东北大学秦皇岛分校

Northeastern University at Qinhuangdao

毕业论文

基于区块链技术的股权众筹平台研究与设计

院　　别	计算机与通信工程学院
专业名称	电子信息工程
班级学号	**2151202**
学生姓名	曹宇聪
指导教师	陈海宴

2019 年 5 月 25 日

基于区块链技术的股权众筹平台研究与设计

摘　　要

随着区块链技术的发展，“互联网+金融”的发展方向逐步迎来改变，这使得区块链技术落地到股权众筹领域成为可能。股权众筹就是通过多个人的融资来开展项目，但目前大多数的众筹平台存在中心化数据库不可靠，数据造假严重的问题。本文设计的股权众筹平台是基于区块链技术的，依靠区块链技术的去中心化，不可篡改以及交易记录可追溯的特点，解决了第三方信任问题。

本论文首先设计了基于智能合约和区块链技术的股权众筹平台框架。在设计过程中，充分利用系统分层而治的思想，将股权众筹系统划分为三层架构：业务层，接口层，数据层。业务层负责的是系统前端与用户的交互功能，包括用户模块、项目创建模块、投资模块等，其中用户模块主要完成用户注册与登录等功能。数据层负责的是对数据传输的统一预处理和封装。接口层负责底层区块链网络的通信和平台数据库信息通信。本文搭建了实验环境并完成了股权众筹系统的编写与部署：使用 Apache 作为网站的服务器，使用 MySQL 作为平台的数据存储介质，网站后台逻辑采用 PHP 语言编写，并结合一些前端框架构造简洁且体验良好的网站界面。在区块链网络中，本文使用的是以太坊提供的开发框架 Truffle 和 Ganache，开发了一份智能合约并上线，使用模拟账户进行交易。最后对整个框架进行测试，确保各个功能的正常使用。

关键词：区块链，众筹平台，智能合约，去中心化

Research and Design of Equity Crowdfunding Platform Based on Blockchain Technology

Abstract

With the development of blockchain technology, the development direction of “Internet + Finance” has gradually changed, which makes it possible for blockchain technology to fall into the field of equity crowdfunding. Equity crowdfunding is to carry out projects through the financing of multiple people, but most of the current crowdfunding platforms have problems of unreliable centralized databases and serious data fraud. The equity crowdfunding platform designed in this paper is based on blockchain technology. It relies on the decentralization of blockchain technology, the inability to tamper with and the traceability of transaction records, and solves the third-party payment problem.

This paper first designs a framework of equity crowdfunding platform based on smart contracts and blockchain technology. In the design process, the idea of this system stratification and governance is fully utilized to divide the equity crowdfunding system into three layers: business layer, interface layer and data layer. The business layer is responsible for the interaction between the system front end and the user, including user modules, project creation modules, and investment modules. The user modules mainly perform functions such as user registration and login. The data layer is responsible for the uniform pre-processing and encapsulation of data transmission. The interface layer is responsible for communication of the underlying blockchain network and platform database information communication. This paper builds the experimental environment and completes the preparation and deployment of the equity crowdfunding system: using Apache as the server of the website, using MySQL as the data storage medium of the platform, the background logic of the website is written in PHP language, and combined with some front-end frameworks to construct and experience Good website interface. In the blockchain network, this article uses the development framework provided by Ethereum Truffle and Ganache, developed a smart contract and went online, using the demo account for trading. Finally, the entire framework is tested to ensure the proper use of

each function.

Key Words: Blockchain, Crowdfunding Platform, Decentralization, Smart Contract

目　　录

1 绪论

1.1 研究背景

随着互联网技术的蓬勃发展，大数据、云计算等衍生技术逐渐与传统各行各业相融合，毫无疑问的是信息时代已经不可避免地来临。当金融业结合互联网技术后，“互联网+金融”模式势必打破了传统的投融资模式[1]，并随着该行业逐步兴起。

在使用各种金融产品时，必须签署某种基于人与人之间贷款的合约[2]。将有一个中央机构，它将通过建立信用评级系统支持的衍生品提供可靠的信贷担保。在金融领域，信任无疑是最重要的问题，但目前的金融网络市场仍处于健康状况不佳的状态。此外，中央集权组织领导的统一记账机制面临着容易丢失数据和不透明交易等问题；而适当的分布式记账机制可以提供高可靠性的交易环境。本课题为了寻找一种有效的方法来解决这些问题，建立了一个点对点对等的信任网络，各个节点通过自主编程创造合同，并结合数字加密机制和签名来实现分布式交易网络。此外，分散的记账系统[3]降低了建立信任评级系统的成本，这对互联网时代的金融产品的发展至关重要。下表 1.1 我们将中心化平台与去中心化平台进行了对比。

表 1.1 去中心化平台与中心化平台对比

	中心化系统	去中心化系统
维护成本	中	低
安全	高	极高
可靠	高	极高
交易成本	中	低

大多数数据，如传统众筹平台的资金，都存储在外部集中数据库中。数据不是公开和透明的，同时维护成本很高。本课题提供的新的众筹平台不仅安全可靠，而且众筹平台的维护成本和用户项目的交易成本都会有大幅下降。在传统的众筹中，投资者必须确认他们的投资资金是否进入了项目创建者的账户，项目创建者还必须确保资金已经成功交付。作为项目中介，股权众筹平台只是二者之间的桥梁，负责他们的沟通和互动，不承担任何商业或法律责任的风险，而这样的中间人就可能导致数据的不真实[29]。为了实

现产品生产自动化和执行金融合同自动化，本课题应用的是软件工程方法理论，当合同执行业务逻辑满足条件时，自动执行合同。

鉴于现存众筹平台现存各种问题根源在于其本质是一个中心化的平台，本文提出一种基于区块链技术和智能合约的股权众筹平台系统。使用以太坊平台(Ethereum)提供的Solidity 智能合约语言和底层的区块链技术构建一个去中心化、去信任的点对点分布式交易系统。

1.2 研究意义

随着近年来比特币又迎来一个波段，区块链技术作为比特币的底层技术开始受到计算机行业程序开发人员和其他各行业人员的追捧，在此之后分布式记账本(Distributed Ledger)技术掀起了新一波的浪潮。区块链具有的特点如下：分布式的网络鲁棒很好，可以使部分节点长时间处于一种异常状态；已发生的交易记录具有不可篡改性，节点达成共识并已经提交的区块不可被删除或者重新写入；可以很好的保护隐私，区块链中使用了大量密码学知识，这保证了数据安全。区块链技术具有的典型业务特性包括：可信任性，它可以提供高度可信的点对点交易平台，不需要第三方中介权威机构从中做背书；成本低廉，相比于传统的金融交易操作[5]，区块链技术能够通过执行智能合约来处理更复杂的交易；安全有保障，区块链技术有利于安全、有保障地审计和管理账目，以及进行清算。

区块链技术并不是像外界夸大的革命性的创新技术，其实更多的是传统计算机的多种技术融合并增添新的个性化理念而产生的新生代产品。其次，基于区块链技术的商业应用也一步步促进其本身的发展。金融机构和科技公司可以尝试利用区块链技术的诸多特性来构建去中心化解决方案或者改变原来的系统运作模式。

1.3 国内外研究现状

世界上第一个众筹项目是在2001年启动的,这个项目是一个关于音乐的众筹项目，它面向的是热爱音乐的艺术家以及粉丝，可以说它也是众筹模式的鼻祖。从此以后众筹模式走向普罗大众，虽然上文提及到的音乐项目在一开始很小，但由于其新颖的模式、低门槛，类似的众筹模式依旧受到越来越多的人追捧。

目前，众筹模式分为四种：捐赠模式众筹、奖励模式众筹、股权众筹和债券众筹。

股权众筹模式已经是比较普遍的众筹模式了，投资者对项目或者公司机构进行资金支持，并获得一定的股权回报。而债券众筹就类似于 p2p 借贷，投资者对项目方进行投资，到一定的时间期限后，项目方将本金和利息一并还给投资方。

发达国家，例如西欧诸国以及北美美国加拿大等发达国家，对股权众筹的法律法规和行业发展规划治理比国内要完善。与此同时，它们还可以刺激 PE 和 VC 以外的一些大型投资银行、金融组织，并创建众筹网站。这样的第三方平台可以达到相对有利的开发空间。众筹起源于美国，在欧洲和美国的短期内继续迅速发展。据统计，截至 2018 年，美国已经创建了 8000 多万次众筹项目，参与达 350 万人次，目标募集率为 180%。从技术上讲，在国外，尤其是美国，拥有世界上最完整的金融市场，治理和规模越来越成熟，法律和监管部门也在逐步改善。在这种环境下，ICO (Initial Crypto-Token Offering)是发展区块链项目作为新的商业模式的重要融资渠道之一。由于区块链金融在全球范围内仍然处于野蛮状态，学术界没有系统性的研究方向，然而与区块链相关的技术知识是非常广泛的，不仅要理解区块链的技术原则，还要理解债券发行和众筹经济等相关理论。

目前，许多金融机构和一些金融技术公司已经建立了联盟[6]，共同推进区块链技术的研究和发展，并有高质量的区块链技术应用项目。代表机构包括 R3、DAO(分散的自治组织)、后贸易联盟等，这些集团联盟的建立不仅可以加速对区块链技术的研究，还可以为区块链产业的发展制定标准。

众筹股权对中国大陆来说仍然是一种新的在线的融资方式。近年来，许多公司看到它的前景，这些公司改进筹资模式，迅速开发出某些站点，但同时暴露出一些问题，想要结合互联网和金，需要各部门适当且妥善支持。目前该行业暴露出的问题既要考虑到股权众筹这种经济模式本身，同时也要考虑到股权众筹平台这个第三方中介机构以及整个互联网时代这个大环境所面临的问题[7]。

著名的比特币平台就是公有区块链，所以可以说公有区块链最具权威性。联合区块链也叫联盟链，是由几个公司或者组织共同来维护的一条数据区块链。在这个区块链中，由该群体内部指定的多个预选节点作为记账人，每一个区块链的生成由所有预选节点共同决定，其他的接入节点可以参与交易，但不会去关注记账的过程，其他任何节点都可以通过该区块链开发的 API 进行数据的交易。记账人的多少，以及如何决定这个记账人是问题的关键所在。私有区块链是由一个公司内部自己使用的记账技术，当然也可以是

个人，其独享该区块链的写入权限。

德勒利用分布式账本高透明度、低信任成本开发的 Rubix 平台使购销过程更透明可靠，审计和检查更实时高效。 Linq 在 2015 年 实现了基于区块链的股权交易平台，该平台解决了股权交易交易不透明、信息不对称的问题。overstack 提出了利用区块链技术发行金融证券的方法，降低了其发行成本，并为其提供数字化所有权证明。 智能合约(Smart Contract)最早在 1994 年由 Nick Szabo 提出，他将其描述为“智能合约是执行合同条款的计算机化交易协议。在满足合同目标的情况下，对信用中介结构的需要最小，合约的执行成本与交易成本最小”。金融产品的创新，使得对于设计支持结构化的金融产品的软件系统提出了挑战，这为智能合约的发展带了应用场景。

赵赫博士使用区块链技术来防止黑客攻击和高安全数据加密，并提供了一种方法来保证抽样数据的真正可靠性。

1.3 论文研究内容

本文主要研究基于现有的国家政策、法律和法规，利用现有的众筹平台的优势，并通过研究其特征，为投资者和企业家创建了一个数据平台。

在使用智能合约和区块链技术构建去中心化的股权众筹平台，实现公正可信的投资融资新模式。设计智能合约的基本思路是以 Ethereum 作为底层区块链网络，设计智能合约逻辑框架，并编写 Solidity 智能合约特定领域语言文件，最后建立了基于 Web 的系统前端平台与后端数据接口之间的映射。智能合约可以被理解为现代化的契约，受益于数字货币的发展和区块链技术的迭代更新，后者为智能合约提供了一个可以信任的执行环境，这使数字货币的交易变得更加安全，契约的执行变得更加公平。而区块链技术最高光的特性就是去中心化，最典型的应用场景就是数字货币交易，所以区块链成为设计智能合约应用、构建合约执行系统的前提条件。在股权众筹领域，还没有系统化的实现基于区块链技术与智能合约的平台，本文提出的解决方案是从系统的前端展示到后端数据传输再到区块链技术的统一，设计并实现了一个完整的股权众筹平台。

1.4 本文的结构安排

第一部分 绪论。详细阐述了基于区块链技术股权众筹平台的研究背景、意义以及国内外对本系统的研究现状。

第二部分 对关键技术介绍。研究相关的关键技术 Etherum 和 Solidty。

第三部分 系统分析。确定系统的主要功能，并对需求进行分析，对每个模块的功能进行分析和设计，分析数据实体之间的关系，对数据表进行设计。

第四部分 系统实现与测试。对功能模块、系统进行详细代码逻辑编写，对显示界面进行简要设计。按照测试样例进行系统检测。

第五部分 对论文进行总结、致谢以及参考文献。

1.5 本章小结

本章首先对研究背景进行描述，然后对本课题的选取进行叙述，紧接着介绍国内外现状，然后对股权众筹系统做简要介绍，分析本项目的作用关系。最后讨论论文的组织结构。

2 区块链与智能合约的理论研究

2.1 区块链技术

2.1.1 概述

区块链是一个分布式共享账本，而它要做的事情就是让所有接入进来的节点之间建立信任关系。从狭义上来讲，区块链是一种按照时间顺序将数据进行区块组合的数据结构，并在密码学的基砥上，保证数据的不可篡改行为。从广义上来讲，区块链技术是利用块链式数据结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、利用密码学的方式保证数据传输和访问的安全、利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算范式。第三方支付平台是大家比较熟悉的一个平台，这个平台采用的就是一个中央大数据，所有的数据交互都需要经过这个大数据库来完成，这个数据库需要时时保持畅通，而且不能出现一点问题，因此第三方支付平台的最大问题在于如何来维护好这个中央数据库。而要维护好这个数据库就比较复杂。区块链采用的是去中心化的技术，所谓的去中心化，就是把这个第三方需要维护的大数据库取消了，谁来维护这些交易信息呢。

众所周知，在一个网络中，会有成千上万个节点，现在让每一个节点来保存所有的这些交易信息，即每一个节点都是一个数据库的维护者，同时也是数据的共享者。这就成功的将一个大数据库给“去”掉，因此是去中心化的。仅仅去中心化是不能保证数据安全的。需要将各个区块进行链接，在每一个区块中都保留着多笔交易的信息，而区块之间的链是有很大的关系的。下一个区块的生成是根据上一个区块的内容来生成的，区块与区块之间是环环相扣的，更改一个区块，将造成下一个区块内容的不一致。因此数据是安全的。现在的区块链可以分为三种。公有区块链，联合区块链以及私有区块链。顾名思义，公有区块链是世界上任何一个接入进来的节点都可以和这个网上的节点进行交易，并且交易能够得到该公有区块链的认可，任何人都可以参与其中。公有区块链是最早的区块链，也是应用最广泛前景最好的区块链。

著名的比特币平台就是公有区块链，所以可以说公有区块链最具权威性。联合区块链也叫联盟链，是由几个公司或者组织共同来维护的一条数据区块链。在这个区块链中，由该群体内部指定的多个预选节点作为记账人，每一个区块链的生成由所有预选节点共

同决定，其他的接入节点可以参与交易，但不会去关注记账的过程，其他任何节点都可以通过该区块链开发的 API 进行数据的交易。记账人的多少，以及如何决定这个记账人是问题的关键所在。私有区块链是由一个公司内部自己使用的记账技术，当然也可以是个人，其独享该区块链的写入权限。私有区块链目前没有得到广泛的应用。本次采用的区块链是布比区块链，布比区块链[9]已经广泛应用于数字资产、股权债券、贸易金融、供应链溯源、商业积分、联合征信、公示公证、数据安全等领域，并正在与交易所、银行等主流金融机构开展应用试验和测试。以多中心化信任为核心，致力于打造新一代价值流通网络，让数字资产都自由流动起来。

2.1.2 共识算法

分布式系统中最根本和最重要的问题是一致性问题，它涉及多个服务节点的操作集合，根据商定协议，它们具有一定程度的协同作用。与此形成对比，共识指的是一致性过程，也指分布式系统中的不同节点协调事件(或多个事务)的过程。共识算法决定了供给分配系统的过程，大多数节点都同意该过程。在这个分布式系统中，这个过程的定义可能非常广泛，例如，多个交易的顺序、节点设置等。这个问题也经常被称为终端复制。他从同一种状态中得到外部的命令，从而使同样的结果得以保证。

理论上，如果分布式系统中的节点能够在理想条件下稳定工作(即时响应、高带宽)和节点之间的连接立即发生，那么就很容易实现共识过程——通过广播和投票并应答。但现实中的系统都不是十全十美的，如通信系统(物理限制速度延迟，通信延迟处理)，涉及的任何细枝末节都有可能发生故障(设计的系统越大，系统失效概率越大)，如通信网络中断、节点和恶意节点甚至消息篡改。一般来说，失败(故障或故障稳定性)却不被篡改信息的情况称为“非拜占庭错误”或“故障式错误”。伪造信息的响应则被称为拜占庭故障，相应的节点是拜占庭节点[9]。很明显，由于存在“动乱”，在拜占庭故障中达成共识更加困难。

2.1.3 拜占庭问题与算法

拜占庭问题，也被称为拜占庭将军的问题，讨论了如何在允许几个节点做坏事(消息可以伪造)的情况下达成共识。拜占庭容错(BFT)讨论了一种允许拜占庭错误的共识算法。

1982 年，莱斯利·兰波特和其他学者在一篇文章《The Byzantine Generals Problem》

中首次提出了拜占庭问题。这是一种虚构的模型，用来解释异步系统中的共识问题。拜占庭是古代罗马帝国的首都，由于其广阔的领土，一些守卫边界(系统中的几个节点)的将军必须通过信使传递信息，做出某些一致的决定。但是，由于将军们可能有叛徒(系统中的节点是错误的)，这些叛徒将向不同的将军发送不同的信息，试图阻止达成共识。这种情况与分配系统中的几个节点之间的共识非常相似。拜占庭问题是讨论如何让看起来忠诚的将军们就这种情况达成协议[10]。

算法整体的过程如下：首先，通过旋转算法或随机选择将某一个节点选择为主节点，称之为一个视图(View)，直到主节点切换。在某一视图中，客户请求<REQUEST,operation,timestamp,client>到主节点，主节点负责所有其他行节点的广播请求。所有节点处理完成请求，并返回处理结果<REPLY,view,timestamp,client,id_node,response>。最后客户检查是否有至少 $f+1$ 从不同节点获得相同的结果作为最终结果。

拜占庭的“不稳定算法”往往会影响最恶意的“混乱”存在，而商定的结果往往会受到广泛场景的影响。

2.2 密码学与安全技术

毫无疑问，密码学作为信息技术安全技术的核心的重要性。没有现代密码学和信息安全技术，人类社会将无法完全进入信息时代。密码学和安全技术的最新进展广泛用于区块链技术和分布式超级账本，特别是身份验证和隐私保护技术。

2.2.1 Hash 算法与数字摘要

哈希算法(hash 或散列)，通常被称为指纹或文摘算法，是非常基本和非常重要的算法。任何长度的二进制未加密明文串都可以与较短(通常是固定长度)的二进制 hash 值相匹配，不同的开源文本很难匹配相同的 hash 值。

当前流行的哈希算法包括国际 MD、Secure Hash Algorithm (SHA)和国内 SM3 算法。SHA 的算法是由美国研究所 NIST 开发的。为了提高安全，NIST 后来开发了更安全的 SHA-224、SHA-256、SHA-384 和 SHA-512 算法。与 SHA-3 相关的新一代算法也在研究中[10]。

2.2.2 加解密算法

现代加解密系统的典型组件包括算法和密钥(包括加密密钥[10]、解密密钥)。其中，加解密算法自身是固定不变的，并且一般是公开可见的；密钥则是最关键的信息，需要安全地保存起来，甚至通过特殊硬件进行保护。一般来说，密钥必须在加密前随机产生于某种算法[10]，且越长加密就越复杂。加解密的典型过程如下图所示。加密过程中，加密算法和加密密钥以获得加密文本。解密过程中，密码索引会经由解密算法和解密键解密以获得解码。加密过程中，通过加密算法和加密密钥，对明文进行加密，获得密文；解密过程中，通过解密算法和解密密钥，对密文进行解密，获得明文[14]。

加密和解码算法是现代密码学的主要技术，可以从项目概念和应用程序脚本使用场景分为两种基本类型，如下表 2.1 所示。

表 1.1　加密和解码算法

算法类型	特点	优势	缺陷	代表算法
对称加密	加解密的密钥相同	计算效率高，加密强度高	需提前共享密钥，易泄露	DES、3DES、AES、IDEA
非对称加密	加解密的密钥不相同	无需提前共享密钥	计算效率低，存在中间人攻击可能	RSA、ElGamal、椭圆曲线算法

2.2.3　非对称加密

不对称的加密是最伟大的现代加密技术发明，它能有效地解决对称的加密技术需要有安全的密钥分配问题。不对称的加密密钥和解密密钥由名字命名为私钥以及公钥。私钥通常用随机数算法来生成，随后，就可以用私钥计算得出公钥。其中公钥通常都是公开状态的，而且能供他人使用，而私钥则属于个体，并且必须受到保护。

不对称加密算法的优点是公钥和私钥是分开的，分配密钥不需要安全通道。缺点是其处理速度(生成过程和解密过程)比较慢，产生的密钥的安全强度也要更低。不对称加密算法的安全性通常是基于数学问题，包括经典的数学问题，如大数质因子分解、离散对数和椭圆曲线。不对称加密代表算法包括：RSA、ElGamal、椭圆曲线(Elliptic Curve Crytosystems，ECC)、SM2 等系列算法[15]。

本论文使用私钥来完成 MetaMask 轻钱包的账户验证，连接到区块链网络。

2.3　以太坊项目

2.3.1 以太坊项目概览

在区块链技术中，以太坊项目也是一个著名的开源项目。作为一个公共平台，以太坊进一步扩大了数字货币交易的能力，支持更复杂、更灵活的应用场景，支持智能合约功能。在此之后，区块链的应用从基于 UTXO 的数字货币交易扩展到图灵完备的通用计算领域。用户不再局限于使用比特币脚本支持的简单逻辑[16]，但他们可以自己设计合同中的任意复杂逻辑。这为创建各种高级别应用程序打开了大门，这些应用程序非常重要。

以太坊项目最初的目标是创建一个平台来运行智能合约 (Smart Contract 平台)。该平台支持完整的图灵应用程序，并根据商定的智能约逻辑自动执行。理想情况下，不应有简单、检查、欺诈和第三方干预。以太平台目前支持 Golang、c++和 Python 等几种语言实现的客户端。由于以太坊平台基于比特币网络内核实现的基本理念，许多设计特性与比特币网络非常相似。

基于以太坊项目，以太坊团队目前使用开放区块链平台以太坊网络。智能合同开发人员可以很容易地开发去中心化的应用程序(DApp)，使用官方可用的工具和专有的以太坊开发语言。这些应用程序将使用以太虚拟机(EVM)。用户通过以太币购买燃料(Gas)来支持部署的应用程序。以太坊的底层也是 P2P 网络平台，类似于比特币网络，智能合约在网络以太虚拟机中运行。网络本身是公开的，任何人都可以访问网络中的数据并参与其中，提供资源来运行以太虚拟机[16]。

2.3.2 权益证明算法

数字货币交易需要验证整个网络，这要求分布式节点有一个一致的共识算法来确保特定问题的通过。目前市场上的大多数数字货币都使用工作量证明机制(POW)，一些数字货币则使用权益证明机制(POS)作为资本保护机制。

POW 是对已经完成的交易的验证。交易可以包括多种类型，如比特币中的哈希计算，这取决于数字货币的不同概念和设计算法。工作量证明算法的特点是，它需要大量的计算成本，比特币是用这种算法开发的。POW 的目标是防止恶意伪造虚假交易。伪造者需要花费超过 50%的比特币网络计算能力来完成一个特定块的破解，巨大的成本与回报不成比例。

在 POW 算法中，新交易记录传输到整个网络的节点后，每个节点都通过计算力竞

争进行工作负载核查，比特币是一个数学问题，完成任务的第一个节点将得到包装区块的权利，并根据激励措施获得一定数额的数字货币奖励。从技术上讲，POS(Proof Of Stake)可以减少因复杂计算而导致的计算资源的消耗，其包装权力取决于节点账户中储存的数字货币数量。作为交易机制的一部分，交易发起者必须提供一定的消耗 Gas 作为交易成本来支付打包区块的费用，而矿工必须提供一定的储备金来为恶意行为者提供惩罚，从而限制他们的行为。

2.3.3 以太坊和 ERC-20 标准

Ethereum 的新设计是基于 ERC-20 标准，由 ERC-20 提供并实现的数字货币发行技术开启了 ICO 这一新的商业模式。ERC-20 标准是数字货币发行人的技术支持。根据 ERC-20 标准智能合同发布令牌，可以立即与以太钱包兼容，也可以进行交易。

Ethereum 的智能合同编程使用了 Solidity 语言，并在 EVM 环境中运行。EVM 就像一个沙箱，使智能合同独立于以太网块，提供安全和精确的执行。

2.3.4 区块链服务器环境

VMware 虚拟机 VMWare (Virtual Machine ware)是一家虚拟 PC 软件公司，允许在同一台机器上同时运行两个或多个 Windows、DOS 和 LINUX 系统。与“多载”系统相比，VMWare 采用了一个完全不同的概念。多载系统一次只能运行一个系统，当系统切换时，计算机必须重新启动。VMWare 实际上是“同时的”，在主系统平台上有多个操作系统，作为标准 Windows 应用程序切换。在每个操作系统中，用户可以在不干扰实际硬盘数据的情况下进行虚拟分解和调整。甚至可以通过网络地图连接多个虚拟机，作为一个本地网络，这是非常方便的。VMware 操作系统的性能远低于直接安装在硬盘上的系统。

Truffle 是世界一流的开发环境，为以太坊的测试框架和资产管道，致力于使以太坊开发者的生活更容易。

Ganache 是为以太坊开发准备的个人区块链钱包，可以用它执行智能合约，研发应用和执行测试用例。

MetaMask 是一款插件形式的以太坊轻客户端，开发过程中使用 MetaMask 和 DApp 进行交互是个很好的选择。

2.3 区块链的特点

(1) 去中心化。因为区块链采用的是分布式存储的技术，因此不存在中心化的数据库。所有接入该区块链的节点都具有相同的权利和义务，它们共同来维护这个区块链系统。

(2) 自治性。区块链采用协商好的规范和协议来使所有接入的节点进行数据的交互，因此人为的干预将不起作用，其通过这套协议来维持区块的生成以及交易的进行。

(3) 不可篡改性。信息一旦写入区块链，并且被所有节点所共识，那么这个数据将不可被更改。因为更改本条数据将会导致下一个区块的生成不一致，共识是不能达成的。写入区块链中的数据具有很高的可靠性。

(4) 匿名性。任何加入区块链的节点都共同遵守着相同的算法，因此两个加入的节点是不需要信任对方的，因为信任是被区块链验证过的。这样一来，就可以保证加入的节点随意的进行数据交互了。

2.4 区块链的运转

区块链的要能正确运转起来，首先就必须要解决重复支付的问题，也就是不能造假币。比特币的创始人中本聪就提出来了一个很好的方案，给每一笔交易盖时间戳。在区块链上每十分钟会形成一个区块，把这十分钟的全网交易都正确的盖上时间戳。那么问题来了，选谁来盖这个时间戳。这就出现了矿工，矿工是接入该区块链上的节点，矿工可以去竞争这十分钟一个区块的记账权，竞争的规则是正确记账的同时要去解难题，谁能最快的并且正确的解出这道题，那它就是这十分钟区块的合法记账人，并得到一定数量的比特币奖励。这就是基于区块链技术的众筹平台的设计与实现俗称的挖矿过程，其本质是建立区块链去中心化的信任过程。中本聪也在比特币白皮书中详细的介绍了信任系统是如何建立的。第一步。每一笔交易为了让别人认可有效，必须要广播给该网络上的所有节点。第二步。每个节点都要正确无误的给十分钟的每一笔交易盖上时间戳并记入该区块中。第三步。每个节点都会为了获得比特币的奖励而去解难题，从而获得合法的记账权。第四步。一个节点获得该区块的记账权之后，它向全网所有节点公布所有盖时间戳的交易，其它节点进行监督核对。第五步。其它节点核对无误之后开始下一个区块记账的竞争。此时就已经形成了一个合法记账的区块链。因此，可以看出，每一个区块的生成都需要以上五步的审核，如此进行下去就形成了一条完整的区块链。

2.5 点对点网络与区块链网络

区块链就是一个复杂的点对点对等网络。点对点对等网络之间的节点是平等的，同级的。因此每一个节点既可以当客户端也可以当服务器。点对点网络使用互联网让各节点之间直接进行通讯關，因此在一定程度上使得网络的沟通变得更加容易，实现了资源的共享，真正的消除了中间商，也就是所谓的中心化服务器。其特点就是可以直接连接到网络上的另一个节点，进行数据的交互，免去了中间的服务器，点对点网络另一个特点就是将数据的管理权限交给了每一个节点，而不是服务器。在现在的互联网上，有很多都是点对点网络的服务比如通讯软件 ICQ，MSN，OICQ 等都是点对点网络的实际应用。点对点网络与区块链网络的比较点对点网络从其特点上来看，跟区块链有着很大的相似。下面对点对点网络和区块链网络进行详细的比较。

非中心化与区块链不同的是，点对点网络的资源和服务分散在所有的节点上，节点与节点之间传输数据是可以直接进行的，而区块链则需要进行消息的广播。相比与区块链网络，点对点网络无需中间的服务器，因此可以很大程度上避免瓶颈问题的出现。

可扩展性在点对点网络中，用户可以随时加入这个网络，每加入一个节点，点对点网络的资源也在同步的增加，因此用户对于资源的需求也将更容易得到满足，从理论上来讲，点对点网络的可扩展性是无限大的。与点对点网络不同的是，传统的 FTP 文件下载模式是随着用户的增加下载速度将会变得越来越慢。但是点对点网络正好相反，节点越多，下载的速度越快。区块链的扩展性也是相当大的，这一点和点对点比较符合。

健壮性点对点网络具有高容错的特点，因为网络中的资源是分布在每一个节点中的，即使一个节点遭到了破坏，整个网络也并没有受到多大的影响，而且点对点网络具有自我调节能力，在一些节点离开之后，能够自动调节其网络拓扑结构，因此节点的加入和离开都比较方便，而且不会影响这个网络。区块链网络也有这相似是特点，但相比之下，区块链网络中节点越多，网络的健壮性就越好。综合的来说，区块链网络是点对点网络的升级版，在点对点网络的基础上，增加了一些区块链对数据的验证，从而保证了区块链网络的安全性和可靠性。

3　智能合约总体分析与设计

3.1　构建智能合约

3.1.1　区块链技术框架

区块链的区块是一种数据结构，它将数据按时间顺序存储在一个链表中，可以无限延伸，就像一本帐一样。块的结构由分布式、非中心的主节点、点对点的计算机网络维护。链表中的每个块包含多个交易事务，事务代表数据库状态的变化，例如，资金从一个账户转移到另一个账户。交易由网络中的几个节点验证，并最终保存在块中。每个单元包含一个签名的 hash，它包含了连接列表中的最后一个单元的内容，区块链用图 3.1 来表示：

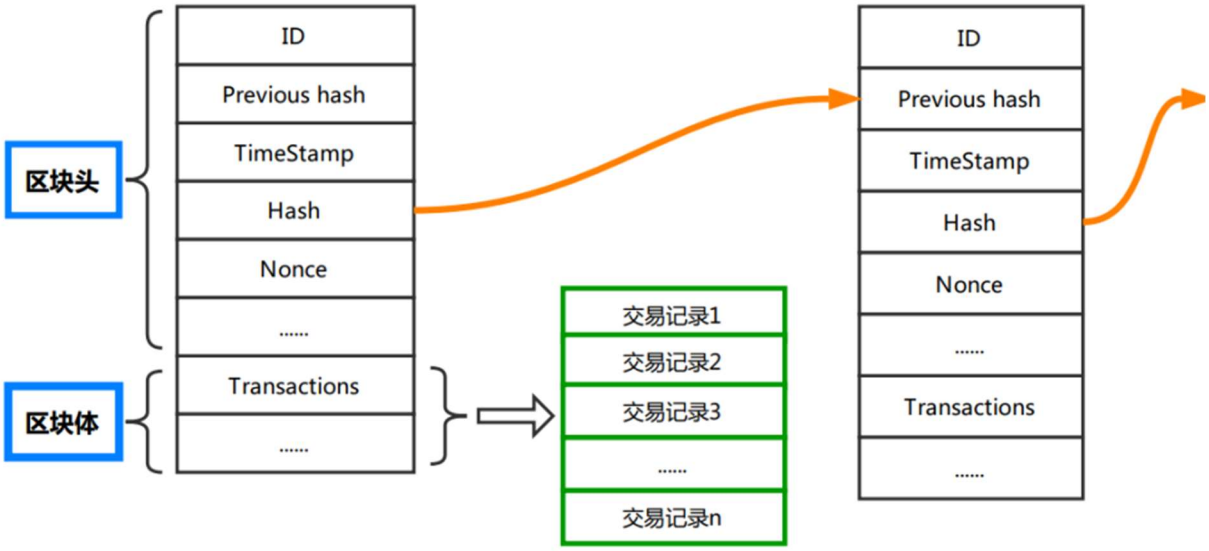

图 1.1　区块链框架

区块链的地址是系统中唯一的账户。地址实际上是一把公钥，相应的私钥属于创建账户的用户。没有私钥做数字签名，就不能从这个帐户输出资产。在加密和保护交易时，块还提供数据库状态的分布式协议。某节点找到了证明工作量的方法，可以指定下一个块并发布，网络上的其他节点检查新创建的块。获胜的节点[18]将获得两个奖项：获得新创建的数字货币奖励，并向创建交易的一方收费。

3.1.2 以太坊的智能合约

集成智能合约被添加到构建以太坊区块链区块中——一种具有价值、存储数据、封装代码和执行计算任务的应用程序。和比特币一样，以太坊也包含一种叫做以太的数字货币。以太被计算机节点挖掘出来，其交易被节点检查，随后被保存在分布式共识链中。以太可以在账户和智能合约之间传输。

智能合约允许匿名方签订限制协议，每个成员都完全了解交易。资产可以在账户之间转移，也可以放在与第三方的智能合约中。因为合同是一串代码，开发人员可以做应用程序能做的任何事情，故想象力是唯一的限制因素。

3.2 股权众筹合约的分析与设计

3.2.1 系统框架

在 2.2 小节分析以太坊平台中智能合约的编程方法，本课题提出了基于 Ethereum 区块链服务的智能合约设计路线，对可编程智能合约的系统框架定义。

可编程的智能合约系统框架分为三个层次：应用层、接口层、服务层。

3.2.2 区块链的应用程序 DApp

DApp 是由智能合约和后台代码组成的，它是区块链分散式应用程序。在 DApp 中，不仅需要开发前台界面的应用程序，还要分析和设计后台逻辑的智能合约代码，这是因为以太坊 DApp 的所有服务和逻辑都是由区块链驱动的。事实上，这正是以太坊生态圈进展迅速的最大原因。

以太坊去中心化的核心是它使用图灵完备的脚本语言的能力，而以太坊智能合约有四种语言可供选择：Serpent、Solidity、Mutan、LLL，这些语言都是为基本语言而设计的。目前，Solidity 是首选语言，因为它具有内置 Serpent 语言的所有功能，句法类类似于使用广泛的 Java。与较低的语言能力相结合，Solidity 语言有助于执行完整的智能合约系统。

这些智能合约代码在 JsonPRC 模式下提供给应用程序，智能合约通过以太网传输给所有节点，通知这些节点启动 ABI 智能合约，然后调用的 ABI 将在该节点的虚拟机器上运行。最后，完成的过程和结果被包装成一个新的区块并链接到主链，整个网络通过

同步实现区块链的统一[19]。

本文对基于区块链的股权众筹平台系统框架图进行研究和设计，并得到如下图 3.2 所示的结果。

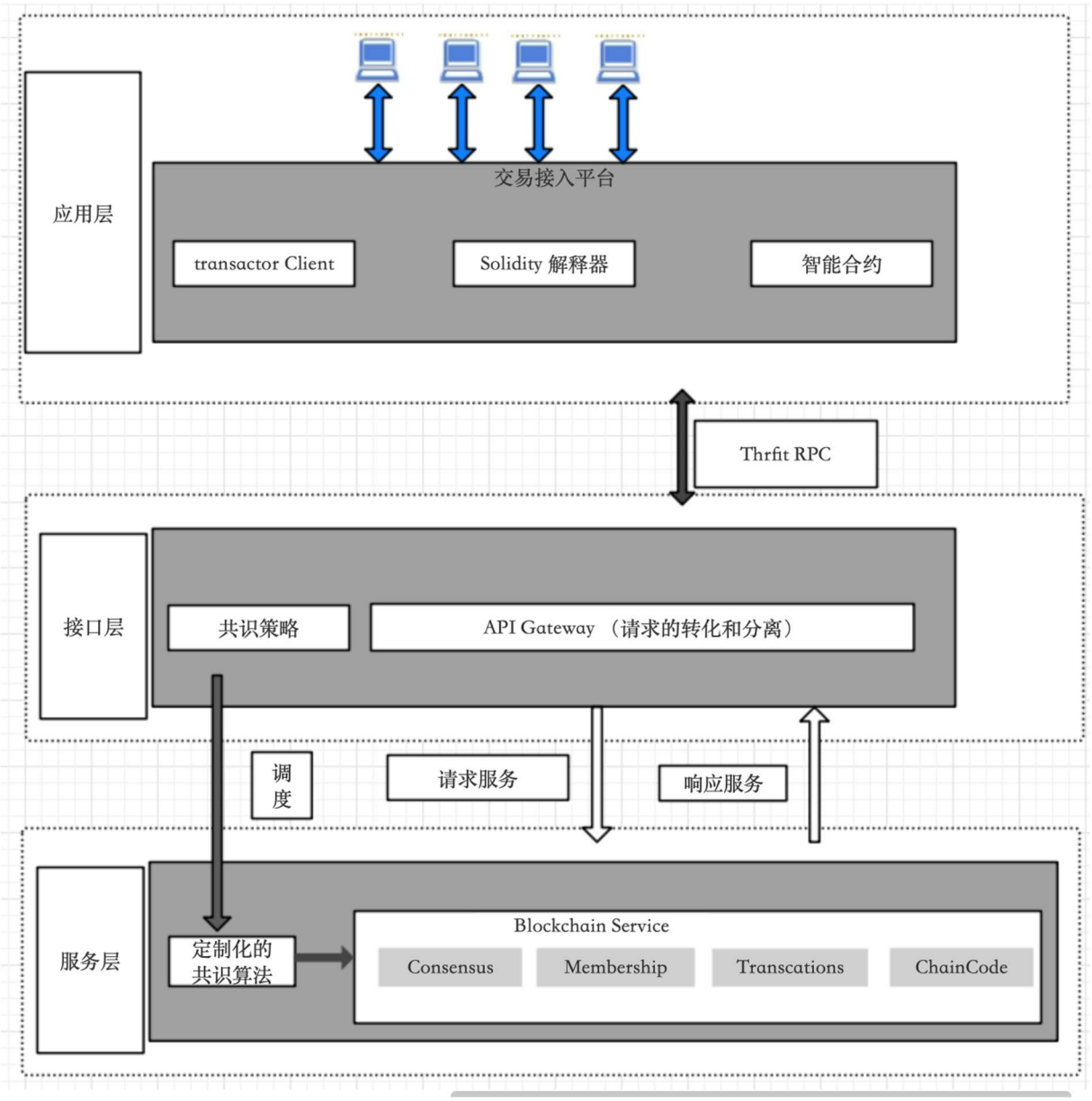

图 3.2　智能合约应用框架图

3.3　项目需求分析

在现代中国社会中，许多人手里拿着很多钱，又有许多人手里拿着好主意。矛盾的

是，前者找不到投资的对象，后者找不到支持创造力的资本。自改革开放以来，无数的前辈向我们解释了“发展就是硬道理”的基本原则。马云、雷军和刘建东等新一代企业家通过互联网塑造了中国新企业发展的神话。他们的事业激励了许多年轻人在“大众创业、万众创新”时代进行风险投资。许多初创企业在早期和中期遭受了挫折，甚至失败:原因是它们缺乏资金支持，环境太小。因此，解决这两个基本条件值得我们考虑和实践。

目前，国内外众筹平台都使用单一的数据库管理系统来管理平台数据，因此用户数据面临着伪造和欺骗信息的风险，这损害了项目参与者和投资者的权利。基于本文所讨论的基于区块链技术的股权众筹平台具有去中心化的特点，并使用分布式数据库。

3.4 智能合约应用分析

本文设计的股权众筹原型系统包含应用层的平台设计和服务层的智能合约设计。智能合约采用以太坊提供的平台进行开发。

智能合同的特点是低成本、安全的执行环境和简单的审计，使智能合约具有多种用途。传统的企业合同是建立在外部权威的批准之上的，使用智能合约取代了这种情况，保护合同参与者的权利，减少了对正义的争论。

首先，智能合同主要用于股票交易、数字资产管理和其他场景。在集权系统中实现知识合同具有分散化的特征，每个账户记录都提供了同等的能力，可以保证交易的透明度和公平。其次，智能合约提供数据保护。由区块链和智能合约编写的哈希签名和共识机制可以有效地防止数据伪造和限制数据的访问。最后，块交易是按时间顺序进行的，智能合约可以用来规范化工作过程和验证数据完整性[20]。

由于区块链技术和智能合约仍处于开发阶段，并且对系统有潜在风险，鉴于不成熟技术的现实，系统的可用性和智能合约的有效性必须充分反映在众筹系统的设计开发中。这还要求系统管理员检查智能合约的语义和词汇方面，并检查智能合约的后续执行。

3.5 区块链技术下的股权众筹系统分析

3.5.1 功能需求分析

根据用户的需要，为了确保正确的信息管理，本文中讨论的众筹平台不仅实现了启动和跟踪用户众筹的基本功能，而且具有良好的用户经验，并将相关的所有交易转移到区块链上。区块链交易保证数据安全和用户权利不会受到侵犯。

众筹平台的主要功能应该包括：

用户模块：用户模块是一组函数，如用户注册与登录、信息请求和信息更改。项目创建模块：注册平台用户可以通过平台创建新项目。进行投资模块：注册的平台用户可以查看和投资已经被审查的项目。控制模块：管理员的账户可以验证新的在线项目，并控制平台用户。

3.5.2 业务流程分析

项目发起人在股权众筹平台上发布项目信息，所有平台用户都可以接收和浏览相关信息。在创建了项目信息后，它被转移到管理面板进行检查。该项目将在平台上发布，如果失败，将返回投资者。项目符合，投资者选择投资项目，并根据项目信息和风险分析进行交易。在众筹成功之后，众筹参与者将根据智能合约条款获得一定数量的令牌，但未能将资金归还给参与者，项目结束。平台的业务流程如图 3.3 所示。

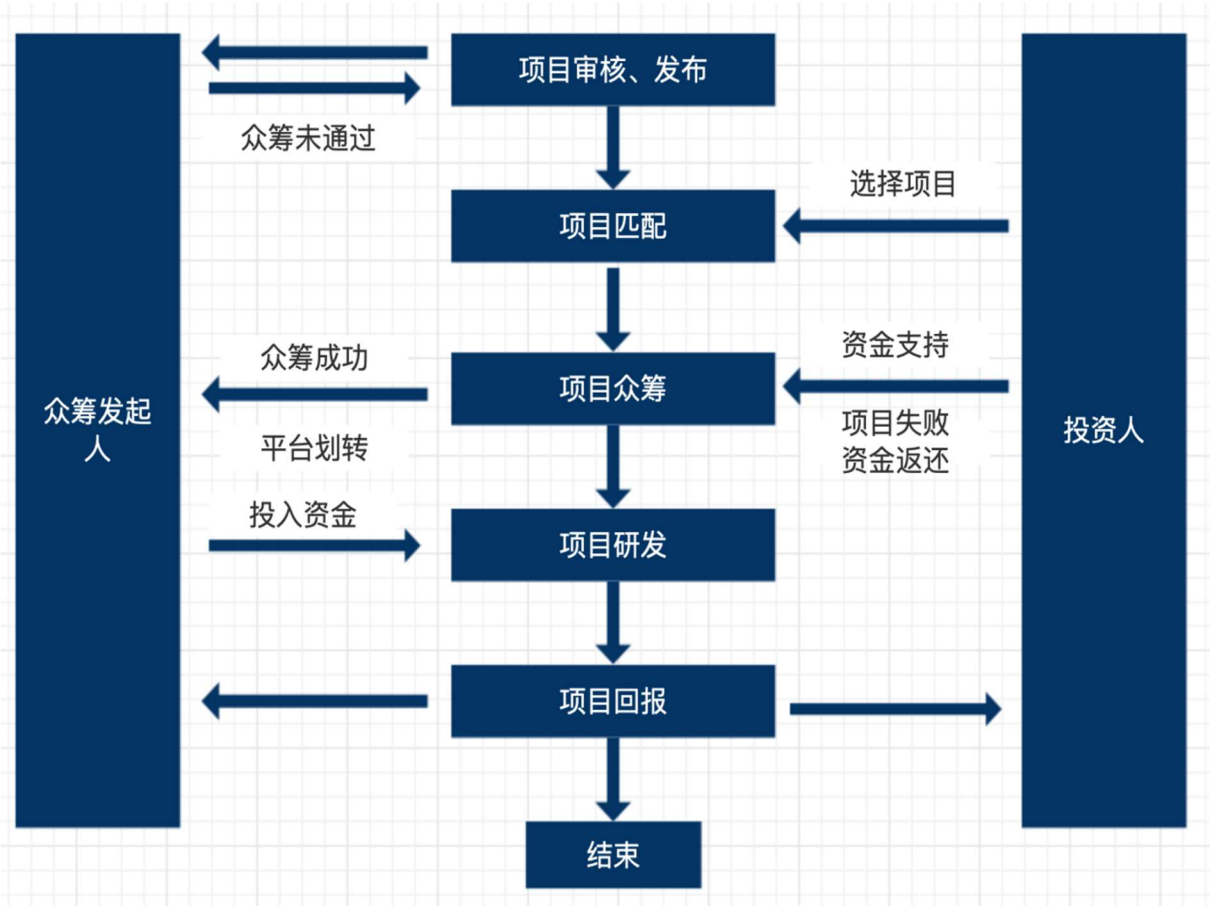

图 3.3 业务流程图

3.6 数据库设计

通过对计算机数据库应用技术的使用，可使数据存储过程中得到所需的技术支持，进而为数据资源的实践应用水平。一个好的数据库可以保证系统更加流畅，数据库的设计也是比较重要的一部分。本众筹系统的数据库包含四个表，接下来对每个表进行讲解。

3.6.1 用户功能模块

用户功能模块主要负责管理用户信息，这样用户就可以进入系统，更新自己的信息，查看自己的信息。注册功能：允许访客记录可以执行需要许可功能的用户。为了确保用户不是垃圾邮件，所有注册用户都必须通过电子邮件激活才能进入网站。当用户完成邮箱激活功能时，允许用户进入系统。更新：进入系统的用户可以更新自己的信息，包括用户的姓名、性别、个人数据和其他信息。更改密码：进入系统的用户可以更改输入的密码。

用户功能模块的数据表设计如表 3.1 所示：

表 3.1 用户模块数据表设计

字段名称	数据类型	长度	允许为空	说明
id	int	11	NOT NULL	主键，自增类型
username	varchar	20	NOT NULL	用户姓名
gender	enum	2	NOT NULL	性别
email	varchar	255	NOT NULL	用户电子邮箱地址
staff	enum	1	NOT NULL	是否为管理员

3.6.2 项目创建模块

项目创建模块负责保存新创建的用户项目信息，一旦批准，其他用户可以搜索和浏览项目。在本模块中实现的具体功能，以及数据库表设计情况如下表 3.2 所示。基本信息：新项目的基本信息，包括名称、分类、目标、完成时间、描述等。上传项目图片：从被本地上传与新项目相关的图像。检查信息:检查并确认上述新项目的全部信息，以确保可靠的信息，并最终上传到数据库中。

表 3.2 项目创建模块数据表设计

字段名称	数据类型	长度	允许为空	说明
id	bigint	20	NOT NULL	主键，自增类型
creator_id	bigint	20	NOT NULL	创建项目的用户 id
created	datetime	-	NOT NULL	创建项目的时间
expires	datetime	-	NOT NULL	项目持续时间
contract_address	varchar	255	NOT NULL	项目在区块链的地址
amount_invested	bigint	20	NOT NULL	已筹备资金
investment_wanted	bigint	20	NOT NULL	目标资金
status	enum	1	NOT NULL	状态(是否关闭)
main_description	text	-	NOT NULL	对项目的主要描述
investor_count	int	20	NOT NULL	已投资项目的人数
main_picture	varchar	255	NOT NULL	图片的证明材料地址

3.6.3 项目信息模块

该项目的信息模块是对上传项目的详细映射，供投资者咨询。基本信息：包括项目名称、开始和结束时间。项目描述：更详细的项目描述吸引投资者。创建者的信息：关于该项目的主要信息是为了促进双方之间的了解。评论：所有注册的平台用户都可以在这里评论。支持者：展示支持这个项目的用户。更新：稍后引入项目的信息更新。编辑项目：项目创造者编辑项目信息的权利。

项目更新表 investments_faq 如下表 3.3 所示：

表 3.3 项目更新数据表设计

字段名称	数据类型	长度	允许为空	说明
id	bigint	20	NOT NULL	主键，自增类型
investment_id	bigint	20	NOT NULL	项目 id
message	text	-	NOT NULL	更新信息

3.6.4 信息管理模块

信息管理模块是管理系统信息的功能模块，包括用户信息、项目信息、审计队列和系统调整。用户管理：管理员用户可以编辑账户信息、姓名、性别、电子邮件地址、密码更新、激活、管理、删除用户等。项目管理：项目人员可以进入后台编辑项目。审计：新创建的项目必须得到背景调查的批准，才能在前景中看到。设置：系统功能的综合调整。

项目审核数据表设计如表 3.4 所示。

表 3.4 项目审核数据表设计

字段名称	数据类型	长度	允许为空	说明
id	bigint	20	NOT NULL	主键，自增类型
user_id	bigint	20	NOT NULL	项目创建者 id
investment_id	bigint	20	NOT NULL	项目 id
status	enum	1	NOT NULL	审核状态(0，1，2)

4 系统详细设计与实现

4.1 股权众筹平台系统模型

众筹是一种商业模式，用来在互联网平台上为广大用户筹集资金。这些商业模式的参与者是项目创造者、投资者和金融平台的创造者。本文所讨论的众筹平台基于区块链技术和智能合约，区块链技术提供了众筹的透明度和可靠性，智能合约规定了项目参与者的行为限制和融资控制[21]。区块链采用协商好的规范和协议来使所有接入的节点进行数据的交互，因此人为的干预将不起作用，其通过这套协议来维持区块的生成以及交易的进行。

在计算机领域，众筹项目的参与者与系统角色相匹配，项目的创造者是区块链网络系统的终端，区块链即为众筹平台。

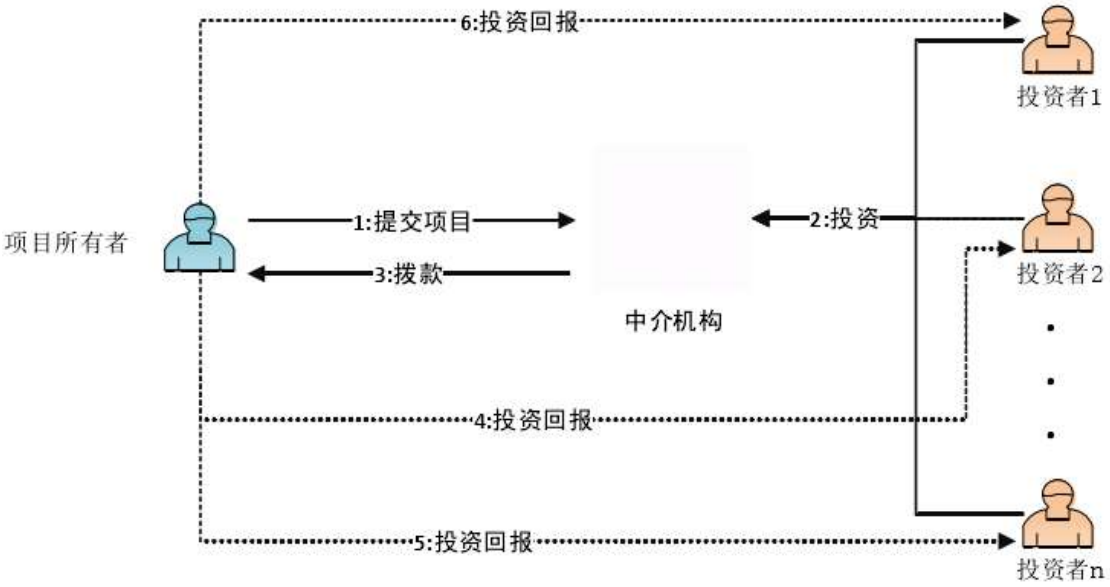

图 4.1 传统众筹平台

传统的基于中间机构的股权众筹平台(如图 4.1)主要有两个问题：首先，中介机构伪造数据和虚假营销，损害项目参与者的利益；其次，中介机构使用项目资金，在雇用期间非法挪用资金会损害项目参与者的利益[21]。

本课题所讨论的基于智能合约的众筹平台(如图 4.2)系统有效地解决了传统众筹平台的两个主要问题：众筹合同的限制和规则允许众筹项目直接将资产转换成数字货币，使用不可逆转的修改和跟踪能力，使众筹平台透明可靠。

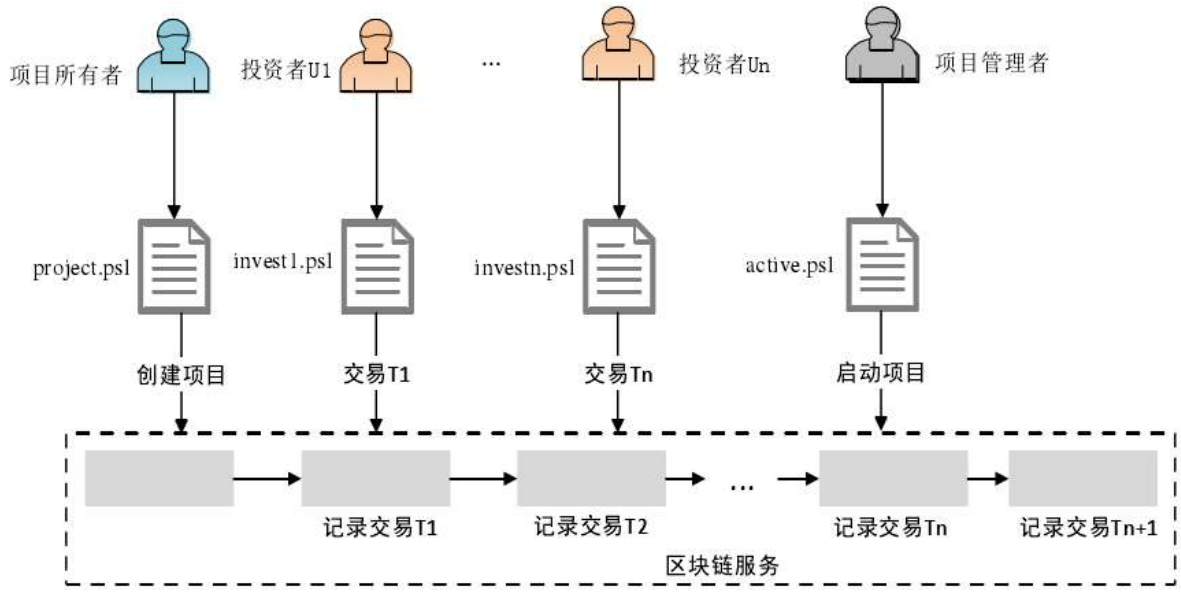

图 4.2 基于区块链的众筹平台

4.2 网站架构

本文实现的股权众筹平台在设计上按照层次结构划分为三部分：应用层、服务层、数据访问层。这是按照软件设计理论的分层而治的思想进行划分的，优点就是便于每个模块的代码编写规范以及后期维护工作的展开。

(1) 应用层是程序级别负责与用户的互动，为投资者和项目参与者提供了良好的工作体验。本文开发的股权众筹平台使用 PHP 作为应用程序的主要 web 应用开发语言，页面设计也使用更受欢迎的 bootstrap 框架环境[21]。

(2) 服务层提供三个功能模块来创建智能合约、众筹项目查看、投资清算众筹项目的项目。服务层和应用层之间的相互作用使用 Ajax 技术和 MeatMast 插件来实现合约交互。服务层承担的功能是业务逻辑处理，负责接收、解析、过滤并验证从应用层传递来的数据服务请求，并执行相关业务代码，最后将处理结果返回到应用层进行展示以及返回到数据层进行持久化存储。

(3) 数据访问层主要负责存储、同步和备份数据。该系统使用两个数据库服务器，一个是 Mysql 数据库，用于平台信息的存储与数字化管理；另一个是区块链，它在开发初期的分布式数据库，用户交易事务存储在其上。数据层存储程序运行产生的数据对象信息，为上层的服务层和应用层提供数据服务，以保证数据的有效性和可靠性的读取与保存[21]。

4.3 智能合约架构

基于以太坊的去中心化应用架构如图 4.3 所示：

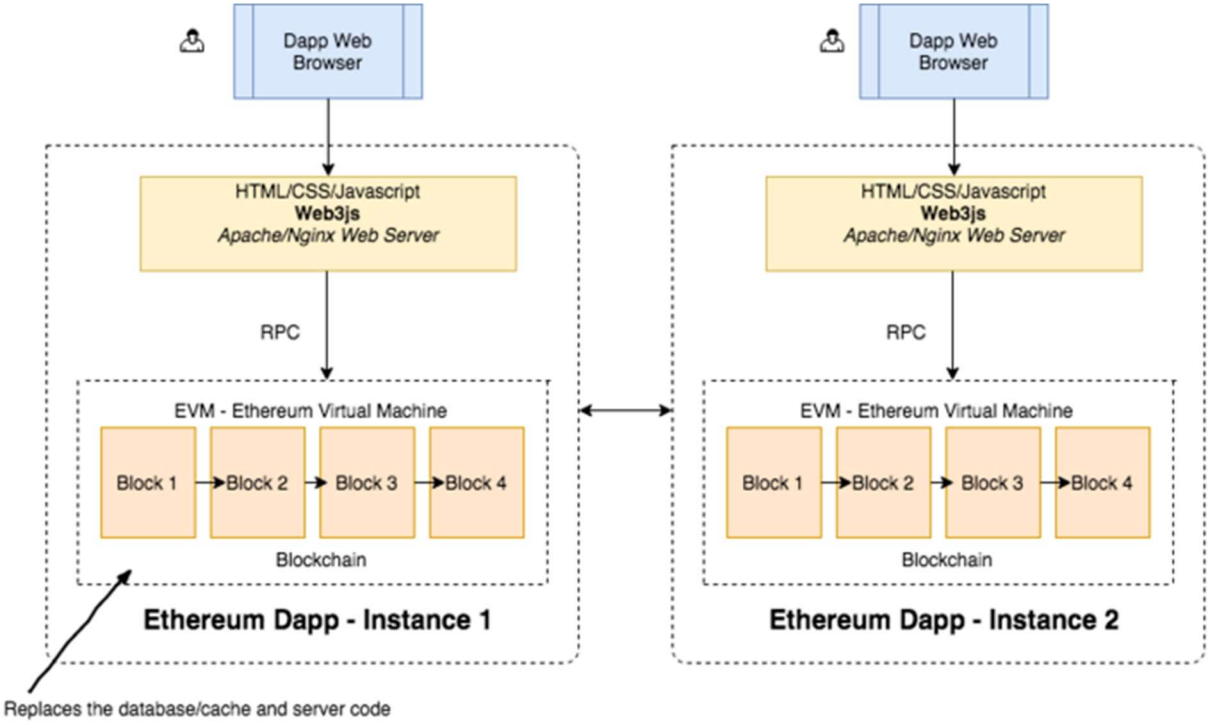

图 4.3 DApp 架构

每个客户端(浏览器)都是与各自的节点应用实例进行交互，而不是向一个中心化的服务器请求服务。

在理想的去中心化环境中，任何想要与 DApp 互动的人都必须在自己的电脑或手机上运行一个完整的区块链节点——简而言之，每个节点都必须启动一个完整的区块链，这意味着用户必须在使用去中心化应用之前下载整个区块链[25]。

但我们并不是生活在乌托邦里，不可能希望每个用户都能先启动一个完整的节点，然后使用你的应用程序。但去中心化的主要想法是不使用集中服务器。因此，在区块链社区中出现了一些解决方案，比如 Infura，它提供了一个共享的区块链节点和 Metamask 浏览器插件。有了这些解决方案，就不需要花费太多的硬盘、内存和时间来启动和运行一个完整的集成节点，可以利用分散化的优势。

Web3.js 是以太坊官方的 Javascript API，可以帮助智能合约开发者使用 HTTP 或者 IPC 与本地的或者远程的以太坊节点交互。Web3 与 geth 通信使用的是 JSON-RPC ，这

是一种轻量级的 RPC(Remote Procedure Call)协议。web3-eth 用来与以太坊区块链和智能合约交互。web3-shh 用来控制 whisper 协议与 p2p 通信以及广播。web3-bzz 用来与 swarm 协议交互。web3-utils 包含了一些 Dapp 开发有用的功能。

4.4 功能模块详细设计

4.4.1 用户功能模块详细设计

注册功能：用户填写邮箱信息、账户名称和密码，首先确定正确的输入格式，通过：$match = '/^[_\w\d\x{4e00}-\x{9fa5}]{'.$minLen.','.$maxLen.'} 这条正则表达式来判断输入是否合法。然后确定是否在系统中注册了邮箱。如果没有注册，对用户输入的密码进行加密$password = encrypt_password($password)，数据库中保存密文[25]。

登录功能：用户输入邮箱和密码信息。首先确定正确的输入格式。这取决于数据库中的邮箱是否存在，以确定用户是否存在。用户登录成功后，将用户名放到 Session 之中进行暂时保管，这里首先要申请一个 Session()对象，然后 username 将保存到 Session 中，Session 的生命周期一般理解为从浏览器打开到关闭。

更改密码：用户输入原密码和新密码，以确定密码格式是否符合上传要求。如果是的话，确定原密码是否正确。具体实现为$sql = "UPDATE ".self::$user_table_name." SET password = '{$new_password}' ，如果是正确的，将当前用户的密码更新到新的密码，并提示成功更改的密码[27]。

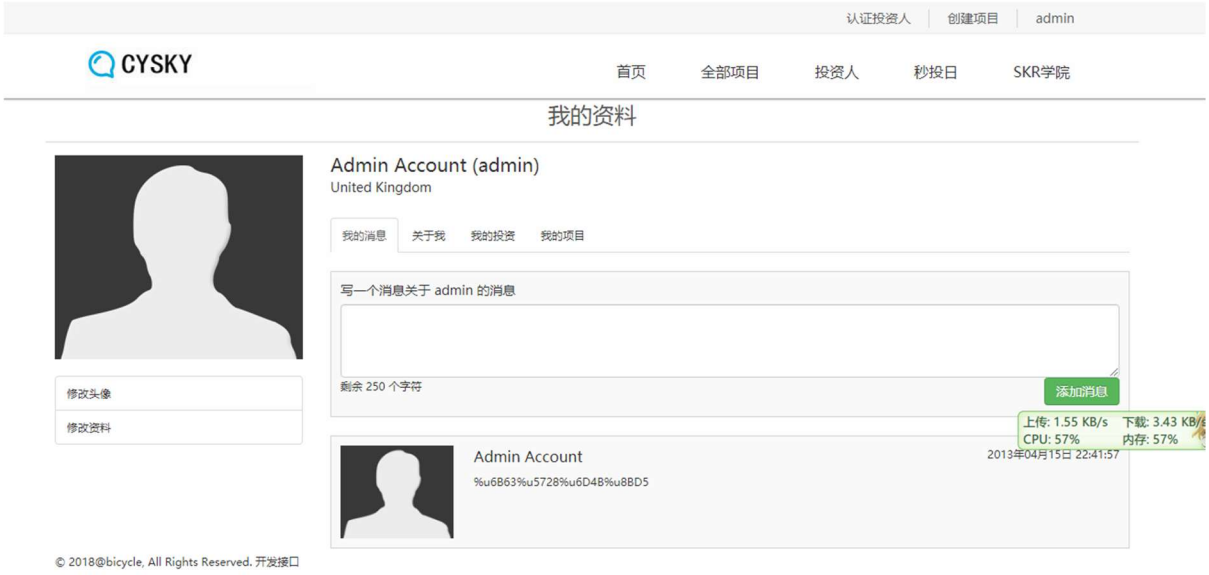

图 4.4 用户模块

4.3.2 项目创建模块详细设计

项目创建模块详细划分具有三部分功能如图 4.5 所示，包含基本信息，上传图片，检查信息。如图 4.5 所示。基本信息：基本信息是用户创建项目的基本描述，需要填写的内容有标题、分类、融资目标、结束时间、投资信息、项目完整信息、描述、主要内容。其中分类是系统设置好的，从数据库中读取详细信息后展示并供用户选择。结束时间默认为 30 天，计算方法为服务器时间减去项目创建日期。上传图片：上传图片是用户对项目的进一步描述功能，需要做到对文件的检查是否符合上传要求。上传时采用 Ajax 方法[28]，并在成功后返回具体信息；上传失败则返回错误信息。检查信息：检查信息是保证用户对所发布项目信息的再一次检查。检查信息时，信息应该对用户封闭，不可再修改。

图 4.5 项目创建模块

4.3.3 项目信息模块详细设计

基本信息：包括项目标题，开始时间结束时间、支持进度、融资目标、项目的具体描述。项目的具体信息从数据库中读取，在前端中展示给用户。项目描述：更加详细的项目介绍。创建者信息：项目创建者的基本信息介绍，有利于双方联系。评论：平台所有已注册用户均可以在此发表评论意见。支持者：显示已支持该项目的用户。所有评论和支持者从对应的数据库查询后显示。更新：后期对于项目介绍信息的更新。更新内容可以编辑后上传给数据库。编辑项目：属于项目创建者的权限，用于对项目的信息编辑。

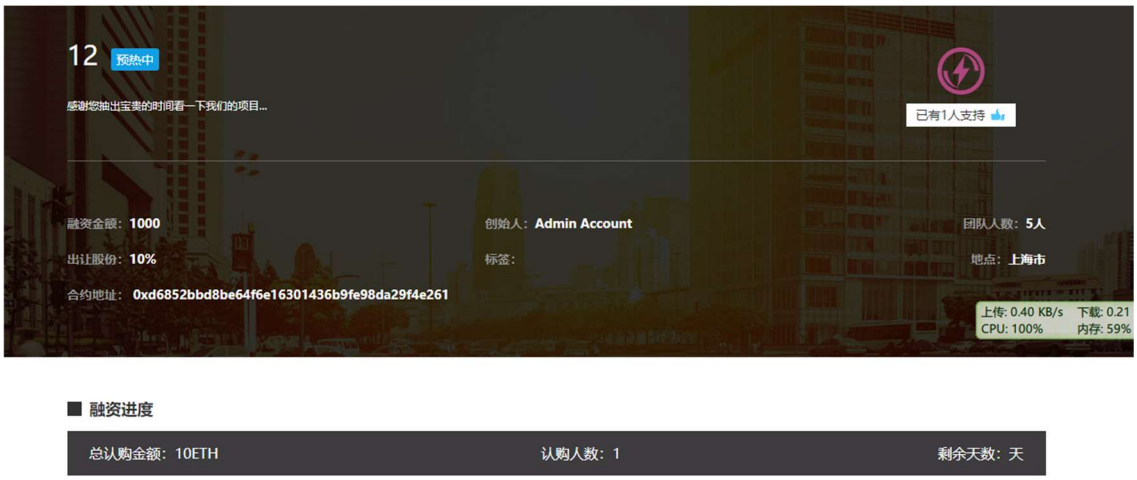

图 4.6　项目信息模块

在项目信息中，投资人可以发起跟投功能。跟投也是区块链的数据操作，需要将数据保存到区块链中，即是记录交易行为。同发起众筹项目一样，跟投操作需要一个公共账户，这个公共账户与第三方账户有本质区别，第三方账户是在系统平台中有用户自身维护管理的账户，而前者公共账户则是存在于区块链中的，因此不可能存在伪造数据或者篡改用户信息的情况发生。公共账户的作用就是为最后的结算设立的账户体系，众筹成功则将币由智能合约直接打给项目方，而交易记录也会保存在区块链中。在浏览器中使用 MetaMask 软件，预先连接到区块链网络(本文在 VMware 虚拟机运行 Ganache 模拟区块链，输出端口为 8545)，输入用户私钥即可完成账户对区块链的连接，还可以显示账户余额信息。

交易完成返回的字节码信息如下表 4.1 所示：

表 4.1　区块链信息

字段名称	数据类型	说明
transaction	hash 地址	交易记录区块地址
contract address	hash 地址	合约创建地址
number	int	区块高度
block time	datetime	区块链分布式系统时间

实验结果如图 4.7 所示：

```
Transaction: 0x0386fd87f385bc799817b2c767abf1a8954ce538505cfdccca38972426f2e74d
Contract created: 0x09359422defb43759efee60da2879331261f610b
Gas usage: 277462
Block Number: 1
Block Time: Tue Dec 18 2018 16:43:05 GMT+0800 (CST)
```

图 4.7　区块交易结果

4.3.4　项目管理模块详细设计

用户管理：可以编辑账户信息，姓名、性别、激活、是否可以管理，删除用户等。项目管理：工作员工登录后台可以对项目进行编辑。审核：用户新建的项目需要后台审核批准才会在前台显示。设置：系统的综合功能调整。以审核项目为例，对数据库项目状态进行更新：$database->query("UPDATE investments SET status = '1' WHERE id = '{$id}' "); 设置状态 status 为 1。设计效果如图 4.10：

图 4.8　项目管理模块

4.4　运行环境及处理函数

本文所论述的系统运行环境配置如下表 4.2 所示：

表 4.2　配置

名称	类型	版本
操作系统	Windows	Version 10
服务器	Apache	2.4.37
数据库	MySQL	5.6
区块	Ganache-cli	7

搭建好了一个运行在本地的模拟区块链，本文的股权众筹系统就可以通过访问此区块链网络进行智能合约部署工作以及数字货币交易。在以太坊的世界里，可以使用 Solidity 语言来编写业务逻辑/应用代码(也就是合约：Contract)， 然后将合约代码编译为以太坊字节码，并将字节码部署到区块链上。

利用合约工厂的方法调用其他 Funding 合约方法 ：

(1) 存储参与者的众筹项目。创建 PlayerToFundings 合同，为每个用户存储所有参与的项目。将 p2f 合约初始化，利用的是 FundingFactory 的构造函数。当项目创建者创建新的众筹项目时，调用(FundingFactory.createFunding)，p2f 引用这些合同给每个项目。当潜在投资者参与众筹时，对应函数为 Funding.support()，调用 p2f 的 join()函数来保存众筹项目。

(2) 读取自己参与的所有众筹项目。投资人作为参与者调用 FundingFactory 合约的 getPlayerFundings 函数。调用 p2f 的 getPlayerFundings 函数。p2f 将存储的 playerToFundings[player] 信息(参与的所有项目地址)返回给 FundingFactory。FundingFactory 最终把调用者参与的所有 address[]返回给调用者。

完整合约参数见表 4.3。

表 4.3 完整合约参数

参数名称	含义或解释
playerToFundings	参与者=> 合约地址数组
fundings	存储所有已经部署的智能合约的地址
creatorToFundings	创建者=> 合约地址数组
getFundings()	路人 查看所有众筹项目列表
getCreatorFundings()	创建者 查看调用者创建的所有众筹项
flag	众筹成功标记
manager	众筹发起人地址(众筹发起人)
goalMoney	目标募集的资金
endTime	默认众筹结束的时间,为众筹发起后的一个月

4.5 交易区块链状态

本文描述的众筹体系中有三种交易情景：第一，投资者投资于众筹项目；第二，众筹项目在智能合约出发条件下产生了代币作为投资证据；第三，项目失败后将返回给投资者。在众筹体系中，持续的贸易将继续产生区块，新的区块将被分配给区块链，区块链的长度将继续增长。新生成的块包括块的高度、块的地址、之前块的地址以及交易信息，包括时间。

通过 ganache_cli 产生虚拟账户，并在命令中使用参数“-h” 来指定区块链网络的节点地址，“-p”指定节点端口号，“-e”指定虚拟账户的初始余额。例如：nohup ganache-cli -h 192.168.88.128 -p 8545 -e 1000 > ganache_cli.out 2>&1 &/. 这里指定了虚拟区块链的节点地址以及端口号，并初始化了十个虚拟账户，每个账户余额为 1000ETH。在 chrome 浏览器中打开 MetaMask 插件，首先连接到区块链网络，成功后导入私钥即可看到账户余额。

在以太坊网络中部署智能合约时会消耗 gas，于是 ganache_cli 后台查看到交易的具体信息。本课题中，选择 Truffle 框架来完成智能合约的部署。

项目目录结构如下介绍：

contracts/ 包含项目中所有智能合约的文件夹，涉及到的合约源码文件都放置在这里，还包含一个重要的合约 Migrations.sol，该合约是用来部署 truffle 框架的其他合约。

migrations/ 用来处理部署(迁移)智能合约 ，迁移是一个额外特别的合约用来保存合约的变化。

test/ 智能合约测试用例文件夹。

truffle_config.js 是部署的配置文件，这里要更改区块链网络的节点地址和端口号。

```
development:{
  host: "192.168.88.128",
  port: 8545,
  network_id: "*"
}
```

在 contracs 文件夹中有已经编写好的 solidity 文件，它用来和以太坊虚拟机 EVM 交互来描述智能合约。使用 truffle 可以将 solidity 文件编译成 ABI 字节码的形式，产生

的.json 文件可以在 Web3.js 中作为合约实例来实现合约读取与执行。接下来用 truffle migrate 命令将编译好的智能合约部署到区块链网络中去。

我们编写、部署和测试智能合约，然后为合约编写用户界面，这样合约就可以真正使用。外部接口上的 Javascript 允许控制整个应用程序的 App 对象，init 函数上传智能合同信息并初始化 web3。Web3 是一个与以太节点实现通信的库。我们用 web3 来处理智能合同。

4.6 性能与测试

在前文论述并创建的股权众筹平台中，需要对其运行状况和性能进行指标评定。本节主要制定恰当的测试用例，对股权众筹平台的整体性能-调用区块链智能合约耗时状况进行测试和分析[29]。

一般来说，工程项目的测试环境包含其在使用条件下的硬件环境和软件环境，本课题所使用的测试环境汇总如下表 4.4 所示：

表 4.4 测试环境

硬件环境	Intel Core i5 2430M @ 2.40GHz
软件环境	Ubuntu 16.04 server Truffle v5.0.2 Ganache CLI v6.2.5 (ganache-core: 2.3.3) Node v9.5.0

在智能合约性能测试执行过程中给出如下表 4.5 所示的测试用例，在得到的测试结果中给出用户信息查询、项目信息查询、交易记录查询和部分测试结果。

对测试结果进行分析：

(1) 随着客户接入数量增加，单个节点验证请求、处理交易请求会增大，这样增加了每一笔交易的处理时间。

(2) 终端的请求数量增大对查询执行性能的影响程度比区块链网络验证节点的影响程度大。

针对执行性能的提升，本课题提出了三种解决方案：

(1) 在交易请求服务中引入请求负载均衡服务器，分散请求响应事件。

(2) 对交易数据进行缓存，可以使用 redis 数据库缓冲客户端发送过来的数据，实现快速访问。

表 4.5 测试用例

测试用例 ID	测试用例名称	测试条件	测试过程
T1	用户信息查询	用户已经在系统注册	根据用户 ID 查询表数据
T2	项目信息查询	众筹发起方已经创建或者项目参与者参与的项目存在	根据当前用户的权限级别查询项目表中的数据
T3	交易记录查询	投资者已经投资的项目	根据当前用户的权限级别查询投资表数据
T4	项目审批	用户具备管理员权限	审批激活待审批项目上线

在本课题创建的股权众筹系统中，使用统一的调用 ID 将使用流程中不同阶段的性能反应记录到日志中去。

本系统调用智能合约的方式采用的是 REST API，后期要对后端的 API 调用方法和过程进行优化，简化操作过程、构建分布式集群系统，为系统提供高性能并发访问环境。

结　论

本文首先对众筹这个新型的融资模式进行调查研究，分析出当前国内外该领域发展状况以及遇到的信任问题，在细致体验和思考当前已经存在众筹平台的使用场景后，提出使用区块链技术和智能合约来做出解决方案的思路。本文主要完成了三大任务：一是设计了完整的众筹 Web 平台，根据软件工程理论设计出三层的股权众筹系统架构，包括应用表示层、数据封装层、业务逻辑层，应用表示层负责用户界面交互，数据封装层提供了数据检验、数据分发、数据库系统查询的功能，业务逻辑层介于上述二者之间，负责系统所有功能调度，详细划分为用户模块、项目创建模块等；二是完成了智能合约的设计和具体实现，首先针对业务逻辑层的各个功能进行详细分析和拆分，然后使用以太坊的开发框架和特定领域语言进行封装，最后落地到 DApp 的应用；三是测试了本文设计的平台功能，针对各个测试用例进行分析，包括零输入检验等。

本文使用区块链技术实现众筹平台的功能。缺点包括没有实际意义上的以太坊网络功能。它只使用平台上的虚拟交易功能。目前，众筹在中国仍处于相对不完整的开发阶段。此外，众筹模型可信度的问题无法解决，因此众筹很难解决。区块链技术并不是万金油，但提供了一种分布式共识机制的思路来解决传统交易产生的信任问题，我相信，在不久的将来，基于区块链技术的众筹模式将受到欢迎，也将取得新的突破。

致　　谢

时间飞逝，大学生活即将结束，在这段时间里我学到了很多。我感谢母校给了我一个好的学习和生活环境，我感谢导师给了我一个好的研究环境。我在这里度过了一段难忘的时光。在这段时间里，我得到了导师、同学和家人的帮助和关心。我很感激，我想向他们表达我诚挚的谢意。首先，我要感谢陈海宴老师，他帮助我学习和完成毕业设计。陈老师是一个谦虚、容易相处、博学的老师。在毕业设计过程中，我的老师专注于发展专业实践技能，并鼓励我不断克服项目中的困难。从选择题目、研究、发展到实现论文写作，我们从导师那里得到了严格的指导，这让我不断地思考、以严格的态度和深刻的专业知识展示技术解决方案，并鼓励我大胆迎接新挑战。通过定期的学术研讨会，老师给了我们理解不同领域知识的方法，我们想出了一种全面思考问题和纪律的方法。

感谢实验室里所有的学生，我们在常规项目中进行了讨论和分享，这使我能够不断提高自己的技能，发展团队意识，发展组织和协调技能。生活中的交流和互动增加了学习的乐趣。我感谢我的家人不断的鼓励。在上学的路上，我的家人在物质和精神上都给了我强大的支持。最后，我要感谢所有的评论家对我论文的审查和细致指导。老师的意见和建议对我帮助很大，也帮助我改进了毕业设计，并进行下一步的研究。

参考文献

[1] 周鲜华, 张羽兮, 魏春波. 区块链技术的去中心化众筹平台搭建研究[J]. 会计之友, 2019(01): 148-154.

[2] 魏生, 戴科冕. 基于区块链技术的私募股权众筹平台变革及展望[J]. 广东工业大学学报, 2019, 36(02): 37-46.

[3] 张帅, 延安, 贾敏智. 基于区块链的众筹智能合约设计[J]. 计算机工程与应用, 2019, 55(08): 220-225.

[4] 张建中. 美国地区新闻媒体探索新的商业模式：众筹+区块链[N]. 中国社会科学报, 2018-11-08(003).

[5] 祁健, 戴杨. 区块链技术下的股权众筹融资模式探究[J]. 经济研究导刊, 2018(17): 151-152.

[6] 杨东. 区块链让众筹和共票成为中国原创的制度理论[J]. 金融博览, 2018(10): 36-37.

[7] 宋文鹏, 王振燕. 区块链在众筹平台中的应用[J]. 信息技术与标准化, 2017(03): 28-30.

[8] 黄洁华, 高灵超, 许玉壮, 白晓敏, 胡凯. 众筹区块链上的智能合约设计[J]. 信息安全研究, 2017, 3(03): 211-219.

[9] 陈志东, 董爱强, 孙赫, 胡凯. 基于众筹业务的私有区块链研究[J]. 信息安全研究, 2017, 3(03): 227-236.

[10] 赵大伟. 区块链技术在产品众筹行业的应用研究[J]. 吉林金融研究, 2017(04): 1-5.

[11] 张佳. 2017 区块链新金融论坛暨中关村众筹联盟成立两周年大会在京举行[J]. 中国品牌, 2017(08): 95.

[12] 韩笑天, 马超群. 基于区块链技术的股权众筹模式构建[J]. 商业经济研究, 2017(20): 169-172.

[13] 王凯正. 基于区块链技术的众筹平台的设计与实现[D]. 内蒙古大学, 2017.

[14] 金巍. IP 区块链 众筹 2016 文化金融聚焦 3 课题[N]. 中国出版传媒商报, 2016-11-11(013).

[15] 赵大伟. 区块链能否将奖励众筹送上快车道？[N]. 华夏时报, 2016-11-21(022).

[16] Jodie Moll, Ogan Yigitbasioglu. The role of internet-related technologies in shaping the

work of accountants: New directions for accounting research[J]. The British Accounting Review, 2019.

[17] Lichen Cheng, Jiqiang Liu, Chunhua Su, Kaitai Liang, Guangquan Xu, Wei Wang. Polynomial-based modifiable blockchain structure for removing fraud transactions[J]. Future Generation Computer Systems, 2019, 99.

[18] Vida J. Morkunas, Jeannette Paschen, Edward Boon. How blockchain technologies impact your business model[J]. Business Horizons, 2019, 62(3).

[19] Rakesh Shrestha, Rojeena Bajracharya, Anish P. Shrestha, Seung Yeob Nam. A new-type of blockchain for secure message exchange in VANET[J]. Digital Communications and Networks, 2019.

[20] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, A. S. Albahri, M. A. Alsalem, K. I. Mohammed. Based blockchain-PSO-AES techniques in finger vein biometrics: A novel verification secure framework for patient authentication[J]. Computer Standards & Interfaces, 2019.

[21] Matteo Montecchi, Kirk Plangger, Michael Etter. It’s real, trust me! Establishing supply chain provenance using blockchain[J]. Business Horizons, 2019, 62(3).

[22] Lin Zhong, Qianhong Wu, Jan Xie, Zhenyu Guan, Bo Qin. A secure large-scale instant payment system based on blockchain[J]. Computers & Security, 2019, 84.

[23] Alex Hughes, Andrew Park, Jan Kietzmann, Chris Archer-Brown. Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms[J]. Business Horizons, 2019, 62(3).

[24] Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, Rajkumar Buyya. FogBus: A Blockchain-based Lightweight Framework for Edge and Fog Computing[J]. The Journal of Systems & Software, 2019, 154.

[25] Jannis Angelis, Elias Ribeiro da Silva. Blockchain adoption: A value driver perspective[J]. Business Horizons, 2019, 62(3).

[26] Yinan Li, Wentao Yang, Ping He, Chang Chen, Xiaonan Wang. Design and management of a distributed hybrid energy system through smart contract and blockchain[J]. Applied

Energy, 2019, 248.

[27] Joseph McIsaac, Joseph Brulle, John Burg, Gregory Tarnacki, Christian Sullivan, Rick Wassel. Blockchain Technology for Disaster and Refugee Relief Operations[J]. Prehospital and Disaster Medicine, 2019, 34(s1).

[28] Mari Greenberger. Block what? The unrealized potential of blockchain in healthcare[J]. Nursing Management (Springhouse), 2019, 50(5).

[29] Portland State University; PSU researchers develop blockchain protocol to prevent counterfeit pharmaceutical sales[J]. NewsRx Health & Science, 2019.

附　　录

A Recruitment and Human Resource Management Technique Using Blockchain Technology for Industry 4.0

The Blockchain technology is considered as one of the major catalysts transforming the next generation of industry structure. This digital ledger technology is integrating the computational and physical components of the smart industry by providing a fair, cost-effective, accurate, traceable and secure system. However, current industries are facing tremendous pressure to maintain a tenable human resource management system. Since the world is now on its way to adopt another industrial revolution (industry 5.0), industries of all level are in need of a smart employee hiring and management system. Internet-based and computer-aided human resource management system has already been a popular way of hiring and employee evaluation. Gascó in 2004, described, using a case-study conducted at Telefonica which is a renowned telecommunication company in Spain, how the use of Information Technology (IT) can enhance the overall performance of Human Resources Management (HRM) systems. However, the authenticity of human resource information directly affects the motivation, rate and efficiency of human resource management. Job seekers may hide unpleasant results and present fraudulent information during the recruitment process. Some job applicants submit inflated resume supplementing with fake training and diploma certificates, references, awards, promotions and so forth whereas others deliberately exaggerate their qualifications and abilities. A recent (2017) survey by business review Australia identified that a $250,000 monetary loss to the hiring company can happen yearly due to a bad hiring of an employee on an annual salary $100,000. In a study by Robert Half conducted amongst Human Resource (HR) Managers in Australia identified that an unsuccessful hiring in an organization may negatively affect productivity (55%), lower staff morale (23%), and cause financial loss (19%). As a result, organizations end up paying a huge amount of cost just to get rid of bad hiring. For example, Amazon pays employees $5,000 bonus for cancelling a job contract. Another survey reveals that 74 percent of employers had wrong hiring. It was also identified that 45 percent worker's

skills did not match with their claim and 33 percent had lied about their qualifications. Survey founds, 10 percent out of 2,257 hiring managers stated that they do not have adequate tools to find the right person to hire. Associated Press news stated, India faces a very high rate of employment through fake certificates stating one single state had identified total 1832 such cases in the year 2017. On the contrary, a trustable, middlemen less, independent and transparent recruitment with efficient HRM systems are a precondition for the successful implementation of Fourth Industrial Revolution (Industry 4.0), or even preparation for the Fifth (Industry 5.0). Blockchain technology possesses significant potentials to eliminate these problems. Blockchain technology has already taken off in several industries because of the fact that it can offer a smart, well- connected, selforganized, transparent, decentralized and immutable system. In this study, we propose a fast, efficient, transparent recruitment and human resource management system using Blockchain (BC) to reduce the risk faced by human resource authority. The proposed system, thereby, provides authentic and effective decision support information for the human resource management of an organization.

中文译文:

工业 4.0 使用区块链技术的招聘和人力资源管理技术

区块链技术被认为是改变下一代产业结构的主要催化剂之一。这种数字分类帐技术通过提供公平，经济，准确，可追溯和安全的系统，集成了智能行业的计算和物理组件。然而，当前的行业正面临着维持稳定的人力资源管理系统的巨大压力。由于世界正在走向另一场工业革命(工业 5.0)，各级工业都需要一个聪明的员工招聘和管理系统。基于互联网和计算机辅助的人力资源管理系统已经成为招聘和员工评估的流行方式。 Gascó在2004 年使用在西班牙着名电信公司 Telefonica 进行的案例研究描述了如何使用信息技术(IT)来提高人力资源管理(HRM)系统的整体绩效。然而，人力资源信息的真实性直接影响着人力资源管理的动力，速度和效率。求职者可能会在招聘过程中隐藏不愉快的结果并提供欺诈性信息。一些求职者提交夸大的简历，补充假培训和文凭证书，参考，奖励，促销等等，而其他人故意夸大他们的资格和能力。澳大利亚商业评论最近(2017 年)的一项调查发现，由于员工年薪 10 万美元的雇佣情况不佳，招聘公司每年可能发生 250,000 美元的金钱损失。在 Robert Half [4]在澳大利亚人力资源(HR)经理中进行的一项研究中发现，在一个组织中招聘不成功可能会对生产力产生负面影响(55％)，员工士气降低(23％)，并导致经济损失(19 ％)。结果，组织最终只是为了摆脱糟糕的招聘而支付了大量的费用。例如，亚马逊为取消工作合同向员工支付 5,000 美元奖金。另一项调查显示，74％的雇主雇用错误。还发现，45％的工人技能与他们的索赔不符，33％的人对他们的资格撒谎。调查发现，2,257 名招聘经理中有 10％表示他们没有足够的工具来找到合适的人选。美联社的新闻称，印度通过假证书面临着非常高的就业率，表明一个州在 2017 年确定了 1832 个此类案件。相反，一个可信赖的，中间人较少，独立和透明的招聘与高效的人力资源管理系统是成功实施第四次工业革命(工业 4.0)的前提条件，甚至是第五次工业革命的准备工作(工业 5.0)。区块链技术具有消除这些问题的巨大潜力。区块链技术已经在几个行业中起飞，因为它可以提供智能，连接良好，自组织，透明，分散和不可变的系统。在本研究中，我们提出了一种快速，高效，透明的招聘和人力资源管理系统，使用区块链(BC)来降低人力资源管理部门面临的风险。因此，所提出的系统为组织的人力资源管理提供真实有效的决策支持信息。

```
pragma solidity >=0.5.0;


contract PlayerToFundings{
// 参与者=> 合约地址数组
mapping(address => address[]) playerToFundings;

function join(address player, address fundingAddress) public {
playerToFundings[player].push(fundingAddress);
}

function getPlayerFundings(address player) public view returns(address[] memory){
return playerToFundings[player];
}

}


contract FundingFactory {

//存储所有已经部署的智能合约的地址
address[] public fundings;

// 创建者=> 合约地址数组
mapping(address => address[]) creatorToFundings;

PlayerToFundings p2f;
constructor() public{
PlayerToFundings p2fAddress = new PlayerToFundings();
p2f = PlayerToFundings(p2fAddress);
```

```
27  }
28
29  function createFunding(string memory _projectName, uint _goalMoney) public
    returns(address[] memory){
30  Funding funding = new Funding(_projectName, _goalMoney, msg.sender, p2f);
31  fundings.push(address(funding));
32
33  // 把创建者创建的合约地址保存到其数组中
34  creatorToFundings[msg.sender].push(address(funding));
35  return fundings;
36  }
37
38  // 路人 查看所有众筹项目列表
39  function getFundings() public view returns(address[] memory){
40  return fundings;
41  }
42
43  // 创建者 查看调用者创建的所有众筹项目地址
44  function getCreatorFundings() public view returns(address[] memory){
45  return creatorToFundings[msg.sender];
46  }
47
48  // 参与者
49  function getPlayerFundings() public view returns(address[] memory){
50  return p2f.getPlayerFundings(msg.sender);
51  }
52  }
```

```
53  contract Funding {
54
55  // 众筹成功标记
56  bool public flag = false;
57  // 众筹发起人地址(众筹发起人)
58  address public manager;
59  // 项目名称
60  string public projectName;
61  // 众筹参与人需要付的钱
62  uint public supportMoney;
63  // 目标募集的资金(endTime 后,达不到目标则众筹失败)
64  uint public goalMoney;
65  // 默认众筹结束的时间,为众筹发起后的一个月
66  uint public endTime;
67  // 众筹参与人的数组
68  address[] public players;
69  mapping(address=>bool) playersMap;
70  //众筹参与人 amount 记录
71  mapping(address=>uint) playersRecord;
72
73  PlayerToFundings p2f;
74
75  // 已投票的用户地址
76  mapping(address=>bool) votedMap;
77  uint votedCount = 0;
78
79  //构造函数
80  constructor(string memory _projectName, uint _goalMoney, address sender,
```

```
    PlayerToFundings _p2f) public{
81  manager = sender;
82  projectName =_projectName;
83  goalMoney = _goalMoney;
84  endTime = now + 4 weeks;
85  p2f = _p2f;
86  }

87  // 我要支持(需要付钱)
88  function support() public payable {
89  //require(msg.value == supportMoney);
90  // 检查是否符合要求
91  // 防止一个人重复此操作
92  require(!votedMap[msg.sender]);
93  //标记已经付款
94  votedMap[msg.sender] = true;
95  votedCount++;
96  // 放进集合中
97  players.push(msg.sender);
98  playersMap[msg.sender] = true;
99  playersRecord[msg.sender] = msg.value;
100 p2f.join(msg.sender, address(this));
101 }
102
103 // 所有参与者
104 function getPlayers() public view returns(address[] memory){
105 return players;
106 }
```

```
107 //  所有参与者个数
108 function getPlayersCount() public view returns(uint){
109 return players.length;
110 }
111 //  获取合约的余额
112 function getTotalBalance() public view returns(uint){
113 return address(this).balance;
114 }
115 //  获取剩余天数
116 function getRemainDays() public view returns(uint) {
117 return (endTime - now) / 60 / 60 / 24;
118 }
119 //  检查众筹状态
120 function checkStatus() public onlyManagerCanCall {
121 require(!flag);
122
123 //  到期
124 require(now >= endTime);
125 // getTotalBalance 金额>= goalMoney
126 require(address(this).balance >= goalMoney);
127
128 flag = true;
129 }
130
131 modifier onlyManagerCanCall {
132 require(msg.sender == manager);
133 _;
134 }
```

```
135 }
```