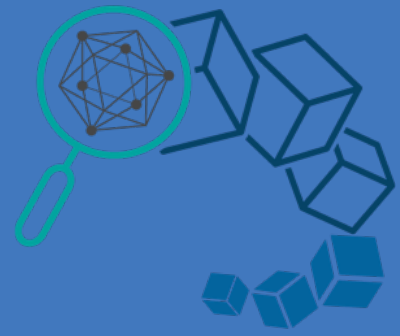
HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS



Hyperledger Fabric Deployment

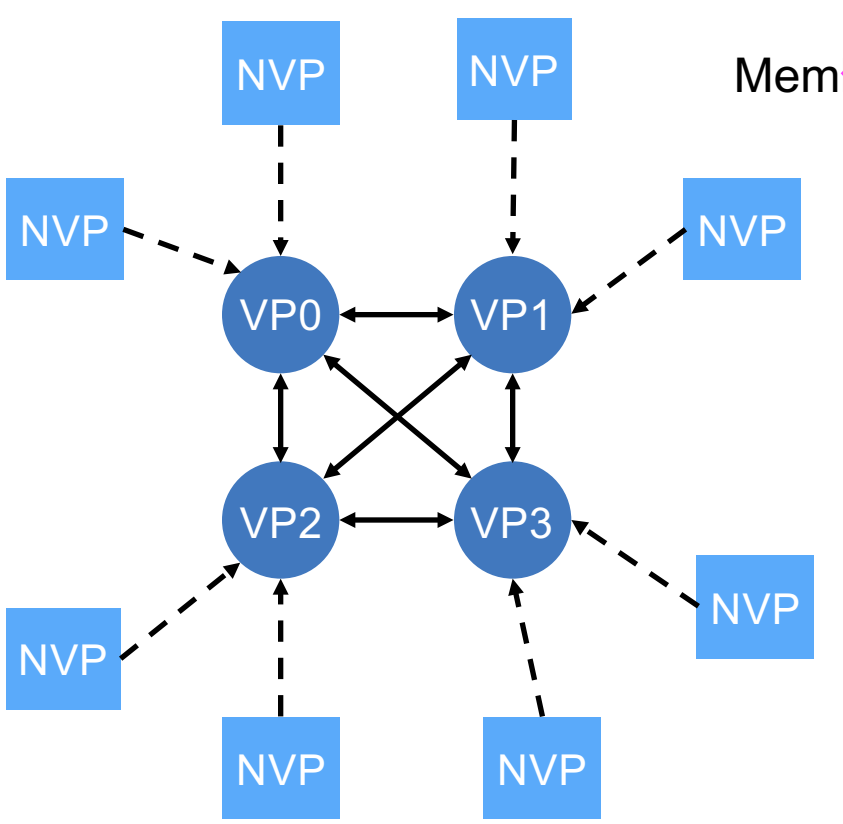
Baohua Yang

Nov, 2017

About Me

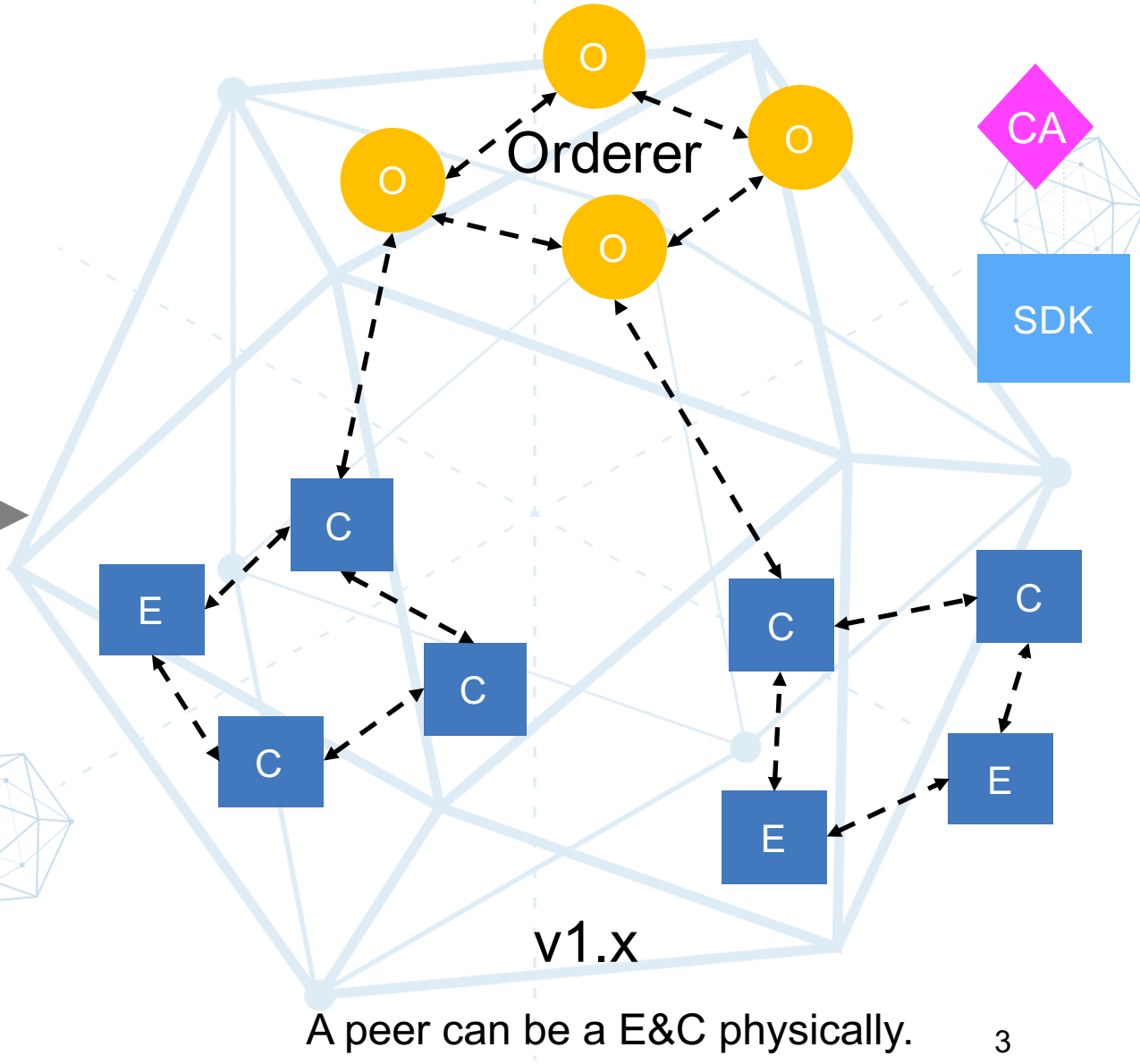
- **Interested Areas**
  - Fintech, Cloud and Analytics
- **Technical Leader**
  - Senior Researcher/Architect in IBM, Oracle
- **Open-Source Contributor**
  - Hyperledger, OpenStack, OpenDaylight, etc.
- **Hyperledger Developer**
  - Core designer & committer of Fabric, Cello, sdk etc.
  - Hyperledger Technical Steering Committee (TSC) Member
  - Hyperledger Technical Working Group China Chair

Fabric 1.x Deployment Scenarios



Member Service

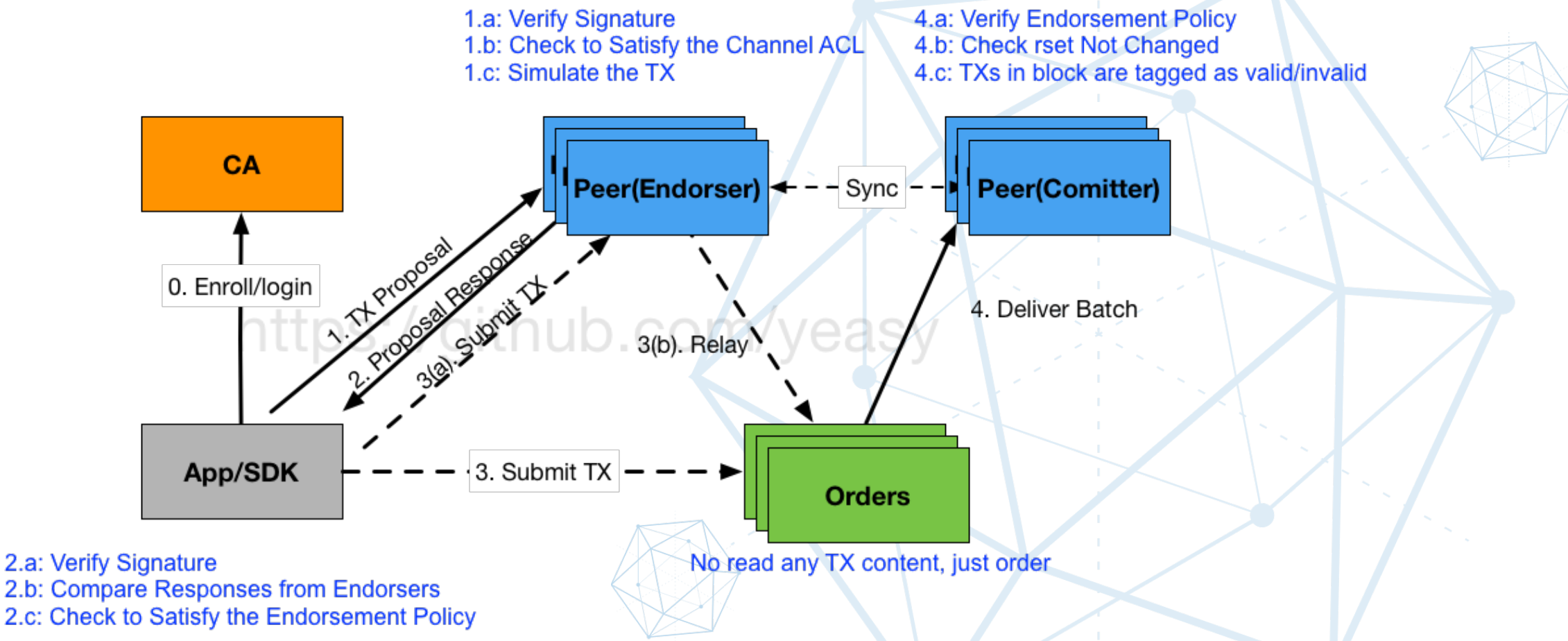
v0.6



v1.x

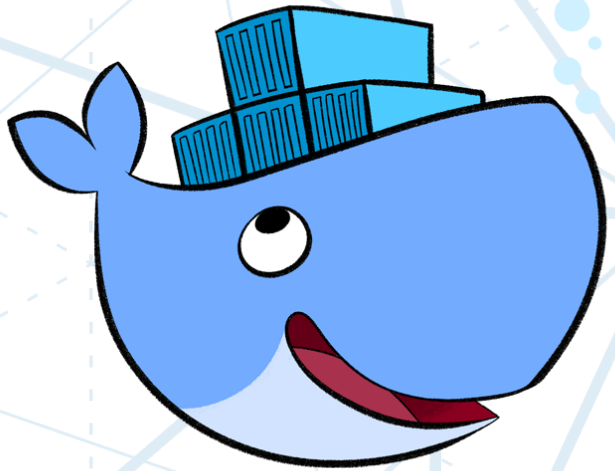
A peer can be a E&C physically.

Fabric 1.0 Workflow



Environment Setup – Docker Installation

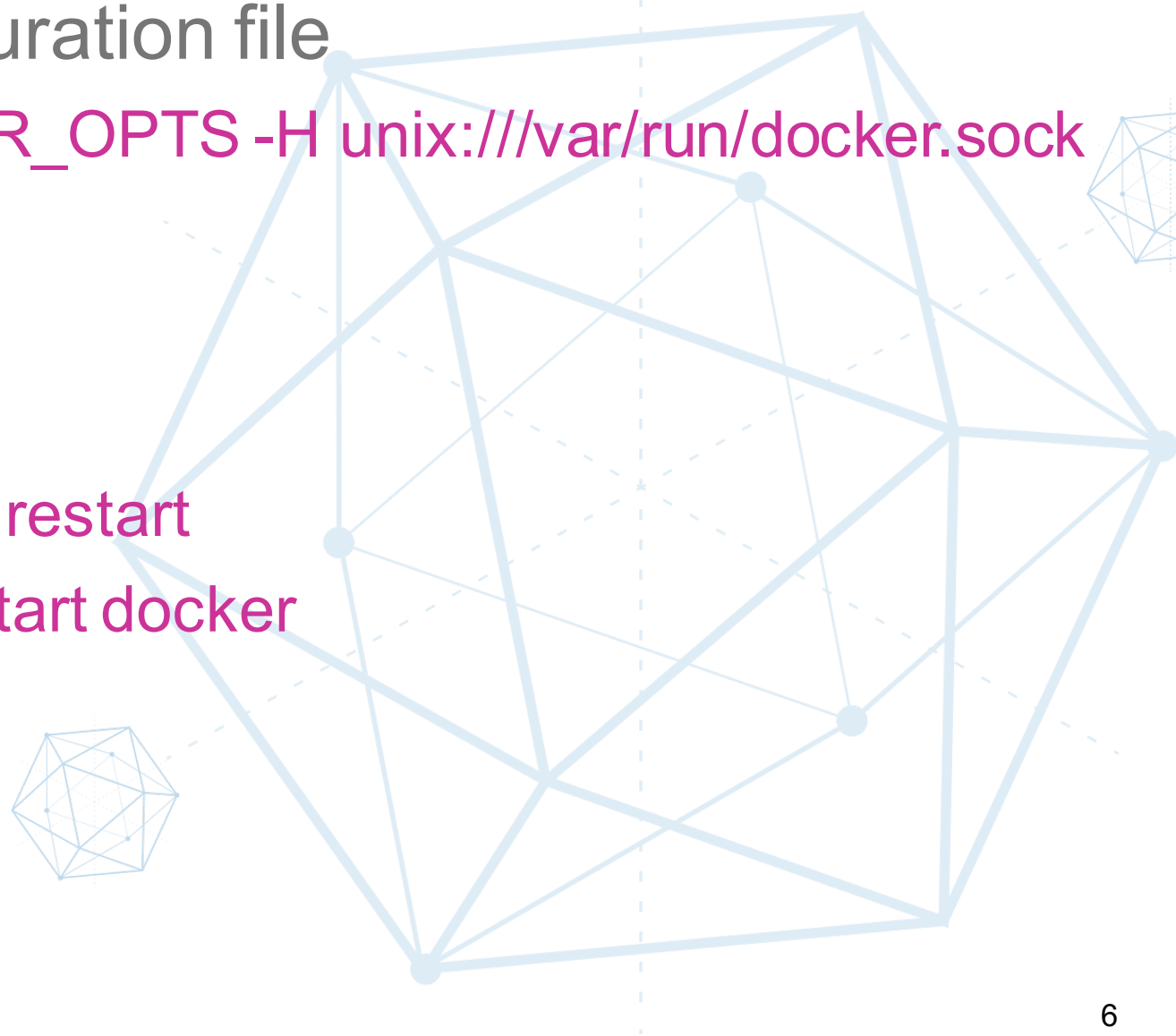
- Docker 1.12+
- Linux
  - 64 bit
  - kernel 3.10+
  - curl -sSL https://get.docker.com/ | sh
- Mac
  - Docker for Mac
- Docker-Compose 1.7.0+
  - pip install docker-compose>=1.7.0



* Non-container deployments are supported.

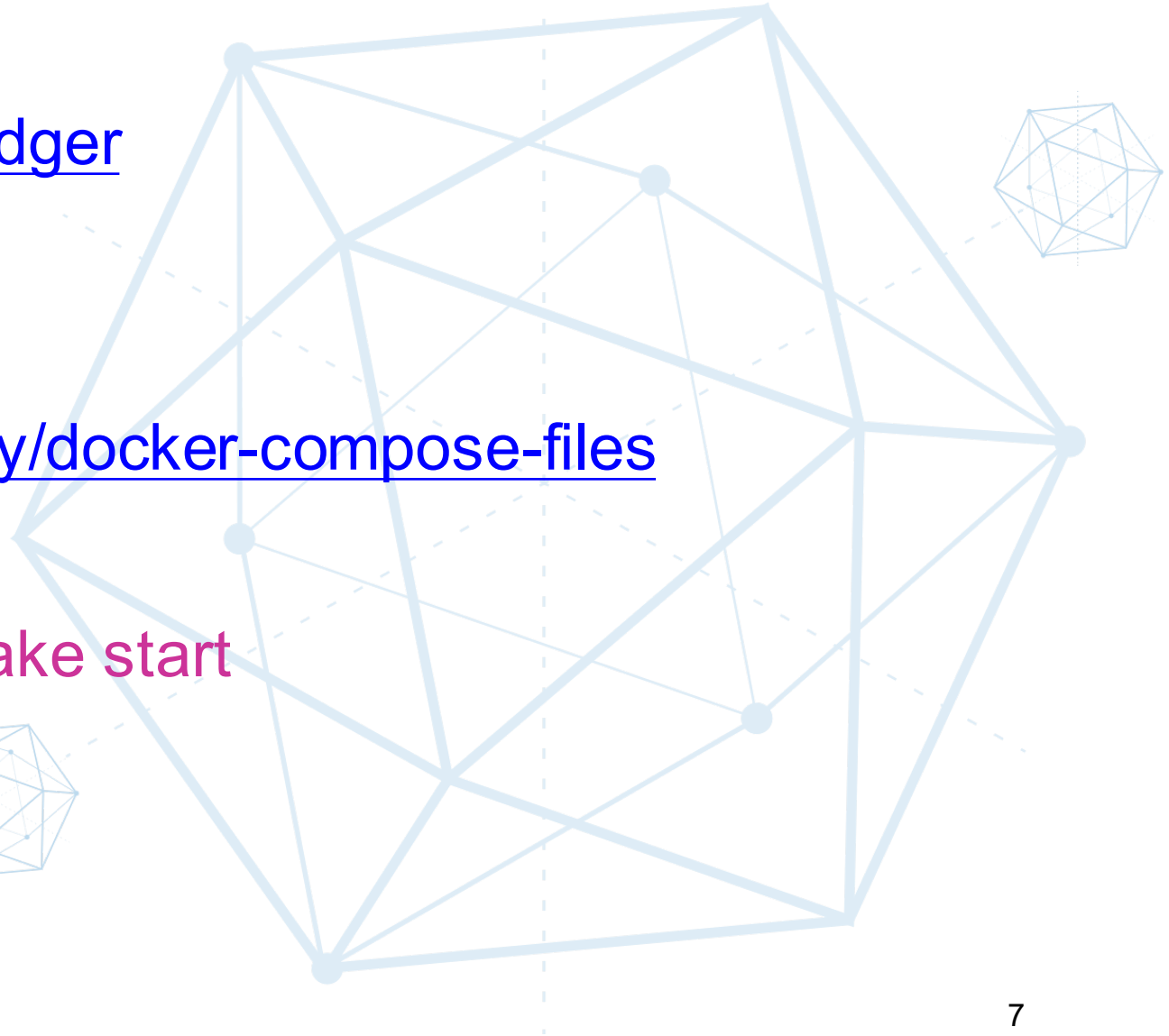
Environment Setup - Configuration

- Update the Docker configuration file
  - DOCKER_OPTS="$DOCKER_OPTS -H unix:///var/run/docker.sock -H tcp://0.0.0.0:2375"
- Restart Docker Daemon
  - Upstart: sudo service docker restart
  - Systemd: sudo systemctl restart docker



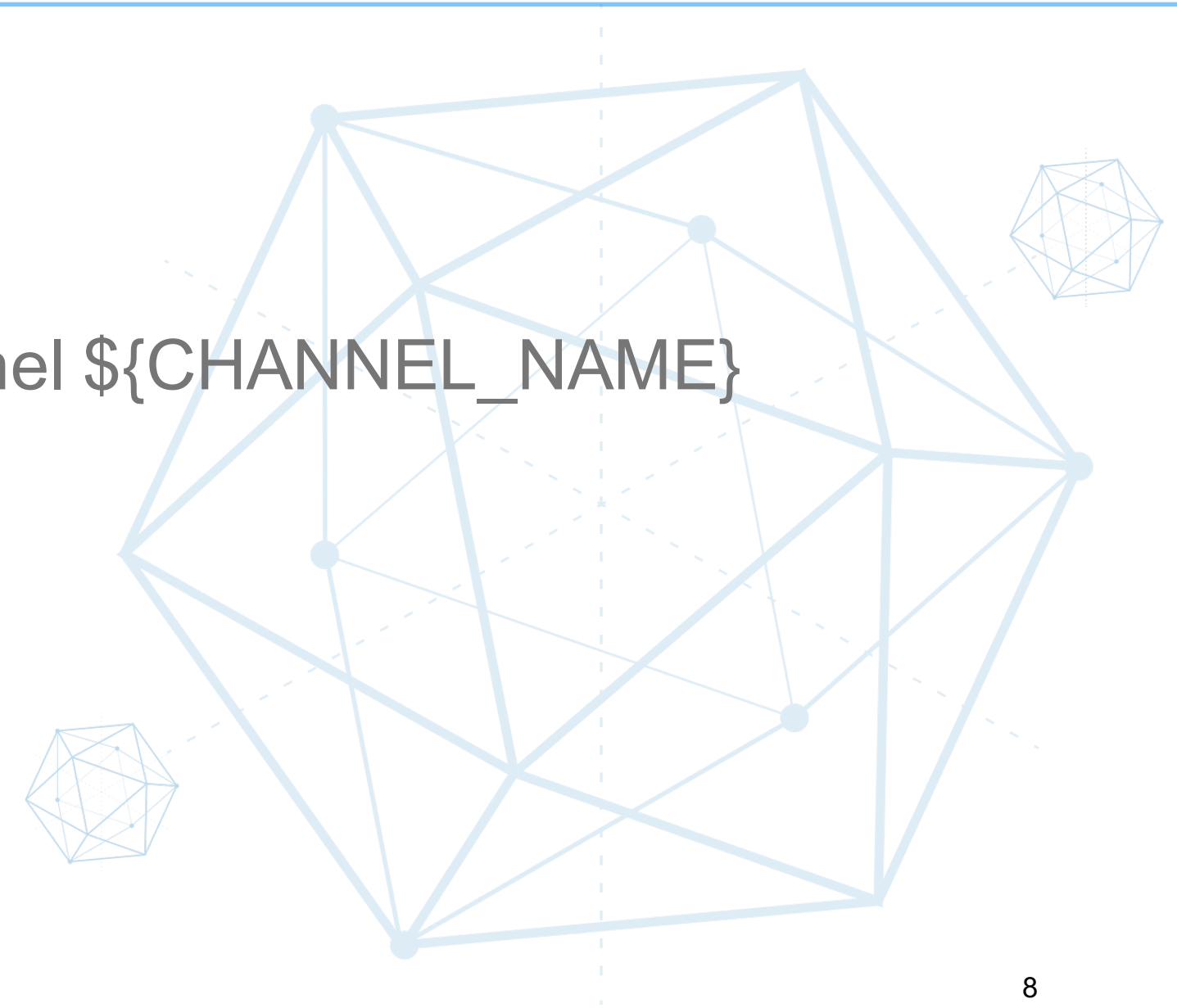
Fabric 1.x Bootup in 3 steps

- Get Docker images
  - https://hub.docker.com/r/hyperledger
  - https://hub.docker.com/r/yeasy
- Get Compose file
  - git clone https://github.com/yeasy/docker-compose-files
- Start fabric
  - cd hyperledger_fabric/1.0.4 & make start



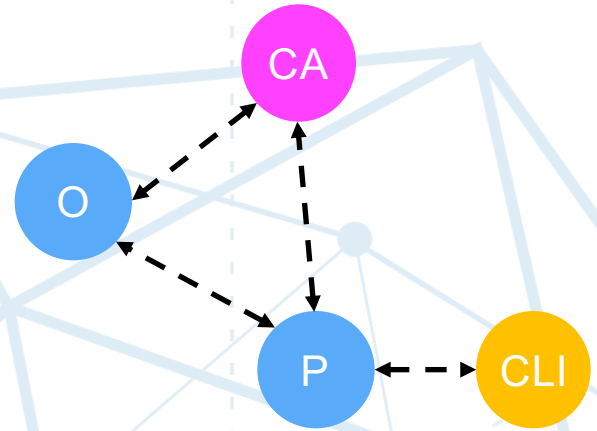
Topology

- 2 organizations
  - Org1: peer0, peer1
  - Org2: peer0, peer1
- Joined to the same channel ${CHANNEL_NAME}



Play Transactions

- Check container status
  - make ps
- Enter the cli container
  - make cli




```
$ make ps
CONTAINER ID        IMAGE                                    COMMAND                  CREATED
f6686986fe18        hyperledger/fabric-tools:x86_64-1.0.4    "bash -c 'cd /tmp;..."   6 seconds ago
c7f274bf60bc        yeasy/hyperledger-fabric-peer:1.0.4      "peer node start"        6 seconds ago
c6c5f69f2d53        yeasy/hyperledger-fabric-peer:1.0.4      "peer node start"        6 seconds ago
3cad0c519e6f        yeasy/hyperledger-fabric-peer:1.0.4      "peer node start"        6 seconds ago
8b371209f6b8        yeasy/hyperledger-fabric-peer:1.0.4      "peer node start"        6 seconds ago
ba1f00a9c83c        hyperledger/fabric-orderer:x86_64-1.0.4  "orderer start"          6 seconds ago
```

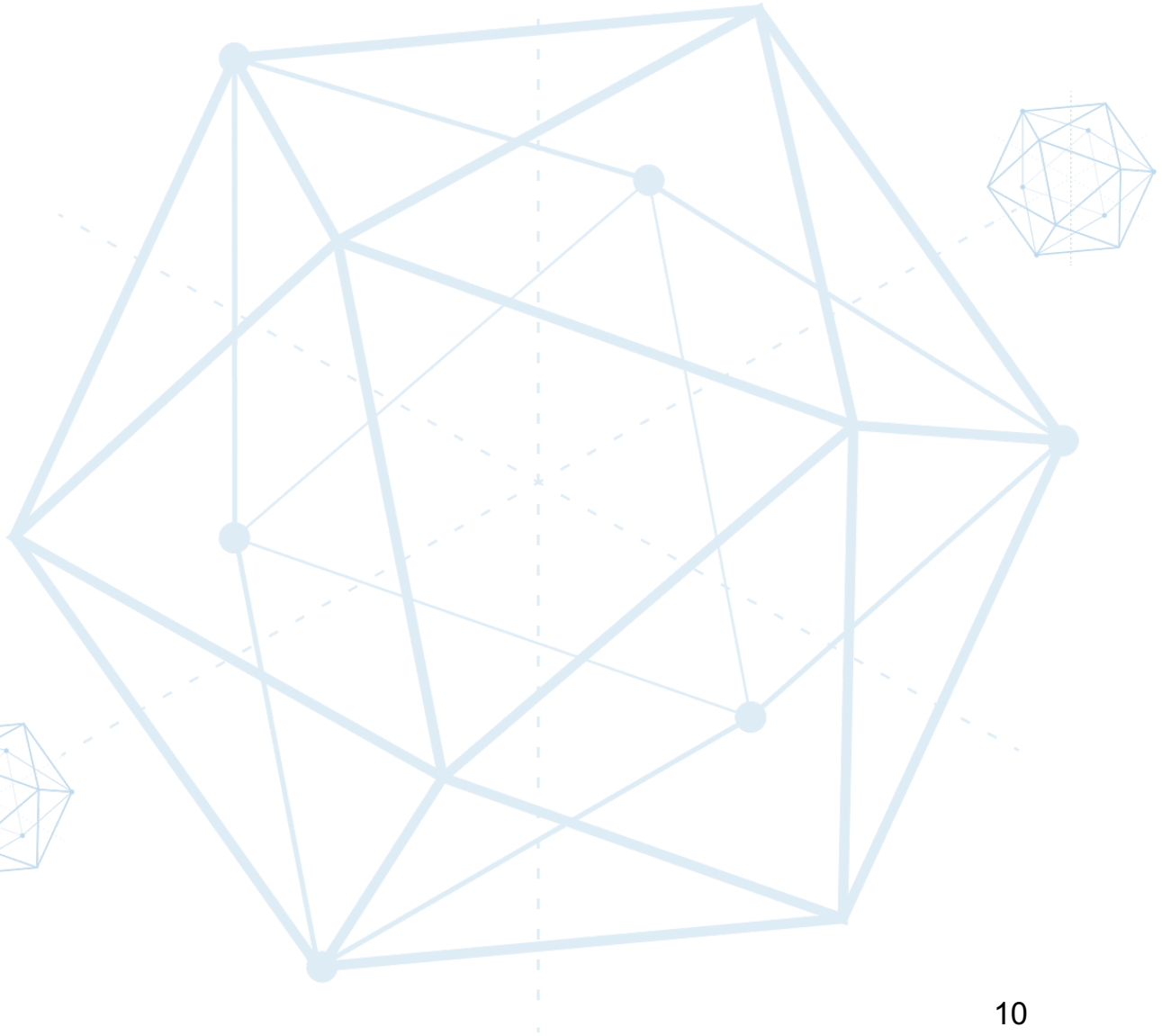
Steps for E2E chaincode

- make restart
- make test_channel

- make test_cc_install
- make test_cc_instantiate

- make test_cc_invoke_query

- make stop clean



Steps for E2E chaincode

- *make cli*
- *cd /tmp/ && source scripts/func.sh*

- *chaincodeQuery* **$**{*CHANNEL_NAME*} 1 0 **$**{*CC_NAME*} **$**{*CC_QUERY_ARGS*} #100
- *chaincodeInvoke* **$**{*CHANNEL_NAME*} 1 0 **$**{*CC_NAME*} **$**{*CC_INVOKE_ARGS*}
- *chaincodeQuery* **$**{*CHANNEL_NAME*} 1 0 **$**{*CC_NAME*} **$**{*CC_QUERY_ARGS*} #90

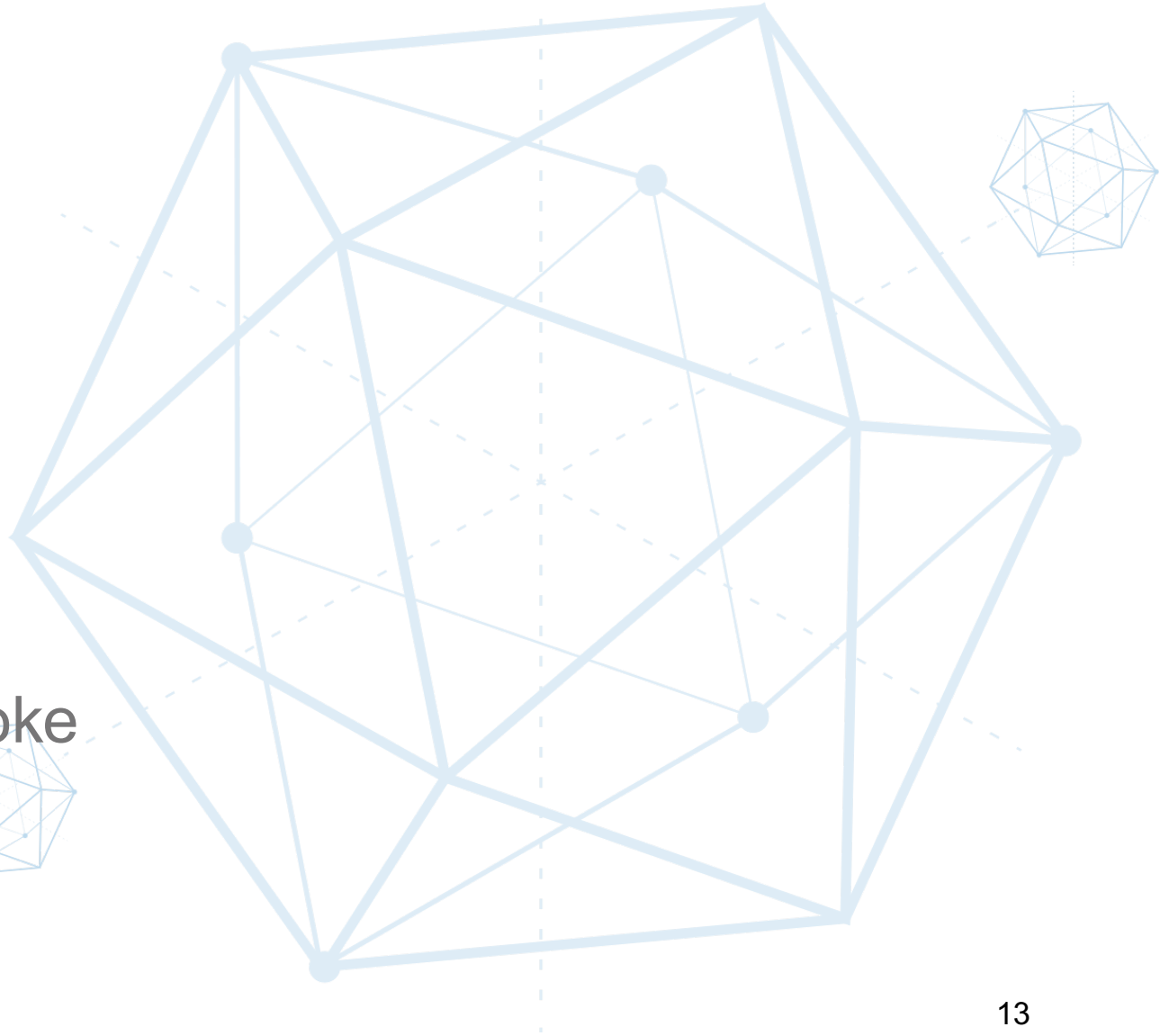
Play Transactions cont.

- Install/instantiate chaincode
  - CC_PATH=github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02
  - peer chaincode install -v 1.0 -n test_cc -p $CC_PATH -c '{"Args":["init", "a", "100", "b", "200"]}'
  - peer chaincode instantiate -v 1.0 -n test_cc -p $CC_PATH -c '{"Args":["init", "a", "100", "b", "200"]}' -o orderer0:7050
- Invoke chaincode
  - peer chaincode invoke -n test_cc -c '{"Args":["query","a"]}'
  - peer chaincode invoke -n test_cc -c '{"Args":["invoke","a","b","10"]}'

```
$ docker ps
CONTAINER ID        IMAGE                          COMMAND
c0abb4b9206b        dev-peer0-test_cc-1.0          "chaincode -peer.a..."
c1cf099e1f76        hyperledger/fabric-peer        "bash -c 'while tr..."
0b67c42fd5cc        hyperledger/fabric-peer        "peer node start -..."
80b5fb85636e        hyperledger/fabric-orderer     "orderer"
f3680e5889b0        hyperledger/fabric-ca          "fabric-ca-server ..."
```

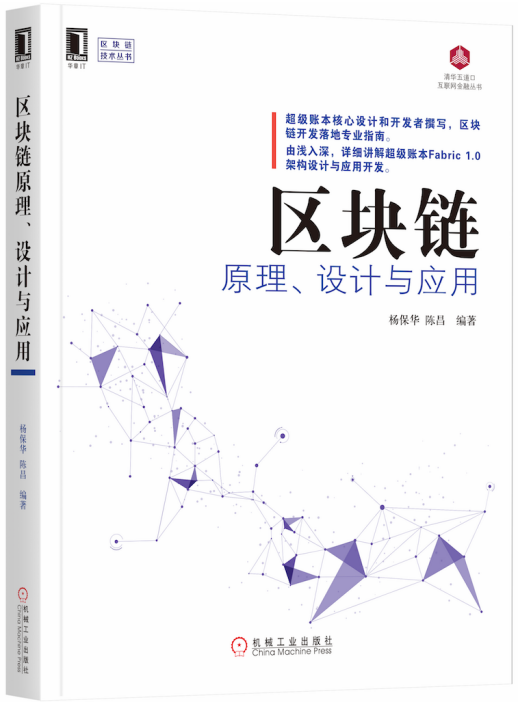
More on Using Fabric

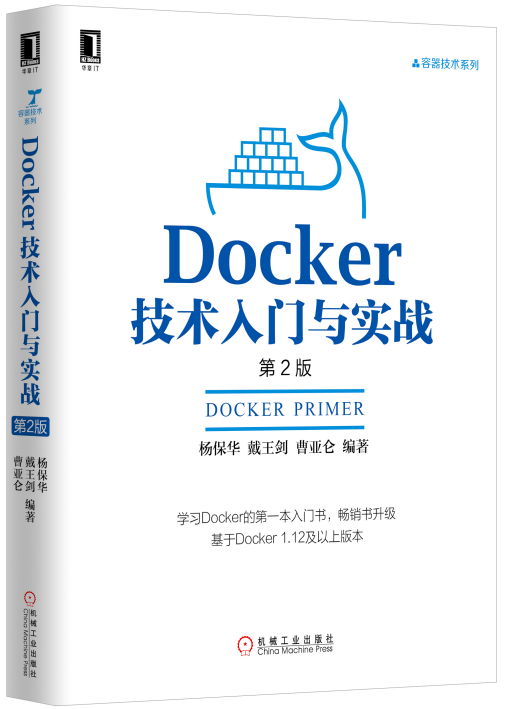
- Application interactions
  - APIs: gRPC
  - SDK: Node, Python, Java
- Commands
  - Peer start/stop
  - Channel create/join
  - User enroll/login
  - Chaincode install/instantiate/invoke



Reference

- Hyperledger Project
- Hyperledger Wiki
- 《区块链原理设计与应用》
- 《Docker 技术入门与实战》
- github.com/yeasy/blockchain_guide
- github.com/yeasy/docker_practice





HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Questions?

Thank You!
@baohua

Slides available at github.com/yeasy/seminar-talk#hyperledger