**创智链侧链系统**

**用户手册**

**北京新创智链科技有限公司**

目录

[1、引言 3](#_Toc30102039)

[1.1 编写目的 3](#_Toc30102040)

[1.2 概念解释 3](#_Toc30102041)

[2、关于侧链 3](#_Toc30102042)

[2.1 侧链注册 6](#_Toc30102043)

[2.1.1 注册组织准备 6](#_Toc30102044)

[2.1.2 执行注册 7](#_Toc30102045)

[2.1.3 链上数据查询 8](#_Toc30102046)

[2.2 侧链创世 9](#_Toc30102047)

[2.2.1 创世准备 9](#_Toc30102048)

[2.2.2 创世transaction 9](#_Toc30102049)

[2.3侧链治理 13](#_Toc30102050)

[2.3.1 侧链管理快速入门 14](#_Toc30102051)

[2.4.1 节点管理快速入门 20](#_Toc30102052)

[2.4跨链trasnsfer 25](#_Toc30102053)

[2.4.1准备账户 25](#_Toc30102054)

[2.4.2基础通证跨链trasnsfer 25](#_Toc30102055)

[2.4.3跨链事务 31](#_Toc30102056)

[2.4.4结果查询 40](#_Toc30102057)

# 1、引言

## 1.1 编写目的

本文档提供的对象主要是侧链开发，运营以及使用用户。本文档对侧链主要功能和操作进行介绍，为用户操作、使用侧链相关内容提供参考和指导。

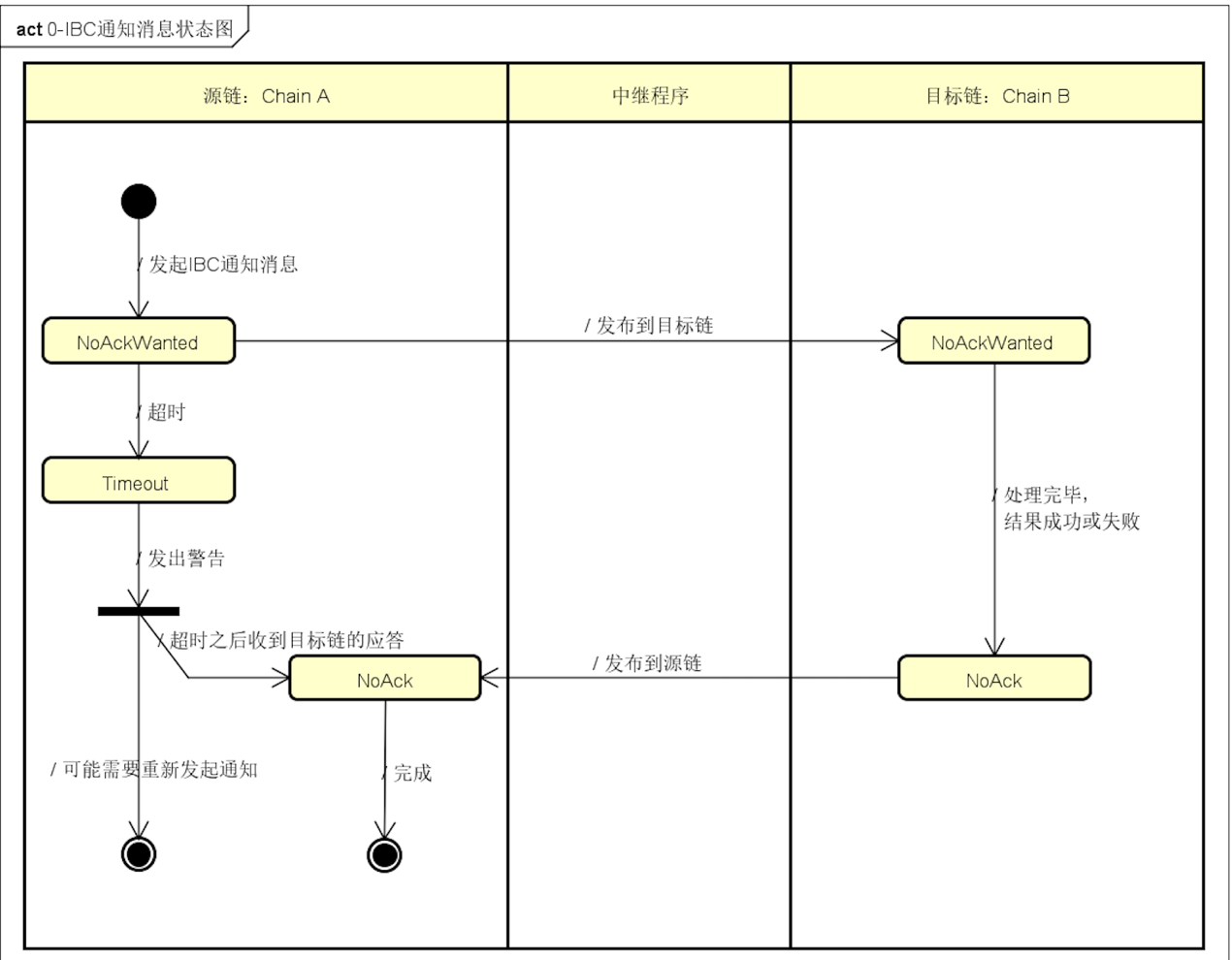
## 1.2 概念解释

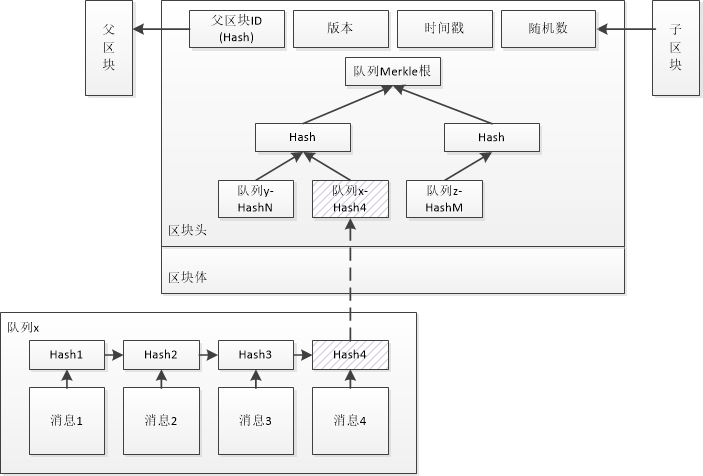
侧链通过双向锚定技术、IBC合约以及中继服务，实现区块链上的数字通证在主链以及侧链之间的互相转移。 侧链本身是独立的区块链，有自己的节点网络，代码以及数据也是相对独立的，所以它在运行过程中不会增加主链的负担，避免数据过度膨胀的情况出现。 通过侧链技术，我们可以实现不同业务之间的应用隔离，隐私保护，同时可以提高transaction吞吐量。

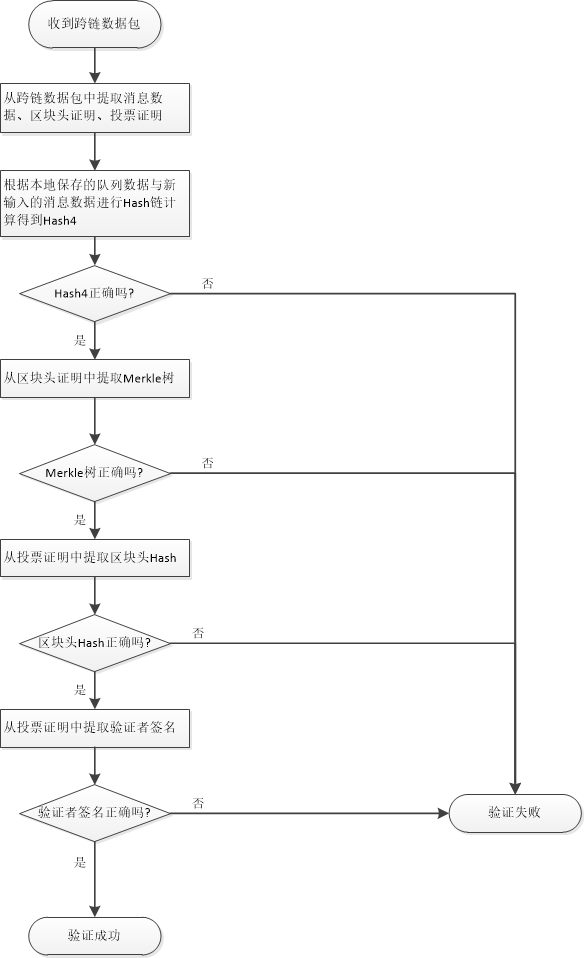
侧链技术进一步扩展了区块链技术的应用范围和创新空间，使传统区块链可以支持多种通证类型，以及小微支付、智能合约、安全处理机制、财产注册等，并可以增强区块链的隐私保护。利用侧链，我们可以轻松的建立各种智能化的应用如金融合约，股票、期货、衍生品等。

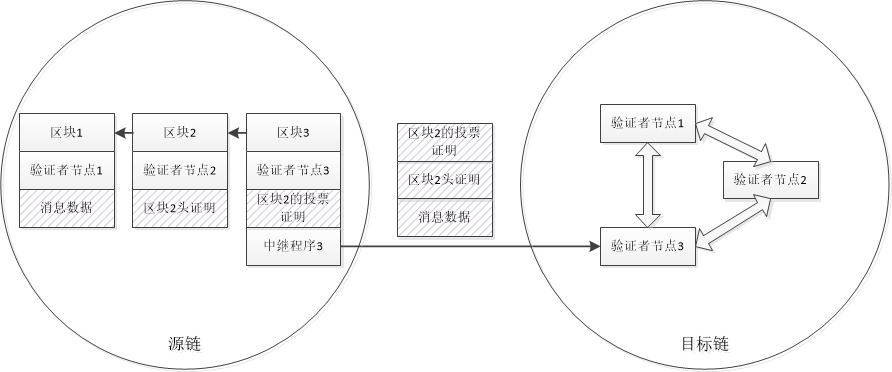
# 2、关于侧链

本序列文档操作示例基于gic以及giw工具，学习gic工具使用。详情请参考《创智链区块链客户端用户说明书》。以下为创智链侧链系统技术设计和流程文档。

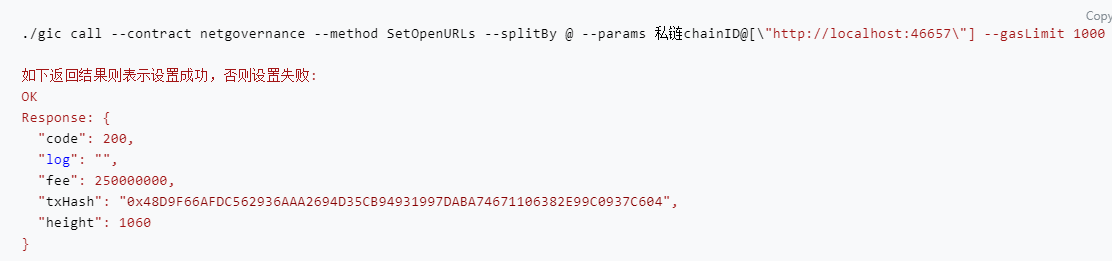








## 2.1 侧链注册



### 2.1.1 注册组织准备

1. 注册侧链之前需要先注册一个组织，组织名称自己拟定，注册侧链时需要指定侧链所属组织；侧链不能属于创世组织。
2. 注册组织：

我们要用测试账户 scowner在私有链上注册一个用于测试的组织 smartCity，命令如下：

[... pieces]# ./gic registerOrg -k=wal -n=scowner -p= SCOwner@2019 -g=500000 -o= smartCity

OK

Response: {

"code": 200,

"log": "",

"fee": 1250000000,

"txHash": "0x866BB086AA4CAA93D30F26FAD4AF4772F214E97198C10807F9F62FF2DADEA816 ",

"height": 931,

"orgID": "org54QQs2oCssBezq5P4fkdrKivULHZEudHE "

}

1. 设置主链的公开访问路径列表

注册完组织后还需要设置主链节点的公开访问路径列表，即OpenURLs，如["http://localhost:46657"]，侧链与主链通讯依赖OpenURLs，没有设置将导致侧链向主链发送的跨链transaction无法完成；下面以主链chainID为 local，OpenURLs为 ["http://localhost:46657"]为例，展示如何使用gic工具完成主链OpenURLs的设置：

./gic call --contract netgovernance --method SetOpenURLs --splitBy @ --params 私链chainID@[\"http://localhost:46657\"] --gasLimit 10000000 --orgName genesis --name owner --password Ab1@Cd3$

如下返回结果则表示设置成功，否则设置失败:

OK

Response: {

"code": 200,

"log": "",

"fee": 250000000,

"txHash": "0x48D9F66AFDC562936AAA2694D35CB94931997DABA74671106382E99C0937C604",

"height": 1060

}

命令行中选项 params 的值即为设置主链OpenURLs需要的参数，释义如下：

第一个参数：local 为主链的ChainID

第二个参数：[\"http://localhost:46657\"]为需要设置的主链公开访问路径列表

1. 准备侧链owner账户

注册侧链指定的侧链owner账户将决定后续的一序列操作将由哪个账户进行操作，该账户可以与注册组织的账户为同一账户。本文档我们使用在主链生成的账户账户名为scowner的账户作为侧链的owner账户。

1. 准备侧链名称

每条侧链都需要有确定的名称，侧链名称全局唯一。

侧链名称将用于生成侧链chainID，生成规则为：主链ID+[+侧链名称+]。

侧链名称可以与它所属组织名称相同。

### 2.1.2 执行注册

在准备工作完成后，就可以使用执行侧链注册操作，下面以组织名称 smartCity、侧链名称 smartCity、侧链owner账户地址localLvP9ForVPgagpSuH23MTp9jfW3GVxWfDK为例展示如何使用gic进行侧链的注册：

./gic call --contract netgovernance --method RegisterSideChain --splitBy @ --params smartCity@smartCity@localLvP9ForVPgagpSuH23MTp9jfW3GVxWfDK --gasLimit 100000 --orgName genesis --name owner --password Ab1@Cd3$

如下返回结果表示操作成功:

OK

Response: {

"code": 200,

"log": "",

"fee": 125000000,

"txHash": "0x1861879BD4D0C8631806948454586DCA2C5B94850BA44CAE12DB4908084F6121",

"height": 1436

}

命令行中选项 params 的值即为注册侧链时需要的参数，释义如下：

第一个参数：smartCity 为侧链名称

第二个参数：smartCity 为侧链所属组织名称

第三个参数：localLvP9ForVPgagpSuH23MTp9jfW3GVxWfDK 为侧链 owner 的账户地址

### 2.1.3 链上数据查询

以上操作完成后就可以从链上查询侧链的注册信息，以确认的数据无误，查询命令如下:

./gic query -k /sidechain/local[smartCity]/chaininfo

返回结果如下:

OK

Response:

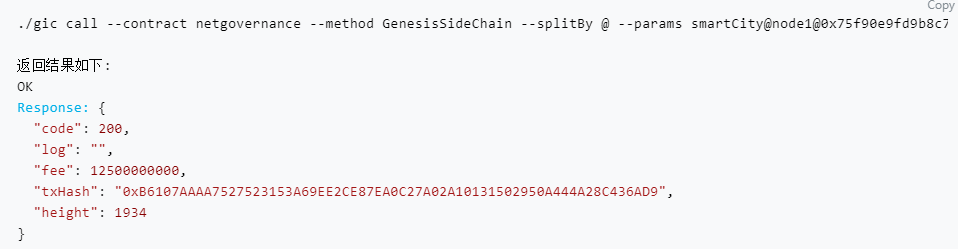
Code: 200

Key: /sidechain/local[smartCity]/chaininfo

Value: {"sideChainName":"smartCity","chainID":"local[smartCity]","NodeNames":null,"orgName":"smartCity","owner":"localLvP9ForVPgagpSuH23MTp9jfW3GVxWfDK","height":0,"status":"init","gasPriceRatio":""}

从查询结果中可以看到sideChainName为smartCity，chainID为local[smartCity]，状态为init表示侧链处于初始化状态等待创世。

## 2.2 侧链创世



### 2.2.1 创世准备

1. **搭建主链的观察者节点**

侧链创世需要先搭建主链的观察者节点，然后使用此观察者节点进行侧链创世。

1. **导出节点公钥**
   1. 搭建完主链的观察者节点之后需要导出观察者节点的节点公钥，此公钥在发送侧链创世transaction时会用到。
   2. 在主链的观察者节点执行以下命令导出节点公钥。

export TMHOME=/etc/tmcore

/usr/local/tmcore/bin/tendermint show\_validator

返回结果如下:

{"type":"AC26791624DE60","value":"dfkOn9m4x8wqxOz7i+hlpBo3dNey0TJ+mluT3IoYopo="}

75f90e9fd9b8c7cc2ac4ecfb8be865a41a3774d7b2d1327e9a5b93dc8a18a29a

75f90e9fd9b8c7cc2ac4ecfb8be865a41a3774d7b2d1327e9a5b93dc8a18a29a即为节点公钥

注：导出节点公钥时一定要设置环境变量export TMHOME=/etc/tmcore，否则导出的数据会不正确。

### 2.2.2 创世transaction

1. **执行transaction**

导出节点公钥之后，就可以向主链发送侧链创世transaction，需要使用侧链注册1.3准备的owner账户进行创世，下面以侧链名称smartCity、侧链节点名称node1、侧链节点公钥0x75f90e9fd9b8c7cc2ac4ecfb8be865a41a3774d7b2d1327e9a5b93dc8a18a29a、侧链的奖励地址local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK、侧链节点公开访问路径OpenURLshttp://侧链ip:46657、gasPrice调整比例1.000为例展示如何进行侧链的创世：

./gic call --contract netgovernance --method GenesisSideChain --splitBy @ --params smartCity@node1@0x75f90e9fd9b8c7cc2ac4ecfb8be865a41a3774d7b2d1327e9a5b93dc8a18a29a@local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK@http://侧链ip:46657@"1.000" --gasLimit 10000000 --orgName genesis --name scowner --password SCOwner@2019

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 12500000000,

"txHash": "0xB6107AAAA7527523153A69EE2CE87EA0C27A02A10131502950A444A28C436AD9",

"height": 1934

}

命令行中选项 params 的值即为侧链创世时需要的参数，释义如下：

第一个参数：smartCity 为 ChainName 侧链名称,创世后侧链的ChainId为主链ChainId[侧链ChainName]

第二个参数：node1 为 nodeName 侧链创世节点的名字

第三个参数：0x75f90e9fd9b8c7cc2ac4ecfb8be865a41a3774d7b2d1327e9a5b93dc8a18a29a 为 nodePubKey 侧链创世节点的公钥（即2.2.1导出的节点公钥，公钥前面加0x）

第四个参数：local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK 为 rewardAddr 侧链创世节点在侧链的接收手续费奖励的地址

第五个参数：http://侧链ip:46657 为 openURL 侧链公开的 URL

第六个参数："1.000" 为 gasPriceRatio gasPrice调整比例，小数点后保留三位小数

创世信息可以通过查询收据进行确认，查询命令如下:

./gic tx --txhash 0xB6107AAAA7527523153A69EE2CE87EA0C27A02A10131502950A444A28C436AD9 --chainid local

返回结果如下:

OK

Response: {

.....

"tags": {

.....

"/0/1/netgovernance.genesisSideChain": {

"Name": "netgovernance.genesisSideChain",

"ContractAddr": "",

.....

"genesisInfo": "{\"chain\_id\":\"local[smartCity]\",\"chain\_version\":\"2\",\"genesis\_time\":\"2019-11-30T11:09:56Z\",\"app\_hash\":\"\",\"app\_state\":{\"organization\":\"genesis\",\"gas\_price\_ratio\":\"1.000\",\"token\":{\"address\":\"local[smartCity]AJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA\",\"owner\":\"local[smartCity]ETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL\",\"name\":\"LOC\",\"symbol\":\"LOC\",\"totalSupply\":0,\"addSupplyEnabled\":false,\"burnEnabled\":false,\"gasprice\":2500},\"rewardStrategy\":[{\"name\":\"validators\",\"rewardPercent\":\"100.00\",\"address\":\"\"}],\"contracts\":[{\"name\":\"token-basic\",\"version\":\"2.1\",\"code\":\"token-basic-2.1.tar.gz\",\"codeHash\":\"ff66c796d153764491a3dfdbf95cd4dcb2bb9af6be683adc4825816dfcbe8372\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"59EA39CEB1AAA03FB25A1E5B1E9947548FAD9EACCA87AFDF2BA9BE784A929F18B21FCFF31419889951BBA79655D3CF7BABB95546CEC9E8621B4B4FB177EA5707\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"CF23FDCFD92408F43634D3BA7C89919D41416F269B3FB07025E148D6458B9E36772F5C473A279F17845B068AD23B983ECF8DE8326431192DFACA0691BB812B07\"}},{\"name\":\"token-issue\",\"version\":\"2.1\",\"code\":\"token-issue-2.1.tar.gz\",\"codeHash\":\"05cbacf200ad36c084efa13f372583615c3ae1dd12dgiec3c7e778889da256ea\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"9DAA25E40414F1F07E3D0798627465E36CF1DE7367AEECD820D5ABB95D2E2560F5056C2B86FBBF692C89FC3AB3179FAC8D5C273B28B5E54C6022CE5DD8EAF90C\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"3C73104013C634AC888E1E5063247C0778274B01B3F833A49B29E8518D4D059E7AAD13AFDB230DE28DF015CDDDE19033FF2C1785F273F74DC00C907A585CD20B\"}},{\"name\":\"governance\",\"version\":\"2.1\",\"owner\":\"local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK\",\"code\":\"governance-2.1.tar.gz\",\"codeHash\":\"32faaea316ee5f385b673c2f8d60637e4aa9c8166fca998cb5080bff7fceeacf\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"084631935319C2B7F0BADAD04103D0628BC79F606523907D76951A0056066DF79E5B0CC7094C46027B20C3299E257B8D47B173153C69B11B80F8E5DEBC289204\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"69ABDF8F61C5D96A169F5B2B2D853662D3CF279125AFD31DA4BD4FE883005F79CE716DC8816714B9584D2DEA6CEBCCDA6158856B731D26EFE9146817D891340D\"}},{\"name\":\"organization\",\"version\":\"2.1\",\"code\":\"organization-2.1.tar.gz\",\"codeHash\":\"b8f55cbb984553d68dfebbd608727db13162ff1b0aae36345aefeabd4951bb6d\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"3F66F5F4A25E1F67C53F2D36D216E299475F65AAD92A5F90D47E2B77B32B716787ADA9D9CCDCB2CD6EF31071E171063C57496F13D008518DBE1A6C5AE71A5107\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"0AC47FE7FB20BF9A66415CDD6BB52CC3A595E32FA2BEBF907C693BD054B674F5A9E4D1727D243B5F75298993122231734DE75E04764DABC86D2EF4B43694A900\"}},{\"name\":\"smartcontract\",\"version\":\"2.1\",\"code\":\"smartcontract-2.1.tar.gz\",\"codeHash\":\"41836992a6eb8cf3787323393841a89959d050fbe7ca5a18349f8d308e3b9dcf\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"AB18C8E319C2F844E780C9D4D6AEC2AD0874A2FD0CFD2CC719679A87407CE0FA23226304362F3C4D2500CBD06E1D84DBB6BA7AADE0B63F73FB3D83122D4E340B\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"96B0C8EEB55341060D1CDCC2F84A250F0C5EC9B1AFE232235E0D5F29EBE46F8FEF070AD3EE2AE3C5FD804AFF82170CA4917AB15921C23D453AFBA6D10E46440D\"}},{\"name\":\"igi\",\"version\":\"2.1\",\"code\":\"igi-2.1.tar.gz\",\"codeHash\":\"a684f89185e6f98f83dc78f10340ca5899e8fd2eb601ff3f8db4b0b038152747\",\"codeDevSig\":{\"pubkey\":\"5e8339cb1a5cce65602fd4f57e115905348f7e83gie38dd77694dbe1f8903c9\",\"signature\":\"64BD0994180A3EBCE404105772F74197A3EE5CAE35436D1E6DFB5EF67CB88F2E7B6ADC2FB23858AAF39455FC238D93B328863EE7D46D29756219F0640AEA340E\"},\"codeOrgSig\":{\"pubkey\":\"0c27b3396bdc1b486af51e84accbdd23a0d55a49c34fa51cf0ed6aedccf984d4\",\"signature\":\"2034DB17354A8F8604F659947A087BBE8D864EAF40CFF2EA19C2614219BDE01A15F84A5ACE31B1206F8B9DCB809B909753B6D7233F61C2B3C2421CCBF1F9D709\"}}],\"orgBind\":{\"orgName\":\"smartCity\",\"owner\":\"local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK\"},\"mainChain\":{\"openUrls\":[\"http://192.170.41.72:46657\"],\"validators\":{\"local582JbHcuDcQMdMVdzdwKJTy4ekhZk9MFb\":{\"nodepubkey\":\"6C549B3002D1E699732DFF1B78F011065BBA3BF95FC327AA896A1132D4BDB7E9\",\"power\":10,\"name\":\"local\",\"nodeaddr\":\"local582JbHcuDcQMdMVdzdwKJTy4ekhZk9MFb\"}}}},\"validators\":[{\"nodepubkey\":\"75F90E9FD9B8C7CC2AC4ECFB8BE865A41A3774D7B2D1327E9A5B93DC8A18A29A\",\"power\":10,\"reward\_addr\":\"local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK\",\"name\":\"node1\",\"nodeaddr\":\"local[smartCity]AKHbp57Sqfv8G39KHmsntWodxAfPDPTrd\"}]}",

"openURLs": [

"http://侧链ip:46657"

],

"sideChainID": "local[smartCity]"

}

},

.....

从上面的genesisInfo可以看到侧链信息：

"validators\":[{\"nodepubkey\":\"75F90E9FD9B8C7CC2AC4ECFB8BE865A41A3774D7B2D1327E9A5B93DC8A18A29A\",\"power\":10,\"reward\_addr\":\"local[smartCity]LvP9ForVPgagpSuH23MTp9jfW3GVxWfDK\",\"name\":\"node1\",\"nodeaddr\":\"local[smartCity]AKHbp57Sqfv8G39KHmsntWodxAfPDPTrd\"}]}",

中的"nodepubkey"应该和1.2中导出的节点公钥字母转换成大写的一致。

"openURLs"为创世时设置的侧链地址

1. **结果确认**

想要确认侧链是否创世成功，需要在主链以及侧链上分别进行确认，查询命令如下。

在主链进行确认：

在主链上用以下命令查询:状态变为ready

./gic query -k /sidechain/local[smartCity]/chaininfo

返回结果如下:

OK

Response:

Code: 200

Key: /sidechain/local[smartCity]/chaininfo

Value: {"sideChainName":"smartCity","chainID":"local[smartCity]","NodeNames":["node1"],"orgName":"smartCity","owner":"localLvP9ForVPgagpSuH23MTp9jfW3GVxWfDK","height":1934,"status":"ready","gasPriceRatio":"1.000"}

在侧链进行确认：

在侧链上用以下命令查询:

./gic query -k /genesis/chainid

返回结果如下:

OK

Response:

Code: 200

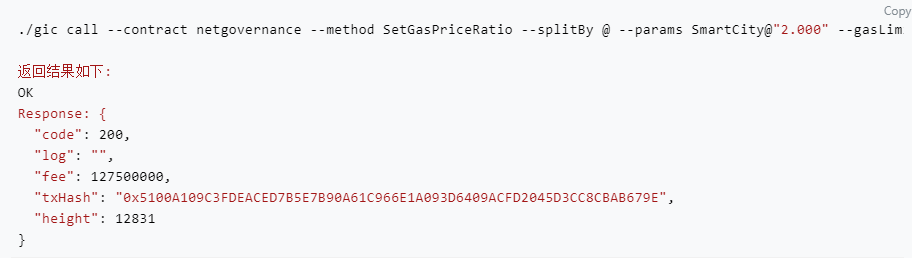
Key: /genesis/chainid

Value: "local[smartCity]"

可以看出原主链观察者的chainid已经变为创世侧链的chainid

## **2.3侧链治理**





### 2.3.1 侧链管理快速入门

1. **调整侧链公开的URL**

查询侧链的所有者scowner在主链local上的账户余额，以确保能够支付设置侧链OpenURLs所消耗的gas，使用gic工具查询命令如下:

./gic balance --accAddress="localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local

返回结果如下:

OK

Response: [

{

"tokenAddress": "localAJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "9977498750000"

}

]

进行侧链OpenURLs设置，没有设置将导致主链向侧链发送的跨链transaction无法完成，下面以侧链名称smartCity、侧链公开URL["http://ip1:46657"]为例，展示如何使用gic工具完成侧链OpenURLs的设置：

./gic call --contract netgovernance --method SetOpenURLs --splitBy @ --params smartCity@"[\"http://ip1:46657\"]" --gasLimit 100000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 125000000,

"txHash": "0xCE0E3359A967E60A73316E7AE7726D7890AFD05B70FDCA319BC5E50E8E161464",

"height": 11503

}

命令行中选项 params 的值即为设置侧链OpenURLs需要的参数，释义如下：

第一个字段 smartCity 为侧链名称

第二个字段 ["http://ip1:46657"] 为侧链对外开放的 url 列表,他可以是多个,如:["http://ip1:46657", "http://ip2:46657"]

完成侧链 openURLs 的调整后,可以查询侧链的 openURLs 是否设置成功,命令如下:

./gic query -k /sidechain/local[smartCity]/openurls

返回结果如下:

OK

Response:

Code: 200

Key: /sidechain/local[smartCity]/openurls

Value: ["ip1:46657"]

经查询后发现,侧链smartCity的 openURLs 就是上面设置的 urls .

如查询的结果不是上面设置的 urls ,可以通过用设置 urls 时返回的 txhash查询这笔transaction的收据:

./gic tx --txhash 0xCE0E3359A967E60A73316E7AE7726D7890AFD05B70FDCA319BC5E50E8E161464

OK

Response: {

"txHash": "0xCE0E3359A967E60A73316E7AE7726D7890AFD05B70FDCA319BC5E50E8E161464",

"txTime": "2019-11-30 05:42:54.886886465 +0000 UTC",

"code": 200,

"log": "",

"blockHash": "0x3aa75d9b3c08adf5d7140377b50ee02ffdb62agi",

"blockHeight": 4102,

"from": "localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG",

"nonce": 2,

"gasLimit": 100000,

"fee": 125000000,

"note": "",

"messages": [

{

"smcAddress": "local4f5YLBNi336r1zUYZTga9PXVJifDqdg3U",

"smcName": "netgovernance",

"method": "SetOpenURLs(string,[]string)",

"to": "",

"value": ""

}

],

"tags": {

"/0/0/std::fee": {

......

},

"/0/1/netgovernance.setOpenURL": {

"Name": "netgovernance.setOpenURL",

"ContractAddr": "",

"ReceiptBytes": "eyJzaWRlQ2hhaW5JRCI6ImxvY2FsW3l5XSIsIm9wZW5VUkxzIjpbImh0dHA6Ly8xNzIuMTYuNDIuMTM2OjQ2NjU3IiwiaHR0cDovLzE3Mi4xNi40Mi4xMzU6NDY2NTciXX0=",

"ReceiptHash": "0x45F4E7344F40ED8E27BCE4ED63B0D5678BF4B095970F6A3B901490E67DF60D3D",

"Receipt": {

"openURLs": [

"http://ip1:46657"

],

"sideChainID": "local[SmartCity]"

}

},

"/1/0/totalFee": {

......

},

"/1/1/transferFee": {

......

},

"/1/2/transferFee": {

......

},

"/1/3/transferFee": {

......

},

"/1/4/transferFee": {

......

}

}

}

可以找到tag中为netgovernance.setOpenURL的收据,检查一下Receipt字段中的 openURLs与sideChainID是否正确.

1. **调整侧链的燃料价格比例**

使用gic工具可以调整侧链的燃料价格比例,下面以侧链名称为smartCity、燃料价格比例2.000为例展示如何调整侧链的燃料价格比例:

./gic call --contract netgovernance --method SetGasPriceRatio --splitBy @ --params SmartCity@"2.000" --gasLimit 100000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 127500000,

"txHash": "0x5100A109C3FDEACED7B5E7B90A61C966E1A093D6409ACFD2045D3CC8CBAB679E",

"height": 12831

}

命令行中选项 params 的值即为侧链创世时需要的参数，释义如下：

第一个字段 SmartCity 为侧链名称.

第二个字段 "2.000" 为燃料价格比例,注意该格式必须是精确到小数点后三位的正数.

如果想要查询我们上面的操作是否成功了,可以使用以下命令查询:

在主链上用以下命令查询:

./gic query -k /sidechain/local[SmartCity]/chaininfo

返回结果如下:

OK

Response:

Code: 200

Key: /sidechain/local[SmartCity]/chaininfo

Value: {"sideChainName":"SmartCity","chainID":"local[SmartCity]","NodeNames":["node1"],"orgName":"SmartCity","owner":"localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","height":748,"status":"ready","gasPriceRatio":"2.000"}

在侧链上用以下命令

./gic query -k /genesis/gaspriceratio --chainid local[SmartCity]

返回结果如下

OK

Response:

Code: 200

Key: /genesis/gaspriceratio

Value: "2.000"

经查询可以看到调整侧链SmartCity的 gasPriceRatio 成功。如果查询结果与预期的不一致,可以用以下办法查询收据:

./gic tx --chainid local --txhash 0x5100A109C3FDEACED7B5E7B90A61C966E1A093D6409ACFD2045D3CC8CBAB679E

OK

Response: {

"txHash": "0x5100A109C3FDEACED7B5E7B90A61C966E1A093D6409ACFD2045D3CC8CBAB679E",

"txTime": "2019-11-30 05:58:31.242099551 +0000 UTC",

"code": 200,

"log": "",

"blockHash": "0x00b041f365468cbd9f9d911e20e1f9ef643dd0c4",

"blockHeight": 5014,

"from": "localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG",

"nonce": 3,

"gasLimit": 100000,

"fee": 127500000,

"note": "",

"messages": [

{

"smcAddress": "local4f5YLBNi336r1zUYZTga9PXVJifDqdg3U",

"smcName": "netgovernance",

"method": "SetGasPriceRatio(string,string)",

"to": "",

"value": ""

}

],

"tags": {

"/0/0/std::fee": {

......

},

"/0/1/netgovernance.setGasPriceRatio": {

"Name": "netgovernance.setGasPriceRatio",

"ContractAddr": "",

"ReceiptBytes": "eyJjaGFpbk5hbWUiOiJ5eSIsImNoYWluSUQiOiJsb2NhbFt5eV0iLCJnYXNQcmljZVJhdGlvIjoiMi4wMDAifQ==",

"ReceiptHash": "0xA20E055CA12756C18D74A17052BA2FE1A505380086158F549EC78F605165790A",

"Receipt": {

"chainID": "local[SmartCity]",

"chainName": "SmartCity",

"gasPriceRatio": "2.000"

}

},

"/0/2/std::fee": {

......

},

"/0/3/igi::packet/local->local[SmartCity]": {

"Name": "igi::packet/local->local[SmartCity]",

"ContractAddr": "",

"ReceiptBytes": "",

"ReceiptHash": "0x4B9FF741906AABC2977AA366F048C00F77E6CDFCC44F00B1FB0A8222BB27C5F2",

"Receipt": {

"contractName": "netgovernance",

"fromChainID": "local",

"igiHash": "6A16A363A54215F5C0E1F632D689FBD60C614FF24D00514AC15B363CE2D36A05",

"orgID": "orgJgaGConUyK81zibntUBjQ33PKctpk1K1G",

"queueID": "local->local[SmartCity]",

"receipts": [

{

"key": "LzAvbmV0Z292ZXJuYW5jZS5zZXRHYXNQcmljZVJhdGlv",

"value": "eyJuYW1lIjoibmV0Z292ZXJuYW5jZS5zZXRHYXNQcmljZVJhdGlvIiwiY29udHJhY3RBZGRyZXNzIjoibG9jYWw0ZjVZTEJOaTMzNnIxelVZWlRnYTlQWFZKaWZEcWRnM1UiLCJyZWNlaXB0Qnl0ZXMiOiJleUpqYUdGcGJrNWhiV1VpT2lKNWVTSXNJbU5vWVdsdVNVUWlPaUpzYjJOaGJGdDVlVjBpTENKbllYTlFjbWxqWlZKaGRHbHZJam9pTWk0d01EQWlmUT09IiwicmVjZWlwdEhhc2giOiJBMjBFMDU1Q0ExMjc1NkMxOEQ3NEExNzA1MkJBMkZFMUE1MDUzODAwODYxNThGNTQ5RUM3OEY2MDUxNjU3OTBBIn0="

}

],

"seq": 1,

"state": {

"log": "",

"status": "NoAck",

"tag": "NotifyPending"

},

"toChainID": "local[SmartCity]",

"type": "notify"

}

},

"/1/0/totalFee": {

......

},

"/1/1/transferFee": {

......

},

"/1/2/transferFee": {

......

},

"/1/3/transferFee": {

......

},

"/1/4/transferFee": {

......

}

}

}

找到收据名称为netgovernance.setGasPriceRatio,看Receipt字段的chainID以及gasPriceRatio是否设置正确.

### 2.4.1 节点管理快速入门

1. **节点准备**
2. 经过之前的操作，已经搭建了一个侧链，但是侧链只有一个验证者节点，接下来需要增加这条侧链的验证者节点以进行共识。
3. 需要搭建侧链观察者节点，以便可以将侧链的观察者节点转为侧链的验证者节点。
4. **导出节点公钥**

在准备好的侧链观察者节点机器上使用以下命令导出节点公钥，节点公钥会在将侧链的观察者转为验证者时用到，导出节点公钥步骤参考：2.2.1章节

1. **添加验证者节点**注：在增加侧链验证者节点数（验证者节点小于四个）时，一定要确保添加的节点公钥是正确的,否则会导致该侧链无法正常工作.

如果想要添加一个验证者节点,需要通过上面得到的节点公钥使用以下命令添加，下面以验证者节点名称node2、节点公钥0x0d180d63841102960cd86d88cf1f79e6695891d85385b4abf25f97b94cfaf2c0、奖励地址local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG、节点权重10 为例展示如何添加验证者节点：

./gic call --contract governance --method NewValidator --params node2@0x0d180d63841102960cd86d88cf1f79e6695891d85385b4abf25f97b94cfaf2c0@local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG@10 --gasLimit 1000000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local[SmartCity]

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 255000000,

"txHash": "0xC6619B5D446CB718E555FB51C7E7FDCA46EE57E28683C979413EBA1F0FDF6A64",

"height": 3819

}

命令行中选项 params 的值即为侧链创世时需要的参数，释义如下：

第一个字段 node2 是你添加的这个验证者节点的节点名称

第二个字段 0x0d180d63841102960cd86d88cf1f79e6695891d85385b4abf25f97b94cfaf2c0 是 0x加上你上面得到的该节点的公钥

第三个字段 local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG 是奖励地址

第四个字段 10 是该节点的权重

通过以下命令检查侧链SmartCity是否成功添加该节点:

./gic query -k /igi/local[SmartCity] --chainid local[SmartCity]

返回结果如下

OK

Response:

Code: 200

Key: /igi/local[SmartCity]

Value: {"local[SmartCity]33KDwKPQrtwdgTejxeGspyR7u4fefXTb7":{"nodepubkey":"0D180D63841102960CD86D88CF1F79E6695891D85385B4ABF25F97B94CFAF2C0","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node2","nodeaddr":"local[SmartCity]33KDwKPQrtwdgTejxeGspyR7u4fefXTb7"},"local[SmartCity]91TcFC3vPqsS3DWh5TksV1xCKZLWEftpD":{"nodepubkey":"10366FE4D47A9EA4AD54C864143B7FBD3ECA247995C663BCC3C8835D29317A30","power":10,"rewardaddr":"local[SmartCity]2V6b1EW2cExW5UtuUX9XXQhJCaaxXzLV6","name":"node1","nodeaddr":"local[SmartCity]91TcFC3vPqsS3DWh5TksV1xCKZLWEftpD"}}

可见侧链SmartCity已成功添加新节点 node2.

注:当前节点总数小于四个时,新增节点的权重必须等于10,否则会加节点失败。

1. **调整验证者节点权重**

在调整验证者节点权重时，必须保证当前节点大于等于四个，下面以节点公钥0x2113d7a30d47a08ca79c8ebgi32a9c73a65a488b3f93da3a1eb42c57b35f9c9f、节点权重12为例展示如何调整验证者节点权重：

./gic call --contract governance --method SetPower --params 0x2113d7a30d47a08ca79c8ebgi32a9c73a65a488b3f93da3a1eb42c57b35f9c9f@12 --gasLimit 1000000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local[SmartCity]

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 175000000,

"txHash": "0xB0BB34E92ED02745460D4C6FD98F0B70B9B199A7B11B54C64F0281FA48D520F6",

"height": 830

}

命令行中选项 params 的值即为侧链创世时需要的参数，释义如下：

第一个字段 0x2113d7a30d47a08ca79c8ebgi32a9c73a65a488b3f93da3a1eb42c57b35f9c9f 是 0x加上该节点的公钥

第二个字段 12 是该节点的权重

若想确认节点权重是否设置成功，可以通过下面的命令查询验证:

./gic query -k /igi/local[SmartCity] --chainid local[SmartCity]

返回结果如下:

OK

Response:

Code: 200

Key: /igi/local[SmartCity]

Value: {"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE":{"nodepubkey":"E1464972D492C1FB916336658068F0B7A92B458A6D7D0D010BDE5F035BC32A90","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node2","nodeaddr":"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE"},"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS":{"nodepubkey":"F3D77C12669A83AF7E922A7190001A40304AF01AD545A464D786A332761FB494","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node4","nodeaddr":"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS"},"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q":{"nodepubkey":"2113D7A30D47A08CA79C8EBBC32A9C73A65A488B3F93DA3A1EB42C57B35F9C9F","power":12,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node3","nodeaddr":"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q"},"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18":{"nodepubkey":"D781A3CE164052F8E103D5F9DE216659438ABF2D2F956FF76483A7E6B9C93117","power":10,"rewardaddr":"local[SmartCity]8LtT8AonWgJ8nMCEdAR5UGrbRfUmuoeiz","name":"node1","nodeaddr":"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18"}}

经查询node3节点的权重成功修改为了12.

注：如果当前节点数小于4个时,不允许修改节点的权重。

1. **删除验证者节点**

删除节点即为将该节点的权重设为0。

在删除节点前,可以查询一下当前节点情况：

./gic query -k /igi/local[SmartCity] --chainid local[SmartCity]

返回结果如下:

OK

Response:

Code: 200

Key: /igi/local[SmartCity]

Value: {"local[SmartCity]33KDwKPQrtwdgTejxeGspyR7u4fefXTb7":{"nodepubkey":"0D180D63841102960CD86D88CF1F79E6695891D85385B4ABF25F97B94CFAF2C0","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node5","nodeaddr":"local[SmartCity]33KDwKPQrtwdgTejxeGspyR7u4fefXTb7"},"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE":{"nodepubkey":"E1464972D492C1FB916336658068F0B7A92B458A6D7D0D010BDE5F035BC32A90","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node2","nodeaddr":"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE"},"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS":{"nodepubkey":"F3D77C12669A83AF7E922A7190001A40304AF01AD545A464D786A332761FB494","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node4","nodeaddr":"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS"},"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q":{"nodepubkey":"2113D7A30D47A08CA79C8EBBC32A9C73A65A488B3F93DA3A1EB42C57B35F9C9F","power":12,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node3","nodeaddr":"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q"},"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18":{"nodepubkey":"D781A3CE164052F8E103D5F9DE216659438ABF2D2F956FF76483A7E6B9C93117","power":10,"rewardaddr":"local[SmartCity]8LtT8AonWgJ8nMCEdAR5UGrbRfUmuoeiz","name":"node1","nodeaddr":"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18"}}

此处以删除node5（即节点公钥为0d180d63841102960cd86d88cf1f79e6695891d85385b4abf25f97b94cfaf2c0的节点）节点为例,:

./gic call --contract governance --method SetPower --params 0x0d180d63841102960cd86d88cf1f79e6695891d85385b4abf25f97b94cfaf2c0@0 --gasLimit 1000000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local[SmartCity]

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 175000000,

"txHash": "0x2CA4F72EBA8D525E8AFF3B325024E1E87D673EF2FF6B8287AA87487657AF9AF5",

"height": 1352

}

通过以下命令可以验证节点删除的情况:

./gic query -k /igi/local[SmartCity] --chainid local[SmartCity]

返回结果如下:

OK

Response:

Code: 200

Key: /igi/local[SmartCity]

Value: {"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE":{"nodepubkey":"E1464972D492C1FB916336658068F0B7A92B458A6D7D0D010BDE5F035BC32A90","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node2","nodeaddr":"local[SmartCity]CFCHxFV6z6djWJR5FH1uSrJ8eN7AYj6gE"},"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS":{"nodepubkey":"F3D77C12669A83AF7E922A7190001A40304AF01AD545A464D786A332761FB494","power":10,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node4","nodeaddr":"local[SmartCity]F5GqPDn294pdydBfxSgPEspbU5RPpwnUS"},"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q":{"nodepubkey":"2113D7A30D47A08CA79C8EBBC32A9C73A65A488B3F93DA3A1EB42C57B35F9C9F","power":12,"rewardaddr":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","name":"node3","nodeaddr":"local[SmartCity]Gbh2JuVFg6orPcpCpQvAdv5gYT5kFQV3Q"},"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18":{"nodepubkey":"D781A3CE164052F8E103D5F9DE216659438ABF2D2F956FF76483A7E6B9C93117","power":10,"rewardaddr":"local[SmartCity]8LtT8AonWgJ8nMCEdAR5UGrbRfUmuoeiz","name":"node1","nodeaddr":"local[SmartCity]MwKS9rCK2nBNsevTdBBeiqKCrqQRkAK18"}}

经查询,这个节点node5已成功删除.

注：当前节点数小于四个时,不允许删除节点

## **2.4跨链trasnsfer**

在本章，我们将以搭建的侧链SmartCity和侧链EShop为基础，体验一下如何进行跨链trasnsfer。

本区块链支持三种模式的跨链trasnsfer，分别是：主链到侧链、侧链到主链以及侧链到侧链的跨链trasnsfer。



### 2.4.1准备账户

为侧链准备一个转入转出账户，现在我们假设创建的账户用户名为 scAccount，密码为 Ab1@Cd3$，在侧链 SmartCity 的账户地址为 local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG，在侧链EShop的账户地址为 local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG。

### 2.4.2基础通证跨链trasnsfer

1. **主链到侧链trasnsfer**

主链到侧链的trasnsfer，我们使用主链的owner账户去给 步骤1 中准备的 SmartCity 侧链的转入转出账户地址 local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG trasnsfer，trasnsfer前我们先查询一下该账户在侧链 SmartCity上的账户余额，如下我们以主链的基础通证为 LOC 为例展示如何查询账户余额：

./gic balance --accAddress="local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local[SmartCity]

返回结果如下表示查询成功:

OK

Response: [

{

"tokenAddress": "local[SmartCity]AJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "0"

}

]

如上我们可见新建账户的基础通证余额为0。

下面我们还是以基础通证 `LOC` 为例，展示如何从主链的`owner`账户向侧链的`scAccount`账户进行主链到侧链的跨链trasnsfer操作:

./gic transfer --name owner --password Ab1@Cd3$ --token LOC --gasLimit 100000 --to local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG --value 100 --chainid local

返回结果如下表示操作成功:

OK

Response: {

"code": 200,

"log": "",

"fee": 5000000,

"txHash": "0x6C8C4A3324C87ECE2EC85F03D25D3994AB5BE31183AED6F7D8044CA2CCC0F25F",

"height": 25974

}

命令行中选项 to 为侧链账户 scAccount 的账户地址，value为跨链trasnsfer金额，单位为 LOC。

等待5秒左右，我们再到侧链上去查询一下账户地址local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG对应的账户余额:

./gic balance --accAddress="local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local[SmartCity]

返回结果如下表示操作成功:

OK

Response: [

{

"tokenAddress": "local[SmartCity]AJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "100000000000"

}

]

因为余额显示的单位为 cong ，所以需要在数值 100 基础上乘上10的9次方。如果查询账户余额未到账(可能是还未到账),可以通过获取跨链事务Hash到目标链查询这笔跨链事务的状态(参考2.4.3 跨链事务).如果该笔跨链事务状态显示已完成,但是再次查询余额依然未到账,可以根据第二章跨链事务查询transaction失败原因.

1. **侧链到主链trasnsfer**

我们上面给侧链的scAccount转入了100个LOC，现在如果我们想把资金转回到主链的话可以通过侧链向主链trasnsfer的方式完成，首先我们在主链查询账户scAccount 对应的地址localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG在主链上的账户余额：

./gic balance --accAddress="localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local

返回结果如下:

OK

Response: [

{

"tokenAddress": "localAJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "0"

}

]

然后用以下命令进行trasnsfer：

./gic transfer --name scAccount --password Ab1@Cd3$ --token LOC --gasLimit 100000 --to localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG --value 10 --chainid local[SmartCity]

返回结果如下表示成功:

OK

Response: {

"code": 200,

"log": "",

"fee": 10000000,

"txHash": "0x20B5238E777BDA4D7F883C0698C608C30874A080DCB2FAAAFD1C8EF4FBD98098",

"height": 6225

}

查询一下收款地址的账户余额：

./gic balance --accAddress="localHMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local

OK

Response: [

{

"tokenAddress": "localAJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "10000000000"

}

]

经查询发现将资金从侧链转回主链成功,确认跨链trasnsfer成功.如果查询账户余额未到账(可能是还未到账)，可以通过获取跨链事务Hash到目标链查询这笔跨链事务的状态(参考第二章 跨链事务)。如果该笔跨链事务状态显示已完成，但是再次查询余额依然未到账，可以根据第二章跨链事务查询transaction失败原因。

1. **侧链到侧链trasnsfer**

那么如果想要从一个侧链往另一个侧链trasnsfer,我们应该怎么做呢?

首先我们准备好另一个侧链EShop，

首先我们先查询一下`EShop`链的地址`local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG`的账户余额：

./gic balance --accAddress="local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local[EShop]

返回结果如下表示操作成功:

OK

Response: [

{

"tokenAddress": "local[EShop]AJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "0"

}

]

然后我们开始从侧链SmartCity给侧链EShoptrasnsfer：

./gic transfer --name scAccount --password Aa1@Cd3$ --token LOC --gasLimit 100000 --to local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG --value 10 --chainid local[SmartCity]

返回结果如下表示操作成功:

OK

Response: {

"code": 200,

"log": "",

"fee": 10000000,

"txHash": "0x25FEA54305EDE927A7FDC47AE28FE3B2D29A8FB54A85080CBD7C738A7B0FB7C8",

"height": 6559

}

在侧链EShop上再次查询local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG的账户余额：

./gic balance --accAddress="local[EShop]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="LOC" --chainid local[EShop]

返回结果如下表示操作成功:

OK

Response: [

{

"tokenAddress": "local[EShop]AJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA",

"tokenName": "LOC",

"balance": "10000000000"

}

]

确认跨链trasnsfer成功。如果查询账户余额未到账(可能是还未到账)，可以通过获取跨链事务Hash到目标链查询这笔跨链事务的状态(参考第二章 跨链事务)。如果该笔跨链事务状态显示已完成,但是再次查询余额依然未到账，可以根据第二章跨链事务查询transaction失败原因。

1. **通证激活**

如果我们想我们的侧链支持某种类型的标准通证，我们可以使用通证激活功能；如下我们以在侧链SmartCity上激活通证TST为例，展示侧链激活通证的操作：

./gic call --contract token-issue --method Activate --splitBy @ --params "TST"@"SmartCity" --gasLimit 100000 --orgName genesis --name scowner --password SCOwner@2019 --chainid local

返回结果如下表示操作成功:

OK

Response: {

"code": 200,

"log": "",

"fee": 131000000,

"txHash": "0x177CE19688464BCCC35F78288B28E1D0E766ACF302A7A3E994D86100F7D2CAAE",

"height": 47177

}

我们可以到侧链上查询一下，是否生成了该通证对应的信息：

./gic query -k "/token/name/TST" --chainid local[SmartCity]

返回结果如下:

OK

Response:

Code: 200

Key: /token/name/TST

Value: "local[SmartCity]8GPJL6ipTyYRNr5xTyGWWFayd2k1EoLYu"

由上可以得到该通证的地址，我们还可以用以下命令查该通证的信息：

./gic query -k "/token/local[SmartCity]8GPJL6ipTyYRNr5xTyGWWFayd2k1EoLYu" --chainid local[SmartCity]

返回结果如下:

OK

Response:

Code: 200

Key: /token/local[SmartCity]8GPJL6ipTyYRNr5xTyGWWFayd2k1EoLYu

Value: {"address":"local[SmartCity]8GPJL6ipTyYRNr5xTyGWWFayd2k1EoLYu","owner":"local[SmartCity]ETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL","name":"TST","symbol":"TST","totalSupply":0,"addSupplyEnabled":false,"burnEnabled":false,"gasprice":2500}

至此,我们完成了通证Diamond Coin的激活,我们可以用以下命令尝试该通证的trasnsfer:

./gic transfer --name owner --password Ab1@Cd3$ --token "TST" --gasLimit 100000 --to local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG --value 100 --chainid local

返回结果如下:

OK

Response: {

"code": 200,

"log": "",

"fee": 5250000,

"txHash": "0xE1B35FF40EFA83A972C2E1260E2677937025842BCA87B5C61F515CEA2B1BFB40",

"height": 48518

}

通过上述命令我们完成了通证Diamond Coin的跨链trasnsfer,我们试着查询一下该笔transaction是否成功到账:

./gic balance --accAddress="local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG" --tokenName="Diamond Coin" --chainid local[SmartCity]

返回结果如下:

OK

Response: [

{

"tokenAddress": "local[SmartCity]8GPJL6ipTyYRNr5xTyGWWFayd2k1EoLYu",

"tokenName": "TST",

"balance": "100000000000"

}

]

经查询,通证TST成功到账.如果查询账户余额未到账,可以通过查询跨链事务来了解这笔跨链trasnsfer的状态,详细参考第二章 跨链事务.

### 2.4.3跨链事务

1. 跨链事务Hash获取

跨链事务 Hash 即igiHash,我们可以通过它获取跨链事务状态以及跨链事务信息.

我们以 2.1 中主链给侧链SmartCitytrasnsfer为例子,我们得到trasnsfer高度为 25974,我们可以通过以下命令获取igiHash:

./gic block -t 25974

OK

Response: {

"blockHeight": 25974,

"blockHash": "0xf8f381cd2c5bae781aebd8524edad20af4654158",

"parentHash": "0xf838ca7c49b12e8cd3839fe7ce3d0f8d2ee72bd8",

"chainID": "local",

"validatorHash": "0xf2c0a2f00f78c4b8cef2c7e56dd62e191ab7d0b7",

"consensusHash": "0xf66ef1df8ba6dac7a1ecce40cc84e54a1cegi6a5",

"blockTime": "2019-11-28 07:27:33.528200078 +0000 UTC",

"blockSize": 1039,

"proposerAddress": "localMp5Xi3o2rnXjzbCuzo5R4m6qv7fbfNgkW",

"txs": [

{

"txHash": "0x6c8c4a3324c87ece2ec85f03d25d3994ab5be31183aed6f7d8044ca2ccc0f25f",

"txTime": "2019-11-28 07:27:33.528200078 +0000 UTC",

"code": 200,

"log": "",

"blockHash": "0xf8f381cd2c5bae781aebd8524edad20af4654158",

"blockHeight": 25974,

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"nonce": 61,

"gasLimit": 100000,

"fee": 5000000,

"note": "",

"messages": [

......

],

"tags": {

......,

"/0/4/igi::packet/local->local[SmartCity]": {

"Name": "igi::packet/local->local[SmartCity]",

"ContractAddr": "",

"ReceiptBytes": "",

"ReceiptHash": "0x075860DF0B94E3A2D346150DD1E17F99D4B171F1B689A87C39CA0977AFBA8D40",

"Receipt": {

"contractName": "token-basic",

"fromChainID": "local",

"igiHash": "1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397",

"orgID": "orgJgaGConUyK81zibntUBjQ33PKctpk1K1G",

"queueID": "local->local[SmartCity]",

"receipts": [

{

"key": "LzAvc3RkOjp0cmFuc2Zlcg==",

"value": ""

},

{

"key": "LzEvdG9rZW5iYXNpYy5Bc3NldENoYW5nZQ==",

"value": ""

}

],

"seq": 5,

"state": {

"log": "",

"status": "NoAckWanted",

"tag": "RecastPending"

},

"toChainID": "local[SmartCity]",

"type": "tcctx"

}

},

......

}

}

]

}

找到收据名称包含igi::packet的收据,在Receipt结构中得到igiHash.

1. 跨链事务数据

通过上面查询到的igiHash: 1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397,我们可以用以下命令得到跨链事务数据:

./gic query -k /igi/1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397/packets

返回结果如下:

OK

Response:

Code: 200

Key: /igi/1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397/packets

Value: [

......,{

"fromChainID": "local",

"toChainID": "local[SmartCity]",

"queueID": "",

"seq": 0,

"orgID": "orgJgaGConUyK81zibntUBjQ33PKctpk1K1G",

"contractName": "token-basic",

"igiHash": "1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397",

"type": "tcctx",

"state": {

"status": "NoAck",

"tag": "Confirmed",

"log": ""

},

"receipts": [{

"key": "LzAvc3RkOjp0cmFuc2Zlcg==",

"value": ""

}, {

"key": "LzEvdG9rZW5iYXNpYy5Bc3NldENoYW5nZQ==",

"value": ""

}, {

"key": "LzEvc3RkOjp0cmFuc2Zlcg==",

"value": ""

}, {

"key": "LzIvdG9rZW5iYXNpYy5Bc3NldENoYW5nZQ==",

"value": ""

}, {

"key": "LzEvdG9rZW5iYXNpYy5Bc3NldENoYW5nZQ==",

"value": ""

}]

}

]

1. 跨链事务数据解析

根据上面的操作,我们成功拿到了一笔跨链transaction的跨链事务数据,我们现在分析一下拿到的这个跨链事务数据.

type Packet struct {

FromChainID string // 发起链的链ID（A）

ToChainID string // 目标链的链ID（B）

QueueID string // 跨链通讯队列（A->B）

Seq uint64 // (A->B)这个队列上跨链通讯包序号,从0开始累加+1

OrgID string // 组织ID

ContractName string // 合约名称

IgiHash types.Hash // 跨链事务哈希,通过此哈希从区块链上确认最终执行结果

Type string // 跨链通讯类型："tcctx", "notify"

State State // 状态

Receipts []types.KVPair // 当前状态下需要传递到另一侧的应用层数据

}

type State struct {

Status string // 状态："NoAckWanted", "NoAck"

Tag string // 表示业务层的状态标识：

Log string // 异常日志

}

注:

Type 即跨链通讯类型："tcctx" 表示transaction, "notify"表示通知

Status 即跨链事务状态：

"NoAckWanted"跨链通讯协议状态，申明需要接收方返回应答.

"NoAck" 表示跨链通讯协议状态，申明不需要接收方返回应答.

Tag 即跨链事务的业务层状态标识:

"NotifyPending" 跨链通知在发起侧的状态，声明下一步需要在目标侧执行通知业务.

"Success" 跨链通知在目标侧的状态，声明业务层已经完成了通知操作.

"Failure"跨链通知在目标侧的状态，声明无法执行业务层的通知操作，原因可能为合约版本不一致等.

"RecastPending" 跨链transaction在发起侧的状态，声明下一步需要在目标侧执行重铸业务，或者由枢纽链进行中转), "ConfirmPending" 跨链transaction在中转侧及目标侧的状态，声明下一步需要在中转侧执行确认中转业务，或者需要在发起侧执行最终确认业务.

"Confirmed" 跨链通知在发起侧的状态，声明业务层已经成功完成了跨链操作.

"CancelPending" 跨链transaction在中转侧及目标侧的状态，声明下一步需要在中转侧执行取消中转业务，或者需要在发起侧执行最终取消业务.

"Canceled" 跨链transaction在发起侧的状态，声明业务层已经成功取消了跨链操作.

该跨链事务数据是由 Packet 结构的切片组成(如图上所示),一个 Packet 结构代表一个事务,完整的跨链事务一般由一个到多个这样的事务组成(通常情况通知在通知发起链只有1个这样的事务,在通知接收链有2个; transaction在transaction发起链有3个这样的事务,在目标链只有2个这样的事务).

1. 跨链收据解析

在上面我们已经得到了一个跨链事务数据,里面有个字段 receipts 的切片,里面存储的数据是跨链transaction的收据,其中除了跨链transaction的收据还有手续费的收据,我们这里只解析跨链transaction的收据:

{

"key": "LzEvdG9rZW5iYXNpYy5Bc3NldENoYW5nZQ==",

"value": ""

}

上面这段收据是从 receipts 字段中拿到的一个收据,通过base64解码我们可以知道这是一个通证变更收据即收据名称为tokenbasic.AssetChange,我们将 value 中的值通过base64解码后得到收据信息:

{"name":"tokenbasic.AssetChange","contractAddress":"localHLDhqyoJ1vRuo8re9GxFAFU2Bksw2kNup","receiptBytes":"","receiptHash":"CFFCDB653B9944C4508DAD753EB195857EA4619E08094EF3470D4790A6F7CE40"}

然后我们再把 receiptBytes字段的值通过base64解码后得到收据:

{"version":"2.1","type":"destroy","token":"localAJrbk6Wdf7TCbunrXXS5kKvbWVszhC1TA","from":"localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL","to":"local[SmartCity]HMpSyt2Jk5BiFiR2j45nGGjToF9miUVaG","value":100000000000,"igiHash":"1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397","changeItems":[{"ChainID":"local","Address":"local332Br8KKq5g47aksZvmuCCw62uoC3HSb6","PeerChainID":"local[SmartCity]","PeerChainBalance":120000000000}]}

注:

type 为通证变更的类型:

"lock" 表示在transaction发起链,该笔资金已被锁定

"recast" 表示在transaction目标链,该笔资金已被重铸

"destory" 表示在transaction发起链,该笔资金已被销毁

PeerChainID 为对侧链的ChainID

PeerChainBalance 为对侧链的该通证的总余额(这是整个链的总余额, 通证为 token字段对应的通证)

注:在 receipts 字段中有多个通证变更收据,这是因为它把前一个链发送给它的收据给保存到该字段了(即第二个 packet 中的收据包含第一个 packet 的收据, 第三个 packet 中的收据包含第二个 packet 中的收据).总的来说如果你想看这笔跨链transaction最后的收据,你应该看最后一个通证变更收据.

### 2.4.4结果查询

通过上面的到的igiHash 1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397,我们使用以下命令可以得到这笔跨链事务的当前状态.

./gic query -k /igi/1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397/state

返回结果如下:

OK

Response:

Code: 200

Key: /igi/1BA13B9559EFC9A2B6E0FD589ED58A5DD829BAD9690986F3319D60EA70CF1397/state

Value: {"status":"NoAck","tag":"Confirmed","log":""}