
Table of Contents

Preface	xiii
Quick Glossary	xxv
1. What Is Ethereum?	1
Compared to Bitcoin	1
Components of a Blockchain	2
The Birth of Ethereum	3
Ethereum’s Four Stages of Development	5
Ethereum: A General-Purpose Blockchain	6
Ethereum’s Components	6
Further Reading	7
Ethereum and Turing Completeness	8
Turing Completeness as a “Feature”	8
Implications of Turing Completeness	9
From General-Purpose Blockchains to Decentralized Applications (DApps)	10
The Third Age of the Internet	10
Ethereum’s Development Culture	11
Why Learn Ethereum?	12
What This Book Will Teach You	12
2. Ethereum Basics	13
Ether Currency Units	13
Choosing an Ethereum Wallet	14
Control and Responsibility	15
Getting Started with MetaMask	17
Creating a Wallet	17
Switching Networks	20

Getting Some Test Ether	21
Sending Ether from MetaMask	22
Exploring the Transaction History of an Address	24
Introducing the World Computer	26
Externally Owned Accounts (EOAs) and Contracts	26
A Simple Contract: A Test Ether Faucet	27
Compiling the Faucet Contract	29
Creating the Contract on the Blockchain	31
Interacting with the Contract	33
Viewing the Contract Address in a Block Explorer	34
Funding the Contract	35
Withdrawing from Our Contract	36
Conclusions	39
3. Ethereum Clients.....	41
Ethereum Networks	42
Should I Run a Full Node?	42
Full Node Advantages and Disadvantages	43
Public Testnet Advantages and Disadvantages	44
Local Blockchain Simulation Advantages and Disadvantages	44
Running an Ethereum Client	45
Hardware Requirements for a Full Node	45
Software Requirements for Building and Running a Client (Node)	47
Parity	48
Go-Ethereum (Geth)	49
The First Synchronization of Ethereum-Based Blockchains	51
Running Geth or Parity	52
The JSON-RPC Interface	52
Remote Ethereum Clients	54
Mobile (Smartphone) Wallets	55
Browser Wallets	56
Conclusions	57
4. Cryptography.....	59
Keys and Addresses	59
Public Key Cryptography and Cryptocurrency	60
Private Keys	62
Generating a Private Key from a Random Number	63
Public Keys	64
Elliptic Curve Cryptography Explained	65
Elliptic Curve Arithmetic Operations	68
Generating a Public Key	69

Elliptic Curve Libraries	70
Cryptographic Hash Functions	70
Ethereum’s Cryptographic Hash Function: Keccak-256	72
Which Hash Function Am I Using?	73
Ethereum Addresses	73
Ethereum Address Formats	74
Inter Exchange Client Address Protocol	75
Hex Encoding with Checksum in Capitalization (EIP-55)	76
Conclusions	78
5. Wallets.....	79
Wallet Technology Overview	79
Nondeterministic (Random) Wallets	81
Deterministic (Seeded) Wallets	82
Hierarchical Deterministic Wallets (BIP-32/BIP-44)	82
Seeds and Mnemonic Codes (BIP-39)	83
Wallet Best Practices	84
Mnemonic Code Words (BIP-39)	85
Creating an HD Wallet from the Seed	92
HD Wallets (BIP-32) and Paths (BIP-43/44)	92
Conclusions	97
6. Transactions.....	99
The Structure of a Transaction	99
The Transaction Nonce	100
Keeping Track of Nonces	102
Gaps in Nonces, Duplicate Nonces, and Confirmation	104
Concurrency, Transaction Origination, and Nonces	104
Transaction Gas	105
Transaction Recipient	107
Transaction Value and Data	108
Transmitting Value to EOAs and Contracts	110
Transmitting a Data Payload to an EOA or Contract	110
Special Transaction: Contract Creation	112
Digital Signatures	115
The Elliptic Curve Digital Signature Algorithm	115
How Digital Signatures Work	115
Verifying the Signature	116
ECDSA Math	116
Transaction Signing in Practice	118
Raw Transaction Creation and Signing	119
Raw Transaction Creation with EIP-155	120

The Signature Prefix Value (v) and Public Key Recovery	120
Separating Signing and Transmission (Offline Signing)	121
Transaction Propagation	123
Recording on the Blockchain	124
Multiple-Signature (Multisig) Transactions	124
Conclusions	125
7. Smart Contracts and Solidity.....	127
What Is a Smart Contract?	127
Life Cycle of a Smart Contract	128
Introduction to Ethereum High-Level Languages	129
Building a Smart Contract with Solidity	131
Selecting a Version of Solidity	132
Download and Install	132
Development Environment	133
Writing a Simple Solidity Program	133
Compiling with the Solidity Compiler (solc)	134
The Ethereum Contract ABI	134
Selecting a Solidity Compiler and Language Version	135
Programming with Solidity	136
Data Types	136
Predefined Global Variables and Functions	138
Contract Definition	141
Functions	141
Contract Constructor and selfdestruct	143
Adding a Constructor and selfdestruct to Our Faucet Example	144
Function Modifiers	145
Contract Inheritance	146
Error Handling (assert, require, revert)	148
Events	149
Calling Other Contracts (send, call, callcode, delegatecall)	152
Gas Considerations	158
Avoid Dynamically Sized Arrays	158
Avoid Calls to Other Contracts	158
Estimating Gas Cost	158
Conclusions	160
8. Smart Contracts and Vyper.....	161
Vulnerabilities and Vyper	161
Comparison to Solidity	162
Modifiers	162
Class Inheritance	163

Inline Assembly	164
Function Overloading	164
Variable Typecasting	164
Preconditions and Postconditions	166
Decorators	166
Function and Variable Ordering	167
Compilation	168
Protecting Against Overflow Errors at the Compiler Level	169
Reading and Writing Data	169
Conclusions	170
9. Smart Contract Security	171
Security Best Practices	171
Security Risks and Antipatterns	172
Reentrancy	173
Real-World Example: The DAO	177
Arithmetic Over/Underflows	177
Real-World Examples: PoWHC and Batch Transfer Overflow (CVE-2018-10299)	181
Unexpected Ether	181
Further Examples	185
DELEGATECALL	185
Real-World Example: Parity Multisig Wallet (Second Hack)	190
Default Visibilities	191
Real-World Example: Parity Multisig Wallet (First Hack)	192
Entropy Illusion	193
Real-World Example: PRNG Contracts	195
External Contract Referencing	195
Real-World Example: Reentrancy Honey Pot	199
Short Address/Parameter Attack	200
Unchecked CALL Return Values	202
Real-World Example: Etherpot and King of the Ether	203
Race Conditions/Front Running	204
Real-World Examples: ERC20 and Bancor	207
Denial of Service (DoS)	207
Real-World Examples: GovernMental	210
Block Timestamp Manipulation	210
Real-World Example: GovernMental	212
Constructors with Care	212
Real-World Example: Rubixi	213
Uninitialized Storage Pointers	213

Real-World Examples: OpenAddressLottery and CryptoRoulette Honey Pots	215
Floating Point and Precision	216
Real-World Example: Ethstick	217
Tx.Origin Authentication	218
Contract Libraries	219
Conclusions	220
10. Tokens.....	221
How Tokens Are Used	221
Tokens and Fungibility	223
Counterparty Risk	223
Tokens and Intrinsicity	224
Using Tokens: Utility or Equity	224
It's a Duck!	225
Utility Tokens: Who Needs Them?	225
Tokens on Ethereum	227
The ERC20 Token Standard	227
Launching Our Own ERC20 Token	231
Issues with ERC20 Tokens	242
ERC223: A Proposed Token Contract Interface Standard	244
ERC777: A Proposed Token Contract Interface Standard	244
ERC721: Non-fungible Token (Deed) Standard	247
Using Token Standards	248
What Are Token Standards? What Is Their Purpose?	249
Should You Use These Standards?	249
Security by Maturity	250
Extensions to Token Interface Standards	250
Tokens and ICOs	251
Conclusions	252
11. Oracles.....	253
Why Oracles Are Needed	253
Oracle Use Cases and Examples	254
Oracle Design Patterns	255
Data Authentication	258
Computation Oracles	259
Decentralized Oracles	261
Oracle Client Interfaces in Solidity	262
Conclusions	265

12. Decentralized Applications (DApps).....	267
What Is a DApp?	268
Backend (Smart Contract)	269
Frontend (Web User Interface)	269
Data Storage	270
Decentralized Message Communications Protocols	271
A Basic DApp Example: Auction DApp	271
Auction DApp: Backend Smart Contracts	272
Auction DApp: Frontend User Interface	275
Further Decentralizing the Auction DApp	277
Storing the Auction DApp on Swarm	278
Preparing Swarm	278
Uploading Files to Swarm	279
The Ethereum Name Service (ENS)	281
History of Ethereum Name Services	281
The ENS Specification	282
Bottom Layer: Name Owners and Resolvers	282
Middle Layer: The .eth Nodes	284
Top Layer: The Deeds	286
Registering a Name	286
Managing Your ENS Name	290
ENS Resolvers	292
Resolving a Name to a Swarm Hash (Content)	293
From App to DApp	295
Conclusions	296
13. The Ethereum Virtual Machine.....	297
What Is the EVM?	297
Comparison with Existing Technology	300
The EVM Instruction Set (Bytecode Operations)	300
Ethereum State	303
Compiling Solidity to EVM Bytecode	304
Contract Deployment Code	307
Disassembling the Bytecode	308
Turing Completeness and Gas	314
Gas	314
Gas Accounting During Execution	315
Gas Accounting Considerations	315
Gas Cost Versus Gas Price	316
Block Gas Limit	317
Conclusions	318

14. Consensus.....	319
Consensus via Proof of Work	320
Consensus via Proof of Stake (PoS)	320
Ethash: Ethereum’s Proof-of-Work Algorithm	321
Casper: Ethereum’s Proof-of-Stake Algorithm	322
Principles of Consensus	323
Controversy and Competition	323
Conclusions	324
A. Ethereum Fork History.....	325
B. Ethereum Standards.....	333
C. Ethereum EVM Opcodes and Gas Consumption.....	339
D. Development Tools, Frameworks, and Libraries.....	345
E. web3.js Tutorial.....	365
F. Short Links Reference.....	369
Index.....	371