OKIEM EKSPERTA

Blockchain

Przewodnik po łańcuchu bloków — od kryptografii po DeFi i NFT

Wydanie IV

Imran Bashir

Helion

packt

Tytuł oryginału: Mastering Blockchain: A technical reference guide to the inner workings of blockchain, from cryptography to DeFi and NFTs, 4th Edition

Tłumaczenie: Tomasz Walczak

ISBN: 978-83-289-0391-3

First published in the English language under the title 'Mastering Blockchain - Fourth Edition' – (9781803241067).

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/blocz4
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
https://ftp.helion.pl/przyklady/blocz4.zip

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

O autorze **19**

O korektorach merytorycznych **20**

Przedmowa **21**

ROZDZIAŁ 1
ABC łańcucha bloków **25**

Rozwój technologii łańcucha bloków 25
Droga do dojrzałości 26
Rosnące zainteresowanie 27
Systemy rozproszone 28
Twierdzenie CAP 30
Twierdzenie PACELC 32
Historia łańcucha bloków 33
Bitcoin 34
Elektroniczne pieniądze 34
Wprowadzenie do łańcucha bloków 36
Architektura łańcucha bloków 37
Uniwersalne elementy łańcucha bloków 40
Działanie łańcucha bloków 44
Zalety i funkcje łańcucha bloków 45
Ograniczenia technologii łańcucha bloków 47
Typy łańcuchów bloków 49
Rozproszone rejestry 50
Wspólny rejestr 50
Publiczne łańcuchy bloków 50
Prywatne łańcuchy bloków 50
Częściowo prywatne łańcuchy bloków 51
Rejestr oparty na uprawnieniach 51
W pełni prywatne i zastrzeżone łańcuchy bloków 51
Łańcuchy bloków z tokenami 52
Łańcuchy bloków bez tokenów 52
Łańcuchy bloków warstwy pierwszej 52
Łańcuchy bloków warstwy drugiej 53
Podsumowanie 54

ROZDZIAŁ 2
Decentralizacja 55
Wprowadzenie do decentralizacji 55
Metody decentralizacji 59
Eliminowanie pośrednictwa 60
Decentralizacja oparta na współzawodnictwie 60
Ilościowe ujęcie decentralizacji 61
Zalety decentralizacji 62
Ocena wymagań 63
Decentralizacja całego ekosystemu 65
Składowanie danych 65
Komunikacja 66
Moc obliczeniowa 67
Decentralizacja w praktyce 68
Inteligentne kontrakty 69
Autonomiczne agenty 69
Zdecentralizowane organizacje 69
Zdecentralizowane organizacje autonomiczne 69
Zdecentralizowane korporacje autonomiczne 70
Zdecentralizowane społeczności autonomiczne 70
Zdecentralizowane aplikacje (DApps) 71
Wymogi stawiane zdecentralizowanym aplikacjom 73
Operacje w zdecentralizowanych aplikacjach 74
Projekt zdecentralizowanej aplikacji 74
Innowacyjne trendy 75
Zdecentralizowany internet 76
Web 1 76
Web 2 76
Web 3 77
Podsumowanie 77

ROZDZIAŁ 3
Kryptografia symetryczna 78
Wprowadzenie do kryptografii 78
Usługi zapewniane dzięki kryptografii 79
Podstawowe mechanizmy kryptograficzne 81
Podstawowe mechanizmy kryptografii bez kluczy 82
Podstawowe mechanizmy w kryptografii z kluczami symetrycznymi 94
AES 101
DES 101
Działanie algorytmu AES 102
Szyfrowanie i deszyfrowanie z użyciem algorytmu AES 103
Podsumowanie 105

ROZDZIAŁ 4
Kryptografia asymetryczna 106
Podstawy matematyczne 106
Kryptografia asymetryczna 107
Klucze publiczny i prywatny 108
Algorytmy w kryptografii asymetrycznej 109
System IES 110
Wprowadzenie do algorytmu RSA 111
Szyfrowanie i deszyfrowanie z użyciem algorytmu RSA 112
Wprowadzenie do ECC 114
Matematyka wykorzystywana w ECC 115
Problem logarytmu dyskretnego w ECC 120
Generowanie kluczy za pomocą algorytmu ECC 123
Podpisy cyfrowe 125
Algorytm tworzenia podpisów cyfrowych za pomocą RSA 126
Algorytm ECDSA 128
Różne typy podpisów cyfrowych 131
Zagadnienia kryptograficzne i technologia łańcucha bloków 136
Szyfrowanie homomorficzne 136
Dzielenie sekretu 137
Systemy składania zobowiązań 137
Dowody ZKP 138
Schematy kodowania 144
Funkcje VRF 145
Podsumowanie 145

ROZDZIAŁ 5
Algorytmy osiągania konsensusu 147
Wprowadzenie do konsensusu 147
Odporność na błędy 148
FLP 149
Analizowanie i projektowanie 150
Model 151
Procesy 151
Założenia dotyczące czasu 151
Klasyfikacja 152
Algorytmy 153
Algorytmy CFT 153
Algorytmy BFT 158
Wybór algorytmu 185
Nieodwracalność 186
Szybkość, wydajność i skalowalność 186
Podsumowanie 187

ROZDZIAŁ 6
Architektura Bitcoina 188
Wprowadzenie do Bitcoina 188
Klucze kryptograficzne 190
Klucze prywatne w Bitcoinie 190
Klucze publiczne w Bitcoinie 191
Adresy 192
Typowe adresy Bitcoin 193
Zaawansowane adresy Bitcoin 195
Transakcje 195
Transakcje coinbase 196
Cykl życia transakcji 197
Struktura danych transakcji 199
Język Script 202
Błędy związane z transakcjami 207
Łańcuch bloków 208
Struktura 208
Blok początkowy 210
Bloki nieaktualne i osierocone 210
Forki 211
Cechy Bitcoina 212
Górnicy 213
Dowód pracy 214
Systemy wydobywania 216
Kopalnie 218
Sieć 219
Typy komunikatów 220
Oprogramowanie klienckie 225
Filtry Blooma 226
Portfele 227
Podsumowanie 230

ROZDZIAŁ 7
Bitcoin w praktyce 231
Bitcoin w rzeczywistym świecie 231
Płatności w bitcoinach 232
Innowacje w Bitcoinie 235
Dokumenty BIP 235
Zaawansowane protokoły 236
Rozszerzone protokoły oparte na Bitcoinie 241
Alternatywne kryptowaluty oparte na Bitcoinie 243

Instalowanie Bitcoina ... 244
Typy klientów i narzędzi ... 244
Przygotowywanie węzła Bitcoina ... 245
Uruchamianie węzła w sieci testnet ... 246
Uruchamianie węzła w sieci regtest ... 248
Dalsze eksperymenty z interfejsem bitcoin-cli ... 249
Obsługa Bitcoina za pomocą narzędzia działającego w wierszu poleceń ... 251
Stosowanie interfejsu JSON-RPC ... 252
Korzystanie z interfejsu HTTP REST ... 253
Programowanie w świecie Bitcoina ... 254
Podsumowanie ... 255

ROZDZIAŁ 8
Inteligentne kontrakty ... 256

Wprowadzenie do inteligentnych kontraktów ... 256
Definicje ... 257
Cechy ... 258
Praktyczne zastosowania ... 259
Kontrakty ricardiańskie ... 261
Szablony inteligentnych kontraktów ... 264
Wyrocznie ... 266
Dowody generowane z wykorzystaniem oprogramowania i sieci ... 268
Dowody z wykorzystaniem sprzętu ... 269
Typy wyroczni dla łańcuchów bloków ... 272
Usługi wyroczni dla łańcuchów bloków ... 276
Umieszczanie inteligentnych kontraktów w łańcuchu bloków ... 276
The DAO ... 278
Postępy w technologii inteligentnych kontraktów ... 279
Solana Sealevel ... 279
Digital Asset Modeling Language ... 280
Podsumowanie ... 282

ROZDZIAŁ 9
Architektura Ethereum ... 283

Wprowadzenie do Ethereum ... 283
Kryptowaluta ... 285
Klucze i adresy ... 286
Konta ... 290
Transakcje i komunikaty ... 292
Drzewa MPT ... 292

Komponenty transakcji ... 294
Format RLP ... 298
Paliwo ... 299
Typy transakcji ... 301
Komunikaty ... 303
Sprawdzanie poprawności i wykonywanie transakcji ... 304
Stan i składowanie w łańcuchu bloków Ethereum ... 305
Maszyna EVM w Ethereum ... 308
Środowisko uruchomieniowe ... 311
Stan maszyny ... 312
Bloki i łańcuchy bloków ... 312
Blok początkowy ... 314
Sprawdzanie poprawności, finalizowanie i przetwarzanie bloków ... 315
Mechanizm zmiany trudności wydobywania bloku ... 316
Węzły i górnicy ... 317
Mechanizm osiągania konsensusu ... 318
Forki w łańcuchu bloków ... 319
Sieć Ethereum ... 320
Mainnet ... 320
Sieci testowe ... 320
Sieci prywatne ... 321
Prekompilowane inteligentne kontrakty ... 325
Języki programowania ... 327
Portfele i oprogramowanie klienckie ... 328
Portfele ... 328
Geth ... 328
Uproszczone klienty ... 329
Protokoły pomocnicze ... 329
Whisper ... 329
Swarm ... 330
Podsumowanie ... 331

ROZDZIAŁ 10
Ethereum w praktyce ... 332

Płatności w Ethereum ... 332
Innowacje w Ethereum ... 334
Bomba trudności ... 334
EIP 1559 ... 335
The Merge i nadchodzące aktualizacje ... 337
Programowanie z użyciem klienta Geth ... 338
Instalowanie i konfigurowanie klienta Geth ... 338

Tworzenie nowego konta za pomocą klienta Geth 339
Kierowanie zapytań do łańcucha bloków za pomocą klienta Geth 340
Konfigurowanie środowiska programistycznego 343
Łączenie się z sieciami testowymi 344
Tworzenie sieci prywatnej 345
Wprowadzenie do środowiska IDE Remix 356
Komunikowanie się z łańcuchem bloków Ethereum za pomocą narzędzia MetaMask 359
Instalowanie narzędzia MetaMask 359
Tworzenie i zasilanie konta za pomocą narzędzia MetaMask 360
Stosowanie narzędzia MetaMask i środowiska IDE Remix do umieszczania inteligentnego kontraktu w łańcuchu 362
Podsumowanie 376

ROZDZIAŁ 11
Narzędzia, języki i platformy dla programistów Ethereum 377

Języki 378
Kompilator języka Solidity 378
Instalowanie kompilatora solc 378
Eksperymentowanie z kompilatorem solc 379
Narzędzia, biblioteki i platformy 381
Node.js 381
Ganache 381
Truffle 384
Drizzle 385
Inne narzędzia 385
Pisanie i dodawanie kontraktów 386
Pisanie inteligentnych kontraktów 386
Testowanie inteligentnych kontraktów 387
Dodawanie inteligentnych kontraktów 387
Język Solidity 387
Funkcje 389
Zmienne 393
Typy danych 394
Struktury sterujące 398
Zdarzenia 399
Dziedziczenie 400
Biblioteki 401
Obsługa błędów 401
Podsumowanie 402

ROZDZIAŁ 12
Programowanie w Ethereum z użyciem biblioteki Web3 403
Interakcja z kontraktami za pomocą interfejsu Web3 i klienta Geth 403
Dodawanie kontraktów 404
Stosowanie kompilatora solc do generowania interfejsu ABI i kodu 408
Kierowanie zapytań do kontraktów za pomocą klienta Geth 409
Interakcje z klientem Geth za pomocą żądań POST 412
Interakcja z kontraktami za pomocą frontendów 413
Instalowanie javascriptowej biblioteki web3.js 413
Tworzenie obiektu web3 414
Tworzenie javascriptowego pliku app.js 415
Tworzenie strony internetowej używanej jako frontend 418
Wywoływanie funkcji kontraktu 419
Tworzenie strony internetowej używanej jako frontend 419
Dodawanie kontraktów i komunikowanie się z nimi za pomocą platformy Truffle 422
Instalowanie i inicjalizowanie platformy Truffle 422
Kompilowanie, testowanie i przenoszenie kontraktów za pomocą platformy Truffle 423
Interakcja z kontraktem 427
Używanie platformy Truffle do testowania i dodawania inteligentnych kontraktów 429
Umieszczanie danych w zdecentralizowanym magazynie w systemie IPFS 433
Podsumowanie 435

ROZDZIAŁ 13
The Merge i późniejsze aktualizacje 436
Wprowadzenie 436
Ethereum po aktualizacji The Merge 437
Beacon Chain 439
Interfejs P2P (sieci) 452
The Merge 452
Sharding 463
Plan przyszłych prac nad Ethereum 472
Podsumowanie 473

ROZDZIAŁ 14
Hyperledger 474
Projekty w ramach programu Hyperledger 475
Rejestry rozproszone 475
Biblioteki 477

Narzędzia ... 479
Projekty specyficzne dla dziedziny ... 479
Architektura wzorcowa w programie Hyperledger ... 480
Cele projektowe związane z platformą Hyperledger ... 482
Hyperledger Fabric ... 484
Najważniejsze zagadnienia ... 484
Komponenty ... 489
Aplikacje ... 492
Mechanizm osiągania konsensusu ... 494
Cykl życia transakcji ... 495
Fabric 2.0 ... 497
Nowy sposób zarządzania cyklem życia kontraktów chaincode ... 497
Nowe wzorce stosowania kontraktów chaincode ... 499
Podsumowanie ... 500

ROZDZIAŁ 15
Tokenizacja ... 501

Tokenizacja w łańcuchu bloków ... 502
Zalety tokenizacji ... 502
Wady tokenizacji ... 504
Rodzaje tokenów ... 505
Tokeny wymienialne ... 505
Tokeny niewymienialne ... 506
Stabilne tokeny ... 507
Tokeny inwestycyjne ... 508
Proces tokenizacji ... 508
Oferty tokenów ... 509
Pierwsza oferta tokenów ... 509
Oferty STO ... 510
Oferty IEO ... 510
Oferty ETO ... 510
Oferty DAICO ... 511
Inne oferty tokenów ... 511
Standardy związane z tokenami ... 512
ERC-20 ... 513
ERC-223 ... 513
ERC-777 ... 513
ERC-721 ... 514
ERC-884 ... 514
ERC-1400 ... 514
ERC-1404 ... 514

ERC-1155 ... 515
ERC-4626 ... 515
Tworzenie tokenów ERC-20 ... 516
Tworzenie kontraktu w języku Solidity ... 517
Dodawanie kontraktu za pomocą maszyny wirtualnej JavaScriptu w środowisku Remix ... 521
Dodawanie tokenów w narzędziu MetaMask ... 525
Nowe koncepcje ... 527
Tokenomia (ekonomia tokenów) ... 527
Inżynieria tokenów ... 528
Taksonomia tokenów ... 528
Podsumowanie ... 528

ROZDZIAŁ 16
Korporacyjne łańcuchy bloków ... 529

Rozwiązania korporacyjne a łańcuch bloków ... 530
Czynniki wpływające na powodzenie ... 531
Czynniki ograniczające ... 532
Wymagania ... 534
Prywatność ... 534
Wydajność ... 535
Zarządzanie dostępem ... 536
Dodatkowe wymagania ... 536
Łańcuchy korporacyjne a łańcuchy publiczne ... 539
Architektura korporacyjnego łańcucha bloków ... 540
Projektowanie rozwiązań opartych na korporacyjnym łańcuchu bloków ... 544
TOGAF ... 544
Metoda rozwoju architektury ... 545
Łańcuch bloków w chmurze ... 548
Obecnie dostępne korporacyjne łańcuchy bloków ... 549
Wyzwania związane z korporacyjnymi łańcuchami bloków ... 552
Interoperacyjność ... 552
Brak standaryzacji ... 552
Zgodność z przepisami ... 553
Wyzwania biznesowe ... 553
Łańcuch VMBC ... 554
Komponenty ... 554
Protokół osiągania konsensusu ... 555
Architektura ... 555
VMBC for Ethereum ... 557

Quorum ... 558
Architektura ... 558
Kryptografia ... 560
Prywatność ... 561
Kontrola dostępu oparta na uprawnieniach ... 564
Wydajność ... 566
Wymienne algorytmy osiągania konsensusu ... 567
Konfigurowanie sieci Quorum z algorytmem IBFT ... 567
Instalowanie i uruchamianie narzędzia Quorum Wizard ... 568
Przeprowadzanie transakcji prywatnej ... 570
Dołączanie klienta Geth do węzłów ... 571
Wyświetlanie transakcji w narzędziu Cakeshop ... 573
Dalsze sprawdzanie w kliencie Geth ... 574
Inne projekty na platformie Quorum ... 577
Wtyczka dla środowiska Remix ... 577
Architektura oparta na wtyczkach ... 577
Podsumowanie ... 578

ROZDZIAŁ 17
Skalowalność ... 579

Czym jest skalowalność? ... 579
Trylemat łańcuchów bloków ... 580
Metody zwiększania skalowalności ... 583
Rollupy ... 595
Podsumowanie ... 619

ROZDZIAŁ 18
Prywatność ... 620

Prywatność ... 620
Anonimowość ... 621
Poufność ... 621
Techniki osiągania prywatności ... 622
Warstwa 0. ... 623
Prywatność oparta na dowodach ZK ... 633
Przykład ... 650
Podsumowanie ... 657

ROZDZIAŁ 19
Bezpieczeństwo w łańcuchach bloków ... 658

Bezpieczeństwo ... 658
Warstwy i ataki w łańcuchach bloków ... 660
Warstwa sprzętu ... 661

Warstwa sieci ... 663
Warstwa łańcucha bloków ... 664
Warstwa aplikacji łańcucha bloków ... 668
Warstwa interfejsu ... 673
Ataki na łańcuchy bloków warstwy 2. ... 676
Warstwa kryptografii ... 677
Narzędzia i mechanizmy do analizy zabezpieczeń ... 681
Formalna weryfikacja ... 683
Bezpieczeństwo inteligentnych kontraktów ... 688
Solgraph ... 690
Modelowanie zagrożeń ... 691
Regulacje i zgodność z przepisami ... 693
Podsumowanie ... 694

ROZDZIAŁ 20
Zdecentralizowana tożsamość ... 695

Tożsamość ... 695
Tożsamość cyfrowa ... 696
Tożsamość w Ethereum ... 719
Tożsamość w świecie Web3, DeFi i metawersum ... 720
Projekty łańcuchów bloków specyficznych dla tożsamości suwerennej ... 723
Hyperledger Indy, Aries, Ursa i AnonCreds ... 723
Inne projekty ... 724
Inne inicjatywy ... 724
Wyzwania ... 725
Podsumowanie ... 725

ROZDZIAŁ 21
Zdecentralizowane finanse ... 726

Wprowadzenie ... 726
Rynki finansowe ... 728
Handel ... 728
Giełdy ... 729
Zastosowania łańcucha bloków w finansach ... 732
Ubezpieczenia ... 733
Rozliczenia potransakcyjne ... 733
Zapobieganie przestępstwom finansowym ... 734
Płatności ... 736
Zdecentralizowane finanse ... 737
Cechy rozwiązań z obszaru DeFi ... 739
Warstwy w DeFi ... 740

Podstawowe elementy ekosystemu DeFi ... 741
Usługi DeFi ... 743
Zalety DeFi ... 756
Uniswap ... 758
Wymiana tokenów ... 758
Pula płynności na giełdzie Uniswap ... 759
Podsumowanie ... 762

ROZDZIAŁ 22
Zastosowania i przyszłość łańcuchów bloków ... 763

Zastosowania ... 763
Internet rzeczy ... 764
Architektura internetu rzeczy ... 765
Warstwa obiektów fizycznych ... 766
Warstwa urządzeń ... 766
Warstwa sieci ... 766
Warstwa zarządzania ... 766
Warstwa aplikacji ... 767
Korzyści z łączenia internetu rzeczy z łańcuchem bloków ... 767
Implementowanie internetu rzeczy opartego na łańcuchach bloków w praktyce ... 770
Konfigurowanie Raspberry Pi ... 772
Konfigurowanie pierwszego węzła ... 774
Konfigurowanie węzła w Raspberry Pi ... 775
Instalowanie Node.js ... 776
Budowanie obwodu elektronicznego ... 777
Tworzenie i uruchamianie kontraktu w języku Solidity ... 778
Administracja publiczna ... 783
Kontrola graniczna ... 783
Wybory ... 785
Identyfikacja obywateli ... 786
Opieka zdrowotna ... 787
Media ... 788
Łańcuchy bloków a sztuczna inteligencja ... 789
Wybrane nowe trendy ... 791
Wybrane wyzwania ... 793
Podsumowanie ... 796

ROZDZIAŁ 23

Inne rozwiązania z obszaru łańcuchów bloków 798

Wprowadzenie 798
Kadena 798
Pact 801
EOS 803
Zasoby 804
Komponenty 804
Programowanie w łańcuchu bloków EOS 806
Tezos 806
Architektura 808
Sieć 808
Klient 808
Węzeł 808
Jednostka zatwierdzająca 809
„Piekarz” 809
Oskarżyciel 810
Konta 810
Tworzenie kontraktów 812
Portfele 813
Ripple 813
Transakcje 815
Interledger 817
Stellar 819
Protokół osiągania konsensusu w systemie Stellar 819
Rootstock 820
Solana 822
Dowód historii 823
Projekty łańcuchów bloków dla warstwy przechowywania 827
Storj 827
MaidSafe 828
Inne platformy 829
MultiChain 829
Tendermint 829
Podsumowanie 830

Rozdział 2

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 zdecentralizowane aplikacje i platformy do zapewniania decentralizacji.

Oto zagadnienia omawiane w tym rozdziale:

- wprowadzenie do decentralizacji,
- pełna decentralizacja ekosystemu,
- decentralizacja w praktyce,
- innowacje.

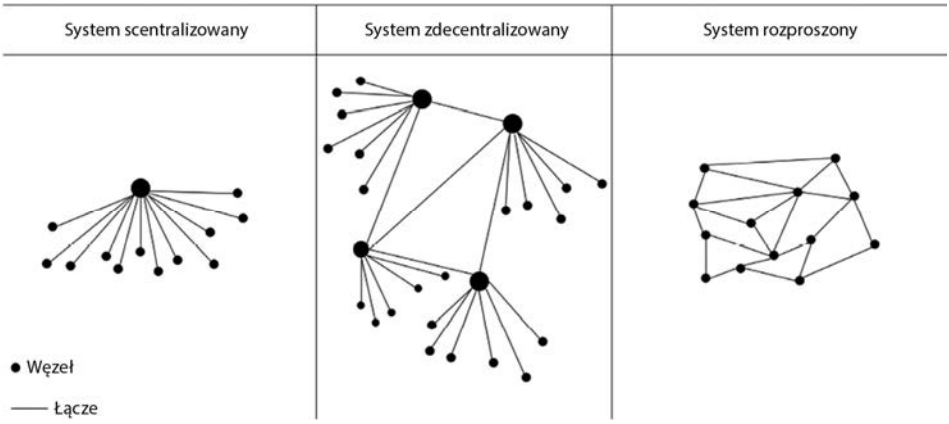
Wprowadzenie do decentralizacji

Decentralizacja jest jedną z najważniejszych korzyści 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 osiągania konsensusu. Ten model umożliwia każdemu współzawodnictwo o pozycję jednostki podejmującej decyzje. To współzawodnictwo odbywa się na podstawie mechanizmu osiągania konsensusu, który omawiam w rozdziale 5. „Algorytmy osiągania konsensusu”.

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.

Infrastruktura teleinformatyczna tradycyjnie jest oparta 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, rozproszone i zdecentralizowane. 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, Yahoo!, App Store firmy Apple itd., posługują się tym tradycyjnym modelem dostarczania usług.

W **systemie rozproszonym** dane i obliczenia są rozdzielane między wiele węzłów sieci, którą użytkownicy postrzegają jako jeden spójny system.

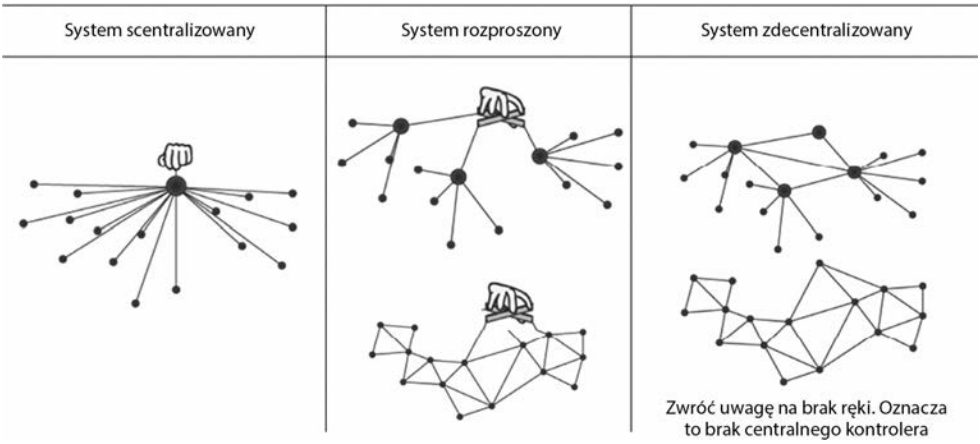
Uwaga

Czasem to pojęcie jest mylone z **przetwarzaniem równoległym.** Odmiany obu tych modeli są używane do osiągania odporności na błędy i zwiększania szybkości. 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. W systemach równoległych nadal występuje centralna jednostka zarządzająca, która kontroluje wszystkie węzły i zarządza przetwarzaniem. To oznacza, że system jest z natury scentralizowany.

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 jest osiąganie **konsensusu w środowisku zdecentralizowanym**. Ten mechanizm pojawił się wraz z Bitcoinem i umożliwia użytkownikom uzgadnianie rzeczy za pomocą algorytmu osiągania konsensusu, bez konieczności udziału centralnej, zaufanej trzeciej strony, pośrednika lub dostawcy usług.

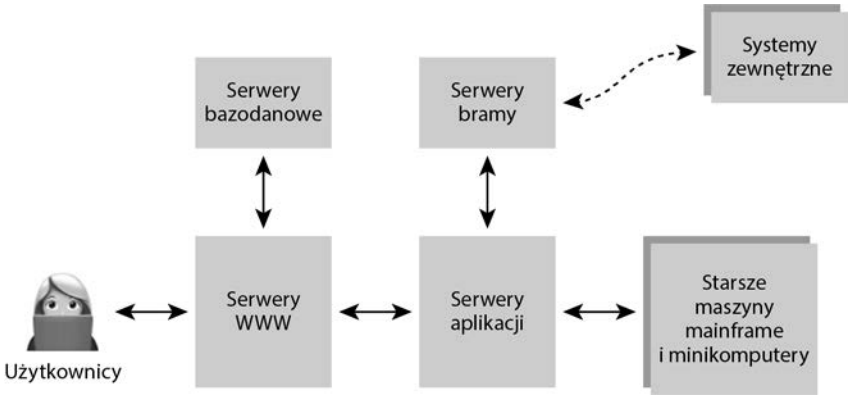
Można też spojrzeć na różne typy sieci z innej perspektywy, tak jak na rysunku 2.2, gdzie jednostka kontrolująca daną sieć jest przedstawiona w formie symbolicznej ręki.



Rysunek 2.2. Różne typy sieci i systemów ilustrujące decentralizację we współczesnym ujęciu

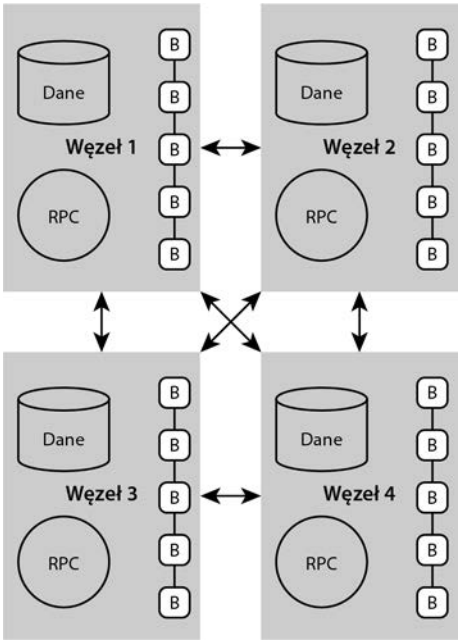
Na tym rysunku widoczny jest tradycyjny scentralizowany model z centralnym kontrolerem. Tak działa standardowy model klient – serwer. Pośrodku pokazane są systemy rozproszone, w których występuje centralny kontroler, ale także wiele rozproszonych węzłów. Po prawej stronie zwróć uwagę na to, że nie jest widoczna ręka (kontroler) kontrolująca sieci. Jest to główna różnica między sieciami zdecentralizowanymi a rozproszonymi. W ujęciu topologicznym system zdecentralizowany może wyglądać jak system rozproszony, jednak nie ma centralnej jednostki kontrolującej sieć.

Różnice między systemami rozproszonymi i zdecentralizowanymi są pokazane w praktycznym ujęciu na rysunku 2.3.



Rysunek 2.3. Tradycyjny system rozproszony obejmuje wiele serwerów pełniących różne funkcje

Na rysunku 2.4 pokazany jest oparty na łańcuchu bloków system zdecentralizowany, w którym identyczna replika aplikacji i danych jest przechowywana w każdym węźle w całej sieci.



Rysunek 2.4. System zdecentralizowany oparty na łańcuchu bloków. Zwróć uwagę na bezpośrednie połączenia w sieci P2P i identyczne repliki bloków (danych)

W tabeli poniżej pokazane jest porównanie systemów scentralizowanych i zdecentralizowanych.

Aspekt	Scentralizowane	Zdecentralizowane
Własność	Dostawca usług	Wszyscy użytkownicy
Architektura	Klient – serwer	Rozproszona, różne topologie
Bezpieczeństwo	Podstawowe	Bardziej bezpieczne
Wysoka dostępność	Nie	Tak
Odporność na awarie	Ograniczona, jeden punkt podatności na awarie	Wysoce odporne, ponieważ usługa jest replikowana
Odporność na zmowę	Niska, ponieważ system jest kontrolowany przez grupę lub nawet pojedynczą jednostkę	Wysoka, ponieważ algorytmy osiągania konsensusu zapewniają ochronę przed napastnikami
Architektura aplikacji	Jedna aplikacja	Aplikacja replikowana we wszystkich węzłach sieci
Zaufanie	Klienci muszą ufać dostawcy usług, czyli zaufanej trzeciej stronie	Wzajemne zaufanie nie jest wymagane
Koszty dla klienta	Wysokie	Niskie

W tej tabeli uwzględniłem tylko wybrane ważne aspekty, dlatego ta lista nie jest kompletna, ale powinna umożliwiać dobre porównanie obu modeli.

Warto zauważyć, że także w systemach scentralizowanych można zwiększyć odporność na awarie (możliwość kontynuowania pracy systemu nawet po awarii niektórych komponentów) dzięki replikacji danych. Jednak w systemie zdecentralizowanym odporność na awarie jest wyższa, ponieważ po pierwsze system jest rozproszony, a po drugie system jest zdecentralizowany, dlatego żadna jednostka nie może samodzielnie zmanipulować go i zyskać nieproporcjonalnej przewagi. W prostej architekturze klient – serwer, gdzie usługi są zapewniane przez tylko jeden centralny serwer lub serwer główny i rezerwowy, odporność na awarie jest zdecydowanie niższa niż w zdecentralizowanym rozproszonym systemie łańcucha bloków, ponieważ łańcuchy bloków są zwykle replikowane w setkach lub tysiącach jednostek w różnych lokalizacjach geograficznych z całego świata.

Teraz omówię metody używane do uzyskania decentralizacji.

Metody decentralizacji

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

Wskazówka

Banki centralne i organy kształtujące politykę pieniężną stwierdziły, że mogą wykorzystać łańcuch bloków do emisji regulowanej waluty cyfrowej, **waluty cyfrowej banku centralnego**, która pomoże uprościć realizację polityki monetarnej i fiskalnej. Choć jest to ważna zmiana, która może doprowadzić do powstania bezpieczniejszego, wydajniejszego i uczciwszego ekosystemu finansowego, można się spodziewać, że będzie to system scentralizowany, gdzie za regulacje i emisję waluty będzie odpowiadać bank centralny lub organy kształtujące politykę pieniężną danego kraju.

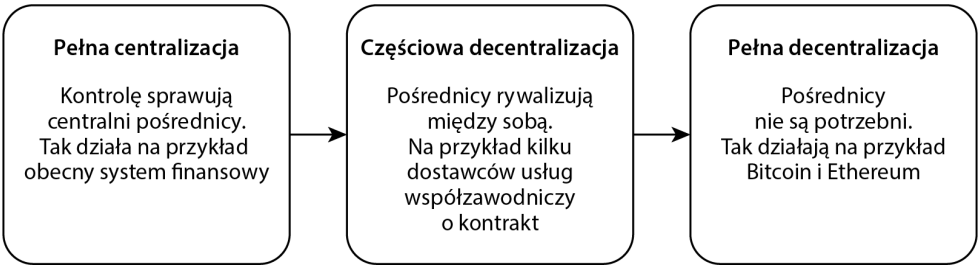
Jednak ten model może być stosowany nie tylko w finansach, ale też w wielu innych branżach. W sektorze opieki zdrowotnej zamiast polegać na zaufanej trzeciej stronie (na przykład szpitalnej kartotece) pacjenci mogą w pełni kontrolować swoją tożsamość i swoje dane oraz udostępniać je bezpośrednio tylko tym jednostkom, do których mają zaufanie. Łańcuch bloków jako uniwersalne rozwiązanie może pełnić funkcję zdecentralizowanego systemu zarządzania dokumentacją medyczną, w której dokumentację można bezpiecznie przekazywać bezpośrednio między różnymi jednostkami (szpitalami, firmami farmaceutycznymi, pacjentami) na poziomie globalnym bez żadnych organów centralnych. Choć zapewnienie współdziałania różnych standardów rejestrowania i kategoryzowania danych medycznych nie jest łatwe, łańcuch bloków może przynajmniej zapewnić platformę wymiany danych między różnymi jednostkami służby zdrowia.

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 to ich usługi zostały wybrane.

Ilościowe ujęcie decentralizacji

Na rysunku 2.5 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 2.5. Poziomy decentralizacji

Można też przedstawić inną skalę decentralizacji, rozciągającą się od **minimalnej osiągalnej decentralizacji** do **maksymalnej akceptowalnej decentralizacji**. Ponadto można skupić się na znalezieniu **optymalnego punktu decentralizacji** na skali dla określonego scenariusza, czyli maksymalnego poziomu decentralizacji przy minimalnej centralizacji, jaka będzie optymalna w konkretnym zastosowaniu przy danych warunkach.

Można zadać sobie pytanie o to, jak mierzyć poziom decentralizacji. Służy do tego „współczynnik Nakamoto". Ta miara jest obliczana na podstawie kilku czynników. Określa ona liczbę jednostek, nad którymi trzeba przejąć kontrolę, by móc zmanipulować sieć łańcucha bloków. Im wyższa jest wartość współczynnika Nakamoto, tym bardziej sieć jest zdecentralizowana.

Uwaga

Wartości współczynnika Nakamoto znajdziesz na stronie *https://nakaflow.io/*.

Każdy zdecentralizowany system składa się z kilku zdecentralizowanych podsystemów. Jeśli któryś podsystem jest scentralizowany, cały system też jest uznawany za scentralizowany. Na przykład łańcuch bloków może składać się z kilku podsystemów, związanych z górnikami, klientami, programistami, giełdami, węzłami i własnością. Można stwierdzić, że z powodu monopolu kopalń Bitcoin może zostać uznany za scentralizowany łańcuch bloków. Najważniejszym aspektem wspomnianego współczynnika jest to, że najpierw należy wymienić podsystemy zdecentralizowanego systemu, następnie ustalić, ile jednostek musi zostać przejętych, by zyskać kontrolę nad poszczególnymi podsystemami, i ostatecznie użyć minimum spośród tych wartości, aby uzyskać ogólny faktyczny poziom decentralizacji systemu. Na przykład łańcuch bloków Ethereum można uznać za scentralizowany, ponieważ większość zmian jest przesyłanych przez niewielką grupę programistów, co oznacza centralizację na poziomie programistów.

Uwaga

Współczynnik Nakamoto został wprowadzony przez Balajiego S. Srinivasana i Lelanda Lee w artykule dostępnym na stronie *https://news.earn.com/quantifying-decentralization-e39db233c28e*.

Zalety 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, jak zagwarantować, że cyfrowe zasoby powiązane z takimi kluczami nie staną się bezwartościowe z powodu zaniedbań lub błędów w kodzie? Co się stanie, jeśli klucze prywatne zostaną utracone z winy użytkownika? Co zrobić, jeśli z powodu błędu w kodzie inteligentnego kontraktu zdecentralizowana aplikacja stanie się podatna 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 pytań:

Pytanie	Tak/Nie	Zalecane rozwiązanie
Czy potrzebna jest wysoka przepustowość obsługi danych?	Tak	Zastosuj tradycyjną bazę danych.
	Nie	Scentralizowana baza danych nadal może być przydatna, jeśli spełnione są inne warunki. Na przykład jeżeli użytkownicy sobie ufają, możliwe, że nie ma potrzeby stosowania łańcucha bloków. Jeśli jednak z jakichkolwiek powodów nie da się zapewnić zaufania, łańcuch bloków może okazać się pomocny.
Czy aktualizacje są kontrolowane przez centralną jednostkę?	Tak	Posłuż się tradycyjną bazą.
	Nie	Możesz zbadać, w jaki sposób może pomóc publiczny lub prywatny łańcuch bloków.
Czy użytkownicy sobie ufają?	Tak	Użyj tradycyjnej bazy.
	Nie	Zastosuj publiczny łańcuch bloków.
Czy użytkownicy są anonimowi?	Tak	Zastosuj publiczny łańcuch bloków.
	Nie	Wykorzystaj prywatny łańcuch bloków.

Pytanie	Tak/Nie	Zalecane rozwiązanie
Czy konsensus ma być utrzymywany w ramach konsorcjum?	Tak	Wykorzystaj prywatny łańcuch bloków.
	Nie	Zastosuj publiczny łańcuch bloków.
Czy wymagana jest ścisła niemodyfikowalność danych?	Tak	Użyj łańcucha bloków.
	Nie	Zastosuj scentralizowaną, tradycyjną bazę danych.

Udzielenie odpowiedzi na wszystkie te pytania może pozwolić zrozumieć, czy łańcuch bloków jest potrzebny lub czy nadaje się do rozwiązania problemu. 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 wewnętrznie przez 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.

Teraz znasz już różne metody decentralizacji i wiesz, jak zdecydować, czy łańcuch bloków jest potrzebny w danym scenariuszu. Przyjrzyj się teraz procesowi decentralizacji, czyli temu, jak zdecentralizować istniejący system.

Ocena wymagań

Już przed łańcuchami bloków i Bitcoinem istniały inne systemy (np. systemy wymiany plików BitTorrent i Gnutella), które można uznać za — w pewnym stopniu — zdecentralizowane. Ale z powodu braku nagród poziom aktywności społeczności stopniowo spadał. 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. Jednak to Ethereum stał się lepszym wyborem z powodu swobody i możliwości zaprogramowania dowolnej logiki biznesowej w łańcuchu bloków za pomocą **inteligentnych kontraktów**. Ponadto także nowsze łańcuchy, na przykład Polkadot, Solana i Cardano, są używane przez wielu programistów jako platformy do uzyskiwania decentralizacji.

Współczynnik Nakamoto jest dla różnych łańcuchów inny i powinien być decydującym czynnikiem przy ocenie platform łańcuchów bloków w określonym scenariuszu. Choć najlepiej jest zapewnić sobie jak najwyższy poziom decentralizacji, w niektórych sytuacjach można z niej częściowo zrezygnować. Najwyższy współczynnik Nakamoto ma

Bitcoin, natomiast niektóre łańcuchy mają bardzo niską wartość tego współczynnika. Wybór platformy zależy od zastosowania i wymogów użytkowników. W niektórych scenariuszach częściowa rezygnacja z decentralizacji jest akceptowalna.

Arvind Narayanan i in. zaproponowali w książce *Bitcoin and Cryptocurrency Technologies* 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:

- *Co jest decentralizowane?* Może to być dowolny system, na przykład do zarządzania tożsamością lub handlu.
- *Jaki poziom decentralizacji jest wymagany?* Wymagana może być pełna lub częściowa decentralizacja.
- *Jaki łańcuch bloków jest używany?* Może to być Bitcoin, Ethereum lub dowolny inny łańcuch bloków odpowiedni w danym scenariuszu.
- *Jakie mechanizmy zabezpieczeń są stosowane?* Mogą to być na przykład mechanizmy oparte na atomowości, sprawiające, że transakcja albo jest wykonywana w całości, albo wcale. To deterministyczne podejście gwarantuje integralność systemu. W innych mechanizmach wykorzystuje się na przykład reputację, dzięki czemu w systemie mogą działać jednostki o różnym poziomie zaufania.

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:

- *Co jest decentralizowane?* System transferu pieniędzy.
- *Jaki poziom decentralizacji jest wymagany?* Eliminowanie pośrednictwa.
- *Jaki łańcuch bloków jest używany?* Bitcoin.
- *Jakie mechanizmy zabezpieczeń są stosowane?* 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.

Aby uzyskać pełną decentralizację, także środowisko powiązane z łańcuchem bloków musi być zdecentralizowane. Przyjrzyj się teraz kompletnemu ekosystemowi potrzebnemu do decentralizacji.

Decentralizacja całego ekosystemu

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 majątek, 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*) posiada obie te cechy. Ma on zastąpić protokół HTTP i doprowadzić do powstania zdecentralizowanej sieci WWW. System IPFS składuje dane w tablicach DHT Kademlia, a wyszukiwanie obsługuje za pomocą **acyklicznych grafów skiero**wanych skrótów (ang. *Merkle directed acyclic graph*). Tablice DHT i acykliczne grafy skierowane zostaną szczegółowo opisane w rozdziale 4. „Kryptografia asymetryczna” i rozdziale 17. „Skalowalność”.

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. W takich sieciach używane są technologie bezprzewodowe, na przykład **BLE** (ang. *Bluetooth Low Energy*), do tworzenia sieci komunikacyjnych niewymagających połączenia z internetem. Choć takie sieci w porównaniu z internetem mają ograniczone możliwości, zapewniają zdecentralizowaną alternatywę i pozwalają znajdującym się stosunkowo blisko siebie węzłom komunikować się bezpośrednio bez konieczności korzystania z internetu lub centralnej jednostki takiej jak dostawca usług internetowych. Wyobraź sobie teraz sieć, która umożliwia użytkownikom kontrolowanie komunikacji — w żadnej sytuacji nikt nie może jej wyłączyć. Sieć tego typu daje dużo korzyści na przykład w obliczu klęsk żywiołowych lub na obszarach objętych działaniami wojennymi. Innym zastosowaniem jest organizowanie protestów przeciwko reżimom, które blokują dostęp do internetu. Przykładową aplikacją do komunikowania się w trybie offline jest **bridgefy**. Używanie takich sieci może być następnym krokiem w kierunku decentralizacji sieci komunikacyjnych w ekosystemie łańcucha bloków.

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

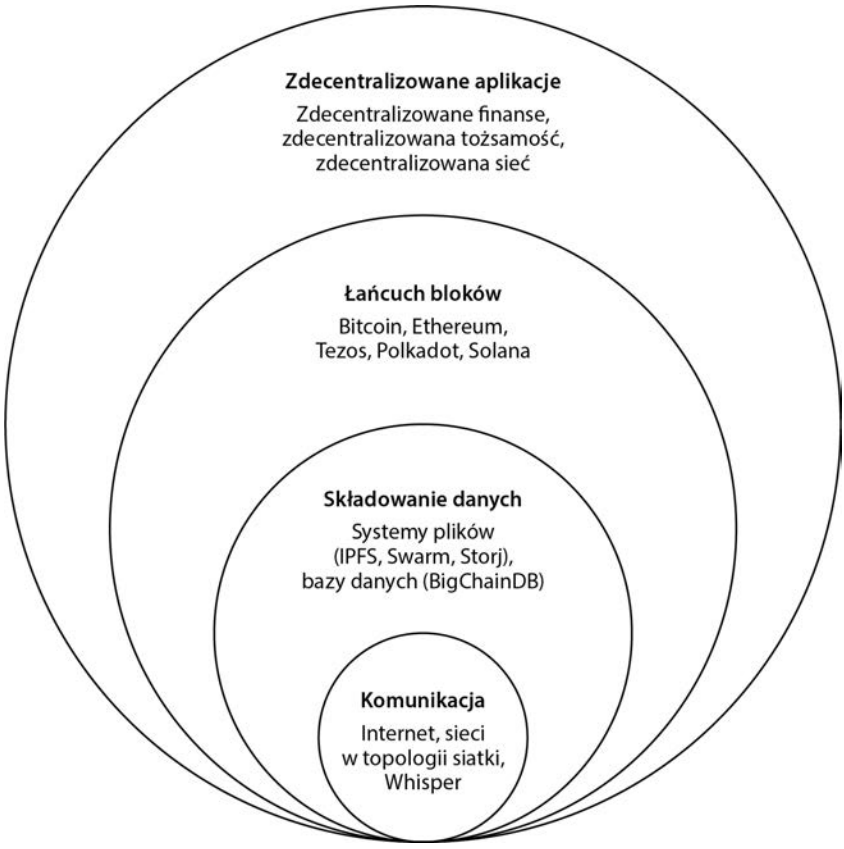
rozwiązania, np. Gmail i Outlook, które oferują dodatkowe usługi dla użytkowników końcowych. Dlatego naturalnym wyborem jest korzystanie z jednej z takich scentralizowanych usług, ponieważ są one wygodniejsze, bezpieczniejsze i przede wszystkim bezpłatne. Jest to jeden z przykładów ilustrujących, że 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.

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 2.6 pokazany jest zdecentralizowany ekosystem.



Rysunek 2.6. 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.

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 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.

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.

Wraz z powstaniem zdecentralizowanego modelu w mediach i pracach naukowych zaczęto stosować różne pojęcia i modne słowa. Omawiam je w następnym podrozdziale.

Decentralizacja w praktyce

Poniżej przedstawiam zagadnienia, które warto omówić w kontekście decentralizacji. Przedstawiona tu terminologia jest często stosowana w literaturze dotyczącej decentralizacji i jej zastosowań.

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 8. „Inteligentne kontrakty".

Autonomiczne agenty

Autonomiczny agent to oprogramowanie (oparte na sztucznej inteligencji lub zaprogramowane w tradycyjny sposób) działające na rzecz właściciela, aby osiągnąć pożądane cele. Autonomiczny agent wymaga tylko niewielkiego lub nawet zerowego udziału właściciela.

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 zapewniania 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.

To zagadnienie jest związane z kwestią algokracji, czyli formy zarządzania i systemu społecznego, w której algorytmy komputerowe zarządzają usługami publicznymi (systemem prawnym, regulacjami, nadzorem, ekonomią, polityką i podejmowaniem decyzji publicznych), kontrolują je i automatyzują. Łańcuch bloków i zdecentralizowane aplikacje umożliwiają — zwłaszcza po połączeniu ich ze sztuczną inteligencją — wprowadzenie algokracji. Początkowo uważano, że sztuczna inteligencja (a nawet tradycyjne oprogramowanie) umożliwi zarządzanie algorytmiczne. Obecnie łańcuch bloków w połączeniu ze sztuczną inteligencją zapewniają bardziej eleganckie rozwiązanie.

Trzeba jednak zauważyć, że z algokracją związane są zarówno możliwe korzyści, jak i zagrożenia. Wzrost zależności od algorytmów w zarządzaniu można postrzegać jako zagrożenie dla aktywnego udziału ludzi i podejmowania przez nich decyzji w rzeczywistym życiu. Jest to prawdą w tradycyjnej algokracji, bez użycia łańcucha bloków. Jednak gdy dodać łańcuch bloków, sytuacja staje się lepsza. W łańcuchu bloków, dzięki zdecentralizowanemu modelowi i zarządzaniu przez społeczność, algorytmy zarządzające także podlegają zatwierdzaniu i kontroli ze strony społeczności (społeczeństwa) operującej danym łańcuchem. Tak więc łańcuch bloków można potraktować jako rozwiązanie problemu możliwej utraty kontroli nad procesem podejmowania decyzji. Tę odmianę algokracji można nazwać „blokokracją", czyli zarządzaniem przez łańcuch bloków. W tej wizji w łańcuchu bloków działają inteligentne kontrakty ze sztuczną inteligencją (algorytmy) odpowiedzialne za zarządzanie.

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.

Na ogólnym poziomie zdecentralizowane aplikacje to programy, które mogą działać za pomocą jednej z wymienionych poniżej metod. Zdecentralizowane aplikacje można zaliczyć do typu I, typu II i typu III.

- **Aplikacje typu I**. Takie aplikacje działają we własnym łańcuchu bloków. Są to na przykład zdecentralizowane aplikacje oparte na inteligentnych kontraktach działające w Ethereum. W razie konieczności korzystają one z natywnych tokenów, na przykład z ETH w łańcuchu bloków Ethereum. **Ethlance** jest zdecentralizowaną aplikacją, która korzysta z ETH do łączenia pracowników z pracodawcami.

  Uwaga

  Więcej informacji na temat aplikacji Ethlance znajdziesz na stronie *https://ethlance.com*.

- **Aplikacje typu II**. Te aplikacje używają istniejącego łańcucha bloków. Oznacza to, że korzystają z łańcucha bloków typu I oraz niestandardowych protokołów i tokenów. Na przykład zdecentralizowana aplikacja do obsługi tokenów oparta na inteligentnych kontraktach może działać w łańcuchu bloków Ethereum. Przykładem może być aplikacja **DAI**, która korzysta z łańcucha bloków Ethereum, ale ma własną stabilną kryptowalutę oraz mechanizmy jej dystrybucji i kontrolowania. Następnym przykładem jest aplikacja **Golem** z własnym tokenem, GNT, i platformą transakcyjną opartą na łańcuchu bloków Ethereum. Golem udostępnia **zdecentralizowany rynek** mocy obliczeniowej, na którym użytkownicy udostępniają moc obliczeniową w sieci P2P. Przykładową aplikacją zdecentralizowaną typu II jest sieć OMNI. Jest to korzystająca z łańcucha bloków Bitcoina warstwa oprogramowania umożliwiająca handel niestandardowymi aktywami cyfrowymi i walutami cyfrowymi.

  Uwaga

  Więcej informacji na temat sieci OMNI znajdziesz na stronie *https://www.omnilayer.org*, sieć Golem jest opisana na stronie *https://golem.network*, a aplikację DAI poznasz na stronie *https://makerdao.com/en/*.

- **Aplikacje typu III**. Te aplikacje korzystają z protokołów aplikacji zdecentralizowanych typu II. Na przykład sieć SAFE używa protokołów sieci OMNI.

  Uwaga

  Więcej informacji na temat sieci SAFE znajdziesz na stronie *https://safenetwork.tech*.

Innym przykładem pomagającym zrozumieć różnicę między poszczególnymi typami aplikacji zdecentralizowanych jest token USDT (Tether). W pierwotnej wersji w USDT jest używana warstwa sieci OMNI (zdecentralizowana aplikacja typu II) opartej na sieci Bitcoina. USDT jest dostępny także w sieci Ethereum, gdzie używane są tokeny ERC-20. Ten przykład pokazuje, że USDT można uznać za zdecentralizowaną aplikację typu III, w której jest używany protokół z sieci OMNI (zdecentralizowanej aplikacji typu II), która sama jest oparta na Bitcoinie (zdecentralizowanej aplikacji typu I). Również w Ethereum USDT można uznać za zdecentralizowaną aplikację typu III, ponieważ używa ono łańcucha bloków Ethereum (zdecentralizowanej aplikacji typu I) i standardu ERC-20, który opracowano na potrzeby działania w Ethereum.

Uwaga

Więcej informacji na temat tokena Tether znajdziesz na stronie *https://tether.to*.

W kilku ostatnich latach pojęcie „zdecentralizowana aplikacja" coraz częściej jest używane do opisywania dowolnej kompletnej zdecentralizowanej aplikacji z łańcucha bloków, obejmującej interfejs użytkownika (zwykle internetowy), inteligentne kontrakty

i łańcuch bloków pełniący funkcję hosta. Jednoznaczne rozróżnienie między poszczególnymi typami zdecentralizowanych aplikacji nie jest obecnie powszechnie uwzględniane, ale istnieje. Często nie podaje się typu i używa się określenia „zdecentralizowana aplikacja" dla wszystkich rozwiązań tego typu.

Obecnie istnieją tysiące różnych zdecentralizowanych aplikacji działających na rozmaitych platformach (w łańcuchach bloków). Dostępne są różne rodzaje zdecentralizowanych aplikacji z obszarów mediów, serwisów społecznościowych, finansów, gier, ubezpieczeń czy opieki zdrowotnej. Istnieją też różne zdecentralizowane platformy (łańcuchy bloków), w tym Ethereum, Solana, Avalanche, Polkadot i EOS. Na stronie *https://thedapplist.com* znajdziesz wybrane statystyki dotyczące zdecentralizowanych aplikacji.

Wymogi stawiane zdecentralizowanym aplikacjom

Aby aplikacja została uznana za zdecentralizowaną, musi spełniać wymienione niżej kryteria:

- **Decentralizacja**. Taka aplikacja powinna być w pełni zdecentralizowana. Żadna jednostka nie powinna kontrolować jej działania. Wszystkie zmiany w aplikacji muszą wynikać z konsensusu osiąganego na podstawie informacji zwrotnych ze społeczności.
- **Otwarty dostęp do kodu źródłowego**. Aplikacja musi zapewniać otwarty dostęp do kodu źródłowego, aby umożliwiać publiczną analizę i zapewniać przejrzystość.
- **Kryptograficzne zabezpieczenia**. Zmiany stanu i dane aplikacji muszą być kryptograficznie zabezpieczone oraz składowane w łańcuchu bloków, aby uniknąć scentralizowanych punktów podatności na awarie. Warto zauważyć, że dane nie muszą być szyfrowane, aby zapewnić poufność, ale należy je chronić przed manipulacjami ze strony nieuprawnionych osób. Trzeba zapewnić integralność danych, uwierzytelnianie i niezaprzeczalność.
- **Dostępność nagród**. Aplikacja musi używać kryptograficznych tokenów, aby zapewnić dostęp i nagrody jednostkom, które robią coś wartościowego na rzecz aplikacji (np. górnikom w Bitcoinie). Z tego wymogu można zrezygnować w łańcuchach należących do konsorcjów, gdzie tokeny mogą być używane do przekazywania wartości, ale nie muszą być kryptowalutą.
- **Dowód wartości**. Tokeny (jeśli są używane) muszą być generowane przez zdecentralizowane aplikacje z wykorzystaniem metod osiągania konsensusu i standardowego algorytmu kryptograficznego. Wygenerowane tokeny stanowią dowód wartości przekazywanej kontrybutorom (np. górnikom).

Zdecentralizowane aplikacje udostępniają obecnie różne usługi, w tym związane z finansami, grami, mediami społecznościowymi i zarządzaniem łańcuchami dostaw.

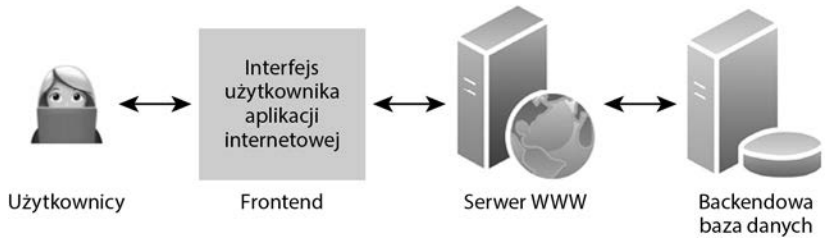
Operacje w zdecentralizowanych aplikacjach

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

Projekt zdecentralizowanej aplikacji

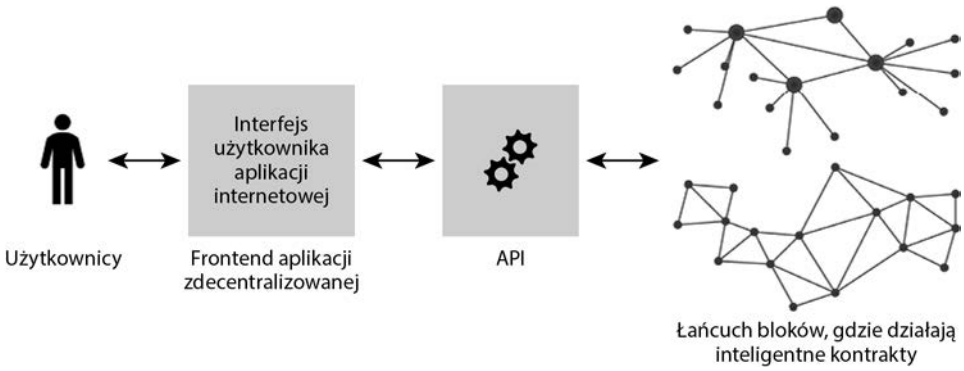
Zdecentralizowana aplikacja to program działający w zdecentralizowanej sieci, na przykład rozproszony rejestr. Zdecentralizowane aplikacje są od niedawna bardzo popularne dzięki powstaniu różnych platform zdecentralizowanych, takich jak Ethereum, Solana, EOS czy Tezos.

Tradycyjne aplikacje mają interfejs użytkownika i zwykle serwer WWW lub serwer aplikacji oraz backendową bazę danych. Tak wygląda typowa architektura klient – serwer zilustrowana na rysunku 2.7.



Rysunek 2.7. Architektura tradycyjnej aplikacji (ogólny model klient – serwer)

Z kolei w zdecentralizowanej aplikacji backendem jest łańcuch bloków, co ilustruje rysunek 2.8. Istotnym elementem, który odgrywa kluczową rolę w tworzeniu zdecentralizowanej aplikacji, jest działający w łańcuchu bloków inteligentny kontrakt z wbudowaną logiką biznesową.



Rysunek 2.8. Ogólna architektura zdecentralizowanej aplikacji

Warto zauważyć, że w zdecentralizowanej aplikacji frontendem może być zarówno „gruby" klient, jak i aplikacja mobilna i frontend internetowy (internetowy interfejs użytkownika). Zwykle jest używany frontend internetowy, często pisany za pomocą frameworka javascriptowego, na przykład Reacta lub Angulara.

W poniższej tabeli przedstawione są najważniejsze cechy poszczególnych typów zdecentralizowanych obiektów i różnice między nimi.

Obiekt	Autonomiczny?	Oprogramowanie?	Właściciel?	Kapitał?	Status prawny?	Koszt
DO	Nie	Nie	Tak	Tak	Tak	Wysoki
DAO	Tak	Tak	Nie	Tak	Rozpoczęto prace	Niski
DAC	Tak	Tak	Tak	Tak	Nieokreślony	Niski
DAS	Tak	Tak	Nie	Możliwe	Nieokreślony	Niski
DApp	Tak	Tak	Tak	Opcjonalny (tokeny)	Nieokreślony	Zależny od scenariusza

Można oczekiwać, że wszystkie te rozwiązania zostaną uregulowane i otrzymają w przyszłości status prawny, a przy tym pozostaną zdecentralizowane.

Uwaga

Zapoznaj się na przykład z artykułem na temat legalności autonomicznych organizacji zdecentralizowanych: *https://fedsoc.org/commentary/fedsoc-blog/the-legal-status-of-decentralized-autonomous-organizations-do-daos-require-new-business-structures-some-states-think-so*.

Platformą decentralizacji, na której można tworzyć i hostować zdecentralizowane aplikacje, może być dowolna sieć łańcucha bloków, na przykład Bitcoin, Ethereum, Solana, Hyperledger Fabric i Quorum.

Dzięki szybko rozwijanym innowacjom i ewolucji łańcucha bloków pojawiło się wiele nowatorskich trendów, które omawiam w następnym podrozdziale.

Innowacyjne trendy

Wraz z rozwojem łańcucha bloków pojawiły się różne pomysły, w których wykorzystano aspekt decentralizacji do udostępnienia bardziej zorientowanych na użytkownika i w pełni zdecentralizowanych usług. Niektóre z najważniejszych idei w tym obszarze to zdecentralizowany internet, zdecentralizowana tożsamość i zdecentralizowane finanse.

Zdecentralizowany internet

„Zdecentralizowany internet" to określenie opisujące wizję internetu, który nie jest kontrolowany przez żadne centralne władze ani przez grupę podmiotów. Pierwotnie internet rzeczywiście miał być zdecentralizowany, a rozwój otwartych protokołów (na przykład HTTP, SMTP czy DNS) sprawił, że każdy mógł z nich bezpłatnie korzystać. Szybko stały się one częścią internetu. Nadal tak jest, jednak wraz z pojawieniem się warstwy powyżej tych protokołów, **warstwy sieciowej**, wprowadzono infrastrukturę usługową, co musiało doprowadzić do przejęcia kontroli przez duże firmy nastawione na zysk. Dowodem na to jest rozwój korporacji Facebook, Google, Twitter czy Amazon, które oczywiście oferują użytkownikom wysokiej jakości usługi, ale kosztem powstania bardziej kontrolowanych, scentralizowanych i zamkniętych systemów.

Otwarte i bezpłatne protokoły, planowane i rozwijane jako rozwiązania zdecentralizowane, są obecnie zdominowane przez potężne organizacje komercyjne z całego świata. Prowadzi to do poważnych obaw o prywatność i ochronę danych. Modele biznesowe tego typu dobrze funkcjonują i są popularne z uwagi na wysoki poziom standaryzacji i świadczonych usług. Stanowią jednak zagrożenie dla prywatności i decentralizacji z powodu dominacji niewielkiej liczby korporacji w całym internecie.

Prognozuje się, że łańcuch bloków może zmienić tę sytuację i pozwolić na rozwój zdecentralizowanego internetu, nazwanego w skrócie Web 3 i zgodnego z pierwotnym przeznaczeniem internetu.

Teraz przyjrzymy się ewolucji internetu w ostatnich kilkudziesięciu latach. Główne osiągnięcia dzielę na trzy podstawowe etapy: Web 1, Web 2 i Web 3.

Web 1

Jest to oryginalna sieć World Wide Web opracowana w 1989 roku. Była to epoka, w której na serwerach przechowywano statyczne strony internetowe, a użytkownicy mogli zwykle tylko je wczytywać.

Web 2

Jest to epoka w większym stopniu nastawiona na usługi i aplikacje działające w internecie. Rozpoczęła się około 2003 roku. Sklepy internetowe, sieci społecznościowe, media społecznościowe, blogi, wymiana plików multimedialnych, mashupy i aplikacje internetowe to główne rozwiązania z tej epoki. Obecnie korzystamy właśnie z sieci Web 2, a choć internet stał się bogatszy i bardziej interaktywny, wszystkie usługi nadal są scentralizowane. Sieć Web 2 wygenerowała olbrzymią wartość ekonomiczną i zapewnia usługi niezbędne do codziennej działalności biznesowej, użytku osobistego, interakcji społecznych i prawie wszystkich dziedzin życia. Jednak obawy o prywatność, potrzeba zaufanej trzeciej strony i wycieki danych to rzeczywiste problemy, z którymi trzeba sobie poradzić. Typowe przykłady scentralizowanych usług z sieci Web 2 to Twitter, Facebook, Dokumenty Google czy serwisy poczty elektronicznej takie jak Gmail i Hotmail.

Web 3

Jest to wizja zdecentralizowanego internetu, który ma zrewolucjonizować obecny sposób korzystania z niego. Ta epoka ma być w pełni zorientowana na użytkownika i zdecentralizowana, bez kontroli ze strony jednego organu, dużej organizacji lub firmy internetowej. Oto niektóre przykłady zgodne z wizją Web 3:

- **Steemit**. Jest to platforma społecznościowa oparta na łańcuchu bloków Steem i kryptowalucie STEEM. Ta kryptowaluta jest przyznawana kontrybutorom za udostępniane przez nich treści. Im więcej głosów ktoś otrzyma, tym więcej tokenów zarobi. Więcej informacji znajdziesz na stronie *https://steemit.com*.
- **Status**. Jest to zdecentralizowana platforma komunikacyjna o wielu zastosowaniach, która umożliwia bezpieczną i prywatną komunikację. Więcej informacji znajdziesz na stronie *https://status.im*.
- **IPFS**. Jest to protokół typu P2P dla hipermediów i składowania danych, który umożliwia przechowywanie i udostępnianie danych w zdecentralizowany sposób w sieci P2P. Więcej informacji znajdziesz na stronie *https://ipfs.io*.

Inne przykłady to OpenSea (rynek do handlu tokenami NFT), UniSwap (zdecentralizowana giełda kryptowalut) i Angur (zdecentralizowana giełda). W Web 3 prawdopodobnie będą powszechnie używane trójwymiarowe wirtualne światy, tak zwane metawersy.

Inne szybko rozwijające się i ekscytujące aplikacje to usługi **DeFi** (ang. *decentralized identity and decentralized finance*), które omawiam w dalszych rozdziałach książki.

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 opisałem wybrane innowacyjne trendy związane ze zdecentralizowanymi aplikacjami, a także poruszyłem temat algokracji i blokokracji.

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.

Już dziś zacznij wdrażać technologię, o której inni dowiedzą się jutro!

Potencjał łańcucha bloków jest ogromny. W skrócie: blockchain to publiczny rejestr, który staje się ogólnoświatową, zdecentralizowaną księgą służącą do rejestrowania, inwentaryzacji i organizowania transferów wszelkiego rodzaju aktywów. Łańcuchy bloków są podstawą kryptowalut, znajdują też zastosowanie w wielu innych sektorach. Zrozumienie tej technologii jest warunkiem skorzystania ze wszystkich jej zalet.

To kolejne, uzupełnione i zaktualizowane wydanie praktycznego podręcznika dla każdego, kto chce korzystać z tej technologii w praktyce. Opisuje techniczne podstawy łańcuchów bloków, kryptografii i protokołów osiągania konsensusu, przedstawia też bardziej zaawansowane zagadnienia, jak tworzenie zdecentralizowanych aplikacji (DApp) przy użyciu inteligentnych kontraktów czy też łączenie internetu rzeczy z łańcuchami bloków, korporacyjne łańcuchy bloków i tokenizacja. Dodatkowo w książce znalazły się nowe rozdziały na temat zdecentralizowanych finansów, zdecentralizowanej tożsamości, prywatności, skalowalności i bezpieczeństwa w łańcuchach bloków, a także rozważania na temat przyszłości tej fascynującej technologii.

W trakcie lektury:

- poznasz mechanizmy działania bitcoina, Ethereum i innych łańcuchów bloków
- zrozumiesz zastosowania kryptografii w łańcuchach bloków
- poznasz algorytmy osiągania konsensusu i zasady tworzenia inteligentnych kontraktów
- nauczysz się zapewniać skalowalność, prywatność i bezpieczeństwo w łańcuchach bloków
- poznasz nowe trendy, w tym tożsamość zdecentralizowaną i suwerenną, DeFi, tokeny NFT i metaverse
- uzyskasz wgląd w przyszłość technologii łańcuchów bloków

Imran Bashir jest doświadczonym projektantem i architektem rozwiązań, zajmował się też zarządzaniem infrastrukturą i usługami informatycznymi, a także ochroną informacji. Obecnie zajmuje się łańcuchami bloków i obliczeniami kwantowymi. Jest członkiem IEEE. Piastował wysokie stanowiska techniczne w różnych organizacjach z całego świata.

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-0391-3
9 788328 903913
Cena: 169,00 zł
packt