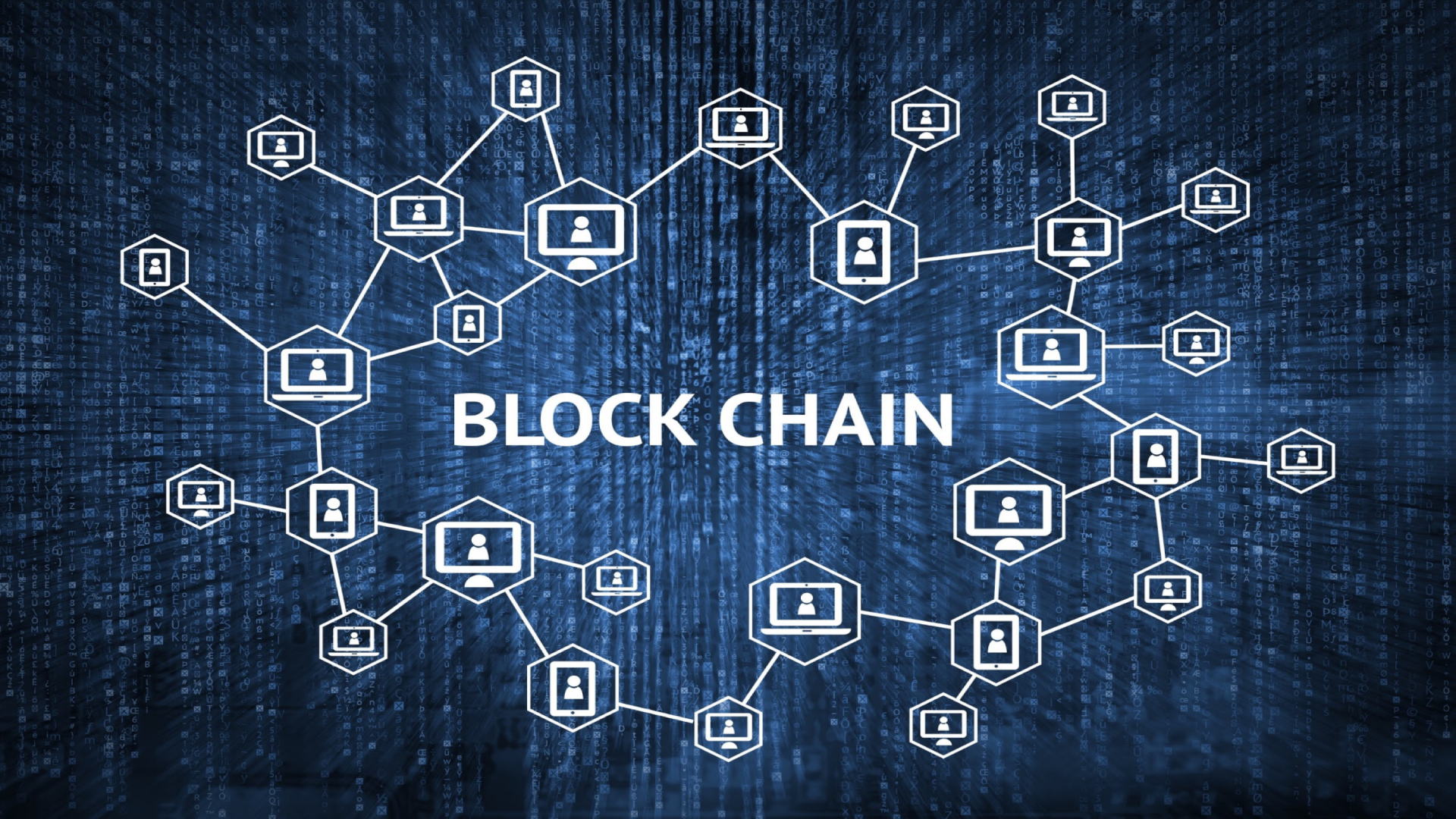
Blockchain consensus Protocols in the Wild

Tao Wang, Lihang Pan
ECS 265


BLOCK CHAIN

- Apache Kafka (CFT)
- PBFT (BFT)

Kafka Architecture

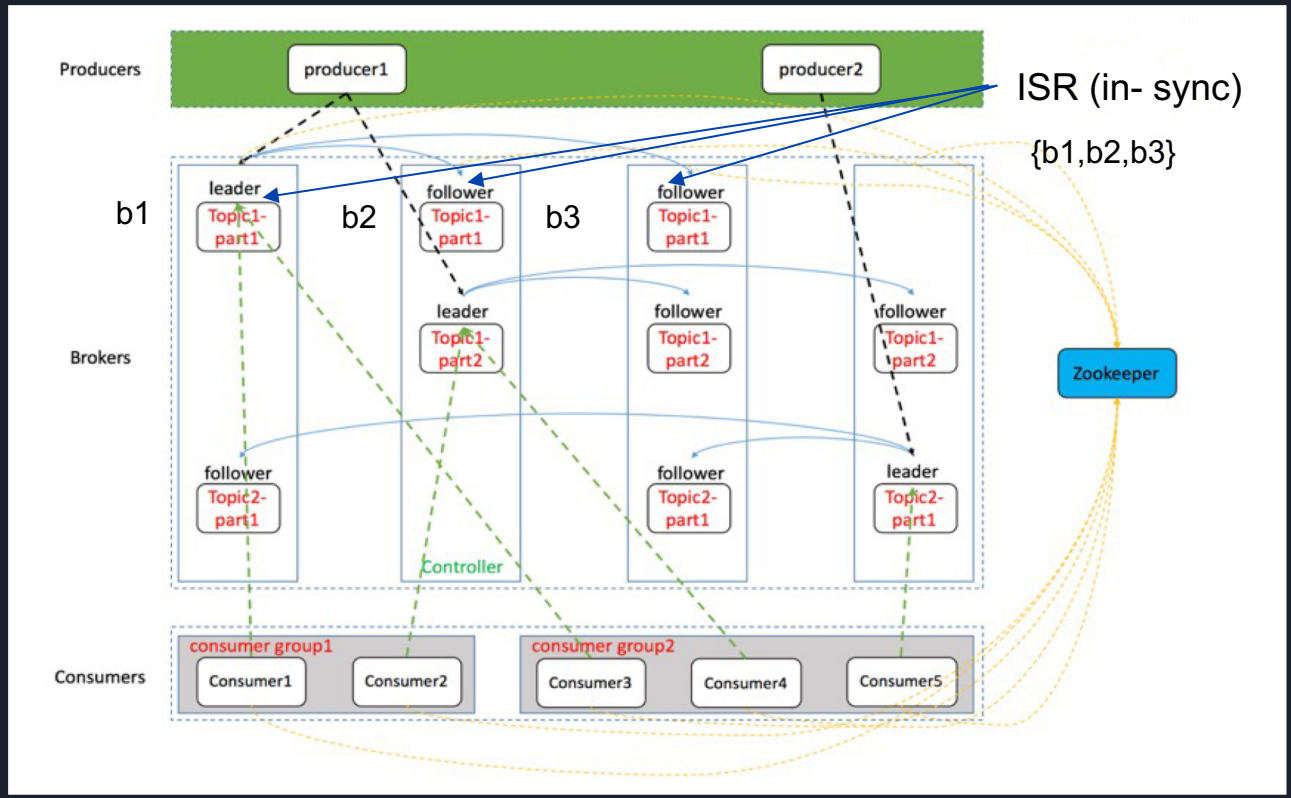


Photo credit to “Apache Kafka, Cluster Architecture”

Kafka High Watermark(HW) and Log End Offset(LEO)

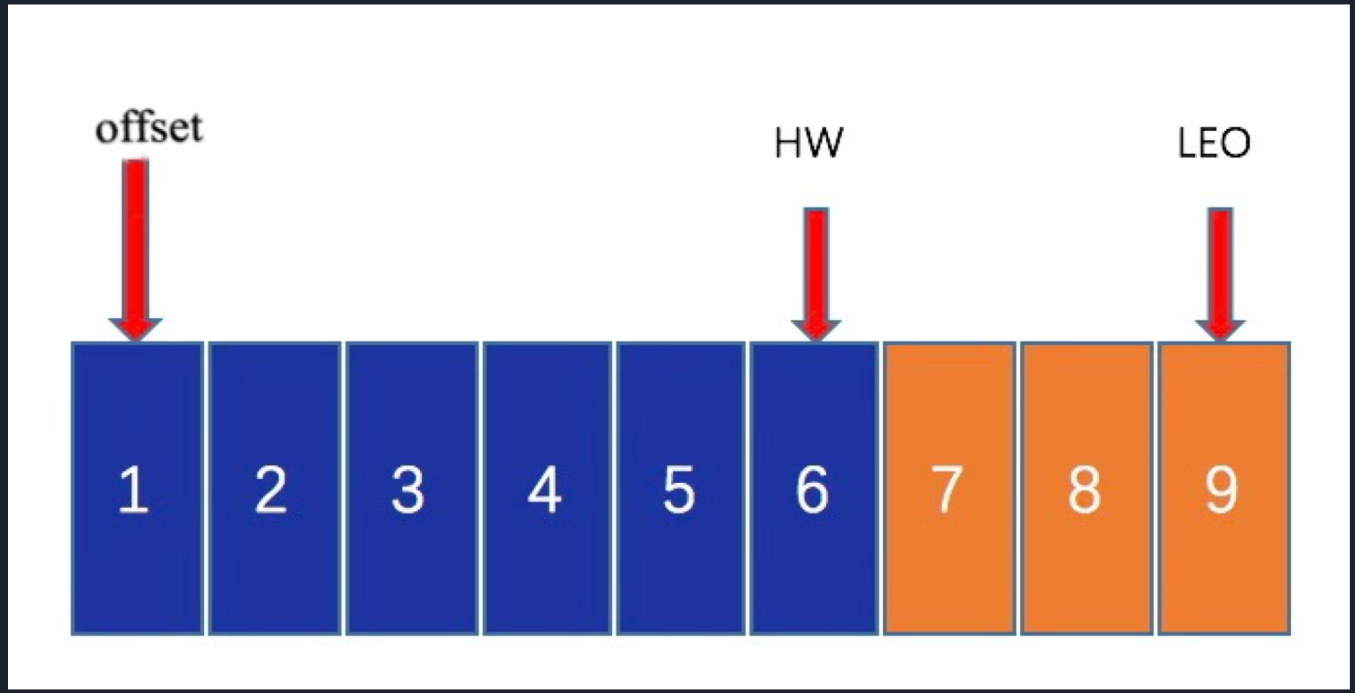


Photo credit to “In-depth Kafka Message queue principles of high-reliability”

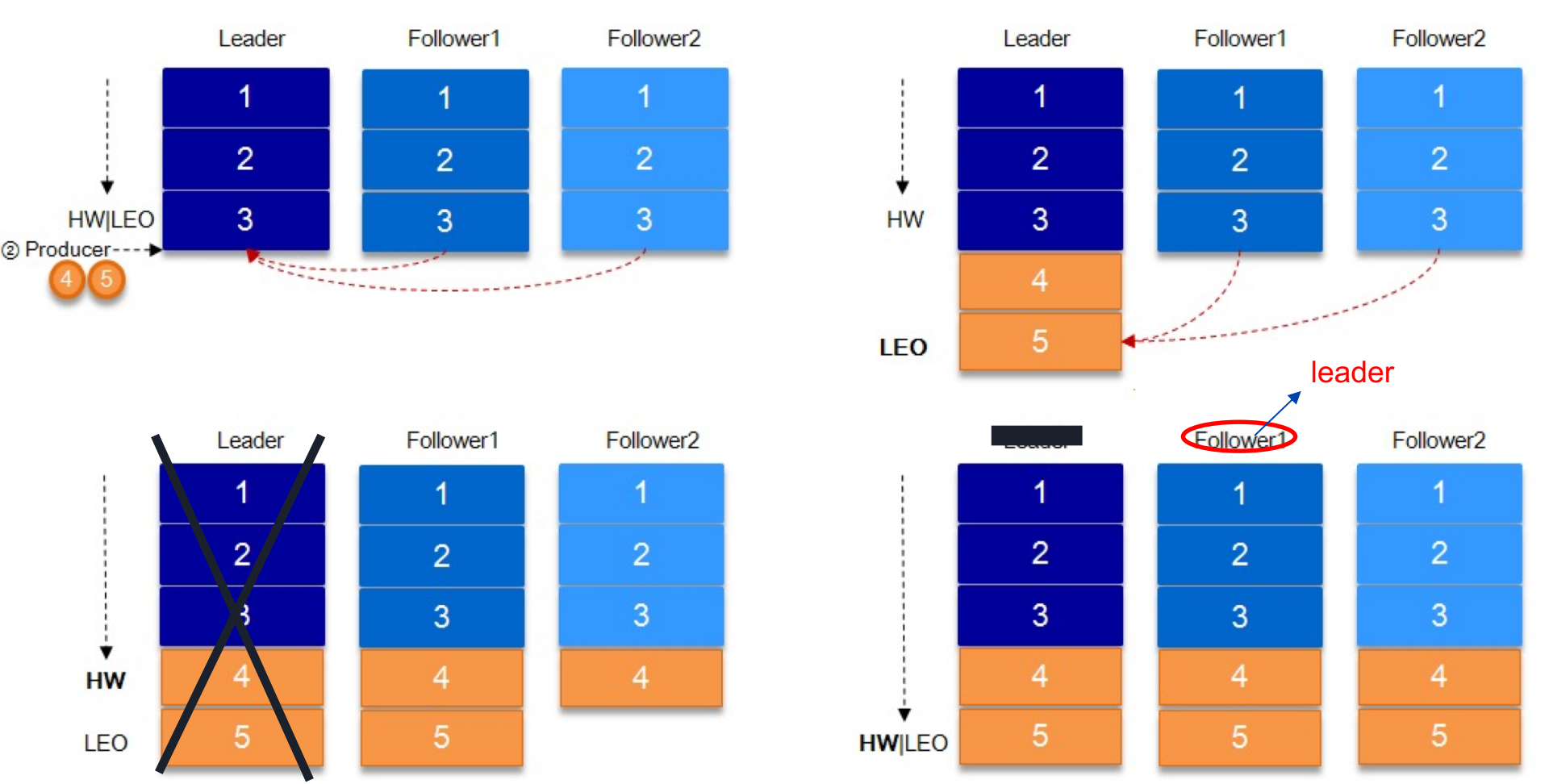
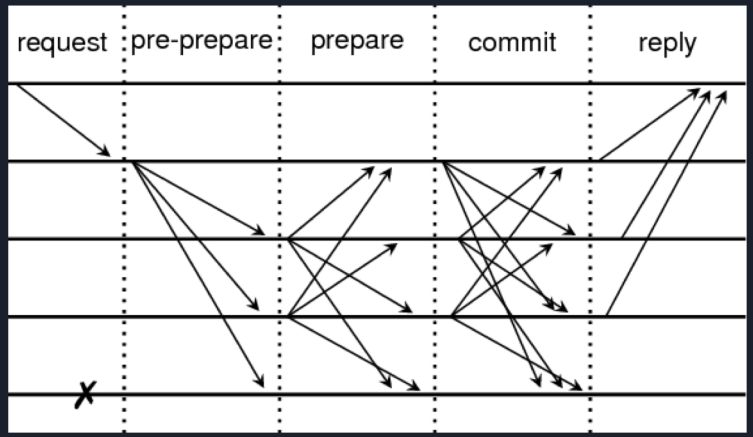


Photo credit to “In-depth Kafka Message queue principles of high-reliability”

Kafka

	Network model	Any f node crashed
Safety	Asynchrony	f< n/2
Liveness	Eventual Synchrony	f< n/2

PBFT



	Network model	Any f node subverted
Safety	Asynchrony	f< n/3
Liveness	Eventual Synchrony	f< n/3

Photo credit to “ Message pattern in PBFT”

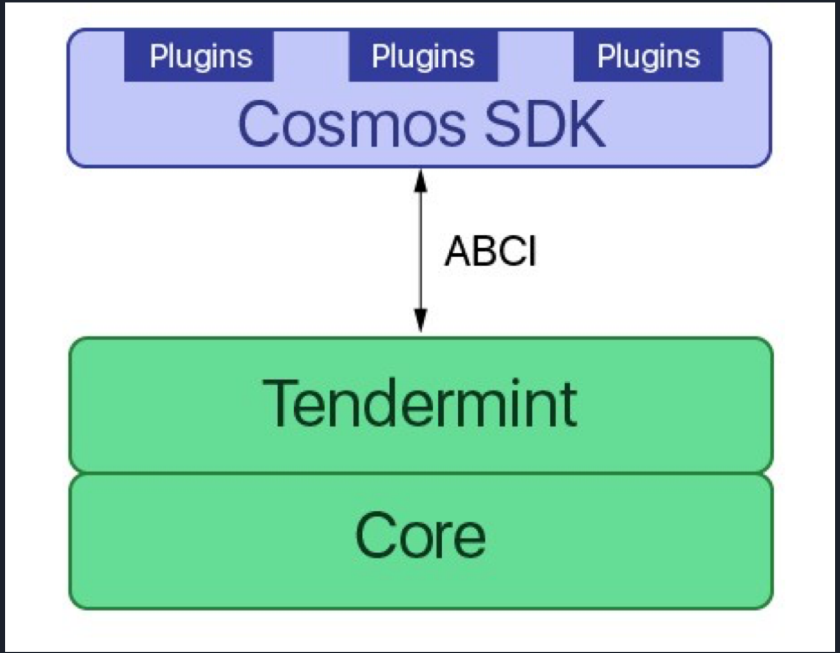
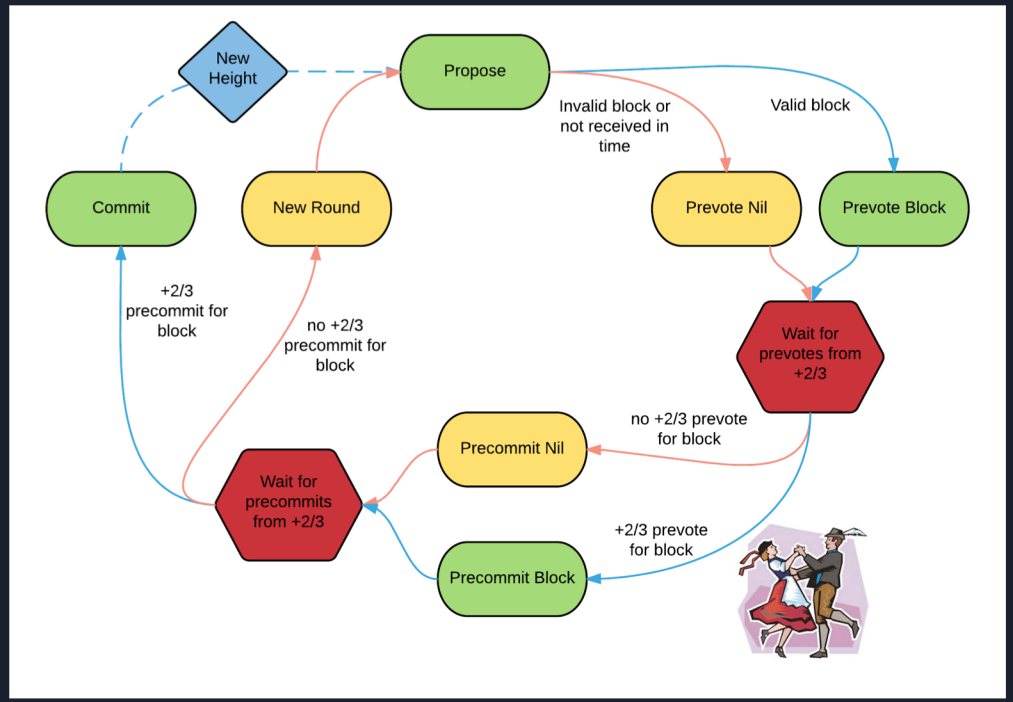


Photo credit to “Tendermint Explained- Bringing BFT based PoS to the Public Chain Domain

Tendermint



1: leader rotation

2: PoLC lock mechanism

3: Gossip protocol

Photo credit to “Tendermint Explained- Bringing BFT based PoS to the Public Chain Domain”

Gossip p2p communication (k=3)

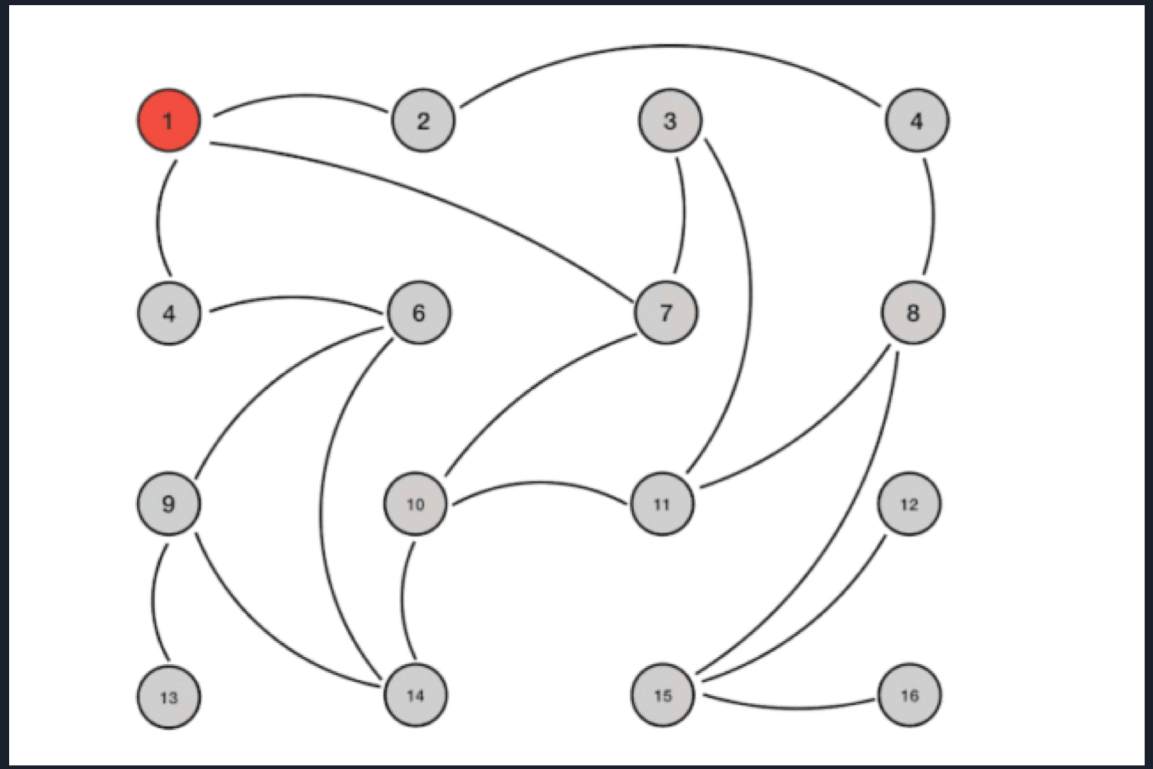


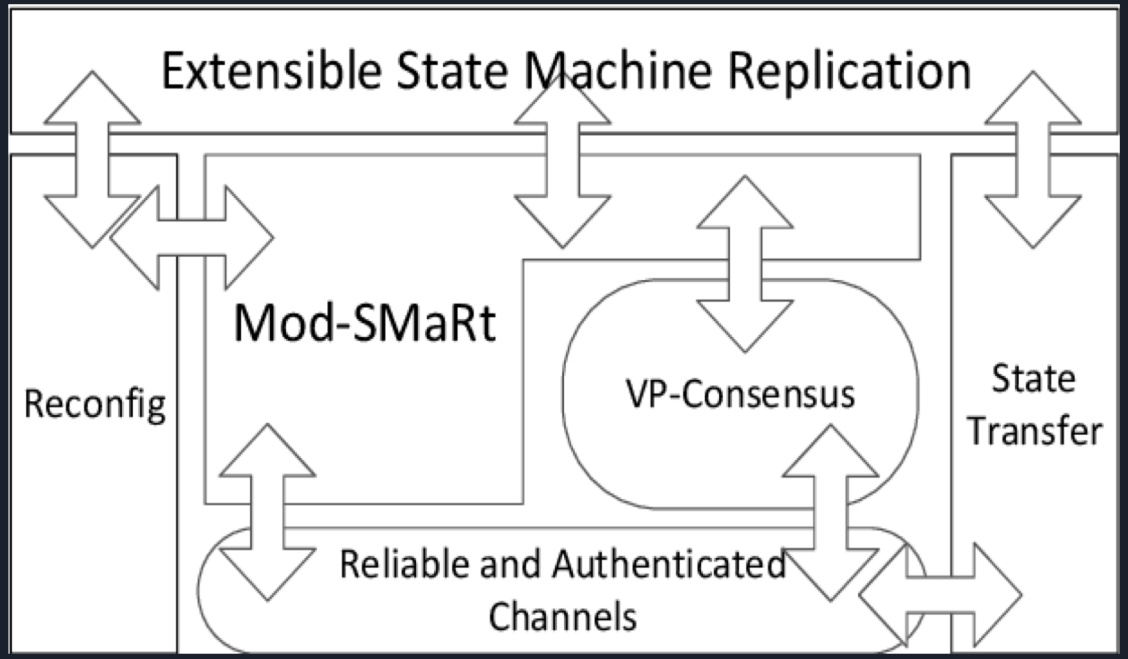
Photo credit to “ Introduction to Gossip, Just my thoughts”

Tendermint

	Network model	Any f node subverted
Safety	Eventual Synchrony	f< n/3
Liveness	Eventual Synchrony	f< n/3


symbiont

BFT-SMaRt

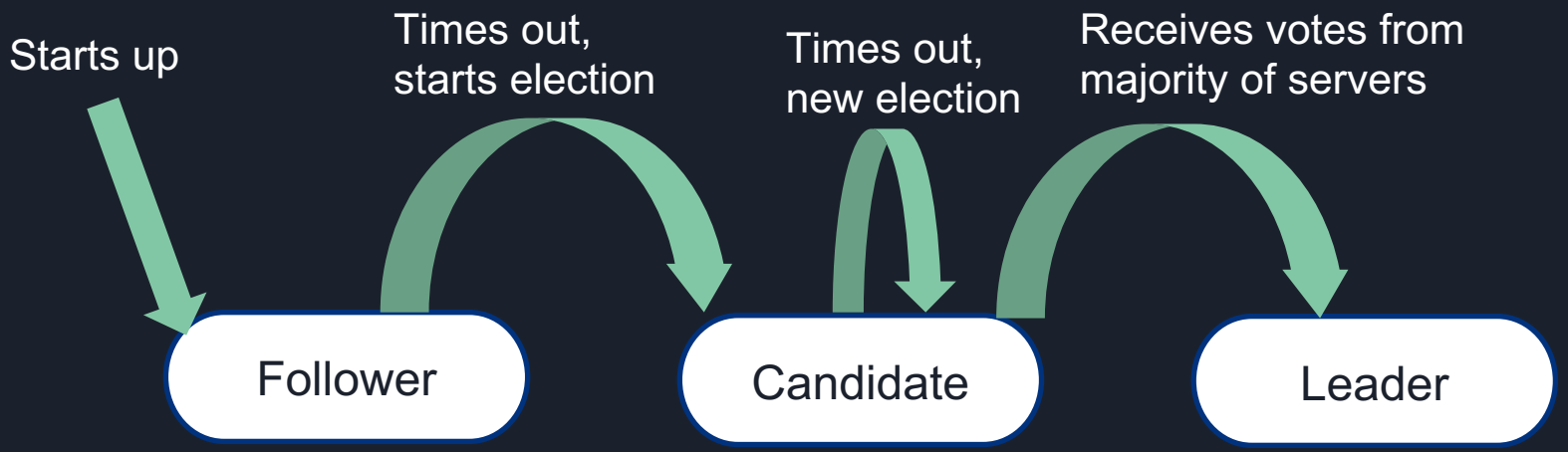


- Total Order Multicast
- State Transfer
- Reconfiguration

Photo credit to “State Machine Replication for the Masses with BFT- SMaRt”

- Raft (CFT)
- BFT- SMaRt (BFT)

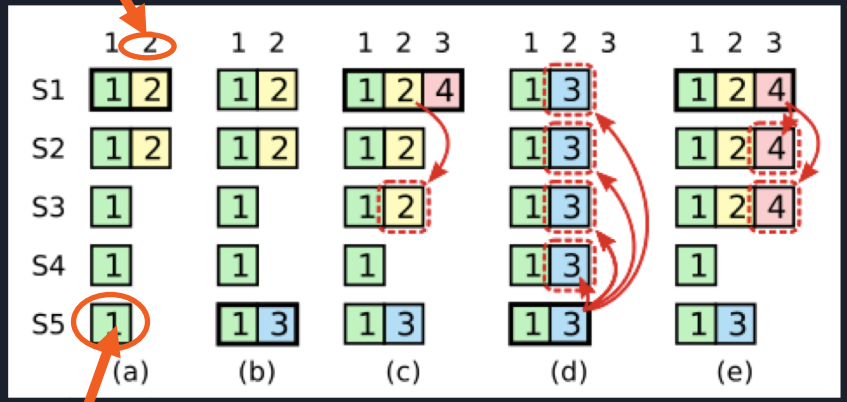
Raft

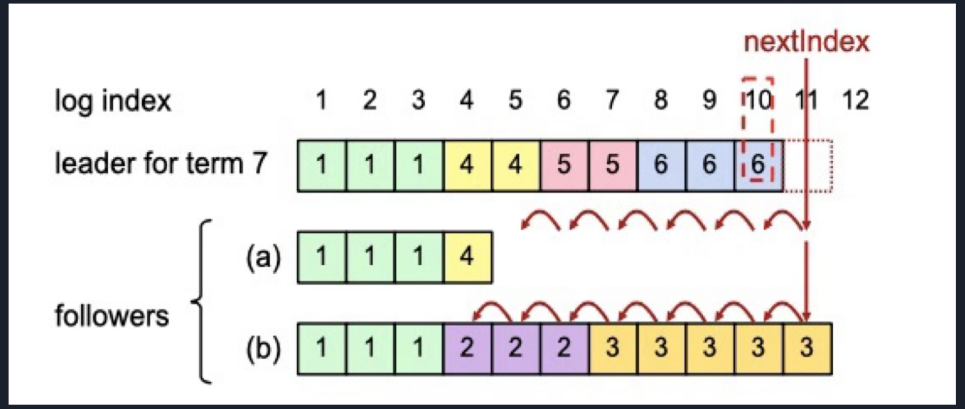

Starts up
Times out, starts election
Times out, new election
Receives votes from majority of servers
Follower
Candidate
Leader

Raft

Log index

http://thesecretlivesofdata.com/raft/#overview





Log term

Photo credit to “Talk note: Raft, the understandable Distributed protocol”

Raft

	Network model	Any f node crashed
Safety	Asynchrony	f< n/2
Liveness	Eventual Synchrony	f< n/2

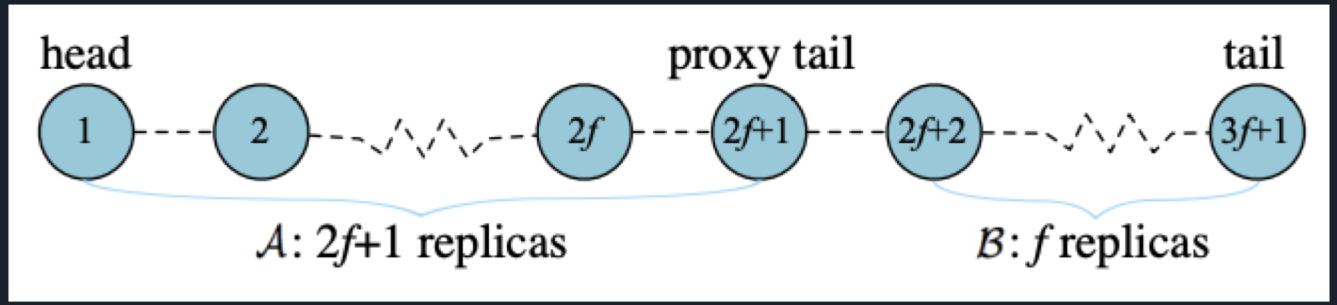
Iroha – Sumeragi

- Blockchain platform under the Hyperledger Project
- Inspired by original V0.6 design of Fabric
- “Heavily inspired” by BChain, a chain-style Byzantine replication protocol
  - Arranges the n nodes linearly only receive message from predecessor and send to successor
  - Has a leader (does not become bottleneck)

	Notary nodes	Any t nodes crash	Any f nodes subverted
Safety	n	$t < n/3$	$f < n/3$
Liveness	n	$t < n/3$	$f < n/3$

Bchain

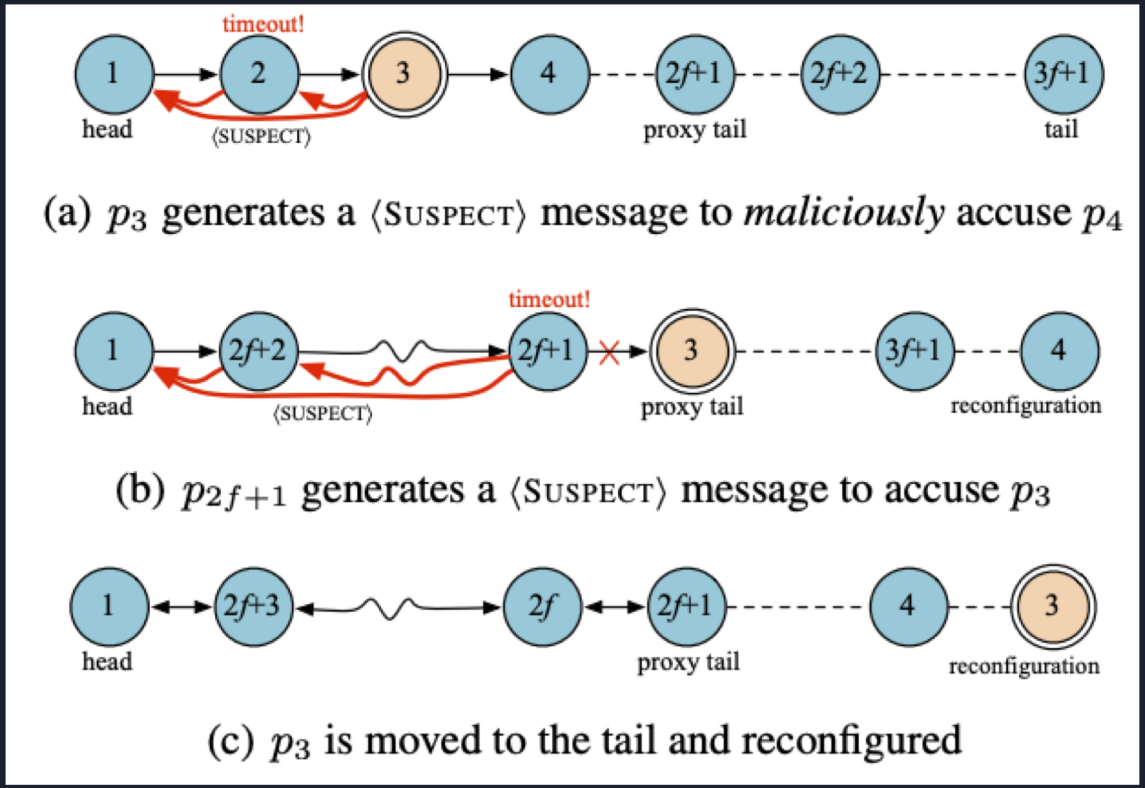
- Performs comparably to other modern protocol in fault free case
- Can quickly recover its steady state performance in face of failure
- High throughput low latency (Chain replication)
- BF detection mechanism - Re-chaining
- Asynchronous environment



BChain (Cont.)

1. Chaining
2. Re-chaining
3. Viewchange

Rechaining



Credit to BChain: Byzantine Replication with High Throughput and Embedded Reconfiguration, Sisi Duan

Kadena – Juno and ScalableBFT

- A platform for running smart contracts
- Claims to use a “Byzantine Fault Tolerant Raft” protocol
- Eventual synchrony as timing assumption
- Later deprecated -> “proprietary BFT-consensus protocol” ScalableBFT
  - Inspired by “the Tangaroa protocol”
  - Over 7000 transaction per second 256 nodes
  - Design and implementation are proprietary
- Since they are proprietary, it is not clear how it actually works nor why one should trust it
  - No summary table

Chain – Federated Consensus

- For an institutional consortium to issue and transfer financial assets on permissioned blockchain
- Uses a federated consensus protocol to reach consensus
- N nodes in the network
- One of the nodes is configured as “block generator”
  - Periodically assembles non-executed transaction into blocks -> submit for approval to block signer
- Other nodes are called “block signers”
  - Validates the block , checking the signature of the generator ->sign endorsement for the block
- When a node receives q endorsement for a block (q = 2f + 1) -> appends the block to its chain

	Generic nodes	Any t nodes crash	Any f nodes subverted	Special nodes	Any s special nodes crash	Special nodes subverted
Safety	n	$t < n/3$	$f < n/3$	$m = 1$	–	–
Liveness	n	$t < n/3$	$f < n/3$	$m = 1$	–	–

Quorom

- From JP Morgan Chase
- Enterprise-focused version of Ethereum (Financial use-case)
  - Do not want to expose its entire record of transactions to the public
- Executing smart contracts with the Ethereum virtual machine

https://www.bizjournals.com/pittsburgh/news/2019/09/24/exclusive-j-p-morgan-private-bank-comes-to.html

Quorum – QuorumChain

- QuoromChain as alternative of Ethereum's POW
  - Smart contract deployed with the genesis block (`Blockvoting`)
  - Nodes within a Quorum network can be given the Voter or Block-maker role
  - Block-maker proposes blocks
  - Voters responsible for voting on the validity of blocks
  - Smart-contract tracks whether the votes received are from valid Voters, and whether the number of votes received for a particular block is greater than the threshold
  - Synchronized

	Generic nodes	Any t nodes crash	Any f nodes subverted	Special nodes	Any s special nodes crash	Special nodes subverted
Safety	n	$t < n/3$	$f < n/3$	$m = 1$	–	–
Liveness	n	$t < n/3$	$f < n/3$	$m = 1$	–	–

Quorum - Raft

- Recent consensus option available for Quorum
- Uses implementation of ETCD as alternative of Ethereum's POW
  - Raft for consensus
- Each Ethereum node is also a Raft node
- Synchrony only for liveness

Ethereum	Raft
Minter	Leader
Verifier	Follower

	Generic nodes	Any t nodes crash	Any f nodes subverted
Safety	n	$t < n/2$	–
Liveness	n	$t < n/2$	–

MultiChain

- Bitcoin blockchain is not yet suitable for institutional financial transactions.
- Permissioned blockchains in the financial industry and for multi-currency exchanges in a consortium
- Consensus mechanism called mining
  - Resolves the dilemma posed by private blockchains, in which one participant can monopolize the mining process
  - Define a parameter 0<= *mining diversity* <=1
    - 1 means every node must mine (Round robin)
    - 0 means no restriction

Further platforms

Hydrachain

- Permissioned distributed ledger using Ethereum infrastructure
- Initially inspired by “Tendermint”
- Correctness unclear
  - Explanation and review required

The Swirlds hashgraph algorithm

- Implemented in an open-source consensus platform “Babble”
- Targets consensus in permissioned blockchain
- Operates in a “Completely asynchronous” model
- No independent validation or analysis available

Q A

Thank you!!

Reference

https://www.bizjournals.com/pittsburgh/news/2019/09/24/exclusive-j-p-morgan-private-bank-comes-to.html

https://www.blockchainexpert.uk/blog/hydrachain-dapp