



## Master Thesis

im Rahmen des  
Universitätslehrganges „Geographical Information Science & Systems“  
(UNIGIS MSc) am Interfakultären Fachbereich für Geoinformatik (Z\_GIS)  
der Paris Lodron-Universität Salzburg

zum Thema

# **Distributed Ledger Technologien in der Liegenschaftsverwaltung**

-

Mehrwert in der Verwaltung von Liegenschaften  
mittels Blockchaintechnologie

vorgelegt von

**Stefan Rutka, B.Sc.**  
103465, UNIGIS MSc Jahrgang 2015  
stefan.rutka@posteo.de

Gutachter

Ao. Univ. Prof. Dr. Josef Strobl

Zur Erlangung des Grades  
„Master of Science (Geographical Information Science & Systems) – MSc (GIS)“

Tübingen, den 20. Februar 2019

## ERKLÄRUNG DER EIGENSTÄNDIGEN ABFASSUNG DER ARBEIT

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen ist. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind entsprechend gekennzeichnet.

*Stefan Rutka*

Tübingen, 20.02.2019  
Stefan Rutka

## ZUSAMMENFASSUNG

Distributed Ledger Technologien sind Arten von Datenbanken, die oft auf verschiedene Standorte und mehrere Anwender verteilt sind. Hierunter wird auch die Blockchain-Technologie eingeordnet, welche sich aktuell an wachsender Aufmerksamkeit erfreut.

Blockchains sind eine noch junge und vielversprechende Methode, um Transaktionen fälschungssicher, dezentral und transparent zu dokumentieren. Sie bieten mit Ihren technischen Neuerungen für viele bestehende Probleme ganz neue Möglichkeiten, um Lösungsansätze zu entwickeln, wie erste Leuchtturmprojekte in verschiedenen Branchen erfolgreich aufzeigen.

Auch die Geoinformatik als dynamisches und interdisziplinäres Forschungsfeld ist gefordert, (neue) Anwendungsfälle zu explorieren sowie Vor- und Nachteile der noch jungen Technologie abzuwägen. Seit 2018 steigt die Intensität der Diskussion um den abgeleiteten Begriff Geoblockchain bemerkenswert.

Diese Arbeit liefert einen Beitrag zur jungen Debatte um die Anwendung von Blockchaintechnologien im raumbezogenen Kontext. Hierzu wird anhand der amtlichen Liegenschaftsadministration untersucht, ob durch den Einsatz der Technologie ein Mehrwert erzielt werden kann. In diesem Zusammenhang wird auch untersucht, inwiefern sich die neue, dezentrale Datenhaltung von der klassischen und etablierten zentralen Datenhaltung unterscheidet und ob die Technologie an sich für den gewählten Anwendungsfall Verbesserungen bringt.

Praktisch umgesetzt werden in dieser Arbeit einige ausgewählte Anwendungsfälle aus der amtlichen Liegenschaftsverwaltung, zum einen die Eigentümerverwaltung von Flurstücken im Allgemeinen, zum anderen aber auch „technische Sonderfälle“ wie Flurstücksteilung, -verschmelzung oder Neuordnung. Die Blockchain soll hier den gesamten Lebenszyklus eines Flurstücks abbilden: Von der Entstehung über diverse Änderungen bis hin zur „Auflösung“ bzw. Nachfolge.

Im Rahmen eines Proof-of-Concept wird unter Verwendung des Hyperledger Frameworks, welches von der LINUX Foundation gesteuert und maßgeblich gefördert wird, dieser Anwendungsfall untersucht. Das Framework kann mittels verschiedener Open Source Lizenzen genutzt werden und wird neben der Community auch von namhaften Unternehmen wie IBM, Daimler oder SAP verwendet und gefördert.

Es wird erwartet, dass obgleich einer nutzbringenden prototypischen Umsetzung noch viel Forschung und Entwicklung geleistet werden muss, um wirklich große Anwendungen in Serienreife zu bringen. Das ist allerdings nur eine Frage der Zeit. Ein viel höherer Aufwand besteht darin, die Blockchain so zu konzipieren, dass die Vorteile gezielt genutzt und die Nachteile minimiert werden.

Keywords: Blockchain, Geoblockchain, Hyperledger, Liegenschaftsverwaltung, Distributed Ledger

# INHALTSVERZEICHNIS

Erklärung der eigenständigen Abfassung der Arbeit .....	2
Zusammenfassung .....	3
Inhaltsverzeichnis .....	4
1. Einleitung .....	6
1.1. Motivation .....	6
1.2. Problemstellung .....	6
1.3. Literaturüberblick .....	7
1.4. Forschungsfrage .....	8
1.5. Struktur der Arbeit .....	9
2. Technische Grundlagen Blockchain .....	10
2.1. Begriff und Definition .....	10
2.1.1. Distributed Ledger Technologien .....	10
2.1.2. Blockchain .....	10
2.1.3. Beispiele für Blockchain-Anwendungen .....	12
2.1.4. Geoblockchain .....	13
2.1.5. Beispiele für Geoblockchain-Anwendungen .....	14
2.2. Funktionsweise & Aufbau .....	15
2.2.1. Ledger & World-State .....	18
2.2.2. Peer Network .....	18
2.2.3. Membership Services & Membership Service Provider .....	21
2.2.4. Smart Contracts / Chaincode .....	21
2.2.5. Transaktionen .....	23
2.2.6. Consensus-Algorithmen .....	23
2.3. Öffentliche vs. private Blockchain .....	23
2.4. Abgrenzung zu zentraler Datenhaltung .....	24
2.5. Entwicklung & Stand der Technik .....	26
2.6. Chancen & Risiken der Technologie .....	27
3. Anforderungsanalyse .....	29
3.1. Use Case .....	29
3.2. Anforderungen .....	35
3.3. Zwischenevaluation .....	36
3.4. Verwendete Software .....	38
3.5. Verarbeitete Daten .....	39
4. Umsetzung .....	40
4.1. Grundlagen zur Anwendung .....	40
4.2. Architektur .....	41
4.3. Chaincode .....	44
4.4. Datenmodell .....	46

4.5. Methoden .....	46
4.5.1. Invoke .....	46
4.5.2. recordFlurstueck .....	47
4.5.3. initLedger .....	48
4.5.4. queryFlurstueck.....	48
4.5.5. queryAllFlurstueck.....	49
4.5.6. changeFlurstueckEigentuemer.....	50
4.5.7. Kombinationen .....	50
4.6. Aufbau der Anwendung FlurstueckApp .....	51
4.7. Benutzeroberfläche .....	51
4.8. SQL für Zugriff auf aktuellen Geodatenbestand.....	52
5. Ergebnisse .....	53
5.1. Initialisierung neuer Datensätze .....	53
5.2. Abfragen der Blockchain .....	54
5.2.1. Gesamten Inhalt abfragen.....	54
5.2.2. Spezielle Inhalte abfragen.....	55
5.3. Änderung von Eigentumsverhältnissen .....	55
5.4. Sonderfälle.....	57
5.4.1. Verschmelzung von Flurstücken.....	57
5.4.2. Teilung von Flurstücken .....	59
5.4.3. Umlegung von Flurstücken.....	60
5.5. Verwaltung von Geometrien in der Blockchain .....	61
5.5.1. Keine Geometrien .....	62
5.5.2. Koordinaten als WKT .....	62
5.5.3. GeoJSON.....	62
5.6. Erfüllung der Anforderungen .....	62
6. Diskussion.....	64
6.1. Beantwortung der Forschungsfrage .....	64
6.2. Fazit .....	66
6.3. Ausblick.....	67
Literaturverzeichnis .....	69
Tabellenverzeichnis.....	71
Abbildungsverzeichnis.....	72
Abkürzungsverzeichnis.....	73

# 1. EINLEITUNG

## 1.1. MOTIVATION

Die raschen Entwicklungen innerhalb der Informationstechnologie im Allgemeinen und in der Geoinformatik im Speziellen werfen ständig neue Fragen auf, die im Sinne grundlegender Forschung untersucht werden müssen.

Distributed Ledger Technologien, wie beispielsweise Blockchains, erfreuen sich aktuell einer wachsenden Aufmerksamkeit. Im Angesicht der Digitalisierung und dem Aufkommen neuer Technologien, (Forschungs-) Ansätze oder Leitbildern wie Internet-of-Things, Smart Cities oder Industrie 4.0 befindet sich unsere Gesellschaft aktuell in einer dynamischen und vielseitigen Transformation mit vielen neuen Mitteln und Möglichkeiten.

Gerade die Blockchaintechnologie als junge und vielversprechende Methode der dezentralen Datenspeicherung mit Ihren Vor- und sicherlich auch Nachteilen drängt sich als echte Alternative zu klassischen zentralen Datenbanken/Datenbanksystemen geradezu auf. Sie bietet das Potenzial unsere Gesellschaft in Bezug auf elektronische Datenverarbeitung auf Jahre hinweg zu verändern, weiterzuentwickeln und zu prägen (Schlatt et al. 2016).

Die Geoinformatik als dynamischer und interdisziplinärer Fachbereich ist gefordert, sich mit diesen Entwicklungen wissenschaftlich auseinanderzusetzen.

Diese Arbeit soll die Anwendung der Blockchaintechnologie in Bezug auf räumliche Fragestellungen untersuchen und zur noch jungen wissenschaftlichen Diskussion über Chancen und Risiken von Geoblockchains beitragen.

## 1.2. PROBLEMSTELLUNG

Obwohl der Tatsache vieler bestehender, aktueller Anwendungsfälle von Blockchain-technologien scheint die Umsetzung auf neue Fragestellung nicht in jedem Fall trivial. Neue Technologien bedürfen gerade in der Phase der Einführung viel Forschung, Entwicklung und auch Aufklärung, bis eine breite Akzeptanz zustande kommt. Der Einsatz neuer Technologien bedeutet -abseits der zu erwartenden Vorteile- immer auch einen Grad an Unsicherheit, der viele Fragen aufwirft:

- ⊕ Ist eine Systemumstellung sinnvoll?
- ⊕ Wie sicher ist die neue Technologie?
- ⊕ Welche Risiken bestehen?
- ⊕ Welche Erfolgs- und Misserfolgsgeschichten gibt es?
- ⊕ Wie weit ist der Stand der Entwicklung/Stand der Forschung?
- ⊕ Wie zukunftssicher ist eine Investition in diese Technologie?
- ⊕ Lohnt sich die Investition auch wirtschaftlich?

Diese Arbeit untersucht die Anwendbarkeit von Distributed Ledger Technologien im Kontext raumbezogener Aufgaben. Als Fallbeispiel wurde ein Einsatzszenario aus der Liegenschaftsverwaltung herangezogen, dabei hat sich der Autor auch an der Diskussion um dem Begriff der „Cryptogovernance in Land Administration“ orientiert (Adams & Tomko 2018, Chapron 2017, Lemmen et al. 2016). Der Begriff umschreibt den Einsatz einer dezentralen Verwaltung unseres Landbesitzes mittels einer digitalen Währungseinheit oder eines

digitalen Besitzeignisses/Eigentumsnachweises. Im Grunde soll die eingesetzte Blockchain Möglichkeiten aufzeigen um einige wichtige Kriterien der Landverwaltung verbessern und unter Umständen auch einen direkten Handel der Flurstücke zwischen Käufer und Verkäufer ermöglichen, ohne dass der Einsatz einer Kontrollinstanz nötig wäre.

Eine Geoblockchain könnte, angelehnt an bestehende erfolgreiche Blockchains, helfen, das technische Umfeld dafür zu schaffen – eine dezentrale und verteilte Datenhaltung.

Hierzu muss sichergestellt werden, dass grundlegende Aufgaben der Liegenschaftsverwaltung, welche aktuell größtenteils zentral abgewickelt werden, auch in diesem dezentralen Szenario funktionieren. Dabei ist vor allem drauf zu achten, ob die Datenhaltung integer, manipulationssicher und entsprechend verfügbar ist. In das Szenario einbezogen werden sollen neben einer (bisher) zentralen datenhaltenden Stelle (Landesamt) auch Datennutzer wie Kommunen oder Datenlieferanten wie öffentlich bestellte Vermessungsingenieure. Die Geoblockchain soll durch eine entsprechende Administration in der Lage sein, die unterschiedlichen Akteure mit den notwendigen Rechten auszustatten, damit die Verwaltung der Gesamtdaten deutlich einfacher wird.

Bei der Betrachtung des Mehrwertes wird auch Pohlmann miteinbezogen, der der Ansicht ist, dass gerade die Blockchaintechnologie in der Lage ist durch Dezentralität Prozesse zu vereinfachen und auch das Vertrauen in und die Validität von Daten zu steigern (Pohlmann 2018).

Im Rahmen dieser Thesis werden zentrale Teile eines Szenarios mittels eines Proof-of-Concept umgesetzt und diskutiert. Dabei ist zu erwarten, dass nicht alle Teile der Blockchain im Detail geprüft werden können. Der Fokus liegt in der Machbarkeit und einer generellen Umsetzung im Fokus der Verwaltung von Flurstücken und den dafür benötigten Bestandteilen des Frameworks. Ob eine unabhängige Kontrollinstanz wirklich entfallen könnte, muss vermutlich nicht nur auf technischer, sondern auch auf politischer Ebene diskutiert werden.

### 1.3. LITERATURÜBERBLICK

Um sich der Fragestellung dieser Arbeit hinsichtlich der Kombination von räumlichen Themen und der Anwendung von Distributed Ledger Technologien anzunähern, lohnt sich ein Rückblick auf einen Meilenstein mit dem bekannten Whitepaper von Satoshi Nakamoto über die Kryptowährung Bitcoin aus dem Jahr 2008. Nakamoto beschreibt hier den Aufbau einer Kryptowährung, welche komplett unabhängig von Banken oder Finanzinstituten sicher funktionieren soll. Auch das eingesetzte Peer-to-Peer-Netzwerk und die für die Transaktionen zuständige Technologie werden beschrieben, die sogenannte Blockchain (Nakamoto 2008).

Zu Beginn der 2010er Jahre blieben die Veröffentlichungen bzgl. Anwendungsfälle der Technologie noch überschaubar, in den letzten Jahren jedoch stieg die Zahl rapide an. Erklären lässt sich das auch durch den Einstieg namhafter Finanzinstitute und Technologiekonzerne in die Forschung und Entwicklung von Blockchains (Schlatt et al. 2016), sowie den auch medial wahrgenommenen Hype um den rasante Kursanstieg verschiedener Kryptowährungen in den Jahren 2016 und 2017, allen voran der populäre Bitcoin.

Inzwischen liefern renommierte Organisationen, wie das Institute of Electrical and Electronics Engineers (kurz: IEEE), zahlreiche hochwertige Artikel sowie verschiedene Publikationen zum Thema. Abseits grundlegender Publikationen findet sich hier auch eine Vielzahl an Veröffentlichungen zum Hyperledger-Framework, welches unter der Regie der LINUX Foundation gesteuert und von zahlreichen Unternehmen wie IBM oder DAIMLER unterstützt und gefördert wird. Das Framework Hyperledger Fabric wird in dieser Arbeit Softwarekomponente eingesetzt.

Seit 2015 entwickelt sich eine überschaubare, aber lebhafte Debatte um die Nutzung von Blockchaintechnologie im Umfeld räumlicher Herausforderungen, darunter wird häufig die spezielle Anwendung in Bezug auf Landbesitz bzw. Flurstückseigentum diskutiert (Chapron 2017, Adams & Tomko 2018). Der Begriff des „Kryptokatasters“ wird inzwischen auch von Unternehmen aus der freien Wirtschaft aufgegriffen und in Bezug auf konkrete Anwendungen rege geforscht (z.B. ConsultingWhere 2015) und Lösungen entwickelt.

Abseits klassischer Literaturzugänge und Grundlagenforschung haben sich dem Autor zwei weitere Medienangebote als wertvolle Informationsquellen zum Thema erwiesen.

Zum einen finden sich im Internet einige rege Diskussionen in verschiedenen einschlägigen Foren, doch hier wird häufig über Kryptowährung und die Technologie im Allgemeinen diskutiert. Einen ersten Einstieg in Bezug auf räumliche Fragestellungen lieferte der SLACK-Channel „blockchaingeospatial“ (<https://blockchaingeospatial.slack.com/>), der von Jonas Ellehaug, welcher im späteren Verlauf dieser Arbeit auch noch als Autor zitiert wird, gegründet und moderiert wird. Er versucht mit diesem Medium räumlichen Fragestellungen in Bezug auf die Blockchain ein einfaches, aber effizientes Forum zu bieten, welches sich aus Sicht des Autors inzwischen zu einem kleinen Netzwerk entwickelt hat.

Eine weitere hilfreiche Quelle waren verschiedene Massive Open Online Courses bei edX – siehe <https://www.edx.org/>. Dort bieten zahlreiche Universitäten und Institutionen (z.B. MIT, Harvard University und die LINUX Foundation) sowohl kostenfreie als auch kostenpflichtige Onlinekurse zu den verschiedensten Themen der Informationstechnologie, im Speziellen auch Kurse zu Blockchain-Grundlagen bis hin zur Einführung in die Hyperledger-Projekte. Die Kurse dienen auch als Grundlage der technischen Dokumentationen.

## 1.4. FORSCHUNGSFRAGE

Diese Arbeit beschäftigt sich grundlegend damit, ob Distributed Ledger Technologien bzw. Blockchains den Zugriff auf Geodaten im Allgemeinen und auf amtliche Geodaten im Speziellen in Bezug auf wesentliche Aspekte wie Manipulationssicherheit, Verwaltungsaufwand und Aktualität wirklich optimieren können.

Dabei ist zunächst zu klären, ob die bisherigen Methoden wirklich nicht ausreichend sind. In dieser Frage wird gegenübergestellt, ob der State-of-the-art, also klassische zentrale Datenbanken, die von einzelnen Sachbearbeitern in Landesämtern gepflegt werden und der Anwender (hier die Kommunen) einen jährlichen Auszug bekommen dem aufgezeigten Anwendungsfall einer verteilten Datenhaltung und damit auch einem verteilten Datenzugriff unterlegen ist.

Weiterhin wird geprüft, ob Blockchaintechnologie im genannten Anwendungsfall wirklich fähig ist, ausreichende Aktualität und Zuverlässigkeit aufzubauen. Es muss sichergestellt werden, dass Informationen rund um das Flurstück eindeutig verifizierbar und richtig sind.

Schlussendlich wird betrachtet, ob der Proof-of-Concept einer realen Umsetzung standhalten könnte. Hierbei soll auch geklärt werden, ob eine entsprechende Einführung und Nutzung nur (technisch) hochspezialisierten Institutionen mit entsprechendem Knowhow vorbehalten ist oder ob die Umsetzung auch einen Mehrwert für eine breite Masse mit sich bringt.

Zusammenfassend leiten sich folgende Teil-Forschungsfragen ab:

1. *Entsprechen klassische zentrale Datenhaltungen im Umfeld der Liegenschaftsverwaltung noch den Anforderungen?*
2. *Kann Blockchaintechnologie generell für eine Verbesserung der Aktualität von Liegenschaftsdaten sorgen?*
3. *Ist die Nutzung von Blockchaintechnologie im Umfeld amtlicher Geodaten überhaupt realistisch?*

## 1.5. STRUKTUR DER ARBEIT

Im Rahmen dieser Arbeit sollen die abgeleiteten Forschungsfragen eingehend untersucht und beantwortet werden. Nachdem in dieser Einleitung Motivation und Forschungsfragen präsentiert sowie ein erster Literaturüberblick zum Thema gegeben wurde, werden im darauffolgenden Kapitel die notwendigen Grundlagen zur Blockchaintechnologie aufgearbeitet. Nach einer grundlegenden Anforderungsanalyse folgt die Umsetzung. Es werden die grundlegende Systemarchitektur erörtert, Chaincode sowie Datenmodell vorgestellt und das Zusammenspiel von Anwendung und Blockchain dargestellt. Zuletzt werden die Ergebnisse beschrieben der Arbeit vorgestellt und auf dieser Grundlage die Forschungsfragen diskutiert. Im Fazit wird der finale Stand der Arbeit nochmal im Gesamtkontext betrachtet sowie weitere Möglichkeiten und Potenziale der Anwendung im Ausblick beschrieben.

## 2. TECHNISCHE GRUNDLAGEN BLOCKCHAIN

### 2.1. BEGRIFF UND DEFINITION

#### 2.1.1. DISTRIBUTED LEDGER TECHNOLOGIEN

Distributed Ledger bedeutet im Deutschen so viel wie „Verteiltes Kassenbuch“. Da die Literatur diesen Begriff allerdings so gut wie nie ins Deutsche übersetzt, bleibt auch der Autor beim englischen Begriff der Distributed Ledger Technologien (kurz: DLT).

Es handelt sich bei DLT um eine dezentrale Datenbank, die in einem Netzwerk den beteiligten Akteuren Zugriff auf Lese- oder Schreibrechte einräumt. Die räumliche Verteilung kann hierbei unterschiedliche Ausprägungen haben, z.B. einzelne Standorte, verschiedene Länder, Regionen usw. (Metzger 2018).

Der grundlegende Unterschied zu zentralen Datenbanken besteht darin, dass keine zentrale Instanz benötigt wird, um Einträge in der Datenbank vorzunehmen.

DLT bestehen i.d.R. aus drei Kernkomponenten (vgl. EDX 2018a):

- Einem Datenmodell das den Gesamtzustand des Ledgers abbildet
- Definierte Transaktionen die den Zustand des Ledgers verändern können
- Einem Konsensprotokoll, welches zwischen allen Teilnehmern regelt welche Transaktionen akzeptiert und in welcher Reihenfolge dem Ledger zugefügt werden

#### 2.1.2. BLOCKCHAIN

Die Blockchain ist eine bestimmte Ausprägung der DLT. Im Vergleich zur DLT an sich spielen Smart Contracts bei der Blockchain eine charakteristische Rolle.

Sie zu beschreiben und zu definieren fällt auf den ersten Blick nicht leicht, da sich bestehende Definitionen in der Literatur zum Teil stark unterschieden oder sich auf unterschiedliche Blockchain-Szenarien beziehen. Erschwerend kommt hinzu, dass „eine Blockchain“ nicht einfach ein Softwareprodukt ist, welches installiert wird und dann läuft – sondern aus verschiedenen, in sich schon umfangreichen Komponenten besteht.

Auch das Fraunhofer Institute For Applied Information Technology (kurz: FIT) kam bei einer Studie zu dem Entschluss, dass es aufgrund der Neuheit der Technologie noch keine einheitliche Definition gibt und verweist an dieser Stelle auf verschiedene Autoren (Schlatt et al. 2016).

Im Folgenden soll eine Annäherung mit dem Begriff stattfinden, obgleich im späteren Verlauf, bei der Vorstellung der Softwarekomponenten, eine weitere Eingrenzung des Begriffs im Zusammenhang mit dem eingesetzten Blockchain-Framework „Hyperledger Fabric“ stattfindet.

Eine Blockchain ist aus technologischer Sicht eine dezentrale, verteilte Datenbank, die zwischen den Teilnehmern des Netzwerks geteilt und an jedem Knoten (=Teilnehmer) repliziert wird (Christidis & Devetsikiotis 2016).

Oder wie es Schlatt et al. und Condos et al. festhalten: „Grundsätzlich ist die Blockchain ein elektronisches Register für digitale Datensätze, Ereignisse oder Transaktionen, die durch die Teilnehmer eines verteilten Rechnernetzwerkes verwaltet werden.“ (Schlatt et al. 2016, Condos, Sorrell und Donegan 2016).

Das „elektronische Register“ wird i.d.R. als Ledger bezeichnet, was übersetzt so viel bedeutet wie Konto bzw. Kassenbuch. Hierin werden die Transaktionen aufgezeichnet (Condos, Sorrell und Donegan 2016).

Die Literatur spricht von einigen „zwingenden“ Eigenschaften der Blockchain. Diese unterscheiden sich aber nach Betrachtungsweise, Entwicklungsstand über die Jahre und vermutlich auch persönlicher Vorlieben der Autoren, so setzen Niranjanamurthy, Nithya und Jagannatha 2018 die Anzahl der Eigenschaften, welche eine Blockchain aufzeigen muss, auf sechs fest:

- (1) Dezentrale Datenhaltung
- (2) Transparenz
- (3) Open Source
- (4) Autonomie
- (5) Unveränderbarkeit
- (6) Anonymität

Heumüller & Richter 2018 zählen hingegen lediglich vier grundlegende Eigenschaften:

- (1) Dezentrale Datenhaltung
- (2) Konsens
- (3) Unveränderbarkeit der Transaktionen hinsichtlich der Reihenfolge
- (4) Unveränderbarkeit der Transaktionen gegen nachträgliche Manipulationen

Wobei sie den Eigenschaften der Transparenz, Open Source und Anonymität keine Beachtung schenken. Das erklärt sich auch anhand der Literatur und der Entwicklungen in den vergangenen Jahren:

War der Transparenzbegriff in den Anfangsjahren der Blockchaintechnologie noch eine der „festgezurrten“ Eigenschaften vor allem bei der Bitcoin-Blockchain (wo sie es auch bis heute noch ist), ist der Transparenzbegriff heute dehnbarer geworden. Ebenso ist die Anonymität, für welche bspw. die Bitcoin-Blockchain sorgt, in modernen angewandten Szenarien vor allem innerhalb von geschlossenen Organisationen nicht zweckerfüllend – die LINUX Foundation arbeitet im Rahmen Ihres Hyperledger-Projekts bspw. an einem Framework welches für eindeutige Identifizierungen von Personen sorgen soll.

Als letzte wichtige Eigenschaft von Niranjanamurthy, Nithya und Jagannatha 2018, die bei anderen Autoren nicht als zwingend vorausgesetzt wird, wird Open Source eingebracht. Fest steht, dass viele Projekte, die zu Beginn Ihres Lifecycles noch hinter verschlossenen Türen entwickelt wurden – bspw. stammt Hyperledger Fabric ursprünglich aus dem Hause IBM – basieren die meisten aktuellen Projekte und Anwendungen auf Open Source, sehr viele davon auf den Hyperledger Frameworks. Das ist zwar für alle Beteiligten von Vorteil, im Grunde genommen aus technologischer Sicht nicht zwingend erforderlich.

Der Autor setzt folgende als eindeutige Eigenschaften einer Blockchain fest (vgl. Meinel, Gayvoronskaya und Schnjakin 2018):

- (1) Dezentral
- (2) Autonom
- (3) Sicher

Weitere Eigenschaften wie Open Source, Anonymität und Transparenz sind zwar bei den meisten Blockchains vorhanden, aber im Grunde rein optionale Eigenschaften.

### 2.1.3. BEISPIELE FÜR BLOCKCHAIN-ANWENDUNGEN

Die populärste Blockchain-Anwendung ist vermutlich Bitcoin, der dahinter steckende Anwendungsfall einer Kryptowährung sicherlich der am häufigsten vorkommende. Hinter der Idee von Kryptowährung liegt das Verlangen nach einer autoritäts-unabhängigen Bezahlungsmethode. Die Autoritäten, von denen man sich hier loslösen wollte, sind in erster Linie Banken.

Prof. Dr. Norbert Pohlmann vom Institut für Internet-Sicherheit an der Westfälischen Hochschule Gelsenkirchen bestätigt der Blockchain-Technologie die Fähigkeit „in vielen Bereichen zentrale Instanzen, wie Banken, Notare oder Treuhänder, abzulösen“ (Pohlmann 2018).

Eine Krypto- oder Digitalwährung verwaltet Einheiten in Form von sog. „Tokens“. Diese Tokens werden, je nach Blockchain, unterschiedlich benannt und verwaltet. Die bekannteste und älteste Währung ist der Bitcoin. Im Grunde genommen ist er als Währung mehr oder weniger anerkannt, seit im Jahr 2010 zwei Pizzen in den USA für 10.000 Bitcoin-Einheiten den Besitzer wechselten. Die Nachfrage in den Folgejahren, grundsätzlich vergleichbar mit der eines Aktienmarktes wenn auch ungleich dynamischer, zeigt das rasante Wachstum und die damit steigende Popularität eindrucklich.

Besaßen die 10.000 Bitcoins im Jahr 2010 noch etwa 40 USD, so besitzen sie heute einen Wert von knapp 37 Millionen USD (Aktualisiert: 20.01.2019). Die spannende Frage ist, bleibt Blockchain ein Mittel zum Zweck, um Kryptowährung zu ermöglichen oder setzt sich die Technologie auch unabhängig dieses einen Anwendungsfalles durch?

Ein spannender Anwendungsfall wird aktuell mit medialer Aufmerksamkeit begleitet: Die US-Regierung untersucht seit 2015, ob die Anwendung von Blockchain-Technologie die Verwaltungskosten des Gesundheitsapparates senken kann. Die Kosten sind deshalb so hoch, da das klassische Gesundheitswesen eine Vielzahl von Vorgängen bei einer Vielzahl von Institutionen pro Patient meist aufwendig dokumentiert – und zwar meist jede Stelle für sich. Laut Forbes belaufen sich die Verwaltungskosten im US-amerikanischen Gesundheitswesen auf 380 USD pro Datensatz, d.h. pro Patient (Arnold 2018). Grund genug also, um mittels dezentraler Verteilung der Daten eine Vielzahl von Zugängen zum Patientendatensatz zu ermöglichen: Alle Beteiligten könnten sich direkt mittels Blockchain „verständigen“, bspw. könnte folgender Prozess direkt erfasst werden:

- Patient ist krank und geht zum Arzt
- Arzt verschreibt Medikament
- Patient geht zur Apotheke
- Apotheker prüft, ob Rezept valide
- Patient bezahlt
- Krankenkasse erfährt von den Kosten und erstattet die Auslage automatisch

Vor allem der Patient (natürlich auch die beteiligten Stellen, v.a. die Kasse) spart sich auf eindrückliche Art und Weise eine Menge Zeit – und Kosten. Dem Autor ist bewusst, dass diese Vorteile hinsichtlich der sensiblen Daten grundsätzlich auch Nachteile mit sich bringen können – deshalb sollte die Datensicherheit natürlich in jedem digitalen Szenario entsprechend berücksichtigt werden.

Diesem Anwendungsfall widmen sich auch Boulos et al. im International Journal of Health Geographics (Boulos et al. 2018) und nimmt dabei auch auf den Mehrwert einer räumlichen Komponente Bezug (s. Kapitel 2.1.5).

Das letzte Beispiel beschreibt die Anwendung einer Blockchain im Bereich Energie und Bezahlung. So arbeiten der Energiekonzern RWE und das ebenfalls deutsche Startup-Unternehmen slock.it an einem Bezahlmodell für das Laden der Akkus von Elektroautos, hier soll der Blockchain-Wert, also der Token, mit einem Gegenwert, in diesem Fall einer Energie-Einheit gekoppelt werden.

Das Projekt „Blockcharge“ plant, das Elektroautos künftig bereits ab Werk über ein integriertes Wallet (=Geldbörse) verfügen, damit der Ladevorgang an einer Elektroladesäule automatisch prozessiert werden kann. Nach der Vorstellung beider Unternehmen soll damit zum einen der Fahrer um den Anmelde- und Bezahlvorgang entlastet werden, zum anderen könnte diese Technologie auch ein international einheitliches Bezahlssystem schaffen, unabhängig von einzelnen Landeswährungen und dort vorhandenen Organisationen, welche Ladesäulen betreiben und wiederum unterschiedliche Arten der Bezahlung anbieten.

Zudem könnte diese Art der Bezahlung künftig auch dafür sorgen, dass Elektroautos bspw. an roten Ampeln auch für wenige Sekunden geladen werden, die entsprechende Hardware vorausgesetzt. Sog. Micropayments, die heute schon im Internet of Things Verwendung finden, sind für Blockchains einfach handhabbar (BitcoinBlog.de 2016).

#### *2.1.4. GEOBLOCKCHAIN*

Wie können räumliche Daten auf Blockchains verwendet werden? Die Frage ist grundlegend für die Anwendung der Technologie in der Geoinformatik. Der Autor unterscheidet in der Anwendung hierbei in zwei grundsätzliche Richtungen:

a) Speicherung räumlicher Daten als Attribute innerhalb der Blockchain:

Dies kann verglichen werden mit den Anfängen des Speicherns von einfachen Koordinatenfolgen in einer Datenbank. Bevor die Datenbankentwicklung eigene räumliche Erweiterungen bzw. spezielle Geometrieattribute hervorbrachte, waren die Geometrien einfache Zeichenfolgen innerhalb eines Datensatzes und wurden erst durch weitere Schritte, wie z.B. Geokodierung, innerhalb eines Geoinformationssystems dargestellt. Dieselbe Möglichkeit besteht heute bereits innerhalb einer Blockchain. Aus den Recherchen ging nicht hervor, dass eine der relevanten Blockchains bereits über ein fertiges eigenes standardisiertes, räumliches Datenmodell verfügt, wenngleich daran gearbeitet wird.

## b) Proof of Location Services

Hier setzt der sog. „Proof of Location“ (kurz: PoL) an – ein System zur Verifizierung von Standorten. Im Prinzip soll sichergestellt werden, dass in räumlichen Anwendungen verifizierte Standorte zur Verfügung stehen oder der eigene Standort mittels Validierung verifiziert wird. Ein bekanntes Protokoll ist das FOAM-Protokoll, welches im Kapitel 2.1.5 näher vorgestellt wird.

Des Weiteren gibt es natürlich die Möglichkeit, eine Blockchain nur zu einem bestimmten Zweck einzusetzen, welcher, betrachten wir den o.g. Fall a), überhaupt keine Geodaten an sich speichern können muss, sondern „nur“ die Richtigkeit und Aktualität eines Geodatensatzes verifizieren muss, der bspw. in einer separaten, u.U. auch zentralen Datenbank liegen kann. In diesem Spannungsfeld bewegt sich auch diese Arbeit.

### *2.1.5. BEISPIELE FÜR GEOBLOCKCHAIN-ANWENDUNGEN*

Ein auf PoL spezialisiertes Unternehmen ist FOAM, welches derzeit ein eigenes räumliches Protokoll (FOAM-Protokoll) auf Basis der Ethereum-Blockchain aufbaut. Dieses soll, im Gegensatz zum Global Positioning System (kurz: GPS) manipulationssicher und ausfallsicher sein, da gerade GPS z.B. in städtischer Dichte doch sehr anfällig bzgl. der Genauigkeit reagieren kann, da es auf den Empfang von mind. 4 zeitgestempelten Signaldaten von Satelliten angewiesen ist – und somit nur eingeschränkt zur Verfügung steht (Blockchainwelt 2018).

FOAM möchte erreichen, dass Nutzer eine Alternative zum bekannten GPS verwenden können und setzt hierbei auf eine „Low-Power Long Range Spread-Spectrum-Funktechnologie“ (kurz: LoRa). Mittels Mikrozahungen sollen Menschen dazu angeregt werden, eigene LoRa-Knotenpunkte einzurichten und zu betreiben, die dann verschlüsselte und fälschungssichere Standortinformationen für Anwender des Netzwerks bereitstellen. Langfristig könnten so, profitierend aus den Schwachstellen des GPS, ein eigenes System zur Verfügung gestellt werden, auf dessen Grundlage weitere Anwendungen wie z.B. Location Based Services (kurz: LBS) aufbauen könnten. Das entsprechende Whitepaper hierzu erschien im Januar 2018 (Foamspace Corp 2018).

Spannend ist auch die geobasierte Kryptowährung CEVEN (vormals CLEO) von der CBN Foundation, im Jahr 2018 auf den Markt gebracht. Hinter CEVEN und CBN steckt im Wesentlichen ein deutsches Startup aus München, die Firma Cloudeo. Bereits seit einigen Jahren betreibt die Firma einen Online Store für Geodaten und möchte damit Angebot und Nachfrage zusammenbringen. Auf der einen Seite steht der hohe Aufwand für die Erfassung räumlicher Daten, z.B. luftgestützt mittels Befliegungen, zum anderen der i.d.R. recht hohe Preis für diese Daten. Dieser hohe Preis bedingt, dass oftmals nur wenige Organisationen, die entsprechendes Interesse an großen Datenmengen und auch die dafür notwendigen finanziellen Mittel aufweisen, diese Daten auch tatsächlich beziehen können.

Der Cloudeo Store soll hier auch kleine Datenmengen zu erschwinglichen Preisen mittels Kryptowährung (hier: CEVEN) anbieten können, denn genau das ist in der Vergangenheit der Knackpunkt gewesen: Oftmals hat es sich aufgrund des hohen bürokratischen Aufwands (Angebot, Auftrag, Lieferung, Rechnungstellung, Buchung, etc.) einfach nicht gelohnt Kleinstmengen an Daten abzugeben. Genau das übernimmt der Store und die Kryptowährung: Sie senken den bürokratischen Aufwand auf ein Minimum. Im Moment ist

der Store fokussiert auf Rasterdaten, das soll sich aber laut Whitepaper in naher Zukunft weiterentwickeln (CBN Foundation 2018).

Denkbar wäre an dieser Stelle auch eine Kombination der genannten Beispiele: FOAM erfasst mittels PoL gesicherte Daten in ihrem eigenen Netzwerk und verkaufen die Daten später auch in Kleinstmengen mittel des Cloudeo Store.

Das nächste Beispiel stammt wieder aus dem Gesundheitswesen. Kamel Boulos beschreibt im International Journal of Health Geographics verschiedene Herausforderungen und Szenarien der Blockchain-Technologie inkl. Mehrwert einer räumlichen Dimension.

Im Detail diskutiert er, ob durch die persönlichen Daten im Datensatz des Patienten in Verbindung mit dem Internet of Things neue Angebote möglich wären, wie bspw. die Lieferung verschriebener Medikamente mittels Drohnen. Die Blockchain würde die gesamte Aufgabe der Validierung und Koordination übernehmen, der Patient benötigt lediglich ein Smartphone, mit dessen Hilfe die Position verifiziert und damit die Lieferung zugestellt werden kann. (Boulos et al. 2018)

In einem weiteren Anwendungsfall kommt er Crowd-Based-Information zu sprechen, also einem Szenario in dem jeder Besitzer eines Smartphones ein potenzieller Datengenerator ist. Die Blockchain könnte hier mittels Mikrozahlungen und einer geeigneten App den Anwender nach bestimmten Informationen fragen – wenn der Nutzer antwortet, erhält er dafür einen geringen Betrag, ausgezahlt mittels Kryptowährung. In einem Katastrophenfall könnte die App so bspw. Fotos der Umgebung einfordern, um schnell eine Übersicht über die Lage vor Ort zu bekommen. (Boulos et al. 2018)

Neben solchen Fällen, in dem die Daten vor allem für Nicht-Regierungs-Organisationen große Relevanz bedeuten können, ist die Möglichkeit Daten zu erheben natürlich auch für Unternehmen interessant – dadurch könnte die Plattform bzw. das Netzwerk finanziert werden.

Zuletzt kommt ein Beispiel, welches maßgeblich Idee und Entwicklung der Fragestellung dieser Thesis angetrieben hat. Die nationale Vermessungsbehörde in Schweden untersucht und entwickelt zusammen mit verschiedenen Unternehmen schon seit 2016 den Einsatz von Blockchaintechnologie (Rizzo 2016). Gemeinsam mit dem Unternehmen Chromaway wird versucht, die menschliche Fehlerquote beim Transfer von Daten sowie die allgemeine Sicherheit und Fehleranfälligkeit der Daten zu erhöhen bzw. zu minimieren. Beratend ist zudem noch die Firma Kairos Future involviert, die im März 2017 einen anschaulichen Bericht verfasst haben, in dem die Möglichkeiten eindrucksvoll dargestellt werden. So ist beispielsweise ein Ansatz im Projekt, das ein Eigentümer sein Flurstück direkt an eine andere Person verkaufen kann. Mittels intelligenter Routinen innerhalb der Blockchain sollen alle notwendigen Prüfungen (z.B. Bonität & Berechtigung) direkt ausgeführt werden, ohne auf eine dritte Instanz angewiesen zu sein (KAIROS Future et al. 2017).

## 2.2. FUNKTIONSWEISE & AUFBAU

An dieser Stelle möchte der Autor auf hilfreiche und z.T. offene Onlinekurse der gemeinnützigen LINUX-Foundation hinweisen, die bei der Einarbeitung ins Thema äußerst hilfreich sind, da die Inhalte qualitativ hochwertig aufbereitet sind (EDX 2018a & EDX 2018b).

Die Beschreibung der Funktionsweise & der Aufbau einer Blockchain sind im Grunde ähnlich, aber je nach Produkt gibt es mehr oder weniger relevante Unterschiede. In dieser Arbeit orientiert sich die gesamte Dokumentation am Hyperledger Framework der LINUX Foundation, welche im späteren Verlauf noch vorgestellt wird. Gerade die einzelnen Komponenten, wie z.B. die eingesetzten Smart Contracts, können sich im Vergleich zu anderen Frameworks doch erheblich unterscheiden. Die Beschreibung von Funktionsweise und Aufbau erhebt daher keinen Anspruch auf Vollständigkeit, jedoch einen hohen Anspruch an Verständlichkeit dem Leser gegenüber.

Bevor jedoch die einzelnen Komponenten und deren Funktionsweise im Detail erklärt werden, wird im Folgenden eine beispielhafte Abhandlung einer Blockchain-Transaktion beschrieben – am Beispiel eines fiktiven, hier nicht näher definierten „Wertes“.

- Im Grunde genommen ist jeder Lese- oder Schreibzugriff auf einen Ledger (= Konto, Kassenbuch) bzw. an die Kette zunächst eine Art Anfrage und kann z.B. eine Art Überweisung sein.
- Diese Anfrage kommt, je nach eingesetztem Framework, i.d.R. von einer Eingabemaske bzw. Applikation, die mittels eines Software-Development-Kits (SDK) diese Anfrage an einen Peer (= Knoten im Netzwerk) weiterleitet.
- Der Peer kommuniziert die Anfrage weiter an den Chaincode, daher den eingesetzten Code, der nun zu entscheiden hat ob bspw. die Anfrage berechtigt ist oder nicht und falls ja, was im Anschluss passieren soll (=Aktion). Der Chaincode prozessiert also die Anfrage.
- Falls keine Einwände bestehen, wird der Chaincode die Anfrage bestätigen und via SDK zurück zur Applikation schicken. Die Applikation wird, wieder mittels SDK, die Anfrage an den sog. Ordering-Service schicken.
- Dieser fasst viele einzelne Anfragen aus einem Netzwerk in einem Block zusammen, welcher dann an eine Vielzahl von Peers im Netzwerk gesendet wird.
- Die Peers werden dann, jeder für sich, diesen Block validieren und an die Kette hängen bzw. in das „Kassenbuch“ schreiben. Hierzu müssen i.d.R. mind. 51% der Peers zustimmen.
- Im Anschluss gilt die Transaktion des Wertes als erledigt bzw. abgewickelt und ist nun einer fester Bestandteil des Ledgers bzw. der Kette.
- Der Gesamtzustand der Kette ist der sog. „World-State“ also der Gesamtzustand oder Status, gerne auch bezeichnet als Kontostand

Hierbei wird deutlich, warum Blockchain-Technologie auch als „Internet der Werte“ bzw. im Englischen „Internet of Value“ (kurz: IoV) bezeichnet wird (siehe auch: Meinel, Gayvoronskaya und Schnjakin 2018). Einen Überblick über die Funktionsweise einer Blockchain liefert der Merkle-Tree in Abbildung 1. Der Merkle-Tree, auch "binary hash tree" genannt, ist die Grundlage der Datenstruktur, um auch große Datenmengen effizient in der Blockchain zu speichern. Die Merkle-Trees werden dazu verwendet, die Transaktionen in Blöcken zu organisieren und zeitgleich den digitalen Fingerabdruck aller Blöcke und Transaktionen zu generieren und zu verwalten. Dadurch ist ein effizienter Zugriff gewährleistet, selbst wenn einzelne Transaktionen innerhalb der Blockchain verifiziert werden müssen.

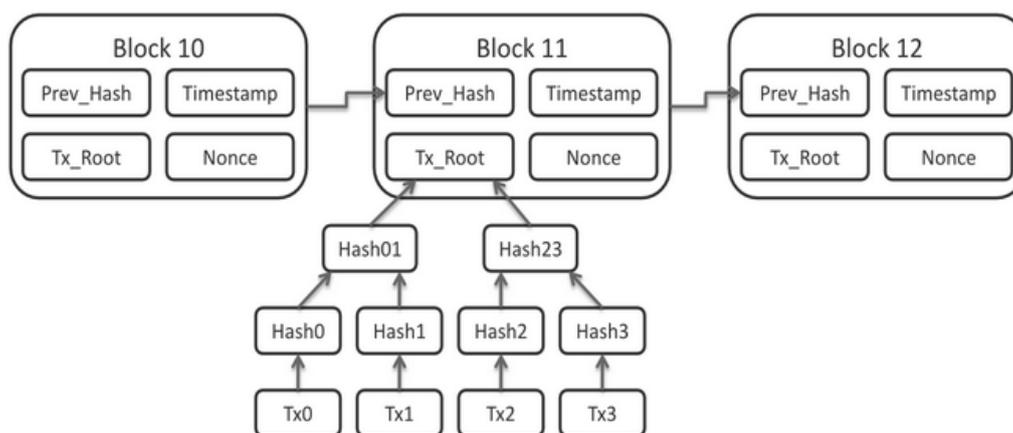


Abbildung 1: Merkle-Tree (EDX 2018a)

Die Abbildung zeigt, dass einzelne Transaktionen (kurz: Tx), z.B. Tx0, Tx1, usw. jeweils mit einem eigenen Hash (Hash0, Hash1, ...) belegt werden. Diese Hashes werden ihrerseits zusammengefügt (z.B. Hash01), wodurch mehrere Transaktionen zusammengefasst werden. Eine definierte Menge Transaktionen wird dann in einem Block zusammengefasst, welcher die Gesamtheit aller Transaktionen mittels eines Hashes verifiziert (Tx\_Root) und zudem den Hash des vorangegangenen Blocks beinhaltet (Block 11 verweist mit Prev\_Hash auf Block 10).

Anhand des Merkle-Trees kann auch die Sicherheit einer Blockchain erläutert werden (s. Abbildung 2). Wird versucht auch nur eine Transaktion oder einen Datensatz zu manipulieren, wirkt sich das direkt auf die gesamte Kette aus (Hier alles ab Block 11). Die Transaktion wird für ungültig erklärt und nicht zugelassen.

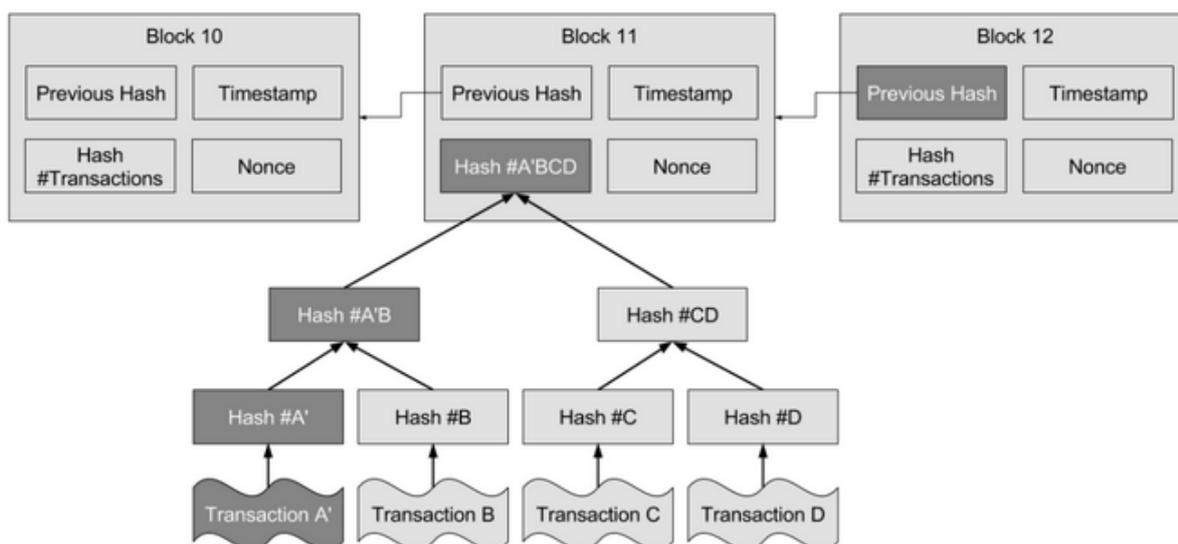


Abbildung 2: Auswirkung manipulierter Transaktionen anhand Merkle-Trees (EDX 2018a)

Doch um diese Werte vertrauenswürdig und sicher zu verwalten, benötigt es einiger weiterer Komponenten, welche erst in ihrer Kombination zu einer funktionierenden Blockchain im Sinne der eigentlichen Definition beitragen (vgl. EDX 2018b) und die Tätigkeiten im Hintergrund ausführen, damit auch die Blockchain an sich, wie hier als Merkle-Tree

dargestellt, funktionieren kann. Hierzu werden die Komponenten im Folgenden detailliert vorgestellt.

### *2.2.1. LEDGER & WORLD-STATE*

Ledger bedeutet Konto, Hauptbuch oder Wirtschaftsbuch. Dieser Begriff wird häufig auch synonym zum Begriff Kette (=Chain) eingesetzt. Unabhängig vom Einsatz der Begrifflichkeit ist der Ledger das Zentrum der Blockchain, nämlich eine verteilte und unveränderbare Aufzeichnung aller durchgeführten Transaktionen im Netzwerk. Der Gesamtzustand aller Transaktionen bildet den World-State, also den Gesamtzustand oder auch Kontostand der Blockchain. Das Blockchain funktioniert, wenn alle diesen Zustand kennen und ihm somit auch vertrauen.

Der Ledger besteht also aus der versionierten Liste aller Blöcke sowie der einzelnen Transaktionen, welche die Blöcke erzeugen. Ein Ledger kann mit einem klassischen Kassenbuch verglichen. Wie auch ein Kassenbuch kann ein Ledger eine bestimmte Verwaltungsaufgabe besitzen, also zum Beispiel alle Verkäufe, Einkäufe oder Überweisungen dokumentieren.

Am Beispiel des digitalen Ledgers in Hyperledger Fabric (vgl. LFD271 2018) kann ein der Ablauf verdeutlicht werden: Jede Aktion wird dokumentiert (= geloggt) – diese Aktivitäten sind Transaktionen und die Gesamtheit aller Transaktionen bildet den Transaction-Log. Eine Transaktion ist i.d.R. die kleinste Einheit innerhalb einer Blockchain.

Neben dem Transaction Log werden im World State die zu den Transaktionen zugehörigen IDs und Werte geführt und versioniert. Jede Änderung wird ausschließlich hinzugefügt (Append-Only). Es werden keine Einträge gelöscht. Dieses Vorgehen ist das Grundwesen einer Blockchain, da jeder Zustand (World State) zu jedem Zeitpunkt nachvollzogen werden kann. Aus diesem Grund wird häufig auch von einem „Immutable Ledger“ gesprochen, also einem unveränderlichen Hauptbuch.

Auch bei Hyperledger Fabric wird der Ledger auf die Knoten (Peers) im Netzwerk verteilt. Alle Knoten haben die aktuelle Version des Ledgers und führen ihn fort. An dieser Stelle wird bereits deutlich, dass diese Verteilung sicherheitstechnisch einen großen Vorteil bieten kann, da sie quasi fälschungssicher ist. Dieser Aspekt wird im Abschnitt Verteilte Netzwerke im folgenden Kapitel näher beschrieben.

### *2.2.2. PEER NETWORK*

Das Peer Network speichert, aktualisiert und verwaltet den Ledger. Ein Peer repräsentiert einen Teilnehmer des (Computer-) Netzwerks. Bei einem Blockchainnetzwerk handelt es sich um ein dezentrales Netzwerk, für ein Verständnis lohnt es sich, Grundlagenarbeit zu Kommunikationsnetzwerken zu betrachten. Nach Baran, Boehm & Smith (1964) kategorisieren und vergleichen drei Arten von Netzwerken:

#### **Zentrale Netzwerke**

Zentrale Netzwerke werden gerne, wie in Abbildung 3 (links) dargestellt, als sternförmige Client-Server Architektur dargestellt. Zentral steht der Server, der mit allen mit allen Knoten verbunden ist. Die Knoten wiederum senden alle Daten an den Server, also den zentralen

Knoten im Netzwerk, welcher wieder Daten prozessieren und neue Daten oder Antworten an den Knoten zurück- oder an andere Knoten weitergeben kann.

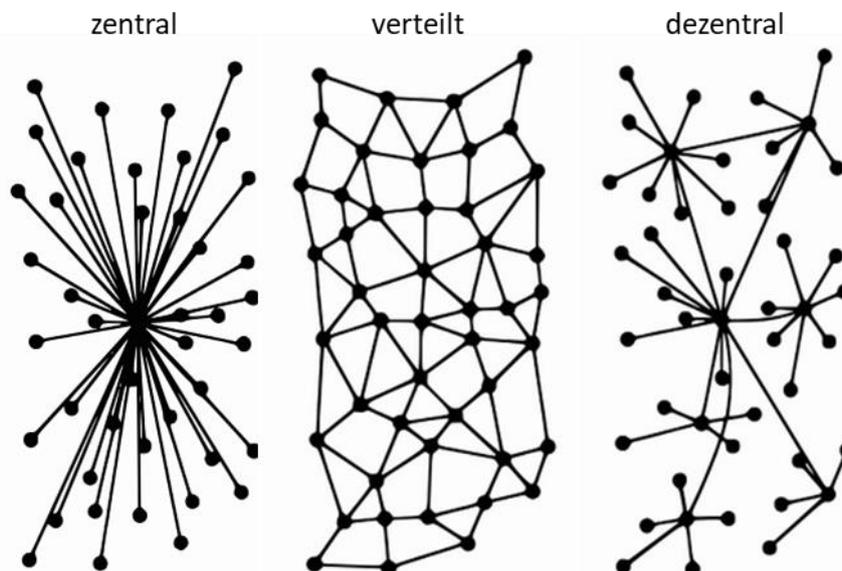


Abbildung 3: Übersicht Netzwerkartenn (verändert nach Baran, Boehm & Smith 1964)

Zentrale Netzwerke haben den großen Vorteil, dass Sie einfach gewartet werden können. Im Prinzip muss nur der zentrale Knoten/Server kontrolliert werden. Auf der anderen Seite birgt das gleichzeitig ein entsprechend hohes Risiko: Fällt der Server aus, stehen die sternförmig verteilten Knoten ohne zentrale Anlaufstelle da. Das führt zu einem Komplettausfall des Netzwerks bzw. des Systems. Das ganze Vertrauen liegt auf einem Knoten – was auch bedeutet, dass i.d.R. nicht einsehbar ist, was innerhalb dieses zentralen Knotens passiert. Zentrale Netzwerke sind meist nicht dazu ausgelegt, besonders transparent zu sein. Die Abhängigkeit ist also dementsprechend hoch.

### Verteilte Netzwerke

In einem verteilten Netzwerk gibt es diesen zentralen Knoten nicht mehr (s. Abbildung 3 Mitte). Jeder Knoten bzw. Peer kann sich mit anderen Knoten verbinden sowie Daten vermitteln oder über verschiedene andere Knoten hinweg prozessieren.

Verteilte (auch: vermaschte) Netzwerke haben den Vorteil einer hohen Belastbarkeit und sind preisgünstig. Im Gegensatz zu einem zentralen Netzwerk, bei dem der zentrale Knoten die gesamte Belastung des Netzwerks aushalten und entsprechend leistungsfähig sein muss, werden in einem verteilten Netzwerk keine hohen Anforderungen an die Knoten gestellt, da auch die Masse aus „schwachen“ Knoten ein belastbares Netzwerk erzeugt. Ein weiterer Vorteil ist die räumliche Ausbreitung: Das Netzwerk kann beliebig wachsen, indem einfach weitere Knoten hinzugefügt werden.

Allerdings ist die Stärke auch schwache Knoten einbinden zu können u.U. die größte Anfälligkeit eines verteilten Netzwerks: Sollten die Mehrheit der Knoten nicht stark genug sein, kann es – gerade im Bezug auf Blockchains – doch einen negativen Einfluss auf die Prozessierungsdauer haben. Bei Blockchains könnte somit die Dauer ansteigen, um neue Blöcke zu erzeugen und an die Kette zu hängen.

Der wohl größte Vorteil ist hierbei die Ausfallsicherheit. Einzelne Knoten im Netzwerk können ohne Probleme ausgeglichen werden. Im Gegensatz zur Client-Server Struktur wird hier dem ganzen Netzwerk „vertraut“.

### Dezentrale Netzwerke

Eine Kombination von aus zentralen und verteilten Netzwerken wird als dezentrales Netzwerk bezeichnet. Ein solches Netzwerk ist definitiv nicht zentral, da es nicht nur einem zentralen Server vertraut, aber nicht komplett verteilt, was die angedeuteten Sub-Netzwerke in Abbildung 3 darstellen sollen, die ihrerseits eher zentralisiert ausgerichtet sind.

Dezentrale Netzwerke können, abhängig vom Anwendungsfall, die Stärken beider erstgenannten Netzwerke kombinieren, indem sie bspw. die Sicherheit eines verteilten Netzwerks verwenden, für einzelne Aufgaben oder Teile aber die eher zentral ausgerichtete Struktur eines sternförmigen Client-Server Netzwerkes verwenden.

Diese Kombination birgt aber auch ihre eigenen Risiken: Das Netzwerk kann zwar insgesamt wesentlich effizienter handeln als ein Verteiltes, einzelne Teile des Netzwerks wiederum besitzen eine höhere Ausfallwahrscheinlichkeit.

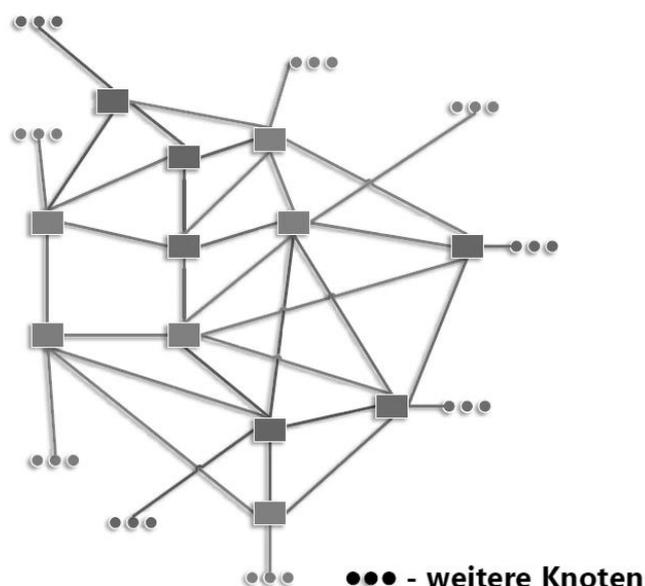


Abbildung 4: Peer(-to-Peer)-Netzwerk (Meinel et al. 2018)

Die Blockchaintechnologie an sich besteht i.d.R. aus einem verteilten Netzwerk mit unzähligen Peers, daher wird meist auch von Peer-Netzwerken gesprochen, wie verteilte Netzwerke auch bei Meinel bezeichnet werden (Meinel et al. 2018). Abbildung 4 zeigt ein solches Peer-Netzwerk, wobei die unterschiedlichen Farben schon andeuten, dass unterschiedliche Bereiche oder Verantwortlichkeiten bestehen können. Mittels entsprechender Konfigurationen oder Integration von zentralen Netzwerken kann auch ein verteiltes Netzwerk schnell Teil eines dezentralen Netzwerks sein.

Der Autor hat bei den Recherchen festgestellt, dass die Grenzen in nicht-wissenschaftlicher Literatur hierbei oft verwischen und der Begriff „verteilt“ schnell mit dem Begriff „dezentral“ gleichgesetzt oder vertauscht wird.

### 2.2.3. MEMBERSHIP SERVICES & MEMBERSHIP SERVICE PROVIDER

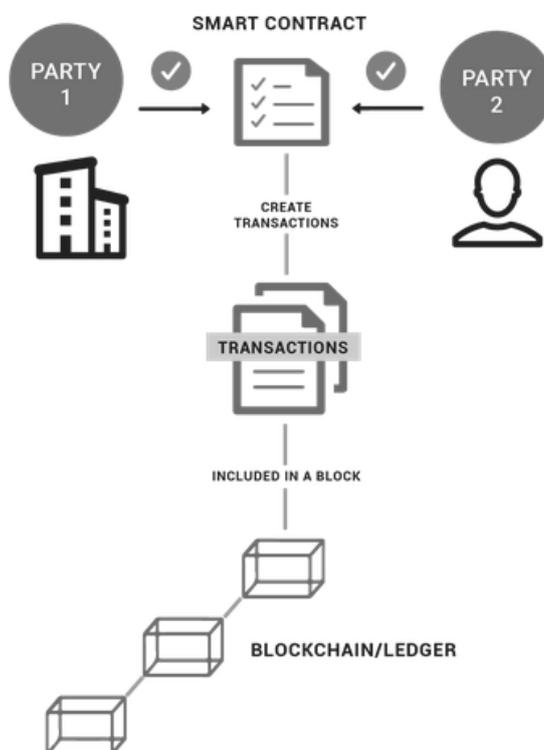
Der Membership-Service verwaltet die Authentifizierung bei privaten Blockchains. Dort wird das Rechtekonzept verwaltet und den Nutzern oder Nutzergruppen zugeordnet. Private Blockchains verlangen bekannte Identitäten. Zum Vergleich: Die Bitcoin-Blockchain ist eine öffentliche Blockchain, an der jeder partizipieren und Transaktionen durchführen kann. Für die meisten Projekte ist eine offene (public) Blockchain nicht sinnvoll, z.B. wenn es um Organisationsinterne Blockchains geht. Die Unterschiede werden im Kapitel 2.3 detailliert erörtert.

Zum Membership-Service gehört der Membership-Service-Provider (MSP), eine Schnittstelle welche die Anmeldung und Zugriffsrechte auf die Blockchain verwaltet. Der MSP kann in externe Anwendungen integriert werden, um den Zugriff auch von Drittprogrammen zu ermöglichen.

### 2.2.4. SMART CONTRACTS / CHAINCODE

Smart Contracts (dt. Intelligente Verträge) werden auch als Chaincode bezeichnet und laufen „on top“ auf der Blockchain. Mittels Smart Contracts können bestimmte Aktionen in Abhängigkeit von bestimmten Ereignissen gesteuert werden. Ein gutes Beispiel wäre die Auszahlung einer Summe X an einen Versicherungsnehmer einer Hochwasserschutzversicherung wenn das eigene Flurstück unter Wasser steht. Im Prinzip wird in diesem Beispiel eine einfache Wenn-Dann-Funktion vorgestellt.

**BLOCKCHAIN AND SMART CONTRACTS - FLOW DIAGRAM**



**Abbildung 5: Smart Contracts Flussdiagramm (EDX 2018a)**

Die Contracts laufen ebenfalls auf den Peers im Netzwerk verteilt und entsprechende Aktionen, wie z.B. die o.g. Auszahlung, werden ebenfalls als neue Transaktion an den

Ledger angehängt. Smart Contracts zielen im Wesentlichen darauf ab, bestimmte Aufgaben effizienter auszuführen und gleichzeitig eindeutig zu protokollieren.

Im Hyperledger Fabric Framework entspricht der Chaincode den „Smart Contracts“. Dieser wird auf den Peers ausgeführt und verarbeitet die Transaktionen. Der Chaincode erlaubt es den Teilnehmern im Netzwerk somit, Transaktionen auf dem Ledger einzubringen und somit den World State aktualisieren zu lassen.

Der Chaincode in Hyperledger ist in Go geschrieben und programmierbar und sog. Channel erreichbar. Entwickler können mit ihm Geschäftsverträge, Besitzdefinitionen und gemeinschaftlich betriebene dezentrale Applikationen erzeugen. Der Chaincode ist hier die Tür zur Blockchain.

Der Code verwaltet somit den Zustand des Ledgers, welcher durch die aufgerufenen Transaktionen aus den Apps permanent fortgeführt wird. Auch wird Besitz, wie bspw. Bei Bitcoin, durch bestimmten Code zugewiesen und verwaltet. Dieser kann nicht durch weiteren Chaincode manipuliert werden.

Damit die Applikationen mit den Chaincode kommunizieren können muss dieser auf jedem Peer installiert sein, der Transaktionen bestätigt ( → Konsensfindung).

Die verfügbaren Frameworks und Systeme gehen unterschiedlich mit Chaincode um. Hyperledger sieht zwei Wege für Chaincode bzw. Smart Contracts vor:

- ⊖ Individuelle Contracts entwickeln, welche auf unabhängigen Instanzen betrieben werden
- ⊖ Dezentrale Applikationen erzeugen, die einen oder mehrere Code-Instanzen verwalten und zugänglich machen. Die Nutzer greifen damit über Applikationen auf die Chaincode-Instanzen zu (effizienter)

Hyperledger Fabric bringt eigene Interfaces mit, die beim Entwickeln des Chaincode verwendet werden können: „ChaincodeStub“ and „ChaincodeStubInterface“. Mit ChaincodeStub kann über bereitgestellte Funktionen mit dem zugrunde liegenden Ledger kommuniziert werden, z.B. um Abfragen, Updates oder Löschungen zu erzeugen. Die API beinhaltet:

- ⊖ `func (stub *ChaincodeStub) GetState(key string) ([]byte, error)`  
Gibt den Wert des Schlüssels vom Ledger zurück. **GetState** liest hierbei aber (noch) nicht die Datensätze aus, die erst noch an den Ledger überwiesen werden müssen, z.B. durch **PutState**. Wenn der Schlüssel noch nicht vorhanden ist, gibt die Applikation den Error zurück (nil, nil)
- ⊖ `func (stub *ChaincodeStub) PutState(key string, value []byte) error`  
Fügt den speziellen Schlüssel und den Wert in den zu schreibenden Datensatz als Proposal ein. **PutState** agiert aber nicht mit dem Ledger, bis über die Konsensfindung eine Transaktion erfolgreich abgeschlossen wurde.
- ⊖ `func (stub *ChaincodeStub) DelState(key string) error`  
Dokumentiert den Schlüssel, welcher über die Transaktion gelöscht werden soll. Der Schlüssel und der zugehörige Wert werden dann von Ledger entfernt wenn die Transaktion erfolgreich hinzugefügt wurde.

Zwei Methoden müssen hierbei beachtet werden:

- ➔ **Init**  
Wird dann aufgerufen, wenn mittels Chaincode eine Transaktion eingebracht wird, die entweder einen neuen Datensatz erzeugen oder einen bestehenden aktualisieren soll
- ➔ **Invoke**  
Wird dann aufgerufen, wenn die aufgerufene Transaktion zur Prozessierung freigegeben wurde

Bei der Entwicklung von Chaincode müssen beide Methoden eingebaut werden

### 2.2.5. TRANSAKTIONEN

Transaktionen sind i.d.R. die Ergebnisse abgeschlossener Verträge zwischen einzelnen Teilnehmern im Netzwerk. Der Verlauf von Transaktionen wird im Ledger gespeichert, die Gesamtheit aller Transaktionen bildet den World State. Bei einer abgeschlossenen Transaktion werden alle Peers im Netzwerk benachrichtigt, sodass das gesamte Netzwerk aktuell gehalten wird.

### 2.2.6. CONSENSUS-ALGORITHMEN

Tabelle 1: Übersicht der wichtigsten Konsens-Algorithmen

Kurz	Bezeichnung	Anwendungsfälle
<b>PoW</b>	Proof of Work	Ethereum & Bitcoin
<b>PoS</b>	Proof of Stake	Ethereum
<b>PoET</b>	Proof of Elapsed Time	HL Sawtooth
<b>BFT</b>	Byzantine Fault Tolerance	Stellar, Ripple
<b>SBFT</b>	Simplified Byzantine Fault Tolerance	HL Fabric
<b>PoA</b>	Proof of Authority	HL Iroha
<b>YAC</b>	Yet-Another-Consensus	HL Iroha

## 2.3. ÖFFENTLICHE VS. PRIVATE BLOCKCHAIN

Blockchains werden in unterschiedlichen Ausprägungen verwendet. Wichtig ist vor allem, dass für unterschiedliche Szenarien und Anwendungsfälle die passende „Art“ der Blockchain gefunden bzw. definiert wird. Die meisten Blockchain-Technologien können zwar grundsätzlich meist für alle Ausprägungen verwendet werden, allerdings gibt es entsprechende Spezialisierungen. Folgende Ausprägungen von Blockchains werden in der Literatur diskutiert:

- (1) Öffentliche Blockchains
- (2) Private Blockchains

Die Bitcoin-Blockchain ist bspw. eine öffentliche und zulassungsfreie Blockchain. Das bedeutet, jeder kann beitreten, die Transaktionen einsehen oder gar eigene Transaktionen tätigen. Ein Unternehmen hat vielleicht Interesse an einer Blockchain, um eine Supply Chain abzubilden, an der auch Dritte beteiligt sind. In diesem Szenario möchte das Unternehmen

sicher nicht, dass jeder teilnehmen kann, würde sich also für eine zulassungsbeschränkte Blockchain entscheiden, die zudem in einem geschlossenen Netzwerk läuft, also eine private Lösung darstellt. Dadurch können alle Vorteile der Blockchain innerhalb eines geschlossenen Netzwerks mit unterschiedlichen Berechtigungen für jeden Partner genutzt werden, ohne das Risiko einzugehen, dass Interna in unbefugte Hände gelangen.

Die Diskussion um die Vor- und Nachteile privater und öffentlicher Blockchains prägen auch Heumüller und Richter (Heumüller & Richter 2018). In dieser Frage geht es meist um die Frage nach der Zulassungsbeschränkung an sich und die Frage wie das Vertrauen verteilt bzw. wem gegenüber vertraut wird. Bei einer öffentlichen und zulassungsfreien Blockchain wie Bitcoin wird das Vertrauen weder Institutionen (z.B. Banken) noch Menschen gegenüber (z.B. Banker) entgegen gebracht, sondern der Technologie an sich. Heumüller und Richter schreiben, dass die Blockchain unter diesen Umständen selbst zur Institution werden kann.

Ganz gegensätzlich werden in einem privaten, z.B. einem Unternehmensnetzwerk die Teilnehmer ausgewählt und geprüft. Nur wem „Vertrauen“ gebührt, „darf“ zugelassen werden, was dann wiederum durch einen sog. Membership-Service gesteuert werden kann (s. Kapitel 2.2.3). Private Blockchains eignen sich besonders gut für B2B-Anwendungen.

Die Auswahl ob öffentlich oder privat hängt also ausschließlich vom Use Case ab. Die meisten Blockchains haben durch Ihre oft modulare Bauweise meist ein entsprechendes Modul zur Umsetzung der Zugriffskontrolle parat.

## 2.4. ABGRENZUNG ZU ZENTRALER DATENHALTUNG

Wichtig ist die zugrundeliegende Datenhaltung innerhalb einer Blockchain. Im Gegensatz zu klassischen Datenbanksystemen liegen die Daten nämlich nicht „an einem Platz“ „in einer Datenbank“ sondern sind digital und räumlich auf die Peers verteilt. Der Vergleich vom Client-Server-Ansatz und Peer-to-Peer-Netzwerken in Abbildung 6 macht den Unterschied deutlich.

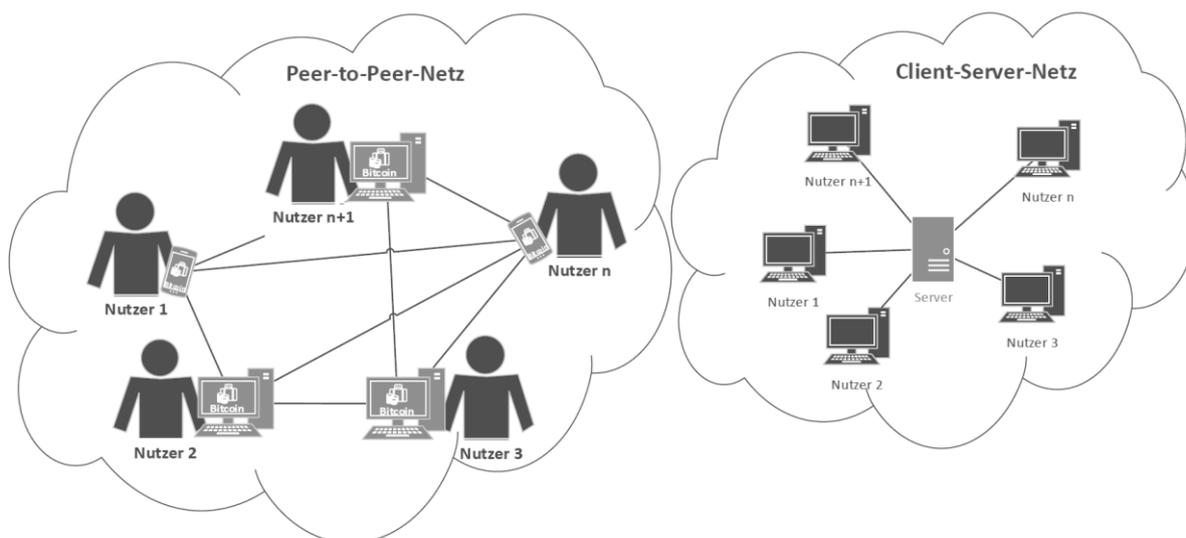


Abbildung 6: Peer-to-Peer vs. Client-Server-Netzwerk (Meinel et al. 2018)

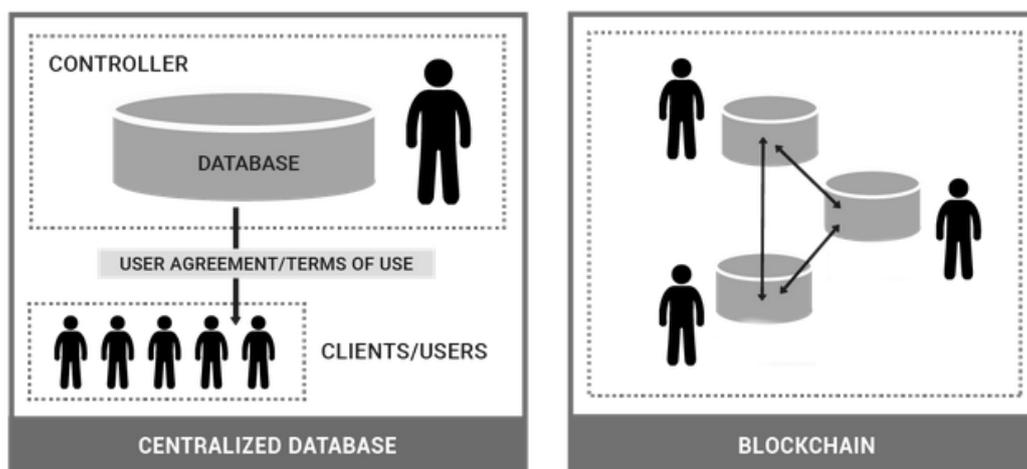
Im Wesentlichen geht es um dieselbe Diskussion, die auch Baran, Boehm & Smith bereits in den 1960er Jahren führten (vgl. Kapitel 2.2.2) und damit die Eigenschaften, die eine

Blockchain von einer klassischen relationalen Datenbank unterscheidet (Baran, Boehm & Smith 1964).

Tabelle 2 zeigt wesentliche in dieser Arbeit diskutierte Unterschiede auf. Vor allem geht der Autor auf die technischen Möglichkeiten ein: Relationale Daten verfügen natürlich über die Möglichkeit, Daten zu erzeugen, zu ändern oder auch zu löschen. Blockchains hingegen im Sinne eines Anhängens im Grunde „nur“ neue Daten erzeugen, die aber im Prinzip eine neue Version der alten Daten sind (Append-Only). Die Blockchain macht sich durch diese Eigenschaft im Grunde fälschungssicher.

**Tabelle 2: Unterschiede zwischen Datenbanken und Blockchain**

	Relationale Datenbanken	Blockchain
<b>Techn. Möglichkeiten</b>	Append, delete, change, etc.	append-only
<b>Veränderbar, z.B. Business-Regeln</b>	Ja, jederzeit	Nein, nach Einführung niemals
<b>Ausfallsicherheit</b>	Gering, da oft zentral bzw. nur ein zentraler Knoten	Hoch, da einzelne Peers ohne Einschränkungen ausfallen können



**Abbildung 7: Zentrale Datenbanken vs. Blockchain (EDX 2018b)**

Auch Chiu & Koepl greift einen wichtigen Mehrwert der Blockchain auf, nämlich die nicht vorhandene Notwendigkeit einer dritten Instanz, welche dafür verantwortlich ist, dass alle Transaktionen sauber laufen. Sie vergleicht zentralisierte Systeme wie PayPal, bei dem es notwendig ist, das Person A welche Person B Geld überweisen möchte, PayPal vertraut, dass alles ordentlich abläuft (Chiu & Koepl 2018).

Auch die Kurse der LINUX Foundation helfen, die Unterschiede herauszuarbeiten (EDX 2018b): Eine Blockchain ist eine reine „write-only“ Struktur, bei dem neue Einträge ans Ende des Ledgers angehängt werden. In einer klassischen Datenbank können bestehende Einträge auch verändert werden (vgl. Tabelle 2).

Bei einer bestimmten Menge Einträge werden neue Blöcke gebildet und mit einem neuen Hash versehen. Zum neuen Hash wird der Hash des vorherigen Blocks gelinkt, sodass eine eindeutige Reihenfolge entsteht, die später nicht mehr geändert werden kann. Nicht mal

Administratoren könnten diese Reihenfolge ändern. In klassischen Relationen Datenbanken können Administratoren i.d.R. frei schalten und walten.

Beide Systeme haben ihre Vor- und Nachteile und spezifischen Anwendungsfälle. Eine Diskussion Datenbank vs. Blockchain ist aus Sicht des Autors daher nicht zielführend. Vielmehr muss der Frage nachgegangen werden, wie sich die Technologien ergänzen können.

## 2.5. ENTWICKLUNG & STAND DER TECHNIK

Der eigentliche Ursprung der Idee zur Blockchain lag in dem Anliegen eine Möglichkeit zu schaffen, finanzielle Transaktionen ohne einen Dritten, also z.B. eine Bank, abzuwickeln (Meinel, Gayvoronskaya und Schnjakin 2018, Nakamoto 2008). Bitcoin sollte genau das ermöglichen, wie Abbildung 8 zeigt.

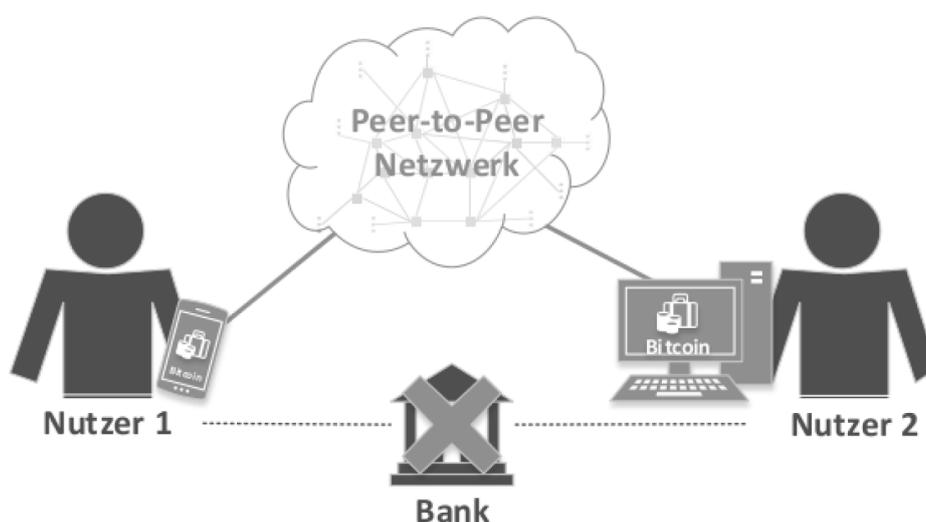


Abbildung 8: Das Bitcoin-Prinzip (Meinel et al. 2018)

Die Idee des dezentralen Zahlungsnetzwerks an sich entstand allerdings schon vor Bitcoin, wenn auch kein Versuchsaufbau glückte, da es entweder Probleme mit der Sicherheit oder mit dem Konzept im Allgemeinen gab (Meinel, Gayvoronskaya und Schnjakin 2018). Man erkennt, dass Bitcoin auf schon bekannten und etablierten Einzeltechnologien aufbaut und diese erfolgreich miteinander verknüpft. Einzelkomponenten, die heute in der Blockchaintechnologie eingebunden werden, sind bspw. Peer2Peer-Netzwerke oder die verschiedene Methoden zur Kryptographie.

Seit der ersten populären Anwendung Bitcoin hat sich viel entwickelt. Bitcoin selbst war laut Coin Market Cap im Oktober 2017 schon bei einer Marktkapitalisierung von 100 Milliarden USD. Im Schatten dieser Entwicklung haben sich bereits über 1000 Start-Up-Unternehmen im Ökosystem der Kryptowährung gegründet, die überwiegend auch Anwendung auf der Bitcoin-Blockchain anbieten, sich oftmals aber auch mit der Entwicklung privater Blockchains beschäftigen, bspw. im Hyperledger Umfeld.

Natürlich besteht die entsprechende Nachfrage, man muss nur einen Blick auf die an Hyperledger beteiligten Unternehmen werfen oder Blockchain-relevante Medien studieren: Neben Unternehmen stürzen sich Verwaltungen, Universitäten, Banken und auch Versicherungen auf die Entwicklung, Forschung und Nutzung von Blockchaintechnologien.

Gupta beschreibt in der Harvard Business Review die Entwicklungen der letzten 10 Jahre auch unter der Frage: In welchem Maße wird sich Blockchain etablieren? Neben der Nennung von bekannten Ledgern wie Bitcoin oder Ethereum landet auch Gupta in ihrem Artikel bei der Entwicklung von Geokryptowährung wie dem CLEO-Token oder geo-basierten Smart-Contracts (vgl. Gupta 2017 und CBN Foundation 2018).

Heute gibt es vielseitige Möglichkeiten, wie auch Pohlmann beschreibt: Neben dem Angebot von OpenSource-Systemen gibt es eine Reihe von Services, hier Blockchain-as-a-Service (BaaS). Anbieter wie Microsoft bieten auf Basis von MS Azure mit ihrem Produkt „Bletchley“ eine fertige Lösung, die sofort konfiguriert und verwendet werden kann. Auch IBM verfügt mit „Bluemix“ ein BaaS-Angebot (Pohlmann 2018).

Bei der oft diskutierten Frage nach der Marktreife und der grundlegend zurückhaltenden Meinung konservativer Unternehmen oder Branchen gegenüber neuen Technologien kann hier seit der Einführung eines eigenen ISO-Komitees im Jahr 2016 entgegengewirkt werden. Mit Gründung des ISO/TC 307 wurde der Grundstein für die fortlaufende künftige Standardisierung gelegt (ISO 2016). Das Komitee arbeitet an Standards wie der ISO/NP TS 23635 die Richtlinien für die Verwaltungen definieren soll. Damit sollten auch die letzten Zweifler verstanden haben, dass es sich bei Distributed Ledger Technologien um keine Eintagsfliege handelt.

## 2.6. CHANCEN & RISIKEN DER TECHNOLOGIE

Das letzte theoretische Kapitel soll einen kurzen Überblick über die Chancen und Risiken der Technologie erläutern, hier ohne konkreten Bezug auf räumliche Anwendung. Die adaptierten Chancen und Risiken werden im Schlussteil der Arbeit im Kapitel 6.2 berücksichtigt.

Das Fraunhofer Institute For Applied Information Technologie (FIT) hat sich eingehend mit dem Thema Blockchain und den Vor- und Nachteilen der Technologie beschäftigt (Schlatt et al. 2016). Tabelle 3 zeigt die zusammengetragenen Argumente aus dem Whitepaper.

**Tabelle 3: Chancen und Risiken von Blockchain-Systemen (Schlatt et al. 2016)**

Chancen	Risiken
⇒ Detaillierte Zugangskontrolle	⇒ Hoher Energiekonsum durch PoW
⇒ Anonymität	⇒ Geringe Skalierbarkeit
⇒ Hohe Datenintegrität	⇒ Mangelnde Interoperabilität der Systeme
⇒ Hohe Netzausfallsicherheit	⇒ Sicherung privater Schlüssel
⇒ Kein Vertrauen für Interaktionen notwendig	⇒ Irreversibilität von Transaktionen
⇒ Hohe Prozessintegrität	⇒ Keine garantierte Anonymität
⇒ Große Transparenz	⇒ Mögliche Attacken
⇒ Kurze Dauer der Transaktionsabwicklung	
⇒ Programmierbarkeit der Transaktionen	

Neben den z.T. bereits erwähnten Argumenten wie hoher Netzausfallsicherheit, großer Transparenz und hoher Integrität beschäftigt sich das FIT auch kritisch mit möglichen Attacken und der möglichen Anonymität. So gibt es verschiedene Szenarien, in dem eine

Blockchain wohl über einen sog. 51%-Angriff geknackt und dem Ledger eine neue Wahrheit, also eine neuer World-State verpasst werden könnte. In diesem Fall müssten über die Hälfte aller beteiligten Peers zeitgleich manipuliert werden. Die Dimension eines solchen Angriffes sei zeitgleich der einzige Schutz dagegen.

Auch der hohe Energiekonsum durch das sog. „Mining“ steht hoch in der Kritik. Beim Mining kann jeder beteiligte Knoten mit weiterer Rechenleistung versehen werden und dem Knoten als Ressource zur Verfügung gestellt werden. Dies erfolgt in Form von Hardware, z.B. dem Arbeitsspeicher von Grafikkarten. Belohnt wird die bereitgestellte Ressource u.a. mit Kryptowährung, eine Langezeit auch in Deutschland beliebte Einnahmequelle. Nachdem die Rechenleistung aber immer größer wird, steigt auch der Energieverbrauch. In Deutschland ist sind die Stromkosten bereits seit einiger Zeit höher als die möglichen Einnahmen, weshalb das sog. Mining oft in Länder mit niedrigen Stromkosten ausgelagert werden.

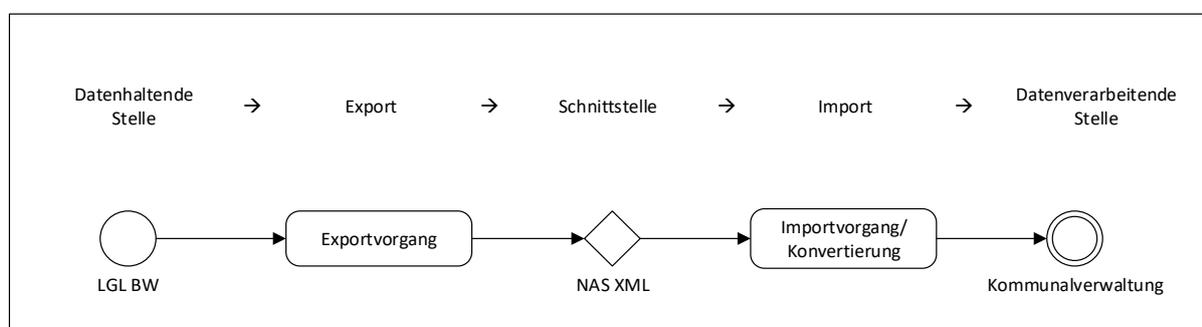
## 3. ANFORDERUNGSANALYSE

### 3.1. USE CASE

Für die vorliegende Arbeit wird kein neuer Anwendungsfall entwickelt, sondern ein bekannter Anwendungsfall vereinfacht und auf die Fragestellung bzw. in technischer Hinsicht auf die Blockchain-Anwendung übertragen.

Wie bereits in der Zusammenfassung sowie im Kapitel 1 erläutert geht es um klassische Flurstücksverwaltung in der amtlichen Liegenschaftsverwaltung. Die Blockchain soll den gesamten Lebenszyklus eines Flurstückes abbilden: Von der Entstehung eines Flurstückes über den Verkauf, bis hin zur Teilung oder Verschmelzung und damit auch der Löschung.

Basierend auf diesem Use Case werden anschließend die Forschungsfragen bearbeitet und geprüft, ob der erwartete Mehrwert durch die Anwendung dieser Technologie eintritt bzw. eintreten könnte.

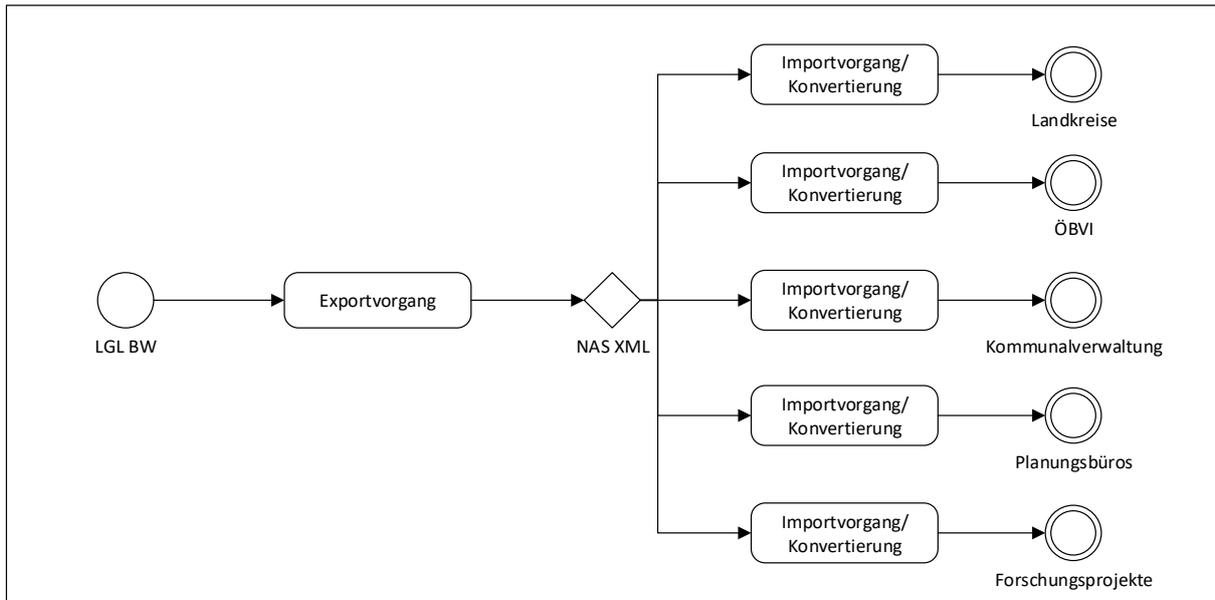


**Abbildung 9: Export von ALKIS-Daten**

Abbildung 9 zeigt den grundlegenden Anwendungsfall, dem Autor auch aus der beruflichen Erfahrung heraus ausgesetzt ist. Baden-Württemberg hat von seinen über 1100 Kommunalverwaltungen sehr viele Kommunen, die die amtlichen Liegenschaftsdaten in einem Geoinformationssystem verwenden. Die Liegenschaftsdaten werden im Automatisierten Liegenschaftskataster Informationssystem vorgehalten (kurz: ALKIS) und die Daten nach eigener Erhebung in einem Großteil der Fälle im Schnittstellenformat NAS (Normbasierte Austauschschnittstelle) geliefert und mittels der verschiedensten Konvertierungsprogramme in unterschiedliche Dateiformate und Datenbanksysteme übertragen. Dabei werden die Daten exportiert, die exportierten ALKIS-NAS-Daten importiert oder konvertiert und anschließend zur Verwendung im GIS bereitgestellt.

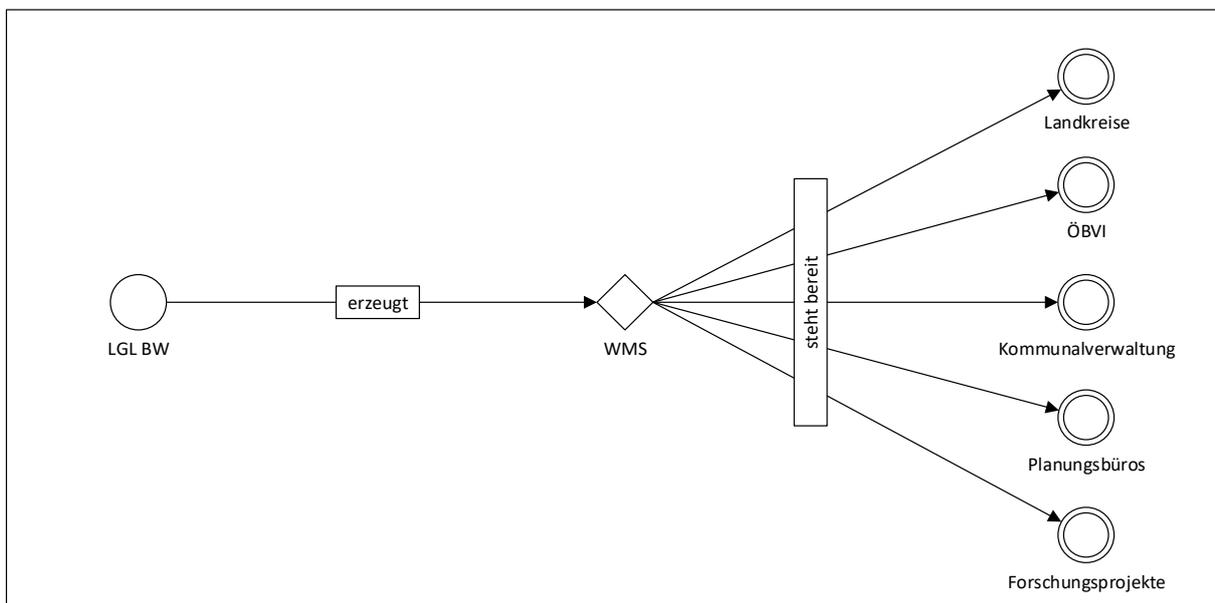
Neben Kommunalverwaltungen, die in Baden-Württemberg die Daten über eine Rahmenvereinbarung mit dem Städte- und Gemeindetag zur Verfügung gestellt bekommen, gibt es weitere Stellen, die die ALKIS-Daten für verschiedene Projekte verwenden.

So benötigen neben den Kommunalverwaltungen auch Landkreise, Öffentlich bestellte Vermessungsingenieure, Planungsbüros oder Forschungsprojekte die amtlichen Daten. Natürlich sollten die Daten immer auf dem aktuellsten und vollständigsten Stand sein. Um ALKIS-Daten tagesaktuell zu verwenden, werden Dienste verwendet.



**Abbildung 10: Überblick über verschiedene ALKIS-Anwender**

In Baden-Württemberg versucht das Landesamt für Geoinformation und Landentwicklung eines OGC-konformen Web Map Service (WMS) anzubieten. Dieser hätte den Vorteil, dass er, vorausgesetzt er basiert auf den originären Daten, möglichst aktuell wäre. Zum Vergleich: Eine einfache Kommunalverwaltung erhält nach eigenen Erhebungen die ALKIS-Daten oftmals zu einmal pro Jahr (eigene Auswertung von 60 betreuten Kommunalverwaltungen in Baden-Württemberg), da wäre die Nutzung eines WMS eine gute Lösung, da sie sowohl den Export- als auch den erneuten Importvorgang überflüssig macht. Das wird beim Vergleich zwischen Abbildung 10 und Abbildung 11 deutlich.



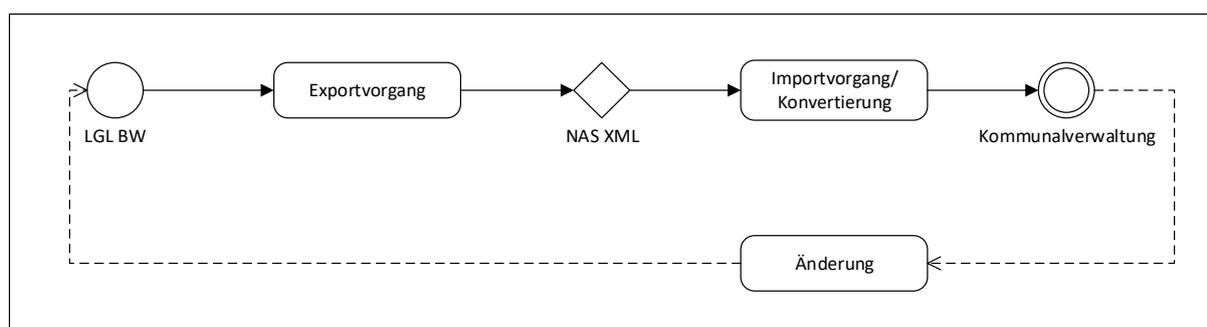
**Abbildung 11: Bereitstellung von ALKIS-Daten als WMS**

Der WMS würde zentral bereitstehen und alle Beteiligten könnten auf ihn zugreifen. Unglücklicherweise gibt es einige Gründe bzw. Hemmschwellen, die gegen eine Verwendung des WMS sprechen:

- Unzureichende Internetverbindung im ländlichen Raum (u.a. Regionen Schwarzwald & Schwäbische Alb)
- Zusätzliche Kosten, da der WMS nicht wie in anderen Bundesländern kostenfrei ist
- Nicht alle notwendigen Informationen stehen per WMS bereit
- ALKIS-NAS-Daten werden oft auch direkt nach DXF/DWG oder Shape exportiert, das ist mit einem WMS nicht möglich

In den ersten Überlegungen wurde nur die „Richtung“ von der datenhaltenden Stelle und Anwender betrachtet. Untersucht man, welche Aktionen in umgekehrter Richtung vorkommen, erkennt man u.a. Flurstücksänderungen sowie Fehlermeldungen in den ALKIS-Daten.

Von den Flurstücksänderungen sind die meisten Kommunalverwaltungen (in Bezug auf die ALKIS-Daten) nicht betroffen, da diese über die Katasterämter und die zuständigen ÖbVI abgewickelt werden. Es besteht daher bei den kleinen Kommunen i.d.R. nicht der Anspruch die ALKIS-Daten direkt zu verändern – nur bei größeren katasterführenden Einheiten (wie in Baden-Württemberg z.B. die Städte Karlsruhe, Stuttgart und Reutlingen). In den kleineren Ämtern sind die festgestellten Fehler viel schwerwiegender. Es kommt vor, dass Änderungen nicht übernommen werden, diese Fehler werden dann gemeldet. Abbildung 12 veranschaulicht den Vorgang.



**Abbildung 12: Änderungsmeldung ALKIS**

Anschließend werden diese Änderungen eingepflegt und stehen der Kommunalverwaltung dann nach etwa einem Jahr mit der nächsten ALKIS-Lieferung wieder zur Verfügung. Bis dahin muss die Verwaltung mit nicht-aktuellen Daten weiterarbeiten, wodurch Fehler und Missverständnisse auftreten können. Das passiert natürlich nicht nur bei einem Anwender, sondern bei vielen, wie Abbildung 13 darstellt.

Unter Umständen kann es hier vorkommen, dass ein Fehler gleich mehrfach gemeldet wird, da er sowohl der Kommunalverwaltung als auch dem Landkreis aufgefallen ist. Durch die Tatsache, dass weitere Beteiligte wie Planungsbüros trotzdem noch eine Zeit auf falschen Daten arbeiten, können – trotz der Meldung – weitere Fehler entstehen.

Die Datenhaltung beim Landesamt für Geoinformation und Landentwicklung Baden-Württemberg (kurz: LGL) ist zentral und kann in Bezug auf die Anwendung von Blockchain-Technologien gerne anhand der Grundlagen aus Kapitel 2 gerne diskutiert werden.

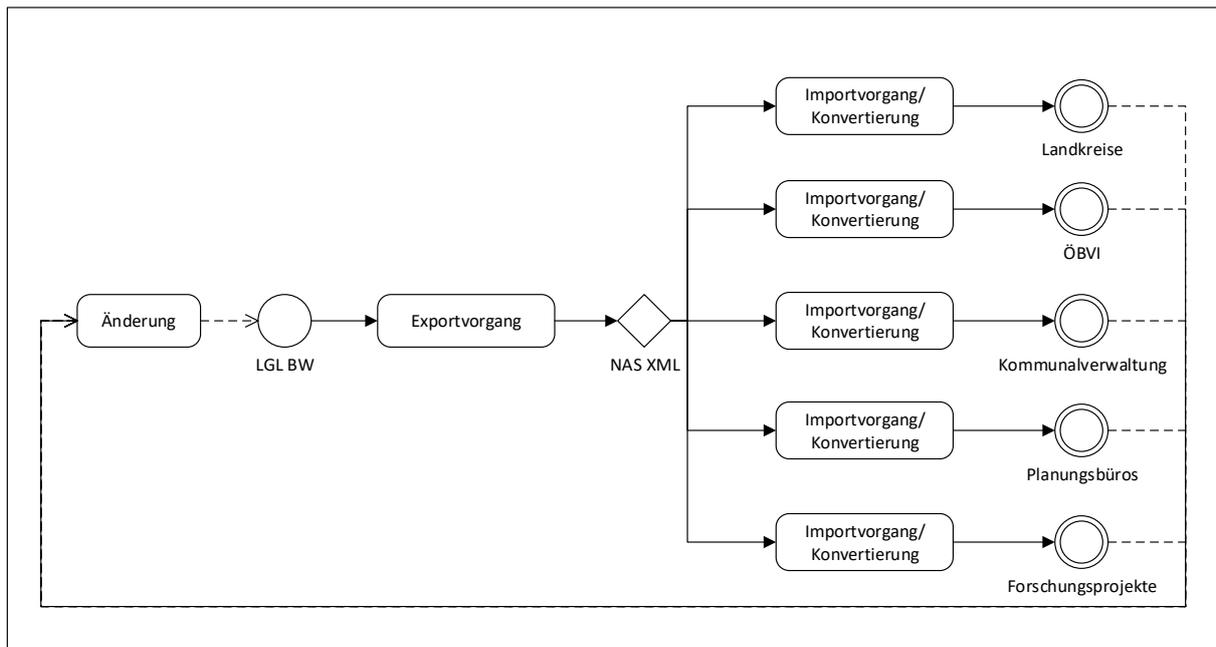


Abbildung 13: Änderungen im ALKIS bei n Anwendern

Genau hier setzt der Anwendungsfall für diese Arbeit an: Kann Blockchaintechnologie hier einen Mehrwert liefern, z.B. durch dezentrale Datenhaltung? Kann den Daten vertraut werden, da die zentrale Instanz wegfallen würde – und dadurch ggf. auch die potenziellen Fehler? Könnte die Aktualität der Daten überprüft werden, wie im Falle des Planungsbüros, dass noch gar nicht weiß, dass sich eine Datenänderung ergeben hat?

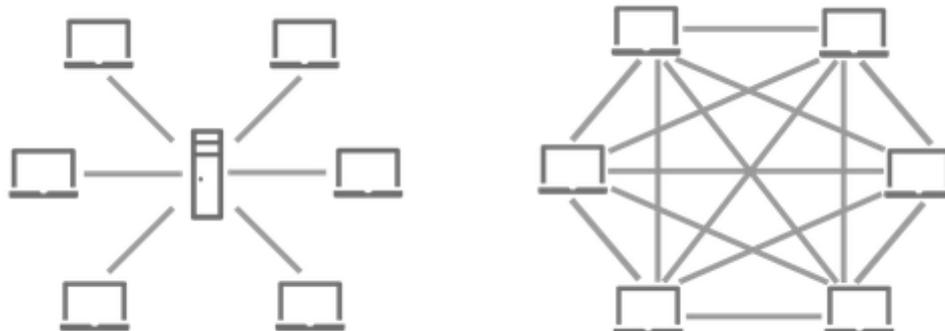


Abbildung 14: Ist-Zustand vs. Soll-Zustand (EDX 2018b)

Durch eine Adaption der zentralen Datenhaltung (Ist-Zustand) auf eine verteilte Datenhaltung (Soll-Zustand) könnten die Vorteile der Blockchain-Technologie auch für Prozesse im ALKIS-Umfeld zur Verfügung stehen und zeitgleich einige unbeliebte Nachteile der aktuell zentralen Datenhaltung beseitigt werden.

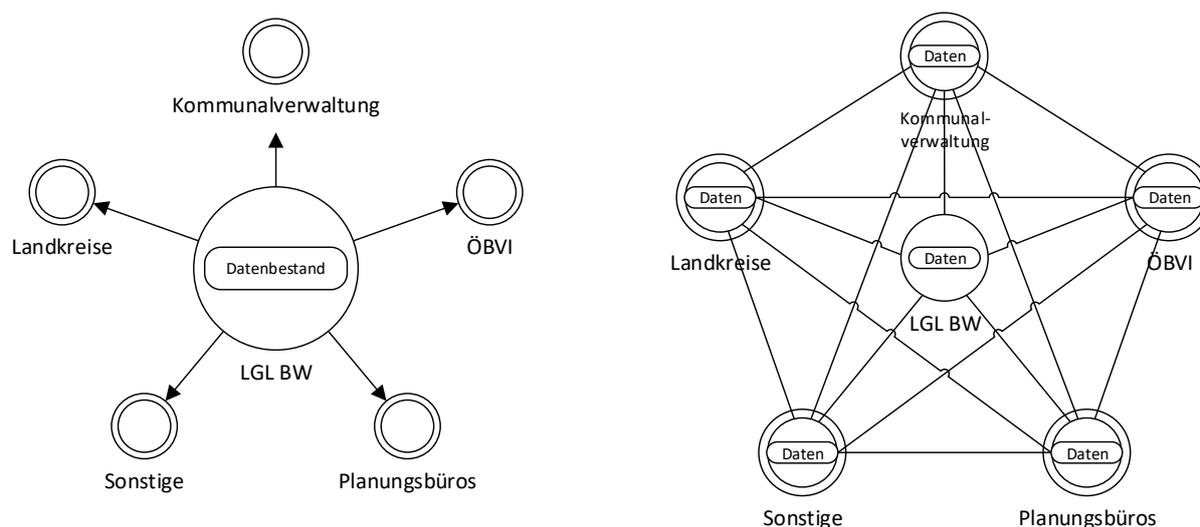


Abbildung 15: Datenbestand zentral vs. dezentral

So ist das Ziel dieser Arbeit, dass die Daten im Prinzip überall zeitgleich zur Verfügung stehen – jede Änderung im Bestand, bzw. jeder neue Zustand in der Blockchain wäre sofort verfügbar bzw. jedes Flurstück könnte auf seine Echtheit/Gültigkeit überprüft werden.

Der Autor hat hier lange überlegt, wie der Mehrwert einer Blockchain auch mit überschaubarem Aufwand integriert werden könnte. Eine Speicherung der ALKIS-Daten inklusive Geometrie schied schnell aus, da aktuell noch kein Standard für Geodatenhaltung innerhalb der Blockchain besteht und zudem die vielen verschiedenen Geoinformationssysteme wohl jeweils eine eigene Schnittstelle benötigen.

In der Abwägung kam der Gedanke, nicht die Geodaten selbst, sondern nur den Zustand eines Objektes in der Blockchain zu speichern. Der Zustand bedeutet, dass gewisse Eigenschaften gespeichert werden und jeder diese Eigenschaften überprüfen kann.

Als Beispiel wird auf Basis eines Attributes „Flurstückskennzahl“ eines beliebigen Flurstückes ein eindeutiger Hash erzeugt und zusammen mit einem Schlüssel in der Blockchain gespeichert. Ein Objekt Flurstück mit der Eigenschaft Flurstückskennzahl 08-09876-000-03001/051 könnte also mittels eines erzeugten Hashes (hier SHA-1)

Wert	Hash
08-09876-000-03001/051	f8f89ab5d12c76f50a2fda95053849296fb8a68e

im Ledger gespeichert werden. Jeder Knoten, z.B. ein Planungsbüro als Teil des Netzwerks, könnte also mit der Kombination Flurstückskennzahl und erzeugtem Hash den Ledger anfragen, ob das Flurstück gültig ist. Wird die Eingabe, die also durchaus auch aus einem anderen System erzeugt werden könnte, vom Ledger bestätigt, weiß der Teilnehmer, dass das Flurstück noch existiert.

Vielleicht ist das Flurstück zwar existent und daher gültig, aber bereits verkauft, kann die Blockchain keine Aussage treffen. Aus diesem Grund müssen weitere Attribute für die Hash-Erzeugung einbezogen werden. In unserem vereinfachten Datenmodell könnte das ein Feld für den Eigentümer sein:

Attribut	Wert
Flurstückskennzahl	08-09876-000-03001/051
Eigentümer	Stefan Rutka

Mit diesem Ansatz sollen folgende Fragen an die Blockchain gestellt werden können:

- Ist das Flurstück noch aktuell?
- Ist die Größe des Flurstücks noch gültig?
- Ist der Eigentümer noch gültig?

Mit diesem ersten Schritt könnten alle Teilnehmer im Netzwerk die Gültigkeit ihrer Geodaten im Einzelfall prüfen und wären nicht darauf angewiesen, neue Daten zu bestellen (und zu importieren/konvertieren).

Solange nur diese Richtung der Anfrage betrachtet wird, bietet die Blockchain aber noch keinen Vorteil – diese Anfragen könnten auch ohne weiteres an ein zentral gehaltenes Datenbanksystem gestellt werden. Zieht man aber weitere Stellen, die u.U. auch Geodaten ändern dürfen, wie z.B. katasterführende Städte hinzu, erschließt sich ein erster Vorteil: Eine Änderung der Daten würde unverzüglich zu einem neuen Hash führen und z.B. ein Landkreis, der seine Daten prüft, würde auf der Stelle sehen, dass sich eine Änderung ergeben hat. Es müssten die Änderungen also nicht separat erst in das zentrale System zurückgespielt werden, wie bereits in Abbildung 12 angedeutet. Je mehr Teilnehmer also auch Daten bearbeiten, desto interessanter wird die Anwendung einer Blockchain.

Auch Stellen ohne Interesse an den Geodaten an sich könnten an der Blockchain partizipieren, z.B. einen Notar, der den Verkauf eines Flurstücks abwickelt. Dieser könnte mittels einer Applikation und einer Schreibberechtigung den Eigentümerwechsel von A nach B in der Blockchain aktualisieren und jeder Knoten weiß sofort Bescheid, dass das Eigentumsverhältnis in Ihrem Datenstamm nicht mehr gültig ist. In diesem Beispiel wird also die Meldekette, die von der Unterschrift des Notars bis hin zur Änderung der Geodaten beim LGL stark verkürzt und die Institution der zentralen, datenhaltenden Stelle überflüssig.

Im letzten Teil wird auch betrachtet, wie Daten aus der Blockchain zurückgespielt werden können – diese Diskussion kann aber unabhängig vom oberen Beispiel betrachtet werden. Geplant ist, die Geodaten mittels einer Schnittstelle direkt aus der Blockchain zu erzeugen bzw. einzelne Datensätze in Fremdsystemen einfach zu aktualisieren.

Für die Änderung eines Eigentümers könnte schon ein einfaches UPDATE SQL-Statement genügen, bei einer Flurstücksteilung (aus einem werden zwei) könnte eine Kombination aus DELETE- (den alten Datensatz) und einem CREATE-Statement (zwei neue Teilflächen erzeugen) diese Aufgabe dezentral durchgeführt werden. Die zentrale Stelle müsste also nicht einmal mehr Daten exportieren, sondern lediglich die Zugriffsberechtigungen kontrollieren. Auch wäre hier eine Kombination aus öffentlicher und privater Blockchain denkbar, da eine Prüfung auf Gültigkeit von jeder Person durchgeführt werden sollte, eine Erzeugung kompletter Datensätze inkl. Eigentümerinformationen aber alleine schon aus datenschutzrechtlichen Gründen nicht jeder Person zugänglich sein sollte.

Betrachtet man also die aufgeführten Prozesse und Tätigkeiten sowie die steigende Anzahl Teilnehmern mit verschiedenen Aufgaben im Netzwerk, wird der Einsatz einer Blockchain attraktiver.

## 3.2. ANFORDERUNGEN

Nun muss definiert werden, welche Komponenten grundlegend zum Einsatz kommen und welche davon konfiguriert und auf die Fragestellung angepasst werden müssen. Beispiel: Können wir bei den meisten Frameworks auf bestehende Netzwerkdefinitionen zurückgreifen, die direkt „out of the box“ verwendet werden können, müssen andere Komponenten mal mehr, mal weniger aufwendig konfiguriert oder programmiert werden.

Die nachfolgende Tabelle 4 zeigt die wichtigsten Anforderungen, die aus dem im vorigen Kapitel vorgestellten Use Case, mit grundlegender Beschreibung und Verweis auf die entsprechende Komponente.

**Tabelle 4: Anforderungen an die Softwarekomponenten**

Begriff & Komponente	Kurzbeschreibung
<b>Framework</b>	Das gewählte Blockchain Framework muss modular aufgebaut sein, sodass notwendige Module einfach miteinander kombiniert werden können. Die Blockchain soll sowohl privat als auch öffentlich betrieben werden können
<b>Zugang</b>	Eine Kontrollkomponente zur Prüfung der Zugangsberechtigung soll integriert werden, dabei ist wichtig, dass unterschiedlich komplexe Vorgehensweisen integriert werden können, bei Bedarf beispielsweise auch eine sichere 2-Faktor-Authentifizierung
<b>Berechtigung</b>	Das Framework muss in der Lage sein, verschiedene Benutzergruppen zu verwalten und Berechtigungen verteilen zu können, der Unterschied kann bspw. in der Bearbeitung oder dem Umfang von Informationsabfragen liegen
<b>Systemintegration/API</b>	Das Framework muss eine Schnittstelle bieten, um externe Systeme anzubinden. Das kann von einem sehr einfachen User-Interface über Datenaktualisierungen in einem Geoinformationssystem bis hin zur Einbindung einer externen Active-Directory gehen
<b>Chain Code/Smart Contracts</b>	Eine Komponente für die Entwicklung eigener Smart contracts auf der Basis von Chain Code muss gewährleistet sein. Dadurch können später verschiedene Aufgaben definiert werden, bspw. das Updateverhalten des Ledgers oder auch die Ausgabe von Daten in Form eines Exports
<b>Hashing-Algorithmus</b>	Eingesetzter Algorithmus für Kryptographie. Nicht alle Frameworks unterstützen nativ die wichtigsten Algorithmen, im Idealfall bietet das Framework bereits eine integrierte Komponente
<b>Transaktionen</b>	Die Transaktionen müssen anhand des gewählten Use Case definiert und umgesetzt werden. Die grundlegenden Anforderungen aus dem Use Case sind hierbei zu erfüllen
<b>User Interface</b>	Vgl. auch API: Ein auf Basis der API gestaltetes User Interface soll die Anfragen an die Blockchain in einfacher Form zurückgeben, etwa als HTML-Dokument oder eingebettet in Fremdsysteme
<b>Datenmodell/Ledger</b>	Die Definition des Datenmodells hat maßgeblichen Einfluss auf den Ledger selbst. Hier gilt es zu berücksichtigen, dass alle notwendigen Informationen abgelegt werden, um eine langfristige Nutzung des Ledgers gewährleisten zu können

Die in Tabelle 4 aufgeführten Anforderungen sind die Grundlage der Fallstudie und bilden den Einstieg in die Konzeption der Blockchain. An dieser Stelle der Ausarbeitung kamen dem Autor aber auch Zweifel an der Notwendigkeit des Einsatzes von Blockchaintechnologie allgemein für den gewählten Anwendungsfall, aber auch im speziellen über den Umfang der Prototypentwicklung auf. deshalb soll das nachfolgende Kapitel in einer Art Zwischen-Evaluation diesen Aspekt kritisch hinterfragen.

Bevor es jedoch dazu übergeht, müssen an dieser Stelle auch die Anforderungen dokumentiert werden, die an den Ledger selbst gestellt werden. Dazu werden die gebräuchlichsten Änderungs- und Fortführungsfälle von Flurstücksdatenbanken herangezogen und in der folgenden Tabelle dargestellt. Diese Fälle werden in der Umsetzung auch praktisch berücksichtigt und dokumentiert.

**Tabelle 5: Anforderungen an den Ledger**

Anwendungsfall	Beschreibung
Initialisierung neuer Datensätze	Vor allem zu Beginn müssen einmalig alle Objekte neu initialisiert werden, später lediglich vereinzelte, neue Flurstücke
Änderung im Eigentumsverhältnis	In diesem Fall wird keine Änderung an Geometrie oder Größe der Flurstückes vorgenommen, sondern lediglich der Eigentümer geändert
Sonderfall 1: Verschmelzung	In diesem Spezialfall werden 2 oder mehr Flurstücke zu einem neuen zusammengefasst
Sonderfall 2: Teilung	In diesem Spezialfall wird ein Flurstück in 2 oder mehr neue Flurstücke geteilt
Sonderfall 3: Umlegung	In diesem Spezialfall wird eine Vielzahl von Flurstücken in eine neue Vielzahl von Flurstücken umgelegt

Die in Tabelle 5 aufgeführten Anforderungen sind elementar für die Beantwortung der grundlegenden Frage, ob sich Blockchaintechnologie im Einsatz bei der Verwaltung von Flurstücken eignet. Sollten gerade auch die Sonderfälle nicht abgebildet werden können, macht die Verwendung u.U. wenig Sinn.

### 3.3. ZWISCHENEVALUATION

Nach Entwicklung der Fragestellung stellte sich dem Autor die grundlegende Frage, ob auf einem bestehenden Blockchain-Ökosystem aufgesetzt oder eine eigene (Test-)Umgebung geschaffen wird. Die Frage steht maßgeblich mit der verwendeten Zeit in Abhängigkeit, ist der Aufwand doch wesentlich höher eine eigene Umgebung aufzubauen (auch wenn natürlich bestehende Komponenten zum Einsatz kommen). Im Grunde besteht die Frage darin, ob die zu prüfenden (Test-) Daten in eine bestehende Blockchain eingespielt und dort gespeichert werden (z.B. innerhalb einer Ethereum-Umgebung) oder ein eigenes Framework verwendet wird. Das Hyperledger Framework bietet hierfür die goldene Mitte, da sowohl vorhandene Komponenten direkt eingesetzt, aber auch konfiguriert und angepasst werden können. Ebenfalls hätte eine Anforderungsanalyse auf eine andere Blockchain erfolgen müssen, im Sinne der Frage ob das Framework in der Lage ist die notwendigen Anforderungen zu erfüllen. Bei Bitcoin ist das bspw. schwierig, da das Framework für einen bestimmten Anwendungsfall entwickelt wurde und wenig Flexibilität bietet.

Die zweite begleitende technische Frage war, ob die Geodaten direkt in der Blockchain gespeichert werden (können) oder ob die Datenhaltung der Geodaten getrennt abgewickelt wird. Hierzu war es wichtig, die aus den Recherchen bekannten Anwendungen mit Geo-Aspekt zu vergleichen und hinsichtlich der Anforderungen zu untersuchen. An dieser Stelle kamen dem Autor Zweifel, ob eine Blockchain überhaupt notwendig ist. In verschiedenen Diskussionen mit Bekannten und GIS-Fachleuten musste der Einsatz der Technologie verteidigt werden. Mit Hinweis auf den Vorteil der hohen Aktualität, die eine Blockchain in einem Peer-Netzwerk bietet, wurde öfter darauf hingewiesen, dass doch der Einsatz von OGC-konformen Standards wie WMS, WFS oder WFS-T ebenfalls eine hohe Aktualität bieten (vgl. auch Abbildung 11 – WMS-Beispiel). Dem ist grundsätzlich zuzustimmen, allerdings bleibt in diesem Fall die Bearbeitung und Bereitstellung zentral, was nach wie vor einige Prozesse, wie bspw. das Melden von Fehlern im Datenbestand und die daraus resultierende, aufwendige Aktualisierung hinsichtlich der zeitlichen Achse einschränkt oder sogar „lahmt“.

Die nachfolgende Tabelle 6 soll mit Verweis auf „konkurrierende“ Technologie zur Entscheidungsfindung beitragen und die aufkommenden Zweifel ausräumen, dazu sollen die Vor- und Nachteile von Blockchain und den wichtigsten alternativen Technologien verglichen werden. Dazu werden die grundlegenden Eigenschaften einer Blockchain zum Vergleich herangezogen (vgl. Kapitel 2.1 Begriff und Definition):

**Tabelle 6: Blockchain-Vorteile im Vergleich zu Alternativen (nach Blockchainwelt 2018b)**

Stichwort	Blockchain	Alternative
<b>Änderungen/Updates</b>	Initial	fortlaufend
<b>Datenhaltung</b>	Manipulationssicherheit durch Dezentralität	Zentralität bietet Risiko
<b>Transaktionsrate</b>	Im Vergleich schlecht, da Power abhängig vom Netzwerk	Power abhängig vom Server – i.d.R. schneller
<b>„Ethisch/Moralisch“</b>	Gemeinschaftlich (Konsens)	Institutionell (Abhängig)
<b>Prüfung</b>	Transparent, jede Transaktion nachvollziehbar	Abhängig von Institution
<b>Vertrauen</b>	Kein Vertrauen notwendig	Vertrauen wichtig, Partner muss „seriös“ sein

Auch ein Online-Kurs der LINUX-Foundation hinterfragt den Einsatz mittels eines gut gestalteten Argumentenkataloges pro/contra Blockchain und verweist mehrfach darauf, die Anforderungen zu prüfen um zu einer richtigen Entscheidung zu kommen (EDX 2018a). Einige Fragen, die gestellt werden sollten, sind:

- ☉ Wie viele Teilnehmer sind im System?
- ☉ Wie sieht die räumliche Verteilung
- ☉ der Teilnehmer aus?
- ☉ Welche Anforderungen werden an die Performanz gestellt?

Die grundlegenden Argumente sind in der nachfolgenden Tabelle 7 aufgeführt:

Tabelle 7: Pro &amp; Contra zum Einsatz von Blockchain am Beispiel von Hyperledger

Unter welchen Umständen sollte ...Blockchain verwendet werden?	...Blockchain nicht verwendet werden?
<ul style="list-style-type: none"> <li>➤ Bedarf an einer gemeinsamen Datenbank ist vorhanden</li> <li>➤ Teilnehmer haben unterschiedliche Interessen oder auch kein Vertrauen zueinander</li> <li>➤ Es sind viele Teilnehmer mit unterschiedlichen Berechtigungen, gerade auch Schreib-Berechtigung vorhanden</li> <li>➤ Es sind Dritte in die Prozesse eingebunden, z.B. Notare, Lizenzgeber oder Datenanbieter</li> <li>➤ Einsatz von Kryptographie ist erforderlich</li> <li>➤ Das bisherige System ist in viele kleine Abschnitte und Datenbanken/Subsysteme entlang der Prozesskette unterteilt</li> <li>➤ Es gibt fest definierte Regeln</li> <li>➤ Hoher Bedarf an einer eindeutigen Historie/Verlaufs ist gegeben</li> <li>➤ Anzahl von Transaktionen pro Sekunde übersteigt nicht 10.000 Stk.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Es geht um streng geheime Informationen</li> <li>➤ Es geht um statische, wenig dynamische Daten und die Datenmenge ist sehr groß (Big Data)</li> <li>➤ Die Regeln für Transaktionen ändern sich regelmäßig</li> <li>➤ Daten von Drittanbietern müssen gespeichert werden, auf deren Aktualisierung kein Einfluss besteht</li> </ul>

### 3.4. VERWENDETE SOFTWARE

Für den Blockchain-Teil der Anwendung werden ausschließlich Open-Source-Komponenten herangezogen. Das beginnt beim Betriebssystem, welches die Umgebung der für die verschiedenen Komponenten bereitgestellt, in diesem Fall LINUX Ubuntu.

Darauf aufbauend wurden grundlegend benötigte Komponenten installiert, um zu einem späteren Zeitpunkt mit dem Hyperledger Framework fortzufahren. Die Komponenten und deren Aufgabe bzw. Zusammenhang werden im Folgenden aufgelistet.

Tabelle 8: Übersicht eingesetzter Softwarekomponenten für Blockchain-Applikation

Software	Beschreibung	Version
<b>Virtual Box</b>	Virtualisierungsumgebung	v5.2.18
<b>LINUS Debian Ubuntu</b>	Betriebssystem	v18.04.01
<b>cURL</b>	Programmbibliothek	v7.58.0
<b>Node.js</b>	Entwicklungsumgebung f. Netzwerkanwendungen	v8.10.0
<b>npm package manager</b>	Paketmanager für Node.js	v3.5.2
<b>Go Language</b>	Programmiersprache	v1.11
<b>Docker CE</b>	Containervirtualisierung	v18.06.1
<b>Docker Compose</b>	Verwaltung mehrerer Container	v1.17.1
<b>Hyperledger Fabric</b>	Blockchain Framework (hier als Docker-Image)	v1.1.0

Des Weiteren wurden auch proprietäre Systeme eingesetzt, im Wesentlichen um die Testgeodaten zu erzeugen und aufzubereiten sowie später die Daten zu visualisieren.

Für die Aufbereitung wurde GeoMedia Desktop (Hexagon) herangezogen, für die Visualisierung und auch die Aufbereitung der Eingabemasken (ersichtlich in den Schaubildern in Kapitel 5) wurde M.App Enterprise verwendet (ebenfalls Hexagon).

**Tabelle 9: Übersicht eingesetzter Softwarekomponenten für Geovisualisierung**

Software	Beschreibung	Version
<b>PostgreSQL/PostGIS</b>	Räumliches DBMS	v10.5/2.4
<b>GeoMedia Desktop</b>	Desktop GIS	v16.5.0.3138
<b>M.App Enterprise</b>	GIS-Plattform	v16.5.604.1

### 3.5. VERARBEITETE DATEN

In diesem Fall werden fiktive Daten verarbeitet, welche sich am Amtlichen Liegenschaftskatasterinformationssystem (kurz: ALKIS) orientieren. ALKIS wurde in Baden-Württemberg im Jahr 2011 eingeführt und ist ein bundeseinheitlicher Standard für die Führung der Liegenschaftskataster.

Im Speziellen untersuchen wir eine abstrahierte Version der Objektart Flurstück inkl. dessen Beziehung auf das Liegenschaftsbuch, in dem die Eigentümer zugeordnet sind. Zur einfacheren Darstellung des Konzept wurde auf eine exakte Übernahme des Datenmodells verzichtet und eine vereinfachte Version gewählt, da die Komplexität und der Umfang des echten Datenmodells keinen Einfluss auf die Fallstudie und den Proof-of-Concept hat.

Für die Testreihe wurde ein einfaches Datenmodell mit folgenden grundlegenden Informationen gewählt:

Objektbezeichnung: flurstueck  
Geometrietyp: Polygon

Hinzu kommen ein Primärschlüssel sowie verschiedene Attribute, welche in unterschiedlichem Umfang bei den einzelnen Spezialfällen benutzt werden. Dazu gehören u.a. ID/gmlid, Flächengröße, amtliche Flächengröße, Eigentümer. Die Eigentümer werden in diesem Beispiel absichtlich nicht als separate Tabelle geführt, wohlbewusst, dass ein Flurstück N Eigentümern zugeordnet sein kann. Die relationale Beziehung einzelner Tabellen spielt für diese Arbeit aber ebenfalls keine Rolle.

Eine detaillierte Beschreibung des Testdatenmodells erfolgt in Kapitel 4.4.

**Tabelle 10: Übersicht Daten**

Attribut	Beschreibung	Beispiel
<b>gmlid</b>	Eindeutige Identifikationsnummer	DEBWL0010905xpGv
<b>flskz</b>	Eindeutige Bezeichnung des Flurstückes	087930-000-02572/0001
<b>amtlflaeche</b>	Eingetragene Grundbuchfläche	2137
<b>istgueltig</b>	Besitzt der Datensatz Gültigkeit	Ja = 1, Nein = 0
<b>eigentuemer</b>	Name des Eigentümers	Müller, Werner
<b>zeitstempel</b>	Zeitstempel der Transaktion auf dem Ledger	2019-01-01
<b>version</b>	Version des Flurstücks unter derselben gmlid	1

Hinweis: Auf die vollständige Abbildung des ALKIS-Datenmodells wurde verzichtet.

## 4. UMSETZUNG

### 4.1. GRUNDLAGEN ZUR ANWENDUNG

Für die Umsetzung der Anwendungen werden Bestandteile des Hyperledger Fabric Frameworks verwendet. Manche Komponenten werden angepasst auf die eigenen Anforderungen, hierzu gehören der Ledger, der Chaincode und Teile der Applikation. Einige Komponenten werden 1:1 übernommen und nicht angepasst. Dafür gibt es verschiedene Gründe:

- Komplexität zu hoch (z.B. Konsensalgorithmen)
- Spielt für diese Arbeit keine zentrale Rolle (z.B. Ordering Service), sondern muss einfach „funktionieren“

Der Fokus dieser Arbeit liegt an zwei Bestandteilen, wobei der Peer die zentrale Rolle spielt, Hier werden die zentralen Bestandteile der Blockchain im Netzwerk verteilt. Ein Peer rollt den Chaincode, den Ledger inkl. Blockchain und damit auch den World State aus (s. Abbildung 16).

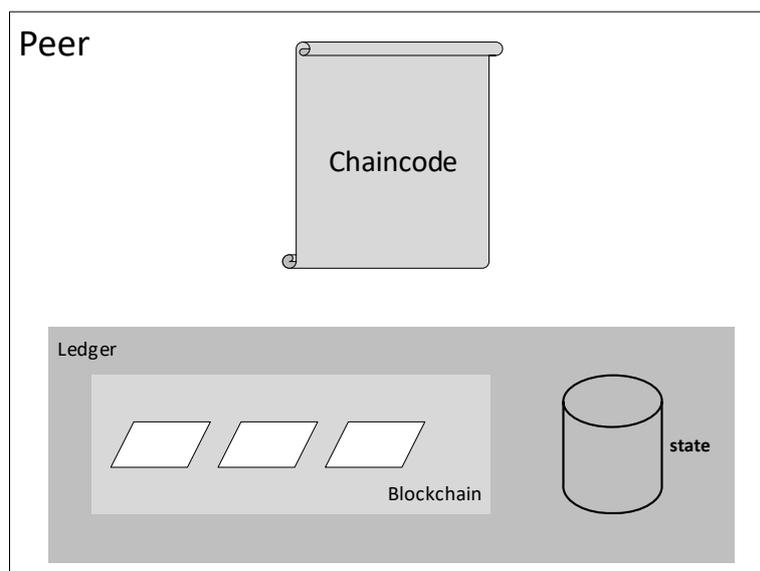


Abbildung 16: Peer

Hier werden auch die größten Anpassungen vorgenommen, damit die Applikation und das Netzwerk Flurstücke verwalten können. Definiert wird das Datenmodell, welches mittels Chaincode auf den Ledger geschrieben wird, damit kann die Blockchain entstehen und daraus resultiert automatisch der World State.

Der größte Eingriff muss am Chaincode vorgenommen werden. Hier werden die zentralen Methoden für unterschiedliche Zugriffe definiert wie Query, Init oder Update. Die einzelnen Punkte und Anpassungen werden in den folgenden Kapiteln detailliert erläutert.

## 4.2. ARCHITEKTUR

Die Hyperledger Fabric Architektur setzt sich, wie die anderen Hyperledger-Projekte auch, aus modularen Bestandteilen zusammen. Dadurch können einzelne Module auch quer durch die Hyperledger-Frameworks verwendet werden. Abbildung 17 zeigt die einzelnen Module.

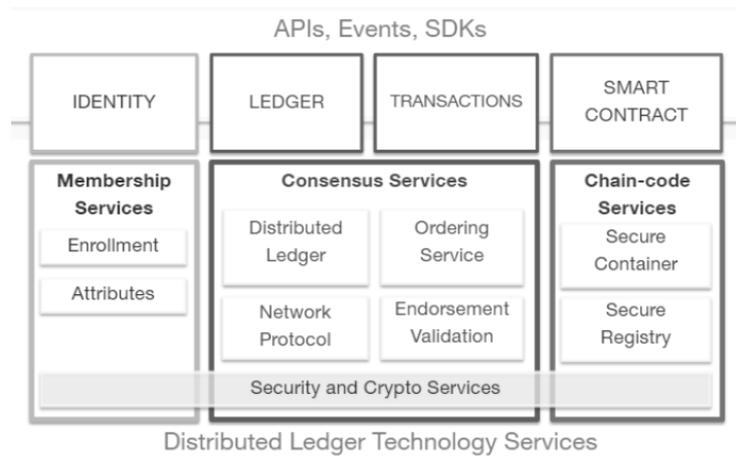


Abbildung 17: Architektur von Hyperledger Fabric (LFD 271 2018)

Die Verwaltung von Identitäten und somit auch Zugriffsrechten (Membership Services) bildet das Tor zum Framework, welches mittels API und SDK angesprochen wird. Zentral stehen die Transaktionen, welche auf den Ledger geschrieben werden. Damit das funktioniert, bringt Hyperledger Fabric einen eigenen Konsens-Dienst mit. Die letzte Säule des Frameworks sind die Smart Contracts bzw. der Chaincode. Hier wird definiert wie Transaktionen aufgebaut werden und ablaufen zu haben. Zuletzt gibt es verschiedene Sicherheits- und Kryptoservices, die auch ausgetauscht werden können ohne einzelne Komponenten des Frameworks zu beeinflussen.

Die meisten Module werden als vorgefertigte Docker-Images mit der Installation von Hyperledger Fabric mitgeliefert.

Ein klassischer Hyperledger Fabric Workflow wird in Abbildung 18 gezeigt. Die Fabric SDK steht zwischen Anwender und Netzwerk. Durch die Fabric Certificate Authority (CA) werden ECerts ausgestellt und somit die Zugriffsberechtigung auf das Blockchain Netzwerk verwaltet.

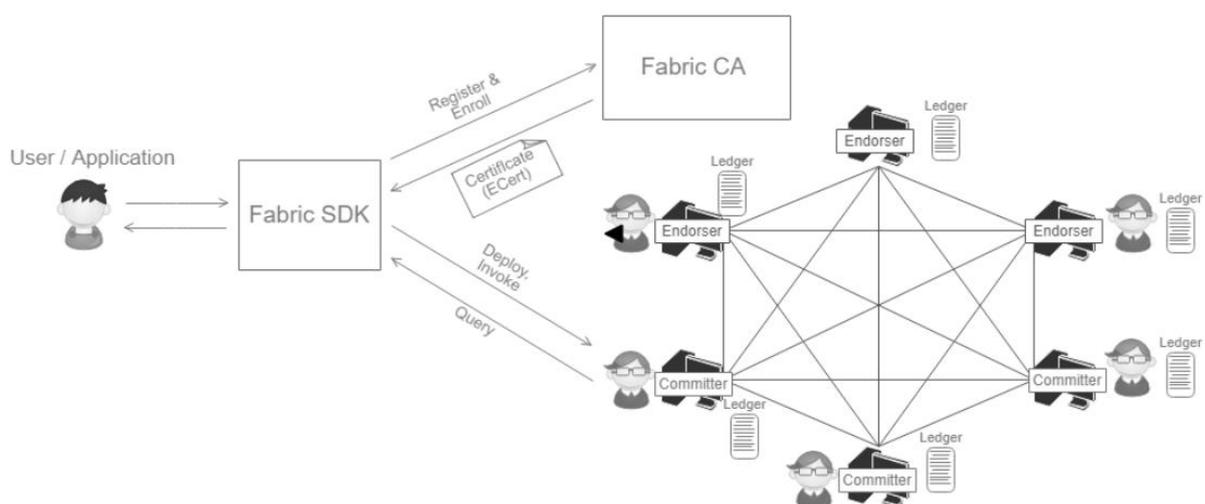


Abbildung 18: Hyperledger Fabric Workflow (LFD 271 2018)

Ebenfalls bietet die SDK Zugriff auf die Peers, bzw. zunächst den lokalen Peer des Teilnehmers. Die Endorser simulieren die eingehende Transaktion auf den Ledger, die Committer validieren die Transaktionen. Die Abbildung zeigt auch, dass nicht jeder Peer gleich aufgebaut sein muss.

Angelehnt an die Möglichkeiten des Hyperledger Fabric Frameworks wurde auf Grundlage der Anforderungsanalyse ein eigener Workflow entwickelt, welcher in diesem Kapitel vorgestellt wird und dokumentiert ist. Abbildung 19 zeigt die zentralen Prozesse der Applikation.

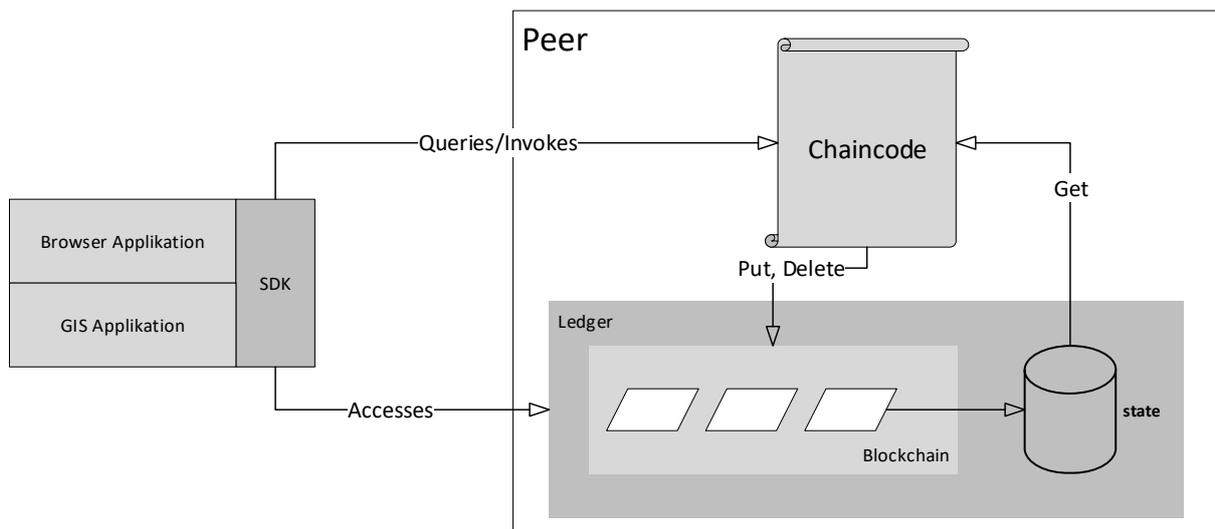


Abbildung 19: Prozessdiagramm der Applikation

Der Prozess gliedert sich in folgende Einzelschritte:

- Die Applikation und der Chaincode werden definiert und das Netzwerk gestartet
- Ein Netzwerkteilnehmer greift über eine Applikation mittels SDK auf das Netzwerk zu
- Die Applikation kann per Chaincode Aufrufe (Abfragen und Transaktionen) erzeugen
- Diese Aufrufe werden von der definierten Prozesslogik (Smart Contracts) des Chaincode gehandhabt
- Mittels Put- oder Delete-Kommando kann eine valide Transaktion an den Ledger angehängt werden
- Mittels Get-Kommando kann lesend auf den Gesamtzustand der Blockchain zugegriffen werden, dieser Zugriff wird aber nicht aufgezeichnet, er ist keine Transaktion sondern eine Query
- Die Applikation kann mittels API auf direkt Informationen der Blockchain abrufen (z.B. Queries) ohne über den Chaincode gehen zu müssen

Hier gliedern sich auch die geplanten Zugriffe für dieses Szenario. Der Chaincode wird zunächst nur exemplarisch via Browser-Applikation angesprochen. Bei erfolgreichem Zugriff könnte dieser Zugriff selbstverständlich auch auf GIS-Applikationen erweitert werden.

Als GIS-Zugriff wird in dieser Applikation beispielhaft der direkte SDK-Zugriff auf den Ledger verwendet. Durch diesen lesenden Zugriff soll der Ledger regelmäßig angesprochen und die Transaktionen zur weiteren Verarbeitung in eine PostGIS-Datenbank ausgelesen werden. Durch entsprechende Trigger können neue Datensätze jederzeit hinzugefügt werden. Auf diese Datenbank kann nun auch mit jedem Standard-GIS-Werkzeug zugegriffen werden.

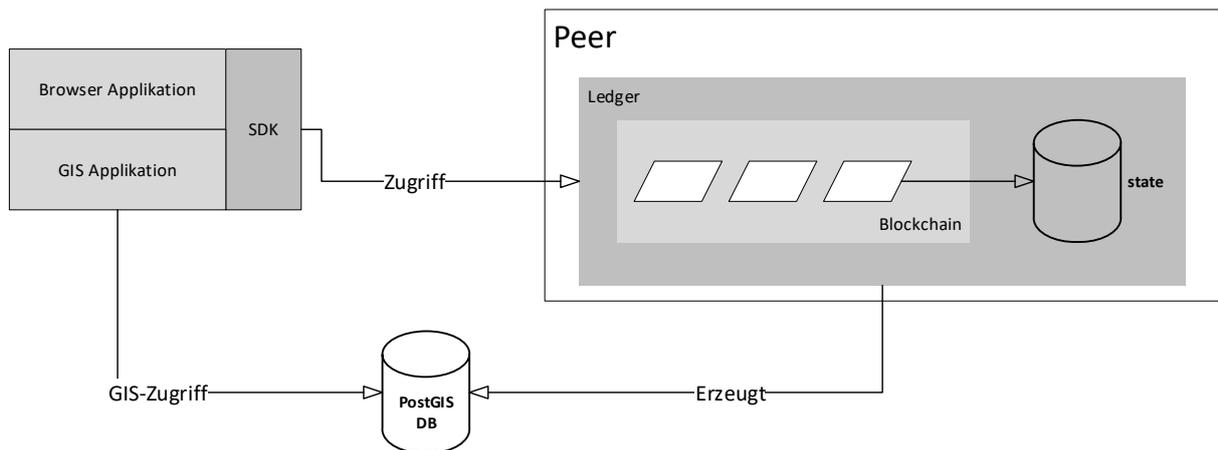


Abbildung 20: Einbindung einer PostGIS-Datenbank

Abbildung 20 zeigt dieses Vorgehen. Der Vorteil ist, dass man nur notwendige Zugriffe über die SDK und die Blockchain selbst laufen lassen muss. Ein Großteil der eigentlichen Nutzung findet ausgelagert auf eine Geodatenbank, hier PostGIS, statt. Ein weiterer Vorteil ist hier auch, dass PostGIS ein weit verbreiteter Standard ist und von den meisten GIS-Programmen umfassend unterstützt wird.

Zunächst wird der detaillierte Workflow der sog. FlurstueckApp vorgestellt. Aufbauend auf Abbildung 19 zeigt Abbildung 21 die Zusammenhänge bezogen auf die Flurstücks-Applikation.

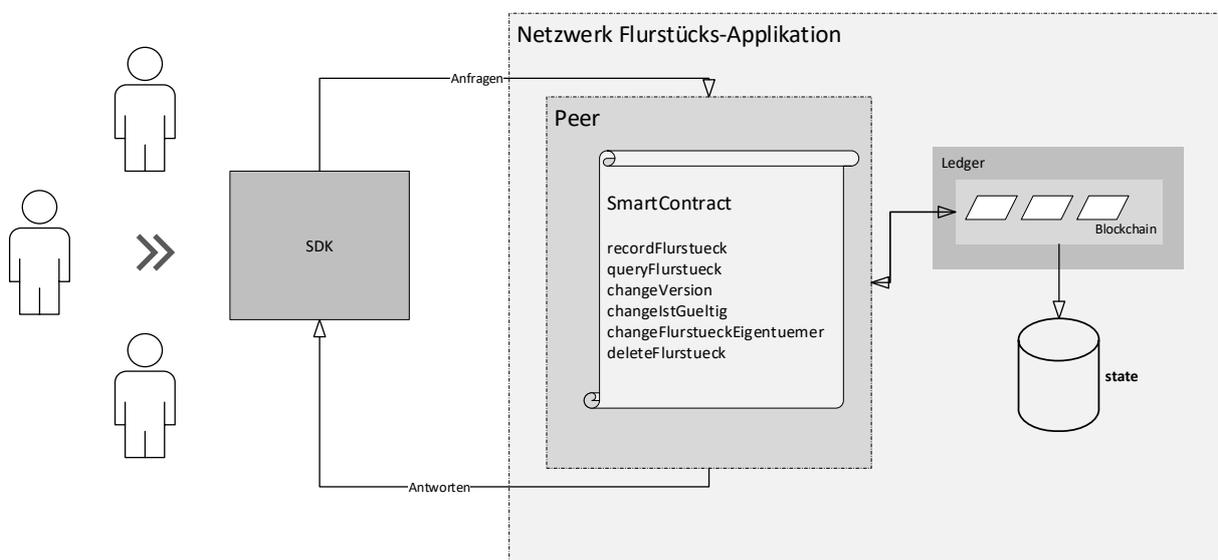


Abbildung 21: Workflow Flurstücks-Applikation

Das grundlegende Prinzip:

- Verschiedene Nutzer (z.B. Ämter, Kommunen, ÖbVI, Bürger) können mittels Applikation mit dem Netzwerk interagieren
- Je nachdem, ob auf den Ledger geschrieben werden soll (bspw. wenn ein Flurstückeeigentümer geändert wird) oder ob nur eine Abfrage erfolgen soll (z.B. ob das Flurstück noch gültig ist), wird eine Anfrage an den Peer gesendet
- Der (Endorsing) Peer wird Chaincode verwenden, um die Transaktion zu simulieren. Wenn keine Einwände vorliegen, wird die Transaktion an den Orderservice gesendet.

Dieser organisiert viele Anfragen aus dem gesamten Netzwerk und packt viele angefragte Transaktionen in einem Block zusammen.

- Der neue Block wird an die (Committing) Peers im Netzwerk verteilt
- Jeder dieser Peers wird den Block validieren und auf seinen eigenen Ledger schreiben. Die Transaktion war erfolgreich und findet sich im World-State wieder
- Jeder weitere Zugriff, z.B. lesend, wird diese Änderung beinhalten

### 4.3. CHAINCODE

Der Chaincode wird in der Programmiersprache Go angelegt (vgl. auch Kapitel 2.2.4, S. 21). Zunächst müssen mit einem einfachen Statement die Abhängigkeiten importiert werden. Der Befehl „import“ hilft dabei:

```
package main
import (
    "fmt"
    "github.com/hyperledger/fabric/core/chaincode/shim"
    "github.com/hyperledger/fabric/protos/peer"
)
```

Im nächsten Schritt muss der Code-Container erstellt werden:

```
type flurstueckChaincode struct {
}
```

Im nächsten Schritt muss die Init Methode implementiert werden. Über diese Methode wird der Zugriff auf den Ledger mittels API definiert:

```
func (t * flurstueckChaincode) Init(stub shim.ChainCodeStubInterface) peer.Response {
    // Get the args from the transaction proposal
    args := stub.GetStringArgs()
    if len(args) != 2 {
        return shim.Error("Argumente falsch. Schluessel und Wert werden erwartet.")
    }
    // Store key and value on the ledger
    err := stub.PutState(args[0], []byte(args[1]))
    if err != nil {
        return shim.Error(fmt.Sprintf("Datensatz konnte nicht erzeugt werden: %s", args[0]))
    }
    return shim.Success(nil)
}
```

Jetzt ist die Invoke Methode an der Reihe, diese wird bei einer neuen Transaktion aufgerufen, bspw. wenn ein Anwender eine Änderung eintragen möchte. In unserem Fall möchten wir entweder bestimmte Werte eines Datensatzes zurückbekommen oder ein Update für einen bestimmten Datensatz einbringen.

```
func (t * flurstueckChaincode) Invoke(stub shim.ChainCodeStubInterface) peer.Response {
    // Extract the function and args from the transaction proposal
    fn, args := stub.GetFunctionAndParameters()
    var result string
    var err error
    if fn == "set" {
        result, err = set(stub, args)
    } else { // assume 'get' even if fn is nil
        result, err = get(stub, args)
    }
    if err != nil { //Failed to get function and/or arguments from transaction proposal
        return shim.Error(err.Error())
    }
    // Return the result as success payload
    return shim.Success([]byte(result))
}
```

Es gibt zwei Basisaktion, die ein Anwender anstoßen kann: get and set.

- Mit get kann eine Anfrage ab- und eine Antwort über den Wert eines bestehenden Datensatzes zurückgegeben
- Die set Methode wird verwendet um einen neuen Datensatz zu erstellen oder einen bestehenden Datensatz zu verändern

Diese Aktionen werden mit GetFunctionAndParameters aufgerufen. Damit werden die Funktionsnamen und Parameter vom Rest des Codes isoliert.

Jede Transaktion ist grundsätzlich ein get oder set. Im Folgenden wird die set-methode eingebracht:

```
func set(stub shim.ChaincodeStubInterface, args []string) (string, error) {
    if len(args) != 2 {
        return "", fmt.Errorf("Argumente falsch. Schluessel und Wert erwartet.")
    }
    err := stub.PutState(args[0], []byte(args[1]))
    if err != nil {
        return "", fmt.Errorf("Failed to set asset: %s", args[0])
    }
    return args[1], nil
}
```

Die set-methode erzeugt oder modifiziert einen Datensatz. Die Identifikation läuft über einen bestimmten Schlüssel und den entsprechenden Wert. In unserem Fall könnten eindeutige Flurstücke über die Schlüssel gmlid oder flskz ermittelt werden – beide sind einzigartig und eindeutig.

Um Änderungen zu vollziehen, gibt es zwei Möglichkeiten: Entweder ein einzelner Wert mittels PutState wird durch ein neuen ersetzt (z.B. der Eigentümer), dann muss das Feld version ebenfalls angepasst werden (z.B. von 1 auf 2). Die Alternative ist, einen komplett neuen Datensatz mit den entsprechenden Werten zu erzeugen.

```
func get(stub shim.ChaincodeStubInterface, args []string) (string, error) {
    if len(args) != 1 {
        return "", fmt.Errorf("Falsche Argumente. Schlüssel erwartet.")
    }
    value, err := stub.GetState(args[0])
    if err != nil {
        return "", fmt.Errorf("Datensatz konnte nicht gefunden werden: %s with error: %s", args[0], err)
    }
    if value == nil {
        return "", fmt.Errorf("Datensatz nicht vorhanden: %s", args[0])
    }
    return string(value), nil
}
```

Die Get-methode wird aufgerufen um einen Wert für einen bestimmten Schlüssel zurückzukekommen. Wenn es diesen Schlüssel nicht gibt, wird ein Error zurückgegeben. Wenn der Schlüssel vorhanden ist wird mittels GetState-methode die Abfrage an den World-State für den bestimmten Schlüssel abgeschickt. Falls der Schlüssel sich *noch* nicht auf dem Ledger befindet, wird wieder ein Error zurückgegeben. Vielleicht wurde in diesem Fall die Transaktion noch nicht erfolgreich abgeschlossen. Im letzten Fall wird einfach der entsprechende Wert für den speziellen Schlüssel zurückgegeben.

Die folgenden Zeilen zeigen noch die Main-Funktion, welche die Start-Funktion aufruft. Beim Starten der Instanz startet die Main-Funktion automatisch den Chaincode:

```
func main() {
    err := shim.Start(new(FlurstueckChaincode))
    if err != nil {
        fmt.Println("Konnte FlurstueckChaincode nicht starten")
    } else {
        fmt.Println("FlurstueckChaincode erfolgreich gestartet")
    }
}
```

Sind die grundlegenden Befehle definiert. Hierbei wurden die Standard-Zugriffsmethoden des Hyperledger Fabric-Frameworks verwendet und für die eigene Applikation angepasst. Der Großteil des Codes ist über GitHub frei verfügbar (vgl. Hyperledger Fabric 2019).

Im nächsten Schritt, und dieser ist für unsere Applikation essentiell, muss die Objektklasse definiert werden bzw. die Informationen, welche später mittels API übergeben werden.

## 4.4. DATENMODELL

Unser eingeführtes Datenmodell (s. Kapitel 3.5) hat sieben Eigenschaften. Als Struktur wird die json-Library verwendet. Hier wird der Datensatz für den Ledger definiert.

```
type Flurstueck struct {
    gmlid integer `json:"gmlid"`
    flskz string `json:"flskz"`
    amtlflaeche number `json:"amtlflaeche"`
    istgueltig integer `json:"istgueltig"`
    eigentuemer string `json:"eigentuemer"`
    zeitstempel string `json:"zeitstempel"`
    version integer `json:"version"`
}
```

## 4.5. METHODEN

### 4.5.1. INVOKE

Wie bereits beschrieben ist die Invoke-Methode für neu eingebrachte Transaktionen (Proposals) der Anwender verantwortlich. Innerhalb dieser Methode wurden verschiedene Transaktionstypen definiert. Die Anzahl dieser Typen hängt von den unterschiedlichen Aufgaben ab, die vorgenommen werden können sollen.

Im vorliegenden Anwendungsbeispiel werden folgende Typen definiert:

- recordFlurstueck
- queryFlurstueck
- queryAllFlurstueck
- changeFlurstueckEigentuemer

Die einzelnen Typen können im Prinzip auch beliebig aneinander gereiht werden um somit weitere Aufgaben zu lösen. Z.B. kann ein changeFlurstueckEigentuemer auf ein queryFlurstueck erfolgen, da das Flurstück zunächst ermittelt werden muss, bevor die Änderung des Eigentümers von statten gehen kann.

Diese Typen werden im Verlauf der nächsten Kapitel detailliert vorgestellt. Die Liste könnte auch im bestehenden Anwendungsfall erweitert werden, z.B. um:

- changeVersion
- changeGueltigkeit
- deleteFlurstueck

Die folgenden Kapitel und Abbildungen zeigen die verschiedenen Typen in Abhängigkeit zu den verschiedenen Anwendungsfällen.

Die Invoke-Methode an sich besteht aus mehreren Sektionen. Zunächst werden Parameter zur Methode geprüft, um zu wissen, welche Funktion aufgerufen werden soll.

```
func (s *SmartContract) Invoke(APIstub shim.ChaincodeStubInterface) sc.Response {
    // Retrieve the requested Smart Contract function and arguments
    function, args := APIstub.GetFunctionAndParameters()
    // Route to the appropriate handler function to interact with the ledger
    appropriately
    if function == "queryFlurstueck" {
        return s.queryFlurstueck(APIstub, args)
    } else if function == "initLedger" {
        return s.initLedger(APIstub)
    } else if function == "recordFlurstueck" {
        return s.recordFlurstueck(APIstub, args)
    } else if function == "queryAllFlurstueck" {
        return s.queryAllFlurstueck(APIstub)
    } else if function == "changeFlurstueckEigentuemmer" {
        return s.changeFlurstueckEigentuemmer(APIstub, args)
    }
    return shim.Error("Ungueltige Chaincode-Funktion!")
}
```

Hier können alle benötigten Funktionen eingebettet werden, um Sie später mittels API aufrufen und in verschiedenen Anwendungen integrieren zu können.

#### 4.5.2. RECORDFLURSTUECK

Zu Beginn steht die Erzeugung eines neuen Datensatzes, unserem Fall mit recordFlurstueck. Diese Methode wird zu Beginn der Blockchain verwendet, um Datensätze aus anderen Datenbanken zu importieren oder später auch in Spezialfällen wie z.B. Grundstücksteilung Anwendung finden. Abbildung 22 zeigt, wie ein Teilnehmer, hier das LGL Baden-Württemberg, mittels der Invoke-Methode recordFlurstueck einen neuen Datensatz auf die Blockchain schreibt.

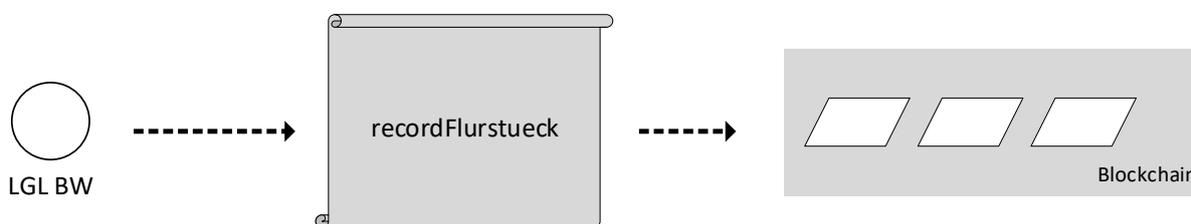


Abbildung 22: Invoke – recordFlurstueck

Um ein Flurstück zu erfassen, sind in diesem Beispiel sieben Argumente notwendig, genauer gesagt die Eingabe aller Attribute wird vorausgesetzt.

```

func (s *SmartContract) recordFlurstueck(APIstub shim.ChaincodeStubInterface, args
[]string) sc.Response {
    if len(args) != 7 {
        return shim.Error("Ungültige Anzahl Argumente. Es werden 7 erwartet!")
    }
    var Flurstueck = Flurstueck {gmlid: args[1], flskz: args[2], amtlflaeche:
args[3], istgueltig: args[4], eigentuemer: args[5], zeitstempel: args[6],
version: args[7]}
    FlurstueckAsBytes, _ := json.Marshal(Flurstueck)
    err := APIstub.PutState(args[0], FlurstueckAsBytes)
    if err != nil {
        return shim.Error(fmt.Sprintf("Flurstueck erzeugen fehgeschlagen: %s",
args[0]))
    }
    return shim.Success(nil)
}

```

### 4.5.3. INITLEDGER

Der Befehl `recordFlurstueck` wird im fortlaufenden Betrieb der Blockchain verwendet. Zu Beginn müssen einmal Daten importiert werden. Zu diesem Zweck wird der Befehl `initLedger` verwendet. Die ersten Datensätze zum Testen wurden aus selbst erstellten, ALKIS-ähnlichen Testdaten erzeugt:

```

func (s *SmartContract) initLedger(APIstub shim.ChaincodeStubInterface) sc.Response {
    flurstueck := []Flurstueck{
        Flurstueck{gmlid: "DEBWL0112405APbf", flskz: „082855-001-04105/0000“,
amtlflaeche: "3343", istgueltig: „1“, eigentuemer: "Müller, Werner",
zeitstempel: "1999-12-31"},
        Flurstueck{gmlid: "DEBWL0112405APGI", flskz: „082855-001-04105/0000“,
amtlflaeche: "3150", istgueltig: „1“, eigentuemer: "Müller, Thomas",
zeitstempel: "1999-12-31"},
        Flurstueck{gmlid: "DEBWL0112405APJa", flskz: „082855-001-04037/0000“,
amtlflaeche: "653", istgueltig: „1“, eigentuemer: "Maier, Moritz",
zeitstempel: "1999-12-31"},
        Flurstueck{gmlid: "DEBWL0112405APJd", flskz: „082855-001-03952/0000“,
amtlflaeche: "2465", istgueltig: „1“, eigentuemer: "Bauer, Heinz",
zeitstempel: "1999-12-31"},
        Flurstueck{gmlid: "DEBWL0112405APJE", flskz: „082855-001-04106/0000“,
amtlflaeche: "346", istgueltig: „1“, eigentuemer: "Bauer, Heinz",
zeitstempel: "1999-12-31"},
        Flurstueck{gmlid: "DEBWL0112405APIz", flskz: „082855-001-03952/0000“,
amtlflaeche: "16454", istgueltig: „1“, eigentuemer: "Bauer, Irmgard",
zeitstempel: "1999-12-31"},
    }
    i := 0
    for i < len(Flurstueck) {
        fmt.Println("i is ", i)
        FlurstueckAsBytes, _ := json.Marshal(Flurstueck[i])
        APIstub.PutState(strconv.Itoa(i+1), FlurstueckAsBytes)
        fmt.Println("Hinzugefügt!", Flurstueck[i])
        i = i + 1
    }
    return shim.Success(nil)
}

```

### 4.5.4. QUERYFLURSTUECK

In einem weiteren Schritt sehen wir uns den Fall einer Abfrage an. Mit `queryFlurstueck` kann ein Teilnehmer, z.B. ein Mitarbeiter in einer Kommunalverwaltung prüfen, wer der Eigentümer eines speziellen Flurstücks ist. Die `Invoke`-methode ruft den definierten Typ `queryFlurstueck` auf, der `chaincode` prüft den Zustand der Blockchain und gibt bei erfolgreicher Abfrage den Wert (hier den Namen des Eigentümers) zurück (s. Abbildung 23).

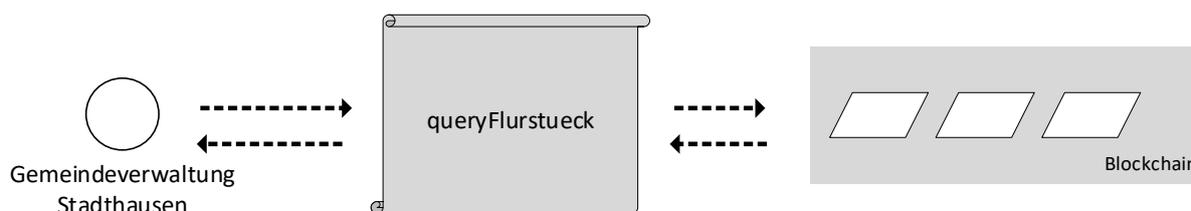


Abbildung 23: Invoke – queryFlurstueck

```
func (s *SmartContract) queryFlurstueck(APIStub shim.ChaincodeStubInterface, args
[]string) sc.Response {
    if len(args) != 1 {
        return shim.Error("Falsche Anzahl an Argumenten! 1 erwartet!")
    }
    FlurstueckAsBytes, _ := APIStub.GetState(args[0])
    if FlurstueckAsBytes == nil {
        return shim.Error("Flurstueck kann nicht gefunden werden!")
    }
    return shim.Success(FlurstueckAsBytes)
}
```

#### 4.5.5. QUERYALLFLURSTUECK

Teilnehmer mit den entsprechenden Berechtigungen haben natürlich auch die Möglichkeit auf einen Gesamtdatenbestand zuzugreifen. Das könnte beispielsweise das statistische Landesamt sein, damit ermittelt werden kann, wie sich die verschiedenen Nutzungen der Flurstücke in Baden-Württemberg verteilen. Das entsprechende queryAllFlurstueck-Statement gibt ohne Einschränkung alle Daten zurück, ähnlich Abbildung 23.

```
func (s *SmartContract) queryAllFlurstueck(APIStub shim.ChaincodeStubInterface)
sc.Response {
    startKey := "0"
    endKey := "999"
    resultsIterator, err := APIStub.GetStateByRange(startKey, endKey)
    if err != nil {
        return shim.Error(err.Error())
    }
    defer resultsIterator.Close()
    // buffer is a JSON array containing QueryResults
    var buffer bytes.Buffer
    buffer.WriteString("[")
    bArrayMemberAlreadyWritten := false
    for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()
        if err != nil {
            return shim.Error(err.Error())
        }
        // Add a comma before array members, suppress it for the first array
        member
        if bArrayMemberAlreadyWritten == true {
            buffer.WriteString(",")
        }
        buffer.WriteString("{\"Key\":")
        buffer.WriteString("\"")
        buffer.WriteString(queryResponse.Key)
        buffer.WriteString("\"")
        buffer.WriteString(", \"Record\":")
        // Record is a JSON object, so we write as-is
        buffer.WriteString(string(queryResponse.Value))
        buffer.WriteString("}")
        bArrayMemberAlreadyWritten = true
    }
    buffer.WriteString("]")
    fmt.Printf("- queryAllFlurstueck:\n%s\n", buffer.String())
    return shim.Success(buffer.Bytes())
}
```

#### 4.5.6. CHANGEFLURSTUECKEIGENTUEMER

Die Abfragen können und müssen teilweise für die Erfüllung der Aufgaben kombiniert werden. Das erwähnte Beispiel der Eigentümeränderung ist ein solcher Fall. Der Prozess der Eigentümeränderung ist überschaubar.

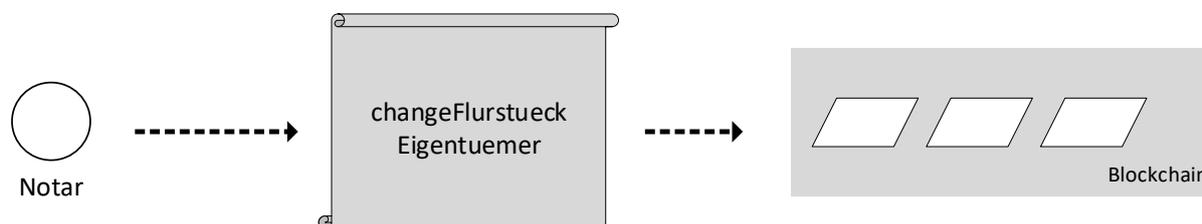


Abbildung 24: Invoke – changeFlurstueckEigentuemer

Ein Notar, welcher gerade den Kaufvertrag zweier Personen legitimiert hat, gibt die neuen Informationen in die Applikation ein, es wird changeFlurstueckEigentuemer aufgerufen, damit der Datensatz auf der Blockchain geändert wird und somit eine neue Version des Flurstückes entsteht (vgl. Abbildung 24).

changeFlurstueckEigentuemer benötigt 2 Argumente: gmlid um das Flurstück zu identifizieren und den Namen des Eigentümers (eigentuemer), um die Eintragung vorzunehmen:

```
func (s *SmartContract) changeFlurstueckEigentuemer(APIstub
shim.ChaincodeStubInterface, args []string) sc.Response {
    if len(args) != 2 {
        return shim.Error("Falsche Anzahl Argumente. 2 werden erwartet.")
    }
    FlurstueckAsBytes, _ := APIstub.GetState(args[0])
    if FlurstueckAsBytes != nil {
        return shim.Error("Flurstück nicht gefunden!")
    }
    flurstueck := Flurstueck{}
    json.Unmarshal(FlurstueckAsBytes, &flurstueck)
    // Normally check that the specified argument is a valid Eigentuemmer of
    // Flurstueck but here we are skipping this check for this example.
    flurstueck.Eigentuemer = args[1]
    FlurstueckAsBytes, _ = json.Marshal(flurstueck)
    err := APIstub.PutState(args[0], FlurstueckAsBytes)
    if err != nil {
        return shim.Error(fmt.Sprintf("Eigentuemmer konnte nicht geändert werden:
%s", args[0]))
    }
    return shim.Success(nil)
}
```

#### 4.5.7. KOMBINATIONEN

Zuvor muss der Notar das Flurstück jedoch ausfindig machen. Der Verkäufer kommt mit der Besitzurkunde auf ihn zu, somit kann er das Flurstück im System ermitteln und im Anschluss wie oben gezeigt den Eigentümerwechsel vollziehen. Abbildung 25 zeigt die kombinierte Version der zwei Invoke-Statements, zunächst die Abfrage um das Flurstück aufzurufen und im Anschluss die changeFlurstueckEigentuemer-Methode um den Eigentümer zu wechseln.

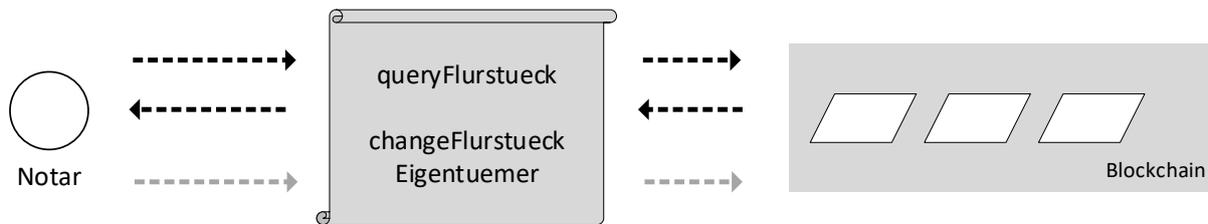


Abbildung 25: Kombination query- und changeEigentuererFlurstueck

## 4.6. AUFBAU DER ANWENDUNG FLURSTUECKAPP

Abbildung 26 zeigt die Dateistruktur der Hyperledger-Applikation.

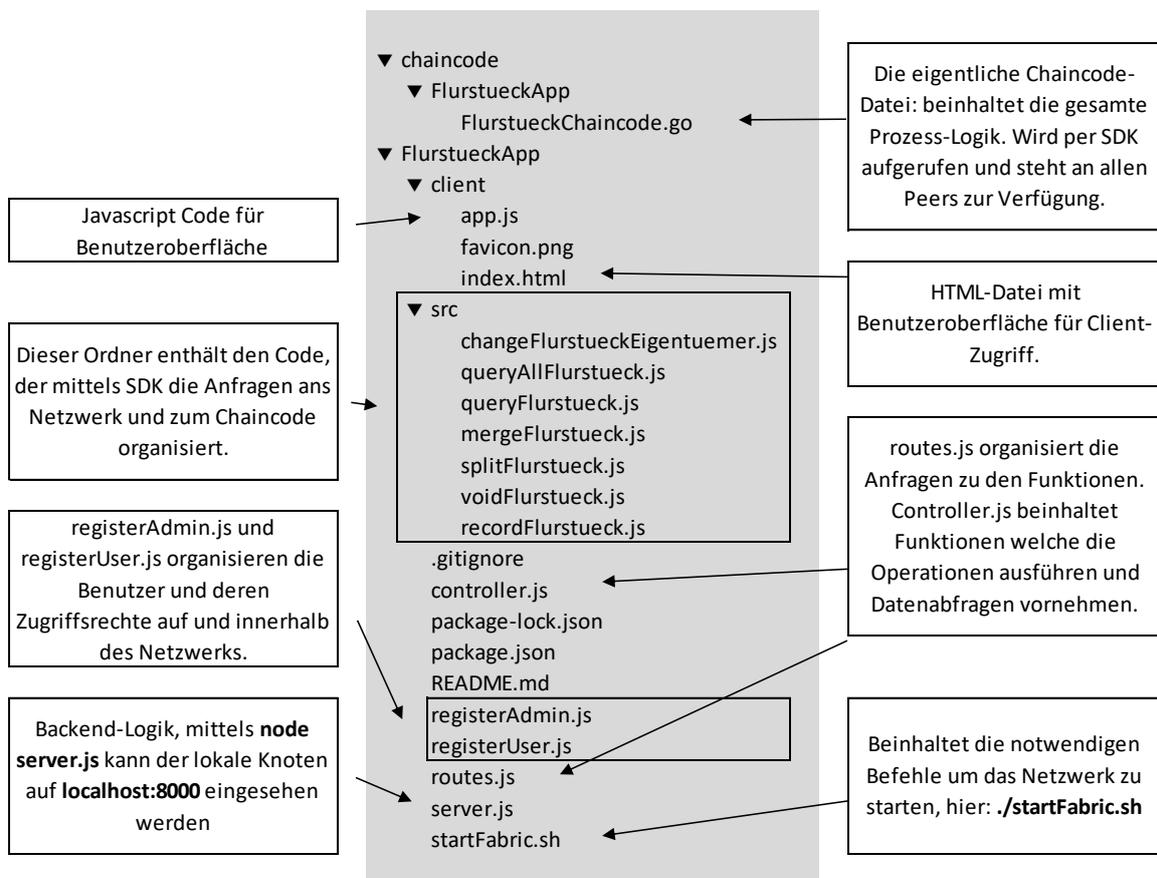


Abbildung 26: Dateistruktur der Anwendung FlurstueckApp (verändert nach EDX 2018a)

## 4.7. BENUTZEROBERFLÄCHE

Eine einfache Möglichkeit, um Zugriff auf eine Blockchain zu gewähren, ist eine Browserbasierte Benutzeroberfläche. Eine solche Benutzeroberfläche kann einfache Aufgaben an der Blockchain ermöglichen, wie z.B. einen Eigentümer ändern oder eine Abfrage nach einer bestimmten Größe, z.B. der amtlichen Fläche eines Flurstücks.

Ebenso können hiermit Listen in Tabellenform angezeigt und exportiert werden. Diese Möglichkeiten genügen einer Vielzahl von Anwendern bereits. Da es sich bei Anwendung um einfache HTML-Skripte in Kombination mit JavaScript handelt, ist auch die Speicherkapazität

beim dezentralen Ausrollen über die Peers kein Problem. Die verschiedenen Oberflächen werden in Kapitel 5 vorgestellt.

## 4.8. SQL FÜR ZUGRIFF AUF AKTUELLEN GEODATENBESTAND

Der Zugriff mittels Geoinformationssystem auf den Inhalt einer Blockchain bringt einige Fragen auf. Zunächst muss klar sein, welche Daten im GIS verwendet werden sollen. Ist bspw. beim Zugriff eines Notars via Browser durchaus die Historie eines Flurstücks von Interesse, ist bei einem Anwendungsfall mittels GIS eher der Bedarf an möglichst aktuellen Daten. Um performant auf die Daten zugreifen zu können wird in diesem Szenario auf einen Datenbestand in einer Geodatenbank zugegriffen (vgl. Abbildung 20, S. 43). In diesem Fall nutzen wir für einen Großteil der Aufgaben eine PostGIS-Datenbank. Mit dem PostGIS-Datenmodell können wir der Mehrheit verfügbarer Geoinformationssysteme Zugriff auf die Daten ermöglichen und viele einfache Aufgaben, wie z.B. Ermittlung eines Eigentümers oder Prüfung eines Flurstücks auf Aktualität können rasch umgesetzt werden.

Durch Triggerfunktionen und Chaincode kann die Datenbank automatisch aktualisiert werden, sobald neue Datensätze oder Änderungen an der Blockchain angehängt werden – damit ist auch eine Aktualität gewährleistet. Die PostGIS-Datenbank ist hierbei immer ein Abbild des Ledgers am Peer. Der Ledger und somit auch der Inhalt einer PostGIS-Datenbank liefern zunächst den gesamten Datenbestand. Am Beispiel des Flurstückes mit der gmlid „DEBWL0112405APBf“ wird das erläutert: Abbildung 27 zeigt alle Versionen des Flurstückes, welches einst von Hans-Jürgen (Version = 1) an seinen Sohn Werner vererbt wurde (Version = 2). Beide Datensätze befinden sich in der Datenbank.

```
SELECT * FROM flurstueck
WHERE gmlid = 'DEBWL0112405APBf';
```

	gmlid character varying (16)	flskz character varying (24)	amtflaeche integer	istgueltig integer	eigentuemer character varying (255)	zeitstempel character varying (255)	version integer
1	DEBWL0112405APBf	082855-001-03875/0000	3234	0	Müller, Hans-Jürgen	1960-06-15	1
2	DEBWL0112405APBf	082855-001-03875/0000	3234	1	Müller, Werner	1999-12-31	2

Abbildung 27: Alle Versionen eines Flurstückes in PostGIS

Bevor die Daten ungefiltert verwendet werden ist es sinnvoll, den gewünschten Datenbestand einzugrenzen. Ist man am aktuellen Datenbestand interessiert, d.h. „Welche Objekte sind jetzt gültig?“, kann man die Daten mit einem einfachen SQL-Statement filtern. Um beim Beispiel des o.g. Flurstückes zu bleiben:

```
SELECT * FROM flurstueck
WHERE gmlid = 'DEBWL0112405APBf'
AND istgueltig = 1;
```

	gmlid character varying (16)	flskz character varying (24)	amtflaeche integer	istgueltig integer	eigentuemer character varying (255)	zeitstempel character varying (255)	version integer
1	DEBWL0112405APBf	082855-001-03875/0000	3234	1	Müller, Werner	1999-12-31	2

Abbildung 28: Gültige Versionen in PostGIS

Die Anpassung der Abfrage für den Gesamtbestand aktueller Flurstücke lautet entsprechend:

```
SELECT * FROM flurstueck
WHERE istgueltig = 1;
```

## 5. ERGEBNISSE

### 5.1. INITIALISIERUNG NEUER DATENSÄTZE

Die Initialisierung neuer Datensätze ist elementar wichtig bei der Einführung der Blockchain, genauer gesagt bei der Erzeugung des Ausgangsbestandes. Sämtliche Datensätze, die später in der Blockchain dokumentiert werden sollen, müssen aus dem Ursprungssystem übergeben (=initialisiert) werden. Zum Abschluss der Initialisierung steht der sog. World State, also der Gesamtzustand aller Verhältnisse.

Ansonsten muss die Initialisierung neuer Datensätze dann wieder vorgenommen werden, wenn neue Flurstücke erzeugt werden – und hier kommt es darauf an, wie man ein neues Flurstück definiert bzw. wann man von einem neuen Flurstück spricht. In der vorliegenden Arbeit ist das etwa bei einer Umlegung der Fall, wenn bspw. Ackerland zu neuem Baugebiet umgelegt wird, denn ab diesem Zeitpunkt sind die Flurstücke neu angeordnet und die bis dahin gültigen Flurstücke werden ungültig. In diesem Fall entsteht die Initialisierung aus der Notwendigkeit der Umlegung, daher der Schaffung neuer Flurstücke bei gleichzeitiger Löschung bestehender Flurstücke. Dieses Beispiel wird im Folgenden Sonderfall 3 genauer erläutert.

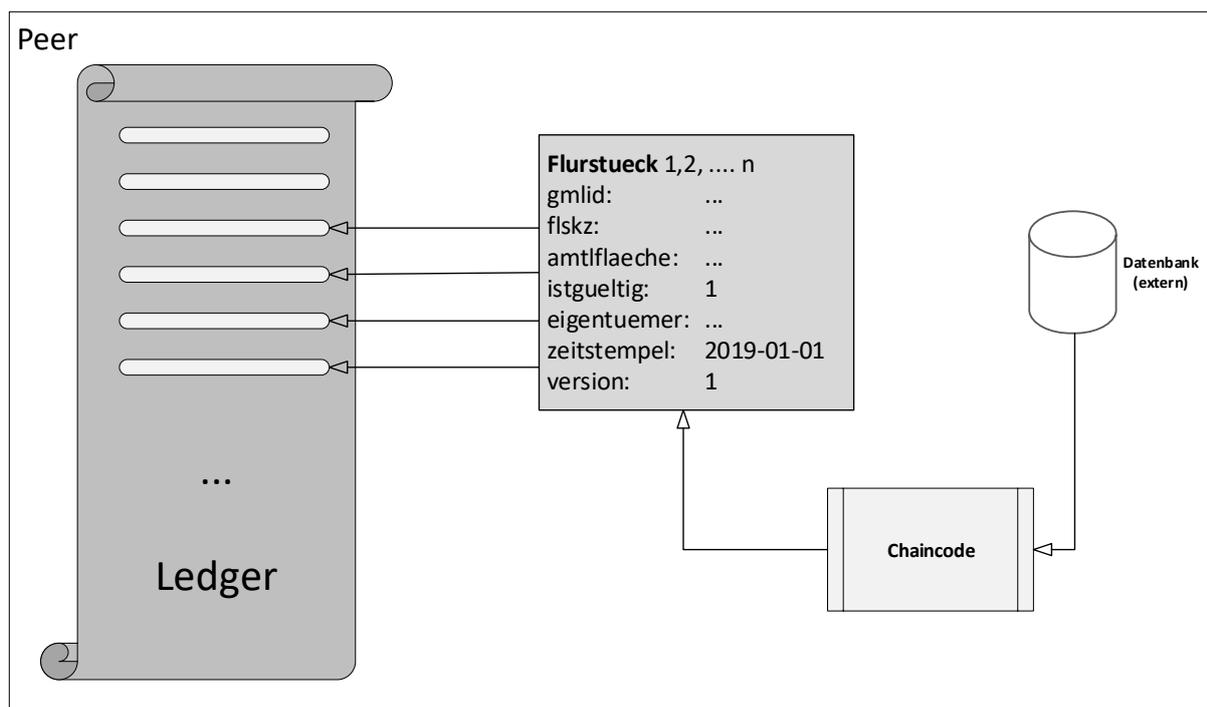


Abbildung 29: Prozessdiagramm Initialisierung neuer Datensätze (Erstinitialisierung)

Abbildung 29 zeigt den Prozess der Erstinitialisierung der Daten. Dabei werden aus der bisherigen Datenbank die gewählten Informationen per Schnittstelle an das Framework übergeben. Per definierten Chaincode werden die Eigenschaften nun auf den Ledger geschrieben und erzeugen somit pro Objekt eine neue Transaktion. Nach Abschluss der Erstinitialisierung steht also der erste World-State, also der erste Gesamtzustand der Blockchain, der als Ausgangsbasis für die Fortführung des Ledgers, also in diesem Fall für die Fortführung des (Teil-)Katasters dient.

An dieser Stelle erfolgt nochmal der Hinweis, dass eine Neubildung eines Flurstücks z.B. durch Umliegung im weiteren Verlauf (s. Kapitel: 5.4.3) detailliert erläutert wird und nicht identisch mit der Erstinitialisierung aller Datensätze ist.

## 5.2. ABFRAGEN DER BLOCKCHAIN

### 5.2.1. GESAMTEN INHALT ABFRAGEN

Die Frage nach einem Gesamtbestand ist vor allem für statistische oder regulatorische Zwecke sinnvoll. Um Hyperledger Fabric nach dem Gesamtbestand zu fragen kann wird `queryAllFlurstueck` eingesetzt:

```
// queryAllFlurstueck - keine Argumente vorausgesetzt
const request = {
  chaincodeId: 'flurstueckChaincode',
  txId: tx_id,
  fcn: 'queryAllFlurstueck',
  args: []
};
return channel.queryByChaincode(request);
```

Das Ergebnis in Tabellenform zeigt alle Objekte in der Blockchain mit sämtlichen Eigenschaften. Um eine Auswahl treffen zu können wird, wie in Kapitel 4.8 vorgestellt, SQL eingesetzt.

fiskz ^	amtflaechen	istgueltig	eigentuemer	zeitstempel	version
082855-001-00010/0002	437	1	Müller, Werner	1999-12-31	1
082855-001-00011/0001	340	1	Müller, Werner	1999-12-31	1
082855-001-00025/0001	481	1	Müller, Werner	1999-12-31	1
082855-001-00026/0000	655	1	Müller, Werner	1999-12-31	1
082855-001-00027/0000	554	1	Müller, Werner	1999-12-31	1
082855-001-00027/0001	295	1	Müller, Werner	1999-12-31	1
082855-001-00028/0000	255	1	Müller, Werner	1999-12-31	1
082855-001-00029/0001	627	1	Müller, Werner	1999-12-31	1
082855-001-00032/0000	491	1	Müller, Werner	1999-12-31	1
082855-001-00033/0000	631	1	Müller, Werner	1999-12-31	1
082855-001-00033/0001	320	1	Müller, Werner	1999-12-31	1
082855-001-00034/0000	388	1	Müller, Werner	1999-12-31	1
082855-001-00034/0001	773	1	Müller, Werner	1999-12-31	1
082855-001-00036/0000	816	1	Müller, Werner	1999-12-31	1
082855-001-00036/0001	132	1	Müller, Werner	1999-12-31	1

Abbildung 30: Abfrageergebnis `queryAllFlurstueck`

### 5.2.2. SPEZIELLE INHALTE ABFRAGEN

Auch spezielle Inhalte müssen abgefragt werden können, bspw. um die Informationen zu einem bestimmten Flurstück zu erhalten. Mit der folgenden Query erhält man alle Datensätze zu einem bestimmten Flurstück, d.h. unter Umständen auch alle Versionen.

```
// queryFlurstueck - benötigt Argument
const request = {
  chaincodeId: 'flurstueckChaincode' ,
  txId: tx_id,
  fcn: 'queryFlurstueck' ,
  args: ['1']
};
return channel.queryByChaincode(request);
```

Die Sucheingabe kann über eine Maske, die Ausgabe wie bei queryAllFlurstueck über eine Tabelle erfolgen.

Abbildung 31: Eingabemaske queryFlurstueck

Alternativ könnten die Eigenschaften des ausgewählten Flurstückes auch als Formular angezeigt werden.

Abbildung 32: Ergebnisformular FlurstueckEigenschaften

## 5.3. ÄNDERUNG VON EIGENTUMSVERHÄLTNISSEN

Die Änderung von Eigentumsverhältnissen ist in unserem fiktiven Szenario eher eine unspektakuläre Änderung, denn lediglich ein Teil des Datensatzes, ein Attribut, wird geändert – dennoch wird ganz im Sinne der vollständigen Nachvollziehbarkeit aller Transaktionen ein neuer, aktualisierter Datensatz erzeugt.

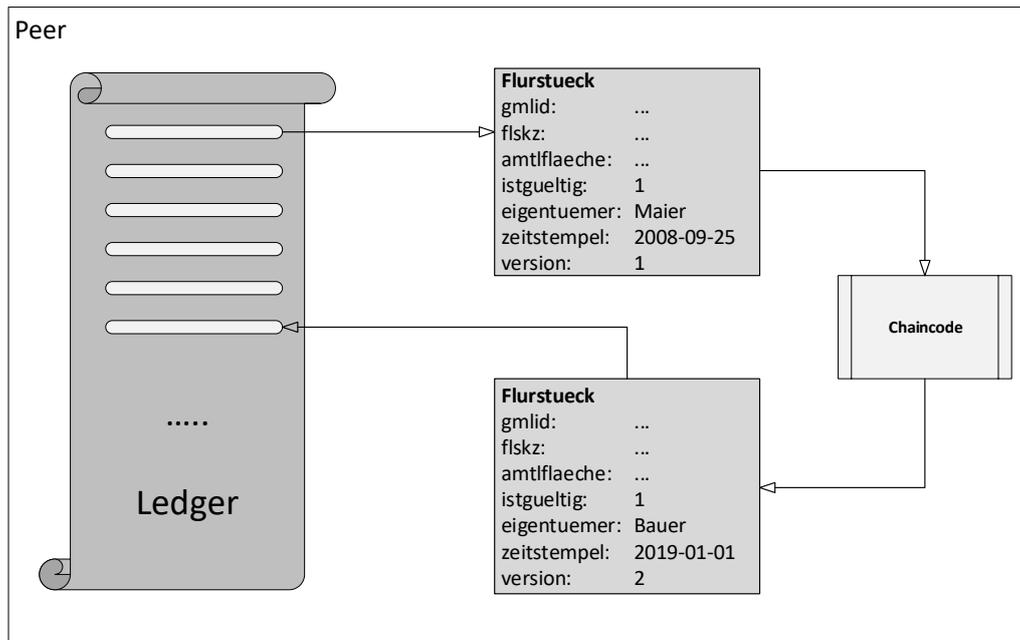


Abbildung 33: Prozessdiagramm Eigentümerwechsel

Abbildung 33 zeigt den Prozess des Eigentümerwechsels: Das betroffene Flurstück wird per Chaincode prozessiert und am Ende des Ledgers mit dem neuen Zustand angehängt. Der neue Zustand wird hierbei durch den Eigentümerwechsel von Maier zu Bauer dargestellt. Der Ledger ist nun in der Lage, den neuen Eigentümer als Information auszugeben.

```
// changeFlurstueckEigentuemer - 2 Argumente nötig
var request = {
  chaincodeId: 'flurstueckChaincode',
  fcn: 'changeFlurstueckEigentuemer',
  args: ['1', 'Bauer'],
  chainId: 'flurstueckChannel',
  tx_id: tx_id
};
return channel.sendTransactionProposal(request);
```

Die Eingabe setzt zwei Argumente voraus: Den Namen des neuen Eigentümers und die eindeutige Zuordnung des Datensatzes, hier wird die Transaktions-ID herangezogen, denkbar wäre aber auch jedes andere Attribut.

Im Anschluss erhält man folgende Meldung:

```
The transaction has been committed on peer localhost:7053
event promise all complete and testing complete

Successfully sent transaction to the orderer.
Successfully sent Proposal and received ProposalResponse: Status - 200, message - "OK",
metadata - "", endorsement signature: 0D 9
```

Daraus kann geschlossen werden, dass das Proposal mittels SDK an den Peer übermittelt, geprüft, validiert und schließlich an den Ledger angehängt wurde.

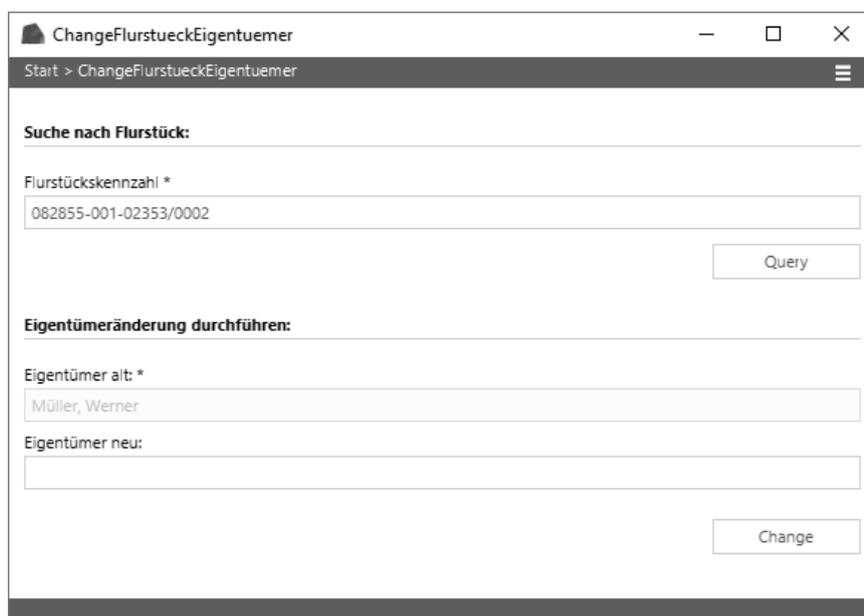


Abbildung 34: Eingabemaske changeFlurstueckEigentuer

## 5.4. SONDERFÄLLE

### 5.4.1. VERSCHMELZUNG VON FLURSTÜCKEN

Bei der Verschmelzung von Flurstücken wird aus zwei oder mehreren Flurstücken ein neues Flurstück gebildet, daher ist das neue Flurstück die Summe der Ausgangsflurstücke. Abbildung 35 zeigt exemplarisch 5 Flurstücke auf, von denen die 3 zentralen Flurstücke vereinigt werden sollen. Dies kommt häufig vor, wenn bspw. Landwirte Flurstücke tauschen, um die Bewirtschaftung der eigenen Felder effizienter zu gestalten.

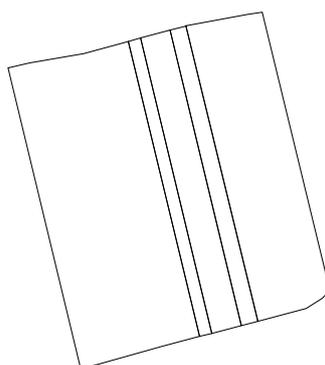


Abbildung 35: Flurstücke vor der Verschmelzung

Aus den 3 zentralen Flurstücken in Abbildung 35 entsteht nun ein neues zentrales Flurstück (vgl. Abbildung 36) welches in Summe die Einzelteile der 3 ursprünglichen Flurstücke abbildet.

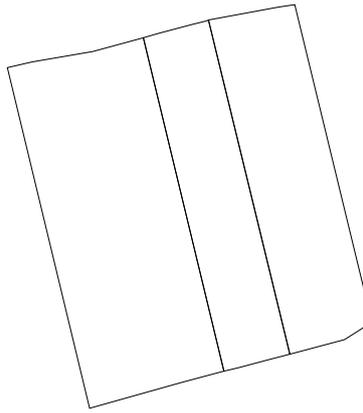


Abbildung 36: Flurstücke nach der Verschmelzung

In diesem Spezialfall wird auch der Prozess innerhalb der Anwendung etwas komplexer: Alle vorherigen Flurstücke müssen im Ledger durch ein neues ersetzt werden. Abbildung 37 zeigt den Prozess anhand zweier Flurstücke, welche mittels Chaincode in ein neues Flurstück prozessiert und an den Ledger angehängt werden. Anhand der Fläche kann beispielhaft der Flächenzuwachs nachvollzogen werden. Dieses Prinzip kann auf beliebig vielen aneinandergrenzenden Flurstücken vollzogen werden. In diesem Fall sollte der Chaincode vorher prüfen, ob die Flurstücke überhaupt topologisch aneinandergrenzen.

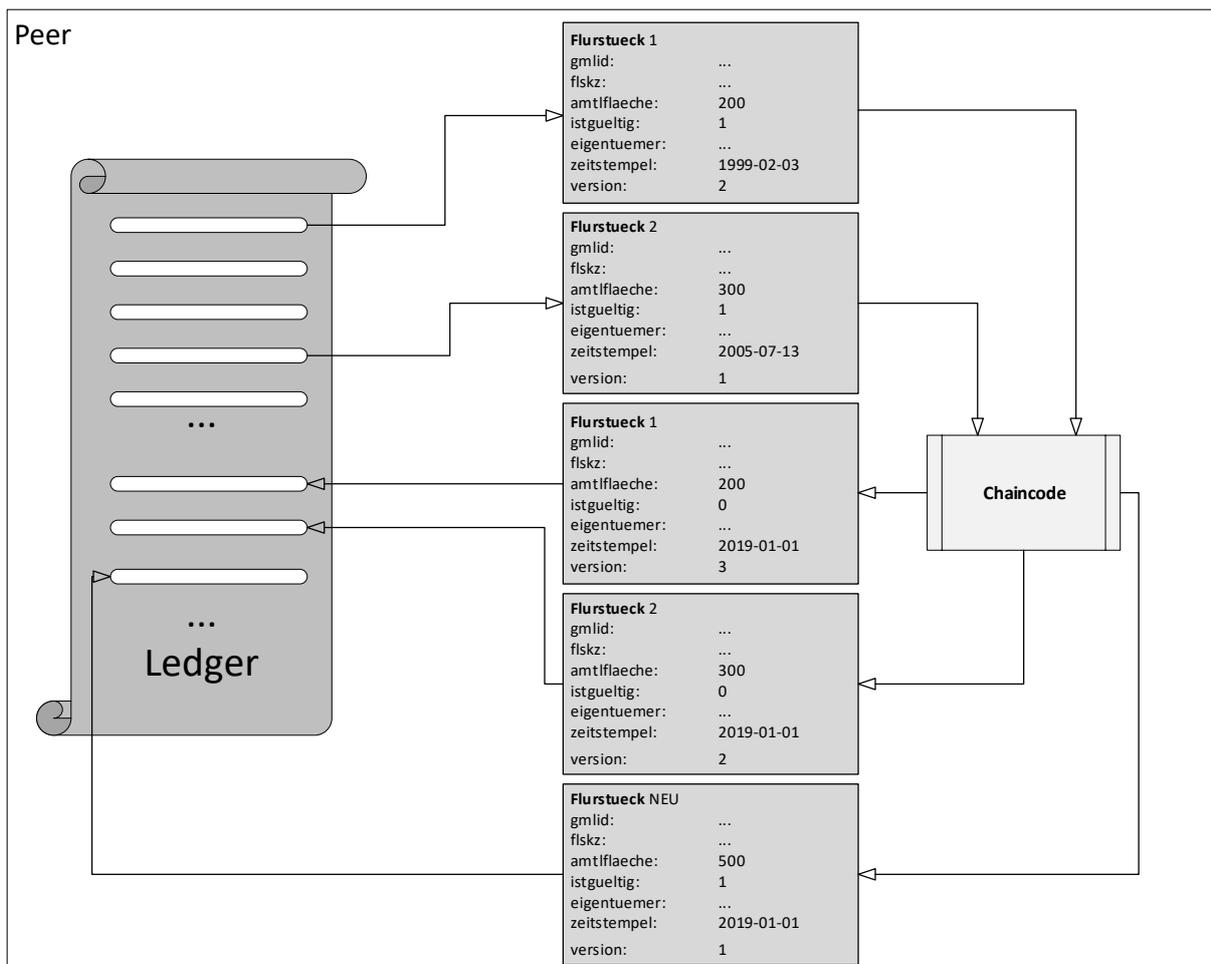
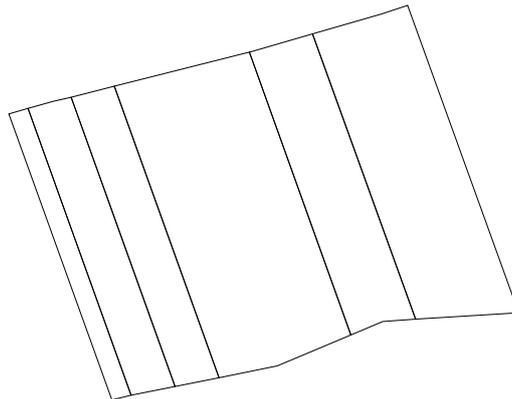


Abbildung 37: Prozessdiagramm Verschmelzung zweier Flurstücke

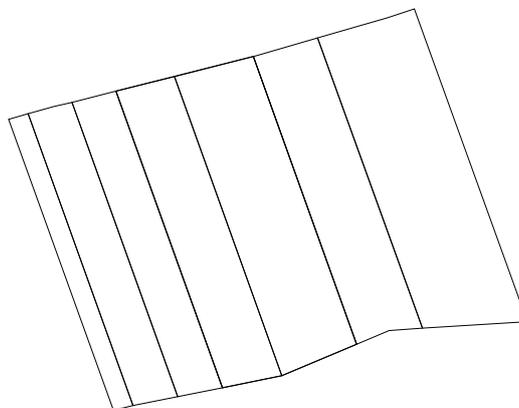
Technisch bedeutet dies, dass zwei bestehende Flurstücke ihre Gültigkeit verlieren und dafür ein neues initiiert werden muss.

#### 5.4.2. TEILUNG VON FLURSTÜCKEN

Technisch ist dieser Fall mit Sonderfall 2 vergleichbar, der Unterschied besteht lediglich darin, dass ein Flurstück seine Gültigkeit verliert und durch zwei neu initiierte Flurstücke abgelöst wird. Die Abbildung 38 zeigt die Flurstücke vor und Abbildung 39 nach der Teilung.



**Abbildung 38: Flurstücke vor der Teilung**



**Abbildung 39: Flurstücke nach der Teilung**

In Abbildung 40 wird der zugehörige Prozess anhand der Teilung eines bestehenden in zwei neue Flurstücke gezeigt. Das alte Flurstück wird in einer neuen Version ungültig und die zwei neuen Flurstücke initiiert. In diesem Fall haben sich Vater und Sohn das Flurstück geteilt, was sich auch an der Gesamtfläche beider Flurstücke widerspiegelt.

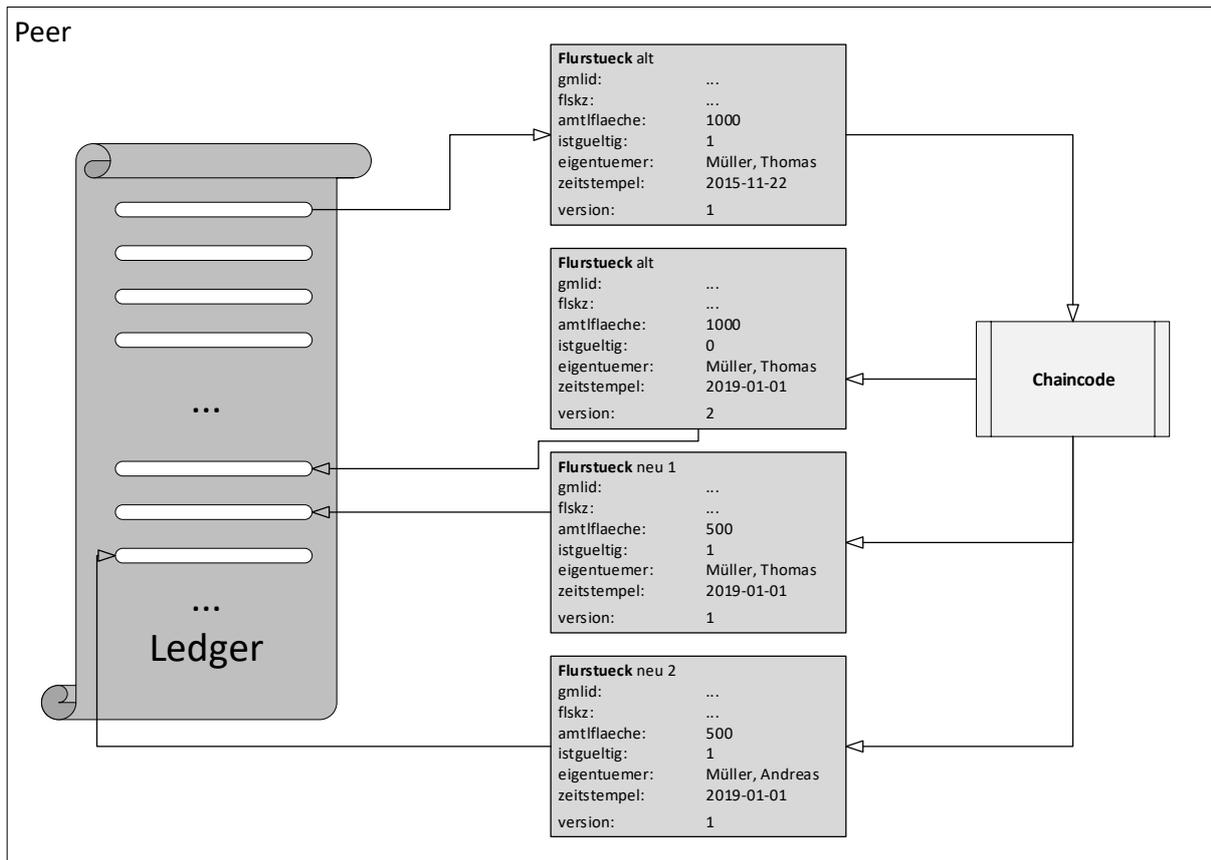


Abbildung 40: Prozessdiagramm Teilung zweier Flurstücke

### 5.4.3. UMLEGUNG VON FLURSTÜCKEN

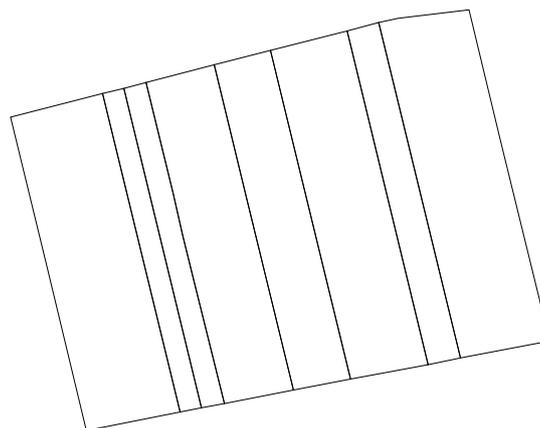


Abbildung 41: Flurstücke vor der Umlegung

Abbildung 41 zeigt die Flurstücke vor, Abbildung 42 nach der Umlegung. Dieser Fall kommt nicht nur bei der Umlegung von Acker- auf Bauland vor, sondern auch bei der Flurbereinigung.

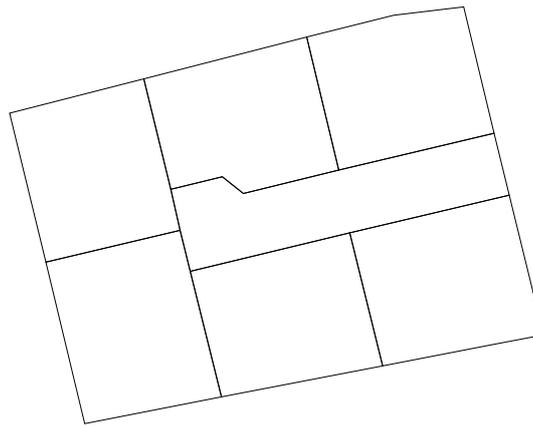


Abbildung 42: Flurstücke nach der Umlegung

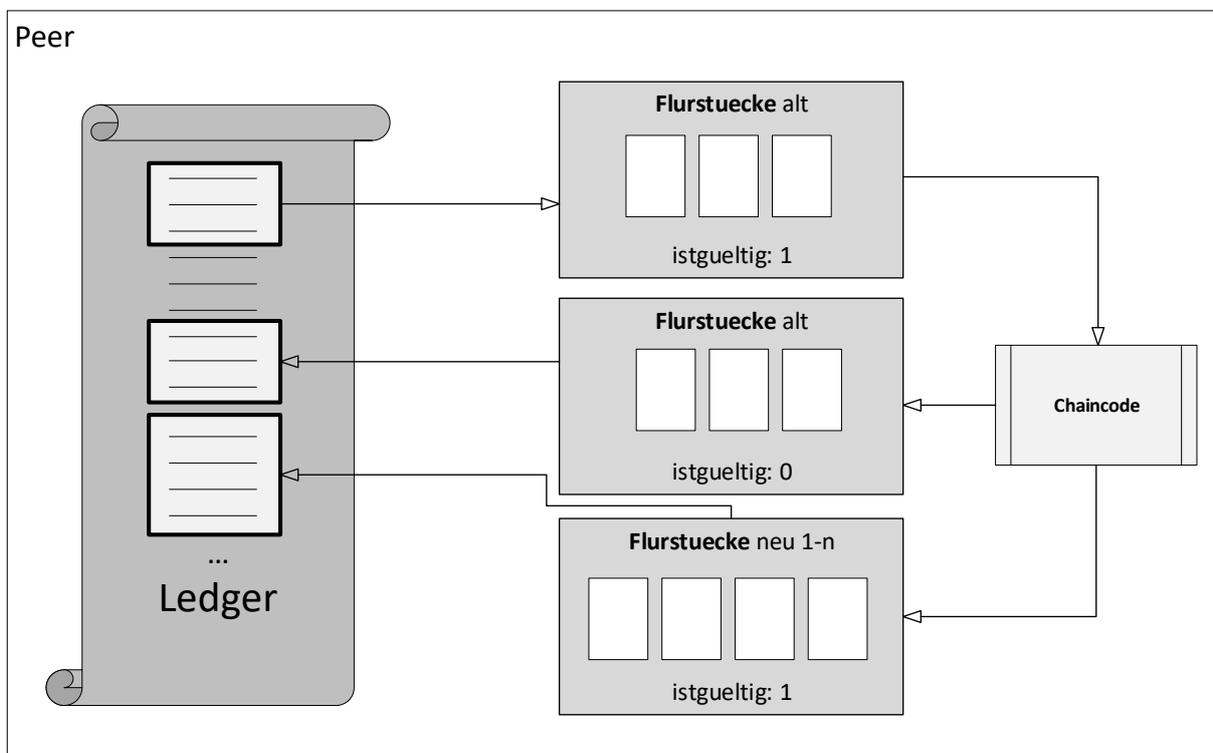


Abbildung 43: Prozessdiagramm Umlegung Flurstücke

Auch dieser Sonderfall ist mit den vorherigen vergleichbar, außer dass es um eine Vielzahl Objekte geht, die durch eine Vielzahl neu initiiertes Objekte abgelöst wird. Die alten Flurstücke werden in einer neuen Version ungültig und die neuen Flurstücke initiiert.

## 5.5. VERWALTUNG VON GEOMETRIEN IN DER BLOCKCHAIN

Hinsichtlich der Verwaltung von Geometrien bzw. räumlicher Daten auf der Blockchain kam der Autor auf kein eindeutiges Ergebnis, da es abhängig vom Anwendungsfall teilweise nicht zwingend notwendig ist, echte Geometrien zu führen.

### 5.5.1. KEINE GEOMETRIEN

Denkbar einfach ist die Variante, überhaupt keine Geometrien in der Blockchain zu führen, sondern nur Metadaten zu einem Datensatz. In diesem Anwendungsfall wurde das an verschiedenen Stellen diskutiert: Um z.B. eine Prüfung auf Aktualität zu starten, müssen in der Blockchain nicht zwingend Geometrien vorhanden sein. Die Daten könnten wie gewohnt in verschiedenen Formaten, hauptsächlich NAS, ausgeliefert und konvertiert werden. Die Prüfung der Aktualität läuft dann auf Basis eindeutiger Eigenschaften der Flurstücke, z.B.:

- Amtliche Fläche
- gmlID
- Eigentümer

Die Blockchain-Applikation kann dann antworten, ob das Flurstück noch existent ist oder nicht.

### 5.5.2. KOORDINATEN ALS WKT

Der zweite Fall wäre eine Speicherung der Koordinaten als WKT (Well-Known-Text). Hier könnte ein Textfeld die Koordinaten führen, welche bei Bedarf in eine Geometrie überführt werden, z.B. durch Geokodierung oder Konvertierungsprogramme.

Denkbar wäre in diesem Fall auch, einen Konverter direkt in den Chaincode zu integrieren, sodass am Peer beim entsprechenden Anwender die gewünschten Daten – bei vorhandener Berechtigung – direkt in ein bestimmtes Format, z.B. GML oder Geopackage konvertiert werden.

### 5.5.3. GEOJSON

Bei der Arbeit mit dem Hyperledger Framework wurde schnell ersichtlich, das Hyperledger Fabric auf JSON-Bibliotheken zugreift. Eine Erweiterung auf GeoJSON sollte auch aufgrund der vorhandenen Standards für das Format kein Problem darstellen.

Die Hyperledger SDK könnte also direkt GeoJSON zurückgeben, welche in einem GIS direkt dargestellt werden können. Mit diesem Ansatz könnten auch Privatpersonen detaillierte Flurstückinformationen z.B. in Google Earth visualisieren.

## 5.6. ERFÜLLUNG DER ANFORDERUNGEN

Alle grundlegenden Anforderungen an die Änderungen innerhalb eines Datensatzes der zur Flurstücksverwaltung verwendet wird, werden erfüllt.

Neben grundlegenden Eigenschaften eines Flurstückes, wie dem Besitzverhältnis, werden auch die in den Absätzen 5.4 bis 5.4.3 aufgeführten Sonderfälle berücksichtigt und umgesetzt. Die anfängliche Sorge, wie die Sonderfälle umgesetzt werden könnten, erwies sich als grundlos, da die Sonderfälle durch den Prozess der „Löschung“ bzw. Neuinitialisierung der Objekte in der Blockchain genauso mühelos abgebildet werden können, wie einfache Änderungen am Besitz.

Interessant ist auch, dass die Handhabung der Sonderfälle an sich seitens Chaincode identisch behandelt werden, der einzige Unterschied besteht in der Anzahl Flurstücke, welche ihre Gültigkeit verlieren bzw. neu initiiert werden (1:N, N:1, N:M).

Durch die mitgelieferte SDK von Hyperledger Fabric lassen sich beliebige technische Zugriffe einrichten, sei es ein Browser-basiertes Userinterface wie in dieser Arbeit, oder ein Zugriff über eine GIS-Software, wie in Kapitel 4.8 skizziert.

## 6. DISKUSSION

### 6.1. BEANTWORTUNG DER FORSCHUNGSFRAGE

Die Arbeit „Distributed Ledger Technologien in der Liegenschaftsverwaltung - Mehrwert in der Verwaltung von Liegenschaften mittels Blockchaintechnologie“ untersuchte ein Szenario, welches die Liegenschaftsverwaltung durch den Einsatz von Blockchaintechnologie künftig weg von einer zentralen Datenhaltung hin zu einer dezentralen Datenhaltung unterstützen könnte. Im Detail beschäftigt sie sich mit drei Teil-Forschungsfragen, welche an dieser Stelle beantwortet werden.

#### 1. *Entsprechen klassische zentrale Datenhaltungen im Umfeld der Liegenschaftsverwaltung noch den Anforderungen?*

Angesichts der Diskussion um aktuelle Prozesse beim Datenaustausch von ALKIS-Daten sowie eingesetzte Technologien und Standards im Vergleich zum dezentralen Szenario kommt der Autor zu dem Schluss, die Frage noch nicht eindeutig klären zu können – im Zweifelsfall jedoch kritisch zurückhaltend die Frage mit ja, klassische zentrale Datenhaltung entspricht (noch) den Anforderungen der Liegenschaftsverwaltung, zu beantworten.

Auch wenn mit der vorliegenden Arbeit dem Einsatz von Distributed Ledger Technologien grundsätzlich positiv gegenüber gestanden werden kann, bedeutet das im Moment nicht, dass der Einsatz einer Geoblockchain in der Liegenschaftsverwaltung durch die aktuellen Anforderungen notwendig ist.

Erklärbar ist das durch die Tatsache, dass die wesentlichen Anforderungen, die auch an eine Blockchain in diesem Szenario gestellt wurden, aus technischer Sicht mit den vorhandenen Mitteln wie Datenexport oder Bereitstellung der Daten als OGC-konforme Webdienste bedient werden können. Aus diesen Gründen kommt der Autor zu dem Schluss, dass die aktuell zentrale Datenhaltung den Anforderungen grundsätzlich (noch) entspricht.

Mit Verweis auf die fortlaufende Digitalisierung von Prozessen geht der Autor allerdings davon aus, dass sich die Anforderungen in den kommenden 3 bis 5 Jahren durchaus dahingehend entwickeln, entweder die zentrale Datenhaltung technologisch zu erweitern oder die Datenhaltung gleich auf dezentrale Technologien umzustellen – möglich wäre natürlich auch eine Kombination. Die Möglichkeiten, welche Blockchain-Frameworks wie Hyperledger hinsichtlich Aktualität und Dokumentation bieten, wären gerade für die Liegenschaftsverwaltung ein nützliche Weiterentwicklung.

#### 2. *Kann Blockchaintechnologie generell für eine Verbesserung der Aktualität von Liegenschaftsdaten sorgen?*

Diese Frage kann der Autor mit einem klaren Ja beantworten. Eine Blockchain bietet hinreichend Möglichkeiten, um weg von einer zentralen Datenhaltung und –pflege hin zu einer verteilten Fortführung der Daten zu gelangen, egal ob es sich hierbei um katasterführende Städte oder die Vermessung eines Flurstückes durch einen öffentlich bestellten Vermessungsingenieur handelt – durch entsprechende Rollenkonzepte und Berechtigungen können die Daten von unterschiedlichsten Stellen eingepflegt und aktualisiert werden. Selbst von einer Kontrollinstanz, die aktuell z.B. durch ein Landesamt repräsentiert wird und die Daten beim Einpflegen auf Richtigkeit prüft, müsste im Grunde

nicht abgewichen werden – allerdings könnten Prüfungen, Kontrollen und Validierungen zum großen Teil durch automatisierten Chaincode und Smart-Contracts umgesetzt werden, wodurch der personelle Verwaltungsaufwand auf ein Minimum reduziert werden könnte. Die grundsätzlich positive Beantwortung der Frage resultiert also maßgeblich auf den technischen Möglichkeiten der Distributed Ledger Technologien. Ob der Einsatz im Einzelfall für alle Anwendungsfälle sinnvoll ist, ist damit nicht gesagt. Der Einsatz lohnt sicherlich immer erst ab dem Erreichen einer bestimmten Anzahl von Kriterien, die im Einzelfall sicherlich untersucht werden müssen.

*3. Ist die Nutzung von Blockchaintechnologie im Umfeld amtlicher Geodaten überhaupt realistisch?*

Diese Frage möchte der Autor in zweierlei Hinsicht beantworten: Zum einen zeigt die vorliegende Arbeit, dass der Einsatz von Blockchains bereits in verschiedenen Ländern in Bezug auf Landverwaltung bereits eingesetzt wird und dies aller Voraussicht nach auch in weiteren Ländern, welche den Einsatz aktuell prüfen bzw. darüber forschen, der Fall sein wird. Eine Nutzung ist daher auch im Umfeld deutscher Liegenschaftsdaten durchaus realistisch.

Auf der anderen Seite haben wir mit ALKIS einen bundesweit etablierten Datenbestand, in den meisten Bundesländern vielfältige Angebote zu Datenbezug und – Bereitstellung sowie akzeptierte Austauschformate. Wie auch in Frage 1 aufgezeigt, ist hier anzunehmen, dass ein Einsatz nicht zeitnah in den kommenden Jahren erfolgt, sondern eher mittelfristig untersucht und anschließend ggf. schrittweise eingeführt wird.

Aber gerade auch die Möglichkeit, nicht alle Daten in den Ledger aufzunehmen zu müssen, sondern zu Beginn nur bestimmte Anwendungen oder einfache Prüfroutinen über die Blockchain laufen zu lassen, machen einen Einsatz grundsätzlich interessant. Der Autor hatte zu Beginn der Arbeit die Sorge, dass der Einsatz einer Geoblockchain nur hochspezialisierten Unternehmen oder einigen wenigen speziellen Anwendungsfällen vorbehalten sein wird. Die vorliegende Arbeit hat die vorliegenden Zweifel aber ausgeräumt. Blockchaintechnologie ist definitiv einsatzfähig und bereits an einem weit gereiften Entwicklungsstadium angekommen, wenngleich dieser in wirklich sehr kurzer Zeit erreicht wurde. In der Geoinformatik spricht aktuell nur das Fehlen geeigneter Standards zum nächsten großen Schritt, aber glücklicherweise wird daran bereits gearbeitet.

## 6.2. FAZIT

Unter Berücksichtigung der Erläuterungen kommt der Autor auch zu dem Schluss, dass durch die Verwendung von Distributed Ledger Technologien im Allgemeinen und Blockchaintechnologie im Speziellen definitiv ein Mehrwert auch für die amtliche Liegenschaftsverwaltung geschaffen werden kann.

Junge Technologien bringen allerdings immer auch ein Für und Wider mit sich. Gerade wenn sich wirklich wesentliche Kerneigenschaften eines bestehenden Systems ändern, wie in dieser Arbeit am Vergleich zentraler/dezentraler Datenhaltung aufgezeigt, ist es essentiell wichtig sich differenziert mit den Argumenten auseinander zu setzen und mit Fokus auf den eigentlichen Anwendungsfall ein passendes Urteil bzgl. Einführung oder Nichteinführung zu fällen.

Auch wenn Blockchains bzw. Geoblockchains wie in Kapitel 2.6 erläutert eine Menge Chancen mit sich bringen, dürfen auch die Nachteile nicht aus den Augen verloren werden, gerade wenn es wie bei Kryptowährungen wie Bitcoin um einen enorm hohen Energieverbrauch geht. Aus persönlicher Sicht des Autors dürfen auch technische Neuerungen mit großem Potenzial nicht „um jeden Preis“ angewandt werden.

Das vorliegende Fallbeispiel benötigt diesen enorm hohen Aufwand zum Glück nicht, da der Ledger sicher nicht die Anzahl Transaktionen aufzeichnet, wie es bei einer öffentlichen Kryptowährungsblockchain der Fall ist.

Dennoch muss man auch hier abwägen, ob der Einsatz sinnvoll ist. Die Arbeit zeigt die Vorteile auf: Es entstehen völlig neue Möglichkeiten, die Daten dezentral zu speichern und Schritt für Schritt bestehende Routinen beim Datenexport abzulösen, hier bestünde auch weiterführender Forschungsbedarf, wie in Kapitel 6.3 noch angeregt wird. Auch die mögliche Einbindung verschiedener Stellen oder Akteure, d.h. neben Landesämtern könnten die katasterführenden Städte auch direkt im System den Bestand fortführen, zeigt weitreichendes Potenzial auf, da es mit Städten, Gemeinden, Planungsbüros, Zweckverbänden, öffentlich bestellten Vermessungsingenieuren und zuletzt auch dem Bürger als Flurstückseigentümer eine Menge beteiligter Stellen und Personen gibt.

Die Chancen für die Einführung einer Art Liegenschaftskataster auf Blockchain-Basis sind jedoch sicherlich in Ländern höher, in denen es noch kein funktionierendes Katasterwesen gibt wie in Deutschland, diese Aussage bestätigt auch Prof. Dr. Gilbert Fridgen in einem Interview mit der ZEIT (Clauß 2017).

Vor der Einführung einer Blockchain muss man sich aber in jedem Fall differenziert mit dem Szenario auseinandersetzen, idealerweise mit einer umfassenden Anforderungsanalyse und einem stabilen Proof-of-Concept.

Der Autor hat ebenfalls festgestellt, dass sich einige der in Kapitel 2 aufgezeigten Vorteile der Blockchain auch in diesem Anwendungsfall deutlich hervorzuheben sind, gerade auch hinsichtlich der Frage, ob die Einführung einer Blockchain überhaupt sinnvoll ist. Die wohl eindeutigste Erkenntnis des Autors ist, dass je mehr Akteure an einer Blockchain partizipieren, desto besser und wertschöpfender die Anwendung. Hinzufügen muss man, das neben der reinen Anzahl an Akteuren auch die Anzahl unterschiedlicher Tätigkeiten auf und die unterschiedlichen Anforderungen an die Blockchain eine wichtige Rolle spielen.

Betrachtet man die Fragestellung in dieser Hinsicht detailliert bezogen auf eine mögliche Einführung bei den Landesämtern, ist davon auszugehen, dass auch dort in den kommenden Jahren intensive Untersuchungen vorangetrieben werden und mittelfristig zumindest teilweise mit der Nutzung dieser Technologien gerechnet werden sollte, denn selbstverständlich müssen nicht alle Aufgaben von der Blockchain selbst abgewickelt werden. Auch die Nutzung von Blockchain als System einer eindeutigen Dokumentation/Historisierung ist an sich schon interessant und kann einen Mehrwert aufzeigen.

Diese Arbeit soll einen Beitrag zur Debatte um die Einführung von Distributed Ledger Technologien in Bezug auf räumliche Fragestellungen liefern. In den kommenden Jahren wird sich zeigen, ob sich die Technologie neben den bestehenden Technologien etablieren kann oder wieder von der Bildfläche verschwinden wird. Auf jeden Fall beschäftigt sich das Vermessungswesen in Europa bereits jetzt mit der Blockchaintechnologie, wie auch der Jahresbericht 2017/18 der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder in der Bundesrepublik Deutschland (kurz: AdV) aufzeigt (AdV 2018). Auch das Open Geospatial Consortium hat im aktuellsten „Whitepaper on Land Administration“ vom Februar 2019 die Mehrwerte der Blockchain-Technologie erwähnt (Open Geospatial Consortium 2019).

### 6.3. AUSBLICK

Die hier vorgestellte Arbeit zeigt in sich schon Entwicklungspotenzial weiterer räumlicher Fragestellungen auf Blockchaintechnologien auf.

Das kann neben der Arbeit an der Entwicklung von Standards und Protokollen, vielleicht für die Erzeugung eines Hashs direkt aus einem Geometriefeld auch eine Auseinandersetzung mit der Frage des Vertrauens in eine Blockchain sein, da auf zentrale Instanzen verzichtet werden kann. Diesem Komplex müssen sich aber wohl eher die Soziologen und Psychologen widmen, wenngleich dieser Punkt sicher auch auf die zukünftige Entwicklung, gerade auch hinsichtlich der gesellschaftlichen Akzeptanz oder gar künftigen Forderungen nach dezentralen Technologien, einen enormen Einfluss entwickeln könnte.

Alternativ können auch einige Punkte aus dem vorliegenden Anwendungsfall konkreter untersucht werden, z.B.

- Wie könnten Flurstücke mittels Kryptowährung (Tokens) direkt zwischen Personen gehandelt (verkauft, vertauscht) werden?
- Wie könnte ein Initial Coin Offering aus Sicht der Liegenschaftsverwaltung aussehen? Bei der Initialisierung einer Blockchain muss jedes Flurstück neu erstellt werden – wie kann das technisch umgesetzt werden?
- Einführung diverser Flurstücks-„Token“: Von Verpachtungen über Erbpachtverträge bis hin zu Vermietungen – die meisten Vertragsverhältnisse sind zeitlich befristet – eine Blockchain könnte durch geeigneten Chaincode eine Rückgabe eines Flurstücks von Person A an B nach einer Zeit X automatisch durchführen und dokumentieren.
- Flurstücksbezogene Informationen wie Bodenrichtwerte oder Baulasten könnten ebenfalls eindeutig dokumentiert werden

- ➔ Datenexport aus dem Ledger: Der Ledger könnte alle Eigenschaften eines Objekts speichern und eine datennutzende Stelle könnte mittels einer Schnittstelle direkt aus der Blockchain heraus Geodaten zur weiteren Verarbeitung erzeugen.

Die möglichen Forschungsfragen zeigen durchaus weiterführendes Potenzial der Technologie. Wichtig ist, dass neben reinen „Spielereien“ in den kommenden Jahren weitere positive und mehrwertbringende Anwendungsfälle entwickelt und umgesetzt werden, damit die Technologie sich etablieren kann und auf einer breiten Community steht.

## LITERATURVERZEICHNIS

Adams B. und Tomko M. (2018): Critical Look at Cryptogovernance of the Real World: Challenges for Spatial Representation and Uncertainty on the Blockchain. DOI: 10.4230/LIPIcs.GISCIENCE.2018.18

AdV (2018): Tätigkeitsbericht 2017/2018. Stand: 08/2018.

Arnold A. (2018): Is Blockchain The Answer To A Better Healthcare Industry? URL: <https://www.forbes.com/sites/andrewarnold/2018/08/26/is-blockchain-the-answer-to-a-better-healthcare-industry/#7769ec5675a8>. Letzter Zugriff: 30.12.2018.

Baran P., Boehm S.P., Smith J.W. (1964): On distributed Communications.

BitcoinBlog.de (2016): RWE und slock.it wollen Ethereum für Elektroautos nutzen. URL: <https://bitcoinblog.de/2016/02/26/rwe-und-slock-it-wollen-ethereum-fuer-elektroautos-nutzen/>. Letzter Zugriff: 20.01.2019.

Blockchainwelt (2018): FOAM – Proof of Location Protokoll erklärt. URL: [https://blockchainwelt.de/foam-proof-of-location-protokoll/#Die\\_Schwachstellen\\_von\\_GPS](https://blockchainwelt.de/foam-proof-of-location-protokoll/#Die_Schwachstellen_von_GPS). Letzter Zugriff: 20.01.2019).

Blockchainwelt (2018b): Unterschied: Blockchain versus traditionelle Datenbank. URL: <https://blockchainwelt.de/unterschied-blockchain-versus-traditionelle-datenbank-erklaert/> Letzter Zugriff: 13.02.2019

Boulos K. et al. (2018): Geospatial blockchain: promises, challenges and scenarios in health and healthcare. In: International Journal of Health Geographics 17:25.

CBN Foundation (2018): CLEO Blockchain Network (CBN) Geodata and Geoservices For IoT and Mobility (Whitepaper). URL: <https://docplayer.net/86531927-Cleo-blockchain-network-cbn-geodata-and-geoservices-for-iot-and-mobility.html>. Letzter Zugriff: 20.01.2019.

Chapron G. (2017): The environment needs cryptogovernance. URL: <https://www.nature.com/news/the-environment-needs-cryptogovernance-1.22023>. Letzter Zugriff: 09.09.2018.

Chiu J. & Koepl T. V. (2018): Incentive Compatibiliy on the Blockchain. In: Bank of Canada Staff Working Paper 2018-34.

Christidis K. & Devetsikiotis M. (2016): Blockchains and Smart Contracts for the Internet of Things.

Clauß U. (2017): Diese Technik revolutioniert das Internet. URL: <https://www.welt.de/wissenschaft/article168522567/Diese-Technik-revolutioniert-das-Internet.html>. Letzter Zugriff: 09.02.2017.

Condos J., Sorrell W. H. und Donegan S. L. (2016): Blockchain Technology: Opportunities and Risks. URL: <http://legislature.vermont.gov/assets/Legislative-Reports/blockchain-technology-report-final.pdf>. Letzter Zugriff: 09.09.2018.

ConsultingWhere (2015): Crypto-Cadastre: Is the future of land administration written in the Blockchain? URL: <http://www.consultingwhere.com/blog/crypto-cadastre-is-the-future-of-land-administration-written-in-the-blockchain/>. Letzter Aufruf: 29.08.2018.

EDX (2018a): LINUX Foundation Training Course „Blockchain for Business – An Introduction to Hyperledger Technologies“. URL: <https://www.edx.org/course/blockchain-for-business-an-introduction-to-hyperledger-technologies>. Letzter Zugriff: 20.10.2018.

EDX (2018b): LINUX Foundation Training Course „Blockchain: Understanding Its Uses and Implications“. URL: <https://www.edx.org/course/understanding-blockchain-and-its-implications>. Letzter Zugriff: 20.10.2018.

Foamspace Corp (2018): FOAM Whitepaper. URL: [https://foam.space/publicAssets/FOAM\\_Whitepaper.pdf](https://foam.space/publicAssets/FOAM_Whitepaper.pdf). Letzter Zugriff: 20.01.2019.

Gupta, V. (2017): A Brief History of Blockchain. URL: <https://hbr.org/2017/02/a-brief-history-of-blockchain>, letzter Zugriff: 16.08.2018

Heumüller E. & Richter S. (2018): Das Blockchain-Ökosystem als Analyse-Ansatz. In: *Wirtschaftsinformatik & Management* 3|2018, S. 60-65.

Hyperledger Fabric (2019): GitHub repository. URL: <https://github.com/hyperledger/fabric/> Letzter Zugriff: 16.02.2019.

ISO (2016): ISO/TC 307 – Blockchain and distributed ledger technologies. URL: <https://www.iso.org/committee/6266604.html>. Letzter Zugriff: 27.01.2019.

KAIROS Future et al. (2017): The Land Registry in the blockchain – testbed. URL: [https://chromaway.com/papers/Blockchain\\_Landregistry\\_Report\\_2017.pdf](https://chromaway.com/papers/Blockchain_Landregistry_Report_2017.pdf). Letzter Zugriff: 21.07.2018).

Lemmen C., Knight P., Beentjes B., Vos J., de Boer A., Unger E.M., Dijkstra P. & van der Heide J. (2016), Blockchain-Bitcoin functionality in land administration. In: *Proceedings of FIG working week 2016: recovery from disaster*, 2.-6. Mai 2016 in Christchurch/New Zealand.

LFD271 (2018): LINUX Foundation Training Course „Hyperledger Fabric Fundamentals“. URL: <https://training.linuxfoundation.org/training/hyperledger-fabric-fundamentals-lfd271/>. Letzter Zugriff: 20.10.2018.

Meinel C., Gayvoronskaya T. und Schnjakin M. (2018): Blockchain: Hype oder Innovation. In: *Technische Berichte Nr. 113 des Hasso-Plattner-Instituts für Digital Engineering an der Universität Potsdam*.

Metzger J. (2018): Revision von Distributed Ledger Technologie (DLT) vom 19.02.2018. URL: <https://wirtschaftslexikon.gabler.de/definition/distributed-ledger-technologie-dlt-54410/version-277444>. Letzter Zugriff: 10.02.2019.

Nakamoto S. (2008): Bitcoin: A Peer-to-Peer Electronic Cash System, White Paper. URL: <https://bitcoin.org/bitcoin.pdf>. Letzter Zugriff: 04.08.2018.

Niranjanamurthy M., Nithya B.N., Jagannatha S. (2018): Analysis of Blockchain technology: pros, cons and SWOT. DOI: <https://doi.org/10.1007/s10586-018-2387-5>.

Open Geospatial Consortium (2019): OGC White Paper on Land Administration. URL: <http://docs.opengeospatial.org/wp/18-008r1/18-008r1.html>. Letzter Zugriff: 14.02.2019.

Pohlmann N. (2018): Eine vertrauenswürdige Zusammenarbeit mit Hilfe der Blockchain-Technologie. In: Bartsch M., Frey S. (Hrsg.) Cybersecurity Best Practices. Springer Verlag, Wiesbaden.

Rizzo P. (2016): Sweden Tests Blockchain Smart Contracts for Land Registry. URL: <https://www.coindesk.com/sweden-blockchain-smart-contracts-land-registry>. Letzter Zugriff: 07.01.2019.

Schlatt V., Schweizer A., Urbach N. und Fridgen G. (2016): Blockchain Grundlagen, Anwendungen und Potenziale. Projektgruppe Wirtschaftsinformatik des Fraunhofer-Instituts für Angewandte Informationstechnik FIT.

## TABELLENVERZEICHNIS

Tabelle 1: Übersicht der wichtigsten Konsens-Algorithmen .....	23
Tabelle 2: Unterschiede zwischen Datenbanken und Blockchain .....	25
Tabelle 3: Chancen und Risiken von Blockchain-Systemen (Schlatt et al. 2016) .....	27
Tabelle 4: Anforderungen an die Softwarekomponenten .....	35
Tabelle 5: Anforderungen an den Ledger .....	36
Tabelle 6: Blockchain-Vorteile im Vergleich zu Alternativen (nach Blockchainwelt 2018b) ...	37
Tabelle 7: Pro & Contra zum Einsatz von Blockchain am Beispiel von Hyperledger .....	38
Tabelle 8: Übersicht eingesetzter Softwarekomponenten für Blockchain-Applikation.....	38
Tabelle 9: Übersicht eingesetzter Softwarekomponenten für Geovisualisierung .....	39
Tabelle 10: Übersicht Daten .....	39

# ABBILDUNGSVERZEICHNIS

Abbildung 1: Merkle-Tree (EDX 2018a).....	17
Abbildung 2: Auswirkung manipulierter Transaktionen anhand Merkle-Trees (EDX 2018a) .	17
Abbildung 3: Übersicht Netzwerkarten (verändert nach Baran, Boehm & Smith 1964) .....	19
Abbildung 4: Peer(-to-Peer)-Netzwerk (Meinel et al. 2018).....	20
Abbildung 5: Smart Contracts Flussdiagramm (EDX 2018a) .....	21
Abbildung 6: Peer-to-Peer vs. Client-Server-Netzwerk (Meinel et al. 2018) .....	24
Abbildung 7: Zentrale Datenbanken vs. Blockchain (EDX 2018b) .....	25
Abbildung 8: Das Bitcoin-Prinzip (Meinel et al. 2018) .....	26
Abbildung 9: Export von ALKIS-Daten .....	29
Abbildung 10: Überblick über verschiedene ALKIS-Anwender.....	30
Abbildung 11: Bereitstellung von ALKIS-Daten als WMS.....	30
Abbildung 12: Änderungsmeldung ALKIS.....	31
Abbildung 13: Änderungen im ALKIS bei n Anwendern .....	32
Abbildung 14: Ist-Zustand vs. Soll-Zustand (EDX 2018b) .....	32
Abbildung 15: Datenbestand zentral vs. dezentral.....	33
Abbildung 16: Peer.....	40
Abbildung 17: Architektur von Hyperledger Fabric (LFD 271 2018) .....	41
Abbildung 18: Hyperledger Fabric Workflow (LFD 271 2018) .....	41
Abbildung 19: Prozessdiagramm der Applikation.....	42
Abbildung 20: Einbindung einer PostGIS-Datenbank .....	43
Abbildung 21: Workflow Flurstücks-Applikation .....	43
Abbildung 22: Invoke – recordFlurstueck.....	47
Abbildung 23: Invoke – queryFlurstueck.....	49
Abbildung 24: Invoke – changeFlurstueckEigentuemmer .....	50
Abbildung 25: Kombination query- und changeEigentuemmerFlurstueck .....	51
Abbildung 26: Dateistruktur der Anwendung FlurstueckApp (verändert nach EDX 2018a) ...	51
Abbildung 27: Alle Versionen eines Flurstückes in PostGIS .....	52
Abbildung 28: Gültige Versionen in PostGIS .....	52
Abbildung 29: Prozessdiagramm Initialisierung neuer Datensätze (Erstinitialisierung).....	53
Abbildung 30: Abfrageergebnis queryAllFlurstueck .....	54
Abbildung 31: Eingabemaske queryFlurstueck.....	55
Abbildung 32: Ergebnisformular FlurstueckEigenschaften.....	55
Abbildung 33: Prozessdiagramm Eigentümerwechsel .....	56
Abbildung 34: Eingabemaske changeFlurstueckEigentuemmer .....	57
Abbildung 35: Flurstücke vor der Verschmelzung.....	57
Abbildung 36: Flurstücke nach der Verschmelzung .....	58
Abbildung 37: Prozessdiagramm Verschmelzung zweier Flurstücke .....	58
Abbildung 38: Flurstücke vor der Teilung.....	59
Abbildung 39: Flurstücke nach der Teilung.....	59
Abbildung 40: Prozessdiagramm Teilung zweier Flurstücke .....	60
Abbildung 41: Flurstücke vor der Umlegung .....	60
Abbildung 42: Flurstücke nach der Umlegung .....	61
Abbildung 43: Prozessdiagramm Umlegung Flurstücke.....	61

## ABKÜRZUNGSVERZEICHNIS

AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
API	Application Programming Interface
BaaS	Blockchain as a Service
BFT	Byzantine Fault Tolerance
CA	Certificate Authority
DAPPS	Decentralized Applications
DBMS	Datenbankmanagementsystem
DLT	Distributed Ledger Technologie
GPS	Global Positioning System
HTML	Hypertext Markup Language
IBM	International Business Machines Corporation
IoT	Internet of Things
IoV	Internet of Values
ISO/TC	International Organisation for Standardization/Technical Committee
ISO/NP TS	New Proposal/Technical Specification
LBS	Location Based Services
LGL (BW)	Landesamt für Geoinformation und Landentwicklung (Baden-Württemberg)
LoRa	auch LoRaWAN: Long Range Wide Area Network
MSP	Membership Service Provider
NAS	Normbasierte Austauschschnittstelle
ÖbVI	Öffentlich bestellter Vermessungsingenieur
OGC	Open Geospatial Consortium
PoA	Proof of Authority
PoET	Proof of Ellapsed Time
PoL	Proof of Location
PoS	Proof of Stake
PoW	Proof of Work
REST	Representational State Transfer
SBFT	Simplified Byzantine Fault Tolerance
SDK	Software Development Kit
SHA	Secure Hash Algorithmus
SQL	Structured Query Language
Tx	Transaktion
WFS	Web Feature Service
WFS-T	Web Feature Service with Transactions
WKT	Well-Known-Text
WMS	Web Map Service
YAC	Yet Another Consensus