智能合约升级

如何在fabic中升级chaincode

姜耀国，华为

智能合约

- 智能合约（尼克•萨博）
  - 是用来验证、执行合约的计算机协议
  - 在预先设定的条件满足的情况下自动执行
  - code is law
- 用区块链实现智能合约
  - 去中心化
  - 公开透明
  - 不可更改性
    - 是指历史记录不能修改
    - 只能增加新的记录

问题

智能合约是否应该可以被修改呢？

- 2016年The DAO被盗事件
  - 智能合约出现漏洞，大量以太币被盗走
  - 拖沓的修复流程，经历多次数据分叉
  - 项目最终失败
- 带给我们的启发
  - 不能认为智能合约一旦发布即可永久地、完美地运行下去。现实中的法律会不断地修改，那么数字世界的规则也应该如此
  - 企业级、商业化的应用仍然需要依赖某些中心化的组织的支持

基本概念

- **chaincode**
  - 智能合约在fabric上的实现方式
  - 规定了在区块链上对数据（world state）进行操作的逻辑
  - 加载到区块链网络上之后，所有人都必须按照chaincode实现的逻辑对数据进行操作。
    - 购买资产逻辑：购买账户扣钱-资产所有人变更
  - 是运行在区块链网络上的一段实际的计算机程序

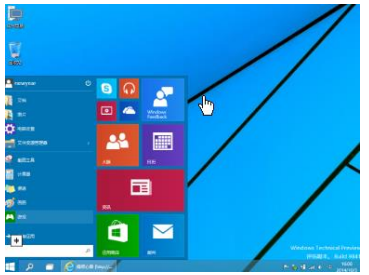
升级面临的需求

- Chaincode面临计算机程序的通用问题
  - 程序可能会存在漏洞/bug
    - The DAO被盗事件
  - 需求会变更
    - 业务逻辑增加/删除/修改
    - 功能完善
- 升级的基本要求
  - 不能随便改动chaincode
  - 升级记录公开可查

实现方案

- System chaincode
  - 是一种fabric内置的chaincode
  - 随fabric启动而启动
  - 限制为只能由fabric调用（invoke/query）
- Lifecycle system chaincode
  - 管理chaincode生命周期
  - 类似操作系统中的应用管理模块

Fabric →

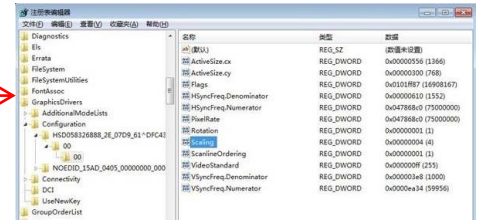


操作系统

LFSCC →

应用管理器

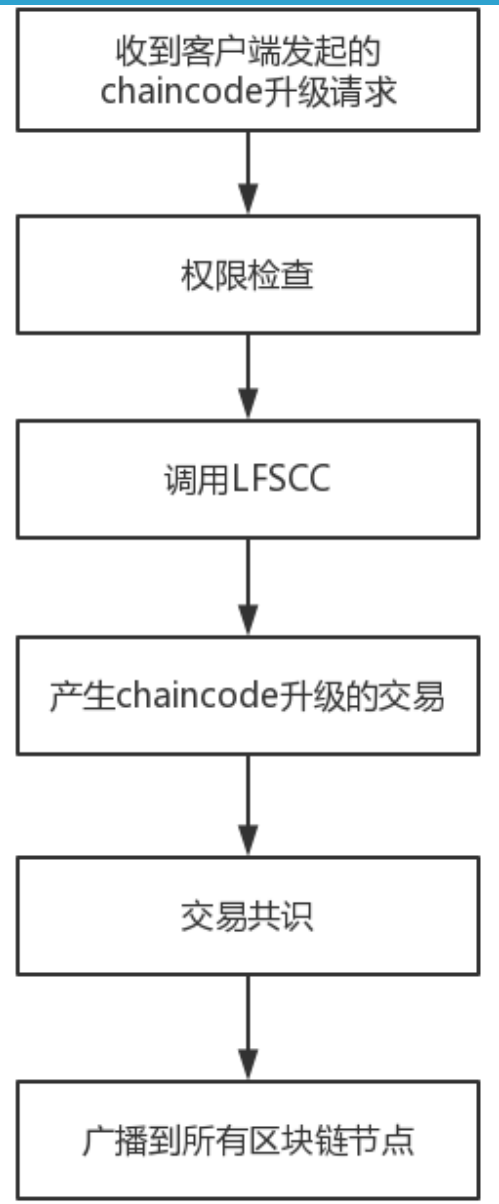
World state of LFSCC →



注册表

实现方案

- Lifecyce system chaincode接口
  - Deploy：加载chaincode代码到区块链网络
  - Upgrade：更新区块链上已有的chaicode代码
  - Query：查询chaincode信息
    - getid
    - getdepspec
    - Getccdata
- 通过对lifecycle system chaincode进行调用，产生chaicode加载/升级的交易记录，通过query类接口可进行查询
- 升级交易最终会广播到区块链所有节点，这样所有节点就有了新的chaincode代码



更多需求

- 升级过程中对数据进行修改
  - 因为漏洞/bug，产生不合法数据，需要进行修正
  - 因为需求变更，数据结构产生变化
    - 资产增加新的属性
    - 数据表增加新的列
    - 增加新的数据
- 这个修改同样要求公开

实现方案

- re-init接口
  - 在升级的代码中实现re-init接口
  - re-init接口代码中包含需要对数据（world state）进行的逻辑操作
  - 在升级chaincode的过程中，re-init接口中的代码会被运行，对chaincode数据进行修改，所有的修改会随着升级交易写入账本而生效
- 具体upgrade命令示例
  - peer chaincode upgrade -n mycc -p github.com/…/… -c '{"Function":"re-init", "Args": "……."]}'

Q&A

THANK YOU

感谢聆听

HAVE A NICE DAY!