

技术白皮书

(可能会添加一些细节)

去中心化金融系统

CREDITS

Version 1.5/12.09.2017

Contents 目录

Abstract摘要	3
Introduction引言	3
1. Network ledger网络分类账本	4
Definitions定义	4
Network Nodes网络节点	4
The Last Saved Block最后保存的区块	4
Synchronization of nodes节点同步	5
2. Network Consensus网络共识	5
Consensus Comparison共识比较	5
The Concept of the Main Network Node主网络节点的概念	6
Equipment of Network Nodes网络节点的设备	7
Consensus Building共识建立	7
Building and Initiating the ledger建立并初始化分类账本	7
Transactions not Included in the Register不在注册表中的交易	8
3. Transaction Processing交易处理	8
Transactions交易	8
Consensus Building共识建立	8
Transaction Processing共识处理	8
Ledger Entry Structure分类账本条目结构	8
CREDITS ledger Structure CREDITS分类账本结构	9
Block Size区块大小	9
Search for Transaction Participants搜索交易参与者	9
Data Transmission Channel数据传输通道	9
Action in the System系统中的行为	10
Adding a Transaction for Validation添加交易进行验证	11
Cost of Transactions交易成本	11
4. Smart Contracts智能合约	11
Introduction介绍	11
Entities实体	11
Smart Contract Method智能合约方法	12
Virtual Executable Machine虚拟可执行机	12
Value Term价值条款	12
Performing the Smart Contract Terms执行智能合约条款	12
Data Sources数据源	13
5. Implementation Plan实施计划	13
Technical Plan of Project Implementation项目实施的技术计划	13
CREDITS cryptocurrency CREDITS加密货币	14

Abstract摘要

CREDITS平台是一个去中心化的金融系统，基于点对点（P2P）原则参与者之间可直接进行交易。该平台扩大了在分布式分类账、自我执行智能合约和CREDITS加密货币基础上使用金融服务的潜力。该系统旨在将所有参与者团结在一个站点上，为他们提供创建和使用金融服务的平台；每个人都可以提供服务并使用它。由于具有明确和均衡的技术系统，CREDITS平台为现代去中心化金融服务开发提供了新的技术解决方案和网络参与者互动交互的全新概念模型。

Introduction引言

完全点对点布局用于服务提供系统，它允许直接在参与者之间形成金融服务：货币转移，货币和价值交换，信贷，基金和其他服务。根据系统参与者的平等原则，一切服务都不需要额外的中介。因此，每个人都会获得更便宜、更快、更好的服务。

世界正在基于点对点平等原则向着人民之间的直接互动发展。一场革命发生了！从推翻大众媒体就可以清楚地看到：直到20世纪90年代，报纸，杂志和电视都是信息的主要提供者。今天，意见领袖是博客，是存在YouTube频道和社交网络上的，热钱投资于众筹和ICO，信息存储在去中心化云系统中。

金融业也许是少数几个滞后的行业之一，抵制了去中心化的引入和参与者之间的直接互动。虽然在技术上，创造去中心化的金融服务比创建无人驾驶车辆容易得多。

相应的技术环境是需要的，以建立基于分布式分类账的去中心化金融产品和服务体系：

1. 高执行速度（以秒为单位），以及以低成本交易，同时处理大量交易（数十万/秒）的能力（用于微支付和非现金交易）。

2. 建立一个系统，将所有参与者和性质上的金融去中心化服务所需的物品结合起来：用户个人化，KYC，信用记录局，法定货币结算中心，提款和兑换加密货币等。

这是目前阻碍点对点金融产品发展的两大基本任务。

我们向您介绍这些任务的解决方案，我们在CREDITS金融系统的帮助下实现。

CREDITS单一技术去中心化平台可以结合所有金融服务参与者，使用分布式分类账的原则可以安全快速地执行所有交易。自我执行智能合约和联合投票系统的原则为所有参与者创造了与各种金融产品互动的独特机会。该平台开启了新的巨大市场和在金融和其他行业使用区块链项目和服务的新潜力，以前由于速度和交易成本的限制而无法使用。

1. Network ledger网络分类账本

Definitions定义

1. 系统是一组分散的网络节点，执行处理，保存交易，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. 一个网络节点就是一台计算机，它安装完整网络客户端，连接到公共系统，验证交易并将其写入分类帐。
3. 分类帐本是系统确认并存储在所有网络节点上的交易列表。
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. 智能合约是系统项目、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。智能合约的关键特性是其去中心化和独立于中心来源。
6. 智能合约的方式是程序代码，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本中。
7. 缔约方是最终的网络参与者和系统用户。

Network Nodes网络节点

我们使用几种类型的节点，这取决于其基于免费访问和节点连接构建去中心化网络的目的:

1. 公共节点 (OY) 是参与有效性的交易验证的节点，但具有最小的信任因子。它也是可靠节点的角色候选者，并且是网络中下一个周期中的节点角色选择中的当前处理节点。
2. 信任节点 (DY) 是参与交易验证的节点，具有最大信任因子 (1)，是当前处理节点和公共节点的角色候选。在数学计算数量的选择和节点之间的投票周期中，该节点不能被信任。数学计算取决于节点的数量和网络的复杂性。
3. 网络的主节点 (TY) 是参与验证的节点，负责向交易分类帐块添加交易。在投票周期的数学计算数中，该节点不能成为受信任的节点或当前处理节点，该数学计算取决于节点数量和网络复杂度。

该系统使用信任因子 - 从0到1的绝对分数值，以可信节点数+1的数学方式表示为网络中的总节点数。可信节点的最大数量不能超过网络节点的50%。

The Last Saved Block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。

通过分类帐区块内容，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。这节省了网络带宽。

在同步期间，首先只检查块号。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（LR）。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区块，以便始终保持相关最新信息。所有区块形成一组等待添加到分类帐的交易记录。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，可靠性越高。

Synchronization of nodes 节点同步

每个新节点在确定定义和彻底信任验证后启动并同步。为了提高信息处理速度，所有进程都是同时处理的，彼此独立。如果没有输入变量，则创建一个空的分类帐存储 - 在RAM中保留一个空格以进一步简化访问。在所需分类帐不可用的情况下，向受信任节点发送请求以接收为同步帐户创建的所有交易。

如果输入参数是表征交易的对象，则开始在所有运行的同步线程中进行搜索。该操作产生数字代码 - 当前线程的可信节点分类帐中的位置编号或值小于零的错误编号。如果线程方法以连接错误结束，则线程将完全结束。

2. Network Consensus 网络共识

CREDITS的共识是群体决策的一种方法。目的是开发所有网络节点可接受的最终解决方案。

Consensus Comparison 共识比较

去中心化CREDITS分类帐原则的定义，以比较不同类型的共识：

- 分类帐可用性（节点可以随时将数据写入分类帐并从中读取）；
- 所有参与网络节点的可修改性；
- 所有系统节点的一致性（所有节点看到分类帐完全相同的版本，更改后能更新）；
- 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）。

对比参数	Credits 特殊的 PoW 和 PoC	PoW	PoS
识别生成区块节点的原则	数学函数计算。确认最后一个分类帐副本的存储。	执行数学函数的迭代计算，具有变化的复杂性。	搜索参与者（竞争节点）中的最大堆栈。

51%攻击	不太可能，因为有必要在资源中有一个完整的分类帐和计算能力，并且动态选择可信节点。	可能，但在使用资源方面会非常昂贵。	可能，但昂贵，因为需要增加自己的堆栈。
对在现场完成的工作进行补偿，以添加到分类帐/区块链	自动计算取决于每个操作的佣金。	提供区块挖掘的修复。	提供区块挖掘的修复。

The Concept of the Main Network Node 主网络节点的概念

所有网络节点都是去中心化的，没有优先级。需要定义将要处理存储在不同网络节点处的交易队列的网络节点。之后，它必须在分类帐中写入新生成的交易区块。

CREDITS平台使用自己的组合协议来提高交易处理速度，保证数据存储、处理和传输交易的完全安全性。该协议是基于所有分类帐交易的数学函数的计算，应用工作证明原则。它通过计算整个内容值的校验（哈希码），来准确地确定该节点上的分类帐和软件更新最新版本需要的存储量（容量证明）。文件的大小也被确定为证明这是最新的副本以及系统中记录的最新事务的哈希码。

为了成为主要的网络节点，节点根据最后一个存储的分类帐搜索其计算的散列函数值。我们在网络节点之间组建健康的竞争环境，使得有机会成为主要节点；并生成和存储新的分类帐。

在计算函数并获得结果后，将其发送到所有网络节点进行验证。结果包含计算的时间戳和基于分类帐文件和软件功能计算的值。所有节点接收计算值，比较分配给主网络服务器的计算时间，验证并确认节点的信任因子，并确认其参与竞争的机会 - 成为主要网络节点。

在收到所有网络节点的确认后，由正确计算函数值并包含时间戳的节点形成一个列表。接收到正确结果并在最快时间内获得确认的节点成为当前的主要网络节点。

SHA2算法概念用于计算文件的散列和。

SHA2系列的哈希函数是建立在MerkleDamgard结构的基础上的。

加入后的初始消息被分为区块，每个块分为16个字。该算法传递每个消息块通过64或80次迭代（轮）的循环。在每次迭代时，将转换2个字，其余的字定义转换函数。每个块处理的结果被总结。和是哈希函数值。然而，内部状态是基于先前区块处理的结果而初始化的。因此，不可能独立处理区块并合计出结果。

Equipment of Network Nodes 网络节点的设备

我们正在努力建立一个具有最快可能交易处理特性的平台，因此我们建议使用物质激励来维持网络节点处于最佳状态：高性能服务器设备和高互联网带宽。

作为物质补偿，主网络节点的所有者将从其处理分类帐的每个交易的多个佣金中获得 CREDITS 货币的报酬。其余（ $\frac{1}{2}$ ）旨在用于用户支持、当前功能和新产品开发的总体项目开发预算。百分比可以改变，也通过网络节点的联合投票将其分配到速率形成系统，在初始货币发行至少三年之后。

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。

Consensus Building 共识建立

因此，我们有所有节点选择的主网络节点。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。建立最后一个相关分类帐的结果就是共识解决方案。

整个过程可分为以下几个阶段：

1. 搜索主网络节点；
2. 建立可信节点；
3. 接收交易列表，并建立一个候选名单待列入分类帐；
4. 处理候选名单，表决节点（信任和公共节点具有不同的权重因子（信任因子））；
5. 从候选名单中移除未确认交易，未经验证的或未确认的；
6. 建立要添加到分类帐的已确认交易列表；
7. 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；
8. 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注册表中。

Building and Initiating the ledger 建立并初始化分类账本

整个过程可以按照以下顺序进行：

1. 系统网络的终端用户生成一个交易。
2. 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）。
3. 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。
4. 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。
5. 一个数字被分配给该区块，由一个时间戳和一个转换成哈希码的节点标识符组成，然后该区块被放置在共识模块中。
6. 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。

7. 这个哈希码是区块的一种签名，也是创建这个带有交易信息区块的签名。
8. 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。

Transactions not Included in the Register 不包括在注册表中的交易

未包含在即用交易列表中的交易被标记为被拒绝。关于此的信息将立即显示在交易的发送方（发起方）。

不包括在分类帐中的交易保留在候选组中，并存储在网络节点中。服务器收到的所有新交易在协商一致时都到达那里，然后搜索过程重新开始。网络的这种连续循环操作允许在相当短的时间段内执行交易，同时保持信息的高度可靠性和相关性。

3. Transaction Processing 交易处理

Transactions 交易

交易是系统的最小单位，通知平台执行合约方法或直接在账户之间转账而不创建智能合约，然后将结果放在点对点网络中。

Consensus Building 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，以及共同构建算法 – 一种有限状态自动机通过的算法。通过循环（时间步长），每个时间步长，交易被提取并放置于池（一维数组）中进行共识。放入池后，所有交易都将被发送到受信节点以接收响应。如果接收到响应，则发送请求的交易可以被添加到该验证器的分类帐中。之后，它被发送到网络中的下一个验证器。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账的标记。

Transaction Processing 共识处理

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，而不管其余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。并且总是给出输出结果 – 正的、任何其他或错误。因此，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。

Ledger Entry Structure 分类账本条目结构

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（TTH - Tiger Tree Hashing）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。接下来，在每个对的链中再次计算内部Tiger Hash。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，计算资源的负担非常高。在我们看来，这不是数据存储的合理使用。

CREDITS ledger Structure CREDITS分类账本结构

我们放弃Merkle树，并在CREDITS系统中使用交易分类帐;每个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，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。

Block Size 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。

Search for Transaction Participants 搜索交易参与者

CREDITS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（帐户）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（orgraph）。

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（帐户）总是可以连接到具有有向边（交易）的另一个顶点;
- 图形（帐户）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。由于它是顶点的有限序列，其中每个顶点（除了最后一个）通过边缘连接到序列中的下一个顶点。

Data Transmission Channel 数据传输通道

主网络节点和CREDITS网络的公共节点之间的每个通信通道是单独的线程（多线程），当执行交易时，数据以加密形式发送。

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，并按照DiffieHellman算法以加密形式发送。

RC4算法像任何密码一样，是建立在一个伪随机位发生器的基础上的。该密钥被写入发生器输入中，并伪随机数位在输出端被读取。密钥长度可以是40到2048位。生成的位具有均匀的分布。

DiffieHellman算法允许双方使用未受保护的信道接收公共密钥，从而不受通信信道改变的影响。所接收的密钥可用于使用对称加密来交换消息。该算法基于计算离散对数的复杂度。在其中，如在具有公共密钥的诸多其他算法中，计算被模除得到一个大的素数P。

首先，以特殊方式选择一个小于P的自然数A。如果我们要加密值X，那么我们计算

$$Y = AX \text{ mod } P.$$

有了X很容易计算Y。从Y计算X的逆问题是相当复杂的。指数X被称为离散对数Y。因此，知道计算离散对数的复杂度，可以在任何通信信道上公开发送数值Y，因为具有大的模数P，初始值X几乎不可能被选择。用于生成密钥的DiffieHellman算法是基于这个数学事实。

系统中的任何操作都与时间戳，先前区块的数量，用户的登录名和智能合约ID相关联。这样可以在执行时查找重复项。如果发现重复，那么我们从池中获取第一个交易，其余的被认为是非法的。

Action in the System 系统中的行为

该系统中的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。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。

为了在当前验证器节点的所有可信节点中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）。由于端口上的负载分配，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RAM和CPU负载（使用CPU周期）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（使用同步算法以加密形式），这增加了对来自其他验证器节点的请求的响应速度。

Adding a Transaction for Validation 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 - 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、期望值的数量和其他系统信息，必要时可以更改。

Cost of Transactions 交易成本

系统使用CREDITS货币，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用于服务。

交易的成本可以根据系统的特定用户的网络负载而变化，这在理论上可以在一定的高峰时间引导大量的交易。我们建议使用物质方法和对系统用户的影响来控制网络负载。

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。

4. Smart Contracts 智能合约

Introduction 介绍

CREDITS系统中的智能合约是一种电子算法，描述了一系列条件，通过这些条件可以将现实世界或数字系统中的动作和事件相关联。

要实施自主控制的智能合约，需要完全排除人为因素的去中心化的环境，并且要使智能合约成本转移，则需要独立于中央机构的加密货币。

Entities 实体

CREDITS中的智能合约由以下实体组成：

1. 属性（公共变量） - 系统实体存储CREDITS系统中合约工作所需的公共数据。
2. 方法是CREDITS系统实体，负责在进行交易（合约中的行动）时观察行为的逻辑和顺序。

CREDITS系统的参与者使用修改合约属性的方法签署智能合约，启动验证合规条件和协调的过程。

智能合约在双方签字后生效。为了确保自动履行义务，必须使存在完全自动执行合同条款的环境。这意味着智能合约只能在不受限制地访问智能合约项的可执行代码的环境中存在。

所有合同条款必须具有数学描述和明确的执行逻辑。因此，智能合约的主要原则是双方之间契约关系的完全自动化和可靠性。

Smart Contract Method智能合约方法

CREDITS智能合约方法是系统实体，负责遵守交易期间的行为逻辑和顺序（合约下的行动）。

动作的逻辑和顺序由包含命令的程序代码（模块）描述；其顺序执行允许获得所需的结果。该代码可以处理系统命令（例如，分配命令），用户命令（单独编写的函数），合约属性（可以从任何合约方法进行静态或动态初始化变量），以及只能所有者连接（第三方）合约的任何其他第三方合约的方法。为了更普及，开发是使用脚本语言（例如JavaScript）。

该方法（程序代码）允许使用所有广泛使用的脚本语言操作符（命令）（赋值，条件和无条件跳转），创建函数和过程（子程序），连接第三方库。

Virtual Executable Machine虚拟可执行机

CREDITS系统的合约方式在系统的虚拟环境（Virtual Machine，以下简称虚拟机）中执行。当为特定合约调用方法时，VM会分配一个内存区域并加载其中包含方法和初始化变量的合约字节码（或在调用其他合约方法时重新定义）。VM开始处理方法字节码，在运行时，变量和代码被加载到其存储区域中，并且命令被连续执行，它们的结果被传送到点对点网络以便随后存入分类帐本中。

执行方法的发起者是系统的用户，代表该方法的启动。

Value Term价值条款

CREDITS加密货币还是合约单位的价值术语的一个指标，用于比较两个完全不同的单位，并在双方执行或接受合约时达成共识。不必注册每个单独的值/网关组合，CREDITS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CREDITS货币而言都是流动的，这意味着任何价值对于任何其他价值都可以是流动的。

Performing the Smart Contract Terms执行智能合约条款

CREDITS系统中的合约期限是触发（已检查）关闭（完成）合同所需的字段的值。

完成智能合约条款是当检查到相应的期望值的触发（期望）字段时的过程。找到解决方案以满足合约条款有三种可能的方法：

1. 两个或多个缔约方之间缔结合约转让价值。在这种情况下，合约履行是从接收方向转让方提供相当价值的成本。
2. 合同双方交易价值签订条约，但必须在履行一定条件（例如向接收方交付价值）的情况下付款。
3. 将一个价值转换为另一个价值的合约，以CREDITS代币的形式成本等价物放置在系统中。在这种情况下，平台开始寻找通过其他合约中的转换来交换另一个价值的最短路径。合约的任何履行都可以通过一次交易或几次交易完成，这将提供一个机会来收集所需单位数量的价值来完成合约。

Data Sources 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，CREDITS使用第三方数据提供商。需要在系统中引入额外的数据源是由于关于一个或多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）。

要使用第三方信息系统，该平台可以调用集成总线，通过远程访问，并以CREDITS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由标准信息系统的信息以外的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，或者表示不可能接收所需响应的错误代码以及消除错误的可能步骤。

5. Implementation Plan 实施计划

Technical Plan of Project Implementation 项目实施的技术计划

	S1	S2	S3	S4	S5
	Pre-Alpha	Alpha	Beta	Release candidate	Release
储存, 共识 mFA 共识	FA : 关键 设计实现	mFA : 关键 设计实现 PoW (Proof-of-Work) 和 PoC (Proof-of-Capacity)	mFA 优化	—	—

数据储存	去中心化分类账本, NoSQL 储存实现	信息包历史记录	-	区块链备份	-
CVM (Credits虚拟机)	设计和实现	与生态系统整合	优化	检查错误	-
第三方系统	-	设计和实现	集成到整个系统	-	优化
推理机	正式规范和关键设计要素	推理整合到区块链	推理优化	-	-
用户界面	实现	网页用户体验设计	-	-	-
钱包	钱包正式规范		用户体验设计应用测试	-	Android, iOS, 桌面 钱包
RPC & REST API	正式规范	区块链浏览器	-	第三方浏览器	-

CREDITS cryptocurrency CREDITS加密货币

系统release版本发布后，将发行固定数量的10亿个CREDITS。它们将被交换为在初始代币发售期发行的ERC20标准代币。他们将以固定汇率兑换：1 ERC20标准代币= 1 CREDITS 货币单位。