# EOS.IO 技术白皮书

**草案：2017 年 6 月 26 日 (@dayzh (https://steemit.com/@dayzh))**

**摘要：** EOS.IO 软件引入一种新的区块链架构设计，它使得去中心化的应用可以横向和纵向的扩展。 这通过构建一个仿操作系统的方式来实现，在它之上可以构建应用程序。 该软件提供帐户、身份验证、数据库、异步通信和跨越数百个 CPU 内核或集群的应用程序调度。 由此产生的技术是一种区块链架构，它可以扩展至每秒处理百万级交易，消除用户的手续费，并且允许快速和轻松的部署去中心化的应用。

**PLEASE NOTE: CRYPTOGRAPHIC TOKENS REFERRED TO IN THIS WHITE PAPER REFER TO CRYPTOGRAPHIC TOKENS ON A LAUNCHED BLOCKCHAIN THAT ADOPTS THE EOS.IO SOFTWARE. THEY DO NOT REFER TO THE ERC-20 COMPATIBLE TOKENS BEING DISTRIBUTED ON THE ETHEREUM BLOCKCHAIN IN CONNECTION WITH THE EOS TOKEN DISTRIBUTION.**



**免责声明：** 本 EOS.IO 技术白皮书草案仅供参考。 block.one does not guarantee the accuracy of or the conclusions reached in this white paper, and this white paper is provided “as is”. block.one does not make and expressly disclaims all representations and warranties, express, implied, statutory or otherwise, whatsoever, including, but not limited to: (i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title or noninfringement; (ii) that the contents of this white paper are free from error; and (iii) that such contents will not infringe third-party rights. block.one and its affiliates shall have no liability for damages of any kind arising out of the use, reference to, or reliance on this white paper or any of the content contained herein, even if advised of the possibility of such damages. In no event will block.one or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive or special for the use of, reference to, or reliance on this white paper or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill or other intangible losses.

# 背景

区块链技术是通过 2008 年诞生的比特币货币得以被认知，自从那之后企业家和开发者就不断的尝试推广这一技术，以便在单一的区块链平台上支持更为广泛的应用程序。

而一些区块链平台努力的支持可运作的去中心化应用，具体的应用比如 BitShares 去中心化交易所 (2014) 和 Steem 社交媒体平台 (2016) 已经成为每天被成千上万活跃用户重度使用的区块链。 他们能做到这些，是通过性能的提升达到每秒处理上千交易，消除手续费和提供堪比已经存在的中心化服务的用户体验。

已存在的区块链平台承担着大量的交易费和有限的可计算能力，这都阻碍了区块链技术的大面积应用。

# 区块链应用的要求

为了赢得广泛的应用，构建在区块链之上的应用需要一个灵活性足以满足以下要求的平台：

## 支持成百上千的用户

像 Ebay、Uber、AirBnB 和 Facebook 这样企业，他们需要区块链技术能处理每日数以千万的活跃用户。 在某些情况下，除非用户群体达到一个极庞大的量级否则应用并无用武之地，因此一个可以处理极其庞大用户的平台是至关重要的。

## 免费的使用

Application developers need the flexibility to offer users free services; users should not have to pay in order to use the platform or benefit from its services. 一个可以免费供用户使用的区块链平台或许将赢得更为广泛的使用。开发者和企业可以制订有效的货币化战略。

## 简单升级和 bug 修复

企业构建区块链基础的应用需要能够为应用增加新特性的灵活性。

所有非同凡响的软件都会受到 bug 的影响，即便是经过了最严格意义上的验证。这个平台必须具有足够的鲁棒性以便应对不可避免出现的 bug。

## 低延时

一个好的用户体验需要延时时间在数秒内就能收到可靠的反馈。 高延时会阻碍用户，并且会让构建在区块链上的应用比已有的非区块链应用缺乏竞争力。

## 时序性能

一些应用因为顺序依赖关系的执行步骤而不能使用并发算法实现。 比如交易所就需要足够的时序性能来处理很高的交易量，因此高时序性能处理的平台是必须的。

## 并发性能

大型可扩展应用需要将工作量分配到多 CPU 和计算机之上。

# 共识算法 (DPOS)

EOS.IO 软件使用唯一能满足区块链之上应用性能需求的去中心化共识算法，委托股权证明 (DPOS)。 Under this algorithm, those who hold tokens on a blockchain adopting the EOS.IO software may select block producers through a continuous approval voting system and anyone may choose to participate in block production and will be given an opportunity to produce blocks proportional to the total votes they have received relative to all other producers. For private blockchains the management could use the tokens to add and remove IT staff.

EOS.IO 软件使得区块准确的每 3 秒生成一个并且在任何时间点都只有一个被授权的生产者来生成区块。 如果一个区块在规定时间之内未被生产出来则这一区块将被跳过。 当一个或多个区块被跳过发生时，在区块链中会有一个 6 秒及以上的间隔。

在 EOS.IO 软件中，区块通过 21 名生产者轮流产生。 在每一轮的开始时，21 个唯一的区块生产者被选出。 获票最高的前 20 名自动在没轮被选中，剩余的一个生产者通过得票比例选出。 被选中的生产者通过从区块取到的时间作为伪随机数来打乱其顺序。 打乱顺序是为确保这些生产者与其他生产者保持均衡的连通性。

如果一个生产者错过了一个区块并且在过去的 24 小时内没有生产任何的区块，那么它将被从候选中移除，直到它在区块链中通知它要开始再次生产区块的意图。 这样通过最小化区块丢失数量（因被证实不可靠的节点不作为导致）来确保网络操作的稳定性。

在一般情况下，一个 DPOS 区块链不会经历任何的分叉，因为区块生产者是通过合作而非竞争的方式来生产区块。 即便真的出现了分叉，共识也将自动的切换到最长的链上。 之所以会这样运作，是因为区块添加到一个区块链分叉的速率与公用同一共识的区块生产者比例是相关的。 换句话说，具有更多生产者的区块链分叉会比拥有较少生产的那一个条增长的速度更快。 而且，没有一个生产者会同时在两个分叉上同时生产区块。 如果一个区块生产者被抓到做这样的事儿，那么这个生产者将很可能被投票投出。 这些双重生产行为对应密码学凭证可以用来自动的删除这些滥用者。

## 交易确认

通常 DPOS 区块链 100% 会有区块生产者参与。一个交易从广播开始后平均 1.5 秒就可以 99.9% 被认为是确认了。

在一些特殊情况下例外，软件出现 bug，网络拥塞，或一个恶意的区块生产者制造了两个或更多的分叉。 为了确保一个交易绝对是不可逆的，一个节点可以选择等待 21 个区块生产者中的 15 个给出确认。 基于通常的 EOS.IO 软件配置，在一般情况下这需要平均 45 秒的时间。 默认情况下，所有的节点将认为当 21 个生产者中有 15 个给出确认后这一区块就是不可逆的了，并且不管长度如何都不会切换到没有这一区块的分叉。

在分叉开始的 9 秒内，一个节点就可以警告用户他们极可能正处于分叉中。 在连续丢失 2 个区块后，有 95% 的概率可以确认一个节点处于分叉中。 在连续丢失 3 个区块后就有 99% 的概率确认。 可以通过节点丢失、近期参与比率和其他参数来构建鲁棒性预测模型，从而快速的警告操作者出现了问题。

对于这种警告的反应完全取决于商业交易的性质，但最简单的做法就是等待 15/21 的确认直到警告消失。

## 股权证明的交易 (TaPoS)

EOS.IO 软件需要每一个交易包含最近一个区块头的哈希值。这个哈希值有两个目的：

1. 防止不包含区块引用的交易在分叉时重放发生；和
2. 通知网络对应的用户和他们的股份当前在某个具体的分叉上。

随着时间的推移，所有的用户直接确认区块链，在这一链条上难以伪造假的链条，因为假的链条根本无法从合法链条上迁移交易。

# 帐户

EOS.IO 软件允许所有的帐户使用一个唯一的人类可读的名称来索引，长度在 2 到 32 个字符之间。 这个名称由帐户创建者自己选择。 所有的帐户必须在创建时用极少的帐户余额来注资，从而覆盖存储帐户信息的成本。 帐户名称也支持命名空间，比如 @domain 这个帐户的拥有者是唯一可以创建 @user.domain 帐户的人。

在一个去中心化的场景中，应用开发者将会为新用户注册成本买单。 Traditional businesses already spend significant sums of money per customer they acquire in the form of advertising, free services, etc. 比起来，资助一个新的区块链帐户的花费简直微不足道。 值得庆幸的是，对一个已经在另一个应用注册过的用户并不需要再创建新的帐户。

## 消息 & 处理

每个帐户可以发送结构化的消息给其他的帐户，并且可以定义脚本来处理他们接收到的消息。 EOS.IO 软件给每个帐户提供了只有自己的消息处理脚本能访问的私有数据库。 消息处理脚本同样可以给其他帐户发送消息。 消息和自动化的消息处理的结合决定了 EOS.IO 如何定义智能合约的。

## 基于角色的权限管理

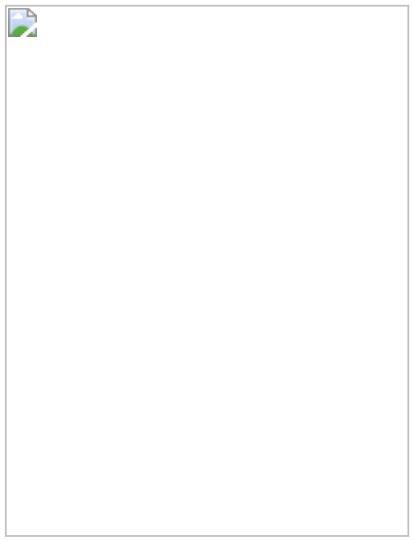
权限管理涉及判定一条消息是否被正确的授权。 权限管理最简单的形式就是检查一个交易包含必须的签名，但这意味着必须的签名是已知的。 一般情况下，权威必然是独立的个体或者个体组成的群体，并且是被划分开的。 EOS.IO 软件提供了声明式的权限管理系统，通过管理谁可以在什么时间做什么来给用户细力度和高维度的控制。

授权和权限管理被标准化和脱离应用的商业逻辑是不可取的。 这使得管理权限的工具得以被开发，既满足常规的需求又为性能优化提供了重要的可能性。

每一个帐户可以被任何权重组合的其他帐户和私钥管控。 这创建了分层级的权利结构，这反映了现实中的权限分配方式，并且让多用户共同管理资产变得从未如此简单。 多用户控制是安全最大的贡献者，并且，当用户使用得当，它可以极大的消除因被黑而导致被盗窃的风险。

EOS.IO software allows accounts to define what combination of keys and/or accounts can send a particular message type to another account. 举个例子，可以指定一个密钥给一个用户的社交媒体账号，同时另一个密钥访问交易所。 甚至可以给其他帐户权限来代表自己而无需分配给他们密钥。

### 命名的权限级别



在 EOS.IO 软件中，帐户可以定义命名的权限级别，每一个是由更高级别的命名权限派生而来。 每一个命名的权限级别定义了一个权威；一个权威是多重签名阈值校验，它包含密钥和 / 或其他帐户的命名权限级别。 打个比方，一个帐户的"朋友"权限级别可以被设置为由该帐户的任何一个朋友无差别的控制。

另一个例子在 Steem 区块链中，它包含三个硬编码的命名权限级别：拥有，活跃和发帖。 发帖权限就只能进行如投票和发帖的社交活动，而活跃权限可以做除了变更拥有之外的所有的事情。 拥有权限的意思是冷存储并且有能力做任何事。 The EOS.IO software generalizes this concept by allowing each account holder to define their own hierarchy as well as the grouping of actions.

### 命名的消息处理群组

EOS.IO 软件允许每个帐户将他们自己的消息组织到一个命名和嵌套的群组中。 这个命名的消息处理群组可以在其他帐户配置他们权限级别时被引用。

最高级别的消息处理群组是帐户名称，最低级别的是一个帐户接收到的单独的消息类型。 这些群组可以被这样的方式引用： **@accountname.groupa.subgroupb.MessageType**.

在这样的模型之下，交易所合约可以通过将挂单的创建和取消分组，从而与充值提现分离开。 交易所合约的这样分组对用户而言带来了方便。

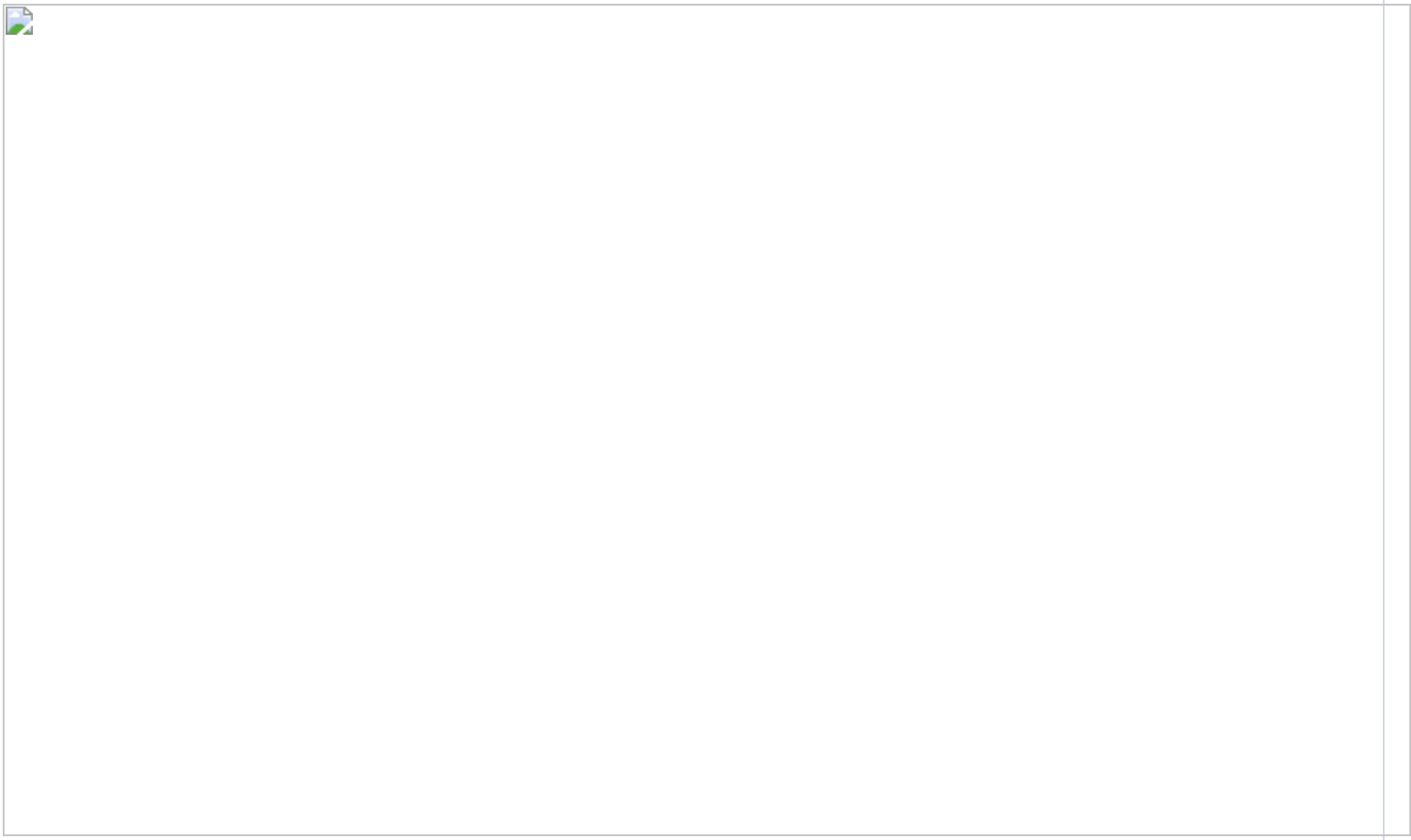
## 权限映射

EOS.IO 软件允许每个帐户定义从任意帐户的一个命名的消息处理群组与自己的命名的权限级别之间建立映射。 举个例子，一个帐户所有者可以将自己社交媒体应用与自己的"朋友"权限群组建立映射。 有了这个映射，任何朋友可以以这一帐户的身份在这一帐户的社交媒体上发帖。 尽管他们将以帐户所有者的身份发帖，他们仍然使用自己的密钥来签名消息。 这意味着总是可以辨识出是哪一个朋友在以何种方式使用帐户。

## 评估权限

当 **@alice** 以 "**Action**" 类型发送一条消息给 **@bob** 时，EOS.IO 软件首先会检查 **@alice** 是否为 **@bob.groupa.subgroup.Action** 定义过权限映射。 如果什么都没有找到，紧接着检查 **@bob.groupa.subgroup** 映射，然后是 **@bob.groupa**，最后 **@bob** 将被检查。 如果都没有找到，那么假定映射为命名的权限群组 **@alice.active**。

一旦一个映射被识别，则通过阈值多签名流程验证签名权威，并且关联权威与命名的权限。 如果失败了，则跃迁至父权限，直至拥有者权限，**@alice.owner**。



### 默认权限群组

The EOS.IO technology also allows all accounts to have an "owner" group which can do everything, and an "active" group which can do everything except change the owner group. 所有其他的全新群组派生自"活动"群组。

### 权限并行评估

权限评估过程是"只读"的，并且通过交易对权限的变更在一个区块结束之前不会起作用。 这意味着对所有的交易对应的密钥和权限评估可以被并行执行。 此外，这意味着一个快速的权限验证是可行的，它无需启动会引起回滚需求的高成本的应用逻辑。 最后，这意味着交易权限可以被评估即便接收到等待的交易，并且之后无需再重新评估。

从各方面考虑，权限验证占据了验证交易计算量的很大比例。 让其只读和普遍的并发处理将会使得性能有一个质的飞跃。

当从消息日志中重新生成确定性状态时不再需要重复的权限验证。 事实是一个交易如果被包含近了一个被认为不存在问题的区块时它就有足够的理由跳过这 步这将极大减少因为区块链增长拉去过去记录时的计算量。

## 带强制性延时的消息

时间是安全中的一个关键组成部分。 在大多数情况下，一个私钥在没有被使用前都无从知晓它是否被偷窃。 当人们有需要密钥的应用在每天联网使用的电脑上运行时，基于时间的安全会更为重要。 EOS.IO 软件让应用开发者可以指明消息必须在被加到一个区块之前等待最小的时间间隙。 During this time they can be cancelled.

用户可以在消息广播出去后通过邮件或者文字消息的形式收到通知。 如果他们没有授权，那么他们可以使用帐户恢复流程来恢复帐户，并收回消息。

这个必须的延时由操作敏感性决定。 为一杯咖啡付款可以没有任何的延时，几秒之内就不可逆了，而购买一个房子也许需要 72 消失的结算期。 转移整个帐户到一个新的控制可能需要长达 30 天。 具体的延时选择由开发者和用户自己来做选择。

## 恢复被盗窃的密钥

EOS.IO 软件提供给用户一种找回自己失窃密钥控制权的方式。 一个帐户的所有者可以使用过去 30 天任何活跃的拥有者密钥与事先指定的合作者帐户给出的批准来重置自己帐户的密钥。 帐户的恢复合作者在没有所有人帮助的情况下无法重置帐户的控制权。

黑客尝试进行恢复流程是无意义的，因为他们已经"控制"了帐户。 此外，就算他们真的进行这一流程，恢复合作者也会询问身份证明和多因素认证 (手机和邮件)。 这会让黑客脱作出让步或者无功而返。

这一流程与简单的多重签名有很大差异。 在多重签名中，另一个公司要参与所有转账的执行，但在恢复流程中，它却只在恢复时才起作用对每天的转账无从干预。 这大大的降低了参与者的成本和法律责任。

# 应用程序的确定性并行执行

区块链共识取决于确定性 (可重现的) 的行为。 这意味着所有的并行计算必须是不能互斥或者具有其他锁特性的。 没有了锁就必须有一些方式可以确保所有的帐户只可以读取和写入他们自己的私有数据库。 这也意味着每个帐户处理消息是顺序的，而并发只能在帐户层面进行。

In an EOS.IO software-based blockchain, it is the job of the block producer to organize message delivery into independent threads so that they can be evaluated in parallel. 每个帐户的状态由且只由发送给它的消息决定。 进度表由区块生产者输出并且会被确定性的执行，但是生成进度表的过程却不一定是确定性的。 这意味着区块生产者可以使用并发算法来调度交易。

并行执行的一方面意味着当一个脚本生成了一个新的消息，它不会立即被发送，而被安排在下一个轮训中发送。 不能立马发出的原因是接受者可能在另一个线程中活跃的变更自己的状态。

## 最小化通信延迟

延迟是一个帐户从发出一条消息给另一个帐户，直到收到回应的这段时间。 我们的目标是在一个单独的区块中包含两个帐户交换消息的来去信息，而不用在每条消息间等待 3 秒钟。 为了做到这一点，EOS.IO 软件将每个区块划分为循环。 每个循环划分为线程，每个线程包含了交易的一个列表。 每一个交易包含了待发送的消息集合。 这个结构可以被可视化为一个树，其中交互层彼此并行，各自被顺序的执行。

```
区块
  循环（顺序）
    线程（并行）
      交易（顺序）
        消息（顺序）
          接受者和被通知帐户（并行）
```

在一个循环中生成的交易可以在后续的任何一个循环或者区块中被发送。 区块生产者会持续不断的向区块中添加循环直到最大的墙上时间到了或者没有更多的新交易要发送。

可以对一个区块使用静态分析来验证同一个循环内不存在两个线程包含同一帐户下对交易的变更。 只要保持不变一个区块就可以并行的运行所有的线程。

## 只读消息的处理

有些帐户可以在传递/失败的基础上处理消息而不修改内部状态。 如果是这样的话，那么这些处理程序可以并行执行，只要只有一个特定的帐户的只读消息处理程序包含在一个或多个线程在一个特定的周期。

## 多帐户的原子化交易

有时我们需要确保消息自动的被多个账户传递和接收。 在这种情况下，消息会被放在同一个交易内，账户会被分配到同一个线程，并且消息被顺序的添加。 这种情况对性能是不理想的，当用户使用涉及到"账单"时，他们将在交易内以账户唯一索引被列入其中。

基于性能和成本原因最好减少涉及两个或多个重度帐户的原子性操作。

## 区块链状态的部分评估

扩展区块链技术使得组件化成为必要。每个人不应该执行所有的事务，尤其是当其只需要运行应用的一个小的子集。

一个交易所应用开发者运行一个完整节点位的是为其用户展现所有的状态。 这个交易所应用没有与社交网络建立关联的必要性。 EOS.IO 软件允许任何的完整节点选择应用的任何子集来执行。 传递给其他应用的消息可以被安全的忽略掉，因为应用程序的状态完全由传递给它的消息派生。

这与其他帐户的沟通有一些重要的影响。 最重要的是，不能假定其他帐户的状态可以在同一台机器上访问。 这也意味着，虽然很容易启用"锁"来允许一个帐户同步调用另一个帐户，如果其他帐户不驻留在内存中，这种设计模式就会出现问题。

所有账户帐户间的状态通信必须通过包含在区块链中的消息进行。

## 自主最优调度

EOS.IO 软件并不能为区块生产生者为任何其他帐户送达的任何信息负责。 每个区块生产者要对计算的发杂读和处理一个消息的时间自己进行主观上的预测。 这同时适用于用户生成的和脚本自动生成的交易。

On a launched blockchain adopting the EOS.IO software, at a network level all transactions are billed a fixed computational bandwidth cost regardless of whether it took .01ms or a full 10 ms to execute it. 然而，每个单独的区块生产者要通过自己的算法来计算资源的消耗。 当一个区块生产者断定一个交易或者帐户消耗了不相称的大量的计算资源时，他们可以在生成自己的区块时拒绝该交易；但是，如果其他区块生产者认为交易是有效的，他们就仍需要处理交易。

一般而言，只要一个区块生产者认为交易在资源使用限度内是有效的，那么其他区块生产者就也要接受，但可能交易传递给生产者就要花费 1 分钟。

在某些情况下，生产者可以创建包含可接受范围之外的数量级的块。 在这种情况下，下一个区块生产者可能会选择拒绝区块和束缚将被第三个生产者打破。 这和因为区块过大导致的网络延时没什么打不同。 社区会注意到模式的异常并最终会将票从流氓生产者哪里删掉。

这种对计算成本的主观评估将区块链从必须精确和确定的预测一些东西要花多长时间来运行这一问题中解放出来。 有了这一设计就不需要精确的数指令，将极大的增加优化的可能性又不必打破共识。

# Token 模型与资源使用

**PLEASE NOTE: CRYPTOGRAPHIC TOKENS REFERRED TO IN THIS WHITE PAPER REFER TO CRYPTOGRAPHIC TOKENS ON A LAUNCHED BLOCKCHAIN THAT ADOPTS THE EOS.IO SOFTWARE. THEY DO NOT REFER TO THE ERC-20 COMPATIBLE TOKENS BEING DISTRIBUTED ON THE ETHEREUM BLOCKCHAIN IN CONNECTION WITH THE EOS TOKEN DISTRIBUTION.**

All blockchains are resource constrained and require a system to prevent abuse. With a blockchain that uses EOS.IO software, there are three broad classes of resources that are consumed by applications:

1. 带宽和日志存储 (磁盘)；
2. 计算与计算储备 (中央处理器)；
3. 状态存储 (内存)。

带宽和计算有两部分，瞬时使用和长期使用。 一个区块链维持着所有消息的日志，这些日志最终由完全节点存储和下载。 通过消息日志可以重现所有应用的状态。

可计算债务是一个必须通过消息日志重新构建状态的计算结果。 如果可计算债务增长变得臃肿则有必要通过快照方式记录区块链状态，并丢弃区块链历史。 如果可计算债务增长过快，则它需要花费 6 个月时间来重放等值与 1 年的交易。 这很不可取，因此，可计算债务需要被细心的管理。

区块链状态存储是通过访问应用逻辑获取的信息。 它包括诸如挂单和账户余额等信息。 如果状态从未被应用读取则它不会被存储。 比如，博客发布的内容和评论如未被应用逻辑读取则他们就不应该存储在区块链状态中。 同时，发布的内容 / 评论的存在、投票的数量和其他属性要作为区块链状态的部分被存储下来。

区块生产者对外发布她们可用的带宽，计算能力和状态。 EOS.IO 允许帐户按比例消耗一个 3 天对赌合约中的可用资源。 举个例子，如果一个基于 EOS.IO 的区块链启动了，一个帐户持有所有 token 发行总量的 1%，那么帐号就具有使用 1% 状态存储空间的能力。

Adopting the EOS.IO software on a launched blockchain means bandwidth and computational capacity are allocated on a fractional reserve basis because they are transient (unused capacity cannot be saved for future use). The algorithm used by EOS.IO software is similar to the algorithm used by Steem to rate-limit bandwidth usage.

## 客观与主观的度量

如前所述，检测计算使用的性能和优化的影响很大；因此，所有资源的使用限制，最终都是主观的，执行依靠个人的算法和区块生产者进行估计。

也就是说，有一些事情是微不足道的客观衡量。 发送的消息数和存储在内部数据库中的数据的大小是便宜的客观衡量。 的 EOS.IO 软件让区块生产者采用相同的算法应对客观的量，但可以在主观量上选择采用更严格的主观测量算法。

## 接收方付费

传统上来说，企业为办公场地、计算力和其他为了运行企业而需要的成本买单。 客户从企业购买具体的产品，产品销售产生的利润来盖过企业运作的成本。 类似的，没有哪个网站要求来访者为盖过运作成本而支付。 因此，去中心化应用也不应该强制用户因为使用了区块链而直接为区块链支付。

A launched blockchain that uses the EOS.IO software does not require its users to pay the blockchain directly for its use and therefore does not constrain or prevent a business from determining its own monetization strategy for its products.

## 委托能力

A holder of tokens on a blockchain launched adopting the EOS.IO software who may not have an immediate need to consume all or part of the available bandwidth, can give or rent such unconsumed bandwidth to others; the block producers running EOS.IO software on such blockchain will recognize this delegation of capacity and allocate bandwidth accordingly.

## 分离交易成本与 Token 价值

EOS.IO 软件的一个主要优点就是应用可用的带宽完全独立于 token 的价格。 If an application owner holds a relevant number of tokens on a blockchain adopting EOS.IO software, then the application can run indefinitely within a fixed state and bandwidth usage. In such case, developers and users are unaffected from any price volatility in the token market and therefore not reliant on a price feed. In other words, a blockchain that adopts the EOS.IO software enables block producers to naturally increase bandwidth, computation, and storage available per token independent of the token's value.

A blockchain using EOS.IO software also awards block producers tokens every time they produce a block. Token 的值将影响其能购买的带宽、存储和计算资源；这一模型会自然的利用 token 值的上涨来增加网络的性能。

## 状态存储成本

由于带宽和计算资源可以被委托，因此应用的状态存储需要应用程序的开发者持有 token 直到状态被删除。 如果状态永远不会被删除那么 token 实质上从流通中被抹除。

每一个用户帐户需要一个确定数量的存储；因此每一个帐户必须保持一个最小的余额。随着网络存储能力的不断提升，余额的最小余额需求将会下降。

## 块奖励

A blockchain that adopts the EOS.IO software will award new tokens to a block producer every time a block is produced. In these circumstances, the number of tokens created is determined by the median of the desired pay published by all block producers. EOS.IO 软件可以配置限定生产者回报的上限从而确保 token 的每年增长比例不会超过 5%。

## 社区效益应用

In addition to electing block producers, pursuant to a blockchain based on the EOS.IO software, users can elect 3 community benefit applications also known as smart contracts. 这三个应用将接收至多一个按照配置百分比对应的 token 年供应量减去每年提供给区块生产者的 token 量。 这些智能合约将按照每个应用接收到的 token 持有者的票的比例对应的 token。 这些应用或者智能合约可以被 token 持有者选出的新的应用或智能合约所替代。

# 治理

治理是人们在主观问题上达成共识的过程，而这无法完全用软件算法来捕获。 An EOS.IO software-based blockchain implements a governance process that efficiently directs the existing influence of block producers. 没有了定义好的治理流程，之前的区块链依赖临时的、非正式和常常充满争议的方式治理，直接导致不可预知的结果。

A blockchain based on the EOS.IO software recognizes that power originates with the token holders who delegate that power to the block producers. 区块生产者被授予有限的检查权威来冻结帐户，升级有缺陷的应用程序，对底层协议提出硬分叉的改进建议。

Embedded into the EOS.IO software is the election of block producers. 在对区块链没有做任何变更之前他们必须认可它。 如果区块生产者拒绝 token 持有者所预期的变更他们就会被投出。 如果区块生产者未经 token 持有者的授权作出变更，其他的非生产、完整验证 (交易所等) 会拒绝这些变更。

## 冻结帐户

有时一个智能合约的行为处于一种一场或不可预测的状态并且无法按照预期执行；另一些时候一个应用或帐户也许发现了一个可以销毁不可想像数量资源的漏洞。 当这些问题不可避免的发生时，区块生产者有能力来扭转这一局面。

所有区块链上的区块生产者都有能力来决定哪些交易被加到区块中，这给了他们冻结帐户的能力。 A blockchain using EOS.IO software formalizes this authority by subjecting the process of freezing an account to a 17/21 vote of active producers. 如果生产者滥用权利他们会被投出，而对应冻结帐户就将解冻。

## 更改帐户代码

When all else fails and an "unstoppable application" acts in an unpredictable manner, a blockchain using EOS.IO software allows the block producers to replace the account's code without hard forking the entire blockchain. 与冻结一个帐户类似，更改帐户代码需要 17/21 这样的生产者票形。

## 宪法

EOS.IO 应用使得区块链创建了一个点对点的服务条款协议或者绑定用户到一个合约，这都需要用户对其签名，简称“宪法”。 宪法的内容定义了仅仅依靠代码无法在用户间履行的义务，同时通过建立管辖权和可选的法律来解决相互间的争端。 每个在网络广播的交易都必须将宪法的哈希值作为签名的一部分，从而显性的将签名者绑定在合约中。

宪法还定义了人类可读意图的源代码协议。 这个意图是用来识别错误和功能之间的差异，当错误发生时，引导社区对什么是适当或不当修复。

## 升级协议 & 宪法

The EOS.IO software defines a process by which the protocol as defined by the canonical source code and its constitution, can be updated using the following process:

1. 区块生产者对宪法提出改建意见并获得 17/21 批准。

2. 区块生产者持续 17/21 品准连续 30 天。
3. 所有用户需要使用新的宪法来做签名。
4. 区块生产通过变更代码的方式来影响宪法并且提交一个 git 记录的哈希值。
5. 区块生产者持续 17/21 品准连续 30 天。
6. 7 天后改为会起影响的代码，给所有完整节点 1 周时间在确认源码后进行升级。
7. 所有未升级到最新代码的节点被自动关掉。

按照 EOS.IO 的默认配置，添加新特性升级区块链的流程需要 2 到 3 个月，而修复一般的 bug 不需要更改宪法需要 1 到 2 个月时间。

### 紧急变更

区块生产者可以推荐软件的变更当 bug 是伤害性 bug 或安全溢出影响用户使用的。 一般来说，这可能是对宪法的加速更新，引进新的功能或修复无害的错误。

# 脚本 & 虚拟机

EOS.IO 首先会是一个平台用于协同用户间认证消息的传递。 脚本语言和虚拟机的具体实现与 EOS.IO 技术的设计是分离的。 任何语言或者虚拟主机，只要确定并适合沙盒，带有足够的运行效率均可以和 EOS.IO 软件 API 对接。

## 模式定义的消息

所以用户间发送的消息都是通过模式定义定义出来的，它是区块链共识状态的一部分。 这个模式允许消息在二进制与 JSON 格式之间无缝的转换。

## 模式定义的数据库

数据库状态也是通过类似的模式来定义。 这是为了确保所有应用存储的数据是可以转化为人类可读的 JSON 但存储和控制时使用高效的二进制。

## 分离授权与应用

To maximize parallelization opportunities and minimize the computational debt associated with regenerating application state from the transaction log, EOS.IO software separates validation logic into three sections:

1. 验证消息是否内部一致；
2. 验证所有前提条件是否有效；
3. 修改应用程序状态。

验证消息的内部一致性是只读的并且无需访问区块链状态。 这意味着它可以以最大并发来执行。 验证前提条件，比如需要的余额数，是只读的因此也可以受益与并行计算。 只有更改应用状态时需要写入权限并且必须顺序的执行每个应用。

身份认证是一个验证消息可被使用的只读过程。 应用程序实际上在发挥作用。 同一时间两者都需要被计算，然而一旦消息被包含进区块它就不再需要进行消息验证的操作了。

## 虚拟机独立架构

It is the intention of the EOS.IO software-based blockchain that multiple virtual machines can be supported and new virtual machines added over time as necessary. 因此，本文并不讨论任何特定的语言或者虚拟机。 That said, there are two virtual machines that are currently being evaluated for use with an EOS.IO software-based blockchain.

### Web 组建 (WASM)

网络组建是一种为了构建高性能的 web 应用而新兴的 web 标准。 只需要进行少量的更改 Web 组建就可以被制作为确定性的和沙盒化的。 Web 组建的好处是它有着广泛的产业支持并且它可以让智能合约使用熟知的语言进行开发，比如 C 或 C++。

以太访开发者已经开始更改 Web 组建来提供合适的沙盒与确定性在他们的以太访式 Web 组建 (WASM)。 这种方式让 EOS.IO 很容易的与之适配和对接。

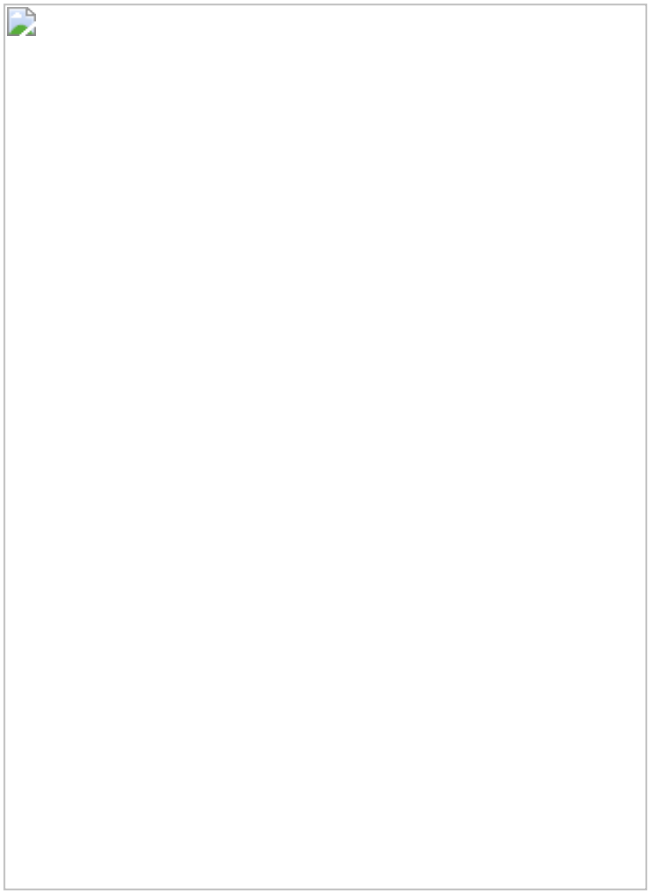
### 以太访虚拟机 (EVM)

这个虚拟机已经被众多已有的智能合约所采用并且可以通过适配应用与 EOS.IO 区块链中。 It is conceivable that EVM contracts could be run within their own sandbox inside an EOS.IO software-based blockchain and that with some adaptation EVM contracts could communicate with other EOS.IO software blockchain applications.

# 跨链通信

EOS.IO 软件被设计为跨区块链通信友好的。 这是通过生成消息存在证明与消息时序证明变的简单而实现的。 这些证明与应用架构设计相结合，即围绕消息细节的跨链传输和有效性验证时隐藏应用程序开发者的架构设计。

## 用于轻客户端的 Merkle 证明 (LCV)



如果客户端不需要处理所有的交易会让多区块链间的整合更为轻松。 毕竟，一个交易所只需要关心交易所的入账和出账，别无他求。 如果交易所链条可以使用资金的轻量 merkle 证明，而不必非要完全依赖对它区块生产者的信任会是一个不错的主意。 至少一个链的区块生产者在与其他区块链同步时更乐意保持尽可能小的开销。

LCV 的目标能产生相对轻量存在性证明，使得任何追踪相对轻量数据集的人可以验证其有效性。 在这种情况下，目的是为了证明一个特定的交易是包含在一个特定的区块中，区块包含在一个特定的区块链的已验证历史中。

比特币支持通过全节点的完整记录获取每年 4MB 大小的区块头信息来验证交易。 每秒 10 个交易，一个有效的证明需要 512 个字节。 这对于有 10 分钟间隔的区块链没有问题，但是对于 3 秒间隔区块链就显得不那么“轻量”了。

EOS.IO 软件使得任何一个人只要他拥有包含交易所对应区块之后的随意一个不可逆的区块头，他就可以进行轻量证明。 使用下面展示的哈希链结构就可以使用少于 1024 字节的大小来完成任意交易的存在性证明。 如果假设校验节点在过去几天内所有的区块头一直增长 (2MB 的数据)，那么验证这些交易将只需要 200 字节就够了。

将生产的区块与恰当的哈希链做关联使得开销增幅很小，这意味着没有理由不使用这种方式来生成区块。

当需要验证其他链时，有譬如 时间/ 空间/ 带宽 的多样化优化可以做。 追踪全部区块头 (420 MB/年) 将保持证明体积的轻巧。 只追踪最近的头可以提供最小长期存储和证明大小来获得。 另外，一个区块链可以使用懒惰的评估方法，即它记住过去证明的中间值哈希。 新证明只需要包含指向已知稀疏树的链接。 确切的方法将取决于那些包含对 Merkle 证明引用的交易所在的外部区块的比例。

一定密度的联系后，将变得更为高效，一个链会包含另一个链整个区块的历史和消除证据一起，这样就不需要通信便可以验证了。 出于性能原因，应最小化的跨链证明的频率。

## 跨链通信的延时

当与外部区块链进行通信时，区块生产者必须等待直到 100% 确信一个交易已经被另一个区块链确认为不可逆后才会接收它成为一个有效的输入。 Using an EOS.IO software-based blockchain and DPOS with 3 second blocks and 21 producers, this takes approximately 45 seconds. If a chain's block producers do not wait for irreversibility it would be like an exchange accepting a deposit that was later reversed and could impact the validity of the blockchain's consensus.

## 完备性证明

当使用来自外部区块链的 Merkle 证明时，在已知所有交易均已验证和已知没有交易被跳过或遗忘之间有一个重要的差异。 虽然不可能证明所有最近的交易是已知的，但有没有间隙的交易历史是可以被证明的。 EOS.IO 软件在每个用户的每个传递的消息上分配了一个序列号。 一个用于可以使用这些序列号来证明所有的消息由某个特定帐户处理，只需要看它是否是按序执行的。

# 总结

EOS.IO 软件是从证明概念的经验和最佳实践设计而来，它代表了区块链技术的重要进步。 该软件是全球可扩展区块链社会伟大蓝图中的一部分，它将应用去中心化并得以轻松的发布和治理。