ODIN 与超级账本 Fabric 结合应用方案示例

超级账本 Fabric1.0 调用接口的 URI 形式定义:

fabric:[server_ip1:port1,ip2:port2,....]/channel_id/contract_id/function_name(argv1,argv2,....,argn)

合约里针对注册管理标识、更新数据块等具体功能可以定义若干 function

在此基础上,在 ODIN 标识协议框架下定义 ppk:odin_id1/odin_id2/.../function_name(arg1,arg2,...,argn)

这样的 URI 形式来映射支持通用的、分布式的方法调用, 其中的 arg 参数可以递归采用符合 ODIN 定义的 URI 资源标识, 与函数式编程方法结合可以很好地满足应用开发需求。

传统 WEB

下一代对等 WEB

访问 访问 http://www.demo.com/test.php 应用程序 ppk:479110.1304/register newdevice('id','pubkey')# 通过 DNS 协议从 DNS 服务器将 www.demo.com 通过 ODIN 协议从比特币区块链上将 479110.1304 解析为 URI: 命名解析协议 解析为 IP 地址 62.56.78.212 fabric:[10.6.2.189:3456,10.6.3.19:3456]/demochannel/odin/process odin interest 向实际 IP 地址上运行的 HTTP Server 服务平台 动态连入 Fabric 区块链系统的组成节点之一发出请求 发出请求 HTTP Server 调用执行 PHP 代码返回结果 业务逻辑 Fabric 区块链系统调用执行对应的智能合约和方法返回结果

```
举例:对于一级标识示例 479110.1304 配置其 AccessPoint 参数为:
  fabric:[10.6.2.189:3456,10.6.3.19:3456]/demochannel/odin/process odin interest
输入参数: JSON 格式的 ODIN 标识协议请求兴趣包数据
   "ver":1.
   "uri":"ppk:479110.1304/register newdevice('id','pubkey')#",
         JSON 格式的 ODIN 标识协议应答数据包
输出结果:
   "ver":1.
   "content":{
     "uri":"ppk:479110.1304/register_newdevice('id','pubkey')#20170910125623156.0",
     "block id":" 当前数据区块标识",
     "lastblock_id":"上一数据区块标识,该字段取值可以为空字符串,表示没有上一连续区块",
     "chunk sn": 当前数据子块编号,从 0 开始,
     "chunk size": 当前区块内数据子块数量,取值为0时表示是动态流数据,子块数目不确定
     "content type":" 子块正文类型,采用 HTTP 协议相关定义,如 text/html,image/jpg 等 ",
     "content length": 子块正文长度,
     "data":" 子块正文数据,如果是二进制需要采用 Base64 编码 ",
   "signature":{
     "algo":" 签名算法类型,如 MD5withRSA,SHA1withRSA,SHA256withRSA",
     "sign":"Base64 编码的签名数据 ",
     "utc": 到秒值的签名生成时间戳 ,
```

```
在应用程序里调用 ppk:479110.1304/register_newdevice('id','pubkey')#
该请求会被相应的 sdk 解析,并实际调用
 fabric:[10.6.2.189:3456,10.6.3.19:3456]/demochannel/odin/process_odin_interest(
    {"ver":1,"uri":"ppk:479110.1304/register_newdevice('id','pubkey')#"}
获得返回结果
如果是获取数据内容,也可以同样处理。比如在应用程序里调用
   ppk:479110.1304/device list#1.0
该请求会被相应的 sdk 解析,并实际调用
 fabric:[10.6.2.189:3456,10.6.3.19:3456]/demochannel/odin/process_odin_interest(
    {"ver":1,"uri":"ppk:479110.1304/device_list#1.0"}
获得返回结果
```