Blockchain: Funktionsweise und Applikationsmöglichkeiten

Alexander Kaucher
Hochschule Bonn-Rhein-Sieg
Campus Sankt Augustin
Grantham-Allee 20, 53757 Sank Augustin
E-Mail: alex.kaucher@smail.inf.h-brs.de

***Abstract*—Ziel dieser Arbeit ist es, die Funktionsweise einer Blockchain zu beschreiben sowie unterschiedliche Applikationsmöglichkeiten vorzustellen. Dazu wurden Erkenntnisse aus diversen wissenschaftliche Arbeiten verknüpft und zusammengetragen. Diese Arbeit ist an jeden Leser mit einem bestehenden Interesse an das Thema Blockchain gerichtet.**

1. Motivation

Die erste verteilte Blockchain wurde 2008 von der anonymen Person oder Gruppe *Satoshi Nakamoto* konzipiert. Satoshi Nakamotos Arbeit beschreibt das elektronische Peer-to-Peer Zahlungssystem *Bitcoin*, welches grundlegende Probleme bei Online-Zahlungen lösen soll [1]. Vor Bitcoin galten exklusiv Finanzinstitute als vertrauenswürdige dritte Partei, welche elektronische Transaktionen zwischen zwei weiteren Parteien ermöglichten. Durch den Einsatz einer verteilten Blockchain werden nicht umkehrbare elektronische Transaktionen zwischen zwei Parteien direkt möglich. Eine dritte Partei, welche vertrauenswürdige Transaktionen garantiert, wird damit obsolet. Die Notwendigkeit einer vertrauenswürdigen dritten Partei wird bei einer Blockchain durch einen kryptografischen Beweis ersetzt.

Bis kryptografisch bewiesene Transaktionen letztendlich in eine Blockchain aufgenommen werden, müssen diese im verteilten Blockchain Netz zunächst akzeptiert werden. Dazu überprüfen andere Teilnehmer zusätzlich die Korrektheit des kryptografischen Beweises. Durch die Kombination von kryptografischen Beweisverfahren und Sicherstellung des Beweises im Netz mit Hilfe von verteilten Konsensmechanismen wird zusätzlich eine Lösung für das bis dahin bestehende *Double Spending Problem* gegeben.

Eine verteilte Blockchain wurde damit erstmals eingesetzt, um die Kryptowährung Bitcoin zu konzipieren. Sie stellt ein Peer-to-Peer System dar, welches als Alternative zu den bis dahin vorhandenen zentralisierten Systemen zur Transaktionsverifizierung eingesetzt werden kann und gleichzeitig grundlegende Probleme eines solchen Systems löst. Weitere Anwendungsfälle, bei denen eine Blockchain als Grundlage zur Transaktionsverifizierung dient, umfassen unter anderem *Smart Contracts* und *Dezentrale Autonome Organisationen*. Zusätzlich wird eine Blockchain auch beispielsweise bereits im öffentlichen Sektor, im Rechtswesen sowie im Bereich der *Internet of Things* eingesetzt.

2. Funktionsweise

Eine Blockchain (dt. Blockkette) ist eine chronische Aneinanderreihung bzw. Verkettung von Blöcken, welche über eine kryptografische Signatur miteinander verknüpft sind. Inhalt eines Blocks kann dabei alles sein, was digital abgebildet werden kann. Bei einer verteilten Blockchain hält jeder Teilnehmer des Peer-to-Peer Netzwerkes eine eigene Kopie der Blockchain. Durch die Verknüpfung der einzelnen Blöcke mittels kryptografischen Signaturen und der Verteilung der Blockchain auf allen Teilnehmern wird eine Manipulation bereits im Netz bestätigter Blöcke nahezu unmöglich und damit sicher vor Betrug.

Nachfolgend wird die Funktionsweise einer Blockchain anhand der Begriffe *Hash*, *Block*, *Blockchain*, *verteilte Blockchain*, *Tokens* und *Coinbase* im Detail erläutert [2]. Hierbei gilt es zu beachten, dass es sich bei der Funktionsweise von Absatz 2.1 *Hash* bis Absatz 2.6 *Coinbase* um eine beispielhafte Illustration eines fiktiven Systems handelt. Die Funktionweisen aus Absatz 2.1 *Hash* bis 2.4 *Verteilte Blockchain* sind dabei essentiell für eine Verteilte Blockchain. Die Funktionsweisen aus 2.5 *Tokens* und 2.6 *Coinbase* sind fiktiv und aus didaktischen Gründen mit aufgeführt. Sie dienen der Illustration eines stark vereinfachten, fiktiven kyptografischen Währungssystems.

Im Falle eines kryptografischen Währungssystems enthalten Blöcke Informationen über Transaktionen. Um den Inhalt eines Blocks eine gewisse Struktur zu geben (Transaktion) werden Tokens definiert. Zur Verifizierung der Transaktionen sowie zur Sicherstellung, ob das transferierte Gut überhaupt existiert, wird zusätzlich eine Coinbase benötigt.

2.1. Hash

Als Hash wird das Ergebnis der Anwendung einer Hashfunktion bezeichnet. Eine Hashfunktion bildet ein Datum beliebiger Länge auf einen Hash konstanter Länge ab. Zusätzlich ist die Eigenschaft gegeben, dass das gleiche Datum immer auf den selben Hash abgebildet wird. Dadurch ist es in der Informationstheorie möglich, die Integrität von Daten sicherzustellen. Der Inhalt eines Blocks wird unter anderem mit Hilfe einer Hashfunktion in einer Blockchain digital signiert.

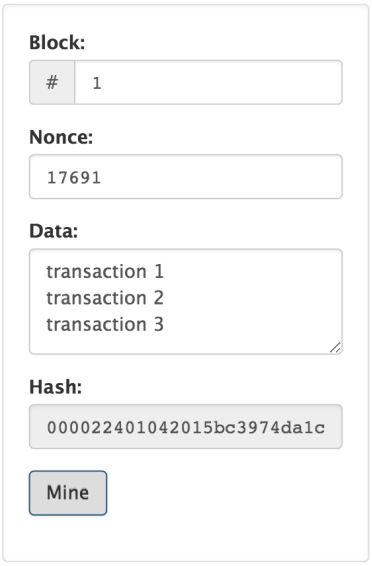

Abbildung 1. Ein Beispielprogramm, welches einen Block erstellt und die Signatur berechnet hat. Um den Block mit der Nummer 1 zu signieren, muss die Nonce den Wert 17691 darstellen.

2.2. Block

Bei der Erstellung der digitalen Signatur eines Blocks werden üblicherweise neben den Inhaltsdaten auch die Metadaten eines Blocks berücksichtigt, wie beispielsweise die Blocknummer. Die Signatur eines Blocks muss allerdings definierte Kriterien erfüllen. Wie in Abbildung 1 dargestellt muss der Hash beispielsweise mit vier vorangestellten Nullen beginnen. Anstatt nur über den Inhalt und die Blocknummer den Hash zu berechnen, wird ein dritter Parameter (Nonce) gesucht, welcher dazu führt, dass der Hash die definierten Kriterien erfüllt.

2.2.1. Mining. Der Prozess des Findens eines gültigen Hashes wird als *Mining* bezeichnet. Bei einer verteilten Blockchain in einem Peer-to-Peer Netzwerk versuchen sogenannte *Miner* als erstes für einen neuen Block einen gültigen Hash zu finden und diesen im gesamten Netz zu verteilen. Sollte von dem Großteil der Netzteilnehmer ein solcher Block für gültig befunden und damit akzeptiert werden, wird er in der Blockchain aufgenommen. Das in Abbildung 1 dargestellte Beispielprogramm verfügt über einen *Mine* Button. Bei Ausführung der *Mine* Funktion berechnet das Programm für alle Nonces, beginnend bei der Zahl 1, den Hash über *Blocknummer*, *Nonce* und *Data*. Sobald ein Hash gefunden wird, welcher den definierten Kriterien entspricht, stoppt das Programm.

2.3. Blockchain

Eine Blockchain besteht aus Blöcken, welche in einer chronischen Reihenfolge miteinander verknüpft sind. Jeder Block kennt zusätzlich zu den in Abbildung 1 dargestellten Informationen den Hash des vorherigen Blocks. Dadurch ist es in einer Blockchain möglich, ausgehend von einem beliebigen Block, rückwärts bis zum Ursprungsblock zu traversieren.

Durch die Verkettung der Blöcke führt eine Manipulation sowie ein anschließendes *Remining* eines vergangenen Blocks zu invaliden Folgeblöcken. Deshalb ist ein *Remining* des betroffenen Blocks in der Vergangenheit nicht ausreichend um Änderungen innerhalb eines vergangenen Blocks durchzuführen. Es müssen anschließend alle Folgeblöcke durch ein *Remining* korrigiert werden, damit der Hash eines Blocks durch die Änderung des Hashes des vorangestellten Blocks erneut valide wird. Der Aufwand, welcher betrieben werden muss, um einen Block zu *minen* ist in realen Systemen relativ rechenwenaufwändig. Diese Eigenschaft stellt eine der wichtigsten stärken einer Blockchain dar: **Resistenz gegen Manipulation vergangener Blöcke**.

2.3.1. Remining. Der Begriff des Reminings beschreibt die erneute Berechnung eines gültigen Hashes für einen Block, welcher bereits einen gültigen Hash besessen hat, wobei dieser durch die nachträgliche Manipulation der Daten invalide wurde.

2.4. Verteilte Blockchain

Eine verteilte Blockchain ist eine dezentrale, bei allen Teilnehmern des Peer-to-Peer Netzwerkes gespeicherte und verwaltete Blockchain. Jeder Teilnehmer hält eine identische Kopie der gesamten Blockchain. Dadurch wird das oben beschriebene Remining nahezu unmöglich. Der Vorteil wird anhand eines Beispielszenarios verdeutlicht.

Beispielszenario: Gegeben sei eine verteilte Blockchain mit drei Teilnehmern (A, B und C). Jeder Teilnehmer hält eine identische Kopie der Blockchain. Teilnehmer B möchte einen bereits in der Vergangenheit liegenden Block nachträglich manipulieren. Dafür nimmt er die Manipulation vor, betreibt das Remining für den betroffenen Block sowie für alle Folgeblöcke bis zum aktuellsten Block. Damit stimmen alle Hashes und somit die lokale Blockchain von Teilnehmer B. Sobald Teilnehmer B jedoch diese manipulierte Blockchain im Peer-to-Peer Netzwerk an Teilnehmer A und Teilnehmer C verbreiten möchte, werden beide Teilnehmer die manipulierte Blockchain ablehnen, weil sie in diesem Falle beispielsweise durch eine Mehrheitsabstimmung die Möglichkeit haben, ihre Kopien als die wahren Kopien zu identifizieren[1].

Neben der Erkennung von Manipulationsversuchen einzelner Teilnehmer besteht eine weitere wichtige Aufgabe bei einer verteilten Blockchain darin, neue Blöcke in die Blockchain aufzunehmen. Wie im Abschnitt 2.2.1 *Mining* beschrieben, versuchen Miner als erstes einen Hash für einen neuen Block zu berechnen. Dadurch ist es möglich, dass

1. Somit ist in einem solchen Szenario die Blockchain vor Manipulation genau so lange sicher, bis die Mehrheit der Peer-to-Peer Teilnehmer beispielsweise die Absicht hat, die Blockchain zu manipulieren. In diesem Falle würde bei einer Mehrheitsabstimmung die manipulierte Version als die wahre Blockchain angesehen werden.

in einem bestimmten Zeitinterval mehrere Miner versuchen einen neuen Block in die Blockchain aufzunehmen. Dabei kann es zu Konflikten kommen, wie z.B. dass drei unterschiedliche Miner teilweise komplett unterschiedliche und teilweise komplett identische Transaktionen in ihren Block eingetragen haben. Damit solch eine ***Aktualisierung der Blockchain*** konfliktfrei und korrekt im gesamten Peer-to-Peer Netzwerk durchgeführt werden kann, existieren bereits verteilte Konsensmechanismen. Diese werden detailliert im Abschnitt 3 erläutert.

2.5. Tokens

Mit Hilfe von Tokens kann dem Inhalt eines Blocks eine definierte Struktur zugrunde gelegt werden [2]. Beispielsweise kann bei Geldtransaktionen damit widerspruchsfrei der Absender, Empfänger sowie die Währung definiert werden. Zusätzlich vereinfachen Tokens die Möglichkeit, Transaktionen bis zu ihrem Ursprungsblock zurück zu verfolgen. In Abbildung 2 ist ein Block mit beispielhaften Geldtransaktionen zu sehen, welche mit Hilfe von Tokens strukturiert wurden. Dabei definiert *$* die Währung, *From:* den Absender und → den Empfänger.

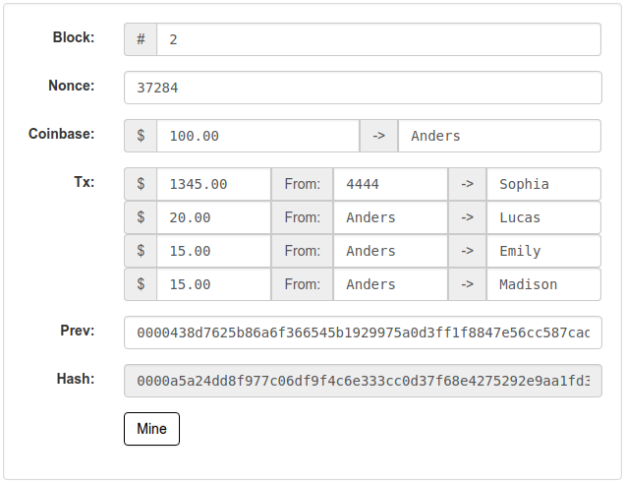

Abbildung 2. Ein Beispielprogramm, welches einen Block mit Vorgänger, Coinbase, Transaktionen, Nonce und Hash erzeugt hat.

2.6. Coinbase

Wie in dem Beispiel in Abbildung 2 zu sehen ist, existiert keine Information darüber, ob das transferierte Gut überhaupt bei dem entsprechenden Teilnehmer vorhanden ist. Zu diesem Zweck definiert [2] eine *Coinbase*[2]. Mit jedem neuen Block können zu transferierende Güter (in dem Beispiel Dollar) ohne Vorbedingung erzeugt werden. Durch eine Coinbase wird es damit möglich Transaktionen zu validieren, indem die Transaktionen eines Teilnehmers bis zur Coinbase zurück verfolgt werden können und damit sicherstellen, dass der entsprechende Teilnehmer über die zu transferierenden Güter verfügt.

Durch die Einführung von Tokens und die damit verbundene Strukturierung der Daten innnerhalb der Blöcke wird die Verfolgbarkeit von Transaktionsbewegungen möglich. Mit Hilfe der Coinbases wird den Transaktionen einen Wert zugewiesen, welcher bis zum Ursprung zurück verfolgbar und nachvollziehbar ist. In dem bisher beschrieben Beispiel ist der Teilnehmer *Anders* authorisiert eine Coinbase mit dem Dollarwert 100 zu erzeugen, um dieses Gut anschließend unter die anderen Teilnehmer zu verbreiten.

3. Aktualisierung der Blockchain

Das Beispielszenario aus Abschnitt 2.4 beschreibt einen bösartigen Netzteilnehmer, welcher versucht, in der Vergangenheit bereits abeschlossene Blöcke nachträglich zu manipulieren. Neben Manipulationsveruschen dieser Art existieren auch Manipulationsversuche für zuknftige, noch zu schreibende Blöcke. Dabei versucht ein bösartiger Netzteilnehmer absichtlich widersprüchliche Transaktionen zu versenden, welche dieselben Transaktionsinputs an unterschiedliche Adressen transferieren. Dies ist als das *Double Spending Problem* bekannt. Neben bösartigen Netzteilnehmern kann es bei einem Peer-to-Peer Netzwerk auch zu Verzögerungen oder gar Ausfällen von Netzknoten kommen und somit unterschiedliche Transaktionen in den jeweiligen Blöcken der Netzknoten enthalten sein [3].

Eine verallgemeinerte, theoretische Beschreibung der hier dargestellten Probleme ist ein in der theoretischen Informatik lange bekanntes Problem, welchen als das Problem der byzantinischen Generäle beschrieben wird [3]. Dabei müssen sich mehrere Generäle über einen Schlachtplan mittels Boten einigen, wobei einige Generäle bösartig sein können. Somit stellt das Peer-to-Peer Netz einer verteilten Blockchain eine vergleichbare Herausforderung dar [4].

Durch den in Bitcoin erstmalig angewandten verteilten Konsensmechanismus wird das Problem der byzantinischen Generäle gelöst. Dieser verteilte Konsensmechanismus wird oft als die **größte Innovation** hinter Bitcoin angesehen [3] und wird als ein sogenanntes *Proof-of-Work* Schema beschrieben [1].

3.1. Proof-of-Work

Proof-of-Work umfasst am Beispiel Bitcoin das Lösen einer rechenaufwändigen, mathematischen Aufgabe, welche von einem Mining-Netzknoten gelöst werden muss. Dabei handelt es sich bei Bitcoin um ein ähnliches Verfahren wie beispielhaft in Abschnitt 2.2.1 *Mining* dargestellt.[3] Ein Mining-Netzknoten versucht demnach als erstes für eine Menge von offenen Transaktionen, welche in den nächsten

2. Der Name Coinbase impliziert dabei nicht, dass es sich immer um Geld oder geldähnliche Tauschgüter handeln muss.

3. Das in Abschnitt 2.2.1 *Mining* beschriebene Beispielverfahren ist sehr stark vereinfacht dargestellt. Die zu lösende Aufgabe ist im realen Bitcoin System wesentlich komplexer und wird nach einer definierten Zeit erneut erschwert.

Block geschrieben werden sollen, diese Aufgabe so schnell wie möglich zu lösen. Nachdem der Mining-Netzknoten als erster den entsprechenden Hash gefunden hat, sendet er seinen Block an das Peer-to-Peer Netzwerk. Die Empfänger berechnen nun ebenfalls den Hash-Wert und nehmen den Block in ihre Blockchain auf, falls die Lösung valide ist. Demnach ist eine Menge von Transaktionen erst dann vollzogen, wenn sie in die Blockchain aufgenommen wurde. [1]

3.1.1. Beispiel. Das oben beschriebene Proof-of-Work Verfahren wird in Abbildung 3 in zwei Phasen unterteilt [3]. Die erste Phase *Block-Validierung* stellt die Suche nach dem Hash dar, während die zweite Phase *Blockchain-Update* das Senden eines neuen Blocks an das Peer-to-Peer Netzwerk abbildet. Der Block besteht aus einer Menge offener Transaktionen sowie einem Block-Header. Nun wird anhand definierter Kriterien eine Hashfunktion so lange angewendet, bis der Hashwert kleiner als ein bestimmter Zielwert ist. Anschließend wird der Block versendet.

3.1.2. Problem: Gabelung. Da mehrere Miner gleichzeitig an der Suche nach neuen, validen Blöcken arbeiten, kann es vorkommen, dass zwei Mining-Netzknoten beinahe gleichzeitig ihre Lösung finden und diese an das Peer-to-Peer Netzwerk versenden. Dieses Phänomen tritt bei ca. 1,69% aller Blöcke auf und wird als *Gabelung* bezeichnet [4]. Daraus folgt, dass zu einem bestimmten Zeitpunkt mehrere valide Versionen einer Blockchain existieren können. Anstatt das Problem sofort zu lösen, arbeiten die Mining-Netzknoten so lange auf Basis ihrer Blockchain weiter, bis sie über eine längere Blockchain benachrichtigt werden. Die jeweils *längste* bekannte Blockchain wird vom Netzwerk letztendlich als korrekt erachtet, wobei die *Länge* durch die höchste aggregierte Rechenleistung in Form des Proof-of-Work Schemas bestimmt wird.

Nachdem eine Gabelung vom Netzwerk erfolgreich beseitigt wurde kann es vorkommen, dass Transaktionen, welche vorher in einer anderen validen Blockchain enthalten waren, folglich nicht in der aktuellsten Blockchain vorhanden sind. Solche Transaktionen werden anschließend wieder für die Miner freigegeben, um sie in zukünftigen Blöcken aufzunehmen. Deshalb werden etwa sechs Blöcke ab der Transaktionsvollendung als angemessene Bestätigungszeit angesehen [5].

3.1.3. Belohnung. Das Proof-of-Work Schema von Bitcoin sieht es vor, Miner für ihre Investition von Rechenleistung dadurch zu motivieren, dass sie für einen gefunden Block eine gewisse Anzahl an Bitcoins erhalten. Folglich werden mit jedem neu gefundenen Block auch neue Bitcoins erzeugt. Neben den neu erzeugten Bitcoins können auch Transaktionsgebühren, welche in den Transaktionen inkludiert wurden, an die Miner ausgeschüttet werden [3].

3.1.4. Kritik. Aufgrund der immer weiter erschwerten, zu lösenden, mathematischen Aufgaben in dem Proof-of-Work Schema von Bitcoin wird auch stetig mehr rechenleistung benötigt, um neue Blöcke zu finden. Deshalb wird das Bitcoin System als energieineffizient und nicht umeltfreundlich kritisiert.

3.2. Proof-of-Stake

Bei Proof-of-Stake wird im Gegensatz zu Proof-of-Work kein Wettbewerb um das schnellste Lösen einer mathematischen Aufgabe zur Generierung neuer Blöcke betrieben. Hier liegt die Grundidee darin, den Erzeuger des nächsten Blocks dadurch zu bestimmen, welcher einen großen Anteil an der Währung bzw. generell Werten in der Blockchain hält [6]. Dadurch besteht ein Anreiz für eine korrekte Aufrechterhaltung des Systems bei genau diesen Teilnehmern.

3.2.1. Beispiel. *Peercoin*[4] wählt den nächsten Blockerzeuger aus der Menge aller Teilnehmer basierend auf dem Alter seiner Coins. Für jeden Teilnehmer wird berechnet, wie viele Coins er besitzt. Anschließend wird diese Zahl mit dem Alter der Coins multipliziert. Der Teilnehmer mit der größten Zahl ist für die Erzeugung des nächsten Blocks zuständig.

Dabei müssen Coins für die Berücksichtigung der Berechnung mindestens 30 Tage alt sein. Folglich haben Teilnehmer, welche ältere sowie größere Mengen an Coins besitzen, eine höhere Chance als Erzeuger des nächsten Blocks gewählt zu werden. Nachdem Coins für das Erzeugen und signieren eines Blocks verwendet wurden, wird ihr Alter auf 0 zurück gesetzt. Das maximale Alter für Coins liegt bei 90 Tagen, um vor einer Dominierung der Blockerzeugung von Teilnehmern mit sehr vielen alten Coins zu schützen.

Ein Vorteil gegenüber dem Proof-of-Work Schema besteht darin, neue Blöcke und somit auch neue Coins ohne großen Rechenaufwand erzeugen zu können (siehe Abschnitt 3.1.4). Dazu soll der Schutz vor Betrug bei Proof-of-Stake höher als bei Proof-of-Work sein. Der Erwerb von mehr als der Hälfte aller Coins wird kostenspielieger als der Erwerb von mindestens 51% der benötigten *Hashing Power* eingeschätzt.

3.2.2. Kritik. Weil kein großer Rechenaufwand zur Blockgenerierung benötigt wird, kann der Teilnehmer, welcher den nächsten Block generiert, unter der Voraussetzung eines *Consensus Failure*[5], für mehrere Blockchain Historien stimmen. Dies würde dazu führen, dass der *Consensus Failure* in der Zukunft nie behoben würde.[6] Dieses Phänomen ist als das *Nothing at Stake* Problem bekannt. Dadurch können in

4. https://peercoin.net

5. Ein **Consesus Failure** tritt auf, sobald mehrere Versionen einer validen Blockchain im Netzwerk bekannt sind. Wie bei Proof-of-Work handelt es sich bei Proof-of-Stake im Abschnitt 3.2.1 um eine stark vereinfachte Darstellung eines Proof-of-Stake Schemas. In realen Systemen kann es aufgrund der erhöhten Komplexität des Proof-of-Stake Schemas durchaus zu mehreren validen Blockchain Versionen führen.

6. Bei Proof-of-Work wäre das beispielsweise eine nie gelöste Gabelung. Wird dieser Gedanke weiter geführt und bei jeder Blockgenerierung angewendet, so wird es in der Theorie nie eine längste Blockchain geben, welche im Netzwerk als die Richtige anzusehen ist.

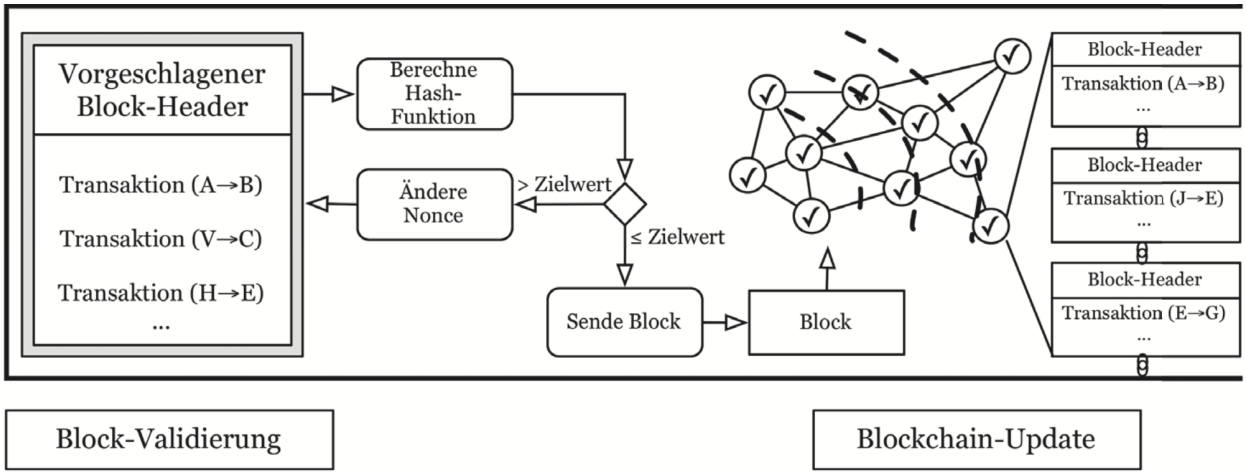

Abbildung 3. Die zwei Phasen des Proof-of-Work Schemas von Bitcoin: **Block-Validierung** und **Blockchain-Update**.

den unterschiedlichen Blockchain Versionen auch Transaktionen doppelt vorkommen und somit das ursprünglich zu lösende *Double Spending Problem* erneut hervorrufen.

3.3. Proof-of-Burn

Das Proof-of-Burn Schema stellt einen zum Proof-of-Work und Proof-of-Stake Schema alternativen, verteilten Konsensmechanismus dar und wurde mit der Kryptowährung *Slimcoin*[7] von *Iain Stewart* erfunden [7]. Das Prinzip ist dabei das gleiche wie bei dem Proof-of-Work Schema mit dem Unterschied, dass keine große Rechenleistung zur Blockerzeugung benötigt wird [8]. Damit ist das Proof-of-Burn Schema im Gegensatz zum Proof-of-Work Schema energieeffizient und umweltfreundlich.

3.3.1. Beispiel: Slimcoin. In einer Proof-of-Burn Blockchain bekommen die Miner erst dann das Recht neue Blöcke zu erzeugen, wenn sie zuvor beweisen können, dass sie bereits existierende Coins, welche durch eine Proof-of-Work Blockchain erzeugt wurden, an eine sogenannte *Burn*-Adresse gesendet haben. Coins, welche an eine Burn-Adresse gesendet wurden, können nicht zurück erstattet werden und sind somit ab diesem Zeitpunkt nutzlos. Dieser Prozess soll beabsichtigt teuer sein und simuliert den Kauf eines *virtuellen Mining-Rigs*. Erst durch den Erwerb eines virtuellen Mining-Rigs ist ein Miner dazu berechtigt an der Erzeugung neuer Blöcke mitzuwirken. Das virtuelle Mining-Rig existiert eine definierte Zeit lang und wird dabei mit zunehmeder Zeit immer langsamer, bis es irgendwann nutzlos wird. Dies soll dem Kauf eines echten Hardware-Mining-Rigs dahingehend simulieren, dass die Hardware mit voranschreitender Zeit immer langsamer gegenüber neuerer Hardware wird (*Moore's Law*). Somit ist das *Verbrennen* von Coins eine Analogie zu dem Kauf eines Mining-Rigs, um dieses für die Generierung neuer Blöcke zu nutzen.

7. http://www.slimco.in

3.3.2. Vorteile gegenüber Proof-of-Work. Während für die Blockerzeugung bei einer Proof-of-Work Blockchain echte Hardware und Rechenleistung zur Generierung neuer Blöcke benötigt wird, fallen bei einer Proof-of-Burn Blockchain lediglich die *Kosten* dieser Hardware und Rechenleistung an. Dadurch ist der Energieverbrauch bei einer Proof-of-Burn Blockchain sehr gering verglichen zu der Proof-of-Work Blockchain. Außerdem muss ein Miner keine Investition für echte Hardware tätigen, da er durch das Verbrennen seiner Coins virtuelle Hardware erlangt. Durch das Erlangen der virtuellen Hardware wird er dazu berechtigt an der Blockerzeugung mitzuwirken und kann dafür die *Leistung* seiner virtuellen Hardware verwenden.

3.3.3. Vorteile gegenüber Proof-of-Stake. Während bei dem Proof-of-Stake Schema die Halter vieler alter Coins das Recht bekommen neue Blöcke zu erzeugen, versucht das Proof-of-Burn Schema hier das vereinfachte Modell *Reiche werden Reicher* zu vermeiden. Erst durch die Investition von Coins, bei denen nur der Wert und nicht das Alter der Coins wichtig ist, kann ein Recht zur Blockerzeugung erlangt werden. Außerdem sind *verbrannte* Coins nie wieder verwendbar, während bei einer Proof-of-Stake Blockchain ein Halter vieler alter Coins durch einen Hackerangriff verletzbar wird. Wenn seine Coins gestohlen werden, kann ein Hacker das Recht erlangen, bei einer Proof-of-Stake Blockchain an der Blockerzeugung mitzuwirken.

3.3.4. Kritik. Das Proof-of-Burn Schema versucht zu garantieren, dass auf lange Zeit gesehen mehr Coins durch die virtuellen Mining-Rigs erzeugt werden können als ursprünglich dafür *verbrannt* wurden. Dies bietet neuen Blockchains zwar ein gutes initiales Potential, bringt aber auch die Gefahr mit sich, dass Proof-of-Burn Blockchains nur aus dem Grund der Blockerzeugung favorisiert werden, ohne den Nutzen hinter dem System zu hinterfragen.

Die Kernidee von Proof-of-Burn Blockchains basiert auf zuvor aus Proof-of-Work Blockchains erzeugtem Gut und wird damit wahrscheinlich nur in der Kryptowährung

Anwendung finden. Dafür bietet das Proof-of-Burn Schema eine gute Möglichkeit an, auf Basis bereits bestehender Kryptowährungen neue Kryptowährung zu erzeugen.

3.4. Proof-of-Activity

Während das Proof-of-Burn Schema eine Alternative zum Proof-of-Work und Proof-of-Stake Schema darstellt, handelt es sich bei dem Proof-of-Activity Schema um einen Hybrid dieser beiden Schemata. Dabei liegt der Fokus und die Motivation darin, ein sicheres System für zukünftige Angriffe gegen die Kryptowährung Bitcoin präventiv zu entwickeln. Durch die Kombination von Proof-of-Work und Proof-of-Stake soll ein höherer Schutz vor Angriffen gewährleistet werden, als er bei dem reinen Einsatz von Proof-of-Work oder Proof-of-Stake zur Blockerzeugung bisher vorhanden ist [9].

3.4.1. Beispiel. Um einen neuen Block zu Erzeugen versuchen Miner wie bei Proof-of-Work zunächst eine rechenaufwendige, mathematische Aufgabe zu lösen. Sobald ein Mining-Netzknoten als erster die Aufgabe gelöst hat bereitet er den Block vor, um ihn im gesamten Netz wie bei Proof-of-Work zu verteilen. Der Unterschied zu Proof-of-Work ist jedoch, dass es sich bei dem Block zunächst um einen unfertigen Block (Template) handelt und das der Block nicht von allen Teilnehmern validiert werden muss, sondern in einem Proof-of-Stake ähnlichen Prozess überführt wird. Basierend auf die in dem Blockheader vorhandenen Informationen werden zufällig Teilnehmer zur Blockvalidierung bestimmt. Der Algorithmus zur Bestimmung der Teilnehmer berücksichtigt dabei eine Gleichverteilung der vorhandenen Coins. Folglich haben Teilnehmer, welche eine große Anzahl an Coins halten, eine höhere Chance zu den gewählten Validierungsinstanzen zu gehören. Erst nachdem alle Validierungsinstanzen den unfertigen Block signiert haben wird der Block in die Blockchain aufgenommen. Anschließend wird die Belohnung zwischen den Minern und den beteiligten Validierungsinstanzen verteilt.

3.4.2. Vorteil gegenüber einem reinen Proof-of-Work Schema. Während ein Angreifer bei einem reinen Proof-of-Work System *nur* 51% der vorhandenen *Hashing Power* besitzen muss, reicht dies bei Proof-of-Activity noch nicht aus. Der Angreifer könnte lediglich die unfertigen Blöcke erzeugen. Um diese auch signieren zu können, muss er aufgrund des Gleichverteilungsalgorithmus zusätzlich einen sehr großen Teil der Coins besitzen. Folglich muss er die Sicherheitshürden von Proof-of-Work und Proof-of-Stake Systemen überwinden, um für eine Proof-of-Activity Blockchain gefährlich zu werden.

3.4.3. Vorteil gegenüber einem reinen Proof-of-Stake Schema. Bei einem Proof-of-Stake System werden neue Blöcke zu einer sehr hohen Wahrscheinlichkeit nur von den Teilnehmern erzeugt, welche einen sehr großen Anteil alter Coins besitzen. Diese Eigenschaft führte in der Vergangenheit zu verschiedenen Kritiken. Zum einen wird dem Schema vorgeworfen, es würde zu einem *Reiche werden Reicher* Modell führen und zum anderen könnte das Peer-to-Peer Netz von einigen, wenigen Teilnehmern kontrolliert werden. Diese Eigenschaften können in einem Proof-of-Activity System nicht auftreten, da ein großer Teil der Belohnung für die Erzeugung eines neuen Blocks an die Miner ausgeschüttet wird [10].

3.4.4. Kritik. Obwohl der Schwierigkeitsgrad der rechenaufwendigen, mathematischen Aufgabe konstant bleiben kann, darf er einem bestimmten Minimum an Schwierigkeit nicht unterliegen. Aus diesem Grund wird dem Proof-of-Activity Schema ebenfalls, wie auch schon dem Proof-of-Work Schema, eine zu hohe Energieineffizienz vorgeworfen.

Zusätzlich werden dem Schema aufgrund des Gleichverteilungsalgorithmus ebenfalls die gleichen Kritikpunkte wie dem Proof-of-Stake Schema vorgeworfen.

Insgesamt sorgt Proof-of-Activity damit für einen besser gesicherten, verteilten Konsensmechanismus als die reinen Proof-of-Work und Proof-of-Stake Schemata. Gleichzeitig unterliegt es jedoch der Kritik beider Systeme.

3.5. Weitere verteilte Konsensmechanismen

Das in Abschnitt 3.1 *Proof-of-Work* dargestellte Verfahren wurde ursprünglich zur Realisierung des Bitcoin Systems implementiert und gehört deshalb zu den ersten, bekannten verteilten Konsensmechanismen. Mit der Entwicklung neuer Kryptowährungen wurden neuere Mechanismen entwickelt, welche das Ziel verfolgen, die Schwachstellen bzw. negativen Eigenschaften bereits bestehender Mechanismen zu beseitigen. Einen alternativen Ansatz zum Proof-of-Work Schema stellt dabei das in Abschnitt 3.2 *Proof-of-Stake* beschriebene Verfahren dar. Basierend auf den Ideen dieser verteilten Konsensmechanismen entstanden die in den Abschnitten 3.3 *Proof-of-Burn* und 3.4 *Proof-of-Activity* beschriebenen Verfahren. Bei Proof-of-Burn handelt es sich um einen alternativen Ansatz zu Proof-of-Work und Proof-of-Stake, während es sich bei Proof-of-Activity um eine Kombination dieser handelt. Neben den vier beschriebenen Verfahren existieren bereits zahlreiche weitere Verfahren. Dazu gehören unter anderem beispielsweise *Proof-of-Publication* [4] und *Proof-of-Storage* [6].

Die dargestellten verteilten Konsensmechanismen und Beispiele bezogen sich bisher auf realen, stark vereinfachten Kryptowährungssystemen. Die Blockchain findet heutzutage allerdings auch schon Anwendung ausserhalb der Kryptowährungen. Das nachfolgende Kapitel beschreibt weitere Applikationsmöglichkeiten der Blockchain.

4. Blockchain Applikationen

Die erste praktische Anwendung der Blockchain stellen *Kryptowährungen* bzw. die konkrete Kryptowährung Bitcoin dar. Mittlerweile haben sich neben den Kryptowährungen noch weitere Anwendungsfälle für die Blockchain ergeben. Einen zweiten, bereits praktischen Anwendungsfall stellen die sogenannten *Smart Contracts* dar. Diese bestehen aus

computergestützen Verträgen, welche ohne die Notwendigkeit eines Mittelsmann konfliktfrei zwischen zwei Parteien verhandelt und abgeschlossen werden können. Basierend darauf und mit Hilfe der Smart Contracts können *Dezentrale Autonome Organisationen* (DAO) erschaffen werden. Eine DAO ist eine blockchain-basierte, autonome, strukturierte nicht-natürliche Organisationseinheit, die ohne jegliche zentrale Weisung selbständig Entscheidungen auf der Basis unveränderlichen Computercodes trifft [11].

Außerhalb der Verwendung im Bereich der Wirtschaft wird eine mögliche Rolle der Blockchain unter anderem im *Öffentlichen Sektor* sowie im *Rechtswesen* diskutiert. Zuletzt wird eine mögliche Verwendung der Blockchain im Bereich der *Internet of Things* beschrieben.

4.1. Kryptowährungen

Kryptowährungen gab es schon vor Bitcoin. Diese waren jedoch zentralisiert und boten damit einen einzelnen Angriffspunkt an, weshalb diese letztendlich gescheitert sind [12]. Durch seinen dezentralisierten Ansatz hat das Bitcoin System dieses Problem gelöst. Das dezentralisierte Peer-to-Peer Netz erstellt und verifiziert dabei Transaktionen innerhalb des Netzwerks. Durch ein Proof-of-Work Schema werden anschließend neue Währungseinheiten erzeugt. Diese sind dabei in ihrem Wert nicht von einer Regierung oder Organisation gestützt [3].

Die Webseite *coinmarketcap*[8] listet zu diesem Zeitpunkt (stand 18. Dezember 2017) mittlerweile 1364 Kryptowährungen mit einer kombinierten Marktkapitalisierung von annähernd 600 Mrd. US-Dollar auf.

Nachfolgend einige Vor- und Nachteile von Kryptowährungen[9] [3]:

4.1.1. Vorteile.

- Teilbarkeit in sehr kleine Einheiten
- Geringe Transaktionskosten und -zeiten durch die Umgehung von Itermediären
- Teilweise erhöhte Privatsphäre, da keine Bindung an Bankkonten, die einen Nachweis der Identiät benötigen
- Unmöglichkeit der Fälschung von Kryptowährungen aufgrund der Eigenschaften der Blockchain
- Erhöhte Sicherheit aufgrund der Umgehung von Intermediären

4.1.2. Nachteile.

- Aktuell eine hohe Fluktuation der Wechselrate von Kryptowährungen
- Probleme hinsichtlich des Konsumentenschutzes, da Transaktionen irreversibel sind
- Instrumentualisierung des hohen Grads an Anonymität für kriminelle Aktivitäten: insbesondere Geldwäsche, die Bezahlung illegaler Güter oder Dienstleistungen und Terrorfinanzierung sowie potenziell als Mittel zur Steuerhinterziehung
- Hackerangriffe möglich, sofern die Währungen online in sogenannten E-Wallets gespeichert werden

4.2. Smart Contracts

Bereits 1997 wurde das Konzept der *Smart Contracts* von Nick Szabo eingeführt und als computerbasiertes Transaktionsprotokoll definiert, welches die Bedingungen eines Vertrages implementiert. Aufgrund ihrer Eigenschaft bietet die Blockchain erstmals ein geeignetes Medium zur Implementierung solcher Kontrakte [3].

Bei Smart Contracts handelt es sich um Computerprogramme, welche Entscheidungen basierend auf bestimmten Konditionen treffen können. Dazu können externe Informationen als Inputs dienen, welche dann über festgelegte Regeln des Vertrages eine bestimmte Aktion ausführen. Ziel dabei ist es, menschliche Interaktionen zu automatisieren sowie die Bearbeitungszeit, welche bei herkömmlichen Verträgen auftauchen, zu eliminieren. Die eingesetzten Algorithmen können diese Kontrakte dann verifizieren und anschließend ausführen oder verwerfen.

Die Plattform *Ethereum*[10] bietet eine Möglichkeit zur Implementierung von Smart Contracts basierend auf einer Blockchain an. Beispielsweise können Besitztümer wie Autos oder Wohnungen über einen *Smart Key* und ein Blockchain System ohne physische Schlüsselübergabe vermietet werden.

4.2.1. Beispiel. Person A möchte von Person B ein Auto mieten. Dazu wird ein Smart Contract erzeugt. Dieser beschreibt zwei Bedingungen. Die erste Bedingung fordert, dass Person A seinen digitalen Autoschlüssel (Smart Key) an den Kontrakt *transferiert*. Die zweite Bedingung fordert, dass Person B den ausgehandelten Betrag an den Kontrakt *transferiert*[11]. Zusätzlich können im Smart Contract Regeln für die Zugangs- und Nutzungsberechtigung hinterlegt werden. Die Zahlungseingänge sowie die Berechtigungsverwaltung erfolgen transparent, sicher und unveränderbar über die Blockchain. Erst nachdem beide Bedingungen eingetroffen sind wird der ausgehandelte Betrag an Person A und der Smart Key an Person B von dem Smart Contract weitergeleitet. Eine dritte Bedingung könnte ein Zeitpunkt sein, zu dem die ersten beiden Bedingungen erfüllt werden müssen. Der Kontrakt wird abgebrochen sobald der Zeitpunkt überschritten wurde.

Nachfolgend einige Chancen und Risiken von Smart Contracts gegenüber bisheriger Vertragsschließungen zwischen zwei Parteien [3]:

8. https://coinmarketcap.com

9. Eine vollständige Ausarbeitung zum Thema Kryptowärhungen bildet [15].

10. https://ethereum.org

11. Dabei handelt es sich um Beträge in Kryptowährungen.

4.2.2. Chancen.

- Autonome Ausführung des Vertrages; störende Eingriffe dritter Parteien in der Ausführung folglich nicht möglich
- Vertragsausführung in Echtzeit
- Geringe Vertrags-, Durchsetzungs-, und Compliance-Kosten im Vergleich zu regulären Verträgen; allgemein niedrigere Kosten der Ausführung, da Smart Contracts aufgrund ihrer Implementierung via Quellcode leicht zu standardisieren sind
- Möglichkeit, die Ausführung eines Smart Contracts von externen Ereignissen abhängig zu machen
- Fairer Austausch zwischen zwei Vertragsparteien ohne itermediäre Partei möglich, selbst wenn sich die Vertragsparteien nicht gegenseitig vertrauen
- Minimierung der Interaktion zwischen den Vertragsparteien

4.2.3. Risiken.

- Exakte und garantierte Ausführung eines Smart Contracts nach seiner Implementierung; Unmöglichkeit des Rückzugs einzelner Vertragsparteien kann jedoch auch als Vorteil gesehen werden
- Hohe Abhängigkeit von dem jeweils ausführenden System
- Rechtliche Probleme, wie beispielsweise die Relation von Smart Contracts zu konventionellem Vertragsrecht oder dem Verbraucherschutz; generell Frage der rechtlichen Verantwortung, da Verträge durch ein Computerprogramm anstelle einer rechtlichen Entität ausgeführt werden
- Einschränkung des Umfangs von Smart Contracts durch die Notwendigkeit, die jeweiligen Interaktionen durch Daten ausdrücken zu können
- Maximale Vorteile von Smart Contracts bei der Verwendung durch viele Unternehmen, wobei jedoch zunächst ein Fachkräftemangel für die Implementierung auftreten könnte

4.3. Dezentrale Autonome Organisationen

Eine DAO wird sehr generisch als ein dezentrales Netzwerk autonomer Subjekte definiert, denen eine leistungsmaximierende Produktionsfunktion zugrunde liegt [13]. Sie sind als neuartige Organisationform anzusehen in denen sowohl Menschen als auch Geräte miteinander kooperieren können[12]. Eine DAO agiert auf Basis der ihr zugrundeliegenden Geschäftsregeln und Prozesse, welche durch Smart Contracts vorgegeben sind. Folglich operiert eine DAO ohne menschlichen Einfluss. Sobald eine DAO über eine Blockchain implementiert wurde, kann sie für ihre erbrachten Leistungen sowohl von ihren Nutzern Kompensation einfordern als auch für notwendige Ressourcen selbst bezahlen [3].

12. Wenn solch eine Organisation ein Profitiziel verfolgt handelt es sich um eine *Decentralised Autonomous Corporation* (DAC).

4.3.1. Beispiel. Eine imaginäre *SonntagsFußballDAO*, gegründet von 30 sportbegeisterten Männern in der Nachbarschaft, kann dafür sorgen, dass ein gemeinsames Fuballspiel automatisch vereinbart wird, sobald mindestens 11 Teilnehmer Sonntags zwischen 15-17 Uhr Zeit haben und vor Ort sind. In diesem Fall müsste ein Dienstleister die Verfügbarkeit der Teilnehmer in ihren Kalendern prüfen und sicherstellen, dass der Fuballplatz frei ist. Eventuell muss die Online-Buchung des Platzes durchgeführt werden. Nachdem mindestens 11 Teilnehmer zugesagt haben, wird allen der Termin in den Kalender geschrieben. Die jeweiligen Regeln, auf Basis derer die Dienstleister agieren, sind Bestandteile des Smart Contracts [11].

4.3.2. Kritik. Zu den Vorteilen einer DAO gegenüber regulären Organisationen zählt, dass jede Entscheidung transparent in der Blockchain nachvollzogen werden kann. Zusätzlich liegt das Vertrauen in dem der DAO zugrundeliegenden, offenen und überprüfbaren Quellcode, anstatt bei einer zentralen Organisation.

Durch die Struktur und dezentralisierte Eigenschaft einer DAO werden jedoch Fragen hinsichtlich Haftung und Verantwortung bisher nicht eindeutig beantwortet [3].

4.4. Öffentlicher Sektor

Mit Hilfe der Blockchain Technologie entstanden bereits Szenarien, in denen einzelne Bereiche des öffentlichen Sektors effizienter gestaltet werden können. So könnten in den kommenden zehn Jahren Steuern erstmalig durch die Regierung über ein Blockchain-System eingezogen werden. Dieses Anwendungsbeispiel wird als technologisch bereits möglich eingestuft [3]. Als Vorteile werden dabei eine erhöhte Transparenz sowie eine Verringerung der administrativen Kosten in Bezug auf die Bezahlung und Einforderung von Steuern genannt.

Ein weiteres Szenario wird für Länder beschrieben, in denen Korruption in der Verwaltung ein Problem darstellt. So könnten beispielsweise Wahlen mittels der Blockchain korruptionsfrei und transparent gehalten werden. Jeder Bürger könnte schließlich mit seinem öffentlichen Schlüssel seine Stimme einmalig abgeben [13].

4.5. Rechtswesen

Neben dem Einsatz der Blockchain Technologie im öffentlichen Sektor wird außerdem über mögliche Anwendungsfelder im Rechtswesen diskutiert. Dabei liegt die Betrachtung auf die Eigenschaften einer manipulationssicheren Datenbank verbunden mit den Konzepten der Smart Contracts und DAOs. Dadurch sollen Anwendungsfälle im Bereich der Verifikation von Urheberschaft und Dokumenteninhalten, der Übertragung von Eigentumsrechten sowie der Vertragsdurchsetzung entstehen [3].

Als möglicher Vorteil wird das Rechtsmanagement digitaler Objekte gesehen, welches durch Smart Contracts realisiert wird. Folglich würden viele Rechtsgeschäfte keine Anwälte mehr benötigen. Zusätzlich werden Schwachstellen

herkömmlicher rechtlicher Verträge, die durch Mehrdeutigkeiten der natürlichen Sprache entstehen, durch einen eindeutigen Smart Contract Programmcode beseitigt.

Als potentieller Nachteil wird die automatische Ausführung eines Smart Contracts nach seiner erfolgreichen Implementierung gesehen. Im herrkömmlichen Recht ist ein Vertragspartner dagegen frei, jederzeit den Vertrag zu brechen. Zusätzlich wird wie in Abschnitt 4.2.3 *Smart Contracts - Risiken* bereits beschrieben ein eventuelles Problem darin bestehen, angemessene Schutzmechanismen für Verbraucher in einem Smart Contract zu implementieren.

Darüber hinaus wird kritisiert, dass Smart Contracts, abgesehen von der Bezahlung illegaler Güter und Dienstleistungen mit Kryptowährungen, in Zukunft auch für weitere illegale Zwecke verwendet werden können, wie beispielseise die automatische Bezahlung bei der Veröffentlichung vertraulicher Informationen [3].

4.6. Internet of Things

Die Kernidee hinter dem Begriff *Internet of Things* (IoT) beschreibt eine Vernetzung von Alltagsgegenständen über das Internet. Dabei versuchen diese Gegenstände autonom nützliche Ziele zu erreichen. Um diese Ziele zu erreichen werden die Alltagsgegenstände mit Fähigkeiten zur Wahrnehmung, Erkennung und Verabeitung von Informationen sowie der Kommunikation mit anderen Objekten und Diensten ausgestattet.

Eine der größten Herausforderungen stellt dabei die Interoperabilität der unterschiedlichen Geräte dar. Es existiert noch kein geeignetes Architekturmodell, um effektive Konnektivität, Kontrolle, Kommunikation und Anwendungen für die heterogenen Geräte und Applikatonen zu unterstützen [3]. Die Blockchain könnte als Basis solch einer Architektur dienen, wobei durch ihre dezentrale Natur das entstehende Peer-to-Peer Netz gleichzeitig die Notwendigkeit einer Cloudarchitektur ersetzt.

4.6.1. Beispiel. IBM und Samsung erarbeiteten gemeinsam einen Proof-of-Concept namens ADEPT (*Autonomous decentralized Peer-to-Peer Telemetry*), im Rahmen dessen eine Waschmaschine autonom Handlungen durchgeführt hat [14]. In dem Konzept bildet die Blockchain die Grundlage des Systems. Dazu wurde aufgrund ihrer Fähigkeit, Smart Contracts und DAOs implementieren zu können, die Ethereum-Blockchain gewählt.

Die Waschmaschine konnte die Menge des vorhandenen Waschmittels registrieren und bei Bedarf mittels Smart Contracts ihr eigenes Waschmittel nachbestellen und selbst bezahlen. Im Falle eines Schadens konnte sie über die Blockchain ihren Garantiestatus überprüfen und einen geeigneten Handwerker bestellen, der abhängig vom Garantiestatus bezahlt wurde.

Darüber hinaus wurde im Rahmen des ADEPT-Prototypen ein Handelsplatz für Energieversorgung geschaffen. Beispielsweise konnte die Waschmaschine darüber mit dem Micro-Grid einer Gemeinde kommunizieren und im Gegenzug für Energie mittels eines Vertrages zwischen dem Besitzer und der Gemeinde eine bestimmte Anzahl an Waschgängen für Gemeindemitglieder anbieten [3].

4.6.2. Kritik. Ein kritischer Punkt eines vernetzten IoT-Systems mit einer Blockchain als Grundlage könnte die Leistungsfähigkeit der einzelnen Geräte sein. So kann es bei sehr kleinen, leistungsschwachen Alltagsgegenständen zu Verzögerungen kommen. Je nach Anwendungsfall, Einsatzgebiet und Ziel müssten minimale Gerätevoraussetzungen gefordert und überprüft werden.

5. Potentiale und Risiken der Blockchain

Durch die Veröffentlichung des ersten Whitepapers über das Bitcoin System [1] rückte die Blockchain mit zunehmender Zeit immer weiter in den Fokus, als eigenständiges, technologisches System identifiziert zu werden. Mit Bitcoin wurde erstmals ein alternativer, dezentralisierter Ansatz zu den bis dahin bekannten, zentralisierten Zahlungs- und Transaktionssystemen vorgestellt. Die wichtigen Eigenschaften dieses Systems liegen allerdings der Blockchain Architektur zugrunde.

Das dezentralisierte System bietet viele für den Finanzsektor positive Eigenschaften wie etwa eine hohe Sicherheit gegen Netzausfälle, eine sehr hohe Datenintegrität sowie Transparenz in der Transaktionshistorie. Auch kürzere Transaktionszeiten sowie direkte Transaktionen zwischen zwei Entitäten ohne die Notwendigkeit des Vorhandenseins eines Mittelsmannes sind möglich. Aufgrund dieser Eigenschaften entstanden neben Bitcoin relativ schnell weitere, alternative Kryptowährungen mit leichten Anpassungen der Blockchain Architektur[13].

Neben der Verwendung der Blockchain im Finanzsektor konnten bereits weitere Einsatzmöglichkeiten der Blockchain identifiziert werden. Dazu zählen bereits existierende Plattformen wie Ethereum mit ihrer Möglichkeit, Smart Contracts und dezentralisierte autonome Organisationen implementieren zu können. Weitere denkbare Anwendungsfelder befinden sich im Bereich des öffentlichen Sektors und im Rechtswesen. Zusätzlich bietet die Blockchain Architektur diverse Eigenschaften um als solide Basis für autonom agierende Alltagsgegenstände im Bereich der Internet of Things eingesetzt werden zu können.

Abschließend werden zusammengefasst die eindeutigen Potentiale und Risiken der Blockchain aufgelistet [3]:

5.1. Potentiale

- Detaillierte Zugangskontrolle
- Pseudonymität
- Hohe Datenintegrität
- Kein Vertrauen für Interaktionen notwendig
- Hohe Prozessintegrität
- Große Transparenz
- Kurze Dauer der Transaktionsabwicklung
- Programmierbarkeit der Transaktionen

13. Siehe beispielsweise Abschnitt 3.2 *Proof-of-Stake* für Peercoin oder Abschnitt 3.3 *Proof-of-Burn* für Slimcoin.

5.2. Risiken

- Hoher Energiekonsum durch Proof-of-Work[14]
- Geringe Skalierbarkeit[15]
- Mangelnde Interoperabilität der unterschiedlichen Blockchain Systeme
- Irreversibilität von Transaktionen
- Keine garantierte Anonymität
- Mögliche Attacken

6. Fazit

Durch den Einsatz einer Blockchain konnte erstmalig das *Double Spending Problem* gelöst werden. Die Kernidee liegt dabei nicht darin das Problem selbst, sondern die umgebenden Faktoren, welche überhaupt erst zu dem *Double Spending Problem* geführt habenn, zu verändern. Dabei geht es um die Ablösung der zentralisierten Instanz durch einen grundlegenden dezentralisierten Ansatz. Bei diesen Ansätzen herrschte bis zur Bekanntmachung von Bitcoin weiterhin das Problem des Finden eines verteilten Konsens[16]. Die Kombination aus Dezentralisierung und funktionierendem verteilten Konsens erlangt seit 2008 kontinuierlich mehr Aufmerksamkeit. Weltweit arbeiten Menschen an Abwandlungen der Blockchain wie sie bei Bitcoin bekannt wurde, und entwickeln dabei alternative Blockchain Systeme mit anderen diversen Vor- und Nachteilen für unterschiedliche Anwendungsfälle.

Eine offene Frage dabei ist jedoch, ob sich diese Lösungen auf lange Zeit gesehen für beständig erweisen und somit bisherige Systeme ersetzen bzw. zumindest als positive Ergänzung dienen oder ob sie nur von kurzer Dauer sind. Zusätzlich bleibt offen, ob sich weitere nutzvolle Anwendungsfelder für die Blockchain ergeben und auch die Frage, ob diese überhaupt nötig sein werden.

Auch falls die Blockchain in der Zukunft nicht die nötige Akzeptanz erreichen sollte um weitläufig eingesetzt zu werden, so herrscht durch die Bekanntmachung von Bitcoin jedenfalls eine Erkenntnis darüber, dass dezentralisierte Peer-to-Peer Systeme mit funktionierendem verteiltem Konsens ein sehr großes Potential bieten.

14. Gilt nicht für alle Blockchain Systeme, siehe Abschnitt 3 *Aktualisierung der Blockchain*.

15. Dieser Kritikpunkt bezieht sich auf die maximale Blockgröße im Bitcoin System von 1 Megabyte. Dadurch kann das System nur eine begrenzte Anzahl an Transaktionen durchführen [3].

16. Das Problem der byzantinischen Generäle.

Literatur

[1] **Nakamoto, Satoshi**: *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, URL: http://bitcoin.org/bitcoin.pdf (abgerufen am 11.11.2017)

[2] **blockchain.mit.edu**, URL: http://blockchain.mit.edu/how-blockchain-works (abgerufen am 25.11.2017)

[3] **Fraunhofer Institute for Applied Information Technology FIT**: *Blockchain: Grundlagen, Anwendung und Potenziale*, 2016

[4] **Tschorsch, Florian & Scheuermann, Björn**: *Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies*, 2015

[5] **Böhme, Rainer et al.**: *Can We Afford Integrity by Proof-of-Work? Scenarios Inspired by the Bitcoin Currency*, 2013

[6] **Narayanan, Arvind et al.**: *Bitcoin and Cryptocurrency Technologies*, 2016, URL: https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf (abgerufen am 15.12.2017)

[7] **P4Titan**: *Slimcoin - A Peer-to-Peer Crypto-Currency without Proof-of-Burn*, 17.05.2014

[8] **Proof-of-Burn**: *What is Proof of Burn (ELI5)?*, (o.J.), URL: http://slimco.in/proof-of-burn-eli5 (abgerufen am. 17 Dezember 2017).

[9] **Bentov, Iddo et al.**: *Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake*, 2014

[10] **Lee, Charles**: *Proof of Activity Proposal*, 21.08.2012, URL: https://bitcointalk.org/index.php?topic=102355.0 (abgerufen am 17.12.2017)

[11] **Datarella**: *Eine Dezentrale Autonome Organisation DAO - Was ist das?*, 03.05.2016, URL: https://datarella.de/dezentrale-autonome-organisation-dao-was-ist-das/ (abgerufen am 18.12.2017)

[12] **Forte, Pasquale et al.**: *Beyond Bitcoin - Part 1: A critical look at blockchain-based systems*, 01.12.2015

[13] **Duivestein, Sander et al.**: *Design to Disrupt - Blockchain: cryptoplatform for a frictionless economy*, 2015, URL: http://labs.sogeti.com/wp-content/uploads/2015/08/D2D-3_EN-web.pdf (abgerufen am 19.12.2017)

[14] **Panikkar, Sanjay et al.**: *ADEPT: An IoT Practitioner Perspective*, 2015, URL: https://de.scribd.com/doc/252917347/IBM-ADEPT-Practictioner-Perspective-Pre-Publication-Draft-7-Jan-2015 (abgerufen am 20.12.2017)

[15] **Kattwinkel, Oliver**: *Eine Einführung in das Themengebiet der Kryptowährungen*, Hochschule Bonn-Rhein-Sieg, 2018