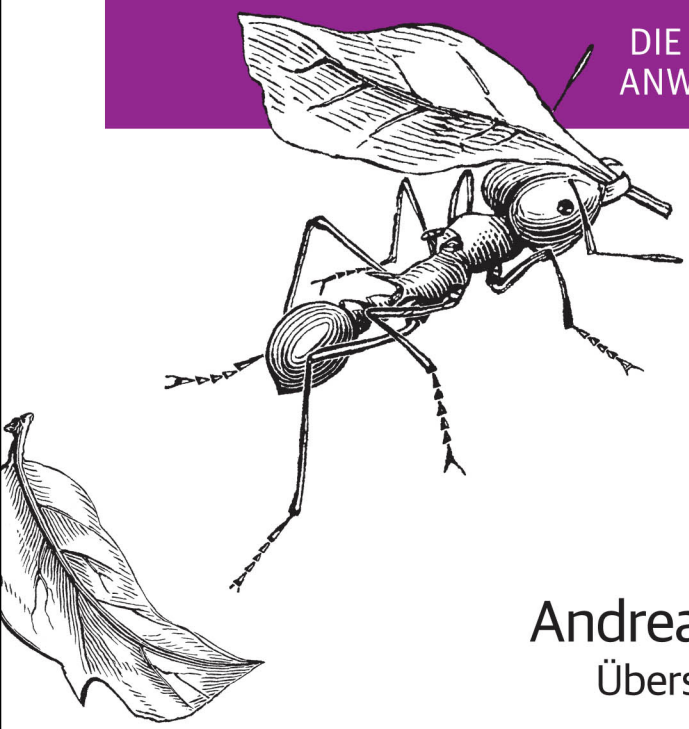
O'REILLY®

2. Auflage
US-Bestseller

Bitcoin & Blockchain

Grundlagen und Programmierung

DIE BLOCKCHAIN VERSTEHEN
ANWENDUNGEN ENTWICKELN



Andreas M. Antonopoulos
Übersetzung von Peter Klicman

2. AUFLAGE

Bitcoin und Blockchain – Grundlagen und Programmierung

Die Blockchain verstehen, Anwendungen entwickeln

Andreas M. Antonopoulos

Deutsche Übersetzung von
Peter Klicman

O'REILLY®

Andreas M. Antonopoulos

Lektorat: Ariane Hesse
Übersetzung: Peter Klicman
Korrektorat: Sibylle Feldmann, www.richtiger-text.de
Herstellung: Susanne Bröckelmann
Umschlaggestaltung: Michael Oréal, www.oreal.de
Satz: III-satz, www.drei-satz.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, mediaprint-druckerei.de

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: 978-3-96009-071-7
PDF: 978-3-96010-171-0
ePub: 978-3-96010-172-7
mobi: 978-3-96010-173-4

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.

2. Auflage 2018

Wieblinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of Mastering Bitcoin, 2nd Edition (Second Release: 2017-07-21), ISBN 9781491954386 © 2017 Andreas M. Antonopoulos, 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.

5 4 3 2 1 0

Dedicated to my mum, Theresa (1946–2017)
She taught me to love books and question authority
Thank you, mum

Inhalt

Vorwort

Ein Bitcoin-Buch schreiben

Mitte 2011 stolperte ich das erste Mal über Bitcoin. Meine erste Reaktion war: »Pfft! Nerd-Geld!«, und ich ignorierte es für weitere sechs Monate, ohne seine Bedeutung zu erkennen. Diese Reaktion habe ich bei vielen der klügsten Menschen, die ich kenne, beobachtet, was mich ein bisschen tröstet. Als ich in einer Mailinglistendiskussion das zweite Mal über Bitcoin stolperte, entschied ich mich, das Whitepaper von Satoshi Nakamoto zu lesen, um die maßgebliche Quelle zu studieren und herauszufinden, worum es denn da eigentlich ging. Ich erinnere mich immer noch an den Moment, als ich diese neun Seiten gelesen hatte und begriff, dass Bitcoin nicht einfach eine digitale Währung, sondern ein Vertrauensnetzwerk ist, das die Basis für weit mehr als nur Währungen sein konnte. Die Erkenntnis, dass das »kein Geld, sondern ein dezentralisiertes Vertrauensnetzwerk« ist, schickte mich auf eine viermonatige Reise, in der ich jedes Quäntchen an Informationen über Bitcoin, das ich finden konnte, aufsaugte. Es hatte mich gepackt, und wie besessen verbrachte ich täglich zwölf Stunden und mehr vor dem Bildschirm, in denen ich las, schrieb, programmierte und so viel lernte, wie ich konnte. Nachdem ich aus diesem Zustand wieder erwachte, war ich zehn Kilogramm leichter und hatte mich entschieden, zukünftig an Bitcoin zu arbeiten.

Zwei Jahre später, nachdem ich eine Reihe kleiner Start-ups gegründet hatte, um verschiedene Bitcoin-bezogene Dienste und Produkte zu erforschen, entschied ich, dass es an der Zeit wäre, mein erstes Buch zu schreiben. Bitcoin hatte mich in einen Kreativitätsrausch versetzt und meine Gedanken bestimmt. Das war die aufregendste Technologie, der ich seit Beginn des Internets begegnet war – Zeit also, meine Leidenschaft für diese faszinierende Technologie mit einem breiteren Publikum zu teilen.

Leserkreis

Dieses Buch richtet sich hauptsächlich an Entwickler. Wenn Sie eine Programmiersprache beherrschen, lehrt Sie dieses Buch, wie kryptografische Währungen funktionieren, wie man sie nutzt und wie man Software entwickelt, die mit ihnen

arbeitet. Die ersten Kapitel eignen sich ebenfalls als ausführliche Einführung in Bitcoin für Nichtprogrammierer, also für diejenigen, die die innere Funktionsweise von Bitcoin und Kryptowährungen verstehen wollen.

Warum sind Ameisen auf dem Cover?

Die Blattschneiderameise ist eine Spezies, die in einem Kolonie-Superorganismus ein hochkomplexes Verhalten zeigt. Doch jede einzelne Ameise agiert nach einem Satz einfacher Regeln, die durch soziale Interaktion und das Ausschütten chemischer Duftstoffe (Pheromone) gesteuert wird. Laut (englischer) Wikipedia bilden Blattschneiderameisen nach dem Menschen die größten und komplexesten Tiergesellschaften. Blattschneiderameisen essen keine Blätter, vielmehr nutzen sie sie, um einen Pilz anzubauen, der die zentrale Futterquelle der Kolonie bildet. Diese Ameisen betreiben also Landwirtschaft!

Zwar bilden Ameisen eine kastenbasierte Gesellschaft und haben eine Königin, die für den Nachwuchs sorgt, doch es gibt weder eine zentrale Autorität noch einen Anführer. Das hochgradig intelligente und komplexe Verhalten, das eine aus mehreren Millionen Ameisen bestehende Kolonie zeigt, ist eine emergente Eigenschaft der Interaktion von Individuen in einem sozialen Netzwerk.

Die Natur demonstriert, dass ein dezentralisiertes System robust, komplex und unglaublich ausgereift sein kann, ohne eine zentrale Autorität, eine Hierarchie oder komplexe Teile zu benötigen.

Bitcoin ist ein kunstvolles dezentralisiertes Vertrauensnetzwerk, das eine Vielzahl finanzieller Prozesse unterstützen kann. Dennoch folgt jeder Knoten im Bitcoin-Netzwerk nur einigen wenigen einfachen mathematischen Regeln. Die Interaktion zwischen vielen Knoten führt zu diesem ausgeklügelten Verhalten, nicht die Komplexität eines einzelnen Knotens oder das in ihn gesetzte Vertrauen. Wie eine Ameisenkolonie ist das Bitcoin-Netzwerk ein robustes Netzwerk einfacher Knoten, die einfachen Regeln folgen, um erstaunliche Dinge ohne zentrale Koordinierung zu erreichen.

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 verwendet, die Sie wortwörtlich eingeben müssen.

`Nichtproportionalschrift kursiv`
: Wird für Text verwendet, der durch benutzereigene oder durch den Kontext bestimmte Werte ersetzt wird, und für die Kommentare in Listings, um eine bessere Lesbarkeit zu gewährleisten..

Mit diesem Symbol wird ein Tipp oder ein Vorschlag angezeigt.

Mit diesem Symbol wird ein allgemeiner Hinweis angezeigt.

Hiermit wird eine Warnung angezeigt.

Codebeispiele

Die Beispiele sind in Python bzw. C++ geschrieben und verwenden die Kommandozeile Unix-artiger Betriebssysteme wie Linux oder macOS. Alle Code-Snippets finden Sie im Github-Repository (*https://github.com/bitcoinbook/bitcoinbook*) im *code*-Unterverzeichnis des Main-Repository. Laden Sie sich den Buchcode herunter, probieren Sie die Codebeispiele aus oder senden Sie Korrekturen über GitHub.

Alle Code-Snippets können für die meisten Betriebssysteme mit einer minimalen Installation der Compiler und Interpreter für die entsprechenden Sprachen repliziert werden. Wenn nötig, stellen wir grundlegende Installationsanweisungen und schrittweise Beispiele der 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 den Beispielen arbeiten, sollten Sie diese 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. Die privaten Schlüssel und die zugehörigen öffentlichen Schlüssel etwa sind alle echt. Sämtliche Beispieltransaktionen,

Blöcke und Blockchain-Referenzen wurden in die Blockchain eingetragen und sind Teil des öffentlichen »Kassenbuchs«, d.h., man kann sie sich auf jedem Bitcoin-System ansehen.

Verwendung der Codebeispiele

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. Zum Beispiel benötigen Sie nicht unsere Erlaubnis, wenn Sie ein Programm unter Zuhilfenahme mehrerer Codestücke aus diesem Buch schreiben. Eine Frage mit einem Zitat oder einem Codebeispiel aus dem Buch zu beantworten, erfordert ebenfalls keine Genehmigung. Signifikante Teile von Beispielcode aus dem Buch für die eigene Produktdokumentation zu verwerten, ist dagegen 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: »Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, Inc. 2017, ISBN 978-1-491-95438-6«.

Einige Auflagen dieses Buchs werden unter einer Open-Source-Lizenz wie CC-BY-NC (*https://creativecommons.org/licenses/by-nc/4.0/*) angeboten. In diesem Fall gelten die Bedingungen dieser Lizenz.

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.

Bitcoin-Adressen und -Transaktionen in diesem Buch

Die Bitcoin-Adressen, Transaktionen, Schlüssel, QR-Codes und Blockchain-Daten in diesem Buch sind größtenteils real. Das bedeutet, dass Sie die Blockchain durchgehen und den größten Teil real nachverfolgen können. Sie können also die Blockchain durchsuchen, 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 entweder in diesem Buch abgedruckt oder »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.

Den Autor kontaktieren

Sie erreichen mich, Andreas M. Antonopoulos, über meine persönliche Website: *https://antonopoulos.com/*

Informationen zu *Mastering Bitcoin*, zur Open Edition und zu Übersetzungen finden Sie hier: *https://bitcoinbook.info/*

Folgen Sie mir auf Facebook: *https://facebook.com/AndreasMAntonopoulos*

Folgen Sie mir auf Twitter: *https://twitter.com/aantonop*

Folgen Sie mir auf LinkedIn: *https://linkedin.com/company/aantonop*

Mein herzlicher Dank an alle meine Förderer, die meine Arbeit durch monatliche Spenden unterstützen. Meine Patreon-Page finden Sie hier: *https://patreon.com/aantonop*

Danksagungen

Dieses Buch spiegelt die Bemühungen und Beiträge vieler Menschen wider. Ich bin sehr dankbar für die Hilfe, die ich von Freunden, Kollegen, aber auch völlig Fremden erhalten habe, die mich dabei unterstützt haben, diesen technischen Leitfaden zu Kryptowährungen und Bitcoin zu schreiben.

Es ist unmöglich, zwischen der Bitcoin-Technologie und der Bitcoin-Community zu unterscheiden, und dieses Buch ist ebenso ein Produkt dieser Community wie ein Buch über die Technologie. Meine Arbeit an diesem Buch wurde vom Anfang bis zum Ende von der Community befürwortet, angefeuert und unterstützt. Neben vielem anderen ermöglichte mir dieses Buch, über zwei Jahre Teil dieser wundervollen Community zu sein, und ich bin mehr als dankbar, in dieser Community akzeptiert worden zu sein. Eine große Menge an Menschen haben das Buch beeinflusst, und es sind viel zu viele, um sie beim Namen zu nennen. Es sind Menschen, die ich auf Konferenzen, Events, Seminaren, Meet-ups, beim Pizza-Plausch oder bei privaten Treffen kennengelernt habe, ebenso wie bei Twitter, auf reddit, bitcointalk.org und GitHub. Jede Idee, Analogie, Frage, Antwort und Erläuterung in diesem Buch wurde an irgendeinem Punkt durch die Community inspiriert, getestet und verbessert. Ich danke euch allen für die Unterstützung. Ohne euch hätte es dieses Buch nie gegeben, und ich bin euch für immer dankbar.

Der Weg zum Autor begann natürlich lange vor dem ersten Buch. Meine erste Sprache war Griechisch (und damit war auch mein erster Unterricht in Griechisch). Deshalb ich belegte im ersten Jahr an der Universität einen Schreibkurs. Ich danke meiner damaligen Lehrerin Diana Kordas, die mir in diesem Jahr dabei half, Selbstvertrauen und Fertigkeiten zu sammeln. Später schrieb ich für das *Network World Magazine* und entwickelte meine Fertigkeiten als technischer Autor im Bereich Data Center. Ich danke John Dix und John Gallant, die mir meinen ersten Job als Kolumnist bei *Network World* gaben, sowie meinem Lektor Michael Cooney und meinem

Kollegen Johna Till Johnson, die meine Kolumnen lektorierten und für eine Veröffentlichung aufbereiteten. Vier Jahre lang 500 Wörter pro Woche zu schreiben, sorgten für ausreichend Erfahrung, um ernsthaft über ein Dasein als Autor nachzudenken.

Vielen Dank auch an diejenigen, die mich unterstützten, nachdem ich meinen Buchvorschlag bei O'Reilly eingereicht hatte, indem sie Empfehlungen aussprachen und sich den Entwurf genauer ansahen. Mein Dank geht an John Gallant, Gregory Ness, Richard Stiennon, Joel Snyder, Adam B. Levine, Sandra Gittlen, John Dix, Johna Till Johnson, Roger Ver und Jon Matonis. Besonderer Dank geht an Richard Kagan und Tymon Mattoszko, die frühe Fassungen prüften, und an Matthew Taylor, der diese Fassung lektorierte.

Dank an Cricket Liu, Autor des O'Reilly-Titels *DNS and BIND*, der mich bei O'Reilly vorgestellt hat. Ein Dank auch an Michael Loukides und Allyson MacDonald von O'Reilly, die Monate daran arbeiteten, dass dieses Buch Wirklichkeit wurde. Allyson war besonders aufmerksam, wenn Abgabefristen verstrichen und Ergebnisse fehlten. Bei der zweiten Ausgabe gab Timothy McGovern die Richtung vor, Kim Cofer übernahm das Lektorat, und Rebecca Panzer sorgte für viele neue Diagramme.

Die ersten Entwürfe der ersten Kapitel waren die schwersten, schlicht weil Bitcoin ein kompliziertes Thema ist. Sobald ich einen Aspekt herauspickte, musste ich direkt schon wieder das große Ganze betrachten. Wiederholt blieb ich hängen und war frustriert, wenn ich versuchte, ein Thema leicht verständlich rüberzubringen, indem ich eine Geschichte um ein schwieriges technisches Thema herum erzählen wollte. Letztendlich entschied ich mich dafür, die Geschichte des Bitcoins über die Geschichten derjenigen zu erzählen, die Bitcoins nutzen. Das Buch zu schreiben, wurde dadurch erheblich einfacher. Ich schulde meinem Freund und Mentor Richard Kagan Dank, der mir dabei half, die Geschichte zu entwirren und meine Schreibblockaden zu überwinden. Ich danke Pamela Morgan, die frühe Fassungen jedes Kapitels der ersten und zweiten Auflage Korrektur las und die richtigen Fragen stellte. Mein Dank geht auch an die Entwickler der »San Francisco Bitcoin Developers Meetup«-Gruppe sowie an Taariq Lewis und Denise Terry, die dabei halfen, das frühe Material zu testen. Dank ebenfalls an Andrew Naugler für den Entwurf der Infografiken.

Während ich das Buch schrieb, machte ich frühe Fassungen auf GitHub verfügbar und lud dazu ein, diese zu kommentieren. Über 100 Kommentare, Vorschläge, Korrekturen und Beiträge sind daraufhin eingegangen. Für diese Beiträge bedanke ich mich explizit in »Early Release Draft (GitHub-Beiträge)« auf Seite XXI. Zuallererst gilt mein Dank meinen freiwilligen GitHub-Lektoren Ming T. Nguyen (erste Auflage) und Will Binns (zweite Auflage), die auf GitHub unermüdlich Pull-Requests kuratiert, verwaltet und aufgelöst, Reports veröffentlicht und Bug-Fixes vorgenommen haben.

Sobald die erste Fassung stand, wurde sie mehrfach von technischen Korrektoren überarbeitet. Vielen Dank an Cricket Liu und Lorne Lantz für deren sorgfältiges Korrekturlesen sowie ihre Kommentare und die Unterstützung.

Verschiedene Bitcoin-Entwickler steuerten Codebeispiele, Korrekturen und Kommentare bei. Dank an Amir Taaki und Eric Voskuil für Beispielcode und viele gute Kommentare, Chris Kleeschulte für den Bitcore-Anhang, Vitalik Buterin und Richard Kiss für Codebeiträge und ihre Hilfe bei der Mathematik elliptischer Kurven, Gavin Andresen für Korrekturen und Kommentare, Michalis Kargakis für Kommentare und Beiträge sowie Robin Inge für Fehlerkorrekturen der zweiten Auflage. Auch bei der zweiten Auflage erhielt ich wieder Hilfe von vielen Bitcoin-Core-Entwicklern, darunter Eric Lombrozo, der Segregated Witness entmystifizierte, Luke Dashjr, der mir beim Kapitel über Transaktionen half, Johnson Lau, der (unter anderem) das Kapitel zu Segregated Witness Korrektur las, und viele andere. Ich danke Joseph Poon, Tadge Dryja und Olaoluwa Osuntokun, die Lightning Networks erklärten, meinen Text Korrektur lasen und Fragen beantworteten, wenn ich nicht weiterkam.

Meine Liebe für Wörter und Bücher 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 im Alter von 80 Jahren veröffentlichte, lehrte mich logisches und analytisches Denken und schürte meine Vorliebe für Wissenschaft und Technik.

Ich danke euch allen für eure Unterstützung während meiner Reise.

Early Release Draft (GitHub-Beiträge)

Viele Beitragende lieferten Kommentare, Korrekturen und Ergänzungen zum Early Release Draft auf GitHub. Ich danke euch allen für euren Beitrag zu diesem Buch.

Nachfolgend eine Liste wichtiger GitHub-Beitragender mit deren GitHub-IDs in Klammern:

- Alex Waters (alexwaters)
- Andrew Donald Kennedy (grkvlt)
- bitcoinctf
- Bryan Gmyrek (physicsdude)
- Casey Flynn (cflynn07)
- Chapman Shoop (belovachap)
- Christie D'Anna (avocadobreath)
- Cody Scott (Siecje)
- coinradar
- Cragin Godley (cgodley)
- dallyshalla
- Diego Viola (diegoviola)
- Dirk Jäckel (biafra23)
- Dimitris Tsapakidis (dimitris-t)
- Dmitry Marakasov (AMDmi3)
- drstrangeM
- Ed Eykholt (edeykholt)
- Ed Leafe (EdLeafe)

- Edward Posnak (edposnak)
- Elias Rodrigues (elias19r)
- Eric Voskuil (evoskuil)
- Eric Winchell (winchell)
- Erik Wahlström (erikwam)
- effectsToCause (vericoin)
- Esteban Ordano (eordano)
- ethers
- fabienhinault
- Frank Höger (francyi)
- Gaurav Rana (bitcoinsSG)
- genjix
- halseth
- Holger Schinzel (schinzelh)
- Ioannis Cherouvim (cherouvim)
- Ish Ot Jr. (ishotjr)
- James Addison (jayaddison)
- Jameson Lopp (jlopp)
- Jason Bisterfeldt (jbisterfeldt)
- Javier Rojas (fjrojasgarcia)
- Jeremy Bokobza (bokobza)
- JerJohn15
- Joe Bauers (joebauers)
- joflynn
- Johnson Lau (jl2012)
- Jonathan Cross (jonathancross)
- Jorgeminator
- Kai Bakker (kaibakker)
- Mai-Hsuan Chia (mhchia)
- Marzig (marzig76)
- Maximilian Reichel (phramz)
- Michalis Kargakis (kargakis)
- Michael C. Ippolito (michaelcippolito)
- Mihail Russu (MihailRussu)
- Minh T. Nguyen (enderminh)
- Nagaraj Hubli (nagarajhubli)
- Nekomata (nekomata-3)
- Robert Furse (Rfurse)
- Richard Kiss (richardkiss)
- Ruben Alexander (hizzvizz)
- Sam Ritchie (sritchie)
- Sergej Kotliar (ziggamon)
- Seiichi Uchida (topecongiro)
- Simon de la Rouviere (simondlr)
- Stephan Oeste (Emzy)
- takaya-imai
- Thiago Arrais (thiagoarrais)
- venzen
- Will Binns (wbnns)
- wintercooled
- wjx
- Wojciech Langiewicz (wlk)
- yurigeorgiev4

Glossar

Dieses Glossar umfasst viele der im Zusammenhang mit Bitcoin und Blockchain verwendeten Begriffe. Die Begriffe kommen im gesamten Buch ständig vor.

Adresse
: Eine Bitcoin-Adresse ist ein aus Buchstaben und Ziffern bestehender String wie beispielsweise 1DSrfJdB2AnWaFNgSbv3MZC2m74996JafV. Eigentlich handelt es sich um die base58check-codierte Version eines 160-Bit-Public-Key-Hash. Genau wie Sie jemanden bitten, Ihnen eine E-Mail an Ihre E-Mail-Adresse zu schicken, senden Ihnen andere Bitcoins an Ihre Bitcoin-Adressen.

Belohnung (Reward)
: In jedem neuen Block enthaltene Menge an Bitcoin, die der Miner erhält, der die Lösung für den Proof-of-Work gefunden hat, momentan (März 2018) 12,5 BTC pro Block.

Bestätigungen
: Ist eine Transaktion in einen Block aufgenommen worden, hat sie ihre erste Bestätigung erhalten. Sobald ein weiterer Block geschürft wird, hat die Transaktion zwei Bestätigungen und so weiter. Bei sechs oder mehr Bestätigungen gilt eine Transaktion als unumkehrbar.

BIP
: Bitcoin Improvement Proposals. Eine Reihe von Vorschlägen/Anträgen, die die Mitglieder der Bitcoin-Community eingereicht haben, um den Bitcoin zu verbessern. Zum Beispiel ist BIP-21 ein Vorschlag zur Verbesserung des Bitcoin-URI-Schemas (Uniform Resource Identifier).

Bitcoin
: Der Name einer Währungseinheit (des Coins), des Netzwerks und der Software.

Block
: Eine Gruppierung von Transaktionen, die mit einem Zeitstempel und einem Fingerprint des vorherigen Blocks markiert sind. Der Block-Header wird gehasht, um den Proof-of-Work zu erzeugen und damit die Transaktionen zu

validieren. Gültige Blöcke werden der Haupt-Blockchain durch Netzwerkkonsens hinzugefügt.

Block, veralteter (stale)
: Erfolgreich geschürfter Block, der nicht in die momentan beste Blockchain aufgenommen wurde, weil ein anderer Block der gleichen Höhe die Chain als Erster erweitert hat.

Blockchain
: Eine Liste validierter Blöcke. Jeder Block ist (bis zurück zum Genesis-Block) mit seinem Vorgänger verlinkt.

byzantinischen Generäle, Problem der
: Ein zuverlässiges Computersystem muss mit Fehlern einer oder mehrerer seiner Komponenten umgehen können. Eine fehlerhafte Komponente kann ein häufig übersehenes Verhalten zeigen, nämlich das Senden widersprüchlicher Informationen an unterschiedliche Teile des Systems. Das Problem des Umgangs mit dieser Art Fehler wird abstrakt oft als das Problem der byzantinischen Generäle beschrieben.

Coinbase
: Ein spezielles Feld, das als Input für Coinbase-Transaktionen verwendet wird. Mit der Coinbase kann man seinen Anspruch auf die Blockbelohnung geltend machen und bis zu 100 Byte beliebige Daten eintragen. Nicht zu verwechseln mit der Coinbase-Transaktion.

Coinbase-Transaktion
: Die erste Transaktion eines Blocks. Sie wird immer von einem Miner erzeugt und enthält eine einzelne Coinbase. Nicht zu verwechseln mit Coinbase.

Cold Storage
: Das Offlinespeichern von Bitcoins (oder zumindest eines Teils davon). Cold Storage erreicht man durch die Erzeugung privater Schlüssel und deren Offlinespeicherung in einer sicheren Umgebung. Cold Storage ist für jeden wichtig, der Bitcoins hält. Ist Ihr Computer online, ist er für Hackerangriffe anfällig. Er sollte deshalb nicht zur Speicherung signifikanter Bitcoin-Mengen genutzt werden.

Colored Coins
: Ein Open-Source-Bitcoin-2.0-Protokoll, das es Entwicklern erlaubt, digitale Vermögenswerte (Assets) auf der Bitcoin-Blockchain aufzusetzen und so die Funktionalität des Bitcoins über eine Währung hinaus zu erweitern.

Difficulty
: Eine netzwerkweit gültige Einstellung, die festlegt, wie viel Rechenleistung nötig ist, um den Proof-of-Work zu berechnen.

Difficulty Retargeting
: Eine netzwerkweite Neuberechnung der Difficulty. Erfolgt einmal alle 2.016 Blöcke und berücksichtigt die Hashing-Leistung der vorangegangenen 2.016 Blöcke.

Difficulty Target
: Eine Difficulty, bei der alle Berechnungen im Netzwerk im Schnitt alle zehn Minuten einen Block finden.

Double-Spending
: Double-Spending bedeutet, Geld erfolgreich zweimal ausgeben zu können. Bitcoin schützt sich vor Double-Spending, indem es jede Transaktion verifiziert, die in die Blockchain aufgenommen wird. Dabei wird sichergestellt, dass die Inputs der Transaktion nicht bereits eingelöst wurden.

ECDSA
: Elliptic Curve Digital Signature Algorithm, kurz ECDSA, ist ein kryptografischer Algorithmus, über den Bitcoin sicherstellt, dass die Mittel nur von ihren rechtmäßigen Besitzern ausgegeben werden können.

Extra Nonce
: Als die Difficulty anstieg, sind Miner häufig alle 4 Milliarden möglichen Werte für die Nonce durchgegangen, ohne einen Block gefunden zu haben. Weil das Coinbase-Skript zwischen 2 und 100 Bytes Daten speichern kann, begannen die Miner damit, diesen Platz für eine zusätzliche Nonce zu nutzen und so einen wesentlich größeren Bereich von Block-Header-Werten nach einem gültigen Block abzusuchen.

Fork
: Ein Fork tritt ein, wenn zwei oder mehr Blöcke die gleiche Blockhöhe aufweisen, sodass sich die Blockchain »teilt« (engl. Fork). Das passiert üblicherweise, wenn zwei oder mehr Miner nahezu gleichzeitig einen Block finden. Kann auch als Teil eines Angriffs vorkommen.

Gebühren
: Der Sender einer Transaktion fügt eine Gebühr für die Verarbeitung seiner Transaktion hinzu.

Geheimer Schlüssel (Secret Key oder auch Private Key)
: Die geheime Zahl, die die Bitcoins freigibt, die an die dazugehörige Adresse gesendet wurden. Ein solcher geheimer Schlüssel ist z. B.: 5J76sF8L5jTtzE96r66Sf8cka9y44wdpJjMwCxR3tzLh3ibVPxh.

Genesis-Block
: Der erste Block in der Blockchain, der zur Initialisierung der Kryptowährung verwendet wurde.

Hard Fork
: Ein Hard Fork ist eine permanente Teilung der Blockchain. Tritt häufig ein, wenn veraltete Nodes ihre UTXO-Datenbank nicht mehr aktualisieren können.

Hardware-Wallet
: Eine Hardware-Wallet ist eine spezielle Bitcoin-Wallet, die die privaten Schlüssel des Nutzers auf einer sicheren Hardware speichert.

Hash
: Der digitale Fingerabdruck einer binären Eingabe.

Hashlock
: Ein Hashlock ist eine Art Schuld, die das Einlösen eines Outputs erst erlaubt, wenn bestimmte Daten öffentlich werden. Hashlocks haben die nützliche Eigenschaft, dass alle anderen Hashlocks, die über den gleichen Schlüssel gesichert sind, ebenfalls geöffnet werden können, sobald dieser Hashlock bekannt ist. Auf diese Weise lassen sich mehrere Outputs erzeugen, die durch den gleichen Hashlock geschützt sind und zur gleichen Zeit eingelöst werden können.

HD-Protokoll
: Das Hierarchical Deterministic (HD) Key Creation and Transfer Protocol (BIP-32). Erlaubt die hierarchische Erzeugung von Child-Schlüsseln aus einem einzigen Parent-Schlüssel.

HD-Wallet
: Wallets, die das HD-Protokoll (BIP-32) verwenden.

HD-Wallet-Seed
: HD-Wallet-Seed oder Root-Seed ist ein (potenziell kurzer) Wert, der als Seed-Wert zur Generierung des privaten Master-Keys und Master-Chain-Codes verwendet wird.

HTLC
: Ein Hashed Time Lock Contract, kurz HTLC, ist eine Klasse von Hashlocks und Timelocks nutzenden Zahlungen, bei der der Empfänger einer Zahlung diese vor einer festgelegten Frist durch eine kryptografische Zahlungsbestätigung bescheinigt. Anderenfalls kann die Zahlung nicht mehr an ihn überwiesen werden, und der Betrag geht an den Zahlungspflichtigen zurück.

Konsens
: Konsens besteht, wenn mehrere Nodes (üblicherweise ein Großteil der Nodes im Netzwerk) die gleichen Blöcke in der lokal validierten »besten« Blockchain haben. Nicht zu verwechseln mit den Konsensregeln.

Konsensregeln
: Die Regeln für die Blockvalidierung, denen alle Full Nodes folgen, um den Konsens mit anderen Nodes zu erhalten. Nicht zu verwechseln mit Konsens.

KYC
: Know Your Customer (KYC), zu Deutsch etwa »kenne deinen Kunden«, ist der Prozess, mit dem ein Unternehmen die Identität seiner Kunden verifiziert. Der Begriff beschreibt auch die gesetzlichen Vorgaben, die diesen Prozess regeln.

LevelDB
: LevelDB ist ein als Bibliothek ausgelegter festplattenorientierter Schlüssel/Wert-Speicher. Die Bibliothek ist Open Source und für viele Plattformen verfügbar.

Lightning Network
: Lightning Network ist eine vorgeschlagene Implementierung von HTLCs (Hashed Time Lock Contracts) mit bidirektionalen Zahlungskanälen, bei denen Zahlungen über mehrere Peer-to-Peer-Zahlungskanäle sicher geroutet werden können. Das erlaubt den Aufbau eines Netzwerks, bei dem jeder Peer im Netzwerk jeden anderen Peer bezahlen kann, selbst wenn es keinen direkten Kanal zwischen den beiden gibt.

Locktime
: Locktime, oder etwas technischer nLockTime, ist der Teil einer Transaktion, der festlegt, zu welcher Zeit oder zu welchem Block eine Transaktion frühestens in die Blockchain eingefügt werden kann.

Mempool
: Der Bitcoin-Mempool ist die Sammlung aller Transaktionsdaten eines Blocks, die von den Bitcoin-Nodes verifiziert, aber noch nicht bestätigt wurden.

Merkle Root
: Die »Wurzel« eines Merkle Tree, d. h. der Stamm aller gehashten Paare in einem Baum. Block-Header müssen eine gültige Merkle Root für alle Transaktionen in einem Block enthalten.

Merkle Tree
: Ein Baum, der durch das Hashing gepaarter Daten (der »Blätter«) erzeugt wird. Diese Paare werden dann immer weiter gehasht, bis nur noch ein einziger Hash übrig bleibt: die Merkle Root. Beim Bitcoin sind die Blätter fast immer Transaktionen eines einzelnen Blocks.

Miner
: Eine Netzwerk-Node, die durch wiederholtes Hashing einen gültigen Proof-of-Work für neue Blöcke findet.

Multisignatur
: Die Multisignatur (Multisig) verlangt mehr als einen Schlüssel für die Autorisierung einer Bitcoin-Transaktion.

Netzwerk
: Ein Peer-to-Peer-Netzwerk, das Transaktionen und Blöcke an jede Bitcoin-Node im Netzwerk propagiert.

Nonce
: Die Nonce in einem Bitcoin-Block ist ein 32-Bit-Feld (4 Byte) und wird so gesetzt, dass der Hash des Blocks eine bestimmte Anzahl führender Nullen enthält. Die restlichen Felder dürfen nicht verändert werden, da sie eine definierte Bedeutung haben.

Off-Chain-Transaktionen
: Eine Off-Chain-Transaktion bewegt Werte außerhalb der Blockchain. Während eine On-Chain-Transaktion – die üblicherweise einfach als »Transaktion« bezeichnet wird – die Blockchain modifiziert und darauf angewiesen ist,

dass die Blockchain deren Gültigkeit validiert, verwendet eine Off-Chain-Transaktion andere Methoden, um eine Transaktion festzuhalten und zu validieren.

Opcode
: Operationscodes (kurz Opcodes) der Bitcoin-eigenen Skriptsprache (Script), die in Pubkey- oder Signaturskripten Daten ablegen oder bestimmte Funktionen durchführen.

Open-Assets-Protokoll
: Das Open-Assets-Protokoll ist ein einfaches, aber leistungsfähiges Protokoll, das auf der Bitcoin-Blockchain aufsetzt. Es erlaubt die Emission und den Transfer selbst definierter »Assets« (Vermögenswerte). Das Open-Assets-Protokoll ist eine Weiterentwicklung des Colored-Coins-Konzepts.

OP_RETURN
: Ein Opcode, der in einem der Outputs einer OP_RETURN-Transaktion verwendet wird. Nicht zu verwechseln mit einer OP_RETURN-Transaktion.

OP_RETURN-Transaktion
: Ein Transaktionstyp, der standardmäßig seit Bitcoin Core 0.9.0 (und höher) weitergeleitet und geschürft wird. Fügt beliebige Daten zu einem nachweislich nicht einlösbaren Pubkey-Skript hinzu, das von Full Nodes nicht in deren UTXO-Datenbank gespeichert werden muss. Nicht zu verwechseln mit dem OP_RETURN-Opcode.

Output
: Output, Transaktions-Output oder TxOut ist das Ergebnis einer Transaktion und besteht aus zwei Feldern: einem Wertfeld zur Übertragung von null oder mehr Satoshis und einem Pubkey-Skript, das die Bedingungen festlegt, zu denen diese Satoshis eingelöst werden können.

P2PKH
: Transaktionen an eine Bitcoin-Adresse enthalten ein P2PKH- oder Pay-to-Public-Key-Hash-Skript. Ein Output, der an ein P2PKH-Skript gekoppelt ist, kann freigegeben (eingelöst) werden, indem man einen Public Key und eine mit dem dazugehörigen privaten Schlüssel erzeugte digitale Signatur vorlegt.

P2SH
: P2SH, oder Pay To Script Hash, ist eine mächtige neue Art von Transaktion, die den Einsatz komplexer Transaktionsskripte stark vereinfacht. Bei P2SH ist das komplexe Skript, das die Bedingungen für die Freigabe des Outputs (Redeem-Skript) festlegt, nicht im Locking-Skript enthalten. Stattdessen enthält das Locking-Skript nur einen Hash auf das Redeem-Skript.

P2SH-Adresse
: P2SH-Adressen sind Base58Check-codierte 20-Byte-Hashes eines Skripts. P2SH-Adressen verwenden das Versionspräfix 5, was Base58Check-codierte Adressen ergibt, die mit einer 3 beginnen. P2SH-Adressen verstecken die

gesamte Komplexität, d. h., die eine Zahlung vornehmende Person bekommt das Skript nicht zu sehen.

P2WPKH
: Die Signatur eines P2WPKH (Pay to Witness Public Key Hash) enthält die gleichen Informationen wie ein P2PKH, steht aber im Witness- und nicht im scriptSig-Feld. Der scriptPubKey wird ebenfalls modifiziert.

P2WSH
: Der Unterschied zwischen P2SH und P2WSH (Pay to Witness Script Hash) besteht darin, dass der kryptografische Beweis aus dem scriptSig- in das Witness-Feld verschoben wird. Der scriptPubKey wird ebenfalls modifiziert.

Paper-Wallet
: Im engeren Sinne ist eine Paper-Wallet ein Dokument, das alle Daten enthält, die notwendig sind, um eine beliebige Anzahl privater Bitcoin-Schlüssel zu erzeugen. Für die meisten Menschen ist sie eine Möglichkeit, Bitcoins (samt Paper Keys und Freigabecodes) offline in einem physischen Dokument zu speichern.

Pool-Mining
: Beim Pool-Mining tragen mehrere Clients zur Generierung eines neuen Blocks bei und teilen sich die Erlöse entsprechend der beigetragenen Leistung.

Proof-of-Stake
: Proof-of-Stake (PoS) ist eine Methode, nach der das Blockchain-Netzwerk einer Kryptowährung versucht, einen netzwerkweiten Konsens zu erzielen. Beim Proof-of-Stake müssen die Nutzer den Besitz einer bestimmten Menge der Währung (ihren »Anteil«, engl. Stake) nachweisen.

Proof-of-Work
: Ein Stück Daten, dessen Auffinden einen beträchtlichen rechnerischen Aufwand verlangt. Beim Bitcoin müssen Miner eine numerische Lösung für den SHA256-Algorithmus finden, die eine netzwerkweite Zielvorgabe, das sogenannte Difficulty Target, erfüllt.

RIPEMD-160
: RIPEMD-160 ist eine kryptografische 160-Bit-Hashfunktion. RIPEMD-160 ist eine erweiterte Version von RIPEMD mit einem Hashergebnis von 160 Bit. Man erwartet, dass sie für die nächsten zehn Jahre (oder mehr) sicher ist.

Satoshi Nakamoto
: Satoshi Nakamoto ist der Name, der von jener Person (oder den Personen) verwendet wurde, die den Bitcoin entworfen und die ursprüngliche Referenzimplementierung entwickelt hat. Als Teil der Implementierung hat er auch die erste Blockchain-Datenbank entwickelt, wobei er als Erster das Double-Spending-Problem für Digitalwährungen gelöst hat. Seine wahre Identität ist bis heute nicht bekannt.

Script
: Bitcoin verwendet ein Skripting-System für Transaktionen. Script orientiert sich an Forth und ist eine einfache, stackbasierte Sprache, die von links nach rechts verarbeitet wird. Sie verzichtet bewusst auf Schleifen und ist daher nicht Touring-vollständig.

ScriptPubKey (alias Pubkey Script)
: ScriptPubKey, oder Pubkey Script, ist ein in Outputs enthaltenes Skript, das die Bedingungen festlegt, die erfüllt werden müssen, um die Satoshis einlösen zu können. Die Daten, die zur Erfüllung dieser Bedingungen benötigt werden, können in einem Signaturskript bereitgestellt werden.

ScriptSig (aka Signaturskript)
: ScriptSig, oder Signaturskript, steht für die Daten, die der Einlösende generiert. Diese Daten werden fast immer als Variablen verwendet, um die Bedingungen eines Pubkey-Skripts zu erfüllen.

Segregated Witness
: Segregated Witness ist eine Erweiterung des Bitcoin-Protokolls. Sie trennt die Signaturdaten von den Bitcoin-Transaktionen ab. Segregated Witness ist ein Soft Fork, der die Regeln des Bitcoin-Protokolls technisch restriktiver handhabt.

SHA
: Der Secure-Hash-Algorithmus, kurz SHA, ist eine Familie kryptografischer Hashfunktionen, die vom National Institute of Standards and Technology (NIST) veröffentlicht wurde.

Soft Fork
: Ein Soft Fork ist eine temporäre Teilung der Blockchain. Tritt auf, wenn Miner nicht aktualisierte Nodes nutzen, die einer neuen Konsensregel nicht folgen. Nicht zu verwechseln mit Fork, Hard Fork, Software Fork oder Git Fork.

SPV (Simplified Payment Verification)
: SPV, oder Simplified Payment Verification, ist eine Methode, um zu verifizieren, ob bestimmte Transaktionen in einem Block enthalten sind, ohne den gesamten Block herunterladen zu müssen. Diese Methode wird von einigen leichtgewichtigen Bitcoin-Clients genutzt.

Timelocks
: Ein Timelock verhindert, dass Bitcoins vor einem bestimmten Zeitpunkt (oder einer bestimmten Blockhöhe) eingelöst werden können. Timelocks spielen bei vielen Bitcoin-Verträgen eine wichtige Rolle, einschließlich Zahlungskanälen und gehashten Timelock-Verträgen.

Transaktion
: Einfach ausgedrückt die Übertragung von Bitcoin von einer Adresse an eine andere. Genauer formuliert, ist eine Transaktion eine signierte Datenstruktur,