UNIVERSITY of NICOSIA

**Week 4 – Session 8**

# Permissionless vs Permissioned Architectures

BLOC 512: Blockchain Systems and Architectures

# Session Objectives

- The scope of this session is to explain the different types of distributed ledger architectures in terms of access control and to explore some notable examples of notable use-case projects and different implementations.

# Agenda

1. Permissioned vs Permissionless blockchains
2. Basic consensus Algorithms
3. Avalanche (Repeated Random sub-sampling)
4. Cardano (PoS - Ouroboros)
5. Cosmos (Tendermint BFT-based Proof-of-Stake)

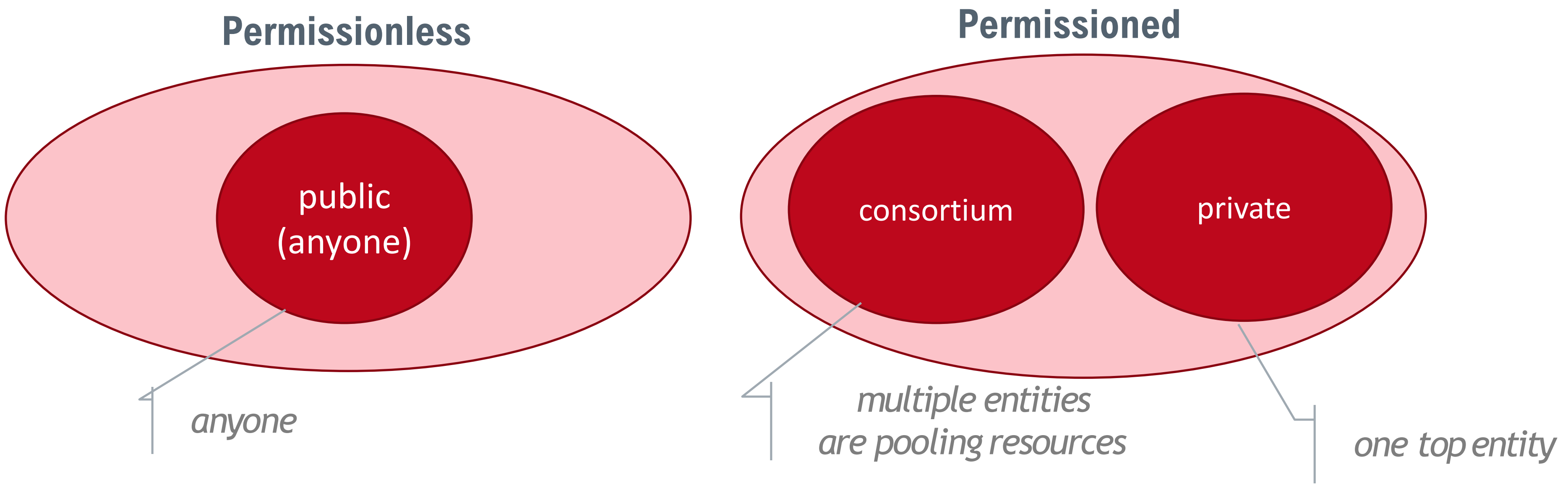
# Permissionless vs Permissioned Architectures

# Categories of Blockchains (from session 2)

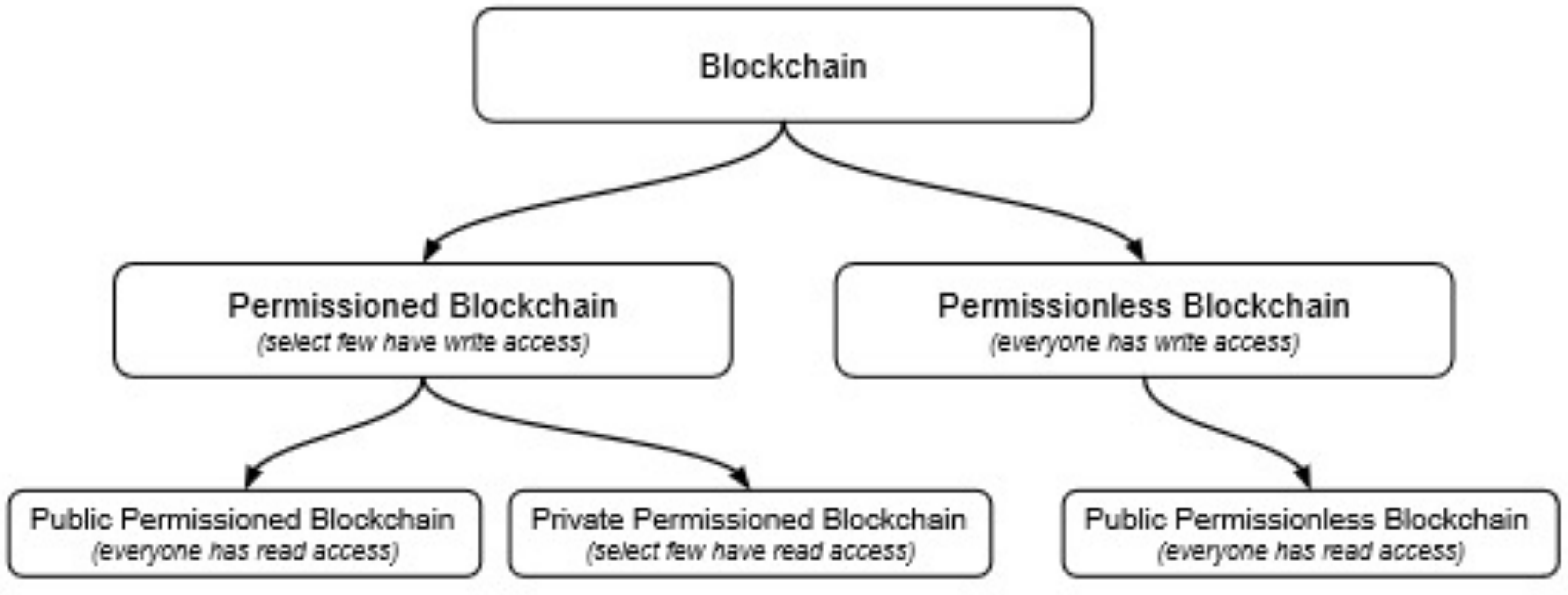
- We started our course by emphasizing the ***openness*** of blockchain in Bitcoin. Its two main properties:
  - *commonly verifiable transactions*, and
  - *impossibility of changing historical data*,

are undergoing substantial evaluations of how they can be used in a closed, ***private*** environment.

- We can cluster blockchains in two main clusters based on their accessibility:



# Blockchains categories based on permissions



Blockchain

Permissioned Blockchain
*(select few have write access)*

Permissionless Blockchain
*(everyone has write access)*

Public Permissioned Blockchain
*(everyone has read access)*

Private Permissioned Blockchain
*(select few have read access)*

Public Permissionless Blockchain
*(everyone has read access)*

# Categories of Blockchains (from session 2)

| ***Property*** | **Permissionless** | **Permissioned** | |
|---|---|---|---|
| | | **Private Blockchains** | **Consortium Blockchains** |
| ***Consensus Participation*** | Open, all nodes | One organization | A selected group of nodes |
| ***Consensus Algorithm*** | Variations of PoW and PoS | BFT adaptations/ variations | BFT adaptations/ variations |
| ***Centrally Managed*** | No | Fully | Partially |
| ***Read Permissions*** | Public | Read privileges can be assigned | Read privileges can be assigned |
| ***Write Permissions*** | High | Low | Low |
| ***Privacy*** | Limited | Could be preserved | Could be preserved |
| ***Immutability*** | Nearly impossible to tamper | As long as there is agreement – could be tampered | As long as there is agreement – could be tampered |
| ***Throughput*** | Low | High | High |
| ***Scalability*** | Poor | Good | Good |

# Categories of Blockchains

### Public Blockchains

- Provide the foundation for largest interoperability as they serve as a common hub for all connected services
- They are censorship resistant, without entities with special access, consequently, enjoy increased trust
- Suitable for global deployment, like digital currencies, global computation platforms or data repositories

### Private Blockchains

- Lower costs (vs. public), as validation is performed without external threats – minimized validation
- High adaptability and high throughput as the system is easier to update and optimize
- Suitable for managing data coming from dispersed sources, like branch offices or complex supply chains, where the origin and the transport of data could be compromised

### Consortium Blockchains

- Lower costs, higher speed and increased security for: sharing data, communicating and reconciling between organizations in a certain ecosystem as there is one shared database, which is trusted by all participants

# Overview

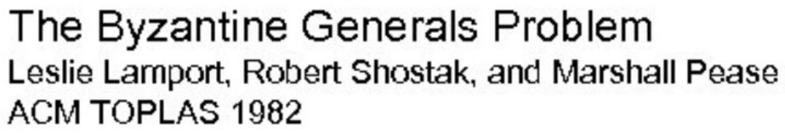
- Private blockchains are also known as permissioned blockchain
  - only authorized participants can join the network
  - governed only by a single entity
  - consensus in a private blockchain is less complex compared to public blockchain environments
- Consortium blockchains have privileged permissioned nodes across the network and share many of the same advantages such as privacy, efficiency, scalability, performance as with the case of a private blockchain, but operate under the governance of a group
  - Similar to a private blockchain, a consortium blockchain can limit the participants regarding the different access levels
  - Subsets of organisations can also be created to have their own channels to communicate and can have isolated data only for respective associations
  - Consensus participants of a consortium blockchain are likely to be a group of pre-approved nodes on the network

# Basic Consensus Algorithms

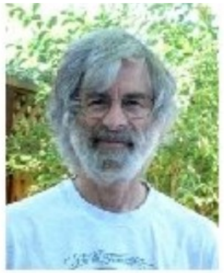
# Consensus in Blockchains

**Consensus** = participants' agreement about the current state of a ledger

# Byzantine generals problem



The Byzantine Generals Problem
Leslie Lamport, Robert Shostak, and Marshall Pease
ACM TOPLAS 1982

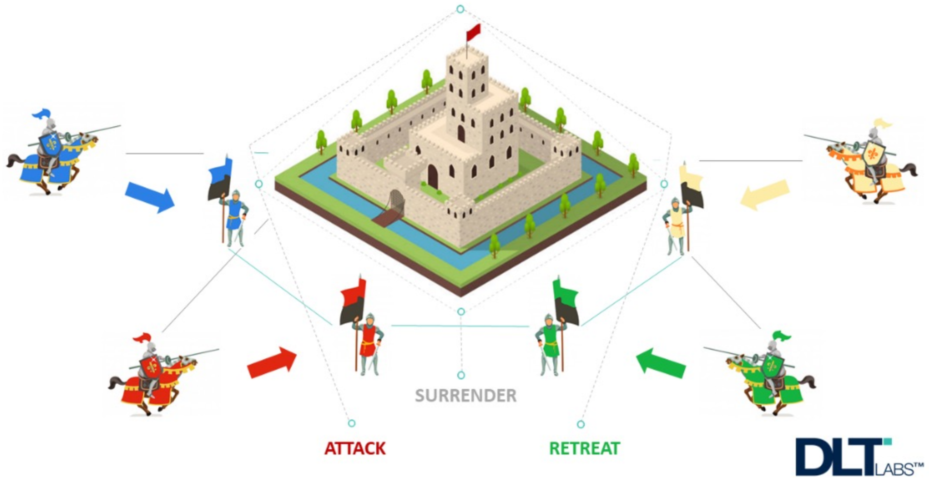


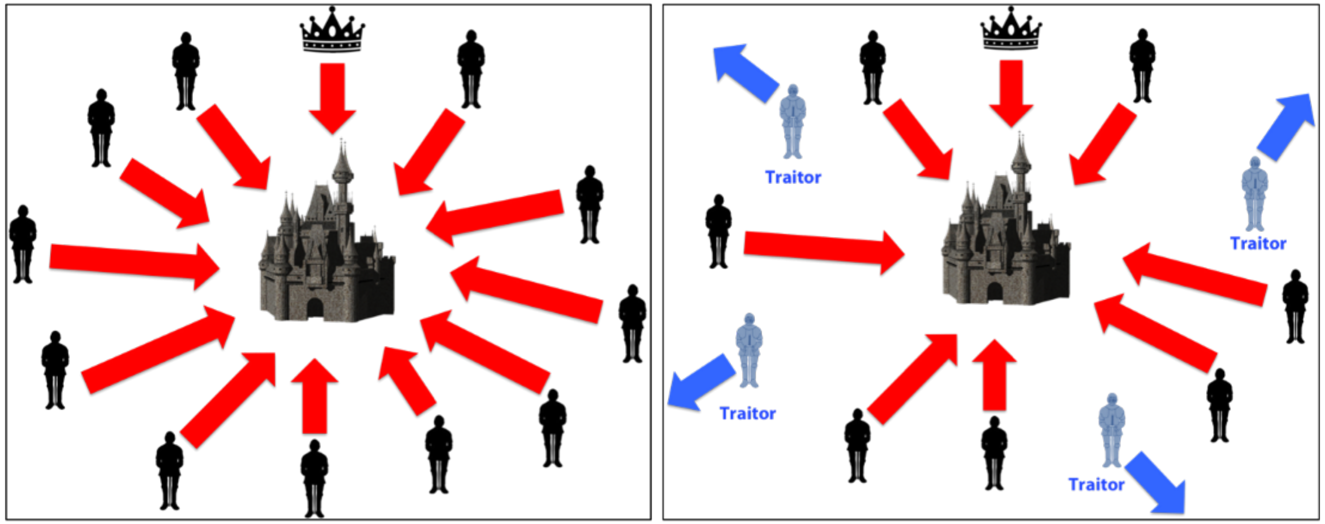
Dr. Lamport

**Byzantine Generals Problem** = How can a distributed system agree on a decision if someone of the participants acts dishonestly or fail

# Byzantine generals problem

## Byzantine Generals’ Problem





Coordinated Attack Leading to Victory

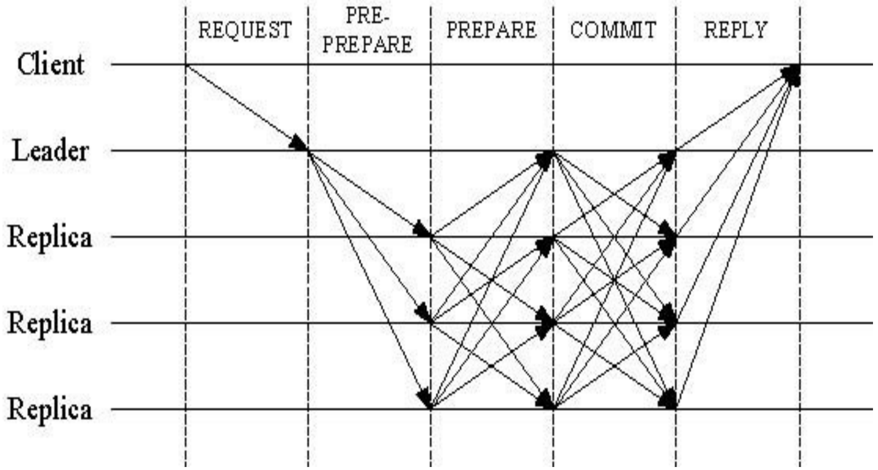
Uncoordinated Attack Leading to Defeat

Source: https://medium.com/@2infiniti/a-primer-to-lft-loop-fault-tolerance-consensus-algorithm-d692bdece85a

# Byzantine fault tolerance

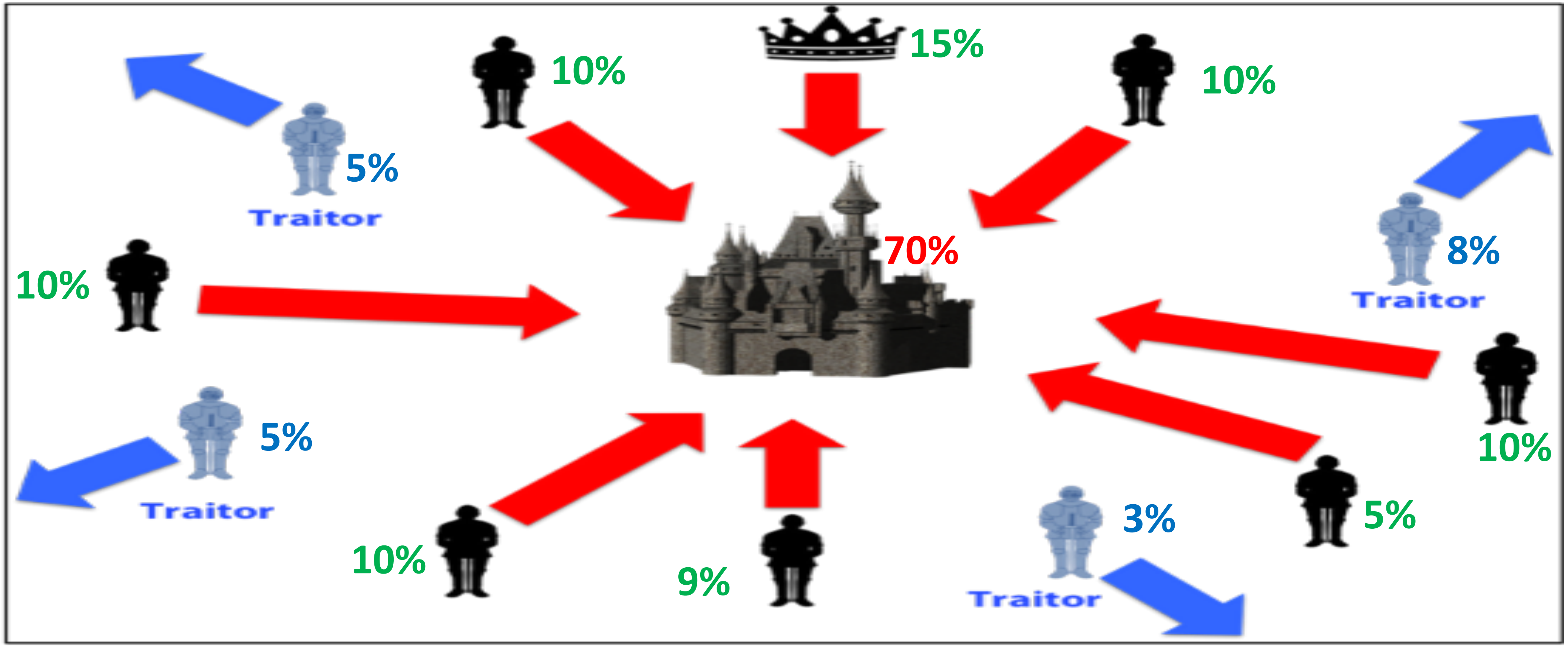
- In a BFT algorithm, we assume there will be Byzantine nodes - network faults or other “bad things” can happen, be it drop, delay of messages, duplication of messages or re-ordering of messages.
- Goal: The basic idea here is to get the same messages from enough nodes to know that non-faulty nodes are in same state

- A client sends a request to invoke a service operation to the leader
  1. The leader multicasts the request to the replicas
  2. Replicas execute the request and send a reply to the client
  3. The client waits for replies from different replicas with the same result; this is the result of the operation.
  4. In a normal case, the leader node run a 3-phase protocol to coordinate the number of replicas



Optional reading: https://www.usenix.org/legacy/events/osdi99/full_papers/castro/castro_html/castro.html

# Byzantine fault tolerance



Adapted from: https://medium.com/@2infiniti/a-primer-to-lft-loop-fault-tolerance-consensus-algorithm-d692bdece85a

# practical Byzantine Fault Tolerance (pBFT)

- Introduced in the late 90s by Barbara Liskov and Miguel Castro
- pBFT was designed to work efficiently in asynchronous  systems.
- Optimized for low overhead time
- Goal: to solve many problems associated with already available Byzantine Fault Tolerance solutions.
- Application areas include distributed computing and blockchain.

Optional reading: https://www.usenix.org/legacy/events/osdi99/full_papers/castro/castro_html/castro.html

# Proof of Work (PoW)

- Selects a miner for the next block generation.
- One miner (prover) demonstrates to the other miners (verifiers) that that a certain amount of a specific computational effort has been expended.
- The fundamental concept behind this algorithm is to overcome a complex mathematical problem and give a solution easily.
- The complexity of the “puzzle” depends on the number of participants, the existing power and the network load.
- The hash of each block includes the hash of the previous block, which increases security and avoids any block breach from happening.
- This mathematical puzzle requires a lot of computational power and thus the node that solves the puzzle gets to mine the next block (Lucas and Paez, 2019).
- Proposed by Nakamoto’s paper for the Bitcoin blockchain network.

# Proof of Stake (PoS)

- Proposed as an alternative to PoW
- Instead of investing in costly hardware to solve a complicated puzzle, validators depositing and locking coins as a stake.
- Staked coins are used to validate transaction and secure the network.
- Rewards are offered to incentivize users to lock up their coins.
- Once validators stake their coins, they are allowed to start validating the blocks.
- Validators are selected randomly to validate or mine a new block
- All validators get a reward proportionate to their bets based on the actual blocks added in the blockchain, and their stake increases accordingly.
- These percentages may differ from one blockchain to one other.
- Example: a validator who stakes 2% of the coins available can mine only 2% of the blocks.

# Avalanche

# Avalanche - key features

AVALANCHE

- Proposed in 2018 and launched on mainnet in September 2020 by Ava Labs
- Layer 1 blockchain and smart contracts platform
- Supports the development of multi-functional Dapps and enterprise blockchain solutions
- Compatibility with the EVM → developers port Dapps over from Ethereum
- Consensus: PoS

**Key features**

- Low latency block time (~1second)
- Highly Scalable
- High performance
- High throughput
- Interoperable
- Efficient transaction ecosystem
- 4,500 TPS
- Fast
- Eco-friendly
- Low cost

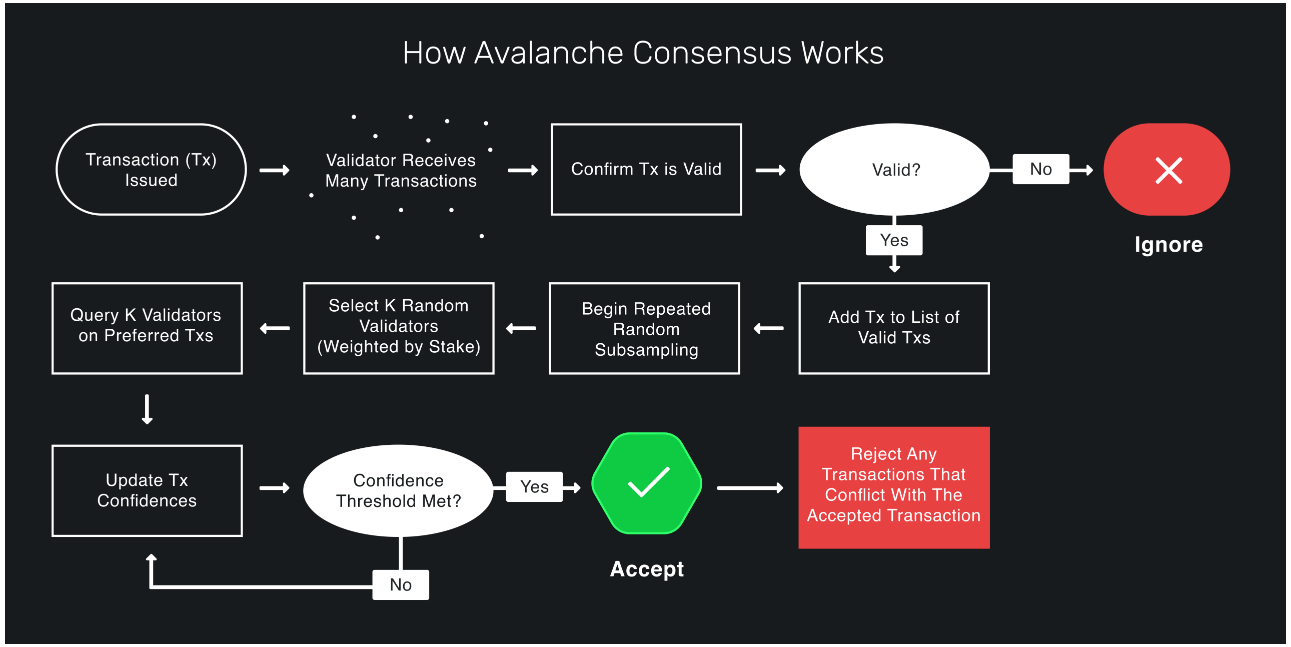
# Avalanche – consensus PoS

AVALANCHE

- Users can support Avalanche by staking their coins
- Staked coins are locked and cannot be used during the locking period
- Two ways to stake AVAX: Validators or Delegators
- **Validators**: validate and secure the chain (create new blocks, process transactions)
  - Validators can be considered as active nodes
  - To get a reward, validators should be correct and online at least 80% of the time.
- **Delegators** support the network, but they have a passive role. They trust an existing node to validate; thus they delegate their staked AVAX to the delegator and get rewarded for their support.
  - Minimum delegation fee rate is 2%
  - Staking period: From 2 weeks to 1 year
  - Validators stake minimum 2000 AVAX  and  Delegators at least 25 AVAX
  - Maximum weight for validators= (Validators stake + delegations) is minimum 3m AVAX or 5 times the stakes

# Avalanche – consensus PoS (Cont.)

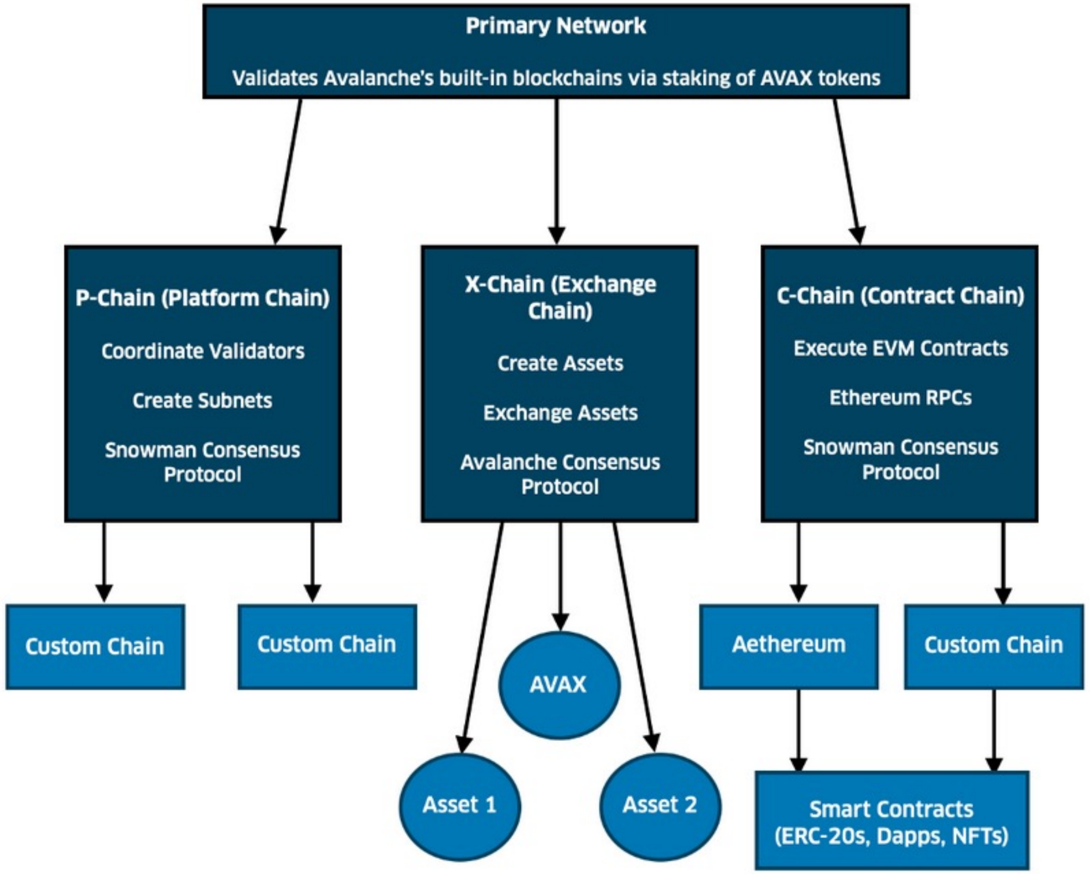
- In brief, Avalanche uses a consensus algorithm that is based on a technique know as sub-sampled voting (to be exact, the technique is known as "repeated sampling").
- They key aspect of the proposed consensus algorithm is making this technique scalable so that it is suitable to permissionless systems
  - The repeated random sub-sampling process is used to build confidence that a transaction is valid through the network's consensus.


How Avalanche Consensus Works
Transaction (Tx) Issued
Validator Receives Many Transactions
Confirm Tx is Valid
Valid?
No
Ignore
Yes
Add Tx to List of Valid Txs
Begin Repeated Random Subsampling
Select K Random Validators (Weighted by Stake)
Query K Validators on Preferred Txs
Update Tx Confidences
Confidence Threshold Met?
Yes
No
Accept
Reject Any Transactions That Conflict With The Accepted Transaction

Source: https://docs.avax.network/

# Avalanche - architecture

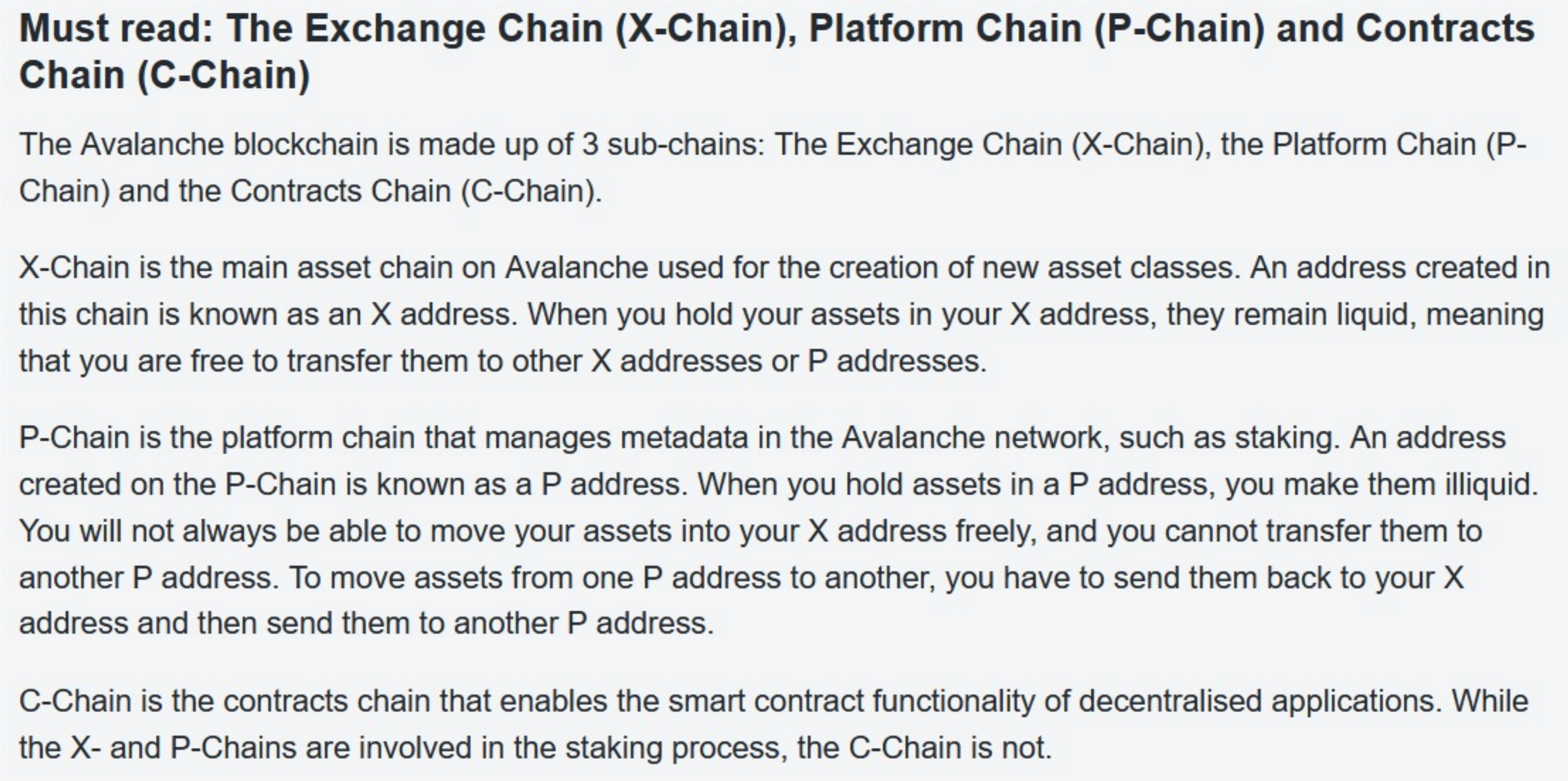
AVALANCHE



- Unique approach
- 3 integrated blockchains
- Validated by a common set of validators:
  - P-Chain: Creates subnets
  - X-Chain: Creates and exchanges assets
  - C-Chain: EVM contract execution
- Most of the activity happens in C-chain

- **Subnets:** allow developers to build customizable implementations with custom rulesets.

# Avalanche - architecture



### Must read: The Exchange Chain (X-Chain), Platform Chain (P-Chain) and Contracts Chain (C-Chain)

The Avalanche blockchain is made up of 3 sub-chains: The Exchange Chain (X-Chain), the Platform Chain (P-Chain) and the Contracts Chain (C-Chain).

X-Chain is the main asset chain on Avalanche used for the creation of new asset classes. An address created in this chain is known as an X address. When you hold your assets in your X address, they remain liquid, meaning that you are free to transfer them to other X addresses or P addresses.

P-Chain is the platform chain that manages metadata in the Avalanche network, such as staking. An address created on the P-Chain is known as a P address. When you hold assets in a P address, you make them illiquid. You will not always be able to move your assets into your X address freely, and you cannot transfer them to another P address. To move assets from one P address to another, you have to send them back to your X address and then send them to another P address.

C-Chain is the contracts chain that enables the smart contract functionality of decentralised applications. While the X- and P-Chains are involved in the staking process, the C-Chain is not.

https://www.finder.com.au/how-to-stake-avalanche

# Cardano

# Cardano

- Cardano project began in 2015 and founded by Charles Hoskinson (Ethereum co-founder)
- Launched in 2017 (IOHK, Cardano Foundation, Emurgo)
- Considered as 3rd generation blockchain (1st generation – e.g., bitcoin, 2nd generation – e.g., Ethereum)
- Combines the characteristics of the previous 2 generations
- **Main aim:** to address issues faces by other blockchains (e.g., interoperability, scalability and regulatory compliance)
- Open source
- Decentralised
- Public blockchain
- Consensus: PoS (Ouroboros)
- Cryptocurrency: ADA

# Cardano - key features

- Scalability: Processes large number of transactions by ensuring good performance
- Interoperability:
  - Supports cross chain transactions, multiple token types and smart contract languages
  - Users can interact with multiple coins across various blockchains
- This is mainly achieved by Cardano's design which comprises of two layers:
  - Settlement Layer (CSL): The CSL is used to transfer ADA between accounts and to record transactions.
  - Computation Layer (CCL): The CCL contains the smart contract logic that developers can leverage to programmatically move funds.

○ To guarantee sustainability, the treasury system is controlled by Cardano community and refilled with new mined Ada that are used as additional funding.

# Cardano - benefits

CARDANO

- System design:
  - Cardano is written in Haskel (functional design language)
  - New components are tested in isolation
- Security is guarantee through the Ouroboros (Cardano PoS protocol)
- Decentralization:
  - 2000+ distributed stake pools run by community
  - Blocks and transactions are validated by the community – no central authority
- Seamless upgrades:
  - There are differences in hard forks compared to other “traditional” blockchains
  - Cardano uses hard fork combinator that combines protocols, and allows transitions without system interruption or restart.
- Low power consumption
- Applied research

# Cardano – development themes

- **Shelley** — decentralization
- **Byron -** foundation establishment
- **Goguen** — smart contracts
- **Basho** – scalability
- **Voltaire** — governance

# Cardano – development themes

CARDANO

**Shelley** — decentralization

- Introduced a decentralised ledger
- Guarantees enhanced user experience in:
  - stake pool operation,
  - delegation preferences,
  - incentives

**Goguen** — smart contracts

- multi-functional system for DApps building,
- smart contract support,
- Custom token issuance

**Byron -** foundation establishment

- Based on PoS
- Allows users to buy and sell Ada
- Delivered  Daedalus and Yoroi wallets
- Provides block explorer to browse the blockchain

- **Basho** - scalability
- Improve scalability and interoperability
- Improve underlying performance

**Voltaire** — governance

- Decentralised governance & decision making

# Cardano – PoS Consensus

- Orobouros is the proof-of-stake (PoS) consensus algorithm used by Cardano
  - Divides time into epochs and slots
  - Within each slot, a slot leader is randomly chosen and is responsible for selecting the blocks to be appended on the blockchain (Note: only mCore nodes can be elected to become slot leaders)
- Types of nodes:
  - mCore nodes: Stake ADA tokens and participate in governance
  - Relay nodes: Send data between mCore nodes and the public internet
  - Edge nodes: Broadcast cryptocurrency transactions
- Ouroboros enables two types of blocks that are appended on-chain:
  - Genesis blocks: Include the list of all the slot leaders associated with the epoch and contain a series of main blocks
  - Main blocks: Contain all transaction information, proposals for software updates and the list of votes for these updates.

https://docs.cardano.org/explore-cardano/monetary-policy

# Cardano – monetary policy

CARDANO

- Goal
  - Reward people who participate in the network
  - Secure the funding treasury
- A virtual pot is used for rewards and funding treasury

- Rewards
  - Staking rewards for delegators and stake pool operators:
  - Transaction fees for specific period added to a virtual pot
  - ρ is also added to the pot (ρ = % of ada reserves)
  - Like other blockchains there is a higher reward at the beggining which is progressively reduced

- Funding treasury
  - A % (τ) of the virtual pot for a specific period is transferred to the funding treasury
  - Funding treasury resources are used to develop Cardano (through voting processes)

https://docs.cardano.org/explore-cardano/monetary-policy

# Cardano – transaction fees in Cardano

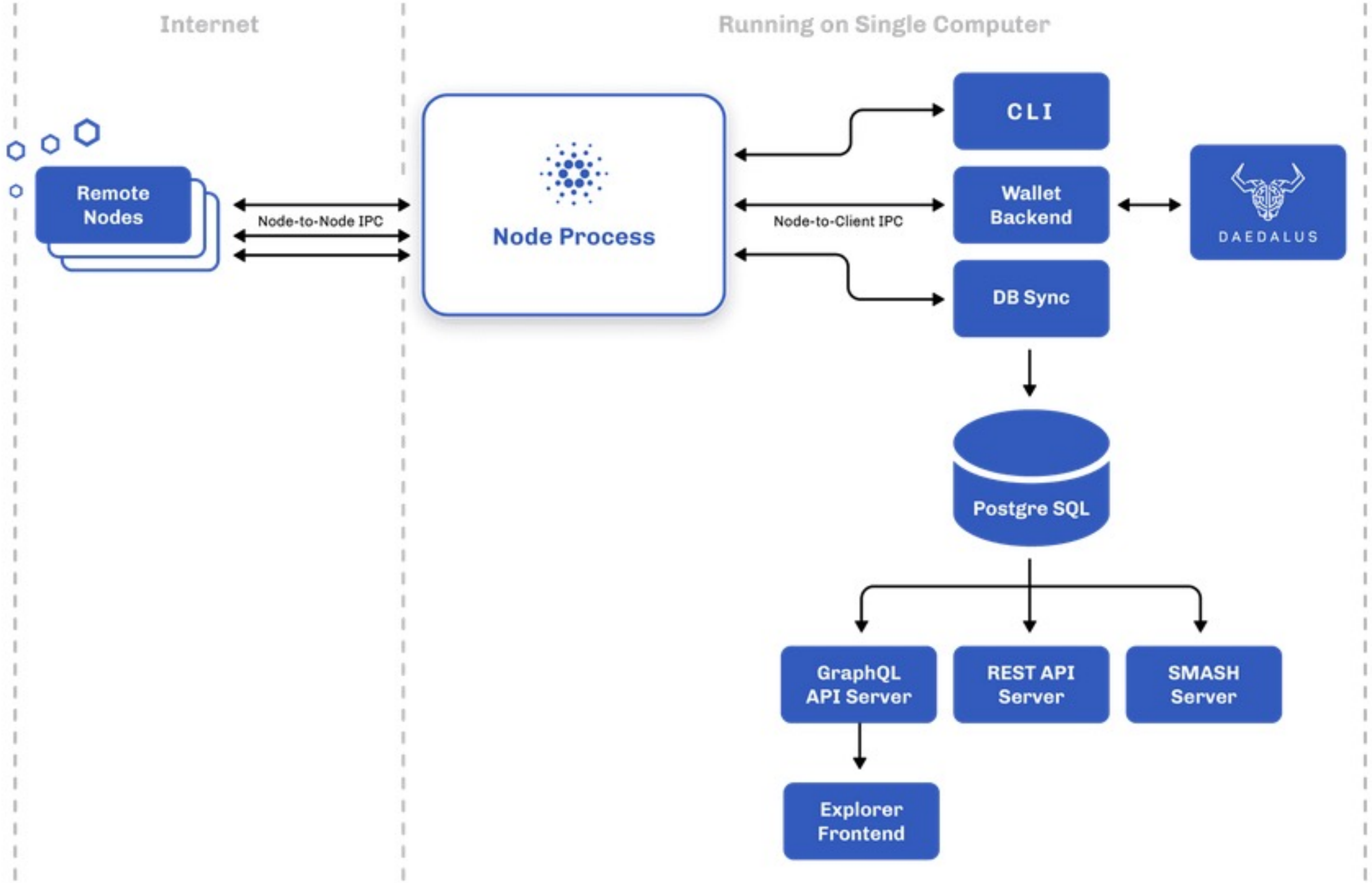
CARDANO

- Transaction fee = T(x)
- Transaction size = size (tx) (calculated in bits) (Cardano transactions have small number of bits)
- a and b parameters = Cardano protocol parameters
- A change in the values of a or b → a hard fork (as it influences the transactions accepted by Cardano)
- a = transaction cost is related to the size of transaction
  - the larger the transaction the more resources we need to store and process it
- b = a fix value (number) proposed by cardano
- b was introduced to prevent Distributed-Denial-of-Service
- It actually eliminates the possibility to have an attacker creating large number (e.g. millions) of small transactions that will crash the system

- $T(x)= (a*size(Tx)) + b$

https://docs.cardano.org/explore-cardano/monetary-policy

# Cardano - architecture



Source: https://docs.cardano.org/explore-cardano/cardano-architecture/overview

# Cardano – architecture components

## 1. Nodes

- Executing the Ouroboros protocol
- Validating and relaying blocks
- Producing blocks (some nodes)
- Providing information about the state of the blockchain to other local clients

## 2. Node process – consists of

- consensus,
- ledger and networking
- Command Line Interface (CLI)
- logging, and monitoring

## 3. Node-to-Node IPC protocol

- Allows the communication and exchange of blocks and transactions among nodes

## 4. Node-to-Client IPC

- Allows local applications (e.g. wallet backends or blockchain explorers) to interact with the blockchain via the node..
- Applications access the raw chain data and query the current ledger state.
- Provides the ability to submit new transactions to the system

## 4. Command line interface (CLI)

- Text-based but powerful tool
- Does almost everything
- Inconvenient as it lacks GUI

## 5. Daedalus wallet

- Send and receive Ada
- Offers wallet front-end (graphical) for users and backend for system communication

## 6. Cardano-db-sync

- Cardano node stores only the blockchain itself and associated information needed to validate the blockchain.
- This design principle is about minimising code complexity, and reducing computational cost and resource use, to keep the node's local interfaces as minimal as possible and to use external clients to provide a variety of convenient interfaces and extra functionality.
- historical blockchain information is provided by a separate component using an Structured Query Language (SQL) database.

Source: https://docs.cardano.org/explore-cardano/cardano-architecture/overview

# Cosmos

# A quick review of blockchain architectural layers

COSMOS

### Fundamental Layers

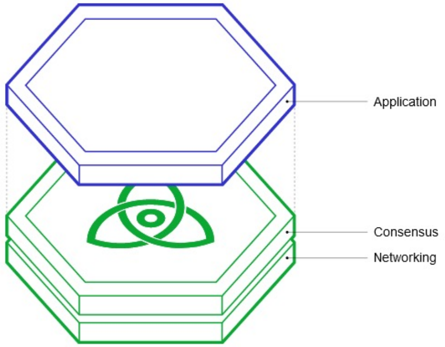
- **Application:** transactions processing, update the state of the system
- **Network:** exchange messages related to transactions and decision making (consensus)
- **Consensus:** supports nodes agreement

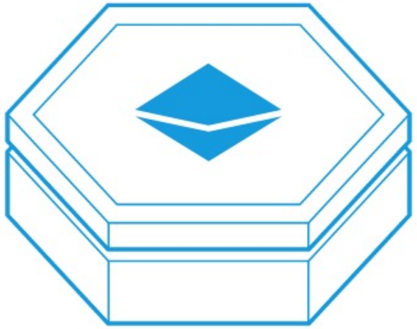
### Bitcoin

- Monolithic
- PoW
- Development Approaches: (a) fork (b) on top
- Bitcoin script – not user friendly

### Ethereum

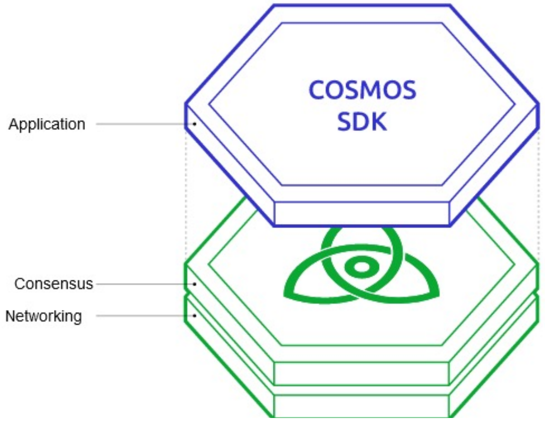
- EVM
- Smart contracts
- Dapps
- Problems
  - Scalability: Dapps 15 TPs
  - Limited flexibility for programmers
  - Sovereignty: all Dapps share  the same underlying environment
  - Ethereum approval is required for any change in EVM





# Cosmos – a novel approach

- Launched: March 2019 (ICO in 2017) - initially built by Tendermint
- Open-source community project
- Aim: Establish an internet of blockchains allowing them to communicate in a decentralised way
- Addresses scalability and interoperability issues of existing blockchains
- Cosmos = Decentralised network of multiple independent blockchains (zones)



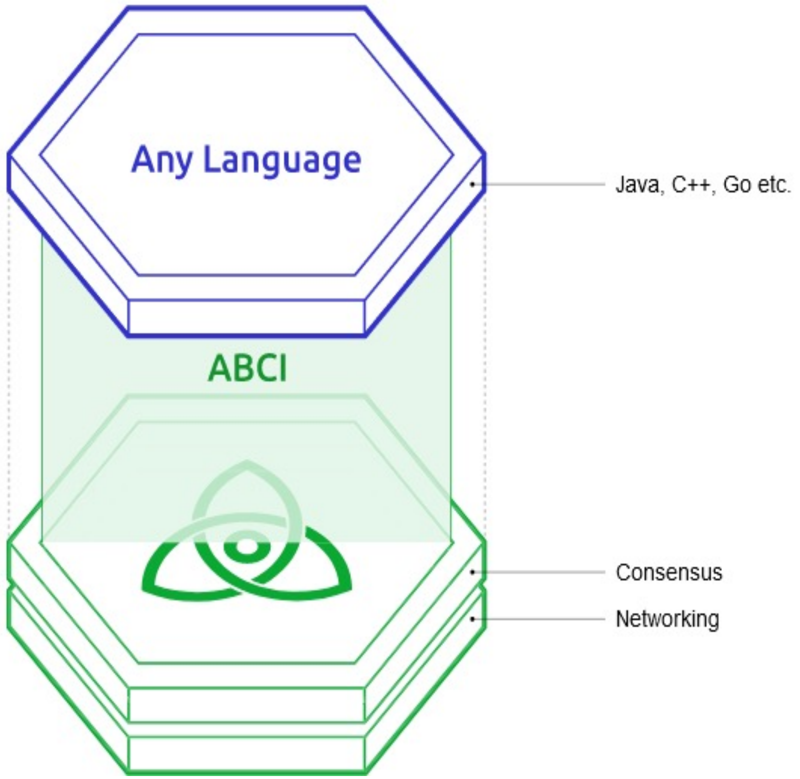
**Key features**

- Speed (transactions processing)
- Blockchains maintain sovereignty
- Excellent solution for multiple projects / cases
- Scalable
- Interoperable
- Tendermint BFT consensus: suitable to scale public PoS chains
- Provides sound solutions for all three blockchain layers
- Open source tools:
  - Tendermint
  - Cosmos SDK
  - Inter Blockchain Communication (IBC)

# Cosmos – Tendermint BFT

COSMOS

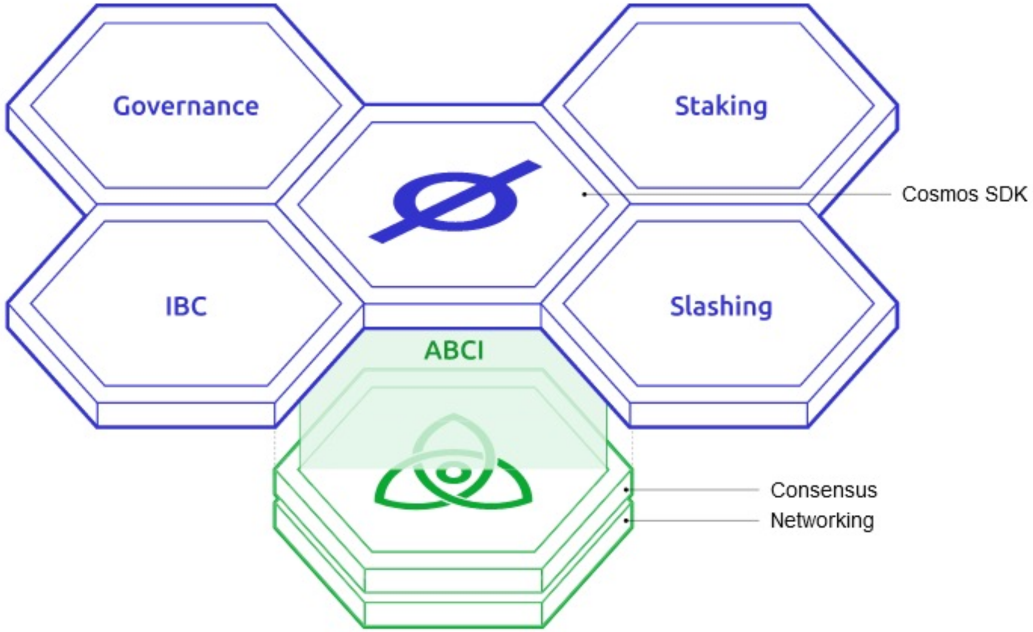
- Tendermint BFT consensus: suitable to scale public PoS chains
- Overcomes the monolithic nature of bitcoin or Go-Ethereum
- Handles networking and consensus as a generic engine (Tendermint BFT engine)
- Speeds up development time
- Tendermint BFT engine connects with Application Blockchain Interface (ABCI) protocol
- Allows programming in any language
- Build private and public blockchain solutions (developers work on application layer and configure appropriately).
- High performance (thousands of transactions per second)
- Faster (instant infinity due to tendermint algorithm)
- Security: accountability and fault tolerance



https://v1.cosmos.network/intro

# Cosmos – Cosmos SDK

- Cosmos SDK = cryptocurrency application framework
  - allowing developers to build blockchains using the Tendermint consensus algorithm
- Shortens the process of building secure blockchain applications on Tendermint BFT
  - Offers common blockchain protocol functionality (i.e., staking, governance, tokens).
  - Developers can create plugins to add any additional feat
- Modularity:
  - Easy to build modules for Cosmos SDK
  - Easy to reuse code
- Ethermint* ports EVM into a Cosmos SDK module
- Import codebase from Golang



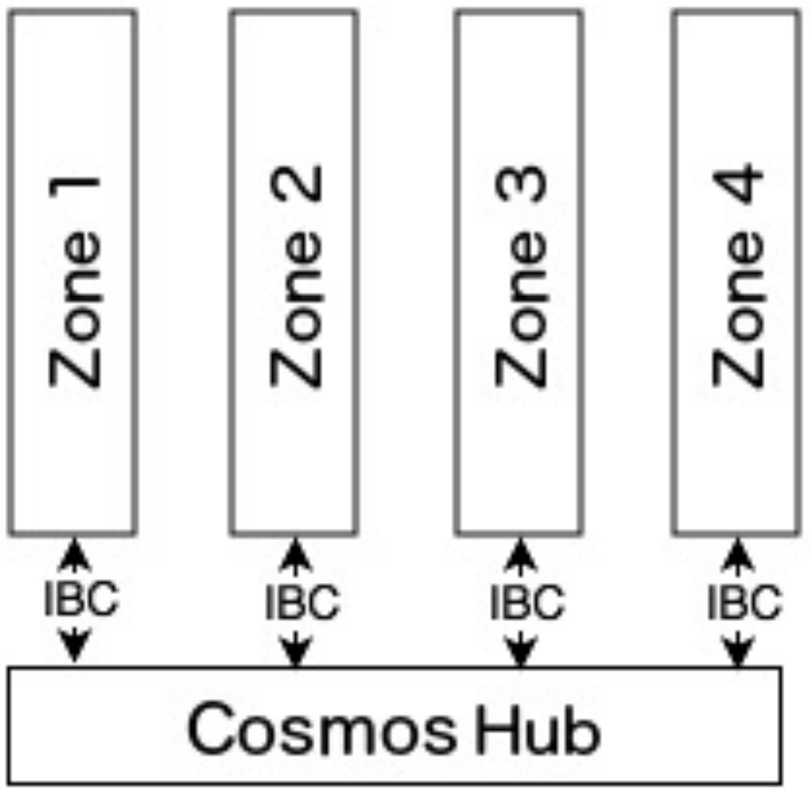
- *Ethermint = Ethereum on Tendermint

https://v1.cosmos.network/intro

# Cosmos - architecture

COSMOS

- Main components
- Cosmos Hub
  - Multiple assets PoS
  - Has simple governance mechanism
  - Enables network upgrades
- IBC
  - Behaves like a “TCP” for blockchains
- Zones
- These are other blockchains operate in parallel – connected to cosmos
- Secure move of tokens from one zone to another
- Cosmos hub: token logistics about interzone tokens transactions



https://v1.cosmos.network/resources/whitepaper

# Cosmos - Validators

- Validators = users who lock an amount of tokens  for a certain period of time
- Bond transaction = a transaction that locks the coins
- Bond transaction # Unbond transaction
- After unbonding tokens remain locked for a predefined period (unbonding period). Then the user can spend them
- Validators Participate in voting to agree on the next block
- Validator's voting power = number of tokens that s/he locked
- A new block is append when there is a 2/3 majority of validators
- Fork happens when there are two blocks at the same heigh with 2/3 votes

# Cosmos – Atom token

- Cosmos native token for Cosmos Hub
- Initial Coin Offering (ICO) in 2017
- ATOM token is mainly used:
  - as a spam-prevention mechanism,
  - as staking tokens, and
  - as a voting mechanism in governance

# Self-Assessment Exercises and Further Readings

# Self-Assessment Exercise

In this session, we exposed several features that distinguish distributed ledgers as permissioned or permissionless and examined notable use-cases. For this self-assessment exercise, choose a blockchain/DLT protocol of your preference and perform an exploratory study that covers:

- Background information about the blockchain you choose
- Type of governance (e.g., permissioned or permissionless, public or private)
- Characteristics and key features
- Consensus mechanism used
- Governance, voting scheme, participation levels for nodes
- Areas of application
- Describe and analyse the architecture
- Compare with other DLTs or blockchains
- Report exemplar use cases and explain their functionality and instantiation of this blockchain
- Pros vs Cons vs Trade-offs

# Conclusions

# Conclusions

- Permissioned and permissionless blockchains differ in terms of access and validations
- In this session we examined 3 layer 1 blockchains namely: Avalanche, Cardano and Cosmos
- Avalanche: Supports the development of multi-functional Dapps and enterprise blockchain solutions. It is based on PoS consensus algorithm and its coin is AVAX.
- Cardano is another interesting blockchain that focuses on scalability, interoperability and regulatory compliance
- Cosmos is a novel blockchain framework that increases interoperability and scalability

# Glossary

# Glossary

**Public Blockchains**: provide the foundation for largest interoperability as they serve as a common hub for all connected services; they are censorship resistant, without entities with special access, consequently enjoy increased trust; suitable for global deployment, like digital currencies, global computation platforms or data repositories.

**Private Blockchains:** lower costs (vs. public), as validation is performed without external threats – minimized validation; high adaptability and high throughput as the system is easier to update and optimize; - suitable for managing data coming from dispersed sources, like branch offices or complex supply chains, where the origin and the transport of data could be compromised

**Consortium Blockchains:** lower costs, higher speed and increased security for: sharing data, communicating and reconciling between organizations in a certain ecosystem as there is one shared database, which is trusted by all participants

**Staking:** The process of depositing and locking up cryptocurrency / tokens to participate in a blockchain's Proof-of-Stake (PoS) consensus mechanism

# Glossary

**Proof of Burn (PoB):** Validators “burn” coins instead of investing in costly hardware equipment by sending them to an address where they are irretrievable from. By committing the coins to an unattainable address, validators receive a system-based privilege to mine on a random selection process. Therefore, burning coins means validators have a long-term commitment in return for their short-term losses. Depending on how the PoB is implemented, miners may burn the Blockchain application's native currency or an alternative chain's currency, such as bitcoin. The more coins they burn, the greater their chances of being selected for the next block to be mine.

**Proof of Capacity:** Validators are expected to spend their hard-drive space instead of investing in costly hardware or burning coins (BiKi.com, 2020). The more hard-drive space the validators provide, the higher their chances of being selected for mining the next block and winning compensation for the deposit.

# Glossary

**Proof of Elapsed Time:** PoET is considered as one of the fairest CA which chooses the next block using only fair means. It is usually used in in permissioned blockchains where every validator on the network gets a fair chance to propose the next block. To be more specific, the nodes in the network are waiting for a random timeframe, and then adding a proof of that time in the next bloc. Then, the generated blocks are broadcasted in the network in order to be considered by the rest of the participants. The node that “wins” the proposition of the next block, is the one having the least timer value in the block. Finally, the block of the winner node is then attached on the chain. Moreover, PoET provides additional checks and thresholds in order to avoid nodes from always winning. (Chen et al., 2017)

# Further Readings

# Further Reading

- [Xu et al., 2019] Xu, X., Weber, I., & Staples, M. (2019). Architecture for blockchain applications. Springer.
- [Zheng, 2017] Zheng, Zibin, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An overview of blockchain technology: Architecture, consensus, and future trends." In 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557-564. IEEE, 2017.
- [Xu et al., 2017] Xu, Xiwei, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso, and Paul Rimba. "A taxonomy of blockchain-based systems for architecture design." In 2017 IEEE International Conference on Software Architecture (ICSA), pp. 243-252. IEEE, 2017.
- [Ioini, 2018] El Ioini, Nabil, and Claus Pahl. "A review of distributed ledger technologies." In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 277-288. Springer, Cham, 2018.
- Swanson, Tim. "Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems." Report, available online, Apr (2015).
- Avalanche consensus protocol whitepaper: https://arxiv.org/pdf/1906.08936.pdf

UNIVERSITY *of* NICOSIA

# Questions?

**Contact Us:**

Twitter: @mscdigital
Instructor’s Email: christodoulou.kl@unic.ac.cy

Course Support:
Mark Wigmans - wigmans.m@unic.ac.cy
Marios Touloupos - touloupos.m@unic.ac.cy

IT & live session support: dl.it@unic.ac.cy