下一代去中心化应用平台
Next-Generation Decentralized Application Platform

Asch.io阿希

一个基于侧链技术的去中心化应用平台

V1.0.2

目录

0 引言

比特币的出现使得去中心化的货币系统成为可能，经过几年的发展，人们发现比特币背后的区块链技术潜力巨大，可以被广泛应用在各行各业。为了更好地利用区块链技术， 出现了以以太坊为代表的一批应用平台，它们封装了底层协议、建设了基础设施，为开发者提供了更加友好、也更加灵活的接口，使得开发者的关注点能够集中在业务逻辑上，很大程度上提高了开发效率。本文提出的 ASCH 系统也是一种去中心化应用的开发平台，接下来我们会详细阐述这一系统的特色、原理及应用场景。

1 概述

1.1 去中心化应用

去中心化应用是一种具有以下特点的应用。

1. 必须完全开源，自主运行，不能被中心化的组织、机构或个人操控，可以被改进以响应市场需求，但必须经过用户们的共识。
2. 数据必须被安全、公开、冗余的存储在一个分布式网络中，以避免被篡改和单点故障。
3. 应用访问者需要消耗令牌，而应用贡献者可以获得令牌的奖励。
4. 应用必须使用一种价值证明的密码学算法来生成令牌。

去中心化应用可以通过授权系统的权益人来投资应用的开发，从而有潜力达到自给自足。去中心化应用还有公开透明、安全可靠、去信任等优点。因此，可以想像去中心化应用在支付、数据存储、云计算、电子商务等领域将有非常可观的前景，它所产生的价值甚至有可能会超过 Visa、Dropbox、亚马逊等跨国公司的市值。

1.2 侧链是什么

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若干网络交易的信息，用于验证其信息的有效性（防伪）和生成下一个区块，对于普通用户来说它就像一个公有账本，记载所有的交易记录，对于开发者来说可以理解为一个分布式的数据库。区块链这个数据库的特点是去中心化、开放、自治、不可篡改，区块链与去中心化应用息息相关，非常适合为去中心化应用

提供存储功能。

侧链是一种特殊的区块链。它使用一种叫做“SPV 楔入”的技术实现与其他区块链之间的资产转移，这使得用户能用已有的资产来使用新的加密货币系统。人们不必再担心比特币难于采纳创新和适应新需求，只要创造一个侧链，然后对接到比特币的区块链中即可，通过继承和复用比特币强大的区块链，还避免了新货币的流动性短缺和市场波动等问题。并且由于侧链是一个独立的、隔离的系统，侧链中出现的严重问题只会影响侧链本身，这极大地降低了创新的风险和成本。

1.3 ASCH 是什么

ASCH 是一个去中心化的应用平台。它提供了一系列的 SDK 和 API 来帮助开发者构建基于 Javascript 和侧链技术的去中心化应用。ASCH 通过提供定制侧链、智能合约、应用托管等一体化的行业解决方案，致力于打造一个易于使用、功能完备、即插即用的系统。利用 ASCH 生态系统，开发者可以快速迭代他们的 Javascript 应用，并发布到系统内置的应用商店中，这些应用可以被平台中的分布式节点下载并执行，并服务于普通用户，整个过程都由诚实安全的 ASCH 侧链共识网络提供安全保证。

ASCH 系统本身也是一个完全开放的、去中心化的应用，内置有代币，单位为 XAS， 中文名叫阿希币。阿希币可以通过双向楔入的方式与侧链或 Dapp 进行交互，作为所有Dapp 之间资产转换的桥梁和媒介，这些代币将在系统发布之前以 ico 的方式预售给投资人。系统一旦发布，ASCH 最初的核心团队将不再掌控系统的走向，只有系统的权益人和代币的拥有者决定系统将来的发展。

1.4 ASCH 面向哪些用户

ASCH 平台除提供一些基本服务外，还将提供技术和工具上的支持，主要面向以下群体：

1.4.1 开发者

开发者可以根据 ASCH 平台的应用开发规则和商业行为准则，并按照相关的规范进行开发和提交 Dapp。Dapp 的商业模式或免费，或定价销售，或按增值服务付费。采用何种商业模式完全由开发者决定。

1.4.2 企业

ASCH 平台提供的工具可以非常容易地创建一个完整的区块链，更重要的是可以楔入到ASCH

平台的主链或者比特币的区块链中，实现与成熟电子货币的对接，这对中小型企业， 特别是初创企业是非常有吸引力的。

中小企业可以通过区块链技术提供原本封闭在企业内部、互联网内部的信息和数据， 甚至与监管机构的相关系统数据相互链接，增强透明度，以此树立良好的形象，赢得投资者、金融机构的信任度，顺利拿到融资或项目合同等。

中小企业主动公开和开放资料，已成无法阻挡的趋势。因为现在有很多的公开渠道来获取数据，中小企业已经越来越难隐瞒它们不想让外界知道的信息。我们可以大胆预测， 在未来区块链将是帮助中小企业发展的重要武器。

1.4.3 普通用户

普通用户可以通过 ASCH 内置的应用商店进行下载、安装和使用去中心化应用，这跟手机平台的应用商店是类似的模式。ASCH 系统支持多种类型的去中心化应用，普通用户在消费这些应用的同时，还可以通过贡献内容来获得收益。开发者与普通用户将共同组成一个繁荣的生态系统。

2 设计理念

2.1 完备脚本 vs 侧链

比特币为人称道的一个设计上的亮点就是它的脚本引擎。基于这套脚本引擎，不但可以实现普通的转账功能，还可以实现多方签名、抵押担保、博彩等智能合约应用。但是出于安全和实现难度的考虑，比特币的脚本系统设计的较为简陋，做了非常多的限制，比如它不支持循环、脚本长度受限、只支持几种标准的交易类型。

以太坊的最大特色就是极大地扩展了这个脚本引擎的功能，加入了读取区块链、计费、跳转等新指令，还解除了栈内存、函数调用深度以及脚本长度限制等。以太坊自称他们的脚本语言达到了图灵完备，利用这样的脚本，开发者可以实现几乎任何可以用数学方式表述的功能。

自以太坊以来，扩展脚本成为了一种实现去中心化开发平台的流行方式，但这种方式有一个很大的缺点就是，应用代码本身及应用产生的数据都存在同一个区块链中，造成了区块链的快速膨胀。以太坊试图通过优化和压缩区块和交易本身来延缓这种膨胀，也只是一种治标不治本的方法。此外，基于脚本实现的应用之间是共享同一个账本的，像区块产生时间等参数是无法被定制的，这无疑限制了应用的个性化。

侧链机制是通过另一个维度实现扩展性的，每个侧链运行在不同的分布式节点网络中，有独立的受众、投资人和开发团队。这种天然的分片解决方案，不但解决了区块链的膨胀问题，而且每个应用都拥有一套个性化的账本，其共识机制、区块参数、交易类型都是可以被定制的，所以我们认为侧链与完备交易脚本相比，是一种成本更低、更加灵活、也更加易用的解决方案。

2.2 账户 vs UTXO

在比特币及其衍生系统中，是没有一个所谓的账户来存储用户的余额的，用户的余额是通过整个系统的交易状态转换来实现的。这里要引入一个术语，UTXO（unspent transaction outputs），即未花费的交易输出。每个 UTXO 都有一个面值和所有者，一笔交易包括一个或多个输入和一个或多个输出。每个输入包含一个对现有 UTXO 的引用和由与所有者地址相对应的私钥创建的密码学签名，如果一个用户拥有这个私钥，那么他就可以消费这个 UTXO 对应的币值，也就是说一个用户的余额就是他所有拥有的所有私钥对应的 UTXO 的币值总和。UTXO 主要优点是高度的私密性，用户可以为每一笔交易生成一个新的地址，从而使得用户无法被追踪，这对于货币来说是好事，但对于各式各样的Dapp 来说，就未必了。账户相对于 UTXO 来说，有以下几个优点：

1. 节省空间。举例来说，如果某个用户有 5 个 UTXO，需要的存储空间是（20 + 32 + 8）* 5 = 300 字节（其中 20 字节为地址，32 字节为交易号，8 字节为交易额），而账户仅需要 20+8+2=30 字节（20 字节位地址，8 字节位余额，2 字节为随机数）。
2. 利于监督。账户的存在使得电子货币很容易被区分，因为我们只要知道这些币来自哪些账户即可。
3. 简单、易于编码和理解。
4. 常量级引用。轻客户端能以常数时间访问一个用户的账户任意数据，而在 UTXO 系统中，每当有交易发生时，数据引用将发生变化。

ASCH 平台本身并不是一个纯粹的货币系统，要容纳各种各样的应用，综合比较起来， 账户对于我们来说是一种更好的选择。

2.3 关系数据库 vs 非关系数据库

目前大多数的区块链系统都选择使用模型较简单的非关系数据库来存储数据，比如berkeley db，leveldb 等，这些数据库一般都提供一些简单的数据结构，比如 btree、hashtable、queue 等，它们一般不支持 SQL 对数据进行操作，虽然这些数据库对于一般的电子货币系统来说足够了，但对于应用平台来说是远远不够的，特别是对于金融、银行、电子商务等领域，目前主流的存储系统都是采用了关系数据库，因为关系数据有以下几个优点：

1. 事务处理；
2. 数据更新开销非常小；

3.可以进行 join 等复杂查询。

我们选择的 sqlite 是一种性能极佳的轻量级嵌入式关系数据库，容量最高支持 2T，数据文件可在不同字节序机器之间自由共享，特别是对 SQL 的支持，将为 Dapp 开发者提供极大的便利。

3 系统特点

3.1 易用

3.1.1 开发语言

开发者可以使用 Javascript 语言以及海量的 npm 库来构建他们的应用。相对于比特币的 C++语言及栈式脚本、以太坊的新语言 Solidity，Javascript 流行度更高、受众更广、上手更容易的一门语言。此外，关系数据库的加入也是 ASCH 系统的一大特色，使得去中心化的应用开发模式与传统 Web 应用的开发模式已经非常相近了。ASCH 平台的应用开发门槛可以说是同类产品中最低的。

3.1.2 工具

ASCH 系统提供了一个命令行工具，只需要根据提示输入一些配置项，就可以快速的建立一个侧链，并可在侧链上开发任意类型的应用。其次，系统还提供了一系列的 API 帮助用户构建复杂的智能合约应用，这些 API 涵盖共识、强随机数、数据库、密码学等方面。

3.1.3 部署

开发者只需要把自己的 Dapp 提交到 GitHub，然后在 Web 钱包或轻钱包中注册，就完成了

部署的工作，之后 Dapp 将被显示在应用商店中被用户下载和使用。

3.2 灵活

开发者可以随意定制其侧链的各项参数，比如区块产生速度、交易类型、交易费等等，甚至可以实现一个新的共识机制，比如开发者可以使用权益证明或工作量证明的共识机制来取代默认的委托人权益证明机制。

3.3 安全

ASCH 系统的一大亮点是使用了一个增强 DPOS 的共识算法，在 DPOS 的基础上加入打了一个高效的实用拜占庭容错算法，极大地降低了网络分叉的可能性，只要不超过 1/3 节点联合做恶，系统就不会分叉，也就没有双重支付的风险。其次，系统在一些小的细节方面也是尽量从安全角度进行了考量。比如采用 BIP39 标准算法的口令助记符、二级交易密码、多重签名账户等。

4 技术细节

4.1 共识机制

ASCH 系统采用的共识机制是基于 DPOS 的，也是使用了委托人选举的制度，但是在算法的后半部分采用了一个优化后的 PBFT 算法变种，这个算法可以在 t < n / 3 时，以O(n^2) 消息复杂度，O(1) 的时间复杂度使忠诚的节点达成一致，不会分叉，其中 t 表示拜占庭节点（即可能发生任意行为的节点，比如网络延迟、停机、恶意攻击等等）的个数，n 表示所有节点的个数。

4.1.1 委托人选举

ASCH 系统的委托人选举制度与 DPOS 是类似的，核心系统是由 101 个委托人节点组成，委托人是被社区选举的可信账户，得票最高的 101 个委托人负责生产区块。得票排名未进入前 101 名的账户被称为候选人，当他们将来获得足够多的选票并进入前 101 名后， 将成为正式的委托人。

每个 ASCH 用户都有权利投票给最多 101 位委托人，选票的权重是由用户持有的 XAS 数量决定。

每一个选举周期产生 101 个区块，每一次投票和委托人排名的变化将体现在下一个周期。每个区

块产生的间隔时间是 10 秒，新创建的区块会被广播到网络中并添加到区块链中。每当新的区块被添加到区块链中，该区块之前的所有交易的确认次数加一，得到 6 个确认后，可以认为交易是安全的，如果数额较小的交易，可以允许更小的确认次数，相反，数额较大的交易可以通过增加确认数来保证安全性。

如果有少数委托人发生故障，比如被攻击或者宕机，就会错失区块，这会被记录在案，这将影响该节点的在线率，进而影响社区的投票。因此委托人的竞选是需要严肃对待的，委托人应当由有一定网站运营经验的人来做，委托人要保障自己节点的稳定性，并以此促进整个系统的安全和稳定。

4.1.2 拜占庭容错

ASCH 系统与 DPOS 的不同主要体现在算法的后半部分。

DPOS 采用的方法是，首先对当前 round 的委托人列表进行随机的排序（保证每一轮的委托人顺序不同，也无法预测下一轮委托人顺序），然后通过 round-robin 的方式依次让每个委托人创建区块。这个算法的主要缺点是，如果某个委托人节点叛变了，他可能会广播多个不一致的区块，这些区块间可能包括双重支付交易，导致整个网络被分叉了。当然，如果只有一个委托人叛变的话，这个分叉很快就可以通过下一次最长链同步的方法来消除，但是随着叛变节点的增加，消除分叉的时间将越来越长，少量节点的联合叛变将严重影响系统的安全性，即使一个交易达到 6 次确认，也很可能是不安全的。

为了解决这个问题，我们引入了 PBFT（Pratical Byzantine Fault Tolerance）算法。PBFT 算法也是使用 round-robin 的方式选择委托人，但是选出委托人后并不立即创建区块，而是首先发起一个提议（propose），这个提议的目的是确定下一次区块的 hash。当超过 2/3 的节点都赞成该提议时，才接受由提议人创建的下一个区块，下一个区块的hash 必须与当前 round 达成共识的区块 hash 一致。从本质上来说，PBFT 算法的加入解决了委托人权利滥用的问题，使得委托人的记账能力更为可控。

4.2 侧链与 Dapp

ASCH 系统提供了一个命令行工具，可以用来轻松创建一个基础的侧链系统，侧链的开发者也可以深度定制自己的侧链，侧链拥有自己的数据库、共识机制、交易类型以及账户体系。侧链可以

托管在独立的委托人节点集群中，这就自然形成了一种分片的机制，延缓了主区块链的膨胀。

每一个 Dapp 对应一个侧链，侧链的核心逻辑使用 Node.js 开发，界面部分可以使用任意前端技术，比如 Qt，Html，Javascript 等等，前端与后端之间一般通过 json rpc 协议通讯。Dapp 的作者或者所有者可以跟踪自己的 Dapp 被使用的情况，加密货币是基于社区的共识，但 Dapp 更像是一家私人拥有的公司。Dapp 内的交易是由主节点处理的，主节点是由 Dapp 所有者运行的，Dapp 所有者必须拥有一个 ASCH 账号，这个账号类似多重签名的账号，它的主要任务是在 Dapp 主节点创建共识并签名新的区块，如多重签名钱包。一旦一个新的 Dapp 区块被创建，并且在主节点内被签名，这个区块需要被计算出SHA256 哈希，然后 Dapp 所有者提交这个哈希值给 ASCH 区块链，然后存储该哈希值为Dapp 区块，一旦 ASCH 区块链收到一条包含 Dapp 哈希值的交易，由受托人对比这条哈希值与上一个哈希值，并将它保存到 ASCH 区块链，在未来，当主节点同步网络，用户将通过 ASCH 区块链来验证所有 Dapp 区块，想从 ASCH 区块链中移走上一个 Dapp 区块将是不可能的事情。相同的功能，以比特币区块链来替代 ASCH 区块链将同样适用，API 在比特币区块链上工作的方式是一样的，通过比特币区块链来保证 Dapp 的安全性。开发者可以使用 XAS 和 BTC 来作为其 Dapp 的货币，使用 Dapp 时可能需要存入或者取出资地方金，当 ASCH 或者 BTC 被发送到 Dapp 的地址时，资金会在其 Dapp 的账户内出现，用户便可在 Dapp 内使用该资金，BTC 和 ASCH 的存入方式是一样的，都是发送到 Dapp 的特定地址，然后资金就会出现在 Dapp 账户内。Dapp 的账户都是由 Dapp 的作者创建的， 所有存入的 ASCH 或者 BTC 都将被存储在这个地址内，考虑到安全性，只推荐使用了多重签名可信任的签名者的 Dapp 账户。从 Dapp 取款是由主节点负责处理的，当有人发送一条取款请求，Dapp 主节点就会处理它并且把资金从 Dapp 的地址上移出到 ASCH 区块链 上，或者比特币区块链上。开发者可以在他们自己的 Dapp 里面发行令牌，而且使用此令牌作为该 Dapp 的流通货币，这些令牌在该 Dapp 内可像 XAS 或者 BTC 一样使用，但是它不能直接从一个 Dapp 转移到另一个 Dapp，他们必须通过 ASCH 主链来转移。

4.3 沙箱和 VM

沙箱是一种按照安全策略限制程序行为的执行环境。早期主要用于测试可疑软件等， 比如黑客们为了试用某种病毒或者不安全产品，往往可以将它们在沙箱环境中运行。经典的沙箱系统的实现

途径一般是通过拦截系统调用，监视程序行为，然后依据用户定义的策略来控制和限制程序对计算机资源的使用，比如读写磁盘等。

ASCH 系统使用了 Node.js 的 VM 模块实现沙箱机制。VM 模块是对 Javascript 的 v8 引擎的封装，可以用来执行纯粹的 Javascript 代码，但无法使用系统层的 API，比如文件系统、网络传输相关的模块，并且由于没有 require 函数，第三方库也没法轻易导入进来，甚至无法进行模块化开发，这就需要 Dapp 的开发者使用 browserify 的技术将常用的第三方库打包成一个 js 文件，ASCH 的主链系统才能加载并运行。对于一些必须的系统级API，则通过进程间通讯的方法为侧链提供，这样兼顾了安全性与功能的完备性。

4.4 交易

ASCH 系统内建了一个交易抽象层，核心系统的几乎所有功能都是建立在交易上的，比如转账、投票、应用商店、充值、提现等。侧链本身也可以实现自己的不同类型的交易。交易之间的区别主要是交易类型和 asset。基础交易的数据结构如下，扩展部分会根据类型的不同分别存在不同的 asset 表中。

```
Transaction {
    required VARCHAR(20)      id;
    required VARCHAR(20)      blockId;
    required TINYINT          type;
    required INT required     timestamp;
    VARCHAR(21) optional      senderId;
    VARCHAR(21) required      recpientId;
    BIGINT required           amount;
    BIGINT required           fee;
    BINARY(64) optional        signature;
    BINARY(64) optional        signSignature;
    TEXT required              signatures;
    BINARY(32)                 senderPublicKey;
}
```

以投票交易举例来说， votes 实体通过交易 id 来关联到一个基础交易中。

```
Asset_Votes {
    required VARCHAR(20)        transactionId;
    optional  TEXT                      votes;
}
```

4.5 账户系统

ASCH 的每个账户由一个口令、一对公私钥、一个地址组成。用户还可以额外设置一个二级密码。注意这里与比特币有所不同的是，每个账户仅对应一个地址，而比特币中每个钱包对用多个地址和私钥。

口令（passphrase）是符合 BIP39 标准的用于产生确定性钱包的助记符。这种助记符与二进制或十六进制字符相比对人类记忆更友好。口令的生成方式是将一个 32bit 倍数长度的熵转换成若干个单词，ASCH 系统选择的熵长度为 128bit，将转换成 12 个单词。口令作为一级密码，由用户保管，不对外公开，一旦丢失用户将失去对应账户的所有权。口令形式如下：

```
barely decline dust stamp protect color certain cup arena busy latin shell
```

密钥对包括公钥和私钥，是以口令的 sha256 哈希做种子，再通过 ed25519 爱德华兹曲线签名算法生成的。形式如下：

```
公钥:
9989388b220a13465e49f52df5ba28ba08eb1e7a973320347f9687a107dc2f9a
私钥:
91e891f653e3ed0232d8c7de2e72b625d50d48593fc0fb570c0db25c5e44569a9989388b220a13465e49f52df5ba28ba08eb1e7a973320347f9687a107dc2f9a
```

账户地址是取公钥的 sha256 哈希的前 8 位，逆序后转换成 bignumber，其形式如下

```
5034187504202890358
```

4.6 客户端

ASCH 系统将提供三种客户端程序。

完整版客户端是针对超级用户、委托人和开发者的最佳解决方案，它可用于Windows，Mac OS 以及 Linux，但它只允许 Linux 运行受托人节点。

轻钱包的用户可以通过连接到完整版钱包以连接到网络，也可以直接调用 API，但前提是完整版钱包的所有者有开放该 API 权限，完整版钱包会通过点对点网络，从其它完整版钱包节点下载完整的区块链。

普通用户将主要使用轻钱包来管理自己的 ASCH 账户，它是一个精简版的 ASCH 钱包，轻钱包支持 Windows 和 Mac OS，它无需安装，它使用的是内嵌式的浏览器，它无法作为网络节点，因为它不下载区块数据，它只通过 http 连接到其它的节点，这样做能带来几点好处。首先它不下载区块数据，这意味着它会一直保持着较小的体积，不占容量；其次它不向网络广播密钥，所有数据在本地设计上签名，可以做所有类型的交易，如果你想运行一个受托人节点，你可以使用轻钱包注册一个受托人账号，但你无法使用轻钱包来运行受托人节点来创建区块，为了运行受托人节点，你需要下载完整版钱包，并运行在 Linux 上。Dapp 用户可以使用轻钱包来管理已安装的 Dapp。Dapp 的 API 和节点的API 也可供开发人员调用，这使得开发人员可以使用 Node.js 快速且简单地创建Javascript Dapp。

移动版客户端允许用户通过移动终端来操作自己的 ASCH 账户，它将提供 iOS 与安卓两种版本，并于苹果应用商店和安卓应用商店提供下载。它的后端将基于我们的桌面版的解决方案，与桌面版的区别将在于移动版钱包界面将使用响应式技术，自适应移动终端屏幕，并根据移动设计调整了一些交互方式。该 APP 使用了专为移动终端定制了易用的界面，类似于 Bitcoin 和一些常用的银行类 App 的界面，而且它将支持在内部运行所有你喜爱的 Dapp。

4.7 性能

一笔交易信息通过优化和压缩后大概占 100 字节，我们算算系统达到 1 万 TPS 时候需要消耗的带宽。因为出块间隔为 10 秒，那么每次出块需要包含 10 万个交易，也就是说要包含 10M 字节的交易数据，这 10MB 的数据需要在 10 秒内广播到全网，按最理想情况下，第一跳广播到 10 个节点，第二跳广播至 100 个节点，每一跳要在 5 秒内传输完毕， 服务器需要的带宽是 10MB * 10

/ 5 = 20MB，考虑到中间的带宽损耗和非理想情况，我们认为至少要 40MB 的带宽才能够满足 1 万 TPS 的吞吐量。这个带宽要求显然不低，但是相信 1 万 TPS 给委托人带来的收益远大于网络维护的费用。2014 年双十一支付宝的吞吐量峰值达到了 8.59 万每秒。ASCH 系统在交易吞吐量方面还是有优化空间的，这也是将来我们的重点要投入的方向。

5 应用场景

5.1 令牌系统

使用 ASCH 工具创建的第一个 hello world 应用，就是一个最基本的令牌系统了。

开发者可能不需要编写代码，只要在 genesis.json 文件里修改一些创世参数，就可以发布一个令牌系统了。ASCH 系统中的令牌与以太坊的子货币一样，可以表示黄金、股票、抵押物、或任意其他资产，这些令牌可以与转入侧链中的 XAS 通过去中心化的方式进行交易，从而实现流通，也可以在中心化的交易所与其他货币进行交易。

5.2 仲裁合约

假设一个买家想跟一个不认识的人进行交易，一般情况下如果交易顺利进行的话，双方都不希望有第三方介入，但是如果某个环节出了问题，比如买家对商品不满意时，他们就希望有一个中间人来做调解。这个中间人可能会要求买卖双方出示一些证据，然后做出判决，比如把钱退还一部分给买家。这个业务流程如下：

1. 买卖双方共同选择一个中间人
2. 买家使用三方的公钥创建一个 2-3 的多重签名账户，然后转账到这个账户，再以该账户为发起人，以卖家的账户为接收人，签署一个交易并发布出去。此时这个交易是不能立即被确认的，只有 3 个人中的 2 个人共同签名，才会生效。
3. 卖家发货给买家。
4. 如果买家收到货物后，检查没问题，使用自己的私钥对刚才的交易进行签名。然后卖家再次签名后，交易就顺利完成。
5. 如果买家对货物不满意，可以向中间人发起申诉并出示证据，卖家也可以出示证据， 最终由中

间人与买卖其中的一方达成一致，共同签署，完成交易，结束仲裁。

5.3 去中心化交易所

根据是否支持法币，可以分为两种程度的去中心化。如果不支持法币，可以实现完全的去中心化，如果支持法币，则只能实现半去中心化，即法币通过网关出入，但交易信息公开。

完全的去中心化交易所又分为两种，一种是点对点的交易，通过 ASCH 系统提供的“原子跨链交易API”来实现。另一种是挂单交易，挂单交易要求卖方从其他区块链转入一定的资产到 ASCH 侧链中，这个转入操作通过父链冻结资产的 SPV 证明达成。此外，由于关系数据库的支持，利用联表查询和索引功能，可以很容易的实现一个效率不错的撮合引擎。

5.4 存在性证明

存在性证明可以用于登记文件版权、专利等，其基本原理是将要存储的文件的哈希值存入到 ASCH 的侧链中，以此来证明某个特定文件存在，还可以加上时间戳、当事人的数字签名等元数据，来证明他们是在何时持有这些文件的。这些信息无法伪造、无法篡改， 不会暴露数据和隐私，在需要的时候随时可以验证且不依赖第三方机构。

5.5 物联网

物联网中存在海量的联网设备，很难有一个中央机构来管理所有的设备和各节点的身份。ASCH 的侧链是一个很好的解决方案，首先，它解决了节点间信任问题，设备间彼此相连形成分布式网络，通过共识算法来保证设备间交易的合法，并且可追踪、可审查、可分析。其次，不同种类的设备可以接入不同的侧链，这是我们前面提到的天然分片机制，避免了总账本的爆炸式增长。我们试想下，在一个基于区块链的物联网中，一个自动售货机不但可以监控和报告它自身的存货，还可以通过分析历史交易数据智能地从分销商那里进行招标并自动完成付款。

6 总结

ASCH 是一个去中心化的应用平台，其设计初衷是为了降低开发者的门槛，比如使用 Javascript 作为应用编程语言，支持关系数据库来存储交易数据，使得开发一个 Dapp 与传统的

Web 应用非常相似，相信这对开发者和中小型企业有很大的吸引力，只有开发者的生产力提高了，整个平台的生态才能够更迅速的繁荣起来。ASCH 在设计上也是开放的，并没有局限于某个细分领域，比如金融、文件存储、版权证明等，其提供的 API 都是较底层和抽象的，它们可以被自由组合实现各种不同的应用。在共识机制方面，ASCH 继承并增强了 DPOS 算法，大大降低了分叉几率和双重支付风险。另外，ASCH 的侧链即应用模式不但延缓了区块链膨胀问题，还使得 Dapp 更加的灵活和个性化。ASCH 是一个具有前瞻性的、低成本的一站式应用解决方案，相信将成为新一代去中心化应用的孵化器。