

# 区块链大作业：基于区块链地供应链金融平台

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级	专业（方向）	软件工程
学号	17343141	姓名	姚东烨
电话	13246859092	Email	<a href="mailto:894816193@qq.com">894816193@qq.com</a>
开始日期	2019.10.25	完成日期	2019.12.13

## 一、项目背景

传统供应链金融中，供应链上传递的交易信息是不透明的。导致处于应付账款传递的下游企业融资十分困难。因为金融机构在决定是否同意该项融资时，需要对该下游企业进行详细的信用分析以评估其还款能力，并且也要验证该下游企业所持有应付账款的真实性，这个过程将增加很多经济成本。

出现上述这个问题是由于车企的信用信息无法在供应链上传递以及交易信息的不透明化导致的。所以我们选择将供应链金融与区块链结合，通过区块链的交易信息透明化来解决这个问题。

## 二、方案设计

### • 存储设计

#### ◦ 应付账款信息：

使用合约存储，包含5个成员变量：

address fr: 标记该应付账款的付款方  
address to: 标记该应付账款的收款方  
uint money: 标记该应付账款的金额  
uint DDL: 标记该应付账款的收款期限  
bool finished: 标记该应付账款是否完结

```
contract Loan{
    address fr;
    address to;
    uint money;
    uint DDL;
    bool finished;
    //get set....
}
```

**备注：**为5个成员变量提供适当的 **get**、**set** 函数，方便外部使用。

#### ◦ 用户资产信息

使用 **mapping (address => uint)** 进行存储

```
mapping (address => uint) private balances;
```

- 用户信用等级信息

使用 **mapping (address => uint)** 进行存储

```
mapping (address => uint) public grade;
```

- 转账信息存储

使用 **event** 存储

```
event sent(address from, address to, uint amount);
```

- 应付账款信息存储（未完成）

使用 **event** 存储

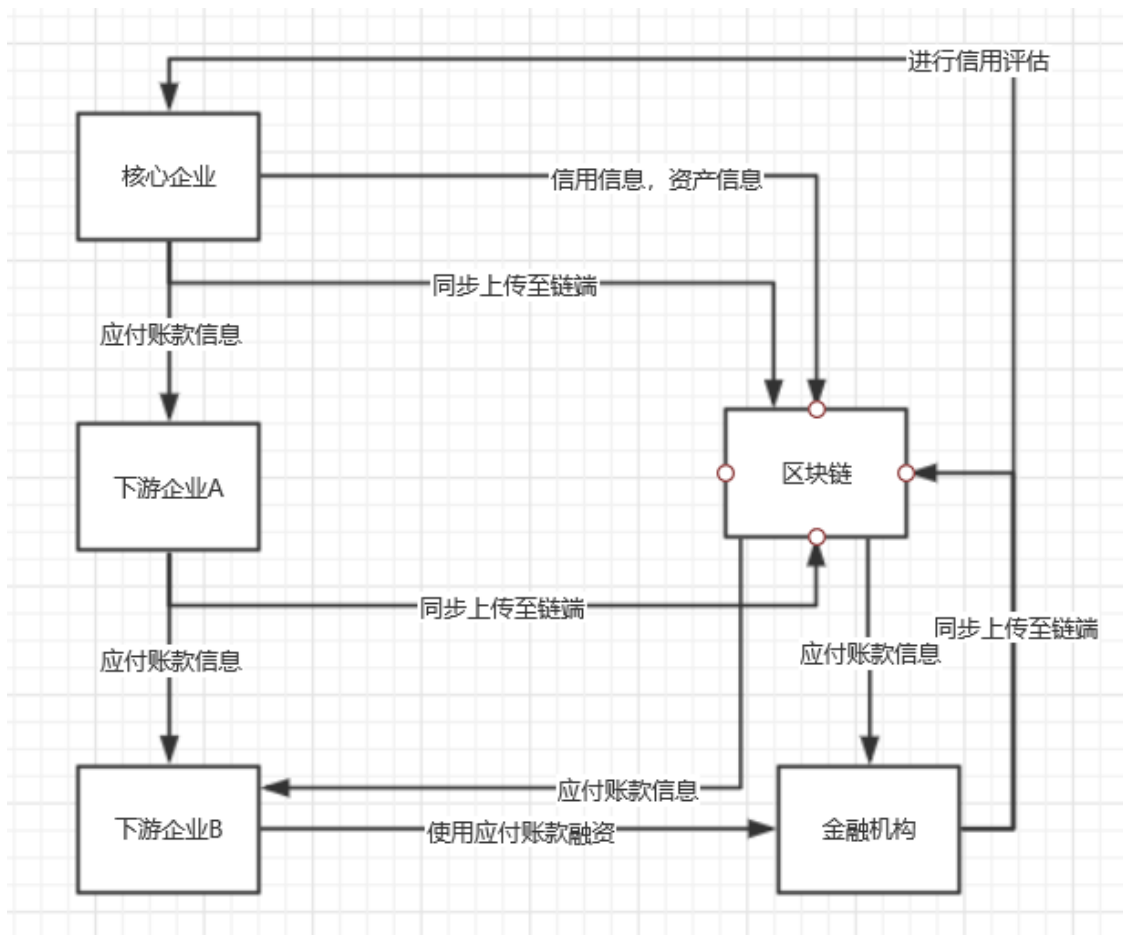
```
event loanRecord(address from, address to, uint amount, uint DDL, bool  
finished);
```

- 应付账款信息存储（已完成）

使用 **event** 存储

```
event finishedRecord(address loanAddress);
```

- 数据流图



- 核心功能

- Get Balance

使用私钥查询余额

```
function getBalance() public returns(uint) {
    return balances[msg.sender];
}
```

- Issue (Bank)

银行专属功能，给各企业发钱

```
function issue(address receiver, uint amount) public {
    if (msg.sender != bank) return;
    balances[receiver] += amount;
}
```

- Evaluate (Bank)

银行专属功能，对各企业进行信用评估，进行评估的企业需要花费20块钱，但是这时使用应付账款的前提

```
function evaluate(address companyAddress) public {
    if (msg.sender != bank) return;
    balances[bank] += 20;
    balances[companyAddress] -= 20;
    grade[companyAddress] = 50;
}
```

- **Transfer (By cash)**

通过现金转账，提供收款方地址以及金额即可，但必须保证余额充足。

```
function sendCash(address receiver, uint amount) public {
    if (balances[msg.sender] < amount) return;
    balances[msg.sender] -= amount;
    balances[receiver] += amount;
    emit sent(msg.sender, receiver, amount);
}
```

- **Transfer (By loan)**

通过应付账款支付，需提供收款方地址，以及金额，需保证账户进行过信用评估，且信用等级高于50

```
function sendLoan(address receiver, uint amount, uint time) public
returns(address) {
    if (grade[msg.sender] < 50) return address(0x0);
    Loan loan = new Loan(msg.sender, receiver, amount, now+time);
    emit loanRecord(msg.sender, receiver, amount, now+time, false);
    return loan;
}
```

- **Transfer (By other's loan)**

通过部分转让应付账款进行支付，需要应付账款地址、收款方地址，以及转让金额。另外需要保证：转让金额不大于应付账款金额，原应付账款收款方是自己，原应付账款付款方信用等级大于50，原应付账款未完结且未过期

返回的是交给新收款方的应付账款地址，原应付账款地址不变，金额改变。

```
function sendOthersLoan(address loanAddress,address receiver,uint
amount) returns(address){
    if(Loan(loanAddress).getFinished()==true) return address(0x0);
    if(Loan(loanAddress).getTo()!=msg.sender) return address(0x0);
    if(grade[Loan(loanAddress).getFrom()]<50) return address(0x0);
    if(Loan(loanAddress).getDDL(<now) return address(0x0);
    if(Loan(loanAddress).getMoney(<amount) return address(0x0);
    Loan send=new
Loan(Loan(loanAddress).getFrom(),receiver,amount,Loan(loanAddress).getD
DL());
    emit
loanRecord(Loan(loanAddress).getFrom(),receiver,amount,Loan(loanAddress
).getDDL(),false);
    if(Loan(loanAddress).getMoney(>amount){
        Loan(loanAddress).setMoney(Loan(loanAddress).getMoney()-amount);
    }
    return send;
}
```

- **Bank Financing**

向金融机构融资，需保证应付账款收款方是自己，且应付账款未完结

```
function loan(address loanAddress) public{
    if(Loan(loanAddress).getFinished()==true) return;
    if(Loan(loanAddress).getTo()!=msg.sender) return;
    Loan(loanAddress).setFinished();
    emit finishedRecord(loanAddress);
    Loan loan=new
Loan(Loan(loanAddress).getFrom(),bank,Loan(loanAddress).getMoney(),Loan
(loanAddress).getDDL());
    emit
loanRecord(Loan(loanAddress).getFrom(),bank,Loan(loanAddress).getMoney(
),Loan(loanAddress).getDDL(),false);
    balances[msg.sender]+=Loan(loanAddress).getMoney();
}
```

- **Pay Loan**

为应付账款支付，使用私钥以及应付账款地址即可，同样需要保证余额充足。

```
function payLoan(address loanAddress) public{
    if(Loan(loanAddress).getMoney()<=balances[msg.sender]){
        balances[msg.sender]-=Loan(loanAddress).getMoney();
        balances[Loan(loanAddress).getTo()]+=Loan(loanAddress).getMoney();
        Loan(loanAddress).setFinished();
        emit finishedRecord(loanAddress);
    }
}
```

### 三、功能测试

- **Get Balance**

## 查询余额

Get Balance

PrivateKey:

40f43c02a236016346a0d

Confirm

余额为0x96块钱

[illegible]

- **Issue (Bank)**

给BMW发100块钱

Issue (Bank)

To:

00a5a0f69de719191a304

Amount:

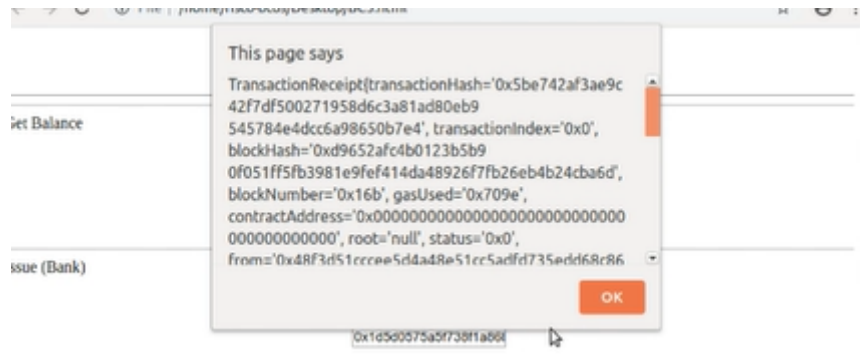
100

PrivateKey:

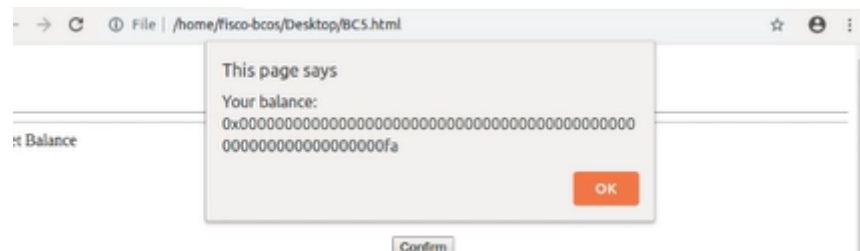
96e63cd79b4fd536612e04

Confirm

获得交易回执



查询余额为0xfa块钱

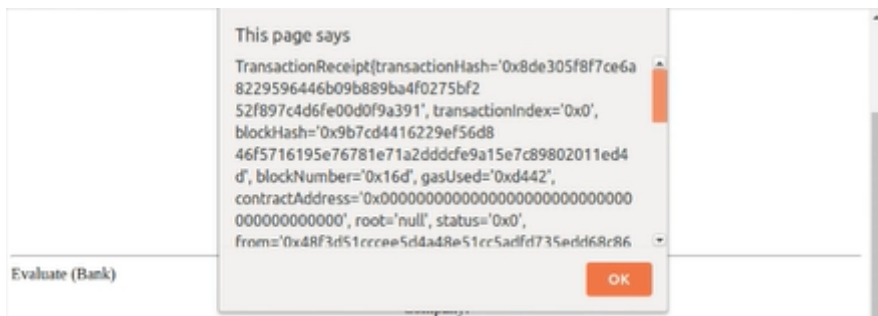


- Evaluate (Bank)

金融机构评估信用等级



获得回执

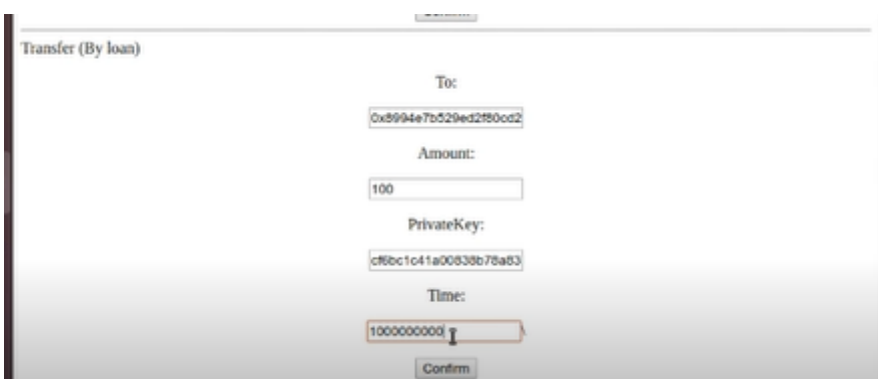


- Transfer (By cash)

与教程中的功能相同，这里不再测试

- Transfer (By loan)

发送Loan



获得应付账款地址







OurBank

File | /home/fisco-bcos/Desktop/BC5.html

OurBank

Get Balance

PrivateKey:

Confirm

Issue (Bank)

To:

Amount:

PrivateKey:

Confirm

Evaluate (Bank)

Company:

PrivateKey:

Confirm

## 五、心得体会

之前听师兄师姐说，我们的区块链课程是一门导论课程，就是介绍一些区块链的知识，同学们自己通过课堂理解，读一些论文，再查询一下就可以写完读后感结课了。但是这学期我们这个课程与微众银行合作，多了这个期末大作业，我一开始知道的时候是比较害怕的，因为自己在之前的课程中都没有独立开发过一个应用，前端后端也不是了解的很清楚，更别说这次还要与链端交互了。所以我认为这次大作业不仅仅考察了我们区块链的知识，还更多地考察了我们实践动手能力。事实上，我从很早就开始了大作业的完成，因为自己啥都不懂，所以肯定要笨鸟先飞。我一开始处理的就是后端与链端交互的问题，因为前端我打算使用的是JavaScript，所以后端我也想用js来写。所以我一开始就使用了Node.js的SDK进行研究。当我跟着教学网站做到最底的时候，我发现我得到的只是一个CLI，我并不知道要怎么设计一个后端程序来模拟或者说控制这个CLI来根据前端发来的请求与链端做对应的交互。无奈之下，只好选择了我们实训刚复习过的Java SDK来进行后端处理。Java SDK似乎更适合初学者，当我通过一系列环境配置以及初始化后，得到了一个Web3j对象后，我就可以愉快地与链端交互了，之后所要做的就是逻辑的设计，这部分并不难。因为我们在大作业的第二阶段就把要用的sol合约写好了，只需要尽情地调用即可。可是大作业的关卡并不是在这里就结束了，当我处理前端与后端交互的时候，我又遇到了难缠的跨域问题（似乎只有Chrome有这个问题），我首先尝试了使用servlet，但是当我把应用部署到Tomcat后，与链端的交互又开不了了，因为自己十分不熟悉Tomcat的工作原理，所以换用了Java自带的HttpServer进行请求处理，最终解决了问题。中间由于一些课程DDL的靠近，所以拖了一些区块链作业的时间，当我开始设计我的前端时，时间已经没有太多了，所以我在菜鸟教程以及W3S上学了一些基础的JavaScript知识后，就拼凑出现在的简陋前端页面，如果能预留更多时间给区块链作业，那么也不至于只是使用几个输入框就组成前端吧，这也是本次作业的比较大的遗憾。

总而言之，在本次大作业中，我不仅对区块链的知识以及工作原理进行了更深一步的了解，还通过这次大作业丰富了自己的应用开发经历，算是收获良多。希望自己在以后能将更多的区块链知识运用在工作学习中，毕竟日常的需要透明化的地方还有很多。最后，谢谢TA和老师们的指导，才帮助我一步一步将大作业完成。