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 [14], [10].

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 [12]. 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 [11][15][7].

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 [19] 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 [12].

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 [11]. Die Daten welche letztendlich bestehen, können z.B. Geldtransferinformationen (Bitcoin), Smart Contracts (Ethereum, selbstausführende Verträge mit Programmlogik, siehe 2.1.4), oder simple Dokumente oder Informationen sein [7][15][18]. 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 [11].

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" [8].

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 [15]. Diese Manipulation wird durch verschiedene Verfahren erschwert, welche genauer in den Kapiteln 2.1.1 und 2.1.2 erklärt werden.

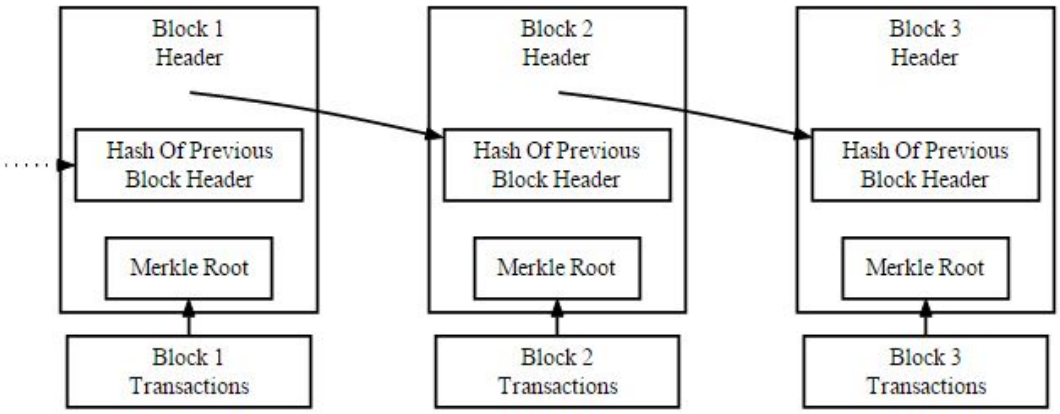

Abbildung 2.1: Verkettung von Blöcken durch Block Header Hashes

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], [12].

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. 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 dieser im Netzwerk

[2]Single Point of Failure: Komponente eines Systems, dessen Ausfall den Ausfall des gesamten Systems bewirkt [6].

verteilt. Jede Node hängt ihn an ihre lokale Blockchain an, und beginnt mit der Erstellung des nächsten Blocks [15][7].

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. Dies 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 [11]. 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].

Der Proof-of-Work ist nur einer der zur Verfügung stehenden Konsensmechanismen (Siehe Kapitel 5.1). Er bedarf jedoch genauerer Erklärung, da er in den zwei bekanntesten Blockchain-Anwendungen (Bitcoin und Ethereum) erfolgreich genutzt wird, und am meisten Erfolg verspricht (?). Der Proof-of-Work ist eine Art Rätsel, welches mit Rechenleistung 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 2.2 erläutert werden. Um unterschiedliche Hashwerte für gleiche Blöcke zu erhalten, gibt es im Block Header eine Nonce[3], welche verändert wird [15]. Alle Nodes im Bitcoin-Netzwerk benötigen im Durchschnitt 10 Minuten um einen Proof-of-Work zu erbringen [11], bei einer Hash Rate[4] von ca. 13.000.000 Terrahashes pro Sekunde [5]. Bei Ethereum beträgt die Zeit ungefähr 14 Sekunden [2], bei einer Hash Rate von ca. 150 TH/s [3].

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 [11], 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 [11]. Dieser Vorgang wird auch in den Abbildungen 2.2 bis 2.5 dargestellt. Das Forking ist auch der Grund, warum Transaktio-

[3]Nonce: Eine "Zahlen- oder Buchstabenkombination, [...] die nur ein einziges Mal in dem jeweiligen Kontext verwendet wird"[9].

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

nen erst als bestätigt selten, sobald sie in einem Block stehen, welcher eine gewisse Anzahl an Nachfolgern hat. Dies wird genauer im Kapitel 2.2 beschrieben. Wie genau der Proof-of-Work das Netzwerk absichert, wird im Kapitel 2.1.2 erklärt.

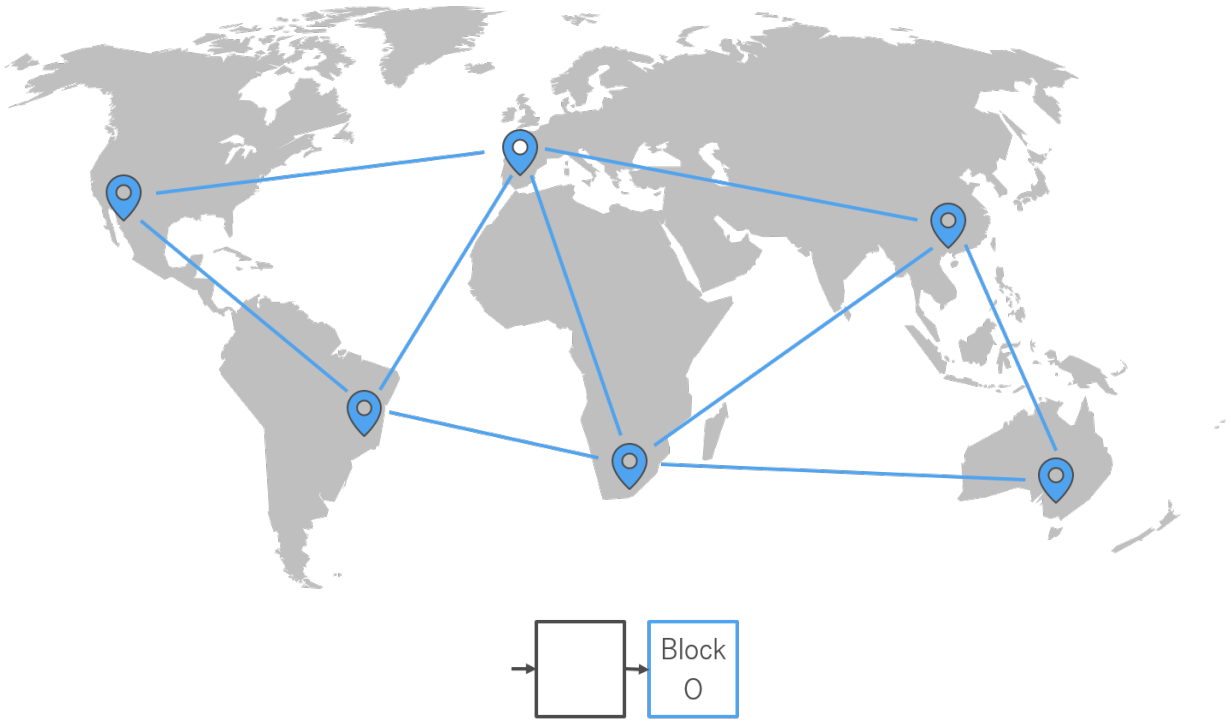

Abbildung 2.2: Fork-Visualisierung - Vor dem Fork besitzen alle Nodes Block O als letzten Block

Neben dem Proof-of-Work gibt es noch weitere Konsensmechanismen, wie Proof-of-Stake, Proof-of-Authority oder Practical Byzantine Fault Tolerance [16], [13] . 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 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. Wenn man die Hashrate eines modernen PC mit der des Bitcoin-Netzwerks vergleicht, welches 10 Minuten braucht um einen gültigen

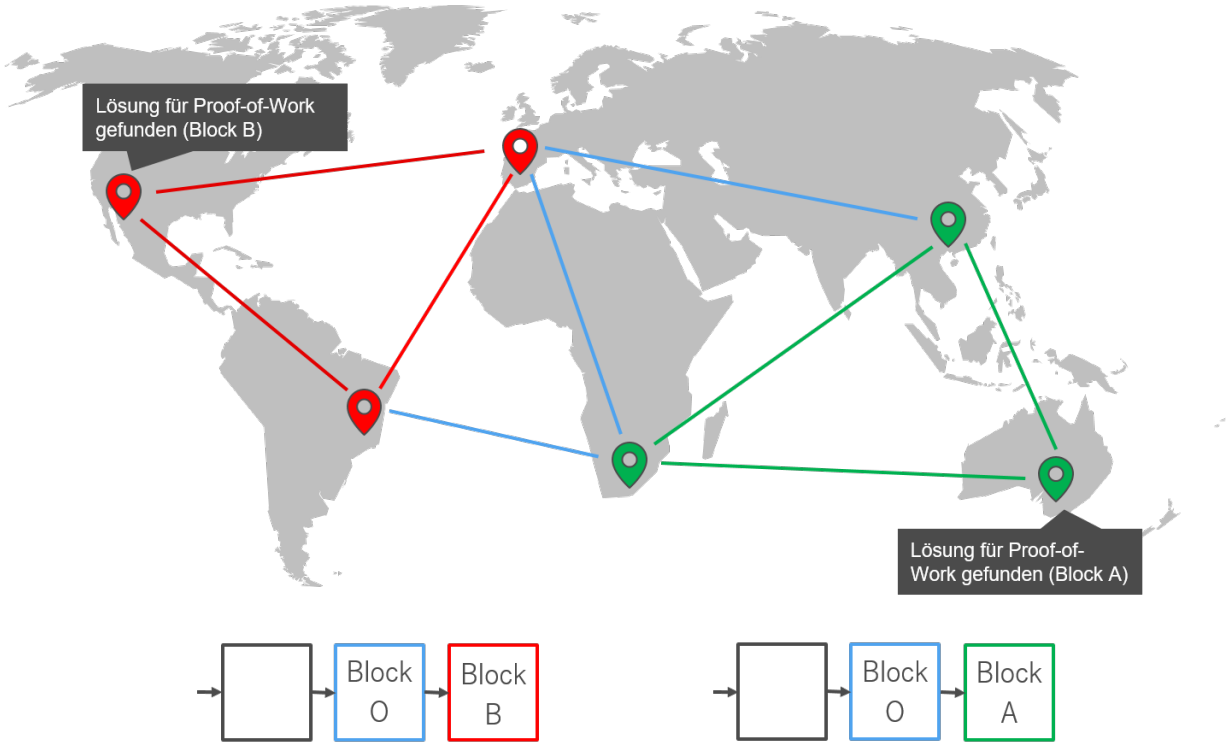

Abbildung 2.3: Fork-Visualisierung - 2 Nodes finden zur ungefähr gleichen Zeit einen Block und verbreiten ihn im Netzwerk, womit 2 Versionen der Blockchain bestehen

Block-Hash zu finden, wird der benötigte Aufwand klar. 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 als die bestehende. 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 [17].

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 nur möglich DoS-Angriffe auszuführen indem er verhindert das bestimmte Transaktionen in Blöcke aufgenommen werden, oder im Falle von Kryptowährungen die sogenannte Double-Spending-Attacke [11].

- Wie sicher ist die Blockchain ?
- Sicherheitsfaktoren

2.1.4 Blockchaintypen

- Public/Permissioned/Private Blockchains

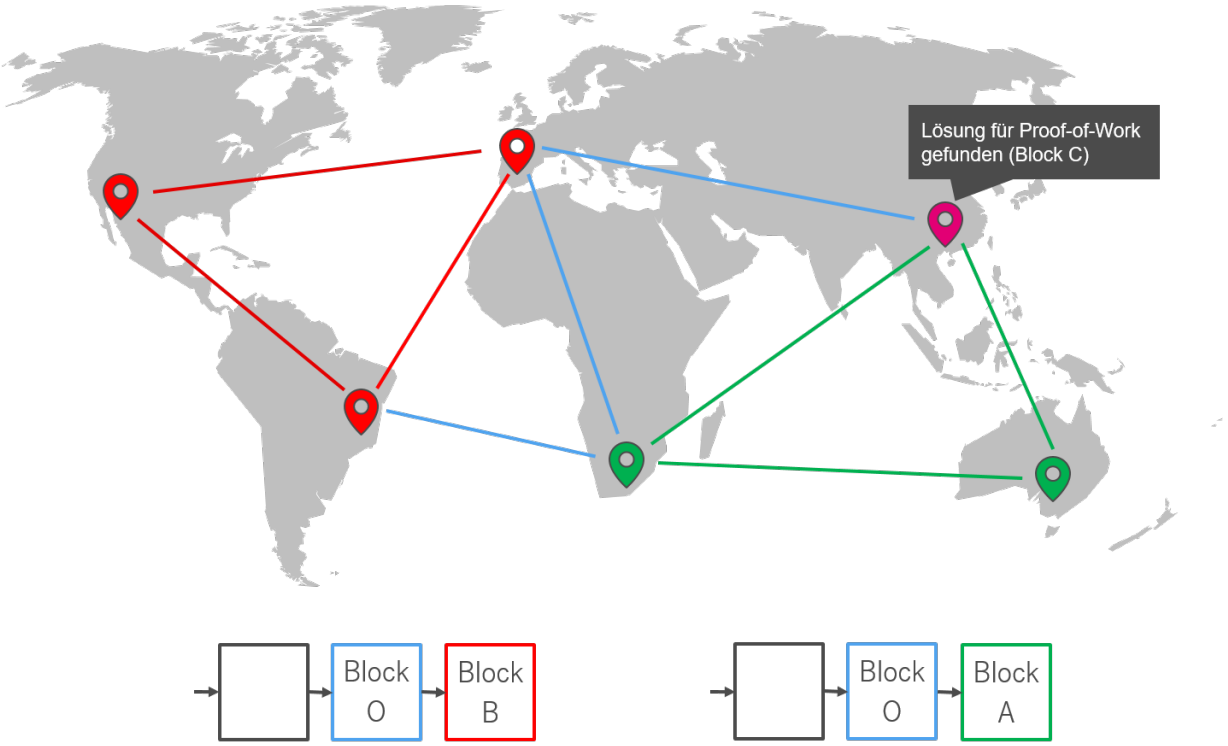

Abbildung 2.4: Fork-Visualisierung - Eine Node, welche Block A zuerst erhalten hat, hängt daran einen neuen Block C an

2.2 Exemplarische Anwendungsfälle

- Wozu kann die Blockchain genutzt werden ?

2.3 Probleme für den B2B-Bereich

- Welche Nachteile hat die Blockchain im B2B-Bereich ?
- Blockerstellungszeit –> Warum konstant, 10 Minuten etc. ?
- Bestätigungszeit für eine Transaktion

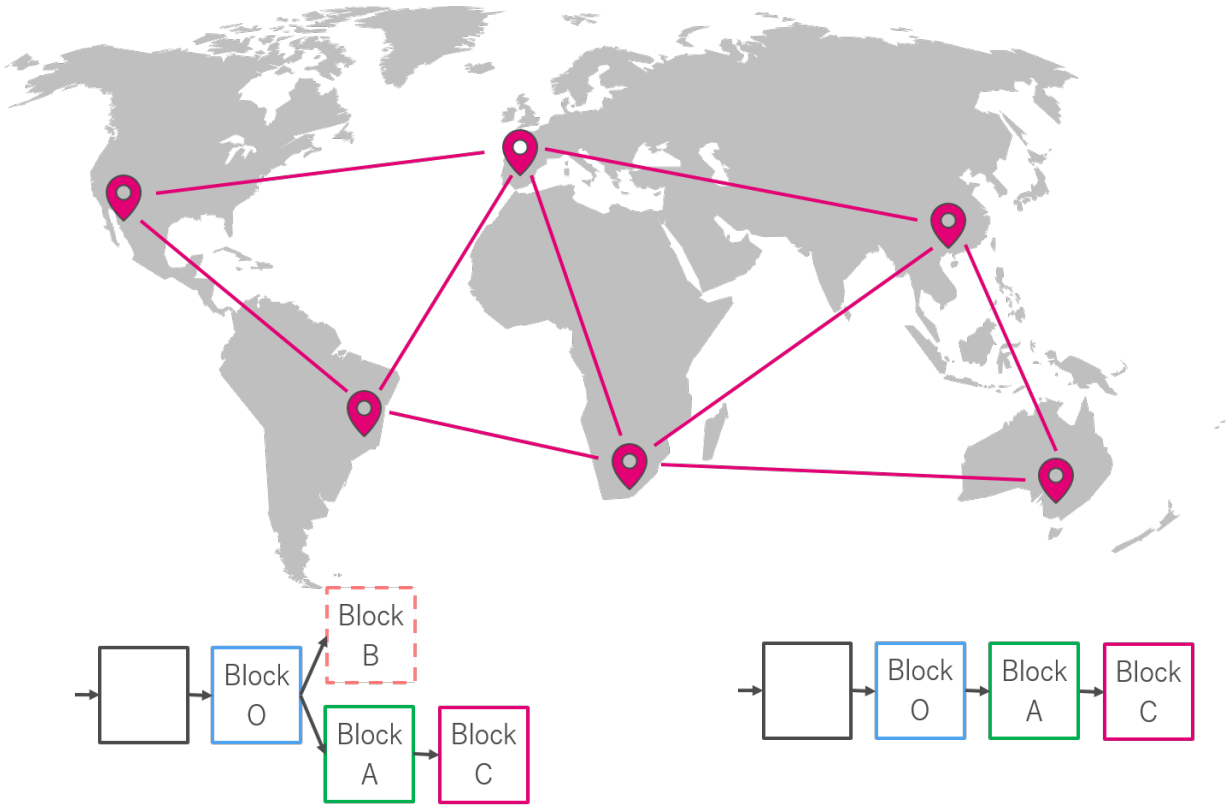

Abbildung 2.5: Fork-Visualisierung - Block C verbreitet sich im Netzwerk, rote Nodes sehen zwei Blockchains und akzeptieren die längere

Kapitel 3

Dezentraler Wartungsmarkt - Konzept

3.1 Beschreibung

- Funktion des Wartungsmarktes

3.2 Anforderungen

- Funktionale und nichtfunktionale Anforderungen

Kapitel 4

Aktueller Stand der Technik/Related Work

4.1 Dezentrale Anwendungen ohne Blockchain

- Benötigt man die Blockchain für dezentrale Anwendungen ?
- P2P versus Blockchain
- Synchronisieren der Daten in verteilten Systemen

4.2 Klassische B2B-Anwendungen

- Wie funktionieren aktuelle/klassische B2B-Anwendungen ?

4.3 Dezentrale Anwendungen/Märkte

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

Kapitel 5

Evaluierung Permissioned Blockchains für B2B

5.1 Skalierbarkeit

- Sind Permissioned Blockchains skalierbar ?

5.2 Konsensmechanismen

- Gibt es Konsensmechanismen welche mit einer Permissioned Blockchain funktionieren ?

5.3 Datenschutz

- Müssen alle Daten öffentlich verfügbar sein ?

Kapitel 6

Dezentraler Wartungsmarkt - Prototyp

6.1 Technologieauswahl

- Welche Blockchain wird genutzt um die Anforderungen zu erfüllen ?

6.2 Modell

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

6.3 Gerätesimulation durch Bosch XDK

- Simulation eines IOT-Geräts durch einen Bosch XDK

6.4 Programmlogik

- Funktion der Transaktionen

6.5 Benutzeroberflächen

- UIs für die Interaktion mit der Blockchain

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

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

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

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

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

[7] Ethereum White Paper, December 2017.

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

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

[10] Saveen A. Abeyratne and Radmehr P. Monfared. Blockchain ready manufacturing supply chain using distributed ledger. 2016.

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

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

[13] 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.

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

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

[16] 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.

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

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

[19] 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.