Oyster 白皮书（版本：0.7b）

2017 年 9 月

Bruno Block
bruno@oyster.ws
oysterprotocol.com

随着访客为去中心化存储账本执行工作量证明，Oyster 协议促进网站悄然创造流量收入。

前言

尽管互联网呈指数级增长，Web 内容变现机制依然停滞不前。广告干扰隐私，偏离目的内容，同时打破网站的设计连续性。鉴于大家普遍忽视线上广告且对其有负面情绪，使用广告拦截器已成为主流。广告拦截器一旦侦测到广告内容，就会拦截并限制查看，在这种情况下，内容发布商日益受挫。内容发布商因广告拦截器而损耗大量资金，或因广告拦截器报复机制而错失海量查看量。因此，整个广告格局逐渐沦为无效、低效且侵入性的困境，无前瞻性和全方位的解决方案可部署。

同时，当前没有存储服务能做到既便捷又私密。如果您选择便捷，就选择标准云存储公司，其会限制隐私和匿名。闭源软件意味着您绝不可能真正享受公司允诺的服务。如果您选择隐私，那就难以碰到易访问且简单明了的 Web 界面，就是“上传”按钮也不易找到。

Oyster 协议才是真正一举两得的方案。协议引進截然不同的方案，促进内容发布者和内容消费者形成均衡之势和协作关系。因此，任何使用 Web 浏览器的用户可以以去中心化、匿名、安全和可靠的方式存储和检索文件。

下表列出组成 Oyster 生态系统的各当事方：

存储用户 - 用户可支付 Oyster 珍珠币来上传文件

职责
- 向两个代理节点支付正确数量的 Oyster 珍珠币。
- 尽管实现自动化，但最终由用户决定选择使用哪两个代理节点。
- 在本地浏览器中加密和拆分文件，然后将相应部分发送给选定的代理节点。
- 验证代理节点所安装数据映射的完整性。
- 通过分布式声誉系统，共享代理节点合约。
- 安全存储 Oyster 句柄，以便稍后从 Tangle 中检索文件。

奖励
- 安全、可靠且匿名存储文件。

网站所有者 - 运营网站的组织或个人

职责
- 向 Web 节点提供内容/商品/服务。
- 向网站 HTML 添加 Oyster 协议脚本。

奖励
- 获得 Web 节点发现的 Oyster 珍珠币作为酬劳。

Web 节点 - 正在访问网站的 Web 浏览器

职责
- 通过工作量证明搜索藏宝图，发现隐匿的 Oyster 珍珠币。
- 向代理节点提交发现的宝藏，代表网站所有者申领。
- 为代理节点执行工作量证明，获得 Web 节点身份及新藏宝图。
- 为 Web 节点执行工作量证明，获得 Web 节点身份及旧藏宝图。
- 向已执行足量工作量证明的 Web 节点发送 Web 节点身份和旧藏宝图。
- 通过分布式声誉系统，共享代理节点合约。

奖励
- 从相应网站所有者处获得内容/商品/服务使用权限。
- 将工作量证明的负担转移给其他适用的 Web 节点。

代理节点 - 有访问 Tangle 和区块链权限的网络设备

职责
- 通过互为邻居的节点，维持与 Tangle 的连接。
- 向 Web 节点和存储用户提供访问 Tangle 的权限。
- 如果适用，为新文件上传执行工作量证明。
- 将存储用户的珍珠币提交至区块链合约并处于埋藏状态。
- 解开因正确执行工作量证明而发现的宝藏。
- 保持以太币正余额，解开发现的宝藏。
- 在分布式声誉系统中建立声誉分数。
- Web 节点间的代理对等连接初始化。
- 向执行工作量证明的 Web 节点发送新藏宝图。

奖励
- 收集剩余的新埋藏宝藏，赚取 Oyster 珍珠币。
- 收取新埋藏宝藏中的费用，赚取 Oyster 珍珠币。
- 将工作量证明的负担转移给适用的 Web 节点。

IOTA Tangle - 称之为“有向无环图”的分布式账本

职责
- 保留已执行工作量证明的数据。
- 在不同地理位置分布数据的冗余副本。
- 负载均衡存储负担，例如：群体智能。

奖励
- 网络经验加强的攻击向量抵御力。
- 为交易带来更快速的平均确认时间。

以太坊区块链 - 具备智能合约功能的分布式账本

职责

- 提供实现 Oyster 珍珠币（代币）原生属性的智能合约框架。

奖励

- 区块链矿工收到代理节点用以太币支付的费用。

Tangle 机制

IOTA Tangle 意指有向无环图，即无区块分布式账本。IOTA Tangle 实时虚拟化见此处。每笔提交的交易必须为前两笔交易执行工作量证明，实现确认机制。这两笔交易则相应形成枝与干的关系。请参阅此处，详细了解如何向 Tangle 广播交易。每笔交易均有有效负载能力，用于保留由存储用户上传的数据。节点相互对接形成网状网络，交易由此网络传播，同时各节点保留交易的冗余副本。这种机制带来极大的数据副本冗余，进而大大降低数据丢失的风险，同时并不依赖集中托管供应商。

Tangle 节点旨在达到物理存储极限饱和状态时，自动删除旧数据（称之为“自动快照”，Tangle 暂无此功能）。这就是说，交易数据最终从节点删除。因此寻找隐匿 Oyster 珍珠币时，Web 节点执行工作量证明，将交易记录数据重新存储于 Tangle。该机制确保数据在整个 Tangle 节点拓扑中保留，绝不会被不可恢復地删除。

IOTA Tangle 在其路线图中包含许多创新，尤其是群体智能。群题智能对于 Oyster 协议至关重要，因为它去除维持账本完整所需的各 Tangle 节点瓶颈。类似于将 RAID 1 驱动器阵列设置转换为 RAID 10 设置。实施群体智能进一步增强 Oyster 协议的可扩展特性。

网络带宽是跨 Tangle 可靠地提交数据的最稀缺资源。Tangle 节点本身受限于网络接口带宽。鉴于 Oyster 协议旨在更可靠地向 Tangle 提交数据，访问 Tangle 的带宽将得到所获 Oyster 珍珠币的份额奖励。符合 Oyster 协议规范的 Tangle 节点称为代理节点。代理节点作为 Tangle 连接 Web 节点和存储用户的桥梁。

约 1 KB 左右大小的数据块存储于 Tangle，供交易的有效负载使用。SHA256 哈希作为在 Tangle 存储和检索数据的参考基础。SHA256 哈希被选中代表数据时，其转换成三进制形式，代表交易的接收地址。为了从 Tangle 中检索数据，哈希再次转换成三进制形式，以生成接收地址，该地址下的所有交易随后将得以恢复。最早颁发时间戳的交易包含代表选定哈希的有效负载数据。

Tangle 上的初始文件存储

存储用户想通过 Oyster 协议上传文件时，文件在本地浏览器中拆分为小块并加密。隔离机制仅允许用户访问相应加密密钥（称之为 Oyster 句柄），确保行恶者无法检索数据。

Oyster 句柄前 8 个字符代表文件名。其通常从上传至浏览器的文件名复制，但还可由存储用户定制，作为自身参考。原始哈希是 64 个字符长随机输入 SHA256 哈希，在存储用户的浏览器中生成，一致性尽可能高。

句柄最后 8 个字符是将原始哈希区分于整个加密密钥的加密盐。盐用于在原始哈希被发现后进一步保护数据，以免后续削弱哈希算法或彩虹表攻击创世哈希。因此，完整 80 个字符长的句柄是用于加密和解密数据拆分的完整加密密钥。Oyster 协议还支持向加密方案添加密码。原始哈希初始化代表数据拆分的 SHA256 哈希序列。数据先拆分为 1 KB 左右大小的数据块，然后各数据块分别用整个句柄作为密钥进行加密。各数据块随后由哈希迭代代表（创世、N1、N2 等），最终以 Tangle 交易的形式提交，每笔交易由两个代理节点提交。

Web 节点与存储用户之间的分布式声誉系统跟踪绩效最佳的代理节点，因此系统代表存储用户自动选择最适合的两个代理节点。Oyster 协议规定选择两个代理节点，进而数据映射中安装珍珠币数量最多的两个代理节点相互竞争。绩效更佳的代理节点在特定会话中收到较少珍珠币，但获得较多声誉，因此未来将收获更多珍珠币收入。

创世哈希值由存储用户提交给两个代理节点。一个代理节点旨在提交创世哈希（Alpha 节点）向下的数据映射，而另一（Beta 节点）旨在提交 NX 哈希（代表序列最后迭代）向上的数据映射。正确数量的 Oyster 珍珠币发送给 Alpha 指定的代理节点。Alpha 节点收到完整数量的珍珠币及 Beta 节点的以太坊地址。Alpha 节点收到上述数据以及表明身份的加密签署语句，则会将一半珍珠币发送给 Beta 节点。各节点的任何缺陷将通过分布式声誉系统报告，随即将大幅削弱整个 Oyster 网络中 Web 节点和存储用户的声誉。存储用户支付的珍珠币数量是最终埋藏于数据映射中珍珠币数量的一半。代理节点允许保存数据映射正确安装后剩余的珍珠币。

默认情况下，选定的代理节点负责执行工作量证明，将各数据块附于 Tangle。用于发送交易的 Tangle 地址是序列相应哈希迭代（创世、N1、N2 等）的三进制形式。然而，如果对等连接代理和新创世哈希有足够需求，代理节点能代表 Web 节点的工作量证明任务。

如需了解跨 Tangle 执行全部工作量证明的加增如何降低交易确认时间和增加网络的一般安全性，详细信息请见此处。

使用代理节点埋藏珍珠币

Oyster 珍珠币旨在埋藏于数据映射，而数据映射定义已上传文件的结构和内容。将珍珠币埋藏于数据映射并非由存储用户自己完成，而是由代理节点完成此项任务，原因如下：

- 代理节点有访问以太币余额的权限，因此支付所需的 GAS 费用，将所有珍珠币移入正确的指定目标。每个相当于 X GB 的用户数据需要在以太坊区块链上交易。

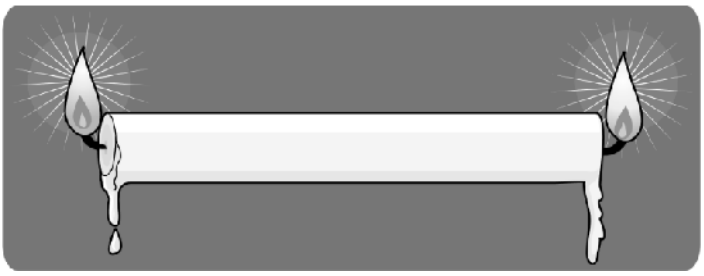
- 存储用户执行大量复杂区块链交易（包括调用定制智能合约函数）时会复杂到无法完成。这种复杂性传递给代理节点时，存储用户需通过典型以太坊钱包将珍珠币一次性发送给 Alpha 指定的代理节点。

- 由代理节点埋藏珍珠币于数据映射，将大大减少红鲱鱼攻击向量。

恶意存储用户佯装将珍珠币埋藏于数据映射，实则未埋藏，这种恶意行为称之为红鲱鱼攻击。如果行恶者上传的垃圾数据不含任何珍珠币，则 Web 节点白费时间寻找宝藏（无宝藏）。最终，Web 节点会意识到数据映射不在 Oyster 协议规范内，但那时 Web 节点花费的精力预计比恶意攻击者投入的精力要大。因此这会导致攻击成功并取得潜在收益。然而，由于 Web 节点依赖于代理节点接收创世哈希（定义整个数据映射），这将大大减少红鲱鱼攻击。这是因为如果代理节点开始分发代表无效数据映射（正确的位置没有正确数量的珍珠币）的创世哈希，那么 Web 节点易于报告代理节点，从而毁掉其声誉和未来流量。尽管代理节点有与声誉分数相关的一致身份，但存储用户和 Web 节点则要灵活得多。不仅难以为 Web 节点和存储用户建立一致的加密身份，而且 Oyster 协议定义了每 X 次寻宝数量重置身份所需的 Web 节点。存储用户没有可辨别的身份例外，无法以会话为基础与代理节点进行协商。

一旦存储用户提交珍珠币用于支付，大约一半珍珠币埋藏于数据映射，另一半由两个代理节点收取作为报酬。安装数据映射的两个代理节点就像两头点着的蜡烛。

烛蜡代表数据映射，而两头烛火各自代表一个代理节点。代理节点有权保留数据映射全部安装（或烛火完全熄灭）后剩余的珍珠币。默认经济压力表示代理节点十分缓慢或根本不安装数据映射（燃烧一头蜡烛）有益处。如果 Alpha 节点以每秒 10 个单位的速度燃烧蜡烛，且 Beta 节点以每秒 2 个单位的速度燃烧蜡烛，那么两头最终会在某处相遇，但 Beta 节点剩余更多珍珠币，这些珍珠币有权被保留。这种经济形势逻辑延伸为，两个节点均不燃烧蜡烛，都想将尽可能多的珍珠币占为己有。



分布式声誉系统将这种经济激励机制倒转过来。代理节点意指初始值为零（即最低数值）的已分配加密身份。Web 节点和存储用户寻求与拥有最高声誉分数的代理节点交易，尽管还会考虑延迟和其他选择限制因素。节点的平均蜡烛燃烧速度线性增加，其声誉分数也将大幅增加。这种机制会造成代理节点竞相燃烧更多蜡烛，尽管短期内珍珠币收入会减少。有意尽快燃烧蜡烛的代理节点短期内赚取的珍珠币将变少，但长远来看赚取的珍珠币将显著增加。因此，这种方式消除了代理节点经济激励机制的弊端。

代理节点将 Oyster 珍珠币埋藏于数据映射，便会调用 Oyster 合约的特别埋藏函数。数据映射扇区代表自选定哈希起的 1,000,000 个哈希（创世 - N999,999、N1,000,000 - N1,999,999 等）。因此扇区持有 X GB 用户数据。各扇区必须至少埋藏入一个珍珠币宝藏，有时扇区因两个代理节点之间校准不佳而包含两个宝藏。珍珠币在扇区内的具体位置由两个代理节点随机选择。因此，各扇区的珍珠币数量决定文件应在 Tangle 中保留多久。1 枚珍珠币将确保 X GB 用户数据在 Tangle 中保存一年。因此，Oyster 合约在计划存储时限内锁定珍珠币。期间，Web 节点执行工作量证明以找到隐匿的珍珠币。

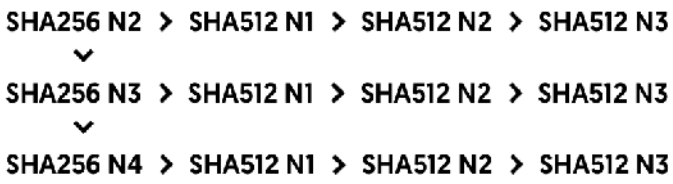
珍珠币必须处于埋藏状态，Web 节点才能申领。珍珠币还可在称之为 Epoch 的单独时区申领。Oyster 协议将 Epoch 定义为 1 年时长。这就是说如果数据映射扇区包含 2 枚珍珠币宝藏（4 年存储时间），那么就有 4 个可用 Epoch：第 1 年、第 2 年、第 3 年、第 4 年。Oyster 合约允许各 Epoch 申领刚好 0.5 枚珍珠币。Web 节点代表调用函数的网站所有者申领珍珠币。

Oyster 珍珠币寻宝

Web 节点搜索数据映射扇区，寻找隐匿的 Oyster 珍珠币。数据映射由称之为创世哈希的单个 SHA256 哈希定义。Web 节点从代理节点和其他 Web 节点那里获取创世哈希。创世哈希不是免费获得的，Web 节点必须执行合约币节点定义的特定工作量证明任务。从 Tangle 引用两个未确认交易来定义工作量证明任务：一个为指定的“枝”，另一个为指定的“干”。完成工作量证明后，Web 节点用刚提交的交易身份来响应合约币节点。合约币节点随后检查 Tangle，以验证引用的交易与此前指定的枝和干身份匹配，并验证交易指向已执行工作量证明的数据。合约币节点验证已执行工作量证明，就会发送创世哈希进行交换。

为了寻宝，Web 节点选择源自新赚取创世哈希的随机数据映射扇区。Web 节点随后参考 Tangle，查看工作量证明是否由另一 Web 节点在当前 Epoch 期限内执行。如果执行了工作量证明，那么 Web 节点将放弃该扇区，另行挖掘其他扇区。这是因为即使 Web 节点会找到该扇区的宝藏，也极有可能在 Oyster 合约的 Epoch 期限内，另一 Web 节点已申领了珍珠币。如果未在当前 Epoch 期限内执行工作量证明，那么 Web 节点会依次滚动该扇区的各连续哈希。例如：如果选中扇区 5，那么 Web 节点从哈希 N5,000,000 依次滚动到 N5,999,999，其代表 X GB 上传文件的数据。久而久之，可逐渐部署与该行为不符的全新 Web 节点策略，例如：欺诈性 Web 节点。如果在预计近期不会有可申领宝藏的扇区内执行工作量证明，则表明 Web 节点在欺诈。于是，其他非欺诈 Web 节点会限制对欺诈性 Web 节点的访问权限，最终该扇区仅对欺诈性 Web 节点开放。因此，欺诈性 Web 节点希望下一 Epoch 期限到来前，该扇区始终仅对自己开放。有关 Web 节点交互的博弈论机制可能要复杂和先进得多，有效寻宝策略在不断研究和精进下日趋复杂，于是形成囚徒困境。

对于遇到的每个哈希，Web 节点首先运用设备的 GPU（采用 WebGL2），在 Tangle 上为相应交易执行工作量证明。此后，Web 节点检索相应交易 1 KB 左右大小的有效负载，并计算当前 SHA256 哈希的 SHA512 哈希。Web 节点随后使用 SHA512 哈希作为解密密钥，尝试解锁有效负载。如果成功解锁，则就说明宝藏含有珍珠币。如果未成功解锁，则计算当前 SHA512 的 SHA512 哈希，然后滚动至序列（称之为“哈希链”）中的下一链接。如果 Web 节点达到 Oyster 协议规定的哈希链大小的上限，则会移动至数据映射中的下一 SHA256 哈希并周而复始地进行。



Oyster 协议规定每个扇区必须至少有一个宝藏。如果整个扇区内未找到宝藏，则会宣布该数据映射无效，并且分布式声誉系统的参与者会受到警告。这会导致最初为无效数据映射引入创世哈希的代理节点声誉降级。如果 Web 节点滚动完单个 SHA512 哈希链需几天时间，那么滚动完整个扇区需数月时间。这能防止 Web 节点在突发尖峰时刻消耗大量数据，以便将 X GB 数据消耗合理分配给数月（每天小于 5 MB）。因此，Web 节点不会承担带宽密集型任务，以免承受有限/昂贵的数据连接计划。执行 SHA256、SHA512 及解密函数均使用 CPU 指令。即 Web 节点通过 GPU 与其他 Web 节点和代理节点执行工作量证明协商，同时通过 CPU 在数据映射扇区内寻宝。

将私有种子密钥纳入藏有珍珠币的以太坊地址，从而将珍珠币埋藏入数据映射。因此，Web 节点发现宝藏时，其通过 HTML5 本地存储 (localstorage) 指令，存储和保护私有种子密钥。尽管可发现宝藏，但 Web 节点面临两大困境：

- Web 节点无法直接访问以太坊区块链，并且会难以调用复杂的合约函数。

- 以太坊地址包含珍珠币，但没有适用于 Gas 的以太币。因此，必须首先将以太币发送至别处地址，从而允许私有种子密钥生成以太坊矿工可接受的交易。

鉴于以上两种困境，Oyster 协议定义 Web 节点与代理节点协作，以解开宝藏。Web 节点安全发送私有种子密钥，和代理节点检查扇区内是否确实有珍珠币。确认有珍珠币后，代理节点发送极少量以太币至该地址，作为交易的 Gas。然后，代理节点提交交易至区块链。交易调用 Oyster 合约申领函数，将珍珠币申领至网站所有者的以太坊地址，其与发现宝藏的 Web 节点相符。但存在两项当务之急：

- Web 节点担忧代理节点不会用与网站所有者对应的以太坊地址申领珍珠币，造成珍珠币遭窃。

- 代理节点需要发送以太币至源自私有种子密钥的地址，促进含宝藏的 Web 节点申领之交易。其可能是伪装成 Web 节点的行恶者，期望代理节点将少量以太币发送至其控制的地址。一旦代理节点发送以太币，在最少量 Gas 支付完成后，行恶者会窃取边际量 Gas。利润率可能非常小，但可以无限循环，因此会造成代理节点产生重大财务损失。

为了解决 Web 节点的忧虑，分布式声誉系统会快速降低该代理节点的声誉。代理节点拥有一致的加密身份，以便建立声誉并说服 Web 节点和存储用户使用其服务。Web 节点将仅使用最具声誉的代理节点，解开已发现的宝藏。如果代理节点在单个 Epoch 期限内，在单个数据映射中的单个扇区窃取珍珠币，那么弊大于利。代理节点有可能会损失价值数千美元的未来潜在收入，而得到的只不过是不到一美元的收益。因此 Web 节点有信心与声誉佳的代理节点做交易。

为化解代理节点的忧虑，代理节点不接受第三方解开宝藏的请求，除非能验证整个扇区的工作量证明近期在 Tangle 完成。这就是说行恶者必须在整个扇区执行工作量证明，才能说服代理节点解开其宝藏。为解释攻击向量徒劳无功：行恶者必须耗费价值 5 美元的电力去完成工作量证明难题，才能获取 1 美分的收益。因此，如果代理节点证明可申领宝藏的扇区已完成工作量证明，那么代理节点向宝藏地址发送以太币以促进 Oyster 合约申领函数在经济上是可行的。

Web 节点与代理节点协作

Oyster 生态系统内发生一次主要交互是，Web 节点执行大量工作量证明，从其他 Web 节点和代理节点处购买信息。因此，Web 节点需要稳定访问 Tangle，才能保持操作正确。尽管 Web 节点整天直接访问 Tangle 在技术上是可行的，但当前库实现和硬件/带宽局限会限制 Web 节点成为 Tangle 网络的轻客户端。也就是说 Web 节点需要轻客户端中间主机来发送 Tangle 请求和提交。Tangle 轻客户端主机独立于代理节点存在，但其中绝大部分因无法通过 SSL 发送请求而令 Web 节点无法使用。Oyster 协议要求代理节点通过 SSL 发送所有 Tangle 请求。这是由于期望大多数

运行 Oyster 协议的网站将通过 SSL 托管，因此 Web 节点运行逻辑必须通过 SSL 加载，并且传出或传入通信也必须通过 SSL。

代理节点还能实现 Web 节点在对等连接中直接与其他 Web 节点交互。需按照 WebRTC 标准，使用 PeerJS 库。因此，运行 PeerJS 服务器软件的代理节点实现 Web 节点相互直接通信。

Web 节点不断需要创世哈希，特别是新创世哈希，因为一般而言首个 Epoch 期限内埋藏的 Oyster 珍珠币尚未被申领。如果代理节点包含与存储用户的文件上传会话，那么代理节点会保留创世哈希。Oyster 网络处于典型均衡状态时，代理节点始终保留过量新创世哈希，但 Web 节点始终处于对创世哈希的过量需要状态。

代理节点不会免费供应新创世哈希，而是确保必须执行大量用于交换的工作量证明。这种机制主要是为了防止行恶者轻松获取创世哈希及检索隐匿的宝藏。因此，工作量证明负担加重会令行恶者从代理节点寻找创世哈希在经济上根本不可能。工作量证明要求还消除对利他行为的依赖，这是 Oyster 协议极力避免的。

存储用户给代理节点带来多少负担，代理节点就会给 Web 节点带来多少工作量证明任务负担。因此，如果 Oyster 网络区域理想平衡，那么代理节点绝不会不断执行工作量证明来满足 Web 节点的需求。Web 节点与代理节点之间的交换序列与 Web 节点之间的交换序列是一样的。交换序列如下：

- Web 节点询问代理节点是否有新创世哈希或可靠的邻居身份（每个序列仅请求一种信息类型）。
- 代理节点会予以回应，同时表示有新创世哈希或可靠的临近身份。代理节点还会表示请求的工作量证明之负担程度。负担程度根据供需约束下的 Oyster 网络经济状态而波动。
- 如果 Web 节点认可工作量证明负担程度，便会接受该作业。
- 代理节点发送 Tangle 上三笔交易的三个引用。一笔交易包含此前对代理节点造成负担的相关存储用户数据。另两笔为 IOTA 算法推荐的以用于确认的未确认交易。相互指定地形成枝干交易。
- Web 节点在 Tangle 上执行 replayBundle 函数，因此可按代理节点的具体要求手动设置交易的枝与干。
- Web 节点完成工作量证明并提交完整 Tangle 交易后，其会向代理节点发送新提交交易的身份。
- 代理节点验证代表实时 Tangle 上的正确数据的引用交易身份，并按此前规定正确分配枝与干。
- 如需完成更多工作量证明以满足商定的负担程度，那么会按上述规定完成。
- 双方确定负担程度满足后，代理节点会向 Web 节点交付创世哈希或邻居身份用于交换。代理节点迫于分布式声誉系统压力交付商定的信息。

创世哈希逐渐从代理节点迁移至 Web 节点的集体意识。如果所有 Epoch 期限内的所有扇区均已申领，Web 节点会有意忘记创世哈希。这表明数据即将过期，除非存储用户向宝藏添加更多珍珠

币以延长数据受保障的生命周期，否则无法再得到工作量证明展示的保障。Web 节点运用 HTML5 本地存储 (localstorage) 指令来保留数据，其中包括已知创世哈希。如果本地存储指令承担的空间趋于饱和，那么 Web 节点会对网络所有者删除最不可能带来盈利的创世哈希，实现数据修剪。

将创世哈希从代理节点迁移至 Web 节点至关重要。与存储用户的上传会话完成时，创世哈希仅立刻存在于该会话的两个代理节点。Web 节点寻找创世哈希的积极意义在于其确定创世哈希存在于 Oyster 网络，因此代理节点专属持有创世哈希时，便能消除现存的初始风险。如果创世哈希即将被 Oyster 网络的集体意识忘却，则不会再用工作量证明来维系。因此在延长期限内，Tangle 不再负责在其节点拓扑中保留数据。

Web 节点至 Web 节点的交互

由于 Oyster 网络经济中存在供需制约动机，因此 Web 节点相互之间进行对等交互。对等连接通过 PeerJS 库实现，其依据 WebRTC 标准。为了实现 Web 节点之间的相互通信，其需要在整个 Oyster 网络相互识别身份。因此，各 Web 节点采纳加密伪持久身份。除非 Web 节点在寻宝过程中需要擦除存储内容并从零开始（如同刚加入 Oyster 网络），否则身份是可靠一致的。这是为了形成网络 Web 节点拓扑中的动态周转周期。如果 Web 节点趋于无限期坚持与相同邻居通信，那么网络会过于静态且无法回应环境变化。只要多数 Web 节点遵循 Oyster 协议有关身份刷新的规定，将迫使少数 Web 节点无意与邻居维持长久关系。

Web 节点首次引入 Oyster 网络或近期重置身份时，由于没有邻居，因此必须建立邻居列表。因此，Web 节点面临无法摆脱（Catch 22）的困境。邻居身份由其他 Web 节点共享，但由于最初无人知晓，因此无法主动请求。但即使通过 Web 节点参与的分布式声誉系统共享代理节点身份，初始解决方案看似也要向代理节点提出请求。因此，Web 节点假定初始可信的代理节点作为默认参考。网站所有者能更改信任哪些代理节点的默认值，因此需始终确保网络遵守去中心化原则。

代理节点用于代理相互已知身份的 Web 节点之间的初始连接。因此，代理节点保留的列表包含近期活跃的 Web 节点及其身份。新 Web 节点会从代理节点购买这些身份，换取执行交换序列所规定的工作量证明之权限。新 Web 节点不断从代理节点购买身份，同时也从刚结成新关系的其他 Web 节点购买身份。因此，根据收益递减定律，新 Web 节点的邻居列表大幅扩大，直至达到利益追求的饱和点。鉴于（通过工作量证明）寻找邻居所花费的电能可用于购买创世哈希和寻求宝藏，从而限制 Web 节点购买太多邻居。由于邻居列表已扩大，Web 节点现可从分布式声誉系统收到足够的声誉声明。因此，Web 节点能与新代理节点通信，同时还能确保原有默认代理节点受到控制。因为 Web 节点身份定期重制，因此迫使 Web 节点不断寻找新邻居。

Web 节点能执行与其他 Web 节点和代理节点的交换序列。这种机制允许 Web 节点执行工作量证明，以换取创世哈希、邻居身份等有价值的信息。交换序列描述如下：

- Web 节点询问邻居是否有新创世哈希或邻居身份（每个序列仅请求一种信息类型）。
- 邻居会予以回应，同时表示有新创世哈希或可靠的临近身份。邻居还会表示请求的工作量证明之负担程度。负担程度根据供需约束下的 Oyster 网络经济状态而波动。
- 如果 Web 节点认可工作量证明负担程度，便会接受该作业。
- 邻居向 Tangle 上的三笔交易发送三个引用。一笔交易包含此前对邻居造成负担的相关存储用户数据。另两笔为 IOTA 算法推荐的用于确认的未确认交易。相互指定地形成枝干交易。
- Web 节点在 Tangle 上执行 replayBundle 函数，因此可按邻居的具体要求手动设置交易的枝与干。
- Web 节点完成工作量证明并提交完整 Tangle 交易后，其会向邻居发送新提交交易的身份。

- 邻居验证代表实时 Tangle 上的正确数据的引用交易身份，并按此前规定正确分配枝与干。
- 如需完成更多工作量证明以满足商定的负担程度，那么会按上述规定完成。
- 双方确定负担程度满足后，邻居会向 Web 节点交付创世哈希或邻居身份用于交换。

一般而言，来自代理节点的创世哈希负担程度要大于来自其他 Web 节点的负担程度。这是因为代理节点持有相对较新的创世哈希，而 Web 节点持有相对较旧的创世哈希。由于对未申领的宝藏期望较高，所有新创世哈希的现行价格预计会更高。这还意味着 Web 节点先会向代理节点请求创世哈希，然后才会向其他 Web 节点提出请求。这种机制确保来自代理节点的创世哈希始终向 Web 节点的集体意识快速有效迁移，因为 Web 节点不太会受数据丢失影响。

Web 节点之间相互通信时，其会测量一贯通信类型的连接延迟。这就是说，对于有效负载交换大小一致的所有通信/交易，会测量从连接初始化到完成所需时间。因此，Web 节点能推算出邻居之间大概的相对距离。该连接延迟信息得以保留，相比远邻，Web 节点逐渐趋于选择近邻。该行为令 Web 节点的邻居列表逐渐优化，从而主要与邻近的 Web 节点通信。尽管相比延迟，网络更信赖代理节点的声誉，但该优化法同样适用于 Web 节点的代理节点列表。

延迟优化令 Oyster 网络渐成去中心化的低延迟网状网络，其带有高效跳跃路径，便于开发第三方应用程序。例如：有技能的程序员组成的任何小组可编写去中心化 JavaScript 电话服务，其扩展 Web 节点协议核心逻辑并使用自己的以太坊代币。因此，该扩展可作为开源代码发布并在 Oyster 社区内共享。于是，网站所有者可添加扩展程序，追求电话服务子经济可承担的额外收入。这种机制促进网站因吸引创意内容发布者而盈利，进而实现 Oyster 协议目标。电话呼叫机制简易的运行于 Web 节点提供的 API 呼叫，从而将音频数据包经由 Oyster 网络拓扑高效发送至目标接收者。因此，Oyster 协议作为扩展平台，成为去中心化代码开发和部署的基石，提供经优化的网状网络节点拓扑，并通过简单 API 实现自动化节点跳跃逻辑交互。

内容消费权利

互联网的基本社交合约是交换信息。网站所有者投资以制作/获取/交付原创内容和/或服务，同时还承担托管费用。天下没有免费的午餐，必须有经济桥梁来证明网站所有者的投资是合法的。

尽管需要该经济桥梁，但互联网整体上已诉诸于广告交换这一普通解决方案。广告总令人分心、离题、侵犯隐私和打断各处网站的设计连续性。因此，广告在更大的互联网社区遭致集体鄙视。作为正当回应，广告拦截器已成为有害于互联网经济的主流力量。因此，令创意内容发布者处于困境，因为其依然需要证明制作和托管内容的费用是合法的。在相关政策的摆布下，创意内容发布者只好决定和幻想中心化的广告交换。

Oyster 协议脱胎于老旧的广告范例，允许创意内容发布者完全自主掌控内容盈利。访客可支付入场费，但无需为令人不快、离题和分心的广告买单。随着资金回流至已深受其害的创意内容发布者，内容数量和质量会停止下降并再次升高。这会促使访客继续访问并通过 Oyster 协议消耗计算资源。

网站所有者启用 Oyster 协议十分简单。其仅需向网站 HTML 添加一行代码，全面启用 Oyster 协议并自动收到珍珠币付款，例如：

```
<script id="o.ws" data-payout= “ETH_ADDRESS”
src= “https://oyster.ws/webnode.js"></script>
```

如果访客不同意用自己的计算资源去交换，禁用 Oyster 协议也很简单。在这种情况下，拦截标记会安装于已禁用 Web 节点的 HTML5 本地存储 (localstorage) 区域。那么，访客的设备不会再执行任何工作量证明或寻宝任务，但会在网站所有者的网站上启用 JavaScript 标记以标定弃用。因此，网站所有者可选择轻松拦截，避免向不同意为其寻宝的任何人发送内容。

鉴于 Web 节点使用 HTML5 本地存储 (localstorage) 指令以保留数据，尽管被不同网站所有者调用，但依然保留相同的身份和工作队列。寻宝会话初始化后，Web 节点将实施调用的网站所有者的以太坊地址与申领者永久关联。

例如：有人正使用自己的笔记本电脑浏览四个喜爱的网站，其中两个网站启用了 Oyster 协议。访问启用 Oyster 协议的网站 A，访客的笔记本电脑将成为 Web 节点，将任何活跃的寻宝归于网站 A 的所有者。因此，发现的任何珍珠币会以网站 A 的以太坊地址在 Oyster 合约中申领。此后，有人访问启用了 Oyster 协议的网站 B。笔记本电脑将作为 Web 节点并保留相同的加密身份、创世哈希的收集、其他 Web 节点和代理节点的身份、及任何待定的目标数据映射。如果在网站 B 的司法管辖区内发起任何新寻宝，那么找到的任何珍珠币将归于网站 B 的所有者。

Oyster 珍珠代币功能

鉴于 Oyster 网络是完全去中心化系统，其要求有去信任机制管理其参考值代币：Oyster 珍珠币。Oyster 珍珠币是以太坊区块链上符合 ERC20 标准的代币，其包含启用 Oyster 协议功能的特定属性。

埋藏函数为珍珠代币的特定函数。埋藏以太坊地址可防止珍珠币遭提取，但依然允许存入珍珠币。依然可向埋藏地址存入珍珠币，允许存储用户延伸数据的生命周期，避免蓄意的数据过期。最初将文件上传至 Tangle 时，代理节点调用 Oyster 合约的埋藏函数。代理节点埋藏于数据映射的珍珠币由 Oyster 合约保留，因此不可花费。

Web 节点寻宝时，其会遇到已调用埋藏函数的以太坊地址的私有种子密钥，避免珍珠币被一次性全部提取。因此，只要有人检索私有种子密钥，无论是否有 Web 节点，将无法通过可调用所有符合 ERC20 标准的代币之普通转移函数来提取任何珍珠币。Web 节点必须请求代理节点代表网站所有者的以太坊地址调用申领函数。仅处于埋藏状态的以太坊地址可调用申领函数。Oyster 合约计算特定扇区的 Epoch 期限，并向申领者分配与 Epoch 期限相当的珍珠币。如果扇区内相当于两个或以上 Epoch 期限的珍珠币未被申领，那么应奖励申领者所有这些珍珠币。Oyster 合约无需将合约指标纳入其中，因为每个隐匿的以太坊地址已具体代表一个扇区。

申领者（网站所有者）调用已发现宝藏的 Web 节点后，代理节点用其以太坊地址调用申领函数。申领函数还定义了费用地址变量。代理节点调用申领函数时，其提交自己的以太坊地址作为费用变量。因此，Oyster 合约自动分配代理节点为解开宝藏所应赚取的费用。因此，代理节点收到的费用经一致同意且可审计。申领函数执行后，任何可申领的珍珠币将直接发送至网站所有者的以太坊地址，而双方确定的百分比将为代理节点保留，作为代理费。

因此，Oyster 珍珠币是交换的关键媒介，将网站所有者、Web 节点、代理节点和存储用户的经济动因联系在一起。

文件验证与检索

如需通过 Oyster 协议上传文件，存储用户的客户端选择两个代理节点向 Tangle 提交数据。数据从文件首尾两端开始处理，每端由一个代理节点执行，类似于燃烧两端的蜡烛。到特定阶段，两个代理节点将在数据映射中点附近相遇。然而，在不太可能的情况下，一个代理节点出现缺陷而不执行工作量证明（并自行保留 Oyster 珍珠币），而另一无缺陷的代理节点完成整个数据映射。存储用户的客户端会注意到缺陷，并通过分布式声誉系统加密报告有缺陷的节点。

文件完全提交至 Tangle 后，客户开始下载整个数据映射以验证其完整性。客户端使用代理节点而非（执行上传）的头两个节点来访问 Tangle 和下载数据映射。该验证阶段在技术上可略过，但建议采纳，因为在极不可能情况下，两个代理节点会与存储用户共谋或双双有缺陷，那么存储用户客户端能通过分布式声誉系统加密报告攻击。诚信的代理节点随后选择执行不诚信的代理节点未尽任务，尽管不从 Oyster 用户处寻求任何珍珠币支付。诚信的代理节点寻求执行该数据纠正操作，因为能极大促进加密声誉，从而收入增加前景愈发光明。

验证流程不仅防范不诚信的代理节点，而且还保证数据映射不会因程序错误或执行漏洞而有缺陷。用于上传验证和公正数据检索的下载序列定义如下：

- 客户端从 Oyster 句柄检索原始哈希，并作为 SHA256 函数的输入进行提交，从而生成创世哈希。
- 客户端计算 SHA256 哈希链中选定哈希的三进制数据（创世哈希针对一次迭代）。
- 客户检索与此前步骤中三进制数据相关联的 Tangle 交易有效负载数据。依据（Web 节点和存储用户采用的）分布式声誉系统，通过选定的代理节点执行检索。如果正在执行验证流程，用于访问 Tangle 的代理节点不能与执行初始上传的代理节点相同。
- 一旦检索有效负载数据，客户端试图用整个 Oyster 句柄作为加密密钥来解锁。Oyster 协议还允许密码用于加密方案，而仍要警惕存储用户忘记密码的风险。如果 Oyster 句柄或可选密码丢失，那么文件会永久丢失。
- 如果有效负载数据解锁，那么其属于组成已上传文件的数据序列的一部分。如果尚未解锁，其作为包含隐匿宝藏的 SHA512 哈希链之参考。随着客户端逐步搜索数据映射，其检查每个扇区（1,000,000 个 SHA256 哈希）至少一个 SHA512 哈希链作参照，否则宣布数据映射无效并使用分布式声誉系统执行恰当的程序。
- 检索单个哈希的有效负载数据后，其存储于存储用户的永久储存设备且不受相应的内存使用分配限制。
- 客户端将当前 SHA256 哈希提交至 SHA256 函数，计算 SHA256 哈希链中的下一迭代。计算出的哈希是哈希链中的下一迭代。（例如：创世哈希之后是 N1 哈希）。
- 该流程周而复始，直至从 Tangle 获取整个文件。
- 文件各个部分粘接起来后，将整个内容与嵌入其中的校验和作比较，从而确保数据完整性。

存储用户可通过任何客户端访问相关文件，即使并非代理节点的 Tangle 节点也可访问。仅有两项事宜需要检索该数据：Oyster 句柄及 IOTA Tangle 的通用访问权限。

分布式声誉系统

存储用户要求代理节点埋藏珍珠币，而 Web 节点要求代理节点申领珍珠币。在两个实例中，宝贵的珍珠币被发送至代理节点进行处理。因此，存储用户和 Web 节点采用监控系统，确保代理节点受到控制。该系统称之为分布式声誉系统，运作方式与易趣 (eBay) 类似。代理节点类似于 eBay 卖家，而 Web 节点/存储用户类似于 eBay 买家。Web 节点/存储用户仅与能找到的声誉最高的代理节点进行交易。代理节点遴选算法还考虑网络延迟、流量限制、协议禁止等其他因素（例如：存储用户将同一代理节点同时用于上传和验证流程）。尽管有上述单独标准，仅有相对较少的最高声誉代理节点收到绝大多数流量，方能获得绝大部分珍珠币收入。因此，代理节点一定要诚信，才会有收益。

所有声誉分数最初均为零。不存在负的声誉分数，否则恶意代理节点会生成初始分数为零的全新加密身份，从而消除负面声誉。变换声誉以实施诈骗这一有缺陷的策略相当于 eBay 欺诈卖家不断生成新账户。大家都会忽视毫无声誉的新帐户。而声誉较高的诚信卖家将得到绝大多数生意。

诚信的代理节点授予 Web 节点和存储用户访问 Tangle 的权限，从零分开始逐渐积累声誉。真实访问 Tangle 是可验证且可容错的，引用更多有声誉的代理节点以确认真实 Tangle 正在被访问。有诚信的代理节点收集足量初始声誉，便开始接收 Web 节点/存储用户执行有价值基础的交易（埋藏和申领珍珠币）的请求。鉴于 Beta 节点处理更少珍珠币价值，代理节点最初作为 Beta 节点被分配。如果正确执行有价值基础的交易，那么代理节点的声誉将开始大幅提升。进而有更多 Web 节点/存储用户请求进行有价值基础的交易。不诚信的代理节点很少能大幅提升声誉。

各代理节点的身份是生成 PGP 密钥的基础，因此必须保密。如果 PGP 密钥遭泄露，系统会快速降低节点声誉（和未来收入前景），因为行恶者快速利用其来获取短期收益。代理节点保护 PGP 密钥的能力优于且保证保护宝藏数据映射坐标和宝藏私有种子密钥的能力。

代理节点和 Web 节点/存储用户同意执行有价值基础的交易前，需协商两项条款：Tangle 的最低限度及区块链的最低限度。账本的最低限度确定合约成功执行后及合约期限到达前应有交易的最小范围。存储用户的客户端定义数据映射各 SHA256 哈希的所有枝干交易，作为 Tangle 应包含的最小范围。作为回应，代理节点设定区块链的最小范围，其包含将珍珠币转移至受控地址。
如果提出的合同符合 Oyster 协议和双方的环境背景，那么双方会用各自的 PGP 签名数字化签署合约。

代理节点始终成对安装数据映射，同时相互竞争以埋藏最多珍珠币。相反，存储用户的客户端向各节点分配指定枝/干。一旦数据映射安装完成，起观察作用的 Web 节点可估计各代理节点的性能，将合约定义的指定枝/干与 Tangle 实际引用的枝/干作比较。因此，分布式声誉系统的参与者就最佳绩效人员达成一致同意。执行大多数工作量证明的代理节点，其声誉分数会提升；而执行少量工作量证明的代理节点，其声誉不会变化或降低（视绩效程度而定）。因此，代理节点出于（有关收入形成）的长远利益而尽快执行工作量证明。经济压力确保存储用户快速高效上传文件。

代理节点能抵消 Web 节点因希望购买创世哈希或 Web 节点身份而带来的工作量证明负担。鉴于具有较好声誉的代理节点有更多交换序列与更多 Web 节点交互，一般安装数据映射速度更快，因

此可抵消更多指定枝/干工作量证明。如果合约到期后，已签署合约有数量为零或近似零的指定枝/干与 Tangle 匹配，那么该状况将被视为有缺陷并且代理节点的声誉将大幅降低。

Web 节点需要解开发现的宝藏时，其在合约中定义区块链的最低限度。最低限度规定来自宝藏地址的珍珠币应在 Oyster 合约的网站所有者的以太坊地址申领。该合约未定义 Tangle 的最低限度。如果代理节点同意条款，则会使用 PGP 密钥数字化签署协议。仅在 Web 节点收到合约的签署副本时，才会向代理节点发送宝藏的私有种子密钥。如果合约到期前未达到区块链最低限度，那么 Web 节点会引用签署的合约，通知分布式声誉系统其他参与方任务失败。因此，Web 节点构成的网状网络逐渐达成共识：代理节点并未执行应尽任务，因此降低其声誉和未来收入前景。

结论

Oyster 协议旨在解决创收、匿名且可访问存储及去中心化应用程序开发和部署。如同以太坊区块链为代币创建提供简单明了的框架，Oyster 协议为访问去中心化网状网络提供简单明了的框架。

Web 节点意指日常使用的计算机、智能手机、汽车、冰箱，可以是任何安装现代 Web 浏览器的设备。其相互直接通信，仅需代理节点偶尔提供连接代理。其会久而久之自动选择延迟较低的邻居，从总体上优化节点拓扑。可使用热门语言 JavaScript 便捷编写扩展程序，因此开发人员可获得访问全球网状网络的权限。这种机制为便捷编写去中心化应用程序和访问性能优越且延迟低的网状网络培育了沃土。

Oyster 协议激发数百万网站的隐匿收入潜能，化解个人和企业的存储难题，并打造开发人员亟需的网状网络平台。