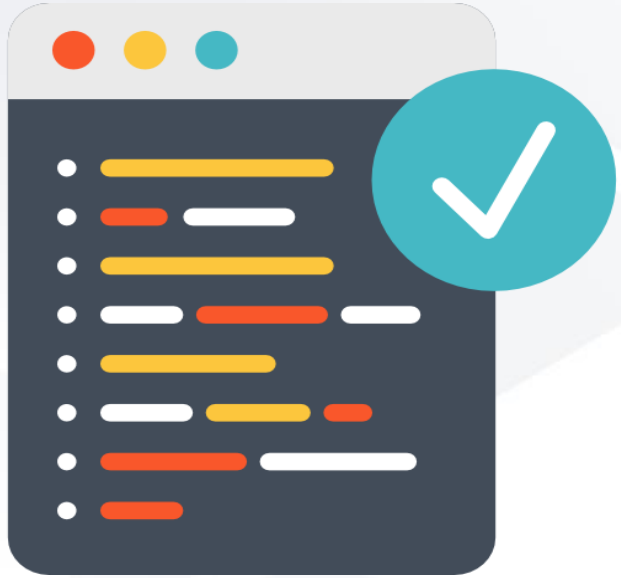
MÓDULO VI

Super Lab

BLOCKCHAIN
DEVELOPER
HYPERLEDGER

Fluxo do Lab

- Pré-requisitos
- Instalação
- Ativação da rede
- Criação de 1 canal
- Criação de vários canais
- Join do canal
- Deploy do chaincode
- Invocação de transações
- Entendimento do cenário



BLOCKCHAIN DEVELOPER HYPERLEDGER

Pre-reqs

- Editor de texto
- Ideal conhecer um pouco de Java ou programação
- Conhecimento básico de Blockchain, bases de dados, assinaturas e hashes

Ferramentas

- Ubuntu 16 ou acima ou MacOs 10.12
- curl
- git
- Docker versão acia de 17
- Docker compose acima de 1.29
- golang 1.17.x
- node verão 8.9 ou acima
- npm versão 5.x ou acima
- python 2.7x

BLOCKCHAIN DEVELOPER HYPERLEDGER

Instalações

Curl e Golang

- sudo apt-get install curl
- sudo apt-get install golang
- export GOPATH=$HOME/go
- export PATH=$PATH:$GOPATH/bin

Instalação de node, npm e python

- sudo apt-get install nodejs
- sudo apt-get install npm
- sudo apt-get install python

Instalações

Docker e Docker-compose

- sudo apt-get remove docker docker-engine docker.io containerd runc
- sudo apt-get update
- sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
- sudo apt-get update
- sudo apt-get install docker-ce docker-ce-cli containerd.io
- sudo docker run hello-world
- sudo apt-get install docker-compose

Verificações

Docker e Docker-compose

- Node –v
- Npm –v
- Go version
- Docker –version
- Docker-compose –version
- Python –v

Instalação de Blockchain

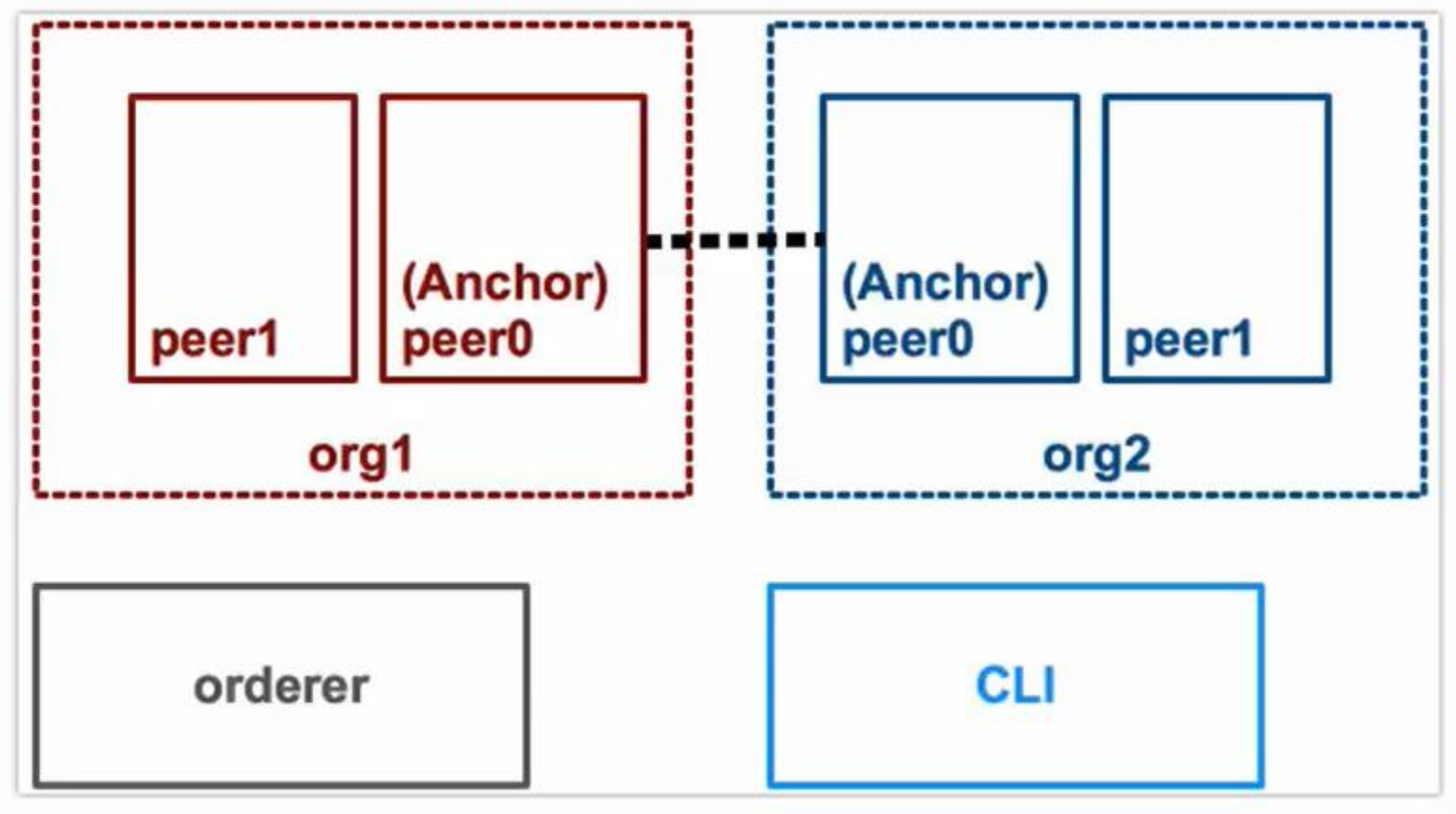
- Clonar https://github.com/hyperledger/fabric-samples
- mkdir -p $HOME/go/src/github.com/<your_github_userid>
- cd $HOME/go/src/github.com/<your_github_userid>
- curl -sSL https://bit.ly/2ysbOFE | bash -s

Execução da rede

- cd fabric-samples/test-network
- ./network.sh –h
- ./network.sh down
- ./network.sh up

BLOCKCHAIN DEVELOPER HYPERLEDGER

Estrutura da Rede



- Anchor comunica entre organizações
- Orderer gera o consenso
- CLI permite enviar comandos para a rede

BLOCKCHAIN DEVELOPER HYPERLEDGER

Output

```
Using docker and docker-compose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=2.4.2
DOCKER_IMAGE_VERSION=2.4.2
/root/go/src/github.com/bernardo9999/fabric-samples/test-network/../bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
```

Output

```
Generating CCP files for Org1 and Org2
Creating network "fabric_test" with the default driver
Creating volume "compose_orderer.example.com" with default driver
Creating volume "compose_peer0.org1.example.com" with default driver
Creating volume "compose_peer0.org2.example.com" with default driver
Creating peer0.org1.example.com ... done
Creating orderer.example.com    ... done
Creating peer0.org2.example.com ... done
Creating cli                    ... done
```

Criação de canais

- ./network.sh createChannel

Channel 'mychannel' joined

Vamos criar mais de um canal

- ./network.sh createChannel -c channel1
- ./network.sh createChannel -c channel2

Caso se queira levantar o ambiente com o canal

- ./network.sh up createChannel

Execução - chaincode

Em Hyperledger o Smart contract chama-se chaincode

- Instala o **chaincode peer chaincode install** (procurar) para ambos peers
- Instância o chaincode **peer chaincode instantiate** e envia argumentos para o método (procurar comando e identificar métodos
- Consulta o peer com o comando **peer chaincode query**
- Executa o invoke **peer chaincode invoke** – transfere asset6 para Christopher
- Instala o chaincode em outro peer **chaincode peer chaincode install**
- Consulta o chaincode em outro peer (peer1 org2)
- Retorna o valor de asset6

Preparando para interagir com o Chaincode

- export CORE_PEER_TLS_ENABLED=true
- export CORE_PEER_LOCALMSPID="Org1MSP"
- export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
- export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
- export CORE_PEER_ADDRESS=localhost:7051

Deploy do Chaincode

- ./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go

Preparando para interagir com o Chaincode

- export PATH=${PWD}/../bin:$PATH
- export FABRIC_CFG_PATH=$PWD/../config/

Exportemos as variáveis para nos conectarmos com a primeira organização

- export CORE_PEER_TLS_ENABLED=true
- export CORE_PEER_LOCALMSPID="Org1MSP"
- export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
- export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
- export CORE_PEER_ADDRESS=localhost:7051

Interagindo com o Chaincode

Vamos inicializar o ledger

- peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"InitLedger","Args":[]}’

-> INFO 001 Chaincode invoke successful. result: status:200

Interagindo com o Chaincode

Vamos a consultar o ledger

- peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'


```
[
  {"ID": "asset1", "color": "blue", "size": 5, "owner": "Tomoko", "appraisedValue": 300},
  {"ID": "asset2", "color": "red", "size": 5, "owner": "Brad", "appraisedValue": 400},
  {"ID": "asset3", "color": "green", "size": 10, "owner": "Jin Soo", "appraisedValue": 500},
  {"ID": "asset4", "color": "yellow", "size": 10, "owner": "Max", "appraisedValue": 600},
  {"ID": "asset5", "color": "black", "size": 15, "owner": "Adriana", "appraisedValue": 700},
  {"ID": "asset6", "color": "white", "size": 15, "owner": "Michel", "appraisedValue": 800}
]
```

Interagindo com o Chaincode

Vamos a gravar no ledger

- peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"TransferAsset","Args":["asset6","Christopher"]}'

-> INFO 001 Chaincode invoke successful. result: status:200

Trocando de Organização

Exportemos as variáveis para nos conectarmos com a segunda organização

- export CORE_PEER_TLS_ENABLED=true
- export CORE_PEER_LOCALMSPID="Org2MSP"
- export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
- export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
- export CORE_PEER_ADDRESS=localhost:9051

Interagindo com o chaincode da 2da Organização

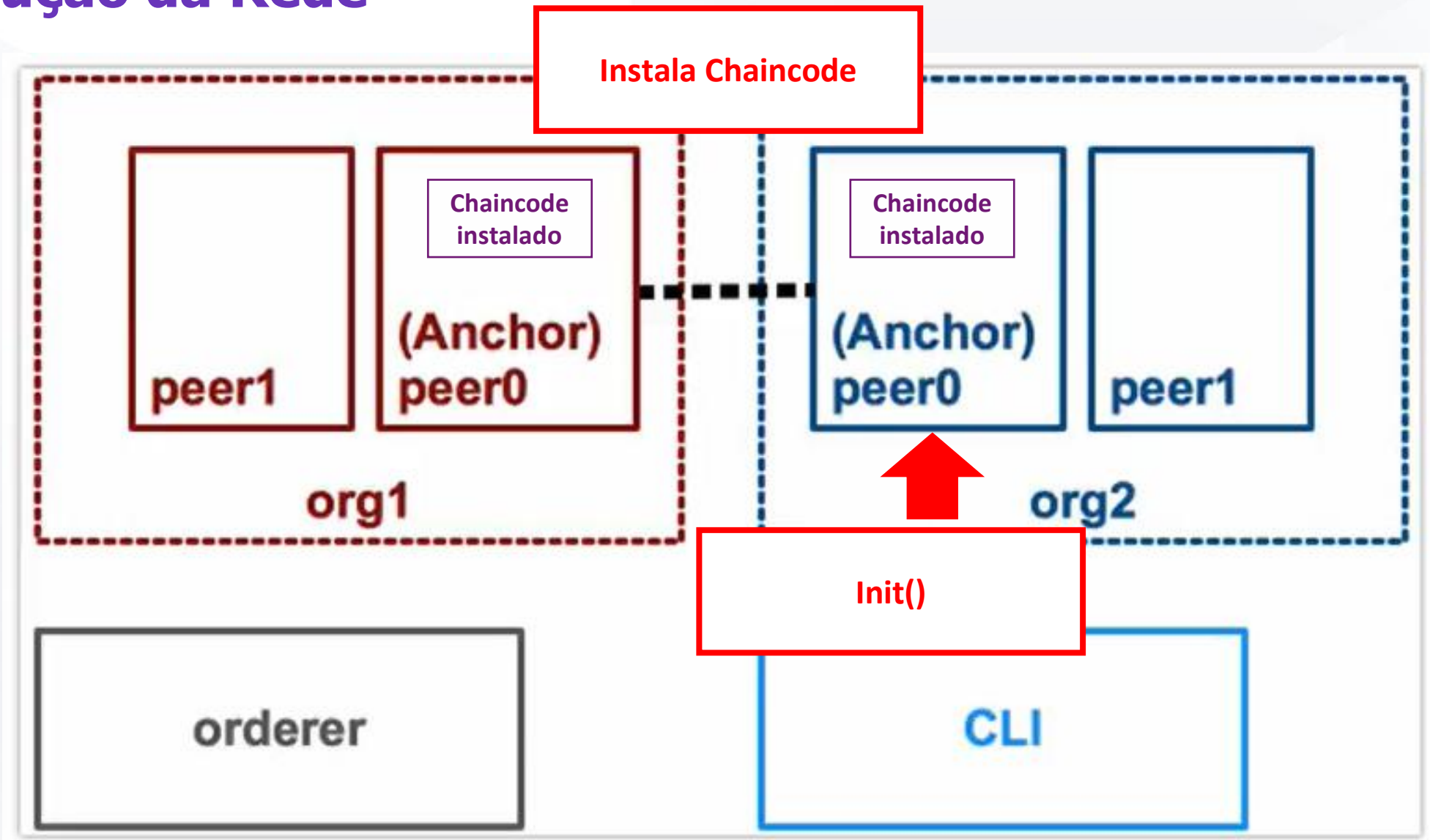
- peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","asset6"]}'

{"ID":"asset6","color":"white","size":15,"owner":"Christopher","appraisedValue":800}

Você concluiu o laboratório!

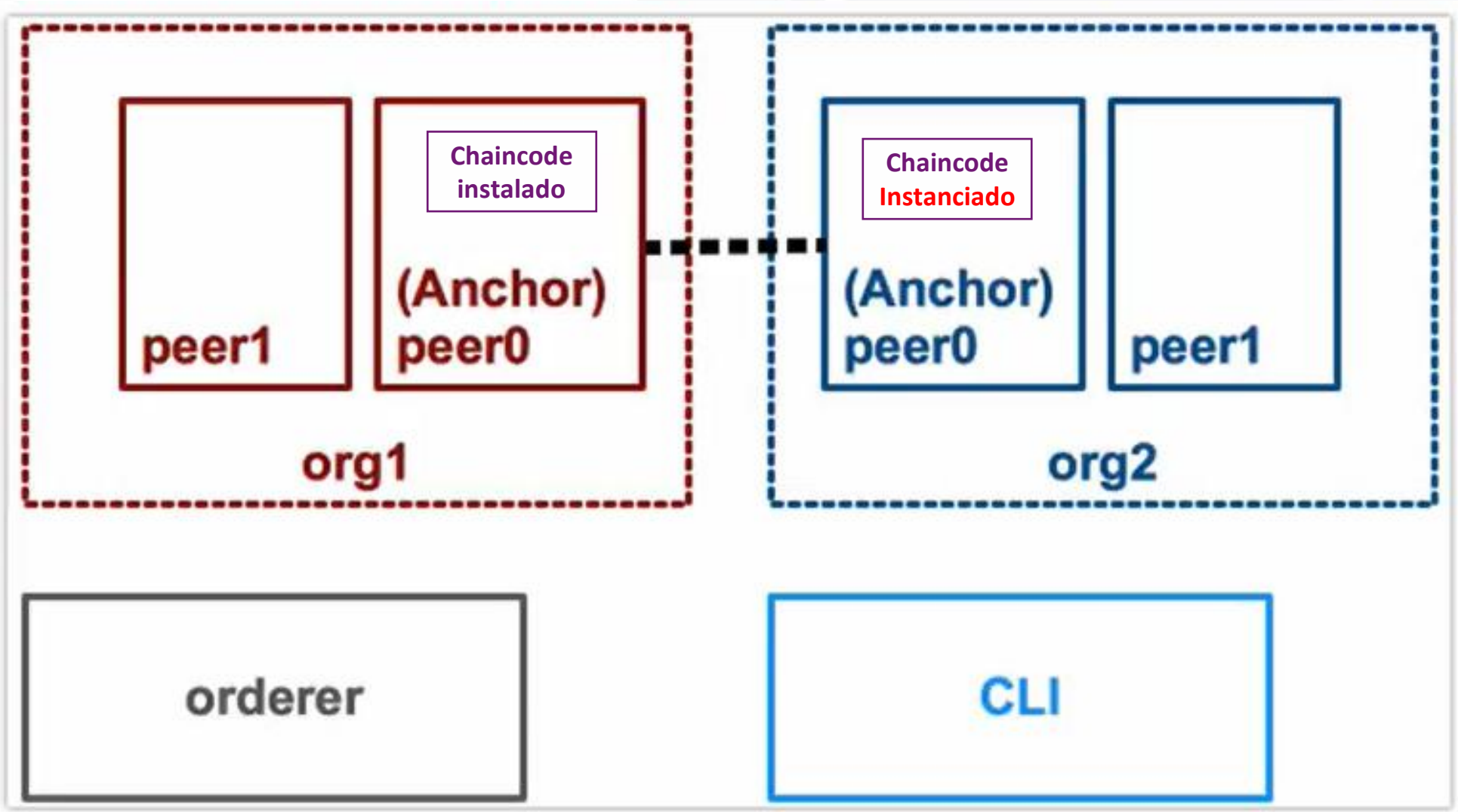
BLOCKCHAIN DEVELOPER HYPERLEDGER

Execução da Rede



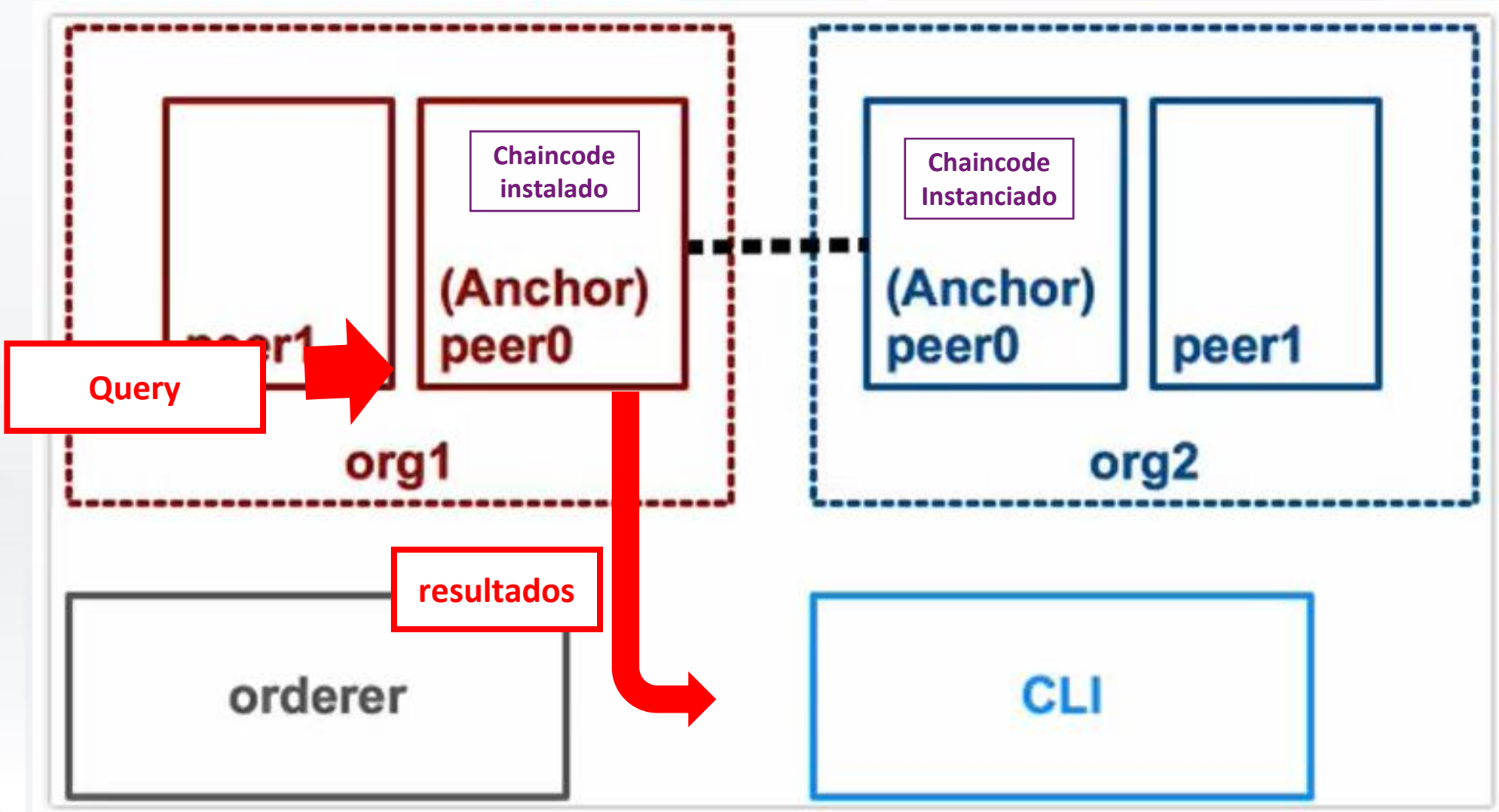
BLOCKCHAIN DEVELOPER HYPERLEDGER

Execução da Rede



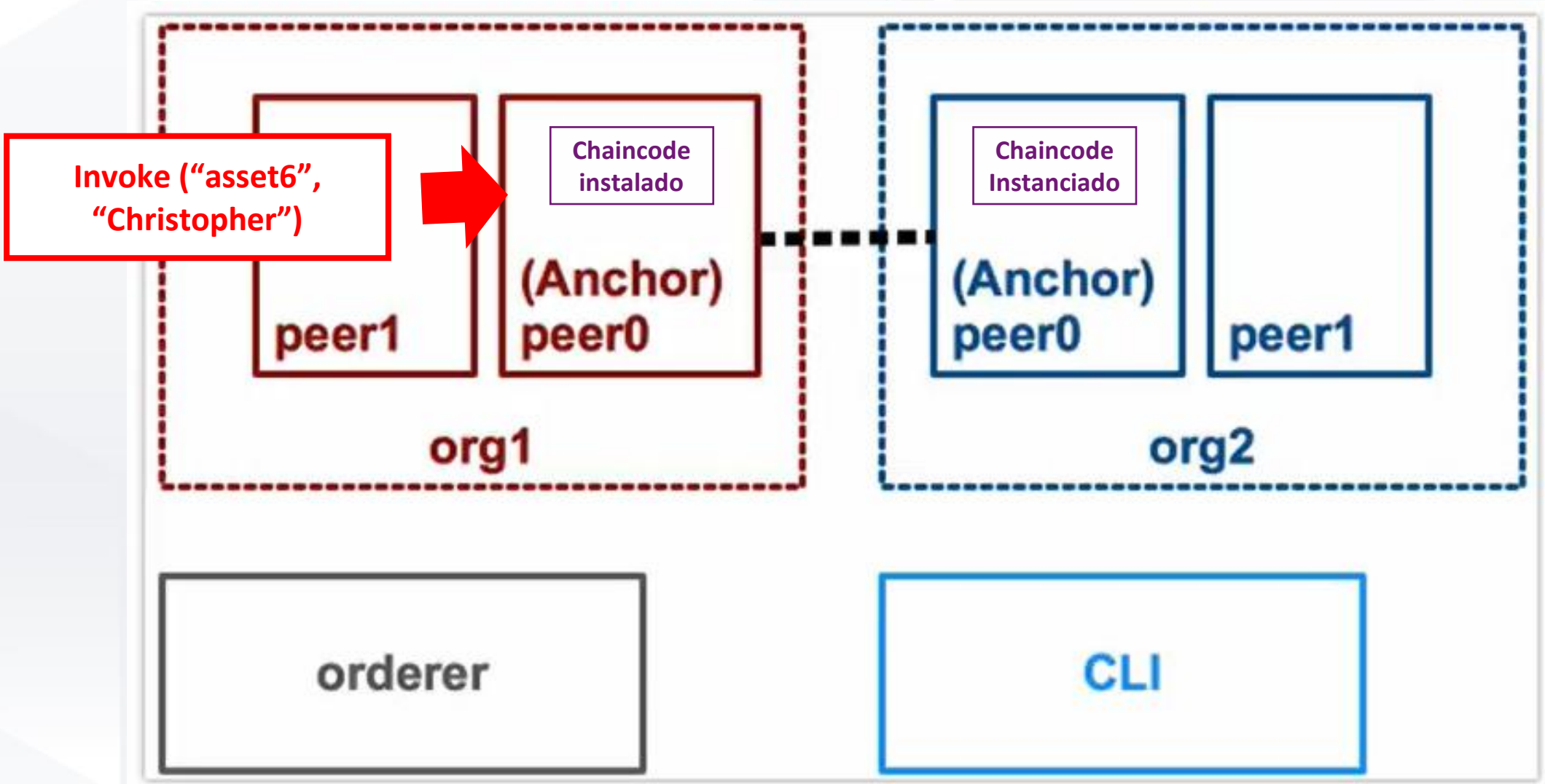
BLOCKCHAIN DEVELOPER HYPERLEDGER

Execução da Rede



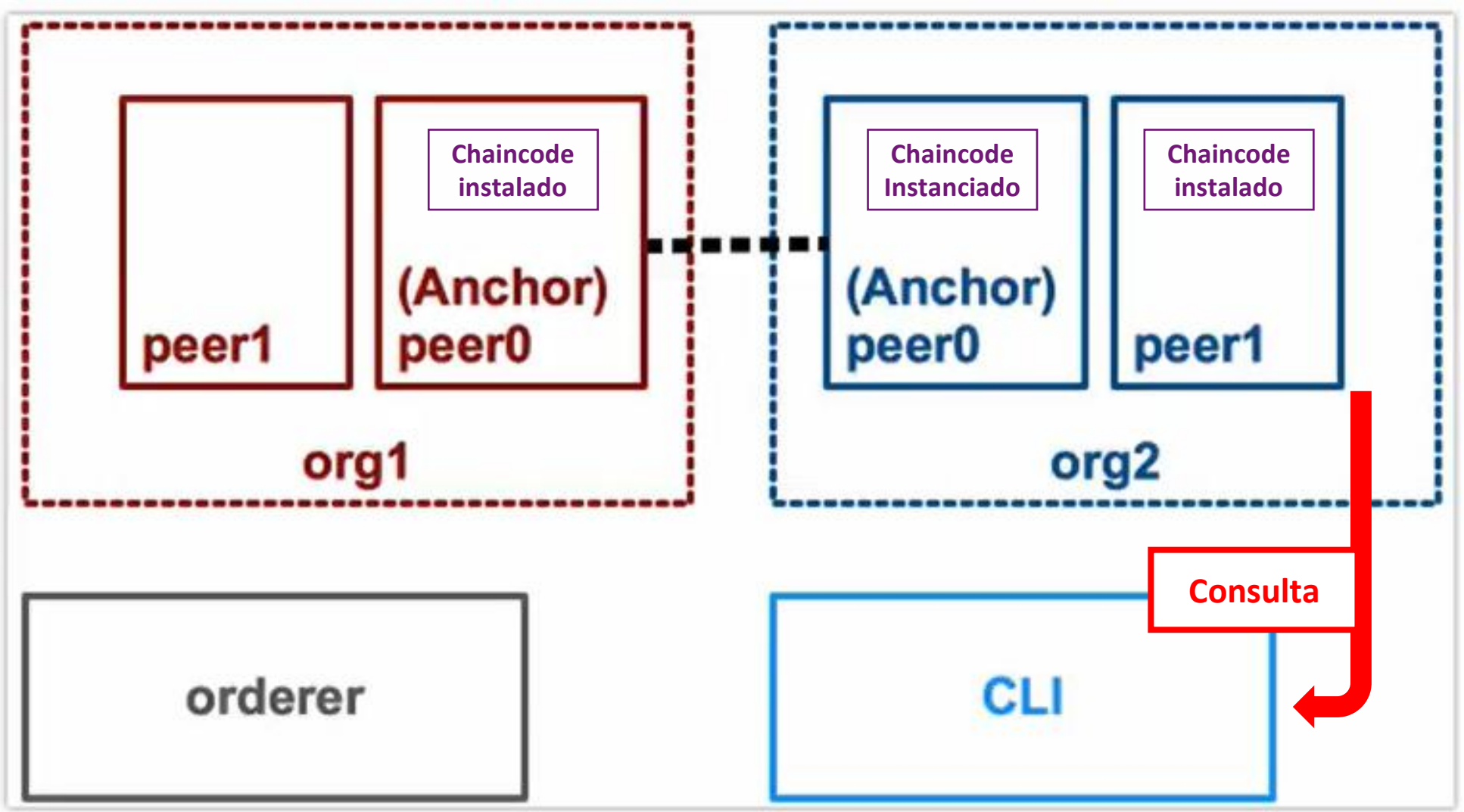
BLOCKCHAIN DEVELOPER HYPERLEDGER

Execução da Rede

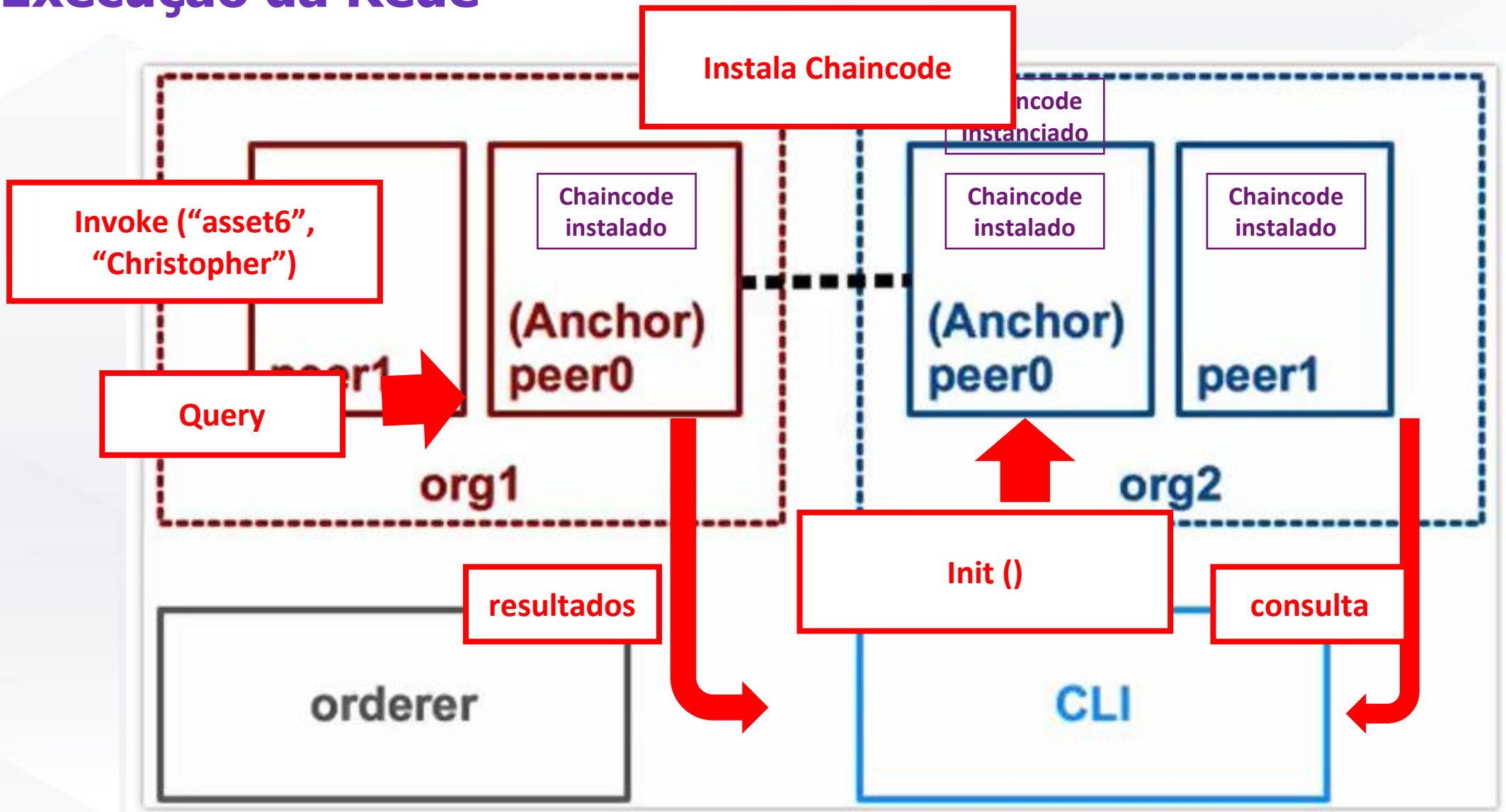


BLOCKCHAIN DEVELOPER HYPERLEDGER

Execução da Rede



Execução da Rede



BLOCKCHAIN DEVELOPER HYPERLEDGER

MÓDULO VI

Fim