基于超级账本开发一个区块链应用——票据背书

一 为什么要做基于区块链技术的数字票据？

票据现有的形式有纸质票据和电子票据。纸质票据是传统的票据形式，需要在票据上字或者加盖有效印章才能生效。电子票据是基于央行牵头开发完成的电子商业汇票系统(ECDS)，银行或者企业通过直连或者网银接人，所有的票据承兑、交易等都需要通过ECDS 才能完成，是典型的中心化系统。

纸质票据和电子票据都有一些尚未解决的痛点。

- 票据真实性问题: 票据的贸易背景信息可能并不真实存在或者存在偏差，票据信息也容易被克隆和伪造，市场中存在的假票很难识别。
- 票据安全性问题: 纸质票据在携带过程有遗失或者损坏的风险，也存在打款和背书不同步的情况。
- 票据信用风险问题: 可能存在汇票到期后承兑人未及时将相关款项划入持票人账户的情况。
- 票据处理效率问题: 在途时间会造成资金结算延后，监管和审计成本也会很高。
- 票据违规交易问题: 票据交易主体或者中介机构可能存在一票多卖、清单交易、出租账户等违法违规行为。

借助区块链技术，可以在实现原有的票据业务的基础上解决现有的一些痛点。

a. 在不依赖可信第三方平台的基础上实现票据信息的真实性。 纸质票据和电子票据的交易双方都依赖可信的第三方平台或者票据实物验证票据的真伪。但是在实际的操作过程中，票据的真实信息是很难确认的。超级账本的成员管理利用底层的安全机制确保上链信息的真实性，基于记录到分布式共享账本中的票据信息，杜绝假票、克隆票、变造票等伪假票据。

b. 分布式共享账本避免违规交易 纸质票据中的一票多卖和电子票据中打款背书不同步等违规交易存在的原因是信息的不同步。基于区块链技术的数字票据利用分布式共享账本记录的信息具有不可篡改性，一旦交易完成并记录到账本中，就不会存在纸质票据和电子票据中的数据丢失和信息不同步的问题，交易双方都没有机会进行违规交易。

c. 智能合约实现票据业务自动处理 智能合约功能可以实现复杂的业务逻辑，比如业务操作前可以基于不可篡改的账本信息搜集和评估参与者的信用，拒绝信用评级低的交易方提交的请求，业务操作后可以自动实现转账等功能，解决不同操作导致的信息不同步问题。

d. 数字票据提升票据处理效率 分布式共享账本在区块链网络中有多份数据拷贝，交易结束后就完成了对账，利用本地的账本能快速检索信息，做更多的业务处理，提升票据处理的效率。

e. 区块链技术提高票据安全性 票据安全性包括物理安全、数据安全、网络安全等多个方面。纸质票据在携带过程有遗失或者损坏的风险，属于物理安全范畴。纸质票据和电子票据都可能存在数据被篡改的可能性，属于数据安全范畴。传输过程中被中间人攻击等属于网络安全范畴。基于区块链技术的数字票据综合了多种底层技术，分布式账本解决了物理安全和数据安全的问题，数字签名和安全传输解决网络安全的问题。

二 票据背书需求分析

票据背书的应用开发实例会对票据的应用场最进行简化，我们实现的业务逻辑包括票据发布、票据背书、票据签收，票据拒收、票据查询等操作，实际的票据业务需要根据实际需求做调整。

1.票据发布

票据发布操作生成一个票据，包括如下5类信息。

a. 票据基本信息

- 票据号码
- 票据金额
- 票据种类
- 票据出票日期
- 票据到期日期

b. 出票人信息

- 出票人名称
- 出票人证件号码

c. 承兑人信息口

- 承兑人名称
- 承兑人证件号码

d. 收款人信息

- 收款人名称
- 收款人证件号码

e. 持票人信息

- 持票人名称
- 持票人证件号码

2. 票据背书

票据背书是转让票据权利的重要方式和手段。发票票据需要先获取持票人持有的票据，填写被背书人的信息:

- 被背书人名称;
- 被背书人的证件号码。

发起票据背书的请求后会提交给被背书人。被背书人接收到票据背书的请求后，可以择签收票据或者拒绝背书。

3. 票据背书签收

票据背书签收实现了票据权利的转让，签收前需要仔细地查阅待签收票据的信息，确保内容的完整性。票据签收的过程需要用被背书人的签名密钥对签收的内容进行数字签名，实现防抵赖的功能。

4. 票据背书拒收

被背书人可以拒绝对票据背书的请求进行签收，拒绝以后，票据的持有人还是票据背书的发起者。拒绝背书的操作也会记录到票据的历史流转信息中。

5. 票据信息查询

票据信息查询可以获取自己持有的票据和待签收的票据，也可以根据票据号码查询票据的详细信息，包括票据的历史流转信息。

三 票据背书架构设计

根据票据背书的需求分析，下面设计一个简单的架构，再定义票据背书的数据模型。

1. 票据背书的分成架构

我们将基于区块链的数字票据进行分层设计，包括Hyperledger Fabric底层平台 智能合约、业务层和应用层，如下图所示。

1.1. 应用层

用户登录, 发布票据, 发起背书, 签收背书, 拒绝背书, 我的票据, 待签收票据, 票据详情

1.2. 业务层

用户管理, 票据管理, 其它...

1.3. 智能合约(链码)

票据发布, 票据背书, 票据背书签收, 票据背书拒签,

查询持票人票据, 查询待签收票据, 根据票据号码查询票据详情

1.4. 区块链底层平台

CA, 区块链, 账本

每层的主要功能如下。

- 区块链底层平台: 提供分布式共享账本的维护、状态数据库维护、智能合约的全生命周期管理等区块链功能，实现数据的不可篡改和智能合约的业务逻辑。根据第11章的内容搭建区块链网络以后，默认就提供了这部分功能。另外，通过fabric-ca提供成员注册和注销等功能。
- 智能合约: 智能合约通过链码来实现，包括票据发布、票据背书、票据背书签收、票据背书拒绝等链码调用功能，链码查询包括查询持票人票据、查询待签收票据、根据链码号码查询票据信息等。
- 业务层: 业务层是应用程序的后端服务，给Web应用提供RESTful的接口，处理前端的业务请求。后端服务的基本功能包括用户管理和票据管理，通过Hyperledger Fabri提供的Go SDK和区块链网络进行通信。业务层也可以和其他的业务系统进行交互。
- 应用层，Web应用采用jQuery+HTML+CSS 的前端架构编写页面，提供用户交互的界面操作，包括用户操作的功能

  业务操作的功能。用户是内置的，只提供用户登录和用户退出操作。业务操作包括发布 查询持票人持有的票据、发起票据背书、查询待签收票据、签收票据背书、拒绝票据背书等功能。 各个层之间采用不同的接口，业务层的Go SDK、智能合约和区块链底层平台之间用gRPC的接口。

2. 票据背书的数据模型

下面看一下链码设计的数据模型，包括票据数据结构定义和票据状态定义。

a. 票据数据结构

票据信息包括票据基本信息、出票人信息、承兑人信息、收款人信息、持票人信息、待背书人信息、拒绝背书人信息、票据状态和票据背书历史等，数据结构定义如下:

```
//票据数据结构
type Bill struct {
    BillInfoID string `json:BillInfoID`                  //票据号码
    BillInfoAmt string `json:BillInfoAmt`                //票据金额
    BillInfoType string `json:BillInfoType`              //票据类型
    BillInfoIsseDate string `json:BillInfoIsseDate`     //票据出票日期
    BillInfoDueDate string `json:BillInfoDueDate`       //票据到期日期
    DrwrCmID string `json:DrwrCmID`                     //出票人证件号
码
    DrwrAcct string `json:DrwrAcct`                     //出票人名称
    AccptrCmID string `json:AccptrCmID`                 //承兑人证件号
码
    AccptrAcct string `json:AccptrAcct`                 //承兑人名称
    PyeeCmID string `json:PyeeCmID`                     //收款人证件号
码
    PyeeAcct string `json:PyeeAcct`                     //收款人名称
    HoldrCmID string `json:HodrCmID`                     //持票人证件
号码
    HoldrAcct string `json:HodrAcct`                     //持票人名称
    WaitEndorserCmID string `json:WaitEndorserCmID`     //待背书人证件
号码
    WaitEndorserAcct string `json:WaitEndorserAcct`     //待背书人名称
    RejectEndorserCmID string `json:RejectEndorserCmID`//拒绝背书人证
件号码
    RejectEndorserAcct string `json:RejectEndorserAcct`//拒绝背书人名
称
    State string `json:State`                          //票据状态
    History []HistoryItem `json:History`               //背书历史
}
```

票据历史信息包含了票据的流转信息，比如票据发布、票据签收、票据拒绝等都会记录到历史信息中，数据结构定义如下:

```
// 背书历史item结构
type HistoryItem struct {
    TxId  string `json:"txId"`
    Bill Bill `json:"bill"`
}
```

票据历史信息是智能合约自动完成的。

b. 票据状态模型

票据状态定义如下：

票据状态	NewPublish	EndrWaitSign	EndrSigned	EndrReject
状态说明	票据新发布	票据等待签收	票据签收成功	票据被拒绝签收

票据发布以后进人票据新发布状态NewPublish,票据持票人可以提交票据背书的操作进行票据权利转移，进入票据等待签收的状态EndrWaitSign

票据被背书人接收到票据背书请求后，可以选择签收票据或者拒绝签收，票据签收成功进入状态EndrSigned,持票人转移为被背书人;

拒绝签收进人状态EndrReject,持票人保持不变还是原有的持票人。

处于Endrsigned 和EndrRejet 状态的持票人都可以再次发起票据背书的请求，进入下一轮的操作。


区块链部落
专注于区块链技术
识别图中二维码关注我们