IBM微讲堂互动课堂

- Zoom 100人，同时录像，斗鱼实时转播
- 要求全程静音

- 有问题可以通过以下两种方式提问：
  - 在聊天窗口，输入问题
  - 举手，主持人允许后，可以语音提问

广告：公有云IBM Cloud

- Data & Analytics
- DevOps
- Finance
- Functions
- Integrate
- Internet of Things
- Mobile
- Network
- Schematics
- Security
- Storage
- Watson
- Web Apps New

BM Blockchain
latform Features

ccelerate your blockchain network
ith tools to develop, govern &
perate decentralized ecosystems

earn more →

Operate

Regulated industries need to deploy and operate decentralized networks with a production-ready, hyper-secure service. IBM Blockchain Platform allows networks to start small then scale elastically without disruption as membership and transaction volumes increase, and it is built on the world's most performant Linux compute optimized to deliver transaction speed and integrity. **More**

Govern

Starting up a blockchain network and governing it across a group of members once it is operational can take significant time and effort. Only IBM Blockchain Platform provides a built in, democratic and decentralized network management experience assuring no one member has complete control. **More**

Develop

IBM Blockchain has the right set of tools to get any programmer quickly started developing blockchain applications. Developers begin by experimenting with existing samples and tutorials online, then move on to create their own application with a local development environment or collaborate with others in the cloud. **More**

Fabric源码入门

郭剑南
Email: guojiannan@cn.ibm.com
Rocket.chat: @guoger

课程目标

- 本系列课程主要面向对于Fabric基本概念已经有所了解的同学（Fabric基础课程请见ibm.biz/opentech-ma）

- 通过学习Fabric源码，进一步了解Fabric工作机制，解答实验和使用中碰到的问题，进行二次开发，贡献社区。

课程安排

- 1月18日《Fabric源码入门和排序节点解析》
  通过e2e_cli这个例子，初步了解Fabric中一个典型交易处理过程的内部实现，以及Orderer的内部工作原理

- 1月25日《Fabric Peer解析》
  分析Fabric Peer实现，包括交易处理过程和账本结构

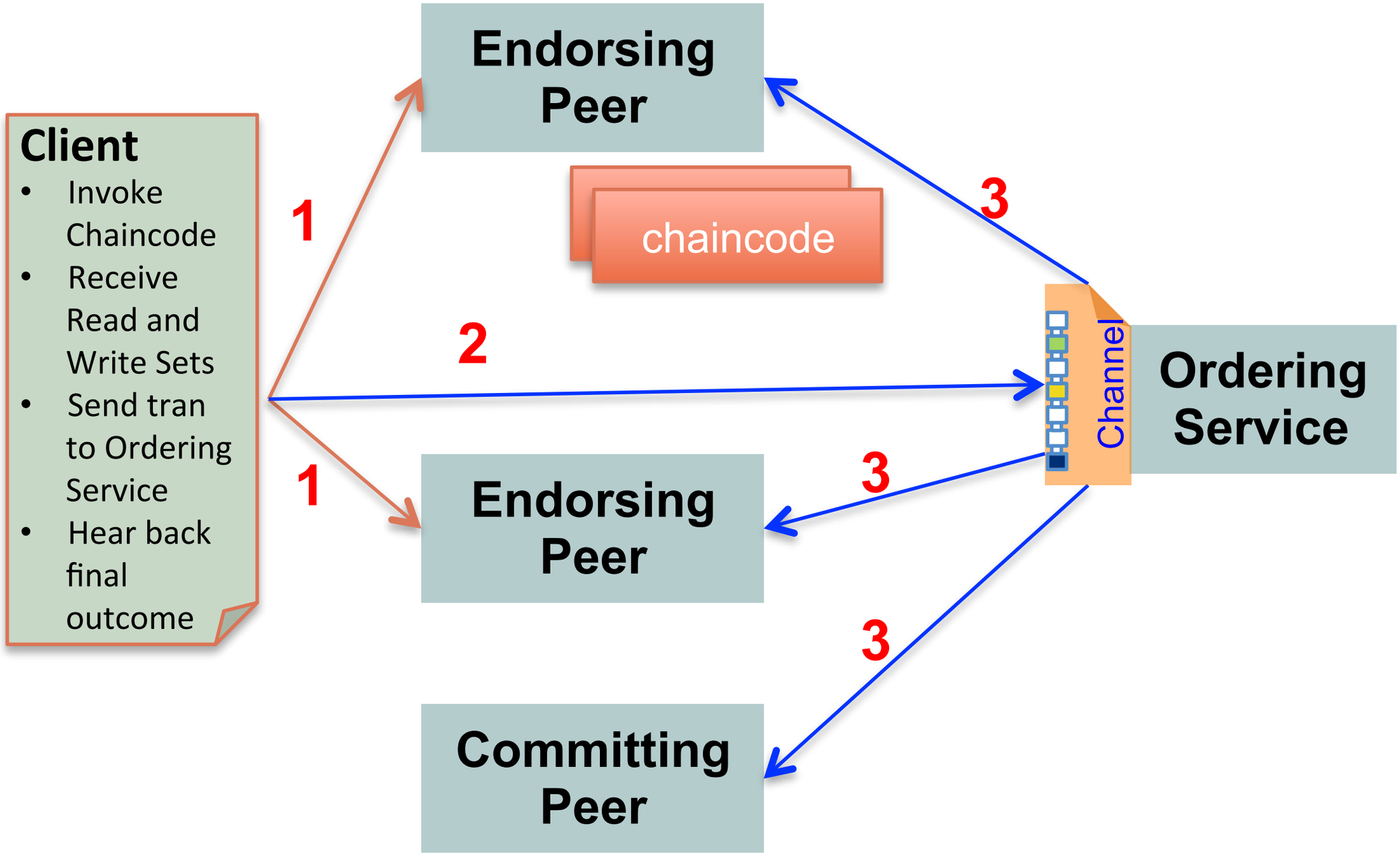
- 2月1日《Fabric Chaincode, SDK》
  分析Fabric Chaincode的接口和实现，以及SDK的设计

Fabric源码入门

- 一个典型的Fabric工作流
  - create channel
  - join channel
  - install chaincode
  - instantiate chaincode
  - chaincode query/invoke

- Orderer解析
  - 通过broadcast接口提交一笔交易
  - 通过deliver接口获得一个区块

- Fabric社区贡献流程


Client
• Invoke Chaincode
• Receive Read and Write Sets
• Send tran to Ordering Service
• Hear back final outcome
Endorsing Peer
chaincode
Endorsing Peer
Committing Peer
Channel
Ordering Service
1
1
2
3
3
3

create channel

- Orderer <- ConfigUpdate Tx targeting non-existing channel
- Config Tx is separated into its own block
- WriteConfigBlock
- create new chain

join channel

- invoke cscc.JoinChain
- CreateLedger
- CreateChain -> start gossip service, leader election and connect to orderer
- InitChain -> DeploySysCCs(cid)

install chaincode

- Collect necessary bits and produce `ChaincodeDeploymentSpec`

- chainless invocation targeting `lscc`

- `lscc` puts spec to filesystem at `$peer.fileSystemPath/chaincodes`

instantiate chaincode

- invocation targeting `lscc` in ‘mychannel’
- build and launch container with github.com/fsouza/go-dockerclient
- Call init of new chain code
- Endorse ProposalResponse
- Broadcast to Orderer so that other peers are aware of deployment

invoke chaincode

- proposal -> `ProcessProposal`
- simulate
- return endorse proposal
- broadcast
- validate
- commit to ledger

Fabric源码入门

- 一个典型的Fabric工作流
  - create channel
  - join channel
  - install chaincode
  - instantiate chaincode
  - chaincode query/invoke

- Orderer解析
  - 通过broadcast接口提交一笔交易
  - 通过deliver接口获得一个区块

- Fabric社区贡献流程

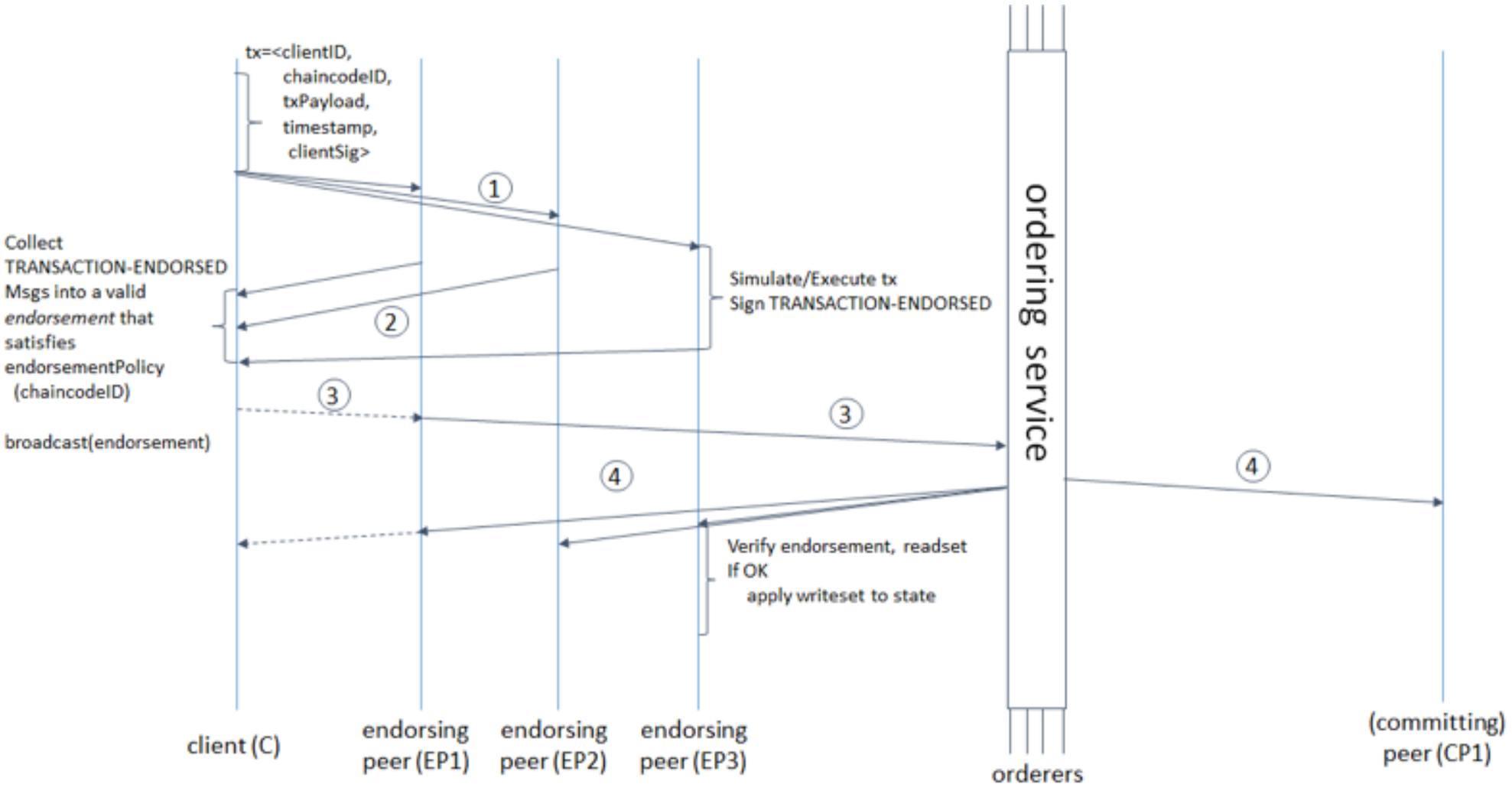
- 排序节点在Fabric中的作用
- 排序节点的运行机制
- 排序节点的配置

- 排序节点在Fabric中的作用
- 排序节点的运行机制
- 排序节点的配置

排序节点在Fabric中的作用

- Atomic Broadcast
- Total Order

排序节点在Fabric中的作用



排序节点在Fabric中的作用

- 通过`**broadcast**`接口，接受client发送的Tx
- 将这些Tx进行**排序**（FIFS）
- 排序后的Tx**根据一定规则打包成区块**（block）
- 通过`**deliver**`接口将区块发送给peer或client
- 排序节点**仅仅是**Fabric共识机制的**一部分**

- 排序节点在Fabric中的作用
- 排序节点的运行机制
- 排序节点的配置

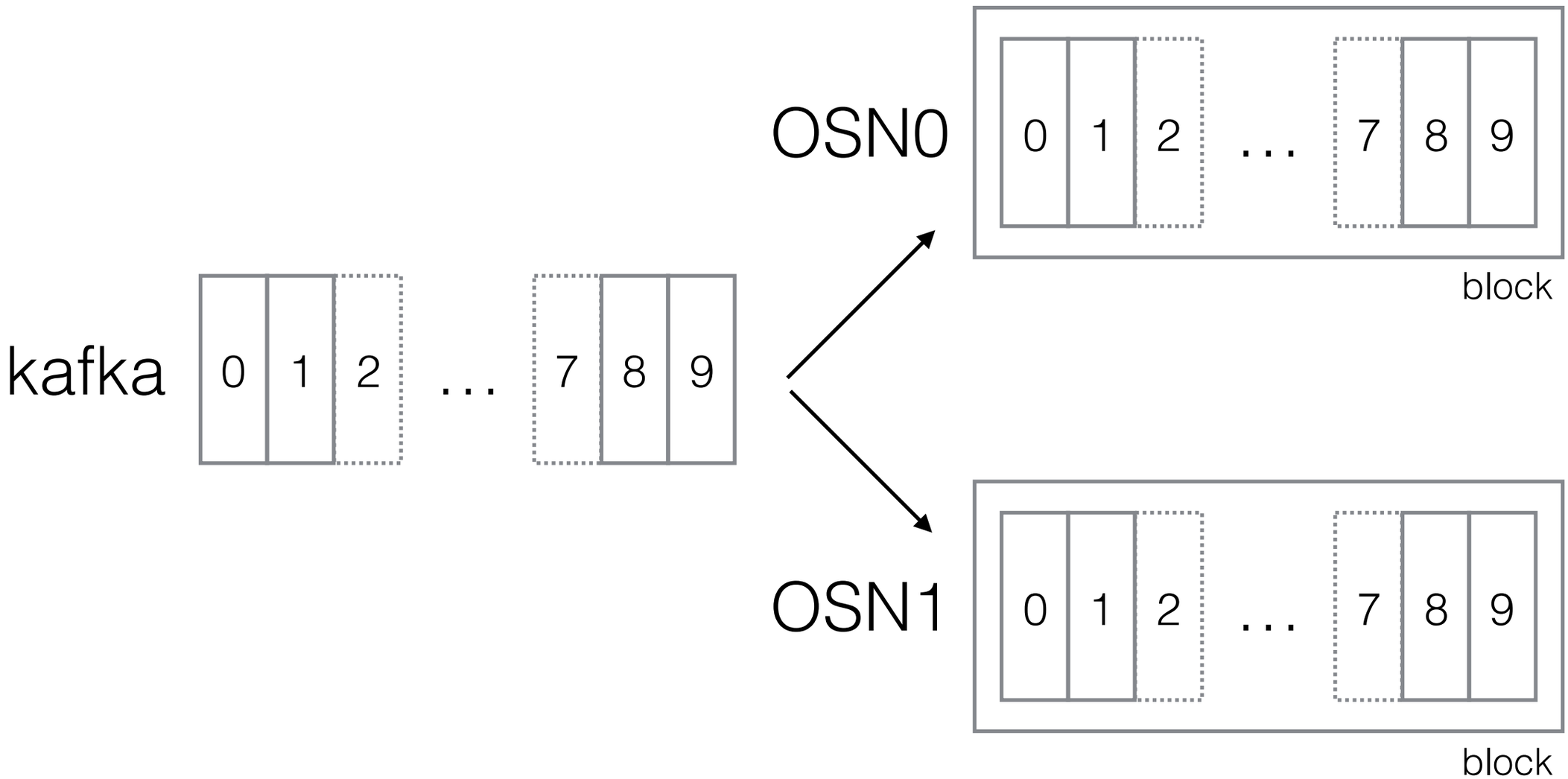
排序节点的运行机制

排序节点将收到的Tx序列化，并按一定规则打包成区块

排序节点的运行机制

- Q：多个排序节点如何对于Tx的顺序达成一致?
  A：使用同一个Kafka partition作为消息队列来序列化收到的Tx

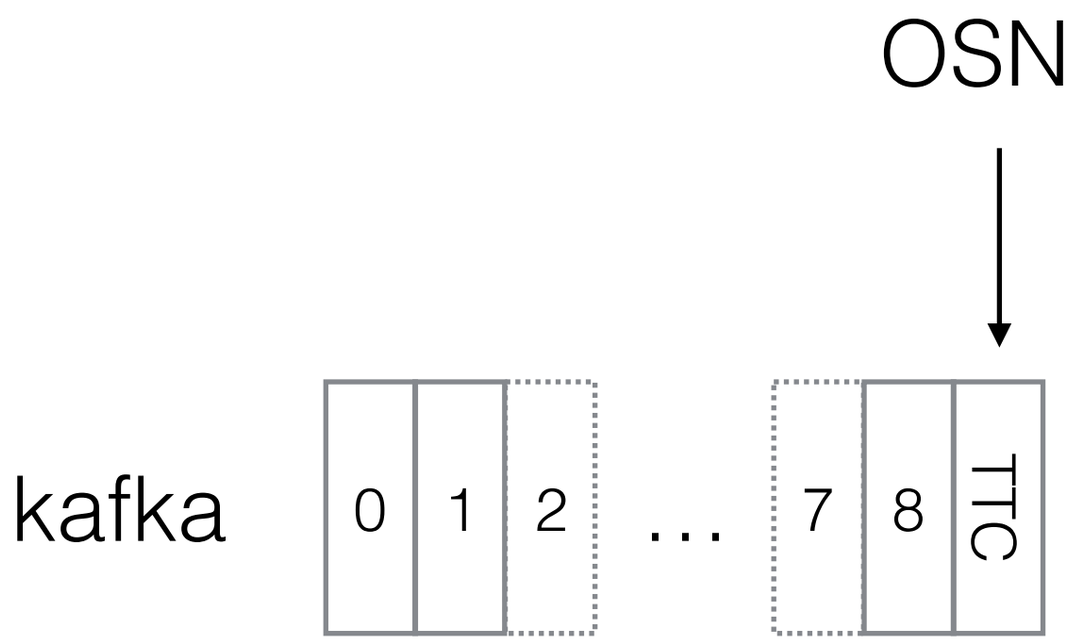
排序节点的运行机制



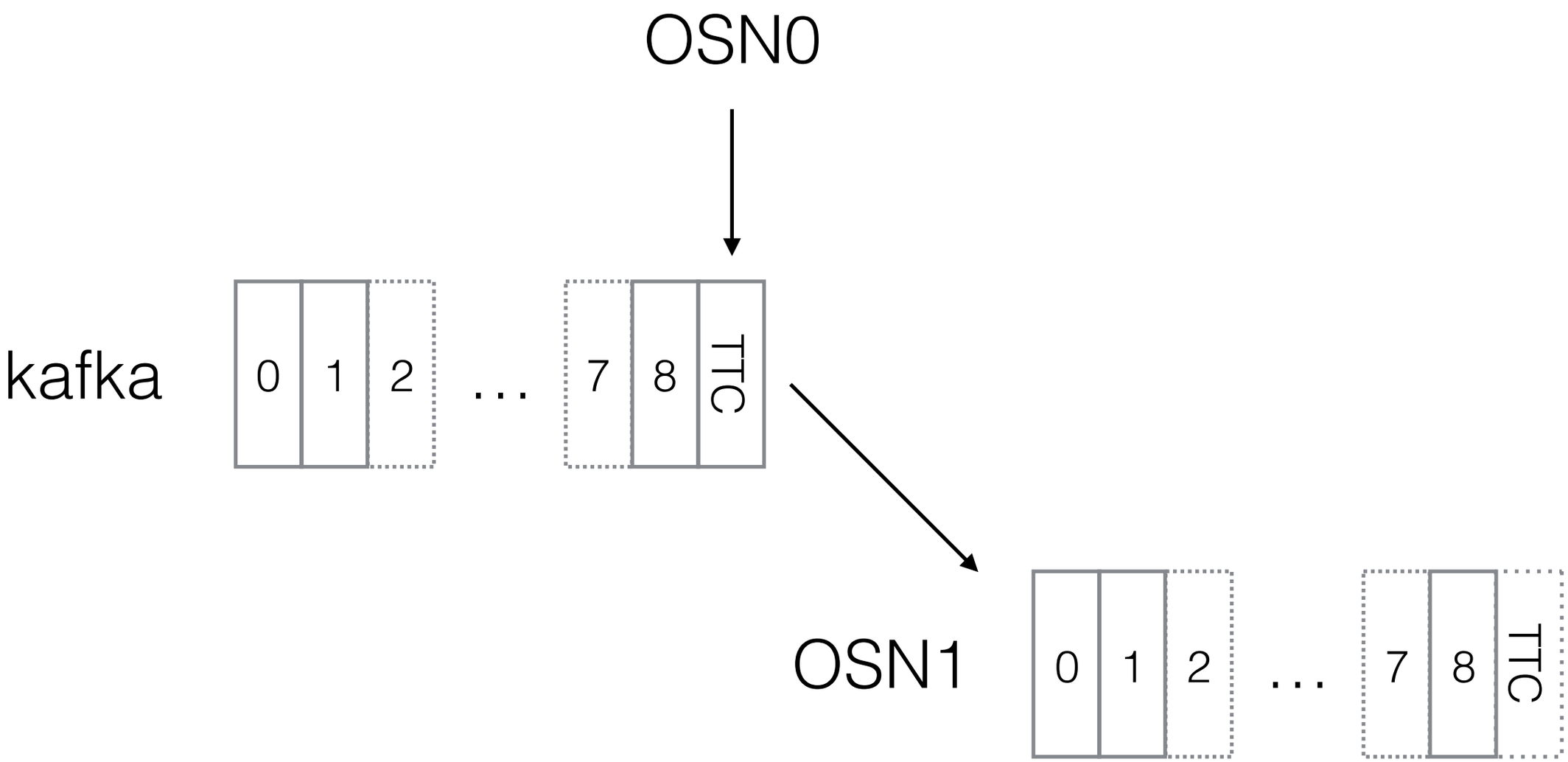
排序节点的运行机制

- Q：当batchsize未满足时，如何避免队列中的Tx一直处于等待?
  A：使用Time-To-Cut

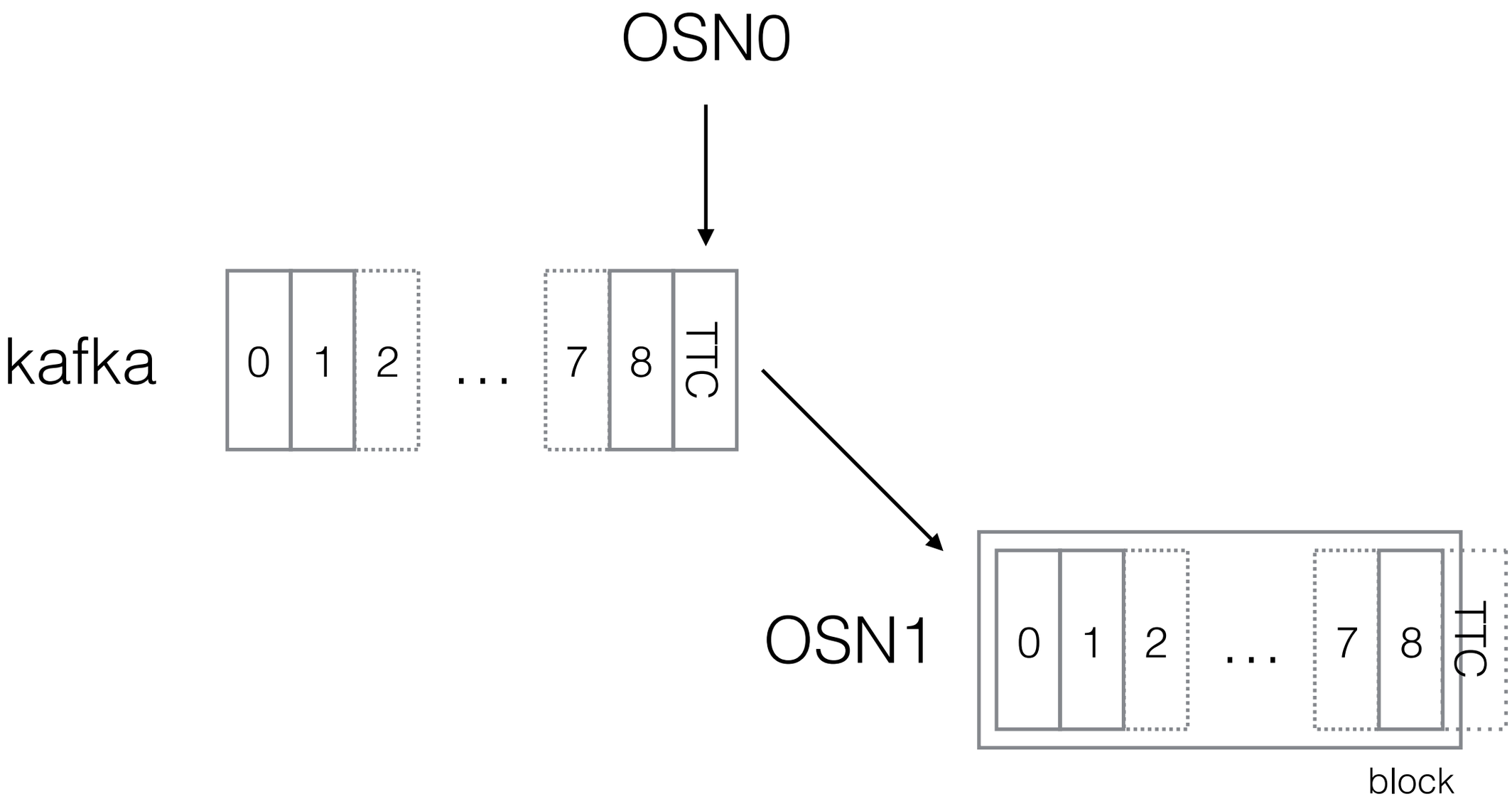
排序节点的运行机制



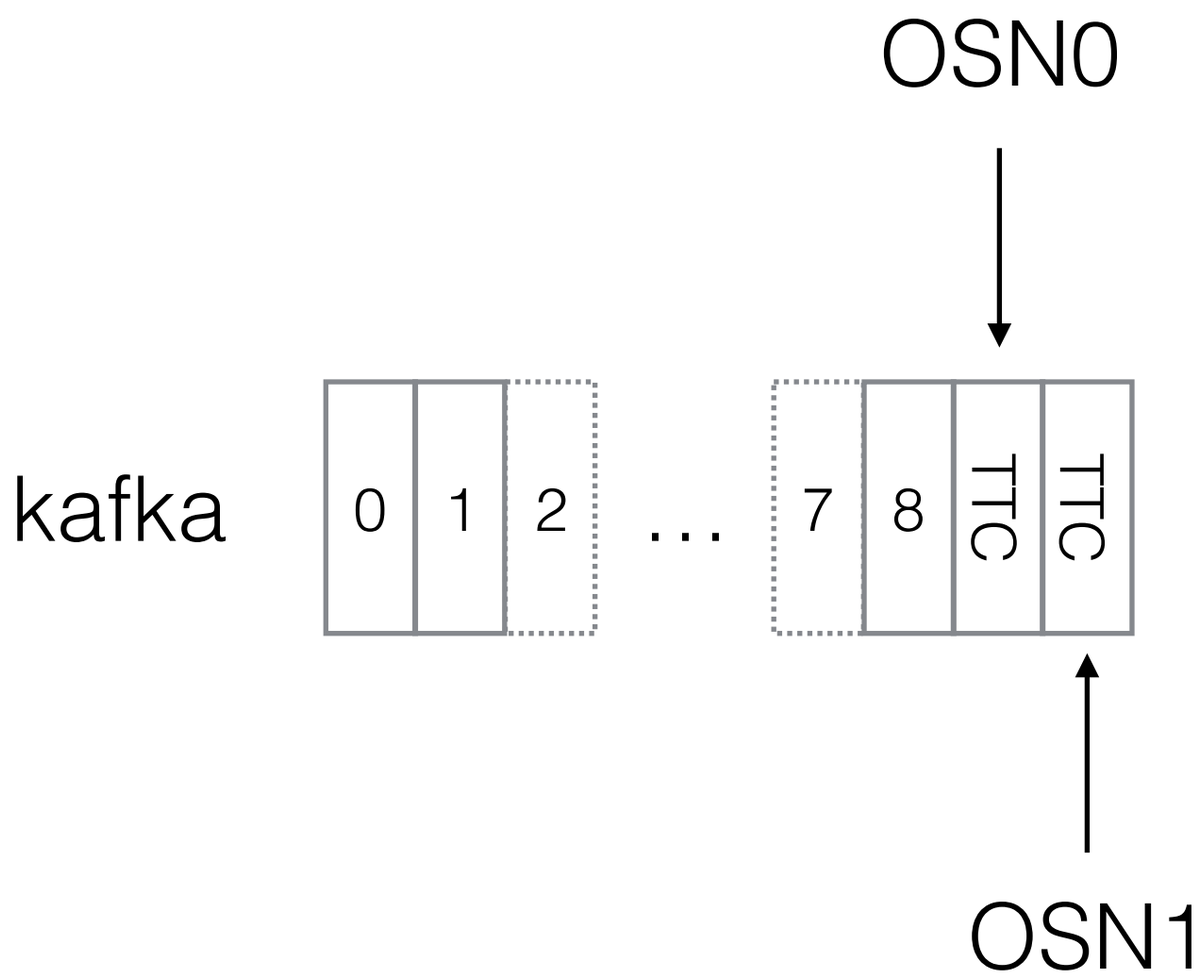
排序节点的运行机制



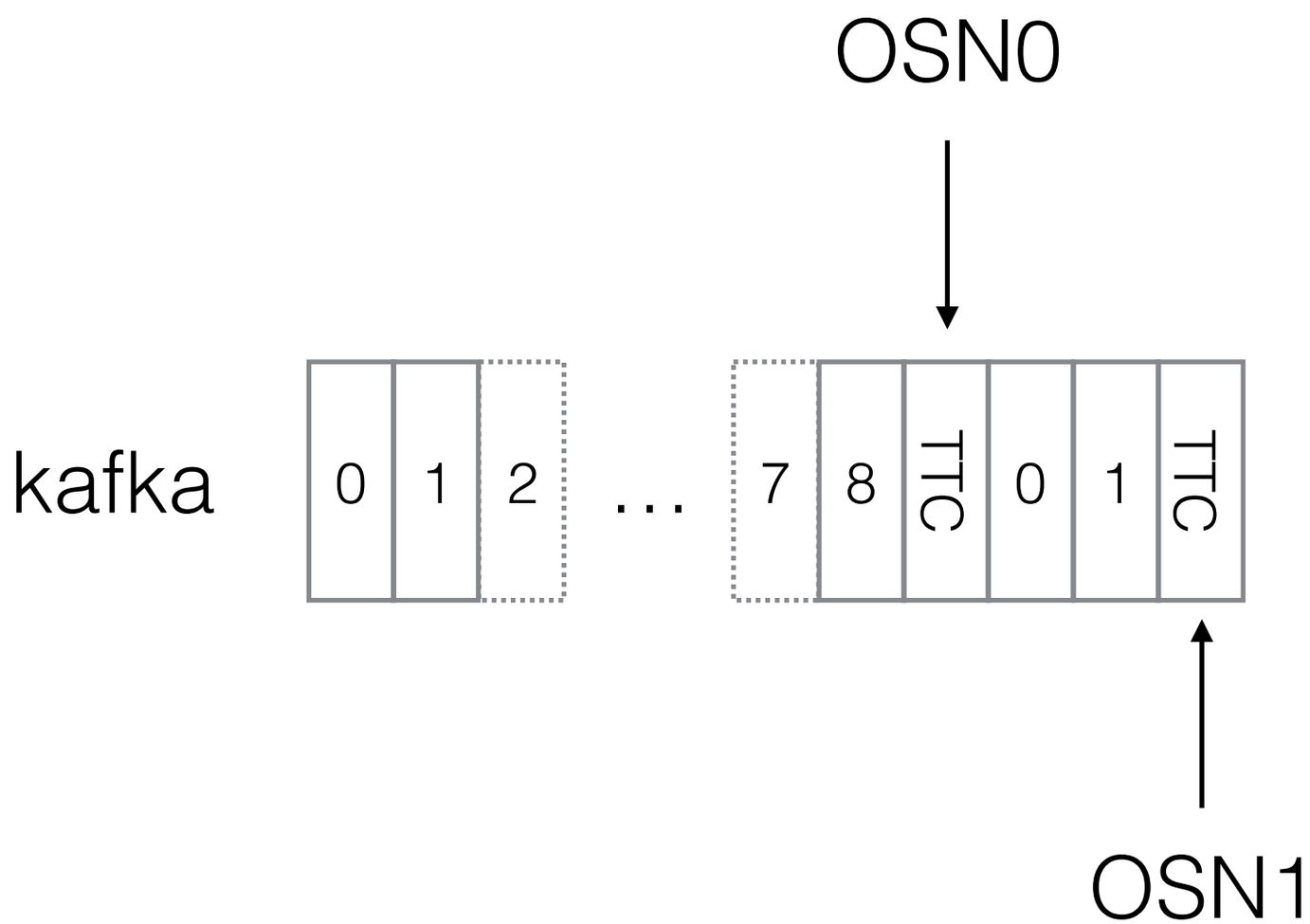
排序节点的运行机制



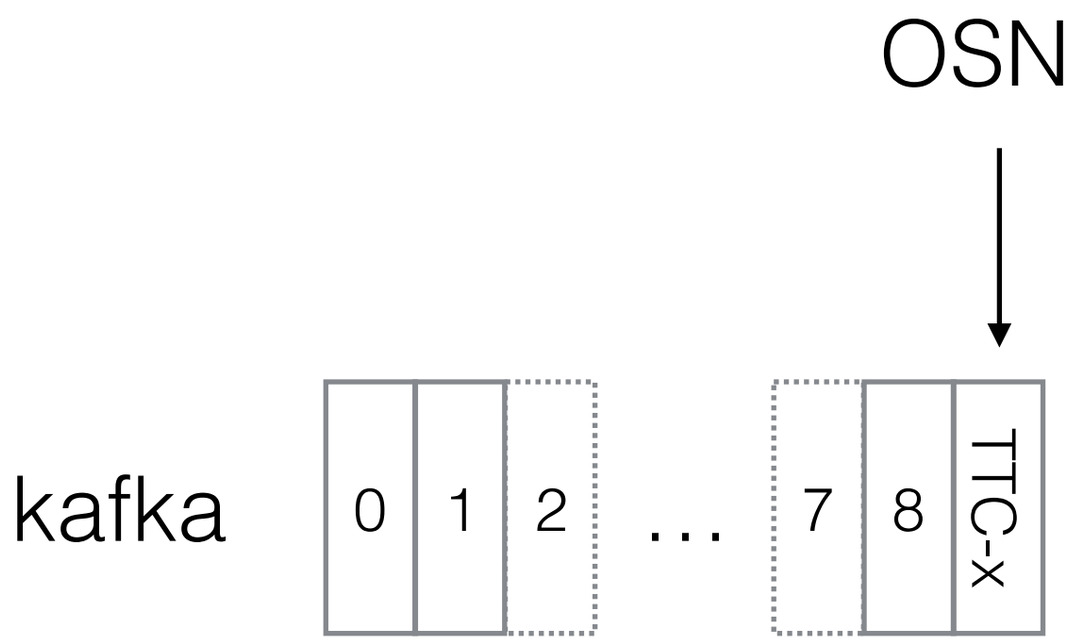
排序节点的运行机制



排序节点的运行机制

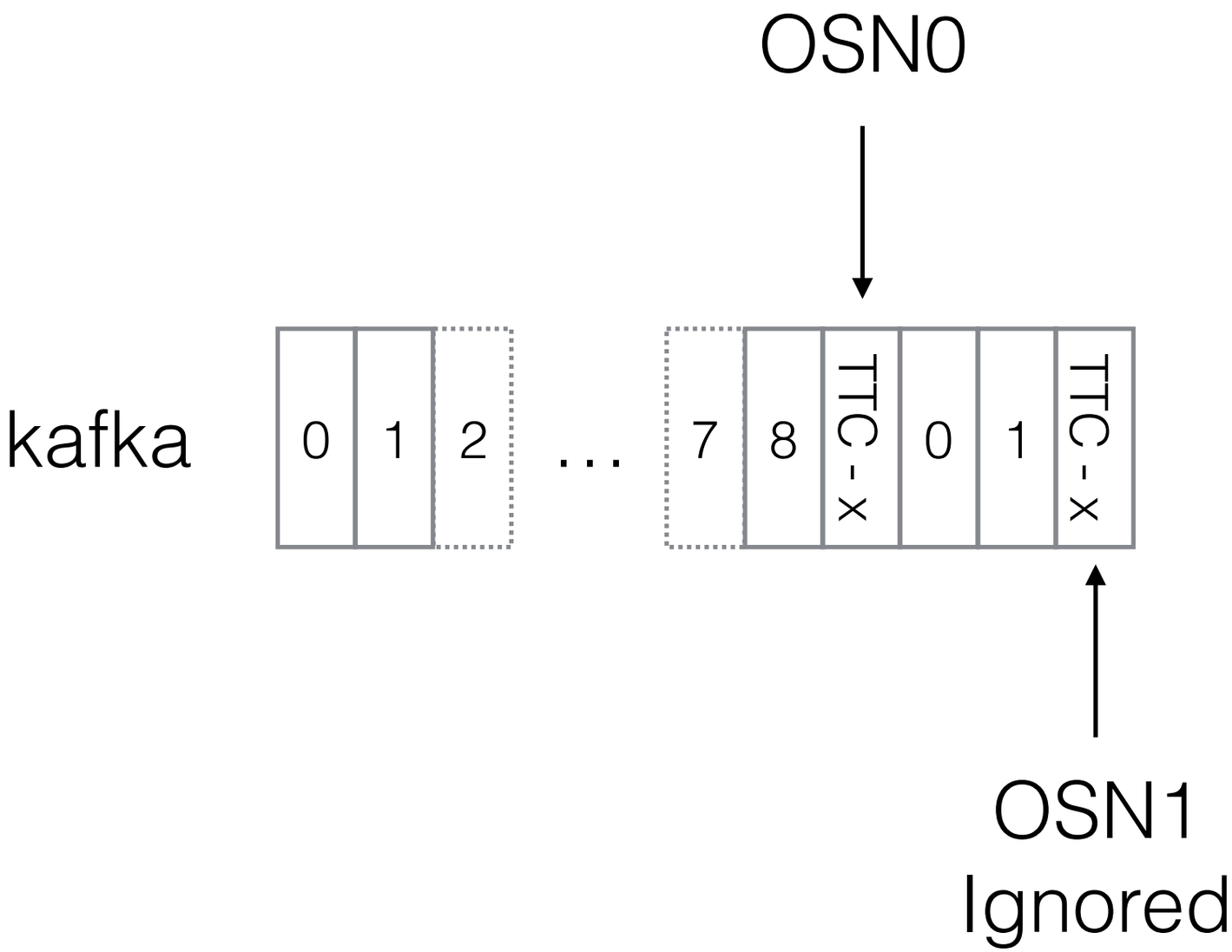


排序节点的运行机制

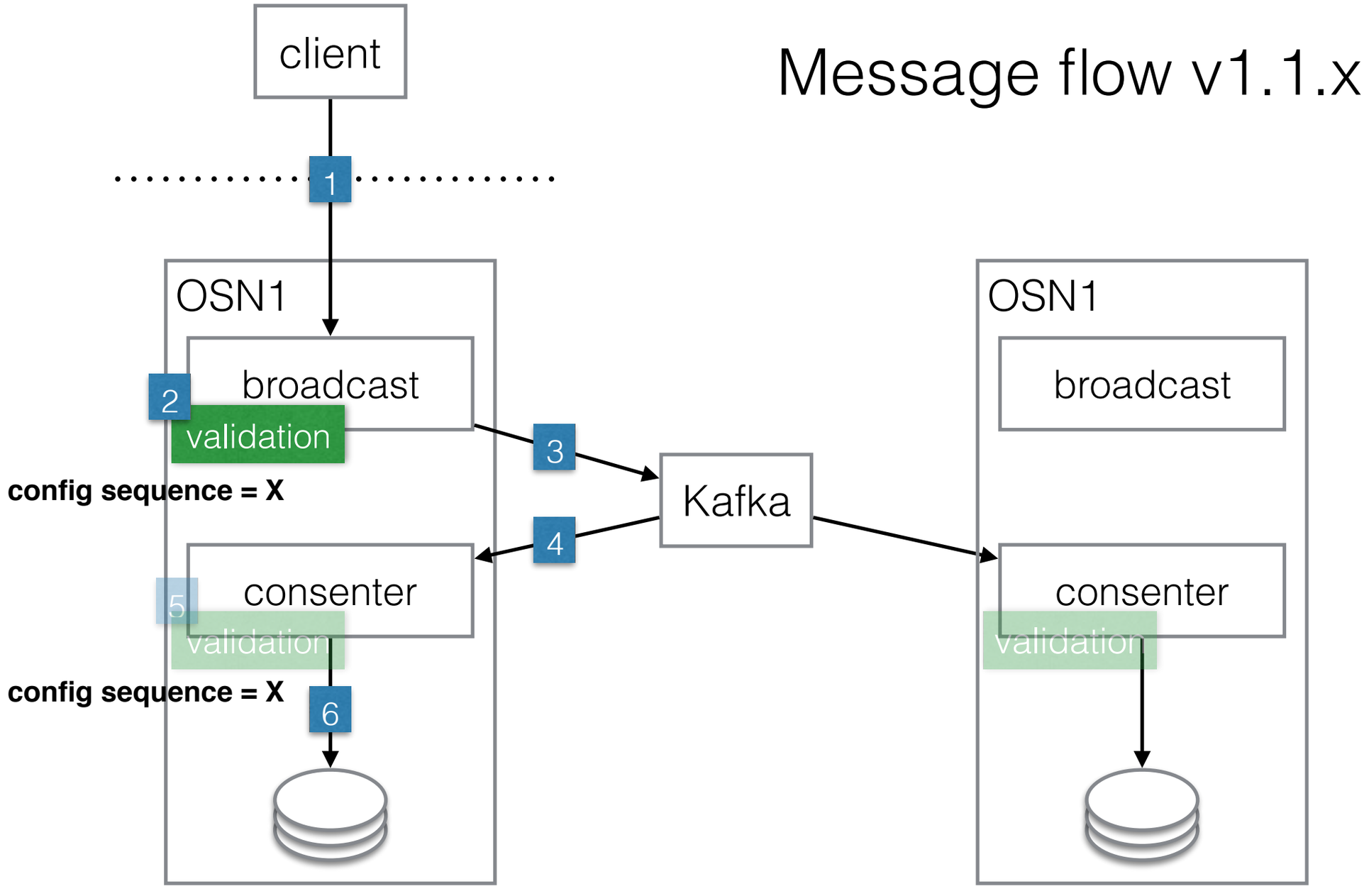


x = next block number

排序节点的运行机制



排序节点的运行机制



Message flow v1.1.x

排序节点的运行机制

排序节点的运行机制

排序节点的运行机制

- 排序节点接受Tx的类型
  - Normal
  - Config

- Normal Tx包含ProposalResponse以及其他

- Config Tx用于创建或配置channel。Config Tx单独成块

- 排序节点在Fabric中的作用
- 排序节点的运行机制

- 排序节点的配置

排序节点的配置

- 常量
  使用spf13/viper库进行配置的解析
  ORDERER_*, orderer.yaml
- 变量
  configtx.yaml -> genesis block

Fabric源码入门

- 一个典型的Fabric工作流
  - create channel
  - join channel
  - install chaincode
  - instantiate chaincode
  - chaincode query/invoke

- Orderer解析
  - 通过broadcast接口提交一笔交易
  - 通过deliver接口获得一个区块

- Fabric社区贡献流程

社区贡献

- Gerrit + JIRA + Rocket.Chat
- TDD
- git push/git review
- add maintainers for that module as reviewers
- wait for a green build, re-trigger build if necessary