.foodchain.*"
    operation: ALL
    resource(r): "io.dev.foodchain.*"
    condition: (r.owner.getIdentifier() === p.getIdentifier())
    action: ALLOW
}
```

Listing 3: Berechtigungsdefinition

Das letzte Element des *Business Network Archive (BNA)* ist die Abfragedefinition. Hier können für die Verwendung innerhalb der Transaktionslogik oder direkter Anfragen über externe Anwendungen Abfragen formuliert werden die eine ähnliche Syntax verwenden wie in der Structured Query Language (SQL). Listing 4 zeigt eine einfache Abfrage, um alle *Assets* vom Typ *Material* zu selektieren für die gilt, das die boolesche Eigenschaft *bonus* den Wert *wahr* hat und das die Eigenschaft *type* gleich dem Parameter *_$type* ist. Parameter die innerhalb einer Anfrage definiert werden über den Präfix *_$*, müssen beim Aufruf der Abfrage mit übergeben werden. Über den eindeutigen Namen *selectBonusMaterials* lässt sich diese Abfrage direkt in der Transaktionslogik ausführen.

```
query selectBonusMaterials {
    statement:
        SELECT io.dev.foodchain.Material
            WHERE (bonus == true) AND (type == _$type)
            ORDER BY [status ASC]
}
```

Listing 4: Abfragedefinition

In der *statement* Eigenschaft einer *Query* können jeweils folgende Operatoren verwendet werden:

- *SELECT* ist ein obligatorischer Operator und definiert standardmäßig das Register und den Asset- oder Teilnehmertyp, der zurückgegeben werden soll.

- *FROM* ist ein optionaler Operator, der ein anderes Register für die Abfrage festlegt.
- *WHERE* ist ein optionaler Operator, der die Bedingung definiert, die auf die selektierten anzuwenden sind.
- *AND* ist ein optionaler Operator, der zusätzliche Bedingungen definiert.
- *OR* ist ein optionaler Operator, der alternative Bedingungen definiert.
- *CONTAINS* ist ein optionaler Operator, der Bedingungen für Array-Werte definiert.
- *ORDER BY* ist ein optionaler Operator, der die Sortierung der Ergebnisse definiert.

Damit ist das *Business Network Archive (BNA)* vollständig und kann in einem *Hyperledger Fabric* Netzwerk installiert und instantiiert werden. Ein Netzwerkadministrator kann anschließend *Participants* erzeugen und mit der digitalen Identität verknüpfen. Der Netzwerkteilnehmer ist daraufhin in der Lage Transaktionen im Netzwerk abzusetzen, um mit dem *Smart Contract* zu interagieren und Geschäftsvorgänge entsprechend abzubilden.

6.3. Schnittstelle

Damit die Funktionalität des *Blockchain* Netzwerks in bestehende IT Landschaften integriert werden kann bietet *Hyperledger* die Möglichkeit einen *Smart Contract* in Form einer *REST*[19] *API* für externe Anwendungen freizugeben.

Mit dem Ansatz einer *REST Application Programming Interface (API)* wird die Idee einer programmiersprachen unabhängigen Schnittstelle realisiert. Nahezu jede moderne Programmiersprache ist in der Lage simple *HTTP* Anfragen zu formulieren, über das Internet abzusetzen und das Resultat auszuwerten. Dadurch kann eine breite Masse an UI Technologien und Frameworks verwendet werden, um für

[19] *Representational State Transfer (REST)* steht für ein Programmierparadigma für verteilte Systeme. Dabei wird der Zustand einer Ressource nicht gesondert gespeichert (Session) sondern über den *Uniform Resource Identifier (URI)* codiert.

den Endanwender entsprechende Applikationen zur Nutzung der *Smart Contract* Funktionalität zu implementieren. Ebenfalls sind externe Systeme wie beispielsweise ERP-Systeme über die *REST API* in der Lage mit dem *Blockchain* Netzwerk zu kommunizieren und *Asset* Operationen auszuführen.

Die *REST* Schnittstelle wird dabei nach der *OpenAPI*[20] Spezifikation generiert und zur Verfügung gestellt. Bei der Generierung wird aus den Modelldefinitionen und der Transaktionslogik die Klassen- und Funktionsdokumentation extrahiert und in der Schnittstellendokumention dargestellt.

Wird die *REST* Schnittstellen über den *Hyperledger Composer REST Server* ausgeliefert erhält man eine Übersicht aller *Assets, Participants, Transactions & Queries* die über den *Smart Contract* abgebildet worden sind. Abbildung 28 zeigt einen Ausschnitt der Oberfläche. Darauf sind drei *API Endpunkte* abgebildet namentlich *generateMockTransactionData*, *Manufacturer* und *Material*. Die ersten beiden Endpunkte bilden eine *Transaction* und einen *Participant* aus dem *Smart Contract* ab. Der dritte Endpunkt zeigt die Dokumentation einer *HTTP GET* Operation für ein *Material* mit Beispielwerten eines Resultats. Darunter sind alle weiteren möglichen *HTTP* Operationen inklusive des codierten *URI*.

[20] OpenAPI (ursprünglich Swagger) bietet eine Spezifikation und ein Framework zum Beschreiben, Erzeugen, Konsumieren und Visualisieren von REST Schnittstellen (OpenAPI Initative, 2018; Purushothaman, 2015).

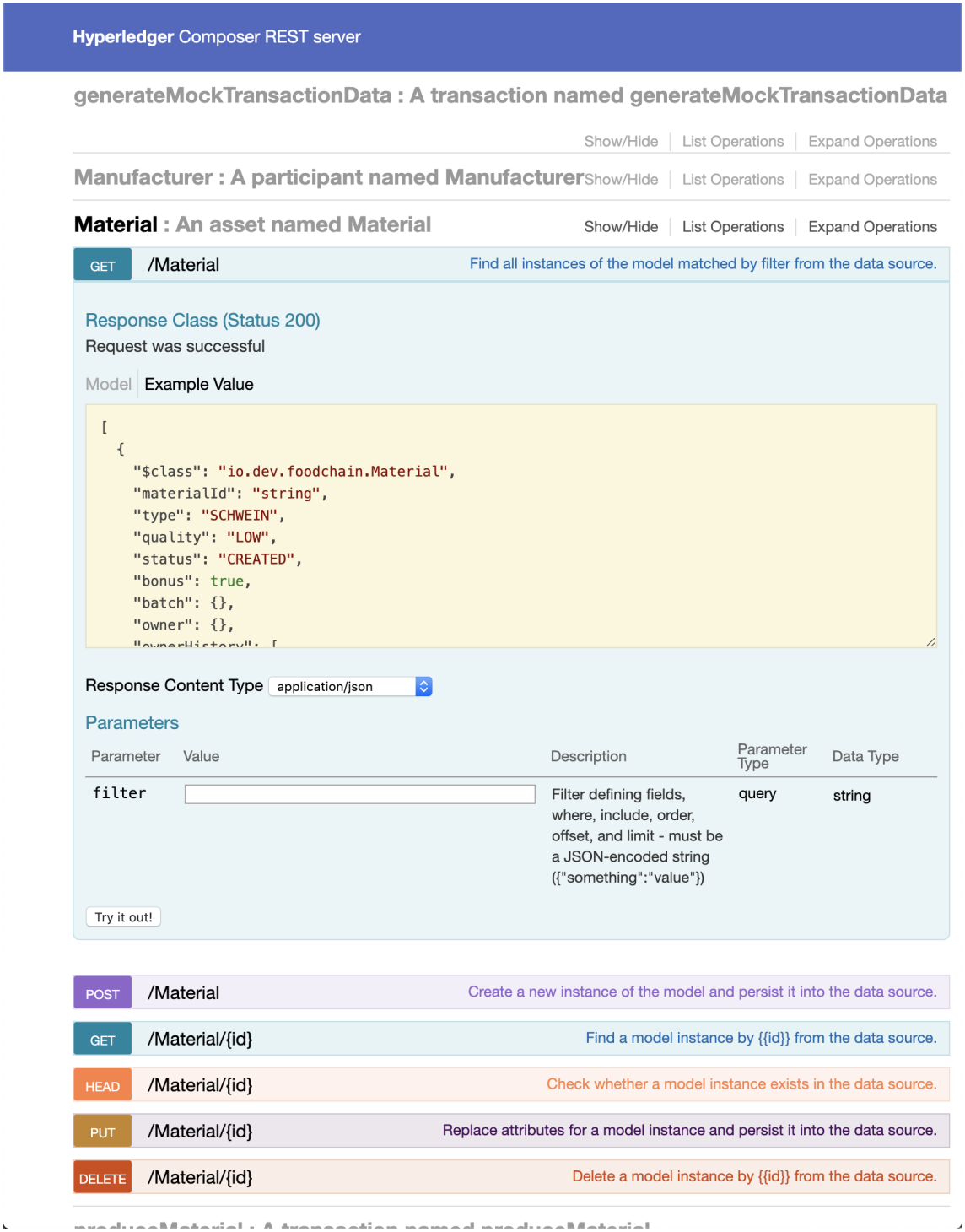

Abbildung 28: Weboberfläche der REST API

6.4. Zusammenfassung technische Umsetzung

In diesem Kapitel wurde gezeigt wie die Basis einer *Blockchain* Lösung mit *Hyperledger Fabric* und *Composer* realisiert wird. Dazu wurde, wie zuvor im Systementwurf konzeptioniert, ein *Peer* Netzwerk bestehend aus zwei Organisationen mit jeweils zwei *Peer* Knoten aufgespannt. Ebenfalls wurden zwei Zertifikatstellen für beide Organisationen in die Systemarchitektur mit aufgenommen. Darauf aufbauend ist die Geschäftslogik mit *Hyperledger Composer* implementiert worden. Dazu gehört die Definition von *Assets, Participants, Transactions* und *Events*. Zu jeder *Transaction* wird eine *Transaction Processor Function* in JavaScript implementiert. Aus diesen Komponenten wurde das *Business Network Archive* gebaut, welches im zuvor aufgespannten *Hyperledger Fabric Blockchain* Netzwerk deployt wurde. Eine Kapselung des *Smart Contracts* in eine *REST* Schnittstelle dient dazu externen Anwendungen, wie beispielsweise Endanwender Applikationen, die Funktionalität des *Smart Contracts* bereitzustellen. Dabei wurde die *REST API* nach der *OpenAPI* Spezifikation modelliert.

7. Evaluation

Gemäß der beschriebenen Vorgehensweise aus Kapitel 1.3 wurden das modellierte Lösungskonzept im Allgemeinen sowie der implementierte Prototyp hinsichtlich der Realitätsnähe, Übertragbarkeit und Innovationsgehalt durch ein Experteninterview evaluiert. Als Interviewpartner wurde der stellvertrende IT-Leiter des Westfleisch Konzern herangezogen. Die Auswahl erfolgte auf Grund der langjährigen Berufserfahrung innerhalb der Fleischwarenindustrie, sowie dem tiefen Prozesswissen auch für angrenzende Produktionsschritte. Ebenfalls ist der Interviewpartner im Forschungsprojekt Resource-Efficent, Economic and Intelligent Foodchain (REIF)[21] vertreten und kann hierdurch Kompetenzen im bereich der neuen Technologien wie künstliche Intelligenz (KI), *Blockchain* sowie *Internet of Things (IoT)* vorweisen. Das Interview fand in den Büroräumen der Westfleisch SCE mbH in Münster statt. Wie Ritchie et al. (2013) schreiben, dient dies dazu dem Interviewpartner ein möglichst komfortables und ruhiges Umfeld zu bieten.

Das Interview verlief nach dem im Anhang C beschriebenen Interviewguide. Dabei wurde zunächst eine kurze Vorstellung der Position des Befragten im Konzern gegeben und anschließend anhand einer Präsentation die Ergebnisse diskutiert. Zum besseren Verständnis wurde noch eine kurze Demo des entwickelten Prototypen durchgeführt, damit der Befragte ein klares Bild vom umgesetzten Lösungskonzept und Systementwurf bekommt. Eine Transkription des vollständigen Interviews befindet sich in Anhang D. Die im folgenden genannten Zeilenangaben beziehen sich auf das Transkript.

Das Oberthema *Blockchain* war dem Befragten nicht fremd, da er über das Forschungsprojekt REIF ebenfalls mit den Problemen der Rückverfolgung und manipulationssicherer Transaktionsverarbeitung in Berührung gekommen ist. Dies zeigen die folgenden Aussagen:

> *„Thema Blockchain (..) ist ja in aller Munde zur Zeit. Damit haben wir auch Berührungspunkte im Forschungsprojekt, da wir dort auch versuchen die Wertschöpfungskette der Lebensmittelbranche zu optimieren.“ (Z. 41f)*

[21] Projektwebsite `https://ki-reif.de`

Der gewählte Ansatz das *Blockchain* Netzwerk mit der *Hyperledger Fabric* Software umzusetzen auf Grund der inherenten Eigenschaften und einiger Vorteile, die im Lösungskonzept (Kapitel 4) beschrieben wurden, konnten vom Interviewpartner bestätigt werden.

> *„Macht Sinn, Hashwerte sagen mir noch was aus meinem Studium ((lachen)) auch wenn das schon etwas länger her ist.“ (Z. 83f)*
> *„Also hast du dich für dieses Hyperledger entschieden auf Grund der Geschwindigkeit und dem Fokus der Software auf den industriellen Sektor?“ (Z. 106f)*

Im Interview wurde nochmal deutlich, wie wichtig ein gemeinsames Netzwerk zur Chargenrückverfolgung eigentlich ist. Der Befragte beschreibt die Schwierigkeiten der verschiedenen Dateiformate bei der Integration von Zulieferen und Kunden, welche beim Einsatz eines gemeinsamen *Blockchain* Netzwerks wegfallen würden.

> *„Also können wir über die Smart Contracts unsere Geschäftslogik abbilden bzw. auch unsere Zulieferer und Endkunden. Wenn man bedenkt das wir so knapp 130 Kunden haben und jeder Kunde uns ein anderes Format für ihre Chargeninformationen vorgibt bzw. nutzt, dann würde so ein System schon wirklich Sinn machen allein aus Gründen der Standardisierung.“ (Z. 134ff)*

Der Befragte war positiv überrascht über die gewählte Benutzeroberfläche, welche mit dem *SAP UI5 Framework* im SAP Fiori Design modelliert wurde. Hier konnten vom Befragten noch Anmerkungen für eine zukünftige weiterentwicklung des Prototypen entgegen genommen werden.

> *„Also ich hab gesehen das du die Oberflächen mit Fiori modelliert hast. [...] Da könnte man sich sicherlich nochmal mit den Fachabteilungen hinsetzen und gucken das man da einen Feinschliff reinbekommt. Ich mein, für einen Prototyp ist das aus meiner Sicht völlig ausreichend, aber wenn man sowas dann auf einer Messe präsentieren möchte vielleicht im Zusammenspiel mit einem KI Systen ((lachen)) dann muss sowas ja heutzutage alles sehr gut aussehen.“ (Z. 161ff)*

Bezüglich des Potentials des Prototypen hat der Befragte erwähnt, dass Lösungen bzw. Systeme die über die Unternehmensgrenze hinweg funktionieren sollen ein gewisser Anreiz für die Teilnehmer geschaffen werden muss damit sie überhaupt an so einem System teilnehmen.

> *„Der entscheidende Punkt ist aus meiner Sicht ist oft die Marktdurchdringung. Du kannst noch so tolle Systeme und Technologien entwickeln, wenn niemand am Markt oder in der Branche dieses System nutzt, aus welchen Gründen auch immer, dann wird dieses System keinen Erfolg haben. Deshalb sollte man im Blick behalten, das mit so einem System eine Art „Win-Win“ Situation hergestellt wird. Wenn ich als Teilnehmer des Netzwerk etwas hineingebe muss ich auch immer etwas herausbekommen, sonst sinkt mein Interesse dieses System zu verwenden.“ (Z. 182ff)*

Als mögliche weitere Ausbaustufen des Prototyps wären beispielsweise die Integration von Veterinärinformation zu den untersuchten Tieren sowie Daten zu den verwendeten Futtermitteln genannt.

> *„Natürlich, wie gesagt generell könnte man so ein System für sämtliche Tierarten erweitern, die wir so durch die Produktionswerke schieben. Obendrauf wäre es ziemlich interessant Auswertungen der Veterinäre mit zu erfassen. Endkunden wollen wissen wieviel Antibiotika in ihrer Wurst steckt. Grade bei Hühnerfleisch, da legen die Käufer sehr viel Wert drauf mittlerweile. Außerdem wird immer öfter nicht nur auf die Art und Weise der Haltung geschaut, sondern auch was die Tiere während ihres Lebens als Futter bekommen haben.“ (Z. 197ff)*

Abschließend wurde vom Befragten noch hinzugefügt, dass ein solches *Blockchain* Netzwerk vom Ansatz her als eine Vorstufe zur gesamten Optimierung der Wertschöpfungskette angesehen werden kann. Diese Optimierung wird aktuell durch den Befragten im Forschungsprojekt REIF erarbeitet.

> *„Demnach hast du mit deiner Arbeit ein ganzen Stück an Vorarbeit für das Forschungsprojekt REIF geleistet und bewiesen das eine Rückverfolgbarkeit mit dieser Technologie vom Landwirt bis zum Endkunden machbar ist.“ (Z. 220ff)*

8. Abschlussbetrachtung

8.1. Zusammenfassung

In dieser Masterarbeit wurde analysiert, wie sich eine Chargenrückverfolgung mittels der *Blockchain-Technologie* realisieren lässt. Dazu wurden die in Abschnitt 1.2 gestellten Forschungsfragen anhand der *Design Science Methode* nach Hevner (2007) bearbeitet. Die einzelnen Teilfragen wurden in den Kapiteln 4 und 5 mit einer der Fragestellung passenden Methodik näher betrachtet. Die Forschungsfrage *FF1.1* sowie *FF1.2* sind über die Grundlagenkapitel abgedeckt worden. In diesen Kapiteln wurde detailliert beschrieben, welche Anforderungen und Daten zur Realisierung einer Chargenrückverfolgung in der Fleischwarenindustrie von nöten sind. Neben einer ausführlichen Beschreibung der Wertschöpfungskette im fleischverarbeitenden Gewerbe wurde die *Blockchain-Technologie* und ihre Ausprägungen behandelt. Forschungsfrage *FF1.3* wurde mittels einer SWOT-Analyse mit anschließender Nutzwertanalyse entgegen getreten. Aus den Ergebnissen der Analyse wurde dann im Kapitel 5 ein entsprechendes System Design abgeleitet, welches für den Anwendungsfall passend ist und die in *FF1.1* ermittelten Anforderungen erfüllt. Nach dem Systementwurf folgte die prototypische Implementierung des zuvor modellierten Systems auf Basis der *Hyperledger Fabric Blockchain* in Kombination mit dem *Hyperledger Composer* Framework zur *Smart Contract* generierung. Evaluiert wurde der Prototyp anhand eines Experteninterviews. Die Befragung einer Person mit direktem Bezug zu den behandelten Geschäftsprozessen sowie der nötigen Kompetenz bezüglich neuartiger Technologien wie *Blockchain* und IoT stellt eine für diese Arbeit ausreichend gesicherte Evaluation der Ergebnisse aus Systementwurf und dem resultierenden Prototyp dar.

8.2. Reflexion

Während der Anforderungsanalyse hat sich gezeigt, das die *Blockchain-Technologie* in den Fachabteilungen des Praxispartners zwar bekannt war, ihre möglichen Einsatzwecke jedoch noch vollkommen unklar sind. *Blockchain* wurde stets mit der Kryptowährung Bitcoin assoziiert. So war es schwierig die Anforderung entsprechend spezifisch und nicht zu allgemein zu erheben ohne das wichtige Aspekte des

zu modellierenden System außer acht gelassen werden. Auf Grund der Komplexität im realen Umfeld der Chargenrückverfolgung wurde der aufgenommen Prozess sowie das zu Grunde liegende Datenmodell soweit vereinfacht, das die Funktionalität der Chargenrückverfolgung weiterhin auf die Realität im Unternehmen abgestimmt war. Allerdings konnten eine Vielzahl an Sonderfällen, die grade bei der Verarbeitung von Schweinen auftreten, nicht beachtet werden. Die Vertragssituation zwischen Landwirten und den verarbeitenden Betrieben basiert oft auf mündlichen Absprachen bzw. sind hierfür großzügige Toleranzen in den Verträgen erfasst um nachträgliche Anpassungen beispielsweise bei den Preisen für bestimmte Tiere möglich zu machen. Da der Fokus dieser Arbeit auf der generellen Machbarkeit einer Chargenrückverfolgung mittels der *Blockchain-Technologie* lag, wurden diese Freiheitsgrade nicht weiter betrachtet bzw. in der prototypischen Implementierung beachtet. Solch eine komplexe Wertschöpfungskette wie sie in der Fleischwarenindustrie vorliegt könnte nur schwer die im wissenschaftlichen Kontext dieser Arbeit intendierte notwendige Übertragbarkeit und Reproduzierbarkeit gewährleisten.

8.3. Ausblick

Aus wissenschaftlicher Perspektive bietet sich die naheliegendste Fortsetzung dieser Arbeit sicherlich in der Implementierung des vorgestellten Systementwurfs in einem konkreten betrieblichen Umfeld an. Der in dieser Arbeit entwickelte Prototyp kann hierbei als Grundlage zur Erforschung weiterer Prozesse die über eine *Blockchain* abgebildet werden herangezogen werden. Dabei könnte der gezeigte Systementwurf mit entsprechendem Aufwand für weitere Tierarten, Veterinärinformationen oder Futtermitteldaten erweitert werden. Ebenfalls wäre es denkbar eine vorhandene *Internet of Things (IoT)* Lösung zu integrieren, um Sensordaten aus den Betriebsstätten bzw. während des Transports direkt in die *Blockchain* einfließen zu lassen. Hierdurch könnte der Informationsgehalt für Aussagen zu einer *Charge* oder dem gesamten Lebenszyklus eines einzelnen Tieres vom Landwirt bis zum Endkunden noch einmal deutlich erhöht werden. Außerdem bietet sich eine Integration der Blockchaindatenbasis mit vorhandenen ERP-Systemen an. ERP-Systeme halten eine große Menge an Stamm- und Bewegungsdaten aus dem betrieblichen Kontext vor. Lassen sich diese

Daten mit den Transaktionsdaten des *Blockchain* Netzwerks verknüpfen eröffnen sich weitere Anwendungsgebiete beispielsweise im Bereich von Business Intelligence.

Literatur

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

Andersen, D., Balakrishnan, H., Kaashoek, F., und 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.

Bahrudin, S. S. M., Illyas, M. I., und Desa, M. I. (2011). Tracking and tracing technology for halal product integrity over the supply chain. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE.

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

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

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

Bundesregierung (1993). Los-Kennzeichnungs-Verordnung.

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

Buterin, V. u a. (2013). Ethereum white paper. *GitHub repository*, Seiten 22–23.

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.

Castro, M., Liskov, B., u a. (1999). Practical Byzantine fault tolerance. In *OSDI*, Band 99, Seiten 173–186.

Chase, J. M. (2016). Quorum white paper. Technischer bericht, Accessed 2018-02-15.[Online]. Available: https://github. com/jpmorganchase ....

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

Dick, J., Hull, E., und 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. und 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., und Wang, J. (2017). Untangling Blockchain: A Data Processing View of Blockchain Systems. *CoRR*, abs/1708.05665.

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

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

Efken, J., Deblitz, C., Kreins, P., Krug, O., Kueest, S., Peter, G., und 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.

Ferraro, P., King, C., und Shorten, R. (2018). IOTA-based Directed Acyclic Graphs without Orphans. *arXiv preprint arXiv:1901.07302*.

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. und 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., und 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.

Junaini, S. N. und Abdullah, J. (2008). MyMobiHalal 2.0: Malaysian mobile halal product verification using camera phone barcode scanning and MMS. In *2008 International Conference on Computer and Communication Engineering*. IEEE.

Kassim, M., Yahaya, C. K. H. C. K., Zaharuddin, M. H. M., und Bakar, Z. A. (2012). A prototype of Halal product recognition system. In *2012 International Conference on Computer & Information Science (ICCIS)*. IEEE.

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

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

McEntire, J. und Kennedy, A. W. (2019). *Food Traceability : From Binders to Blockchain*. Practical Approaches. 1st ed. 2019.. Auflage.

Meier, A. und Stormer, H. (2018). Blockchain = Distributed Ledger + Consensus. *HMD Praxis der Wirtschaftsinformatik*, 55(6):1139–1154.

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

Mohamad, M. A., Mansor, S., Ahmad, N., Adnan, W. A. W., und Wali, I. M. (2016). THE RELIABILITY OF HALAL PRODUCT TRANSPORTATION USING GPS TRACKING SYSTEM. *Journal of Theoretical & Applied Information Technology*, 90(2).

Mohammed, A., Wang, Q., und Li, X. (2016). A study in integrity of an RFID-monitoring HMSC. *International Journal of Food Properties*, 20(5):1145–1158.

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.

OpenAPI Initative (2018). OpenAPI Specification 3.0.2.

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., und Kuechler, B., Herausgeber (2012). *Design Science Research in Information Systems. Advances in Theory and Practice*. Springer Berlin Heidelberg.

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

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

Pohl, K. V. und Pohl, K. (2015). Basiswissen Requirements Engineering : Aus- und Weiterbildung zum Certified Professional for Requirements EngineeringFF-oundation Level nach IREB-Standard.

Popov, S. (2018). The Tangle - Iota.

Purushothaman, J. (2015). *RESTful Java Web Services*. Packt Publishing.

Ritchie, J., Lewis, J., Nicholls, C. M., Ormston, R., u a. (2013). *Qualitative research practice: A guide for social science students and researchers*. sage.

Samaniego, M. und 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)*, Seiten 433–436. IEEE.

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

Schiener, D. (2017). Current Role of The Coordinator. https://domschiener.gitbooks.io/iota-guide/content/chapter1/current-role-of-the-coordinator.html. abgerufen am 09.10.2019.

Schärer, T. (2019). Kryptografische Hash-Funktion.

Siepermann, C., Vahrenkamp, R., Siepermann, M., und 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. Auflage.

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

Strecker, O. (2010). *Marketing für Lebensmittel und Agrarprodukte*. DLG-Verlag.

Strübing, J. (2002). Just do it? *KZfSS Kölner Zeitschrift für Soziologie und Sozialpsychologie*, 54(2):318–342.

Tan, M. I. I., Razali, R. N., und Husny, Z. J. (2012). The adoption of halal transportations technologies for halal logistics service providers in Malaysia. In *Proceedings of World Academy of Science, Engineering and Technology*, Nummer 63. World Academy of Science, Engineering and Technology.

Tian, F. (2017). A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In *2017 International Conference on Service Systems and Service Management*. IEEE.

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., und Bouayad, H. (2018). Supply Chain Management based on Blockchain: A Systematic Mapping Study. *MATEC Web of Conferences*, 200:00020.

Trienekens, J. und 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. und Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123.

Valenta, M. und Sandner, P. (2017). Comparison of ethereum, hyperledger fabric and corda. *[ebook] Frankfurt School, Blockchain Center*.

Visser, C. und Hanich, Q. A. (2017). How blockchain is strengthening tuna traceability to combat illegal fishing.

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

Wald, P. (2017). *Blockchain. Das disruptive Potential im Finanzsektor*. GRIN Verlag.

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

Yergeau, F., Sperberg-McQueen, M., Maler, E., Paoli, J., und 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., und 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.

A. Anforderungen

A.1. 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.	

A.2. 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*

A.3. 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*

B. Listings

B.1. Hyperledger Fabric Peer Dockerfile

```
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}
```

B.2. Hyperledger Fabric *Network Connection Profile*

```
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
```

```
                    }
                }
            }
        },
        "organizations": {
            "Org1": {
                "mspid": "Org1MSP",
                "peers": [
                    "peer0.org1.example.com"
                ],
                "certificateAuthorities": [
                    "ca.org1.example.com"
                ]
            }
        },
        "orderers": {
            "orderer.example.com": {
                "url": "grpc://orderer.example.com:7050"
            }
        },
        "peers": {
            "peer0.org1.example.com": {
                "url": "grpc://peer0.org1.example.com:7051"
            }
        },
        "certificateAuthorities": {
            "ca.org1.example.com": {
                "url": "http://ca.org1.example.com:7054",
                "caName": "ca.org1.example.com"
            }
        }
    }
```

B.3. Hyperledger Composer Model Definition

```
namespace io.dev.foodchain


```

```
4 /**************************************************************************/
5 /* Participant definitions                                                */
6 /**************************************************************************/
7
8 abstract participant Company identified by gln {
9   o String gln
10  o String name
11  o Address address
12  o CompanyType type
13 }
14
15 participant Farmer extends Company {}
16
17 participant Slaughterhouse extends Company {}
18
19 participant CuttingPlant extends Company {}
20
21 participant Manufacturer extends Company {}
22
23 participant Wholesale extends Company {}
24
25 participant Retailer extends Company {}
26
27 participant Consumer identified by consumerId {
28  o String consumerId
29  o String name optional
30 }
31
32 /**************************************************************************/
33 /* Asset definitions                                                      */
34 /**************************************************************************/
35
36 asset Material identified by materialId {
37  o String materialId
38  o MaterialType type
39  o MaterialQuality quality
40  o MaterialStatus status default = 'CREATED'
41  o Boolean bonus optional
42  --> Batch batch optional
```

```
43   --> Company owner optional
44   --> Company[] ownerHistory optional
45   o TransportLog[] transportLog optional
46   o SensorData[] sensorData optional
47 }
48
49 asset Batch identified by batchId {
50   o String batchId
51   o DateTime timestamp
52   o String description optional
53   o Integer availableMaterialCount default = 0
54   --> Material[] materials
55   o BatchStatus status
56 }
57
58 asset BatchNetwork identified by batchNetworkId {
59   o String batchNetworkId
60   o String description
61   --> Batch[] nodes optional
62   o Edge[] edges optional
63 }
64
65 /**************************************************************************/
66 /* Concept & Enumeration definitions                                      */
67 /**************************************************************************/
68
69 concept Address {
70   o String street
71   o String number
72   o String postCode
73   o String country
74 }
75
76 concept TransportLog {
77   o DateTime timestamp
78   o Location location
79 }
80
81 concept Location {
```

```
82   o String latitude
83   o String longitude
84   o String altitude
85   o String description optional
86 }
87
88 concept SensorData {
89   o DateTime timestamp
90   o String key
91   o String value
92 }
93
94 concept Edge {
95   o String edgeId
96   --> Batch source
97   --> Batch target
98   o String weight optional
99 }
100
101 enum CompanyType {
102   o LANDWIRT
103   o SCHLACHTHOF
104   o ZERLEGUNG
105   o PRODUKTION
106   o GROSSHANDEL
107   o EINZELHANDEL
108   o TRANSPORT
109 }
110
111 enum MaterialType {
112   o SCHWEIN
113   o SCHWEINEHAELFTE
114   o KOPF
115   o RUECKEN
116   o VORDERKEULE
117   o RUMPF
118   o HINTERKEULE
119   o SALAMI
120   o BRATWURST
```

```
121 }
122
123 enum MaterialQuality {
124   o LOW
125   o MID
126   o HIGH
127 }
128
129 enum MaterialStatus {
130   o CREATED
131   o REGISTERED
132   o AVAILABLE_FOR_PRODUCTION
133   o IN_TRANSIT
134   o PROCESSED
135 }
136
137 enum BatchStatus {
138   o CREATED
139   o REGISTERED
140   o AVAILABLE_FOR_PRODUCTION
141   o IN_TRANSIT
142   o PROCESSED
143 }
144
145 /**************************************************************************/
146 /* Transaction definitions                                                */
147 /**************************************************************************/
148
149 transaction produceMaterial {
150   o Material newMaterial // new material object
151   --> Material[] inputMaterial optional // processed material (set status to '
      processed')
152   o Integer inputMaterialCount optional
153   --> Batch[] inputBatch optional // processed batch (set status to 'processed')
154   o Integer inputBatchCount optional // how many materials from this batch are
      used for production
155 }
156
157 transaction transportMaterial {
```

```
158   --> Material material
159   --> Company destination
160 }
161
162 transaction changeMaterialOwnership {
163   --> Material material
164   --> Company newOwner
165 }
166
167 transaction sellMaterial {
168   --> Material material
169 }
170
171 transaction setMaterialStatus {
172   --> Material material // material to be modified
173   o MaterialStatus newStatus
174 }
175
176 transaction addBatchNetworkNode {
177   --> BatchNetwork network // batch network to be modified
178   --> Batch node
179   o Edge edge optional
180 }
181
182 transaction addBatchNetworkEdge {
183   --> BatchNetwork network // batch network to be modified
184   o Edge edge
185 }
186
187 transaction addSensorData {
188   --> Material material // material to be modified
189   o SensorData data
190 }
191
192 transaction addTransportLog {
193   --> Material material // material to be modified
194   o TransportLog logEntry
195 }
196
```

```
197 transaction generateMockMasterData {}
198 transaction generateMockTransactionData {}
199 transaction removeMockMasterData {}
200 transaction removeMockTransactionData {}
201
202 /**************************************************************************/
203 /* Event definitions                                                      */
204 /**************************************************************************/
205
206 event changedMaterialOwnershipNotification {
207     --> Material changedMaterial
208 }
```

C. Interviewguide

1. Allgemeines

- Vorstellung vorläufige Ergebnisse Masterarbeit - Uni Oldenburg
- Thema „Chargenrückverfolgung in der Fleischwarenindustrie“
- Interviewverlauf
  - Kurze Einleitung: technischer / beruflicher Hintergrund Befragter
  - Präsentation der Ergebnisse
  - Diskussion im Anschluss
  - Zwischenfragen / Anmerkungen jederzeit möglich
  - Dauer: circa 30 Minuten
- Informationen und Angaben werden nur für wissenschaftliche Zwecke verwendet
- Gespräch wird aufgezeichnet - Einverstanden?
- Nach dem Interview kann eine verschriftlichte Form des Interviews ausgehändigt werden
- Fragen?
- Aufnahme beginnt (*Aufnahme starten*)

2. Einleitung

- Beschreibe deine aktuelle Tätigkeit
- Inwiefern spielen innovative Ideen in deinem Beruf eine Rolle?

3. Präsentation Ergebnisse

4. Diskussion

- Was ist dein erster Gedanke zum gerade präsentierten Prototyp?
- Wie denkst du über das Konzept, die Chargenrückverfolgung über eine Blockchain abzuwickeln?
- Wie ist deine Einschätzung zur Implementierung?
- Was hälst du vom gewählten Anwendungsfall bzw. den Rahmenbedingungen?
- Wie schätzt du das Potential des gezeigten Prototypen ein?
- Fallen dir weitere Anwendungsfälle ein?

5. Ende des Interviews

- Information: Interview ist vorbei, Aufnahme wird gestoppt (*Aufnahme stoppen*)
- Vielen Dank für die Teilnahme
- Sollten noch Fragen aufkommen, bitte kontaktieren

D. Transkription Experteninterview

Datum und Ort:	14.10.2019
Dauer:	31:24 Minuten
Interviewer:	Nils Lutz
Datum der Transkription:	15.10.2019
Transkriptionsregeln:	Vollständige Sprachglättung Pausenlänge in Sekunden: (.)/(..)/(...) Nonverbale Äußerungen: (*lachen*) Nicht-sprachliche Ereignisse: ((*Unterbrechung*)) **I:** Interviewer, **B:** Befragter

I: Guten Morgen, kannst du vielleicht einfach mit einer Beschreibung deiner Position im Konzern beginnen?

B: Ja klar, also ich habe bei Westfleisch angefangen als IT-Koordinator in einem unserer Produktionswerke in Coesfeld und bin aktuell bei Westfleisch stellvertetender IT-Leiter auf Konzernebene. Zum Verständnis Westfleisch ist ein Zusammenschluss aus vielen einzelnen Unternehmen mit jeweils eigenen IT-Abteilungen. Es gibt Produktionswerke für die verschiedenen Produkte die wir anbieten, Transport- und Logistik Unternehmen um die Roherzeugnisse innerhalb der Unternehmensgruppe zu bewegen, sowie Finanzverwaltungsunternehmen. Meine Aufgabe ist es auf strategischer Ebene die IT-Abteilungen all dieser einzelnen Unternehmen zentral zu steuern, damit wir die, von der Konzernleitung definierten, Ziele erreichen können. Außerdem bin ich der Projektleiter auf unserer Seite für das Forschungsprojekt REIF, was ja in Zusammenarbeit mit der Firma CompanyMind, Jade Hochschule, TU München und dem Frauenhofer Institut offiziell im September vom BMWi ausgezichnet worden ist.

I: Danke, ich würde dir dann jetzt erstmal gerne meine Ergebnisse zeigen und anschließend mit dir in eine offene Diskussion gehen, um dein Feedback dazu mitnehmen zu können.

B: Klingt gut, dann zeig mal her!

I: Moment. ((*Öffnen der Präsentation*)) Also Arbeitstitel ist „Chargenrückverfolgung in der Fleischwarenindustrie - Konzeption und prototypische Implementierung einer Blockchain Lösung“. Ich habe mich also damit beschäftigt ob es möglich ist die Chargenrückverfolgung über eine Blockchain abzubilden. Ich bin zu dem Thema grundsätzlich über die Technologie gekommen. Das heißt ich wollte was mit der Blockchain machen und habe mir dazu Probleme aus der Wirtschaft gesucht die man eventuell mit einer Blockchain besser lösen könnte als mit bisherigen Lösungen. Und da wir ja schon etwas länger mit euch zusammenarbeiten innerhalb der App-Entwicklung war es für mich halt naheliegend einen Use-Case aus der Fleischwarenindustrie zu nehmen. Mir war ja bekannt, das ihr bereits SAP Global Track & Trace im Einsatz habt zum Chargenmanagement und SAP ebenfalls auf den Blockchain Zug aufgesprungen ist. Das war so die Motivation im Grunde wieso die Konstellation Blockchain, Fleischwarenindustrie, Westfleisch zustande gekommen ist. Man hört ja immer wieder mal in den Medien, dass Produkte verunreinigt sind und es große Rückrufaktionen gibt. Letztendlich habe ich dann bei meiner Recherche herausgefunden, dass solche Rückrufaktionen teils ziemlich lange dauern in der Vorbereitung. Also es gibt Fälle wo zum Beispiel eine verunreinigte Snickers Charge knapp 4 Wochen noch im Umlauf war, einfach weil es so lange gedauert hat rauszufinden a) wo die Verunreinigung begonnen hat und b) in welche Produktionschargen diese Verunreinigung dann weiter getragen wurde.

B: Okay, so weit konnte ich folgen ((*lachen*)). Thema Blockchain (..) ist ja in aller Munde zur Zeit. Damit haben wir auch Berührungspunkte im Forschungsprojekt, da wir dort auch versuchen die Wertschöpfungskette der Lebensmittelbranche zu optimieren. Bevor wir da jetzt tiefer einsteigen, kannst du mir einen kurzen Abriss zur Technologie geben?

I: Ja klar, habe ich sowieso in der Präsentation mit drin auf Grund der Aktualität des Themas. ((*Folienwechsel*)) Ich versuch das mal so kurz wir möglich zu halten, wenn irgendwas noch unklar sein sollte einfach eben zwischenfragen. Also du kannst dir eine Blockchain grundsätzlich erstmal als Datenbank vorstellen. Diese Datenbank garantiert dir jetzt, dass alle Datensätze die darin erfasst wurden zu keinem Zeitpunkt mehr ungewollt verändert werden kön-

nen. Zusätzlich verspricht die Technologie eine Art Failsafe Betrieb. Das heißt es ist wie ein Cluster DBMS zu verstehen, fällt ein Knoten des Systems aus ist dadurch nicht das Gesamtsystem betroffen und es arbeitet weiter. Dann hast du sicherlich schon von Smart Contracts gehört. Diese musst du dir vorstellen wie eine „Stored Procedure“ in einer Datenbank, nur das du ein paar mehr Möglichkeiten hast Geschäftslogik darin auszudrücken. So weit ist das nichts neues, der Clou ist jetzt aber das so ein System im besten Fall vollständig dezentral aufgestellt ist. Das heißt diese „Datenbank“ wird nicht zentral bei Westfleisch betrieben, sondern es wird ein Netzwerk mit den verschiedenen Teilnehmer der Wertschöpfungskette aufgespannt. Ich hab mich ja für die Chargenrückverfolgung entschieden, entsprechend besteht mein konzipiertes Netzwerk aus Landwirten, Mästern, Produktionswerken und dem Groß- bzw. Einzelhandel. Jeder dieser Teilnehmer betreibt mindestens einen Knoten auf dem das Blockchain System arbeitet. Wenn jetzt ein neuer Datensatz erfasst werden soll, sagen wir mal ein Landwirt will seine Schweine zur Schlachtung anmelden. Dann spricht man bei der Blockchain von Transaktionen. Mit dieser Transaktion wird dem Blockchain System gesagt, es soll ein neuer Datensatz mit den erfassten Informationen angelegt werden. Der Systemknoten des Landwirts verschickt diese Transaktion dann an alle Knoten im Netzwerk und lösen dadurch den Smart Contract, also die Geschäftslogik aus. Alle Knoten prüfen dann ob die Transaktion nach der Geschäftslogik valide ist und stimmen dann darüber ab, ob der Datensatz hinzugefügt werden soll. Gehen wir mal davon aus, die Transaktion war valide und wurde hinzugefügt. Jetzt ist es nicht mehr möglich das ein einzelner Knoten nachträglich Werte des Datensatz verändert ohne das alle anderen Knoten davon etwas mitbekommen würden, dazu verwendet die Blockchain Hashwerte und digitale Signaturen wie du es aus der kryptographischen Verschlüsselung kennst. Sprich, sollte ein Knoten doch etwas an einem Datensatz verändern könnten alle anderen Teilnehmer dies sofort herausfinden in dem sie einfach die Hashwerte der Datensätze miteinander vergleichen und feststellen würden, dass da etwas nicht passt. Daher kommt eigentlich auch der Begriff Blockchain. Alle Datensätze werden in einer Kette aus Blöcken gespeichert und der aktuellste Block referenziert immer auf den Hashwert des vorigen Blocks.

B: Macht Sinn, Hashwerte sagen mir noch was aus meinem Studium ((*lachen*)) auch wenn das schon etwas länger her ist. (...) Wie hast du das jetzt angepasst, um es für eine Chargenrückverfolgung nutzbar zu machen?

I: ((*Folienwechsel*)) Also angefangen habe ich mit der Suche nach einem geeigneten Blockchain System für den industriellen Einsatz. Man kennt ja Bitcoin, aber die Blockchain von Bitcoin eignet sich in diesem Fall eher weniger, da es eine öffentliche Blockchain ist in der jeder als Netzwerkteilnehmer mitmachen kann und alle Transaktionen öffentlich einsehbar sind. Zusätzlich ist die Verarbeitungsgeschwindigkeit bei vollständig öffentlichen Blockchains aktuell noch vergleichsweise langsam. Da sind konventionelle Datenbanksysteme um ein vielfaches schneller. Es gibt allerdings auch Blockchain Systeme die speziell für den Einsatz in der Industrie entwickelt wurden. Dazu gehört unter anderem Hyperledger Fabric. Das wurde ursprünglich mal von IBM konzipiert und entwickelt und dann als Open-Source Software an die Linux Foundation übergeben. Dadurch kann zum Beispiel jeder in den Quelltext gucken und sicherstellen, das das System wirklich so arbeitet wie es angepriesen wurde. Außerdem sind Blockchain Netzwerke die mit Hyperledger Fabric gebaut wurden in der Regel nicht öffentlich und es gibt klar definierte Regeln wer als Teilnehmer in dem Netzwerk auftreten kann. Es ist nicht zwingend notwendig, das jeder Teilnehmer auch immer die Transaktionen validiert. Ein Unternehmen kann zum Beispiel drei Knoten im Netzwerk betreiben, aber nur zwei Knoten validieren neue Transaktionen und der dritte Knoten sorgt einfach für mehr Ausfallsicherheit für das gesamte Netzwerk.

B: Also hast du dich für dieses Hyperledger entschieden auf Grund der Geschwindigkeit und dem Fokus der Software auf den industriellen Sektor?

I: Ja genau, die Geschwindigkeit war ein ausschlaggebender Punkt. Dazu war es für mein Konzept ebenfalls wichtig das ich die Geschäftslogik möglichst in einer Sprache implementieren kann die ich auch verstehe ((*lachen*)). Und natürlich der Aspekt das Hyperledger Fabric ein private Blockchain System ist, wo ich kontrollieren kann wer alles im Netzwerk mitmachen kann. Zusätzlich gibt es für Hyperledger Fabric ein Framework das sich Hyperledger Composer nennt, mit dem man dann die Smart Contracts entwickeln kann die dann später

in das Netzwerk installiert werden. Ich würde dir als nächstes einfach mal ein kurzes Beispiel zeigen wie so eine Transaktion im Netzwerk abläuft und welche Zugriffsmöglichkeiten man dann hat, um mit dem System als Anwender bzw. aus anderen Anwendungen heraus zu interagieren. ((*Demo am System*))

B: Okay, das ist ziemlich interessant. Mir kommen da auch direkt ein paar Einsatzgebiete bei uns ((*lachen*)) aber erzähl du erstmal weiter.

I: Ja im Grunde würde ich jetzt gerne in die offene Diskussion über gehen, um einfach mal von dir zu hören was du so darüber denkst und wie du grad schon meintest wo man es vielleicht noch einsetzen könnte.

B: Also erstmal muss ich sagen, das sieht für einen Prototypen schon relativ ausgereift aus, wenn man es jetzt auf den Use-Case Chargenrückverfolgung für Schweine belässt. Generell denke ich, ich habe den Ansatz dahinter verstanden. Würden wir dieses System jetzt bei Westfleisch einsetzen, müssten wir nur gucken, das wir alle unsere Zulieferer und Kunden mit ins Boot holen, das die sich alle so einen Blockchain Knoten hinstellen, richtig?

I: Genau, es lässt sich in der Theorie auch mit nur einem Knoten betreiben, dann geht aber der Sinn eines dezentralen Systems verloren. Daher habe ich es auch so konzipiert, dass jeder Teilnehmer im Minimum einen Knoten betreiben muss, um die Eigenschaften der Blockchain ausreizen zu können.

B: Absolut! (..) Also können wir über die Smart Contracts unsere Geschäftslogik abbilden bzw. auch unsere Zulieferer und Endkunden. Wenn man bedenkt das wir so knapp 130 Kunden haben und jeder Kunde uns ein anderes Format für ihre Chargeninformationen vorgibt bzw. nutzt, dann würde so ein System schon wirklich Sinn machen allein aus Gründen der Standardisierung. Ich habe ja auch das Forschungsprojekt REIF im Hinterkopf bei der ganzen Geschichte. Da versuchen wir ja auch mit KI und maschinellem Lernen unsere Produktionsplanung zu optimieren. Wenn jetzt die Landwirte in der Lage sind über dein System ihr Vieh bei uns anzumelden und wir mit dem maschinellem Lernen dann ebenfalls auf die Daten zugreifen könnten (...) ja dann erhöht sich zum einen bei uns die Planungssicherheit und zum anderen könnten wir Informationen zu den Chargen bzw. Produkten in beide Richtungen der Wertschöpfungskette bereitstellen mit der Garantie das diese Daten der Wahrheit

entsprechend. Das bringt ja Vertrauen und grade der Endverbraucher wird immer kritischer und möchte soviele Informationen zu seinem Produkt wie möglich haben und sich dabei auch sicher sein, das die Unternehmen ihm da die Wahrheit erzählen.

I: Da unterstützt dich die Technologie dann mit den kryptographischen Methoden wie die digitalen Signaturen. Sowas kennen viele ja bereits von Websiten wenn sie das kleine grüne Schloss neben der URL sehen. Dann wissen sie das diese Seite sicher ist, weil sie sich indirekt darauf verlassen können das die Zertifikate offiziell beglaubigt wurden sozusagen. Ich mein für die meisten reicht sicher das grüne Schloss und was da im Hintergrund läuft ist nicht so wichtig, aber wenn dann doch mal etwas ist grade bei Lebensmitteln oder zum Beispiel Finanzen möchte man als Endanwender ja doch eine gewisse Sicherheit haben das alles mit rechten Dingen zugeht. Kannst du vielleicht noch etwas zur Implementierung und den damals gewählten Rahmenbedingungen sagen?

B: Also ich hab gesehen das du die Oberflächen mit Fiori modelliert hast. Das kenne ich ja von den anderen Apps, die ihr bereits für uns entwickelt habt. Da könnte man sich sicherlich nochmal mit den Fachabteilungen hinsetzen und gucken das man da einen Feinschliff reinbekommt. Ich mein, für einen Prototyp ist das aus meiner Sicht völlig ausreichend, aber wenn man sowas dann auf einer Messe präsentieren möchte vielleicht im Zusammenspiel mit einem KI Systen ((*lachen*)) dann muss sowas ja heutzutage alles sehr gut aussehen. Sonst kriegt man die Kunden nicht abgeholt. Zu deiner Implementierung im Backend kann ich nicht ganz soviel sagen, die Datenmodelle sehe schlüssig aus und die Demo hat ja gezeigt, dass man damit den Geschäftsprozess komplett abbilden kann. Da würde ich sagen: Passt! Die Rahmenbedingungen hattest du glaub ich zu Beginn deiner Arbeit mit unserer Infrastruktur Abteilung abgestimmt richtig?

I: Genau, so war das. Ich wollte halt die Rahmenbedingungen so definieren, das das System nicht nach der Konzeption bzw. der prototypischen Implementierung in der Schublade verschwindet. Deswegen war mir wichtig da schon früh eine Art Sparringspartner aus der Industrie zu haben.

B: Ja dann denke ich das die Rahmenbedingungen schon erfüllt sind, sicher lassen sich noch weitere finden wenn man das System vom Prototypenstatus in einen

Pilotbetrieb überführen will, aber das war ja nicht Bestandteil deiner Arbeit. Insofern passt das.

I: Also würdest du sagen der Prototyp besitzt ein gewisses Potential?

B: Definitiv! (..) Der entscheidende Punkt ist aus meiner Sicht ist oft die Marktdurchdringung. Du kannst noch so tolle Systeme und Technologien entwickeln, wenn niemand am Markt oder in der Branche dieses System nutzt, aus welchen Gründen auch immer, dann wird dieses System keinen Erfolg haben. Deshalb sollte man im Blick behalten, das mit so einem System eine Art „Win-Win" Situation hergestellt wird. Wenn ich als Teilnehmer des Netzwerk etwas hineingebe muss ich auch immer etwas herausbekommen, sonst sinkt mein Interesse dieses System zu verwenden. Und wie du erklärt hast, steigt die Sicherheit und der Informationsgehalt etc. mit der Anzahl der Teilnehmer. Ich denke da immer an das Beispiel mit den Elektroautos. Die Technologie ist super, aber wenn wir kein Ladenetz haben wird niemand sich ein Elektroauto kaufen. So einfach ist. ((*lachen*))

I: Danke (..) du hattest am Anfang schon gesagt das dir direkt weitere Anwendungsfälle in den Kopf kommen. Kannst du darüber noch was sagen? Danach wäre ich auch durch mit meinen Fragen. ((*lachen*))

B: Natürlich, wie gesagt generell könnte man so ein System für sämtliche Tierarten erweitern, die wir so durch die Produktionswerke schieben. Obendrauf wäre es ziemlich interessant Auswertungen der Veterinäre mit zu erfassen. Endkunden wollen wissen wieviel Antibiotika in ihrer Wurst steckt. Grade bei Hühnerfleisch, da legen die Käufer sehr viel Wert drauf mittlerweile. Außerdem wird immer öfter nicht nur auf die Art und Weise der Haltung geschaut, sondern auch was die Tiere während ihres Lebens als Futter bekommen haben. Das geht soweit, das solche Angaben einen höheren Preis des Endprodukts vollständig rechtfertigen und der Markt für solche Produkte ist da. Das lässt sich nicht mehr bestreiten. Aus Sicht der Wertschöpfungskette, also sagen wir mal aus Sicht eines Großhandels werden Transport- und Logistikinformationen immer wichtiger. Ich will wissen, wann und wie lange waren die Tiere unterwegs und wie ist ihr Zustand während des Transports. Sowas wird aktuell schon über ein paar wenige Sensoren und die Prüfung durch den Spediteur

ermittelt. Könnte man diese Informationen jetzt noch mit in die Blockchain packen hätte man alles an einem Ort, was widerrum Futter für die KI bzw. die Algorithmen des maschinellen Lernen ist ((*lachen*)) Ich denke da lässt sich noch einiges mit machen, wenn man so ein System im Sinne der Industrie 4.0 einsetzt und möglichst viele Informationen dort reinspeichert und sich stets sicher sein kann das dort nachträglich niemand an den Daten rumpfuschen kann. Demnach hast du mit deiner Arbeit ein ganzen Stück an Vorarbeit für das Forschungsprojekt REIF geleistet und bewiesen das eine Rückverfolgbarkeit mit dieser Technologie vom Landwirt bis zum Endkunden machbar ist.

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 21. Oktober 2019 Nils Lutz