白皮书

版本 0.5

LALIN Hugo

contact@virgocoin.io

Virgo:一种高性能的加密货币

创建去中心化应用程序。

目录

介绍

自区块链兴起后，已证明其能够在无可信第三方的情况下促使经济更具流动性。
区块链有机会改变许多需要信任的行业，如金融、数据管理甚至是供应链。
这项技术本应成为这些行业的标准，但由于其复杂性，很难在行业中站稳脚跟。
事实上，目前存在的各区块链与项目数量一样多，从而削弱了其优势。
区块链虽然能提高流动性交换，但必须根据自身使用情况不断持有、交换大量货币，这样一来就变得非常复杂且费劲。
与此同时，我们需要信任众多不同的区块链技术和参与者，是他们构成了如此多的项目。信任问题从而加剧了矛盾。
围绕“区块链三元悖论”，即：可扩展性、安全性和去中心化，许多相关的解决方案应运而生。

所有这些都意味着，许多配备了创新性解决方案的项目正试图让更多人接受。但大多数时候，公众对加密领域了解得不多，从而和该领域有一定的隔阂。

Virgo 的主要关注点不只在于加密货币，而是为开发人员提供最佳工具，来创建去中心化应用程序，改变应用程序的分布。
针对区块链三元悖论，这些工具主要由高性能加密货币组成。在接下来的步骤中，以 AppStore 的风格构建易于使用且完全分散的 dApps 分发平台。

Virgo 账本使用 DAG（有向无环图）作为数据结构，提供达成一致性的方法，引入一种新类型的交易，在 DAG 上方形成一条链，沿同一方向进行交易。
结合 DAG 和“传统区块链”，Virgo 成功地将这两个领域的优点融合在一起，提供一种方法，让可扩展性、安全性和去中心化达成一致。

应用程序成为了我们数字生态系统的重要组成部分。
随着其使用频率的增加，关键问题如可用性、吞吐量、软件真实性保证、知识产权保护和应用程序开发人员权利等，已对该生态系统产生重大影响。
在我们的白皮书中，我们呈现了首个去中心化分布式应用程序分发平台，使用分布式文件系统解决上述问题。

应用程序的“存储”自动运行、易于使用，可弥补目前凸现的一些主要弊端。
该技术可消灭所有第三方中介角色，创建一个新的生态系统：
用户获取的 Virgocins(VGO)一经存储于钱包中，就可直接用于购买服务、使用应用程序。
因通过综合广告或被动挖掘（用户的个人电脑将在不使用时被挖掘）分发 VGO，一些应用程序可以免费使用，甚至可以获得回报。
在我们的模式下，一旦用户获得 virgocoins，便可把钱花在自己喜欢的应用程序上，创造循环经济。

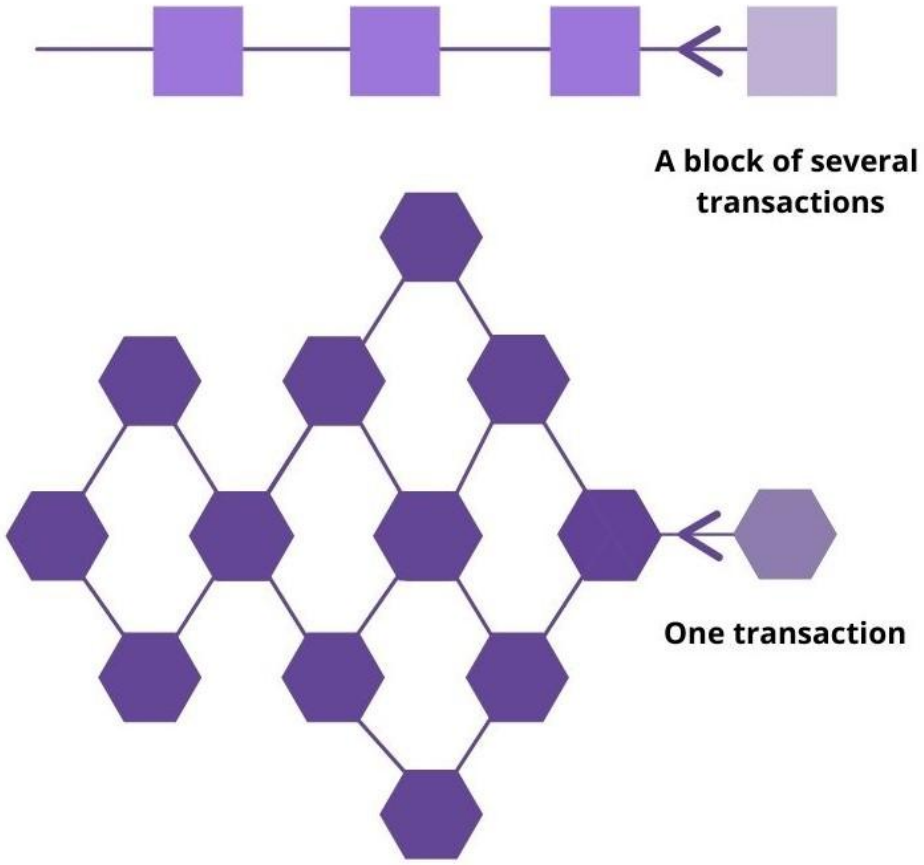
还应注意的是，目前现有应用程序的审批流程非常复杂，由集中式架构进行管理。该结构具备复杂的质量保证流程和不透明的分发策略。通过使用 Virgo，应用程序审批将变得更加普遍化和透明化。
为保证应用程序验证过程去中心化，同时也保护用户，任何人都可以创建、维护一个已认可应用程序的名单和该名单的用户评论区。默认将热门名单提供给用户，但用户可导入任意名单以扩展其可选项。

I - VIRGO 账本

1 / 操作

1.1 / 有向无环图

不同于大多数的加密货币，VIRGO 并非基于区块链，而是基于有向无环图（或 DAG），可以将其视为二维区块链。



如上图所示，主要区别在于每个顶端单元（理解区块）可以链接到多个“父母”单元（此前出现过），而不是链接到传统区块链上的一个区块。

在像比特币这样的传统加密货币中，将交易分组成区块就可以给它们下命令，通过挖矿，固定时间就能生成区块，每个区块都指向前一个区块，整个区块便生成一条链。

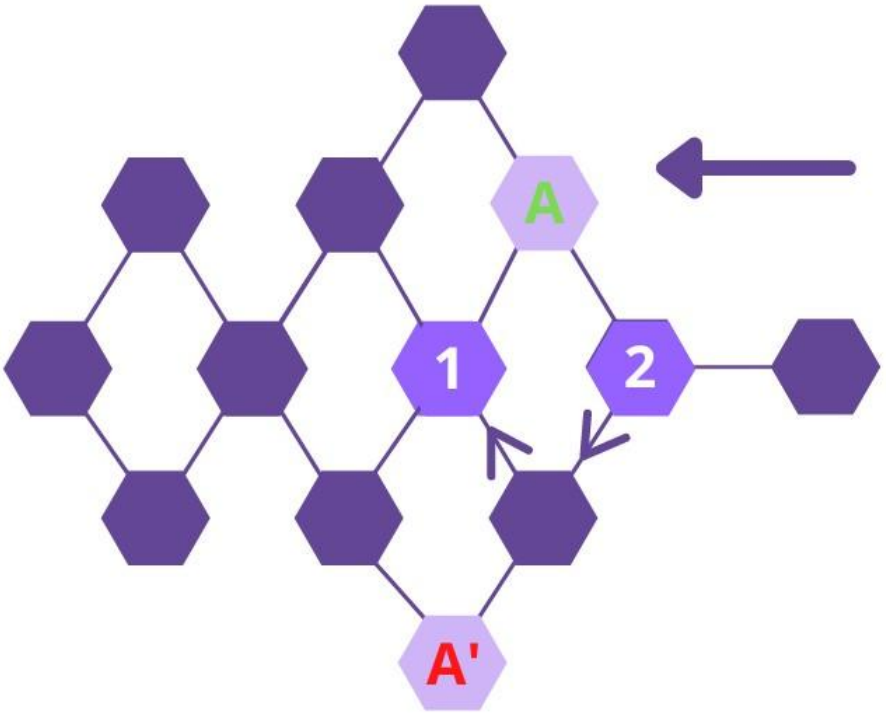
因此，我们通过查看各自区块的顺序，在另一笔交易前验证前一笔交易，就能按照正确的顺序进行计数，花相同的钱完成交易。

这些区块须有足够间距且间距要小，这样一来，网络中的大多数计算机有时间接收最近的一个区块，否则网络两端接收到不同的区块时，区块链就形成两个不同版本，出现“分叉”。这限制了网络每秒可管理的交易数量，

如果达到这一限制，则首选费用最高的交易。这样一来，发送交易就变得非常昂贵（例如，比特币已经超过 70 美元）

有了 Virgo，图表可以让我们摆脱区块。现在，每个顶端单元都是一项交易，自发形成一定顺序，因为每项交易都涉及此前一项或多项交易（图表就是这样生成的，甚至是如图 2 那样），因此它们不再需要同时到达网络中的所有计算机。

但使用 DAG 会产生一个新问题：交易不一定有明确的顺序。



例如，在这个图中，我们通过回顾图表得知交易 2 转到交易 1，从而确定交易 1 是先发出的。但是在 A 和 A '之间无路径，所以我们无法得知哪项交易是首先发出的

如果不对 A 和 A'投入相同的资金，也不是什么问题。但如果情况如此，则有必要确定好它们的顺序。不能随意决定，因为链中会产生分叉，也不能简单拒绝这两项交易，否则任何人都可以取消付款、通过发行一项与旧交易竞争的新交易取回他们的钱。

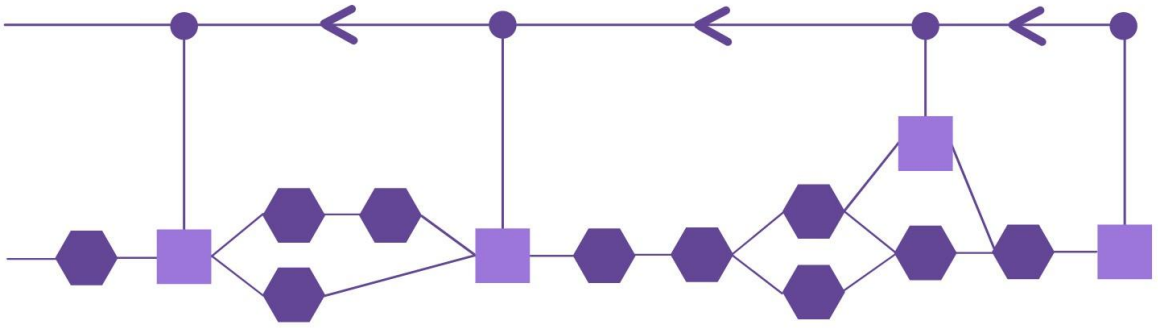
1.2 / 信标链

为了解决该问题和日期交易的问题，我们引入了一种新的交易类型：信标。
信标交易通过引用一个或多个父母交易（如正常交易）与 DAG 合并在一起；除此之外，其指向之前的信标交易。
通过这种方式，我们获得了覆盖整个 DAG 的信标交易链，类似于传统区块链。

为了保持 Virgo 的去中心化，任何人都可以发出信标；但有必要为解决方案的剩余部分保留一种形式的链条；因此，我们将实施类似于比特币那样的工作证明机制：
为了促成有效的信标交易，其哈希值必须小于某个值。

最小要求值与难度成反比，该值由算法进行调整。考虑到最后两个信标间的时间，因此，即使分配给任务的算力不同（设备速度、参与者数量），信标之间的间距也是恒定的。

若是想创建新信标，必须给信标增加一个称为 nonce 的任意值，直到其哈希值符合要求为止，与网络的其他部分相比，就其算力而言，需要一定时间。



可能会在短时间内找到多个指向相同父母单元的信标，尤其是在网络过载的情况下。
在这种情况下，被认为有效的信标是那些形成链条的信标，它代表了最有效的工作证明（因此困难重重）。

每个有效信标都会确认它与它所属的有效信标间的交易。

为此，我们将从正在确认的信标交易开始，从一项交易到另一项交易，直到交易满足验证条件；每项交易，我们都采用如下确认逻辑：

1. 如果交易已由另一个信标处理，请在此停止。

2. 如果交易试图花费已支付过或已被拒绝的费用，节点会拒绝交易并与其父母交易一起继续进行。

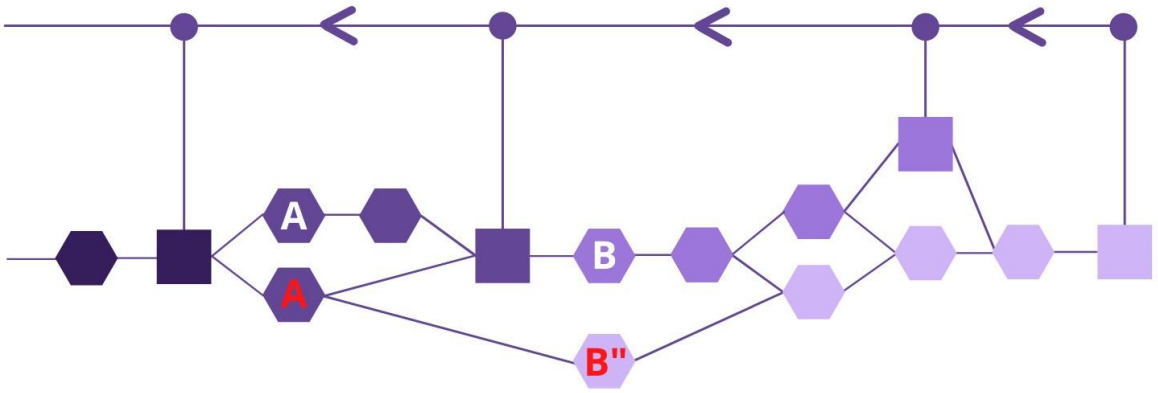
3. 检查另一项未决交易是否需要花费相同的资金，如果是，将该交易添加到冲突交易列表中，并继续其父母交易。

4. 如果交易未触发之前的验证，请确认交易并继续父母交易。

处理完所有交易后，对每个冲突交易执行以下逻辑：

1. 如果交易与该信标确认的一项或多项交易冲突，则拒绝该交易（因此也将拒绝最终的竞争对手）

2. 否则进行确认，因为并发交易仅由后续信标处理（因此将被拒绝）。



例如，在这里，A 和 A ”同时存在，且都被拒绝，因为它们经相同的信标确认。

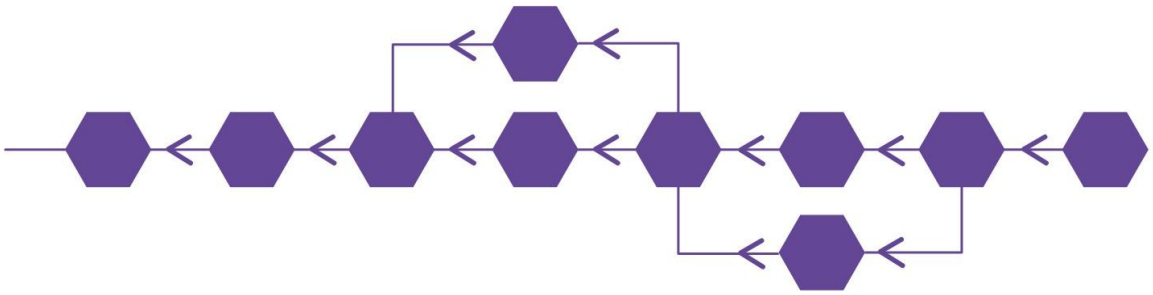
B 和 B ”也同时存在，但此处 B 在 B ”之前被视为信标，因此 B 被确认，而 B ”被拒绝。

应注意，具有明确顺序的两项交易间不可能存在冲突，因为子交易将自动视为无效交易。

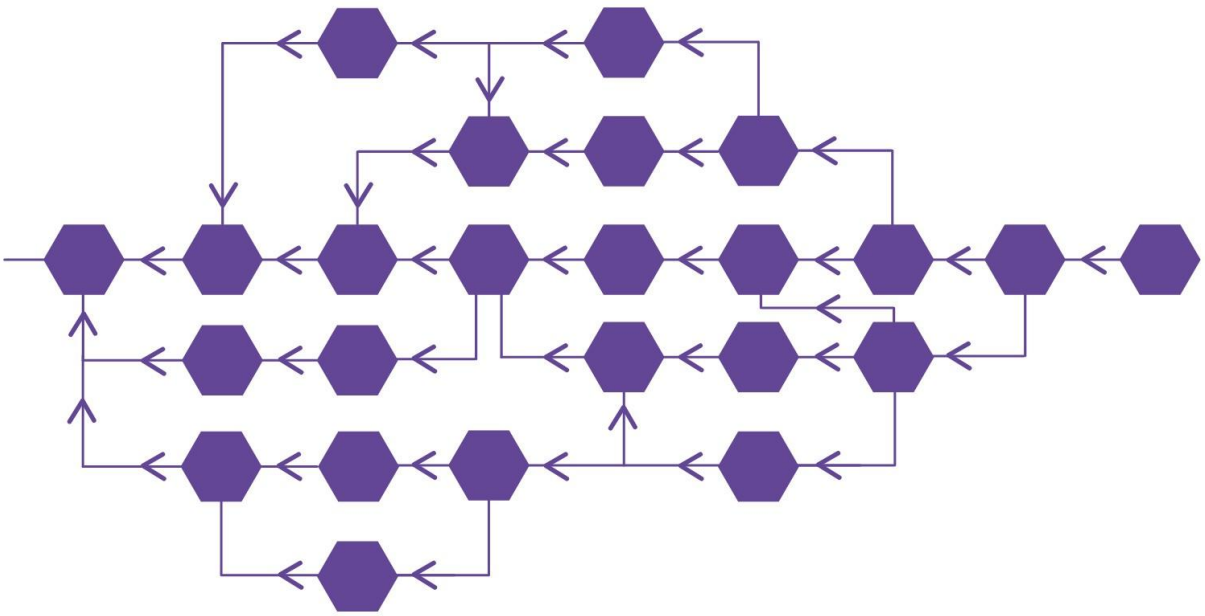
每个信标之间可以进行的交易数量不受限制。因此，与区块链不同，每秒的交易数量不受区块大小及其传输速度的限制；
而是取决于形成网络的节点处理这些数据的速度如何，以及这些数据的算力和 Virgo 代码的优化。

1.3 / 网络过载

节点让用户明确选择哪些交易可作为父母交易。为了扩展 DAG，在请求时，节点作为父母级提供无子交易。因此，如果每秒的交易数量很少，很少有用户会为其交易选择相同的父母级，因此 DAG 看起来就像是一条交易链。



相反，网络负载较高时，多个用户尝试同时创建交易，便会选择同一父母级。
接着，图表拓宽，许多交易处于同一高度。



该原则逻辑上也适用于信标交易：
如果网络过载，被驱逐的信标交易的数量增加，因为矿工需要更长时间才能看到新的信标交易。

每一个被驱逐的信标都少了一次扩展主链的机会，因此也少了一次保护网络的机会，更重要的是，其浪费了算力。

除此之外，矿工算力越小，发现信标时越有可能被驱逐，这增加了挖矿的集中度：
如果 A 和 B 同时找到一个区块，A 拥有 30%的算力，B 拥有 10%的算力，A 信标被逐出的机率有 70%，而 B 信标被逐出的机率有 90%。
这意味着，如果网络过载到有很大几率产生一个陈腐信标时，实际上，A 将比 B 更高效。

为避免这种情况，我们可以简单地实施 Yonatan Sompolinsky 和 Aviv Zohar 在 2013 年 12 月提出的幽灵协议，该协议已经在以太坊中实施。

幽灵协议建议在计算最长链时考虑陈腐信标，但正如这次在以太坊白皮书中所建议的那样，把挖掘回报归因于这些信标。
因此，在陈腐信标中投入的计算工作不会被浪费，矿工在所有情况下都会得到奖励，减少不平等。

任何情况下都不能接受陈腐信标：它们必须有难度限制，以避免矿工在低难度信标上挖矿获取回报。

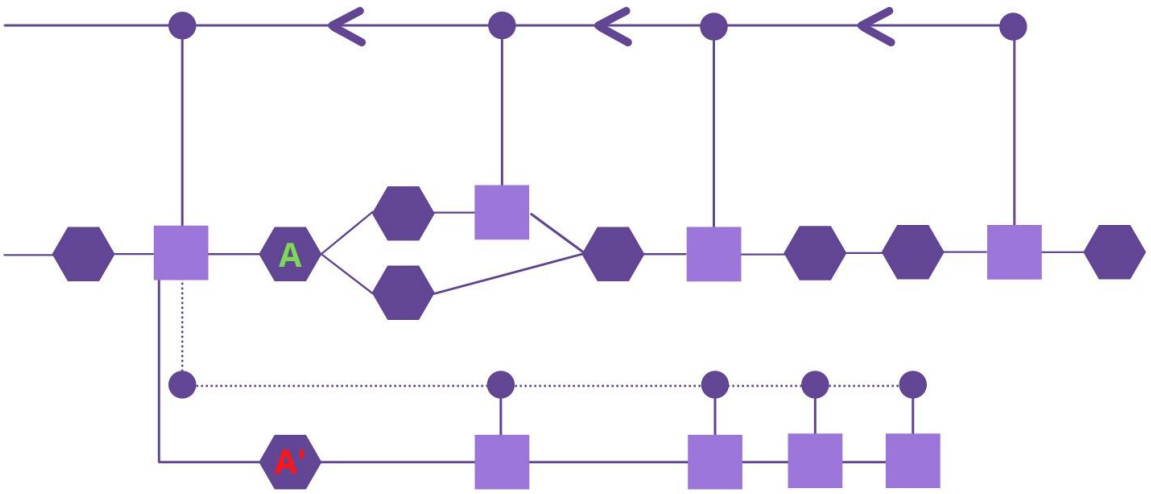
随着以太坊的普及，我们将使用术语“叔信标”来表示已被接受的陈腐信标。

因此，在 Virgo 实施幽灵协议的过程如下：

1. 标记必须指定父母信标，以及可选的一个或多个陈腐信标
2. 信标 B 包含的叔信标必须具有以下属性：
3. 它必须是 B 的 k 代祖先的子后代，其中 2<=k<=7。
4. 它不能是 B 的祖先。
5. 它不能包含在 B 的父母级中。
6. 对于包括在内的每一叔信标，B 的矿工将获得 12.5%的额外奖励，而相关叔信标的矿工将获得剩余的 87.5%。

1.4 / 撤销交易

与任何分布式账本一样，Virgo 上的交易可以在非常特定的情况下取消，这是因为网络在发生分叉时能够重新同步。
为了使攻击者能够减少交易，必须创建一个比主信标更大的信标链，此新链中需包含一项交易，同先前接受过的交易竞争。



例如在这里，攻击者首先向商户发送交易（**A**），以换取服务。服务一旦提供，攻击者就会创建一项交易 **A'**，使用与 **A** 相同的资金，但指向的是他控制的地址，而不是商户的地址。
一开始，**A'** 会被拒绝，但攻击者随后将创建一个优于 **A** 的信标链。非法链接成为主要链接，**A'**将被接受，**A** 将被拒绝。

需要注意，从事这种活动时，攻击者不得进行任意更改，例如创建不属于他的硬币或花费费用。这依然算是无效交易。因此，唯一能做的就是收回他已花费的费用。

攻击者试图生成其竞争链时，网络的其余部分继续拓宽主链。因此，要使其竞争链有机会成为最大的链，攻击者必须拥有至少一半以上的网络算力，称为 51%攻击。

在 Virgo，这种攻击更受限制：
对于传统区块链，如果主链被空链取代，矿工将不得不重新纳入已包含的交易，所需时间与取消区块的数量成比例。

对于 Virgo 来说，一旦攻击结束，诚实的矿工将从两条链中获得交易作为他们的父母链，这些交易将在信标中重新确认，几秒钟即可。

1.5 / 发行和交易费

Virgo 网络有一种基础货币——Virgocoin，主要用于价值转移。
主要面额为 VGO，相当于 108 个基本单位。
换言之，Virgocoin 将支持 8 位精度小数。

为支持开发，30,032,000 VGO 将按如下方式进行预开采和分配：

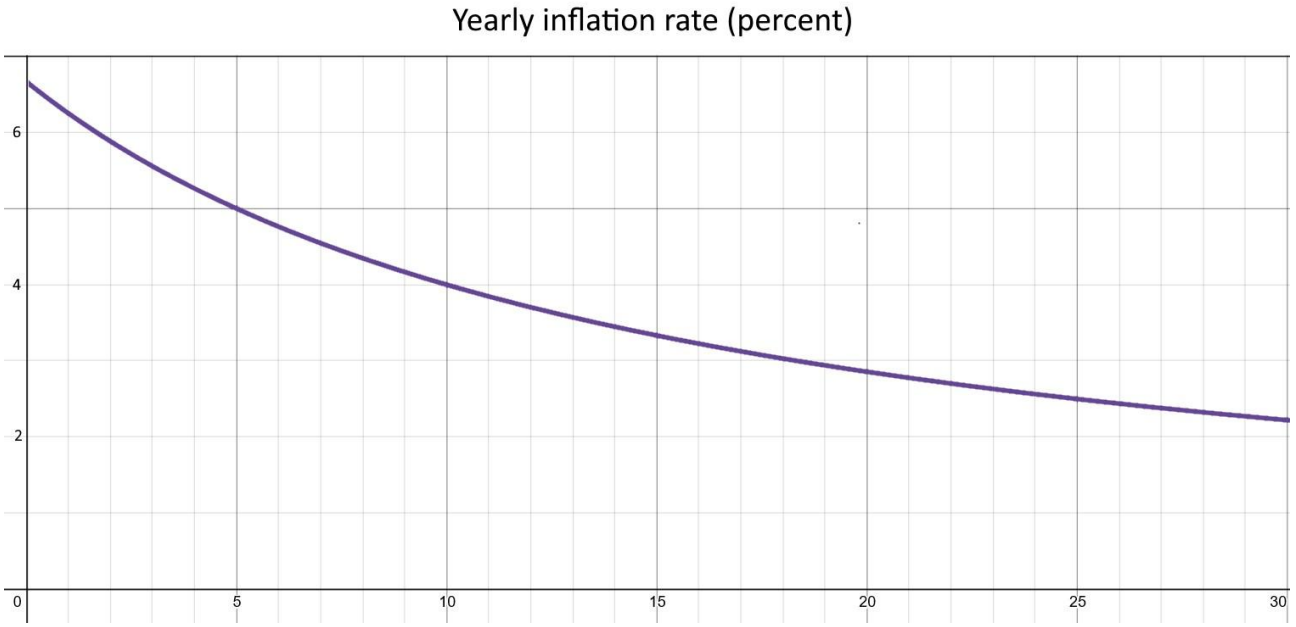
- 15%将供团队使用
- 15%将通过发展津贴进行分配
- 10%将通过营销活动进行分配
- 60%将留待 ICO 期间出售或在必要时销毁。团队无需赔付这 60%。

在计算资源上，创建一个信标并确保网络的安全十分昂贵。所以，鼓励网络成员保护信息安全，提供经济动机便十分有必要。

因此，按照惯例，每个信标只有一次花费，为交易的发行人分配一定数量的新硬币。
所以，即便有人恰好拥有很大一部分网络算力，他们也更倾向于脚踏实地获取回报，而非发动双花交易。

回报设置为每个信标 2 个 VGO，平均每 30 秒生成 1 个信标，每年创造出的 VGO 达 2,102,400。

与比特币不同的是，即使每个区块的回报不趋近于 0，但年通货膨胀率仍然如此：



选择不限制流通硬币的最大数量既是为了持续给予矿工一定激励，也是为了避免财富集中。
众所周知，由于各种原因，每年都会损失一定数量的硬币；随着年通货膨胀率的下降，它和每年的损失量或达到平衡，流通中的硬币数量将趋于稳定，甚至达到轻微的通货紧缩。

因不时产生回报且无障碍，理论上来说，Virgo 可以在没有交易费的情况下运作。
但是，取消任何费用都将为滥用行为打开大门，如 Penny-Spend 攻击，不必要地增加了 DAG 的规模和维护节点所需的存储空间。

为限制滥用，交易必须包含与其所占用的存储空间成比例的费用，预先明确每个字节在协议中的价格。

这些交易费用不会分配给矿工，而是会被销毁，以降低挖矿引起的通货膨胀。
这样一来，将摒弃费用估算算法。这些算法极其复杂，最终往往高估必要的交易费用。

挖矿盈利能力更具稳定性和可预测性，网络的安全性也能保障，同时保持每笔交易的低成本，因此适合小额支付。

1.6 / 挖矿的中心化

基于工作量证明，以相同方式挖掘任何加密货币：矿工使用精确算法计算区块的哈希值（在我们的例子中为信标交易），例如，Sha-256，反复添加随机信息，直到该哈希值与前一区块中定义的标准匹配为止。

该算法的问题在于，造出专用于此任务的机器（ASIC，专用集成电路），比普通大众可用的硬件（如CPU）快几个数量级。

然后，挖矿很快成为一项只有在工业层面和电力成本较低的地区才有利可图的活动。
这将使网络集中化，对攻击和区域中断更加敏感。
例如，截至本文撰写之时，中国占比特币散列率的 60%以上，而该国一个地区的断电切断了 40%的网络算力，导致费用激增。

为防止网络集中化，Virgo 将使用 RandomX 作为哈希算法。

RandomX 最初是为 Monero 开发的，是一种通用处理器优化算法。
它执行时使用随机代码（因此得名）以及几种 RAM 技术来最小化专用硬件（ASIC 和 FPGA）的效率优势。

因此，用 RandomX 进行挖掘时，唯一的高性能硬件是任何计算机中都包含的通用处理器。
这样一来，很多人都能接触到 Virgo 的矿产，且前期无需大量投资。
此外，选择一个相当罕见的算法将保护 Virgo 早期免受哈希率攻击。

1.7 / 表现

Virgo 由于自身的设计，每秒交易量没有具体限制。这意味着其表现只取决于代码的优化、执行 Virgo 机器的算力及网络速度，而基于区块链的加密货币在大多数情况下都受到区块传输大小和速度的限制。

当前的执行力为：在 AWS EC2’c3.2xlarge”服务器上，每秒处理 1000 多项交易，且无需任何优化。这很具发展前景，在 Virgo 的使用初期已足够。

新增一项交易时，大部分时间用于验证该交易的加密签名。目前使用的方法很大程度上可优化。
超级节点概念的实现是有可能的，由同一实体控制的多个服务器协同工作以提高整体处理能力（例如，管理 DAG 的主服务器和专门处理签名验证过程的“从属”服务器）。

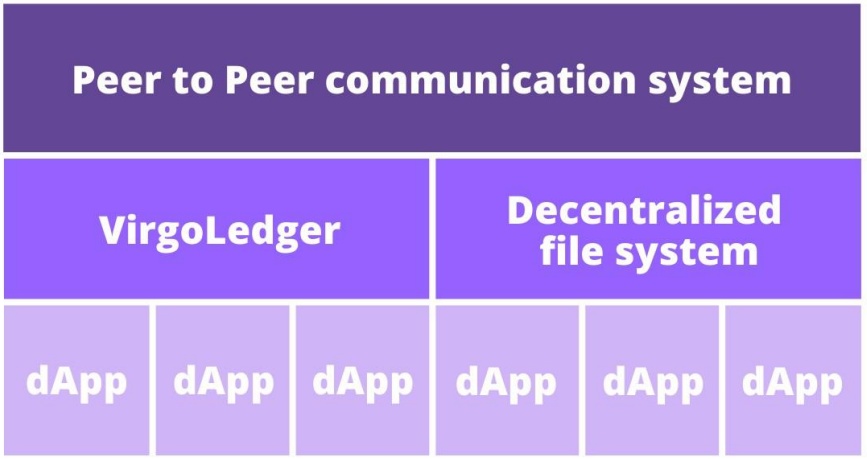
II - 去中心化应用存储

正如介绍中所说，在全球范围内，我们不相信这样一个强大、安全和去中心化的支付协议能让技术更广泛地为人所用。

分布式账本有潜力将自身与其他对等技术相结合，创建比集中式账本更健全、性能更高的应用程序和服务，同时避免任何形式的审查或授权。

Virgo 的首要目标是创建一套开源工具和协议，让任何人都可以尽可能轻松地创建分布式应用程序。

Virgo 账本是该套件的第一块砖，其余如点对点通信或内容分发库已经存在或即将问世。
这些砖块可相互依赖、共同成为应用程序的基础。单独的砖块也可视作 Virgo 的附加模块。



它们可以通过一个去中心化平台分发，类似于 AppStore 或 Firefox 扩展，尽可能易于访问和使用。
最终目标是通过单一平台向用户提供完整的分布式服务生态系统。

无论是云存储、VPN、分散金融、安全通信还是简单支付；
用户只需通过其钱包一键安装所需服务后使用即可。
全球都可享受到这种简易性带来的便利。

1 / 操作

从本质上说，Virgo 商店将同 Virgo 钱包模块挂钩。
它允许用户查阅与钱包互为补充的应用程序列表，然后一键下载、安装。

之后，应用程序可与钱包关联，例如，要求用户付款或向其分配资金。

1.1 / 去中心化

为使 Virgo 尽可能去中心化并且使用灵活，需通过点对点文件系统下载应用程序，如 BitTorrent 或 IPFS。

为了确保已下载文件的完整性，钱包将提前获取各应用程序的哈希值。需将下载数据的哈希值与参考哈希值进行比较，以防应用程序被恶意篡改。

应用程序、相关信息（例如其描述或网站）、版本的哈希值以及用户对它们的意见都将列在数据库中。任何人都可以创建和维护这些数据库，单击即可将其导入钱包。

这些数据库的维护者将对每个版本的数据库进行签名，然后通过与应用程序相同的文件系统进行分发。
因此，钱包可及时获取最新的应用程序和评论，不必依赖于集中式系统。

为简化安装，钱包将默认嵌入最热门的列表，其中一个将由 Virgo 团队维护。

最终便是一个完全去中心化的应用程序分发系统，既不损害安全性也不损害易用性。