

中山大学数据科学与计算机学院本科生实验报告

(2019 年秋季学期)

课程名称：区块链原理与技术

任课教师：郑子彬

年级	2017	专业 (方向)	软件工程
学号	17343128	姓名	幸赞
电话	13246843092	Email	312618502@qq.com
开始日期	2019.11.11	完成日期	2019.12.13

一、 项目背景

基于已有的开源区块链系统 FISCO-BCOS

(<https://github.com/FISCO-BCOS/FISCO-BCOS>), 以联盟链为主, 开发基于区块链或区块链智能合约的供应链金融平台, 实现供应链应收账款资产的溯源、流转。

传统供应链金融：某车企（宝马）因为其造车技术特别牛，消费者口碑好，所以其在同行业中占据绝对优势地位。因此，在金融机构（银行）对该车企的信用评级将很高，认为他有很大的风险承担的能力。在某次交易中，该车企从轮胎公司购买了一批轮胎，但由于资金暂时短缺向轮胎公司签订了 **1000 万**的应收账款单据，承诺 **1 年**后归还轮胎公司 **1000 万**。这个过程可以拉上金融机构例如银行来对这笔交易作见证，确认这笔交易的真实性。

在接下里的几个月里，轮胎公司因为资金短缺需要融资，这个时候它可以凭借跟某车企签订的应收账款单据向金融结构借款，金融机构认可该车企（核心企业）的还款能力，因此愿意借款给轮胎公司。但是，这样的信任关系并不会往下游传递。在某个交易中，轮胎公司从轮毂公司购买了一批轮毂，但由于租金暂时短缺向轮胎公司签订了 **500 万**的应收账款单据，承诺 **1 年**后归还轮胎公司 **500 万**。当轮毂公司想利用这个应收账款单据向金融机构借款融资的时候，金融机构因为不认可轮胎公司的还款能力，需要对轮胎公司进行详细的信用分析以评估其还款能力同时验证应收账款单据的真实性，才能决定是否借款给轮毂公司。这个过程将增加很多经济成本，而这个问题主要是由于该车企的信用无法在整个供应链中传递以及交易信息不透明化所导致的。

二、 方案设计

区块链+供应链金融：

将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

实现功能：

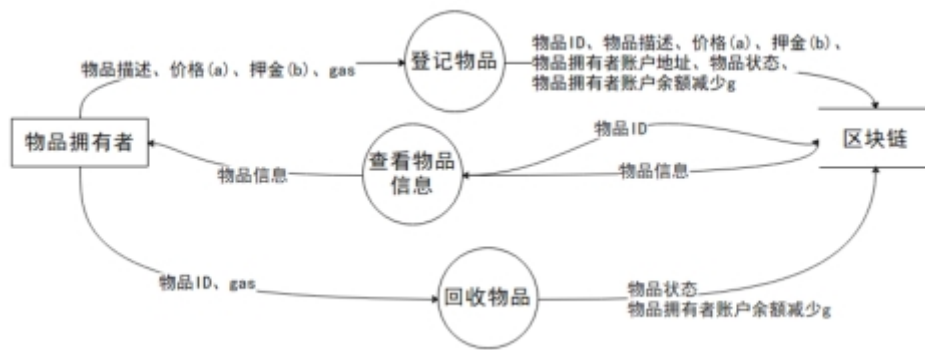
功能一：实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

功能二：实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

功能三：利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

功能四：应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

数据流图示例：



代码分析：主要利用了 4 个函数来分别实现所要求的 4 个功能，其中用银行来部署条约，设计了两个 map 分别查看收据的情况和每个账号的余额。然后一个账单是有拥有者 (come)，钱借给了谁 (to)，收据编号 (number)，收据的金额 (mount)，通过收据的转让或者结算来实现整个系统的运作，同时使各个用户的金额发生改变。

三、 功能测试

先启动 **webase**：

```

fisco-bcos@fiscobcos-VirtualBox:~$ cd webase-deploy/
fisco-bcos@fiscobcos-VirtualBox:~/webase-deploy$ python deploy.py installAll
=====
               W E B A S E
            \  \  \  \  \  \  \
=====
===== envrionment check... =====
check git...
check finished sucessfully.
check openssl...
check finished sucessfully.
check curl...
check finished sucessfully.
check nginx...
check finished sucessfully.
check java...
check finished sucessfully.
check FISCO-BCOS node port...
check finished sucessfully.
check WeBASE-Web port...
check finished sucessfully.
check WeBASE-Node-Manager port...
check finished sucessfully.
check WeBASE-Front port...
check finished sucessfully.
check database connection...
check finished sucessfully.
===== envrionment ready... =====
=====
===== deploy start... =====
=====
===== FISCO-BCOS install... =====
FISCO-BCOS节点目录nodes已经存在。是否重新安装？[y/n]:n
===== FISCO-BCOS start... =====
try to start node0
try to start node1
node1 start successfully
node0 start successfully

```

启动成功：

```

try to start node1
node1 start successfully
node0 start successfully
===== FISCO-BCOS end... =====
=====
===== WeBASE-Web install... =====
webase-web.zip编译包已经存在。是否重新下载？[y/n]:n
webase-web.zip编译包已经解压。是否重新解压？[y/n]:n
===== WeBASE-Web start... =====
[sudo] password for fisco-bcos:
===== WeBASE-Web start success! =====
===== WeBASE-Web end... =====
=====
===== WeBASE-Node-Manager install... =====
webase-node-mgr.zip编译包已经存在。是否重新下载？[y/n]:n
webase-node-mgr.zip编译包已经解压。是否重新解压？[y/n]:n
WeBASE-Node-Manager数据库webasenodemanager已经存在，是否删除重建？[y/n]:n
是否初始化数据(首次部署或重建库需执行)？[y/n]:n
===== WeBASE-Node-Manager start... =====
===== WeBASE-Node-Manager start success! =====
===== WeBASE-Node-Manager end... =====
=====
===== WeBASE-Front install... =====
webase-front.zip编译包已经存在。是否重新下载？[y/n]:n
webase-front.zip编译包已经解压。是否重新解压？[y/n]:n
WeBASE-Front数据库webasefront已经存在，是否删除重建？[y/n]:n
===== WeBASE-Front start... =====
===== WeBASE-Front start success! =====
===== WeBASE-Front end... =====
=====
===== deploy end... =====
===== version v1.2.0 =====
=====

```

然后输入 **localhost : 5000** 就可以进入 webase 网页了



可以看到一些部署过的合约

新增用户		新增用户并生成公钥地址			
用户名称	用户ID	用户描述	用户公钥地址信息	用户状态	操作
Metal	 700008		 0xaa186f0f174...	正常	修改
Lu Yi	 700007		 0x4b4473c1dd...	正常	修改
Tyre	 700006		 0x1c23a38458...	正常	修改
Bank	 700005		 0x328088543b...	正常	修改
Alice	 700004		 0xc7f45d59ec...	正常	修改
Is	 700003		 0x8566286d7c...	正常	修改
Bob	 700002		 0xf67969138cf...	正常	修改
共 7 条 10条/页 < 1 > 前往 1 页					

然后便开始实现功能：

功能一：实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

```
function IssueReceivables(address receive,uint amount,uint num){
    balances[msg.sender]-=amount;
    balances[receive]+=amount;
    receipts[num].come=msg.sender;
    receipts[num].to=receive;
    receipts[num].mount=amount;
    receipts[num].number=num;
}
```

发送交易

合约名称: Myproject4

合约地址:

用户:

方法:

参数:

receive	d6c04b58b15636a
amount	100

轮胎借给轮毂 100 块，收据编号为 0

交易内容

×



0x1c23a3845883559d24d6cb197154fd53ccbe5f5

0x4b4473c1dd6b45f39ab865104d6c04b58b15636a

100

100

100

100

Copy

查看轮毂的 balance

发送交易

×

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11 ⓘ

方法: function balances

参数: 865104d6c04b58b15636a

ⓘ 如果参数类型是数组, 请用逗号分隔, 不需要加上引号, 例如: array1,array2. string等其他类型也不用加上引号。

取消

确定



功能一实现。

功能二：实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

```
function TransferOfReceivables(uint one, uint two){  
    if(receipts[one].come!=msg.sender)  
        return;  
    if(receipts[one].to==receipts[two].come){  
        receipts[one].mount-=receipts[two].mount;  
        receipts[two].come=receipts[one].come;  
    }  
}
```

例：轮胎借给了轮毂 100，轮毂借给金属公司 50，则将收据转换为轮胎借给轮毂 50，轮胎借给金属公司 50

发送交易

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11

用户: Lu Yi

方法: function IssueReceiv

参数:

receiver	0c0b9a88c1e311ba
amount	50
token	1

❗ 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如: array1,array2, string等其他类型也不用加上引号。

取消 确定

查看之前的两个收据:

交易内容

copy

```
[
  "",
  0,
  "0x1c23a38458835559d24d6cb197154fd53ccbe5f5",
  "0x4b4473c1dd6b45f39ab865104d6c04b58b15636a",
  100
]
```

交易内容



1

0x4b4473c1dd6b45f89ab865104d6c04b58b15636a

0xaa186f0f174dfcc9140d01e8ec0b9a88c1e311ba

50



进行收据转让：

交易内容


合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11 

用户: Tyre 

方法: function  TransferOfR 

参数:	one	0
	two	1

 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如：array1,array2。string等其他类型也不用加上引号。

取消

确定

转让完后查看收据

交易内容

```
[{"id": "0", "data": [{"id": "0x1c23a38458835559d24d6cb197154fd53ccbe5f5", "value": "0x4b4473c1dd6b45f39ab865104d6c04b58b15636a"}], "status": "50"}]
```

copy

交易内容

```
[{"id": "1", "data": [{"id": "0x1c23a38458835559d24d6cb197154fd53ccbe5f5", "value": "0xaa186f0f174dfcc9149d01e8ec0b9a88c1e311ba"}], "status": "50"}]
```

copy

两个收据都成了轮胎公司借出的，转让完成，功能二实现。

功能三： 利用应收账款向银行融资上链，供应链上所有可以利用应

收账款单据向银行申请融资。

以上两张收据都是轮胎公司持有，所以轮胎公司可以向银行融资（因为是基础版，就没有加入银行信用鉴定，之后会做）。

```
function FinancingFromBank(uint own){
    if(receipts[own].come!=msg.sender||receipts[own].mount<=0)
        return;
    balances[msg.sender] += receipts[own].mount;
    receipts[own].come=bank;
}
```

用第一张收据向银行提钱：

发送交易

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11

用户: Tyre

方法: function FinancingFr

参数: own 0

❗ 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如：array1,array2。string等其他类型也不用加上引号。

取消 确定

发送交易

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11

方法: function balances

参数: 4d6cb197154fd53ccbe5f5

❗ 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如：array1,array2。string等其他类型也不用加上引号。

取消 确定

查询轮胎公司的钱

交易内容



COPY

为-50（因为开始借了 100 出去，现在融资融回来 50）。

融资功能实现。

功能四：应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款。

```
function AccountsSettlement(uint own){
    if(receipts[own].come!=msg.sender||receipts[own].mount<=0||receipts[own].mount>balances[receipts[own].to])
        return;
    balances[msg.sender] += receipts[own].mount;
    balances[receipts[own].to] -= receipts[own].mount;
    receipts[own].mount = 0;
}
```

支付欠款：以收据 1 为例，金属公司欠轮胎公司 50

轮胎公司收钱

[illegible]

Mypro
Mypro
Mypro
Mypro
Mypro
Mypro
Mypro
Ballot
Asset
HelloV

发送交易

×

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11 ⓘ

方法: function balances

参数: 4d6cb197154fd53ccbe5f5

ⓘ 如果参数类型是数组, 请用逗号分隔, 不需要加上引号, 例如: array1,array2. string等其他类型也不用加上引号。

取消 确定

mpang
s;
s;

s,ui

查询轮胎公司的钱:

Mypro
Mypro
Mypro
Mypro
Mypro
Mypro
Mypro
Ballot
Asset
HelloV

发送交易

×

合约名称: Myproject4

合约地址: 0x64e007e8c5375d9933142fd11 ⓘ

方法: function balances

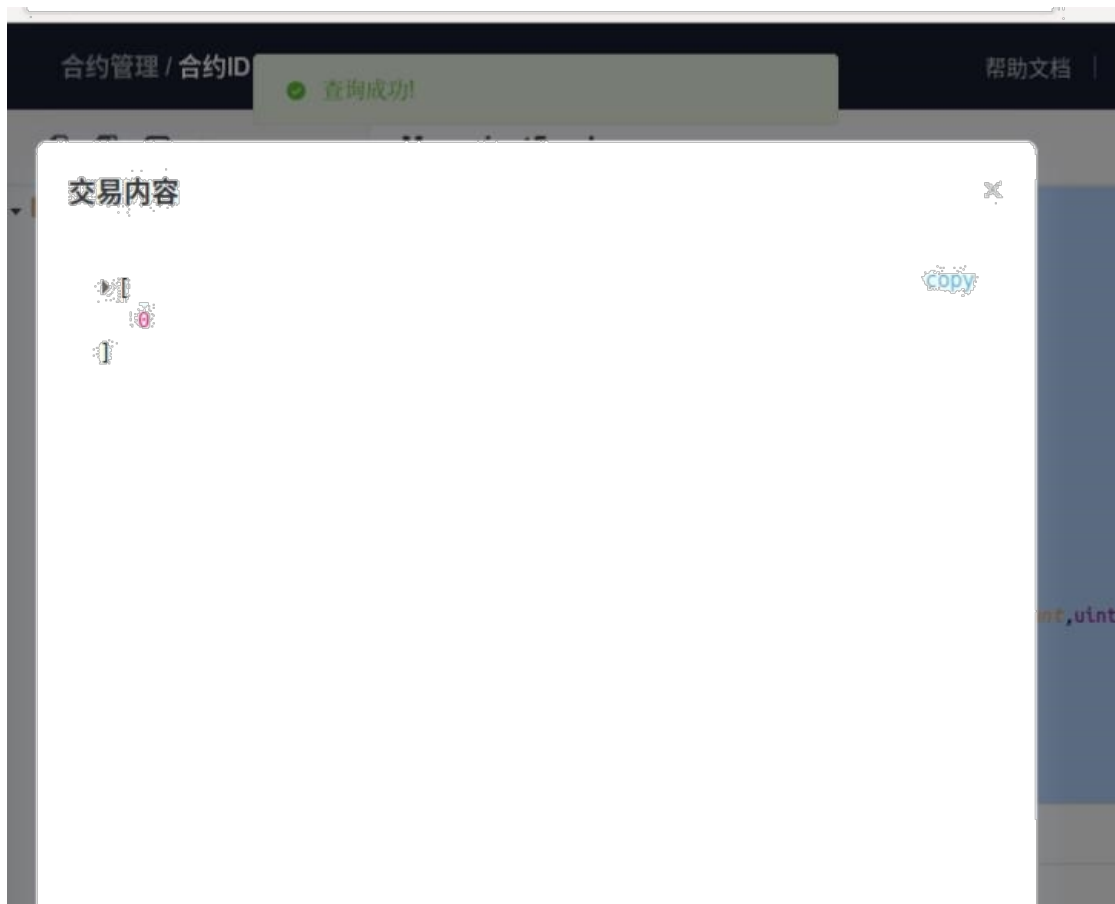
参数: 4d6cb197154fd53ccbe5f5

ⓘ 如果参数类型是数组, 请用逗号分隔, 不需要加上引号, 例如: array1,array2. string等其他类型也不用加上引号。

取消 确定

mpang
s;
s;

s,ui



为 0，因为刚才为-50，现在收回来了

然后查看两个收据数据：

发送交易

合约名称: Myproject4

合约地址: ⓘ

方法:

function

receipts

参数:

ⓘ 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如：array1,array2。string等其他类型也不用加上引号。

取消

确定

发送交易

×

合约名称: Myproject4

合约地址:

0x64e007e8c5375d9933142fd11

?

方法:

function

receipts

参数:

1

?

 如果参数类型是数组, 请用逗号分隔, 不需要加上引号, 例如: array1,array2。string等其他类型也不用加上引号。

取消

确定

交易内容

```
[{"to": "0x328088543b65be2ea0513d0af35fa6687d1e6a08", "value": "50"}, {"to": "0x4b4473c1dd6b45f39ab865104d6c04b58b15636a", "value": "0"}]
```

收据 0 是向银行融资的, 所以发起者改为了银行, 然后还没收回。



收据 1 欠款已经结清，所以金额变为了 0

还款结清功能实现。

四、 界面展示

界面因为没有自己写前端，所以就在 webase 上跑的代码。在代码测试的一步已经展示了界面，就是 webase 的运行界面。

五、 心得体会

本次大作业其实是想做前端和后端的，但是由于能力有限，看了 java 和 nodejs 的 SDK 都没办法看懂，试着操作了一下也只停留在勉强用 nodejs 的 SDK 来调用 getblocknumber 或者 getversion 等函数，用 deploy 部署之后可以成功，用 call 调用合约的一些函数也都是 ok 的，只是返回的地址无法解码，试了很久都没办法得到具体的值，所以就放弃了用 nodejs 来做前后端，只是借用 webase 的平台来勉强当个界面部署调用合约，但整个的链还是在 fisco-bcos 上部署的。通过这次作业，让我对区块链的认识比以前更深了，也更清楚了区块链的运作方式以及 POW 共识机制，也希望我在以后的学习中能收获更多的知识！

