

# 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	17 级	专业（方向）	软件工程
学号	17343108	姓名	王然
电话	13246820667	Email	923206599@qq.com
开始日期	2019.12.10	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，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。

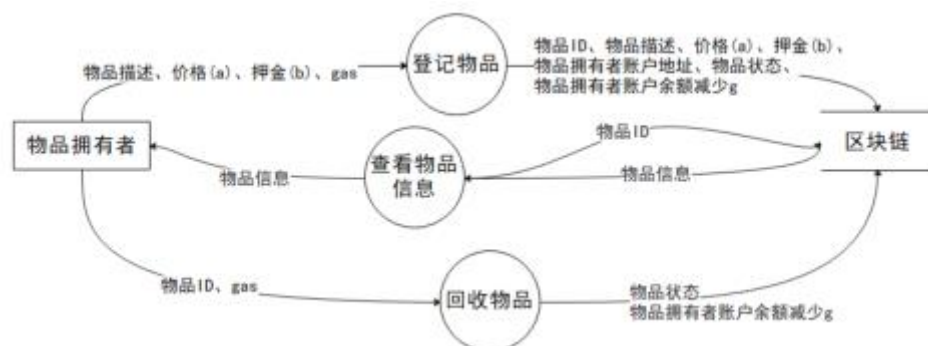
区块链+供应链金融：

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。

## 二、 方案设计

存储设计、数据流图、核心功能介绍（文字+代码）

数据流图示例：



合约内部使用一个 Receipt 结构体来记录收据，每个收据为结构体 Receipt 储存收据的金额、收据编号、收据的发出方（借方）、收据的接收方（贷方）。结构体 Receipt 如下：

```
struct Receipt
{
    uint amount;
    uint number;
    address c; //come
    address t; //to
}
```

核心功能：

### 1. 实现采购商品一签发应收账款

```
function PurchaseGoodsIssueReceivables(address receive, uint _amount, uint _number)
{
    balance[msg.sender] -= _amount;
    balance[receive] += _amount;
    receipt[_number].amount = _amount;
    receipt[_number].number = _number;
    receipt[_number].c = msg.sender;
    receipt[_number].t = receive;
}
```

该功能将用于记录一方向另一方借钱后提供的收据信息。

### 2. 实现应收账款的转让

```
function TransferReceivables(uint x, uint y)
{
    if(receipt[x].c != msg.sender)
    {
        return;
    }
    if(receipt[y].c == receipt[x].t)
    {
        receipt[x].amount -= receipt[y].amount;
        receipt[y].c = receipt[x].c;
    }
}
```

当收据 1 为 A 公司借给 B 公司 m 元，收据 2 为 B 公司借给 C 公司 n 元（m>n）时，该功能将使得收据 1 变为 B 公司欠 A 公司（m-n）元，收据 2 变为 C 公司欠 A 公司 n 元。

### 3. 利用应收账款向银行融资

```
function UseReceivablesToFinanceBank(uint x)
{
    if(receipt[x].amount <= 0 || receipt[x].c != msg.sender)
    {
        return;
    }
    balance[msg.sender] += receipt[x].amount;
    receipt[x].c=bank;
}
```

当某公司要向银行融资时，更新资产余额。

#### 4.应收账款支付结算

```
function PaymentSettlementOfAccountsReceivables(uint x)
{
    if(receipt[x].amount <= 0 || receipt[x].c != msg.sender || receipt[x].amount > balance[receipt[x].t])
    {
        return;
    }
    balance[msg.sender] += receipt[x].amount;
    balance[receipt[x].t] -= receipt[x].amount;
    receipt[x].amount = 0;
}
```

当收据中的欠款方进行债务偿还后，对双方的资产进行更新。

### 三、功能测试

在 VirtualBox 中启动 webase

先输入指令 `cd webase-deploy/`，进入 `webase-deploy` 文件中。接下来执行 `python deploy.py installAll` 指令，以启动 webase。

```
~$ cd webase-deploy/
~/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
  node0 start successfully
  node1 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 =====
=====
```

可以看到 webase 已经成功启动。  
输入 localhost:5000 进入 webase 的网站。  
四个私钥用户的创建如下：

bank	700012	0x07634a10d9581d6e518cfeabda06e...	正常	修改
CarCompany	700011	0x618e8fabb4c051d643ccbaba5c172...	正常	修改
HubPlant	700010	0x82cc33f41b61e338b1ce09505fc5f1...	正常	修改
TireFactory	700009	0x9897202367750979f3b0c2bd44bcd...	正常	修改

在网站中部署合约 project.sol，部署好合约之后开始分别实现四个功能。

功能一：实现采购商品—签发应收账款交易上链。

轮胎厂向车企借 100：

发送交易

合约名称: project

合约地址: 0xf68ba26e3a44ff451101f745539 ⓘ

用户: TireFactory ▾

方法: function ▾ PurchaseGo ▾

参数: 

receive	0x618e8fabb4c051c
_amount	100
_number	1



合约地址:	0xf68ba26e3a44ff451101f745539	
用户:	CarCompany	
方法:	function	PurchaseGo
参数:	receive	0x82cc33f41b61e3c
	_amount	50
	_number	4

[illegible]

```
▶ [
  50,
  4,
  "0x618e8fabb4c051d643ccbaba5c172dd9bdc6820e",
  "0x82cc33f41b61e338b1ce09505fc5f1877446a1b5"
]
```

- 7 -



合约地址:	0xf68ba26e3a44ff451101f745539		
用户:	TireFactory		
方法:	function	TransferRec	
参数:	x	1	
	y	4	

```
{  
    transactionHash: "0xa5ec14b627fc583dffa5722c727df9b6523852c053f7b46e2e7a41f18364117  
        0"  
    ,transactionIndex: 0  
    ,blockHash: "0x9fe2c6c6aa482f927fad73cf3dd621c5da51b458bc4cd9e5f7b8ea485d4bd342"  
    ,blockNumber: 85  
    ,gasUsed: 34319  
    ,contractAddress: "0x0000000000000000000000000000000000"  
    ,root: null  
    ,status: "0x0"  
    ,from: "0x9897202367750979f3b0c2bdd4bcd66acba82020"  
    ,to: "0xf68ba26e3a44ff451101f74553994cd6da885e8d"  
    ,input: "0xfc56aa3f00000000000000000000000000000000000000000000000000000000000000  
        0100000000000000000000000000000000000000000000000000000000000000000000004"  
    ,output: "0x"  
    ,logs: []  
    ,logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000  
        000000000000000000000000000000000000000000000000000000000000000000000000"  
    ,blockNumberRaw: "0x55"  
    ,transactionIndexRaw: "0x0"  
    ,gasUsedRaw: "0x860f"  
    ,statusOK: true  
}
```

```
▶ [
  50,
  1,
  "0x9897202367750979f3b0c2bdd4bcd66acba82020",
  "0x618e8fabb4c051d643ccbaba5c172dd9bdc6820e"
]
```



```
[
  50,
  4,
  "0x9897202367750979f3b0c2bdd4bcd66acba82020",
  "0x82cc33f41b61e338b1ce09505fc5f1877446a1b5"
]
```

## 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。

### 对车企进行融资：

合约名称: project

合约地址:	0xf68ba26e3a44ff451101f745539		
用户:	CarCompany		
方法:	function	UseReceival	
参数:	x	1	

交易内容:

```
{  
  transactionHash: "0x85c18be7ea35b1fa7340841e285857fbeb49f91186beb29fe4a2102277940  
79"  
  transactionIndex: 0  
  blockHash: "0xc7140d79ac551c79fd5154f7e333d7476e86a7fba79994d22cf48578199c0fbf"  
  blockNumber: 87  
  gasUsed: 22341  
  contractAddress: "0x0000000000000000000000000000000000"  
  root: null  
  status: "0x0"  
  from: "0x618e8fab4c051d643ccbaba5c172dd9bdc6820e"  
  to: "0xf68ba26e3a44ff451101f74553994cd6da885e8d"  
  input: "0x2783c9420000000000000000000000000000000000000000000000000000000000  
001"  
  output: "0x"  
  logs: []  
  logsBloom: "0x000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000"  
  blockNumberRaw: "0x57"  
  transactionIndexRaw: "0x0"  
  gasUsedRaw: "0x5745"  
  statusOK: true  
}
```

查询车企的账户：

合约名称: project

合约地址:	0xf68ba26e3a44ff451101f745539	
方法:	function	balance
参数:	0x618e8fabb4c051d643ccl	
▶ [		
0		
]		

车企原本借出 50，现在融资回来 50 一共为 0，功能三实现。

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。

在收据 1 中，轮胎厂欠车企 50

合约名称: project

合约地址:	0xf68ba26e3a44ff451101f745539	
用户:	CarCompany	
方法:	function	PaymentSet
参数:	x	1

然后对车企的账户进行查询：

合约名称: project

合约地址:	0xf68ba26e3a44ff451101f745539	
方法:	function	balance
参数:	0x618e8fabb4c051d643ccl	

发现欠的钱已经还清了：

```
▶ [
  0
]
```

功能四实现。

#### 四、 心得体会

这次作业主要就是对区块链的应用进行实践，发现了区块链原来有这样意想不到的作用，在我们的生活中与我们时刻紧密相连、密不可分。区块链的存在优化了我们的生活，为我们的生活提供了更多的便利。区块链的未来目前来看是很光明的，是有着无限的未来的，而本学期的课程也让我对区块链有了一个最基本的认识，收获良多。

Github 地址：<https://github.com/GoldBear98/chain>