

RASMART

WHITE PAPER

Ver 1.0

目录

1. Rasmart系统特征

2. 引言

2.1 问题和解决方案

2.2 区块链作用

定义

3.1 网络节点

3.2 最后保存的区块

3.3 节点同步

4. 网络共识

4.1 共识比较

4.2 主要节点与记录节点的概念

4.3 网络节点的设备

4.4 共识建立

4.5 建立并初始化分类账本

4.6 分类账中不包括的交易

5. 交易处理

5.1 交易

5.2 共识建立

5.3 交易处理

5.4 分类账本条目结构

5.5 RS分类账本结构

5.6 区块大小

5.7 搜索交易参与者

5.8 数据传输通道

5.9 系统中的行为

5.10 添加交易进行验证

5.11 交易成本

6. Rasmart分析

6.1 网络结构

6.2 网络中节点的识别和寻址

6.3 网络组织

6.4 DHT

6.5 PEX

6.6 流量

6.7 分类账结构

6.8 存储库和Merkle树

6.9 交易链和区块链

6.10 完整性同步

7. 数据库实体

- 7.1 零钱包
- 7.2 交易
- 7.3 任意实体
- 7.4 智能合约
- 7.5 交易
- 7.6 分叉问题解决
- 7.7 决策模块
- 7.8 共识实施
- 7.9 交易图的并行化
- 7.10 将交易图划分为部分
- 7.11 佣金
- 7.12 佣金计算方法

8. 智能合约

- 8.1 介绍
- 8.2 实体
- 8.3 智能合约方法
- 8.4 虚拟可执行机器
- 8.5 价值条款
- 8.6 执行智能合约条款
- 8.7 数据源
- 8.8 API

9. 安全和威胁能力

- 9.1 中间人攻击
- 9.2 51攻击
- 9.3 RSA密钥破解
- 9.4 女巫攻击
- 9.5 “双重浪费”问题
- 9.6 流量拦截和替代
- 9.7 本地存储改变

10. 实施计划

- 10.1 项目技术实施计划
- 10.2 ICO
- 10.3 RS加密货币

1. 系统特征

Rasmart是一个基于区块链技术的平台。 RS专门用于扩展基于分布式分类账的金融服务功能、自我执行的智能合约和加密货币。

目前，区块链是一种快速发展的技术；作为一个基于区块链平台Rasmart使用高级功能会满足区块链形成的一系列要求。

1. 交易速度。定性特征，显示整个系统中的每秒交易数量。
2. 灵活使用。 Rasmart提供灵活和容易适应的API，以便集成到第三方服务中。
3. 安全。由于0.2秒的区块处理时间和一致性协议建设，各种攻击的可能性会消失。
4. 自己的智能合约。Rasmart系统的智能合约很快，因为使用经过时间考验的程序语言（Lua），其优点是快速执行以及不需要大量计算资源的解释器，它们共同提供交易大容量和快速执行+具有各种功能的不断更新的sdk。

2. 引言

比特币作为第一个区块链平台，推动了参与者之间分散式相互关系的发展，其中价值交换不按中央机构发生。实际上，这正是因为如此，在相当短的时间内比特币受到热情的用户欢迎。在比特币网络中，只有一个操作 - 这就是没有中间人参与的用户之间的交换。

以太坊是一个平台，允许基于区块链技术创建在智能合约技术上的任何分散式在线服务。

不过基于区块链的第一个平台没有找到用户的认可，只有少数人使用这些平台。此外，以每秒几次交易的低速不允许增加大众观众。所有随后的区块链平台中，使用了首先应用于比特币的经典方案。

今天，只有金融行业拒绝参与者之间分散（直接）相互关系的积极采用。虽然从技术和组织的角度来看，创建分散式金融服务系统比经典银行更容易。为此拥有所有必要的知识和经验。

因此，要创建基于分布式分类账构建的最方便的分散式金融服务系统，将需要：

1. 处理数据高速度（每秒数十万次交易），而交易成本应尽可能低；
2. 为用户创建最方便的统一系统，可以统一实现分散式金融服务的所有参与者和元素：法定货币支付中心、用户个人信息、信用历史、加密货币终端等。

RS平台能执行这些任务。

Rasmart是一个分散式统一平台，能够统一所有金融服务参与者，根据分布式分类账的原则快速、安全地进行交易。联邦系统和自我执行的智能合约提供创建独特解决方案的机会，建立在不同金融产品之间的相互关系。Rasmart提供释放区块链项目的巨大潜力，该平台不仅可以用于金融行业，还可以用于由于技术（低速）和财务（交易高成本）约束而无法使用分布式分类账的其他行业。

2.1 问题

网络延迟是许多区块链项目的主要限制之一。一个比特币交易的平均时间为10分钟或更长。与此同时，银行可以立即进行交易，交易时间不超过1秒。

技术问题

我们发现以下问题：

容量指网络处理数据所需要的时间；

交易速度；

延迟是与系统期待响应时间相比系统响应实时。

存储数据量是网络中存储的数据量（以太网一直增长几十个GB）；

高成本是所有操作的关键参数，特别是两中：

1. 物联网操作允许将各种项目组合成一个全球网络；
2. 交易成本低的微支付，例如：小型购买（咖啡厅结账、商店购物）、各种小额贷款。

这些问题和局限都阻碍区块链技术在许多行业（包括金融行业）的广泛传播。

区块链技术可以实现任意数据库，这种数据库具有以下特征：

可用性。分布式数据库同时存储在网络的各个节点上。也就是说，即使没有与网络的其余部分连接，也可以在任何节点上访问所有信息。

完整性。区块链使伪造成为不可能，并且允许快速发现伪造事实和地址。

隐私性。由于非对称数据加密，只有信息接收者才能访问该信息。电子签名保证信息真正属于发送者。

可靠性。在各个节点上存储数据库允许在任何节点发生故障后完全恢复信息。即使仅有一个未损坏的节点，网络也可以完全恢复。

可扩展性。允许通过添加节点来快速、简单地增加或者缩小网络的大小。为了连接到网络，只需注册并连接到任何现有节点。

集成性。由于API，可以将任何软件连接到分布式数据库，这种软件能够接收并保存其中的任何信息。

由于以上特征，数据库的使用范围超出金融企业。这就是安全信息交换、电子文件流通、报告自动化和各种信息收集的通用方法。假如行业自动化需要很多参与者和信息存储的高可靠性，该系统将受到欢迎。

2.2 区块链作用

区块链是一串使用密码学方法相关联产生的数据块。每一个数据块中包含了前一个数据块的加密函数（或散列算法）。如果我们只有两个数据块，比如链中的第一个和最后一个，那么我们可以100%确定它们之间的所有块都是可靠的。

基于这种原则而建立的数据库允许通过记录新值来添加信息，所有先前的值保持不变，从而创建一致的更改历史记录。信息无法删除。因此，可以发现数据库中的任何操作，然后检查其真实性。对于开放式分类账，这是一个良好的解决方案。

不过，开放式数据库也能提供保存机密信息的功能。使用接收者的公钥只允许私钥持有者访问信息。只要密钥长度足够，RSA加密算法就可以保证信息加密。网络中其他参与者只能看到添加信息的事实，但无法访问这种信息。

该平台为用户提供分布式分类账的存储、保护和同步的技术机制。信息存储的结构和格式没有限制。RSA仅在数据分类账中添加有关连接性和真实性的信息

3. 定义

1. 系统是一组分散的网络节点，执行处理、保存、转入交易，执行和确认智能合约的条款。处理来自第三方系统的请求，在请求时提供信息数据。
2. 网络节点是安装完整网络客户端的一台计算机，连接到公共系统，用于进行交易、数据存储和传输。
3. 分类账是在所有网络节点上系统确认并存储的交易清单。
4. 交易是系统事项，表示执行智能合约方法的请求或网络上的任何操作，并将结果记录在区块链系统中。
5. 智能合约是计算机协议，促进、验证或确保符合交易条款。他们通常有一个用户界面，经常模拟合同关系的逻辑。智能合约的关键特征是其分散性和独立性。
6. 智能合约的方法是程序代码，该代码负责计算智能合约条款的工作结果并将其记录到分类账中。
7. 缔约方是最终的网络参与者和系统用户。

3.1 网络节点

为了建立一个基于节点的免费访问和连接的独立、分散的网络，我们同时使用几种节点，具体取决于其用途：

1. 普通节点参与交易验证的有效性，但具有最小信任因子，是在下一个选择网络节点角色的循环期间可以成为可信节点或当前处理节点。
2. 可信节点参与交易验证，具有比较高的信任因子。
3. 网络主节点积蓄从普通节点发送的交易并将这些交易分发到可信节点。
4. 记录网络节点是在一轮中选择的可信节点之一，形成数据块，将它记录在数据库中并发送给网络中的所有参与者，获取散列并形成新一轮。

3.2 最后保存的区块

区块的通用分类帐（CRB）是所有系统节点中区块的整个通用分类帐的同步状态。

通过分类帐区块概念，我们是指存储信息的单位，其包含先前区块的哈希码和与该分类帐相关的与先前区块相关联数据的列表。在从另一个节点接收到该区块时，它将根据该号码将其置于区块的共同分类帐中。因此，可以节省网络带宽。

在同步期间，仅对区块号进行检查。如果该节点缺少该区块，则会下载并保存该节点。

因此，系统随时包含最新的分类帐副本。我们把它命名为最后一个分类帐。达成共识后，由负责分类帐形成的节点自动创建。该区块被发送到所有系统节点，以便保持所有系统节点中分类帐状态的最新统一。每个节点与网络中的所有其他节点相关联，并与其交易不断地交换新的消息（区块与交易）。因此，由区块组成一套交易，然后将这种交易添加到分类帐中。同时，每个服务器为其他服务器生成一组假设的候选节点和建议的一组交易。在检查时会作出决定，确定是否将它们添加到分类帐中。

因此，可以在多个服务器（系统节点）上多次存放分类帐数据，并保护所有信息。系统中的节点越多，可靠系越高。

4. 网络共识

RS一致性协议是一种群体决策技术，目的是开发所有网络节点可接受的最终解决方案。

4.1 共识比较

为了不同类型的共识做比较，首先必须确定RS分散式分类帐的原则：
分类帐可用性。节点可以随时将数据写入分类账并从中读取；
可修改性。网络各个节点都可以进行更改；
一致性。系统各个节点彼此一致，所有节点看到分类帐完全相同的版本；
抵抗分离。如果一个节点变得不可操作，这不影响整个分类帐的操作。

4.2 主要节点与记录节点的概念

网络各个节点都是独立的、没有优先级。

为了确保数据存储安全、提高交易处理速度和进一步的信息传输，Rasmart平台使用专有的组合协议。

记录节点对交易数据块进行签名并将其保存到分类帐中，然后将形成的数据块发送给系统中的所有参与者。系统中各个参与者将收到的数据块写入其分类帐，并将写入的数据块的散列发回记录节点。记录节点形成主节点和可信节点的新列表，并将生成的列表发送给各个网络参与者，从而创建Rasmart平台的新一轮。

4.3 网络节点设备

我们的目标是创建一个具有交易处理最高速度的平台，因此我们提出通过强大的服务器和互联网高容量在最佳条件下使用物理刺激来维护网络节点。

作为物质补偿，每个节点所有者将从其处理的分类账的交易费用金额中获得RS代币的报酬。其余旨在可信节点（参与BFT决策的节点）之间分配。百分比可以改变；在ICO之后三年内，通过网络节点的联合投票将其分配到速率形成系统。

因此，我们鼓励服务器所有者将该服务器保持在最高性能，并保持高质量的高速通信通道。

4.4 共识建立

主节点的主要任务是：从候选状态获取交易，从所有节点添加到分类帐，处理它们，构建最后一个相关分类帐，并将新建分类帐发送到所有网络节点。交易处理和建立最后一个相关分类帐的过程正是寻找一个共识解决方案。创建最新分类账的结果就是一致性协议的解决方案。

这个过程可以分为以下阶段：

- 搜索主节点；
- 形成可信节点列表；
- 接收交易列表，并建立一个候选名单待列入分类帐；
- 处理候选名单，（在节点之间进行投票）；
- 从候选名单中移除未确认交易，未经验证的或未确认的；
- 建立要添加到分类帐的已确认交易列表；
- 将带有时间戳和包含该交易信息哈希码的交易添加到分类帐中；
- 将交易发送到所有网络节点；收到后，将其添加到所有节点的分类帐中。

4.5 建立并初始化分类账本

该过程可描述如下：

- 系统网络的终端用户生成一个交易；
- 当满足其中指定的智能合约的所有条件时，用户通过使用平台软件调用所需的方法来发起动作（交易）；
- 为了遵循区块链的基本原理，验证器的内核跟踪最新分类帐版本的同步和不变性；
- 在建立共识的时候，在周期中收到的所有交易都收集在区块中；
- 每个区块接收自己的编号，该编号由标签和节点标识符组成，该标识符已被转换为哈希码。接下来，将该块放置在寻找共识模块中；
- 在编辑白名单候选人之后，不仅将交易的哈希码写入分类帐，还要写入区块的哈希码，并总是基于它来验证源代码；
- 该哈希是一种独特的交易块签名；另外，它包含交易块创建过程和时间的详细信息；
- 在使用联合搜索算法达成共识后，添加到区块的交易将传递给验证器的内核以写入分类帐。

4.6 分类账中不包括的交易

交易清单不包含的交易是标记为未确认的。交易启动者立即收到关于交易未确认的信息。未包含在分类帐中的交易被拒绝。

5. 交易处理

5.1 交易

交易是系统的最小单位，包含与合同方法有关的或在无需创建智能合约情况之下资产之间直接转移的信息。将来，结果保存到等网络。

5.2 共识建立

该系统使用联合模型来建立共识 - 投票构建可信赖的验证器节点。此外，应用共同构建算法 - 一种有限状态自动机通过的算法。共识的特征就是周期性：每个周期交易从网络中提取并放入池中。然后所有交易都被发送到可信节点以获得响应。如果获得响应，为这种交易发送请求，以便将该交易添加到验证器的分类帐中。当交易合法性得到充分确认的链条末端建立共识时，交易发送到验证器，并带有将其写入并存入分类帐的标记。

5.3 交易处理

为了实现系统的去中心化特性，每个服务器必须具有两个分类帐存储库，并且也是所有交易完整的处理程序。

系统使用系统内核的概念。通过内核，我们是指执行特定生产任务的数据处理程序，而不管其他系统组件的可用性和可操作性。每个内核在输入时，在执行任务时，都会收到一系列用于处理的变量。每个内核在输出时，提供结果：否定、肯定、错误等等。除主数据集外，系统内核始终包含响应代码。这种结构是每个过程的最高可能速度所必需的，它们必须彼此独立工作。

5.4 分类帐本条目结构

为了实现重要的分类帐性能，但同时在不影响安全性的前提下，我们建议使用分类帐数据库，而不必从上一个块的哈希码和交易处理结果构建Merkle树。

Merkle树是一种用于检查数据完整性的哈希函数，用于获得链的唯一标识符并恢复序列。数据分为多个小部分 - 使用Leaf Tiger Hash单独散列的区块。然后由每对散列值计算内部Tiger哈希值。如果哈希没有成对，那么它将不变地传输到新的链。重复此过程，直到剩下-一个散列。

当分类帐使用Merkle树进行操作时，交易处理速度非常低。同时，优化水平很低，所以计算资源的负担非常高。因此，为了保持高速，需要许多强大、昂贵的节点，不过以后可能根本无法获得回报。在我们看来，这不是数据存储的合理使用。

5.5 RS分类账本结构

我们放弃Merkle树，并在RS系统中使用交易分类帐。每个RS条目由交易区块的哈希码组成，除了分类帐之外还添加到候选列表中。此外，这种条目在生成时具有节点标识符和时间戳。分类帐条目包含交易方向、其初始和最终账户、注销类型、注销单位数量、存款类型以及存款单位数量。这个原则增加了交易处理的速度，增加了非法分类帐变更的复杂性，并排除了分类帐入口的可能变化。也就是说，信息处理过程快速、准确地进行。

5.6 区块大小

时间是以搜索主节点和可信节点的周期为单位，周期时间是根据网络复杂度计算得到。每单位时间，网络包含从上一个周期结束时生成并转移到网络的N个交易，直到下一个周期的开始，以获得“添加到分类帐的候选者”的状态。从网络N选择的交易被放置在区块上。区块大小取决于其中的交易数量。

5.7 搜索交易参与者

RS点对点网络可以表示为图形，用户帐户以顶点的形式和多个可能的交易以连接两个顶点的有向边的形式。由于所有边缘都有初始和终点顶点，因此您可以随时构造一个有向图。

如果我们采取以下条件进行鉴定：

- 任何交易总是有发送者和接收者；
- 任何顶点总是可以连接到具有有向边（交易）的另一个顶点；
- 图形的任何顶点都具有有限数量的有向边（传入和传出交易）。

关于上述内容，我们可以说该图表包含了满足必要交易条件和构建一个简单链条所需的路线。

简单链是向图中没有重复顶点的路线。由于向图本身到目前为止还不为我们所知，因此，从图论角度来看，我们得使用未知的向图构建路线。众所周知，在这类向图中，遍历的长度将等于 $\Theta(nm)$ ，其中n是顶点数，m是边数。对于任何图形，存在长度为0 (nm) 的遍历，也存在具有最小可能遍历长度的图形 – $Q(nm)$ 。

未知图的遍历是我们最初不了解其拓扑的情况，并且我们只能在沿着图移动的过程中识别它。我们可以清楚地看到，从每个顶点中出来哪个边，但是只能通过从头到尾跟随每个边来找出它到达哪个顶点。实际上，这是一个机器人走迷宫问题的类比：机器人位于迷宫内，但不知道怎么从迷宫里出去。如果状态的数量受到严格限制，则机器人是有限状态机。这样的机器人是所谓的图灵机的完全模拟，其中磁带被图形替换，并且其单元不仅与顶点有关，而且还与向边有关。

5.8 数据传输通道

为了确保网络安全，验证器节点之间的所有数据以加密形式传输，并且基于网络库节点之间的每个连接是低等级的。如果数据传输发生错误，则线程应自动中断，相应的条目将被写入日志记录系统，然后到日志文件。数据通过代表变量传输。传输的数据都使用对称RC4算法进行加密。由于该算法在公共密钥下工作，所以当在节点之间建立连接时该密钥被传送。

5.9 系统中的行为

该系统中的一项行动是一次交易，其特征在于将值从账户到账户的最简单的转移或将合约方式的结果转移给验证者，以便随后在共识的搜索子系统中搜索解决方案。

为了防止具有相同标识符的同一块中的交易的重复，系统使用了一个协议，即唯一真实和正确的交易是首先发送到验证器子系统进行处理的交易。由于已经在验证器系统中记录了从当前帐户已经进行的交易，并且帐户中没有剩余的值进行交易，因此无法找到共识。因此，双重浪费的问题被解决。

当交易被执行时，信息被接收到验证器并被确认，关于分类帐状态改变的信息被自动地从可信任列表分发到所有节点，之后分类帐被同步。为了始终拥有一个最新的交易分类帐本，需要在每次所有节点的分类帐本中同步新到达的交易。为了解决这个问题，应该使用一个独立的同步端口（如果有这样的机会）。这个机会将增加验证器内核的处理信息的速度。同步线程总是执行，它是循环的。

分配随机存取存储器和中央处理器负载的优先级低于平均值。存储器存储最后1,000个操作和它们的帐户状态。这增加了对来自其他验证器节点的请求的响应速度。

5.10 添加交易进行验证

将交易添加到分类帐仅在协商一致的情况下立即从验证子系统中调用，并编制白名单，并将其中的交易保存在分类帐中。来自第三方系统的调用是不可能的，以提高安全性。

传入参数 - 表征交易的对象。结果值 - ResultValue <0 - 执行中止错误，结果值是可能的错误代码 /0 <ResultValue - 函数执行时没有错误，结果是分类帐中条目的编号。

传入参数是包含交易的唯一标签的对象、发件人、收件人、传输的值、值对应信息、所需值、传输值的数量、期望值的数量。

5.11 交易成本

系统使用RS代币，可以用于：

- 作为系统使用的内部支付方式；
- 交换系统内的不同货币；
- 交换系统内的各种值；
- 根据智能合约 创建和处理业务；
- 从系统中购买第三方来源的信息；
- 系统内支付服务。

交易的成本可以根据系统的负载水平而变化，我们建议使用物质方法来控制网络负载。

系统运行前三年执行交易的成本将针对不同类型的交易和经营单独设定。将来，将会开发一种用于自动生成交易成本的算法。

6. Rasmart分析

从概念角度来看，Rasmart包含Node（系统主要部分）以及为最终账户而创建的、根据Node提供API工作的一系列辅助模块（钱包，监视器，oracles）。

6.1 网络结构

网络由能够通过UDP协议（OSI模型的第四级）交换消息的节点组成。

从拓扑学角度来看，网络是覆盖网络（以太网上），使用唯一的节点标识符进行寻址，并且根据从节点标识符收集的有序方向图执行路由。

覆盖网络到真实网络基础设施的映射及其路由完全由节点软件的传输层执行。

6.2 网络中节点的识别和寻址

每个网络参与者都有一对非对称加密密钥：收件人的公钥进行加密，发件人的私钥进行签名。

节点和其网络流量的标识基于公钥的散列函数。

对于使用Diffie-Hellman算法的加密和电子签名，RSA算法（(Rivest-Shamit-Adlema)可以在密钥长度为1024位的素数字段上使用，ECDHE算法（椭圆曲线加密算法）也可以使用。

ECDHE算法使用时，对于可比较的加密强度，256位的密钥长度就足够了。

6.3 网络组织

网络是双向循环图。节点标识符被排序并锁定成环，使得对每个节点存在“左边的邻居”和“右边的邻居”。

创建消息后，节点将数据包发送给所有邻居。它们寻找邻居中的数据包接收者。如果找到了，它们会直接向接受者发送消息。如果没找到，那么那些“左边的邻居”将数据包发送给所有“右边的邻居”，那些“右边的邻居”将数据包发送给所有“左边的邻居”。

流量冗余可防止UDP丢失、消息路径上的节点断开以及伪造数据包。

该方案仅适用于所有节点都能够随时接收UDP数据包的情况。对此的主要要求是没有路由器的扁平网络和具有所谓“灰色地址”的分层嵌套专用网络。NAT外部的节点只有在响应其请求时才能接收UDP流量，并且该请求应来自其以前接受者，并且在一段时间内应该进行，该段时间是路由上的ARP表记录的生存时间。如果节点考虑到这些条件并调整其行为，则他们可以参与网络操作。通过分析入站流量和出站流量，节点可以了解要遵循的行为模型。

该分析的标准是节点是否向它尚未发送消息的节点收到消息。如果不是这种情况，节点安排其工作以便以“打开窗口”的间隔接收信息，并且该方式必须保持。

6.4 DHT

环组织使用哈希分布的相等性，这允许在有序的地址列表中查找邻居并在它们之间快速插入新出现的节点。

在torrent网络中使用的DHT算法以“度量”的概念运行，利用该概念可以补充选择邻居的机制，因此实际上它们位于不同大陆的不同数据中心。这是对捕获整个决策块并控制网络的女巫攻击的保护。

考虑到节点的响应速度，度量也可以优化交互的速度，这将由于非常接近的节点而增加网络的速度。

实际上，必须找到中庸之道：从安全性的角度来看足够可靠，并且从交易能力的角度来看足够快。

6.5 PEX

节点交换机制（对等交换）允许在连接新节点时重建网络。进入在有序图中，节点找到自己的位置，并且邻居重新分配自己邻居的子集，使得UDP的“波长”保持大致相等。

6.6 流量

网络在UDP上运行。节点生成的数据包具有发送者、接收者、命令以及此命令的任意有效负载（payload）。要创建广播消息，只需指定一个不存在的接受者地址（例如，0）。

许多命令使节点能够在网络上注册，请求丢失信息以同步分类账，创建交易，发送参与决策块的申请以及参与决策块所需的命令。

继续传输消息的节点使用短期签名对它进行签名。该签名的特殊功能是具有自己的生存周期（TTL），它更紧凑并不占用UDP数据包中的大量空间。鉴于TTL较短，这不是安全问题，但加快了对此类数据包的检查和处理，使得无法进行未经授权的干涉或“外部”数据包的发送。

在网络上发生短期签名的验证时，被拒绝的数据包不会超出第一次重定向，并且不会计入更高级别的节点软件。

大包被分成几个小包，它们作为单独的消息发送，并且仅在接收者节点上收集在一起。这是另外一个防止拦截的措施。

6.7 分类账结构

分类账是随机数据的基础，其样本在每个节点处存储和复制。在数据上构建了负责信息完整性、真实性和信息现实性的其他结构。

从技术上讲，数据库被以“键值”存储库的形式为组织，其中任意值的键是其哈希值。该数据库的本地存储库是可进入数据库Google的Level DB。

6.8 存储库和Merkle树

在数据记录上构建二叉树，其每个节点等于来自较低级别哈希的成对关联的哈希。

因此，整个数据树被“折叠”成一个“根”哈希，它显示整个数据库的状态。可以检查整个树并且在进行未经授权的更改的情况下，哈希将改变、不再与其他节点上的相同哈希，这将是违反数据库完整性的标志。这是一种Merkle树的修改算法。

6.9 交易链和区块链

交易链是一个记录列表，描述对数据库所做的更改并引用数据库的当前状态（Merkle树的根哈希）。这些记录被组合成由区块创建者签名的区块，并包括前一个区块的哈希，形成一个简单的链接列表 - 所谓的区块链（“区块链” - 来自英语“块链”）。该机制可保护已输入的数据免受更改，并对合法进行这些更改的参与者进行身份验证。

6.10 同步和完整性控制

由于数据库是一组键值，数据库同步作为节点之间单个记录的交换。交易链或Merkle哈希不一致时，确定改变和/或丢失记录的位置并且对网络进行其请求。这不仅允许更新最后信息，而且还确保在网络中长时间缺少节点或本地数据库中的数据不可逆转丢失之后初始同步和数据库恢复。

每次节点启动时都执行完整性监视，以及数据库的本地副本的计算状态是否与从网络接收的最新分类账状态不匹配。任何违规都会启动同步周期。

7. 数据库实体

7.1 零钱包

零钱包是网络中参与者的个人帐户。为了发送和接收资金，参与者需要一个已启动的节点和一对电子签名密钥。零钱包与节点的标识符是公钥的哈希。

7.2 交易

执行传输时，网络参与者创建交易，使用自己的私钥对其进行签名并将其发送到网络。金融交易是一个实体，其中包含与发送者、接受者、转账金额、操作货币和附加数据有关的信息。这些交易形成块链，是监视数据库正确性的机制部分。

7.3 任意实体

除转移信息外，分类账还可以包含任何格式的其他信息。但是，这种信息无法直接保存。每个这样的实体都拥有一个以智能合约为的所有者，可以创建、更改和删除分类账中德数据。

7.4 智能合约

智能合约作为存储过程存储在数据库中。它是一个程序代码，可以通过其标识符来引用。智能合约的标识符是其源代码的哈希值。

7.5 交易

参与者创建的交易尚未成为区块链的一部分，不包含在交易链中。在此之前，必须检查其正确性。此检查归结为两件事情：第一，发送者必须拥有比他要转移的资金更多的资金；第二，交易必须使用发送者的私钥进行签名。未通过的交易被拒绝，并在已确认的交易基础上创建区块，该区块由验证器签名并进入到区块链中。

7.6 分叉问题解决

区块链本质上容易受到“分叉”（“分叉”来自于英语“fork”）的影响。在存在单连接列表的情况下，树也可以存在 - 没有什么能阻止几个块引用一个父节点。如双链表中，这不可能“关闭”到后续块的链接，因为在写入分类账之后，区块无法更改。

像其他区块链一样，我们通过网络中任何时候只能有一个记录块来解决这个问题。验证器中的哪台将会在验证轮中得到解决。

7.7 决策模块

决策模块执行交易验证和新块的保存。这种一个节点的子集，彼此独立地执行检查，并相互交换解决方案的结果。在不可信的环境中 - 由最终用户控制的节点组成的分散式网络，原来是不可信的 - 这种机制允许检测受损节点并获得正确的结果。

这种方法的基础是“拜占庭将军问题”的解决，允许根据部分数据的可靠性做出正确的决定。

7.8 共识实施

共识就是关于交易有效性的决定；在所有节点交换关于发送给它们的交易是否有效的之后，可以达成共识。对准备记录的所有交易做出决定之后，选择记录区块的节点，然后决策模块从新节点重新形成并周期重复。通过这种方式，可以实现对分叉的保护 - 网络中始终只有一个节点可以添加新块。

7.9 决策模块并行化

如果所有收集的交易被分成子集，那么所有准备工作可以同时完成，并且从周期到决策周期，记录权将从一个记录节点转移到另一个记录节点。在这样的方案中，记录几乎是连续的，并且网络交易容量将增加许多倍。

7.10 将交易图划分为部分

以连接图形式表示一组交易，其中节点是发送者和接收者，并且它们之间的连接价格等于传输量的金额，允许将图形划分为子集。

为此，每个子集必须形成一个图形，其中每个节点的结果平衡是非负的。这个问题的解决与“背包问题”的解决一样。该功能的目的是“完成”准备好的交易列表，以便丢弃违反图形的非负性的交易的最小数量。

在此阶段丢弃的交易将转移到下一轮重新验证和“包装”。

7.11 佣金

佣金是一种激励网络参与者将其计算资源用于支持网络功能的方法。我们感兴趣的是让节点使用功能强大的设备并拥有高质量的互联网频道。更快响应的节点更有可能进入决策模块并获得完成工作的佣金。

7.12 佣金计算方法

该表包含了发布交易形成佣金金额（系统从发送者收取的费用）的原因列表。

交易佣金与交易轮之间的依赖性

- | | | |
|------|---|---|
| I. | <ul style="list-style-type: none">每轮交易数量。一轮交易次数越多，佣金越低。 | <ul style="list-style-type: none">每轮的最小交易数为1。名义价值=每轮10,000笔交易。一轮中处理的交易数上限设置为65,536。 |
| II. | <ul style="list-style-type: none">该轮中可信节点的数量。一轮中可信节点的数量越多，佣金越多。 | <ul style="list-style-type: none">名义价值为101。 |
| III. | <ul style="list-style-type: none">实际交易规模。交易区块链中占的“磁盘空间”越多，佣金就越多。 | <ul style="list-style-type: none">交易的实际规模从.....到...名义价值。没有限制价值。 |
| IV. | <ul style="list-style-type: none">发送者的“交易”积极性。发送者生成的交易越多，佣金越高。 | <ul style="list-style-type: none">交易积极性的下限为0。名义价值= $5,000 * 10,000$。交易积极性的上限设置为每秒1,310,720次交易。 |

8. 智能合约

智能合约是在创建交易时执行的程序代码。交易接受者作为程序代码的标识符。智能合约的标识符是其源代码的哈希值。

智能合约发布是包含其源代码的交易创建。智能合约被添加到数据库中，并在网络中其标识符作为接受者的出现交易时执行。执行时，智能合约会创建、修改或删除数据库中的某些实体，从而在调用之间存储其状态。

此外，智能合约可以从区块链中读取信息并创建交易。例如，收集对体育赛事的投注的智能合约，并且在数据库中出现关于该事件的信息之后，它按照所形成的“银行”的总分布比例分配奖金。它做出决定所依据的信息由另一个智能合约添加到系统中。因此，智能合约的本质是与外部世界有关的任何事实 - 货币报价，体育比赛的结果，公证人对文件进行合法认证等等。

智能合约使用Lua语言编写，并且具有隔离虚拟机内的所有功能。Lua标准库被禁用时，智能合约无法与外界交互。

智能合约仅在特别编写的SDK之内接收所有必要的功能。

在接收者作为智能合约标识符的交易接收过程中，决策块中的参与者执行其代码，并且在其完成时交换虚拟机状态的哈希，包括所有变量和创建的实体。如果决策块中所有参与者执行智能合约的结果一致，则将交易保存在区块链中，并且所有修改的分类账实体通过网络同步。

从算法角度来看，智能合约不受限制，通过导致永恒的周期的寄生或错误的代码，这会带来网络故障的风险。为了避免这种情况，智能合约是根据严格限制的时间执行的，然后它将停止并且交易被视为无效。这种交易的费用无论如何都是收费的，并在决策块的参与者之间分配。

智能合约可以更改的实体应该由他初始化。任何其他实体都可以被它读取，但不能被记录。

智能合约没有输入和输出方式，他们无法执行网络请求。他们所能做的就是从任何实体的分类账的本地副本中读取，以及实体创建、修改和删除。

8.1 虚拟可执行机

Lua解释器提供一个基于寄存器的独立虚拟机，其中程序按顺序执行。在虚拟机内，周围操作系统的资源不可用，必须提供所有库（包括标准输入输出库）。因此，保证虚拟机只能访问专门编写的SDK，并且不能写入文件或建立网络连接。

8.2 价值条款

Rasmart加密货币还是合约单位的价值，用于比较两个完全不同的单位，以便达成共识。RS加密货币作为用于实现价值转移功能的桥梁。这是可能的，因为任何价值对于RS货币而言都是流动的。

8.3 数据源

为了正确和完全成熟的工作，检查和提供附加信息，以使更平衡和最优的解决方案，RS平台使用第三方数据提供商。这是由于关于一个合约方的公开信息不足。例如，获得借款人的信用状况作出决定是否发放信贷。

该平台可以调用集成总线，允许与第三方系统的数据系统合作。通过远程访问，并以RS币支付的付费形式为系统参与者生成对第三方系统（站点）的数据呈现格式的请求。

请求以加密形式发送到为信息系统提供的端口和地址。请求的结果可以是对包含作出决定的必要信息的服务的任何响应。

8.4 API

每个节点都提供一个REST API，用于向网络发出请求。请求可以作为交易创建、智能合约发布、交易列表读取以及从分类账中检索值。

API仅接受本地请求，对于计划与网络和分布式分类账交互的任何软件，必须拥有自己的本地节点。零钱包是一种执行传输的应用程序，是第三方软件的一个示例，它通过直接安装在同一工作站的节点的API与网络进行通信。

9. 安全和可能的威胁

9.1 中间人攻击

通过交易的电子签名解决，该交易验证发送者。

9.2 51攻击

随着网络的增长，进入验证轮的概率可以忽略不计。

9.3 RSA密钥破解

数学证明了2048位（256字节）的RSA密钥的可靠性。由于缺少量子计算机，使用舒尔算法的攻击是无关紧要的。

9.4 女巫攻击

在受损节点的“全面包围”情况下才可能。一个可靠的节点足以检测到攻击。在选择可信验证节点的过程中，该过程也因熵元素而变得复杂。

9.5 “双重浪费”问题

只有一个记录节点可以保护网络免于“分叉”，因此消除两次使用相同资金的可能性。

9.6 流量拦截和替代

同一个数据包被多次复制并以不同的方式进行。要拦截流量，需要控制整个网络环境。就互联网而言，这似乎不太可能。数据包还提供轻量级的电子签名，足以验证短期的数据包。

9.7 本地存储改变

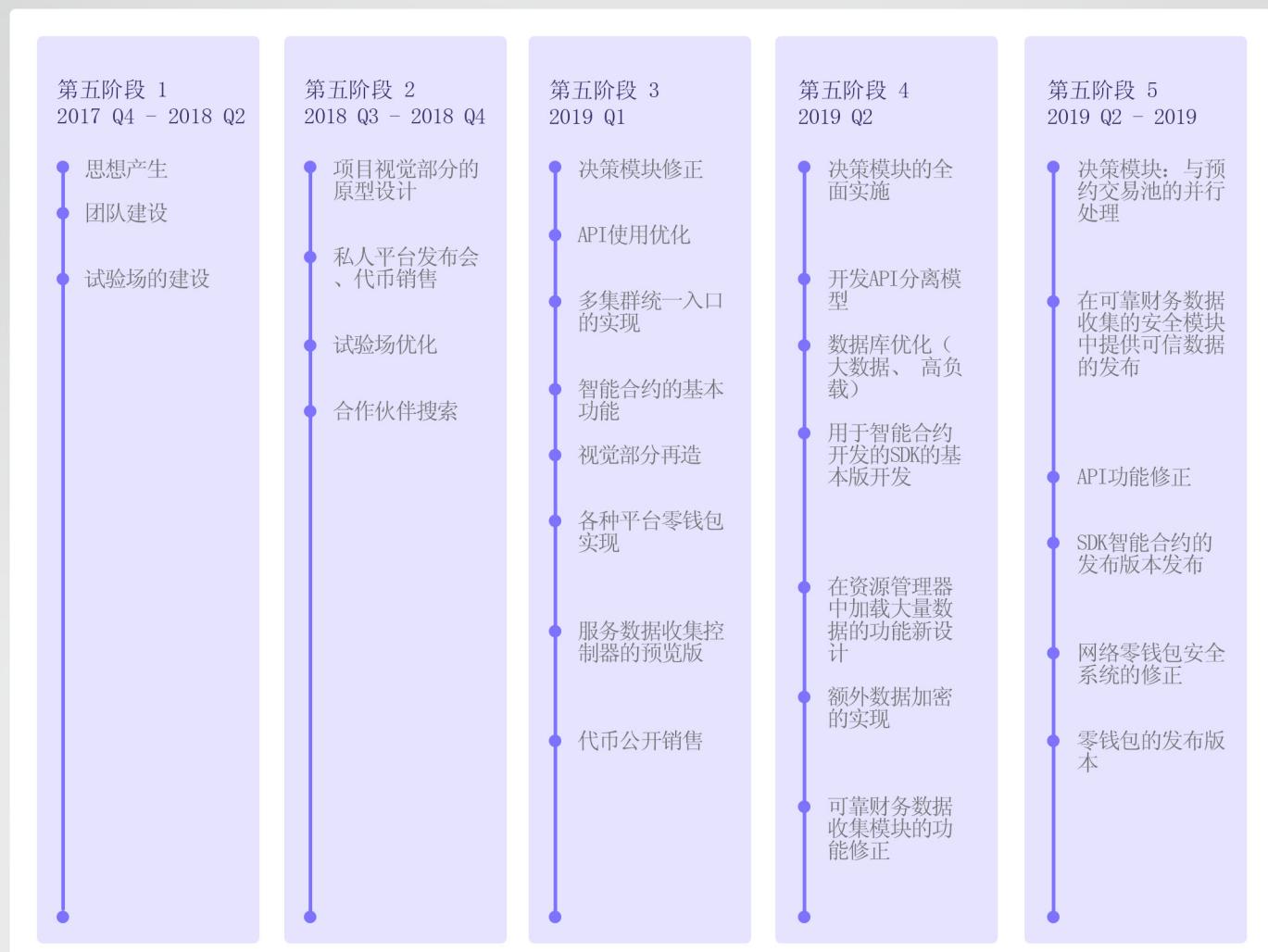
对本地存储的任何干预都导致重新计算数据库的状态（Merkle树的根哈希值和最后一个区块）。如果这些数据与实际数据不符，恢复之前网络将停止与受损节点交互，直到它被恢复。

10. 实施计划

10.1 项目技术实施计划

10.2 ICO

路线图



代币特征:

代币: RAS

代币类型: 实用型代币

代币特征:

代币价格: \$0.23

硬上限(hardcap): \$50,000,000

发行: 500,000,000RAS

交易最低数量: \$

接收货币: ETH

公开销售:

500 000 000 RAS – 代币首次发行

225 000 000 RAS – ICO

67 000 000 RAS – 营销和顾问

90 000 000 RAS – 平台支持

59 000 000 RAS – 储备基金

59 000 000 RAS – 团队和合作伙伴

ICO 表

阶段	持续时间, 天	最小购买, 美元	奖金, %
私人销售	60	10 000	30
预售	20	5 000	20
主要的销售阶段	10	500	5-10

* - 对于在所有阶段上的大量购买, 可以获得奖金。代币将从公司基金或团队利率中分出。

10.3 Rasmart加密货币

系统工作版的启动之后，将发行代币的固定数量，金额为500 000 000RS。然后RS代币可以兑换在首次销售期间发行的ERC20代币。交换可以按固定的兑换率速率进行，即1个ERC20代币=1个RS代币。

ICO之后我们打算将交易速度提高到每秒400,000次交易。

当前智能合约机制是基于一个又简单又有效的Lua程序语言。其安全毫无疑问。为进行金融交易，我们打算对语言安全进行深刻审计并研究其虚拟机的保护方法。

我们计划通过开发第二级路由机制来提高网络速度，这有助于加快搜索每个节点的最快路径。

