DGT

The Blockchain **Handbook**

everything you'd like to ask

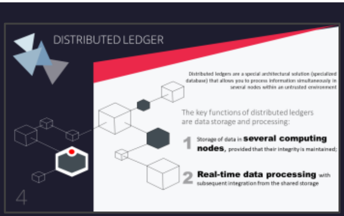
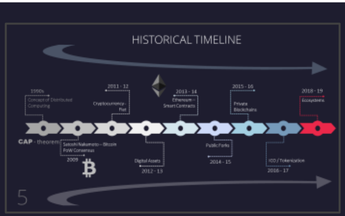
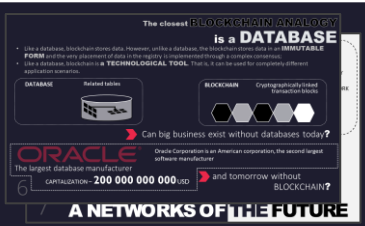
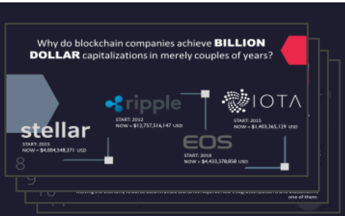
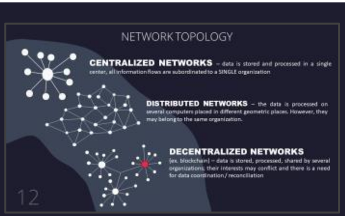
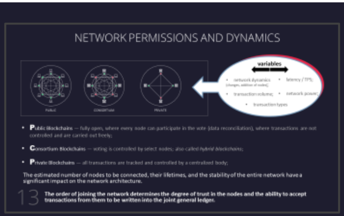
Blockchain is a decentralized, distributed & immutable digital ledger consisting of records packed into linked blocks that are written and read on multiple nodes (computers).

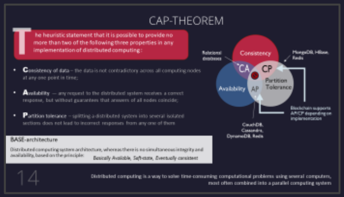
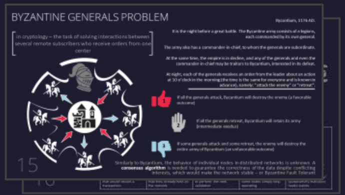
Blockchain is also the de facto name of a class of technologies that support distributed computing and information storage on several nodes. A more accurate title for these is DLT: **Distributed Ledger Technology**

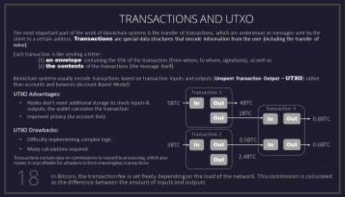
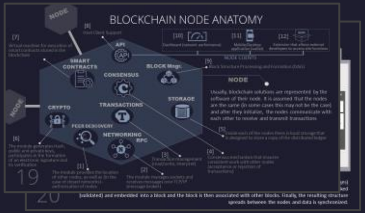
DGT is a distributed computing platform based on DLT. In this document, we have collected basic information addressed to a wide range of readers who are beginning to get acquainted with the amazing decentralized world.

HANDBOOK CONTENTS

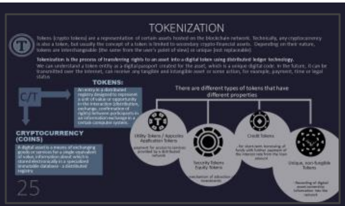
ABOUT BLOCKCHAIN

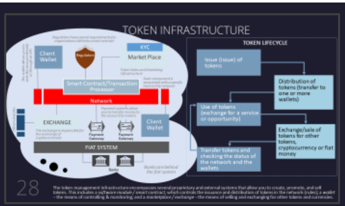







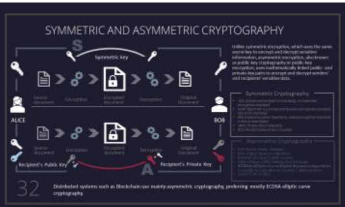
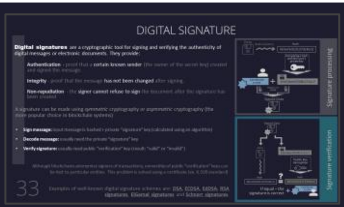
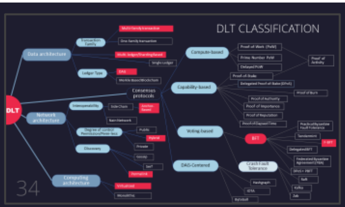



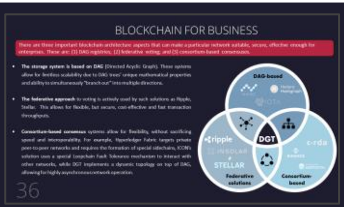
TOKENIZATION



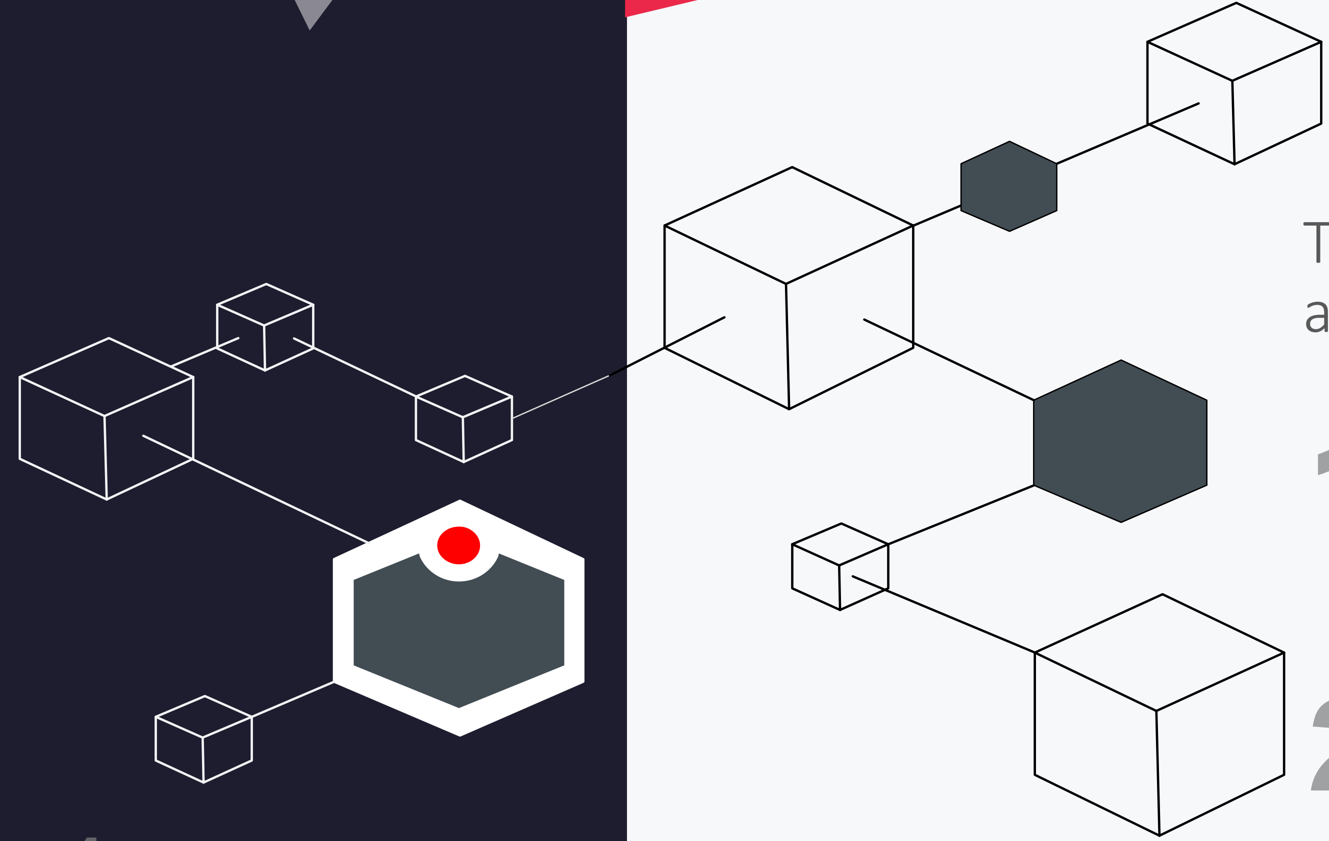


BLOCKCHAIN: ADVANCED



DISTRIBUTED LEDGER



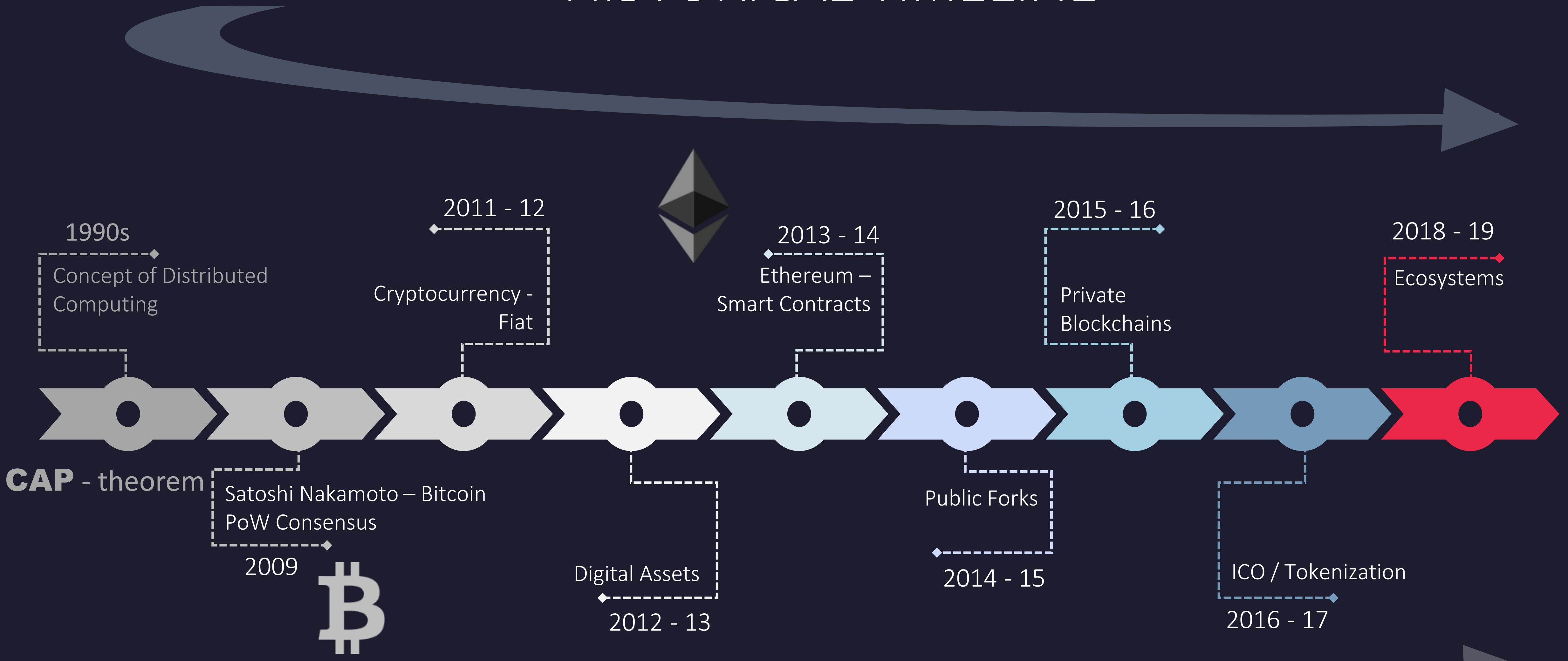
Distributed ledgers are a special architectural solution (specialized database) that allows you to process information simultaneously in several nodes within an untrusted environment

The key functions of distributed ledgers are data storage and processing:

1 Storage of data in **several computing nodes**, provided that their integrity is maintained;

2 **Real-time data processing** with subsequent integration from the shared storage

HISTORICAL TIMELINE



1990s
Concept of Distributed Computing

CAP - theorem

2009
Satoshi Nakamoto – Bitcoin
PoW Consensus

2011 - 12
Cryptocurrency - Fiat

2012 - 13
Digital Assets

2013 - 14
Ethereum – Smart Contracts

2014 - 15
Public Forks

2015 - 16
Private Blockchains

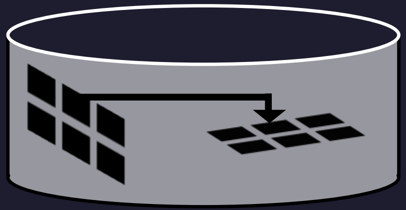
2016 - 17
ICO / Tokenization

2018 - 19
Ecosystems

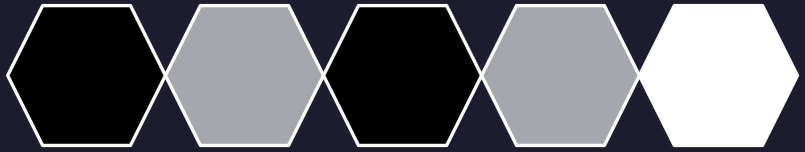
The closest BLOCKCHAIN ANALOGY is a DATABASE

- Like a database, blockchain stores data. However, unlike a database, the blockchain stores data in an **IMMUTABLE FORM** and the very placement of data in the registry is implemented through a complex consensus;
- Like a database, blockchain is **a TECHNOLOGICAL TOOL**. That is, it can be used for completely different application scenarios.

DATABASE Related tables



BLOCKCHAIN Cryptographically linked transaction blocks



Can big business exist without databases today**?**

ORACLE®

The largest database manufacturer

Oracle Corporation is an American corporation, the second largest software manufacturer

CAPITALIZATION – **200 000 000 000** USD

and tomorrow without BLOCKCHAIN**?**

Blockchain is a:

- New economic model
- Zero-touch network
- New program-architectural paradigm
- Immutable repository of information
- Information processing using the consensus algorithm
- Distributed virtual computer executing smart contracts

Blockchain is NOT:

- BITCOIN OR OTHER CRYPTOCURRENCY
- ETHEREUM OR OTHER PUBLIC NETWORK
- SIMPLE DATABASE
- ANOTHER CORPORATE SYSTEM

BLOCKCHAIN - IS **A NETWORKS OF THE FUTURE**

Why do blockchain companies achieve **BILLION DOLLAR** capitalizations in merely couples of years?

ripple

START: 2012
NOW = $12,757,516,147 USD

IOTA

START: 2015
NOW = $1,403,365,129 USD

stellar

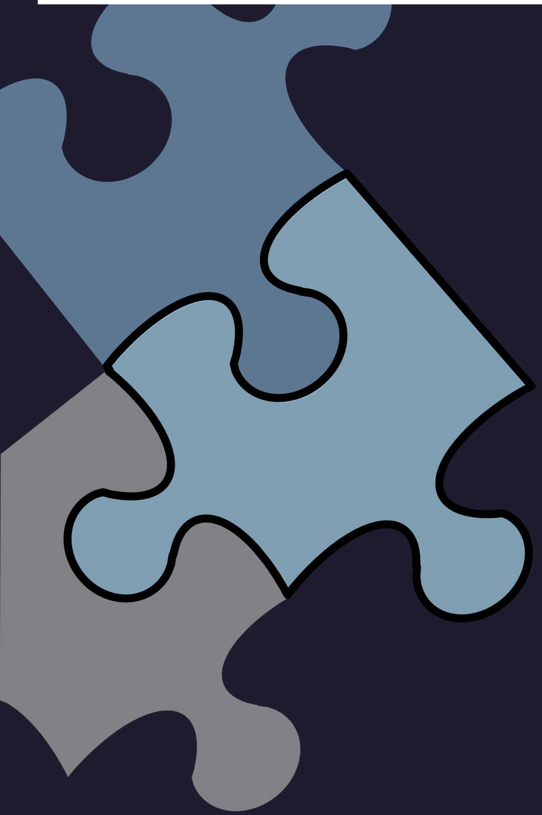
START: 2015
NOW = $4,004,548,371 USD

EOS

START: 2018
NOW = $4,433,378,858 USD

Why these solutions are popular:

Blockchain allows you to combine disparate solutions into a single bundle:



Modern organizations add value to their existing assets by increasing:

INTEROPERABILITY

IN THE NEW

DIGITAL REALITY

BLOCKCHAIN ADVANTAGES

Transaction costs

The cost of computing inside the blockchain is minimal and can even be zero

Transaction speed

Speeds can reach millions of transactions per second with the right network architecture

Risk mitigation

Increased transparency of operations; the data itself is unchanged, protected from counterfeiting and fraud

Confidentiality

Blockchain operates only with keys, while the identities of users are known only to the final businesses in the value chain

Reducing barriers

Decentralization easily attracts new participants, and each grows the value of the network

Blockchain solutions have many advantages. Still, they are only needed when working in a distributed environment without trust.

In all other cases, databases show better performance and efficiency.

Moving the economy towards decentralized scenarios requires new integration patterns and blockchain is one of them.

DECENTRALIZED NETWORKS

Decentralized networks eliminate

INTERMEDIARIES and REDUNDANCY

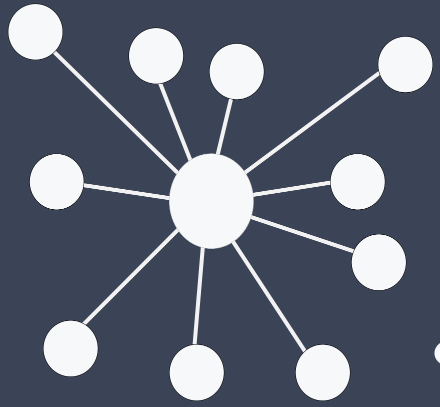
and allow businesses to easily attract

NEW PARTNERS

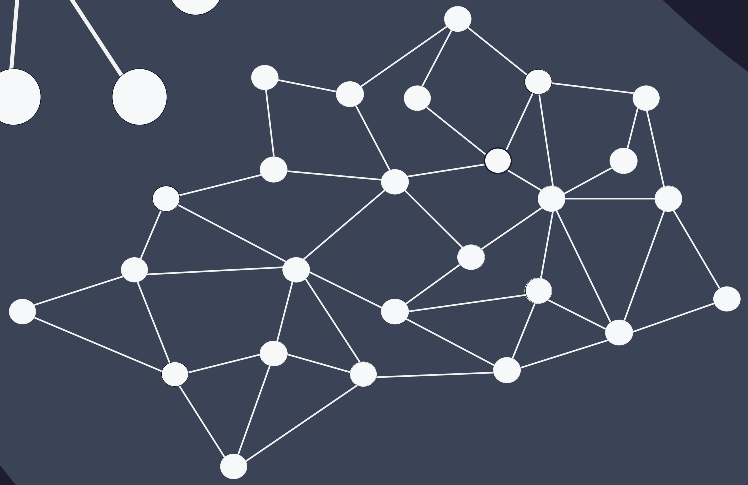
BLOCKCHAIN IS **INDISPENSABLE** WHERE YOU HAVE :

- Many participants over a non-trusted network;
- A conflict environment; contradictions in the motivation of participants;
- The presence of digital entities (electronic money, records);
- A need for a single standard

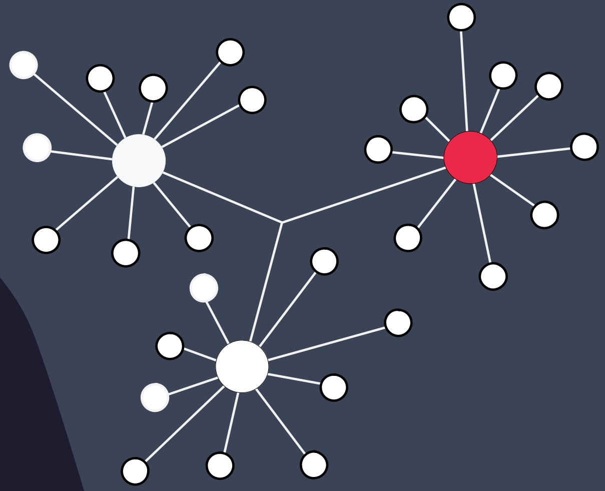
NETWORK TOPOLOGY



CENTRALIZED NETWORKS – data is stored and processed in a single center, all information flows are subordinated to a SINGLE organization



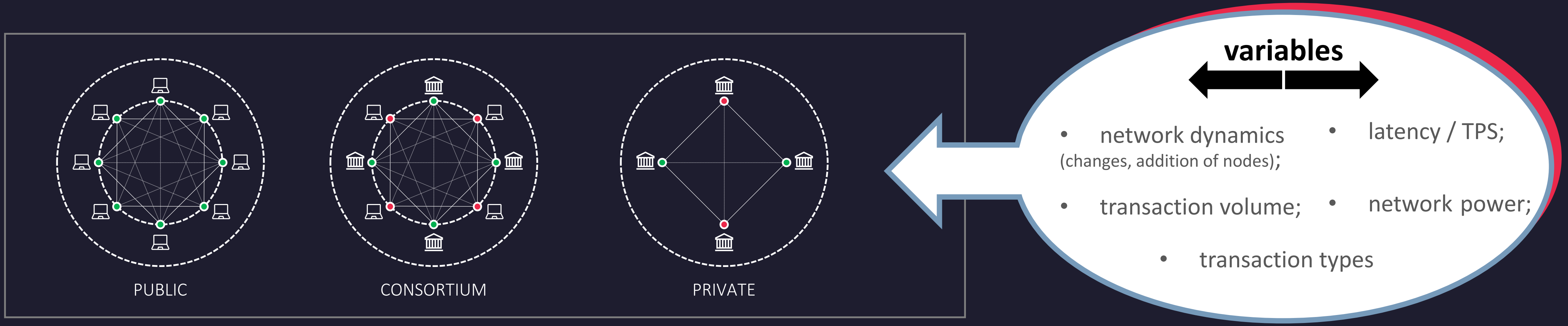
DISTRIBUTED NETWORKS – the data is processed on several computers placed in different geometric places. However, they may belong to the same organization.



DECENTRALIZED NETWORKS

(ex. blockchain) – data is stored, processed, shared by several organizations; their interests may conflict and there is a need for data coordination / reconciliation

NETWORK PERMISSIONS AND DYNAMICS



- network dynamics (changes, addition of nodes);
- latency / TPS;
- transaction volume;
- network power;
- transaction types

- **P**ublic Blockchains — fully open, where every node can participate in the vote (data reconciliation), where transactions are not controlled and are carried out freely;
- **C**onsortium Blockchains — voting is controlled by select nodes; also called *hybrid blockchains*;
- **P**rivate Blockchains — all transactions are tracked and controlled by a centralized body;

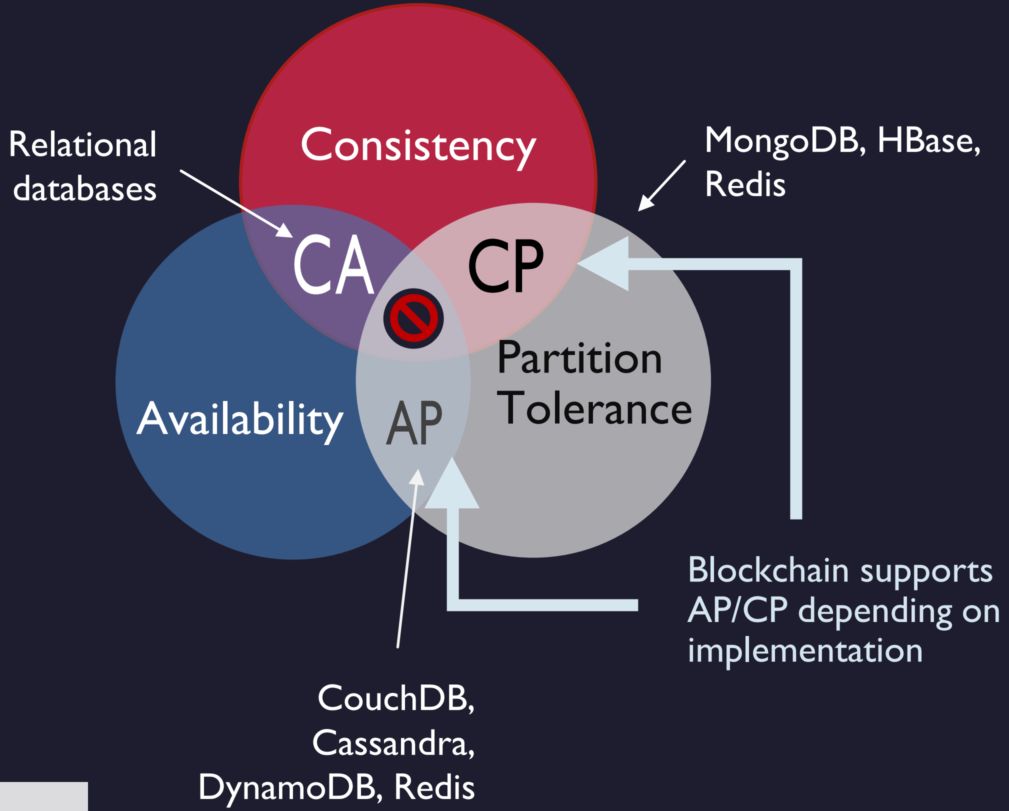
The estimated number of nodes to be connected, their lifetimes, and the stability of the entire network have a significant impact on the network architecture.

The order of joining the network determines the degree of trust in the nodes and the ability to accept transactions from them to be written into the joint general ledger.

CAP-THEOREM

The heuristic statement that it is possible to provide no more than two of the following three properties in any implementation of distributed computing :

- **Consistency of data** – the data is not contradictory across all computing nodes at any one point in time;
- **Availability** — any request to the distributed system receives a correct response, but without guarantees that answers of all nodes coincide;
- **Partition tolerance** – splitting a distributed system into several isolated sections does not lead to incorrect responses from any one of them



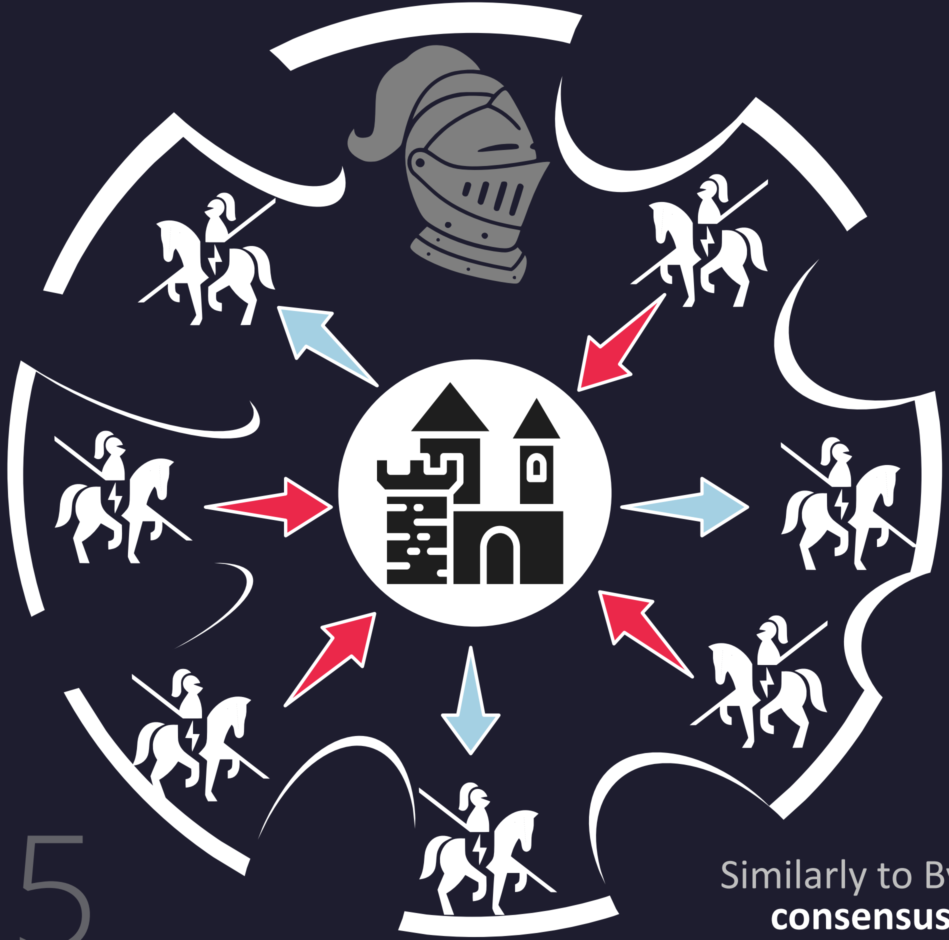
BASE-architecture

Distributed computing system architecture, whereas there is no simultaneous integrity and availability, based on the principle: *Basically Available, Soft-state, Eventually consistent*

Distributed computing is a way to solve time-consuming computational problems using several computers, most often combined into a parallel computing system

BYZANTINE GENERALS PROBLEM

[in cryptology – the task of solving interactions between several remote subscribers who receive orders from one center]



Byzantium, 1176 AD.

It is the night before a great battle. The Byzantine army consists of *n* legions, each commanded by its own general.

The army also has a commander-in-chief, to whom the generals are subordinate.

At the same time, the empire is in decline, and any of the generals and even the commander-in-chief may be traitors to Byzantium, interested in its defeat.

At night, each of the generals receives an order from the leader about an action at 10 o’clock in the morning (the time is the same for everyone and is known in advance), namely: "attack the enemy" or "retreat".

1 If all the generals attack, Byzantium will destroy the enemy (a favorable outcome)

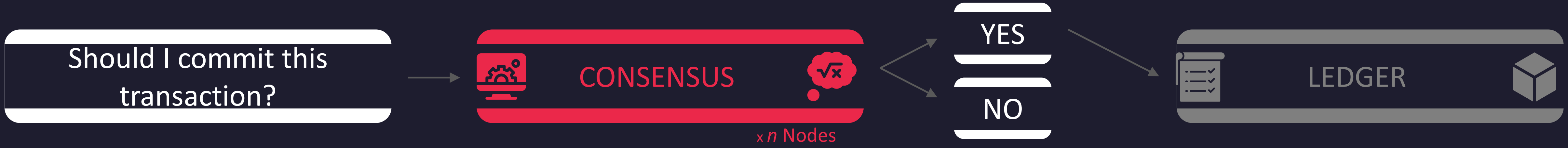
2 If all the generals retreat, Byzantium will retain its army (intermediate exodus)

3 If some generals attack and some retreat, the enemy will destroy the entire army of Byzantium (an unfavorable outcome)

Similarly to Byzantium, the behavior of individual nodes in distributed networks is unknown. A **consensus algorithm** is needed to guarantee the correctness of the data despite conflicting interests, which would make the network stable – or Byzantine Fault Tolerant

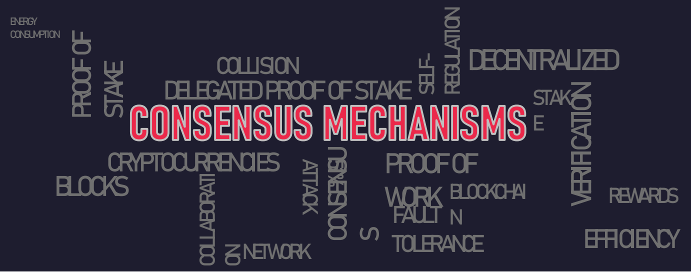
BLOCKCHAIN CONSENSUS

CONSENSUS: a math-based proof mechanism within distributed networks that ensures a set of processes record and preserve a consistent and correct data value proposed by one of those processes; *informally: "we all agree on something"*



The main task of the consensus is to eliminate conflicts and ensure the integrity of data. It must have three core properties:

- **Validity –** the value agreed upon must be proposed by some set process (safety)
- **Termination –** at least one non-faulty process eventually decides (liveness)
- **Agreement –** all deciding processes agree on the same value (safety)



! The consensus mechanism depends on the level of decentralization, distribution; and affects transactions per second, latency, and security

Proof-of-Work (PoW)	Proof-of-Stake (PoS)	Delegated POS (DPoS)	Crash Fault Tolerant (CFT)	Byzantine Fault Tolerant (BFT)
Miners compete using massive computational efforts to find the "winning proof-of-work" that would record a transaction.	Miners can mine or validate block transactions depending on the quantity of crypto that they already hold on the network	Users of the network vote their quantity of crypto holdings to elect delegates ("witnesses") to perform the next validation	A consensus that adds resilience and reaches conclusions even in "crash" events: when some nodes simply stop operating	A consensus that adds resilience and reaches conclusions both in "crash" and "Byzantine" (purposefully malicious node) events

ADDRESSES, PRIVATE AND PUBLIC KEYS

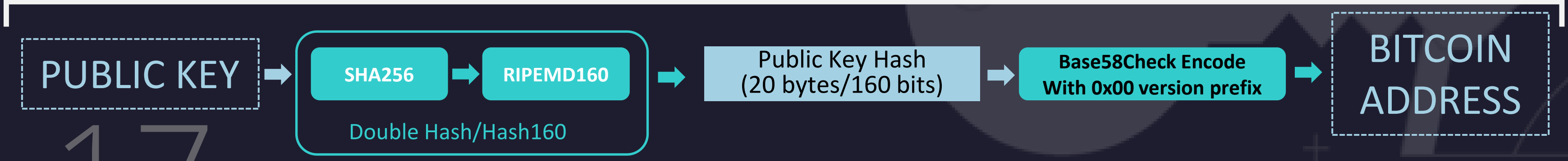
BLOCKCHAIN: Nodes exchange transactions (“**messages**”) signed with the user’s account (“**key pair**”) including an address (“**wallet address**”)

Key Pair – private key & public key. *Depends on the cryptography used in blockchain system, such as elliptic cryptography – ECDSA. Difference curves have different properties; ex. Bitcoin and Ethereum use the secp251k1 curve.*

- **Private key** – a large random number represented as a 256-character binary code; the “password”
- **Public key** – calculated from the elliptic curve. *The key property of this operation is the impossibility (complexity) of reversal.*

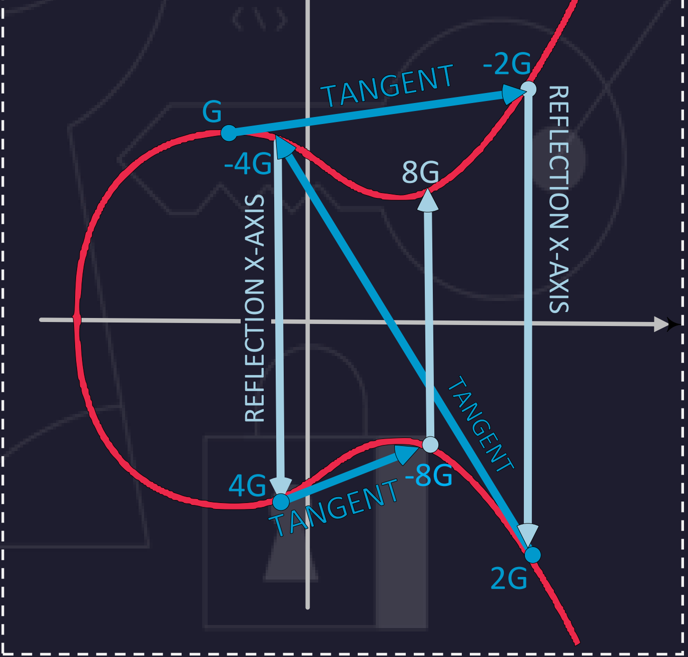
Wallet address – simple public key transformations. *Performed differently on different networks. Ex. Bitcoin derives the address from a public key using one-way cryptographic hashing.*

Ex. Bitcoin address generation



PUBLIC KEY → SHA256 → RIPEMD160 (Double Hash/Hash160) → Public Key Hash (20 bytes/160 bits) → Base58Check Encode With 0x00 version prefix → BITCOIN ADDRESS

Public key calculation over elliptic curve



TRANSACTIONS AND UTXO

The most important part of the work of blockchain systems is the transfer of transactions, which are understood as messages sent by the client to a certain address. **Transactions** are special data structures that encode information from the user (including the transfer of value).

Each transaction is like sending a letter:

(1) **an envelope** containing the title of the transaction (from whom, to whom, signatures), as well as
(2) **the contents** of the transactions (the message itself).

Blockchain systems usually encode transactions based on transaction inputs and outputs (**Unspent Transaction Output – UTXO**) rather than accounts and balances (Account-Based Model)

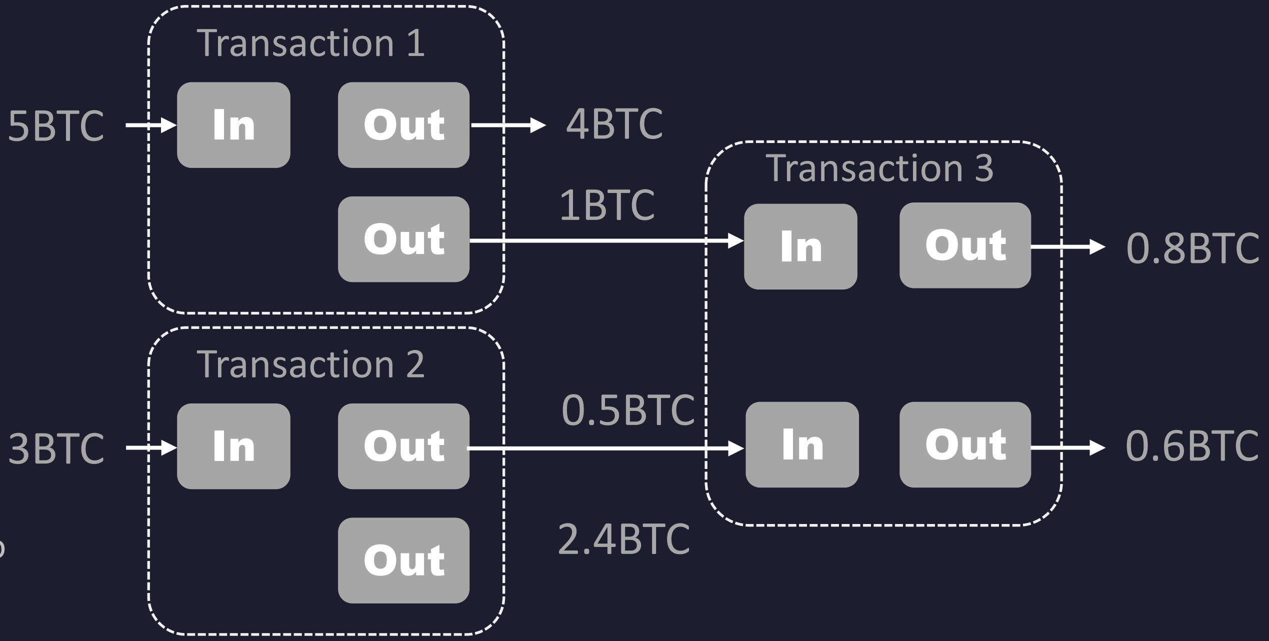
UTXO Advantages:

- Nodes don't need additional storage to check inputs & outputs; the wallet calculates the transaction
- Improved privacy (no account link)

UTXO Drawbacks:

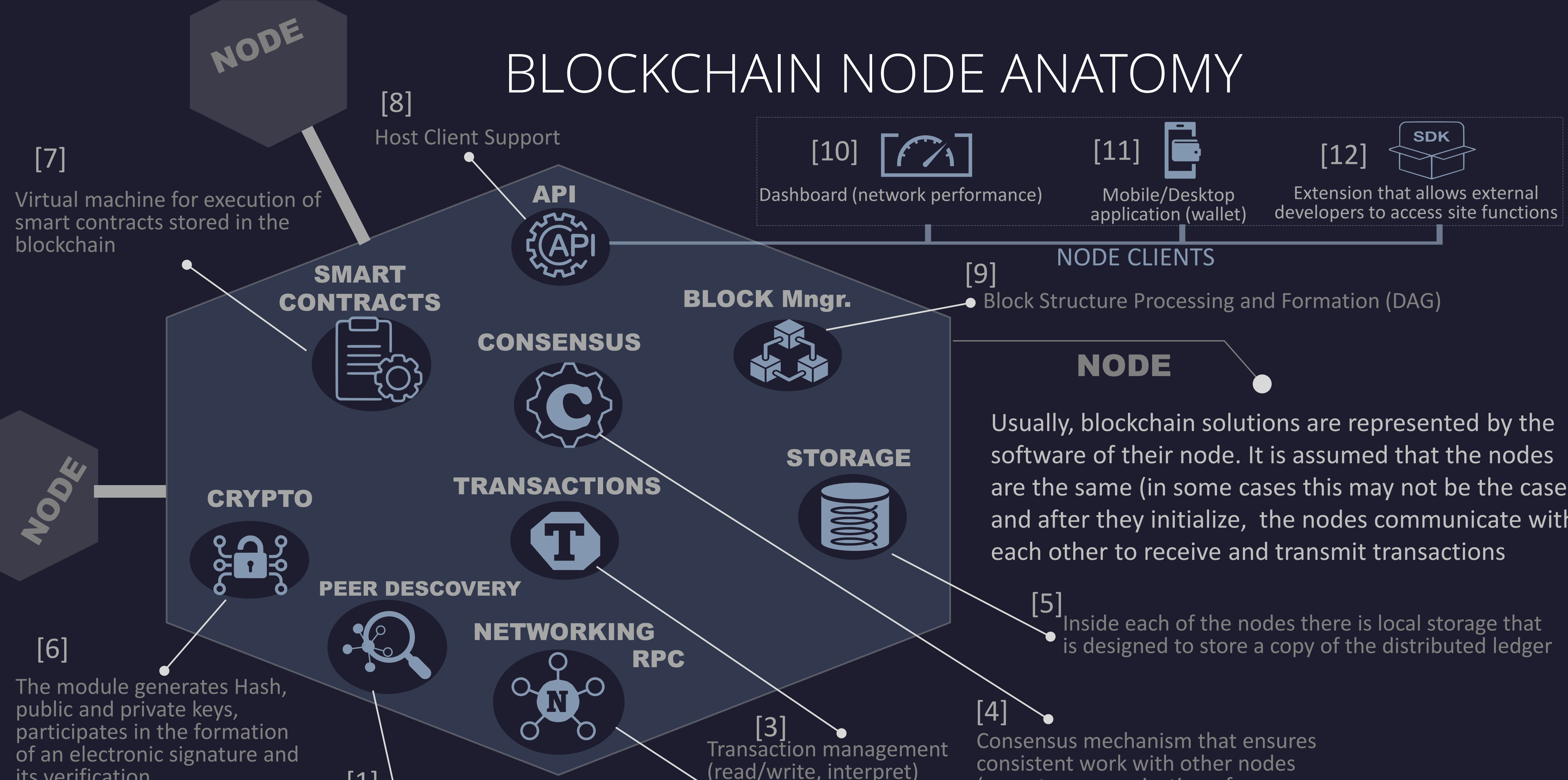
- Difficulty implementing complex logic
- Many calculations required

Transactions contain data on commissions to reward tx processing, which also makes it unprofitable for attackers to form meaningless transactions



In Bitcoin, the transaction fee is set freely depending on the load of the network. This commission is calculated as the difference between the amount of inputs and outputs

BLOCKCHAIN NODE ANATOMY



[7]
Virtual machine for execution of smart contracts stored in the blockchain

[8]
Host Client Support

[10] Dashboard (network performance)

[11] Mobile/Desktop application (wallet)

[12] Extension that allows external developers to access site functions

[9] Block Structure Processing and Formation (DAG)

NODE

Usually, blockchain solutions are represented by the software of their node. It is assumed that the nodes are the same (in some cases this may not be the case) and after they initialize, the nodes communicate with each other to receive and transmit transactions

[5] Inside each of the nodes there is local storage that is designed to store a copy of the distributed ledger

[6]
The module generates Hash, public and private keys, participates in the formation of an electronic signature and its verification

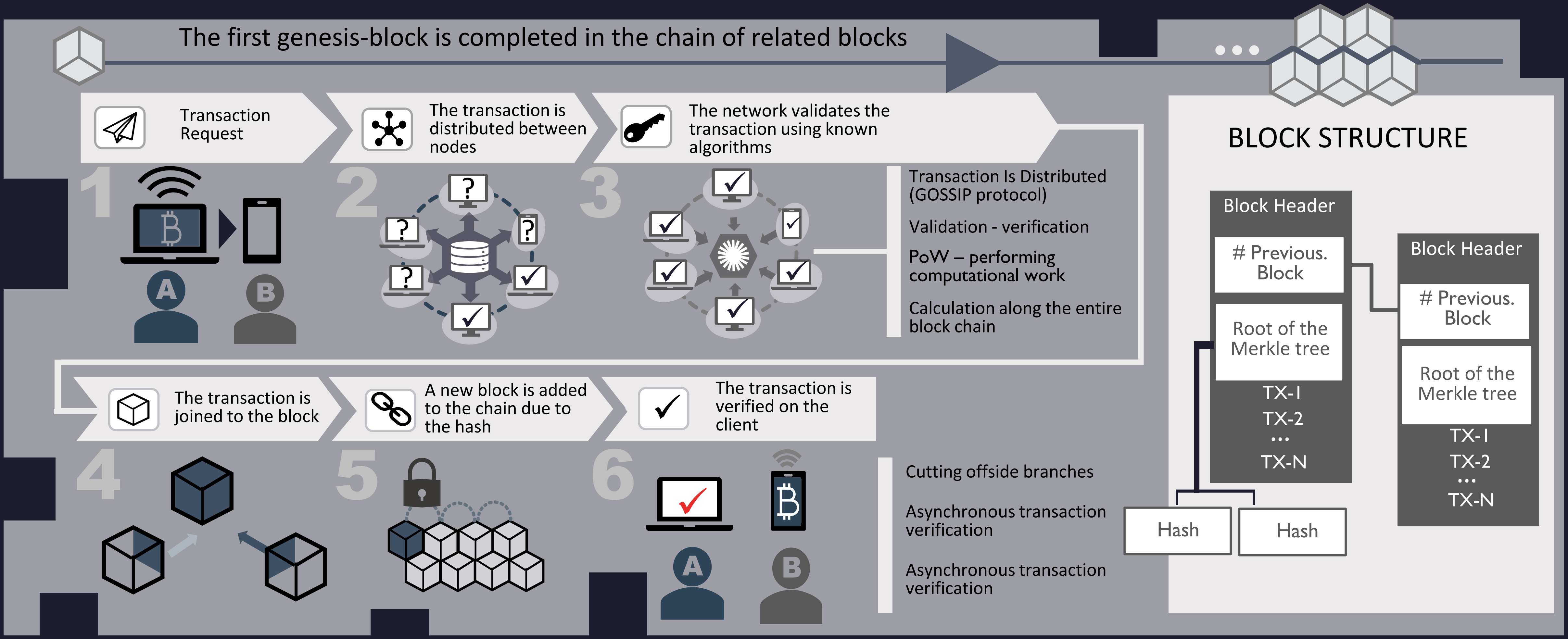
[4]
Consensus mechanism that ensures consistent work with other nodes (acceptance or rejection of transactions)

[3]
Transaction management (read/write, interpret)

[1]
The module provides the location of other nodes, as well as (in the case of closed networks) - authorization of nodes

[2]
The module manages sockets and receives messages over TCP/IP (message broker)

HOW BLOCKCHAIN WORKS



The first genesis-block is completed in the chain of related blocks

1 Transaction Request

2 The transaction is distributed between nodes

3 The network validates the transaction using known algorithms

Transaction Is Distributed (GOSSIP protocol)

Validation - verification

PoW – performing computational work

Calculation along the entire block chain

4 The transaction is joined to the block

5 A new block is added to the chain due to the hash

6 The transaction is verified on the client

Cutting offside branches

Asynchronous transaction verification

Asynchronous transaction verification

BLOCK STRUCTURE

Block Header

# Previous. Block

Root of the Merkle tree

TX-I

TX-2

...

TX-N

Block Header

# Previous. Block

Root of the Merkle tree

TX-I

TX-2

...

TX-N

Hash

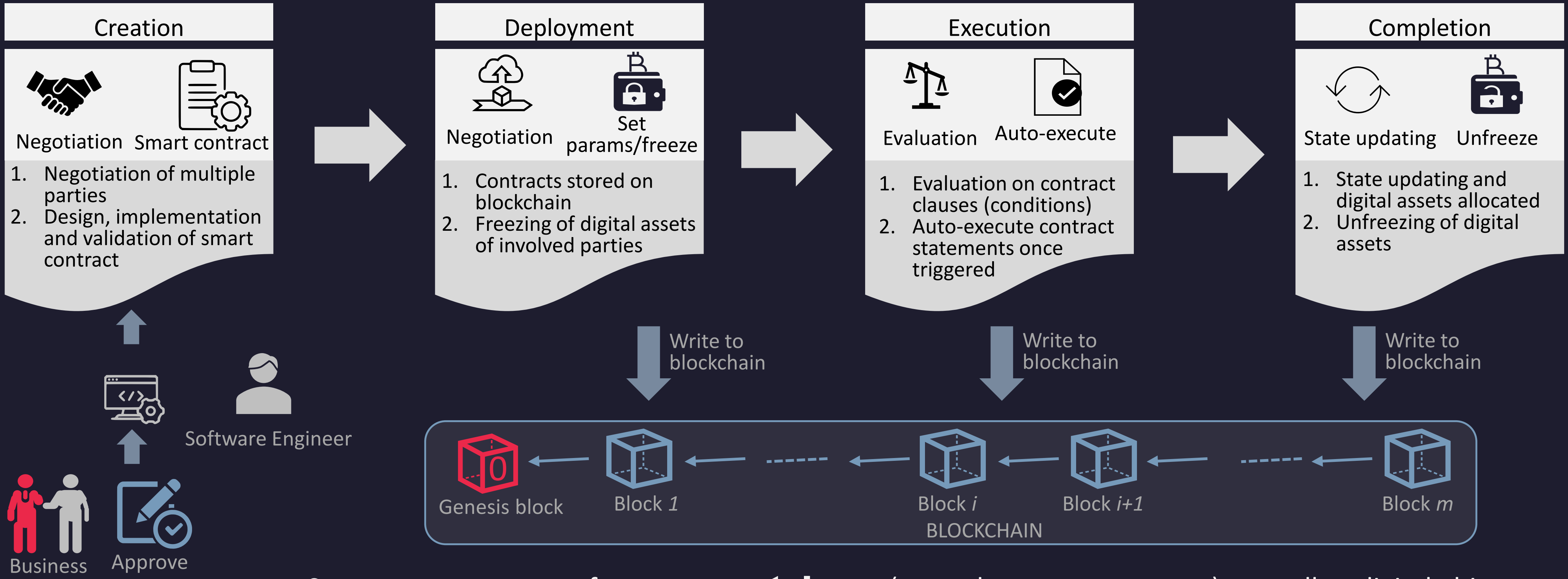
Hash

In a broad sense of the word, the work of the blockchain is associated with the passage of transactions (messages) between nodes. The initiative comes from the node client associated with the node, then this transaction is checked (validated) and embedded into a block and the block is then associated with other blocks. Finally, the resulting structure spreads between the nodes and data is synchronized.

SMART CONTRACTS

A **smart contract** is a mini-program (class) that represents a digital object. This program is stored in the blockchain network and when a certain event occurs, it is called (summoned) and executed, changing the state of specified registers. The program itself does not change.

Smart contracts can be written in any interpreted language, the greatest application was in the Ethereum network (most often implemented in the Solidity language). Smart contracts require decentralized platforms for their execution, some of their methods are called for free (ex. reading), while some require validation and payment for miners (Ethereum calls this payment – “gas”).



Creation

Negotiation   Smart contract

1. Negotiation of multiple parties
2. Design, implementation and validation of smart contract

Deployment

Negotiation   Set params/freeze

1. Contracts stored on blockchain
2. Freezing of digital assets of involved parties

Execution

Evaluation   Auto-execute

1. Evaluation on contract clauses (conditions)
2. Auto-execute contract statements once triggered

Completion

State updating   Unfreeze

1. State updating and digital assets allocated
2. Unfreezing of digital assets

Smart contracts most often represent **tokens** (secondary cryptocurrency), as well as digital objects.

ABOUT TOKENS

CRYPTOCURRENCY

Cryptocurrency is a type of digital currency based on records in a decentralized payment system implemented on the basis of a particular blockchain platform.

Cryptocurrency is represented by records of calculated sums for transfers between participants of a blockchain network.

- Cryptocurrency is **inextricably linked with the blockchain platform** on which it is implemented.
- Cryptocurrencies are **very volatile**, that is, their value can change unexpectedly and quickly, although in the long term most of them show steady growth. There are also special currencies ("stablecoins") that are designed to be volatility-resistant.
- Most blockchain currencies **do not allow transactions to be reversed** (with the exception of multi-signing, which gives some option to cancel a transaction). This means that a transfer to a non-existent address or an incorrect payment cannot be canceled and the "money is gone".
- The most important mechanism that determines cryptocurrencies is its **emission**, which generally limits the marginal money supply, and also regulates the speed of the flow of money, thus preventing devaluation (=inflation).
- The means to exchange a **certain crypto to fiat** and vice versa are important. There are special services or exchanges that can be used.

There is an equation in monetary theory:

$$M \cdot V = P \cdot Q$$

- it also holds true for cryptocurrencies, whereas you can determine the (M)onetary supply based on the (V)elocity of money, and the (P)rice level with the index of (Q)uantity of services produced

HOW TO START USING CRYPTOCURRENCY

As of 2021, the use of cryptocurrency for payments is still limited practically and legally. Purchasing crypto is legal in most regions (with exceptions, ex. China) from certified services and crypto exchanges. ! If you want to purchase crypto, you need to know that:

Investing in cryptocurrencies is very risky. Although the price of cryptocurrencies has been growing in the long term, these tools are very volatile and require a good understanding of the market, how to buy them on the decline and sell at the peak.

There are several services for the sale and purchase of common cryptocurrencies. [Coinbase, Wealthsimple, Binance, Gemini, Coinmama]are centralized companies offering the purchase of popular cryptocurrencies on credit cards with some restrictions:

- □ *Daily purchase restriction:* 500 – 5 000 USD;
- □ *Strict Know Your Customer* – KYC personal disclosure procedure of nationality and location (you may be denied service);
- □ *Limit on transfers from / to arbitrary "wallet addresses"* they create for you. These mostly aren't real wallets, but names for your account that is irrelevant to the blockchain network outside of the company you're using.

You can also use crypto exchanges that trade cryptocurrencies. Centralized crypto exchanges include [e-Toro, Kraken, FTX, Gate.io, Poloniex]. Decentralized crypto exchanges (DEX) trade directly between users: [Uniswap, Sushiswap, Raydium, Binance DEX]. When working with crypto exchanges, keep in mind that unlike "true" wallets, they hold your crypto on the account they store and that's vulnerable to hacking.

In many countries cryptocurrency is considered to be a security and is subject to appropriate regulation, including the payment of capital gains taxes, as well as the control of cross-border transactions.

To work with cryptocurrency directly, you will need a crypto wallet. A crypto wallet is a program, device, or other medium that stores information about public and private keys. Well-known wallets are Trust (ETH, ERC-20), Exodus (BTC, ETH, ERC-20 ,...), Electrum, Jaxx (BTC, ETH, Dash, Zcash), Mycelium (BTC, ETH, ERC-20), MyEtherWallet (ETH, ERC-20), MetaMask (ETH, ERC-20). Wallets can have a complex organization (HD wallets store a whole tree of keys) or can be simple – a pair of keys to be written on paper and stored away.

TOKENIZATION

Tokens (crypto tokens) are a representation of certain assets hosted on the blockchain network. Technically, any cryptocurrency is also a token, but usually the concept of a token is limited to secondary crypto-financial assets. Depending on their nature, tokens are interchangeable (the same from the user's point of view) or unique (not replaceable)

Tokenization is the process of transferring rights to an asset into a digital token using distributed ledger technology.

We can understand a token entity as a digital passport created for the asset, which is a unique digital code. In the future, it can be transmitted over the Internet, can receive any tangible and intangible asset or some action, for example, payment, time or legal status

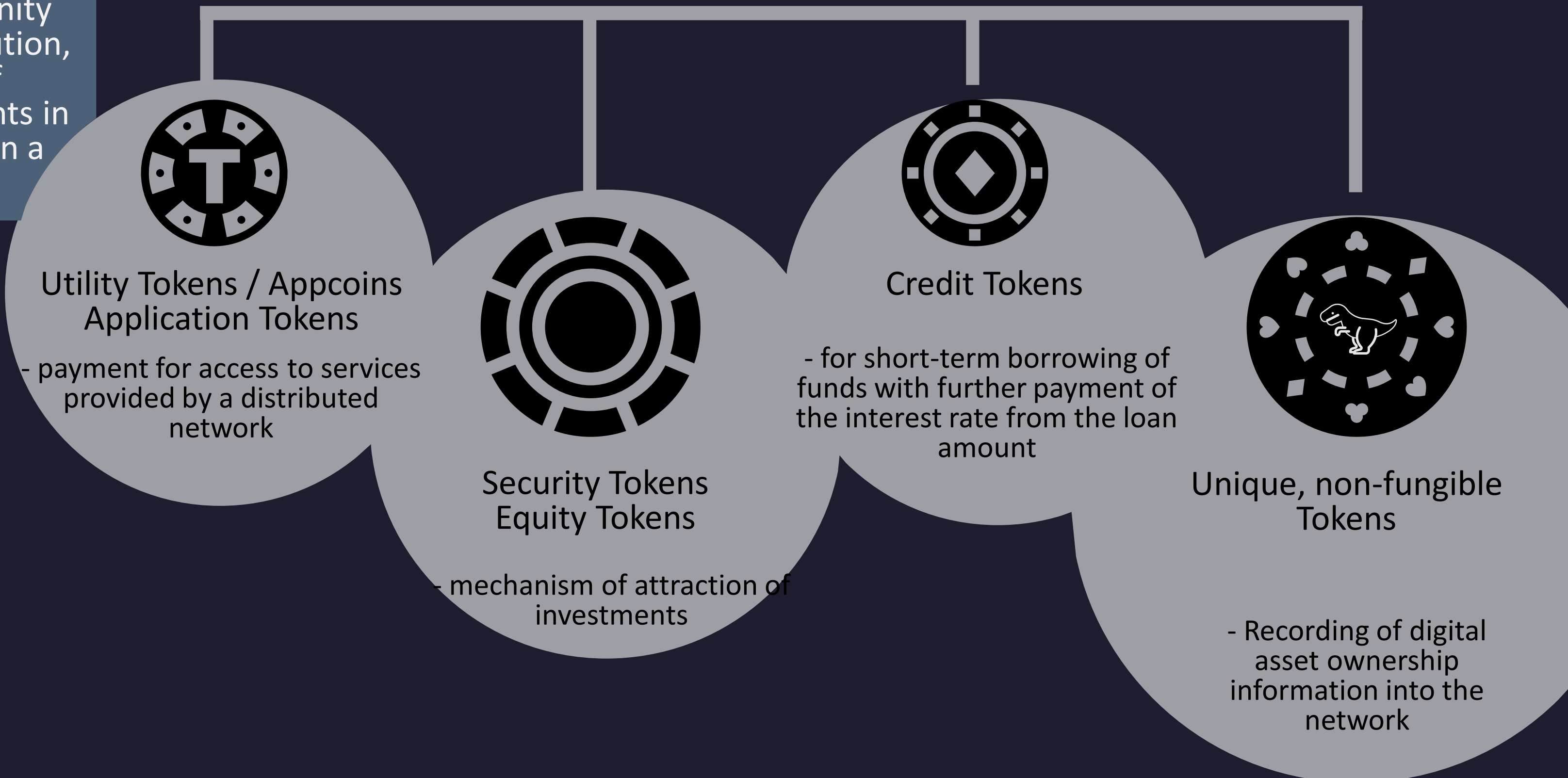
TOKENS:

An entry in a distributed registry designed to represent a unit of value or opportunity in the interaction (distribution, exchange, confirmation of rights) between participants in an information exchange in a certain computer system.

CRYPTOCURRENCY (COINS)

A digital asset is a means of exchanging goods or services for a single equivalent of value, information about which is stored electronically in a specialized immutable database - a distributed registry

There are different types of tokens that have different properties



Utility Tokens / Appcoins Application Tokens

- payment for access to services provided by a distributed network

Security Tokens Equity Tokens

- mechanism of attraction of investments

Credit Tokens

- for short-term borrowing of funds with further payment of the interest rate from the loan amount

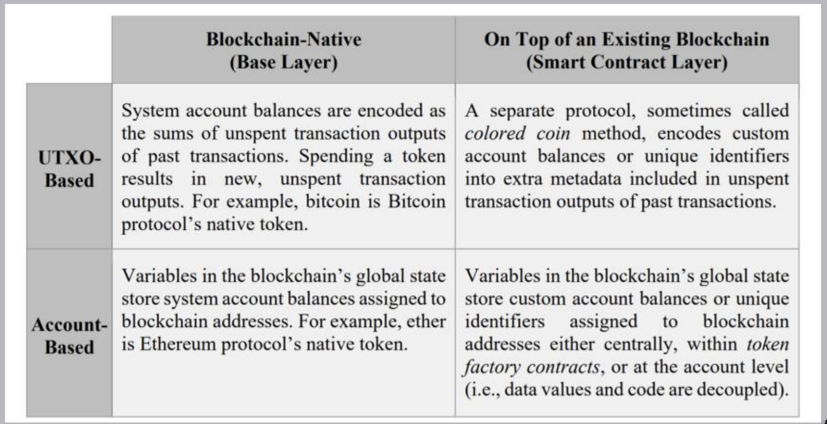
Unique, non-fungible Tokens

- Recording of digital asset ownership information into the network

TOKEN STANDARDS

NISTIR 8301

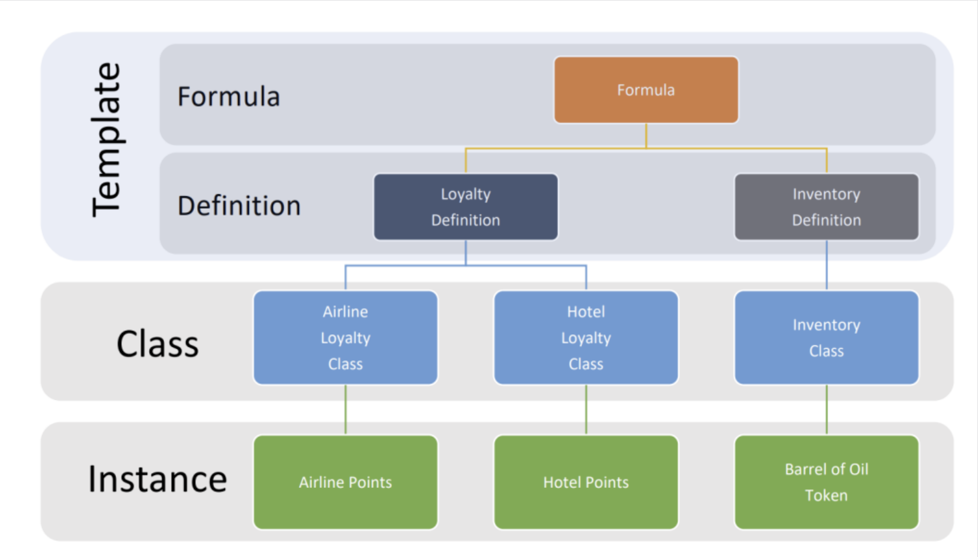
Blockchain Networks: Token Design and Management Overview



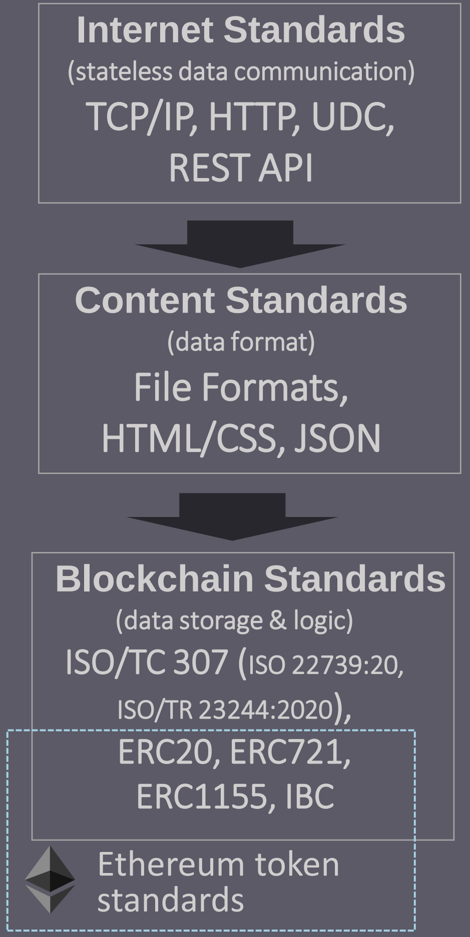
	Blockchain-Native (Base Layer)	On Top of an Existing Blockchain (Smart Contract Layer)
UTXO-Based	System account balances are encoded as the sums of unspent transaction outputs of past transactions. Spending a token results in new, unspent transaction outputs. For example, bitcoin is Bitcoin protocol's native token.	A separate protocol, sometimes called *colored coin* method, encodes custom account balances or unique identifiers into extra metadata included in unspent transaction outputs of past transactions.
Account-Based	Variables in the blockchain's global state store system account balances assigned to blockchain addresses. For example, ether is Ethereum protocol's native token.	Variables in the blockchain's global state store custom account balances or unique identifiers assigned to blockchain addresses either centrally, within *token factory contracts*, or at the account level (i.e., data values and code are decoupled).

HYPERLEDGER

TOKEN FRAMEWORK



EVOLUTION OF STANDARDS



Internet Standards
(stateless data communication)
TCP/IP, HTTP, UDC, REST API

Content Standards
(data format)
File Formats, HTML/CSS, JSON

Blockchain Standards
(data storage & logic)
ISO/TC 307 (ISO 22739:20, ISO/TR 23244:2020),
ERC20, ERC721, ERC1155, IBC

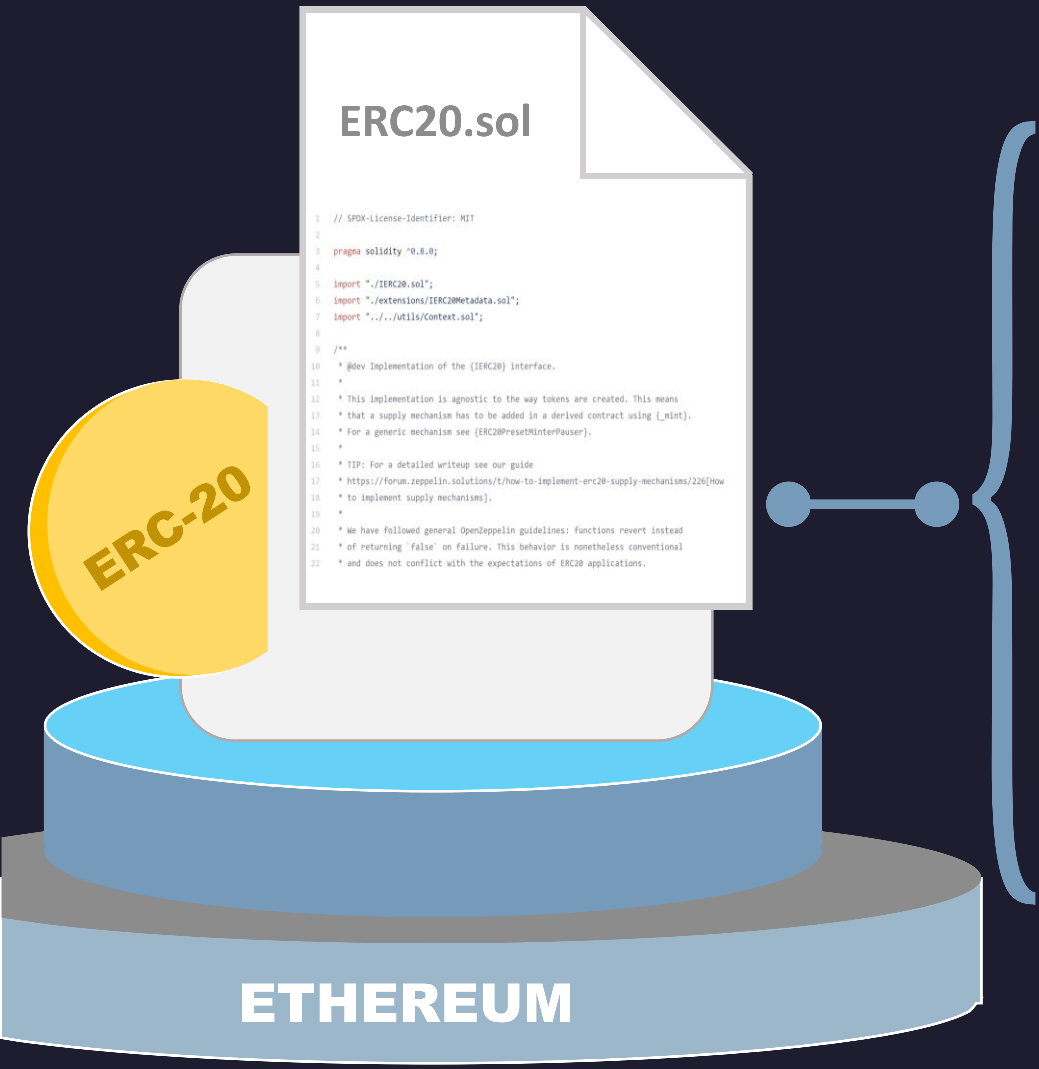
Ethereum token standards

Issuance of tokens (crypto-tokens) means the creation of computer programs written according to certain standards that determine methods of issuing and distributing tokens.

Token systems are associated with the network on which they are issued (the distributed ledger in which they are stored), as well as certain standards, on the basis of which the relevant rules are checked - their release and transference.

In the Ethereum network, tokens are represented by a smart program. So the issuance, transfer of tokens and other rules require the execution of a smart contract, which cannot be changed after its uploaded.

THE CLASSIC ETHEREUM BASED ERC-20 TOKEN



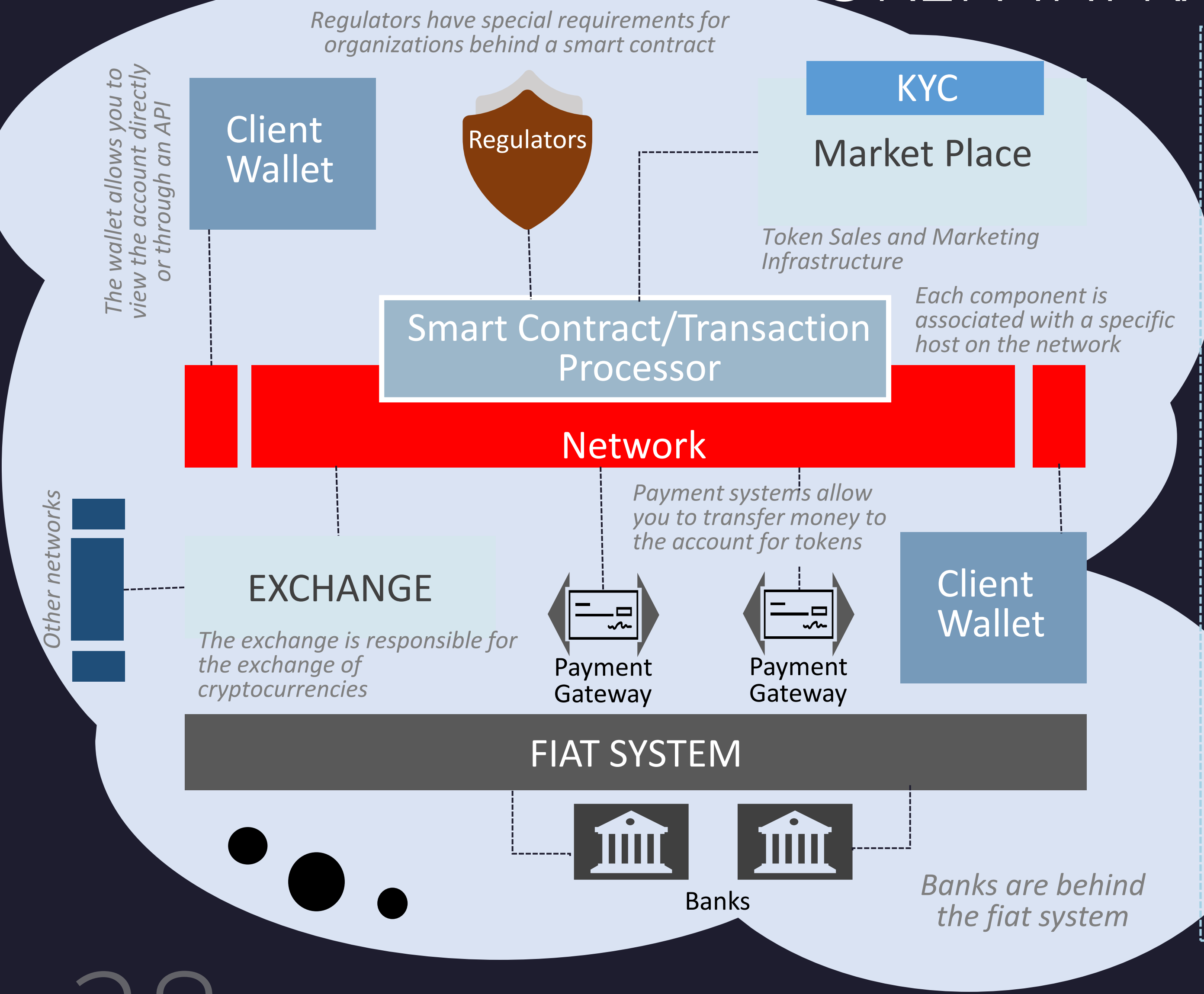
```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "./IERC20.sol";
import "./extensions/IERC20Metadata.sol";
import "../../utils/Context.sol";

/**
 * @dev Implementation of the {IERC20} interface.
 *
 * This implementation is agnostic to the way tokens are created. This means
 * that a supply mechanism has to be added in a derived contract using {_mint}.
 * For a generic mechanism see {ERC20PresetMinterPauser}.
 *
 * TIP: For a detailed writeup see our guide
 * https://forum.zeppelin.solutions/t/how-to-implement-erc20-supply-mechanisms/226[How
 * to implement supply mechanisms].
 *
 * We have followed general OpenZeppelin guidelines: functions revert instead
 * of returning `false` on failure. This behavior is nonetheless conventional
 * and does not conflict with the expectations of ERC20 applications.
```

#	FUNCTION	PARAMETERS	RETURN	DESCRIPTION
1	Name	-	String	Token Title
2	Symbol	-	String	Token Symbol (like ₿)
3	Decimals	-	Unit8	Fractional part of the token
4	totalSupply	-	Unit256	Total number of tokens
5	balanceOf	address_owner	Unit256	Balance of tokens at the owner of the smart contract
6	Transfer	address_to, _value	Bool	Moving tokens from the owner's address to another user's address
7	transferFrom	address_from, address_to, _value	Bool	Transfer of tokens from one address to another
8	Approve	address_spender, _value	Bool	Delegation of the ability to manipulate tokens (for example, for an exchange)
9	allowans	address_owner, address_spender	Unit256	Number of tokens to manipulate the delegate (exchange)

Tokens issued according to the ERC-20 standard on Ethereum contains a set of standard-specific methods (ex. address whitelists and blacklists; time limits; etc.)

Issuing tokens, transferring them, calling to methods is done from a wallet. Some of these actions require gas (ETH), while others are free.

TOKEN INFRASTRUCTURE



Regulators have special requirements for organizations behind a smart contract

The wallet allows you to view the account directly or through an API

Client Wallet

Regulators

KYC

Market Place

Token Sales and Marketing Infrastructure

Smart Contract/Transaction Processor

Each component is associated with a specific host on the network

Network

Other networks

Payment systems allow you to transfer money to the account for tokens

EXCHANGE

The exchange is responsible for the exchange of cryptocurrencies

Payment Gateway

Payment Gateway

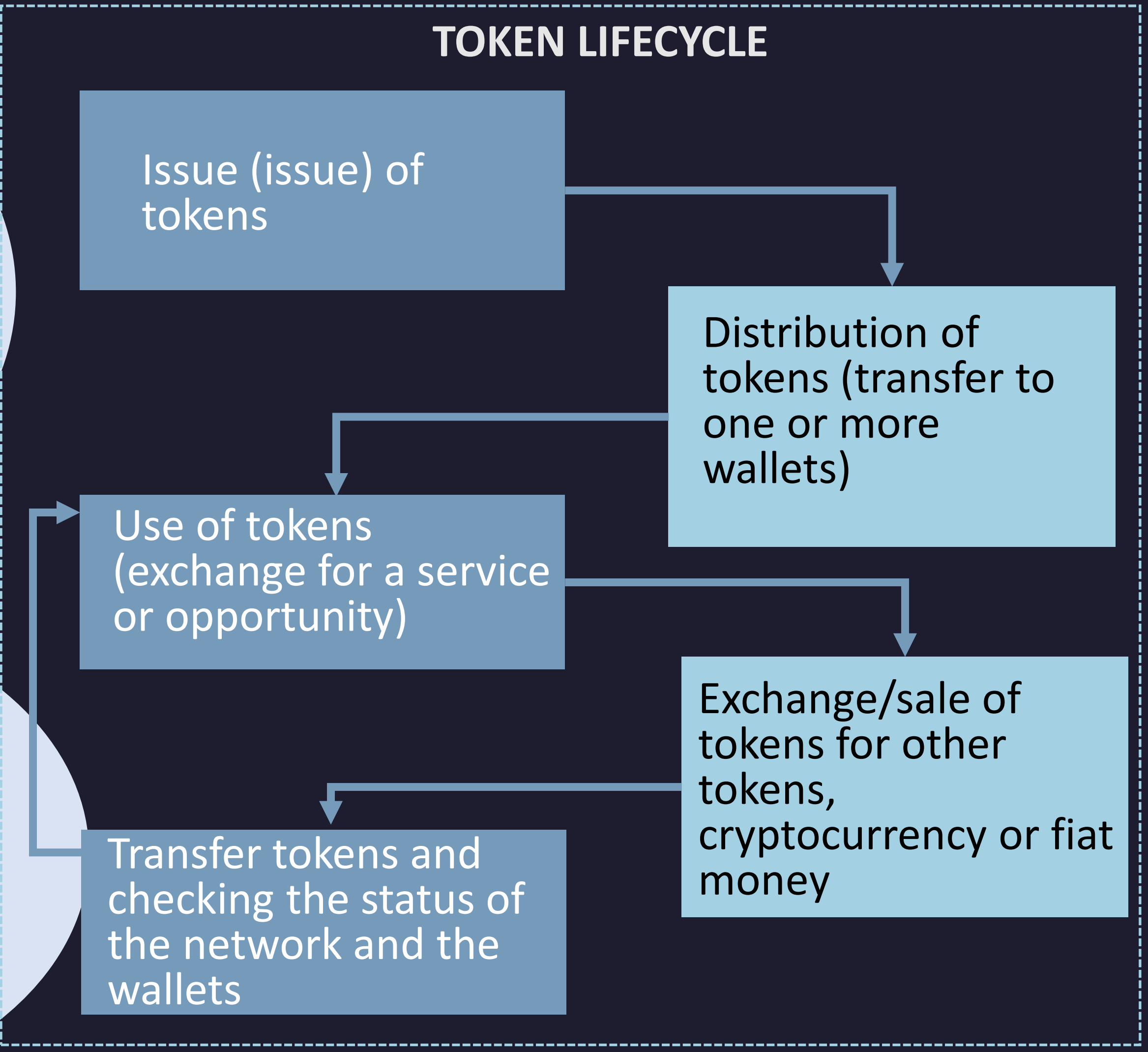
Client Wallet

FIAT SYSTEM

Banks

Banks are behind the fiat system

TOKEN LIFECYCLE



Issue (issue) of tokens

Distribution of tokens (transfer to one or more wallets)

Use of tokens (exchange for a service or opportunity)

Exchange/sale of tokens for other tokens, cryptocurrency or fiat money

Transfer tokens and checking the status of the network and the wallets

The token management infrastructure encompasses several proprietary and external systems that allow you to create, promote, and sell tokens. This includes a *software module / smart contract*, which controls the issuance and distribution of tokens in the network (rules); a *wallet* – the means of controlling & monitoring; and a *marketplace / exchange* – the means of selling and exchanging for other tokens and currencies.

TOKEN ETHEREUM STANDARDS

FEATURE	ERC-20	ERC-1400	ERC- 1155	ERC-777	ERC-721 [NFT]
Proposal	Standard ICO token – basic token transfer functionality	Full or partial ownership of an object, additional methods for the "possession" of securities	Multi-token, the main idea is to save gas by supporting callbacks as a replacement for events	Interchangeable tokens, extends ERC-20 due to more complex interactions - callbacks (hooks), voluntary rejection of sent tokens, redirection of received tokens to other addresses	Non-Fungible Tokens (NFT); allows metadata storage, contains a reference to digital objects outside the network (for example, in IPFS), contains access control
Immutable Cap Table	✓	✓	✓	✓	✓
Open-Source Codebase	✓	✓	✓	✓	✓
Controller Access (Token Recovery Process)	×	✓	×	✓	✓
Compliance	×	✓	×	×	×
Issue / Redemption	×	✓	×	×	✓
Permission Management W/ Multiple Agents	×	✓	✓	✓	✓
Event Management	×	✓	✓	✓	×
Partially / Fully Non-Fungible	×	✓	×	×	✓
	link	link	link	link	link

BLOCKCHAIN: ADVANCED

CRYPTOGRAPHIC HASH FUNCTION

A cryptographic hash function (CHF) is a one-way (feasibly unreversible) mathematical function that maps data of *arbitrary size* (“message”) to a bit array of a *fixed size* (“hash value”).

HISTORY

Galileo Galilei observed the rings of Saturn, which he mistook for "ears". Unsure, but wanting to assert himself as the pioneer of this discovery, he posted a cryptic message by rearranging letters:

smaismrmilmepoetaleumibunenugttauiras.

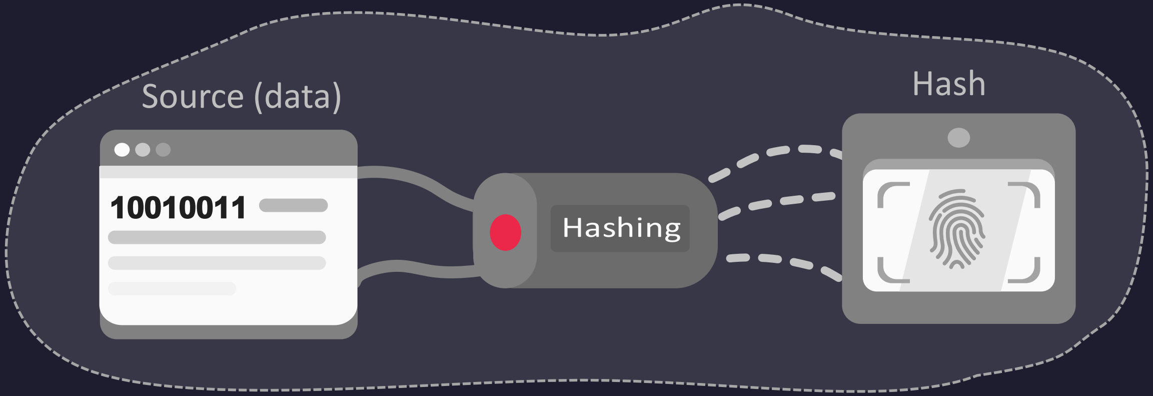
In 1610, he revealed the original phrase:

Altissimum planetam tergeminum obseruaui,

which in Latin means "the highest planet triple observed". Thus, at the time of publication of the first message, the original meaning was not disclosed, but it was possible for him to allude to it at a later point.

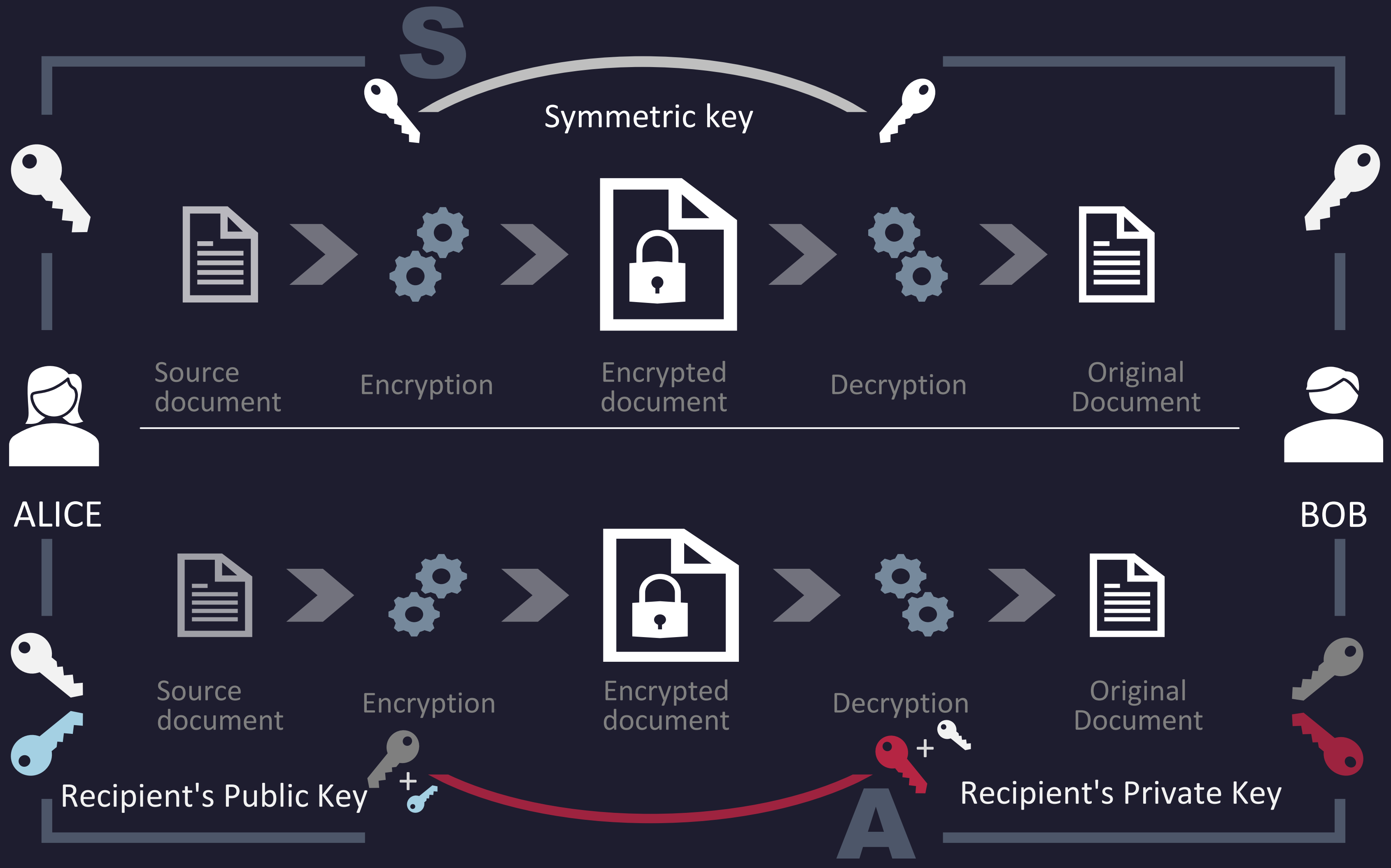
There are several algorithms that implement hashing strings and byte arrays/files:

[CRC32] [MD5] [Keccak (SHA-3)] [GOST R 34.11-94] [SHA-256] [SWIFFT] [HAVAL] [Streebog]



Example Hello $\xrightarrow{\text{SHA-256}}$ 185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969

SYMMETRIC AND ASYMMETRIC CRYPTOGRAPHY



Unlike symmetric encryption, which uses the same secret key to encrypt and decrypt sensitive information, asymmetric encryption, also known as public-key cryptography or public-key encryption, uses mathematically linked public- and private-key pairs to encrypt and decrypt senders' and recipients' sensitive data.

Symmetric Cryptography

- AES (Advanced Encryption Standard), an American encryption standard
- GOST 28147-89 is a Soviet and Russian encryption standard, also a CIS standard
- DES (Data Encryption Standard), a data encryption standard in the United States
- 3DES (Triple-DES, triple DES)
- RC2 (Rivest Cipher or Ron's Cipher)

Asymmetric Cryptography

- RSA (Rivest-Shamir-Adleman)
- DSA (Digital Signature Algorithm)
- El Gamal (El Gamal Cipher System)
- Diffie-Hellman (Diffie-Hellman Key Exchange)
- **ECDSA (Elliptic Curve Digital Signature Algorithm**) is a public-key algorithm for creating a digital signature.
- GOST R 34.10-2012

Distributed systems such as Blockchain use mainly asymmetric cryptography, preferring mostly ECDSA elliptic curve cryptography.

DIGITAL SIGNATURE

Digital signatures are a cryptographic tool for signing and verifying the authenticity of digital messages or electronic documents. They provide:

Authentication - proof that a certain known sender (the owner of the secret key) created and signed the message.

Integrity - proof that the message has not been changed after signing.

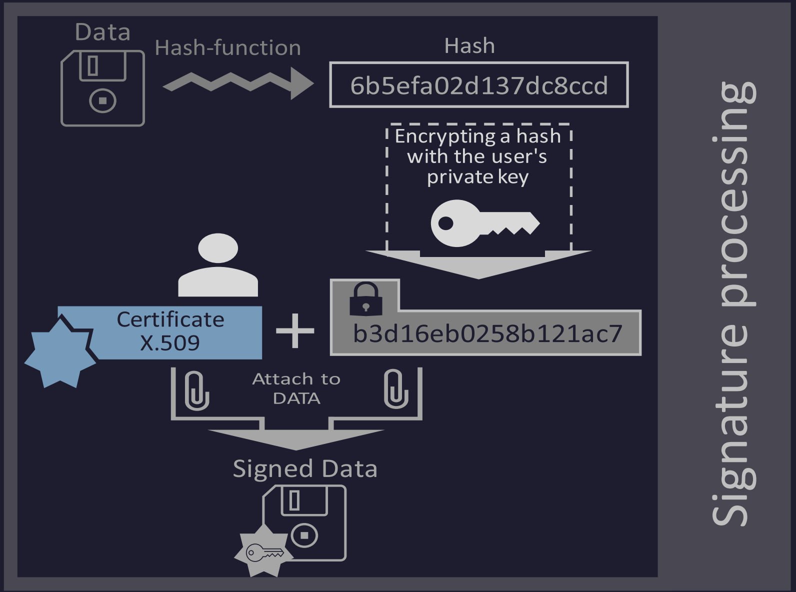
Non-repudiation - the signer cannot refuse to sign the document after the signature has been created

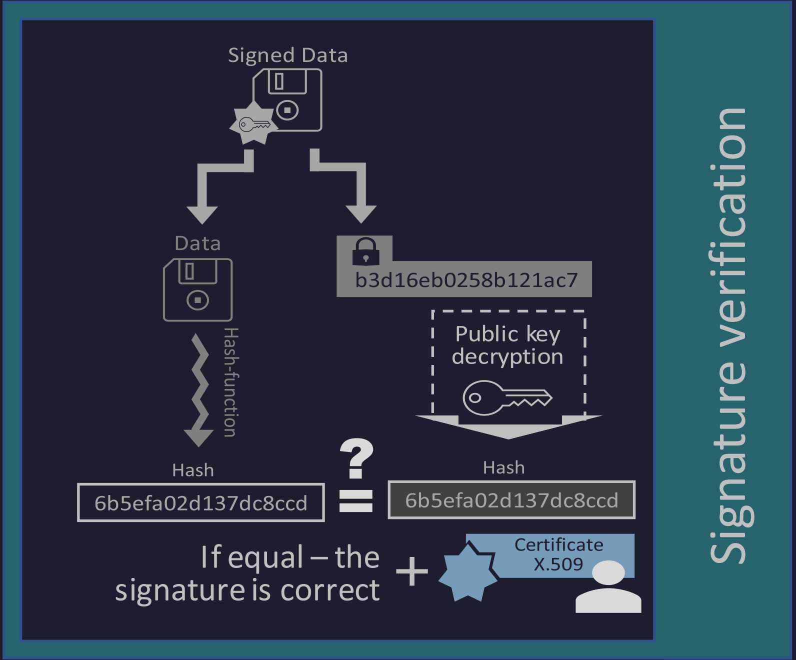
A signature can be made using *symmetric* cryptography or *asymmetric* cryptography (the more popular choice in blockchain systems)

- **Sign message:** input message is hashed + private “signature” key (calculated using an algorithm)
- **Decode message:** usually need the private “signature” key
- **Verify signature:** usually need public “verification” key (result: “valid” or “invalid”)

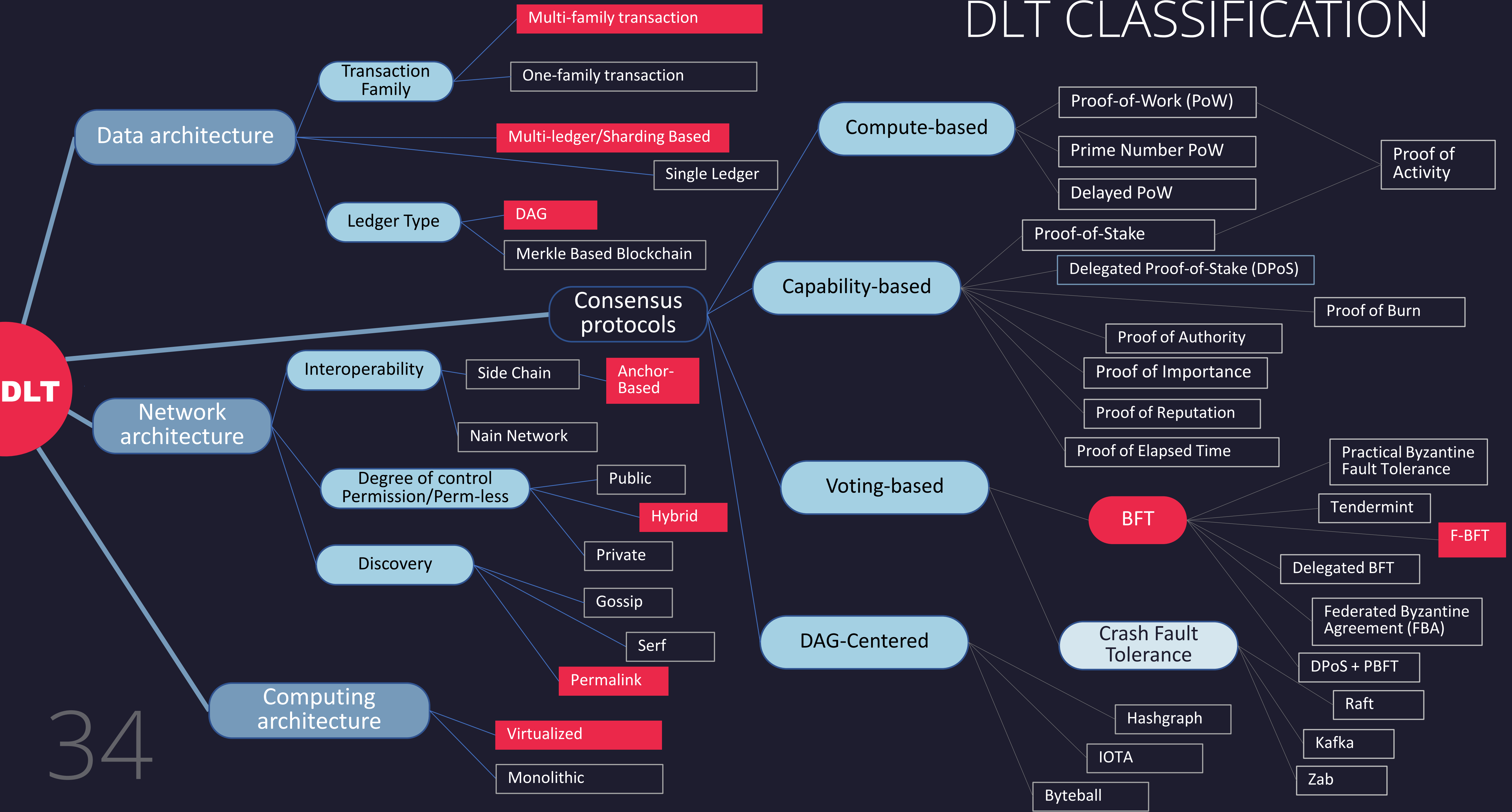
Although blockchains anonymize signers of transactions, ownership of public “verification” keys can be tied to particular entities. This problem is solved using a certificate (ex. X.509 standard)

Examples of well-known digital signature schemes are: DSA, ECDSA, EdDSA, RSA signatures, ElGamal signatures and Schnorr signatures.





DLT CLASSIFICATION



- DLT
  - Data architecture
    - Transaction Family
      - Multi-family transaction
      - One-family transaction
    - Multi-ledger/Sharding Based
    - Single Ledger
    - Ledger Type
      - DAG
      - Merkle Based Blockchain
  - Consensus protocols
    - Compute-based
      - Proof-of-Work (PoW)
        - Proof of Activity
      - Prime Number PoW
      - Delayed PoW
    - Capability-based
      - Proof-of-Stake
      - Delegated Proof-of-Stake (DPoS)
      - Proof of Burn
      - Proof of Authority
      - Proof of Importance
      - Proof of Reputation
      - Proof of Elapsed Time
    - Voting-based
      - BFT
        - Practical Byzantine Fault Tolerance
        - Tendermint
        - F-BFT
        - Delegated BFT
        - Federated Byzantine Agreement (FBA)
        - DPoS + PBFT
      - Crash Fault Tolerance
        - Raft
        - Kafka
        - Zab
    - DAG-Centered
      - Hashgraph
      - IOTA
      - Byteball
  - Network architecture
    - Interoperability
      - Side Chain
        - Anchor-Based
      - Nain Network
    - Degree of control Permission/Perm-less
      - Public
      - Hybrid
      - Private
    - Discovery
      - Gossip
      - Serf
      - Permalink
  - Computing architecture
    - Virtualized
    - Monolithic

COMPARISON OF MAIN CONSENSUS FAMILIES

	PROOF-OF-WORK	PROOF-OF-STAKE	BFT-BASED PROTOCOLS	**Federative-BFT (F-BFT)**
NODE IDENTIFICATION	Fully open, **public network**	Fully open, **public network**	Node ID's managed for **private networks**	**Hybrid network** based on a flexible KYC mechanism
THROUGHPUT	! Limited: due to risk of forks	! Poor: better than BFT but still slow	**Great:** (ten thousand TPS)	**Great:** as for all BFT
LATENCY	! High: each block to be approved by many	**Low**	**Very low:** defined by network delays	**Very low:** federative structure allows for better organization of communication within (functionally) close groups
POWER CONSUMPTION	! High: useless computing work	Low	**Great:** does not require high computing power	**Great:** same as BFT
SCALABILITY	Many participants	! Encounters the "rich get richer" problem at scale and demotivates participants	! Limited: small # of nodes; closeness to centralized tech	**High:** horizontal & vertical scalability due to federal structure and DAG ledger
CORRECTION OF SELECTION PROCESSES	No	No	**Yes**	**Yes**

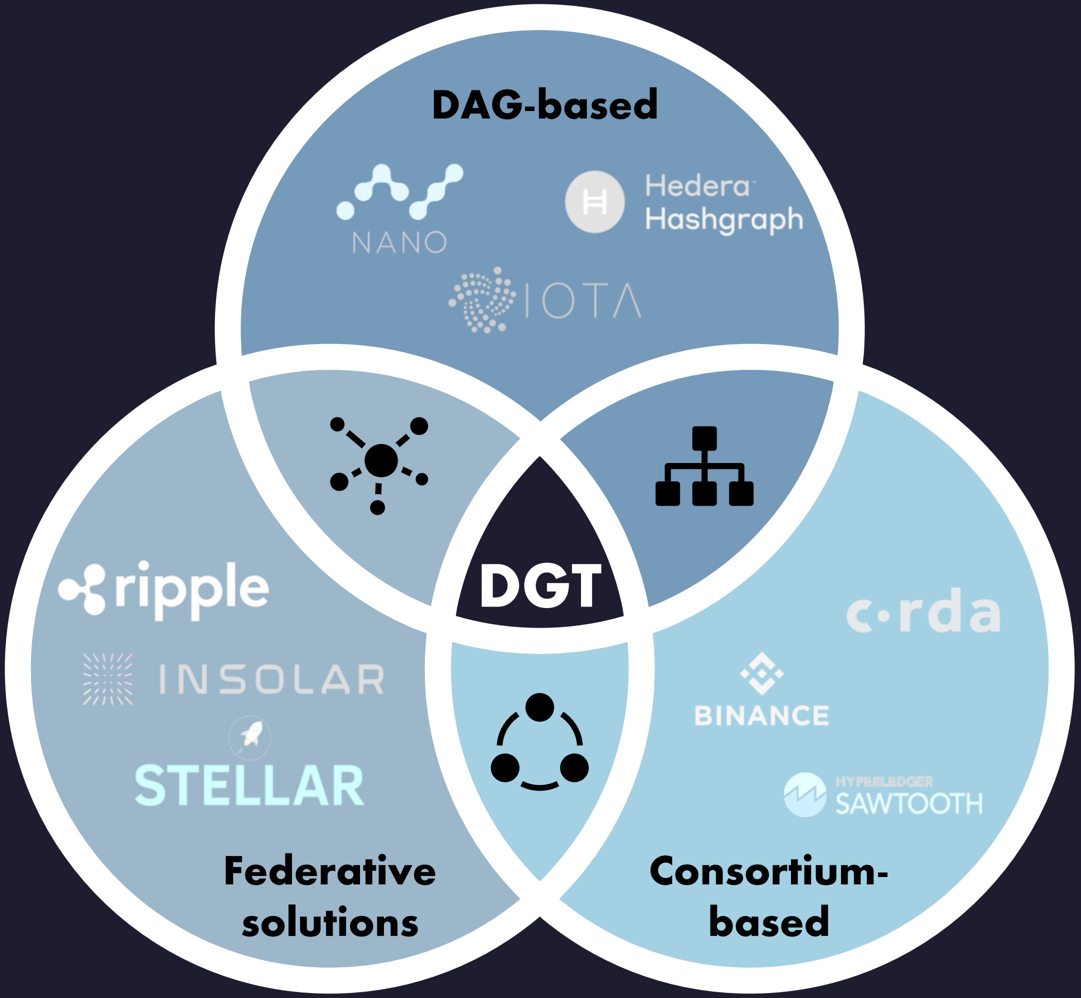
BLOCKCHAIN FOR BUSINESS

There are three important blockchain architecture aspects that can make a particular network suitable, secure, effective enough for enterprises. These are: (1) DAG registries; (2) federative voting; and (3) consortium-based consensuses.

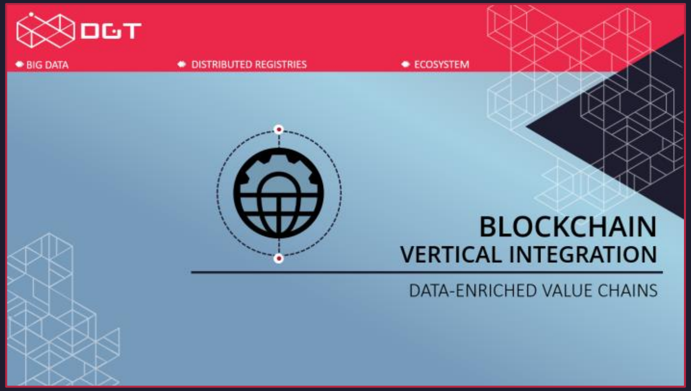
- **The storage system is based on DAG** (Directed Acyclic Graph). These systems allow for limitless scalability due to DAG trees' unique mathematical properties and ability to simultaneously "branch out" into multiple directions.

- **The federative approach** to voting is actively used by such solutions as Ripple, Stellar. This allows for flexible, but secure, cost-effective and fast transaction throughputs.

- **Consortium-based consensus** systems allow for flexibility, without sacrificing speed and interoperability. For example, Hyperledger Fabric targets private peer-to-peer networks and requires the formation of special sidechains, ICON's solution uses a special Loopchain Fault Tolerance mechanism to interact with other networks, while DGT implements a dynamic topology on top of DAG, allowing for highly asynchronous network operation.



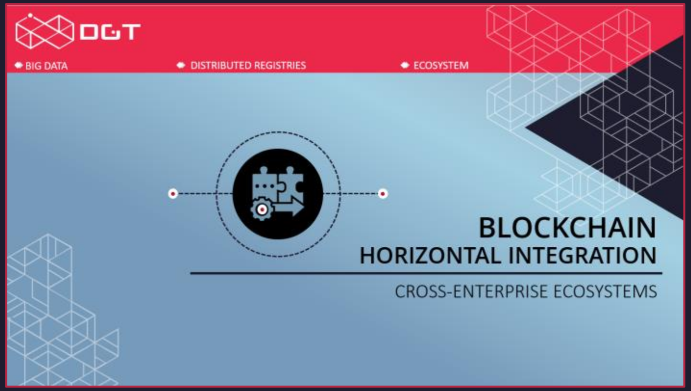
LEARN MORE


DGT

BIG DATA · DISTRIBUTED REGISTRIES · ECOSYSTEM

BLOCKCHAIN
VERTICAL INTEGRATION

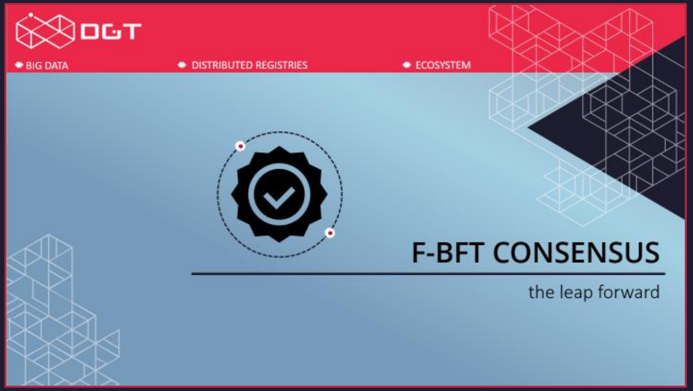
DATA-ENRICHED VALUE CHAINS


DGT

BIG DATA · DISTRIBUTED REGISTRIES · ECOSYSTEM

BLOCKCHAIN
HORIZONTAL INTEGRATION

CROSS-ENTERPRISE ECOSYSTEMS


DGT

BIG DATA · DISTRIBUTED REGISTRIES · ECOSYSTEM

F-BFT CONSENSUS

the leap forward

DGT

COMMODITIES · DECENTRALIZED REGISTRIES · MARKETPLACE

TOKENIZING
COMMODITIES

DGT COMMODITY MARKETPLACE

DGT

BIG DATA · DISTRIBUTED REGISTRIES · ECOSYSTEM

NON-FUNGIBLE
TOKENS

OVERVIEW

DGT

BIG DATA · DISTRIBUTED REGISTRIES · ECOSYSTEM

TECHNICAL DEEP DIVE

platform, technology, implementation

CONNECT TO DGT