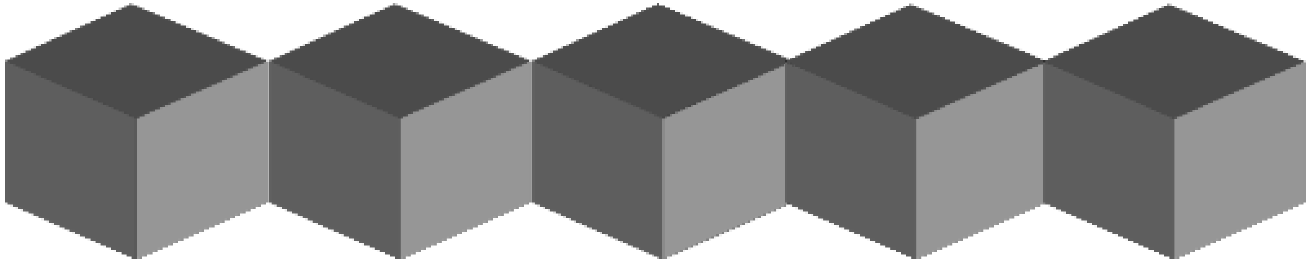
Course Outline

1. Cryptocurrency and Block chain
2. Delving into BlockChain
3. Bitcoin and Block chain
4. Bitcoin Mining
5. Ethereum
6. Setting up private Blockchain Environment using Ethereum Platform
7. Hyperledger
8. Setting up Development Program using Hyperledger composer
9. Create or Deploy our private Blockchain on Multi chain
10. Prospect of Blockchain

Delving into BlockChain



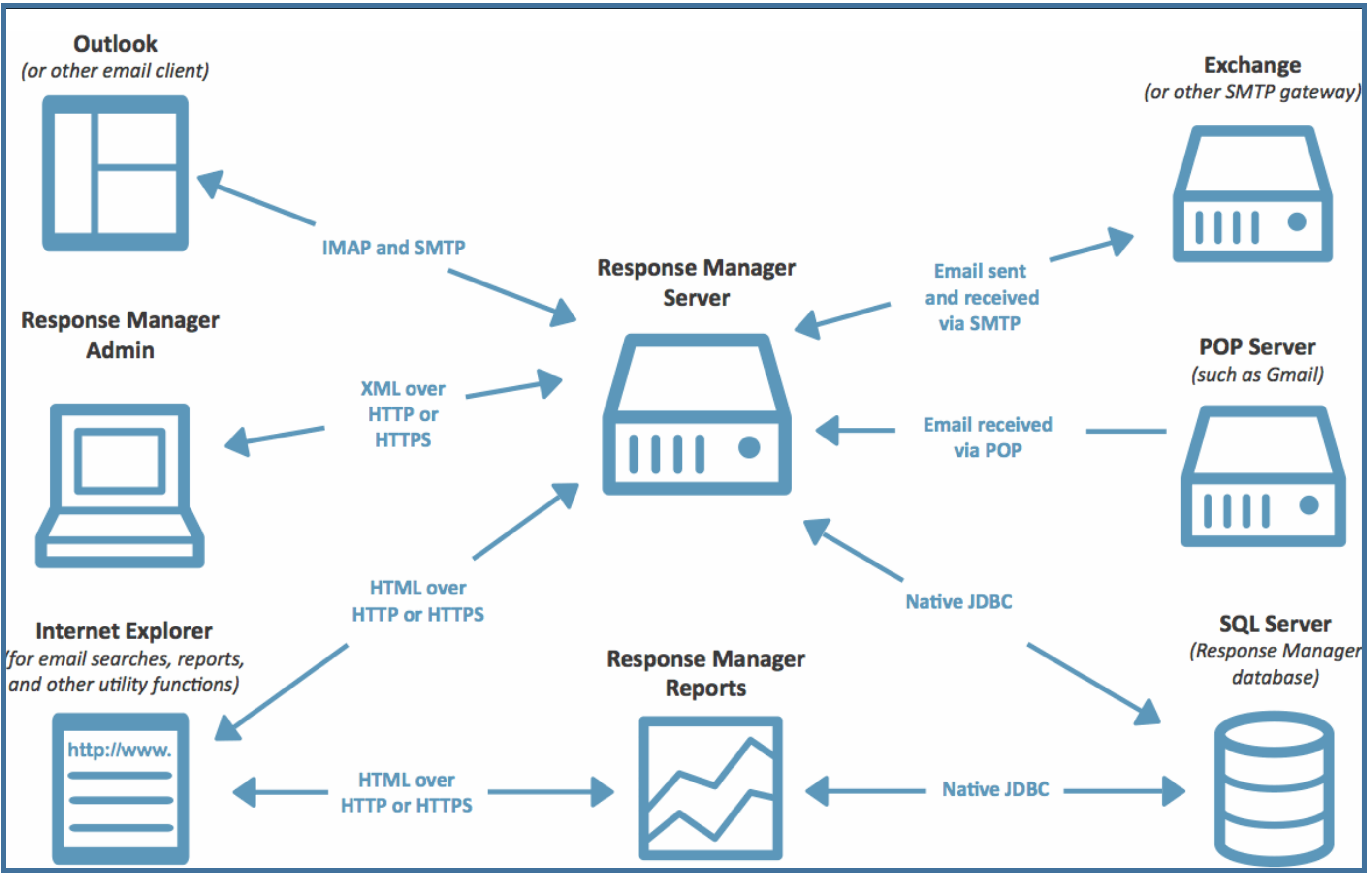
Agenda

At the end of this session you will be able to:

- Understand Blockchain
- Define Blockchain and Distributed Ledger Technology
- Identify Blockchain Structure
- Define Blockchain Ecosystem
- Describe Types of Blockchain

Database Overview

What if we don't have databases ?



What if we don’t have databases ?

> Did You Know, there exist an immutable & non tamper able database.

Blockchain

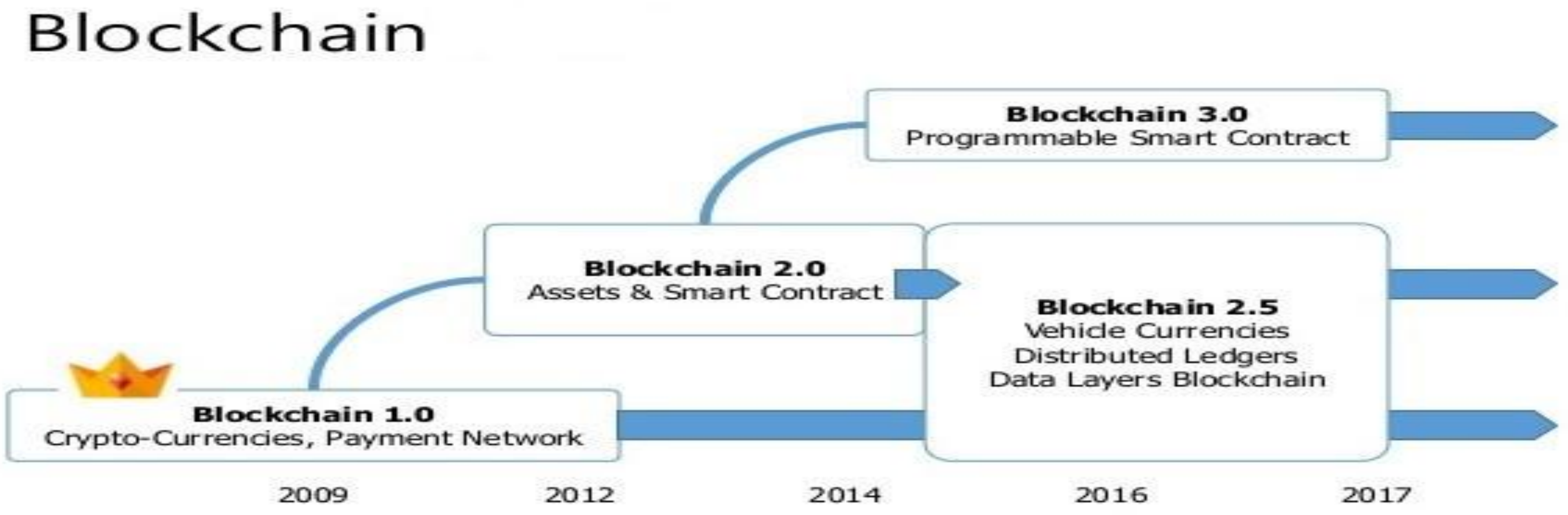
How Blockchain Started?

> In 2008, Satoshi Nakamoto (An anonymous person or persons), first gestated and implemented the first blockchain database as a infrastructure for the bitcoin, the first cryptocurrency ever created and the most successful of all time.

How Blockchain Started?

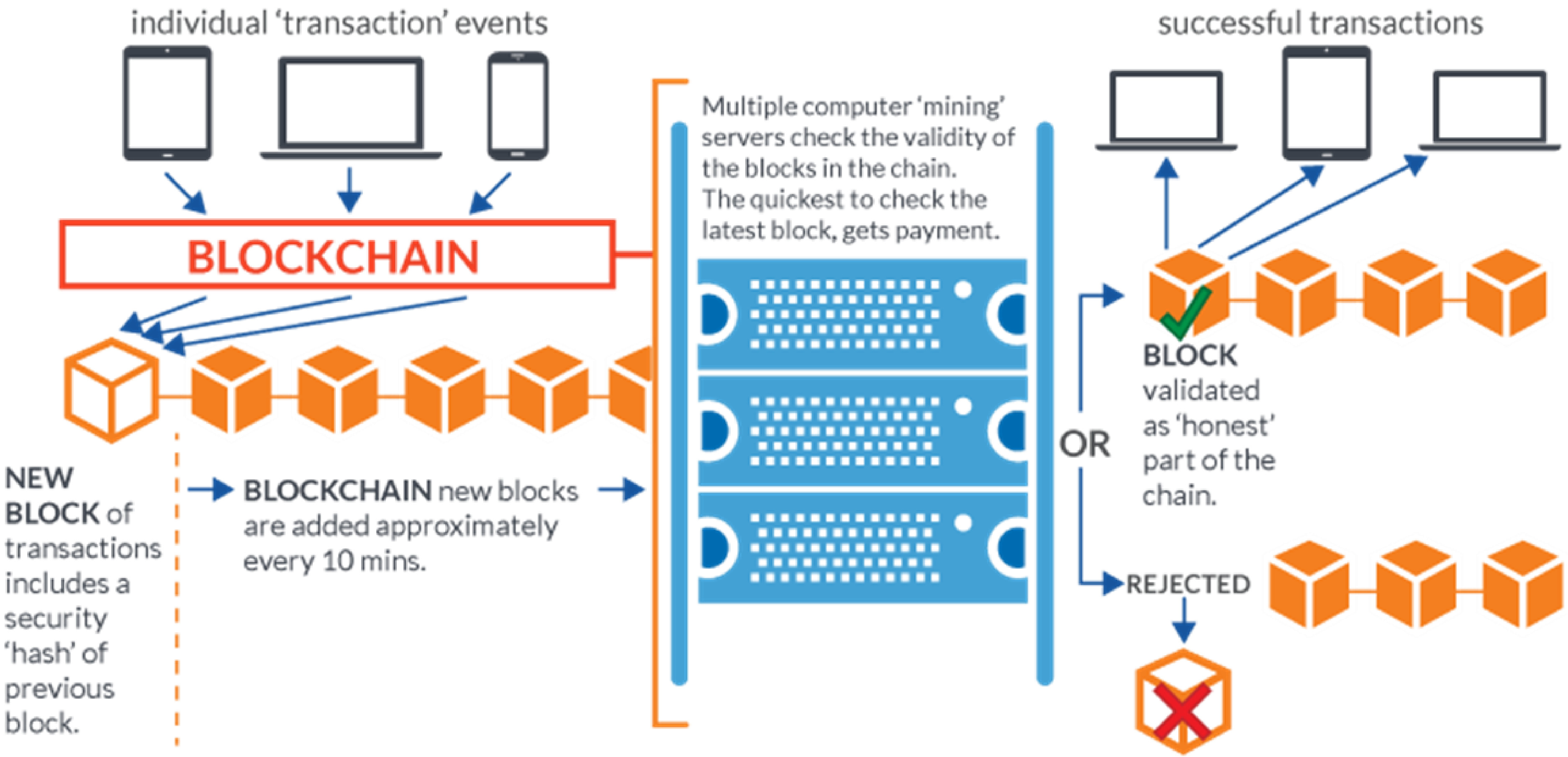
- Satoshi Nakamoto used ‘block’ and ‘chain’ separately in his paper in October 2008.
- Later with time, it became a single word ‘Blockchain’

How Blockchain has evolved ?



Note: Smart Contracts are unable to access external data or events based on time or market conditions. Calling code or data outside of a Smart Contract or blockchain breaks the general trust barrier and authenticity of transactions

Blockchain – Flow Diagram


individual 'transaction' events
BLOCKCHAIN
NEW BLOCK of transactions includes a security 'hash' of previous block.
BLOCKCHAIN new blocks are added approximately every 10 mins.
Multiple computer 'mining' servers check the validity of the blocks in the chain. The quickest to check the latest block, gets payment.
OR
successful transactions
BLOCK validated as 'honest' part of the chain.
REJECTED

Blockchain Innovations

01

Bitcoin was the first Implementation of Blockchain Application in 2008

The person or persons who designed bitcoin , as

part of the implementation, he also devised the

first blockchain database to be used and is the most successful till date.

02

Blockchain as application was the second innovation

It was essentially the realization that the underlying technology used by bitcoin could be separated and used for other operations also

03

The THIRD was the most "smart contract" embodied in a Blockchain 2.0 system with introduction to Ethereum

_It built little computer programs directly into blockchain that allowed financial instruments

04

The FOURTH critical innovation is "proof of stake"

Current generation is secured by "proof of work."The new system is replacing them with complex financial instruments, for similar or higher degree of security

Blockchain and Distributed Ledger Technology

Blockchain vs. Traditional Databases

Let’s understand more about Blockchain & see how it differs from traditional databases and later on we’ll discuss about the distributed ledger technology.

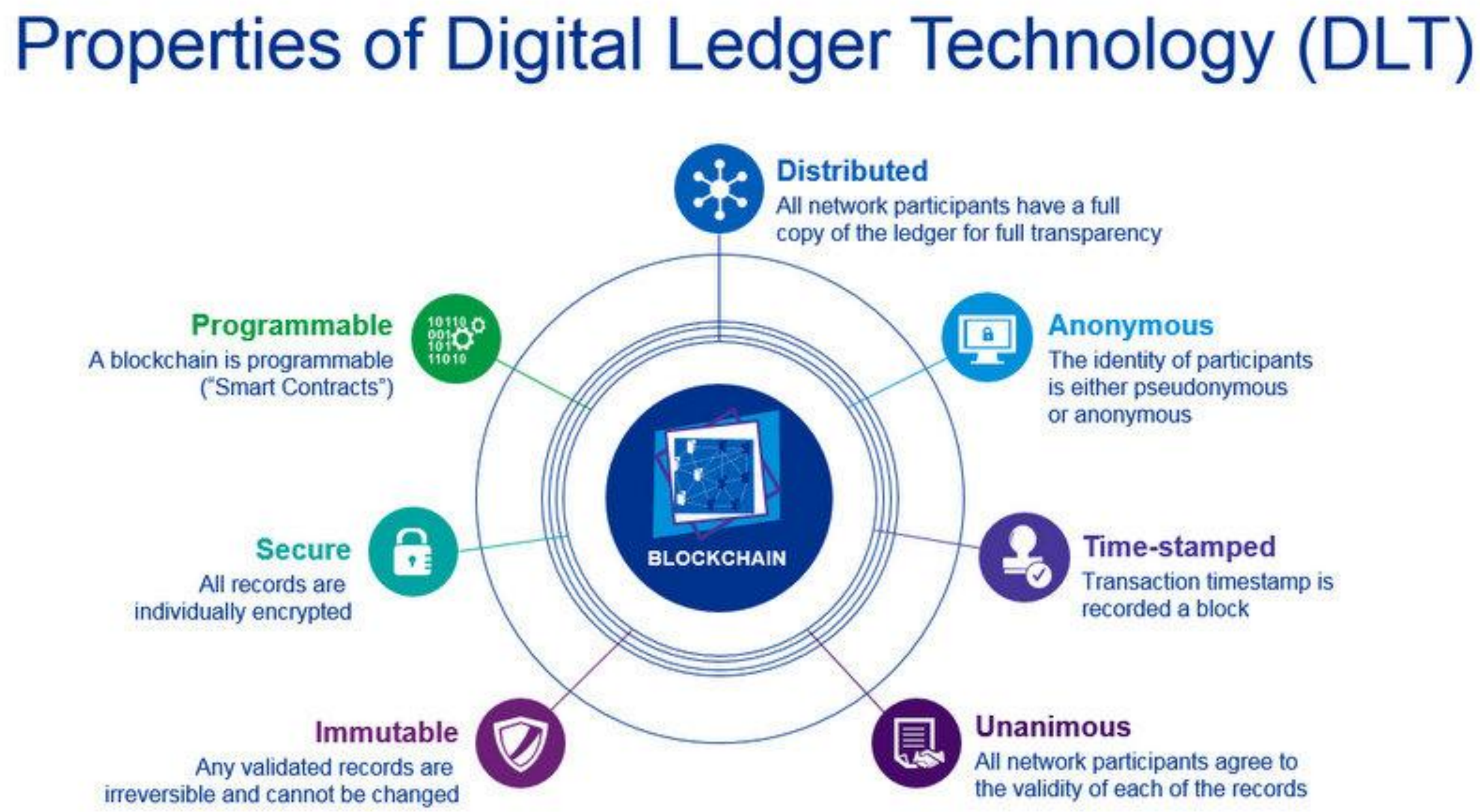
Blockchain vs. Traditional Databases

Characteristic	Block chain	Legacy Database
Data Ownership	Maintained through cryptographic key pairs and native cryptographic algorithms	Established via central authority
Privacy and Security	Cryptographic Authentication	Configuring each row based on enforcement from a central authority
Access Control	Inherently identical for all permissioned nodes	Centrally administered
Trust	Native via immutable records	Established via central authority
Data quality	Immutable records with automatic conflict resolution through consensus for transaction	Complex conflict resolution processes requires manual intervention
Database Validity	Continuous	Provided only for single instances in time
Data Propagation	Quick Propagation across all network nodes	Managed through multi version currency control(MVCC) and through custom synchronization
Enforce data Transformation	Built into Data layer Logic	None
Currency and Synchronization	Consensus yields Identical copies	Involves complex checking between central DB and users DB to ensure agreement
Reliability and availability	Peer to peer networking for distributed data replication across all nodes	Potential single point failure
Stored procedures	Smart Contracts	Not available
Transaction Creation	Available to all permissioned parties	Managed via central authority
Fraudulent/Malicious Changes	Immutability through reliance on previous block	Not available where current keys and check constraints remain insufficient

Distributed Ledger Technology (D.L.T.)

> “People think of blockchain technology and distributed ledger technology as one and the same. Interestingly enough, that is not the case. Let me show you the difference.”

Distributed Ledger Technology (D.L.T.)



Properties of Digital Ledger Technology (DLT)

Distributed
All network participants have a full copy of the ledger for full transparency

Anonymous
The identity of participants is either pseudonymous or anonymous

Time-stamped
Transaction timestamp is recorded a block

Unanimous
All network participants agree to the validity of each of the records

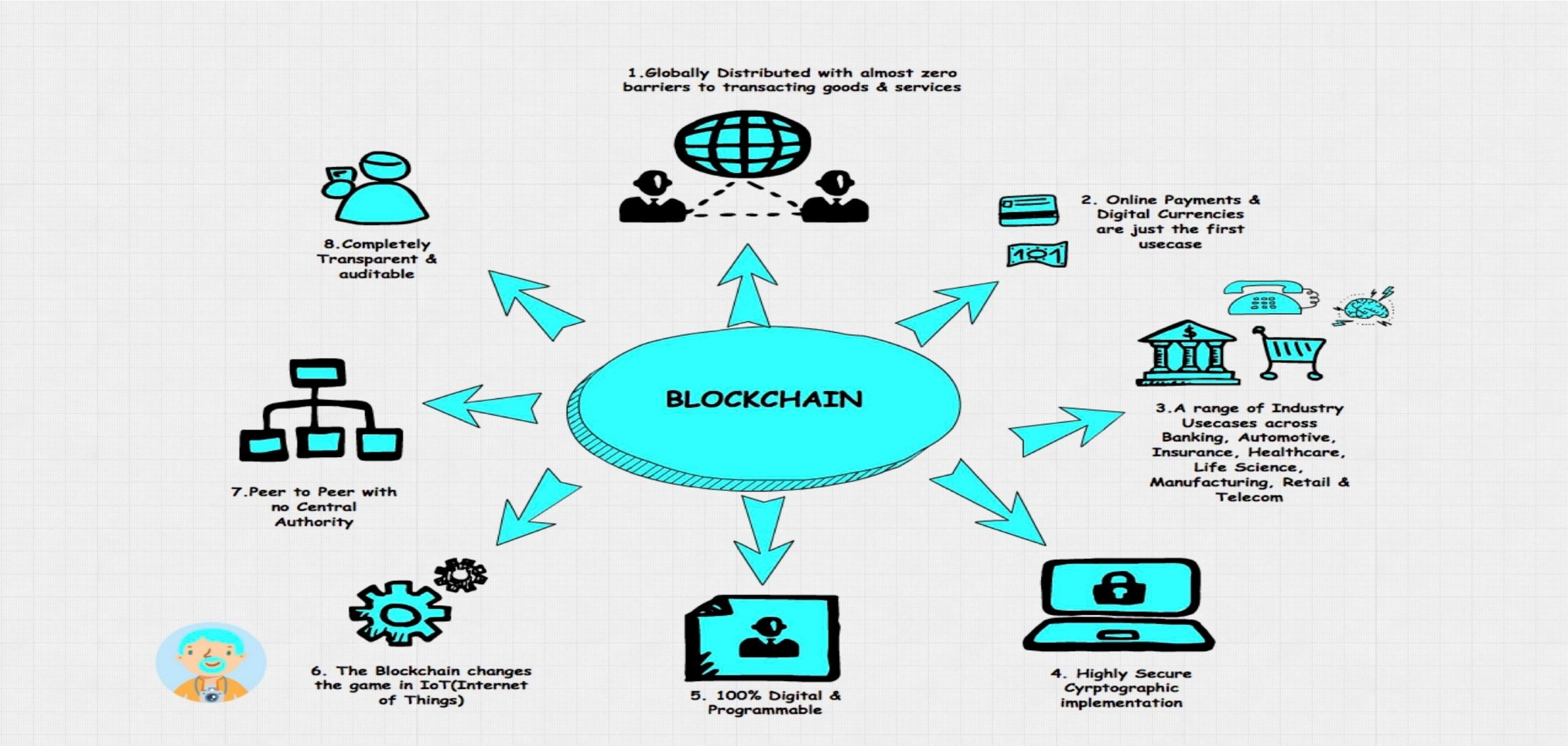
Immutable
Any validated records are irreversible and cannot be changed

Secure
All records are individually encrypted

Programmable
A blockchain is programmable ("Smart Contracts")

All blockchains are distributed ledgers, but <u>not</u> all distributed ledgers are blockchains

Distributed Ledger Technology (D.L.T.)



BLOCKCHAIN

1.Globally Distributed with almost zero barriers to transacting goods & services

2. Online Payments & Digital Currencies are just the first usecase

3.A range of Industry Usecases across Banking, Automotive, Insurance, Healthcare, Life Science, Manufacturing, Retail & Telecom

4. Highly Secure Cyrptographic implementation

5. 100% Digital & Programmable

6. The Blockchain changes the game in IoT(Internet of Things)

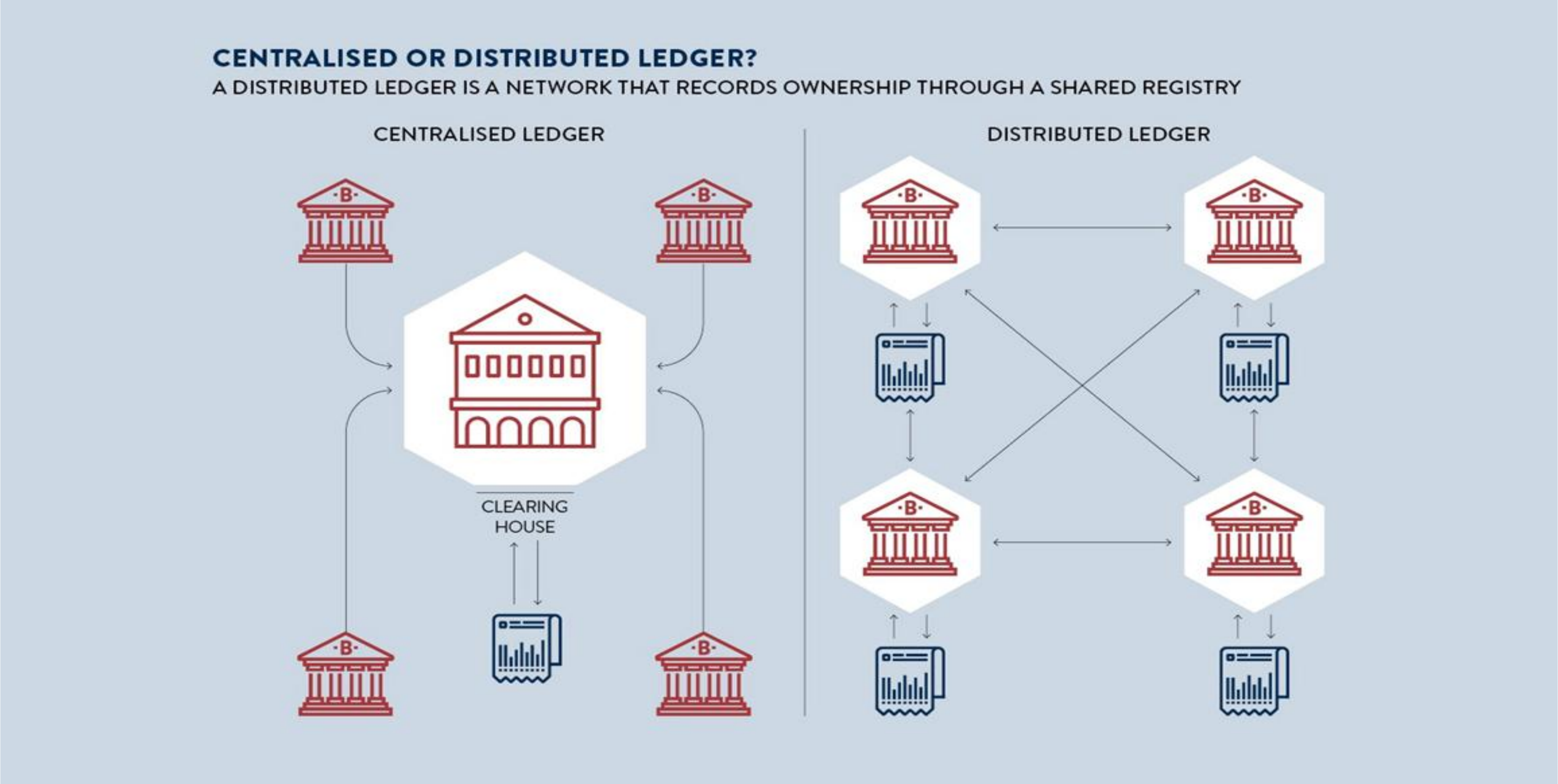
7.Peer to Peer with no Central Authority

8.Completely Transparent & auditable

Comparing Popular Blockchain Frameworks

Characteristic	Ethereum	Hyperledger Fabric	R3 Corda
Description of Platform	Generic blockchain platform	Modular block chain platform	Specialized distributed ledger platform for financial industry
Governance	Ethereum developers	Linux Foundation	R3
Mode of operation	Permission less, public or private	Permissioned, private	Permissioned, private
Consensus	• Mining based on Proof of work(PoW) • Ledger level	• Broad understanding of consensus that allows multiple approaches • Transaction Level	• Specific understanding of consensus (i.e notary nodes) • Transaction Level
Smart Contracts	Smart contract code(e.g. solidity)	Smart contract code(e.g. Go, Java)	• Smart contract code(e.g. Kotlin, Java) • Smart legal contract(legal prose)
Currency	• Ether • Tokens via smart contract	• None • Currency and tokens via chaincode	None

Distributed Ledger and Blockchain


CENTRALISED OR DISTRIBUTED LEDGER?
A DISTRIBUTED LEDGER IS A NETWORK THAT RECORDS OWNERSHIP THROUGH A SHARED REGISTRY
CENTRALISED LEDGER
DISTRIBUTED LEDGER
B
CLEARING HOUSE

Blockchain Structure

Blockchain Structure

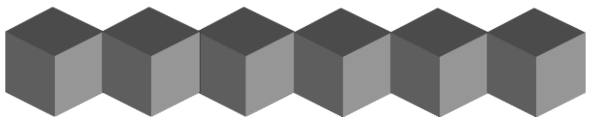
> Let’s see what goes inside the block of bitcoin blockchain.

Block Structure in Bitcoin Blockchain

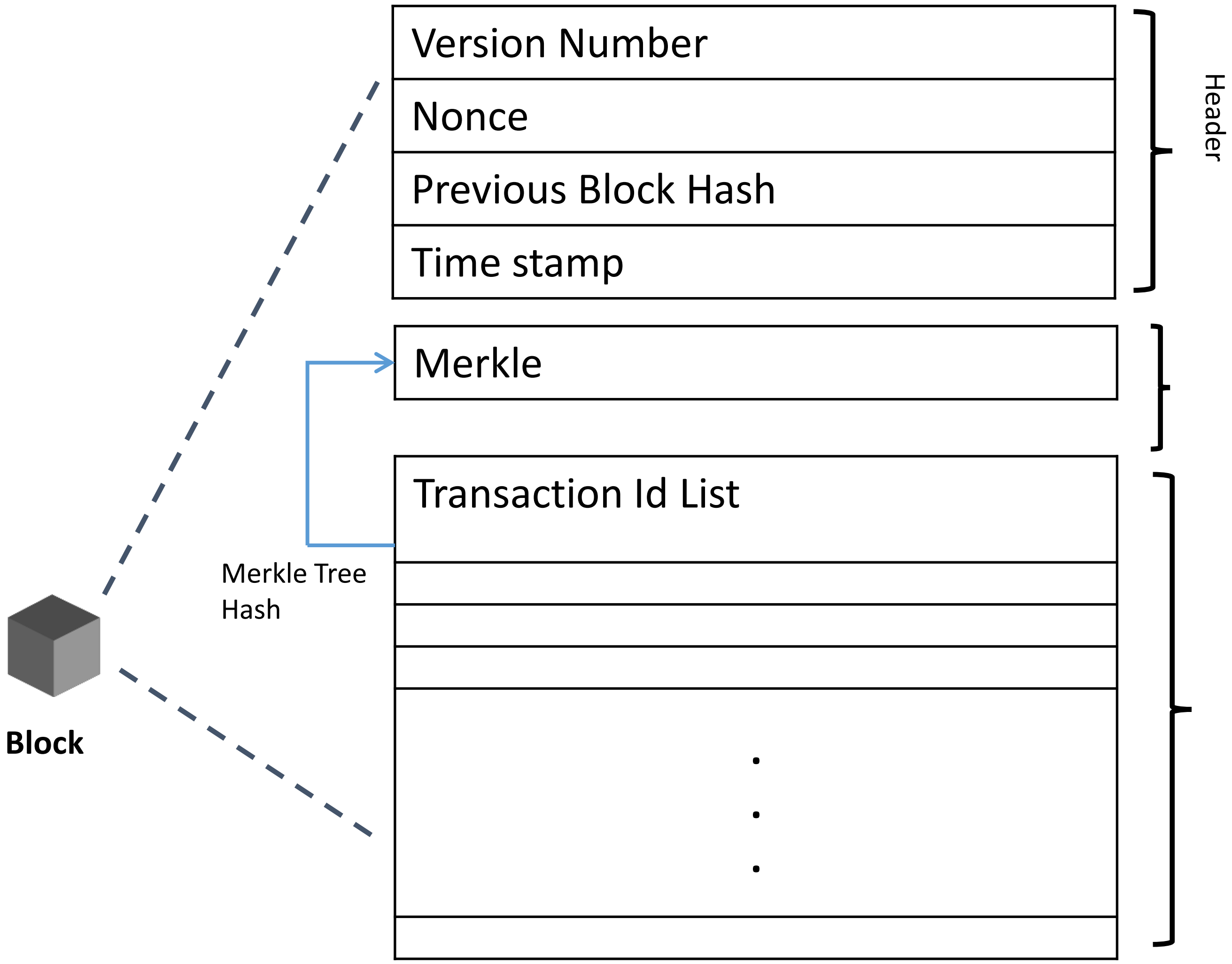
Header- Contains version info, nonce, previous block id and time stamp

Merkle- A hash built from block's transaction identifiers

List of records- Identification hashes that was included into the block's Merkle tree



Blockchain



Structure Of The Bitcoin Block Header

Consists of three sets of block metadata

- Reference to a previous block hash
- The difficulty, timestamp and nonce
- The Merkle root

Size	Field	Description
4 bytes	Version	A version number To track software/protocol upgrades
32 bytes	Previous Hash Block	A reference to the hash of the previous block in the chain
32 bytes	Merkle Root	A hash of the ƌoot of the Meƌkle tƌee of this ďloĐk͛s ƌeĐoƌds
4 bytes	Timestamp	The approximate creation time of this block
4 bytes	Difficulty target	The proof-of-work algorithm difficulty target of the block
4 bytes	Nonce	A counter used for proof-of-work algorithm

Block Identifiers: Block header hash & height

Block Header Hash

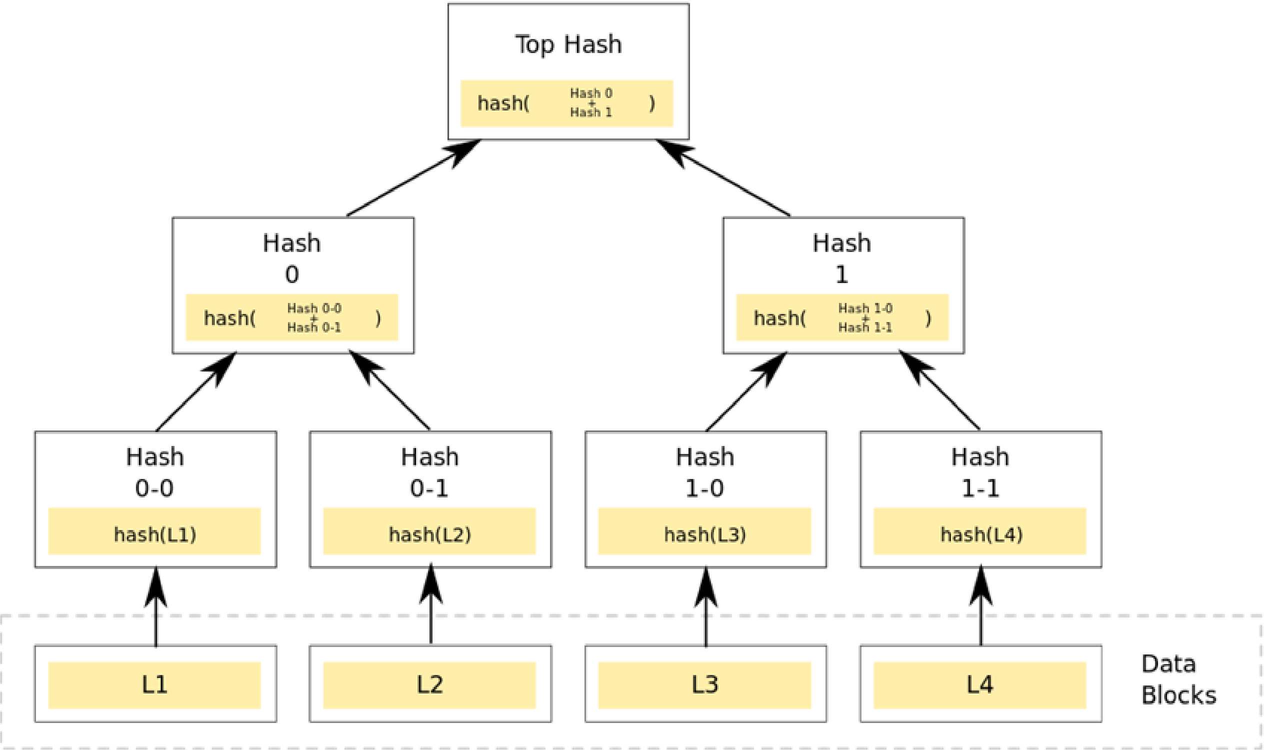
- The primary identifier of a block is its cryptographic hash
- A digital fingerprint, made by hashing the block header twice resulting 32-byte hash
- The block hash identifies a block uniquely
- The block hash is not included inside the block's data structure

Block Height

- It is the position of the block in the blockchain
- The first ever block created is at block height zero
- Each block added on top has one position higher in the blockchain
- It is also not a part of the block's data structure
- Each node dynamically identifies a block's height in the blockchain

Merkle Trees

- Also known as binary hash tree
- A data structure used for summarizing and verifying the integrity of large sets of data
- It contains cryptographic hashes
- Displayed upside down with the “root” at top and the “leave” at the bottom



A block of one or more new records is collected and such records are then hashed, and the hashes are then paired, hashed, paired again, and hashed again until a single hash remains

That single hash is called the Merkle root of a merkle tree

Advantages of Merkle trees

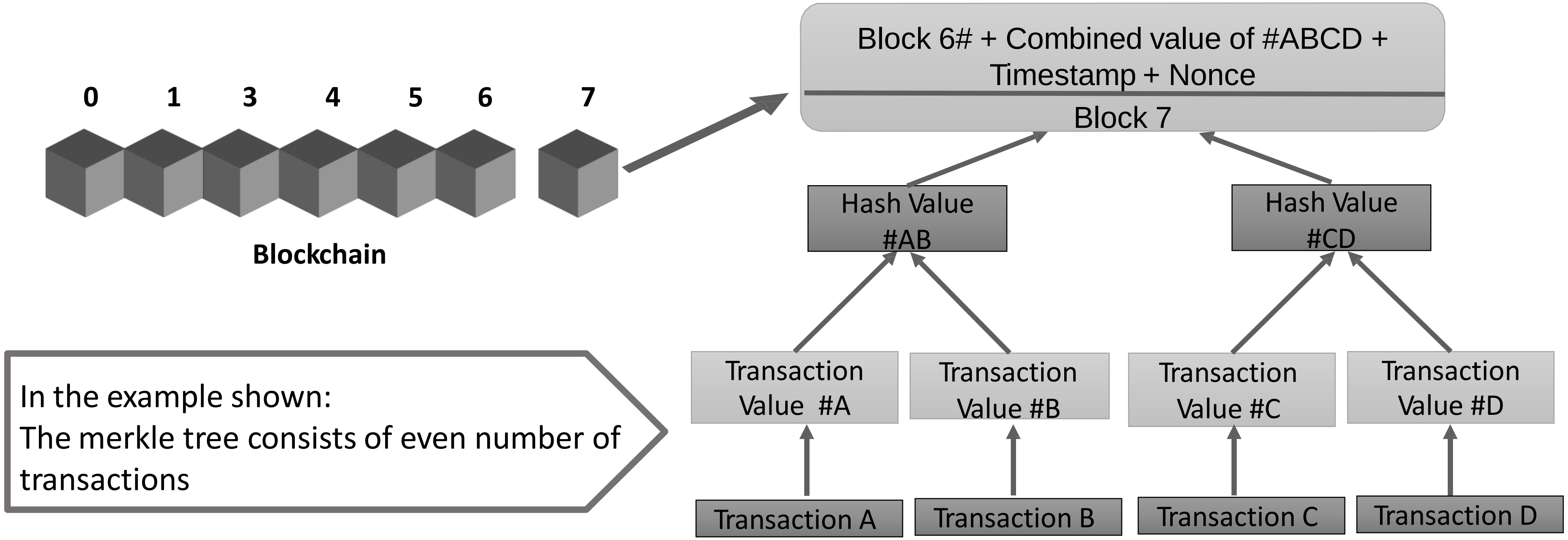
Merkle tree proofs and management requires only a very small and terse amount of information to be transmitted across a network

Advantages

A tree constructed by hashing paired data (the leaves), then pairing and hashing the results until a single hash remains, the merkle root. In Bitcoin, the leaves are almost always transactions from a single block.

Merkle trees require little memory / disk space and proofs are computationally easy and fast

Merkle Tree of Even Number of Transactions

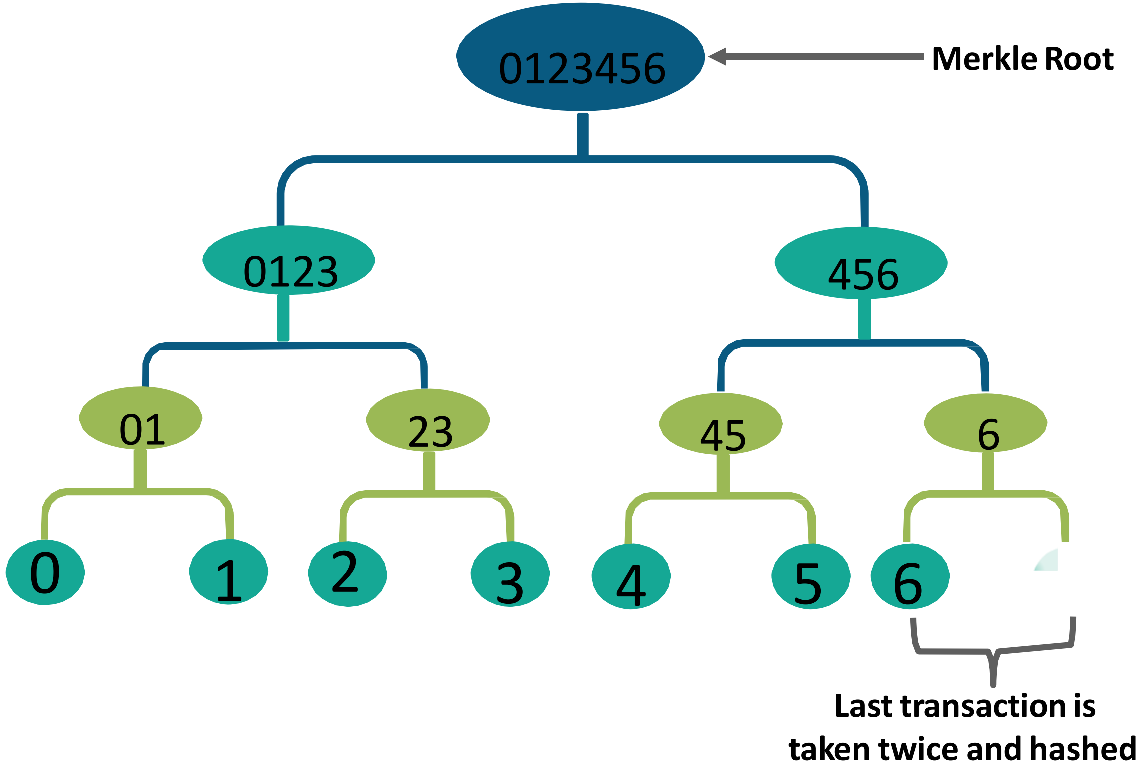


In the example shown:
The merkle tree consists of even number of transactions

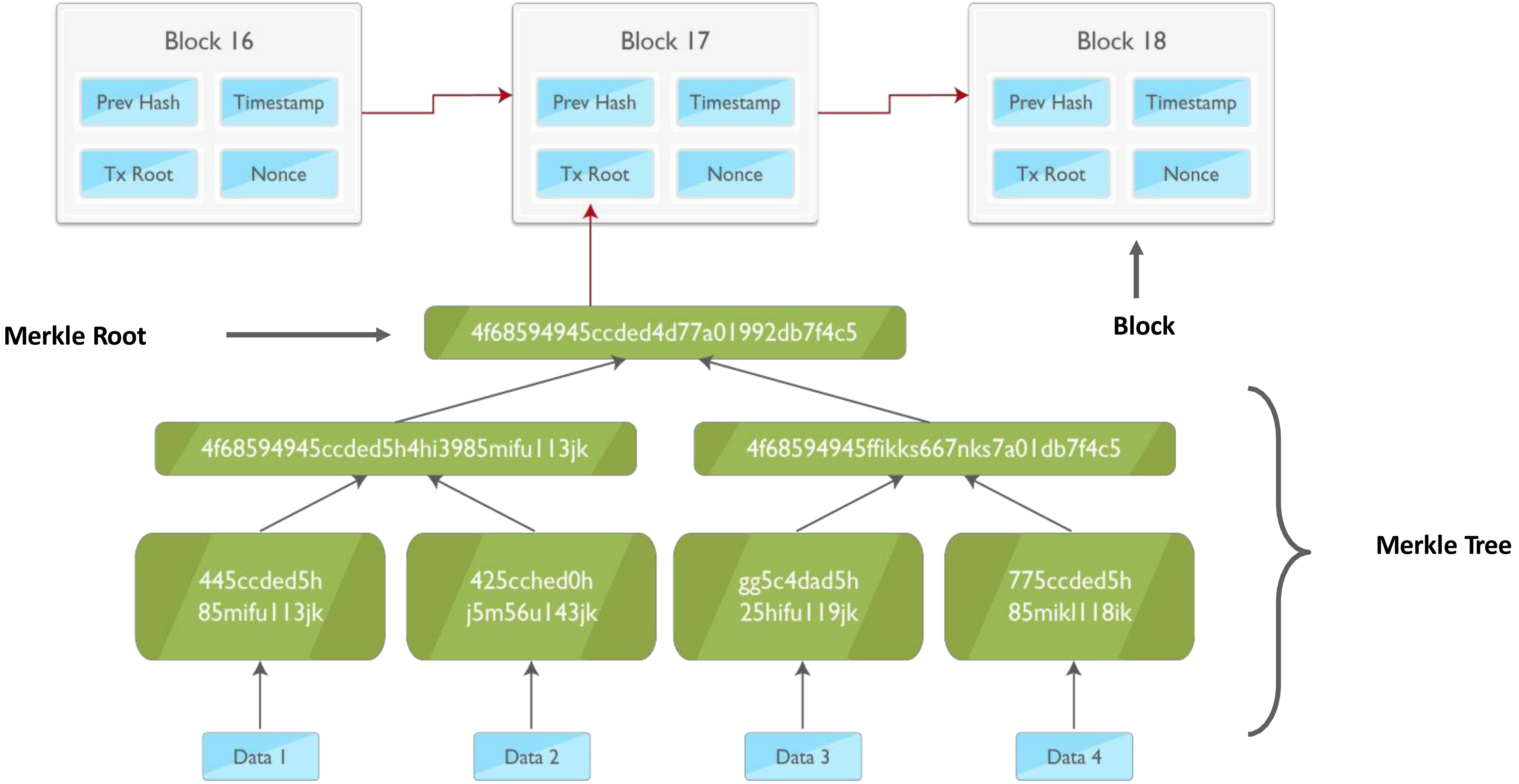
Merkle Trees of Odd Number of Transactions

Let’s suppose there are odd number of transactions in a block
In this case: The last transaction is hashed with itself

Refer the shown infographic



Blockchain Illustration



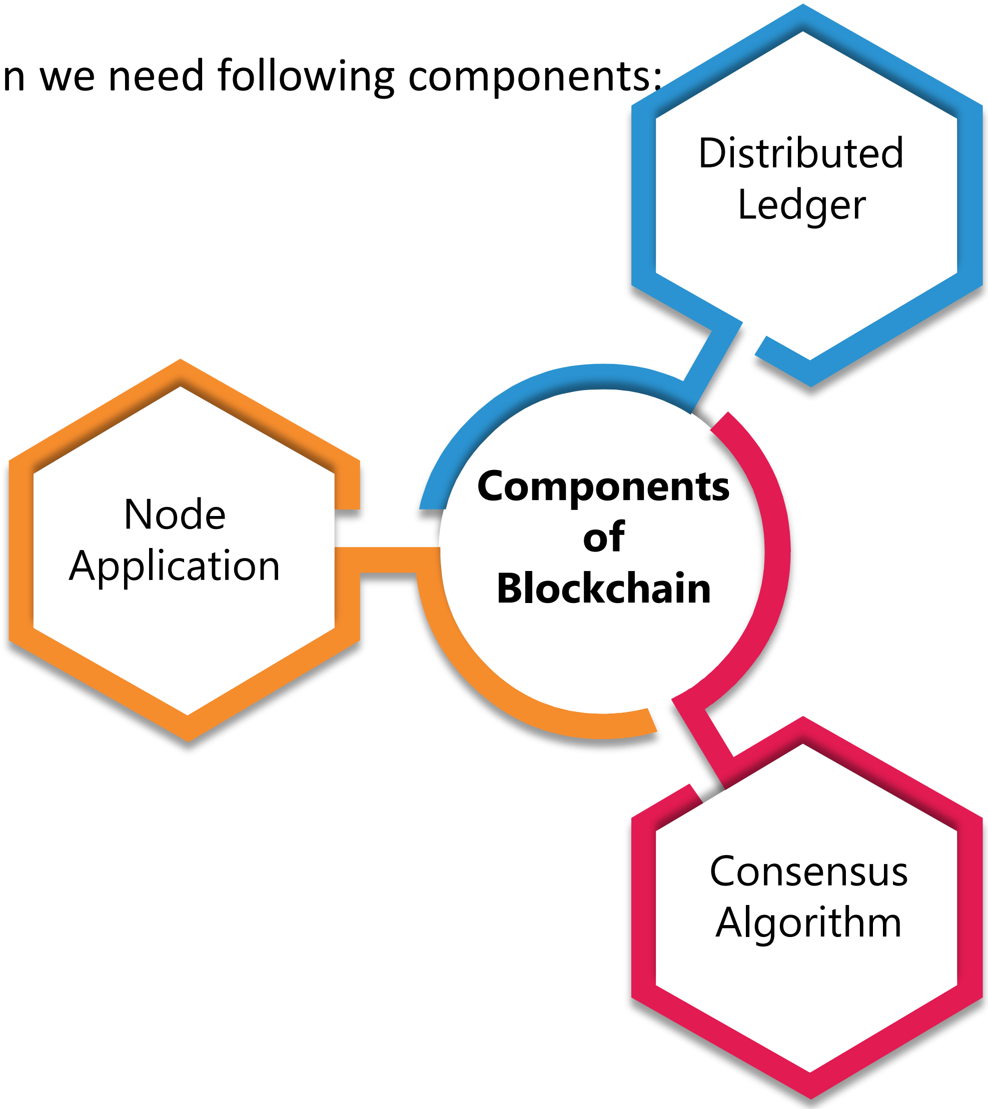
Blockchain Ecosystem

Components of Blockchain Ecosystem

> Let’s have a look at the components required to implement blockchain

Main Components of a Ecosystem

To implement blockchain we need following components:



Cryptography and Consensus Algorithms

> We know that cryptography and consensus algorithms are key constituents of any blockchain.

Cryptography Algorithms

Triple DES
Triple Des uses three individual keys with 56 bits each. The total key length adds up to 168 bits.

01

RSA
RSA is public key encryption algorithm and the standard for encrypting data sent over the internet

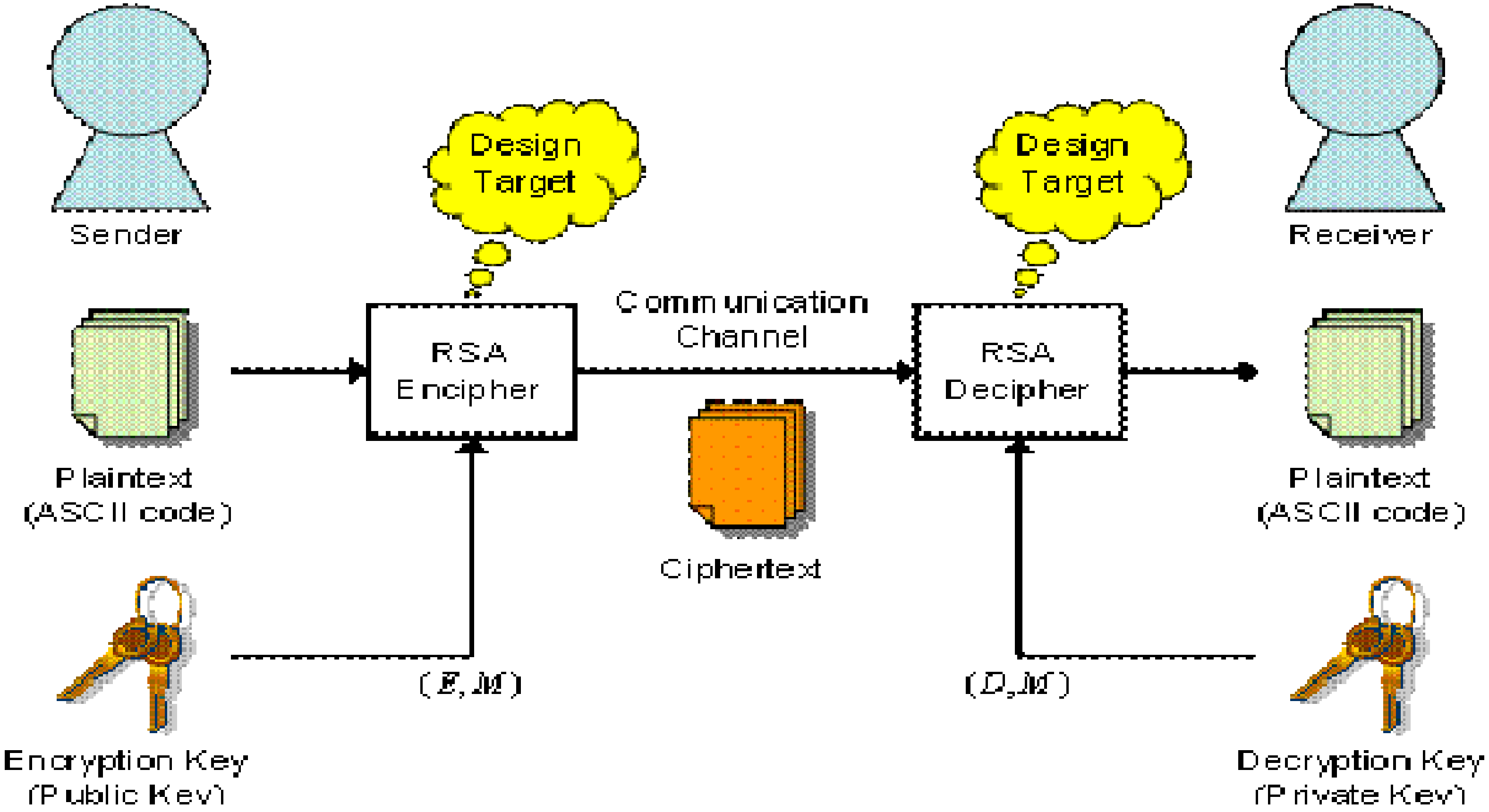
02

Blowfish
Known for its speed and effectiveness symmetric cipher splits messages into blocks of 64 bits and encrypts them individually

03

RSA Algorithm

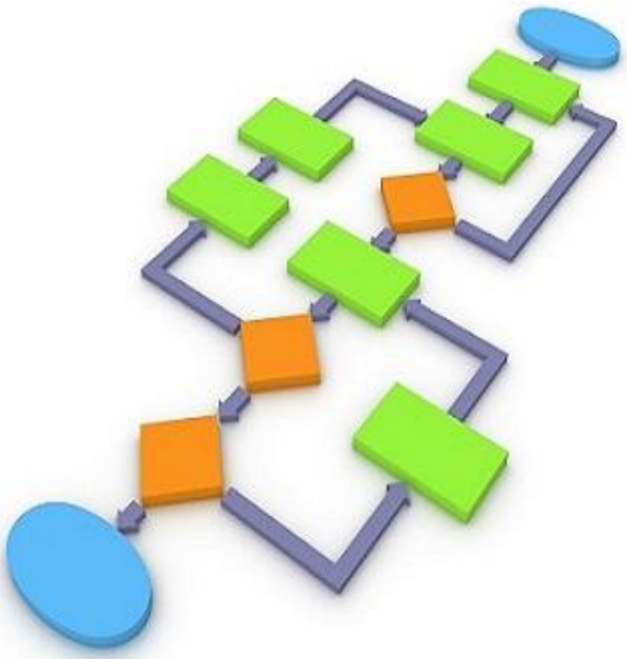
RSA is a widely used encryption algorithm in blockchain technology, let’s discuss RSA algorithm extensively.



RSA in action

> So we now have a public key and a private key and a method for encrypting and decrypting using those keys, let’s see an example to clear the details

Security of RSA



There are different approaches used in attacking the RSA algorithm:

- Brute force: It involves all possible secret keys

- Mathematical attacks: In mathematical attack we are using different techniques, which is similar in effort to
- factor the product of two primes

Consensus Algorithms

> Let’s talk about the popular consensus algorithms

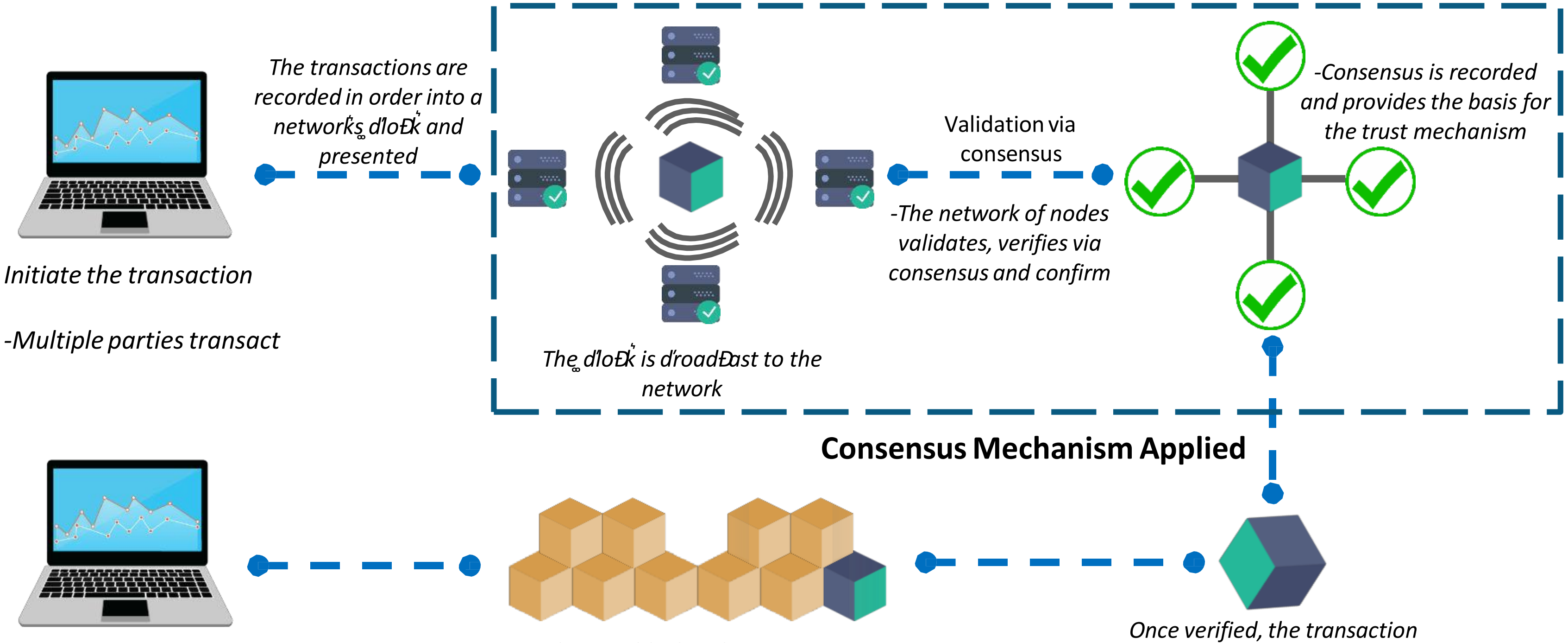
Consensus

> The consensus provides the technical infrastructure layer for blockchains. This makes it one of the most critical components when assessing real-world use cases
>
> Consensus is key, because without a central authority, the participants have to agree on the rules and how to apply them

Consensus does two things:

1. It ensures that the next block in a blockchain is the one and only version of the truth
2. It keeps powerful adversaries from derailing the system

Consensus is the Heart of Blockchain



Initiate the transaction

-Multiple parties transact

The transactions are recorded in order into a networḱs ďloĐk and presented

The ďloĐk is ďroadĐast to the network

Validation via consensus

-The network of nodes validates, verifies via consensus and confirm

-Consensus is recorded and provides the basis for the trust mechanism

Consensus Mechanism Applied

Once verified, the transaction becomes a part of new block for the ledger

The new block is then added to the existing blockchain

The transaction is complete

What is Needed for Consensus

1 The basic acceptance of laws, rules and norms to decide a transaction

2 The common acceptance of industry and institution that apply these laws and rules

Basic Parameters of A Consensus Mechanism

1. Decentralized Mechanism
A single central authority must not provide transaction decision finality

2. Quorum Architecture
Nodes exchange messages in predefined mechanism, which may include stages or tiers

3. Validation
Process provides means to verify the participants identities

4. Integrity
It enforces the authentication of the transaction integrity & valid

5. Non Cancellation
This provides means to verify that the supposed sender really sent the message

6. Confidentiality
This provides means to verify that the supposed sender really sent the message

7. Fault Endurance
The network operates efficiently and quickly even if some nodes fail or are slow

8. Fulfillment
It considers throughput, liveness, latency

Consensus History

Byzantine General Problem

The Byzantine army has completely encircled the city.

The army has many divisions and each division has a general. The generals communicate between each as well as between all lieutenants within their division only through messengers.

All the generals or commanders have to agree upon one of the two plans of action. Exact time to attack all at once or if faced by fierce resistance then the time to retreat all at once. The army cannot hold on forever. If the attack or retreat is without full strength then it means only one thing — Unacceptable brutal defeat.

If all generals and/or messengers were trustworthy then it is a very simple solution. However, some of the messengers and even a few generals/commanders are traitors.

Results
No solution exists if less than or equal to 2/3 generals are loyal

Solution to Byzantine General Problem

> The practical byzantine fault tolerance algorithm (PBFT), which is used to establish consensus in blockchain systems, is only one of those potential solutions

PBFT

> The solution came in 1999, when Miguel Castro and Barbara Liskov introduced the PBFT algorithm
>
> PBFT can process an enormous number of direct peer-to-peer messages with minimal latency

How it works:

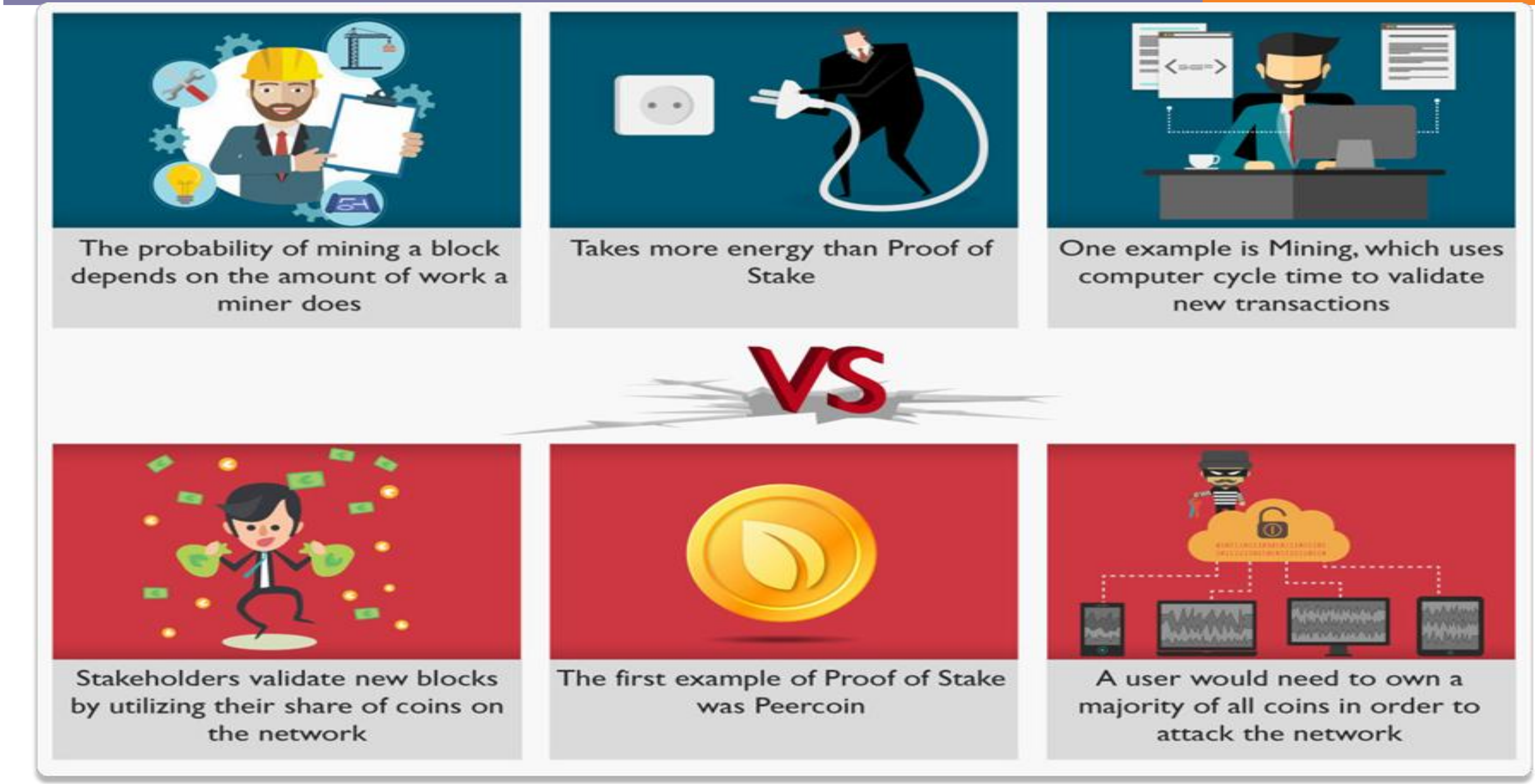
- Asynchronous distributed system where nodes are connected by a network
- Byzantine failure model
  - faulty nodes behave arbitrarily
  - independent node failures
- Cryptographic techniques to prevent spoofing and replays and to detect corrupted messages
- Very strong adversary

This method of establishing consensus requires less effort than other method

Proof-of-Stake

- Proof of Stake (PoS) is a category of consensus algorithms for public blockchains that depend on a validator's economic stake in the network.
- Newer kind of consensus algorithm
- Pioneered by Peercoin (2011), now many versions exist (NXT, Tendermint, Flying Fox, etc)Your chance of being picked to create the next block depends on the fraction of coins in the system you own
- A participant with nothing to lose has no reason not to behave badly. This is called nothing at stake problem.

Proof-of-Work v/s Proof-of-Stake



The probability of mining a block depends on the amount of work a miner does

Takes more energy than Proof of Stake

One example is Mining, which uses computer cycle time to validate new transactions

Stakeholders validate new blocks by utilizing their share of coins on the network

The first example of Proof of Stake was Peercoin

A user would need to own a majority of all coins in order to attack the network

Proof-of-Elapsed Time

- Intel developed its own alternative consensus protocol called proof-of-elapsed time
- Proof of Elapsed Time (PoET), a Nakamoto-style consensus algorithm that is designed to be a production-grade protocol capable of supporting large network populations.
- PoET relies on secure instruction execution to achieve the scaling benefits of a Nakamoto-style consensus algorithm without the power consumption drawbacks of the Proof of Work algorithm.
- PoET simulator, which provides PoET-style consensus on any type of hardware, including a virtualized cloud environment.
- Example : (PoET) is used by HYPERLEDGER SAWTOOTH

Proof of Elapsed Time - Advantages

For the purpose of achieving distributed consensus efficiently, a good lottery function (PoET) has several characteristics:

01 **Fairness:**
The function should distribute leader election across the broadest possible population of participants

02 **Investment:**
The cost of controlling the leader election process should be proportional to the value gained from it

03 **Verification:**
It should be relatively simple for all participants to verify that the leader was legitimately selected

Types of Blockchain

Blockchain Types

Public: A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network.

Examples – Bitcoin, Ethereum, Dash, Factom

Consortium: controlled by a consortium of members. Only predefined set of nodes have access to write the data or block.

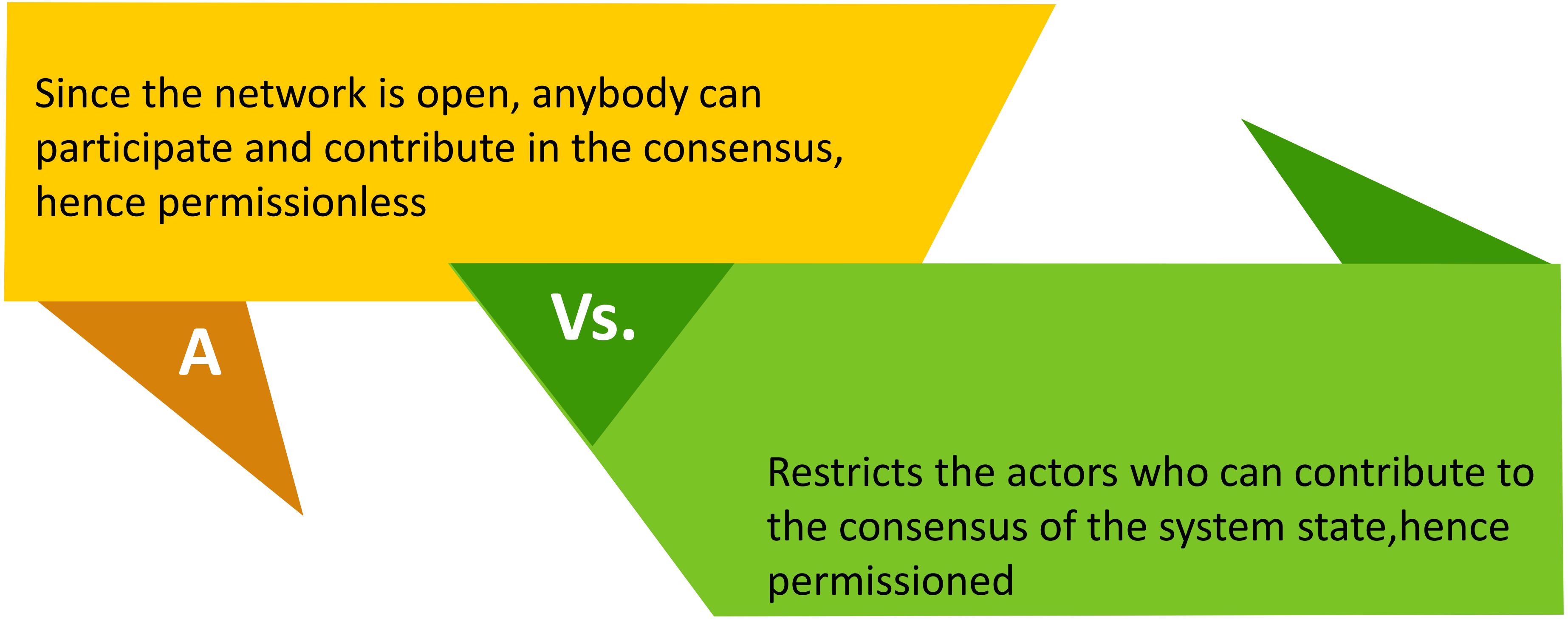
Examples- Ripple ,R3 & Hyperledger1.0

Private: A private blockchain network requires an invitation and must be validated by either the network starter or by a set of rules put in place by the network starter.

Examples- Multichain, Blockstack

Permissioned and Permissionless Blockchain

Properly permissioned blockchain networks differ from unpermissioned blockchain networks solely based on the access control layer built into the blockchain nodes



Since the network is open, anybody can participate and contribute in the consensus, hence permissionless

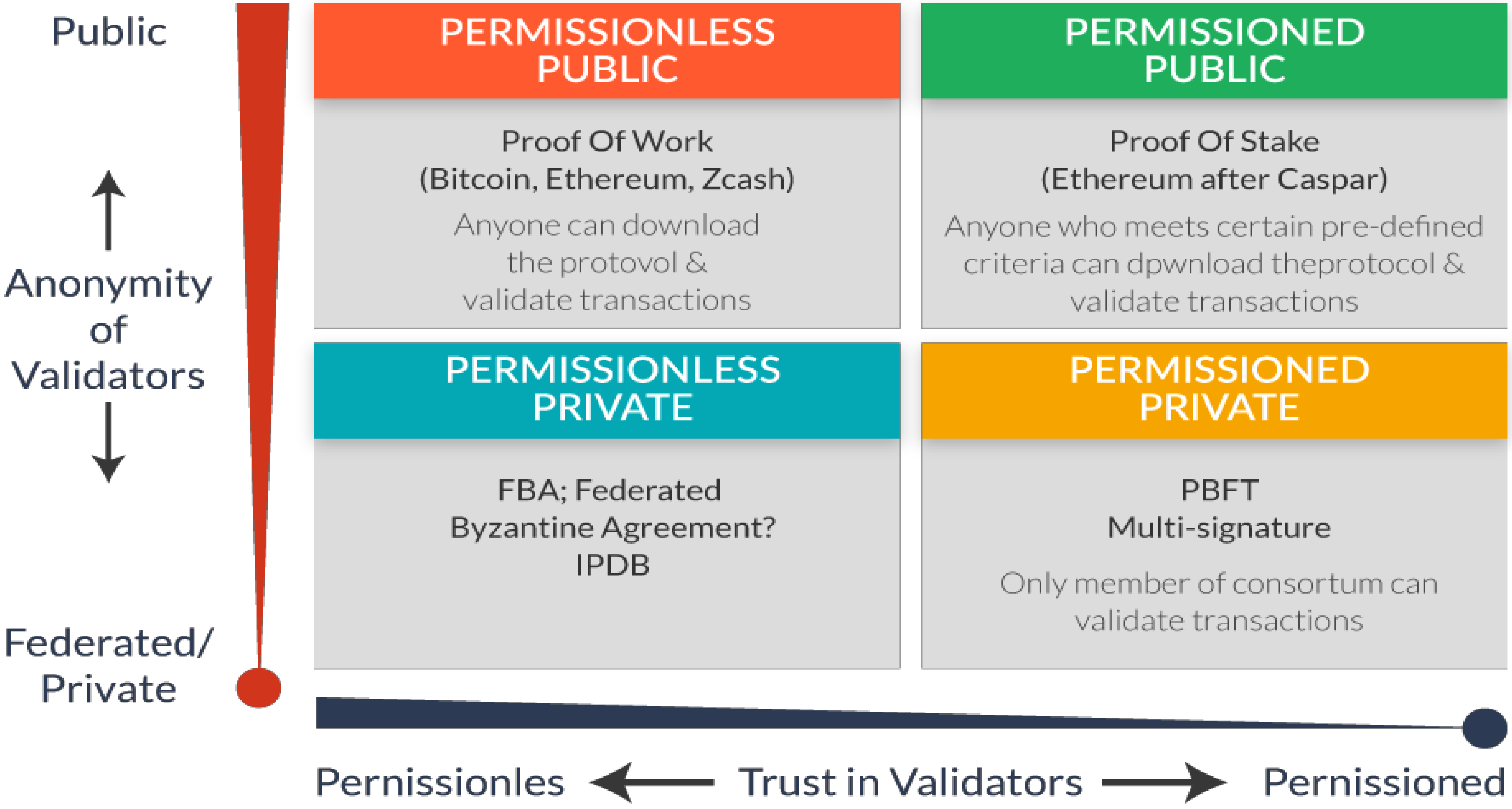
Restricts the actors who can contribute to the consensus of the system state,hence permissioned

A: Permission less B: Permissioned

Public v/s Private Blockchains

Features	Public	Private
Access	Public read/write access to database	Permissioned read and/or write access to database
Speed	Lower Performance	Faster Performance
Security	Proof-of-work/ Proof-of-stake	Pre-approved Participants
Identity	Anonymous/ Pseudonymous	Pre Approved identities
Asset	Native Assets	Any Native asset

Public v/s Private Blockchains



Public

Anonymity of Validators

Federated/ Private

PERMISSIONLESS PUBLIC

Proof Of Work
(Bitcoin, Ethereum, Zcash)

Anyone can download the protovol & validate transactions

PERMISSIONED PUBLIC

Proof Of Stake
(Ethereum after Caspar)

Anyone who meets certain pre-defined criteria can dpwnload theprotocol & validate transactions

PERMISSIONLESS PRIVATE

FBA; Federated Byzantine Agreement?
IPDB

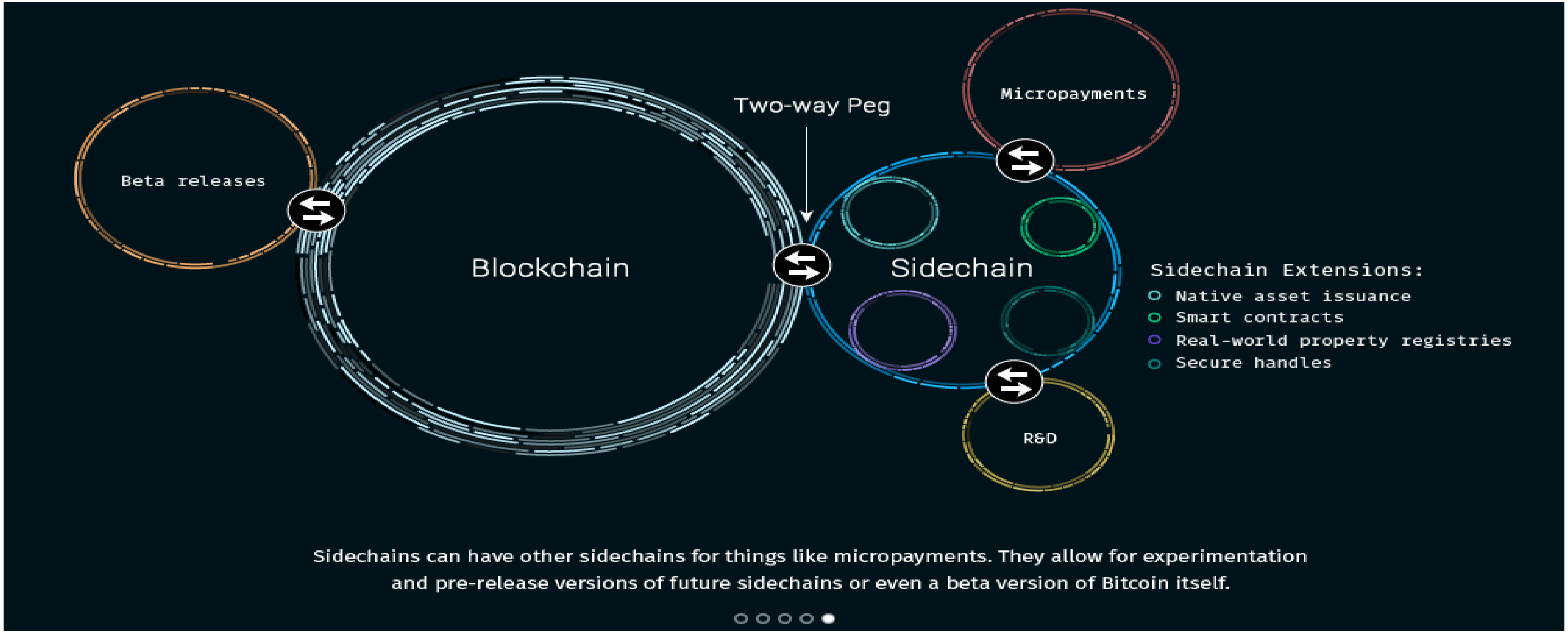
PERMISSIONED PRIVATE

PBFT
Multi-signature

Only member of consortum can validate transactions

Pernissionles ← Trust in Validators → Pernissioned

Side Chains



Sidechains can have other sidechains for things like micropayments. They allow for experimentation and pre-release versions of future sidechains or even a beta version of Bitcoin itself.

Sidechains are separate blockchains which is connected to other blockchains through the use of two-way peg which allows transfer of digital coins or assets between blockchains at a fixed or otherwise deterministic exchange rate

What Side Chains Offers?

1. Side chains Enhances Blockchains performance and privacy protections

2. Sidechaining is any mechanism that allows tokens from one blockchain to be securely used within a completely separate blockchain but still moved back to the original chain if necessary.

3. Side chains can have other side chains for micro payments

Transfer of Assets in Side Chains

Using Rootstock as an example, in order to transfer assets from one chain to the other

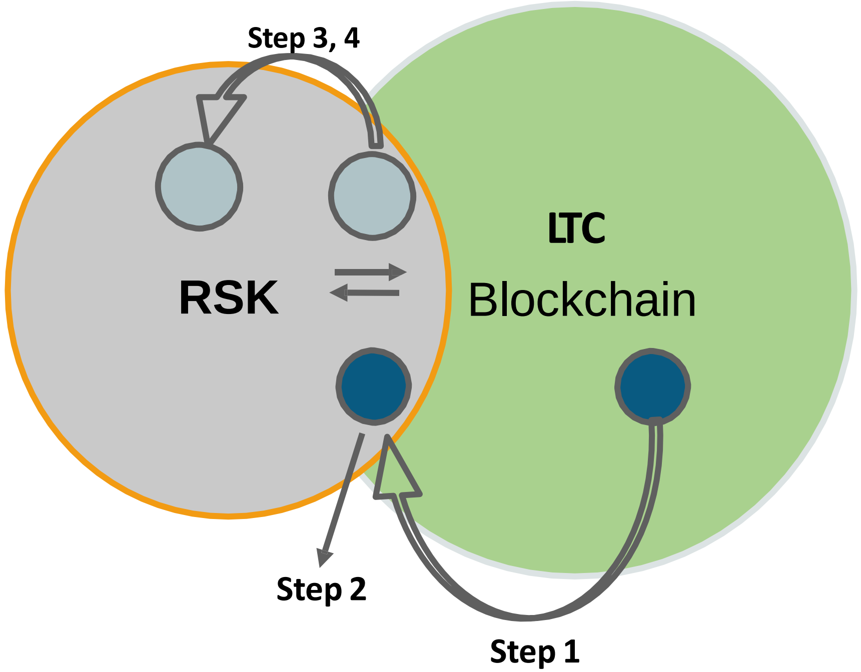
Step 1: A user on the parent first has to send their coins to a special output address

Step 2: Coins will consequently become locked and un-spendable

Step 3: After, the transaction completes, SPV then confirms it across the chains

Step 4: After waiting out a contest period, which is just a secondary method to help prevent double spending, the equivalent amount will becredited and spendable on the sidechain and vice versa

Sidechains have their own miners to help protect them from nefarious actors and attacks against the network



Platforms for implementing Blockchain

> There are many time of platforms which can be used to implement blockchain. Let's see them.

Various platform for implementing Blockchain

Ethereum: An open blockchain platform that lets anyone build and use decentralized applications that run on blockchain technology. Like Bitcoin no one controls or owns Ethereum

Multichain: A platform for the creation and deployment of private Blockchains (permissioned Blockchains) either within or between organizations

Hydrachain: A joint development effort of brainbot technologies and the Ethereum project. supports the creation of scalable blockchain based applications

Hyperledger: It is an open source collaborative effort created to advance cross-industry Blockchain technologies

Various platform for implementing Blockchain

Openchain: Well suited for organizations wishing to issue and manage digital assets. It takes a different approach than the standard Bitcoin approach on implementing Blockchain

IBM Bluemix: Built on top of the Hyperledger project and offers additional security and infrastructure facilities for enterprises

Chain: Yet another Blockchain platform well suited for financial applications. Based on "Chain Core" which is an enterprise software product

IOTA: A revolutionary new blockless distributed ledger which is scalable, lightweight and for the first time ever makes it possible to transfer value without any fees

Various platform for implementing Blockchain

BigChainDB: An open source system that “starts with a big data distributed database and then adds blockchain characteristics-decentralized control, immutability and the transfer of digital assets”

Corda: A distributed ledger platform with pluggable consensus

Quorum, an open source distributed ledger and smart contract platform based on Ethereum

Stellar, an open-source, distributed payments infrastructure that provides RESTful HTTP API servers which connect to Stellar Core, the backbone of the Stellar network.

Thank You

Email us – support@intellipaat.com

Visit us - https://intellipaat.com