

一元购技术方案

V0.2（草稿）



2018-2-18

# 版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **时间** | **描述** | **修改人** | **授权人** |
| 0.1 | 2018年2月14号 | 创建说明书草案 | 郑嘉文 |  |
| 0.2 | 2018年2月18号 | 草案定稿 | 郑嘉文 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 参考文档

|  |  |  |
| --- | --- | --- |
| **简介** | **时间** | **文档名** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目录

[版本历史 2](#_Toc506767340)

[参考文档 3](#_Toc506767341)

[1 痛点 5](#_Toc506767342)

[2 技术选型 5](#_Toc506767343)

[3 系统架构 6](#_Toc506767344)

[3.1 注册登录 7](#_Toc506767345)

[3.2 代币合约 8](#_Toc506767346)

[3.3 智能合约 8](#_Toc506767347)

[4 工作量评估 9](#_Toc506767348)

# 痛点

*“一元购”是把一件商品平分成若干“等份”出售，其中每份1元，所有等份加起来的价值大于等于当前物品的价值（一般都是所有等份与物品价值相符合）。当一件商品所有“等份”售出后进行抽奖，幸运者即可获得此商品。是一种新型互联网产品，将购物与娱乐相结合。一元购，又叫一元云购，它是把商品分成若干等份来卖，一份1元，然后，从中抽取一名幸运者最终获得该商品，所以，也就是你支付1元，就有机会得到一个高价值的商品，这是一种新的营销方式，目前来看效果非常好，因为降低了消费门槛，人人都愿意试一下运气*。

*-- 百度百科*

一元购目前面临的风险主要是法律风险：“一元购”这种以商品销售行为掩盖公然开设网络赌场的网络博彩行为，相当于一个个庄家利用互联网开设的一个个线上赌场。所以所有的一元购技术方案必须经过合规认定。法律风险不在本方案讨论范围内。

从技术层面来说，对“一元购”的最大的质疑是其公平公正性。

* 游戏规则不透明

大多数的一元购平台对于游戏规则都是语焉不详，尤其是对于规则的细节以及数据的来源。

* 存在暗箱操作的可能

“一元购”的中奖算法大部分是“商品最后购买时间前网站所有商品的最后50~100条购买时间记录”，虽然是取自服务器时间，且精确到毫秒，但这属于站内数据，而站内数据意味着“一元购”平台对这些数据是可控的。在一个自定游戏规则，且缺乏有效监督的网络平台，很难保证公平公正。

* 不具备事后追踪

由于一元购平台保有所有的数据，对于事后的追查，没有很高的公信力

# 技术选型

方兴未艾的区块链技术很好的解决了上一节里描述的痛点。

* 不可更改的共享账本机制(immutable share ledger)

由于共享账本，所有的信息都是公开透明的。而且信息一旦上链，信息即不可篡改

* 可溯源(Traceability)

所有程序运行的结果都在链上，只要具有一定的IT知识和技巧，都可以到链上查询

* 匿名

链上所有ID的代表就是一个地址，和人的物理信息，社会身份信息没有必然的联系。极好的保护了用户的隐私。

在考察了市场上大多数的链后，以太坊（Ethereum）无疑是最适合用来做一元购技术平台的选择。

* 以太坊自诞生以来，到现在已经是一个成熟的基础设施应用。

已经有很多的Dapps在以太坊上面运行。比如其中最著名的以太猫。这充分证明了以太坊的健壮性和可编程性能。相对于其他的公链，具有无可比拟的优势

* 以太坊已经拥有了一个庞大的P2P节点网络，和矿工人群。

由于以太坊已经建设成功了庞大的社区群，可以很轻松的抵御51%攻击和双花攻击。让一元购平台聚焦在实现商业逻辑，而不用花大力气，大成本在建立公链，和保护公链免受攻击上。

* 以太坊提供了完备编程语言和编程接口

以太坊提供了Solidity智能编程语言和定义好的编程接口。在此基础上的平台开发和维护都变得比较简单和轻松。

* 以太坊的ERC20代币

以太坊可以很容易的发行代币，从而很容易的建立基于代币的经济学模型。ERC20代币在ICO风行的时候得到了大规模的验证。现在几乎所有的钱包都支持ERC20代币。

* 代码开源

所有智能合约代码均位于以太坊的公链上，任何人可以随意查看，理解并验证。

* 以太坊的工具完备

以太坊目前具备了公链圈里最完备的工具集。用户稍加训练，即可掌握。

当然，以太坊也不是十全十美的，从技术上来说，也面临着很多不确定性：

* 网络拥堵

前段时间以太猫流行的时候，造成了以太网络的拥堵。平常发送一个币几分钟就可以到帐，而拥堵的时候有可能几个小时也无法到帐。这个是由于以太坊本身的设计造成的。简言之，就是以太坊的基础设施建设跟不上需求

* 交易费用高昂

随着ETH的日渐坚挺，由于所有基于以太坊的交易都是要付Gas费用，导致交易费用高企。

* 技术方向

最现实的方向改变就是Vitalik主导的POS改造。如果Vitalik的POS改造确定发生并且被社区接受，对现有以太坊上运行的Dapps会造成什么样的影响，尚不确定。

* 于现实世界的连接性

以太坊并没有接口直接连接到实体世界。比如，天气预报的数据就无法实时接入以太网络

# 系统架构

本系统的架构基于以太坊和以太坊提供的EVM机制以及智能合约机制（Solidity）。为了便于EVM调试，将使用zeppelin库。系统主要由3个部分（Component）组成

* 注册登录

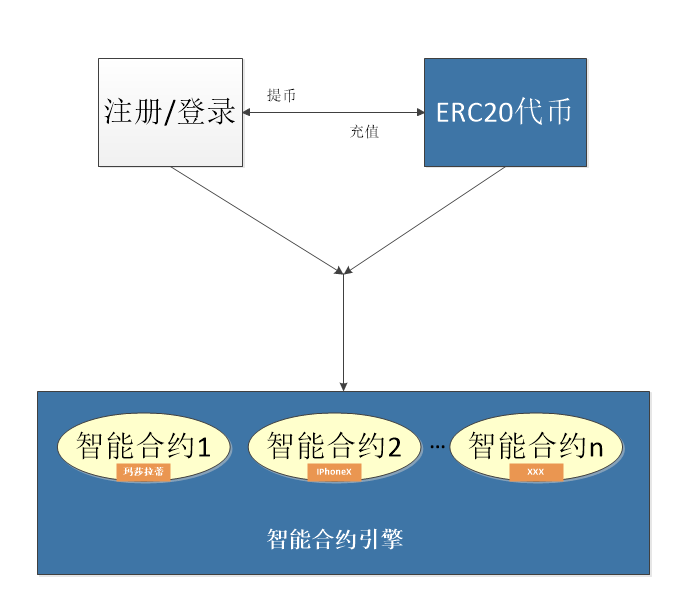
所有想参与一元购的用户，必须在指定的网站上注册，获取相应的以太坊地址。用于购买代币，提币，投资某个特定的活动（从技术上来说，就是一个智能合约）。如果已经有以太坊地址的，可以略过这一步。

* ERC20代币合约

所有参与一元购活动的用户必须通过购买平台的代币实现。当然，相应利益的获取也会以代币的形式获取。

* 智能合约的引擎

每次一元购的活动，就是一个智能合约。满足合约预先指定的条件，该合约被触发自动执行。



## 注册登录

注册登录模块主要的功能是给每个参加一元购活动的用户分配以太坊地址。此功能的目的就是为用户提供傻瓜式服务，获取参与一元购活动的ID。整个登录过程包括：

* 给每个注册用户分配一个公钥/私钥对

公钥作为用户身份的唯一ID,用来收发代币。

私钥有两种选择：

* + 私钥存储在服务器端。

这样的好处，是可以在用户忘记私钥的时候，由服务器找回私钥。为此付出的代价就是：服务器是中心化的，信息容易受到攻击，而且容易被故意泄露

* + 私钥由用户自己保存和管理，服务器不保存副本

好处就是安全性高。潜在的风险就是如果用户管理不善，弄丢了私钥，相应的数字资产也就丢失了，无法找回。

* 每个用户注册过程中提供助记词

主要是通过助记词来复原用户的私钥

* 如果用户已经拥有以太坊公钥/私钥对，可以略过这一步

## 代币合约

一元购用户必须通过购买一元购平台代币的形式参与一元购活动。所以，一元购平台必须能发行代币。这个已经是以太坊的标准 – ERC20

contract ERC20 {

        function totalSupply() constant returns (uint totalSupply);         //获取总的发行量

     function balanceOf(address \_owner) constant returns (uint balance); //查询账户余额

        function transfer(address \_to, uint \_value)returns(bool success); // 发送Token到某个地址(转账)

        function transferFrom(address \_from, address \_to, uint \_value) returns (bool success); //从地址from 发送token到to地址

        function approve(address \_spender, uint \_value)returns(bool success);//允许\_spender从你的账户转出token

        function allowance(address \_owner, address \_spender) constant returns (uint remaining);//查询允许spender转移的Token数量

        event Transfer(address indexed \_from, address indexed \_to, uint \_value);//transfer方法调用时的通知事件

        event Approval(address indexed \_owner, address indexed \_spender, uint \_value); //approve方法调用时的通知事件

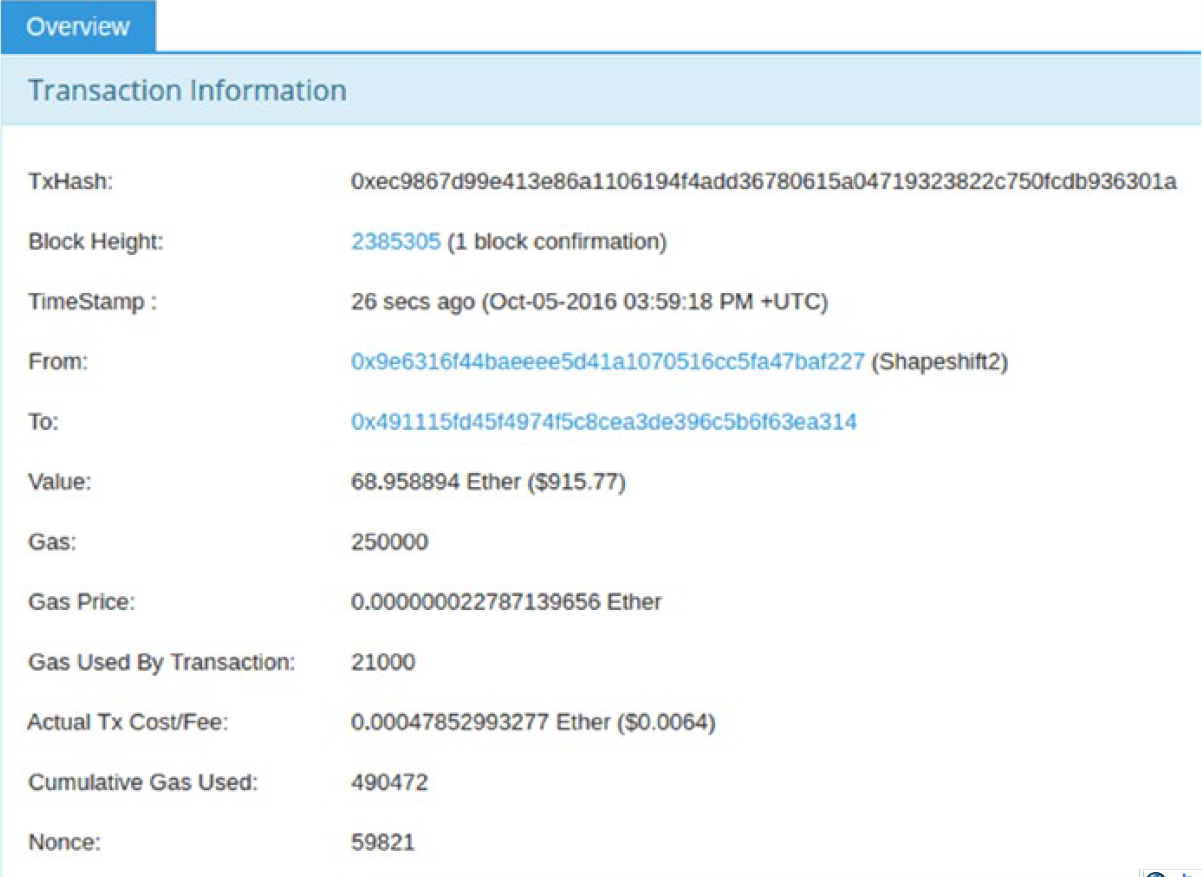
}

## 智能合约

对于每一次一元购活动，会将包含活动的商业逻辑的代码以智能合约的方式部署在以太坊公链上。所有的代码都是公开透明的，可以通过以下工具查看：

* <https://www.etherchain.org/>
* <https://etherscan.io>
* 类似的工具还有很多

下面是一个在以太坊公链上查询交易的截图：



鉴于现在交易费用(Gas)高企，而且在EVM里使用持久化存储都需要付Gas，建议使用尽量简单的，尽量随机的算法。

# 工作量评估

工作量初步评估如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **工作** | **工作量（人天）** | **备注** |
| 1 | ERC20代币合约编写及部署 | 5 |  |
| 2 | 注册和登录功能（Web端） | 5 |  |
| 3 | 注册和登录功能（移动端） | 5 | 可选（Optional） |
| 4 | 一元购活动商业逻辑编码实现及测试 | 10-20天 | 依赖于商业规则的复杂程度2周到4周不等 |
| 5 | 整体测试 | 5天 |  |
| 总共 |  | 30天-40天 |  |
| 待定 | 有必要的话，可以邀请数学家参与，设计尽量随机的算法，保证平台的公正性 | 未定 |  |