

超级账本核心设计和开发者撰写，区块链开发落地专业指南。

由浅入深，详细讲解超级账本Fabric 1.0架构设计与应用开发。

# 区块链

## 原理、设计与应用

杨保华 陈昌 编著



机械工业出版社  
China Machine Press

## 作者简介

### 杨保华

博士，毕业于清华大学。超级账本（Hyperledger）大中华区技术工作组主席，IBM 大中华区 Blockchain 技术社区首席顾问，资深研究员。曾主持多个大规模系统平台的架构设计和研发实施，是区块链、云计算、大数据等技术的早期研究者和实践者。他热爱开源技术，曾贡献于 OpenStack、OpenDaylight 等开源项目，是超级账本 Fabric 项目的核心设计和开发者，Cello 和 Fabric-SDK-Py 项目的发起人。个人主页为 <https://yeasy.github.com>。

### 陈昌

毕业于清华大学。纸贵科技 CTO，曾任 IBM 高级研究员。技术方向包括云计算、区块链、机器学习等。他是区块链技术的早期研究和推动者，是超级账本（Hyperledger）项目的核心开发者。他有丰富的区块链应用实践经验，曾负责金融行业区块链解决方案的架构设计和实施，并主导开发了若干区块链服务平台。

区块链  
技术丛书

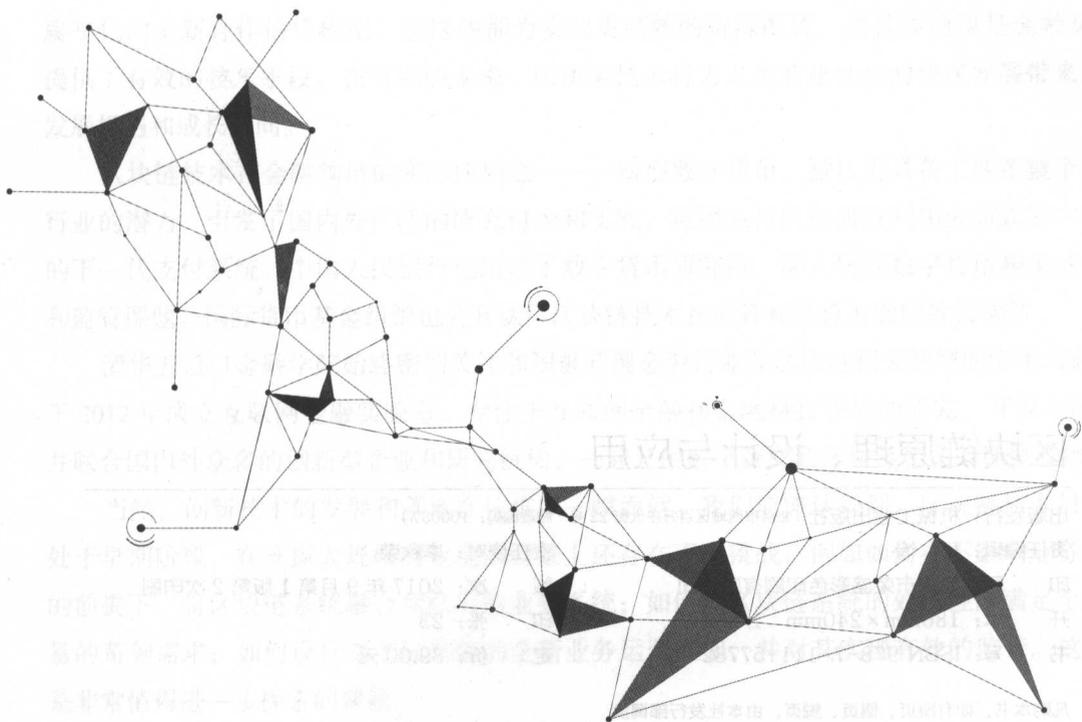
图书在版编目(CIP)数据

区块链：原理、设计与应用 / 杨保华，陈昌 编著. — 北京：机械工业出版社，2017.8

# 区块链

## 原理、设计与应用

杨保华 陈昌 编著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

区块链原理、设计与应用 / 杨保华, 陈昌编著. —北京: 机械工业出版社, 2017.8 (2017.9重印)

(区块链技术丛书)

ISBN 978-7-111-57782-9

I. 区… II. ①杨… ②陈… III. 电子商务-支付方式-研究 IV. F713.361.3

中国版本图书馆 CIP 数据核字 (2017) 第 197360 号

## 区块链原理、设计与应用

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴怡

责任校对: 李秋荣

印刷: 北京市荣盛彩色印刷有限公司

版次: 2017 年 9 月第 1 版第 2 次印刷

开本: 186mm × 240mm 1/16

印张: 23

书号: ISBN 978-7-111-57782-9

定价: 89.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

金融是人类文明发展过程中经济运行的基础，自诞生起，金融领域就伴随经济发展的阶段和商业模式的变迁不断涌现出先进的技术手段，这些都大大提升了社会和经济的运转效率。从延续了近千年的纸质记账，到二十世纪的电子化交易，再到影响现在及未来的互联网、大数据、人工智能和区块链，金融行业和金融科技领域始终以开放的姿态迎接新技术和新变化，并不断进行自我革新和升华。

区块链技术是金融科技领域当下最受人关注的方向之一。区块链作为一个新兴技术，具备去中心化、防篡改、可追溯等众多金融领域十分需要的特点。它可以实现多方场景下开放、扁平化的全新合作信任模型，而这些都为实现更高效的资源配置，更具体地说是金融交易，提供了有效的技术手段。在可见的未来，区块链技术将为人类商业社会的快速发展带来更多发展机遇和成长空间。

区块链技术在金融领域的实际应用之一——新型数字货币，被认为具备了变革整个金融行业的潜力，引发了国内外广泛的研究讨论和实践。英国央行已在研发利用分布式账本技术的下一代支付系统。中国人民银行也组建了数字货币研究所，深入研究数字货币相关的技术和监管课题。国际货币基金组织也公开认可区块链技术在清算和结算方面的独特优势。

清华五道口金融学院始终密切关注和积极开展金融行业及区块链相关领域的学术与研究，于2012年成立互联网金融实验室，专注于互联网金融和金融科技领域的研究、开发与孵化，并联合国内外众多的创新型企业 and 研究机构，一起开展数字资产和区块链相关的课题和项目。

当然，创新技术的发展和落地往往难以一蹴而就。我们应该认识到，区块链技术目前仍处于早期阶段，在支撑大规模商业应用场景上还存在不少挑战，例如如何在不影响业务运行的前提下，将区块链系统融合到已有的业务系统；如何让区块链系统的处理性能满足金融交易的苛刻需求；如何设计基于区块链的全新业务运营框架，并对其实现有效的监管。这些都是非常值得进一步探索的课题。

在此之际，很欣喜地看到有这样一本系统讲解区块链技术及实践的书籍出版。与其他介绍区块链的图书不同，本书并没有局限在阐述区块链的思想、概念和应用场景等理论知识层面，而是进一步从实现角度剖析了区块链平台的架构、设计，并提供了大量一手的开发实践案例，特别是全球区块链领域首屈一指的开源项目——超级账本。这些都将帮助读者更深刻

图书在版编目(CIP)数据

地理理解和掌握区块链技术的核心原理与应用方法。

本书作者在技术体系的经验和视野、创新意识、国际化合作等方面都展现出了作为金融科技专家的综合素养，让我们对中国金融业进入下一个全新的发展阶段的人才储备充满了信心。我们愿意跟作者们一起，共同关注、共同努力于中国金融科技的未来。

廖理，教授，博士生导师，清华大学五道口金融学院

2017年8月于清华五道口

## Preface 前 言

区块链和机器学习被誉为未来十年内最有可能提高人类社会生产力的两大创新科技。如果说机器学习的兴起依赖于新型芯片技术的发展，那么区块链技术的出现，则是来自商业、金融、信息、安全等多个领域众多科技成果和业务创新的共同推动。

比特币网络自横空出世，以前所未有的新型理念支持了前所未有的交易模式；以太坊项目站在前人肩膀上，引入图灵完备的智能合约机制，进一步释放了区块链技术的应用威力；众多商业、科技巨头，集合来自大型企业的应用需求和最先进的技术成果，打造出支持权限管理的联盟式分布式账本平台——超级账本……开源技术从未如今天这样，对各行各业都产生着极为深远的影响。本书在剖析区块链核心技术时，正是以这些开源项目（特别是超级账本 Fabric 项目）为具体实现进行讲解，力图探索其核心思想，展现其设计精华，剖析其应用特性。

我们在写作中秉承了由浅入深、由理论到实践的思想，将全书分为两大部分：理论篇和实践篇。前三章介绍了区块链技术的由来、核心思想及典型的应用场景。第 4 ~ 5 章重点介绍了区块链技术中大量出现的分布式系统技术和密码学安全技术。第 6 ~ 8 章分别介绍了区块链领域的三个典型开源项目：比特币、以太坊和超级账本。第 9 ~ 11 章以超级账本 Fabric 项目为例，具体讲解了安装部署、配置管理，以及使用 Fabric CA 进行证书管理的实践经验。第 12 章重点剖析了超级账本 Fabric 项目的核心架构设计。第 13 章介绍了区块链应用开发的相关技巧和示例。最后，本书还就热门的“区块链即服务”平台进行了介绍，并讲解应用超级账本 Cello 项目构建区块链服务和管理平台的相关经验和知识。

相信读者在阅读完本书后，在深入理解区块链核心概念和原理的同时，对于区块链和分布式账本领域最新的技术和典型设计实现也能了然于心，可以更加高效地开发基于区块链平台的分布式应用。

在本书长达两年时间的编写过程中，得到了来自家人、同事以及开源社区开发者和技术爱好者的众多支持和鼓励，在此表示感谢！

最后，希望本书能为推动区块链技术的进步和开源文化的普及做出一点微薄的贡献！

作者

2017 年 8 月于北京

# 目 录 Contents

|                     |                      |           |
|---------------------|----------------------|-----------|
| 序 言                 | 3.2.1 银行业金融管理        | 22        |
| 前 言                 | 3.2.2 证券交易           | 24        |
|                     | 3.2.3 众筹投资           | 25        |
|                     | 3.3 征信和权属管理          | 26        |
|                     | 3.4 资源共享             | 28        |
|                     | 3.5 贸易管理             | 29        |
|                     | 3.6 物联网              | 30        |
|                     | 3.7 其他场景             | 31        |
|                     | 3.8 本章小结             | 33        |
| <b>理论篇</b>          | <b>第4章 分布式系统核心问题</b> | <b>34</b> |
| <b>第1章 区块链思想的诞生</b> | 4.1 一致性问题            | 34        |
| 1.1 从实体货币到数字货币      | 4.1.1 定义与重要性         | 34        |
| 1.2 站在巨人的肩膀上        | 4.1.2 问题与挑战          | 35        |
| 1.3 了不起的社会学实验       | 4.1.3 一致性要求          | 36        |
| 1.4 潜在的商业价值         | 4.1.4 带约束的一致性        | 36        |
| 1.5 本章小结            | 4.2 共识算法             | 37        |
| <b>第2章 核心技术概览</b>   | 4.2.1 问题与挑战          | 38        |
| 2.1 定义与原理           | 4.2.2 常见算法           | 38        |
| 2.2 技术的演化与分类        | 4.2.3 理论界限           | 38        |
| 2.3 关键问题和挑战         | 4.3 FLP不可能原理         | 39        |
| 2.4 趋势与展望           | 4.3.1 定义             | 39        |
| 2.5 认识上的误区          | 4.3.2 正确理解           | 39        |
| 2.6 本章小结            | 4.4 CAP原理            | 40        |
| <b>第3章 典型应用场景</b>   |                      |           |
| 3.1 应用场景概览          |                      |           |
| 3.2 金融服务            |                      |           |

|              |                      |           |              |                   |           |
|--------------|----------------------|-----------|--------------|-------------------|-----------|
| 4.4.1        | 定义                   | 40        | 5.3.3        | 安全性               | 59        |
| 4.4.2        | 应用场景                 | 41        | 5.4          | 数字证书              | 59        |
| 4.5          | ACID 原则              | 41        | 5.4.1        | X.509 证书规范        | 60        |
| 4.6          | Paxos 算法与 Raft 算法    | 42        | 5.4.2        | 证书格式              | 61        |
| 4.6.1        | Paxos 算法             | 42        | 5.4.3        | 证书信任链             | 62        |
| 4.6.2        | Raft 算法              | 45        | 5.5          | PKI 体系            | 63        |
| 4.7          | 拜占庭问题与算法             | 45        | 5.5.1        | PKI 基本组件          | 63        |
| 4.8          | 可靠性指标                | 48        | 5.5.2        | 证书的签发             | 63        |
| 4.8.1        | 几个 9 的指标             | 48        | 5.5.3        | 证书的撤销             | 66        |
| 4.8.2        | 两个核心时间               | 49        | 5.6          | Merkle 树结构        | 66        |
| 4.8.3        | 提高可靠性                | 49        | 5.7          | 布隆过滤器             | 67        |
| 4.9          | 本章小结                 | 49        | 5.8          | 同态加密              | 68        |
| <b>第 5 章</b> | <b>密码学与安全技术</b>      | <b>50</b> | 5.9          | 其他问题              | 70        |
| 5.1          | Hash 算法与数字摘要         | 50        | 5.10         | 本章小结              | 71        |
| 5.1.1        | Hash 定义              | 50        | <b>第 6 章</b> | <b>比特币——区块链思想</b> |           |
| 5.1.2        | 常见算法                 | 51        |              | <b>诞生的摇篮</b>      | <b>72</b> |
| 5.1.3        | 性能                   | 51        | 6.1          | 比特币项目简介           | 72        |
| 5.1.4        | 数字摘要                 | 52        | 6.1.1        | 比特币大事记            | 73        |
| 5.1.5        | Hash 攻击与防护           | 52        | 6.1.2        | 其他数字货币            | 74        |
| 5.2          | 加解密算法                | 52        | 6.2          | 原理和设计             | 75        |
| 5.2.1        | 加解密系统基本组成            | 53        | 6.2.1        | 基本交易过程            | 75        |
| 5.2.2        | 对称加密算法               | 53        | 6.2.2        | 重要概念              | 76        |
| 5.2.3        | 非对称加密算法              | 54        | 6.2.3        | 创新设计              | 78        |
| 5.2.4        | 选择明文攻击               | 55        | 6.3          | 挖矿                | 80        |
| 5.2.5        | 混合加密机制               | 56        | 6.3.1        | 基本原理              | 80        |
| 5.2.6        | 离散对数与 Diffie-Hellman |           | 6.3.2        | 挖矿过程              | 81        |
|              | 密钥交换协议               | 57        | 6.3.3        | 如何看待挖矿            | 81        |
| 5.3          | 消息认证码与数字签名           | 57        | 6.4          | 共识机制              | 82        |
| 5.3.1        | 消息认证码                | 58        | 6.4.1        | 工作量证明             | 82        |
| 5.3.2        | 数字签名                 | 58        | 6.4.2        | 权益证明              | 83        |

|                             |           |     |                               |                        |     |
|-----------------------------|-----------|-----|-------------------------------|------------------------|-----|
| 6.5                         | 闪电网络      | 83  | 7.5                           | 安装客户端                  | 100 |
| 6.6                         | 侧链        | 85  | 7.5.1                         | 从 PPA 直接安装             | 100 |
| 6.6.1                       | SPV 证明    | 85  | 7.5.2                         | 从源码编译                  | 101 |
| 6.6.2                       | 双向挂钩      | 86  | 7.6                           | 使用智能合约                 | 102 |
| 6.6.3                       | 最新进展      | 87  | 7.6.1                         | 搭建测试用区块链               | 102 |
| 6.7                         | 热点问题      | 87  | 7.6.2                         | 创建和编译智能合约              | 104 |
| 6.7.1                       | 设计中的权衡    | 87  | 7.6.3                         | 部署智能合约                 | 105 |
| 6.7.2                       | 分叉        | 87  | 7.6.4                         | 调用智能合约                 | 106 |
| 6.7.3                       | 交易延展性     | 88  | 7.7                           | 智能合约案例: 投票             | 106 |
| 6.7.4                       | 扩容之争      | 89  | 7.7.1                         | 智能合约代码                 | 107 |
| 6.7.5                       | 比特币的监管和追踪 | 90  | 7.7.2                         | 代码解析                   | 109 |
| 6.8                         | 相关工具      | 91  | 7.8                           | 本章小结                   | 111 |
| 6.9                         | 本章小结      | 92  |                               |                        |     |
| <b>第 7 章 以太坊——挣脱数字货币的枷锁</b> |           |     | <b>第 8 章 超级账本——面向企业的分布式账本</b> |                        |     |
| 7.1                         | 以太坊项目简介   | 93  | 8.1                           | 超级账本项目简介               | 112 |
| 7.1.1                       | 以太坊项目简史   | 94  | 8.2                           | 社区组织结构                 | 114 |
| 7.1.2                       | 主要特点      | 95  | 8.2.1                         | 基本结构                   | 114 |
| 7.2                         | 核心概念      | 95  | 8.2.2                         | 大中华区技术工作组              | 114 |
| 7.3                         | 主要设计      | 97  | 8.3                           | 顶级项目介绍                 | 115 |
| 7.3.1                       | 智能合约相关设计  | 97  | 8.3.1                         | Fabric 项目              | 116 |
| 7.3.2                       | 交易模型      | 97  | 8.3.2                         | Sawtooth 项目            | 117 |
| 7.3.3                       | 共识        | 97  | 8.3.3                         | Iroha 项目               | 117 |
| 7.3.4                       | 降低攻击      | 98  | 8.3.4                         | Blockchain Explorer 项目 | 117 |
| 7.3.5                       | 提高扩展性     | 98  | 8.3.5                         | Cello 项目               | 118 |
| 7.4                         | 相关工具      | 98  | 8.3.6                         | Indy 项目                | 118 |
| 7.4.1                       | 客户端和开发库   | 98  | 8.3.7                         | Composer 项目            | 118 |
| 7.4.2                       | 以太坊钱包     | 99  | 8.3.8                         | Burrow 项目              | 119 |
| 7.4.3                       | IDE       | 100 | 8.4                           | 开发必备工具                 | 119 |
| 7.4.4                       | 网站资源      | 100 | 8.4.1                         | Linux Foundation ID    | 119 |
|                             |           |     | 8.4.2                         | Jira——任务和进度管理          | 119 |

|       |                         |     |                                |            |     |
|-------|-------------------------|-----|--------------------------------|------------|-----|
| 8.4.3 | Gerrit——代码仓库和 Review 管理 | 120 | 9.4.4                          | 启动 Peer 节点 | 151 |
| 8.4.4 | RocketChat——在线沟通        | 121 | 9.4.5                          | 操作网络       | 152 |
| 8.5   | 贡献代码                    | 121 | 9.4.6                          | 基于容器方式     | 156 |
| 8.6   | 本章小结                    | 126 | 9.5                            | 链码的概念与使用   | 157 |
|       |                         |     | 9.5.1                          | 链码操作命令     | 158 |
|       |                         |     | 9.5.2                          | 命令参数       | 158 |
|       |                         |     | 9.5.3                          | 安装链码       | 159 |
|       |                         |     | 9.5.4                          | 实例化链码      | 162 |
|       |                         |     | 9.5.5                          | 调用链码       | 165 |
|       |                         |     | 9.5.6                          | 查询链码       | 167 |
|       |                         |     | 9.5.7                          | 升级链码       | 168 |
|       |                         |     | 9.5.8                          | 打包链码和签名    | 169 |
|       |                         |     | 9.6                            | 使用多通道      | 170 |
|       |                         |     | 9.6.1                          | 通道操作命令     | 170 |
|       |                         |     | 9.6.2                          | 命令选项       | 171 |
|       |                         |     | 9.6.3                          | 创建通道       | 172 |
|       |                         |     | 9.6.4                          | 加入通道       | 174 |
|       |                         |     | 9.6.5                          | 列出所加入的通道   | 175 |
|       |                         |     | 9.6.6                          | 获取某区块      | 176 |
|       |                         |     | 9.6.7                          | 更新通道配置     | 177 |
|       |                         |     | 9.7                            | SDK 支持     | 178 |
|       |                         |     | 9.8                            | 生产环境注意事项   | 179 |
|       |                         |     | 9.9                            | 本章小结       | 181 |
|       |                         |     | <b>第 10 章 超级账本 Fabric 配置管理</b> | <b>182</b> |     |
|       |                         |     | 10.1                           | 简介         | 182 |
|       |                         |     | 10.1.1                         | 配置文件       | 182 |
|       |                         |     | 10.1.2                         | 配置管理工具     | 183 |
|       |                         |     | 10.2                           | Peer 配置剖析  | 183 |
|       |                         |     | 10.2.1                         | logging 部分 | 184 |
|       |                         |     | 10.2.2                         | peer 部分    | 184 |

## 实 践 篇

### 第 9 章 超级账本 Fabric 部署和使用 128

|        |                        |     |
|--------|------------------------|-----|
| 9.1    | 简介                     | 128 |
| 9.2    | 本地编译安装                 | 129 |
| 9.2.1  | 操作系统                   | 130 |
| 9.2.2  | 环境配置                   | 130 |
| 9.2.3  | 获取代码                   | 131 |
| 9.2.4  | 编译安装 fabric-peer 组件    | 131 |
| 9.2.5  | 编译安装 fabric-orderer 组件 | 132 |
| 9.2.6  | 编译安装 fabric-ca 组件      | 133 |
| 9.2.7  | 编译安装辅助工具               | 133 |
| 9.2.8  | 获取 chaintool           | 133 |
| 9.2.9  | 安装 Go 语言相关工具           | 134 |
| 9.2.10 | 示例配置                   | 134 |
| 9.3    | 使用 Docker 镜像           | 134 |
| 9.3.1  | 安装 Docker 服务           | 134 |
| 9.3.2  | 安装 docker-compose      | 135 |
| 9.3.3  | 获取 Docker 镜像           | 135 |
| 9.3.4  | 镜像 Dockerfile          | 138 |
| 9.4    | 启动 Fabric 网络           | 143 |
| 9.4.1  | 网络拓扑                   | 143 |
| 9.4.2  | 准备相关配置文件               | 144 |
| 9.4.3  | 启动 Orderer 节点          | 150 |

|        |                          |     |
|--------|--------------------------|-----|
| 10.2.3 | vm 部分                    | 188 |
| 10.2.4 | chaincode 部分             | 189 |
| 10.2.5 | ledger 部分                | 190 |
| 10.3   | Orderer 配置剖析             | 191 |
| 10.4   | cryptogen 生成组织身份配置       | 194 |
| 10.4.1 | 配置文件                     | 195 |
| 10.4.2 | 子命令和参数                   | 196 |
| 10.4.3 | 生成密钥和证书文件                | 196 |
| 10.4.4 | 查看配置模板信息                 | 198 |
| 10.5   | configtxgen 生成通道配置       | 199 |
| 10.5.1 | configtx.yaml 配置文件       | 199 |
| 10.5.2 | 命令选项                     | 203 |
| 10.5.3 | 生成 Orderer 初始区块并<br>进行查看 | 203 |
| 10.5.4 | 生成新建通道交易文件并<br>进行查看      | 211 |
| 10.5.5 | 生成锚节点更新交易文件              | 215 |
| 10.6   | configtxlator 转换配置       | 215 |
| 10.6.1 | RESTful 接口               | 215 |
| 10.6.2 | 解码为 Json 格式              | 216 |
| 10.6.3 | 编码为二进制格式                 | 217 |
| 10.6.4 | 计算配置更新量                  | 217 |
| 10.6.5 | 更新通道配置                   | 218 |
| 10.7   | 本章小结                     | 219 |

## 第 11 章 超级账本 Fabric CA 应用 与配置

|        |                 |     |
|--------|-----------------|-----|
| 11.1   | 简介              | 220 |
| 11.2   | 安装服务端和客户端       | 221 |
| 11.2.1 | 本地编译            | 221 |
| 11.2.2 | 获取和使用 Docker 镜像 | 223 |

|        |               |     |
|--------|---------------|-----|
| 11.2.3 | 示例 Dockerfile | 223 |
| 11.3   | 启动 CA 服务      | 225 |
| 11.4   | 服务端命令剖析       | 228 |
| 11.4.1 | 全局命令参数        | 228 |
| 11.4.2 | init 命令       | 230 |
| 11.4.3 | start 命令      | 230 |
| 11.5   | 服务端配置文件解析     | 231 |
| 11.6   | 与服务端进行交互      | 235 |
| 11.7   | 客户端命令剖析       | 237 |
| 11.7.1 | 全局命令参数        | 237 |
| 11.7.2 | enroll 命令     | 239 |
| 11.7.3 | getcacert 命令  | 240 |
| 11.7.4 | reenroll 命令   | 241 |
| 11.7.5 | register 命令   | 241 |
| 11.7.6 | revoke 命令     | 242 |
| 11.8   | 客户端配置文件解析     | 243 |
| 11.9   | 生产环境部署        | 245 |
| 11.10  | 本章小结          | 247 |

## 第 12 章 超级账本 Fabric 架构与设计

|        |               |     |
|--------|---------------|-----|
| 12.1   | 整体架构概览        | 248 |
| 12.1.1 | 核心特性          | 248 |
| 12.1.2 | 整体架构          | 249 |
| 12.1.3 | 典型工作流程        | 249 |
| 12.2   | 核心概念与组件       | 251 |
| 12.2.1 | 网络层相关组件       | 252 |
| 12.2.2 | 共识相关组件        | 254 |
| 12.2.3 | 权限管理相关组件      | 255 |
| 12.2.4 | 业务层相关组件       | 257 |
| 12.3   | gRPC 消息协议     | 262 |
| 12.3.1 | Envelope 消息结构 | 262 |

|               |                          |            |               |                          |            |
|---------------|--------------------------|------------|---------------|--------------------------|------------|
| 12.3.2        | 客户端访问 Peer 节点            | 263        | 13.3          | 链码开发 API                 | 295        |
| 12.3.3        | 客户端、Peer 节点访问<br>Orderer | 265        | 13.3.1        | 账本状态交互 API               | 296        |
| 12.3.4        | 链码容器和 Peer 节点之间<br>的操作   | 265        | 13.3.2        | 交易信息相关 API               | 296        |
| 12.3.5        | 多个节点之间的操作                | 266        | 13.3.3        | 参数读取 API                 | 297        |
| 12.4          | 权限管理和策略                  | 267        | 13.3.4        | 其他 API                   | 297        |
| 12.4.1        | 策略应用场景                   | 267        | 13.4          | 应用开发案例一：<br>转账           | 298        |
| 12.4.2        | 身份证书                     | 268        | 13.4.1        | 链码结构                     | 298        |
| 12.4.3        | 权限策略的实现                  | 268        | 13.4.2        | Init 方法                  | 299        |
| 12.4.4        | 通道策略                     | 272        | 13.4.3        | Invoke 方法                | 300        |
| 12.4.5        | 背书策略                     | 273        | 13.5          | 应用开发案例二：<br>资产权属管理       | 301        |
| 12.4.6        | 实例化策略                    | 273        | 13.5.1        | 链码结构                     | 301        |
| 12.5          | 用户链码                     | 274        | 13.5.2        | Invoke 方法                | 303        |
| 12.5.1        | 基本结构                     | 274        | 13.6          | 应用开发案例三：<br>调用其他链码       | 312        |
| 12.5.2        | 链码与 Peer 的交互过程           | 275        | 13.7          | 应用开发案例四：<br>发送事件         | 313        |
| 12.5.3        | 链码处理状态机                  | 277        | 13.8          | 开发最佳实践小结                 | 314        |
| 12.6          | 系统链码                     | 279        | 13.9          | 本章小结                     | 316        |
| 12.7          | 排序服务                     | 281        | <b>第 14 章</b> | <b>区块链服务平台设计</b>         | <b>317</b> |
| 12.7.1        | gRPC 服务接口                | 282        | 14.1          | 简介                       | 317        |
| 12.7.2        | 链和账本管理                   | 283        | 14.1.1        | 参考架构                     | 318        |
| 12.7.3        | 通道配置更新                   | 284        | 14.1.2        | 考量指标                     | 318        |
| 12.7.4        | 共识插件                     | 286        | 14.2          | IBM Bluemix 云区块链<br>服务   | 319        |
| 12.8          | 本章小结                     | 288        | 14.3          | 微软 Azure 云区块链服务          | 321        |
| <b>第 13 章</b> | <b>区块链应用开发</b>           | <b>290</b> | 14.4          | 使用超级账本 Cello 搭建<br>区块链服务 | 324        |
| 13.1          | 简介                       | 290        | 14.4.1        | 基本架构和特性                  | 324        |
| 13.2          | 链码的原理、接口与结构              | 292        |               |                          |            |
| 13.2.1        | Chaincode 接口             | 292        |               |                          |            |
| 13.2.2        | 链码结构                     | 293        |               |                          |            |
| 13.2.3        | 链码基本工作原理                 | 294        |               |                          |            |

14.4.2 环境准备 ..... 325

14.4.3 下载 Cello 源码 ..... 325

14.4.4 配置 Worker 节点 ..... 325

14.4.5 配置 Master 节点 ..... 326

14.4.6 使用 Cello 管理区块链 ..... 327

14.4.7 基于 Cello 进行功能扩展 ..... 330

14.5 本章小结 ..... 330

## 附 录

附录 A 术语表 ..... 334

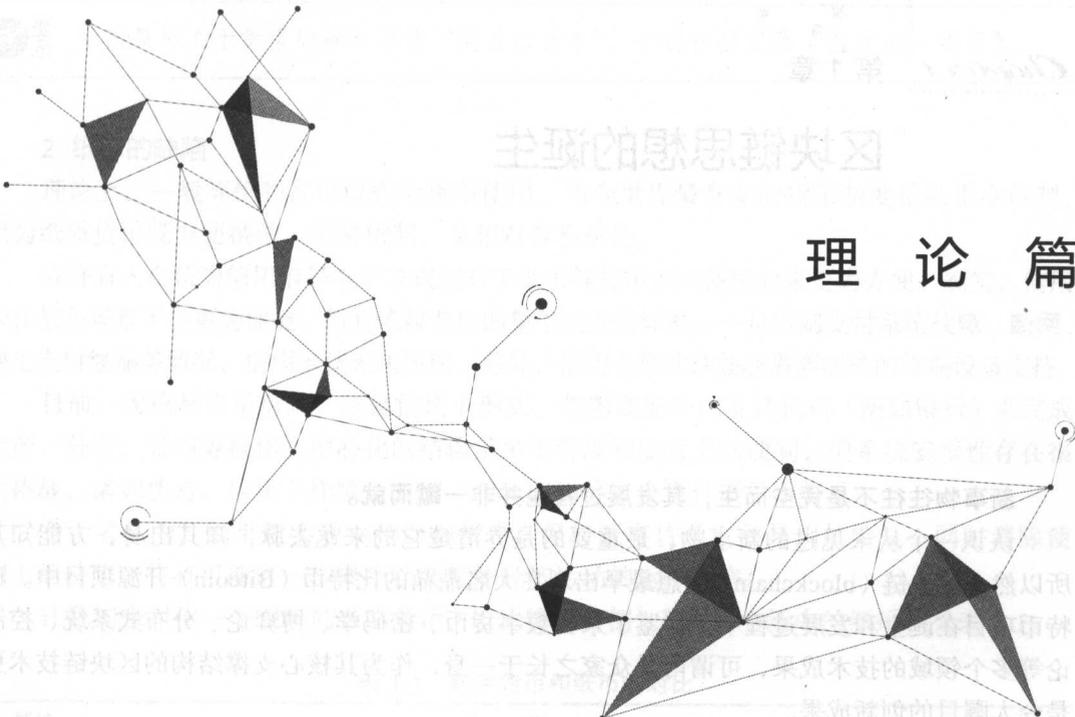
附录 B 常见问题解答 ..... 338

附录 C Golang 开发相关 ..... 342

附录 D ProtoBuf 与 gRPC ..... 349

附录 E 参考资源 ..... 353

# 理论篇

- 
- 第1章 区块链思想的诞生
  - 第2章 核心技术概览
  - 第3章 典型应用场景
  - 第4章 分布式系统核心问题
  - 第5章 密码学与安全技术
  - 第6章 比特币——区块链思想诞生的摇篮
  - 第7章 以太坊——挣脱数字货币的枷锁
  - 第8章 超级账本——面向企业的分布式账本

# 区块链思想的诞生

新事物往往不是凭空而生，其发展过程也并非一蹴而就。

认识一个从未见过的新事物，最重要的是弄清楚它的来龙去脉，知其出身，方能知其所以然。区块链（blockchain）思想最早出现在大名鼎鼎的比特币（Bitcoin）开源项目中。比特币项目在诞生和发展过程中，借鉴了来自数字货币、密码学、博弈论、分布式系统、控制论等多个领域的技术成果，可谓博采众家之长于一身，作为其核心支撑结构的区块链技术更是令人瞩目的创新成果。

本章将从数字货币的历史讲起，简要介绍区块链思想诞生的摇篮——比特币项目的诞生和发展过程，并初步剖析区块链技术带来的潜在商业价值。通过阅读本章内容，读者可以了解区块链技术产生的背景、原因，以及在诸多商业应用场景中的潜在价值。

## 1.1 从实体货币到数字货币

区块链最初的思想诞生于无数先哲对于用数字货币替代实体货币的探讨和设计中。

### 1. 货币的历史演化

众所周知，货币是人类文明发展过程中的一大发明。其最重要的职能包括价值尺度、流通手段、贮藏手段等。很难想象离开了货币，现代社会庞大而复杂的经济和金融体系如何保持运转。也正是因为它如此重要，货币的设计和发行机制是关系到国计民生的大事。

历史上，在自然和人为因素的干预下，货币的形态经历了多个阶段的演化，包括实物货币、金属货币、代用货币、信用货币、电子货币、数字货币等。近代以前相当长的一段时间里，货币的形态一直是以实体的形式存在，可统称为“实体货币”。计算机诞生后，为货

币的虚拟化提供了可能性。

同时，货币自身的价值依托也不断发生演化，从最早的实物价值、发行方信用价值，直到今天的对科学技术和信息系统（包括算法、数学、密码学、软件等）的信任价值。



中国最早关于货币的确切记载“夏后以玄币”，出现在恒宽的《盐铁论·错币》。

## 2. 纸币的缺陷

理论上，一般等价物都可以作为货币使用。当今世界最常见的货币制度是纸币本位制，因为纸质货币既方便携带、不易仿制，又相对容易辨伪。

或许有人会认为信用卡等电子方式相对于纸币等货币形式使用起来更为方便。确实，信用卡在某些场景下会更为便捷，但它依赖背后的集中式支付体系，一旦碰到支付系统故障、断网、缺乏支付终端等情况，信用卡就无法使用。另外，信用卡形式往往还需要额外的终端设备支持。

目前，无论是货币形式，还是信用卡形式，都需要额外的支持机构（例如银行）来完成生产、分发、管理等操作。中心化的结构带来了管理和监管上的便利，但系统安全性存在很大挑战，诸如伪造、信用卡诈骗、盗刷、转账骗局等安全事件屡见不鲜。

有学者认为，如果能实现一种新型货币，保持既有货币方便易用的特性，同时消除使用上的缺陷，将有可能进一步提高社会整体经济活动的运作效率。

让我们来对比一下现有的数字货币（以比特币为例）和现实生活中的纸币，见表 1-1。

表 1-1 数字货币和纸币的对比

| 属性 | 分析  | 优势方  |
|----|---|------|
| 便携 | 大部分场景（特别是较大数额支付时）下数字货币将具备更好的便携性   | 数字货币 |
| 防伪 | 两者各有千秋，但数字货币整体上会略胜一筹。纸币依靠的是各种设计（纸张、油墨、暗纹、夹层等）上的精巧，数字货币依靠的则是密码学上的保障。事实上，纸币的伪造时有发生，但数字货币的伪造目前还无法实现      | 数字货币 |
| 辨伪 | 纸币即使依托验钞机等专用设备仍会有误判情况，数字货币依靠密码学易于校验   | 数字货币 |
| 匿名 | 通常情况下，两者都能提供很好的匿名性。但都无法防御有意的追踪  | 持平   |
| 交易 | 对纸币来说，谁持有纸币谁就是合法拥有者，交易通过纸币自身的转移即可完成，无法复制。对数字货币来说则复杂得多，因为任何数字物品都是可以复制的，但数字形式也意味着转移成本会更低。总体上看，两者适用不同的情景 | 持平   |
| 资源 | 通常情况下，纸币的生产成本要远低于面额。数字货币消耗资源的计算则复杂得多。以比特币为例，最坏情况下可能需要消耗接近甚至超过面值的电能                                    | 纸币   |
| 发行 | 纸币的发行需要第三方机构的参与，数字货币则通过分布式算法来完成发行。在人类历史上，通胀和通缩往往是不合理地发行货币造成的，而数字货币尚缺乏大规模验证，还有待观察                      | 持平   |
| 管理 | 纸币发行和回收往往通过统一机构，易于监管和审计；目前出现的数字货币在这方面还缺乏足够支持和验证   | 纸币   |