Quorum & JPM coin

Jul 2020

Cheng-Han (Hank) Tsai
hanktsai68@gmail.com

Outlines

01 Quorum Overview

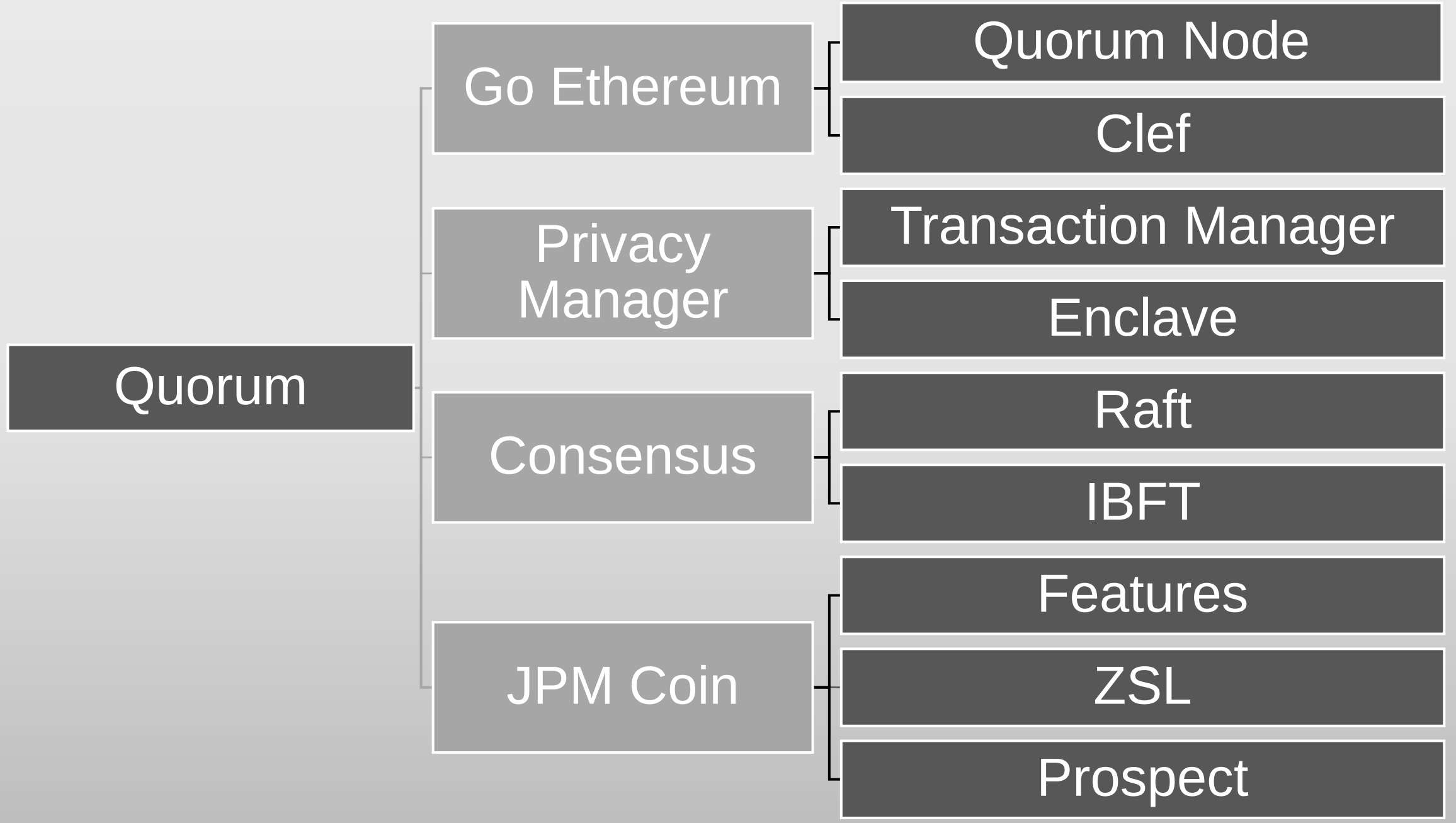
Basic concept of Quorum, the blockchain implementation behind JPM coin.

02 Consensus

Introduction to Raft and IBFT consensus algorithm, which Quorum applies to substitute for PoW of Ethereum.

03 JPM Coin & Quorum

Introduction to JPM coin and how JPM coin manages the double-spending problem in private contract.


Quorum
Go Ethereum
Quorum Node
Clef
Privacy Manager
Transaction Manager
Enclave
Consensus
Raft
IBFT
JPM Coin
Features
ZSL
Prospect

Quorum Overview

The first look at JP Morgan’s blockchain implementation

What is Quorum?

- ✓Go Ethereum
  - ✓Based on Ethereum
  - ✓Built and maintained by JP Morgan
  - ✓A blockchain platform aimed at enterprise users
  - ✓Cares about clients' privacy
- ✓Private Transaction
  - ✓Supports private transactions and private contracts
  - ✓Content of a private transaction can only be known by its participants and regulators
  - ✓Still, private transactions can be decentralized and satisfy non-repudiation
- ✓Permissioned Blockchain
  - ✓Nodes need to acquire permission before joining the blockchain
  - ✓Regulators and Admin Agents can be specified in the blockchain

Main Features

Public & Private Transaction

Segmentation of Local Database

Permissioned Mechanism

Consensus

Faster block minting time

Based on Ethereum

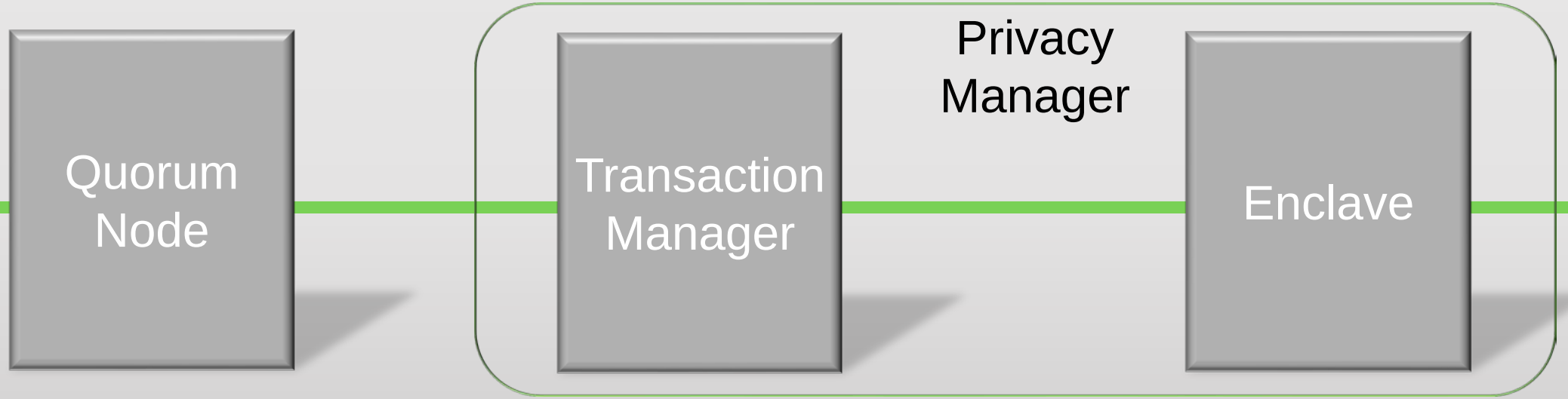
Smart contracts can be written in Solidity, just like public Ethereum.

Main Features

- ✓Public & Private Transaction
  - ✓Clients can choose whether the transaction can be verified by all nodes in the transaction
- ✓Segmentation of Local Database
  - ✓One is for public state, the other for private state
- ✓Permissioned Mechanism
  - ✓Quorum permissioned mechanism works as a firm organization, which specifies network manager, sub organization to control the access of related accounts
- ✓Consensus
  - ✓Raft & IBFT
  - ✓No fork
  - ✓Faster minting time

Quorum Client

To make private transaction possible, a Quorum client need to be endowed with several components in addition to a blockchain node.



- ✓ Succeeded from Ethereum node
- ✓ Block related work

- ✓ Interacts with other transaction managers
- ✓ Responsible for storing and sending encrypted transaction data
- ✓ No access to private key

- ✓ Cryptographic service for transaction manager
- ✓ Isolated component
- ✓ Handles keys (public/private)
- ✓ Hardware or software

Privacy Manager (Tessera Service)

Transaction Manager

- ✓ A P2P network among transaction managers
- ✓ Broadcasts key information
- ✓ Stores state data
- ✓ Gateway to distribute private transaction

Call for encryption/decryption
Call for keys

Enclave

- ✓ Key access
- ✓ Fetches default public key for attached Quorum node
- ✓ Return public keys on call from Transaction manager
- ✓ Encrypts/Decrypts private transactions
- ✓ Adds recipients of a private transaction

Return encrypted/decrypted transactions
Return keys

Private Transaction

- ✓ Transaction Contents
  - ✓ Besides the contents of an Ethereum transaction, Quorum adds a new parameter privateFor[]
  - ✓ privateFor[] stores the public keys of the participants and the regulator
  - ✓ Private transactions only affect private state database
  - ✓ The participants and payload will be replaced with a hash value on chain
- ✓ ECDSA
  - ✓ A signature comprises (r, s, v), where v is for public key recovery. For private transaction, v is 37 or 38
  - ✓ For public transaction, v is 27 or 28
  - ✓ Public key recovery: $r^{-1}(sR - SHA(x)) = Q = dG$
  - ✓ A node won't process transaction with $v = 37, 38$, unless it is a participant.

Lifecycle of a Private Transaction

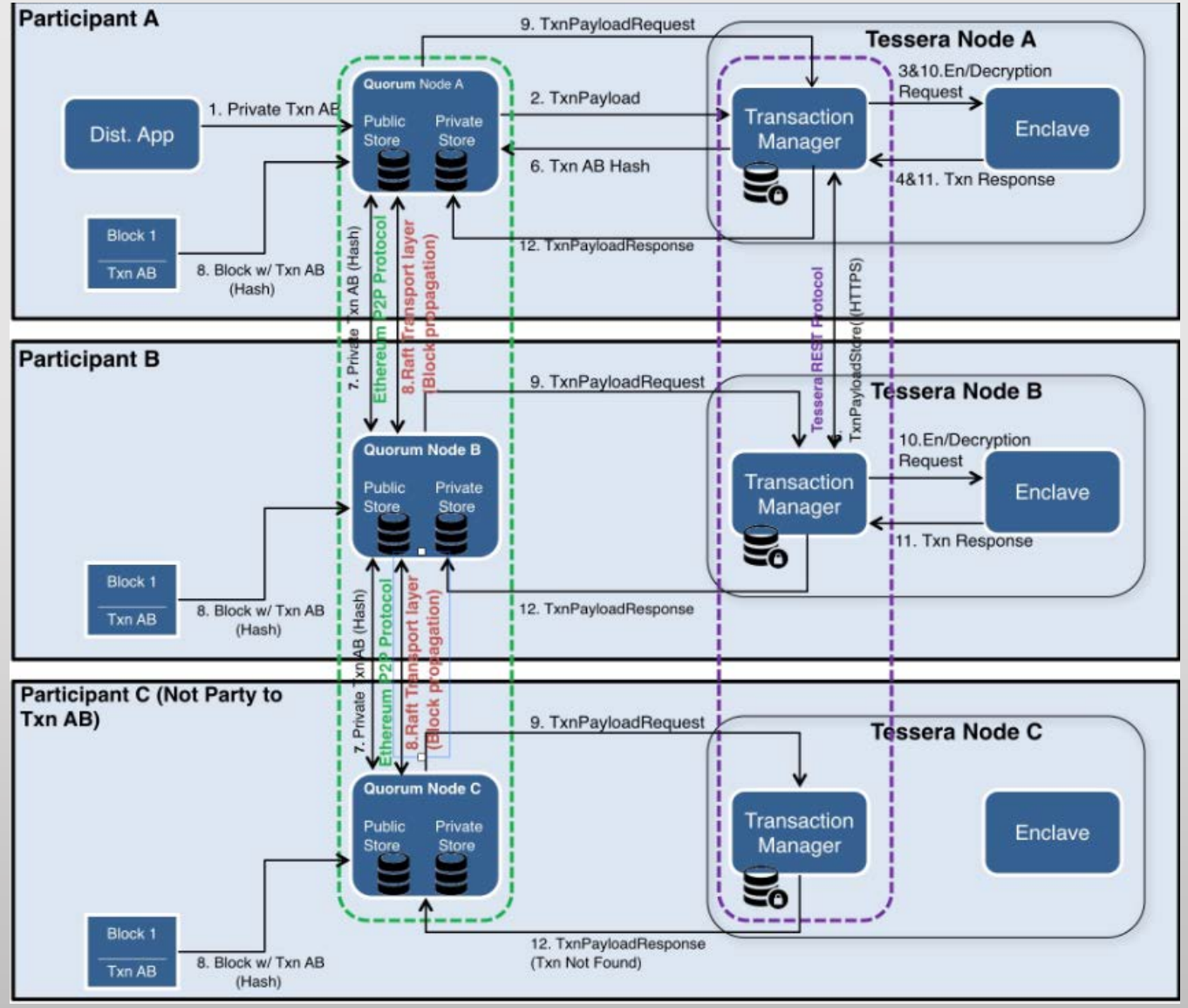


Figure from goquorum.com

Lifecycle of a Private Transaction

Suppose that there is a private transaction from Party A to Party B

Step 1

Party A sends the transaction to the Quorum node, which later passes it to the transaction manager

Step 2

Party A's Transaction manager calls A's enclave to generate a symmetric key and encrypt the payload. Then the enclave encrypt the key with share keys with each participant

Step 3

Party A's transaction manager sends the encrypted payload and key to Party B. B is able to decrypt the transaction using the encrypted symmetric key

Lifecycle of a Private Transaction

Suppose that there is a private transaction from Party A to Party B

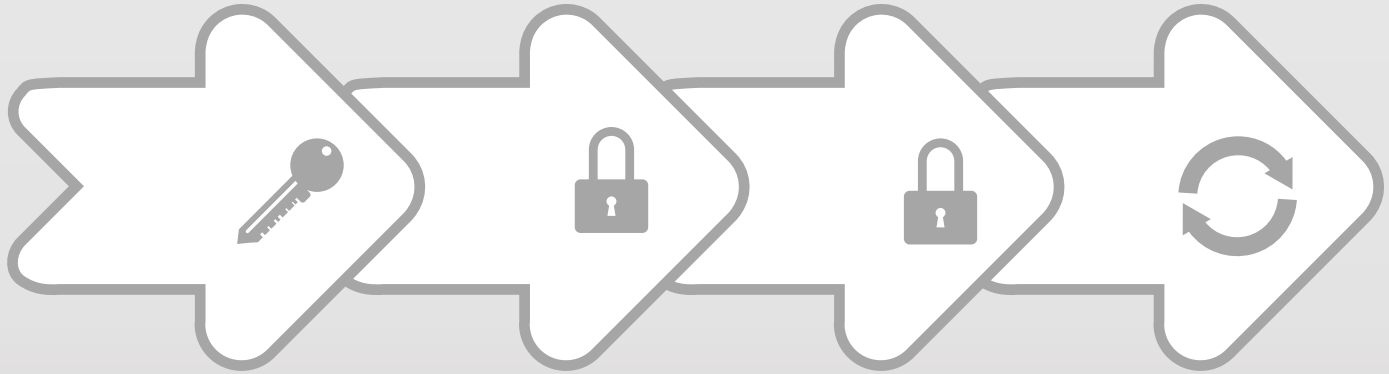
Step 4

- Party A’s computes the SHA3-512(whitepaper 256?) value for the payload, sending it back to the Quorum node.

Step 5

- Party A’s node now replaces the “data” fields with the hash value, making the transaction on chain with Ethereum protocol

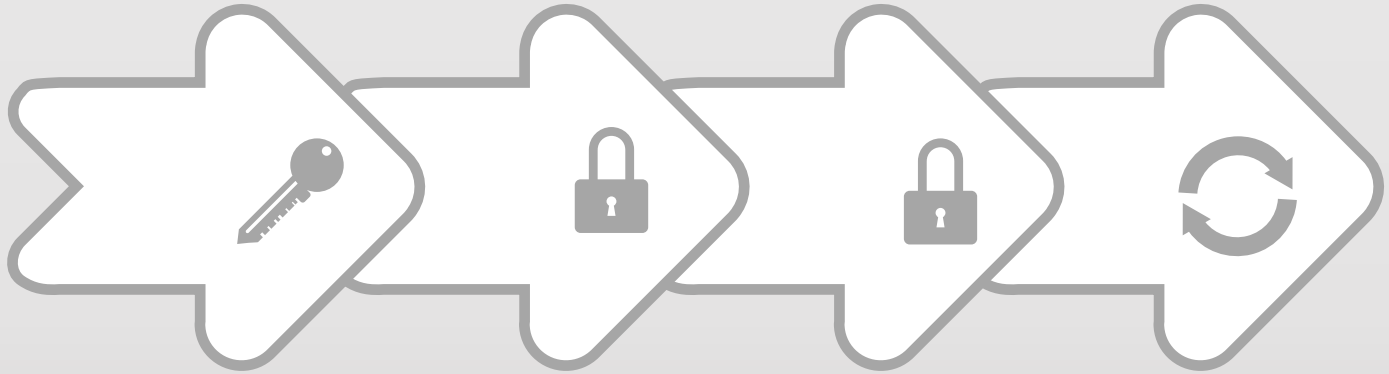
Step 2 in Depth: Enclave



01 **Generates symmetric key (tx-key)**
In addition to tx-key, the enclave should also generate two nonces.

02 **Encrypts the transaction payload**
AES-256 encryption using tx-key for the payload and one of the nonces.

Step 2 in Depth: Enclave



03 Encrypts tx-key

Derives share keys with each participants with ECDH (i.e. using one's private key with the other's public key). Encrypts tx-key and the other nonce with each share key.

04 Returns to transaction manager

Returns the encrypted payload, encrypted tx-keys, both nonces and the public keys of sender and recipients.

Cryptography

- ✓ECC
  - ✓NaCl library by default
  - ✓Secp256r1 by default
  - ✓ECDH key exchange
  - ✓ECDSA
- ✓Symmetric Cryptography
  - ✓AES256 with GCM mode
  - ✓GCM is mandatory
- ✓Message Digest (Transaction Manager)
  - ✓SHA3-512 (Whitepaper 256?)
- ✓Clef (Go-Ethereum Node)
  - ✓SHA3-256
  - ✓ECDSA (Secp256k1 by default)
  - ✓Key management

An Example: One Contract per Bank

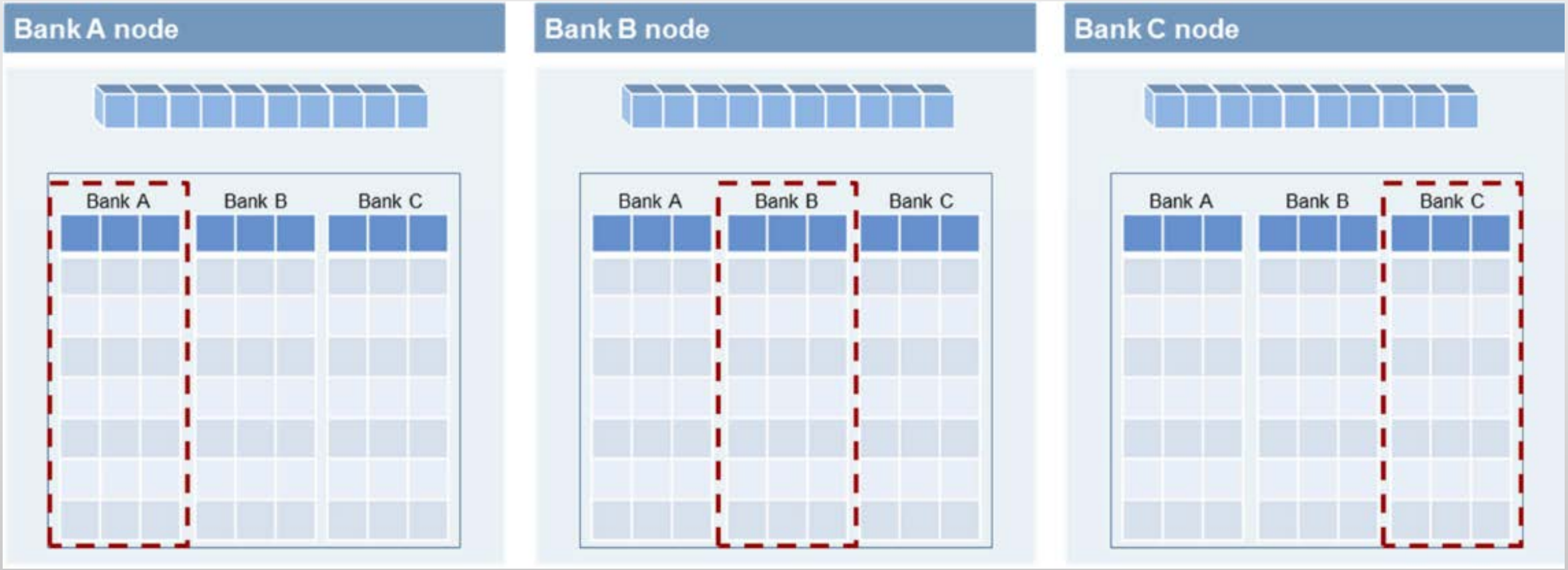


Figure from goquorum.com

Permission

- ✓Enhanced Permissions Model
- ✓Definition
  - ✓ Network
  - ✓ Organization
  - ✓ Sub organization
- ✓Written in smart contracts
  - ✓ Initiation of contract extension can only be done by network admin or org admin account

Permission

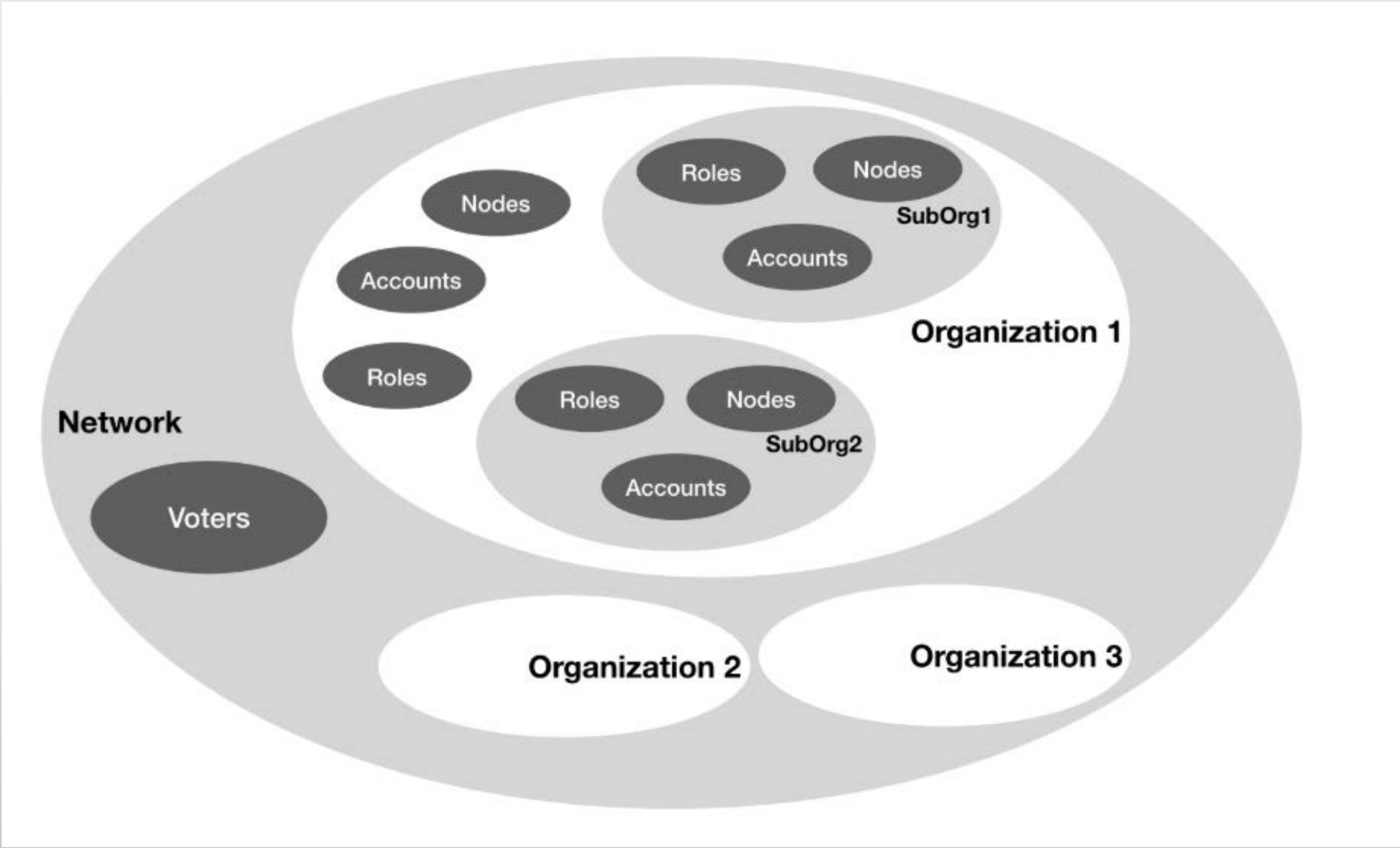


Figure from goquorum.com

Consensus

Raft and IBFT

RAFT

A fast consensus algorithm for replicated state machine (such as blockchain). A Quorum node is also a Raft node.

- ✓Features
  - ✓No fork on the blockchain
  - ✓Fast block minting time
  - ✓Efficiently forms a sole consensus (finality)
  - ✓No faulty tolerance
  - ✓Two phase algorithm: Leader election → Log replication

https://raft.github.io/

RAFT

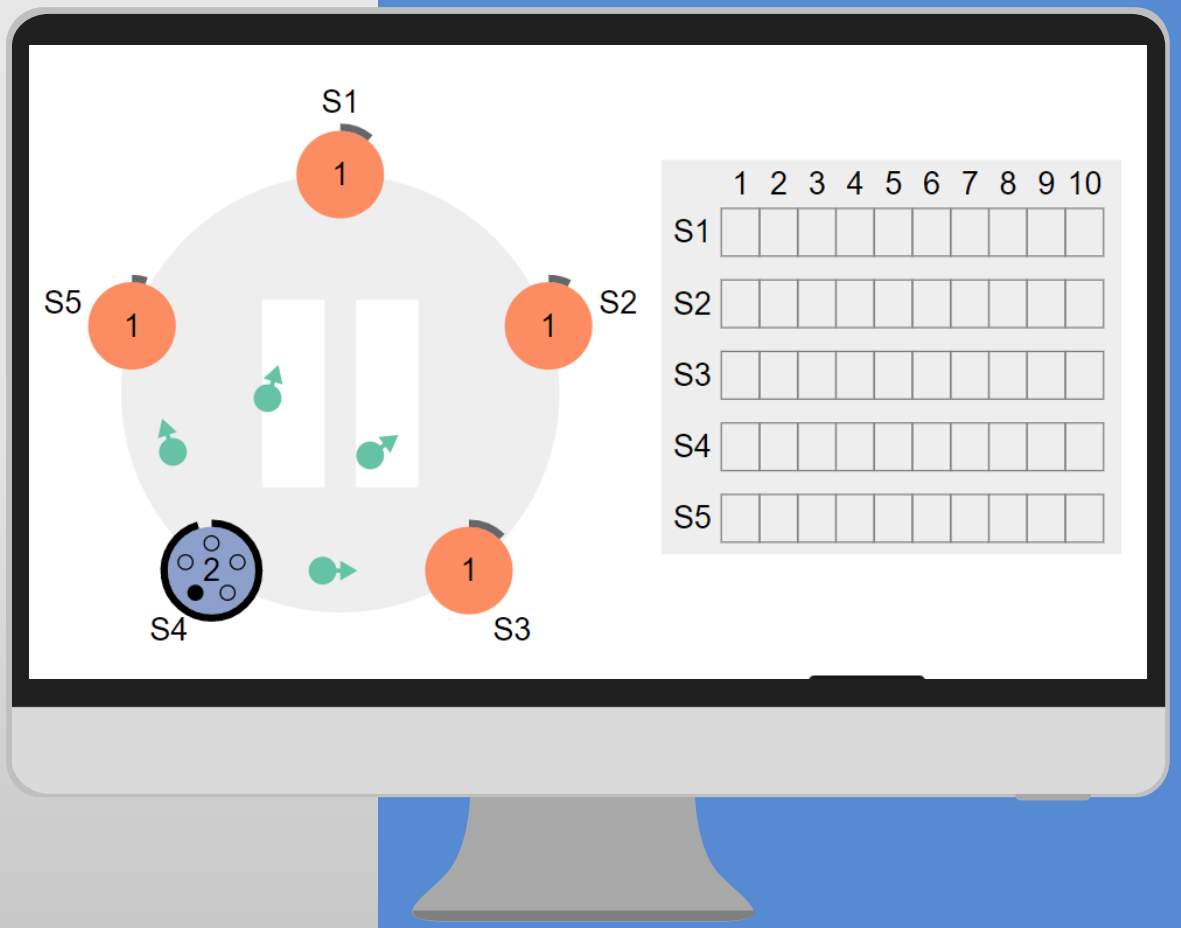
- ✓ Leader & Followers
  - ✓ Each “term” will have a single leader
  - ✓ Leader is responsible for minting the block
  - ✓ Followers wait for “heartbeat” from the leader
  - ✓ Followers comprises verifiers and learners
  - ✓ When receiving a block, followers (verifiers) confirm it and send back confirmation to the leader
  - ✓ After receiving half of the verifiers’ confirmations, leader send the commitment to all nodes
  - ✓ Followers append the block once receiving commitment

https://raft.github.io/

Leader Election

Each “term” must have a single leader

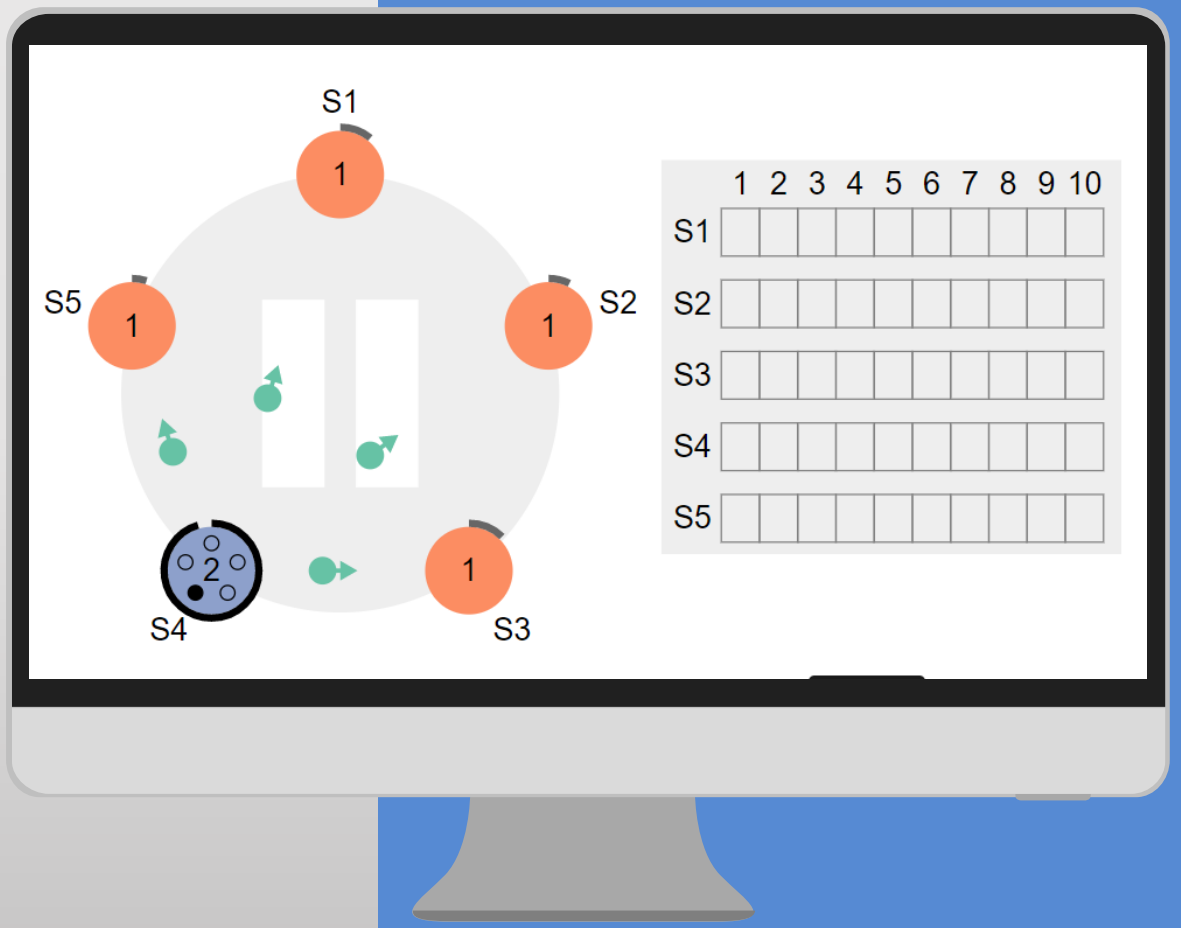
1. Two ways to enter leader election: leader crash or system initialization. Nodes will enter a “sleep” status for random timeout
2. Once a node wakes up, it becomes a “candidate” (term++) and sends RequestVote RPCs to all nodes. It will also vote itself to be the leader
3. A node vote on respond of any RequestVote RPCs if it hasn’t do so



Leader Election

Each “term” must have a single leader

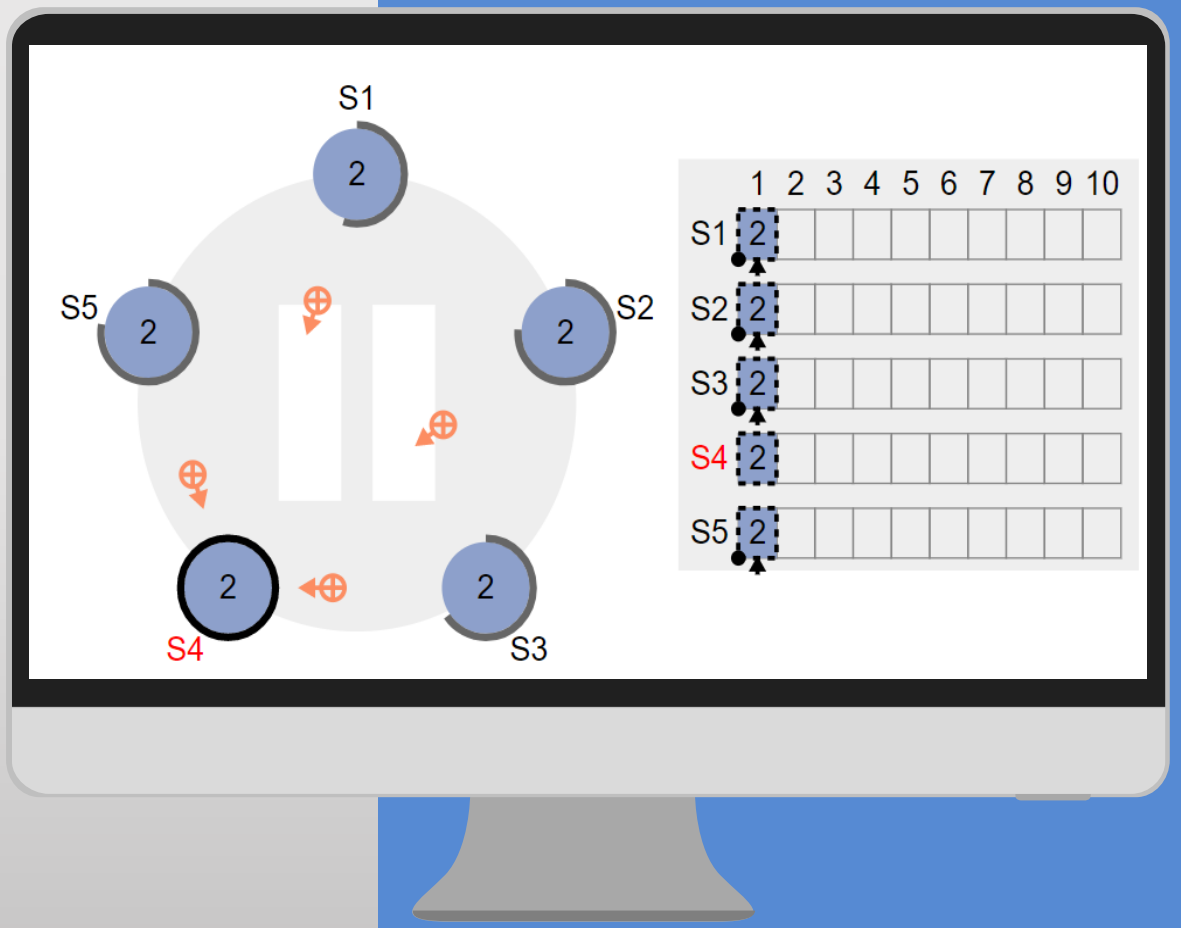
4. Nodes disregard RequestVote RPCs if they believe that there is an existing leader
5. The leader is determined once a node has received more than half of total nodes



Log Replication

How does the consensus form?

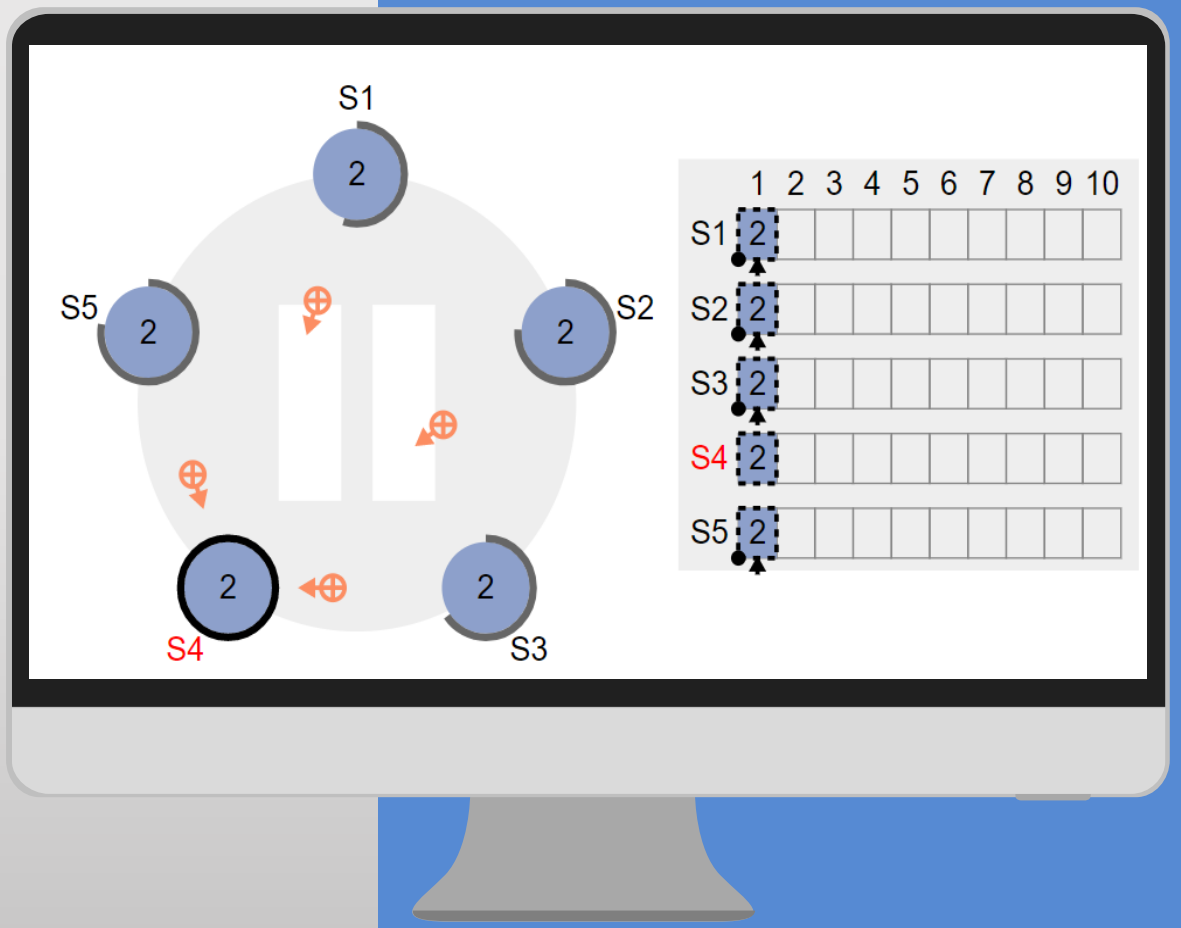
1. The leader should send “heartbeat” to all followers regularly
2. The block should be attached with heartbeat once it is minted by the leader
3. Verifiers will verify the block and send back confirmation to the leader



Log Replication

How does the consensus form?

4. Once receiving half of confirmations, the leader sends “commitment”(AppendEntries RPCs) to all nodes
5. Followers append the block to the blockchain



Crash

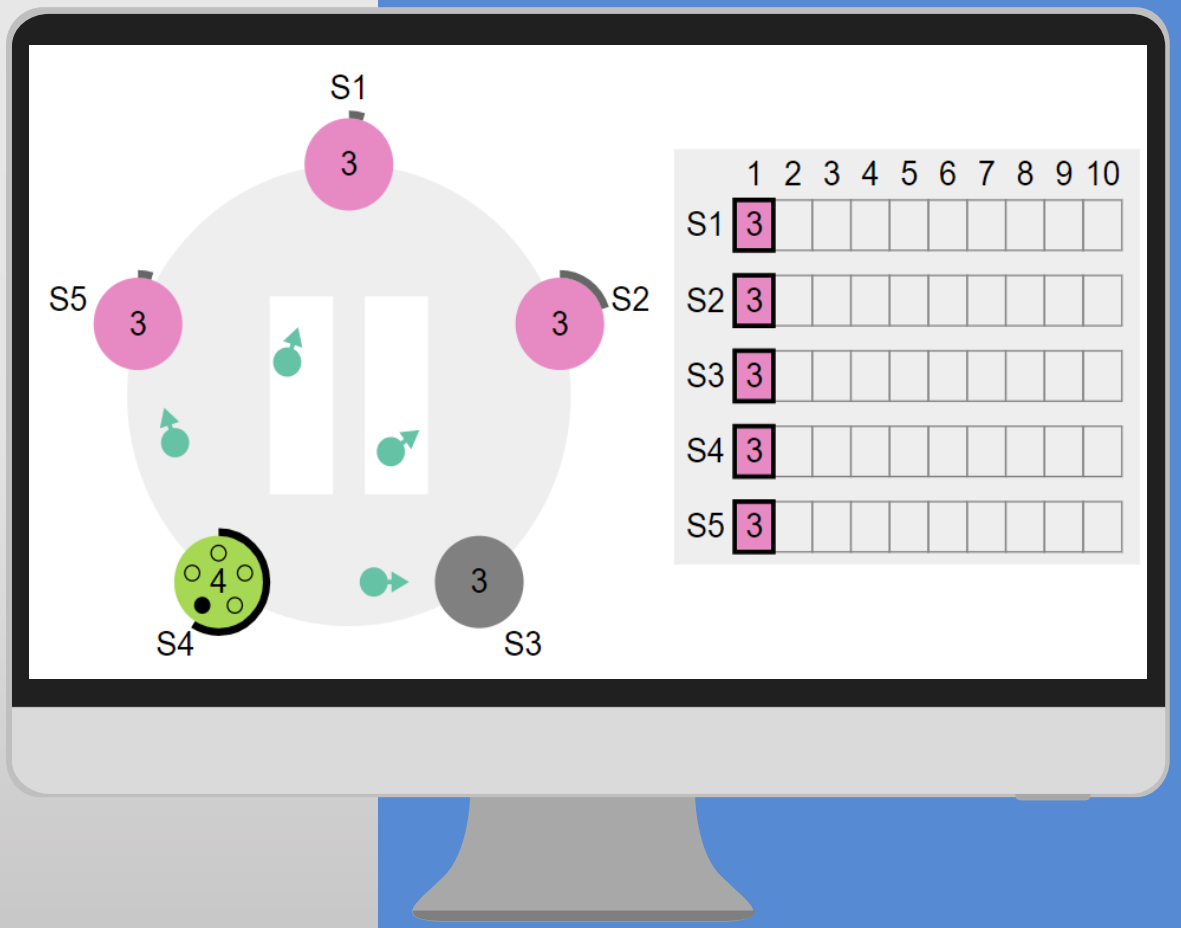
What happens when a node crashes?

Leader crash

- ✓ The heartbeat will stop
- ✓ Starts a new election

Follower crash and asymmetric network

- ✓ Local blockchain will be inconsistent with the others’
- ✓ Once available, leader will check which previous block is consistent with the node
- ✓ In fact, at the beginning of a term, the leader always checks the latest consistent block with each node



IBFT

Istanbul Byzantine Fault Tolerant

- ✓ Byzantine Fault Tolerant
  - ✓ Byzantine Generals Problem
  - ✓ IBFT allows up to 1/3 of nodes be malicious
  - ✓ Adapted if nodes do not trust each other
- ✓ Proposer and Validators
  - ✓ Proposer of each round is responsible of minting blocks
  - ✓ Validators verify the block and “vote” on the block
- ✓ Features
  - ✓ No fork on the blockchain
  - ✓ Self-verified blockchain consensus mechanism
  - ✓ Three phase algorithm: Pre-prepare → Prepare → Commit
  - ✓ Extra contents of a block header

An IBFT Round

Pre-prepare

- ✓ Proposer mints the block (block proposal)
- ✓ Proposer broadcast it to the network with the PRE-PREPARE message (Validators enter pre-prepare state)
- ✓ Makes sure that all nodes are in the same round
- ✓ Validators verify the block, broadcasting PREPARE message

Prepare

- ✓ A node enter prepare state once receiving 2/3 of PREPARE messages
- ✓ A node then broadcasts COMMIT message

Commit

- ✓ A node enter commit state once receiving 2/3 of COMMIT messages
- ✓ The block is appended and confirmed

To trigger a round change

- ✓ Round change timer expires
- ✓ Invalid PRE-PREPARE message
- ✓ Block insertion fails

JPM Coin

JPM Coin: A Digital Currency Run on Quorum

The First Look

Issued by JP Morgan Chase

- ✓ Stable coin: USD/JPM Coin = 1
  - ✓ The value is backed by JP Morgan Chase
- ✓ Not a cryptocurrency
- ✓ Rather than money, it is more like Transaction Deposits (Interbank Certificates of Deposits, for financial institution case)
- ✓ For institutional customers passing JP Morgan KYC only
- ✓ First issuance at Feb. 2019

Features

- ✓RTGS
  - ✓Real Time Gross Settlement
  - ✓Network Admin (i.e. JP Morgan Chase) as the Clearing House
- ✓Blockchain
  - ✓Decentralized ledger is almost impossible for a hacker to hack
  - ✓It is easy for customers to trace the transaction process
  - ✓Lower cost for settlement
  - ✓Lower possibility of settlement fault
- ✓Quorum
  - ✓Quorum makes protecting client’s privacy and business secrets become possible on blockchain
  - ✓Consensus algorithm makes settlement faster and more efficient
  - ✓Regulators are able to monitor the whole network

How Does JPM Coin Work?

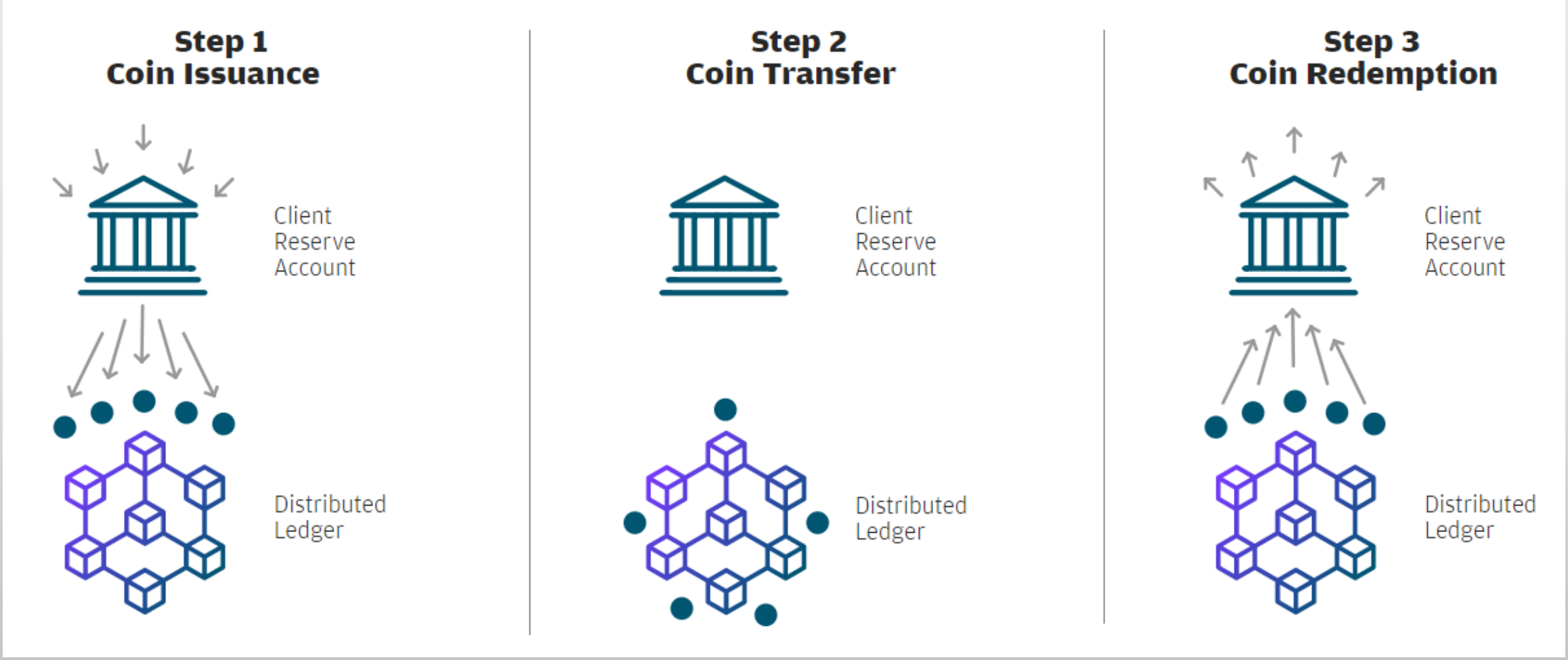
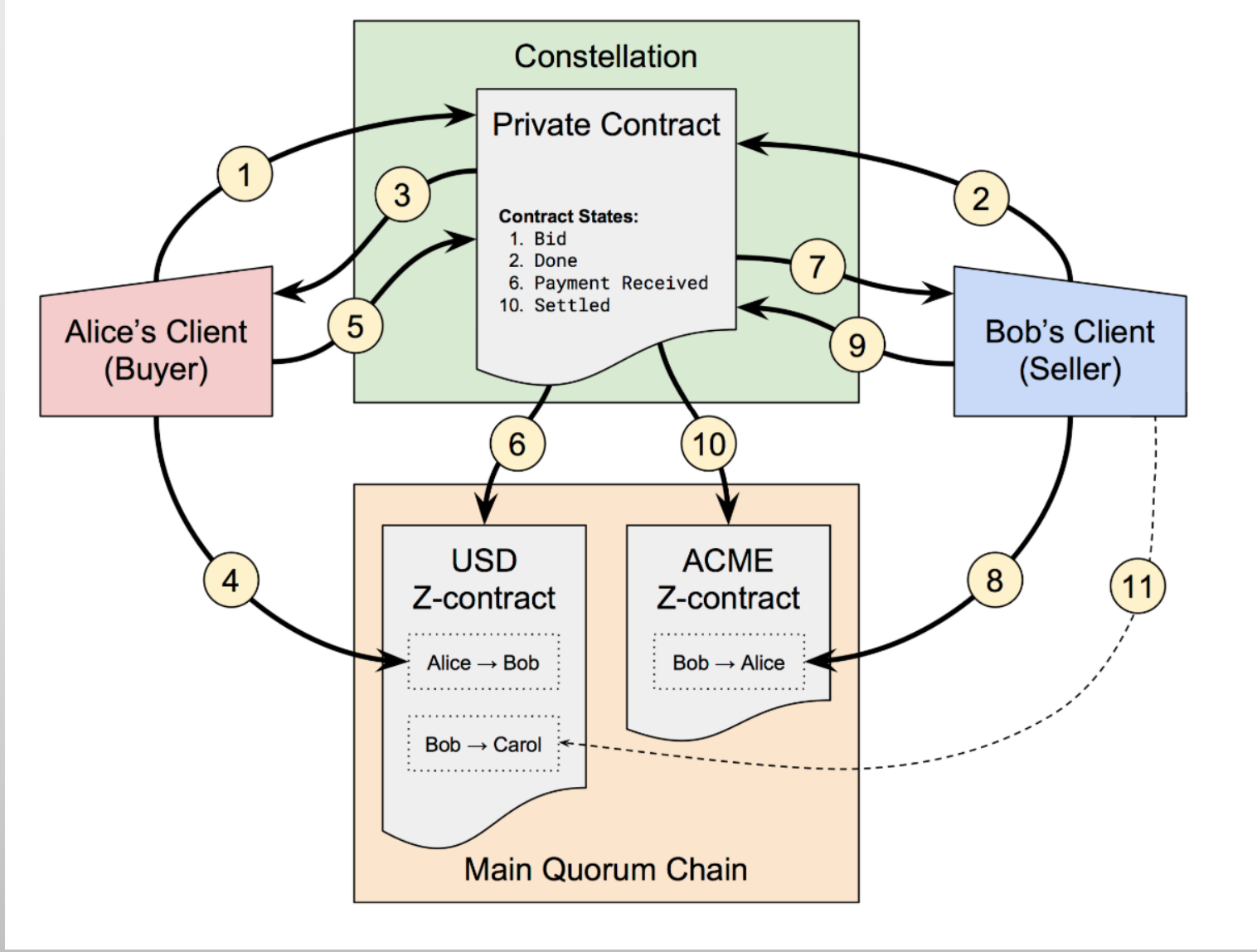


Figure from jpmorgan.com

Double Spending

- ✓ Double Spending on Private Transaction
  - ✓ Go-Ethereum can prevent double spending for public transaction
  - ✓ There should be a solution to prevent double spending for private transaction → zero-knowledge proof !
- ✓ Proof of Concept (POC)
  - ✓ Demonstrates how Zero-Knowledge Security Layer (ZSL) work on Quorum
  - ✓ Designed by Zcash team
  - ✓ Using zk-snarks and UTXO
  - ✓ Issuance and redemption of “shielded assets” called z-tokens
  - ✓ A private transaction and a public z-contract are involved

Proof of Concept



https://github.com/jpmorganchase/quorum/wiki/ZSL

ZSL

Zcash protocol on Quorum

- ✓ Note
  - ✓ The basic transaction unit of z-token (address in Bitcoin)
  - ✓ Comprises (PK, v, rho)
  - ✓ $Nullifier = SHA256\left(0x01, sk_{sender}, rho_{Input}\right)$
  - ✓ $Commitment = SHA(Ouput\ Note)$
  - ✓ A note should always be sent through private transaction

ZSL

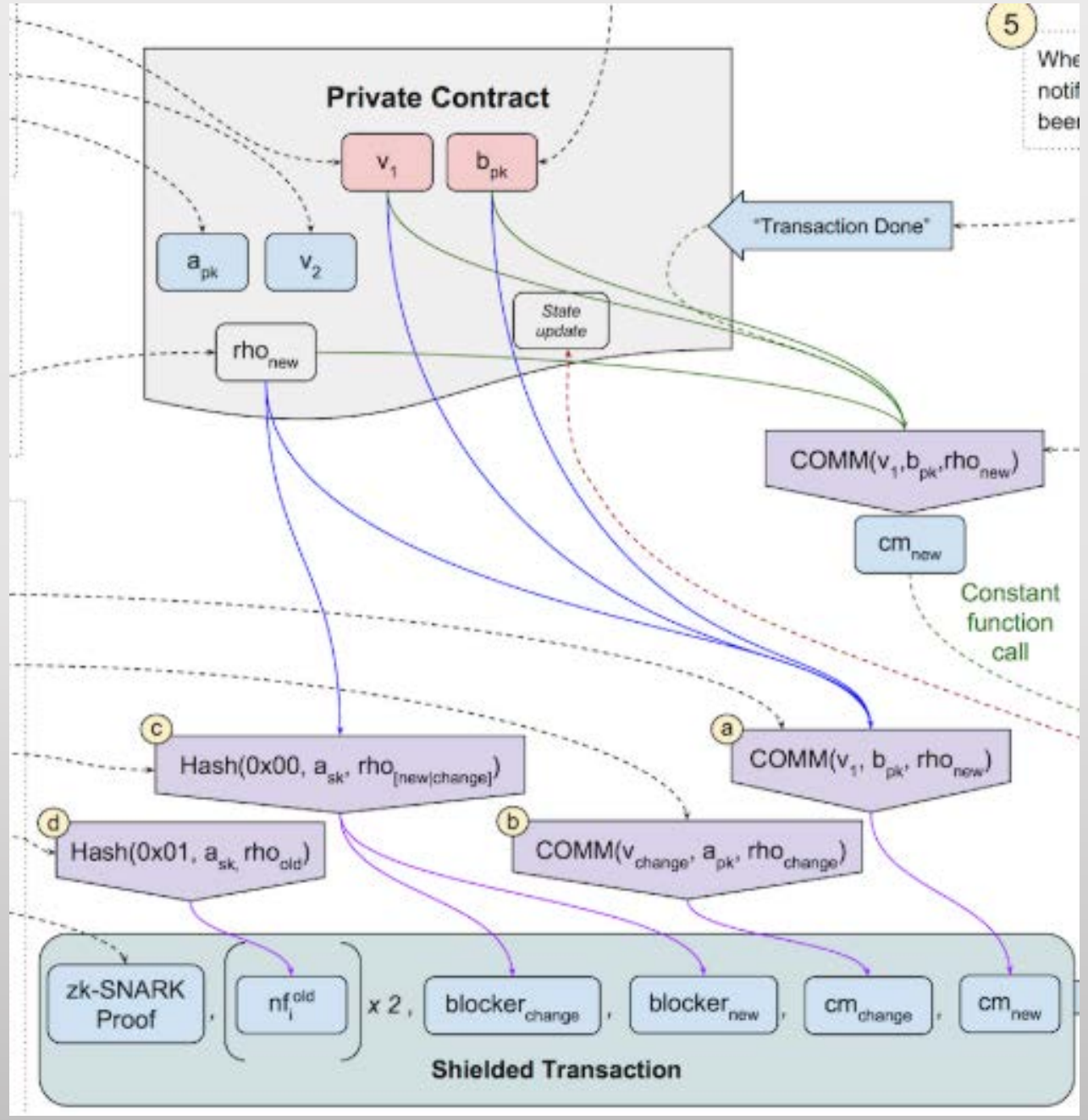
Zcash protocol on Quorum

- ✓ Shielded Transaction (z-contract)
  - ✓ A public contract
  - ✓ Minted on chain without knowing contents of nullifiers and commitments
  - ✓ Confirmed the transaction has done
- ✓ Double Spending Prevention
  - ✓ Every nodes can check whether the same nullifier is spent twice
  - ✓ Sender needs to proof that she owns (PK, sk, rho) to compute the nullifier and the amount of nullifier is at her free deposal with zk-snarks

ZK-SNARKS

A quick look at zero-knowledge proof

- ✓ Main Target
  - ✓ Proof the acquisition of the solution $f(x_1, \dots x_n) = 0$ without revealing the solution itself
- ✓ Mathematics Required
  - ✓ Transform f into quadratic form
    - ✓ $P(n) \equiv s \cdot C(n) - s \cdot A(n) * s \cdot B(n) = H(n) * Z(n)$
    - ✓ Proves that one knows the s vector
  - ✓ Homogenous Hiding
    - ✓ Homogenous function such as $E(x) = xG$
  - ✓ KCA
    - ✓ Based on $E(ax + by) = aE(x) + bE(y)$
  - ✓ Elliptic Curve Pairing (Bilinear map)
    - ✓ $e(rP, kP) = e(P, rkP) \Rightarrow e(R, Q) = e(P, S)$



https://github.com/jpmorganchase/quorum/wiki/ZSL

Future for Quorum & JPM Coin

International transfer

- ✓Extended to other major currencies
- ✓Lower transfer cost and time hugely
- ✓International clearing

Assets transaction

- ✓Transaction over financial assets
- ✓Again, the time cost decreases
- ✓Lower default rate

Future for Quorum & JPM Coin

Lending System

✓ Lending condition and default rate computation can be written on smart contracts

If such coin is issued by FED…?

Q & A

Thank You