软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(1):102−138 [doi: 10.13328/j.cnki.jos.007452] [CSTR: 32375.14.jos.007452]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

国产区块链软件发展趋势与核心技术分析*

何嘉昊[1], 杨 森[1], 潘子玲[2], 陈 厅[1], 沈 剑[2]

[1](电子科技大学 计算机科学与工程学院, 四川 成都 611731)

[2](浙江理工大学 信息科学与工程学院, 浙江 杭州 310018)

通信作者: 陈厅, E-mail: brokendragon@uestc.edu.cn

摘 要: 区块链作为一种分布式账本技术, 凭借加密和共识机制保证数据的安全、透明和不可篡改性, 为各行业提供了革命性的解决方案. 在国内, 基于区块链技术开发的软件得到了广泛关注和应用, 从金融领域的跨境支付、供应链金融以及政务领域等, 区块链软件都展示了巨大的潜力. 这些应用不仅能够提高业务流程的效率和透明度, 降低信任成本, 还为传统行业的数字化转型提供新的思路和方法. 以调研国产区块链软件的发展趋势与核心技术为目的, 从技术层面探讨关键技术突破、促进技术融合创新, 为技术标准制定提供依据, 从而提升国产区块链技术竞争力、开拓应用场景并规范行业发展. 为此, 围绕以下 3 个核心问题展开调研: (1) 国产区块链软件发展趋势如何? (2) 国产区块链软件具有哪些核心技术? (3) 国内外区块链软件在核心技术上有哪些不同? 为了回答这些问题, 首先通过 3 种途径收集了 1 268 个区块链软件, 并结合所属公司、CTO 等信息对区块链软件进行筛选, 最终得到 103 个国产区块链软件. 随后, 基于公司信息对区块链软件的基本信息进行统计, 并从软件发展历程、软件分布以及软件关系这 3 个角度分析当前国产区块链软件的发展趋势. 考虑技术文档或开发文档等关键信息的重要性, 进一步筛选出 39 个包含技术信息的高质量区块链软件. 接下来, 从区块链技术的 6 个层次对这 39 个区块链软件的核心技术情况进行统计与分析, 并基于这些分析对比中外区块链软件在核心技术方面的差异. 最终, 得到 28 个现象和 13 个见解. 这些发现有助于区块链研究人员、开发者以及从业者了解国产区块链发展的现状, 并为未来使用、改进国产区块链软件提供见解.

关键词: 国产区块链软件; 发展趋势分析; 核心技术统计与分析; 中外核心技术对比; 应用拓展

中图法分类号: TP311

中文引用格式: 何嘉昊, 杨森, 潘子玲, 陈厅, 沈剑. 国产区块链软件发展趋势与核心技术分析. 软件学报, 2026, 37(1): 102–138. http://www.jos.org.cn/1000-9825/7452.htm

英文引用格式: He JH, Yang S, Pan ZL, Chen T, Shen J. Analysis of Development Trend and Core Technologies of Chinese Blockchain Software. Ruan Jian Xue Bao/Journal of Software, 2026, 37(1): 102–138 (in Chinese). http://www.jos.org.cn/1000-9825/7452.htm

Analysis of Development Trend and Core Technologies of Chinese Blockchain Software

HE Jia-Hao[1], YANG Sen[1], PAN Zi-Ling[2], CHEN Ting[1], SHEN Jian[2]

[1](School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

[2](School of Information Science and Engineering, Zhejiang Sci-tech University, Hangzhou 310018, China)

Abstract: Blockchain, as a distributed ledger technology, ensures data security, transparency, and immutability through encryption and consensus mechanisms, offering transformative solutions across various industries. In China, blockchain-based software has attracted widespread attention and application, demonstrating considerable potential in fields such as cross-border payments, supply chain finance,

* 基金项目: 国家重点研发计划 (2023YFB2703700); 国家自然科学基金重点项目 (62332004, U21A20465); 四川省自然科学基金杰出青年科学基金 (2023NSFSC1963); 浙江省领军型创新团队 (2023R01001)
收稿时间: 2024-10-22; 修改时间: 2025-03-18; 采用时间: 2025-04-23; jos 在线出版时间: 2025-08-27
CNKI 网络首发时间: 2025-09-05

and government services. These applications not only enhance the efficiency and transparency of business processes but also reduce trust costs and offer new approaches for the digital transformation of traditional industries. This study investigates the development trends and core technologies of Chinese blockchain software, focusing on key technological breakthroughs, promoting integration and innovation, and providing a foundation for the formulation of technical standards. The aim is to enhance the competitiveness of Chinese blockchain technologies, broaden application scenarios, and support the standardized development of the industry. Three core research questions are addressed: (1) What are the development trends of Chinese blockchain software? (2) What are the core technologies involved? (3) What are the differences in core technologies between Chinese and foreign blockchain software? To address these questions, 1268 blockchain software entries have been collected through three channels. Based on information regarding affiliated companies and chief technology officers (CTOs), 103 Chinese blockchain software entries are identified. A statistical analysis of basic software attributes is conducted, examining development trends from three perspectives: software development history, distribution, and interrelationships. Given the importance of technical and development documentation, 39 high-quality blockchain software entries containing detailed technical information are further selected. Subsequently, a statistical and analytical evaluation of the core technologies of these 39 software systems is conducted across six technical layers of blockchain architecture. Based on this analysis, differences in core technologies between Chinese and foreign blockchain software are compared. In total, 28 phenomena and 13 insights are identified. These findings provide researchers, developers, and practitioners with a comprehensive understanding of the current state of Chinese blockchain development and offer valuable references for future adoption and improvement of Chinese blockchain software.

Key words: Chinese blockchain software; development trend analysis; core technical statistics and analysis; comparison of core technologies between China and other countries; application expansion

区块链技术自 2008 年比特币白皮书[1]提出以来, 逐渐发展成为一项颠覆性创新, 深刻改变了金融[2]、供应链管理[3]、物联网[4,5]、城市空中交通[6]多个领域. 它通过去中心化、不可篡改的分布式账本, 为信任和数据共享问题提供了全新的解决方案[7]. 随后, 以太坊平台提出了智能合约的概念[8], 推动了区块链技术的进一步发展, 涌现出许多基于区块链技术的新应用, 例如去中心化应用 (decentralized application, DApp)[9]、非同质化代币 (non-fungible token, NFT)[10]以及去中心化金融 (decentralized finance, DeFi)[11].

在这个过程中, 我国也通过政策推动和技术创新并行的方式逐步探索和深化应用区块链技术. 早期, 我国市场对比特币等数字货币交易展现出极大兴趣, 催生了相关交易软件. 2017 年, 政府加强加密货币监管, 并推动区块链技术在非金融领域的应用, 各行业区块链软件逐步开发. 到 2019 年, 区块链技术被提升至国家战略高度, 伴随政策支持和激励措施, 推动了技术研发与产业应用. 为了推动数字经济转型和金融科技创新, 推出了区块链服务网络 (blockchain service network, BSN)[12]和数字人民币 (electronic Chinese Yuan, e-CNY)[13]等项目. 以上众多应用都是由区块链软件组成的, 这些软件为区块链网络的运行提供服务, 包括区块链底层软件[1,8]、区块链即服务 (blockchain as a service, BaaS)[14]以及区块链浏览器和分析工具[15–18]等多个领域的软件. 本文在研究区块链软件时仅关注区块链底层软件, 具体而言, 本文关注的是支持区块链网络核心功能的组件, 包括数据结构、区块传播机制、共识协议、合约执行引擎等. 这些底层软件决定了区块链的性能、安全性和可扩展性.

然而, 目前缺乏针对国产区块链软件发展趋势与核心技术的系统性研究. 这可能导致研究人员和从业者无法全面了解国产区块链软件的发展现状, 不利于我国区块链技术及相关产业的持续推进. 尽管一些机构发布了区块链行业分析报告, 但这些报告通常存在局限性, 往往仅从金融等单一行业的视角分析区块链软件的类型和分布等基础信息, 缺乏全局性与深度. 系统性研究国产区块链软件具有重要的学术与实践意义. 首先, 深入分析国产区块链软件的发展趋势, 有助于全面了解国产区块链软件的类型、地域分布以及生态系统中的相互关系, 为行业从业者提供全景视角, 推动区块链技术的应用与扩展. 其次, 深入探讨国产区块链的核心技术, 如共识算法、合约执行引擎和数据传输协议等, 能够帮助研究人员和行业实践者了解国内技术发展的关键优势与挑战, 推动技术创新. 最后, 通过中外区块链软件的对比分析, 可以揭示在核心技术上的差异, 帮助评估国产区块链技术的国际竞争力, 并为技术升级提供借鉴. 然而, 系统性研究国产区块链软件面临 3 大挑战: 1) 如何有效收集和筛选国产区块链软件的数据, 确保数据的全面性和准确性; 2) 如何科学、全面地展示国产区块链软件的发展趋势; 3) 如何深入分析国产区块链软件的核心技术, 如共识机制、智能合约执行、数据结构优化等.

本文针对国产区块链软件发展趋势与核心技术进行系统性的研究. 为了尽可能地收集国产区块链软件, 首先

通过 3 种途径进行区块链软件收集以确保数据收集的全面性, 最终收集到 1 268 个区块链软件. 随后对收集到的软件进行筛选, 去除重复以及非底层区块链软件, 最终筛选得到 103 个国产区块链软件. 为了科学、全面地展示国产区块链软件的发展趋势, 我们统计了区块链软件官网、所属公司等基本信息, 从区块链软件发展历程、软件分布以及软件关系这 3 个方面分析了国产区块链软件的发展趋势. 为了分析国产区块链软件的核心技术, 从数据层、网络层、共识层、合约层、应用层以及其他技术这 6 个层次对国产区块链软件使用的核心技术进行统计与分析. 由于并非所有的区块链软件都具备详细的技术信息, 因此对 103 个国产区块链软件进行了进一步评估, 从中筛选得到 39 个高质量的国产区块链软件进行技术分析. 最后, 选择了 6 个具有代表性的国外区块链软件, 通过分析这些软件的核心技术来对比国内外区块链软件.

本文的主要贡献包括以下内容.

(1) 对国产区块链软件的发展趋势和核心技术进行系统性地研究. 我们设计了国产区块链软件的筛选标准以及高质量区块链软件的过滤标准. 筛选得到了 103 个国产区块链软件以分析国产区块链软件的发展趋势, 并从中筛选出 39 个高质量区块链软件进行核心技术分析.

(2) 从 3 个方面对国产区块链软件的发展趋势进行分析, 从 6 个层次对国产区块链软件的核心技术进行了分析, 并且将其与国外 6 个具有代表性的区块链软件进行了对比.

(3) 提供了统计结果并分析得到了 28 个有趣的现象以及 13 个见解. 这有助于研究人员和相关从业者全面了解国产区块链软件发展以及核心技术. 统计结果公布在 GitHub (https://github.com/L-INGLING/Chain-Platform) 中.

本文第 1 节为方法论, 介绍国产区块链软件的收集、筛选与数据统计. 第 2 节为区块链软件发展趋势分析. 第 3 节为区块链软件核心技术分析. 第 4 节为中外区块链软件对比. 第 5 节为总结.

1 方法论

本节将对国产区块链软件发展趋势与核心技术分析的详细步骤进行介绍, 包括软件收集、筛选以及数据分析. 具体过程如图 1 所示, 通过 3 种途径进行区块链软件收集. 随后, 我们设计了 3 项国产区块链软件筛选标准、2 项高质量区块链软件过滤标准, 基于这些标准人工分析以筛选国产区块链软件. 最后, 通过浏览官网、阅读技术/开发文档进行数据统计与分析.

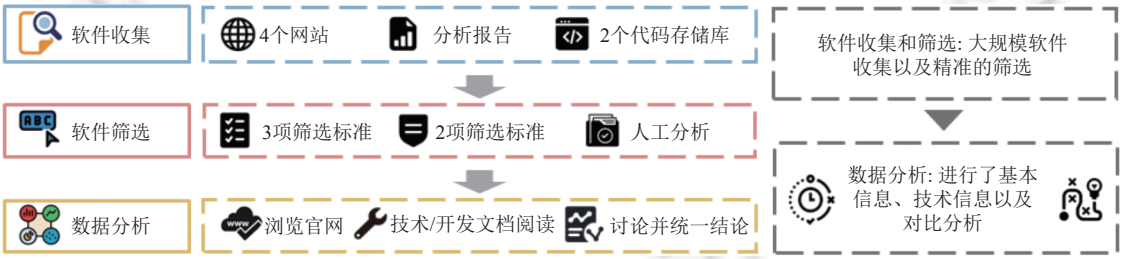
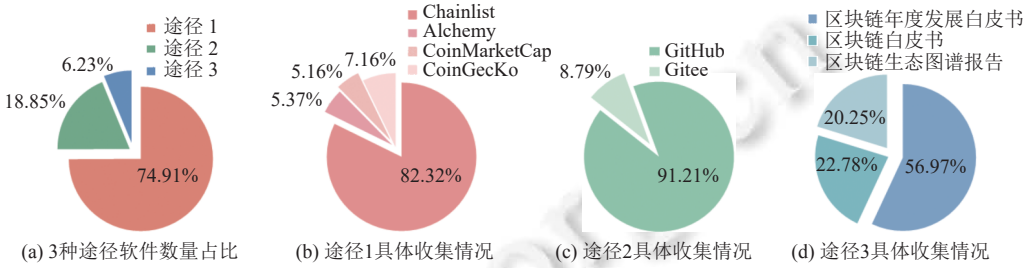


图 1 方法论概述

1.1 软件收集

我们通过 3 种途径进行了区块链软件收集, 包括: (1) 区块链软件记录网站; (2) 区块链代码仓库; (3) 区块链行业分析报告. 其中区块链软件记录网站指那些专门用于记录区块链软件的网站, 包括 Chainlist[19]、Alchemy[20]、CoinMarketCap[21]以及 CoinGecKo[22]. 其中 Chainlist 是 EVM (Ethereum virtual machine) 网络的列表, 网站中记录了大量 EVM[23]系列的区块链软件. Alchemy 中记录了年度最受欢迎的区块链软件. CoinMarketCap 主要关注加密货币市场, 此外它还提供了很多区块链软件的信息. CoinGecKo 类似于 CoinMarketCap, 同样提供了广泛的加密货币和区块链软件的信息. 考虑到一些初创区块链项目可能会选择在代码仓库网站上发布区块链软件, 因此在搜索中我们考虑了 GitHub[24]以及 Gitee[25]两个大型代码仓库. 其中, GitHub 是全球范围内最大的代码仓库, 使用量巨大. Gitee 则是我国的代码仓库, 发展迅猛. 在上述两个代码仓库中使用了以下 6 个关键词进行了搜索: “blockchain”

“blockchain platform”“blockchain software”“decentralized application platform”“decentralized Apps”以及“distributed ledger platform”. 由于区块链软件记录网站中记录的大多数为国外区块链软件, 因此对国内的区块链行业分析报告进行收集与分析, 它是一些官方或者区块链研究机构通过对一段时间内区块链软件的发展情况调研, 得到的相关区块链产业发展情况的总结. 我们主要参考了中国信息通信研究院、赛迪区块链研究院以及链上产业区块链研究院这 3 家单位的区块链行业分析报告. 这 3 家单位的权威性、专业性、研究广度和深度、数据可靠性以及对行业的影响力, 使其成为获取区块链软件相关信息的重要参考来源. 基于以上 3 种途径, 我们对 2008 年 1 月–2024 年 6 月所有上线的区块链软件进行了收集, 3 种途径所收集到的区块链软件数量如图 2 所示.



(a) 3种途径软件数量占比 (b) 途径1具体收集情况 (c) 途径2具体收集情况 (d) 途径3具体收集情况

图 2 不同途径收集到区块链软件数量占比情况

图 2 展示了 3 种途径收集到的区块链软件数量情况以及每种途径的具体收集情况. 我们共收集得到了 1 268 个区块链软件. 其中途径 1、2、3 收集的区块链软件数量分别为 950、239 以及 79. 途径 1 中, 从 Chainlist 网站上共收集到了 782 个区块链软件, 远高于同途径的其他网站 (Alchemy: 51, CoinMarketCap: 49, CoinGecKo: 68). 途径 2 中, 我们从区块链代码仓库 GitHub 中, 通过关键词搜索的方式共获得了 218 个区块链软件, 占比超过了 90%. 途径 3 中, 从区块链年度发展白皮书中收集到的区块链软件数量为 45, 比其他两种方式收集的区块链软件数量总和还多.

1.2 软件筛选

软件筛选是为了对获取到的区块链软件数据进行清洗, 保证能够对具有研究价值的国产区块链软件进行发展趋势与核心技术分析. 首先, 我们设计了 3 条筛选标准用于筛选国产区块链软件.

• 区块链软件的开发团队位置是否属于中国: 考察开发团队的主要活动地点和办公地点是否在中国, 以了解项目的技术开发和日常运营是否主要在中国进行. 随着远程工作的普及, 许多区块链项目的开发团队是全球分布的, 没有固定的“位置”. 因此需要其他信息的补充.

• 区块链软件注册地或区块链软件的知识产权是否属于中国: 检查项目的法人注册地、公司总部以及相关的知识产权注册地是否在中国, 这有助于了解项目的法律归属和保护权利的地理位置.

• 区块链软件的创始人或 CTO 是否为中国人: 确认项目的创始人或技术负责人 (如 CTO) 的国籍或出生地是否为中国, 这可以反映项目的创始背景和技术导向.

以上 3 条标准属于平行关系, 我们认为中国本土区块链软件应该至少满足其中的一条或多条标准. 根据这些标准可以筛选得到属于中国的区块链软件. 随后, 通过对初步筛选后的区块链软件实施进一步过滤, 以确保所纳入分析的样本具备唯一性且聚焦于底层系统, 具体过滤标准如下.

• 没有重复的区块链软件.

• 是底层区块链软件而非区块链应用软件.

对于第 1 点, 由于不同途径的搜索结果中可能包含的名称不同, 可能会导致区块链软件收集的重复. 例如, 唯链的英文名为 Vechain[26], 在数据合并时可能会重复记录. 对于第 2 点, 由于本文研究仅关注区块链底层软件的发展趋势以及核心技术, 基于区块链软件的应用等评估超出了本文的研究范围. 例如, 联通区块链服务这类 BaaS 应用[27]以及火币[28]这类基于区块链软件的金融服务等. 这里我们需要去除的内容包括: (1) BaaS 应用; (2) 金融服务.

由于在分析区块链软件核心技术时需要获取软件使用的技术信息, 而部分国产区块链软件仅具有官网, 或开发/技术文档信息严重缺失. 针对这种情况, 我们对 103 个国产区块链软件进行了进一步评估, 剔除没有技术或开发文档的区块链软件或近一年内没有更新开发/技术文档信息的区块链软件, 最终得到了 39 个包含详细技术信息的区块链软件.

在区块链软件的数据收集与筛选工作中, 我们招募了 4 名成员 (1 名博士, 3 名硕士, 均至少有 1 年以上的区块链开发经验) 进行人工分析, 每位成员的分析过程保持相对独立, 并且分析结果都至少有两位以上的成员进行重复验证. 在区块链软件数据收集过程中, 我们共统计得到了 1 268 个区块链软件, 其中包含 201 个区块链软件符合筛选标准, 属于国产区块链软件. 经过过滤后, 最终得到了 103 个不重复的底层国产区块链软件, 我们将基于这 103 个区块链软件进行统计以分析国产区块链软件的发展趋势. 需要注意的是, 在这 103 个软件中仅有 39 个包含详细的技术信息, 可以用于分析国产区块链软件的核心技术. 筛选过程中, 不同国产区块链软件符合标准的情况如图 3 所示.

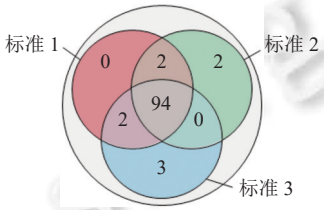


图 3 国产区块链软件所符合的标准

如图 3 所示, 大部分的区块链软件能够同时满足筛选标准 1、2、3, 只有少部分软件平台仅符合其中的一项或两项. 例如, BSC (binance smart chain)[29]、波场链[30]和 CKB[31]仅符合标准 3, 即它们的创始人是中国人. 此外, IRISnet[32]和 PlatON[33]符合标准 1、3, 这是因为这些平台主要是由中国人员或实验室开发的, 但是所属公司的注册地却在海外.

1.3 数据分析

我们基于调研目标, 设计了覆盖国产区块链软件发展趋势与核心技术特征的问题体系, 并在表 1 中对各问题及其对应的数据类型进行了系统整理. 数据采集过程中, 主要通过浏览区块链软件官方网站、查阅技术文档与开发文档等途径, 提取回答各问题所需的关键信息. 为确保数据的准确性与一致性, 所有收集数据均由项目组成员协同讨论并确认. 对于在关键问题 (即前两个核心问题) 上无法获取有效信息的软件样本, 予以剔除处理, 以提升样本数据的有效性与分析结果的可靠性.

表 1 调研问题与收集数据的类型

问题	收集数据的类型
问题1: 国产区块链软件发展趋势如何?	1) 国产区块链发展历程; 2) 国产区块链软件分布; 3) 国产区块链软件关系
问题2: 国产区块链软件具有哪些核心技术?	1) 数据层: 数据结构、数据库、加密算法; 2) 网络层: 数据传输协议、传播机制; 3) 共识层; 4) 合约层: 合约编程语言、合约执行引擎; 5) 应用层; 6) 其他技术: 隐私保护技术、可扩展性技术、跨链技术
问题3: 国内外区块链软件在核心技术上有哪些不同?	1) 数据层; 2) 网络层; 3) 共识层; 4) 合约层; 5) 应用层; 6) 其他技术

下面将介绍回答每个问题的具体步骤. 在回答问题 1 时, 我们主要从国产区块链软件发展历程、分布以及软件之间的关系进行了统计与分析, 这些数据可以从区块链官网、相关新闻报道以及行业报告中获取. 在统计区块链软件发展历程时, 主要是统计区块链软件的发布时间以及截止时间. 这里的发布时间是指软件上线时间, 截止时间则是指区块链软件最后一次新闻发布或技术文档更新时间. 当一个区块链软件的官网咨询或代码仓库超过 1 年的时间没有更新时, 我们认为该区块链软件已经停止更新, 例如, inchain[34], 该区块链软件官方 GitHub 的最新更新时间已经是 6 年前. 区块链软件分布是指软件所属公司的注册地分布, 我们通过统计该分布以分析国产区块链行

业发展在地域方面的差异. 至于国产区块链软件关系则是根据官网公布的信息, 统计不同区块链软件之间的关系, 例如至信链[35]是基于长安链[36]开发的. 在实际统计时, 我们对 103 个国产区块链软件的发展趋势进行了统计与分析, 具体内容见第 2 节.

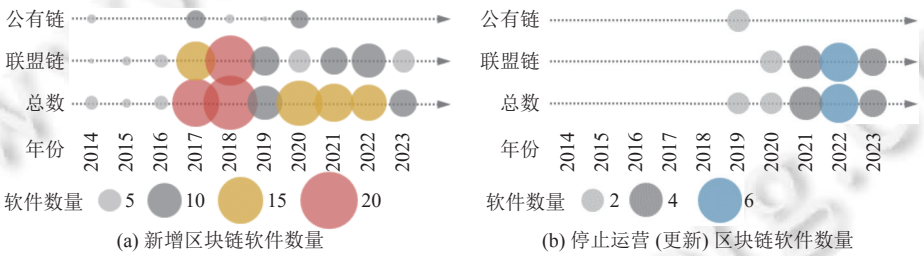
在回答问题 2 时, 我们主要是从区块链的 6 个层次对区块链软件所使用的技术进行分析. 袁勇等人[37]在分析区块链技术时, 提出从数据层、网络层、共识层、激励层、合约层以及应用层这 6 个层次进行分析. 由于国产区块链软件在技术方面的发展重点在于提升区块链的性能, 包括扩展性、安全性和互操作性等方面, 并且这些技术方面的改进有助于推动区块链的广泛应用, 而激励层的发展并非主要关注点. 因此, 我们在进行技术分析时并没有对区块链软件激励层的技术进行统计与分析. 此外, 袁勇等人的研究已是 8 年前, 相比之下近些年来区块链的隐私保护、可扩展性等技术发展迅速, 国产区块链软件在这类技术上的发展现状研究具有较大意义. 因此我们对国产区块链软件的这类技术进行了统计与分析, 称为“其他技术”.

在回答问题 3 时, 我们选择了国外人气较高的 2 类、共计 6 个区块链软件, 分别是 Hyperledger Fabric[38,39]、CordaR3[40,41]、Quorum[42,43]、Ethereum[44]、Polygon[45]以及 Solana[46,47], 并基于问题 2 的 6 个层次对国外区块链软件的核心技术进行分析, 在每个层次中我们仅讨论当前国产区块链软件整体技术发展方向以及水平与国外具有代表性的区块链软件的不同. 在此基础上总结并回答当前国内外区块链软件发展差异以及国产区块链软件未来的发展方向.

2 国产区块链软件发展趋势如何?

2.1 区块链软件发展历程

区块链软件发展历程是指区块链软件从发布到停止运营 (更新) 的整个过程, 其中停止更新是指 1 年内没有进行官网、技术文档或 GitHub 代码更新. 我们对每年新发布的区块链软件以及停止运营 (更新) 的区块链软件数量进行了统计, 具体信息如图 4 所示.



(a) 新增区块链软件数量 (b) 停止运营 (更新) 区块链软件数量

图 4 国产区块链软件发展情况

图 4 描述了国产区块链软件的发展情况, 包含两个部分, 其中图 4(a) 和 (b) 分别从总数、公有链数量以及联盟链数量这 3 个角度对每年新增/停止运营 (更新) 的区块链软件数量进行了统计. 通过分析这些数据, 可以得到以下 3 个现象.

现象 1: 国产区块链软件最早于 2014 年发布, 在 2017 年和 2018 年, 区块链软件的新发布数量达到了顶点, 均超过了 20. 产生这一现象的原因主要包括以下几个方面: 1) 技术方面: 2015 年, 以太坊的发布标志着图灵完备的智能合约的出现, 这为开发者提供了更加便捷的区块链应用开发工具. 智能合约的广泛应用促进了区块链技术的实际落地. 因此, 区块链相关公司数量在 2016 年显著增加, 并在 2017 年和 2018 年迎来了爆发式增长期, 标志着行业发展的重要转折点. 2) 政策方面: 在此期间, 我国政府对区块链技术的发展持鼓励态度, 并出台了多项政策予以支持. 例如, 2017 年工信部发布的《软件和信息技术服务业发展规划 (2016–2020 年)》[48]中, 明确提出区块链等创新技术要达到国际先进水平, 这直接推动了区块链技术的研发与应用. 此外, 2018 年工信部发布的《2018 年信息化和软件服务业标准化工作要点》[49], 推动了区块链技术在标准化建设方面的努力, 为行业发展提供了规范与保障. 政策的支持为区块链企业提供了资本、技术支持和市场准入的便利, 进一步促使区块链软件的快速推出与应

用. 3) 财富效应: 虚拟货币价格的飞涨是区块链公司大规模增加的重要推动力. 例如, 2014 年, 以太坊成功筹集了 1 800 万美元, 2017 年以太坊的市值已接近 190 亿美元. 许多首次代币发行项目发行的 Token 价格也大幅上涨. 这种新型的国际融资方式吸引了大量机构、资本和创业者进入区块链领域, 进一步推动了区块链行业的繁荣.

现象 2: 自 2020 年起, 停止运营 (更新) 的区块链软件数量逐渐增加. 这主要与以下几个因素相关: 监管加强、技术瓶颈、市场泡沫破裂以及缺乏成熟的商业应用模式. 监管加强方面, 自 2020 年开始, 政策监管的加强对区块链软件的运营产生了直接影响. 我国政府加大了对虚拟货币交易和非法融资活动的监管, 出台了一系列政策来遏制虚拟货币的炒作和非法融资行为. 政府的政策导向使得一些区块链项目面临运营困境, 部分公司由于担心触犯政策法规, 不得不调整或终止原本计划中的区块链项目. 这种在虚拟货币监管方面的政策不确定性导致了很多区块链项目不得不放缓进展或完全停止运营. 技术瓶颈是另一个主要原因, 当时的区块链软件很难处理大量的并发交易, 容易产生交易积压的问题, 这阻碍了其在商业方面的应用. 市场泡沫的破裂主要与加密货币市场的过度炒作有关, 导致投资者损失惨重, 并且对区块链的热情大幅下降. 最后, 缺乏成熟的商业模式表现在多个方面, 尤其是供应链领域, 虽然一些区块链项目旨在改善供应链的透明度和可追溯性, 但在实际推广中遇到了较大困难. 例如, 一些供应商可能不愿意采用区块链技术, 因为这需要投入成本进行系统改造, 而短期内难以看到直接的经济回报. 由于缺乏可行的商业模式, 投资者和企业在投入资源时变得更加谨慎, 进一步影响了区块链软件的增长与停运.

现象 3: 自 2015 年起, 每年新增/停止运营的区块链软件类型大多数为联盟链. 这可能与市场需求、技术成熟度等因素密切相关. 首先, 联盟链更适合用于企业之间的合作, 尤其是在供应链管理、金融服务等领域, 这些场景需要可信且可控的环境, 推动了联盟链的快速发展. 因此新增的国产区块链软件以联盟链为主. 然而, 尽管联盟链具有许多优势, 但市场竞争激烈, 技术瓶颈和缺乏成熟商业模式仍然导致一些项目无法持续运营, 形成停运现象. 这也是联盟链类型的区块链软件停止运营数量多的原因.

见解 1: 技术的成熟和安全性是行业发展的关键, 未来区块链的发展需要持续关注技术创新, 尤其是在可扩展性和安全性方面的突破, 以促进其在各个行业的广泛应用.

2.2 区块链软件分布

通过调查国产区块链软件所属公司的注册地, 我们对国产区块链软件分布情况进行了统计, 从地域的角度对当前国产区块链行业的发展情况进行分析. 统计结果如图 5 所示. 除了区块链软件的地理分布外, 图 5 中还包括了具体的软件分类, 我们对不同链类型的具体区块链软件名称进行了统计, 如表 2 所示.

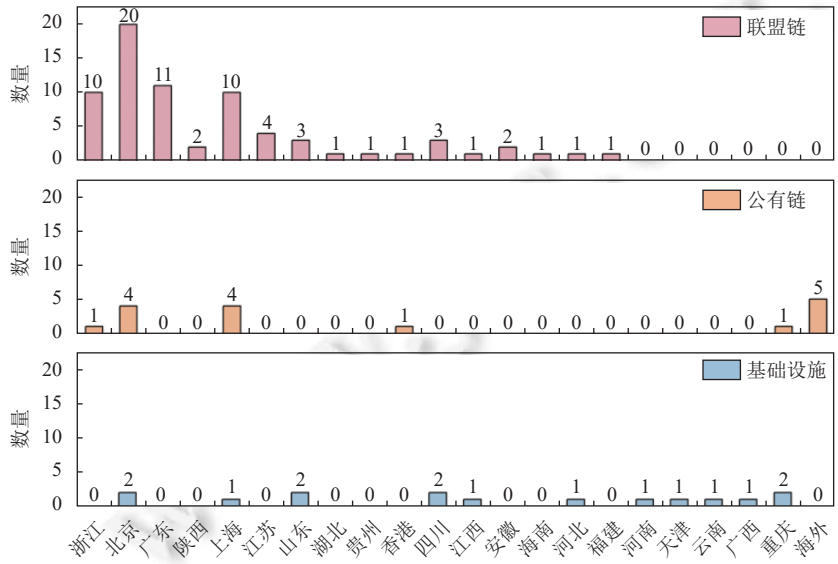


图 5 国产区块链软件分布图

表 2 区块链软件类型

链类型	软件名称	数量
公有链	星火链, 树图链, BSC, NEO, Aelf, 星云链, Ontolog, 波场链, CKB, IRISnet, PlatON, Mixin, 火币生态链, 印链, 维优元界, 超脑链	16
联盟链	蚂蚁链, 长安链, 百度链, 趣链, 天河链, 京东链, FISCO BCOS, 至信链, 华为链, Z-Ledger, RepChain, 迅雷链, 海峡链, 云链, 万纳链, Chain33, BubiChain, IRITA, 梧桐链, 磁云数字区块链M0, 中移链, 延安链, 文昌链, 泰安链, 武汉链, 安顺链, 唯链, 之江链, 网易链, BSN, 壹账链, 电科信链, 网录区块链, 众安链, 骏途链, 元镜联盟链, Hspeed链, 新版链, 亿条链, 瞬元智能链, 星蓝云链, 浪潮云链, 质安链, 银链, 合链, 瑞策联盟链, 象链, TrustSQL, 原本链, Towerchain, 博雅正链, 蜂巢链, TDChain, New Spiral, 达朴链网, 晶格链, 联通链, 常享链, 保交链, HuoChain, 太一链, RivSpace, 银信超级链, StarCloudChain, 蓉易链, 中信科技区块链, 文创链, 农信链, 信医链, 旅信链, 中数链, 文交链	72
区块链基础设施	BSI, CA联盟可信“存证”链, 蜀信链, 桂链, 渝快链, 昆明市区区块链公共服务平台, 江西省BaaS开放平台, 泉城链, 中招公信链, 海河智链, 河南链, 目录链, 浦江数链, 山城链, 雄安链	15

如图 5 所示, 我们按照区块链类别对各个省级行政区注册的区块链软件数量进行了统计. 结合图 5 和表 2 分析可以得到国产区块链软件分布上存在的 4 个现象.

现象 4: 除注册在海外的 5 个区块链软件外, 我国有 21 个省级行政区拥有区块链软件. 根据区块链软件数量排名, 前 3 名分别是北京、上海、浙江和广东 (并列第 3). 这一分布可能与为这些地区在经济、科技、金融等方面具有强大的实力和优势. 北京拥有丰富的高校和科研机构资源, 能够培养和吸引大量高素质的区块链技术人才. 同时, 北京的政策支持力度大, 为区块链技术的发展提供了良好的政策环境. 上海具有完善的金融基础设施和活跃的金融市场, 为区块链技术在金融领域的应用提供了广阔的空间. 浙江和广东则以其发达的民营经济和创新氛围著称, 企业对新技术的接受度高, 投资环境良好, 这些因素共同促进了区块链技术的创新和发展.

现象 5: 大多数区块链软件的注册地位于各个省的省会城市, 仅少部分区块链软件注册在非省会城市. 具体地, 安顺链 (贵州省安顺市)[50]、泰安链 (山东省泰安市)[51]以及众安链 (江西省宜春市)[52]注册在非省会城市, 广东的 11 个区块链软件中只有一个注册在广州, 其余均在深圳. 这一现象可能归因于省会城市通常是一个省的政治、经济、文化中心, 拥有更好的基础设施、人才资源、政策支持和投资环境. 相比之下, 非省会城市在这些方面可能相对较弱. 然而, 安顺、泰安和宜春这 3 个城市可能在某些特定领域或具有独特的优势, 吸引了区块链企业的注册. 例如, 安顺可能在大数据产业方面有一定的发展基础, 泰安可能在旅游产业数字化方面有需求, 宜春可能在新能源或其他特定领域有创新应用场景. 至于广东省的区块链软件分布, 深圳凭借其在科技创新和金融创新方面的优势, 称为区块链企业的聚集地. 深圳拥有众多的高新技术企业和创新型企业, 形成了良好的创新生态系统. 同时, 深圳的金融市场活跃, 风险投资机构众多, 为区块链技术的发展提供了充足的资金支持. 相对而言, 广州虽然是重要的经济中心和交通枢纽, 但在科技创新和金融创新方面可能相对逊色, 因此广东省大多数区块链软件注册在深圳.

现象 6: 在符合标准的区块链软件中, 超过 70% 的软件属于联盟链, 公有链和区块链基础设施的数量相当. 在区块链软件分类时, 一些区块链软件被归类为区块链基础设施, 这类软件为区块链网络运行和发展提供底层技术架构和服务体系, 为应用开发提供必要的技术支持和保障. 基于这些基础设施开发的应用, 几乎都是联盟链类型. 因此, 符合标准的国产区块链软件中有超过 85% 的软件属于联盟链, 这一现象可能与国产区块链应用场景更多地集中在企业之间的合作、供应链管理等领域密切相关, 这些场景更适合采用联盟链的模式. 此外, 联盟链相较公有链具有更好的性能和扩展性、能够保护数据隐私、并且更加符合监管需求. 当然, 随着技术的不断进步, 公有链也将在一些特定领域得到进一步的发展.

现象 7: 仅有 5 个省级行政区拥有公有区块链软件, 分别是北京、上海、香港、浙江和重庆. 其中重庆市的印链科技有限公司已于 2019 年清算倒闭. 这是因为公有区块链软件的开发和运营需要更高的技术水平、更严格的安全保障和更广泛的用户参与. 北京、上海、香港、浙江等地在经济实力、科技水平、人才资源等方面具有优势, 能够满足公有区块链软件的发展需求. 重庆尽管在推动区块链技术的发展方面也有积极举措, 但在某些方面可能仍然存在不足. 印链科技的倒闭可能是由于多种原因造成的, 如市场竞争激烈、技术创新不足、资金链断裂等. 这也反映了区块链行业的高风险性和不确定性.

见解 2: 区域资源配置、市场需求、政策支持和技术创新在区块链发展的过程中具有重要影响, 区块链行业从业者在选择发展方向时, 必须综合考虑这些因素, 以确保项目的可持续发展.

2.3 区块链软件关系

区块链软件关系是指一些区块链软件可能会在其他区块链软件的基础上, 根据特定的应用场景进行开发, 这样可以继承原软件的成熟技术架构和功能模块, 并在此基础上进行改进和创新. 这种开发方式有助于促进技术融合, 并能够更好地满足特定场景的需求. 我们通过官方信息、区块链行业报告等途径统计不同区块链软件之间的关系, 以分析国产区块链软件发展趋势.

如图 6 所示, 与主干相连接的区块链软件是一些基础区块链软件, 与其相连接的平台是基于基础区块链软件开发的区块链软件. 通过分析国产区块链软件关系, 得出以下两个现象.

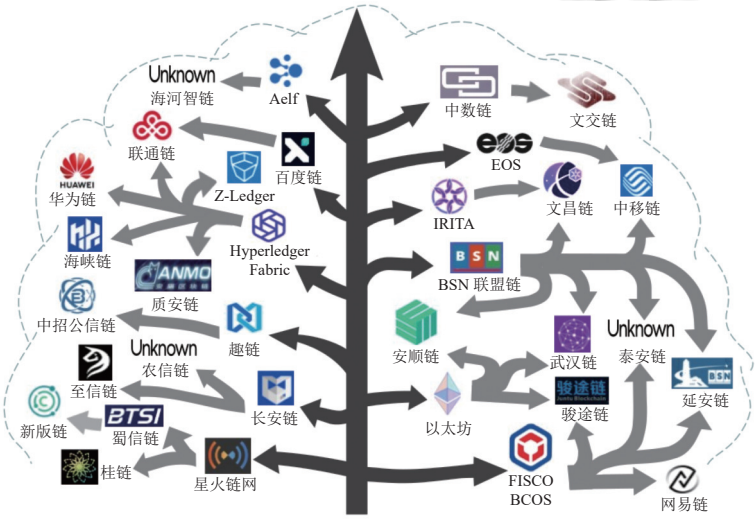


图 6 国产区块链软件关系

现象 8: 国产区块链软件大多基于成熟的区块链技术平台开发的, 其中基于 Hyperledger Fabric[38]、以太坊[44]、FISCO BCOS[53]、星火链网[54]以及 BSN (blockchain-based service network) 联盟链[12]开发的区块链软件最多. 例如华为链是基于 Hyperledger Fabric 开发, 而网易链是基于 FISCO BCOS 开发. 选择这些成熟平台进行开发在技术和生态方面都具有诸多优势. 在技术层面, 已有平台具有成熟稳定的基础架构、丰富的功能模块以及强大的安全性保障, 能够为开发者提供可靠的基础、缩短开发周期并降低安全风险. 在生态方面, 这些平台拥有庞大的开发者社区, 提供丰富学习资源和技术支持, 广泛的用户基础也使得软件上线后容易吸引用户并扩大规模, 良好的互联互通性便于与其他区块链系统交互和集成.

现象 9: 当前存在两个较为特殊的区块链软件——BSN 联盟链和星火链网, 国内许多区块链软件都是基于两者进行开发, 并且在其上实现跨链交互. 其中, BSN 联盟链是一个公共基础设施网络, 由国家信息中心牵头, 并与中国移动、中国银联等单位联合发起并建立. BSN 是基于区块链技术和共识机制的全球性基础设施网络, 是面向工业、企业、政府应用的可信、可控、可扩展的联盟链. BSN-DDC (blockchain-based service network-distributed digital certificate) 基础网络是 BSN 产品体系中专门面向我国市场的公网产品. 它通过建立一个公共分布式云服务网络, 为区块链和 NFT 等分布式技术以及相关应用在我国的发展提供基础设施能力支撑. BSN-DDC 网络由泰安链、文昌链、武汉链等各具特色的开放联盟链组成. 每条链由其技术方直接经营并提供基础支撑和优化迭代. 星火链网是中国信息通信研究院于 2021 年 8 月 3 日发布的底层区块链软件. BIF-Core 的发布标志着整个“星火链网”国家级的基础设施主链正式运行, 该技术将助力企业数字化转型和未来城市建设, 旨在实现不同区块链之间的互联互通和互操作. 通过构建统一的标识和标准体系, 星火链网为不同的区块链提供接入和交互的接口, 促进区块链

之间的数据流通、业务协同和价值传递, 打破区块链之间的孤岛效应, 推动区块链技术在更广泛的领域和场景中应用和发展. 目前接入星火链网的区块链软件包括: 蜀信链[55]、桂链[56]等.

见解 3: 成熟技术平台、公共基础设施的战略重要性以及跨链互操作性在推动区块链发展时发挥着关键作用.

3 国产区块链软件具有哪些核心技术?

本节, 我们将数据层、网络层、共识层、合约层、应用层以及其他技术这 6 个层次对具有详细技术信息的 39 个高质量国产区块链软件所使用的技术进行统计与分析, 图 7 展示了 6 个层次的具体信息.

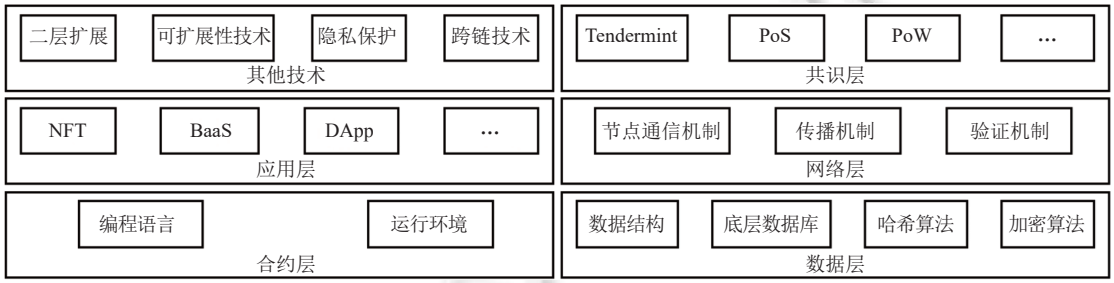


图 7 区块链软件技术分析层次

值得注意的是, 在统计区块链软件在不同层次所使用的技术时, 可能会出现大于或小于 39 的情况. 前者是因为一个区块链软件可能采用了多种技术, 如蚂蚁链[57]同时使用了 Merkle 树和 MPT (Merkle patricia trie)[44]两种数据结构. 后者是因为部分区块链软件的官网或技术文档中没有具体介绍所使用的技术, 如部分区块链软件仅声明其使用了对称加密算法, 但并未具体说明是哪种算法.

3.1 数据层

区块链技术是一种去中心化的分布式账本系统, 通过共识算法确保数据的安全和透明性. 其核心是数据层, 这是整个区块链系统的基础, 负责存储和管理区块链中的交易记录和状态信息. 数据层的有效运行依赖于 3 大关键概念: 数据结构、数据库和加密算法. 数据结构决定了数据的组织和访问模式, 数据库则负责保存和管理这些数据, 而加密算法则确保数据的安全性和隐私性. 在区块链软件中, 统计分析这些技术的使用情况和效率对于理解系统性能、优化资源管理、提升安全性具有重要意义. 这种分析有助于发现潜在的优化点, 提升区块链应用的整体表现和安全保障.

3.1.1 数据结构

区块链软件数据层的数据结构主要涉及区块链底层的区块和交易的存储方式, 例如比特币中用于验证交易的 Merkle 树、以太坊中管理账户状态和交易的 MPT, 以及星火链中用于组织交易的 DAG (directed acyclic graph).

Merkle 树作为一种数据结构, 能够快速验证交易的存在性和正确性, 已在比特币中得到了广泛应用. 随着智能合约的引入, 以太坊迅速发展, 并采用了一种名为 MPT 的数据结构. MPT 结合了 Merkle 树和 Patricia 树的优点, 更高效地处理和存储账户状态等信息. 与 MPT 和 Merkle 树不同, DAG 是一种专门用于区块存储的数据结构. 在 DAG 结构中, 每个新的数据单元可以指向多个之前的数据单元, 这在处理高并发交易时具有显著优势. 然而, DAG 也面临数据确认规则复杂和安全性验证难度大的挑战. 我们按照时间顺序对不同区块链软件数据层的数据结构进行了统计与分析, 结果如图 8 所示.

现象 10: 目前大多数国产区块链软件 (共 24 个) 在数据层中采用 Merkle 树, 仅有少量软件 (不超过 10 个) 采用 MPT 和 DAG 两种数据结构. 产生这一现象的原因主要包括以下几点: 1) 技术成熟度: Merkle 树技术简单易懂, 易于实现, 且具有较高的技术成熟度; 2) 适用性: Merkle 树完全能够满足大多数常见区块链应用的需求, 如验证交

易和区块的完整性; 3) 兼容性: Merkle 树能够兼容传统的区块链架构和共识机制. 与 Merkle 树相比, MPT 和 DAG 两种数据结构的使用相对较少, 这可能与其技术成熟度、应用场景受限以及开发难度等因素有关. 然而, MPT 不仅能够高效验证数据完整性, 还能提供对数据的快速查找和更新功能. DAG 具有高并发处理能力和高可扩展性, 能够支持大规模的分布式数据处理. 随着区块链软件的不断发展及对于软件性能的逐渐提升, 这两种类型的数据结构使用量可能会逐渐增加.

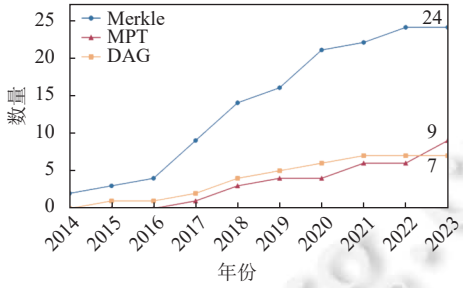


图 8 国产区块链软件数据层数据结构统计

3.1.2 数据库

数据库作为区块链技术架构中的基础层, 负责对系统中的所有交易和数据记录进行持久化存储和管理. 为了满足不同的应用需求和系统设计, 当前区块链系统使用的数据库类型呈现多样化发展. 根据数据库的类型和特性, 我们调查的国产区块链软件所使用的数据库可分为 3 种类型, 即键值型数据库 (key-value database)、关系型数据库 (relational database) 以及文档型数据库 (document database). 具体信息如表 3 所示.

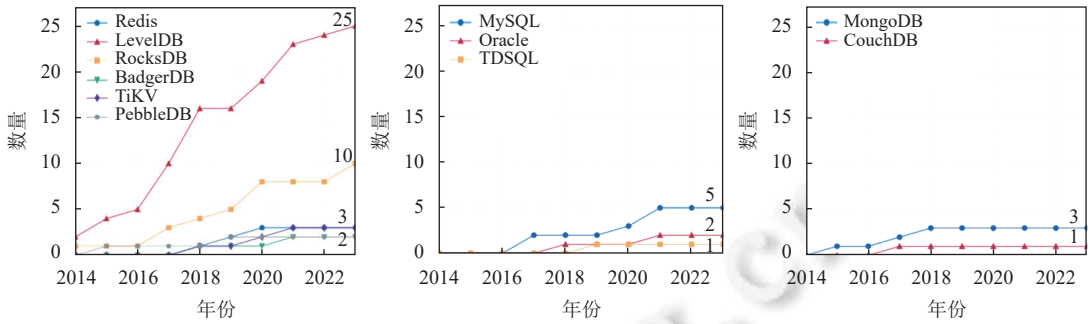
表 3 不同区块链软件数据库使用情况及分类

类别	数据库	区块链软件
键值型	LevelDB	NEO[58], 波场链[30], 万纳链[59], BubiChain[60], 趣链[61], 天河链[62], FISCO BCOS[53], 迅雷链[63], ONTology[64], 磁云数字区块链M0[65], 百度链[66], RepChain[67], Chain33[68], PlatON[33], 梧桐链[69], 唯链[26], 树图链[70], 华为链[71], IRITA[72], 长安链[36], 文昌链[73], 泰安链[51], 武汉链[74], 海峡链[75], 安顺链[50]
	RocksDB[76]	NEO, FISCO BCOS, 星云链[77], 京东链[78], 蚂蚁链, 星火链[54], 华为链, CKB[31], 延安链[79], 之江链[80]
	Redis[81]	京东链, 至信链, Aelf[82]
	BadgerDB[83]	Chain33, 长安链
	TiKV[84]	京东链, 星火链, 长安链
	PebbleDB[85]	万纳链, BNB Chain[29]
关系型	MySQL	天河链, FISCO BCOS, 华为链, 长安链, 泰安链
	Oracle	Mixin[86], 泰安链
	TDSQL[87]	至信链
文档型	MongoDB[88]	万纳链, 天河链, Mixin
	CouchDB[89]	Z-Ledger[90]

关系型数据库建立在关系模型的基础上, 采用二维表格的方式存储数据, 并通过行和列的形式组织数据, 确保数据的一致性和完整性[91]. 在所调查的区块链软件中, 常见的关系型数据库包括 MySQL、Oracle 和 TDSQL. 其中, TDSQL (Tencent distributed SQL) 是腾讯推出的一款兼容 MySQL 的分布式关系型数据库, 具有自主可控和高一致性的特点. 与关系型数据库不同, 非关系型数据库在数据存储方式和数据关系的管理上有显著差异. 例如, 键值型数据库使用简单的键值对存储数据, 每个键唯一地映射到一个值, 这使得数据查找速度快, 并且适合分布式存储和处理[92]. 在所调查的国产区块链软件中, 使用了 6 种具体的键值型数据库, 分别是 Redis、LevelDB、RocksDB、BadgerDB、TiKV 和 PebbleDB. 文档型数据库作为另一种非关系型数据库, 以文档为核心数据模型, 这些文档以键值对的形式存储, 并可包含复杂的数据结构, 如数组和子文档[93]. 在所调查的国产区块链软件中, 常见

的文档型数据库包括 MongoDB 和 CouchDB.

我们对不同区块链软件所使用或兼容的底层数据库进行了统计, 并按照时间顺序对新增区块链软件所采用的数据库情况进行了分析, 如图 9 所示.



(a) 键值型 (b) 关系型 (c) 文档型

图 9 新增国产区块链软件数据库使用/兼容情况

现象 11: 键值型数据库是当前国产区块链软件的主流选择. 具体而言, 接近 50% 的区块链软件支持 LevelDB、超过 15% 的区块链软件支持 RocksDB, 这两种键值型数据库的使用率超过了 65%. 相比之下, 支持或兼容关系型和文档型数据库的区块链软件仅有 8 个和 4 个. 这一现象的产生, 主要与键值型数据库在高性能、结构简单和强可扩展方面的优势相关. 虽然使用关系型数据库的区块链软件在与现有数据库系统的整合中可以降低复杂性, 并且关系型数据库能够支持复杂的业务逻辑, 但其通常存在性能瓶颈, 且复杂的数据结构也可能带来不必要的性能开销. 然而, 随着区块链技术的发展, 关系型数据库将在对数据一致性和复杂关系处理要求极高的领域中得到更广泛的应用, 例如在数字货币交易记录、复杂用户身份信息及其关联关系的存储和管理中. 另一方面, 文档型数据库因其适应快速开发和迭代的能力, 以及对非结构化数据的支持, 具有一定优势. 然而, 由于区块链中的数据结构相对固定和简单, 键值型数据库更能满足当前需求. 随着区块链应用的不断扩展, 文档型数据库的应用可能会逐渐增多, 特别是在处理物联网中的半结构化和非结构化数据, 以及在供应链溯源中处理包含图片和文字等多样化数据的场景中.

见解 4: 性能要求低、应用场景有限等因素导致当前区块链软件大多选择简单的数据结构或数据库, 但随着区块链软件的不断发展, 性能需求的增长以及应用类型增加, 具有并发数据处理能力以及高可扩展性的复杂数据库的使用量可能会逐渐上升.

3.1.3 加密算法

为了保证数据的安全性和完整性, 区块链技术结合了多种加密算法, 如 ECC (elliptic curve cryptography)[94]、SM2[95]、ECSDA (elliptic curve digital signature algorithm)[96]等. 其中保证数据安全性的算法可以根据加密方式不同分为对称加密和非对称加密两大类. 对称加密算法使用相同的密钥进行加密和解密, 而非对称加密则使用一对密钥, 公钥用于加密数据, 私钥用于解密数据[97]. 保证数据完整性的算法根据验证过程不同可以分为签名算法[98]和哈希算法[99]. 签名算法用于验证消息来源的完整性、真实性和不可否认性, 接收方通过发送方的公钥来验证签名的有效性. 具体地, 发送方使用签名算法对数据进行处理, 基于数据计算出一个独特的签名. 接收方在接收到数据和签名后, 使用发送方的公钥对签名进行验证. 如果数据在传输过程中被篡改, 计算出的签名将与接收到的签名不一致, 从而表明数据的完整性遭到破坏. 哈希算法则用于验证数据的完整性, 该算法具有雪崩效应[100], 即输入数据的微小变化会导致哈希值的巨大差异. 具体地, 发送方将数据和对应的哈希值一起发送给接收方, 接收方收到后对数据重新计算哈希值, 如果重新计算出的哈希值与接收到的哈希值一致, 那就说明数据在传输过程中没有被修改, 从而保证了数据的完整性. 加密算法的具体分类如表 4 所示.

从表 4 可以看出, 相较于比特币和以太坊平台, 国产区块链软件所使用的加密技术在种类和能力上有了显著

提升, 能够适应更复杂的应用场景. 其中, 较为特殊的加密算法包括国密算法、椭圆曲线加密算法、零知识证明算法以及秘密共享算法. 国密算法是由国家密码局认定的一系列国产加密算法, 主要包括 SM2 (椭圆曲线公钥密码算法)、SM3 (哈希算法) 和 SM4 (分组密码算法). 其中, SM2 是我国在 2010 年发布的一款自主设计的椭圆曲线公钥密码算法, 符合我国的安全需求, 并得到了广泛应用. SM4 是于 2006 年发布的一款对称加密算法, 广泛用于数据加密. ECC 是一种基于椭圆曲线数学理论的非对称加密算法. 除了 SM2 之外, Secp256k1[101]是另一种常见的椭圆曲线加密算法, 广泛用于比特币和以太坊等区块链系统中, 因其高安全性、高效率和广泛应用而受到青睐.

表 4　区块链加密算法分类

加密算法	算法类型	具体算法名称
保证安全性	对称加密算法	SM4, AES[102]
	非对称加密算法	SM2, Secp256k1, ElGamal[103], Paillier[104]
	其他	后量子密码, BulletProofs[105], 同态加密[106], Shamir[107], 代理重加密
保证完整性	签名算法	分布式签名, 环签名, ECDSA, BLS聚合签名, EDDSA, 群签名
	哈希算法	SM3, SHA-256[108], RIPEMD160, Keccak-256[109], BLAKE2[110]

传统加密算法在特定场景下存在一定的局限性. 例如, 当需要向第三方证明某些信息或完成某些操作, 但又不希望泄露具体细节时, 传统加密算法可能无法满足需求. 零知识证明 (zero knowledge proof, ZKP)[111]算法能够很好地满足这一需求, 在保护信息隐私的同时实现证明. 目前, 相关的零知识证明算法包括 BulletProofs++[112]和 zk-SNARKs[113]. BulletProofs++是一种高效且紧凑的零知识证明机制, 主要侧重于提高效率和降低证明大小; 而 zk-SNARKs 是一种强大且广泛应用的零知识证明体系, 以简洁性和非交互性为特点, 尤其在区块链等特定场景中表现出色. 在涉及多人协作且包含敏感信息的场景中, 秘密共享技术至关重要. 为此, 国产区块链软件分别使用了 Shamir 算法[107]和代理重加密算法[114]. Shamir 算法主要用于秘密共享和密钥管理, 例如在分布式系统中安全地分享密钥或在密码学中保护敏感信息, 常见应用场景包括密钥分发和秘密共享. 代理重加密算法则用于在不可信环境中实现安全的密文转换和共享. 它允许授权的代理将用一方公钥加密的密文转换为用另一方公钥加密的密文, 而代理无法获取原始明文. 该算法的主要应用场景包括数据贡献和云存储.

我们对不同区块链软件所使用的加密算法进行了统计, 结果如图 10 所示.

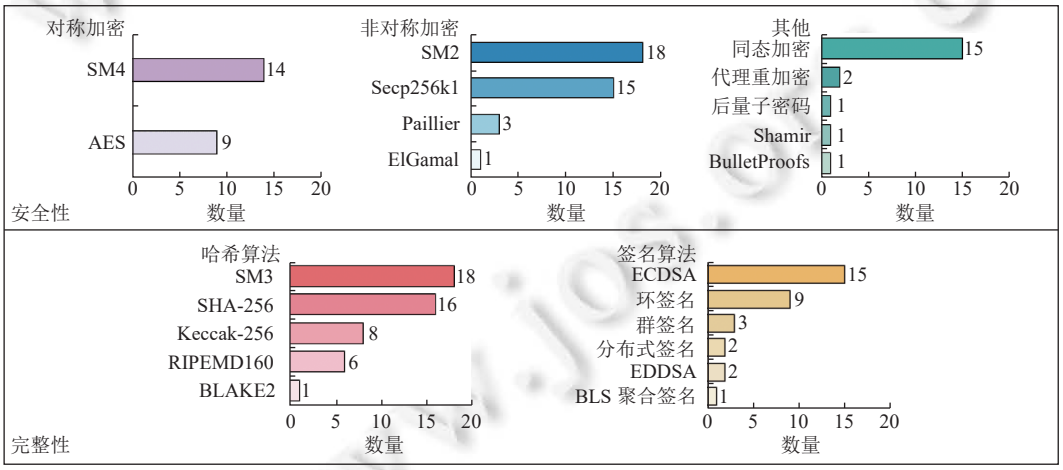


图 10　链加密算法使用情况

现象 12: 通过分析可以发现, 国产区块链软件在加密算法的选择上呈现多元化, 其中国产加密算法如 SM4、SM2、SM3 等广泛应用, 同时国际通用算法也因成熟度和兼容性被部分采用. 具体地, 在保证数据安全性的算法中, 共有 23 个区块链软件明确说明了其所使用的对称加密算法, 其中使用 SM4 的区块链软件有 14 个, 使用 AES

(advanced encryption standard) 的有 9 个. 这主要是由于政策导向下国产化需求使得 SM4 受到青睐, 同时 SM4 在性能和安全性上表现优异, 且有完善技术生态与社区支持. 尽管 AES[102]也具较高安全性和通用性, 但在国产区块链软件领域受 SM4 影响相对更大. 在非对称加密算法中, 除了 18 个区块链软件使用了 SM2 算法外, 15 个区块链软件使用了 Secp256k1 算法, 进一步证明了该算法的广泛应用. 在其他加密技术中, 同态加密算法的使用范围最广, 主要由于其灵活性强、适用范围广的特点.

在保证完整性的算法中, 共有 32 个区块链软件声明其所使用的签名算法, 其中使用较多的算法分别是 ECDSA (15 个) 和环签名算法 (9 个). 其中 ECDSA 的广泛使用得益于其高安全性、高效率和标准化特性, 而环签名算法则能够提供高水平的身份隐私保护, 并适应不同的应用场景和安全需求. 在哈希算法中, 18 个区块链软件支持国密哈希算法 SM3, 16 个支持 SHA-256 (secure hash algorithm 256). SM3 的广泛使用一方面得益于其安全性和适用性的不断提高, 另一方面得益于政策的推动. SHA-256 作为一种国际上广泛使用的哈希算法, 具有较高的知名度和成熟度, 考虑到兼容性问题, 大多数区块链软件也会选择支持该算法.

3.2 网络层

区块链网络层对于区块链至关重要, 它是连接区块链节点、实现信息交互和数据传输的关键层级. 网络层包含数据传输协议和传播机制两部分内容, 它们是确保区块链系统高效、稳定运行的关键要素. 其中, 数据传输协议如同信息传递的轨道, 规定了数据在节点之间流动的规则和方式, 决定了数据的封装格式、传输的可靠性以及效率. 常见的数据传输协议包括基于 HTTP/2 的 gRPC 等, 这些协议能够保障数据在网络中的快速、准确传输, 为区块链的稳定运行提供坚实的基础. 传播机制则像信息扩散的引擎, 当新的交易产生或新的区块被挖出时, 传播机制负责将这些关键信息迅速、有效地传递到网络中的各个节点. 它决定了信息是以广播的方式广泛传播, 还是通过特定的路由策略有针对性地传递.

3.2.1 数据传输协议

国产区块链软件所使用的数据传输协议主要包括 P2P、HTTPS 以及 gRPC. 其中, P2P (peer-to-peer) 协议是一种去中心化的网络架构, 节点间可以直接相互通信, 无需通过中央服务器[115]. P2P 协议在区块链网络中广泛应用, 因为它支持高效的分布式数据共享, 增强了网络的抗审查性和鲁棒性. HTTPS (hypertext transfer protocol secure) 是一种在 HTTP 基础上加入 SSL/TLS 加密层的传输协议, 主要用于确保数据在传输过程中的机密性和完整性[116]. gRPC 是由 Google 开发的高性能、开源的远程过程调用 (RPC) 框架, 基于 HTTP/2 协议, 并且支持多种语言和双向流式通信, 适用于需要低延迟和高吞吐量的区块链应用[117].

我们对不同国产区块链软件所使用的数据传输协议进行了统计, 具体如表 5 所示.

表 5 国产区块链软件使用数据传输协议情况

协议名	区块链软件
P2P	NEO, 波场链, 万纳链, BubiChain, 趣链, 天河链, FISCO BCOS, Z-Ledger, 迅雷链, 星云链, ONTology, 磁云数字区块链M0, 百度链, 京东链, RepChain, Chain33, Mixin, PlatON, 唯链, 蚂蚁链, 至信链, BNB Chain, 梧桐链, 星火链, 树图链, 华为链, IRITA, Aelf, CKB, 长安链, 文昌链, 泰安链, 武汉链, 海峡链, 中移链[118], 延安链, 安顺链, 之江链
gRPC	NEO, 波场链, 趣链, 星云链, 百度链, PlatON, BNB Chain, IRITA, Aelf, CKB, 文昌链, 泰安链, 延安链
HTTPS	BubiChain, FISCO BCOS

现象 13: 国产区块链软件普遍采用 P2P 协议, gRPC 和 HTTPS 的使用量较少. P2P 协议的广泛使用 (38 个), 主要得益于该协议的去中心化特性、高效的数据传输能力以及较低的运维成本. P2P 协议允许节点之间直接进行点对点通信, 避免了对中央服务器的依赖, 从而增强了网络的抗审查性和容错能力, 尤其适合去中心化的区块链网络. 相比之下, 使用 gRPC 协议的区块链软件数量相对较少 (13 个). 这是因为 gRPC 协议更适用于高效的远程过程调用 (RPC) 场景, 尤其在需要双向流式通信和低延迟、高吞吐量的应用中表现突出. 然而, 在区块链的典型场景中, 节点间的通信和数据分发往往更适合通过 P2P 这种分布式协议进行. 此外, gRPC 的实现和配置相对复杂, 增加了开发和运维的难度. 尽管如此, 随着区块链技术与边缘计算、物联网等新兴技术的不断融合, gRPC 在这些领域的设备间高效通信和数据传输需求下, 可能会获得更多的应用和关注.

HTTPS 协议的使用最少 (2 个), 这是因为 HTTPS 相较于 P2P 和 gRPC, 其在性能上存在一定的开销, 且有一定的中心化倾向. 此外, HTTPS 在区块链中的应用较为有限, 主要适用于需要强安全性和合规性的数据传输场景, 特别是金融领域的区块链应用. 在这些受严格监管的行业中, 法律和合规要求可能会促使更多地使用 HTTPS 来确保数据传输的安全性和合规性.

见解 5: 当前国产区块链软件普遍采用 P2P 协议是合理选择, 因其契合区块链去中心化需求, 而 gRPC 和 HTTPS 在当前适用性较弱; 未来随着技术融合与特定领域需求, gRPC 和 HTTPS 可能在不同场景发挥更大作用, 区块链软件协议选择将更加多元化.

3.2.2 传播机制

与传统互联网的数据传播机制相比, 区块链的传播机制具有更高的去中心化程度, 并且不需要依赖第三方的信任背书. 区块链的主要数据传播机制包括 Gossip[119]、洪泛[120]、PubSub[121]、选择性转发等. Gossip 协议是一种分布式信息传播协议, 节点随机选择其他节点进行信息交换. 通过节点之间的定期信息交换, 信息逐渐传播到整个网络. Gossip 协议具有简单、可扩展性强的优点, 但其传播速度相对较慢, 且可能产生消息冗余. 洪泛机制中, 源节点将信息发送给所有邻居节点, 而这些邻居节点接收到信息后, 再转发给各自的邻居节点, 直到信息覆盖整个网络或达到预设的传播范围. 虽然这种机制传播速度快, 但会产生大量冗余消息, 可能导致网络拥塞. PubSub 机制通过发布者 (Publisher) 和订阅者 (Subscriber) 之间的互动来实现信息传播. 发布者将信息发布到特定的主题, 订阅了该主题的订阅者会接收到相应的信息. 该机制能够实现信息的精准推送, 减少不必要的消息传播, 但需要一个中心化或分布式的协调组件来管理主题和订阅关系. 选择性转发机制中, 节点不会盲目地转发所有接收到的信息, 而是根据一定的策略 (如基于内容、目的地、网络状况等) 选择部分信息进行转发. 这种机制能够减少网络中的冗余流量, 提高传播效率, 但也需要更复杂的决策逻辑. 不同区块链软件所使用的传播机制如表 6 所示.

表 6 国产区块链软件使用传播机制情况

机制名	区块链软件
洪泛	趣链, 百度链, PlatON, 星火链, 中移链
Gossip	FISCO BCOS, Z-Ledger, 星云链, RepChain, Chain33, PlatON, Mixin, 唯链, 树图链, CKB, 泰安链, 延安链
选择性转发	趣链, FISCO BCOS, 中移链
PubSub	FISCO BCOS, RepChain

现象 14: 大部分区块链软件使用 Gossip 协议进行区块传播 (12 个), 而其他传播机制的使用相对较少. Gossip 协议被广泛采用, 主要是因为它在传播效率和网络资源开销之间取得了良好的平衡. 此外, Gossip 的随机信息交换模式提高了网络的容错性, 适合大规模去中心化网络的需求. 相比之下, 洪泛机制 (5 个) 的使用较少, 主要是因为该机制容易导致大量消息冗余, 消耗过多的网络资源. 在大规模网络中, 洪泛机制的性能往往急剧下降, 增加了网络拥塞的风险. 选择性转发机制 (3 个) 虽然能够减少冗余流量并提高传播效率, 但其复杂的策略设计和调整要求导致开发和维护成本较高, 因此被较少使用. PubSub 机制 (3 个) 则由于其适用场景有限, 并且依赖特定的发布/订阅中间件, 增加了系统的潜在故障点和复杂性, 因此也不如 Gossip 普及. 此外, 统计的区块链软件数量少于 39, 这是因为部分区块链软件并未在官网、技术文档或开发文档中明确声明其使用的传播机制.

3.3 共识层

区块链共识层通过确保分布式网络中的所有节点对交易和区块达成一致, 从而实现去中心化的信任机制, 保障区块链系统的安全性、完整性和一致性, 其主要内容为共识协议. 相比传统的中心化共识, 去中心化共识具备稳定性高、参与广泛的特点、但效率较低. 当前国产区块链软件使用的共识协议大致可以分为 4 类: 证明类共识协议 (proof of work, PoW)[122]、投票类共识协议 (proof-of-stake, PoS)[123]、拜占庭容错 (Byzantine fault tolerance, BFT) 类共识协议[124,125], 以及混合类共识协议[126].

证明类共识要求节点通过完成特定的计算任务或提供某种形式的证明来达成共识, 节点需要证明自己对特定资源的投入或对系统规则的遵守, 以此来确定交易或区块的合法性. 投票类共识则通过投票来表达对交易或区块

的认可或反对, 并根据投票结果决定是否达成共识. 拜占庭容错类共识旨在解决分布式系统中, 即使存在恶意或故障节点 (即拜占庭节点) 的情况下, 如何实现可靠的共识. 混合类共识则融合了多种不同类型的共识机制, 结合各自的优势来增强系统的性能和安全性. 我们对国产区块链软件所使用的共识协议进行了统计, 如图 11 所示. 图中不同颜色表示不同的共识协议类型, 方块表示对应共识协议在国产区块链软件中的使用数量. 例如, 有 5 个国产区块链软件支持 PoW 协议.

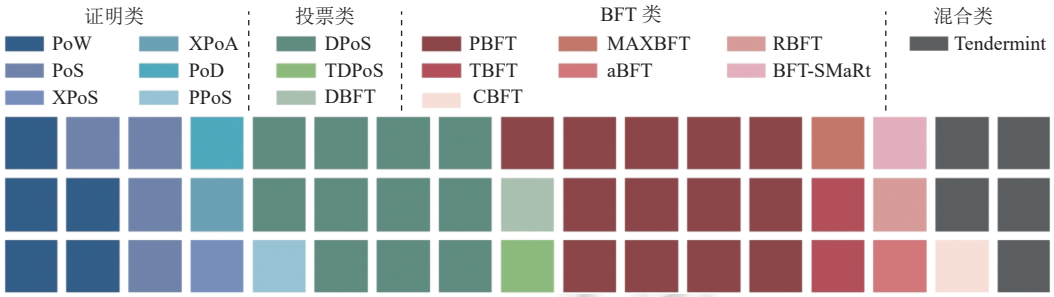


图 11 不同区块链软件共识协议图

现象 15: 支持 BFT 类共识协议的区块链软件最多 (20 个), 支持投票类和证明类共识协议的区块链软件相对较少 (13 个), 支持混合类共识协议的区块链软件最少 (5 个).

BFT 协议使用较多主要是因为该协议在容错能力和安全性方面表现良好, 适合私有链和联盟链环境, 因此被广泛应用. BFT 协议能够处理节点之间的信任问题, 这在商业环境中尤为重要. 在 BFT 类协议中, 经典的拜占庭容错算法 PBFT (practical Byzantine fault tolerance)[127]使用最广泛, 其他协议则在 PBFT 的基础上进行了优化. 例如, TBFT (Tendermint Byzantine fault tolerance)[128]在 PBFT 的基础上引入了特定的阈值条件或优化策略, 以提高性能或适应不同的应用场景. MAXBFT[129]则在特定条件下对 PBFT 进行性能优化. 投票类协议 (如 DPoS) 通常适合参与者数量较少的网络, 面对较大规模的网络时可能会遇到性能瓶颈. 证明类协议 (如工作量证明和权益证明) 在公共链中较为常见, 但在我国, 监管政策可能对这类协议的应用产生了影响, 尤其是对挖矿活动的限制. 支持混合类共识算法 Tendermint[126]的区块链软件有 5 个. 与证明类和投票类共识算法相比, Tendermint 具有更高的效率, 且在节点数量增加时, 其性能下降相对较缓, 展现了更好的可扩展性. 因此, 未来混合类共识算法在区块链领域仍然具有较大的潜力. 此外, 我们还观察到一些国产区块链软件使用的共识协议之间存在着一定的关系 (现象 16), 如图 12 所示.

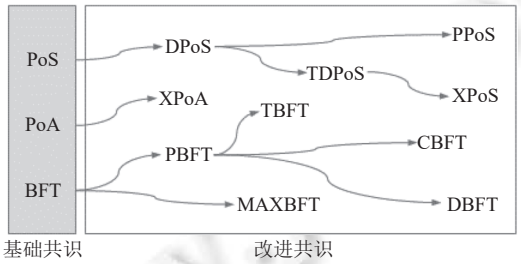


图 12 共识协议关系

委托权益证明 (delegated proof of stake, DPoS) 是 PoS 的一种改进版本. PoS 是根据节点持有代币的数量和时间来决定记账权, 而 DPoS 在 PoS 的基础上, 通过持币者投票选出代表节点, 由这些代表节点负责记账和验证交易. DPoS 进一步提高了交易处理速度和网络效率, 同时增强了社区成员对网络决策和运营的参与度. PlantON 和百度链在 DPoS 的基础上进行了改进, 提出了 TDPoS (trusted delegated proof of stake) 和 PPoS (PlatON proof of stake) 两种共识协议. TDPoS 是百度链自研的一种类似 DPoS 的算法, 它允许节点抵押治理 Token 以投票选举候

选节点, 系统根据治理 Token 的抵押总数进行排序, 选定候选节点后, 候选人组按照时间片进行轮换. 百度链提出的 XPoS (xuper proof of stake) 在 TDPoS 的基础上增加了 Chained-BFT 组件, 以确保系统在指定区块高度下账本不可回滚, 保证业务数据的持久性. 其中, Chained-BFT 是一种将区块链结构与拜占庭容错机制结合的共识协议, 通过将共识决策嵌入区块链的线性结构中, 实现高效、可验证的拜占庭容错共识. PlatON 的共识协议 PPoS 通过在小规模备选节点中利用 VRF (verifiable random function)[130]和概率分布随机选取验证节点参与 BFT 共识, 在保证共识效率的同时避免过于中心化, 提升系统安全性和去中心化程度. XPoA (xuper proof of authority) 同样是百度链自研的共识协议, 适用于联盟链场景, 节点可以使用专有合约进行权益转移, 并根据业务需求对联盟链中的共识节点进行变更, 变更需经过联盟内指定节点的签名.

实用拜占庭容错 (PBFT) 是一种在分布式系统中能够容忍拜占庭错误的共识算法, 通过多节点交互和投票机制确保系统的一致性和可靠性. 长安链、PlatON 和 NEO 分别在 PBFT 的基础上进行了改进, 提出了 TBFT、CBFT 以及 DBFT. 其中, 长安链的 TBFT (trusted Byzantine fault tolerance) 是一种高度可靠的共识机制. 它在面对分布式系统中可能出现的拜占庭错误时, 通过一系列严格的验证和交互流程, 确保各节点在复杂的网络环境下能够达成一致的交易确认, 为长安链的稳定运行提供了坚实的基础, 保障了数据的完整性和系统的安全性. PlatON 的 CBFT (consensus Byzantine fault tolerance) 是 PlatON 网络独有的共识机制, 融合了先进的技术理念和创新的设计思路, 在确保系统具备强大的容错能力的同时, 能够高效地处理大量的交易请求. 通过多节点之间的协作和信息交互, CBFT 能够快速地达成共识, 保障 PlatON 网络的稳定运行和数据的安全可靠. NEO 的 DBFT (delegated Byzantine fault tolerance) 是一种具有创新性的共识算法. 它通过委托的方式, 让一部分节点代表全体参与共识过程, 有效地提高了共识的效率. 在面对可能的拜占庭错误时, DBFT 能够通过严格的验证和投票机制, 确保交易的合法性和系统的稳定性, 为 NEO 区块链的发展提供了强大的技术支撑. 此外, 长安链在 BFT 的基础上进行了改进, 得到了 MAXBFT, 该协议是一种高效的拜占庭容错共识机制, 采用先进密码学技术和分布式算法, 具备强大容错能力、性能优化及可扩展性, 为长安链提供安全可靠的技术支撑. 它能在面对拜占庭错误时确保系统一致性, 满足处理高并发交易的需求, 适应网络发展扩大.

见解 6: BFT 类共识协议在当前得到了广泛应用, 投票类与证明类共识协议的局限性限制了其使用, 混合类共识协议具有较强的应用潜力. 同时, 一些国产区块链软件提出独特共识协议, 展现了国内在区块链共识机制创新方面的积极探索, 未来有望持续推动区块链技术在不同场景下的优化与发展.

3.4 合约层

区块链的合约层是实现智能合约执行和自动化协议管理的核心, 使区块链能够在无需信任中介的情况下, 自动、安全地执行预定的合约条款. 该层构建于区块链虚拟机之上, 包含了各种商业逻辑和算法, 是实现区块链系统灵活编程和数据操作的基础. 在这一层中, 智能合约编程语言 (如 Solidity 等) 扮演着关键角色, 提供了一种精确且高效的方式, 供开发者定义和描述合约的功能、条件以及操作流程. 同时, 智能合约执行引擎负责解析和执行智能合约代码, 处理合约的逻辑与运算, 以实现其预定功能.

3.4.1 合约编程语言

智能合约的概念最早由以太坊提出并成功应用于实践, 使得区块链技术能够支持去中心化金融、社会治理等诸多应用场景. 相比之下, 比特币等早期加密货币平台仅使用了非图灵完备的简单脚本语言来编写代码, 这些代码主要用于控制交易的执行, 功能较为有限. 由于我们所研究和收集的区块链软件并不涉及加密货币平台, 因此在讨论合约层时, 我们将不包括对这些简单脚本语言的分析, 而是专注于更复杂和功能丰富的智能合约体系. 这些智能合约不仅支持图灵完备的编程语言, 还能够实现更为复杂的自动化逻辑, 推动区块链技术在更广泛领域的应用.

在统计国产区块链软件在智能合约编程语言方面的使用情况前, 我们首先对智能合约编程语言进行了分类, 包括: 1) 专为智能合约编程设计的语言; 2) 传统编程语言. 随后, 我们对每种类型编程语言中具体语言以及对应的区块链软件进行了统计, 如表 7 所示.

表 7 智能合约编程语言分类

编程语言分类	具体语言	区块链软件
专为智能合约编程设计的语言	Solidity	蚂蚁链, 长安链, 百度链, 趣链, 天河链, FISCO BCOS, 星火链, 树图链, BNB Chain, 华为链, 迅雷链, 海峡链, 万纳链, 波场链, CKB, IRISnet, PlatON, Mixin, 磁云数字区块链M0, 延安链, 文昌链, 泰安链, 武汉链, 安顺链, 唯链, 之江链
传统编程语言	Go	蚂蚁链, 长安链, 百度链, 趣链, 天河链, 至信链, BSC, 华为链, Z-Ledger, 海峡链, Chain33, IRITA, NEO, CKB, Mixin, 梧桐链, 之江链
	C++	蚂蚁链, 长安链, 百度链, 天河链, FISCO BCOS, 迅雷链, 万纳链, BubiChain, IRITA, PlatON, 磁云数字区块链M0, 中移链, 之江链
	Java	蚂蚁链, 百度链, 趣链, 天河链, 京东链, BSC, 华为链, 海峡链, IRITA, NEO, ONTology, PlatON, 之江链
	C	迅雷链, BubiChain, IRITA, NEO, Aelf, ONTology, CKB, 磁云数字区块链M0
	JS	京东链, 星火链, RepChain, BubiChain, NEO, 星云链, ONTology, 波场链, CKB
	Python	京东链, BubiChain, IRITA, NEO, ONTology, CKB, PlatON
	Rust	长安链, 京东链, IRITA, 文昌链
	C#	NEO, Aelf, ONTology
	TypeScript	蚂蚁链, NEO, 星云链
	AssemblyScript	华为链, 之江链
	Scala	RepChain
	Swift	BSC
	TinyGo	长安链

第 1 类语言是专门为区块链上的智能合约开发而设计的新兴编程语言. 这些语言的设计初衷就是为了应对区块链环境的独特需求和挑战, 如分布式账本的管理、加密算法的运用以及智能合约的执行效率和安全性. 这些语言通常具备特定的语法和功能, 能够更高效地处理区块链系统中的复杂逻辑, 同时减少合约开发中的潜在漏洞和安全风险. 例如, Solidity 和 Vyper[131]等语言就是为此目的而开发, 它们不仅支持图灵完备, 还为智能合约的开发者提供了强大的工具集, 使得开发过程更加便捷和安全. 第 2 类是传统编程语言, 如 C++、Java、Python 等. 这类最初并非为区块链或智能合约设计, 但它们拥有广泛的开发者社区和丰富的开发资源, 已经被逐步适应并用于智能合约的开发. 这些语言通常通过特定的框架和运行环境, 如 WebAssembly[132]或 JVM (Java virtual machine), 被引入区块链开发中. 然而, 由于区块链的独特性, 在使用这些传统语言编写智能合约时, 往往需要进行额外的适配和优化工作, 以处理诸如交易的原子性保障、合约的安全性检查以及性能的调优等问题.

不同区块链软件可能会支持多种智能合约编程语言, 因此我们对国产区块链软件支持编程语言情况进行了统计, 如图 13 所示. 通过这项统计, 我们可以了解不同类型的编程语言在国产区块链开发中的应用情况, 以及它们在智能合约开发中的优势与不足.

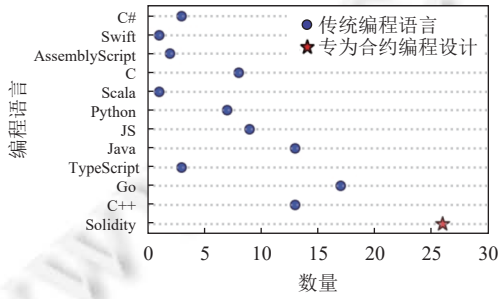


图 13 国产区块链软件支持编程语言情况

通过分析表 7 和图 13, 可以得到国产区块链软件在编程语言支持方面的两个现象.

现象 17: 目前国产区块链软件支持的编程语言中, Solidity 依然占据主导地位. 主要原因有两个方面: 首先, 以

太坊在区块链领域的影响力巨大, 许多区块链软件直接或间接采用了以太坊的技术框架. 其次, 围绕 Solidity 语言的开发资源和工具链相当丰富, 活跃的开发者社区提供了大量的学习资源和支持, 使得该语言在智能合约开发中占据了重要位置. 其次是 Go 语言, 其支持量仅次于 Solidity. 这主要归功于 Hyperledger Fabric, 这是一个在企业级区块链领域广泛应用的框架, 具有较高的影响力. Fabric 的许多核心模块和功能都是使用 Go 语言开发的, 因此在国产的区块链软件架构中, 往往能看到 Fabric 的影子. 此外, Fabric 还支持使用 JavaScript 和 TypeScript 开发链码(即智能合约), 这进一步扩大了它的适用范围和影响力. 随着区块链技术的发展, 对性能和可扩展性的需求日益增加, Java 和 C++这两种传统编程语言也被越来越多地用于开发智能合约. 这两种语言在处理大规模数据和高并发场景方面表现优异, 尤其是在需要跨平台部署的应用中, 它们的优势更为明显. Java 和 C++的技术成熟度和广泛的开发者基础, 也将使它们成为智能合约开发的热门选择.

现象 18: 一些小众的智能合约编程语言也得到了国产区块链软件的支持. 例如, Scala 作为一种在大数据处理领域颇具影响力的语言, 广泛应用于如 Spark 这样的框架中. 随着区块链与大数据技术的逐渐融合, 支持 Scala 使得区块链能够更方便地与大数据技术整合. Swift 则是苹果公司开发的编程语言, 在苹果的生态系统中有广泛应用. 随着区块链技术在移动端的发展, 支持 Swift 有助于区块链应用与苹果设备上的软件进行无缝集成与交互.

见解 7: 随着对性能和可扩展性需求的上升, Java 和 C++作为传统编程语言, 越来越多地被用于智能合约开发. 这两种语言在处理大规模数据和高并发场景中表现优异, 表明市场对高效能解决方案的需求正在推动对区块链项目这些成熟技术的关注; 一些小众语言如 Scala 和 Swift 的引入, 表明区块链技术正在与其他领域 (如大数据和移动端应用) 逐步融合. 这种趋势不仅扩大了区块链技术的应用场景, 也为开发者提供了更多选择, 增强了市场的多样性.

3.4.2 合约执行引擎

智能合约执行引擎是智能合约系统的核心组件, 负责读取、解析和执行智能合约的代码. 它根据预定的规则和逻辑处理输入数据, 进行相应的计算和状态更改, 并最终输出执行结果. 国产区块链软件中所使用的合约执行引擎可以大致分为 4 类: 虚拟机类、容器类、新兴格式类以及其他类型. 每种类型下包含的具体执行引擎以及对应的区块链软件如表 8 所示.

表 8 智能合约执行引擎分类

执行引擎类型	具体执行引擎	区块链软件
虚拟机类	EVM	蚂蚁链, 长安链, 百度链, 趣链, 天河链, FISCO BCOS, 星火链, 树图链, BSC, Z-Ledger, 迅雷链, 万纳链, Chain33, IRITA, ONTology, CKB, IRISnet, PlatON, Mixin, 磁云数字区块链M0, 延安链, 文昌链, 泰安链, 武汉链, 安顺链, 唯链, 之江链
	JVM	Z-Ledger, RepChain, Chain33, Aelf
	NeoVM	NEO, ONTology
	Native	百度链, 之江链
	XVM	百度链
	HVM	趣链
	BVM	趣链
	GraalVM	京东链
	NVM	星云链
	CKB-VM	CKB
	CVM	至信链
	MYVM	之江链
	TVM	波场链
容器类	Docker	长安链, 华为链, Z-Ledger, Aelf, 梧桐链
	Kubernetes	华为链
新兴执行格式类	WASM[132]	蚂蚁链, 长安链, FISCO BCOS, 华为链, 迅雷链, 万纳链, Chain33, BubiChain, IRITA, ONTology, PlatON, 磁云数字区块链M0, 中移链, 延安链, 文昌链
	SC-Runtime	海峡链

虚拟机类执行引擎, 例如 EVM, 具有独立的指令集和运行逻辑, 能够提供封闭且安全的执行环境, 有效地隔离不同智能合约之间的影响, 确保合约执行的确定性和安全性. 类似的执行引擎还有京东链的 GraalVM、百度链的 XVM 与 Native、趣链的 BVM 与 HVM、NEO 的 NeoVM、星云链的 NVM、CKB 的 CKBVM、至信链的 CVM、之江链的 MYVM, 以及波场链的 TVM. 这些虚拟机类的合约执行引擎都是根据国产区块链软件在特定应用场景下的需求和性能优化目标而开发的. 容器类执行引擎, 例如 Docker[133]和 Kubernetes[134], 提供了轻量级的隔离和资源管理, 使智能合约能够在独立的环境中高效运行, 并具有良好的资源利用效率和部署灵活性. 新兴执行格式类, 如基于 WebAssembly (WASM) 技术的执行环境, 具有高效、跨平台的优势, 适应了智能合约在性能和功能需求不断发展的趋势.

我们对国产区块链软件所支持的不同类型的合约执行引擎进行了统计, 结果如图 14 所示.

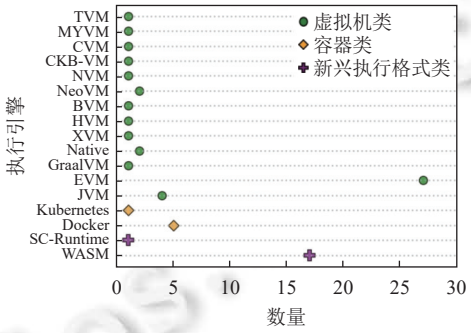


图 14　不同区块链软件支持合约执行引擎情况

现象 19: 国产区块链软件支持最多的 3 种合约执行引擎分别是: EVM、WASM 和 Docker. EVM 被广泛支持是因为以太坊在区块链领域的巨大影响力和广泛的开发者基础. 许多区块链项目直接或间接借鉴了以太坊的技术架构, 而 EVM 作为以太坊智能合约的执行环境, 具有成熟的生态系统和丰富的开发工具, 使得开发者能够快速上手并构建智能合约应用. WASM 受到支持是因为它具有高效性、可移植性和跨平台特性. WASM 可以在多种编程语言中编译生成, 能够在不同的区块链平台上高效执行, 为开发者提供更多的选择和灵活性, 同时也有助于提高区块链应用的性能和可扩展性. Docker 被众多国产区块链软件支持是因为它提供了一种轻量级的容器化解决方案, 可以方便地打包和部署区块链应用, 实现应用的快速部署和隔离, 提高开发和运维效率. 同时, Docker 的广泛应用也使得开发者更容易将现有的应用和工具集成到区块链环境中.

现象 20: 虽然 JVM 仅有 4 个区块链软件使用, 但其他区块链软件通过其他执行引擎支持 Java 语言. 除了 Z-Ledger、RepChain、Chain33 以及 Aelf 这 4 条直接使用 JVM 的区块链外, 趣链首创的 HVM、京东链支持的 GraalVM、NEO 开发的 NeoVm 同样能够支持 Java 语言. HVM 内置数据表结构, 可以实现业务数据可视化, 在保证智能合约执行的安全性、确定性、可终止性的前提下, 提供了一系列灵活的应用模式和工具方法集, 以满足复杂多样的业务场景需求. 京东链支持的 GraalVM 除了能够加速用 Java 和其他 JVM 编写的应用程序执行外, 同时还支持 JavaScript、Ruby、Python 和许多其他流行语言. GraalVM 的多语言功能可以在单个应用程序中混合多种编程语言, 同时消除外语调用成本. NeoVM 是一个用于执行 NEO 智能合约的轻量级虚拟机. 同时结合 NeoCompiler 编译器技术, 可将 Java、C#等高级语言编译为统一的 NeoVM 指令集, 从而实现跨平台. 可以看到, 传统编程语言在区块链领域得到了一定程度的应用, 这与 Java 成熟的开发生态、安全性和稳定性等因素相关. 并且 JVM 还能够提供跨平台执行的能力, 这意味着用 Java 编写的智能合约可以在不同的操作系统和硬件平台上运行, 无需进行大量的移植工作. 这对于需要在多种环境中部署的区块链应用来说非常重要.

见解 8: 国产区块链软件对不同合约执行引擎的支持体现了技术选择多元化, 以满足不同场景需求. 尽管传统编程语言在区块链领域虽有限应用但有价值, 但其成熟生态等优势与区块链结合带来了更多可能.

3.5　应用层

袁勇等人[37]依据区块链技术应用的现状, 将区块链的主要应用大致归纳为数字货币、数据存储、数据鉴证、

金融交易、资产管理和选举投票这 6 个场景. 根据收集到的国产区块链软件官网信息, 我们对相关应用场景进行了归纳整合, 最终确定了 6 个场景, 分别为: 一是金融服务, 即利用区块链技术实现更加安全、高效且透明的各类金融交易. 得益于区块链去中心化、防篡改和可追溯的特性, 金融机构可通过智能合约实现自动清算与结算, 降低中介成本, 提升交易效率; 二是智慧政务与智慧城市, 通过区块链提升政务数据的安全性、可信度和共享性, 有助于打破部门数据壁垒, 实现政务系统之间的协同与透明; 三是溯源存证与供应链管理, 区块链天然适合用于记录商品或信息在生产、运输、加工、销售等环节的全流程流转, 确保来源可追溯、过程可验证、去向可追踪、责任可追究; 四是区块链即服务 (BaaS), BaaS 平台通过云计算提供基础区块链设施与开发环境, 使得企业和开发者无需自建底层系统, 即可快速搭建、部署与管理区块链应用. 该模式大大降低了区块链开发门槛与技术成本, 助力中小企业、传统行业更便捷地实现数字化转型; 五是非同质化代币 (non-fungible token, NFT), 是一种具有唯一性、不可分割、不可替代特性的数字资产, 每一个 NFT 在区块链上都有独一无二的标识, 广泛应用于数字艺术品、虚拟地产、游戏资产、门票凭证等场景; 六是去中心化应用 (decentralized application, DApp), DApp 运行在区块链上, 通过智能合约实现逻辑执行. 其核心优势在于抗审查、高透明、可自主管理, 广泛用于 DeFi、DAO (decentralized autonomous organization) 治理、身份认证等领域.

我们对不同区块链软件已实现的应用进行了统计, 这些信息大多来源于区块链软件官网, 结果如表 9 所示.

表 9　不同区块链软件应用

区块链软件	金融服务	智慧政务、城市	溯源存证与供应链管理	BaaS	NFT	DApp
蚂蚁链	√	√	√	—	—	—
长安链	√	√	√	—	—	—
百度链	√	√	√	—	√	—
趣链	√	√	√	—	√	—
天河链	√	√	√	√	√	—
京东链	√	√	√	—	—	—
FISCO BCOS	√	—	√	—	—	—
至信链	√	√	√	√	√	—
星火链	√	√	√	—	—	—
树图链	√	—	—	—	√	—
币安链	√	—	—	—	—	√
华为链	√	√	√	√	—	—
Z-Ledger	√	√	√	—	—	—
RepChain	√	√	√	√	—	—
迅雷链	√	√	√	√	√	—
海峡链	√	—	√	—	√	—
万纳链	√	—	√	—	√	—
Chain33	—	—	√	—	—	—
BubiChain	√	—	√	—	√	—
IRITA	√	√	√	—	—	—
NEO	—	—	—	—	√	—
Aelf	√	—	—	—	√	√
星云链	—	—	—	—	√	—
ONTology	√	—	√	—	—	—
波场链	√	—	—	—	√	√
CKB	√	—	—	—	√	√
IRISnet	√	—	—	—	√	√
PlatON	√	—	—	—	√	√
Mixin	√	—	—	—	√	√
梧桐链	—	√	√	—	—	—

表 9 不同区块链软件应用 (续)

区块链软件	金融服务	智慧政务、城市	溯源存证与供应链管理	BaaS	NFT	DApp
磁云数字区块链M0	√	√	√	—	—	—
中移链	√	√	√	√	—	—
延安链	—	√	√	√	—	—
文昌链	—	√	√	—	—	—
泰安链	—	—	√	—	—	—
武汉链	—	—	—	—	—	—
安顺链	—	—	√	—	√	—
唯链	—	—	—	—	√	—
之江链	√	√	√	—	—	—
合计	29	19	27	7	19	7

现象 21: 表 9 中最后一行记录了每种应用对应的区块链软件总数. 可以看到, 已有区块链软件大部分都实现了与金融服务、智慧政务、城市以及溯源存证与供应链管理相关的应用. 其中支持金融服务的国产区块链软件最多 (30), 支持溯源存证与供应链管理的软件数量接近 (29), 支持智慧政务和城市的软件数量相对较少 (22), 支持 DApp 的软件数量最少, 仅为 7.

痛点切合度高、市场潜力大以及政策推动等因素是支持金融服务区块链软件数量多的原因. 首先, 传统金融行业存在信任壁垒高、交易结算流程复杂成本高、金融监管和信息安全隐患大等痛点. 区块链技术的去中心化、数据不可篡改性和对私密数据的加密保护等特点, 正好与金融行业的本质需求高度契合. 例如, 区块链可以为交易双方建立信用基础, 解决信任问题; 点对点的交易方式无需第三方中介, 可提升交易效率、降低成本; 分布式系统和权限管理能保障信息安全和监管的有效性. 其次, 金融领域的业务规模庞大, 交易频繁, 资金流动量大. 区块链技术在金融领域的应用, 如跨境支付、供应链金融、贸易融资、智能证券、票据、征信等, 能够带来巨大的经济效益和效率提升. 此外, 政府对金融科技的发展持积极支持的态度, 将区块链作为核心技术自主创新的重要突破口, 出台了一系列政策鼓励金融机构加强金融科技战略部署, 强化区块链等技术在金融领域的应用. 政策的支持为支持金融服务的区块链软件的发展提供了良好的政策环境.

而技术适用性强、产业需求迫切以及数据基础较好等隐私使得大量区块链软件支持溯源存证与供应链管理. 区块链的分布式账本、不可篡改和可追溯等特性, 天然适用于溯源存证和供应链管理. 在供应链管理中, 能够实现供应链上各环节参与者的信息及时共享、准确记录和追溯, 提高供应链的透明度和协同效率, 降低信任成本和风险. 对于产业需求迫切这点, 在当前的商业环境下, 消费者对产品的质量、来源和安全性越来越关注, 企业也需要提高供应链的管理水平和透明度来增强竞争力. 数据基础较好是指供应链管理和溯源存证涉及大量的数据记录和传输, 而许多企业在信息化建设方面已经有了一定的基础, 具备了将区块链技术应用于现有业务流程的条件. 这使得区块链在这两个领域的应用落地相对容易, 推动了相关软件的发展.

国产区块链软件中支持 DApp 功能的产品相对较少, 主要受国内监管环境和应用定位影响. 国内区块链更强调“去币化”“产业化”应用, App 的金融属性与开放式智能合约生态受到限制, 导致生态开发者数量和应用场景不足. 相较海外公链, 产区块链平台在面向开发者的智能合约支持与商业化 DApp 生态建设方面发展较慢.

进一步地, 我们通过统计探索了区块链软件应用与地域以及链类型之间的关系, 如图 15 所示.

现象 22: 从图 15 中可以看到, 属于北京、广东和上海的区块链软件总共支持的应用数量多, 分别是 41、25、14. 从应用类型的角度看, 属于北京的区块链软件支持所有类型的应用. 从链类型的角度来看, 超过 75% 的区块链软件应用是基于联盟链开发的.

见解 9: 金融服务和溯源管理类应用在区块链软件应用中具有核心地位, 随着政府对数字化转型的重视, 智慧政务类应用将逐渐增多, 具有很大的增长潜力. 区域间发展不平衡以及联盟链的主流地位反映出当前国产区块链软件在应用领域缺乏多样性.

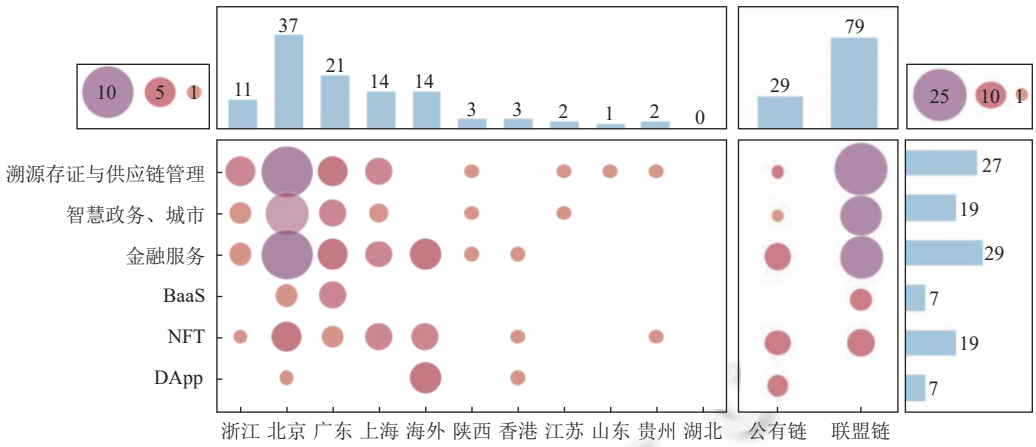
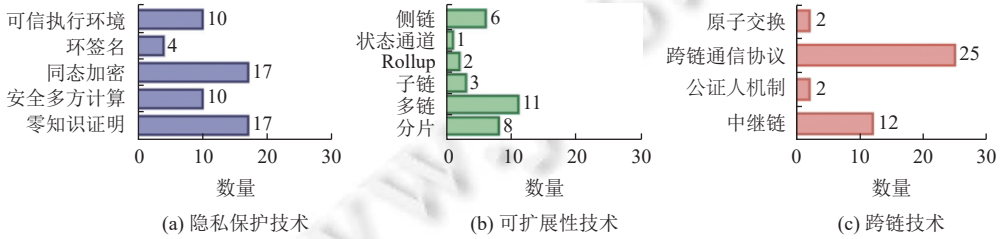


图 15 区块链软件应用数量与地域/链类型之间的关系

3.6 其他技术

在区块链技术的发展过程中, 其在广泛应用的同时, 也面临一系列亟需解决的关键挑战. 首先, 隐私保护不足已成为受到广泛关注的问题. 区块链网络普遍强调数据的公开与透明性, 尽管已有部分系统引入了加密机制, 但在特定复杂场景下, 仍可能存在用户隐私被恶意分析与推断的风险. 例如, 在涉及商业敏感信息或个人隐私数据的场景中, 即使交易内容经过加密处理, 攻击者仍然可通过公开交易记录与先进的数据分析技术, 对敏感信息进行重构与追踪[135]. 这不仅威胁用户的数据安全, 也限制了区块链技术在金融、政务等隐私要求较高领域的落地与推广[136]. 其次, 可扩展性不足是当前制约区块链系统性能的重要瓶颈. 随着区块链应用场景的不断扩展, 系统所需处理的交易规模呈指数级增长. 然而, 现有区块链在吞吐量和延迟控制方面难以满足高并发场景的实际需求. 例如, 在金融领域, 实时处理大规模交易是系统运行的基本要求, 但现有技术的不足可能会导致出现交易拥堵、延迟等问题, 影响金融市场的效率和稳定性. 在物联网场景中, 海量设备生成的数据需要被快速写入链上, 但现有区块链技术难以应对如此庞大的数据量, 限制了区块链技术在物联网领域的广泛应用. 此外互操作性差也是当前阻碍区块链技术融合应用的重要因素. 多数区块链系统间难以实现信息与价值的直接交互, 造成技术生态割裂, 形成信息孤岛. 在涉及多方协作的业务场景中, 如供应链管理、跨境贸易等, 不同机构可能部署各自的区块链平台, 若平台间无法互联互通, 将导致信息流动不畅, 难以实现协同管理与数据追溯, 进而降低系统整体效能.

针对上述挑战, 国产区块链软件在实践中探索了多种解决方案, 主要包括隐私保护技术、可扩展性技术与跨链技术. 相关技术的应用情况如图 16 所示. 这些技术的不断演进为区块链系统的安全性、性能与兼容性提升提供了有力支撑, 正逐步推动区块链向更广泛、更深入的行业应用扩展.



(a) 隐私保护技术 (b) 可扩展性技术 (c) 跨链技术

图 16 不同区块链软件其他技术使用情况

3.6.1 隐私保护技术

在隐私保护方面, 为应对区块链中数据隐私与身份泄露风险, 部分区块链软件引入了包括如零知识证明 (zero knowledge proof, ZKP)[111]、安全多方计算 (secure multiparty computation, SMPC)[137]、同态加密 (homomorphic

encryption, HE)[138]、环签名 (ring signature)[139]以及可信执行环境 (trusted execution environment, TEE)[140]在内的多种主流隐私保护技术. 表 10 中对上述隐私保护技术原理与功能做了简要概述.

表 10 隐私保护技术描述

技术名称	描述
零知识证明	允许一方在不泄露具体信息的情况下, 向另一方证明其某个声明是正确的
安全多方计算	使多个参与方在不暴露各自私有数据的前提下, 共同计算一个函数的结果
同态加密	允许在加密数据上直接进行计算, 计算结果仍然是加密的, 解密后可得正确结果
环签名	允许签名者在一组用户中生成一个签名, 使得无法确定具体的签名者, 从而保护其身份隐私
可信执行环境	提供一个隔离的运行环境, 保证在此环境中执行的程序及其数据不会被外部访问或篡改

现象 23: 根据图 16(a) 的统计结果分析, 目前国产区块链软件中, 支持零知识证明和同态加密的区块链软件数量最多 (17 个), 支持可信执行环境和安全多方计算的区块链软件相对较少 (10 个), 支持环签名技术的区块链软件最少 (4 个). 从技术特性来看, 零知识证明能在不暴露具体内容的前提下验证信息的真实性, 适用于需要数据机密性验证但希望不暴露内容的应用场景. 同态加密允许在密文上直接进行计算, 保障数据处理过程中的安全性和隐私性, 在对安全性要求极高的场景中具有显著优势. 这两种技术因通用性强, 开发资料丰富, 逐步成为国产区块链软件在隐私保护层的主流选择. 相较而言, 安全多方计算和可信执行环境更适用于特定业务场景. 安全多方计算适用于多方协作计算, 能在参与方不暴露原始数据的前提下共同完成统计与建模任务. 可信执行环境为敏感计算提供可信计算环境, 保障代码执行过程中的机密性和完整性, 适用于对执行环境安全性有严格要求的场景. 由于部署复杂性和运行环境依赖性较高, 两者尚未形成大规模普及. 至于环签名, 其在实现用户匿名性和不可追踪性方面具有独特优势, 然而由于算法复杂度高、性能开销大, 以及相关开发文档和社区资源相对匮乏, 导致当前支持该技术的国产区块链软件较少. 多数开发团队更倾向于使用成熟度高、社区活跃度高的隐私保护方案.

3.6.2 可扩展性技术

为解决区块链在处理大规模交易和数据时面临的性能瓶颈问题, 近年来, 研究者和开发者提出了多种可扩展性技术, 包括分片 (sharding)[141,142]、多链 (multi-chain)[143]、子链 (subchain)[144]、Rollup、状态通道 (state channel)[145]以及侧链 (sidechain)[146]. 这些技术在结构设计与应用逻辑方面各具特点, 表 11 中对上述 6 类技术的基本原理与主要优势进行了系统梳理.

表 11 可扩展性技术描述

技术名称	描述
分片	区块链网络划分为多个小部分 (分片), 每个分片处理特定的交易, 从而提高整体处理能力
多链	通过同时运行多个独立的区块链, 各自处理特定任务, 以提高网络的效率和可扩展性
子链	在主链之下建立的独立链, 用于处理特定应用或交易, 以减轻主链的负担
Rollup	将大量交易批量处理并压缩成单个交易, 以降低主链上的数据存储需求并提高吞吐量
状态通道	允许用户在链下进行多次交易, 只有最终结果在链上结算, 从而减少链上交易数量
侧链	独立于主链的链, 通过双向锚定技术与主链互通, 以实现不同链间的资产转移和交互

现象 24: 通过对国产区块链软件的调研统计发现, 仅有 20 个国产区块链软件在官网或技术文档中明确声明了采用了某类可扩展性技术. 其中, 支持多链技术的区块链软件最多, 共 11 个. 支持分片技术的有 8 个, 支持侧链技术仅有 6 个, 而子链和 Rollup 技术的使用量相对较少, 分别为 3 个和 2 个, 而状态通道技术的使用量最少, 仅为 1.

从具体技术使用情况来看, 多链技术的广泛应用主要得益于其良好的结构分离能力和业务解耦特性. 通过在同一生态系统中部署多条并行链, 不同链可分别承担特定的业务功能或服务模块, 提升系统并发能力和处理效率. 该模式不仅增强了系统的灵活性, 还便于模块化维护与扩展, 因而在企业级区块链部署中尤为常见. 分片技术则通过将区块链网络划分为多个可并行处理的分片, 每个分片维护独立的状态和交易记录, 从而提升整体的吞吐能力. 尽管分片技术在理论上具有较高的扩展潜力, 但在实际应用中, 其跨分片通信机制、安全性保障和系统一致性维

护面临重大挑战, 对系统设计与实现提出了更高要求, 进而限制了其推广规模.

侧链技术允许开发者在主链之外构建独立运行的区块链系统, 并通过跨链机制与主链进行资产或信息交互. 侧链具备灵活、定制化的特性, 能够降低主链负载、提升隐私保护能力及交易效率. 然而, 侧链生态的碎片化风险较高, 用户在使用不同侧链时面临选择成本增加、资产流动性降低等问题. 此外, 跨链通信的安全性及共识机制设计仍是技术难点, 也影响了其在国产区块链系统中的普及程度. 子链技术与某些类型的 Rollup 机制存在相似性, 均依赖于主链作为最终状态确认或数据可用性的保障层. 这一依赖关系提升了系统的一致性和安全性, 但也在一定程度上降低了链的自治性与灵活性. Rollup 通过将大量链下交易打包后再提交至主链, 显著提升了交易处理效率并降低交易费用, 但当前相关实现仍存在安全验证机制复杂、抗审查能力不足等问题. 子链则以结构简单、交易成本低为特点, 在资源受限场景中具备一定优势.

相较而言, 状态通道技术适用于高频率、点对点的交互场景, 可将大量交互迁移至链下执行, 仅在开闭通道时进行链上记录, 从而极大减轻主链负载. 然而, 该技术在参与方规模、交易类型及资金流转方式等方面存在较强限制, 难以适应用户规模庞大、交互频繁的复杂应用环境, 导致其在国产区块链系统中应用比例极低.

3.6.3 跨链技术

区块链跨链技术是一种连接不同区块链网络、实现资产和信息交互流通的关键技术. 通过跨链技术, 不同区块链上的数字资产、智能合约和数据等可以安全、高效地转移与共享, 从而打破区块链之间的孤立状态, 促进资源的整合与协同, 发挥各自的优势, 最终形成更强大的区块链生态系统. 根据对 39 个区块链软件的调研, 仅有 28 个软件明确介绍了其采用的跨链技术. 所使用的跨链技术主要包括中继链 (relay chain)[147,148]、公证人机制 (notary mechanism)[149]、跨链通信协议 (cross-chain communication protocol)[150]以及原子交换 (atomic swap)[151]这 4 种跨链技术. 表 12 对这 4 种跨链技术进行了简要说明.

表 12 跨链技术描述

技术名称	描述
中继链	作为多个区块链之间的连接平台, 中继链允许不同链之间进行信息和资产的互操作
公证人机制	通过信任的第三方对交易或信息进行验证和记录, 以确保跨链交易的安全和可靠性
跨链通信协议	一套规则和标准, 允许不同区块链之间交换信息和资产, 实现互操作性
原子交换	一种去中心化的交易方式, 使得不同加密货币之间的交换能够在不信任的环境中安全进行

现象 25: 通过对图 16(c) 的分析, 发现这 28 个区块链软件中, 有多达 25 个使用了跨链通信协议. 相比之下, 采用中继链技术的区块链软件数量较少, 仅为 12 个; 而使用公证人机制和原子交换技术的区块链软件数量更为有限, 分别为 2 个.

首先, 跨链通信协议是一种标准化规则, 旨在促进不同区块链之间的安全信息和资产交换. 该技术使不同区块链网络能够相互通信, 并确保交易的有效性和一致性, 从而解决了资产跨链转移的难题. 跨链通信协议具备较强的灵活性、安全性, 能够有效整合不同链的优势, 优化资源配置和交易成本, 因此被广泛应用. 与此相比, 中继链技术是一种特定的架构, 作为连接不同区块链的中心枢纽, 专门处理并转发来自各个链的信息和交易. 虽然中继链技术在特定生态系统或网络中有效, 但其主要应用于连接特定区块链之间, 局限性较为明显. 公证人机制相对较为中心化, 依赖于可信的第三方公证人进行信息验证. 在追求去中心化理念的区块链平台中, 这种机制的广泛应用受到限制, 影响了其普及度.

原子交换技术虽然能够实现无需第三方信任的跨链资产交换, 但在实际应用中面临技术复杂性、兼容性问题以及对交易双方要求较高的挑战, 导致其实际应用数量较少.

见解 10: 国产区块链软件在解决数据隐私、可扩展性和跨链交互问题上进行了积极的探索. 在隐私保护技术方面, 零知识证明和同态加密技术的应用最广泛. 在可扩展性技术上, 多链技术的使用频率较高, 而其他技术的采用则更多依赖于具体的应用场景. 这表明, 国产区块链软件在技术选型上具备较强的灵活性, 能够根据实际需求和应用背景, 选择适合的技术方案.

4 国内外区块链软件在核心技术上有哪些不同?

本节选择了 6 个具有代表性国外区块链软件进行分析, 结合已有国产区块链软件的发展情况, 总结当前国产区块链软件的优势与不足. 所选的 6 个区块链软件中, 包含 3 个联盟区块链软件和 3 个公有区块链软件. 邵奇峰等人[152]认为, Linux 基金会的 Hyperledger Fabric、R3 联盟的 Corda 和 EEA 的 Quorum 是 3 个具有代表性的联盟区块链软件, 基于此, 我们选择这 3 个作为对比联盟链. 首先, Hyperledger Fabric 由 Linux 基金会主导, 得到了众多企业和机构的支持和参与. 它具有高度的可定制性和灵活性, 适用于各种企业级应用场景, 如供应链金融、贸易金融等. 其次, Corda 专为金融行业设计, 旨在处理金融机构之间的复杂业务流程和交互. 在隐私保护方面, Corda 表现出色, 能够满足金融领域对数据保密性的严格要求. 最后, Quorum 由摩根大通基于以太坊开发, 在保留以太坊智能合约功能的同时, 加强了隐私保护和性能优化, 主要用于金融机构之间的联盟合作.

在选择国外具有代表性的公有区块链软件, 我们参考了 CoinMarketCap 以及 DappRadar[153]的数据, 从区块链软件的市场价值以及 DApp 市场规模的角度, 选择了 Ethereum、Polygon 和 Solana 这 3 个平台. 作为首个引入智能合约概念的区块链软件, Ethereum 的重要性不言而喻, 其区块链软件价值以及 DApp 市场规模均排名第 2. 此外, Ethereum 还拥有强大的开发生态以及广泛的认可度. 在公有链可扩展性问题上, Polygon 通过结合二层网络和侧链技术, 优化了交易速度和成本, 同时兼容 Ethereum, 降低了开发者的迁移成本. 其 DApp 市场规模仅次于 Ethereum. Solana 于 2020 年推出, 凭借其惊人的交易处理速度而著称, 能够支持大量并发交易, 适用于高频交易和大规模应用. 此外, Solana 采用了一系列创新的技术架构和共识机制, 显著提高了区块链性能, 并推动了技术发展. 目前 Solana 的市场价值排名第 5, 是排名前 5 中最“年轻”的区块链软件.

接下来, 我们将从数据层、网络层、共识层、合约层、应用层以及其他技术 6 个方面对国外具有代表性的区块链软件进行分析. 分析过程中, 我们主要关注与国产区块链软件不同的技术, 对相同的技术不再重复描述. 通过这一对比分析, 我们旨在揭示当前国产区块链软件发展存在的不足, 并为其未来的发展提供借鉴.

4.1 数据层

4.1.1 数据结构

在数据结构设计方面, 联盟链平台 Corda 采用了一种基于交易历史的链式结构. Corda 中的交易并非孤立存在, 而是相互关联, 形成一条有序的、可溯源的交易链. 每一笔新的交易均建立在前一笔交易的状态基础之上, 并明确记录对前序状态的更改信息. 例如, 初始交易可确立某资产的基本属性与归属关系, 后续交易则可能调整资产的价值、变更其所有权或添加新的约束条件. 该线性结构不仅保障了数据的连续性与准确性, 也提升了交易流程的可追溯性和透明度, 为金融业务提供了清晰可信的交易轨迹, 是 Corda 保障金融交易合规性与可验证性的基础之一. 值得注意的是, Corda 的交易历史仅在相关参与方之间共享, 避免了传统区块链全网广播带来的数据泄露风险, 有助于保护商业机密与个人隐私; 同时, 该设计也降低了数据冗余, 更符合企业级场景的实际需求. 然而, 该结构的局部共识特性可能削弱系统的全局信任机制与整体安全性, 且其独特的数据结构对系统的开发与维护也提出了更高的要求.

总体来看, 相较于国产区块链软件在性能优化上对传统数据结构进行微调, 部分国外区块链平台如 Corda 则在数据结构层面进行了全新设计, 以适应对隐私保护与可验证性的更高要求. 随着区块链在政务、医疗、金融等敏感场景中的落地, 隐私保护将成为核心诉求. 因此, 如何在隐私保护、全局信任与系统安全性之间实现平衡, 将是国产区块链软件未来在数据结构设计中亟需突破的关键方向. 例如, 探索将零知识证明机制嵌入数据结构体系, 使得数据在隐私保护前提下仍具备可验证性, 是可行的发展路径之一.

4.1.2 数据库

现象 26: 在数据库设计方面, 为应对传统数据库在区块链环境中面临的并发处理能力不足、扩展性有限等挑战. Solana 提出了专为区块链高性能运行场景设计的账户数据库系统 Cloudbreak[154]. 该系统通过多项关键性技术实现了并发友好的数据读写能力, 包括内存映射文件的高效使用、账户状态的分解存储、在多块 SSD 设备上的并行读写机制以及类似垃圾回收的存储管理策略. Cloudbreak 不仅在保持数据一致性与完整性的同时实现了水平

扩展能力, 还显著提升了网络整体的处理性能与可扩展性. 然而, 该架构也存在一定局限性, 例如对数据库通用性的削弱以及需从零构建定制化数据库逻辑的开发成本问题. 相比之下, 当前大多数国产区块链软件尚未构建具有区块链特征的专用数据库系统. 以腾讯链为例, 其使用了腾讯自研的企业级分布式数据库 TDSQL (Tencent distributed SQL). 虽然该系统在高可用性与分布式事务处理方面具备优势, 但其并非为区块链特性而设计, 因此在与区块链系统集成时面临如下 3 方面挑战.

- 数据结构和存储方式差异.
- 共识机制差异.
- 性能和可扩展性的挑战.

数据结构和存储方式差异主要表现在两个方面. (1) 数据模型不匹配, 区块链的数据结构具有独特的链式结构和分布式存储特点, 每个区块包含前一个区块的哈希值, 从而形成一个不可篡改的链式结构, 这种结构确保了数据的安全性和完整性. 然而, TDSQL 作为一种关系型数据库, 其数据存储方式和结构与区块链存在显著差异. 区块链采用的是非传统的分布式链式存储方式, 而 TDSQL 则基于表格的行列结构进行数据存储. 将 TDSQL 与区块链融合时, 必须解决数据模型的转换和适配问题, 以确保能够正确存储和读取区块链中的数据. (2) 数据冗余问题, 在区块链系统中, 为了保证数据的安全性和可靠性, 通常每个节点都会要保存完整的账本数据, 这导致了大量的数据冗余. 与此相比, TDSQL 的设计侧重于高效的存储和查询, 因此对于区块链中这种冗余的数据存储模式可能不太适应, 需额外进行优化处理, 以避免存储资源的浪费和性能下降

共识机制差异主要表现为共识算法的兼容性和共识效率的影响. (1) 对于共识算法的兼容性, 区块链的核心是共识机制, 它用于保证分布式系统中各节点的数据一致性. 常见的区块链共识算法有工作量证明、权益证明、实用拜占庭容错等. 然而, TDSQL 作为传统的数据库系统, 通常采用集中式的事务处理和一致性协议. 与区块链融合时, 需要将 TDSQL 的事务处理机制与区块链的共识算法进行兼容和整合, 以确保数据的一致性和可靠性. 这种兼容可能需要对 TDSQL 的事务处理逻辑进行修改和扩展, 从而增加了系统的复杂性和开发难度. (2) 区块链的共识过程需要消耗大量的计算资源和时间, 尤其是在大规模的分布式网络中, 共识效率可能会成为系统的瓶颈. 尽管 TDSQL 在高并发的事务处理方面具有一定的优势, 但在与区块链融合后, 可能会受到区块链共识机制的影响, 导致事务处理的延迟增加, 影响系统的性能和响应速度.

性能和可扩展性的挑战也可能会导致性能瓶颈以及可扩展性差的问题. 尽管 TDSQL 在处理大规模数据和高并发事务时表现出色, 但当与区块链融合后, 可能会受到区块链性能的影响, 导致系统的整体性能下降. 因此, 必须对 TDSQL 和区块链的性能进行优化和调整, 以提升系统的处理能力和响应速度. 随着区块链应用的不断扩展, 系统的规模和数据量将逐步增加, 因此系统必须具备良好的可扩展性. TDSQL 作为一种分布式数据库, 具有一定的可扩展性, 但在与区块链融合后, 可能会面临新的可扩展性挑战. 例如, 区块链的节点数量增加可能会导致 TDSQL 的负载增加, 需要对 TDSQL 的分布式架构进行优化和扩展, 以满足系统的可扩展性需求.

见解 11: 目前国产区块链软件缺乏一个具有普适性、原生支持区块链特性的通用数据库系统, 这一缺口制约了系统在高性能、高可靠和复杂共识环境下的性能表现. 未来亟需面向区块链需求, 研发具有链式结构支持、原生共识集成和多副本冗余优化能力的新型国产区块链数据库系统, 以解决“结构差异、机制差异与性能瓶颈”这 3 大挑战.

4.2 网络层

国外区块链软件大多在网络层的数据传播机制方面进行了改进或优化. 较为经典的例子是 Hyperledger Fabric 针对数据隐私保护需求提出的通道机制, 该机制为特定成员提供了一个私有的账本共享以及交易环境. 通过配置文件, 用户可以定义成员资格、访问权限等相关规则, 确保只有授权的成员能够参与特定通道的交易和数据共享. 该机制对于区块链在商业领域的应用至关重要, 且许多国产联盟链在设计传播机制时都参考了该方案. Corda 的数据传播机制与之相似, 但它没有将参与方集中在一个通道中, 而是维护一个交易参与方信息表. 具体的交易信息经过加密后在交易参与方之间进行传播, 这种设计不仅保护了交易隐私, 还减少了网络中的数据流量和

处理负担. 除了保护数据传播过程中的隐私, 数据高效传播也是网络层传播机制较为关注的重点之一. Solana 提出了一个名为 Turbine 的数据传播机制, 将数据分割后以并行数据流的方式进行传输, 极大地提高了数据传播的速度和效率. Turbine[155]机制的优势在于其高吞吐量和低延迟, 使得 Solana 能够处理大量的交易和数据, 从而为用户提供快速、流畅的区块链体验.

相比之下, 国产区块链软件对于传播机制的关注度较低, 大部分系统直接使用现有的网络层技术. 然而, 区块链数据隐私保护、交易处理能力在一定程度上依赖于高效的传播机制. 因此, 国产区块链软件在未来的发展中, 可能需要根据具体应用场景, 优化已有的传播机制, 或创新性地提出新的传播机制以解决隐私保护或性能瓶颈问题.

4.3 共识层

表 13 展示了 6 种具有代表性的国外区块链软件所支持的共识协议. 在这些区块链软件中, 所采用的共识协议类型和设计各有不同, 反映了不同区块链网络在性能、安全性以及去中心化等方面的具体需求. 具体而言, 联盟链类型的软件通常倾向于采用 BFT 类的共识协议, 而公有链类则广泛使用 PoS 类共识协议. 这一差异主要源于联盟链与公有链在应用场景、参与者信任模型及对系统特性要求上的不同. 联盟链通常面向预定的可信参与者群体, 因此其共识协议更侧重于提供较高的容错能力和处理效率; 而公有链则更强调去中心化和开放性, 因而选择能够保障网络安全性的 PoS 类协议.

表 13　6 种国外区块链软件共识协议支持情况

区块链软件名称	描述
Hyperledger Fabric	CFT, BFT, Kafka, Raft, Solo
Corda	Raft, BFT
Quorum	Raft, IBFT, PoA, QBFT
Ethereum	PoW, PoS
Polygon	Polygon PoS
Solana	PoS-based Tower BFT, PoH

具体而言, Quorum 提出了 IBFT (Istanbul Byzantine fault tolerance)[156]以及 QBFT (Quorum Byzantine fault tolerance)[157]两种共识算法. IBFT 是一种基于拜占庭容错的共识算法, 能够在相对较短的时间内达成共识, 提升区块链的处理速度, 并具备一定的容错能力. 而 QBFT 是 Quorum 在 IBFT 基础上的改进, 进一步优化了性能, 并增强了安全性, 为应对恶意攻击和错误情况提供了更强的保障. Polygon 则采用 PoS 共识协议来确保网络的安全性, 验证者通过质押 MATIC 代币[158]来获得区块验证权, 并且可以通过委托其代币给验证者来获得奖励. 与传统 PoS 共识协议相比, Polygon 在多链、质押与委托机制以及检查点机制等方面进行了创新改进. Solana 提出了 PoS-based Tower BFT[159], 它结合了传统 BFT 算法和 PoS 特性, 允许网络节点快速达成共识, 显著提高交易的吞吐量. 在区块链网络中, 低延迟能够加快交易确认速度, 从而改善用户体验和支持实时应用. PoH (proof of history)[160]是 Solana 另一项核心创新. 它创建了一种历史记录, 使得网络可以在不依赖全局时间的情况下验证事件发生的顺序. 这一创新有效减少了共识协议中节点时间同步的需求, 从而显著降低了延迟.

进一步结合国内外环境, 可以发现, 国内外区块链软件在共识协议的创新性、选择以及性能与安全性权衡等方面存在显著差异. 在创新性方面, 国外区块链软件在共识协议的创新上较为活跃, 频繁提出新的共识算法或改进现有协议. 例如, 权益证明 (PoS)、委托权益证明 (DPoS) 等共识机制, 均是国外区块链社区率先提出并付诸实践的. 而国内在共识协议的创新上也在不断努力, 但相较而言, 更多侧重于基于现有共识协议的优化和改进, 以适应特定行业和应用场景的需求. 例如, 中国人民银行数字货币研究所联合清华大学推出的 Dashing 协议[161], 在传统联盟链共识协议基础上进行创新, 满足了金融科技场景下高安全、高延展、高吞吐和低延迟同时满足的难题. 在共识协议选择方面, 不同国家的监管政策对区块链软件的选择具有较大影响. 某些国家对区块链的监管相对宽松, 这使得区块链软件在共识协议的选择上更为自由多样. 特别是一些小型的区块链项目或实验性项目可以尝试使用新颖但可能风险较高的共识协议. 相反, 国内对区块链的监管较为严格, 尤其是在金融领域, 要求区块链软件的共识协议必须符合监管要求, 具有较高的安全性和可控性. 因此, 国内的区块链项目在选择共识协议时会更为谨慎,

更倾向于使用经过验证的、成熟的共识协议, 并根据监管政策的变化进行相应的调整与优化.

在性能和安全性权衡方面, 国外区块链软件在设计共识协议时, 有时更侧重于追求高性能和可扩展性, 可能会在一定程度上牺牲部分安全性. 例如, 一些采用 DPoS 共识机制的区块链项目为了提高交易处理速度, 减少了节点的数量, 这虽然提升了性能, 但可能会导致一定的中心化风险. 与此不同, 国内区块链应用更加注重安全性和稳定性, 在共识协议的选择上会更加谨慎地权衡性能和安全性. 例如, 在金融领域的区块链应用中, 通常会优先选择安全性高、容错性强的共识协议, 尽管这可能会导致性能上的一些损失, 但能够保证交易的安全性和可靠性.

4.4 合约层

我们对国外 6 个区块链软件在合约层中使用的技术进行了统计, 结果如表 14 所示.

表 14 6 种国外区块链软件合约层技术使用情况

区块链软件名称	合约编程语言	合约执行引擎
Hyperledger Fabric	Java, Go, NodeJS	JVM, Docker
Corda	Kotlin, Java	JVM
Quorum	Solidity, Vyper	EVM
Ethereum	Solidity, Vyper, Yul, Bamboo, LLL	EVM, eWASM, WASM
Polygon	Solidity, Vyper	EVM
Solana	Rust, C	SeaLevel

4.4.1 合约编程语言

从表 14 中可以看到, 这 6 个国外区块链软件均支持两种以上的合约编程语言, 其中 Solidity, Vyper, Yul[162], LLL (low-level lisp-like language)[163], Bamboo[164]是专门为智能合约设计的编程语言. 前 4 种语言是为以太坊合约开发设计的, 其中 Solidity 是目前使用最为广泛的智能合约语言, 专为以太坊合约开发设计. Vyper 强调安全性和可读性. Yul 则是以太坊虚拟机的中间语言, 用于优化智能合约性能. LLL 是用于以太坊的低级语言, 适用于对性能要求较高的合约. Bamboo 则是一种实验性语言, 重点关注安全性和易用性. 传统编程语言如 Go、Java 虽然被用于智能合约开发, 但其在安全性、性能和合约升级方面仍面临挑战.

现象 27: 相较而言, 国产区块链软件缺乏专门为智能合约编程语言, 这使得过度依赖国外技术可能会导致国产区块链软件在市场竞争中的劣势. 这种依赖性可能导致以下问题: 无法完全满足特定行业需求、开发效率低下以及潜在的安全性风险等问题.

4.4.2 合约执行引擎

从表 7 中可以看到, 大部分公有链都使用了专为智能合约设计的编程语言, 而联盟链仍然主要使用传统编程语言. 这主要是因为传统编程语言具有成熟的开发者基础、较好的性能表现以及成熟的生态系统. 虽然传统编程语言在开发智能合约时仍然面临很多挑战, 但国外的联盟链使用了不同的合约执行引擎以应对. 如 Hyperledger Fabric 使用 Docker 容器来运行以 Go 语言开发的智能合约 (在 Hyperledger Fabric 中称为链码). 这种设计允许链码在隔离的环境中运行, 从而增强了安全性和可管理性. 对国外区块链软件使用的合约执行引擎进行统计后, 发现仅有 Hyperledger Fabric 和 Ethereum 支持多种合约执行引擎, 而其他软件均支持一种合约执行引擎且大部分为 EVM. 其中 Ethereum 的 eWASM[165]和 Solana 的 SeaLevel[166]是较为特殊的案例. Solana 支持使用 Rust 进行合约开发, 该种编程语言能够提供内存安全和并发安全. 而 eWASM 是以太坊未来的执行引擎, 旨在替代 EVM, 通过支持 WebAssembly (WASM) 提高性能和灵活性. 为了更好地兼容 Solana 系统, Solana 设计了一种高性能的并行智能合约执行环境—SeaLevel, 其中由 Rust 编写的合约会被编译成 BPF (Berkeley packet filter)[167]字节码. 这种字节码能够在 SeaLevel 上高效运行. SeaLevel 具有并行执行、高吞吐量、动态费用以及可扩展等特点, 使得 Solana 能够突破传统区块链智能合约执行的性能限制, 为去中心化应用提供更高效、可扩展的运行环境.

现象 28: 相较而言, 国产区块链软件在合约执行引擎方面的创新并不逊色, 通常会针对特定行业应用进行更加深入的性能优化, 以支持高并发和大规模交易. 然而, 国外区块链软件的合约执行引擎面向更广泛的应用场景且支持更多开发者参与引擎开发和改进. 相比之下, 国产区块链软件更多依赖于特定公司的技术栈, 这在一定程度上

反映了国内区块链技术在生态系统上的局限性.

见解 12: 国产区块链软件在合约执行引擎方面的创新并不落后, 并且更加注重特定行业应用进行的深入性能优化. 然而, 国产区块链软件对特定公司的技术栈依赖, 反映了国内区块链技术在生态系统上的局限性.

4.5 应用层

我们对 6 个国外区块链软件支持的应用进行了统计, 结果如表 15 所示.

表 15 国外区块链软件应用

区块链软件	金融服务	智慧政务、城市	溯源存证与供应链管理	BaaS	NFT	DApp
Hyperledger Fabric	√	√	√	√	—	—
Corda	√	√	√	—	√	—
Quorum	√	√	√	√	√	—
Ethereum	√	√	√	√	√	√
Polygon	√	√	—	—	√	√
Solana	√	—	—	—	√	√
合计	6	5	4	3	5	3

从表 15 中可以看到, 国外区块链软件在金融服务、智慧政务以及 NFT 方面的应用较为广泛. 然而, 仅仅针对前文提到的 6 个国外区块链软件进行分析, 无法全面反映其应用现状. 因此我们进一步分析了国外大多数区块链软件的应用案例, 发现其主要应用场景集中在去中心化金融、NFT 以及跨境支付等领域. 这些应用场景虽然也出现在国产区块链软件中, 但是由于文化和监管环境差异, 国产区块链软件在这些领域的尝试较少. 总体而言, 其国产区块链软件的应用场景更多集中在企业级应用和平台生态构建上. 具体而言, 企业级应用是指基于区块链技术为政府和企业提供数据共享平台建设、数据管理等服务, 旨在通过区块链实现更加高效、透明和安全的企业管理. 平台建设则是通过整合区块链技术与现有云服务、金融平台等, 形成一体化的解决方案, 为各类应用提供支持. 与之不同, 国外区块链软件则更倾向于通过代币激励机制鼓励用户参与, 并探索新型的经济模式, 如 DeFi 和 NFT 市场等. 这些新兴经济模式具有较高的风险, 但也伴随有潜在的高回报.

见解 13: 监管环境、政策以及市场需求等因素导致当前国内外区块链软件在应用层方面出现较大差异, 国外区块链软件更倾向于探索新兴经济模式, 在承担高风险的同时追求高回报. 而国产区块链软件则更多的是利用区块链技术提供稳定的数据共享、管理等服务, 重点关注企业级应用和平台生态的构建.

4.6 其他技术

在其他技术发展方面, 国外区块链软件发展迅速, 然而, 以上 6 个具有代表性的区块链软件使用的其他技术覆盖面并不完全. 为此, 我们对国外区块链软件在隐私保护、可扩展性以及跨链 3 种技术进行了全面的调研, 具体信息如表 16 所示.

表 16 国外区块链软件其他技术使用情况

技术类别	具体技术
隐私保护技术	零知识证明, 安全多方计算, 同态加密, 环签名, 可信执行环境, 混淆, 可验证计算
可扩展性技术	分片, 多链, 子链, Rollup, 状态通道, 侧链, 状态压缩
跨链技术	中继链, 公证人机制, 跨链通信协议, 原子交换

在隐私保护技术的使用方面, 相较于国产区块链软件, 国外区块链软件使用到了混淆与可验证计算两种隐私保护技术. 例如 Monero[168]、Zcash[169]以及 Dash[170]等区块链软件使用了混淆技术有效地保护用户的身份隐私或交易隐私. 而 Oasis[171]则使用了可验证计算来进一步保护交易隐私. 鉴于技术成熟度低和合规需求等因素, 国产区块链较少使用这些技术. 在可扩展性技术方面, 国外区块链软件使用了状态压缩这一技术. 例如 Zilliqa[172]通过分片技术和状态压缩, 减少节点存储需求, 进而提升了网络的可扩展性. Elrond[173]采用状态压缩和分片技术, 既保障了安全性, 又提高了交易速度, 并有效降低了交易费用. 相比之下, 国产区块链软件较少使用这一技术, 表明在可扩展

性方面的技术创新相对滞后.

5 总 结

本文对国产区块链软件的发展趋势和核心技术进行了统计与分析. 首先, 收集并筛选得到了 103 个国产区块链软件, 通过质量评估得到了 39 个质量较高的区块链软件. 为了全面了解国产区块链软件的发展趋势, 我们从软件发展历程、软件分布以及软件关系这 3 个角度进行了统计与分析. 接下来, 从区块链技术的 6 个层次, 深入分析了 39 个高质量国产区块链软件的核心技术, 涵盖了数据层、网络层、共识层、合约层、应用层等关键技术领域. 通过对这些技术的详细统计与分析, 进一步了解了国产区块链在性能、安全性、隐私保护以及可扩展性等方面的优势与挑战. 此外, 我们挑选了 6 个具有代表性的国外区块链软件, 针对其所采用的先进技术进行了对比分析, 揭示了国内外区块链软件在技术创新、共识协议、智能合约执行引擎以及跨链技术等方面的差距. 在统计与分析过程中, 我们共发现了 28 个有趣的现象和 13 个见解. 这些现象和见解为国产区块链软件的发展提供了宝贵的参考和洞察. 尤其是在技术创新、应用扩展和行业合作方面, 观察到国产区块链软件在多个层面与国际先进水平仍有一定的差距, 但也展现出较大的潜力与提升空间. 综上所述, 本文不仅提供了对国产区块链软件发展现状的全面分析, 还为未来的研究方向提供了指导. 未来的研究可以从以下几个方面进一步展开: 一是加大对区块链核心技术的创新和突破, 尤其是在智能合约优化、跨链技术、隐私保护技术等领域, 推动国产区块链软件向更高的技术水平迈进; 二是探索更多适合我国市场和文化的区块链应用场景, 尤其是在金融、政务、供应链等领域, 结合行业需求进行定制化开发; 三是关注区块链的监管和合规问题, 确保技术发展与政策法规的协调发展, 促进区块链在国内外市场中的应用推广.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. https://bitcoin.org/bitcoin.pdf

[2] Treleaven P, Brown RG, Yang D. Blockchain technology in finance. Computer, 2017, 50(9): 14–17. [doi: 10.1109/MC.2017.3571047]

[3] Queiroz MM, Telles R, Bonilla SH. Blockchain and supply chain management integration: A systematic review of the literature. Supply Chain Management, 2020, 25(2): 241–254. [doi: 10.1108/SCM-03-2018-0143]

[4] Reyna A, Martín C, Chen J, Soler E, Díaz M. On blockchain and its integration with IoT. Challenges and opportunities. Future Generation Computer Systems, 2018, 88: 173–190. [doi: 10.1016/j.future.2018.05.046]

[5] Cao YL, Li JJ, Chao KL, Xiao JM, Lei G. Blockchain meets generative behavior steganography: A novel covert communication framework for secure IoT edge computing. Chinese Journal of Electronics, 2024, 33(4): 886–898. [doi: 10.23919/cje.2023.00.382]

[6] Liu Y, Lyu C, Bai F, Parishwad O, Li Y. The role of intelligent technology in the development of urban air mobility systems: A technical perspective. Fundamental Research, 2024, 4(5): 1017–1024. [doi: 10.1016/j.fmre.2023.08.006]

[7] Meng XF, Li Y, Liu K, Liu Y, Yang B, Song X, Liao GQ, Wang SZ, Yu ZQ, Chen LB, Pan X, Lin YM. Spatial data intelligence and city metaverse: A review. Fundamental Research, 2025, 5(3): 1169–1193. [doi: 10.1016/j.fmre.2023.10.014]

[8] Buterin V. Ethereum white paper. GitHub Repository, 2013, 1: 22–23.

[9] Buterin V. A next-generation smart contract and decentralized application platform. White Paper. 2014. https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf

[10] Wang Q, Li RJ, Wang Q, Chen SP. Non-fungible token (NFT): Overview, evaluation, opportunities and challenges. arXiv:2105.07447, 2021.

[11] Zetzsche DA, Arner DW, Buckley RP. Decentralized finance. Journal of Financial Regulation, 2020, 6(2): 172–203. [doi: 10.1093/jfr/fjaa010]

[12] BSN. Blockchain-based service network user manual. 2024 (in Chinese). https://zhuanwang.bsnbase.com/static/tmpFile/%E5%8C%BA%E5%9D%97%E9%93%BE%E6%9C%8D%E5%8A%A1%E7%BD%91%E7%BB%9C%E7%94%A8%E6%88%B7%E6%89%8B%E5%86%8C.pdf

[13] e-CNY. 2024. https://en.wikipedia.org/wiki/Digital_renminbi

[14] Singh J, Michels JD. Blockchain as a service (BaaS): Providers and trust. In: Proc. of the 2018 IEEE European Symp. on Security and Privacy Workshops (EuroS&PW). IEEE, 2018. 67–74.

[15] Etherscan. https://etherscan.io/

[16] BSCscan. https://bscscan.com/
[17] Chainalysis. Blockchain intelligence for investigations, risk, and security. 2024. https://www.chainalysis.com/
[18] Blockchain explorer. https://www.blockchain.com/explorer
[19] Chainlist. https://chainlist.org/
[20] Alchemy. https://www.alchemy.com/
[21] CoinMarketCap. https://coinmarketcap.com/
[22] CoinGecKo. https://www.coingecko.com/zh
[23] Hildenbrandt E, Saxena M, Rodrigues N, Zhu XR, Daian P, Guth D, Moore B, Park D, Zhang Y, Stefanescu A, Rosu G. KEVM: A complete formal semantics of the Ethereum virtual machine. In: Proc. of the 31st IEEE Computer Security Foundations Symp. (CSF). Oxford: IEEE, 2018. 204–217. [doi: 10.1109/CSF.2018.00022]
[24] GitHub. https://docs.github.com/en
[25] Gitee. https://gitee.com/
[26] Vechain. https://www.nanojclean.com/home
[27] China Unicom Blockchain. Application scenarios. 2021 (in Chinese). https://support.cucloud.cn/document/127/652/674.html?id=674&arcid=945
[28] HTX. https://www.htx.com/zh-cn/opend/newApiPages/
[29] Binance smart chain documentation. 2024. https://docs.bnbchain.org/bnb-smart-chain/
[30] Tron: Advanced decentralized blockchain platform. 2024. https://tron.network/static/doc/white_paper_v_2_1.pdf
[31] Nervos CKB docs. 2024. https://docs.nervos.org/docs/getting-started/how-ckb-works
[32] IRISnet documents. 2024. https://www.irisnet.org/docs/
[33] PlatON overview. 2024. https://devdocs.platon.network/docs/en/
[34] inchain. 2024. https://github.com/inchaincodes/inchain
[35] Zhi Xin Chain. Product introduction. 2024 (in Chinese). https://console.zxchain.qq.com/docs?docpath=doc/content/%E5%BF%AB%E9%80%9F%E5%85%A5%E9%97%A8/%E5%87%86%E5%A4%87%E5%B7%A5%E4%BD%9C.html
[36] ChainMaker. Introduction to the underlying technology platform of Changan Chain. 2024 (in Chinese). https://docs.chainmaker.org.cn/v2.3.7/html/quickstart/%E9%95%BF%E5%AE%89%E9%93%BE%E5%9F%BA%E7%A1%80%E7%9F%A5%E8%AF%86%E4%BB%8B%E7%BB%8D.html
[37] Yuan Y, Wang FY. Blockchain: The state of the art and future trends. Acta Automatica Sinica, 2016, 42(4): 481–494 (in Chinese with English abstract). [doi: 10.16383/j.aas.2016.c160158]
[38] Hyperledger Fabric. Hyperledger Fabric docs. 2024. https://hyperledger-fabric.readthedocs.io/en/latest/index.html
[39] Androulaki E, Barger A, Bortnikov V, *et al*. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In: Proc. of the 13th EuroSys Conf. Porto: ACM, 2018. 30. [doi: 10.1145/3190508.3190538]
[40] CordaR3. Corda open source edition key concepts. 2024. https://docs.r3.com/en/platform/corda/4.12/community/about-corda/corda-key-concepts.html
[41] Brown RG, Carlyle J, Grigg I, Hearn M. Corda: An introduction. R3 CEV. 2016. https://docs.r3.com/en/pdf/corda-introductory-whitepaper.pdf
[42] Quorum. Enterprise Ethereum client. 2024. https://goquorum.readthedocs.io/
[43] Baliga A, Subhod I, Kamat P, Chatterjee S. Performance evaluation of the Quorum blockchain platform. arXiv:1809.03421, 2018.
[44] Wood G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014, 151(2014): 1–32.
[45] Polygon. POL: One token for all Polygon chains. 2024. https://polygon.technology/papers/pol-whitepaper
[46] Yakovenko A. Solana: A new architecture for a high performance blockchain v0. 8.13. 2018. http://solana.io/solana-whitepaper.pdf
[47] Yakovenko A. Solana: A new architecture for a high performance blockchain v0.8.13. White Paper, 2018. https://whitepaper.io/coin/solana
[48] The “Software and Information Technology Services Industry Development Plan (2016–2020)” issued. Dual Use Technologies and Products, 2017, (3): 5 (in Chinese). [doi: 10.19385/j.cnki.1009-8119.2017.03.005]
[49] The 2018 key points of standardization work in information and software service industry focuses on six key areas of standards development. Information Technology and Standardization, 2018, (4): 4 (in Chinese).
[50] An Shun Chain docs. 2023 (in Chinese). https://docs.anshun.tech/kai-fa-zhi-nan/kuai-su-ru-men
[51] Tai An Chain. Develop and use smart contracts. 2024 (in Chinese). http://taianexplorer.bsnbase.com/
[52] Annchain docs. 2024. https://github.com/annchain/document

[53] FISCO BCOS. FISCO BCOS 2.0 technical document. 2024 (in Chinese). https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/
[54] Xinghuo Chain. Xinghuo Chain development guide. 2024 (in Chinese). https://bif-doc.readthedocs.io/zh-cn/3.0.0/
[55] Shu Xin Chain. Shu Xin Chain technical document. 2024 (in Chinese). https://www.btsi.org.cn/resourceCenter
[56] Gui Chain. Blockchain development services. 2024 (in Chinese). 2024. https://www.gxbaas.cn/develop-docs/1733190606800301.html
[57] AntChain. Overview of the contract platform. 2019 (in Chinese). https://antdigital.com/docs/11/143759?Source=bs_baidu_my_55480&bd_vid=8090731222145646492
[58] NEO. NEO documentation. 2024. https://docs.neo.org/docs/index.html
[59] Venachain. Venachain introduction. 2024 (in Chinese). https://www.wxblockchain.com/#/home
[60] BubiChain. BubiChain introduction. 2024 (in Chinese). https://docs.bubi.cn/docs/intro/index/
[61] Hyperchain. Hyperchain blockchain browser. 2024 (in Chinese). https://hyperchain.readthedocs.io/zh-cn/tmp-2.12.0_a/qulian%20blockchain%20browser.html
[62] TiChain. TiChain product documents. 2024 (in Chinese). https://thgy.yuque.com/ldgxec/tichain/vnvcgp?#
[63] Thunder Chain. Contract open platform development guide. 2024 (in Chinese). https://blockchain.xunlei.com/site/docopen.html
[64] ONTology. ONTology developer center. 2024. https://docs.ont.io/
[65] Magcloud Digital. White paper on Magcloud blockchain M0 technology. 2020 (in Chinese). https://cyjf.oss-cn-beijing.aliyuncs.com/dea737b7578b48b999868c5a44917a6b.pdf
[66] XuperChain. XuperChain official documentation 5.3. 2024 (in Chinese). https://xuper.baidu.com/n/xuperdoc/v5.3/quickstart/quickstart.html
[67] RepChain. RepChain-doc. 2024 (in Chinese). https://zls201624.github.io/repchain-doc/RepChain-2.0.html
[68] Chain33. Platform introduction. 2024 (in Chinese). https://chain.33.cn/document/60#
[69] WuTong Chain. Introduction document. 2024 (in Chinese). https://github.com/tjfoc/wutongchain/blob/master/sample/chaincode/doc/README.MD
[70] Conflux Chain. 2025 (in Chinese). https://www.tree-graph.org.cn/
[71] Huawei Chain. Guide to blockchain service development. 2025 (in Chinese). https://support.huaweicloud.com/devg-bcs/bcs-devg-pdf.pdf
[72] IRITA. IRITA document. 2024 (in Chinese). https://irita.bianjie.ai/docs/
[73] Wenchang Chain. Help document. 2024 (in Chinese). https://docs.avata.bianjie.ai/
[74] Wuhan Chain. 2024 (in Chinese). https://github.com/BSN-DDC/wuhanchain?tab=readme-ov-file
[75] Straitchain. Straitchain technical white paper-All. 2022 (in Chinese). https://baijiahao.baidu.com/s?id=1732312699883732607&wfr=spider&for=pc
[76] Dong SY, Kryczka A, Jin YQ, Stumm M. RocksDB: Evolution of development priorities in a key-value store serving large-scale applications. ACM Trans. on Storage (TOS), 2021, 17(4): 26. [doi: 10.1145/3483840]
[77] Nebulas. Document. 2019 (in Chinese). https://wiki.nebulas.io/zh-cn/latest/go-nebulas/README.html
[78] JDChain. JDChain technology practice white paper. 2020 (in Chinese). https://storage.jd.com/jd.block.chain/2020%E4%BA%AC%E4%B8%9C%E5%8C%BA%E5%9D%97%E9%93%BE%E6%8A%80%E6%9C%AF%E5%AE%9E%E8%B7%B5%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf
[79] Yanan Chain. BSN Yanan Chain user manual. 2024 (in Chinese). https://ddc.bsnbase.com/yanan/book/index.html
[80] Zhijiang Chain. Introduction to smart contract development. 2022 (in Chinese). https://www.zhijiangchain.org.cn/
[81] Redis. Redis cloud quick start. 2023. https://github.com/redis/redis
[82] Aelf. Whitepaper 2.0. 2024. https://aelf.com/
[83] Ge Z, Loghin D, Ooi BC, Ruan PC, Wang TW. Hybrid blockchain database systems: Design and performance. Proc. of the VLDB Endowment, 2022, 15(5): 1092–1104 [doi: 10.14778/3510397.351040]
[84] Huang DX, Liu Q, Cui Q, *et al*. TiDB: A raft-based HTAP database. Proc. of the VLDB Endowment, 2020, 13(12): 3072–3084. [doi: 10.14778/3415478.3415535]
[85] Raju P, Kadekodi R, Chidambaram V, Abraham I. PebblesDB: Building key-value stores using fragmented log-structured merge trees. In: Proc. of the 26th Symp. on Operating Systems Principles. Shanghai: ACM, 2017. 497–514. [doi: 10.1145/3132747.3132765]
[86] Mixin. Mixin developers. 2024. https://developers.mixin.one/docs/introduction
[87] Chen YX, Pan AQ, Lei HL, Ye AD, Han S, Tang Y, Lu W, Chai YP, Zhang F, Du XY. TDSQL: Tencent distributed database system. Proc. of the VLDB Endowment, 2024, 17(12): 3869–3882. [doi: 10.14778/3685800.3685812]
[88] Bradshaw S, Chodorow K. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. 3rd ed., Sebastopol: O'Reilly Media, 2019.

[89] Anderson JC, Lehnardt J, Slater N. CouchDB: The Definitive Guide: Time to Relax. Sebastopol: O'Reilly Media, 2010.
[90] Z-Ledger. Ziggurat blockchain white paper. 2018 (in Chinese). https://static.zhigui.com/Ziggurat_Blockchain_WhitePaper_2018.pdf
[91] Codd EF. Relational database: A practical foundation for productivity. Communications of the ACM, 1982, 25(2): 109–117. [doi: 10.1145/358396.358400]
[92] Puangsaijai W, Puntheeranurak S. A comparative study of relational database and key-value database for big data applications. In: Proc. of the 2017 Int'l Electrical Engineering Congress (iEECON). Pattaya: IEEE, 2017. 1–4. [doi: 10.1109/IEECON.2017.8075813]
[93] Han J, E HH, Le G, Du J. Survey on NoSQL database. In: Proc. of the 6th Int'l Conf. on Pervasive Computing and Applications. Port Elizabeth: IEEE, 2011. 363–366. [doi: 10.1109/ICPCA.2011.6106531]
[94] Hankerson D, Vanstone S, Menezes A. Guide to Elliptic Curve Cryptography. New York: Springer, 2004. [doi: 10.1007/b97644]
[95] SM2. Crate SM2. 2024. https://docs.rs/sm2/latest/sm2/
[96] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA). Int'l Journal of Information Security, 2001, 1(1): 36–63. [doi: 10.1007/s102070100002]
[97] Simmons GJ. Symmetric and asymmetric encryption. ACM Computing Surveys, 1979, 11(4): 305–330. [doi: 10.1145/356789.356793]
[98] Koblitz N. An elliptic curve implementation of the finite field digital signature algorithm. In: Proc. of the 18th Annual Int'l Cryptology Conf. Lecture Notes in Computer Science. 1998. 327–337. Berlin: Springer. [doi: 10.1007/BFb0055739]
[99] Hash algorithm. 2024. https://en.wikipedia.org/wiki/Cryptographic_hash_function
[100] Ramanujam S, Karuppiah M. Designing an algorithm with high avalanche effect. IJCSNS Int'l Journal of Computer Science and Network Security, 2011, 11(1): 106–111.
[101] Secp256k1. Crate Secp256k1. 2024. https://docs.rs/secp256k1/latest/secp256k1/
[102] Nechvatal J, Barker E, Bassham L, Burr W, Dworkin M, Foti J, Roback E. Report on the development of the advanced encryption standard (AES). Journal of Research of the National Institute of Standards and Technology, 2001, 106(3): 511–577. [doi: 10.6028/jres.106.023]
[103] Tsiounis Y, Yung M. On the security of ElGamal based encryption. In: Proc. of the 1st Int'l Workshop on Public Key Cryptography. Pacifico Yokohama: Springer, 1998. 117–134. [doi: 10.1007/BFb0054019]
[104] Paillier P, Pointcheval D. Efficient public-key cryptosystems provably secure against active adversaries. In: Proc. of the 1999 Int'l Conf. on the Theory and Application of Cryptology and Information Security. Singapore: Springer, 1999. 165–179. [doi: 10.1007/978-3-540-48000-6_14]
[105] Bünz B, Bootle J, Boneh D, Poelstra A, Wuille P, Maxwell G. BulletProofs: Short proofs for confidential transactions and more. In: Proc. of the 2018 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2018. 315–334. [doi: 10.1109/SP.2018.00020]
[106] Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 2018, 51(4): 79. [doi: 10.1145/3214303]
[107] Barrett P. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In: Proc. of the 1987 Conf. on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1987. 311–323. [doi: 10.1007/3-540-47721-7_24]
[108] Appel AW. Verification of a cryptographic primitive: SHA-256. ACM Trans. on Programming Languages and Systems, 2015, 37(2): 7. [doi: 10.1145/2701415]
[109] Bertoni G, Daemen J, Peeters M, van Assche G. The Keccak SHA-3 submission. 2011. https://keccak.team/files/Keccak-submission-3.pdf
[110] Aumasson JP, Neves S, Wilcox-O'Hearn Z, Winnerlein C. BLAKE2: Simpler, smaller, fast as MD5. In: Proc. of the 11th Int'l Conf. on Applied Cryptography and Network Security. Banff: Springer, 2013. 119–135. [doi: 10.1007/978-3-642-38980-1_8]
[111] De Santis A, Persiano G. Zero-knowledge proofs of knowledge without interaction. In: Proc. of the 33rd Annual Symp. on Foundations of Computer Science. Pittsburgh: IEEE, 1992. 427–436. [doi: 10.1109/SFCS.1992.267809]
[112] Eagen L, Kanjalkar S, Ruffing T, Nick J. BulletProofs++: Next generation confidential transactions via reciprocal set membership arguments. In: Proc. of the 43rd Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Zurich: Springer. 249–279. [doi: 10.1007/978-3-031-58740-5_9]
[113] Pinto AM. An introduction to the use of zk-SNARKs in blockchains. In: Proc. of the 1st Int'l Conf. on Mathematical Research for Blockchain Economy. Santorini: Springer, 2020. 233–249. [doi: 10.1007/978-3-030-37110-4_16]
[114] Canetti R, Hohenberger S. Chosen-ciphertext secure proxy re-encryption. In: Proc. of the 14th ACM Conf. on Computer and Communications Security. Alexandria: ACM, 2007. 185–194. [doi: 10.1145/1315245.1315269]
[115] Loe AF, Quaglia EA. You shall not join: A measurement study of cryptocurrency peer-to-peer bootstrapping techniques. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 2231–2247. [doi: 10.1145/3319535.

3345649]
[116] Durumeric Z, Ma Z, Springall D, Barnes R, Sullivan N, Bursztein E, Bailey M, Halderman JA, Paxson V. The security impact of HTTPS interception. In: Proc. of the 2017 Network and Distributed System Security (NDSS) Symp. 2017. San Diego, 2017.
[117] Wang XW, Zhao H, Zhu JK. GRPC: A communication cooperation mechanism in distributed systems. ACM SIGOPS Operating Systems Review, 1993, 27(3): 75–86. [doi: 10.1145/155870.155881]
[118] CMBaaS. 2024 (in Chinese). https://it.10086.cn/services/cmbaas/
[119] Demers A, Greene D, Hauser C, Irish W, Larson J, Shenker S, Sturgis H, Swinehart D, Terry D. Epidemic algorithms for replicated database maintenance. In: Proc. of the 6th Annual ACM Symp. on Principles of Distributed Computing. Vancouver: ACM, 1987. 1–12. [doi: 10.1145/41840.41841]
[120] Wang X, Jiang X, Liu YX, Wang JP, Sun Y. Data propagation for low latency blockchain systems. IEEE Journal on Selected Areas in Communications, 2022, 40(12): 3631–3644. [doi: 10.1109/JSAC.2022.3213330]
[121] Banerjee-Mishra T, Sahni S. PubSub: An efficient publish/subscribe system. IEEE Trans. on Computers, 2015, 64(4): 1119–1132. [doi: 10.1109/TC.2014.2315636]
[122] Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S. On the security and performance of proof of work blockchains. In: Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security. Vienna: ACM, 2016. 3–16. [doi: 10.1145/2976749.2978341]
[123] Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In: Proc. of the 37th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2017. 357–388. [doi: 10.1007/978-3-319-63688-7_12]
[124] Castro M, Liskov B. Practical Byzantine fault tolerance. In: Proc. of the 3rd Symp. on Operating Systems Design and Implementation. New Orleans: USENIX Association, 1999. 173–186.
[125] Veronese GS, Correia M, Bessani AN, Lung LC, Verissimo P. Efficient Byzantine fault-tolerance. IEEE Trans. on Computers, 2013, 62(1): 16–30. [doi: 10.1109/TC.2011.221]
[126] Buchman E. Tendermint: Byzantine fault tolerance in the age of blockchains [MS. Thesis]. Guelph: The University of Guelph, 2016.
[127] Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery. ACM Trans. on Computer Systems, 2002, 20(4): 398–461. [doi: 10.1145/571637.571640]
[128] Decouchant J, Kozhaya D, Rahli V, Taïani F, Zappa N. DAMYSUS: Streamlined BFT consensus leveraging trusted components. In: Proc. of the 17th European Conf. on Computer Systems (EuroSys 2022). Rennes: ACM, 2022. 1–16. [doi: 10.1145/3492321.3519568]
[129] MAXBFT. The self-developed assembly line consensus algorithm project of Changan Chain 2.0 version. 2024 (in Chinese). https://git.chainmaker.org.cn/chainmaker/consensus-maxbft
[130] Micali S, Rabin M, Vadhan S. Verifiable random functions. In: Proc. of the 40th Annual Symp. on Foundations of Computer Science. New York: IEEE, 1999. 120–130. [doi: 10.1109/SFFCS.1999.814584]
[131] Vyper. Overview. 2024. https://docs.vyperlang.org/en/stable/
[132] WebAssembly. Guides. 2024. https://developer.mozilla.org/en-US/docs/WebAssembly
[133] Docker. Dockerdocs. 2024. https://docs.docker.com/docker-hub/
[134] Kubernetes. Kubernetes documentation. 2024. https://kubernetes.io/docs/home/
[135] Bathula PN, Sreenivasulu M. An integrated blockchain framework for secure data sharing in IoT fog computing. Tsinghua Science and Technology, 2025, 30(3): 957–977. [doi: 10.26599/TST.2024.9010082]
[136] Wang C, Zhang N, Wang C. Managing privacy in the digital economy. Fundamental Research, 2021, 1(5): 543–551. [doi: 10.1016/J.FMRE.2021.08.009]
[137] Lindell Y. Secure multiparty computation. Communications of the ACM, 2020, 64(1): 86–96. [doi: 10.1145/3387108]
[138] Martins P, Sousa L, Mariano A. A survey on fully homomorphic encryption: An engineering perspective. ACM Computing Surveys, 2017, 50(6): 83. [doi: 10.1145/3124441]
[139] Bender A, Katz J, Morselli R. Ring signatures: Stronger definitions, and constructions without random oracles. Journal of Cryptology, 2009, 22(1): 114–138. [doi: 10.1007/s00145-007-9011-9]
[140] Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: What it is, and what it is not. In: Proc. of the 2015 IEEE Trustcom/BigDataSE/ISPA. Helsinki: IEEE, 2015. 57–64. [doi: 10.1109/Trustcom.2015.357]
[141] Zamani M, Movahedi M, Raykova M. RapidChain: Scaling blockchain via full sharding. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 931–948. [doi: 10.1145/3243734.3243853]
[142] Xiong A, Zhang W, Song Y, Wang D, Li D, Guo QL, Bai DS. Asynchronous consensus algorithm integrating dynamic weight sharding

strategy. Chinese Journal of Electronics, 2025, 34(2): 600–611. [doi: 10.23919/cje.2023.00.313]
[143] Wood G. Polkadot: Vision for a heterogeneous multi-chain framework. White Paper, 2016, 21(2327): 4662.
[144] Rizun PR. Subchains: A technique to scale Bitcoin and improve the user experience. Ledger, 2016, 1: 38–52. [doi: 10.5195/ledger.2016.40]
[145] Dziembowski S, Faust S, Hostáková K. General state channel networks. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 949–966. [doi: 10.1145/3243734.3243856]
[146] Singh A, Click K, Parizi RM, Zhang Q, Dehghantanha A, Choo KKR. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. Journal of Network and Computer Applications, 2020, 149: 102471. [doi: 10.1016/j.jnca.2019.102471]
[147] Relay Chain. 2024. https://coinmarketcap.com/academy/glossary/relay-chain
[148] Liang ZY, Chen J, Du RY. XPull: A relay-based blockchain intercommunication framework achieving cross-chain state pulling. Chinese Journal of Electronics, 2024, 33(5): 1261–1273. [doi: 10.23919/cje.2023.00.004]
[149] Ou W, Huang SY, Zheng JJ, Zhang QL, Zeng G, Han WB. An overview on cross-chain: Mechanism, platforms, challenges and advances. Computer Networks, 2022, 218: 109378. [doi: 10.1016/j.comnet.2022.109378]
[150] Belchior R, Vasconcelos A, Guerreiro S, Correia M. A survey on blockchain interoperability: Past, present, and future trends. ACM Computing Surveys, 2021, 54(8): 168. [doi: 10.1145/3471140]
[151] Herlihy M. Atomic cross-chain swaps. In: Proc. of the 2018 ACM Symp. on Principles of Distributed Computing. Egham: ACM, 2018. 245–254. [doi: 10.1145/3212734.3212736]
[152] Shao QF, Zhang Z, Zhu YC, Zhou AY. Survey of enterprise blockchains. Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2571–2592 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5775.htm [doi: 10.13328/j.cnki.jos.005775]
[153] Top Blockchains. 2024. https://dappradar.com/rankings/chains
[154] Staheli D, Mancuso VF, Leahy MJ, Kalke MM. Cloudbreak: Answering the challenges of cyber command and control. Lincoln Laboratory Journal, 2016, 22(1): 60–73.
[155] Solana Turbine. 2024. https://www.helius.dev/blog/turbine-block-propagation-on-solana
[156] Moniz H. The Istanbul BFT consensus algorithm. arXiv:2002.03613, 2020.
[157] Quorum Byzantine fault tolerant consensus protocol. 2024. https://www.linkedin.com/pulse/ethereum-quorum-byzantine-fault-tolerant-qbft-consensus-singh-ibctf
[158] MATIC token. 2024. https://coinmarketcap.com/currencies/polygon/
[159] PoS-based tower BFT. 2025. https://www.helius.dev/blog/consensus-on-solana
[160] Proof of history. 2025. https://solana.com/it/news/proof-of-history
[161] Duan SS, Zhang HB, Sui X, Huang BH, Mu CC, Di G, Wang XY. Dashing and star: Byzantine fault tolerance with weak certificates. In: Proc. of the 19th European Conf. on Computer Systems. Athens: ACM, 2024. 250–264. [doi: 10.1145/3627703.3650073]
[162] Yul. Documentation. 2024. https://docs.soliditylang.org/en/latest/yul.html
[163] Low-level lisp-like language. LLL introduction. 2024. https://lll-docs.readthedocs.io/en/latest/lll_introduction.html
[164] Bamboo: A language for morphing smart contracts. 2024. https://github.com/Betterpath/Bamboo
[165] eWASM. Ethereum WebAssembly (ewasm). 2024. https://ewasm.readthedocs.io/en/mkdocs/
[166] SeaLevel. Crate SeaLevel_tools. 2024. https://docs.rs/sealevel-tools/latest/sealevel_tools/
[167] Berkeley packet filter. BPF documentation. 2024. https://docs.kernel.org/bpf/
[168] Monero. Overview. 2024. https://docs.getmonero.org/interacting/overview/
[169] Zcash. Zcash documentation. 2019. https://zcash.readthedocs.io/en/latest/
[170] Duffield E, Diaz D. Dash: A privacycentric cryptocurrency. 2015. https://www.academia.edu/16195039/Dash_A_Privacy_Centric_Crypto_Currency
[171] OASIS. OASIS docs. 2024. https://docs.oasis.io/general/
[172] Zilliqa. Zilliqa 2.0 whitepaper. 2024. https://drive.google.com/file/d/1XqSySl0w_OtxyxBZ0ahS06cqIv-vd10M/view
[173] Elrond. Elrond whitepaper. 2019. https://whitepaper.io/document/510/elrond-whitepaper

附中文参考文献

[12] BSN. 区块链服务网络用户手册. 2024. https://zhuanwang.bsnbase.com/static/tmpFile/%E5%8C%BA%E5%9D%97%E9%93%BE%E6%9C%88D%E5%A%A1%E7%BD%91%E7%BB%9C%E7%94%A8%E6%88%B7%E6%89%8B%E5%86%8C.pdf
[27] 联通链 BaaS 平台. 应用场景. 2021. https://support.cucloud.cn/document/127/652/674.html?id=674&arcid=945
[35] 至信链. 产品简介. 2024. https://console.zxchain.qq.com/docs?docpath=doc/content/%E5%BF%AB%E9%80%9F%E5%85%A5%E9

%97%A8/%E5%87%86%E5%A4%87%E5%B7%A5%E4%BD%9C.html
[36] 长安链. 长安链底层技术平台介绍. 2024. https://docs.chainmaker.org.cn/v2.3.7/html/quickstart/%E9%95%BF%E5%AE%89%E9%93%BE%E5%9F%BA%E7%A1%80%E7%9F%A5%E8%AF%86%E4%BB%8B%E7%BB%8D.html
[37] 袁勇, 王飞跃. 区块链技术发展现状与展望. 自动化学报, 2016, 42(4): 481–494. [doi: 10.16383/j.aas.2016.c160158]
[48] 《软件和信息技术服务业发展规划 (2016~2020 年)》发布. 军民两用技术与产品, 2017, (3): 5. [doi: 10.19385/j.cnki.1009-8119.2017.03.005]
[49] 2018 年信息化和软件服务业标准化工作要点发布 聚焦六领域重点标准研制. 信息技术与标准化, 2018, (4): 4.
[50] 安顺链技术文档. 2023. https://docs.anshun.tech/kai-fa-zhi-nan/kuai-su-ru-men
[51] 泰安链. 开发和使用智能合约. 2024. https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/app_dev/index.html
[53] 飞梭. FISCO BCOS 2.0 技术文档. 2024. https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/
[54] 星火链. 星火链开发指南 3.0. 2024. https://bif-doc.readthedocs.io/zh-cn/3.0.0/
[55] 蜀信链. 技术文档. 2024. https://www.btsi.org.cn/resourceCenter
[56] 桂链. 区块链开发服务. 2024. https://www.gxbaas.cn/develop-docs/1733190606800301.html
[57] 蚂蚁链. 合约平台概述. 2019. https://antdigital.com/docs/11/143759?Source=bs_baidu_my_55480&bd_vid=8090731222145646492
[59] 万纳链. Venachain 简介. 2024. https://www.wxblockchain.com/#/home
[60] 布比链. 布比区块链介绍. 2024. https://docs.bubi.cn/docs/intro/index/
[61] 趣链. 趣链区块链浏览器. 2024. https://hyperchain.readthedocs.io/zh-cn/tmp-2.12.0_a/qulian%20blockchain%20browser.html
[62] 天河链. 天河链产品文档. 2024. https://thgy.yuque.com/ldgxec/tichain/vnvcgp?#
[63] 迅雷链. 合约开放平台开发指南. 2024. https://blockchain.xunlei.com/site/docopen.html
[65] 磁云数字. 磁云区块链 M0 技术白皮书. 2020. https://cyjf.oss-cn-beijing.aliyuncs.com/dea737b7578b48b999868c5a44917a6b.pdf
[66] 百度超级链. XuperChain 官方文档 5.3. 2024. https://xuper.baidu.com/n/xuperdoc/v5.3/quickstart/quickstart.html
[67] RepChain. RepChain 文档. 2024. https://zls201624.github.io/repchain-doc/RepChain-2.0.html
[68] Chain33 平台. 产品文档. 2024. https://chain.33.cn/document/60#
[69] 梧桐链. 介绍文档. 2024. https://github.com/tjfoc/wutongchain/blob/master/sample/chaincode/doc/README.MD
[70] Conflux 树图公有链. 2025. https://www.tree-graph.org.cn/
[71] 华为云区块链. 区块链服务开发指南. 2025. https://support.huaweicloud.com/devg-bcs/bcs-devg-pdf.pdf
[72] IRITA. IRITA 文档. 2024. https://irita.bianjie.ai/docs/
[73] 文昌链. 帮助文档. 2024. https://docs.avata.bianjie.ai/
[74] 武汉链. 2024. https://github.com/BSN-DDC/wuhanchain?tab=readme-ov-file
[75] 海峡链. 海峡链技术白皮书-整体篇. 2022. https://baijiahao.baidu.com/s?id=1732312699883732607&wfr=spider&for=pc
[77] 星云. 介绍文档. 2019. https://wiki.nebulas.io/zh-cn/latest/go-nebulas/README.html
[78] 京东智臻链. 京东区块链技术实践白皮书. 2020. https://storage.jd.com/jd.block.chain/2020%E4%BA%AC%E4%B8%9C%E5%8C%BA%E5%9D%97%E9%93%BE%E6%8A%80%E6%9C%AF%E5%AE%9E%E8%B7%B5%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf
[79] 延安链. BSN 延安链使用手册. 2024. https://ddc.bsnbase.com/yanan/book/index.html
[80] 之江链. 智能合约开发简介. 2022. https://www.zhijiangchain.org.cn/
[90] 纸贵科技. 纸贵区块链白皮书. 2018. https://static.zhigui.com/Ziggurat_Blockchain_WhitePaper_2018.pdf
[118] 中移链. 2024. https://it.10086.cn/services/cmbaas/
[129] MAXBFT. 长安链 2.0 版本自研流水线共识算法项目. 2024. https://git.chainmaker.org.cn/chainmaker/consensus-maxbft
[152] 邵奇峰, 张召, 朱燕超, 周傲英. 企业级区块链技术综述. 软件学报, 2019, 30(9): 2571–2592. http://www.jos.org.cn/1000-9825/5775.htm [doi: 10.13328/j.cnki.jos.005775]

作者简介

何嘉昊, 博士生, 主要研究领域为区块链安全, 软件安全.
杨森, 硕士生, 主要研究领域为区块链安全, 软件安全.
潘子玲, 硕士生, 主要研究领域为区块链安全, 区块链地址分类, 机器学习.
陈厅, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为区块链安全, 软件安全, 软件工程.
沈剑, 博士, 教授, 博士生导师, 主要研究领域为公钥密码学, 数据安全, 区块链.