h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt
- Fachbereich Informatik -

Permissioned Blockchains für B2B

Prototypische Implementierung eines dezentralisierten Wartungsmarktes

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt von

Eric Nagel
Matrikelnummer

Referent:	Prof. Dr. Andreas Müller
Korreferent:	Björn Bär

Abstract

Traditionelle B2B-Anwendungen mit multiplen Geschäftspartnern als Teilnehmer bringen verschiedene Probleme mit sich. Wenn jedes Unternehmen seine eigenen Daten speichert, erfolgt der Zugriff auf diese, für Kooperationspartner, aufwändig über Schnittstellen. Die Daten könnten sich auch bei einer einzelnen, eventuell nicht vertrauenswürdigen Instanz befinden, welche die Kontrolle über diese hat. Um dieses Problem zu lösen wird eine dezentrale B2B-Anwendung mittels der Blockchain-Technologie entwickelt. Die bekanntesten Implementationen dieser, wie Bitcoin und Ethereum, bringen jedoch Nachteile hinsichtlich Datenschutz, Sicherheit und Transaktionsdurchsatz mit sich, welche im B2B-Bereich nicht wünschenswert sind. Diese werden analysiert, um anschließend eine Aussage über den Nutzen von B2B-Blockchain-Anwendungen zu treffen, und sinnvoll den dezentralen Wartungsmarkt zu entwickeln.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Kapitel 1

Einführung und Motivation

Klassische B2B-Anwendungen bringen diverse Probleme hinsichtlich der Datenhaltung mit sich. Eigene Daten können bei jeden Geschäftspartner selber gespeichert werden, was jedoch den Zugriff auf diese, aufgrund von aufwendig einzurichtenden Schnittstellen und uneinheitlichen Datenformaten, erschwert. Eine andere Möglichkeit ist die Speicherung bei einem zentralen Unternehmen. Dieses hätte jedoch die Kontrolle über die Daten,womit alle anderen Parteien diesem vertrauen müssten. Diese Faktoren machen B2B-Anwendungen für die Teilnehmer unattraktiv und erschweren die Entwicklung [21], [**?**].

Um diese Probleme zu lösen, wird ein Prototyp einer dezentralen B2B-Applikation basierend auf der Blockchain-Technologie entwickelt. Sie erlaubt es dezentrale Systeme aufzubauen, in welchen sich die Parteien nicht vertrauen. Alle Daten würden bei jedem Teilnehmer des Netzwerks gespeichert werden. Trotzdem sind diese nicht lösch- oder manipulierbar, alle Transaktionen sind lückenlos nachvollziehbar und es besteht ein gemeinsamer Konsens über den Datenbestand [17]. Bei der zu entwickelnden Applikation handelt es sich um einen automatisieren und dezentralisierten Wartungsmarkt. Teilnehmer an diesem sind Unternehmen und Wartungsdienstleister. Die Unternehmen besitzen IoT-Geräte, welche automatisch erkennen, dass sie eine Wartung benötigen. Sie legen für die Wartung einen Smart-Contract an, welcher von Wartungsdienstleistern angenommen werden kann. Diese melden sich an dem Gerät an und loggen die durchgeführten Wartungsschritte. Die Maschine schließt nach durchgeführter Wartung den Vertrag. Somit besteht ein automatisierter Wartungsmarkt zwischen mehreren Unternehmen, in welchen Wartungen verfolgbar und unveränderbar dokumentiert werden sowie kein Vertrauen zwischen den Parteien nötig ist.

Bekannte Blockchain-Implementationen, wie Bitcoin oder Ethereum, bringen jedoch Probleme mit sich, welche im B2B-Bereich von Nachteil sind. So sind alle Daten öffentlich einsehbar, der Transaktionsdurchsatz ist gering und die Konsensmechaniken sind unter bestimmten Umständen unsicher und resultieren in hohen Energieverbrauch [13][25][10].

Ziel dieser Arbeit ist es, die Probleme der Blockchain-Technologie für den B2B-Bereich zu analysieren und basierend auf den Ergebnissen die Entwicklung einer dezentralen B2B-Anwendung zu beschreiben sowie zu evaluieren. Dazu werden zunächst die grundlegenden Kon-

zepte der Blockchain-Technologie erklärt, um ein besseres Verständnis für die Vor- und Nachteile dieser zu erhalten. Anschließend werden die Probleme für B2B-Anwendungen anhand der Anforderungen an dem Wartungsmarkt genauer betrachtet und analysiert. Daraufhin erfolgt die Beschreibung der Anwendungsentwicklung. Zuletzt wird ein Fazit zur Lösung der Probleme und des entwickelten Systems gezogen.

Kapitel 2

Blockchain-Grundlagen

2.1 Funktionsweise

Die Funktionsweise der Blockchain wird hauptsächlich an Bitcoin erklärt. Als erste Blockchain-Anwendung [32] und aufgrund der relativ geringen Komplexität liefert es die Grundlage für die Funktion der Technologie. Andere Implementationen, wie Ethereum oder Ripple, funktionieren nach dem gleichen Prinzip.

2.1.1 Allgemein

Wenn der Begriff "Die Blockchain" auftaucht, ist damit meistens die Blockchain-Technologie gemeint. Es gibt nicht nur eine global bestehende Blockchain und auch nicht nur eine Implementation der Technologie, was man an Bitcoin oder Ethereum sehen kann.

Allgemein kann man die Blockchain als Datenstruktur bezeichnen, welche verteilt, nicht löschbar und unmanipulierbar gespeichert werden kann. Weiterhin verifizieren jegliche Teilnehmer am Netzwerk ausgeführte Transaktionen, womit ein gemeinsamer Konsens über den Datenbestand besteht [17].

In einer Blockchain werden Transaktionen in Blöcken gespeichert. Dabei handelt es sich um Operationen, welche Daten erstellen, verändern, oder löschen. Aus diesen lässt sich letztendlich der aktuelle Datenbestand ermitteln. So erfolgt z.B. bei Bitcoin keine Speicherung des aktuellen Guthabens der Teilnehmer. Es wird nur aus allen bestehenden Transaktionen berechnet [13]. Die Daten welche letztendlich bestehen, können z.B. Geldtransferinformationen (Bitcoin), Smart Contracts (Ethereum, selbstausführende Verträge mit Programmlogik, siehe 2.2), oder simple Dokumente oder Informationen sein [10][25][30]. Die Blöcke setzen sich zusammen aus den Transaktionen sowie den Block Header, welcher verschiedene Metadaten, wie zum Beispiel den kombinierten Hash[1] aller Transaktionen, enthält [13].

Die Blöcke sind miteinander verkettet. Jeder Block Header enthält den Hash des vorherigen

[1] Hash: Ergebnis einer Operation, welche "eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit fester Länge abbildet" [11].

Block Headers (Siehe Abb. 2.1). Dies ist ein wichtiges Feature zum Schutz der Blockchain vor Angriffen. Wenn ein Angreifer die Transaktionen eines Blocks zu seinen Gunsten verändern würde, würde sich der Hash des Block Headers ändern. Dieser müsste dann im darauffolgenden Block Header stehen, womit sich allerdings auch der Hash dieses Blocks ändert. Letztendlich müssten alle folgenden Blöcke manipuliert werden, um eine gültige Blockchain zu erhalten [25]. Diese Manipulation wird durch verschiedene Verfahren erschwert, welche genauer in den Kapiteln 2.1.1 und 2.1.2 erklärt werden.

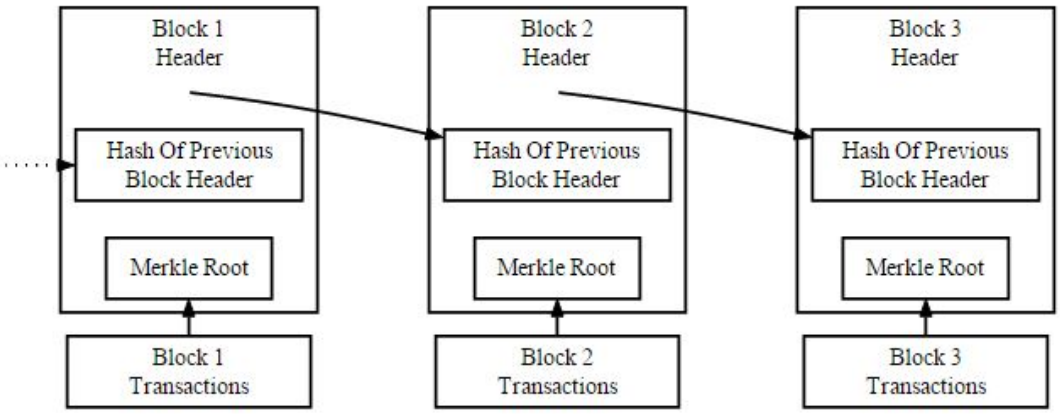

Abbildung 2.1: Verkettung von Blöcken durch Block Header Hashes

Die Blockchain ist verteilt gespeichert. Jeder Teilnehmer hat die Möglichkeit Sie auf seinen Rechner zu speichern. Somit besteht keine zentrale Instanz, welche die Kontrolle über die Daten hat. Weiterhin gibt es keinen Single Point of Failure[2], [17].

2.1.2 Konsensmechanismen

Aufgrund der verteilten Datenhaltung, muss es Verfahren geben, um die Daten synchron, und auf einen Stand, auf welchen sich alle Teilnehmer geeinigt haben, zu halten. Dazu gibt es die sogenannten Konsensmechanismen, welche gleichzeitig die Unmanipulierbarkeit der Daten sicherstellen. Bevor diese erklärt werden können, muss zunächst genauer auf die Funktion des Netzwerks eingegangen werden.

Wenn ein Teilnehmer eine Transaktion ausführt, wird diese, vorausgesetzt dass sie valide ist (Genauer im nächsten Absatz erklärt), an alle Nodes (Teilnehmer, welche die Blockchain speichern) im Netzwerk weitergeleitet und im Transaktionspool aufgenommen. Dieser enthält alle noch nicht in Blöcken vorkommenden Transaktionen. Diese werden in einen neuen Block aufgenommen, und jede Node beginnt mit der Erstellung von diesem. Das Erstellen wird durch verschiedene Konsensmechaniken realisiert. Bei Bitcoin und Ethereum findet der Proof-of-Work Anwendung (Genauer im folgenden Absatz erklärt). Sobald eine Node einen Block erstellt, wird

[2]Single Point of Failure: Komponente eines Systems, dessen Ausfall den Ausfall des gesamten Systems bewirkt [9].

dieser im Netzwerk verteilt. Jede Node hängt ihn an ihre lokale Blockchain an, und beginnt mit der Erstellung des nächsten Blocks [**?**].

Damit eine Transaktion valide ist, muss sie bestimmte Voraussetzungen erfüllen. So muss sie unter anderen mit den Private Key des Senders signiert sein. Mittels seines Public Keys kann überprüft werden, ob wirklich er der Sender der Nachricht ist und ob die Transaktion manipuliert wurde. Dieses Verfahren wird auch in der Abbildung 2.2 visualisiert. Das Signieren trägt zur Sicherheit der Blockchain bei, da ein Angreifer somit keine Transaktionen manipulieren oder im Namen eines anderen ausführen kann. In Bitcoin ist eine weitere Kondition, dass der Transaktionsersteller die zu sendenen Bitcoins besitzt [13]. In Systemen wie Ethereum und Hyperledger Fabric, in welchen eigene Programmlogik abgebildet werden kann, können weitere Konditionen festgelegt werden. So muss z.B. ein Teilnehmer die nötigen Rechte haben um eine Transaktion auszuführen [1].

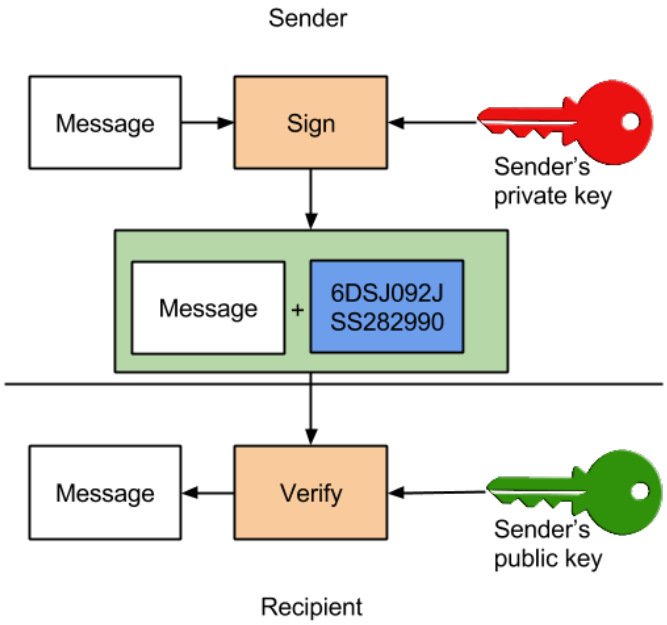

Abbildung 2.2: Signieren und Verifizieren von Nachrichten. Der Sender signiert die Nachricht mit seinen Private Key und der Empfänger kann diese mit den Public Key des Senders verifizieren.

Der Proof-of-Work ist nur einer der zur Verfügung stehenden Konsensmechanismen (Siehe Kapitel 5.1). Er bedarf jedoch genauerer Erklärung, da er in aktuellen Blockchain-Implementationen vorwiegend genutzt wird. Der Proof-of-Work ist eine Art Rätsel, welches mit der Rechenleistung von Nodes gelöst werden muss, um einen Block zu erschaffen. Genauer gesagt, muss für einen Block ein Hash gefunden werden, welcher einen bestimmten Wert unterschreitet. Desto kleiner dieser ist, desto höher ist die Schwierigkeit. Um wachsender Rechenleistung und Teilnehmerzahl entgegen zu wirken, also die Zeit für die Erstellung eines Blockes ungefähr gleich

zu halten, kann die Schwierigkeit angepasst werden. Dies ist aufgrund verschiedener Faktoren nötig, welche genauer im Kapitel **??** erläutert werden. Um unterschiedliche Hashwerte für gleiche Blöcke zu erhalten, gibt es im Block Header eine Nonce[3], welche verändert wird [25]. Alle Nodes im Bitcoin-Netzwerk benötigen im Durchschnitt 10 Minuten um einen Proof-of-Work zu erbringen [13], bei einer Hash Rate[4] von ca. 13.000.000 TH/s (Terrahashes pro Sekunde) [8]. Bei Ethereum beträgt die Zeit ungefähr 14 Sekunden [5], bei einer Hash Rate von ca. 150 TH/s [6]. Damit die Nodes eine Motivation haben, Rechenleistung für das Erstellen von Blöcken zu nutzen, erhalten sie bei Erbringung des Proof-of-Work eine Belohnung in Form von Währung [25] [10].

Um vollständig zu verstehen, wie der Proof-of-Work funktioniert, muss das Forking erklärt werden. Wenn eine Node einen Proof-of-Work erbringt, also einen Block erstellt, wird dieser an alle anderen Nodes weitergeleitet. Im Bitcoin-Netzwerk dauert es bei einer maximalen Blockgröße von 1MB [13], zwischen 6 und 20 Sekunden, bis ein Block mindestens 90% aller Nodes erreicht hat [**?**]. In dieser Zeit kann es vorkommen, dass eine weitere Node einen Block erstellt. Auch dieser wird im Netzwerk verteilt, womit 2 Versionen der Blockchain existieren: Eine endet mit Block A, und die andere mit Block B. Dies ist der sogenannte Fork. Das Netzwerk muss sich nun darauf einigen, welche der beiden Versionen beibehalten werden soll. Deshalb gilt: Die längere Blockchain ist die gültige. Die Nodes probieren an den zuerst erhaltenen Block (A oder B) einen neuen anzuhängen. Gelingt dies, ist eine der beiden Blockchains länger als die andere. Diese wird dann von allen Nodes als die richtige akzeptiert [13]. Dieser Vorgang wird auch in den Abbildungen 2.3 bis 2.6 dargestellt. Theoretisch ist es möglich, dass ein Fork über mehrere Blöcke bestehen muss. Die Wahrscheinlichkeit dafür ist jedoch gering, da mehrmals nacheinander mindestens 2 Nodes zur ungefähr gleichen Zeit einen Block erstellen müssen. Dies ist auch der Grund, warum Transaktionen erst als bestätigt gelten, sobald sie in einem Block stehen, welcher eine gewisse Anzahl an Nachfolgern hat. Denn erst dann ist die Sicherheit gegeben, dass die Transaktion nicht in einem Fork vorhanden ist, welcher eventuell verworfen wurde. Wie genau der Proof-of-Work das Netzwerk absichert, wird im Kapitel 2.1.2 erklärt.

Neben dem Proof-of-Work gibt es noch weitere Konsensmechanismen, wie Proof-of-Stake, Proof-of-Authority oder Practical Byzantine Fault Tolerance [27], [18] . Diese werden im Kapitel 5.1 genauer beschrieben und analysiert.

2.1.3 Nichtangreifbarkeit/Immutability

Viele Faktoren tragen zur Nichtangreifbarkeit und Unveränderlichkeit der Blockchain bei. Da alle Nodes die ausgeführten Transaktionen auf Validität prüfen, können diese nicht ohne Berechtigung, im Namen einer anderen Identität, oder mit unzureichenden Konditionen ausgeführt werden. Der wichtigste Faktor ist jedoch der genutzte Konsensmechanismus in Verbindung mit

[3]Nonce: Eine "Zahlen- oder Buchstabenkombination, [...] die nur ein einziges Mal in dem jeweiligen Kontext verwendet wird"[12].

[4]Hash Rate: Anzahl der in einer Zeiteinheit berechneten Hashwerte [7].

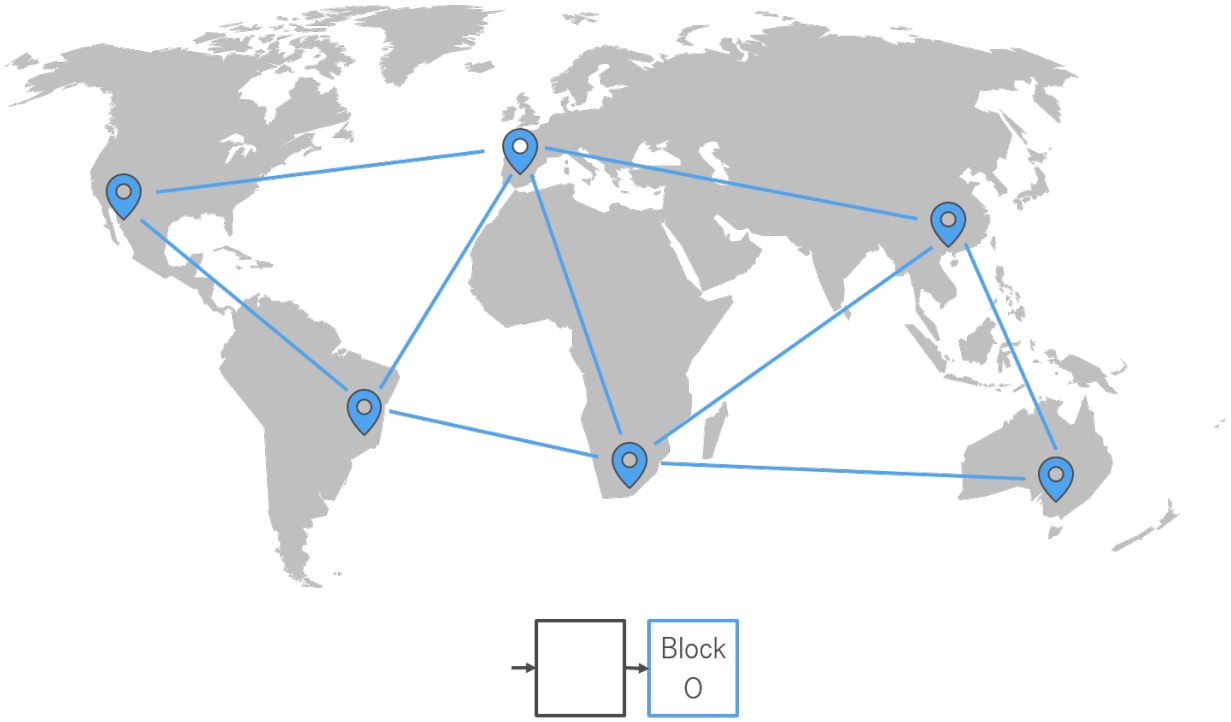

Abbildung 2.3: Fork-Visualisierung - Vor dem Fork besitzen alle Nodes Block O als letzten Block [13].

den verketteten Blöcken. Durch ihm wird sichergestellt, dass bestehende Daten nicht gelöscht oder manipuliert werden können.

Ein Beispiel dafür kann am Proof-of-Work gezeigt werden. Ein Angreifer probiert eine Transaktion aus einen bestehenden Block zu entfernen. Dazu würde er die Transaktion bei seiner lokalen Blockchain entfernen. Nun ist jedoch der Hash des Blockes sowie der Block selber nicht valide und würde von keiner Node akzeptiert werden. Der Angreifer muss also erneut einen Proof-of-Work für den manipulierten Block erbringen. Dies wäre für eine Einzelperson jedoch Zeitaufwändig, wenn man bedenkt, das extra für diesen Zweck produzierte Hardware eine Hash Rate von bis zu 13,5 TH/s erreicht [24]. Dies Wenn der manipulierte Block nun noch Nachfolger hat, muss aufgrund des neuen Hashes auch für diese der Proof-of-Work erbracht werden. Hinzu kommt, dass die Blockchain des Angreifers erst von allen Nodes akzeptiert wird, wenn sie länger ist. Er müsste also schneller als das gesamte Bitcoin-Netzwerk Blöcke erschaffen können. Dies ist nur möglich, wenn er 51% der Rechenleistung des Netzwerks besitzt. Deshalb wird dieser Angriff auch 51%-Angriff genannt [28] [10].

An dieser Stelle sollte erähnt werden, dass auch wenn ein 51%-Angriff erfolgt, die Angriffsmöglichkeiten beschränkt sind. Der Angreifer kann keine unvaliden Transaktionen sowie Blöcke erstellen. Ihm ist es möglich DoS-Angriffe auszuführen indem er verhindert das bestimmte Transaktionen in Blöcke aufgenommen werden. Genau so kanner die Historie der Daten verändern, indem er eine Transaktion aus einem Block entfernt und diese nicht erneut in einen Block aufnimmt. Im Falle von Kryptowährungen gibt es außerdem die sogenannte Double-Spending-Attacke. Diese funktioniert folgendermaßen: Ein Angreifer sendet z.B. Bitcoins an

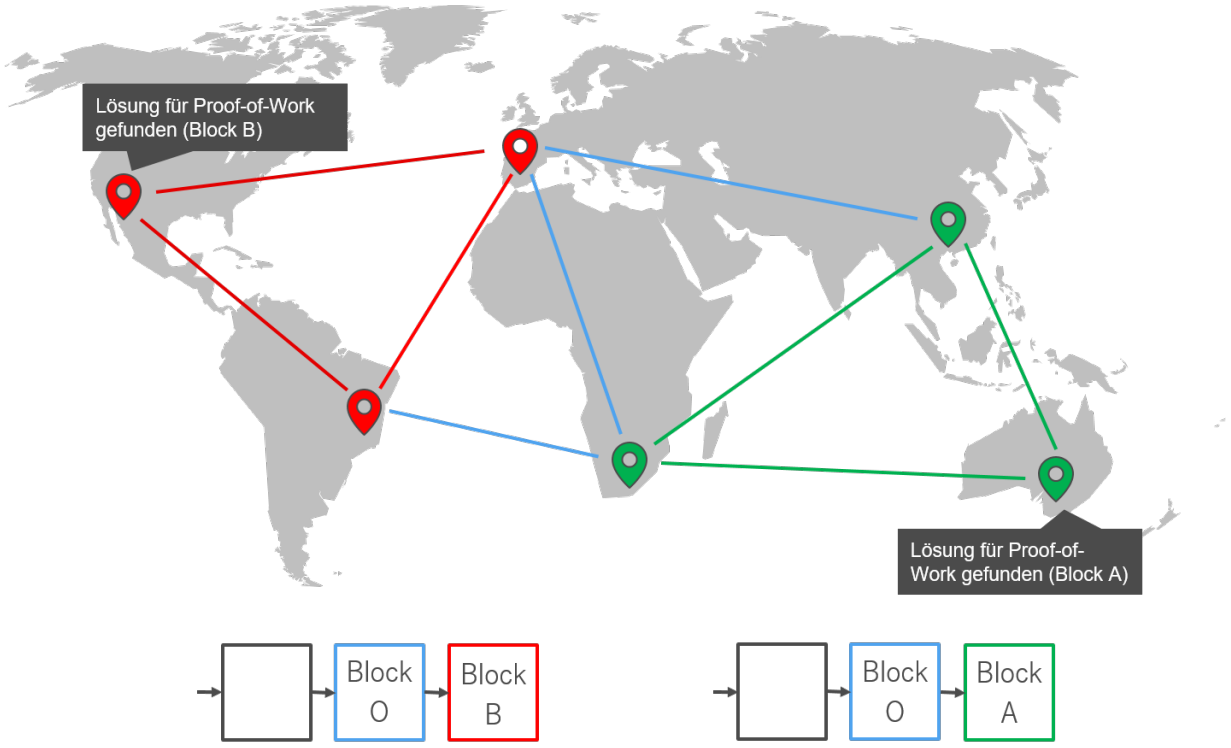

Abbildung 2.4: Fork-Visualisierung - 2 Nodes finden zur ungefähr gleichen Zeit einen Block und verbreiten ihn im Netzwerk, womit 2 Versionen der Blockchain bestehen [13].

einen Händler. Dieser wartet auf die Bestätigung der Transaktion in einen Block sowie auf nachfolgende Blöcke. So stellt er sicher, dass die Transaktion nicht in einem eventuell verworfenen Fork stand. Erst dann versendet er die Ware. Anschließend ersetzt der Angreifer die Transaktion durch eine Zahlung an sich selber und erstellt die längere Blockchain, womit der Händler letztendlich kein Geld erhalten hat [10]. Auch zu bedenken ist, dass ein Nutzer mit 51% der Rechenleistung wenig Motivation hat Angriffe auszuführen, da er für jeden erstellten Block Kryptowährung als Belohnung erhält. Der Wert der Kryptowährung würde sinken, wenn Angriffe auf die Blockchain entdeckt werden. Deshalb besteht für die sogenannten Miner eine Motivation, ehrlich zu arbeiten [13].

2.2 Blockchaintypen

Es gibt 3 Typen von Blockchains, welche die zugelassenen Teilnehmer bestimmen. Bisher wurden nur Public Blockchain-Anwendungen, wie Bitcoin und Ethereum erwähnt. In diesen gibt es keine Teilnehmerbeschränkungen, jeder kann am Netzwerk teilnehmen und die Blockchain öffentlich einsehen. Anders ist dies bei Permissioned und Private Blockchains. Die beiden Begriffe werden in einigen wissenschaftlichen Arbeiten gleichgesetzt (Siehe [20], [**?**], [23]). Hier folgt jedoch eine Unterscheidung. Dabei ist eine Private Blockchain, eine Blockchain welche nur von einen Nutzer verwendet wird. Da eine solche Anwendung keinen Sinn macht, da keine Vorteile der Blockchain genutzt werden können, wird darauf nicht genauer eingegangen. Interessanter

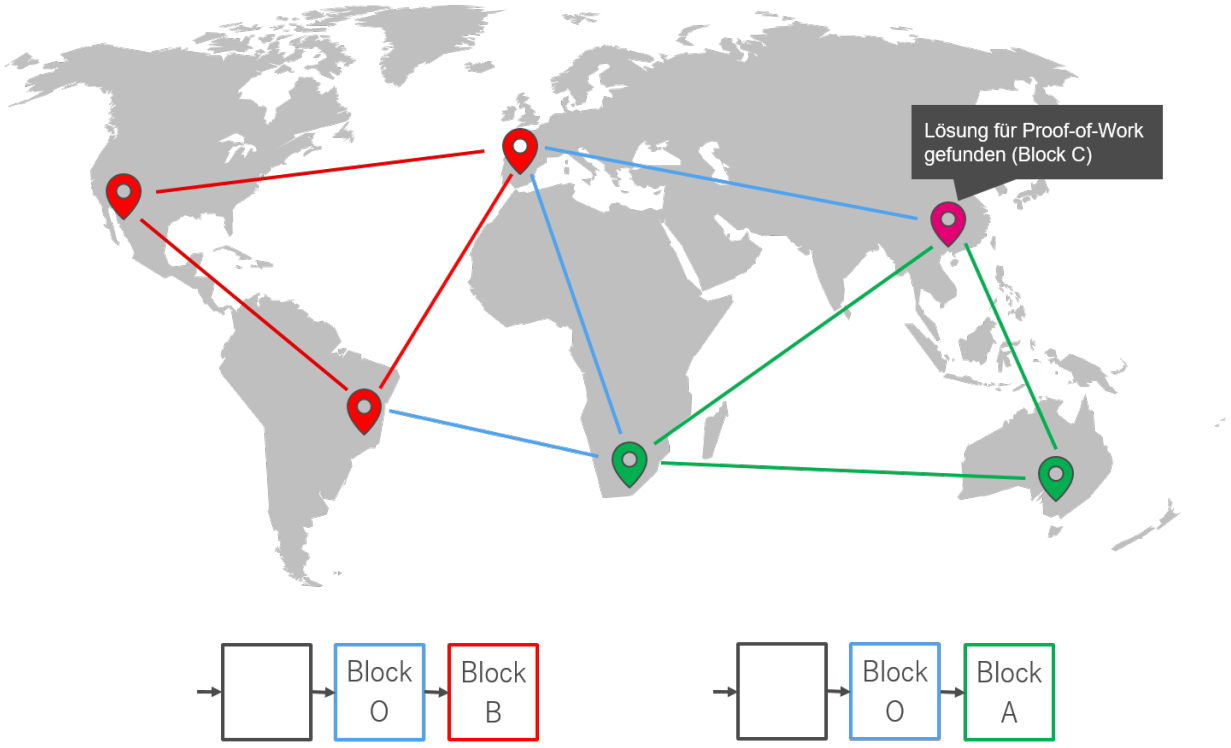

Abbildung 2.5: Fork-Visualisierung - Eine Node, welche Block A zuerst erhalten hat, hängt daran einen neuen Block C an [13].

sind Permissioned Blockchains, an welchen nur zugelassene Nutzer teilnehmen dürfen. Nur diese sind berechtigt, Transaktionen auszuführen und die Daten einzusehen [23]. Dies bietet sich vor allem bei B2B-Anwendungen an, welche von verschiedenen Unternehmen genutzt werden sollen. In diesen kann es aufgrund von z.B. sensiblen Daten nötig sein, das nur bestimmte Parteien Zugriff auf die Blockchain haben. An dieser Stelle sollte auch erwähnt werden, dass es möglich ist Blockchain-Implementationen wie z.B. Ethereum als Permissioned Blockchain zu nutzen [26].

2.3 Exemplarische Anwendungsfälle

Die Blockchain wird als revolutionäre Technologie angepriesen (Siehe [29]). Trotzdem ist es wichtig zu wissen, für welche Zwecke sie wirklich geeignet ist. Grundsätzlich macht eine Blockchain Sinn, wenn mehrere Parteien, welche sich nicht vertrauen, mit einen System interagieren wollen, welches von keiner dritten zentralen Instanz verwaltet wird [31]. Um eine bessere Vorstellung zu solchen Anwendungen zu erhalten, werden im Folgenden verschiedene Exemplarische Anwendungsfälle genannt und beschrieben.

Der erste Anwendungsfall, mit welchen die Blockchain-Technologie auch entstanden ist, sind Kryptowährungen. Mit Ihnen ist es möglich Geld zwischen beliebigen Parteien zu übertragen, ohne dass die Transaktionen von einer eventuell nicht vertrauenswürdigen Bank oder ähnlichem kontrolliert und verwaltet werden [28].

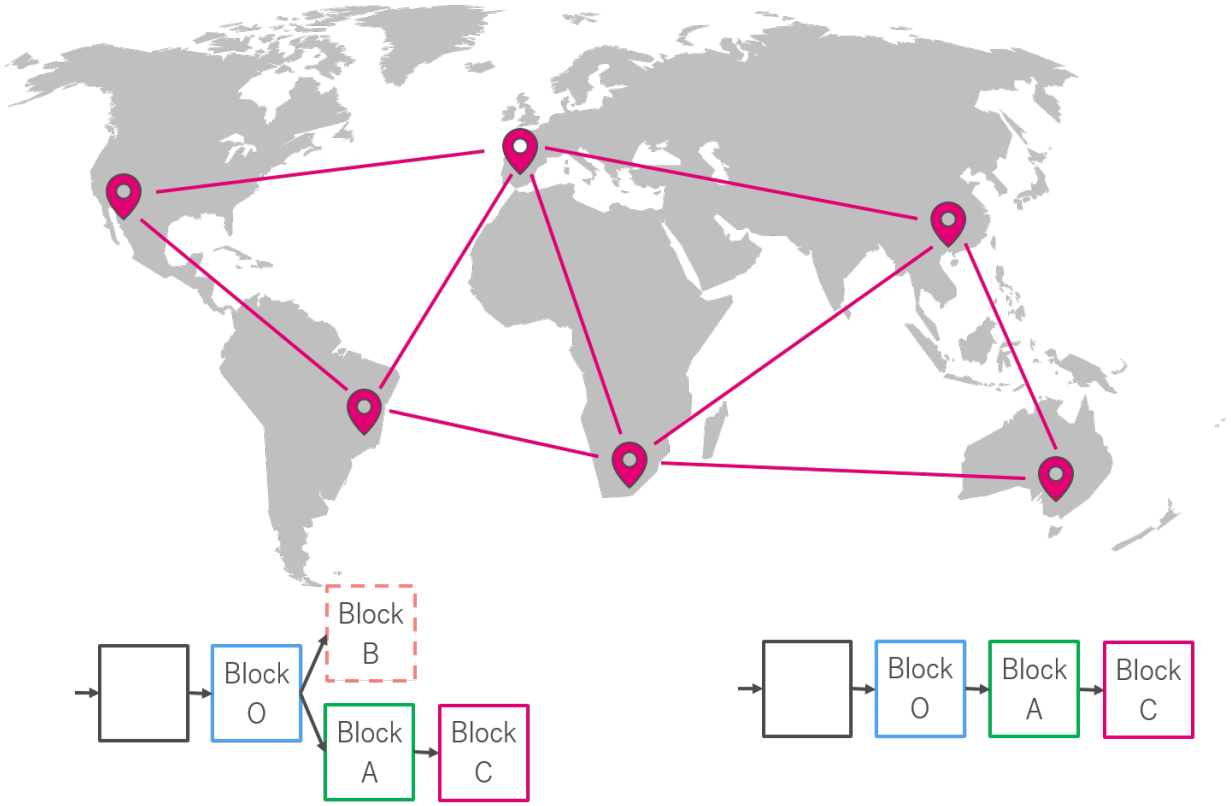

Abbildung 2.6: Fork-Visualisierung - Block C verbreitet sich im Netzwerk, rote Nodes sehen zwei Blockchains und akzeptieren die längere [13].

Weitere Anwendungsfälle ergeben sich mit der Möglichkeit Programmlogik auf der Blockchain abzubilden. So können beispielsweise dezentrale Online-Wahlen realisiert werden. Die Stimmen würden in der Blockchain gesammelt werden, und können so letztendlich nicht mehr von z.B. einer korrupten Regierung manipuliert werden [16].

Ein weiterer Anwendungsfall, insbesondere für den B2B-Bereich, wäre Supply Chain Management. Über eine digitale Lieferkette sollen Material- und Informationsflüsse zu Produkten und Dienstleistungen aufgebaut und verwaltet werden [22]. Dies erlaubt Unternehmen das automatisieren von Prozessen und das verbesserte reagieren auf Ereignisse (z.B. Lieferverspätungen). Weiterhin ist es dem Unternehmen möglich, dem Kunden exakt aufzuzeigen wo es und seine Unterprodukte produziert wurden. In klassischen B2B-Anwendungen müsste jedes Unternehmen, welches ein Teil der Supply Chain ist, die relevanten Daten durch z.B. Programmierschnittstellen bereitstellen. Diese müssten dann von allen Schnittstellen abgefragt werden um die Supply Chain zu erstellen. Aufgrund dieses Aufwandes werden oft Dritte eingestellt, welche sich um den Aufbau und um die Datenintegration der Supply Chain kümmern. Den Aufwand, sowie die eventuell nicht vertrauenswürdige dritte Partei könnte man durch die Nutzung der Blockchain überspringen. In dieser könnte jedes Unternehmen die relevanten Daten speichern, ohne das aufwändige Erstellen von Schnittstellen. Die Supply Chain wäre direkt in der Blockchain vorhanden, und kein Unternehmen muss Datenmanipulation oder ähnliches befürchten [21].

Auch dezentrale Märkte sind für B2B-Anwendungen interessant. Der zentrale Marktplatt-

formbetreiber, wie z.B. Ebay, welcher persönliche Informationen speichert und Gebühren für den Verkauf von Artikeln verlangt, wäre hinfällig. Nutzer könnten Waren untereinander verkaufen, während die Blockchain als Notar für den Warenaustausch dient [14].

Ein weiteres Beispiel wären Blockchain Sharing-Systeme. So könnte ein dezentrales Fahrradleihsystem aufgebaut werden. Nutzer würden mit ihrem Smartphone, über ihre in der Blockchain hinterlegte Identität (Im Falle von z.B. Ethereum die Wallet-Adresse[5]), das Fahrrad entsperren. Dieser erkennt automatisch die gefahrene Distanz sowie die Nutzungsdauer. Über einen Smart Contract würde anschließend die automatische Zahlung erfolgen. Neben der Automatisierung besteht der Vorteil, dass mehrere Unternehmen oder auch Privatpersonen Leihfahrräder anbieten können, ohne dass sie einer zentralen Instanz mit der Verwaltung vertrauen müssen [3], [19].

[5]Wallet: Speichert z.B. bei Ethereum und Bitcoin den Private Key des Nutzers und wird z.B. als Adresse für Zahlungen genutzt [4].

Kapitel 3

Dezentraler Wartungsmarkt - Konzept

3.1 Allgemein

Ziel dieser Arbeit ist die Entwicklung einer prototypischen B2B-Applikation in Form eines automatisierten sowie dezentralisierten Wartungsmarktes. Teilnehmer an diesem sind multiple Unternehmen und Wartungsanbieter. Erstere besitzen IoT-Geräte, welche erkennen können, dass sie eine Wartung benötigen. Die Wartungsanbieter erhalten die Informationen zur Wartung, und können sich für diese anmelden. Anschließend würden sie diese durchführen und dabei die Wartungsschritte loggen.

In klassischen B2B-Anwendungen wäre die Realisierung dieses Systems auf eine von 2 Arten erfolgt. Bei ersterer gäbe es eine dritte Partei, welche den Markt verwaltet, und bei welcher sich alle Unternehmen und Wartungsanbieter anmelden müssen (z.B. Ebay). Die andere Möglichkeit wäre, dass eines der teilnehmenden Unternehmen den Markt verwaltet. Bei beiden Optionen müssten die Teilnehmer am Markt ihre Daten einer eventuell nicht vertrauenswürdigen zentralen Instanz zur Verfügung stellen.

Um dies zu verhindern, wird der Wartungsmarkt auf Basis der Blockchain-Technologie implementiert. Somit können beliebig viele Unternehmen und Wartungsanbieter an dem System teilnehmen, ohne dass eine Datenmanipulation durch die Teilnehmer oder eine zentrale Instanz befürchtet werden muss.

3.2 Anforderungen

Es ergeben sich verschiedene Anforderungen an das zu entwickelnde System. Die Spezifizierung dieser ist wichtig, denn auf Basis von diesen wird eine Blockchain-Implementation ausgewählt und auf verschiedene Probleme analysiert. Dieser werden hier zunächst einmal aufgelistet um einen Überblick zu erhalten.

Es ergeben sich verschiedene funktionale Anforderungen:

- Registrieren und Identifizieren von Unternehmen, Wartungsanbietern und Geräten in der Blockchain

- Wartungsgeräte kündigen Wartungen in Form eines Smart Contracts in der Blockchain an
- Wartungsanbieter können den Smart Contract unter bestimmten Konditionen annehmen
- Wartungsanbieter loggen Wartungsschritte in der Blockchain
- Gerät überprüft ob die Wartung erfolgt ist, und schließt den Contract
- Nur bestimmte Teilnehmer können bestimmte Transaktionen ausführen
- Private Transaktionen sollen zwischen Teilnehmern möglich sein

Folgende nicht-funktionale Anforderungen existieren:

- Hoher Transaktionsdurchsatz und geringe Transaktionszeiten
- Nichtangreifbarkeit der Daten

Einige Anforderungen sollten genauer erklärt werden:

> "Wartungsgeräte kündigen Wartungen in Form eines Smart Contracts in der Blockchain an"

Es kann verschiedene Gründe für die Wartung geben. So kann z.B. ein Wartungsdatum erreicht werden, oder Sensorwerte weisen auf einen Fehler hin.

> "Wartungsanbieter können den Smart Contract unter bestimmten Konditionen annehmen"

Eine dieser Konditionen könnte sein, dass der Wartungsanbieter bereits Erfahrungen mit der Wartung von bestimmten Geräten hat. Diese Information könnte ebenfalls aus der Blockchain abgefragt werden. Weiterhin darf der Vertrag z.B. noch nicht von einen anderen Anbieter akzeptiert worden sein.

> "Gerät überprüft ob Wartung erfolgt ist, und schließt den Contract"

Die Überprüfung kann anhand der geloggten Schritte sowie Sensorwerten erfolgen, welche vor und nach der Wartung existiert haben.

> "Nur bestimmte Teilnehmer können bestimmte Transaktionen ausführen"

Die verschiedenen Teilnehmer haben unterschiedliche Rechte. So soll es z.B. einen Unternehmen nicht möglich sein, Wartungsverträge zu bearbeiten oder zu akzeptieren.

> "Private Transaktionen sollen zwischen Teilnehmern möglich sein"

Bei Blockchains wie Bitcoin und Ethereum sind alle Daten in der Blockchain für alle Teilnehmer einsichtbar. Aufgrund von sensiblen Daten kann es allerdings vorkommen, das nicht alle Transaktionen für alle Teilnehmer sichtbar sein sollen. Im Falle des Wartungsmarktes sollen z.B. Preisabsprachen zwischen Unternehmen und Wartungsdienstleistern privat erfolgen.

"Hoher Transaktionsdurchsatz und geringe Transaktionszeiten"

In Bitcoin ist lediglich ein Transaktionsdurchsatz von 7 Transaktionen pro Sekunde möglich [32]. Hinzu kommt, dass es ca. zwischen 30 Minuten und 16 Stunden dauern kann, bis eine Transaktion bestätigt ist [15]. Darauf wird auch genauer im Kapitel 5.1 eingegangen. In dem zu entwickelnden System ist die Skalierbarkeit wichtig. Je nach der Anzahl der am Netzwerk teilnehmenden Unternehmen und Wartungsanbieter wird eine höherer Transaktionsdurchsatz benötigt. Insbesondere wenn tausende von Maschinen z.B. ihren Status in der Blockchain loggen.

"Nichtangreifbarkeit der Daten"

Die Nichtangreifbarkeit wird durch die genutzte Konsensmechanik realisiert. Der am häufigsten genutzte Proof-of-Work ist in einem Netzwerk mit wenig Teilnehmern allerdings unsicher, da es einfach ist 51% der Rechenleistung zu erreichen. Weiterhin führt er zu einen hohen Stromverbrauch (Im Bitcoin-Netzwerk der Verbrauch von ca. 3.500.000 US-Haushalten [2]), welcher nicht erwünscht ist.

An dieser Stelle muss darauf hingewiesen werden, dass das System um viele nützliche Features erweiterbar ist. So könnte zum Beispiel eine Bewertung der Wartungsanbieter anhand bestimmter Faktoren erfolgen. Da es sich jedoch nur um eine prototypische Implementation handelt, werden nur die Features implementiert, welche für einen Proof-of-Concept eines solchen Systems benötigt werden.

Kapitel 4

Aktueller Stand der Technik/Related Work

4.1 Blockchain-Technologie

4.2 Dezentrale Anwendungen/Märkte

- Gibt es bereits andere dezentrale Märkte ?
- Bereits existierende Wartungsmärkte ?

Kapitel 5

Evaluierung Permissioned Blockchains für B2B

5.1 Skalierbarkeit

...Im Falle von Bitcoin gilt eine Transaktion als bestätigt, sobald Sie in einem Block vorkommt und 6 Nachfolger hat. Zum einen entsteht eine Wartezeit dadurch, dass 6 mal ein Proof-of-Work erbracht werden muss, und zum anderen priorisieren Miner die Transaktionen nach der Transaktionsgebühr. Desto höher diese ist, destro größer ist die ausgezahlte Belohnung [15].

- Transaktionsdurchsatz –> Vor allem wichtig bei IoT-Geräten
- Bestätigung von Transaktionen ?
- Datenmenge und Redundanz

5.2 Konsensmechanismen

- Nachteile Proof-of-Work
- Analyse anderer Konsensmechanismen

5.3 Datenschutz

- Müssen alle Daten öffentlich verfügbar sein ?
- Private Transaktionen realisieren

Kapitel 6

Dezentraler Wartungsmarkt - Prototyp

6.1 Technologieauswahl

6.2 Hyperledger Fabric und Composer

- Was sind die Besonderheiten an Hyperledger Fabric ?
- Was ist Hyperledger Composer ?

- Welche Blockchain wird genutzt um die Anforderungen zu erfüllen ?

6.3 Modell

- Architekturen, Sequenzdiagramme, Workflows etc.
- Datenmodell (Participants, Assets, Transaktionen)
- Netzwerk

6.4 Gerätesimulation durch Bosch XDK

- Simulation eines IOT-Geräts durch einen Bosch XDK

6.5 Programmlogik

- Funktion der Transaktionen

6.6 Benutzeroberflächen

- UIs für die Interaktion mit der Blockchain

6.7 Evaluierung

- Analyse des Systems in Bezug auf Anforderungen und Blockchain-Probleme

Kapitel 7

Fazit und Ausblick

- Kurze Zusammenfassung
- Ausblick geben/Erweiterbarkeit des Systems beschreiben
- Ausblick zu Problemen von B2B-Blockchains geben

Literaturverzeichnis

[1] Access Control Language | Hyperledger Composer. https://hyperledger.github.io/composer/unstable/reference/acl_language.

[2] Bitcoin Energy Consumption Index. https://digiconomist.net/bitcoin-energy-consumption.

[3] Blockchain Bikes. http://futurefluxfestival.nl/en/program/blockchain-bikes/.

[4] Blockchain Wallet. https://wirexapp.com/guides/mobile-wallets/blockchain-wallet/.

[5] Ethereum Average BlockTime Chart. https://etherscan.io/chart/blocktime.

[6] Ethereum Network HashRate Growth Chart. https://etherscan.io/chart/hashrate.

[7] Glossar - Bitcoin. https://bitcoin.org/de/glossar.

[8] Hash Rate. https://blockchain.info/hash-rate.

[9] Single Point of Failure. *Wikipedia*, July 2016. Page Version ID: 156306981.

[10] Ethereum White Paper, December 2017.

[11] Kryptologische Hashfunktion. *Wikipedia*, November 2017. Page Version ID: 170625494.

[12] Nonce. *Wikipedia*, August 2017. Page Version ID: 167799632.

[13] Andreas M. Antonopoulos. *Mastering Bitcoin*. O'Reilly, Sebastopol CA, first edition edition, 2015. OCLC: ocn876351095.

[14] Elyes Ben Hamida, Kei Leo Brousmiche, Hugo Levard, and Eric Thea. Blockchain for Enterprise: Overview, Opportunities and Challenges. In *The Thirteenth International Conference on Wireless and Mobile Communications (ICWMC 2017)*, Nice, France, July 2017.

[15] Steven Buchko. How Long Do Bitcoin Transactions Take? https://coincentral.com/how-long-do-bitcoin-transfers-take/, December 2017.

[16] Amy Castor. An Ethereum Voting Scheme That Doesn't Give Away Your Vote. https://www.coindesk.com/voting-scheme-ethereum-doesnt-give-away-vote/, April 2017.

[17] Michael Crosby. BlockChain Technology: Beyond Bitcoin. Technical report, 2016.

[18] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. 2017.

[19] Andreas Fischer. Das IoT in der Blockchain. https://www.com-magazin.de/praxis/internet-dinge/iot-in-blockchain-1228562.html.

[20] Vincent Gramoli. On the danger of private blockchains. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL'16)*, 2016.

[21] Kari Korpela, Jukka Hallikas, and Tomi Dahlberg. Digital Supply Chain Transformation toward Blockchain Integration. January 2017.

[22] Winfried Krieger. Definition » Supply Chain Management (SCM) « | Gabler Wirtschaftslexikon. http://wirtschaftslexikon.gabler.de/Definition/supply-chain-management-scm.html.

[23] Wenting Li, Alessandro Sforzin, Sergey Fedorov, and Ghassan O. Karame. Towards Scalable and Private Industrial Blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, pages 9–14, New York, NY, USA, 2017. ACM.

[24] What is Bitcoin Mining? Learn about Bitcoin mining hardware. https://www.bitcoinmining.com/bitcoin-mining-hardware/.

[25] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.

[26] Mercury Protocol. How To: Create Your Own Private Ethereum Blockchain, December 2017.

[27] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos. Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 253–255, September 2017.

[28] Melanie Swan. *Blockchain: Blueprint for a New Economy.* O'Reilly, Beijing : Sebastopol, CA, first edition edition, 2015. OCLC: ocn898924255.

[29] Don Tapscott and Alex Tapscott. *Die Blockchain-Revolution: Wie die Technologie hinter Bitcoin nicht nur das Finanzsystem, sondern die ganze Welt verändert.* Plassen Verlag, Kulmbach, 1 edition, October 2016.

[30] Hyperledger Fabric Team. Hyperledger Whitepaper. https://docs.google.com/document/d/1Z4M_qwILLRehPbVRUsJ3OF8Iir-gqS-ZYe7W-LE9gnE/edit?usp=embed_facebook, 2016.

[31] Karl Wüst and Arthur Gervais. Do you need a Blockchain? Technical Report 375, 2017.

[32] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, December 2017.