

# 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	17343091	专业（方向）	软件工程
学号	17343091	姓名	牛凌宇
电话	15013027809	Email	1058818647@qq.com
开始日期	12.09	完成日期	12.13

### 一、 项目背景

中小型企业融资难、融资贵是一直以来我国金融发展存在的问题。

#### 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：

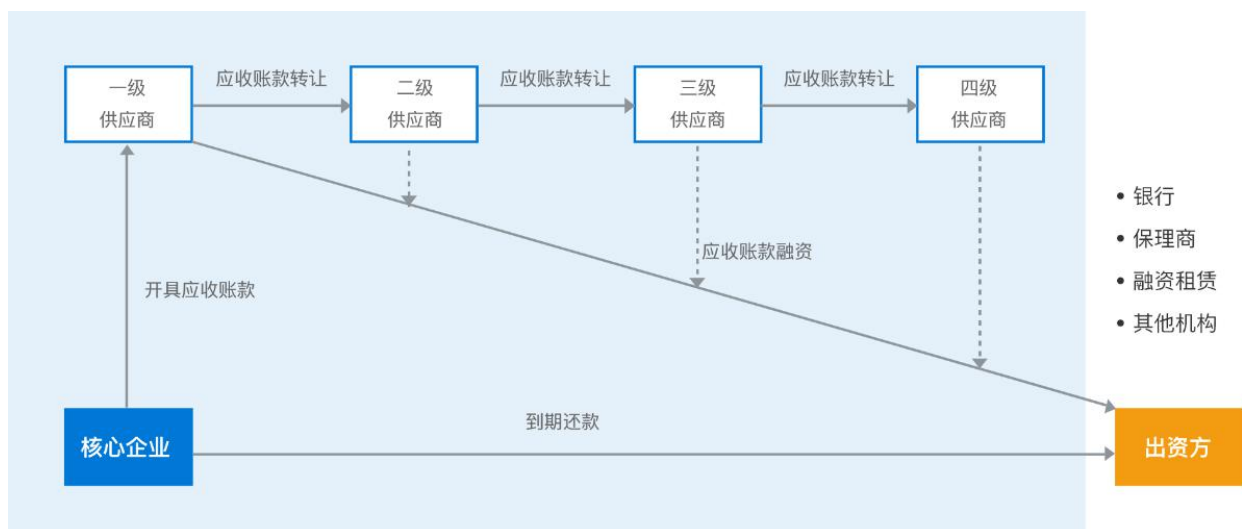
将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

但是区块链可以有效解决该问题。

### 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：



将供应链上的每笔交易和应收款单据上链，同时引进第三方可新机构确认信息的交易，例如银行等，确保真实性。同时，可以允许融资、将账款转让等，让上游核心企业的信用传递到下游企业，减小下游企业融资难度。

### 三、 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。

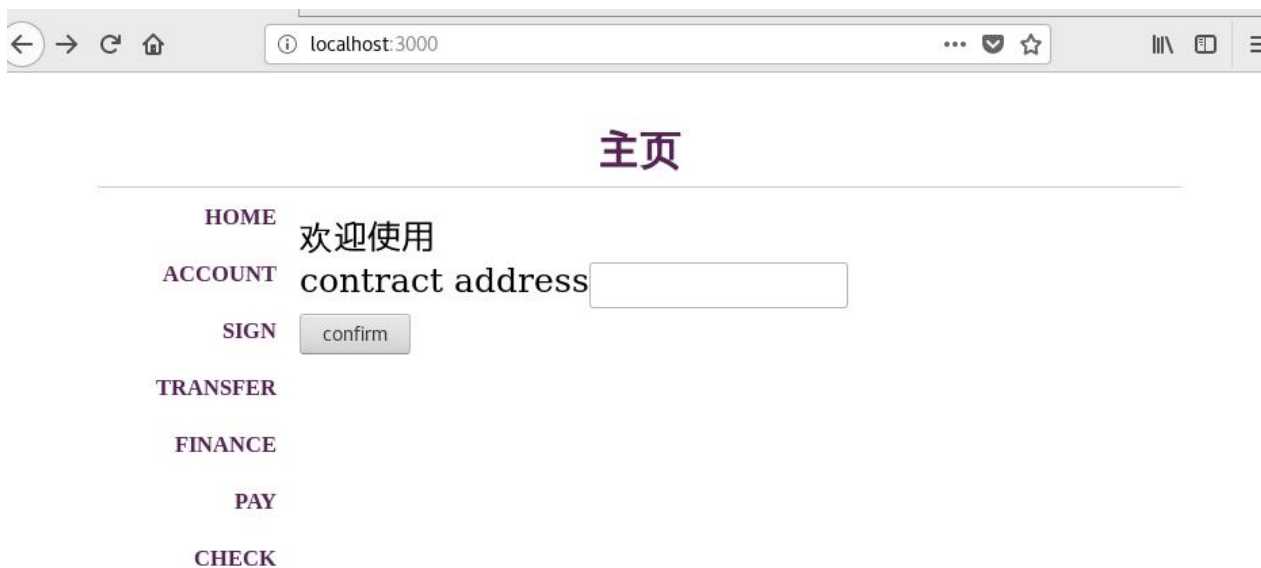
功能三：利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

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。

功能主要体现在智能合约上，代码如下：

### 四、 功能测试

前端界面如下：



合约存储设计:

```
struct Company {  
    bool registered; // 判断公司是否注册  
    address addr; // 公司公钥  
    uint current_amount; // 现有资产  
    uint debt_amount; // 债务总额  
    uint bond_amount; // 债券总额  
    mapping (address => uint) bond; // who owes you  
    mapping (address => uint) debt; // you owes who  
}
```

```
struct Receipt {  
    address from; // 发起者  
    address to;  
    uint amount; // 资金变动  
}
```

注册:

```
function register(address addr, uint amount) public returns(bool) {  
    if (companies[addr].registered == true) {  
        emit RegisterEvent(-2, addr, amount); // 公司已经注册过了  
        return false;  
    }  
    Company memory res = Company(true, addr, amount, 0, 0);  
    companies[addr] = res;  
    company_num++;  
    emit RegisterEvent(0, addr, amount);  
    return true;  
}
```

签发应收账款：

```
// 签发应收账款

function owe(address from, address to, uint amount) public {

    if (companies[to].registered == false) {

        emit OweEvent(-1, from, to, amount); // 债主不存在

        return;

    }

    companies[from].debt[to] += amount;

    companies[from].debt_amount += amount;

    companies[to].bond[from] += amount;

    companies[to].bond_amount += amount;

    receipts.push(Receipt(from, to, amount));

    emit OweEvent(0, from, to, amount);

}
```

应收账款转让:

```
// 应收帐单转让

function transfer(address x, address y, address z, uint amount) public returns(bool) {

    if (companies[x].registered == false || companies[z].registered == false) {

        emit TransferEvent(-1, x, y, z, amount); // 公司不存在

        return false;

    }

    if (companies[x].debt[y] < amount || companies[y].debt[z] < amount) {

        emit TransferEvent(-2, x, y, z, amount); // 所欠金额小于转让金额

        return false;

    }

    companies[x].debt[y] -= amount;

    companies[x].debt[z] += amount;

    companies[y].bond[x] -= amount;

    companies[y].debt[z] -= amount;

    companies[y].bond_amount -= amount;

    companies[y].debt_amount -= amount;

}
```

```
companies[z].bond[y] -= amount;
companies[z].bond[x] += amount;

receipts.push(Receipt(y, z, amount));
receipts.push(Receipt(y, x, amount));

emit TransferEvent(0, x, y, z, amount);
return true;
}
```

融资:

```

// 利用应收账款向银行融资

function finance(address v, uint amount) public returns(bool) {

    uint total_amount = companies[v].current_amount + companies[v].bond_amount
    - companies[v].debt_amount;

    if (amount > total_amount * 10) {

        emit FinanceEvent(-1, v, amount); // 融资额过高

        return false;

    }

    bank.bond[v] += amount; // 银行记录债务

    companies[v].debt[bank.addr] += amount; // 公司记录债务

    companies[v].current_amount += amount; // 公司现金增加

    receipts.push(Receipt(v, bank.addr, amount));

    emit FinanceEvent(0, v, amount); // 成功

    return true;

}

```

还钱:



```
// 下游企业要求核心企业支付欠款（到期还钱）

function payback(address from, address to, uint amount) public returns(bool)
{

    if (companies[to].registered == false) {
        emit PaybackEvent(-1, from, to, amount); // 公司错误
        return false;
    }

    if (companies[to].debt[from] < amount) {
        emit PaybackEvent(-2, from, to, amount); // 索要金额大于所欠金额
        return false;
    }

    if (companies[to].current_amount < amount) {
        emit PaybackEvent(-3, from, to, amount); // 公司现金不够偿还债务
        return false;
    }
}
```

```
    companies[to].current_amount -= amount;

    companies[to].debt[from] -= amount;

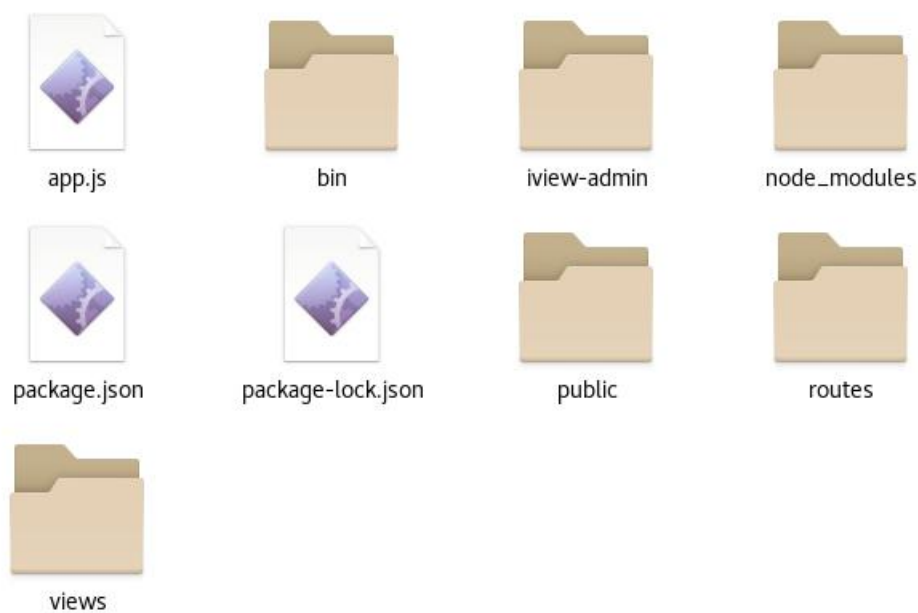
    companies[from].current_amount += amount;
    companies[from].bond[to] -= amount;

    receipts.push(Receipt(from, to, amount));

    emit PaybackEvent(0, from, to, amount);

    return true;
}
}
```

利用 express 搭建前段框架（默认端口 3000）：



通过 webase 提供接口，发送 post 请求，对链端部署的合约进行操作。

```
[cowcow1007@localhost webase-deploy]$ python deploy.py installAll
```

```
===== 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]:

```

=====
Generating CA key...
=====
Generating keys ...
Processing IP:127.0.0.1 Total:2 Agency:agencyA Groups:1,2
=====
Generating configurations...
Processing IP:127.0.0.1 Total:2 Agency:agencyA Groups:1,2
=====
Group:1 has 2 nodes
Group:2 has 2 nodes
=====
[INFO] Execute the following command to get FISCO-BCOS console
  bash <(curl -s https://raw.githubusercontent.com/FISCO-BCOS/console/master/to
s/download_console.sh)
=====
[INFO] FISCO-BCOS Path      : /home/cowcow1007/webase-deploy/nodes/fisco-bcos
[INFO] IP List File        : nodeconf
[INFO] Start Port           : 30300 20200 8545
[INFO] Server IP            : 127.0.0.1:2
[INFO] State Type           : storage
[INFO] RPC listen IP        : 0.0.0.0
[INFO] Output Dir            : /home/cowcow1007/webase-deploy/nodes
[INFO] CA Key Path           : /home/cowcow1007/webase-deploy/nodes/cert/ca.key
=====
[INFO] All completed. Files in /home/cowcow1007/webase-deploy/nodes
=====
          FISCO-BCOS      start... =====
try to start node0
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]:

===== script init success! =====
===== WeBASE-Node-Manager start... =====
Port 5002 has been occupied by other server PID(25001)

```

## 发送交易



合约名称: DebtTrans

合约地址: 0x59d3aeb645a1b5c9c1b6eb03! ⓘ

用户: comp3 ▾

方法: function ▾ set\_balance ▾

参数: company address  
amount uint256

ⓘ 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