CONSENSUS

CFT vs BFT

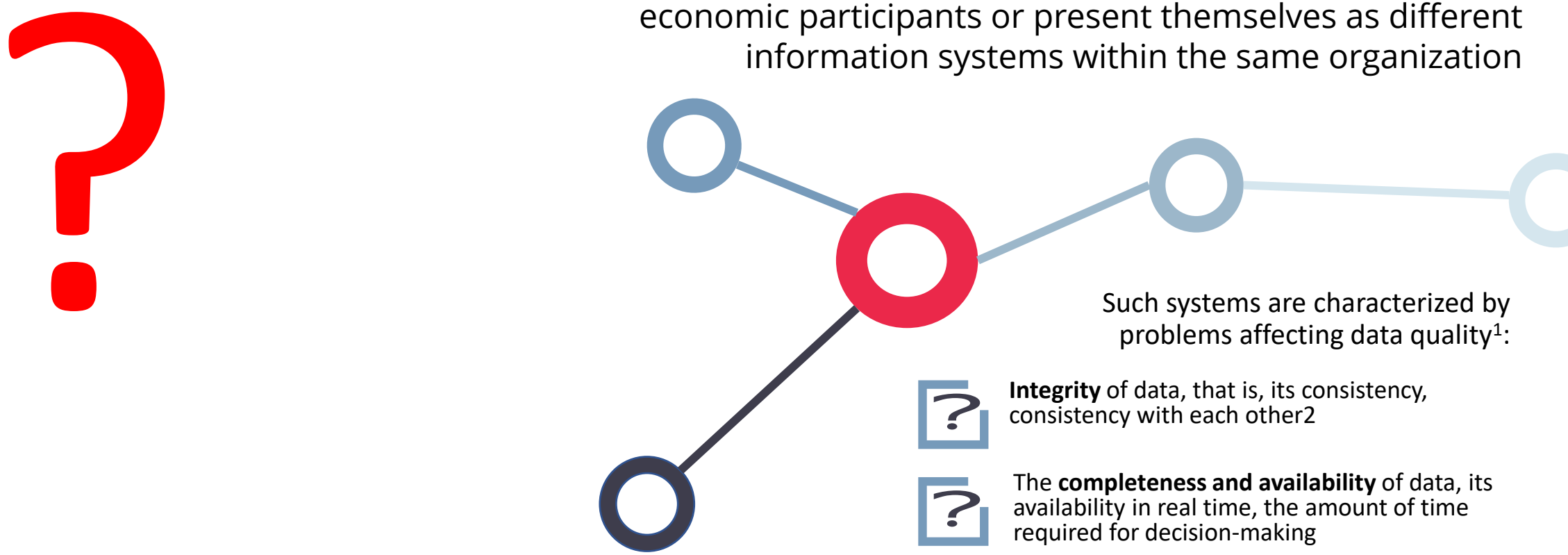
Valery Khvatov, DGT Network

18.08.2020

DISTRIBUTED / MULTIAGENT SYSTEM

Modern information systems are increasingly being **multi-agent networks** in which data is processed by different nodes.

Such agents (nodes) may be owned by independent economic participants or present themselves as different information systems within the same organization



Such systems are characterized by problems affecting data quality[1]:

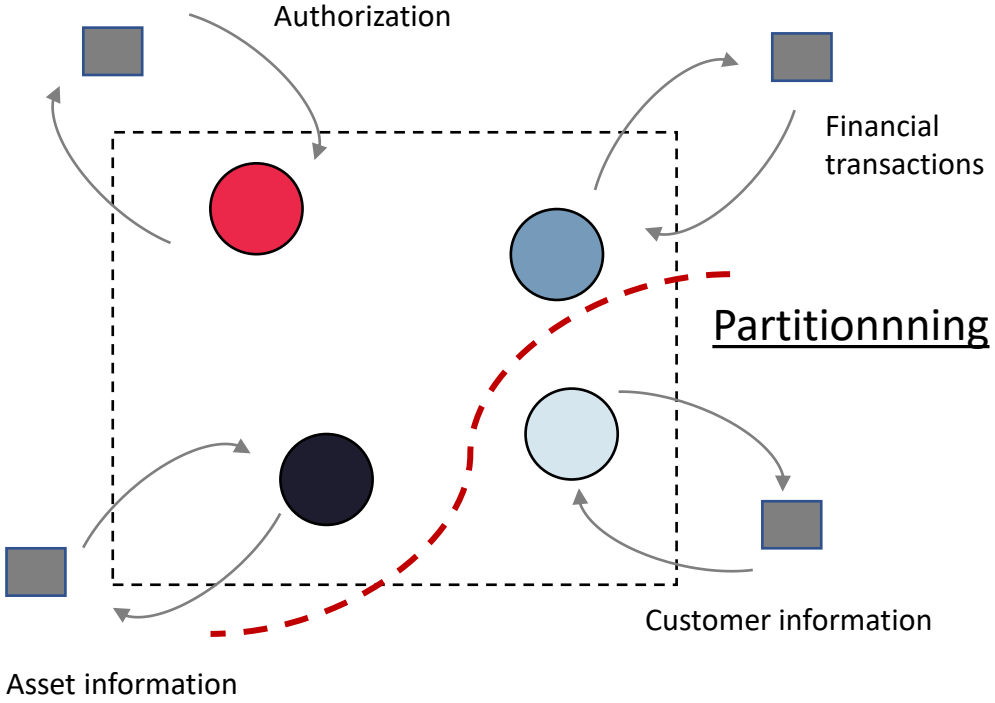
- **Integrity** of data, that is, its consistency, consistency with each other2
- The **completeness and availability** of data, its availability in real time, the amount of time required for decision-making
- **Data privacy**, access to information is only made by eligible participants

[1)] a feature showing how fit the data to use

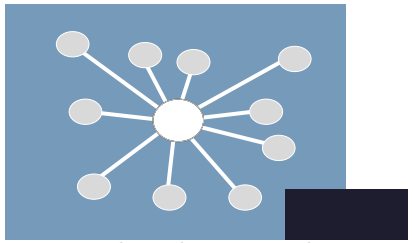
[2)] state of information in which there is no change or change is made only intentionally by entities entitled to it

DECENTRALIZATION AND DISTRIBUTION

Interacting with each other participants are united in a network that can have different topology - centralized, decentralized, distributed. The degree of distribution and decentralization may vary



Networks can break down into fragments (paration), nodes may not be available because of their SLA[3] - all of these causes affect the totality of data on the network, their integrity and availability.



Centralized network

Depending on the network's characteristics, its characteristics affect the data:

- Throughput - How many transactions can be completed over a period, TPS
- Latency - how long it takes each transaction;

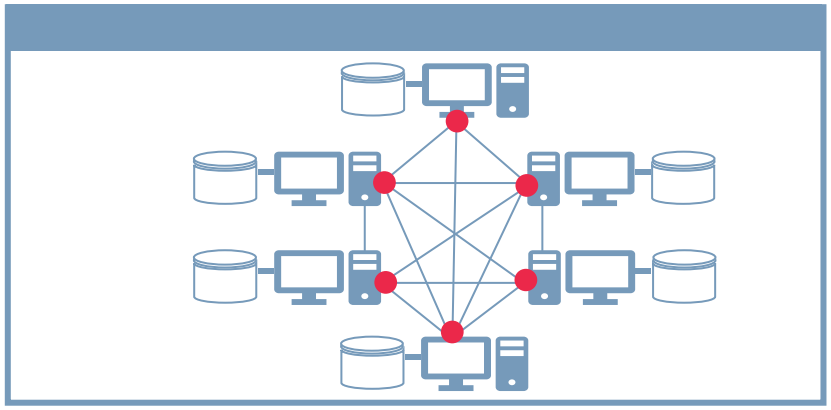
Decentralized network

Distributed network

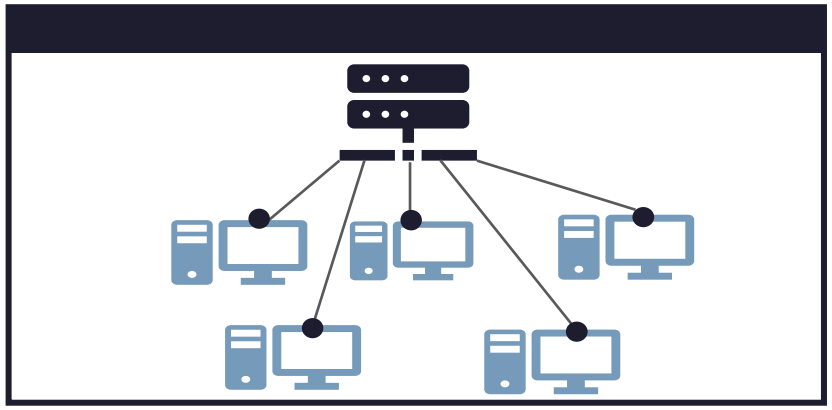
The consensus mechanism (reconciliation) of data depends on the level of decentralization and distribution and affects TPS and latency

[3] SLA (Service Level Agreement) – a measure of power and performance (productivity) for distinct nodes

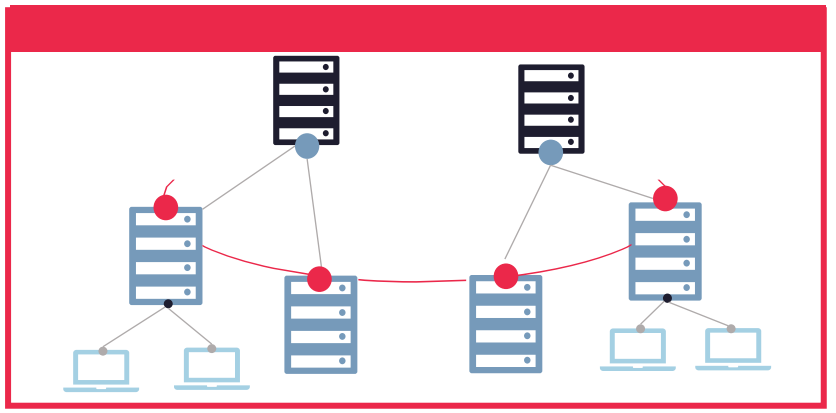
NETWORK TOPOLOGY



Peer-to-peer (P2P) network



Multi-rank (client-server) network



Hybrid multi-rank networks

Client-server networks

- ✓ **Control of the composition of the network -** a strict definition of connected networks (strong authentication);
- ✓ **single data collection point**, with the ability to control data at one point
- ☹ **requires a powerful server.** There is no horizontal scalability option.
- ☹ **there is no independence of the work of individual nodes,** uniformity is required

Hybrid networks can compensate for the shortcomings of client-server networks and P2P networks through flexible policies

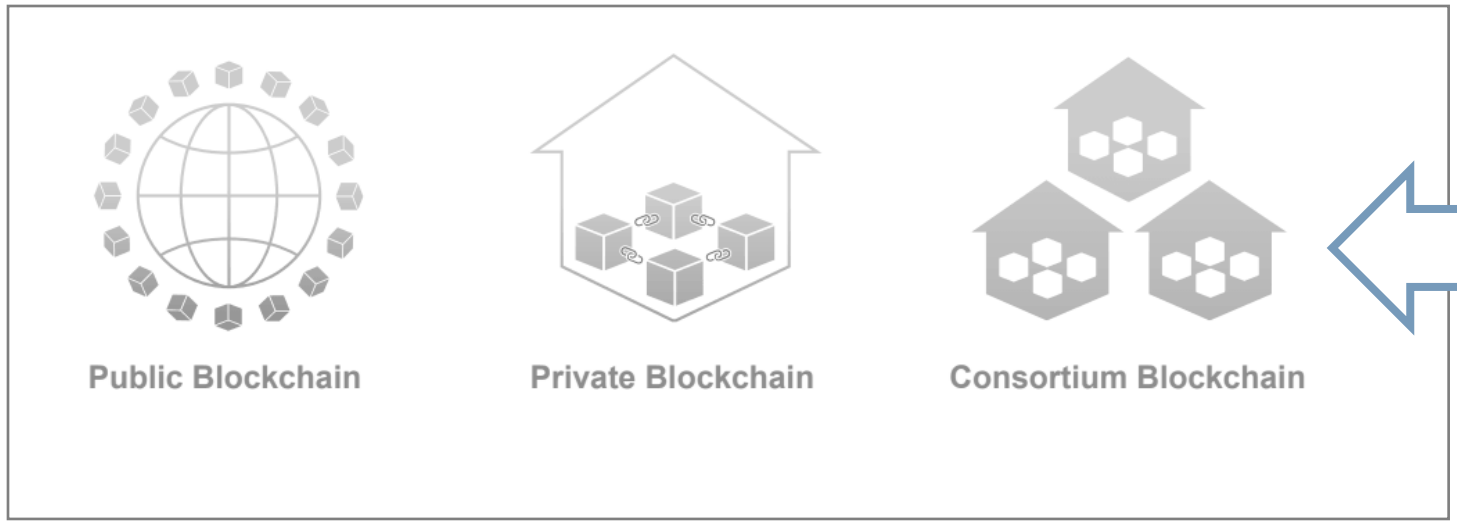
P2P- nets

- ✓ **Scalability** — No "narrow neck" is created at certain points of the network, as information exchanges can occur directly between the end nodes;
- ✓ **Stability**— The network is maintained when almost any number of nodes are disconnected from the network;
- ✓ **Privacy** — user data can be stored and calculations can be done directly on personal computers, without the involvement of a third trusted party;
- ☹ **Vulnerability** to "Byzantine attacks" in terms of data substitution and traffic manipulation;
- ☹ **Data quality control** requires special checks

The mechanism of consensus (reconciliation) of data depends on the topology of the network

[5)] SCALABILITY is the system's ability to cope with increased load: vertical scaling - increasing performance by increasing the power of individual components; horizontal scaling - parallel processing and performance gains due to structure

NETWORK OPENNESS AND DYNAMICS



Network dynamics (changes, nodes joining);
Network power
Transaction volume
Latency/TPS
Having different transaction families

- **Public ledgers** are fully open, where everyone can participate in an negotiation where transactions are not controlled by anyone and are carried out freely;
- **Consortium** - it controls the approval process of selected nodes;
- **Private blockchain** - all transactions are monitored and controlled by a centralized body

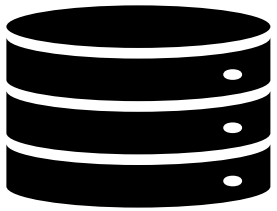
The planned number of connected nodes, their life time, the stability of the entire network have a significant impact on the network architecture.

How to join a network determines the degree of trust in nodes and the ability to accept transactions from them for write-in transactions in a general registry.

DATA STORAGE

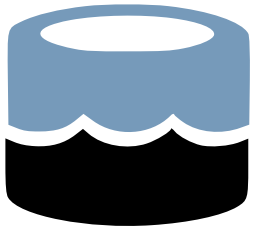
The organization of the data storage system, synchronization of data, the order of insertion of data into the general registry for nodes imposes restrictions on the architecture of the data and the general approach to integration

Distributed Ledger Technologies (distributed ledger technologies) include a class of blockchain solutions



Relational databases

The distribution of data sources is decided by replication and ETL[6] processes

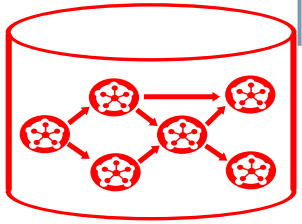


Data lakes (NoSQL databases)

Rejection of full principles ACID[7], transition to BASE architecture

Blockchain

Storage of data in blocks that are subject to restrictions. In fact, block storage is a linear linked list



DAG – based solutions

Storage of data in DAG - graph databases, which in addition to the data are stored links, branching is allowed

The mechanism of consensus (reconciliation) of data is largely determined by the order of reading and recording of data messages (transactions), different consensuses use different algorithms to record data

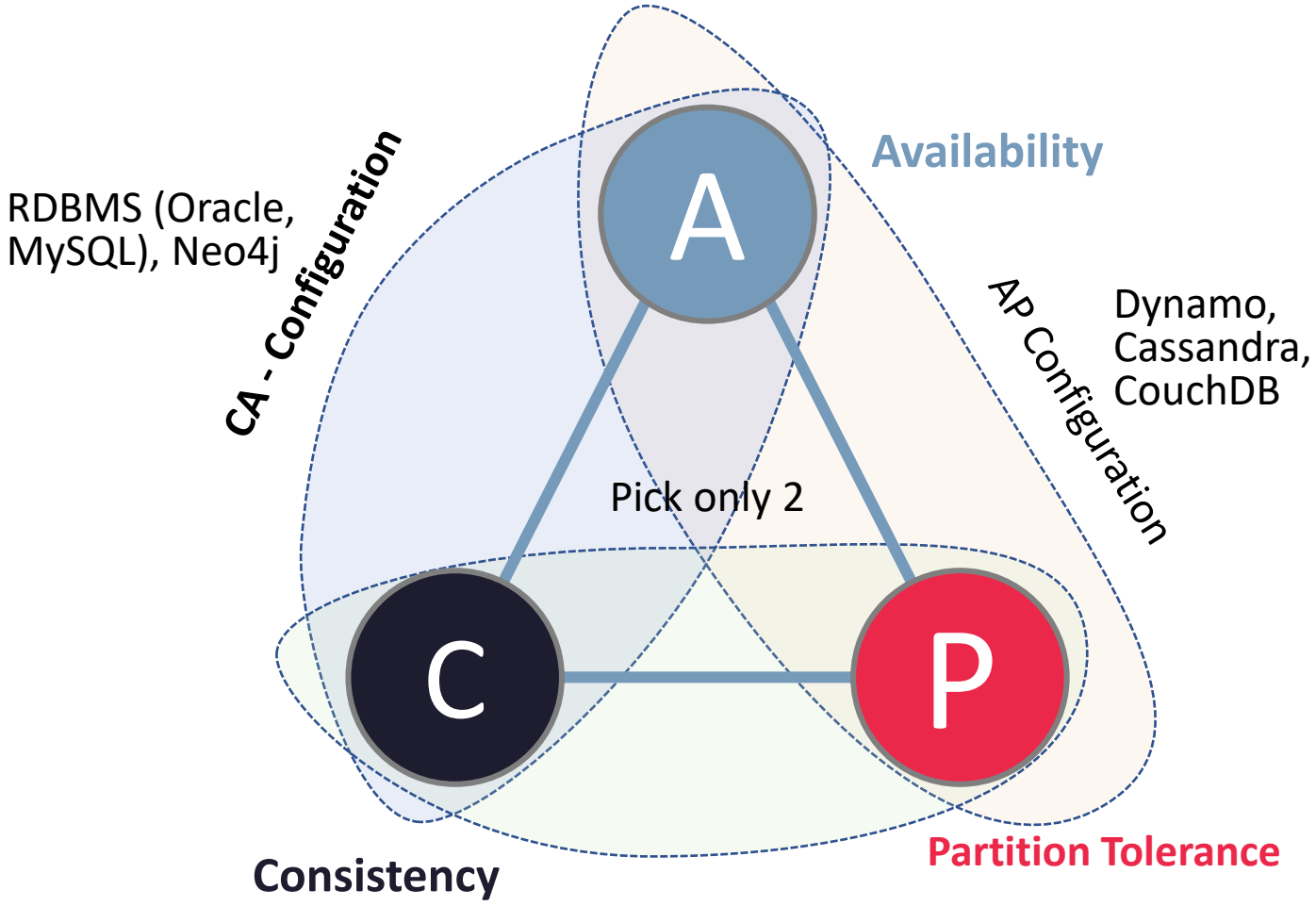
6) ETL – Extract, Transform, Load: literally "extract, conversion, download," the process of transferring data to central storage

7) ACID — transactional system requirements (Atomicity - atomicity, Consistency - consistency, Isolation - isolation, Durability - durability)

7) BASE — the blockchain and big data storage architecture that provides basic availability, soft state, eventual consistency;

8) DAG — Directed Acyclic Graph – oriented directional graph. The structure is often used for computational tasks because of the ability to topological sorting carried out over the final time;

CAP THEOREM



CP - Configuration

BigTable, Hbase, Redis

CAP Theorem (Brewer's theorem) — it is a heuristic assertion that in any implementation of distributed computing it is possible to provide no more than two of the three following properties: consistency of data (in all computational nodes at one point in time data does not contradict each other), availability (any request to the distributed system is completed with a correct response), resistance to separation (partition tolerance)

PACELC theorem — CAP extension: If the network (P) is separated from a distributed computer system, you must choose between availability (A) and consistency (C), but in any case, even if the system works normally in the absence of separation, you need to choose between latency (L) and consistency (C).

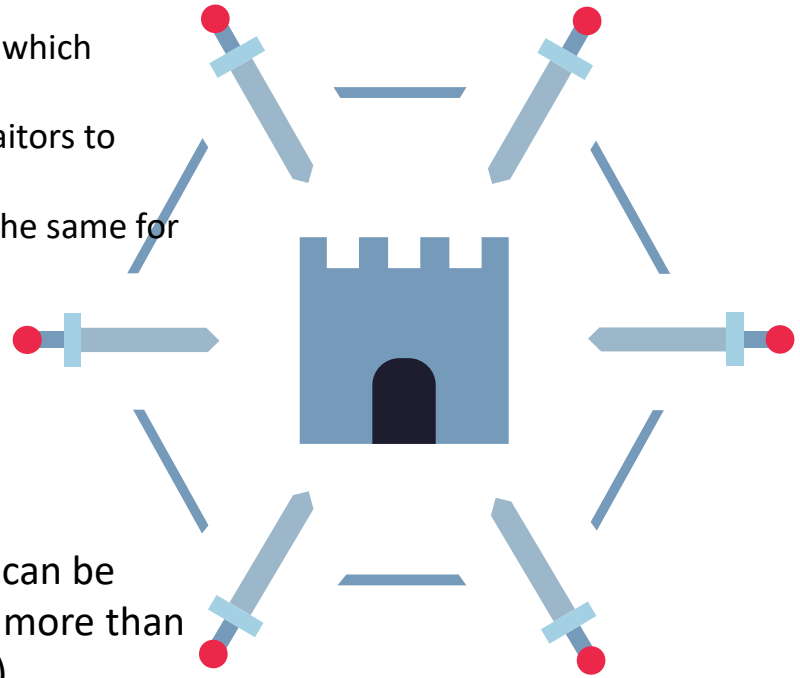
BASE architecture Basically Available, Soft-state, Eventually consistent — basic availability, unsustainable state, consistency eventually

BYZANTINE ATTACKS

All information systems are vulnerable. Distributed systems are vulnerable to so-called "Byzantine attacks" - Byzantine Fault (from the classic task of the Byzantine Generals). Reflects the peculiarity of distributed systems in failures occurring under unknown status of a system component (node) that may not function properly or be unavailable

THE TASK OF THE BYZANTINE GENERALS

Byzantium. The night before the great battle with the enemy. The Byzantine army consists of n legions, each of which commands its general. The army also has a commander-in-chief, to which the generals are subordinated.
At the same time, the empire is in decline, and any of the generals and even the commander-in-chief can be traitors to Byzantium, interested in its defeat.
At night, each of the generals receives an order from the leader on the option of action at 10 a.m. (the time is the same for all and known in advance), namely, "attack the enemy" or "retreat."



Possible outcome of the battle:

- If all the faithful generals attack - Byzantium will destroy the enemy (favorable outcome).
- If all the faithful generals retreat, Byzantium will retain its army (intermediate exodus).
- If some loyal generals attack and some retreat, the enemy will destroy the entire Army of Byzantium (unfavorable outcome)

According to Lambert's theorem in a system with m wrongly working nodes ("disloyal generals") can be reached agreement (BFT) only if there are 2m/1 faithful processors ("loyal generals"), i.e. strictly more than 2/3 of the total number of processors (provided that messages can be changed when forwarded).

In general, with a variable number of nodes, the theoretical BFT task is not solvable, but for systems with limitations there are algorithms PBFT, PAXOS, RAFT

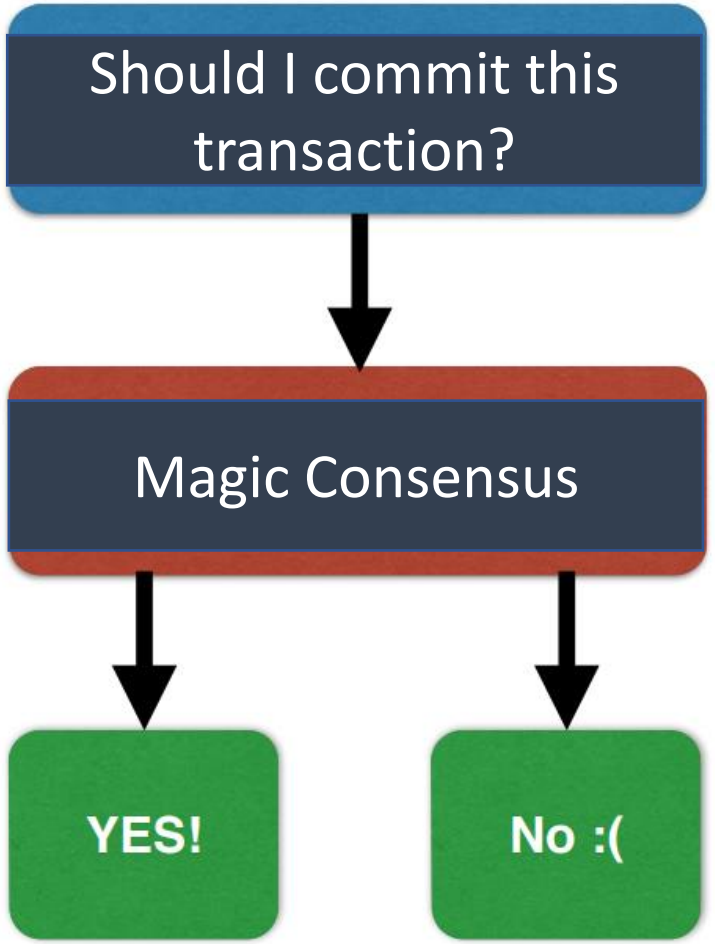
The consensus mechanism can be perceived as resilience in the attack vector (double waste) Sybill - attack, attack 51%)

CONSENSUS

Informal: “we all agree on something”

Consensus is the problem of having a set of processes agree on a value proposed by one of those processes

- **Validity**: the value agreed upon must have been proposed by some process (SAFETY)
- **Termination**: at least one non-faulty process eventually decides (LIVENESS)
- **Agreement**: all deciding processes agree on the same value (SAFETY)



DISTRIBUTED CONSENSUS

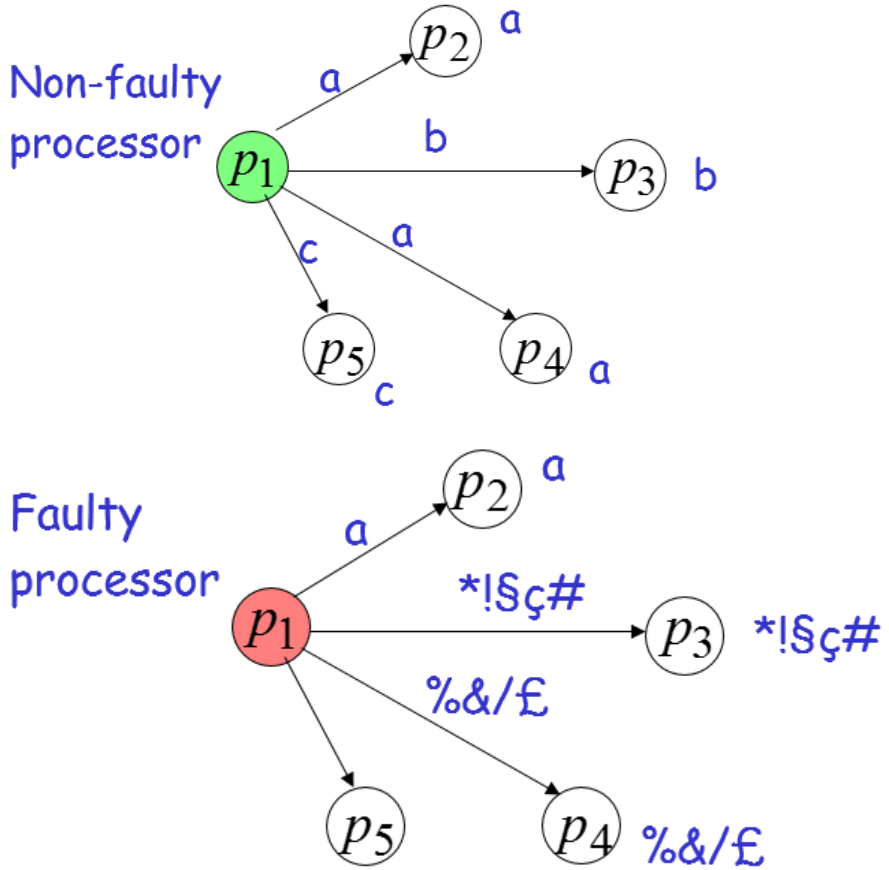
Distributed consensus is impossible when at least one process might fail

No algorithm solves consensus in every case

BFT (BYZANTINE FAULT TOLERANCE) / CFT (CRASH FAULT TOLERANCE)

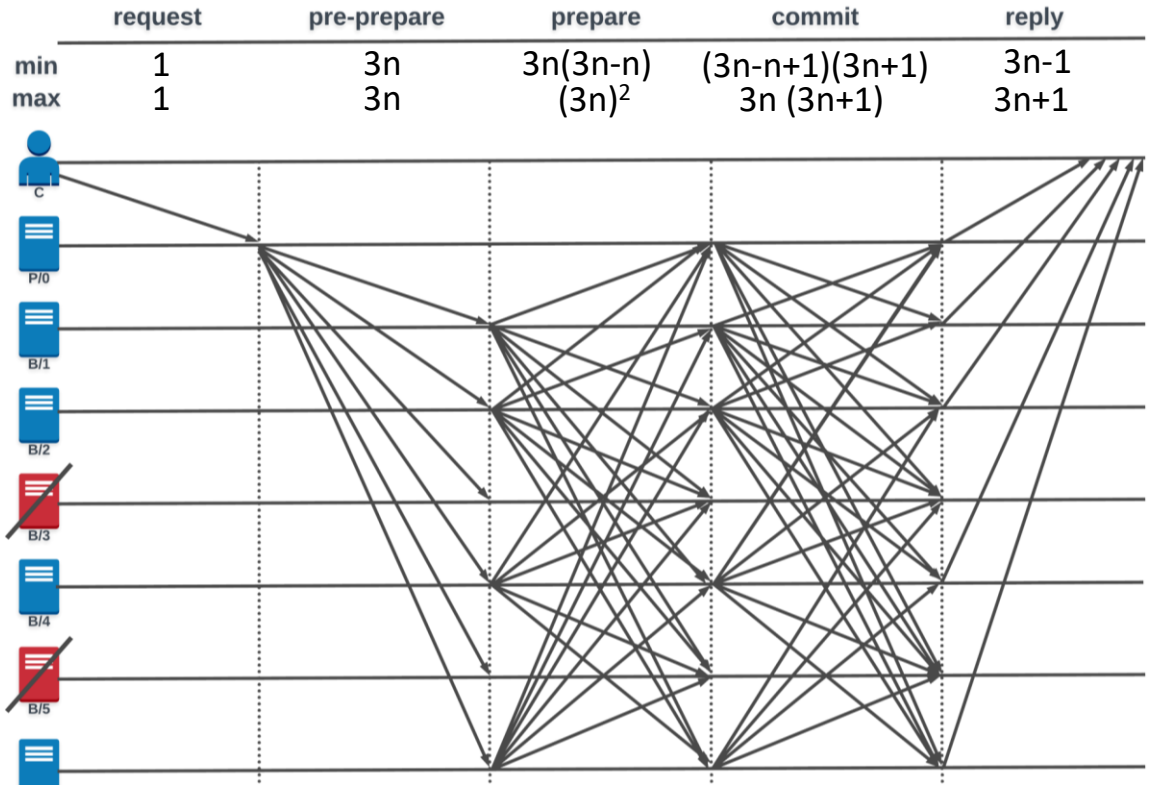
Crash fault tolerance (CFT) is one level of resiliency, where the system can still correctly reach consensus if components fail. Byzantine fault tolerance (BFT) is more complex and deals with systems that may have malicious actors

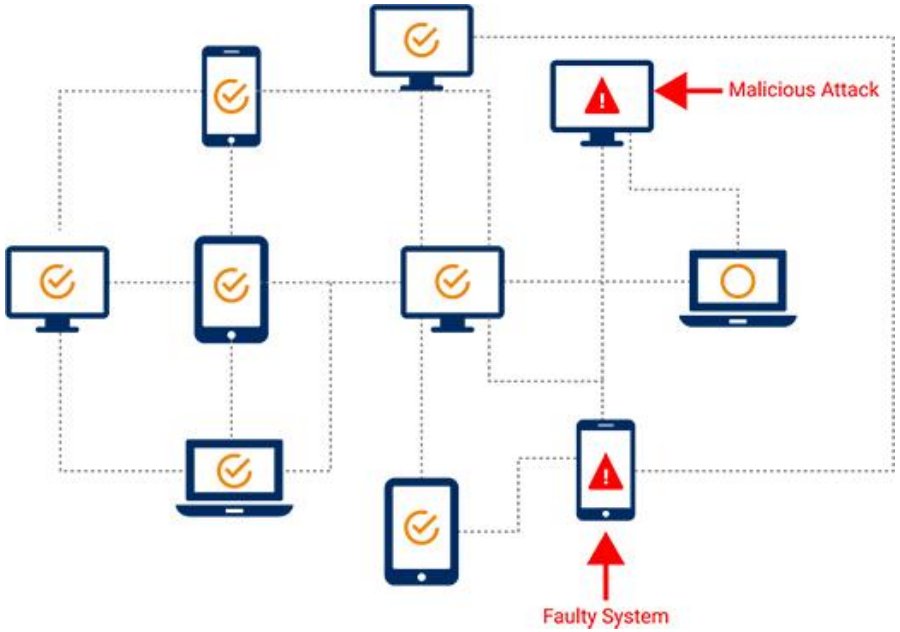
- Networks can fail — messengers can be captured or killed.
- Man-in-the-middle attacks can send forged messages — messengers can be compromised.
- Hardware or software components can break or crash — generals can be killed.
- Malicious components can send malicious messages — generals can be traitorous.



NUMBER OF RIGHT NODES

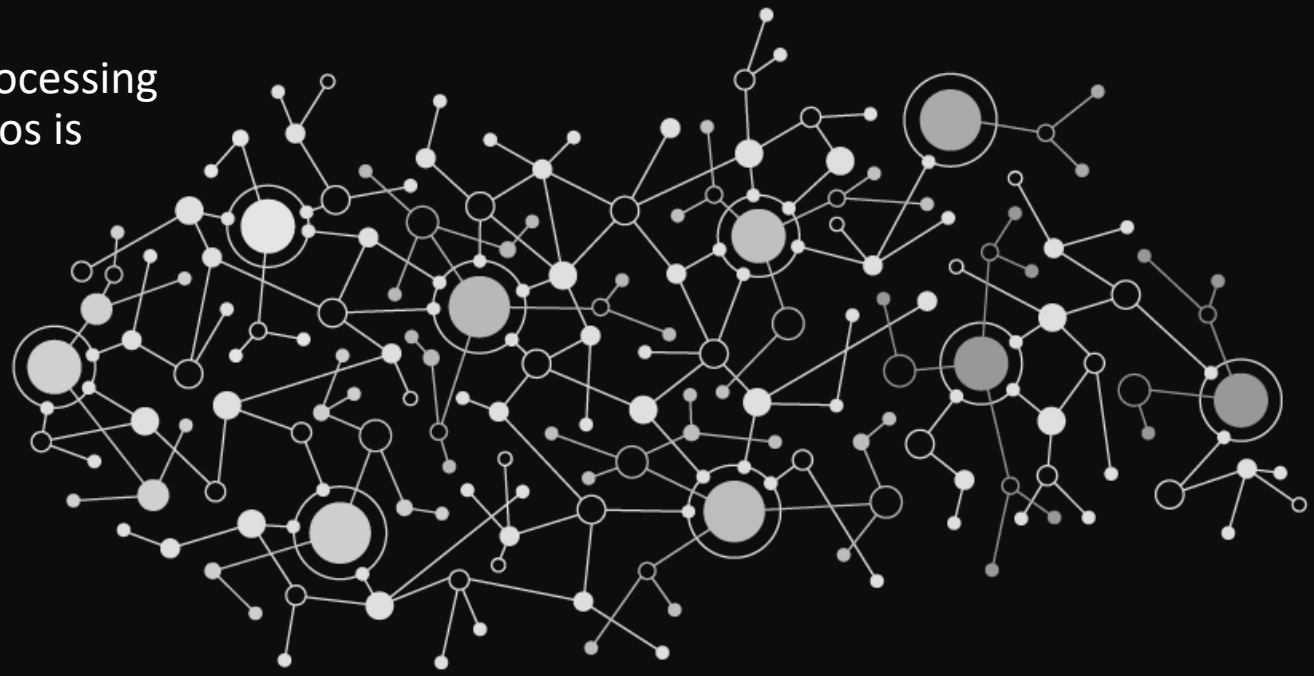
	Crash Faults & Partitions	Non-Crash Faults
Asynchronous CFT (Paxos)	$\frac{n-1}{2}$	0
Asynchronous BFT (PBFT)	$\frac{n-1}{3}$ (combined)	$\frac{n-1}{3}$ $(combined)$





THE PROBLEM OF CHOOSING A DLT PLATFORM

There are a lot of implementations of distributed source processing systems. Their classification and choice for practical scenarios is difficult due to the following circumstances:



- There is confusion between consensus mechanisms (algorithms) and platforms, platforms are much larger, some allow different consensuses to be used;
- The high level of hype around cryptocurrencies attracted a large number of non-professional teams that released clones and secondary solutions;

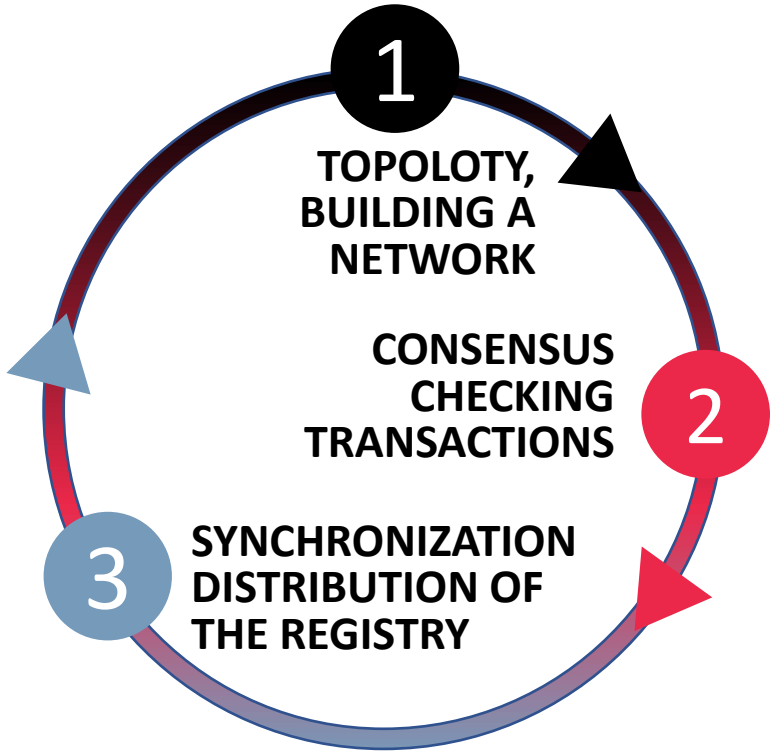
- The solutions market has not stabilized, in particular, there has been no separation between platform developers and protocols, network operators and application developers[9];
- Negative impact of non-decentralized solutions such as crypto-exchanges, Hyperledger Fabric platform, etc.;
- The complexity of the choice in terms of multidimensional criteria including the level of network closure, workloads, storage requirements

9) 2ND GLOBAL ENTERPRISE BLOCKCHAIN BENCHMARKING STUDY, 2019: A three-layer mental model for exploring the blockchain landscape

REFERENCE ARCHITECTURE

The reference architecture allows you to address the functionality of the platform as part of a single system approach.

From a computational perspective, any distributed platform solves three fundamental problems of distributed ledgers:



Presentation Level

Graph Visualization Transaction Forms Dashboard CLI

API / Application Level

Network Stat Transaction Receipts Node Stat Logging

Business Logic

Synchronization Timeouts Smart-contracts Event Handlers

Consensus Layer

Verification Validation Proofing Arbitrage

Network Layer

Peer-to-Peer Topology Permalinks

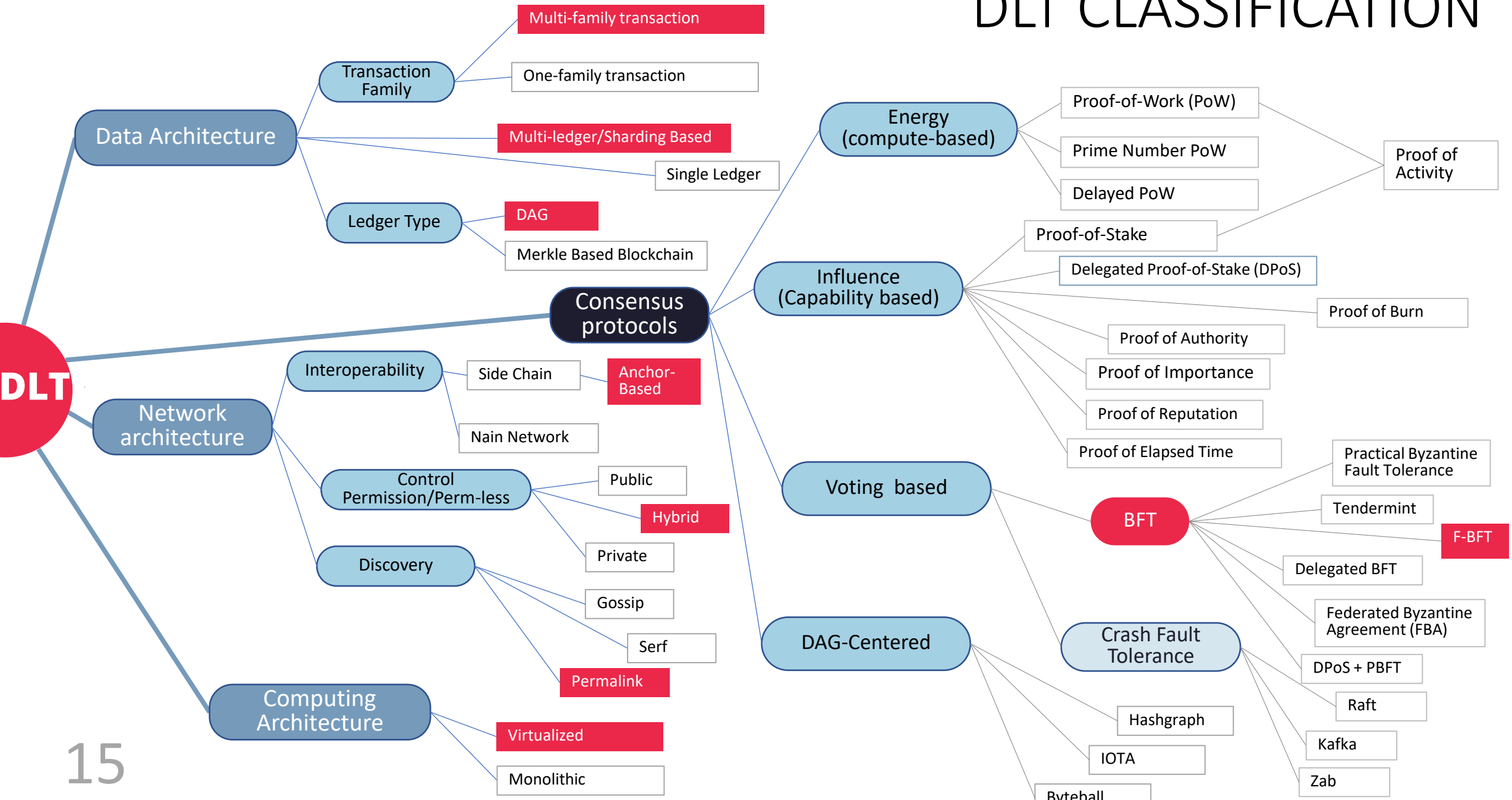
Data Layer

Digital Signature Hash Merkle tree Transactions DAG Encryption

Hardware / Infrastructure Layer

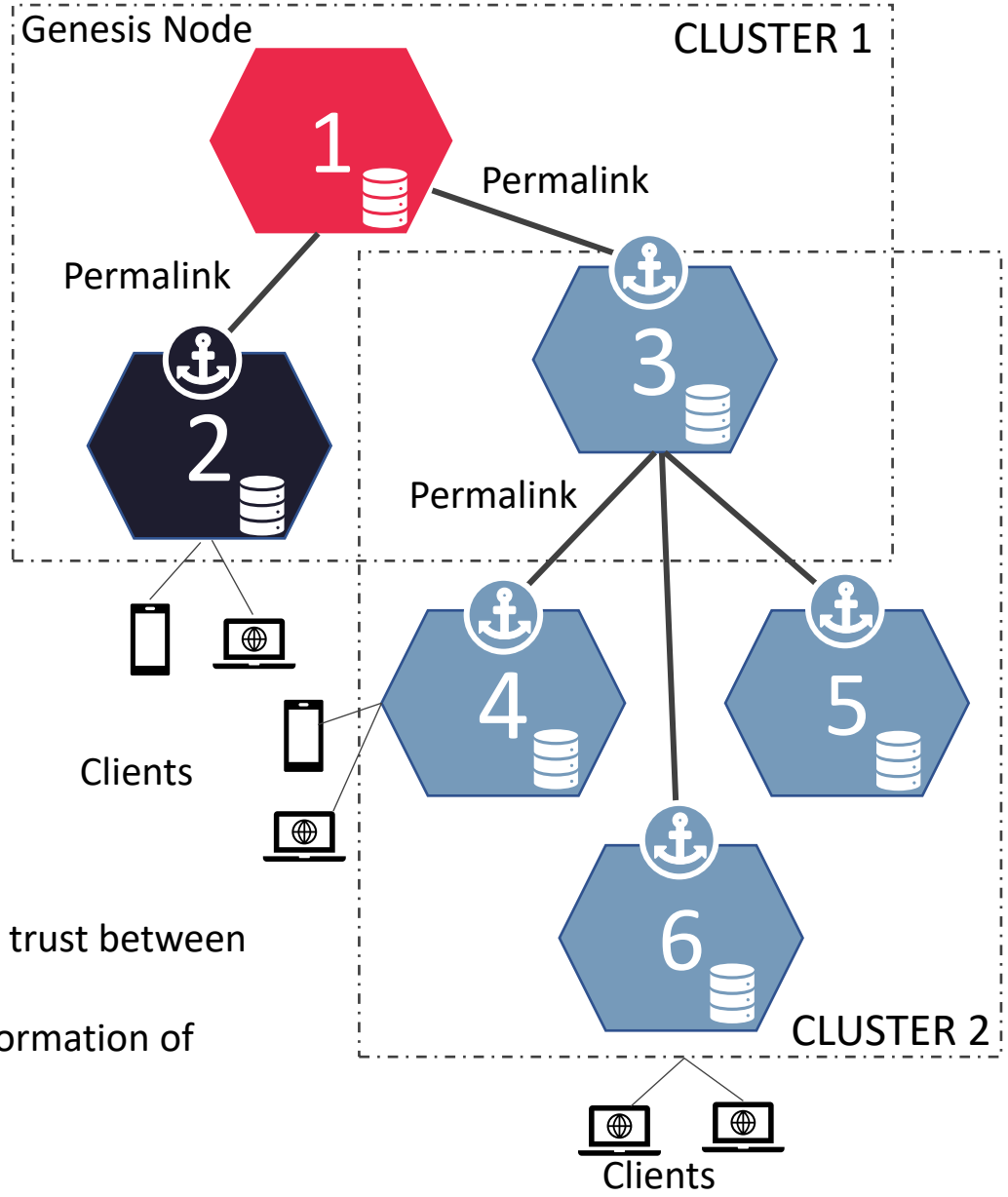
Virtual Machine Containers Services Messaging

DLT CLASSIFICATION



FEDERATED DGT NETWORK

DGT builds a network with cluster topology in mind. The cluster refers to a group of nodes that carry out the primary round of voting. Clusters can form complex structures.



- The size of the cluster that nodes attach to it is determined by the topology processor, which is a separate family of transactions;
- Inside each cluster, a variable leader is defined, which changes after several rounds of voting. If the leader is not asked for a certain time, the procedure for selecting a new leader takes place;
- Voting is initially done in a cluster, then a special algorithm is selected by an arbitrator outside the cluster;
- F-BFT prevents "double-spending" attacks by varying the "voting" times within the cluster and the characteristic DAG synchronization time (state) that is carried out through the permals;

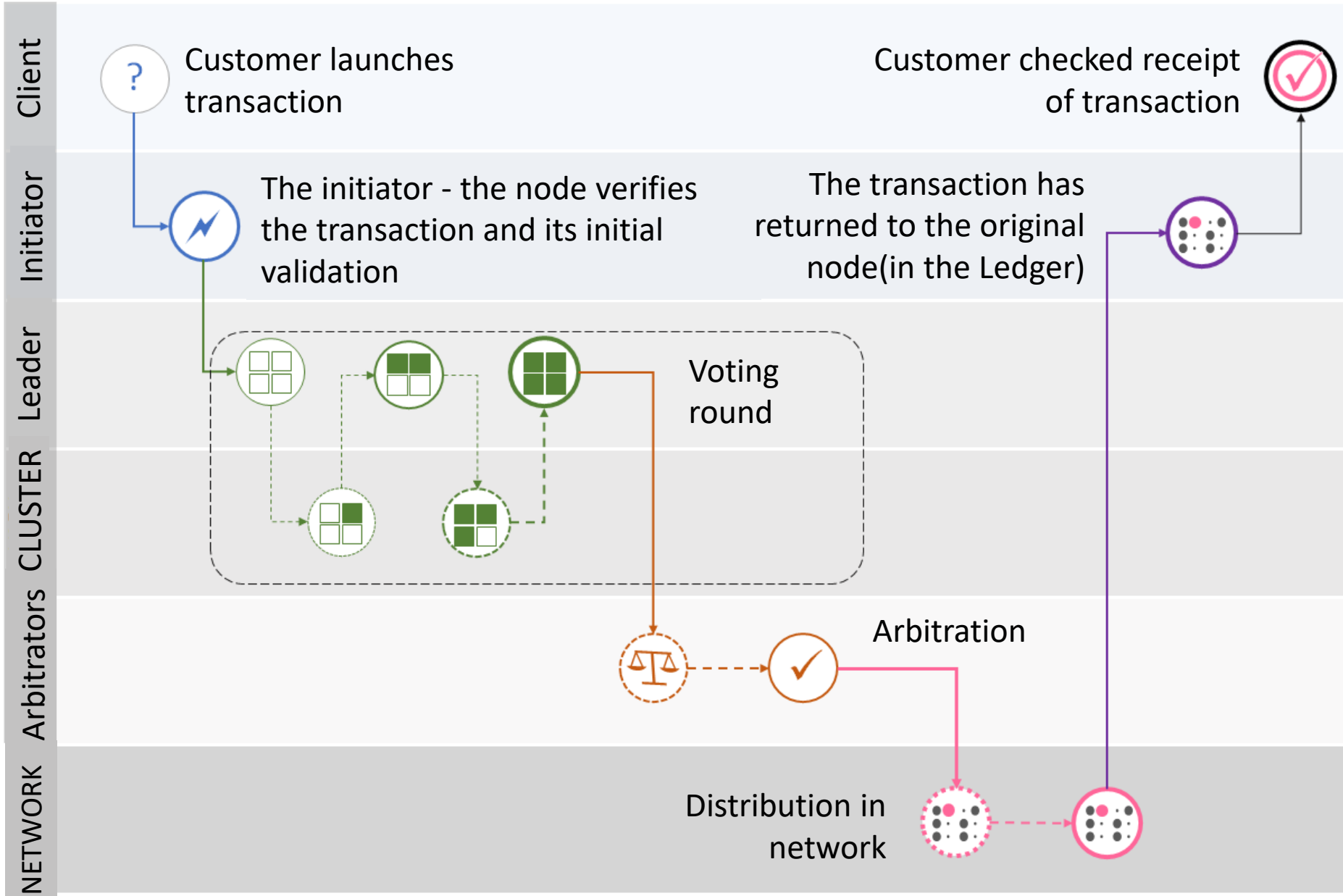
Clustering delivers the following benefits:

- Formation of "topologically close" group nodes, which improves trust between nodes (reduces the risk of attacks) and improves performance;
- Allows you to provide "sharding" of the network, including the formation of private branches of DAG;
- Increases the horizontal scalability of the network.

TRANSACTION PATH

The transaction is initiated by the customer, the system works asynchronously and requires re-applying for a "check".

The full voting cycle implements a 2 phase vote for a transaction with a random selection of arbitrators from a given ring



THANK YOU