目录

[基于区块链技术投票应用 3](#_Toc19209)

[一． 绪论 3](#_Toc26436)

[1.1 选题背景 3](#_Toc5347)

[1.2 国内外现状 4](#_Toc16379)

[1.3 研究目标与内容 5](#_Toc20040)

[二．项目技术探究分析 6](#_Toc12886)

[什么是区块链技术？ 6](#_Toc24636)

[1.1 区块链起源 7](#_Toc29575)

[1.2 区块链概念 7](#_Toc8480)

[1.3 区块链类型 8](#_Toc26265)

[1.4 核心技术 9](#_Toc23891)

[1.5 区块链技术应用领域 11](#_Toc19943)

[1.6 以太坊智能合约 14](#_Toc16213)

[1.6.1 以太坊 14](#_Toc7483)

[1.6.2 EVM 18](#_Toc14828)

[1.6.3 智能合约 21](#_Toc6618)

[三．系统需求分析 23](#_Toc3194)

[3.1 系统概述 23](#_Toc8208)

[3.2 系统业务分析 24](#_Toc31453)

[1. 投票类型及其发起人 24](#_Toc25079)

[2. 此系统业务内容 24](#_Toc23963)

[3. 系统需求分析 25](#_Toc20872)

[投票流程图如下3.1 25](#_Toc5160)

[四． 系统概要设计 27](#_Toc14401)

[4.1 应用建设目标 27](#_Toc29309)

[4.2 设计原则 27](#_Toc20703)

[4.3 总体架构 28](#_Toc19559)

[4.4 智能合约代码解释 29](#_Toc7163)

[1.合约变量 29](#_Toc11923)

[2. 准备选票 31](#_Toc28525)

[3. 投票阶段 34](#_Toc29118)

[4. 结束投票 35](#_Toc31907)

[五． 系统模块介绍 36](#_Toc17691)

[1.1 投票管理Dapp模块代码演练 36](#_Toc5162)

[1.1.1配置模块 36](#_Toc16840)

[1.1.2 监控与等待模块 39](#_Toc27647)

[1.1.3 更新DApp的用户界面模块 42](#_Toc2518)

[1.1.4 开始新的投票 模块 45](#_Toc31183)

[1.2 参与者投票Dapp模块代码演练 46](#_Toc12439)

[1.2.1 配置模块 46](#_Toc3734)

[1.2.2 显示提案模块 48](#_Toc27202)

[1.2.3 监控模块 49](#_Toc16435)

[1.2.4 投票信息模块 50](#_Toc26039)

[六. 投票 Dapp 功能演示 52](#_Toc6293)

[1. 现在访问投票管理器。 52](#_Toc2414)

[2. 决定谁可以投票 54](#_Toc32629)

[3. 选民投票 56](#_Toc21720)

[4. 结束投票并取得结果 57](#_Toc21589)

[七. 总结局限性与改进 58](#_Toc3741)

[参考文献 59](#_Toc11035)

# 基于区块链技术投票应用

## 作者:郭斌

## 绪论

### 选题背景

投票调查活动是在生活中常见的选举方式，小到学校的学生干部选拔， 企业的发展方案选择和董事会选举，大到人大代表议案建议的评审， 都是投票活动的应用场景， 同时它也在其中扮演着非常重要的角色。 比如审议通过的议案建议，与我们的政策息息相关； 市场研究决定了公司的后续决策和发展计划。 随着技术的进步和网络技术的创新，电子投票应运而生。

线上的投票网站、微信投票程序的出现，解决了传统投票人力资源的消耗，摆脱了空间和时间的限制，提高了投票过程和机票过程的效率。区块链本质上是链式数据结构，其中数据块以时间序列顺序方式连接。 在此基础上，加密方法用于保证无法修改的分布式帐簿。 区块链的核心是全球可编程和开放的分布式数据库。针对线下的投票效率非常低、造成了浪费资源，线上的网络投票频频遭遇刷票、欺诈选票以及电子投票系统的安全性、公平性、匿名性等问题，本文利用区块链技术的优势， 设计并实现了基于区块链的电子投票系统，以确保所有投票数据的安全性。可靠性，真实性和完整性。投票结果可验证且不可篡改， 提升公众信任度。

本文介绍了以太坊开发的关键技术，分析了区块链技术应用于投票系统的优势和可行性。给出了基于区块链的电子投票应用的需求分析、系统架构设计，包括区块链层和智能合约的设计，以及各个模块和数据库的详细设计方案，实现了公开网络投票、大规模投票、代表( 授权) 投票等多种投票方式的主要功能，解决了电子投票系统中的信任问题，实现了分散化。最后，给出了以太坊电子投票系统的可行性结论和展望。

### 国内外现状

随着国内互联网的快速发展和互联网普及率的提高，更快，更方便，用户导向的在线投票系统逐渐取代了线下投票。同时，随着微信、微博越发受到广大网民的喜爱和欢迎 新流行的微博投票、微信公众号投票也逐渐成为近几年的流行趋势。但是现有的网上投票系统依然存在投票人的隐私没有得到正当保护，投票数据面临被篡改的危险、 第三方可信度无法得到保证、投票结果不可验证等诸多问题。

区块链技术具备两大特性，即不能篡改和交易可追溯，这样有利于解决线上投票无法保证公开、公正的一些问题，例如欺诈选票、不诚实的第三方、选民隐私泄露、恶意敌手攻击等问题，保障偷票方合法权益。在网络技术领域，区块链技术可以说是一项革命性的创新， 也是当前研究的热门话题。

国外互联网发展一直优于国内， 但是国内外面临着相同的问题： 网上投票的公平性和公平性以及数据的真实完整性无法得到保证。所以在重大的选举活动中，尤其是由政府开展的重要投票选举中，多数国家和政府机构还是选择了传统的线下投票方式， 虽然费时费力，但在国家作为诚信第三方的强有力监督下，篡改选票的可能性相对网络投票更小一些。因此，网络上自由组织开展的大规模电子投票依然需要技术的优化和革新。区块链技术的价值也吸引了许多外国组织的探索和研究。使投票更加高效、便捷，同时保证投票的公正和安全是所有国内外开发者共同的目标。

### 1.3 研究目标与内容

为了解决线上电子投票系统存在的一系列问题， 设计一个去中心化的、可信赖的以太坊电子投票系统，满足个人、企业以及政府的投票、调研、 选举等多种活动的需求， 该投票系统需要实现如下三个目标：

(1) 确保投票数据不会被篡改并有效避免欺诈性投票：

(2) 投票数据具有可追溯性和可验证性；

(3) 保护选民和选票的隐私， 实现匿名投票；

区块链技术具有很多适合投票系统的优点， 包括数据可追溯、 去中心化、 安全可信等， 交易数据一旦被认证为有效， 并且存入区块链就无法被任何人篡改，且可以被所有节点进行查看和验证操作。 区块链是全球范围内的不依赖第三方、保证数据真实性和完整性的值得被公众信赖的数据库， 而以太坊能够灵活地利用智能合约进行开发应用程序， 因此基于区块链技术的以太坊平台正是最具适应性的底层平台和技术支持， 确保电子投票系统的公平性和安全性。

本文旨在将区块链技术和以太坊平台应用于电子投票系统的设计和实现， 充分利用区块链技术分散， 数据不被篡改， 可靠安全的特点。 设计一个保护用户隐私、 保障投票安全的电子投票系统， 真正解决线上投票存在的篡改投票结果、 刷票等问题。

## 二．项目技术探究分析

### 什么是区块链技术？



### 区块链起源

区块链起源于比特币，2008年11月1日，一位自称中本聪(Satoshi Nakamoto)的人发表了《比特币:一种点对点的电子现金系统》一文，阐述了基于P2P网络技术、加密技术、时间戳技术、区块链技术等的电子现金系统的构架理念，这标志着比特币的诞生。两个月后理论步入实践，2009年1月3日第一个序号为0的创世区块诞生。几天后2009年1月9日出现序号为1的区块，并与序号为0的创世区块相连接形成了链，标志着区块链的诞生。

近年来，世界对比特币的态度起起落落，但作为比特币底层技术之一的区块链技术日益受到重视。在比特币形成过程中，区块是一个一个的存储单元，记录了一定时间内各个区块节点全部的交流信息。各个区块之间通过随机散列(也称哈希算法)实现链接，后一个区块包含前一个区块的哈希值，随着信息交流的扩大，一个区块与一个区块相继接续，形成的结果就叫区块链。

### 区块链概念

什么是区块链？从科技层面来看，区块链涉及数学、密码学、互联网和计算机编程等很多科学技术问题。从应用视角来看，简单来说，区块链是一个分布式的共享账本和数据库，具有去中心化、不可篡改、全程留痕、可以追溯、集体维护、公开透明等特点。这些特点保证了区块链的“诚实”与“透明”，为区块链创造信任奠定基础。而区块链丰富的应用场景，基本上都基于区块链能够解决信息不对称问题，实现多个主体之间的协作信任与一致行动。

区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链（Blockchain），是比特币的一个重要概念，它本质上是一个去中心化的数据库，同时作为比特币的底层技术，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一批次比特币网络交易的信息，用于验证其信息的有效性（防伪）和生成下一个区块。

比特币白皮书英文原版实并未出现 blockchain 一词，而是使用的 chain of blocks。最早的比特币白皮书中文翻译版中，将 chain of blocks 翻译成了区块链。这是“区块链”这一中文词最早的出现时间。

### 区块链类型

公有区块链：

公有区块链（Public Block Chains)是指：世界上任何个体或者团体都可以发送交易，且交易能够获得该区块链的有效确认，任何人都可以参与其共识过程。公有区块链是最早的区块链，也是应用最广泛的区块链，各大bitcoins系列的虚拟数字货币均基于公有区块链，世界上有且仅有一条该币种对应的区块链。

联合（行业）区块链：

行业区块链（Consortium Block Chains)：由某个群体内部指定多个预选的节点为记账人，每个块的生成由所有的预选节点共同决定（预选节点参与共识过程），其他接入节点可以参与交易，但不过问记账过程(本质上还是托管记账，只是变成分布式记账，预选节点的多少，如何决定每个块的记账者成为该区块链的主要风险点），其他任何人可以通过该区块链开放的API进行限定查询。

私有区块链：

私有区块链（Private Block Chains)：仅仅使用区块链的总账技术进行记账，可以是一个公司，也可以是个人，独享该区块链的写入权限，本链与其他的分布式存储方案没有太大区别。传统金融都是想实验尝试私有区块链，而公链的应用例如bitcoin已经工业化，私链的应用产品还在摸索当中。

我们此项目主要采取在 Rinkeby 测试链上工作。

### 核心技术

分布式账本：

分布式账本指的是交易记账由分布在不同地方的多个节点共同完成，而且每一个节点记录的是完整的账目，因此它们都可以参与监督交易合法性，同时也可以共同为其作证。

跟传统的分布式存储有所不同，区块链的分布式存储的独特性主要体现在两个方面：一是区块链每个节点都按照块链式结构存储完整的数据，传统分布式存储一般是将数据按照一定的规则分成多份进行存储。二是区块链每个节点存储都是独立的、地位等同的，依靠共识机制保证存储的一致性，而传统分布式存储一般是通过中心节点往其他备份节点同步数据。没有任何一个节点可以单独记录账本数据，从而避免了单一记账人被控制或者被贿赂而记假账的可能性。也由记账节点足够多，理论上讲除非所有的节点被破坏，否则账目就不会丢失，从而保证了账目数据的安全性。

非对称加密：

存储在区块链上的交易信息是公开的，但是账户身份信息是高度加密的，只有在数据拥有者授权的情况下才能访问到，从而保证了数据的安全和个人的隐私。

共识机制：

共识机制就是所有记账节点之间怎么达成共识，去认定一个记录的有效性，这既是认定的手段，也是防止篡改的手段。区块链提出了四种不同的共识机制，适用于不同的应用场景，在效率和安全性之间取得平衡。

区块链的共识机制具备“少数服从多数”以及“人人平等”的特点，其中“少数服从多数”并不完全指节点个数，也可以是计算能力、股权数或者其他的计算机可以比较的特征量。“人人平等”是当节点满足条件时，所有节点都有权优先提出共识结果、直接被其他节点认同后并最后有可能成为最终共识结果。以比特币为例，采用的是工作量证明，只有在控制了全网超过51%的记账节点的情况下，才有可能伪造出一条不存在的记录。当加入区块链的节点足够多的时候，这基本上不可能，从而杜绝了造假的可能。

智能合约：

智能合约是基于这些可信的不可篡改的数据，可以自动化的执行一些预先定义好的规则和条款。以保险为例，如果说每个人的信息（包括医疗信息和风险发生的信息）都是真实可信的，那就很容易的在一些标准化的保险产品中，去进行自动化的理赔。在保险公司的日常业务中，虽然交易不像银行和证券行业那样频繁，但是对可信数据的依赖是有增无减。因此，笔者认为利用区块链技术，从数据管理的角度切入，能够有效地帮助保险公司提高风险管理能力。具体来讲主要分投保人风险管理和保险公司的风险监督。

### 区块链技术应用领域

金融领域

区块链在国际汇兑、信用证、股权登记和证券交易所等金融领域有着潜在的巨大应用价值。将区块链技术应用在金融行业中，能够省去第三方中介环节，实现点对点的直接对接，从而在大大降低成本的同时，快速完成交易支付。

比如Visa推出基于区块链技术的 Visa B2B Connect，它能为机构提供一种费用更低、更快速和安全的跨境支付方式来处理全球范围的企业对企业的交易。要知道传统的跨境支付需要等3-5天，并为此支付1-3%的交易费用。Visa 还联合 Coinbase 推出了首张比特币借记卡，花旗银行则在区块链上测试运行加密货币“花旗币”。

物联网和物流领域

区块链在物联网和物流领域也可以天然结合。通过区块链可以降低物流成本，追溯物品的生产和运送过程，并且提高供应链管理的效率。该领域被认为是区块链一个很有前景的应用方向。

区块链通过结点连接的散状网络分层结构，能够在整个网络中实现信息的全面传递，并能够检验信息的准确程度。这种特性一 定程度上提高了物联网交易的便利性和智能化。区块链+大数据的解决方案就利用了大数据的自动筛选过滤模式，在区块链中建立信用资源，可双重提高交易的安全性，并提高物联网交易便利程度。为智能物流模式应用节约时间成本。区块链结点具有十分自由的进出能力，可独立的参与或离开区块链体系，不对整个区块链体系有任何干扰。区块链 +大数据解决方案就利用了大数据的整合能力，促使物联网基础用户拓展更具有方向性，便于在智能物流的分散用户之间实现用户拓展。

公共服务领域

区块链在公共管理、能源、交通等领域都与民众的生产生活息息相关，但是这些领域的中心化特质也带来了一些问题，可以用区块链来改造。区块链提供的去中心化的完全分布式DNS服务通过网络中各个节点之间的点对点数据传输服务就能实现域名的查询和解析，可用于确保某个重要的基础设施的操作系统和固件没有被篡改，可以监控软件的状态和完整性，发现不良的篡改，并确保使用了物联网技术的系统所传输的数据没用经过篡改。

数字版权领域

通过区块链技术，可以对作品进行鉴权，证明文字、视频、音频等作品的存在，保证权属的真实、唯一性。作品在区块链上被确权后，后续交易都会进行实时记录，实现数字版权全生命周期管理，也可作为司法取证中的技术性保障。例如，美国纽约一家创业公司Mine Labs开发了一个基于区块链的元数据协议，这个名为Mediachain的系统利用IPFS文件系统，实现数字作品版权保护，主要是面向数字图片的版权保护应用。

保险领域

在保险理赔方面，保险机构负责资金归集、投资、理赔，往往管理和运营成本较高。通过智能合约的应用，既无需投保人申请，也无需保险公司批准，只要触发理赔条件，实现保单自动理赔。一个典型的应用案例就是LenderBot, 是 2016 年由区块链企业 Stratumn、德勤与支付服务商 Lemonway 合作推出，它允许人们通过 Facebook Messenger 的聊天功能，注册定制化的微保险产品， 为个人之间交换的高价值物品进行投保，而区块链在贷款合同中代替了第三方角色。

公益领域

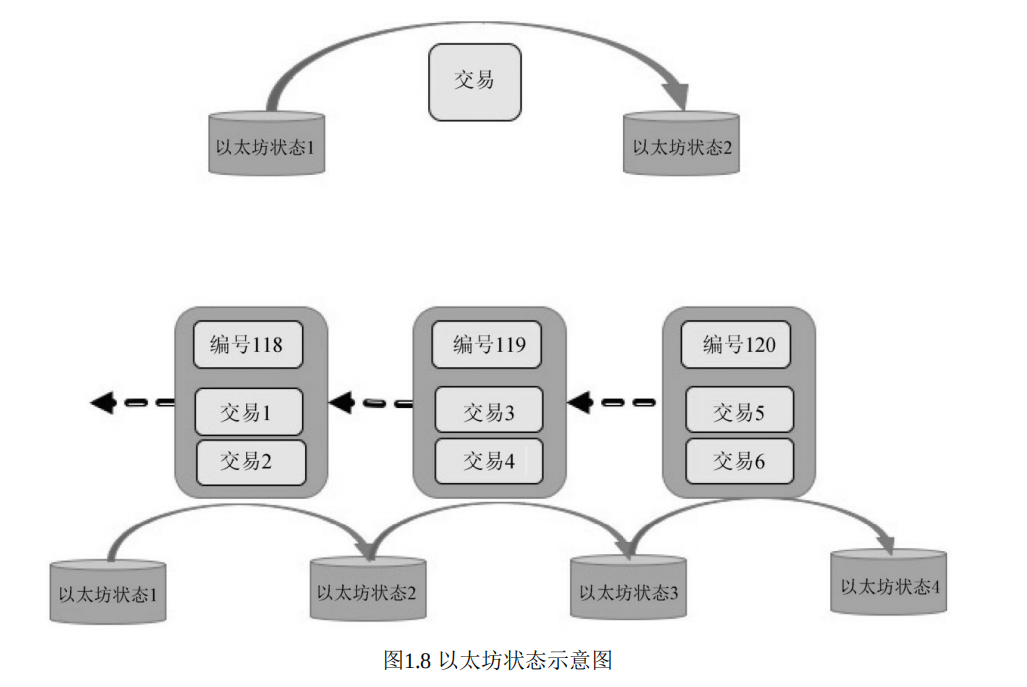
区块链上存储的数据，高可靠且不可篡改，天然适合用在社会公益场景。公益流程中的相关信息，如捐赠项目、募集明细、资金流向、受助人反馈等，均可以存放于区块链上，并且有条件地进行透明公开公示，方便社会监督。

### 以太坊智能合约

在讲解了区块链的基本知识之后，我们聚焦于以太坊这条公链的独有特征，除了拥有上述区块链的基本功能，以太坊还在比特币网络的基础上增加了以太坊智能合约虚拟机，即EVM。在增加了EVM 之后，以太坊就成了一个可编程的去中心化平台，任何系统开发者在支付一定的部署费用之后，就可以拥有一套完全去中心化的业务系统，这样的业务系统被称作 DAPP.

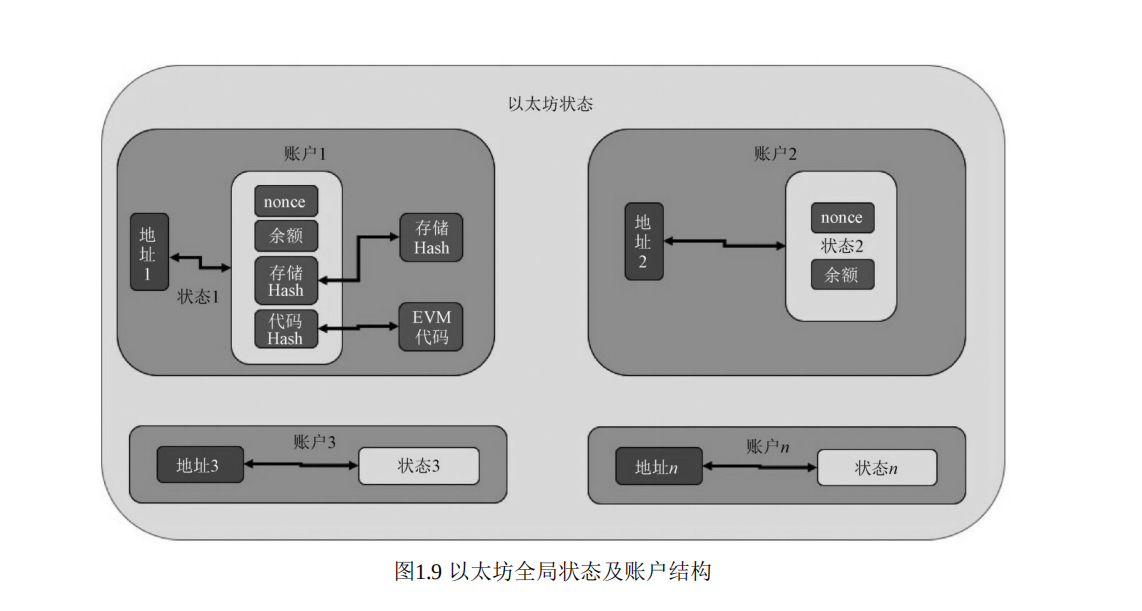
### 以太坊

在前一节讲解区块链基本知识的基础上，这里简单讲解以太坊公链特有的公链特性。首先以太坊可以视作大型的状态机，由分布在全球的 以太坊节点来运行这个状态机，每一次交易的产生都会修改状态机的状态。将交易打包成区块之后，可以以区块为单位来衡量状态的迁移，如 图1.8所示。

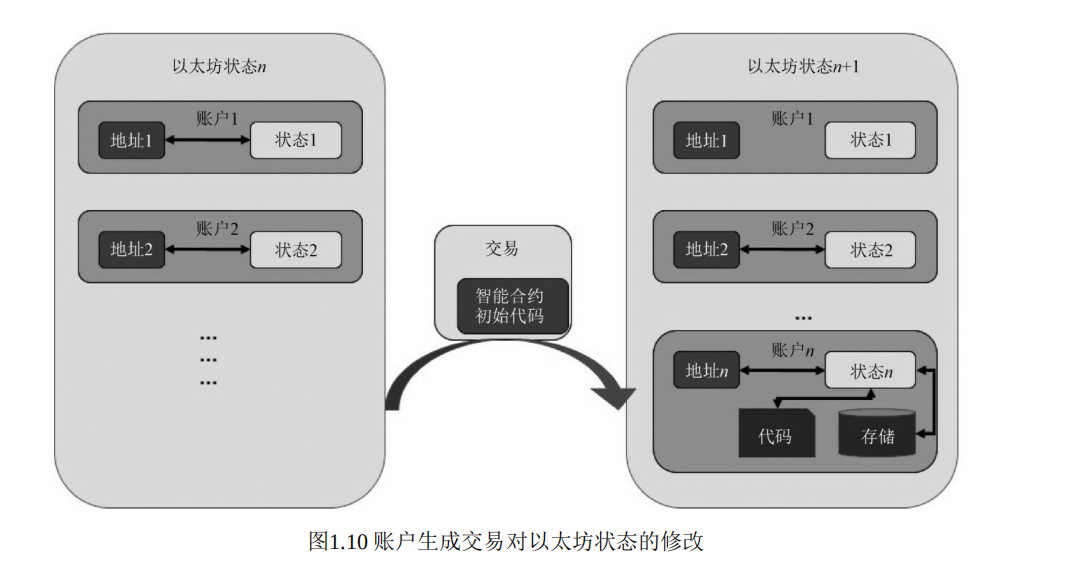


以太坊的状态是由以太坊上的所有账户组成的，也就是说，以太坊 上的所有账户组成了以太坊的全局状态，以太坊账户是地址与账户状态的一个映射结构。以太坊账户又分为两种：一种是外部账户，也就是用于存放用户余额和转账的账户；另一种是智能合约账户，是在部署智能合约时生成的一个关于该智能合约的区块链地址及其状态的映射关系。

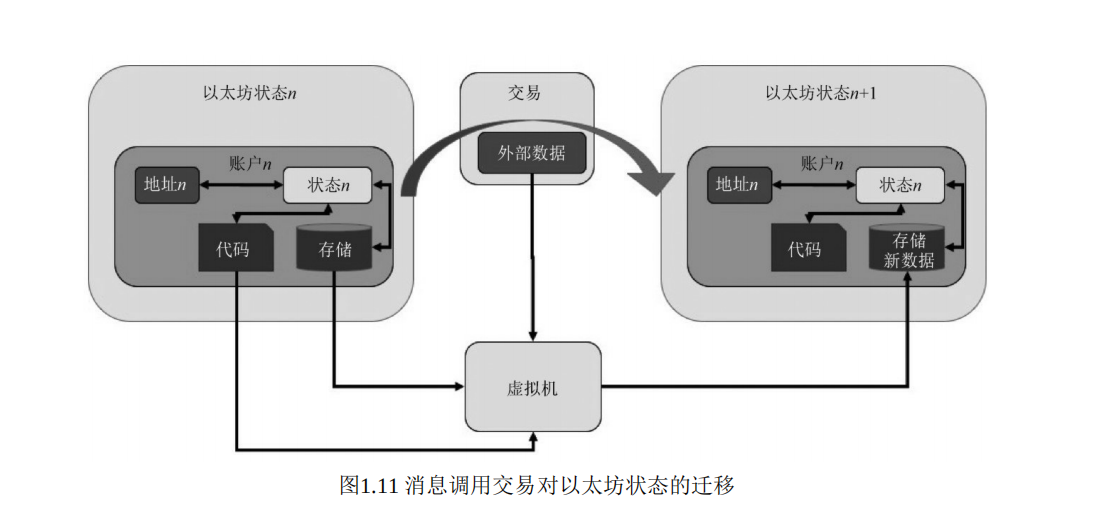
以太坊全局状态及账户结构如图1.9所示。账户1是智能合约账户，其地址映射到一个账户状态，这个状态由4部分组成：nonce、余额、存储数据的Hash值、EVM代码的Hash值。存储空间是智能合约虚拟机运行时需要的存储介质，EVM 代码是智能合约代码编译并部署到以太坊之后的数据。账户2是外部账户，该账户的状态仅仅由nonce和余额组成。以太坊账户由这两类账户组成，所有这些账户组成了以太坊的全局状态。



以太坊上的账户之间会有系统调用，这里仅仅讲解与DAPP相关的系统调用。首先是账户生成交易对以太坊状态的修改，如图1.10所示。每次创建和部署智能合约，都会创建一个新的账户，创建账户的代码包含在交易中，交易可以由外部账户发起，也可以由合约账户发起，创建合约账户的代码通过Solidity编程语言编写，通过以太坊的智能合约编译器编译生成。



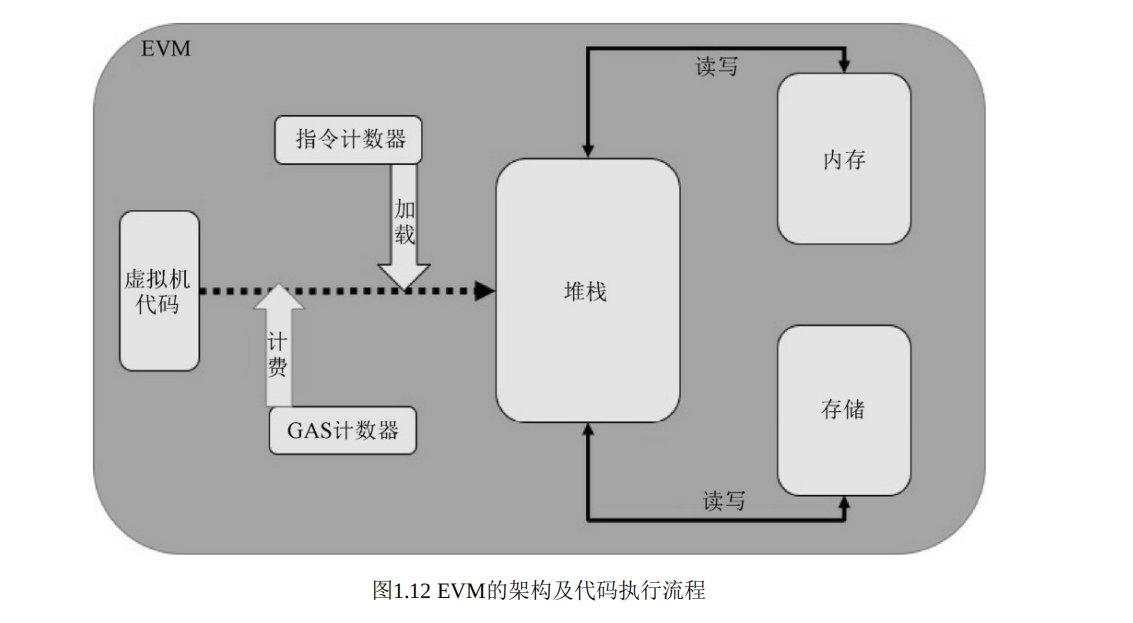
其次，账户生成之后，会开辟账户存储空间和以太坊虚拟机可以理解并执行的代码，如图1.11所示。当智能合约生成之后，可以通过ABI接口调用该智能合约，这种信息调用也会对区块链的状态发送读写操作。通过ABI接口传入外部数据，然后通过以太坊智能合约虚拟机读取合约代码，结合输入的数据和区块链上存储的数据，修改区块链状态，并将最新的状态信息存放在以太坊公链上。



### EVM

在以太坊之前的公链项目中，如果需要修改某条公链的某些特征或者增加对某种场景的支持，开发者必须在原有的公链设计的基础上修改系统底层源代码，并重新维护一套公链生态，这往往被称作硬分叉。比如比特币的众多分叉币，开发人员为了修改诸如区块大小的参数，需要重新分裂一条公链，这是一项极其繁重的工作，不仅会消耗资源、时间，有时候还会分裂原有公链的共识社区。而以太坊 EVM的出现，使得任何需要实现某一行业具体逻辑的开发者，无须复制以太坊的整套代码，然后修改出符合自己逻辑的公链，而是基于以太坊现有的公链网络和矿机组织、共识社区，通过EVM提供的编程API来编写智能合约，就可以完成一套区块链系统，一套满足自己业务需求的系统，该系统具有区块链所有的通用特征：去中心化、公开透明、无法篡改等。

因此，EVM的出现使得对区块链编程成为可能，具有EVM的以太坊公链技术可以视作是对原有区块链技术的一次重大革新。如果把区块链比作PC操作系统，那么EVM就是类似于JVM一样的运行环境，Solidity智能合约就是类似于JAVA的高级编程语言，而EVM出现之前的比特币网络仅有有限的指令可以对区块链编程，有点类似于底层的汇编语言编程。从这个角度来讲，EVM使得区块链高级语言编程成为可能。EVM作为一个离线的、独立的运行环境，它无法访问外界的文件系统、网络接口等资源，目前EVM的指令中尚未支持对这些资源的操作。当编写的智能合约被编译成 EVM 能够理解的代码之后，EVM会在自己独立的运行环境中执行用户编写的智能合约程序，EVM的架构及代码执行流程如图1.12所示。



通过智能合约编译器，可以将智能合约编译为虚拟机能够理解的指令，然后通过以太坊账户发起一个交易，将这些指令部署到区块链上，部署成功后会得到一个以太坊的地址，这个地址指明了该合约代码存放的位置.

当合约被调用时，可以根据合约的地址找到指令集的存储位置，虚拟机通过程序指令计数器记录当前指令的执行位置，随着程序的执行，指令计数器也会随之变化，它存储了下一条需要执行的指令的地址（偏移地址）。在指令中会有JUMP之类的跳转指令，因此指令计数器的数值并非总按顺序逐渐增加。EVM的指令执行不在寄存器中进行，而是在一个被称作栈的内存空间中进行。这个栈最多可以容纳1024条栈指令，每条栈指令可以达到256位，也就是说EVM是256位的计算机，这个长度的指令特别适合Keccak-256 hash数据及椭圆函数运算。

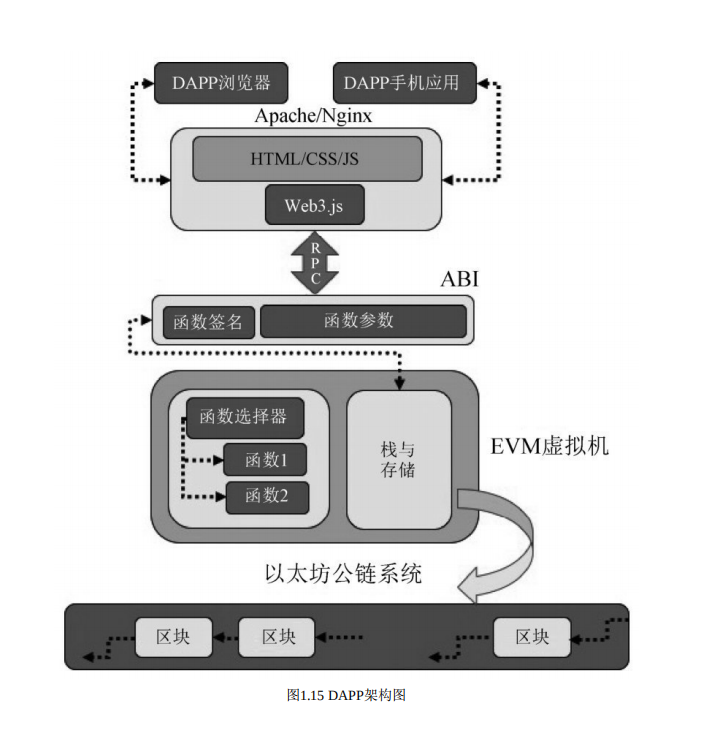
智能合约的指令中包含了操作码和操作数，EVM加载到栈中之

后，根据操作码运算，在执行操作码的过程中，会产生中间状态的临时数据，这些数据会被存放在栈空间上、内存中或者持久存储的空间上。指令中也有从内存或者持久存储空间中读取数据的操作，有些操作需要在区块链状态的基础上进行下一步的操作，这些状态信息就存储在持久存储空间上；有些操作是存储的运算，这些数据一般通过内存进行读写。使用内存和持久存储空间需要支付一定的 GAS 费用。GAS 指的是EVM执行指令时消耗的代价，这个代价以以太坊的数字货币ETH来表示。

### 智能合约

Solidity编程语言解决了编写智能合约的不友好的问题，但是当合约编译并部署之后，对于一般的使用者来说，访问这些接口的门槛较高。为了使广大用户理解并方便快捷地访问区块链及区块链上的智能合约系统，开发者必须提供操作界面和结果查看界面，来简化用户访问和操作区块链的方式。因此一套完整的区块链 DAPP，除智能合约这些可以查询和改变区块链状态的代码外，还需要用户操作界面及连接用户操作与智能合约代码的接口。

图1.15所示为DAPP架构图，该架构图包含了从用户通过可视化界面发起操作到以太坊状态机发生改变的全流程。首先，用户通过 Web界面或者手机 APP 将操作数据发送到一个传统的业务服务器，该业务服务器是传统互联网中心化的服务器，但是与传统系统不同的是，该系统没有像传统互联网设计那样将数据放入中心化的数据库存储，而是通过一个Web3.0接口，将数据传送到以太坊区块链公链上。该接口是一个JSON RPC协议，该协议有很多代码实现。目前最流行的是运行在Web容器中的Web3.js模块。Solidity编程语言经过编译之后，除了交易需要的合约初始化代码之外，还有API接口等描述文件，Web3.js通过这些描述文件，可以构建与以太坊智能合约虚拟机进行通信的模块，通过JS（全称JavaScript，下文都简称JS）代码将用户的操作数据传入以太坊公链上的合约地址，智能合约虚拟机会根据函数签名和加载的函数参数，在虚拟机内执行编译成 EVM Code的智能合约。



如果涉及区块链数据的读取，则虚拟机会读取区块链上的区块数据。如果虚拟机的指令代码修改以太坊公链的状态，那么通过调用相关的状态机指令，并消耗一定的GAS之后，就可以将修改操作提交到以太坊区块链公链网络中，这些操作往往以交易的方式体现。在虚拟机执行任务结束后，其对区块链状态的修改会被矿工打包。当状态修改被全网共识时，虚拟机对公链网络的状态修改也相应成功。可以通过查询相关的执行结果，将执行状态返回给用户交互系统，这样终端用户就可以通过交互系统查看DAPP操作的执行结果。

## 三．系统需求分析

### 3.1 系统概述

在不违反法律法规的基础上，我们建立了基于区块链的电子投票系统，该系统实现了在保护用户隐私的情况下进行的投票管理和计票功能。实现了所有注册该系统的用户均可自动成为以太坊中的一个节点，并在投票前对自己的投票进行承诺，在投票不受他人干扰的情况下，任何参与和旁观者均可计票的真正透明、公开、匿名的投票功能。

### 3.2 系统业务分析

1. 投票类型及其发起人

已注册meta mask账户的用户进入投票管理界面发起投票选举活动（活动内容上链），并自动成为此投票活动的管理者，需花费一定Gas。

1. 此系统业务内容
2. 投票管理者添加投票成员，可显示投票成员情况
3. 添加完毕后，点击开始投票并在此期间不能再添加投票成员
4. 管理者发送此投票合约地址给投票成员
5. 投票成员进入投票界面开始投票
6. 后台投票信息自动更新并统计票数
7. 投票管理者在指定时间内结束投票，此投票完成

1. 系统需求分析

3.3.1 系统功能性需求分析

1. 用户登录与注册

用户登录和注册功能包括：注册meta mask账户，并登录投票管理界面发起投票。

2. 添加投票的人（仅限该投票发起人添加）

投票发起人必须在meta mask中是发起此投票的账户才能添加投票成员，添加的投票成员必须先注册meta mask账户，拥有账户的人将自己的账户地址发送给投票管理者，由投票管理者添加投票成员并上链。添加完毕后点击开始投票，此后不能再添加成员。

1. 投票成员开始投票

管理者在开始投票后将其发起的投票合约地址发送给各位投票成员，投票成员进入投票界面开始投票并可查询此投票的票数与自己的相关信息。

1. 结束投票及统计票数

管理者在指定时间内结束投票并由系统自动统计票数，后台数据内容可实时显现出哪些成员投票了，与哪些成员没有投票，结束投票后不可进行投票，所有数据皆在测试链上，可查询。

投票流程图如下3.1

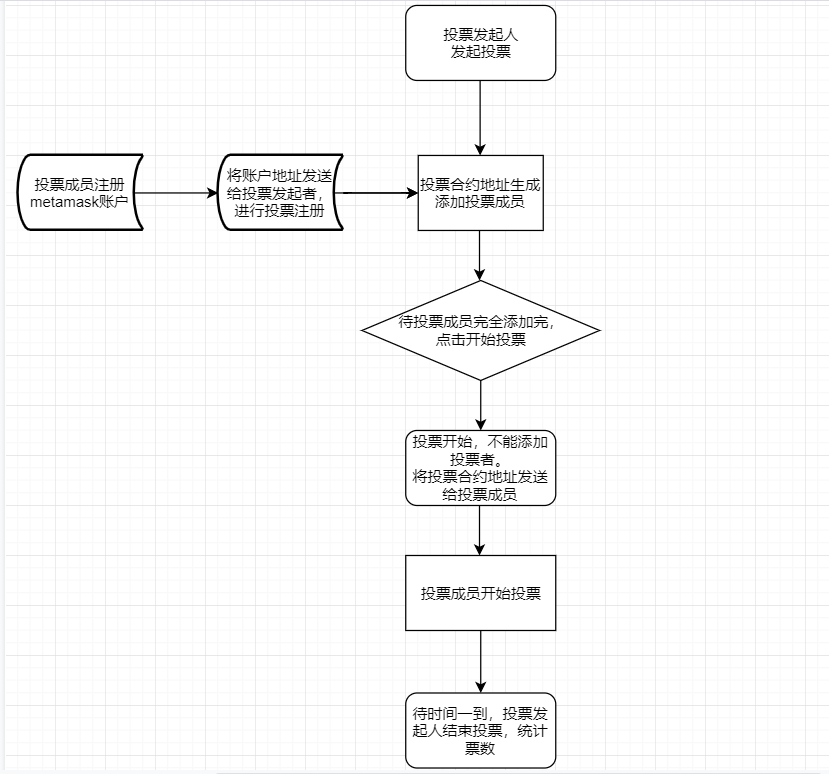


图3.1投票流程图

在用户注册成为投票管理者的同时，以太坊将会从用户对应的meta mask账户中扣除相应的ether，且之后 作为防止用户不进行最终投票的惩罚。此外，以太坊底层的P2P网络层将作为电子投票系统中的广播信道使用。投票人的承诺、 经过加密后的投票以及零知识证明， 都将通过transaction的形式写在区块链上，保证投票的公开透明。

## 系统概要设计

### 4.1 应用建设目标

电子投票与区块链技术相结合能有效解决目前电子投票在使用中存在的一些实际问题，如传统意义的投票系统依然需要可信赖的CA认证机构、无法保证投票的公开透明等等。我们的目标是提出第一个可以实现投票互联网投票协议，最大限度的选民隐私使用BlockChain。与先前提出的BlockChain电子投票协议不同， 这是一个不依赖任何可信权限进行计算的实现。统计或保护选民的隐私。相反，开放投票网络是一个自私的协议，每个投票者都控制着他们自己投票的隐私，这样它才能被破坏。 与所有其他选民合谋。协议的执行是使用协商一致机制强制执行的，协商一致机制还可以确保以太坊块链的安全。

### 4.2 设计原则

1． 继承性：保护已经完成的投票和交易。系统的建设和更新必须保证在尽量不改变原有设计和设备的基础上，保护己完成的交易，同时实现旧系统的更新换代和功能性升级。

2． 安全性：系统的安全性主要取决于硬件安全性，软件安全性和应用安全性，可为系统提供安全可靠的操作环境。

3． 先进性：系统的设计需要采用比较新颖的开发框架和技术，使得本系统可以与不同的操作系统进行数据的对接，未来更好的更新， 并能够更快地适应未来的发展需求。

4． 友好性：系统的操作界面简单实用，用户可以很快速的掌握如何进行投票、发起投票、查看、等系统的常规操作。

5． 稳定性：系统能够对外界的攻击和变化做出应对措施，同时能够解决内部出现的运行错误等问题，保证系统可以平稳可靠地维持日常运行。

### 4.3 总体架构

基于区块链的电子投票应用， 其总体架构图如图4- 1所示，通过系统的设计与实现，对投票信息进行数据上链、汇总和实时跟踪、 形成投票人和投票发起人之间的投票数据交易的实时共享、 史记录查询和投票计票工作，利用区块链技术，对投票和交易进行在线监督、追溯查询和对以太坊后台数据的调取访问等操作，提供给各方合法用户实时有效的投票数据，提升了用户投票的隐私性，减少了对可信赖第三方的依赖程度，最大程度上实现了去中心化、保障了公开、匿名的投票原则。

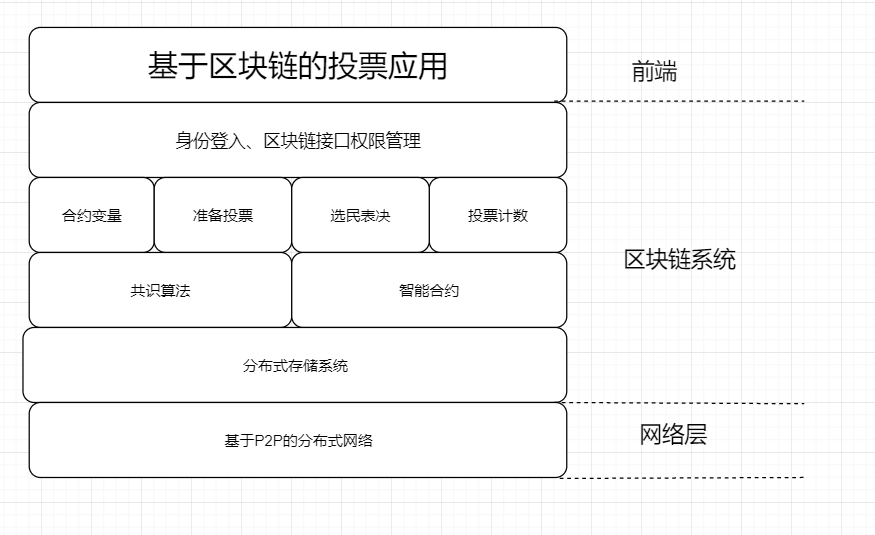
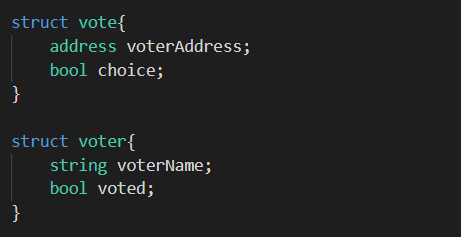


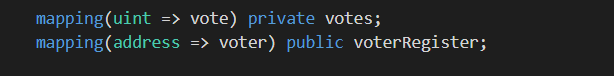
图4-1

### 4.4 智能合约代码解释

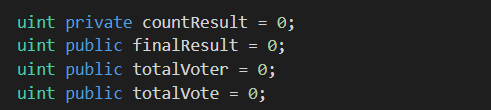
1.合约变量



投票分为两部分，投票者的钱包地址，投票者的选择以及赞成或反对。赞成由值TRUE表示，而反对由值FALSE表示。选民具有3个属性，他的名字以及他是否投票。

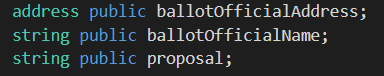


vote存储在名为votes的映射中，而选民voter存储在名为voterRegister的映射中。请注意，投票是私人的，因此选民无法直接阅读，但是voterRegister公开的，因此任何人都可以检查什么是合格的选民。有人会问为什么我不将我的选票和选民存储在数组中。在以太坊区块链中使用数组的挑战在于，当您试图遍历一个相当大的数组的记录时，您可能会开始耗尽Gas。

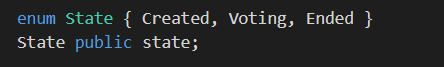


countResult是一个没人看到的私有变量。它告诉您说赞成的总票数。首次初始化合约时，该合约为0。实际上，我撒谎了，在以太坊区块链中，没有任何私有变量是真正私有的。稍加窥探，就可以读取任何变量。countResult 用于跟踪选民投票获得的投票总数。

finalResult将存储最终赞成票数。它保持为0，直到投票结束。投票结束时countResult，finalResult（私有）值移至（公开）值。totalVoter用于跟踪选民登记册中的选民总数，并totalVote用于跟踪所投的总数。跟踪这些数字的变化，例如在进行新的投票时而不是在投票过程结束时进行投票，因为我避免了遍历映射的需要（因为您无法遍历Solidity中的映射）我不希望使用数组，因为极有可能遇到Gas限制问题。

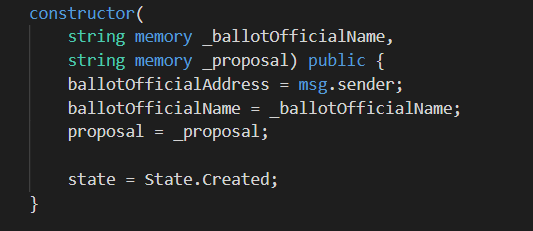


投票管理者的姓名，钱包地址和提案被保留为公共变量，每个人都可以看到，是公开的。

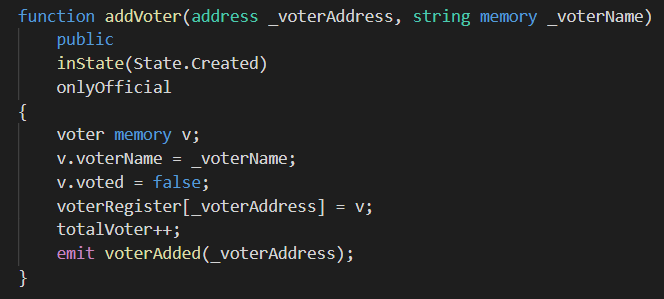


投票合同通过3个字符。投票管理者创建时，其状态为“ Created”。一旦他指示投票开始，投票合同即变为“ Voting”状态。开始计数时，投票状态变为“ Ended”状态。

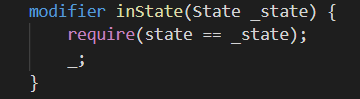
1. 准备选票



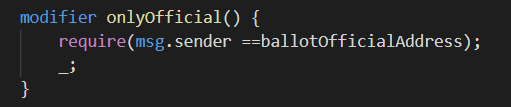
管理者通过提供\_ballotOfficialName和来初始化与构造函数的合同\_proposal。构造函数读取其meta mask钱包地址并更新ballotOfficialAddress。这很重要，因为可能会怀疑是否是合法选票的人们可以将他们所知道的管理者的meta mask钱包地址与此处的地址进行比较。此时，合同的状态为“ Created”。



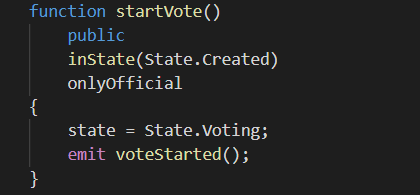
接下来，主席将选民添加到voterRegister映射中。这涉及在映射中输入选民的钱包地址及其姓名。这里指出，只有当合同处于“ Created”状态时此功能才可执行，因此一旦投票开始，就不允许任何人（甚至管理者）都不能添加新的选民。还有onlyOfficial，这意味着只有投票管理者本人可以运行此功能。您不希望选民能够将自己添加到voterRegister映射中吧！



inState的修饰符。它检查以确保合同当前处于提供给inState()修饰符的变量状态.

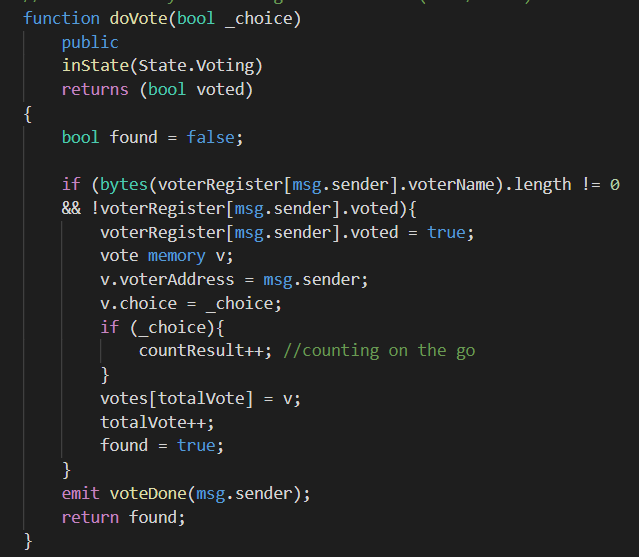


onlyOfficial()声明的修饰符会检查以确保调用此函数的人的钱包地址与中保存的地址相同ballotOfficialAddress。这样可以确保只有管理者本人才可以调用该功能。如果投票发起者拒绝发起该次投票，那么除非有人偷了他的钱包和私人钥匙，否则这将证明他是他本人进行并同意发起。



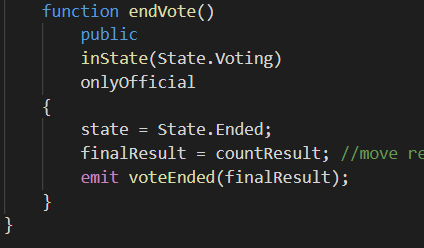
该过程为允许主席开始投票过程。请注意，此功能仅在合同状态为“ Created”时运行，因此管理者（发起者）无法重新启动投票过程。它还会检查onlyOfficial该功能是否可以运行，因此除管理者本人外，其他任何人都无法执行。将合同状态从“ Created”更改为“ Voting”后，进入投票开始阶段，现在可以开始投票了。

1. **投票阶段**



这里调用此功能可让选民投票。如合同所述，选民只有在合同状态为“ Voting”时才能投票。If语句条件声明会检查2个条件：选民只有在选民名册中找到自己的名字而没有选民时才能投票，他的投票已记录到投票映射中，并且他的投票状态已更改为true，因此他无法再次投票。如果他做出“同意”选择，我将加1到countResult。这样一来，我避免了在投票过程结束时遍历所有选票进行计数的方法，因为这样做的内在成本很高，而且如果获得大量选票，合同很可能会耗尽Gas。

1. 结束投票

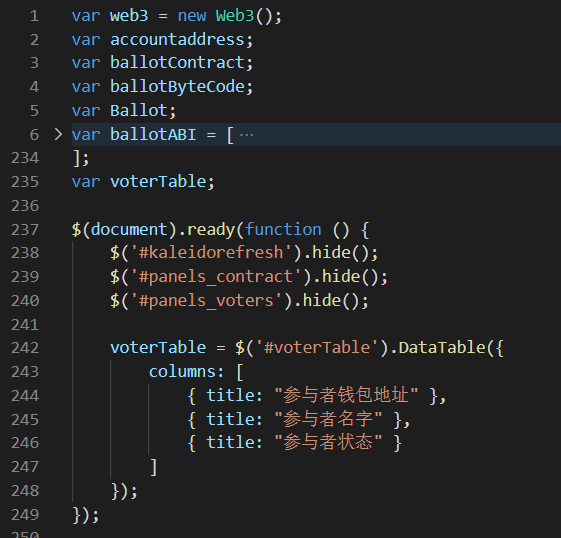


投票发起人运行endVote() 功能结束投票。运行此功能的必需状态为“ Voting”，并onlyOfficial确保只有该投票发起者才能运行此功能。该函数将合同状态更改为“ Ended”，以便投票者不可以继续投票。然后，这个finalResult与countResult（我用来累积选民投票的累计票数的私有变量）的值更新（每个人都可以阅读的公共变量）。最后触发结束投票并统计票数。

## 系统模块介绍

### 1.1 投票管理Dapp模块代码演练

### 1.1.1配置模块



DApp首先设置将在整个投票生命周期中使用的全局变量，即：

（1）accountaddress：这是管理投票过程的发起人的以太坊账户。执行投票智能合约时使用此地址。

（2）ballotContract：这将存储在部署智能投票合同后将生成的智能投票合约的地址。

（3）ballotByteCode：这里是智能投票选票的字节代码为ekpt的位置。稍后，我们将在此变量中填充字节码。

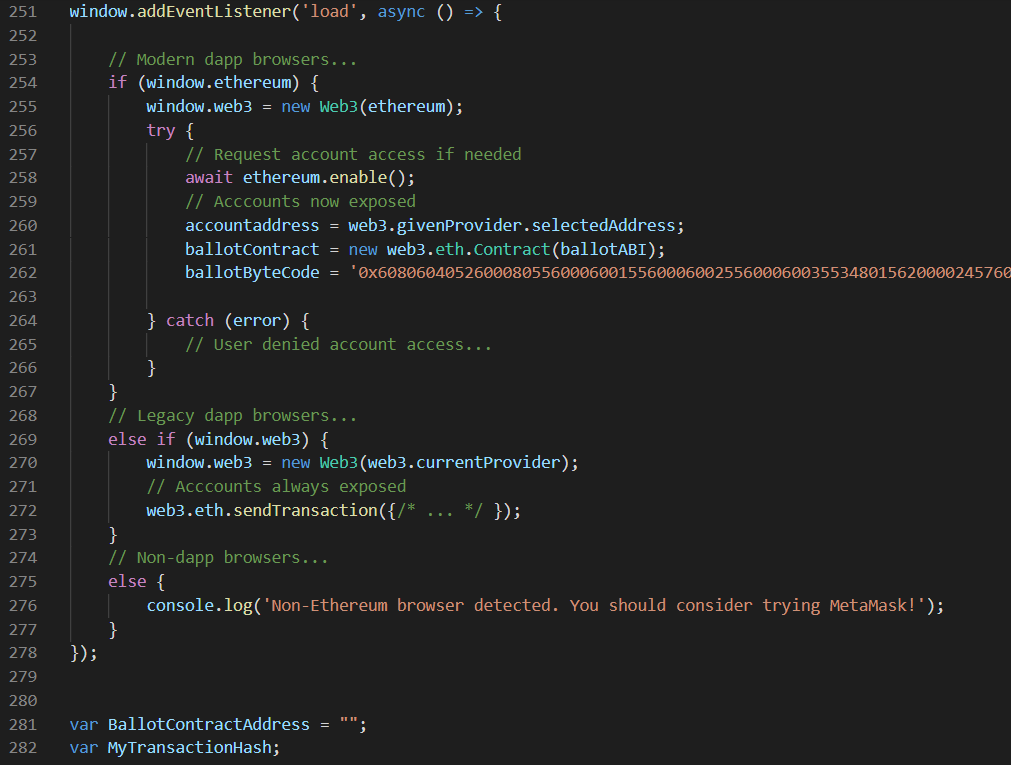
（4）Ballot：此变量存储我们执行和监视的投票对象。

（5）ballotABI：（我在上面的代码段中将其截断了）这是投票智能合约的ABI。

（6）voterTable：这将存储一个数据表，用于跟踪选民，即投票参与者他们的地址和投票状态。

页面加载后，我们将执行以下操作：

1. 隐藏显示选票合同信息的面板，因为刚加载页面时尚未部署选票合同。
2. 隐藏在此阶段也显示投票者的面板。还没有选民。
3. 我还加载数据表，其中显示选民有3列信息，即地址（包含选民的meta mask钱包地址），姓名（这是他们的真实姓名，佐伊·艾莉等）和状态（是否已投票或不）。

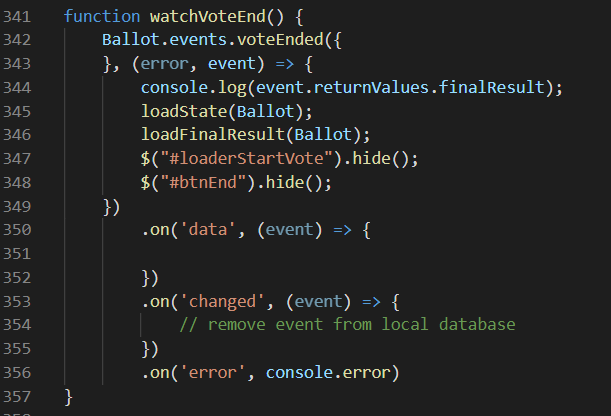


页面加载时，此代码检查浏览器是否随附以太坊钱包（例如MetaMask）。找到后，我们将钱包的活动地址加载到accountaddress 变量中。我们还将选票合同的ABI加载到ballotContract变量中，并将选票合同的字节码加载到ballotByteCode变量中。编译选票智能合约时，将获得选票的字节码。

如果您的用户在没有以太坊钱包插件的浏览器中运行此DApp，它将弹出一条控制台消息，告诉您无法运行DApp。在实时实施中，您可能希望将这些用户重定向到页面以下载数字钱包。

### 1.1.2 监控与等待模块

在DApp的监视部分，监控投票智能合约将在投票生命周期的各个点触发的各种事件。



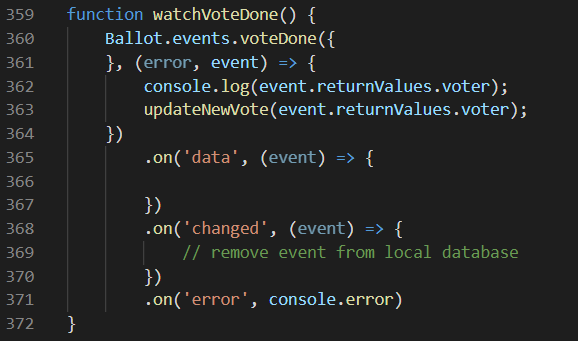
在该watchVoteEnd() 功能中，监控并等待投票智能合约触发voteEnded事件。这表示投票已结束，现在该执行以下操作：

（1）执行loadState()将选票的当前状态更改为“已结束”。

（2）执行loadFinalResult()以显示投票结果。

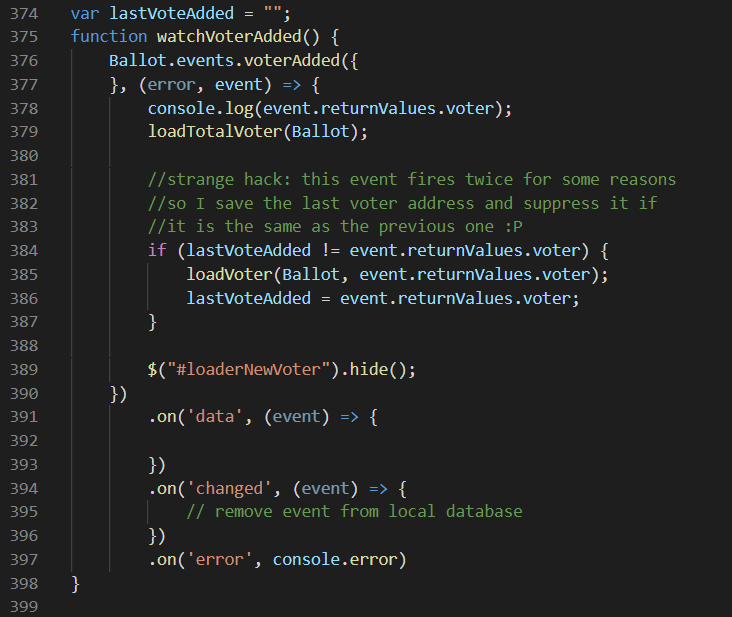
（3）自投票结束以来，隐藏旋转式装载机动画GIF图像，我们不再等待任何内容在后台加载。

（4）自投票结束以来，请隐藏[结束投票]按钮，因此不应允许投票管理者再次尝试再次结束投票。

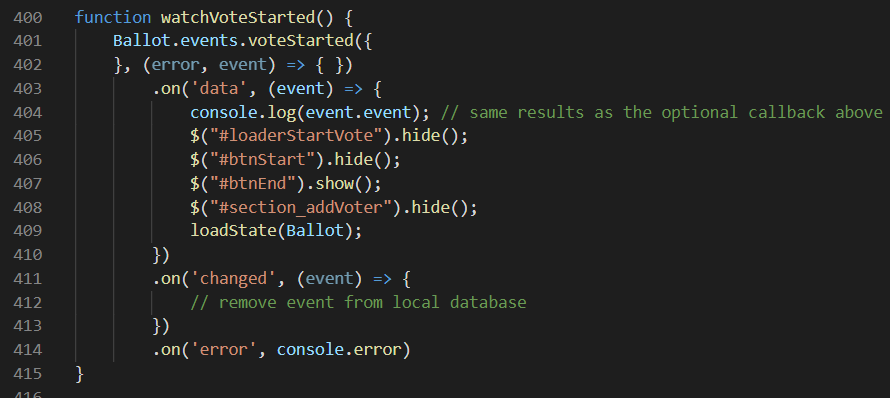


在watchVoteDone()功能中，选民刚刚投票。

因此，我们执行updateNewVote()更新选民表以将其状态从“未投票”更改为“已投票”。



在watchVoterAdded()函数中，我们监控并等待新的投票者被添加到投票者寄存器中。添加他后，我们将停止旋转加载程序GIF图像，然后执行loadTotalVoter()以更新屏幕上显示的投票者总数。



在watchVoteStarted() 功能中，我们监视由投票智能合约触发的事件，以表明投票已经开始。当投票发起者按下[开始投票]时，会发生这种情况。发生这种情况时，我们：

（1）loaderStartVote由于已经开始投票，请停止旋转的GIF图像。

（2）由于您无法重启已经开始的投票过程，因此我们隐藏了[开始投票]按钮。

（3）我们现在显示[结束投票]按钮，因为现在可以结束已经开始的投票过程。

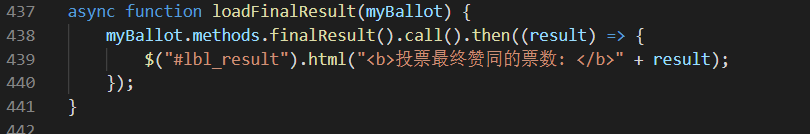
（4）由于无法再将选民添加到已经开始的投票过程中，因此我们隐藏了[添加投票参与者]按钮。

### 1.1.3 更新DApp的用户界面模块

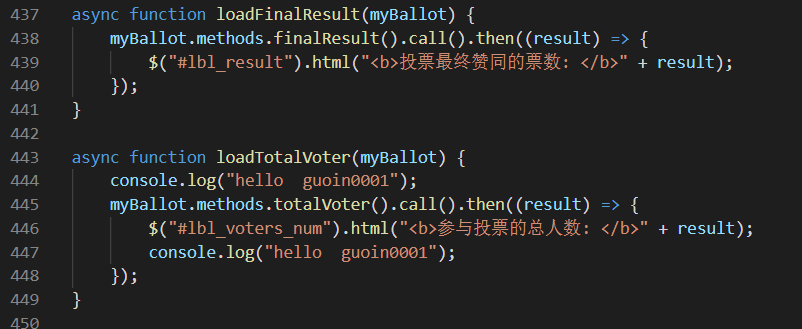
本节中的功能通过从Ballot智能合约中读取相应的值并在其中显示来更新 投票管理者 DApp用户界面的各个部分。



该loadBallotContract() 功能从区块链读取投票管理者的姓名，提案和投票智能合约的地址，并将其显示在DApp UI的相应部分。它还从投票智能合约中加载投票过程的最终结果、投票者总数、投票总数和投票过程的当前状态。

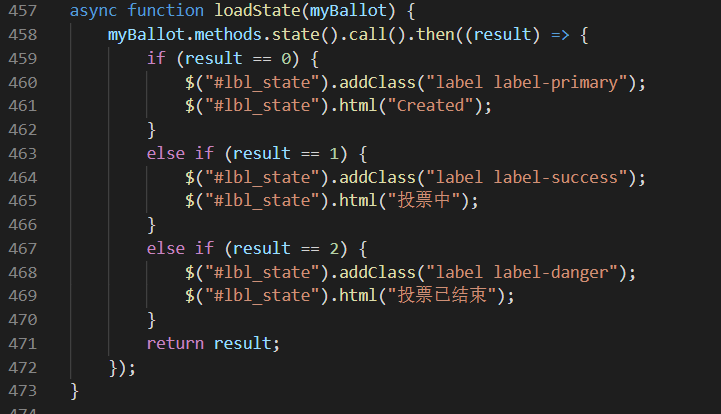


该loadFinalResult()功能从区块链上的投票智能合约读取投票结果，并将其显示在DApp用户界面的相应部分。



loadTotalVoter()功能从区块链上的Ballot Smart Contract中读取选民总数的值，并将其显示在DApp UI的相应部分中。

该loadTotalVotes()功能从区块链上的投票智能合约读取总投票数的值，并将其显示在DApp UI的相应部分中。



该loadState()功能读取区块链上的选票智能合约的当前状态，并将其显示在DApp UI的相应部分中。投票过程具有3种状态：

（1）Created：已设置表决提议且投票合同刚刚初始化时。

（2）投票中：进行投票时

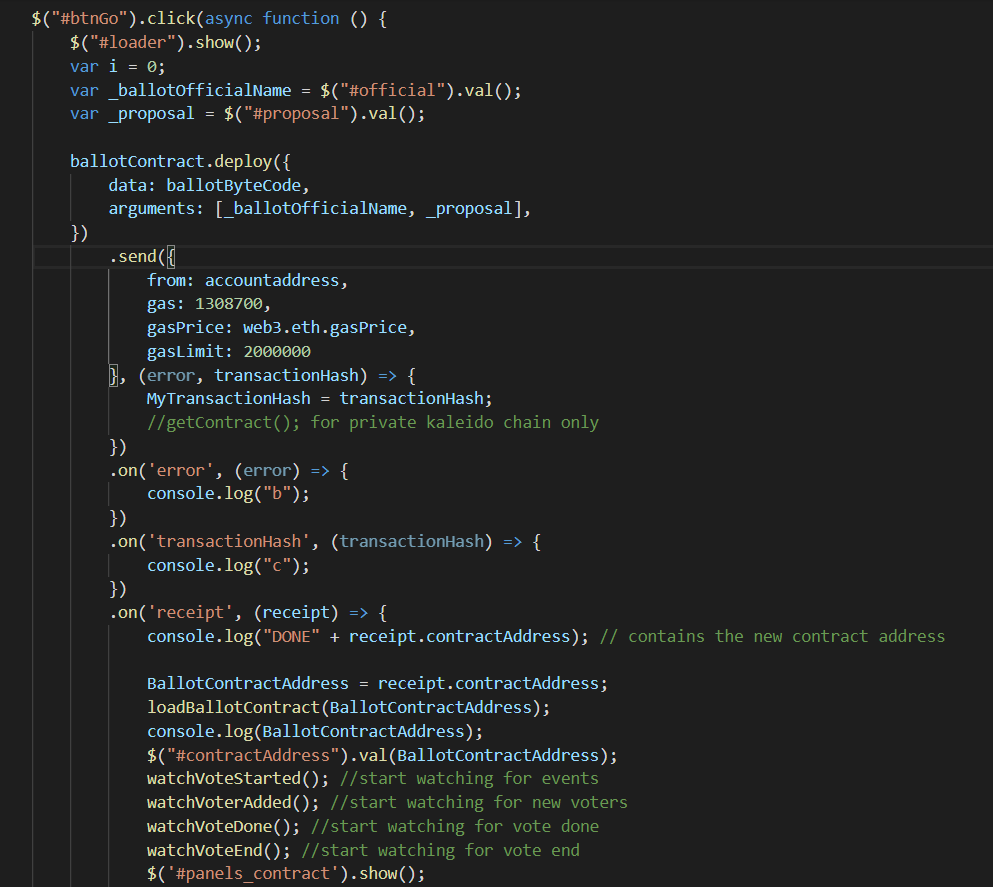
（3）投票已结束：投票结束时



loadVoter()功能从区块链上的投票智能合约中读取投票者的详细信息。它检查该选民是否已投票。如果有，它将其陈述从“未投票”更改为“已完成投票”。它还检查该选民是否是新的。如果是，则将其添加到投票登记表中。

updateNewVote() 功能从区块链上的投票智能合约中读取投票人的投票状态，并在DApp UI的相应部分中将其状态更改为“已投票”

### 1.1.4 开始新的投票 模块



投票管理者开始新的投票过程时，按下[Go]按钮。它触发将在区块链上部署新的投票智能合约。

成功部署后，它将BallotContracAddress使用刚刚部署在区块链上的新投票智能合约的地址填充该字段。它执行loadBalloContract()以在DApp的UI上更新选票的正式名称和提案。然后，它监视投票合同开始投票时触发的新事件。

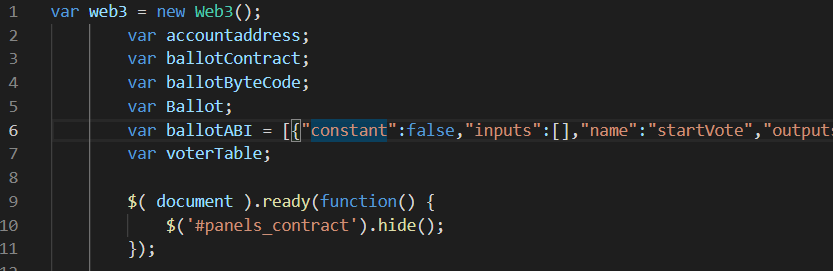
它使合同面板可见，以便用户可以查看合同详细信息。它使投票者面板可见，以便主席可以添加新的投票者。它还显示[开始投票]按钮，以便主席可以按下它来宣布投票已经开始。

它禁用[结束投票]按钮，因为如果不先启动投票过程就无法结束投票过程。

它还显示在选票簿中显示选民的部分，以便主席可以开始将选民添加到投票智能合约中。

### 1.2 参与者投票Dapp模块代码演练

### 1.2.1 配置模块



DApp首先设置将在整个投票生命周期中使用的全局变量，即：

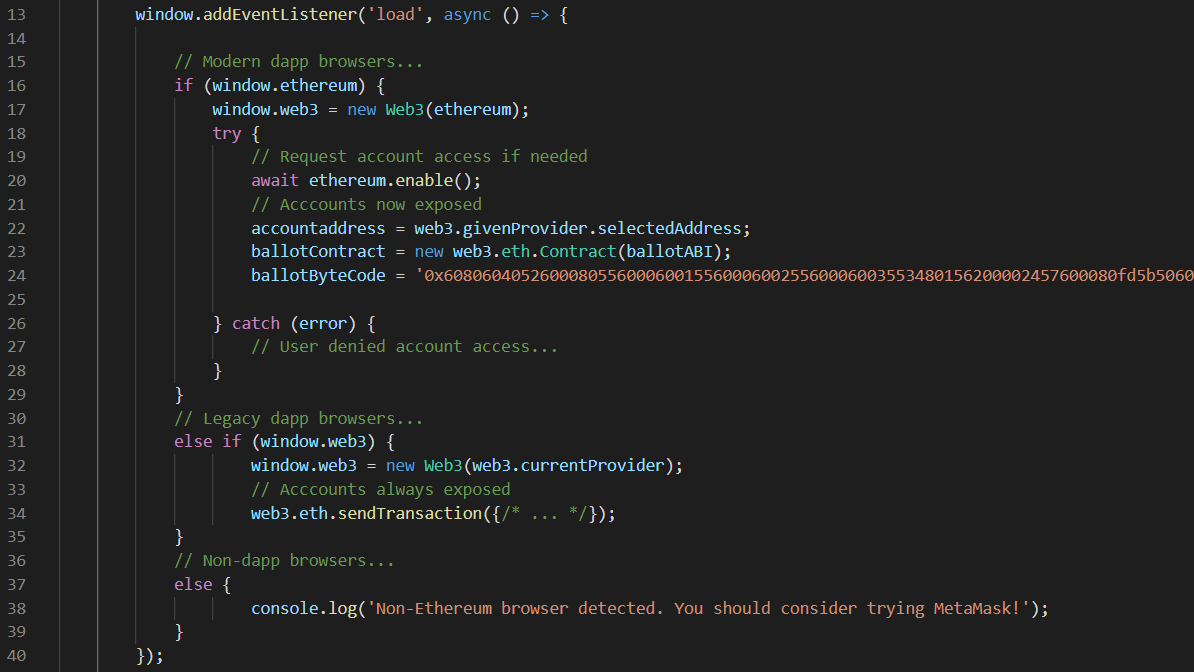
（1）accountaddress：这是管理投票过程的主席的以太坊账户。执行Ballot智能合约时使用此地址。

（2）ballotContract：这存储了选民将提供的选票智能合约的地址。主席在开始投票程序后，将向选民提供Ballot Smart Contract的地址。

（3）Ballot：此变量存储我们执行和监视的投票对象。

（4）ballotABI：（我在上面的代码段中将其截断了）这是Ballot智能合约的ABI。

加载页面时，我一直panels\_contract处于隐藏状态，因为既然此时选民还没有提供他的钱包地址，那么有投票权的DApp不必告诉他提案是什么。



页面加载时，此代码检查浏览器是否随附以太坊钱包（例如MetaMask）。找到后，我们将钱包的地址加载到accountaddress 变量中。我们还将选票合同的ABI加载到ballotContract变量中。

如果您的用户在没有以太坊钱包插件的浏览器中运行此DApp，则它将在浏览器的控制台中弹出一条消息，告知您无法运行DApp。在实时实施中，您可能希望将这些用户重定向到页面以下载数字钱包。

### 1.2.2 显示提案模块

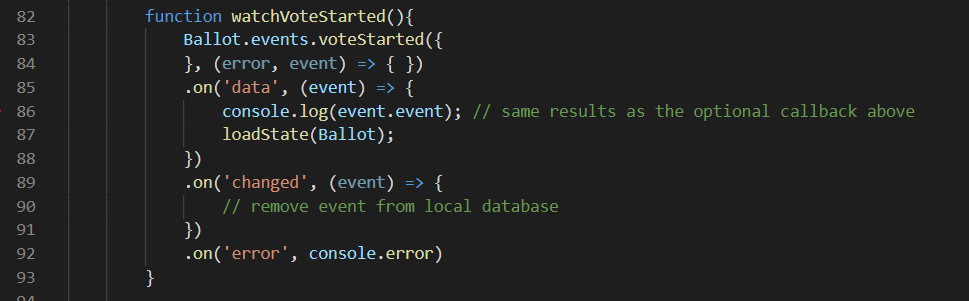


在getContract当选民提供投票合同的地址，并按下[GO]按钮功能被调用。在此，将载入选票合同，并显示选票的正式名称和提案详细信息。

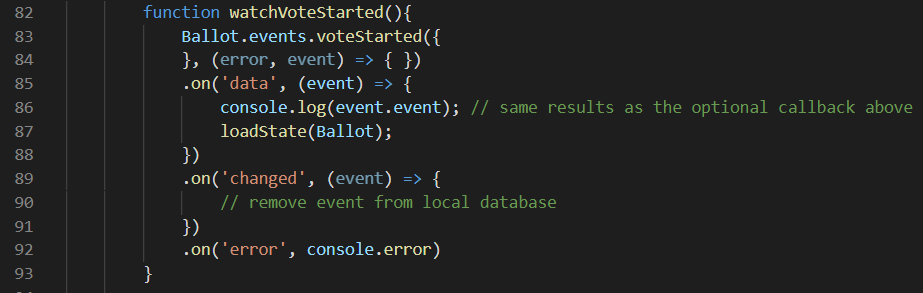
它执行loadFinalResult()以显示投票结果（如果投票完成）。它还执行loadState()以显示投票过程的当前状态-处于“Created”，watchVoteStarted()，watchVoteEnd()表示“投票中”或“已完成投票”的状态。

而且watchVoteDone() 功能会等待并观察投票智能合约引发这些事件。这些事件在投票过程中的各个时间点触发。

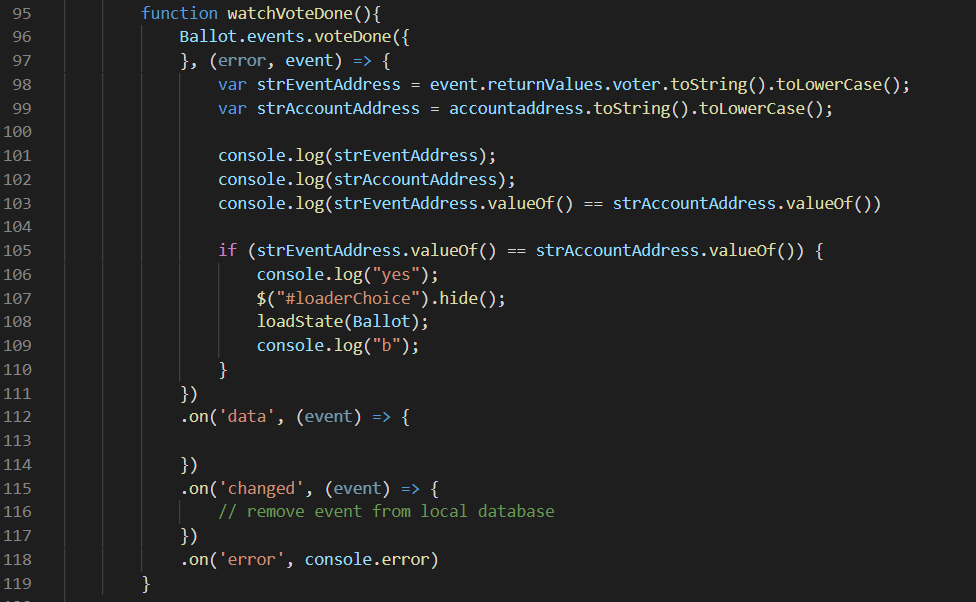
### 1.2.3 监控模块



watchVoteEnd()等待投票智能合约触发voteEnded事件。发生这种情况时，表示投票管理者已结束投票过程。loadState()执行以在用户界面中显示一条消息，以告知投票者投票已结束。loadFinalResult()执行以向选民展示结果。

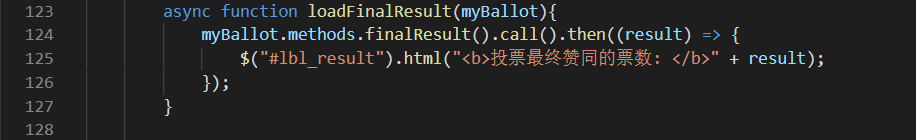


watchVoteStarted() 等待投票智能合约触发voteStarted事件。发生这种情况时，表示投票管理者已开始投票过程。loadState() 执行以在用户界面中显示一条消息，以告知投票者投票已经开始。

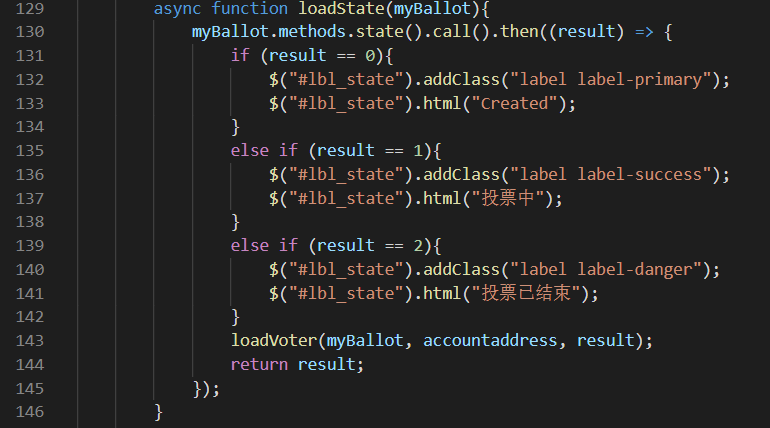


watchVoteDone()等待voteDone事件被触发。voteDone当选民投票被触发。由于voteDone每次选民中的任何选民进行投票时voteDone都会触发该代码，因此代码将检查此事件是否由该选民本人触发，而不是由恰好同时进行投票的另一个选民触发。然后，它将重新加载选票的状态，loadState()以隐藏[赞同]和[反对]按钮。

### 1.2.4 投票信息模块



loadFinalResult()从投票智能合约中获得最终投票结果，并将其显示在投票 DApp的用户界面中。



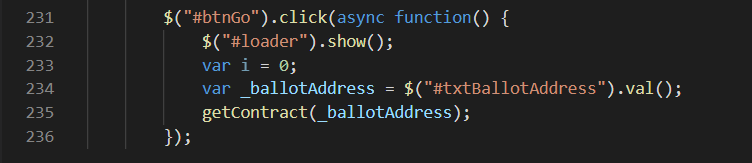
loadState() 查找投票智能合约的当前状态，并在投票DApp的用户界面中显示“Created”，“正在投票中”或“投票已结束”。



loadVoter() 检查选票智能合约上的选民登记册，以查看该选民的钱包地址是否已注册为选民。如果他不是，那么他就没有资格投票。[赞同]和[反对]按钮被隐藏。

它还会检查“投票智能合约”的当前状态是否允许投票。如果是这样，则打开[赞同]和[反对]按钮供选民按下。

最后，它检查选民先前是否已经投票。如果他有，则[赞同]和[反对]按钮将被隐藏，并且投票仅看到“已完成投票”通知。



用户按下[GO]按钮以加载Ballot智能合约。它执行getContract()以使选民看到提案并对其进行投票。

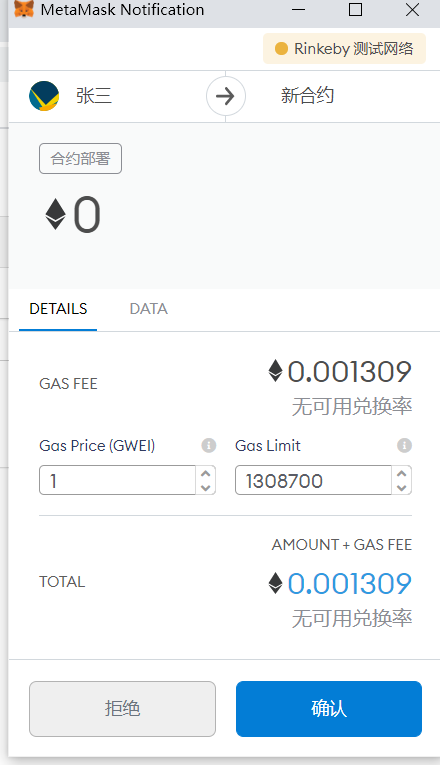
## 投票 Dapp 功能演示

### 现在访问投票管理器。

让您的投票正确的名字（默认情况下我们把张三当作投票发起人）和提案内容（默认情况下，我们问：李四是否应该当选班长）。按“发起投票-Go”继续。



您将需要确认此合同部署事务。它需要付费，但是由于我们在Rinkeby上，所以它是免费的。

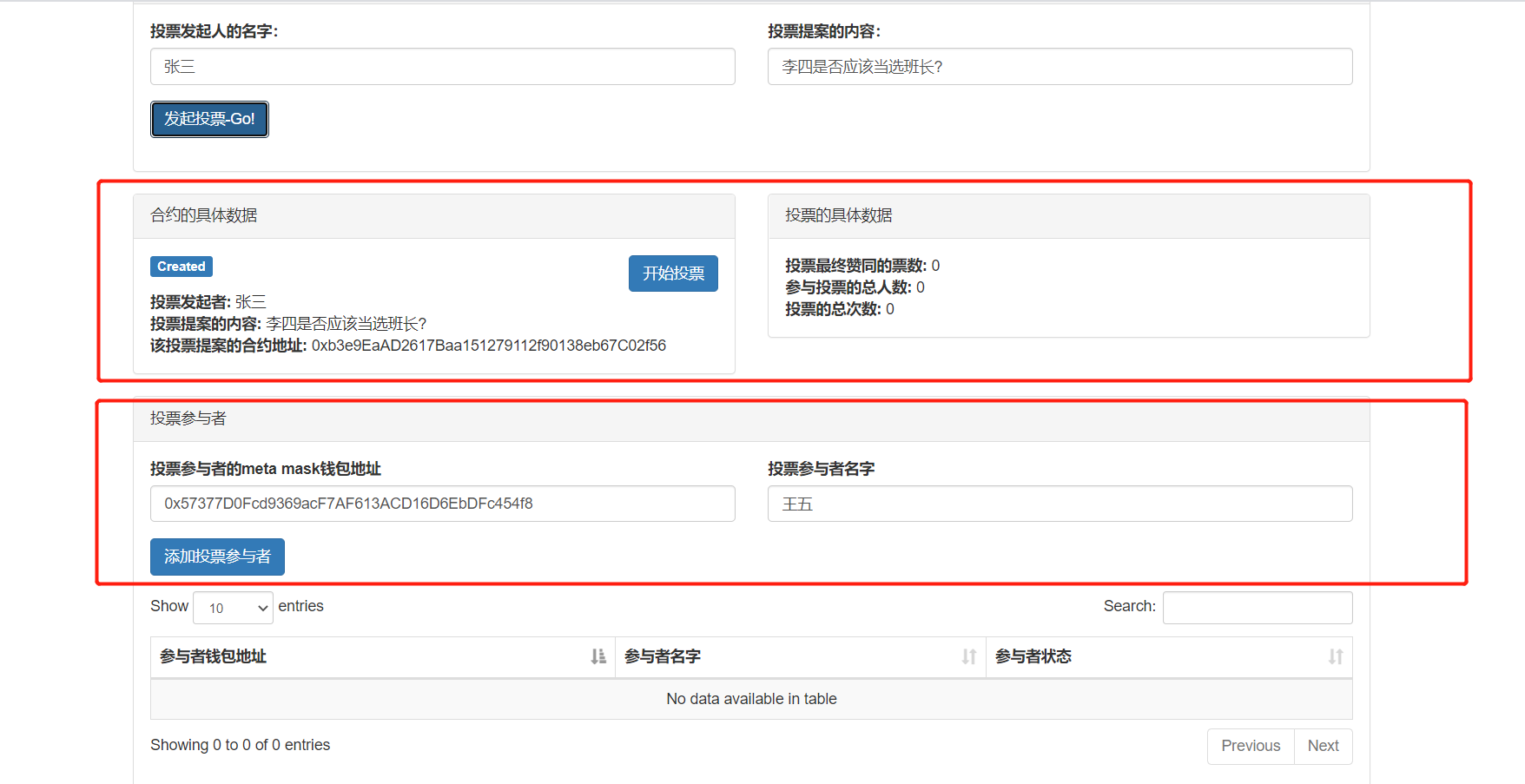


稍等一下 投票合同最终被部署到以太坊区块链网络上。请注意出现的3个部分：

（1）合约详细信息：您可以在此处查看此投票合同的详细信息，其中包括投票管理员的姓名，提案以及刚刚部署了该合约的区块链上的地址。

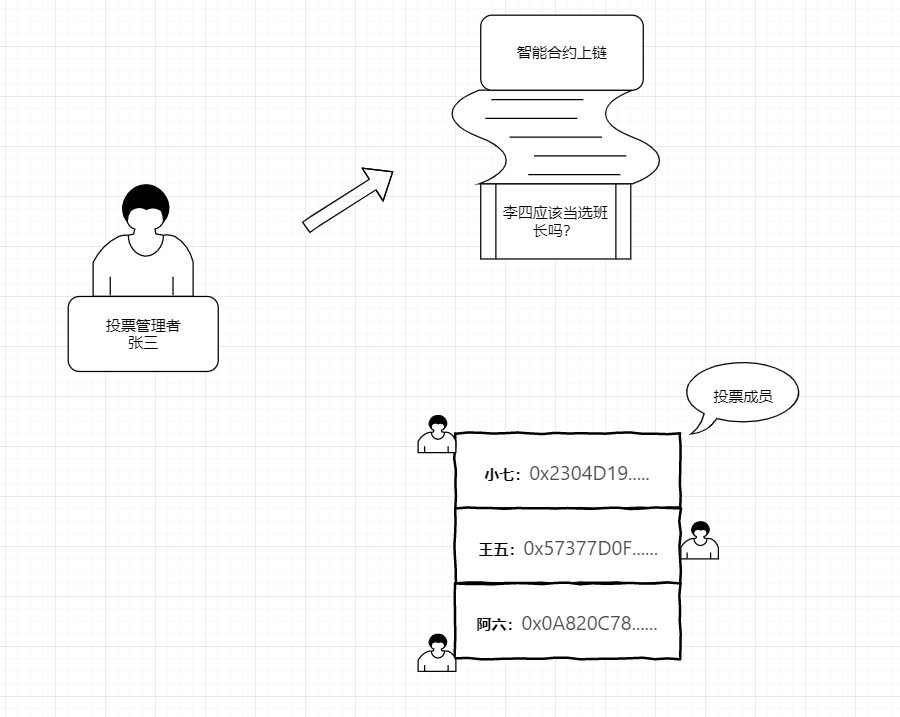
（2）投票详细信息：您可以在此处检查投票结果，选民总数和投票总数。

（3）添加选民：作为管理者，您可以在此部分添加新的选民。



### 决定谁可以投票

作为该投票的管理者，您现在需要确定谁是选民。您将已经收集了您的3个投票者小七，王五和阿六的钱包地址。



现在输入他们的姓名和钱包地址。然后按[添加投票参与者]进行添加。这是给这王五的。



接下来添加阿六、小七的。注意：他们都是未投票的状态。



投票成员添加完后，管理者点击“开始投票”按钮。



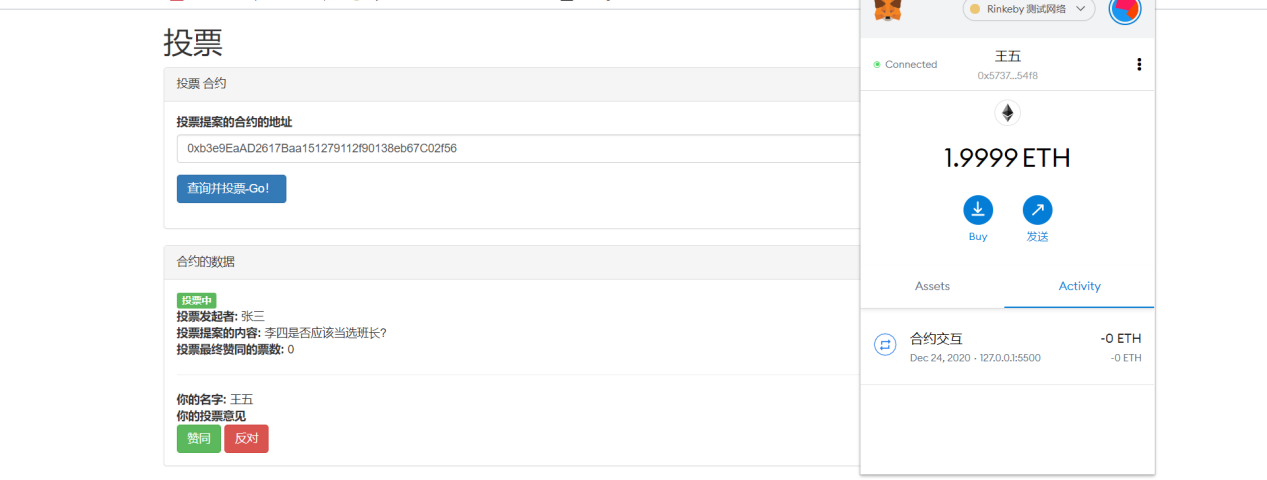
之后就不能再进行添加投票成员，且允许已被添加的投票成员进行投票。管理者界面为“投票中”状态。



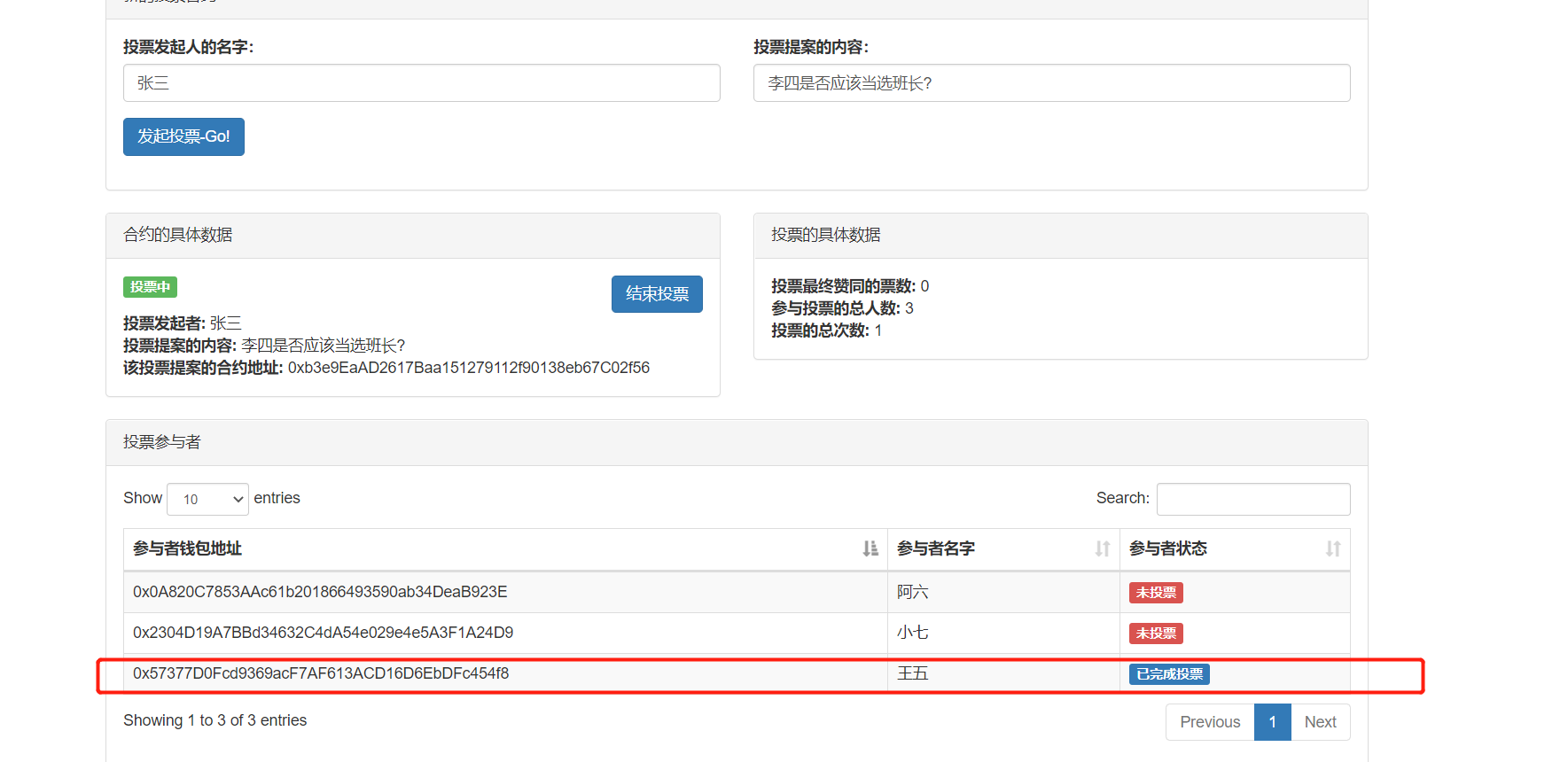
### 选民投票

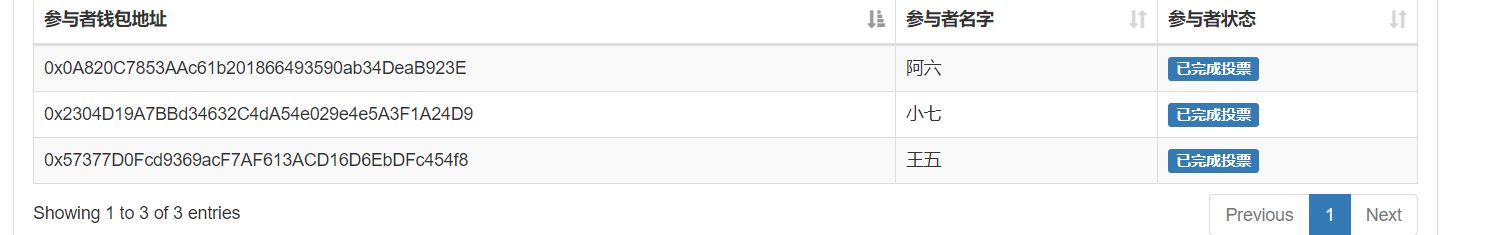
更改您的MetaMask钱包以使用王五的钱包地址。访问“投票”页面。粘贴管理者部署此合同时生成的投票地址。然后按[查询并投票]。这将在“合同详细信息”部分中检索投票的详细信息。

您现在可以投票[赞同]或[反对]。



如果切换回“投票管理者的选票管理器”屏幕，您会注意到王五现在已经处于“已完成投票”状态。



之后阿六，小七陆续投票。可看到三者已完成投票。

### 结束投票并取得结果



“结束投票”只有管理者才能结束。更改您的MetaMask帐户以指向管理者的钱包。切换回管理者的投票管理器屏幕，然后按[结束投票]结束该投票过程。最后出现如上图结果。

显示结果。在我的情况下，每3个人中有2个人投票“是”，因此结果说“ 2”。

请注意，“投票的具体数据”部分还显示投票者总数和投票总次数。

## 总结局限性与改进

这样，一个完整的投票就结束了。

但是具有几个致命的局限性。这是关于它如何做得更好的讨论。

想象一下，这是一次有1000万选民的全国大选。遍历投票记录肯定会超出在以太坊上执行智能合约的上限。智能合约一次只运行一行代码。在别人投票之前等待他人投票的等待时间会非常缓慢。单一的合约是单一的失败点。如果这在投票过程中中断，那么将进行整个选举。而且我确信在负载测试期间我们还会遇到更多问题。

作为改进，我提议将来的实施将包括一个主投票合同，该合约产生多个子合约，每个子合约包含相同的提议，但最多注册为n个（其中n可以是某个特定人群中的总人数）地区）选民。

注册的选民将收到其个性化的投票合约以进行投票。可以根据地理位置，注册时间或任意地完成此分配。

投票结束时，每个投票合约都将跟踪自己的选票并将其结果报告给主选票合约。然后，总投票合同将合并并报告最终结果。

这种方法模仿了全国性选举的方式，在这种选举中，选区自己计算选票，而最终选票则在全国范围内进行盘点。

## 参考文献

1. 正版区块链工程实践：行业解决方案与关键技术 (区块链技术丛书)
2. 区块链DAPP-开发入门,代码实现,场景应用
3. 区块链白皮书2019
4. 正版区块链以太坊DApp开发实战
5. 精通比特币Mastering-Bitcoin
6. 精通以太坊：开发智能合约和去中心化应用 (O’Reilly精品图书系列)
7. JavaScript权威指南（原书第6版）
8. <https://medium.com/coinmonks/voting-on-a-blockchain-solidity-contract-codes-explained-c677996d94f2>
9. <https://baike.baidu.com/item/%E5%8C%BA%E5%9D%97%E9%93%BE/13465666?fr=aladdin>
10. <http://www.doc88.com/p-1436120443255.html>
11. Thereum.The mix ethereum dapp development tool．https:// github.com/ethereum/mix
12. AITZHAN N Z，SVETINOVIC D.Security and privacy in decentralized energy trading through multi-signatures，blockchain and anonymous messaging streams[J].IEEE Transactions on Dependable and Secure Computing，2018，15（5）：840-852.