Blockchain Technology

What is Blockchain?

At its most basic level, blockchain is literally just a chain of blocks, but not in the traditional sense of those words. When we say the words “block” and “chain” in this context, we are actually talking about digital information (the “block”) stored in a public database (the “chain”).

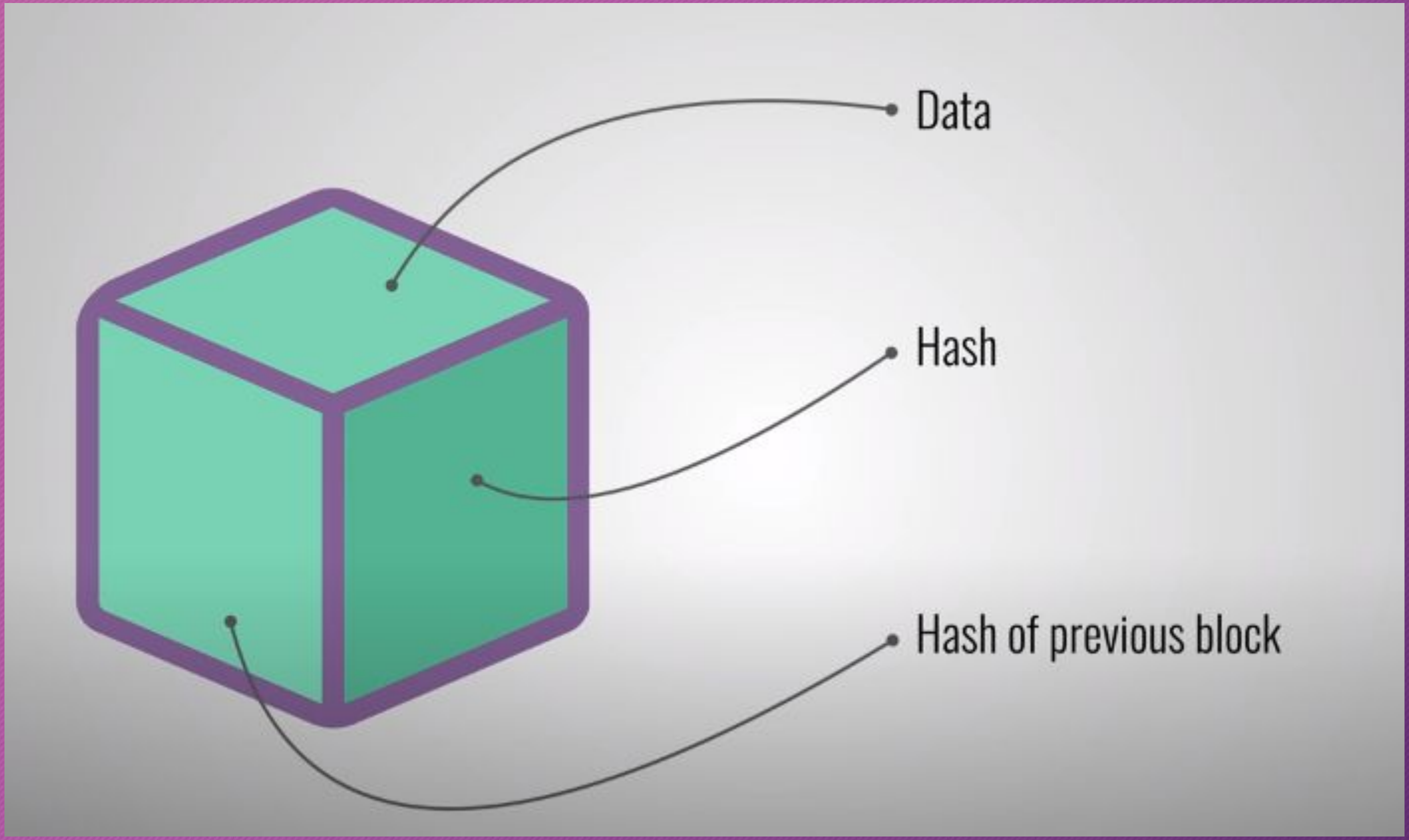
What exactly is Blockchain?

The blockchain is general terms is defined in following manner:

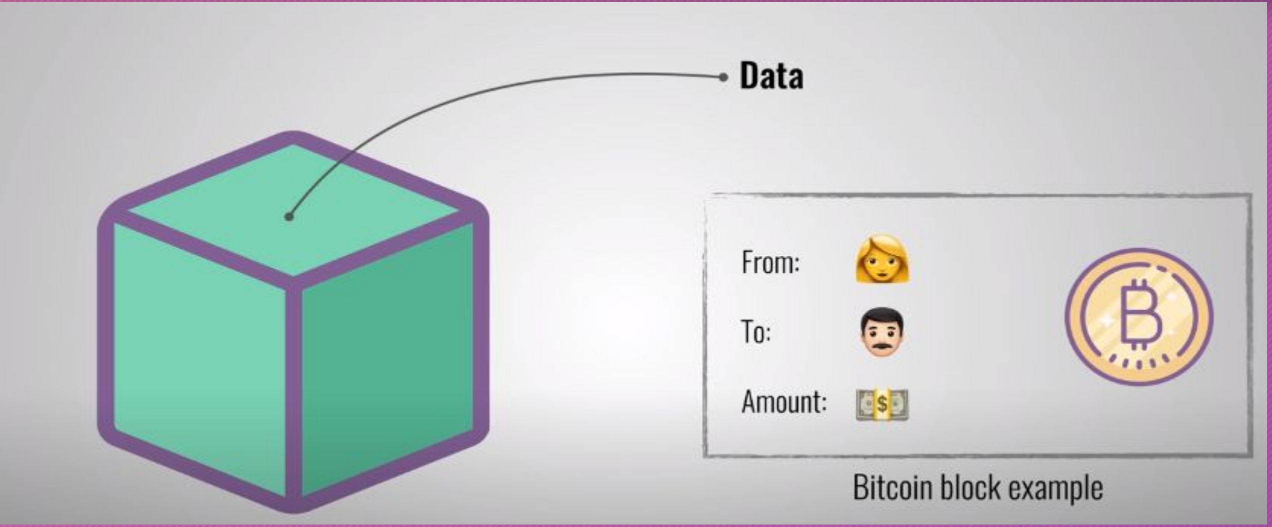
- A technology that permits transactions to be recorded permanently.
- A technology that cryptographically secure the system and chains data in chronological order.
- A technology that remove intermediaries and create trust through the algorithm.

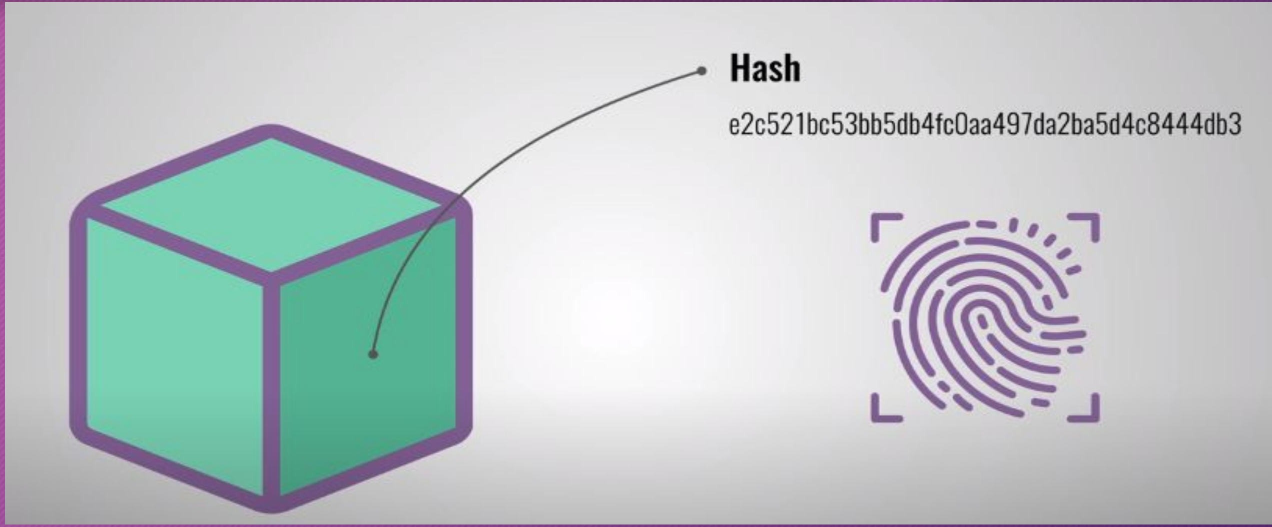
Blockchain was first introduced in the white paper for Bitcoin in 2009 by unknown person named as Satoshi Nakamoto. Currently, blockchain is used my multiple organizations to tackle their problems and provide a better solution

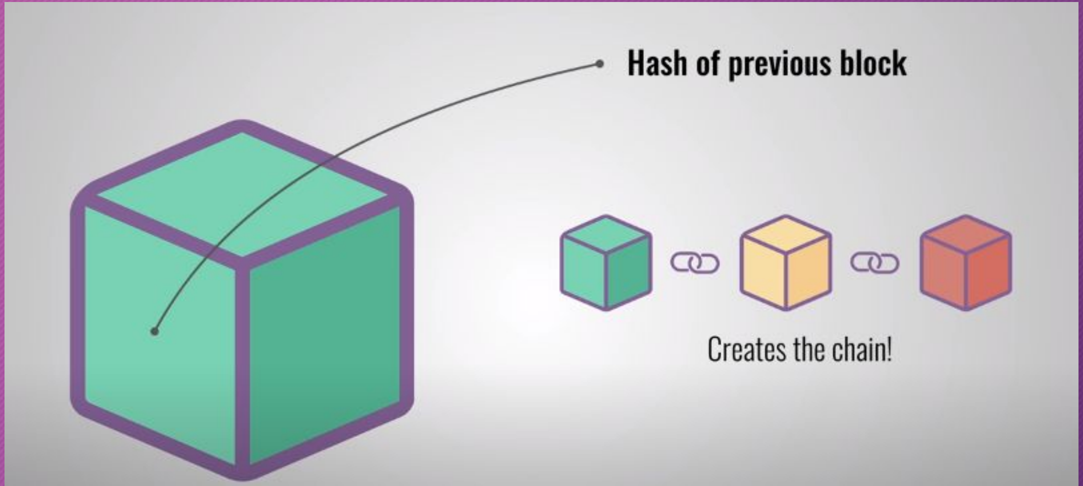
What is a Block?



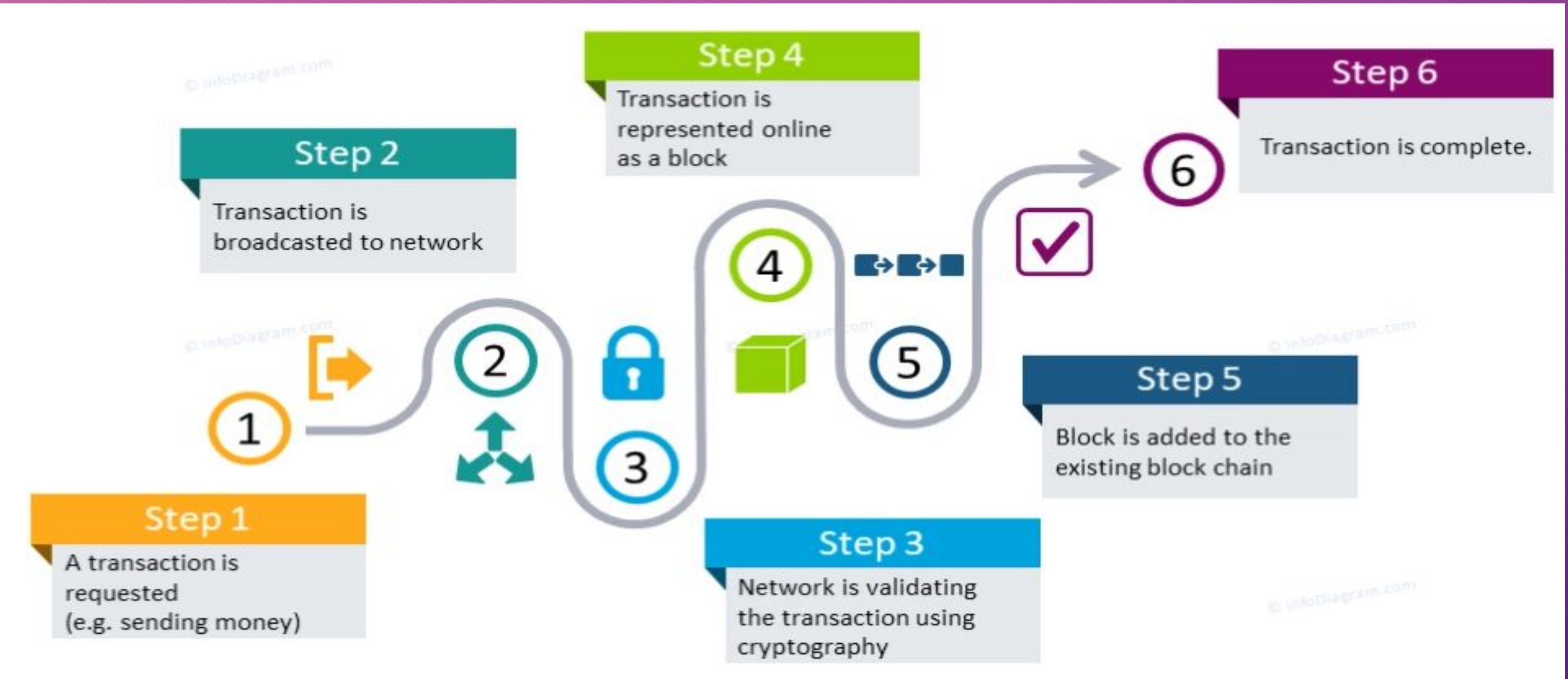
Bitcoin Example







Working of Blockchain.



Step 1

A transaction is requested (e.g. sending money)

Step 2

Transaction is broadcasted to network

Step 3

Network is validating the transaction using cryptography

Step 4

Transaction is represented online as a block

Step 5

Block is added to the existing block chain

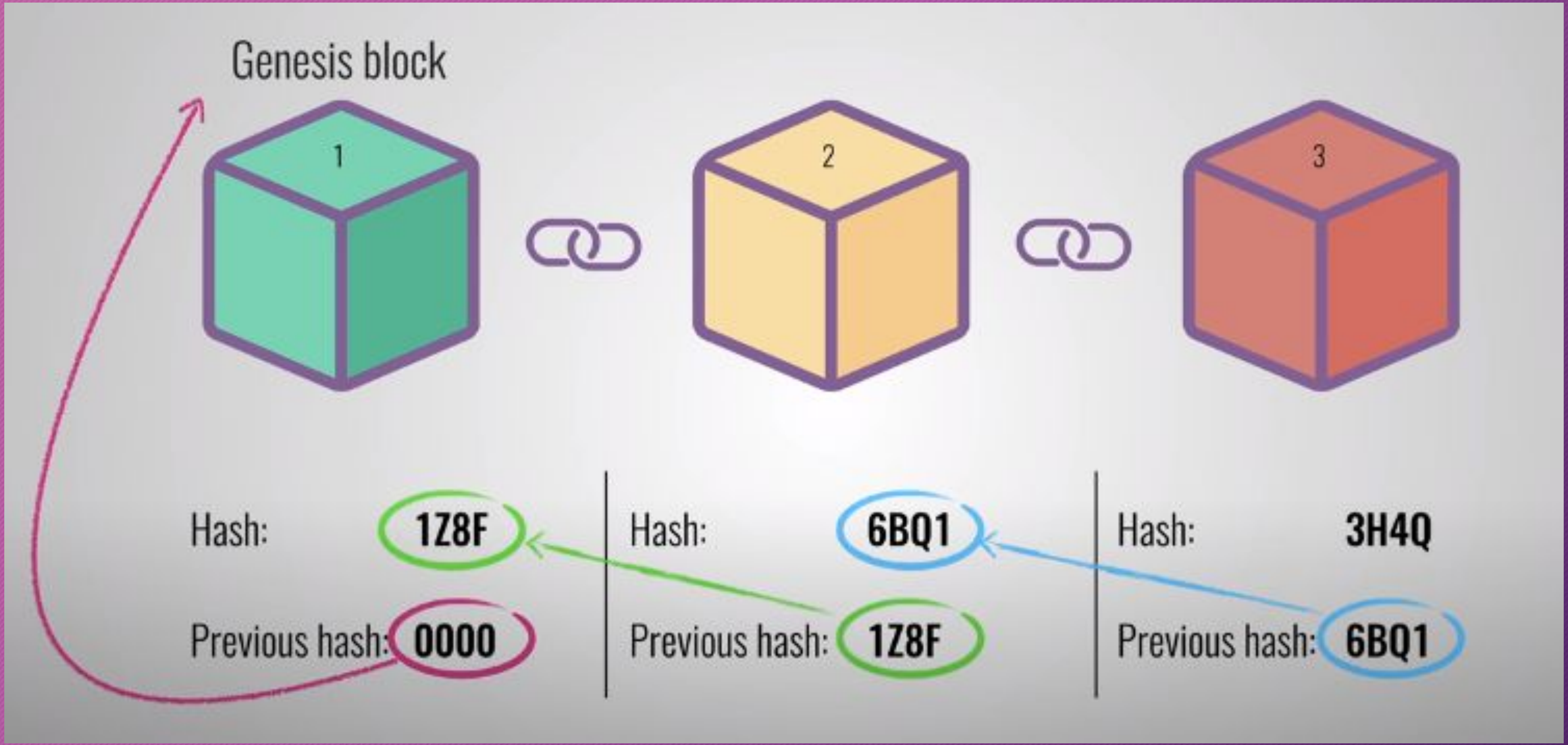
Step 6

Transaction is complete.

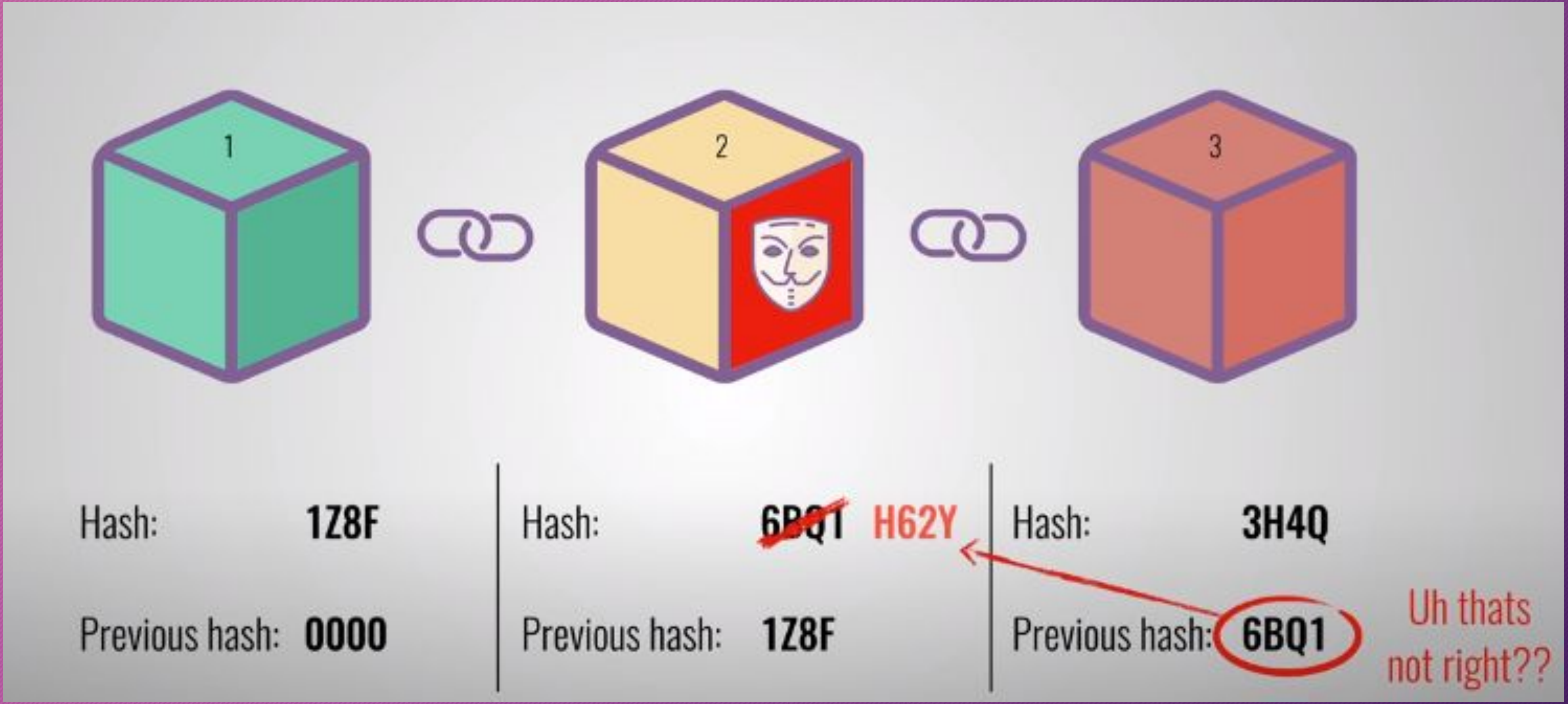
Hash Generator

https://passwordsgenerator.net/sha256-hash-generator/

Blockchain Example



Intrusion



Hash: 1Z8F

Previous hash: 0000

Hash: ~~6BQ1~~ H62Y

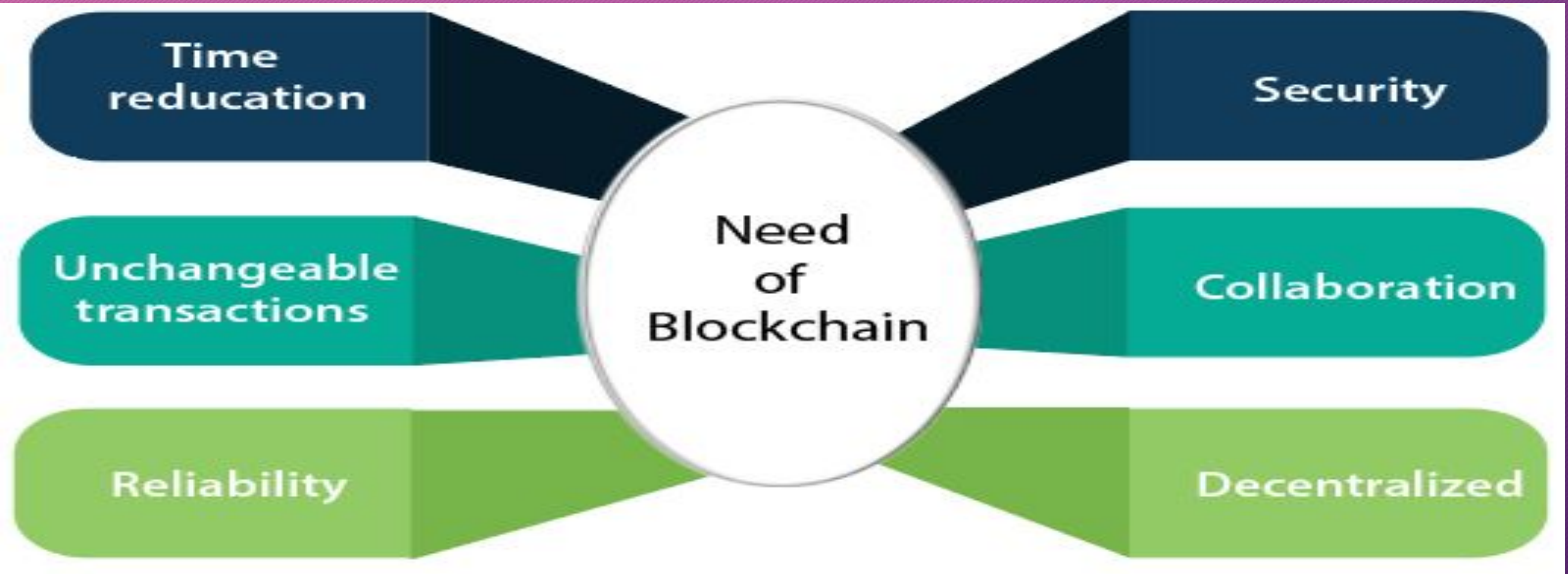
Previous hash: 1Z8F

Hash: 3H4Q

Previous hash: 6BQ1

Uh thats not right??

Why we need blockchain?



Continue…

Time reduction: In the financial industry, blockchain can allow the quicker settlement of trades. It does not take a lengthy process for verification, settlement, and clearance. It is because of a single version of agreed-upon data available between all stakeholders.

Unchangeable transactions: Blockchain register transactions in a chronological order which certifies the unalterability of all operations, means when a new block is added to the chain of ledgers, it cannot be removed or modified.

Reliability: Blockchain certifies and verifies the identities of each interested parties. This removes double records, reducing rates and accelerates transactions.

Continue...

Security: Blockchain uses very advanced cryptography to make sure that the information is locked inside the blockchain. It uses Distributed Ledger Technology where each party holds a copy of the original chain, so the system remains operative, even the large number of other nodes fall.

Collaboration: It allows each party to transact directly with each other without requiring a third-party intermediary.

Decentralized: It is decentralized because there is no central authority supervising anything. There are standards rules on how every node exchanges the blockchain information. This method ensures that all transactions are validated, and all valid transactions are added one by one.

Types of Blockchain

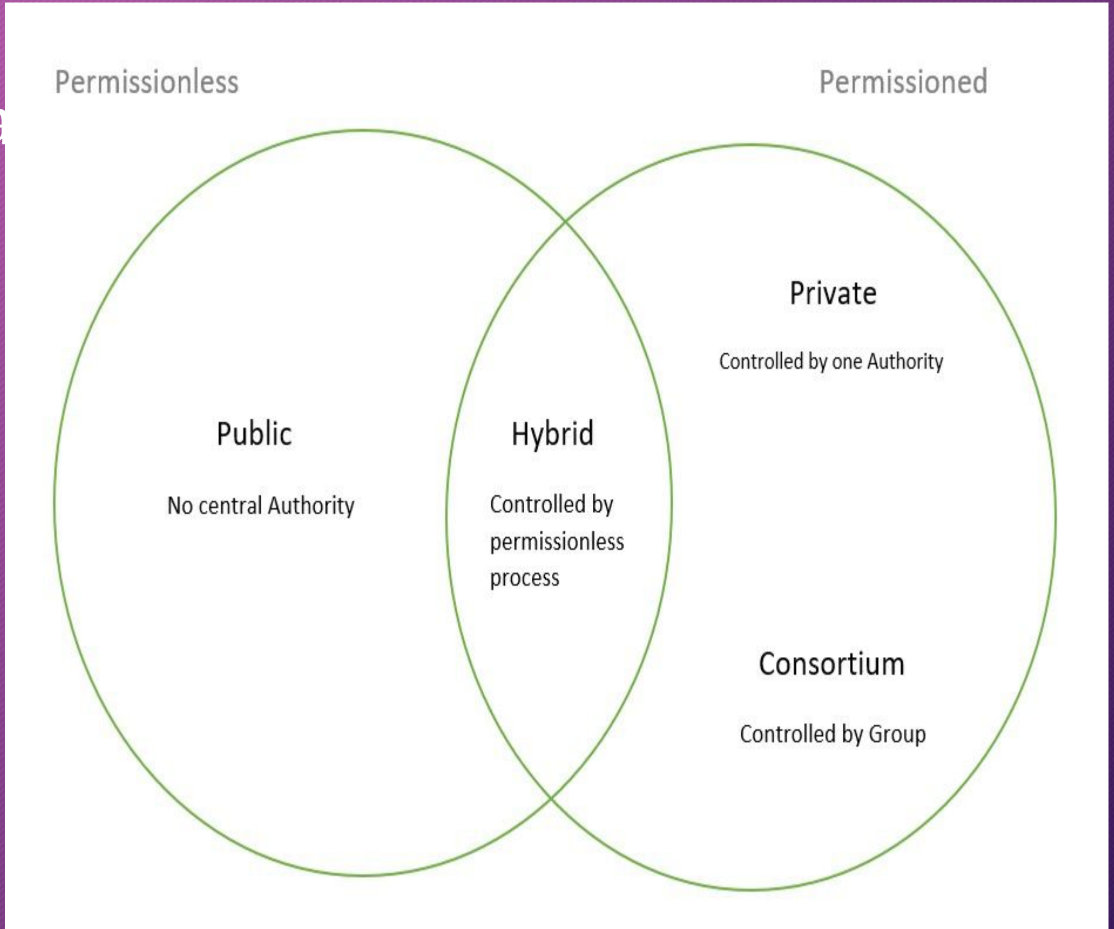
There are 3 types of Blockchain

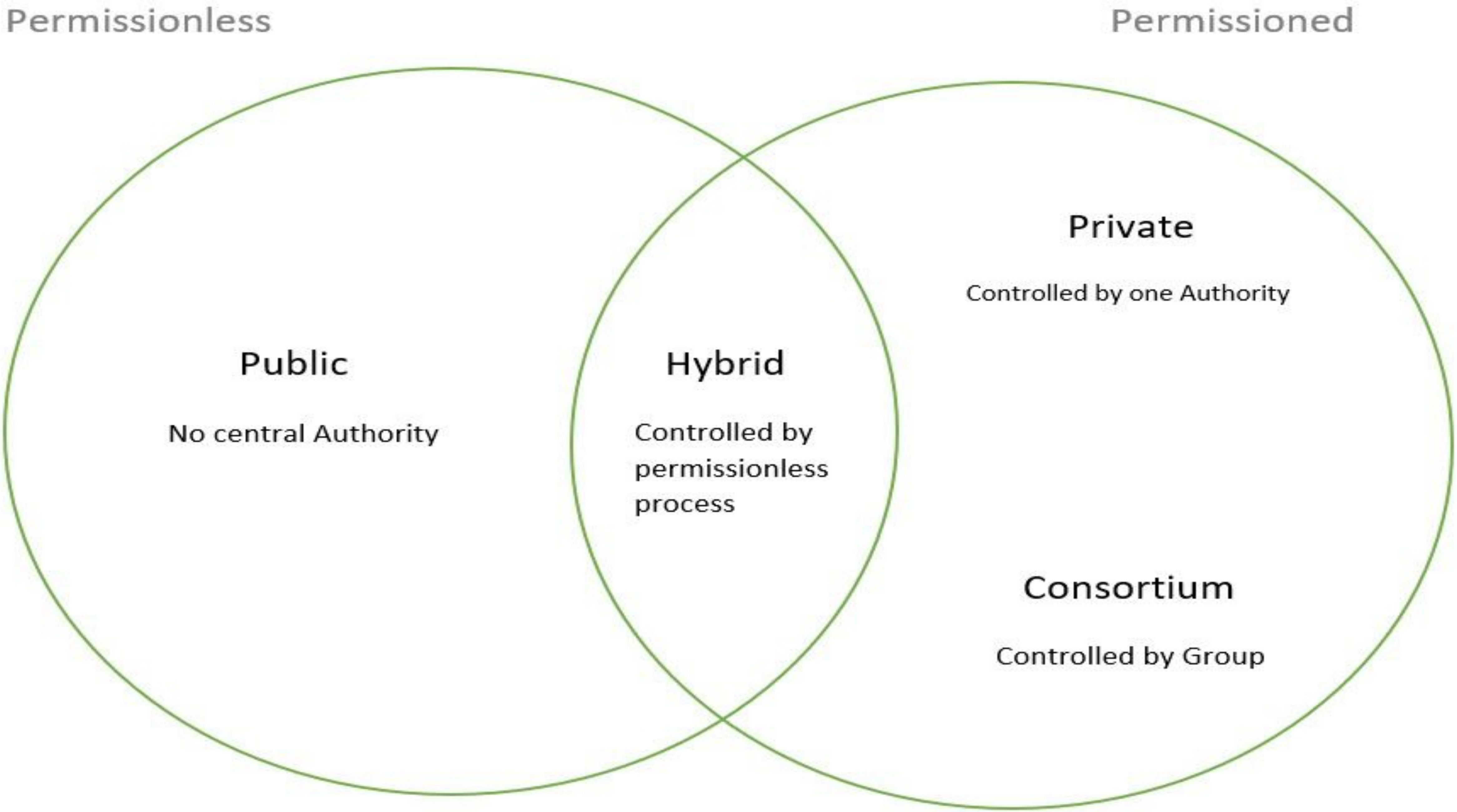
Public blockchain

Private blockchain or permissioned blockcha

Hybrid blockchain

Federated or consortium blockchain





Permissionless

Permissioned

Public

No central Authority

Hybrid

Controlled by permissionless process

Private

Controlled by one Authority

Consortium

Controlled by Group

1. Public Blockchain

These blockchains are completely open to following the idea of decentralization. They don’t have any restrictions, anyone having a computer and internet can participate in the network.

As the name is public this blockchain is open to the public, which means it is not owned by anyone.

Anyone having internet and a computer with good hardware can participate in this public blockchain.

All the computer in the network hold the copy of other nodes or block present in the network

In this public blockchain, we can also perform verification of transactions or records. Example: Bitcoin, Ethereum.

Advantages of public Blockchain

Trustable: There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network

Secure: This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records

Anonymous Nature: It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity in order to participate.

Decentralized: There is no single platform that maintains the network, instead every user has a copy of the ledger.

Disadvantages of public blockchain

Processing: The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
Energy Consumption: Proof of work is high energy-consuming. It requires good computer hardware to participate in the network
Acceptance: No central authority is there so governments are facing the issue to implement the technology faster.

2. Private Blockchain

These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.
These are not as open as a public blockchain.
They are open to some authorized users only.
These blockchains are operated in a closed network.
In this few people are allowed to participate in a network within a company/organization.
Example: Hyperledger, Corda.

Advantage of private blockchain

Speed: The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.

Scalability: We can modify the scalability. The size of the network can be decided manually.

Privacy: It has increased the level of privacy for confidentiality reasons as the businesses required.

Balanced: It is more balanced as only some user has the access to the transaction which improves the performance of the network.

Disadvantages of private blockchain

Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.
Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.

3. Hybrid Blockchain

It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

It is a combination of both public and private blockchain.

Permission-based and permissionless systems are used.

User access information via smart contracts

Even a primary entity owns a hybrid blockchain it cannot alter the transaction

Example: Ripple network and XRP token.

Advantages of Hybrid blockchain

Ecosystem: Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don’t have access to the network

Cost: Transactions are cheap as only a few nodes verify the transaction. All the nodes don’t carry the verification hence less computational cost.

Architecture: It is highly customizable and still maintains integrity, security, and transparency.

Operations: It can choose the participants in the blockchain and decide which transaction can be made public.

Disadvantages of Hybrid Blockchain

Efficiency: Not everyone is in the position to implement a hybrid Blockchain. The organization also faces some difficulty in terms of efficiency in maintenance.

Transparency: There is a possibility that someone can hide information from the user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.

Ecosystem: Due to its closed ecosystem this blockchain lacks the incentives for network participation.

4. Consortium Blockchain

It is a creative approach that solves the needs of the organization.
This blockchain validates the transaction and also initiates or receives transactions.
Also known as Federated Blockchain.
This is an innovative method to solve the organization’s needs.
Some part is public and some part is private.
In this type, more than one organization manages the blockchain.
Example: Tendermint and Multichain

Advantages of consortium blockchain

Speed: A limited number of users make verification fast. The high speed makes this more usable for organizations.

Authority: Multiple organizations can take part and make it decentralized at every level. Decentralized authority, makes it more secure.

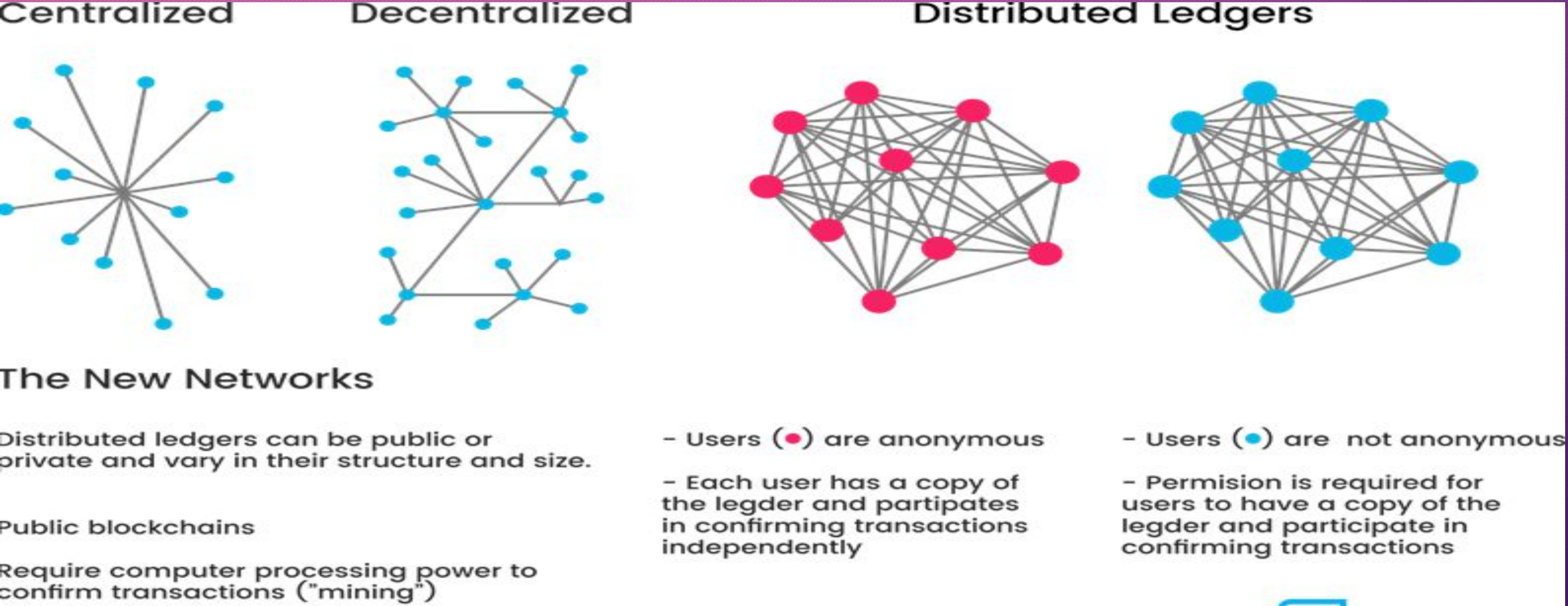
Privacy: The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.

Flexible: There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.

Disadvantage of consortium blockchain

Approval: All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.
Transparency: It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.
Vulnerability: If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain

Network Types



The New Networks

Distributed ledgers can be public or private and vary in their structure and size.

Public blockchains

Require computer processing power to confirm transactions ("mining")

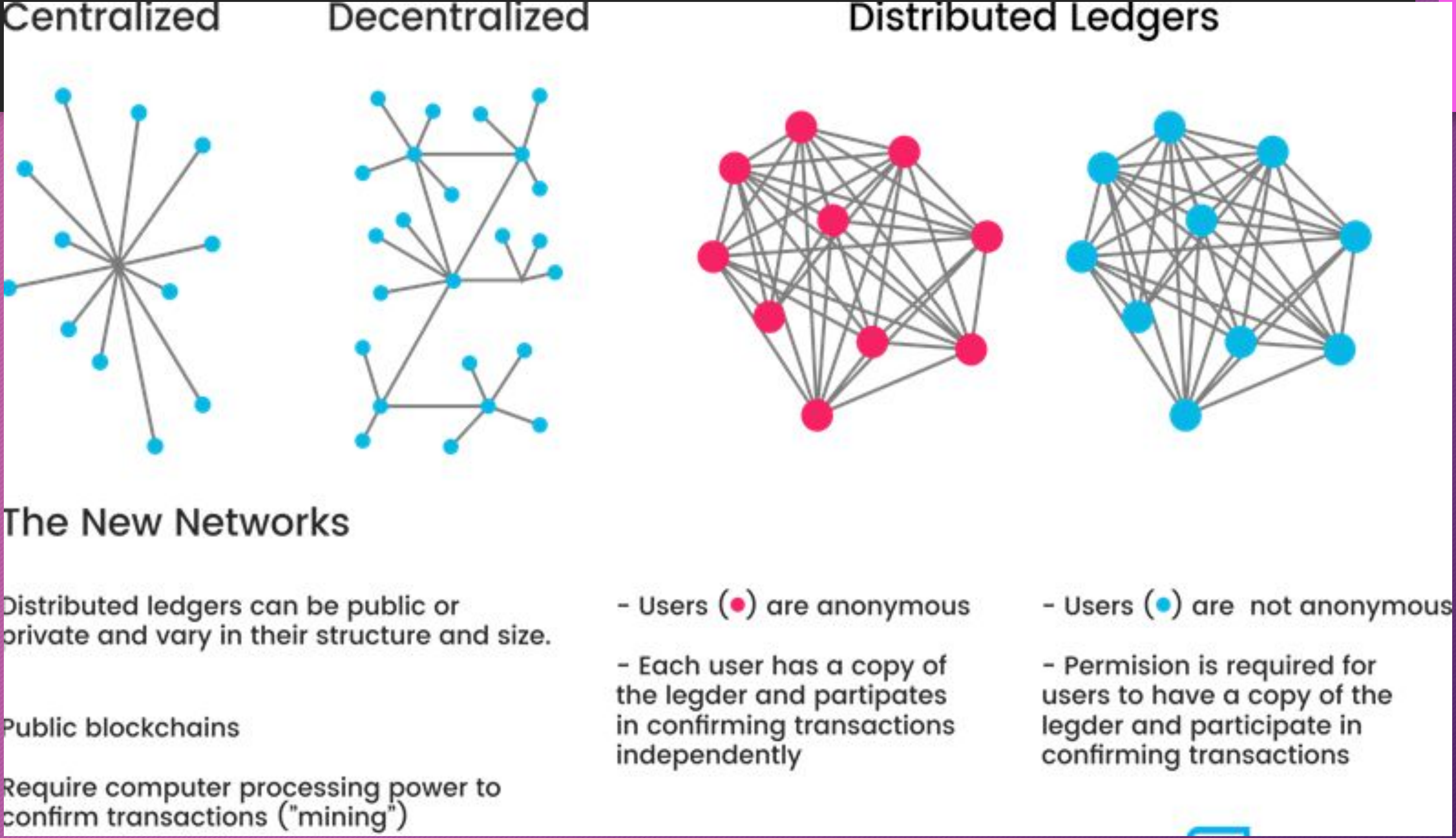
- Users (●) are anonymous

- Each user has a copy of the legder and partipates in confirming transactions independently

- Users (●) are not anonymous

- Permision is required for users to have a copy of the legder and participate in confirming transactions

Network Types



Centralized

Decentralized

Distributed Ledgers

The New Networks

Distributed ledgers can be public or private and vary in their structure and size.

Public blockchains

Require computer processing power to confirm transactions ("mining")

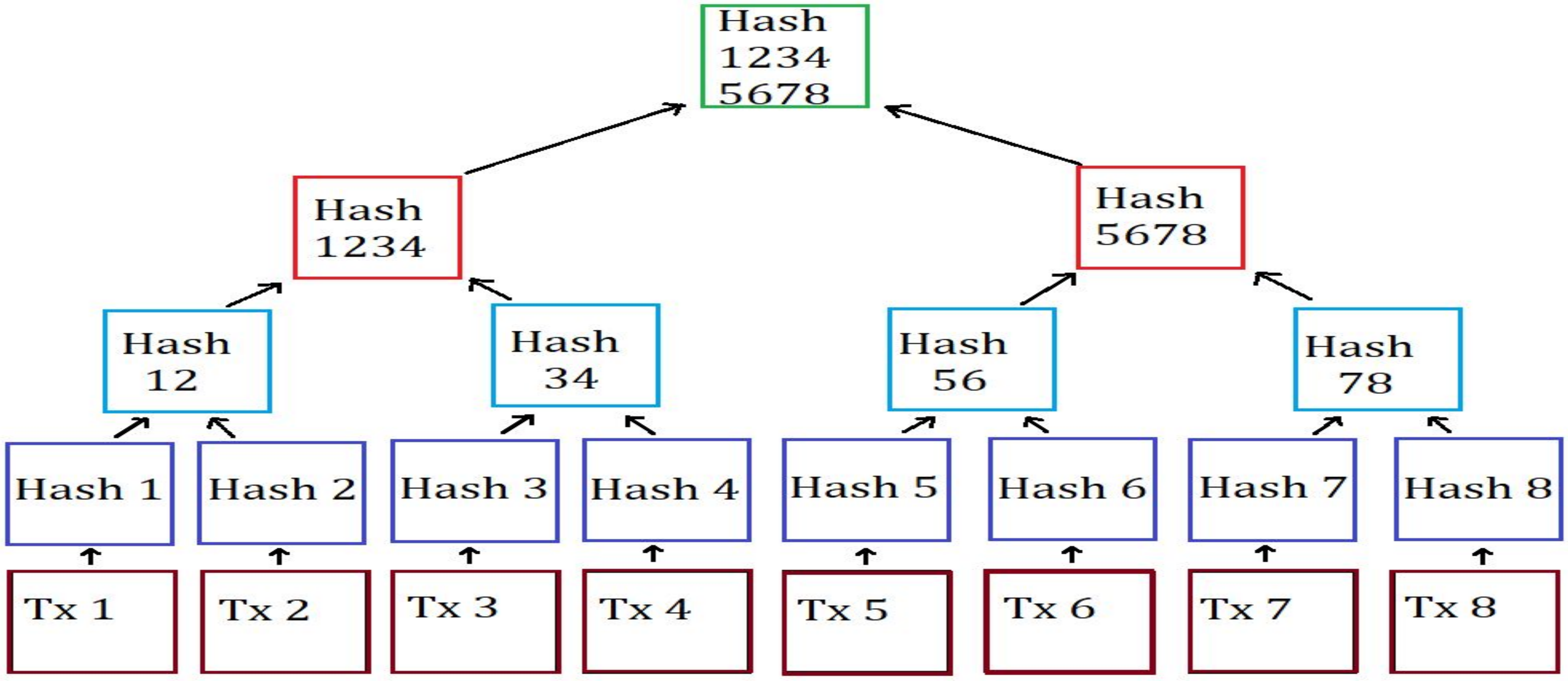
- Users (●) are anonymous

- Each user has a copy of the legder and partipates in confirming transactions independently

- Users (●) are not anonymous

- Permision is required for users to have a copy of the legder and participate in confirming transactions

Merkel Tree


Hash 1234 5678
Hash 1234
Hash 5678
Hash 12
Hash 34
Hash 56
Hash 78
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Hash 7
Hash 8
Tx 1
Tx 2
Tx 3
Tx 4
Tx 5
Tx 6
Tx 7
Tx 8

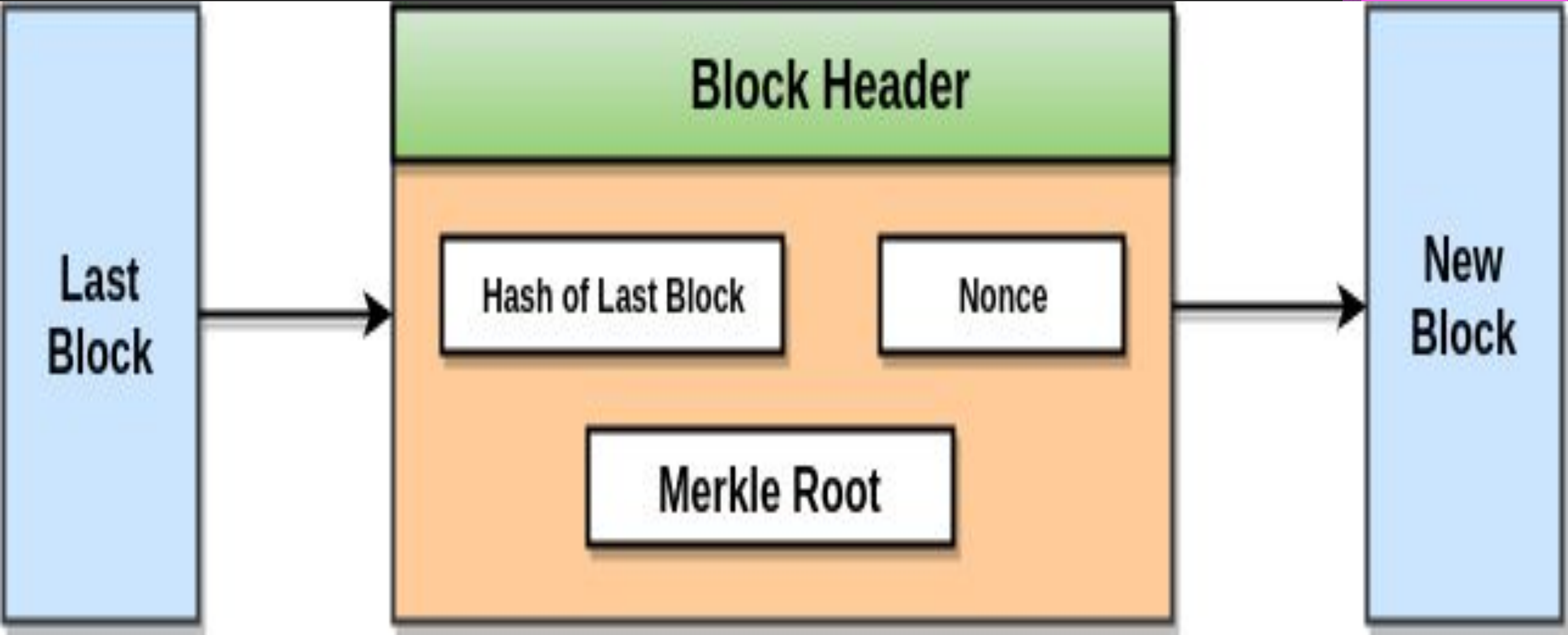
Continue…

Merkle tree is a fundamental part of blockchain technology. It is a mathematical **data structure** composed of hashes of different blocks of data, and which serves as a summary of all the transactions in a block.

It also allows for efficient and secure verification of content in a large body of data. It also helps to verify the consistency and content of the data.

Both Bitcoin and Ethereum use Merkle Trees structure. Merkle Tree is also known as **Hash Tree**.

Continue…


Last Block
Block Header
Hash of Last Block
Nonce
Merkle Root
New Block

How Merkle tree works?

A Merkle tree stores all the transactions in a block by producing a digital fingerprint of the entire set of transactions.
 It allows the user to verify whether a transaction can be included in a block or not.
Merkle Root is stored in the **block header**.
 The block header is the part of the bitcoin block which gets hash in the process of mining. It contains the hash of the last block, a Nonce, and the Root Hash of all the transactions in the current block in a Merkle Tree.
So having the Merkle root in block header makes the transaction **tamper-proof**. As this Root Hash includes the hashes of all the transactions within the block, these transactions may result in saving the disk space.

Continue…

The Merkle Tree maintains the **integrity** of the data.
If any single detail of transactions or order of the transaction's changes, then these changes reflected in the hash of that transaction.
This change would cascade up the Merkle Tree to the Merkle Root, changing the value of the Merkle root and thus invalidating the block.
So everyone can see that Merkle tree allows for a quick and simple test of whether a specific transaction is included in the set or not.

Benefits of Merkle Tree

It provides a means to maintain the integrity and validity of data.
It helps in saving the memory or disk space as the proofs, computationally easy and fast.
Their proofs and management require tiny amounts of information to be transmitted across networks.

Proof of work

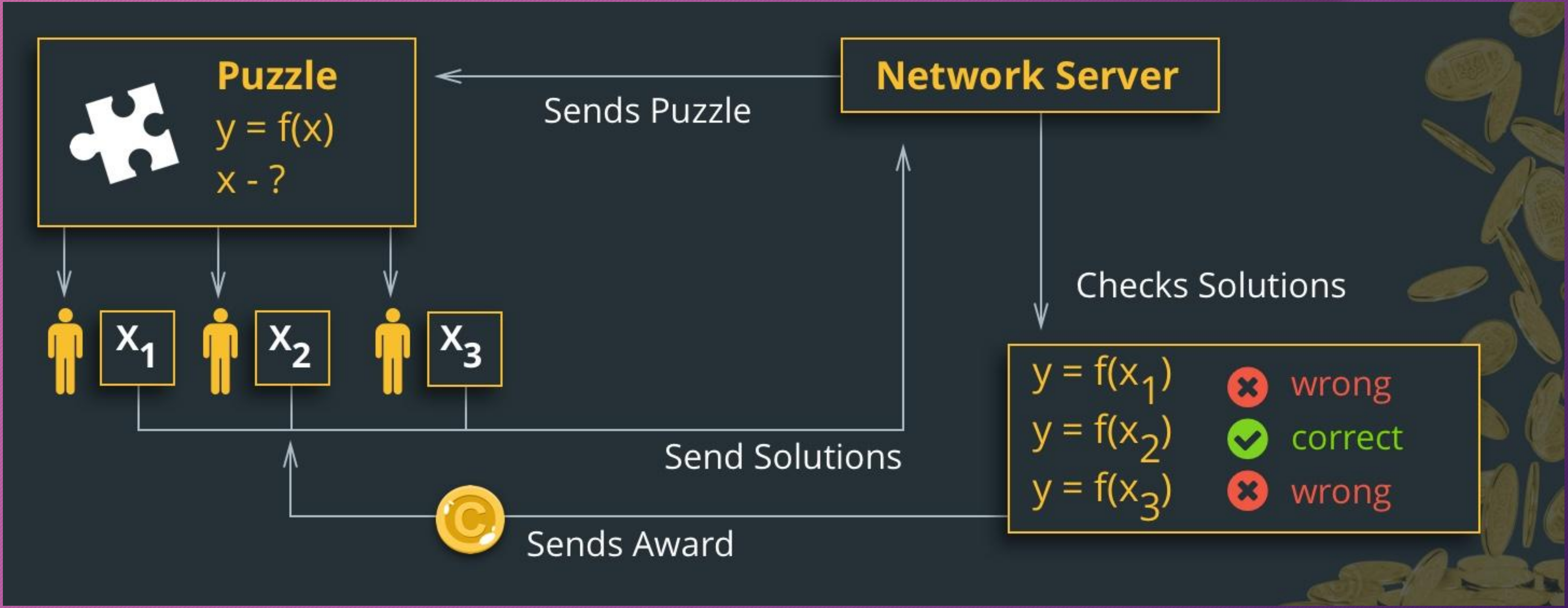
Proof-of-Work, or PoW, is the original consensus algorithm in a Blockchain network.

In Blockchain, this algorithm is used to confirm transactions and produce new blocks to the chain.

With PoW, miners compete against each other to complete transactions on the network and get rewarded.

The main working principles are a complicated mathematical puzzle and a possibility to easily prove the solution.

Proof of work – Working Principle



Proof of Work – Why?

Defense from DoS attacks. PoW imposes some limits on actions in the network. They need a lot of efforts to be executed. Efficient attack requires a lot of computational power and a lot of time to do the calculations. Therefore, the attack is possible but kind of useless since the costs are too high.

Mining possibilities. It doesn’t matter how much money you have in your wallet. What matters is to have large computational power to solve the puzzles and form new blocks. Thus, the holders of huge amounts of money are not in charge of making decisions for the entire network.

Proof of work – Applications

The most famous application of PoW is Bitcoin.
It was Bitcoin that laid the foundation for this type of consensus.
The puzzle is Hashcash.
This algorithm allows changing the complexity of a puzzle based on the total power of the network.
The average time of block formation is 10 minutes.
Bitcoin-based cryptocurrencies, such as Litecoin, have the similar system.
Ethereum also uses PoW.

Proof of work – Disadvantages

Huge expenditures. Mining requires highly specialized computer hardware to run the complicated algorithms. The costs are unmanageable Mining is becoming available only for special mining pools. These specialized machines consume large amounts of power to run that increase costs.

“Uselessness” of computations. Miners do a lot of work to generate blocks and consume a lot of power. However, their calculations are not applicable anywhere else. They guarantee the security of the network but cannot be applied to business, science or any other field.

51% attack

A 51 percent attack, or majority attack, is a case when a user or a group of users control the majority of mining power.

The attackers get enough power to control most events in the network.

They can monopolize generating new blocks and receive rewards since they're able to prevent other miners from completing blocks.

A 51% attack happens when a malicious user in a network acquires control of a given blockchain's mining capabilities. It implies that the attackers will have more than 50% mining power and can mine faster than everyone else.

The attackers can stop the confirmation and order of new transactions. The malicious agents can then rewrite parts of a blockchain and reverse the transactions. A 51% attack usually bypasses the blockchain's security protocols. The attack's impact can be mild or severe, depending on the mining power of the attacker. The hash power is more critical in the attacks. If the attacker possesses a higher percentage, the likelihood of attacking the system is also high. The damages caused by the attack are also dependent on the same factor.

Continue…

The 51% attack is an attack on the blockchain, where a group controls more than 50% of the hashing power—the computing that solves the cryptographic puzzle— of the network. This group then introduces an altered blockchain to the network at a very specific point in the blockchain, which is theoretically accepted by the network because the attackers would own most of it.1
Changing historical blocks—transactions locked in before the start of the attack—would be extremely difficult even in the event of a 51% attack. The further back the transactions are, the more difficult it is to change them.

Applications

Business Networks

Participants are customers, suppliers, banks.

Cross geography and regulatory boundary.

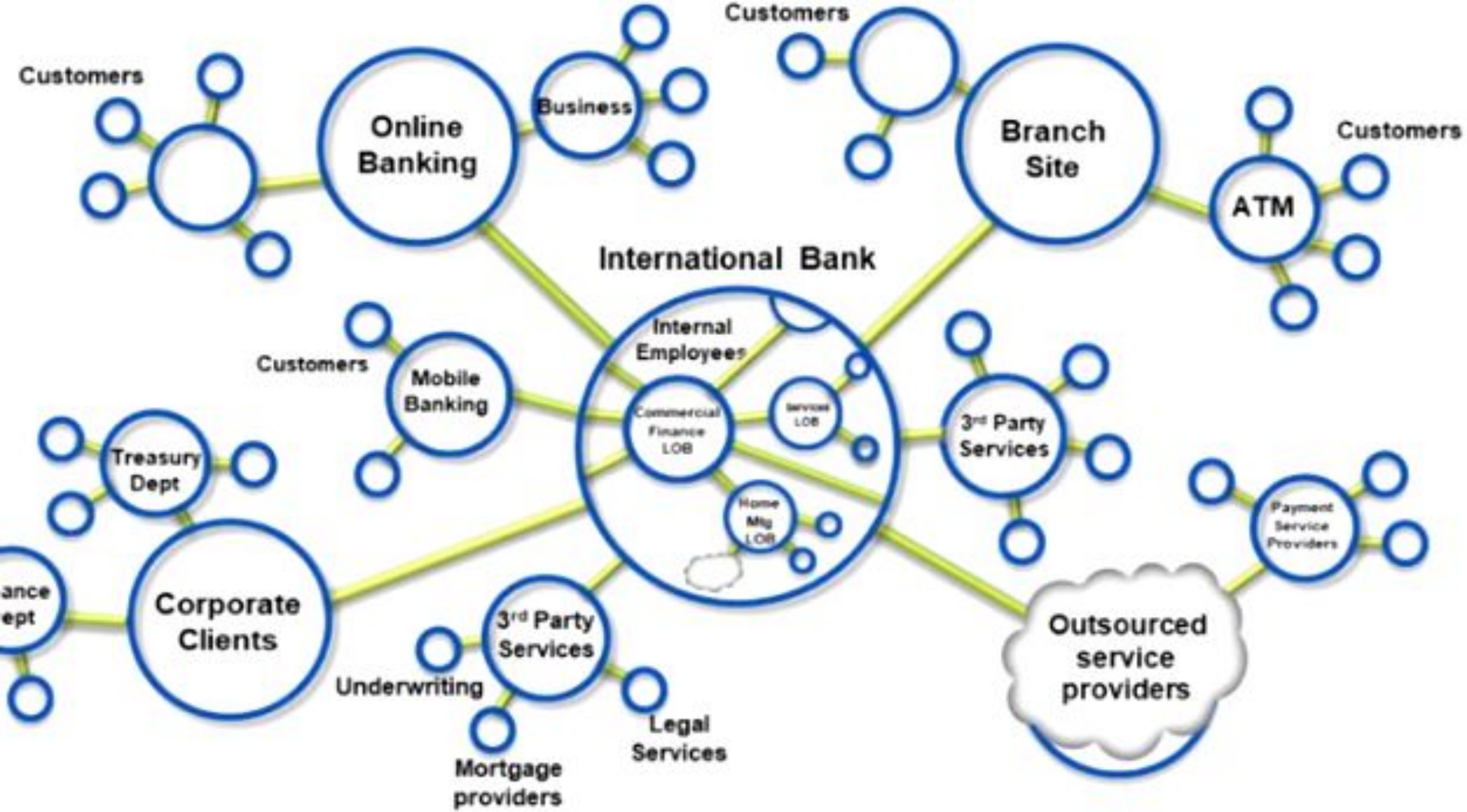
Wealth

Is generated by the flow of goods and services across business network in transactions and contracts.

Business Networks

Public (markets, auctions)

Private (supply chain financing, bonds)


Customers
Online Banking
Business
Customers
Branch Site
Customers
ATM
International Bank
Internal Employees
Customers
Mobile Banking
Commercial Finance LOB
Services LOB
Home Mtg LOB
3rd Party Services
Treasury Dept
Payment Service Providers
ance
ept
Corporate Clients
3rd Party Services
Underwriting
Legal Services
Mortgage providers
Outsourced service providers

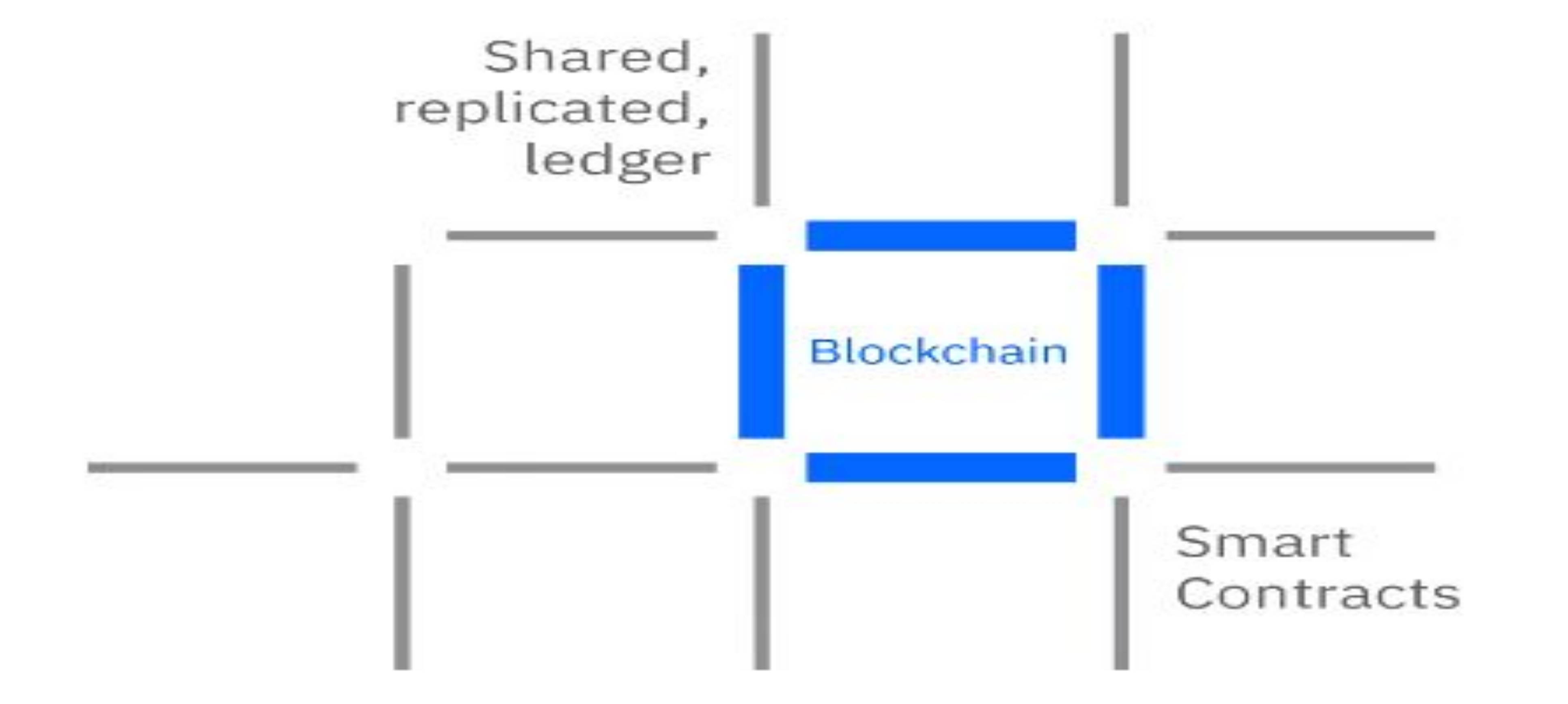
Assets

- Tangible
  - House, etc.
- Intangible
  - Financial – bond
  - Intellectual – patents
  - Digital – data
- Cash
  - Anonymous asset - No records, Difficult to track

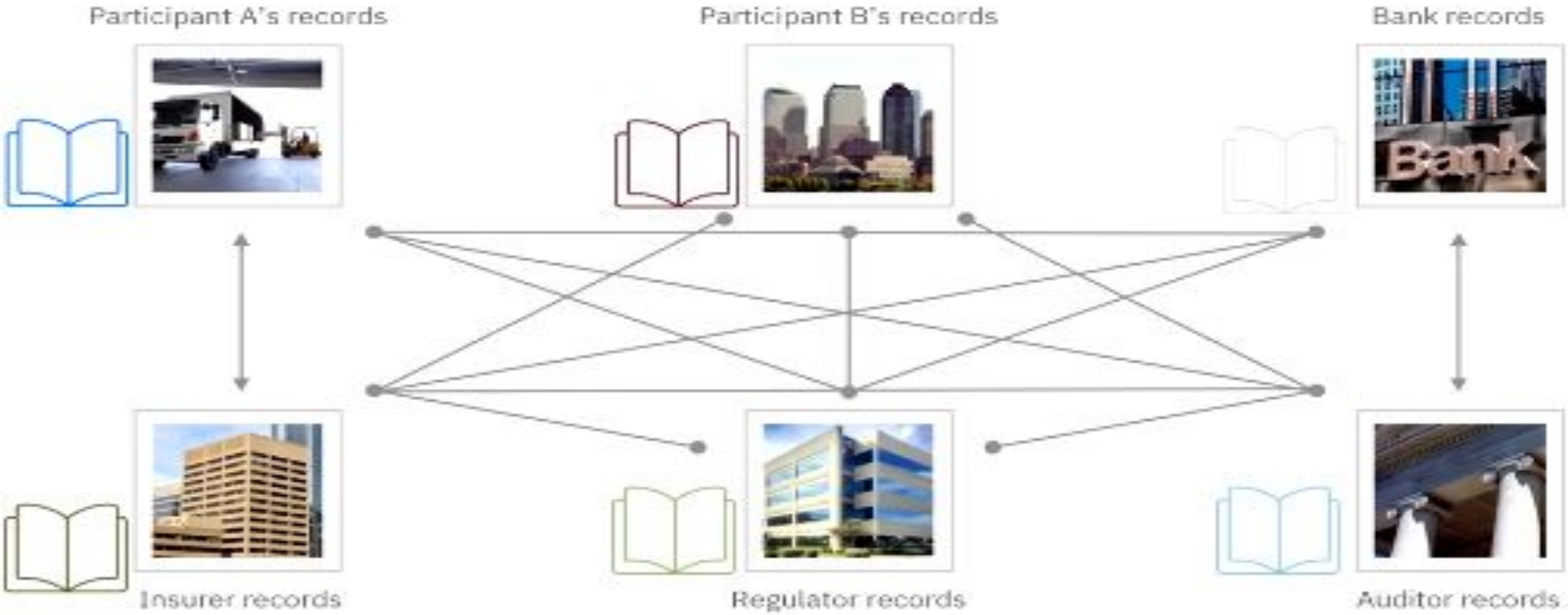
Ledgers, Transaction and Contract

Ledger – An important log of all transactions – Describes input/output of the business.
Transaction – An asset transfer between participants.
Contract – The terms and conditions for a transaction.

Business context of Blockchain

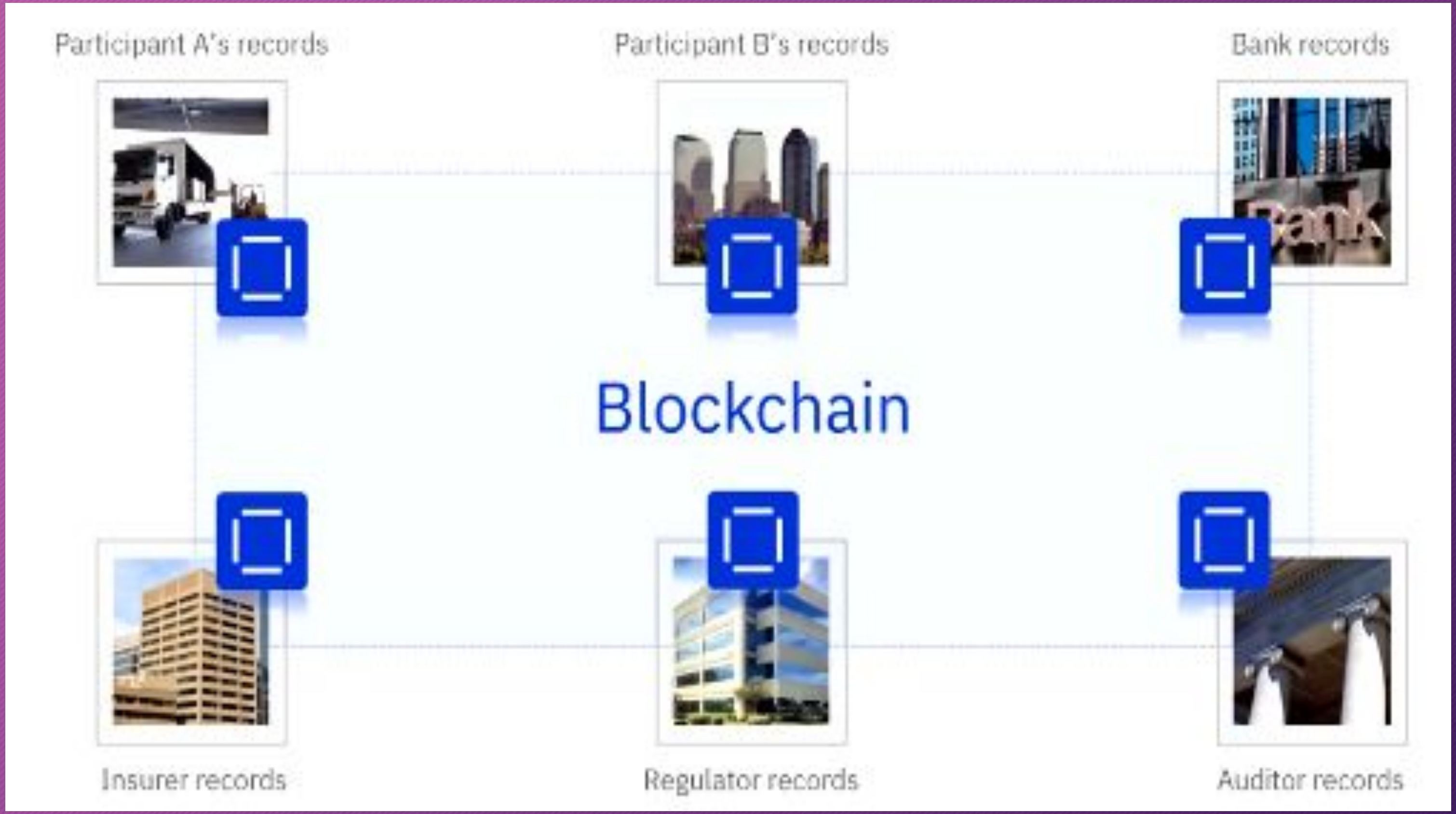


Traditional Business Network



Blockchain Business Network

Consensus – Mutual Agreement
Provenance – Transparency of history
Immutability – Tamper free
Finality – Once committed, cannot be reversed.



Requirements of blockchain for business

Assets – Participants decide which assets to share.

Identity – Participants know who they are dealing with; information shared is need-to-know (KYC).

Endorsement – Participants give provable endorsement (instead of Proof of Work).

Smart Contracts

Smart contracts are lines of code that are stored on a blockchain and automatically execute when predetermined terms and conditions are met.

At the most basic level, they are programs that run as they’ve been set up to run by the people who developed them.

The benefits of smart contracts are most apparent in business collaborations, in which they are typically used to enforce some type of agreement so that all participants can be certain of the outcome without an intermediary’s involvement.

Ethereum uses smart contracts. Smart Contracts are implemented using the programming language called Solidity.

Active Networks

IBM is making blockchain real for business with cross industry solutions and 100s of active networks. The following are some of initiatives of IBM.

Tennet – Blockchain is used to balance the electricity grid in Germany by storing and releasing electricity to and from the batteries.

We Trade – Is a blockchain based consortium of banks based in Europe. This provides trade finance to small and medium customers.

Tradelens

Tradelens is an open, extensible platform for sharing, shipping events, messages and documents across all the actors and systems in a supply chain ecosystem.

Provides shared visibility and shared state for container shipments.

Increased speed and transparency for cross border transactions through real time access to container events.

Reduced cost and increased efficiency through paperless trade.

Food Trust for supply chain transparency

IBM Food Trust is a set of modules providing traceability to improve food transparency and efficiency.

Blockchain is used to create a trusted connection with shared value for all ecosystem participants, including end consumers.

Reduce impact of food recalls through instant access to end to end traceability data to verify history in the food network and supply chain.

Helping farmers.

Help address food born illnesses.

IBM World Wire

IBM Blockchain World Wire is an integrated network for real time clearing and settlement.

Allows banks and financial institutions to send and settle payments around the globe with finality in a matter of seconds.

Eliminates enduring challenges that have long hampered the cross border payments industry.

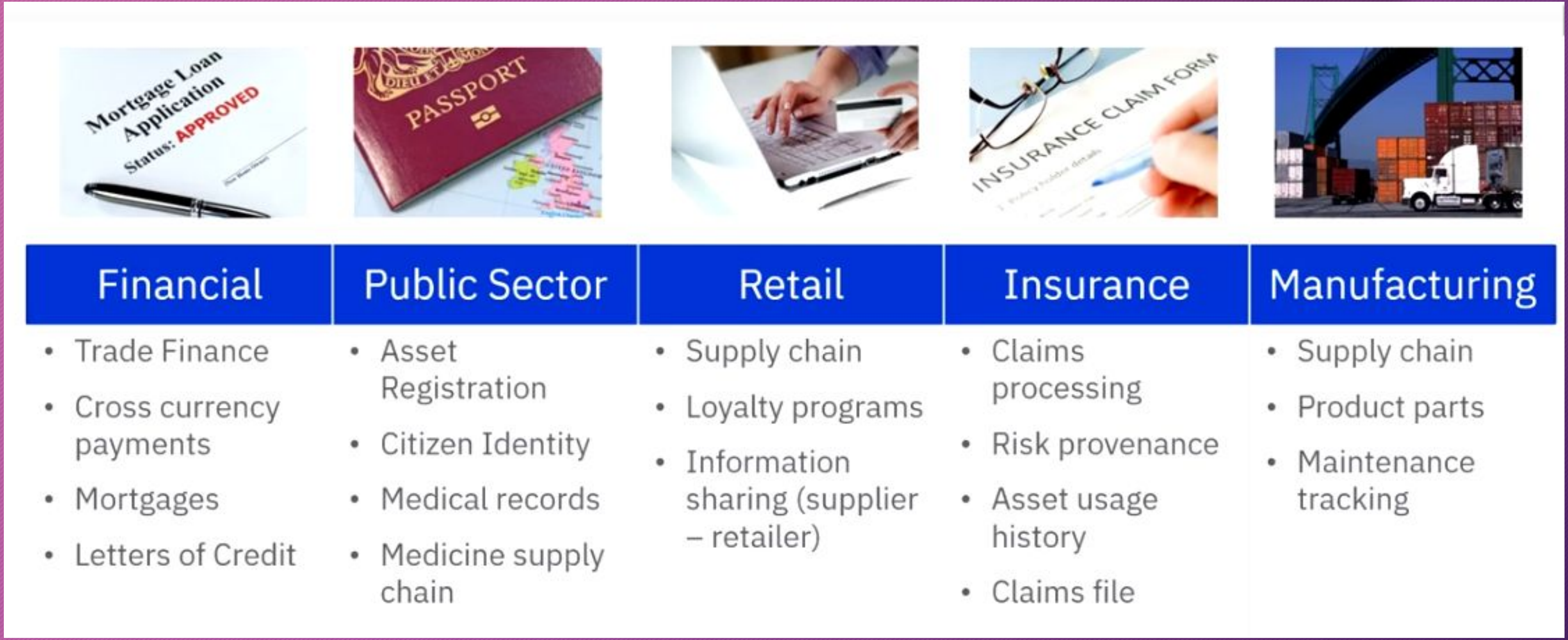
Higher visibility of transactions with reduced disputes.

Secure network with interaction and eligibility criteria as well as robust access controls.

Decentralized and trusted identity

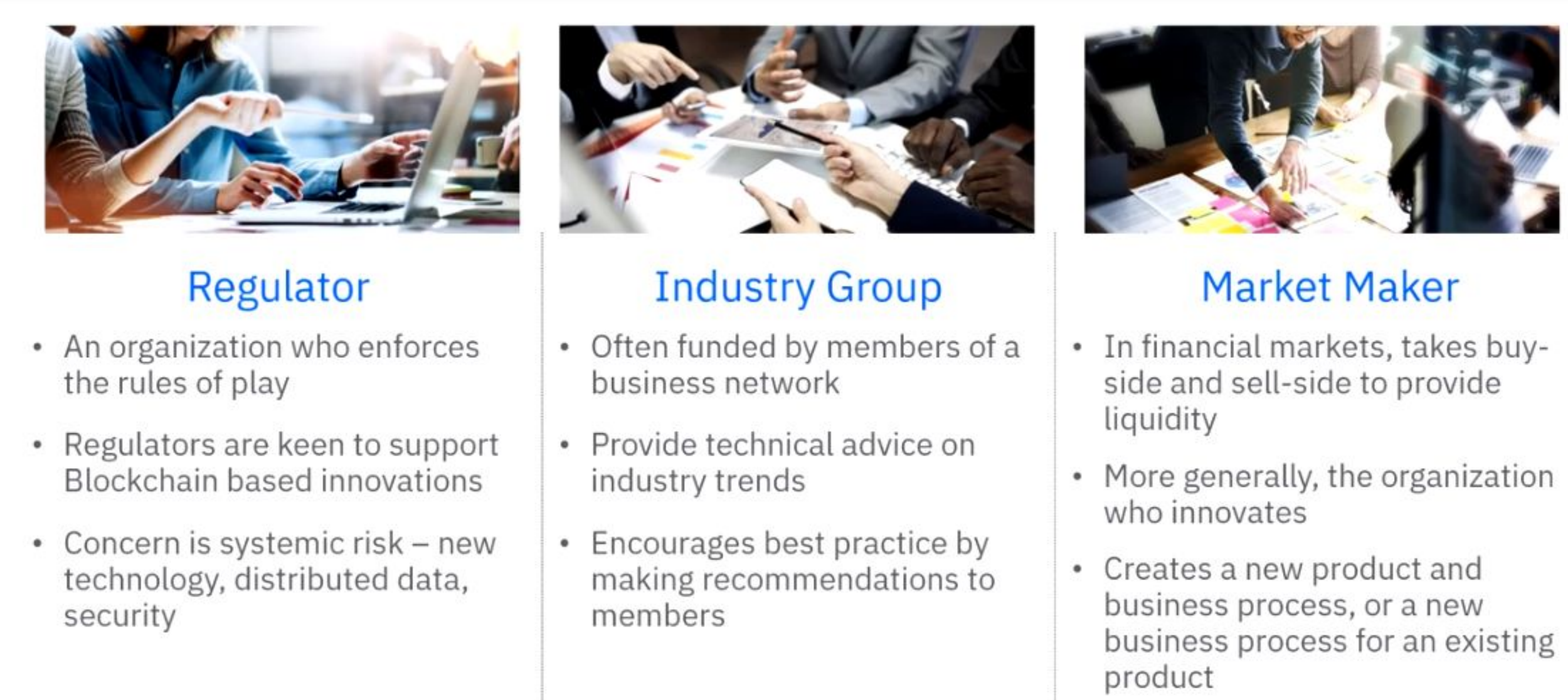
IBM initiative pushes identity to the edge of the network.
Cryptographic point to point exchange of identity.
Based on Hyperledger Indy technology.
Every person, organization and thing has a digital wallet to control the flow of their identity.

Further examples by industry



Financial	Public Sector	Retail	Insurance	Manufacturing
• Trade Finance • Cross currency payments • Mortgages • Letters of Credit	• Asset Registration • Citizen Identity • Medical records • Medicine supply chain	• Supply chain • Loyalty programs • Information sharing (supplier – retailer)	• Claims processing • Risk provenance • Asset usage history • Claims file	• Supply chain • Product parts • Maintenance tracking

Key Players for Blockchain Adoption



Regulator

- An organization who enforces the rules of play
- Regulators are keen to support Blockchain based innovations
- Concern is systemic risk – new technology, distributed data, security

Industry Group

- Often funded by members of a business network
- Provide technical advice on industry trends
- Encourages best practice by making recommendations to members

Market Maker

- In financial markets, takes buy-side and sell-side to provide liquidity
- More generally, the organization who innovates
- Creates a new product and business process, or a new business process for an existing product

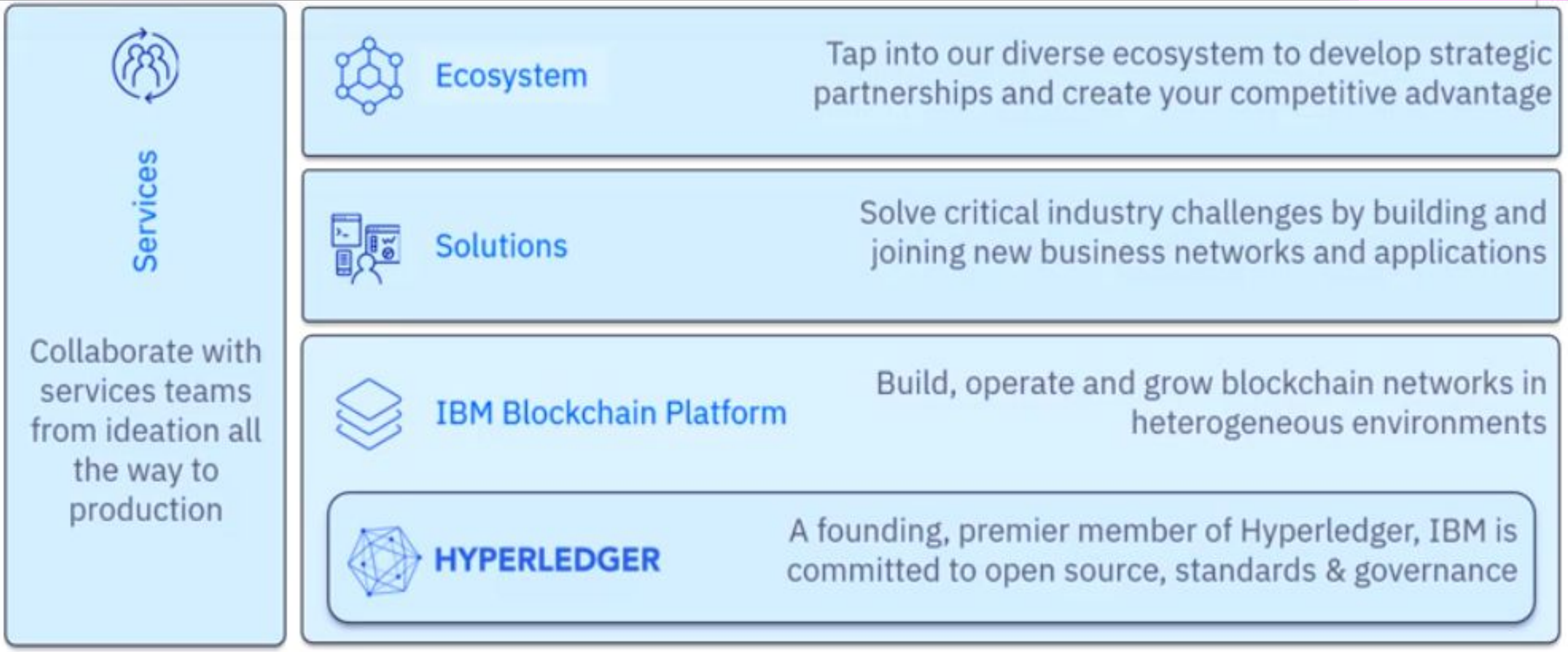
How IBM can help

Security at scale – Enterprise grade security and control on a platform where businesses and industries are reinventing themselves.

Trusted expertise – Reinventing business processes through unrivalled industry and technical knowledge as you start, accelerate and innovate your blockchain network.

Network convening power – Bringing together an expansive partner network of innovators, regulators and suppliers to establish, join or run your blockchain network.

IBM Blockchain Strategy



Services

Collaborate with services teams from ideation all the way to production

Ecosystem

Tap into our diverse ecosystem to develop strategic partnerships and create your competitive advantage

Solutions

Solve critical industry challenges by building and joining new business networks and applications

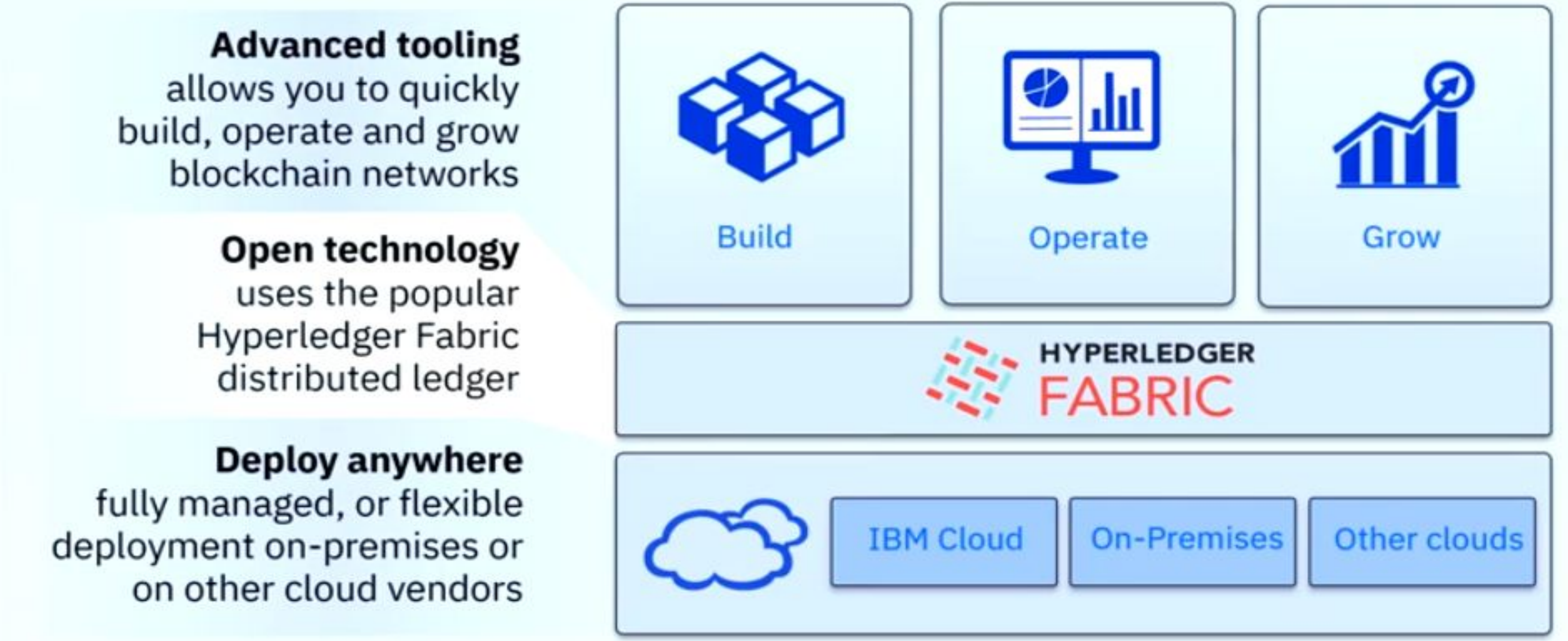
IBM Blockchain Platform

Build, operate and grow blockchain networks in heterogeneous environments

HYPERLEDGER

A founding, premier member of Hyperledger, IBM is committed to open source, standards & governance

IBM Blockchain Platform



Advanced tooling allows you to quickly build, operate and grow blockchain networks

Open technology uses the popular Hyperledger Fabric distributed ledger

Deploy anywhere fully managed, or flexible deployment on-premises or on other cloud vendors

Build

Operate

Grow

HYPERLEDGER FABRIC

IBM Cloud

On-Premises

Other clouds

Hyperledger is not

A Cryptocurrency
A Blockchain
A Company

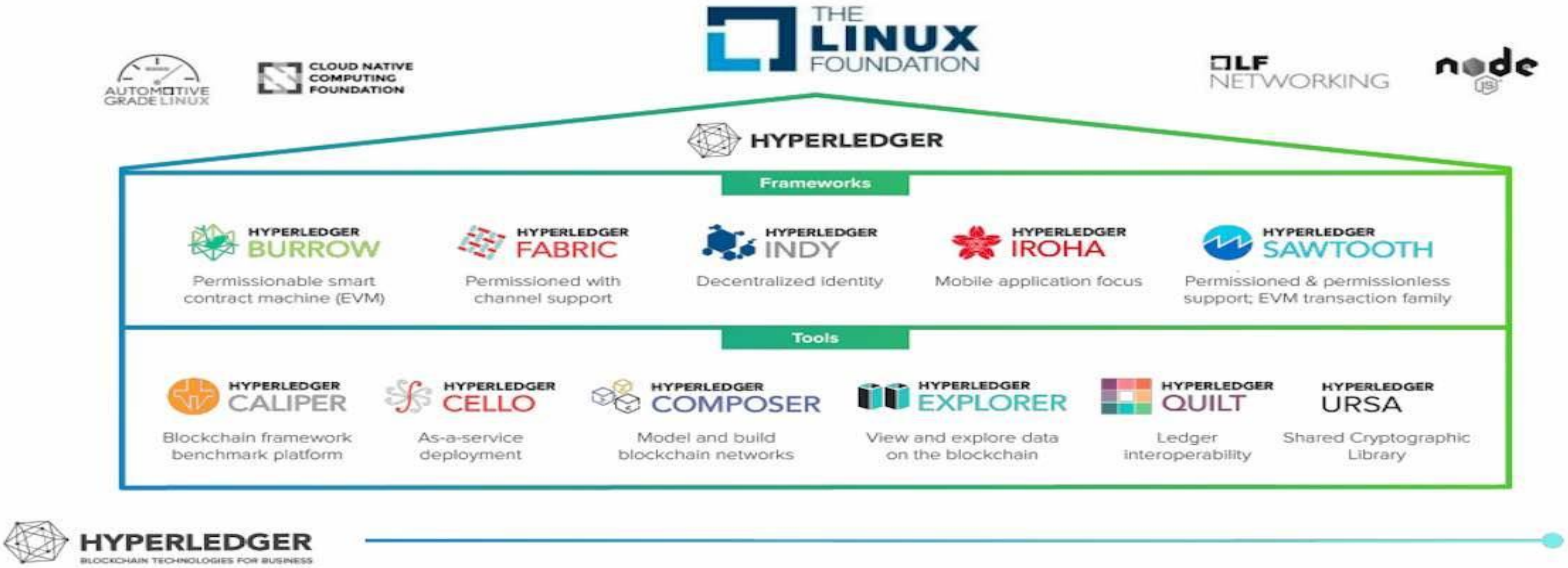
Hyperledger is

A project under Linux foundation.
An open source development project.
An open source community of communities to benefit an ecosystem of Hyperledger based solution providers and users focussed on blockchain related use cases that will work across a variety of industrial sectors.

Hyperledger

IBM Blockchain Platform is underpinned by technology from the Hyperledger project.
Hyperledger is an open source collaborative effort created to advance cross industry blockchain technologies.
The Hyperledger greenhouse includes many projects that anyone can freely use and contribute to.
Open Source
Open Standards
Open governance model.

Hyperledger Greenhouse



THE LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

CLOUD NATIVE COMPUTING FOUNDATION

LF NETWORKING

node JS

HYPERLEDGER

Frameworks

HYPERLEDGER BURROW
Permissionable smart contract machine (EVM)

HYPERLEDGER FABRIC
Permissioned with channel support

HYPERLEDGER INDY
Decentralized identity

HYPERLEDGER IROHA
Mobile application focus

HYPERLEDGER SAWTOOTH
Permissioned & permissionless support; EVM transaction family

Tools

HYPERLEDGER CALIPER
Blockchain framework benchmark platform

HYPERLEDGER CELLO
As-a-service deployment

HYPERLEDGER COMPOSER
Model and build blockchain networks

HYPERLEDGER EXPLORER
View and explore data on the blockchain

HYPERLEDGER QUILT
Ledger interoperability

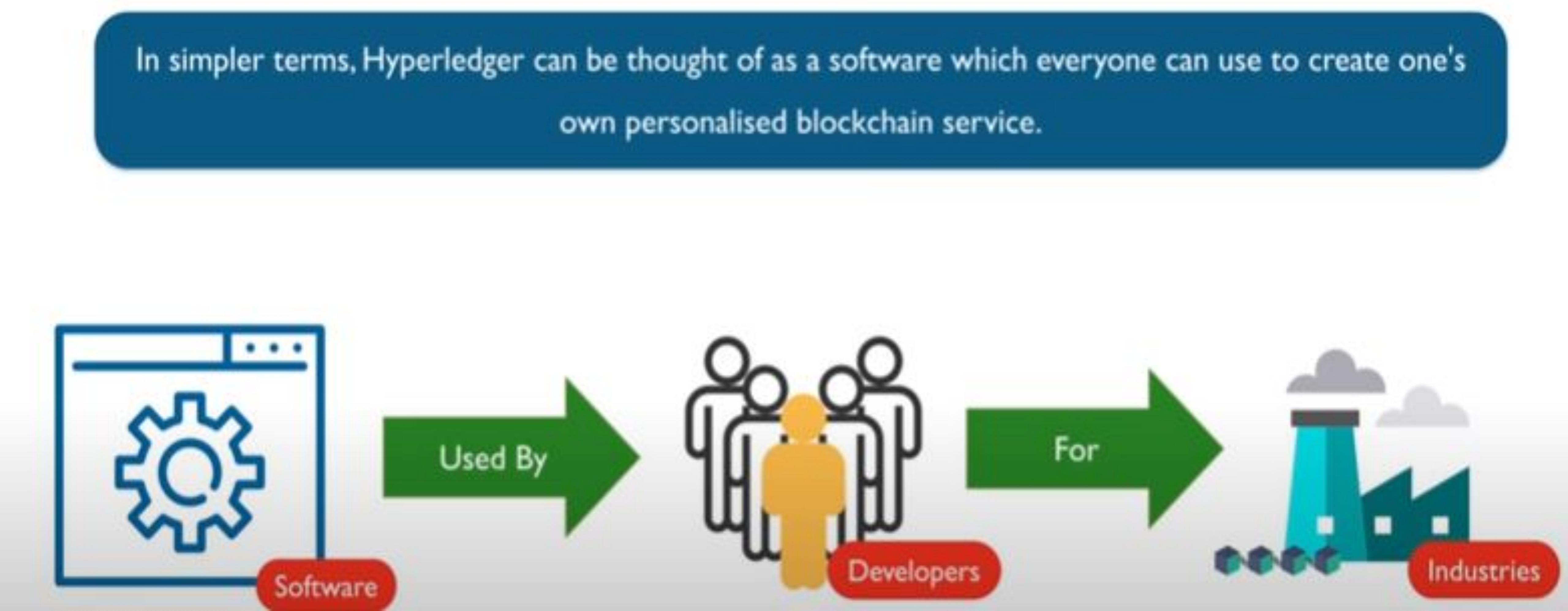
HYPERLEDGER URSA
Shared Cryptographic Library

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Proof of elapsed time

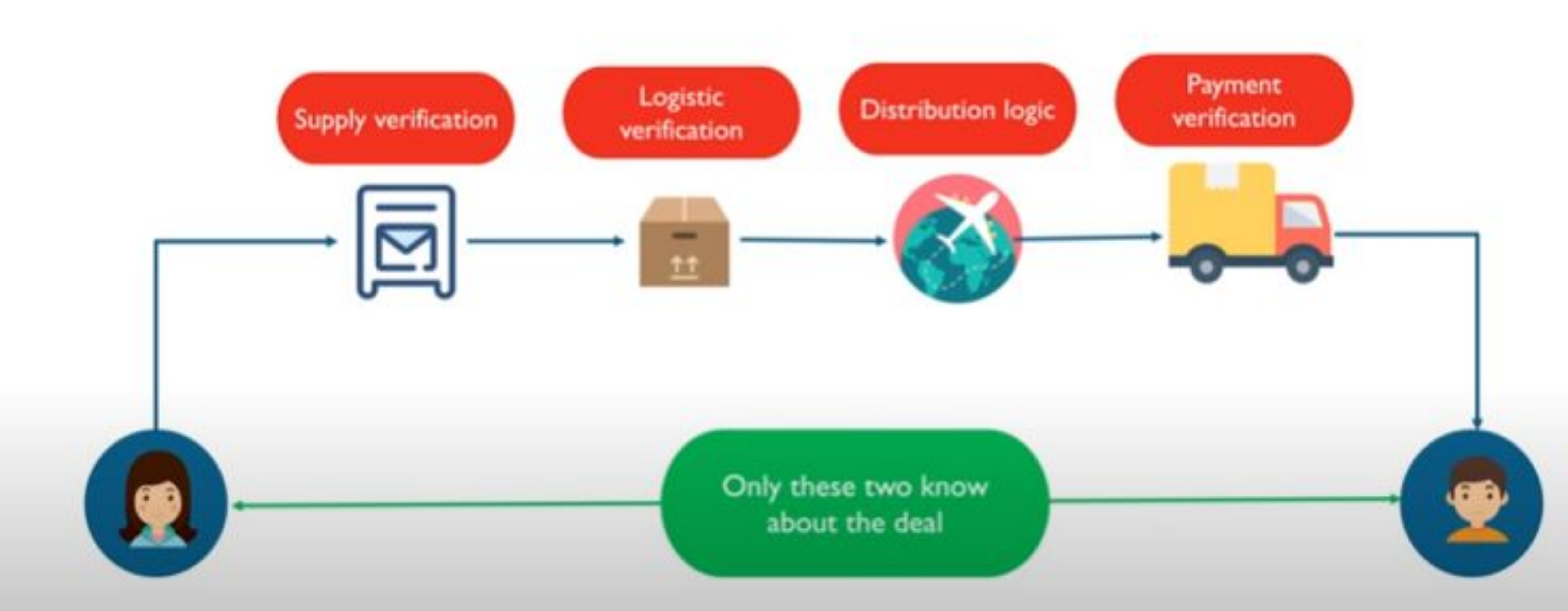
A node downloads PoET code and generates a key.
This key is verified by the existing participants.
The new node is given a timer object which is random.
Each participant in the blockchain network waits a random amount of time.
The first participant to finish waiting gets to be the leader of the new block.

Hyperledger

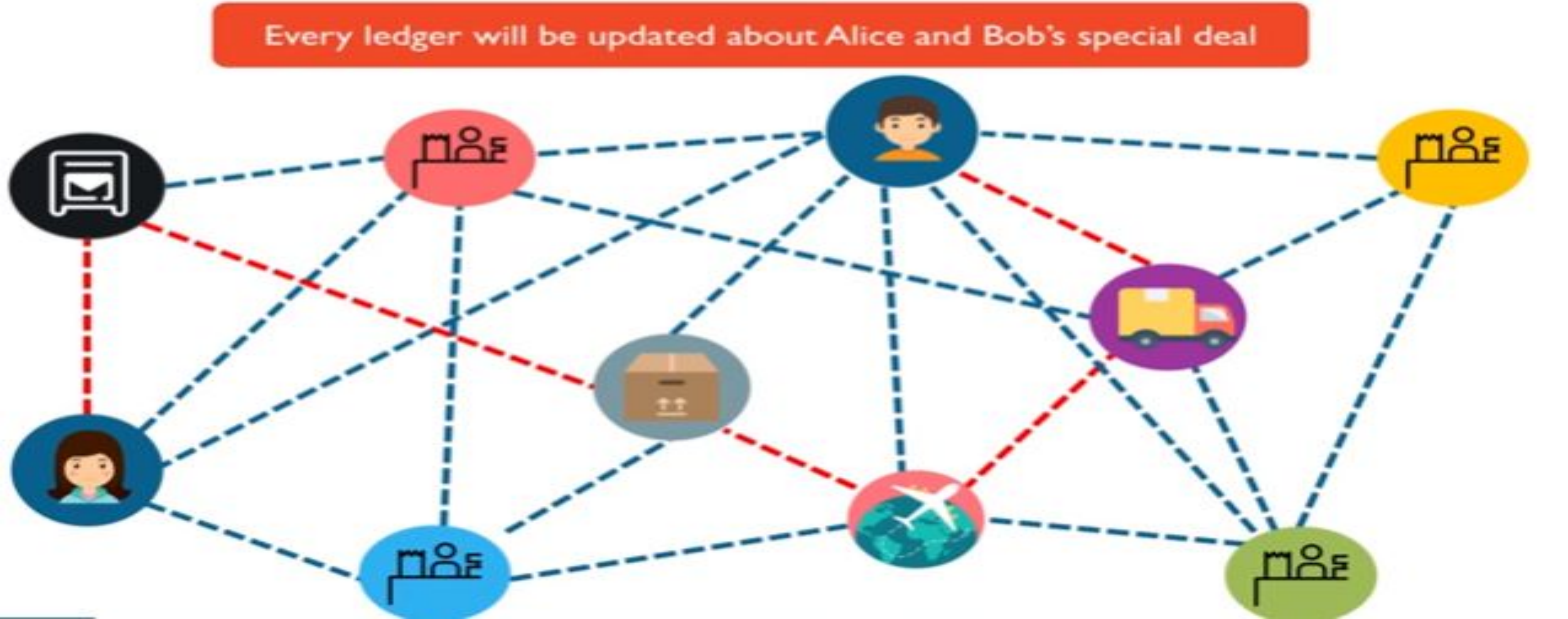


In simpler terms, Hyperledger can be thought of as a software which everyone can use to create one's own personalised blockchain service.

Example – Traditional Trading

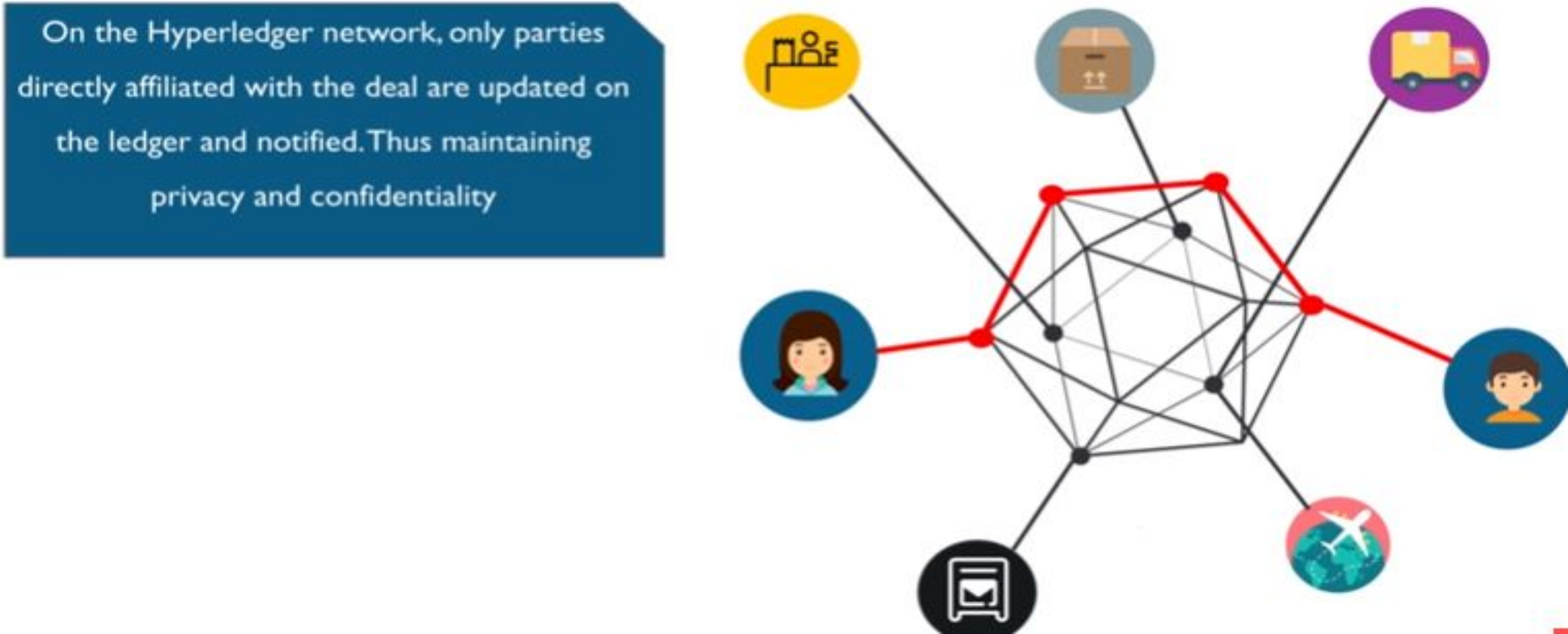


Example – Trading on public blockchain



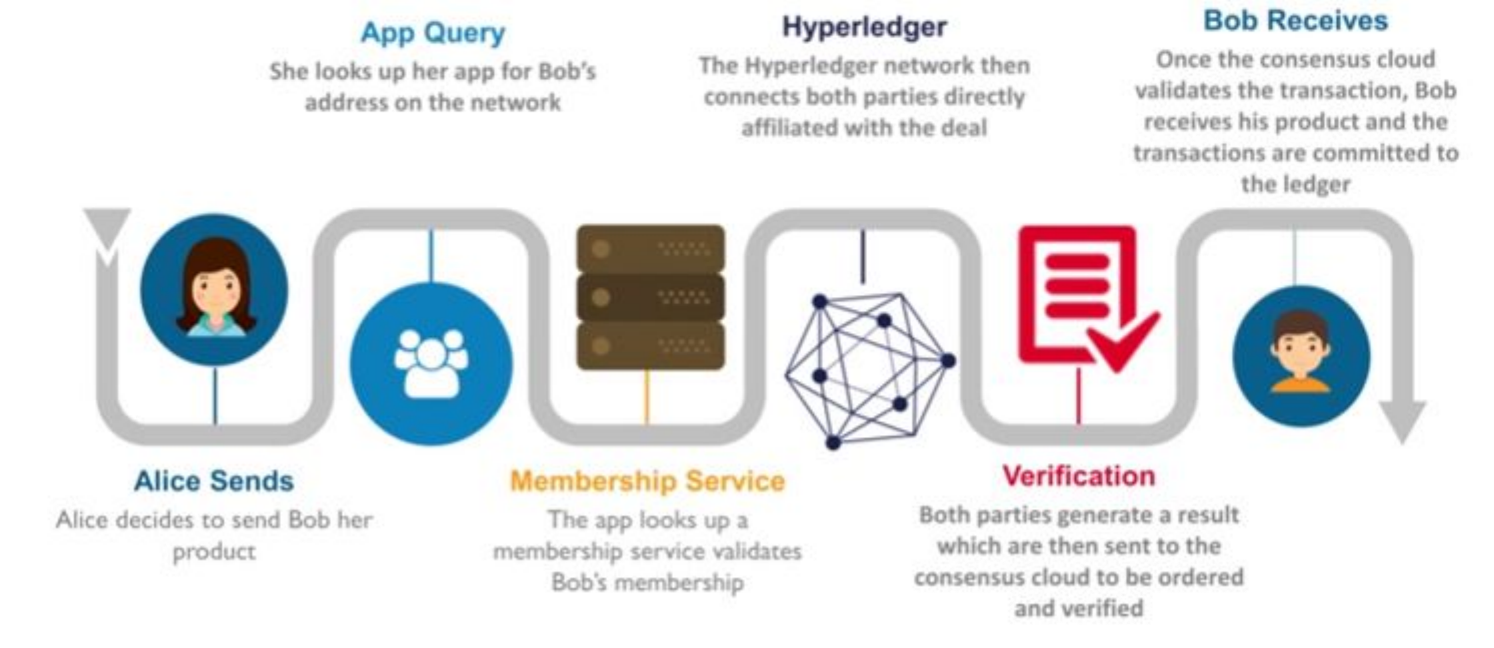
Every ledger will be updated about Alice and Bob's special deal

Example – Hyperledger



On the Hyperledger network, only parties directly affiliated with the deal are updated on the ledger and notified. Thus maintaining privacy and confidentiality

How this works



App Query

She looks up her app for Bob's address on the network

Hyperledger

The Hyperledger network then connects both parties directly affiliated with the deal

Bob Receives

Once the consensus cloud validates the transaction, Bob receives his product and the transactions are committed to the ledger

Alice Sends

Alice decides to send Bob her product

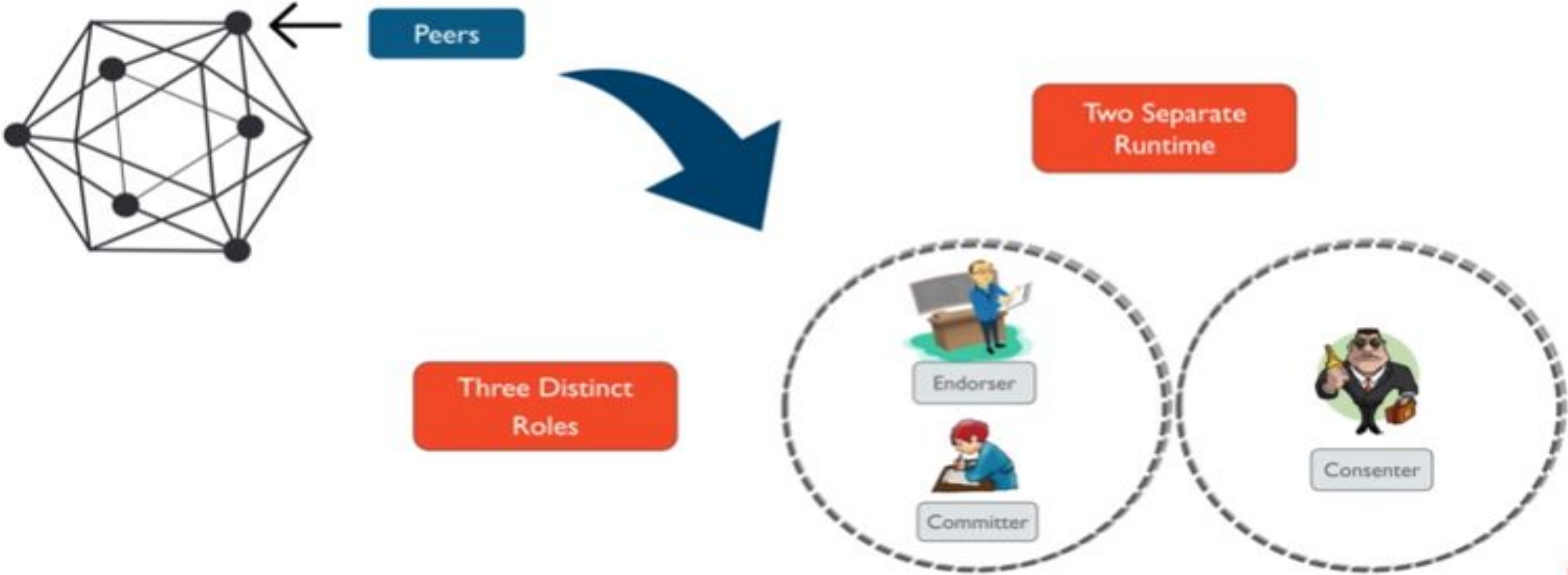
Membership Service

The app looks up a membership service validates Bob's membership

Verification

Both parties generate a result which are then sent to the consensus cloud to be ordered and verified

Different from Blockchain



Peers

Two Separate Runtime

Three Distinct Roles

Endorser

Committer

Consenter

Peer Roles

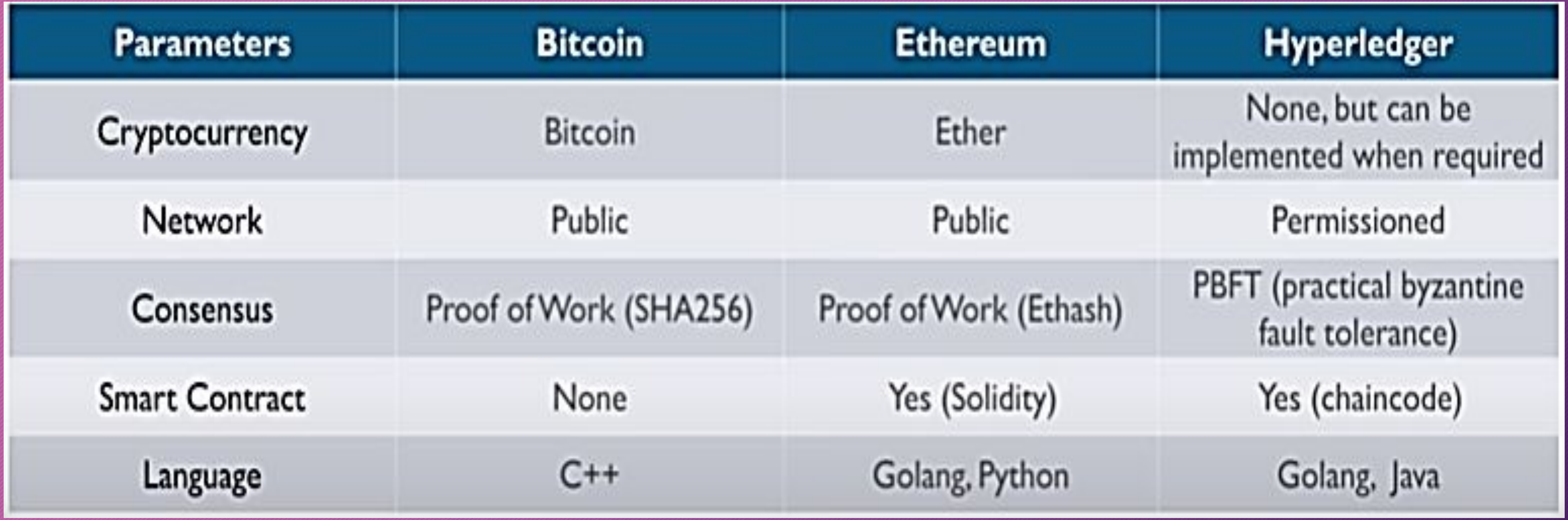
Committers – Append validated transaction to their specific ledger.

Endorsers – Simulate the transactions and prevent unstable and non deterministic transactions.

***All endorsers are committers but all committers may or may not be endorsers*.**

Consenter – Network consensus service. A collection of consensus service nodes (CSNs) will order transactions into blocks according to the networks chosen ordering implementation.

Comparison



Parameters	Bitcoin	Ethereum	Hyperledger
Cryptocurrency	Bitcoin	Ether	None, but can be implemented when required
Network	Public	Public	Permissioned
Consensus	Proof of Work (SHA256)	Proof of Work (Ethash)	PBFT (practical byzantine fault tolerance)
Smart Contract	None	Yes (Solidity)	Yes (chaincode)
Language	C++	Golang, Python	Golang, Java

Pseudo Anonymity

Pseudo-anonymity, means that a person will be linked to a public Bitcoin address, but no one will get to know the actual name or address.

To explain this in simple words, suppose a person sends a sum of money, then the receiver will get to know that the sender is linked to a bitcoin address but will not know the actual address. Hence, we say that bitcoin or any other alt currencies are not entirely anonymous.

	Traditional Database	**Public Blockchain**	**Private Blockchain**
How is governance managed?	Centralized	Decentralized	Federated
Who updates the ledger?	Single party	Unrestricted participants	Restricted participants
How is good behavior incentivized?	N/A	Cryptoeconomics	Reputational risk
Who has read-only access?	Users authorized by the database owner	Anyone	A group of selected actors/contributors
Who has writing access?	Users authorized by the database owner	Anyone	A group of selected actors/contributors
Are transactions anonymous to the public?	Yes	No	Yes
Does it require censorship resistance	No	Yes	No
Examples	Experian	Ethereum, Bitcoin blockchain	Enterprise Ethereum, PegaSys Plus

Crime-as-a-service

- **Crime-as-a-Service** Could Be the Next Big Threat to Your Business. **Crime-as-a-service** is when a professional **criminal** or group of **criminals** develop advanced tools, “kits” and other packaged **services** which are then offered up for sale or rent to other **criminals** who are usually less experienced.
- For example, someone might develop a ransomware kit that’s capable of encrypting important files where the victim must pay a ransom. They will then sell or rent that kit to other lower level cyber criminals, thus enabling them launch attacks.

Sybil attack

- In a **Sybil attack**, the attacker subverts the reputation system of a network service by creating a large number of pseudonymous identities and uses them to gain a disproportionately large influence.
- In the world of cryptocurrencies, a more relevant example is where somebody runs multiple nodes on a **blockchain** network.

Z Cash

- Zcash is a cryptocurrency aimed at using cryptography to provide enhanced privacy for its users compared to other cryptocurrencies such as Bitcoin. Zcash is based on Bitcoin's codebase.
- Zcash is the first widespread application of zk-SNARKs, a novel form of zero-knowledge cryptography. The strong privacy guarantee of Zcash is derived from the fact that shielded transactions in Zcash can be fully encrypted on the blockchain, yet still be verified as valid under the network’s consensus rules by using zk-SNARK proofs.

zk-SNARKs

- The acronym zk-SNARK stands for “Zero-Knowledge Succinct Non-Interactive Argument of Knowledge,” and refers to a proof construction where one can prove possession of certain information, e.g. a secret key, without revealing that information.
- “Zero-knowledge” proofs allow one party (the prover) to prove to another (the verifier) that a statement is true, without revealing any information beyond the validity of the statement itself. For example, given the hash of a random number, the prover could convince the verifier that there indeed exists a number with this hash value, without revealing what it is.

zk-SNARKs

- Example:
  - You need to make a purchase and you tell the seller that you have the sufficient amount of money to pay for that without actually telling him your account balance.
- This is made possible by proving the identity using public private key pair.
- If someone had access to the secret randomness used to generate these parameters, they would be able to create false proofs that would look valid to the verifier. For Zcash, this would mean the malicious party could create counterfeit coins. To prevent this from ever happening, Zcash generated the public parameters through an elaborate, multi-party ceremony.

Blockchain Double Spending

- Double spending means spending the same money twice.
- As we know, any transaction can be processed only in two ways. One is offline, and another is online.

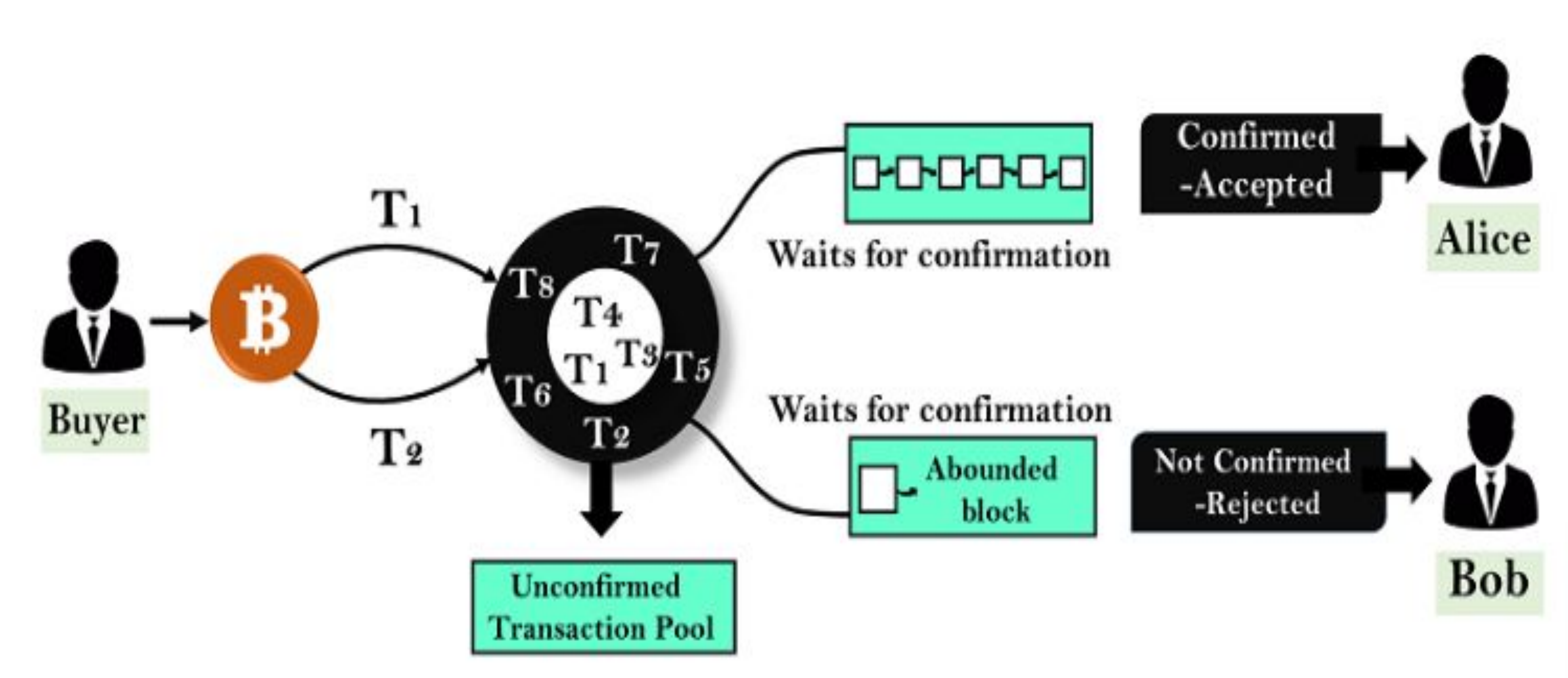
1. **Offline:** A transaction which involves physical currency or cash is known as an offline transaction.

2. **Online:** A transaction which involves digital cash is known as an online transaction.

Continue…

- In a physical currency, the double-spending problem can never arise.
- But in digital cash-like bitcoin, the double-spending problem can arise. Hence, bitcoin transactions have a possibility of being copied and rebroadcasted. It opens up the possibility that the same BTC could be spent twice by its owner.

Example


Buyer
T1
T2
T7
T8
T4
T3
T5
T1
T6
T2
Unconfirmed Transaction Pool
Waits for confirmation
Waits for confirmation
Abounded block
Confirmed -Accepted
Not Confirmed -Rejected
Alice
Bob

Finney Attack

- The Finney attack is named after **Hal Finney**.
- The Finney attack is one of the types of double-spending problem.
- In this attack, the attacker is the miner who mines blocks normally.
- In the block, he includes a transaction which sends some of his coins back to himself without broadcasting the transaction. When he finds a pre-mined block, he sends the same coins in a second transaction. The second transaction would be rejected by other miners, but this will take some time.
- To prevent this attack, the seller should wait for at least six blocks

Vector76 Attack

- The Vector76 attack is a combination of the **Race attack** and the **Finney attack.**
- In this attack, a miner creates two nodes, one of which is connected to the exchange node, and the other is connected to well-connected peers in the blockchain network.
- Now, the miner creates two transactions, one high value, and one low value. Then, the attacker pre-mines a high-value transaction to an exchange service.
- When a block is announced, he quickly sends the pre-mined block directly to the exchange service. When exchange service confirms the high-value transaction, the corrupted attacker sends a low-value transaction to the blockchain network that finally rejects the high-value transaction.

Continue…

- As a result, the corrupted attacker's account is deposited on the amount of the high-value transaction.

- This attack can be protected by disabling the incoming connections and only connecting to well-connected nodes.

Algorand

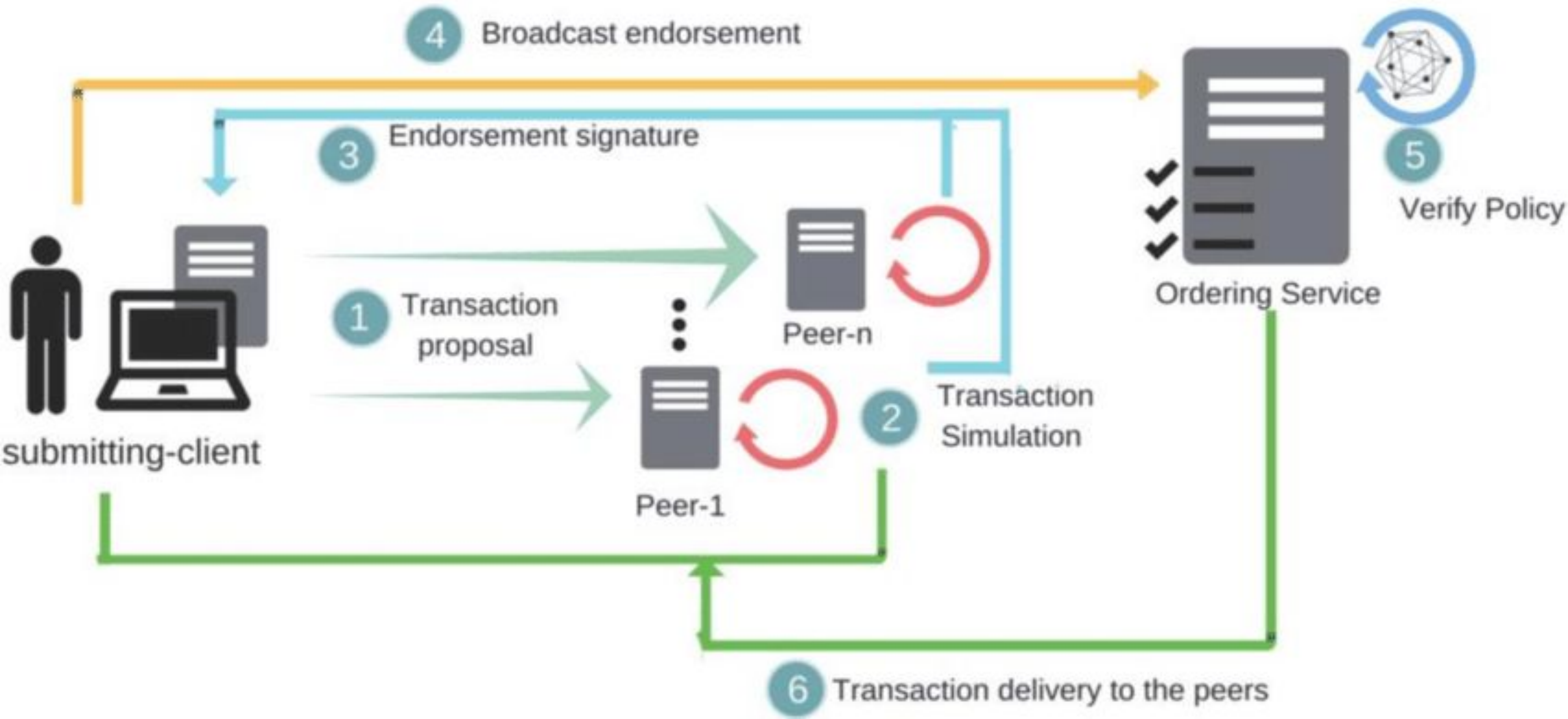
- **Algorand** is a Boston-based open-source software company working towards building a borderless economy. They've developed a permissionless, Pure Proof-of-Stake (PoS) protocol with open participation, scalability, security and transaction finality.
- Proposal phase: a single token is randomly selected, andits owner proposes the next blocks. However, this proposer is only known to the whole network during the propagation phase: it is already too late to interfere. In Pure PoS, every token has the same power in being selected.
- Voting round: a committeeof ownersof 1,000 random tokens is selected, approving the block proposed by the first user. As opposed to

Sharding

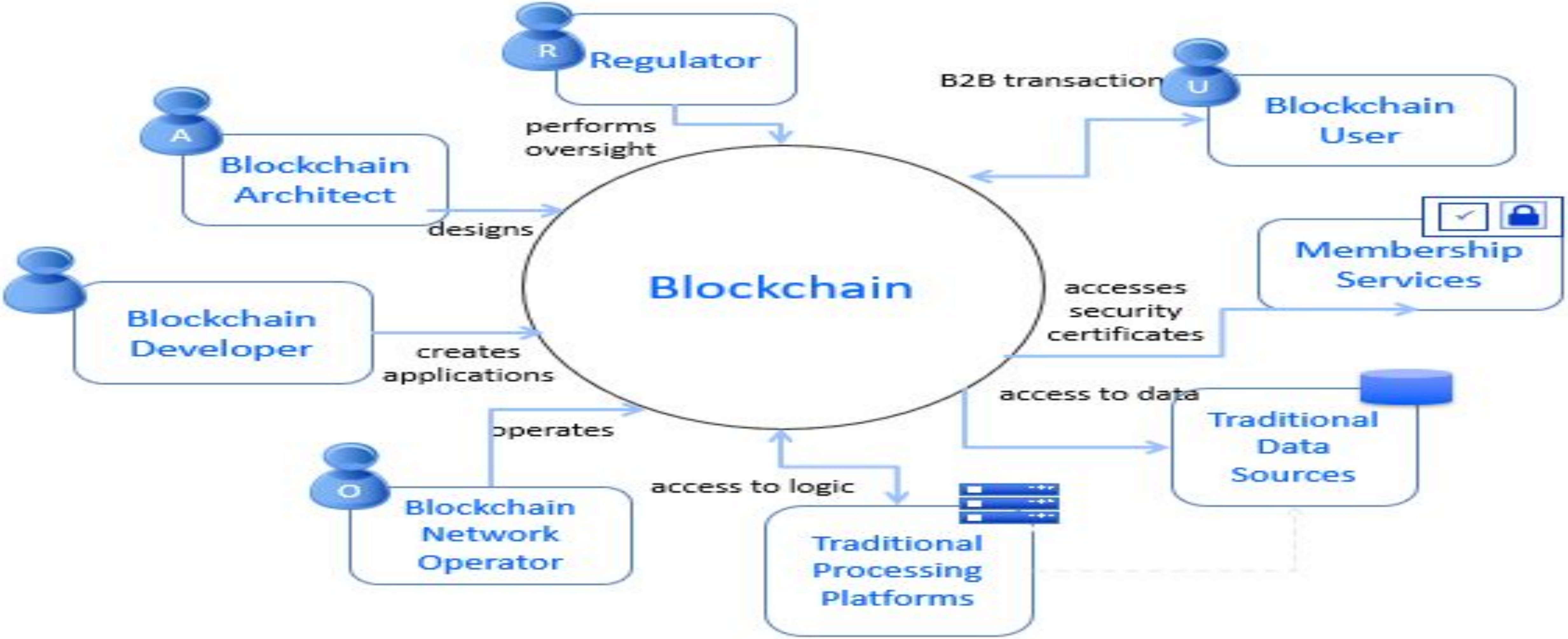
- **Sharding** is a type of database partitioning that separates very large databases the into smaller, faster, more easily managed parts called data shards. The word **shard** means a small part of a whole.
- Sharding could be the key to allowing blockchains to scale, while maintaining the privacy and security features that make the distributed ledger technology so efficient. But there are hurdles that need to be addressed.
- Sharding is a way of partitioning to spread out the computational and storage workload across a peer-to-peer (P2P) network so that each node isn't responsible for processing the entire network's transactional load. Instead, each node only maintains information related to its partition, or shard.

Advantages of Hyperledger

- Permissioned membership
- Performance, scalability, and levels of trust
- Data on a need-to-know basis
- Rich queries over an immutable distributed ledger
- Modular architecture supporting plug-in components
- Protection of digital keys and sensitive data


4 Broadcast endorsement
3 Endorsement signature
5
Verify Policy
Ordering Service
1 Transaction proposal
Peer-n
Peer-1
submitting-client
2 Transaction Simulation
6 Transaction delivery to the peers

Actors in Blockchain



Continue…

1. **Blockchain Architect -**
   Responsible for architecture and design of the blockchain. Blockchain Architect is the one who is going to design, how the blockchain solution is going to be built. He will figure out what is some information that needs to get stored, what are the transactions and the business logic that needs to be embedded onto the network, and so on.

2. **Blockchain Developer -**
   The developer of applications and smart contracts that interact with the blockchain and are used by the blockchain users. The blockchain developer is the one who is going to take what has been an architect and then develop the actual code that will run on the blockchain network itself.

Continue…

3. Blockchain Network Operator -
Manages and monitors the blockchain network. Each sub-work or the business in the network has a blockchain network operator. He also runs the blockchain network.

4. Traditional Processing Platforms -
An existing computer system may be used by the blockchain to augment processing. The system may also need to initiate the request to the blockchain. Other systems send or get information that is required to build a blockchain solution.

5. Traditional Data Sources -
An existing computer system may provide data to influence the behavior of the smart contract. They are also part of the overall solution to store external data.

Continue…

6. Membership Services -

It manages different types of certificates, which are required to run a permission blockchain. Membership services provide the identity for users to come and transact on the blockchain. For example, if you open an account with the bank, they give you a username and password, a kind of login to access web services, Membership services is going to do more than that not only username and password but also give a digital certificate that will allow you to transact on the network.

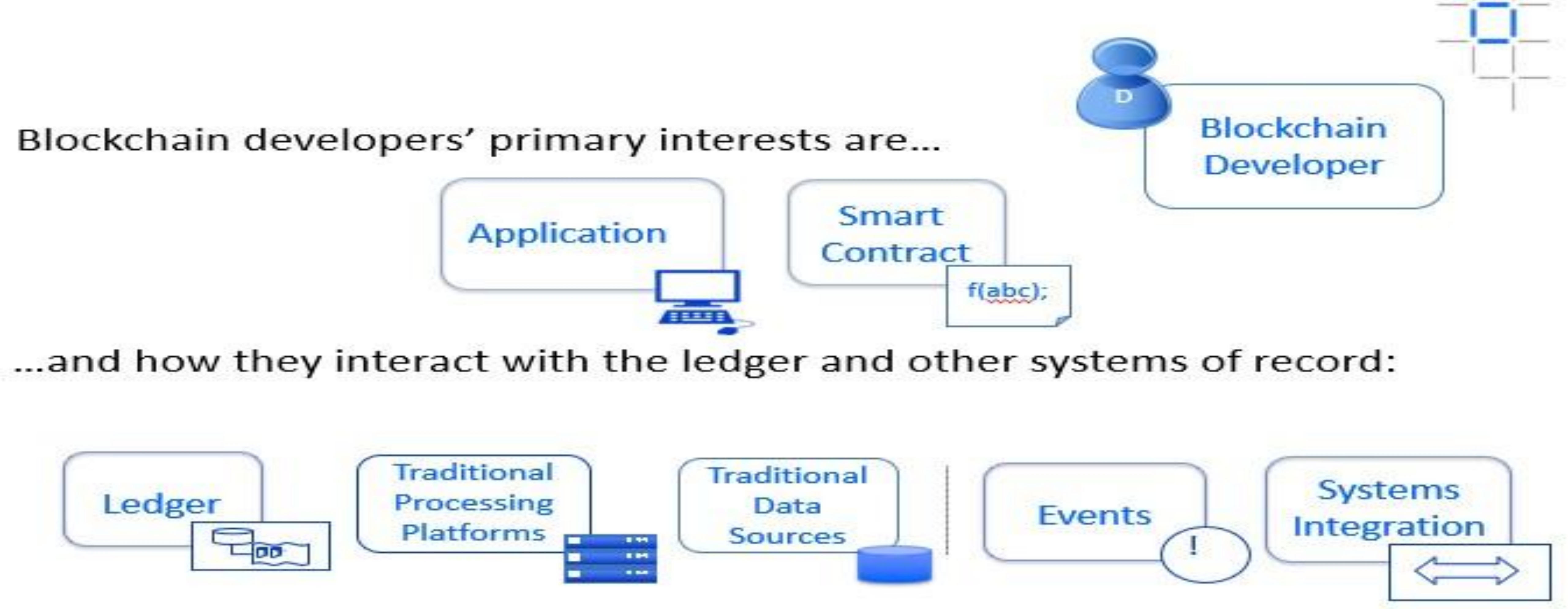
7. Blockchain User -

The business user, operating in a business network. User experiences the application of that blockchain solution. They are not aware of blockchain. Blockchain user is the one who is going to perform the business transactions on the blockchain, So, these users could belong to multiple organizations that are participating in that blockchain.

Continue…

8. **Blockchain Regulator** -
The overall authority is a business network. Specifically, regulators are required to read the ledger's content broadly. The Blockchain regulator is an optional one, they might have only, read-only access onto the network where they see the transactions being performed are legitimate or not, following policies or not, etc.

The Blockchain Developer



Blockchain developers' primary interests are...

Blockchain Developer

Application

Smart Contract

f(abc);

...and how they interact with the ledger and other systems of record:

Ledger

Traditional Processing Platforms

Traditional Data Sources

Events

!

Systems Integration

Components of Blockchain Solution

Ledger		List of transactions maintained by peers
Smart Contract	f(abc);	Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network
Events		Emits notifications of operations on network
Systems Management		Enables us to create/monitor blockchain components
Wallet		Securely manages a user’s credentials
Systems Integration		Integrate blockchain with eternal systems

Hyperledger Composer

- **Hyperledger Composer** is a set of collaboration tools for business owners and developers that make it easy to write chaincode (also known as smart contracts) for Hyperledger Fabric and**Decentralized Applications (DApps)**. With Composer, youcan quickly build Proof-of-Concept anddeploy chaincode to the blockchainin a short amount of time. Hyperledger Composer consists of the following toolsets:
- **A modeling language called CTO**: A domain modeling language that defines a business model, concept, and function for a business network definition
- **Playground**: Rapid configuration, deployment, and testing of a business network
- **Command-line interface (CLI) tools**: The client command-line

Hyperledger Composer

- Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier.
- Instead of developing smart contracts from scratch, Composer provides a convenience layer and business-level abstractions to implement smart contracts on Fabric. Composer also makes it easier for you to connect to your business network from a web or mobile application

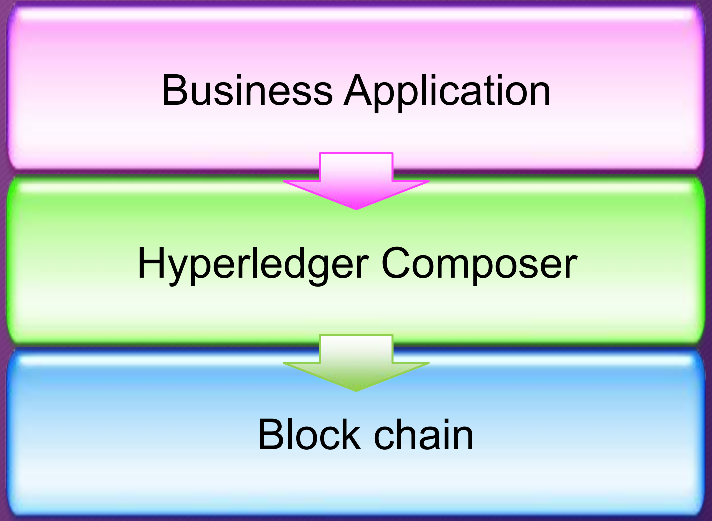
Hyperledger Composer

- Hyperledger Composer is a set of collaboration tools for building blockchain business networks that make it simple and fast for business owners and developers to create smart contracts and blockchain applications to solve business problems

- Extensive
- Open development toolset and
- Framework to make developing Blockchain applications easier.

Hyperledger Composer

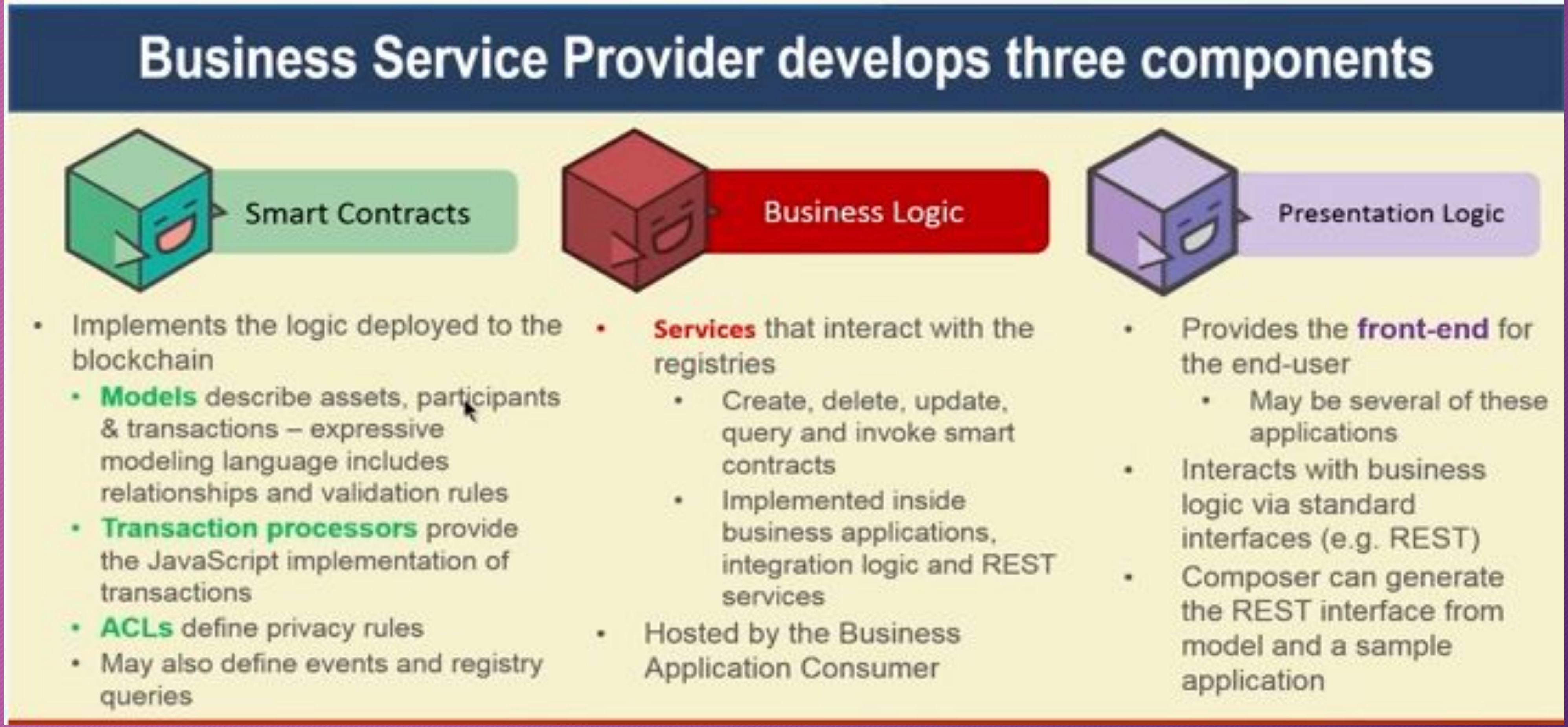
- A suite of high level application **abstractions** for business networks
- Emphasis on business-centric vocabulary for quick solution creation
- Reduce risk, and increase understanding and flexibility
- Features
  - Model your business networks, test and expose via APIs
  - Applications invoke APIs transactions to interact with business network
  - Integrate existing systems of record using REST APIs



Developing With Hyperledger Composer

- Hyperledger Composer provides multiple tools to facilitate the development of business networks. The tools permit both online and local development.
- The Hyperledger Composer Playground provides a user interface for the configuration, deployment, and manual testing of a business network.
- Advanced Playground features permit users to manage the security of the business network, invite participants to business networks, and connect to multiple blockchain business networks.
- Hyperledger Composer Playground is not intended to facilitate source code version control or the development of automated testing suites for use within build automation.
- For these requirements, you can use various version control tools and code editors, which have tools to provide syntax highlighting and scenario-based testing.

Developing Mainly three component in composer



Business Service Provider develops three components

Smart Contracts

- Implements the logic deployed to the blockchain
  - **Models** describe assets, participants & transactions – expressive modeling language includes relationships and validation rules
  - **Transaction processors** provide the JavaScript implementation of transactions
  - **ACLs** define privacy rules
  - May also define events and registry queries

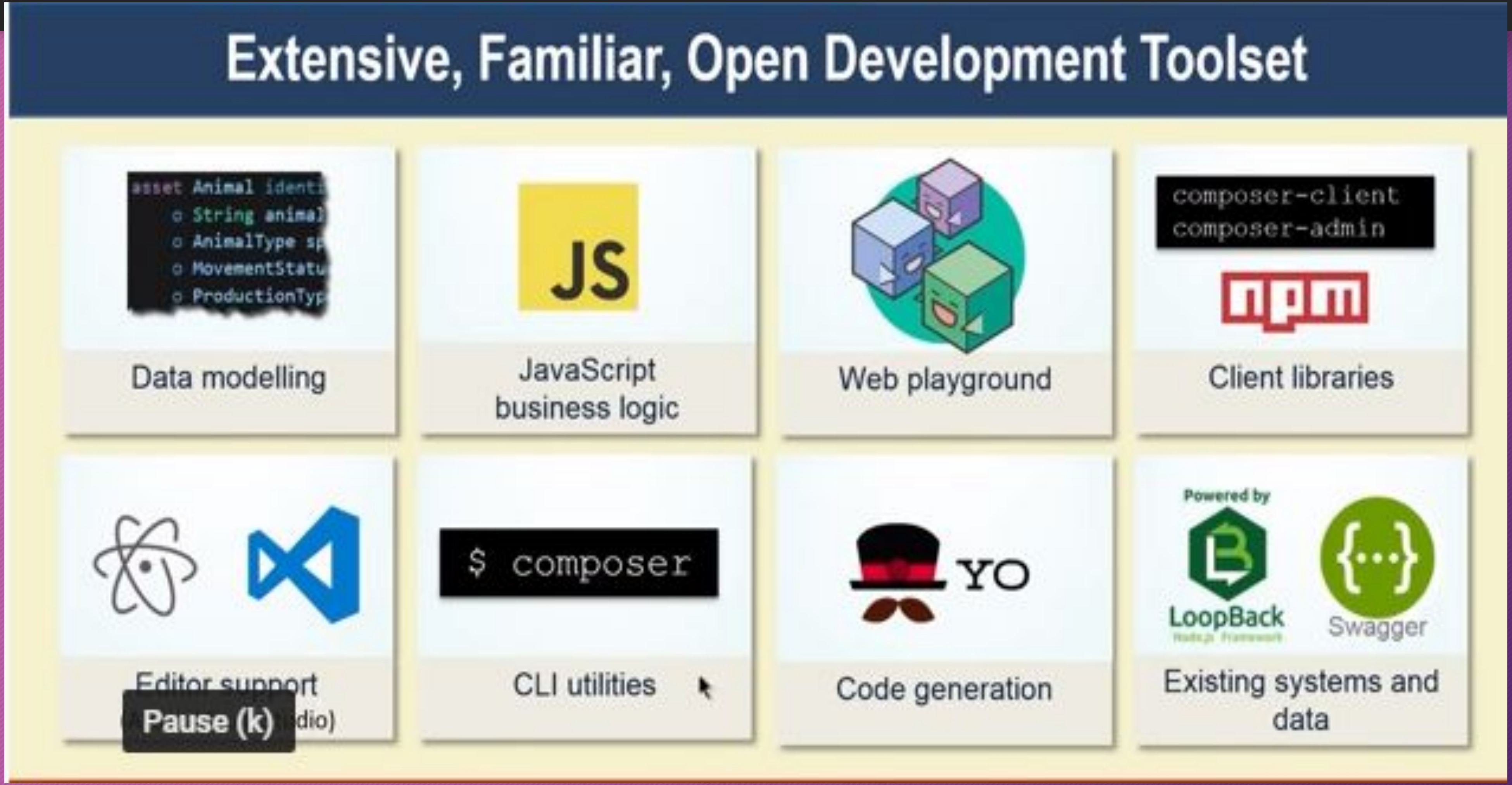
Business Logic

- **Services** that interact with the registries
  - Create, delete, update, query and invoke smart contracts
  - Implemented inside business applications, integration logic and REST services
- Hosted by the Business Application Consumer

Presentation Logic

- Provides the **front-end** for the end-user
  - May be several of these applications
- Interacts with business logic via standard interfaces (e.g. REST)
- Composer can generate the REST interface from model and a sample application

Toolset in composer



Extensive, Familiar, Open Development Toolset

```
asset Animal identi
    o String animal
    o AnimalType sp
    o MovementStatu
    o ProductionTyp
```

Data modelling

JavaScript business logic

Web playground

```
composer-client
composer-admin
```

Client libraries

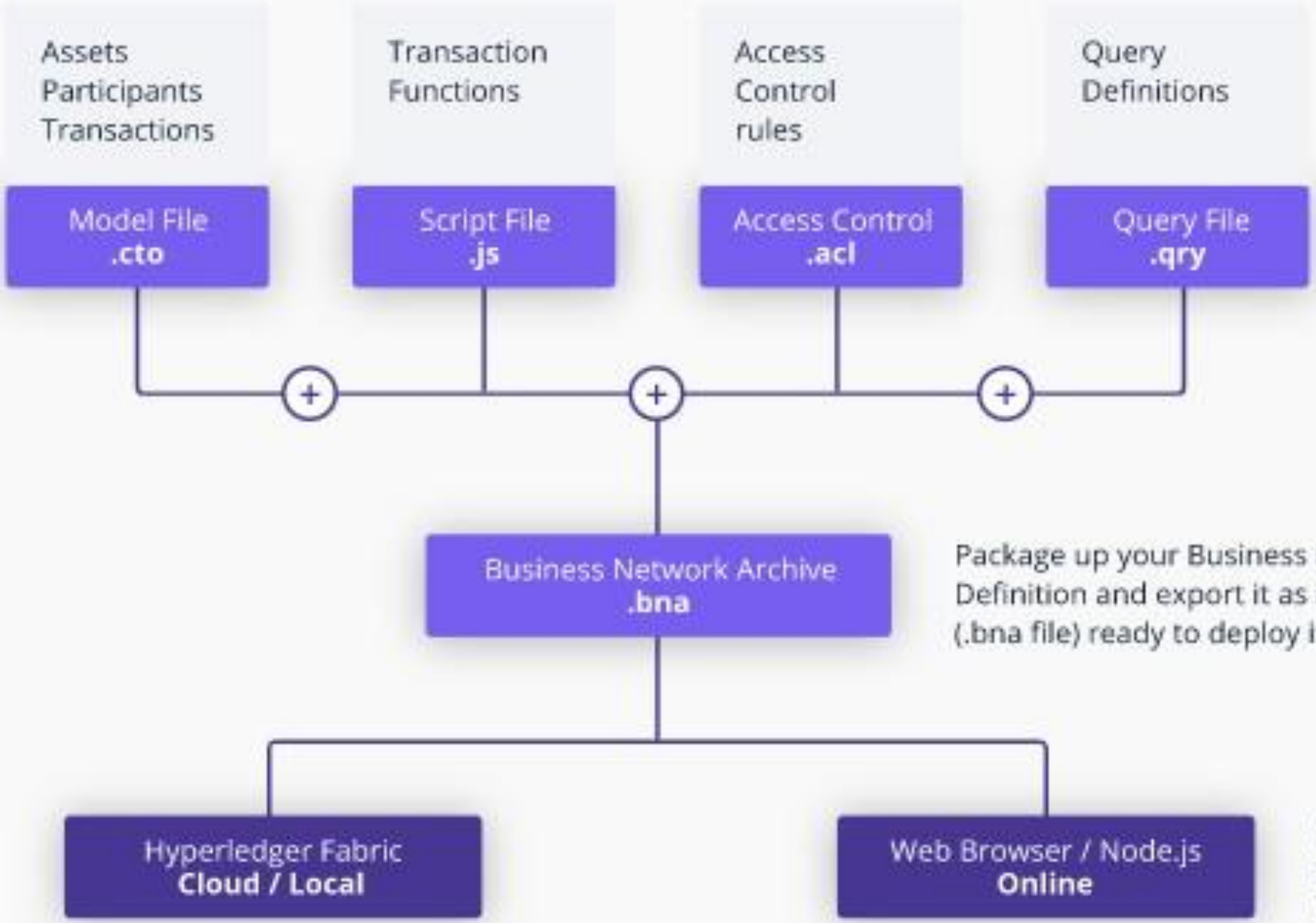
Editor support
dio)

```
$ composer
```

CLI utilities

Code generation

Existing systems and data

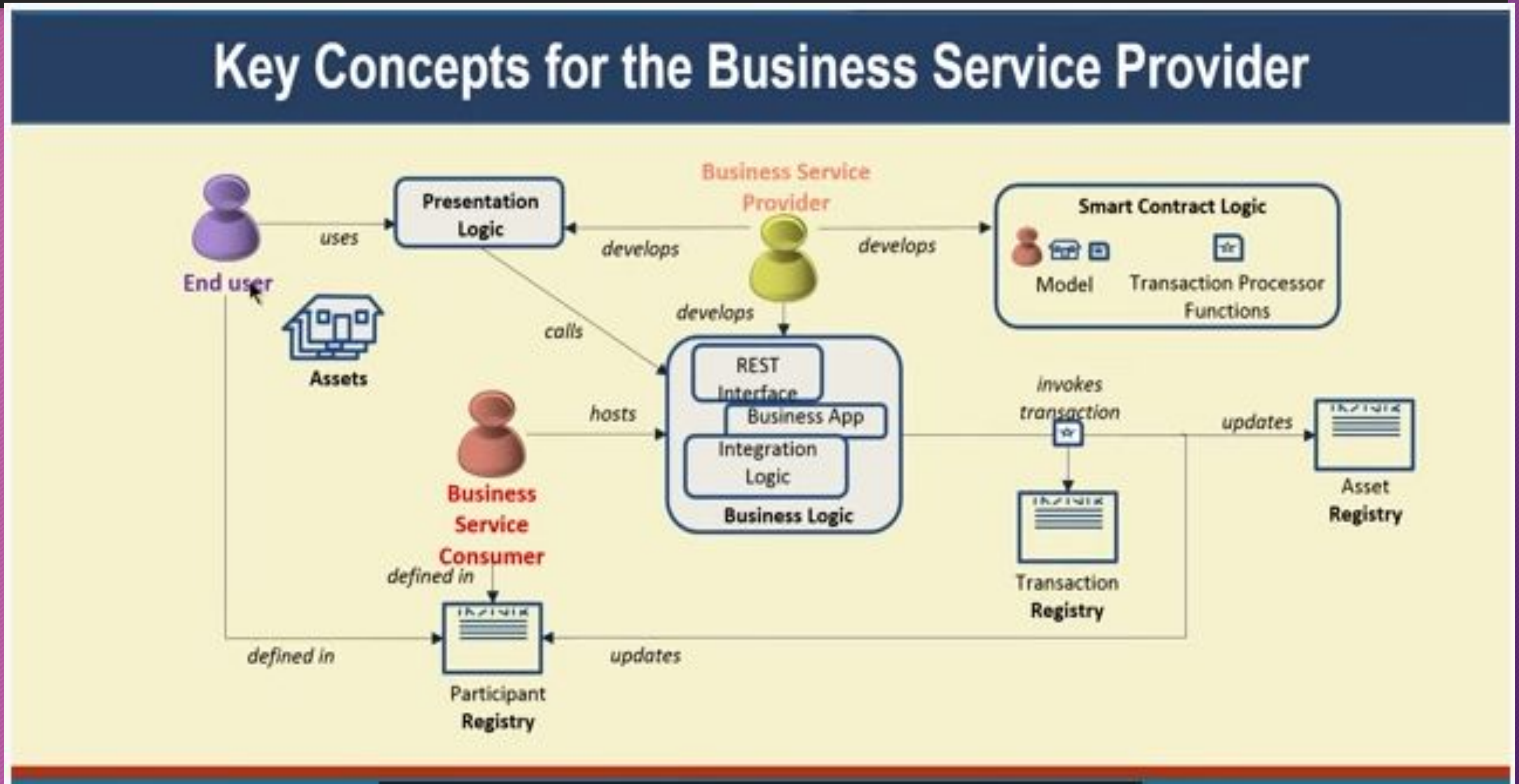


Use Composer to create a Business Network Definition, comprised of Model (.cto), Script (.js), ACL (.acl) and Query (.qry) files.

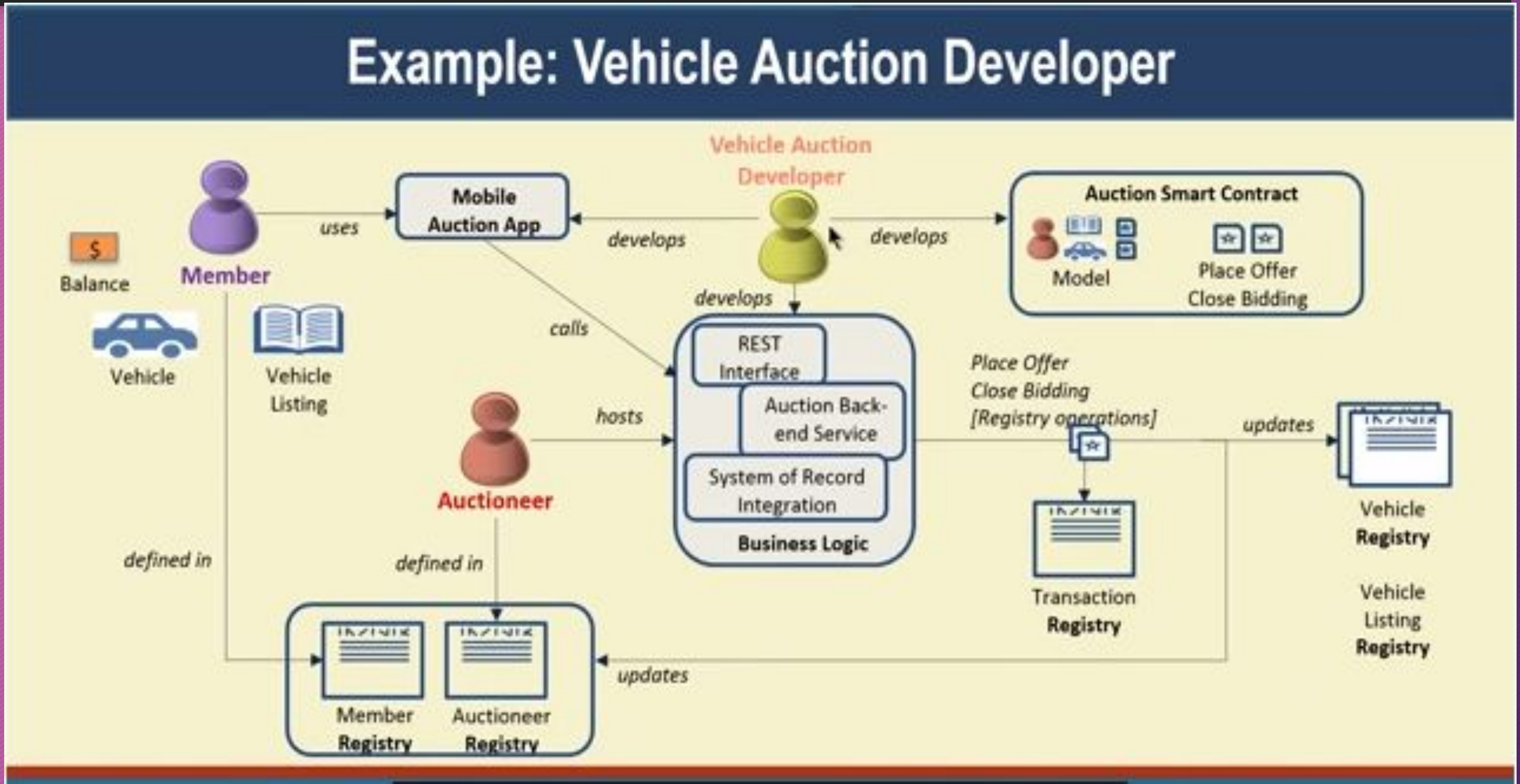
Package up your Business Network Definition and export it as an archive (.bna file) ready to deploy it somewhere.

Use ID Cards (which include connection profiles and credentials) to deploy your Business Network Definition to a distributed ledger.

Concept of composer

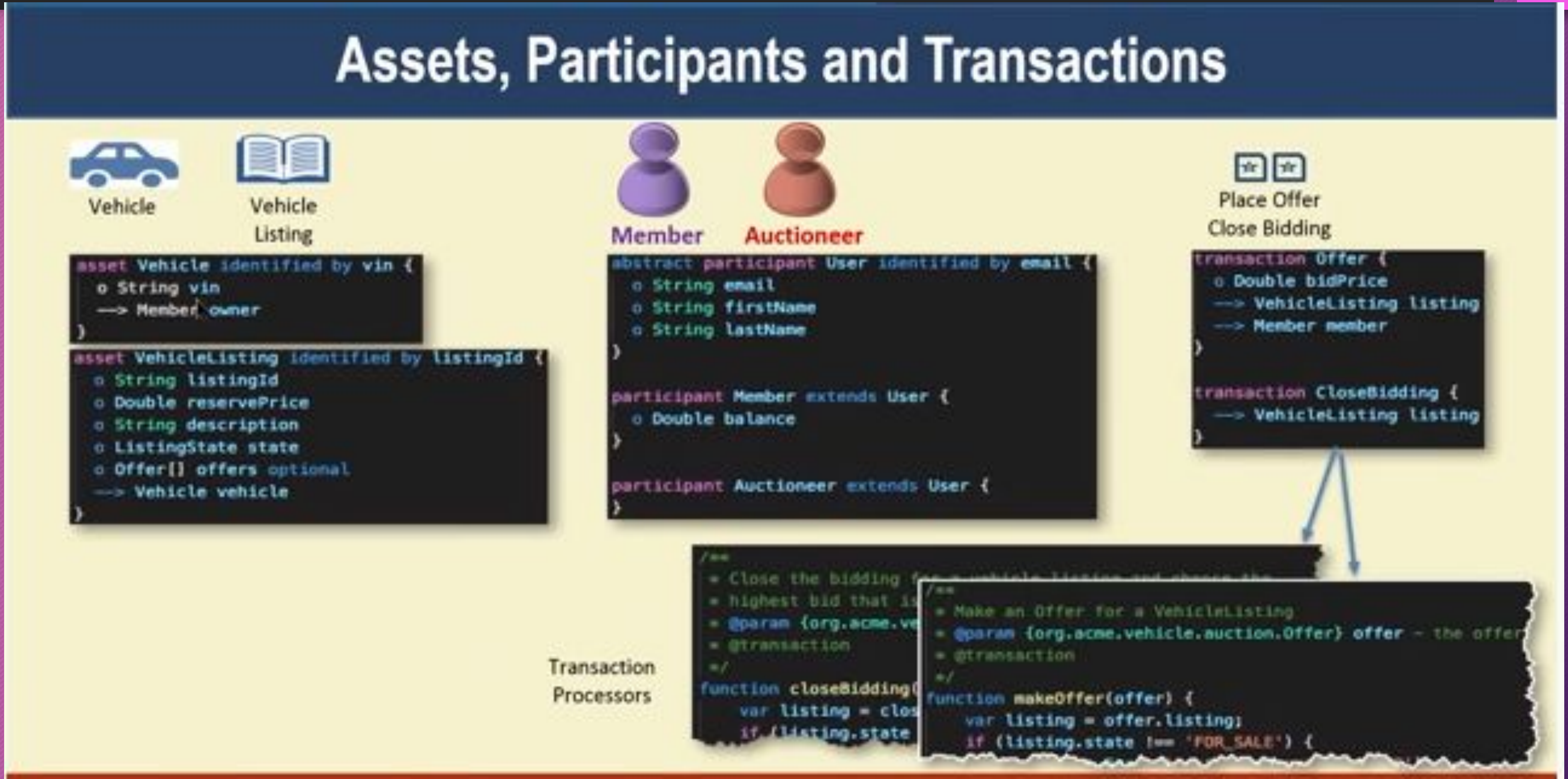


Example



Example: Vehicle Auction Developer

Defining process



Assets, Participants and Transactions

```
asset Vehicle identified by vin {
  o String vin
  --> Member owner
}
```

```
asset VehicleListing identified by listingId {
  o String listingId
  o Double reservePrice
  o String description
  o ListingState state
  o Offer[] offers optional
  --> Vehicle vehicle
}
```

```
abstract participant User identified by email {
  o String email
  o String firstName
  o String lastName
}

participant Member extends User {
  o Double balance
}

participant Auctioneer extends User {
}
```

```
transaction Offer {
  o Double bidPrice
  --> VehicleListing listing
  --> Member member
}

transaction CloseBidding {
  --> VehicleListing listing
}
```

```
/**
 * Close the bidding
 * highest bid that is
 * @param {org.acme.ve
 * @transaction
 */
function closeBidding(
    var listing = clos
    if (listing.state
```

```
/**
 * Make an Offer for a VehicleListing
 * @param {org.acme.vehicle.auction.Offer} offer - the offer
 * @transaction
 */
function makeOffer(offer) {
    var listing = offer.listing;
    if (listing.state !== 'FOR_SALE') {
```

Hyperledger Fabric

- Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to consensus that enables performance at scale while preserving privacy.

Hyperledger – Case Study - Honeywell

- Honeywell Aerospace creates online parts marketplace with Hyperledger Fabric.
- Model the marketplace on popular e-commerce sites
- Use blockchain to overcome the trust barrier
- Choose an effective blockchain framework
- Set high standards that encourage quick adoption
- Provide a stream of new benefits

How does Hyperledger Composer work in practice?

For an example of a business network in action; a realtor can quickly model their business network as such:

- **Assets:** houses
- **Participants:** buyers and homeowners
- **Transactions:** buying or selling houses, and creating and closing listings
- Participants can have their access to transactions restricted based on their role as either a buyer, seller, or realtor. The realtor can then create an application to present buyers and sellers with a simple user interface for viewing open listings and making offers. This business network could also be integrated with existing inventory system, adding new houses as assets and removing sold properties. Relevant other parties can be registered as participants, for example a land registry might interact with a buyer to transfer ownership of the land.

Peer Channel

- Create – Create a channel and write genesis block to a file.
- Fetch – Fetch a specified block and write it to a file.
- Getinfo – Get Blockchain information of a specified channel.
- Join – Joins a new peer to the channel.
- List – List of channels a peer has joined.
- Signconfigtx - Signs the supplied configtx update file **in** place on the filesystem.
- Update - Signs **and** sends the supplied configtx update file to the channel.

Composer Package

Composer supports four types:

- **library**: This is the default. It will simply copy the files to /vendor.
- **project**: This denotes a project rather than a library. For example application shells like the Symfony standard edition, CMSs like the SilverStripe installer or full fledged applications distributed as packages.
- **metapackage**: An empty package that contains requirements and will trigger their installation, but contains no files and will not write anything to the filesystem.
- **composer-plugin**: A package of type composer-plugin may provide an installer for other packages that have a custom type. Only use a custom type if you need custom logic during installation.

Byfn.sh

- The build your first network (BYFN) scenario provisions a sample Hyperledger Fabric network consisting of two organizations, each maintaining two peer nodes. It also will deploy a “Solo” ordering service by default, though other ordering service implementations are available.

Crypto generator

- Cryptogen tool to generate the cryptographic material (x509 certs and signing keys) for our various network entities. These certificates are representative of identities, and they allow for sign/verify authentication to take place as our entities communicate and transact.

Crypto generator

Cryptogen consumes a file — crypto-config.yaml — that contains the network topology and allows us to generate a set of certificates and keys for both the Organizations and the components that belong to those Organizations. Each Organization is provisioned a unique root certificate (ca-cert) that binds specific components (peers and orderers) to that Org. By assigning each Organization a unique CA certificate, we are mimicking a typical network where a participating Member would use its own Certificate Authority. Transactions and communications within Hyperledger Fabric are signed by an entity's private key (keystore), and then verified by means of a public key (signcerts).

Crypto generator

You will notice a count variable within this file. We use this to specify the number of peers per Organization; in this case there are two peers per Org. After we run the cryptogen tool, the generated certificates and keys will be saved to a folder titled crypto-config. Note that the crypto-config.yaml file lists five orderers as being tied to the orderer organization. While the cryptogen tool will create certificates for all five of these orderers, unless the Raft or Kafka ordering services are being used, only one of these orderers will be used in a Solo ordering service implementation and be used to create the system channel and mychannel.

Configuration Transaction Generator

The configtxgen tool is used to create four configuration artifacts:

- orderer genesis block,
- channel configuration transaction,
- and two anchor peer transactions - one for each Peer Org.

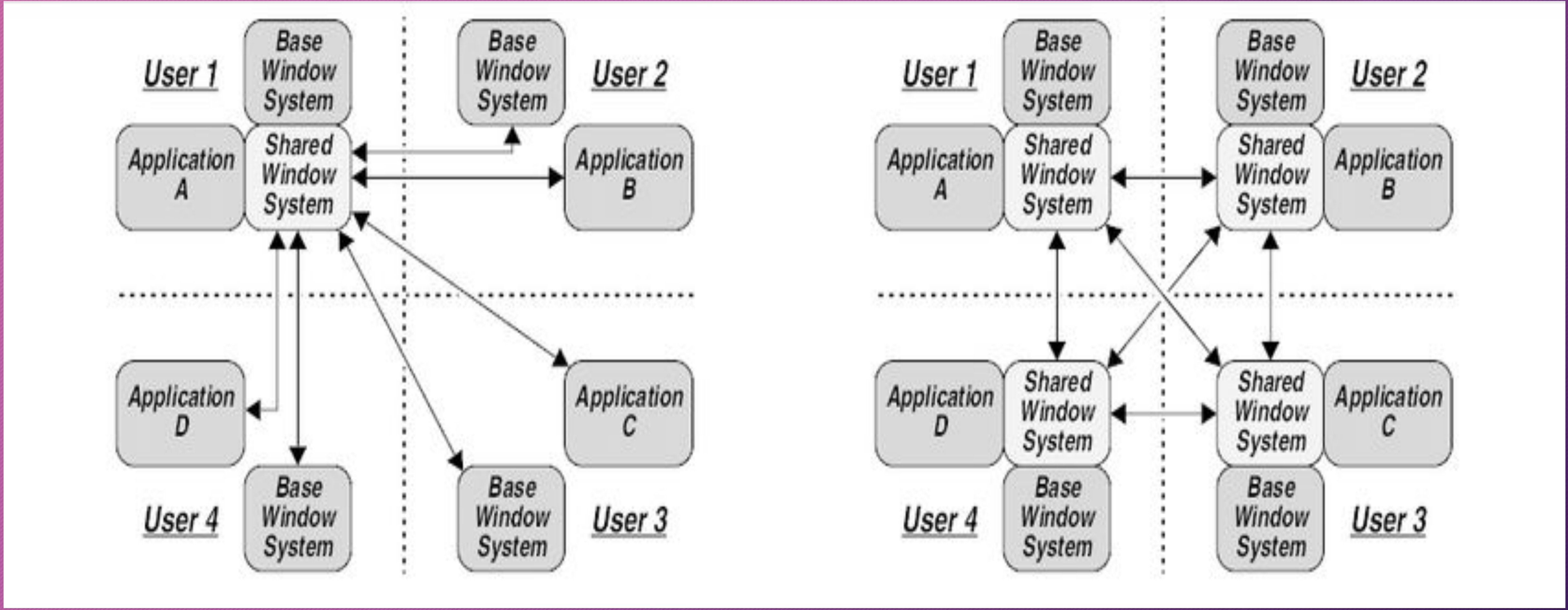
The orderer block is the Genesis Block for the ordering service, and the channel configuration transaction file is broadcast to the orderer at Channel creation time.

The anchor peer transactions, as the name might suggest, specify each Org's Anchor Peer on this channel.

Configuration Transaction Generator

- Configtxgen consumes a file - configtx.yaml - that contains the definitions for the sample network. There are three members - one Orderer Org (OrdererOrg) and two Peer Orgs (Org1 & Org2) each managing and maintaining two peer nodes. This file also specifies a consortium - SampleConsortium - consisting of our two Peer Orgs.
- TwoOrgsOrdererGenesis: generates the genesis block for a Solo ordering service.
- SampleMultiNodeEtcdRaft: generates the genesis block for a Raft ordering service. Only used if you issue the -o flag and specify etcdraft.
- SampleDevModeKafka: generates the genesis block for a Kafka ordering service. Only used if you issue the -o flag and specify kafka.
- TwoOrgsChannel: generates the genesis block for our channel, mychannel

Software Architecture



Advantages of Distributed System

- Higher computing power
- Cost reduction
- Higher reliability
- Ability to grow naturally

Higher Computing Power

- The computing power of a distributed system is the result of combining the computing power of all connected computers. Hence, distributed systems typically have more computing power than each individual computer. This has been proven true even when comparing distributed systems comprised of computers of relatively low computing power with isolated super computers.

Cost Reduction

- The price of mainstream computers, memory, disk space, and networking equipment has fallen dramatically during the past 20 years. Since distributed systems consist of many computers, the initial costs of distributed systems are higher than the initial costs of individual computers. However, the costs of creating, maintaining, and operating a super computer are still much higher than the costs of creating, maintaining, and operating a distributed system. This is particularly true since replacing individual computers of a distributed system can be done with no significant overall system impact.

Higher Reliability

- The increased reliability of a distributed system is based on the fact that the whole network of computers can continue operating even when individual machines crash. A distributed system does not have a single point of failure. If one element fails, the remaining elements can take over. Hence, a single super computer typically has a lower reliability than a distributed system.

Ability to Grow Naturally

- The computing power of a distributed system is the result of the aggregated computing power of its constituents. One can increase the computing power of the whole system by connecting additional computers with the system. As a result, the computing power of the whole system can be increased incrementally on a fine-grained scale. This supports the way in which the demand for computing power increases in many organizations. The incremental growth of distributed systems is in contrast to the growth of the computing power of individual computers. Individual computers provide identical power until they are replaced by a more powerful computer. This results in a discontinuous growth of computing power, which is only rarely appreciated by the consumers of computing services.

The Disadvantages of Distributed Systems

- Coordination overhead
- Communication overhead
- Dependency on networks
- Higher program complexity
- Security issues

Coordination Overhead

- Distributed systems do not have central entities that coordinate their members. Hence, the coordination must be done by the members of the system themselves. Coordinating work among coworkers in a distributed system is challenging and costs effort and computing power that cannot be spent on the genuine computing task, hence, the term coordination overhead.

Communication Overhead

- Coordination requires communication. Hence, the computers that form a distributed system have to communicate with one another. This requires the existence of a communication protocol and the sending, receiving, and processing of messages, which in turn costs effort and computing power that cannot be spend on the genuine computing task, hence, the term communication overhead.

Dependencies on Networks

- Any kind of communication requires a medium. The medium is responsible for transferring information between the entities communicating with one another. Computers in distributed systems communicate by means of messages passed through a network. Networks have their own challenges and adversities, which in turn impact the communication and coordination among computers that form a distributed system. However, without any network, there will be no distributed system, no communication, and therefore no coordination among the nodes, thus the dependency on networks

Higher Program Complexity

- Solving a computation problem involves writing programs and software. Due to the disadvantages mentioned previously, any software in a distributed system has to solve additional problems such as coordination, communication, and utilizing of networks. This increases the complexity of the software.

Security Issues

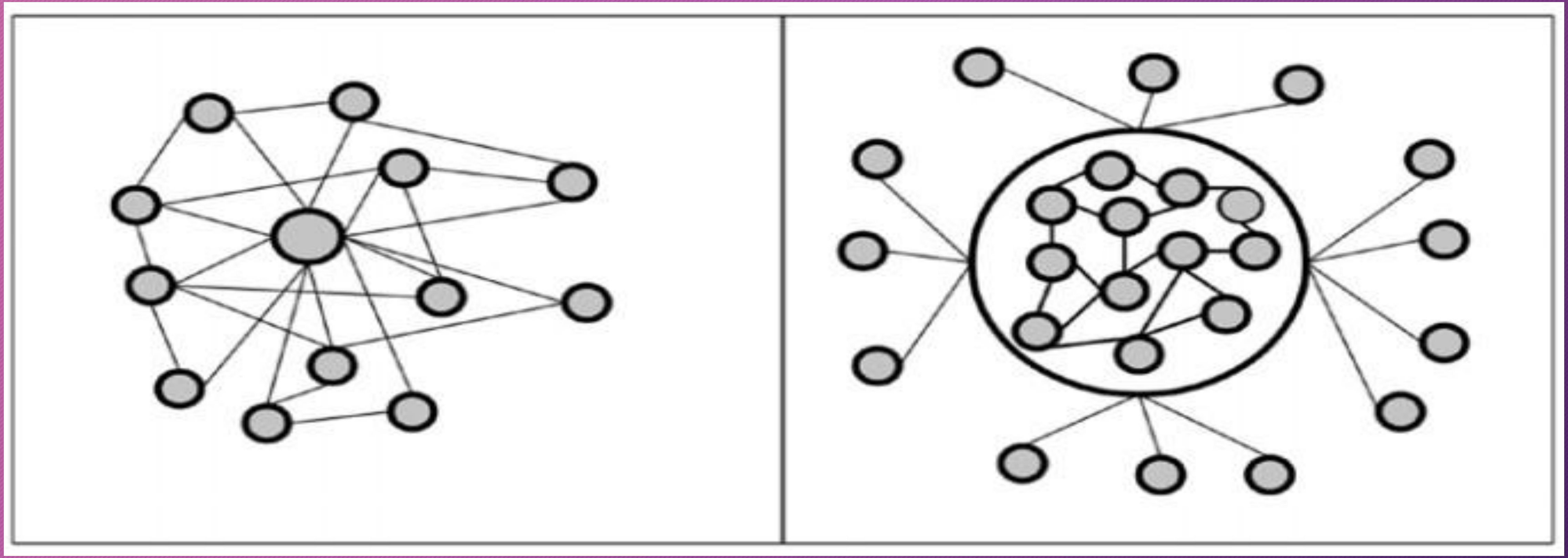
- Communication over a network means sending and sharing data that are critical for the genuine computing task. However, sending information through a network implies security concerns as untrustworthy entities may misuse the network in order to access and exploit information. Hence, any distributed system has to address security concerns. The less restricted the access to the network over which the distributed nodes communicate is, the higher the security concerns are for the distributed system.

Distributed Peer-to-Peer Systems

- Peer-to-peer networks are a special kind of distributed systems. They consist of individual computers (also called nodes), which make their computational resources (e.g., processing power, storage capacity, data or network bandwidth) directly available to all other members of the network without having any central point of coordination. The nodes in the network are equal concerning their rights and roles in the system. Furthermore, all of them are both suppliers and consumers of resources.

Mixing Centralized and Distributed Systems

- Centralized and distributed systems are architectural antipodes. Technical antipodes have always inspired engineers to create hybrid systems that inherit the strength of their parents. Centralized and distributed systems are no exception to this. There are two archetypical ways of combining these antipodes, and they need to be understood since they will become important when learning about blockchain applications in the real world. They are centrality within a distributed system and the distributed system inside the center.



Peer to Peer System - Napster

- The music industry has worked for a long time in the following way: musicians made contracts with studios, which recorded the songs, produced and marketed the music records on a variety of media (e.g., vinyl, tape, or CD), which in turn were sold to the customers via a variety of distribution channels, including department stores and specialized shops. The studios actually worked as intermediaries between musicians and people who enjoy listening to music. Music studios could maintain their role as intermediaries due to their exclusive knowledge and skills in producing, marketing, and distributing records. However, in the first decade of the 2000s, the environment in which the music studios operated changed dramatically.

Peer to Peer System - Napster

- The digitalization of music, the availability of recording equipment at affordable prices, the growing spread of privately used PCs, and the emergence of the Internet made music studios dispensable. The three functions of music studios—producing, marketing, and distributing records—could be done by the artists and the consumers themselves. Napster played a major role in the replacement of the music studios as intermediaries. With Napster, people no longer relied on the music studios to get the latest hits. It was possible to share individual music files with people all over the world without the need to buy any CDs. The peer-to-peer approach of Napster, actually being a kind of a digital sharing bazaar for mp3 files, gave consumers access to a wider range of music than ever before, making the music studios partly dispensable and causing them significant losses.

The Potential of Peer-to-Peer Systems

- The power of peer-to-peer systems is not restricted to the music industry. Each industry that mainly acts as a middleman between producers and customers of immaterial or digital goods and services is vulnerable to being replaced by a peer-to-peer system. This statement may sound a bit abstract, but you may discover many middlemen for immaterial and digital goods and services around you once you recognize the largest of them all: the financial industry.

Centralized Peer to Peer

- Peer-to-peer systems aredistributed computer systems by construction since they are made of individual nodes that share their computational resources among others. However, there are also peer-to-peer systems that still utilize elements of centralization. Centralized peer-to-peer systems maintain central nodes to facilitate the interaction between peers, to maintain directories that describe the services offered by the peer nodes, or to perform look-ups and identification of the nodes. An example of a centralized peer-to-peer system is Napster, which maintained a central database of all nodes connected with the system and the songs available on these nodes.

What exactly is a peer to peer system?

- Peer-to-peer systems are distributed software systems that consist of nodes (individual computers), which make their computational resources (e.g., processing power, storage capacity, or information distribution) directly available to another. When joining a peer-to-peer system, users turn their computers into nodes of the system that are equal concerning their rights and roles. Although users may differ with respect to the resources they contribute, all the nodes in the system have the same functional capability and responsibility. Hence, the computers of all users are both suppliers and consumers of resources. For example, in a peer-to-peer file sharing system, the individual files are stored on the users' machines. When someone wants to download a file in such a system, he or she is downloading it from another person's machine, which could be the next door neighbor or someone located halfway around the world.

The Link Between Peer-to-Peer Systems and the Blockchain

- The blockchain can be considered a tool for achieving and maintaining integrity in distributed systems. Purely distributed peer-to-peer systems may use the blockchain in order to achieve and to maintain system integrity. Hence, the link between purely distributed peer-to-peer systems and the blockchain is its usage for achieving and maintaining integrity in purely distributed systems.

The Link Between Peer-to-Peer Systems and the Blockchain

- The relation between purely distributed peer-to-peer systems to the blockchain is that the former uses the latter as a tool to achieve and maintain integrity. Hence, the argument that explains the excitement about and the potential of the blockchain is: Purely distributed peer-to-peer systems have a huge commercial potential as they can replace centralized systems and change whole industries due to disintermediation. Since purely distributed peer-to-peer systems may use the blockchain for achieving and maintaining integrity, the blockchain becomes important as well. However, the major fact that excites people is the disintermediation. The blockchain is only a means to an end that helps to achieve that.

Vulnerabilities of Peer to peer system

- Counterfeiting bank notes is a severe crime in any country because it undermines the foundation and functioning of the economy by creating purchasing power that is not backed up by valuable resources. As a result, most bank notes are equipped with security features that make counterfeiting impossible or prohibitively costly at least. These security features, such as unique numbers, watermarks, or fluorescent fibers, work well with physical bank notes and other physical goods. But what happens if money or goods become digital and are managed in distributed peer-to-peer systems of ledgers? This step explains a specific vulnerability of distributed peer-to-peer systems used for managing ownership that is equivalent to counterfeiting bank notes. As it turns out, this vulnerability is a prominent example of violated system integrity.

Vulnerabilities of Peer to peer system

- Let's consider a peer-to-peer system for managing ownership of real estate. In such a system, the ledgers that keep track of ownership information are maintained by the individual computers of its members instead of being maintained in a central database. Hence, each peer maintains his or her own copy of the ledger. As soon as the ownership of a house is transferred from one person to another, all the ledgers of the system need to be updated in order to contain the latest version of reality. However, passing information forward among peers and updating the individual ledgers require time. Until the last member of the system receives the new information and updates his or her copy of the ledger, the system will not be consistent. Some peers already know about the latest transfer of ownership, while other peers have not yet received that information. The fact that not all ledgers have up-to-date information makes them prone to be exploited by anyone who already has the latest information.

The Double Spending Problem

- Let’s also imagine the following situation. Person A sells his house to person B. The transfer of ownership from A to B is documented in one of the ledgers in the peer-to-peer system. This particular ledger needs to inform other peers about this transfer, who in turn inform other peers as well, until eventually all peers learn about the transfer of ownership from A to B. However, suppose that person A quickly approaches another ledger of the system and demands to document a different transfer of ownership of the identical house: the sale from person A to person C. If this peer has not yet learned about the transfer of ownership from A to B that happened in the past, this peer will approve and document the transfer of ownership from A to C for the identical house. Hence, A was able to sell his house twice by exploiting the fact that distributing information about his first sell requires time. But B and C cannot own the house at the same time. Only one of them is supposed to be the new and lawful owner. Hence, the situation is called the double spending problem.

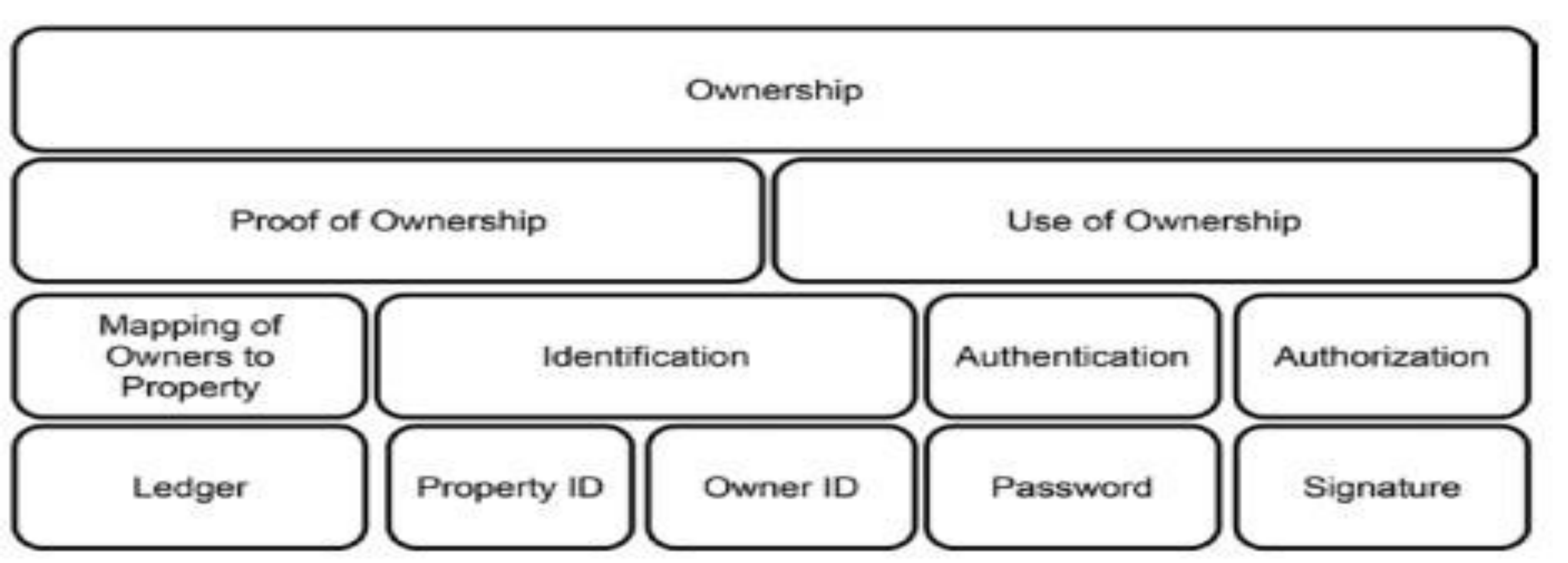
The Double Spending Problem

- Similar to the term blockchain, the term double spending is ambiguous as it is used to refer to the following concepts:
    - A problem caused by copying digital goods
    - A problem that may appear in distributed peer-to-peer systems of ledgers
    - An example of violated integrity in purely distributed peer-to-peer systems

Solving Double Spending as a Problem of Copying Digital Goods

- The problem of spending digital money or any other digital assets more than once just by copying the data is actually a problem related to the nature of ownership. Any accepted means of mapping data that represents digital goods to their owners will solve that problem, regardless of its specific implementation. Even a physical central book or (more realistically) an electronic ledger, regardless of its architecture (centralized or peer-to-peer), can ensure that a digital good will only be spent once, provided the ledger works correctly all the time.

Concepts of Ownership



Ownership

Proof of Ownership

Use of Ownership

Mapping of Owners to Property

Identification

Authentication

Authorization

Ledger

Property ID

Owner ID

Password

Signature

Concepts of Ownership

- The concepts on each layer can be seen as realizations of the concepts in layers above them. For example, the proof of ownership requires identification of owners and property alike as well as the mapping between owners and property. The use of ownership requires identification as well as authentication and authorization to ensure that only the legitimate person uses the property. The boxes in the very bottom row represent the implementation layer. They show, for example, that password and signature are concepts used to implement authentication and authorization. A ledger can be seen as a concrete implementation of a mapping between owners to their property.

Solving Double Spending as an Example of Violated Integrity in Distributed Peer-to-Peer Systems

- In thiscontext, the architecture of the system is specified but the application domain is leftunspecified. Hence, solutions on this level focus on achieving and maintaining integrity in distributed peer-to-peer systems, regardless of their concrete usage. However, the concrete usage of a distributed peer-to-peer system determines the meaning of integrity. For example, a simple file-sharing application may consider differentaspects for defining integrity as compared to a system that manages ownership in a digital currency. Hence, the question of whether the blockchain-technology-suite is the right tool for achieving and maintaining system integrity cannot be answered without knowledge of the specific application goals. Hence, it could be possible that in specific

Hashing Data

- Hash functions are small computer programs that transform any kind of data into a number of fixed lengths, regardless of the size of the input data. Hash functions only accept one piece of data at any given time as input and create a hash value based on the bits and bytes that make up the data. Hash values can have leading zeros in order to provide the required length. There are many different hash functions that differ among others with respect to the length of the hash value they produce. An important group of hash functions is called cryptographic hash functions, which create digital fingerprints for any kind of data.

Properties of hash values

Cryptographic hash functions have the following properties:

- Providing hash values for any kind of data quickly
- Being deterministic
- Being pseudorandom
- Being one-way functions
- Being collision resistant

Deterministic

- Deterministic means that the hash function yields identical hash values for identical input data. This means that any observed discrepancies of the hash values of data must be solely caused by the discrepancies of the input data and not by the internals of the hash function.

Pseudorandom

- Being pseudorandom means that the hash value returned by a hash function changes unpredictably when the input data are changed. Even if the input data were changed only a little bit, the resulting hash value will differ unpredictably. To put it differently, the hash value of changed data must always be a surprise. It should not be possible to predict the hash value based on the input data.

One-Way Function

- A one-way function does not provide any way to trace its input values by its outputs. Hence, being a one-way function means that it cannot be used the other way around. To put it differently, it is impossible to recover the original input data based on the hash value. This means that hash values do not tell you anything about the content of the input data in the same way as an isolated fingerprint does not tell you anything about the person whose finger created it. One-way functions are also said to be noninvertible.

Collision Resistant

- A hash function is called collision resistant if it is very hard to find two or more distinct pieces of data for which it yields the identical hash value. Or, to put it differently, if the chance to receive an identical hash value for distinct pieces of data is small, then the hash function is collision resistant. In this case, you can consider the hash values created by the hash function as being unique and hence being usable to identify data. If you obtained an identical hash value for different pieces of data, you would face a hash collision. A hash collision is the digital equivalent to having two people with identical fingerprints. Being collision resistant is mandatory for hash values to be usable as digital fingerprints. How collision resistant hash functions work internally is beyond the scope of this book, but you can be assured that huge effort has been spent on reducing their risk to produce hash collisions.

Hash generator

- http://www.blockchain-basics.com/HashFunctions.html

Tasks to develop blockchain

- Describing ownership
- Protecting ownership
- Storing transaction data
- Preparing ledgers to be distributed in an untrustworthy environment
- Distributing the ledgers
- Adding new transaction to the ledgers
- Deciding which ledgers represents the truth

Task 1: Describing Ownership

- Before you can start developing the blockchain, you need to ask yourself what you want to do with it. Since you will want to design a software system that manages ownership, you have to decide how to describe ownership first. It turns out that transactions are a good way to describe any transfer of ownership, and the complete history of transactions is the key to identifying the current owners.

Task 2: Protecting Ownership

- Describing ownership by using transactions is just the starting point. Moreover, you need a way to prevent people from accessing the property of others. In real life, you can easily prevent people from using your car or from entering your house by using doors with locks. It turns out that cryptography provides a way to protect transactions on an individual level, similar to the way doors with locks protect your individual car or house.

Task 3: Storing Transaction Data

- Describing ownership by means of transactions and having security measures that protect ownership on the level of individual transactions are important steps toward the goal of designing a software system that manages ownership. However, you need a way to store the whole history of transactions, as this history is used to clarify ownership. Since the transaction history is the core element in clarifying ownership, it must be stored in a secure way. It turns out that the blockchain-data-structure is the digital equivalent to a ledger.

Task 4: Preparing Ledgers to Be Distributed in an Untrustworthy Environment

- The best way to prevent the transaction history from being changed is to make it unchangeable. This means the ledgers and therefore the transaction history cannot be changed once written. As a result, you will not have to fear that the ledgers will be tampered with or forged because they cannot be changed in the first place. However, having a distributed peer-to-peer system of ledgers that can never be changed sounds like a very secure but pretty useless thing because it will not allow you to add new transactions. Hence, the challenge of the blockchain-data-structure is to be unchangeable, on the one hand, while accepting new transactions being added to it, on the other hand.

Task 5: Distributing the Ledgers

- Once the ledger is append-only, you can create a distributed peer-to-peer system of ledgers by making copies of it available to everyone who asks for it. However, just providing copies of append-only ledgers does not fulfill your goals. A distributed system that manages ownership involves interaction between the peers or nodes, respectively.

Task 6: Adding New Transactions to the Ledgers

- The distributed peer-to-peer system will consist of members whose computers maintain individual copies of an append-only blockchain-data-structure. Since the data structure allows you to add new transaction data, you will have to ensure that only valid and authorized transactions are added. It turns out that this is possible by allowing all members of the peer-to-peer system to add new data and additionally turning each member of the peer-to-peer system into supervisors of their peers.

Task 7: Deciding Which Ledgers Represent the Truth

- Since the transaction history is the basis for identifying lawful owners, having different conflicting transaction histories is a serious threat to the integrity of the system. Hence, it is important to find a way either to prevent the emergence of different transaction histories in the first place or to find a way to decide which transaction history represents the truth. Due to the nature of a purely distributed peer-to-peer system, the former approach is not possible. As a result, you need a criterion for how to find and choose one transaction history that represents the truth. But there is another problem: there is no central authority in a purely distributed peer-to-peer system that can declare which transaction history has to be chosen. It turns out that one can solve that problem by making every node in the peer-to-peer system decide on its own which transaction history represents the truth in a way that the majority of the peers independently agree on that decision.

How it works?

- The procedure to add a new block to the blockchain-data-structure, is not computationally expensive because it only requires adding the hash reference that points to the current head of the chain to the new block header and declaring it as the new head of the chain. The challenge of making the blockchain-data-structure immutable is to make adding a new block a computationally expensive task.

The following aspects need to be considered in the course of achieving this:

- Compulsory data of block headers
- The process of creating a new block header
- Validation rules for block headers

Compulsory Data

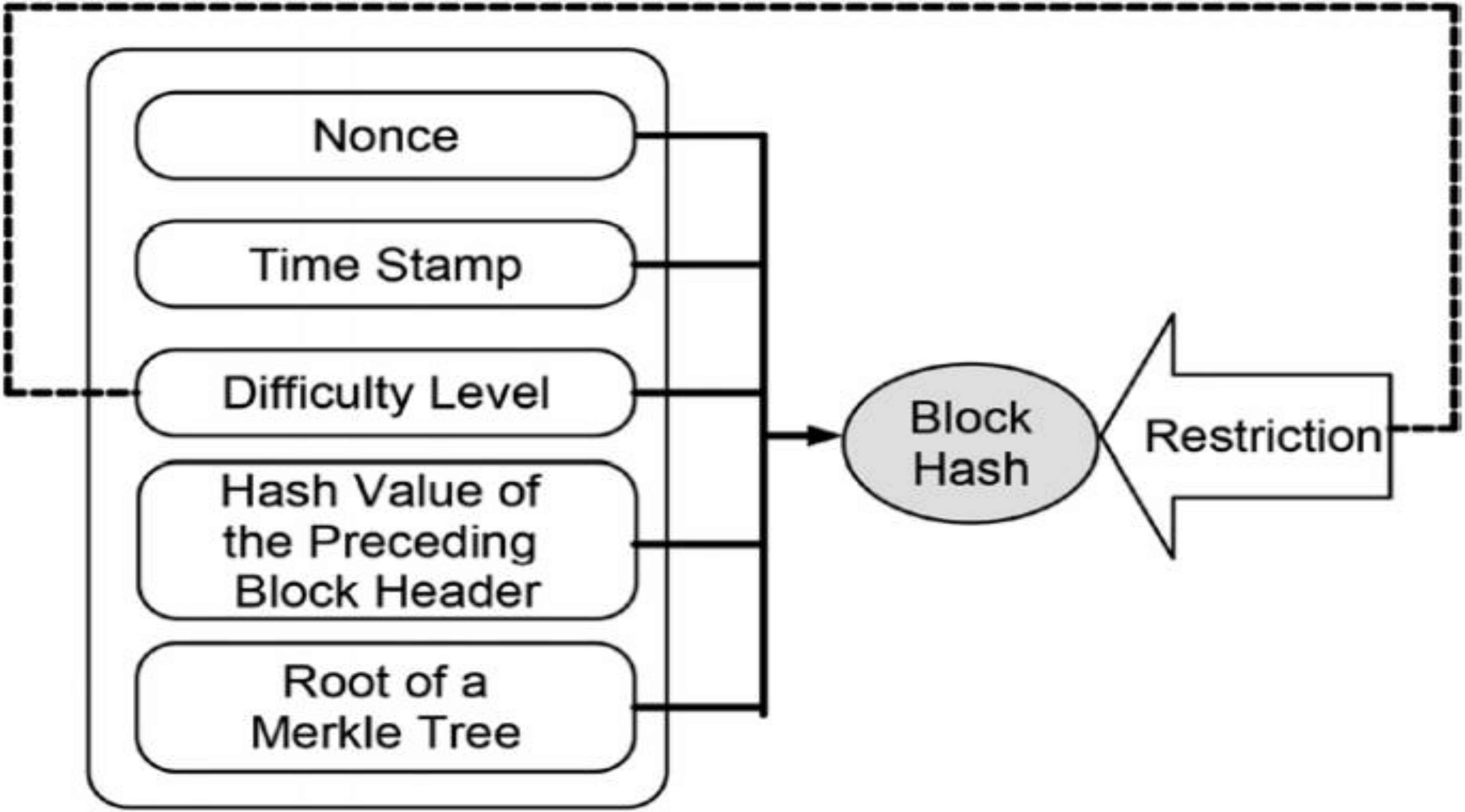
Every block header of the blockchain-data-structure has to carry at least the following data:

- The root of a Merkle tree containing transaction data
- A hash reference to the header of the preceding block
- The difficulty level of the hash puzzle
- The time when solving the hash puzzle started
- The nonce that solves the hash puzzle

The Process of Creating A New Block

Creating a new block involves the following steps:

- Get the root of the Merkle tree that contains the transaction data to be added.
- Create a hash reference to the header of that block that will be the predecessor from the new block header’s point of view.
- Obtain the required difficulty level.
- Get the current time.
- Create a preliminary block header that contains thedata mentioned in points 1 to 4.
- Solve the hash puzzle for the preliminary block header.
- Finish thenew block by addingthenonce that solves thehash puzzle to the preliminary header


Nonce
Time Stamp
Difficulty Level
Hash Value of the Preceding Block Header
Root of a Merkle Tree
Block Hash
Restriction

Validation Rules

Every block header of has to fulfill the following rules:

- It must contain a valid hash reference to a previous block.
- It must contain a valid root of a Merkle tree containing transaction data.
- It must contain a correct difficulty level.
- Its time stamp is after the time stamp of its preceding block header.
- It must contain a nonce.
- The hash value of all the five pieces of data combined together fulfills the difficulty level.

The validation rules ensure that only those blocks are added to the blockchain-data-structure for which the hash puzzle was solved and the computational costs were paid.

Why It Works?

- The blockchain-data-structure makes any change of its data stand out due to the fragility of the hash references with respect to changes of the data being referred. This causes the need to rewrite all blocks that are affected by a manipulation. The hash puzzle causes costs for every block that needs to be rewritten in the course of embedding a manipulation. The accumulated costs of rewriting the blockchain-data-structure in the course of embedding a manipulation make it unattractive to manipulate the transaction history in the first place. As a result, the blockchain-data-structure becomes an immutable append-only data store.

The Costs of Manipulating the Blockchain Data-Structure

Let’s assume we were going to try to manipulate a particular piece of transaction data that is part of a Merkle tree whose root belongs to a block header located 20 blocks below the current head of the blockchain-data-structure. Embedding the manipulated transaction data requires the following work:

- Rewrite the Merkle tree to which the manipulated transaction belongs.
- Rewrite the block header to which the root of the rewritten Merkle tree belongs.
- Rewrite all succeeding block headers up to the head of the blockchain-data-structure.

The Costs of Manipulating the Blockchain Data-Structure

- Point 2 requires the solution of a hash puzzle because changing the Merkle root changes the hash value of the block header and hence the solution of its hash puzzle. Point 3 requires solving 20 hash puzzles due to successive changes of the hash references to the previous block header. Under the assumption that solving a hash puzzle takes on average 10 minutes, we would need in total 210 minutes to embed a manipulation in a transaction that belongs to a block header located 20 blocks below the current head. These huge costs deter nodes from changing the blockchain-data-structure.

Is software code mature enough to replace the law?

- In a distributed ledger technology environment, smart contracts are agreed based on a software code and on the agreed date executed (sometimes mercilessly) as the contract itself is the law. Although this unalterable nature (or immutability) is the core strength of this technology and enhances trust amongst parties, it also needs to be mature enough to replace the law.
- There have been instances in the past when some of the well-known DLTs had to be "hard forked" – a phenomenon whereby the governing code has to be replaced with a new one. In 2016, for example, Ethereum had to be hard forked after long debate amongst the community as an unexpected code path allowed users to withdraw funds and an unknown user

Standards are underdeveloped and not mature yet

- Being at a stage of rapid technological development, there are no mature standards addressing distributed ledger technology yet. At this point, there are various competing proprietary and community-managed platforms and frameworks. The absence of international standards carries risks related to customer lock-in, lack of interoperability, privacy and security.
- There are international efforts ongoing in these areas, including ISO Technical Committee 307 on Blockchain and Distributed Ledger Technologies and work in ITU's standardization sector ITU-T.

Energy requirement can be high

- A methodology to build consensus for entering a new data block amongst participating nodes is a core feature of blockchain. There exist several possible ways of reaching consensus, each with its own advantages and disadvantages.
- The one that is employed by Bitcoin and Ethereum, the most famous of blockchain implementations, is proof-of-work (PoW). It works on the principle of “hard to create, easy to verify”, which means lot of energy needs to be spent by the node to earn incentive tokens. For a large chain like Bitcoin, estimates suggest data size exceeding 100 gigabytes and electricity requirements more than the entire country of Ireland.
- Although this is true for the PoW methodology, other alternatives such as proof-of-stake (PoS), Byzantine fault tolerance algorithm, and delegated proof-of-stake model require less energy. However, they come with their own disadvantages, for example in the case of PoS, users with more stakes will have greater control on decision-making.

Trusting the blockchain developers and managers

- A very high level of trust is placed on the developers and managers of the blockchain. It is a new technology where a large number of entities are innovating to create solutions. The focuses, owners and software implementations vary.
- Implementations of these technologies are largely dependent on the community of developers backing the project or the owner. A decision to soft fork or hard fork a project, or to change the cryptography algorithm, will be driven by the nodes and participants in the blockchain. These decisions are driven by codes that govern the consensus and the community developing it.
- At the same time, it is important to build resilience into the networks so

Increased responsibility on the user

- By its very design, blockchain implementation does not have a central authority – at least in the case of public blockchains such as Bitcoin – which puts additional responsibility on the user. There is no entity to go to in the event of individuals losing private keys (or incurring losses as a result of revealing a private key).
- Also, there is no feature to restore forgotten passwords and usernames that individuals are used to. Individuals need to exercise great caution, just as on the Internet, before publishing anything. The importance of entering the correct data is very important too as it is very difficult to make corrections later.

Implementing data privacy legislation

- Data protection and privacy is a major concern and initiatives to prevent their abuse are being taken by countries and regions (e.g. Association of Southeast Asian Nations (ASEAN), European Union General Data Protection Regulation (EU GDPR).
- For example, the EU GDPR has instituted the “right to forget” whereas the design of DLTs is oriented towards “never to forget”. Although there is a possibility of keeping identification unknown in the system, it raises security concerns largely in relation to anti-money laundering (AML) activities and know your customer (KYC) requirements.

Policy and regulatory risks

- The policy and regulatory framework around blockchain is in its infancy and therefore entails high risks. The fluctuations in the price of Bitcoin and the reports of hacking of cryptocurrency have resulted in increased regulation by a number of countries and has attracted regulatory interest.
- These regulationsvary from a complete ban on holding cryptocurrency (e.g. Bangladesh), a ban or regulation on cryptocurrency trading (China, Saudi Arabia) to a ban on holding initial coin offerings (ICOs). A number of blockchain projects, especially those dealingwith currency or cross-border transactions, requires KYC compliance and it is important to understand the national framework before delving into these projects.
- At the same time, governments see DLTs as a high potential technology

Speed of transactions

- The speed of transaction is an important element as some of the public blockchains do not have high transaction speeds. On Bitcoin blockchain, a new block emerges on average every ten minutes but is not guaranteed; and this block time is different for every blockchain.
- For scalability, it is important to understand the requirement of applications in terms of speed (transactions per second (tps)) before choosing a solution. Theoretically, Visa network can handle about 50,000 tps, which is a lot more than is offered by most mature blockchains today.

Malicious users

- In the absence of identification of a third party, the system is prone to risks from malicious users in systems that are pseudonymous, that is with no requirement to disclose identity.
- Although DLTs are designed to disincentivize malicious intent, there can be situations where malicious users have greater incentives to game the system and at least cause harm in the short term and may call for a hard fork. These situations are more likely where they gain greater control of the system.

Identity and security

- Public blockchains carry out transactions based on the public and private key of the individual and do not keep the mapping of the identity with the key. This raises security constraints for the law enforcers and applications where identity is important.
- In contrast, there are privacy concerns in disclosing identity on permission less blockchains that require data to be public facing and transaction histories to be disclosed. Most DLTs use encryption algorithms that are hard to break by normal non-quantum computers.
- Going forward, where quantum computing (relying on cubits rather than bits) gains momentum and enhances computing powers, these encryptions are not secure enough. There have been a large number of successful attacks on DLTs and there are security risks associated with DLTs30 (e.g. blockchain attacks, phishing, malware, cryptojacking, endpoint miners implementation vulnerabilities, wallet theft