IBM

Open by design™

IBM开源技术微讲堂

区块链和HyperLedger系列

第六讲

Hyperledger中的共识机制

扫码入群，与讲师互动

http://ibm.biz/opentech-ma

“区块链和HyperLedger”系列公开课

- 每周四晚8点档
  1. 区块链商用之道
  2. HyperLedger项目与社区概览
  3. HyperLedger Fabric架构解读
  4. ChainCode实战
  5. HyperLedger 中的共享账本
  6. **HyperLedger中的共识机制**
  7. HyperLedger中的隐私与安全
  8. HyperLedger应用案例赏析

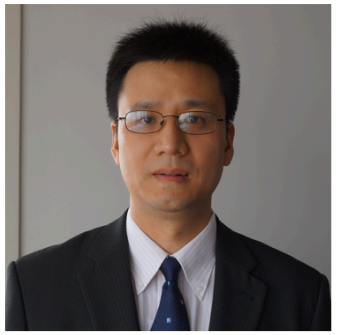
Q&A环节

扫码入群，与讲师互动

课程Wiki：http://ibm.biz/opentech-ma
往期视频：http://list.youku.com/albumlist/show?id=49106065

讲师介绍

夏勇博士



个人概述

- IBM 思想领袖级顾问
- IBM大中华区GBS顾问中心负责人
- IBM大中华区GBS区块链中心负责人
- **IBM全球技术研究院院士** (IBM AoT Member)

学术界

- **国际需求工程委员会理事**（www.ireb.org）(全球共五位理事中，唯一的华人理事）
- **复旦大学客座教授**
- **上海交通大学 企业博士生导师**
- **会议共同主席 / 专家评审委员会成员 / 工业论坛主席：**
  - 2017年: The 29th International Conference on Advanced Information Systems Engineering
  - 2016年: **The 24th IEEE International Requirements Engineering Conferenc**e
  - 2015 和 2016 年： The Asia Pacific Requirements Engineering Symposium,
  - 2015年: 第六届CSTQB国际软件测试高峰论坛
  - 2015年: 互联网环境下的需求工程研讨会
- 多年来在国际一流期刊和会议发表有影响力的论文近20篇
- **瑞士苏黎世大学经济学院计算机系博士**

工业界

- IBM全球技术研究院院士(IBM AoT Member, IBM GBS大中华区仅有的两名院士之一)。现担任IBM GBS顾问中心和区块链中心负责人，主要负责IBM在Fintech/区块链和认知计算方面业务和技术发展。同时，作为业务和技术总架构师和思想领袖级顾问， 为摩根大通 (J.P. Morgan Chase)， 新加坡星展银行(DBS)及中国工商银行和中国银行等客户提供最高管理层咨询。

  之前作为IBM金融解决方案负责人，负责的团队为国际和国内一线银行提供解决方案。客户包括汇丰银行，瑞士联合银行，新澳银行，中国银行，中国建设银行，美亚保险，Suncorp等20多家金融机构。领域包括核心银行，投资银行和财富管理等等。所领导团队每年提供方案的合约总值，最高时接近上亿美元。
- 加入IBM前，在瑞士苏黎世的瑞士信贷银行总部 (Credit Suisse — https://www.credit-suisse.com/) “全球银行核心部” ，担任总架构师。成功完成投资银行和商业银行的大型项目，包括 “结构化产品业务流程” ， “证券产品（股票，债券和金融衍生品）的目标系统规划” ， “信贷评估系统” ， “欧洲重点市场的税务系统” ， “银行合规系统架构” 等。具体工作包括：(1) 定义应用系统技术路线图，维护和完善公司内部的开发规范，设计流程以及系统构建标准; (2) 设计，评估和决定战略性的解决方案; (3) 企业架构和流程的设计，建模和优化; (4) 金融系统(交易)算法设计和量化分析; (5) 作为技术总负责人, 负责管理和协调大型项目以及与银行最高级管理层的沟通。
- 再之前，在丹麦电讯瑞士分公司(北欧最大电信公司之一)，担任高级信息专家

工业证书

- 特许金融分析师 – Chartered Financial Analyst (CFA) 三级候选人

议程

- Blockchain - Hyperledger
- Distributed system and its related issues
- Consensus
  - Permissioned (voting) consensus
  - BFT in details
  - Constraints related to (Vanilla) BFT
  - Consensus mechanism in Hyperledger v1.0
- Summary
- Q&A

Blockchain - Hyperledger

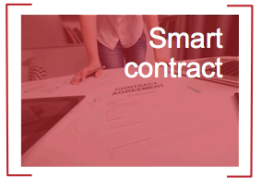
BLOCKCHAIN EXPLAINED

What?

For the first time in over a thousand years we have an opportunity to change the mechanisms of recording value exchange…

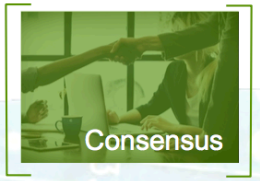
…through Blockchain

Append-only distributed system of record shared across business network



Business terms embedded in transaction database & executed with transactions

Ensuring appropriate visibility; transactions are secure, authenticated & verifiable



All parties agree to network verified transaction

Why?

Saves time

Transaction time from days to near instantaneous

Removes cost

Overheads and cost intermediaries

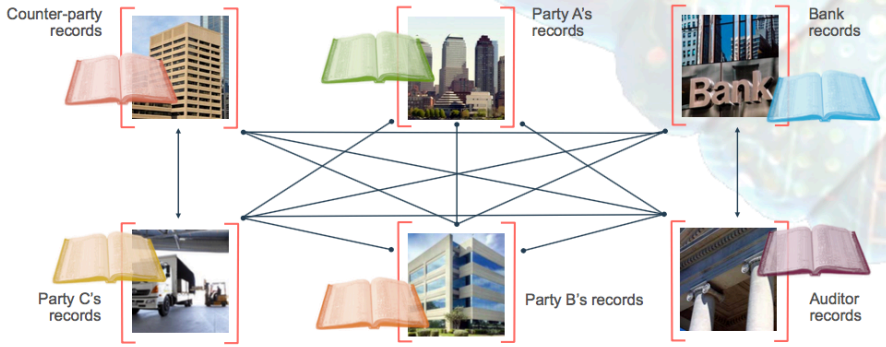
Reduces risk

Tampering, fraud & cyber crime

How?

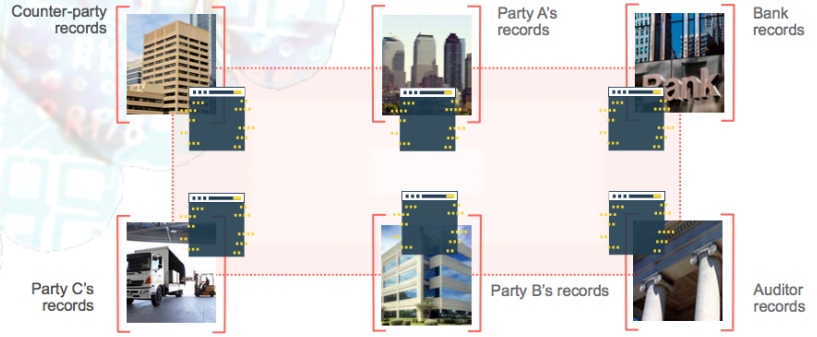
Engagement model

Problem …



… Inefficient, expensive, vulnerable

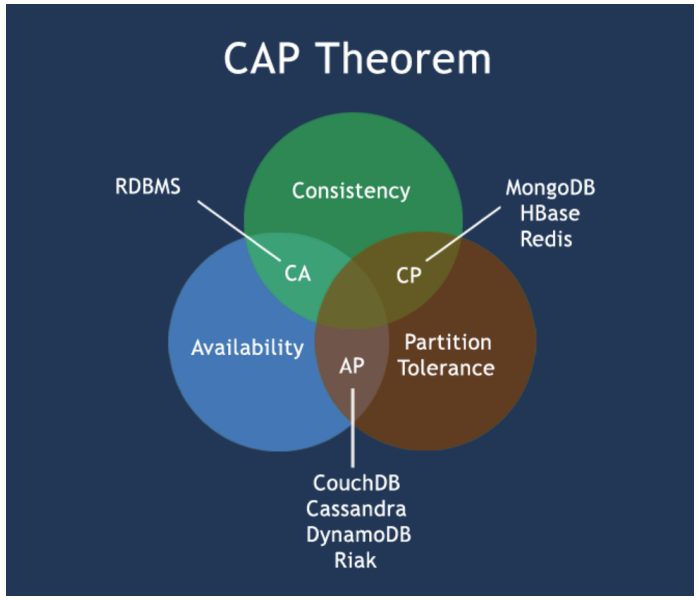
Solution …



… Consensus, provenance, immutability, finality

Distributed Systems

- CAPM

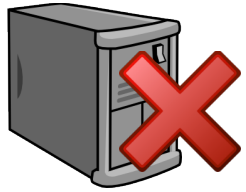


- Temporal Orders
  - Synchronization
  - …
- …

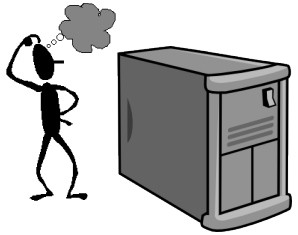
Permissioned (voting) Consensus

- Classical Distributed Computing protocols (since ‘80s)
  - Voting based
  - Consensus despite **machine** faults and (temporary) **network** partitions

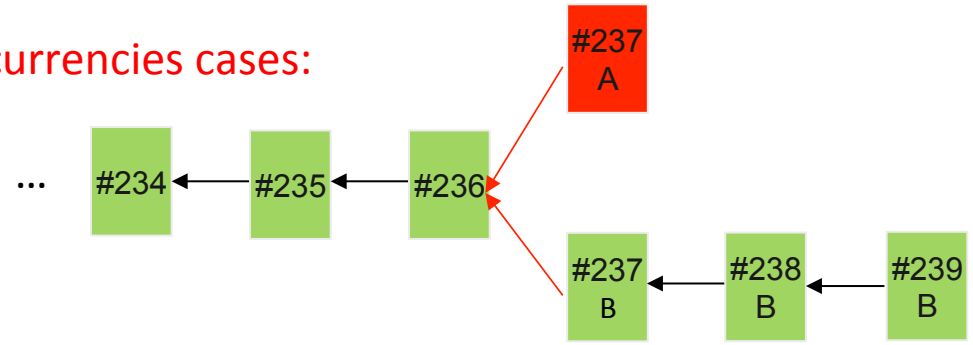
What machine faults?



- Crash faults (CFT): A machine simply stops execution and halts
  - Paxos, RAFT, Zookeeper AB,…
- Non-crash (a.k.a. Byzantine) faults (BFT)

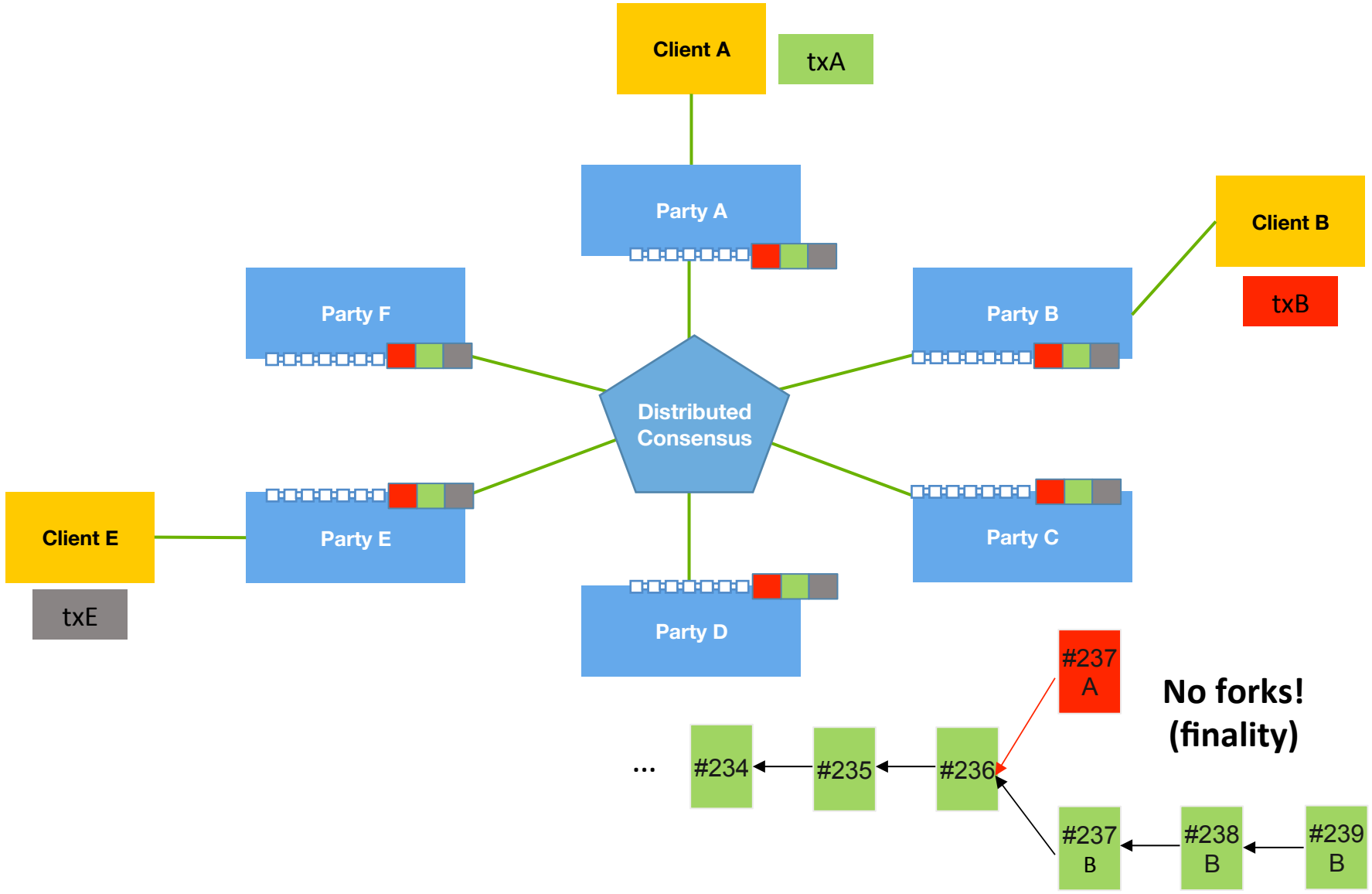


Some cryptocurrencies cases:



No forks!

Growing Permissioned Blockchains



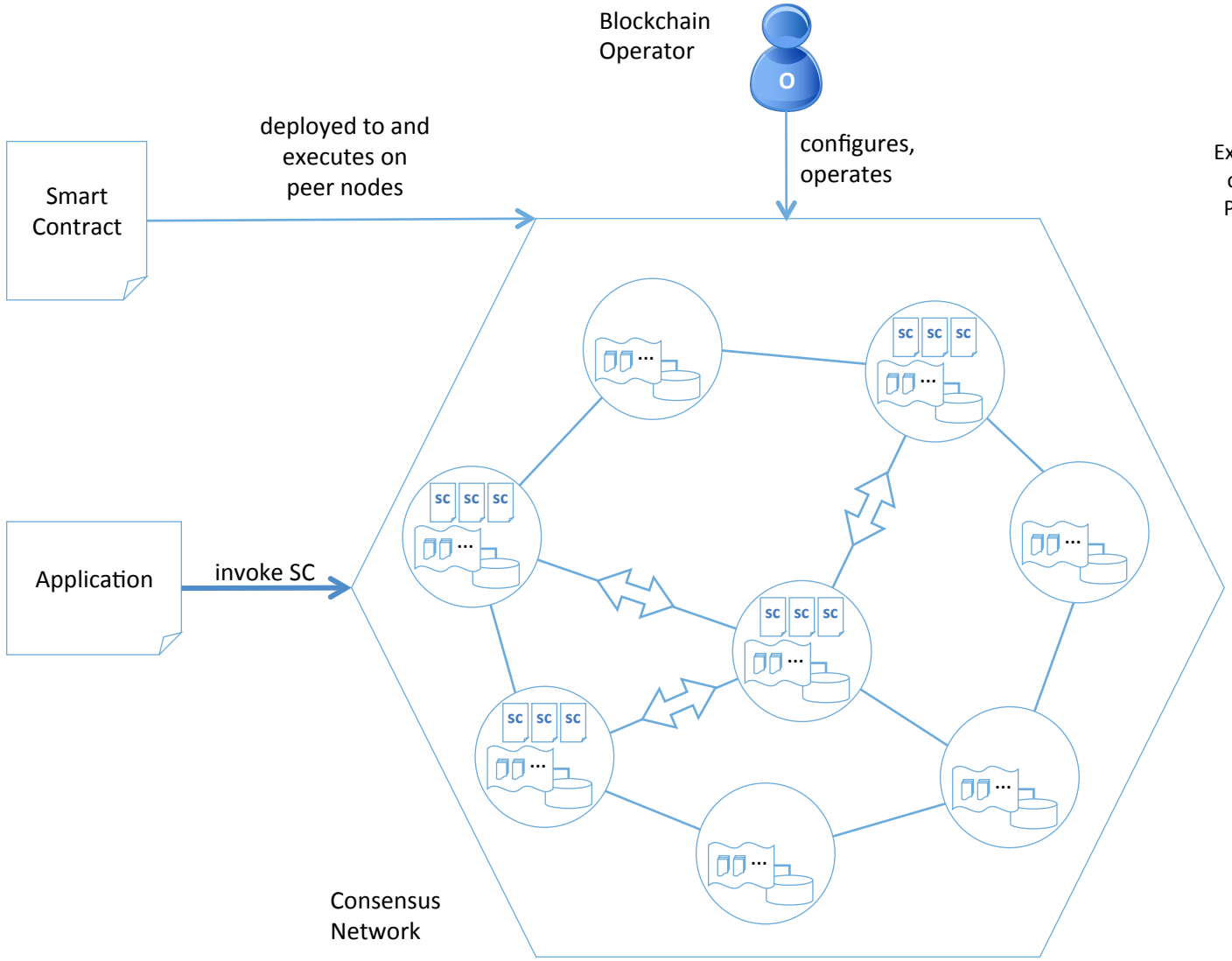
Permissioned consensus guarantees

- (Vanilla) BFT: Up to n/3 Parties can be malicious, not follow protocol
- CFT – Up to n/2 Parties can fail by crashing

- New models – XFT [OSDI 2016]
  - Consensus with up to n/2 corrupt, Byzantine Parties (with certain assumptions on network partitions)

Fault model	CFT	XFT	BFT
Number of Nodes	2f+1	2f+1	3f+1
Tolerating Byzantine Nodes	no	yes	yes
Performance	Good	Practically as good as CFT	Worse than CFT

BFT Consensus (example of PBFT [TOCS2002])



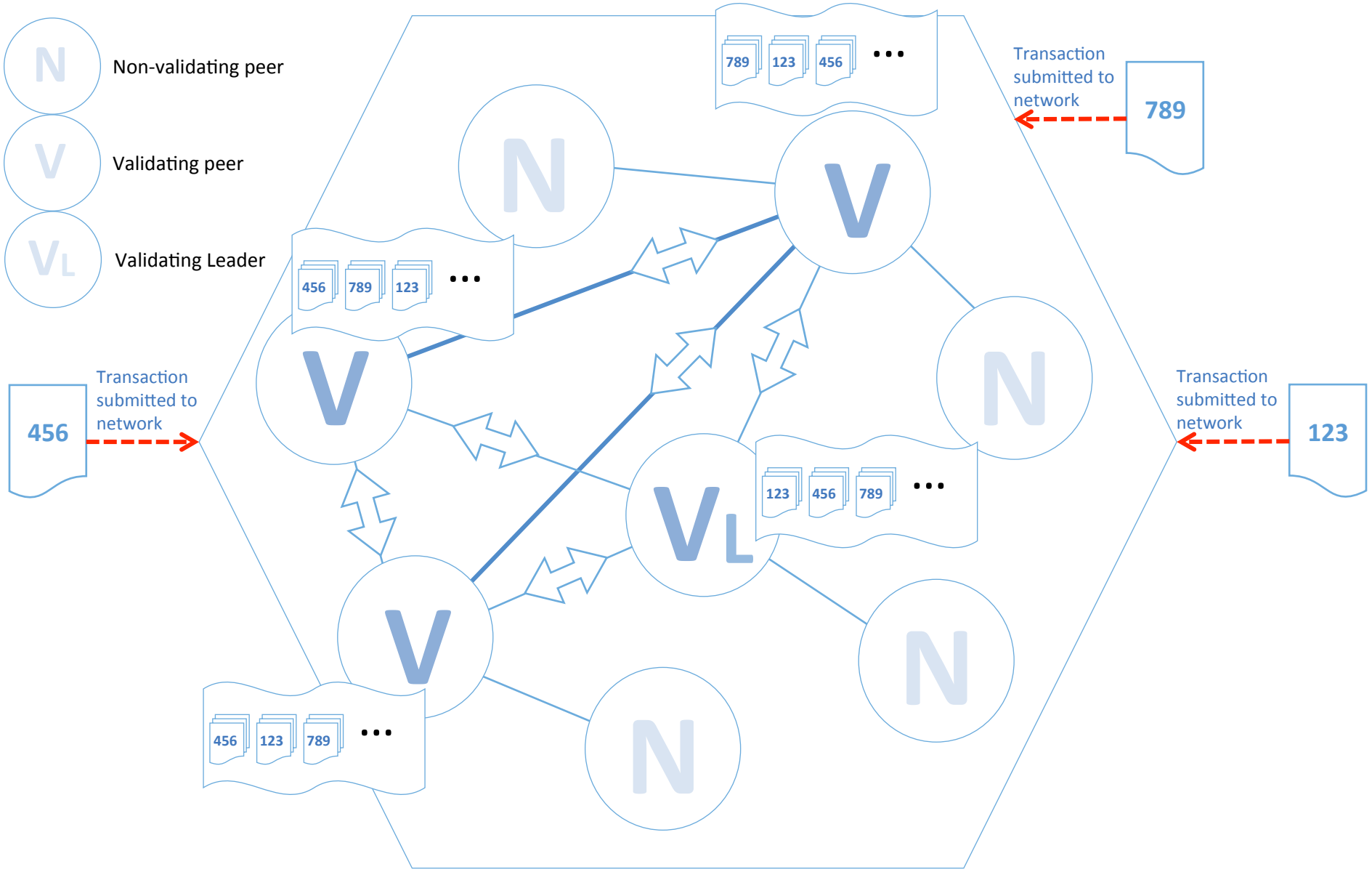
peer

Exact network structure depends on consensus mechanism. e.g. PBFT has leader, validating and non-validating peers

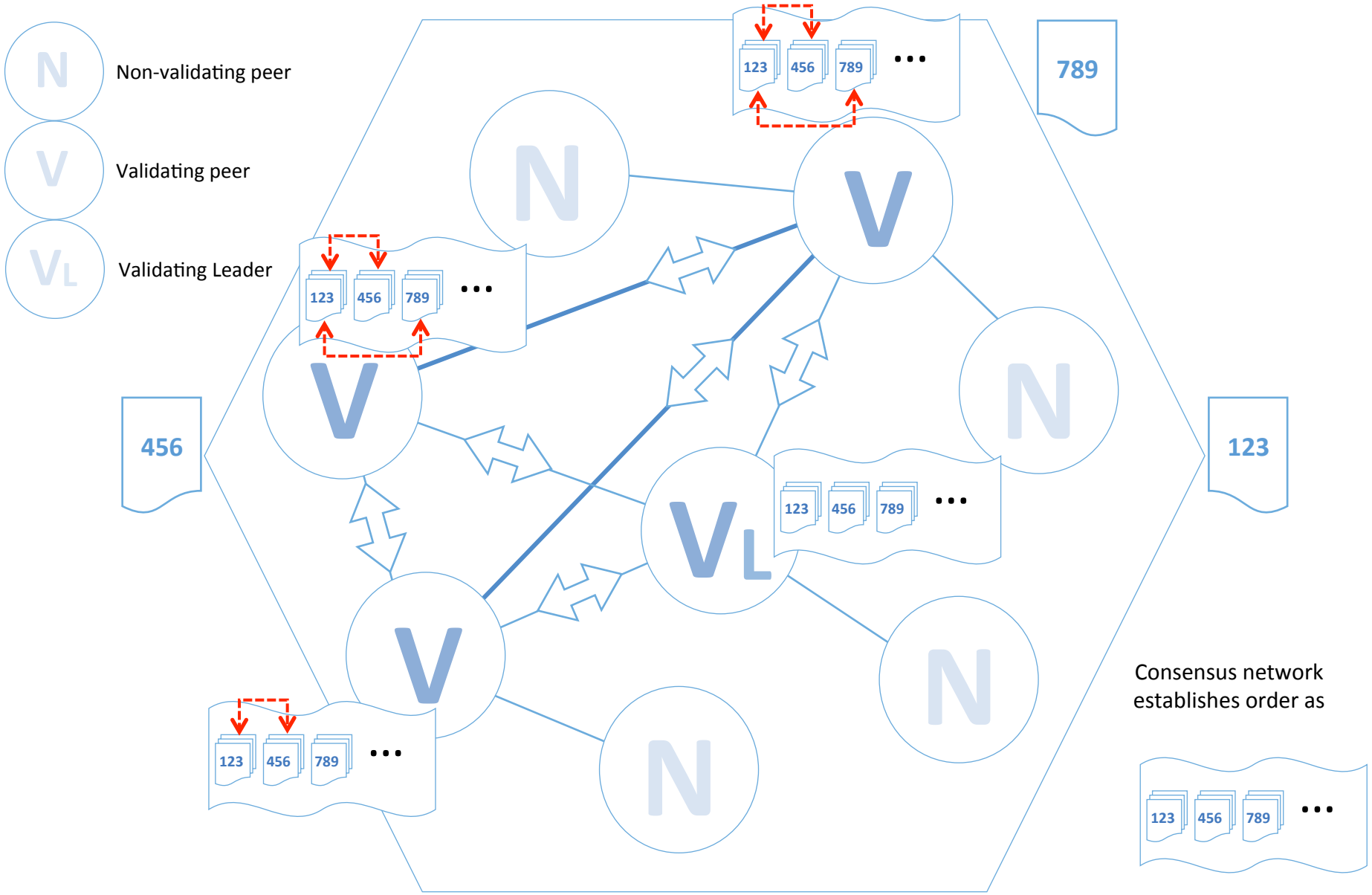
consensus msg

consensus messages flow between appropriate peers to ensure the Blockchain SC transactions are kept in order; world state is kept consistent through local transaction replay

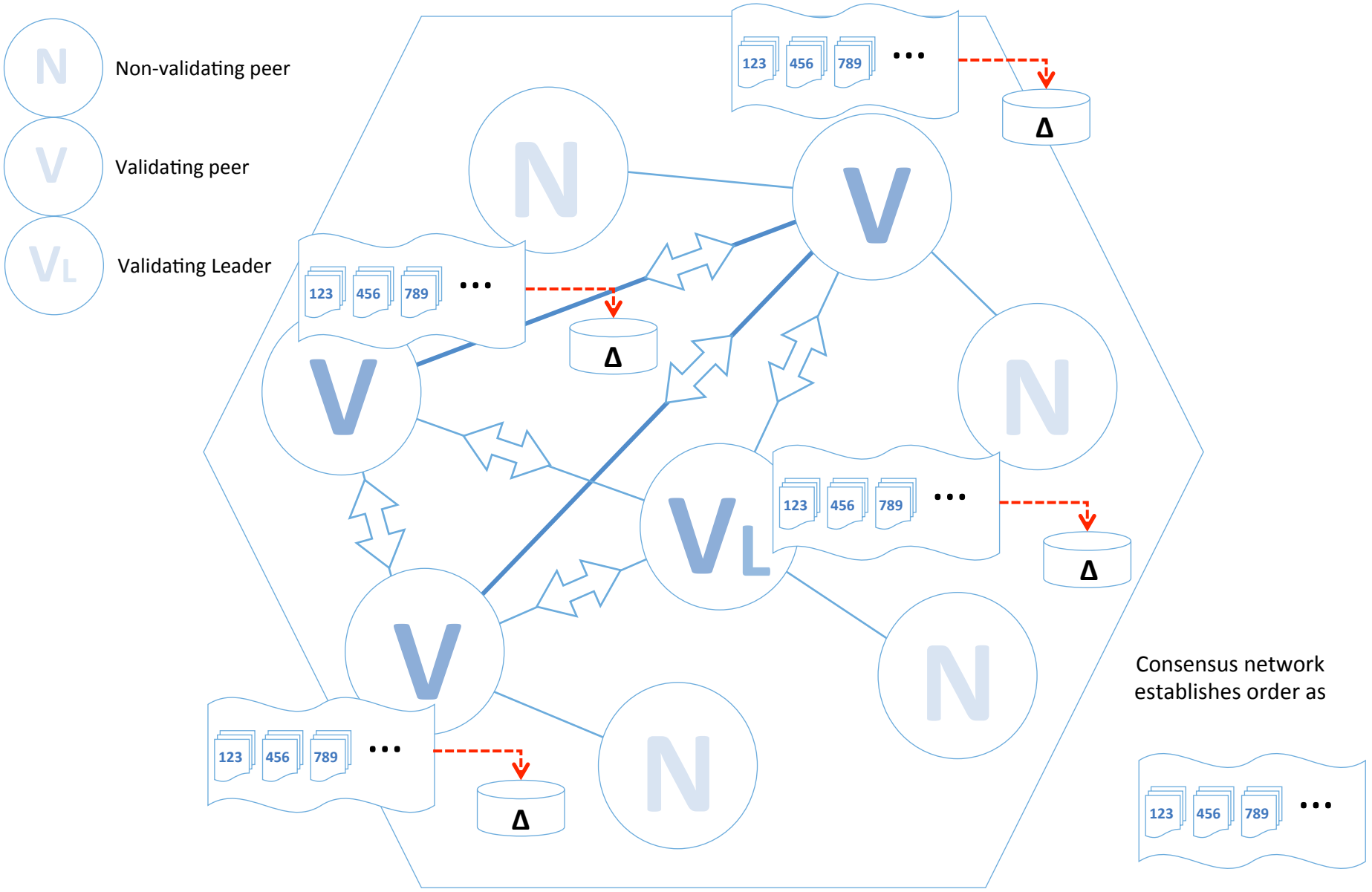
How a PBFT Network Works (1/4) – Submission



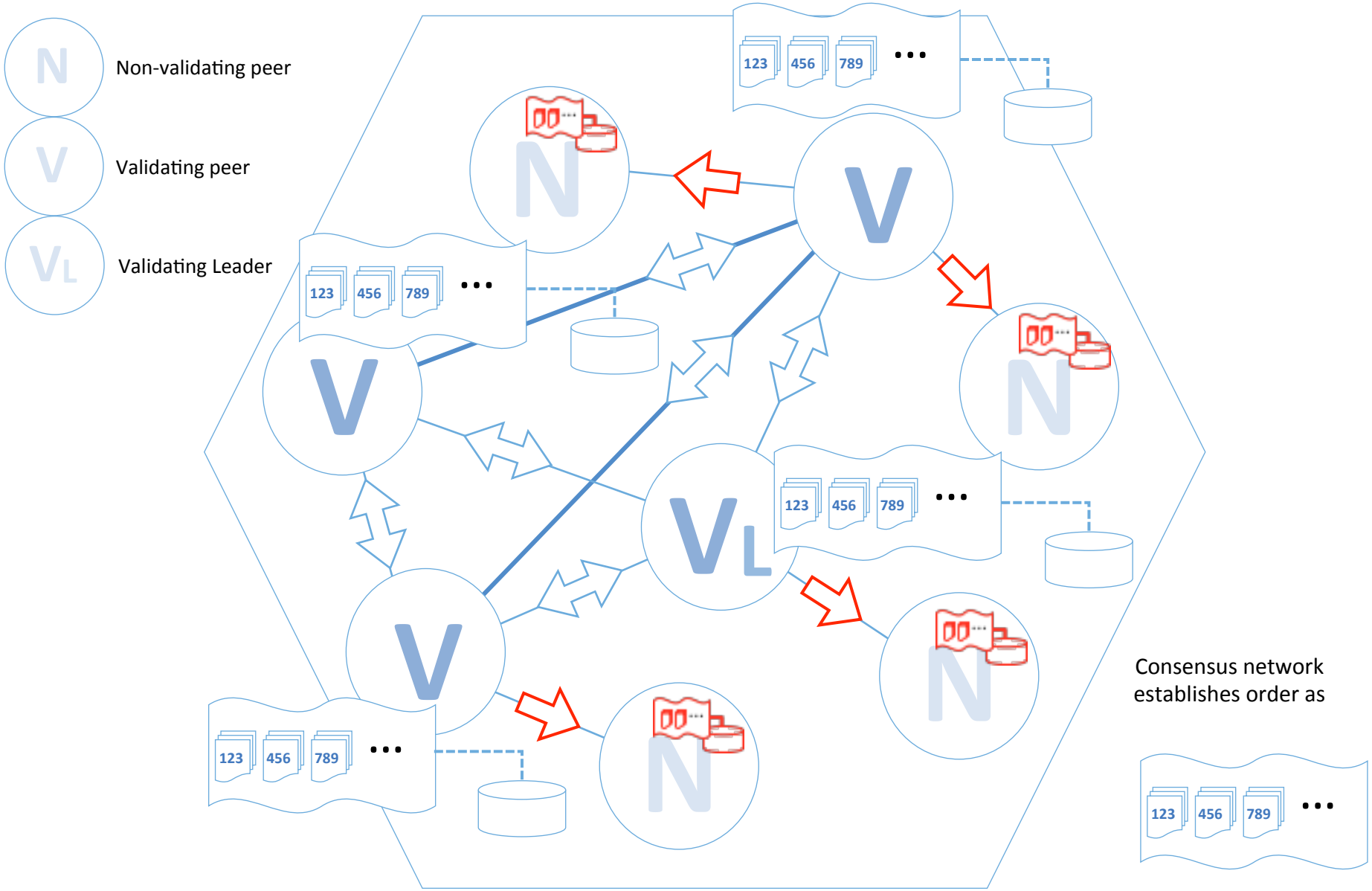
How a PBFT Network Works (2/4) – Ordering



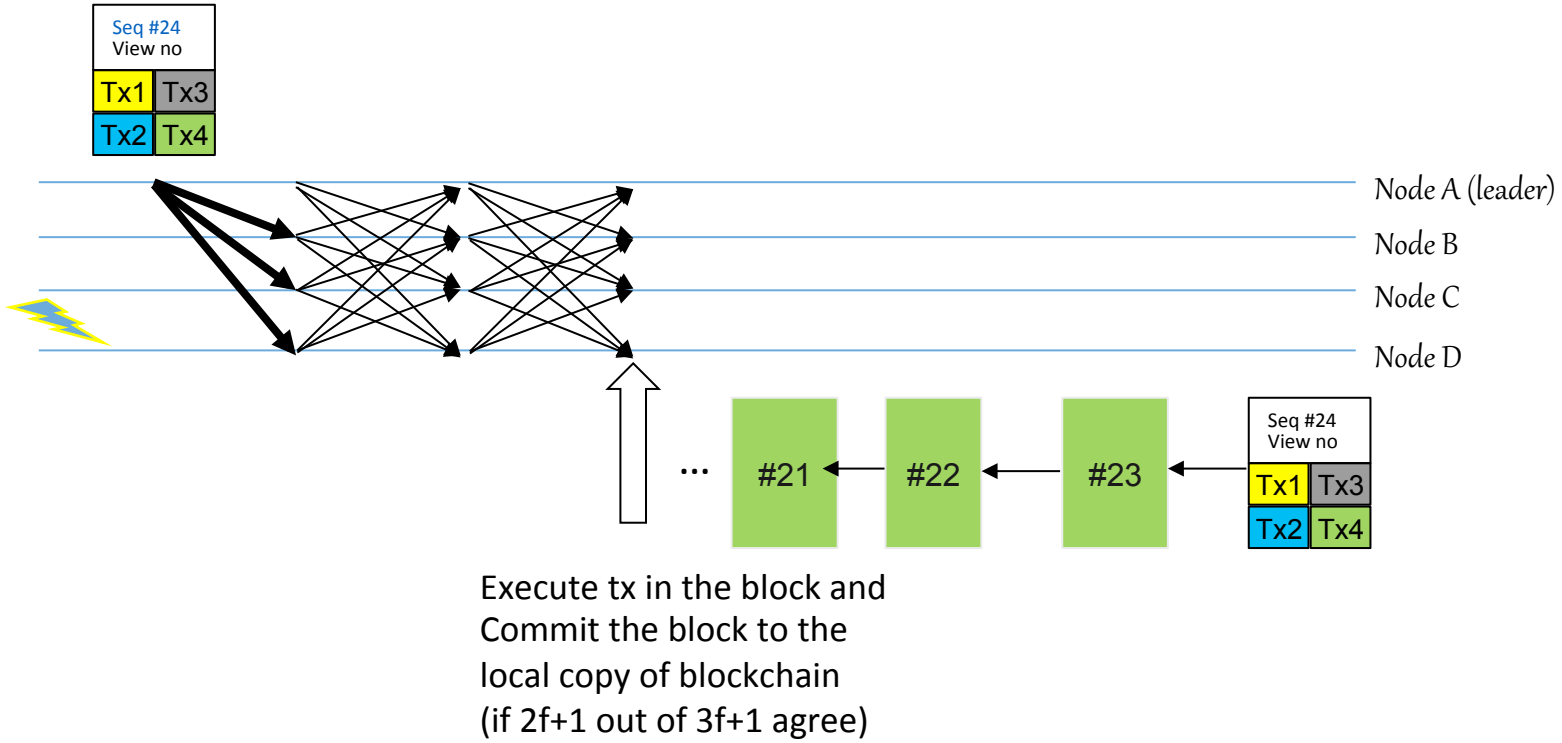
How a PBFT Network Works (3/4) – Execution



How a PBFT Network Works (4/4) – Distribution



BFT Consensus (example of PBFT [TOCS2002])



Execute tx in the block and
Commit the block to the
local copy of blockchain
(if 2f+1 out of 3f+1 agree)

Many other things burden the implementation (it is not simple as it might look)

- Non-deterministic value
- Leader election
- State transfer (new, slow Party)
- Reconfiguration

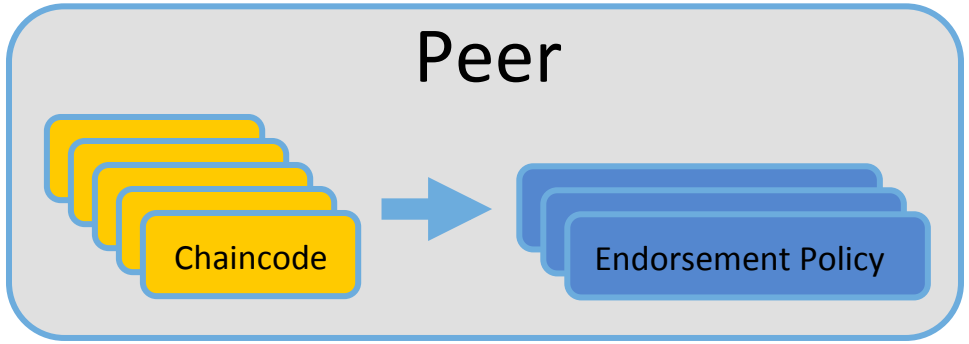
Nodes and roles

	**Peer**: Commits transactions, maintains ledger and state
	**Endorsing peer**: Specialised peer that receives a transaction proposal for endorsement, responds granting or denying endorsement
	**Ordering peer**: Approves the inclusion of transaction blocks into the ledger and communicates with peer and endorsing peer nodes

Endorsement Policies

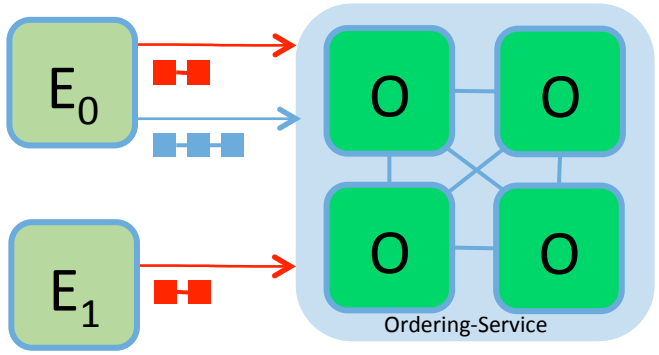
An endorsement policy describes the conditions by which a transaction can be endorsed. A transaction can only be considered valid if it has been endorsed according to the policy.

- Peers maintain a set of endorsement policies
- An endorsement policy is specified on deployment of chaincode



Channels

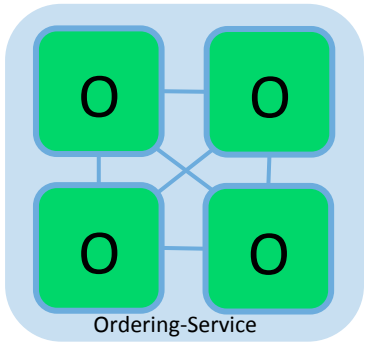
Nodes send/receive messages to the ordering-service via channels.



– Enables chaincode privacy
  • Chaincode deployed to certain nodes
– Messages partitioned into separate channels
  • Transactions stored depending on node and channel
– Nodes can connect to one or more channels

Ordering Services

The ordering service packages transactions into blocks to be delivered to peers. Communication with the service is via channels.



Different configuration options for the ordering service include:

- **SOLO**
  - Single node for development
- **Kafka / Zookeeper**
  - 1:n nodes providing Crash Fault Tolerance
  - Odd number of nodes recommended
- **SBFT**
  - 1:n nodes providing Byzantine Fault Tolerance

IBM开源技术微讲堂

区块链和HyperLedger系列

Q&A

扫码入群，与讲师互动

更多信息，请访问：http://ibm.biz/opentech-ma