

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	专业（方向）	软件工程
学号	17343109	姓名	王思博
电话	13531931585	Email	1033720276@qq.com
开始日期		完成日期	

一、项目背景

基于区块链、智能合约等，实现基于区块链的供应链金融平台。实现供应链应收账款资产的溯源、流转。

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。

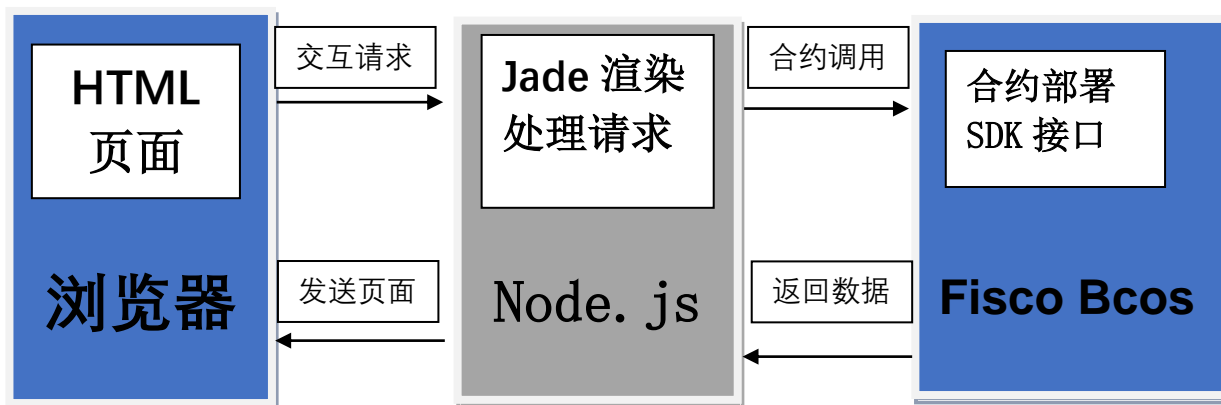
二、 方案设计

前端+后端+链端

前端采用 **html**，在浏览器展示和操作。

后端采用 **node.js** 的 **express** 框架，负责页面的跳转以及链端数据的调用，通过命令行打开运行。

链端采用 **fisco bcos** 的 **nodeJS SDK**，主要是部署合约以及提供调用接口，通过命令行打开运行。



链端

源代码位于 public/contracts，共四个 sol 文件

companyAndBank.sol 文件定义了结构体 company 表示厂商和银行对象，两者通过字段 status 进行区分，isCompany 和 isBank 实现厂商以及银行的查询功能，addCompany 和 addBank 实现厂商和银行的注册功能，能调用查询以避免重复注册，companys 和 banks 保存所有已注册厂商和银行的信息。

receiptAndFunc.sol 文件定义了结构体 receipt 表示账单，receipts 记录所有有效的欠账信息，fun1-5 五个函数实现基于 receipt 的所有基本操作，fun1 实现功能一：实现采购商品—签发应收账款 交易上链，会自动合并相同债券和债务方的欠账，避免 A 欠 B,B 欠 A 的循环，fun2 实现功能二：实现应收账款的转让上链，会计算支付一方的所有应收款避免超额支付，fun3 实现功能三：利用应收账款向银行融资上链，会计算所有应收款避免超额融资，fun4 实现功能四：应收账款支付结算上链，fun5 实现一个自定义功能，应付账款支付结算上链。

main.sol 文件集成所有功能，并实现三个函数打印当前厂商银行以及账单的信息，listCompany 打印厂商信息，listBank 打印银行 信息，listReceipt 打印账单信息，主要用于测试以及展示。

Console.Sol 是辅助类，提供 log 函数用于打印。

后端：采用 nodejs SDK 和 express 框架，

使用服务之前，首先需要初始化全局的 Configuration 对象，用以为各个服务提供必要的配置信息。Configuration 对象位于 nodejs-sdk/packages/api/common/configuration.js，配置文件位于 nodejs-sdk/packages/cli/conf/config.json。

```
const Configuration = require('./nodejs-sdk/packages/api/common/configuration').Configuration;
Configuration.setConfig('./nodejs-sdk/packages/cli/conf/config.json');
```

Web3jService 类位于 nodejs-sdk/packages/api/web3j

提供访问 FISCO BCOS 2.0+ 节点 JSON-RPC；部署合约；调用合约等功能

```
var api=require('../nodejs-sdk/packages/api/index.js');
var web3j=new api.Web3jService();
var ad="0xf18b20ef02bbac594755ee8e778415fb967b5c8";
```

ad 是合约地址，在部署时返回的 contractAddress 字段，由于每次部署需要的时间很长，采用 cookies 技术，在初次运行时自动进行部署，保存在 cookies 中，下次打开时通过检测 cookies 可以避免重复部署。

```
res.cookie("contractAddress",ad,{maxAge:"10000000",httpOnly:false});
```

后端代码主要在 routes/index.js 中，利用 express 框架自带的 Router 进行页面跳转和处理请求，接受前端的请求，根据请求从链端拉取数据或调用函数，再渲染页面加上数据返回给前端。

前端：

采用 jade 生成 html，位于 views/index.jade，三个变量 title, isDeploy, choices，分别表示标题，是否显示部署界面，以及附加表格。

public/javascripts/index.js 文件包含前端的 JS 代码，主要用于界面跳转。

三、 功能测试

首先需要运行 fisco,进入 fisco/nodes/127.0.0.1/目录，运行 start_all.sh，成功的话会显示如下界面

```
yrty@yrty-VirtualBox:~/Desktop/my_project/fisco/nodes/127.0.0.1$ ./start_all.sh
try to start node0
try to start node1
try to start node2
try to start node3
node0 start successfully
node2 start successfully
node1 start successfully
node3 start successfully
```

然后返回 my_project 根目录,运行 npm start 启动 express 项目。成功的话如下

```
yrty@yrty-VirtualBox:~/Desktop/my_project$ npm start

> my-project@0.0.0 start /home/yrty/Desktop/my_project
> node ./bin/www

listening at port 3000
```

然后在浏览器地址输入 localhost:3000,回车进入前端页面，成功的话如下

Fisco Bcos



Welcome to Fisco Bcos App

部署

FISCO BCOS

关于使用请看测试视频

四、 界面展示

启动页面

Fisco Bcos



Welcome to Fisco Bcos App

部署

FISCO BCOS

功能选择页面

Fisco Bcos



合约已部署

返回

首页

注册

厂家

银行

基本功能

功能1

功能2

功能3

功能4

功能5

打印

打印厂家

打印银行

打印账单

FISCO BCOS

功能调用页面

Fisco Bcos



功能1,调用中

返回

首页

注册

厂家

银行

基本功能

功能1

功能2

功能3

功能4

功能5

打印

打印厂家

打印银行

打印账单

功能1: a向b支付c元

a

b

c

提交

更详细请看测试视频。

五、 心得体会

经过三个阶段实现了该项目，对 fisco 的架构设计以及原理应用有了逐步加深的理解，同时对其背后包含的区块链技术的原理和应用也理解更加深刻，但是 fisco 是个很大的项目，区块链技术也是包罗万象，我个人目前的理解还是十分浅薄，对其原理的一些细节不甚了解，应用起来也并不十分得心应手。此外还学习了 solidity 以及智能合约的一些内容，和以往写过的编程语言有很大不同，算是入门。

关于加分项

功能：为每个基础功能提供了类型检测以及避免超额支付融资等优化，基本可能不用考虑用户输入的问题，另外还附带实现了一个自定义功能，应付账款支付结算上链。

底层：无

合约：无

前端：采用 HTML 界面，比起命令行应该算得上用户友好，交互逻辑也简洁明了，同时合约中带有一定纠错功能，用户输入也无需过分关心。

其他：无

不足之处：

后端代码基本集中在路由跳转部分，而且有些代码可以转移到前端，例如用 **JS** 添加表格而不是重新渲染发送页面。

链端代码仅仅实现了比较基础的函数，还可以添加许多基于基础操作的高级操作，例如实现两个厂商的兼并，消除欠账循环（A 欠 B,B 欠 C,C 欠 A）等。

代码存在某个未找到的 bug,导致可以成功利用 sdk 调用合约，但运行结果却没有写入链端，因为无法写入，打印功能也就无法体现。