XT Chain V1.0

XT Chain

项目白皮书

打造万物平衡的全领域生态公链

XT Chain

Bilanx Chain

前 言

在人类社会化进程中，社会以优胜劣汰的形式进步和更迭。从远古石器时代

到如今的互联网、共享经济时代，每一次核心技术出现，都会极大解决当下社会

中生产、经济、沟通等问题，推动社会进步。随着社会飞速发展，科技进步，生活节奏几何倍增，信息不可靠、信用资源缺失的情况愈发严重，政府、企业、个人之间的信任体系愈发脆弱，沟通和交易成本增加。

我们认为区块链技术在这个经济快速发展的时代，以其去中心化，防篡改，

高度透明等特性，会成为继 PC 互联网、移动互联网后又一个革新人类社会的技术，将会让社会各种关系的信任变得更加简单。区块链自 21 世纪初期发展起来，是目前全世界认为最有潜力、最具想象力的一种技术革新。在人类的发展史上共经历过三次工业革命，第一次以蒸汽机的发明为标志，让机器代替了手工劳动；第二次以电能的突破、应用和内燃机的发明为标志，直接推动人类进入电气化时代；第三次以电子计算机、核能、空间技术、生物工程的发明和应用为标志，不仅推动人类社会的巨大变革，更深刻地影响了人类的生活和思维方式。每一次工业革命都带来生产力的巨大提升，而作为生产要素之一的生产关系，改变并没有那么巨大，依旧是自上而下、金字塔层级的中心化组织。组织的业务越复杂，层级越多，效率提升就越困难。区块链是去中心化、去信任化的网络，可以实现点对点价值交换，被人们称之为价值互联网。

XT Chain 认为区块链技术最有可能改进当前的生产关系。

在 XT Chain 的帮助下，我们可以创造这样一个世界——一个人和人直接相连，去信任化的，在社区或者社会共识下，相互协作、点对点相互交换、价值驱动的世界。

Contents | 目 录

行业背景

Business Background

打造万物平衡的全领域生态公链

Create a Full-Field Ecological Public Chain that Balances Everything

I 行业背景

※ 公链之争

公链，作为各种 DAPP 应用的底层基础设施。只有搭建高速、稳定的基础公链，才能使各种落地应用的侧链和智能合约有底层操作系统可用，才是孕育杀手级应用的土壤和基石。可扩展性是公有区块链的基本前提，特别是当高吞吐量，高并发性，稳定性和安全性是影响用户体验的关键因素时。

在 2020 年，可扩展性仍然是一项对于公链的艰巨挑战，因为由第三方审核的主流区块链平台的 TPS（包括比特币，以太坊，Ripple，EOS，Litecoin，Cardano，Tron 和 IOTA）通常低于 1000 TPS。这些区块链平台中的大多数不能满足现实应用程序的速度和并发性要求。

为了解决这一问题，区块链行业在进行以下尝试：其一是大区块，即提升每一个区块的储存空间，此方法在比特币中曾有过试用，比特币一个区块大小仅有 1MB 储存空间，在 2017 年年底时 SegWit2x 将区块大小从 1MB 提升至 2MB，但出于安全考虑，最终取消 SegWit2x 硬分叉；其二是链下交易，即在主链外加入闪电网络或者侧链，此方法以太坊在尝试，提前支付一些以太坊或比特币作为押金，之后在链下通过其他方法来跟其他人进行交易；其三是代理人共识协议，即多人成为超级节点，形成小团体。EOS 在尝试用这种“议会制”共识产生区块，再将区块广播给整个网络，从而达成整个网络的共识。

然而这三种尝试方式都不能完美解决区块链的交易速度，去中心化和安全性的矛盾。

XT Chain 正是致力于打造一个可拓展、可相容的区块链底层构架，基于公式算法的创新共识机制相结合的 Paxos 共识机制。在保证 BFT 系统强一致性的前提下，提升了系统的整体交易吞吐能力以及系统稳定性，可以稳定达到百万的 TPS，交易确认时间控制在 3s，真正突破行业瓶颈，实现区块链技术商业价值。

※ DAPP 市场痛点

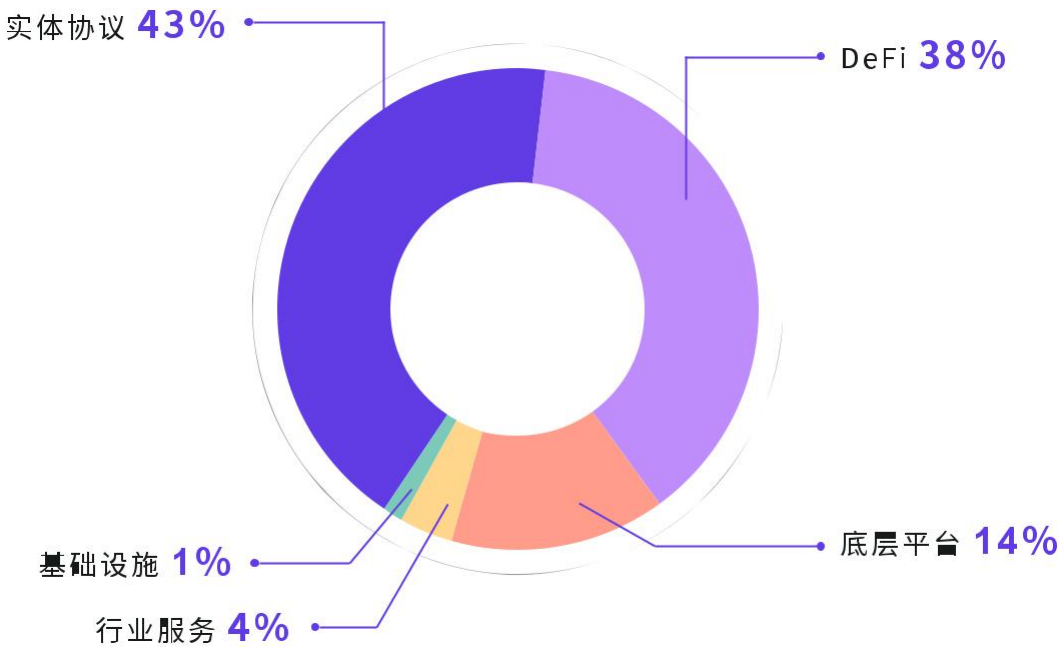
在 2017 年由 ETH 所开启的公链热潮中，DAPP 作为一种去中心化的应用设施，进入到了人们的视野中。

DAPP 是基于智能合约的应用，可以跑在一台能与区块链节点交互的中心化服务器上，也可以跑在任意一个区块链平等节点上。与一般的 Web 应用 APP 跑在普通的服务器上相比，它们需要提交交易到区块链并且从区块链而不是中心化数据库读取重要数据。相对于典型的用户登录系统，用户有可能被表示成一个钱包地址而其它用户数据保存在本地。

对于传统的 Web 应用 APP，服务器通常由服务提供商（如 AWS，Heroku 或 VPS）提供，所有用户都与这一个中心应用程序进行交互，所有用户数据也都存储在该服务商平台上，用户对于平台有极大的依赖性。

2019 年以来，Defi（Decentralized Finance）概念大热并且经历了 “服务于金融业”到“彻底改变金融业”的升华，成为了区块链十年发展史上的里程碑，在区块链行业占据重要地位。

金融业是区块链技术应用的重点发展方向，有超过三分之一的区块链项目可归类为 Defi。以学界对金融的定义为准，并根据 BICS（Blockchain Industry Classification Standard）分类，在目前市值前 1000 名的通证中，至少有 38% 的区块链项目直接服务



在 2017 年由 ETH 所开启的公链热潮中，DAPP 作为一种去中心化的应用设施，进入到了人们的视野中。

DAPP 是基于智能合约的应用，可以跑在一台能与区块链节点交互的中心化服务器上，也可以跑在任意一个区块链平等节点上。与一般的 Web 应用 APP 跑在普通的服务器上相比，它们需要提交交易到区块链并且从区块链而不是中心化数据库读取重要数据。相对于典型的用户登录系统，用户有可能被表示成一个钱包地址而其它用户数据保存在本地。

对于传统的 Web 应用 APP，服务器通常由服务提供商（如 AWS，Heroku 或 VPS）提供，所有用户都与这一个中心应用程序进行交互，所有用户数据也都存储在该服务商平台上，用户对于平台有极大的依赖性。

2019 年以来，Defi（Decentralized Finance）概念大热并且经历了“服务于金融业”到“彻底改变金融业”的升华，成为了区块链十年发展史上的里程碑，在区块链行业占据重要地位。

金融业是区块链技术应用的重点发展方向，有超过三分之一的区块链项目可归类为 Defi。以学界对金融的定义为准，并根据 BICS（Blockchain Industry Classification Standard）分类，在目前市值前 1000 名的通证中，至少有 38% 的区块链项目直接服务于金融行业，包括非银金融、钱包 & 交易、通证资管、稳定通证、银行服务和支付结算。

Compound 则是去中心化借贷平台，存币者可以将自己拥有的 Token 转入 Compound 智能合约（存币），并在未来时刻将存的 Token 从 Compound 智能合约转回自己的地址（取币）。借币者可以将存入的 Token 作为抵押品从 Compound 借币。借币者借到的 Token 可以与自己存入的 Token 在数量和类型上不一致，但要满足超额抵押率要求。如果借币者的抵押品不够，Compound 协议会强制清算抵押品。

但是 Compound 平台的风险也是显而易见的，因为基于超额抵押，抵押品价格波动非常大，如果急剧下跌，可能出现抵押品不足的情况。按照协议有两种处理方式，一种是补充抵押品，第二是智能合约会清算抵押。因此，一旦价格急剧下跌，不管是追加抵押品还是抵押品的处置，都会在以太坊的区块链上进行交易，就会出现交易拥堵的情况，市场资产风险就很难进行出清。

针对于 Defi 借贷应用，XT Chain 提出了两个重要解决思路。一是走向点对点借贷模式，让存币和借币期限匹配，在借鉴早期的 ETHLend 的基础上做了更多的升级；二是通过算法来动态管理存币和借币的期限。

目前，Defi 整体锁仓量已经超过 43 亿美金，但是整个市场仍处于发展初期，依然面临着公链性能、社会需求和安全风险等问题。

纵观整个 DAPP 市场，仍然存在巨大的痛点，即存在一种“同而不和”的尴尬局面，正所谓有币无链的缺“链”之痛。

目前 DAPP 在共识机制必须与主链保持一致（例如以太坊目前的 POW+POS），因

此在业务逻辑和实现场景上难免会出现削足适履的局面，所以当前大部分的 DAPP 主要集中去中心化交易，以及基于转账交易场景构建的养成类及博彩类游戏，应用场景十分单一。

更重要的是，由于在以太坊上缺少降低原有 APP 码农参与 DAPP 开发门槛的基础服务商，导致存量本就不多的区块链应用开发人才无法满足几何级别增长的开发需求，整个行业生态的多样性严重不足，DAPP 经济的普及与落地进度缓慢。再就是，现在大多数公链项目所提出的目标仍然只是围绕着如何提高转账交易的性能，并没有针对如何改善智能合约层面的流量处理以及提高其灵活性投入足够的关注。

公链的 DAPP 生态建设是重中之重，这将真正考验公链的实力和可持续性发展水平。旨在解决区块链行业的发展痛点，为面临瓶颈阶段的公链市场纾困，为去中心化商业生态赋能，XT Chain 应运而生。

通过搭建聚合了公链、钱包、DAPP、数字资产交易所等内容的一体化去中心化商业生态，XT Chain 打破了行业原有的商业壁垒，降低了市场准入门槛，用户无需再经历繁琐的充提和兑换环节，可以真正体验到去中心化的一站式服务。

在 DAPP 层面，Defi 和游戏双管齐下打造多样性的应用场景，在高性能底层公链的扎实基础上，每个开发者都可以参与到 XT Chain 并创建自己 DAPP。对于普通投资者和用户而言，资产上链锁定，同时在 Defi 游戏、线上商城等场景拥有多样化的使用场景和盈利方式，同时基于公链主网通证 XT 未来广阔的升值空间，用户可以利用通证体系实现价值变现，感受复利投资的福祉，最终 XT Chain 便形成了具备自驱力的 DAPP 生态闭环。

XT Chain 是一种全新的区块链体系架构，定位为易用的高性能区块链平台，旨在实现分布式应用的性能扩展，以满足现实世界的真实商业需求。这是通过创建一个可以构建应用程序的类似操作系统的架构来实现的。该体系架构提供账户、身份与许可证管理、策略管理、数据库、异步通信以及在数以千计的 CPU、FPGA 或群集上的程序调度。该区块链为一个全新的体系架构，通过低延时高并发硬件加速技术，可实现每秒数百万

个交易，且达到秒级确认。

以太坊在一定程度上实现了基于计算机应用方面的基础设置，但是层次还是很低，特别是其智能合约只能根据链上其他账户的状态或其他智能合约的状态来触发，并且以太坊的智能合约会在所有的节点上都运行一次，极大的浪费了算力。

但 XT Chain 团队设计的模型使得链上链下能够打通，应用通过设置一系列的可信数据源，在现实中的特定事件发生后即可启动链上的智能合约和智能应用。应用通过设置阈值，可以指定智能合约仅在少量节点上执行，极大的节省了全网算力，也可以指定少量节点进行主要数据的传输和存储，这就使得大数据分析以及人工智能运算在区块链应用成为可能。

XT Chain 的愿景是构建未来全球商业区块链基础设置，成为区块链世界的 ios 操作系统，无论是机构用户还是个人用户都能很方便的在 XT Chain 上搭建自己的智能合约、区块链应用、甚至另一个区块链。

II

XT Chain项目介绍

XT Chain Project Introduction

打造万物平衡的全领域生态公链

Create a Full-Field Ecological Public Chain that Balances Everything

Ⅱ XT Chain 项目介绍

※ 项目简介

XT Chain 是一条基于 Paxos 共识机制的公有链，未来将打造涵盖“XT 去中心化交易所 + 公链钱包 +Defi 产品 + 区块链跨境电商平台 + 多元化 DAPP 应用”的区块链全领域生态。

XT Chain 类似于以太坊，可供开发者通过编写智能合约开发应用程序服务于用户。XT Chain 也是一款全民共建共享的区块链开发应用系统，致力于解决现有比特币、以太坊、EOS 三大区块链基础网络因其开发语言所导致的拓展性差、二次开发难度大等问题。

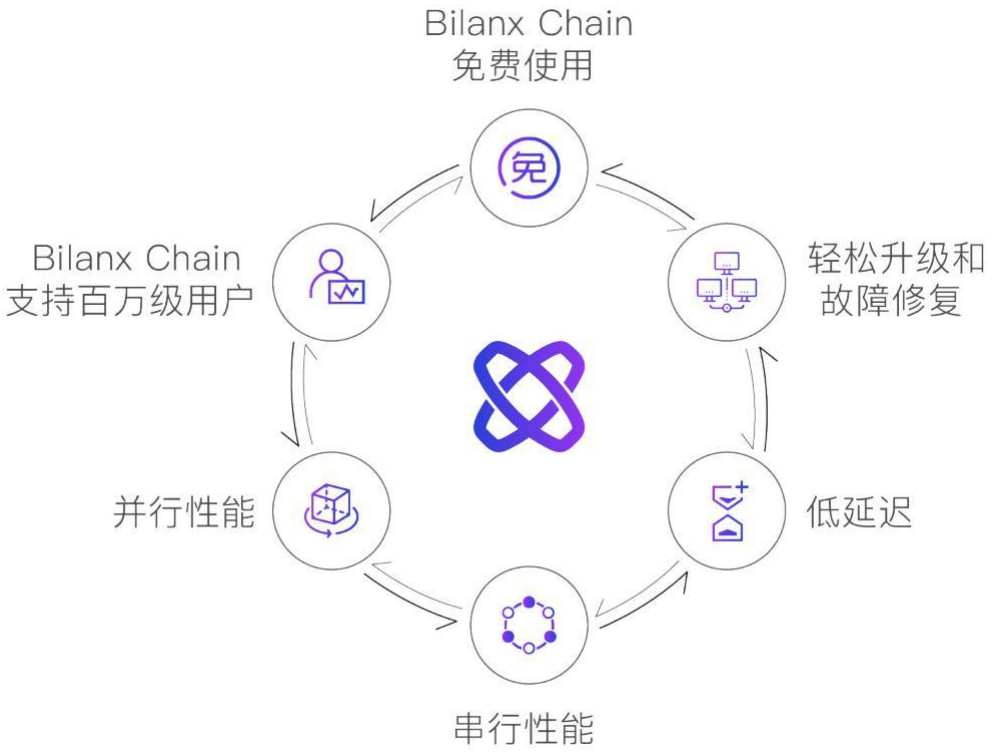
XT Chain 采用使用更加广泛、二次开发难度更低的 Java 语言进行操作系统开发，并且封装大量的 API 接口以便非专业区块链开发人员来使用，大的降低了普通开发人员进行去中心化区块链应用的开发，大幅促进区块链技术的发展。

针对当前区块链行业的挑战，XT Chain 在区块链技术和理念上进行了一系列的创新：包括基于 PBFT 和 DPOS 共识机制相结合的 Paxos 共识机制，区块链主控合约的理念和实现，去中心化钱包的应用，交易账本和智能合约账本的分离等，使得 XT Chain 成为区块链世界与现实商业世界的桥梁。

※ XT Chain 来源

XT 在拉丁语意为平衡，寓意为万物平衡

※ 核心优势



* 核心优势

XT Chain 支持百万级用户

如 Ebay，Uber，AirBnB 和 Facebook 这样的应用，需要能够处理数千万日活跃用户的区块链技术。在某些情况下，应用程序可能无法正常工作，除非达到了大量用户，因此可以处理大量用户数量的平台至关重要

XT Chain 免费使用

有时候应用开发人员需要灵活的为用户提供免费服务，用户不必为了使用平台而付出费用。可以免费使用的平台自然可能会得到更多的关注，有了足够的用户规模，开发者和企业可以创建对应的盈利模式。

轻松升级和故障修复

基于 XT Chain 的应用程序在进行功能迭代的时候自然需要能支持软件升级。所有软件都有可能受到 bug 的影响，一个区块链底层平台在遭遇 bug 的时候，需要能够从 BUG 中修复错误。

低延迟

及时的回馈是良好用户体验的基础。延迟时间如果超过了几秒钟，会大大影响用户体验，严重降低程序的竞争力。

串行性能

有些应用程序由于命令执行必须是顺序的，从而无法用并行算法进行实现。诸如交易所之类的应用经常需要处理大量的串行操作，因此一个成功的区块链架构需要具有强大的串行性能。

并行性能

大规模应用程序需要在多个 CPU 和计算机之间划分工作负载，XT Chain 可实现多端并行功能。

※ 全领域生态

XT Chain 公有链，所建立起来的是一个完全点对点的“去中心化”网络，其设计模型使得链上链下能够打通，应用通过设置一系列的可信数据源，在现实中的特定事件发生后即可启动链上的智能合约和智能应用。应用通过设置阈值，可以指定智能合约仅在少量节点上执行，极大的节省了全网算力，也可以指定少量节点进行主要数据的传输和存储，这就使得去中心化金融、供应链、物联网、分布式商业、商品溯源等区块链应用场景成为可能。

XT Chain 搭建的全新区块链体系架构提供账户、身份与许可证管理、策略管理、数据库、异步通信以及在数以千计的 CPU、FPGA 或群集上的程序调度。该区块链为一个全新的体系架构，通过低延时高并发硬件加速技术，可实现每秒数百万个交易，且达到秒级确认，是一个性能远高于以太坊底层操作系统。

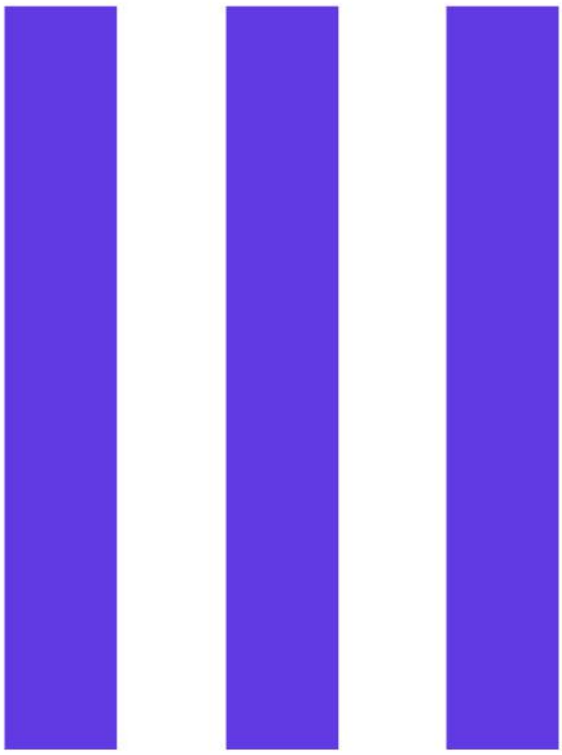
同时，XT Chain 的的智能合约执行引擎 XT ChainVM（XT Chain Virtual Machine）采用模块化可拔插的设计方式，首先开发的是支持 Java 语言的执行引擎 XT Chain JVM，后续会提供了支持 Solidity 语言的执行引擎 XT Chain EVM。 XT Chain JVM 是为了最大程度利用开源社区在智能合约技术和经验方面的积累，提

高智能合约的重用性而借鉴了以太坊的 EVM 的虚拟机。XT ChainVM 的智能合约实现完全兼容 Ethereum 的智能合约规范，使用 JAVA 作为智能合约的开发语言，通过微服务的架构设计以及多重安全检查机制为原生 Java 智能合约执行提供了一个高性能安全的执行沙盒，最大限度规避代码漏洞的问题。

XT Chain 的 DAPP 生态可以提供给开发人员更广阔的舞台，与更为丰富的行业相结合。基于区块链自身去中心化、不可篡改、可匿名等特点，区块链天然适合于与一些领域结合，诸如内容、游戏等。简而言之，人们可能不需要很多公链，但会需要很多应用。DAPP 极有可能引爆下一波区块链用户流量。区块链的应用场景不应仅仅是“炒币”，一个好的行业状态应该是用户为了使用区块链的服务和产品而持币，这样的流量才是经久不衰，穿越牛熊的。因此，未来 XT Chain 的 DAPP 生态想象空间巨大。

XT Chain 在 Dapp 上进行了大量技术投入，就是为了解决行业难题，为今后区块链系统的 Dapp 引入开辟一条崭新道路，在兼容时下主流公链合约（erc20 合约等）的同时，就主流开发语言 JAVA 有着大大的友好度。

过硬的底层技术加持。未来成熟的底层公链应该具备比较完善的技术特质，诸如：“创新的智能合约、“分层”、“分片”、“侧链”、“跨链”、“多链并行”、“代理重加密”和“海量存储”等技术，XT Chain 将这些技术在自己身上一一变为了现实，开创了公链界的一个先河。



技术架构和安全体系

Technical Architecture And Security System

打造万物平衡的全领域生态公链

Create a Full-Field Ecological Public Chain that Balances Everything

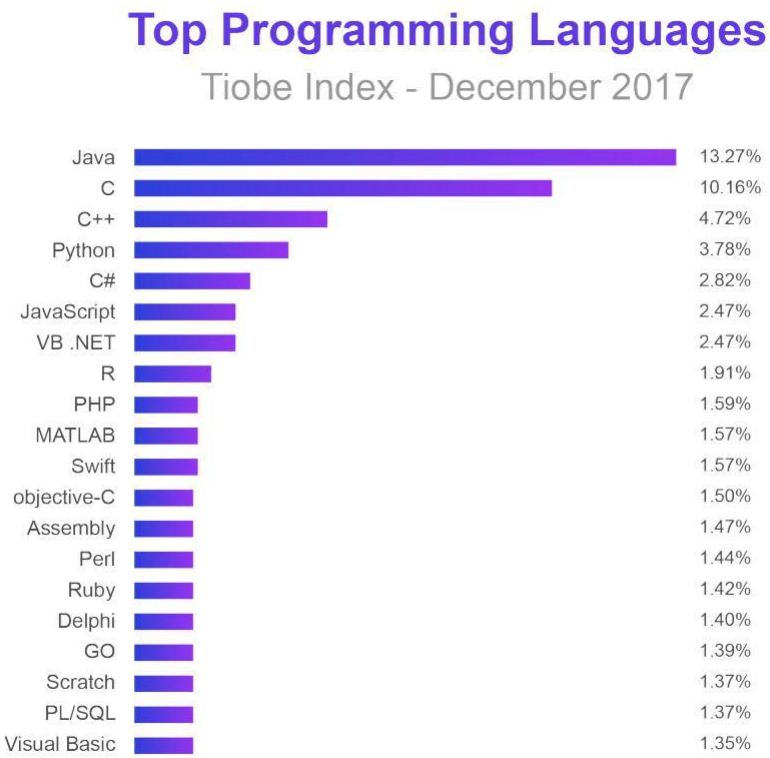
Ⅲ 技术架构和安全体系

※ 设计理念

编程语言 (Programming Language)，是用来定义计算机程序的形式语言。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。

编程语言俗称 " 计算机语言 "，种类非常的多，总的来说可以分成机器语言、汇编语言、高级语言三大类。电脑每做的一次动作，一个步骤，都是按照已经用计算机语言编好的程序来执行的，程序是计算机要执行的指令的集合，而程序全部都是用我们所掌握的语言来编写的。所以人们要控制计算机一定要通过计算机语言向计算机发出命令。目前通用的编程语言有两种形式：汇编语言和高级语言。

流行的区块链开发语言，开发语言流行度排行



三大区块链基础网络开发语言

语言网络	主开发语言	智能合约开发语言
比特币系列	C++	——
以太坊	Go	Solidity、LLL、Serpent
EOS	C++	C、C++、WebAssembly
XT Chain	java/go	solidity

Java 语言

Java 是由 Sun 最初设计用于嵌入程序的可移植性 " 小 C++"。在网页上运行小程序的想法着实吸引了不少人的目光，于是，这门语言迅速崛起。事实证明，Java 不仅仅适于在网页上内嵌动画 - 它是一门极好的完全的软件编程的小语言。" 虚拟机 " 机制、垃圾回收以及没有指针等使它很容易实现不易崩溃且不会泄漏资源的可靠程序。计算机语言虽然不是 C++ 的正式续篇，Java 从 C++ 中借用了大量的语法。它丢弃了很多 C++ 的复杂功能，从而形成一门紧凑而易学的语言。不像 C++，Java 强制面向对象编程，要在 Java 里写非面向对象的程序就像要在 Pascal 里写 " 空心粉式代码 " 一样困难。

Java 语言的优点：二进制码可移植到其他平台、程序可以在网页中运行、内含的类库非常标准且极其健壮、自动分配合垃圾回收避免程序中资源泄漏、网上数量巨大的代码例程。

在三大区块链基础网络开发语言中，C++ 比较主流，但是公认的难度很大的语言，需要很高的技术水平才能把控。Go 语言还没有能够被广泛使用，仍存在一些短板。至于 Solidity、LLL、Serpent、WebAssembly 这几个智能合约开发语言，更是很少有人听说过。

这些主网开发语言和智能合约的开发语言的复杂性和高学习成本，严重制约了区块链应用的发展。

JAVA 作为一个使用最广泛的开发语言，融入到了当今各个行业的各个系统中去。同时，以太坊的 token 和智能合约方式是目前使用最为广泛和成熟的。使用 JAVA 语言进行开发，并完成以太坊标准的智能合约机制，构建简单易懂的区块链技术开发社区，能够促进区块链技术快速融入现有行业中去。

基于上述原因，XT Chain 选择用 JAVA 语言来实现主网的开发。XT Chain 技术路线是用纯 Java 语言，基于 PBFT 算法的改进，实现了 XT Chain 的高性能共识算

法 ABFT。XT Chain 主网实现了以太坊的 ERC20 token 标准【后续会扩展为 ERC-827 标准】，使用 Java 语言为智能合约的开发语言。并且封装大量的 API 接口以便非专业区块链开发人员来使用，大的降低了普通开发人员进行去中心化区块链应用的开发，更快促进区块链技术的发展。

※ 技术方案

XT Chain 通过构建一个需求导向、奖励数据贡献、分包实现数据结构化的社群网络，打造一个安全、高效、可溯源、无数据沉淀、可再深度开发的数据交易平台，通过数据竞赛和数据去中心化交易来系统化地解决上述痛点。结合区块链网络与数据价值发掘、交换的需求，XT Chain 的构架设计遵循如下基本原则：

交易可信

事务历史记录进入区块链并永久存证，交易双方低成本信任交易网路，这要求网络兼具可靠性与私密性，同时规避数据沉淀等中介问题。

激励相容

经济体系设计促进网络节点发布数据，基于数据开发衍生价值发掘功能，同时参与整个生态开发促进数据价值发掘、交换成本不断下降。

精细化交易模式

通过网络机制设计促进数据资产交易更趋性细化。

市场化数据定价

通过网络机制设计保证市场化数据定价更细致化。

支持高并发交易

支持高并发量数据交换，实验室网络环境可达百万级，成为未来海量人工智能、物联网、机器人进行大规模数据采集、交换、边缘计算结果交换等能力的基础设施。

支持数据质量验证

支持对数据进行采样、交叉比对、格式比对、类型识别、范围识别等自动化验证手段，在保障交易安全的同时，默认提供多种数据质量验证的能力设置。

支持衍生数据服务

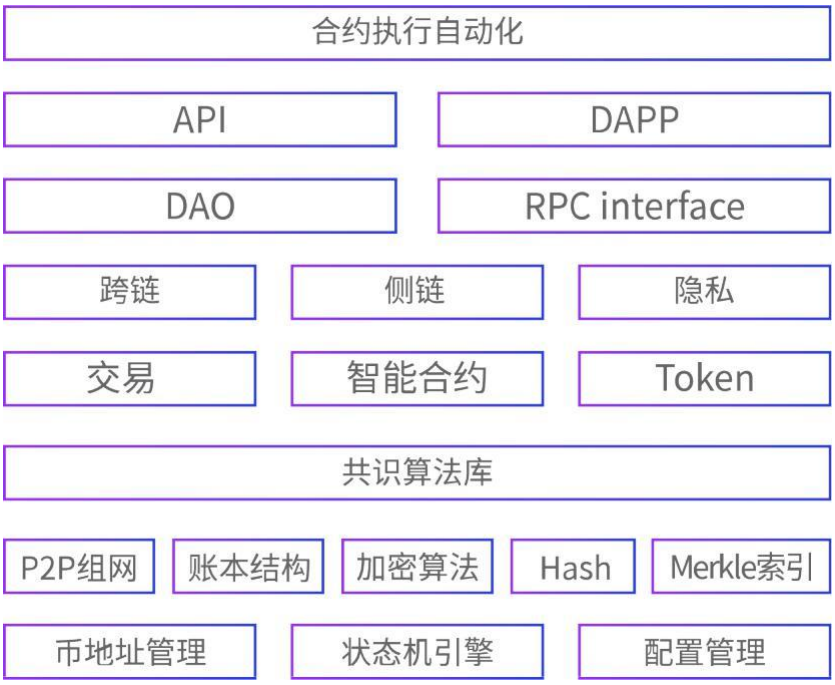
支持对数据进行可编程模型计算的功能，支持开发者采用通用语言编制更复杂的数据分析工具并接入 XT Chain，支持接入 Oracle 网络提升智能合约功能范围。

支持跨链区块链服务

支持在数据存储、计算能力方面具有成熟解决方案的区块链服务接入 XT Chain。

整体架构

XT Chain 遵循成熟的六层技术架构，自上而下分别为：数据层、网络层、共识层、激励层、合约层、应用层。



* 整体架构

1）数据层 / Data Layer

数据层是整个区块链技术中最底层的数据结构，主要描述区块链的最基本的物理形式，是一个区块 + 链表的数据结构，包括有：区块链的区块数据、哈希函数、merkel 数、非对称公私钥数据加密技术、时间戳技术等内容。

2）网络层 / Network Layer

区块链网络本质是一个 P2P（Peer-to-peer 点对点）的网络，网络中的资源和服务分散在所有节点上，信息的传输和服务的实现都直接在节点之间进行，而无需中间环节或中心化的服务器介入。每一个节点既接收信息，也产生信息，节点之间通过维护一个共同的区块链来同步信息，当一个节点创造出新的区块后便以广播的形式通知其他节点，其他节点收到信息后对该区块进行验证，并在该区块的基础上去创建新的区块，从而达到全网共同维护一个底层账本的作用。所以网络层会涉及到 P2P 组网机制、数据传播机制、数据验证机制等的设计，而这些设计都能影响到区块信息的确认速度，所以，网络层是如何突破区块链技术可扩展这个瓶颈的重要研究方向。

3）共识层 / Consensus Layer

共识层封装了共识算法和共识机制，能让高度分散的节点在去中心化的区块链网络中高效地针对区块数据的有效性达成共识，是区块链的核心技术之一，也是区块链社群的治理机制。它的主要作用是决定了谁来进行记账，而记账的方式又影响整个系统的安全性和可靠性。

目前至少有数十种共识机制算法，包含工作量证明（POW）、权益证明（POS）、权益授权证明（DPOS）等等。

4）激励层 / Actuator Layer

激励层就是大家常说的挖矿机制，它将经济因素集成到区块链技术体系中来并设计出一套经济激励模型，鼓励节点来参与区块链的安全验证工作，包括经济激励的发行机制和分配机制等。

激励层主要出现公链当中，因为公有链必须激励参与记账的几点，并且惩罚不遵守规则的节点，才能让整个系统朝着良性循环的方向发展。而在私有链当中，则不一定需要进行激励，因为参与记账的节点，往往是在链外完成了博弈，通过强制或自愿，来要求参与记账。

5）合约层 / Contract Layer

合约层主要包括各种脚本、代码、算法机制及智能合约，是区块链可编程的基础。将代码嵌入区块链或是令牌中，实现可以自定义的智能合约，并在达到某个确定的约束条件的情况下，无需经由第三方就能够自动执行，这是区块链去中心化、信任机器的基础。

关于合约方面，第一代区块链并不完善，如比特币本身只具有简单脚本的编写功能，只能进行交易，无法用于其他的领域或是进行其他的逻辑处理（当然，当初中本聪对比特币的定义仅仅是点对点的支付系统，并没有想让比特币成为一个操作系统）。而以以太坊为代表的第二代区块链则极大的强化了编程语言协议，实现了图灵完备，理论上可以实现任何功能的应用。若将比特币看成全球账本的话，以太坊可以看作是一台全球计算机，任何人都可以上传和执行任意的应用程序，并且程序的有效执行能得到保证。

6）应用层 / Application Layer

应用层是区块链的展示层，封装了区块链的各种应用场景和案例，类似于电脑操作系统上的应用程序、互联网浏览器上的门户网站、搜寻引擎、电子商城或是手机端上的 APP 等等。如目前搭建在 ETH、EOS、NEO 等公链上的各类区块链 DAPP 应用等，未来的可编程金融和可编程社会也将会是搭建在应用层上。

创新的共识机制

区块链中的核心理念即共识机制。参与者达成共识是整个 XT Chain 区块网络的核心，如果没有中央机构，区块链的参与者也需要对既定的条款达成一致。Edward Shils 的”共识理念”使共识的促成需以下条件：1) 团体成员共同接受法律、规则和规范。2) 团体成员一致认可实施这些法规的机构。3) 身份认同或团结意识，这样团体成员才会承认他们就达成的共识而言是平等的。

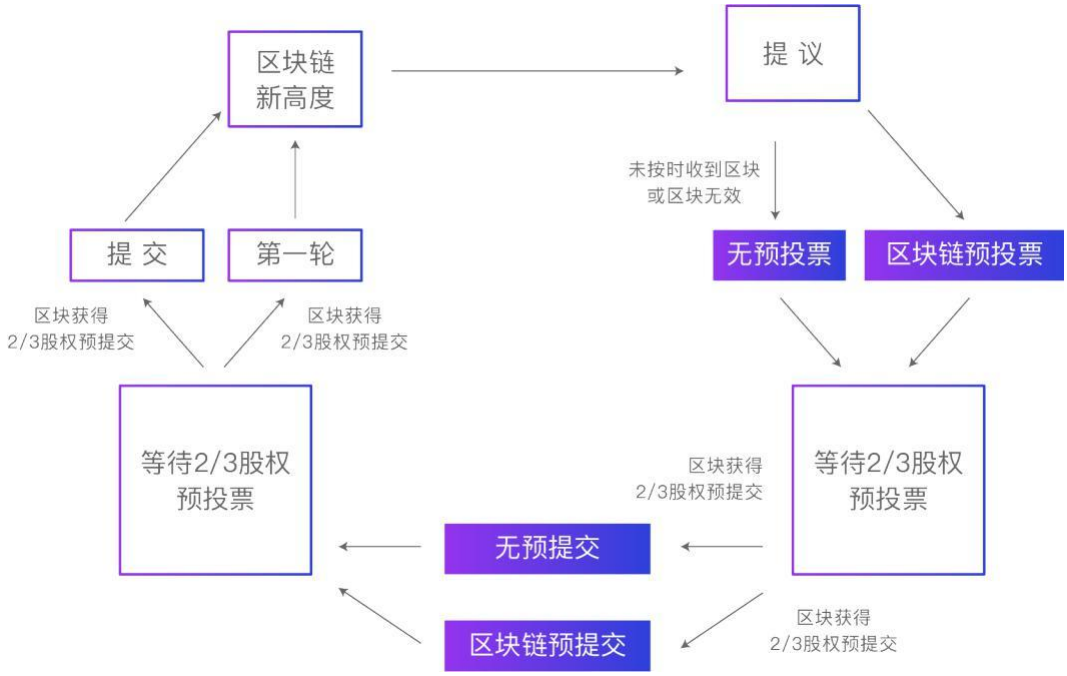
目前区块链行业中，代表性公链为 BTC 和 ETH，其交易确认时间长、吞吐量性能低下、严重依赖算力竞争的记账确认机制存在安全隐患。

XT Chain 共识模块算法基于 PBFT 算法并结合 DPOS 节点代表选举规则进行改进而成，实现了高性能共识算法 Paxos。在保证 BFT 系统强一致性的前提下，提升了系统的整体交易吞吐能力以及系统稳定性，可以稳定达到百万的 TPS，交易确认时间控制在 3s。

1）DPOS 共识机制

XT Chain 基于 DPOS 共识机制是对传统主流公链 BTC（POW 机制）、和 ETH（POS 机制）的改革，创新提出 Paxos 分布式一致性算法，在可信的加密货币网络中，提供事务处理和去中心化的共识协议的一种方法，目的是为了减少基于 POW 共识机制中的算力浪费和资源开销。

DPOS 共识机制和董事会投票表决有些类似。在一个去中心化系统中，将决策权力分发给所有持币者，而当持币者投票超过 51% 时，则认为该决定被通过，并且该决定不可逆。在该机制中有一个重要角色叫做代表，代表是生成区块的节点，想要成为代表首先要支付一定的保障金来保证代表的可信性。而用户则拥有选举代表的权利。每个用户可以投票选举一个值得信任的代表，在全网中获票最多的前 n 个代表则有生产区块的权利，这 n 个代表持有的票数相当于该节点持有的股数。这 n 个代表将按时间表进行轮流生产区块，生成的区块通过的股票数超过 51% 则认为区块生成成功，代表每生成一个区块将从区块中交易的手续费中获得收益，而这些收益也将是代表维持在线参与的一种奖励机制。



DPOS 共识机制中，代表的可靠性显得格外重要，持币者选取代表时，可以看到代表出块的错误率，从而可以正确的选择代表。另一方面，代表出块是按时间轮流产生，因此当一个代表错误的产生一个区块时，只要不得到 51% 的认可，那么该区块将会在下一个时间段由其他的代表生成。可以说 DPOS 共识机制相比与 POS 共识机制更加有效。

DPOS 共识机制是真正意义上摆脱挖矿的共识机制，然而该机制依赖于所有参与者的投票，当参与度不够的时候，那么代表往往会集中在全网中持有大量选票的持有人手中，从而失去了去中心化的特点，但是就另一方面来说第三方应用平台的使用规模以及使用频率与区块链的火热程度奠定了强大的用户群体，从而将这种风险降为最低。

2）PBFT 算法

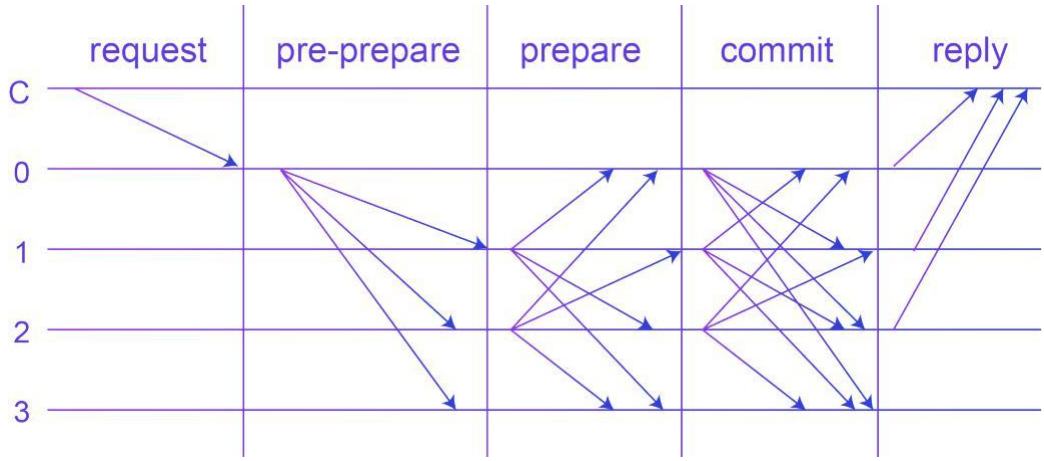
BFT（Byzantine Fault Tolerance），即拜占庭容错，是分布式计算领域的容错技术，拜占庭容错来源于拜占庭将军问题。PBFT（Practical Byzantine Fault Tolerance），即实用拜占庭容错算法，由 Miguel Castro 和 Barbara Liskov 在 1999 年发表的论文《Practical Byzantine Fault Tolerance and Proactive Recovery》中提出。

PBFT 算法可以工作在异步环境中，并且通过优化解决了原始拜占庭容错算法效率不高的问题，将算法复杂度由指数级降低到多项式级，使得拜占庭容错算法在实际系统应用中变得可行，目前已得到广泛应用。PBFT 算法可以在失效节点不超过总数 1/3 的情况下同时保证 Safety 和 Liveness。PBFT 算法采用密码学相关技术 (RSA 签名算法、消息验证编码和摘要) 确保消息传递过程无法被篡改和破坏。

PBFT 是一种状态机副本复制算法，即服务作为状态机进行建模，状态机在分布式系统的不同节点进行副本复制。每个状态机的副本都保存了服务的状态，同时也实现了服务的操作。将所有的副本组成的集合使用大写字母 R 表示，使用 0 到 |R|-1 的整数表示每一个副本。为了描述方便，通常假设故障节点数为 f 个，整个服务节点数为 |R|=3f+1 个， f 是有可能失效的副本的最大个数。尽管可以存在多于 3f+1 个副本，但额外的副本除了降低性能外不能提高可靠性。

所有的副本在一个被称为视图（View）的轮换过程中运作。在某个视图中，一个副本作为主节点（primary），其它的副本节点作为备份节点（backups）。视图是连续编号的整数。主节点由公式 p=v mod|R| 计算得到，v 是视图编号，p 是副本编号，|R| 是副本集合的个数。当主节点失效的时候就需要启动视图轮换过程。

PBFT 算法实现流程如下：



其中 C 为发送请求端，0123 为服务端，3 为宕机的服务端，具体步骤如下：

1.Request：请求端 C 发送请求到任意一节点，这里是 0

2.Pre-Prepare：服务端 0 收到 C 的请求后进行广播，扩散至 123

3.Prepare：123，收到后记录并再次广播，1->023，2->013，3 因为宕机无法广播

4.Commit：0123 节点在 Prepare 阶段，若收到超过一定数量的相同请求，则进入 Commit 阶段，广播 Commit 请求

5.Reply：0123 节点在 Commit 阶段，若收到超过一定数量的相同请求，则对 C 进行反馈

拜占庭容错能够容纳将近 1/3 的错误节点误差，IBM 创建的 Hyperledger 就是使用了该算法作为共识算法。

PBFT 算法具有高交易通量和吞吐量，高可用性，易于理解。同时也具备以下缺点：

A、计算效率依赖于参与协议的节点数量，由于每个副本节点都需要和其它节点进行 P2P 的共识同步，因此随着节点的增多，性能会下降的很快，但在较少节点的情况下可以有不错的性能，并且分叉的几率很低，不适用于节点数量过大的区块链，扩展性差。

B、系统节点是固定的，无法应对公有链的开放环境，只适用于联盟链或私有链环境。

C、PBFT 算法要求总节点数 n>=3f+1(其中，f 代表作恶节点数)。系统的失效节点数量不得超过全网节点的 1/3，容错率相对较低。

PBFT 算法的节点数量是固定的，节点身份提前确定，无法动态添加或删除，只能适用于节点数目固定的联盟链或私有链场景中。PBFT 在很多场景都有应用，在区块链场景中，一般适合于对强一致性有要求的私有链和联盟链场景，但如果能够结合 DPOS 节点代表选举规则，也可以应用于公有链，并且可以在一个不可信的网络里解决拜占庭容错问题。

3) Paxos 算法

根据 DPOS 和 PBFT 算法的原理和优缺点，XT Chain 制定了自己的 DPOS 节点代表选举规则，并解决区块链分布式一致性问题，形成 XT Chain 的 Paxos 算法。

Paxos 算法解决的问题正是分布式一致性问题，即一个分布式系统中的各个进程如何就某个值（决议）达成一致。

为了实现集群的高可用性，用户的数据往往要多重备份，多个副本虽然避免了单点故障，但同时也引入了新的挑战。

假设有一组服务器保存了用户的余额，初始是 100 块，现在用户提交了两个订单，一个订单是消费 10 元，一个订单是充值 50 元。由于网络错误和延迟等原因，导致一部分服务器只收到了第一个订单（余额更新为 90 元），一部分服务器只收到了第二个订单（余额更新为 150 元），还有一部分服务器两个订单都接收到了（余额更新为 140 元），这三

者无法就最终余额达成一致。这就是一致性问题。

一致性算法并不保证所有提出的值都是正确的（这可能是安全管理员的职责）。我们假设所有提交的值都是正确的，算法需要对到底该选哪个做出决策，并使决策的结果被所有参与者获悉。

Paxos 算法运行在允许宕机故障的异步系统中，不要求可靠的消息传递，可容忍消息丢失、延迟、乱序以及重复。它利用大多数 (Majority) 机制保证了 2F+1 的容错能力，即 2F+1 个节点的系统最多允许 F 个节点同时出现故障。一个或多个提议进程 (Proposer) 可以发起提案 (Proposal)，Paxos 算法使所有提案中的某一个提案，在所有进程中达成一致。系统中的多数派同时认可该提案，即达成了一致。最多只针对一个确定的提案达成一致。

Paxos 将系统中的角色分为提议者 (Proposer)，决策者 (Acceptor)，和最终决策学习者 (Learner):

Proposer: 提出提案 (Proposal)。Proposal 信息包括提案编号 (Proposal ID) 和提议的值 (Value)。

Acceptor：参与决策，回应 Proposers 的提案。收到 Proposal 后可以接受提案，若 Proposal 获得多数 Acceptors 的接受，则称该 Proposal 被批准。

Learner：不参与决策，从 Proposers/Acceptors 学习最新达成一致的提案（Value）。

Learner：不参与决策，从 Proposers/Acceptors 学习最新达成一致的提案（Value）。Paxos 算法通过一个决议分为两个阶段（Learn 阶段之前决议已经形成）：

. 第一阶段：Prepare 阶段。Proposer 向 Acceptors 发出 Prepare 请求，Acceptors 针对收到的 Prepare 请求进行 Promise 承诺。

. 第二阶段：Accept 阶段。Proposer 收到多数 Acceptors 承诺的 Promise 后，向 Acceptors 发出 Propose 请求，Acceptors 针对收到的 Propose 请求进行 Accept 处理。

. 第三阶段：Learn 阶段。Proposer 在收到多数 Acceptors 的 Accept 之后，标志着本次 Accept 成功，决议形成，将形成的决议发送给所有 Learners。Paxos 算法流程中的每条消息描述如下：

Prepare: Proposer 生成全局唯一且递增的 Proposal ID (可使用时间戳加 Server ID)，向所有 Acceptors 发送 Prepare 请求，这里无需携带提案内容，只携带 Proposal ID 即可。

Promise: Acceptors 收到 Prepare 请求后，做出“两个承诺，一个应答”。

两个承诺：

1. 不再接受 Proposal ID 小于等于（注意：这里是 <= ）当前请求的 Prepare 请求。

2. 不再接受 Proposal ID 小于（注意：这里是 < ）当前请求的 Propose 请求。一个应答：

不违背以前作出的承诺下，回复已经 Accept 过的提案中 Proposal ID 最大的那个提案的 Value 和 Proposal ID，没有则返回空值。

Propose: Proposer 收到多数 Acceptors 的 Promise 应答后，从应答中选择 Proposal ID 最大的提案的 Value，作为本次要发起的提案。如果所有应答的提案 Value 均为空值，则可以自己随意决定提案 Value。然后携带当前 Proposal ID，向所有向所有 Acceptors 发送 Propose 请求。Acceptors 发送 Propose 请求。

Accept: Acceptor 收到 Propose 请求后，在不违背自己之前作出的承诺下，接受并持久化当前 Proposal ID 和提案 Value。

Learn: Proposer 收到多数 Acceptors 的 Accept 后，决议形成，将形成的决议发送给所有 Learners。

智能合约

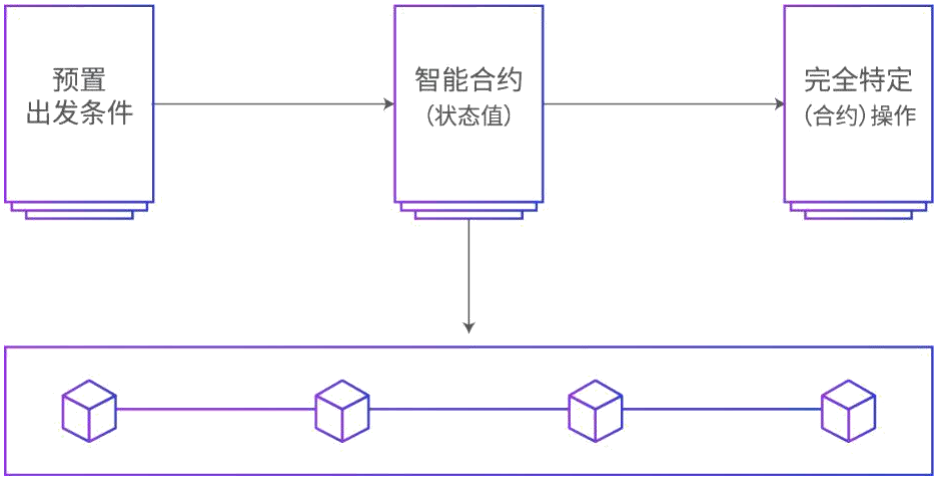
智能合约是一种只有通过区块链才能实现的新技术。普通、标准的合同涵盖了当事人之间协议的条款，且常通过法律来强制执行；智能合约是数字化的，存储在区块链中，并使用加密代码强制执行协议。换句话说，智能合约只是软件程序，与所有程序一样，它们完全按照程序员的意图执行。智能合约就像编程应用程序一样："一旦出现，就去执行。"基本上通过数学计算，智能合约可以协商协议中的条款，自动验证履行，甚至执行约定的条款，所有这些都不需要通过中央组织来批准。智能合约使公证人、代理人和律师等中间人几乎毫无意义。

智能合约的概念最初是由计算机科学家、密码学家 Nick Szabo 于 1993 年构思出来的。在 1994 年的一篇文章中，Nick 写道："智能合约的总体目标是满足共同的合同条件（例如付款项、留置权、保密性，甚至强制执行），最大限度地减少异常以及对可信中介的需求。相关的经济目标包括减少欺诈损失、仲裁和执行成本以及其他交易成本。现今存在的一些技术可以被视为粗略的智能合约，例如 POS 终端和（信用卡）、电子数据交换（EDI）以及公共网络带宽的 agoric 分配。

尽管智能合约在 2009 年比特币诞生时才出现一线生机，但以太坊完全接受了它，使得在其分布式账本中执行和存储智能合约成为可能。以太坊的平台专为执行智能合约而设计，使交易成为可能且无可挑剔。在许多方面，智能合约是所有区块链技术的基石。此外，许多新兴的区块链初创公司依赖于智能合约有望创造的革命。

就像有一个验证比特币交易的节点网络一样，智能合约也使用节点网络来验证协议的各个方面是否已经完成。他们不需要像律师这样的中间人来验证这些方面是否存在，这些节点和智能合约中的代码本身就可验证。这也使得智能合约透明且可被所有相关方追溯。因此，各方之间的信任不再具有争议。某些时候律师仍会被需要，但大部分工作都已完成。

最后，由于智能合约嵌在所有数据都以分散的分布式方式存储的区块链中，因此直到合同履行完成，没有人能够控制资金。这笔钱通常是区块链的本地加密货币，就像以太坊的以太币一样。



* 智能合约

XT ChainVM（(XT Chain Virtual Machine)）是智能合约的执行环境，为应用层提供智能控制逻辑，与上层高级语言解析转换相结合，灵活支持虚拟机基础应用；通过定制化的 API 操作实现虚拟机的外置接口，可以灵活的与账本数据和外部数据进行交互。

XT Chain 的 的 智 能 合 约 执 行 引 擎 XT ChainVM（XT Chain Virtual Machine）采用模块化可拔插的设计方式，首先开发的是支持 Java 语言的执行引擎 XT Chain JVM，后续会提供了支持 Solidity 语言的执行引擎 XT Chain EVM。 XT Chain JVM 是为了最大程度利用开源社区在智能合约技术和经验方面的积累，提

高智能合约的重用性而借鉴了以太坊的 EVM 的虚拟机。XT ChainVM 的智能合约实现完全兼容 Ethereum 的智能合约规范，使用 JAVA 作为智能合约的开发语言，通过微服务的架构设计以及多重安全检查机制为原生 Java 智能合约执行提供了一个高性能安全的执行沙盒。

XT Chain 实现了更彻底的去中心化的同时大大提高了 TPS。就拿现有的一代、二代、三代公链 TPS 做比较，BTC 的 TPS 仅仅只有 7 次 / 秒，ETH 的 TPS 有 30 次 / 秒， EOS 也只有 3900 次 / 秒，而 XT Chain 的 TPS 达到了惊人的百万次 / 秒，完全达到

了商业级别的应用，其共识机制也是最公平、最安全的，代码均为原创编写，而非一些公链的 Ctrl+V。

XT Chain 未来真正引入 Dapp。就目前来说，时下的主流公链存在难以扩展、缺乏互操作性等不足，在区块链上开发 Dapp 还需要自建模块。底层公链是实现区块链技术应用的基础，Dapp 如果发现某一底层公链具备更好的技术系统，入驻几率会大大提升。

据 XT Chain 团队技术负责人介绍，XT Chain 在 Dapp 上进行了大量技术投入，就是为了解决上述难题，为今后区块链系统的 Dapp 引入开辟一条崭新道路，在兼容时下主流公链合约（erc20 合约等）的同时，就主流开发语言 JAVA 有着大大的友好度。

过硬的底层技术加持。未来成熟的底层公链应该具备比较完善的技术特质，诸如：“创新的智能合约、“分层”、“分片”、“侧链”、“跨链”、“多链并行”、“去重加密”和“海量存储”等技术，XT Chain 将这些技术在自己身上一一变为了现实，开创了公链界的一个先河。

XT Chain 生态，极大提高了社区用户的积极性和持币能力，这为未来 XT Chain 推广商业应用提供社区能动性，同时极其简单的一键 Token 功能，相比以太坊价格昂贵和复杂的创建合约发币等所消耗的 gas 而言，XT Chain 有着极其低廉的燃料费。

XT Chain 落实商业应用。商业应用引进诸如：电商、支付、物流、游戏、工业机器人、信息溯源等上链，推动区块链技术向传统行业延伸，带动传统行业转型升级，从而享受到 XT Chain 带来的技术红利。

去中心化交易环境。XT Chain 建立的去中心化交易环境，让用户的资产去中心化托管，让所有交易都上链，无法篡改，公开、透明可查，此去中心化交易非交易所，而是商业应用所建立的通证，通过链上交换，完成交易，通过合同等方式完成商业通证的有效权，并展开一系列商业活动。

简单来说，XT Chain 相当于为区块链技术应用到各行各业提供了一个比现有公链网络更好的系统平台，试想，有了 XT Chain 这样更完善、先进的公链系统，其区块链技术应用到商业化前景将更加广阔，XT Chain 的持有价值不可估量。

可以说，整个区块链产业中，底层公链布局都是在刚刚起步阶段，像 XT Chain 这样以生态建设为重点的潜力公链来说，未来一切皆有可能。

柔性跨链机制

XT Chain 通过一系列有针对性的协同智能合约，及异步通讯、状态机和哈希锁定技术，实现一套通用的柔性跨链机制，打通各个区块链系统的通信瓶颈，让各种数字资产互联互通，适当的跨链协同机制有效保证内部各条并行链之间，以及与其他公链之间的共识和价值的有效和可靠传递。

跨链技术包括两个部分：一个是 XT Chain 与外部链的互联互通，XT Chain 与其他链通过一个公共的智能合约来实现，适配其他链的特征，基于状态机的异步操作，完成与其他链的交互。另外一个就是基于 XT Chain 平台的其他链之间的互通。

XT Chain 也提供一个更复杂的智能合约来支持其他链之间的互联互通，由于要支持两种不同类型的其他链，智能合约结合中继链完成不同类型链的互联互通。

跨链交易是区块链网络之间的去信任消息，这是一个关键的基础设施组件，用于链路间通信。跨链交易最初是在源块上创建的，然后在最终到达目标区块链之前通过桥梁和连接网络进行处理和转发。如前所述，跨链交易的创建者必须使用 XT Chain 作为通信支付交易费用，从而激励每个交叉点的参与者。

XT Chain 跨链通信通过适配器来实现，适配器会创建一个兼容的区块头。

XT Chain 设计了层级侧链机制来解决跨链交易匹配不同链区块生成速度的问题，根据链的区块生成速度把链划分到不同的层，然后为每一层提供一个专有的适配链或者适配模块来带动同层的跨链交易。

多链并行机制

经典的区块链网络，如比特币网络、以太坊等都是采用单链结构，所有的事务和交易都是在一条链上进行。单链结构的优点是交易和共识流程比较简单，在区块链发展早期能够很好地满足用户需求。但是随着区块链技术的发展和市场对区块链的需求不断增强，单链架构逐渐暴露出很多无法解决的痛点：

1）整体吞吐量和性能存在瓶颈：比特币只有 7 TPS 以及需要 6 个区块的确认机制，以太坊出块间隔也需要 10-20 秒，这些都严重阻碍了日益增长的区块链业务发展需求。

2）链内业务相互干扰：单链架构很容易由于个别业务的繁忙而造成整个系统拥堵，很多正常的交易都得不到及时处理和确认；封闭的网络结构：无法实现不同链之间的跨链交互，无法满足多平台之间的业务交互需求。为了克服单链结构的局限性，XT Chain 采取多链并行结构。

并行多主链机制：XT Chain 可以引出多条主链，每条主链负责专门的业务领域，相互独立又相互关联，主链之间耦合比较少，发挥并行处理的优势，对过程性区块，引入封存策略，超期数据进行历史归档，提升系统处理效率。多链并行解决不同业务、不同形态链的功能支持，同时提高性能；跨链共识实现数据审计和价值流通。

由于现实世界的不同业务具有各种各样的特殊性，如前所述单链结构是很难完美的支撑多种异构业务的。在 XT Chain 中，每一条链只服务于最小功能集合的业务，每个内聚型的业务运行在单独的链，这样既能做到有效的安全隔离，也能实现计算和资源的有效利用最大价值，不同链之间通过跨链协议进行交互，实现价值交换。

XT Chain 多链结构能满足现实世界各种不同类型的复杂业务需求，不同类型不同特性的业务在不同的子链运行，比如计算密集型、IO 密集型、混合型分别在不同链上良好地运行；不同安全等级要求的业务也可以在不同层次运行，比如针对银行的业务需求，在数据的保密和安全以及事务的强一致性会有更高的要求，因此可以隔离在最安全的一层。

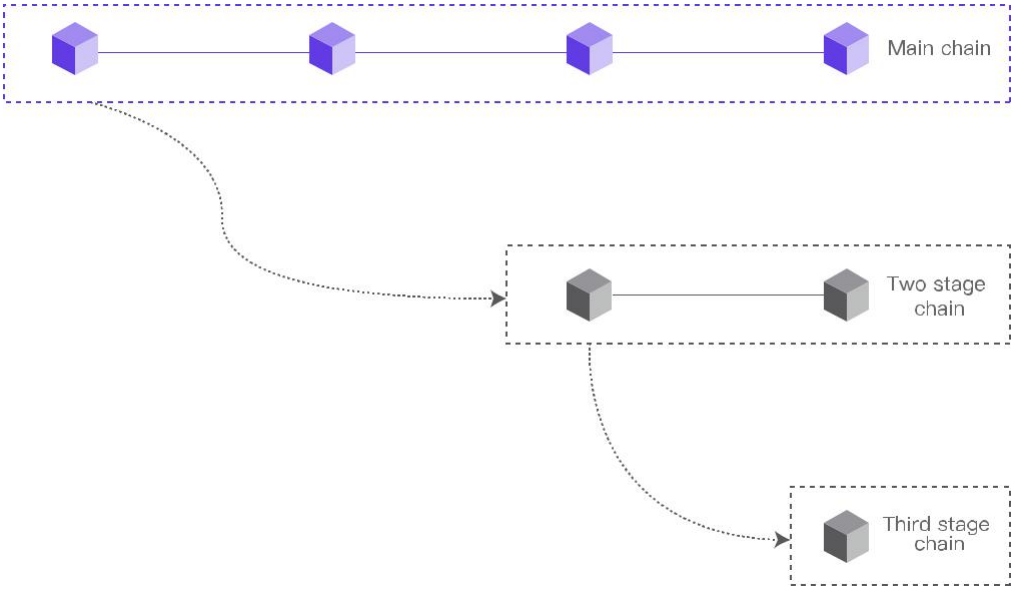
并行侧链方案

在 XT Chain 的生态系统中，被主链索引过的链都是侧链，每条侧链都被设计成只处理一种特殊类型的交易。当一条侧链需要验证从另一条侧链发来的信息时，必须包含 XT Chain 主链的区块头信息。

面对一些业务主链，其区块中的交易记录可按需引出侧链。XT Chain 引入侧链方案，各个侧链可以并行操作。即每个应用都可以独立开设一个侧链。XT Chain 区块链提供内置的、完善的、易用的侧链支持，侧链有多种共识算法模块供用户选择，侧链可以发行代币，主链和侧链可进行双向资产转移。所有侧链与主链共享算力，因此所有侧链都具有和主链一样的安全性。同时整体系统能耗可以实现最小化，避免分头挖矿带来巨大能源消耗和碳排放的问题。主链按区块链规则增长，主链区块中的记录的变动部分由侧链记录，实现区块信息固定部分与变动部分的有机结合。侧链记录的是主区块交易的附属数据，不影响其他交易信息，各个侧链可以并行操作，侧链的交易记录由智能合约或相关利益方签名确认即可。

在该侧链体系中，侧链同样可以拥有自己的侧链，但必须遵循从上到下的共识继承关系，子链需要继承全部上级链的共识，并同步全部上级链的消息，但同样也可以通过向主链或其他父级链的共识和系统来保障共享服务的性能、安全性和消息数据服务。在此基础上，次级链基于上级链的应用模型来开发自己独立的应用场景，并与上级链隔离。

XT Chain 侧链在系统设计原理上并不限于一层，而是可以建立多层次链。如下图所示：所谓的多层辅链结构，就是从侧链上再衍生出下一级侧链，上一层链称为父链，衍生出的链称为子链。



* 并行侧链

除了支持第三方能够比较方便的在 XT Chain 公链上构建侧链外，XT Chain 本身还会架构一些提供基础服务的侧链，比如 ID 服务、Token 发行服务、快速支付服务以及数字资产交易服务等，它们都是 XT Chain 基础设施的重要组成部分。

※ 安全体系

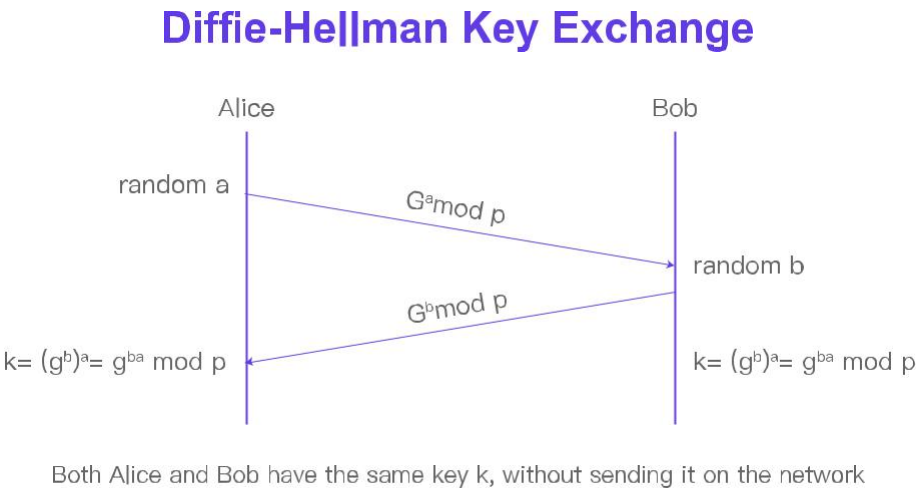
椭圆曲线 Diffie 一 Hellman 密钥交换

椭圆曲线密码学（英语：Elliptic curve cryptography，缩写为 ECC），一种建立公开密钥加密的算法，基于椭圆曲线数学。椭圆曲线在密码学中的使用是在 1985 年由 Neal Koblitz 和 Victor Miller 分别独立提出的。ECC 的主要优势是在某些情况下它比其他的方法使用更小的密钥——比如 RSA 加密算法——提供相当的或更高等级的安全。

ECC 的另一个优势是可以定义群之间的双线性映射，基于 Weil 对或是 Tate 对；双线性映射已经在密码学中发现了大量的应用，例如基于身份的加密。

椭圆曲线 Diffie-Hellman 密钥交换（Elliptic Curve Diffie–Hellman key Exchange，缩写为 ECDH），是一种匿名的密钥许可协议（Key-agreement protocol）。在这个协定下，双方通过 Diffie-Hellman 密钥交换算法，利用由椭圆曲线加密建立的公钥与私钥对，在一个不安全的通道中，建立起安全的共有加密资料。这是 Diffie-Hellman 密钥交换的变种，采用椭圆曲线加密来加强安全性。

XT Chain 利用椭圆曲线加密算法生成密钥对，密钥对包括一个私钥和由其衍生出的公钥。私钥用于发送数据时的数字签名，公钥用于验证数据的来源。数字签名保证了链上数据的一致性，防止数据被恶意篡改。



Both Alice and Bob have the same key k, without sending it on the network

私钥 k 是从 0 到 1.158*〖10〗^77(略小于 2^256) 之间随机选出的一个 256 位二进制数字，为了简便，一般用 64 位十六进制数字表示。公钥可以通过椭圆曲线算法从私钥计算得到：K=k*G，这是不可逆转的过程。G 是椭圆曲线算法中的生成点。

基于椭圆曲线加密算法生成的公私钥对的数据签名算法叫做椭圆曲线数字签名算法(Elliptic curve Digital Signature Algorithm)，其中签名密钥是用户的私钥，被签名的“信息”是设备上传或更新的数据，公式如下：

$$Sig=F_{sig}(F_{hash}m,dA)$$

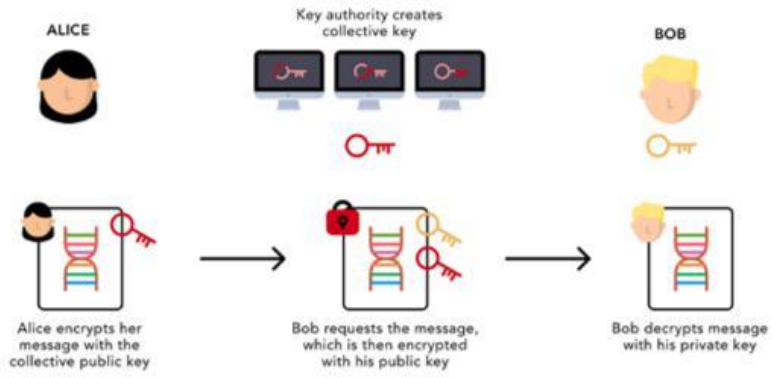
其中，dA 是签名的私钥，m 是被签名的信息，F_hash 是哈希函数，F_sig 是签名算法， Sig 是数字签名。

当用户向链上发送或者更新数据时，用自己的私钥对数据进行签名，并同时公布对应的公钥。其他节点根据发送或者更新的数据以及该设备公布的公钥对签名进行验证。只有拥有能产生这种公钥的私钥拥有者才能对要发送的信息生成特定签名。从而可以验证数据由某个设备发出，并且该设备无法篡改已发送的信息。

代理重加密

代理重加密（英语：Proxy Re-Encryption）是一种密文间的一种密钥转换机制，整个密文传送过程涉及三个主体：授权人、代理人、被授权人。授权人可以用自己的私钥解密用自己公钥加密的密文，但是无法将密文内容与被授权人分享，因为被授权人没有授权人的私钥。代理重加密解决了私钥拥有者（授权人）与其他人分享密文内容的问题。

代理重加密就是委托可信第三方或是半诚实代理商将自己公钥加密的密文转化为可用另一方私钥解开的密文从而实现密码共享。现实世界中绝大多数提供计算服务的公司信誉度较低，科学的解决办法是加密后放到云上，让其获得密文形式，而我们又想让我们愿意共享秘密的对方获得密文的明文内容，也就是原本我们公钥加密后的密文，只有我们的私钥才能解开，转化为对方私钥也能够解开。这个过程就是代理重加密。



授权人用自己的私钥、被授权人的公钥通过重加密密钥生成算法，生成重加密密钥，将这个密钥交给代理人。代理人接收到授权人的密文消息之后，通过重加密密钥对密文重加密，这个过程中代理人无法解密密文。然后代理人将重加密之后的密文发送给被授权人。被授权人用自己的私钥解密密文。整个代理重加密方案涉及以下几个算法：

密钥生成算法 ***KeyGen***（）：该算法为用户生成公钥 / 私钥对。

重加密密钥生成算法 ***ReKyGen***（$skey_i$,$pkey_j$,w,）：给定授权人的私钥 $skey_i$ 和被授权人的公钥 $pkey_j$，该算法可以生成重加密密钥 $rekyi_i \rightarrow j$

第一层加密算法 $Enc1(pkey_i, m)$：给定被授权人公钥 $pkey_j$ 和明文 m，该算法生成第一层密文 CT_j 。该层密文无法被进一步转换。

第二层加密算法 $Enc_2(pkey_i, m)$：给定授权人的公钥 pk_i 和明文 m，该算法可生成第二层密文 CT_i。该层密文可以通过重加密算法进一步转化为第一层密文。

重加密算法 $Re\ Enc(CT_1, rekey_{i \to j})$：给定授权人公钥 pki 和第二层密文 CT_j，该算法可以利用重加密密钥 $rekey_{i \to j}$ 生成第一层密文 CT_j 。

第一层解密算法 $Dec_1(CT_j, skey_j)$：给定发送给被授权人的第一层密文 CT_j 和被授权人私钥 $skey_i$ 该算法可以解密得到明文。

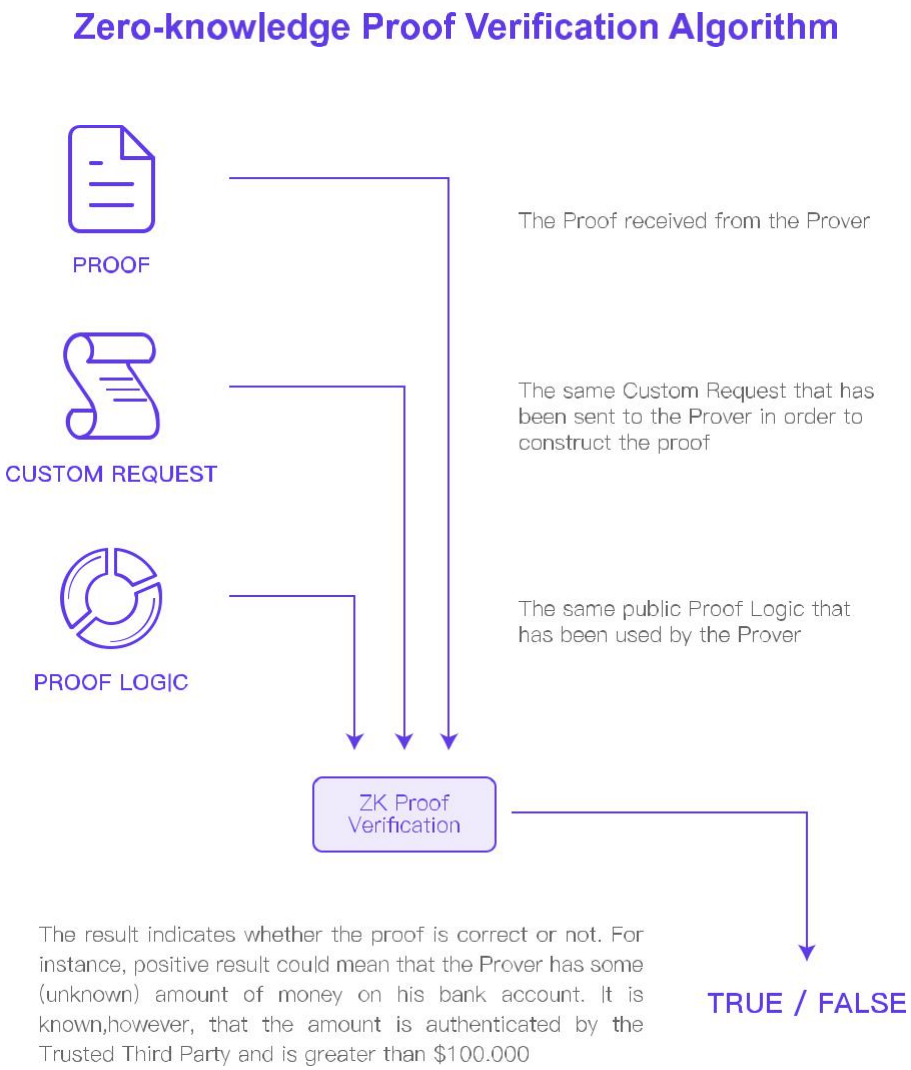
第二层解密算法 $Dec_2(CT_i, skey_i)$：给定针对授权人的第二层密文 CT_i 和授权人私钥 $skey_i$，该算法可以揭秘得到明文。

代理重加密方案的正确性要求对于任意条件 w ，任意明文 m，以及任意有效密钥对 $(pkey_i, skey_i)$，$(pkey_j; skey_j)$，均有如下等式成立：

$$Dec_2(Enc_2(pkey_i, m), skey_i) = m,$$
$$Dec_1(Enc_1(pkey_j, m), skey_j) = m,$$
$$Dec_1(ReEnc(Enc_2(pkey_i, m), ReEeyGen(skey_i, pkey_j))skey_j = m.$$

非交互式零知识证明

零知识证明（英语：Zero—Knowledge Proof），是由 S.Goldwasser、S.Micali 及 C.Rackoff 在 20 世纪 80 年代初提出的。它指的是证明者能够在不向验证者提供任何有用的信息的情况下，使验证者相信某个论断是正确的。零知识证明实质上是一种涉及两方或更多方的协议，即两方或更多方完成一项任务所需采取的一系列步骤。证明者向验证者证明并使其相信自己知道或拥有某一消息，但证明过程不能向验证者泄漏任何关于被证明消息的信息。



* 零知识证明算法

XT Chain 采用零知识证明机制，对用户数字资产进行确权。设 ***Hash*** 函数 ***H*** 满足随机预言机，其输出值长度为 ***n***，统计零知识的证明成立，资产确权。

证明：任给多项式时间的概率图灵机 TM^1， 其观察值为随机变量 (M, Σ')，M 其中为被签名的消息随机变量，Σ' 是签名随机变量，M , Σ' 且相互独立，(M, Σ') 诱导出随机变量 $(H(M), \Sigma')$，其中 H 为 ***Hash*** 函数，则：

$$\forall h,\sigma \ Pr\left((H(M),\Sigma')=(h,\sigma)\right)=\frac{1}{2^{2n}}$$

对于零知识证明，$\forall h$

$$Pr\left((H(M),\Sigma')=(h,\sigma)\right)=\begin{cases}\frac{1}{2^{2n}}, & Sign(h)=\sigma \\ 0, & else\end{cases}$$

$$\left| Pr\left((H(M),\Sigma)=(h,\sigma)\right)-Pr\left((H(M),\Sigma')=(h,\sigma)\right)\right| \le \frac{1}{2^{2n}}$$

因为 ***TM*** 是多项式时间的概率图灵机，因此在多项式时间内，

$$\sum_{|\{h,\sigma\}|\le n^c} \left| Pr\left((H(M),\Sigma)=(h,\sigma)\right)-Pr\left((H(M),\Sigma')=(h,\sigma)\right)\right| \le \frac{n^c}{2^{2n}}$$

证明成立，资产确权完成。

当资产进行交易，权属，移时，采用，交互零知识证明。即通过利用一，双方共享的公用随机串（比如第三方支付、第三方物流、保险等）来实现零知识证明，确定权属转移完成，资产重新确权，绑定收币方。

开源雾计算框架

*OpenFog 开源雾计算架构图

雾计算（Fog Computing）这个名字由美国纽约哥伦比亚大学的斯特尔佛教授起的，他当时的目的是利用“雾”来阻挡黑客入侵。后来思科首次正式提出，赋予雾计算的新含义。雾计算是一种面向物联网的分布式计算基础设施，可将计算能力和数据分析应用扩展至网络 “边缘”，它使客户能够在本地分析和管理数据，从而通过联接网络获得实时的见解。

在 2012 年由萨尔瓦多等人在一篇关于云数据安全的文章中提出，通过使用假信息做诱 饵，“钓”出窃密的“鼹鼠”，进而达到保护用户真实信息的目的与云计算将数据、数据处 理和应用程序全部保存在云中不同，雾计算将他们分散在网络边缘的设备中。

即在云服务器 和物联网（IOT）设备之间，利用网络设备（路由器、手机、开关、机顶盒、代理服务器等）或者专用设备提供计算、存储和网络通信服务，使得数据和计算更靠近终端设备，进而降低云服务器的计算和存储开销，并且提高了应用系统的回应速度和网络带宽。“雾计算”这一名称是因为相对云而言，雾更接近地面。雾计算没有强力的计算能力，因为提供算力的都是计算机周边和外围以及零散的计算设备。

雾计算（Fog Computing）主要使用的是边缘网络中的设备，数据传递具有极低延时。雾计算具有辽阔的地理分布，是具有大量网络节点的大规模传感器网络。雾计算移动性好，

手机和其他移动设备之间可以互相直接通信，信号不必到云端甚至基站去绕一圈，支持很高的移动性。

在 XT Chain 中，雾计算部分采用 OpenFog 开源部分，OpenFog 可以满足三个基本需求：低延时，维护用户隐私，多渠道链接。

IV

核心团队

Core Team

打造万物平衡的全领域生态公链

Create a Full-Field Ecological Public Chain that Balances Everything

IV 核心团队

Denis Abramov CEO

毕业于苏黎世联邦工业大学 (Eidgenössische Technische Hochschule Zürich) 管理经济学专业，曾任俄罗斯金融科技巨头公司 Diasoft 特许金融分析师。2014 年起深度参与过 ETH 及 EOS 等多个国际知名区块链项目的运营工作，具备传统金融及区块链金融的成熟运营管理经验。2018 年起担任 XT Chain 全球生态 CEO 一职。

Alexander Korolenko CTO

毕业于苏黎世联邦工业大学 (Eidgenössische Technische Hochschule Zürich) 计算机与信息科学专业，曾任俄罗斯金融科技巨头公司 Diasoft 信息安全工程师，2014 年与 Denis 共同进入区块链行业，曾参与过 ETH 及 EOS 项目开发工作，具备丰富的行业经验及专业的实践技术，2018 年再次携手 Denis 联合发布 XT Chain，并担任 CTO。

V

通证发行

Token Issuance

V 通证发行

※ 发行计划

·项目名称：XT Chain

·项目简称：XT

·通证发行数量：拟共发行通证总量恒定为 3300 万枚，永不增发。

·接收币种：XT，XT 是基于 XT Chain 技术发行的去中心化数字资产。

※ 代币分配

XT 是由 星图基金会发行的应用于公链的燃料消耗 Token，其分配比例如下：

2200 万 XT：创世区块释放，用于 Token 联动

1100 万 XT：搭建初始生态 DAPP 开发者奖励

VI

战略规划

Strategic Planning

打造万物平衡的全领域生态公链

Create a Full-Field Ecological Public Chain that Balances Everything

VI 战略规划

2019

- 11 项目立项

2020

- 01 去中心化公链生态方案制定
- 02 产品概念与逻辑初步构建
- 03 启动底层架构开发

2021

- 03 技术框架底层架构完成
- 04 移动端钱包、区块链浏览器开发完毕
- 05 进行主网测试
- 06 技术白皮书发布
- 07 代码开源
- 08 项目全面启动
- Q3 钱包APP上线
- Q4 登录全球主流数字资产交易平台

2022

- Q1 首批DAPP上线
- Q2 BILANX去中心化交易所上线
- 下半年 建设全领域生态，落地DEFI、跨境电商、支付、游戏、短视频、物流、信息溯源等多个领域

VII

风险提示

Risk Warning

VII 风险提示

该文档只用于传达信息之用途，并不构成买卖 XT Chain 股份或证券的相关意见。任何类似的提议或征价将在一个可信任的条款下并在可应用的证券法和其它相关法律允许下进行，以上信息或分析不构成投资决策或具体建议。本文档不构成任何关于证券形式的投资建议，投资意向或教唆投资。本文档不组成也不理解为提供任何买卖行为，或任何邀请买卖任何形式证券的行为，也不是任何形式上的合约或者承诺。XT Chain 明确表示相关意向用户明确了解 XT Chain 的风险，投资者一旦参与投资即表示了解并接受该项目风险，并愿意个人为此承担一切相应结果或后果。XT Chain 团队明确表示不承担任何参与 XT Chain 项目造成的直接或间接的损失，包括：

1、因为用户交易操作带来的经济损失；
2、由个人理解产生的任何错误、疏忽或者不准确信息；
3、个人交易各类区块链资产带来的损失及由此导致的任何行为。

XT 是基于 XT Chain 发行的加密货币。我们无法保证 XT 一定会增值，在某种情况下也有价值下降的可能，没有正确使用其 XT Chain 的人有可能失去使用 XT Chain 的权利，甚至会可能失去他们的 XT。XT Chain 不是一种所有权或控制权，控制 XT Chain 并不代表对 XT Chain 或 XT Chain 应用的所有权，XT Chain 并不授予任何个人任何参与、控制，或任何关于 XT Chain 及 XT Chain 应用决策的权利。

风险提示：许多数字资产交易所因为安全性问题而停止运营。我们非常重视安全，但世界上不存在绝对意义上的 100% 安全，例如：由于不可抗力导致的各种损失。我们承诺尽一切可能确保您的资产安全。

VIII

参考文献

Reference

VIII 参考文献

[1] Nick Szabo.Formalizing and securing relationships on public networks.First Monday,2(9),1997

[2] Back,"Hashcash-a denial of servicecounter-measure,"
http：//www.hashcash.org/papers/hashcash.pdf.2002

[3] Vitalik Buterin.Ethereum：A Next-Generation Smart Contract and Decentralized Application Platform.2013

[4] Gavin Wood.Ethereum：A Secure Decentralized Generalised Transaction Ledger.2018

[5] W.Dai,"b-money,"http：//www.weidai.com/bmoney.txt.1998

[6] AndreasAntonopoulos：MasteringBitcoin：Unlocking Digital Cryptocurrencies.2014

[7] Sheldon M.Ross.A First Course in Probability.2009

[8] Nash John."Non-Cooperative Games"The Annals of Mathematics.1951

[9] Schlegel,H.：Reputation Currencies.Institute of Customer Experience.

[10] Marko Vukolić：The Quest for Scalable Blockchain Fabric：Proof-of-Work vs.BFT Replication.2016

[11] Adam Back,Matt Corallo：Enabling blockchain innovations with pegged sidechains

[12] Vitalik Buterin：Zk-SNARKs：Under the Hood

[13] Eli Ben-Sasson：Zerocash：Decentralized Anonymous Payments from Bitcoin

[14] Petar Maymounkov：Kademlia：A Peer-to-Peer Information System Based on the XOR Metric

[15] Everett Hildenbrandt：VM：A Complete Semantics of the Ethereum Virtual Machine

[16] L.LamPoRt,Constructing digital signatures from a one-way function,Technical RePoRt SRI-CSL-98,SRI
International Computer Science Laboratory,Oct.1979.

[17] "Winternitz one-time signature scheme"
https：//gist.github.com/karlgluck/8412807#comment-1258433