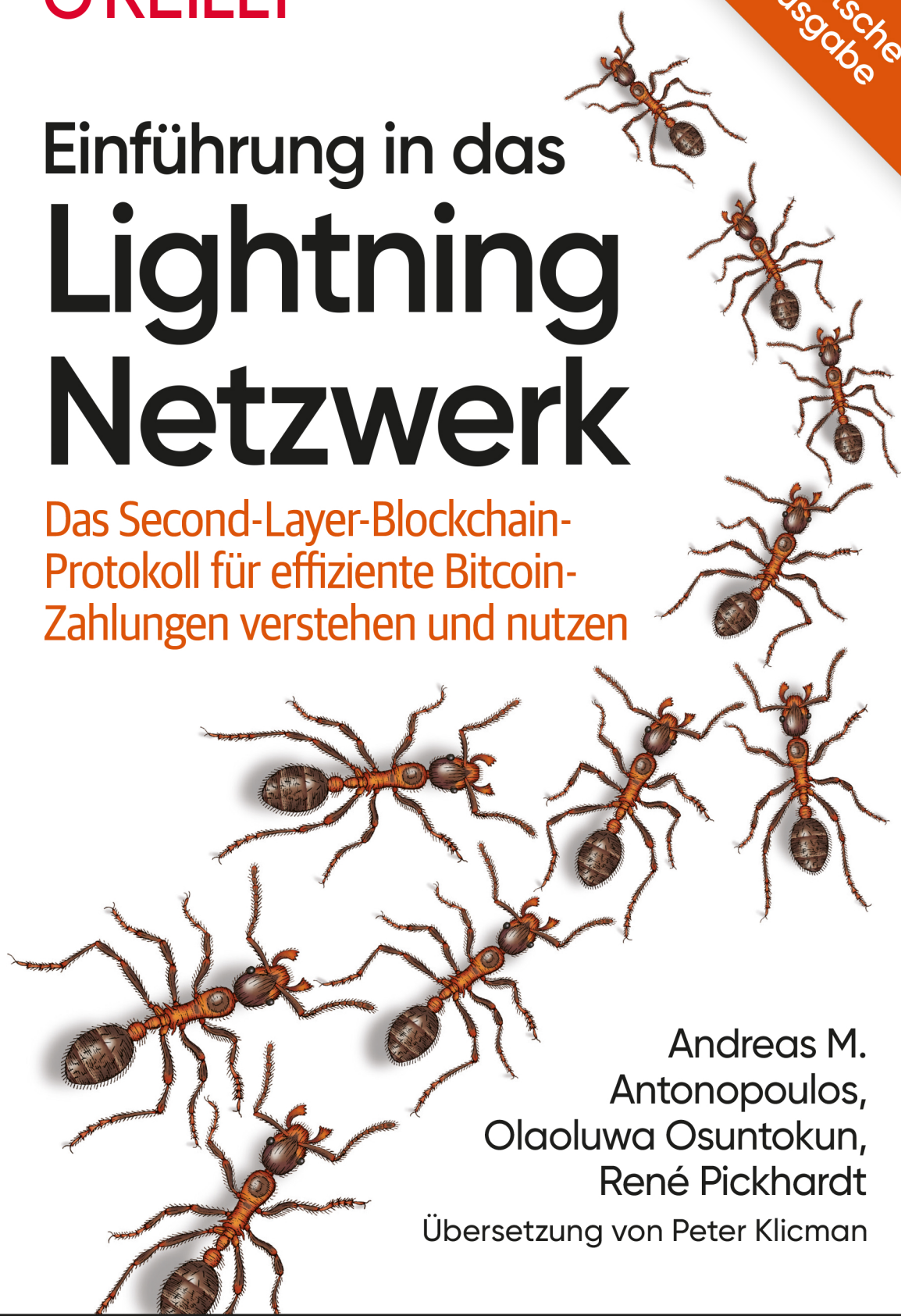


O'REILLY®

Deutsche
Ausgabe

Einführung in das Lightning Netzwerk

Das Second-Layer-Blockchain-
Protokoll für effiziente Bitcoin-
Zahlungen verstehen und nutzen



Andreas M.
Antonopoulos,
Olaoluwa Osuntokun,
René Pickhardt

Übersetzung von Peter Klicman

Coypright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

Einführung in das Lightning-Netzwerk

*Das Second-Layer-Blockchain-Protokoll
für effiziente Bitcoin-Zahlungen verstehen und nutzen*

***Andreas M. Antonopoulos, Olaoluwa Osuntokun
und René Pickhardt***

*Deutsche Übersetzung von
Peter Klicman*

O'REILLY®

Andreas M. Antonopoulos, Olaoluwa Osuntokun, René Pickhardt

Lektorat: Ariane Hesse

Übersetzung: Peter Klicman

Fachliche Unterstützung: René Pickhardt

Korrektorat: Sibylle Feldmann, www.richtiger-text.de

Satz: Jörg Liedtke, Flensburg

Herstellung: Stefanie Weidner

Umschlaggestaltung: Michael Oréal, www.oreal.de

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-201-8

PDF 978-3-96010-736-1

ePub 978-3-96010-737-8

mobi 978-3-96010-738-5

1. Auflage 2023

Translation Copyright © 2023 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.

O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Authorized German translation of the English edition of *Mastering the Lightning Network*

ISBN 9781492054863 © 2022 aantonop Books LLC, René Pickhardt, and uuddlrirbas LLC.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: kommentar@oreilly.de.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

Vorwort	17
Vorwort zur deutschen Ausgabe	25

Teil I Das Lightning-Netzwerk verstehen

1 Einführung	31
Grundlegende Konzepte des Lightning-Netzwerks	32
Vertrauen in dezentralisierten Netzwerken	33
Fairness ohne zentrale Autorität	34
Vertrauenswürdige Protokolle ohne Intermediäre	35
Ein Fairness-Protokoll in Aktion	36
Sicherheits-Primitive als Grundbausteine	37
Beispiel für Fairness-Protokolle	38
Motivation für das Lightning-Netzwerk	39
Blockchains skalieren	40
Die das Lightning-Netzwerk definierenden Eigenschaften	42
Anwendungsfälle, Nutzer und ihre Geschichten	42
Fazit	43
2 Erste Schritte	45
Alice' erste Lightning-Wallet	45
Lightning-Nodes	46
Lightning-Explorer	46
Lightning-Wallets	47
Testnet-Bitcoin	50
Balance zwischen Komplexität und Kontrolle	51
Download und Installation einer Lightning-Wallet	52
Eine neue Wallet anlegen	53
Verantwortung für die Schlüsselverwahrung	53
Mnemonische Wörter	53
Die mnemonische Phrase sichern	54

Bitcoin auf die Wallet laden	55
Bitcoin beschaffen	55
Bitcoin empfangen	55
Von Bitcoin zum Lightning-Netzwerk	59
Lightning-Netzwerk-Kanäle	59
Einen Lightning-Kanal öffnen	61
Eine Tasse Kaffee über das Lightning-Netzwerk kaufen	63
Bobs Café	63
Eine Lightning-Rechnung	64
Fazit	66
3 Wie das Lightning-Netzwerk funktioniert	67
Was ist ein Zahlungskanal?	68
Grundlagen eines Zahlungskanals	68
Zahlungen über Kanäle routen	69
Zahlungskanäle	70
Multisignaturadressen	71
Funding-Transaktion	71
Commitment-Transaktionen	73
Betrug mit vorherigem Zustand	74
Ankündigung des Kanals	77
Den Kanal schließen	78
Rechnungen	82
Zahlungshash und Preimage	83
Zusätzliche Metadaten	84
Zustellung der Zahlung	85
Das Peer-to-Peer-Gossip-Protokoll	85
Wegfindung und Routing	86
Quellbasierte Wegfindung	87
Onion-Routing	88
Algorithmus zur Zahlungsweiterleitung	90
Verschlüsselung der Peer-to-Peer-Kommunikation	91
Überlegungen zum Vertrauen	92
Vergleich mit Bitcoin	93
Adressen versus Rechnungen, Transaktionen versus Zahlungen	93
Outputs wählen versus Pfad finden	94
Change-Output bei Bitcoin versus kein Wechselgeld bei Lightning	94
Mining-Gebühren versus Routing-Gebühren	95
Gebühren basierend auf Traffic versus angekündigte Gebühren	95
Öffentliche Bitcoin-Transaktionen versus private	
Lightning-Zahlungen	96
Warten auf Bestätigungen versus sofortiger Zahlungseingang	96

Senden beliebiger Summen versus Kapazitätsbeschränkungen	97
Anreize für Großbeträge versus Kleinbeträge	97
Blockchain als Kassenbuch versus Blockchain als	
Gerichtssystem	98
Offline versus online, asynchron versus synchron	98
Satoshis versus Millisatoshis	99
Gemeinsamkeiten von Bitcoin und Lightning	99
Monetäre Einheit	99
Unumkehrbarkeit und Finalität von Zahlungen	99
Vertrauen und Gegenparteirisiko	100
Genehmigungsfreier Betrieb	100
Open Source und Open System	100
Fazit	100
4 Lightning-Node-Software	101
Die Lightning-Entwicklungsumgebung	102
Die Kommandozeile nutzen	102
Das Buch-Repository herunterladen	103
Docker-Container	103
Bitcoin Core und Regtest	105
Den Bitcoin-Core-Container erzeugen	106
Das c-lightning-Lightning-Node-Projekt	109
c-lightning als Docker-Container	109
Ein Docker-Netzwerk einrichten	110
Ausführen der bitcoind- und c-lightning-Container	110
c-lightning aus dem Quellcode installieren	112
Vorab benötigte Bibliotheken und Pakete installieren	112
Den c-lightning-Quellcode kopieren	113
Den c-lightning-Quellcode kompilieren	113
Das Lightning-Netzwerk-Daemon-Node-Projekt	115
Der LND-Docker-Container	116
Die bitcoind- und LND-Container ausführen	117
LND aus dem Quellcode installieren	118
Den LND-Quellcode kopieren	119
Den LND-Quellcode kompilieren	119
Das Eclair-Lightning-Node-Projekt	120
Der Eclair-Docker-Container	120
Die bitcoind- und Eclair-Container ausführen	121
Eclair aus dem Quellcode installieren	123
Den Eclair-Quellcode installieren	123
Den Eclair-Quellcode kompilieren	124

Ein Netzwerk diverser Lightning-Nodes aufbauen.	124
docker-compose zur Orchestrierung von Docker-Containern	
nutzen	125
docker-compose-Konfiguration	125
Das Demo-Lightning-Netzwerk starten.	126
Kanäle öffnen und eine Zahlung routen	127
Fazit	129
5 Eine Lightning-Netzwerk-Node betreiben.	131
Eine Plattform wählen	132
Warum ist Zuverlässigkeit für den Betrieb einer	
Lightning-Node wichtig?	132
Hardware für Lightning-Nodes	133
Betrieb in der Cloud.	133
Eine Node zu Hause betreiben.	134
Welche Hardware wird zum Betrieb einer Lightning-Node	
benötigt?	135
Serverkonfiguration in der Cloud wechseln.	136
Einen Installer oder Helfer nutzen	137
RaspiBlitz	137
Mynode	138
Umbrel.	138
BTCPay Server	140
Bitcoin-Node oder leichtgewichtige Lightning-Node	140
Wahl des Betriebssystems	141
Wahl der Lightning-Node-Implementierung	142
Eine Bitcoin- oder Lightning-Node installieren	143
Hintergrunddienste	143
Prozessisolation	144
Node starten.	144
Node-Konfiguration.	146
Netzwerkkonfiguration	147
Die Sicherheit Ihrer Node	151
Betriebssystemsicherheit	152
Node-Zugriff	153
Node- und Kanal-Backups	154
Hot-Wallet-Risiko	156
Sweeping von Guthaben	156
Uptime und Verfügbarkeit einer Lightning-Node	159
Fehler tolerieren und die Dinge automatisieren.	159
Die Node-Verfügbarkeit überwachen	160
Watchtower	161

Kanalmanagement	162
Ausgehende Kanäle öffnen	162
Eingehende Liquidität beschaffen	166
Kanäle schließen	167
Kanäle wieder ausgleichen	167
Routing-Gebühren	168
Node-Management	169
Ride The Lightning	170
lndmon	170
ThunderHub	170
Fazit	171

Teil II Das Lightning-Netzwerk im Detail

6 Architektur des Lightning-Netzwerks	175
Die Lightning-Netzwerk-Protokoll-Suite	175
Lightning im Detail	176
7 Zahlungskanäle	179
Eine andere Art, das Bitcoin-System zu nutzen	180
Bitcoin-Eigentümerschaft und -Kontrolle	181
Diversität der (unabhängigen) Eigentümerschaft und Multisig	181
Gemeinsamer Besitz ohne unabhängige Kontrolle	182
»Gespernte« und nicht einzulösende Bitcoin verhindern	182
Einen Zahlungskanal aufbauen	182
Private und öffentliche Node-Schlüssel	183
Node-Netzwerkadresse	183
Node-Kennungen	183
Nodes als direkte Peers verbinden	184
Den Kanal aufbauen	184
Peer-Protokoll für das Kanalmanagement	185
Nachrichtenfluss beim Aufbau des Kanals	185
Die Funding-Transaktion	188
Eine Multisignaturadresse generieren	189
Die Funding-Transaktion konstruieren	189
Signierte Transaktionen halten, aber nicht veröffentlichen	190
Refunding vor Funding	190
Die vorsignierte Refund-Transaktion konstruieren	190
Transaktionen ohne Veröffentlichung verketteten	191
(Ver-)Formbarkeit von Transaktionen (Segregated Witness)	192
Die Funding-Transaktion veröffentlichen	194

Zahlungen über den Kanal senden	195
Das Guthaben aufteilen	195
Konkurrierende Commitments	196
Mit alten Commitment-Transaktionen betrügen.	197
Alte Commitment-Transaktionen widerrufen	197
Asymmetrische Commitment-Transaktionen	198
Verzögerte Auszahlung von to_self (Timelock)	199
Widerrufsschlüssel.	200
Die Commitment-Transaktion	201
Den Kanalzustand verändern	203
Die commitment_signed-Nachricht.	204
Die revoke_and_ack-Nachricht	204
Widerruf und neuer Commit.	205
Betrug und Sanktionierung in der Praxis	205
Die Kanalreserve: das Spiel am Laufen halten	208
Den Kanal schließen (kooperatives Schließen)	209
Die shutdown-Nachricht	209
Die closing_signed-Nachricht	210
Die Kooperatives-Schließen-Transaktion.	211
Fazit	212
8 Routing in einem Netzwerk aus Zahlungskanälen	213
Routing einer Zahlung	213
Routing versus Wegfindung	215
Ein Netzwerk von Zahlungskanälen aufbauen	215
Ein reales Beispiel für das Routing	216
Fairness-Protokoll.	222
Implementierung atomarer vertrauensfreier Multihop-	
Zahlungen	222
Ein erneuter Blick auf das Spenden-Beispiel.	223
On-Chain- versus Off-Chain-Abwicklung von HTLCs	224
Hash Time-Locked Contracts.	225
HTLCs in Bitcoin-Skript	226
Zahlungs-Preimage und Hashverifikation	227
HTLCs von Alice zu Dina propagieren	227
Rückpropagation des Secrets	228
Signaturbindung: Diebstahl von HTLCs verhindern	230
Hashoptimierung.	232
Kooperations- und Timeout-Fehler	233
Kürzere Timelocks	234
Fazit	235

9	Kanalbetrieb und Zahlungsverweiterung	237
	Lokal (einzelner Kanal) versus geroutet (mehrere Kanäle)	238
	Zahlungen weiterleiten und Commitments aktualisieren mit HTLCs	238
	Nachrichtenfluss für HTLC und Commitment	239
	Zahlungen mit HTLCs weiterleiten	239
	Einen HTLC hinzufügen	239
	Die update_add-HTLC-Nachricht	240
	HTLC in Commitment-Transaktionen	241
	Neues Commitment mit HTLC-Output	242
	Alice' Commit	242
	Bob bestätigt das neue Commitment und widerruft das alte	244
	Bobs Commit	246
	Mehrere HTLCs	248
	Den HTLC einhalten	249
	HTLC-Propagation	249
	Dina erfüllt den HTLC mit Chan	249
	Bob verrechnet den HTLC mit Alice	250
	Einen HTLC aufgrund eines Fehlers oder Verfallsdatums entfernen	253
	Eine lokale Zahlung vornehmen	254
	Fazit	255
10	Onion-Routing	257
	Ein anschauliches Beispiel des Onion-Routings	258
	Einen Pfad wählen	258
	Die Schichten aufbauen	259
	Die Schichten abschälen	261
	Einführung in das Onion-Routing von HTLCs	262
	Alice wählt den Pfad	262
	Alice konstruiert die Nutzdaten	264
	Schlüsselgenerierung	267
	Die Onion-Schichten verpacken	271
	Onions fester Länge	271
	Die Onion verpacken (Zusammenfassung)	272
	Dinas Hop-Nutzdaten verpacken	273
	Chans Hop-Nutzdaten verpacken	277
	Bobs Hop-Nutzdaten verpacken	278
	Das finale Onion-Paket	279
	Die Onion senden	280
	Die update_add_htlc-Nachricht	280
	Alice sendet die Onion an Bob	280
	Bob überprüft die Onion	281
	Bob generiert Füllbytes	281

Bob »entschleiern« seine Hop-Nutzdaten.	282
Bob extrahiert den äußeren HMAC für den nächsten Hop.	283
Bob entfernt seine Nutzdaten und verschiebt die Onion-Daten nach links.	283
Bob konstruiert das neue Onion-Paket	284
Bob verifiziert die HTLC-Details	284
Bob sendet update_add_htlc an Chan.	284
Chan leitet die Onion weiter	285
Dina empfängt die finalen Nutzdaten	286
Fehler zurückgeben.	286
Fehlermeldungen	287
Spontane Zahlungen per keysend.	289
Benutzerdefinierte Onion-TLV-Records	290
keysend-Zahlungen senden und empfangen	290
Keysend und benutzerdefinierte Records in Lightning-Anwendungen	291
Fazit	291
11 Gossip und der Kanal-Graph.	293
Peers entdecken	296
P2P-Bootstrapping	296
DNS-Bootstrapping	297
SRV-Query-Optionen	300
Der Kanal-Graph.	301
Ein gerichteter Graph.	301
Gossip-Protokoll-Nachrichten	302
Die node_announcement-Nachricht	303
Die channel_announcement-Nachricht.	305
Die channel_update-Nachricht	309
Fortlaufende Pflege des Kanal-Graphen	310
Fazit	310
12 Wegfindung und Zustellung der Zahlung	311
Wegfindung in der Lightning-Protokoll-Suite	311
Wo ist das BOLT?	312
Wegfindung: Welches Problem lösen wir?	312
Wahl des besten Pfads	313
Wegfindung in Mathematik und Informatik	313
Kapazität, Guthaben, Liquidität	314
Unsicherheit der Guthaben	314
Komplexität der Wegfindung	315
Die Dinge einfach halten	316
Wegfindung und Zahlungszustellung.	316

Aufbau des Kanal-Graphen	317
Liquidität: Unsicherheit und Wahrscheinlichkeit	320
Gebühren und andere Kanalmetriken	322
Pfadcandidaten finden.	323
Zustellung der Zahlung (Versuch-und-Irrtum-Schleife)	324
Erster Versuch (Pfad #1)	324
Zweiter Versuch (Pfad #4)	325
Multipart-Zahlungen	326
MPP nutzen.	327
Versuch und Irrtum über mehrere »Runden«.	328
Fazit	330
13 Wire-Protokoll: Framing und Erweiterbarkeit	331
Messaging-Schicht in der Lightning-Protokoll-Suite	331
Wire-Framing	332
High-Level-Wire-Framing.	332
Typcodierung	333
Type-Length-Value-Nachrichtenerweiterungen	334
Das Protocol-Buffers-Nachrichtenformat	334
Vor- und Rückwärtskompatibilität.	334
Type-Length-Value-Format	335
BigSize-Integercodierung	336
Beschränkungen der TLV-Codierung.	336
Kanonische TLV-Codierung.	336
Feature-Bits und Erweiterbarkeit des Protokolls.	337
Feature-Bits als Mechanismus zur Erkennung von Upgrades.	337
TLV für Vor- und Rückwärtskompatibilität	339
Taxonomie des Upgrade-Mechanismus	339
Updates auf Ebene der Kanalkonstruktion.	341
Fazit	341
14 Lightnings verschlüsselter Nachrichtentransport	343
Verschlüsselter Transport in der Lightning-Protokoll-Suite	343
Einführung.	344
Der Kanal-Graph als dezentralisierte Public-Key-Infrastruktur.	344
Warum nicht TLS?.	345
The Noise-Protokoll-Framework.	345
Lightnings verschlüsselter Transport im Detail.	346
Noise_XK: Noise-Handshake des Lightning-Netzwerks	346
Handshake-Notation und Protokollfluss	347
Übersicht auf hohem Niveau	347
Handshake in drei Akten	348
Fazit	358

15	Lightning-Zahlungsanforderungen	359
	Rechnungen in der Lightning-Protokoll-Suite	359
	Einführung	359
	Lightning-Rechnungen versus Bitcoin-Adressen	360
	BOLT #11: Serialisierung und Interpretation von	
	Lightning-Rechnungen	361
	Rechnungscodierung in der Praxis	361
	Das visuell lesbare Präfix	361
	bech32 und das Datensegment	362
	Fazit	364
16	Sicherheit und Privatsphäre im Lightning-Netzwerk	365
	Warum ist Privatsphäre wichtig?	365
	Privatsphäre definieren	365
	Evaluierung der Privatsphäre	366
	Anonyme Menge	367
	Privatsphäre: Unterschiede zwischen	
	Lightning-Netzwerk und Bitcoin	368
	Angriffe auf Lightning	370
	Zahlungsbeträge beobachten	370
	Sender und Empfänger verknüpfen	370
	Kanalguthaben aufdecken (Probing)	372
	Denial of Service	374
	Commitment-Jamming	376
	Einfrieren der Kanalliquidität	376
	Schichtenübergreifende Deanonymisierung	376
	Clustering von On-Chain-Bitcoin-Entitäten	377
	Clustering von Off-Chain-Lightning-Nodes	378
	Schichtenübergreifende Verknüpfung: Lightning-Nodes	
	und Bitcoin-Entitäten	379
	Lightning-Graph	379
	Wie sieht der Lightning-Graph wirklich aus?	380
	Zentralisierung im Lightning-Netzwerk	382
	Ökonomische Anreize und Graph-Struktur	383
	Praktischer Rat zum Schutz der Privatsphäre	383
	Unangekündigte Kanäle	383
	Routing-Erwägungen	384
	Kanäle akzeptieren	385
	Fazit	386
	Quellenangaben und Literaturhinweise	386

17 Fazit	389
Dezentralisierte und asynchrone Innovation	389
Innovationen im Bitcoin-Protokoll und bei Bitcoin-Skript	390
Innovationen des Lightning-Protokolls	390
TLV-Erweiterbarkeit	391
Konstruktion von Zahlungskanälen	391
Opt-in-Ende-zu-Ende-Features	391
Lightning-Anwendungen (LApps)	392
Auf die Plätze, fertig, los!	393
 Anhang A Bitcoin-Grundlagen	 395
 Anhang B Docker: Grundlegende Installation und Nutzung	 415
 Anhang C Nachrichten des Wire-Protokolls	 419
 Anhang D Quellen und Lizenzinformationen	 437
 Glossar	 439
 Index	 457

Vorwort

Das Lightning-Netzwerk (engl. *Lightning Network*, kurz auch LN) ist ein Second-Layer-Peer-to-Peer-Netzwerk, das es uns erlaubt, Bitcoin-Zahlungen »Off-Chain« abzuwickeln, d.h., ohne sie als Transaktionen auf der Bitcoin-Blockchain bestätigen zu müssen.

Das Lightning-Netzwerk bietet uns sichere, günstige, schnelle und deutlich vertraulichere Bitcoin-Zahlungen, und das auch bei sehr kleinen Beträgen.

Basierend auf der Idee von Zahlungskanälen, die erstmals von Bitcoin-Erfinder Satoshi Nakamoto vorgeschlagen wurden, ist das Lightning-Netzwerk ein geroutetes Netzwerk, bei dem Zahlungen über einen Pfad von Zahlungskanälen vom Sender zum Empfänger geleitet werden.

Die ursprüngliche Idee des Lightning-Netzwerks wurde 2015 in der wegweisenden Arbeit »The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments« von Joseph Poon und Thaddeus Dryja vorgeschlagen. Im Jahr 2017 lief im Internet ein Lightning-»Test«-Netzwerk, in dem unterschiedliche Gruppen kompatible Implementierungen entwickelten und einige Kompatibilitätsstandards festlegten. 2018 ging das Lightning-Netzwerk »live«, und die Zahlungen begannen zu fließen.

Im Jahr 2019 vereinbarten Andreas M. Antonopoulos, Olaoluwa Osuntokun und René Pickhardt, beim Schreiben dieses Buchs zusammenzuarbeiten. Wie es scheint, mit Erfolg!

Leserkreis

Dieses Buch richtet sich hauptsächlich an ein technisches Publikum, das die Grundlagen von Bitcoin und anderen Blockchains versteht.

Verwendete Konventionen

Im Buch folgen wir diesen typografischen Konventionen:

Kursivschrift

Wird für neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen verwendet.

Nichtproportionalschrift

Wird für Programmlistings verwendet. Im normalen Fließtext werden damit Programmelemente wie Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter hervorgehoben.

Nichtproportionalschrift fett

Wird für Befehle oder andere Eingaben eingesetzt, die Sie wortwörtlich eingeben müssen.

Nichtproportionalschrift kursiv

Wird für Text verwendet, der durch benutzereigene oder durch den Kontext bestimmte Werte ersetzt wird.



Mit diesem Symbol wird ein Tipp oder ein Vorschlag angezeigt.



Mit diesem Symbol wird ein allgemeiner Hinweis angezeigt.



Mit diesem Symbol wird eine Warnung angezeigt.

Codebeispiele

Die Beispiele sind in Go, C++ und Python geschrieben und verwenden die Kommandozeilen unixoider Betriebssysteme. Alle Code-Snippets finden Sie im GitHub-Repository im *code*-Unterverzeichnis. Laden Sie den Buchcode herunter, probieren Sie die Codebeispiele aus und senden Sie Korrekturen an: GitHub (<https://github.com/lnbook/lnbook>).

Alle Code-Snippets können für die meisten Betriebssysteme mit einer minimalen Installation der Compiler, Interpreter und Bibliotheken für die entsprechenden Sprachen repliziert werden. Wenn nötig, stellen wir grundlegende Installationsanweisungen und schrittweise Beispiele für die Ausgaben bereit.

Einige der Code-Snippets wurden für den Druck aufbereitet. In diesen Fällen wurden die Zeilen mit einem Backslash-Zeichen (\) gefolgt von einem Newline-Zeichen getrennt. Wenn Sie mit diesen Beispielen arbeiten, müssen Sie die beiden Zeichen entfernen und die Zeilen wieder zusammenfassen. Die Ergebnisse sollten dann denen der Beispiele entsprechen.

Alle Code-Snippets verwenden wann immer möglich reale Werte und Berechnungen. Sie können sich also von Beispiel zu Beispiel vorarbeiten und kommen immer zu den gleichen Ergebnissen wie das Buch. So sind beispielsweise die privaten Schlüssel und die dazugehörigen öffentlichen Schlüssel und Adressen alle echt.

Verwendung der Codebeispiele

Bei technischen Fragen oder Problemen mit den Codebeispielen senden Sie bitte eine E-Mail an bookquestions@oreilly.com.

Dieses Buch ist dazu gedacht, Ihnen bei der Erledigung Ihrer Arbeit zu helfen. Im Allgemeinen dürfen Sie den Code in diesem Buch in Ihren eigenen Programmen oder Dokumentationen verwenden. Solange Sie den Code nicht in großem Umfang reproduzieren, brauchen Sie uns nicht um Erlaubnis zu bitten. Der Verkauf oder Vertrieb von Beispielen aus O'Reilly-Büchern ist dagegen genehmigungspflichtig. Signifikante Teile von Beispielcode aus diesem Buch für die eigene Produktdokumentation zu verwenden, ist genehmigungspflichtig.

Wir freuen uns über eine Quellenangabe, verlangen sie aber nicht unbedingt. Zu einer Quellenangabe gehören normalerweise Autor, Titel, Verlagsangabe, Veröffentlichungsjahr und ISBN, hier also: »*Mastering the Lightning Network* by Andreas M. Antonopoulos, Olaoluwa Osuntokun, and René Pickhardt (O'Reilly). Copyright 2022 aantonop Books LLC, René Pickhardt, and uuddlrllrbas LLC, ISBN 978-1-492-05486-3«.

Mastering the Lightning Network wird unter der Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 International License (CC BY-NC-ND 4.0) angeboten.

Sollten Sie befürchten, dass Ihre Verwendung der Codebeispiele gegen das Fairnessprinzip oder die Genehmigungspflicht verstoßen könnte, nehmen Sie bitte unter permissions@oreilly.com Kontakt mit uns auf.

Hinweise auf Unternehmen und Produkte

Alle Hinweise auf Unternehmen oder Produkte dienen der Information, Demonstration oder Referenz. Die Autoren unterstützen keines der genannten Unternehmen oder Produkte. Der Einsatz und die Sicherheit der in diesem Buch vorgestellten Produkte, Projekte oder Codefragmente wurden nicht getestet. Deren Nutzung erfolgt auf eigene Gefahr!

Adressen und Transaktionen in diesem Buch

Die Bitcoin-Adressen, Transaktionen, Schlüssel, QR-Codes und Blockchain-Daten in diesem Buch sind größtenteils echt. Sie können also die Blockchain durchgehen, sich die in den Beispielen enthaltenen Transaktionen genau ansehen und sie mit Ihren eigenen Skripten/Programmen abrufen.

Beachten Sie aber, dass die in diesem Buch zur Generierung von Adressen verwendeten privaten Schlüssel »verbrannt« wurden. Wenn Sie also Geld an diese Adressen senden, ist es für immer verloren, oder es kann von jedem abgeschöpft werden, der die hier abgedruckten privaten Schlüssel kennt.



Bitte senden Sie keinesfalls Geld an irgendeine der in diesem Buch verwendeten Adressen! Ihr Geld landet bei einem anderen Leser oder ist für immer verloren.

Andreas kontaktieren

Sie erreichen Andreas M. Antonopoulos über seine persönliche Website:

<https://aantonop.com>

Folgen Sie Andreas' Kanal auf YouTube:

<https://www.youtube.com/aantonop>

Folgen Sie Andreas' Seite auf Facebook:

<https://www.facebook.com/AndreasMAntonopoulos>

Folgen Sie Andreas auf Twitter:

<https://twitter.com/aantonop>

Folgen Sie Andreas auf LinkedIn:

<https://linkedin.com/company/aantonop>

Andreas möchte sich auch bei allen Förderern bedanken, die seine Arbeit durch monatliche Spenden unterstützen. Sie können Andreas auf Patreon unterstützen unter <https://patreon.com/aantonop>.

René kontaktieren

Sie erreichen René Pickhardt über seine persönliche Website:

<https://ln.rene-pickhardt.de>

Folgen Sie Renés Kanal auf YouTube:

<https://www.youtube.com/user/RenePickhardt>

Folgen Sie René auf Twitter:

<https://twitter.com/renepickhardt>

Folgen Sie René auf LinkedIn:

<https://www.linkedin.com/in/rene-pickhardt-80313744>

René möchte sich ebenfalls bei allen Förderern bedanken, die seine Arbeit durch eine monatliche Spende unterstützen. Sie können René auf Patreon unterstützen unter <https://patreon.com/renepickhardt>.

Sie können seine Arbeit aber auch direkt unter <https://donate.ln.rene-pickhardt.de> über das Lightning-Netzwerk in Bitcoin unterstützen. Dafür ist René ebenso dankbar wie seinen Förderern bei Patreon.

Olaoluwa Osuntokun kontaktieren

Sie erreichen Olaoluwa Osuntokun über seine berufliche E-Mail-Adresse:
laolu@lightning.engineering

Folgen Sie Olaoluwa auf Twitter:
<https://twitter.com/roasbeef>

Danksagungen von Andreas

Meine Liebe zu Wörtern und Büchern verdanke ich meiner Mutter Theresa, die mich in einem Haus aufzog, in dem Bücher jede Wand mit Beschlag belegten. Meine Mutter kaufte mir 1982 auch meinen ersten Computer, obwohl sie sich selbst als »technophob« beschrieb. Mein Vater Menelaos, ein Bauingenieur, der sein erstes Buch mit 80 veröffentlichte, lehrte mich logisches und analytisches Denken und weckte meine Leidenschaft zu Wissenschaft und Technik.

Ich danke euch allen für eure Unterstützung während meiner Reise.

Danksagungen von René

Ich möchte dem deutschen Bildungssystem danken, dem ich das Wissen verdanke, auf dem meine Arbeit aufbaut. Es ist eines der größten mir gemachten Geschenke. Ebenso möchte ich dem deutschen Gesundheitswesen danken und allen Menschen, die in diesem Bereich arbeiten. Ihr Einsatz und ihr Durchhaltevermögen machen sie zu meinen persönlichen Helden, und ich werde nie die Hilfe, Aufmerksamkeit und Unterstützung vergessen, die mir zuteilwurde, als ich sie benötigte. Mein Dank geht an all die Studenten, denen ich etwas lehren durfte und die sich in interessanten Diskussionen und Fragen engagierten. Von ihnen habe ich das meiste gelernt. Ich bin auch der Bitcoin- und Lightning-Netzwerk-Community dankbar, die mich freundlich willkommen hieß, sowie den Enthusiasten und Privatpersonen, die meine Arbeit finanziell unterstützten und das auch weiterhin tun. Ich danke ganz besonders allen Open-Source-Entwicklern (nicht nur Bitcoin und dem Lightning-Netzwerk) und den Menschen, die sie finanzieren, um diese Technik möglich zu machen. Ein besonderer Dank an meine Mitautoren, die den Weg mit mir gegangen sind. Und nicht zuletzt danke ich meinen Liebsten.

Danksagungen von Olaoluwa Osuntokun

Ich möchte dem großartigen Team von Lightning Labs danken, ohne die es kein LND gäbe. Ich möchte auch der ursprünglichen Gruppe von Autoren der BOLT-Spezifikation danken: Rusty Russell, Fabrice Drouin, Conner Frommkchet, Pierre-Marie Padiou, Lisa Neigut und Christian Decker. Nicht zuletzt möchte ich Joseph Poon und Tadge Dryja danken, den Autoren des ursprünglichen Lightning-Netzwerk-Papers, ohne die es kein Lightning-Netzwerk gäbe, über das man ein Buch schreiben kann.

Beitragende

Viele Beitragende lieferten Kommentare, Korrekturen und Ergänzungen zu diesem Buch, als es gemeinschaftlich auf GitHub geschrieben wurde.

Nachfolgend eine alphabetisch sortierte Liste aller GitHub-Beitragenden mit deren GitHub-IDs in Klammern:

- 8go (@8go)
- Aaqil Aziz (@batmanscode)
- Alexander Gnip (@quantumctulhu)
- Alpha Q. Smith (@alpha_github_id)
- Ben Skee (@benskee)
- Brian L. McMichael (@brianmcmichael)
- CandleHater (@CandleHater)
- Daniel Gockel (@dancodery)
- Dapeng Li (@luislee818)
- Darius E. Parvin (@DariusParvin)
- Doru Muntean (@chriton)
- Eduardo Lima III (@elima-iii)
- Emilio Norrmann (@enorrmann)
- Francisco Calderón (@grunch)
- Francisco Requena (@FrankyFFV)
- François Degros (@fdegros)
- Giovanni Zotta (@GiovanniZotta)
- Gustavo Silva (@GustavoRSSilva)
- Guy Thayakorn (@saguywalker)
- Haoyu Lin (@HAOYUatHZ)
- Hatim Boufnichel (@boufni95)
- Imran Lorgat (@ImranLorgat)

- Jeffrey McLarty (@jnmclarty)
- John Davies (@tigeryant)
- Julien Wendling (@trigger67)
- Jussi Tiira (@juhi24)
- Kory Newton (@korynewton)
- Lawrence Webber (@lwebbz)
- Luigi (@gin)
- Maximilian Karasz (@mknoszlig)
- Omega X. Last (@omega_github_id)
- Owen Gunden (@ogunden)
- Patrick Lemke (@PatrickLemke)
- Paul Wackerow (@wackerow)
- Randy McMillan (@RandyMcMillan)
- René Köhnke (@rene78)
- Ricardo Marques (@RicardoM17)
- Sebastian Falbesoner (@theStack)
- Sergei Tikhomirov (@s-tikhomirov)
- Severin Alexander Bühler (@SeverinAlexB)
- Simone Bovi (@SimoneBovi)
- Srijan Bhushan (@srijanb)
- Taylor Masterson (@tjmasterson)
- Umar Bolatov (@bolatovumar)
- Warren Wan (@wlwanpan)
- Yibin Zhang (@z4y1b2)
- Zachary Haddenham (@senf42)

Ohne die Hilfe der oben aufgeführten Personen wäre dieses Buch nicht möglich gewesen. Eure Beiträge demonstrieren die Kraft von Open Source und einer offenen Kultur, und wir sind euch unendlich dankbar.

Vielen Dank.

Quellen

Ein Teil des Materials in diesem Buch stammt aus unterschiedlichen Public-Domain- bzw. Open-License-Quellen. Für andere Teile wurden Genehmigungen erteilt. Details zu Quellen, Lizenzen und Quellenzuordnungen finden Sie in Anhang D.

Vorwort zur deutschen Ausgabe

Als Satoshi Nakamoto Bitcoin 2009 veröffentlichte, dauerte es nicht lange, bis sich eine kleine Gruppe technisch Begeisterter zusammenfand, um die Vor- und Nachteile des neuen Systems zu diskutieren und damit herumzuspielen. Ich gehöre selbst auch zu den Leuten, die eher zufällig auf Bitcoin gestoßen sind, aber seitdem ich Bitcoin entdeckt habe, lässt es mich nicht mehr los.

Das Ziel, das sich Satoshi gesetzt hatte, war, dass Bitcoin ein globales Wertetransfer-Netzwerk werden sollte, ohne Mittelsmänner. In diesem Netzwerk sollten Werte in Form von Bitcoins beliebig hin und her verschoben werden können. Das Ganze durch pseudonyme Adressen ergänzt, um die Privatsphäre der Teilnehmer zu schützen.

Anfangs waren es noch wenige, die eher aus technischem Interesse mitmachten, denn damals wurden Bitcoins noch kein Wert zugeschrieben. Bereits zu diesem Zeitpunkt war aber schon klar, dass, sollte Bitcoin erfolgreich sein, wir schnell an die Grenzen des damaligen Systems stoßen würden. Deshalb mussten neue Ideen her.

Die wahrscheinlich schwierigste Frage war, wie Bitcoin denn skalieren könnte. Wie sollte Bitcoin sich also bei steigender Nachfrage anpassen, um allen Menschen die Möglichkeit zu geben, Bitcoin zu nutzen. Das Problem dabei war nämlich, dass die Blockchain, die von Satoshi für Bitcoin erfundene Technologie, ein geteiltes Medium darstellt, allerdings mit begrenzter Kapazität für das Bearbeiten von Transaktionen. Diese Frage stellten wir uns schon sehr früh, und 2012 war mir klar, dass ich meine Forschung im Rahmen meines Doktorats diesem Thema widmen würde.

Eine potenzielle Lösung waren sogenannte Micropayment Channels. Bei Micropayment Channels, auch Off-Chain-Protokolle genannt, handelt es sich um Systeme, die auf einer Blockchain aufbauen, um deren Nutzen zu erweitern. Bei Off-Chain-Protokollen werden Änderungen nicht mehr dem gesamten Netzwerk mitgeteilt, sondern zunächst nur den Teilnehmern, die an einer Transaktion beteiligt sind. Das gesamte Netzwerk wird erst später informiert. Die Saldierung aller stattfindenden Off-Chain-Transfers hat den Vorteil, dass am Ende nur eine einzige

Transaktion bestätigt werden muss. Und da für diese Bestätigung immer On-Chain-Gebühren anfallen, ist es viel kostengünstiger, wenn eine On-Chain-Gebühr auf beliebig viele Off-Chain-Transfers verteilt werden kann.

Was anfangs noch sehr abstrakt war, wurde durch Experimente in dieser Richtung immer konkreter: angefangen mit den einfachen unidirektionalen Channels von Matt Corallo und Jeremy Spillmann bis hin zu dem Lightning Network Paper von Joseph Poon und Tadge Dryja.

Als Joseph und Tadge 2015 das Lightning Network Paper publizierten, stieß es auf großes Interesse, gerade bei denjenigen von uns, denen der Mangel einer plausiblen Lösung für die Skalierung von Bitcoin unter den Nägeln brannte. Aber Skalierbarkeit sollte nicht der einzige Vorteil des Lightning Network sein, hinzu kommt auch, dass es in Echtzeit Zahlungen ermöglicht und die Privatsphäre potenziell besser schützen kann, weil nicht mehr alles bis in alle Ewigkeit auf der Blockchain gespeichert wird.

Doch die Vorteile brachten wiederum einiges an Komplexität und neuen Konzepten mit sich, die die Nutzer der Technologie erst einmal kennenlernen und verstehen müssen. Beim Lightning-Netzwerk muss man sowohl Bitcoin als auch die Konzepte hinter Lightning verstehen, wodurch der Einstieg alles andere als einfach ist.

Hinzu kommt, dass das Paper an sich sehr komplex ist und eher ein abstraktes Bild als ein voll funktionsfähiges System beschreibt. Es fehlten alle technischen Details, bis auf das Grundgerüst in Form der Bitcoin-Transaktionen.

So passierte erst mal nichts, bis Rusty Russell von Blockstream sich des Problems annahm und anfang, die fehlenden Stellen auszuschmücken, denn eines war uns klar, dieses System musste unbedingt real werden. Kurz darauf fingen dann auch die Kollegen von Acinq und Lightning Labs an, eine Implementierung zu bauen. Im Herbst 2016 entschieden die drei Teams dann zusammenzuarbeiten, um ein einziges großes Netz zu bauen, in dem jeder jedem anderen Teilnehmer Bitcoins senden konnte: schnell und anonym. Und so fing die Lightning Network Specification an, ein Prozess, der bis heute anhält.

Dieses Buch wendet sich an alle, die die Hintergründe hinter dem Lightning Netzwerk interessieren, ob das nun ein Nutzer ist, eine Entwicklerin, die auf Lightning aufbauen will, oder vielleicht sogar ein zukünftiger Protokollentwickler. Es bildet das fehlende Bindeglied zwischen dem Paper von 2015 und dem Protokoll, so wie es heute aktiv im Netzwerk genutzt wird, und die Leserinnen und Leser werden langsam und schrittweise an das komplexe Thema herangeführt.

Bei den Autoren handelt es sich um echte Heavyweights der Bitcoin Community. Andreas Antonopoulos ist langjähriger Speaker und hat die unglaublich wertvolle Fähigkeit, komplexe Themen anschaulich zu vermitteln. Olaoluwa Osuntokun, Entwickler der ersten Stunde, bringt die praktische Erfahrung und das notwendige Detailwissen mit und René Pickhardt den theoretischen Hintergrund, um das Protokoll abstrakt zu behandeln.

Ein solches Buch zu schreiben, ist nicht einfach, denn es handelt sich beim Lightning-Netzwerk um ein bewegliches Ziel, und so beleuchtet das Buch sowohl Konzepte als auch deren konkrete Umsetzungen.

Es handelt sich bei diesem Buch also um ein Nachschlagewerk, über das ich mich damals, als wir das Protokoll entwickelt haben, sehr gefreut hätte.

*Dr. Christian Decker
Researcher, Blockstream*

Das Lightning-Netzwerk verstehen

Eine Übersicht über das Lightning-Netzwerk für jeden, der die grundlegenden Konzepte und die Nutzung des Lightning-Netzwerks verstehen will.

