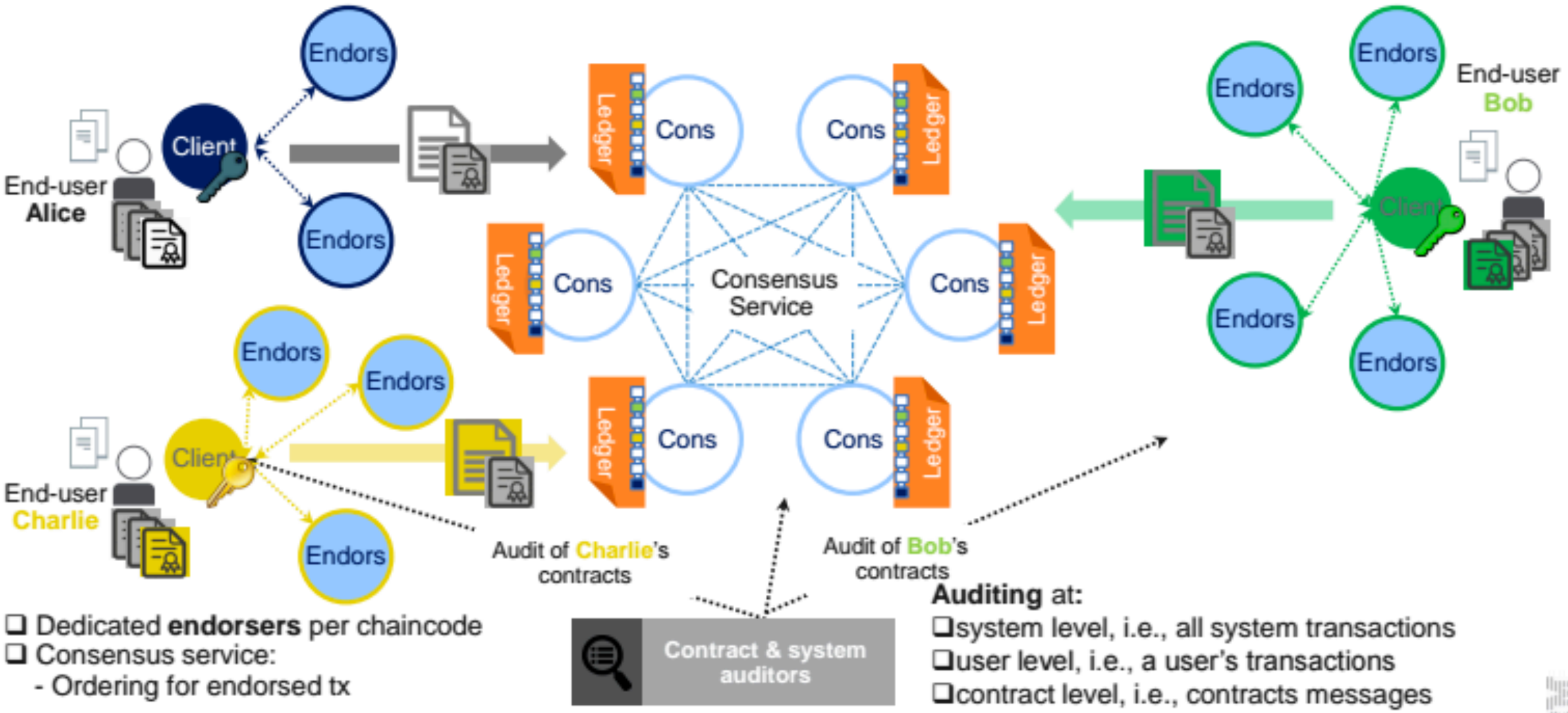
Fabric v1.0.1 區塊鏈部署實務

盧瑞山 教授

Fabric network (v1.0.0 alpha)

Separating transaction endorsement from consensus

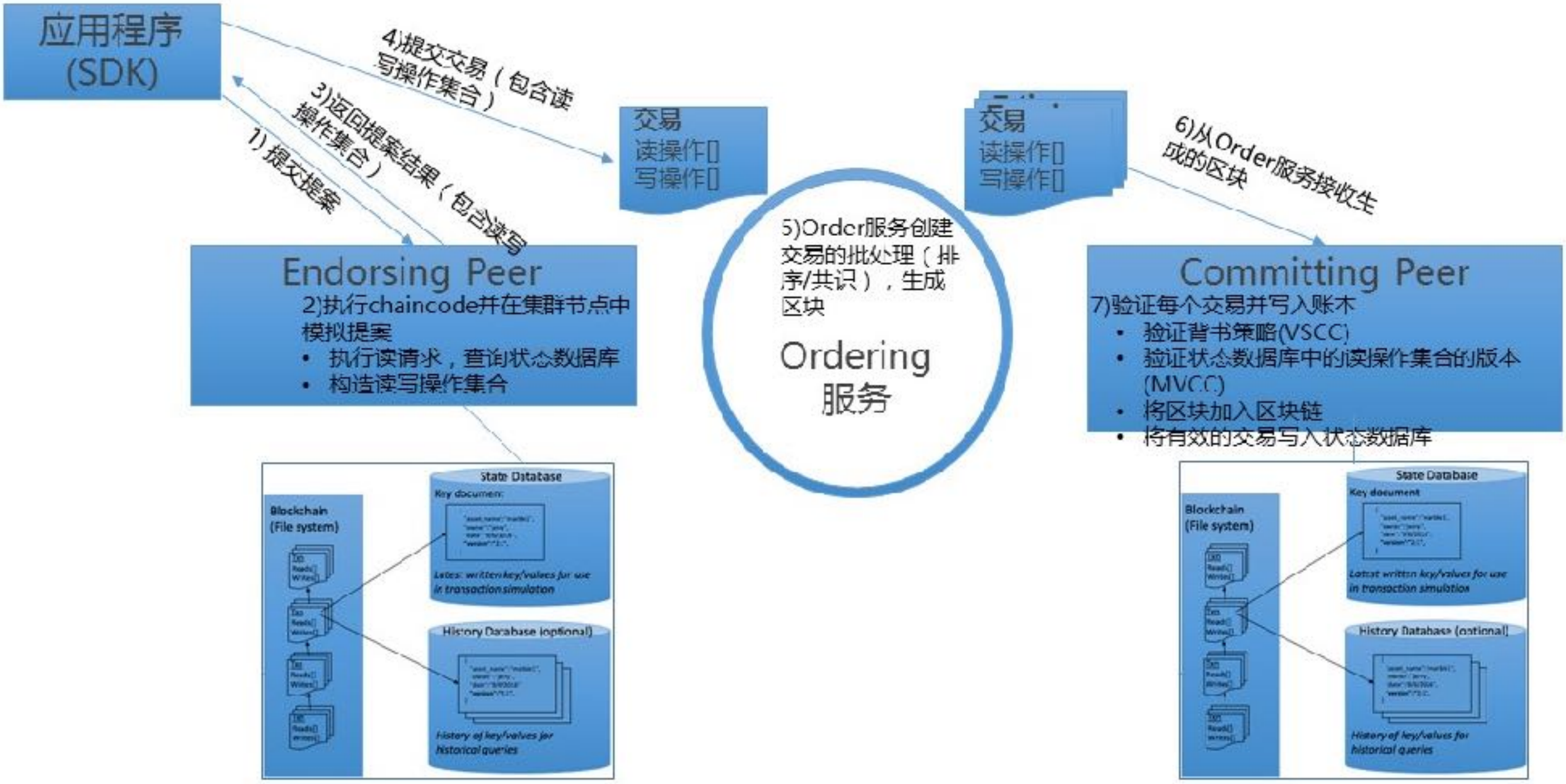


- ❑ Dedicated **endorsers** per chaincode
- ❑ Consensus service:
  - Ordering for endorsed tx

Auditing at:

- ❑system level, i.e., all system transactions
- ❑user level, i.e., a user's transactions
- ❑contract level, i.e., contracts messages

Fabric 1.0 交易的完整生命周期（流程图）



刪除不必要的容器與映像檔

- docker rm -f $(docker ps -aq)

- docker rmi -f $(docker images)

Binary code download

- curl -sSL https://goo.gl/6wtTN5 | bash -s 1.1.0-preview

```
rslu@hyperledger-fabrice-0-6-4peer:~/bin$ ls
configtxgen  configtxlator  cryptogen  get-byfn.sh  get-docker-images.sh  orderer  peer
rslu@hyperledger-fabrice-0-6-4peer:~/bin$ ls -l
total 81540
-rwxrwxr-x 1 rslu rslu 15247600 Aug 10 10:43 configtxgen
-rwxrwxr-x 1 rslu rslu 16710031 Aug 10 10:43 configtxlator
-rwxrwxr-x 1 rslu rslu  7657096 Aug 10 10:43 cryptogen
-rwxrwxr-x 1 rslu rslu      441 Aug 10 10:43 get-byfn.sh
-rwxrwxr-x 1 rslu rslu      757 Aug 10 10:43 get-docker-images.sh
-rwxrwxr-x 1 rslu rslu 20450203 Aug 10 10:43 orderer
-rwxrwxr-x 1 rslu rslu 23412781 Aug 10 10:43 peer
rslu@hyperledger-fabrice-0-6-4peer:~/bin$
```

Docker images

```
===> List out hyperledger docker images
hyperledger/fabric-ca           latest              5f30bda5f7ee        7 days ago          238MB
hyperledger/fabric-ca           x86_64-1.0.1        5f30bda5f7ee        7 days ago          238MB
hyperledger/fabric-tools        latest              259847d24868        7 days ago          1.34GB
hyperledger/fabric-tools        x86_64-1.0.1        259847d24868        7 days ago          1.34GB
hyperledger/fabric-couchdb      latest              dd645e1e92c7        7 days ago          1.48GB
hyperledger/fabric-couchdb      x86_64-1.0.1        dd645e1e92c7        7 days ago          1.48GB
hyperledger/fabric-kafka        latest              cbdc916590a0        7 days ago          1.3GB
hyperledger/fabric-kafka        x86_64-1.0.1        cbdc916590a0        7 days ago          1.3GB
hyperledger/fabric-zookeeper    latest              eb07e5cc9674        7 days ago          1.31GB
hyperledger/fabric-zookeeper    x86_64-1.0.1        eb07e5cc9674        7 days ago          1.31GB
hyperledger/fabric-orderer      latest              bbf2708c9487        7 days ago          179MB
hyperledger/fabric-orderer      x86_64-1.0.1        bbf2708c9487        7 days ago          179MB
hyperledger/fabric-peer         latest              abb05def5cfb        7 days ago          182MB
hyperledger/fabric-peer         x86_64-1.0.1        abb05def5cfb        7 days ago          182MB
hyperledger/fabric-javaenv      latest              2bd60859415d        7 days ago          1.42GB
hyperledger/fabric-javaenv      x86_64-1.0.1        2bd60859415d        7 days ago          1.42GB
hyperledger/fabric-ccenv        latest              7e2019cf8174        7 days ago          1.29GB
hyperledger/fabric-ccenv        x86_64-1.0.1        7e2019cf8174        7 days ago          1.29GB
```

```
cd bin
git clone https://github.com/hyperledger/fabric-samples.git
cd fabric-samples
cd first-network
```

./byfn.sh -m generate

```
rslu@hyperledger-fabrice-0-6-4peer:~/fabric-samples/first-network$ ./byfn.sh -m generate
Generating certs and genesis block for with channel 'mychannel' and CLI timeout of '10000'
Continue (y/n)? y
proceeding ...
/home/rslu/bin/cryptogen

###############################################################
##### Generate certificates using cryptogen tool #########
###############################################################
org1.example.com
org2.example.com

/home/rslu/bin/configtxgen
###############################################################
#########  Generating Orderer Genesis block ##############
###############################################################
2017-08-17 14:31:05.900 UTC [common/configtx/tool] main -> INFO 001 Loading configuration
2017-08-17 14:31:05.923 UTC [common/configtx/tool] doOutputBlock -> INFO 002 Generating genesis block
2017-08-17 14:31:05.924 UTC [common/configtx/tool] doOutputBlock -> INFO 003 Writing genesis block

#################################################################
### Generating channel configuration transaction 'channel.tx' ###
#################################################################
2017-08-17 14:31:05.939 UTC [common/configtx/tool] main -> INFO 001 Loading configuration
2017-08-17 14:31:05.942 UTC [common/configtx/tool] doOutputChannelCreateTx -> INFO 002 Generating new channel configtx
2017-08-17 14:31:05.942 UTC [common/configtx/tool] doOutputChannelCreateTx -> INFO 003 Writing new channel tx

#################################################################
#######    Generating anchor peer update for Org1MSP   ##########
#################################################################
2017-08-17 14:31:05.956 UTC [common/configtx/tool] main -> INFO 001 Loading configuration
2017-08-17 14:31:05.959 UTC [common/configtx/tool] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor peer update
2017-08-17 14:31:05.959 UTC [common/configtx/tool] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer update

#################################################################
#######    Generating anchor peer update for Org2MSP   ##########
#################################################################
2017-08-17 14:31:05.974 UTC [common/configtx/tool] main -> INFO 001 Loading configuration
2017-08-17 14:31:05.977 UTC [common/configtx/tool] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor peer update
2017-08-17 14:31:05.977 UTC [common/configtx/tool] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer update
```

./byfn.sh -m up

```
rslu@hyperledger-fabrice-0-6-4peer:~/fabric-samples/first-network$ ./byfn.sh -m up
Starting with channel 'mychannel' and CLI timeout of '10000'
Continue (y/n)? y
proceeding ...
Creating peer0.org2.example.com
Creating peer1.org2.example.com
Creating peer1.org1.example.com
Creating orderer.example.com
Creating peer0.org1.example.com
Creating cli

 _____   _   _   ____
| ____| | \ | | |  _ \
|  _|   |  \| | | | | |
| |___  | |\  | | |_| |
|_____| |_| \_| |____/

Build your first network (BYFN) end-to-end test

Channel name : mychannel
Creating channel...
CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
CORE_PEER_TLS_KEY_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/server.key
CORE_PEER_LOCALMSPID=Org1MSP
CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
CORE_PEER_TLS_CERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/server.crt
CORE_PEER_TLS_ENABLED=true
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
CORE_PEER_ID=cli
CORE_LOGGING_LEVEL=DEBUG
CORE_PEER_ADDRESS=peer0.org1.example.com:7051
2017-08-17 14:56:41.034 UTC [msp] GetLocalMSP -> DEBU 001 Returning existing local MSP
2017-08-17 14:56:41.034 UTC [msp] GetDefaultSigningIdentity -> DEBU 002 Obtaining default signing identity
2017-08-17 14:56:41.037 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2017-08-17 14:56:41.037 UTC [msp] GetLocalMSP -> DEBU 004 Returning existing local MSP
2017-08-17 14:56:41.037 UTC [msp] GetDefaultSigningIdentity -> DEBU 005 Obtaining default signing identity
2017-08-17 14:56:41.037 UTC [msp] GetLocalMSP -> DEBU 006 Returning existing local MSP
2017-08-17 14:56:41.037 UTC [msp] GetDefaultSigningIdentity -> DEBU 007 Obtaining default signing identity
2017-08-17 14:56:41.037 UTC [msp/identity] Sign -> DEBU 008 Sign: plaintext: 0ABC060A074F7267314D53501280D62D...53616D706C65436F6E736F727469756D
```

./byfn.sh -m up

```
===================== Chaincode is installed on remote peer PEER3 =====================

Querying chaincode on org2/peer3...
===================== Querying on PEER3 on channel 'mychannel'... =====================
CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
CORE_PEER_TLS_KEY_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/server.key
CORE_PEER_LOCALMSPID=Org2MSP
CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
CORE_PEER_TLS_CERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/server.crt
CORE_PEER_TLS_ENABLED=true
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
CORE_PEER_ID=cli
CORE_LOGGING_LEVEL=DEBUG
CORE_PEER_ADDRESS=peer1.org2.example.com:7051
Attempting to Query PEER3 ...3 secs
2017-08-17 14:57:17.565 UTC [msp] GetLocalMSP -> DEBU 001 Returning existing local MSP
2017-08-17 14:57:17.565 UTC [msp] GetDefaultSigningIdentity -> DEBU 002 Obtaining default signing identity
2017-08-17 14:57:17.565 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 003 Using default escc
2017-08-17 14:57:17.565 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 004 Using default vscc
2017-08-17 14:57:17.565 UTC [msp/identity] Sign -> DEBU 005 Sign: plaintext: 0A95070A6708031A0C08CDDDD6CC0510...6D7963631A0A0A0571756572790A0161
2017-08-17 14:57:17.565 UTC [msp/identity] Sign -> DEBU 006 Sign: digest: 4215E31A1C53BA09848D3814177B92F51DDEDCA20CA270C19A5C0D143CC891C6
Query Result: 90
2017-08-17 14:57:26.386 UTC [main] main -> INFO 007 Exiting.....
===================== Query on PEER3 on channel 'mychannel' is successful =====================

======== All GOOD, BYFN execution completed ===========


 _____   _   _   ____
| ____| | \ | | |  _ \
|  _|   |  \| | | | | |
| |___  | |\  | | |_| |
|_____| |_| \_| |____/
```

docker ps

```
rslu@hyperledger-fabrice-0-6-4peer:~$ docker ps
CONTAINER ID        IMAGE                                                                                                  COMMAND                  CREATED
    STATUS              PORTS                                              NAMES
cffd0ca02a9b        dev-peer1.org2.example.com-mycc-1.0-26c2ef32838554aac4f7ad6f100aca865e87959c9a126e86d764c8d01f8346ab   "chaincode -peer.a..."   4 minutes ago
    Up 4 minutes                                                           dev-peer1.org2.example.com-mycc-1.0
cdaafda1d6d9        dev-peer0.org1.example.com-mycc-1.0-384f11f484b9302df90b453200cfb25174305fce8f53f4e94d45ee3b6cab0ce9   "chaincode -peer.a..."   4 minutes ago
    Up 4 minutes                                                           dev-peer0.org1.example.com-mycc-1.0
eea01c5c1149        dev-peer0.org2.example.com-mycc-1.0-15b571b3ce849066b7ec74497da3b27e54e0df1345daff3951b94245ce09c42b   "chaincode -peer.a..."   5 minutes ago
    Up 5 minutes                                                           dev-peer0.org2.example.com-mycc-1.0
406b62e832e5        hyperledger/fabric-tools                                                                               "/bin/bash -c './s..."   5 minutes ago
    Up 5 minutes                                                           cli
a8870c14d2d8        hyperledger/fabric-peer                                                                                "peer node start"        5 minutes ago
    Up 5 minutes        0.0.0.0:8051->7051/tcp, 0.0.0.0:8053->7053/tcp     peer1.org1.example.com
39e6f6c1aa01        hyperledger/fabric-peer                                                                                "peer node start"        5 minutes ago
    Up 5 minutes        0.0.0.0:7051->7051/tcp, 0.0.0.0:7053->7053/tcp     peer0.org1.example.com
0f7a17fc1e59        hyperledger/fabric-peer                                                                                "peer node start"        5 minutes ago
    Up 5 minutes        0.0.0.0:10051->7051/tcp, 0.0.0.0:10053->7053/tcp   peer1.org2.example.com
a2809aff3724        hyperledger/fabric-orderer                                                                             "orderer"                5 minutes ago
    Up 5 minutes        0.0.0.0:7050->7050/tcp                             orderer.example.com
baa493881a88        hyperledger/fabric-peer                                                                                "peer node start"        5 minutes ago
    Up 5 minutes        0.0.0.0:9051->7051/tcp, 0.0.0.0:9053->7053/tcp     peer0.org2.example.com
```

./byfn.sh -m down

```
rslu@hyperledger-fabrice-0-6-4peer:~/fabric-samples/first-network$ ./byfn.sh -m down
Stopping with channel 'mychannel' and CLI timeout of '10000'
Continue (y/n)? y
proceeding ...
WARNING: The CHANNEL_NAME variable is not set. Defaulting to a blank string.
WARNING: The TIMEOUT variable is not set. Defaulting to a blank string.
Stopping cli ... done
Stopping peer1.org1.example.com ... done
Stopping peer0.org1.example.com ... done
Stopping peer1.org2.example.com ... done
Stopping orderer.example.com ... done
Stopping peer0.org2.example.com ... done
Removing cli ... done
Removing peer1.org1.example.com ... done
Removing peer0.org1.example.com ... done
Removing peer1.org2.example.com ... done
Removing orderer.example.com ... done
Removing peer0.org2.example.com ... done
Removing network net_byfn
cffd0ca02a9b
cdaafda1d6d9
eea01c5c1149
Untagged: dev-peer1.org2.example.com-mycc-1.0-26c2ef32838554aac4f7ad6f100aca865e87959c9a126e86d764c8d01f8346ab:latest
Deleted: sha256:9227825d684159e847a1ee0bb83e4194a426c7aea83b4fe3c9406849bb89036e
Deleted: sha256:d5c049349f2e32e13b05f6e660143dd3c8bc60427d009e0f4b610fb54d2696ab
Deleted: sha256:9625d5893114c5717da4c3bd0a8a0716e1c7f02aa1fe959b54208a885da800cb
Deleted: sha256:28a687681e5d10915995cd98ed7a3731d5408986439b3c288c7aabc99585ddd9
Deleted: sha256:d245d67e9c1980f61ffa9d6ecd96c6cbaff7ad1fe89103dba726c9a820f1f346
Deleted: sha256:030d632aec1ea74cfd41888bd015b4b6c139d82a67d6fd9aac921fb38a00f447
Deleted: sha256:75023970ab2a7215dbbc13e7b4ca984b248390a1832f05cb6a305ed5c1f0dd45
Untagged: dev-peer0.org1.example.com-mycc-1.0-384f11f484b9302df90b453200cfb25174305fce8f53f4e94d45ee3b6cab0ce9:latest
Deleted: sha256:8b9763c55022bd55f4387079f0a9141495eead5779b697f07b85fb3ccaa883e5
Deleted: sha256:5f8fd8084a8ced73620d4219f51eddc1bec055ee19f17cd5306331f06500806e
Deleted: sha256:74dc1b9798a55b862ee8477e51aad3650349f9439a76dead829215c079219bed
Deleted: sha256:e030e9ebf81bc5cb2a3043aefe15914824ef3869a8f7293eb61d6173a0e79b85
Deleted: sha256:24e229c9d610787fc9f84eb8791c381dc0849c4582451a267c7e24a7ba63aadb
Deleted: sha256:9cb8863ef81a6ba602479526c63c642db10261a56ff89ae383f48f818e5b4722
Deleted: sha256:bd4d8bea72ce563ebbc612333980f4f1684c05780acff55b080ca16c138cf719
Untagged: dev-peer0.org2.example.com-mycc-1.0-15b571b3ce849066b7ec74497da3b27e54e0df1345daff3951b94245ce09c42b:latest
Deleted: sha256:4da0b1563726dcf6c5ad16876f3a43b724237173639fad75bb32b420791cb2f4
Deleted: sha256:f2b6886d2c40c00349481f34525924d854b83317a3536390b0ad7516f60449e5
Deleted: sha256:3a16796d8235306e20d37389791e3ffd5cd264a814f1e8fb7fe33506e96d7d66
Deleted: sha256:b3700f75a9374a5827834bf9d3f77f66dc8bb79e7aa2cbf67a0e6da5d8f216b6
Deleted: sha256:d052bd41257d84aff87b2648d4314e8da8747b5f5da4485c65141296a6465a0a
Deleted: sha256:2b00154b2732cb463023c1ac18d94a39d58ca58677aa92d4012ed91b3387b2d3
Deleted: sha256:3245ce6d2b7f677d0a677a5e738160a867abefcd2a5fd772840b57c255881a6f
```

進入節點發出交易請求

- docker exec -it peer0.org1.example.com bash
- docker exec -it peer1.org2.example.com bash
- peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'
- 進入cli ===>docker exec -it cli bash 然後執行下行指令：會成功
- 若進入peer然後執行下行指令：會失敗
- peer chaincode invoke -o orderer.example.com:7050 --tls $CORE_PEER_TLS_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n mycc -c '{"Args":["invoke","a","b","10"]}'

結論

- 1. 不涉及入塊的行為，可以直接與peer互動。 從query可得到回傳結果，得到應證。
- 2. 正式的交易必須由cli負責發起，再傳給peer.
- 3. 這個byfn 這個範例，跑起來的4個節點，都是背書節點

Cli

- CLI will be used for the creation and joining of the channel and the node SDK will be used for the client authentication, and chaincode functions utilizing the channel.

chaincode

注意：你必須在將要執行你的chaincode的通道上的所有背書peer節點安裝chaincode

hyperledger/fabric提供了四個命令來管理chaincode的生命週期：package，install，instantiate和upgrade

Chain code Example

Chain Code development

- $GOPATH/src/
- mkdir -p $GOPATH/src/sacc && cd $GOPATH/src/sacc
- touch sacc.go

Housekeeping

- As with every chaincode, it implements the Chaincode interface in particular, Init and Invoke functions. So, let’s add the go import statements for the necessary dependencies for our chaincode. We’ll import the chaincode shim package and thepeer protobuf package. Next, let’s add a struct SimpleAsset as a receiver for Chaincode shim functions.

Package main

- package main

- import (
- "fmt"
- "github.com/hyperledger/fabric/core/chaincode/shim"
- "github.com/hyperledger/fabric/protos/peer"
-)

- // SimpleAsset implements a simple chaincode to manage an asset
- type SimpleAsset struct {
- }

Initializing the Chaincode

- // Init is called during chaincode instantiation to initialize any data.
- **func (t *SimpleAsset) Init**(stub shim.ChaincodeStubInterface) peer.Response {
- }

Call ChaincodeStubInterface.GetStringArgs

- // Init is called during chaincode instantiation to initialize any data. Note that chaincode upgrade also calls this function to reset or to migrate data, so be careful to avoid a scenario where you inadvertently clobber your ledger's data!

- func (t *SimpleAsset) **Init**(stub shim.ChaincodeStubInterface) peer.Response {

- // Get the args from the transaction proposal

- **args := stub.GetStringArgs()**

- **if len(args) != 2 {**

- **return shim.Error("Incorrect arguments. Expecting a key and a value")**

- **}**

- }

call ChaincodeStubInterface.PutState

- // Init is called during chaincode instantiation to initialize any data. Note that chaincode upgrade also calls this function to reset or to migrate data, so be careful to avoid a scenario where you inadvertently clobber your ledger's data!
- func (t *SimpleAsset) **Init**(stub shim.ChaincodeStubInterface) peer.Response {
- // Get the args from the transaction proposal
- args := stub.GetStringArgs()
- if len(args) != 2 {
- return shim.Error("Incorrect arguments. Expecting a key and a value")
- }
- // Set up any variables or assets here by calling stub.PutState() // We store the key and the value on the ledger
- **err := stub.PutState(args[0], []byte(args[1]))**
- **if err != nil {**
- **return shim.Error(fmt.Sprintf("Failed to create asset: %s", args[0]))**
- **}**
- **return shim.Success(nil)**
- **}**

Invoking the Chaincode

- // Invoke is called per transaction on the chaincode. Each transaction is
- // either a 'get' or a 'set' on the asset created by Init function. The Set
- // method may create a new asset by specifying a new key-value pair.
- func (t *SimpleAsset) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
- // Extract the function and args from the transaction proposal
- fn, args := stub.GetFunctionAndParameters()
- }

Invoking the Chaincode

- **func (t *SimpleAsset) Invoke(stub shim.ChaincodeStubInterface) peer.Response {**
- // Extract the function and args from the transaction proposal
- fn, args := stub.GetFunctionAndParameters()
- var result string
- var err error
- if fn == "set" {
- **result, err = set(stub, args)**
- } else {
- **result, err = get(stub, args)**
- }
- if err != nil {
- return shim.Error(err.Error())
- }
- // Return the result as success payload
- return shim.Success([]byte(result))
- }

Implementing the Chaincode Application

```
// Set stores the asset (both key and value) on the ledger. If the key exists, it will override the value with the new one
func set(stub shim.ChaincodeStubInterface, args []string) (string, error) {
    if len(args) != 2 {
        return "", fmt.Errorf("Incorrect arguments. Expecting a key and a value")
    }
    err := stub.PutState(args[0], []byte(args[1]))
    if err != nil {
        return "", fmt.Errorf("Failed to set asset: %s", args[0])
    }
    return args[1], nil
}
```

```
// Get returns the value of the specified asset key
func get(stub shim.ChaincodeStubInterface, args []string) (string, error) {
    if len(args) != 1 {
        return "", fmt.Errorf("Incorrect arguments. Expecting a key")
    }
    value, err := stub.GetState(args[0])
    if err != nil {
        return "", fmt.Errorf("Failed to get asset: %s with error: %s", args[0], err)
    }
    if value == nil {
        return "", fmt.Errorf("Asset not found: %s", args[0])
    }
```

Pulling it All Together

- package main
- import (……)
- // SimpleAsset implements a simple chaincode to manage an asset
- type SimpleAsset struct { }
- func (t *SimpleAsset) Init(stub shim.ChaincodeStubInterface) peer.Response {……}
- func (t *SimpleAsset) Invoke(stub shim.ChaincodeStubInterface) peer.Response {…… }
- func set(stub shim.ChaincodeStubInterface, args []string) (string, error) {……}
- func get(stub shim.ChaincodeStubInterface, args []string) (string, error) {……}
- // main function starts up the chaincode in the container during instantiate
- func main() {
- 	if err := shim.Start(new(SimpleAsset)); err != nil {
- 		fmt.Printf("Error starting SimpleAsset chaincode: %s", err)
- 	}
- }

Building Chaincode

- Now let’s compile your chaincode.

- go get -u --tags nopkcs11 github.com/hyperledger/fabric/core/chaincode/shim

- go build --tags nopkcs11

Blockchain explorer安装

安裝步驟

- git clone https://github.com/hyperledger/blockchain-explorer
- Sudo apt-get install mysql-server
- 設定密碼123456
- Cd blockchain-explorer
- Npm install
- sudo systemctl start mysql
- mysql -u root -p < db/fabricexplorer.sql 匯入資料庫
- cd fabric-docker-compose-svt/
- ./download_images.sh
- ./start.sh
- Cd ~/blockchain-explorer
- ./start.sh
- 注意報錯的可能原因 之前8080被佔用、刪除 /tmp/fabric-client-kvs_peerOrg1/ 這個憑證目錄

mysql的表目錄

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| fabricexplorer     |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

```
mysql> show tables ;
+----------------------------+
| Tables_in_fabricexplorer   |
+----------------------------+
| blocks                     |
| chaincodes                 |
| channel                    |
| peer                       |
| peer_ref_channel           |
| transaction                |
| write_lock                 |
+----------------------------+
7 rows in set (0.00 sec)
```

```
mysql> select *  from blocks;
+----+---------+----------------------------------------------------------------+------------------------
----------------------------------+------------+---------+
| id | blocknum | datahash                                                       | prehash
                                  | channelname | txcount |
+----+---------+----------------------------------------------------------------+------------------------
----------------------------------+------------+---------+
|  6 |        0 | c6c1adcfc42f508200f30327154910bd5b6396050549412a416e3d8b46bf53e3 |
                                  | mychannel   |       1 |
|  7 |        1 | 3e861797e59b467486a54aeec5bbe2ebf4f3422457b43e32ba73219c0b1981ce | e3c92d457cba54e9744feb35
5e030366dbba6868442c7a95fc0a5dd134ccb68f | mychannel   |       1 |
|  8 |        2 | ea2af6f653f37236c34682b9d6d52617501ec874b34d2737b7df0980dcd8c47b | 5d66d9aaa3757567f06df2a
927206ba86f0ca8244ce199584f22cccf1533914 | mychannel   |       1 |
|  9 |        3 | 4a93815641043aa8e1fe1e53309363dec5e8d4495a7cc6fa4259e46edf866ab5 | 76b1530d8a7753374a3ac304
46fc5d5f2dc4886901e280f13fcc432fa3b0186d | mychannel   |       1 |
| 10 |        4 | 859f84448da00a46c31dd29e4e4a5ffec6ac002a8a1d60ed1a68b576020387ed | 1b00daf1c27bdbc0f359fb34
fac31c6d18565ba9c0fd5fcd7439c19b12ff4383 | mychannel   |       1 |
+----+---------+----------------------------------------------------------------+------------------------
----------------------------------+------------+---------+
5 rows in set (0.00 sec)
```

```
mysql> select *  from chaincodes ;
+----+------+---------+-----------------------------------------------------------------+-----------
-+----------+
| id | name | version | path                                                            | channelname
 | txcount |
+----+------+---------+-----------------------------------------------------------------+-----------
-+----------+
| 10 | mycc | 1.0     | github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02 | mychannel
 |       2 |
+----+------+---------+-----------------------------------------------------------------+-----------
-+----------+
1 row in set (0.00 sec)
```

```
mysql> select *  from transaction ;
+----+-------------+---------+------------------------------------------------------------------+-----------
---------+---------------+
| id | channelname | blockid | txhash                                                           | createdt
         | chaincodename |
+----+-------------+---------+------------------------------------------------------------------+-----------
---------+---------------+
|  6 | mychannel   |       0 |                                                                  | 2017-10-16
13:11:26 |               |
|  7 | mychannel   |       1 |                                                                  | 2017-10-16
13:11:48 |               |
|  8 | mychannel   |       2 |                                                                  | 2017-10-16
13:11:48 |               |
|  9 | mychannel   |       3 | 702f397a5657497ca0ca0158dd00def1f0bf576aec07faa32ee2ce8f263ace2b | 2017-10-16
13:11:48 | mycc          |
| 10 | mychannel   |       4 | 5a6ee623871ebf50dd355cc18ebb2fdd2cd568a2f5fae6b03d416b43135bcdb4 | 2017-10-16
13:12:31 | mycc          |
+----+-------------+---------+------------------------------------------------------------------+-----------
---------+---------------+
5 rows in set (0.00 sec)
```

104.199.153.103:8080

- FABRIC EXPLORER

Select Channel

CHANNEL

API

HELP

mychannel

PEER 2

BLOCK 4

TX 5

CHAINCODE 1

BLOCK #4

number	4
previous_hash	1b00daf1c27bdbc0f359fb34fac31c6d18565ba9e...
data_hash	859f84448da00a46c31dd29e4e4a5ffec6ac002a8...
Transactions	5a6ee623871ebf50dd355cc18ebb2fdd2cd568a2f...

BLOCKLIST

Block	TXNs
#4	1
#3	1
#2	1
#1	1
#0	1

BLOCKVIEW

Identifier [number, hash, tag]

Block

Transaction

Find

NO TRANSACTION

PEERLIST

name	org	mspid	request
peer1	peerOrg1	Org1MSP	grpc://112.124.115.82:7051