**本次演示，分为两个环节，首先是智能法律合约建模语言环节，其次是对智能法律合约中运行时性质的验证。**

**首先是根据本项目的智能法律合约建模语言对PPT中的合约进行建模，建模工具基于BPMN业务流程执行引擎的的二次开发，主要增加了参数输入面板(BPMN面板)和用户输入面板(电子合同的前端)**

**基于Beagle原则的智能法律合约建模，需要体现以下原则：**

**流程原则**

**问责原则**

**回滚原则**

**托管原则**

**共识原则**

**预言机原则**

**其中共识原则和预言机原则已经在区块链中实现了，托管原则一般在合约中是有一个法律权威机构或者代理机构作为执行主体参与相关职能。**

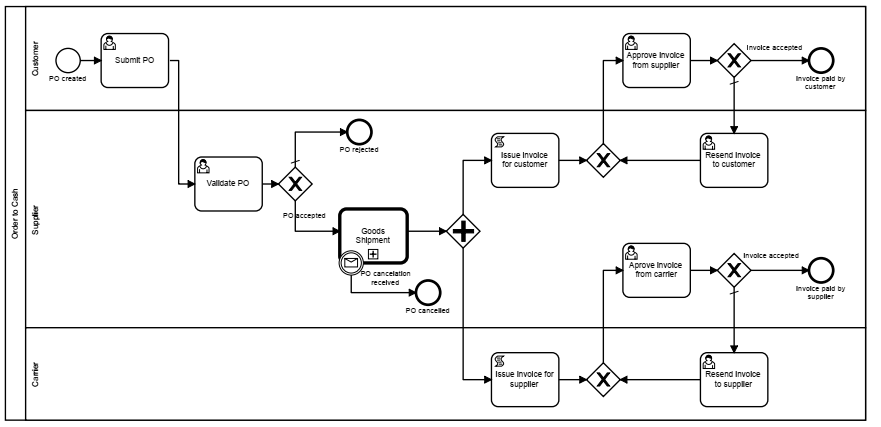
**因此接下来演示流程原则、问责原则和回滚原则这三个原则：**

**1.首先是流程原则，流程原则要求 在设计合约模板时将合约抽象为流程模型，并且具有可配置变量的合约模板。**

**该部分分为两部分，首先是建模，其次是对模型参数的输入(通过建模环境展现)**

**建模是针对合约开发者，相当于将一份并未填写的合同模板建为等价的流程模型。**

**第一次建模可得以下的模型，该模型中Good shipment作为一个子流程被主流程调用。**

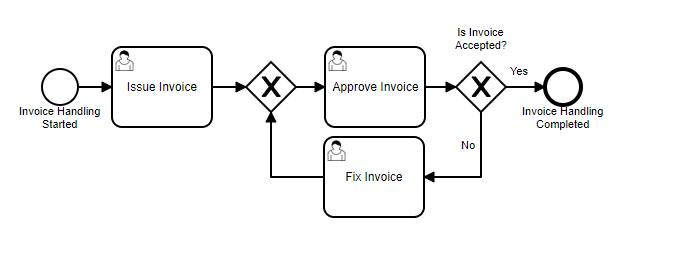


**此外可以发现分别给消费者和供应商开具发票的流程除了参数外是基本一样的，因此可以建立一个调用子流程，**

**这也是使用流程原则的好处——即对于相同流程的子流程，可以采取复用的原则。**

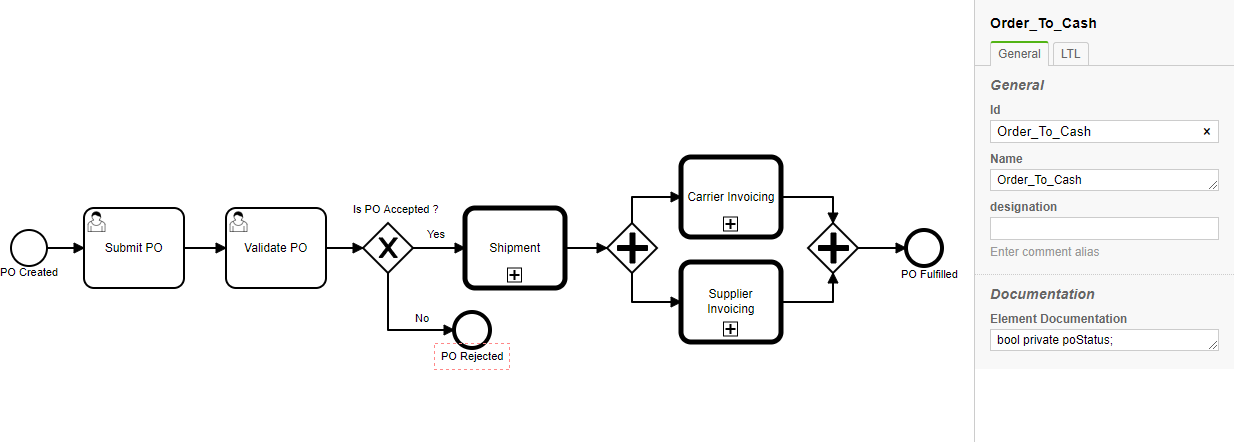
**对开发票子流程的建模如下：**

**展示调用开发票调用子流程的BPMN图：**



**只需要将每个任务的执行者绑定为不同的对象即可,这会在具体建模环境的演示中展示。**

**在将开发票的流程复用同一个调用子流程后，最终得到的主流程图如下：**



**可以看到，两个开发票的子流程变为了调用子流程，**

**右侧是参数面板**

**2.问责原则**

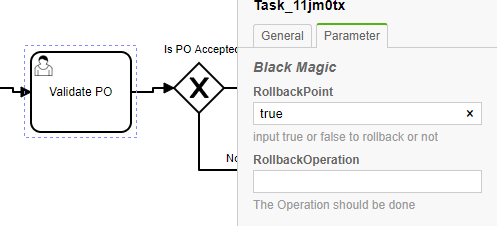
**<这一块我将会放入一个加入了问责流程的新主流程图>**

**合约中具有问责流程机制，即发票应在规定时间内开具完成，否则将会有法律机构介入追责，**

**这是 加入了问责机制后的**

**2.流程的回滚原则**

**回滚原则的实现是在某一个任务中插入回滚点。**



**该任务被标记为回滚点，则一旦后续的任务执行出错，比如运行时验证检测出合约执行异常**

**则合约进行回滚——回滚的主要目的是将合约中的原语操作还原，包括合约中最关键的转账行为。**

**转账的原语操作并不在模型中设置而是由用户以太坊账号进行完成。**

**这是钱包的界面。**

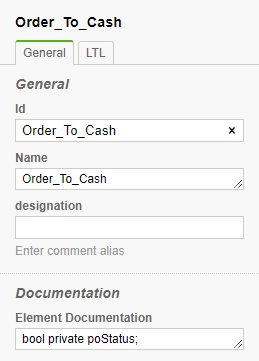
**3. 参数的输入展示**

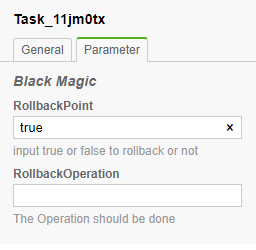
**3.1 模型上的参数的输入**

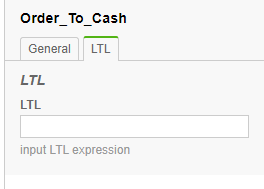
**大致介绍，模型上的参数更加偏向代码，因此需要建立一个前端模板，以便于用户像填写合同一样填写参数，参数经解析器解析插入至模型中，从而使BPMN模型文件具有参数。**

**介绍参数界面**

**其中的一个界面，介绍界面的元素**



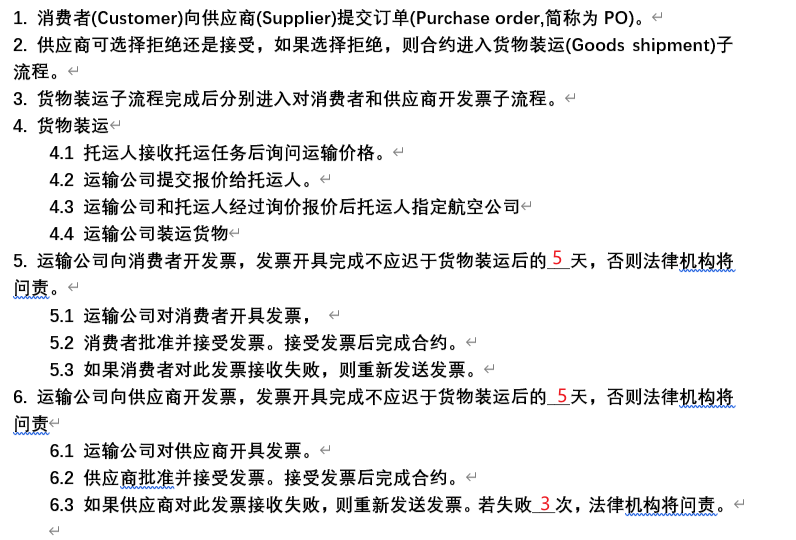




**3.2 用户输入的参数**

**填写一个json文件，并将JOSN文件展示后得到的带有参数的模型文件展示出来。**

**给出一个后续期望的用户输入参数的前端界面。 类似如下界面：**



**二、运行时验证展示**

**1、展示模型中的运行时验证性质**

**1.1展示模型中的运行时验证性质，采用之前合约模型中的一条性质**

**1.2展示该性质生成的自动机的图形，以便观众理解**

**2、展示该性质的监控器**

**2.1 监控器的生成，运行**

**展示rvtool输入LTL后的运行的日志，和上次录制类似**

**2.2 对合约的监控执行**

**(这里的目标系统还是用之前的演示过的事件仿真生成器，只不过模拟的是智能合约的事件的生成)**

**2.3 分别对正确的示例， 以及错误的示例进行展示**