智能合约在供应链信息共享中的应用*[1]

刘亦琛[1],张珺[1]

[1](内蒙古大学 计算机学院,内蒙古 呼和浩特 010021)

通讯作者: 刘亦琛, E-mail: mlyichen@163.com

摘 要 当前在供应链运行中仍存在很多问题，比如参与方间信息共享程度低、交互不方便、产品生产流通信息易缺失等等，区块链具有的信息真实记录、不可篡改、可追溯的特性能让这些问题得到很好的解决。本文基于以太坊区块链提出一套用于供应链信息共享的智能合约业务逻辑方法，对供应链信息平台的运行方式和流程进行了设计与描述，最后在本地测试环境对智能合约部分功能进行了验证。

关键词: 以太坊;智能合约;供应链;信息共享平台

Application of Smart Contracts in Infomation Sharing in Supply Chain

LIU Yi Chen[1],ZHANG Jun[1]

[1](School of Computer Science; Inner Mongolia University, Hohhot 010021, China)

Abstract At present, there are still many problems in the operation of the supply chain, such as low level of information sharing and inconvenient interaction among participants,easy loss of product production and circulation information, etc. The blockchain has real information record, non-tamperable and traceable characteristics, which can make these problems be solved very well. Based on the Ethereum blockchain, this paper proposes a set of smart contracts logic methods for supply chain information sharing. The design and description of the operation mode and process of the supply chain information platform are carried out. Finally, part of smart contracts' function is implemented in the local test environment.

Key words: Ethereum; smart contract; supply chain; plantform of information sharing

供应链是指围绕核心企业，从原材料开始，制成中间产品以及最终产品，最后由销售网络把产品送到消费者手中的、将供应商，制造商，分销商直到最终用户连成一个整体的功能网链结构。很多企业都已经开始探索供应链管理的优化，也开发出一些供企业使用的供应链管理系统。但是，供应链目前在运行中仍然存在很多问题，比如由参与方多导致的信息难及时共享、信息透明度低，由流程复杂、环节众多导致的参与方间协同困难、产品信息追溯困难等。这些问题在传统的供应链解决方案下还是不能很好解决，因此有必要寻找新的思路[1]。

区块链是以加密货币的发明为根源而被发掘出的一项技术[2]，以密码学、P2P 技术[3]等为基础，形成了多方共识、数据不可篡改、便于追溯的核心特性。这些特征恰与供应链运作管理中的需求与特征相吻合，因而被认为是解决供应链管理难题的关键技术出路[4]。

智能合约[5]的概念早在 20 世纪 90 年代就由尼克萨博提出，他认为智能合约是一套以数字形式定义的承诺，包括合约参与方可以在计算机系统上执行这些承诺的协议。但是由于在传统计算机中，难以解决的信任问题，所以智能合约无法真正被使用。而区块链的出现，恰好为智能合约的运行提供了可信的设施，所以智能合约适合在区块链上运行。

结合了智能合约与区块链特征的以太坊[6]是进行去中心化应用开发的平台，以太坊不仅提供了智能合约的运行环境，还为开发者提供了友好的面向 JavaScript 的接口方法，因此本文选择基于以太坊进行相关业务的设计和开发。

目前国内外对于区块链在供应链中的应用都做了一定的研究和实践，比较有代表性的项目有 Everledger 的钻石来源鉴定[7]，沃尔玛超市的猪肉跟踪[8]，马士基的物流区块链[9]，京东的商品溯源等[10]。这些项目都积极探索了区块链的供应链方案落地实施，为我们研究区块链在供应链领域的应用提供了借鉴。但是，直接在以太坊上设计相关业务的案例还比较少[11][12]，也基本限于追溯方面。因此本文将在以太坊上探索实现一系列供应链中需要的功能。本文的创新点在于设计了一种基于以太坊的联盟链的运行方式，并将其运用在供应链情景上；针对供应链运作中各类参与方的需求设计编写了智能合约，很好地利用了以太坊中智能合约访问区块链的便利性和编程时的灵活性，达到对供应链运作优化的目的。

1 基于区块链的供应链信息共享平台体系

1 基金项目：国家自然科学基金（61772502，61672499）

1.1 运行方式

该平台将基于以太坊搭建联盟链，供应链上参与方即区块链的各个维护节点。另有管理员角色，负责智能合约的发布、系统的维护、地址的分配等，同时参与区块链共识。管理者角色可由监管部门承担。在该体系下，从原材料供应商、加工商到物流、零售商，供应链运行的每一步相关信息都将被记录在区块链上，消费者或者任何该平台的用户可以方便地追查到这些信息。区块链的不可篡改性保障了信息的真实可信，同样也为监管或追责提供了便利。同时，将业务逻辑在区块链上实现，也让所有行为都以交易的形式被记录在区块链上，能在一定程度上监督参与者的行为，减少失信的发生。

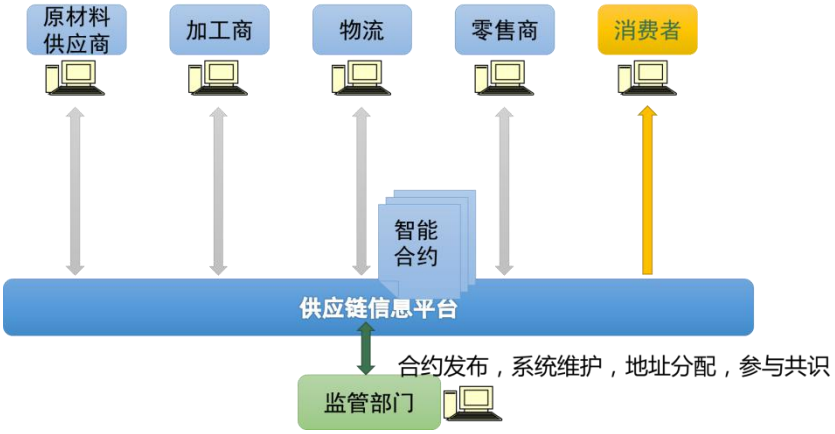

Fig.1 Schematic diagram of supply chain information platform operation

图 1 供应链信息共享平台运行示意图

管理者在平台运行前就将区块链地址进行分配，让企业和其区块链地址唯一绑定，这样地址的行为就代表了企业的行为，同时通过对特定区块链地址进行权限分配就实现了对相应用户的权限控制。

原材料供应商、加工商、物流、零售商分别登记各自的基本信息，并登记产品流经本环节时的各项特征参数以及对产品所做的处理。如产品名称、数量、操作环境条件、操作情况等。这些信息通过智能合约存储在区块链上，消费者要对一件商品进行追溯时，从区块链读出之前所登记的信息即可。

参与方之间的业务往来也基于区块链完成。区块链通过智能合约记录订货活动的每一步，跟踪订单生成的全过程。同时，通过设计智能合约的内置逻辑能帮助实施参与方的信用评估，形成良好的循环机制，有利于供应链的健康发展。

1.2 平台架构安排

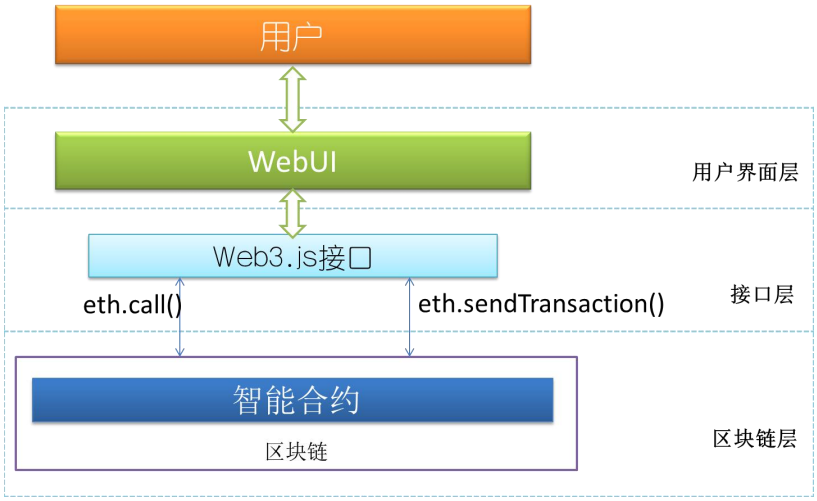

Fig.2 Architecture diagram of Information sharing platform for supply chain

图 2. 供应链信息共享平台架构图

如图 2 所示，该平台由用户界面层、接口层及区块链层构成。用户通过平台提供 web 操作界面，调用编写的区块链接口，访问区块链上的智能合约函数，完成对区块链数据的读写。读区块链使用 eth.call（消息调用）方式访问区块链，写使用 eth.sendtransaction（发送交易）方式访问。

2 信息共享关键业务设计

该平台中的关键业务有：在线订货，信息登记，信息查询等。

2.1 在线订货

订货是供应链中上下游企业间最常进行的活动，在该平台中，订货流程设计为发布、查询、接单、选择并最终生成订单几步。

2.1.1 发布者发布订货需求

在供应链中，有货物的需求者，那么它就是订货方。订货方可以在平台上发布其订货需求，包括需要的货物名称、数量等。订货需求数据结构如下：

```
struct orderReq{
    string supplierName;//供应商名称
    string shipmentName;//货物名称
    uint amount;//货物数量
    uint expTime;.//需求到期时间
    uint isProcessed;//当前订货需求是否已被确认过，是为 1，否为 0
    receiveRec[] receiveRecs;//当前订货需求的接单记录
    ...//其他信息
}
orderReq[] orReqs;
```

在该数据结构中，包含了一个到期时间 expTime，它是用来控制每一条需求的可操作性的。如果一条需求已经过了有效期，那么是不能被执行后续操作如接单等的。这种设计增强了智能合约自动化处理效率，方便了用户的使用。

2.1.2 供应商查询订货需求

供应商可以在平台上查询已被发布的订货需求信息。查询分为按照需求商名称查询及按照货物名称查询，使用者可以任选一种展开查询。以按货物名称查询为例，伪代码如下：

```
function searchByShipmentName (string _shipmentName) returns (string, uint, uint,){
    for(uint p=0;p<orReqs.length;p++){
            if(sha3(orReqs[p].shipmentName) ==sha3( _shipmentName))
            return (orReqs[p].supplierName, orReqs[p].amount, orReqs[p].exptime);
        }
}
```

通过比较输入的货物名称条件与已存在的订货需求中的货物名称的 sha3 值，当匹配时，返回该订货需求的供应商名称、订货数量及到期时间等关键信息。

2.1.3 供应商接单

接单记录数据结构：

```
struct receiveRec{
    orderReq orderreqOfReceiRec;//接单记录对应的订货需求
    uint price;//货物售价
    uint arriveTime;//承诺到货时间
}
```

只有在当前需求未被确认并且未过期的前提下，供货者才能对一个需求进行接单。判定代码：

```
require(orderreqOfReceiRec.isProcessed==0&&orderreqOfReceiRec.expTime>now);
```

同时，接单操作需要提交自己货物的售价及能提供货物的时间等信息，将来一旦被选择，所提交的信息都会被写进订单。而承诺时间会作为是否成功发货的考量指标之一。

```
function receiveOrder(uint _price, uint _arriveTime){
require(orderreqOfReceiRec.isProcessed==0&&orderreqOfReceiRec.expTime>now);
    /*
    接单记录结构体赋值
    */
}
```

2.1.4 发布者确认并生成订单

发布者在个人中心可以查看每一条已发布的需求的接单情况，然后通过比较接单者的情况，选择最合适的合作供应商进行确认。一旦确认，双方订单即生成，建立契约关系。同时其他接单者的接单记录变为无效，请求自动失效。建立的订单将对供货方产生责任约束，督促其完成承诺。

最终生成的订单结构体：

```
struct order{
    string shipmentName;//预定货物名称
    uint shipmentAmount;.//预定货物数量
    uint arriveTime;//承诺送达时间
    ...//其他信息
}
```

订单约定了发货的内容、数量、到货时间等，作为自动付款合约触发时用来匹配比较的条件。

2.1.5 发布者删除订货需求

当不再需要一项订货需求，发布者可以删除该需求。删除操作的执行前提条件是，当前执行者（交易发送者）是该需求的发布者，同时无人接单。判断条件：

```
require((msg.sender==当前需求.发布者)&&(当前需求.接单记录数组.length==0));
```

2. 2 信息登记

信息登记分为企业基本信息登记、环节信息登记、收发货登记三类。

2.2.1 基本信息登记

基本信息登记即是使用该平台的企业，首先需要在平台上注册自己的企业信息。包括企业名称、企业标识编码、企业所在地、企业联系方式、企业简介等内容。基本信息的登记是为了让企业之间能互相把握基本情况，为供应链决策提供参考。数据结构设计：

```
struct supplierInfo{
    string name;//企业名称
    uint id;//企业标识编码
    string place;//企业所在地
    uint phoneNumber;//企业联系方式
    string introduction;//企业介绍
    uint reputation;//企业信誉度值
}
mapping (address=>supplierInfo) supInfo;//由企业地址到其信息结构体的映射
```

2.2.2 环节信息登记

各阶段负责人通过准确收集所需信息并完成录入实现环节信息的登记，每个角色只能对自己所在的环节的信息进行登记。利用 solidity 语言的自定义修饰符（modifier）特性实现。如为只有生产商可以使用的功能（生产环节信息登记）加上权限管理：

```
produceAddr=0x...;//生产商的区块链地址，唯一,由系统管理员分配
//定义了 onlyProducer 修饰符，被该修饰符修饰的函数只能由 produceAddr 这个地址调用
modifier onlyProducer {
require (msg.sender==produceAddr);//只有调用者是 produceAddr 才能执行此函数
}
```

给待执行函数加上 onlyProducer 修饰符：

```
function setProduceInfo onlyProducer {
    /*
    环节信息结构体一一赋值;
    */
}
```

环节信息登记时，同时将记录每一步处理的时间戳信息。

2.2.3 收货发货登记

这类登记指的是供应链上的企业在收到上游发来的货物或者将货物发给下游时需要执行的登记动作。收发货登记需要录入的是

货物的名称、数量、当前所在位置等等。

我们安排了一种判断发货成功和发货失败的认定规则，即认为当收货方收到货物的信息与发货方承诺发送的货物信息一致时，认为发货成功，否则发货失败。发货成功会为发货方增加一定的信誉度值，而失败则会减低发货企业的信誉度值。信誉度将来又会被其他企业在选择合作方时作为参考，所以每一个企业都会努力维护自身的信誉度。这个信誉度值修改的过程由可信的不受人为干预的智能合约完成，所以保障了信息的真实可信。信誉度评估策略也能更好地督促企业诚信经营、积极维护自身信誉。

在该类登记中，数据的录入会触发信誉度值管理合约以及货款交付合约。以收货登记为例，具体实现过程如下及关键判断规则如下：

```
function setReceiveInfo(uint _amount,address_supplier ){
  if((shipment.amount==_amount)&&(now<shipment.arriveTime)){//当发货成功
  supInfo[_supplier].reputation+=1;//发货方信誉度增加;
  supplier.transfer(_price) //执行自动付款活动，向 supplier 转账
  }else{
     //减低发货方信誉度值
     if(supInfo[_supplier].reputation-=1>=0)
          supInfo[_supplier].reputation-=1;//减低发货方信誉度值
          else {
          supInfo[_supplier].reputation=0;//信誉度最低为 0
          emit ShipmentFailed(shipment.name);//触发一个事件 ShipmentFailed，通知用户发货失败
          }
  }
}
```

2.3 信息查询

信息的查询包括对企业基本信息的查询、对供应链环节上信息的查询和每个登录用户对自己账户余额变动的查询。

企业基本信息查询：企业可以对其他企业的基本信息进行查询，包括名称、代码、所在地、联系方式、经营范围以及在登记环节被更新的信誉度值等。企业间信息互查可以增强企业之间的相互了解和促进信息透明化。

供应链环节信息查询（追溯）：消费者或任何想要追溯一个产品的生产流通全过程的人通过该平台的追溯功能可以实现对产品进行信息追溯。用户只需输入产品编号，就可以查询到该产品从原材料产地到最后消费者手中的全过程全记录信息。这些信息都是经过所有区块链节点验证后打包存储在链上的，因而没有经过篡改，具备真实可信的基础。查询除的信息可以作为监管追责的依据。

余额变动记录查询：这项功能的完成基于当进行资金转移时就要记录每一笔资金转移的来源和去处，这样才能在将来需要查询时有依据可以遵循。具体实现方法为：

记录每一次执行转账操作的消息发送者（msg.sender），并放入当前资金转入方（supplier）的个人余额信息结构体（accountBalance）中:

```
struct accountBalance{
      address from;//记录转账资金来源
      uint balance;//余额
};
mapping (address =>accountBalance) personalAccoBalance;//每个账户从其地址到其余额信息结构体的映射
function transfer (address to, uint value){//转账函数
      personalAccoBalance[to].from=msg.sender;//本次余额增加的来源账户，即来自哪个地址
      personalAccoBalance[to].balance+=value;//给余额增加值
}
```

2.4 平台内资金流通机制设计

在该平台中的自动付款环节涉及到账户间的转账操作，由于以太坊联盟链中的以太币不具有实际价值，故又专门在该平台设计了一种用于完成支付活动的通证。本系统的资金流通被设计为基于该内置通证的账户间余额转移。平台所使用的通证最初将来源于平台管理员的发放。在初始化时，账户余额被写入每一个账户的属性中，在发生资金流转时，对相应账户属性值进行修改，完成从付款方到收款方的“转账”操作。

3 功能测试

以需求的发布为例，对上述功能进行基于 Remix 的测试。

（1）部署智能合约

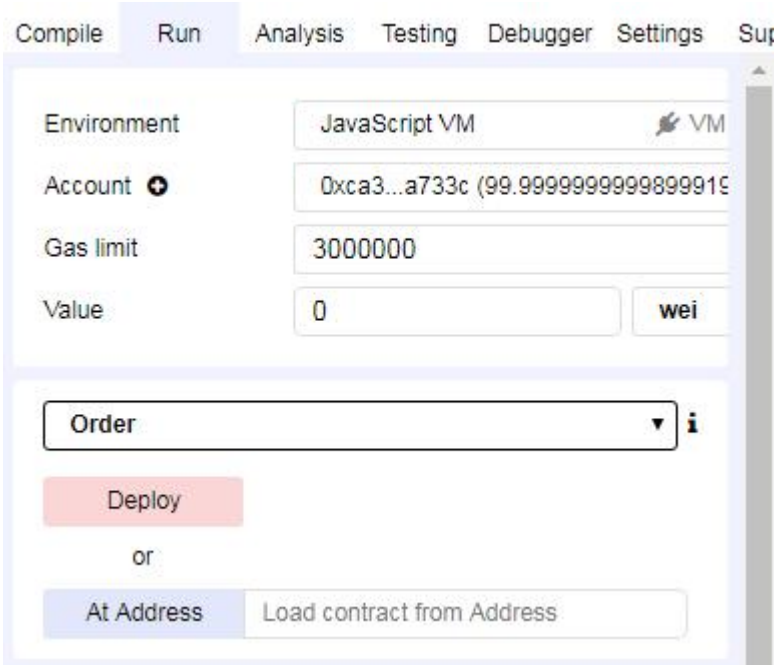

Fig.3 deploy the smart contract in Remix

图 3 在 Remix 中部署智能合约

如图 3 所示，在 Remix 中对订货（Order）智能合约进行部署，点击 Deploy 按钮，控制台输出图 4 所示信息，表明部署成功。

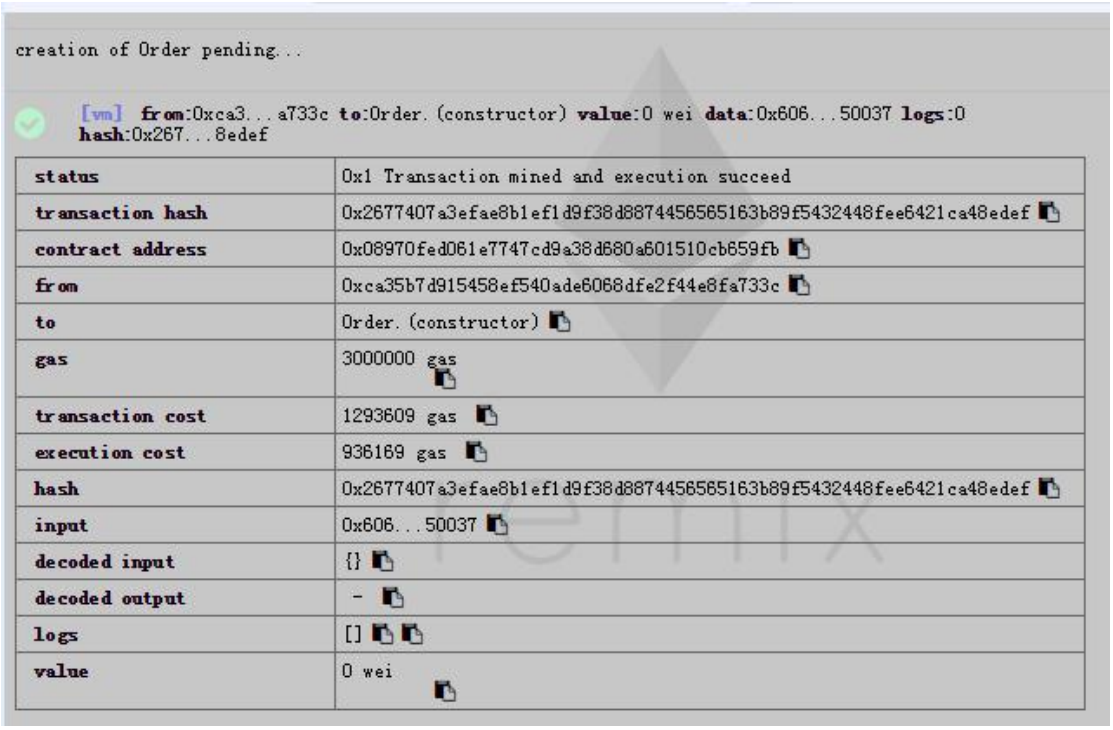

Fig.4 deploy contract *Order* successfully

图 4 Order 合约部署成功

（2）输入函数所需参数，发送交易

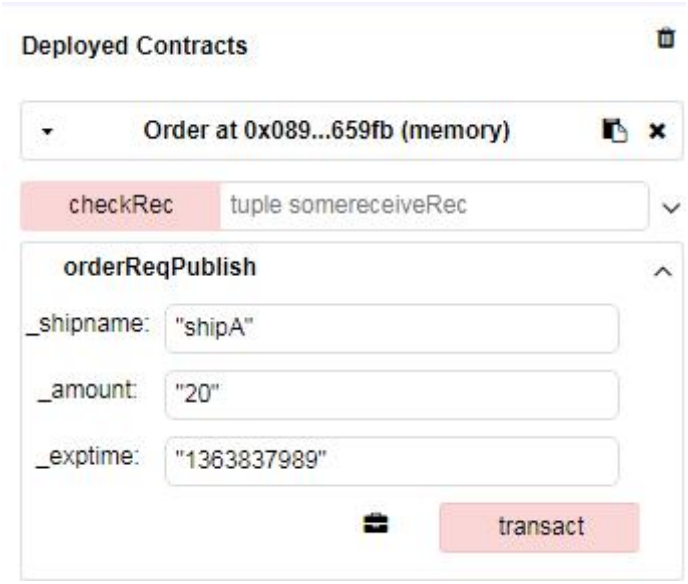

Fig.5 input the parameters for tested function

图 5 为测试函数输入参数

如图 5 所示，为需求发布函数输入三个参数，点击“transact”。这里将商品名称命名为“shipA”，数量为 20，过期时间是一个毫秒数“1363837989”。

（3）交易发送成功，控制台以日志形式输出 event 捕获的信息。

Fig.6 function excuted successfully and the event output the logs

图 6 函数执行成功，事件输出日志

在此我们捕获了商品名称_shipname，商品数量_amount 以及过期时间_exptime。_exptime 以 UTC 时间表示，将来在前端代码中会转换成易读的本地时间。

4 总结与展望

供应链管理与运行优化一直是备受关注的话题。本文介绍了供应链的现状、区块链的特征及智能合约的相关概念，针对供应链管理的优化设计了一个供应链信息共享平台，提出了平台的架构及实现策略。同时，设计了一套用于供应链信息共享的业务流程方法，包括建立订单、信息登记和信息追溯等，并基于以太坊进行了程序开发和实现。最后在 Solidity 在线集成开发环境 Remix 下对部分功能进行了测试。

本设计还有很多可以继续改进之处，如在登记环节可尝试结合物联网技术实现，这样既可以保证信息的真实性又可以保证信息的及时性。同时，信息在录入区块链之前可以采取一定的加密手段做预处理，以在一定程度上保护部分隐私或敏感数据。另外，对于企业的信誉控制方法可以进一步完善，对于未正常履行订单协议的参与方，为其设计一定的惩罚机制等。

References:

[1] Baruffaldi G, Sternberg H. Chains in Chains-Logic and Challenges of Blockchains in Supply Chains[J]. 2018.

[2] Swan M. Blockchain: Blueprint for a New Economy. O’Reilly Media, Inc., 2015.

[3] Pu H E, Ge Y U, Zhang Y F, et al. Survey on Blockchain Technology and Its Application Prospect[J]. Computer Science, 2017 (in Chinese with English abstract).

[4] The Supply Chain Has No Clothes: Technology Adoption of Blockchain for Supply Chain Transparency.[J]. Logistics, 2018, 2(1): 2.

[5] Nick Szabo. The idea of smart contracts.http://szabo:best:vwh:net/smart contracts idea:html,1997.

[6] .Ethereum Foundation. Ethereum's white paper. https://github:com/ethereum/wiki/wiki/White-Paper,2015.

[7] Lomas N. Everledger is using blockchain to combat fraud, starting with diamonds[J]. URL: https://techcrunch. com/2015/06/29/everledger, 2015.

[8] Higgins S. Walmart: Blockchain food tracking test results are ‘very encouraging'[J]. 2017.

[9] Groenfeldt T. IBM and Maersk apply blockchain to container shipping[J]. URL: https://www. forbes. com/sites/tomgroenfeldt/2017/03/05/ibm-and-maersk-apply-blockchain-to-container-shipping, 2017.

[10] Jingdong Blockchain. URL:.http://ledger.jd.com/.

[11] Ye X R,Qing SHAO,Rong XIAO. Supply Chain Prototype System Based on Blockchain, Smart Contracts and Internet of Things[J].science &technology review,2017, 35(23):62-69.(in Chinese)

[12] Bocek T, Rodrigues B B, Strasser T, et al. Blockchains everywhere - a use-case of blockchains in the pharma supply-chain[C]// Integrated Network and Service Management. IEEE, 2017:772-777.

附中文参考文献

[3] 何蒲,于戈,张岩峰,鲍玉斌.区块链技术与应用前瞻综述[J].计算机科学,2017,44(04):1-7+15

[11] 叶小榕, 邵晴, 肖蓉. 基于区块链, 智能合约和物联网的供应链原型系统[J]. 科技导报, 2017, 35(23): 62-69.