



Hochschule Darmstadt

– Fachbereich Informatik–

Untersuchung von Hyperledger Fabric für Blockchain-Technologien und dessen Anwendung für eine Kryptowährung hinsichtlich Privatheit und Performance

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc).

vorgelegt von

Bernd Nötscher

Referent : Prof. Dr. Alois Schütte

Korreferent : Prof. Dr.-Ing. Michael Bredel

Bernd Nötscher: *Untersuchung von Hyperledger Fabric für Blockchain-Technologien und dessen Anwendung für eine Kryptowährung hinsichtlich Privatheit und Performance*, © 25. August 2018

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 25. August 2018

Bernd Nötscher

ABSTRACT

Private and permissioned blockchains offer new possibilities for applications in the B2B environment and have recently been the focus of attention.

This thesis examines and evaluates the individual components of Hyperledger Fabric, an open source framework for private and permissioned blockchains.

A concept of a crypto currency and its implementation with Hyperledger Fabric is presented as a practical application. In addition to the Java client, the configuration and setup of the peer-to-peer network using Docker and AWS is explained. Furthermore, this paper describes how the conventional programming language Golang is used for the smart contracts to implement a crypto currency consisting of several blockchains.

This paper shows that this crypto currency achieves a performance of at least 18 transactions per second, with minimal latency and immediate finality, using a consensus based on majority votings. At the same time, privacy (e.g. non-traceability) is guaranteed through the use of encryption methods (such as AES and RSA), pseudonyms and data separation.

In the end, the known crypto currencies Bitcoin and Monero are analyzed and compared in terms of performance, privacy and non-traceability.

Keywords:

Distributed Ledger Technology, Blockchain Technology, Hyperledger Fabric, Smart contracts, Chaincode, Cryptocurrency, Private Blockchains

ZUSAMMENFASSUNG

Private und zugangsbeschränkte Blockchains bieten neue Möglichkeiten für Anwendungen im B2B-Umfeld und stehen entsprechend im Fokus der letzten Zeit.

Diese Thesis untersucht und bewertet die einzelnen Komponenten von Hyperledger Fabric, einem Open-Source-Framework für private und zugangsbeschränkte Blockchains.

Als praktische Anwendung wird ein Konzept einer Kryptowährung und die dazugehörige Implementierung mit Hyperledger Fabric vorgestellt. Dabei wird neben dem Java-Client die Konfiguration und Inbetriebnahme des Peer-to-Peer-Netzwerks durch Verwendung von Docker als auch AWS erläutert. Weiterhin beschreibt diese Arbeit wie die konventionelle Programmiersprache Golang für die intelligenten Verträge verwendet wird um eine Kryptowährung bestehend aus mehreren Blockchains zu implementieren.

Mit dieser Arbeit wird gezeigt, dass diese Kryptowährung mit Hilfe eines auf Mehrheitsentscheidungen basierenden Konsens eine Performance von mindestens 18 Transaktionen pro Sekunde erreicht, bei minimaler Latenz und sofortiger Finalität. Gleichzeitig wird die Privatheit (wie z. B. die Nichtnachverfolgbarkeit) durch die Anwendung von Verschlüsselungsverfahren (wie AES und RSA), Pseudonymen und Datenseparation gewährleistet.

Am Ende werden die bekannten Kryptowährungen Bitcoin und Monero analysiert und mit diesen ein Vergleich hinsichtlich Performance, Privatheit und Nichtnachverfolgbarkeit vollzogen.

Schlüsselwörter:

Distributed Ledger Technologie, Blockchain Technologie, Hyperledger Fabric, Smart contracts, Chaincode, Kryptowährungen, Private Blockchains, Intelligente Verträge

INHALTSVERZEICHNIS

I THESIS

1	EINLEITUNG	2
1.1	Motivation	2
1.2	Ziel der Arbeit	3
1.3	Gliederung	4
2	BLOCKCHAIN-GRUNDLAGEN	5
2.1	Anwendungsmöglichkeiten für eine Blockchain	6
2.2	Weitere Potenziale der Blockchaintechnologie	7
2.3	Blockchainarten	7
3	HYPERLEDGER	9
3.1	Projekte und Frameworks	9
3.2	Hyperledger Fabric	10
3.3	Abgrenzung von Fabric zu anderen Blockchainlösungen	10
3.4	Hyperledger Composer	11
3.5	Vergleich von Fabric und Composer	11
3.5.1	Vorteile von Hyperledger Composer	11
3.5.2	Nachteile von Hyperledger Composer	12
3.5.3	Vorteile von Hyperledger Fabric	13
3.5.4	Nachteile von Hyperledger Fabric	13
3.6	Tools für Hyperledger Fabric	14
4	UNTERSUCHUNG VON HYPERLEDGER FABRIC	15
4.1	Entwicklungsstand	15
4.2	Dokumentation, Beispiele und Bugs	16
4.3	P2P-Netzwerk	17
4.3.1	Orderer	17
4.3.2	Peers	18
4.3.3	Clients	18
4.4	Chaincode	18
4.5	Konsens und Richtlinien	19
4.5.1	Bewilligungsrichtlinie	19
4.5.2	Instanziierungsrichtlinie	20
4.5.3	Konsensarten	21
4.5.4	Konsens mit SOLO	21
4.5.5	Konsens mit Kafka und ZooKeeper	21
4.5.6	Konsens mit PBFT	21
4.6	Transaktionen	22
4.7	Datenspeicherung	23
4.8	Sicherheit	24
4.8.1	Zertifikate und Signaturen	24
4.8.2	Berechtigungen, Identitäten und Mitgliedschaften	25
4.8.3	Isolierte Prozesse	26

4.8.4	Integrität des Chaincodes	26
4.8.5	Integrität der Daten	26
4.8.6	Schutz vor Replay-Angriffen und Double-Spending	26
4.8.7	Schutz vor Denial-of-Service-Angriffen	27
4.8.8	Hardware Security Module	27
4.9	Vertraulichkeit	27
4.10	Fazit zum Entwicklungsstand	28
5	KONZEPT FÜR EINE KRYPTOWÄHRUNG	30
5.1	Autorisierung für eine Überweisung	30
5.2	Pseudonym und Identität	32
5.3	Vertraulichkeit über Höhe eines Kontostands	33
5.4	Besitzstände nicht berechenbar	34
5.5	Zahlungen sollen nicht nachverfolgbar sein	36
5.6	Konsens	38
5.7	Anwendungsfälle	39
5.8	Fazit zum Konzept	40
6	IMPLEMENTIERUNG EINER KRYPTOWÄHRUNG	42
6.1	Entwicklung des Chaincodes	42
6.2	Bereitstellung eines Netzwerks	47
6.2.1	Netzwerktopologie	47
6.2.2	Konfiguration	49
6.2.3	Bereitstellung eines lokalen Netzwerks	49
6.2.4	Konfigurationsdateien	50
6.2.5	Automatisierung	50
6.2.6	Bereitstellung eines Remote-Netzwerks auf verschiedenen Hosts	51
6.3	Entwicklung eines Clients	52
6.4	Offene Punkte	53
6.4.1	Optionale Punkte	53
6.4.2	Notwendige Punkte	54
6.4.3	Feature „Identity Mixer“	55
6.5	Diskussion und Bewertung	56
7	EXPERIMENTE	59
7.1	Peer- und Orderer-Versuchsaufbau	59
7.2	Vorgehensweise	60
7.3	Experimente mit einem CLI-Client	61
7.4	Experimente mit einem Java-Client	61
7.5	Ergebnisse	62
7.5.1	Vergleich von Lesetest mit Schreibtest	62
7.5.2	Vergleich von CLI mit Java	63
7.5.3	Vergleich von t2.micro mit t2.large und m5.large	63
7.5.4	Vergleich: viele Clients vs. ein Client	63
7.5.5	Vergleich von 10 mit 100 und 1000 Transaktionen	63
7.5.6	Vergleich von Überweisung mit Umbuchung	63
7.5.7	Vergleich von Fabric 1.1.1 mit Fabric 1.2	64
7.5.8	Vergleich mit acht Banken und mit t2.2xlarge	64

7.6 Schwierigkeiten	65
7.7 Erkenntnisse	65
8 SKALIERUNG	69
9 ANGRIFFSSZENARIEN	72
9.1 Client	72
9.2 Peer	72
9.3 Orderer	74
10 VERGLEICH MIT ANDEREN KRYPTOWÄHRUNGEN	75
10.1 Performance	75
10.2 Bitcoin	76
10.3 Cryptonote	78
10.4 Monero	80
10.5 Privatheit und Nachverfolgbarkeit	81
11 ZUSAMMENFASSUNG UND AUSBLICK	83
II APPENDIX	
A IMPLEMENTIERUNG EINER KRYPTOWÄHRUNG	86
B EXPERIMENTE	91
LITERATUR	94

ABBILDUNGSVERZEICHNIS

Abbildung 2.1	Schema für eine private Blockchain mit einer Kryptowährung betrieben von Banken	8
Abbildung 3.1	Übersicht der verwendeten Elemente und Teilprojekte von Hyperledger	12
Abbildung 4.1	Systembestandteile von Fabric	16
Abbildung 4.2	Blockstruktur der Blockchain von Hyperledger Fabric .	23
Abbildung 5.1	Schematischer Netzwerkaufbau mit seinen Elementen .	31
Abbildung 5.2	Übersicht über die einzelnen Blockchains bzw. Kanäle	35
Abbildung 5.3	Kontobilanzierung für Soll- und Habenwerte	36
Abbildung 5.4	Beispiel Kontobilanzierung mit Wertübertragung und Verschlüsselung	38
Abbildung 6.1	Verwendete Systembestandteile von Hyperledger Fabric	43
Abbildung 6.2	Überblick über installierte Kanäle und Chaincode auf einem Peer	47
Abbildung 6.3	Ablauf einer Transaktion mit Und-Bewilligungsrichtlinie im P2P-Netzwerk	48
Abbildung 6.4	Bildschirmfoto von einer SSH mit gestarteten Peer- und Orderer-Prozessen	52
Abbildung 7.1	Diagramme zu den wichtigsten Ergebnissen der Experimente	68
Abbildung 8.1	Skalierungsvorschlag für Banken, Konten, Nutzer und Transaktionsvolumen und dessen Ressourcenbedarf . .	71
Abbildung 9.1	Übersicht über Konsequenzen bei Unwirksamkeit der eingesetzten Kryptographie	73
Abbildung 10.1	Vereinfacht dargestellte Transaktion von Bitcoins	77
Abbildung 10.2	Vereinfacht dargestellte Transaktion in CryptoNote . .	79
Abbildung A.1	Bildschirmfoto von der IDE zur Automatisierung von Fabric, Docker und AWS	87

TABELLENVERZEICHNIS

Tabelle 3.1	Bestehende Blockchain-Implementierungen von Hyperledger und deren Versionsnummer.	9
Tabelle 5.1	Relation zwischen Anzahl teilnehmender Banken und Anzahl der Kanäle bzw. Blockchains.	35
Tabelle 5.2	Bewertung der Privatheit und Nichtnachverfolgbarkeit aus Sicht der verschiedenen Systembeteiligten . . .	41
Tabelle 6.1	Setup-Performancetests einer lokalen Fabric-Umgebung	50
Tabelle 6.2	Setup-Performancetests einer Fabric-Umgebung mit AWS	51
Tabelle 7.1	Verwendete Hardware mit AWS	60
Tabelle 7.2	Ergebnisse der Schreibtests mit 1000 Überweisungen mit Java-Client und Fabric 1.2	65
Tabelle 7.3	Ergebnisse der Schreibtests mit 1000 Umbuchungen mit Java-Client und Fabric 1.2	65
Tabelle 7.4	Ergebnisse der Schreibtests mit 1000 Überweisungen mit Java-Client und Fabric 1.2 mit acht Banken	66
Tabelle 7.5	Ergebnisse der Schreibtests mit 1000 Überweisungen mit Java-Client und Fabric 1.2 mit t2.2xlarge	66
Tabelle 7.6	Ergebnisse der Lesetests mit 1000 Lesezugriffen mit CLI-Client	67
Tabelle 7.7	Ergebnisse der Schreibtests mit 1000 Umbuchungen mit Java-Client	67
Tabelle 7.8	Ergebnisse der Schreibtests mit 1000 Überweisungen mit Java-Client	68
Tabelle 10.1	eCoin im Vergleich hinsichtlich Konsens, Transaktionsrate, Latenz und Finalität	76
Tabelle 10.2	eCoin im Vergleich hinsichtlich Performance, Privatheit und Nichtnachverfolgbarkeit	82
Tabelle B.1	Lesetests mit CLI-Client mit 1000 Lesezugriffen	91
Tabelle B.2	1000 Umbuchungen mit CLI-Client	91
Tabelle B.3	10 Umbuchungen mit vielen Java-Clients	91
Tabelle B.4	100 Umbuchungen mit vielen Java-Clients	92
Tabelle B.5	10 Umbuchungen mit einem Java-Client	92
Tabelle B.6	100 Umbuchungen mit einem Java-Client	92
Tabelle B.7	1000 Umbuchungen mit einem Java-Client	93
Tabelle B.8	1000 Überweisungen mit einem Java-Client	93

LISTE DER QUELLLTEXTE

Quelltext 6.1	Mögliche Befehle des Java-Clients.	57
Quelltext 6.2	Beispiele gespeicherter Transaktionen in der Blockchain	58
Quelltext A.1	Konfiguration der SSH-Zugänge für die Knoten im AWS-Netzwerk	87
Quelltext A.2	Automatisierungsbeispiel zur Erstellung von Konfigu- rationen	88
Quelltext A.3	Automatisierungsbeispiel zur SSH-Verwendung	88
Quelltext A.4	Automatisierungsbeispiel zur Kanalerstellung	89
Quelltext A.5	Gekürzter Chaincode-Quelltext für die Kontostandser- mittlung (Quelltext in Golang).	90
Quelltext B.1	Lesezugriff mit CLI-Client	92
Quelltext B.2	Schreibzugriff mit CLI-Client	93

Teil I

THESIS