Cambridge
Centre
for Alternative
Finance

UNIVERSITY OF
CAMBRIDGE
Judge Business School

独家中文翻译合作伙伴

浙江大学 互联网金融研究院
ACADEMY OF INTERNET FINANCE
ZHEJIANG UNIVERSITY

DISTRIBUTED LEDGER TECHNOLOGY SYSTEMS
A Conceptual Framework

分布式账本技术系统

一个概念框架

Michel Rauchs　Andrew Glidden　Brian Gordon　Gina Pieters
Martino Recanatini　François Rostand　Kathryn Vagneur　Bryan Zhang

更多资讯，敬请联络

浙江大学互联网金融研究院
Academy of Internet Finance，Zhejiang University
杭州市余杭塘路866号

浙大AIF(CIFI_ZJU)

Cambridge Centre for Alternative Finance
10 Trumpington Street
Cambridge CB2 1QA
United Kingdom
Email: ccaf@jbs.cam.ac.uk
Tel: +44(0)1223 339111

剑桥大学新兴金融研究中心（CCAF）是剑桥大学嘉治商学院下属的国际性跨学科学术研究中心，致力于传统金融以外的创新工具、渠道和系统的研究，包括众筹、网络借贷、创新型信贷和投资分析、创新支付系统、加密货币、分布式账本技术（如区块链）以及相关的制度和监管创新（如监管沙盒和监管科技）等方面的研究。

目　录

前言 …… **01**
序言 …… **02**
研究团队 …… **03**
声明 …… **08**
致谢 …… **08**
独家授权联合发布声明 …… **08**
摘要 …… **09**
第一章：导言 …… **11**
理论基础 …… 11
报告目标 …… 11
研究方法 …… 12
报告结构 …… 12
第二章：分布式账本技术系统—建立场景 …… **14**
2.1 文献综述 …… 14
2.1.1 定义 …… 14
2.1.2 现有框架 …… 15
2.1.3 前期研究的局限 …… 15
2.2 什么是分布式账本技术系统? …… 16
2.3 相关概念 …… 18
“账本”的概念 …… 18
“私钥”的概念 …… 21
2.4 参与者 …… 22
2.4.1 开发人员 …… 23
2.4.2 管理员 …… 23
2.4.3 网关 …… 23
2.4.4 使用者 …… 24
第三章：框架介绍 …… **25**
3.1 协议层 …… 26
3.2 网络层 …… 26
3.3 数据层 …… 27
参考 / 价值链接 …… 28
3.4 框架整合 …… 28
第四章：系统交互 …… **31**
4.1 系统界限之内 …… 31
4.1.1 层间依赖 …… 31
4.1.2 层次结构 …… 31

4.1.3 权衡：没有一种系统能适合所有模式 …… 32
4.1.4 关于“去中心化”的说明 …… 33
4.2 系统界限之外 …… 35
4.2.1 系统的视角 …… 35
4.2.2 外源和内源引用 …… 37
第五章：框架的深度分析 …… 41
5.1 协议层 …… 41
5.1.1 初始组件 …… 41
5.1.2 变动组件 …… 42
5.2 网络层 …… 45
5.2.1 通信组件 …… 45
5.2.2 交易处理组件 …… 47
5.2.3 验证组件 …… 49
5.3 数据层 …… 52
5.3.1 操作组件 …… 52
5.3.2 账本组件 …… 54
第六章：应用框架—案例研究 …… 56
6.1 比特币 …… 56
协议 …… 56
网络 …… 57
数据 …… 58
6.2 比较分析 …… 59
6.2.1 案例研究 …… 59
6.2.2 这些是分布式账本技术系统吗? …… 60
6.2.3 协议 …… 61
6.2.4 网络 …… 64
6.2.5 数据 …… 68
6.3 比较分布式账本技术系统案例研究中的差异 …… 70
6.3.1 总结框架结果 …… 70
6.3.2 参与方式差异 …… 71
6.3.3 探究当前分布式账本技术系统蓝图 …… 72
6.3.4 关键设计决策和启示 …… 73
第七章：结论 …… 74
7.1 总结 …… 74
7.2 研究贡献 …… 75
7.3 研究不足及未来方向展望 …… 76
附录A：分布式账本系统详解 …… 77
附录B：案例比较 …… 78
附录C：术语表 …… 81

前　言

加密货币、区块链、分布式账本这样的概念已逐渐进入我们的日常生活，并逐渐成为新闻媒体关注的焦点，并由此引起了学术研究人员、行业参与者和政策制定者的激烈讨论。然而由于很多概念目前还没有严格定义的专业术语体系或被普遍接受的分类体系，导致很多词语经常被错误的理解和使用。

如果没有一个系统、完整的认识方法，人们的注意力和分析很容易被事物的表象所吸引，而看不到整体的情况，从而 “见树不见林”，并更容易受到偏见、误解、夸张或冲突观点的影响。

分布式账本是一个具有核心层次、组件、流程以及与其他可适用系统交互的功能系统。因此，我们需要对这个系统进行更加全面、深入的研究并做出概念界定和考察。我们采用“系统性视角”开始这段研究，希望公众不止见到“树”和“林”，还可以对这个复杂的 “分布式账本技术生态系统”有更细致的理解。

在本中心的全球加密货币和区块链基准研究基础上，我们意识到此项任务的挑战性以及自身的局限性，专门组建了由不同背景的研究人员和参与者构成的研究团队。为了对分布式账本技术概念进行界定，并在定义及分类上达成“共识”，我们的研究过程是开放、协作和自我批判的。正如我们所看到的，随着当下分布式账本技术领域自身快速地发展，这项研究将会成为一个开端，并催生和吸引更多的关注和讨论。

在这项研究中，我们采用“系统视角”和一套分析框架将分布式账本技术系统“解构”然后“重建”。这套分析框架将所有的分布式账本技术系统分为三个层次：协议、网络和数据，并阐明这些核心层次是如何通过流程相互作用的，以及它们在系统内的条件从属关系和层次结构。本研究论证了如何改变这些层次及其组件的“配置”，使得“分布式账本技术系统”（以及扩展它们记录的内容和资产）在发挥的功能和运作方式上产生巨大的变化。它同时说明了分布式账本技术系统在更广泛的生态系统中是如何相互作用，“中心化”和“去中心化”应怎样被理解为一个范围内的变化而不是相互对立的，以及区分“本地”和“异地”记录存储的必要性。

我们非常感谢研究团队所有成员的贡献，希望可以借此机会加深对分布式账本技术系统的共同理解。

非常感谢和以贲圣林教授带队的浙江大学互联网金融研究院（AIF)合作，出版本报告的中文版。谢谢浙大团队的大力支持和精心翻译。我们期待可以和AIF携手前行，共同为金融创新的研究学习尽一份力！

张征

执行主任、联合创始人

剑桥大学新兴金融研究中心

序 言

还在路上

社会大众对分布式账本技术、区块链、数字资产等这些新鲜名字的认知度呈现梯度和爆发式发展！比特币的火爆普及了背后的区块链概念，随后分布式账本技术这个拗口的名词开始浮出水面，走进公众视野，引起大家的关注。但是，从技术信徒的“链圈”到资本躁动的“币圈”，从业界到学术界到政府部门，从专业人士到社会公众，大家对基本概念的定义和理解差异之大可谓南辕北辙。有一知半解的，有不求甚解的，加之一些“初心不良”势力出于一己私利而故意混淆视听，形势大有失控的风险！

正是在这样一个背景之下，浙江大学互联网金融研究院（AIF）很高兴和剑桥大学新兴金融研究中心（CCAF）一起发布“分布式账本技术系统：一个概念框架”报告的中文版。报告对目前分布式账本技术系统相关的术语定义的厘清、概念框架的构建、研究模型的应用以及未来研究的建议进行了较为系统的理论概括和案例分析，旨在为这一还在发展初期的新兴领域的未来研究和健康发展贡献智慧！

分布式账本技术显然还处于起步阶段，它的潜在影响是巨大的，前景是诱人的，但我们对它的认知正如技术本身的发展一样“还在路上”。同样作为关注跟踪这一领域的研究机构，CCAF和AIF也“在路上”，所以报告本身不足之处在所难免。中文版的翻译工作具体由浙大AIF区块链研究室组织人员完成，由于行业变化快、时间短和专业能力不足等因素，欢迎指正！

贲圣林、杨小虎、张瑞东、李启雷

浙江大学互联网金融研究院

研究团队

项目主管

Michel Rauchs

剑桥大学新兴金融研究中心加密货币和区块链研究部门主管，作为联合作者发布过两篇关于加密货币和区块链生态系统实证分析的标杆性研究报告。

m.rauchs@jbs.cam.ac.uk

联合作者（按字母顺序排列）

Andrew Glidden

伯克利法学院区块链法律研究的主管。研究领域包括公司治理和公司金融，金融监管，协议设计。

asglidden@berkeley.edu

Brian Gordon

犹他大学大卫埃克尔斯商学院访问学者，加州大学默塞德分校研究员。研究主要集中在战略、企业家精神和创新等。

brianrgordon@gmail.com

Gina Pieters

芝加哥大学经济系讲师，剑桥大学新兴金融研究中心研究员。她的研究检验了在加密货币不同货币和货币体系中的经济意义和行为。

gcpieters@uchicago.edu

Martino Recanatini

剑桥大学新兴金融研究中心访问学生，在马尔凯理工大学攻读银行与金融硕士学位。硕士论文是关于论述分布式账本技术系统对证券后交易服务的影响。

m.recanatini@jbs.cam.ac.uk

François Rostand

剑桥大学新兴金融研究中心访问学生，在剑桥大学攻读化学工程硕士学位。现在伦敦高盛投行工作。 fr339@cam.ac.uk

Kathryn Vagneur

剑桥大学新兴金融研究中心助理研究员，她在工作中考察了监管系统及区块链项目中的治理、管理控制、弹性以及脆弱性。

kvagneur.phd91@london.edu

Bryan Zhang

剑桥大学新兴金融研究中心执行主任、联合创始人。他参与联合撰写了超过15篇新兴金融方面的研究报告。

b.zhang@jbs.cam.ac.uk

其他参与者

Oliver Beige

工业工程师和经济学家（加州大学伯克利分校博士），他将创新经济学应用于早期的技术研发。曾在SAP和梅赛德斯–奔驰公司工作，现在从事区块链与企业系统交叉方面的咨询工作。

beige@cal.berkeley.edu

Jill Carlson

加密货币协议项目投资顾问，同时也为包括IMF等机构在相关问题上提供咨询。

jillruthcarlson@gmail.com

Nic Carter

Castle Island风险投资公司的合伙人，公司总部位于波士顿，专注投资区块链领域。创建了一家为公有链提供开放数据源的公司Coinmetrics.io。毕业于爱丁堡大学，硕士论文为密码学治理结构方面的问题。

nic@castleisland.vc

Michèle Finck

马克斯·普朗克创新与竞争研究所的高级研究员，牛津大学基布尔学院欧盟法律专业讲师。曾著有《欧洲的区块链监管和治理》（剑桥大学出版社，2018年），并向多家机构提供法律和区块链技术交叉方面的咨询。

michele.finck@ip.mpg.de

Larry Sukernik

任职于数字货币集团（DCG），该集团主要投资初创公司和数字资产。曾在安永公司从事三年加密货币和区块链方面的实践工作。

larry@dcg.co

Angela Walch

圣玛丽大学法学院副教授，伦敦大学学院区块链技术研究中心研究员。她对区块链技术的研究主要集中在治理、语言和操作风险方面。

awalch@stmarytx.edu

翻译校审

贲圣林

浙江大学教授，互联网金融研究院院长，现任浙江大学管理学院财务与会计学系主任、教授、博士生导师，浙大互联网金融研究院创始院长，中国人民大学国际货币研究所联席所长。有丰富的国际金融业经历，是海内外各类论坛的常邀演讲嘉宾，在金融科技、新金融、国际金融等领域发表或出版了多篇文章和书籍。

张瑞东

浙大AIF区块链研究室主任，美国内布拉斯加大学林肯分校信息管理学博士，现任美国威斯康辛大学奥克莱尔分校计算机信息系统学终身教授。研究领域包括云计算的架构及优化、区块链及数字货币技术的应用及开发、电子商务、开源技术以及下一代互联网的发展及应用。主要从事计算机网络、数据中心服务器管理及企业级网络资源管理的教学研究工作。多篇学术论文发表在国际学术期刊和重要学术会议上。

杨小虎

浙大AIF区块链研究室首席顾问，互联网金融研究院副院长，浙江大学计算机学院研究员。自1994年起任教于浙江大学，现任浙江大学软件学院常务副院长、计算机学院软件研究所副所长、浙江大学道富技术中心主任。长期从事金融信息技术、云计算、软件工程等方面的研究工作，先后承担了国家支撑计划、国家863计划项目、省部级项目和国际合作项目等20余项，在国内外核心刊物、国际会议发表学术论文60余篇。

李启雷

浙大AIF区块链研究室企业创新顾问，浙江大学软件学院讲师，趣链科技首席技术官（CTO），研究领域为体感人机交互、区块链和移动互联网技术。作为核心研究员参与国家863计划和国家科技支撑计划，在国内外知名学术期刊和会议发表论文9篇，获得国家发明专利1项，软件著作权1项。

参与翻译

刘谆谆

浙江大学管理学院博士生，浙大AIF区块链研究室负责人，主要研究方向为金融科技监管、第三方支付、区块链技术与数字货币。

11620038@zju.edu.cn

罗丹

浙江大学管理学院博士研究生，浙大AIF网贷与众筹研究室负责人。作为主要成员撰写《远求骐骥，吐故纳新：中国网贷行业的现在与未来》等研究报告，参与编制中国金融科技中心指数(FHI)、全球金融科技中心指数(GFHI)等，参与编写《互联网金融理论与实务》教材，参与完成多项省级课题并撰写多篇省级政府咨询要报。主要研究方向：互联网金融，金融市场与监管等。

luodaniris@163.com

陈雪如

浙江大学管理学院博士生，同时担任浙大AIF创业金融研究室负责人，研究方向为创业金融、金融科技等。

chenxueru@zju.edu.cn

郝睿

剑桥大学嘉治商学院新兴金融研究中心研究员、兼数据科学家。专门从事数据库设计、大数据分析及机器学习。她合作出版了多部具有影响力的替代金融行业基基准报告，并合作参与了英国金融监管局英国众筹市场监管大数据分析。她拥有剑桥大学工程系博士学位和诺丁汉大学环境工程一等荣誉学士学位。同时她还获得量化金融和数据科学家专业资格证书。

r.hao@jbs.cam.ac.uk

郝鑫

剑桥大学嘉治商学院新兴金融研究中心访问学生，清华大学博士研究生。参与多篇新兴金融基准报告的调研和数据分析工作。

x.hao@jbs.cam.ac.uk

晁宪金

浙大AIF区块链研究室全球研究助理，香港科技大学硕士，专业方向为电子通信工程。

cxj2827@163.com

鲁福韬

浙江财经大学中国金融研究院研究员，香港大学经济学博士。

futaolu@gmail.com

叶舒元

浙江大学竺可桢学院人社班17级学生，主修金融，浙江大学区块链协会成员，对区块链与金融交叉内容有所探究，并曾多次参与翻译院网文章。

m18961015876@126.com

施璐吉

就读于中国人民大学财政金融学院。曾参加“首届全国区块链保险创新大赛”并入围决赛。

1216833537@ruc.edu.cn

王皓月

浙大AIF区块链研究室全球研究助理，为中国人民大学在读学生，曾在美国市政府和中国人民银行实习；参与翻译英文教材《Financial Markets and Institutions》。

hederlily@gmail.com

顾雨静

浙江大学会计硕士研究生。曾参与2018亚太地区新兴金融行业调研项目等项目研究。

chloegu413@163.com

李昕

浙江大学经济学院本科生。参与2018亚太地区新兴金融行业调研项目等项目研究。

jeff_ff_lixin@foxmail.com

声 明

过去的两年时间内，剑桥大学新兴金融研究中心收到了多家机构的资助（VISA、安永、野村综合研究所、卓越集团），用于进行独立的学术研究。我们同时也是超级账本项目和Linux基金会的联合（学术）成员。

所有的联合作者和参与者所表达的均为个人观点，不代表他们各自所属机构。

致 谢

感谢Andre Boysen和Sarah Douglas (SecureKey)以及不愿意公开透露名字的几位公司代表向我们介绍他们各自的分布式账本技术系统的设计和运作，并为我们提供了有益的意见。同时感谢超级账本项目的Marta Piekarska帮助我们联系分布式账本技术系统的业内人士。

特别感谢Jon Frost为早期草稿提供的宝贵反馈意见，Robert Wardrop (剑桥大学新兴金融研究中心) 和Victoria Lemieux (英属哥伦比亚大学)在前期讨论中提出的有益观点和评论，以及Louis Smith (剑桥大学新兴金融研究中心)对报告的设计和发表做出的贡献。

独家授权联合发布声明

摘 要

分布式账本技术生态体系由于不完整、不一致的定义以及缺乏标准化的术语解释而饱受诟病，这为政策制定者、开发者以及第一次踏足这个领域的人带来了不必要的麻烦。本研究旨在推动国际化的讨论以形成一套共享的、通用的分布式账本技术体系的语言，厘清术语和概念。我们将提供一个正式的分布式账本技术系统的定义以及一系列其关键特性，使之区分于其他系统。

本报告也引入一个概念框架作为一种多维度的工具来检验并比较现有的分布式账本技术系统，我们相信这一框架将有助于大家理解这一技术并能实现多种用途，这包括开发分布式账本技术应用的企业和机构、该领域的风险投资者、学者、监管人士以及希望对分布式账本技术系统有更好、更细微理解的政策制定者。这个框架将分布式账本技术系统分解为一组相互连接的层、组件和流程。正是这些小的、十分简单的程序组合并交互在一起形成了一个复杂的动态系统。

这些构成分布式账本技术的层遵循一种等级结构：协议层主导着网络层和数据层，它可以支配这些层上的任何决定。分布式账本技术系统中的角色和参与者可以被划分为4类。我们将讨论各个角色和参与者如何分布在不同层，这对于考察这个系统的权力结构至关重要。我们强调分布式账本技术系统的去中心化并非一种二元属性，而是一个由每一层系统组件、层级结构和权力结构相互作用形成的连续变量。

我们的框架并未列出在每一层上的各种流程和组件所有可能的配置，而是显示出不同的设计选择（如不同的配置）导致的结果可以产生具有不同特性和特点的系统。这项实践需要应用由框架延伸出的不同方法来分析每一过程。它同时表明了分布式账本技术内在的权衡属性，并根据特定的安全假设、威胁模型和信任关系在一定范围内改变。这其中没有特定的“对”或“错”：应根据案例的需求和目标判断可选的权衡取舍。

我们进一步指出分布式账本技术系统通常不是单独运作的，而是与多种外部系统相配

合：只有系统内的资源传输是由分布式账本技术系统自身自动执行，不需要外部程序的干预。当分布式账本技术系统产生的交易记录需要参照系统外部的对象、事件或事实（如供应链中追踪的项目、被羁押的物理资产），它与外部的关联则尤其明显。对于这些外部物体，需要有网关连接系统内外，同时依赖外部参与者以及现存的法律结构去执行分布式账本技术系统范围以外的决策。

本报告也论证了对于分布式账本技术系统的设计、架构和治理做出不同的决定可以导致系统性能和特性方面的显著差异。我们讨论了不同情况下进行结算的概念并说明不同架构内的交易是由生命周期的。另外，我们通过比较安全模型依赖于内在的经济激励（如需要系统内部产生的本地资产作为补偿）的系统和通过访问控制和信息记录者间的协约义务保障的系统，来研究信息记录者如何被激励。

我们厘清了这个共享的数据记录结构所采用的形式，并根据分布式账本技术系统处理数据的程度将数据类型分类为交易、交易产生的其它后台数据、包含参与主体的数据记录、交易的流水记录和账本。重要的是，我们使用“账本”来表示大量的网络参与者所共同持有的交易记录的集合。

最后，我们应用这个概念框架对六个案例进行对比分析（比特币、以太坊、瑞波币、Alastria、Verified.me，以及一个被称作“X计划”的匿名分布式账本技术系统），并引入一个分布式账本技术系统地图并把12个分布式账本技术系统定位在这个图上。我们发现参与权无经许可的开放系统主要记录内生资源所有权的转移，而具有更明确许可层级的封闭系统通常引入外部对象到系统中，并且依赖网关和外部执行。我们将说明开放的系统涵盖了从完全中心化到整体上实现合理分散化的系统，而封闭式系统目前由于多种原因趋于中心化，不过也有计划逐步将控制权分散。

第一章　导言

理论基础

分布式账本技术的概念在比特币和区块链技术之前就存在。Lamport et al.（1982）将拜占庭将军问题理论化，描述了“计算机系统在一个包含对抗的环境中怎样处理[...]矛盾的信息”。随后的研究引出了第一个“高度适用的拜占庭容错系统”算法，并且延迟很小（Castro & Liskov, 2002）。最早的可辨认的“区块链”概念的出现可以追溯到Haber & Stornetta (1991) and Bayer et al. (1992)，他们提出了在分散式系统里采用加密的哈希函数和默克尔树，用时间戳高效、安全地记录数据，并将加密的数据区块连接形成链。

然而，这些发展相比于最近人们围绕加密货币或者更一般地，区块链技术的热情来说，引起的关注较少。这些新的兴趣吸引了大量的投资，使得分布式账本技术系统类型和应用快速发展，其中很多与比特币以及其众多的模仿者大不相同。

分布式账本技术系统的概念出现在1982年，而最早的“区块链”的概念出现在1991年。

什么是分布式账本技术系统?

分布式账本技术系统（DLT）作为一种涵盖性的术语用于表示一种在没有中心操作员或管理人员的环境下多方参与运作的系统，即使参与方可能不可靠或者居心不良（对抗性环境）。区块链技术通常被认为是宽泛的分布式账本技术系统的一个子集，它采用一种特定数据结构，而这个数据结构是由哈希值链节的数据区块构成的。

伴随着分布式账本技术类型和用途的扩展和演变，它在词汇和术语中被广泛使用，而在不同的文章里它们的定义常常是模糊的、不准确和不一致的。词汇和概念性术语的乱用，将会阻碍分布式账本技术领域的发展，并且造成社会及行业的法律不确定性以及尚未辨别的金融风险。

当下，许多普遍的关注点在于加密数字资产及可在分布式账本技术系统发行和转移的数字代币。然而在分析这些资产的属性以前，重要的是对其底层基础设施以及某种设计决策如何影响所记录数据的性质有一个坚实的理解。

报告目标

本报告旨在建立一套概念性的框架和术语，使其可以容易地运用到各种分布式账本技术系统

中，无论是出现在加密货币（如比特币）之前，还是受比特币启发其后出现的系统当中。我们也同时探寻将这些新科技与传统的数据库以及其他系统进行区分。这个框架旨在提供一种检验、比较现存的分布式账本技术系统及它们特性和特点的多维度工具，也可以作为考察新的分布式账本技术系统提案的有效的分析工具。

这套分析分布式账本技术系统的框架是通用的，可以被应用在各种分布式账本技术类型和模块中，因此也可以独立地添加新层、组件、程序和配置并且不会影响框架的核心。

研究方法

我们采用一个“系统的视角”进行分析，因为这样可以描述各个部分集合在一起是如何共同作用创建一个功能整体，而不是仅仅是一些零散的部分。我们可以评估在它的环境背景下这样一种系统的行为。虽然系统本身是一个更一般的概念，它表示宇宙中某些部分与其他部分的分离，但采用系统的视角是想要使用非简化的方法来描述系统本身的特性。此外，我们还试图在它们的环境以及整个生态中来考察这些系统，而不是将其视作孤立的个体。因此，人们可以考查一个分布式账本技术系统和它的环境之间的互动和联系。

这种方法来源于系统理论，该理论并行地发展在各种不同方向的研究中，最早在20世纪40年代由Ludwig von Bertalanffy（1949）发表第一部著作。他阐明了一个一般的系统理论的概念，阐述了其多学科的性质，并考察了作为系统一般科学的“整体性”。Ervin Laszlo（1972）在系统、系统属性和系统间关系方面提出了一种知识组合，他称之为“系统哲学”。Walter Buckley（1967）和James Grier Miller（1978）进一步完善了Bertalanffy的一般系统理论，其作为一个理论框架和方法可以应用于物理、生物和社会科学等领域。特别值得注意的是Miller的“生命系统”的概念，他提出系统可以具有等级层次和子系统层，并由信息流、能量流和物质流维持。

按照系统论的思想，我们试图把分布式账本技术系统概念化为一组相互关联的、具有等级结构的组件以及它们之间相互作用的过程。它不是一个简单的部件集合，而是层次组件的“配置”及其相互关系和相互作用，而正是这决定了特定分布式账本技术系统的功能和特性。

报告结构

报告的其余部分结构如下：

第二章为对已有文献的回顾，总结其理论概念和框架，并提出它们的局限之处。接着建立一套正式的分布式账本技术系统的定义并强调它需要满足的必要标准，并定义了几个关键术语。

第三章 通过引入概念框架中的各种要素对提出的工具进行高度概述。

第四章研究分布式账本技术系统中的层间依赖关系以及与外部系统的相互作用和关系。

第五章深入探究概念框架中的每一要素，通过现有的分布式账本技术系统的例子概述它们可能的配置和其对系统的影响。

第六章把该框架运用在比特币中，并将其与其他被视作替代性设计的案例研究进行比较。

第七章总结报告并提出建议，这个概念框架可以被如何扩展以及可以应用于何处。

附录A以表格的形式对完整的框架进行呈现；附录B总结了六个案例研究的对比分析（比特币、以太坊、瑞波币、Alastria、Verified.me，和“X计划”）；附录C为一个常用术语的词汇表。

第二章　分布式账本技术系统–设置场景

2.1 分布式账本技术系统 - 设置场景

2.1.1 定义

各种文献中存在有许多不同的分布式账本技术（DLT）系统定义，许多以此为主题的出版物都在序言中列出了自己独特的定义。有些是狭义上的定义，有些定义则很宽泛；有些是矛盾的。 因此，一个大家都能接受的分布式账本技术的定义还有待发掘。

例如，世界银行（2017）将分布式账本技术系统描述为“比较宽泛定义的分享账本落地的一个特例，” 而这个比较宽泛定义的分享账本是指账本上记录的数据可以在不同参与方之间分享。

来自欧洲中央银行（ECB）的Pinna & Ruttenberg（2016）将分布式账本技术描述为一种技术，该技术“允许其用户在交易或账户余额的共享数据库中存储和访问与给定资产及其持有者相关的信息。 该信息散布于用户之间，然后用户可以使用它来结算他们的(资产）转让，例如， 证券和现金，而无需依赖一个可靠的中央验证系统”。 Davidson等人（2016）将分布式账本技术系统视为一个“分布式的、加密安全的且有加密经济激励的共识引擎”。

与之相反，英格兰银行（2017）提供了一组定义分布式账本技术系统的关键架构特征：“分布式账本技术是一个分布式数据库，也就是说每个节点都有一个同步的数据副本，但分布式账本技术传统的分布式数据库架构在三个重要方面有所不同：（i）去中心化；（ii）无信任环境中的可靠性；（iii）加密。” 英格兰银行将其定义概括为：一个能够以分布式和去中心化的方式保存和共享记录，同时通过使用基于共识的验证协议和加密签名来确保其完整性的数据库架构。

同样，Tasca & Tessone（2018）列出了似乎是分布式账本技术系统独有的一系列关键功能：分布式账本技术系统是基于社区共识的分布式帐本，其存储数据不是基于区块链，区块链的原则是（a.)去中心化共识（b.)透明化 (c.)安全性与不可更改性。

其他的定义仅指“区块链技术”，不区分分布式账本技术和“区块链”。 例如，Cong & He（2018）将区块链定义为“分布式数据库，即以“区块”为单位自主维护不断增长的公共记录列表，防止篡改和修订，而Atzori（2015）将其描述为一个“不可逆转且防篡改的公共记录存储库，用于存储可以嵌入信息和指令的文档、合同、财产，其应用广泛。”

如这些示例所示，对于所谓的分布式账本技术系统，没有真正的和通用的定义。 更具挑战性的是，一方面，定义有时过于具体，技术性使得一般受众无法理解;而另一方面，有些过于简单和宽

泛，因此无法观察到与更传统的数据库架构的有意义的差异。 无论是哪一种情况，缺乏通用术语都已导致误解，并且对这项技术可以实现目标的不切实际的期望已经广泛形成。

2.1.2 现存的框架

本体 – 对存在的事物的描述，以及如何根据相似性和差异将它们组合在一起 – 使人们能够在特定的生态系统中向一个共同的术语聚拢。因此，项目团队对以前提出的本体论进行了分析，以此来了解学术界，专业人士和其他撰写过该主题的人员提供的分布式账本技术生态系统的分类建议。我们总结了下面的一些框架，并在2.1.3节中讨论了它们的缺点。

Kada等人（2017）提出了基于两个维度的区块链分类：a）权威的存在和b）有激励的参与

Lemieux（2017）通过档案科学的视角分析区块链，该理论强调记录保持和真实记录的保存。这项工作就记录保存系统的类型构建区块链，即“镜像类型”，“数字记录类型”和“标记化类型”，并检查与正式档案理论评估框架相关的每种类型。

Platt（2017）提出了一个简单而强大的二维框架，根据（a）数据扩散模型（全局与本地）和（b）链上功能（有状态与无状态）对分布式账本技术系统进行分类。

De Kruijff & Weigand（2017）试图解决企业区块链文献中形式化缺乏的问题。 Kruijff使用企业本体理论来区分区块链交易和智能合约的数据逻辑、信息逻辑和所处的层级。

Xu等人在2017年已经开发出了本报告框架所采用的“分层方法”。 他们的研究旨在评估区块链设计决策对软件架构的影响， 所建议的分类法旨在帮助解决基于区块链的系统性能和质量的架构（软件）方面的考虑。

Glaser（2017）使用清晰的术语来作为沟通的基础，并将术语与数字市场模型联系起来，以确定每个组成部分的市场含义。他的想法也是基于Glaser & Bezzenberger（2015）的目标，为P2P分类转移系统和去中心化系统提供早期工具。

最后，Tasca和Tessone（2018）试图在之前的定义之上对分布式账本技术系统提出一个整体的定义。 这种先进的本体论对于区块链技术的分类是非常全面和详细的。

2.1.3先前工作的局限性

针对分布式分类帐技术一直有多个定义，每个定义都在细节上有所不同，这使得我们很难从特定定义推断出能够描述和分类不同类型的分布式账本技术系统的通用、模块化框架。

由于先前工作中对分布式账本技术系统的组件的定义及其明晰程度的缺乏关注，想进一步讨论也很困难。 例如，去中心化通常被视为分布式账本技术系统的二元特征，而不是由各层和其中嵌套

子系统的相互作用产生的连续变量。 这部分来源于当今文献中的示例，这些文献中不将系统分解成不同的组件并检查这些不同元素之间的关系、依赖性和相互作用。

为了克服这些限制，本研究旨在提供分布式账本技术系统的工作定义，并采用一个整体方法，从流程层面构建开发一个通用且耐用的工具。由此产生的概念框架可用于各种目的，包括对现有系统的评估，多个系统的比较分析以及新系统的开发。

2.2 什么是分布式账本技术系统？

第2.1节强调了关于“区块链”或“分布式账本”构成的众多矛盾性定义。不明确的术语和模糊的边界导致“分布式账本技术”演变成一个总称，用于指定各种松散相关的概念（其中包括区块链）。

对分布式账本技术概念的一种解释是其最狭隘（有历史根源）的定义：由加密链接的“数据块”构成的仅支持附加链，由分散的网络维护和更新，网络节点受到经济激励的鼓励非战略性地参与维护和保护系统，以便数据以一种通常被称为“全球总账”的特殊结构组织 – 抵御对抗性干扰，双花，谴责，伪造，勾结，篡改或其他类型的恶意行为。

然而，这种狭隘的定义不仅排除了分布式账簿技术的许多现有和潜在的未开发应用。它还排除了企业应用分布式账本技术一词的一种情况，在这种情况下由于范围过于广泛以致它与更传统的分布式系统之间的界限变得模糊，且许多狭义定义的核心要素缺失或退化。

为了解决这个问题，我们建议采用另一种方法来平衡频谱的两端，该方法侧重于分布式账本技术系统的基本最低要求（即必要和充分条件），而不是阐明全套 分布式账本技术系统可能在理想状态下拥有的属性。我们将分布式账本技术系统视为分布式系统的一种类型或子集，它们具有一系列特定的特征，这些特征可以将它们与更传统的分布式系统区分开来。

分布式账本技术系统被设计为能够在对抗性环境中运行

什么是对抗性环境？

对抗性环境的特点是在一个系统或网络中存在恶意行为者，他们以不恰当的方式使用系统从而破坏它。 分布式账本技术系统中的典型对手通常未经授权试图利用共识规则转移资产，审查其他人的交易或以其他方式破坏网络的运行。 这些恶意行为者可以在系统内部或外部搞破坏。

从本质上讲，分布式账本技术系统是一种“共识机器”：一个多方系统，在没有中心协调员的情况下，参与者就一组共享数据及其有效性达成一致。分布式账本技术系统与传统分布式数据库的

区别在于其能够在对抗环境中支持和维护数据完整性的特性，而且这个特性是植根与其设计中。

分布式账本技术系统是多方“共识机器”

分布式账本技术系统可以在一定范围内容忍主动试图攻击系统的恶意行为者和不可靠而诚实的行为者的存在.这种容忍只延伸到数据的记录和处理范围内;希望一起交易的各方也许能够依赖系统的性能，但仍然必须普遍信任他们的交易对手.因此，分布式账本技术系统可以被描述为一种“将信任委托给端点”的非中介技术（即最终用户）系统。

这些特征取决于系统的体系结构和设计，以及其操作环境;

这些不是某些“自然法则”或“不可变要求”的结果。 同样，对入侵者的容忍并不意味着所有分布式账本技术系统必然在对抗环境中运行，或者它们提供抵御对抗性攻击的无敌防御。

图1提供了由单个实体运行的传统数据库系统，传统分布式数据库和分布式账本系统之间的基本差异的说明。 尽管每个系统都从各种来源获取输入，但是对数据的存储，处理和执行方式的控制因类型而异。

图1：从集中型数据库到分布式分类账

注意：传统的分布式数据库由多个节点组成，这些节点共同存储和处理数据，但是，节点通常由同一实体控制，和有着多个控制者的分布式账本技术系统刚好相反。

分布式账本技术系统是一个电子记录系统，使独立实体能够围绕共享的“账本”建立共识 – 而不依赖于中央协调员来提供记录的权威版本。

一个数据链路终端系统需要能够确保拥有以下属性，或者是已经在现有系统中的，或者是进行最小的更改能够达成的。

a. 共享记录：允许多个当事方共同创建、维护和更新共享的一组权威性记 录（“账本”）

b. 多方共识：使各方能够就一组共享记录在一下情况中达成一致：.

i. 如果没有许可，不依赖于单方或双方协议，并且在没有事先建立信任关系的情况下；

ii. 如果获得许可，则通过多个记录制作者获得某种形式的合同或其他协议的批准和约束。

c. 独立验证：使每个参与者能够独立地验证其交易状态和系统的完整性。

d. 篡改证据：允许每个参与者不费事地检测到没有达成共识的对记录的更改。

e. 防篡改：使当事者很难单方面更改过去的记录（即交易记录历史）。

因此，我们提出以下的正式定义：一个分布式账本技术系统是一个满足以下条件的电子记录系统。

i. 使独立的参与者能够围绕其达成共识；

ii. 拥有经过加密（签字）的权威性排序；

iii. 这些记录通过多个节点复制来存储；

iv. 通过加密哈希链来识别篡改；

v. 作为记录对账/共识流程 即账本” 共享结果的权威版。

因此，分布式账本技术系统的目标是产生一组权威记录，其中涉及多个独立实体的多方共识过程进行验证和执行，所有这些都是在去中心化的条件下进行的。用户创建和公布未受证实的交易（即建立新的账本条目的提议）会和记录的制造者捆绑记录，并添加到账本中。然后，所有的审计员会自动执行现已确认的交易中包含的指令。

2.3 相关概念

“账本”概念

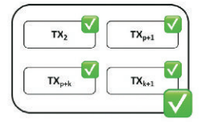
用于描述分布式账本技术系统组成的许多术语之间存在显著的重叠和相似性。这通常会导致术语使用模糊或冲突。例如，术语“账本”。分布式账本技术系统文献不仅赋予“账本”与会计学或金融学等学科中使用的不同含义，并且分布式账本技术文献本身就使用该术语来描述两种截然不同的含义：（i）单个网络节点（ii）大多数节点共同维持的数据集。

在这个项目中我们根据整个网络接受、处理和验证交易数据的范围定义术语：还未达成共识的交易记录池、汇集到一个及节点的分账本、交易记录和账本。

重要概念

交易（transaction）：任何计划的账本变更。尽管有这样的含义，本质上交易不一定是经济上的（价值转移）。

未达成共识的记录（1og）：由单个节点持有的无序有效交易集，尚未依照交易规则纳入网络的正式纪录（即“未经证实”的交易）。



交易记录（record）：受交易规则约束的交易数据。

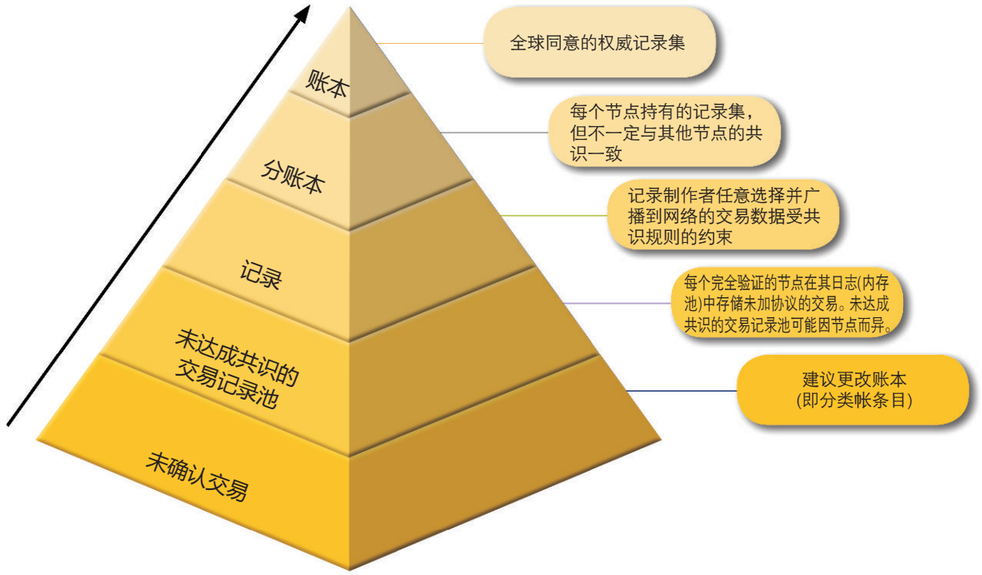
注意：“候选记录（candidate record）”是尚未传播到网络的记录。

分账本（journal）：单个节点持有的记录集，但不定与其他节点的共识一致。分账本是部分的、临时的、异构的：他们可能包含，也可能不包含所有相同的记录。

账本（1edger）：由任意时间点的大部分网络参与者共同持有的权威记录集，以使记录不能被删除和修改（即"最终记录）。

以比特币为例，交易可以是将资产从一个地址转移到另一个；一个节点的分账本是它的内存池（即本地节点从连接的节点收到的未经处理的交易的集合，它们尚未被处理成记录）；记录将是一个被认定的区块；节点的分账本是本地的个人分账本存储的区块链副本，可能不完整或包含网络其余部分未知的数据；账本将是权威的一组区块，这些区块一致被认为是“最终的” – 即被更多工作的子链覆盖的可能性极低。

图2：从交易到记录



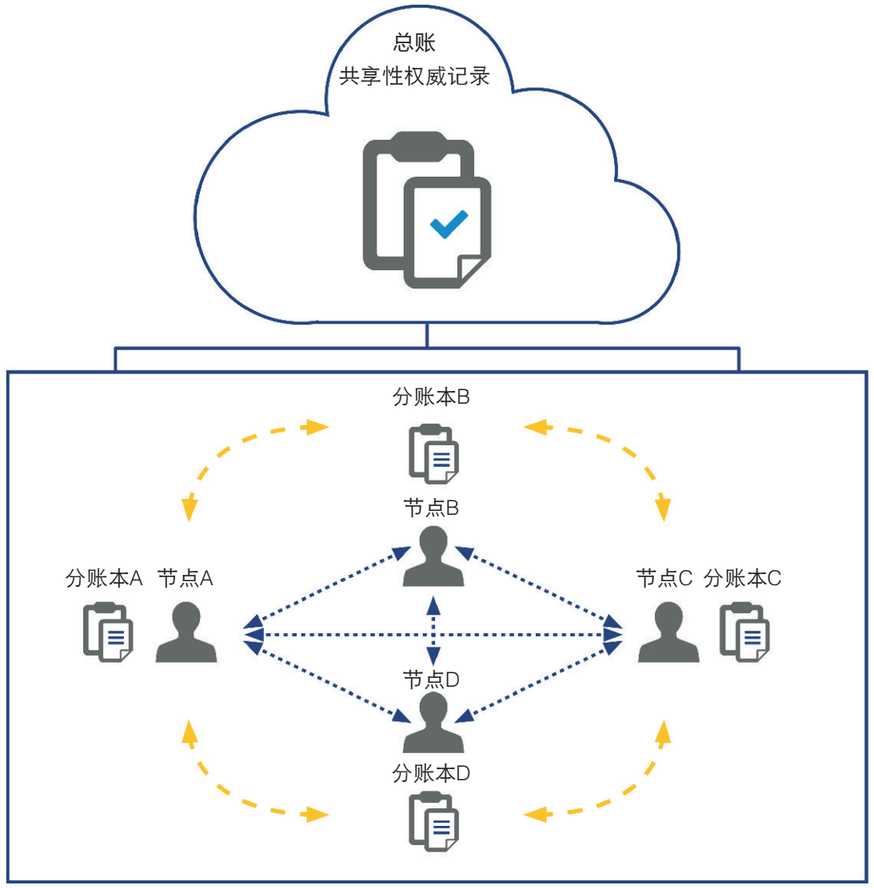
网络中的每个节点都有自己的，可能不完美的账本“副本”（即分账本）。这不仅意味着节点持有的某些数据是临时的和部分的，而且它可能并不总是反映由协议规定的共识机制确定的整套结构化权威记录。

分布式账本技术系统的目标是使结构化记录的这些单个实例（分账本）保持同步，从而导致向单个接受的权威记录集（账本）收敛。这使得一组未必相互信任的独立方能够在不必依赖中央机构的情况下就共享数据集达成协议。从概念上讲，“账本”应该被视为一种潜在的抽象结构，它是由

分布式账本技术系统整体通过不断努力同步每个完整参与者维护的单独的副本生成的（图3）。

所有分布式账本技术系统的核心都是组织和处理产生账本的共享数据。即使参与者或对手不可靠，一个功能性分布式账本技术系统仍会创建和维护账本。

图3：从集中型数据库到分布式分类账



“私钥”概念

分布式账本技术系统中的交易

在分布式账本技术系统中，交易是授权下的尝试 – 由发起者使用私钥加密签名 – 来改变累积记

录的状态（即“状态转换”）。交易通常包含一组指令（例如，代币的发布，代币的转移，更新余额，代币的兑换，事件的描述）。

用户通过将原始数据转化为标准化格式，为交易添加加密签名以进行身份验证，然后将其广播到网络中的其他节点，从而创建交易 – 或者，从技术上讲，以账本条目的形式进行状态转换。 由私钥生成的签名代表着用户对分布式账本技术系统请求反映交易的账本条目的许可.有效签名向分布式账本技术系统提供加密保证，即交易发起者有权制定相应的账本。

如果没有妥善保护，私钥可能被盗，使得小偷可以参与交易，而这些交易无法与发自真正的所有者的交易区分开。

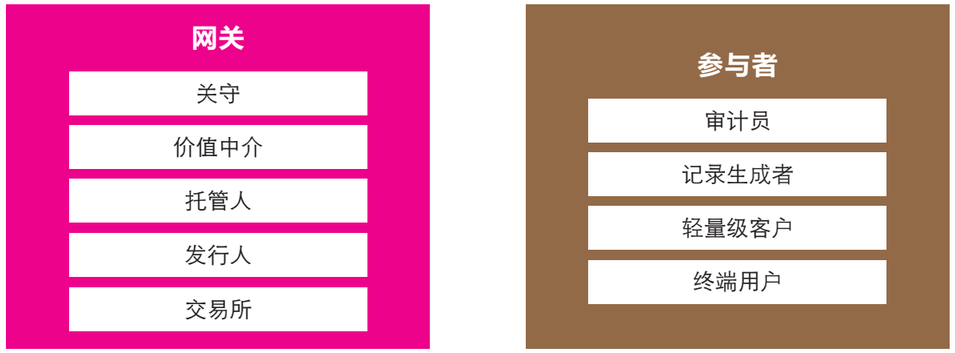
务必注意的是，有效签名不会自动提供证据证明相应私钥的所有者已生成签名。 相反，它会提供一个保证证明私钥持有者已启动交易。 私钥的使用提供了一个强有力的推定，即交易是被授权的。 但是，如果没有妥善保护，私钥可能会被攻击者窃取。 安全地存储私钥会是一项繁琐的任务；众所周知密钥管理十分困难并需要一定的技术熟练程度，这就是为什么它经常外包给第三方进行托管服务。

2.4 参与者（角色）

分布式账本技术系统由执行各种角色的参与者组成。 在此种情况中，参与者是直接或间接与分布式账本技术系统交互的任何实体或个人。 根据他们在系统中扮演的角色，可以将参与者组合成四个关键类别（图4）。

一个实体可以同时承担多个角色的角色，并在多个层上运行。 同样，多个参与者可以同时扮演一个特定角色。

图4：分布式账本技术系统中的参与者类别



2.4.1 开发者

开发人员编写和审查作为分布式账本技术系统及其连接系统的技术构建区块的基础的代码。开发人员可能是被雇佣的专业人员或作为志愿贡献者参与研发。

协议：维护核心协议代码库（或替代实现）。

客户：构建可为分布式账本技术系统提供接口的分布式账本技术客户端。

应用：设计在分布式账本技术系统平台上运行的应用程序。

外部系统：创建基础架构以使协议能够运行或相互作用。

检查和制衡

理想情况下，检查与制衡机制应该产生于参与者和角色的组合，确保任何一方或合作方都不能单方面接管一个分布式账本技术系统。 这反过来也确保了分布式账本技术系统的防篡改特性。

2.4.2 管理员

管理员控制对核心代码库存储库的访问，并可决定添加，删除和修改代码以更改系统规则。 管理员通常会极大的参与管理流程，并对其有绝对的控制权。

管理员的性质和角色因系统而异。例如，封闭和有许可的分布式账本技术系统可能有一个专门的实体担任管理员的角色，而开放的，无许可的系统通常有一组松散连接的“管理员”，其形式是自愿的核心开发者，而不是正式的管理员。 在后一种情况下，这些开发人员实际上并不直接控制代码库；相反，他们提出了由用户独立“批准”的变更（通过选择将提案纳入他们运行的软件中）。

2.4.3 网关

网关通过充当系统与外部世界之间的桥梁，来为系统提供接口。

关守：授予参与者对系统的访问权限。

价值中介：将外部数据传输到系统。

托管人：持有资产。

交易所：促进购买/销售数字资产。

发行人：发行或赎回代表系统中记录的资产的代币。

2.4.4　参与者

网络由互连的参与者组成，通过在彼此之间传递消息进行通信。

稽核员：检查提交的交易和记录的有效性，向网络报告无效记录，以及转发有效的交易和记录。能够对系统状态执行独立审计。通常称为完全/完全验证节点。

记录生成者：生成并提交候选记录集，以便可能包含在账本中。通常被称为矿工或验证人。

轻量级客户：向审计员查询有关特定交易的数据;不完全验证系统。

终端用户：需要网关访问系统的系统的间接用户（例如，保管钱包服务）。

分布式账本技术系统中的参与者可以承担多个角色并在多个系统层上运行。例如，一个实体可以承担多个角色，就像一个角色可以由多个实体执行一样。每个分布式账本技术系统都有不同的参与者、角色和实体;跨层、组件和流程的角色分配和重新分配会影响系统的属性。

第三章　框架的介绍

本部分将从检验组成分布式账本技术系统的必要及充分元素开始，意在为针对分布式账本技术系统的分析和分类提供灵活性。如图5所示，一个分布式账本技术系统可被分割为三个互相依赖的核心层。

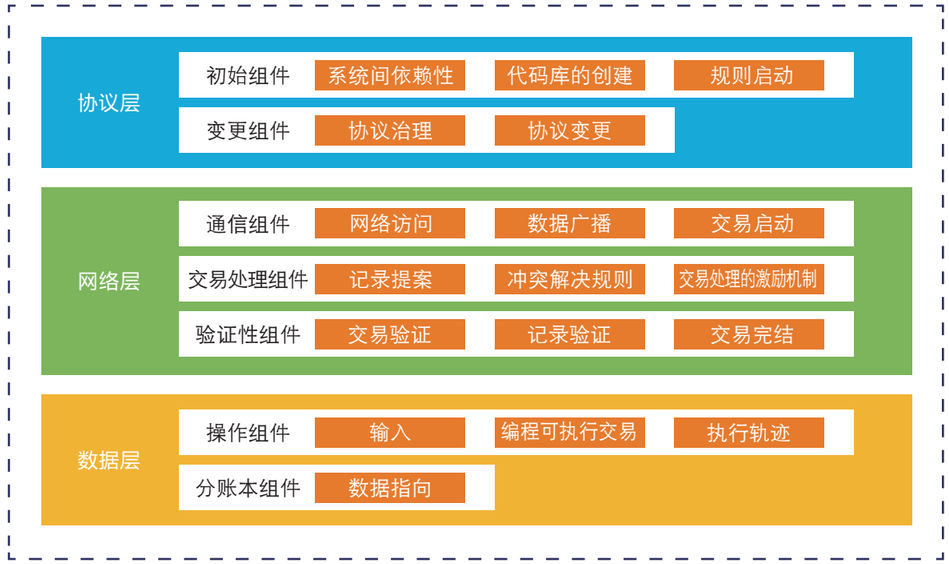
协议层：由一系列软件组成，它们定义了系统运行的规则

网络层：互相连接的参与者以及执行协议的进程

数据层：流经系统且携带了与系统意图为用户展现的设计与功能存在特殊关联的信息

核心层 » 组件 » 进程

图5：分布式账本技术系统框架



每一个核心层由一个或多个用于创造或操作分布式账本技术系统的组件组成。其中，组件是一个系统功能必需的相关进程的逻辑集合。进程是一系列参与者实施的行为，其目的在于达成一个或一系列特殊目标从而使组件成功运作。整体框架将在附录A以表格形式呈现。

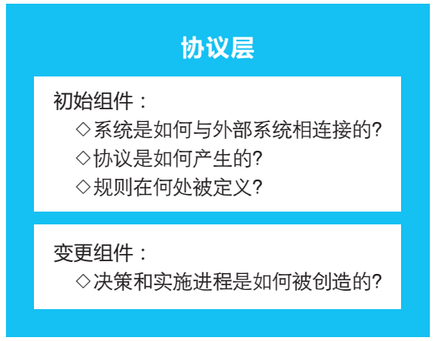
3.1 协议层

协议层是整个分布式账本技术系统的基础：它定义了一套管理系统的正式规则，并对架构设计进行了编纂。此协议可被认为是由所有的系统参与者商定并认可的一套“宪法”。协议层包含了两个组件：

初始组件：在网络启动时对分布式账本技术系统的进程赋予定义。其由初始代码库及体系架构组成，用于确定系统内的参与规则，包括第一个（“初始的”）记录。

变更组件：定义协议如何随着时间推移而演变。它包含了治理层面（比如：如何制定群体决策）以及执行层面上的考虑（比如：决策结果应如何纳入体系）。变动组件不一定是协议明确的一部分。事实上，大部分分布式账本技术系统将治理层面及相关问题进行了“脱链”。

图6：协议层



链上VS链下：

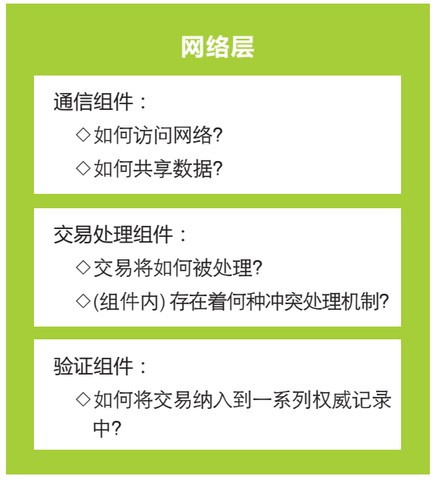
“链下”指的是任何发生于分布式账本技术系统正式边界以外的事。相反地，链上则指的是任何发生于分布式账本技术系统边界以内的事。

3.2 网络层

网络层是由相互连接的参与者构成，他们共同进行数据的储存、分享及处理。网络层是协议层规则的实际执行场所，它展现了参与者们是如何访问系统，数据是如何通过网络被分享，账簿是如何被更新以及参与者们是如何核实交易和账目的有效性的。它包含了三个核心部分：

通信组件的功能包括，确定哪些参与者可成为参与者以及访问网络（开型网络或封闭网络），决定数据如以何种形式被分享（公开或私密）以及谁有权限开启交易（无限制或有限制）。

图7：网络层



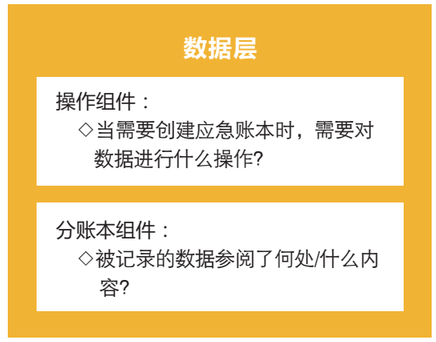
交易处理组件是一系列用于确认共享权威记录的更新机制的进程：1. 哪些参与者拥有更新共享权威记录的权限（无需许可或者需要许可）。2. 如何使参与者们达成对于更新执行的一致意见。

变更组件执行审计行为以验证交易和记录是否遵从协议层规则，即，有效且无冲突。这是分布式账本技术系统的一个重要方面，它使节点拥有了独立校验系统内发生情况的能力。通常，人们人为，存储在分布式账本技术系统中的记录是“不可变的”，即永远无法被逆转。但是，事情并不一定如此：分布式账本技术系统可根据系统设计的不同提供不同程度的交易完结性。这意味着，一项已确认（以及已执行）的交易有可能会被取消。本文的第5.2.3节将会提供有关交易完结过程的详细论述。

3.3 数据层

数据层是指被分布式账本技术系统处理，并以记录形式存储的信息。数据层是系统提供的核心功能。分布式账本技术系统的目的是创建一个共享数据架构——账本——它具有一系列关键特征，最重要的几项分别是持久性、透明性、标准化和防审查。在一系列的由分布式账本技术系统协议定义的信息状态、功能、产权和关系中，账本提供了一项即时的权威性记录，该记录可在系统的现有用户之间共享，亦可随着时间推移、系统用户间的互动而持续更新。数据层由两个组件组成。

图8：数据层



操作组件：用于管理在新记录创建、已有记录修改和代码执行时何种数据以及数据是如何被使用的进程。 这其中可能包含“智能合约”。

分账本组件：关注储存记录的内容（即，哪些记录中的数据曾被参阅，区块中的内容是什么？）

抗审查性：

抗审查性是一个分布式账本技术背景下的常用术语，通常是指某一方或集团丧失单方面执行以下任何一项的能力：

1. 更改系统规则

2. 阻止或审查交易

3. 扣押账户和/或冻结余额

以编程方式执行的交易（智能合约）

以编程方式执行的交易（PETs）是计算机上写好的程序，由特定消息触发，并由系统执行。当代码能够按照所有各方的意图进行实施时，执行的确定性特质降低了各个参与者彼此交互所需的信任级别。例如，这些程序可以用代码替换信托关系，例如监管和托管。这些通常被称为“智能合约”，但并不是自主的或自适应的（“智能的”），也不是法律意义上的合同。相反，它们既可以是合同的证据，也可以是实施合同或协议的技术手段。

参考/价值链接

记录的性质以及记录指向的价值是分类账组件的重要组成部分。记录可以引用内部对象（例如，诸如比特币/BTC或以太/ETH的本机令牌）或系统外部的一些内容（例如，跨供应链跟踪的某一实际物件）。

分布式账本技术系统只能强制执行引用了内生性（内部）对象的记录。

区别内生性（内部）和外生性（外部）对象对于说明分布式账本技术系统的边界起着至关重要的作用：分布式账本技术系统只能自动且独立地执行指向系统内生性资源的交易。一旦记录引用了外生性对象，执行将会变得依赖于外部代理。

在此类情况下，（记录）的执行将会依赖于现行法律和社会经济结构及其他分布式账本技术系统外的规定安排。一些架构（例如比特币）在控制特定资产的参与者不合作的情况下，无法遵守外部代理（例如法院）的决定——这一概念被称为“主权”。本文第4.2.2节将详细讨论了原生，内生性和外生性对象。

3.4 框架整合

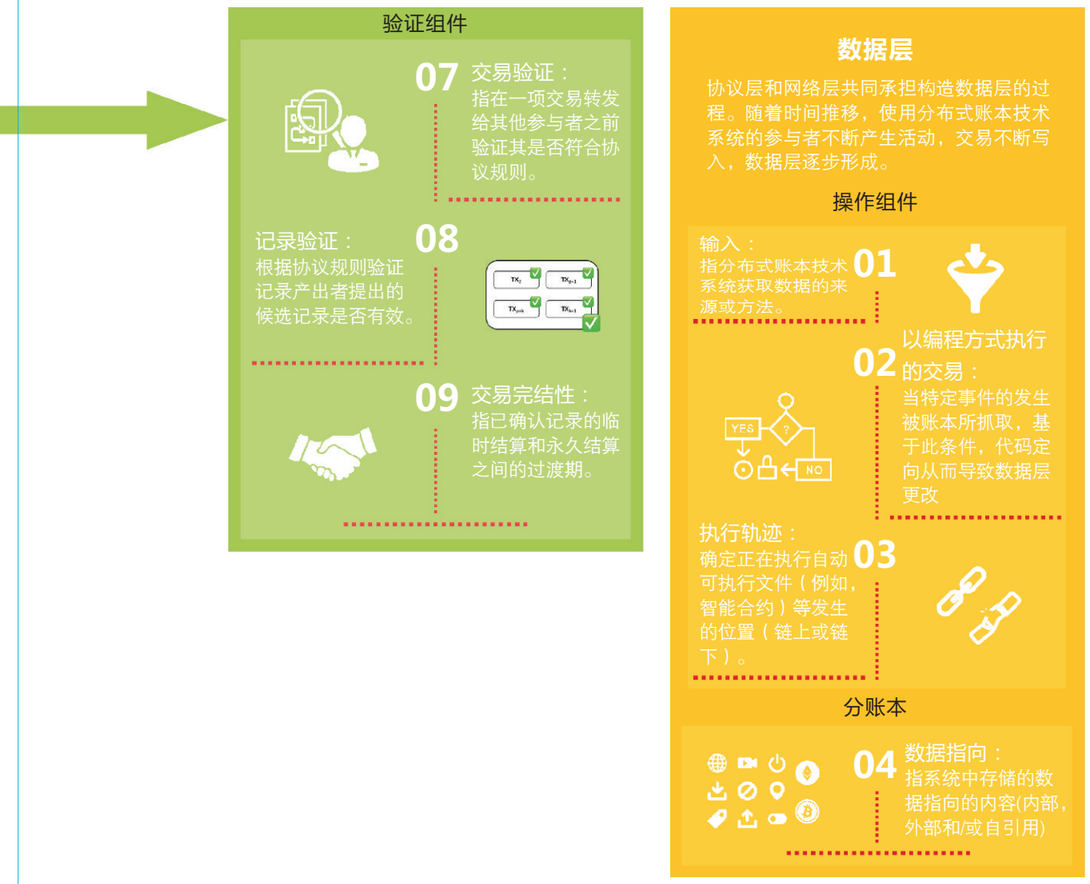
上文提出的概念框架将分布式账本技术系统拆分成三个基本层面：

◇协议层定义、管理和更新系统的全局规则集；

◇网络层实施规则集并执行必要步骤以实现系统范围内的共识；

◇数据层明确数据的性质及含义，在此基础上达成协议。

图9：总结了与分布式账本技术统每一层面功能相关的组件和进程



第四章　系统交互

4.1 系统界限之内

4.1.1层间依赖

分布式账本技术系统由三种相互依赖的层级构成，系统的“较低”层决定了“更高”层。这种排序不是空间意义上的，而是反映了层级概念上和功能上的依赖关系（参见2.3节中的图2）。

协议层定义了一套规则，管理网络中的参与者相互联系的操作。协议层支配网络层，而网络层同时也管理数据层，数据层按时间次序记录条目、对账本进行修改。

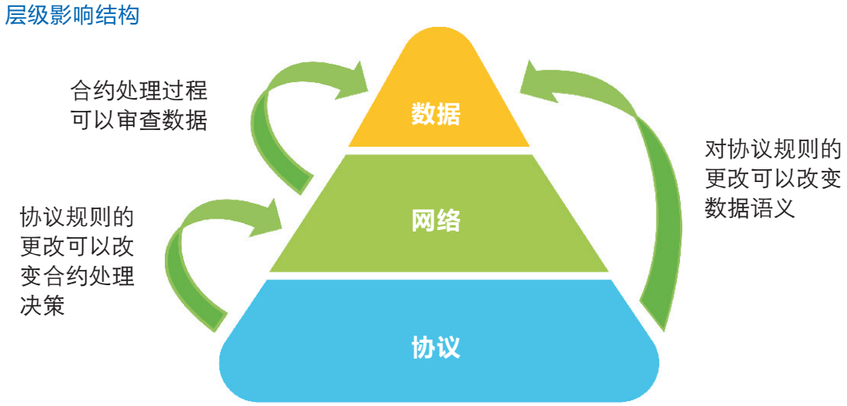
协议只是一个软件，它本身是无生命的。但协议在由网络实现时变得“栩栩如生”。网络通过协议定义进行操作，是一个有独立的服务器和存储空间的系统。在传统IT的架构中，服务器和存储全部由单个公司或政府实体拥有、运营和维护。但分布式账本技术网络与许多传统IT架构不同，前者涉及的许多参与者不一定事先了解对方或相互信任，但他们贡献资源给分布式账本技术网络，以换取相应参与分布式账本技术系统收获的价值。

相应地，协议层和网络层可以构建和维护数据层：一个由多方基于共识创建的共享数据库，它具有特殊属性，如防篡改（参见第2.2节中列出的分布式账本技术的五个关键属性）。

4.1.2 层次结构

在考虑分布式账本技术系统的弹性，稳健性和防篡改性时，重要的是评估各层之间的关系，并了解它们如何相互影响（图10）。

图10：层级影响结构



网络层可以影响数据层处理事务的过程：同伙记录生成器可以通过忽略和拒绝传递相应的事务（即不将它们添加到记录）来决定审查任意数据。这意味着尽管数据层表面上是无需权限的（允许任何人在分布式账本技术系统之上构建应用程序），但它存在被同伙记录生成器审查或操纵的风险。

协议层可以影响网络层和数据层。由于协议规定了系统运行的规则，因此规则的更改可以覆盖交易处理过程中记录生成器在网络层所做的决策。此外，修改协议规则可以改变已处理数据的语义并覆盖数据层的先前配置。

协议层始终可以覆盖网络层和数据层的操作和决策

因此，控制协议层的人有能力直接影响网络层和数据层。在网络层采取的决策通常只影响数据层，但在某些情况下，任何一层都可用于协调网络上的协议变更（例如比特币的BIP信令流程；Decred的链上治理投票模型）。这意味着真正可抵抗外部干扰的系统需要在协议层和网络层都足够去中心化，以避免单方审查和控制。例如，特定的块或事务可以在协议级别被“列入黑名单”。在集中式协议层之上的分散式网络层总是容易受到任意规则变化的影响，这些规则变化推翻了记录生产者达成的共识决策。

4.1.3权衡：没有一种系统能适合所有模式

不同的目标需要不同的设计选择。分布式账本技术系统的一层设计配置可能会影响其他层或组件，并导致不同的系统特性，从而出现好处和坏处的权衡。每个系统都根据其目标、安全性、信任和威胁模型进行权衡。系统可能会偏好某一特定特性，但这种选择将不可避免地以牺牲另一种特性为代价。例如，系统中信任的存在（例如，在封闭的分布式账本技术系统中识别的受管制的实体）允许使用比为最小化参与者之间的信任要求而构建的分布式账本技术系统（例如比特币）更灵活的设计方法。

早期的分布式账本技术系统特别强调保持系统的所有方面都要做到“去中心化”，以便改善网络的抗审查性。这需要付出巨大代价：低效冗余，固有的扩展限制，低吞吐量，缓慢的确认速度，高能源成本和糟糕的用户体验等等。随后的分布式账本技术系统试图解决其中的一些问题，但这些设计选择是以牺牲其他系统属性为代价的，或者是系统更加集中化。

每个设计决策都涉及一系列复杂的权衡

使用当前技术，权衡最常见的是围绕同一组属性（例如，去中心化，验证速度，安全性，参与者激励，复杂性，吞吐量，信任要求，网络规模）。“去中心化/性能”的权衡是讨论最多的问题：通常，分布式账本技术系统越集中，运行速度越快，成本越低，效率越高。

用例需求应规定设计选择和可接受的权衡

一个设计选择很少能在各方面都超过其他的设计选择；一般来说，一个系统不可能做到全是优点而没有任何缺点。因此，在分析具体的设计决策时，应该意识到所涉及的权衡，并仔细评估由此产生的权衡是否可以接受。最终，分布式账本技术系统旨在服务于特定目的：该目的应决定设计选择和可接受的权衡。第6.2.5节中的图23概述了一些对其他系统属性有影响的常见设计选择。

4.1.4关于“去中心化”的说明

“去中心化”是分布式账本技术生态系统中的关键流行语之一，它经常被误认为是目的本身，而不是达到目的的手段。虽然在分布式账本技术应用的许多讨论中这个概念十分重要，但它的定义却十分模糊。

系统理论方法将去中心化视为特权方的缺席，或者相反，参与者选择其信任或交易的对象的能力的缺席。在这种观点下，如果存在某一对象（或对象集合），在任何层中所有使用者都必须与其交互，则系统是集中的。如果任意的对象都可以被忽略或绕过，则系统完全去中心化。然而，这并不意味着也不保证权力的稀释。

Buterin（2017）定义的分布式账本技术系统背景下的去中心化有一个特点，它指通过平台内的用户参与创建的数据结构分布在许多不同的机器上，这些机器在参与者的控制之下，同时参与者也不一定知道或者信任其他人。但是，这种描述过分强调了数据的复制，而忽略了其他关键要素。

另一种观点是量化地考量去中心化，需要多少参与者妥协才能使系统违背预期运行。然而在实践中，测量这个数字，或者在不同系统中进行比较是非常困难的。

在“去中心化”的所有定义中，反复出现的主题是系统是否具有允许自由和开放参与的流程和机构，并鼓励活跃的讨论，而不是将决策或系统管理的职能转移给某一固定的实体。

分布式账本技术系统中的“去中心化”不是二元属性：它是多层行为的积累

因此，鉴于分布式账本技术系统由多个进程和子系统组成，分布式账本技术系统的“去中心化”不是简单的二元属性。集中程度反映了各个层面的相互作用决策和权衡的积累。实际上，识别整个系统中的集中化和去中心化的影响因素更为有用，因为纯粹的去中心化在硬件和软件层面都是很难实现的。

分布式账本技术系统可以在其每个层级具有不同程度的去中心化

例如，数据层可以是去中心化的（即无权限控制的应用程序开发），而网络和协议层由单方控制。或者网络和数据层可以去中心化，但协议层是中心化的。更进一步，特定层内可能存在差异：

例如，网络层中的记录提议和网络访问过程可以由单个授权方执行，而事务验证和记录验证可以在一定程度上去中心化。

澄清分布式和去中心化的流程

去中心化的过程不应与分布式过程混淆。当分布存储或计算时，它被分成多个部分并发生在多个服务器或节点上（“并行化”），与仅使用单个节点相比，可提供更高的效率和更高的弹性。分布式流程仍然可以依靠中央协调员充当记录的权威来源。

当一个进程被去中心化时，多个节点将再次被使用——但在这种情况下，该进程通常在各个节点上复制，这些节点通常由不同的实体控制。这意味着每个节点和其他节点管理相同的存储或执行相同的程序。

这种复制要求是某些分布式账本技术系统难以扩展以适应新用户和交易量增长的核心，因为网络的功能受限于其最薄弱的节点。如果网络试图超过此限制，则弱节点将无法保持同步并将退出网络，从而导致集中化增加。

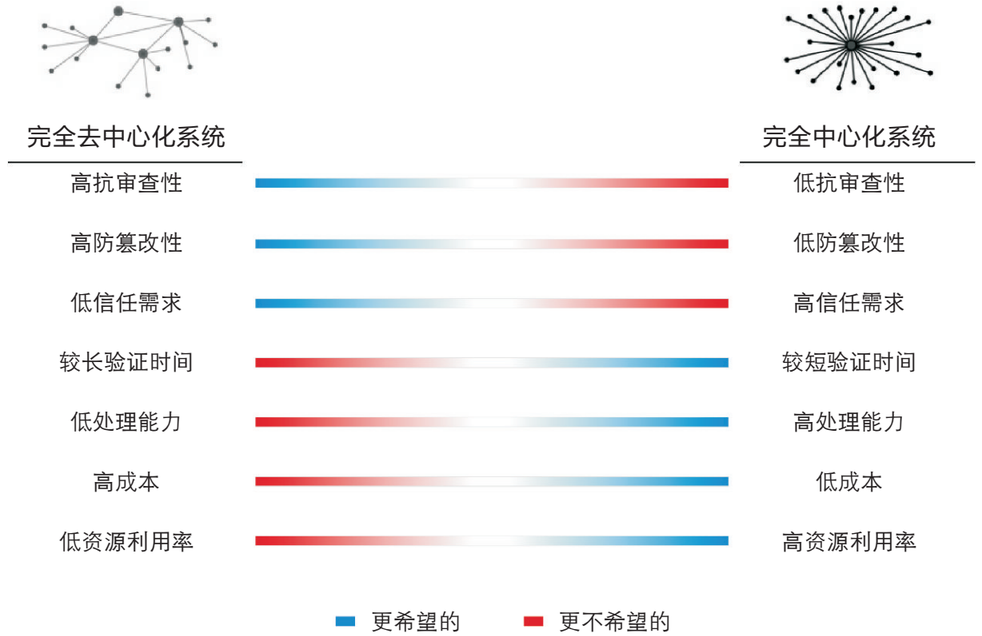
为了确定系统内的潜在权威来源（并最终确定存储的记录），必须注意层次结构。因此，如果没有首先评估在各种情况下每个层内（和之间）的权力机构可能发生的位置和方式，则不能将分布式账本技术系统视为“去中心化”或“中心化”。这些结构可以是流动的并且随着时间的推移而演变，这使得对这样系统的去中心化评估复杂化。大多数分布式账本技术系统在不同层有不同程度的去中心化；有些系统故意选择集中某些方面，以便更好地满足特定目标。

开放，公共和无需许可的分布式账本技术系统（如比特币）力求去中心化以实现抗审查性：任何一方都无法关闭系统，操纵分类账或审查交易。这也增强了弹性，使整个系统能够承受冲击，包括网络参与者的损失。

重要的是要强调使用当前技术导致分布式账本技术系统的设计集中化不仅会影响抗审查性，还会影响其他因素，如安全性，性能（或验证速度）和开销（信息的复杂性）：如前所述，更改 分布式账本技术系统的任何组成部分都需要权衡利弊。

图11说明了分散式和集中式系统之间的一些权衡。 某些分布式账本技术系统在某些方面可能更集中，以强调系统内认为合乎需要的特定属性。鉴于可能存在需要集中化过程的情况，要求系统的所有层完全去中心化以便将其归类为分布式账本技术是不合理的，在实践中也是不可行的。

图11：一种选择



例如，优先考虑记录和交易的验证速度可能会以分类账的复杂性和规模为代价，一些功能（记录保存，智能合约）和记录的规模可能会降至最低。如果使网络集中以提高验证速度，它还可能降低系统的整体安全性或防篡改性。同样，为了提高速度和降低能耗，选择权益证明机制 (PoS)而不是工作量证明机制 (PoW)审查机制可能会影响对参与者的激励，从而影响安全性和防篡改性。

另外，为了提高系统安全性可能会妨碍验证速度，减少交易规模（因为加密交易的空间有限），并阻碍参与者参与，因为运行完全验证节点的成本可能会随着时间的推移而变得非常昂贵。有了这样一个目标，技术的复杂性也可能被限制为附带效应，因为降低复杂性将有助于改善速度和记录规模。最后，允许任何人随意加入或离开的动态使用者网络可能会在网络规模上变得特别庞大，同时由于延迟问题会导致更高的确认时间。

关于系统组成元素集中化或去中心化的所有选择都为分布式账本技术系统同时带来了好处和代价

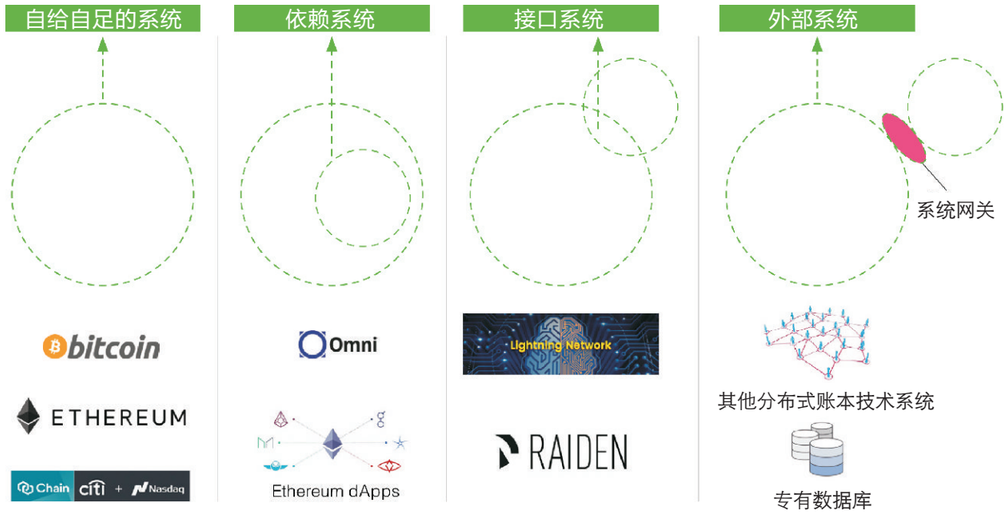
4.2 系统界限之外

4.2.1 系统的视角

分布式账本技术系统很少自给自足。相反，他们经常需要与其他系统不断互动。图12描绘了

分布式账本技术系统部署中看到的不同类型的系统配置。

图12：



自给自足的系统

自给自足的分布式账本技术系统将其持续运行所必需的所有组件合并到其基本架构中，并且系统本身足以实现核心功能。除了更广泛的因特网基础设施（例如，依赖于TCP / IP或类似协议以及底层网络基础设施）之外，这些系统不依赖于其他系统来进行操作。例如比特币和以太网主网等开放系统以及NASDAQ Linq区块链等有权限控制的系统。

根据记录的性质（例如外源/外部），系统可能需要来自外部源的输入。仅此要求不足以将分布式账本技术系统确认于自给自足分类之外。例如，即使没有收到外部数据，代表供应链中资产转移的分布式账本技术系统也应该能够持续存在并发挥作用，尽管它将依赖于网关或接口来提供与资产的创建或物理转移有关的数据。第4.2.2节阐述了自给自足与外部系统之间的关系。

依赖系统

一个非独立的分布式账本技术系统必须与另一个分布式账本技术系统连接才能正常运行。就其本身而言，这样的系统并不是自给自足的。非独立系统的例子是Omni和Counterparty，它们运行在比特币之上，以及在以太坊上运行的dApps（“去中心化式应用程序”）。例如，Omni完全依赖于比特币，因为它是一种跟踪在某些比特币交易中作为任意数据存在的资产的协议。Omni借用比特币的安全性和终结性，同时为交易添加语义内容；它依托于比特币而存在。

接口系统

一个接口分布式账本技术系统是“机会性地”使用由另一个分布式账本技术系统提供的核心功能的系统，但是如果需要，可以容易地重新配置以使用另一个“基础层”分布式账本技术系统。这意味着如果一个系统不复存在，接口系统将能够自己存活至少一段时间，并且可以通过利用替代“基础层”分布式账本技术的功能继续运作。接口系统的长期存活取决于至少一个“基层”分布式账本技术系统的继续存在，并且基层系统的崩溃可能对接口系统产生显著破坏。示例包括“第2层”解决方案，例如基于比特币的闪电网络和基于以太网的雷电网络。这些系统通常旨在提高基础层的可扩展性和功能性，而不会影响网络去中心化或安全性。

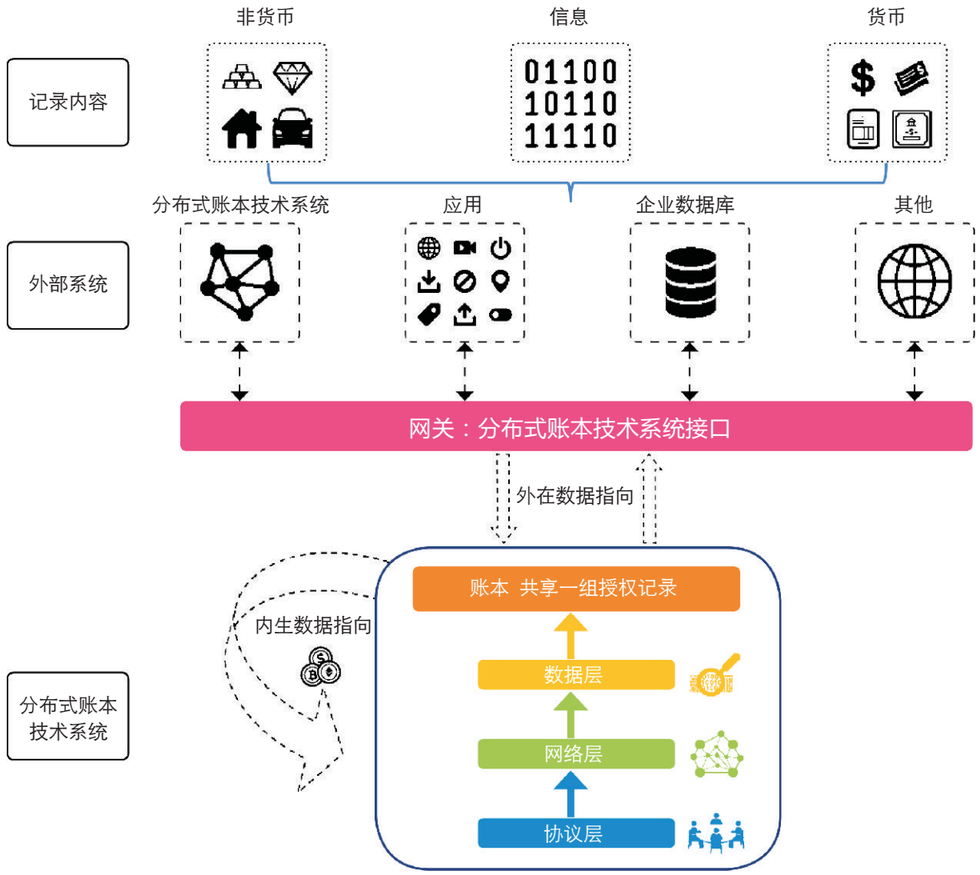
外部系统

外部系统是与“焦点”分布式账本技术系统连接或耦合的任何其他系统。外部系统在架构上与焦点分布式账本技术系统无关或不同。外部系统可以通过网关（通过直接或间接接口）连接到相关系统。这可以是其他分布式账本技术系统以及专有数据库或服务（例如钱包，交换或应用程序）。直接系统网关的示例将是原子交换协议，而间接系统网关将涉及可信中介将令牌从专有数据库传送到系统，或者在两个不兼容的分布式账本技术系统之间进行传输。

4.2.2 外在和内生数据指向

记录可以引用内生数据和/或外生数据。内生数据是完全来自核心系统的信息。 外部数据是指跟踪有关同一实体的信息或分布式账本技术系统外部关系的数据。外生条目可以是资产（货币或非货币）或其他信息的表示。内生数据的一个例子是比特币系统中比特币单位的记录，而外生数据的一个例子可能是追踪全球供应链上的奢侈品手袋的记录。

图13：网关将分布式账本技术系统连接到外部对象



更一般地，如果分账本中的数据仅涉及关于平台上的用户活动的事实，或关于分布式账本技术系统本身的过去历史的事实，则引用类型是内生的。另一方面，如果数据指的是分布式账本技术系统外部的一些状态或用户与分布式账本技术系统的交互，则引用类型是外生的。图13提供了分布式账本技术系统与外部平台之间数据交互的路径表示。

关于原生资产的说明

分布式账本技术系统的原生资产是协议中指定的主要数字资产。根据定义，它们是系统内生的。协议通常使用这些资产来管理记录生产，在网络上支付交易费用，执行“货币政策”或调整激励措施。例如，以太坊的ETH通证是其原生资产，尽管以太坊区块链还承载了大量其他用户定义

的通证（例如，使用ERC20标准）。原生资产通常在系统运作中发挥系统关键作用，因为它们是复杂经济激励设计的重要组成部分。

外部和内生数据之间的区别可能看似肤浅，但事实并非如此。例如，比特币仅作为比特币分布式账本技术系统中的数据记录存在。改变其状态的唯一方法是更改比特币系统内的数据记录。手提包，股票价格或天气读数都是独立于分布式账本技术系统存在的事物的例子，我们可以在不改变跟踪它们的分布式账本技术系统内的记录的情况下改变其状态。

将分布式账本技术与外部系统相连接需要网关作为接口：用特殊的信息源物体作为分布式账本技术系统与外部系统之间的桥梁。在供应链的情况下，这可以是附着在奢侈品上的RFID标签，并且由每个中转站的机器扫描。其他外部系统（例如，其他分布式账本技术系统，应用程序，专有企业数据库等）可以将它们自己的记录信息与原始分布式账本技术系统通信，它提供的信息也将成为一部分数据。

与外部数据源连接需要一个网关;这破坏了分布式账本技术系统自动和独立执行决策的能力

在供应链的例子中，分布式账本技术系统可以正确记录RFID标签的移动，但这些设备可能不一定附着（或嵌入）它们所代表的物体：人们可以想象一个装满了RFID标签的运输箱，虽然它空无一物，但可以欺骗分布式账本技术系统接受虚假交易—有形资产大量转移。类似地，一些RFID标签可能有缺陷，并且交易不一定被记录。

分布式账本技术系统仅具有关于内生数据（即只存在于系统边界内）的有效执行能力（即自动执行决策的能力）。引用外部资源、事实或事件的记录由外部代理提供，外部代理必须通过非系统方式，诚实地报告和/或执行决策。在先前的供应链示例中，对正确记录交易有共同利益的各方需要开发系统以防止或修复任何故障，例如将RFID接口与物理检查耦合。

分布式账本技术系统只能独立和自主地执行涉及内生记录引用的决策

可以写入分账本的内容最终由协议决定。但是，这并不意味着协议必须明确地列出分布式账本技术系统可以记录的所有数据类型。例如，能够支持图灵完备的智能合约的分布式账本技术系统为其用户提供了定义新数据类型的灵活性。

最后，记录还可以引用带有内生和外生性质的数据，在这种情况下，它们被称为“混合”。一个例子是在分布式账本技术系统上直接发布的证券（内生因为它完全存在于系统边界内），证券同时取决于离链现金流（外生因为它需要连接到外部系统）。在混合引用的情况下，确定分布式账本技术系统的执行能力更加困难，因为两个方面之间的关系可能因记录而异。随着企业越来越多地尝试将现有资产转换到分布式账本技术系统上，混合引用是一个快速发展的子领域。因此，将来可能

需要进一步分级。

图14总结了分布式账本技术系统中的记录可以指向的三种引用类型。原生资产完全是内生的，因为它们完全在系统的边界内，不需要与外部世界建立正式联系。相反，完全外生的记录专门引用外部数据，这需要存在（a）接收信息和（b）在分布式账本技术系统之外执行决策的网关。如果没有附加的链接桥接到物质世界，外部数据在系统内是没有意义的。相反，混合记录参考数据，其共享内源和外源特征。因此，执行在某种程度上取决于网关。

图14：三种参考类型

第五章　框架的深度分析

在本章中，我们确定了第三章已经提到框架中协议层、网络层和数据层等通常采用的内部流程的配置。通过这些配置来区分特定的分布式账本技术系统，并在第六章突出他们之间的异同。本章假定读者已经熟悉前面章节介绍过的定义、概念和术语。

5.1 协议层

协议层治理整个系统，负责定义、管理并更新全局规则集。

5.1.1 初始组件

协议层的初始组件指在分布式账本技术系统启动之前要求必需经过和完成的过程。

系统间依赖关系

系统间依赖关系定义了被分析系统的边界，它决定了系统是否可以自行保持运行（例如，自给自足）或是否依赖于另一个系统来正常运行（例如，依赖）。4.2更详细地讨论了可能的配置。

表1：系统间依赖关系

系统类型	描　述
自给自足系统	能够独立运行 – 不依赖于另一个系统。
依赖系统	无法独立运行 – 依赖于另一个系统来运行。
接口系统	能够独立运作 – 长期生存紧密依赖另一个系统。
外部系统	与分布式账本系统交互的自给自足系统（通常充当数据源/接收者）。

代码库创建

代码库创建是指开发一个作为分布式账本技术系统的基础的充分的代码库。代码库可以基于现有框架或从零开始编写。现有流行的框架有无需准入授权的分布式账本技术系统（主要是比特币和以太坊）和需准入授权的分布式账本技术系统（如Hyperledger suite，Corda，Chain和Multichain）的开源代码库。 还有由Digital Asset，Clearmatics和SETL等公司提供的专有平台的闭源代码库。

表2：代码库创建配置

关注点	配　置	描　　述
代码库	现有框架	许多分布式账本系统共享类似的基于现有框架的代码库
	新的/从零开始	代码在意图，编码语言和/或体系结构方面与现有框架有很大不同
开源性	开源	可以分叉：网络可以复制
	闭源	代码库由私人公司或联盟开发，供企业或消费者使用

规则启动

规则启动是指定义在分布式账本技术系统之上运行的规则集。 该过程可以由不同的参与者执行，并且针对特定的分布式账本技术系统。

表3：规则启动配置

关注点	配　置	描　　述
管理员	匿名	创始人实身份仍然隐藏着，运行中用化名。
	志愿者	一组在自愿的基础上合作项目的人; 通常与没有正式治理结构松散联系。
	联盟	正式加入的一组私人和/或公共机构力量，协作开发和管理项目。
	基金会	非营利基金会负责协调和监督活动;根据可能施加信托义务的法律正式注册。
	公司	该项目由指定管理层的单一公司或合资企业实施具体管理。
代码库维护者	开源社区	每个人都有权建议对代码库进行更改; 不一定有形式化的决策过程。
	公司	对代码库的控制完全由公司执行。
	联盟	一组有组织的利益相关者负责集体维护代码库。
形式	形式化协议规范	该协议是形式化定义的–通常以特定文档的形式–所有客户端实现都要遵守。
	参考客户端	参考客户端规定了协议的规则 – 通常在没有形式化的协议规范的情况下。

5.1.2 变更组件

变动组件是指在过程中能够进行适当协议规则。协议变动可以包括消除技术错误（安全漏洞），提升系统的安全性和功能，以及扩展或限制现有协议规则。

协议治理

协议治理是指一系列以有序和合法的方式改变协议的决策过程。 这是一个更为广泛的项目治理

子集，它包含指导和定义协调和行动的全套流程和规范，但可能没有形式化嵌入在分布式账本技术系统。

任何提议的协议变更的基本要素是它的方法是被接受和批准，或者换句话说，如何赋予网络参与者的提案合法性。由于在这种背景下的合法性是一个社会概念，我们发现在分布式账本技术系统中合适确定一些可能的“社会 – 政治”关系。

表4：协议治理配置

配置	描　　述
无政府	由于缺乏中央权力机构，协议变更提案是基于合作和自愿。争议性建议冒着网络破裂的风险，导致永久性分裂。
专政	协议规则变更的决定由单个实体决定（例如个人，公司，矿池）。
分级	个人有能力提出改变，但承认像基金会或控制关键代码库的委员会之类的领导者，协议变更将取决于领导者的同意。或者系统可以明确划分为“头等”和“第二类”用户/节点。
联邦	一组代理人对协议变更进行投票，通过横向关系方案进行链接。 联邦使用者不需要具有相同的声音/权力，甚至不一定彼此了解。
富豪	对协议变更提案进行表决，每个表决权由每个提案人或选民的重要性加权。在富豪统治的情况下，少数选民获得了相当大的权重（例如，由于加权资产的高所有权份额）。
民主	对协议变更提案进行表决，每个表决权由每个提案人或选民的重要性加权。 在民主案件中，少数选民在投票结果方面没有实质性的重要性。

协议治理采用多种形式，通常只是隐式定义。被认为具有无政府（或松散）治理的分布式账本技术系统没有基金会，法人团体或“仁慈的独裁者”来指导决策。这些通常依赖于从免费/开源软件社区继承的治理规范，流程和程序。例如开发人员在邮件列表和会议上的讨论这类非正式流程。在独裁环境中，可能存在这些相同的过程，但有一位公认的领导者有权做出单方面的决定。

在某些情况下，协议治理并不完全适合每一个类别。例如，EOS区块链由区块生产者联盟运行，通过用户/托管人投票（由通证持有者加权）选择。这种安排隐含地将网络划分为“第一类”和“第二类”节点，赋予其等级制、联邦和民主/富豪统治的元素。因此，每个类别应被视为一种机制而不是特定分布式账本技术系统的协议治理体系; 每个系统将展示这些机制的无数排列中的一个，并且每个机制的相对重要性可能随时间而变化。

同样重要的是要认识到“无政府主义”治理模式将始终作为任何开源项目的治理机制存在，与封闭源项目和专有项目不同。这是因为基于开源代码库的分布式账本技术系统在其用户和记录生产者的合作下运行。面对试图强制对用户进行协议更改的问题，他们总是可以选择分叉代码来反转或忽略它。这将导致创建一个独特的分布式账本技术系统，尽管这个系统具有共享历史，直到分歧的时刻（导致网络分裂的“硬分叉”）。这样做的结果是，在某些情况下，即使存在单个“核心”代

码存储库，分布式账本技术系统也可以在治理方面被视为分散的。相反，专有系统可能不允许这种用户驱动的“退出”。

链上治理

链上治理是指在分布式账本技术系统的数据层内并入协议治理功能。目的是使治理形式化，从而提高合法性并避免由于争议或不协调的协议变更导致的网络分裂。已经为分布式账本技术系统开发了各种各样的链上投票方案，从社区情绪的晴雨表到可执行的公民投票。但是，链上治理功能通常只是对其他治理形式的补充。

协议变更

协议变更过程涉及可能提出协议变更的实体，协议变更的资金来源以及变更的实施方式。 实现可以涉及不同的机制，例如提供对特定节点软件的更新，对运行特定软件实例的所有节点的强制升级，以及运行旧版本软件的节点的黑名单。

表5：协议变更配置

关注点	配　置	描　述
提案	公开改动	开放系统允许任何人提出变更建议。
	过滤改动	提案取决于系统的某些要求。例如，Dash和Tezos允许任何人提出更改，条件是通证持有者的批准。其他系统可能采用集中式的初始提交，基于业绩或战略目标进行提案管理。
	授权改动	公司或财团可能会限制谁可以提出变更。
资金	利他主义者	一些协议依赖于志愿者的努力，而其他协议（例如 Monero）可以通过通证持有者或其他相关方的自愿公益捐助来资助开发工作。
	支持开发	基金会，如以太坊基金会，可以通过拨款资助开发工作。虽然这有助于确保一致性和开发人员的责任，但它也可能对协议层产生集中效应。此外基金会本身可能容易被自利方或国家行为者控制。可以通过遵循基金会确定的特定程序来拨付补助金。
	网络化资助的开发	通过发行新通证支持开发工作。发布的程度可以由提供用于满足发展目标的奖励的网络来确定，或者可以由开发者自己定义，但需要网络批准。
	企业赞助开发	公司或财团通过赞助组织资助开发。
实施	运行首选的客户端软件	参与者通过选择运行客户端软件的特定实例来单独实施更改。不需要管理员采取任何操作，但这可能会导致网络从有争议或不协调的更改中分离出来。此模式倾向于减少开发人员或记录生产者对治理流程的控制。
	推向客户	通过直接向客户端推送更新来实现更改,通常由管理员或链上治理系统启动。此模式倾向于优先考虑网络的完整性，但可能倾向于将权力交给开发人员或记录生产者。

不同的分布式账本技术系统可以允许这些机制的混合。例如，以太坊接受其社区以及由赠款支持的开发者的自愿捐款。

5.2 网络层

分布式账本技术系统的网络作为协议规则实现的直接结果而存在。该网络由一组相互联系的参与者和流程组成，这些参与者和流程遵循技术标准（协议）并积极参与通过集成通信渠道交换数据和信息。

5.2.1 通信组件

通信是指分布式账本技术系统中参与者之间的数据交换和共享。

网络访问

网络访问决定了进入分布式账本技术系统的权利; 这是连接到网络的权利。对系统的访问可以是不受限制的，这意味着任何人都可以在任何时间点自由地加入，离开和重新加入系统，或者可能受到负责授予特定实体访问权限的维护者的限制。开放网络通常具有动态和灵活的使用者，而封闭网络可能具有静态/固定使用者。

通常，审计人员通过运行完全验证的节点来直接访问网络：他们被认为是拥有更多权利的“一等”公民，因为他们能够广播、验证和传递交易和记录。参与者还可以通过运行查询交易数据的完整节点的“轻量级客户端”（也称为“SPV节点”）或通过应用程序编程接口（“API”）连接到特定服务器来间接访问网络，该特定服务器设计为接收请求并将响应发送到其他服务器或设备。

表6：网络访问配置

关注点	配　置	描　述
开放性	开放式	无限制的网络访问：只需要下载和运行软件客户端。
	半开放式	访问受到部分限制：潜在参与者需要申请; 通常通过现有网络参与者的链上投票/批准来决定。
	封闭式	访问仅限于经过审查的参与者。 需要一名管理人批准才能加入新使用者。
通道	完全节点	完全执行系统中可用的功能和任务：接收，验证，存储和广播系统中的交易和记录; 执行独立验证。
	轻量级节点	允许执行基本任务（如创建交易）的客户端不能完全验证系统状态。 需要连接到完全节点才能接收系统信息。
	API访问	外部最终用户通过应用程序编程接口（API）连接到完全节点。

通常，系统越开放，就越暴露给恶意行为者。特别是，这些系统容易受到女巫攻击，攻击者会在其中创建大量虚假身份以增加对网络的影响。

女巫攻击是一类恶意行为者通过创建大量虚假身份来获得影响或伪装渎职行为的攻击。

因为身份是一种外生的（即“现实世界”）财产，所以分布式账本技术系统不能通过自身防御这种攻击；它必须依靠外部代理（例如凭证授权机构）或女巫攻击防御机制（例如工作量证明机制或权益证明机制）的干预来减轻这些攻击。

数据广播

数据广播是通过网络向连接节点传输和中继数据的过程。数据可以是原始的和未格式化的，或者是标准化的格式（例如，以交易或记录的形式）。数据可以广播到分布式账本技术系统中的每个节点（通用扩散）或仅与特定的节点子集共享（多通道扩散）。在后一种情况下，数据扩散通常仅限于交易中涉及的交易方或依赖于具体历史交易者；有效地创建一个私人子网络，通常被称为“频道”。这个概念通常被称为分片。

早期的分布式账本技术系统（例如比特币，莱特币）使用通用数据扩散模型，其仍然是主要的广播方法。

为了满足企业的机密性和隐私要求，更新的框架已经实现了多通道扩散模型（例如，Hyperledger Fabric，Corda）。其他如Cosmos，旨在充当“汇集器”，以便独立的分布式账本技术系统可以作为基于应用程序的分片方案的一部分链接在一起。尽管在每个Cosmos子网中技术上都使用了通用数据扩散，但网络间系统类似于多通道扩散。在任何一种情况下，多通道扩散都会阻止节点存储和处理对它们不感兴趣的数据，理论上可以达到更好的规模化。

表7：数据广播配置

配　置	描　述
通用数据扩散	数据被广播到所有节点：汇聚成单一共享记录（全球共识）
多通道数据扩散	数据仅在直接参与特定操作的节点子集之间共享（本地共识）

多通道数据扩散的含义是并非所有网络参与者都需要参与通道达成共识：只需要通道参与者就存储在该通道中的数据达成一致（“本地”共识）。这与具有全球数据扩散的系统显著不同，因为每个单个节点都需要就系统的全球状态达成共识（“全球”共识）；未能通过某些节点子集达成共识可能导致不同意的节点离开，或者分离分布式账本系统（网络分叉）。

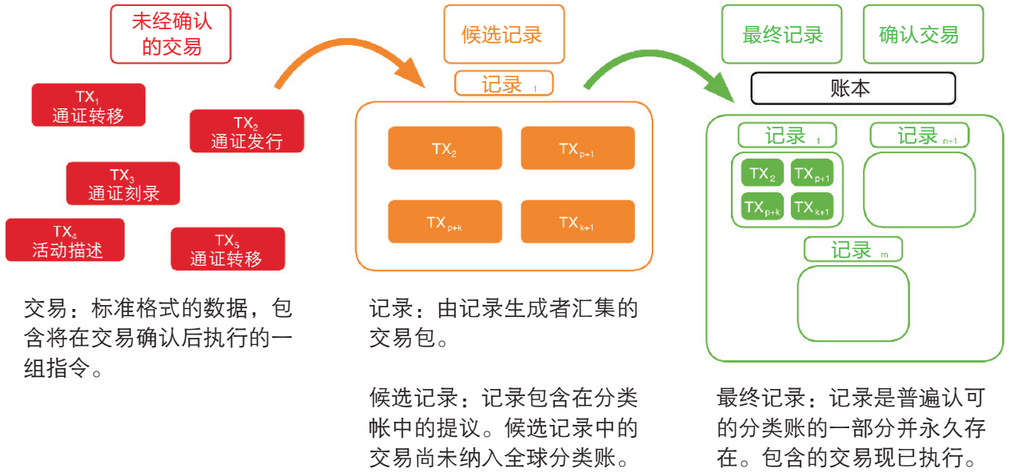
交易启动

交易包含一组指令，一旦需要把交易添加到账本，就要执行这些指令。生成交易可以是不受限制的（即对任何人开放）或限制为选择参与者。用户使用相应的私钥以标准格式签署消息来生成交易。终端用户可以使用不同的接口来创建和向网络广播交易（例如桌面和移动钱包）。

5.2.2 交易处理组件

交易处理描述了将未经证实的交易添加到共享认证记录集所需的一组操作。一旦将交易添加到记录（“已确认”），就认为（临时）结算，这需要执行嵌入在交易中的指令集。但是，单次确认通常不足以为后续交易所依赖；在系统可以依赖的交易输出之前，必须“最终确定”记录。最终性在第5.2.3节中讨论。

图15：概念化分布式账本技术系统中的交易处理



交易：标准格式的数据，包含将在交易确认后执行的一组指令。

记录：由记录生成者汇集的交易包。

候选记录：记录包含在分类帐中的提议。候选记录中的交易尚未纳入全球分类账。

最终记录：记录是普遍认可的分类账的一部分并永久存在。包含的交易现已执行。

记录由分布式账本技术系统用于达成系统状态协议的共识算法所约束。这包括确定建议记录是否有效的过程，以及拒绝无效记录（例如有缺陷或不合规的记录）以及在不同但同等有效的记录中进行选择的过程。

记录提议

记录提议是指记录生产者选择一组未经确认的交易并将它们捆绑在一起以形成候选记录。记录提议可以是无权限的，因为任何网络参与者都有权产生新的候选记录，或者仅允许特定的参与者子集生成候选记录。请注意，这仅指网络参与者，即已被批准加入系统的参与者。

表8：记录提议配置

配　置	描　述
无授权	任何网络参与者都可以创建候选记录
有授权	记录创建仅限于参与者的子集

由于记录受网络共识的约束，因此必须遵守协议规则。在基本级别，它们必须正确地格式化，并且不包含无效或冲突的交易。 此外，每条记录必须包含一个指向前一条记录的参考/指针，如果合适，还包括工作量证明机制或其他女巫攻击防御技术。

共识算法可以根据其难度级别（能耗或财务条款）进行分类。具有无限难度的算法在他们达成共识所需的资源中没有上限。例如，在比特币的工作量证明算法中，找到有效解决方案的难度随着系统额外哈希算力提升而增加。相反，其他算法（例如，实用拜占庭容错算法/ BFT ）不消耗大量资源并且难度有限。

在开放系统中，需要将用于防御女巫攻击的机制合并到共识算法中。受限和封闭系统通常不需要此组件，因为在授予实体加入网络和生成记录的权限之前，通过仔细审查实体来防止女巫攻击。

早期的开放式分布式账本技术系统专门使用工作量证明作为女巫攻击防御机制。工作量证明机制使得生成新记录在计算上很困难（即昂贵且耗时），但是其他人很容易验证它们。 相比之下，新兴的基于权益证明机制的系统使用内部资源（例如原生资产）的赌注来选择下一个记录生产者。工作量证明系统是资源密集型的，但只要参与者的数量很大且分布足够，它们就是健全的。相比之下，权益证明机制系统的资源密集程度低于工作量证明机制系统，但通常也容易受到“无权益”和“磨损”攻击等的攻击。

封闭式分布式账本技术系统通常具有静态使用者体系，因此应该可以对网络中的所有参与者进行完整的概述。它们经常使用诸如轮询调度算法或诸如拜占庭容错算法（ PBFT ），Paxos或Raft之类的算法机制，其中临时选择一个节点作为领导者（即记录生成者）。

共识算法和女巫攻击防御机制是一个活跃的研究领域。有关分布式账本技术系统中使用的各种共识机制的更多信息可以在Seibold和Samman（ 2016 ）中找到。

冲突解决规则

冲突解决规则确定如何解决有效记录的竞争或冲突版本的争议，并取决于使用的共识算法。例如，比特币通过选择具有最大累积工作量证明机制（最长链规则）的分支上的区块来解决由两个相同高度的竞争有效区块引起的临时分裂。Tezos采用权益证明机制的最长链规则，将区块的“权

重”定义为从随机选择的记录生成者那里收到的“背书”数量。其它解决规则包括所有记录生成者的一致同意或通过一定的法定人数阈值。

与所有设计决策一样，每个共识算法都反映了一系列权衡

51％攻击

在分布式账本技术系统中具有大多数“投票”（例如计算能力）的实体或集团产生的记录比网络其余部分更快的攻击。最终，这些记录被显示给网络，导致由于冲突解决规则而替换“诚实”节点的记录。51％攻击是针对分布式账本技术系统的经典攻击。基于工作量证明机制的系统特别容易受到这种攻击；基于权益证明机制的系统中的类似攻击称为“长程”攻击。应该指出的是，在某些情况下，分布式账本技术系统很容易受到不到51％投票权的攻击（例如个人挖矿，理论上只有三分之一的投票权是可行的）。

交易处理的激励机制

激励交易处理是指系统中存在的明确和隐含激励，以鼓励记录生产者通过创建和提交记录来参与交易处理。这些激励可以具有不同的性质（例如货币，法律，社会），并且可以通过协议规则（例如，本地资产中的块奖励）或外部因素（例如参与者之间建立的合同协议）直接表达。 许多分布式账本技术系统使用组合（表9）。

在对分布式账本技术系统进行分类时，这种区别很重要。比特币等开放系统往往通过经济激励设计得到保障，这些设计利用内生网络资源（原生资产）作为经济协调机制来调整激励机制。 从属系统可以使用它们所依赖的系统原有通证。相比之下，具有已知和已审参与者的封闭系统通常通过相互合同约束，依赖于预先建立的授权关系。

表9：激励交易处理配置

	内　在	外　在
货币	区块奖励（补贴+交易费）	付费服务（费用）
非货币	交易生成的需要	合同约束，声誉等

5.2.3验证组件

验证是指确保参与者在认证记录集中独立得出相同结论所需的一组过程。这包括验证未确认交易的有效性，验证记录提议以及审核系统状态。该组件是非分布式账本技术系统的关键区别，因为它为参与者提供了独立审计系统的能力。

交易验证

交易验证包括在将其转发给其他参与者之前验证单个交易是否符合协议规则。这涉及验证交易是否格式正确，是否具有有效签名，并且不与任何其他交易冲突。在某些系统中，交易可能会受到阻碍（例如在满足特定时间或条件之前禁止转移）。这种“累赘”通常是以编程执行的交易（“智能合约”）操作的一部分。

记录验证

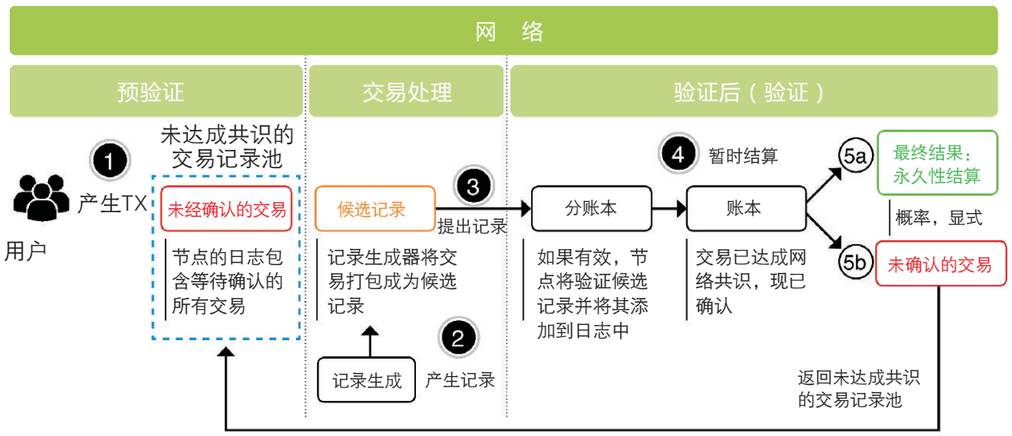
记录验证是根据协议规则验证记录生成者提出的候选记录是否有效。如果审计员认为建议的记录有效，则将其添加到日志并转发给其他节点。虽然确切的过程因系统而异，但通常涉及验证记录中包含的每个交易的有效性和唯一性，以及检查记录建议过程指定的条件是否得到满足（例如，验证是否附有一个有效的工作量证明）。

审计员执行的交易和记录的验证组合提供了从创世起独立计算系统整个状态（全面稽核）的能力。

交易终结

与普遍看法相反，确认的交易或记录不一定是不可逆转的。交易终结性确定何时可以将确认的记录视为最终记录（即不可逆）。终结性可以是概率性的（例如，基于工作量证明机制的系统，在计算上是不可回复的）或显式的（例如，包含必须出现在每个交易历史中的“检查点”的系统）。最终记录也称为永久结算。已经生成但可以恢复的记录暂时结算。在创建记录和终结之间的过渡期之后，暂时结算的记录将永久结算。

图16：分布式账本技术系统中的交易完成过程



暂时结算：在暂时结算期间，可通过替代记录撤销确认记录。在这种情况下，孤立记录中包含的交易将返回到等待包含在新候选记录中的其他未确认交易的日志中。

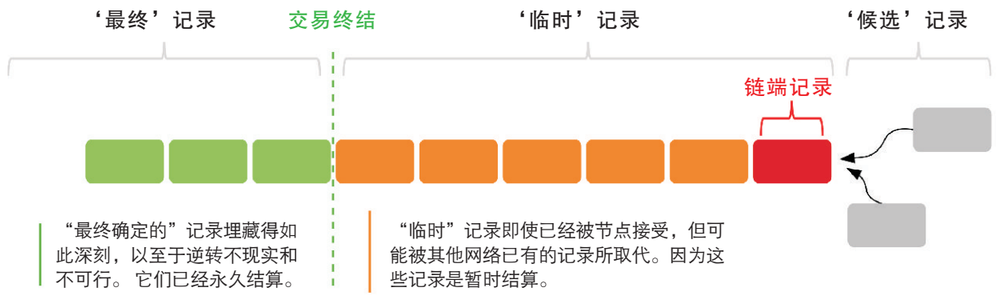


图16提供了结算过程中涉及的步骤的示意图。首先，用户创建交易并将其广播到网络。每个审计员都会验证交易是否符合协议规则。如果认为有效，则节点将交易添加到其日志（也称为“内存池”），该虚拟环境保存等待添加到共享权威记录集的未确认交易。

在交易处理阶段，记录生成者将从他们的内存池中任意选择未经证实的交易，并将它们打包在一起成为候选记录。然后，他们将执行共识机制所要求的步骤，以便将该候选记录提交给网络。节点将审核收到的候选记录及其内容，如果它通过了有效性检查，则记录将添加到节点的日志中。随着交易得到确认和执行，个别日志最终会汇聚到一个共同的账本。

但是，为了替代的竞争记录，可以放弃（孤立）确认的记录：这意味着在暂时结算阶段，确认的交易可以反转 – 在这种情况下，它们作为未经证实的交易返回到日志，等待被列入下一个记录。暂时结算阶段的持续时间取决于系统设计和设置。有些系统几乎立即实现最终结果，而其他系统则具有“概率”终结性，从理论上讲，记录总是可以逆转。然而，在实践中，这种“重组”的可能性随着每个新增记录添加到分类账而迅速减少，因为随着攻击者试图“深入”进入分类账，附加到工作量证明机制挖掘的财务成本会变得过高。只要记录处于临时结算阶段，就不应将其视为“最终”。

通常，用户不会与仅暂时结算的数据进行交互，因为逆转的可能性会产生资产被双花的风险。暂时结算期代表节点的安全系数，有助于确保在用户依赖其输出之前将交易完全合并到分类账（而不是节点的本地期刊），从而防止双花攻击。

一些系统还实施“检查点”以限制“长程”攻击的可能性。在长程攻击中，区块生成器创建竞争子链而不向网络显示记录，然后同时显示所有这些私有记录以使其他节点孤立长期接受的记录。“检查点”是一个诚实的节点永远不会孤立的区块。因此检查点可以限制长程攻击的“范围”。但是，检查点还会在某些条件下（例如“日食”攻击）产生永久性网络分裂的理论风险。

最后，应该注意的是，对协议规则的更改有权改变分类账的形式，这可能直接影响交易最终化。

表10：交易完成配置

最终化	概　率	显　式
暂时	理论上总是; 实际上，时间窗口由网络条件决定	由协议确定的短时间窗口
最终	理论上永远不会; 实际上，在一定的区块深度之后	在通过协议确定的特定区块深度之后

5.3 数据层

分布式账本技术系统的运行规则及方式由协议层决定，并通过网络层实现，协议层与网络层共同决定了数据层的基础。系统使用者可以将实时交易写入账本，数据层会根据每一次记账更新。

5.3.1 操作组件

数据层的操作组件指用户和系统交互产生日志以及衍生账本的过程。

输入

输入指的是分布式账本技术系统的数据源或者获取数据的方法。如章节4.2.2所述，数据源可以分为内部数据源和外部数据源。例如，来自用户与系统交互、内部系统进程引起的状态改变、外部进程（由外部或接口系统初始化的交易）以及智能合约。

一般来说，我们将由用户与分布式账本技术系统在平台上交互产生的记录与交易定义为内部数据源（链上）。同时，我们将其他链下系统与分布式账本技术系统交互产生的结果定义为外部数据源，外部数据源在本质上是可以与核心平台分离的（即4.2.1节图12框架中描述的依赖系统或接口系统）。混合数据源，例如通用状态通道，则允许用户在分布式账本技术系统外运行程序并可以随时向系统传递状态信息，但这一类的技术仍处于初期阶段。

表 11：输入配置

关注点	配　置	描　述
内部数据源	交易	一组用以改变记账状态的加密操作
	记录	一组添加到已认证记录共享集的交易（全球账本）
	自动可执行文件	存在于系统内的程序（或者存在于另一个与本地系统交互的分布式账本技术系统内），且在预设条件达到时会自动执行
外部数据源	传感器	可以向系统广播特定信息的物理设备（例如RFID芯片）
	信息源	收集和整理可以与系统交互的数据的实体（例如一个价格API）
混合数据源	通用状态通道	一种允许用户在分布式账本技术系统外运行程序的交易类型，每一个状态变化代表一个单一的对应者’，最终状态可以随时传送到分布式账本技术系统。

编程可执行交易

除了来自内部数据源或者外部数据源的影响，数据层上的更改还可以来自面向代码的事件，这些事件通过追踪账本上指定状态发生作为条件。一个重要的例子就是编程可执行交易（即“智能合约”）。当编码条件满足时，智能合约会自动执行，并跟随一系列下游事件的发生，例如对账本本身的一些修改。

我们通常将支持开发和执行各种编程可执行交易的分布式账本技术系统称作有状态。用户可以在系统层面构建和运行复杂的编程可执行交易，支持的计算机语言是灵活和通用的，理论上允许编写任何可以想象的程序。

我们将其他仅在基础层支持有限编程可执行交易的分布式账本技术系统称作无状态，这些系统基于简单的脚本语言，仅具有一系列有限的操作代码，允许设计一些相对简单的专用程序。

以太坊（Solidity），Tezos（Michelson）和EOS（WebAssembly）是三个有状态分布式账本技术系统，并且具有图灵完备的系统语言，可以设计复杂的智能合同，它们通常被称为“智能合同平台”。相比之下，分布式账本技术系统比如比特币和Monero只提供支持一种简单的脚本语言允许有限类型的操作。

表12：编程可执行交易配置

类　型	描　述
无状态	系统基础层功能固定，允许执行有限的运算。
有状态	网络参与者通过集成虚拟机在链上执行通用计算。

执行轨迹

执行轨迹决定了交易（例如编程可执行交易）被执行的地点。通常，执行轨迹可以分为在链上（内部）或者链下（外部）。

链上运算是在每个审计员自己的环境（执行引擎）内部执行的。这种环境可以从一个简单的固定用途机器到一个更复杂的拥有丰富图灵完备环境的通用虚拟机（例如，EnthUM虚拟机/EVM）。链上智能合约是由系统内的每个审计员执行的，因此经常被称作‘自我执行’。

链下运算是在系统外部（外部或接口系统）的环境中执行的。尽管链下运算是在分布式账本技术系统上由事件和进程发起的，但执行是“在系统之外”，即相关的工作并不是在分布式账本技术核心系统层上完成的。在这种情况下，外部或接口系统运行核心逻辑交易，分布式账本技术系统则被认作服务外部或接口系统的结算层的功能。

在一些系统中，执行可以在混合侧链中发生。例如，以太坊的Plasma网络通过并行化减轻了节点上的一些计算负担。类似地，Cosmos充当“汇集器”，将其所连接的每个独立系统视为与其协调的较大网络间的“侧链”。

在有限表达能力的无状态分布式账本技术系统中，更复杂的事务逻辑通常被推送到外部或接口系统中并在不同的执行环境中被执行。尽管这种层次方法限制了链上能力，它的优点是减少了基础层的攻击表层，潜在地增强了隐私性、机密性、可拓展性，并使得低延迟要求的应用不被网络延迟所限制。

表 13：执行轨迹

类　型	配　置	描　述
链上	固定用途机器	有限的操作集合
	通用虚拟机	能够执行开放式的操作
链下	协调器	该分布式账本技术系统的目的是发起和控制链下运算
	自动仲裁器	该分布式账本技术系统的核心功能是储存自动可执行文件的结果，结果由链下系统执行。
侧链	子网	侧链运行通常参照各自链上结构相同的分布式账本技术系统，但将计算负荷分配到子网来提高系统的规模化。

分账本组件

引用

账本会随着用户与分布式账本技术系统交互出现，但是，账本本身是抽象的。输入过程和自动可执行文件本身并不直接在账本上运行，而是在日志中运行。节点持有的特定种类的信息和数据结构总是基于某种特定分布式账本技术系统。例如，专注于数字支付的分布式账本技术系统需要保存有关个人用户持有的资产的信息，支持智能合约的分布式账本技术系统必须能够在平台上保存实现智能合约的定制代码。

引用类型

引用数据有四种：内生变量、外生变量、混合变量和自引数据。内生变量（内部引用）用于跟踪关于系统原生变量的信息。例如在比特币中，其中一个内生引用变量被用来跟踪用户在任何特定时间拥有的比特币的数量。当用户向/从其他账户发送和接收比特币时，这个内生变量便会更新。外生变量（外部引用）用于跟踪系统外的变量信息。混合引用则指同时拥有内生变量和外生变量特征的数据。4.2.2节讨论也讨论了这三类引用类型。

第四种引用类型既不是内生变量的也不是外生变量的：这种中性或空数据类型是自引用。例如，智能合约只是一段代码，可以在满足某些条件时执行。虽然智能合约可能需要有关系统外部和内部的数据信息，但代码本身并没有内在地引用自身之外的任何内容（“空引用”）。

表14：引用和值链接的类型

类　型	描　述
内部引用	指的是数据或数字资产，只存在于系统边界内，且不需要连接到外部系统，系统可以自动决定执系统内在的数据和资产。例如 以太坊系统本身的ETH和dApp通证。
外部引用	数据引用的记录是外在的，因此需要连接到外部世界去执行交易。例如，记录系统只记录发生在系统外部的事件和事实。
混合引用	数字资产既有内部引用特征(存在于系统边界内）和外部引用特征（即与外部世界有某种联系）。混合还可以指支持内部引用和外部引用的系统。
自引用	代码段（如智能合同）不引用外生变量或者内生变量，但可能需要内部或外部变量的信息。

第六章　应用框架–案例研究

在本节中，我们将以比特币为案例来说明该框架是如何被应用于分析、表征一个分布式账本技术系统。然后我们将继续对比其他著名的分布式账本技术系统并研究它们的区别在哪里。

本节介绍的所有的分布式账本技术系统是自洽的系统；我们暂不论述独立系统，外部系统以及接口系统。（例如ERC20代币，闪电网络）。

6.1　比特币

比特币的概念最早在2008年10月出现，并于2009年1月正式推出。比特币的目的是创建一个无需依赖可信第三方即可快速结算的数字价值传递和存储系统。

协议

表 15：比特币：协议层

层种类	组件	过程	构　造
协议层	初始	系统间依赖性	自给自足的系统，不依赖于外部系统
		代码库创建	代码库从零开始创建，并且开源
		规则启动	推荐客户端（“比特币核心钱包”）制定规则；不同的实现方式遵守相同的规则集。
	更改	协议管理	无秩序的：通过比特币改进提议（BIP）过程来协调；比特币核心GitHub储存库；
		协议更改	开放式更改：运行软件客户端的选择 (通常是“比特币核心钱包”)

初始

Bitcoin是 一个自给自足的系统，是由以中本聪（Satoshi Nakamoto）为化名的个人或集体通过一个标准客户端的形式发布的开源软件（即“中本聪客户端（Satoshi Client）”，现在被称为“比特币核心（Bitcoin Core）”）。比特币并没有一个正式的协议规范，标准客户端给出了一些规则，而其它客户端也遵循相同的规则（例如Bitcoind、Libbitcoin、Bcoin)。

更改

比特币的管理可以在一定程度上被称为是无政府主义的：没有正式的标准或一套对协议进行更改的标准流程。相反，参与者通过运行某一版本的客户端来实现（并强制执行）他们认为正确的规则集。更改协议需要全球一同努力去说服所有节点更新到支持该更改的客户端版本。

不更新的节点将导致网络分叉，从而有效地创造出新的分布式账本技术系统，新旧两个系统在分叉点之前共享同一个交易历史（例如2017年8月出现的比特币现金（Bitcoin Cash））。因为每个子网络对参与者的价值与该网络上的用户的数量有关，除最具争议性的变化外，系统一般都会鼓励用户在同一时间升级。用户和开发人员可以向承载了相关客户端实现Github存储库申请数据提取。标准客户端“比特币核心”有一个标准化的流程（比特币改进提议，简称BIPs）来探讨协议的更改。然而，只有绝大多数节点决定下载并运行升级后的软件客户端，协议更改才能生效。因此，矿工们在接受相关提议之前会在他们所生产的区块中“标识”出BIPs作为衡量社区情绪的方式。改变比特币的规则是非常困难的，引起争议的隔离见证（SegWit）讨论在2017年8月便直接导致了另一个系统的诞生，即比特币现金。

网络

表 16：比特币：网络层

层种类	组件	过程	构　造
网络层	联络	网络访问	开放：对下载和运行客户端的任何人开放。
		数据传播	通用数据传播：数据在全球范围内传播到整个网络。.
		交易启动	不受限制：任何人都可以用各种方式通过一个节点来提交一个交易。
	交易流程	记录建议	矿工选择未经确认的交易，并创建候选区块。一个有效的候选区块需要将有效的SHA – 256哈希值附加到区块（通过选择一个随机数使哈希结果值足够低）。
		冲突解决规则	节点将采纳最大工作量（最长链机制或者 / 最多工作量证明）的区块链作为合法链。
		交易处理的奖励机制	固有性和货币性：矿工在提交有用区块后，可以获得以本地代币形式（如“比特币”）为主的区块奖励（以及交易费用）。
	验证	交易验证	所有节点在将交易广播到其它节点前会验证所有未经证实的交易。
		记录验证	所有节点在将区块加入到其它日志并传递给其它节点前都会验证工作量证明，区块格式以及区块中的所有交易。
		交易终止	有概率显示：如果一个竞争（更长）的区块链出现，则包含在有效块中的交易则可能会被推翻。一般的经验法则：在交易被确认前需要经过6次确认。

联络

比特币是一种能提供无限制网络访问的开放系统：任何人只需下载和运行软件客户端便可以加入、离开或重新加入网络，而这仅仅受限于其技术能力、设备能力和带宽。在网络中，所有数据都被在全球广播，每一个完整的节点都在其内存池中存储了未确认的交易，以及所有比特币区块链形式的经确认的交易。在交易规则下，任何人都可以提交交易。系统外部的终端用户可以使用钱包软件创建一个交易，并通过另一个通道发送到完整节点上，从而将交易在网络上广播。

交易流程

交易记录

生产者（即矿工）选择内存池中未确认的交易，并将它们捆绑在一起成为候选区块。在向网络提交候选区块之前，矿工需要在候选区块上附加有效的基于随机散列的工作量证明（PoW)。

这个挖掘过程一般需要大量的算力去找到一个有效的解决方案，因为在网络中匹配哈希值是非常困难的。如果两个不相关的矿工在同一时间内找到一个有效的解决方案，网络会根据“最长链规则”来决定两个候选块中哪一个将被接受。矿工每成功挖掘一个区块将会得到系统内固有的收益从而激励矿工处理交易：除了交易费用，成功挖矿的的矿工还会被分配本地的新单元代币（比特币）。

比特币所有节点都会验证交易两次。首先，每个节点在向其它节点广播交易前会检查传入的未确认交易的有效性。这可以防止在网络广泛传播无效交易和占用重要的网络资源。第二步，节点验证包含现在确认的交易区块（交易记录）。如果区块通过有效性测试，则节点将更新其分账本并将该区块广播给相连的节点。比特币的结算是纯偶然性的：从理论上讲，矿工在任何时候都可能重组一个链，从而扭转所有在目前孤儿区块中的交易。然而，在实践中，经6次确认（即受到质疑的区块之上继续挖掘6个区块）之后完成的交易被认为是安全的。

数据

表 17：比特币：数据层

层种类	组件	过程	构　造
数据层	操作	输入	主要的内部（例如以前的输出：UTXOs,脚本） 外部：具有时间戳目的的随机数据（例如通过OP_REtURN）。
		通过编程来执行的交易	固定功能：受限的脚本语言使简单的链上智能合同成为可能（例如多重签名和时间戳合同）。
		执行场所	用于本地资产专业的链上。
	分账本	参考	内在的：系统特有的本地资产（BTC）。

操作

比特币主要以内部资源作为产生新纪录的输入，主要是因为这个系统的主要目的是为本地代币提供安全的价值转移。这意味着内部资产单位（“比特币”）成为交易的输入。比特币也可以作为一个全球性公证产品：数据（或指向外部存储的数据）可以嵌入到比特币交易以实现时间戳防篡改。在自动可执行文件（“智能合约”），比特币只允许通过本地脚本语言（“脚本”）来设计和执行相对简单的程序：多重签名和哈希时间锁定合约是在比特币上实现的简单智能合同的经典例子。

分账本

该系统交易记录的交易描述了本地代币比特币的创建和传输。比特币作为比特币分类账中的条目只存在于比特币系统的边界内。比特币系统产生的交易记录指向了与外部系统没有直接联系的内部值。这意味着比特币系统交易记录的传输不依赖外部代理来执行，而是由网络参与者自动和独立地在链上执行交易。

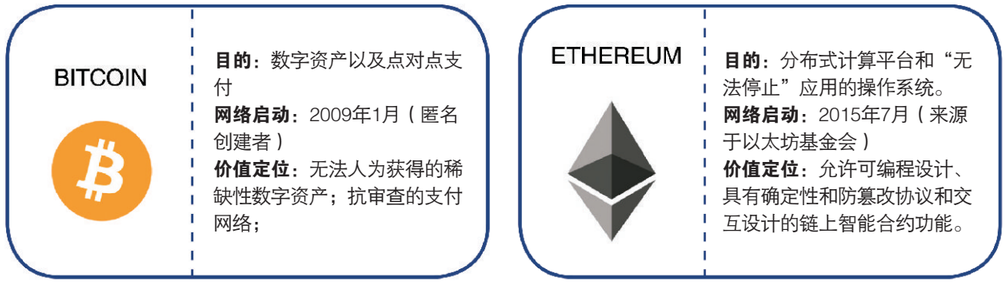
6.2 比较分析

我们开发的框架是一种在不同分布式账本技术系统中处理系统层、组件、进程和参与者之间映射关系的工具。这些系统是动态的和不断发展的，分析也需要频繁地更新。我们通过以下的比较分析尽可能地去描述在报告发布时所指定系统的状态。

6.2.1 案例研究

我们总共选择了六个案例研究来展示框架如何用于比较分析（图17）。每个所选择的系统都具有独特的性质,和受假设与评估影响的设计选择（配置选择）结果所决定的属性。

图17：案例研究概述



ALASTRIA

目的：为西班牙企业和公共部门机构所用的多种类区块链结构。
网络启动：测试网络于2017年12月启动，主网络将于2018年第四季度启动（来源于“Consorcio Red Alastria”协会）。
价值定位：共享链上管理和传输的半公共网络。

RIPPLE

目的：跨境支付网络
网络启动：2012年（来源于OpenCoin，现在为瑞波实验室）
价值定位：快速低成本的跨境支付和货币转换

VERIFIED.ME

目的：可信任的身份助于线上注册。
网络启动：于2016年底进行首次试启动；预计将于2018年第四季度正式启动（来源于SecureKey）
价值定位：为用户提供低成本、高信任度的私人、安全、方便的在线注册服务。

‘PROJECT X’*

目的：全球贸易可共享的基础设施
网络启动：2017年9月（来源于公司Y）
价值定位：通过减少欺诈、失误和协调成本来提高全球贸易的自动化的效率

注意：由于该项目的发起人不愿意公开，因此本节中将其称为由“公司Y”负责发起的”项目X”.

下面的章节只会突出案例研究之间最重要的区别。完整的使用框架比较分析见附录B。

6.2.2 这些是分布式账本技术系统吗?

回想一下，我们对分布式账本技术系统的正式定义需要5个要素。表18总结了我们在案例研究中所包含的系统是否满足这五个要素。我们将从技术层面讨论致使各个系统与比特币不同原因，并作为我们的框架的一部分分。最重要的是，请注意，尽管我们所纳入的系统都已启动（“启动日期”），但并不是所有系统均具有作为分布式账本技术系统的功能。然而，所有这些被纳入的系统都已经发布可以明确指导它们如何获取分布式账本技术系统所需性能的计划。我们把这些系统称为潜在的分布式账本技术系统。

“潜在的”分布式账本技术系统是什么？

值得注意的是，并不是所有的自称是分布式账本技术的系统都可以被认为是根据第3节中给出的标准定义的分布式账本技术系统。特别是Verified.Me和“X项目”–他们所处的状态–不满足定义给出的所有条件。然而，因为他们的结构足以引导这些系统满足所有必要的条件，使得他们有潜力成为分布式账本技术系统。

表18：并非目前所有的案例研究都符合分布式账本技术系统标准

	启动时间	共享记录	多部分共识	独立验证	篡改证明	防篡改
比特币	2009年1月	✓	✓	✓	✓	✓
以太坊	2015年7月	✓	✓	✓	✓	✓
瑞波币	2012年	✓	?	✓	✓	?
Alastria	2018年(测试网)	✓	✓	✓	✓	?
Verified.Me	2016年(试运行)	✓	✓	✓	✓	?
项目X	2017年9月	✓	×	×	✓	×

比特币和以太坊均满足分布式账本技术系统所需的五个属性。瑞波实验室对验证节点的影响使得多方共识和抗篡改特性备受争议。Alastria与Verified.Me的防篡改特性也显得不那么清晰。这些有争议的性质导致三个系统具有争议的分布式账本技术状态——有些人把它们看作分布式账本技术系统，有些人则不认为——但是，我们将他们纳入是因为他们可以满足我们的框架分析条件。项目X仍处于早期阶段，只使用一个验证器，还没有多方共识、独立验证或防篡改的特性，但它确实有一个明确的计划来增加交易记录生产者和独立审验者的数量，因此最终将成为成熟的分布式账本技术系统。

使用该框架来分析和比较尚未具有完全分布式账本技术功能的系统的能力是基于系统的方法的另一个益处。

6.2.3 协议

系统启动

比特币是所有案例里由匿名者发布的开源项目中唯一的分布式账本技术系统。相比之下，尽管基于不同的结构，其他所有的案例研究是由一个已知的实体发布：例如，以太坊由以太坊基金会发布，Alastria主网由“Consorcio Red Alastria”协会发布，而瑞波，Verified.Me 和“X项目”都是由独立公司推出（分别是OpenCoin / Ripple Labs，SecureKey，和“Y公司”）。

代码库

比特币、以太坊和瑞波都是源于基本代码的设计，而Alastria是建立在一个称为Quorum的结构之上（最初由J.P.摩根开发的），是一个以太坊的衍生版本。Verified. Me和“X项目”使用超分类结构框架（图18）。

图18：代码库对比

比特币、以太坊、瑞波和Alastria都是开源的：这意味着网络参与者可以决定分叉项目（即“复制粘贴”代码库），并创建一个有着相似前提的替代系统。相反，Verified. Me和“X项目”闭源，阻止参与者进行系统的克隆。

治理

系统治理是分析分布式账本技术系统差异的关键因素之一。完全开放和无需许可的系统如比特币和以太坊在协议规则更新方面缺乏的形式化的流程和标准。然而，这两个项目就如何处理这一问题上也存在差异：比特币的参考客户比特币核心有一个专门的BIP（比特币改进提案）流程，通过该流程提交、审查、接受或拒绝代码库的更改。虽然理论上BIP的访问对任何人都是开放的，实践表明极少数的核心开发团队的志愿参与给协议变更所带来不成比例的影响。然而隔离见证的案例已经表明在建议的规则变化上达成全球共识是非常困难的。

另一方面，以太坊的发展很大程度上受到以太坊基金会及其发展路线规划的影响。虽然拥有一个EIP（以太坊改进建议）的流程，且用户可以从多个客户端中来选择，但历史已经表明，有一个著名的例外，由以太坊基金会以及其联合创始人Vitalik Buterin提出的升级已被系统参与者毫无争议的接受。

图19：管理对比

在封闭的系统中，管理员通常在管理过程中发挥更重要的作用。例如，“X项目”中协议的改变是由关键客户（一个大型物流公司）分配以及“Y公司”来实施。然而，也存在可替代的更协同的多种模式选择：在Alastria中协议的变更取决于交易记录产生者（验证器），而在Verified.Me中取决于由网络参与者组成的一个指导委员。相比之下，即使验证器可以在所谓的“修正案”中获得决定权，Ripple的关键决策还是由作为“仁慈的独裁者”瑞波实验室来决定。

表19：为基于框架定义的协议层处理过程中出现的主要差异提供了更详细的分析

框架元素			比特币	以太坊	瑞波币	ALASTRIA	VERIFIED.ME	“项目X”
层种类	组件	过程						
协议层	初始	系统内依赖性	自给自足系统	自给自足系统	自给自足系统	自给自足系统	自给自足系统	自给自足系统
		代码库创建	起源创始块；开源；	起源创始块；开源；	起源创始块；开源；	基于Quorum的代码库（以太坊的分支）；开源；	超级分类账；封闭源用户模块；	超级分类账；封闭源；
		规则启动	规则由推荐客户端制定	正式规范的协议（“黄纸”）	规则由推荐协议和推荐客户端制定	正式规范的协议	正式规范的协议	正式规范的协议(默认是超级分类账的推荐)
	更改	协议管理	无政府：节点通过运行软件客户端的选择来投票；但比特币核心钱包参考实施对发展有重大影响。	无政府：以太坊基金会和个别开发者对于总体发展具有重大的影响。	专政：瑞波实验室几乎拥有终极控制权，验证器可以通过“修正”特性进行投票。	民主的：验证节点需要达成一致，没有中央管理者。	联邦：由投票人组成的指导委员会投票	专政：关键客户拥有最终权力

框架元素			比特币	以太坊	瑞波币	ALASTRIA	VERIFIED.ME	“项目X”
层种类	组件	过程						
协议层	更改	协议更改	依靠核心GitHub存储的比特币改进提议(BIP)；运行软件客户端的选择（通用的比特币核心钱包）	以太坊改进建议（EIP）；运行软件客户端的选择（分开或平等）	验证器通过投票赞成“修正”－如果80%同意，更改可以进行实施。	不清楚的：可能由可以选择是否新的客户端来决定。	技术更新由网络端和客户端决定；实质性规则变更需要通过正式的变更管理过程来执行。	公司Y将更新为客户运行的客户端。

6.2.4 网络

网络层显示了所分析系统之间的多种重要的差异。

网络接入

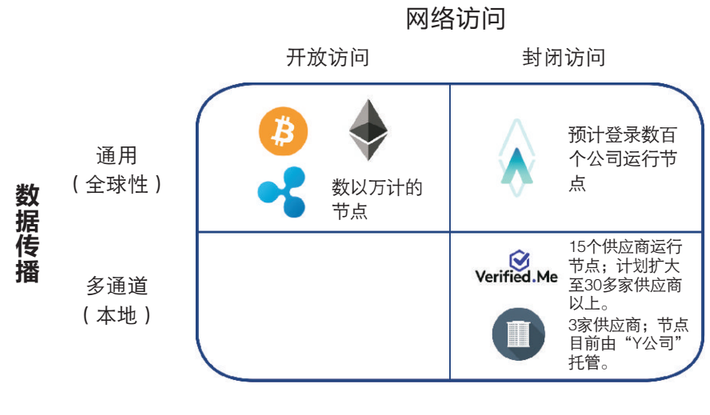
虽然比特币、以太坊和瑞波的网络是普遍可访问的（前两个拥有数千个验证器），但对Alesta、Verified. Me和“X项目”的访问仅限于选定的参与者（图20）。Alastria对任何西班牙业务开放，受半开放式应用程序的影响，验证程序投票决定是否接受新成员。Alastria预计将登陆数百家公司。相比之下，对Verified. Me和“X项目”的访问由单独的网关控制。Verified. Me有大约15个供应商运行所有已验证节点，而“X项目”仅限于三个供应商，所有的节点目前由“Y公司”托管，并且可以通过API调用访问。

图20：网络访问对比

联络

在比特币、以太坊、瑞波和Alastria中，数据被全球广播到网络中的所有节点，这意味着每个节点必须存储和处理从一开始发生的每一个交易。另一方面，Verified. Me和“项目X”是基于支持本地多通道数据传播的超分类结构框架：数据仅在特定信道（即子网络）的参与者之间共享。

图21：网络结构对比



交易处理

当涉及到如何将交易以交易记录的形式包含到分类帐中时，我们所讨论的系统将采取不同的方法。

参与：

在第一步中，我们观察到一旦登陆网络，比特币、以太坊、瑞波和Verified.Me中的任何网络参与者都有权作为交易记录生产者（无需许可的）参与交易处理。这与Alastria 和“X项目”形成对比，其中只有选定的部分网络参与者被授权成为验证者（已许可的）。Alastria将启动大约30个不同的验证器，而“X项目”目前只使用一个验证器，但计划在未来将控制权逐步分发给多个验证器。

交易记录创建和冲突解决：

在交易记录处理方面，这些系统之间存在着明显的差异：在比特币和以太坊中，矿工彼此竞争，并将一个有效的随机散列的工作量证明附加到他们的候选交易记录上。在有两个竞争交易记录的情况下，引入“最长有效链”法则，判定累积最多的随机散列的工作量证明的分类帐版本是权威的。

相反，其他系统使用较少的资源密集型共识机制来实现分类帐的状态：瑞波使用多个共识回合

直到达到“绝对多数”的80%为止，而Alastria计划在第一阶段启动一个相当传统的筏式共识机制。“X项目”因为在这个阶段只有一个验证器所以完全不使用分布式共识机制。因此，在“X项目”实现一个包含多个验证器的共识机制前，我们不能称其为分布式账本技术系统。

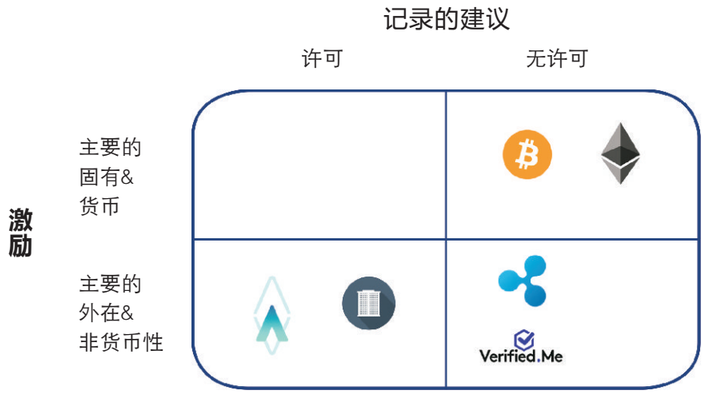
激励

比特币和以太坊是通过经济激励得以保障：矿工们拥有区块补贴形式的内在货币激励（例如新崛起的本地代币单位）和交易费（以原生代币计价）。这些系统中交易记录生产者是在比特币白皮书中描述的的经济激励的基础上执行操作的：“他应该发现，遵守规则比破坏制度和自身财富的有效性更有利可图。”

因此，交易记录生产人专注于自身经济利益以确保系统顺利的运作。

相反，其他系统中的交易记录生产人主要具有非货币性质的外在激励，如声誉（例如被认为是负责人的），可靠性（例如提供良好的服务）和威胁诉讼，以防他们不遵守规则。Verified.Me验证器还具有外在的货币激励，因为它们可以获得提供服务的费用（以本国货币计价）。这些系统中的安全性主要基于访问控制和交易记录生产者之间的合同义务。

图22：交易处理对比



验证

验证是一个至关重要的方面，其可为个人审查者提供独立的、无需依赖第三方系统去验证交易记录的能力。由于比特币，以太坊，瑞波和Alastria使用全局数据传播模型，每个审计员都必须验证和存储曾经生成的每个交易和交易记录。另一方面，在Verified.Me和“Project X”中的审核员使用多渠道数据传播模型，只需要验证其渠道内的交易和交易记录（本地验证）即可。在“X项目”中，由于所有节点当前均由Y公司托管，因此没有正式的独立验证。

B比特币和以太坊只提供概率性的终结：由于用于交易处理的工作量证明机制，确认的交易有可能在任何时间点发生反转。然而，在实践中，随着每个附加交易记录添加到分类账，反转的可能性会降低，因为重新组织分类帐需要为所有后续块重新执行整个工作量证明。因此，一个常见的经验法则是在超过6个（比特币）和24个确认（以太币）之后再考虑交易记录的“准最终”标记。例如在考虑纪录上的额外纪录。瑞波，Alastria，Verified.Me和“X项目”具有确定性的终结，这意味着在特定的临时结算阶段后，交易记录可被视为最终交易记录。即使所有相关案例研究都声称在交易记录确认后立即清算，临时清算的期限通常因系统而异。

表20总结了在网络层的六个案例研究中的比较分析。

表20：比较分析:网络层

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
网络	联络	网络访问	开放和自由	开放和自由	开放和自由	半开的：守门分布在验证器节点	关闭：访问控制执行看门人(SecureKey)	关闭：访问控制执行的看门人Y公司按照正式规定过程
		数据广播	通用数据扩散(公共)	通用数据扩散(公共)	通用数据扩散(公共)	通用数据扩散(公共)	多通道数据扩散（选择性的隐私）	多通道数据扩散（选择性隐私）；但所有节点由公司托管和运营
	事务处理	事务初始化	无限制：任何一个有相应的私钥可以创建并签署事务：需求通过an向网络广播审查器 / 监听器，特殊目的人或第三方方服务(API)	无限制：任何一个有相应的私钥可以创建并签署事务：需求通过an向网络广播审查器 / 监听器，特殊目的人或第三方方服务(API)	无限制：任何一个有相应的私钥可以创建并签署事务：需求通过an向网络广播审查器 / 监听器，特殊目的人或第三方方服务(API)	网络参与者可以创建事务；可能外部用户可以发送签署的交易也通过一个审计师 / 侦听器	受限：选择终端用户集通过API 触发事务节点	限制：关键客户的ERP系统创建事务通过API提交审计/听众之一由Y公司
		记录的建议	无许可：矿工选择未经证实的交易从他们mempool，将它们捆绑在一起成一个候选人。一个有效的候选块需要附加块的有效SHA-256散列标题（通过选择一个nonce使哈希值足够低值）	无许可：矿工选择未经证实的事务他们的mempool，并捆绑他们一起组成一个候选区块。一个有效的候选块需要将有效的Ethash PoW附加到块标题(通过选择nonce)这样哈希值就足够低了值)	许可：验证器选择未经证实的交易和创建一个新的分类帐实例。他们接力赛候选人的“一轮”记录共识：multi-computation新分类	许可：验证器节点（±30种不同实体）选择未经证实的从他们的交易池，包在一起成一个候选人块。Raft-based共识机制来达到协议	许可：验证器节点（所有15个提供者）创建记录包含有关的事务他们只参与交易在。简单的状态改变的建议:一般没有分歧	许可:公司Y-controlled验证器选择未经证实的交易和创建记录(中央集权节点)

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
网络	事务处理	解决冲突规则	最长的有效链规则（即大多数累计PoW）	最长的有效链规则（即大多数累计PoW）	多个共识轮次唯一节点列表（UNL）直到“绝对多数”（80%）达到共识	种族：第一块赢，竞争块被丢弃（很少见一般只有一个月/一次领导）	一般没有争议；所有参与者都同意发生了一些事情。使用的精确一致性算法未知	空操作 (共识忽视)：没有冲突可能
	验证	激励事务处理	内在和货币：区块奖励（新成立的BTC和交易费用）	内在和货币：区块奖励（新兴的ETH和交易费用）	没有金钱奖励，含蓄的外在激励（网络稳健性和弹性）	没有固有的货币激励（没有本地令牌）	（1）外在货币激励：提供者获得服务费目的地服务，以本国货币计价；（2）外在的非货币激励：（a）为客户创造价值，帮助他们的提供者与GAFA竞争；（b）帮助他们减少欺诈	没有内在和货币激励–外在非货币激励(9) 平台顺利运行
		交易确认	审计人员和听众都会对未经证实的交易之前将其中继到连接的节点进行验证	审计人员和听众都会对未经证实的交易之前将其中继到连接的节点进行验证	跟踪节点验证以前未经证实的依赖他们的交易	审计师和听众验证每个未经证实的转发前的连接到节点的交易	每个审核员 / 侦听器都会验证通道中的每笔交易	公司Y控制节点，验证每笔交易发生在一个特定的渠道

6.2.5 数据

操作

比特币的输入通常具有内部性质（例如先前的输出和脚本），而以太坊，瑞波和 Alastria 的输入来同时具有内部和外部资源。 Verified.Me和“X项目”主要来自外部连接系统的外部输入。

以太坊和Alastria是两个支持通用链上计算的系统，可用于直接“链上”（表达）设计和运行复杂的协议和程序。应用程序和一般程序将在系统级自动执行 – 通过所有完全验证的节点（全局数据扩散）或参与该特定协议的那些（多通道数据扩散）。

相比之下，比特币，瑞波，Verified.Me和“X项目”具有相当有限的'链上'计算能力（规定）。 这些系统没有集成运行时环境和虚拟机（VM），这意味着表达式程序不能直接在系统级别执行。然而，更复杂的计算通常在连接但外部的系统中处理和执行。 这种分层方法可以在更具表现力的系统上提供某些优点（例如，更好的扩展性，增加的隐私性，更高的安全性）。

参考和价值链接

比特币，以太坊和瑞波都跟踪只存在于其系统边界内的内生系统变量：原生数字资产（分别是比特币/ BTC，以太坊/ETH和瑞波币/ XRP）。 由于这些资产是系统固有的，因此系统交易记录的所有权转移可以由系统自动有效地执行，而无需外部代理的干预。

相比之下，Verified.Me和“X项目”专门用于跟踪引用系统外部资源和事件的外生系统变量。例如，Verified.Me中的交易记录包含指向存储在外部专有数据库中的身份数据的哈希指针，而“X项目”则跟踪外部ERP系统中存在的保险交易记录。

除了管理其本机数字资产之外，以太坊和瑞波还可用于创建在系统级别引用外部资源的交易记录。一个例子是由瑞币网关发布的Ripple IOU，并作为国家货币的数字表示，由网关保管。涉及IOU的交易引用外部系统中持有的国家货币，这需要外部代理和离线流程来强制执行“现实世界”中的转移。 因此，在交易记录值链接方面，以太坊和Ripple可以被认为是混合的。

由于Alastria尚未正式上线，我们无法确定最终会参考什么样的交易记录。 但我们可以大胆的假设其交易记录方式类似于以太坊，网络参与者将充分利用平台的多功能性并创建引用内生和外生对象的交易记录。

表21概述了数据层中每个案例研究的配置。

表21：比较分析：数据层

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
数据	操作	输入	主要是内部的（例如以前的输出：UTXO，脚本）。外部：任意时间戳的数据使用OP-的目的返回	内部（例如以前的输出：账户、智能合约）以及外部（oracles）	内部（例如以前的输出：账户）以及外部（与建立IOU有关的数据）	内部（例如以前的输出：账户、智能合约）以及外部（oracles、IPFS启用、链下隐私存储）	基础外部(开放ID连接(OIDC)：连接服务协议)。内部 = 以前的输出：哈希负载；赞同指令；接收证明	基础外部（依靠API的）：连接服务协议。内部 = 以前的输出：哈希负载；赞同指令；接收证明
		通过编程来执行的交易	无状态的：有限脚本语言实现多重签名和时间戳合同	在状态的：图灵完备的智能合同允许多用途计算	无状态的：链上具有专用目的的基础计算	在状态的：图灵完备的智能合同语言允许通用计算	无状态的：链上可以实现非常简单的业务逻辑	无状态的：链上可以实现非常简单的业务逻辑
		通过编程来执行的交易	用于运行链上简单脚本的固定功能机器	通用虚拟机：以太坊虚拟机（EVM）	用于基本链上运算的固定功能机器	通用虚拟机：以太坊虚拟机（EVM）允许在链上执行复杂的计算	用来管理用户赞同指令的业务逻辑可以在更高层 (链下) 执行	业务逻辑可以在外部平台（链下规则引擎）执行

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
数据	操作	执行场所	内部的：系统特有的本地资产（BTC）。	（1）内部的（本地资产：ETH；用户定义的代币：dApps）；(2)混合的（无标记的代币和／或外部事件的纪录。	（1）内部的（本地资产：XRP）；(2)混合的（网关发布的IOU和可信任线路）	依赖于使用的案例：如果是本地资产或用户定义的代币则属于内在的。如果是外部与内部的结合则属于混合的。也有可能是完全外部的。	完全外部的：身份数据属于外部系统（如身份来源：政府，银行等）	完全外部的：ERP系统和保险纪录
	分账本	参考	无状态的：有限脚本语言实现多重签名和时间戳合同	在状态的：图灵完备的智能合同允许多用途计算	无状态的：链上具有专用目的的基础计算	在状态的：图灵完备的智能合同语言允许通用计算	无状态的：链上可以实现非常简单的业务逻辑	无状态的：链上可以实现非常简单的业务逻辑

6.3 比较分布式账本技术系统案例研究中的差异

6.3.1 总结框架结果

图23：案例研究之间的主要差异概述

		₿	Ξ	ripple	ALASTRIA	Verified.Me	'Project X'
治理监管	无政府的	✓					
	分级		✓				
	专政			✓			✓
	联邦				✓	✓	
网络访问	开放式	✓	✓	✓			
	半开放式				✓		
	封闭式					✓	✓
交易处理	分散	✓	✓				
	半集中式			✓	✓	✓	
	集中						✓
激励措施	内在	✓	✓				
	外在			✓	✓	✓	✓
参考	内源性（原生）	✓					
	混合		✓	✓	✓		
	外源性					✓	✓

每个选定的案例研究都试图服务于不同的用例和目标，从而产生各种各样的架构和设计决策。像比特币这样的系统正在优化信任最小化和审查阻力，这需要在所有层和流程上进行合理的分散。

这需要以例如性能，吞吐量，速度，扩展和用户体验为代价。此外，缺乏集中治理和决策使各种网络行为者之间的协调变得复杂，减缓了集体行动。

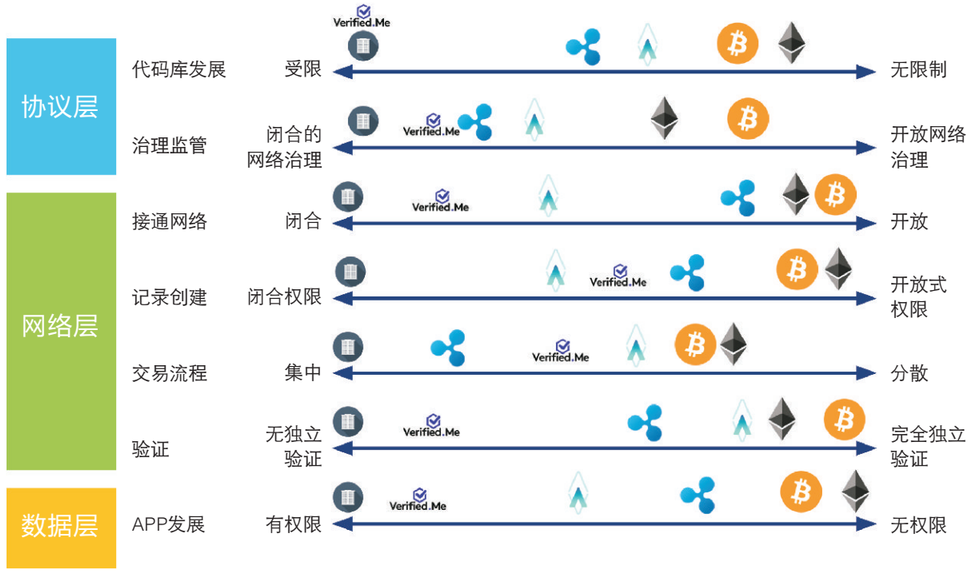
Verified.Me和“X项目”等系统设计用于在受监管的多企业环境下不同的情景中运行。这允许他们选择不同的权衡方式并采用更灵活的方法，代价是更集中的协议和网络层。“X项目”在自举阶段采取了一种非常保守的方法，从一个系统开始，其中每个功能都集中在“将脚趾放入水中并让人们登上船”。随着时间的推移，逐步将控制权交给更多的参与者。该决定背后的原因在于更好地了解在安全的环境中操作时系统的功能和属性：这使参与者获得宝贵的经验和见解，然后可以用来逐步向前发展。与以太坊等开放式实验相比，其原理在于缓慢行动而不是破坏事物。

6.3.2 参与方式的差异

图24显示了如何使用不同组件评估分布式账本技术系统中的“参与”可以产生更细微的最终结论。参与度最高的系统是比特币和以太坊。在目前阶段，“X项目”有目的地选择限制参与来引导系统并在封闭的环境中开始一个学习过程。在剩下的三个系统中，Verified.Me在协议和数据层中的参与度最低，在网络层中不太确定。

在对高度复杂和动态的系统进行比较分析时，选择准确的“关注点”是至关重要的。未能考虑到分布式账本技术系统的多样性可能导致不完整的结论和评估。因此，我们建议使用多个关注点以获得更广泛的 – 因此可能更完整和准确 – 的画面。

图24：“参与”的不同级别



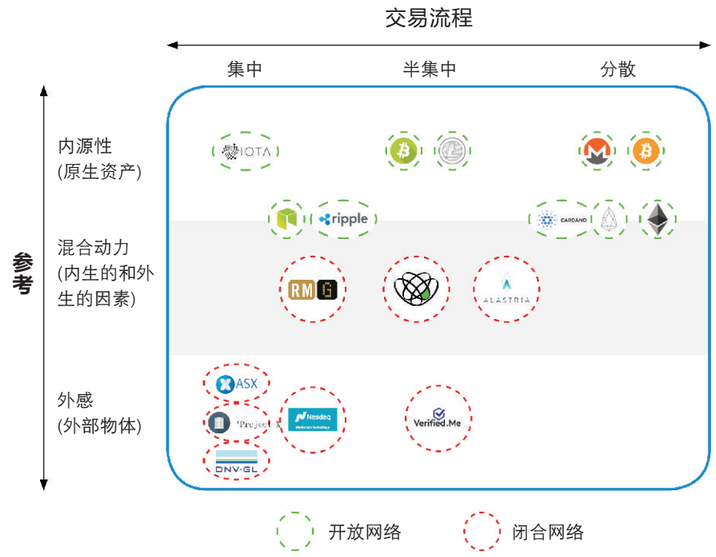
6.3.3 探索当前分布式账本技术系统蓝图

图25展示了通过三个维度映射选定的分布式账本技术系统的蓝图概览。交易处理指在选择交易和向全局分类账添加交易记录方面的集中程度。

参考确定系统参与者产生的交易记录是纯粹是内部的 – 内源的（例如原生数字资产），还是完全外部的 – 外生的。后者指的是专门用于交易记录保存目的的分布式账本技术系统（即跟踪系统外部的信息，例如供应链中的项目）。附加类别表示混合资产（例如，标记化形式的物理资产），其共享内生和外生属性的组合。

网络访问决定了它的分布式账本技术系统级别的可访问性：访问权限可以不受限制，对任何人开放或仅限于必须经历特定选择过程的选定实体组。

图25：当前的分布式账本技术系统架构



可以从下面示意图中得出两个主要观察结果：

开放网络都需要本地资产（通常称为加密货币），该资产被用作经济协调机制，以协调系统参与者的激励措施，以实现共同目标：本地资产在激励交易记录生产者处理交易方面起着至关重要的作用。越来越多的开放式分布式账本技术系统也用于引用非本地资产（例如Ripple，Ethereum，Cardano）。

另一方面，目前封闭的分布式账本技术系统 – 除了一些值得注意的例外（例如World Reserve Trust，Royal Mint Gold） – 主要用于交易记录保存目的，以跟踪和交易记录外部信息。 因此，这

些系统不能单独执行执法；它依赖于外部代理商。

具有开放网络的分布式账本技术系统在从完全集中的交易处理到几乎分散的交易处理的整个范围内操作，而大多数封闭的分布式账本技术系统当前在交易处理方面以更集中的方式操作。这并不令人意外，因为企业在生产中部署新系统时倾向于采取更谨慎和保守的方法。

6.3.4 关键设计决策和启示

图26总结了导致不同分布式账本技术系统的关键设计决策。每种配置都可能对系统的特性，属性和性质产生特定的影响。此外，特定配置的组合可以对二阶效应的水平产生额外的影响：这些可能难以预测，因为它们通常仅在发布后表现出来。

图26：对决策的关键设计和影响

第七章　结论

7.1 总结

比特币出现快10年了，但分布式账本技术（ DLT ）的生态仍处于早期阶段。它仍在不断发展、试验及研发的过程中，新的系统、应用及落地几乎每天都有，大胆的构思、技术假设及漏洞也经常被提出。尽管近年来人们对分布式账本技术的兴趣日益增长且该技术本身也取得了很大进展，但分布式账本技术仍通常被认为是相对不成熟。 目前人们对分布式账本技术系统在成本和效益方面的得失将信将疑，对使用该技术来解决一些具体问题有没有效果也持怀疑态度，这些疑惑使围绕分布式账本技术无处不在的炒作蒙上阴影。

因此，存在很多对分布式账本技术系统属性、特质、特定场景下适用性、尚存技术挑战的误解。人们存在许多误解。有些人在开放的、公共的、无使用许可要求的分布式账本技术系统上发布了他们自己杜撰出来的数字资产，而这些数字资产主要是用作投机工具。除了这些投机性的数字资产，迄今为止， 分布式账本技术系统真正有意义的应用和落地几乎没有。大多数项目仍处于早期试验或验证阶段，我们也不清楚这些项目什么时候会成熟或能否活下去。“区块链”和“分布式账本技术”已成为几乎毫无意义的流行语，在许多情况下，这些词语主要用于营销和公关目的。

基于以上原因，我们提出任何一个分布式账本技术系统必须具备的五个关键属性或标准，但同时这五个关键属性对一个分布式账本技术系统的架构没有或几乎没有任何要求。 任何分布式账本技术系统都是一个分布式的交易记录保管系统，这个系统可以在一个敌对的、没有信任的环境中运行，并可以由多方参与者共同维护和更新； 在任何一个分布式账本技术系统内，每个参与者都需要能够独立地验证所有交易的有效性和完整性，并最终验证整个系统的状态；任何篡改交易历史的企图，都需要检测起来非常容易，但做起来非常困难。

我们发现，与一个分布式账本技术系统必须具备的五个关键属性对比，许多自我声称的“分布式账本技术系统”，根据它们的系统构成和配置，并不符合这些标准。这些自我声称的“分布式账本技术系统”充其量只能被视为具有分布式账本技术基本架构特征的“潜在的分布式账本技术系统”，他们或许最终能演变成“纯”的或真正的分布式账本技术系统。 为了能够分析任何一个分布式账本技术系统的关键属性和权力的分布，我们提出了一个概念框架，就是把任何一个分布式账本技术系统分解为三层， 每一层都有一组组件，这些组件又由某几个过程组成； 这些组件和流程相互作用，而一个流程如何设计会很大程度上影响到相关的流程以及最终的系统属性；每一层都有不同的参与者，这些参与者扮演不同的角色类型以及发挥不同的作用。

我们将该框架应用于六个案例研究来分析这六个案例中所涉及到的系统的共性和差异。这六个系统都是已经在用的真实的系统：比特币，以太坊，Ripple，Alastria，Verified.Me和'Project X'。我们的分析表明，每个系统设计的选择，都是在多个分布式账本技术系统的属性之间进行权衡和取舍；最常见的权衡和取舍是在“权力分散化”和系统绩效之间的取舍。 因此，每个分布式账本技术系统都是多种系统属性及配置选择而合成后的独特结果，要具体系统具体分析。

7.2 研究的贡献

本报告核心问题是，分布式账本技术系统是什么，我们如何从根本上来理解以求大家能达成共识。为此本报告提出一个分布式账本技术系统的正式定义，这个定义包含一系列任何一个分布式账本技术系统都应该满足的标准。 此外，我们也提出了一个概念框架，把任何一个分布式账本技术系统分解为由多个关键层、组件和流程来组成。

该框架有四个目的：

- 如何识别分布式账本技术系统
- 如何分析现有的分布式账本技术系统
- 如何比较不同的分布式账本技术系统
- 通过指出不同设计选择的得失，为新系统的设计提供有用的工具

本报告主要的贡献在于提供了一个概念性工具，可以用于研究分布式账本技术系统是由哪些部分组成的，并理解这些组成部分的依赖关系：在查看分布式账本技术系统上的资产、通证或记录的交易信息之前，我们需要了解前者所基于的基本结构。 系统不仅仅是“分权的”或“集权的”，或简单的非白即黑，相反，在分布式账本技术系统的每一层、每一个组成部分或流程，存在着不同程度的控制和权限。 因此，了解这些层与层之间的依赖关系至关重要，这使得我们可以精确评估在分布式账本技术系统的结构里，每一个分布式账本技术层上都有什么。

对监管机构而言，该框架提供了一个清晰的图景，说明分布式账本技术系统中的权限（如果有的话）在哪里，以及谁可以对最终的技术和结果负责。 例如，在很多场景下，协议层通常由具有修改游戏规则的能力的中央权力控制。 在其他场景下，验证和处理交易的权力可能仅限于单个实体 – 或者仅限于一小组密切相关的实体。这就是说，了解权力如何动态的在分布式账本技术系统中的不同层之间分布很重要。 除了促进对分布式账本技术的一般理解，该框架还确定了分布式账本技术每一层的参与者中有哪些会受到监管部门的监管。

对于打算自己动手开发这些系统的企业、系统工程师和开发人员而言，该框架可用作指导他们，哪些方面是开发新的分布式账本技术系统所必需的。

对于希望了解分布式账本技术企业的投资者而言，该框架可以作为理解一个分布式账本技术解决方案可信度的一个尺度，比如该技术在哪些方面做了取舍，其经济价值是如何产生和获取的。重要的是，它将帮助投资者做出明智的投资决策，而不是被误导，比如，把一个实际上是“集权式“的技术或无法实现其设计目标的技术理解成“分布式”的技术。

对学者和研究人员而言，他们可能会发现该框架可提供一个清楚的理论基础，在这个基础之上，去开发与通信、经济、工业组织和许多其他学科相关的理论。

7.3 研究不足及未来方向展望

本报告中描述的框架及其应用是一个可用于分析各种分布式账本技术系统的模块化的、一般性的工具。该工具使用一个基于定性解释的三层分析的方法。尽管这种分析方式已尽可能客观。但客观地量化分布式账本技术系统的抽象方面（例如“分权化”）的难度，使得本报告的作者们必需作出一些内在的主观的探讨和研究，比如对技术堆栈的概念以及处在每一层的参与者的角色的处理和评估。

未来对分布式技术系统的研究可以关注本报告中描述的过程的更多技术的方面。例如，在进行分布式账本技术系统的案例研究时，该框架可以成为其它具体场景应用开发的第一步。类似地，对于一个特定的分布式账本技术流程，可以添加或开发新的结构配置，包括新过程或组成部分。此外，该框架也可以用于监管和立法研究，比如确定哪些流程或结构配置对应于什么法律框架（例如权限和依赖性）。

附录A：分布式账本系统详解

层级	区块	过　程	描　述
协议层	初始组件	系统间依赖性	调查系统在操作和/或数据级别的依赖性（自足，依赖，接口，外部）。
		代码库创建	选择足够的代码库（现有的，从头开始）作为系统的基础并设置访问条件（开源，闭源）。
		规则启动	定义并同意管理分布式账本技术系统的规则。
	变更组件	协议治理	阐明以有序和合法的方式更改协议的决策过程。
		协议变更	阐明如何执行商定好的要改变的规则。

层级	区块	过　程	描　述
网络层	通信组件	网络访问	决定授予系统访问权限的人员（打开，关闭）。
		数据广播	明确数据的复制方式（通用，多通道）。
		交易启动	确定谁可以创建交易以及如何将这些交易广播到系统（不受限制，受限制）。
	交易处理区块	记录提议	选择一组未确认的交易，将它们打包成为候选记录。通过执行协议规则指定的必要步骤（例如附加有效的工作量证明机制）将候选记录添加到账本。
		冲突解决规则	设置解决同等有效的建议记录之间冲突的规则，以便添加到账本（例如，最长链规则）。
		交易处理的奖励机制	明确交易处理背后的激励性质（内在/外在，货币/非货币）。
	验证组件	交易验证	在将未经确认的事务添加到日志之前，请确认其合法性和有效性。
		记录验证	在将记录添加到日志之前，验证记录是否符合协议规则。
		交易完结	确定确认记录的“暂时”结算和“永久”结算之间的过渡期（确定的，或多大概率）。

层级	区块	过　程	描　述
数据层	操作组件	输入	确定用于生成账本包含数据（内部，外部）的数据源。
		编程可执行交易	明确核心系统层的链上计算的表现程度 – 通常称为智能合约功能（无状态，有状态）。
		执行轨迹	确定正在执行计算的位置（链上，链外）。
	分账本组件	数据指向	确定记录中存储的数据指向的内容（内生的，外生的，混合的，自引用的）。

附录B：案例对比

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
协议层	初始	系统间依赖性	自给自足系统	自给自足系统	自给自足系统	自给自足系统	自给自足系统	自给自足系统
		代码库创建	起源创始块；开源；	起源创始块；开源；	起源创始块；开源；	基于Quorum的代码库（Quorum是以太坊的分叉）；开源；	超级分账本；封闭源用户模块；	超级分账本；封闭源；
		规则启动	规则由推荐客户端制定	正式规范协议（“黄纸”）	规则由推荐协议和推荐客户端制定	正式规范协议	正式规范协议	正式规范协议(默认是超级分账本的推荐)
	变更	协议治理	无政府：节点通过运行软件客户端的选择来投票；但比特币核心钱包参考实施对发展有重大影响。	分级：以太坊基金会和个别开发者对于总体发展具有重大的影响。	专政：瑞波实验室几乎拥有终极控制权，验证器可以通过“修正”特性进行投票。	民主：验证节点需要达成一致，没有中央管理者。	联邦：由投票人组成的指导委员会投票	专政：关键客户拥有最终权力
		协议变更	依靠核心GitHub存储的比特币改进建议（BIP);运行软件客户端的选择（通用的比特币核心钱包）	以太坊改进建议（EIP）；客户端软件可自选(Geth或Parity)	验证器通过投票赞成“修正”–如果80%同意，更改可以进行实施。	不清楚的：可能由可以选择是否新的客户端来决定。	技术更新由网络端和客户端决定；实质性规则变更需要通过正式的变更管理过程来执行。	Y 公司将更新为客户运行的客户端。
网络层	通信	网络访问	开放的，无限制的	开放的，无限制的	开放的，无限制的	半开放：网关通过验证器节点进行分布	封闭的：网关控制访问（公司Y按照正式流程）	封闭的：网关控制访问（公司Y按照正式流程）
		数据传播	通用数据传播（公共的）	通用数据传播（公共的）	通用数据传播（公共的）	通用数据传播（公共的）	多通道数据传播（选择性的隐私）；但是所有节点由公司Y代理运行。	多通道数据传播（选择性的隐私）；
		交易启动	不受限制：任何有相应私钥的人都可以创建和签署交易：需要通过审核器/监听器、SPV客户端或第三方服务（如API）将其广播到网络。	不受限制：任何有相应私钥的人都可以创建和签署交易：需要通过审核器/监听器、SPV客户端或第三方服务（如API）将其广播到网络。	不受限制：任何有相应私钥的人都可以创建和签署交易：需要通过审核器/监听器、SPV客户端或第三方服务（如API）将其广播到网络。	网络参与者可以创建交易；外部用户也可能通过审核器和监听器来传输标记的交易。	受限制的：通过到节点的API来选择一系列终端用户触发的交易。	受限制的：关键客户的ERP系统产生经过API提交给由公司Y的审核器和监听器的交易

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
网络层	交易流程	记录提议	无经许可的：矿工选择未经确认的交易，并创建候选区块。一个有效的候选区块需要将有效的SHA – 256哈希值附加到区块。头（通过选择一个随机数使哈希结果值足够低）	无经许可的：矿工选择未经确认的交易，并创建候选区块。一个有效的候选区块需要将有效的Ethash 工作量证明值附加到区块。头（通过选择一个随机数使哈希结果值足够低）	经许可的：验证器选择未确认的交易并创建新的分类帐。他们将候选人记录传递给“共识回合”；新分类账的多重计算	经许可的：验证器节点（30个左右不同的个体）从内存池中选择未经确认的交易，并将他们归入到一个候选区块。通过筏式共识机制来达到共识。	经许可的：验证器节点(所有15个供应商）创建与他们所涉及交易相关的纪录。简单的状态更改建议：通常没有异议	经许可的：公司Y控制的节点会选择未经确认的交易并且产生纪录(集中式节点)。
		冲突解决规则	最长有效链法则（例如积累最多工作量证明）	最长有效链法则（例如积累最多工作量证明）	在唯一节点列表（UNL）中使用多回合共识直到“绝大多数”(80%)达成共识为止	比赛机制：第一个区块获胜，其余竞争的区块被丢弃（这种情况很罕见，通常同一时间内只有一个矿工。	通常没有冲突：所有参与者都赞同所发生的情况。共识算法的精确度还是未知的。	忽略共识：没有冲突发生的可能
		交易处理的奖励性机制	固有性和货币性：：区块奖励（例如最近挖掘到的比特币和交易费用)	固有性和货币性：区块奖励（例如最近挖掘到的以太币和交易费用）	无报酬、隐性外在激励（网络稳健性和弹性）	无内在的货币激励（无本地代币）	(1) 外在货币激励：供应商得到指定服务支付的激励，以本国货币计价；(2) 外部非货币激励：(a) 给帮助供应商对抗GAFA的客户创造价值；(b) 帮助他们免于欺诈	无内在激励和货币激励—外部非货币激励可以在平台顺利运行
	验证	交易验证	在将交易广播到其它节点前审核器和监听器会验证每个未经确认的所有交易	在将交易广播到其它节点前审核器和监听器会验证每个未经确认的所有交易	在将交易广播到其它节点前跟踪节点验证未经确认的所有交易	在将交易广播到其它节点前审核器和监听器会验证每个未经确认的所有交易	每个审核器和监听器都会验证信道内的所有交易	公司Y控制的节点会验证特定信道内的每一个交易
数据层	运行	输入	主要的内部（例如以前的输出：UTXOs, 脚本）外部：具有时间戳目的的随机数据（例如通过OP_REtURN）。	内部（例如以前的输出：账户、智能合约)以及外部（oracles）	内部（例如以前的输出：账户）以及外部（与建立IOU有关的数据）	内部（例如以前的输出：账户、智能合约）以及外部（oracles、IPFS启用、链下隐私存储）	基础外部（开放ID连接(OIDC)：连接服务协议）。内部 = 以前的输出：哈希负载；赞同指令；接收证明	基础外部（依靠API的）：连接服务协议）。内部 = 以前的输出：哈希负载；赞同指令；接收证明

框架元素			比特币	ETHEREUM	Ripple	ALASTRIA	VERIFIED.ME	项目X
层	组件	过程						
数据层	运行	通过编程来执行的交易	无状态的：有限脚本语言实现多重签名和时间戳合同	在状态的：图灵完备的智能合同允许多用途计算	无状态的：链上具有专用目的的基础计算	在状态的：图灵完备的智能合同语言允许通用计算	无状态的：链上可以实现非常简单的业务逻辑	无状态的：链上可以实现非常简单的业务逻辑
		执行轨迹	用于运行链上简单脚本的固定功能机器	通用虚拟机：以太坊虚拟机（EVM）	用于基本链上运算的固定功能机器	通用虚拟机：以太坊虚拟机（EVM）允许在链上执行复杂的计算	用来管理用户赞同指令的业务逻辑可以在更高层(链下)执行	业务逻辑可以在外部平台（链下规则引擎）执行
	分账本	引用	内部的：系统特有的本地资产（BTC）。	（1）内部的（本地资产：ETH；用户定义的代币：dApps）；(2) 混合的（无标记的代币和/或外部事件的纪录。	（1）内部的（本地资产：XRP）；(2) 混合的（网关发布的IOU和可信任线路）	依赖于使用的案例：如果是本地资产或用户定义的代币则属于内在的。如果是外部与内部的结合则属于混合的。也有可能是完全外部的。	完全外部的：身份数据属于外部系统（如身份来源：政府，银行等）	完全外部的：ERP系统和保险纪录

附录A：分布式账本技术系统结构

管理员

管理员能控制谁可以访问核心代码存储库，可以决定添加、删除和修改代码以更改系统规则。管理员经常参与到系统治理的过程。

候选记录

尚未传播到网络的记录，因此未受到网络共识。

抗审查性

单方或集团无法单方面执行以下任何一项：1）改变系统规则；2）阻止或审查交易；3）扣押账户和/或冻结余额。

确认

为了从账本状态中删除一个交易而必需反转或重写的记录的数目。

共识算法

一个网络用于达成协议和验证记录的所有规则和流程。

开发商

编写和审查代码的角色，该代码是构成分布式账本技术系统及连接系统的技术构建区块的基础。开发人员可以是专业人员或参与志愿贡献者。

分布式账本技术系统

一种电子记录系统，其（i）使独立参与者网络能够围绕（ii）密码验证（“签名”）交易的权威排序建立共识。这些记录通过在多个节点上复制数据来实现（iii）持久化，以及（iv）通过加密哈希将它们链接起来以防篡改。（v）和解/共识过程的共同结果 – “账本” – 作为这些记录的权威版本。

内生数据指向

可以仅通过DLT系统本身创建和传输的数据，其存在仅在系统内有意义。它可以干什么由DLT系统强制自动执行。

外在数据指向

参考某些现实条件并需要从外部引入的数据。这通常需要网关与外部系统建立连接并在分布式账本技术系统之外执行决策。

分叉

把一个分布式账本技术系统分成两个或多个系统。在当两个或多个参与节点大致同时发布有效记录集时，分叉发生。分叉发生的一种可能时其作为攻击的一部分（例如51%攻击），另一种可能是当分布式账本技术系统协议被更改时（如果所有的使用者都要更新系统，则称为硬分叉，否则就是软分叉）。

网关

通过充当系统和外部世界之间的桥梁来为系统提供接口的角色。

混合数据指向

具有内生和外生指向特征的数据。到底用哪一个指向在某种程度上取决于网关。

独立验证

系统能够使每个参与者独立地验证其交易状态和系统的完整性。

分账本

指一个节点持有的记录集，不一定与其他节点的共识一致。分账本可以是部分、临时和异构的：它们可能包含也可能不包含所有相同的记录。

账本

在任何时间点由相当大比例的网络参与者共同持有的权威记录集，使得记录不可能被删除或修改（即“最终”）。

未达成共识的交易记录池

由一个节点持有的无序但有效的交易集合，这些交易尚未被纳入受网络共识规则约束的正式记录（即“未经证实的”交易）。也叫内存池。

多方共识

系统能够使独立方在不需要中心认证的情况下就共享记录集达成一致。

原生资产

协议中规定的主要数字资产（如果有的话）通常用于监管记录生产，在网络上支付交易费用，实施“货币政策”或调整激励措施。

网络

通过互连参与者和流程实现协议。

节点

网络参与者通过共享通道与对方通信。

链下

在正式系统边界之外发生的交互，操作和过程。

链上

系统内（即系统级）发生的交互，动作和过程，并反映在数据层中。

信息通道

用来在分布式账本技术系统和外部系统之间传输信息的网关。

参与者

与网络中的其他参与者互连，并通过在彼此之间传递消息进行通信。

永久性

数据在程序执行后保持可用的能力，以及在任意数量的节点的灾难性丢失中生存。

编程可执行交易

计算机脚本，当由特定消息触发时，由系统执行。当代码能够按照各方的意图进行操作时，执行的确定性降低了各个参与者彼此交互所需的信任级别。它们通常被称为智能合约，因为脚本能够用代码替换某些信托关系，例如

监管和托管。然而，它们不是自主的或适应性的（“聪明的”），也不是法律意义上的合同 – 相反，它们可以是实施合同或协议的技术手段。

协议

一组软件定义的规则，用于确定系统的运行方式。

记录

一系列交易数据，受到网络共识规则的约束，是全球账本的一部分。

记录重组

节点发现已形成新的账本版本，其排除该节点先前认为是账本的一部分的一个或多个记录。这些被排除的记录将会被遗弃。

共同记账

系统使多方能够共同创建、维护和更新共享记录集的能力。

智能合约

请参阅“编程可执行交易”。

篡改证据

参与者能够轻松检测到已确认记录的任意更改。

防篡改

一方很难改变已有记录（即交易历史）的能力。

交易

任何一个交易都是对总账本变更的一个建议。尽管有经济交易这样的含义，交易本质上不一定是经济（价值转移）。交易可以是未确认的（还没有包括在账本中）或已确认的（已写入账本）。

交易终结

确定何时可以将确认的记录视为“最终”（即不可逆）。终结性可以是概率性的（例如，基于工作量证明机制的系统，在计算上是不可回复的）或显式的（例如，包含必须出现在每个交易历史中的“检查点”的系统）。最终记录被视为永久结算，而已生产但可恢复的记录称为暂时结算。

交易处理

指定更新账本的一组流程：（i）哪些参与者有权更新共享的认证记录集（无权限与权限）和（ii）参与者如何就实施这些更新达成一致。也叫挖矿。

验证

确保参与者在账本状态下独立得出相同结论所需的一系列过程。这包括验证未确认交易的有效性，验证记录提议以及审核系统状态。

钱包

一种软件程序，能够存储和管理用于存储和传输数字资产的公钥和私钥。