

Imran Bashir

Blockchain

Zaawansowane
zastosowania
łańcucha bloków

Wydanie 2

Helion 

Packt 

Tytuł oryginału: Mastering Blockchain - Second Edition

Tłumaczenie: Tomasz Walczak

ISBN: 978-83-283-4957-5

Copyright © Packt Publishing 2018.

First published in the English language under the title 'Mastering Blockchain - Second Edition – (9781788839044)'

Polish edition copyright © 2019 by Helion SA
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Dodatkowe materiały do książki można znaleźć pod adresem: <ftp://ftp.helion.pl/przyklady/bloczz.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/bloczz>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorze	13
O recenzencie	14
Przedmowa	15
Rozdział 1. ABC łańcucha bloków	21
Rozwój technologii łańcucha bloków	21
Systemy rozproszone	24
Historia łańcucha bloków i Bitcoina	26
Elektroniczne pieniądze	26
Łańcuch bloków	27
Uniwersalne elementy łańcucha bloków	31
Zalety i ograniczenia łańcucha bloków	34
Warstwy łańcucha bloków	36
Cechy łańcucha bloków	37
Typy łańcuchów bloków	39
Rozproszone rejestry	40
Technologia DLT	40
Publiczne łańcuchy bloków	41
Prywatne łańcuchy bloków	41
Wspólny rejestr	42
W pełni prywatne i zastrzeżone łańcuchy bloków	42
Łańcuchy bloków z tokenami	43
Łańcuchy bloków bez tokenów	43

Konsensus	43
Mechanizmy osiągnięcia konsensusu	43
Rodzaje mechanizmów osiągnięcia konsensusu	44
Konsensus w łańcuchu bloków	45
Twierdzenie CAP i łańcuch bloków	47
Podsumowanie	49
Rozdział 2. Decentralizacja	51
Decentralizacja z użyciem łańcucha bloków	51
Metody decentralizacji	53
Eliminowanie pośrednictwa	53
Decentralizacja oparta na współzawodnictwie	54
Drogi do decentralizacji	55
Jak przeprowadzić decentralizację?	56
Przykładowy schemat analizy decentralizacji	56
Łańcuch bloków i kompletny ekosystem związany z decentralizacją	57
Składowanie danych	57
Komunikacja	58
Decentralizacja a moc obliczeniowa	59
Inteligentne kontrakty	60
Zdecentralizowane organizacje	61
Zdecentralizowane organizacje autonomiczne	61
Zdecentralizowane korporacje autonomiczne	62
Zdecentralizowane społeczności autonomiczne	62
Zdecentralizowane aplikacje (DApps)	62
Wymogi stawiane zdecentralizowanym aplikacjom	63
Operacje w zdecentralizowanych aplikacjach	63
Platformy do decentralizacji	64
Ethereum	64
MaidSafe	64
Lisk	65
Podsumowanie	65
Rozdział 3. Kryptografia symetryczna	67
Korzystanie z narzędzia OpenSSL w wierszu poleceń	67
Wprowadzenie	68
Matematyka	69
Kryptografia	70
Poufność	71
Integralność	71
Uwierzytelnianie	71
Niezaprzeczalność	72
Rozliczalność	73
Podstawowe mechanizmy kryptograficzne	73
Kryptografia symetryczna	74
DES	79
AES	79
Podsumowanie	83

Rozdział 4. Kryptografia klucza publicznego	85
Kryptografia asymetryczna	85
Rozkład liczb całkowitych na czynniki	87
Logarytm dyskretny	87
Krzywe eliptyczne	87
Klucze publiczny i prywatny	88
RSA	88
Problem logarytmu dyskretnego w ECC	94
Funkcje skrótu	102
Algorytm tworzenia podpisów cyfrowych za pomocą RSA	111
Algorytm ECDSA	113
Rynki i transakcje finansowe	118
Handel	119
Giełdy	119
Cykl życia transakcji	121
Osoby przewidujące zlecenia	122
Manipulowanie rynkiem	122
Podsumowanie	123
Rozdział 5. Wprowadzenie do Bitcoina	125
Bitcoin	127
Definicja Bitcoina	129
Bitcoin z lotu ptaka	130
Klucze i adresy cyfrowe	136
Klucze prywatne w Bitcoinie	136
Klucze publiczne w Bitcoinie	138
Adresy w Bitcoinie	139
Transakcje	141
Cykl życia transakcji	142
Struktura danych transakcji	143
Rodzaje transakcji	147
Sprawdzanie poprawności transakcji	150
Łańcuch bloków	151
Struktura bloku	151
Struktura nagłówka bloku	151
Blok początkowy	153
Wydobywanie	156
Zadania górników	156
Nagrody za wydobycie bloku	157
Dowód pracy	157
Algorytm wydobywania	158
Szybkość obliczania skrótów	160
Systemy wydobywania	160
Kopalnie	163
Podsumowanie	165

Rozdział 6. Sieć Bitcoina i płatności	167
Sieć Bitcoina	167
Portfele	175
Portfele niedeterministyczne	175
Portfele deterministyczne	175
Hierarchiczne portfele deterministyczne	176
Portfele pamięciowe	176
Portfele papierowe	176
Portfele sprzętowe	176
Portfele internetowe	177
Portfele mobilne	177
Płatności w bitcoinach	178
Innowacje w Bitcoinie	180
Dokumenty BIP	181
Zaawansowane protokoły	181
Segregated Witness (SegWit)	181
Bitcoin Cash	183
Bitcoin Unlimited	183
Bitcoin Gold	183
Inwestycje w bitcoiny oraz ich kupno i sprzedaż	184
Podsumowanie	185
Rozdział 7. Klienci i interfejsy API Bitcoina	187
Instalowanie Bitcoina	187
Typy klientów Bitcoin Core	187
Przygotowywanie węzła Bitcoina	188
Konfigurowanie kodu źródłowego	190
Konfigurowanie pliku bitcoin.conf	190
Uruchamianie węzła w sieci testnet	190
Uruchamianie węzła w sieci regtest	191
Eksperymentowanie z interfejsem bitcoin-cli	192
Programowanie w świecie Bitcoina i interfejsy uruchamiany w wierszu poleceń	192
Podsumowanie	194
Rozdział 8. Inne kryptowaluty	195
Podstawy teoretyczne	198
Co zamiast dowodu pracy?	198
Różne rodzaje stawek	201
Dostosowywanie trudności i algorytmy zmiany celu	202
Ograniczenia Bitcoina	205
Prywatność i anonimowość	205
Rozszerzone protokoły oparte na Bitcoinie	207
Rozwój alternatywnych kryptowalut	209
Namecoin	211
Handel namecoinami	213
Pozyskiwanie namecoinów	213
Generowanie rekordów w Namecoinie	215
Litecoin	217

Primecoin	220
Handel primecoinami	221
Przewodnik po wydobywaniu	221
Zcash	223
Handel zcashami	225
Przewodnik po wydobywaniu	225
Emisje ICO	230
Tokeny zgodne ze standardem ERC20	231
Podsumowanie	232
Rozdział 9. Inteligentne kontrakty	233
Historia	233
Definicja	234
Kontrakty ricardiańskie	237
Szablony inteligentnych kontraktów	239
Wyrocznie	241
Inteligentne wyrocznie	243
Umieszczanie inteligentnych kontraktów w łańcuchu bloków	243
The DAO	244
Podsumowanie	245
Rozdział 10. ABC łańcucha bloków Ethereum	247
Wprowadzenie	247
Specyfikacja techniczna	248
Łańcuch bloków Ethereum	249
Ethereum z lotu ptaka	250
Sieć Ethereum	254
Mainnet	254
Testnet	254
Sieć prywatna	254
Komponenty ekosystemu Ethereum	255
Klucze i adresy	256
Konta	256
Transakcje i komunikaty	258
Kryptowaluta i tokeny Ether (ETC i ETH)	267
Maszyna EVM	267
Inteligentne kontrakty	271
Podsumowanie	274
Rozdział 11. Jeszcze o Ethereum	275
Języki programowania	276
Wykonywany kod bajtowy	276
Bloki i łańcuchy bloków	284
Poziom opłat	290
Portfele i oprogramowanie klienckie	298
Protokoły pomocnicze	307
Skalowalność, bezpieczeństwo i inne wyzwania	309
Handel i inwestycje	309
Podsumowanie	310

Rozdział 12. Środowisko programistyczne Ethereum	311
Sieci testowe	312
Konfigurowanie sieci prywatnej	313
Identyfikator sieci	314
Plik początkowy	314
Katalog na dane	315
Uruchamianie sieci prywatnej	316
Uruchamianie przeglądarki Mist w sieci prywatnej	321
Dodawanie kontraktów za pomocą przeglądarki Mist	323
Eksplorator bloków prywatnej i lokalnej sieci Ethereum	326
Podsumowanie	329
Rozdział 13. Narzędzia i platformy programistyczne	331
Języki	332
Kompilatory	333
Język Solidity	344
Typy	345
Podsumowanie	356
Rozdział 14. Wprowadzenie do Web3	357
Web3	357
Dodawanie kontraktów	358
Żądania POST	363
Fronton napisany w HTML-u i JavaScriptcie	364
Platformy programistyczne	371
Podsumowanie	397
Rozdział 15. Hyperledger	399
Projekty w ramach programu Hyperledger	399
Fabric	400
Sawtooth Lake	400
Iroha	400
Burrow	401
Indy	401
Explorer	402
Cello	402
Composer	402
Quilt	402
Hyperledger jako protokół	403
Architektura wzorcowa	403
Wymogi i cele projektowe związane z platformą Hyperledger Fabric	405
Fabric	407
Hyperledger Fabric	408
Sawtooth Lake	418
Corda	424
Podsumowanie	430

Rozdział 16. Inne łańcuchy bloków	431
Łańcuchy bloków	431
Kadena	432
Ripple	436
Stellar	441
Rootstock	442
Quorum	444
Tezos	445
Storj	446
MaidSafe	447
BigchainDB	448
MultiChain	448
Tendermint	448
Platformy i frameworki	449
Eris	449
Podsumowanie	451
Rozdział 17. Łańcuch bloków — poza świat walut	453
Internet rzeczy	453
Warstwa obiektów fizycznych	455
Warstwa urządzeń	455
Warstwa sieci	455
Warstwa zarządzania	456
Warstwa aplikacji	456
Eksperyment z internetem rzeczy opartym na łańcuchu bloków	459
Instytucje rządowe	474
Opieka zdrowotna	478
Finanse	478
Multimedia	481
Podsumowanie	481
Rozdział 18. Skalowalność i inne problemy	483
Skalowalność	484
Poziom sieci	484
Poziom osiągnięcia konsensusu	484
Poziom składowania danych	485
Poziom widoku	485
Zwiększenie wielkości bloku	485
Skracanie czasu wydobywania bloków	486
Tablice IBLT	486
Sharding	487
Kanały stanu	487
Prywatny łańcuch bloków	488
Dowód stawki	488
Łańcuchy boczne	488
Prywatność	491
Zaciemnianie z nieodróżnialnością danych	491
Szyfrowanie homomorficzne	492
Dowody ZKP	492

Kanały stanu	493
Bezpieczne obliczenia z udziałem wielu jednostek	493
Wykorzystanie sprzętu do zapewniania poufności	493
CoinJoin	494
Poufne transakcje	494
MimbleWimble	494
Bezpieczeństwo	495
Podsumowanie	501
Rozdział 19. Aktualna sytuacja i przyszły rozwój	503
Pojawiające się trendy	503
Łańcuchy bloków specyficzne dla zastosowań	503
Łańcuchy bloków dla przedsiębiorstw	504
Prywatne łańcuchy bloków	504
Startupy	505
Duże zainteresowanie ze strony naukowców	505
Standaryzacja	506
Usprawnienia	507
Implementacje stosowane w praktyce	507
Konsorcja	508
Rozwiązania problemów technicznych	508
Łączenie z innymi technologiami	508
Edukacja w zakresie technologii łańcuchów bloków	509
Zatrudnienie	509
Kryptoekonomia	509
Badania w dziedzinie kryptografii	510
Nowe języki programowania	510
Badania nad sprzętem i jego rozwój	510
Badania nad metodami formalnymi i bezpieczeństwem	511
Alternatywy względem łańcuchów bloków	511
Prace nad umożliwieniem współdziałania	511
Model BaaS	512
Prace nad ograniczeniem zużycia energii	512
Inne wyzwania	512
Regulacje	512
Ciemna strona	513
Badania nad łańcuchami bloków	515
Inteligentne kontrakty	515
Problemy z centralizacją	515
Ograniczenia funkcji kryptograficznych	515
Algorytmy osiągnięcia konsensusu	515
Skalowalność	516
Zaciemnianie kodu	516
Ważne projekty	516
Zcash dla Ethereum	516
CollCo	517
Cello	517
Qtum	517
Bitcoin-NG	517

Solidus	517
Hawk	518
Town-Crier	518
SETLCoin	518
TEEChan	518
Falcon	519
Bletchley	519
Casper	519
Różne narzędzia	520
Rozszerzenie dla języka Solidity w środowisku Microsoft Visual Studio	520
MetaMask	520
Stratis	520
Embark	521
DAPPLE	521
Meteor	521
uPort	521
INFURA	522
Powiązania z innymi branżami	522
Przyszłość	523
Podsumowanie	525
Skorowidz	527

Decentralizacja

Decentralizacja nie jest nową koncepcją. Od dawna wykorzystywano ją w strategii, zarządzaniu i rządzeniu. Podstawową ideą decentralizacji jest przeniesienie kontroli i władzy na obrzeża organizacji zamiast pozostawiania pełnej kontroli organizacji w rękach jednego centralnego ciała. Takie rozwiązanie zapewnia organizacjom różne korzyści, takie jak wzrost wydajności, przyspieszenie podejmowania decyzji, wzrost motywacji i zmniejszenie obciążenia wyższej kadry menedżerskiej.

W tym rozdziale decentralizacja jest opisana w kontekście łańcuchów bloków. Jednym z podstawowych aspektów łańcucha bloków jest brak centralnej jednostki, która go kontroluje. W tym rozdziale przedstawione zostaną przykłady różnych metod decentralizacji i dróg do jej osiągnięcia. Ponadto szczegółowo opisane zostaną decentralizacja ekosystemu łańcucha bloków, zdecentralizowane aplikacje i platformy do zapewniania decentralizacji. Poznasz też wiele ekscytujących aplikacji i idei, których źródłem są zdecentralizowane łańcuchy bloków.

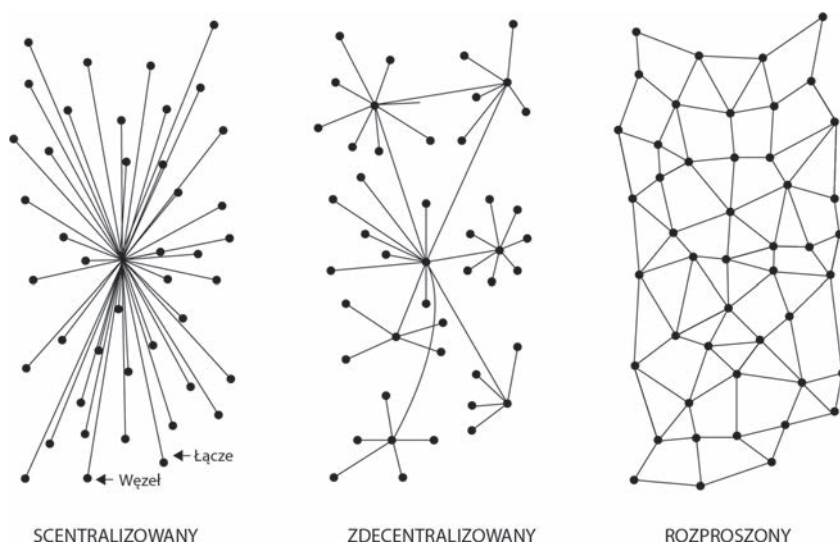
Decentralizacja z użyciem łańcucha bloków

Decentralizacja jest jedną z najważniejszych korzyści i usług zapewnianych przez technologię łańcuchów bloków. Z budowy łańcuchów bloków wynika, że są one doskonałym narzędziem do budowania platformy, która nie wymaga żadnych pośredników i może działać z wieloma różnymi liderami wybieranymi za pomocą mechanizmu osiągnięcia konsensusu. Ten model umożliwia każdemu współzawodnictwo o pozycję jednostki podejmującej decyzje. To współzawodnictwo odbywa się na podstawie mechanizmu osiągnięcia konsensusu, a najczęściej używaną metodą jest tu **dowód pracy**.

Decentralizacja może być stosowana na różnym poziomie: od modeli częściowo zdecentralizowanych po w pełni zdecentralizowane. Zależy to od wymogów i okoliczności. W kontekście łańcuchów bloków decentralizację można postrzegać jako mechanizm umożliwiający modyfikację istniejących aplikacji i paradygmatów oraz budowanie nowych aplikacji w celu zapewnienia pełnej kontroli użytkownikom.

Technologie teleinformatyczne tradycyjnie są oparte na scentralizowanym paradygmacie, w którym serwery bazodanowe lub serwery aplikacji są kontrolowane przez centralną jednostkę, np. administratora systemu. Wraz z pojawieniem się Bitcoina i technologii łańcuchów bloków ten model się zmienił. Obecnie istnieje technologia, która umożliwia każdemu zbudowanie zdecentralizowanego systemu, działającego bez pojedynczego punktu podatności na awarię lub jednej zaufanej jednostki zarządzającej. Taki system może działać autonomicznie lub wymagać interwencji człowieka; zależy to od typu i modelu zarządzania stosowanego w zdecentralizowanej aplikacji działającej w łańcuchu bloków.

Rysunek 2.1 ilustruje różne rodzaje istniejących obecnie systemów: scentralizowane, zdecentralizowane i rozproszone. Ten podział po raz pierwszy przedstawił Paul Baran w książce *On Distributed Communications: I. Introduction to Distributed Communications Networks* (Rand Corporation, 1964).



Rysunek 2.1. Różne typy sieci i systemów

Systemy scentralizowane to tradycyjne systemy informatyczne (klient – serwer), w których występuje pojedyncza jednostka zarządzająca — kontroluje ona dany system i samodzielnie odpowiada za wszystkie jego operacje. Wszyscy użytkownicy scentralizowanego systemu zależą od jednego źródła usług. Większość dostawców usług internetowych, w tym Google, Amazon, eBay, App Store firmy Apple itd., posługują się tym tradycyjnym modelem dostarczania usług.

W **systemie rozproszonym** dane i obliczenia są rozdzielane między wiele węzłów sieci. Czasem to pojęcie jest mylone z *przetwarzaniem równoległym*. Choć oba te rozwiązania w pewnym zakresie się pokrywają, główna różnica między nimi polega na tym, że w systemach przetwarzania równoległego obliczenia są wykonywane przez wszystkie węzły jednocześnie w celu uzyskania wyniku. Platformy przetwarzania równoległego są używane np. do badania i prognozowania pogody, do symulacji i w modelowaniu finansowym. Z kolei w systemie rozproszonym obliczenia nie muszą być wykonywane równoległe, a dane są replikowane w wielu węzłach

postrzeganych przez użytkowników jako jeden spójny system. Odmiany obu tych modeli są używane do osiągnięcia odporności na błędy i zwiększenia szybkości. W systemach równoległych nadal występuje centralna jednostka zarządzająca, która kontroluje wszystkie węzły i zarządza przetwarzaniem. To oznacza, że system jest z natury scentralizowany.

Najważniejsza różnica między systemem zdecentralizowanym a rozproszonym polega na tym, że w systemie rozproszonym występuje centralna jednostka nadrzędna zarządzająca całym systemem. W systemie zdecentralizowanym taka jednostka nie istnieje.

System zdecentralizowany to typ sieci, w której węzły nie są zależne od jednego węzła nadrzędnego. Zamiast tego kontrola jest rozproszona między wiele węzłów. Jest to zbliżone do modelu, w którym każdy dział organizacji odpowiada za własny serwer bazodanowy. W ten sposób kontrola jest odbierana centralnemu serwerowi i przekazywana do działów zarządzających własnymi bazami.

Ważną innowacją w paradygmacie zdecentralizowanym, będącą załączkiem nowej ery decentralizacji aplikacji, jest osiągnięcie **konsensusu w środowisku zdecentralizowanym**. Ten mechanizm pojawił się wraz z Bitcoinem i umożliwia użytkownikom uzgadnianie rzeczy za pomocą algorytmu osiągnięcia konsensusu, bez konieczności udziału centralnej, zaufanej trzeciej strony, pośrednika lub dostawcy usług.

Metody decentralizacji

Do zapewniania decentralizacji można stosować dwie metody: eliminowanie pośrednictwa i współzawodnictwo (decentralizacja oparta na współzawodnictwie). Zostaną one szczegółowo opisane w następujących punktach.

Eliminowanie pośrednictwa

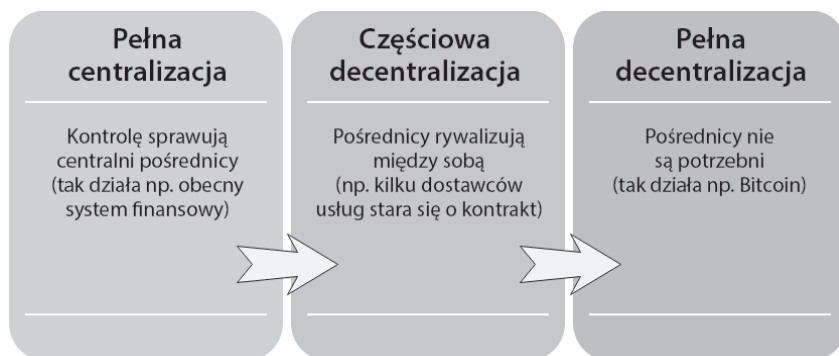
Koncepcję **eliminowania pośrednictwa** można wytłumaczyć na przykładzie. Wyobraź sobie, że chcesz przesłać pieniądze do znajomego z innego państwa. Idziesz do banku, który za opłatą prześle pieniądze do banku w docelowym kraju. W tym scenariuszu bank przechowuje centralną bazę danych, która jest aktualizowana, co potwierdza, że przesłałeś pieniądze. Łańcuch bloków umożliwia przesłanie pieniędzy bezpośrednio do znajomego bez konieczności korzystania z usług banku. Wystarczy do tego adres znajomego w łańcuchu bloków. W ten sposób pośrednik, czyli bank, przestaje być potrzebny, a decentralizacja jest uzyskiwana dzięki *wyeliminowaniu pośrednictwa*. Kwestią dyskusyjną jest to, na ile praktyczna jest decentralizacja przez eliminowanie pośrednictwa w sektorze finansowym z jego ogromnymi wymogami regulacyjnymi. Jednak ten model może być stosowany nie tylko w finansach, ale też w wielu innych branżach.

Decentralizacja oparta na współzawodnictwie

W metodzie obejmującej **współzawodnictwo** różni dostawcy usług konkurują ze sobą o to, by system wybrał ich do świadczenia usług. Ten model nie prowadzi do pełnej decentralizacji. Jednak do pewnego stopnia gwarantuje, że pośrednik lub dostawca usług nie zmonopolizuje usługi. W kontekście łańcuchów bloków można wyobrazić sobie system, w którym inteligentne kontrakty mogą wybrać zewnętrznego dostawcę danych spośród dużej ich liczby na podstawie reputacji, wcześniejszych ocen, recenzji i jakości świadczonych usług.

Ta metoda nie zapewnia całkowitej decentralizacji, ale umożliwia inteligentnym kontraktom dokonywanie swobodnych wyborów na podstawie podanych kryteriów. W ten sposób budowane jest oparte na konkurencji środowisko, w którym dostawcy usług rywalizują między sobą o to, by zostać wybranym na dostawcę danych.

Na rysunku 2.2 pokazane są różne poziomy decentralizacji. Po lewej stronie przedstawione jest tradycyjne podejście, w którym kontrolę sprawuje centralny system. W modelu widocznym po prawej stronie całkowicie wyeliminowano pośrednictwo. Pośrodku widoczni są rywalizujący ze sobą pośrednicy lub dostawcy usług. W tym rozwiązaniu pośrednicy lub dostawcy usług są wybierani na podstawie reputacji lub głosowania, co pozwala uzyskać częściową decentralizację.



Rysunek 2.2. Poziom decentralizacji

Choć decentralizacja zapewnia wiele korzyści, w tym przejrzystość, wydajność, oszczędności, rozwój zaufanych ekosystemów, a w niektórych sytuacjach także prywatność i anonimowość, to jednocześnie wymaga starannego przeanalizowania pewnych problemów, takich jak wymogi z zakresu bezpieczeństwa, błędy programowe i błędy ludzkie.

Oto przykład: jak w zdecentralizowanym systemie, takim jak Bitcoin lub Ethereum, gdzie bezpieczeństwo jest standardowo zapewniane dzięki kluczom prywatnym, zagwarantować, że cyfrowe zasoby powiązane z takimi kluczami nie staną się bezwartościowe po utracie klucza lub że błąd w kodzie inteligentnego kontraktu lub zdecentralizowanej aplikacji nie narazi użytkownika na atak? Przed rozpoczęciem decentralizowania wszystkiego za pomocą łańcucha bloków i zdecentralizowanych aplikacji trzeba zrozumieć, że nie wszystko można i trzeba decentralizować.

To podejście rodzi kilka podstawowych pytań: czy łańcuch bloków naprawdę jest potrzebny? Kiedy przydatny jest łańcuch bloków? W jakich sytuacjach jest on lepszy od tradycyjnych baz danych? Aby odpowiedzieć na te pytania, zastanów się nad prostym zestawem zaprezentowanych tu kwestii:

1. Czy potrzebna jest wysoka przepustowość obsługi danych? Jeśli odpowiedź na to pytanie brzmi „tak”, zastosuj tradycyjną bazę danych.
2. Czy aktualizacje są kontrolowane przez centralną jednostkę? Jeżeli tak jest, posłuż się tradycyjną bazą.
3. Czy użytkownicy sobie ufają? Jeśli tak, użyj tradycyjnej bazy.
4. Czy użytkownicy są anonimowi? Jeżeli tak, zastosuj publiczny łańcuch bloków. W przeciwnym razie wykorzystaj prywatny łańcuch bloków.
5. Jeśli konsensus ma być utrzymywany w ramach konsorcjum, utwórz prywatny łańcuch bloków. W przeciwnym razie zastosuj publiczny łańcuch bloków.

Udzielenie odpowiedzi na wszystkie te pytania może pozwolić zrozumieć, czy łańcuch bloków jest potrzebny. Oprócz postawionych tu pytań trzeba uwzględnić także wiele innych kwestii, takich jak opóźnienie, mechanizmy osiągania konsensusu, to, czy konsensus jest konieczny i gdzie będzie ustalany. Jeśli konsensus jest utrzymywany wewnątrz konsorcjum, należy zastosować prywatny łańcuch bloków. W przeciwnym razie, gdy konsensus ma być osiągany publicznie przez wiele jednostek, należy rozważyć publiczny łańcuch bloków. W trakcie wyboru między łańcuchem bloków a tradycyjną bazą danych należy rozważyć także inne aspekty, np. niemodyfikowalność. Jeśli jest ona niezbędna, należy zastosować publiczny łańcuch bloków; w przeciwnym razie odpowiednim rozwiązaniem może być centralna baza danych.

Wraz z dojrzewaniem technologii łańcuchów bloków mogą pojawić się kolejne pytania dotyczące tego modelu. Na razie jednak podany zestaw pytań wystarcza do zdecydowania, czy rozwiązanie oparte na łańcuchu bloków jest potrzebne, czy nie.

Drogi do decentralizacji

Już przed łańcuchami bloków i Bitcoinem istniały inne systemy (np. systemy wymiany plików BitTorrent i Gnutella), które można uznać za — w pewnym stopniu — zdecentralizowane. Jednak wraz z pojawieniem się łańcuchów bloków zaczęto realizować wiele projektów wykorzystujących tę nową technologię do osiągania decentralizacji. Bitcoin jest dla wielu osób pierwszym wyborem, ponieważ okazał się najbardziej odpornym i bezpiecznym łańcuchem bloków, a w czasie, gdy powstaje ta książka, jego wartość rynkowa wynosi blisko 145 mld dolarów. Wielu programistów do budowania zdecentralizowanych aplikacji korzysta też z innych łańcuchów bloków, takich jak Ethereum. W porównaniu z Bitcoinem Ethereum stał się lepszym wyborem z powodu swobody i możliwości zaprogramowania dowolnej logiki biznesowej w łańcuchu bloków za pomocą *inteligentnych kontraktów*.

Jak przeprowadzić decentralizację?

Arvind Narayanan i in. zaproponowali w książce *Bitcoin and Cryptocurrency Technologies* (Princeton University Press) model, który można wykorzystać do oceny wymogów dotyczących decentralizacji z użyciem łańcucha bloków. W tym modelu zadawane są cztery pytania. Odpowiedzi na nie pozwalają dokładnie zrozumieć, jak zdecentralizować system:

1. Co jest decentralizowane?
2. Jaki poziom decentralizacji jest wymagany?
3. Jaki łańcuch bloków jest używany?
4. Jakie mechanizmy zabezpieczeń są stosowane?

Pierwsze pytanie wymaga określenia, jaki system jest decentralizowany. Może to być dowolny system — np. system sprawdzania tożsamości lub system tradingowy.

Drugie pytanie wymaga określenia wymaganego poziomu decentralizacji na podstawie analizy opisaną wcześniej skali. Można zastosować pełną lub częściową eliminację pośrednictwa.

Trzecie pytanie wymaga od programistów ustalenia, który łańcuch bloków jest odpowiedni w konkretnym zastosowaniu. Może to być łańcuch bloków Bitcoin lub Ethereum albo dowolny inny dostosowany do danej sytuacji.

Ostatnie pytanie, na jakie trzeba odpowiedzieć, dotyczy tego, w jaki sposób gwarantowane będzie bezpieczeństwo zdecentralizowanego systemu. Mechanizm zabezpieczeń może być oparty na atomowości (transakcja jest wtedy wykonywana albo w całości, albo nie jest wykonywana wcale). To deterministyczne podejście zapewnia integralność systemu. Można też zastosować mechanizmy oparte na reputacji, umożliwiające wprowadzenie różnych poziomów zaufania w systemie.

Przykładowy schemat analizy decentralizacji

Jako przykładową aplikację przeznaczoną do decentralizacji zbadajmy system transferu pieniędzy. Cztery podane wcześniej pytania posłużą do oceny wymogów dotyczących decentralizacji tej aplikacji. Oto odpowiedzi na te pytania:

1. System transferu pieniędzy.
2. Eliminowanie pośrednictwa.
3. Bitcoin.
4. Atomowość.

Odpowiedzi wskazują na to, że system transferu pieniędzy można zdecentralizować, eliminując pośrednika, implementując system z użyciem łańcucha bloków Bitcoin i oferując gwarancje bezpieczeństwa za pomocą atomowości. Atomowość gwarantuje, że transakcja albo zostanie wykonana w pełni poprawnie, albo w ogóle nie zostanie przeprowadzona. Wybrany został łańcuch bloków Bitcoin, ponieważ jest najstarszy i sprawdzony.

Opisany schemat można też wykorzystać dla dowolnego innego systemu, który trzeba przeanalizować w kategoriach decentralizacji. Odpowiedzi na postawione cztery proste pytania pomagają doprecyzować, jakie podejście przyjąć w celu decentralizacji systemu.

Łańcuch bloków i kompletny ekosystem związany z decentralizacją

Aby uzyskać pełną decentralizację, konieczna jest także decentralizacja środowiska związanego z łańcuchem bloków. Łańcuch bloków to rozproszony rejestr działający na bazie tradycyjnych systemów odpowiedzialnych np. za składowanie danych, komunikację i obliczenia. Występują też inne aspekty, takie jak tożsamość i bogactwo, do których tradycyjnie stosowane są modele scentralizowane. Decentralizacja musi objąć także te aspekty, aby można było uzyskać odpowiednio zdecentralizowany ekosystem.

Składowanie danych

Dane mogą być przechowywane bezpośrednio w łańcuchu bloków, co pozwala zapewnić decentralizację. Jednak poważną wadą tego podejścia jest to, że łańcuch bloków z natury nie nadaje się dobrze do składowania dużych ilości danych. Może przechowywać proste transakcje i pewną ilość dowolnych danych, jednak z pewnością nie nadaje się do składowania zdjęć lub dużych obiektów z danymi, do czego używane są tradycyjne systemy bazodanowe.

Lepszym sposobem składowania danych jest używanie **rozproszonych tablic mieszających** (ang. *Distributed Hash Table* — DHT). Tablice DHT stosowano pierwotnie w działających w modelu P2P systemach wymiany plików (np. w systemach BitTorrent, Napster, Kazaa i Gnutella). Badania nad tablicami DHT zyskały popularność dzięki projektom CAN, Chord, Pastry i Tapestry. Najbardziej skalowalną i najszybszą siecią był BitTorrent, jednak problem z tym systemem i podobnymi rozwiązaniami polega na tym, że użytkownicy nie mają interesu w przechowywaniu plików w nieskończoność. Użytkownicy zwykle nie utrzymują plików na stałe, a jeśli węzły z wciąż potrzebnymi komuś danymi opuszczą sieć, nie ma sposobu na pobranie tych danych; potrzebne węzły muszą ponownie dołączyć do sieci, aby pliki ponownie stały się dostępne.

Dwoma podstawowymi wymogami w obszarze składowania danych są wysoka dostępność systemu i stabilność łącza. Oznacza to, że dane powinny być dostępne, gdy są potrzebne, a łącza sieciowe zawsze powinny działać. System **IPFS** (ang. *InterPlanetary File System*) autorstwa Juana Beneta posiada obie te cechy. Benet marzy o tym, by dzięki zastąpieniu protokołu HTTP innym rozwiązaniem powstała zdecentralizowana sieć WWW. System IPFS składa dane w tablicach DHT Kademlia, a wyszukiwanie obsługuje za pomocą **acyklicznych grafów skierowanych skrótów** (ang. *Merkle directed acyclic graph*). Tablice DHT i acykliczne grafy skierowane zostaną szczegółowo opisane w rozdziale 4. „Kryptografia klucza publicznego”.

Mechanizm nagradzania za składowanie danych jest oparty na protokole Filecoin. Nagrody są wypłacane właścicielom węzłów, które przechowują dane za pomocą mechanizm Bitswap. Ten mechanizm umożliwia węzłom przechowywanie prostego rejestru bajtów wysyłanych lub otrzymywanych w modelu „jeden do jednego”. W systemie IPFS używany jest też oparty na narzędziu Git mechanizm wersjonowania, który zapewnia strukturę wersji danych i kontrolę nad nimi.

Istnieją też inne narzędzia do składowania danych, np. Ethereum Swarm, Storj i MaidSafe. Ethereum obejmuje własny zdecentralizowany i rozproszony ekosystem, w którym używane są narzędzie Swarm (do składowania danych) i protokół Whisper (do komunikacji). MaidSafe ma zapewnić zdecentralizowaną sieć WWW. Wszystkie te projekty są szczegółowo opisane w dalszych częściach książki.

BigchainDB to następny projekt przeznaczony do decentralizacji warstwy składowania danych, który ma zapewniać skalowalną liniowo, szybką, zdecentralizowaną bazę danych różną od tradycyjnych systemów plików. BigchainDB uzupełnia zdecentralizowane platformy przetwarzania danych i systemy plików, takie jak Ethereum i IPFS.

Komunikacja

Internet (warstwa komunikacji w łańcuchach bloków) jest uznawany za zdecentralizowany. To przekonanie jest w pewnym zakresie prawdziwe, ponieważ pierwotnie internet opracowano jako zdecentralizowany system komunikacji. Usługi takie jak e-mail i składowanie danych w internecie są obecnie oparte na modelu, w którym kontrolę sprawuje dostawca usług, a użytkownicy ufają, że taki dostawca będzie zapewniał na żądanie dostęp do danej usługi. Ten model jest oparty na bezwarunkowym zaufaniu do centralnej jednostki (dostawcy usług), ponieważ użytkownicy nie mają kontroli nad własnymi danymi. Nawet hasła użytkowników są przechowywane w systemach zaufanej trzeciej strony.

Dlatego trzeba zapewnić kontrolę poszczególnym użytkownikom w taki sposób, aby zagwarantować im dostęp do ich danych bez zależności od pojedynczej trzeciej strony. Dostęp do internetu (warstwy komunikacyjnej) jest zależny od **dostawców usług internetowych**, którzy pełnią funkcję centralnego koncentratora dla użytkowników internetu. Jeśli dostawca usług internetowych z jakiegoś powodu przestanie działać, to w opisanym modelu komunikacja będzie niemożliwa.

Inne rozwiązanie to zastosowanie **sieci w topologii siatki**. Choć w porównaniu z internetem taka sieć ma ograniczone możliwości, stanowi zdecentralizowaną alternatywę, umożliwiającą węzłom bezpośrednio komunikowanie się ze sobą bez centralnego koncentratora takiego jak dostawca usług internetowych.

Przykładową siecią w topologii siatki jest FireChat (<http://www.opengarden.com/firechat.html>). Umożliwia ona użytkownikom iPhone'ów bezpośrednie komunikowanie się ze sobą w modelu P2P bez połączenia internetowego.

Wyobraź sobie teraz sieć, która umożliwi użytkownikom kontrolowanie komunikacji — w żadnej sytuacji nikt nie może jej wyłączyć. Mógłby to być następny krok w kierunku decentralizacji sieci komunikacyjnych w ekosystemie łańcucha bloków. Trzeba zauważyć, że ten model może być niezbędny tylko na obszarach, gdzie internet jest cenzurowany i kontrolowany przez rząd.

Wcześniej wspomniano, że pierwotnie internet miał być zdecentralizowaną siecią. Jednak wraz z upływem lat i powstaniem dużych dostawców usług, takich jak Google, Amazon i eBay, kontrola jest w coraz większym stopniu przekazywana w ręce tych ważnych graczy. Na przykład poczta elektroniczna jest w swej istocie zdecentralizowanym systemem. Oznacza to, że każdy może niewielkim nakładem pracy uruchomić serwer poczty elektronicznej i zacząć wysyłać oraz odbierać e-maile. Dostępne są jednak lepsze rozwiązania, np. Gmail i Outlook.com, które oferują dodatkowe usługi dla użytkowników końcowych. Dlatego naturalnym wyborem jest korzystanie z jednej z takich scentralizowanych usług, ponieważ są one wygodniejsze i bezpłatne. Jest to jeden z przykładów ilustrujących, że internet stał się bardziej scentralizowany.

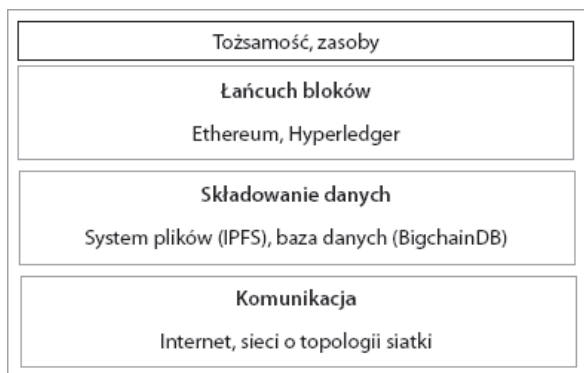
Bezpłatne usługi są jednak oferowane kosztem ujawniania cennych danych osobowych, a wielu użytkowników nie jest tego świadomych. Łańcuch bloków pozwolił znów przedstawić światu wizję decentralizacji, a obecnie aktywnie prowadzone są prace nad opanowaniem tej technologii i uzyskaniem korzyści, jakie może ona oferować.

Decentralizacja a moc obliczeniowa

Za pomocą łańcuchów bloków takich jak Ethereum, gdzie w sieci łańcucha bloków uruchamia się inteligentne kontrakty obejmujące logikę biznesową, można zdecentralizować moc obliczeniową. Inne łańcuchy bloków oferują podobne platformy z warstwą przetwarzania, pozwalające w zdecentralizowany sposób uruchamiać logikę biznesową w sieci.

Na rysunku 2.3 pokazany jest zdecentralizowany ekosystem. W dolnej warstwie internet lub sieci o topologii siatki zapewniają zdecentralizowaną sieć komunikacyjną. W następnej warstwie, składowania danych, decentralizację oferują technologie takie jak IPFS i BigchainDB. W kolejnej warstwie łańcuch bloków pełni funkcję zdecentralizowanej warstwy przetwarzania (warstwy obliczeniowej). Łańcuch bloków może, choć w ograniczonym stopniu, stanowić także warstwę składowania danych, jednak skutkuje to znacznym ograniczeniem szybkości i możliwości systemu. Dlatego do składowania dużych ilości danych w zdecentralizowany sposób lepiej nadają się inne rozwiązania, takie jak IPFS i BigchainDB. Na najwyższym poziomie znajdują się warstwy tożsamości i zasobów. Tożsamość w internecie to bardzo obszerne zagadnienie. Systemy takie jak BitAuth i OpenID zapewniają usługi uwierzytelniania i identyfikacji oraz oferują różny poziom decentralizacji i bezpieczeństwa.

Łańcuch bloków potrafi zapewnić rozwiązania różnych problemów związanych z decentralizacją. **Trójkąt Zooko** (jest to hipoteza związana z tożsamością) wymaga, by system nazw w protokole sieciowym był bezpieczny i zdecentralizowany oraz obejmował nazwy łatwe do zapamiętania i sensowne dla człowieka. Zgodnie ze wspomnianą hipotezą system może mieć tylko dwie z tych trzech cech. Jednak wraz z pojawieniem się łańcucha bloków Namecoin problem został rozwiązany. Ten łańcuch bloków zapewnia bezpieczeństwo, decentralizację i nazwy sensowne



Rysunek 2.3. Zdecentralizowany ekosystem

dla człowieka. To rozwiązanie nie jest jednak uniwersalne i pociąga za sobą wiele wyzwań; użytkownicy muszą na przykład bezpiecznie przechowywać klucze prywatne i zarządzać nimi. To rodzi ogólne pytania o to, czy decentralizacja jest odpowiednia w danej sytuacji.

Decentralizacja nie we wszystkich scenariuszach jest właściwa. Często lepiej sprawdzają się scentralizowane systemy o ustalonej reputacji. Na przykład platformy poczty elektronicznej od poważanych firm, takich jak Google lub Microsoft, zapewniają wyższy poziom usług niż model, w którym odrębne serwery poczty elektronicznej są zarządzane przez użytkowników w internecie.

Rozwijane są liczne projekty rozproszonych i opartych na łańcuchach bloków systemów o większych możliwościach. Na przykład Swarm i Whisper mają obsługiwać zdecentralizowane składowanie danych i komunikację w łańcuchu bloków Ethereum. Swarm i Ethereum zostaną opisane szczegółowo w rozdziale 11. „Jeszcze o Ethereum”.

Wraz z powstaniem zdecentralizowanego modelu w mediach i pracach naukowych zaczęto stosować różne pojęcia i modne słowa. Dzięki pojawieniu się technologii łańcuchów bloków można obecnie tworzyć programowe wersje tradycyjnych fizycznych organizacji; służą do tego **zdecentralizowane organizacje** (ang. *decentralized organization* — DO) i podobne konstrukcje, szczegółowo opisane w dalszej części rozdziału.

W kontekście decentralizacji warto omówić następujące zagadnienia.

Inteligentne kontrakty

Inteligentny kontrakt to zdecentralizowany program. Inteligentne kontrakty nie wymagają do działania łańcucha bloków. Jednak z powodu zapewnianych przez łańcuchy bloków korzyści w zakresie bezpieczeństwa technologia ta stała się standardową zdecentralizowaną platformą wykonywania inteligentnych kontraktów.

Inteligentny kontrakt obejmuje zwykle logikę biznesową i ograniczoną ilość danych. Ta logika jest wykonywana, jeśli spełnione są określone kryteria. Inteligentne kontrakty są stosowane przez użytkowników łańcucha bloków lub działają autonomicznie na rzecz członków sieci.

Więcej informacji na temat inteligentnych kontraktów zawiera rozdział 9. „Inteligentne kontrakty”.

Zdecentralizowane organizacje

Zdecentralizowane organizacje to programy działające w łańcuchu bloków i oparte na działaniu rzeczywistych organizacji, obejmujących ludzi i protokoły. Po dodaniu zdecentralizowanej organizacji do łańcucha bloków, do czego służy inteligentny kontrakt lub zestaw takich kontraktów, następuje decentralizacja i strony komunikują się między sobą na podstawie kodu zdefiniowanego w kodzie oprogramowania zdecentralizowanej organizacji.

Zdecentralizowane organizacje autonomiczne

Zdecentralizowana organizacja autonomiczna (ang. *decentralized autonomous organization* — DAO) jest podobnie jak organizacja zdecentralizowana programem komputerowym działającym na bazie łańcucha bloków. W takim programie umieszczone są reguły zarządzania i logiki biznesowej. Zdecentralizowane organizacje autonomiczne i organizacje zdecentralizowane są prawie identyczne. Główna różnica polega na tym, że DAO są autonomiczne. To oznacza, że są w pełni zautomatyzowane i obejmują logikę wykorzystującą sztuczną inteligencję. Z kolei DO nie posiadają tej cechy i wymagają danych wejściowych od człowieka, aby wykonywać logikę biznesową.

Pierwszym łańcuchem bloków, w którym wprowadzono DAO, był Ethereum. W DAO za jednostkę zarządzającą uważa się kod, a nie ludzi lub papierowe kontrakty. Jednak to człowiek zarządza kodem i ocenia proponowane funkcje na potrzeby społeczności. DAO mogą zatrudniać zewnętrznych pracowników kontraktowych, jeśli posiadacze tokenów (użytkownicy sieci) zapewnią wystarczającą ilość środków.

Najbardziej znanym projektem DAO jest The DAO, w którym w fazie finansowania społecznościowego zebrano 168 mln dolarów. Projekt The DAO opracowano na potrzeby tworzenia funduszu podwyższonego ryzyka mającego zapewniać obsługę zdecentralizowanego modelu biznesowego bez określonej jednostki będącej właścicielem. Niestety, projekt został złamany przez hakerów z powodu błędu w kodzie The DAO, a miliony dolarów w **walucie Ether (ETH)** zostały wyprowadzone z projektu do podrzędnej DAO. Niezbędny był hard fork łańcucha bloków Ethereum, aby odwrócić skutki ataku i rozpocząć odzyskiwanie środków. Ten incydent zapoczątkował debatę na temat bezpieczeństwa, jakości i potrzeby dokładnych testów kodu inteligentnych kontraktów w celu zapewnienia ich integralności i odpowiedniej kontroli. Prowadzone są też — zwłaszcza w środowisku uniwersyteckim — inne projekty, nakierowane na sformalizowanie pisania i testowania inteligentnych kontraktów.

Obecnie DAO nie mają statusu prawnego, choć mogą obejmować inteligentny kod wymuszający przestrzeganie określonych protokołów i warunków. Jednak na razie reguły te nie mają mocy w obowiązującym systemie prawnym. Możliwe, że pewnego dnia **autonomiczne agenty** (czyli działające bez interwencji człowieka fragmenty kodu zamawiane przez organy ścigania lub organy nadzoru) będą obejmować zasady i regulacje, które można będzie umieścić w DAO na potrzeby zapewnienia integralności rozwiązań w kontekście prawa i zgodności z regulacjami. Ponieważ DAO są w pełni zdecentralizowane, można je uruchamiać w dowolnym miejscu. Rodzi to poważne pytania o to, jak zastosować obecny system prawny do różnych obszarów jurysdykcji i lokalizacji geograficznych.

Zdecentralizowane korporacje autonomiczne

Zdecentralizowane korporacje autonomiczne (ang. *decentralized autonomous corporations* — DAC) są podobne do DAO, choć można je uznać za podzbiór tych ostatnich. Definicje DAC i DAO mogą się pokrywać, przy czym różnica między nimi polega na tym, że DAO zwykle są uważane za rozwiązania non profit, natomiast DAC mogą być dochodowe, oferować udziały uczestnikom i wypłacać dywidendy. DAC mogą zarządzać biznesem automatycznie, bez interwencji człowieka, na podstawie zaprogramowanej logiki.

Zdecentralizowane społeczności autonomiczne

Zdecentralizowane społeczności autonomiczne (ang. *decentralized autonomous societies* — DAS) to rozwiązanie, które ma pozwolić całej społeczności funkcjonować z wykorzystaniem łańcucha bloków za pomocą wielu złożonych inteligentnych kontraktów oraz połączenia działających autonomicznie DAO i zdecentralizowanych aplikacji (ang. *decentralized applications* — DApps). Ten model niekoniecznie oznacza podejście „bezpłatne dla wszystkich”. Nie jest też w pełni oparty na ideologii libertariańskiej. Jednak wiele usług świadczonych standardowo przez rządy może być zapewnianych za pomocą łańcuchów bloków. Dotyczy to np. rządowych systemów dowodów osobistych, paszportów, rejestrów aktów prawnych, małżeństw i narodzin. Inna teoria dotyczy tego, że jeśli rząd jest skorumpowany, a scentralizowane systemy nie zapewniają wystarczającego poziomu zaufania niezbędnego społeczeństwu, ludzie mogą uruchomić w łańcuchu bloków własny, wirtualny system, oparty na zdecentralizowanym konsensusie i przejrzystości. Taki scenariusz może wydawać się libertariańskim lub cyberpunkowym snem, jednak dzięki łańcuchowi bloków jest zupełnie realny.

Zdecentralizowane aplikacje (DApps)

Wszystkie wymienione do tej pory idee można przypisać do bardziej ogólnej kategorii zdecentralizowanych aplikacji. DAO, DAC i DO to zdecentralizowane aplikacje działające na bazie łańcuchów bloków w sieci P2P. Te modele reprezentują najnowsze osiągnięcia w technologiach decentralizacji. Zdecentralizowane aplikacje to programy, które mogą działać w odpowiednich łańcuchach bloków (aplikacje typu I), używać istniejących łańcuchów bloków (aplikacje typu II) lub tylko korzystać z protokołów takich łańcuchów (aplikacje typu III).

Wymogi stawiane zdecentralizowanym aplikacjom

Aby aplikacja została uznana za zdecentralizowaną, musi spełniać wymienione niżej kryteria. Ta definicja została przedstawiona w pracy Johnstona i in. *The General Theory of Decentralized Applications, Dapps*.

- Zdecentralizowana aplikacja powinna być w pełni otwartym i autonomicznym oprogramowaniem. Żadna jednostka nie powinna kontrolować większości tokenów. Wszystkie zmiany w aplikacji muszą wynikać z konsensusu osiąganego na podstawie informacji zwrotnych ze społeczności.
- Dane i rejestry operacji w aplikacji muszą być kryptograficznie zabezpieczone oraz składowane w publicznym, zdecentralizowanym łańcuchu bloków, aby uniknąć scentralizowanych punktów podatności na awarie.
- Aplikacja musi używać kryptograficznych tokenów, aby zapewnić dostęp i nagrody jednostkom, które robią coś wartościowego na rzecz aplikacji (np. górnikom w Bitcoinie).
- Tokeny muszą być generowane przez zdecentralizowane aplikacje zgodnie ze standardowym algorytmem kryptograficznym. Wygenerowane tokeny stanowią dowód wartości przekazywanej kontrybutorom (np. górnikom).

Operacje w zdecentralizowanych aplikacjach

Konsensus w zdecentralizowanej aplikacji można uzyskać za pomocą algorytmów osiągnięcia konsensusu, takich jak dowód pracy lub dowód stawki. Do tej pory tylko dowód pracy okazał się zdumiewająco odporny na ataki z udziałem 51% zasobów, czego potwierdzeniem jest działanie Bitcoina. Zdecentralizowane aplikacje mogą rozdzielać tokeny (monety) na podstawie wydobywania, zbiorów i prac programistycznych.

Przykładowe zdecentralizowane aplikacje

W tym miejscu zaprezentowane są przykładowe zdecentralizowane aplikacje.

KYC-Chain

Ta aplikacja zapewnia mechanizmy do bezpiecznego i wygodnego zarządzania danymi typu „poznaj swojego klienta” (ang. *know your customer* — KYC) za pomocą inteligentnych kontraktów.

OpenBazaar

Jest to zdecentralizowana sieć P2P, która umożliwia przeprowadzanie operacji handlowych bezpośrednio między sprzedawcami i klientami, bez korzystania z centralnego operatora takiego jak eBay lub Amazon. Należy zauważyć, że ten system nie jest oparty na łańcuchu bloków. Zamiast tego w sieci P2P używane są tablice DHT, aby umożliwić bezpośrednią komunikację i wymianę danych między węzłami. W tym systemie do obsługi płatności używany jest bitcoin i inne kryptowaluty.

Lazooz

Jest to zdecentralizowany odpowiednik Ubera. Umożliwia wspólne przejazdy samochodem w modelu P2P, a użytkownicy otrzymują nagrody na podstawie dowodu przemieszczania się i mogą zarabiać monety w walucie zooz.

Wiele innych zdecentralizowanych aplikacji zostało zbudowanych na bazie łańcucha bloków Ethereum. Są one przedstawione na stronie <http://dapps.ethercasts.com/>.

Platformy do decentralizacji

Obecnie istnieje wiele platform umożliwiających decentralizację. Podstawową cechą sieci łańcuchów bloków jest zapewnianie decentralizacji. Dlatego każda sieć łańcucha bloków, np. Bitcoin, Ethereum, Hyperledger Fabric lub Quorum, może posłużyć do udostępnienia usługi decentralizacji. Wiele organizacji z całego świata wprowadziło platformy, które mają sprawić, że budowanie rozproszonych aplikacji stanie się łatwe, przystępne i bezpieczne. Dalej opisano niektóre z tych platform.

Ethereum

Na początku listy znajduje się **Ethereum**, ponieważ jest to pierwszy łańcuch bloków, w którym wprowadzono język kompletny w sensie Turinga i maszyny wirtualne. Ten język znacznie różni się od ograniczonych języków skryptowych z bitcoina i wielu innych kryptowalut. Dzięki dostępności kompletnego w sensie Turinga języka Solidity pojawiły się nieskończone możliwości w zakresie rozwoju zdecentralizowanych aplikacji. Ethereum został po raz pierwszy zaproponowany w 2013 r. przez Vitalika Buterina i stanowi publiczny łańcuch bloków do budowania inteligentnych kontraktów i zdecentralizowanych aplikacji. Waluta używana w Ethereum to **ethery**.

MaidSafe

MaidSafe zapewnia sieć **SAFE** (ang. *secure access for everyone*, czyli bezpieczny dostęp dla każdego), która wykorzystuje nieużywane zasoby obliczeniowe użytkowników, takie jak pamięć, moc obliczeniowa i połączenia. Pliki w tej sieci są dzielone na małe porcje danych, szyfrowane i rozprowadzane losowo w sieci. Te dane mogą być pobierane tylko przez ich właścicieli. Ważną innowacją w MaidSafe jest to, że sieć automatycznie odrzuca duplikaty plików. Pomaga to ograniczyć konieczność angażowania dodatkowych zasobów obliczeniowych do zarządzania obciążeniem. Walutą używaną do nagradzania kontrybutorów jest safecoin.

Lisk

Lisk to platforma do obsługi kryptowaluty i budowania aplikacji opartych na łańcuchu bloków. Umożliwia programistom posługiwanie się JavaScriptem do tworzenia zdecentralizowanych aplikacji i umieszczania ich w łańcuchach bocznych. W Lisku do osiągnięcia konsensusu używany jest mechanizm delegowanego dowodu stawki, w którym można wybrać 101 węzłów do zabezpieczenia sieci i proponowania bloków. Na zapleczu używane są środowisko Node.js i język JavaScript, natomiast we frontonie można stosować standardowe technologie: CSS3, HTML5 i JavaScript.

W Lisku walutą w łańcuchu bloków jest **LSK**. Na podstawie Liska powstał też Rise — zdecentralizowana aplikacja i platforma dla waluty cyfrowej. W tej platformie większy nacisk położony jest na bezpieczeństwo systemu.

Bardziej praktyczne wprowadzenie do tych i innych platform znajdziesz w dalszych rozdziałach.

Podsumowanie

W tym rozdziale przedstawiono zagadnienie decentralizacji, która jest podstawową usługą oferowaną przez łańcuchy bloków. Choć koncepcja decentralizacji nie jest niczym nowym, w świecie łańcuchów bloków zyskała nowe znaczenie i spowodowała w ostatnim czasie pojawienie się różnych aplikacji opartych na zdecentralizowanej architekturze.

Rozdział rozpoczął się od wprowadzenia zagadnienia decentralizacji. Dalej omówiono decentralizację w kontekście łańcuchów bloków. Ponadto przedstawiono idee związane z różnymi warstwami decentralizacji w ekosystemie łańcucha bloków oraz kilka nowych koncepcji i pojęć powstałych wraz z pojawieniem się łańcuchów bloków i możliwej dzięki nim decentralizacji. Niektóre z tych pojęć to: DAO, DAC i DApps. W końcowej części opisano kilka przykładowych zdecentralizowanych aplikacji.

W następnym rozdziale opisane zostaną podstawowe koncepcje niezbędne do zrozumienia ekosystemu łańcuchów bloków. Przede wszystkim znajdziesz tam wprowadzenie do kryptografii, która stanowi ważną podstawę technologii łańcuchów bloków.

Skorowidz

A

ABI, application binary interface, 468
acykliczne grafy skierowane skrótów, 57
adres, 31, 346
 URI, 178
adresy
 cyfrowe, 136
 vanity, 140
 w Bitcoinie, 139
 z wielopodpisem, 141
AES, 79
 działanie, 79
 schemat blokowy, 80
algorytm
 AES, 79
 Casper, 292, 519
 Dark Gravity Wave, 204
 DES, 79
 DigiShield, 204
 DSA, 112
 ECC, 90
 ECDSA, 113
 Ethash, 293
 Kimoto Gravity Well, 203
 MIDAS, 205
 PBFT, 451
 RPCA, 437
 RSA, 88
 Scrypt, 218
 SHA, 105
 Triple DES, 79

algorytmy
 dostosowywania poziomu trudności, 210
 generowania skrótów, 210
 krzywych eliptycznych, 87
 osiągania konsensusu, 43, 210, 515
 tworzenia podpisów cyfrowych, 111
 wydobycia, 158
 zmiany celu, 202
analiza, 497, 500
 klucza publicznego, 98
anonimizowanie transakcji, 494
anonimowość, 26, 205
 wbudowana, 207
API, 187, 306
aplikacja
 KYC-Chain, 63
 Lazooz, 64
 MetaCoin, 377
 OpenBazaar, 63
aplikacje
 CorDapp, 428
 w łańcuchu bloków, 414
 zdecentralizowane, 306
architektura Raspberry Pi, 460
arkusz zleceń, 184
arytmetyka modularna, 70
ATP, atomic transport protocol, 440

atrybuty
 ekonomiczne, 121
 sprzedaży, 121
 stron, 121

B

BaaS, blockchain as a service, 512
badania, 505
 na łańcuchami bloków, 515
 nad metodami formalnymi, 511
 nad sprzętem, 510
 w kryptografii, 510
baza danych, 413
 BigchainDB, 448
bezpieczeństwo, 35, 39, 309, 495, 511
 inteligentnych kontraktów, 495
bezpieczne obliczenia, 493
bezpieczny
 kontener, 411
 rejestr, 411
biblioteka, 352
 Bitcoinj, 193
 Libbitcoin, 193
 Pycoin, 193
 Web3, 357
BigchainDB, 448
BIP, Bitcoin improvement proposal, 179, 485, 507

- Bitcoin, 26, 125, 127
 adresy, 139
 Cash, 183
 Gold, 183
 klucze
 prywatne, 136
 publiczne, 138
 transakcje, 141
 Unlimited, 183
- Bitcoin-cli, 188
 Bitcoin-NG, 490
 Bitcoin-qt, 188
 BitcoinD, 187
 blockchain, *Patrz* łańcuch
 bloków
 blok, 30, 32, 284
 początkowy, 30, 153, 286
 błędy
 bizantyjskie, 48
 powodujące wyłączenie, 48
 broker komunikatów, 426
 buforowanie oportunistyczne, 447
 Burrow, 401
- ## C
- całkowita podaż pieniądza, 211
 cechy łańcucha bloków, 37
 Cello, 402
 cena
 bitcoinów, 128
 paliwa, 258
 certyfikat TLS, 212
 ciało, 69
 liczb pierwszych, 70
 skończone, 69
 ciemna strona, 513
 CME, Chicago Mercantile
 Exchange, 125
 CoinJoin, 494
 Colored Coins, 207
 Composer, 402
 Corda, 424
 architektura, 424
 komponenty, 426
 przepływy, 426
 skarbcę, 428
 środowisko
 programistyczne, 429
 transakcje, 425, 427
- udostępnianie mapy sieci, 427
 usługi
 notarialne, 427
 wyroczni, 427
 węzły, 426
 zapewnianie konsensusu, 425
 zarządzanie
 uprawnieniami, 427
- CorDapp, Corda distributed
 application, 428
 Crypto Enclave, 444
 CSS, 364
 cykl życia transakcji, 121, 142, 417
 czas
 do zmniejszenia nagrody, 210
 generowania bloków, 210
 wydobywania bloków, 486
- ## D
- dane, 260
 DAP, decentralized
 anonymous payment, 224
 debugowanie, 337
 decentralizacja, 34, 51
 oparta na
 współzawodnictwie, 54
 definicja Bitcoina, 129
 definicje łańcucha bloków, 28
 definiowanie funkcji, 353
 delegowany dowód stawki, 46
 DES, 79
 deszyfrowanie, 86, 89
 DHT, distributed hash tables, 446
 dodawanie
 bloków, 34
 kontraktów, 323, 343, 358, 471
 punktów, 91, 92
 dokumenty BIP, 179, 181
 dostawca
 MSP, 413
 usług kryptograficznych, 413
 dostępność, 47
- dowód
 aktywności, 47, 202
 depozytu, 46, 201
 pracy, 26, 45, 157, 197, 198
 pracy wielokrotnego
 użytku, 127
 przestrzeni, 47
 składowania, 47, 201
 spalania, 42, 196, 202
 stawki, 42, 46, 201, 488
 TLSNotary, 392
 upływu czasu, 46
 wieku środków, 201
 własności, 196
 ZKP, 127, 476, 492
 znaczenia, 46
- Drivechain, 443
 drzewa skrótów, 109
 drzewo trie, 110, 260
 DSA, Digital Signature
 Algorithm, 112
 dwustronne przyczepianie, 443
 dyrektywa pragma, 355
 działanie
 AES, 79
 Ethereum, 250
 łańcucha bloków, 33
 maszyny EVM, 269
 dziedziczenie, 351
 dzielenie sekretu, 26
- ## E
- EAA, Enterprise Ethereum
 Alliance, 310
 ECC, 90, 94
 klucz prywatny, 100
 ECDSA, 113
 efekt sieciowy, 129
 eGaaS, electronic government
 as a service, 512
 EIP, Ethereum improvement
 proposals, 507
 eksplorator bloków, 253, 326, 328
 elektroniczne pieniądze, 26
 elementy łańcucha bloków, 31
 eliminowanie pośrednictwa, 53
 emisje ICO, 230
 enklawa, 493

e-pieniądze, 26
 Eris, 449
 Ethash, 293
 Ethereum, 64, 196, 247, 275
 adresy, 256
 bloki, 284
 funkcja zmiany stanu, 250
 handel, 309
 inwestycje, 309
 klucze, 256
 komponenty ekosystemu, 255
 komunikaty, 258
 konta, 256
 łańcuchy bloków, 284
 opłaty, 290
 oprogramowanie klienckie, 298
 portfele, 298
 protokoły pomocnicze, 307
 sieci, 254, 255
 prywatne, 313
 testowe, 312
 składowanie stanu, 264
 specyfikacja techniczna, 248
 środowisko
 programistyczne, 311
 transakcje, 258
 uruchamianie portfela, 322
 EthereumJS, 338
 EVM, Ethereum Virtual Machine, 333
 Explorer, 402

F

Fabric, 400, 407
 FBA, federated byzantine agreement, 441
 filtr Blooma, 174, 266
 finalizacja bloku, 287
 finanse, 478
 forki, 290
 format
 minikluczy prywatnych, 137
 WIF, 137
 FPGA, Field Programmable Gate Array, 161
 fronton, 364, 377

funkcja, 352, 353
 dodawania punktów, 273
 iteratora, 271
 kopiowania danych, 273
 odzyskiwania klucza publicznego, 272
 potęgowania modułu, 273
 SaveIdeaHash, 387, 388

funkcje
 Message Digest, 104
 modyfikujące, 354
 rezerwowe, 354
 skrótu, 102, 107, 273
 wewnętrzne, 348
 zewnętrzne, 348

G

Ganache, 338
 transakcje, 375
 wyświetlanie kont, 376
 generator liczb pseudolosowych, 103
 generowanie
 adresów, 228
 klucza prywatnego, 89, 100
 publicznego, 89
 kryptowalut, 38
 liczby względnie pierwszej, 89
 modułu, 89
 podpisów, 86
 giełda, 119
 bitcoinów cex.io, 184
 górnik, 33, 156, 291
 graficzny interfejs użytkownika, 188
 grupa, 69
 abelowa, 69
 cykliczna, 70, 95

H

handel, 119, 309
 litecoinami, 219
 namecoinami, 213
 primecoinami, 221
 zcashami, 225
 HDL, Hardware Description Language, 161

hierarchiczne portfele deterministyczne, 176
 historia łańcucha bloków, 26
 HMAC, 108
 hologram, 137
 HTML, 364
 Hyperledger, 399
 architektura wzorcowa, 403
 cele projektowe, 405
 jako protokół, 403
 projekty, 399
 wymogi, 405
 Hyperledger Fabric, 408
 cykl życia transakcji, 417
 obraz platformy, 415
 protokół P2P, 410
 rozproszony rejestr, 409
 usługi, 408
 węzły, 412

I

IBC, inter blockchain communication, 449
 IBLT, invertible Bloom lookup tables, 486
 ICO, initial coin offering, 230
 IDE, 335
 identyfikacja obywateli, 477
 identyfikator sieci, 314
 ILCP, interledger control protocol, 441
 ILP, interledger protocol, 441
 ILQP, interledger quoting protocol, 441
 implementacja, 507
 kontraktu chaincode, 414
 Indy, 401
 informacje
 o bloku, 280
 o środowisku, 279
 o transakcji, 134, 377
 inicjowanie platformy Truffle, 371
 instalowanie
 biblioteki web3.js, 365
 Bitcoina, 187, 189
 klienta, 299
 Eth, 299
 Parity, 304
 kompilatora

instalowanie
 w Linuksie, 333
 w macOS, 333
 systemu IPFS, 394
 środowiska Node.js, 465
 instrument bazy, 121
 instytucje rządowe, 474
 integralność, 71
 inteligentne kontrakty, 33, 38,
 60, 233, 271, 413, 420, 515
 inteligentne
 prawa własności, 38
 wyrocznie, 243
 interakcja z kontraktem, 377
 interfejs
 ABI, 468
 API, 306, 411
 bitcoin-cli, 192
 CLI, 411
 użytkownika klienta Parity,
 306
 internet rzeczy, 241, 453
 oparty na łańcuchu bloków,
 459
 inwestycje w bitcoiny, 184
 IPFS, 394
 Iroha, 400

J

JavaScript, 364
 jednostka certyfikująca, 112
 jednostki waluty ETH, 267
 język programowania, 276, 510
 DSL, 240
 GPL, 240
 HDL, 161
 JULIA, 269
 LLL, 276, 332
 Mutan, 332
 Pact, 434
 script, 32, 146
 Serpent, 276, 333
 Solidity, 276, 333, 344–356
 Vyper, 276, 333

K

Kadena, 432
 kanały, 412
 stanu, 487, 493

klient, 412
 Bitcoin Core, 171, 187, 189
 Eth, 298
 Ethereum, 298
 Geth, 298, 302, 464
 Parity, 298, 304
 Pyethapp, 298
 klienty uproszczone, 299
 klucze
 prywatne, 88, 96, 136
 publiczne, 88, 97, 138
 kod
 bajtowy, 276
 źródłowy Litecoin'a, 220
 kodowanie Base58Check, 139
 kody
 MAC, 108
 mnemoniczne, 281
 operacji, 146, 277
 przedstawiania, 283
 systemowych, 284
 QR, 132, 251
 uwierzytelniania
 wiadomości, 102
 kolejka LIFO, 268
 komentarze, 356
 kompilator, 333
 języka Solidity, 333
 komponenty
 szkieletu, 412
 transakcji, 121
 kompresowanie komunikatów,
 103
 komunikacja, 58
 IPC, 321
 komunikaty, 262
 konfigurowanie
 kodu źródłowego, 190
 pliku bitcoin.conf, 190
 sieci prywatnej, 313
 węzła, 463
 węzła w Raspberry Pi, 463
 konsensus, 37, 43, 292
 federacyjny, 46
 w Hyperledger Fabric, 416
 w łańcuchu bloków, 45
 konsola Geth, 302
 konsorcja, 508
 konstruktor, 354
 kontrakt chaincode, 414

kontrakty, 150
 natywne, 272
 ricardiańskie, 237
 kontrola granic, 474
 kopalnie, 163, 298
 koparki, 296
 korzeń
 drzewa skrótów, 30
 magazynu danych, 265
 koszty, 35
 kryptoekonomia, 509
 kryptografia, 70
 asymetryczna, 85
 badania, 510
 klucza publicznego, 26, 85
 krzywej eliptycznej, 259
 symetryczna, 67, 74
 kryptowaluta, 33
 ethereum, 196
 Litecoin, 217
 model łańcucha skrótów,
 199
 Namecoin, 211
 Primecoin, 220
 wartość rynkowa, 198
 Zcash, 195, 223
 kryptowaluty alternatywne,
 195, 209
 krzywe eliptyczne, 87
 kwota do zwrotu, 264

L

liczby całkowite, 345
 LIFO, Last In, First Out, 146,
 268
 limit paliwa, 259
 Lisk, 65
 lista
 transakcji, 325
 unikatowych węzłów, 436
 Litecoin, 217
 handel, 219
 kod źródłowy, 220
 portfel, 220
 wydobywanie, 220
 literały, 347
 całkowitoliczbowe, 347
 szesnastkowe, 347
 znakowe, 347

LKIF, legal knowledge
interchange format, 236
logarytm dyskretny, 87, 94
lokalizacja danych, 349

Ł

łańcuch bloków, 21, 27, 151,
284, 431, 453
Bitcoin Gold, 183
Ethereum, 247, 249
Kadena, 432
Quorum, 444
Rootstock, 442
łańcuchy
bi-twin, 220
bloków
alternatywy, 511
badania, 515
bez tokenów, 43
bezpieczeństwo, 495
ciemna strona, 513
dla przedsiębiorstw, 504
edukacja, 509
prywatne, 504
publiczne, 41
specyficzne, 503
standaryzacja, 506
usprawnienia, 507
z tokenami, 43
boczne, 42, 443, 488
przyczepione, 197
Cunninghama, 220
drzewiaste, 489
łączniki, 441

M

magiczne wartości, 168
MaidSafe, 64, 447
Mainnet, 254
manipulowanie rynkiem, 122
maszyna
stanowa, 33
wirtualna, 32
EVM, 267, 269, 333
RVM, 443
mechanizm
osiągania konsensusu, 43,
292
SBTF, 409
ustalania kluczy, 87

mechanizmy
kryptograficzne, 73
oparte na reputacji, 46
menedżer
sieci, 444
transakcji, 444
Message Digest, 104
metadane, 144
bloku, 410
MetaMask, 340
metody decentralizacji, 53
Microsoft Visual Studio, 520
MimbleWimble, 494
miniklucz, 137
moc obliczeniowa, 59
model
aplikacji, 416
BaaS, 512
generowania podpisów, 86
modyfikatory funkcji, 355
modyfikowalność transakcji,
151
MultiChain, 448
multimedia, 481
Multisig, 148

N

nagłówek bloku, 285
nagrody, 157
za wydobycie bloku, 210
Namecoin, 211
generowanie rekordów,
215
handel, 213
pozyskiwanie, 213
narzędzie, 331, 520
Apache Kafka, 409
EthereumJS, 338
Ganache, 338, 373
MetaMask, 340
nheqminer, 229
Node, 338
OpenSSL, 67, 82
Oyente, 499
TestRPC, 339
Truffle, 342
Wireshark, 172
NG, next generation, 517
niemodyfikowalność, 35, 39
nieprzekazywalne zadania, 202

niezabezpieczone połączenie
RPC, 321
niezaprzeczalność, 72
Node, 338, 465
nominały bitcoinów, 136

O

obiekty stanowe, 424
obliczanie skrótów, 160
obraz łańcucha bloków, 30
obsługa
audytów, 406
transakcji, 418
odporność
na kolizje, 103
na podział, 47
odwzorowania, 349
ogólne atrybuty, 121
ograniczenia
Bitcoina, 205
funkcji kryptograficznych,
515
opcje
analiz, 497
w kliencie Geth, 315
OpenSSL
algorytm ECC, 99
algorytm RSA, 96
ECDSA, 115
funkcje skrótu, 107
podpis cyfrowy, 114
operacje
arytmetyczne, 277
dodawania
elementów na stosie, 281
wpisów dziennika, 282
duplikowania, 282
kryptograficzne, 278
logiczne, 278
przestawiania, 282
systemowe, 283
w zdecentralizowanych
aplikacjach, 63
opieka zdrowotna, 478
opłata
transakcyjna, 142
za paliwo, 290
oprogramowanie klienckie, 298
osiąganie konsensusu, 416, 421,
515

- osoby przewidujące zlecenia, 122
- OTP, optimistic transport protocol, 440
- OWPS, open web payment scheme, 440
- Oyente, 499
- ## P
- P2P, peer-to-peer, 28
- PaaS, platform as a service, 482
- paliwo, 289
- para kluczy RSA, 96
- parowanie punktów krzywej eliptycznej, 273
- Pay
 - to Public Key Hash, 147
 - to Script Hash, 147
- PBFT, practical byzantine fault-tolerance, 46, 451
- pierścień, 70
- pisanie kodu kontraktu, 343
- Plasma, 491
- platforma
 - DAPPLE, 521
 - Embark, 521
 - Eris, 449
 - Ethereum, 64
 - Hyperledgea Fabric, 405
 - Lisk, 65
 - MaidSafe, 64
 - Meteor, 521
 - MultiChain, 448
 - Swarm, 308
 - Truffle, 371
 - uPort, 521
- platformy
 - do decentralizacji, 64
 - inteligentnych kontraktów, 37
 - programistyczne, 331, 371
- plik bitcoin.conf, 190
- płatności, 178
- podłańcuchy, 489
- podpis, 259
 - cyfrowy, 111
 - funkcji, 353
- podstan transakcji, 264
- podwajanie punktów, 93, 94
- PoET, proof of elapsed time, 418
- pokwitowania transakcji, 266
- pole inicjujące, 259
- połączenie RPC, 321
- poprawność bloków, 287
- portfel, 175, 298
 - Blockchain, 132
- portfele
 - deterministyczne, 175
 - internetowe, 177
 - mobilne, 177
 - niedeterministyczne, 175
 - pamięciowe, 176
 - papierowe, 176
 - sprzętowe, 176
- pośrednictwo, 53
- poufne transakcje, 494
- poufność, 71, 405, 493
- poziom
 - decentralizacji, 54
 - opłat, 290
 - osiągania konsensusu, 484
 - sieci, 484
 - składowania danych, 485
 - trudności bloku, 288
 - widoku, 485
- pozycja, 120
- pozyskiwanie namecoinów, 213
- Primecoin, 220
 - handel, 221
 - wydobywanie, 221
- problem
 - bizantyjskich generałów, 24
 - logarytmu dyskretnego, 94
- problemy
 - techniczne, 508
 - z centralizacją, 515
- procesor, 160, 162
 - graficzny, 161, 162
- program Tendermint, 448
- programowanie, 192
- projekt
 - Bitcoin-NG, 517
 - Bletchley, 519
 - Burrow, 401
 - Casper, 519
 - Cello, 402, 517
 - CollCo, 517
 - Composer, 402
 - Explorer, 402
 - Fabric, 400
 - Falcon, 519
 - Hawk, 518
 - Indy, 401
 - INFURA, 522
 - Iroha, 400
 - Qtum, 517
 - Quilt, 402
 - Sawtooth Lake, 400
 - SETLCoin, 518
 - Solidus, 517
 - systemu rozproszonego, 25
 - TEEChan, 518
 - Town-Crier, 518
 - Zcash dla Ethereum, 516
- protokoły
 - mieszania, 206, 207
 - oparte na Bitcoinie, 207
 - pomocnicze, 307
 - zaawansowane, 181
- protokół
 - AMQP, 426
 - ATP, 440
 - Bitcoin Cash, 183
 - Bitcoin Unlimited, 183
 - GHOST, 292, 486
 - IBC, 449
 - ILCP, 441
 - ILP, 441
 - ILQP, 441
 - Interledger, 440
 - NG, 517
 - OTP, 440
 - OWPS, 440
 - P2P, 410
 - SCP, 441
 - SegWit, 181
 - Solidus, 518
 - SPSP, 440
 - Swarm, 396
 - TLS, 471
 - TMSP, 448
 - UTP, 440
 - Whisper, 307
- protokoły
 - osiągania konsensusu, 437
 - ślepych podpisów, 118
 - wykrywania węzłów, 170, 173
- prywatność, 205, 405, 491

prywatny łańcuch bloków, 41,
42, 488
przeciwbraz, 103
przeglądarka
 MetaMask, 520
 Mist, 299, 321, 322
 dodawanie kontraktów,
 323
przejrzystość, 34
przekazywanie
 bloków, 490
 wartości, 38
przenośność, 407
przepływ danych, 393
przeskakiwanie kopalni, 203
przesyłanie płatności, 130
przetwarzane konta, 264
przyczepione łańcuchy boczne,
197
przyspieszenie operacji, 35
publiczne łańcuchy bloków, 41
pula transakcji, 143, 183

Q

Quilt, 402
Quorum, 444
 transakcje, 445
QuorumChain, 444

R

Raspberry Pi, 460
 klient Geth, 464
 konfigurowanie węzła, 463
 sterowanie diodą, 473
 układ, 467
regulacje, 512
rejestr
 Corda, 424
 kryptograficznie
 bezpieczny, 28
 kryptograficzny
 Tezos, 445
 oparty na uprawnieniach,
 42
reputacja, 46
Ripple, 436
 osiąganie konsensusu, 438
 transakcje, 438

rodzaje
 stawek, 201
 transakcji, 147
rodzina transakcji, 418
Rootstock, 442
rozkład liczb całkowitych, 87
rozliczalność, 26, 73
rozproszone
 rejestry, 28, 40, 396, 409
 tablice mieszające, 57, 110
rozrachunek potransakcyjny,
479
rozwój
 alternatywnych
 kryptowalut, 209
 technologii, 21
równanie Weierstrassa, 90
RPC, remote procedure call,
172
RSA, 88
 deszyfrowanie, 99
 klucz prywatny, 96
 klucz publiczny, 97
 podpisy cyfrowe, 111
 szyfrowanie, 99
RVM, Rootstock virtual
 machine, 443
rynk
 dłużne, 119
 kapitałowe, 119
 pieniężne, 119
rząd, 69

S

saldo, 265
Sawtooth Lake, 400, 418
 inteligentne kontrakty, 420
 osiąganie konsensusu, 421
 środowisko
 programistyczne, 421
SBTF, Simple Byzantine Fault
 Tolerance, 409
schemat
 analizy decentralizacji, 56
 DAP, 224
 MimbleWimble, 494
schematy kodowania, 118
SCP, stellar consensus
 protocol, 441
Segregated Witness, 181

serwer
 HTTP, 327
 obsługi protokołu kopalni,
 167
SHA, 105
 działanie algorytmu, 106
SHA-256, 106
SHA-3, 107
Sharding, 487
sieci
 internetu rzeczy, 454
 prywatne, 254, 313
 dodawanie kontraktów,
 323
 eksplorator bloków, 326
 identyfikator sieci, 314
 katalog na dane, 315
 plik początkowy, 314
 uruchamianie, 316
 uruchamianie
 przełączarki Mist, 321
 testowe, 312
 w topologii siatki, 58
sieć
 Bitcoin, 167
 Bitcoin Lightning, 180
 Ethereum, 254, 255
 Mainnet, 254
 P2P, 32
 regtest, 191
 Ripple, 436
 SAFE, 64
 Stellar, 441
 Testnet, 254
signcryption, 117
skalowalność, 309, 406, 483,
516
skarbcę, 428
składowanie
 danych, 57, 447
 MaidSafe, 447
 Storj, 446
 w chmurze, 446
 rejestr, 411
skompresowane drzewa trie,
110
skrót kodu, 265
skrypt P2PKH, 148
Solidity, 344–56
Solidus, 517
specyfikatory dostępu, 354

- spójność, 47, 48
 sprawdzanie
 poprawności, 146, 263, 497, 498
 bloków, 287
 transakcji, 37, 150
 SPSP, simple payment setup protocol, 440
 SPV, simple payment verification, 299
 stan
 konta, 265
 maszyny, 271
 po transakcji, 266
 świata, 264, 413
 standard ERC20, 231
 standaryzacja, 506
 startupy, 505
 Stellar, 441
 stopa procentowa, 211
 Storj, 446
 stos LIFO, 146
 Stratis, 520
 struktura, 348
 bloku, 31, 151, 410
 danych transakcji, 143
 łańcucha bloków, 31
 nagłówka bloku, 151
 sterująca, 350
 transakcji coinbase, 149
 wejścia transakcji, 145
 wyjścia transakcji, 145
 symbole matematyczne, 249
 synchronizacja nagłówków i bloków, 171
 system
 B-money, 126
 IPFS, 57, 394
 systemy
 alternatywne, 195
 rozproszone, 24, 52
 składania zleceń, 120
 wydobywania, 160
 zdecentralizowane, 53
 szablony inteligentnych kontraktów, 239
 szkielec, 407
 sztuczna inteligencja, 37
 szybkość
 obliczania skrótów, 160
 przetwarzania transakcji, 182
 szyfrowanie, 86, 89
 homomorficzne, 117, 492
 szyfry, 68
 blokowe, 75
 strumieniowe, 75
- ## Ś
- ślepe podpisy cyfrowe, 26
 środowisko
 Ethereum, 311
 IDE, 335
 IDE Remix, 468, 497
 Node.js, 465
 REPL, 434
 TEE, 494
 uruchomieniowe, 269
 świat
 prawa, 237
 rozliczeń, 237
- ## T
- tablice, 348
 DHT, 110, 446
 IBLT, 486
 mieszające, 110
 technika ZKP, 117
 technologia DLT, 40
 technologie teleinformatyczne, 52
 TEE, Trusted Execution Environment, 494
 Tendermint, 448
 Core, 448
 Socket Protocol, 448
 terminale kasowe, 179
 Testnet, 254
 TestRPC, 339
 testy, 343
 porównawcze, 295
 Tezos, 445
 The DAO, 244
 TLS, transport layer security, 212, 471
 TMTO, time-memory trade-off, 217
 tokeny, 231
 Ether, 267
 tożsamość, 406
 transakcje, 30, 121, 134, 252, 410, 418
 coinbase, 149
 deterministyczne, 406
 finansowe, 118
 początkowe, 237
 poufne, 494
 tworzące kontrakty, 258, 261
 wywoływania
 komunikatów, 258, 261
 związane
 z bezpieczeństwem, 439
 z kontem, 439
 z płatnościami, 439
 ze zleceniami, 439
 trendy, 503
 trójkąt Zooko, 59
 Truffle, 342, 371
 dodawanie kontraktu, 471
 inicjowanie platformy, 371
 interakcja z kontraktem, 377
 konsola, 378
 metody, 378
 wywoływanie metod, 391
 tryb
 CBC, 77
 CTR, 77
 ECB, 77
 generowania strumienia
 kluczy, 78
 kryptograficznych skrótów, 78
 szyfrowania blokowego, 76
 uwierzytelniania
 komunikatu, 78
 twierdzenie CAP, 25, 47
 tworzenie
 kodów MAC, 108
 kont w kliencie Parity, 306
 kontraktu, 386
 obiektu web3, 368
 podpisów cyfrowych, 111
 typ danych address, 346
 typ logiczny, 345
 typy
 funkcyjne, 347
 łańcuchów bloków, 39
 referencyjne, 348
 sieci, 52

U

ubezpieczenia, 478
 układ
 ASIC, 161, 163
 FPGA, 162
 unikatowość, 39
 uproszczenie obecnych modeli, 35
 URI, uniform resource identifier, 178
 uruchamianie
 interfejsu, 192
 portfela Ethereum, 322
 przeglądarki Mist, 321
 sieci prywatnej, 316
 węzła, 190, 191
 usługa
 chaincode, 411
 Counterparty, 208
 Oraclice, 392
 usługi
 łańcucha bloków, 409
 notarialne, 427
 osiągania konsensusu, 409
 udostępniania mapy sieci, 427
 wroczni, 427
 zarządzania członkostwem, 408
 zarządzania
 uprawnieniami, 427
 UTP, universal transport protocol, 440
 uwierzytelnianie, 71
 jednostek, 72
 źródła danych, 72
 użyte paliwo, 266

W

warstwa, 36
 aplikacji, 440, 456
 międzyrejestrowa, 441
 obiektów fizycznych, 455
 rejestru, 441
 sieci, 455
 transportowa, 440
 urządzeń, 455
 zarządzania, 456

wartość, 259
 nonce, 30, 258, 265
 pusta, 148
 WASM, WebAssembly, 268
 Web3, 357
 dodawanie kontraktów, 358
 instalowanie biblioteki, 365
 skrypt instalacyjny, 360
 tworzenie obiektu, 368
 wejścia, 145
 wersje Ethereum, 248
 węzeł, 25, 33, 291
 Bitcoina, 188
 bizantyjski, 25
 węzły
 akceptujące, 412
 pełne, 167
 porządkujące, 412
 przekazujące, 412
 sprawdzania poprawności, 436
 SPV, 167
 statyczne, 316
 użytkownika, 436
 zawierające, 412
 Whisper, 307
 wiek pieniędzy, 211
 wielkość bloku i transakcji, 211
 wiersz poleceń, 67
 uruchamianie interfejsu, 192
 WIF, Wallet Import Format, 137
 Wireshark, 172
 wizualizacja przepływu środków, 134
 współdziałanie, 406
 współzawodnictwo, 54
 wydobywanie, 48, 156, 160, 294, 295
 łączone, 212
 primecoinów, 222
 z użyciem procesora graficznego, 228
 zcashów, 225
 wyjścia, 145
 wykonywanie transakcji, 263

wyliczenia, 347
 wrocznie, 241, 392
 zdecentralizowane, 242
 wysoka dostępność, 35
 wysyłanie bitcoinów, 132
 wywołania, 263
 RPC, 172

Z

zaciemnianie, 491
 kodu, 516
 zadania górników, 156
 zapewnianie bezpieczeństwa, 39
 zapobieganie przestępstwom finansowym, 480
 zapytania o dane, 407
 zarządzanie
 prawami cyfrowymi, 38, 420
 zleceniem, 120
 zasilanie konta, 303
 zastrzeżone łańcuchy bloków, 42
 zatrudnienie, 509
 zaufanie, 34
 zautomatyzowana kontrola graniczna, 476
 zbiór, 69
 Zcash, 195, 223
 handel, 225
 wydobywanie, 225
 zdarzenia, 350, 411
 zdecentralizowane
 aplikacje, 62, 63
 korporacje autonomiczne, 62
 organizacje autonomiczne, 61
 społeczności autonomiczne, 62
 zdecentralizowany ekosystem, 60
 zestaw
 autodestrukcji, 264
 dzienników, 264
 wpisów z dziennika, 266
 ZKP, zero-knowledge proof, 117

zlecenie, 120
 po cenie rynkowej, 120
 transakcji, 121
zmiennie globalne, 349
zmniejszenie kosztów, 35

zużycie energii, 512
zwiększenie
 wielkości bloku, 485
 wydajności łańcuchów
 bloków, 485

Ż

żądania POST, 363

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Blockchain: w sieci węzłów rozproszonych nigdy nic nie zaginie!

Warto dobrze zrozumieć działanie łańcucha bloków. Ta nowatorska technologia intryguje badaczy, technologów i świat finansjery. Najlepiej jest znana z powiązań z bitcoinem i innymi kryptowalutami, jednak to rozwiązanie ma wielkie znaczenie również w finansach, administracji, multimediami i wielu innych branżach. Przejrzyste i kryptograficznie bezpieczne przechowywanie niemodyfikowalnego zapisu transakcji jest właściwością, która decyduje o wielkiej atrakcyjności łańcucha bloków. Bez wątpienia warto dobrze zaznajomić się z tą technologią i nauczyć się wykorzystywać ją w praktyce.

Ta książka jest zaktualizowanym i uzupełnionym wydaniem świetnego przewodnika po świecie blockchaina; skorzysta z niej zarówno programista, jak i prezes konstruujący strategię swojej firmy. Znalazło się tu wyczerpujące omówienie technicznych podstaw łańcuchów bloków i systemów rozproszonych. Przedstawiono mechanizmy związane z kryptowalutami i pisaniem aplikacji wykonywanych w zdecentralizowanej maszynie wirtualnej w łańcuchu bloków Ethereum. Pokazano też inne rozwiązania z tego obszaru, w tym biznesowe platformy łańcuchów bloków rozwijanych w ramach projektu Hyperledger. Okazuje się, że technologia ta znakomicie uzupełnia internet rzeczy, a zwiększanie skalowalności łańcucha bloków daje nowe, fascynujące możliwości.

W tej książce:

- podstawy przetwarzania rozproszonego, decentralizacja procesów i systemów
- wprowadzenie do kryptografii i sieci bitcoina
- techniczne zagadnienia związane z Ethereum i inteligentnymi kontraktami
- wprowadzenie do Web3 i do projektu Hyperledger
- informacje o zastosowaniu łańcucha bloków w różnych dziedzinach
- aktualne prognozy dotyczące przyszłości technologii łańcucha bloków

Imran Bashir od wielu lat rozwija oprogramowanie, tworzy architekturę rozwiązań technicznych, zarządza infrastrukturą i usługami informatycznymi. Jest członkiem prestiżowych stowarzyszeń IEEE i BCS. Od 16 lat realizuje duże projekty informatyczne dla sektora publicznego i branży usług finansowych. Obecnie pracuje dla jednego z banków inwestycyjnych w Londynie, gdzie jest wicedyrektorem działu technologii.

Helion 	<i>Sprawdź nasze szkolenia!</i>	KOD KORZYŚCI <i>Sięgnij po więcej!</i> 	
 helion.pl	 AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	ISBN 978-83-283-4957-5	
 0 801 339900			
 0 601 339900		9 788328 349575	
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 99,00 zł	

Packt