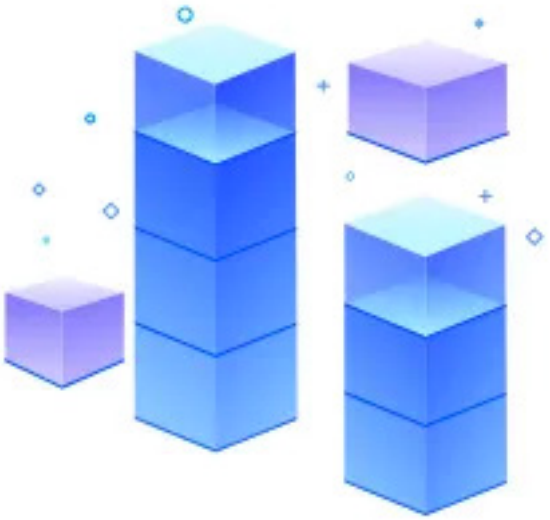
技术白皮书

NEW BLOCKCHAIN
PROGRAMMING PLATFORM

EKT 多链技术

目录



第一章 区块链背景介绍

什么是区块链

有些人因为其底层技术而对区块链感兴趣，另外一些人对它的商业可能性着迷，还有一些人关心它的社会和政治影响，理性的人可能不同意后面两类人的观点。区块链在技术上是有深度的、有趣的、和具有创新性的，而且是建立在坚实理论上的。我们开始开拓比特币、以太坊之外令人炫目的各种区块链项目，相信其中的某些公链也许有一天甚至超越比特币。区块链行业近期的主要发展趋势将会是底层链技术的创新，我们也一直在关注从技术上如何实现更大尺度的去中化。在工信部指导发布的《中国区块链技术和应用发展白皮书》里是这样解释区块链的：广义来讲，区块链技术是利用块链式数据结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、利用密码学的方式保证数据传输和访问的安全、利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算范式。区块链作为最颠覆性的创新技术其主要有以下几点特性：

特性一：去中心化

在传统的中心化网络中，对一个中心节点进行攻击就有可能破坏整个系统，而去中心化的网络采用分布式记录、分布式存储和点对点通信，任意点的权利和义务都是均等的，系统中的数据块由所有节点共同维护。这样就避免了被某个人或机构操纵，无论任一节点遭受攻击或停止工作，都不会影响整个系统的运行。当人们在讨论软件的去中心化时，实际上在讨论的，是三个独立的中心化或去中心化的轴。在没有另一个同在的情况下，很难看清现在的这个事物是中心化还是去中心化的。一般来说，中心化和去中心化互相独立，而这三个轴如下：

a. 架构上的（去）中心化—由多个计算机构成，在大 量计算机节点崩溃时，仍能保持运作；

b. 政治上的（去）中心化—非单个人和组织控制的去中心化政治体系，比如 AWS 就是一个典型由公司控制的非去中心化云计算服务平台；

c. 逻辑上的（去）中心化—系统呈现和维护的接口和数据库结构像是非结构群体，如果把这个系统的使用方和提供方一分为二，他们还能作为完全独立的单元保持运行显然在任何一个维度上，我们都无法定义究竟节点（决策者）小于多少就是“中心化”的，或者节点（决策者）大于多少，系统就是完善“去中心化”的。

特性二：去信任

在区块链系统中，节点之间无需任何信任也可以进行交易，因为整个系统的运作规则是公开透明的，所有的数据内容也是公开的。而在现实生活中，两个完全陌生、没有信任的人或机构要进行交易，需要依赖第三方权威机构（比如银行）作为信用背书，凭借对第三方机构的充分信任来完成交易。区块链所有节点都必须遵守同一交易规则来运作。这个规则是基于密码算法而不是信任，因此在系统指定的规则范围和时间范围内，节点之间是不能也无法欺骗其它节点，自然无需任何第三方介入。

特性三：不可篡改，加密安全

所有散列函数都有一个基本特性：如果两个散列值是不相同的（根据同一函数），那么这两个散列值的原始输入也是不相同的。这个特性是散列函数具有确定性的结果，具有这种性质的散列函数称为单向散列函数。当一份重要数据保管在单个人手中时，是很容易被篡改的，但如果同时有多个备份被不同的人保管的话，篡改的成本就增加了，而且如果每个人还为自己保管的数据加了密码，那被篡改的可能性将进一步降低。之所以说区块链中存储的数据“不可篡改，加密安全”，是因为区块链技术设计了一个机制，使得篡改的成本大大增加，这个机制就是哈希算法。散列函数能将任意原始数据，无论是图片还是音乐，对应到特定的数字，成为哈希值。只要有节点恶意篡改，哈希值就会发生变化，很容易被识别。所以一旦数据经过验证并添加至区块链被储存起来，除非能够同时控制住系统中超过 51% 的节点，否则单个节点上对数据库的修改是无效的，如果有节点想要垫付一个被确认的结果，其付出的代价将远高于收益，因此区块链的数据稳定性和可靠性极高。

公链是区块链发展的前提基础，也是区块链行业未来发展的核心保障。而目前区块链的发展现状是，底层公链的性能没有发展起来，在其上构建的各类 DApp 严重受限于性能 , 各种共识算法都有不完美之处。因此，我们预计 2018 年区块链行业的发展，仍以底层公链为重心，各公链在性能、可扩展性和应用性上将继续角力的局面。底层公链是一切的基础，使用网络编程、链式或 DAG 数据结构、加密算法、数据存储等技术来构建区块网络，通过共识机制和分配机制，实现节点网络的正常运行。

一些新型公链例如 EKT（本项目）提出了多链多共识的观点，也许将不同的工作链在处理交易的过程中承担着不同的角色，对于不同形式的账户地址、不同的交易形式、不同的智能合约虚拟机甚至不同的虚拟货币结算方式都有着不同的工作链对应处理，实现在不同的工作链中统一的交互标准，可以更加兼顾公链平台使用时的效率与公平性。多

链多共识的机制能为后来的区块链项目开发提供了很大的便利，可以适用于任何区块链的应用场景。

第二章 EKT 阐述

2.1 EKT 项目核心

EKT 是一条“多链多共识”的高性能公链，定位为采用 DBFT 共识机制的的 DApp 多链开发平台，开发者可以很容易开发出一个完整的 DApp 并应用于主链上。EKT 的多链架构为“多链多共识，一链一主币”，可通过跨链的报文协议跨公链的资产交换。

EKT 也是一个 DApp 的开发平台，致力于帮助互联网应用和实体经济实现 DApp 大规模落地。如果说 IPFS 的目标是一个去中心化的存储平台的话，那么 EKT 就是一个去中心化的计算平台。

在 EKT 中，我们坚持这样一个理念：一个健全稳定的货币系统中并不需要图灵完备的开发语言，基于此系统的不同应用间应该尽可能实现隔离。因此我们在设计公链构架体系的时候，把 Token 的处理和 DApp 的处理分开了，也就是说在 EKT 上存在两种类型的链：Token 链和 DApp 链。Token 主链是用户价值存储与交换的链，并且只用于用户

价值的存储与交换。Token 主链是一个多链多共识的架构，也就是说在 EKT 中，每一条链都可以拥有自己的共识机制，是一个并行主链的结构。

那么作为一个 DApp 开发者，最关心的肯定是三件事：1 开发难度 2 用户体验 3 社区生态。Token 链的设计就是针对不想自己重新开发一条公链，但想要发行 Token 和链的开发人员。EKT 主要面向的对象是开发者，所以在设计 Token 链的时候最大的想法是设计一个可以吸引大家都来构建的生态体系：一个基于高 TPS，低手续费，可以选择发链用自己的 Token 做交易费，还可以和其他的链共享用户的完善区块链平台，这样在 EKT 平台上开发 DApp 的开发者可以获得更多的用户资源。

EKT 的中心思想是设计一个社区的机制，让开发者可以轻易的开发一个 DApp，其他的交给 EKT 来处理，当然作为每一个开发者所头疼的安全问题也会是 EKT 在设计之初考虑的重点问题。Token 链是专门用于处理 Token 交易的一条链，鉴于 ERC20 代币不断曝出的各种漏洞（虽然漏洞的产生是智能合约开发者的问题，但是我们认为是有更好的方案来实现的），在 EKT 上内置了 Token 对象，开发者只需要定义自己要发的 Token 的数量即可。另外，EKT 的 Token 链是一个多链多共识的结构，也就是说不同的 Token 可以放在不同的 Token 链上进行打包，多链并行极大提高交易处理速度。

EKT 的 DApp 链是供不同开发者开发 DApp 的一条链。我们从智能合约开发语言、数据存储（带有默克尔证明的和私有的不带默克尔证明的存储空间）、效率三个方面进行了优化。EKT 的 DApp 链基本上可以实现与现在的互联网应用相同甚至更快的开发速度，可实现的功能性也与互联网应用没有太大差异，最重要的是，我们可以实现大部分事件的 1 秒执行和确认，安全性要求比较高的事件可以实现 3 秒的确认。

2.2 Token 链

Token 链是一个并行多链的结构，多链多共识，共享用户基础。

EKT 的 Token 链是专门用于处理 Token 交易的一条链。鉴于 ERC20 代币不断曝出

的各种漏洞（虽然漏洞的产生是智能合约开发者的问题，但是我们认为是有更好的方案来实现的），在 EKT 上内置了 Token 对象，开发者只需要定义自己要发的 Token 的数量即可。EKT 的 Token 链是一个多链多共识的结构，也就是说不同的 Token 可以放在不同的 Token 链上进行打包，多链并行极大提高交易处理速度。

之所以设计并行多链有两个原因：

a. 一个 Token 发行者或者链的发行者对共识的要求是不同的，对去中心化程度要求比较高的可以选择 POW，对 TPS 要求比较高的可以选择 DPOS，每个链的 Token 进行交易的时候消耗的是当前链的主币作为交易费，也提高了发行链的灵活性；

b. 并行多链可以共享用户，不同链的拥有者是一个互利关系，可以很方便的进行不同链的资产转移，而且对于多链并行理论上来说整个网络的 TPS 是没有上限的；

Token 链设计目标是针对没有开发能力想要发行 Token 和链的区块链创业团队。因为主要面向的对象是开发者，所以在设计 Token 链的时候最大的想法是设计一个可以吸引大家都来发币的平台（吸引点在于 TPS 高，手续费低，可以选择发链用自己的 Token 做交易费，还可以和其他的链共享用户），这样可以让在 EKT 上开发 DApp 的开发者可以获得更多的用户资源。

2.3 DApp 链

与 Token 链不同，DApp 链是一条独立运行的公有链，所有的智能合约与去中心化应用都在此链上运行。对于 Token 链来说，每一条 Token 链只记录转账交易事件，而对 Token 链处于无知觉状态。也即 DApp 链上的应用可以映射和操作 Token 链，执行转账或者查询账户数据等操作，而 Token 链无法得知 DApp 链的状态，也无从主动做出任何应用运行的动作。这样做的好处是使得钱包与应用分离。

作为目前应用最广泛的智能合约区块链产品以太坊，其采用的是转账带着应用跑的模式，即每一步应用操作都需要转账来核实，应用操作和真实转账纠缠在一起，使得每一次人气应用上线时都会因为操作人数过多而阻塞整个网络。而且因为智能合约与 Token

转账捆绑在一起，以太坊也经历了大大小小数次智能合约引起的丢币，刷币，甚至毁坏整个 Token 代币体系的事件。将 Token 链与 DApp 链分离后，由于 Token 链只有作为货币操作的基本属性，从逻辑上来说就不存在漏洞，而 DApp 链由于无需带着重资产运行，在编程和上线新项目时也无需再过于提心吊胆，项目更新和发展也将变得更加轻松和多样化。

如之前所提到的，在中小型区块链项目里，作为一个 DApp 开发者最关注的肯定是三件事：a. 开发难度　b. 用户体验　c. 社区生态。

Token 链解决了 c， DApp 链则能够解决 a 和 b，我们将就具体如何解决进行展开。

首先 EKT 将极大降低 DApp 开发难度：

a. EKT 提供了一个新的编程语言 AWM，是一个完全事件驱动的语言，大家只需要定义自己可以对外提供的事件和参数即可，不需要 main 函数，然后在自己定义的事件函数里面写上相应的逻辑处理即可；

b. EKT 会对主流的语言进行支持，第一个是 Java，之后将按照社区选举逐步部署 Nodejs 和 Python，让大家可以不用学习新的编程语言就可以写出自己的 Dapp；

c. 我们对 DApp 将完全屏蔽共识机制和区块的概念，开发者只需要关心自己制作功能的逻辑即可；

d. 对数据库和 fs 的支持，EKT 短期会支持 DApp 直接操作 SQL 数据库，也会支持分布式文件系统。

这样下来，在 EKT 上开发 DApp 其实和传统互联网应用的开发没有太大差异了。

最后介绍我们在最关键的部分——用户体验上做的努力，用户体验分为两方面：TPS 和延迟。对于区块链项目大家关注的重点都在 TPS，但其实在比特币或者以太坊上，即使 TPS 上去了，转账和操作的确认延迟也会成为一个巨大的问题。那么如何能建立一个正反馈的，良好用户体验的，良性运行的社区呢？ EKT 的思想是设计一个社区的机制，

让开发者可以轻易的开发一个 DApp，其他的交给 EKT 来处理。

在 EKT 中，允许一些执行顺序不同且对全局一致性没有影响的事件，在区块打包前执行（其实应用中大部分都是这样的事件），对时序性有要求的事件区块打包后执行，这样可以实现大部分事件的秒级确认和执行。其实就是把一些事件先异步执行，然后区块打包的时候进行一致性校验。这样就能较好的降低延迟了。

2.4 Bancor 协议

现今加密货币集中在中心化数字货币交易所汇兑交易，流动性高、人气高的币种可以发挥这种规模优势，让大规模的玩家参与投资。但这也对市值低，不活跃，缺乏流动性的小币种非常的不利，分散的交易对让他们经常无人问津或者有价无市。买卖双方想要在市场随时能够找到自己相对的买方和卖方，传统市场是需要刚好出现有对手方，才能进行成交。而数字资产的浪潮兴起，Token 时代的到来已经让这个缺口逐渐汇集了巨大的流量。

例如：Bob 在某中心化交易所上面挂了一个买单，意图购买 EOS，由于整个 EOS 盘资金流动规模巨大，用一个市价和不夸张的交易量，在深度足够的交易所上面能够很快实现成交。但如果 Bob 买的是比较偏门的，流通量小的币种呢？他挂一个买单，可能一连几天甚至数周都无法成交，因为供给和需求并没有达到足够的高度，导致流动性低到不能支撑足够的买卖。

由于“长尾效应”，前 10% 的持有者占了整个市场 90% 的加密货币，以及 95% 的交易量。这种情况下的“长尾巴”，由于交易机制不够方便有效，流动性极为匮乏。从人们需求的角度来看，大多数的需求会集中在头部，而这部分我们可以称之为主体市场，而分布在尾部的交易需求是个性化的，零散的小量需求。这部分差异化的、少量的需求会在需求曲线上面形成一条长长的“尾巴”，所谓长尾效应就在于其数量上，如果将所有非流行的市场累加起来就会形成一个比流行市场还大的市场。

由此，Bancor 协议的运用有了足够的缘由。它通过代码程序来管理交易，其中的算

法计算率，自动的让市场流通性保持在一个良性的状态，这样就算是流动性很差、交易深度很小的币种也可以维持流通。简单来说，Bancor 即是一种异步价格机制，通过供需关系来调节价格。在这套模型中，存在准备金池 r（或连接器池 connector balance）、杠杆率（或 CW connector weight 或连接器比重）以及智能代币供给量 s（supply 或中转代币）三个重要参数。根据 p*s*CW=r 公式，可以确定智能代币的价格 p(price)。另外，储备金增量与智能代币增量存在一个对应的数学关系。

EKT 计划采用 Bancor 协议来解决多 Token 链之间的交换难题。即通过 Bancor 算法，在储备金池中填充一篮子其他类 Token，采用自己发行 EKT 作为连接代币，从而可以实现类似于交易的货币兑换效果。EKT 作为连接器，可以通过 Bancor 协议兑换所有 Token 链上的代币，更重要的是，开发者需要通过 EKT 来获得运行 DApp 所需的 gas。在上线 DApp 时，gas 由开发者充值到 DApp 合约地址， 每次 DApp 调用都会扣除 gas 币，扣除完之后的 gas 币将回到到储备池，当委托节点挖到矿时，委托人节点拿到对应 gas，从而形成一个闭环。

2.5 异步事件与同步事件

在 Token 链与 DApp 链分离的系统中，我们很容易就能想到一个关键的问题——同步问题：DApp 的操作是否和 Token 链状态更新同步。

EKT 提供两种类型的“事件”，作为 DApp 的开发者可以选择让自己的应用与 Token 链保持实时通信和状态更新，每一个参与者都把自己的每一步操作同步到公链上，以此来保障每一步操作的公开可信，我们把这种状态称之为“同步事件”。另一类对于实时公开结果或者运行环境不像比特币一样的假设参与者都是极端无信的情景下，EKT 同样提供了“异步事件”，也即一个应用参与者可以记录下所有截至上一次与公链同步时至今的所有操作，等到一个事件接近尾声或者一个生命周期结束前再同步和公示。

以德州扑克为例，Alice，Bob，Sam 三个人都互不相信对方时，可以在每一次翻牌

前都申请将当前状态同步到 DApp 主链，每一次翻牌都重新掷骰，在每一局结束时都实时更新和操作锁定在此应用里的账户余额。这也是区块链世界里常见的思维方式。也是 EKT 里同步事件的运作方式。

而以斗地主游戏为例，Alice，Bob，Sam 三人即使都互不信任，可以在每一次开局前由每一个人掷出一个随机数并保存在本地，这样每一次出牌都基于这个随机数的或者某种组合方式之后的计算默克尔子节点，记录在外部矿工节点，在一局结束时，再由这个矿工节点将牌局结果和大家各自的随机数上传到公有链。这样做的好处是，不用每一步都和主链做通信，对网络环境要求非常低，应用运行体验大大增加，而对于公链的资源占用压力也会大大减少。我们相信，很多应用场景下，其实并不需要像比特币或者以太坊那样预设所有参与者互相不信任，或者即使互相不信任，也可以在不用实时同步事件状态的情况下仍然通过这种机制保持结果可信和过程可倒推。

2.6 EKT 区块链设计

EKT 的共识算法是基于路由策略进行拜占庭容错的方案，在 DBFT 的共识基础上，我们未来也计划引入 POS/DPOS 和 POW 方案。在 EKT 中，我们使用公私钥加密和路由策略的机制实现拜占庭容错。

拜占庭将军问题首先是由 Leslie Lamport 等人在 1982 年提出，被称为 The Byzantine Generals Problem 或者 Byzantine Failure。这个问题是这样描述的：

拜占庭帝国想要进攻一个强大的敌国，为此帝国派出了 10 支军队去包围这个帝国。这个敌人虽然不如拜占庭帝国强大，但也足以抵御 5 支常规拜占庭军队的同时袭击。由于某些原因，这 10 支军队无法聚合在一起进行攻击，必须分散然后根据统一的指令一起进攻或者撤退。他们任一支军队单独进攻都毫无胜算，除非有至少 6 支军队同时袭击才能攻下敌国。他们分散在敌国的四周，依靠通信兵相互通信来协商进攻意向及进攻时间。军中可能有叛徒，可能向其他的将军发送错误的指令。在这种情况下如何保持战争指令的统一性进而获取胜利便成为了一个问题。

进一步讲，拜占庭将军的问题可以描述为：

一个发送命令的将军要发送一个命令给其余 n−1 个将军，使得所有忠诚的接收命令的将军遵守相同的命令如果发送命令的将军是忠诚的，那么所有忠诚的接收命令的将军遵守所接收的命令这个问题发展到计算机领域，就是拜占庭容错问题。区块链需要解决的一个核心问题就是如何保证在分布式环境下，各个节点（即使存在恶意节点）的数据能够达成最终的一致性和正确性

EKT 主链上每个路由节点的公钥都是公开的，具体路由策略为：

1. 区块广播

当一个节点完成打包之后，会对区块进行签名。签名完以后节点会把区块和签名广播给网络中的其他节点。当另外一个节点收到区块和签名之后会对签名信息进行校验，以此来确认这个区块是从打包节点广播出去的。其他节点确认完成后，会判断自己节点与打包节点在当前轮的距离，如果满足条件（currentIndex − miningIndex + len(DPoSNodes)) % len(DPoSNodes) < len(DPoSNodes) / 2，则将自己收到的区块和签名继续广播给其他节点。 当一个节点收到两个不同的打包节点的区块和签名之后，会将两个不同的区块和签名发送给所有其他节点。而所有节点则放弃当前区块，进入下一个区块的打包并对当前打包节点的作恶行为进行记录。

2. 区块的校验与投票

在每个区块头上，都会有区块 body 的 Hash 校验值。节点可以向其他节点获取区块 body，对 body 进行处理之后，对当前打包的区块进行投票，所有节点都会把区块的校验结果进行签名，发送给满足（currentIndex − miningIndex + len(DPoSNodes)) % len(DPoSNodes) < len(DPoSNodes) / 2 条件的节点进行唱票。 当任何一个节点收到超过半数对同一个区块的投票之后即可认为当前的区块可写入区块链中，并将区块和投票结果发送给所有的节点，所有节点对区块进行记录。如果投票的数量不足半数则在一定时间内停止唱票，节点将自己的唱票结果发送给其他节点，所有节点在收到其他节点的投票结果之后对结果进行合并，判断最后的投票结果并执行响应的操作。

3. 节点宕机

当一个节点超过一定时间没有出块，当前轮的下一个节点会在 3*interval/2 的时间点开始打包下一个区块，进入下一个区块的打包流程。同理，如果节点连续宕机，判断当前节点是否需要打包的条件是 currentTime – lastBlockTime > (2*(currentIndex – LastIndex)+1)*interval/2，一旦满足当前条件，则当前节点开始打包。如果是最后 n 个区块连续宕机，则按照当前轮的最后一个区块的 Hash 值判断下一轮的顺序，按照递增每个区块加一个出块 interval 的算法进行计算，判断当前打包的节点并进行打包。当超过 n/2 的节点宕机的时候，所有节点会自动停止出块，直到超过 1/2 的节点存活。

这种方案的复杂度在最好情况下是：消息复杂度 O(n^2)，时间复杂度 O(1)。在最差情况也可以达到：消息复杂度 O(n^2)，时间复杂度 O(n)。基于这种路由策略的拜占庭容错机制，系统可以保证在少于 n/2 的节点宕机或者叛变的情况下，系统不会出现分叉，是一种用计算资源换容错性的方案。

2.7 EKT 区块链的优势

效用美学

EKT 生态是一个并行多主链的结构。其中最重要的主链是 EKT 主链，它是维持整个系统生态运转的核心，EKT 主链除了承担交易记录、用户体系以及跨公链资产交换的功能以外，还记录了生态中所有其他链的信息，基于 EKT 主链可以实现天然的跨链资产交换。

在 EKT 多链生态里，其他主链的底层代码和 EKT 主链几乎是一致的，不同的在于其他主链的代币情况和共识算法的区别。其他主链在 fork 了 EKT 主链的代码以后，可以部署到自己的节点独立运行，这种情况下该主链相当于一条独立的区块链，和主链的生态没有产生直接的关联。

如果一条基于 EKT 代码运行的主链想要加入到 EKT 多链生态的话，他该如何操作呢？EKT 多链会提供一个统一的客户端。这个客户端可以看做是其他主链和 EKT 主链进行连

接的入口。其他主链可以在这个客户端里提交自己的主链信息，包括部署节点、代币名称、代币数量以及共识算法等信息。在向 EKT 主链注册成功以后，这条主链就可以共享 EKT 多链技术生态的资源了。

每一条基于 EKT 代码运行的主链都需要有自己的代币，即所谓的“一链一主币”。主币可以充当该链上的交易手续费。另外由于 DApp 开发也可以基于主链，是主链的上层应用。主币的功能及消费都可以在 DApp 里进行定义。

安全优势

公链是区块链发展的前提基础，也是区块链行业未来发展的核心保障。而目前区块链的发展现状是，底层公链的性能尚未发展起来，在其上构建的各类 DApp 严重受限于性能，各种共识算法都有不完美之处，安全问题也令人堪忧。由于智能合约一旦上传，即公开且不可更改，因此大多“区块链 2.0”项目有安全性验证的需求。一些团体也开始致力于应用形式化验证技术，为智能合约和区块链生态提供安全保护。将智能合约转化为数学模型，通过逻辑上的推理演算来验证模型，从而证明智能合约的安全性。

但由于智能合约是“不可变更的”。一旦部署，它们的代码是不能更改的，导致无法修复任何发现的 bug。在潜在的未来里，整个组织都由智能合约代码管控，对于适当的安全性需求巨大。过去的黑客如 TheDAO 等漏洞事件提高了开发者们的警惕，可见我们还有很长的路要走。

各个公链在可扩展性，应用性，共识哲学以及安全建设上的角逐将持续很长一段时间。我们认为既然没有完美的防止漏洞的方法，就把 Token 链和 DApp 链分开，让 Token 这个“对象”尽可能的简洁，是现阶段区块链行业里比较好的解决方案。

无限扩容

多链，即抛弃了“一链治所有”的传统方案，采用“多链分而治之”的新方案重新设计了

一个保障每个合约都能正常运行的公链。这一创新极大程度上简化了架构，降低了数据处理压力，确保一条链上流量激增不会影响到另一条链的效率，在链上进行的任何业务都不会收到其他业务干扰，有效实现了资源隔离。

区块链的互操作性本身就是一些应用的基础需求。想象一个理财应用，用户可以用某项资产交换不同机构的理财产品，不同的资产就需要在多条链上做转移、交换。还有一些 ORACLE 应用同样需要多链间的跨链喂入交互，譬如汇率牌价、天气、股价、特定指标等等。区块链的某些应用在单链上无法完整实现，需要在多链架构下的可扩展性、隔离性、高性能、互操作等特性的帮助下实现。

在 EKT 中 Token 链是一个并行多链的结构，多链多共识，共享用户基础。EKT 的 Token 是链上的一个属性，就像使用了 UTXO 模型的链 UTXO 有其他 Token 一样，我们的转账事件也是内置的。EKT 提供了一套底层的区块链机制，其他的区块链项目可以很容易的基于 EKT 的主链代码部署一套自己的主链。EKT 开发者可以轻易的开发一个 DApp，其他的交给 EKT 来处理， EKT 的“一链一主币，多链多共识”的机制为后来的区块链项目开发提供了很大的便利，可以使用于任何区块链适用的应用场景，而互不影响的 DApp 多链系统，可以一直横向扩展，也就带来了事实上的无限扩容。

2.8 EKT 区块链的应用

随着大批区块链项目的涌现和衰落，区块链的应用成为了项目是否能够长期发展的关键。而一项新技术能否最终落地，最关键的一个要素仍然是适合的应用场景。区块链技术应用场景应该遵循四大原则：

第一，多信任主体：区块链是信任机器，应用环境最好是相互之间没有天然信任关系，需要通过区块链来搭建信任。反之，如果双方是强信任关系，或已有完善的制度保障，使用区块链的必要性就不大。

第二，多方协作：如果该场景协作方多，对账成本高，区块链底层的共享账本之上搭

建的智能合约能够降低对账成本，从而提升效率。

第三，中低频交易：区块链目前的并发性和扩展性还不足以应用于大规模高频交易，比如股票交易所。

第四，商业逻辑完备：区块链节点之间一定要有完备的商业逻辑，形成多赢局面，参与者才有动力使用整条区块链。

在多链多共识的机制下，EKT 区块链将充分利用跨链技术的优势，对多条链的应用场景进行整合并统一进行管理，形成多共识应用场景生态。此项技术将在各大平台领域都能得到充分的应用和落地。EKT 区块链的多链结构能够满足应用场景的使用多样性，这增加了多功能平台的多重整合的可能，同时，多链结构和跨链价值交换又能保持整个生态系统的良性运转。在多个代币形成的架构中，主链将维持着所有代币的平衡和使用，使得所有代币都能正常的运转，带动子链生态的和谐发展。

第三章 技术架构

作为数字加密货币体系的核心支撑技术，区块链技术的发展大致经历了三个阶段，分别是：

- 支持比特币等数字货币的区块链 1.0 阶段；

- 用智能合约实现对数字货币外多应用场景支持的区块链 2.0 阶段；
- 从最近开始流行的 ” 区块链 +” 则是其 3.0 阶段，本项目即采用多链体系的 DApp 区块链

3.1 区块链核心能力

从技术角度来看，EKT 主链并不是一个全新的技术，而是结合了多种现有技术进行的组合式创新。典型的技术构成包括共识算法、P2P 通讯、密码学、数据库技术和虚拟机，这也构成了区块链不可或缺的 5 项核心能力：

- 共有数据（源自共识算法）：参与区块链的各个主体通过合约的决策机制自动达成共识，分享同一份可信的数据账本；分布式（源自 P2P 通讯），实现点对点信息传输；
- 数据的隐私性与安全性（源自密码学），通过公私钥、哈希算法等密码学工具，实现各主体身份和共有信息的安全；
- 数据存储（源自数据库技术和硬件存储计算的发展）：随着时间的累积，区块链的大小也在持续上升，硬件的存储计算能力使得多主体间同时存储相同数据成为可能；
- 数字化合约 /DApp（源自虚拟机技术）：将生成的跨主体的数字化智能合约写入区块链系统，通过预设的出发条件，驱动执行。

3.2 产品模型及设计原则

比特币为人称道的一个设计上的亮点就是它的脚本引擎。基于这套脚本引擎，不但可以实现普通的转账功能，还可以实现多方签名、抵押担保等智能合约应用。但是出于安全和实现难度的考虑，比特币的脚本系统设计的较为简陋，做了非常多的限制，比如它不支持循环、脚本长度受限、只支持几种标准的的交易类型。以太坊的最大特色就是极大地扩展了这个脚本引擎的功能，加入了读取区块链、计费、跳转等新指令，还解除了栈内存、函数调用深度以及脚本长度限制等。以太坊自称他们的脚本语言达到了图灵完备，

利用这样的脚本，开发者可以实现几乎任何可以用数学方式表述的功能。

自以太坊以来，扩展脚本成为了一种实现去中心化开发平台的流行方式，但这种方式有一个很大的缺点就是，应用代码本身及应用产生的数据都存在同一个区块链中，造成了区块链的快速膨胀。以太坊试图通过优化和压缩区块和交易本身来延缓这种膨胀，也只是一种治标不治本的方法。此外，基于脚本实现的应用之间是共享同一个账本的，像区块产生时间等参数是无法被定制的，这无疑限制了应用的个性化。

EKT 区块链设计将遵循以下原则：

- 简洁原则：协议将尽可能简单，即便以某些数据存储和时间上的低效为代价。一个普通的程序员也能够完美地去实现完整的开发说明。这将最终有助于降低任何特殊个人或精英团体可能对协议的影响，并且推进 EKT 区块链作为对所有人开放的协议的应用前景。添加复杂性的优化将不会被接受，除非它们提供了非常根本性的益处。
- 通用原则：没有“特性”是 EKT 设计哲学中的一个根本性部分。取而代之的是，EKT 提供了一个内部的图灵完备的脚本语言以供用户来构建任何可以精确定义的智能合约或交易类型。
- 模块化原则：EKT 的不同部分应被设计为尽可能模块化的和可分的。开发过程中，应该能够容易地让在协议某处做一个小改动的同时应用层却可以不加改动地继续正常运行。类似“短剑 ”(Dagger),“帕特里夏树 ”(Patricia trees) 和 “递归长度前缀编码 ”(RLP, recursive length prefix encoding,) 等创新应该以独立的库的形式实施并且应该特性完整，以便于让其它的协议同样使用，即便 EKT 不需要其中的某些特性 。EKT 开发应该最大地做好这些事以助益于整个加密货币生态系统，而不仅是自身。
- 无歧视原则：协议不应主动地试图限制或阻碍特定的类目或用法，协议中的所有监管机制都应被设计为直接监管危害，不应试图反对特定的不受欢迎的应用。你甚至可以在 EKT 主链之上运行一个无限循环脚本，只要你愿意为其支付按计算步骤计算的交易

费用。

3.3 共识机制—DBFT

EKT 的模块化设计，支持共识机制在内的所有核心功能模块的替换与插拔。EKT 主链默认采用 DBFT 共识机制，任何人都可以参与到区块链网络，每一台设备都能作为一个节点，每个节点都允许获得一份完整的数据库拷贝。节点间基于一套共识机制，通过竞争计算共同维护整个区块链。任一节点失效，其余节点仍能正常工作。共识机制是区块链技术的一个核心问题，它决定了区块链中区块的生成法则，保证了各节点的诚实性、账本的容错性和系统的稳健性。基于区块链技术的不同应用场景，以及各种共识机制的特性，主要可以从性能效率、资源消耗、容错性、监管水平等几个方面进行评价和比较。EKT 共识机制功能组件具备以下功能：

- 支持多个节点参与共识和确认；
- 支持独立节点对区块链网络提交的相关信息进行有效性验证， 防止任何独立的共识节点未经其他共识节点确认而在区块链系统中进行信息记录或修改；
- 具备一定的容错性，包括节点物理或网络故障的非恶意错误，以及节点遭受非法控制的恶意错误，以及节点产生不确定行为的不可控错误；
- 只需要一个区块即可实现确认 , 极大减少确认时间。

在 DBFT 共识算法中，区块链的正常运转依赖于受托人 (Delegates)，这些受托人是完全等价的。受托人的职责主要有：

1. 提供一台服务器节点，保证节点的正常运行；
2. 节点服务器收集网络里的交易；
3. 节点验证交易，把交易打包到区块；
4. 节点广播区块，其他节点验证后把区块添加到自己的数据库；
5. 带领并促进区块链项目的发展。

受托人的节点服务器相当于比特币网络里的矿机，在完成本职工作的同时可以领取区块奖励和交易的手续费。

一个区块链项目的受托人个数由项目发起方决定，一般是 101 个受托人。任何一个持币用户都可以参与到投票和竞选受托人这两个过程中。用户可以随时投票、撤票，每个用户投票的权重和自己的持币量成正比。投票和撤票可以随时进行，在每一轮 (round) 选举结束后，得票率最高的 101（一般为 101，也可以是其他数字，具体由区块链项目方决定）个用户则成为该项目的受托人，负责打包区块、维持系统的运转并获得相应的奖励。

选举的根本目的，是通过每个人的投票选举出社区里对项目发展和运行最有利的 101 个用户。这 101 个用户的服务器节点既可以高效维护系统的运转，而他们也会贡献自己的能力促进区块链项目的发展，这有点类似于“人民代表”制度（但是周期更短、效率更高）。通过这种方式，既达到了去中心化的选举共识，又保证了整个系统的运行效率和减少能源浪费。

3.4 智能合约

智能合约是编程在区块链上的汇编语言。EKT 打造的是非图灵完备的主链和图灵完备的侧链相结合的智能合约平台，提供了一个新的编程语言 AWM——完全事件驱动的语言，大家只需要定义自己可以对外提供的事件和参数即可，不需要 main 函数，然后在自己定义的事件函数里面写上相应的逻辑处理即可。通常人们不会自己写字节码，但是会从更高级的语言来编译它，例如用 Java，之后将按照社区选举逐步部署 Nodejs 和 Python 类似的通用语言。这些通用编码语言确实给区块链的功能性提供了指引，因此代码可以很容易与它进行交互，例如转移密码学货币和记录事件。代码的执行是自动的：要么成功执行，或者所有的状态变化都撤消。这是很重要的，因为它避免了合约部分执行的情况。在区块链环境中，这尤为重要，因为没有办法来撤消执行错误所带来的不好的后果（而且如果对手不配合的话，根本就没有办法逆转交易）。

基于区块链的智能合约不仅能发挥智能合约低成本高效率的优势，而且可以避免恶意行为对合约的正常执行的干扰。将智能合约以代码化的形式写入区块链中，利用区块链技术实现数据存储、读取及执行过程可追踪透明化且不可篡改。此外利用区块链的共识算法构造的状态机系统能使智能合约高效的运行。

智能合约的功能组件：

A 开发运行环境，包括：

(1) 提供编程语言支持，必要时可提供配套的集成开发环境；

(2) 支持合约内容静态和动态检查；

(3) 提供运行载体支持，如虚拟机等；

(4) 对于与区块链系统外部数据进行交互的智能合约，外部数据源的影响范围应仅限于智能合约范围内，不应影响区块链系统的整体运行。

B 存储环境，包括：

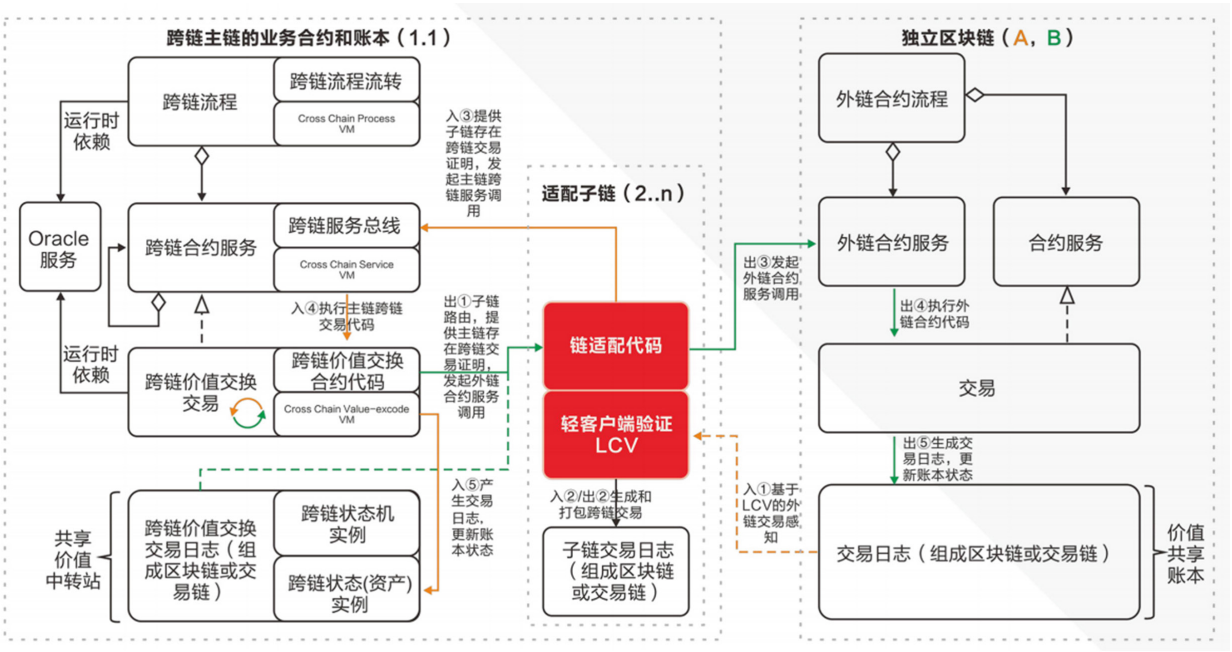
(1) 防止对合约内容进行篡改；

(2) 支持多方共识下的合约内容升级；

(3) 支持向账本中写入合约内容。

3.5 公链 (DApp 链 –Token 链 ––Token 链)

独立区块链完成相关性较高的业务领域的价值生产，要实现社会化商品和价值大流通，就需要跨链交易市场，通过跨链提供的跨链价值交换市场满足价值在不同主体自由等价流通。跨链具有兼容性，可以与现有、未来各种区块链兼容；跨链具有开放性，跨链具备让任何区块链接入的能力；跨链具有标准化潜力，让任何区块链接入，会渐渐形成一种接入标准，有助于推动区块链协议的标准化。根据业务功能、隐私保护、数据隔离、或者性能容量扩展的需求，EKT 建立多个独立的链并行工作，链和链之间可以通过跨链进行操作。



公链 (DApp 链 –Token 链 ––Token 链) 图

在 EKT 区块链网络中：

“主链”构成了信息主干道，不同的母链之间通过链路由协议交换信息。同时，一个主链上承载着不同的同构子链，这些子链是某个垂直领域或多个异业集群的分布式账本实现。子链间的通信则由跨链通信协议实现。通过区块链的分片，提高区块链系统的交易处理能力。相较于一条单独的区块链系统，链集群系统可以通过连接多条子链的方式在交易处理能力上直线增长。交易的请求通过链路由的分配进入不同子链，可以有效规避针对一条子链的集中请求。此外 ,我们可以在链路由上部署同构子链的不同节点数的集群，对于同构链而言，多节点数量的集群会有相对较高的安全性，少节点集群的处理速度则更快。此外，根据节点数量，地理位置，业务分类等不同需求，部署不同的链集群，对应不同需求将请求分发到合适的集群之中处理，帮助链网络根据业务需求灵活部署，为用户提供更高质量的区块链服务。

3.6 非对称加密

SHA–256 加密法（HASH）是一种加密技术，以椭圆曲线理论为基础，在创建密钥

时可做到更快、更小，并且更有效。SHA-256 利用 HASH 网络数据等式的性质来产生密钥，而不是采用传统的方法利用大质数的积来产生。HASH 加密法 SHA-256 （Elliptic Curve Cryptography）是一种加密技术，以 SHA-256 理论为基础，利用有限域上 HASH 的点构成的 Abel 群离散对数难解性，实现加密、解密和数字签名，将 SHA-256 中的加法运算与离散对数中的模乘运算相对应，就可以建立基于 SHA-256 的对应密码体制。

第四章 发展路径

2017 第四季度——EKT 技术团队成立

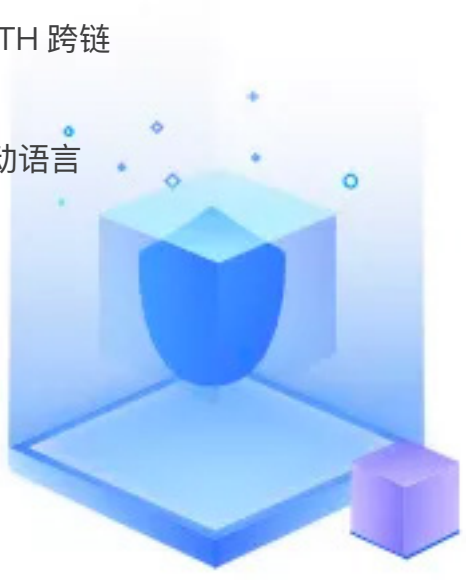
2018 第一季度——技术白皮书上线 & 代码开源

2018 第二季度——测试主网上线 & 标准白皮书上线

2018 第三季度——EKT 主网上线

2018 第四季度——发布主网钱包；实现 EKT 与 ETH 跨链

2019 第一季度——发布 VM 虚拟机和完全事件驱动语言



第五章　团队核心成员

黄湧涛　创始人

资深互联网从业者，曾就职于百度、阿里巴巴、中金等知名公司，连续创业者。具有丰富的互联网和金融行业经验。

周迅　CEO

90 后分布式系统专家，构建区块链行业首个多链多共识公有链 EKT，打造全新一代区块链开发平台，帮助实体经济快速走向区块链生态。

杨钢　CTO

原金山集团副总裁，曾任 WPS 首席架构师，千万行级代码工程经验，负责过日活 6 亿云存储项目金山快盘的研发。

郭小凡　COO

毕业于中国传媒大学广告学系，曾负责蓝色光标、爱奇艺、万合天宜等知名公司市场、运营。

李俊杰
技术合伙人

张祖缘
技术合伙人

王宗纬
客户端负责人

左智焕
前端负责人

盛帅
JAVA 核心工程师

周晶
Android 核心研发

上官
前端工程师

王晓婷
前端工程师

丁源源
前端工程师

裴张强
JAVA 核心工程师

张裕宝
JAVA 核心工程师

李柏林
JAVA 核心工程师

王志
核心运维工程师

贾宇
核心测试工程师

金希
产品经理