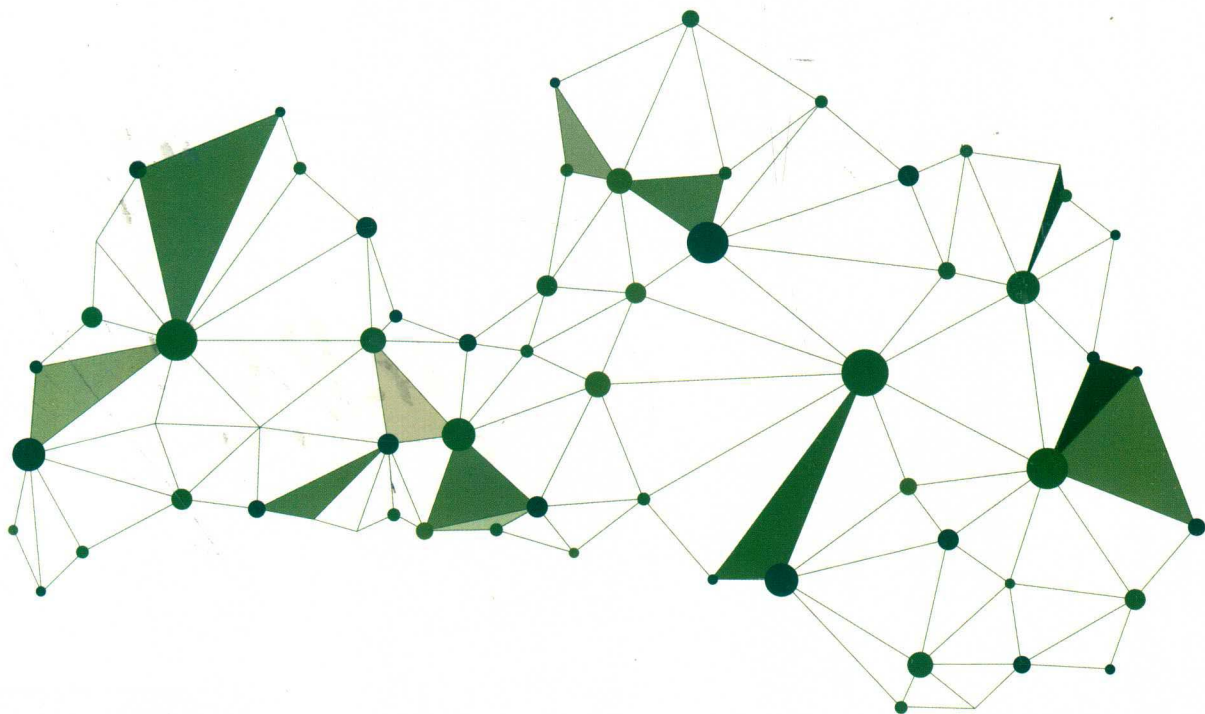


THE SOURCE CODE ANALYSIS  
OF HYPERLEDGER FABRIC

# Hyperledger Fabric

# 源代码分析与深入解读

蔡亮 梁秀波 宣章炯 © 著



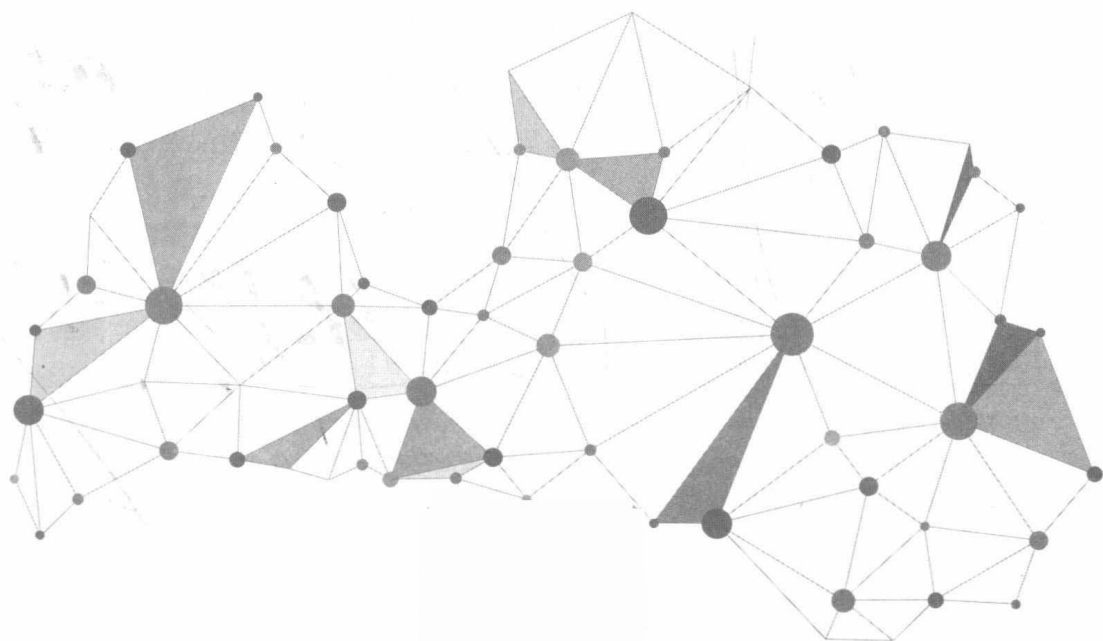
机械工业出版社  
China Machine Press

区块链  
技术丛书

THE SOURCE CODE ANALYSIS  
OF HYPERLEDGER FABRIC

# Hyperledger Fabric 源代码分析与深入解读

蔡亮 梁秀波 宣章炯 ◎著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

Hyperledger Fabric 源代码分析与深入解读 / 蔡亮, 梁秀波, 宣章炯著. —北京: 机械工业出版社, 2018.9

(区块链技术丛书)

ISBN 978-7-111-60870-7

I. H… II. ①蔡… ②梁… ③宣… 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 (2018) 第 210324 号

## Hyperledger Fabric 源代码分析与深入解读

---

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: 张锡鹏

责任校对: 李秋荣

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2018 年 9 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 25

书 号: ISBN 978-7-111-60870-7

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

## 以企业级联盟区块链技术引领我国新一代信息技术的突破式发展

区块链技术及其理念被认为是近年来最具颠覆性和革命性的创新之一，近年来已在金融、贸易、物流、征信、公益、物联网、共享经济等诸多领域崭露头角，被世界各国高度重视。美国、俄罗斯、英国、德国等国相继把区块链上升为国家战略。未来，区块链技术及其理念将对社会生活的各个不同领域产生重要影响，凡是涉及数据共享、信息可信和价值传递的场景都离不开区块链技术的应用。可以说，区块链技术及其理念作为信息共享和信任协作的基础，将重构人类社会未来的生产关系，通过与人工智能、云计算、大数据和物联网等新技术新理念的结合，将极大地提高社会生产力。

区块链将成为未来数字化社会的底层基础设施，从底层技术层面上为各行各业的持续稳定发展提供信任支撑，其价值主要体现在这些方面：（1）减少中间环节，实现实时清算结算，通过智能合约提高业务的自动化程度，从而为各行各业带来业务效率的提升；（2）实现数据的共享与标准化对接，基于数字资产的多行业流通实现价值网络的自繁殖，从而降低业务的拓展成本；（3）基于历史数据不可篡改、全流程可追溯和智能合约的规则约束可增强监管机构的监管能力；（4）该技术的多方可信合作和公平激励机制将成为未来分布式商业的发展基石，为各行业创造更多的合作机会。

在区块链技术快速发展的近十年中，产生了三次重要的技术演进：（1）以比特币为代表的加密数字货币应用将区块链技术引入了人们的视野；（2）以以太坊为代表的可编程区块链平台为区块链技术应用于各个领域提供了可能；（3）以 Hyperledger Fabric 和 Hyperchain 为代表的企业级区块链平台使得区块链技术能够真正应用于商业环境。本书以 Hyperledger Fabric 为讲解对象，对其源代码进行了细致分析和深入解读，以期帮助读者更好地理解和使用企业级的区块链底层平台技术。

Hyperledger Fabric 是 Linux 基金会引导的开源联盟区块链项目，其核心目标是建立开放的、标准化的、企业级的、支持商业应用的分布式账本框架与基础代码，具有高效共识、智能合约、多级加密、权限控制、隐私保护等特性。相较于比特币和以太坊，Hyperledger

Fabric 的核心优势有：（1）实现模块化和可插拔，更能适应复杂的商业应用要求；（2）多通道特性保障数据安全性，不同通道之间实现数据隔离；（3）实现证书的强化管理，提供单独的 Fabric CA 项目；（4）具有较好的可扩展性，解耦了原子排序与其他复杂处理环节；（5）可根据负载进行灵活部署，将交易处理节点逻辑简化为背书节点和确认节点。

本书的撰写团队来自于浙江大学区块链研究中心和杭州趣链科技有限公司，具有丰富的区块链底层核心技术研发和上层商业应用开发的经验。由杭州趣链科技有限公司研发的国产自主可控的联盟区块链平台 Hyperchain 目前已在中国工商银行、中国农业银行、中国建设银行、中国银联、Google、美国道富银行、葡萄牙商业银行等数十家国内外大型公司和机构得到了落地应用，在业内产生了强烈反响。

本书对 Hyperledger Fabric 架构及其设计模式进行分析，深入解读其源代码，并结合 Fabric-sample 项目介绍相关实践技术。相信可以给区块链技术爱好者和相关行业从业人员提供参考，具有很好的理论和实践价值。

陈纯

中国工程院院士，浙江大学计算机科学与技术学院教授  
曾任浙江大学软件学院院长和浙江大学计算机软件研究所所长

## 去除炒作泡沫，回归技术本质

区块链的多中心、自动化和可信任的理念与特性使得它可成为未来信息化社会信息共享、业务协作、价值传输的底层技术支撑。近年来，随着区块链技术的飞速发展和区块链理念的探索应用，以金融、证券、保险、物联网、跨境贸易、能源等为代表的很多行业都开始应用区块链。近几年新注册成立的、业务与区块链有关的公司达数千家，区块链专利申请量由两年前的几百件猛增至五千余件，社会对区块链技术人才的需求非常大，浙江大学等知名高校率先开设了区块链课程，区块链技术和理念深入到社会生活的方方面面已成为大势所趋。随着人们对区块链技术和理念认识的深入，其去除泡沫、回归本质的趋势非常明显，人们讨论的热点逐渐从如何炒币赚钱转向了如何应用区块链来推动各领域的产业升级与变革，区块链的底层核心技术和上层应用探索成为了产业发展的重点。

区块链本质上是提供信任保障的技术，可能发展为未来互联网最重要的基础设施之一。若能构建以区块链技术为基础的价值传输网络，将深刻地改变未来的信息社会形态。以区块链为底层技术支撑的加密数字货币应用给传统金融体系带来了一定的挑战，各国政府和央行都对加密数字货币应用保持了密切的关注。在区块链技术去货币化的探索阶段，业界普遍认识到了该技术所蕴含的巨大价值，通过在区块链上构建可重用、模块化、自动化的智能合约，许多创新的应用场景不断涌现，例如征信管理、跨境贸易、资源共享和版权共享等。以 Hyperledger Fabric 和 Hyperchain 为代表的企业级区块链平台克服了传统区块链技术资源消耗严重、交易性能低下、缺乏隐私保护等问题，提供了图灵完备的智能合约支持，使得开发者可以更便捷地开发和部署各领域的区块链商业应用，大大促进了基于联盟链的区块链商业形态的蓬勃发展。

Hyperledger 是 Linux 基金会于 2015 年发起的推进区块链数字技术和交易验证的开源项目，目标是使得区块链技术在加密数字货币之外也可广泛应用于其他场景。项目成立之初，就有 IBM、摩根大通、思科、Intel 等科技和金融巨头加入。Fabric 是 Hyperledger 最重要的子项目之一，它通过支持插件组件的模块化架构，实现了完备的权限管理、创新

的一致性算法等相关工作，对区块链技术的发展产生了重要影响。对于国内众多想利用 Hyperledger Fabric 开发部署区块链应用的开发者而言，相关的中文专业资料过于稀少和浅显。本书对 Hyperledger Fabric 开发环境的搭建与部署进行了详细地介绍，并对其技术架构和源码进行了深入地分析，为相关开发人员提供了及时的技术参考。

本书的作者团队来自浙江大学区块链研究中心和杭州趣链科技有限公司，在区块链前沿技术研究和产业化应用方面积累丰富。浙江大学区块链研究中心是专注于区块链领域的浙江大学校级研究机构，其目标是攻克区块链的底层技术难关，有效推动我国自主、安全、可控区块链技术的快速发展和产业化推广。杭州趣链科技有限公司是近年来成长极为迅速的区块链平台和应用解决方案提供商，其研发的区块链底层平台 Hyperchian 已在众多大型商业公司和机构的生产环境中投入使用。深厚的理论功底和扎实的实践经验使得本书在理论分析和实践应用方面都颇具特色。相信本书可以在帮助读者深刻理解 Hyperledger Fabric 的技术内涵的基础上，有效提升自己在区块链方面的开发能力。

Julian Gordon

Hyperledger 亚太区副总裁

## Preface 前言

区块链技术自诞生以来就受到社会各界的广泛关注，甚至被认为是继互联网之后最具颠覆性意义的技术革命。如果说最初的互联网实现了信息在全球范围内的高速传播与共享，那么区块链则致力于构建一个可用于价值交换的下一代可信互联网；如果说人工智能等技术改变的是生产力，那么区块链技术改变的是生产关系。据专家预测，区块链在不远的将来会引发政治、经济、文化、科技、军事、教育、医疗、能源等诸多领域的颠覆性革命。

区块链的本质是一个去中心化的分布式账本系统，是一种涵盖了分布式数据存储、点对点传输、共识机制、加密算法等一系列计算机技术的集成创新技术。区块链采用分布式数据库来存储数据，通过自信任的共识机制实现全网数据的一致性，将对称与非对称加密技术相结合以保证数据安全和不可篡改，采用智能合约技术规范化处理数据，最终实现在无需第三方中介机构的情况下进行人与人之间的价值交换，进而解决信任问题。

众所周知，区块链技术源于比特币，最初是作为比特币的底层技术而存在的。因此，区块链的发展也与比特币息息相关。2008年，一个化名为中本聪的人在一篇名为《比特币：一个点对点的电子现金系统》的论文中详细描述了一种去中心化的分布式账本系统，并把这个系统所使用的数字加密货币称为比特币。比特币的诞生开启了区块链1.0时代，这一时期各种基于区块链技术的加密数字货币如雨后春笋般涌现出来，如以太坊、莱特币、瑞波币等。然而此时的区块链技术仅仅用于交易记账，应用范围非常有限。

随着各种加密数字货币的广泛流行，一些有识之士逐渐意识到区块链技术的巨大应用潜力。他们将区块链技术从比特币系统中分离出来，并通过添加智能合约技术构建出一个更通用的去中心化应用开发平台，其中最具代表性的就是以太坊。以太坊的出现标志着区块链迈入了2.0时代，这一时期区块链技术的应用范围得到了极大的扩展，同时涉及股权众筹、证券交易、贷款抵押等诸多金融领域。

然而，以太坊平台作为一个基于公有链的区块链平台，依然存在一些不足之处亟待改进，比如存在共识效率低下、隐私保护缺乏、大规模存储困难和信息难以监管等问题。为了应对大规模商业应用的业务需求，面向企业级应用的联盟区块链技术应运而生。其中的

典型代表就是 Hyperledger Fabric 开源项目，它通过模块化和可插拔设计、权限控制、多链和多通道等技术为区块链技术在各领域的应用打开了广阔的空间。

本书是一本介绍 Hyperledger Fabric 架构、源代码及其底层实现的专业书籍。从编程语言、源码分析、网络节点、加密算法、智能合约、架构设计等多个角度深入解读 Fabric 项目，并通过对项目实例的分析介绍，使读者能够理论结合实践，增强实际动手能力，更好地理解 Fabric 的实现原理。本书是一本实际操作性极强的 Hyperledger Fabric 的专业书籍，通过阅读本书，能够帮助读者零基础快速入门 Fabric。

## 本书结构

本书共分为 14 章。

第 1 章主要介绍如何从零开始，通过准备工作完成多种平台上 Hyperledger Fabric 应用环境的配置。首先，讲解不同系统中 Go 语言的安装与环境配置，并简单介绍了 Go 语言。之后，讲解不同系统中 Docker 的安装和简单使用方法。最后，介绍了 Hyperledger 社区与社区中一些常用软件，方便读者在完成环境配置之余，更好地了解 Hyperledger Fabric 的相关信息。

第 2 章主要讲解 Fabric 的架构并进行分析。首先介绍了 Fabric 整体架构，包括系统架构、交易背书及策略、证实账本和节点账本的检查。然后介绍了 Fabric 交易流程。接着介绍了 Fabric 整体项目结构，并详细阐述了 Fabric 源码中相关缩写的含义。

第 3 章主要介绍了 Fabric 中的四个模块：Logging 日志模块、Error 错误处理框架、Config 配置模块、grpc 服务。理解日志模块有助于理解源码；理解错误处理框架能使读者学会独立处理代码中常见错误的方法；理解配置模块对具象化程序很有帮助；理解 grpc 有助于读者理解如何实现客户端与服务器端的远程调用，从而为进一步理解源码打下基础。

第 4 章主要介绍了 Fabric 中 peer 节点的设计与实现。peer 节点作为 Fabric 中处理交易、存储区块的角色，承担了重要的作用。本章对 CommandLine 进行了解析，介绍了 Admin 及 Endorser 服务的实现。同时具体分析了 Committer 机制及其作用。

第 5 章主要介绍了 order 的设计与实现。orderer 为 Fabric 的排序节点，其为所有的客户端提供统一的交易排序及打包服务，并将区块分发到所有的 leader 节点。本章的内容包括 Orderer 内部机制窥探、kafka 排序服务机制讲解以及 orderer 在 Fabric 中的交互流程。

第 6 章主要介绍了 chaincode 的设计与实现。chaincode 是 Fabric 实现智能合约的方式，利用容器技术将智能合约放置在容器中运行，同时进行调用。本章的内容包括 chaincode 生命周期管理，原理浅析，数据结构分析，SystemChaincode 的讲解，CSCC、ESCC、LSCC、QSCC、VSCC 等概念的分析，SystemChaincode 的注册和实例化，ApplicationChaincode 的部署及实例化，以及 chaincode 操作步骤。

第7章主要介绍了MSP成员服务提供者及Fabric的交易网络。通过Membership Service Providers (MSP) 主题帮助读者更好地理解加密操作、签名、校验、认证等在Fabric中的应用。

第8章主要介绍了Gossip协议。Gossip协议是分布式系统用于保证分布式一致性的一种方式，也是Fabric中传递各种信息及区块信息的手段。本章后对Gossip协议原理进行了解析，介绍了Gossip的服务组件、服务初始化、消息广播、channel通道的设计与实现以及事件机制。

第9章主要介绍了Fabric中BCCSP加密服务提供者的设计与实现。BCCSP为加密服务提供者，为Fabric网络信息传输提供了密码学的保障。本章介绍了密码学相关知识及BCCSP概要，并对BCCSP源码进行了剖析。

第10章主要讲解了Fabric CA的架构设计。Fabric CA是Hyperledger Fabric行使证书机构的功能。本章介绍了Fabric CA用户指南，以及如何使用Fabric-CA-Server、Fabric-CA-Client等指令和HSM（硬件安全模块）。

第11章主要介绍了在Fabric中账本机制的设计与实现。本章的内容包括Ledger架构概述，Ledger中的Block-Storage、VersionedDB、HistoryDB等概念及其作用。

第12章主要通过具体的案例，加深读者对chaincode智能合约的理解，让读者了解Fabric中智能合约的编写方法。

第13章通过介绍一个完整的项目实例的实现来帮助读者更好地理解Fabric，让读者做到理论结合实际，进而提升实战经验。

第14章通过对官网案例进行部署，以体现源码和实际操作的呼应关系。

## 勘误与后续技术支持

由于作者的写作时间和水平有限，本书难免会存在一些纰漏和错误，欢迎广大读者批评指正。勘误请发送至作者邮箱：liangxiubo@hyperchain.cn。对于读者发现的问题，我们将在本书后续印次和版本中加以改正。

为了进一步降低区块链技术使用门槛，让更多的区块链开发者、爱好者以及正在尝试接入区块链技术的企业能够快捷地开发区块链应用，趣链科技于2017年9月14日正式上线了基于联盟链的“开发者平台”。基于该平台，用户可以更方便地创建、发布和使用多中心化的应用程序。通过平台提供的在线智能合约编辑器，用户可便捷、准确地编写智能合约程序；通过平台提供的区块链浏览器，用户可方便地获取链上区块信息、区块链节点状态、节点维护方信息等。欢迎广大区块链相关从业人员访问体验，开发者平台网址为：<https://dev.hyperchain.cn/>。

如需获得更多关于区块链技术的最新技术动态和趣链科技的技术支持，可关注微信公众号：hyperchain。

## 致谢

在此向所有给我们提供指导、支持和鼓励的朋友表示衷心的感谢。

感谢浙江大学计算机科学与技术学院和软件学院为我们提供的良好条件和各种便利，感谢陈纯院士、杨小虎研究员一直以来的关怀和支持。

感谢杭州趣链科技有限公司全体人员的大力支持，特别感谢李伟博士、李启雷博士、邱炜伟博士、尹可挺博士为本书成稿给予的鼎力支持，感谢曹靖、易洋溢、梁健、孟德佳、秦启睿、余清波、何昊等对书稿材料汇编所做的突出贡献，感谢余高成、周健、施泰龙等对书稿校阅所付出的时间和汗水。

感谢机械工业出版社华章公司的编辑们，他们不辞辛苦进行的仔细严谨的审阅和校对工作是本书顺利出版的有力保障。

蔡亮 梁秀波 宣章炯

## Contents 目 录

|                                         |    |  |
|-----------------------------------------|----|--|
| 序一                                      |    |  |
| 序二                                      |    |  |
| 前言                                      |    |  |
| 第 1 章 准备工作                              | 1  |  |
| 1.1 Go 语言环境配置                           | 1  |  |
| 1.1.1 Go 语言简介                           | 1  |  |
| 1.1.2 Go 安装                             | 2  |  |
| 1.1.3 Go 标准包安装                          | 4  |  |
| 1.1.4 第三方工具安装                           | 6  |  |
| 1.1.5 Go 环境配置                           | 7  |  |
| 1.1.6 代码目录结构规划                          | 8  |  |
| 1.1.7 编译应用                              | 9  |  |
| 1.1.8 获取远程包                             | 10 |  |
| 1.1.9 程序的整体结构                           | 11 |  |
| 1.2 安装 Docker                           | 11 |  |
| 1.2.1 macOS                             | 11 |  |
| 1.2.2 Ubuntu                            | 12 |  |
| 1.2.3 Docker 的简易使用                      | 13 |  |
| 1.3 Hyperledger 社区介绍                    | 14 |  |
| 第 2 章 架构分析                              | 18 |  |
| 2.1 Fabric 整体架构                         | 18 |  |
| 2.1.1 概述                                | 18 |  |
| 2.1.2 系统架构                              | 19 |  |
| 2.1.3 交易背书的基本工作流程                       | 24 |  |
| 2.1.4 背书策略                              | 27 |  |
| 2.1.5 证实账本和节点账本检查                       | 28 |  |
| 2.2 Fabric 交易流程                         | 30 |  |
| 2.3 Fabric 整体项目结构介绍                     | 33 |  |
| 2.3.1 Fabric 项目结构                       | 33 |  |
| 2.3.2 Fabric 源码中相关缩写的含义                 | 34 |  |
| 第 3 章 源码分析                              | 37 |  |
| 3.1 Logging 日志模块浅析                      | 37 |  |
| 3.1.1 go-logging 简介                     | 37 |  |
| 3.1.2 flogging                          | 38 |  |
| 3.1.3 init 函数、MustGetLogger 函数<br>与其他函数 | 38 |  |
| 3.2 Error 错误机制设计                        | 39 |  |
| 3.2.1 总体概览                              | 39 |  |
| 3.2.2 使用说明                              | 40 |  |
| 3.2.3 显示错误消息                            | 40 |  |
| 3.2.4 错误处理的一般准则                         | 41 |  |
| 3.3 Config 配置模块的设计                      | 41 |  |

|              |                        |           |              |                              |           |
|--------------|------------------------|-----------|--------------|------------------------------|-----------|
| 3.3.1        | viper 简介               | 41        | 5.2          | kafka 排序服务机制讲解               | 79        |
| 3.3.2        | 安全文件配置                 | 44        | 5.3          | orderer 在 Fabric 中的交互流程      | 82        |
| 3.3.3        | 命令选项配置                 | 44        | 5.3.1        | 建立连接                         | 82        |
| 3.3.4        | 环境变量配置                 | 44        | 5.3.2        | Broadcast                    | 83        |
| 3.4          | grpc 服务                | 45        | 5.3.3        | orderer                      | 83        |
| 3.4.1        | grpc 用法的 Demo          | 45        | 5.3.4        | Deliver                      | 86        |
| 3.4.2        | Fabric 中的 grpc 服务接口和实例 | 46        |              |                              |           |
| <b>第 4 章</b> | <b>peer 的设计与实现</b>     | <b>53</b> | <b>第 6 章</b> | <b>chaincode 的设计与实现</b>      | <b>89</b> |
| 4.1          | CommandLine 解析         | 53        | 6.1          | chaincode 生命周期管理             | 89        |
| 4.1.1        | peer 目录结构              | 53        | 6.1.1        | 打包                           | 89        |
| 4.1.2        | 第三方包                   | 54        | 6.1.2        | 安装 chaincode                 | 91        |
| 4.1.3        | peer 命令结构解析            | 55        | 6.1.3        | 实例化 chaincode                | 91        |
| 4.1.4        | 以 node 为例进行子命令结构解析     | 55        | 6.1.4        | 升级 chaincode                 | 92        |
| 4.1.5        | peer 命令结构              | 55        | 6.1.5        | 停止与启动                        | 93        |
| 4.2          | Admin 及 Endorser 服务的实现 | 56        | 6.1.6        | CLI                          | 93        |
| 4.2.1        | Admin                  | 56        | 6.2          | chaincode 原理浅析               | 94        |
| 4.2.2        | Endorser               | 58        | 6.2.1        | 什么是 chaincode                | 94        |
| 4.2.3        | 频道中的策略检查器              | 64        | 6.2.2        | Chaincode Support 服务         | 95        |
| 4.3          | Committer 的机制          | 66        | 6.2.3        | FSM                          | 95        |
| 4.3.1        | committer.go 分析        | 66        | 6.2.4        | Register                     | 96        |
| 4.3.2        | committer_impl.go 分析   | 67        | 6.2.5        | Handler                      | 97        |
| 4.3.3        | validator.go 分析        | 70        | 6.2.6        | processStream                | 97        |
| 4.3.4        | vscv_validator.go 分析   | 71        | 6.2.7        | HandleMessage                | 97        |
|              |                        |           | 6.2.8        | serialSend 或 serialSendAsync | 99        |
| <b>第 5 章</b> | <b>order 的设计与实现</b>    | <b>73</b> | 6.2.9        | 系统 chaincode                 | 99        |
| 5.1          | orderer 内部机制窥探         | 73        | 6.3          | chaincode 数据结构分析             | 100       |
| 5.1.1        | kingpin                | 73        | 6.3.1        | chaincode 元数据                | 100       |
| 5.1.2        | 模块                     | 74        | 6.3.2        | chaincode 的元工具               | 102       |
| 5.1.3        | 配置                     | 74        | 6.4          | SystemChaincode 讲解           | 103       |
| 5.1.4        | 模块的初始化                 | 75        | 6.4.1        | SystemChaincode              | 104       |
|              |                        |           | 6.4.2        | 预定义和注册                       | 104       |

|        |                          |     |              |                                         |     |
|--------|--------------------------|-----|--------------|-----------------------------------------|-----|
| 6.5    | CSCC 分析                  | 106 | 6.11.6       | Execute                                 | 127 |
| 6.5.1  | 结构体                      | 106 | 6.11.7       | 一路返回                                    | 128 |
| 6.5.2  | 函数                       | 106 | 6.11.8       | 安装后的状态                                  | 129 |
| 6.6    | ESCC 分析                  | 108 | 6.12         | ApplicationChaincode 的实例化               | 129 |
| 6.6.1  | 结构体                      | 108 | 6.12.1       | 概述                                      | 129 |
| 6.6.2  | Init 函数                  | 108 | 6.12.2       | 起点                                      | 130 |
| 6.7    | LSCC 分析                  | 109 | 6.12.3       | 部署                                      | 130 |
| 6.7.1  | 结构体和接口                   | 110 | 6.12.4       | 广播                                      | 139 |
| 6.7.2  | 函数操作                     | 110 | 6.12.5       | 部署后的状态                                  | 139 |
| 6.7.3  | 安装、部署和升级                 | 111 | 6.13         | chaincode 操作步骤                          | 140 |
| 6.7.4  | chaincode stub 接口实现      | 112 | 6.13.1       | 选择一个代码存放位置                              | 140 |
| 6.8    | QSCC 分析                  | 113 | 6.13.2       | 内务处理                                    | 140 |
| 6.8.1  | 结构体                      | 113 | 6.13.3       | 初始化 chaincode                           | 140 |
| 6.8.2  | 函数操作                     | 114 | 6.13.4       | 调用 chaincode                            | 142 |
| 6.8.3  | 路由规则                     | 114 | 6.13.5       | 实现 chaincode 应用                         | 143 |
| 6.9    | VSCC 分析                  | 115 | 6.13.6       | 整合全部代码                                  | 143 |
| 6.9.1  | 结构体                      | 115 | 6.13.7       | 编译 chaincode                            | 145 |
| 6.9.2  | 函数                       | 115 | 6.13.8       | 在开发者模式下测试                               | 145 |
| 6.10   | SystemChaincode 的注册和实例化  | 116 | 6.13.9       | 安装 Hyperledger Fabric 样例                | 145 |
| 6.10.1 | 概述                       | 116 | 6.13.10      | 下载 Docker 镜像                            | 146 |
| 6.10.2 | 安装                       | 117 | 6.13.11      | 1 号终端                                   | 146 |
| 6.10.3 | 部署                       | 117 | 6.13.12      | 2 号终端                                   | 146 |
| 6.10.4 | Launch                   | 118 | 6.13.13      | 3 号终端                                   | 147 |
| 6.10.5 | Execute                  | 123 | 6.13.14      | 测试新的 chaincode                          | 147 |
| 6.10.6 | 部署后状态                    | 124 |              |                                         |     |
| 6.11   | ApplicationChaincode 的部署 | 124 | <b>第 7 章</b> | <b>MSP 成员服务提供者</b>                      | 148 |
| 6.11.1 | 概述                       | 125 | 7.1          | MSP 的设计思路                               | 148 |
| 6.11.2 | 生成签名申请包                  | 125 | 7.1.1        | MSP 配置                                  | 149 |
| 6.11.3 | 处理安装申请                   | 125 | 7.1.2        | 如何生成 MSP 证书和它们的<br>签名匙                  | 150 |
| 6.11.4 | 执行申请                     | 126 | 7.1.3        | MSP setup on the peer &<br>orderer side | 150 |
| 6.11.5 | Launch                   | 127 |              |                                         |     |

|                              |                                |     |                                  |                     |     |
|------------------------------|--------------------------------|-----|----------------------------------|---------------------|-----|
| 7.1.4                        | Channel MSP setup              | 150 | 8.6.3                            | Go SDK 中 Event 相关实现 | 210 |
| 7.1.5                        | 最佳实践                           | 151 |                                  |                     |     |
| 7.2                          | MSP 实现剖析                       | 153 | <b>第 9 章 BCCSP 加密服务提供者的设计与实现</b> | 212                 |     |
| 7.2.1                        | 目录结构                           | 153 | 9.1                              | 密码学相关知识介绍           | 212 |
| 7.2.2                        | MSP 配置                         | 154 | 9.1.1                            | 安全基础                | 212 |
| <b>第 8 章 Gossip 节点间的流言蜚语</b> | 162                            |     | 9.1.2                            | 加密基础                | 213 |
| 8.1                          | Gossip 协议原理解析                  | 162 | 9.1.3                            | 哈希函数                | 214 |
| 8.1.1                        | Gossip 协议 (Gossip protocol)    | 162 | 9.1.4                            | 共享密钥加密              | 214 |
| 8.1.2                        | Gossip 消息传输 (Gossip messaging) | 163 | 9.1.5                            | 公钥加密                | 215 |
| 8.2                          | Gossip 之服务组件                   | 163 | 9.1.6                            | 混合加密                | 216 |
| 8.2.1                        | protos/gossip 分析               | 163 | 9.1.7                            | 消息验证码               | 216 |
| 8.2.2                        | Gossip 服务组件                    | 169 | 9.1.8                            | 数字签名                | 218 |
| 8.2.3                        | gossip 消息发送方式详解                | 176 | 9.1.9                            | 数字证书                | 219 |
| 8.3                          | Gossip 之服务初始化                  | 178 | 9.2                              | BCCSP 概要            | 220 |
| 8.3.1                        | gossipSvc 组件                   | 179 | 9.2.1                            | BCCSP 简介            | 220 |
| 8.3.2                        | chains 组件                      | 185 | 9.2.2                            | 陷阱函数                | 222 |
| 8.3.3                        | leaderElection 组件              | 187 | 9.2.3                            | 为什么要使用 ECDSA        | 223 |
| 8.3.4                        | gossip 服务的停止                   | 193 | 9.2.4                            | 生成签名                | 223 |
| 8.4                          | Gossip 之消息广播                   | 194 | 9.2.5                            | 验证签名                | 224 |
| 8.4.1                        | gossip 服务消息的散播过程               | 194 | 9.3                              | BCCSP 源码剖析          | 224 |
| 8.4.2                        | 消息从何而来                         | 194 | 9.3.1                            | BCCSP 服务结构          | 224 |
| 8.4.3                        | 消息如何散播                         | 196 | 9.3.2                            | BCCSP 中的接口和选项       | 225 |
| 8.4.4                        | 消息去往何方                         | 200 | 9.3.3                            | SW 实现方式             | 227 |
| 8.5                          | channel 通道的设计与实现               | 201 | 9.3.4                            | pkes11 实现方式         | 230 |
| 8.5.1                        | 概述                             | 201 | <b>第 10 章 Fabric CA 架构设计与讲解</b>  | 233                 |     |
| 8.5.2                        | 配置文件                           | 202 | 10.1                             | Fabric CA 用户指南      | 233 |
| 8.5.3                        | 命令                             | 203 | 10.2                             | Fabric-CA-Server    | 240 |
| 8.6                          | 事件机制                           | 208 | 10.2.1                           | 初始化服务端              | 241 |
| 8.6.1                        | Fabric 中 Event 相关实现            | 208 | 10.2.2                           | 算法和密钥长度             | 242 |
| 8.6.2                        | events/producer                | 209 |                                  |                     |     |

|         |                                 |     |        |                        |     |
|---------|---------------------------------|-----|--------|------------------------|-----|
| 10.2.3  | 启动服务端                           | 243 | 11.2   | Ledger 之 Block-Storage | 275 |
| 10.2.4  | 配置数据库                           | 243 | 11.2.1 | peer 节点中的 leveldb      | 276 |
| 10.2.5  | PostgreSQL                      | 243 | 11.2.2 | peer 节点中的账本            | 276 |
| 10.2.6  | PostgreSQL SSL 配置               | 244 | 11.2.3 | 创建                     | 276 |
| 10.2.7  | MySQL                           | 245 | 11.2.4 | 使用                     | 277 |
| 10.2.8  | MySQL SSL 配置                    | 245 | 11.2.5 | idStore                | 278 |
| 10.2.9  | 配置 LDAP                         | 246 | 11.2.6 | 存储账本 ID                | 278 |
| 10.2.10 | 构建一个集群                          | 249 | 11.2.7 | ConstructionFlag       | 278 |
| 10.2.11 | 构建多个 CA                         | 249 | 11.2.8 | 账本恢复                   | 279 |
| 10.2.12 | 登录一个中间 CA                       | 250 | 11.2.9 | BlockStore             | 280 |
| 10.2.13 | 升级服务端                           | 251 | 11.3   | Ledger 之 VersionedDB   | 286 |
| 10.2.14 | 升级一个集群                          | 251 | 11.3.1 | peer 节点使用 VersionedDB  | 286 |
| 10.3    | fabric-ca-client                | 253 | 11.3.2 | 交易模拟器 / 交易查询器          | 288 |
| 10.3.1  | 登录启动用户                          | 253 | 11.3.3 | 重启恢复                   | 298 |
| 10.3.2  | 注册一个新身份                         | 253 | 11.4   | Ledger 之 HistoryDB     | 300 |
| 10.3.3  | 登录一个节点                          | 256 | 11.4.1 | 历史查询器                  | 300 |
| 10.3.4  | 从另一个 Fabric CA 服务器<br>获得 CA 证书链 | 257 | 11.4.2 | 使用                     | 301 |
| 10.3.5  | 重新登录一个身份                        | 257 | 第 12 章 | chaincode 智能合约案例<br>分析 | 303 |
| 10.3.6  | 撤销一个证书或身份                       | 257 | 12.1   | encc_example           | 303 |
| 10.3.7  | 生成一个 CRL                        | 259 | 12.1.1 | chaincode 代码分析         | 303 |
| 10.3.8  | 启用 TLS                          | 259 | 12.1.2 | 使用 EncCC               | 307 |
| 10.3.9  | 基于属性的访问控制                       | 260 | 12.2   | eventsender            | 308 |
| 10.3.10 | 动态更新服务器配置                       | 261 | 12.3   | example01              | 310 |
| 10.3.11 | 联系特定的 CA 实例                     | 265 | 12.4   | example02              | 311 |
| 10.4    | HSM                             | 265 | 12.5   | example03              | 314 |
| 第 11 章  | 账本机制的设计与实现                      | 267 | 12.6   | example04              | 315 |
| 11.1    | Ledger 架构概述                     | 267 | 12.7   | example05              | 317 |
| 11.1.1  | 总览                              | 267 | 12.8   | invokereturnsvalue     | 319 |
| 11.1.2  | ledger 部分摘要                     | 268 | 12.9   | map                    | 320 |
|         |                                 |     | 12.10  | marbles02              | 324 |

|                                            |                                          |            |                          |                                              |            |
|--------------------------------------------|------------------------------------------|------------|--------------------------|----------------------------------------------|------------|
| 12.11                                      | passthru .....                           | 332        | 13.4.5                   | 更新账本 .....                                   | 358        |
| 12.12                                      | sleeper .....                            | 332        | 13.5                     | Balance transfer .....                       | 360        |
| <b>第 13 章 Fabric-samples 项目分析与实践 .....</b> |                                          | <b>334</b> | 13.5.1                   | 预置环境 .....                                   | 360        |
| 13.1                                       | Fabric-samples 项目结构 .....                | 334        | 13.5.2                   | 工件 .....                                     | 360        |
| 13.2                                       | First-network .....                      | 335        | 13.5.3                   | 运行示例程序 .....                                 | 360        |
| 13.2.1                                     | 安装预置环境 .....                             | 335        | 13.5.4                   | 示例——REST APIs 请求 .....                       | 361        |
| 13.2.2                                     | 想要现在运行吗? .....                           | 335        | 13.6                     | Hyperledger Fabric CA 示例 .....               | 365        |
| 13.2.3                                     | 生成网络神器 .....                             | 336        | 13.6.1                   | 运行这个示例 .....                                 | 366        |
| 13.2.4                                     | 启动网络 .....                               | 337        | 13.6.2                   | 了解这个例子 .....                                 | 366        |
| 13.2.5                                     | 关闭网络 .....                               | 339        | 13.7                     | 高性能网络 .....                                  | 367        |
| 13.2.6                                     | 加密生成器 .....                              | 339        | 13.7.1                   | 用例 .....                                     | 368        |
| 13.2.7                                     | 配置交易生成器 .....                            | 340        | 13.7.2                   | 如何使用 .....                                   | 368        |
| 13.2.8                                     | 运行工具 .....                               | 341        | <b>第 14 章 部署教程 .....</b> |                                              | <b>371</b> |
| 13.2.9                                     | 启动网络 .....                               | 342        | 14.1                     | 下载部署环境 .....                                 | 371        |
| 13.2.10                                    | 了解 Docker Compose 技术 .....               | 347        | 14.2                     | 编译 peer、orderer、configtxgen<br>等程序 .....     | 373        |
| 13.2.11                                    | 使用 CouchDB .....                         | 348        | 14.3                     | 部署 .....                                     | 374        |
| 13.2.12                                    | 关于数据持久化的提示 .....                         | 350        | 14.4                     | Crypto Generator .....                       | 374        |
| 13.2.13                                    | 故障排除 .....                               | 350        | 14.4.1                   | crypto-config.yaml .....                     | 375        |
| 13.3                                       | basic-network .....                      | 351        | 14.4.2                   | crypto-config 文件夹 .....                      | 375        |
| 13.4                                       | Fabcar .....                             | 353        | 14.5                     | Configuration Transaction<br>Generator ..... | 376        |
| 13.4.1                                     | 编写第一个应用 .....                            | 353        | 14.6                     | networkUp- 启动 Fabric 网络 .....                | 378        |
| 13.4.2                                     | 下载测试网络 (Getting a<br>Test Network) ..... | 354        | 14.7                     | 运行容器 + 区域链操作 .....                           | 378        |
| 13.4.3                                     | 应用程序如何与网络进行交互 .....                      | 355        | <b>附录 专业术语 .....</b>     |                                              | <b>381</b> |
| 13.4.4                                     | 查询账本 .....                               | 355        |                          |                                              |            |