
fair
block

前言

面向游戏娱乐的第三代区块链技术

2008年中本聪的论文《比特币：一种点对点的电子现金系统》开创了比特币和区块链的概念，这套系统“基于密码学原理而不是基于信用，使得任何达成一致的双方能够直接进行支付，从而不需要第三方中介参与”。区块链的出现不仅仅是催生了“比特币”这一全新的货币体系，更是带来了互联网去中心化的的一次革命。区块链技术从诞生至今，经过10年的持续迭代，已经大大超过最初的“点对点电子现金系统”的定义，成为一种革命性的互联网底层架构。区块链技术将可以构建一个高效而可靠的价值传输体系，推动互联网成为构建社会信任的基础设施。

FairBlock致力于将这一愿景变为现实，基于全新构架的第三代区块链体系，我们试图建立一套去中心的、自由和稳定的全新互联网信任体系，并重点解决区块链应用开发中以游戏等强交互应用为典型代表的泛娱乐领域的开发痛点，推进区块链技术面向普通大众的普及和推广应用。

目录

1, 区块链技术的发展和现状

诞生于2008年的比特币区块链作为第一个大规模稳定运行的区块链网络，被人们定义为“一种能够让整个世界就一个公共拥有的数据库的内容达成一致的机制”。在完成电子货币的功能后，如何让区块链技术服务于货币之外的领域成为第二代区块链技术发展的方向——2014年由Vitalik Buterin发起的以太坊（Ethereum）是最成功的一次尝试。

以太坊项目最大的创新之处在于创造了一个功能完善的、内置图灵完备编程语言的（可深度自由调整的）区块，它允许用户编写任意复杂的完全存在于块链且由区块链传递的合约、自治代理和关系，以太坊的用户可以通过协议内置的脚本语言编码来实现任意交易类型，包括定制货币、金融衍生品、身份系统甚至更复杂的去中心化应用。

以太坊最核心的特性被称为“智能合约”，关于智能合约最通俗的定义是一个“包含价值并且只有满足某些特定条件才能打开的加密箱子”。因为区块内置图灵完备的编程语言同时具备价值知晓（value-awareness）、区块链知晓（blockchain-awareness）和多状态等特性，以太坊通过智能合约可以实现许多过去难以想象的应用。

目前基于以太坊已经衍生出百种应用，目标涵盖金融，物联网，虚拟资产交易，游戏，博彩等诸多行业。去中心化自治族中让许多原本无法运行或运行成本过高的运营模式成为可能。目前比较知名的应用有：

The DAO，用以太币资金创立，目标是为商企业和非营利机构创建新的去中心化营业模式，项目开始仅2个月便遭到黑客攻击，超过360万个以太币被盗，最终以太坊基金会决定以分叉处理此次攻击。

The Rudimental，让独立艺术家在以太坊区块链上进行众筹创作。

FreeMyVunk，虚拟物品交易平台。

Ujo Music，使用智能合约进行音乐销售。

2, 区块链技术面临的问题。

2.1, 性能问题和“挖矿”带来的资源浪费

尽管以太坊通过智能合约极大的拓展了区块链的应用范围，但仍有几个问题制约了区块链的应用范围。首当其冲的就是使用工作量证明（PoW）来建立共识机制所造成的性能问题。加密货币使用区块链作为基础，而区块链本身就需要使用Hash函数来作为数据正确无误的校验，在加密货币上使用工作量证明，是非常顺理成章的设计。具体上讲，分散在各处的计算机，竞赛谁能最早找出，搭配原本要打包的数据的穷举猜测值（nonce），谁就等同获得该区块的打包权（记账权）。此猜测值被找出后，与数据、Hash值一起打包成块后广播，经多数节点确认与承认，打包者就能获得打包该区块所提供的奖励。目前市面上的加密货币包括比特币、以太坊等均使用工作量证明，考虑到摩尔定律带来的性能增长，具体实践中通常会设置成随着参与竞赛的算力增加而提高寻找猜测值的难度，以维持合理的运作速度。

作为目前使用最广泛的共识机制，PoW架构简单可靠，在很大程度上也相当的公平，任何一个节点投入了越多的算力就有越高的几率得到记账权。而恶意者必须投入超过总算力一半的计算能力（51%攻击）才能保证篡改结果，这使得黑客攻击的成本非常高昂，难以实现。

但是，由于记账权（得到奖励的概率）与计算能力成正比，随着加密货币的盛行和增值，任何节点为了赚取更多的奖励必然会不断增加自己的运算能力，而大部分的算力都浪费在毫无意义的Hash计算上，这导致了巨大的算力和能源浪费。有数据指出，目前全球范围浪费在记账权争夺（又称：挖矿）上的电能已经超过一个小型国家的总使用量。遗憾的是，尽管消耗了巨量的能源，由于固有的低效率特性，基于工作量证明（PoW）的区块链网络性能远不能满足需求，以目前最被看好的以太坊为例，目前整个以太坊网络每秒钟仅能处理约25比交易。这种低性能远远无法满足现代互联网应用的要求。

2.2, 高昂的交易成本

节点为了争夺记账权的过程本质上是算力的比拼，这必然带来节点硬件成本的暴涨，在以太坊区块链上进行的交易涉及的费用最终都以以太币（Ether）来结算。每笔交易必须指定一定数量的“Gas”（即指定startgas的值），以及支付每单元Gas所需的费用（即gasprice），在交易执行开始时，startgas*gasprice价值的以太币会从交易发起者账户中扣除。交易执行期间的所有操作，包括读写数据库、发送消息以及每一步的计算都会消耗一定数量的Gas。虽然在以太坊设计中用户为每一种交易操作需要付出的Gas数量是固定的，但支付每单位Gas

所需的费用仍然是由使用者动态设定的。同时，支付的交易费用越高，节点就更有积极性来打包和处理这笔交易。目前在以太坊中进行一比最简单的以太币转账交易所需支付的交易费用已经高达0.1个以太币，这使得基于以太币的小额交易变得不可能。以太坊设计中，一笔复杂合约交易的每一个步骤都需要消耗一定数量的Gas，甚至于在执行某些复杂的智能合约时需要支付的交易费用已经超过合约本身。

2.3, 侧链交互与隔离的矛盾

以太坊另一个亟待解决的问题是侧链。侧链是基于主区块链产生的一个分叉，通常是为了实现某些特定目的或功能。在以太坊诞生之前，这种分叉通常以硬分叉的方式进行。以比特币为例，大家所熟知的"比特币"通常意义上是指Bitcoin Core（BTC），由传奇人物中本聪创立以来已经稳定运行近十年，可以说BTC是数字货币领域当之无愧的老大，目前市值已经突破1700亿美元。然而从技术角度上讲BTC相较于其他新兴数字货币并不占优势，区块大小仅1MB，交易延迟和交易费用都越来越高。

针对BTC面临的问题，BitcoinABC、Bitcoin Unlimited、BitcoinXT等多个团队共同开发了Bitcoin Cash（比特币现金，BCH或BCC）。BCH于2017年8月1日，比特币区块链长度478559时分叉，成为BTC的第一个硬分叉币种。BCH带来了多项升级特性，包括8M容量大区块、支持双向重放保护等。2017年11月13日由于最初采用的紧急难度调整机制(EDA)不稳定（有时会将BCH挖矿难度调高到BTC的三倍以上），BCH再次进行硬分叉并加入DAA难度调整机制。

除此之外，BTC还产生了Bitcoin Gold（比特币黄金，BTG），Bitcoin Diamond（比特币钻石，BCD）等多个硬分叉版本。BTC进行的硬分叉实质上是基于比特币原始区块链网络（主链）的某一时间节点，通过升级或调整代码衍生出来的一套新的区块链（侧链）。新的区块链虽然与原区块链同源，但实际上完全独立于主区块链，而且侧链无法和主链进行通信。同时，侧链需要建立自己的独立节点（矿工）网络。就节点而言，虽然一台服务器设备上可以运行多个不同区块链的节点程序，但一个节点程序仅能服务于唯一的一种区块链。

以太坊则是利用智能合约特性设计了ERC20代币规范。与BTC的分叉不同，基于以太坊发行的代币本质上是运行在主链中的一段智能合约，"代币侧链"以虚拟形态在智能合约中运行，不同于BTC的分叉通常意味着一个拥有独立区块链和节点网络的新"币"诞生。以太坊中

以智能合约形式衍生的"代币侧链"更多是用来发行某些针对特定功能或服务的代币，而通过ERC20标准发行的代币可以立即兼容以太坊钱包和交易所。包括ETH、EOS、XUC、OMG、ITC等知名代币均是基于以太坊发行的。

对于简单的代币发行而言，在以太坊上使用智能合约是一个简单而高效的好主意。但对于某些更复杂的应用而言，在主链中以智能合约形式运行会带来一个严重的安全隐患，由于智能合约形式的"侧链"与主链之间不像BTC侧链那样硬性隔离，侧链设计上的缺陷可能直接影响到主链，使以太坊受到重创的The Dao事件就是典型的例子。

DAO是指去中心化自治组织。其目的是为组织规则以及决策机构编写代码，从而消除书面文件的需要以及减少管理人员，从而创建出一个去中心化管理架构。DAO通常由一组人员来编写运行组织的运行规范（智能合约），合约建立后将开始融资。在该阶段人们添加资金来购买代币，来代表其所有权（这一过程被称之为众销或首次代币发行ICO）。募资完成后DAO就会开始运行，而人们根据所持有的所有权数量对DAO的发展决策进行投票。

然而，作为史上最成功的众筹项目之一，The DAO智能合约代码中的致命漏洞也造成了史上最大的数字抢劫案——黑客利用漏洞窃取了The DAO项目30%的以太币，按当时市值计算约价值5500万美金。The DAO事件也暴露了以太坊智能合约体系的巨大问题，以智能合约形式运行在以太坊主链上的代码无法和主链物理隔离，一旦发生重大BUG将有可能影响到整个区块链网络的安全。

除了安全因素外，以智能合约衍生的侧链本质上是运行在以太坊中的程序，这必然会进一步增加区块链的体积和复杂程度，这对整个以太坊节点网络来说无异于雪上加霜。

2.4, 与现实世界交互的难题

在以太坊中，有两种实体可以发起和接收交易：用户和智能合约。智能合约可以看成是活在以太坊网络上的自动化代理人，它有以太坊的地址与账户金额，可以发送和接收交易。每当有人向合约发送交易后，它就被激活并开始运行自己的程序，例如改变它自己的内部状态或者发送一些交易。

以太坊智能合约的代码由使用低级的基于堆栈的字节码语言写成，也被称为"以太坊虚拟机代码（EVM代码）"。代码由一系列字节构成，每个字节代表一种操作。通常代码执行的

是无限循环，程序计数器每增加1就执行一次操作，直到代码执行完毕或遇到错误、STOP或者RETURN指令。操作仅可以访问三种存储数据的空间：堆栈、内存或合约的长期存储。

用户在设计智能合约时面临的最大问题就是运行在EVM中的代码无法访问和调用区块链网络之外的数据。举例而言，金融衍生品是智能合约最常见的应用，也是最易于用代码实现的应用。在实践过程中的主要挑战是大部分金融衍生品合约都需要配合一个专门用于数据发布的合约，而这需要依赖某特定机构定期进行维护和数据更新，并提供一个接口使其他合约能够通过发送查询消息来获取关键的金融数据。

3, FairBlock区块链网络核心技术

3.1, 共识机制

工作量证明（Proof of work）和 股权证明（Proof of stake）是区块链技术中最重要的两个概念，也是区块链最核心的部分。区块链本身就是一个分布式账本，既然是分布式账本，那么必然存在以下两个问题：

a）如何在去中心化的网络中建立时间序列的概念；

b）当有多个节点完成记录交易的时候，应该采纳谁的记录？

第一代和第二代区块链通常使用POW来解决分布式系统中的共识问题。PoW(Proof of Work)是基于算力计价的共识机制。矿工通过解决一个复杂而无实际意义的数学问题来创建一个区块，并获得一定数量的币作为奖励。每个矿工解决问题的能力完全取决于自身的算力，为了赚取奖励，矿工会互相竞争，不断升级自己的算力，白白耗费大量的资源和能源，导致交易费用不断升高，却无益于提高交易速度。除此之外，持币者无法参与任何决策，决策权集中在少数几个矿池手中，与去中心化理念背道而驰。

而PoS(Proof of Stake)则是第三代区块链的象征，这是一种基于链上货币计价的共识机制。PoS用持币代替了算力，能够让持币者更多的参与到挖矿过程中，而且不需要计算复杂的数学问题，避免了资源和能源的浪费。已有的PoS解决方案主要分为四种：基于拜占庭容错的PoS，基于链的PoS，PoW/PoS混合,基于授权的PoS(DPoS)。基于拜占庭容错的PoS容错率较低，故障节点和恶意节点不超过矿工总数的1/3，且为了达到较短的确认时间限制了

验证者的数量。基于链的PoS本质上是PoW的一个货币计价改编。PoW/PoS混合只是一个过渡方案，最终仍会被一个纯粹的PoS机制所取代。基于授权的PoS通过选举代理人达成共识，牺牲了去中心化的概念，不适合公有链。在研究了已有的PoW机制和PoS机制之后，FairBlock提出了一个全新的PoS方案：基于竞争的PoS(cPoS)

3.1.1, cPOS

FairBlock在创世区块中生成并且分配21亿FBC，之后的区块创建由锻造委员会完成。为了解决PoS机制常见的富者越富问题，除创世区块外，其他区块的创建过程不再产生新币,所有收益均来自于交易费用。

3.1.2, 锻造委员会

锻造委员会是一个智能合约，其中包括数个拥有锻造权利的委员节点，每个委员节点都有机会创建区块。为了激励锻造，成功锻造一个区块将会获得该区块中的所有交易费。

任何节点都可以申请加入锻造委员会，但需要缴纳至少1FBC作为保证金，如果锻造者故意作恶，保证金会被罚没。同时我们有专门的机制避免节点恶意以低于1FBC的保证金金额尝试申请加入委员会。锻造委员的职责是创建新的区块。如果一个锻造委员连续3次不履行锻造义务将会被强制剔除出锻造委员会，而保证金会被扣留一定时间。扣留保证金是一种惩罚机制，用于惩罚不能正常完成其职责的锻造委员。不创建区块的危害比退出锻造委员会的危害更大，惩罚也更加严厉。

3.1.3, 锻造组和主锻造委员

锻造委员的投票权和保证金的数值相关。一个新加入的锻造委员并不会立即获得投票权，需要等待100,000个区块高度以后才会获得投票权。随着区块高度的增加，投票权将不断累积。如果一个锻造委员成功将区块添加到了区块链，则投票权被重置为0。由于所有人都可以查询到每个锻造委员的当前投票权，锻造委员将被按照地址的后两位进行分组，而组中投票权最高的锻造委员当选为主锻造委员，后续锻造组都倾向于验证和认同主锻造委员锻造的区块。

3.1.4, 奖励和作恶惩罚

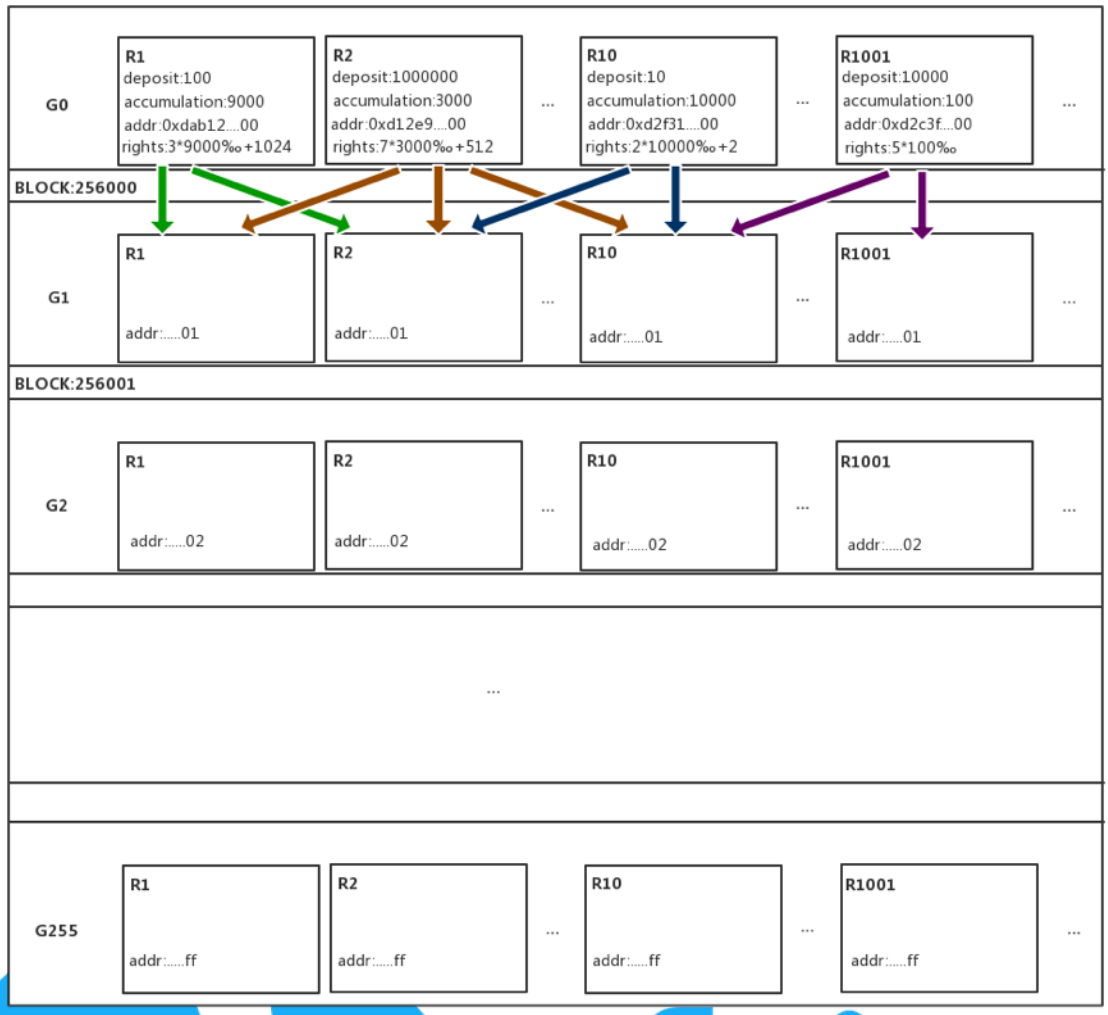
锻造委员的奖励由两部分组成：一是锻造委员创建新区块将获得该区块中所有交易费，二是举报作恶锻造委员将获得该作恶地址的所有保证金。由于cPOS体系下区块的创建过程消耗的资源极低，即使只有交易费作为奖励，锻造委员也能得到可观的利润。如此，类似比特币或以太币的额外奖励会引发富者越富的副作用则不复存在。

3.1.5, 共识建立和锻造过程

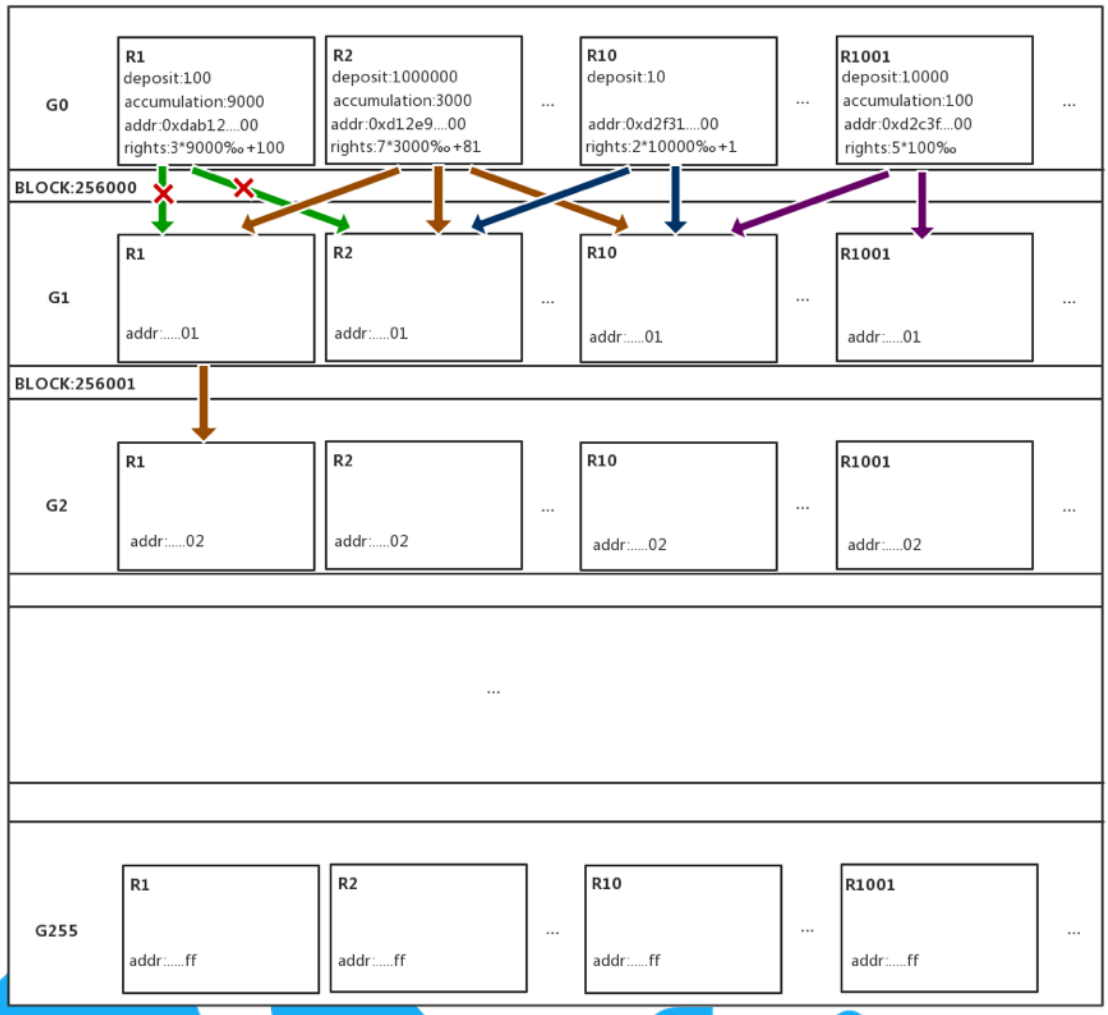
无论链上出现何种形式的分叉，总投票权最高的链便是正确的主链。因为主锻造者的投票权极高，所以分叉将会在极短的区块长度就达成共识，分叉将迅速被消除。

为了便于理解，此处只描述最高投票权锻造者和次高投票权锻造者的锻造过程和选择策略。实际情况更为复杂，但是原理相同。我们约定Rn代表分组中投票权排名第n位的锻造委员。

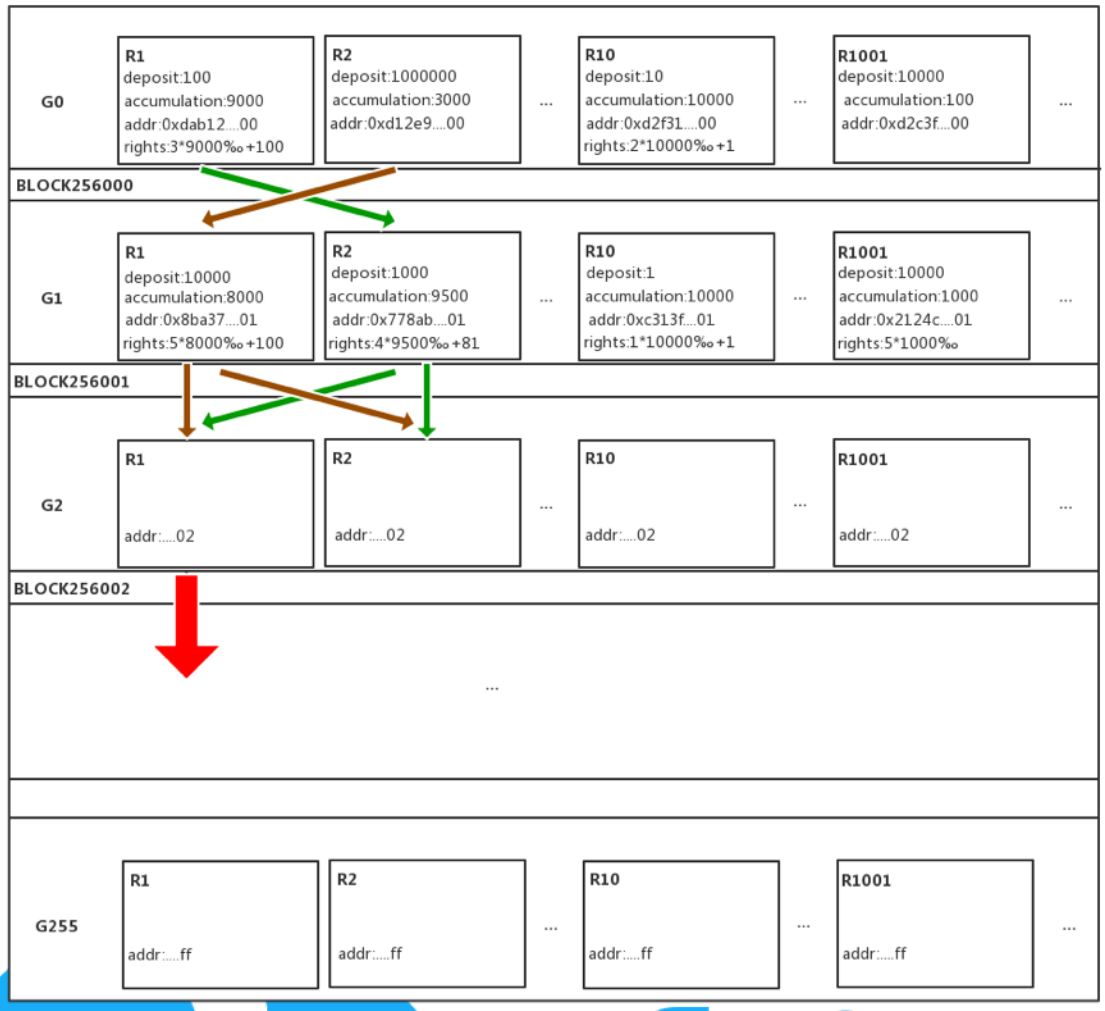
该图描述了在理想环境下一个锻造组的行为。



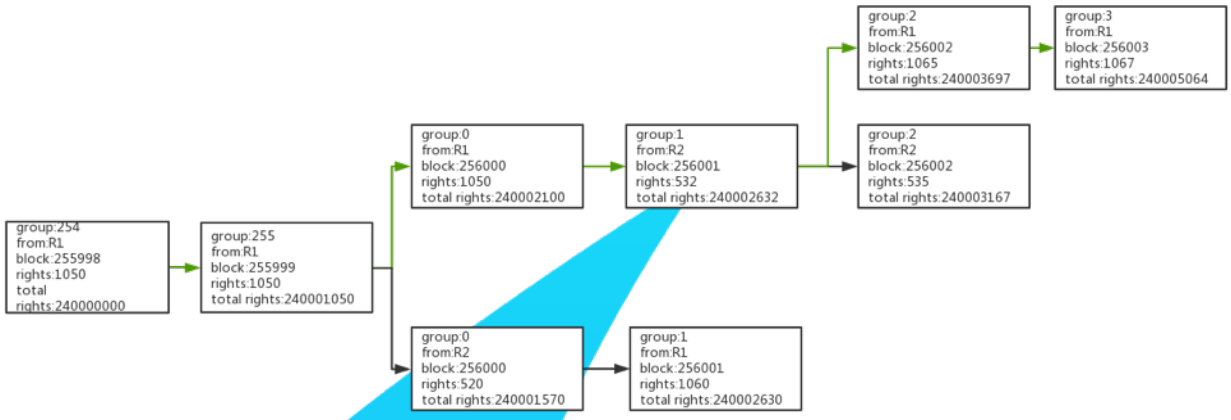
但现实世界的网络环境极为复杂，投票权最高的投票委员发出的广播可能没有被下一个锻造分组接收到。该图描述了这种情况下的备用方案。



由于总投票权和分组、保证金、累积区块高度、排名投票权、地址五者相关，使得锻造委员难以串通作弊，但不排除由于网络原因或者其他未知原因导致的：最高投票权锻造者接受了上一个区块的次高投票权的锻造者创建的区块，如图所示。



基于cPoS机制，分叉总是能在较短的区块高度被消除，如图所示。在区块高度256,000和区块高度256,001产生了分叉,而G2分组的最高投票权锻造者选择了其中一条链，该链的总投票权显著增大，有极高概率胜出。G3分组的锻造者会基于该链继续创建区块。

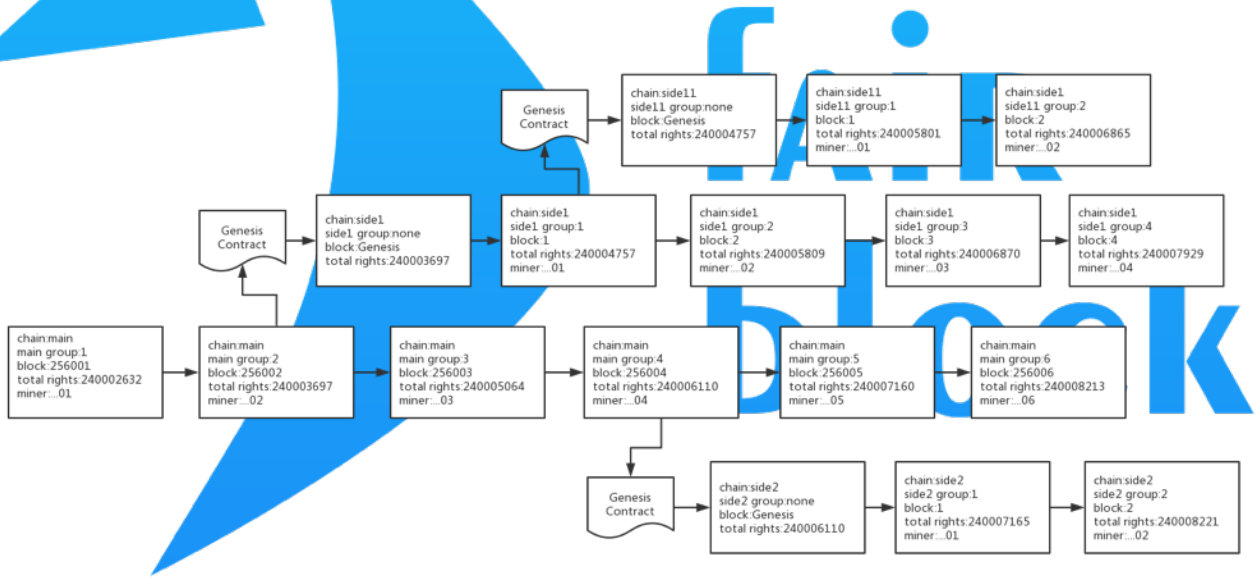


3.2, 超级侧链

超级侧链代码和主链相同，使用相同的共识算法（cPOS），有自己独立的区块链。得益于PoS的共识算法，超级侧链可以直接使用部分主链的节点网络，也可以独立部署节点网络。传统PoW算法极度消耗性能，矿工节点不会在同一个节点上挖两个不同区块链。

3.2.1, 侧链树

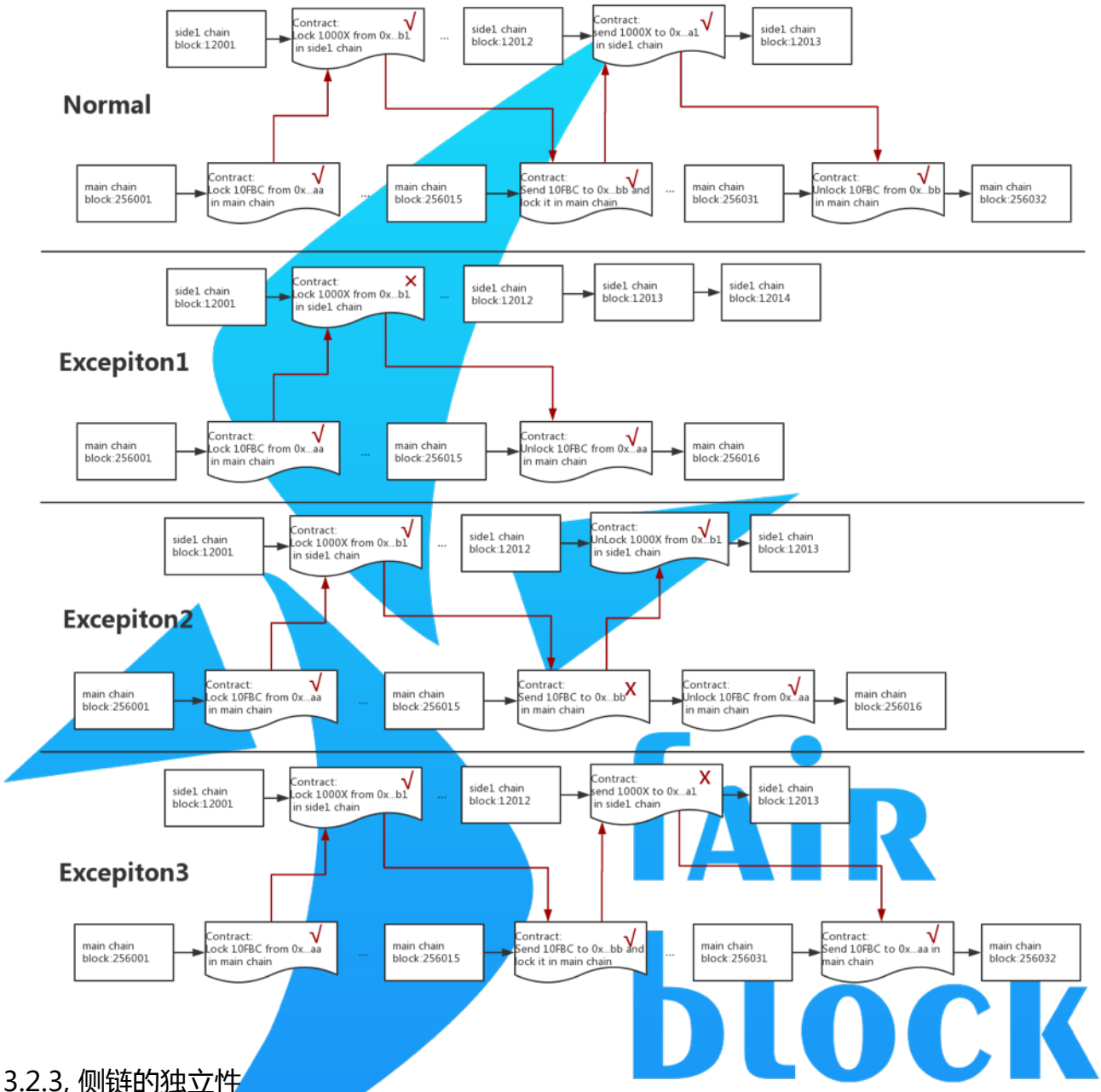
超级侧链支持把指定的侧链作为主链，继续分叉成新的侧链。依次类推，从而形成一个侧链树，如下图所示。



3.2.2, 跨链事务

在每条链上，智能合约或交易都是一个完整事务。由于侧链和主链处在不同的节点网络

中，各自区块链的生成者、速度和规则都不一样，无法完成一个完整事务。因此我们制定了一个跨链事务规范来保证侧链和主链可以运行一个完整事务，如下图所示。



3.2.3, 侧链的独立性

侧链架构的好处是数据独立，不增加主链的通讯和存储负担，避免数据过度膨胀和广播，也避免了代码bug被扩散到其他链上。

独立性既是优点，也是缺点。开发者完成了一个新的数字货币的开发工作后，还要考虑运营问题，也就是说要寻找足够的节点来运行它，否则难以维持足够的安全性。 从这一点来看，与以太坊相比，是缺陷，以太坊上一旦发布一个代币，所有的节点都可以为它工作，并

为它提供安全保障。

3.2.4, 侧链的灵活性

但是侧链架构依然有存在的必要性，因为并非所有的应用都需要那么高的安全性。侧链给用户提供了一种可以选择的灵活性。 比方说，假设主链有1000个节点，那么其中有的侧链比较重要，需要500个节点来运行它，有的侧链不那么重要，可能只需要100个节点，这完全是由用户和开发者决定的。 在以太坊上，则只能选择一种，即全部节点都来运行它。据说以太坊也打算实现一种分片的机制，实际上这在侧链系统中是一个早就解决了的问题。

侧链的灵活性还表现在，所有的区块链参数是可以定制的，简单的比如区块间隔、区块奖励、交易费的去向等。不过最重要的还是业务逻辑，侧链上可以轻易地开发出与你的业务相关的交易类型或者智能合约，甚至可以发布自己的侧链。

3.2.5, 主链与侧链的互利关系

主链与侧链之间是互惠互利的关系，主链为侧链提供基础设施，侧链则可以为主链补充更多的节点和开发者，以壮大整个系统。 侧链的开发者不需要提供机器，可以利用已经存在的主链节点，只需要节点主人拥有该货币。 另外，主链和侧链可以用系统提供的跨链事务规范进行价值交换，相当于为侧链的资产提供了一种价值的媒介。 开发者就不需要考虑交易平台的问题。

3.2.6, 解决生产力问题

只需按照侧链发布规范来写一个智能合约，侧链就得到了自动的发行，开发者只需要关心具体的业务逻辑，在这一点上，与以太坊的代币开发难度相当。而且侧链内的智能合约也使用和主链相同的编程语言，还可以调用主链上的智能合约的代码库，共享相同的开发者社区。

3.2.7, 节点安全问题

这里的安全问题与上面提到的因为节点不足或代码bug导致的问题不一样，这里说的安全问题是指侧链代码对节点造成破坏的问题。 我们希望每一个矿工节点都能信任并接受任意侧链，而且节点的主人无需信任侧链的开发者。这就需要提供一种安全防范的措施，比如，

防止侧链代码读取文件系统、过渡消耗网络、内存或CPU操作。 在我们的系统中，侧链代码会使用沙箱机制隔离的javascript虚拟机，然后我们为这个虚拟机植入一个定制的require和一些常用且安全的模块，最后再加载侧链的代码。 我们还通过容器技术来管理每个侧链的网络、内存或CPU的消耗，这样侧链的安装者就没有任何风险。

3.3, 安全函数

安全函数，就是一个按照系统约定的安全函数规范，实现的一种特定的智能合约。该智能合约可以被智能合约虚拟机执行，从而和外部服务器进行通信。包括获取数据，或者完成一个完整事务。示例代码如下所示：

```
1.  //a committee for manage security function
2.  contract SecurityCommittee {
3.    //the struct for voter
4.    struct Voter {
5.     bool agreement;
6.     address voter;
7.    }
8.    //the struct for security function
9.    struct SecurityFunc {
10.    address applicant;
11.    address securityFunc;
12.    uint32 publickey;
13.    mapping(address => Voter) voters;
14.   }
15.   //the return value of call security function
16.   struct SecurityResult{
17.     uint32 sign;
18.     address securityFunc;
19.     any param;
20.     any result;
21.   }
22.   //global store for all security function
23.   SecurityFunc[] public securityFuncs;
24.   // submit a security function, only can be used after the application is pass
25.   function submit(SecurityFunc func){
26.       securityFuncs.push(func)
27.   }
28.   // vote for a security function, need 2/3 voter's agreement
29.   function vote(address securityFunc, bool agreement) {
30.
31.   }
32.   // check if a security function can be called
33.   function isPass(address securityFunc) constant returns (bool){
34.
35.   }
36.   // calculate the digest by sign and public key from security function
37.   function calDigest(address securityFunc , uint32 sign) returns(uint32){
38.
39.   }
40.   //calculate the Digest of result with param
41.   function calHash(SecurityResult result) returns(uint32){
42.
```

```
43.  }
44.  //check if the hash is just the same
45.  function isSameHash(uint32 src, uint32 dst) returns(bool){
46.
47.  }
48.  //call the security function by address with params
49.  function callSecurity(address securityFunc, any params) returns (SecurityResult){
50.    if(!this.isPass(securityFunc))
51.      throw;
52.    return securityFunc.call(params)
53.  }
54.  //check if the result value is generated by the
55.  function checkSecurityResult(SecurityResult result) returns (bool){
56.    uint32 digest = this.calDigest(result.securityFunc, result.publickey)
57.    uint32 hash = this.calHash(result.result)
58.    return this.isSameHash(digest, hash)
59.  }
60. }
```

3.3.1, 封闭或扩展

在标准意义上的智能合约本质上是一段运行在虚拟机中的代码，代码所操作的数据全部存放在链上，在这种架构下，智能合约代码无法调用和操作区块链之外的数据。这就导致传统的智能合约无法和区块链外的数据进行交互，例如无法进行DNS查询，也无法得到安全随机数。

在FairBlock区块链架构中，我们提供一个用户定义的外部函数调用，运行在FairBlock链中的代码可以通过安全函数调用外部数据，这将极大的拓展智能合约的应用范围。

3.3.2, 如何定义安全

智能合约是被锻造区块链的节点来执行的，我们俗称矿工。如果一个安全函数没有被认为是安全的，矿工会拒绝执行包含该安全函数的智能合约。因此，我们需要引入安全共识。

首先，提供安全函数的智能合约会缴纳保证金。

其次，锻造委员会会对该智能合约进行投票，如果多数矿工投信任票，则安全函数就被信任，从而可以被调用。

最后，不信任投票随时可以进行，如果一旦超过半数，则保证金将被没收。

安全函数应该尽力说服矿工相信自己的安全性。比如，提供源代码地址，提供权威性的证明材料。

3.3.3, 执行和验证

传统的智能合约虚拟机，执行和验证交易都使用相同的逻辑代码。安全函数引入了外部数据，可能导致验证时调用结果不一致。（例如安全函数为生成随机数，每次调用结果都是不一样的）。

因此，FairBlock虚拟机设计了特殊的区块交易规范。我们将执行和验证分开，由虚拟机来分别处理，从而保证所有节点都能对包含安全函数的智能合约进行验证。

3.3.4, 分布式事务

在一个复杂的、需要保证严格一致的应用场景，比如多方交易。一个简单的安全函数调用，是无法满足这类应用的。因此，我们定义了分布式事务安全函数规范，保证了区块链可以和外部数据进行安全和一致的数据交换。

3.3.5, 权益

提供安全函数的外部服务器也是需要成本的，因此提供安全函数的智能合约可以规定每次调用的交易费。有了收益，安全函数的服务商就会有动力去提供更大的带宽，更大计算力。而且也会吸引更多人来提供安全函数，从而引入竞争，降低成本。

4, FairBlock的链上应用方案

FairBlock作为革命性的互联网底层架构，其所能实现的功能远不止数字货币或电子合约，许多我们常用的工作、生活甚至娱乐或游戏应用将可以以全新的形态在区块链上呈现。作为一套去中心化的协作体系，得益于技术团队带来的革命性技术，我们有能力将大多数互联网应用以去中心化的方式进行重构，互联网将变得更安全、更可靠、更富有想象力。

4.1, 高性能要求的应用

现代互联网应用对于响应速度和服务器处理能力的要求越来越高，以知名社交网络Facebook为例，作为社交网络的鼻祖，截至2017年7月，其月活跃用户数已突破20亿，每5次网页访问中就有一个指向Facebook这一全球最大的社交网站上。每天用户在Facebook上分享的内容超过50亿条，“赞”按钮点击次数超过45亿次。Facebook目前存储了超过3000亿张照片，每月照片存储容量约增加10PB（注，单位换算：1PB=1024TB）。另一个对性能要

求极高的应用是在线游戏。由于玩家在游戏中的大多数行为都需要和服务器进行交互和信息确认，传统游戏运营过程中，为了降低服务器的压力通常会分割出许多个独立的服务器。如果游戏应用完全在区块链网络中运行，这对整个网络的数据处理能力将是巨大的挑战。

面对如此庞大的数据量，哪怕将现有的区块链网络全部整合也无法满足需求。即使区块链的用户数量增长数倍，但受限于PoW设计上的低效率，区块链网络的膨胀并不能带来其处理效率的明显增长。相反，新增加的运算资源大多被浪费在算力竞争和防止意外分叉而进行的校验上。

相反完全基于cPoS技术的FairBlock能够极大地提高区块链的处理效率，在小规模的测试网络中我们已经可以实现每秒2000次的处理效率，和以太坊网络每秒25次的低效率相比，这是一次巨大的飞跃。另一方面，由于cPoS区块链中节点不需要靠算力来竞争记账权，整个FairBlock网络的性能将随着网络规模的增加进一步提高，新增加的运算性能也不需要浪费在无意义的Hash竞赛上。在FairBlock的高性能区块链网络中，开发者可以实现许多传统区块链难以支持的应用模式，包括游戏、博彩、社交分享，轻博客等。

区块链的另一个重要应用方向即时通信。在以太坊区块链网络中，哪怕性能不造成瓶颈，高昂的交易费用也阻碍了这类应用的开展。可想而知，发送给朋友的每一条信息都要收取不菲的费用。何况，同样受限于PoW的设计问题，每个节点最主要的收入来自于争夺记账权成功得到的奖励（挖矿），而每笔交易支付的交易费和挖矿奖励相比仅是九牛一毛。这种模式下每一个节点都花费大量的成本来提高运算性能，以提高自己挖矿的成功率。交易（记账）本身反而成为节点并不太愿意干的事情。可以预计，交易费用在将来还会进一步提高。

在Fairblock网络中，得益于优化的PoS机制，挖矿将成为历史。由于不比拼运算性能，任何个人计算机都能成为支撑Fairblock区块链网络的节点，而节点的收益也将全部来源于交易费。传统PoW网络存在的节点中心化问题迎刃而解，而节点充分竞争也将带来更低廉的交易费用。这使得开发者可以在FairBlock中构建完全去中心化的点对点或群组即时通信应用。除此之外，FairBlock也能为小额代币支付，线上游戏这类金额小而交易频率极高的应用提供低成本的去中心化区块链解决方案。

4.2, 与链外世界交互的应用

以以太坊为例，传统意义上的区块链"应用"通常是指某种进行简单条件判断从而自动处理交易的智能合约程序。问题在于，以太坊智能合约使用图灵完备的EVM语言编写，理论上可以实现远远比这复杂得多的高级应用。可是事实上，这类应用大多仍然停留在纸面上。这固然有以太坊本身低性能高费用问题造成的影响，但更主要的原因是以太坊智能合约无法访问区块链外的数据。

以太坊中的区块链应用通常只能依靠人工维护的数据发布合约来实现获取外部数据，可人工的介入却违背了其"去人工化"的设计本意，而且还可能进一步带来出错的风险。而使用安全函数则可以轻松实现这种应用。

而在FairBlock区块链上，开发者可以通过独创的安全函数功能来解决这一问题。在区块链上将部署一套安全的、可防止被篡改的安全函数系统。开发者可以让智能合约应用通过网关自动化获取区块链外的数据，这一过程是双向的，智能合约同样可以通过安全函数向区块链外的地址发送数据。

利用这一特性，开发者将能够在FairBlock区块链上运行更高级的智能合约应用。以前文提到的社交类应用举例，开发者将可以实现内容的分享接口，让用户可以方便的将自己在其他平台看到的图片、文章等内容直接发布到区块链中的社交网络上。另一方面，Fairblock的链上应用支持接入第三方支付，包括各类娱乐应用，游戏应用，博彩应用等都可以轻松实现商业化。而对于第三方支付这类对安全性高度敏感的应用接口，开发者可以在智能合约中约定支付功能的细节和限制，例如支付频率，是否需要二次验证或设置支付限额。

4.3, 在侧链上开发应用

一个区块链网络的复杂程度取决于链上运行的资产和应用的数量。对于某些复杂而与核心区块链资源（货币，身份验证等核心功能）交互并不频繁的区块链应用而言，直接在主链上以智能合约形式运行并不是一个好的选择。为此，FairBlock提供超级侧链来实现这一功能。事实上，我们更鼓励开发者以超级侧链的形式创建自己的链上应用，超级侧链在基础层面独立于主链，但提供通用接口供侧链和主链进行通信，侧链可以直接调用主链中的功能和数据，也可以和主链进行互相操作。开发者建立超级侧链后不同于比特币的硬分叉侧链，不需要自行建立新的节点网络，原有的FairBlock节点会自动为由FairBlock衍生的超级侧链提供服务。

由于超级侧链支持树状多层侧链技术，开发者可以在侧链上再次衍生侧链，以社交网络为例，开发者可以在第一层侧链上运行社交网络的核心架构，例如账户和用户信息等。而将其他功能例如聊天，轻博客，或其他社交网络应用运行在第二层或第三层侧链中。这种架构可以使开发者方便的搭建和管理复杂的区块链应用。

此外，超级侧链也能为开发者提供安全和可隔离的应用开发与测试环境，侧链中出现的问题和BUG也不会影响到主链。类似以太坊"The Dao"的灾难性安全事故在FairBlock中将被严格隔离在侧链上。而复杂的应用运行在独立的超级侧链上既能提高应用本身的执行效率，也能大大降低主链的臃肿程度。独立侧链的另一个应用方向是开发者可以基于FairBlock主链发布自己的数字货币，而通过超级侧链的交互性，能够通过运行在主链上的智能合约实现完全基于链上应用的交易所。

4.4, 时间区块链应用

FairBlock可以分叉成仅保留特定长度的超级侧链，传统区块链保留从创始开始的所有区块，而FairBlock侧链可以支持只保留特定时间长度的区块，这能有效的减少区块链的长度，降低设备的运算和存储压力。这使得FairBlock侧链可以被部署到大部分性能较低的设备上。

此外，以传统游戏应用为例，对大多数用户常规操作而言，服务器通常仅需保留数小时到数天以供回溯查询，除了某些核心内容外，大多数数据并不需要保留太长时间。而如果开发者希望完全基于区块链开发游戏，由于传统区块链必须保存所有数据，这将会造成区块链被大量无用数据占据。而通过时间区块链技术，开发者可以选择仅保留一定区块长度的数据。游戏应用或即时通信类应用的性能和通信速度将能够得到明显提高。

4.5, 需要安全随机数的应用

众所周知，现代计算机无法自行生成真正的"随机数"。常见的替代性解决方案是通过专门的"随机数供应商"来获取随机数，而这类服务商通常使用自然界的某种天然具有随机性的事物来帮助生成随机数。而受限于以太坊虚拟机的封闭性，智能合约只能生成伪随机数，这在用于某些对真随机要求极高的应用时就会带来很大的安全隐患。

以在线游戏应用为例，可以说游戏建筑在随机数之上，游戏中涉及装备掉落、开宝箱等情况都需要使用随机数来控制玩家获得物品的概率，若使用伪随机数将会使得游戏中的概率

性事件变的可以预测。另一个急需安全随机数的应用例如基于区块链网络的线上博彩，所有博彩玩法本质上都是基于概率的游戏，相对于其他复杂的游戏而言，博彩玩法以代码实现是非常简单的，同时博彩行业天生对匿名性，安全性，反作弊能力要求极高，区块链技术可以说是博彩行业最有前景的发展方向之一。目前虽然在以太坊上已经出现了很多链上博彩应用，但受限于以太坊本身无法使用安全随机数的缺陷，目前已有的博彩应用事实上只是通过区块链完成数字货币的交换，而博彩核心的概率运算仍然是在区块链外的服务器上完成，这种模式是不透明和庄家可操纵的。而一旦在应用中引入外部提供的安全随机数，博彩应用将可以完全在区块链上运行，体现区块链透明和公平的原则。

4.6, 链上娱乐时代

区块链应用通常被认为仅面向部分严肃领域，例如金融业，商业或互联网基础服务。但随着区块链技术的进步和计算机性能的提高，越来越多的开发者尝试在区块链上设计各种娱乐性应用。以目前流行的基于以太坊的宠物收集养成游戏Crypto Kitties为例，用户可以花费以太币获得一只随机的“猫”，同时消耗资源将猫养大。用户之间可以自由的交易自己拥有的猫，而猫的价值和稀有度相关。在这款游戏中的“猫”本质上是以太坊中的一种以以太币定价的数字资产，“猫”的获得与交换都是以智能合约的形式在以太坊网络中进行。从某种意义上看，这就是一种由以太币衍生的“猫币”。然而这款游戏几乎也代表了基于传统区块链开发游戏的极限。市面上更多的看似更加复杂的“区块链游戏”本质上只是内嵌了某种数字货币的交易功能罢了，其核心的游戏代码逻辑仍然是以传统模式运行在中央服务器上，并不能真正保护玩家的利益和彻底摆脱游戏厂商操纵的可能性。

而在FairBlock区块链系统中，得益于独有的超级侧链和安全函数等特性，我们能够提供一整套功能完善的API使各类娱乐应用可以完整接入FairBlock区块链，享受安全和去中心化的全新娱乐体验，很重视游戏公平性的玩家将会欢迎甚至只支持区块链游戏。

4.7, 全新的区块链生态

过去，一个稳定运行的区块链项目上的各个参与者赚取收益的主要模式只有两种。“挖矿”和收取交易费用。“挖矿”基于低效率的PoW算法，不仅制约了整个区块链的性能，还浪费了大量的算力和电力资源。同时正因为“挖矿”收益为主交易费收入为辅，造成了交易费高启，

为了整额生态体系的健康，类似比特币或以太币那样极高的交易费用也是不可取的。

FairBlock在设计上彻底放弃了PoW算法这种“挖矿”模式。同时cPoS为节点带来了极低的性能要求和全新的竞争模式。随着节点网络的充分竞争和进一步扩大，FairBlock用户将能够享受极为低廉的交易费用。

与此同时，我们更为区块链网络带来了全新的商业生态。首先是提供安全函数，作为FairBlock区块链体系中最重要的特性之一，可以为区块链应用提供更大的想象空间，而安全函数的设计或提供者将能通过为其他开发者提供服务获得持续的收入。基于安全函数提供功能的复杂性和服务质量进行竞争，这将成为区块链上重要的商业生态之一。

另一个重要的模式则得益于超级侧链技术的出现。FairBlock超级侧链的一个重要特点是支持树状多层侧链结构，开发者可以在侧链上开发新的侧链，这为一个全新的侧链应用开发市场提供了基础。开发者将可以基于FairBlock超级侧链开发独立的区块链应用平台并面向细分市场建立自己的独特生态体系，这将成为FairBlock中最具想象力的商业模式。

基于以上生态模式的可能性，FairBlock真正有望成为互联网上的分布式操作系统。在FairBlock第三代区块链技术的商业生态构建蓝图中，毒瘤式的“挖矿”获利生态将会让位于以程序员开发共建为主获利的生态，获利更将是因为为其他人产生实际价值，也将使社会资源得到更加有效的利用，就如uber带来的共享经济惠及普罗大众，而不再是由少数人堆砌无用硬件把持的军备竞赛游戏。

让我们用区块链使世界更公平！

7, 团队

7.1, 核心团队

发起人：Calvin Ng

资深游戏人和创业者，20余年从业经验，各大游戏公司任职，曾将“魔兽世界”代入中国

Zmax leo

20年从业经验，大型多人在线网络游戏商用开发引擎的创始人

Fenix （ Berkeley ）

7.2, 顾问

Adam Stradling

比特币和区块链开拓者, Bitcoin.com创始人

Tiago

Aptoide创始人 AppCoins发行人

Ryan Terribilini

Google Play运营资深策略师, Ripple 平台合作总监

Gaurang Torvekar

Indorse 联合创始人，以太坊新加坡大会(Ethereum Singapore Meetups)的联合组织人

Andras Kristof

FRD首席区块链架构师，比特币、以太坊和Ripple合作者，《数字代币手册》(Handbook of Digital Currency)合著者

7.3, 合作机构

Aptoide

Gumi

Gobi Partners

Google Play

Bitcoin.com

Kyber

8, 法律