BLOCKCHAINS IN 5G NETWORKS

DILIP KRISHNASWAMY

NOV 30, 2019

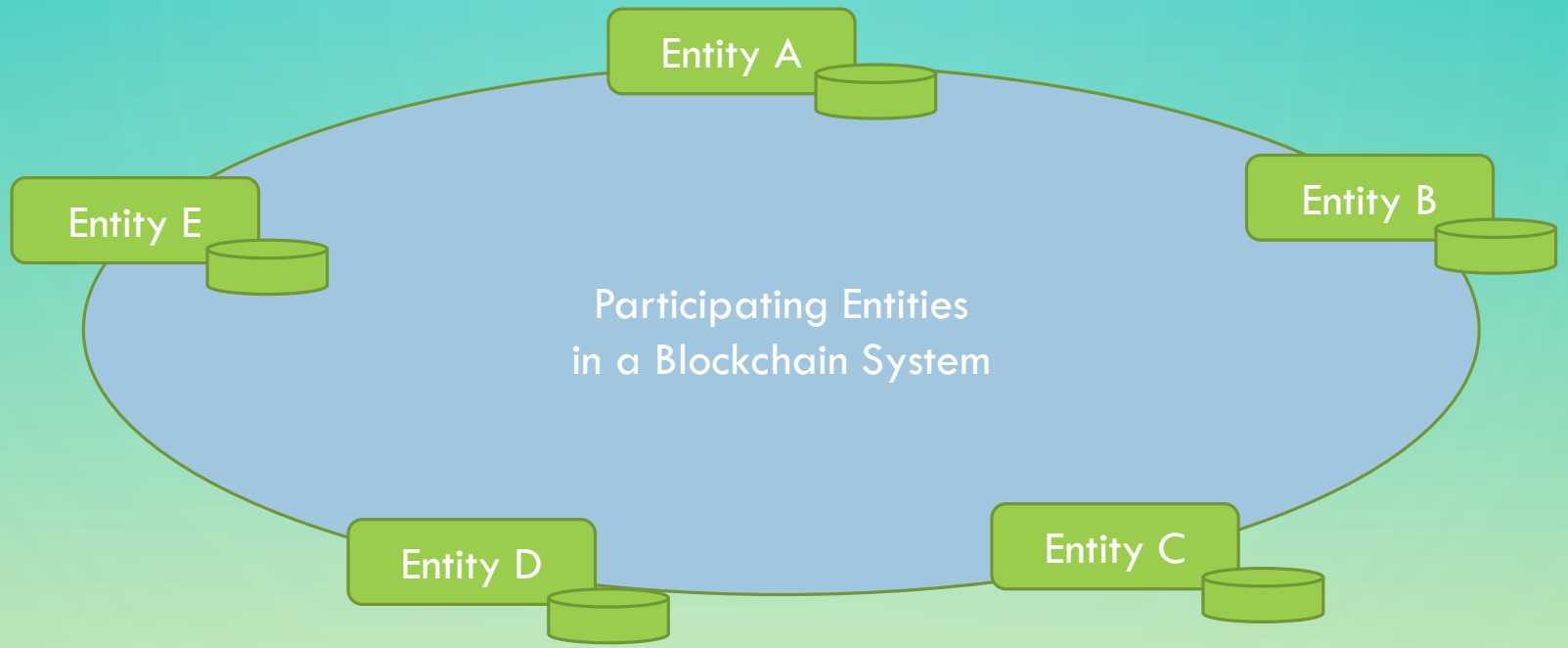
THE BLOCKCHAIN TERM – VARIOUS INTERPRETATIONS

- Blockchain vs DLT?
- Public Blockchain / Private Blockchain ?
- Decentralized Trust System?
- CryptoCurrency/Token System?
- Distributed Consensus?
- Proof of Work / Proof of Stake?
- DAG-based systems?
- Permissioned Blockchain ?
- Distributed Shared Ledger ?
- Selectively Replicated Distributed Ledger?
- Zero Knowledge Proof Systems?

EXAMPLE APPLICATIONS

- Telecom - MNP and UCC (DnD) use-cases Telecom
- Data roaming across networks
- eKYC
- User Identity / Privacy mgmt
- Trade Finance
- Supply Chain Management
- Real Estate Assets
- Financial Assets
- IoT - Automotive, retail, transportation, healthcare
- Trace Provenance of Products (Agriculture, Retail)
- Employment / Education Services
- UN refugees, Migrant Laborers

SO WHAT IS A BLOCKCHAIN SYSTEM?



Typical systems have private programs updating private ledgers

Blockchain systems have shared programs updating shared ledgers

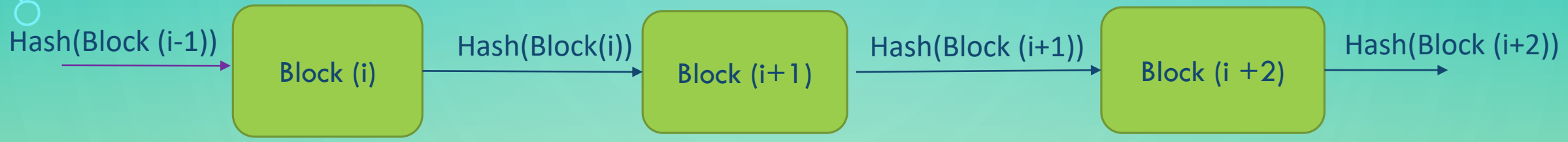
BLOCKCHAIN BENEFITS

- Sharing of Information across Entities that need to know
- Transparency of Information (can help reduce corruption in society)
- Traceability of Information (provenance)
- Can maintain privacy by only exposing information that is needed to be seen by others
- Address different verticals such as banking, finance, real-estate, telecom, supply-chain, food and agriculture, automotive, etc.

SO WHAT IS A BLOCKCHAIN SYSTEM?

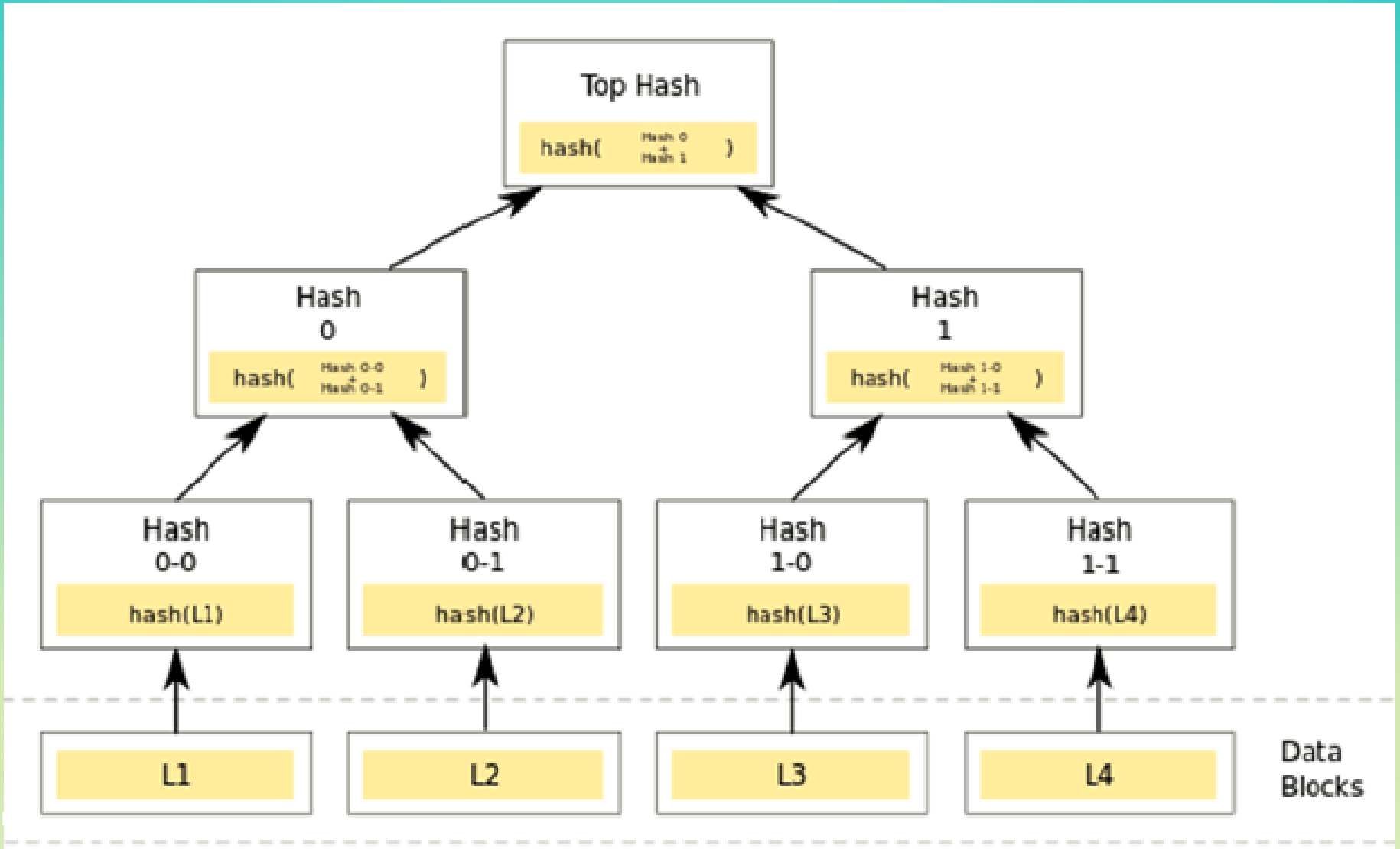
- We also need guarantees that the data in a shared ledger can be trusted
- Nodes need to agree on the data being recorded in their shared ledgers (consensus)
- The data recorded in the shared ledgers should be immutable to enable trust in the data
- The ability to extract provenance information follows based on the immutability
- Let us discuss immutability and then consensus….

TYPICAL BLOCK"CHAINED" TRANSACTION DATA



- A hashed chain of blocks is constructed for immutability
- Each Block can contain multiple transactions
- Blocks are created based on an expiration of a max time duration or based on a max number of transactions to include in a block
- Transaction data can include any information (goes beyond cryptocurrencies)

A Merkle Tree of transactions in each block



THE BYZANTINE GENERALS PROBLEM

- The ability of a distributed system consisting of N nodes to take correct decisions despite failures in the system
- Pairwise messaging is permitted between the N nodes
- Failures can be fail-stop nodes, or nodes behaving incorrectly and differently to different observers, or nodes deliberately behaving maliciously
- The goal is to reach a consensus despite such concerns
- How many number of faults F can such a system tolerate?

THE BYZANTINE GENERALS PROBLEM

- As stated in the paper by Lamport, Shostak, and Pease in 1982
- A commanding general must send an order to his (N – 1) lieutenant generals such that
  a) All loyal lieutenants obey the same order
  b) If the commanding general is loyal, each loyal lieutenant obeys the order that he sends

  - If the commanding general is loyal, then (b) follows from (a)
  - However the commanding general need not be loyal
    - He can send different messages to different lieutenants

THE BYZANTINE GENERALS PROBLEM

If there are N nodes, is majority voting sufficient in the system?

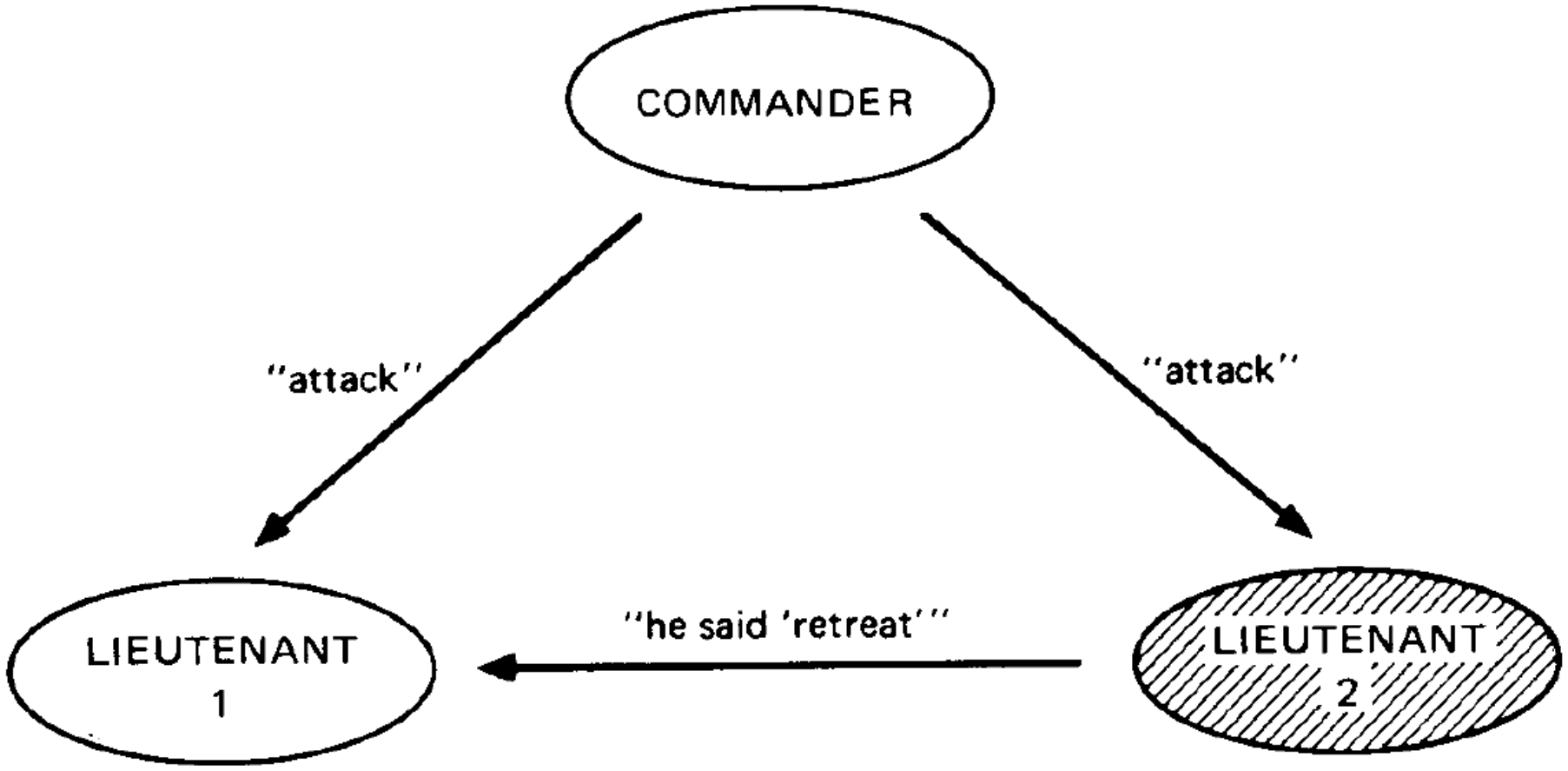


Fig. 1. Lieutenant 2 a traitor.

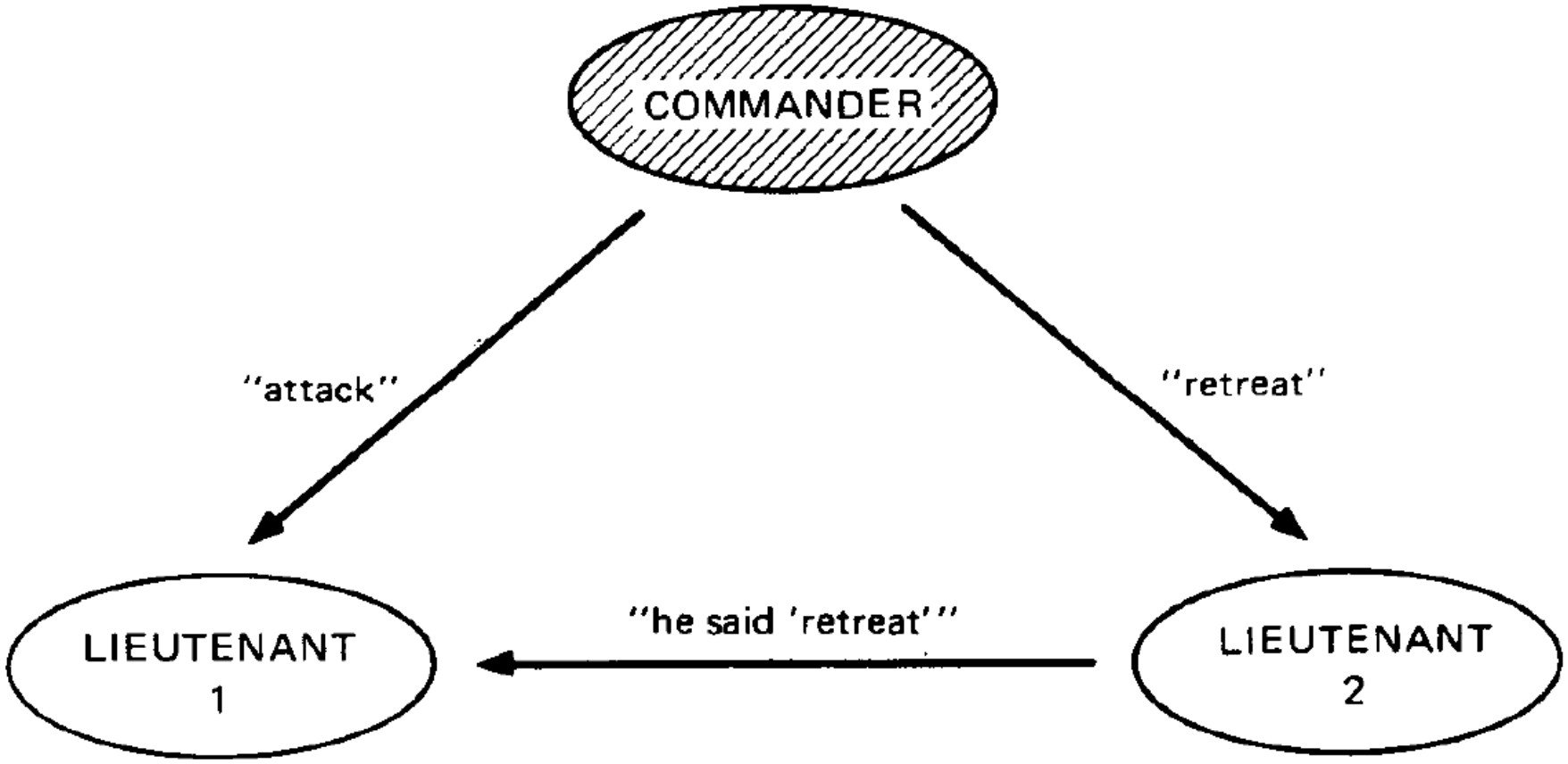
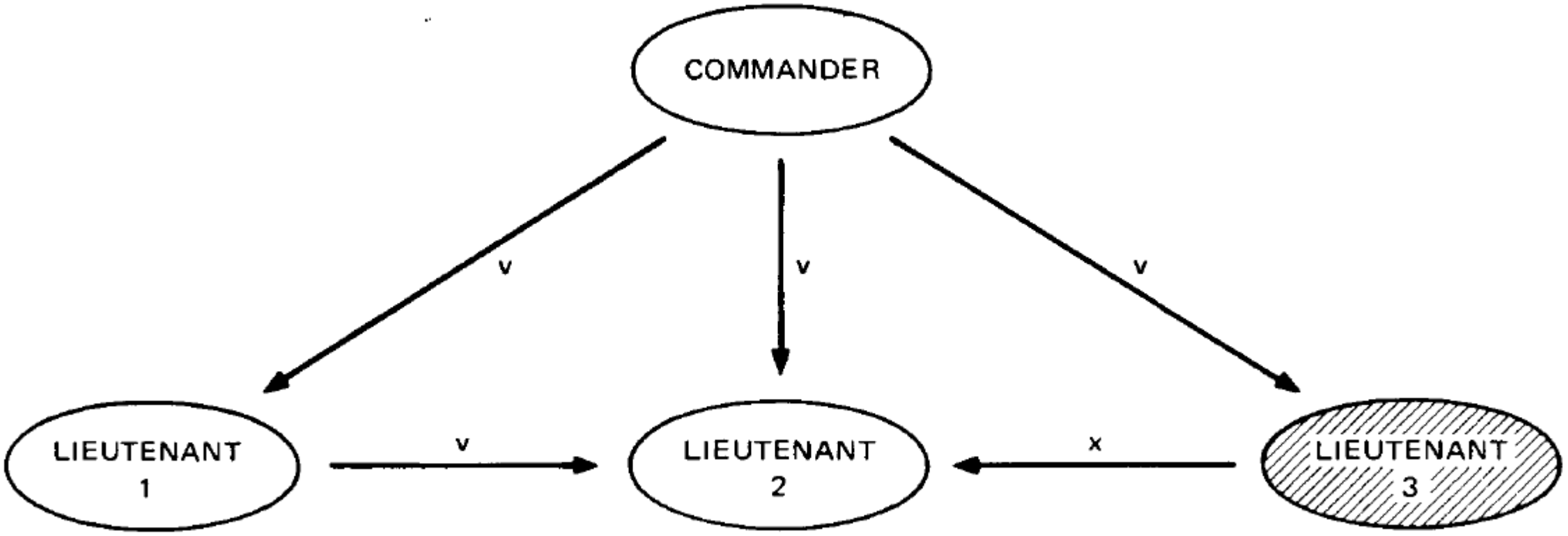
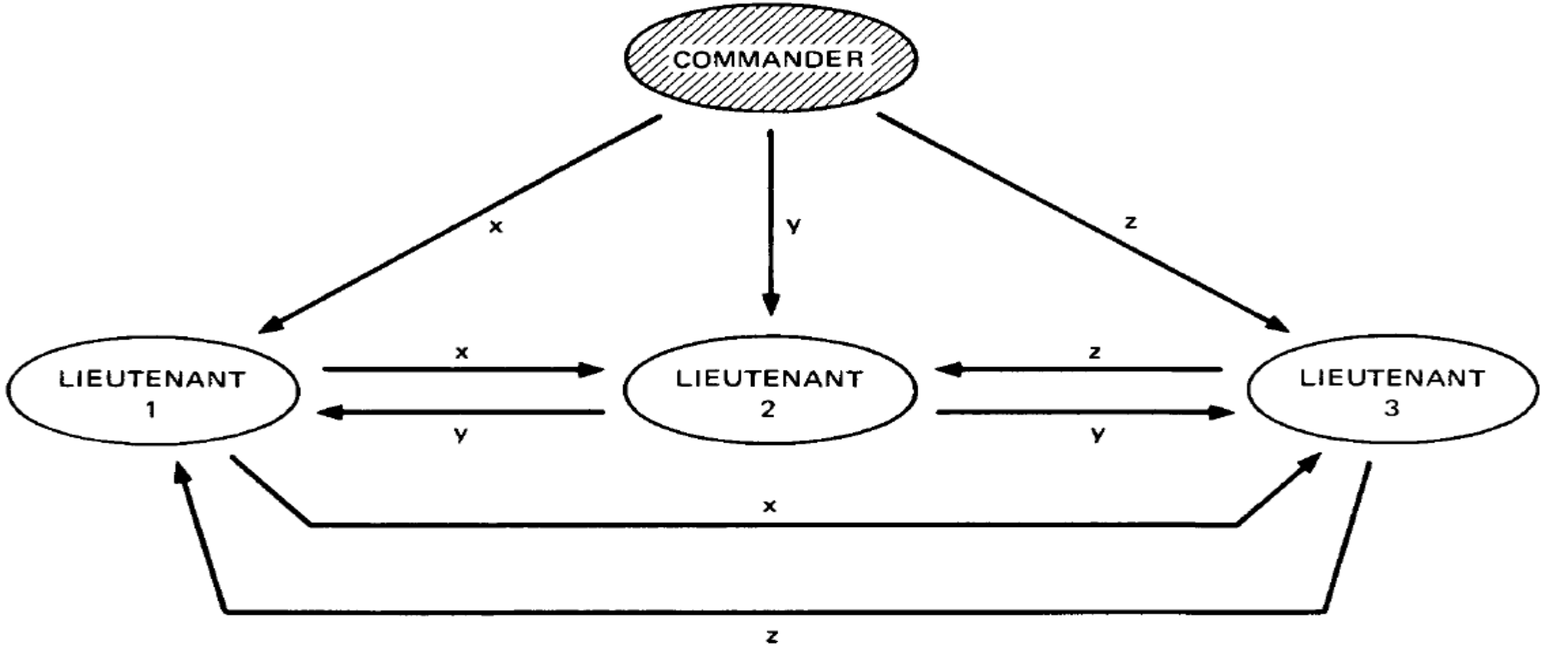


Fig. 2. The commander a traitor.

THE BYZANTINE GENERALS PROBLEM

- So 3 nodes are insufficient to handle 1 traitor or fault
- Will 4 nodes suffice?


COMMANDER
v
v
v
LIEUTENANT 1
v
LIEUTENANT 2
x
LIEUTENANT 3



All Lieutenants receive the same majority (x, y, z) and take the correct decision regardless of which of x, y, and z are equal to each other.

THE BYZANTINE GENERALS PROBLEM

- We need 4 nodes to handle 1 fault
- A system of N nodes can handle F faults, if $N \geq 3F + 1$
- Proof by contradiction
  - Assume that N = 3F can handle F faults
  - This is reduced to a network of 3 nodes that can handle 1 fault --- a contradiction

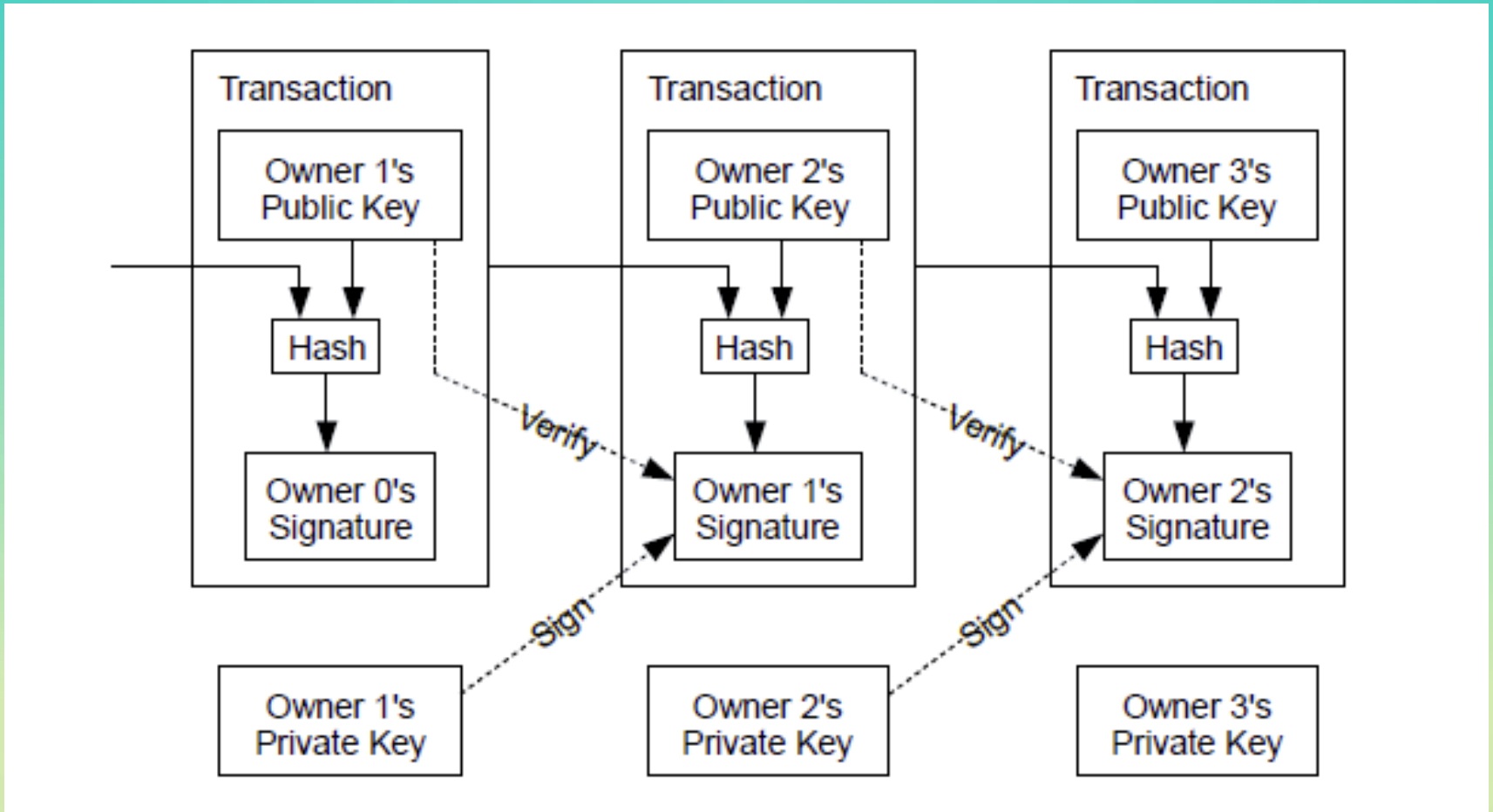
  - For more details, please see Lamport, Shostak, and Pease, "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382-401.

PRACTICAL BYZANTINE FAULT TOLERANCE

- PBFT -> Application of BFT to practical systems
  - A client sends a request to the leader node to invoke a service operation.
  - The leader node multicasts the request to (n -1) backup nodes.
  - The nodes execute the request and then send a reply to the client.
  - The client awaits f + 1 (f represents the maximum number of nodes that may be faulty where n = 3f + 1) replies from different nodes with the same result. This result is the result of the operation.

- Please see Miguel Castro and Barbara Liskov, “Practical Byzantine Fault Tolerance,” Operating Systems Design and Implementation (OSDI), 1999.

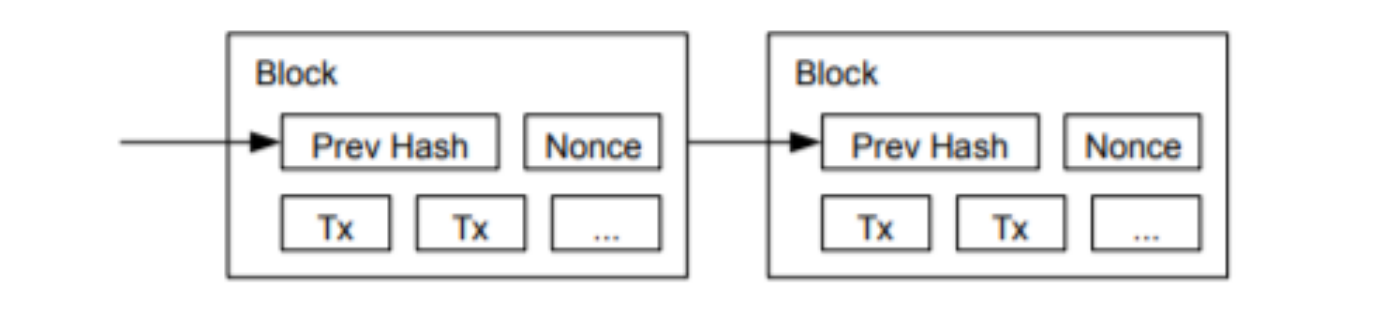
THE BITCOIN BLOCKCHAIN



From https://bitcoin.org/bitcoin.pdf

PROOF OF WORK?

- The system challenges nodes in the system to create a proof
- The work required to create the proof should not be easy



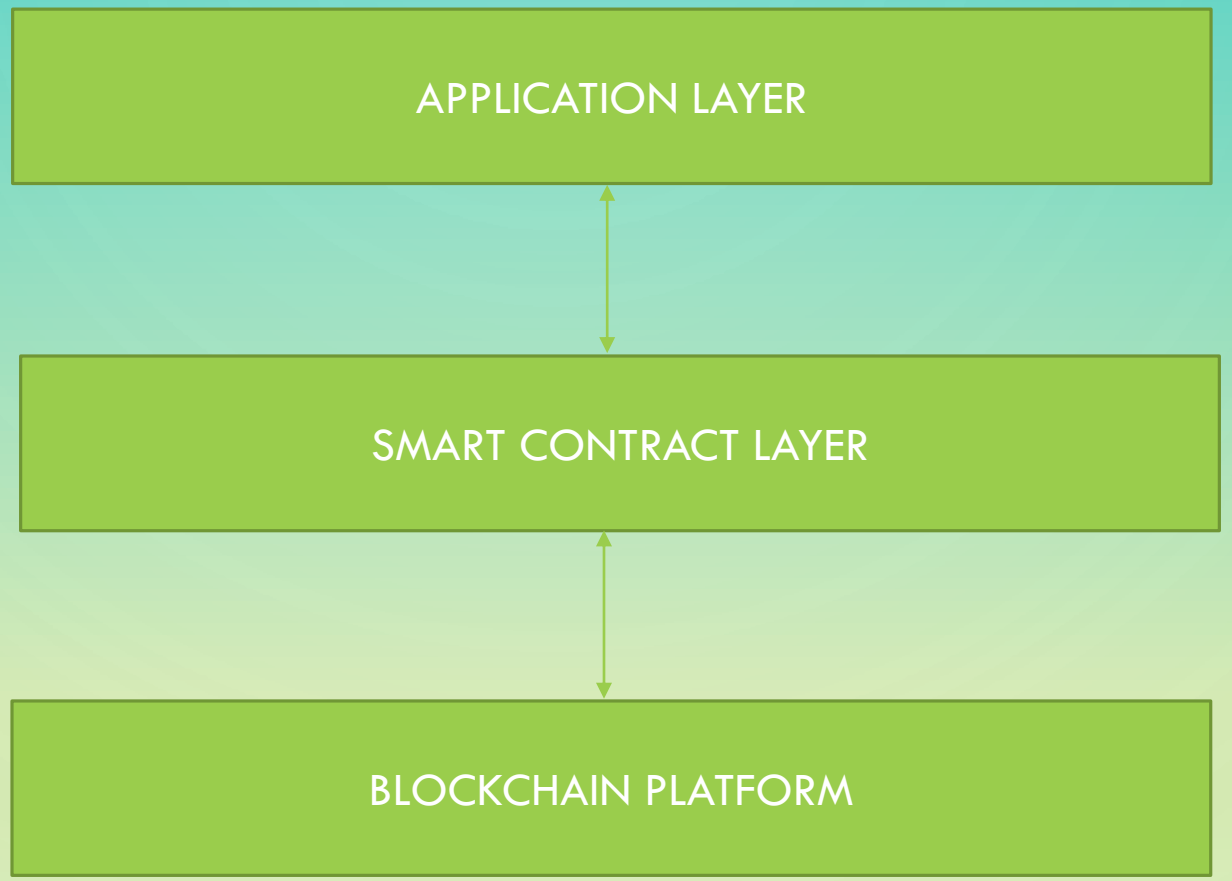
- Proofs can vary
  - Add a nonce to the data to produce a hash with X number of leading number of zeros - X can be increased to increase complexity exponentially
  - Add a nonce to the data to produce a hash such that the hash is less than a certain threshold value

OTHER CONSENSUS ALGORITHMS?

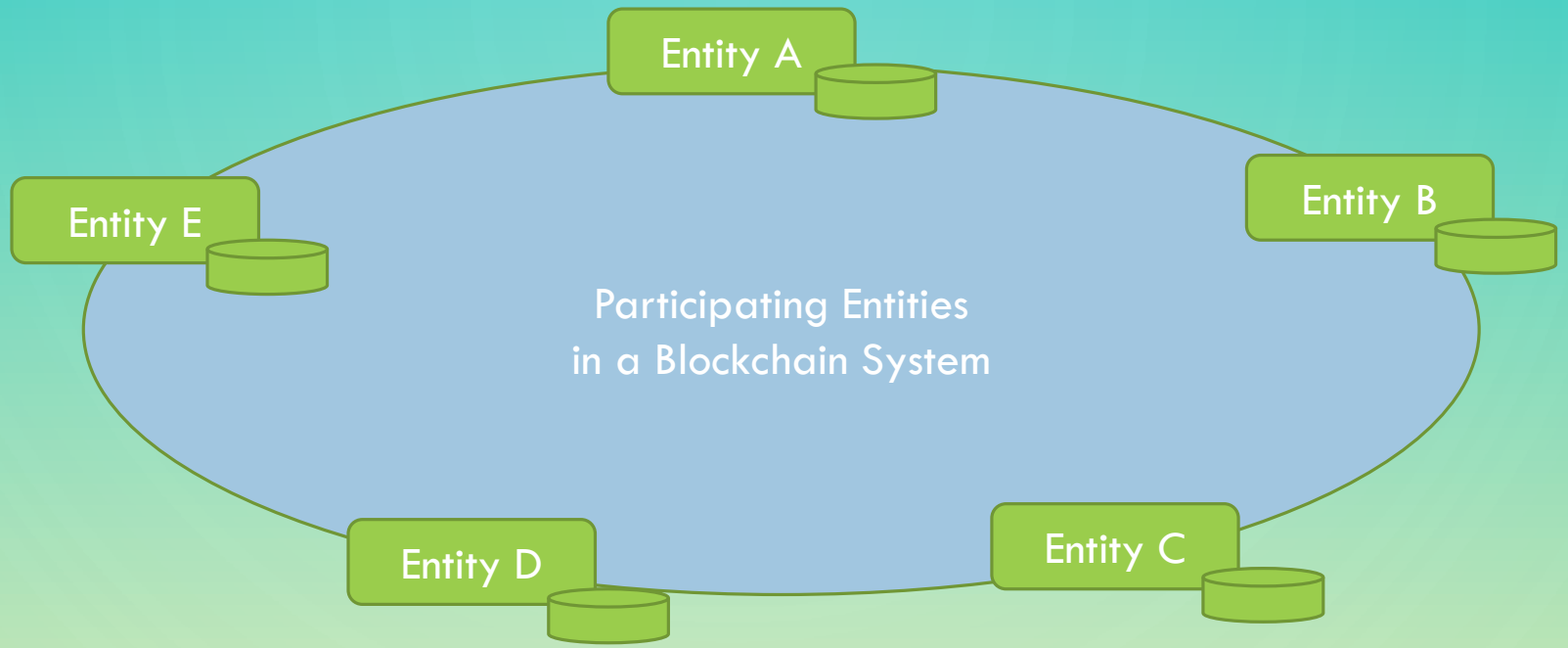
The choice of a consensus algorithm depends on the nature of the problem – one algo will not fit all use-cases. Some practical techniques may be adequate in many situations. Some may be more expensive or more biased than others. It is good to have pluggable consensus on a blockchain platform.

- Proof of Work
  - Creation of proof is very expensive and this makes it very difficult to tamper with the blockchain to create a new longest chain
  - Significant energy wasted in the process
- Majority Voting : > 50%
- All in agreement : 100%
- Percentage of Votes : > X%
- Proof of Stake
  - Creator of the next block is selected based on its stake in the system (ownership, age, wealth, etc.)
  - Can be biased
- Delegated Proof of Stake
  - Other nodes provide nominations – can reduce bias – the stake here is level of trust obtained based on the votes
- Raft Consensus
  - Randomized Leader election with majority voting

TYPICAL SOFTWARE LAYERING IN A BLOCKCHAIN SYSTEM



SMART CONTRACTS



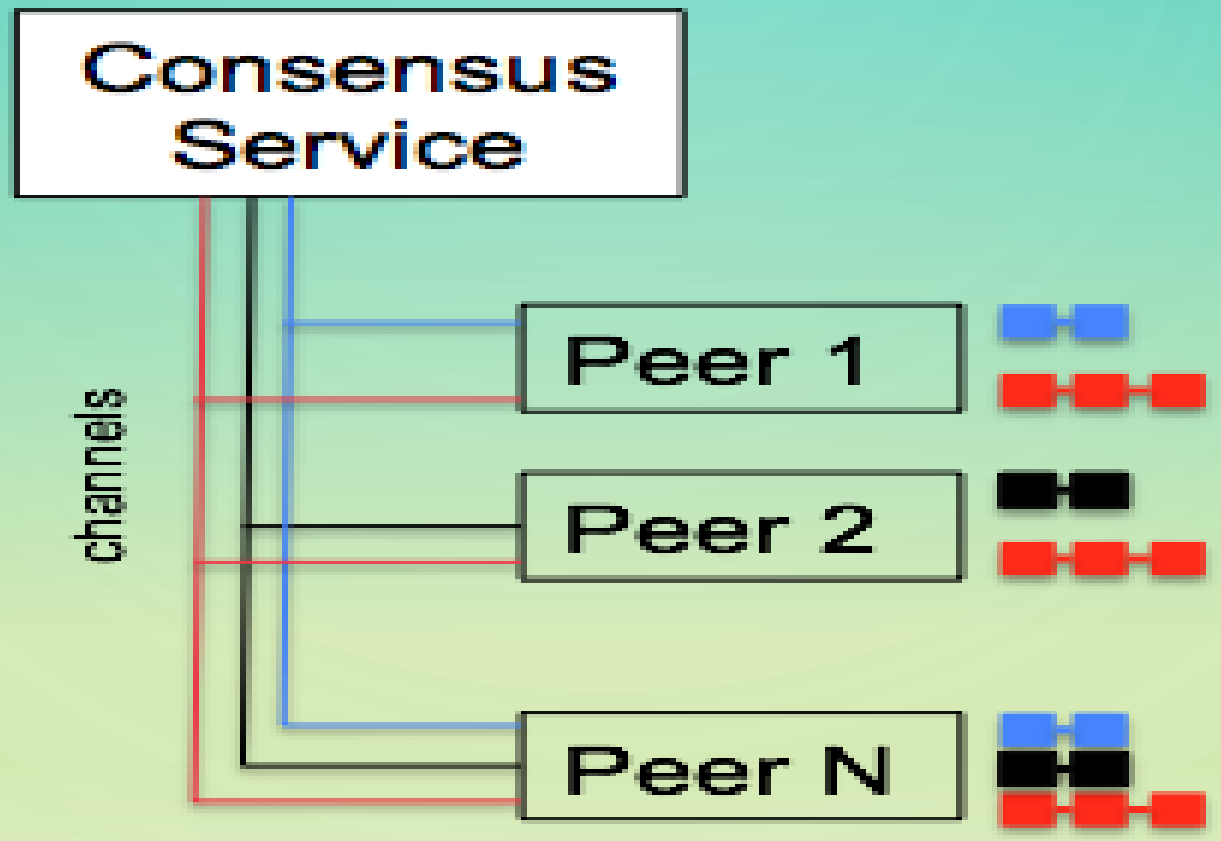
Smart Contracts are the shared programs that update the shared ledgers in a blockchain system

TYPICAL SMART CONTRACT CODE

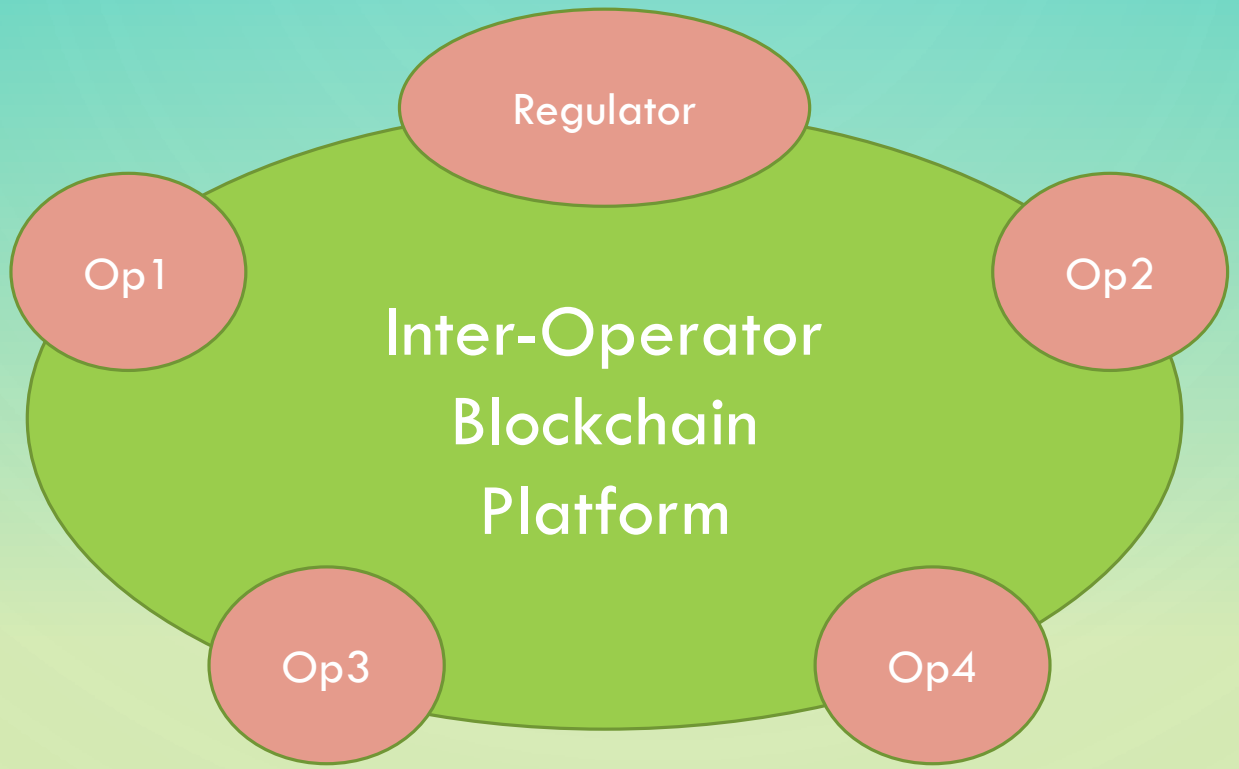
```
if age >= 90 {
    if payment == 0 {
        if upcRecord["Prohibited"] == "No“ {
            result = "Eligible"
        }
    }
}
else {
    result = "Not Eligible"
}
return result;
```

MULTIPLE CHANNELS / LEDGERS

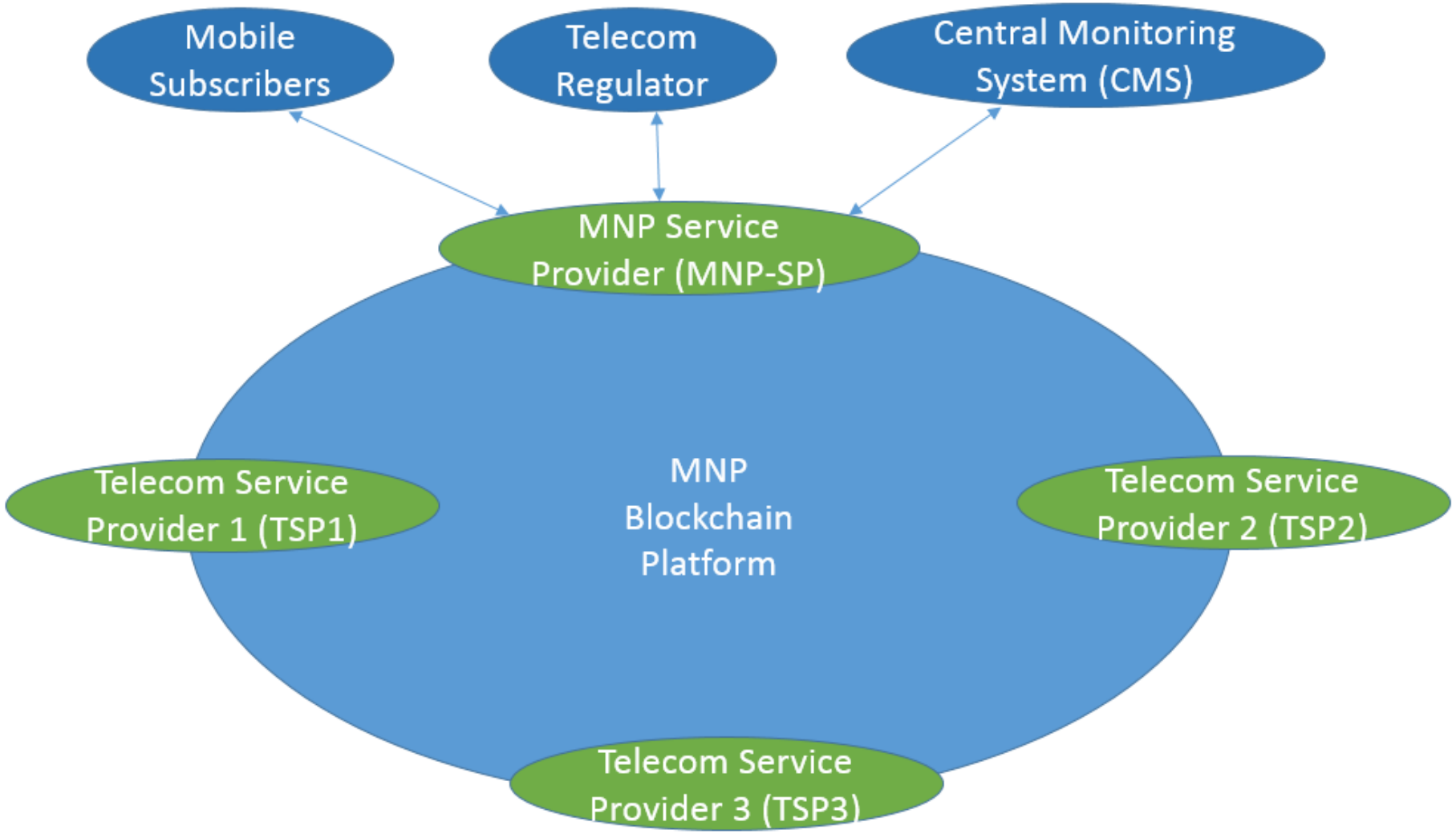
SEPARATE LEDGERS FOR SUBSETS OF PEER NODES MAINTAIN TRANSACTION PRIVACY



INTER-OPERATOR BLOCKCHAIN NETWORKS

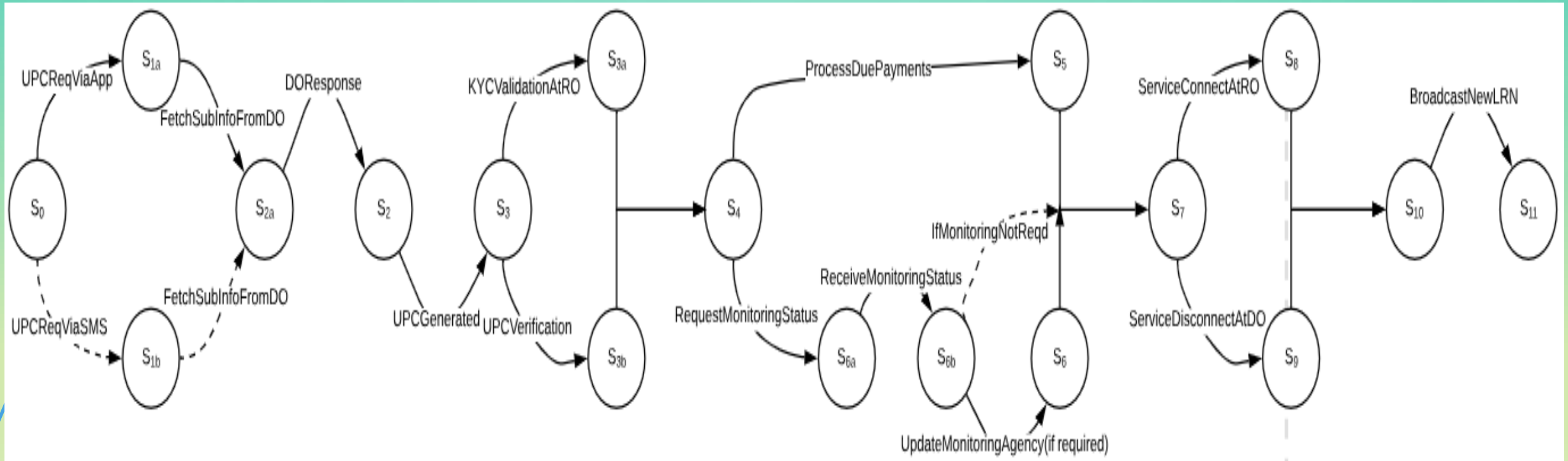


Example Blockchain System (for Mobile Number Portability)



Appeared in IEEE Intl Conf on Blockchain and Cryptocurrency, 2019, Korea
(Collaboration with K. Chauhan, A. Bhatnagar, S. Jha, S. Srivastava, D. Bhamrah, M. Prasad)

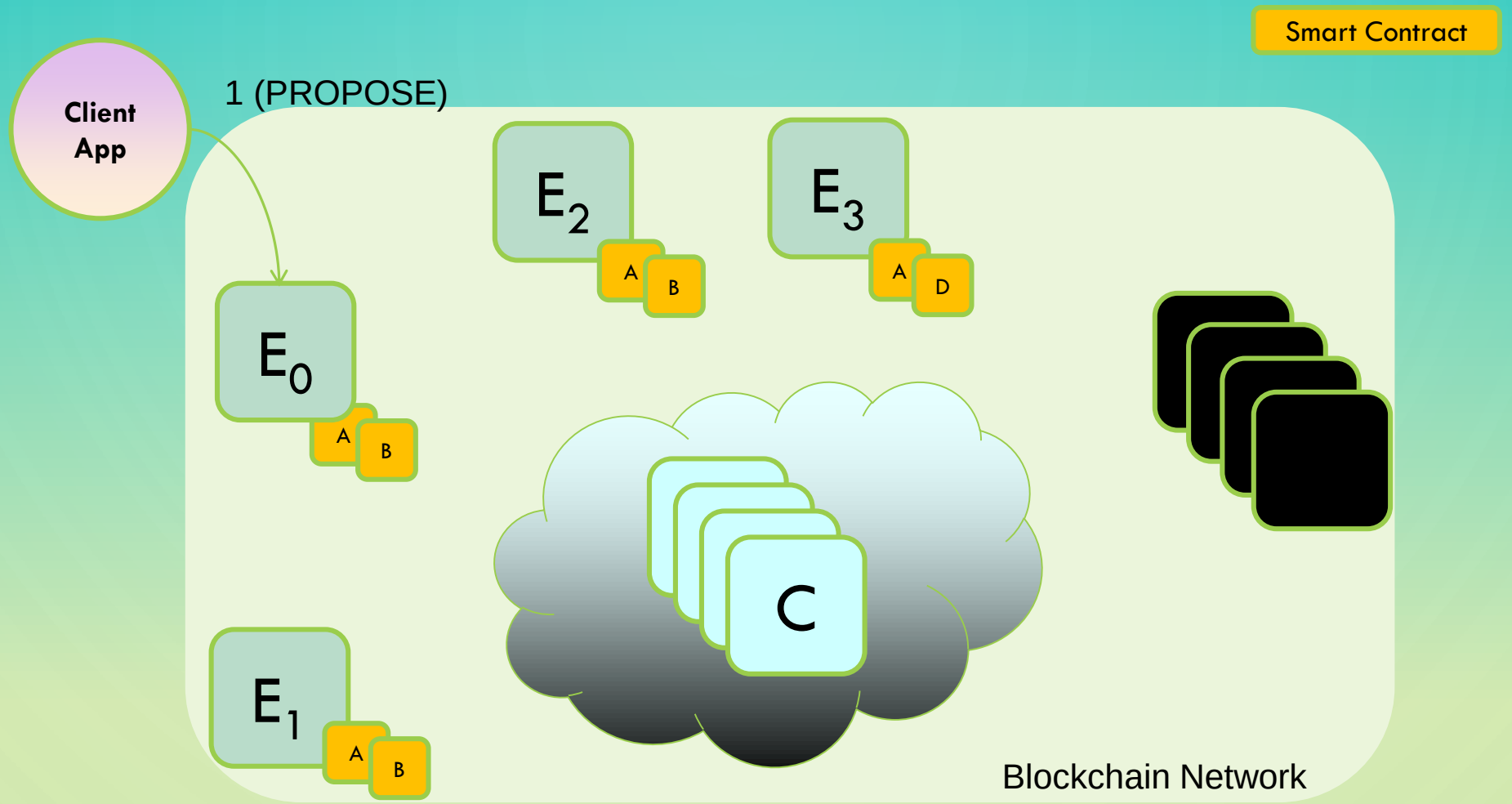
EXAMPLE BLOCKCHAIN TASK PROCESSING STATE TRANSITION DIAGRAM



PHASES IN TRANSACTION PROCESSING

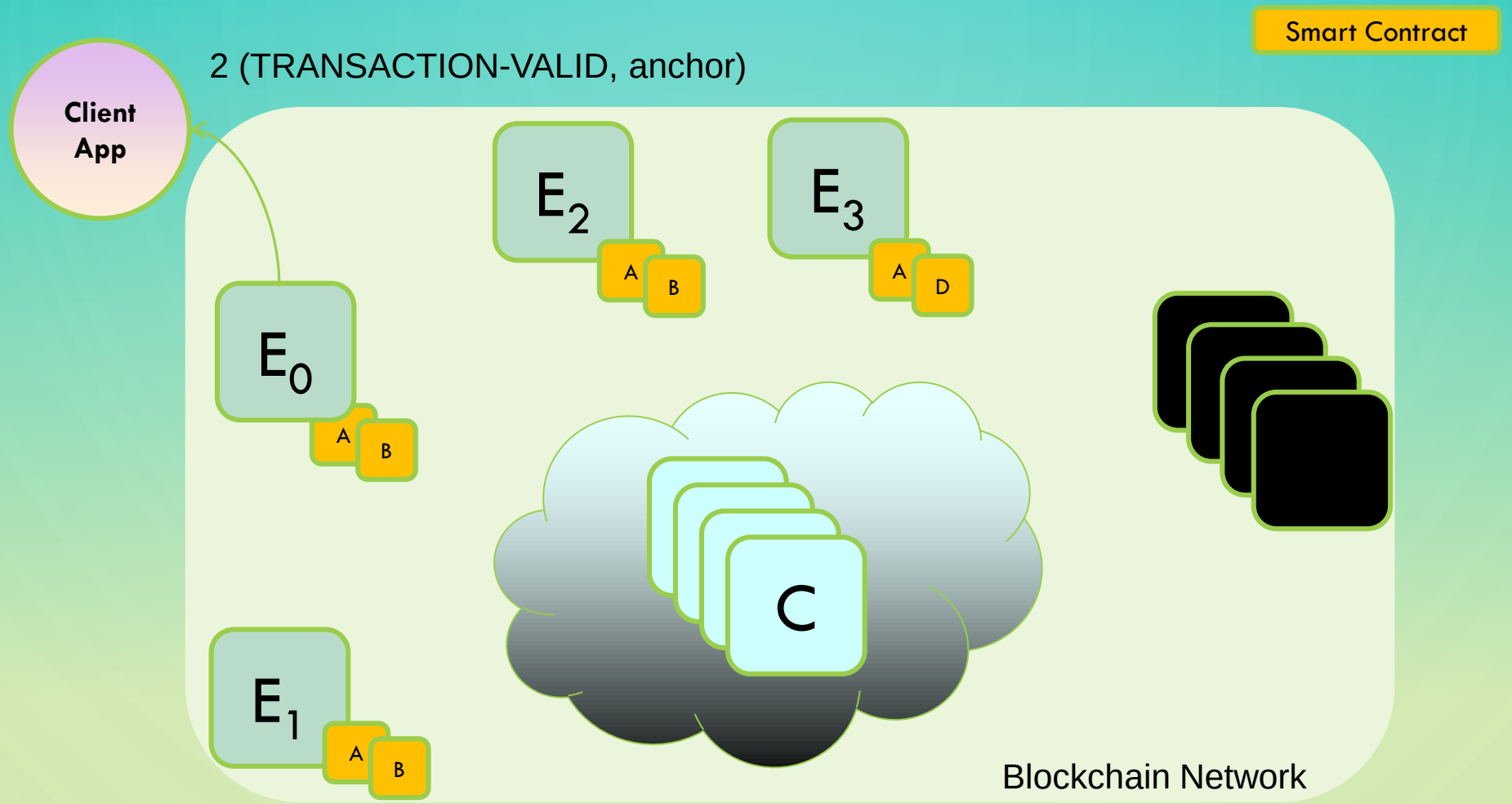
- Simulation / Endorsement Phase
  - Simulation of a Transaction by an Endorser to endorse the Transaction
  - Endorsing a transaction verifying that its content obeys a given smart contract. Endorsers “sign” the contract
- Ordering Phase
  - Neutral Ordering system to order transactions across peers
- Validation / Consenting Phase
  - Consenting the inclusion of a verified transaction in the ledger
  - Consensus controls what goes in the ledger making sure that the ledger is consistent
  - Validation System ChainCode (VSCC) processing
    - Checks endorsements relative to the endorsement policy
  - Multi Version Concurrency Control (MVCC) processing
    - Checks for read-set/write-set violations and discards invalid transactions
- Ledger Commit Phase
  - Commit valid transactions to shared ledgers

A SAMPLE TRANSACTION (1/6)



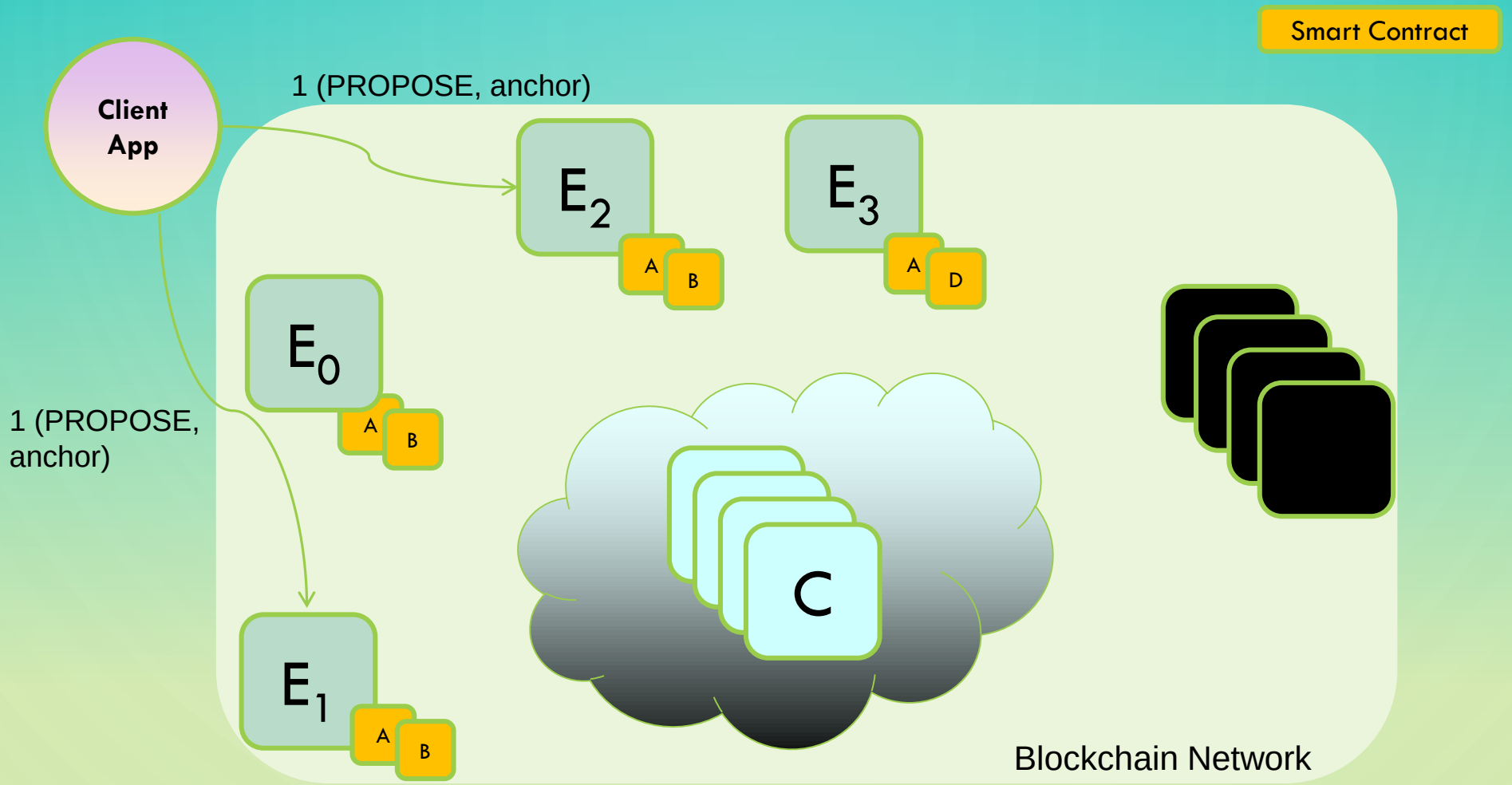
1. The Client App proposes a transaction for **Smart Contract A** to the Endorsing peer E_0. Endorsement policy: "E_0, E_1 and E_2 must sign". E_3 is not part of the policy

A SAMPLE TRANSACTION (2/6)



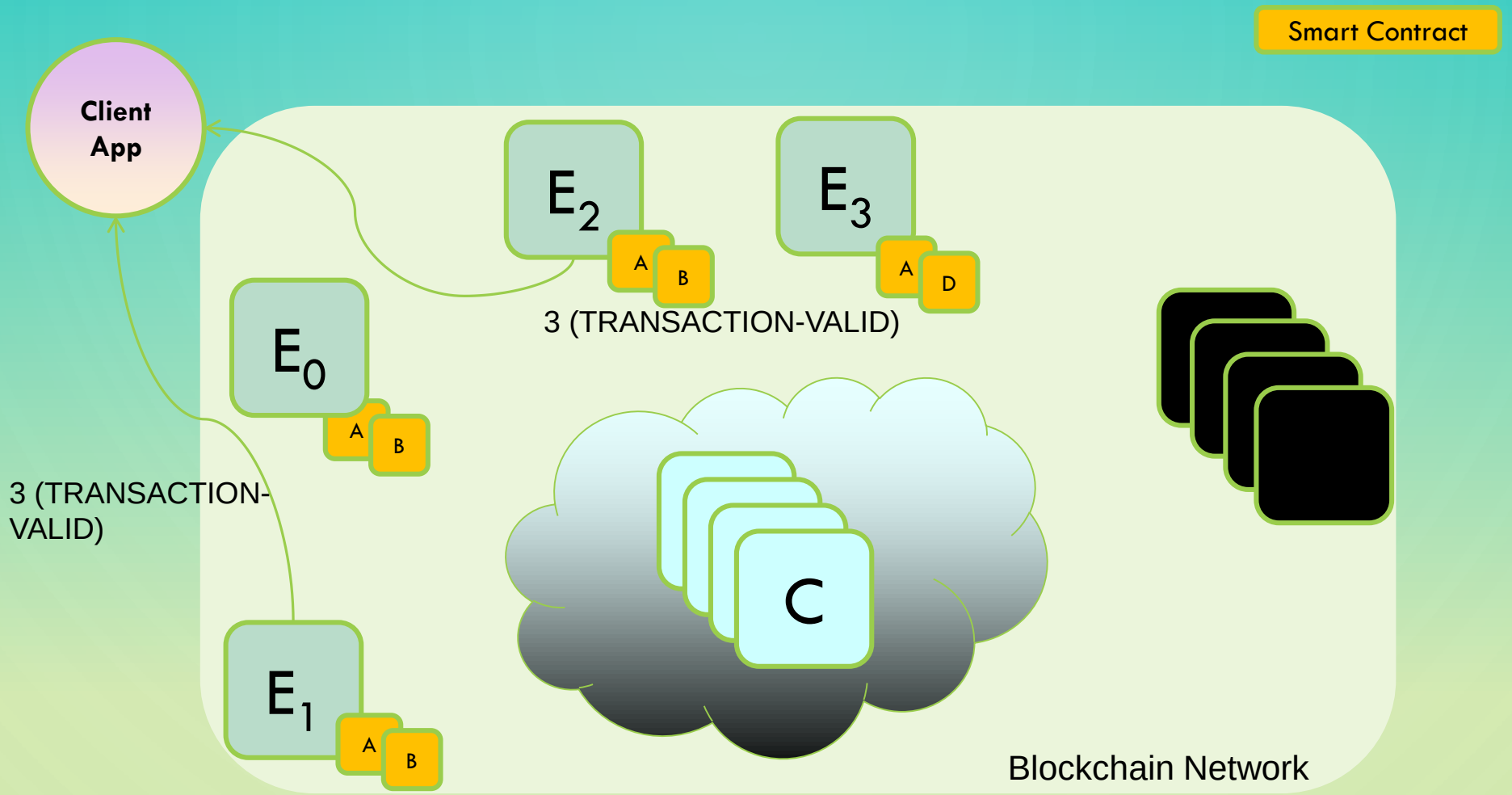
2. Endorsing peer E_0 endorses a tx and (optionally) “anchors it” with respect to the ledger state version numbers. An “anchor” contains all data read and written by contract that are to be confirmed by other endorsers.

A SAMPLE TRANSACTION (3/6)



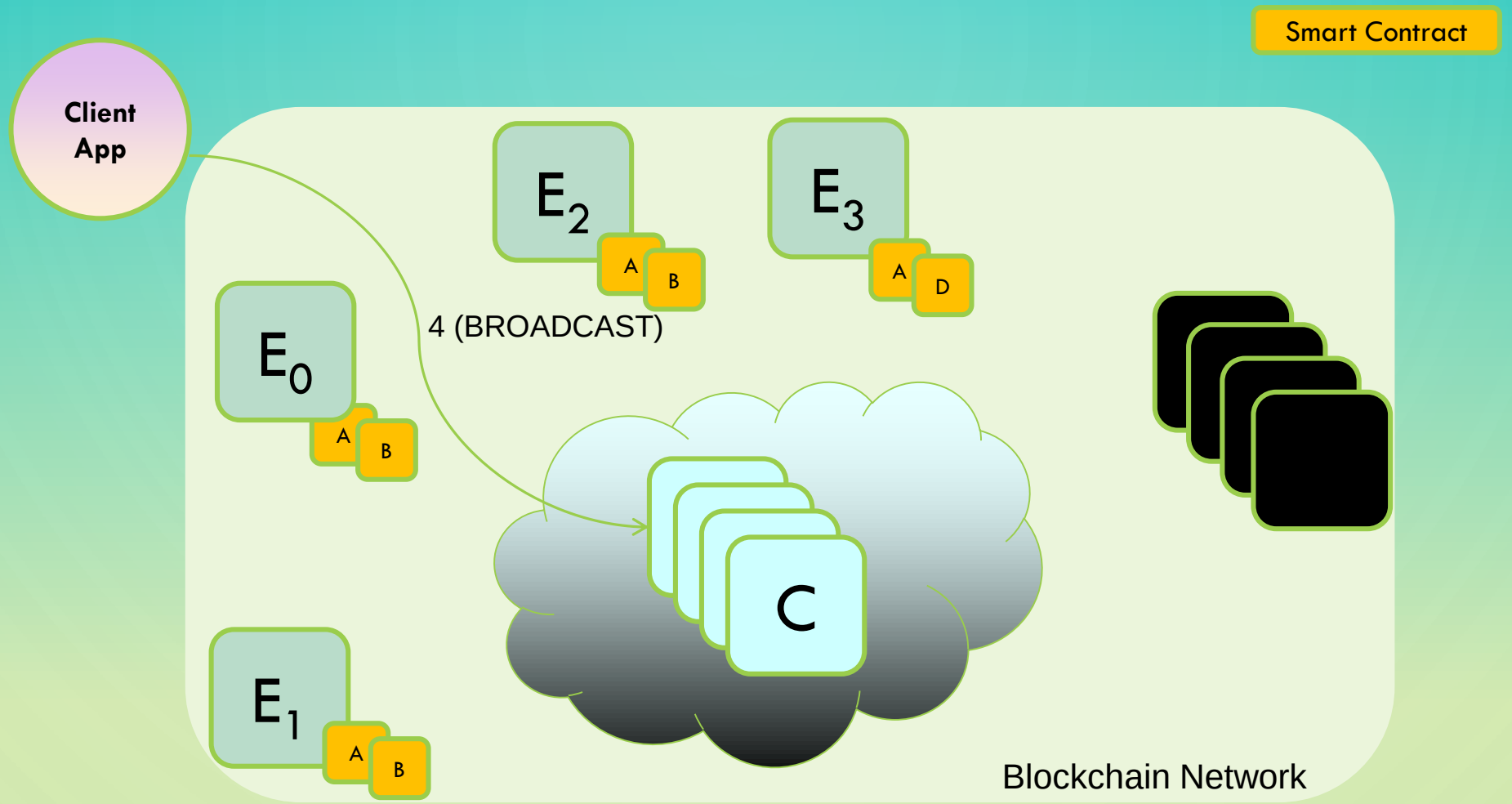
3. The client requests further endorsement from E_1 and E_2.The client may decide to suggest an anchor obtained from E_0. to E_1 and E_2.

A SAMPLE TRANSACTION (4/6)



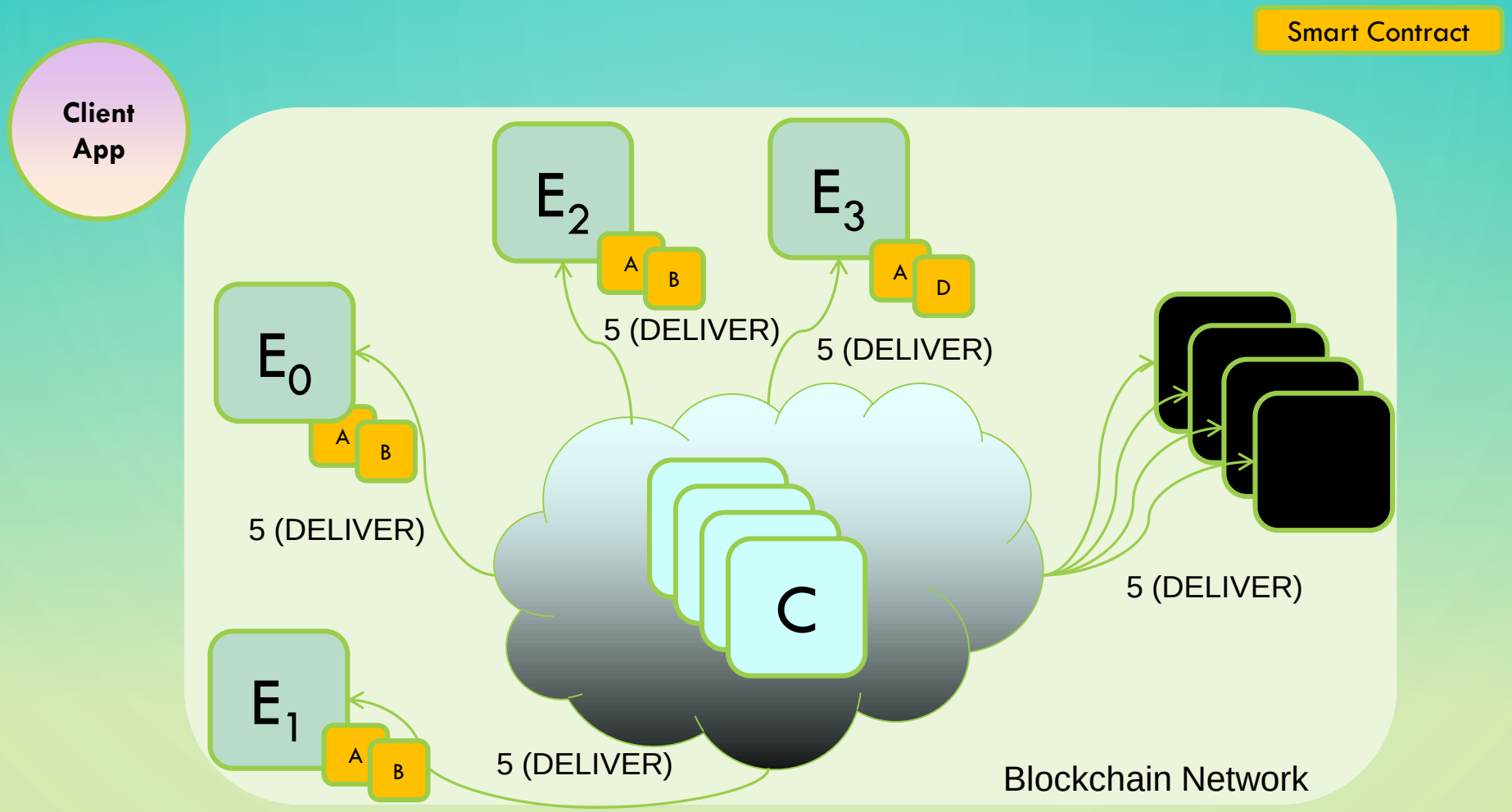
4. The Endorsing peers E_1 and E_2 send the endorsement to client.

A SAMPLE TRANSACTION (5/6)



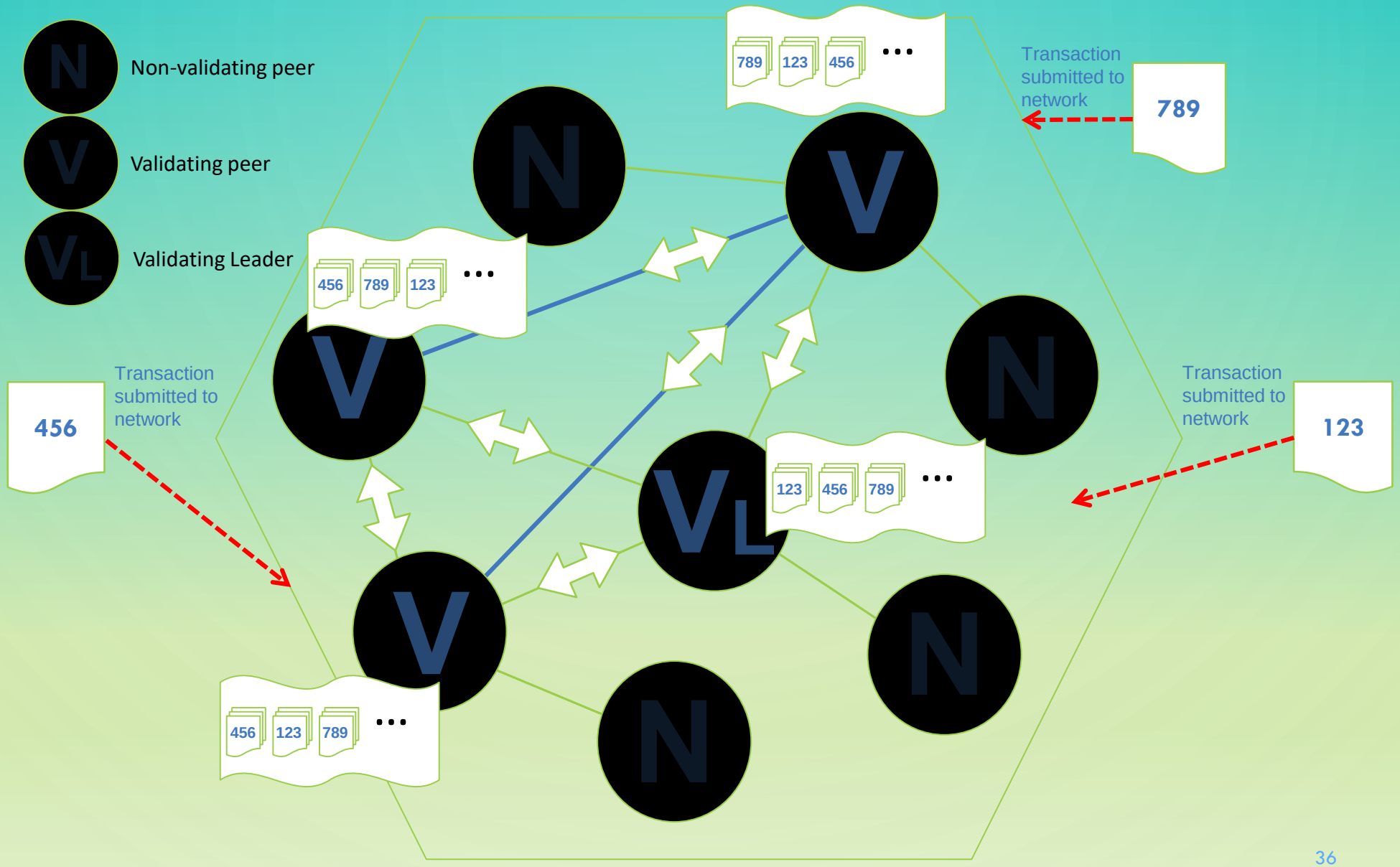
5. Client formats the transaction and broadcasts it to the consenters for inclusion in the ledger

A SAMPLE TRANSACTION (6/6)

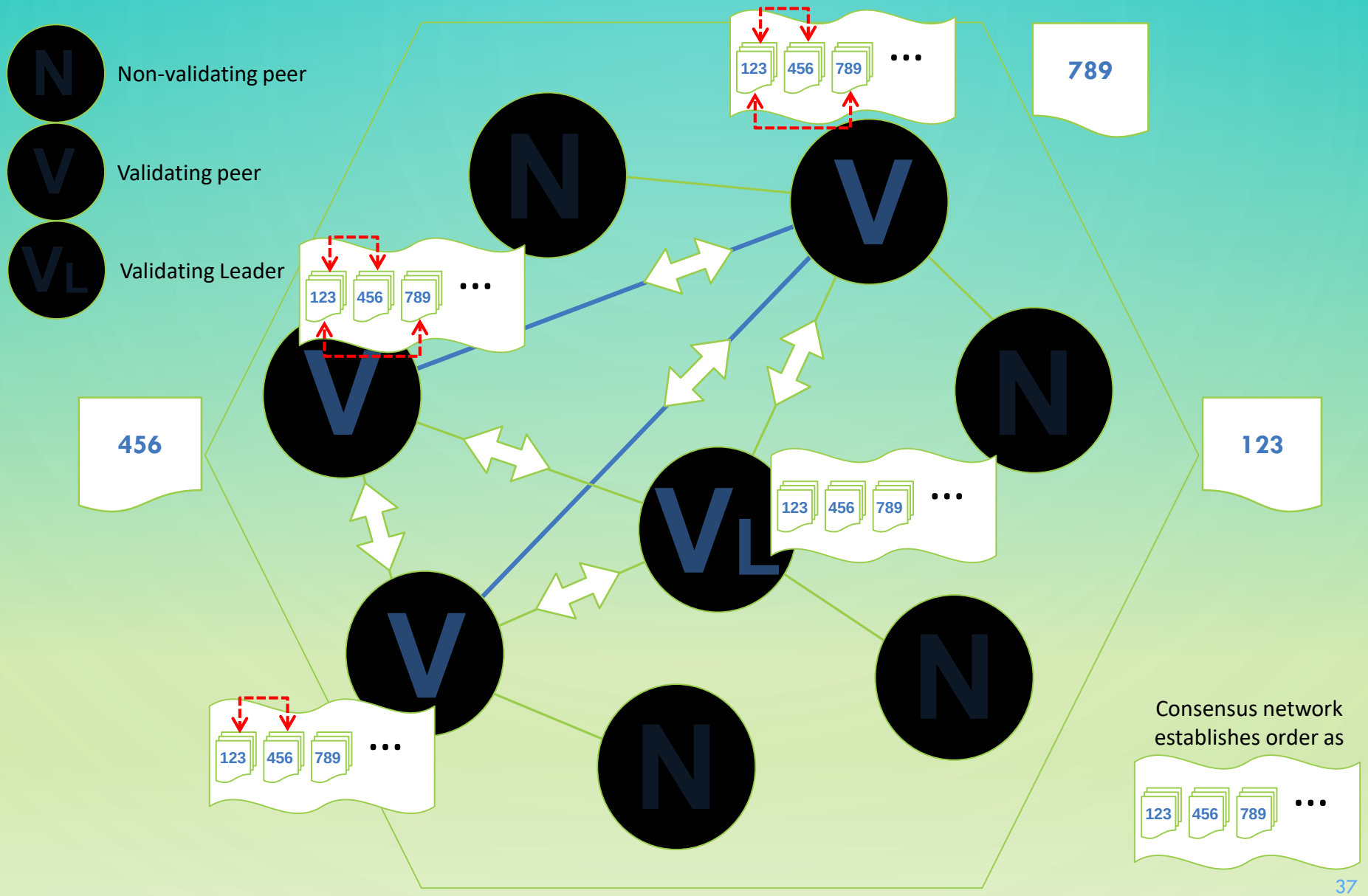


6. The consensus service delivers the next block in the ledger with the consented transaction.

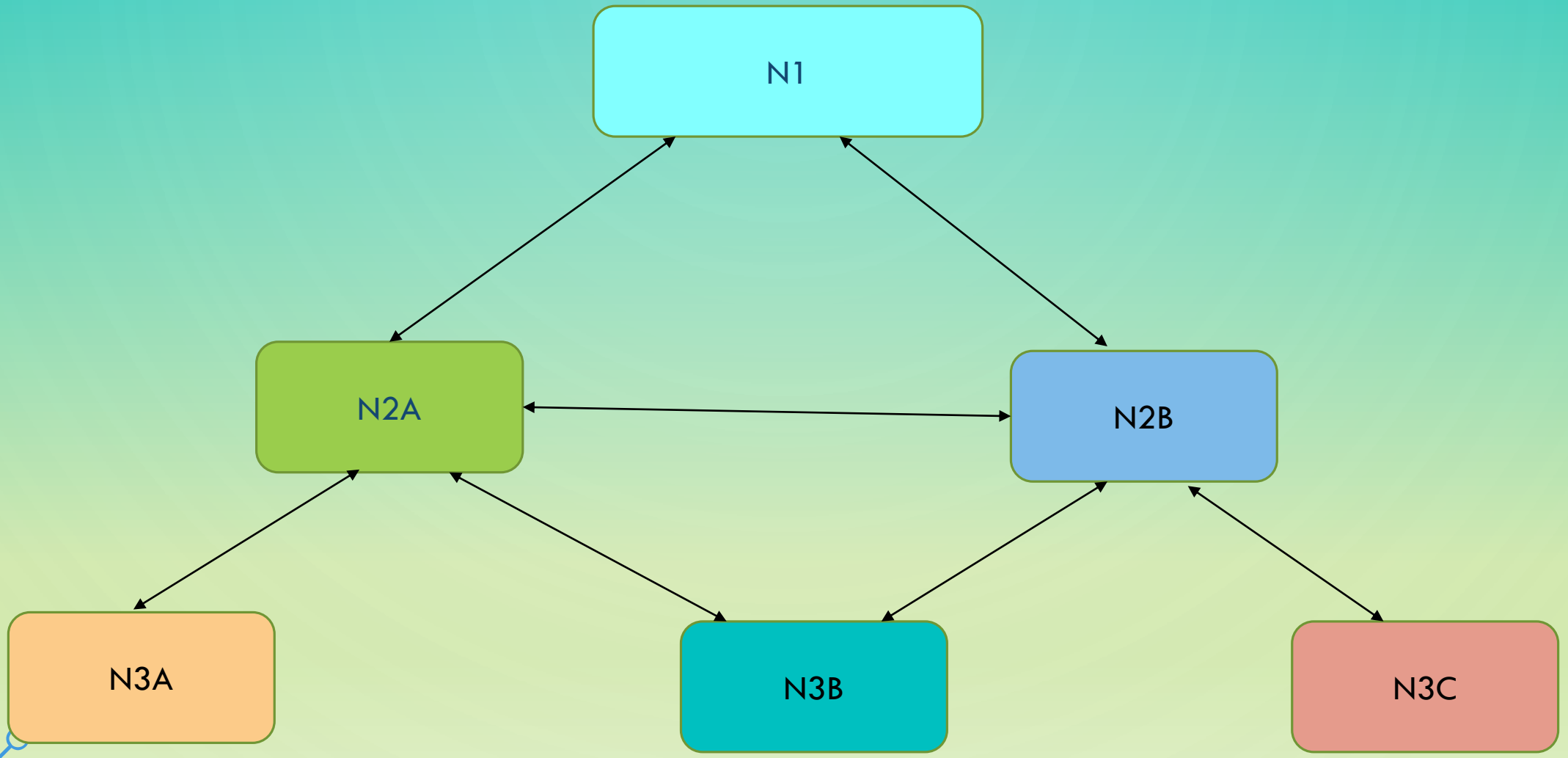
TYPICAL TRANSACTION FLOW ORDERING (1)



TYPICAL TRANSACTION FLOW ORDERING (2)



HIERARCHICAL AND DISTRIBUTED SMART PROCESSING IN EMERGING 5G DATA NETWORKS



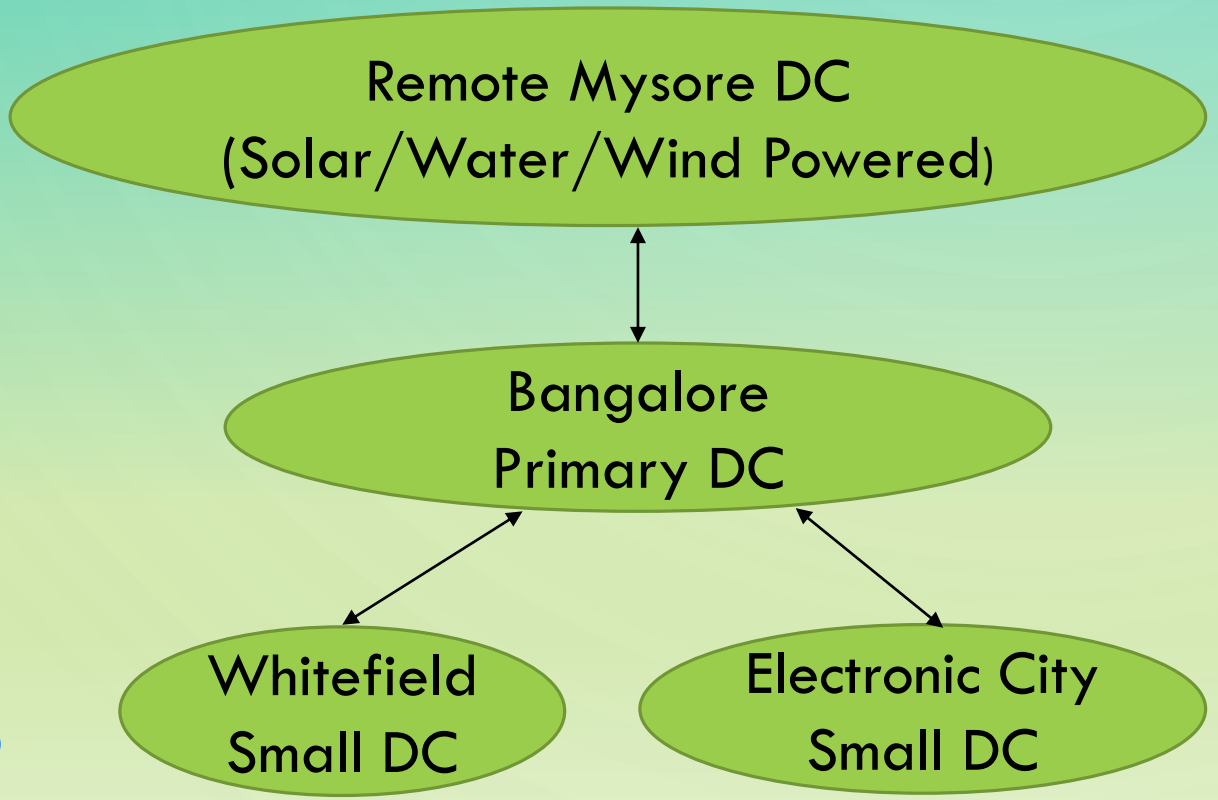
DISTRIBUTED PROCESSING ACROSS DISTRIBUTED CLOUD & EDGE DATA CENTERS

Map users to VNFs on DCs based on latency, availability, DC load, energy, mobility

Dynamically direct new user flows to utilize VNFs at the most appropriate DCs

Dynamically divert new VNF resource requirements away from a DC if heavily loaded

Example Depiction – Serving users in Whitefield and Electronic City



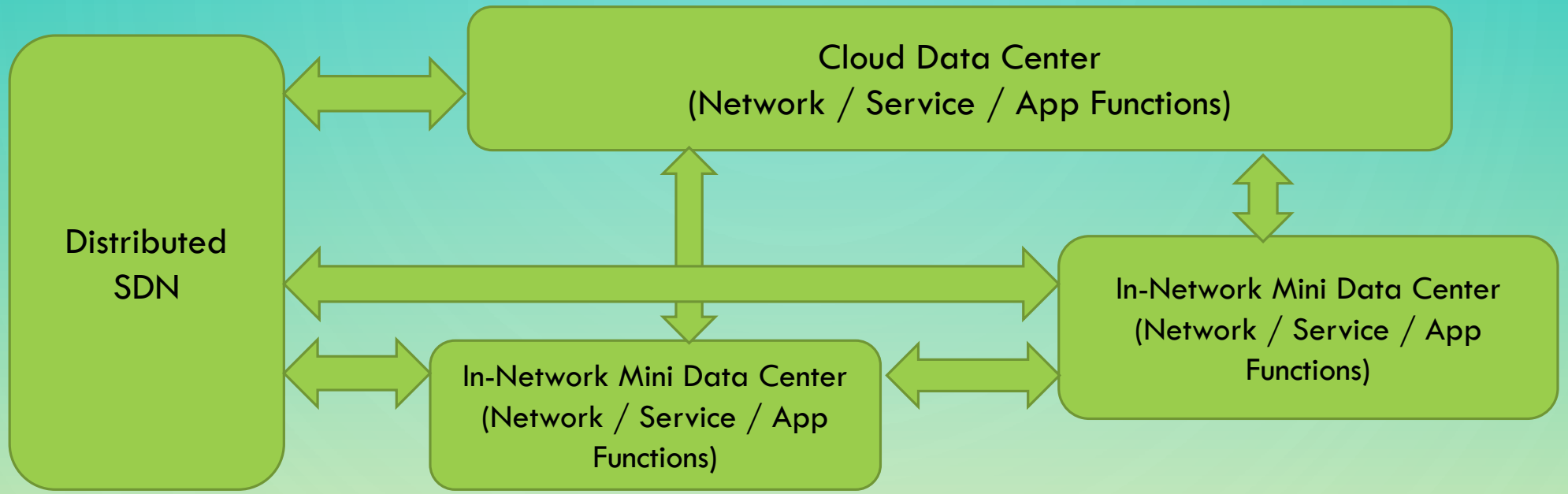
Higher Latency, Low Energy cost, High Capacity

Medium Latency, High Energy cost, Higher Capacity compared to Small DC

Lower Latency, High Energy Cost, Low Capacity,

DISTRIBUTED SDN WITH DISTRIBUTED NFV

Distributed SDN and NFV will enable smart distributed processing of functions across data centers



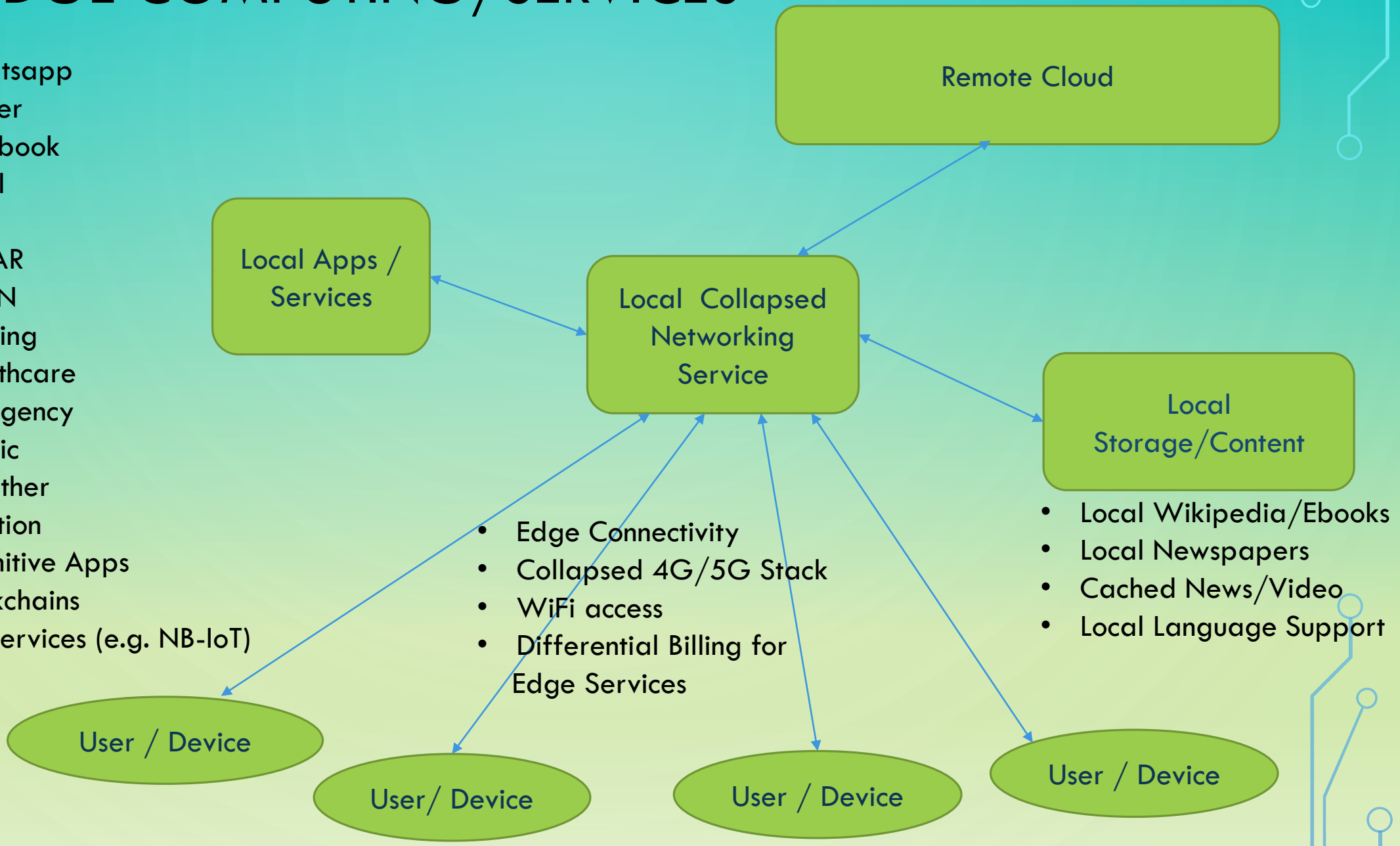
→ *Partition / Collapse / Replicate functions across data centers*

→ *Address latency constraints, user mobility, dynamic resource availability, security (compute/network/storage/energy)*

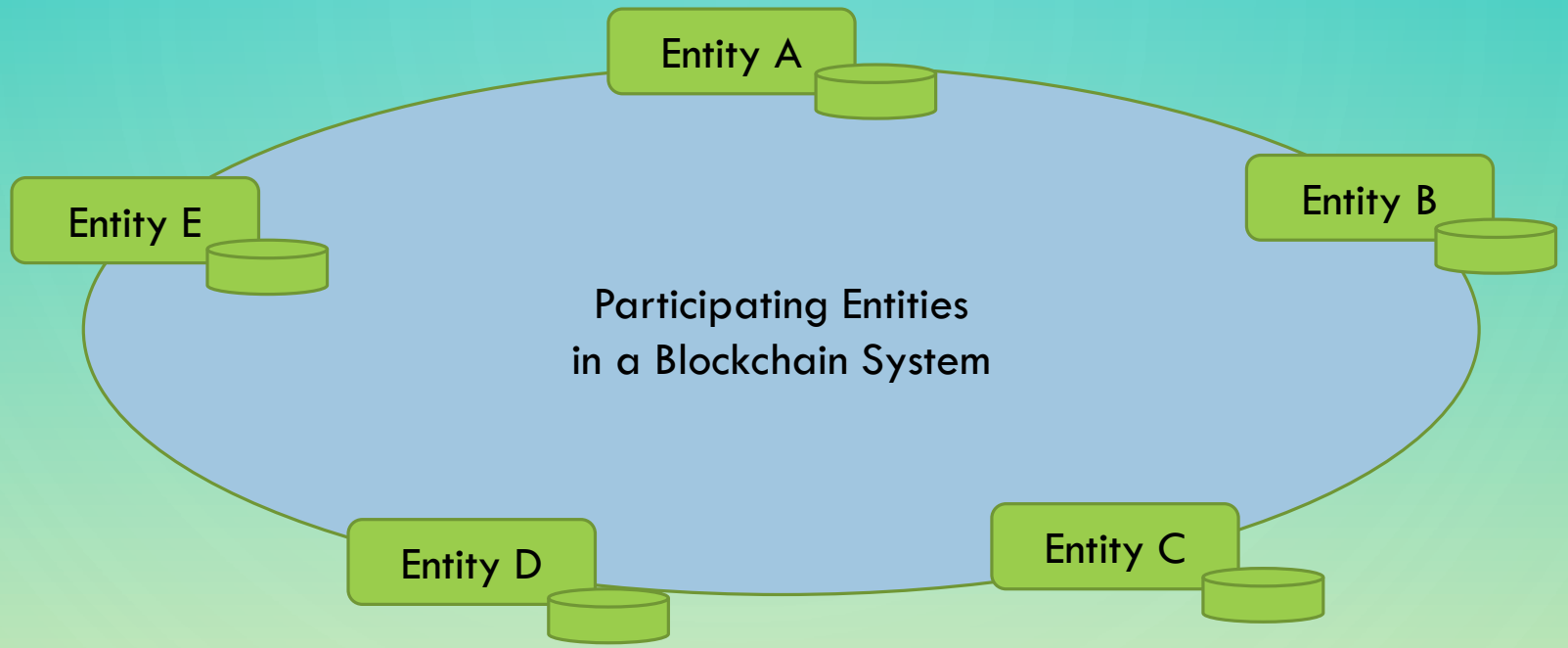
→ *Dynamic Monitoring, Analytics, Optimization, Orchestration, Scaling*

EDGE COMPUTING/SERVICES

- Local Whatsapp
- Local Twitter
- Local Facebook
- Local Email
- Local Web
- Local VR/AR
- Local VCDN
- Local Banking
- Local Healthcare
- Local Emergency
- Local Traffic
- Local Weather
- Local Pollution
- Local Cognitive Apps
- Local Blockchains
- Local IoT Services (e.g. NB-IoT)



BLOCKCHAIN SYSTEMS

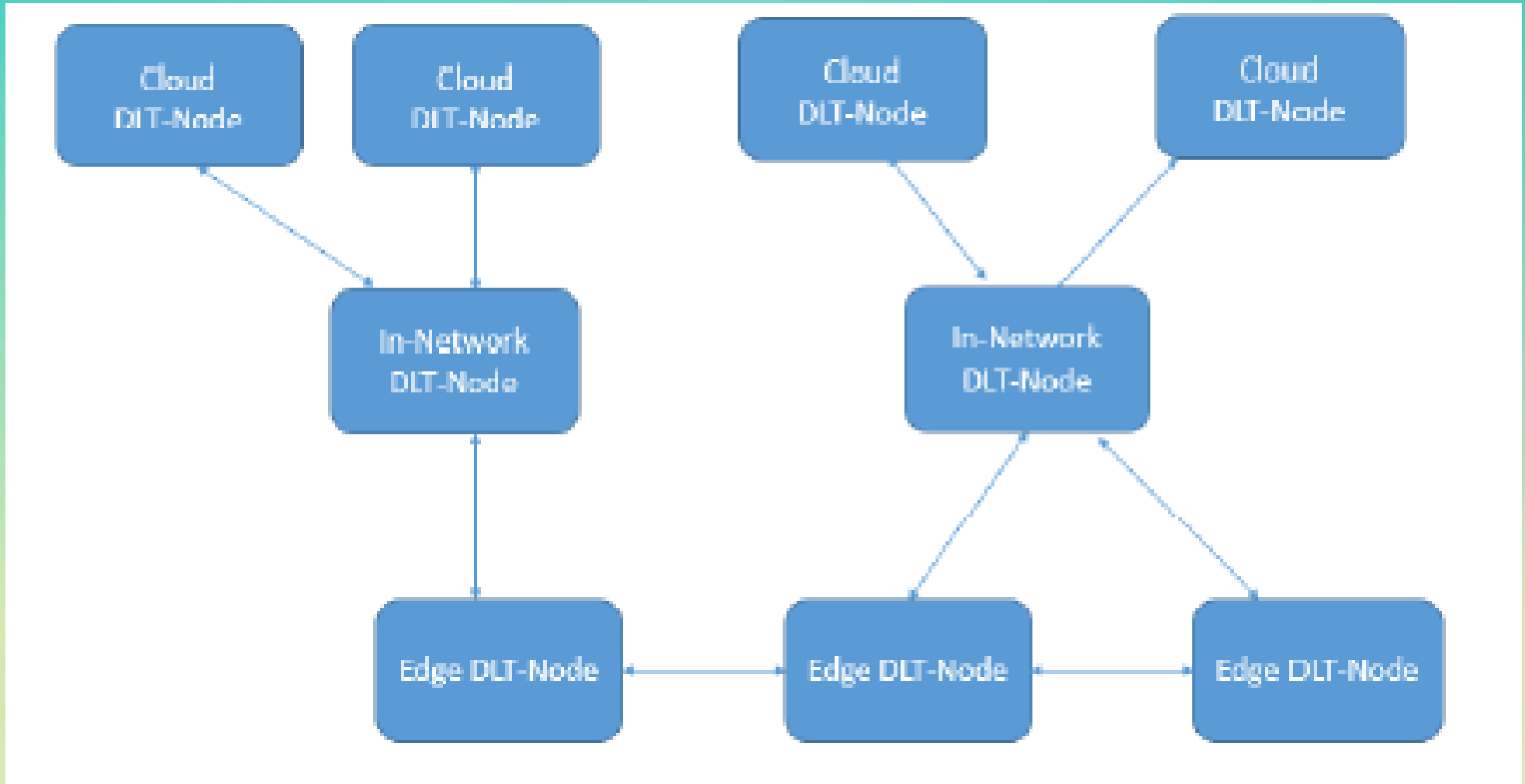


Typical systems have private programs updating private ledgers

Blockchain systems have shared programs updating shared ledgers

Trusted Data, Immutability, Consensus, Provenance, Smart Contracts

HIERARCHICAL AND DISTRIBUTED BLOCKCHAIN/DLT PROCESSING IN EMERGING NETWORKS



EXAMPLE PROPAGATION DELAYS

- Distance between data centers d : 1200 km
- Transaction size: 4 kB
- Number of Transactions per block: 500
- Blocksize: 4kB x 500 = 2 MB
- Light speed $c = 3 \text{ x } 10^8 m/s.$
- Light speed latency = d/c = 4 ms
- Efficiency of transmission in optical fibers = 69%
- Expected propagation delay α_{ij} = 4/0.69 = 5.79 ms

COMMUNICATION COSTS

- Propagation Delays + Serialization Delays
- One-way communication cost

$$\omega_{ij} = \alpha_{ij} + \beta_{ij} L$$

- Two-way communication cost

$$\rho_{ij} = 2\alpha_{ij} + \beta_{ij} L,$$

INTER-DATA CENTER LATENCIES

$$\rho_{mean,trans} = 2\alpha$$

$$\rho_{mean,block} = 2\alpha + \beta L_{block}$$

$$\rho_{max,trans} = 2\alpha_{max}$$

$$\rho_{max,block} = 2\alpha_{max} + \beta L_{block}$$

EXAMPLE SERIALIZABLE DELAYS, AND OVERALL DELAYS

- For a 1 Gbps link, $\beta ij = 1\text{ns/bit}$
- Serializable delay for 2MB = 16 Mbits x 1 ns/bit = *16 ms*
- One way delay = *5.79* + 16 = *21.79 ms*
- Two way delay with acknowledgement = 2 x *5.79* + 16 = *27.58ms*
- For a 10 Gbps link , $\beta_{ij} = 0.1ns/bit$
- Serializable delay for 2MB block = *1.6ms*
- One way delay = *5.79* + 1.6 = *7.39ms*
- Two way delay with ack = 2 x *5.79* + 1.6 = *13.18 ms*

DATA CENTERS IN A HEXAGONAL NETWORK

- Assume 6 data centers in a hexagonal network with vertices separated by a distance r
- Then the mean distance *d* between nodes is given by *d* = $0.2r(4+2\sqrt{3})$
- Mean propagation delay α = $d/(0.69c)$
- Max distance in the network = 2r
- Max propagation delay $\alpha_{max} = 2r/(0.69c)$

A QUESTION FOR CONSIDERATION

- Can we operate blockchain systems such that one can obtain the higher performance that may be possible for distributed execution in a local data center while operating across globally distributed data centers?

TRANSACTION SUBMISSION AND ENDORSEMENTS

- Assume time for a client to submit a transaction is 2α
- Endorsing phase for a transaction can include

= Cost for communication with endorsers + Cost of simulating smart contract (very small)

$= 2\alpha_{max} + \delta_{endorse}$

$\approx 2\alpha_{max}$

- Assume time to create a block of K transactions is δ_{create}
- Average waiting time for a transaction for block creation = $0.5\delta_{create}$

BLOCK SHARING FROM THE ORDERER

- If there is no limitation on the number of gRPC connections between data centers, then the information dissemination cost from the orderer to N nodes is given by

$$(2\alpha + \beta L_{block})$$

- With *q* gRPC connections, the information dissemination cost is

$$\left\lceil \frac{N}{q} \right\rceil (2\alpha + \beta L_{block})$$

- If a gossip network is used with *log (N)* stages for communication to broadcast, then the information dissemination cost is

$$(\log N)(2\alpha + \beta L_{block})$$

  - For recent Fast Fabric proposal (ICBC 2019), one can use the header length instead of the block length to reduce this processing cost

VALIDATION COSTS POST ORDERING, AND COMMIT

- VSCC (Validation System Chain Code) and MVCC (MultiVersion Concurrency Control)

$$\delta_{validate} = \delta_{vscc,block} + \delta_{mvcc,block}$$

- If C concurrent CPUs are available to validate the K transactions in a block, then

$$\delta_{vscc,block} \approx \left\lceil \frac{K}{C} \right\rceil \delta_{vscc,trans})$$

- However, MVCC is sequential hence

$$\delta_{mvcc,block} \approx K\delta_{mvcc,trans}$$

- Ledger commit cost

$$\delta_{commit} \approx K\delta_{commit,trans}$$

END TO END THROUGHPUT

$$X^{-1} \approx K^{-1} \max\left(2\alpha_{max}, \delta_{endorse}, 2\alpha, \delta_{create}, (\log N)(2\alpha + \beta L_{block}), \delta_{validate}, \delta_{commit}\right)$$

$$X^{-1} \approx K^{-1} \max\left(2\alpha_{max}, \delta_{create}, (\log N)(2\alpha + \beta L_{block}), \delta_{validate}, \delta_{commit}\right)$$

END TO END LATENCY

$$\Delta \approx 2\alpha_{max} + \delta_{endorse} + 2\alpha + .5\delta_{create} + \\ + (\log N)(2\alpha + \beta L_{block}) + \delta_{validate} + \delta_{commit}$$

$$\tilde{\Delta} = 60 + 6 + 50 + 150 + 52 + 40 = 358 \text{ ms.}$$

EXAMPLE CALCULATION

- In Androulaki et al, Eurosys 2018, the ordering phase took the longest time for 248 ms for 2MB blocks with 3.06kB for each transaction, so that

$$K \approx 2MB / 3.06\ kB \approx 669 \text{ transactions}$$

- Thus such a system can be expected to give a max throughput of

$$669 \text{ transactions} / 248ms = 2697 \text{ transactions per second}$$

THROUGHPUT FOR LARGE K?

- As $K \rightarrow \infty$

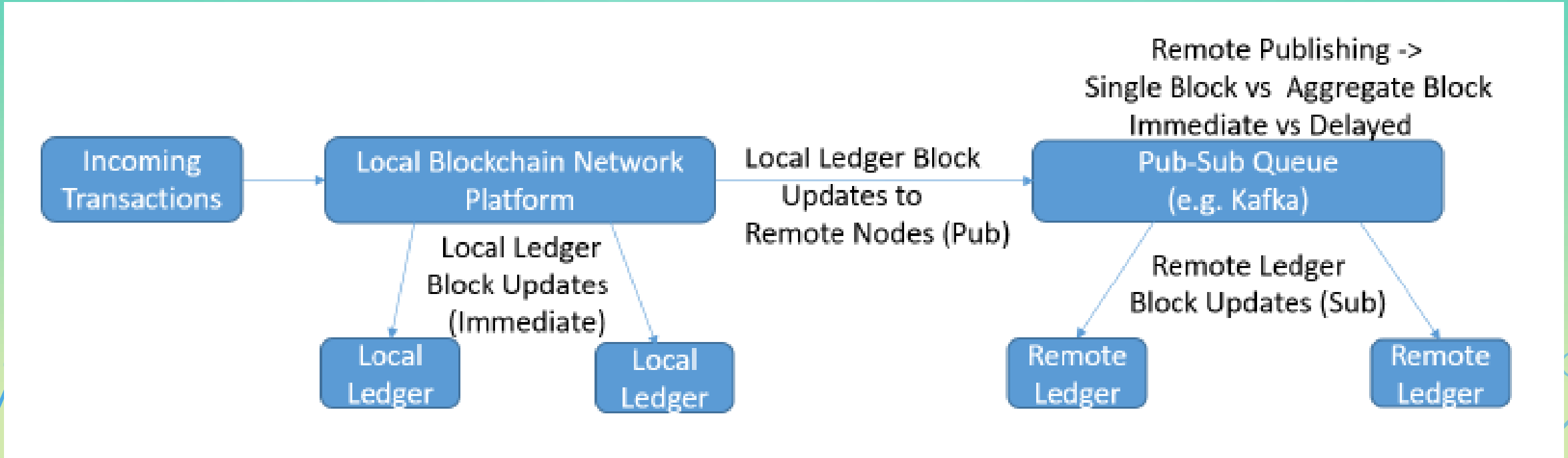
$$L_{block} = K L_{trans}$$

$$\delta_{validate} \approx \delta_{vscc,trans} + K \delta_{mvcc,trans}$$

$$\delta_{commit} \approx K \delta_{commit,trans}$$

$$X^{-1} \approx \max((\log N)\beta L_{trans}, \delta_{mvcc,trans}, \delta_{commit,trans})$$

5G DISTRIBUTED EDGE VS REMOTE BLOCKCHAIN LEDGERS



Appears at 5G Edge Workshop at ICDCN (Intl Conf on Dist Comp & Networking) 2020

DISTRIBUTED BLOCKCHAIN PROCESSING

- Global closure versus local closure

$$\Delta_{local} \approx 2\alpha_{max,local} + \delta_{endorse} + 2\alpha_{local} + .5\delta_{create} + (\log N_{local})(2\alpha_{local} + \beta_{local} L_{block}) + \delta_{validate} + \delta_{commit}.$$

$$\Delta_{global} \approx \Delta_{local} + \delta_{wait} + (\log N_{remote})(2\alpha + \beta M L_{block})$$

- Throughput performance based on edge local constraints

$$X_{local}^{-1} \approx K^{-1} \max\,(2\alpha_{max,local}, \delta_{create}, \delta_{validate} (\log N)(2\alpha_{local} + \beta_{local} L_{block}), \;, \delta_{commit})$$

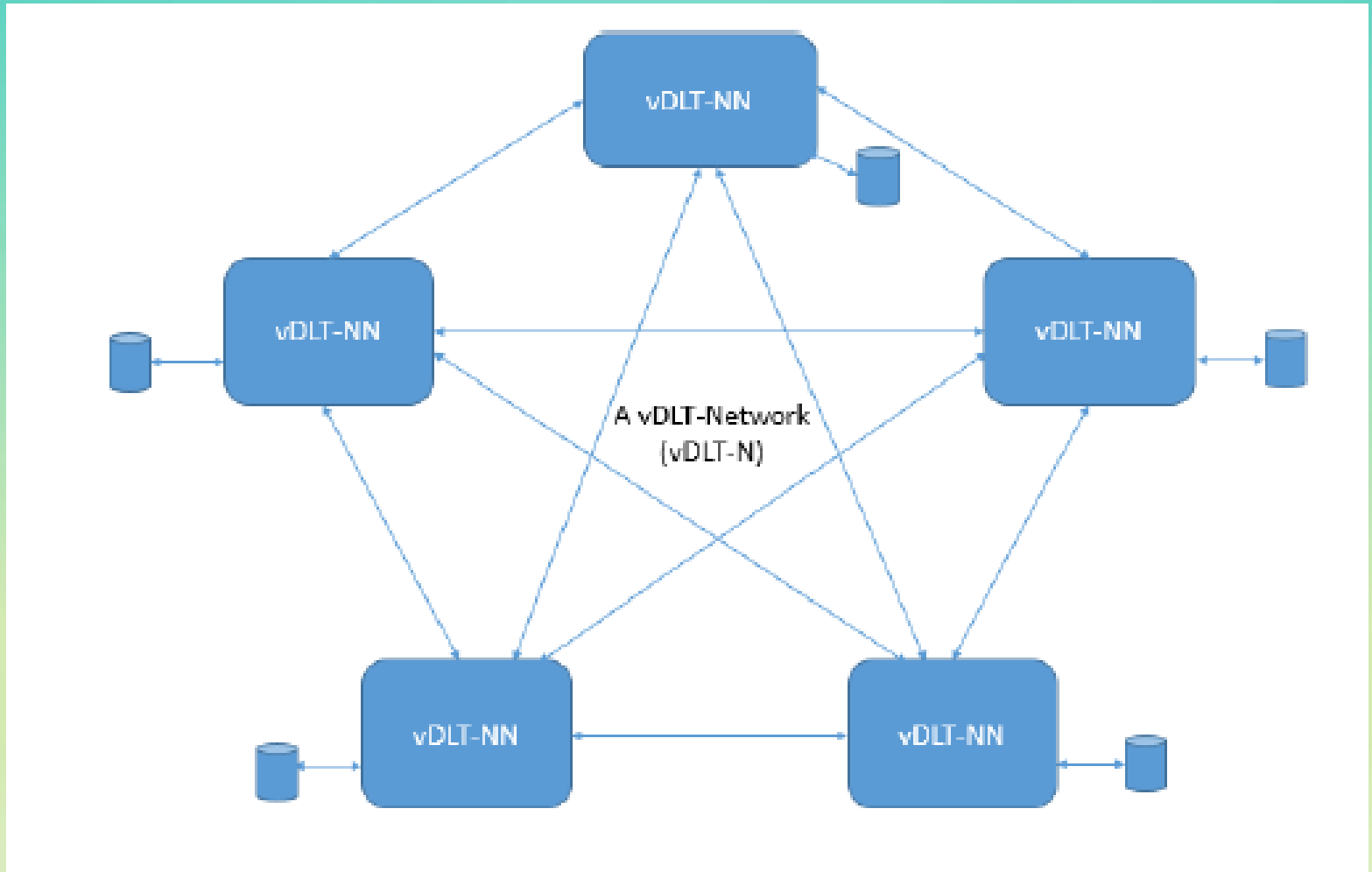
EXAMPLE CALCULATION

To illustrate the benefits of the lazy blockchain network decoupling, consider a distributed blockchain system with α_{max} = 30ms, α = 25 ms, $\alpha_{max,local}$ = 1ms, α_{local} = 0.8ms, $\delta_{sim\_endorse}$ = 6ms, $\delta_{validate}$ = 52ms and δ_{commit} = 40ms, and K = 150 transactions per block. Let us assume that the system has an end-to-end mean delay constraint of 160 ms. Let us assume that ordering phase latency in the local network is 48ms including time for block creation with the distribution to a smaller set of local nodes. Let us also assume that

EXAMPLE CALCULATION CONTD.

distribution to a smaller set of local nodes. Let us also assume that the ordering latency on the larger distributed network including block creation latency = 150ms. From (16), Δ_{local} = 2 + 6 + 1.6 + 48 + 52 + 40 = 149.6 ms, thus meeting the end-to-end mean delay constraint of 160ms. The throughput is dominated by the validation stage given by 150 transactions / 52ms = 2884 tps. Instead if the larger distributed network would have been used, then from (13), Δ = 60 + 6 + 50 + 150 + 52 + 40 = 358 ms, which would have violated the end-to-end mean delay constraints for the system design.

A VIRTUALIZED DLT NETWORK



THE TREND TOWARDS DISAGGREGATION

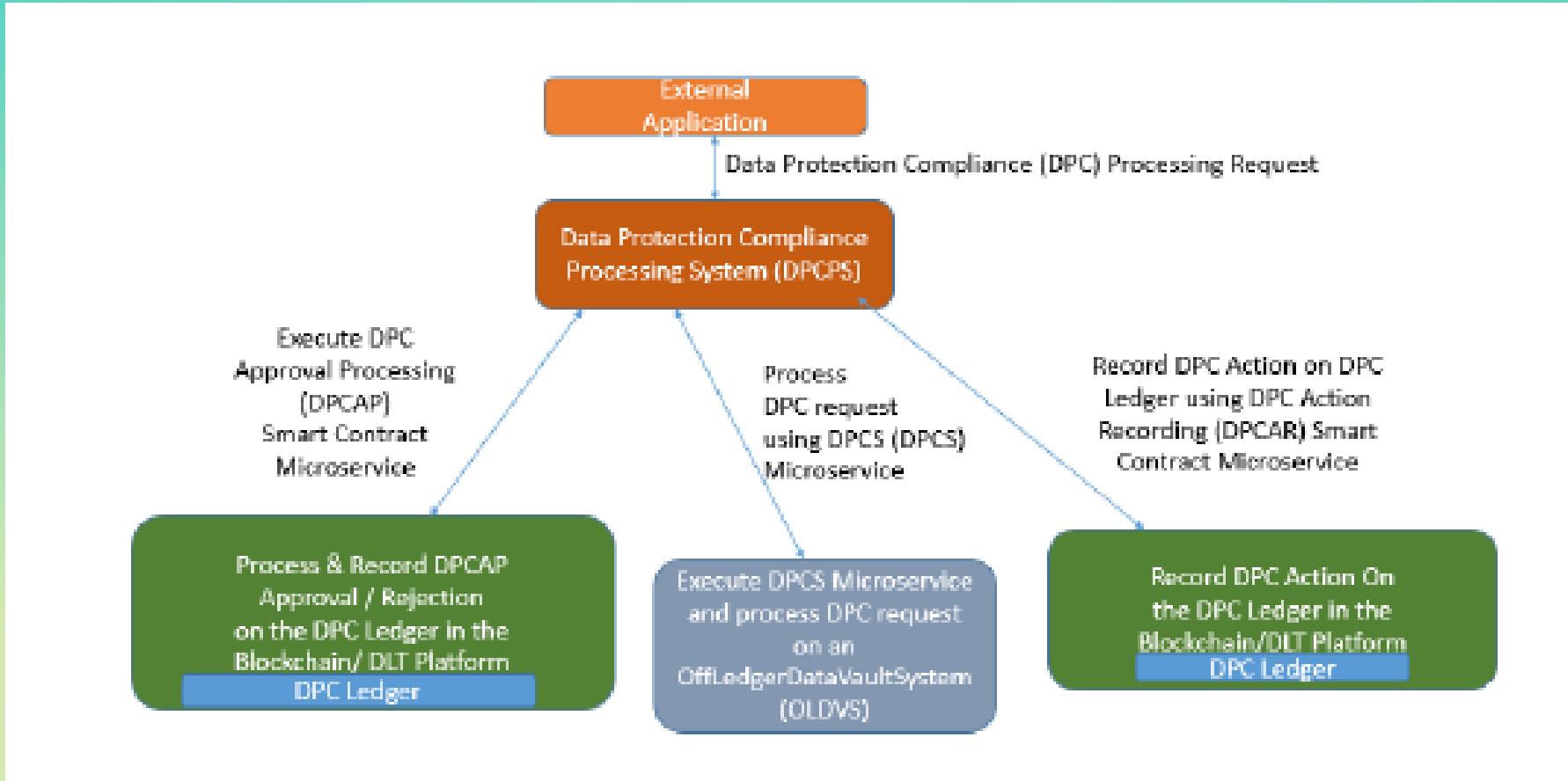
- Break system down into microservices
- Provides scalable replication for each microservice as dynamically needed relative to a monolithic system
- Ease of Development and Deployment
- Typical implementation : Docker containers to host microservices, Kubernetes for orchestration, optional Springboot java to develop microservices, Ribbon load balancing, Netflix Zuul API gateway
- A Blockchain service can be offered as an additional microservice to be invoked when needed in distributed 5G infrastructure (IEEE Globecom 2019 –Blockchain in Telecom workshop)

- Research Challenges
- How much to disaggregate?
- How to dynamically combine microservices to accomplish an end-to-end task?
- How to exploit concurrency?

APPLICATIONS TO VARIOUS DOMAINS

- Emergency Healthcare
- Dynamic Distributed Energy Resource Management
- Automotive/Transportation
- Smart Agriculture
- Tracking children, pets, elderly, migrant laborers
- Retail supply chains and provenance-mgmt.

DATA COMPLIANCE PROCESSING



Appears at IEEE Globecom 2019 (Blockchain in Telecom workshop)

EMERGENCY HEALTHCARE

- IoT/Wearables-based intelligent service hosted at a 5G edge server call ambulance for a patient
- Patient is driven to the nearest hospital
- During the drive, patient data is transferred over a blockchain network across hospitals
  - Patient Data stays off-chain, Event is recorded on-chain, Hash of Data can be stored on-chain
- Patient / care-giver gives consent live, or alternatively a pre-authorized consent for the emergency situation is pre-configured as a smart contract to enable data transfer
- Automated processing and confirmation of insurance on a blockchain ledger channel between the insurance company and the hospital for treatment
- Patient receives needed emergency care seamlessly

DISTRIBUTED SMART ENERGY / TRANSPORTATION MGMT

- Smart Energy Vehicles can automatically detect need for EV-charging, locate the nearest charging station, and perhaps even drive the vehicle to the charging station
- Smart Microgrids with green energy sources can interact with the Core Grid and with each other to determine dynamic energy availability and optimization
- Available energy can be dynamically routed to areas that need energy in an automated distributed smart energy network
- Dynamic Transportation / Traffic Management

CONVERGENCE OF 5G, IOT, AI AND BLOCKCHAIN

- 5G can allocate adequate dynamically scalable end-to-end resources for each IoT application or service in a 5G network (at the edge if required)
- IoT systems can monitor and provide triggers for processing
- Smart Contracts can execute the distributed automated intelligence in the system
- Blockchain platforms can record events in their ledgers, and enable dynamic relevant actions to be taken
- Blockchain for the 5G Control Plane (Network layers or Application layers)
- Overall a distributed smart network to deliver useful services

A FUTURE OF DISTRIBUTED SMART NETWORKS?

Automated Intelligent Quantum Resistant Data Processing/Storage with Smart Agents Interacting across Distributed Cloud/Edge Networks to provide Useful Services to People

THANK YOU

- Questions?

- Email: dilip@ieee.org