Abstract: 内容摘要：为解决加密交易所和交易平台带来的问题，Aphelion 推动了分布式账本技术 (DLT) 的发展，它是建立在 NEO 区块链之上的开源、点对点 (P2P)、去中心化资产分配应用协议。Aphelion 旨在推动被称为分布式交易所资产账本或 DEAL 的代币化交易。智能合约促进了 Aphelion DEAL 交易，由用户创建且不受交易所或交易平台及其相关限制的约束。Aphelion 代币是流动性验证设备 (LVD)，它直接在用户之间实现了 DEAL：即时、安全且自由。

Disclaimer: 免责声明：本白皮书不构成出售证券或股票的报价或邀约，仅供参考。APH 代币是建立于区块链技术之内的实用工具。Aphelion 代币 (APH) 不作为股票或证券销售，也不授予股权或投票权；不论以直接还是间接方式，Aphelion 代币；其亦不以直接或间接的方式授予 Aphelion 公司及其实物财产、虚拟财产、知识财产的所有权不授予债券也不是债务工具；其不向代币持有者支付配送款项、支出款项或利息。如果日后募股，我们将通过恰当的保密渠道完成此过程，并遵守所有必要的法律要求。根据美国证券交易委员会的最新公告，Aphelion 不会向任何美国公民或居民进行销售或接受其出资。根据中国证券监督管理委员会 (CSRC) 和中国人民银行 (PBOC) 的相关规定，Aphelion 不会向中华人民共和国 (PRC) 的任何公民或居民进行销售或接受其出资。根据新加坡金融管理局的相关规定，Aphelion 不会向任何新加坡公民或居民进行销售或接受其出资。

Notice to citizens and residents of the United States of America: **美利坚合众国公民及居民通知书**：本网站及募股说明书还未作为登记声明的一部分向美国证券交易委员会 (SEC) 登记备案。因此，本网站、募股说明书以及其他任何与 APH 代币的报价、销售、认购或购买邀约相关的文件或资料都不得传播或发布，也不得发售或出售 APH 代币，或以直接或间接的方式向美国居民发送认购或购买邀约。对于中华人民共和国的居民和公民（出于本文件及募股说明书的目的，香港、澳门和台湾不包括在内）：不得以直接或间接的方式将 APH 代币销售、发售或出售给中国公众，此文件和募股说明书都未提交至中国证券监管委员会，这些文件以及其中包含的任何与 APH 代币相关的募股资料或信息都不得提供给中国公众，或用于任何让中国公众认购或购买 APH 代币等相关的目的。本网站中包含的信息及募股说明书不构成在中国境内销售、邀约购买或推销任何 APH 代币的建议。

Notice to prospective subscribers in Singapore: 新加坡潜在认购者通知书：本网站及募股说明书还未根据证券及期货法 (SFA)（第 289 章）在新加坡金融管理局注册为招股说明书。因此，

本网站和募股说明书以及任何其他与 APH 代币的发售或销售以及认购或购买邀请相关的文件或资料都不得传播或发布，也不得发售或出售 APH 代币，或以直接或间接的方式邀请新加坡居民认购或购买 APH 代币。

Table of Contents:

1.简介
分布式账本、区块链技术、加密货币及其智能合约正在成为众多行业的搅局者。事实上，专家认为它扰乱世界的程度甚于以往的任何行业。我们已经见证了其在金融行业的迅猛发展。而作为这项新技术的一部分，开发人员正以惊人的速度建立新的工具，正展开激烈竞争，试图找到公众及其机构需要的主流安全解决方案。

1.1 背景

作为区块链生态系统的一部分，比特币 (BTC)、NEO（前身为小蚁）和以太坊 (ETH) 等加密货币已成为数字资产分配的早期领导者。以区块链技术和分布式账本为基础，中本聪于 2008 年开发了首个加密货币，即比特币 (BTC) [1]。从此，很多加密货币便纷纷出现，其市值也出现了前所未有的增长 （2017 年增长了 1000% 以上）。据企业家、风险投资家、银行家和其他专家推测，加密货币最终将成为新的通用货币，因此也会派生一些新的企业。但是，新兴加密货币背后的区块链和分布式账本技术可能更为重要。

1.2 区块链技术

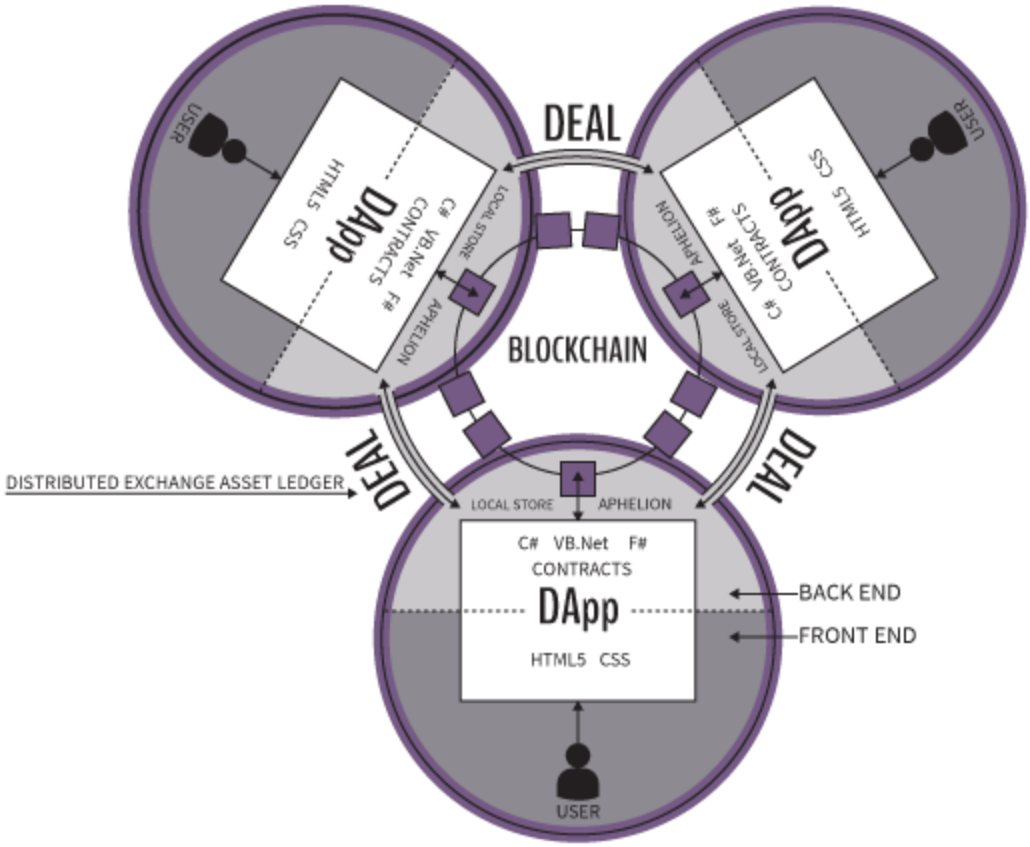
加密货币是建立于区块链技术之上的。“区块链技术是一种可靠的经济交易数字账本，它不仅能记录金融交易，还能记录几乎所有有价值的东西。”[2] 什么是区块链技术？“区块链作为一种历史悠久的底层结构，可以准确地记录所有发生的事情。它会将数据添加到永远无法修改的加密区块中，然后将各个部分散布到由分布式计算机或“节点”组成的世界网络中。区块链始终有一个不变的“账本”可供查看、验证和控制。同时，它没有任何能够用来攻击或破坏记录或数字资产的破坏点。得益于它的分布式账本技术，区块链被应用于各种数字记录和交易中，我们也开始看到有意做出改变的重要行业。

1.3 分布式账本技术

“分布式账本是一种跨越多个网站、区域或参与者的数据库。正如大家所预料，分布式账本必须是去中心化的，否则就和现在大多数公司使用的集中式数据库并无二致。去掉此过程中的中间方正是分布式账本技术这一概念的吸引人之处。此外，企业可以使用分布式账本技术来处理、验证或认证交易或其他类型的数据交换。当多方达成共识时，记录就被储存在账本之中。每条储存在分布式账本中的记录都具有时间标记以及各自的加密签名。分布式账本的所有参与者都可以查看任何有问题的记录。此技术提供了所有被储存在特定数据集中信息的可验证、可审核历史。分布式账本技术在金融圈和政治部门中常被称为 DLT。”[4] 利用 DLT，Aphelion 将以安全、可靠且真正去中心化的方式对 P2P 交易进行去中心化，通过其代币而非交易所来加速这一过程。绕过交易所，让 DEAL 在基于 NEO 区块链的分布式账本上产生，将是跨向加密交易未来的一大步。

1.4 去中心化应用 (DApp)

去中心化应用或简写为 DApp，其后端代码是在去中心化的点对点网络上运行的。DApp 可以有以任何语言编写的前端代码和用户界面（就像应用程序一样），并且能调用后端。此外，其前端可以托管在 Swarm 或 IPFS 等去中心化存储器上。如图所示，如果应用程序 = 前端 + 服务器，那么 DApp = 前端 + 社区 + 合约。Aphelion 合约是在全球 Aphelion 去中心化点对点协议上运行的代码。



1.5 PoW、PoS 和新一代 dBFT

从 PoW 到 PoS“工作证明 (PoW) 是负责网络高能源需求的比特币共识算法，它提供了需要大量资源的系统记账机制。比特币节点、挖矿区块和验证交易必须证明加密任务的表现，从而获得受欢迎区块奖励。所以，任何企图伪造 BTC 交易或损害区块链记录的人都必须超越所有其他的矿工以及他们为了保持比特币良好干净而投入的能源。由此我们发现，得益于 PoW，如果黑客想要对比特币区块链造成破坏，则必须投入相当于一座小的北美城市所消耗的总能源。其他众多加密货币系统最常用的 PoW 替代工具被称为权益证明或 PoS。PoS 非常有潜力，因为它无需区块链节点来执行艰巨或无用的加密任务即可让潜在的攻击付出巨大代价甚至阻止攻击。因此，这种算法将 PoS 区块链的功率要求降低至合理且易于管理的大小，而且可以让他们在无需大量消耗地球能源储备的情况下也能提高可扩展性。PoS 可以替代 PoW，虽然它的能效非常低，但在过去八年中它已经证明自身非常可靠。然而，两个系统都有一个关键问题，在仍然有些反主流文化的加密社区中基本上都没有解决。如果出于某种原因共识被打破，PoS 和 PoW 只能让区块链分叉成两个替代版本。事实上，很多情况下大多数区块链分叉只是为了在短期内又汇聚成一个单一真实数据源，如上图所示。而很多加密爱好者则常常将这个明显的缺陷当成是一种特性，因为它可以保存多个版本的真实数据并争取公开采用，直到产生一个决议。从理论上来看这非常不错，但是如果我们想要看到区块链技术真正扰乱和/或增强金融行业，那么让区块链分成两个替代版本的潜在可能性是无法容忍的。拜占庭容错和 dBFT 拜占庭容错 (BFT) 这个词来源于博弈论和计算机科学中的拜占庭将军问题，它描述了这样一个问题的本质，即让互不信任的代理机构通过不良沟通在分布式系统中达成共识。BFT 算法以这种方式建立了区块链节点之间的关系，即网络对拜占庭将军问题具有可恢复性，即使有些节点显示出恶意甚至发生故障也可以让系统保持共识。为此，授权 BFT（或 dBFT）算法的 NEO 版本承认区块链空间中的两种参与者：一种是专业节点操作者，他们通过运行记账节点来获得收入来源，另外一种是想要获得区块链优势的用户。从理论上来说，PoW 和大多数 PoS 环境中不存在这种差异，但实际上大多数比特币用户不会操作矿工，他们主要分布于由专业人士运营的特殊场所。因此，需通过专门的记账节点间进行的共识游戏来进行区块验证，这些节点是通过一种授权投票过程由普通节点指定的。在每次验证过程中，其中一个记账节点是伪随机指定的，用以将其区块链版本传播至网络的其余部分。如果剩余节点中有三分之二同意此版本，则达成共识，区块链可继续执行。如果同意者不超过三分之二，则会重新指定一个节点，然后将其真实数据版本传播至剩余系统，以此类推直至达成共识。通过这种方法，想要成功攻击系统几乎是不可能的，除非绝大多数网络想要实施金融自杀。此外，系统是防分叉的，所以在每个指定的时刻只存在一个真实数据版本。无需解决复杂的加密难题，节点的操作更加快速而且能与集中式交易方法一争高下。”[5]

1.6 建立于 NEO dBFT 之上的 Aphelion

由于 dBFT 解决了在上述比特币 PoW 及其替代工具 PoS 技术中发现和概括出来的挑战，Aphelion 将作为一种生态友好、开源且完全去中心化的数字资产应用在 NEO 的基础之上建立，为数字资产分配创造最安全和去中心化的应用。这可以让用户进行 DEAL P2P 的交易，而不受交易所和交易平台及其带来的限制和挑战的影响。Aphelion 是一种代币化的 DApp 协议。**为什么使用 NEO？**达鸿飞（创始人）谈到：“NEO 有助于智能合约和项目的快速开发与部署，因为它可以让开发人员使用其熟悉的编程语言。我们以编译器的形式提供了各种高级语言。除了 .Net 和

Java 外，我们未来还将支持 Python 和 Go，如此一来超过 90% 的开发人员都可以使用。与以太坊相比，开发具有更平滑的学习曲线和更短的学习圈，这有利于快速引进项目。”[6]

效率
安全合约
开发语言
可扩展性
ASIC 机器上的 POW 消耗了巨大的能量
伪匿名导致交易缺乏诚信
C++
高峰交易被限制为每秒 3 - 4 次
GPU　矿工共同使用的能源数量比整个国家还要多*
脆弱的合约代码容易遭到黑客攻击**
Solidity 语言
目前的高峰交易达到了每秒 20 次
dBFT 通过高效的方法确保了目标的达成
完整的数字身份可用于实际应用中
C#、.Net、Java、Python 和 Go 覆盖了 90% 的开发人员
交易数量高达每秒 10,000 次

1.7 加密货币市场

“截至 2017 年 4 月，所有加密货币的综合市场价值为 270 亿美元，这相当于 AirBnB 等硅谷成功企业所创造的价值。”[9] 据 bitcoin.com 统计，在 2017 年 8 月末，其市值超过了 1800 亿美元，这意味着加密货币的总市值在这一年内增长了近 1000%。[12.区块链技术问题及随后的加密货币都非常新颖，所以交易平台和交易所存在很多严峻的挑战。目前，数字货币并没有像信息网络一样互相连接。货币现在的兑换模型在利用市场决定的汇率将小规模货币与其他常用货币关联时存在一个关键障碍。此外，交易所和交易平台实际上是作为一种集中式系统来操作的，因此不仅产生了相关问题还违背了去中心化的目的。现在的加密交易所和交易平台所面临的挑战：集中化：规则、费用、非流动性资产和交易所控制了用户钱包的私人密匙，从而使交易所具有资金的完全保管权。复杂性：交易平台和交易所在几乎所有技术层面都缺乏任何交叉一致性。准入障碍：加

入每个平台有不同的规则、批准延迟、传统货币存款与纯数字存款的比较、缺乏即时存款。使用上的挑战：交易被无故封锁、每日最高限额、体验糟糕的 UI、漏洞百出的软件、不够用户友好。延迟：不断的延迟以及性能问题。缺乏支持：完全缺乏客户支持，而且很多知名平台都无法做出回应；等待数周或数月才能得到回复是很常见的事。缺乏安全性：多重攻击、资金丢失、侵犯隐私、网站关闭。缺乏隐私：要求验证、信用卡、驾驶证扫描图、护照。

2.痛点：

2.1 加密货币挑战

由于比特币是一个相对简单的区块链系统，它需要额外的开发协议才能发挥交易所功能。NEO 也可以兼容多种代码语言，但 ETH 只能兼容 Solidity。“例如，你可能以为目前比特币和以太坊使用的工作证明 (POW) 共识机制可以获益，可实际上是有代价的。因为存在一个缺乏目的性的问题。比特币交易是最终目的吗？并非如此。该协议更注重可用性而非目的性 — 这意味着可能会出现分叉和单独的区块，如我们之前看到的，当出现严重的安全隐患或开发人员对于标准产生分歧时，比特币项目往往都会“分叉”。POW 的能源消耗也非常大，所以节点会花费很多电费。”[6]

2.2 集中式交易所

有几个加密货币交易平台和交易所被广泛使用。很明显，他们是 p2p 交易机制，但不是去中心化的。他们是交易发起者之间的中介机构，也因此带来了一系列的挑战。首先，交易所设定了相关规则，比如谁可以交易，什么可以交易以及交易时间。然而，用户账号甚至是发起的交易被无故删除或冻结的案例比比皆是。此外，还有因各种安全漏洞导致数以亿计（美元）被盗的事件。除了交易所面临的这些挑战之外，如今还有众多用户面临完全缺乏支持的问题。这些所谓的去中心化交易所根本不是去中心化的，而是恰好相反的。“从各方面来说，P2P 交易所并不优于普通交易所 — 也存在交易时间长、用例不直观以及流动性低等较为劣势的问题。去中心化交易所大多数缺陷的产生仅仅是因为他们是一个相对较新的服务。例如，Bitsquare 可以说是最早出现的去中心化交易所，但它只持续了三年左右，而且大部分是处于开发阶段。因此，这些交易所必须解决很多问题。例如，目前有大部分的交易所是以小范围内的特定加密爱好者为受众的，而且他们没有满足新用户需要的需求 — 所以，他们往往存在使用不够直观的问题。也因为这些原因 — 受众范围小并且在早期出现的去中心化交易所，其交易量一般比普通交易所要少很多。另外，交易时间长这一问题可能需要较长的时间来解决。这些问题是由于交易展开的方式所造成的 — 交易者必须等待真正的比特币和法定交易完成才能结束交易。最后这个问题再加上低流动性的问题，这意味着 P2P 交易所完全无法满足需快速交易才能及时完成交易的专业交易者。从目前的状况来看，这些交易所只对想要享受其提供的特定优势的人有用 — 例如恢复力强、注重隐私、安全性高以及交易自由。”[11]

买方与卖方之间的平衡
交易所关闭导致资金损失
冻结账户的潜在性

交易所从交易中获取的收益
交易安全风险
要求的存款
去中心化交易所
集中式交易所

2.3 去中心化交易所

有些项目声称是 P2P 去中心化交易所 (DEX)。但很少有像 dApp 一样完全建立于区块链之内的。有的是依赖于组织硬件和专用软件的服务器操作集中式客户端，还有一些不过是需要集成到现有集中式交易所才能有效工作的协议。Aphelion 旨在成为 DEX 的先驱之一，不仅作为 dApp 完全驻留在区块链中，而且只需要一个开源的用户界面来访问数据和控制智能合约，从而交易数字资产。

关注领域：

Ripple

Ripple[12] 是一种提供实时结算系统、货币兑换处和汇款网络的协议。它需要插入到一个现有网络中，而且被设计为在中央银行体系之内工作。Ripple 协议可以通过将区块链技术引进到世界最大的金融机构中来彻底改变银行业。然而，它不提供 P2P 去中心化交易系统。

Shapeshift

Shapeshift[13] 是以功能高度依赖于企业软硬件的操作为基础的服务器。Shapeshift 做出了非常了不起的承诺，即无需在交易平台存入资金也能马上进行点对点的交易。简单搜索一下便知，shapeshift 的集中式服务器基础架构可能会导致用户失去货币或丢失交易，而且在解决棘手问题方面也不会提供帮助。

Loopring

Loopring[14] 是一个目前正处在开发阶段的交易所协议（截至 2017 年 9 月）。loopring 协议需要插入到现有加密货币交易所中，它包括用户授权以及交易所和 loopring 之间的企业集成。如果 loopring 可以克服与现有交易所集成的挑战，则证明它是一个潜力巨大的中介机构。

Bitshares

Bitshares[15]，[16] 是一个工业级的金融区块链智能合约平台。它是真正去中心化技术的优秀代表。有人可能会指出 Bitshares DEX 有一些细微差别，即当资金存入后，您的资产被 Bitshares

作为担保品储存，然后 Bitshares 会向您发放其自身的货币，如您所知，即智能代币。用户必须用仿制实际货币和资产的衍生代币进行交易。例如 Bitshares 版本的美元 bitUSD 或 Bitshares 版本的金币 bitGold。

OpenLedger

OpenLedger[17] Dex 是一种加密货币交易所。和 Bitshares 一样，它可以让用户将真实资产兑换成衍生代币，即存在于 OpenLedger 网络中的智能代币。例如，用户可以通过 Openledger 来交易 Open.BTC 和 Open.ETH，他们分别相当于 OpenLedger 自己的比特币和以太坊。

Bancor

Bancor [18] 协议具有内置的价格发现功能，而且它在智能合约区块链中设有代币的流动性机制。与 Bitshares 和 Openledger 类似，Bancor 通过“智能代币”来获得一个以上储备的真实代币，如此一来任何一方都可以用其储备的任何代币作为交换立即购买或清算智能代币。这一操作是通过智能代币合约直接完成的，它根据一个买卖数量之间的公式来连续计算价格。

0x

0X[19]（零 X）是一种促进 ERC20 代币在以太坊区块链上进行点对点交易的协议。此协议被用于在现有 dApp 之内加速基于代币交易的以太坊。

3.解决方案：

Aphelion 由代币驱动的 DApp 十分具有突破性，它可以通过 DEAL 来实现点对点的资产分配和智能合同，并解决当前的交易所和交易平台所面临的问题。该解决方案是为了消除这些机制的集中化模式，让用户可以自由地设置自己的智能合约，并通过他们的方式直接在区块链上以一个开源、安全、快速且真正去中心化的过程来交换数字资产。Aphelion DApp 和协议代币将解决延迟、资产冻结或被盗等问题，并最终实现加密交易的永久自由。

3.1 P2P 数字资产分配和协议

Aphelion 是新一代的 DApp 和代币协议，它将能够与任何其他 DApp 集成。Aphelion 是真正的开源技术，它不被任何实体、组织或代理机构所拥有或控制。将智能合约技术作为一种拥有自身托管或建设区块代币化系统的协议，Aphelion 用户可以最终消除加密货币交易所和交易平台的阻碍和控制。Aphelion 使用户能够通过他们自己选择的合约方式直接与对方交易。它为用户提供了创新的代币化托管解决方案，使其能够在任何地方与任何人进行 Aphelion 批准的代币交易，以及向任何人转账、发送或从任何人接收 Aphelion 批准的代币。

3.2 使命与愿景说明

使命：建立资产分配真正去中心化的协作式开源区块链技术。愿景：让去中心化的应用来推动世界发展

3.3 Aphelion 技术

NEO 技术：通过 P2P 网络、dBFT 共识、数字证书、超导交易和跨链互操作性等技术，区块链可以以安全有效且具有法律约束力的方式来管理智能资产。数字资产：数字资产是以电子数据的形式存在的可程式化资产。利用区块链技术，资产可以通过去中心化、可靠、可追溯、高度透明且无中介机构的方式实现数字化。在区块链上，用户可以注册、交易和流通多种类型的资产，例如 BTC、ETH、XRP、LTC 和 NEO 等，这里仅列举了部分。

3.4 主要区别

真正的去中心化：Aphelion 是基于节点的 P2P 交易，不受第三方的控制或影响。用户可以通过真正意义上的去中心化方法来设定自己的规则。场所不会遭受破坏，因为根本就没有场所。只有当双方进入 DEAL（分布式交易所资产账本）中时交易才能完成，而且该账本会将其记录到数百万台潜在的机器中。跨语言可扩展性：和其他代币完全不同，Aphelion 开放 Python、.Net、C#、F#、Go 和 Java 等多种语言并且可以通过他们来进行开发，因此它不断具有高度的可扩展性还有利于各种编程人才的加入。下一代 DApp：这是一种 NEO 代币化系统，它利用 DEAL 协议来实现真正完全去中心化的 P2P 交易所准入简单：Aphelion 只需要访问内置于浏览器、应用程序或桌面上的开源 Aphelion 门户即可。安全性：因为数据是真正在分布式区块链账本中被去中心化的，所以它不会被盗或被破坏。控制：Aphelion 用户发起 DEAL 交易并完全掌控其各自的智能合约，所以交易不会产生费用也不受规则约束。

3.5 路线图

2017 年第 1 季度 – 概念与研究
研发区块链选择
确定行业领导者
加密货币市场研究
加密货币交易平台比较分析
2017 年第 2 季度 – 战略与设计
保留法律顾问
创造概念
创造 Aphelion 名称和消息
设计实体模型
SWOT 分析

制定我们的使命
和区块链开发人员一起审查概念
2017 年第 3 季度 — 初始业务部署和前期行销
合并业务部门
确定市场
建立合作关系网络
启动登录网站
合规框架
NEO 是理想之选
形成创始人联盟协议
招募和审查顾问
2017 年第 4 季度 — 营销与 ICO
开发开始
部署营销工作
开发影响者网络
与流动性提供者建立关系
部署 Testnet
管理 GitHub 回购协议
网站改善与后端
KYC 验证实体整合
完成并发布白皮书
开放私人发售
Decstack 渠道
智能合约测试与审核
ICO 交易测试
初始 dApp 开发
2017 年第 4 季度 — 营销和 ICO（继续）

开发开始
部署营销工作
开发影响者网络
与流动性提供者建立关系
初始 NEO 访问
部署 Testnet
管理 GitHub 回购协议
网站改善与后端
KYC 验证实体整合
完成并发布白皮书
开放私人发售
Telegram 频道
智能合约测试与审核
ICO 交易测试
初始 dApp 开发
KYC 审核完成
ICO 开始
ICO 关闭
发行的代币
PR 开始
合规性更新
2018 年第 1 季度 — NEO 年开始
dApp 全面开发开始
跨区块链交易
解决流动性验证问题
营销继续
交易所注册开始

审查
交易所交易的 APH
发行 Aphelion DApp 的初始版本
Aphelion dApp 社区发展与成长
持续的市场分析
NEO 智能经济的领先发展
2018 年，迈向未来 – 2018 年，迈向未来 [tbd]
继续建立开发团队
建立品牌忠诚度并获得狂热的粉丝
将市场扩大到各大洲
利用合作关系来促进创新和一体化
成就非凡

3.6 Aphelion 代币 — 工作原理...

APH 代币是一种新型的数字资产分配。Aphelion 代币就像数字托管或流动性验证设备 (LVD) 一样，它可以同时捕捉来自买方和卖方的条款，并使所提出的智能合约达成一致，然后立即验证流动性并结算 DEAL。Aphelion 的分布式交易所资产账本（或 DEAL）绕过交易所将 P2P 向直接并且真正基于节点的去中心化账本推进。APH 代币化 DEAL 是一种 DApp 直接位于区块链中的协议，因此它可以绕过交易所让 APH 成为流动性验证设备，最终实现即时、安全和完全去中心化的承诺。

让我们来考虑一下这样一种情况，即两个节点想要用不同区块链上的数字资产进行交易。被称为 Alex 的 A 节点想要用其在 A 区块链 (B.A) 上的一些资产来交易 B 区块链 (B.B) 上的最小额度资产。而被称为 Bob 的 B 节点则想要用其在 B.B 上的一些资产来交易 B.A 上的最小额度资产。这两点在两个区块链中都拥有地址，而且两边都可以发起交易合约。这种分布式跨链交易分多个阶段执行，但被视为一项整体的工作。最后，双方交易成功或都回到原始的状态。

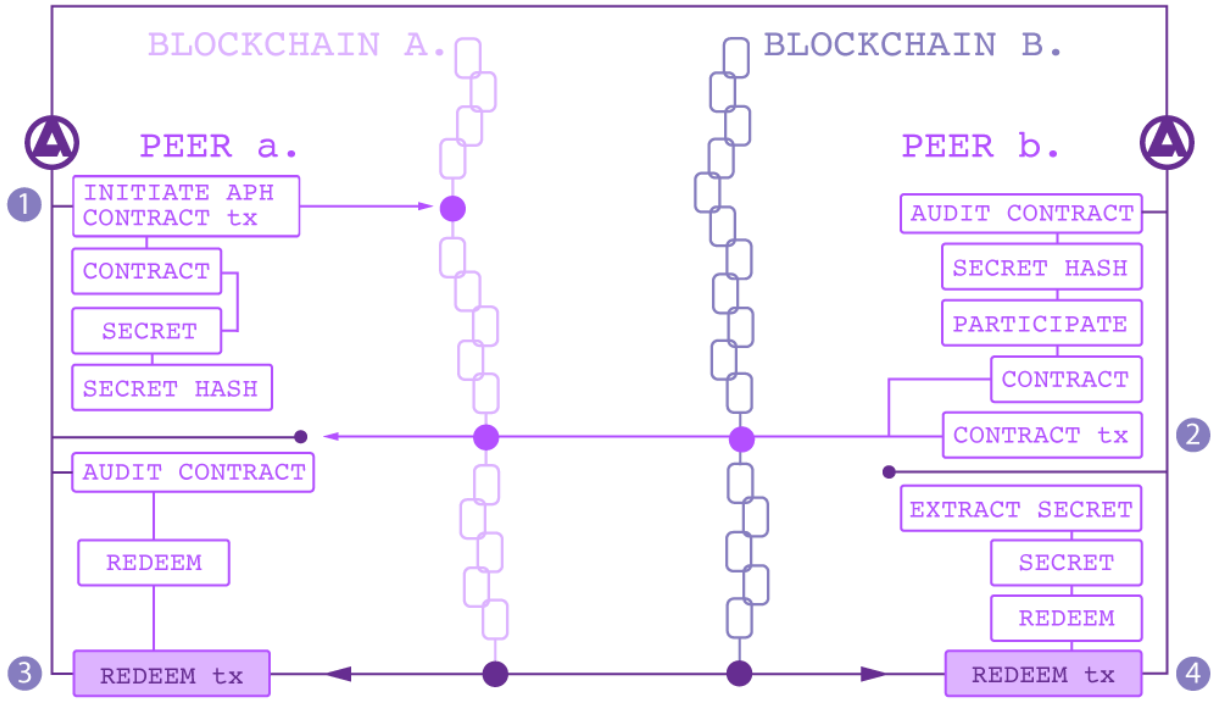
1.假设 Alex 利用 Aphelion 代币启动跨链数字资产交易所。此过程创建了一个包含合约、秘密代码（或简称密码）以及该密码的哈希值的合约交易。同时也将 Alex 的必要资产锁定到 B.A 上并提供其想要在 B.B 上接收资产的地址。

2.接下来，Bob 查看合约（称之为审核），同意条款并决定参加交易。他同意另起一项合约交易，该交易使用来自 Alex 合约交易的密码哈希值。此过程同样将 Bob 需要的资产锁定在 B.B 上，并提供其想要 Alex 将资产发送至 B.A 上的地址。

3.Alex 查看（审核）Bob 发送的内容并决定关闭 DEAL。Alex 通过创建兑换交易来收取 Bob 的付款。此过程自动将 Alex 的密码发布给 Bob，然后又触发一个新的兑换交易 (4)，使 Bob 能够获得 Alex 的付款。两次兑换交易和传统的两阶段提交关系数据库类似，即如果元交易的任何部分失败，则各个交易失败并回到原状。

有一个关键元素没有显示在此简图中，即资产也可以在交易过程中的某些时间点退还或退回到原始钱包中。Alex 的合约交易将包含一段锁定时间，在交易被挖掘但未被兑换时结束。Bob 的合约交易也会包含一个锁定时间，其长度为 Alex 锁定时间的一半。这些锁定时间结束后，特定的一方可以发起退款，于是所有相关资产将被退回。

未来会有怎样的发展？虽然 Aphelion 只是从 NEO 区块链开始的，但其最终愿景是成为连接跨区块链社区的桥梁，而且是去中心化和基于节点之上的。Aphelion 开始是在 NEO 上寻求区块链的固有价值，但其目标是传播自己的协议并延伸至 ETH、BTC 和其他未来区块链中以实现代币的最终使用：完全的区块链不可知性、直接、P2P、跨维度、去中心化的交易所，最终实现完全发挥区块链潜力的承诺。让真正去中心化的 DApp 变成桥梁，DApp 和 Aphelion 代币协议的强大功能和实用性让任何一点都变得无关紧要；这个整体将超越各个部分的总和。



3.7 Aphelion 初始货币发售

Aphelion ICO 正在预售。早期参与者、顾问和所有者都被分配了代币。正式的 ICO 计划于 2017 年 11 月 15 日开始倒计时。可以通过 NEO、BTC 和 ETH 直接在 Aphelion.org 上存入资金，ICO 的目标是三千四百万美元或同等价值的 NE0/BTC/ETH。ICO 将发售 2.2 亿 APH 代币，并通过购买和推荐计划发行奖励代币。

分配细目
ICO 出售 40%
奖励计划 10%
ICO 预售参与者 5%
顾问 15%
组织 30%

3.8 定价结构与时间表

Aphelion ICO 代币价格为 0.2 美元。
NEO 汇率将于 2017 年 11 月 13 日通过一个三天的移动平均值决定。
这个移动平均值是通过 SMA 方法利用来自 coinmarketcap.com 的历史数据决定的。
第一阶段从 2017 年 11 月 15 日的第一个区块开始
ICO 在 2017 年 12 月 7 日的最后一个区块结束
每一轮都会分配共两亿 ICO 代币。所有 ICO 代币很可能在第一轮就销售一空。这样的话，Aphelion 将获得两千八百六十六万美元左右的市值。
在第三轮结束后仍未售出的所有代币将被销毁。
例如，汇率为 30 美元的 NEO：
第 1 轮：1 NEO = 150 APH + 75 APH [共 225 APH]
第 2 轮：1 NEO = 150 APH + 38 APH [共 188 APH]
无奖励：1 NEO = 150 APH
轮次

开始日期
结束日期
持续时间
奖励
有效价格
11/15/2017 第一区块
11/16/2017 第一区块
11/23/2017 第一区块
11/15/2017 最后区块
11/22/2017 最后区块
11/7/2017 最后区块
24 小时
7 天
14 天
无奖励

所得款项用途

65% 用于区块链和 DApp 开发
10% 用于营销
15% 用于运营
10% 用于研发

3.9 Aphelion 智能合约暂停

为维持项目和保护 ICO 参与者，将强制所有创始人和顾问暂停出售 Aphelion 代币 6 个月。此策略将完全公开透明地加入到区块链智能合约中。

4.Aphelion 团队

Aphelion 团队具有强大的全球人才网络，由在区块链技术、金融、经济、营销、安全、软件工程和开发等方面获得非凡业绩的企业家、专家和远见者组成。

团队

Adi Benari 技术顾问 应用区块链
Andrew Morrell 首席软件工程师 嘉信理财开发人员
Colan Sewell 首席分析师 美国 HTC Vive 领导者
Aaron Levin 网络和应用安全 安全顾问
Joel Garcia 区块链开发人员 所有代码
Matt Brozovich Web 开发人员 BrozKnows 创始人
Joshua Finkleman ICO 顾问 区块链资本公司
Astrid Baldissera 法律顾问 Starting Legal 的 CEO
Natalie Wilcox 社交媒体营销 世纪互联
Eric Liss 动态图像设计/UI 自由职业全球领导

Joe Debuzna 软件架构师 工程副总裁 HVR 软件
Michael Jaltuch 创始成员 Orion Technologies 创始人， 线性方法软件
Ian Holtz 创始成员 Orion Technologies 创始人

5.总结

Aphelion 正在建立基于区块链的新一代代币化机制，以解决集中式加密货币交易所和交易平台所面临的挑战。此协议将生成一个被称为分布式交易所资产账本 (DEAL) 的真正点对点智能合约。Aphelion DEAL 是一种建立于 NEO 开源区块链之上的新型 DApp,它支持多种编程语言而且交易快速，此外，它还能让 DEAL 制作者免受规则、延迟和安全漏洞的困扰。加入我们，完成建立协作式开源 P2P 区块链技术的使命，最终实现资产分配的去中心化并将区块链带向未来。

6.参考资料

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system (2008) https://bitcoin.org/bitcoin.pdf
[2] Don & Alex Tapscott, authors Blockchain Revolution (2016)
[3] Rob Marvin, Blockchain: The Invisible Technology That's Changing the World (2017)
[4] JP Buntinx, Distributed Ledger Technology Vs Blockchain Technology (March 25, 2017) https://themerkle.com/distributed-ledger-technology-vs-blockchain-technology/
[5] Blockchain project Antshares explains reasons for choosing dBFT over PoW and PoS (July 17, 2017) https://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275
[6] Daan Pepijn, Here's how NEO plans to top Ethereum and Bitcoin (August 11, 2017) https://thenextweb.com/contributors/2017/08/17/heres-neo-plans-top-ethereum-bitcoin/
[7] Christopher Malmo, Ethereum Is Already Using a Small Country's Worth of Electricity (June 26, 2017)https://motherboard.vice.com/en_us/article/d3zn9a/ethereum-mining-transaction-electricity-consumption-bitcoin
[8] Haseeb Qureshi, A hacker stole $31M of Ether (July 20, 2017) https://medium.freecodecamp.org/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce
[9] Dr Garrick Hileman & Michel Rauchs, Global Cryptocurrency Benchmarking Study, The Cambridge Centre for Alternative Finance (2017)

https://www.jbs.cam.ac.uk/fileadmin/user_upload/research/centres/alternative-finance/downloads/2017-global-cryptocurrency-benchmarking-study.pdf
[10] Jamie Redman, Another All Time High – Bitcoin Breaks Through 5,000 USD on Asian Exchanges (September 2, 2017) https://news.bitcoin.com/bitcoin-hits-5000-usd-new-all-time-high/
[11] Andrew Marshall, P2P Cryptocurrency Exchanges, Explained (APR 07, 2017) https://cointelegraph.com/explained/p2p-cryptocurrency-exchanges-explained
[12] Peter Todd, The ripple protocol consensus algorithm Review. Ripple Labs Inc White Paper (May, 2015) https://raw.githubusercontent.com/petertodd/ripple-consensus-analysis-paper/master/paper.pdf
[13] Shapeshift Reviews http://bittrust.org/shapeshift
[14] Loopring Project Ltd., LOOPRING Decentralized Token Exchange Protocol (Sept 26, 2017) https://github.com/Loopring/whitepaper/raw/master/en_whitepaper.pdf
[15] Fabian Schuh and Daniel Larimer. Bitshares 2.0: Financial smart contract platform (Nov 12, 2015) http://docs.bitshares.org/_downloads/bitshares-financial-platform.pdf
[16] Fabian Schuh and Daniel Larimer. Bitshares 2.0: General overview (2015) http://docs.bitshares.org/_downloads/bitshares-general.pdf
[17] Open ledger (2017) https://openledger.io/
[18] Eyal Hertzog, Guy Benartzi & Galia Benartzi, Bancor Protocol Continuous Liquidity and Asynchronous Price Discovery for Tokens through their Smart Contracts; aka “Smart Tokens” (March 30, 2017) https://www.bancor.network/static/bancor_protocol_whitepaper_en.pdf
[19] Amir Bandeali. Introducing 0x -An Open Protocol For Decentralized Exchange On The Ethereum Blockchain (February 22, 2017) https://blog.0xproject.com/introducing-0x-d51d5231ba53
[20] Binance GAS(was Antcoin), (July 2017) https://binance.zendesk.com/hc/en-us/articles/115000967291-GAS-was-Antcoin-

7.附件

- {COIN_A} holder
- {COIN_B} holder

Process

- The 'superconducting transaction' (also 'on-chain atomic swap') proceeds through two transactions, one on the {COIN_A} blockchain, the other on the {COIN_B} blockchain.
  - [1]:{COIN_A} holder has an unspent amount, A, of {COIN_A} in an address recorded in a transaction on the {COIN_A} blockchain.
    {COIN_A} holder will pay this unspent amount into a {COIN_A} address controlled by {COIN_B} holder through a transaction on the {COIN_A} blockchain.
  - [2]:{COIN_B} holder has an unspent amount, B, of {COIN_B} in an address recorded in a transaction on the {COIN_B} blockchain.
    {COIN_B} holder will pay this unspent amount into a {COIN_B} address controlled by {COIN_A} holder through a transaction on the {COIN_B} blockchain.

Steps

  - {COIN_A} holder 'initiates'.
    - Obtains following information from {COIN_B} holder:
      - {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made.

▪ Creates and publishes contract transaction on {COIN_A} blockchain, with a locktime set by the seller sometime in the future (user set expiry date/time).
▪ This step returns the secret, the secret hash, the contract script, the contract transaction, and a refund transaction that can be sent after (user set expiry date/time) if necessary.

◦ {COIN_B} holder 'audits contract'.
▪ Obtains following information from {COIN_A} holder:
• Swap-script, the output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts.
• Trans, superconducting transaction for {COIN_A} blockchain.
▪ Inspects {COIN_A} blockchain superconducting transaction contract script to review addresses that may claim the output, the locktime, and the secret hash. Also validates that contract transaction pays to the contract and reports the contract output amount.

◦ {COIN_B} holder 'participates'.
▪ Obtains following information from {COIN_A} holder:
• {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made.
• Secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction.
▪ Creates and publishes contract transaction on {COIN_B} blockchain, incorporating also the secret hash from the {COIN_A} blockchain contract transaction 'initiated', above, and with a locktime of (user set expiry date/time).
▪ This step returns the the contract script, the contract transaction, and a refund transaction that can be sent after (user set expiry date/time) if necessary.

◦ {COIN_A} holder 'audits contract'.
▪ Obtains following information from {COIN_B} holder:
• Swap-script, the output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts.
• Trans, superconducting transaction for {COIN_B} blockchain.
▪ Inspects {COIN_B} blockchain superconducting transaction contract script to review addresses that may claim the output, the locktime, and the secret hash. Also validates that contract transaction pays to the contract and reports the contract output amount.

◦ {COIN_A} holder 'redeems'.
▪ Will already have obtained (see prior step) the following information from {COIN_B} holder:
• Swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts.
• Trans, the superconducting transaction for the {COIN_B} blockchain.
▪ Redeems {COIN_B} coins paid into the contract in {COIN_B} blockchain by {COIN_B} holder. Redeeming requires the secret, known only to the {COIN_A} holder up to this point.

◦ {COIN_B} holder 'extracts secret'.
▪ Extracts secret from {COIN_A} holder's redemption transaction. With the secret known, the {COIN_B} holder may claim the {COIN_A} coins paid into the contract in the {COIN_A} blockchain by {COIN_A} holder.

◦ {COIN_B} holder 'redeems'.
▪ Will already have obtained (see 'audit contract' step) the following information from {COIN_A} holder:
• Swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts.
• Trans, the superconducting transaction for the {COIN_A} blockchain.
▪ Redeems {COIN_A} coins paid into the contract in {COIN_A} blockchain by {COIN_A} holder.

Refunds
◦ If a period of time equal to the time-lock (i.e. (user set expiry date/time), in the case of the {COIN_A} blockchain superconducting transaction, and (user set expiry date/time), in the {COIN_B} case) expires after the transaction has been mined but has not been redeemed, the contract output can be redeemed back to the holder's wallet.

Pseudo-code

'initiate', by {COIN_A} holder
{COIN_A} holder runs:
$ 'initiate', with parameters
- [{COIN_A} blockchain, i.e. the blockchain on which {COIN_A} holder's payment will be made]
- [string representing {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made]
- [string representing A, amount of {COIN_A} to be paid to this address]
{
Decode parameter [string representing {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made]. If it conforms with a valid address for the {COIN_A} blockchain, return this address, their-address.
Decode [string representing A, amount of {COIN_A} to be paid to this address]. If it conforms to a valid double-precision floating-point number (i.e. binary64), and is not NaN or +/- infinity, return this number, amount.
Open JSON-RPC connection with the {COIN_A} blockchain.
Generate [secret], a new secret key for the {COIN_A} blockchain.
Calculate [secret-hash], the hash of [secret].
Calculate [lock-time], a locktime (user set expiry date/time) from current time.
Calculate [refund-address], a {COIN_A} address for the refund transaction.

Build the superconducting contract on the {COIN_A} blockchain, with parameters:
- [their-address]
- [lock-time]
- [secret-hash]
- [refund-address]

Return [swap-script], the output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts:
- [{COIN_B} holder's sig] [{COIN_B} holder's pub key] [{COIN_A} holder's secret], or
- [{COIN_A} holder's sig] [{COIN_A} holder's pub key]
Calculate [swap-address-script-hash], a new address script hash of [swap-script].
Calculate [tx-script], a new script to pay the transaction output to [swap-address-script-hash].
Calculate [fee], the fees associated with the transaction.
Calculate [trans], superconducting transaction for {COIN_A} blockchain, with parameters:
- [A, amount]
- [tx-script]
- [refund-address]
- [fee]
- [lock-time]
Sign [trans].
Calculate:
- [refund trans], the refund transaction
- [refund fee], the fee associated with the refund transaction.

Return and Display:
- [secret]
- [secret-hash]
- [swap-script]
- [trans]
- [refund-trans]
- [lock-time]
Publish transaction.
}

'audit contract', by {COIN_B} holder
{COIN_B} holder runs:
$ 'auditcontract', with parameters
- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]
- [string representing swap-script, output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts]
- [string
representing trans, superconducting transaction for {COIN_A} blockchain]
{
Decode parameter [string representing swap-script, output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Decode parameter [string representing trans, superconducting transaction for {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.

Open JSON-RPC connection with the {COIN_A} blockchain.

Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts
Return
- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction
- [lock-time]
Calculate pay to address, with parameters:
- [trans], the superconducting transaction for the {COIN_A} blockchain

Return
- [PubKeyTx], address on {COIN_A} blockchain into which {COIN_A} holder will make payment

Display
- [swap-script-hash], address on {COIN_A} blockchain of superconducting contract
- [amount], value of {COIN_A} to be paid into {COIN_B} holder's address on {COIN_A} blockchain
- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} will be paid
- [refund-address], {COIN_A} holder's address on {COIN_A} blockchain for payment of refund of {COIN_A}
- [lock-time]
}

'participate', by {COIN_B} holder
{COIN_B} holder runs:
$ 'participate', with parameters
- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]
- [string representing {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made]
- [string representing B, amount of {COIN_B} to be paid to this address]
- [string representing secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction]
{
Decode parameter [string representing {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made] . If it conforms with a valid address for the {COIN_B} blockchain, return this address, their-address.
Decode [string representing B, amount of {COIN_B} to be paid to this address]. If it conforms to a valid double-precision floating-point number (i.e. binary64), and is not NaN or +/- infinity, return this number, amount.

Decode [string representing secret-hash, the hash of the new secret key for the {COIN_A} blockchain contract transaction]. If it conforms to a valid hexadecimal string of the right length, return the bytes, their-secret-hash.
Open JSON-RPC connection with the {COIN_B} blockchain.
Calculate [lock-time], a locktime (user set expiry date/time) from current time.
Calculate [refund-address], a {COIN_B} address for the refund transaction.
Build the superconducting contract on the {COIN_B} blockchain, with parameters:
- [their-address]
- [lock-time]
- [their-secret-hash]
- [refund-address]
Return [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts:
- [{COIN_A} holder's sig] [{COIN_A} holder's pub key] [{COIN_A} holder's secret], or
- [{COIN_B} holder's sig] [{COIN_B} holder's pub key]
Calculate [swap-address-script-hash], a new address script hash of [swap-script].
Calculate [tx-script], a new script to pay the transaction output to [swap-address-script-hash].
Calculate [fee], the fees associated with the transaction.
Calculate [trans], superconducting transaction for {COIN_B} blockchain, with parameters:
- [B, amount]
- [tx-script]
- [refund-address]
- [fee]
- [lock-time]
Sign [trans].

Calculate: - [refund trans], the refund transaction
- [refund fee], the fee associated with the refund transaction.

Return and Display:
- [secret]
- [secret-hash]
- [swap-script]
- [trans]
- [refund-trans]
- [lock-time]

Publish transaction.
}

'audit contract', by {COIN_A} holder
{COIN_A} holder runs:
$ 'auditcontract', with parameters
- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]

- [string representing swap-script, output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts]
- [string representing trans, superconducting transaction for {COIN_B} blockchain]
{
Decode parameter [string representing swap-script, output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Decode parameter [string representing trans, superconducting transaction for {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Open JSON-RPC connection with the {COIN_B} blockchain.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts

Return
- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_B} blockchain contract transaction
- [lock-time]

Calculate pay to address, with parameters:
- [trans], the superconducting transaction for the {COIN_B} blockchain
Return
- [PubKeyTy], address on {COIN_B} blockchain into which {COIN_B} holder will make payment

Display
- [swap-script], address on {COIN_B} blockchain of superconducting contract
- [amount], value of {COIN_B} to be paid into {COIN_A} holder's address on {COIN_B} blockchain
- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} will be paid
- [refund-address], {COIN_B} holder's address on {COIN_B} blockchain for payment of refund of {COIN_B}
- [lock-time]
}

'redeem', by {COIN_A} holder
{COIN_A} holder runs:
$ 'redeem', with parameters
- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]
- [string representing swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts:
- [{COIN_A} holder's sig] [{COIN_A} holder's pub key] [{COIN_A} holder's secret], or

- [{COIN_B} holder's sig] [{COIN_B} holder's pub key]]
- [string representing trans, the superconducting transaction for the {COIN_B} blockchain]
- [string representing secret, the secret key for the {COIN_A} blockchain]
{
Decode parameter [string representing swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Decode [string representing trans, the superconducting transaction for the {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, trans.
Decode [string representing secret, the secret key for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, secret.
Open JSON-RPC connection with the {COIN_B} blockchain.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts

Return
- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction
Calculate pay to address, with parameters:
- [trans], the superconducting transaction for the {COIN_B} blockchain

Return
- [PubKeyTy], address on {COIN_B} blockchain into which {COIN_B} holder will make payment
Verify [address] and [PubKeyTy] are equal.
Calculate [pay-script], script to pay a transaction output to [PubKeyTy].
Create [redeemTx], redeem transaction.
Sign [redeemTx].
Publish [redeemTx]
}

'extract secret', by {COIN_B} holder
{COIN_B} holder runs:
$ 'extractsecret', with parameters:
- [string representing redeemTx, the redeem transaction published by {COIN_A} holder on the {COIN_B} blockchain]
- [string representing secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction]
{
Decode [string representing redeemTx, the redeem transaction published by {COIN_A} holder on the {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, redeemTx.

Decode [string representing secret-hash, the hash of the new secret key for the {COIN_A} blockchain contract transaction]. If it conforms to a valid hexadecimal string of the right length, return the bytes, their-secret-hash.
Open JSON-RPC connection with the {COIN_B} blockchain.
Loop over all pushed data, searching for one that hashes to the expected hash.
Return [secret].
Display [secret].
}

'redeem', by {COIN_B} holder
{COIN_B} holder runs:
$ 'redeem', with parameters
- [{COIN_A} blockchain, i.e. the blockchain on which {COIN_A} holder's payment will be made]
- [string representing swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts:
- [{COIN_B} holder's sig] [{COIN_B} holder's pub key] [{COIN_A} holder's secret], or
- [{COIN_A} holder's sig] [{COIN_A} holder's pub key]]
- [string representing trans, the superconducting transaction for the {COIN_Aa} blockchain]
- [string representing secret, the secret key for the {COIN_A} blockchain]
{
Decode parameter [string representing swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Decode [string representing trans, the superconducting transaction for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, trans.
Decode [string representing secret, the secret key for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, secret.
Open JSON-RPC connection with the {COIN_A} blockchain.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts

Return
- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction
Calculate pay to address, with parameters:
- [trans], the superconducting transaction for the {COIN_A} blockchain

Return
- [PubKeyTy], address on {COIN_A} blockchain into which {COIN_A} holder will make payment
Verify [address] and [PubKeyTy] are equal.

```
Calculate [pay-script], script to pay a transaction output to [PubKeyTy].
Create [redeemTx], redeem transaction.
Sign [redeemTx].
Publish [redeemTx]
}

'refund', by either holder
Either holder runs:
$ 'refund', with parameters
- [B, blockchain, i.e. the blockchain on which refund will be made]
- [string representing swap-script, for the superconducting transaction to be
refunded]
- [string representing trans, the superconducting transaction to be refunded]
{
Decode [string representing swap-script, for the superconducting transaction to
be refunded]. If it conforms to a valid hexadecimal string of the right length,
return the bytes, redeemTx.
Decode [string representing swap-script, for the superconducting transaction to
be refunded]. If it conforms to a valid hexadecimal string of the right length,
return the bytes, their-secret-hash.
Open JSON-RPC connection with the blockchain, B.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_A}
blockchain by either of two signature scripts
Return
- [amount], value to be refunded on blockchain, B
- [fees], fees associated with the transaction
- [refund-address], the address on blockchain, B, into which refund will be
made
Calculate [pay-script], script to pay a transaction output to [refund-address].
Create [refundTx], refund transaction.
Sign [refundx].
Publish [refundTx]
}
```