AChain 区块链技术白皮书

为的企业级分布式应用而生的智能合约平台

目录

1 AChain 区块链的背景及意义

自 2008 年 10 月 31 日中本聪发布了比特币的白皮书《Bitcoin: A Peer to Peer Electronic Cash System》至今已经接近 9 年，区块链技术在不断的突破，如图灵完备的智能合约在区块链上的实现，石墨烯技术对交易性能方面的提升，闪电网络对链下交易渠道的完善等等。与此同时，众多基于区块链技术的解决方案层出不穷，如防伪认证、公示公证、供应链金融，征信共享等等。一方面人们认同区块链所代表的去中心化、不可篡改、无需信任等一系列特性所构建的价值传输网络，另一方面区块链应用落地难，如何理解区块链技术并将区块链的技术特性很好的同企业现实需求结合起来一直没有得到很好的解决。

Achain 区块链致力于打造高性能企业区块链平台，开发企业区块链业务应用，消除企业对区块链技术应用的认知恐惧，将区块链技术特性表现成一系列可视化、可配置的行为，为不同行业、不同规模的企业打造一个去中心化应用平台，极大的降低区块链开发的成本从而丰富整个社会的区块链生态。

2 AChain 的设计理念和设计原则

- 新型的智能合约平台，在满足图灵完备的基础上，追求安全与高效。
- 模块化的共识机制，满足企业在不同应用场景中的灵活需求。
- 模块化的非对称加密算法，满足性能、知识产权方面的要求。
- 低成本的检验智能合约的框架，规避智能合约漏洞。
- 通过一系列可视化工具使智能合约、注册、交易、事件通知等区块链概念具象成可视

化工具的一系列行为。

- 区块链上数据与链下数据的无缝融合，更灵活的触发链上合约，更便捷的订阅链上事件。
- 区块链网关的设计，以满足跨链的价值及信息传输。
- 智能合约的升级，以有限和被监督的权限弥补合约中未能发现 BUG。

在解决企业级区块链分布式应用的落地过程中，阻力一方面在于使得企业理解区块链技术去中心化、分布式、不可篡改等核心特征，并形象化的出现在人们眼前。另一方面在于不同的业务场景下灵活配置需求。如 POW（Proof of Work）的共识机制很难被行业应用大规模部署，同时根据参与企业间的信任情况，区块的间隔时间及网络状况导致的分叉成本都会影响共识机制的选择。AChain 默认配置了针对企业区块链环境中更具普世价值的 RDPOS 共识机制（详见 3.4 章），并提供诸如 POS（Proof of Stake）、DPOS（Delegated Proof of Stake）、LPOS（Leased Proof of Stake）、PBFT（Practical Byzantine Fault Tolerance）的共识配置选项。而在加密算法方面，同样可以从 ECC 椭圆加密算法和国密加密算法中进行选择。这种灵活性为不同规模、行业的企业在不同应用场景下提供了足够的可能性，从而为 Achain 的广泛应用提供了基础。而做为公有链的 AChain 同企业基于 AChain 的区块链之间的联通与价值传输将通过跨链网关完成。

然而千里之行始于足下，AChain 为了让企业迈出坚实的第一步，为了区块链分布式应用能更好的落地，并降低企业开发理解区块链分布式应用的成本，AChain 设计了一系列可视化工具，从智能合约的编写、注册、调用、充值、升级、销毁，到链上发生的每一个交易，交易当中的某个事件的发生都可以通过工具得到直观的展示，帮助企业用户理解当前在区块链上实际发生了什么，以及怎样同企业应用的业务逻辑相对应。简而言之，AChain 在易

用性、灵活性两方面着力做到最好，把区块链分布式应用开发成本降到最低。

从业务需求的角度，企业需要图灵完备的脚本来实现在区块链上的业务逻辑，脚本（智能合约）的开发也更符合其他领域程序员的习惯。AChain 主要从四个角度对智能合约做了改进：一个是开发的便捷性，从 LUA 语言，C#语言到 JAVA 语言，使得程序员可以便捷的上手开发智能合约。二是智能合约本身的安全性，从设计之初提供了 0 成本模拟测试智能合约的机制，完备的测试框架下全路径测试，可以对智能合约自身的漏洞最大限度的规避，而所有这一切无需用户消耗代币。不同于现有测试链验证智能合约的方案，这套解决方案使得智能合约在正式公链上去执行智能合约，并与正式环境中的其他智能合约或链上状态产生交互，从而提供比测试链方案更逼近真实环境的完备测试框架。三是智能合约的高效性，从虚拟机的调优入手最大限度的使合约调用接近普通交易的性能。四是在发现智能合约代码本身的漏洞而不得不对智能合约本身进行升级，如何设计好一套协议，保障智能合约参与者的利益，从而在新的共识下升级智能合约。

智能合约的升级本身是极具争议性的题目，但现实商业环境中即使软件经过了最严苛的测试依然可能存在需要升级智能合约的情况，这份权利应该被如何控制和监控，从而避免区块链不可篡改的特性。AChain 从博弈论的角度通过升级协议最终实现智能合约的升级，保障各方利益。根据与智能合约产生交易及充值金额赋予相关账户不同的投票权重，由区块链的出块账户冻结该智能合约的交易，如果对新智能合约代码（新的合约字节码链上 HASH 值）按投票达到了 81%的比例支持，智能合约将按照预定的协议升级，而之前智能合约的状态与存贮都将被保留下来。

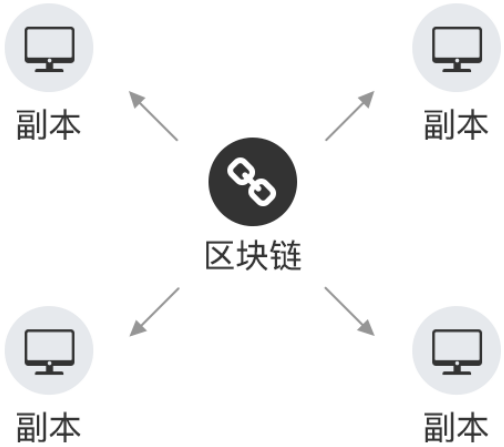
为了满足区块链之间的价值与信息传递，AChain 设计了跨链网关，从而允许 AChain 同其他符合该规范的区块链之间传输价值与信息，避免了不同区块链的一个个孤岛。在现实商业环境中，这一点尤为重要。这种情况类似于在企业为了提高 IT 及业务系统的效率，而跟

进、采纳基于 Cloud 的解决方案。很多情况下出于对隐私、安全性、逐渐过渡的角度，企业以私有云的解决方案开始，但今后从运营的效率角度，公有云是有独特的优势的。这时企业一定会重视 Hybrid Cloud 解决方案，从而保护之前私有云的投资并且享受公有云的种种好处。AChain 的跨链网关正是对应 Hyprid Cloud 的理念，把未来企业的区块链解决方案连接并一定程度上统一起来。

AChain 在设计之初，秉承的理念就是通过开放、创新、合作，让区块链能够更好的服务于当前和未来的企业，通过区块链的分布式节点、安全、无法篡改等特性可以解决企业用户在证据留存、征信、资产数字化等等方面所面临的痛点，使得企业以极低的成本在一个安全、高效、大数据量、大吞吐量、高 TPS 的区块链分布式应用平台上构建自己的系统。

3 AChain 设计体系介绍

3.1 Achain 区块链系统



区块链是一种以密码学技术为基础，以去中心化的方式，对大量数据进行组织和维护的数据结构。区块链建立在分布式的数据节点上，并且通过共识算法进行同步，因此特别适合

作为数字资产的账本。区块链上的数据全部都附有相关人的数字签名，不可伪造。

3.2 账户模型和账户体系

AChain 区块链系统里，每一个客户端是都是一个本地钱包。用户在自己的本地钱包中创建一个或多个账户，并且进行相关的账户操作。每一个账户都有唯一的私钥并对应唯一的地址。账户分为普通账户、注册账户、代理账户、出块账户、合约账户：

普通账户：普通账户是一个本地账户，仅在本地钱包内有效，用于区分当前钱包内的多个账户。只有在本地钱包中才可以通过这个普通账户的账户名向该账户转账。

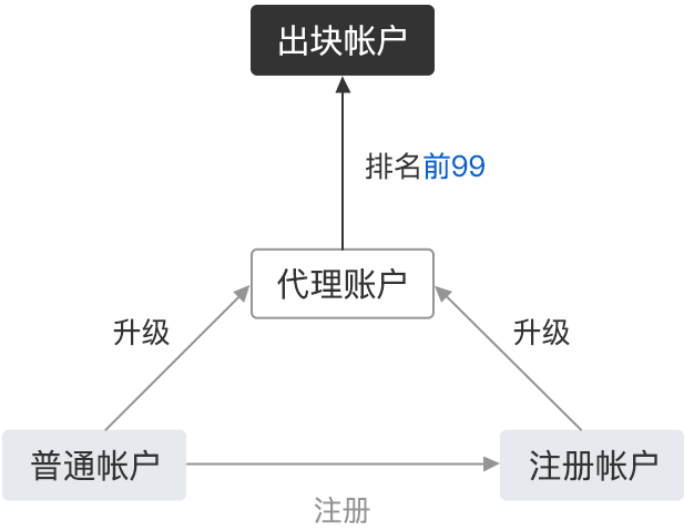
注册账户：普通账户可以升级为注册账户，升级后账户名会被注册到区块链上。用户可以直接使用账户名进行转账操作。区块链上的账户名具有唯一性。升级成注册账户，需要花费一个基本交易手续费。链上的任意账户都可以通过注册账户的账户名向该账户转账。

代理账户：普通账户或注册账户都可以升级成为代理账户，代理账户拥有注册账户的所有功能，并且具有被投票权。

出块账户：当代理账户的排名进入前 99 名（包括第 99 名）时，代理账户可以参与系统出块，从而获得出块收益。

合约账户：当合约被注册上链后会产生一个合约账户，合约账户不属于任何一个用户账户，

也没有公私钥。用户账户可以向合约账户中转账，合约账户也可以向用户账户转账（仅在合约代码中）。



3.3 密码学模型

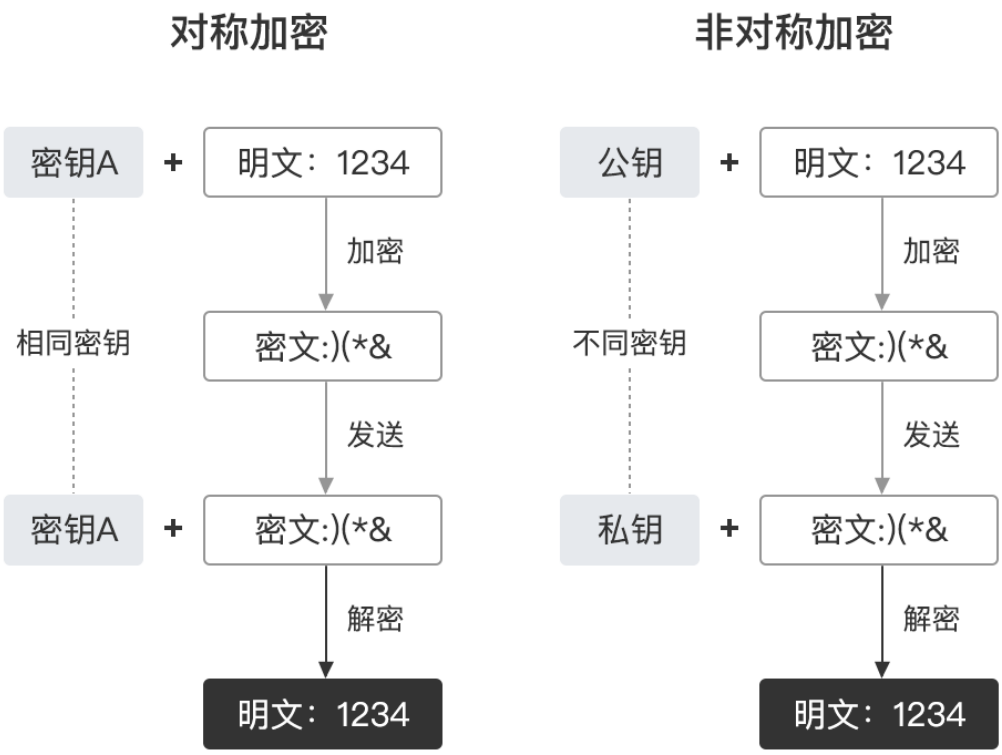
私钥：非公开，是一个 256 位的随机数，由用户保管且不对外开放。私钥通常是由系统随机生成，是用户账户使用权及账户内资产所有权的唯一证明，其有效位长足够大，因此不可能被攻破，无安全隐患。

公钥：可以公开，每一个私钥都有一个与之相匹配的公钥。ECC 公钥可以由私钥通过单向的、确定性的算法生成，候选方案为 secp256r1（国际通用标准）、secp256k1（比特币标准）和 SM2（中国国标）

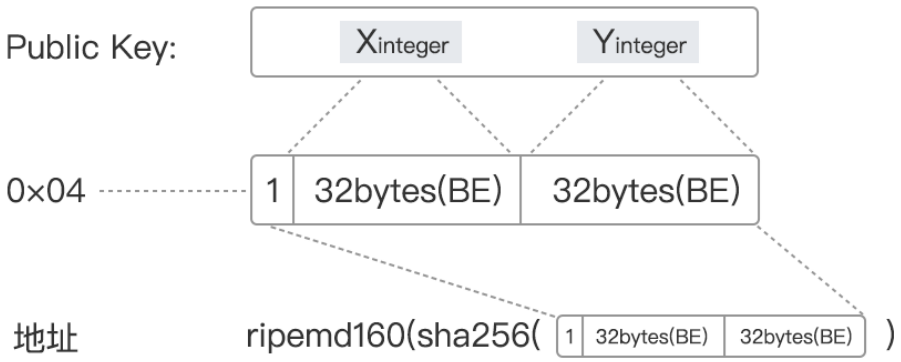
对称式加密就是加密和解密使用同一个密钥，也就是说采用这种加密方法时候，加密方与解密方需要使用同样的密钥进行加密和解密，该方式只需要一个密钥+特定算法对数据内容进行加密，加解密效率比较高，因此在对被广泛使用。但是因为解密方也需要密钥，所以保证

密钥的安全也成为了一个难题。

与上面相反如果加密和解密是采用不同的密钥，也就是非对称加密密钥密码系统，每个通信方均需要两个密钥，即公钥和私钥，这两把密钥可以互为加解密。如果用公钥对数据进行加密，只有用对应的私钥才能解密；如果用私钥对数据进行加密，那么只有用对应的公钥才能解密。因为加密和解密使用的是两个不同的密钥，所以这种算法叫作非对称加密算法。 非对称加密算法实现机密信息交换的基本过程是 :甲方生成一对密钥并将其中的一把作为公钥向其它方公开；得到该公钥的乙方使用该密钥对机密信息进行加密后再发送给甲方；甲方再用自己保存的另一把私钥对加密后的信息进行解密。公钥是公开的，不需要保密，而私钥是由个人自己持有，并且必须妥善保管和注意保密。如下图所示两种加密不同：



地址：:地址是公钥的摘要，是为了用户能够方便交易而产生的，因为公钥较长约有 130 字符，而地址比较短，约有 35 或者 36 个字符。地址由公钥推导生成如下：



推导关系：私钥 >> 公钥 >> 地址。过程均不可逆。拥有私钥便拥有一切。

3.4 共识机制

AChain 平台默认建立在 RDPOS （Result Delegated Proof of Stake）共识机制基础上：

- 产块共识

在系统正常运转的情况下，产块共识保证了在同一个产块周期内只有一个确定的产块代理会产块。且代理生产的块可以通过其他其他节点的验证，并且前一个块和后一个块之间可以通过特定的规则串联在一起（后一个区块中记录前一个区块的区块哈希）

- 产块代理共识/顺序共识

由投票决定新一轮的产块代理，对于所有节点来说，每个代理的投票在所有不同的节点上都是确定的，而且是一致的。

由 random_seed 来决定新一轮的产块顺序。产块顺序是由所有代理节点共同决定，并保证无法被预知和篡改。在所有的节点上也可以达成共识。

- 交易/块验证共识

处于相同状态下的多个节点，进行相同的操作，产生的结果必然是一致的

- 合约共识（特殊的交易共识）

智能合约的同一个操作可能会在不同区块链节点不同时间执行，同状态的节点执行同一个操作，任何时间任何节点的执行结果以及对状态的改变都一样，可以复现执行结果，并且智能合约本身不可篡改，从而可以确保达成共识并且不可伪造执行结果。

这里与 DPOS 的区别是，由谁进行智能合约执行的验证。DPOS 采用的是代理节点执行智能合约同产块节点的执行进行比对。而 RDPOS 会根据产块节点当时打包的结果交易的状态（即这里的 R-Result）来动态决定由代理或全体节点验证智能合约的，针对特殊的智能合约，如智能合约执行时间较长，智能合约内部状态占用空间较大可采用不同的策略，既能满足智能合约被快速验证的情况，也能保障智能合约结果交易过大导致交易无法被打包时，出块节点仅打包结果交易 Hash，从而所有节点自行验证交易 Hash 的情况。

3.5 智能合约

- 智能合约介绍

1995 年，密码学家尼克·萨博（Nick Szabo）提智能合约的概念。他在发表在自己的网站的几篇文章中提到了智能合约的理念。他的定义如下："一个智能合约是一套以数字形式定义的承诺（promises），包括合约参与方可以在上面执行这些承诺的协议。" 他提出"区块链计算机"，将其定义为依靠密码学和共识技术，为存储在区块链数据架构中所有数据，创建不可伪造证据链。

- 区块链上的智能合约

链上存储的一段可执行字节码，可以被外部触发调用。可以由图灵完备的合约语言编写，经过编译后存储到区块链上。支持执行指定数量的字节码之后的停机操作。

区块链是智能合约天然的生存土壤，是由区块链一些特性决定的：

✓ 去中心化的系统

✓ 安全的公私钥密码体系

✓ 交易数据全网共识，不可篡改

✓ 合约中的交易可以通过区块链上资产进行交易

- AChain 区块链上的智能合约

在 AChain 区块链系统中，将智能合约设计为一个包含代码和数据存储的链上对象。合约的拟定者可以用支持的计算机语言描述合约条款，设定执行条件，以及达到执行条件后执行的操作，参与接口等。在合约拟定者将合约注册到区块链上后，其他用户可以通过调用接口来参与合约。在合约语言正确表述合约内容的前提下，在达到执行条件时，系统会按照合约代码的描述执行相应的操作。并不会有现实中参与方拒绝履行条款的现象

3.6 交易验证

- 从交易类型上来看，交易可以分为普通交易和合约交易

普通交易（非合约交易）

普通交易在所有节点上验证方法一致，在所有的节点上达成共识。

合约交易（合约原始交易/合约结果交易）

合约交易分为合约原始交易以及合约结果交易

合约原始交易只有在受托人节点上验证会打开解释器进行验证，并产生结果交易；

除了在交易创建的节点上，在进行交易创建的时候，其他情况下合约交易在普通节点上验证是不会执行解释器，只做基本验证。

合约交易在受托人节点上达成解释器验证共识，并产生结果交易；

在所有普通节点上达成结果交易共识。

- 从交易的本质上来看，交易可以分为确定交易和不确定交易

确定交易（普通交易/合约结果交易）：在一些交易中，如转账，用户可以明确的知道要执行的操作，这些交易的创建时会直接将要执行的操作执行存入交易中。受托人只需按照交易内容执行打包没有问题的交易即可

非确定交易（合约原始交易）：在另外一些交易中，如调用合约，调用者并不能预测合约会执行的操作，因此创建此种交易时仅保存用户请求，当受托人打包交易时，按照合约执行，将结果连同初始的请求一同打包成结果交易，结果交易可以认为是一个确定的交易。

- 交易验证

对于确定交易，验证分为两部分：

签名验证：在涉及操作资产时，如个人资产出账操作，需要确保交易发起者时该笔资产的所有人。系统在操作需要签名的资源时会检查交易中是否存在所需签名。

执行验证：按照交易内容执行交易，再不存在逻辑冲突的情况下，如不从只有 20 余额的账户提 30 出来之类的操作，都可以通过验证。

对于不确定交易，受托人在进行执行验证时会执行发起人的请求根据当前链的状况生成可行的结果存入交易，用于标识发起人的操作的结果。

3.7 区块链交易模拟器

在钱包打开并解锁的状态下，可以打开交易模拟器模式。交易模拟器模式会根据当前的状态产生一个新的缓冲区，在交易模拟器模式下进行的合约相关的交易行为都会被在本地执行合约解释器并验证，并将结果记录在这个缓冲区下，且不会被广播到区块链网络中。当交易模拟器模式被关闭时，这个缓冲区就会被丢弃，不会造成任何影响。

3.8 区块链链上事件

对于普通节点来说，由于在验证接收到的区块的过程中，并不会像代理节点那样执行解释器进行验证，因此对于合约中发生的一些情况缺乏感知能力。因此可以由代理节点在执行相关的合约中达到的一些特定的关键点（需要产生一个通知信息）的地方产生一个合约事件，事件功能可以用来提升用户对交易以及交易中的一些中间环节的感知能力，并且可以对感知到的状况做一些自定制的反馈，提升可交互能力。（通过本地脚本的方式，或者由用户自行定制的一些方式）。

3.9 区块链合约网关以及区块链链间网关

- 针对链下数据的区块链合约网关

链下的事实数据通过业务网关投放入区块链，触发链上合约交易的共识。

- 针对另一条链的区块链链间合约网关

链间数据交换，合约的输入是由另一条链上的交易数据达成的共识，由链间合约网关承担数据转运的工作

4 Achain 实现方案

4.1 合约和虚拟机

合约语言：我们使用 Glua 语言作为 AChain 区块链上智能合约使用的默认编程语言，支持静态编译成字节码然后在区块链网络中根据需要执行字节码。

Glua 是一种图灵完备的编程语言，编译器和字节码虚拟机为在区块链中做了针对性设计和优化。

合约解释器：合约解释器是 Glua 的字节码的解释器，在区块链网络中涉及到智能合约的操作或块同步验证中，区块链节点需要时会从区块链中取出合约字节码，用 Glua 字节码解释器加载字节码，然后使用合适的参数调用需要的 API，得到的运行结果和上下文状态变化会被区块链使用。

一次对智能合约的操作，可能在很多不同节点不同时间调用不定次数，但是同一个操作在不同节点不同时间每次调用的结果和对上下文状态的改变都是一样的。

智能合约的操作，因为需要不同节点的计算机资源进行执行以及占用区块链容量和网络流量，所以智能合约的操作需要扣除一定的执行花费。

合约的生命周期

- 编写 Glua 源码文件
- 使用 Glua 编译期将 Glua 源码文件编译成 Glua 字节码文件（如果编译成功，否则根 据报错返回第一步）
- 使用 Glua 字节码文件注册链上临时合约
- 向合约进行转账
- 使用一定参数调用合约的 API
- 链上临时合约升级到链上永久合约
- 链上临时合约销毁为链上不可用合约（合约依然存在但是不可继续使用）
- 链上合约导出 Glua 字节码文件文件

合约语法主要特性：

- 完整的图灵完备的编程语言
- 编译期静态类型

- 直观易用的语法
- 支持函数闭包和高阶函数，Lambda 表达式，支持函数式编程风格和过程式编程风格
- 合约中支持引用其他合约
- 提供常用内置库
- 智能合约的状态存储的改变，占用区块链存储总容量小

合约字节码解释器的优势： 针对区块链做了安全性优化以及执行开销计算和控制

4.2 可共识的随机数发生器

随机数的计算方法：

使用未来某块的 random_seed 做为生成随机数的依据。random_seed 是通过上一次的 random_seed 和当前出块节点的 previous_secret 计算，相当于是由多个代理节点共同维护并计算出来的结果。

并且 previous_secret 是上轮就已经计算并确定，且在上轮出块后就将其摘要公布，可以被其他节点验证，因此可以认为这是一个可靠的，由多个代理共同维护计算并验证，可共识的随机数生成算法。并且是无法被操纵，无法被推算的随机数。

应用：

- 代理出块顺序
- 合约获取随机数

4.3 区块链合约交易模拟器

模拟合约交易的验证，提供无成本的合约调试方案

打开合约交易模拟器时会在当前状态下产生一个缓存 ,在合约模拟器内创建的合约交易会在缓存内验证并提交。当模拟器关闭 ,此缓存会被丢弃 ,且不会影响链上任何真实的交易数据。

在钱包打开并解锁的状态下，可以打开智能合约交易模拟器。

模拟器会根据模拟器打开当时的数据状态产生一个新的缓冲区

在模拟器中进行的交易行为（与合约相关的行为）都会被在本地验证，并将结果变化记录在这个缓冲区下，且不会被广播到区块链网络中。

当模拟器被关闭时，这个缓冲区就会被丢弃，模拟器中的所有操作也都会失效。

交易模拟器的作用：

降低调试合约的成本（模拟器中调试不需要花费任何真实链上费用）

减少错误合约上链的概率（当模拟器调试成功后，再进行正式上链，在模拟器中模拟上链的合约，一旦退出模拟器，立即失效）

缩短调试合约的时间（本地立即验证，不需要等待出块时间）

离线调试合约（不需要连入区块链 P2P 网络中，甚至不需要连入 Internet）

4.4 区块链链上事件以及链下回调

可以为合约中的事件绑定回调机制，在接收到这种类型的事件时，则触发这个回调。

我们提供了默认的 Script 脚本回调，也可以由用户根据自己的情况自定制。

合约支持事件机制

受托人执行合约，触发某个事件事件，会将其一起打入 BLOCK 中，并进行广播

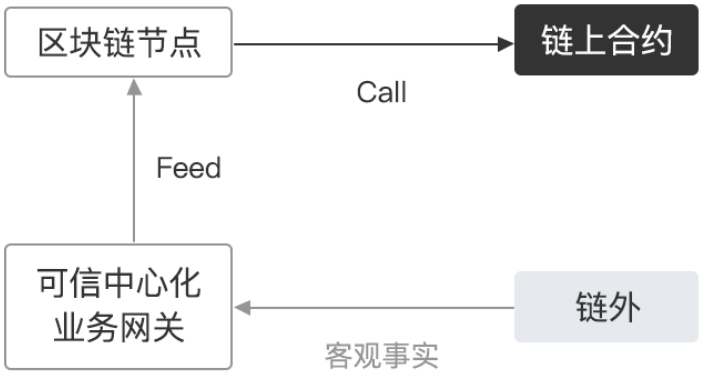
由于一个智能合约在不同的时间点或者不同的外部条件调用下，可能会走入合约代码的不同分支，执行不同的代码逻辑。对于调用者来说，并不能很好得了解合约执行的状况，有了事件机制，用户就拥有了了解合约执行中的状况，以及获取合约执行结果的能力。

拥有了这样的能力，用户可以根据接收到相应的事件，做出相关的反馈动作，比如说再次发起一笔交易，或者发起一个合约的调用，或者一些本地的动作，比如说记录日志，或者记录数据库，或者进行一个 HTTP POST 这些。甚至用户还可以制作一个具有决策能力的程序来对接到我们的区块链中，进行一些实务的决策，并根据决策结果来实施不同的反馈操作。

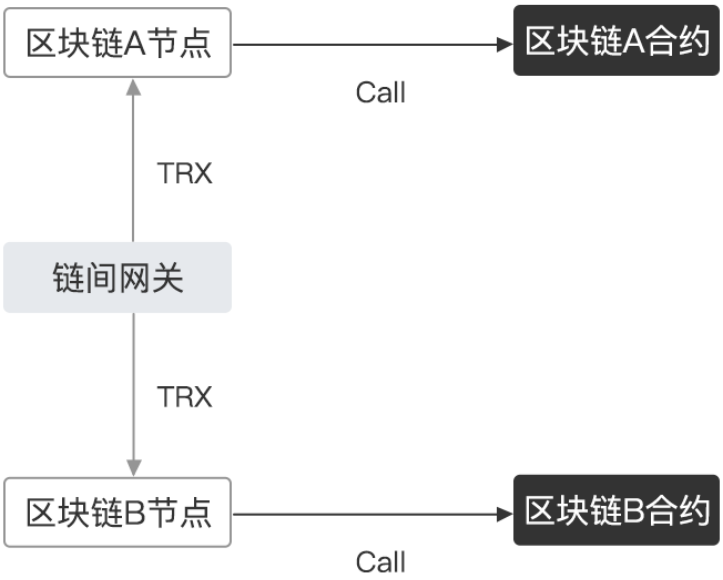
- 智能合约执行结果收集工具
- 链上彩票合约的无人值守投注
- 链上交易所自动量化交易工具
- 链上资产合约的自动汇兑工具

4.5 区块链合约网关以及区块链链间网关

合约网关：



链间网关：



不同链之间的价值交换的媒介

5 Achain 数据指标

指标	数值
普通交易 TPS	1000
合约交易 TPS	100
块大小	10m
代理数量	99
产块间隔	10s
产块时间	3s