腾讯云

# 第7章 FISCO BCOS角色与权限

# 目录

CONTENTS

# 第一章 账户管理

CONTENTS

章节

# 第一章 账户管理

CONTENTS

章节

# 账户是什么？

- 同以太坊一样，FISCO BCOS使用账户来标识和区分每一个独立的用户。在采取公私钥体系的区块链系统里，每一个账户对应着一对公钥和私钥。其中，由公钥经哈希等安全的单向性算法计算后，得到的地址字符串被用作该账户的账户名，即账户地址。而仅有用户知晓的私钥则对应着传统认证模型中的密码，这类私钥对应的地址常被称为外部账户或账户，出于方便，下文均称为账户。
- 除外部账户外，还有一类将智能合约部署到区块链中的合约账户，合约账户由外部账户及其账户中的信息计算而来，在部署合约时确定，调用合约接口时需要合约地址，没有私钥。本节主要介绍外部账户。

# 账户的使用场景

- 在FISCO BCOS中，账户有以下使用场景：
  - 交易需要账户的地址，在区块链系统中，每一次对合约写接口的调用都是一笔交易，而每笔交易都需要账户的私钥签名。
  - 权限控制需要账户的地址，FISCO BCOS权限控制模型，根据交易发送者的外部账户地址，判断是否有写入数据的权限。

# 账户的存储形式

- FISCO BCOS支持PEM和PKCS12两种文件格式来存储私钥，用户可根据需求选择账户存储形式。
  - PEM 格式：使用明文存储私钥，文件后缀为pem；
  - PKCS12格式：使用用户提供的口令加密存储私钥，文件后缀为p12。

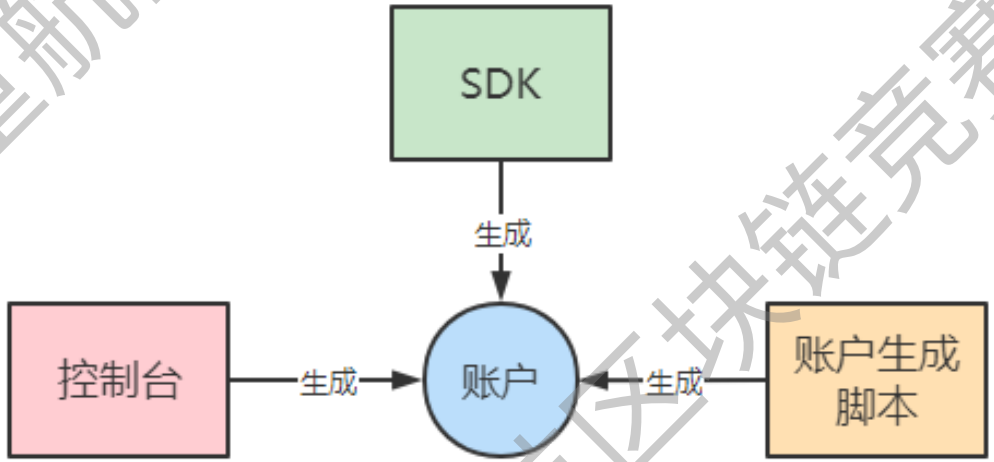
```
$ cat accounts/0x432148430fb0a7bc6a212516e01d802f43ede81a.pem
-----BEGIN PRIVATE KEY-----
MIGEAgEAMBAGByqGSM49AgEGBSuBBAAKBG0wawIBAQQgE2EiqnWvHyRN7pAarsYf
d+ZBe5QoxHVpoM00WWlO+YOhRANCAAQVGgz+mN66OZdp5AlqBf4oEDgCsk7a/Fnu
socZ2lX6hKaDDR45mUs1W8MldpK0ekGnhPrY290Q6kXFR3uPJ89m
-----END PRIVATE KEY-----

$ cat accounts/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.p12
0<82>^A=^B^A^C0<82>^A^C^F       *<86>H<86>÷^M^A^G^A <81>õ^D<81>ò0<81>ï0<81>ì^F
*<86>H<86>÷^M^A^G^A <81>Þ^D<81>Û0<81>Ø0<81>Õ^F^K*<86>H<86>÷^M^A^L
^A^B <81>¬0<81>@0^\^F
*<86>H<86>÷^M^A^L^A^C0^N^D^H¾ªT<82>^?ÃÖ»^B^B^H^@^D<81><88>^H-H@N<8f>ï¢â<8a>û^?
ÍÚ3ÇL EÙU^NæemÌ^R%É<85>¦^B¯^EÊú^Y^Z´p^ZäÉ«¤^VÂ^Qñw^Q^Pª7^U«p»ñ!Vâ<8f>Ð)Ç<8d><9
e>^D;Æ°<83>í"ïR<, õ<88>jºbáTìI<86>^N9&<9d>¸s)"<8a>^G<8c>Ñ»¦¦û^Mÿ^F¸_<84>µ^NÑí¥
²^A7 ^YÇ%æï ^Eº9^YhT<8d>¤m1^W0^U^F     *<86>H<86>÷^M^A ^T1^H^^^F^@k^@e^@y010!0
^F^E+^N^C^B^Z^E^@^D^T^R¯Rx^H-<9b>¾¯KM<90>ik<83>^]Ï%*^@^D^H°£<97>Âz
Ûx^B^B^H^@
```

pem格式私钥可读

PKCS2格式私钥加密

# 账户的生成方式

- FISCO BCOS提供了get_account.sh脚本、控制台和SDK来创建外部账户，同时，控制台和SDK也支持加载创建的账户私钥，用于交易签名。本节主要介绍脚本和控制台生成账户。



# 脚本生成账户

- 获取get_account.sh脚本
  - wget http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/get_account.sh

```
$ wget http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/get_account.sh
--2022-04-24 13:59:38--  http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/get_account.sh
正在解析主机 res.zhonghui.vip (res.zhonghui.vip)... 121.89.193.240
正在连接 res.zhonghui.vip (res.zhonghui.vip)|121.89.193.240|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 301 Moved Permanently
位置：http://res.handge.cn:8999/blockchain/fisco-bcos/01/resource/get_account.sh [跟随至新的 URL]
--2022-04-24 13:59:38--  http://res.handge.cn:8999/blockchain/fisco-bcos/01/resource/get_account.sh
正在解析主机 res.handge.cn (res.handge.cn)... 118.122.97.204
正在连接 res.handge.cn (res.handge.cn)|118.122.97.204|:8999... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 692238 (676K) [application/octet-stream]
正在保存至：“get_account.sh”

get_account.sh                      100%[=============================================================>] 676

2022-04-24 13:59:38 (8.02 MB/s) - 已保存 “get_account.sh” [692238/692238])
```

下载成功

- 修改脚本权限
  - chmod u+x get_account.sh

# 脚本生成账户

- 使用脚本生成PEM格式的私钥账户
  - bash get_account.sh
  - 生成的私钥文件保存在accounts目录

```
$ bash get_account.sh
[INFO] Account Address   : 0x432148430fb0a7bc6a212516e01d802f43ede81a
[INFO] Private Key (pem) : accounts/0x432148430fb0a7bc6a212516e01d802f43ede81a.pem
[INFO] Public  Key (pem) : accounts/0x432148430fb0a7bc6a212516e01d802f43ede81a.public.pem
```

- 指定PEM私钥文件计算账户地址
  - bash get_account.sh -k accounts/0x432148430fb0a7bc6a212516e01d802f43ede81a.pem

```
$ bash get_account.sh -k accounts/0x432148430fb0a7bc6a212516e01d802f43ede81a.pem
[INFO] Account Address   : 0x432148430fb0a7bc6a212516e01d802f43ede81a
```

# 脚本生成账户

- 使用脚本生成PKCS12格式的私钥账户
  - bash get_account.sh -p
  - 生成的私钥文件保存在accounts目录

```
$ bash get_account.sh -p
[INFO] Note: the entered password cannot contain Chinese characters!
Enter Export Password:
Verifying - Enter Export Password:
[INFO] Account Address   : 0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6
[INFO] Private Key (p12) : accounts/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.p12
[INFO] Public  Key (pem) : accounts/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.public.pem
```

输入密码

确认密码

- 指定PKCS12私钥文件计算账户地址
  - bash get_account.sh -P accounts/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.p12

```
$ bash get_account.sh -P accounts/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.p12
Enter Import Password:
[INFO] Account Address   : 0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6
```

输入密码

# 控制台生成账户

- 将控制台连接到节点并启动控制台，使用控制台生成PEM格式的私钥账户
  - newAccount
  - 私钥文件保存在console/account/ecdsa目录

```
[group:1]> newAccount
AccountPath: account/ecdsa/0xca095e590dd772e600ddb139367fda845bf6b695.pem    私钥文件路径
Note: This operation does not create an account in the blockchain, but only creates a local account,
and deploying a contract through this account will create an account in the blockchain
newAccount: 0xca095e590dd772e600ddb139367fda845bf6b695    账户地址
AccountType: ecdsa
```

# 控制台生成账户

- 使用控制台生成PKCS12格式的私钥账户
  - newAccount p12 123456
    - ✓ P12：指定账户的格式为PKCS12
    - ✓ 123456：指定账户的密码
  - 私钥文件保存在console/account/ecdsa目录

```
[group:1]> newAccount p12 123456
AccountPath: account/ecdsa/0xffd36182fc7ae8c6bed3991b6272a45411d5beba.p12   私钥文件路径
Note: This operation does not create an account in the blockchain, but only creates a local account,
and deploying a contract through this account will create an account in the blockchain
newAccount: 0xffd36182fc7ae8c6bed3991b6272a45411d5beba   账户地址
AccountType: ecdsa
```

# 账户的使用

- 控制台启动需要加载私钥，在之前控制台介绍的教程中，为了简化操作，使用了默认启动控制台，默认启动会尝试从account/ecdsa目录下加载一个可用的私钥账户用于发送交易，若加载失败则会创建一个新的PEM格式的账户文件，并将其保存在account/ecdsa目录下。在实际部署中，用户需要自己创建账户并妥善保存账户私钥。一般来讲，控制台有以下几种启动方式：
  - 默认启动：bash console/start.sh
  - 指定群组号启动：bash console/start.sh 2
  - 使用PEM格式私钥文件启动
  - 使用PKCS12格式私钥文件启动

# 账户的使用

- 使用PEM格式私钥文件启动
  - 需要输入参数：群组号、-pem、pem文件路径

```
# 使用PEM格式私钥文件启动
$ bash console/start.sh 1 -pem account/ecdsa/0x432148430fb0a7bc6a212516e01d802f43ede81a.pem
```

- 使用PKCS12格式私钥文件启动
  - 需要输入参数：群组号、-p12、p12文件路径，并输入密码

```
# 使用PKCS12格式私钥文件启动
bash console/start.sh 1 -p12 account/ecdsa/0xd4dd7d4b49f4b5a31afbcefa7f05d9d584c866c6.p12
Enter Export Password:
```

# 账户的使用

- 启动控制台后，也可以通过控制台命令切换账户，通过控制台可查询当前加载的账户，并切换到任一账户。
  - 查看当前账户地址（带有 <= 后缀标记的为当前使用的账户）

```
[group:1]> listAccount
0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2(current account) <=  当前使用的账户
0xca095e590dd772e600ddb139367fda845bf6b695
0xb472f2bee5b7f18c46467fb80ed157d390829cc9
```

  - 切换到PEM格式账户

```
[group:1]> loadAccount account/ecdsa/0xca095e590dd772e600ddb139367fda845bf6b695.pem
Load account account/ecdsa/0xca095e590dd772e600ddb139367fda845bf6b695.pem success!
```

# 账户的使用

- 切换到PKCS12格式账户

```
[group:1]> loadAccount account/ecdsa/0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2.p12 p12
Enter p12 Password:
Load account account/ecdsa/0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2.p12 success!
```

输入密码

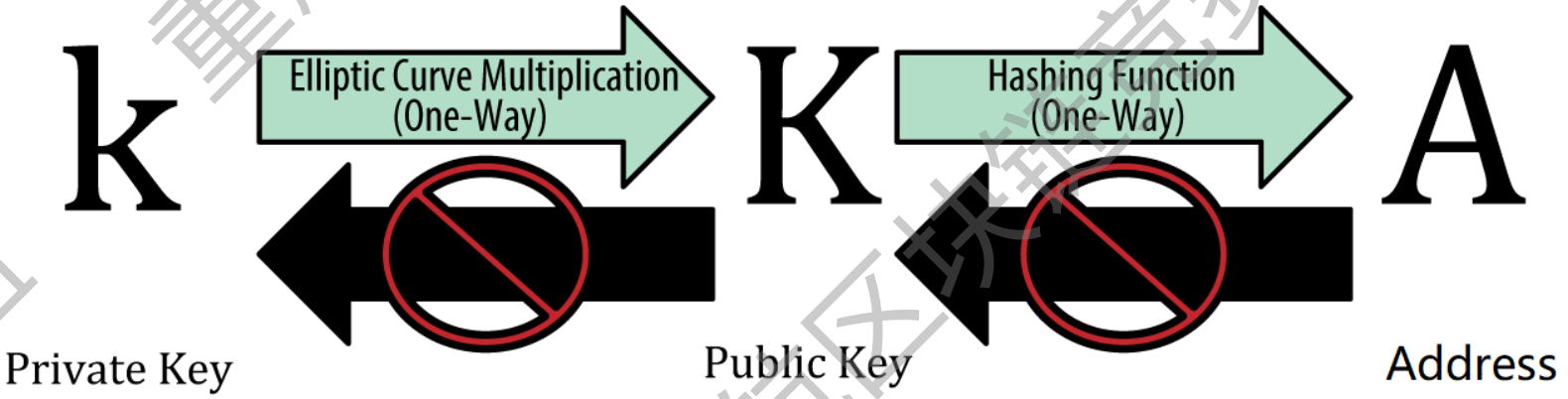
注意加参数“p12”

- 查看当前账户地址

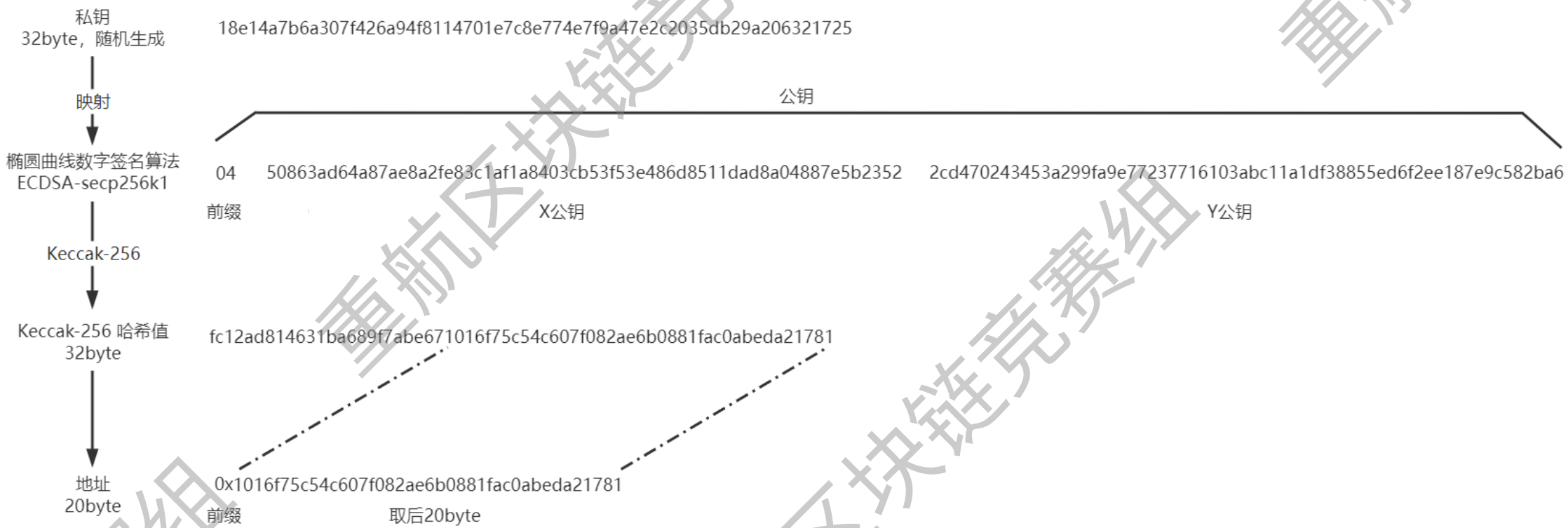
```
[group:1]> getCurrentAccount
0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
```

# 账户的计算过程

- FISCO BCOS的账户地址由ECDSA公钥计算得来，对ECDSA公钥计算keccak-256sum哈希，取计算结果的后20字节的16进制表示作为账户地址，每个字节需要两个16进制数表示，所以账户地址长度为40。FISCO BCOS的账户地址与以太坊兼容。



# 账户的计算过程






腾讯云

# 账户的计算过程

- 使用OpenSSL生成私钥
  - openssl ecparam -name secp256k1 -genkey -noout -out ecprivkey.pem
  - 生成的私钥被保存在ecprivkey.pem文件中
- 查看私钥内容
  - cat ecprivkey.pem

```
$ openssl ecparam -name secp256k1 -genkey -noout -out ecprivkey.pem
$ cat ecprivkey.pem
-----BEGIN EC PRIVATE KEY-----
MHQCAQEEICIcOr5xDRxAT7MrQGzhSCIcxRKxTC3L+LTKRFVn1jwroAcGBSuBBAAK
oUQDQgAE54Gu/XajVSqMjSzoVHdQyDsOG2Kcc9UzC/2im6V+sGt6H6biEdTtTgaB
ItsOF/dG//bt+CHiZibPnxbsQboZXg==
-----END EC PRIVATE KEY-----
```

私钥内容

# 账户的计算过程

- 根据私钥映射公钥
  - openssl ec -in ecprivkey.pem -text -noout 2>/dev/null|sed -n '7,11p' | tr -d ": \n" | awk '{print substr($0,3);}'

```
$ openssl ec -in ecprivkey.pem -text -noout 2>/dev/null|sed -n '7,11p' | tr -d ": \n" | awk '{print substr($0,3);}
e781aefd76a3552a8c8d2ce8547750c83b0e1b629c73d5330bfda29ba57eb06b7a1fa6e211d4ed4e068122db0e17f746fff6edf821e26626c
f9f16ec41ba195e    公钥
```

- 根据公钥计算地址
  - 需要获取keccak-256sum工具：wget http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/kecaak-256sum

# 账户的计算过程

- 根据公钥计算地址
  - openssl ec -in ecprivkey.pem -text -noout 2>/dev/null| sed -n '7,11p' | tr -d ": \n" | awk '{print substr($0,3);}' | ./keccak-256sum -x -l | tr -d ' -' | tail -c 41

```
$ openssl ec -in ecprivkey.pem -text -noout 2>/dev/null| sed -n '7,11p' \
> | tr -d ": \n" | awk '{print substr($0,3);}' | ./keccak-256sum -x -l | tr -d ' -' | tail -c 41
65331ce7747fd190955ffd9e600039eab6ac098e
```

账户地址

  - 可用脚本验证账户地址：bash get_account.sh -k ecprovkey.pem

```
$ bash get_account.sh -k ecprovkey.pem
[INFO] Account Address   : 0x65331ce7747fd190955ffd9e600039eab6ac098e
```

# 第二章 权限控制

CONTENTS

章节

# 权限控制介绍

- 与可自由加入退出、自由交易、自由检索的公有链相比，联盟链有准入许可、交易多样化、基于商业上隐私及安全考虑、高稳定性等要求。因此，联盟链在实践过程中需强调“权限”及“控制”的理念。
- FISCO BCOS平台基于分布式存储，提出分布式存储权限控制的机制，可以灵活，细粒度的方式进行有效的权限管理，为联盟链的治理提供重要的技术手段，以此来体现“权限”的理念。
- 另外，在实际操作中，每个账户使用独立且唯一的公私钥对，发起交易时使用其私钥进行签名，接收方可通过公钥验签知道交易具体是由哪个账户发出，实现交易的可控及后续监管的追溯，以此来体现“控制”的理念。

# 权限控制介绍

- 什么是分布式存储权限控制？
  - 分布式权限控制是基于外部账户(tx.origin)的访问机制，对包括合约部署、表的创建等操作，用表的写操作（插入、更新和删除）进行权限控制，表的读操作不受权限控制。 所以，FISCO BCOS的权限设计是基于表结构的权限管理，为了更直观地观察权限表数据的变化，后续将基MySQL搭建分布式存储的区块链进行权限操作。
- 权限控制的规则
  - 权限控制的最小粒度为表，基于外部账户进行控制；
  - 使用白名单机制，未配置权限的表，默认完全放开，即所有外部账户均有读写权限；
  - 权限设置利用权限表（_sys_table_access_）。权限表中设置表名和外部账户地址，则表明该账户对该表有读写权限，设置之外的账户对该表仅有读权限。

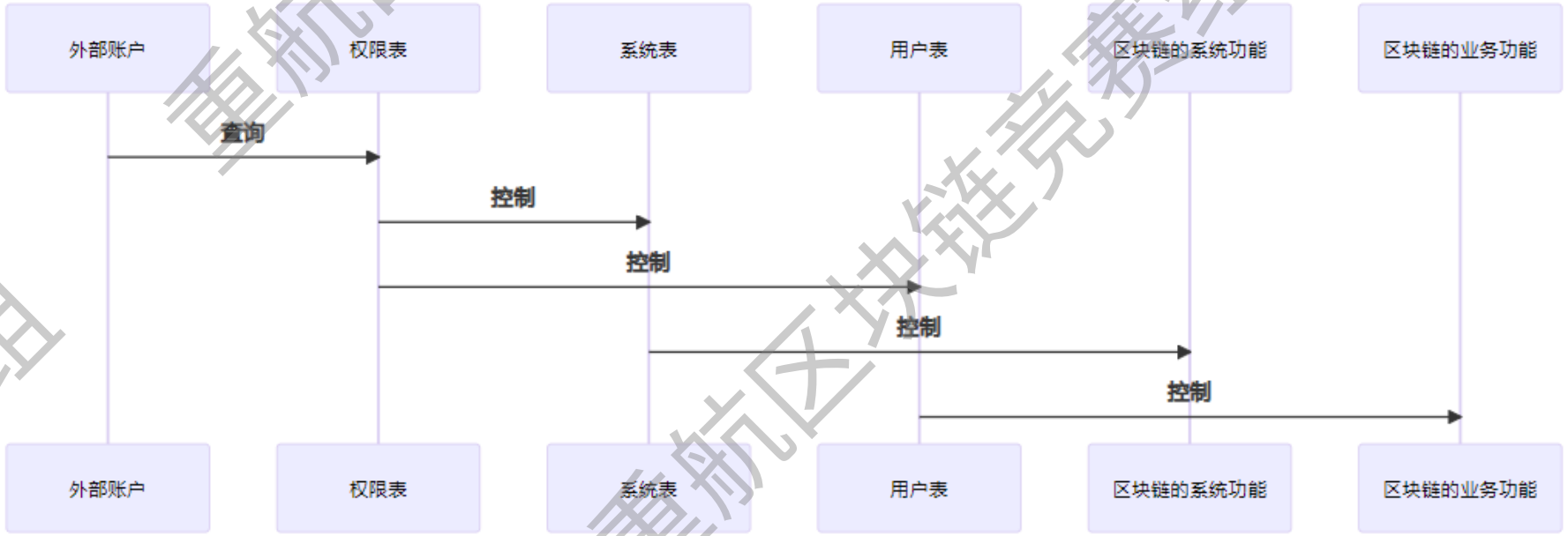
# 权限控制功能分类

- 分布式存储权限控制分为对用户表和系统表的权限控制。用户表指用户合约所创建的表（底层自动添加u_前缀），用户表均可以设置权限。系统表指FISCO BCOS区块链网络内置的表，在建链时被创建，主要系统表如下：

| 表名 | keyField | valueField | 存储数据说明 |
| --- | --- | --- | --- |
| _sys_tables_ | table_name | key_field,value_field | 存储所有表的结构，以表名为主键 |
| _sys_table_access_ | table_name | address,enable_num | 存储每个表的具有写权限的外部账户地址 |
| _sys_consensus_ | name | type,node_id,enable_num | 存储共识节点和观察节点的列表 |
| _sys_cns_ | name | version,address,abi | 存储CNS映射关系 |
| _sys_config_ | key | value,enable_num | 存储需要共识的群组配置项 |

# 权限控制功能设计

- 根据交易信息确定外部账户，外部账户通过查询权限表获取权限相关信息。确定权限后再操作相关的用户表和权限表，系统表用于控制区块链的系统功能，用户表用于控制区块链的业务功能，从而可以控制相关的系统功能和业务功能。如下图所示。



# 第三章 角色权限模型

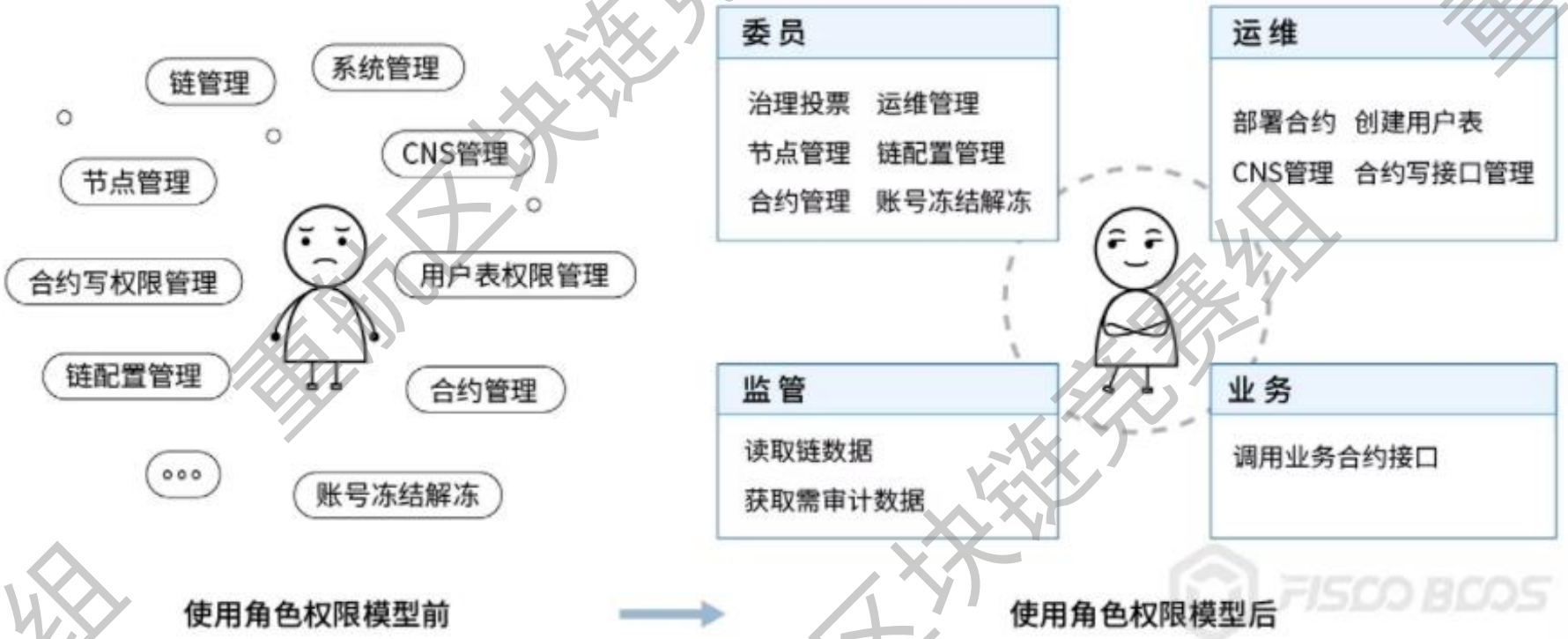
章节

CONTENTS

# 什么是角色权限模型?

- 背景
  - 传统的权限控制是通过控制表的写权限来实现的，由于FISCO BCOS涉及到许多系统表和用户表，用户需要理解对每个权限项的控制内容及如何设置，从而带来了较高的学习成本和使用门槛。
  - 为了降低使用难度，提升用户体验，FISCO BCOS v2.0 新增了角色权限模型，将系统中的不同权限划分到不同的角色，用户根据账号的所属角色来决定其所拥有的权限，同时基于角色引入了链上管理投票模型来治理。

# 什么是角色权限模型?



# 角色定义

- 在FISCO BCOS中，可将区块链的参与者根据角色分为治理方、运维方、监管方和业务方。角色定义避免了既当裁判员又当运动员的情况，又保证了角色之间权责分离、角色互斥。
  - 治理方：治理委员会的委员，简称委员，负责区块链治理；
  - 运维方：复杂区块链运维，该角色由委员添加；
  - 业务方：业务方账号由运维添加到某个合约，可以调用该合约的写接口；
  - 监管方：监管方监管链的运行，能够获取链运行中权限变更记录以及需要审计的数据。

# 角色定义

# 角色对应的权限

| 角色 | 权限操作 | 控制方式 | 默认阈值 | 修改方式 |
| --- | --- | --- | --- | --- |
| 委员 | 增删委员 | 控制写权限表 | 0.5 | 委员投票 |
| | 修改委员权重 | 控制写权限表 | 0.5 | 委员投票 |
| | 修改生效阈值 | 控制写权限表 | 0.5 | 委员投票 |
| | 增删节点（观察/共识） | 控制写共识表 | | 委员可操作 |
| | 修改链配置项 | 控制写配置表 | | 委员可操作 |
| | 冻结解冻合约 | 合约生命周期 | | 委员可操作 |
| | 添加撤销运维 | | | 委员可操作 |
| | 用户表写权限 | | | 委员可操作 |
| 运维 | 部署合约 | _sys_tables_的写权限 | | 运维操作 |
| | 创建表 | _sys_tables_的写权限 | | 运维操作 |
| | 合约版本管理 | CNS | | 运维操作 |
| | 冻结解冻本账号部署的合约 | 修改合约状态 | | 运维操作 |
| 业务 | 调用合约 | 合约表写权限 | | 业务操作 |

# 投票模型

- 计票与生效规则
  - 链上所有治理委员都可以参与投票，只有当有效投票数/委员数>生效阈值，该治理操作才能生效，其委员票数（即权重）和生效阈值皆可通过投票修改；
  - 每次投票操作，会在系统表（_sys_committee_votes_）记录操作内容和投票委员，不重复计算；同时计算有效投票数/委员数，如大于阈值则该操作生效；
  - 投票设置过期时间，根据块高，blockLimit的10倍，固定不可改。

CONTENTS

章节

# 第四章 角色权限操作

# 委员账户权限操作

- 账户准备
  - 为了方便演示和理解FISCO BCOS中的角色权限系统，预先用控制台生成三个账户：
    - ✓账户1：0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
    - ✓账户2：0xca095e590dd772e600ddb139367fda845bf6b695
    - ✓账户3：0xb472f2bee5b7f18c46467fb80ed157d390829cc9
  - 然后打开三个终端，分别用三个账户登录控制台，示例如下：
    - ✓使用账户1登录控制台，其他终端操作类似。

```
$ bash console/start.sh
[group:1]> loadAccount 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
Load account 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 success!
```

# 委员账户权限操作

- 添加委员
  - 在账户1的控制台输入如下命令，添加账户1为委员：grantCommitteeMember 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
  - 初始系统无权限设置记录，任何账户均可使用权限设置，一旦设置权限后，只有委员才可授权委员账号。

```
[group:1]> grantCommitteeMember  0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
{
    "code":1,
    "msg":"Success"
}
```

添加委员成功

# 委员账户权限操作

- 此时，若在账户2所在的控制台尝试将账户2添加为委员，会发生权限错误。因为此时系统中已经存在委员账户1，只能由委员账户进行添加委员的操作。

```
[group:1]> grantCommitteeMember 0xca095e590dd772e600ddb139367fda845bf6b695
{
    "code":-52002,
    "msg":"Invalid request for permission deny"
}
```

权限拒绝

- 在账户1的控制台输入如下命令，可查看委员账户的列表。

```
[group:1]> listCommitteeMembers
-------------------------------------------------------------------------------------
|                address                |                enable_num                |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 账户1已成为委员账户   1                |
-------------------------------------------------------------------------------------
```

# 委员账户权限操作

- 另起新终端登录MySQL，使用root账户连接到db_node0数据库

```
$ mysql -uroot -p123456 -A db_node0
mysql: [Warning] Using a password on the command line interface can be insecure. Welcome to the MySQL monitor. Com

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of the

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

使用root账户连接到db_node0

- 查看权限表（_sys_table_access_）数据

```
mysql> select * from _sys_table_access_;
+--------+-------+----------+--------------------+------------------------------------------+------------+
| _id_   | _num_ | _status_ | table_name         | address                                  | enable_num |
+--------+-------+----------+--------------------+------------------------------------------+------------+
| 100152 |     1 |        0 | _sys_config_       | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
| 100153 |     1 |        0 | _sys_consensus_    | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
| 100154 |     1 |        0 | _sys_table_access_ | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
+--------+-------+----------+--------------------+------------------------------------------+------------+
```

账户1对这些表具有写入权限

# 委员账户权限操作

- 添加账户2为委员
  - 使用账户1的控制台进行投票，由于此时只有账户1是委员，生效阈值=有效票/总票数=1/1=1>0.5，所以立即生效。

```
[group:1]> grantCommitteeMember 0xca095e590dd772e600ddb139367fda845bf6b695
{
    "code":1,
    "msg":"Success"
}
```

  - 再次查看委员账户列表，此时系统中已经有两个委员账户（账户1和账户2）

```
[group:1]> listCommitteeMembers
-------------------------------------------------------------------------------------
|                address                |                enable_num                 |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 <-- 账户1    1                         |
| 0xca095e590dd772e600ddb139367fda845bf6b695 <-- 账户2    2                         |
-------------------------------------------------------------------------------------
```

# 委员账户权限操作

- 撤销委员
  - 在账户1所在的控制台输入如下命令，撤销委员账户2。

```
[group:1]> revokeCommitteeMember 0xca095e590dd772e600ddb139367fda845bf6b695
{
    "code":0,
    "msg":"Success"
}
[group:1]> listCommitteeMembers
-------------------------------------------------------------------------------------
|                   address                |                enable_num                 |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 |                    1                     |
| 0xca095e590dd772e600ddb139367fda845bf6b695 |                    2                     |
-------------------------------------------------------------------------------------
```

账户1投票后，账户2仍是委员

  - 此时系统中有两个委员（账户1和账户2），账户1账户投票后的生效阈值=有效票/总票数=1/2=0.5，而FISCO BCOS系统中默认生效阈值为0.5，所以仍需要账户2进行投票才能生效。

# 委员账户权限操作

- 在账户2所在的控制台输入如下命令，撤销委员账户2。

```
[group:1]> revokeCommitteeMember 0xca095e590dd772e600ddb139367fda845bf6b695
{
    "code":1,
    "msg":"Success"
}

[group:1]> listCommitteeMembers
-----------------------------------------------------------------------------------------
|                   address                |                enable_num                 |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 |                    1                      |
-----------------------------------------------------------------------------------------
```

账户2也投票后，只有账户1是委员

- 账户2账户投票后的生效阈值=有效票/总票数=2/2=1>0,5，所以账户2投票后立即生效，账户2撤销委员操作成功。

# 委员账户权限操作

- 此时再在MySQL中查看权限表数据变化情况。

```
mysql> select * from _sys_table_access_;
+--------+-------+----------+--------------------+------------------------------------------+------------+
| _id_   | _num_ | _status_ | table_name         | address                                  | enable_num |
+--------+-------+----------+--------------------+------------------------------------------+------------+
| 100152 |     1 |        0 | _sys_config_       | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
| 100153 |     1 |        0 | _sys_consensus_    | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
| 100154 |     1 |        0 | _sys_table_access_ | 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 1          |
| 100162 |     2 |        1 | _sys_config_       | 0xca095e590dd772e600ddb139367fda845bf6b695 | 2          |
| 100163 |     2 |        1 | _sys_consensus_    | 0xca095e590dd772e600ddb139367fda845bf6b695 | 2          |
| 100164 |     2 |        1 | _sys_table_access_ | 0xca095e590dd772e600ddb139367fda845bf6b695 | 2          |
+--------+-------+----------+--------------------+------------------------------------------+------------+
```

此状态标志账户不可用

- 撤销委员并不会删除权限表中的数据，而是以status字段来标注该账户是否可用，0表示该账户地址有效，1表示不可用。

# 委员账户权限操作

- 修改委员权重
  - 先在账户1所在的控制台中添加账户3为委员
  - 

```
[group:1]> grantCommitteeMember 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
{
    "code":1,
    "msg":"Success"
}
```

```
[group:1]> listCommitteeMembers
-------------------------------------------------------------------------------
|                address                |               enable_num               |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 |                 1                 |
| 0xb472f2bee5b7f18c46467fb80ed157d390829cc9 |                 6                 |
-------------------------------------------------------------------------------
```

账户3

# 委员账户权限操作

- 在账户1的控制台输入如下命令，更新账户1的权重为2。

```
[group:1]> updateCommitteeMemberWeight 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 2
{
    "code":0,
    "msg":"Success"
}
```

- 

```
[group:1]> queryCommitteeMemberWeight 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
Account: 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 Weight: 1
```

账户1权重为1

效阈值=有效票/总票数=1/2=0.5，而FISCO BCOS系统中默认生效阈值为0.5，所以仍需要账户3进行投票才能生效。

# 委员账户权限操作

- 在账户3的控制台输入如下命令，更新账户1的权重为2。

```
[group:1]> updateCommitteeMemberWeight 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 2
{
    "code":1,
    "msg":"Success"
}
[group:1]> queryCommitteeMemberWeight 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2
Account: 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 Weight: 2
```

账户3成功投票后，账户1的权重为2

- 账户3账户投票后的生效阈值=有效票/总票数=2/2=1>0,5，所以账户3投票后立即生效，账户1修改权重的操作成功，权重为2。

# 委员账户权限操作

- 在账户1的控制台输入如下命令，添加委员2为委员。

```
[group:1]> grantCommitteeMember 0xca095e590dd772e600ddb139367fda845bf6b695
{
    "code":1,
    "msg":"Success"
}

[group:1]> listCommitteeMembers
-------------------------------------------------------------------------------------------
|                   address                |                enable_num                |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 | 账户1            1                      |
| 0xb472f2bee5b7f18c46467fb80ed157d390829cc9 | 账户3            6                      |
| 0xca095e590dd772e600ddb139367fda845bf6b695 |                  9                      |
-------------------------------------------------------------------------------------------
                                              账户2被添加为委员
```

- 账户1投票之前系统中有两个委员（账户1和账户3），但由于账户1的权重为2，账户1投票后的有效阈值==有效票/总票数=2/3>0.5，立即生效，不需要账户3再进行投票。

# 委员账户权限操作

- 修改生效阈值
  - 在账户1的控制台输入如下命令，修改系统阈值为75（默认为50）

```
[group:1]> updateThreshold 75
{
    "code":0,
    "msg":"Success"
}
```

  - 阈值

```
# 查询生效阈值
[group:1]> queryThreshold
Effective threshold : 50%
```

  - ，账户2权重为1，账户3权重为1），生效阈值=有效票/总票数=2/4=0.5=默认有效阈值（0.5），还需要其他委员投票才能生效。

# 委员账户权限操作

- 在账户2的控制台输入如下命令，修改系统阈值为75（默认为50）

```
[group:1]> updateThreshold 75
{
    "code":0,
    "msg":"Success"
}
# 查询生效阈值
[group:1]> queryThreshold
Effective threshold : 75%
```

账户2投票后，系统有效阈值变为75%

- 数=3/4=0.75>0.5，修改系统有效阈值的提议通过。

# 委员账户权限操作

- 在账户1的控制台输入如下命令，撤销账户3的委员权限

```
revokeCommitteeMember 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
{
    "code":0,
    "msg":"Success"
}
[group:1]> listCommitteeMembers
---------------------------------------------------------------------------------------
|                   address                |                enable_num                 |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2 |                    1                     |
| 0xb472f2bee5b7f18c46467fb80ed157d390829cc9 |                    6                     |
| 0xca095e590dd772e600ddb139367fda845bf6b695 |                    9                     |
---------------------------------------------------------------------------------------
```

账户3仍是委员

- ，还需要其他委员投票才能生效。

# 委员账户权限操作

- 在账户2的控制台输入如下命令，撤销账户3的委员权限

```
revokeCommitteeMember 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
{
    "code":0,
    "msg":"Success"
}
[group:1]> listCommitteeMembers
--------------------------------------------------------------------------------------
|                   address                    |                enable_num                |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2   |                    1                     |
| 0xb472f2bee5b7f18c46467fb80ed157d390829cc9   |                    6                     |
| 0xca095e590dd772e600ddb139367fda845bf6b695   |                    9                     |
--------------------------------------------------------------------------------------
```

账户3仍是委员

- ，还需要账户3投票才能生效。

# 委员账户权限操作

- 在账户3的控制台输入如下命令，撤销账户3的委员权限

```
revokeCommitteeMember 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
{
    "code":0,
    "msg":"Success"
}
[group:1]> listCommitteeMembers
--------------------------------------------------------------------------------------
|                   address                    |              enable_num               |
| 0x10b3603fd4923760f5b8e1c0038fe21ef44fd8f2   |                 1                     |
| 0xca095e590dd772e600ddb139367fda845bf6b695   |                 9                     |
--------------------------------------------------------------------------------------
```

账户1

账户2

- 生效。

# 运维账户权限操作

- 新增运维
  - 在账户1的控制台输入如下命令，新增账户3为运维账户

```
[group:1]> grantOperator 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
{
    "code":1,
    "msg":"Success"
}
```

  - 在账户1的控制台输入如下命令，查看运维账户列表

```
# 查看运维列表
[group:1]> listOperators
-------------------------------------------------------------------------------------
|                   address                    |                enable_num                |
| 0xb472f2bee5b7f18c46467fb80ed157d390829cc9   |                13                       |
-------------------------------------------------------------------------------------
```

账户3

  - 运维账户的新增和撤销不需要投票操作，由要实现账户授权即可。

# 运维账户权限操作

- 部署合约
  - 在账户3的控制台输入如下命令，部署HelloWorld合约
  - 

```
[group:1]> deploy HelloWorld
transaction hash: 0x57d8701fe54f485f14f6a97aae01bc97350e99b125b570e256c7ba0d332bd216
contract address: 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5
currentAccount: 0xb472f2bee5b7f18c46467fb80ed157d390829cc9
```

合约地址

  - 

```
[group:1]> deploy HelloWorld
deploy contract for HelloWorld failed!
return message: Permission denied
return code:25
Return values:null
```

权限拒绝

合约。

# 运维账户权限操作

- 冻结合约
  - 在账户3的控制台输入如下命令，冻结刚部署的HelloWorld合约

```
[group:1]> freezeContract 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5
{
    "code":1,
    "msg":"Success"
}
```

  - 在账户3的控制台输入如下命令，尝试调用HelloWorld合约

```
[group:1]> call HelloWorld 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5 get
call for HelloWorld failed, contractAddress is 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5
{
    "code":30,
    "msg":"Frozen contract:0db89a82c08f5f74ee423e47c084c4b862aadbb5"
}
description: Frozen contract:0db89a82c08f5f74ee423e47c084c4b862aadbb5, please refer to https://fisco-bcos-document
```

调用失败，合约被冻结

# 运维账户权限操作

- 解冻合约
  - 在账户3的控制台输入如下命令，解冻刚部署的HelloWorld合约
  -

```
[group:1]> unfreezeContract 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5
{
    "code":1,
    "msg":"Success"
}
```

```
[group:1]> call HelloWorld 0x0db89a82c08f5f74ee423e47c084c4b862aadbb5 get
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return value size:1
Return types: (STRING)
Return values:(Hello, World!)
-------------------------------------------------------------------------------------------
```

合约返回值

# 运维账户权限操作

- 撤销运维
  - 在账户1的控制台输入如下命令，撤销账户C的运维权限

```
[group:1]> revokeOperator 0xf4b5fe6225246bd99caaefe0038ad2783a4d8f49
{
    "code":1,
    "msg":"Success"
}
```

```
# 查看运维列表
[group:1]> listOperators
Empty set.
```

# 作业

1. 哪些情况下需要使用外部账户？
2. 外部账户和合约账户有什么不同？
3. 如何指定账户进行交易？
4. 什么是角色模型？这样设计有什么好处？
5. 治理委员账号有什么权限？可以部署合约吗？
6. 在权限管理系统中，有三个委员，分别是A、B和C，权重分别为1，2，1，阈值为0.6。现在想增加一个委员D，那么投票在哪些情况下会成功？
7. 下面哪些操作需要委员投票决定？

A 添加运维账户 B 增加共识节点 C 修改生效阈值 D 冻结账号合约

# 课程总结

- 本课程详细介绍了
  - FISCO BCSO账户的生成和使用
  - FISCO BCOS账户的计算过程
  - FISCO BCOS权限控制介绍
  - FISCO BCOS角色与权限模型
  - FISCO BCOS不同角色的权限操作



腾讯云

腾讯云

# 谢谢