FERDINAND PORSCHE
FERN FH

Referenzmodell: Blockchain für Patientendaten

Masterarbeit

eingereicht von: **Marcel Zwinger, BA**
Matrikelnummer: 51807382

im Fachhochschul-Masterstudiengang Wirtschaftsinformatik

der Ferdinand Porsche FernFH GmbH

zur Erlangung des akademischen Grades

Master of Arts in Business

Betreuung und Beurteilung: Thomas Krabina, MSc.

Zweitgutachten: Ing. Peter Völkl, MA MSc

Wiener Neustadt, August, 2019

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.

2. dass ich diese Masterarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

3. dass die vorliegende Fassung der Arbeit mit der eingereichten elektronischen Version in allen Teilen übereinstimmt.

<Podersdorf, 07.08.2019> _____

Unterschrift

Kurzzusammenfassung: < Referenzmodell: Blockchain für Patientendaten >

<Die elektronische Verarbeitung von Patientendaten erfolgt in Österreich mit der „Elektronischen Gesundheitsakte“. Die Daten werden dezentral auf den Standorten gespeichert, bei denen sie entstehen. Dadurch ergibt sich der Nachteil, dass bei einem Netzausfall kein Zugriff von außerhalb auf diese Daten besteht. Mithilfe der Blockchain-Struktur lässt sich diese Schwachstelle vermeiden, da hierbei alle Transaktionen bzw. Datensätze auf allen berechtigten Knoten gespeichert werden. Auf der Kehrseite ergibt sich dadurch das Problem zur Sicherstellung des Datenschutzes laut DSGVO und der Datenintegrität.

Durch die Ausarbeitung eines Referenzmodells anhand der „Design Science Research Methodology“ konnte gezeigt werden, dass die rechtlichen Anforderungen an einer Blockchain-Lösung erfüllt sind, wobei die grundlegende Funktionalität von den zwei Beispielfällen „ELGA“ und „Medicalchain“ erhalten bleibt.

Das Referenzmodell wurde anhand von Literaturrecherche der letztgültigen DSGVO sowie ELGA erstellt und mithilfe einer bereits ähnlichen Lösung „Medicalchain“ validiert.

Die Forschungsfrage konnte somit positiv beantwortet werden.>

Schlagwörter:

Blockchain, Patientendaten, ELGA, Datenschutz, Datenintegrität, dezentral, Merkle Tree, Referenzmodell

Abstract: <Reference Model: A Blockchain Solution for Storing Patient Records>

< Health records in Austria are processed via the “Elektronische Gesundheidsakte”. Records are stores in a decentralized manner. They are stored locally but are accessible on the ELGA network. Thus, in case of a network outtake, the data would not be accessible at all. A Blockchain could fix this problem by distributing every transaction to every permissioned note, thus providing a redundant backup. On the other hand, this raises questions about data integrity and data protection according to the GDPR.

A reference model, created by the guidelines of the “Design Science Research Methodology” This method used to create a possible solution for the problems presented above. The model itself was created in an iterative manner. The first iteration was created via literature review of the GDPR and ELGA and the validation process was done by demonstrating, that the model can be applied to another similar solution, in this case “Medicalchain”.

Thus it was possible to successfully validate the reference model and to positively answer the research question. >

Inhaltsverzeichnis

1. ABKÜRZUNGSVERZEICHNIS 1

2. EINLEITUNG 2

2.1 Ziel 4
2.2 Nicht-Ziele 4
2.3 Forschungsfrage 5
2.4 Hypothesen 5
2.5 Aufbau und Methodologie 5

3. REFERENZMODELL 7

3.1 Definition 7
3.1.1 Artefakt 7
3.1.2 Referenzmodell 7
3.1.3 Konstruktionsorientierter Forschungsansatz 8
3.2 Anforderungen an ein Referenzmodell 8
3.3 Qualitätsmerkmale eines Referenzmodelles 8
3.4 Ablauf 9
3.5 Aufbau eines Referenzmodelles 10
3.6 Modellierungssprache 11
3.6.1 UML 11
3.7 Evaluierung des Modells 16

4. PATIENTENDATEN 17

4.1 Definition 19
4.1.1 Personenbezogene Daten 19
4.1.2 Personenbezogenen Gesundheitsdaten 19

4.1.3 Besondere Kategorien personenbezogener Daten 19
4.2 Elektronische Gesundheitsakte 20
4.2.1 ELGA-Gesundheitsdiensteanbieter 20
4.2.2 e-Befund 21
4.2.3 Infrastruktur 21
4.3 Anforderungen an Patientendaten 21
4.3.1 Rechtmäßigkeit der Verarbeitung 22
4.3.2 Transparente Information 22
4.3.3 Berichtigung und Löschung 23
4.3.4 Datenschutzfreundliche Voreinstellung 23
4.3.5 Zusammenfassung der Anforderungen 23

5. BLOCKCHAIN 25

5.1.1 Zusammenfassung 28
5.2 Arten 29
5.3 Verschlüsselungsmethoden 30
5.3.1 Symmetrische Verschlüsselung 30
5.3.2 Asymmetrische Verschlüsselung 31
5.3.3 Digitale Signatur 32
5.3.4 Zusammenfassung 34
5.4 Verteilte Systeme 35
5.4.1 Zentral Vs. Dezentral 35
5.4.2 Peer-to-Peer Netzwerk 37
5.5 Hashfunktionen 38
5.5.1 Hashfunktion 39
5.5.2 Kryptographische Hashfunktionen 39
5.5.3 Hashreferenzen 41
5.5.4 Verkettete Liste 42
5.5.5 Hashbaum 43
5.6 Accounts 44
5.6.1 Erstellen von Accounts 45
5.6.2 Online Vs. Offline 46
5.6.3 Wallets 46
5.6.4 Hierarchisch Deterministische Wallets 47

5.6.5 Skripts 48
5.6.6 Skripts an Hand des Beispiels von Bitcoin 48
5.6.7 Smart Contracts 49
5.7 Speicherung der Transaktionen 50
5.7.1 Transaktions-Basierender Ledger 50
5.7.2 Account-Basierender Ledger 51
5.8 Block-Struktur 51
5.9 Verändern von Daten oder Referenzen 52
5.10 Konsens 54
5.10.1 Proof of Work (PoW) 54
5.10.2 Merged Mining 55
5.10.3 Blockchain Anchoring 57
5.10.4 Mining Rotation 58
5.10.5 Proof of Stake (PoS) 59

6. FESTSTELLEN DER ANFORDERUNGEN 61

6.1 Anforderungen an Patientendaten 61
6.2 Anforderungen an die Blockchain 61
6.2.1 Technische Anforderungen 62
6.2.2 Nicht-Technische Anforderungen 62
6.2.3 Ableitung der ersten UML-Klassen 62
6.2.4 Auswahl der Blockchain Struktur für Patientendaten 63
6.2.5 Auswahl der Konsens-Methode für Patientendaten 66
6.2.6 Auswahl der Account-Art für Patientendaten 70
6.2.7 Auswahl von PKGI/AGI 71
6.2.8 Auswahl der Skript-Sprache für Patientendaten 71

7. ERSTELLEN DES REFERENZMODELLES 72

7.1 Festlegung der Account-Typen 72
7.2 Festlegen der Berechtigungsstruktur 72
7.3 Erstellen von Accounts 75
7.4 Erstellen eines e-Befundes 77

7.5 Erstellen eines Blockes der Patienten-Blockchain 82
7.6 Erstellen einer öffentlichen Blockchain-Transaktion 86
7.7 Zusammenfassung 87

8. ÜBERPRÜFEN DER ANFORDERUNGEN 89

8.1 Validierung im Detail 90
8.2 Ergebnis der Überprüfung 93

9. DEMONSTRATION ANHAND VON MEDICALCHAIN 95

9.1 Aufbau 95
9.2 Berechtigungen 95
9.3 Ablauf 97
9.4 Anforderungen 97
9.5 Überprüfen der Anforderungen 98
9.6 Validierung im Detail 103
9.7 Ergebnis der Validierung 108

10. ANALYSE UND INTERPRETATION DES REFERENZMODELLES 110

11. BEANTWORTUNG DER FORSCHUNGSFRAGE 112

12. ZUSAMMENFASSUNG 113

13. AUSBLICK 115

14. LITERATURVERZEICHNIS 117

15. ABBILDUNGSVERZEICHNIS 121

1. Abkürzungsverzeichnis

AGI. *Address Generation Info*

BC. *Blockchain*

DSGVO. *Datenschutz-Grundverordnung*

DSRM. *Design Science Research Methodology*

ELGA. Elektronische Gesundheitsakte

GDA. *ELGA Gesundheitsdienstanbieter*

PK. *Public Key (Öffentlicher Schlüssel)*

PKGI. *Private Key Generation Info*

PoS. *Proof of Stake*

PoW. *Proof of Work*

SK. *Secre Key (Privater Schlüssel)*

UID. *Unique Identifier*

UML. *Unified Modelling Language*

2. Einleitung

Das österreichische Gesundheitswesen hat mit Stand März 2019 knapp neun Millionen Mitbürger[St19a] zu versorgen. Daraus ergeben sich mittlerweile über 140.000.000 Patientenkontakte jährlich; Tendenz: steigend (siehe Abbildung 1). Rein aus informationstechnischer Sicht betrachtet, bedeutet diese, dass (mindestens) 140 Mio. neue Datensätze pro Jahr entstehen und somit erstellt, erfasst administriert und gesichert werden müssen. Dies passiert im Moment nach keiner einheitlichen Struktur. Daten werden an jener Gesundheitsstelle, an der sie entstehen, gespeichert und sind grundsätzlich nur von dort aus abrufbar bzw. benötigen eine manuelle Kontaktaufnahme zur Übermittlung der Informationen.

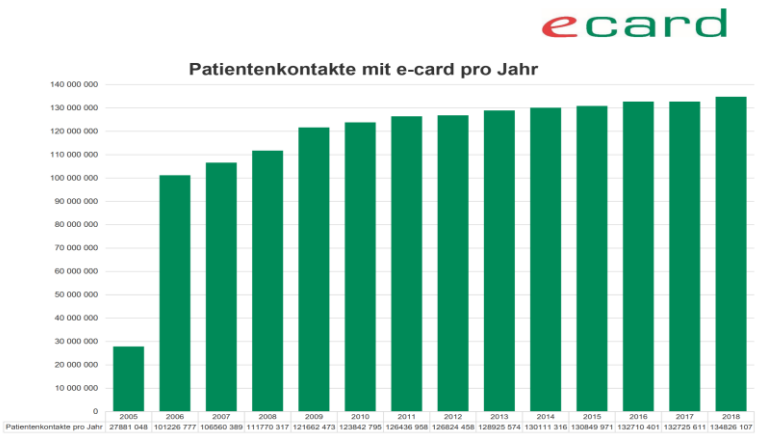

Abbildung 1 - Anzahl an Patientenkontakten pro Jahr
Quelle: https://www.chipkarte.at[Sv19]

Eine Weiterentwicklung diesbezüglich wurde in Österreich mit der elektronischen Gesundheitsakte (ELGA) [El19a] geschaffen, welche sich seit 2015 in der Umsetzungsphase befindet. Diese vernetzt alle Gesundheitsdaten, welche bei einem ELGA Gesundheitsdienstanbieter (GDA) entstehen. Hierdurch können alle Teilnehmer an diesem System ihre relevanten Daten einsehen und pflegen.

Patienten wird es ermöglich, ihre Gesundheitsakte jederzeit zu betrachten, editieren oder löschen und Ärzte können verschiedene Informationen, wie Medikation, Krankheitsbild oder Befunde für ihre Patienten festhalten.[El19b]

Trotz dieser vermeintlichen Verbesserung, der Zugänglichkeit, welche sich durch die Vernetzung ergeben, bleiben folgende potentielle Schwachstellen offen:

1) Es existiert keine implizit gesicherte bzw. unfälschbare Vorgangshistorie: Ein Angreifer oder eine Angreiferin kann bei einem geglückten Hackversuch Patientendaten zum eigenen Gunsten manipulieren, erstellen oder löschen.[Bs19]

2) Bei einem Verbindungsausfall jeglicher Art kann auf Daten auf anderen Speicherorten nicht zugegriffen werden: Dies ergibt sich daraus, dass jeder Datensatz nur auf einer dezentralen Stelle (dort, wo er entsteht) gespeichert wird. Neben eines Netzausfalls durch höhere Gewalt (Stromausfall) können sogenannte Denial of Service DoS bzw. Distributed Denial of Service (DDoS) Attacken die Verbindung zu einem Speicherort lahm legen. Hierbei versendet der Angreifer von mehreren Geräten Requests an das Opfer. Dieses ist zwangsläufig überfordert und die Verbindung wird dadurch unterbunden.[BS98][Bsif00]

Eine Technologie, welches diese Probleme implizit verhindert, stellt die Blockchain-Technologie dar. Durch dessen Aufbau ist es praktisch unmöglich, alte Datenbestände zu manipulieren. Gleichzeitig, stehen alle Daten zu jederzeit an jedem teilnehmenden Knoten, der die entsprechende Berechtigungen besitzt, zur Verfügung. Die Details hierfür werden in Kapitel 5 erläutert.

Verwendung findet diese Methode bereits bei diversen Startup Unternehmen.

Medibloc[LeKh00] will Patientendaten, Spitäler und andere Anbieter aus verschiedenen Ländern miteinander verknüpfen. Im Moment existiert hier nur ein Whitepaper[Le19b] und ein Test-Netzwerk.[Le19a] Aus diesem Grund kann nicht genau gesagt werden ob und wie die Lösung von Lee und Kho funktioniert.

Medicalchain[Me18a] hingegen setzt auf eine Lösung mit zwei Blockchains: Eine private, permissioned Blockchain, in der die Referenzen auf die entsprechenden Datenpakete der zweiten Blockchain, gespeichert werden. [Me18b, S.20] Dies

bedeutet, aber auch, dass zu jeder Zeit beide Blockchains verfügbar sein müssen. Dadurch bleibt bei einem Netzausfall nur ein Teil der Patientendaten zugänglich.

Die Lösungen können somit nicht pauschal für das vorliegende Problem übernommen werden. Es bedarf der Entwicklung einer eigenen Lösung. Hierfür eignet sich das Erstellen eines Referenzmodelles (siehe Kapitel 3) mittels der Methode des Design Science Research nach Hevner entwickelt werden (Kapitel 3.1.3). Im Anschluss wird dieses mit Hilfe der Literaturrecherche und der Anwendung auf einen Konkreten Fall validiert. Diese Vorgehensweise ist deshalb gut geeignet, da durch die Literaturrecherche alle theoretischen Aspekte und durch den konkreten Anwendungsfall zumindest eine praktische Seite abgedeckt wird.

2.1 Ziel

In dieser Arbeit soll nun ein Referenzmodell einer Blockchain-Struktur zur Verwaltung von Patientendaten erstellt werden, welche oben genannte Schwachstellen behebt. Die Anforderungen entsprechen jenen, des vorherrschenden ELGA – Systems.

2.2 Nicht-Ziele

Als Abgrenzung werden für diese Arbeit folgende Nicht-Ziele definiert:

1) Es soll keine funktionsfähige Anwendung mit User-Interface entwickelt werden.
2) Wirtschaftliche Aspekte werden nicht betrachtet (Optimierungen von Durchlaufzeiten, Kosten oder anderen Kennzahlen)
3) ELGA oder e-Card sollen nicht abgelöst werden.
4) Die ethischen Aspekte der Verwendung von ELGA, Blockchain oder anderen Technologien sind nicht Gegenstand dieser Arbeit

5) Rechtliche Anforderungen beschränken sich ebenfalls ausschließlich auf den technischen Blickwinkel

2.3 Forschungsfrage

Nach der Beschreibung des Sachverhaltes, des Zieles und der Nicht-Ziele lässt sich die Forschungsfrage konkretisieren:

Wie muss eine Lösung zur Speicherung der Patientendaten (im Sinne von Krankenakten) mit Hilfe einer Blockchain-Technologie aufgebaut sein, damit diese alle rechtlichen Anforderungen der technischen Informationssicherheit erfüllt und die Sicherheit und Verfügbarkeit der Datenbestände erhöht?

2.4 Hypothesen

Aus der Forschungsfrage lassen sich nachfolgende Hypothesen ableiten, die in der Arbeit schrittweise zu widerlegen oder bestätigen sind.

1) Die Blockchain Struktur eignet sich, alle rechtlichen Auflagen und Anforderungen, welche von ELGA umgesetzt werden, zu erfüllen.
2) Die Blockchain Struktur kann bei der Verarbeitung von Patientendaten die Sicherheit der Datenbestände im Vergleich zu ELGA verbessern.
3) Die Blockchain Struktur kann bei der Verarbeitung von Patientendaten die Verfügbarkeit der Daten im Vergleich zu ELGA verbessern.

2.5 Aufbau und Methodologie

Um zu einem Conclusio zu gelangen erfolgt der Aufbau der Arbeit in der übergeordneten Einteilung Theorie und Empirie. In den Kapiteln 1-5 werden die theoretischen Aspekte der Thematik beschrieben und zusammengefasst. Kapitel 6 und 7 widmet sich der Empirie. Hierbei werden für das Referenzmodell zunächst die Anforderungen erhoben, welche sich durch die vorgehenden Kapitel herleiten lassen.

Aus diesen Anforderungen können nun für das Modell benötigte Prozesse abgeleitet werden. In diese Prozesse wiederum bedarf es unterschiedlicher Rollen, Ereignisse und Tätigkeiten. Die Festlegung dieser Teilaspekte wird schriftlich festgehalten und begründet. Die fertigen Prozesse stellen das Referenzmodell dar. Dieses wird anhand der definierten Anforderungen validiert. Die Anforderungen werden hierbei den entsprechenden Prozessen und Prozessschritten zugewiesen. Sind alle Anforderungen erfüllt (dh können alle Anforderungen zugewiesen werden), so ist das Modell als validiert zu betrachten. Das Conclusio selbst leitet sich durch die Validierung her. Im Anschluss werden die Ergebnisse für aufbauende Arbeiten zusammengefasst und die erlangten Erkenntnisse auf zukünftige Entwicklungen projiziert.

3. Referenzmodell

Zur Beantwortung der Forschungsfrage wurde das Erstellen eines Referenzmodelles gewählt. Dies ermöglicht ein Problem abstrakt zu betrachten und durch eine (oder auch mehrere) Modellierungssprache abzubilden. Wurde das Referenzmodell erstellt, kann es mit den vorhandenen Daten analysiert und validiert werden.

In den nachfolgenden Unterkapiteln folgt zunächst eine Begriffserklärung, Anforderungen an das Referenzmodell selbst, der Ablauf der Konstruktion und Evaluierung sowie der Aufbau des Modelles. Ebenso wird die im späteren Verlauf verwendete Modellierungssprache „Unified Modelling Language“ (UML) [Ra16, S.vii] in ihren wichtigsten Grundzügen beschrieben.

3.1 Definition

Nachfolgend werden relevanten Begriffe im Zusammenhang des Referenzmodelles definiert.

3.1.1 Artefakt

Als Artefakt lassen sich im Zusammenhang dieser Arbeit, Sprachen, Symbole, Modelle, Methoden und Instanzen (dh. eine Implementierung des Modells) verstehen, welche ein nützlichen Ergebnis liefern. [BA16, S.10]

3.1.2 Referenzmodell

Ein Referenzmodell ist ein Informationsmodell, welches nicht nur für einen speziellen Fall sondern auf ein größeres, aber dennoch eingegrenztes Anwendungsgebiet Verwendung finden soll. Es ist sozusagen ein Blueprint, welcher im Anwendungsfall konkretisiert wird. Diese Wiederverwendung für verschiedene Fälle nennt sich Referenzierbarkeit.[Be04, S.1]

3.1.3 Konstruktionsorientierter Forschungsansatz

Hierbei handelt es sich um einen Lösungsansatz, welcher durch die Konstruktion von Artefakten und dessen Evaluierung zu einem Konsensus gelangt.[BA16, S.10]

3.2 Anforderungen an ein Referenzmodell

Die Erstellung des Modells für diese Arbeit folgt dem konstruktionsorientiertem Forschungsansatz. Hierbei lassen sich folgende Anforderungen festhalten:[BA16]

- Abstraktion: Das Modell muss auf eine Gruppe von Problemen angewendet werden können
- Originalität: Es muss einen „innovativen Beitrag“ zum derzeitigen Wissenstand leisten.
- Begründung: Die Begründung für das Modell muss validierbar sein
- Nutzen: Es muss ein Nutzen aus dem Modell entstehen

3.3 Qualitätsmerkmale eines Referenzmodelles

Des Weiteren werden an den konstruktionsorientierten Forschungsansatz sieben Qualitätsmerkmale bzw. Guidelines gestellt, die in dieser Arbeit eingehalten werden müssen:[He10, S.12]

1) Das Referenzmodell wird als Artefakt dargestellt: Dies geschieht, indem das Modell erstellt und auf einen konkreten Fall angewendet wird.
2) Problemrelevanz: Das Modell soll ein ökonomisches sowie bedeutendes Problem lösen: Es löst das Problem der Integrität und Verfügbarkeit von Daten.
3) Die Problemlösung in Bezug auf Qualität, Nützlichkeit sowie Effizient muss mit bekannten Verfahren nachgewiesen werden: Hierfür wird der in Ablauf der Design Science Research Methodology (DSRM) verwendet. (siehe Kapitel 3.4)

4) Die Ergebnisse müssen einen Beitrag zur Wissenschaft darstellen: Die Forschungsfrage behandelt einen weiteren Use Case der Blockchain, welchen es zu beantworten gibt.
5) Strenge der Forschung: Es muss der vorherrschende Wissensstand in Bezug auf Literatur, Theorie und Forschungsmethoden verwendet werden: Es wird die aktuelle Fassung der Datenschutzgrundverordnung sowie die derzeit verfügbaren Literaturwerke, Informationen und White Papers als Literatur und Theorie verwendet.
6) Konstruktion als Suchprozess: Das Modell und dessen Entstehung muss iterativ erfolgen und ebenso dargestellt werden: Bei der Herleitung der Anforderungen wird sukzessive das Modell erstellt bis es am Ende in der Erstfassung vorliegt. Im Anschluss wird dies durch einen konkreten Anwendungsfall validiert.
7) Die Forschungsergebnisse müssen von allen Nutzergruppen verstanden werden können: das es sich um eine Masterarbeit handelt, ist das Niveau implizit vorgegeben.

3.4 Ablauf

Als nächste wird der Ablauf zur Lösungsfindung festgehalten. Einen möglichen Ansatz stellt die Design Science Research Methodology for Information Systems Research (DSRM) nach Hevner zur Verfügung.[He10] Bei der **Problemidentifikation** wird zunächst das Problem beschrieben und dessen Relevanz ausformuliert. Bei der **Zieldefinition** sollen nun das Ziel und die Anforderungen an das Modell definiert werden. Im Anschluss wird das Modell erstellt (**Konstruktion**), auf einen konkreten Fall angewendet (**Demonstration**) und entsprechend evaluiert (**Evaluation**). Abschließend wir das Ergebnis publiziert (**Kommunikation**).[BA16, S.13, 14, 15] Dieser Ablauf ist in Abbildung 2 schematisch dargestellt. Genau genommen bedeutet die Vorgehensweise des „Design Science Research“, dass ein Designer eine relevante Frage durch das

Erstellen von Artefakten beantwortet und durch diesen Vorgang neues Wissen entsteht. [He10, S.5]

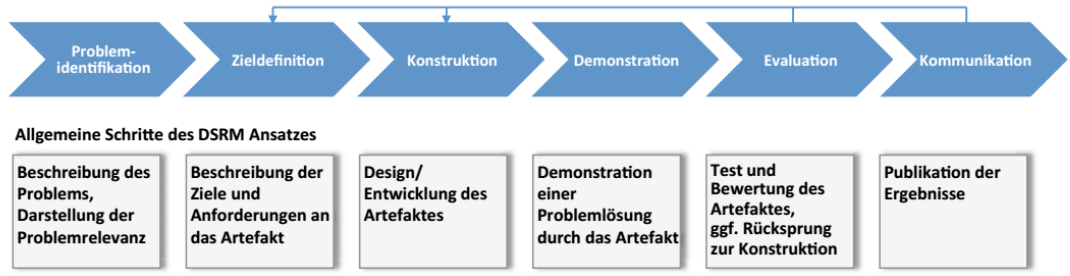

Abbildung 2 - Ablauf eines konstruktionsorientierten Forschungsansatzes
Quelle: siehe [BA16, S.14]

3.5 Aufbau eines Referenzmodelles

Um ein Referenzmodell für einen vorliegenden Fall richtig zu konstruieren, müssen alle relevanten Artefakte berücksichtigt werden. Die Validierung des erstellten Modelles erfolgt dann anhand der erstellten Anforderungen selbst. Hierbei kann es zu mehreren Iterationen des Modelles kommen, bis die Plausibilisierung erfolgreich abgeschlossen wurde.[BA16, S.9]

Des Weiteren handelt es sich bei der Art des Modelles um ein „Beschreibungsmodell“. Hierbei wird der Sachverhalt auf die wesentlichen Aspekte reduziert, um zu einem besseren Verständnis der Zusammenhänge zu gelangen.[BA16, S.21]

3.6 Modellierungssprache

Nun muss noch geklärt werden, was eine Modellierungssprache ist und welche zur Umsetzung in dieser Arbeit Anwendung findet.

Eine Modellierungssprache ist eine künstliche Sprache, welche dazu verwendet wird, einen realen Sachverhalt möglichst genau, aber ebenso leicht verständlich abzubilden. Ebenso soll die Sprache anpassungsfähig sein, um die Implementierung von Änderungen möglichst leicht zu gestalten. Mit der Modellierungssprache werden somit alle Entitäten und deren Wechselwirkung dargestellt, um das zu beschreibende Modell in Bezug auf die Realität abstrahiert darzustellen. Bei der Abbildung konzentriert man sich hierbei auf die relevanten Aspekte des Sachverhaltes. Der Rest wird nicht abgebildet.[BA16, S.26, 27]

3.6.1 UML

In dieser Arbeit wird das Modell mit Hilfe von UML hergeleitet und dargestellt. UML zählt zu den semiformalen Modellierungssprachen, welche einen guten Kompromiss zwischen den Anforderungen der Einfachheit sowie der Korrektheit des Modelles im Vergleich zur Realität bietet.[BA16, S.29]

3.6.1.1 Klassendiagramm

UML bietet unter anderem die Darstellung als Klassendiagramm. Eine Klasse stellt einen Teil oder eine Einheit im Diagramm dar. Die Klassen werden mit den anderen Elementen in Verbindung gebracht, wodurch sich das Modell ergibt. Dargestellt werden Klassen üblicher Weise als Rechtecke. Als weitere Informationen werden Attribute angegeben. Diese können unterschiedliche Bedeutungen annehmen. In dieser Arbeit werden die Typen „Integer“, welche eine Ganzzahl bezeichnet, „String“, welche einen Text beinhaltet „Datetime“ bzw. „Timestamp“, welche eine exakte Zeit- und Datumsangabe spezifiziert, beinhaltet. Weiters können diese als optionale Attribute oder Pflichtfelder definiert werden. In dieser Arbeit werden lediglich

Pflichtfelder ausgewiesen. [BA16, S.29, 30, Ra16, S.7] Abbildung 3 stellt dies schematisch dar.

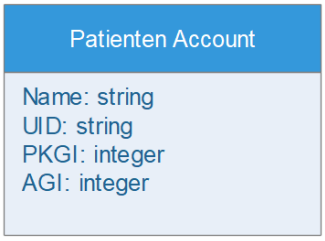

Abbildung 3 - UML Klasse mit Attributen

Um Objekte miteinander zu verknüpfen, werden diese mit einer Linie verbunden. Diese Verknüpfung wird Assoziation genannt. An beiden Enden der Linie (dh bei jedem Objekt) wird eine Zahl angegeben, welche die Multiplizität festlegt. In Abbildung 4 sind drei Klassen miteinander verbunden. Es ist ersichtlich dass ein bestimmter Patienten Account (eine Instanz) 1000 Patienten AGIs besitzen kann, während ein Patienten AGI nur einem Patienten oder einer Patientin zugewiesen werden kann. Der Stern bei der Klasse e-Befund sagt aus, dass beliebig viele, auf keine genaue Anzahl eingeschränkte e-Befunde einer Klasse Patienten AGI zugewiesen werden können.[Ra16, S.9, 10]

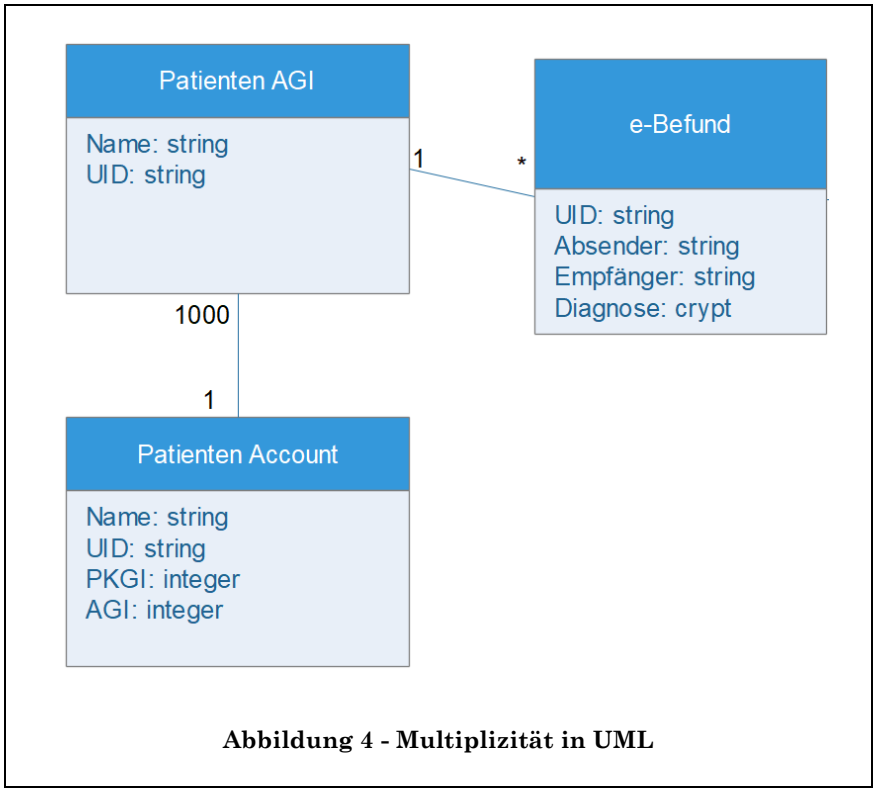

Abbildung 4 - Multiplizität in UML

3.6.1.2 Aktivitätsdiagramm

Um nun einen Prozessfluss darzustellen verwendet man in UML das Aktivitätsdiagramm.

Das erste zu verwendende Objekt wird Aktivität genannt. Die Beschreibung der Aktivität drückt aus, welche Aktion hierbei ausgeführt bzw. getätigt wird. Nehmen mehrere Akteure an einem Prozess teil, werden diese in Schwimmbahnen eingeteilt, um die Übersicht besser wahren zu können. Hiermit ist klar, welcher Akteur welche Aktivität ausführt. Der Beginn und das Ende eines Prozesses werden mit je einem Startknoten und einem Endknoten markiert. Hier kann eine zusätzliche Beschreibung angegeben werden. Stehen nach einer Aktion verschiedene Möglichkeiten bzw. Abzweigungen zur Verfügung, werden diese mit einem Raute-Symbol dargestellt. Die unterschiedlichen Bedingungen, wann welcher Pfad weiter

verfolgt wird, werden entlang des Prozessflusses angegeben. Beinhaltet eine Aktion zu viele Tätigkeiten, wird diese als einen Unterprozess dargestellt. Dieser Unterprozess wird wiederum wie ein Prozess abgebildet. Ob es sich um eine Aktivität oder Unterprozess handelt ist an einem kleinen durchkreuztem Rechteck ersichtlich. [Ra16, S.25–28] In Abbildung 5 sind die in dieser Arbeit verwendeten Symbole aufgelistet.

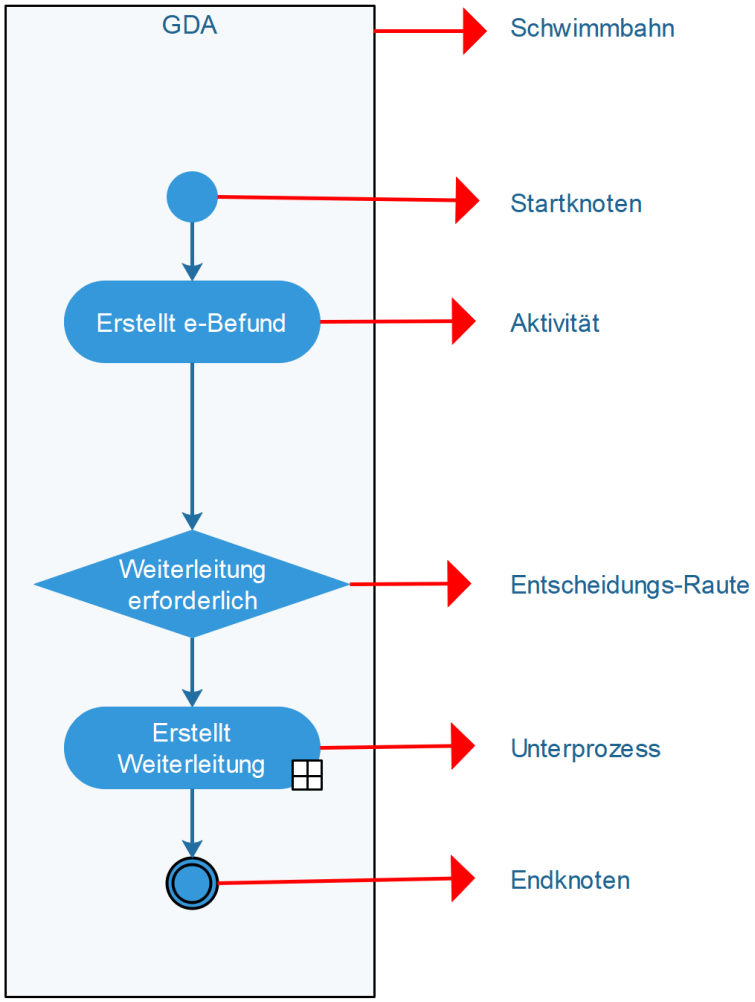

Abbildung 5 - Beispielhaftes Aktivitätendiagramm mit beschrifteten Symbolen
Quelle: in Anlehnung an Abbildung 26

3.7 Evaluierung des Modells

Nach dem Erstellen des Modells muss dieses folglich noch evaluiert bzw. validiert werden. Möglichkeiten hierzu sind die Literaturrecherche sowie die Evaluierung mit einer konkreten Fallstudie.

Sind alle Anforderungen erfüllt, gilt das Modell als validiert. Bei Differenzen müssen diese in einer weiteren Iteration eingearbeitet und das Modell ein weiteres Mal evaluiert werden. Dies passiert so lange, bis keine Diskrepanzen mehr vorliegen.[BA16, S.13, 14]

4. Patientendaten

Unter Patientendaten werden in dieser Arbeit alle Daten verstanden, welche für Personen gelten, die einen Dienst des Gesundheitswesen wiederspiegeln. Im allgemeinen Sprachgebrauch handelt es sich hierbei um Daten, welche von Ärzte, Krankenhäusern oder Apotheken erstellt und gelesen werden können.

Rechtlich gesehen werden Patientendaten in der DSGVO definiert (siehe Kapitel 4.1.). Der Begriff „Patientendaten“ scheint hierbei nicht auf, vielmehr werden „personenbezogenen Daten“ und „personenbezogene Gesundheitsdaten“ angeführt. Aus Gründen der Einfachheit ist im weiteren Verlauf der Arbeit mit „Patientendaten“ die nachfolgende Definition zu verwenden.

Ein beispielhafter Datensatz hierfür lässt sich auf der Homepage von ELGA unter den „Technischen ELGA-Leitfäden“[El19c] finden:

Patientin: Dr. Maria Johanna Musterfrau, BSc
Geschlecht: weiblich | geboren am: 24. Dezember 1961 | SVN: 1111241261

Auftraggeber(in): Musterklinikum Unterstadt

Erstellt von: Amadeus Spital - Labor **An:** Ordination Dr. Empfänger

Dies ist ein Beispielbefund. Bei den Inhalten handelt es sich um synthetische Mustertexte und keinesfalls um personenbezogene Echtdaten oder realistische Befunde. Das Beispiel veranschaulicht die technischen Möglichkeiten unter Verwendung eines Maximums der erlaubten Optionen.

Überweisungsgrund

Text des Zuweisers: Abklärung einer Thrombozytopenie höflich erbeten.

Probeninformation

Material-ID	Probenentnahme	Untersuchtes Material	Probenentnahme durch	Probeneingang	Bemerkung Labor
BL-121201-02	01.12.2012 06:34	BLUT, fossa cubitalis l.	Dr. Humpel	01.12.2012 08:15	leicht hämolytisch
PL-121201-01	01.12.2012 06:34	PLASMA, fossa cubitalis l.	Dr. Humpel	01.12.2012 08:15	

Hämatologie

Blutbild

Analyse	Ergebnis	Einheit	Referenzbereiche	Interpretation	Delta
Leukozyten	26	G/l	4-10	+	d+
Thrombozyten	165	G/l	150-360		d-
Erythrozyten	5.39	T/l	4.60-6.20		
Hämoglobin	16.0	g/dl	14.0-18.0		
Hämatokrit	49.7	%	43.0-49.0	+	
MCH	29.7	pg	27.0-33.0		
MCV	92.2	fl	85.0-95.0		
MCHC	32.2	g/dl	28.0-33.0		
Akt.Lymphoz.rel.mi.	7	%	0-10		

Abbildung 6 - Beispielhafter Laborbefund
Quelle: ELGA Laborbefund[1]

In Abbildung 6 ist zu erkennen, dass es sich bei Patientendaten vom Aufbau her um semistrukturierte Daten mit Freitext und tabellarischer Darstellung handelt. Die exakten Leitfäden zur Erstellung der einzelnen Dokumententypen finden sich ebenfalls auf der ELGA Webpage unter[El19c]. Der Prozess zur Erzeugung eines dieser Dokumente ist von dieser Arbeit abgegrenzt. Wichtig bleibt aber die

Notwendigkeit, dass die Blockchain verschiedene bzw. verschieden große Datensätze unterstützen muss.

4.1 Definition

Bevor die Anforderungen an Patientendaten festgestellt werden können, muss der Begriff der Patientendaten definiert werden. Dieser findet sich in der österreichschein Umsetzung der Datenschutz-Grundverordnung (DSGVO) und können folgendermaßen abstrahiert werden:[Eu16]

4.1.1 Personenbezogene Daten

Art. 4, Abs. 1 der DSGVO definiert den Begriff der „personenbezogenen Daten“, als jene Daten, welche es ermöglichen, eine natürliche Person eindeutig zu identifizieren. Hierbei ist es nicht relevant, auf welche Art und Weise dies durch Daten ermöglicht wird. Hervorgehoben werden durch die DSGVO unter anderem der Name, eine Kennnummer oder auch der Standort sowie psychische oder physische Merkmale.

4.1.2 Personenbezogenen Gesundheitsdaten

Art. 4, Abs. 15 der DSGVO definiert diesen Begriff als jene Daten, welche den Gesundheitszustand einer natürlichen Person beschreiben. Daher es handelt sich hierbei um körperliche, sowie geistige Eigenschaften der Person. Ebenso werden darunter Informationen über erbrachte Dienstleistungen im Gesundheitswesen, wie Medikation oder Behandlungen, verstanden.

4.1.3 Besondere Kategorien personenbezogener Daten

Aufbauend auf den personenbezogenen Daten lassen sich nun besondere personenbezogenen Daten definieren.

Folgende Informationen werden in der DSGVO, Art. 9 Abs. 1 als besondere Kategorien festgehalten:

- Rassische oder ethnische Herkunft
- Politische Meinung
- Religiöse Meinung
- Weltanschauung
- Gewerkschaftszugehörigkeit
- Genetische oder biometrische Daten
- Gesundheitsdaten
- Sexualleben

4.2 Elektronische Gesundheitsakte

Die Elektronisch Gesundheitsakte (ELGA) vernetzt elektronische Patientendaten, welche verteilt an verschiedenen Standorten entstehen. Die Planung dieses Vorgehens begann mit November 2009, die Umsetzung startete schrittweise in den einzelnen Bundesländern im Dezember 2015. Ein genauer Abschluss wird auf der Website nicht festgehalten, lediglich, dass sich die Fertigstellung im „Ziellauf" befindet. Ebenso werden über 12 Millionen elektronische Befunde zum Zeitpunkt des Schreibens auf der Site angeführt. Gleichzeitig werden einige Datenbestände wie der „zentrale Patientenindex" geführt, um Patientendaten eindeutig den richtigen Personen zuordnen zu können. [El19b]

4.2.1 ELGA-Gesundheitsdiensteanbieter

Unter den ELGA-Gesundheitsdiensteanbieter (kurz ELGA-DGA oder DGA) definiert ELGA Krankenanstalten, Pflegeanstalten sowie Ärzte und Ärztinnen. Diese Zielgruppen sind am ELGA Netzwerk angebunden und können einerseits Daten in das System schreiben sowie Daten auslesen, wobei der Zugriff auf jene DGA eingeschränkt ist, die den jeweiligen Patienten oder die jeweilige Patientin im Moment auch behandeln. (ELGA-G, Art.1 Abs.4 §14). [El19b] Standardmäßig können GDAs 28 Tage ab Beginn der Behandlung die benötigten Patientendaten einsehen.[El19d]

4.2.2 e-Befund

Der e-Befund stellt die Basis der Patientendaten dar. Anstelle des Papiers werden die Befunde rein digital gespeichert und übermittelt. In [Hl19] ist ein Implementierungsleitfaden, welcher genau vorgibt, wie ein Befund aufgebaut werden muss, definiert. Ebenso wurden bereits unterschiedliche Arten von Befunden gelistet.[Hl19, S.7]

4.2.3 Infrastruktur

Das ELGA Netzwerk ist dezentral aufgebaut. Die erstellten Patientendaten werden bei den einzelnen GDAs gespeichert und bei Bedarf weitergeleitet. Dadurch existiert jeder Datenbestand garantiert nur einmal.[El19e]

4.3 Anforderungen an Patientendaten

In der DSGVO, Art. 5, Abs. 1 wurden ebenso die rechtlichen Anforderungen, Pflichten und Einschränkungen bei der Verarbeitung von (besonderen) personenbezogenen Daten als Datenschutzgrundsätze festgehalten. Zusammengefasst für diese Arbeit ist aus der DSGVO, Art.5 Abs.1 und aus ELGA-G folgendes zu beachten:[Eu16]

- Personenbezogene Daten müssen für die betroffenen Personen nachvollziehbar verarbeitet werden. Eine betroffene Person muss somit zumindest technisch in Erfahrung bringen können, was wann und wozu mit ihren Daten passiert und wozu diese verwendet wurden. Ebenso müssen diese nach den Werten „Rechtmäßigkeit, Verarbeitung nach Treu und Glauben, Transparenz“ erfolgen. Dies wird in den Kapiteln 4.3.1 sowie 4.3.2 erläutert.
- Der Verwendungszweck muss eindeutig festgelegt sein.
- Die verwendeten Daten müssen richtig und auf dem neuesten Stand sein und bei Bedarf bzw. vor der Verwendung, rechtzeitig aktualisiert werden.

- Die verwendeten Daten dürfen nur so lange gespeichert werden, wie dies für den Verwendungszweck notwendig ist. Es dürfen nur die Patienten selbst und jene ELGA-DGA auf die Patientendaten zugreifen, die eine betreuende oder unterstützende Funktion gegenüber der Patienten besitzen. (ELGA-G, Art.1 Abs.4 $14 (2)) Ausgenommen hiervon sind die Verwendung der Daten für Archivzwecke oder historische Zwecke.
- Die Daten müssen ab dem Beginn der Behandlung für einen beschränkten Zeitraum (28 Tage bei ELGA) für GDAs einsehbar sein.
- Bereits gespeicherte Gesundheitsdaten dürfen nicht geändert werden. Hier ist eine zusätzliche, aktualisierte Version mit einem entsprechendem Vermerk anzulegen.(ELGA-G, Art.1 Abs.1 §20)
- Die Verarbeitung der Daten muss in einer Form geschehen, „die eine angemessene Sicherheit der personenbezogenen Daten gewährleistet". (DSGVO Art. 5 Abs. 1 (f). Eine unbefugte oder unrechtmäßige Verwendung soll genauso ausgeschlossen werden, wie eine unabsichtliche Schädigung oder Löschung der Daten.
- Ebenso muss die Einhaltung dieser Vorgaben im Sinne der „Rechenschaftspflicht" nachgewiesen werden können.

4.3.1 Rechtmäßigkeit der Verarbeitung

Unter der Rechtmäßigkeit der Verarbeitung wird verstanden, dass personenbezogene Daten nur unter gewissen Voraussetzungen weiterverarbeitet werden dürfen. Art. 6 Abs. 1 der DSGVO zählt diese Bedingungen genauestens auf.. Diese werden von dieser Arbeit abgegrenzt, da es bei der Definition des Referenzmodelles um die Erfüllung der technischen Aspekte der Blockchain, jedoch nicht der ethischen Aspekte der handelnden Personen geht.

4.3.2 Transparente Information

Art. 12. Abs. 1 hält fest, dass jegliche Verwendung der Daten an die betroffenen Personen in verständlicher Weise zu übermitteln ist. Die Sprache und Formulierung

soll dabei möglichst einfach gewählt werden. Ebenso ist die Darstellung durch einfach verständliche Symbole möglich. Die Information soll unmittelbar nach der Verwendung der Daten, aber auch auf Anfrage einer betroffenen Person erfolgen. Art. 12. Abs. 2 bis Abs. 4 definieren genauere Zeitfenster. Diese sind in dieser Arbeit nicht von Bedeutung. Wichtig ist, dass Daten und dessen Verarbeitung nachvollziehbar und richtig zur Verfügung stehen und an betroffene Patienten übermittelt werden können.

4.3.3 Berichtigung und Löschung

Art. 16 legt fest, dass eine betroffene Person das Recht besitzt, ihre personenbezogenen Daten berichtigen oder vervollständigen zu lassen.

Art. 17 legt weiters fest, dass eine betroffene Person jederzeit die Löschung ihrer personenbezogenen Daten anfordern kann.

In beiden Fällen existieren Einschränkungen und Anforderungen zu diesen Rechten, welche in Art. 17 Abs.3 festgehalten werden. Wichtig ist hierbei, dass die Möglichkeit einer Berichtigung, Vervollständigung und Löschung grundsätzlich existiert und somit technologisch auch unterstützt werden muss, unabhängig davon, wann dieser Fall eintreten könnte.

4.3.4 Datenschutzfreundliche Voreinstellung

Art. 25. Abs. 1 und Abs. 2 legen fest, dass bei der Verarbeitung von personenbezogenen Daten nur jene Daten verwendet werden sollen, die für den jeweiligen Verwendungszweck auch tatsächlich benötigt werden und gleichzeitig alle Datenschutzgrundsätze eingehalten werden.

4.3.5 Zusammenfassung der Anforderungen

Aus diesen Gesetzestexten ergibt sich folgender verkürzter Sachverhalt:

- Patientendaten dürfen nur dem Zwecke entsprechend verwendet, und daher nicht an Dritte/Unbeteiligte weitergegeben werden.
- Dritte/Unbefugte dürfen daher keinen Zugriff auf diese Daten besitzen
- Patientendaten müssen dem letzten Stand entsprechen und daher von den Beteiligten aktualisiert werden können, wobei der Betroffene die Aktualisierung beim Verantwortlichen anfordert und dieser die Aktualisierung nachvollziehbar durchführt.
- Getätigte Daten bezüglich Behandlungen dürfen nicht geändert werden. Hier ist eine zusätzlicher, aktueller Datensatz anzulegen.
- Die aktuellen Daten und jeglicher Verarbeitungsvorgang muss jederzeit von den Betroffenen einsehbar sein.

5. Blockchain

„Die Blockchain ist ein für alle zugängliches Peer-to-Peer-System.“[Dr17, S.112]

Die Blockchain Technologie wurde erstmal von Satoshi Nakamoto ausgearbeitet, als er an der Kryptowährung Bitcoin arbeitete. Streng genommen handelt es sich beim Erfinder lediglich um ein Pseudonym. Die wahre Identität bleibt bis zum Zeitpunkt des Schreibens geheim.[NA16, S.XXIII]

Bei Bitcoin handelte es sich um eine Entwicklung seitens Satoshi welche unterschiedliche neue Ansätze zur Datenspeicherung und Datensicherheit verwendete. Datensätze werden dezentral auf allen teilnehmenden Knoten gespeichert. Dies bedeutet, dass jeder Knoten jederzeit nach Synchronisation über alle Daten verfügt.(siehe Kapitel 5.4.2)

Im Moment existieren insgesamt knapp unter 490.000 einzigartige Adressen, welche sich beim Bitcoin-Netzwerk registrierten (siehe Abbildung 7) wobei die Zunahme zu Stagnieren scheint.

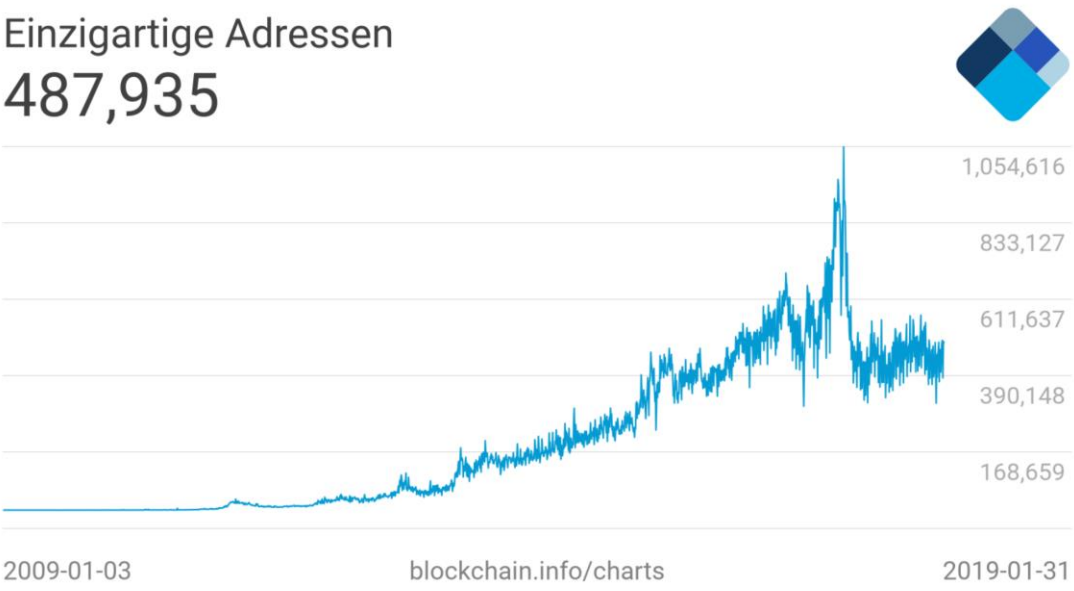

Abbildung 7 - Einzigartige Adressen bei Bitcoin

Quelle: https://www.blockchain.com[Bc19a]

Des Weiteren werden die Daten mit Hilfe einer Baumstruktur, genau genommen eines Merkle Trees (siehe 5.5.5) verlinkt. Dies bewirkt, dass Daten blockweise zusammengefasst sind. Der „Kopf" eines Blockes verweist auf die darunter liegenden Datenpakete und wird weiters als Referenz für den nachfolgenden Block verwendet, wodurch sich eine Kette aus Blöcken ergibt. Dieser Vorgang wird in Kapitel 5.8 hergeleitet und erklärt.

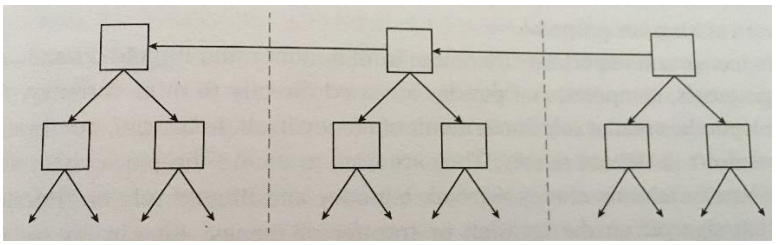

Abbildung 8 - Schematische Darstellung einer Blockchain
Quelle: [NA16, S.XXI]

Wie in Abbildung 8 ersichtlich referenziert jeder Kopf der Blockchain auf den davor gegangenen Kopf. Zusätzlich referenzieren die Köpfe auf darunterliegende Daten oder Referenzen, welche wieder auf weitere Daten referenzieren. Wie in Kapitel 5.9 beschrieben, führt eine Veränderung eines einzigen Wertes dazu, dass die gesamte Blockchain ungültig wird. Eine Manipulation ist daher praktisch unmöglich.

Überdies ist die Methode, wie Blöcke hinzugefügt werden über das Konzept „Proof of Work“ (siehe Kapitel 5.10.1) geregelt. Nur jener Knoten, welcher ein durch Rechenleistung zu lösendes Puzzle knackt (genannt „mining“), erstellt den nächsten Block mit den dazu gehörenden Datensätzen. Andere Teilnehmer validieren die Richtigkeit des gelösten Puzzles. Als Incentive erhalten alle aktiv beteiligen Knoten eine gewisse Anzahl an Bitcoins. Aus diesem Grund entwickelte das System von Bitcoin eine Eigendynamik, da das Hinzufügen von Daten als auch die Sicherstellung der Richtigkeit belohnt wird.

Die Veränderungen der Datenbestände in der Blockchain werden Transaktionen genannt.[Dr17, S.78] Prinzipiell zeigt eine Transaktion eine Veränderung eines oder mehrerer Benutzerkonten an. Bei Bitcoin liegt der Hauptnutzen darin, den Kontostand an Bitcoins des jeweiligen Knotens zu einem bestimmten Zeitpunkt festzuhalten.[Dr17, S.78]Genau diese Transaktionen werden, wie vorhin beschrieben, zu Blöcken zusammengefasst und miteinander „verlinkt“.

Derzeit wurden, wie in Abbildung 9 ersichtlich, im Bitcoin Netzwerk ca 380 Mio. Transaktionen getätigt.

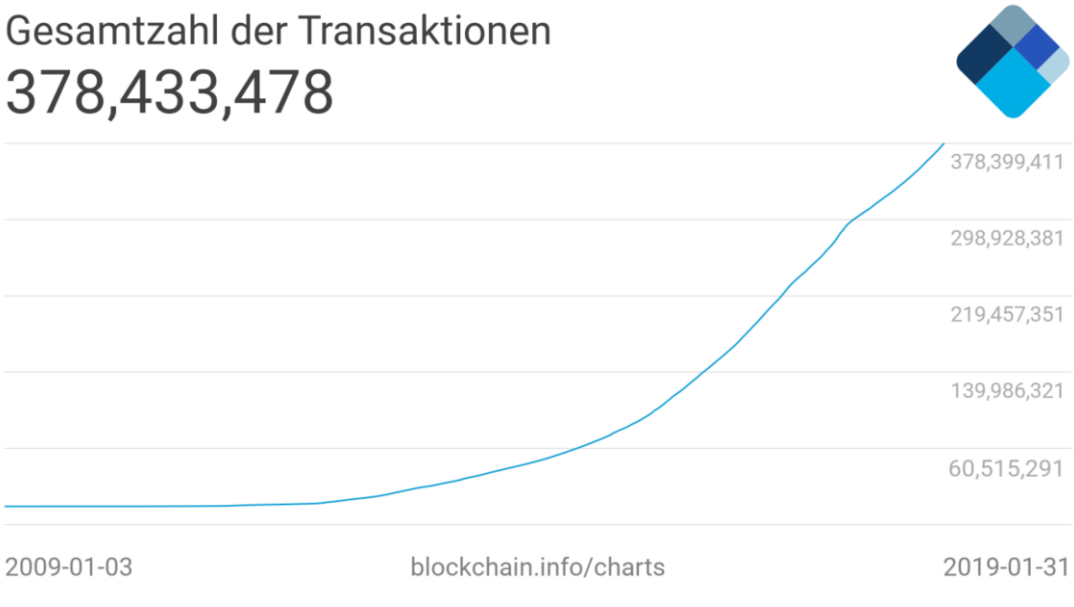

Abbildung 9 - Anzahl der Bitcoin Transaktionen
Quelle: https://www.blockchain.com[Bc19b]

5.1.1 Zusammenfassung

Vereinfacht lässt sich folgender Vorgang festhalten: Durch das Minen von Bitcoins oder das Überweisen einer gewissen Summe von einem Account auf den anderen entstehen Transaktionen. Eine gewisse Menge an Transaktionen wird in einem Block zusammengefasst. Der Kopf des Blockes referenziert einerseits auf die darunterliegenden Datenblöcke und andererseits auf den davor gegangenen Block. Durch diesen Vorgang entsteht die Blockchain. Diese wird auf alle beteiligten Knoten vollständig dh dezentral gespeichert. Mit Hilfe unterschiedlicher Methoden, welche in Kapitel 5.10 genauer beschrieben sind, (unter anderem „Proof of Work“)

wird definiert, wer den nächsten Block hinzufügen darf. Um die Gültigkeit der Blockchain zu wahren, können die restlichen beteiligten Benutzer diesen neuen Block validieren.

Das Ziel einer Blockchain ist es somit, die Integrität in einem Verteilten System sicherzustellen. Dies bedeutet, dass alle Transaktionen in der getätigten Reihenfolge als nicht veränderbare Historie gespeichert werden. Es soll möglich sein, jegliche Manipulationen an Transaktionsdaten (Fälschung, Löschung, Hinzufügen von Transaktionen) vermieden bzw. rasch zu erkennen.[Dr17, S.129, 130]

Dadurch ist einerseits geregelt, wie Daten hinzugefügt und validiert werden und andererseits, dass diese nicht absichtlich oder unabsichtlich verloren gehen können.

In den nächsten Kapiteln wird nun genauer beschrieben, in welche Kategorien sich Blockchains (nachfolgend mit Akronym „BC“ bezeichnet) einteilen lassen, welche Technologien für eine Blockchain benötigt werden und wie diese grundsätzlich aufgebaut ist. Ebenso widmen sich die nächsten Kapiteln der Thematik, wie Daten hinzugefügt, validiert und in der Blockchain dezentral verteilt werden. Das Erstellen und Verwalten der Accounts wird ebenfalls beschrieben. Dadurch soll ein Gesamtbild entstehen, wie Aufbau und Funktionsweise ineinandergreifen.

5.2 Arten

Grundsätzlich lassen sich Blockchains folgendermaßen kategorisieren:[Dr17, S.227, 228]

Öffentliche Vs. private Blockchain: bei einer öffentlichen BC kann jeder Anwender und jede Anwenderin Transaktionen erstellen, sowie einsehen. Bei einer privaten sind diese Aktionen auf einen definierten Userkreis beschränkt

Genehmigungsfreie Vs. genehmigungspflichtige Blockchain: Bei ersterer Variante können von allen Beteiligten die getätigten Transaktionen überprüft,

sowie neue Blöcke hinzugefügt werden. Bei zweiterer werden für diese Tätigkeiten gezielte Berechtigungen an einzelne Knoten vergeben.

Hieraus ergeben sich nach Drescher folgende Möglichkeiten einer BC:[Dr17, S.228]

- **Öffentlich und genehmigungsfrei**: Jeder Anwender und jede Anwenderin kann Transaktionen erstellen, einsehen und überprüfen, sowie neue Knoten hinzufügen.
- **Öffentlich und genehmigungspflichtig**: Jeder Anwender und jede Anwenderin kann Transaktionen erstellen und einsehen. Das Erstellen von Blöcken sowie Überprüfen von Transaktionen obliegt einem kleinen Userkreis.
- **Privat und genehmigungsfrei**: Das Erstellen und Einsehen von Transaktionen ist auf einen vordefinierten Userkreis beschränkt. Allerdings darf jeder Knoten Transaktionen kontrollieren sowie neue Blöcke hinzufügen.
- **Privat und genehmigungspflichtig**: Alle Aktionen obliegen einem eingeschränkten Userkreis.

5.3 Verschlüsselungsmethoden

Um den Inhalt bzw. die Bedeutung von Daten vor Unbefugten geheim zu halten, bedient man sich der Methode der *Verschlüsselung*. Das Gegenstück hierbei stellt die *Entschlüsselung* dar, welche verschlüsselte Daten wieder in ihre Originalform transformiert.[Dr17, S.113]

Grundsätzlich existieren hierbei zwei Methoden:

5.3.1 Symmetrische Verschlüsselung

Hierbei wird für den Vorgang der Ver- und Entschlüsselung derselbe Schlüssel verwendet. Ist man im Besitz dieses einen Schlüssels, so ist man in der Lage, verschlüsselte Daten zu entschlüsseln, sowie verschlüsselte Daten dieser Art zu

erzeugen. Abbildung 10 stellt dies schematisch dar. Die Methode sei zur Vollständigkeit erwähnt, stellt aber im weiteren Verlauf keine Relevanz dar.[Dr17, S.114]

Abbildung 10 - Symmetrische Verschlüsselung (schematisch)
Quelle: [Dr17, S.114]

5.3.2 Asymmetrische Verschlüsselung

Bei dieser Art werden für die Ver- und Entschlüsselung jeweils ein unterschiedlicher Schlüssel (in Kombination: ein Schlüsselpaar) benötigt. In Abbildung 11 ist wieder eine schematische Funktionsweise dargestellt.[Dr17, S.114, 115]

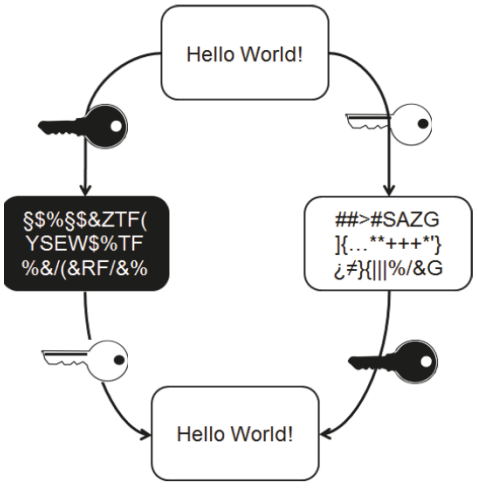

Abbildung 11 - Asymmetrische Verschlüsselung (schematisch)
Quelle: [Dr17, S.115]

Der Ablauf funktioniert generell folgendermaßen: [Dr17, S.114, 115]

- Schlüssel Schwarz erzeugt aus dem Originaltext den Geheimtext Schwarz
- Schlüssel Weiß kann den Geheimtext Schwarz entschlüsseln
- Schlüssel Weiß erzeugt aus einem weiteren Text den Geheimtext Weiß
- Schlüssel Schwarz wiederum kann diesen Geheimtext Weiß entschlüsseln

Zu beachten ist hier, dass Schlüssel Schwarz nicht den Geheimtext Schwarz entschlüsseln kann. Analoges gilt für Schlüssel Weiß. Dies bedeutet, dass eine Person, welche zB. Schlüssel Schwarz besitzt, nur Geheimtext Schwarz erzeugen und nur Geheimtext Weiß entschlüsseln kann, jedoch nicht umgekehrt.[Dr17, S.115]

Typischerweise wird bei einem Schlüsselpaar einer davon als „privater Schlüssel" oder „private Key" („SK" für „secret key") und der zweite als „öffentlicher Schlüssel" oder „public Key" („PK") bezeichnet. Der öffentliche Schlüssel ist für jeden zugänglich, der private Schlüssel ist hingegen nur einer Person oder Organisation zugänglich und darf nicht der breiten Masse zur Verfügung gestellt werden.[Dr17, S.116]

Dies resultiert in zwei mögliche Kommunikationswege [Dr17, S.116, 117] :

- Öffentlich -> Privat: Der öffentliche Schlüssel wird verwendet, um dem Besitzer oder der Besitzerin des privaten Schlüssels vertrauliche Nachrichten zukommen zu lassen. Jeder kann eine Nachricht schicken, jedoch nur eine Person kann diese entschlüsseln und lesen.
- Privat -> Öffentlich: Der Besitzer oder die Besitzern schickt eine verschlüsselte Nachricht, welche potentiell von jedem gelesen werden kann. Der Sinn ist hierbei die sogenannte Unabstreitbarkeit. Die Nachricht kann nur von dieser Person geschickt worden sein, da nur sie den private Key besitzt.

5.3.3 Digitale Signatur

Aufbauend auf die beiden vorangegangenen Kapiteln lässt sich nun das Prinzip der digitalen Signatur erklären. Analog zu einer analogen Signatur dh. Unterschrift auf

einem Stück Papier, verifiziert die digitale Signatur die Echtheit eines Dokumentes und dessen AbsenderIn. [Dr17, S.122, 123]

5.3.3.1 Signieren einer Nachricht

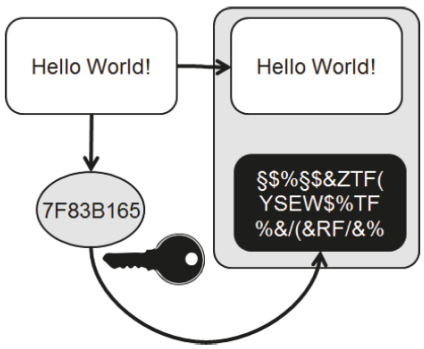

Abbildung 12 - Erstellen der Digitalen Signatur (schematisch) Quelle: [Dr17, S.123]

Wie in Abbildung 12 dargestellt erzeugt die AbsenderIn eine Nachricht (zB: „Hello World“). Aus dieser Nachricht wird ein Hashwert generiert, welcher mit dem privaten „schwarzen“ Schlüssel verschlüsselt wird. Der erzeugte Wert wird dem Originaltext beigefügt.[Dr17, S.123]

5.3.3.2 Verifizieren der Echtheit

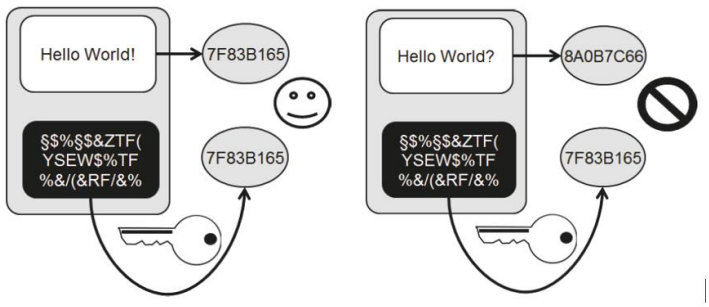

Abbildung 13 - Verifizierung einer Signatur: Echt Vs. Unecht (schematisch)
Quelle: [Dr17, S.124]

Um nun zu überprüfen, ob die digitale Signatur echt ist, wird als erstes der Hashwert aus dem Originaltext erzeugt. Der Geheimtext (die Signatur) wird mit dem public Key entschlüsselt und der erhaltene Hashwert mit dem selbst berechneten Hashwert (siehe Kapitel 5.5) verglichen. Stimmen die Werte überein ist einerseits die Nachricht unverändert und andererseits die Absenderin verifiziert. Stimmen die Werte nicht überein, so ist entweder die Nachricht unecht oder die AbsenderIn nicht diejenige, für die sie sich auszugeben versucht (oder beides). Abbildung 13 veranschaulicht diesen Sachverhalt. [Dr17, S.124]

Vor allem diese Art der Signatur wird für die Funktionsweise der Blockchain benötigt.

5.3.4 Zusammenfassung

Die Kombination aus PK und SK sowie die Möglichkeit, Daten hiermit zu verschlüsseln und nur für bestimmte Personen zugänglich zu machen oder ein Datenpaket zu signieren dient als Grundlage zur Erstellung von Accounts und der sicheren Übertragung von Daten bzw. des Erstellens von gültigen, verifizierten und

verschlüsselten Transaktionen. Auf die Verwendung in diesem Anwendungsfall wird in Kapitel 5.6 sowie 5.6.4 eingegangen.

5.4 Verteilte Systeme

Damit ein System verwendet werden kann, muss es zur Verfügung gestellt werden. Hierbei lassen sich zwei Hauptarten unterscheiden: Zentral und dezentral.

Die Blockchain verwendet hierbei die dezentrale Struktur im speziellen die „Peer-to-Peer“ Architektur.[NA16, S.66] Nachfolgend werden diese drei Arten kurz vorgestellt, um die Verteilung der Daten in einer Blockchain besser verstehen zu können.

5.4.1 Zentral Vs. Dezentral

Bei einem zentralen System bildet ein Server den Ankerpunkt für alle verbundenen Knoten. Hierbei fordern die Clients alle Daten und Services vom Server direkt an. Fällt letzter aus, so stehen dessen Services und Daten nicht mehr zur Verfügung.[LI00, S.78]

Im Gegensatz hierzu stehen die verteilten Systeme (auch „dezentral“ genannt). Hierbei wird ein Client gleichzeitig zum Server und bietet anderen Clients diverse Services an. Das gesamte System wird somit entlastet, da jeder Client eine Teilfunktion übernehmen kann.[LI00, S.79]

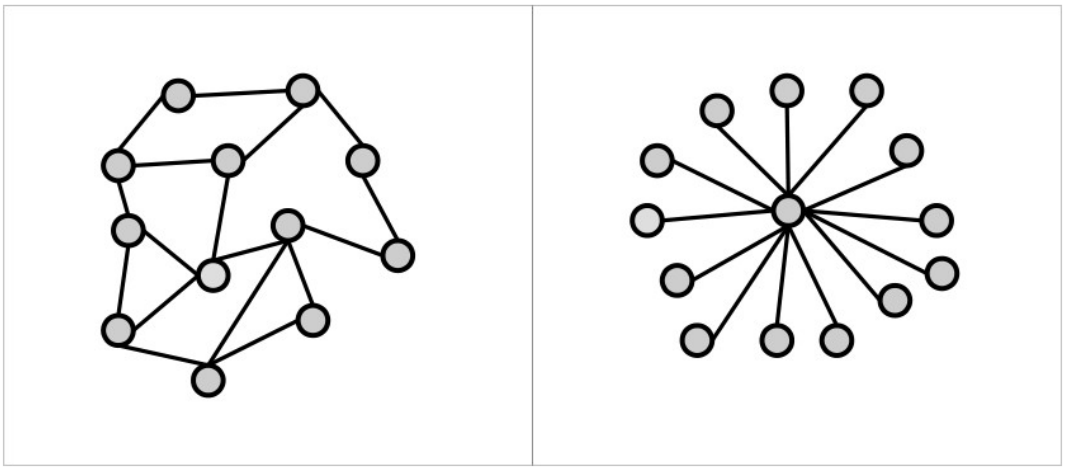

Abbildung 14 - Dezentral vs. Zentral
Quelle: In Anlehnung auf [Dr17, S.31]

Der in Abbildung 14 dargestellte dezentrale Aufbau einer Architektur bietet folgende Vorteile:

Da das System, wie in Abbildung 14 dargestellt, auf mehreren Rechner (Knoten) verteilt ist, führt ein Ausfall eines einzelnen Rechners nicht zum Totalausfall des gesamten Systems („Single Point of Failure“). Andere Rechner im System übernehmen die Aufgaben der ausgefallenen Elemente und stellen die Dienste und Daten weiterhin zur Verfügung. Daraus folgt, dass ein dezentrales System nicht von einer einzigen Komponente ausgeschaltet werden kann. Sollte dies möglich sein, handelt es sich streng genommen um ein „zentrales“ System oder einer hybriden Mischform. Im Umkehrschluss bedeutet dies, dass neue Rechner nicht nur ohne große Störung wegfallen sondern auch mit geringem Aufwand hinzugefügt werden können. [LI10, S.15, 16, 17] [Dr17, S.31, 32]

Auf Grund dieses Aufbaues ergeben sich aber auch Nachteile, auf die Rücksicht genommen werden muss. Da keine zentrale Steuerung der Instanzen existiert,

übernehmen die Knoten selbst die Aufgabe der Koordination. Daraus ergibt sich ein Mehraufwand („Overhead“) an benötigter Rechenleistung. Folglich werden weitere Ressourcen und eigene Kommunikationsprotokolle benötigt, welche den „Overhead“ zusätzlich erhöhen. Verteilte Systeme müssen zwangsläufig mit Hilfe eines Netzwerkes ihre Nachrichten bzw. Informationen austauschen. Steht kein Netzwerk zur Verfügung oder fällt das vorhanden aus, existiert (in diesem Moment) auch kein verteiltes System. Des Weiteren birgt jegliche Übertragung von Daten in einem Netzwerk die Gefahr, Angreifern Opfer zu fallen, welche übertragene Daten potentiell lesen oder manipulieren können. [Dr17, S.32, 33, 34]

5.4.2 Peer-to-Peer Netzwerk

Eine Form des dezentralen Netzwerkes stellt die „Peer-to-Peer“ Architektur dar. Hierbei sind alle Teilnehmer bzw. Knoten gleichgestellt. Dies bedeutet, dass jeder Knoten allen anderen Knoten alle Services und Daten zur Verfügung stellt und somit auch alle Daten von allen Knoten erhält. Streng genommen bedeutet dies, dass keine Client-Server Architektur existiert. Für die exakte Klassifizierung eines Systems ist zu berücksichtigen, dass sich unterschiedliche Misch- und Unterformen von Peer-to-Peer“ Architekturen entwickelt haben. Für das Grundverständnis ist dies aber nicht relevant. [LI10, S.5, 6, 7]

In Abbildung 15 ist anhand von Kazaa[Dy19] ein Peer-to-Peer Netzwerk mit zwei unterschiedlichen Knotentypen abgebildet. In dieser Architektur existieren sowohl „Ordinary Nodes“ als auch „Super Nodes“. Letztere übernehmen zu den definierten Services des Netzwerkes zusätzliche Aufgaben. Zu beachten ist, dass es sich bei diesen „Super Nodes“ um keine dedizierten Server handelt, sondern die Rechner dynamisch (in diesem Fall anhand ihrer Rechenleistung) für diese Funktion ausgewählt werden. [LI10, S.12]

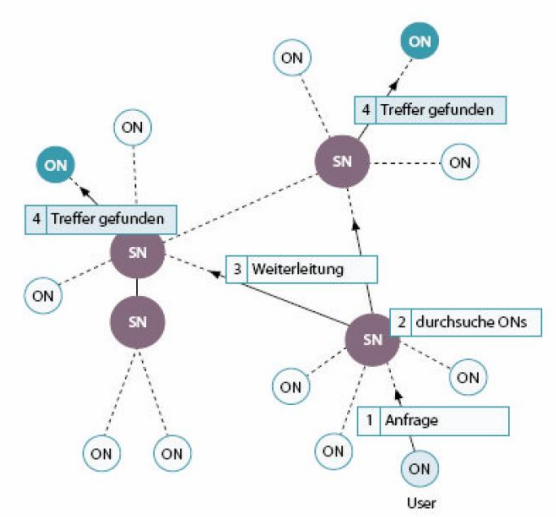

Abbildung 15 - Kazaa Netzwerk mit Super Nodes und Ordinary Nodes
Quelle: [LI10, S.12]

Da die Blockchain-Architektur darauf aufbaut, ihre Transaktionen auf allen beteiligten Knoten gleichermaßen zu verteilen und anzubieten, kann die verwendete Netzwerkarchitektur dieser als „Peer-to-Peer“ eingestuft werden.

5.5 Hashfunktionen

Hashfunktionen, Hashwerte und Hashreferenzen stellen die Basis zur Verifizierung von Daten dar. Um den generellen Aufbau und die Konsistenz einer Blockchain zu verstehen, werden diese Begriffe kurz zusammengefasst und in den nachfolgenden Kapiteln genauer erklärt.

Mit Hashwerten selbst lässt sich die Echtheit von Daten verifizieren, in dem bei der Übermittlung von Daten der Hashwert mitgeliefert wird. Stimmt dieser mit dem selbst berechneten Hashwert überein, so können die Daten als echt deklariert werden.[Dr17, S.90, 91] (siehe Kapitel 5.5.2)

Hashreferenzen stellen die logisch Erweiterung hierzu dar. Der Hashwert wird hier als Verweis auf das jeweilige Datenpaket verwendet. Werden die Daten verändert, kann die alte Referenz die Daten nicht mehr aufrufen.[Dr17, S.101, 102] (siehe Kapitel 5.5.3)

Darauf aufbauend lassen sich Ketten und Bäume erstellen, welche nur dann ihre Integrität bewahren, solange die Daten nicht einseitig verändert werden. Alte Datenbestände können somit nicht unbemerkbar manipuliert werden.[Dr17, S.104, 105] (siehe Kapitel 5.5.4 sowie 5.5.5)

5.5.1 Hashfunktion

Unter einer Hashfunktion versteht man eine Funktion, welche aus einem Eingabewert eine Ausgabe (einen Hashwert) erstellt, sodass aus der Ausgabe nicht mehr der Eingabewert errechnet werden kann.[Dr17, S.91]

Eine weitere Eigenschafft, stellt die Möglichkeit einer Kollision dar. Hashfunktionen bilden eine Eingabe mit beliebiger Länge auf eine Ausgabe mit bestimmter Länge ab. Dh. dass auf eine unendlichen Anzahl an Eingabewerten eine beschränkte Anzahl von Ausgabewerten treffen. Aus dieser Tatsache folgt, dass mehrere Eingabewerte den exakt gleichen Ausgabewert liefern. [NA16, S.2, 3]

5.5.2 Kryptographische Hashfunktionen

Eine besondere Gruppe der Hashfunktionen wird „kryptographische Hashfunktionen“ bezeichnet. Diese müssen folgende Eigenschaften erfüllen: [NA16, S.2] [Dr17, S.90, 91]

- Rasche Berechnung des Hashwerte aus einem beliebig großen Datensatz: Diese Eigenschaft verknüpft zwei Bedingungen: Es soll aus jedem Eingabewert auch ein Ausgabewert errechnet werden können und dies soll möglichst schnell passieren.

- Deterministisch: Aus der Eingabe von identen Daten muss derselbe Hashwert berechnet werden. Es darf somit nicht sein, dass dieselbe Eingabe zu unterschiedlichen Ergebnissen führt.
- Pseudozufällig: Dies bedeutet, dass bei einer Veränderung des Eingabewertes, der Ausgabewert nicht vorhergesagt - dh. der dahinterliegende Algorithmus „entschlüsselt" - werden kann. Ändert sich nur ein Zeichen oder ein Bit bei der Eingabe, so muss das Ergebnis vollkommend „überraschend" ausfallen.
- Einwegfunktionen: Aus dem Ausgabewert darf es nicht möglich sein, den Eingabewert berechnen zu können. Somit ist es unmöglich, aus dem Hashwert die Originaldaten zu generieren.
- Kollisionsresistent: Wie bereits erwähnt, kann es bei der Erstellung der Hashwerte zu Kollisionen kommen. Die Anforderung der Kollisionsresistenz besagt nun, dass es *praktisch*[2] unmöglich sein muss, aus zwei unterschiedlichen Eingaben denselben Ausgabewert zu berechnen. Unterschieden wird zwischen schwacher und starker Kollissionsresistenz. Erstere Eigenschaft besagt, dass es praktisch nicht möglich sein darf zu einem bereits bekannten Wert m0 einen zweiten m1 zu finden, bei denen beide Hashwerte H(m0) = H(m1) ident sind. Die starke Kollissionsresistenz verlangt zusätzlich, dass praktisch unmöglich sein muss, zwei beliebige Werte zu finden, welche den gleichen Hashwert liefern. Der Unterschied zwischen diesen beiden Anforderungen liegt nun darin, dass bei letzterer ein Angreifer oder eine Angreiferin lediglich beliebig viele Eingaben tätigen müsste, um zwangsläufig (und im Durchschnitt $2^{n/2}$, wobei n die Länge des Hashwertes ist) denselben Hashwert zu erzeugen. Um dieses Problem nun

[2] Die Möglichkeit, dass zwei Hashwerte gleich sind ist zwar rein rechnerisch gegeben, aber so gering, dass diese in der Praxis als sehr gering und daher unwahrscheinlich zu betrachten ist.[Dr17, S.91]

auf die geforderte „praktische Unmöglichkeit“ zu reduzieren, muss der Hashwert mindestens doppelt so lang sein, als für einen spezifischen Anwendungsfall benötigt.[Se19]

Aus diesen Eigenschaften ergibt sich letzten Endes, dass jeder Eingabewert einen eindeutiger Ausgabewert liefert, wobei gleichzeitig jeder Ausgabewert nur einem Eingabewert zugewiesen werden kann. Demnach lässt sich ein kryptographischer Hashwert als Fingerabdruck bzw. „Fingerprint“ verwenden. In der Praxis bedeutet dies, dass Datensätze miteinander verglichen werden können, indem man die jeweiligen Hashwerte miteinander vergleicht. [Dr17, S.91–99]

Ausschließlich die kryptographischen Hashfunktionen sind in dieser Arbeit von Relevanz. Wenn nicht ausdrücklich anders erwähnt ist in der restlichen Arbeit mit „Hashfunktion“ und „Hashwert“ immer die kryptographische Variante zu berücksichtigen.

Hashfunktionen lassen sich in beliebiger Reihenfolge miteinander kombinieren. Dies bedeutet, dass man einen Hashwert weitere Male hashen kann, dass mehrere Datensätze gleichzeitig gehasht werden können und dass ein Datensatz mit einem vorhandenen Hashwert gehasht werden kann. All diese Vorgänge liefern einen gültigen Hashwert.[Dr17, S.93–97]

5.5.3 Hashreferenzen

Eine Hashreferenz ist nichts anderes als ein Hashwert, welcher zusätzlich als Referenz (sozusagen als „Ablageort“ oder „Aufbewahrungsort“) auf die Originaldaten verweist. [Dr17, S.101, 102] [DI16, S.108]

Werden nun der Hashwert oder die Originaldaten absichtlich oder unabsichtlich verändert, so kann die Referenz nicht mehr auf die Daten verweisen und verliert somit ihre Gültigkeit. Dies bedeutet, dass bei jeder Änderung der Daten ebenfalls eine neue Hashreferenz erzeugt werden muss. Genau dieser Umstand wird beim Aufbau der Blockchain einen elementaren Bestandteilt darstellen. [Dr17, S.103]

Genau genommen bedient sich diese der verketteten Liste und des Hashbaumes, um die Daten unveränderbar und nachvollziehbar zu speichern. Die nachfolgenden Kapiteln erklären diese Funktionalitäten.

5.5.4 Verkettete Liste

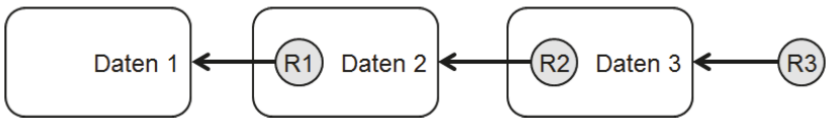

Abbildung 16 - verknüpfte Daten als Kette
Quelle: [Dr17, S.105]

In diesem Fall fungiert jede Hashreferenz als Verweis zu einem vorhergegangenen Datensatz oder auch „Datenglied“. Abbildung 16 veranschaulicht dies auf folgende Art und Weise:[Dr17, S.104, 105]

- Der erste Datensatz beinhaltet noch keine Referenz.
- Datenglied 2 beinhaltet Hashreferenz 1 welche auf Datensatz 1 verweist
- Datenglied 3 beinhaltet Hashreferenz 2 welche auf Datensatz 2 und Hashreferenz 1 verweist
- Hashreferenz 3 verweist auf Datensatz 3 mit Hashreferenz 2, und dient somit als Listenkopf bezeichnet.

Dieser Ablauf lässt sich für theoretisch beliebig viele Datenglieder fortsetzen.

Wichtig hierbei ist, dass durch den Listenkopf auf das zuletzt hinzugefügte Datenglied zugegriffen werden kann und somit in umgekehrter Reihenfolge alle Datenglieder durchlaufen werden können.

Wird nun ein Datenpaket in der Liste verändert, so verändert sich, wie bereits erklärt, ebenfalls dessen Hashwert. Die vorhandene Hashreferenz ist folglich ungültig und kann das Datenpaket nicht mehr aufrufen.

Diese Vorgehensweise findet bei der Verknüpfung der zu Beginn des Kapitels erwähnten Köpfe der Blöcke einer BC Anwendung.

5.5.5 Hashbaum

Der Hashbaum oder auch Merkle Tree stellt einen etwas andere Möglichkeit der Referenzierung bereit. Mit dieser Vorgehensweise ist es möglich, Daten in einem „Block“ zusammenzufassen, sodass eine Hashreferenz als Kopf dient und sich daraus alle anderen Referenzen des Blockes herleiten. (sieh Abbildung 17)

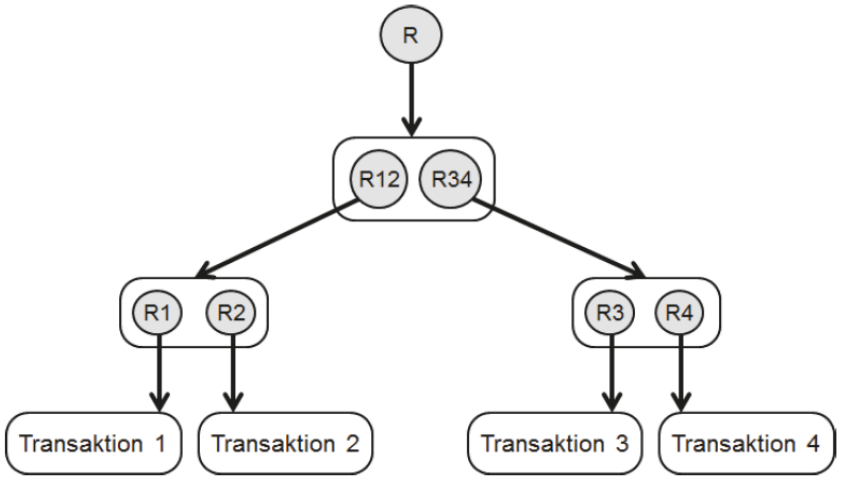

Abbildung 17 - verknüpfte Daten als Baumstruktur
Quelle: [Dr17, S.105]

Wie in Abbildung 17 zu erkennen sind hier nicht mehr einzelne Datenglieder miteinander verknüpft. Stattdessen bedient man sich der hierarchischen Berechnung von Hashwerten. Dies funktioniert, indem zu jedem Datensatz ein eigener Hashwert berechnet wird. Nun werden die Hashwerte paarweise miteinander kombiniert und daraus pro Paar ein neuer Hashwert erzeugt. Aus den beiden neuen Hashreferenzen wird wiederum ein Hashwert berechnet, welcher in diesem Beispiel die oberste Stufe der Hierarchie bildet.[Dr17, S.106]

Kennt man den obersten Knoten, gelangt man durch die weiterführenden Referenzen zu die ursprünglichen Datenpakete.

Wird ein Datenpaket verändert und die Referenzwerte nicht aktualisiert, so verlieren die Hashreferenzen wiederum ihre Gültigkeit, sprich die Daten können nicht mehr aufgerufen werden.

5.6 Accounts

Um Transaktionen bzw. Daten zu speichern und den Usern in der Blockchain zuweisen zu können, müssen diese in einem „digitalen Konto“ („Accounts“) hinterlegt werden. Dies bedeutet, dass jeder Teilnehmer und jede Teilnehmerin mindestens ein Konto benötigt, damit getätigte Transaktionen auch dem User zuordenbar sind.[NA16, S.76, 77]

Die eindeutige Zuweisung bzw. Identifizierung der Konten erfolgt mithilfe der öffentlichen und privaten Schlüssel. Aus dieser Tatsache und der Kombination von PK und SK werden folgende drei Funktionsweisen erreicht:

- **Identifizierung von Benutzerkonten:** Private Keys dienen, wie erwähnt, als Account-Nummer. Dies bedeutet, dass zumindest jeder, der sich des öffentlichen Schlüssels bedient, einen Datenaustausch mit diesem Benutzerkonto initiieren kann. Die zu sendenden Daten (Transaktionen) werden mit der Kontonummer verschlüsselt und können nur mit dem privaten Schlüssel des Kontoinhabers entschlüsselt werden. Hiermit wird sichergestellt, dass jegliche übertragene Geheimdaten ausschließlich von der gewünschten BenutzerIn entschlüsselt und im Originaltext gelesen werden können.[NA16, S.18,19] [Dr17, S.117, 118]
- **Signieren von Transaktionen:** Dieser Vorgang entspricht der in Kapitel 5.3.3.1 beschriebenen Methode der digitalen Signatur. Die AbsenderIn beschreibt alle notwendigen Schritte der Transaktion der Daten. Im

Anschluss wird die digitale Signatur hierzu erzeugt und den Transaktionsdaten beigefügt. [NA16, S.19] [Dr17, S.125, 126]

- **Verifizieren und Autorisieren von Transaktionen:** Analog zu Kapitel 5.3.3.2 können nun die signierten Daten auf Echtheit überprüft werden. Ist dies nicht der Fall, wir die Autorisierung abgelehnt und die Transaktion findet nicht statt. Bei Echtheit wird zusätzlich Überprüft, ob die AbsenderIn über die benötigten Berechtigungen der geforderten Transaktion besitzt. Wenn ja, wird diese durchgeführt. [NA16, S.19] [Dr17, S.126]

5.6.1 Erstellen von Accounts

Die in Blockchain vorherrschende Methode zur Generierung von User-Accounts ist die dezentrale Methode. Dies bedeutet, dass jede Person unabhängig von einer zentralen Stelle einen oder auch beliebig viele Accounts erstellen und verwenden kann. Der User selbst bleibt „anonym“, da für die Blockchain und somit für alle anderen Accounts nur der öffentliche Schlüssel (auch als „Adresse“ bezeichnet) ersichtlich ist. [NA16, S.19, 20]

Die Basis für die Korrektheit bilden die Grunddaten wie Name, Adresse. Dies sind jene personenbezogenen Merkmale, welche eine natürliche Person eindeutig identifizieren. Nur in Kombination dieser beiden Daten kann einer Person auch die korrekte Krankenhistorie zugeordnet werden. Um dies zu gewährleisten benötigt es einer zentralen Stelle (zB einer Behörde oder einer Zertifizierungsstelle[Gl19]), welche die Daten verifiziert. Um initial einen richtigen Datenbestand zu gewährleisten obliegt es dieser zentralen Stelle, die Accounts aller Teilnehmer und Teilnehmerinnen zu erstellen. Hierzu zählt:

- Erstellen des Accounts
- Eingabe der personenbezogenen Daten sowie personenbezogenen Gesundheitsdaten
- Erstellen des PK/SK – Schlüsselpaares
- Zuweisen des PK zum Account

- Übermitteln des SK an den User

Einerseits wird somit die Richtigkeit der eingegebenen Daten und andererseits die Richtigkeit und Echtheit des Verantwortlichen der Blockchain gewährleistet.

5.6.2 Online Vs. Offline

Prinzipiell lassen sich Accounts in Online- und Offline-Account einteilen:[Prus17, S.105, 106]

Online-Account: Ein Account, welcher stets mit dem Internet in Verbindung steht, zB auf Websites und Datenbanken. Hierbei registriert sich ein Benutzer oder eine Benutzerin bei einem Online-Anbieter, welcher alle Transaktionen sowie die Informationen über das SK/PK Schlüsselpaar speichert. Der Vorteil hierbei ist, dass die Daten technisch gesehen abgesichert sind, sofern der ausgewählte Online-Anbieter redundante Backups erstellt. Als Nachteil muss genannt werden, dass man als User dem Anbieter des Accounts vertrauen muss. Es besteht die Möglichkeit, dass der Online-Anbieter die SK mit den PK speichert und somit auch den Account manipulieren kann.[NA16, S.88]

Offline-Account: Ein Account, welcher nicht ständig mit dem Internet in Verbindung steht. zB am persönlichen Rechner, lokal am Smartphone oder ähnlichen privaten Geräten. Der Vorteil hierbei ist, dass man über alle eigene Daten nachvollziehbar verfügt. Der Nachteil hingegen liegt ebenso auf der Hand: Gehen die Geräte verloren oder sind beschädigt, so sind ebenso die Account-Informationen inklusive des privaten Schlüssels verloren.

5.6.3 Wallets

Was passiert nun, wenn man einen Offline Account besitzt, welcher im Moment nicht mit der Blockchain verbunden ist, aber ein andere Knoten eine neue Transaktion übermittelt oder übermitteln möchte? Eine Möglichkeit wäre, einen Online und einen Offline Account zu besitzen. Ersteres erhält die Daten welche zu einem späteren Zeitpunkt auf zweiteres übertragen werden. Konsequenter Weise

benötigt man hier für je ein Schlüsselpaar. Rein technisch wird einfach eine Transaktion von einem zum anderen Knoten initiiert, wobei beide derselben Person gehören. Es ist somit nicht mehr notwendig, dass jeder Knoten zu jeder Zeit online sein muss.[NA16, S.76, 77] Eine Wallet stellt somit die Möglichkeit dar, seinen Account, bzw. jene Transaktionen, welche zwischen Offline und Online Account transferiert wurden, auf tragbare Medien zu speichern.

5.6.4 Hierarchisch Deterministische Wallets

Eine Erweiterung der Wallets stellen die „hierarchisch deterministischen" Wallets dar. Hierbei wird kein reines SK/PK Schlüsselpaar, sondern mithilfe eines Algorithmus eine theoretisch unendlich lange Serie von Schlüsselpaaren für denselben Account erzeugt. Hierbei existiert eine Funktion „Private Key Generation Info" (PKGI) und eine weitere „Address Generation Info" (AGI), welche beide einen Parameter **x** als Eingabe erhalten. Die Funktion „Private Key Generation Info" erstellt einen SK an der x-ten Stelle, die „Address Generation Info", wiederum erstellt den zugehörigen PK an der x-ten Stelle. Wichtig hierbei ist, dass unterschiedliche Adressen, welche mit demselben AGI generiert wurden, nicht in Zusammenhang gebracht werden können. Daher es lässt sich für einen Beobachter nicht feststellen, dass sich diese Adressen auf dieselbe PKGI beziehen. [NA16, S.80]

In der Praxis wird PKGI analog zum SK geheim gehalten. AGI hingegen wird veröffentlicht. Dadurch kann jede Person oder jeder Knoten mit Hilfe eines AGIs eine Transaktion an die x-te Adresse eines zugehörigen Schlüssels schicken. Der Empfängerknoten sieht nun an der x-ten Stelle nach und kann die Transaktion empfangen/bestätigen.

Diese Art der Wallets kann zur Anonymisierung der Daten verwendet werden, da bei geschickter Anwendung nicht mehr nachzuvollziehen ist, welche Transaktionen an wen übertragen wurden. Der genaue Ablauf wird in Kapitel 6.2.6 erklärt.

5.6.5 Skripts

Letzten Endes sollen mit Hilfe der Blockchain – Technologie Daten ausgetauscht werden. Dies erfolgt mit Hilfe von Skripts (in den nachfolgenden Kapiteln 5.6.6 und 5.6.7genauer beschrieben) und wird Transaktion genannt. Genau genommen werden bei jeder Transkation zwei Skripts miteinander kombiniert und ausgeführt, nämlich das Skript des Absender-Knotens und jenes des Empfänger-Knotens. Nur wenn die Kombination dieser beiden Skripts erfolgreich verläuft, wird eine Transaktion durchgeführt und als gültig bezeichnet.[NA16, S.55]

5.6.6 Skripts an Hand des Beispiels von Bitcoin

Was genau sind nun Skripts? Wenn wir „Bitcoin“ als Referenz heranziehen, handelt es sich hierbei um eine Abfolge von (einfachen) Befehlen, welche der Reihe nach (ohne Schleifen) ausgeführt werden. Die Skript-Sprache (einfach „Script“ genannt) ist mit Absicht einfach und restriktiv gehalten, um unkontrollierbare Ergebnisse zu vermeiden.

Folgender (vereinfachter) Basisablauf (definiert in Kapitel 5.6) trifft auf jede Transaktion zu:[NA16, S.57, 58]

- Absender-Skript:
    - Hinzufügen der (verschlüsselten) Daten, welche dem neuen Konto hinzugefügt werden sollen (zB: die Anzahl der zu übertragenden Bitcoins)
    - Verifizieren der Adresse des Empfänger-Kontos
    - Hinzufügen der digitalen Signatur des Absenders, um die Echtheit festzuhalten.
- Empfänger-Skript:
    - Adresse
    - Verifizieren der digitalen Signatur des Absenders
    - (Entschlüsseln der Daten)
    - Hinzufügen der Daten zum Account

Bei Bitcoin existiert zusätzlich eine „Whitelist“. Diese definiert, welche Skript-Arten ausgeführt werden dürfen. Etwaige Missbräuche oder Fehlerquellen werden hierdurch minimiert.[NA16, S.59]

Das Übermitteln von Daten in Script besitzt Limitierungen. Die Hauptfunktionalität stellt das Übertragen von Bitcoins dar. Etwaige andere Daten werden grundsätzlich nicht unterstützt. Es besteht aber die Möglichkeit, andere Daten in die Bitcoin-Blockchain Struktur zu schreiben, ohne dass diese wirklich übertragen werden. Der Befehlt „OP_RETURN“ beendet jedes Skript, egal welche Befehle und Daten in weiterer Folge angeführt werden. Opfert man eine kleine Menge an Bitcoins („Proof of Burn“) so kann man nach dem Ausführen von „OP_RETURN“ prinzipiell jegliche Daten (Zeitstempel, Hashreferenzen, Links) in die Blockchain-Struktur schreiben.[NA16, S.59]

Zu beachten ist, dass weder die Kryptowährungen Bitcoin noch Ethereum die Daten selbst verschlüsselt sondern lediglich „signiert“. Die Möglichkeit der Enkryption besteht. Als Beispiel sei „Zcash“, eine relative neue Kryptowährung, genannt.[DI16, S.105]

5.6.7 Smart Contracts

Eine logische Erweiterung zur vereinfachten Skript Sprache von Bitcoin stellt die Sprache in Ethereum dar. Der Begriff „Smart Contracts“ existiert hierbei seit 1994 und wurde von Nick Szabo das erste Mal verwendet. Er sprach hierbei von Programmen, welche rechtliche Sachverhalte ausführen sollten.[DI16, S.166]

Prinzipiell sind „Smart Contracts“ dezentrale Programme, dh sie laufen, analog zu Skripts, auf allen Knoten der Blockchain ab und können daher weder unterbrochen noch verändert werden. Des Weiteren bieten sie zusätzliche Programmiermöglichkeiten, wie Schleifen und sollen somit die Limitierungen von anderen Skripts, wie jene von Bitcoin aufheben.[DI16, S.164, 167]

Mit Hilfe von „Smart Contracts“ lassen sich digitale sowie (konsequenter Weise) reale Güter liefern. Die zugehörigen Daten (bzw. die Hashreferenz) werden hierbei einfach in die Transkation geschrieben, und erst dann abgearbeitet, wenn die Bedingungen im Code erfüllt wurden.[DI16, S.169, 170]

Es ist somit nicht mehr notwendig, Geld zu opfern (in Form von Proof of Burn) bzw. Daten in unerreichbarem Code (dh. Code, welcher nicht mehr ausgeführt werden kann, da das Skript auf jeden Fall zuvor terminiert wird) zu schreiben.[NA16, S.59] Ethereum unterstützt nicht nur diese Funktionalität, sondern wurde genau für solche Anwendungsfällte geschaffen.[DI16, S.164]

Diese Flexibilität birgt konsequenterweise auch Gefahren. Da es sich um einen „Turing-Complete“ Code handelt [NA16, S.263], sind alle Konsequenzen aller Bedingungen nahezu unmöglich vorauszusehen. Als Beispiel sei Christoph Jentzsch, ein theoretischer Physiker, genannt, welcher auf Grund eines Logikfehlers seines „Smart Contracts“ 50,000,000$ verlor.[DI16, S.53, 54]

5.7 Speicherung der Transaktionen

Die Informationen (seien es nun Bitcoins oder andere Daten), welche mit Transaktionen übermittelt werden, müssen letztlich auch einem Account zugewiesen werden. Hierbei haben sich zwei unterschiedliche Arten etabliert, auf welche in den nachfolgenden Kapiteln eingegangen wird.

5.7.1 Transaktions-Basierender Ledger

Als erstes wird die Methode welche von Bitcoin verwendet wird beschrieben. Streng genommen werden Bitcoins dabei nicht dem Empfänger-Account „hinzugefügt“ und vom Absender-Account „abgebucht“. Ebenso wenig werden die einzelnen Transaktionen nicht im Account selbst gespeichert. Der Account beinhaltet demnach nahezu keine Informationen. Stattdessen beinhaltet die letzte Transaktion den vollständigen Kontostand des Absenders, wobei die zu übertragende Menge an den Empfänger-Account und die restliche Summe an den Absender-Account

„überwiesen“ wird. Dadurch reicht ein Blick auf die letzte Transaktion aus, um einen gültigen Kontostand eines jeden Accounts zu eruieren.[NA16, S.51, 52, 53]

5.7.2 Account-Basierender Ledger

Als gegenteiliger (und vielleicht intuitiverer) Ansatz sei das System von Ethereum erwähnt. Hierbei werden tatsächlich in einzelnen Transaktionen lediglich die zu übertragenden Daten „befördert“. Will man nun einen vollständigen Bestand des Accounts ausweisen, so bleibt einem nichts anderes übrig, als alle Transaktionen zu aggregieren und daraus eine Bilanz zu ziehen. [NA16, S.51, 52, 53]

5.8 Block-Struktur

Nachdem nun Transaktionen sicher getätigt wurden, müssen diese ebenso sicher und nachvollziehbar gespeichert werden. Dies geschieht durch die Kombination eines Hashbaumes und einer verlinkten Kette. Der Hashbaum fasst Datenpakete zusammen, während die Liste die einzelnen Ketten miteinander verknüpft, wobei beide Verfahren nicht separat voneinander angewendet werden sondern eine Einheit dar stellen. Dies lässt sich folgendermaßen erklären:

Die getätigten Transaktionen werden mit Hilfe von Hashreferenzen miteinander verknüpft. Liegt eine gewisse Anzahl an Transkationen vor, wird daraus ein „Block“ erstellt. Dies passiert analog zur Funktionsweise des bereits beschriebenen „Merkel Trees“ und der verlinkten Kette [Bi15, S.7] (siehe Abbildung 18).

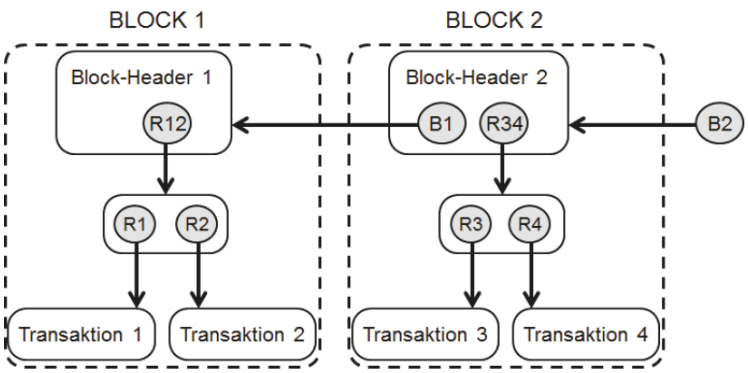

Abbildung 18 – Datenstruktur (schematisch)
Quelle: [Dr17, S.138]

Zu erkennen ist, dass ein Header eines Blockes (mindestens) die Referenz des vorhergegangenen Blockes sowie die Referenz zu den eigenen Daten besitzen muss. (Ausnahmen sind auch hier der erste Block, welcher keinen Vorgänger besitzt, sowie der „Kopf" der gesamten Blockchain (in diesem Fall „B2"), welcher noch keinen Verweis auf Daten besitzt.

Werden neue Transaktionen erstellt beginnt der Ablauf von Vorne: zu den neuen Daten werden Referenzen erstellt und diese wiederum referenziert. Der neue Block-Header setzt sich abermals aus dem ursprünglichen „Kopf" der Kette und der Referenz der neuen Daten zusammen. Im Anschluss wird ein neuer „Kopf" erstellt, welcher auf den letzten Block-Header verweist. [Dr17, S.142, 143]

5.9 Verändern von Daten oder Referenzen

Werden in der Blockchain (unsachgemäße) Veränderungen (Manipulationen) vorgenommen, so können diese aufgrund der Hashreferenzen erkannt werden.

Hierbei ist es irrelevant, wo diese Veränderung stattfindet. Wird, wie in Abbildung 19 dargestellt, eine Manipulation bei einer der ersten Transaktion getätigt, so ist die zugehörige Referenz, und somit die gesamte Kette der Blockchain ungültig. Wird nun die zugehörige Referenz (in Abbildung 19 als R2 bezeichnet) ausgebessert, so

verliert Block-Header 1 seine Gültigkeit. Wird dieser ebenso ausgebessert, so verliert Block-Header 2 seine gültige Referenz auf Block-Header 1. Aus dieser Veranschaulichung lässt sich folgende Verallgemeinerung festhalten: Je mehr Blöcke eine Blockchain besitzt und je älter eine Transaktion ist, desto schwieriger wird es, eine Manipulation vorzunehmen. [Dr17, S.148]

Dem „Kopf“ der gesamten Blockchain kommt dadurch eine doppelte Wichtigkeit zu: Einerseits stellt dieser die „letzte Bastion“ vor Manipulation dar, andererseits ermöglicht er es, neue Daten hinzuzufügen.

Wer den Kopf der Blockchain beherrscht, kontrolliert ebenso die Blockchain und dessen Inhalt. Diese Erkenntnis führt unweigerlich zur Frage, wer diesen „Kopf“ definieren oder editieren darf. Oder umgekehrt formuliert: Wie kann sichergestellt werden, dass der „Kopf“ nicht unsachgemäß verändert wird? Die Antwort liegt im Erreichen eines „Konsens“. Die Lösung zu diesem Problem findet sich im Kapitel 5.10.

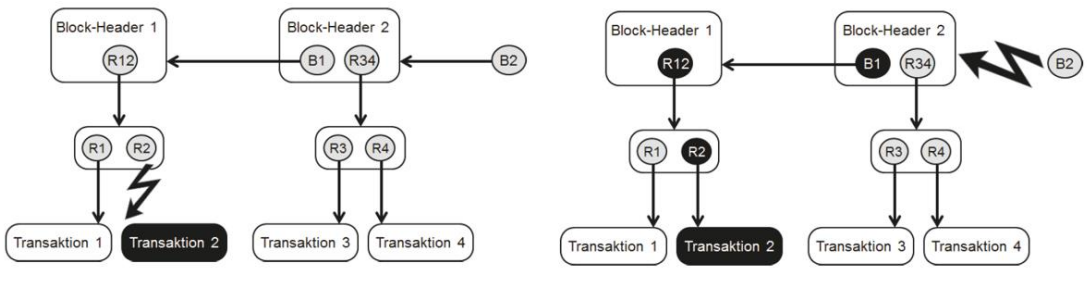

Abbildung 19 - Änderungen in der Blockchain
Quelle: In Anlehnung an [Dr17, S.144, 147]

.

5.10 Konsens

Wurde nun ein neuer Block erstellt, muss auf irgendeine Art und Weise verifiziert werden, dass es sich hierbei auch um einen gültigen Block der Blockchain handelt. Damit wird verhindert, dass eine absichtliche oder unabsichtliche Manipulation des Blockchain-Datenbestandes vorgenommen wird. Es bedeutet, dass die beteiligten Knoten zu einem „Konsens“ (englisch „consensus“ [NA16, S.28]) über die gültige Blockchain-Struktur gelangen. Der neue Block und somit alle verbundenen Daten werden als richtig anerkannt und die Integrität der Blockchain ist bestätigt.[NA16, S.29, 30]

Um nun bestmöglich zu einem „Consensus“ zu gelangen, stehen verschiedene Methoden zur Verfügung. Die gängigsten, welche für das Verständnis des Konzeptes und für diese Arbeit von Bedeutung sind, werden in den nächsten Unterkapiteln erklärt.

5.10.1 Proof of Work (PoW)

Bei dieser Methode wird der Konsens algorithmisch bzw. „trustless“ erreicht. [Bi15, S.11] Dies bedeutet: Um einen neuen Block hinzufügen zu können, muss eine gewisse Rechenleistung erfolgen. Im derzeit bekanntesten Anwendungsfall - der Blockchain „Bitcoin“ - erfolgt dies durch das Lösen (auch „minen“) eines „Hash-Puzzles“, welches nicht durch Logik sondern nur durch „brute force“, errechnet werden kann. Knoten, die sich am Lösen eines Puzzles beteiligen (sogenannte „Miner“), stellen ihre Rechenleistung zur Verfügung. Wurde das „Hash-Puzzle“ gelöst, wird die Richtigkeit von anderen „Minern“ verifiziert und bei Richtigkeit der neue Block zur Blockchain hinzugefügt. Um nun das Errechnen von **gültigen** Blocks lukrativ zu gestalten, erhalten „Miner“, welche einen gültigen Block erstellen, einen Incentive (in diesem Falle „Bitcoins“). Somit ist jeder „Miner“ gezwungen, in seinem eigenen besten Interesse zu handeln, was eine Bildung von Allianzen und Bündelung von Rechenleistungen (sogenannter 51% Angriff [NA16, S.48]) zur

Manipulation verhindert. Solange mindestens 50% aller Knoten vertrauenswürdig sind, bleibt die Integrität gegeben. [NA16, S.41, 43]

Würde ein einzelner Miner versuchen, eine getätigte Transaktion aus der Historie zu löschen oder zu ändern, so müsste er gezwungener Maßen alle Hashreferenzen ändern und für alle bereits berechneten Blöcke und für den neuen Block die Hash-Puzzles im Alleingang neu berechnen, während die restlichen Miner „nur" den neuen Block fertig stellen bräuchten. Je nach Verschlüsselungsart der Transaktionen selbst (zB Private Key), müsste der böswillige Knoten ebenso alle beteiligten privaten Schlüssel aushebeln, um korrekte Hashwerte zu errechnen. Dieses Konzept des „Proof of Work" stellt das Hauptkonzept der Unveränderlichkeit dar.[NA16, S.48, 49]

Abgesehen vom steigenden Energieverbrauch [NA16, S.206, 207] kann die Methode zur Erreichung des Konsens im allgemeinen Kontext als sicher eingestuft werden, solange ausreichend unterschiedliche Knoten an der Berechnung der Hash-Puzzles teilnehmen.

5.10.2 Merged Mining

Hinter diesem Begriff steckt die Idee, eine Blockchain A mit einer anderen (möglichst etablierten und sicheren) Blockchain B zu verifizieren, solange beide denselben PoW Mechanismus benutzen. Erreicht wird dies mit einer „Witness Transaction". Dies bedeutet, dass ein Block A nur dann als gültig betrachtet wird, wenn dieser ebenfalls das PoW-Protokoll von B erfüllt und anhand des PoW von A erstellt wurde. [Bi15, S.15, 16]

Hierzu wird folgendes Beispiel angeführt:

„Newchain" ist eine neue Blockchain mit derzeit stark limitierter User-Anzahl und dem POW – Mechanismus von „Bitcoin". Um das „Proof of Work" Protokoll vor potentiellen „51%" Attacken zu schützen, werden Transaktionen (genau genommen der Hash des Headers des Block, welcher die Transaktionen enthält) aus Newchain

an Bitcoin „übermittelt“. Dieser Hash wird in einer Bitcoin Transaktion „eingebettet“. Nun wird der Block des Bitcoins-Headers errechnet und als gültig verifiziert. Erfüllt der Bitcoin-Header die Anforderungen unsererer „Newchain“, kann dieser für die gültige Erstellung des neuen Blocks auf „Newchain“ verwendet werden. Dieser Vorgang setzt voraus, das die verfügbare Rechenleistung auf beiden Blockchains aufgeteilt werden muss.

Dieser Vorgang verifiziert zwei Sachverhalte:

- Die Daten sind zum Zeitpunkt des erstellten Blockes korrekt
- Die Korrektheit der Daten kann zu einem späteren Zeitpunkt via Bitcoin verifiziert werden.

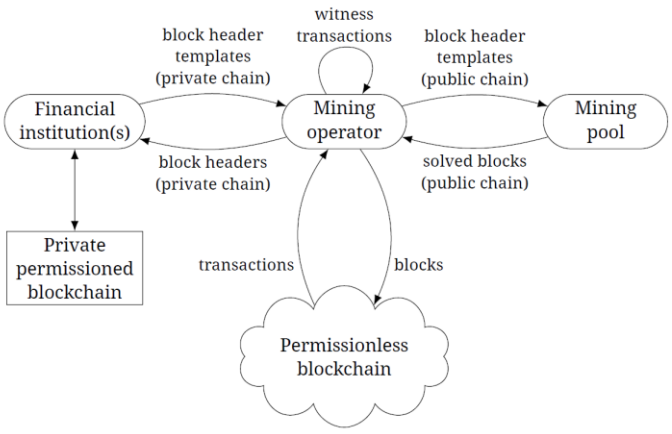

Abbildung 20 - Merged Mining
Quelle: [Bi15, S.17]

In unserem theoretischen Beispiel ist es notwendig, zur Validierung der „Newchain“ gültige Blöcke von Bitcoin zu berechnen. Auf Grund des „PoW“ ist nicht garantiert, in welchem Intervall oder ob dies überhaupt gelingt. Eine bessere Methode wäre daher, Daten rein als Transaktionen in eine andere Blockchain zu schreiben und dies als „Times Tamp“ zu benutzen. Hierfür bietet sich die „Blockchain Anchoring“ an welche im nachfolgenden Kapitel 5.10.3 beschrieben wird.

5.10.3 Blockchain Anchoring

Eine ähnliche Vorgehensweise zu „Merged Mining“ stellt das „Blockchain Anchoring“ dar. Hierbei wird wiederum ein Hash des erstellten Blockes von Blockchain A in einen Block von Blockchain B geschrieben. Der Unterschied hierbei liegt daran, dass nicht dieselben PoW Mechaniken oder generell dieselben Konsens-Methoden verwendet werden müssen. Der Eintrag des Hashes in Blockchain B verifiziert lediglich die Existenz dieses Knoten zu diesem Zeitpunkt. Das Erstellen eines gültigen Blockes in der Blockchain A muss durch ein separates Protokoll gewährleistet sein. Dies wirkt zunächst wie ein Nachteil, kann aber besonders für private und genehmigungspflichtige Blockchains genutzt werden, welche intern ein eigenständiges Konsens-Protokoll nutzen. Überdies reicht es aus, nur wenige Blöcke pro Tag zu übermitteln, um gesamtheitlich zu einem Konsens zu gelangen. Will nun ein Angreifer oder eine Angreiferin eine Manipulation in Blockchain A vornehmen, müsste er zwei unterschiedliche Konsens-Methoden umgehen. [Bi15, S.17, 18, 19]

Streng genommen handelt es sich um kein reines PoW – Protokoll da auch Blockchain B eine andere Art des Proofs verwenden könnte. Da es sich aber um die Verknüpfung mit einer öffentlichen Blockchain handelt, liegt hier ebenfalls der Schluss nahe, sich mit der größten, etabliertesten und sichersten Ausprägung, Bitcoin, zu verknüpfen.

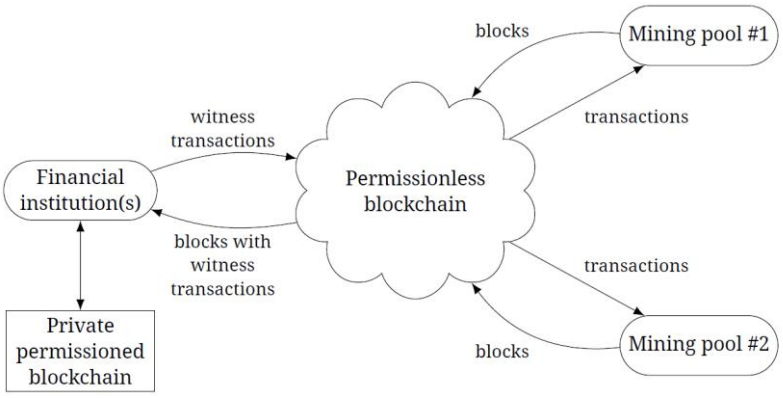

Abbildung 21 - Blockchain Anchoring
Quelle: [Bi15, S.19]

5.10.4 Mining Rotation

Eine alternative Weise, einen Konsens zu erreichen stellt der Ablauf der „Mining Rotation“ dar. Hierbei existieren eine gewisse Anzahl an „Minern“, welche in diesem Fall zur Erstellung eines Blockes den Header mit ihrer digitalen Signatur versehen. Gleichzeitig erfolgt die Erstellung der Blocks in einer zeitlich vorgegebenen Rotation (zB: jeder Miner hat 10min Zeit, um einen Block zu erstellen). Wurde der Block nicht innerhalb dieses Zeitfensters erstellt, springt der Vorgang des Erstellens zum nächsten Miner, welcher seine Transaktionen verarbeiten und den Block-Header kreiert und signiert. Der Block-Header wird nun im Anschluss von allen anderen „Minern“ verifiziert. Ist dieser Ablauf erfolgreich, handelt es sich um einen neuen, gültigen Block der Blockchain. Auffälliges Verhalten, (keine Partizipation beim Erstellen von Blöcken, oder Erstellen von ungültigen Blöcken) kann schnell erkannt und dem Grund nachgegangen werden.[Bi15, S.12, 13]

Will nun ein Angreifer oder eine Angreiferin alte Blöcke ändern, so bräuchte dieser alle privaten Schlüssel der Miner. In der Theorie wäre dieser Vorgang sicherer als PoW, sofern es sich bei allen Teilnehmer und Teilnehmerinnen um eine heterogene Masse handelt. Die Miner operieren hierbei im eigenen Interesse und signieren und verifizieren nur, wenn die Daten nicht manipuliert wurden. Die Situation wird etwas undurchsichtiger für Knoten mit reiner Leseberechtigung. Da sie am Verifizierungsprozess nicht teilnehmen, besitzen sie keine Möglichkeit, über die Gültigkeit der Blockchain Einsicht zu nehmen. Streng genommen könnten die „Miner" unterschiedliche Blockchains betreiben und je nach Bedarf vorweisen oder rückwirkend ändern. Selbstverständlich müssten hier alle „Miner" gleichzeitig und für den gleichen Zweck manipulieren. Egal, wie wahrscheinlich oder unwahrscheinlich dieses Szenario sein mag, es ist für Teilnehmer oder Teilnehmerinnen mit Leseberechtigung nicht auszuschließen.[Bi15, S.13]

Die Methode des Block Anchoring schafft hierbei die nötige Abhilfe. Die Hashcodes der Blöcke sind in einer öffentlichen Blockchain für jede Person einsehbar und somit auch vergleichbar. Dadurch lassen sich getätigte Transaktionen nicht mehr „heimlich" ändern.

5.10.5 Proof of Stake (PoS)

Letztlich sei noch die Methode des „Proof of Stake" erwähne. Es stellt eine weitere Alternative zum Vorgang des „minens", das „virtual Mining" dar. Hierbei ist die Berechnung des Hashpuzzles - zumindest in der Theorie - unabhängig von der Rechenleistung des Knotens. Die Lösung des Puzzles steht nicht mehr in Abhängigkeit von der verwendeten Hardware und gibt dadurch jeden Knoten die gleiche Chance, ein Puzzle zu berechnen. Anstelle der Rechenleistung wird ein „Stake" herangezogen. Je höher dieser „Stake", welcher meist in Form der Kryptowährung gesetzt wird, desto einfacher wird es für den Knoten, das Puzzle zu lösen. [NA16, S.208, 209]

Dies hat aber unweigerlich zur Folge, dass „reiche“ Accounts immer die leichtesten Puzzle zur Berechnung erhalten und dadurch immer reicher werden. Ein Extremfall dieses Effektes ist jener, dass ein „Miner“, welcher 51% des gesamten Stakes erlangt, die Blockchain theoretisch immer nach eigenem Ermessen kontrollieren kann, in dem er auf seinen eigens berechneten Blöcken weiterrechnet. Dieser „Miner“ wäre dadurch in der Lage, 100% des gesamten Stakes zu erlangen. [NA16, S.210, 211]

Ein weiteres Problem stellt die Situation des „nothing-at-stake“ dar. Eine Mining Station, welche seine Ressourcen dazu aufwendet, um eine alternative Blockchain (mit gefälschten Transaktionen) zu erstellen, würde bei einem PoW Protokoll ihre Rechenleistung verschwenden, um diese Blockchain zu erstellen. Schlägt der Versuch fehl, hätte sie erhebliche Zeit und Rechenleistung verschwendet, ohne einen Benefit zu erhalten, während andere Accounts ihren Incentive erhielten. Beim PoS kann der Miner ohne weiteres seinen Stake den auf den gültigen zu lösenden Block setzen und gleichzeitig diesen Stake zur Berechnung einer zweiten Blockchain verwenden. Schlägt der alternative Versuch fehl, wurden keine Ressourcen verschwendet, da der doppelte Stake nirgends aufscheint. Es zahlt sich demnach sogar aus, die Blockchain zu spalten. [NA16, S.209]

Streng genommen wurde weder die Sicherheit von PoW, noch von PoS nachgewiesen. PoW etablierte sich über die Jahre hinweg und hielt bis jetzt jeglicher Kritik stand, während das Konzept des Proof of Stake in der Praxis noch sehr unerforscht ist. [NA16, S.2011]

6. Feststellen der Anforderungen

Nachdem die theoretischen Aspekte festgehalten wurden, sind diese mit einander in Bezug zu bringen und die Anforderungen an die Blockchain Struktur anhand der Anforderungen zur Erreichung des Zieles dieser Arbeit zu definieren. Aus diesem Ergebnis wird im nachfolgenden Kapitel ein Referenzmodell erstellt und validiert.

6.1 Anforderungen an Patientendaten

Wie in Kapitel 4.3.5 zusammengefasst können für diese Arbeit folgende rechtliche Anforderungen an Patientendaten festgehalten werden: Einerseits dürfen Patientendaten nicht an Unbeteiligte weitergegeben werden. Das bedeutet, dass nur GDAs, welche unmittelbar mit dem Patienten oder der Patientin in einem konkreten Behandlungsfall zu tun haben, die zugehörigen Daten (und nur diese) einsehen dürfen. Alle andere Informationen müssen geschützt sein. Die Berechtigung zu Einsicht von Patientendaten muss zeitlich begrenzt sein. ELGA definiert hierbei einen Zeitraum von 28 Tagen. Ebenso besitzen Patienten das Recht, ihre persönlichen Daten ändern zu lassen, oder auch selbst zu ändern. Bereits getätigte Transaktionen dürfen auch rechtlich gesehen nicht verändert werden. Ergeben sich neue Erkenntnisse bei einer Behandlung, ist zu dem vorherrschenden Datensatz ein neuer, aktueller Datensatz anzulegen. Ebenso besitzen alle in Verbindung stehenden GDAs und die Patienten das Recht, ihre Daten einsehen zu dürfen.

6.2 Anforderungen an die Blockchain

Da, wie in Kapitel 5 beschrieben, unterschiedliche Möglichkeiten existieren, eine Blockchain aufzubauen, müssen aus den zuvor beschriebenen Anforderungen an Patientendaten Anforderungen an die Blockchain Struktur definiert werden.

Zur klaren Trennung werden diese in funktionale und nicht-funktionale getrennt. Die funktionalen Anforderungen beschreibt, welche Funktionalitäten ein System rein technisch zu erfüllen hat (zB.: Speichern von Daten). Die nicht-funktionalen

Anforderungen halten wiederum fest wie etwas gemacht wird (zB: Benutzeroberfläche oder Datenintegrität). [Dr17, S.25]

6.2.1 Technische Anforderungen

Formal beschrieben, müssen Patienten und GDAs ihre Accounts erstellen, editieren und löschen können. ELGA-GDAs müssen verschiedene Befunde erstellen, übermitteln sowie lesen können. Patienten haben ebenso das Recht darauf, ihre vollständige Patientenakte einzusehen.

6.2.2 Nicht-Technische Anforderungen

Bereits erstellte Befunde dürfen nicht mehr verändert werden (bei einer Änderung der Behandlung oder ähnliches, ist einfach ein aktueller Befund hinzuzufügen). Weiters dürfen Befunde (sowie Accounts) nur von den zu behandelnden GDAs und dem jeweiligen Patienten bzw. der jeweiligen Patientin eingesehen werden.

6.2.3 Ableitung der ersten UML-Klassen

Aus den technischen und nicht-technischen Anforderungen lassen sich die ersten groben UML Klassen und deren Aufgaben ableiten:

Account:

Jeder Teilnehmer benötigt einen Account, um seine Aufgaben erfüllen zu können. Der Account benötigt somit mindestens einen Namen und einen „Unique Identifier“ (UID) als Schlüsselwert.

e-Befund:

Im e-Befund werden alle Informationen (Entlassungsbrief, e-Medikation, Pflegesituationsbericht, Laborbefund, Befund Bildgebende Diagnostik, Patient Summary, Ärztlicher Befund (generisch), Pathologiebefund, Augenbefund[Hl19, S.7]) sowie Absender und Empfänger gespeichert. Es ist die Transaktions-Art der Blockchain.

GDA:

Ein GDA ist eine Form des Accounts, welche besondere Berechtigungen benötigt. Ein GDA muss Befunde erstellen, lesen und versenden können. Ebenso müssen Account-Informationen stets aktualisiert werden können.

Patient:

Ein Patient (wobei diese UML Klasse einen Patienten sowie eine Patientin beinhaltet) ist im Großen und Ganzen passiver Teilnehmer. Er oder sie muss aber die Möglichkeit besitzen, die Account-Informationen stehts aktuell zu halten und daher zu editieren.

Aufbauend auf diese Klassen und aus den Anforderungen aus Kapitel 6.2.1 sowie 6.2.2 können nun weitere Anforderungen erschlossen werden.

6.2.4 Auswahl der Blockchain Struktur für Patientendaten

Zunächst stellt sich die Frage, welche Art der Blockchain, wie in Kapitel 5.2 beschrieben, für die Verarbeitung von Patientendaten nach den erhobenen Anforderungen.

Auf die Blockchain-Technologie angewandt bedeutet dies:

- Daten dürfen nur von einem eingeschränkten Benutzerkreis (Patienten und Verantwortliche der Blockchain) eingesehen werden.
- Daten dürfen weiters nur von einem eingeschränkten Benutzerkreis (Verantwortliche der Blockchain) editiert, gelöscht oder hinzugefügt werden.
- Daten in der Blockchain dürfen nicht mehr verändert werden. Dies ist implizit bei jeder Blockchain gelöst (siehe Kapitel 5.9).

Stellt man die oben erwähnten Arten von Blockchains gegenüber diesen Anforderungen, so ergibt sich die in Tabelle 1 dargestellte Anforderungsmatrix.

Öffentliche und genehmigungsfreie Blockchains sind für jede Person zugänglich. Jeder Mensch kann einen Account erstellen und jeder Account besitzt die Berechtigung, alle Daten zu lesen, sowie neue Transaktionen und Blöcke zu erstellen. Da beides im exakten Widerspruch zu den beschriebenen Anforderungen für Patientendaten steht, eignet sich diese Blockchain nicht für die Umsetzung dieser Arbeit.

Öffentliche und genehmigungspflichtige BC bieten zwar die Möglichkeit das Erstellen und Überprüfen einzuschränken. Dennoch dürfen alle Anwender auch alle Daten einsehen. Auch dies erfüllt nicht alle Anforderungen der DSGVO. Diese Art der Blockchain kann für diese Arbeit somit nicht verwendet werden.

Private und genehmigungsfreie Blockchains bieten den Vorteil, dass die Daten nicht von allen Anwendern einsehbar sind, allerdings darf jeder Knoten seine eigenen Transaktionen erstellen. Dies würde bedeuten, dass Patienten anonym ihre eigenen Rezepte oder Befunde kreieren könnten.

Private und genehmigungspflichtige Blockchain bieten sowohl die Möglichkeit, die Leseberechtigung einzuschränken dh. Daten zu anonymisieren, als auch die Schreibrechte auf gewisse Knoten (zB Ärzte, Apotheken) zu beschränken. Hierdurch werden alle Anforderungen für Patientendaten laut DSGVO erfüllt. Folglich eignet sich die private und genehmigungspflichtige Blockchain Struktur für die Verarbeitung von Patientendaten.

Aus dieser Gegenüberstellung ist ersichtlich, dass sich die private und genehmigungspflichtige Blockchain-Technologie als *einzige* eignet, um Patientendaten gesetzeskonform zu verarbeiten. Auf die restlichen Möglichkeiten, deren Schwächen oder Sinnhaftigkeit wird im weiteren Verlauf nicht eingegangen.

Tabelle 1 – Lese- und Schreibrechte unterschiedlicher Blockchains

	Eingeschränktes Lesen	Eingeschränktes Schreiben
Öffentlich & genehmigungsfrei		
Öffentlich & genehmigungspflichtig		X
Privat und genehmigungsfrei	X	
Privat und genehmigungspflichtig	X	X

Die Berechtigungen können auf verschiedene Weise aufgeteilt werden. Eine Möglichkeit wäre: [Bi15, S.10]

- Leseberechtigung: Legt fest, welche Knoten welche Daten einsehen können. Mögliche Aufteilungen der Berechtigungen:
    - …der eigenen Transaktionen: zB: Patienten, die nur ihre eigenen Akte einsehen können
    - …eines gewissen Bereich der Historie: zB: Ärzte, die alle Daten ihrer behandelnden Patienten einsehen können
    - …der gesamten Transaktionshistorie: zB: Behörden, die alle Daten einsehen dürfen und müssen.
- „Thin Clients“: [NA16, S.71] Knoten, welche nicht die gesamte Blockchain, sondern nur Daten der eigenen Transaktionen sichern und nicht am Validierungsprozess teilnehmen. Hierbei handelt es sich um einen Spezialfall des vorherigen Punktes.

- Erstellen neuer Transaktionen: Legt fest, welcher Userkreis neue Transaktionen erstellen darf, zB: Ärzte.
- Erstellen neuer Blöcke: Legt fest, wer letztendlich Transkationen zu Blöcken zusammenfassen und diese zur Blockchain hinzufügen darf, zB: Behörden oder andere autorisierte Stellen.

Die Berechtigungen für den Lesezugriff sowie das Erstellen von Transaktionen wurde bereits geklärt. Der Bedarf von Thin Clients und die Berechtigung zum Erstellen der Blöcke wird in den nächsten Kapiteln eruiert.

6.2.5 Auswahl der Konsens-Methode für Patientendaten

Eine nicht-technische Anforderung lautet, dass bereits erstellte Befunde nicht mehr verändert werden dürfen. Dies bedeutet, dass ein Konsens-Protokoll benötigt wird, welches sich für die vorliegende Arbeit eignet.

In den Kapiteln 5.10.1, 5.10.2, 5.10.3, 5.10.4 und 5.10.5 wurden diese bereits beschrieben. Im nachfolgenden Abschnitt werden die einzelnen Methoden auf dessen Anwendungsmöglichkeit in Bezug zur Verarbeitung von Patientendaten analysiert und eine geeignete Methode ausgewählt.

6.2.5.1 Proof of Work

Das Konzept des „Proof of Work“ geht davon aus, dass jeder Knoten für sich dh. im eigenen besten Interesse handelt und somit eine Bildung von Allianzen unwahrscheinlich und uneffektiv bleibt. (siehe Kapitel 5.10.1)

Im Falle dieser Arbeit existiert keine heterogene Menge. Praktisch gesehen besteht die Blockchain für Patientendaten nur aus zwei Seiten und daher zwei Interessen mit limitierter Teilnehmerzahl: Patienten und Anbieter. Unseriöse Patienten oder Anbieter mit starker Rechenleistung könnten sich zusammenschließen und eine 51% Mehrheit erreichen und gegeben falls Daten nach eigenem Ermessen

manipulieren. Der reine „Proof of Work“ ist somit nicht ausreichend, um die Integrität der Blockchain für Patientendaten zu gewährleisten.

Zusätzlich zu diesem Problem müsste man davon ausgehen, dass eine große Anzahl der Patienten und Patientinnen ständig online zu sein hat, um an der Lösung der Hash-Puzzles teilzunehmen.

Auf Grund dieser Limitationen und Annahmen wird diese Methode nicht weiter verfolgt.

6.2.5.2 Mining Rotation

Bei der Methode „Mining Rotation“ wird, wie in Kapitel 5.10.4 beschrieben, die Validierung auf wenige Stellen limitiert, welche abwechselnd einen Block erzeugen und validieren.

Versucht man dies an vorliegenden Daten für Österreich zu testen, ergibt sich folgende Hochrechnung:

- Ca 46.000 praktizierende Ärzte[St19b]
- Ca 9 Millionen Einwohner[St19a]
- Ca. 1400 Apotheken[Oe19]

Hierbei werden alle Ärzte und Apotheken als „Miner“ berechtigt. Die Patienten selbst erhalten lediglich einen „Thin Client“ als Account.

Das ergibt im Worst Case ca 50.000 aktive „Miner“ und 9 Millionen „Thin Clients“.

Wird nun das Rotationsintervall auf 10sec gesetzt [Bi15, S.12] so ergibt sich nach kurzer Schlussrechnung eine gesamte Durchlaufzeit von knapp 6 Tagen dh etwas weniger als einer Woche. Jeder „Miner“ könnte nur ca einmal pro Woche seine Transaktionen in die Blockchain integrieren und seinen Datenbestand verifizieren lassen. Alle Beteiligten an Transkationen müssten im schlimmsten Fall genauso

eine Arbeitswoche warten, bis die Daten in der Blockchain sichtbar wären (zB Rezepte für Apotheken).

Das Ergebnis stellt keine zufriedenstellende Lösung für die Praxis dar. Ein anderer Ablauf für diesen Fall lässt sich erzielen indem die Knoten unabhängig voneinander ihre Blöcke nach der Methode „First Come, First Serve“ erstellen. Jede neue Transaktion wird von allen „Minern“ verifiziert. Ist letztendlich ein Block eines „Miners“ fertig gestellt, bedarf es nur mehr der Verifizierung des Block-Headers und der neue Block wird zur BC hinzugefügt. Der Nachteil ist wiederum der, dass eine genaue Durchlaufzeit nicht berechnet werden kann. Im Endeffekt könnte eine Transaktion und ein Block abermals mehrere Stunden benötigen, bis diese in der Blockchain abgebildet sind.

Die Lösung liegt nun darin, das „Mining“ auf höherer Ebene anzusiedeln. Folgende Annahme: Blöcke können beispielsweise nur bezirksweit erstellt werden. Pro Bezirk existiert somit nur eine „Mining“-Stelle, eine sogenannte „High-Level“-Mining-Stelle welche Blöcke zur BC hinzufügen kann und darf. Diese Stelle wird beim Hinzufügen von allen anderen Bezirk-„Miners“ verifiziert. Die Transaktionen selbst werden von den Ärzten oder anderen „Low-Level-Minern“ durchgeführt, signiert und an die zuständige „Mining-Stelle“ weitergeleitet. Diese überprüft die Gültigkeit sowie Zuständigkeit und fügt die Transaktion einem neuen Block hinzu. Bei ca. 120 Bezirke (94 Bezirke Österreichs[Fr19] plus 23 Bezirke Wiens[Ru19]) und einem Rotationsintervall von 10sec ergibt sich eine Durchlaufzeit von 20min, bis der Block in die Blockchain integriert wurde. Dieser Zeitraum lässt sich wiederum mit der vorher beschriebenen „Push“ – Methode verbessern, indem Bezirk-„Miner“, welche einen fertigen Block erstellt haben, diesen zur Überprüfung pushen.

Der Vorgang der Bezirk-„Miner“ lässt sich selbstverständlich weiter abstrahieren und auf Bundesebene etablieren. Die Durchlaufzeit könnte hiermit auf 90sec gesenkt werden. Auch eigens eingeführte Zwischenebenen wären denkbar, wobei sich hier Unklarheiten in der Zuständigkeit ergäben.

Für diese Arbeit reicht eine Einteilung in „Bezirks-Minern“ aus. Die Optimierungsfrage wurde in der Einleitung als Nicht-Ziel festgehalten.

6.2.5.3 Blockchain Anchoring

Um nun die bereits erstellten Blöcke vor unbefugter Manipulation weiters zu schützen, wurde in Kapitel 5.10.3 die Methode des Blockchain Anchoring beschrieben. Mit Hilfe dieses Vorgehens werden in einem gewissen Intervall verschiedene Block-Header in eine öffentliche Blockchain eingearbeitet. Bitcoin stellt hierbei nur eine Möglichkeit dar, wird aber in der Arbeit verwendet. Befindet sich einmal der private Block-Header in der öffentlichen Blockchain, so würde eine Manipulation der privaten Blockchain bei einer Kontrolle auffallen. Das optimale Intervall ist wiederum kein Gegenstand dieser Arbeit.

6.2.5.4 Ergebnis: Mining Rotation plus Blockchain Anchoring

Aus den vorangegangenen Kapiteln ist das Prinzip der „Mining-Rotation“ mit Hilfe von signierten Blöcken und dem anschließenden, sporadischem Hinzufügen des Hashwertes in eine öffentliche Blockchain eine valide Möglichkeit, Patientendaten dermaßen zu verarbeiten, um einen Einheitlichen Konsens zu erreichen. Wichtig hierbei ist eine von der Situation abhängigen korrekten Einteilung in High-Level- und Low-Level „Miners“, sowie des Festlegen eines passenden Intervalls in dem Block-Head „veröffentlicht“ werden.

Daraus ergeben sich die nächsten UML Klasse:

Block-Header:
Der Block-Header besteht aus dem Hashwert des Blockes und einem Timestamp, welches Datum und Uhrzeit des Erstellzeitpunktes nachweist.

Block:
Der Block setzt sich aus mehreren Transaktionen, deren Referenzwerte und darüber

hinaus die sich hierbei ergebende Baumstruktur zusammen. Ebenso besitzt jeder Block einen Block-Header.

Bitcoin-Transaktion:
Die Bitcoin-Transaktion besteht aus einem Empfänger, Absender und weiteren Informationen. Wichtig für diese Arbeit ist die Möglichkeit, weiter Informationen in die Bitcoin BC integrieren zu können.

„**Miner**“:
Der „Miner“ erstellt aus den getätigten Transaktionen die Blöcke und schickt diese zur Validierung. Er benötigt somit die Berechtigungen auf Transaktionen und auf die Blockstruktur selbst.

„Validierer“:
Der Validierer validiert, ob es sich beim vom Miner gebildeten Block auch um einen gültigen handelt. Hierzu benötigt er die Berechtigungen, Blöcke einsehen zu können.

„**Anchorer**“:
Der Anchorer erstellt letzten Endes eine Bitcoin-Transaktion. Die Berechtigung hierfür liegt außerhalb der e-Befund BC Struktur, wird aber erwähnt, da es ein essentieller Teil dieser Arbeit ist. Optional bietet es sich an, dass eine öffentliche Liste mit allen publizierten Blöcken und deren Position in der Bitcoin BC geführt wird.

6.2.6 Auswahl der Account-Art für Patientendaten

Bei Patientendaten müssen ein große und vor allem unbekannte Anzahl an Daten (zB: Krankenstände, Medikamente, Verletzungen, Operationen) übertragen werden. Hierbei wäre es unpraktisch bis unmöglich, in jede Transaktion den gesamten „Kontostand“ mit allen Daten, Behandlungen und anderen gesundheitsbezogenen Daten zu schreiben. Die Art der Account-Basierten Blockchain eignet sich demnach besser, um gezielt Informationen zu transferieren.

6.2.7 Auswahl von PKGI/AGI

Bei der Erstellung von Transaktionen werden jeweils die Empfänger- und Absenderadressen für alle Berechtigten festgehalten. Um dies weiter zu anonymisieren eignet sich die Verwendung der „Private Key Generator Info“ und „Address Generator Info“, siehe Kapitel 5.6.4.

Hierbei wird folgenderweise vorgegangen: Jeder Teilnehmer erhält einen PKGI und AGI. Alle GDAs erhalten Zugriff auf alle AGIs. Wird nun eine Transaktion (ein E-Befund) erstellt so können anstelle der eindeutig identifizierbaren Adressen neu generierte AGIs werden. Ein GDA erstellt hierbei zwei AGIs: eine für die Empfänger und eine für die eigene Absenderadresse. Die die AGIs von Drittpersonen nicht auf die ursprüngliche PKGI zuordbar sind, können Unbefugte keinen Schluss über GDA oder Patienten ziehen.

6.2.8 Auswahl der Skript-Sprache für Patientendaten

Die Übermittlung von Patientendaten muss nicht nur nachvollziehbar sondern auch korrekt ablaufen. Logikfehler im Programm, wie in Kapitel 5.6.7 beschrieben, müssen daher auszuschließen sein. Während „Skript“ zu restriktiv aufgebaut wurde, bieten „Smart Contracts“ eine zu große Flexibilität. Die zu verwendende Skript-Sprache muss den Anforderungen für Patientendaten genüge tragen, aber komplexe Abläufe nicht erlauben. Da beide Extrema gezeigt wurden ist ein Mittelweg folglich möglich. In Anlehnung dessen wird eine limitierte Sprache der „Smart Contracts“ angewandt. GDAs dürfen keine neuen Skripts erstellen, wie es in Ethereum möglich wäre, sondern verwenden aus einem vorgefertigten Baukasten den passenden „Smart Contract“.

7. Erstellen des Referenzmodelles

Die Kriterien für das Modell wurden nun festgehalten. Anhand dessen lässt sich der Ablauf der Prozesse zur Erstellung und Speicherung von Transaktionen in einer privaten Blockchain herleiten.

Das Referenz-Modell soll darstellen:

- Welche Accounts und Berechtigungen benötigt werden
- Wie Accounts erstellt werden
- Wie mit Hilfe von PKGI/AGI anonyme e-Befunde erstellt werden können
- Wie erstellte Transaktionen als Block in die Blockchain eingearbeitet werden
- Wie der erstellte Block verifiziert wird
- Wie der Block-Header in eine öffentliche Blockchain (zB Bitcoin) integriert wird.

7.1 Festlegung der Account-Typen

Bisher wurden folgende Account-Typen definiert: Patient, GDA, Miner, Validierer, Anchorer. Jeder Account benötigt eine eigenes PKGI/GI Schlüsselpaar, einen Namen sowie eine UID.

7.2 Festlegen der Berechtigungsstruktur

Wie in den vorangegangenen Kapiteln festgestellt, werden unterschiedliche Berechtigungen im dezentralen Netzwerk benötigt. Je nach Art des Teilnehmers, lassen sich diese herleiten. Ein Patient oder eine Patientin benötigt lediglich eingeschränkte Leserechte, daher darf er oder sie nur Transaktionen lesen, und hierbei nur seine eigenen. Ebenfalls muss er seine Stammdaten bearbeiten können. Wie und ob die Stammdaten verifiziert werden, liegt außerhalb der Blockchain-Struktur und ist deshalb nicht Gegenstand dieses Themas. GDAs benötigen einerseits Zugriffe auf die von ihnen erstellten e-Befunde sowie auf die für sie

relevanten e-Befunde (siehe Kapitel 4.3.5). Des Weiteren müssen sie neue e-Befunde erstellen können. Die „Miner“ dürfen weder e-Befunde lesen noch neue e-Befunde erstellen, sind aber berechtigt, aus den getätigten Transaktionen einen gültigen Block für die Blockchain-Struktur zu erstellen. Ebenso tritt ein „Miner“ in die Doppelrolle des „Validierers“ um die Blöcke von anderen „Miners“ zu validieren oder gegeben falls abzulehnen. Der „Anchorer“ benötigt in der vorhandenen e-Befund BC keine Berechtigung, außer das Lesen des letzten Blocks oder BC-Headers. Er besitzt die Aufgabe, diesen in eine öffentliche Blockchain zu integrieren (zu verankern) und benötigt dementsprechende Ressourcen, in Abhängigkeit von der gewählten Blockchain selbst. (Berechtigungen, Rechenleistung, Kapital). In Tabelle 2 wurde diese Aufschlüsselung zusammengefasst. Während das Lesen der Blocks oder Block-Header für Patienten und GDAs durch die Applikation eingeschränkt werden kann, müssen die Leseberechtigungen implizit in der Blockchain-Struktur anonymisiert werden. Das Lesen eines Blockes ist für Patienten oder GDAs an sich nicht verboten, aber nicht notwendig. Das Lesen von unbefugten Patientendaten hingegen ist nicht erlaubt. Ebenso ist es nicht verboten, dass alle Patienten über die gesamte Blockchain-Struktur verfügen. Dies wurde auf Grund der Performance allerdings ausgeklammert. Jeder Patient und jede Patientin verfügt somit nur über seine oder ihre eigenen e-Befunde.

Tabelle 2 - Berechtigungsvergabe e-Befund Blockchain

	Lesen	**Alle e-Befunde gespeichert**	**e-Befunde erstellen**	**Block erstellen**	**Block validieren**	**Block-Header verankern**
Patient	Eigen e-Befunde	Nein	Nein	Nein	Nein	Nein
GDA	Eigene und benötigte	**Ja**	**Ja**	Nein	Nein	Nein

	e-Befunde					
Miner	Blocks	**Ja**	Nein	**Ja**	Nein	Nein
Validierer	Blocks	**Ja**	Nein	Nein	**Ja**	Nein
Anchorer	Blocks	Nein	Nein	Nein	Nein	**Ja**

Aus dieser Struktur lassen sich auch die Accounts und deren Berechtigungen als UML Klasse finalisieren.

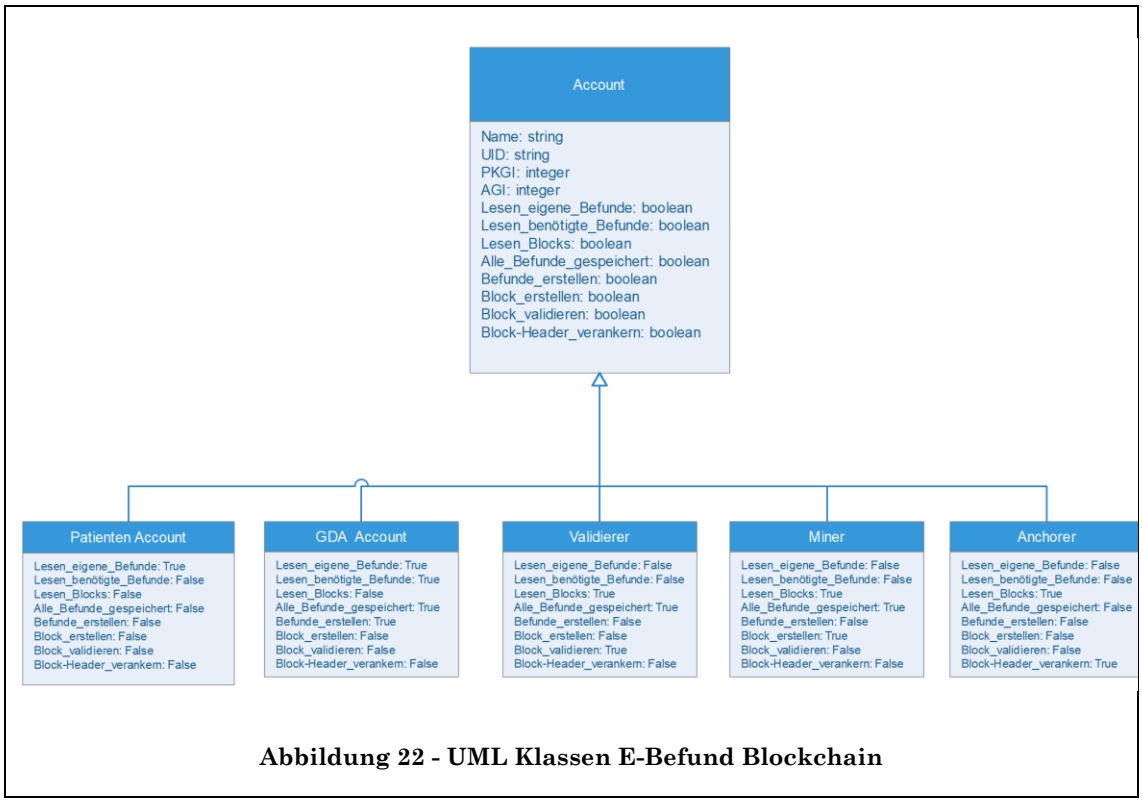

Abbildung 22 - UML Klassen E-Befund Blockchain

Wie in Abbildung 22 dargestellt, wurden die einzelnen Berechtigungen aus Tabelle 2 als „boolean“ hinterlegt. Hierdurch kann jede Berechtigung für jeden Account-Typ

exakt zu definieren. Im späteren Ablauf wird nur mehr auf diese Accounts und deren Berechtigungen verwiesen, ohne diese jedes Mal explizit anführen zu müssen.

7.3 Erstellen von Accounts

Im nächsten Schritt wird definiert, wie Patienten und GDAs ihre Account-Informationen erhalten und was diese genau beinhalten.

Für die Zwecke dieser Arbeit benötigt ein Account einen hinterlegten Namen. Des Weiteren wird jedem Account die PKGI und AGI zugewiesen. Etwaige Zusatzinformationen, wie zugehörige Adresse oder Krankenkasse werden zur Vereinfachung weggelassen, da sie für das Konzept nicht notwendig sind.

GDAs sowie Patienten erhalten ihre Account-Informationen, sobald sie sich registriert haben. Als Registrationsstelle wird in diesem Modell eine Behörde angenommen. Diese übermittelt das PKGI/AGI Paar an den entsprechenden Empfänger. Des Weiteren wird eine Liste aller AGIs der GDAs und eine separat geführte Liste aller Patienten AGIs aktualisiert. GDAs können nun diese Listen einsehen und eine lokale Kopie hiervon erstellen, um immer über die letztgültigen AGIs zu verfügen.

Da die öffentlichen AGIs später benötigt werden, um die E-Befunde zu anonymisieren, muss auch jeder GDA darauf Zugriff haben. Eine behördlich geführte Liste bietet hier eine brauchbare Lösung. In Abbildung 24 wird der Vorgang als UML Aktivitätendiagramm dargestellt. Die Listen sind schematisch in Abbildung 23 als Klassendiagramm abgebildet.

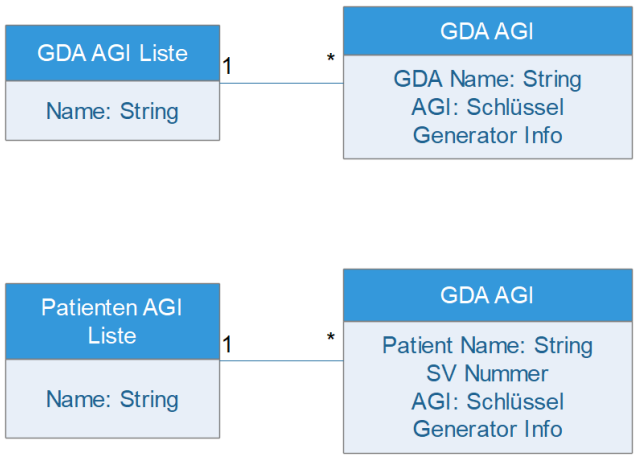

Abbildung 23 - GDA Listen

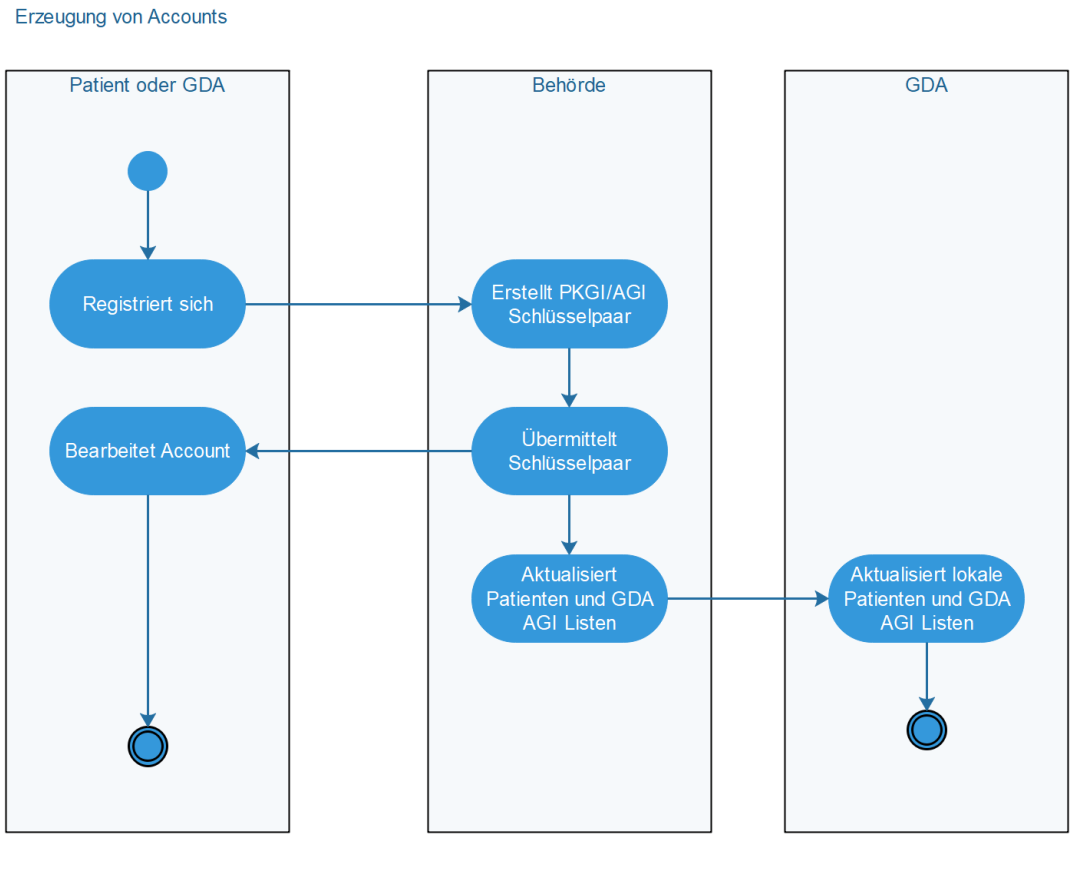

Abbildung 24 - Erzeugung von Accounts

7.4 Erstellen eines e-Befundes

Ein e-Befund kann nun folgendermaßen angelegt werden. Der Arzt erstellt im System einen neuen e-Befund. Hierbei füllt er die entsprechenden Befund-Daten aus. Anschließend wird dieser Datensatz mit der Signatur des Arztes signiert. In einem weiteren Schritt wird nun der signierte Datensatz mit einem zufällig gewählten Pateinten AGIx, wobei x zwischen 1-1000 liegt. verschlüsselt und kann somit nur mit dem entsprechenden PKGIx entschlüsselt werden. Die AGIx wird gleichzeitig als Empfängeradresse festgelegt. Als Absender-Adresse verwendet der GDA die eigene AGIy, wobei y wiederum zufällig zwischen 1-1000 gewählt wird. (Der

Zufallsalgorithmus ist hierbei nicht entscheidend, sondern dass dadurch die Accounts auf nicht nachvollziehbare Weise mit einem e-Befund verknüpft werden.)

Dadurch ergibt sich das in Abbildung 25 dargestellt Klassen-Diagramm. Es ist zu erkennen, dass (auf Grund des Zufallsfaktors) verschiedene e-Befunde an eine der 1000 AGI geknüpft sein können. Dadurch ergeben sich pro Befund 1Mio Möglichkeiten, mit demselben Patienten Account und GDA Account verknüpft zu werden. Ein Außenstehender erkennt hierbei keine Zusammenhänge und kein Schema, wodurch er auf die wahren Identitäten keine Rückschlüsse ziehen kann. Ebenso kann ein Unbefugter die Patientendaten nicht einsehen, da er nicht über den benötigten PKGI besitzt. Die Daten selbst sowie die Identität des Patienten bzw. der Patientin und GDAs bleiben anonymisiert.

Der e-Befund wird nun an den Empfänger „weitergeleitet", daher als Transaktion in der Blockchain mit der entsprechenden AGI publiziert. Benötigen weitere GDAs Zugriff auf diese Daten wird ein temporärer e-Befund erstellt mit entsprechend neuer Verschlüsselung, einem Zeitfenster und einer neuen AGIx des GDAs. Der GDA beantragt Zugriff auf Daten. Dies muss streng genommen nicht via Blockchain passieren. Der entsprechende GDA, welche die Daten besitzt, erstellt ein Paket mit den benötigten Unterlagen, welche als e-Befund zusammengefasst werden. Solange das gesetzt Zeitfenster gilt, gewährt der Patient Zugriff auf die Daten, indem er diese entschlüsselt und an den GDA weiterleitet.

Auch in diesem Prozessfluss eingezeichnet wurden die Miner, welche im Anschluss aus den getätigten Transaktionen eine Block erstellen. Dies wird im nächsten Kapitel erklärt.

In Abbildung 26 wird der Ablauf schematisch dargestellt, Abbildung 27 sowie Abbildung 28 beschreiben die dazugehörigen Unterprozesse.

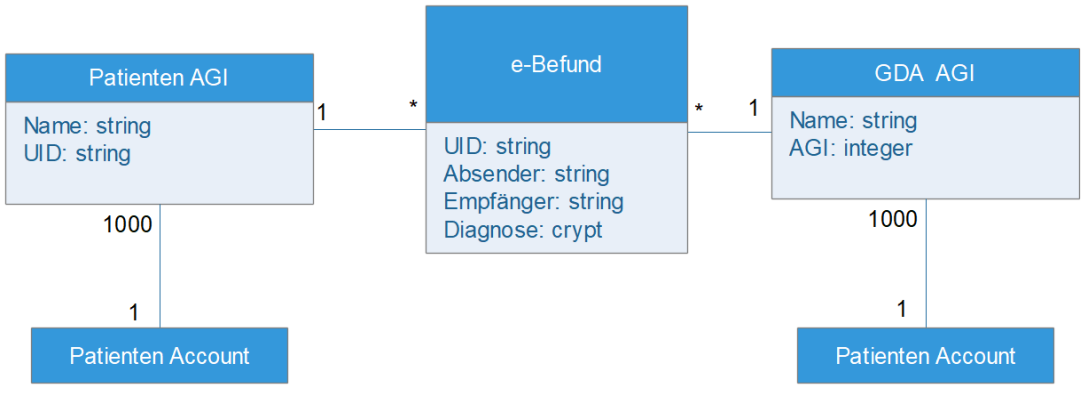

Abbildung 25 - Account und e-Befund

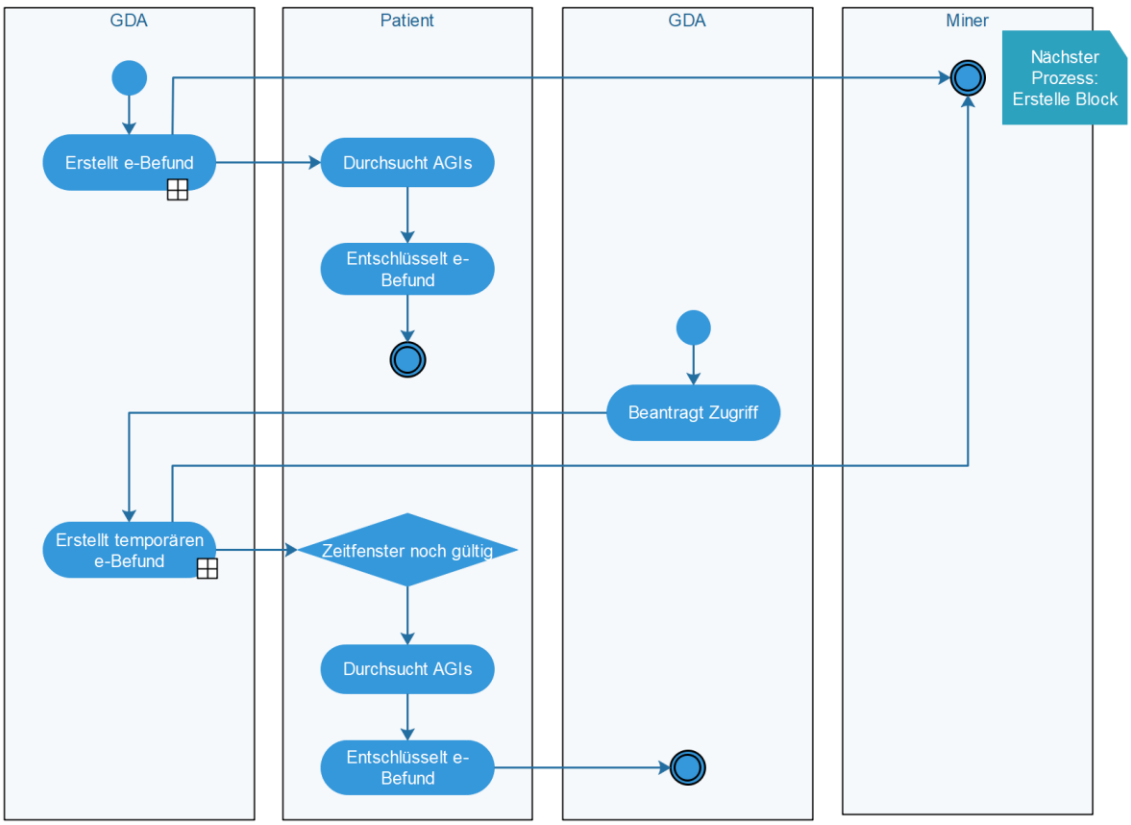

Abbildung 26 – Prozess: Erstellen eines e-Befundes

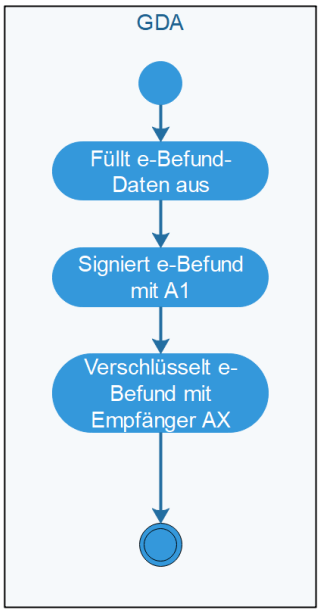

Abbildung 27 - Unterprozess Erstellt e-Befund

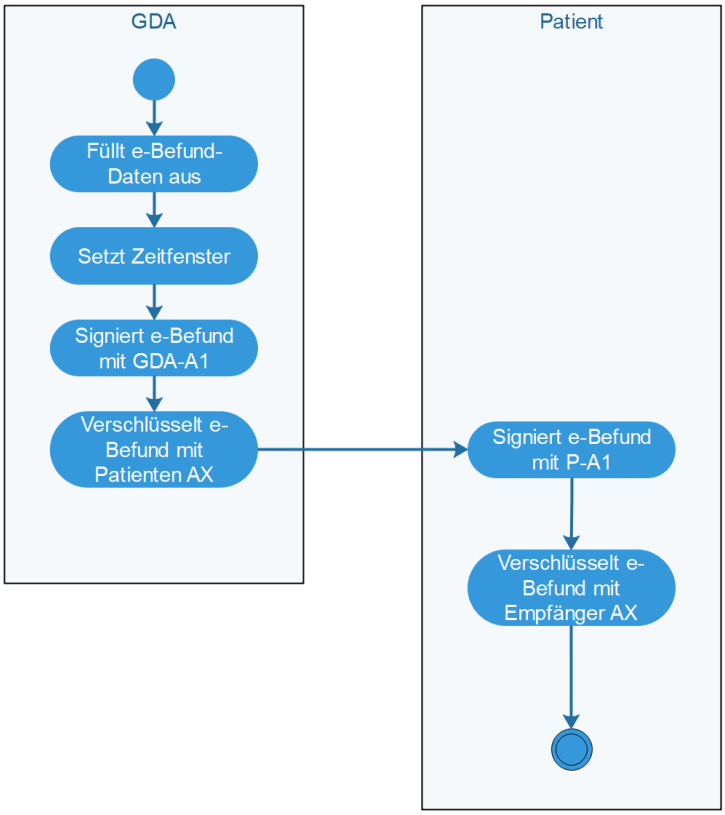

Abbildung 28 - Unterprozess Erstellt temporären e-Befund

7.5 Erstellen eines Blockes der Patienten-Blockchain

Nachdem eine gewisse Anzahl an e-Befunden als Transaktionen getätigt wurden, müssen diese in die Blockchain-Struktur eingefügt werden. Die exakte Zahl muss nicht am Modell festgelegt werden und kann auch je nach Bedarf variieren. Wie in Kapitel 5.8 beschrieben besteht ein Block nicht nur aus Transaktionen sondern ebenso aus deren Hashwerten, welche wiederum gehasht wurden. Der oberste Hashwert des daraus entstandene Merkel Tree bildet den Hashwert des Blockes, genau genommen des Block-Headers, welcher zusätzlich eine Datums-, und

Zeitangabe beinhaltet. Dieser beschriebene Aufbau der Blockchain ist in Abbildung 29 zu sehen.

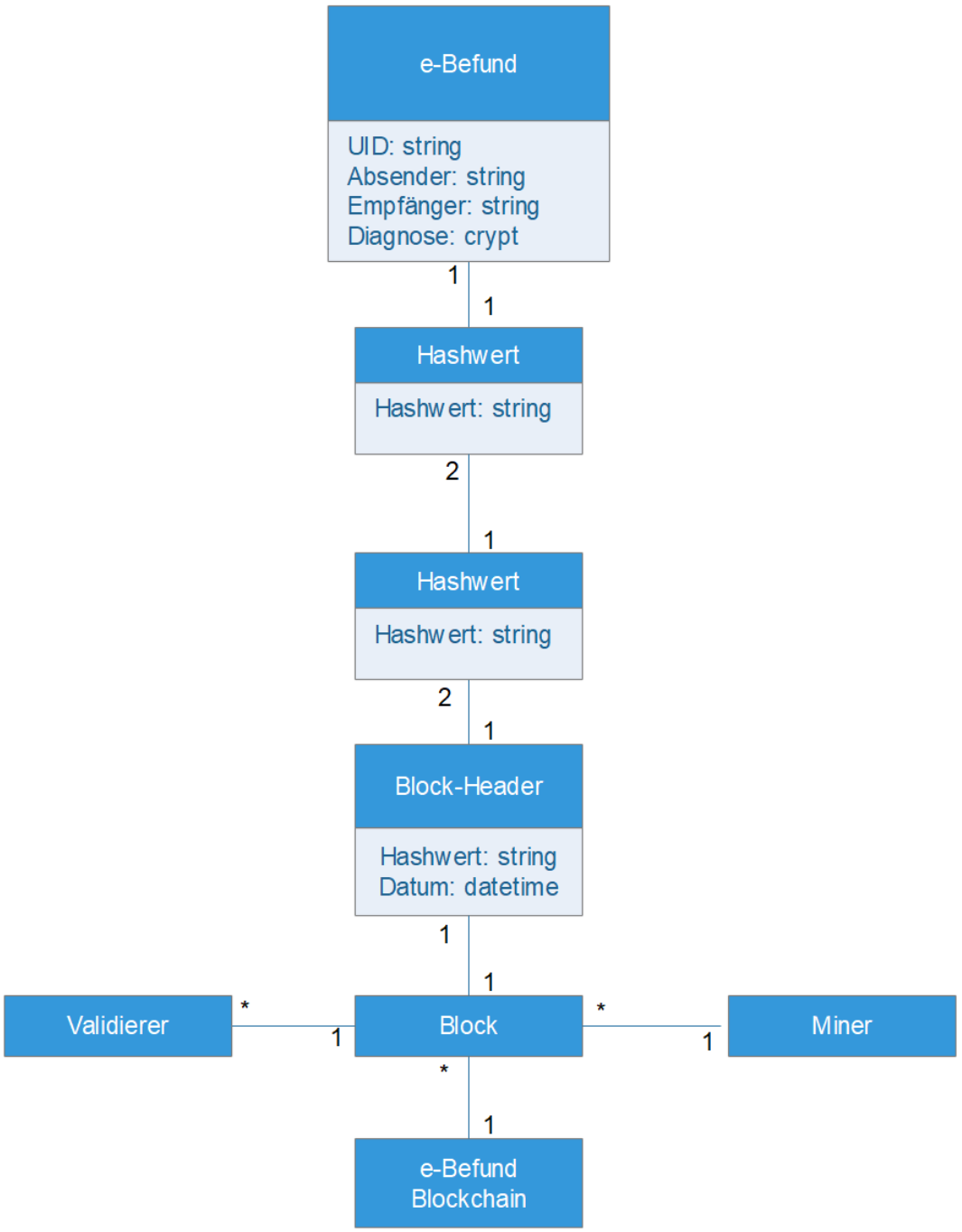

Abbildung 29 - Aufbau der e-Befund Blockchain

Erstellt wird dieser Block und Block-Header von einem „Miner“, validiert von den restlichen Minern, welche als Validierer fungieren, solange sie selbst keinen Block erstellen. (siehe Kapitel 6.2.5.2). Die Validierung erfolgt ebenfalls anhand der Hashwerte, stimme diese nicht überein (siehe Kapitel 5.9) so wird der Block zurück gewiesen und muss vom Miner neu berechnet bzw. erstellt werden. Tritt dies öfters auf (zB zwei Mal pro Monat, zwei Monate in Folge), so kann ein auffälliger Miner diverse Strafen erhalte oder ganz ausgeschlossen werden. Stimmen alle Werte überein, so validieren alle beteiligten den Block mit ihrem privaten Schlüssel. Der Block wird hiermit in die Blockchain-Struktur aufgenommen und der neue Blockheader an den „Anchorer“ übermittelt. Der zugehörige Prozessfluss ist in Abbildung 30 dargestellt.

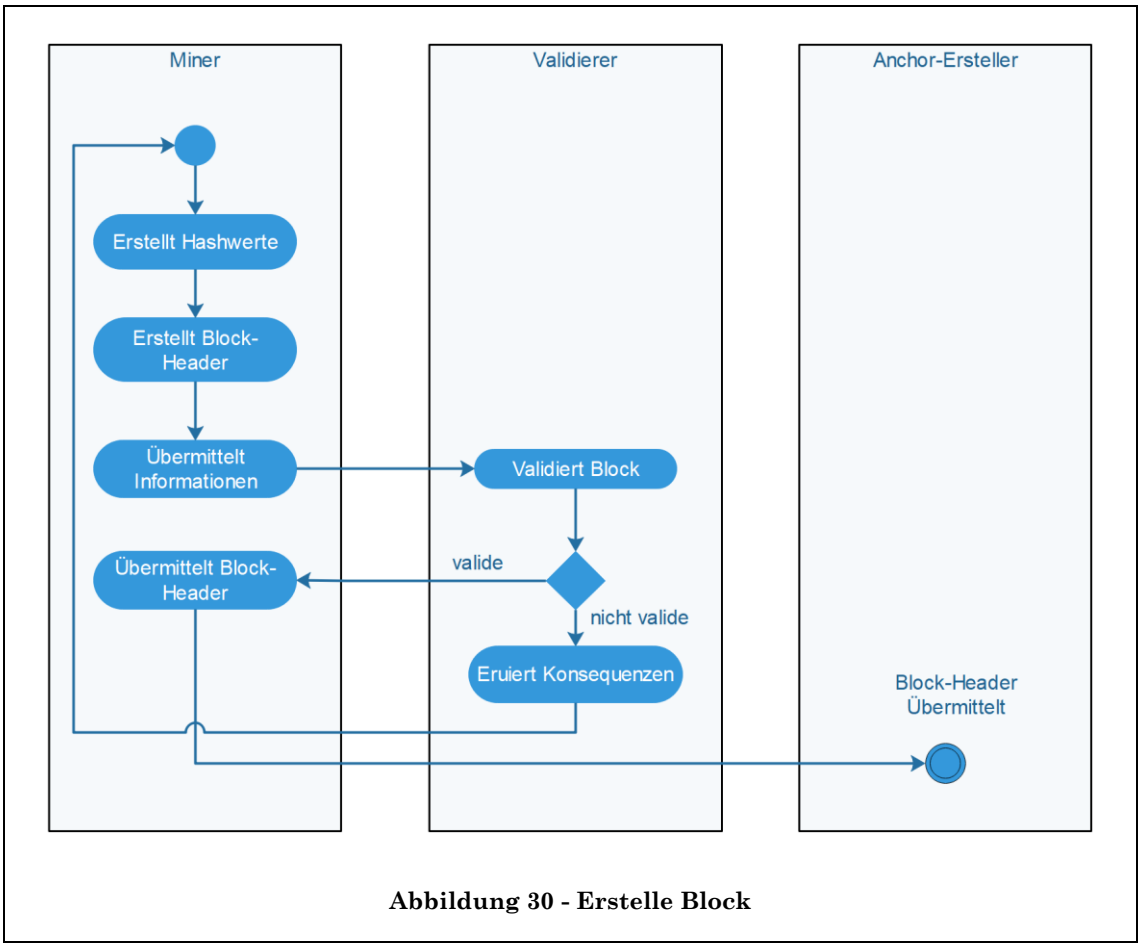

Abbildung 30 - Erstelle Block

7.6 Erstellen einer öffentlichen Blockchain-Transaktion

Als nächstes muss sichergestellt werden, dass im Nachhinein keine Manipulationen an den vorhandenen Daten passieren kann. Hierzu wurde die Methode des „Blockchain Anchorings“ (siehe Kapitel 5.10.3) gewählt.

Hierbei wird in gewissen Abständen der aktuelle Blockheader der e-Befunde-BC in eine öffentliche Blockchain eingebettet. Diese zweite Blockchain kann, muss aber nicht Bitcoin sein.

Durch dieses Einbetten des Headers ist für jeden eine Historie der Block-Headers ersichtlich, ohne irgendetwas über die Daten selbst zu erfahren. Bisher erstellte Blöcke in der e-Befund BC können somit nicht mehr im Nachhinein unbemerkt manipuliert werden. Um Zeit, Geld oder Ressourcen zu sparen, eignet es sich, nicht jeden Block zu verankern. Ein tägliches Verankern eines einzigen Headers um dieselbe Zeit wäre hierbei eine effiziente Alternative. Ist dies der Fall, so eignet sich eine separat geführte Liste über die öffentlich zugänglichen Block-Header zu führen. Die muss in diesem Fall aktualisiert werden.

Ein Beispielhafter Prozessablauf wurde in Abbildung 31 dargestellt, der genaue Ablauf variiert hierbei von der gewählten öffentlichen Blockchain.

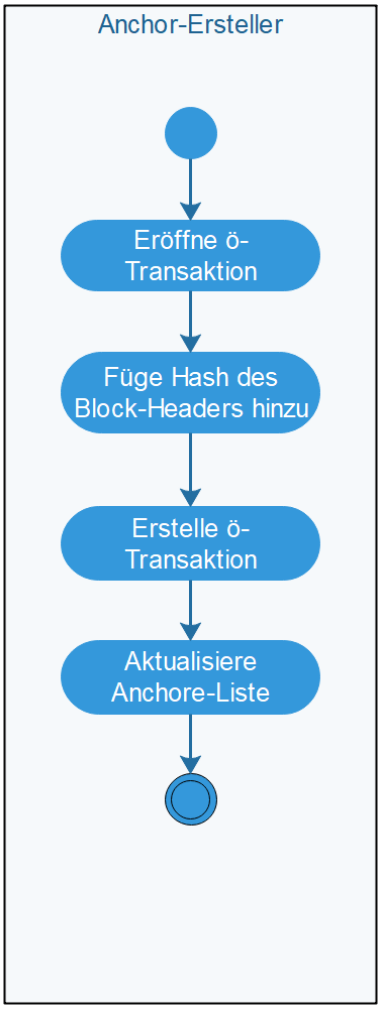

Abbildung 31 - Erstelle öffentliche Transaktion

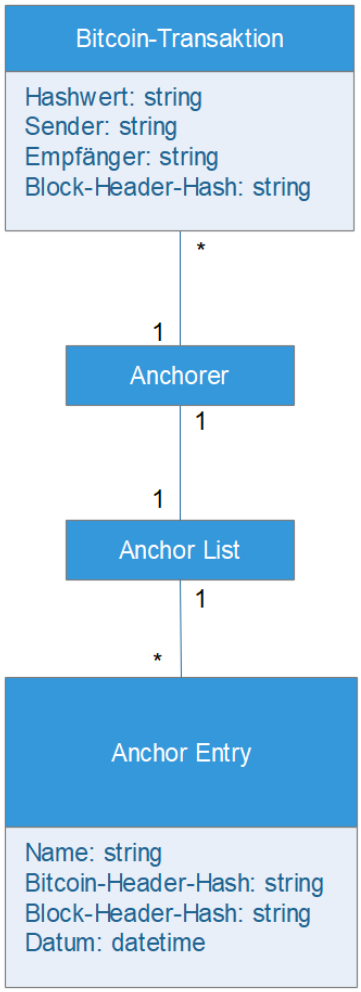

Abbildung 32 - Anchor List

7.7 Zusammenfassung

In den voran gegangenen Kapiteln wurden die einzelnen Prozesse im Detail erläutert. Zusammengefasst bedeutet dies, dass jeder Teilnehmer einen Account bei der Behörde anfordert, um seine Zugangsdaten zu erhalten. GDAs erhalten darüber

hinaus Zugriff auf die AGI Listen. Die Berechtigungen werden von der Behörde direkt vergeben. (Der genaue Ablauf wurde von der Arbeit wiederum ausgegrenzt). Wird nun ein neuer e-Befund jeglicher Art benötigt, erstellt der zuständige GDA diesen und leitet ihn an den entsprechenden Empfänger (sei es nun ein Patient bzw. eine Patientin oder ein GDA) signiert und verschlüsselt von einer zufällig gewählten an eine zufällig gewählte AGI weiter. Der Empfängerknoten kann den e-Befund mit seinem privaten Schlüssel entschlüsseln und einsehen. Wurden genug Transaktionen getätigt, werden diese von dem momentan zuständigen Miner zu einem Block „zusammengefasst". Im Anschluss findet die Validierung statt. Fand diese erfolgreich statt, verankert nun die Rolle des „Anchorers" den Block in einer öffentlichen Blockchain und aktualisiert eine Liste über alle veröffentlichten Block-Header. Die Rolle des Miners wechselt sich in einer vorgegebenen Rotation ab.

8. Überprüfen der Anforderungen

Nachdem nun das Referenzmodell erstellt wurde, muss dieses auf ihre Anwendbarkeit überprüft werden. Wie in Kapitel 3.1.3 beschrieben, soll das Modell an ein konkretes Beispiel angewendet werden. In dieser Arbeit wurden die rechtlichen Anforderungen laut DSGVO sowie die Anforderungen an ELGA festgehalten. Aus diesem Grunde werden beide Anforderungen dem Referenzmodell gegenüber gestellt und evaluiert. Hierdurch ergibt sich eine Kombination aus Evaluierung durch Literatur und Anwendungsfall.

Die Anforderungen lauteten konkret:

- A1: Patientendaten dürfen nur dem Zwecke entsprechend verwendet, und daher nicht an Dritte/Unbeteiligte weitergegeben werden.
- A2: Dritte/Unbefugte dürfen daher keinen Zugriff auf diese Daten besitzen
- A3: Patientendaten müssen dem letzten Stand entsprechen und daher von den Beteiligten aktualisiert werden können, wobei der Betroffene die Aktualisierung beim Verantwortlichen anfordert und dieser die Aktualisierung nachvollziehbar durchführt.
- A4: Getätigte Daten bezüglich Behandlungen dürfen nicht geändert werden. Hier ist ein zusätzlicher, aktueller Datensatz anzulegen.
- A5: Die aktuellen Daten und jeglicher Verarbeitungsvorgang muss jederzeit von den betroffenen einsehbar sein.
- A6: Der Zugriff auf die Daten für Dritte ist zeitlich beschränkt.

Die abgeleiteten Prozesse aus dem vorherigen Kapitel lauteten:

- P1: Erstelle Account
- P2: e-Befund
- UP21: Erstelle e-Befund
- UP2.2: Erstelle temporären e-Befund
- P3: Erstelle Block

- P4: Erstelle öffentliche Transaktion

Folglich werden alle Prozesse mit allen Anforderungen kombiniert und eine Entscheidung getroffen, ob die Anforderung erfüllt wurde (JA), nicht zu trifft (n/a) oder nicht erfüllt wurde (NEIN).

8.1 Validierung im Detail

P1-A1: Diese Anforderung ist erfüllt, die Zugangsdaten und das PKGI/AGI Schlüsselpaar wird nur an den registrierten Teilnehmer weiter geleitet. Der AGI wird veröffentlicht, bietet aber keine Rückschlüsse über die Person bzw. dessen Account-Informationen oder Transaktionshistorie selbst. (JA)

P1-A2: Dritte/Unbefugte besitzen keinen Zugriff auf den PKGI, der AGI ist öffentlich zugänglich, analog zu jedem Schlüsselpaar. (JA)

P1-A3: Patienten können ihre Daten jederzeit aktualisieren. Der Verifizierungsprozess auf Gültigkeit liegt außerhalb der Blockchain-Struktur selbst. (JA)

P1-A4: Trifft nicht zu, da der Prozess nicht auf e-Befunde zugreift. (n/a)

P2-A5: Trifft nicht zu, da der Prozess nicht auf e-Befunde zugreift. (n/a)

P2-A1: Bei sachgemäßer Handhabung stehen die Patientendaten ausschließlich den Personen, welche darauf Zugriff benötigen, zur Verfügung. (JA)

P2-A2: Die Patientendaten können nur von den jeweiligen Personen eingesehen werden. Für alle anderen sind die Daten verschlüsselt und somit nicht einsehbar. (JA)

P2-A3: Trifft nicht zu, da hier keine Daten aktualisiert werden. (n/a)

P2-A4: Erstellte Transaktionen können in der Blockchain-Struktur nicht mehr geändert werden, ohne dass die zugehörigen Hashreferenzen ungültig werde. Man

müsste alle Referenzen neu berechnen. In Kombination mit P3 ist dies nicht möglich. (JA)

P2-A5: Getätigte Transaktionen können jederzeit von den Betroffenen (Patienten, GDA) eingesehen werden. (Behandelnde GDAs behalten somit jederzeit Überblick über ihre e-Befunde, erhalten aber keine weiteren Informationen, sobald die Behandlung zu Ende ist.) (JA)

UP2.1-A1: Bei sachgemäßer Handhabung stehen die Patientendaten ausschließlich den Personen, welche darauf Zugriff benötigen, zur Verfügung. (JA)

UP2.1-A2: Die Patientendaten können nur von den jeweiligen Personen eingesehen werden. Für alle anderen sind die Daten verschlüsselt und somit nicht einsehbar. (JA)

UP2.1-A3: Trifft nicht zu, da hier keine Daten aktualisiert werden. (n/a)

UP2.1-A4: Trifft nicht zu, da hier nur die Daten für den e-Befund erstellt werden. (n/a)

UP2.1-A5: Trifft nicht zu, da hier nur die Daten für den e-Befund erstellt werden. (n/a)

UP2.1-A1: Bei sachgemäßer Handhabung stehen die Patientendaten analog zu UP2.2 ausschließlich den Personen, welche darauf Zugriff benötigen, zur Verfügung. (JA)

UP2.1-A2: Die Patientendaten können ebenfalls nur von den jeweiligen Personen eingesehen werden. Für alle anderen sind die Daten verschlüsselt und somit nicht einsehbar. (JA)

UP2.1-A3: Trifft nicht zu, da hier keine Daten aktualisiert werden. (n/a)

UP2.1-A4: Trifft nicht zu, da hier nur die Daten für den e-Befund erstellt werden. (n/a)

UP2.1-A5: Trifft nicht zu, da hier nur die Daten für den e-Befund erstellt werden. (n/a)

P3-A1: Die e-Befunde werden zum Zwecke der Weiterverarbeitung von den Minern eingesehen. Diese erstellen hieraus einen gültigen Block. (JA)

P3-A2: Die Miner können den Inhalt des e-Befundes nicht einsehen, da sie nicht über den richtigen Schlüssel verfügen. Ebenso können sie Empfänger und Absender nicht identifizieren. (JA)

P3-A3: Trifft nicht zu, da hier keine Daten aktualisiert werden. (n/a)

P3-A4: Durch das Erstellen von Blöcken werden die Transaktionen mit Hashreferenzen miteinander in Bezug gebracht. Ändert sich eine Transaktion, so ändern sich auch alle betroffenen Referenzen. (JA)

P3-A5: Trifft nicht zu, da hierbei keine Patientendaten verarbeitet werden, (n/a)

P4-A1: Trifft nicht zu da hierbei keine Patientendaten verarbeitet werden, (n/a)

P4-A2: Trifft zu, da der „Anchorer“ keinen Zugriff auf die Patientendaten besitzt (JA)

P4-A3: Trifft nicht zu, da hierbei keine Patientendaten aktualisiert werden (JA)

P4-A4: Trifft nicht zu, da hierbei keine Patientendaten aktualisiert werden (JA)

P4-A5: Der letztgültige Bock-Header wird in einer öffentlichen Blockchain eingefügt und kann somit von jedem eingesehen werden. Ein späteres Ändern der e-Befund

Blockchain ist unbemerkt nicht möglich. Die Patientendaten sind nicht einsehbar, sehr wohl aber die Validität der gesamten Blockchain-Struktur. (JA)

In der nachfolgenden Matrix werden die Ergebnisse dieser Evaluierung dargestellt.

Tabelle 3 - Validierung der Anforderungen anhand der Prozesse

	A1	A2	A3	A4	A5	A6
P1	Ja	Ja	Ja	n/a	n/a	n/a
P2	Ja	Ja	n/a	Ja	Ja	Ja
UP2.1	Ja	Ja	n/a	n/a	n/a	Ja
UP2.2	Ja	Ja	n/a	n/a	n/a	Ja
P3	Ja	Ja	n/a	Ja	n/a	n/a
P4	n/a	Ja	n/a	n/a	Ja	n/a

8.2 Ergebnis der Überprüfung

Im vorhergegangenen Kapitel wurden alle Anforderungen den Prozessen gegenübergestellt. Hierbei konnte jede zutreffende Anforderung erfüllt werden.

Die Anforderungen leiteten sich aus den gesetzlichen Rahmenbedingungen für Patientendaten sowie der Elektronischen Gesundheitsakte ab.

Auf Grund dessen lässt sich schlussfolgern, dass sich das festgelegte Referenzmodell eignet, eine Blockchain-Struktur zur Verarbeitung von Patientendaten, im Speziellen der ELGA-Patientendaten, zu definieren.

9. Demonstration anhand von Medicalchain

Um nun das bereits erstellte Referenzmodell auf seine weitere Verwendbarkeit zu validieren, muss dieses mit Hilfe eines konkreten Anwendungsfalls demonstriert (und somit validiert) werden. (Siehe Kapitel 3.1.3) Hierbei wurde die in der Einleitung beschriebene Lösung „Medicalchain“ heran gezogen.

9.1 Aufbau

Bei „Medicalchain“ handelt es sich um eine Kombination von zwei verschiedenen Blockchains. Der Aufbau und die Funktionsweise werden im dazugehörigen Whitepaper auf https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf folgender Maßen definiert: [Me18b]

BC1 dient zur Verwaltung der Patientendaten, BC2 als Transaktionshistorie. In der Transaktionshistorie wird die Referenz der zugehörigen Patientendaten in der zweiten Blockchain eingebettet.
Weiters werden die Transaktionen bei BC2 mit Hilfe von Smart Contracts abgeschlossen.[Me18b, S.12]

9.2 Berechtigungen

Eine exemplarische Verteilung der Berechtigungsstruktur von Medicalchain wird, wie folgt, angeführt:[Me18b]

Practitioner: Diese dürfen Datensätze erstellen und editieren, wenn sie die Berechtigung hierfür besitzen. Ebenso dürfen Sie Berechtigungen für anderen Practitioners anfordern. Die Rolle kann analog zu GDAs betrachtet werden. [Me18b, S.13]

Patient: Patienten dürfen ihre Daten einsehen, Practitioners auf ihre Datensätze bzw. einen Teil ihrer Datensätze zugreifen oder editieren lassen. Ebenso besitzen sie die Möglichkeit, Berechtigungen wieder zu annullieren, sowie bestimmte Attribute

in ihren Patientendaten zu editieren. Ebenso existiert ein „Emergency“ Kontakt, welcher bei einem Notfall auf die Daten des Patienten zugreifen und Berechtigungen weiter vergeben darf. [Me18b, S.13] Für das Referenzmodell werden sie als GDA-Write bezeichnet.

Research Institution: Dürfen lediglich zur Verfügung gestellte Datensätze einsehen. Diese Rolle kann als GDA, welcher keine Transaktionen erzeugt, betrachtet werden. [Me18b] Die entsprechende Rolle für das Modell heißt dementsprechend GDA-Read.

Zu beachten ist, dass eine GDA-Stelle je nach Anwendungsfall die Rolle eines GDA-Write oder eines GDA-Read einnehmen kann.

In Tabelle 4 ist die Aktualisierung der Rollen dargestellt.

Tabelle 4 - Berechtigungen für "Medicalchain"

	Lesen	**Alle e-Befunde gespeichert**	**e-Befunde erstellen**	**Block erstellen**	**Block validieren**	**Block-Header verankern**
Patient	Eigen e-Befunde	Nein	Nein	Nein	Nein	Nein
GDA-Write	Eigene und benötigte e-Befunde	**Ja**	**Ja**	Nein	Nein	Nein
GDA-Read	Eigene und	**Ja**	Nein	Nein	Nein	Nein

	benötigte e-Befunde					
Miner	Blocks	**Ja**	Nein	**Ja**	Nein	Nein
Validierer	Blocks	**Ja**	Nein	Nein	**Ja**	Nein
Anchorer	Blocks	Nein	Nein	Nein	Nein	**Ja**

9.3 Ablauf

Ein Practitioner erstellt einen Befund. Die Patientendaten selbst werden symmetrisch verschlüsselt und in der BC1 abgelegt. In der BC2 wird der symmetrische Schlüssel mit dem PK des Empfänger verschlüsselt und als Transaktion in BC2 abgelegt. Der Empfänger kann nun mit seinem SK den symmetrischen Schlüsseln deschiffrieren und die Daten in BC1 einsehen.[Me18b, S.14, 15]

Soll der Zugriff wieder eingeschränkt werden, so passiert dies, indem auf BC1 das ursprüngliche Datenpaket mit einem neuen symmetrischen Schlüssel chiffriert wird. Der alte kann somit das Datenpaket nicht mehr öffnen und der ursprüngliche Empfänger besitzt keinen Zugriff mehr.[Me18b, S.14, 15]

9.4 Anforderungen

Aus den Kapiteln 9.1, 9.2 und 9.3 und dem Whitepaper lassen sich nun die benötigten Anforderungen zur Demonstration herleiten.

- **A1:** Accounts werden direkt von „Medicalchain SA“ via MedToken erzeugt und an die Benutzer verteilt.[Me18b, S.33] Die User selbst werden von Civic [Ci19] auf Echtheit überprüft und verifiziert. [Me18b, S.12]
- **A2:** Patientendaten sind verschlüsselt und können nur von Patienten und berechtigten „Practitioners“ eingesehen werden.[Me18b, S.14, 15]
- **A3:** Der Zugriff auf Patientendaten kann von den jeweiligen Patienten jederzeit und direkt vergeben oder blockiert werden.[Me18b, S.14, 15]
- **A4:** Patienten können ihre Stammdaten direkt editieren
- **A5:** Zum Erstellung und Validieren von Blöcken wird Ethereum und folglich „Proof of Stake“ verwendet.
- **A6:** Bei der Skriptsprache handelt es sich um „Smart Contracts“

9.5 Überprüfen der Anforderungen

Als nächstes müssen die Anforderungen überprüft und gegeben falls Änderungen am bereits erstellten Referenzmodell erstellt werden.

A1: Die Accounts werden direkt von „Medicalchain SA“ via MedToken zur Verfügung gestellt.[Me18b, S.37] Um dies auf das vorliegende Referenzmodell auszuweiten, muss die Rolle der „Behörde“ abstrahiert werden auf eine Vertrauensstelle. Je nach Anwendungsfall kann dies weiterhin eine Behörde oder die zugehörige Firma, wie Medicalchain sein. Eine Zertifizierungsstelle ist ebenfalls denkbar. Abbildung 33 reflektiert diese Änderung.

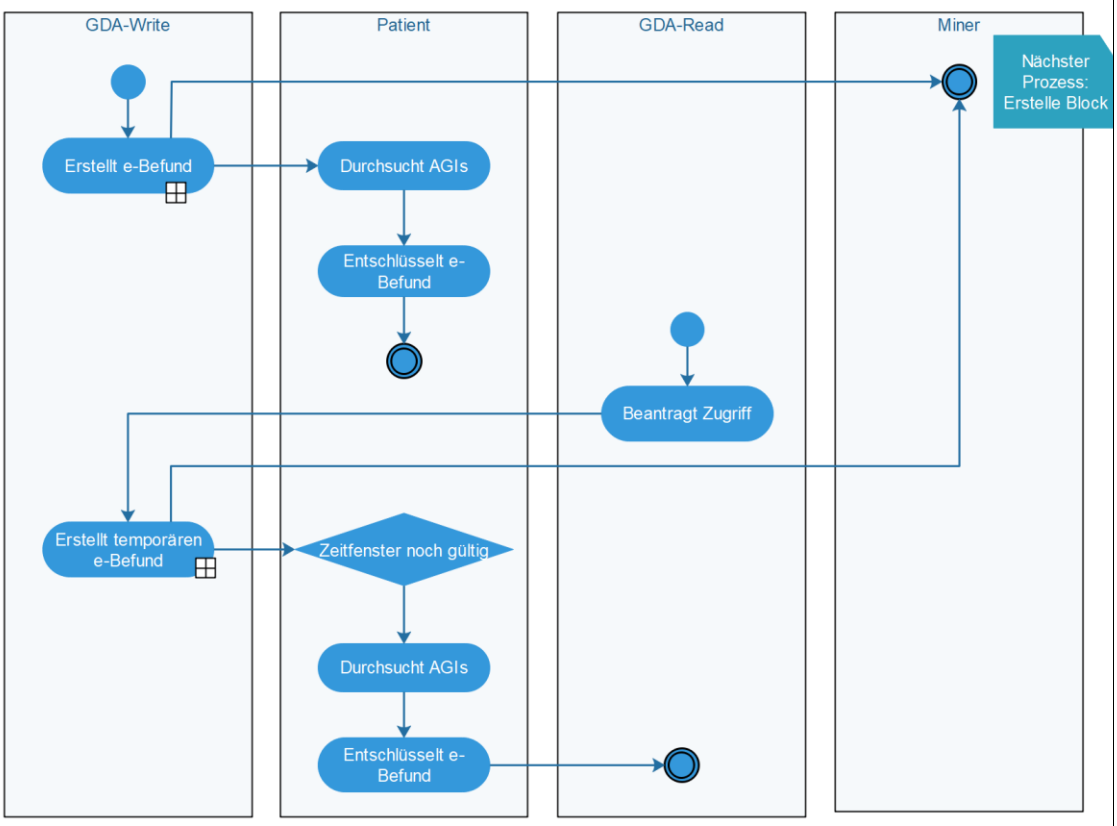

Abbildung 33 - Korrektur: Erzeugung von Accounts

A2 und A3: Medicalchain verwendet zwei Blockchains. Es soll gezeigt werden, dass dies nicht notwendig ist und das in Kapitel 7 erstellte Referenzmodell ausreicht, die Patientendaten zu verarbeiten. Analog zu ELGA dürfen die erstellten Daten (nachfolgend ebenfalls als e-Befunde bezeichnet) nur von den dafür berechtigten „Practitioner“ (GDA-Read und GDA-Write) eingesehen werden. Der Patient oder die Patientin definiert hierbei, ob der Zugriff zu einzelnen Daten gewährleistet wird, oder nicht. Die Daten werden symmetrisch verschlüsselt. Dieser Schlüssel wird nach Beendigung der Berechtigung geändert und der e-Befund mit einem neuen verschlüsselt. Im erstellten Referenzmodell werden Daten direkt mit dem PK des

Patienten-Accounts und des Empfänger-Accounts verschlüsselt und das Enddatum des erlaubten Zugriffes definiert.

Dies kann mit dem vorhandenen Modell nicht mehr aktiv blockiert werden. Hierzu ist eine weitere Transaktion notwendig, welche ein neues Enddatum in der Vergangenheit setzt und auf die Transaktion mit dem e-Befund verweist. Das System erkennt nun, dass das Enddatum bereits abgelaufen ist und verweigert den Zugriff. Da es sich bei dieser Methode technisch gesehen nur um die Veränderung eines Datums handelt, kann hier jeder beliebiger Zeitpunkt gesetzt werden. Diese Methode eignet sich somit ebenfalls, um bestehende Berechtigungen zu verlängern oder zu verkürzen. Da sowohl GDAs als auch Patienten die Berechtigungen verändern können wurden hierfür zwei Prozess erstellt. Abbildung 34 stellt den Ablauf des Patienten dar, Abbildung 35 jenen des GDAs.

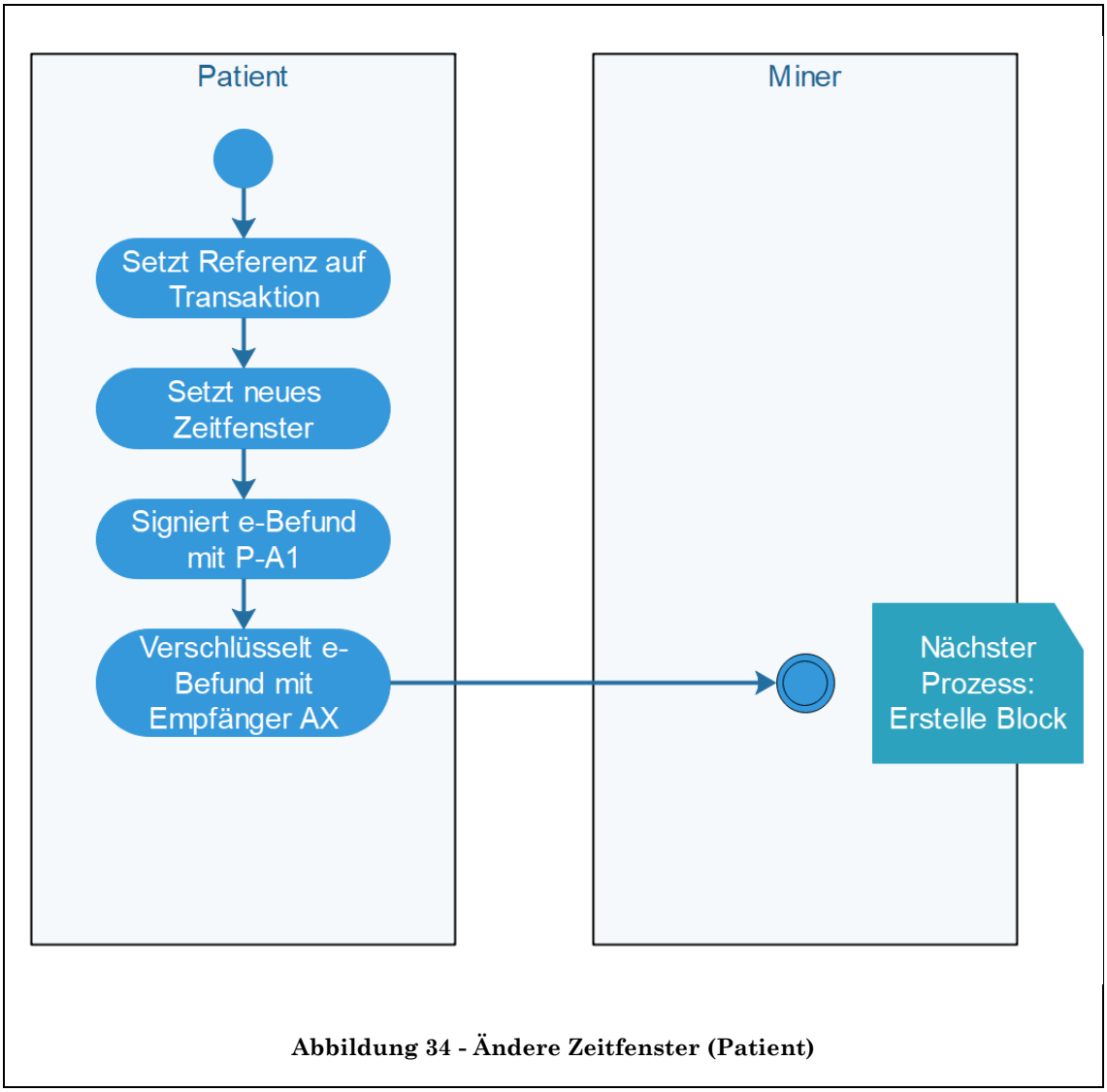

Abbildung 34 - Ändere Zeitfenster (Patient)

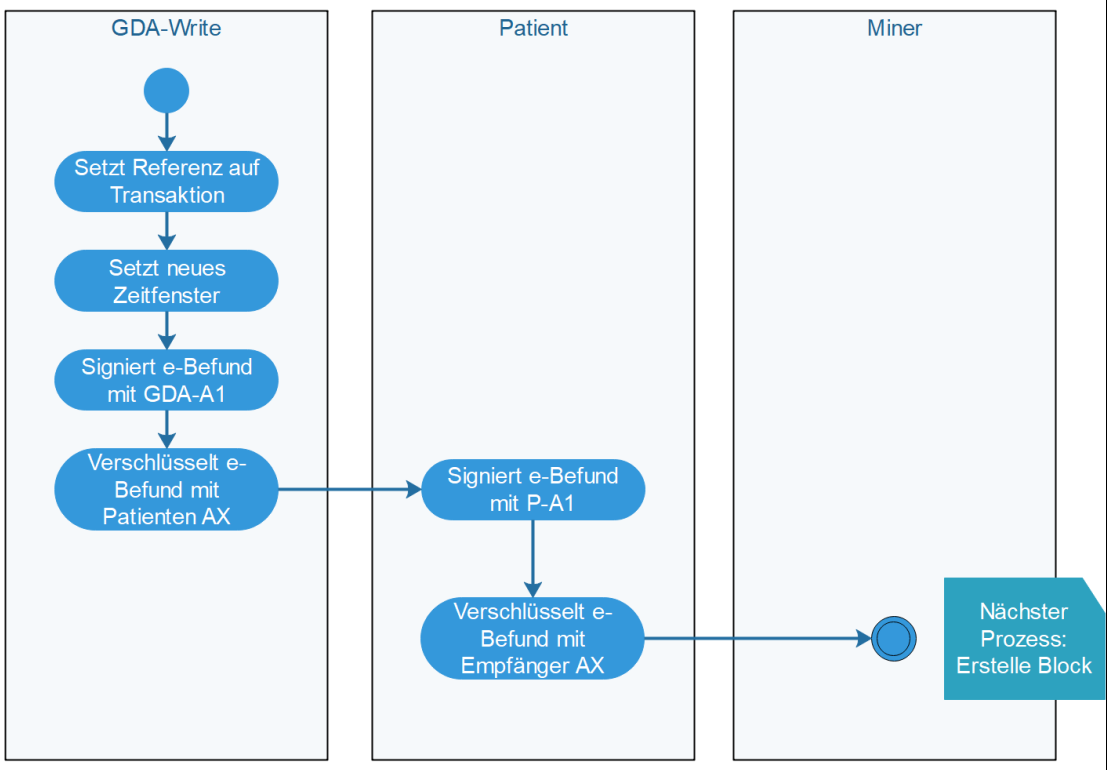

Abbildung 35 - Ändere Zeitfenster (GDA)

A4: Des Weiteren müssen Patienten in der Lage sein, ihre Stammdaten zu editieren. Diese Daten werden in der vorgeschlagenen Lösung des Referenzmodelles im Account selbst gespeichert und können von den Benutzern jederzeit aktualisiert werden. Für alle weitere Aktualisierungen wird eine neue Transaktion erstellt.

A5: Die Erstellung und Validierung von Blöcken erfolgt mit Hilfe des „Proof of Stake“ (Kapitel 5.10.5). Obgleich im vorliegenden Referenzmodell explizit die Mining Rotation gewählt wurde, besteht durchaus die Möglichkeit, andere Konsens-Protokolle zu verwenden, wenn dies explizit verlangt wird. Die Gründe **gegen** die Verwendung wurden in Kapitel 5.10.5 bereits genannt. Technisch kann das

Referenzmodell dies aber unterstützen. Der Prozessfluss ist abstrakt genug, wie Abbildung 30 zeigt.

A6: Als Skriptsprache wurden die „Smart Contracts“ gewählt. Analog zu A5 wurden die Gründe **gegen** die Verwendung dieser in Kapitel 6.2.8 genannt. Technisch stellt es aber kein Hindernis dar, andere Skriptarten, wie „Smart Contracts“ zu implementieren, wenn dies gefordert ist.

Die Rolle des „Anchorer“ findet bei der Lösung von Medicalchain keine Anwendung. Da es sich um eine Erweiterung zur Absicherung des Konsens handelt, wäre es auch hier sinnvoll, ein Blockchain-Anchoring zu implementieren.

9.6 Validierung im Detail

Wie in Kapitel 9.6 werden nun die Anforderungen aus Kapitel 9.4 übersichtlich den einzelnen Prozessen gegenüber gestellt.

Die Anforderungen sind zur einfachen Übersicht noch einmal aufgelistet:

- **A1:** Accounts werden direkt von „Medicalchain SA“ via MedToken erzeugt und an die Benutzer verteilt.[Me18b, S.33] Die User selbst werden von Civic [Ci19] auf Echtheit überprüft und verifiziert. [Me18b, S.12]
- **A2:** Patientendaten sind verschlüsselt und können nur von Patienten und berechtigten „Practitioners“ eingesehen werden.[Me18b, S.14, 15]
- **A3:** Der Zugriff auf Patientendaten kann von den jeweiligen Patienten jederzeit und direkt vergeben oder blockiert werden.[Me18b, S.14, 15]
- **A4:** Patienten können ihre Stammdaten direkt editieren
- **A5:** Zum Erstellung und Validieren von Blöcken wird Ethereum und folglich „Proof of Stake“ verwendet.
- **A6:** Bei der Skriptsprache handelt es sich um „Smart Contracts“

Ebenso wurden die Prozesse noch einmal zusammengefasst:

- P1: Erstelle Account
- P2: e-Befund
- UP2.1: Erstelle e-Befund
- UP2.2: Erstelle temporären e-Befund
- P3: Erstelle Block
- P4: Erstelle öffentliche Transaktion
- P5: Ändere Zeitfenster (GDA-Write)
- P6: Ändere Zeitfenster (Patient)

P1-A1: Der User beantragt den Account, eine „Vertrauensstelle“ verifiziert den Account und übermittelt die Daten. (Ja)

P1-A2: Dritte oder Unbefugte besitzen keinen Zugriff auf den PKGI. Der AGI ist öffentlich zugänglich, analog zu jedem Schlüsselpaar. (Ja)

P1-A3: Nicht zutreffend, da dies in P5 und P6 geregelt wird (n/a)

P1-A4: Die Aktualisierung der Stammdaten ist für Patienten jederzeit möglich. (Ja)

P1-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

P1-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

P2-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P2-A2: Nicht zutreffend, da dies in P1, UP2.1, und UP2.2 definiert und validiert wurde (n/a)

P2-A3: Nicht zutreffend, da dies in P5 und P6 geregelt wird (n/a)

P2-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P2-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

P2-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

UP2.1-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

UP2.1-A2: Die e-Befunde werden in der Blockchain-Struktur verschlüsselt gespeichert (PK des Patienten und des Empfängers) und können nur von diesen Personen eingesehen werden. (Ja)

UP2.1-A3: Nicht zutreffend, da dies in P5 und P6 definiert und validiert wurde (n/a)

UP2.1-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

UP2.1-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

UP2.1-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

UP2.2-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

UP2.2-A2: Die e-Befunde werden in der Blockchain-Struktur verschlüsselt gespeichert (PK des Patienten und des Empfängers) und können nur von diesen Personen eingesehen werden. (Ja)

UP2.2-A3: Nicht zutreffend, da dies in P5 und P6 definiert und validiert wurde (n/a)

UP2.2-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

UP2.2-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

UP2.2-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

P3-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P3-A2: Nicht zutreffend, da dies in P1, UP2.1, und UP2.2 definiert und validiert wurde (n/a)

P3-A3: Nicht zutreffend, da dies in P5 und P6 geregelt wird (n/a)

P3-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P3-A5: Das Referenzmodell unterstützt bei Bedarf auch andere Konsens-Protokolle. (Ja)

P3-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

P4-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P4-A2: Nicht zutreffend, da dies in P1, UP2.1, und UP2.2 definiert und validiert wurde (n/a)

P4-A3: Nicht zutreffend, da dies in P5 und P6 geregelt wird (n/a)

P4-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P4-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

P4-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

P5-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P5-A2: Nicht zutreffend, da dies in P1, UP2.1, und UP2.2 definiert und validiert wurde (n/a)

P5-A3: Die Rolle GDA-Write hat die Berechtigung, ein neues Zeitfenster zu setzen. Es wird eine Transaktion mit Referenz zur den ursprünglichen Daten und dem neuen Enddatum gesetzt. (Ja)

P5-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P5-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

P5-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

P6-A1: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P6-A2: Nicht zutreffend, da dies in P1, UP2.1, und UP2.2 definiert und validiert wurde (n/a)

P6-A3: Die Rolle Patient hat die Berechtigung, ein neues Zeitfenster zu setzen. Es wird eine Transaktion mit Referenz zur den ursprünglichen Daten und dem neuen Enddatum gesetzt. (Ja)

P6-A4: Nicht zutreffend, da dies in P1 definiert und validiert wurde (n/a)

P5-A5: Nicht zutreffend, da dies in P3 definiert und validiert wurde (n/a)

P5-A6: Wurde außerhalb der Prozesse definiert und validiert(n/a)

Es ist somit zu erkennen, dass auch bei der Validierung der einzelnen Prozesse im Detail, alle Anforderungen erfüllt werden können. Auf P2 sowie P4 treffen keine ausgewiesenen Anforderungen zu. P2 erfüllt die implizite Anforderungen der Unveränderbarkeit der Blockchain-Struktur, während P4 von „Medicalchain“ nicht explizit gefordert wird.

Das Resultat wurde in Tabelle 5 zusammengefasst:

Tabelle 5 - Validierung der Anforderungen anhand der Prozesse

	A1	**A2**	**A3**	**A4**	**A5**	**A6**
P1	**Ja**	**Ja**	n/a	**Ja**	n/a	n/a
P2	n/a	n/a	n/a	n/a	n/a	n/a
UP2.1	n/a	**Ja**	n/a	n/a	n/a	n/a
UP2.2	n/a	**Ja**	n/a	n/a	n/a	n/a
P3	n/a	n/a	n/a	n/a	**Ja**	n/a
P4	n/a	n/a	n/a	n/a	n/a	n/a
P5	n/a	n/a	**Ja**	n/a	n/a	n/a
P6	n/a	n/a	**Ja**	n/a	n/a	n/a

9.7 Ergebnis der Validierung

Die DSRM (siehe Kapitel 3.4) fordert nach dem Erstellen des Modells die Anwendung dessen auf einen praktischen Fall. Um dies umzusetzen, wurde das Referenzmodell auf „Medicalchain“ angewendet.

Die in Kapitel 9.4 beschriebenen Anforderungen konnten, nach Adaption des Referenzmodelles umgesetzt und erfüllt werden. Folglich eignet sich das erstellte

Referenzmodell ebenso, andere Anwendungsfälle, im diesem Fall „Medicalchain“, zu umzusetzen. Die Demonstration ist somit als erfolgreich einzustufen und das Referenzmodell als validiert zu betrachten.

10. Analyse und Interpretation des Referenzmodelles

In den vorangegangenen Kapiteln 8 sowie 9 wurde gezeigt, dass sich das Referenzmodell grundsätzlich eignet, alle rechtlichen Rahmenbedingungen zu erfüllen. Somit stünde einer Blockchain-Lösung beim Verarbeiten von Patientendaten nichts im Wege. Allerdings wurde der Mehrwert bis jetzt noch nicht in Betracht gezogen. Was genau soll die Blockchain verbessern? Hierfür wird Bezug zu den anfänglich definierten Hypothesen genommen:

1) Die Blockchain Struktur eignet sich, alle technischen und rechtlichen Auflagen und Anforderungen, welche von ELGA umgesetzt werden, zu erfüllen
2) Die Blockchain Struktur kann bei der Verarbeitung von Patientendaten die Sicherheit der Datenbestände verbessern
3) Die Blockchain Struktur kann bei der Verarbeitung von Patientendaten die Verfügbarkeit der Daten verbessern

Ad1) Dieser Punkt wurde bereits im vorherigen Kapitel 8.2 validiert.

Ad2) Die beim GDA gespeicherten Patientendaten sind ebenso verschlüsselt wie die Patientenakten in der e-Befund BC und können somit gleichermaßen gehackt werden. Der Vorteil von zweiterem liegt darin, dass eine unbemerkte Manipulation der Daten nicht möglich ist. Schafft es ein Angreifer oder eine Angreiferin bei ELGA (siehe Kapitel 1), die Daten zu verändern so kann dies nicht mehr nachvollzogen werden, während dies in der Blockchain Struktur per Design auffällt. Die Hypothese kann somit hinsichtlich der Integrität der Daten bzw. Datensicherheit bestätigt werden.

Ad3) Da die Patientendaten bei der Elektronischen Gesundheitsakte nur am Entstehungsort (dh. beim jeweiligen GDA) gespeichert werden, würde ein Ausfall, bewirken, dass diese Daten temporär nicht zur Verfügung stehen. Auch regionale Netzausfälle können den Zugriff behindern. Die Blockchain-Struktur hingegen

verbessert diese Situation, da jeder berechtigte Knoten, seine eigenen relevanten Datensätze besitzt und dadurch jederzeit einsehen kann. Zusätzlich verfügt jeder GDA und jeder Miner über die vollständige Blockchain-Struktur. Dies bedeutet, dass bei einem Ausfall von einem Knoten dennoch die Daten für alle anderen erreichbar bleiben. Die Verfügbarkeit der Daten ist somit gegeben und die Hypothese kann bestätigt werden.

Daraus folgt, dass alle drei Hypothesen bestätigt werden können.

11. Beantwortung der Forschungsfrage

Da alle drei Hypothesen bestätigt werden konnten, wird die Forschungsfrage mit dem in Kapitel 7 erstellen Referenzmodell beantwortet. Es ist somit möglich ein Referenzmodell zur Verarbeitung von Patientendaten mit Hilfe einer Blockchain-Struktur zu erstellen, welches alle rechtlichen Auflagen berücksichtigt und die Sicherheit, in Form der Integrität der Daten, sowie Verfügbarkeit verbessert.

12. Zusammenfassung

In dieser Arbeit wurden die grundlegenden Schwachstellen des derzeitigen Systems zur Speicherung und Verwaltung von Patientendaten mit Hilfe des Systems ELGA erläutert. Hierzu zählten, dass die Verfügbarkeit bei einem regionalen Ausfall eingeschränkt ist sowie, dass Daten bei einem Angriff manipuliert werden können. Eine mögliche Verbesserung dieser Problematiken wurden in der (privaten) Blockchain-Architektur vermutet. Diese bieten durch ihren Aufbau implizit die Möglichkeit, Daten dezentral und vollständig auf verschiedenen Rechnern bzw. Knoten zu verteilen. Dadurch sind Datenbestände (Transaktionen) einerseits bei jedem berechtigten Knoten lokal gespeichert und verfügbar, andererseits müsste eine Veränderung eines alten Datenbestandes auf allen berechtigten Knoten durchgeführt werden. Dies wird auf zwei unterschiedliche Arten unterbunden. Die Adresse, mit welche Transaktionen verlinkt werden ist sogleich der Hash, dadurch ist jede spätere Veränderung in der Blockchain sofort ersichtlich, da sich der Hashwert des geänderten Datensatzes ändert und somit die Referenz ungültig wird. Zweitens existiert ein Verfahren, um zu ermitteln, wie neue und gültige Blöcke zur Blockchain hinzugefügt werden und welcher Knoten dies bewerkstelligen darf. Im vorliegenden Modell wurde die Methode des „Mining Rotation“ mit „Blockchain-Anchoring“ kombiniert, um sowohl in der privaten Blockchain, sowie für alle nachvollziehbar den Konsens zu gewähren. Jeder neue Block wird intern im Rotationsverfahren erstellt und validiert. Der Header wird nun als Nachweis in eine öffentliche Blockchain integriert. Dies garantiert somit, dass keine ungültigen Blöcke hinzugefügt werden und gleichzeitig, dass vorhandene Daten nicht mehr verändert werden können. Gleichzeitig ist durch die dezentrale Verteilung aller Daten gewährleistet, dass diese bei einem Ausfall weiterhin verfügbar und somit weder temporär noch permanent ausfallen können. Zusätzlich zu diesen Verbesserungen müssen aber auch die existierenden rechtlichen Anforderungen erfüllt sein. Diese wurden anhand der letztgültigen DSGVO abgeleitet. Aufbauend darauf wurde ein Referenzmodell erstellt, welches die rechtlichen Anforderungen sowie die geforderten Verbesserungen erfüllt. Die Daten dürfen nur von unmittelbar

im Zusammenhang stehenden Personen (daher der Patient oder Patientin, und zu behandelnder GDA) einsehbar sein. Durch die Erstellung von verschlüsselten Inhalten und der Übermittlung an zufällig gewählten Zwischen-Adressen (AGI) kann für einen Dritten nicht nachvollzogen werden, welche Daten an wen gesendet wurden. Die Validierung erfolgte anhand der Gegenüberstellung aller Anforderungen mit den ausgearbeiteten Prozessen in Bezug auf ELGA sowie anhand eines weiteren Beispiels „Medicalchain“. Letzten Endes konnte gezeigt werden, dass sich das Referenzmodell eignet, alle rechtlichen Notwendigkeiten zu erfüllen und gleichzeitig die festgehaltenen Schwachstellen zu verbessern.

13. Ausblick

Obgleich sich eine Blockchain eignet, Patientendaten verschlüsselt, anonym, nachvollziehbar und de facto ausfallsicher zur Verfügung zu stellen, können die in den Hypothesen definierten Verbesserungen vorerst nur theoretisch bestätigt werden . Es folgt zwar, dass eine höhere Verfügbarkeit und Replikation der Daten die Ausfallsicherheit erhöht. Dennoch wird nicht jeder e-Befund unmittelbar nach der Erstellung benötigt. Ein kurzfristiger Systemausfall könnte hier eventuell zu keinen nennbaren Einbußen für GDAs oder Patienten führen, obwohl die Daten theoretisch nicht zur Verfügung standen. Hier müsste man zwei Systeme simulieren, um festzustellen, ob diese theoretischen Verbesserungen in einer praxisnahen Umsetzung auch tatsächlich stattfinden, dh auch für die Teilnehmer und Teilnehmerinnen eine messbare (dh durch KPIs festgelegte) Verbesserung hinsichtlich der praktisch benötigten Verfügbarkeit eintritt. Dieser Vorgang würde sich für weiter Arbeiten eignen.

Ebenso ist bei der vorliegenden Lösung eine zentrale Stelle, welche die Account-Informationen verwaltet, vorgegeben. Dies ist auf Grund der Identifikation des Patienten bzw. der Patientin und der Überprüfung der Patientendaten notwendig und wird bei ELGA ebenso gehandhabt. Es widerspricht hier dennoch der puristischen dezentralen Grundidee einer Blockchain, auch wenn die zuständige Behörde für Accounts generell keinen Einfluss auf den Aufbau und Ablauf der Blockchain selbst besitzt oder besitzen muss. Wird eine Lösung zur Verarbeitung von Patientendaten abseits der ELGA (oder der Krankenakte) angestrebt, so ist es, je nach Art und Verwendung der Daten selbst, nicht notwendig, eine zentrale Zertifizierungsstelle zu integrieren. Das Referenzmodell wäre für diese Anwendungsfälle zu adaptieren.

Weiters ist zu beachten, dass neben den verwendeten Konsens-Protokollen und Funktionsweisen von Blockchains, andere Protokolle existieren (siehe Kapitel 5.10), und hier sicherlich in Zukunft weitere Möglichkeiten hinzukommen werden. Eine logische Verbesserung des Modells könnte sein, zusätzliche Protokolle im

Baukastensystem in das Referenzmodell zu integrieren und dieses somit weiter zu abstrahieren. Die Anwendungsgebiete ließen sich dadurch potentieller weise deutlich erweitern.

Eine Entwicklung eines Prototyps würde sich aufbauend auf diese Arbeit ebenso anbieten. Hierbei könnte man Spezifikationen, wie die Benutzeroberfläche für Pateinten, GDA und Behörde, die zu verwendende Skriptsprache und dessen Befehle sowie Steuerung der Berechtigungsstruktur konkretisieren und die allgemeine Funktionalität der vorliegenden Lösung zu verbessern.

14. Literaturverzeichnis

[BA16] BARTSCH, STEFAN: *Ein Referenzmodell zum Wertbeitrag der IT, Entwicklung und Management von Informationssystemen und intelligenter Datenauswertung*. Wiesbaden : Springer Vieweg, 2015 — ISBN 978-3-658-09300-6

[Be04] BECKER, JÖRG ; DELFMANN, PATRICK: *Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung*, 2004 — ISBN 978-3-7908-2698-2

[Bi15] BITFURY GROUP ; GARZIK, JEFF: *Public versus Private Blockchains Public versus Private Blockchains Part 1: Permissioned Blockchains*, White Paper, 2015

[Bc19a] BLOCKCHAIN LUXEMBOURG S.A.: *Number Of Unique Addresses Used*. URL https://www.blockchain.com/de/charts/n-unique-addresses?timespan=all. - abgerufen am 2019-03-07

[Bc19b] BLOCKCHAIN LUXEMBOURG S.A.: *Total Number of Transactions*. URL https://www.blockchain.com/de/charts/n-transactions-total?timespan=all. - abgerufen am 2019-03-07

[Bs19] *BSI für Bürger - Hacker*. URL https://www.bsi-fuer-buerger.de/BSIFB/DE/Risiken/Hacker/hacker_node.html. - abgerufen am 2019-08-03

[BS98] *BSI für Bürger - Distributed Denial of Service - Verteilte Denial-of-Service-Attacken (DDoS)*. URL https://www.bsi-fuer-buerger.de/BSIFB/DE/Risiken/DoS/DDoS/DDoS.html. - abgerufen am 2019-08-03

[Bsif00] *BSI für Bürger - DoS - Denial of Service*. URL https://www.bsi-fuer-buerger.de/BSIFB/DE/Risiken/DoS/dos_node.html. - abgerufen am 2019-08-03

[Ci19] *Civic Secure Identity Ecosystem - Decentralized Identity & Reusable KYC*. URL https://www.civic.com/. - abgerufen am 2019-07-01. — Civic Technologies, Inc.

[DI16] DIEDRICH, HENNING: *Ethereum: blockchains, digital assets, smart contracts, decentralized autonomous organizations*. Preview 3. Lexington, KY : Wildfire Publishing, 2016 — ISBN 978-1-5239-3047-0

[Dr17] DRESCHER, DANIEL ; LENZ, G. (Übers.): *Blockchain Grundlagen: eine Einführung in die elementaren Konzepte in 25 Schritten*. 1. Auflage. Frechen : mitp, 2017 — ISBN 978-3-95845-654-9

[Dy19] DYNAMIC DOMAINS LLC: *Kazaar*. URL https://kazaa.descargar.es/contact/en/. - abgerufen am 2019-03-07

[El19a] ELGA GMBH: *ELGA: Startseite*. URL https://www.elga.gv.at/index.html. - abgerufen am 2019-03-07

[El19b] ELGA GMBH: *ELGA: Wissenswertes zu ELGA*. URL https://www.elga.gv.at/faq/wissenswertes-zu-elga/index.html. - abgerufen am 2019-03-14

[El19c] ELGA GMBH: *ELGA: Leitfäden*. URL https://www.elga.gv.at/technischer-hintergrund/technische-elga-leitfaeden/index.html. - abgerufen am 2019-03-07

[El19d] ELGA GMBH: *ELGA: Datensicherheit*. URL https://www.elga.gv.at/faq/datenschutz-und-datensicherheit/index.html. - abgerufen am 2019-03-14

[El19e] ELGA GMBH: *ELGA: Technischer Aufbau im Überblick*. URL https://www.elga.gv.at/technischer-hintergrund/technischer-aufbau-im-ueberblick/index.html. - abgerufen am 2019-03-07

[Fr19] EMEINNÜTZIGER VEREIN „FREUNDE DES AUSTRIA-FORUMS- VEREIN ZUR FÖRDERUNG DER DIGITALEN ERFASSUNG VON DATEN MIT ÖSTERREICHBEZUG“: *Liste der Bezirke und Statutarstädte in Österreich*. URL https://austria-forum.org/af/AustriaWiki/Liste_der_Bezirke_und_Statutarst%C3%A4dte_in_%C3%96sterreich. - abgerufen am 2019-02-01

[Eu16] EU: VERORDNUNG (EU) 2016/679 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung) (2016)

[Gl19] GLOBALSIGN: *Zertifizierungsstellen & Vertrauenshierarchien*. URL https://www.globalsign.com/de-de/ssl-information-center/zertifizierungsstellen-vertrauenshierarchien/. - abgerufen am 2019-02-01

[He10] HEVNER, ALAN R. ; CHATTERJEE, SAMIR: *Design research in information systems: theory and practice*, *Integrated series in information systems*. New York ; London : Springer, 2010 — ISBN 978-1-4419-5652-1

[Hl19] HL7 AUSTRIA MEDIAWIKI-BEARBEITER: *Implementierungsleitfaden*. URL https://wiki.hl7.at/index.php?title=Implementierungsleitf%C3%A4den. - abgerufen am 2019-03-07. — Implementierungsleitfaden

[LeKh00] LEE, EUNSOL ; KHO, ALLEN: *MediBloc - Reinventing Your Healthcare Experience!* URL https://medibloc.org. - abgerufen am 2018-04-30. — MediBloc

[Le19a] LEE, EUNSOL ; KHO, ALLEN: *MediBloc Testnet Wallet*. URL https://testnet-wallet.medibloc.org/#/. - abgerufen am 2019-05-02

[Le19b] LEE, EUNSOL ; KHO, ALLEN: *MediBloc technical whitepaper. Contribute to medibloc/whitepaper development by creating an account on GitHub* : MediBloc, 2019

[LI00] LIX, ROBERT: *Verteilte Systeme Client-Server-Computing für Studenten und Praktiker*. Wiesbaden : Vieweg+Teubner Verlag, 2000 — ISBN 978-3-322-93969-2

[LI10] LIX, ROBERT: *Moderne Applikationen von Peer-to-Peer-Technologien und dezentralen Netzen*, 2010 — ISBN 978-3-8366-4579-9

[Me18a] *Medicalchain - Blockchain for electronic health records*. URL https://medicalchain.com/en/. - abgerufen am 2018-04-30

[Me18b] MEDICALCHAIN: *Medicalchain Whitepaper 2.1*, 2018

[NA16] NARAYANAN, ARVIND: *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton : Princeton University Press, 2016 — ISBN 978-0-691-17169-2

[Oe19] ÖSTERREICHISCHE APOTHEKERKAMMER: *Aktuelle Apothekenanzahl*. URL https://www.apotheker.or.at/internet/oeak/NewsPresse.nsf/ca4d14672a08756bc125697d004f8841/2344b973f6958fdcc1256b37002fca3a?OpenDocument. - abgerufen am 2019-02-01

[Prus17] PRUSTY, NARAYAN: *Building Blockchain projects: develop real-time practical DApps using Ethereum and JavaScript*. Birmingham Mumbai : Packt, 2017 — ISBN 978-1-78712-214-7

[Ra16] RANDEN, HENDRIK JAN VAN ; BERCKER, C. ; FIEML, J. (Übers.): *Einführung in UML: Analyse und Entwurf von Software*. Wiesbaden : Springer Vieweg, 2016 — ISBN 978-3-658-14412-8

[Ru19] RUSSMEDIA DIGITAL GMBH: *Bezirke in Wien: Übersicht, Karte und Wissenswertes*. URL https://www.vienna.at/features/bezirke-wien. - abgerufen am 2019-02-01

[Se19] SECUPEDIA: *Hash-Funktion*. URL https://www.secupedia.info/wiki/Hash-Funktion. - abgerufen am 2019-05-02

[Sv19] SOZIALVERSICHERUNGS-CHIPKARTEN BETRIEBS- UND ERRICHTUNGSGESELLSCHAFT M.B.H. - SVC: *e-card Statistiken*. URL https://www.chipkarte.at/cdscontent/?contentid=10007.727761&viewmode=content. - abgerufen am 2019-03-07

[St19a] STATISTA: *Bevölkerung von Österreich von 2009 bis 2019 (in Millionen Einwohner)*. URL https://de.statista.com/statistik/daten/studie/19292/umfrage/gesamtbevoelkerung-in-oesterreich/. - abgerufen am 2019-02-01

[St19b] STATISTIK AUSTRIA: *Ärzte und Ärztinnen 2017 absolut und auf 100.000 Einwohner nach Bundesländern*. URL https://www.statistik.at/web_de/statistiken/menschen_und_gesellschaft/gesundheit/gesundheitsversorgung/personal_im_gesundheitswesen/022351.html. - abgerufen am 2019-02-01

15. Abbildungsverzeichnis

Abbildung 1 - Anzahl an Patientenkontakten pro Jahr Quelle: https://www.chipkarte.at[Sv19] 2

Abbildung 2 - Ablauf eines konstruktionsorientierten Forschungsansatzes Quelle: siehe [BA16, S.14] 10

Abbildung 3 - UML Klasse mit Attributen 12

Abbildung 4 - Multiplizität in UML 13

Abbildung 5 - Beispielhaftes Aktivitätendiagramm mit beschrifteten Symbolen Quelle: in Anlehnung an Abbildung 26 15

Abbildung 6 - Beispielhafter Laborbefund Quelle: ELGA Laborbefund 18

Abbildung 7 - Einzigartige Adressen bei Bitcoin Quelle: https://www.blockchain.com[Bc19a] 26

Abbildung 8 - Schematische Darstellung einer Blockchain Quelle: [NA16, S.XXI] 27

Abbildung 9 - Anzahl der Bitcoin Transaktionen Quelle: https://www.blockchain.com[Bc19b] 28

Abbildung 10 - Symmetrische Verschlüsselung (schematisch) Quelle: [Dr17, S.114] 31

Abbildung 11 - Asymmetrische Verschlüsselung (schematisch) Quelle: [Dr17, S.115] 31

Abbildung 12 - Erstellen der Digitalen Signatur (schematisch) Quelle: [Dr17, S.123] 33

Abbildung 13 - Verifizierung einer Signatur: Echt Vs. Unecht (schematisch) Quelle: [Dr17, S.124] 34

Abbildung 14 - Dezentral vs. Zentral Quelle: In Anlehnung auf [Dr17, S.31] 36

Abbildung 15 - Kazaa Netzwerk mit Super Nodes und Ordinary Nodes Quelle: [LI10, S.12] 38

Abbildung 16 - verknüpfte Daten als Kette Quelle: [Dr17, S.105] 42

Abbildung 17 - verknüpfte Daten als Baumstruktur Quelle: [Dr17, S.105] 43

Abbildung 18 – Datenstruktur (schematisch) Quelle: [Dr17, S.138] 52

Abbildung 19 - Änderungen in der Blockchain Quelle: In Anlehnung an [Dr17, S.144, 147] 53

Abbildung 20 - Merged Mining Quelle: [Bi15, S.17] 56

Abbildung 21 - Blockchain Anchoring Quelle: [Bi15, S.19] 58

Abbildung 22 - UML Klassen E-Befund Blockchain 74

Abbildung 23 - GDA Listen 76

Abbildung 24 - Erzeugung von Accounts 77

Abbildung 25 - Account und e-Befund 79

Abbildung 26 – Prozess: Erstellen eines e-Befundes 80

Abbildung 27 - Unterprozess Erstellt e-Befund 81

Abbildung 28 - Unterprozess Erstellt temporären e-Befund 82

Abbildung 29 - Aufbau der e-Befund Blockchain 84

Abbildung 30 - Erstelle Block 85

Abbildung 31 - Erstelle öffentliche Transaktion 87

Abbildung 32 - Anchor List 87

Abbildung 33 - Korrektur: Erzeugung von Accounts 99

Abbildung 34 - Ändere Zeitfenster (Patient) 101

Abbildung 35 - Ändere Zeitfenster (GDA) 102