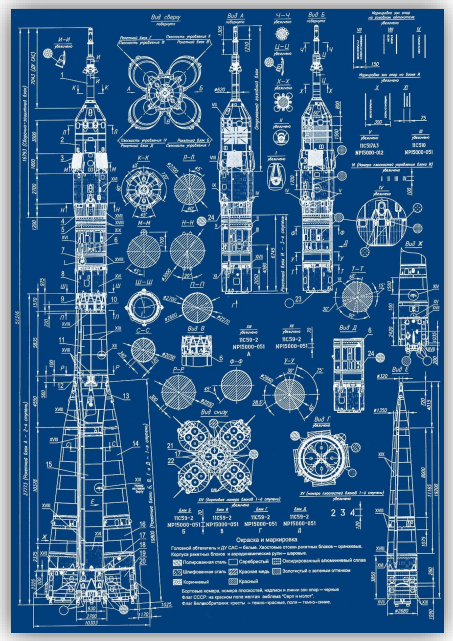
COURSE REVIEW

BCDV 1001 – Introduction to Blockchain (and Blockchain Architecture)

Summary

- Course Evaluation
- Who should take this course
- Assignment & Use Cases
- What is covered vs. no covered in this course
- Myth vs. Reality
- Idea of Blockchain and 3rd party
- Problem Solved
- Byzantine General's Problem and PoW
- Centralized vs. Distributed vs. Decentralized
- Blockchain 1.0 2.0 3.0
- Transaction process explained from 100,000 ft – 10,000 ft – 1,000ft
- Smart Contracts
- Use Case Sample
- Merkel Root
- Difficulty target/level
- Hash (SHA-256)
- Hashing Power
- Mem-pool
- Blockchain Header
- Mining
- nonce
- Ethereum
- Ether (ETH)
- ERC tokens
- Blockchain Benefits
- Types of blockchains
    - Public vs. Private vs. Consortium
    - Open vs. Closed
    - Permissioned vs. Permissionless
- Blockchain Layers
- Consensus Mechanisms
- Scalability
- Blockchain Trilemma
- Data and blockchain
- Side chains / Off-chain
- Cost of data
- Oracles and Smart contracts
- Types of Oracles
- Database on blockchain
- Oracles and Blockchain
- Ethereum 2.0
- Shreding
- Proof of Stake
- Hyperledger
- Hyperledger Fabric
- Hyperledger Sawtooth



EVALUATION SYSTEM

- Participation: (class & updates) **10%**
- Lab Exercises: 5 @ 3% each >> total of **15%**
- Lab Tests: 5 @ 3% each >> total of **15%**
- Assignment: (groups of 2, 3 up to 4): **30%**
- Final Exam: **30%**

- Important to attend all classes as big chunk of your mark is the assignment and the final exam, which we work on and prepare each class.
- 4-week course consists of six 4-hour sessions (last two classes reserved for your assignment presentation and final exam).

Who has completed this course…

- Programmers | Engineers | Human Resources | Criminology | Teachers | Computer Scientists | Government employees | Banking and Financial Industry | Architects | Construction | Entrepreneurs | +more

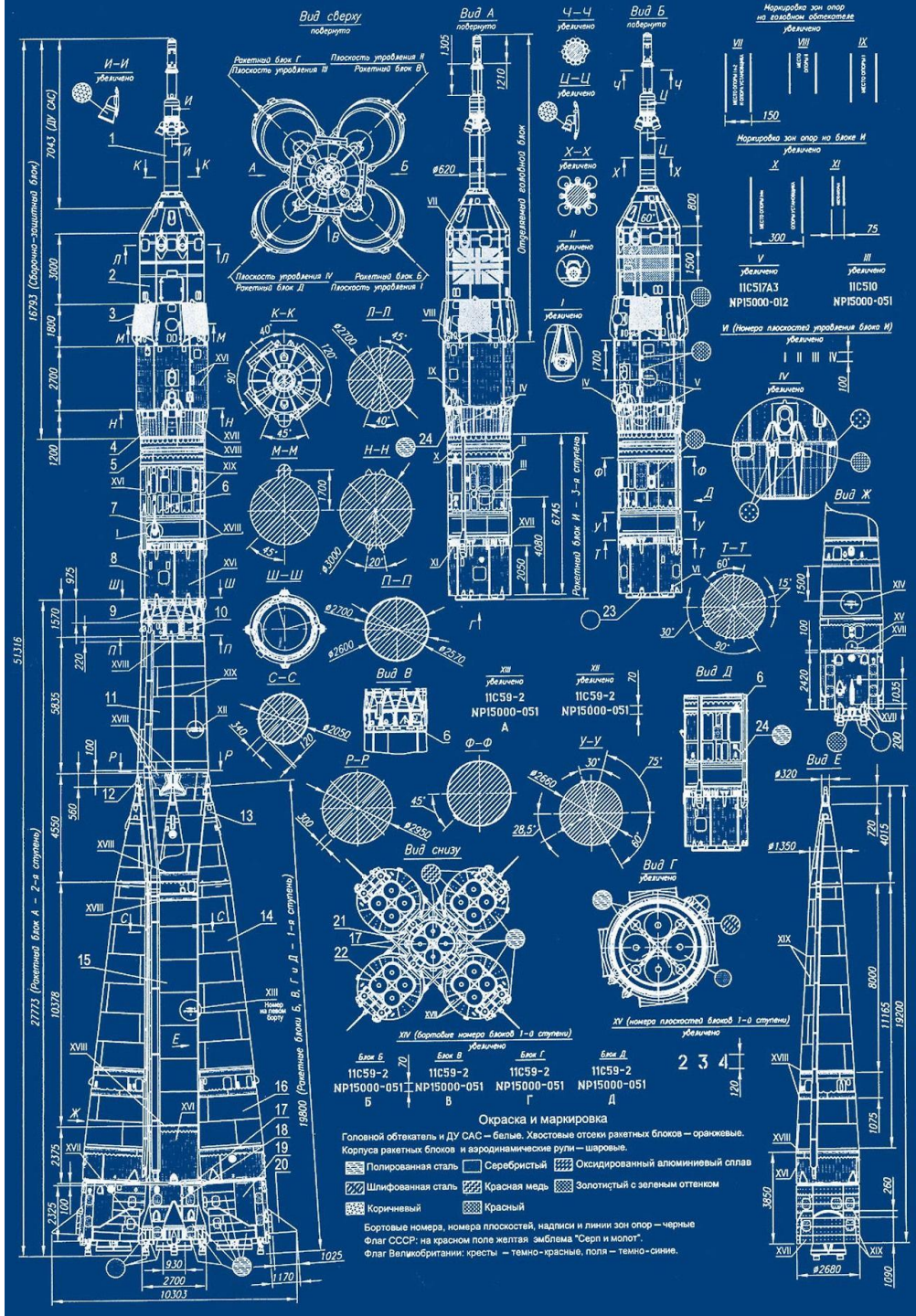
- From:

  Canada | USA | Russia | United Kingdom | France | Italy | Germany | China | Japan | South Korea | Brazil | Ukraine | Serbia | Singapore | Saudi Arabia | Pakistan | Turkey | Iran | Palestine | India | Mexico | Ecuador | Trinidad and Tobago | Philippines | Bangladesh | Zambia | Barbados | Armenia | +more

***Point to be made** – blockchain will impact every profession everywhere around the world, thus people that take this course come from diverse professional and cultural background. Getting ready to what is to come!*

What will do is…

- Explain Blockchain…
- Find a problem and build a solution….
- Break blockchain into building blocks…
- Work on Blockchain Architecture…
- Look under the hood, and take things apart



Use Cases

[pick one per group]

SUPPLY CHAIN	Organic foods ◆ Clothing Industry – combating sweat shops and child labor ◆ Sustainable/eco friendly products and ingredients
IDENTITY	Solution for refugees ◆ Solution for combating sex slave trade ◆ Solution for individual to retain control of personal data
GAMING	Tradable tokenized assets ◆ Character achievements ◆ Upgradable digital content ◆ A "multiverse" of cross-platform content, characters, and worlds
DRM/DECENTRALIZED MARKETPLACES	Art ◆ Music ◆ Publishing
DIRECT DEMOCRACY PLATFORM	Voting on community issues ◆ Transparency for government spending
LAND REGISTRY	Transparent record of deed and land patent ownership
FINANCIAL	Social impact investment platform
HEALTH	Patient data ◆ Prescriptions
LOCATION and AUTOMATION	Tracking and Logistics
other	Please see the instructor for advice and consultation

INTRODUCTIONS

Be prepared to answer following questions:

- Where were you in 2009… when did you first time hear about it and by whom?
- **Past life** >> Educational and Professional experience…
- **Current life** >> Why am I involved now with blockchain?

Not Covered in course but could be mentioned

- Regulatory & Legal advice
- Financial and Investment advice
- Political and social-economic views

*This program is tailored for **developers** but many have completed the programme and went on to have successful carriers in the blockchain industry as non-developers – this being said we will cover as much as possible beside already mentioned blockchain development.*

Myth vs. Reality

- Will Governments will disappear…
- Will Banks bankrupt…
- No more need for lawyers…
- Satoshi Nakamoto…

Blockchain

[Quick Recap]

- Five common blockchain myths create misconceptions about the advantages and limitations of the technology

- bitcoin is not blockchain…
- **B**itcoin vs. **b**itcoin ???
- Blockchain is like hot sauce…
- Is it legal?

	Myth	Reality	
1	Blockchain is Bitcoin	Bitcoin is just one crypto-currency application of blockchain	Blockchain technology can be used and configured for many other applications
2	Blockchain is better than traditional databases	Blockchain's advantages come with significant technical trade-offs that mean traditional databases often still perform better	Blockchain is particularly valuable in low-trust environments where participants can't trade directly or lack an intermediary
3	Blockchain is immutable or tamper-proof	Blockchain data structure is append only, so data can't be removed	Blockchain could be tampered with if >50% of the network-computing power is controlled and all previous transactions are rewritten—which is largely impractical
4	Blockchain is 100% secure	Blockchain uses immutable data structures, such as protected cryptography	Overall blockchain system security depends on the adjacent applications—which have been attacked and breached
5	Blockchain is a "truth machine"	Blockchain can verify all transactions and data entirely contained on and native to blockchain (eg, Bitcoin)	Blockchain cannot assess whether an external input is accurate or "truthful"—this applies to all off-chain assets and data digitally represented on blockchain

McKinsey&Company

One Step Back – 99 Forward…

- What is disruptive? Which tech is disruptive now or in the past?
- Is Distributed Ledger technology (DLT) same as Blockchain…
- Why blockchain?
- What is bitcoin, isn't it blockchain or blockchain is bitcoin?

The New Idea? Not really…

[economic point of view]

- Magazine - The Economist, publishes a paper in Jan 9, 1988; "**Get Ready for the New World Economy**";
  - *"THIRTY years from now, Americans, Japanese, Europeans, and people in many other rich countries, and some relatively poor ones will probably be paying for their shopping with the same currency."*
  - *"In time … its value against national currencies would cease to matter, because people would choose it for its convenience and the stability of its purchasing power."*
  - *"As telecommunications technology continues to advance, these transactions will be cheaper and faster still."*

- Friedrich Hayek (Nobel Prize in Economics), 1984 [link]
- Milton Friedman (Nobel Prize in Economics), 1999 [link]
- David Chaum (DigiCash), Wei Dai (B-money), Nick Szabo (Bit Gold)
- Last 25+ years there have been multiple attempts to solve what bitcoin did

BLOCKCHAIN

- Is it a new idea, a new tech? Bitcoin combines 3 distinct technologies:
  - **Game theory** (incentive to secure a network)
  - **P2P** (file sharing of a common ledger)
  - **Encryption**/Cryptography
- Blockchain 1.0 – the **killer app** called **bitcoin** – electronic money, also known as **crypto**.
- What is the problem that bitcoin solves and what did Satoshi Nakamoto succeed in? **bitcoin solves a mathematical problem**; the **double spend problem (reversibility of electronic payments) in a electronic network without a 3rd party (created decentralized system). It secures ownership of an digital asset via a distributed computation.**
- Why is it important? **Eliminates 3rd party**.

source: 'Mastering Bitcoin – programming the Open Blockchain' by Andreas M. Antonopoulos

Problems solved – the big solution

- A practical novel solution to a Distributed Computing Problem – known as "**The Byzantine Generals' Problem**" (PoW) – a mathematical problem solved;
- Uses **concept of Proof-of-Work to achieve a consensus without a central trusted authority**, represents a breakthrough in distributed computing and has wide applicability beyond currency;
- It can be used to achieve consensus on decentralized networks **to prove the fairness** of elections, lotteries, asset registries, digital notarization, and more.

BYZANTINE GENERALS PROBLEM

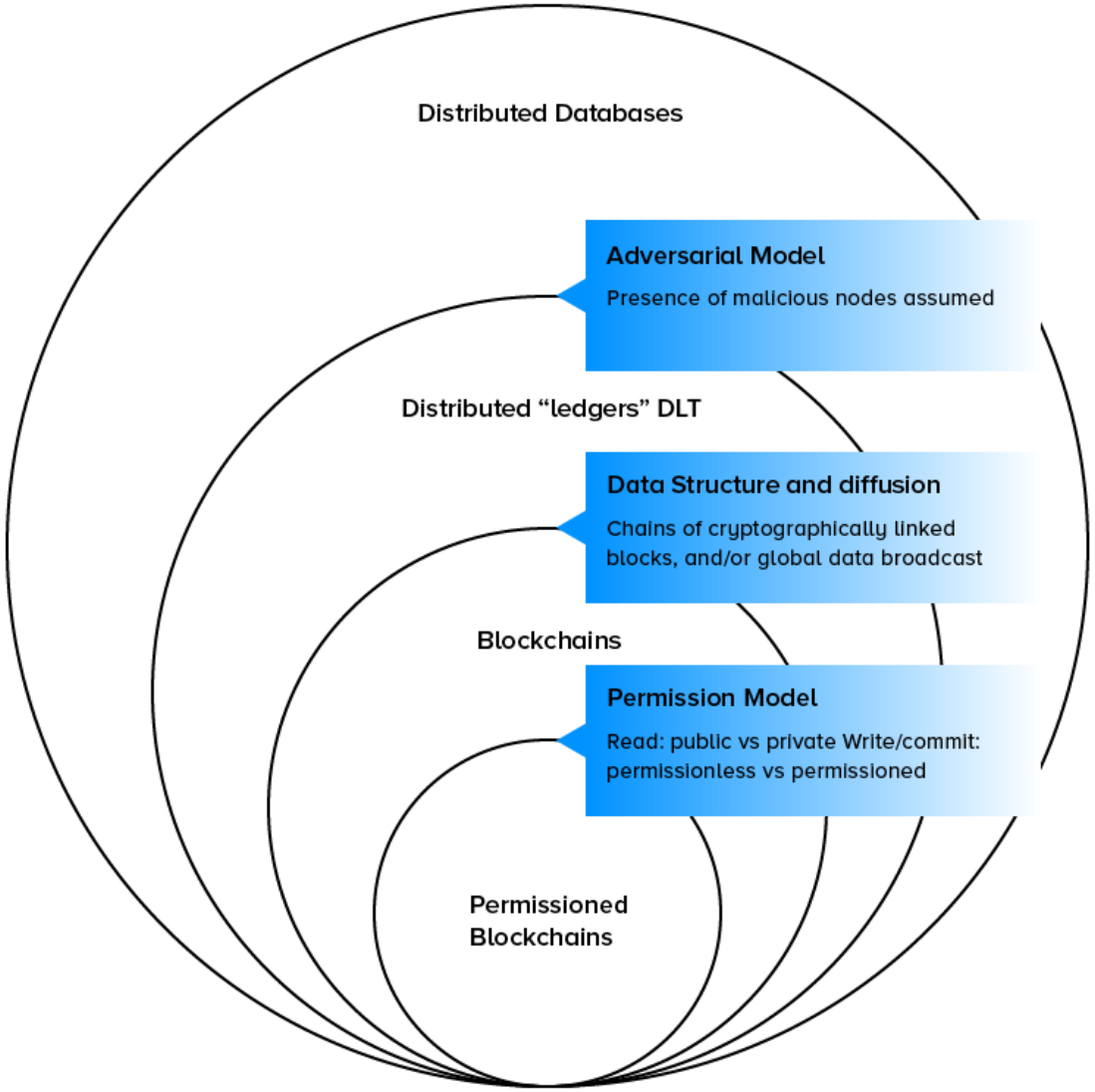
[BYZANTINE FAULT TOLERANCE / BFT]

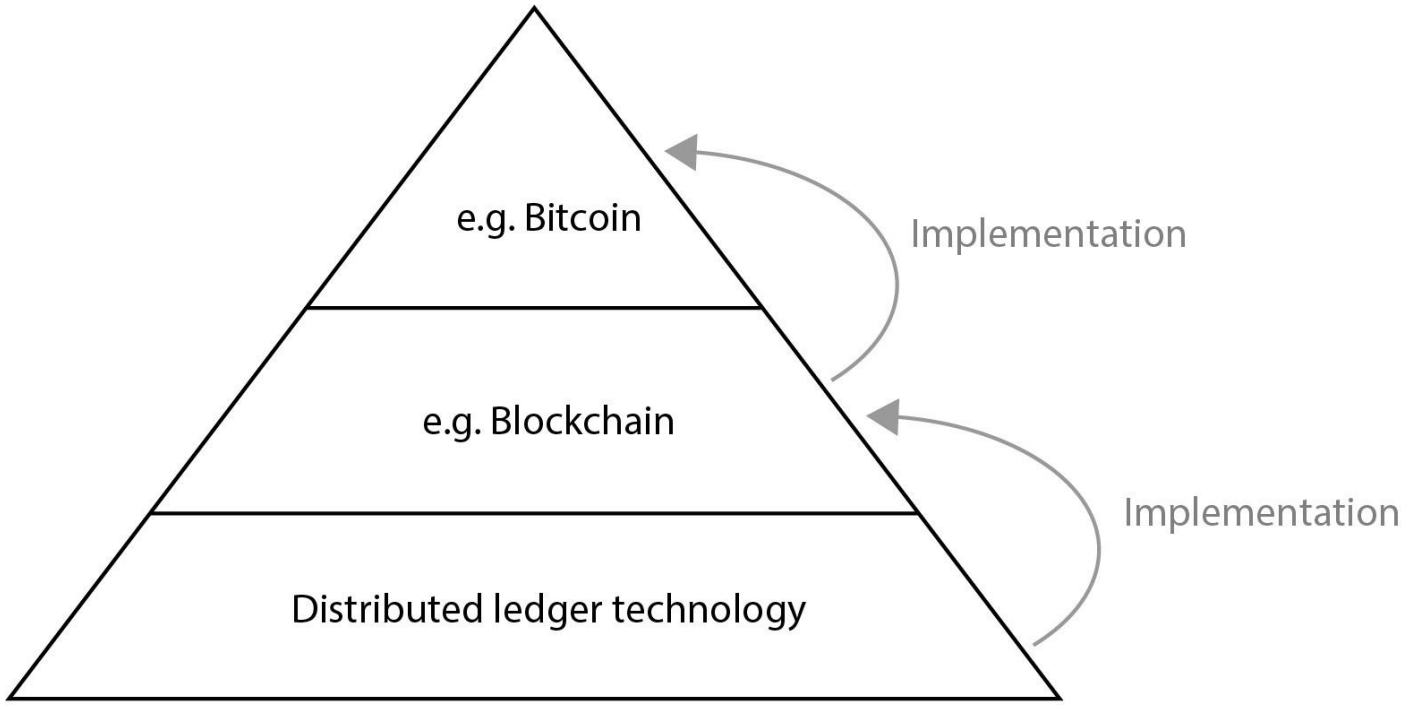
- Byzantine Fault Tolerance Explained [link]
- Marko Vukolic – PoW vs BFT [link]

- If *n* number of Byzantine generals are attacking an enemy city, then what is the maximum number of traitors that can be tolerated and the battle be won?

- If *n* number of nodes are misbehaving (fraudulent, hacked, etc.) towards a ledger, then what is the maximum number of different results and there is still a consensus on what is the true record on the ledger?

DLT vs. Blockchain



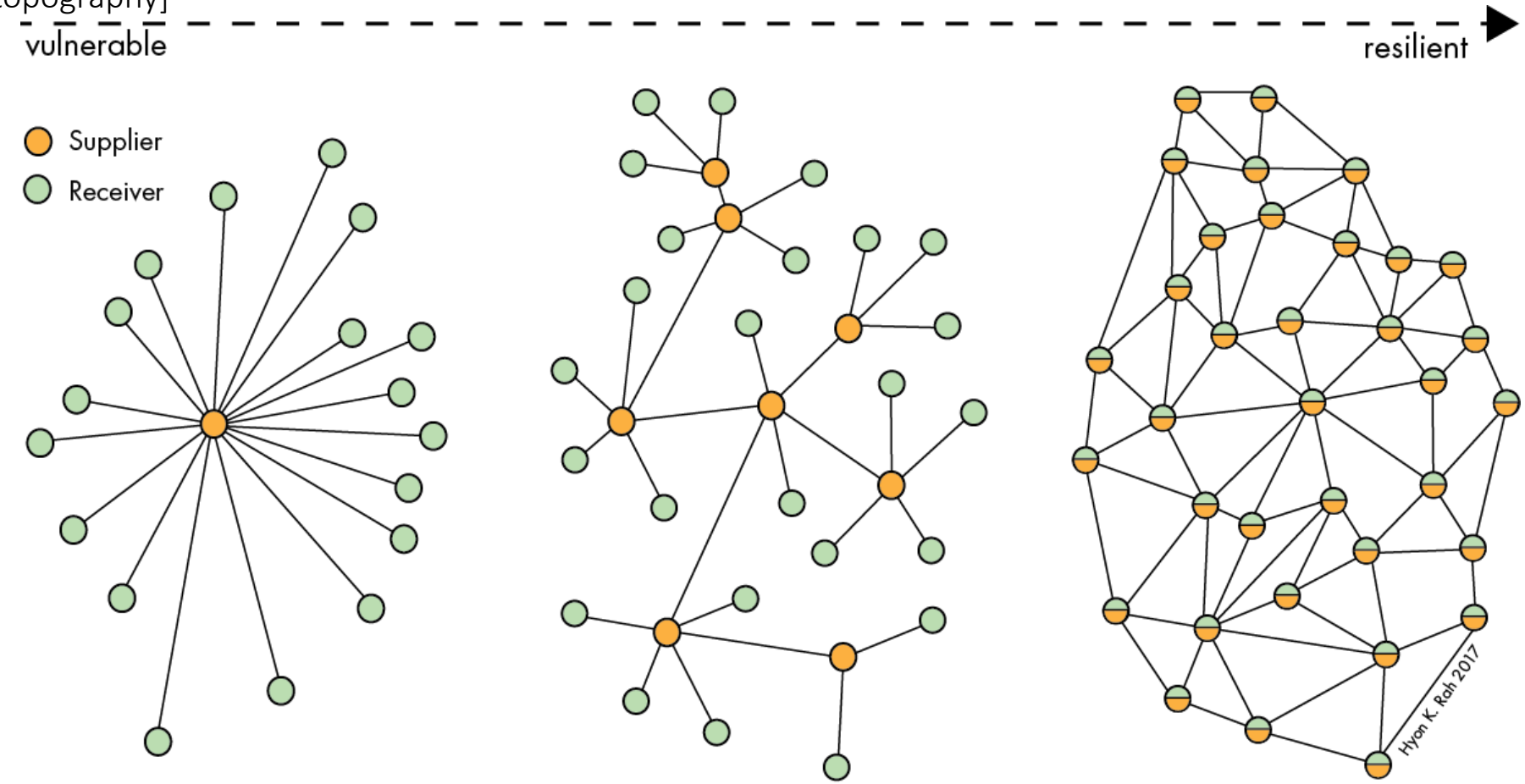


*Blockchain is implementation of DLT, just as email is implementation of internet.

If email was the killer app of internet, what is the killer app of blockchain, and does it exist yet?

Centralized vs Distributed vs Decentralized

[network topography]



A. Centralized System

Failure of the single supplier leads to the failure of the entire system

B. Decentralized System

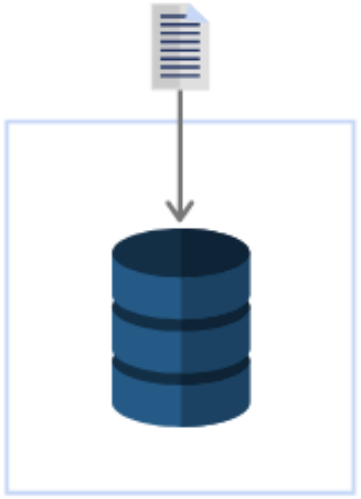
Failure of one supplier leads to localized failure; failure of all suppliers leads to system failure

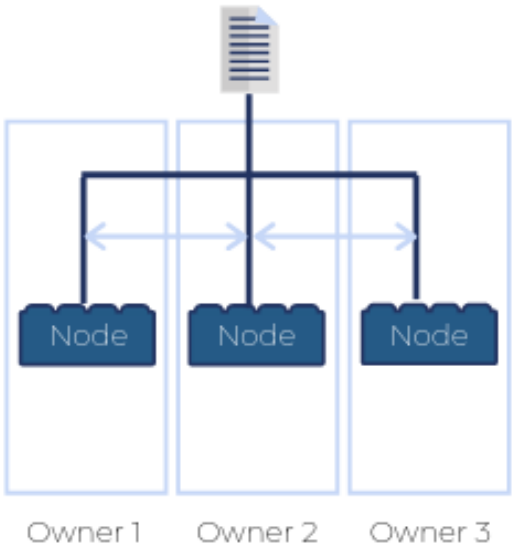
C. Distributed System

Every node can supply to and receive from the other nodes; system failure cannot occur unless all nodes fail

Centralized vs Distributed vs Decentralized

[ledger]





Centralized

- One database, one owner
- Example: Excel Spreadsheet
- Data was lost if owner left company or machine broke

Distributed

- Many database copies, but still one owner
- Example: Cloud Computing
- Is resilient to technical failure, but not to organizational

Decentralized

- Many database copies, everyone in the blockchain is an owner, no master
- Example: Blockchain
- Can withstand technical and organizational failure

Centralized vs Distributed vs Decentralized

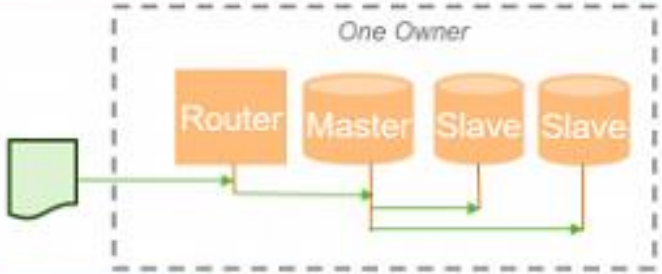
[ledger]

Centralized

- One database, one owner
- Not resilient to organizational failure
- Not resilient to technical failure

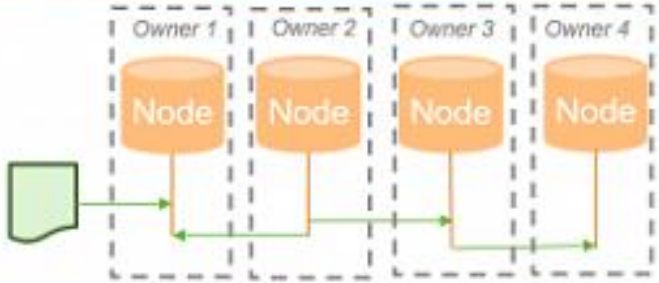
Distributed

- Many database copies, one owner
- Resilient to technical failure
- Not resilient to organizational failure



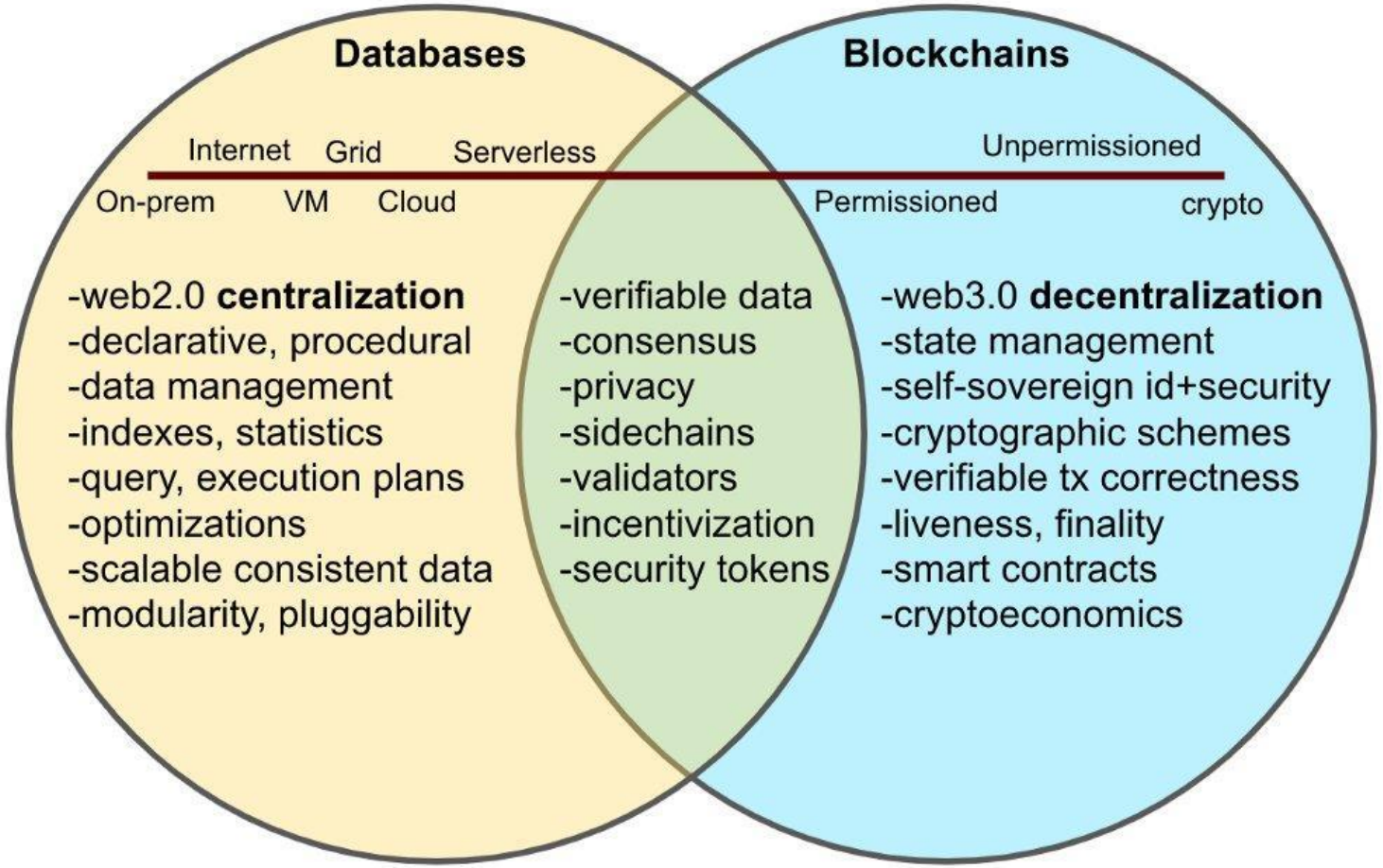
Decentralized

- Many database copies, many owners, no one "master"
- Resilient to technical failure
- Resilient to organizational failure



So what is Blockchain/Bitcoin?

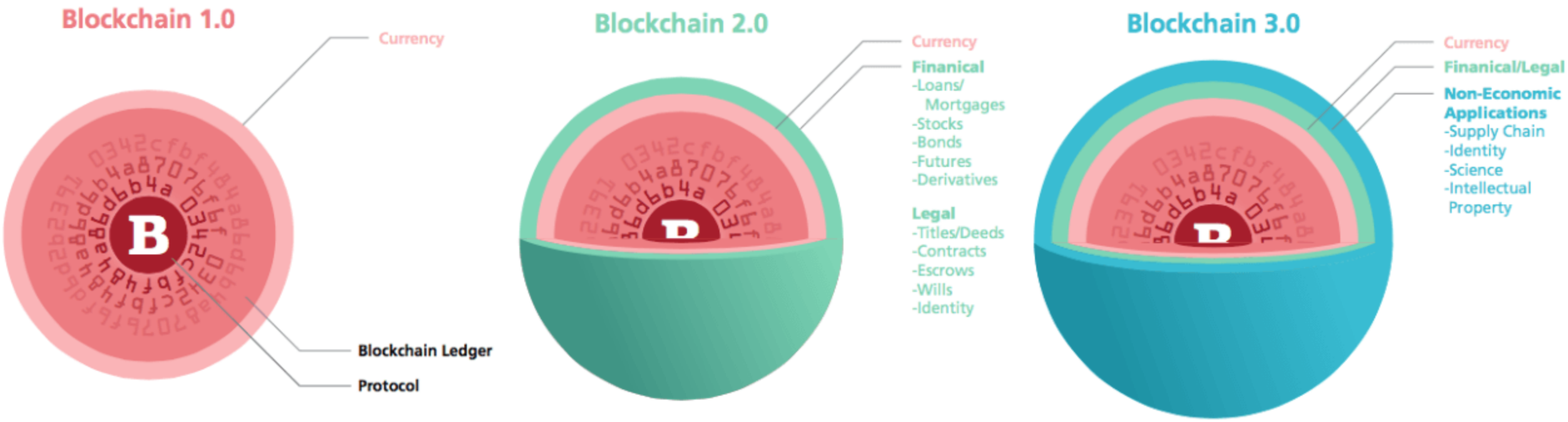
- Network Protocol
- Currency
- Database/Ledger



Bitcoin

- Consists of:
  - A **decentralized peer-to-peer network** (*the bitcoin protocol*);
  - A public transaction **ledger** (*the blockchain*);
  - A **set of rules** for independent transaction validation and currency issuance (*consensus rules*);
  - A **mechanism** for reaching a global **decentralized consensus** on the valid blockchain (*Proof-of-Work*) algorithm;

Blockchain Evolution



Blockchain Evolution (2009 – present)

2009 Bitcoin		• A hard-coded **cryptocurrency application** with limited stack-based scripting language; • Proof-of-Work consensus; • Native cryptocurrency (BTC); • **Permisionless blockchain** system;	Blockchain 1.0
2014 Ethereum		• Distributed applications (**smart contracts**) in a domain-specific language (Solidity); • Prof-of-Work consensus (transition to Proof-of-Stake); • Native cryptocurrency (ETH); • **Permisionless blockchain** system	Blockchain 2.0
2017 Hyperledger Fabric		• **Distributed applications** (chaincodes) if different general-purpose languages (e.g. golang, Java, Node); • Modular/pluggable consensus; • **No native cryptocurrency**; • **Permissioned blockchain** system;	Blockchain 3.0

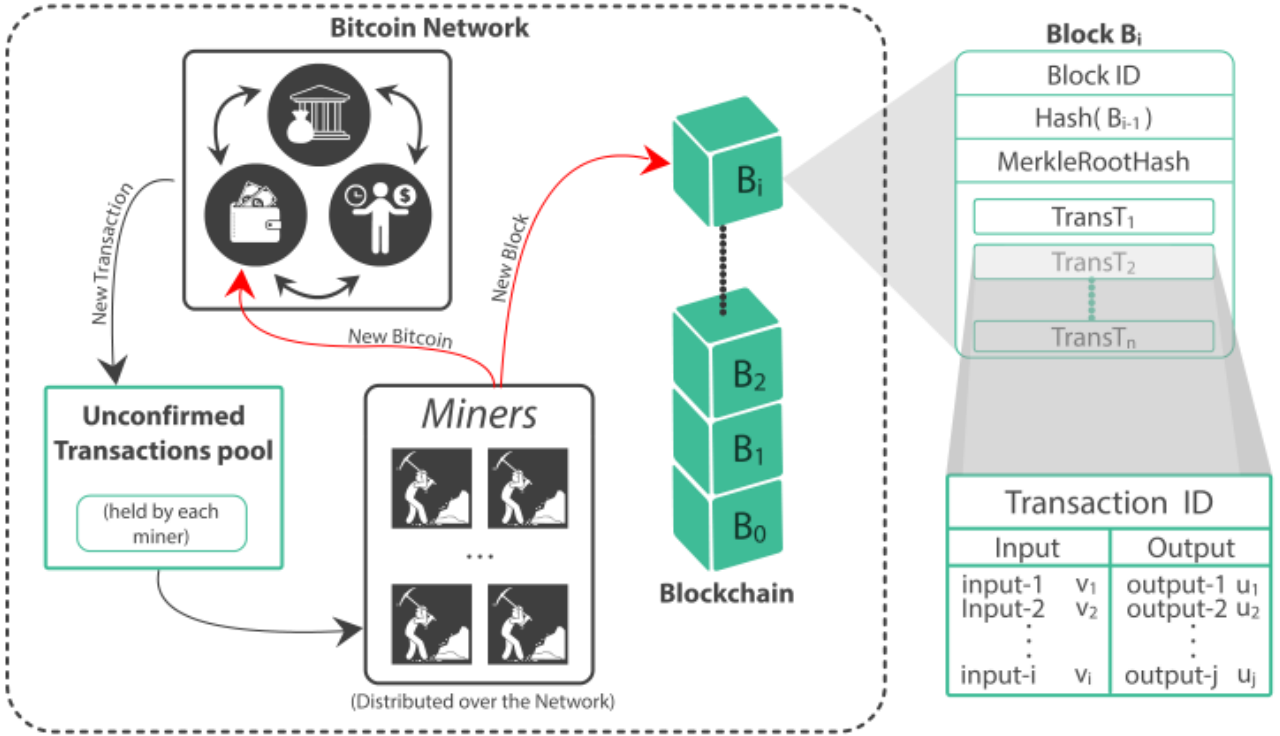
‘Simple’ Block Creation

- Step that a transaction goes through in a Bitcoin network…
- What is a block made of…
- How is a block created and linked with the rest of the chain of blocks…
- What does it involve and what are the other innovations and discoveries used to achieve the blockchain technology…

Simplified explanation

[100,000 ft view from Space]

- The bitcoin transaction
- Creation of a block
- Block consists of many parts not only transaction
- Mining
- Consensus
- Block created, added, linked



10,000 ft view

[a simplified 'simple English' approach]

- John wants to send some money (digital currency) to Anna electronically.
- John uses a digital wallet app to transfer the money to Anna. The app stamps the transaction with John's digital signature. This transaction now need to be verified by the blockchain.
- John's transaction is grouped with other transactions that occur at the same time into a block. The block has a unique ID, the transaction time and the ID of the previous block in the chain.
- The block containing John's transaction is then broadcasted to the entire network to be verified.
- Once verified, the block is added to the head of the blockchain, forming a permanent and transparent record of transactions.
- After the verification process, Anna receives the money from John.

10,000 ft view

[in ‘tech & blockchain’ terms]

- **INITIATE THE TRANSACTION**
  - ✓Multiple parities transact;
  - ✓All transactions are recorded, including the transactions time, date, parties and amounts;
- **POST AND RECORD THE TRANSACTIONS TO THE NETWORK**
  - ✓The transaction is added in order into a network’s ‘block’ and presented;
  - ✓Each node in the network owns a copy of the ledger;

10,000 ft view

[in 'tech & blockchain' terms]

- **BROADCAST**
  - ✓The 'block is broadcasted to every node in the network.
  - ✓The network of computer nodes verify and validate by running a software that continuously replicated the ledger;
- **VALIDATE VIA CONSENSUS AND CONFIRM**
  - ✓multiple parities transact; all transactions are recorded, including the transactions time, date, parties and amounts;
  - ✓Consensus (agreed mathematical mechanism) is recorded and provides the basis for the trust mechanism.

10,000 ft view

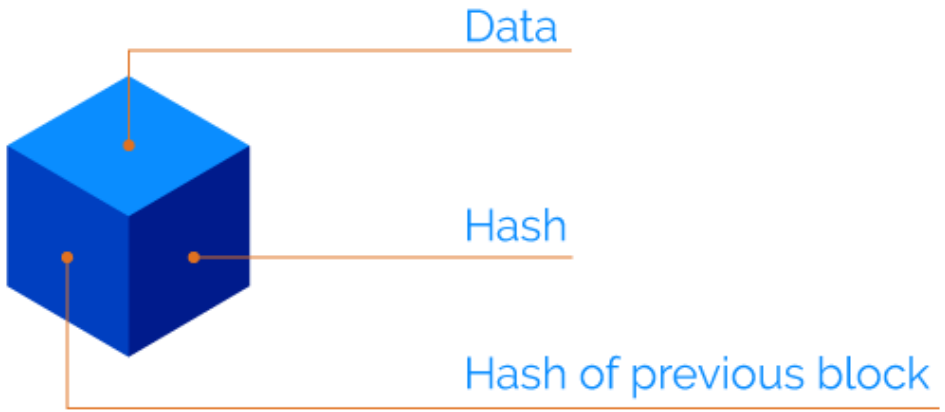
[in 'tech & blockchain' terms]

- **IMMUTABLE, ENCRYPTED BLOCK**
  - ✓The confirmed block is added in linear and chronological order to the chain;
  - ✓This provides a transparent record of transactions, audit trail, and traceable digital fingerprints;
  - ✓Data creates a reliable transaction record;
- **TRANSACTION COMPLETED**
  - ✓Nodes have access to a single shared source of truth;

BLOCK – what is it?

[in blockchain]

- Each block references the previous one and contains data, its own hash, and the hash of the previous block.
- Data stored inside a block may be represented by any value depending on the type of the blockchain.
- A block can store any amount of money, a share in a company, a digital certificate of ownership, a vote, or any other value.



source: rubygarage.org

Inside Bitcoin's Blockchain

Designed by www.fractalphia.com

HEADER

The block header is hashed twice to create the fingerprint which is referred to in the next block.

Technical data

Includes a Magic ID, a version number (to specify which set of protocol rules this block conforms to), the size of this block.

Previous block hash

2x SHA256 hash of previous block header (excluding magic ID & block size). This is the link that creates the chain of blocks.

Merkle Root

Distills all the transactions in the block into a single hash.

Timestamp

Approximate timestamp of when the block was created. Used to figure out mining difficulty re-targets i.e if the network is making blocks too quickly or too slowly.

Difficulty target

Related to mining and how hard it is to successfully mine the block

Nonce

A random number. One of the things you can change when mining to create different hashes, while searching for a suitable hash.

TECHNICAL DATA

Version number

Can be used for specifying which set of protocol rules this transaction confirms to.

Transaction lock time

Something which may be used in future for "future dating" a transaction, like writing a post-dated cheque.

Input count

How many inputs are in this transaction.

Output count

How many outputs does this transaction create.

INPUT

(Technical) Input script length

How much data is in the input.

(Technical) Sequence number

Not really used.

Previous transaction hash & index

This identifies where the coins are coming from, by specifying an output from a previous transaction.

Script data

This is where you "prove" you own the coins and you are allowed to spend it, by signing with the private key of the address that the bitcoins are in.

BLOCK

Blocks are the units of the blockchain, like pages of transactions in a ledger.

HEADER

- Technical data
- Previous block hash
- Merkle Root
- Timestamp
- Difficulty target
- Nonce

TRANSACTION COUNT

How many transactions are in the block, including the coinbase transaction.

BLOCK CONTENT

- Coinbase transaction
- Bitcoin transactions

TRANSACTION

Each transaction is a bitcoin payment

- Technical Data
- Inputs: Which coins are being spent?
- Outputs: Who is getting the coins?

INPUT

- (Technical) Input script length
- (Technical) Sequence number
- Previous transaction hash & index
- Script data

OUTPUT

- (Technical) Output script length
- Amount
- Output script

BLOCK CONTENT

Coinbase transaction

The bit where you get to pay yourself the mining reward (currently 25 BTC) plus the fees from the transactions included in the block.

It's a special transaction where there are no 'inputs' or 'from' addresses.

Bitcoin transactions

This is the main payload of the block. Contains bitcoin payments.

- Transaction
- Transaction
- Transaction
- Transaction

FOLLOWING THE MONEY

Bank accounts vs cryptocurrencies

Bank accounts mix money up. When you pay someone, you don't specify "use those pounds which I earned from my salary" or "use those pounds which I received for my birthday". Money is treated equally once it hits your account, and is untraceable.

On the other hand, with cryptocurrencies, you need to specify exactly which incoming deposits you are spending. This makes every transaction traceable, right back to the creation of the coins.

Inputs and Outputs

Every bitcoin transaction references some incoming deposits as inputs, and spends them entirely as new outputs, with change returned to one of your addresses.

This is like paying £43.50 by taking three £20 banknotes from your wallet and creating two new banknotes: £43.50 and £16.50. You hand over the £43.50 banknote and keep the £16.50 banknote. You can then spend the £16.50 later in one go. The other person can spend the £43.50 later in one go.

Inputs: 3 x £20

Outputs: £43.50 (payment), £16.50 (change)

OUTPUT

(Technical) Output script length

How much data is in this output

Amount

How many bitcoins (actually, Satoshis) are being sent.

Output script

Who (which address/es) are the bitcoins being sent to? Which signatures are needed to re-spend these coins?

What data does each block consists of?

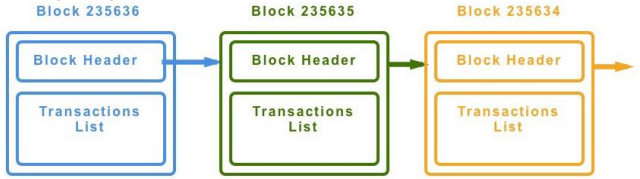
[fly by at 1,000 ft – a closer look]

❖**HEADER** (the block header is hashed twice to create fingerprint which is referred to in the next block

- **Tech Data** – size of the block, Magic ID, version;
- **Previous Block Hash** – SHA256 of previous block header (excluding ID and block size);
- **Merkel Root** – distils all transactions in the block into a single hash;
- **Time Stamp** – when the block was created, used to figure out the mining difficulty re-targets i.e. if the network is making the block too quickly or too slowly;
- **Difficulty Target** – related to the mining and how hard is to successfully mine the block;
- **Nonce** – A random number, you change while mining to create a different hashes, while searching for a suitable hash (target below the current difficulty level).

❖**BITCOIN TRANSACTIONS**

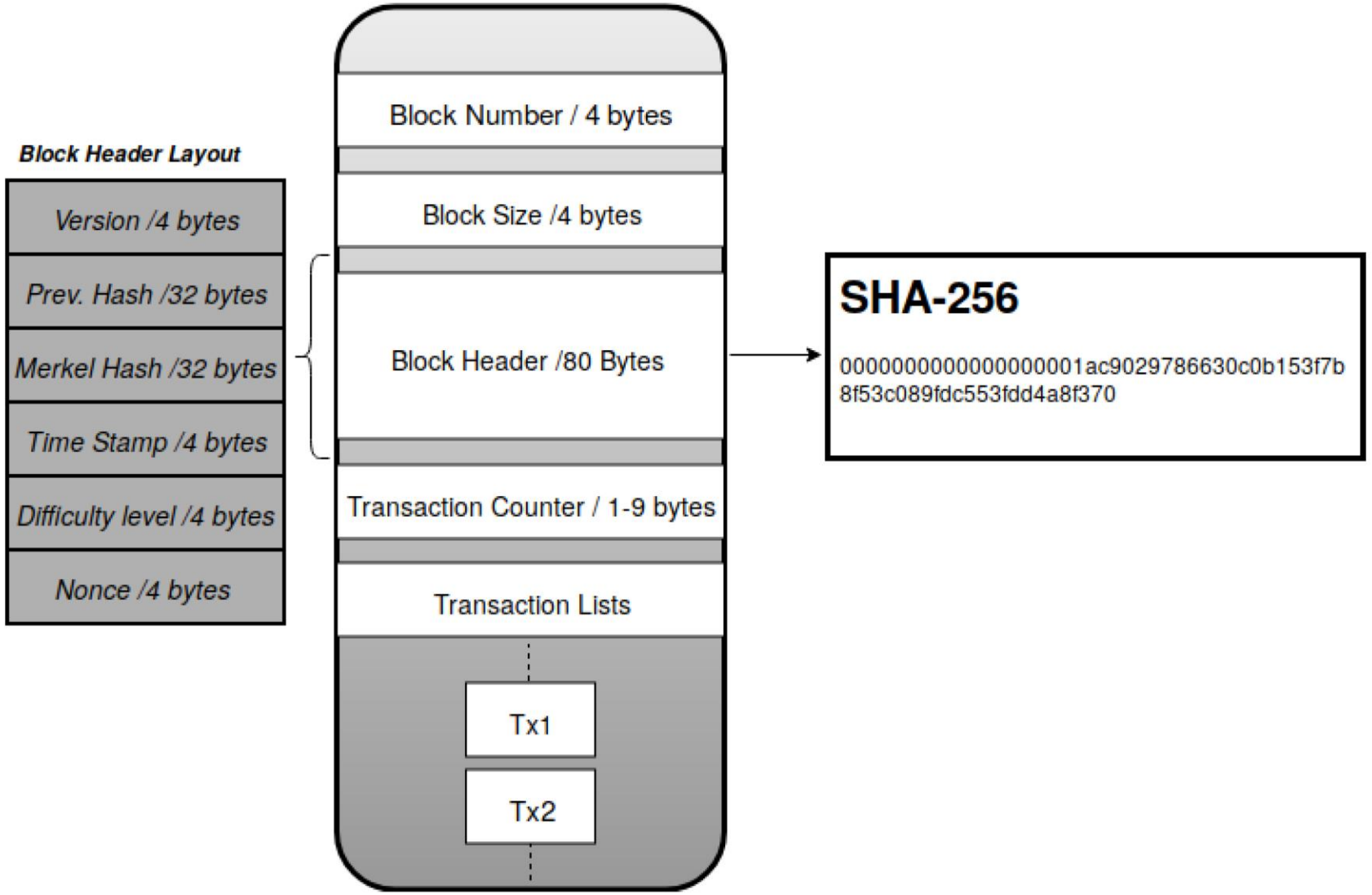
- **Coinbase Transaction** – mining reward + transaction fees to the one who solves the puzzle first;
- **Bitcoin Transactions** – main payload of the block, contains bitcoin payments;



source: bitsonblocks.net

BLOCK STRUCTURE

[bitcoin block]



HASH – securing of the data

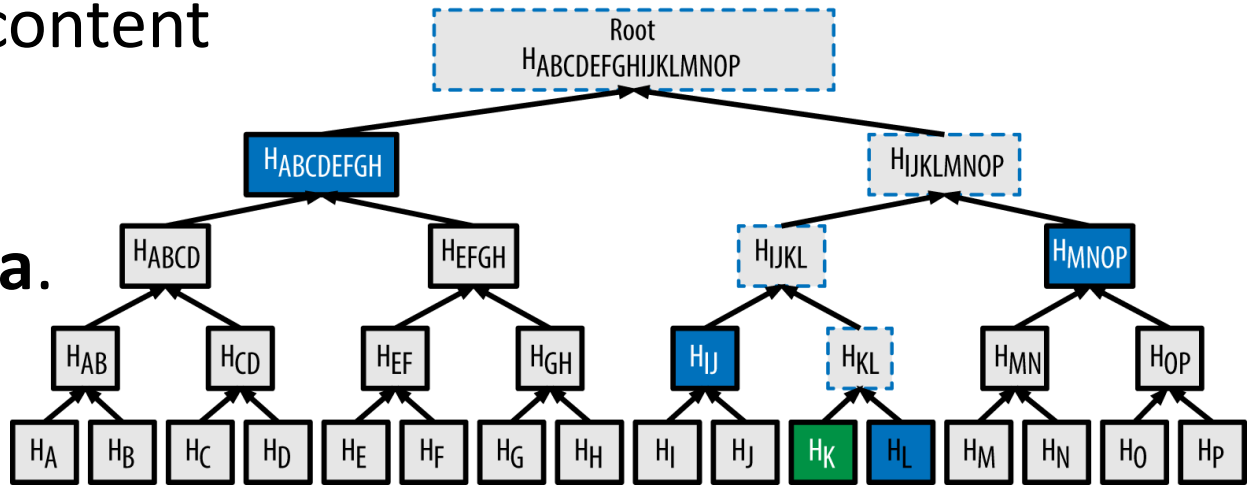
- Bitcoin uses a **SHA256** hash algorithm, email uses a MD5 hash algorithm. Hash is a mathematical function that takes an any size input and converts into encrypted fixed size output;
- A **hash is a fingerprint**, as each **hash is unique** – some qualities are:
  - ✓ Impossible to produce same hash value for different inputs.
  - ✓ The same input will always produce the same output.
  - ✓ Quick to produce a hash for any given data.
  - ✓ Impossible to determine input based on the output (one way only).
  - ✓ Even the slightest change to an input data drastically alters the output.

source: lisk.io

Merkle Root (Hash Tree)

[has nothing to do with Angela Merkel in Germany]

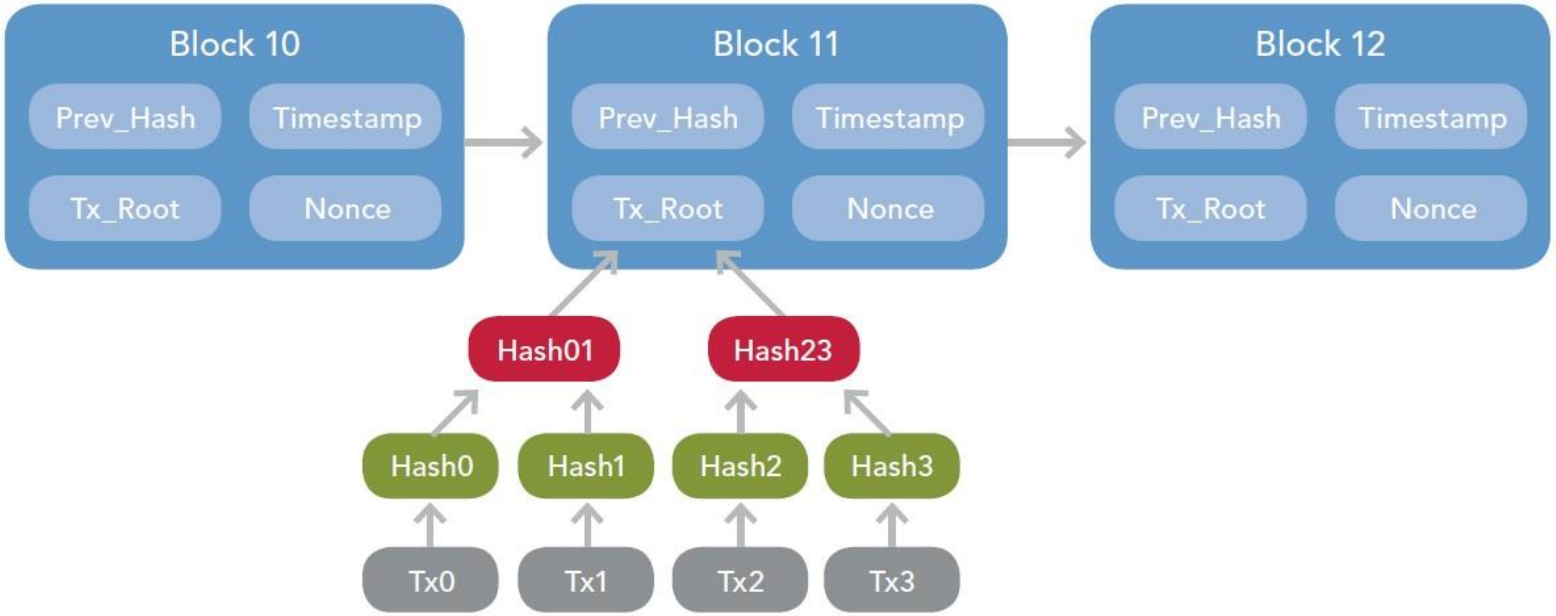
- Fundamental part of blockchain technology - **maintains the integrity of data** – the truth.
- Each transaction in the block (bitcoin blockchain) in hashed. **Hash values are further combined in a system known as a Merkle Tree**. Thus, a Merkle tree **summarizes all the transactions in a block by producing a digital fingerprint** of the entire set of transactions.
- A Merkle tree is a structure that allows for efficient and secure verification of content in a large body of data.
- This structure **helps verify the consistency and content of the data**.



Merkle Root (Tree)

[has nothing to do with Angela Merkel in Germany]

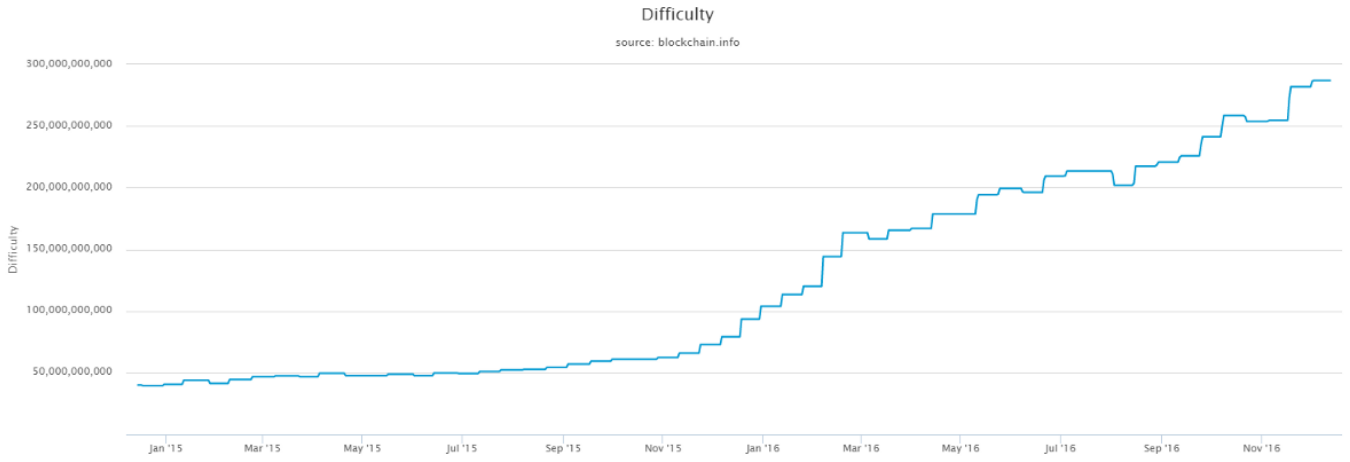
- They are constructed bottom up from hashes of individual transactions known Transaction IDs.
- Using a Merkle tree allows for a quick and simple test of whether a specific transaction is included in the set or not.



DIFICULTY TARGET

- **As more miners (nodes) join the network the more difficult it is to solve the puzzle, and vice versa**.
- Difficulty level adjusts every approximately 2 weeks or 2016 blocks, to **make sure it takes 10 minutes to create a block**.
- Difficulty is set by the protocol. [link]
- As the difficulty increases, so the target value for the hash decreases. This means there have to be more zeros at the start of the hash number.

Date: Mar 24 2019
Difficulty: 6,379,265,451,411
Hash rate: 45,664,560,811 GH/s



NONCE

- Only blocks whose hashes start with a certain number of 0's can be added to the ledger. Why?
- **The idea is to create competition among the miner.** Who will find the number that will contain a run of leading zeros below the difficulty target first. Remember the **SHA256 hash is unpredictable**.
- Thus, the miners has to **add an arbitrary number to the block, known as nonce** (i.e. 9876879).
- Finding the a block with a certain number of 0's is a **brute force task**, and can **only be achieved by testing different combinations** as fast as possible (lots of random guesses/trail and error – which keeps it fair).
- Hash value is unique for the combination of a nonce and input data
- The guessing game is called 'hashing'.

Transactions (Tx)

[private & public key]

- Cryptography also helps one more puzzle – **authenticity**.
- To be part of the bitcoin network all you need is access to internet. Every user in the network is **identified by** a string of characters, known as the **bitcoin address**.
- To transact with someone in bitcoins you need to know their bitcoin address (i.e. email).
- **Your bitcoin address is derived from a public key which is derived from a private key.** Private key should be kept secret like your password to an email.

Transactions (Tx)

[private & public key]

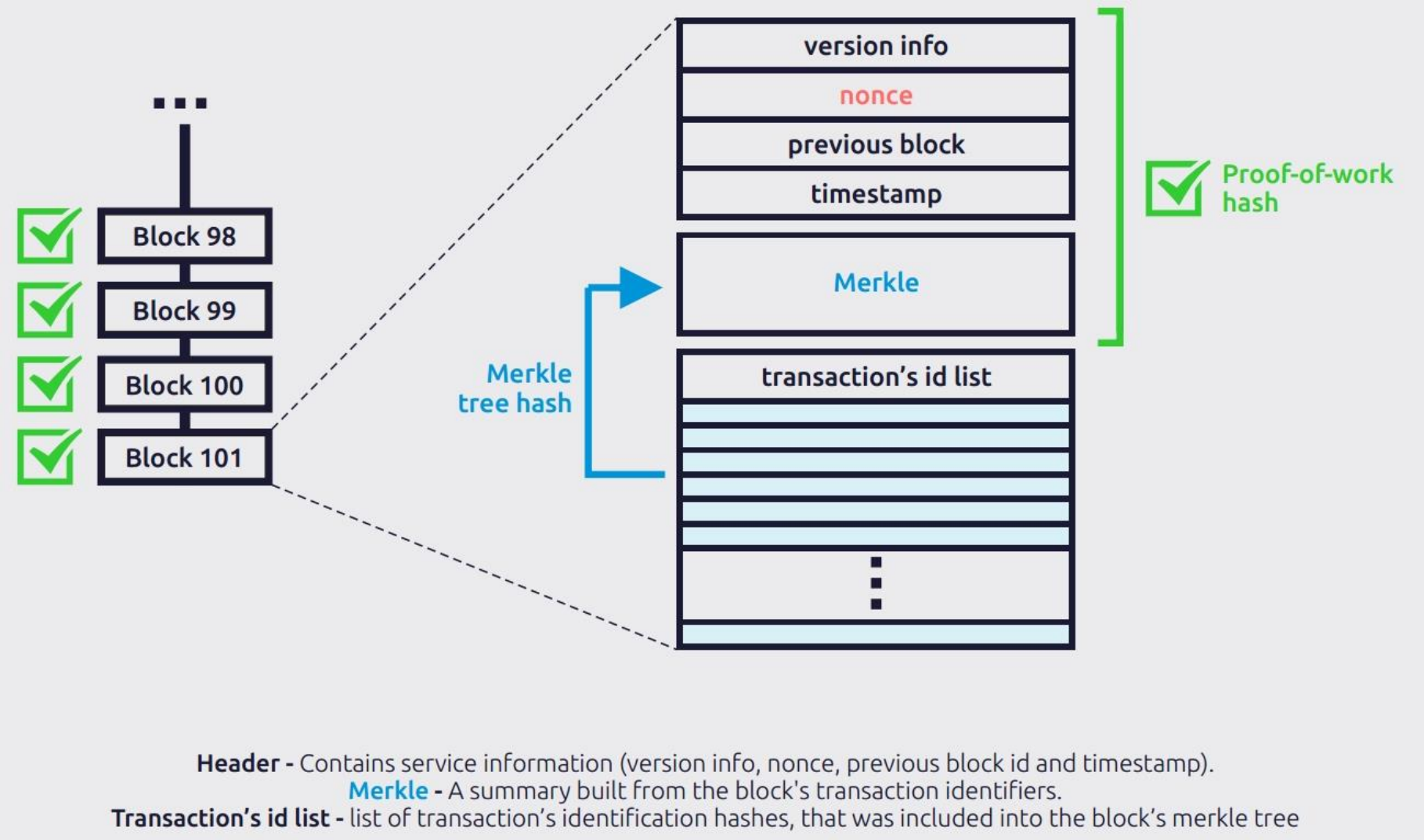
- A transaction signature = hash (private key + bitcoin amount)
- So when sending a bitcoin to someone it includes:
  - My **signature**: hash output of my transaction and my private key
  - My **transaction**: since you can’t determine the amount from the signature
  - My **public key**: to identify myself as the sender

- Where do I send it? To someone's bitcoin address which is derived from his public key.

- Thus, when a transaction is picked up from the mempool by the miner they check if all the components match.

What does a block look like?

[simplified block structure]



SHA256 Hash of Blocks = SHA256 (Block # + Nonce + Data (Coinbase & Transaction Lists) + Previous Block Hash + Time-Stamp)

source: theblockchaintest.com

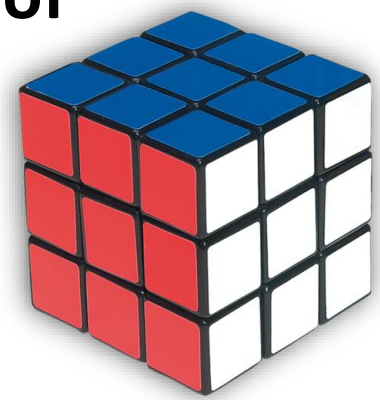
Consensus Mechanisms

- Most blockchains have a lot of things in common and function in similar ways, but one of the ways in which **blockchains can be unique is the way consensus is reached**.
- Consensus mechanisms are protocols that make sure all nodes (device on the blockchain that maintains the blockchain and sometimes processes transactions) are synchronized with each other and **agree on which transactions are legitimate** and are added to the blockchain.
- These consensus mechanisms are crucial for a blockchain in order to function correctly. **Without a good consensus mechanisms, blockchains are at risk of various attacks**.

Proof of Work

[mining]

- It is one of many consensus mechanism; this algorithm is used to **confirm transactions and produce a new block to the chain**.
- **Miners compete** with each other by solving the PoW algorithm in order **to receive the reward.**
- Mining is done with dedicated computer machines which **costs processing power (hardware, energy, and time).**
- Miners are using their hardware to test these nonce, at a **rate of millions per second.**
- Difficult to solve but very easy to verify the correct value!



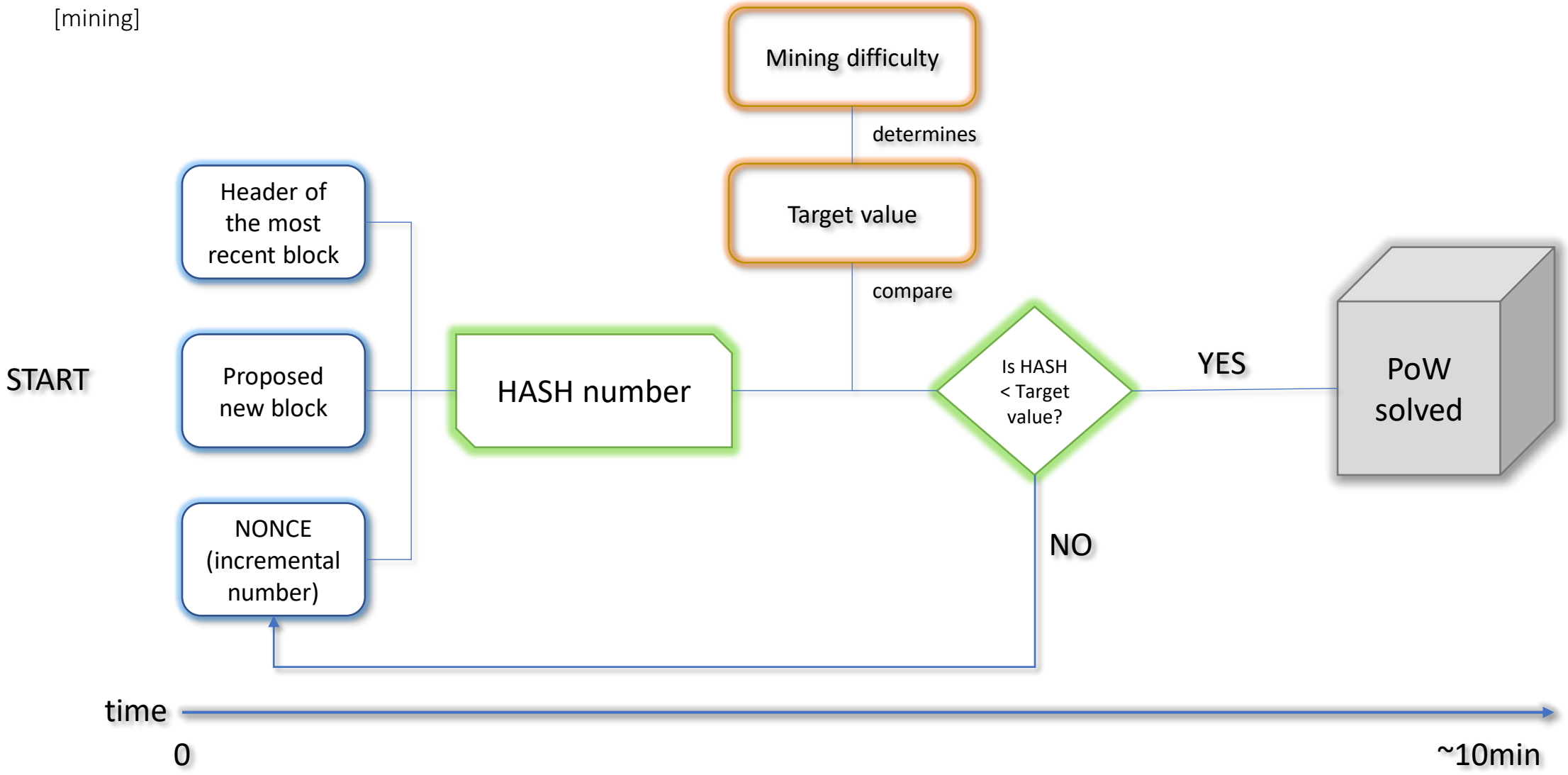
Proof of Work

[mining]

- **The Proof Of Work process is known as mining** and the nodes are known as miners. Miners **solve complex mathematical puzzles** which require a lot computational power.
- **First** of all, they are asymmetric, meaning it **takes a lot of time to find the answer, yet it's easy to verify if an answer is correct**.
- **Secondly**, the only way **to solve these puzzles is to 'guess' the answer**. It is not possible to solve the puzzles quicker using any other **method than trial and error.**
- **Lastly**, the **difficulty of these puzzles changes depending on how fast blocks are mined.** To maintain a consistent supply of new coins, blocks have to be created within a certain time frame. If blocks are created too fast, the puzzles get harder, and if they are created too slow, the puzzles get easier.

Proof of Work

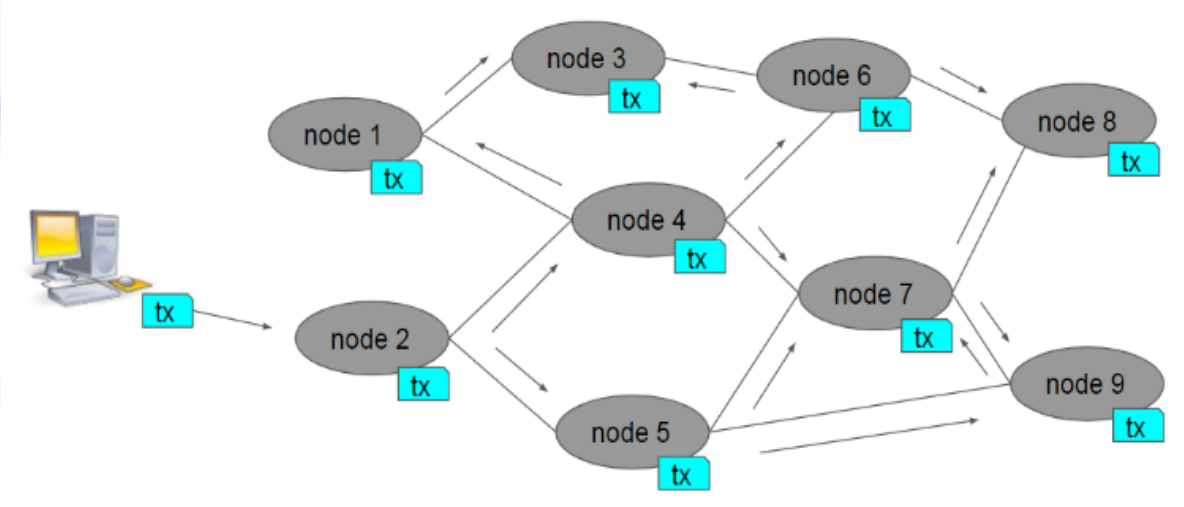
[mining]



Block Creation – Transaction **Network Propagation**

[PoW - the 10 minute heart beat]

- George send a transaction/funds to Ana.
- The node checks if it is valid, and propagates (send) to other nodes if it is – and so on and so forth until all nodes receive the transaction.



Source: unic.ac.ay

Block Creation – **From Transaction to Blocks**

[PoW - the 10 minute heart beat]

- From a bitcoins node perspective, the node receives a transaction (TX) which goes into a memory pool (**mempool**). *Note: not all mempools are same, each node has a different mempool*. Node 1 can have 10 transactions and at the same time Node 2 will have 15 transactions. *Nodes are miners in this case (miners are computers on the network, computers are owned by individuals, groups of people, or companies)*.
- It keeps receiving transactions from all other users on the network…
- At some point it will decide if it will group these TXs into a block.
- It will add the **coinbase** TX which rewards 6.25 BTC to a node's BTC address.
- Finally, it adds the block's header containing important information.

Source: unic.ac.ay

Block Creation – **Mining**

[PoW - the 10 minute heart beat]

STEP 01

- After a node (miner) created a block it will attempt to make it final by propagating it to all other nodes in the network. Multiple nodes will receive the same transaction and will create blocks; nodes choose which TXs to include. They can create and propagate a block at any time.
- A very **difficult computational problem** needs to be solved in order to accept a block as valid. The process of finding the solution requires work (**Proof-of-Work >> PoW**) and is called mining.

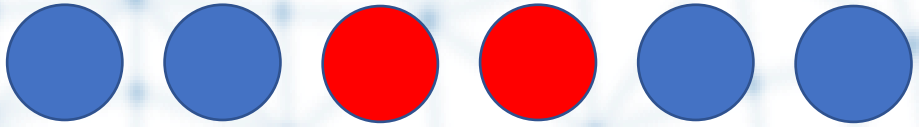
STEP 02

- The PoW puzzle is to compute a cryptographic hash of the new block that we want to create which should be **less than a given number**. Since a hash is random it will take several attempts to find a proper has but other nodes will verify with only one attempt.

Source: unic.ac.ay

Block Creation – **Mining**

[PoW - the 10 minute heart beat]



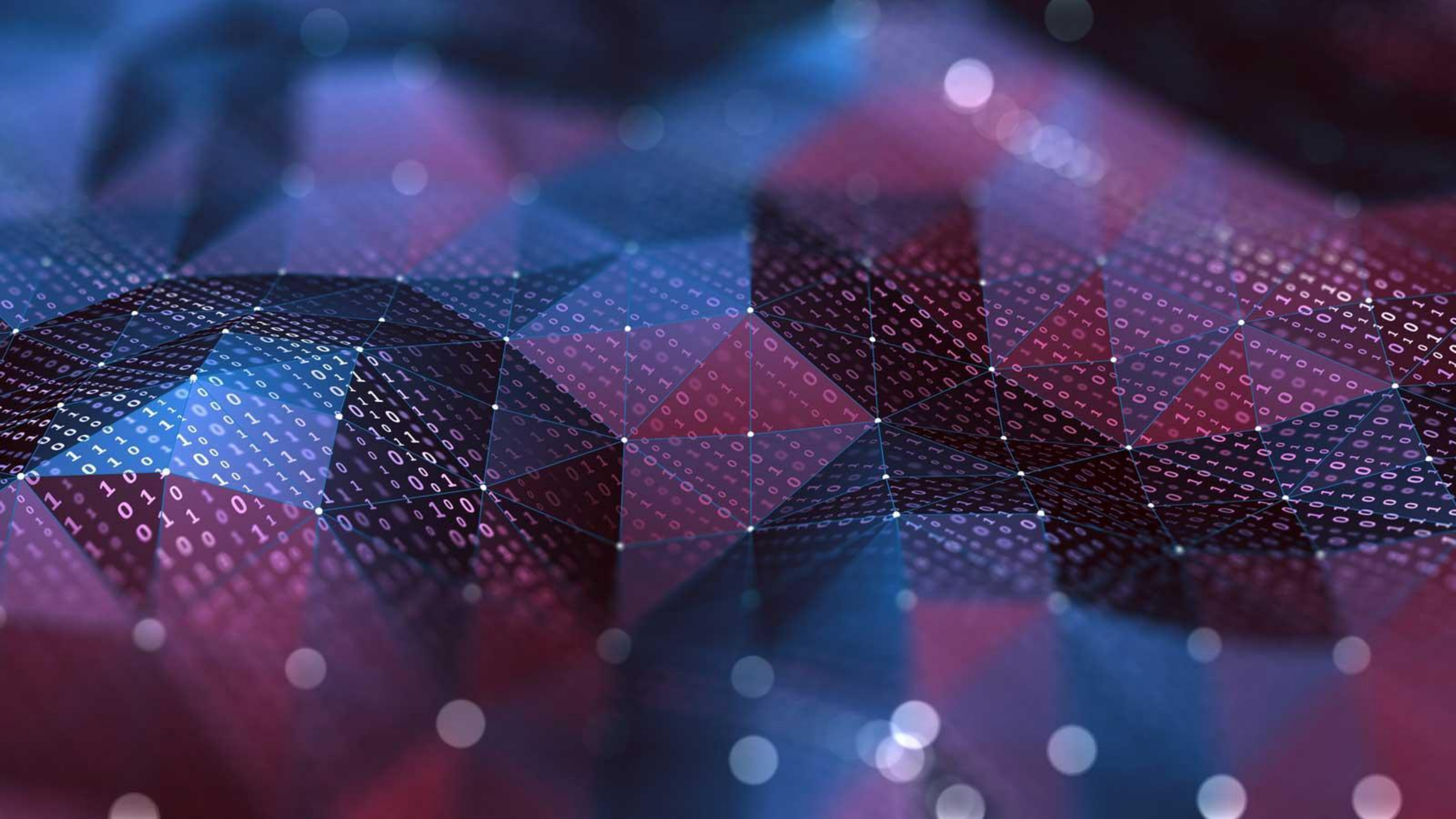
STEP 03

- Puzzle's difficulty automatically adjusts so that it requires approximately 10 minutes to solve. As more miners join the network the block will be created faster. The difficulty of the puzzle is then increased to require ~10 minutes again. (Adjustment happens every 2016 blocks, approximately 2 weeks, if each block takes 10 minutes to mine).

STEP 04

- Additionally, the transactions fees of all the TXs in a block are also awarded to the miner that creates the new block.
- The header of a block contains, previously created block. A block always contains a coinbase tx transaction which is used to pay the mining reward to the miner. The mining reward is available to the miner after 100 confirmations.

Source: unic.ac.ay



Smart Contract – Characteristics

[smart contract are not a technical issue but a business one]

1. Smart contracts **can call other contracts**, if needed. This means that if there are **two parts of a contract**, wherein the first part is concerned with **checking a requirement**, then after the condition gets validated the second part of the contract where the **task might be to carry out** a transaction, will receive the output of the first part and the transaction will take place successfully.
2. Smart contracts act as accounts that need multiple signatures. Unless a required number of people sign, funds won't be transferred.
3. Smart contracts help in **managing the agreement** between two or more users, for example if one person sells an insurance to the other.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

Smart Contract - Characteristics

4. You don’t need to rely on any intermediaries to carry out or confirm your agreements. This **removes the risk of a third party manipulation**, as **execution is done by the network in an automated way**.

5. All the **documents are encrypted in a shared ledger** so nobody can claim to have lost something of yours.

6. Everything is backed up in the blockchain, as all the nodes have a copy of it so you can’t really worry about losing data.

7. The smart contracts are built on a blockchain environment so everything is well encrypted so it would take a really smart hacker to tamper with data.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

Smart Contract - Characteristics

8. Smart contracts **save your time** as all tasks are automated due to them **written in software code**. This also **removes the paperwork** and endless settlements and other legalities.

9. Smart contracts also save you money since no middleman is being paid here, thanks to the blockchain environment!

10. Smart contracts are pretty accurate as they are automated, which **removes the possibility of human errors** that may occur in filling out heaps of forms.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

What is Solution Architecture?

- **Solution** – a way to describe the answer to some problem; and addressing them with appropriate system that provide measurable improvements in place of previously deployed systems.

- **Architecture** – complex or carefully designed structure of something

- **Solution Architect** (job) – involved with the crafting the business need (usually a technical one) and is tasked with crafting the description of solutions that can be to address that need. (choice of weapon – diagrams or other visual aids is a key to success)

https://medium.com/@dltlabs/what-is-solution-architecture-eb14b81c240d

Future

The use of blockchain combined with smartphones, RFID, IoT networks, and mobility applications could create (by 2030) or make almost every if not all industries with near-perfect records.

Further Readings and Resources

- Andreas Antonopoulos [link]
- Try to go over the Blockchain Demo by Anders [link]
- (**Advanced**) How to set up a private Ethereum blockchain and deploy a Solidity Smart Contract on the blockchain — in less than 20 mins! [link]

Cryptocurrency Realtime Visualization Websites

[interactive websites]

- Market Capitalization and Volume – crypto info [link]
- Watch live bitcoin transactions presented as balls – [link]
- Bitnodes across the globe – [link]
- Network Map – [link]

MORE TO FOLLOW IN NEXT SESSION

...

DATABSE or BLOCKCHAIN / A better fit?

DATABASE (CRED)	BLOCKCHAIN (CR)
Performance is Important	Requires transparency and public validation
Large number of trained resources is important	Needs extreme fault tolerance
Don’t need a history, just a snapshot of data	Needs to be infinitely scalable
Ease of maintenance	Full data history and lifecycle is important to capture
Application logic expected to change frequently	No single authority can or should be entrusted with the data
Maintaining centralized control of resources is important	

Where can blockchain be used?

Banking Industries, Business Industries (Supply chain), Healthcare, Retail Industries, Government Agencies, Cyber Security, Game and Entertainment Industries, and other industries where there is record management, IoT, voting, etc.

OSI 7 Layer Model

[how internet works]

Not on exam!
Visualize the internet…
As we will do the same for blockchain…

OSI (Open Source Interconnection) 7 Layer Model

	Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Upper Layers	**Application (7)** Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP		GATEWAY Can be used on all layers	Process
	Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
	Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
	Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Lower Layers	**Network (3)** Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
	Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card —— Switch —— NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch** **Bridge** **WAP** PPP/SLIP	Land Based Layers		Network
	Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

Types of Blockchain

- Public / Private / Consortium (Enterprise, Hybrid…)

- Main considerations when selecting a blockchain framework to develop your application on:
  - ❑PERMISSIONS
  - ❑PERFORMANCE
  - ❑PRIVACY
  - ❑CONSENSUS MECHANISMS

PUBLIC vs PRIVATE blockchain

- The major difference between public and private blockchain is **key generation**: can anyone generate a public and private key-pair to create an identity that can interact with the blockchain, or is there an **identification process** and permissioned access required?

- This impacts both **permissions** and **privacy**.

Permissionless (Open or Public)

1/2

- **Permissionless**/public blockchains are **transparent and openly accessible**. The entire **transaction history is available for anyone to view or download**; they can **run the software and create nodes to participate in the consensus and validation process**; and they can **generate key-pairs** and **addresses**, and **submit transactions** or develop and **deploy smart contracts** which **alter the global state without the need to identify themselves.**

- While this makes for a permisionless and decentralized system where you an **interact with a degree of privacy** in regards to identity, **the transaction history associated with that identity is completely open due to the transparent nature of public blockchains.**

Permissionless (Open or Public)

2/2

- Permissionless blockchains also tend to have a native token that is used to incentivize users and is tied to the **consensus mechanism**. The block **rewards for miners** in Proof-of-Work networks and the fees for executing transactions are distributed in this **token**.
- The **consensus** mechanism that a particular blockchain employs has a large impact on the **performance** of the framework and will affect the speed at which transactions are processed, and the functionality of your application.
- Governance in a permisionless model is generally implemented through the **consensus** layer with a 50% + 1 voting model.

Permissioned (Closed or Private)

1/3

- In a **permissioned**/private network you have the ability to **decide which entities can participate** in terms of clients, validators, and nodes, and define which operations they have access to by **controlling permissions** such as: reading transaction history, calling smart contract functions, or sending transactions.
- The ability to **control the access** to data allows for a **higher degree of privacy** for sensitive material while still ensuring transparency of less personal data records.
- Most private blockchain frameworks have the functionality to create tokenized digital assets, but **few have an economically incentivized consensus model and therefore have no need for a native token**.

Permissioned (Closed or Private)

2/3

- Private blockchain frameworks offer a **variety of different governance models** when it comes to making administrative changes.
- **Hyperledger Sawtooth** can run as either a public or a private blockchain, and has two governance options that are implemented through the Settings Transaction Family. You can choose a single authorized key that is allowed to make changes to the network, or a list of authorized keys and a set minimum number of votes to approve proposed changes.

Permissioned (Closed or Private)

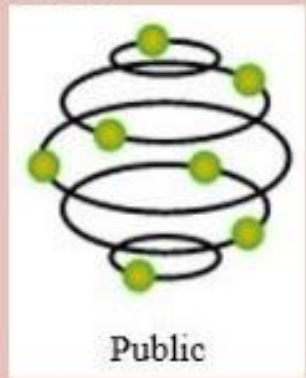
3/3

- Some of the changes contained in this transaction family are your chosen target wait time for Sawtooth's PoET lottery style consensus mechanism, and your preferred maximum number of transactions per block. The low-performance requirements of this consensus mechanism should allow for a very high transaction throughput and significant **performance** improvements over most existing **consensus** mechanisms.

- Because of the separation of the network and the application layer, and the modular design which defines these rules in one place and the rest of the framework references them, it means that on an approved vote changes can be dynamically updated without negatively impacting the applications running on the network.

Types of blockchains

	PUBLIC	PRIVATE/CONSORTIUM
ACCESS	OPEN READ/WRITE	PERMISSIONED READ AND/OR WRITE
SPEED	SLOWER	FASTER
SECURITY	PROOF OF WORK PROOF OF STAKE OTHER MECHANISMS	PRE-APPROVED PARTICIPANTS
IDENTITY	ANONYMOUS PSEUDONYMOUS	KNOWN IDENTITIES



Questions to ask yourself?

- What functionality do you want to have administrative control over?
- Do you want the ability to restrict user access?
- Do you want to allow anyone to join, but to implement a permissioned system in regards to which functions users can call?
- Is there sensitive data handled by your application that needs privacy measures put into place?
- Do you need the speed of near-instant transaction confirmation times?
- Would a hybrid solution best fit your use case with some data stored on a public chain, and private data in a permissioned chain?
- Is your use case better suited to a public chain with private data stored in a centralized database or an encrypted state on a distributed storage option like Storj or IPFS?
- +more

Blockchain-architecture options

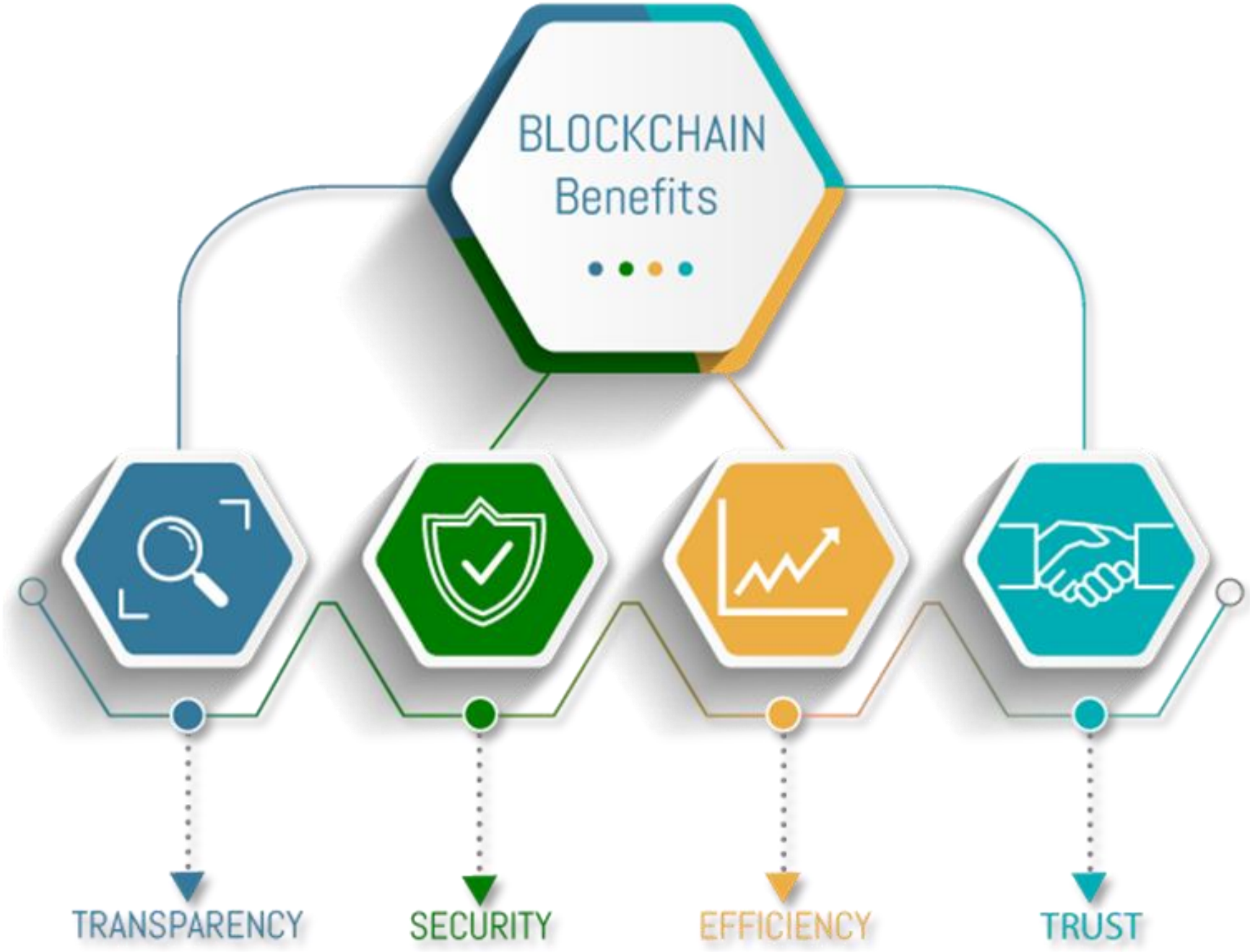
Architecure based on read, write, or commit permissions granted to the participants

Architecture based on ownership of the data infrastructure	Permissionless	Permissioned
Open	● Anyone can join, read, write, and commit ● Hosted on public servers ● Anonymous, highly resilient ● Low scalability	● Anyone can join and read ● Only authorized and known participants can write and commit ● Medium scalability
Closed	● Only authorized participants can join, read, and write ● Hosted on private servers ● High scalability	● Only authorized participants can join and read ● Only the network operator can write and commit ● Very high scalability

McKinsey&Company

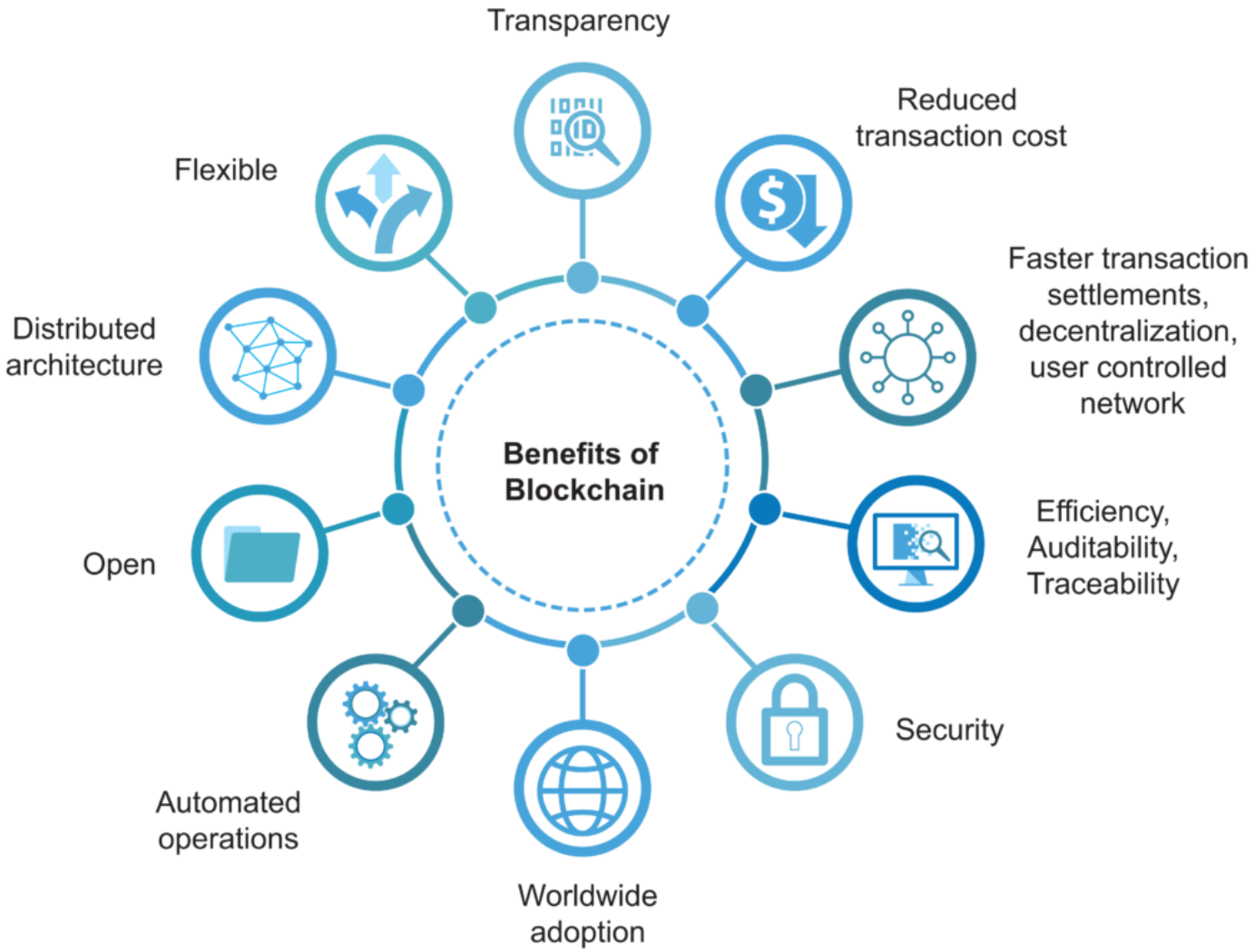
Types of Blockchain

Permissionless vs. **Permissioned Blockchain** (Public vs Private)	**Open vs Closed Blockchain**
Public – where anyone can add a record	**Open** – everyone can read blockchain data
Private – where only certain participants can write data	**Closed** – only certain participants can read data


BLOCKCHAIN
Benefits
TRANSPARENCY
SECURITY
EFFICIENCY
TRUST

Blockchain Benefits

[more]



Consensus Mechanisms

Blockchain are not a truth machine!

So, how do blockchains networks make and agree on decisions?

Consensus Mechanisms

- One of the **primary selling points of blockchains networks is decentralization** – they can **function even in the absence of a central authority.**

- However, since decisions about future development and maintenance of the projects still have to be made, a **consensus algorithm allows network participants to arrive at a common decision.**

Consensus Mechanisms

“The purpose of a consensus algorithm is to allow for the secure updating of a state according to some specific state transition rules, where the right to perform the state transitions is distributed among the users of a particular economic set”

- Vitalik Buterin, Ethereum -

Consensus Mechanisms

- This proliferation of consensus algorithms can be attributed to the **blockchain scalability trilemma**, a term that refers to the technology's various bottlenecks.
- While an ideal distributed network would excel at security, decentralization, and throughput, most digital currencies today have only managed to obtain one or two of those characteristics.
- As a result, **developers are constantly working on new consensus algorithms** to build a close-to-perfect blockchain network.

At which stage is the consensus used

Figure 1: What exactly are blockchains?

Blockchains are a way of ordering and verifying transactions in a distributed ledger, where a network of computers maintains and validates a record of consensus of those transactions with a cryptographic audit trail.

Initiate the transaction.

- Multiple parties transact.
- All transactions are recorded, including the transaction's date, time, parties, and amount wants to do a transaction.

Post and record the transaction to the network.

- The transaction is added in order into a network's 'block' and presented.
- Entries can be added but not deleted.
- Each node in the network owns a full copy of the ledger.

Broadcast.

- The 'block' is broadcast to every party and their nodes in the network.
- The network of computer nodes verify and, validate by running a software that continuously replicates the ledger.

Validate via consensus and confirm.

- The network verifies, validates, and approves; the confirmation is broadcast to the other nodes.
- Consensus (agreed mathematical mechanism) is recorded and provides the basis for the trust mechanism.

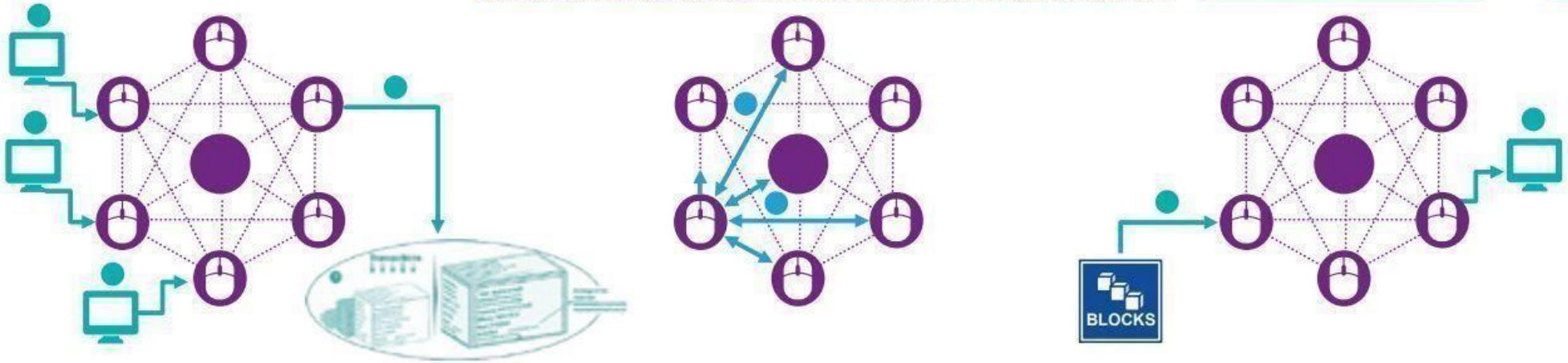
Consensus mechanism applied

Immutable, encrypted block

- The confirmed block is added in a linear and chronological order to the chain.
- This provides a transparent record of transactions, audit trail, and traceable digital fingerprint.
- Data is pervasive and persistent and creates a reliable transaction record.

Transaction completed.

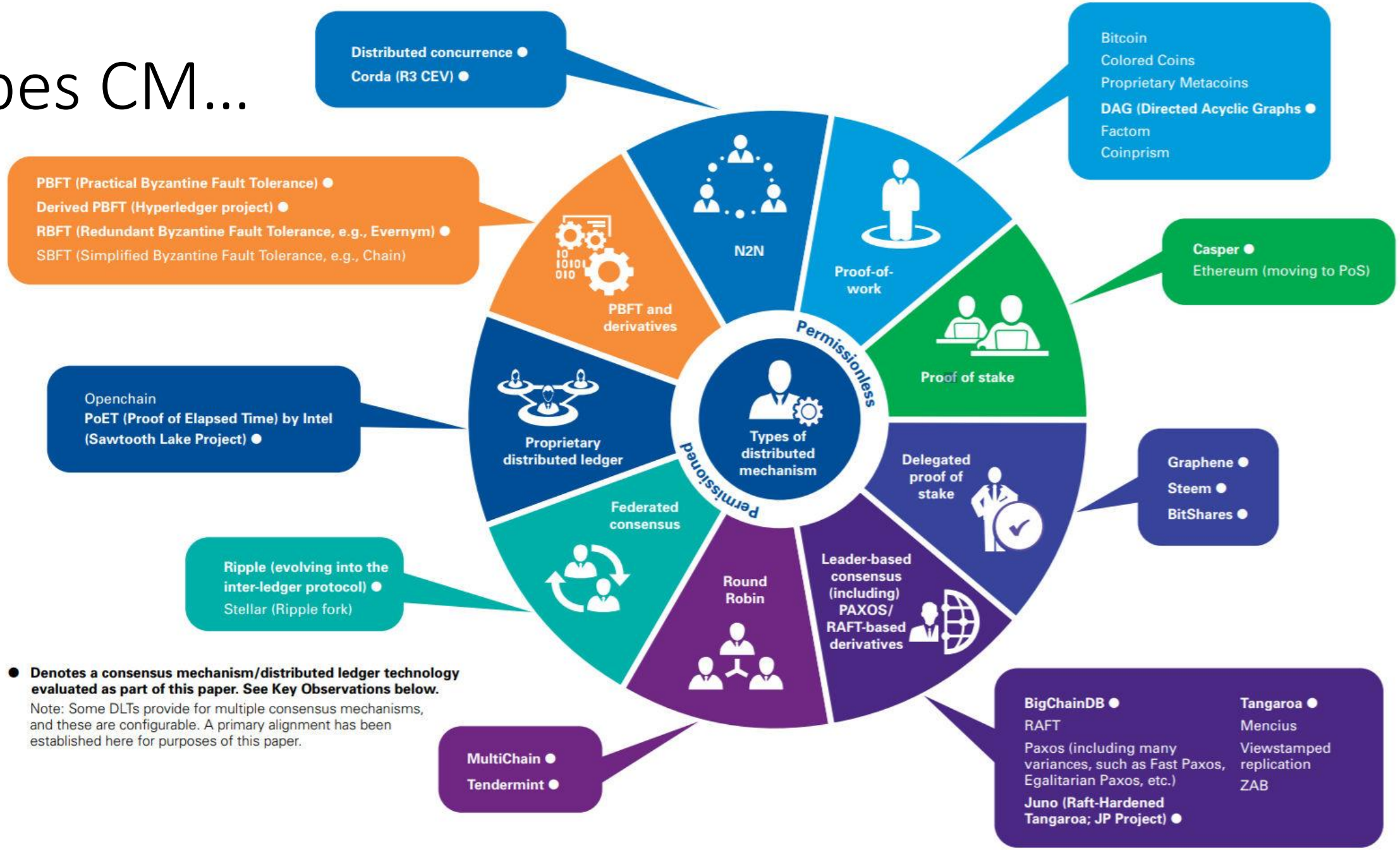
- Nodes have access to a shared single source of truth.
- A completed block gives way to the next block in the blockchain.



Consensus are the Core to Blockchain

- Without central authority, participants **have to agree on rules and how to apply them.**
- Consensus mechanism is a **set of rules and procedures** that maintains a coherent set of facts among the participating nodes.
- The consensus mechanism is at the **core of Blockchain design** and selection.
- There are **many consensus** mechanisms.

Types CM…


Distributed concurrence ●
Corda (R3 CEV) ●
Bitcoin
Colored Coins
Proprietary Metacoins
DAG (Directed Acyclic Graphs ●
Factom
Coinprism
PBFT (Practical Byzantine Fault Tolerance) ●
Derived PBFT (Hyperledger project) ●
RBFT (Redundant Byzantine Fault Tolerance, e.g., Evernym) ●
SBFT (Simplified Byzantine Fault Tolerance, e.g., Chain)
Casper ●
Ethereum (moving to PoS)
Openchain
PoET (Proof of Elapsed Time) by Intel
(Sawtooth Lake Project) ●
Graphene ●
Steem ●
BitShares ●
Ripple (evolving into the
inter-ledger protocol) ●
Stellar (Ripple fork)
● Denotes a consensus mechanism/distributed ledger technology evaluated as part of this paper. See Key Observations below.
Note: Some DLTs provide for multiple consensus mechanisms, and these are configurable. A primary alignment has been established here for purposes of this paper.
MultiChain ●
Tendermint ●
BigChainDB ●
RAFT
Paxos (including many variances, such as Fast Paxos, Egalitarian Paxos, etc.)
Juno (Raft-Hardened Tangaroa; JP Project) ●
Tangaroa ●
Mencius
Viewstamped replication
ZAB
N2N
Proof-of-work
Proof of stake
Delegated proof of stake
Leader-based consensus (including) PAXOS/ RAFT-based derivatives
Round Robin
Federated consensus
Proprietary distributed ledger
PBFT and derivatives
Permissionless
Permissioned
Types of distributed mechanism

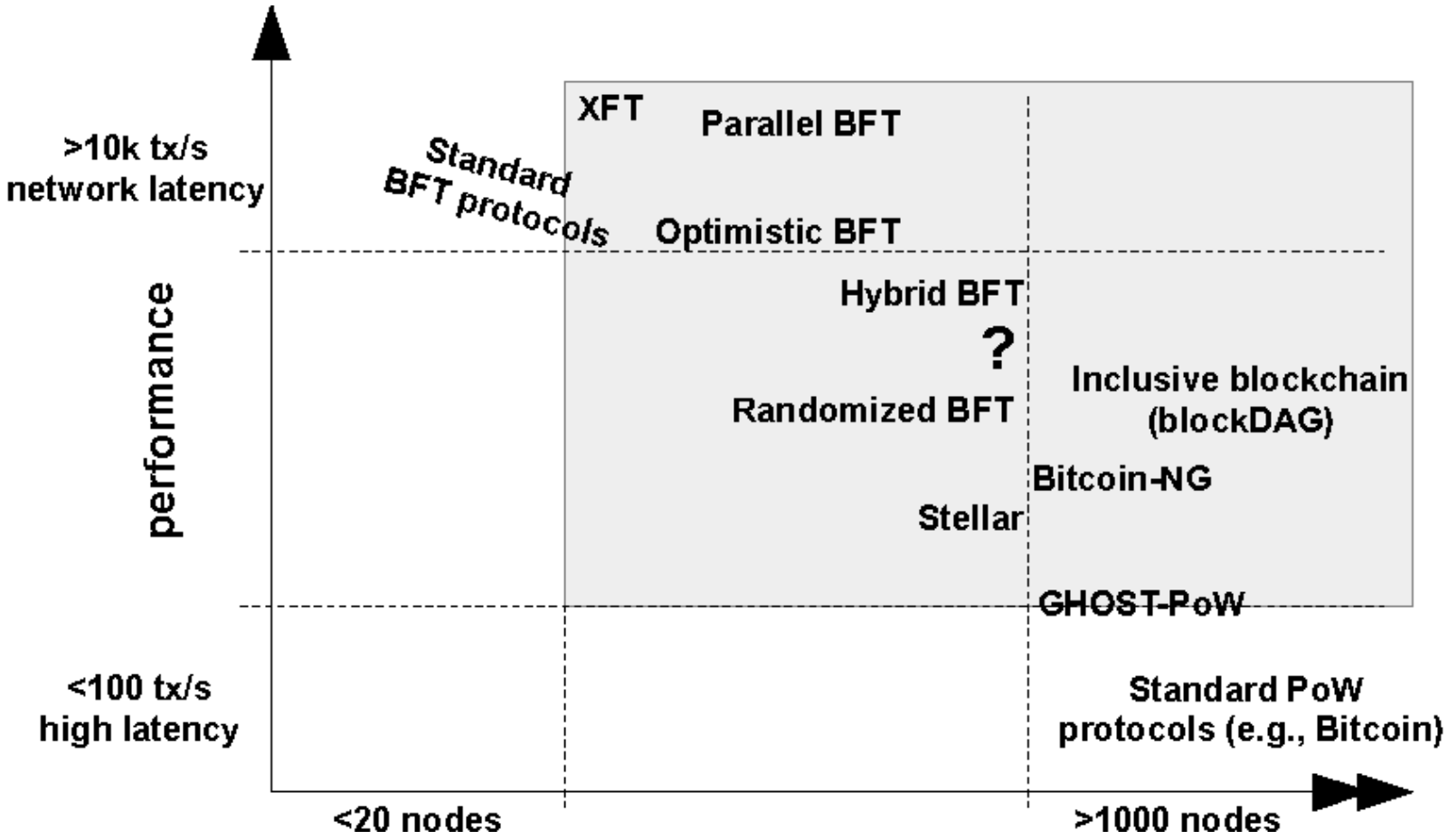
Consensus Mechanisms

	PoW (Bitcoin)	PoS (Ethereum)	PoET (Intel SawTooth)	BFT & Variants (Hyperledger)	Federated BFT (Ripple, Stellar)
Blockchain Type	Permisionless	Both	Both	Permissioned	Permissioned
Transaction Finality	Probabilistic	Probabilistic	Probabilistic	Immediate	Immediate
Transaction Rate	Low	High	Medium	High	High
Token Needed?	Yes	Yes	No	No	No
Cost of Participation	Yes	Yes	No	No	No
Number of Peer Nodes	High	High	High	Low	High
Trust Model	Untrusted	Untrusted	Untrusted	Semi-trusted	Semi-trusted

Source: Whitepaper Understanding Blockchain Consensus Models

Split into groups

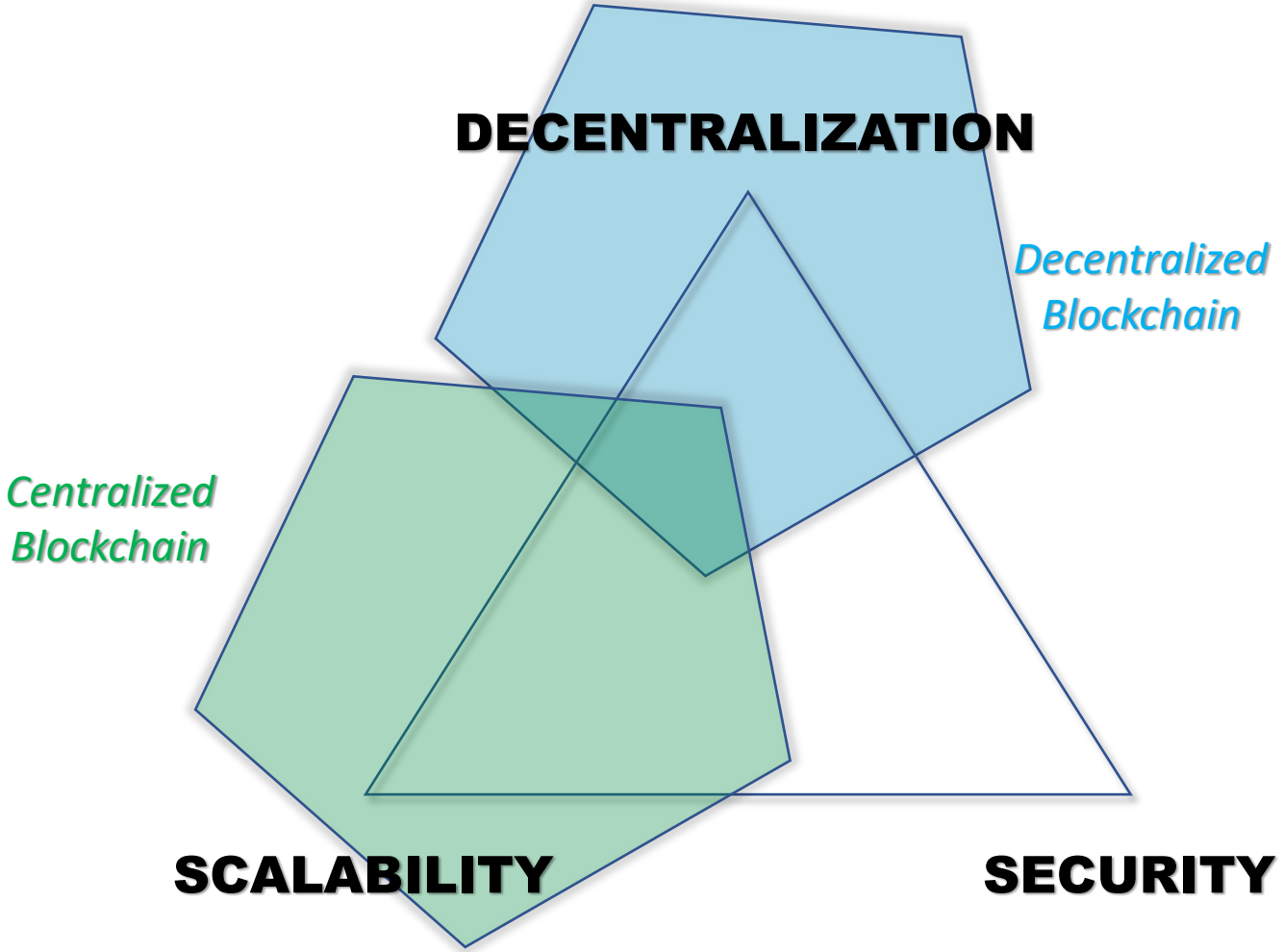
Scalability



Marko Vukolic, IBM research in Switzerland (excellent paper to read - link)

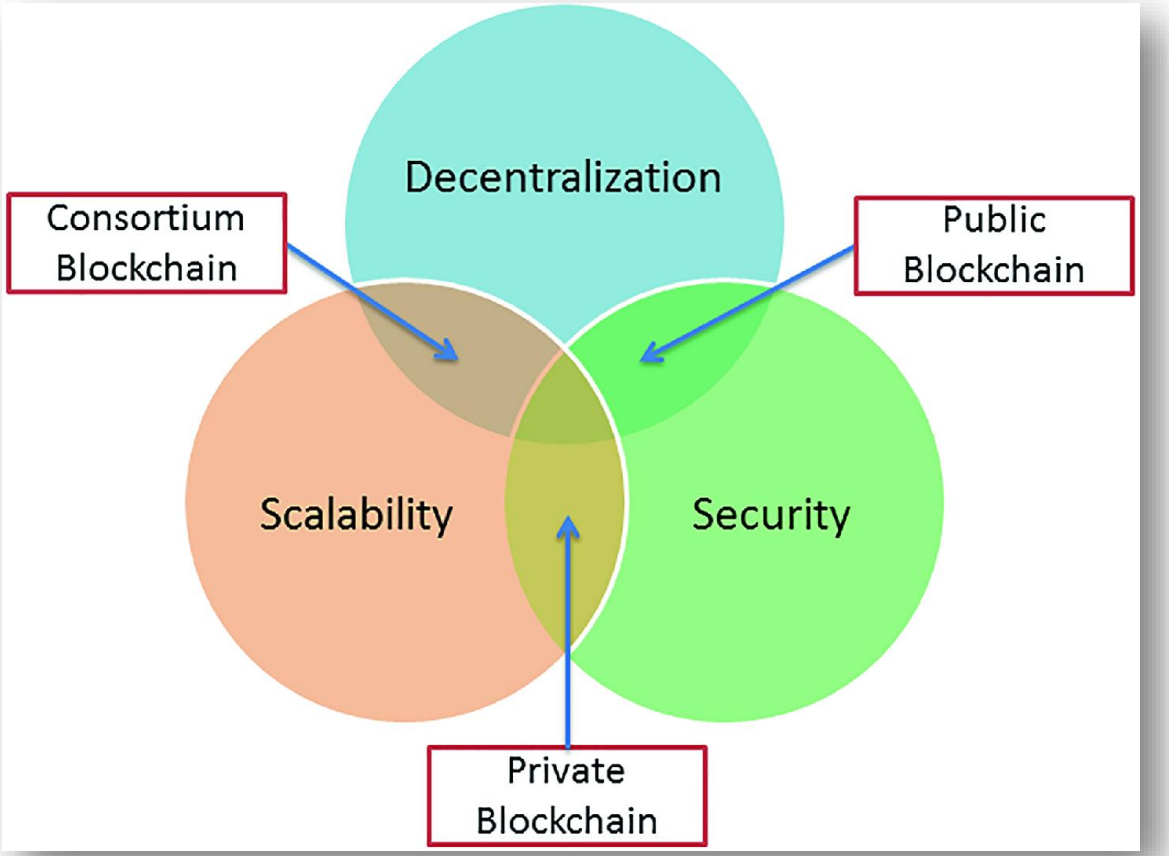
Blockchain Trilemma

[Zooko's Triangle – can't have all 3/3, chose 2/3]



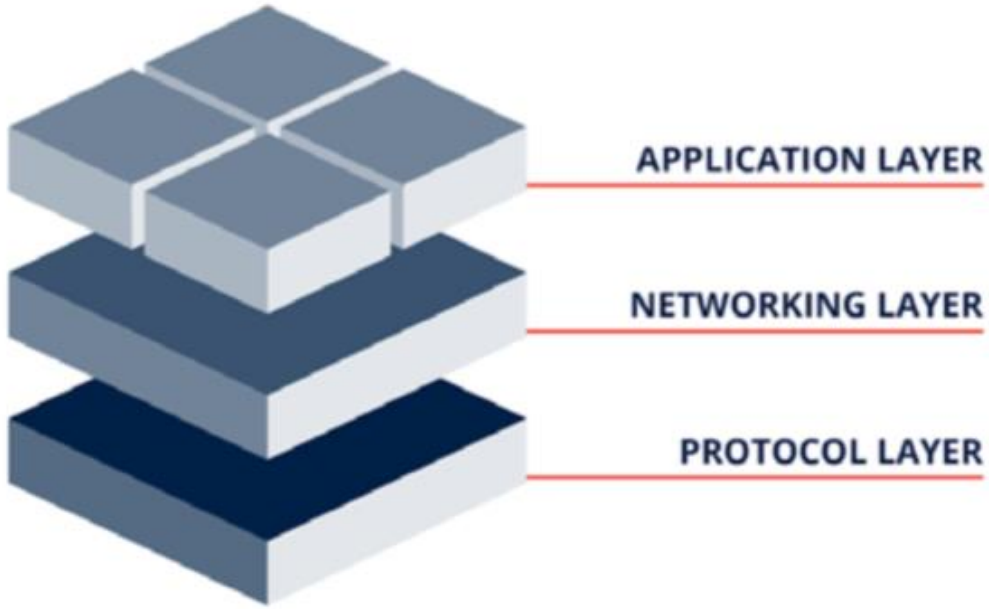
Blockchain Trilemma

[Zooko's Triangle – can't have all 3/3, chose 2/3]



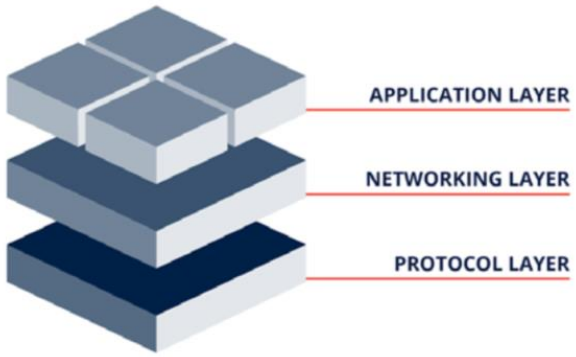
RELAX
REFRESH
RECHARGE
Notes

Blockchain Layer Architecture



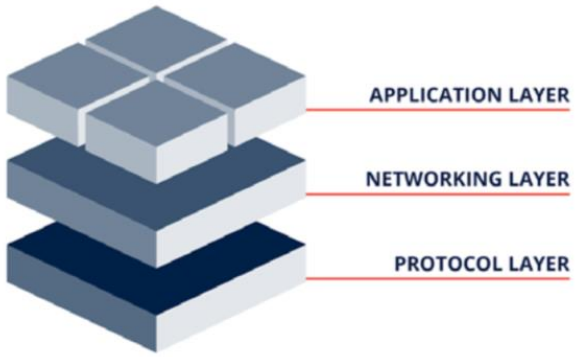
APPLICATION Layer

- **Who** is going to **use the application**? What are the **implications for user experience and design**?
- What is the **existing organizational structure**? And what **behavioral patterns to users have** today? How does this product or service fit into their existing workflow?
- Are there any behavioral or organizational changes necessary to implement this use case?



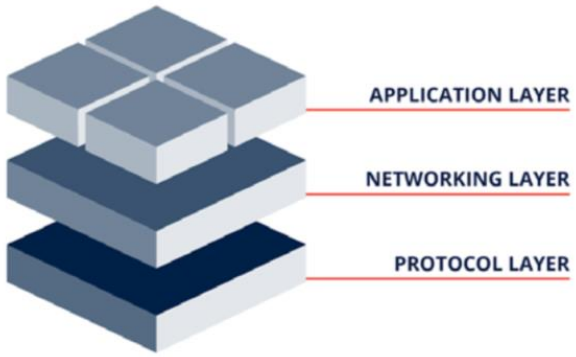
NETWORK Layer

- **Who needs to run a node**? Who has **read access**? Who has **write access**?
- What are the **technology integration requirements**?
- What are the **data storage requirements** regarding archiving and regulation?



PROTOCOL Layer

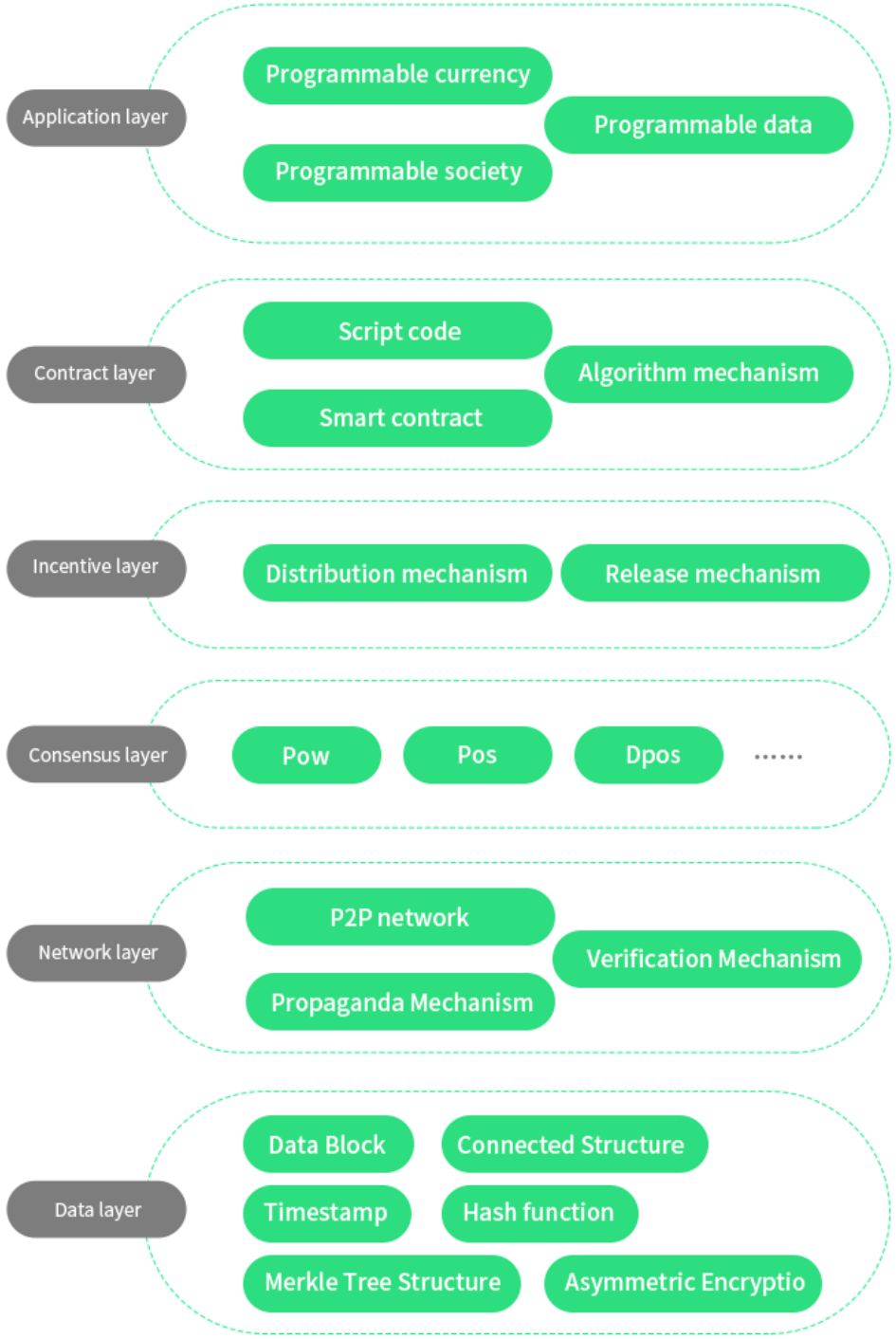
- Is it possible to use **public** blockchains? Or is there a defined need for a **private** implementation?
- What are the **design expectations** regarding **speed**, **programmability**, or **payment** functionality?
- Do you have **developer resources available**? Or is the protocol you're using supported by a robust, sustainable **open-source** developer community with access to resources?



Blockchain Layers

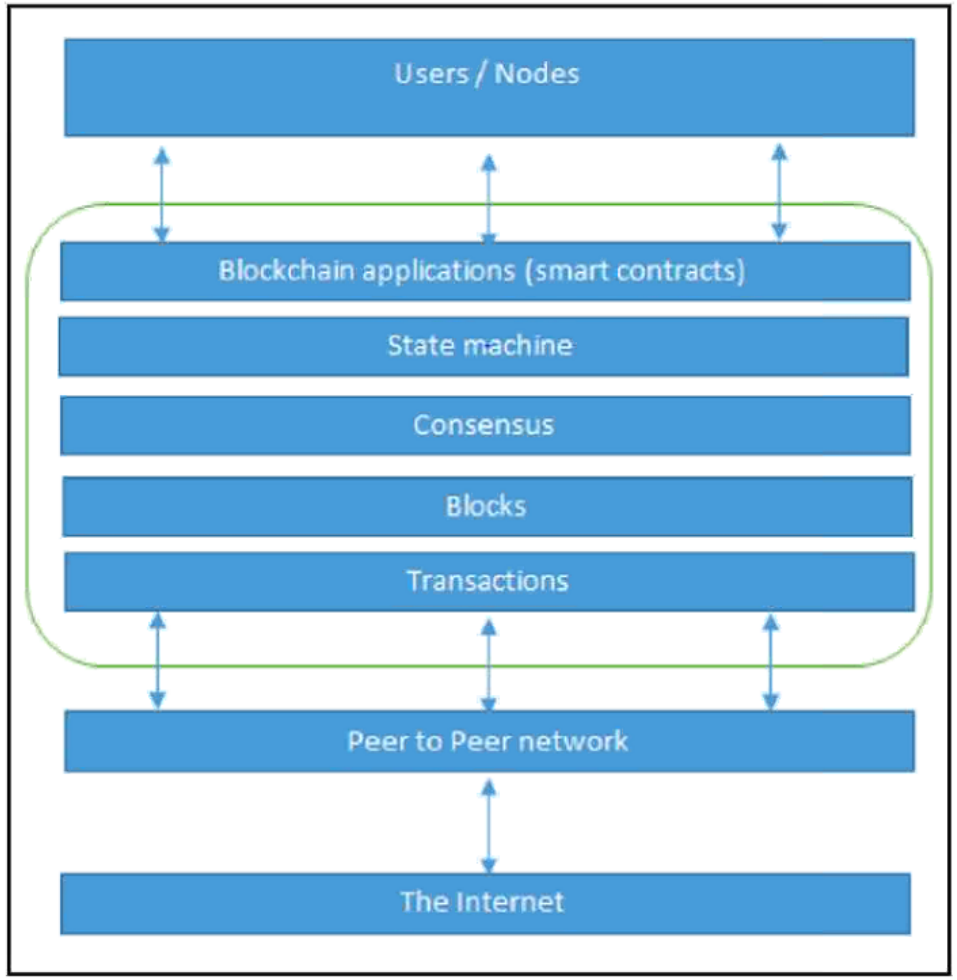
[different approach]

- The blockchain infrastructure is divided into 6 layers, including data layer, network layer, consensus layer, incentive layer, contract layer, and application layer. Each layer completes a core function, and each layer cooperates to achieve a decentralized trust mechanism.



Blockchain Layers

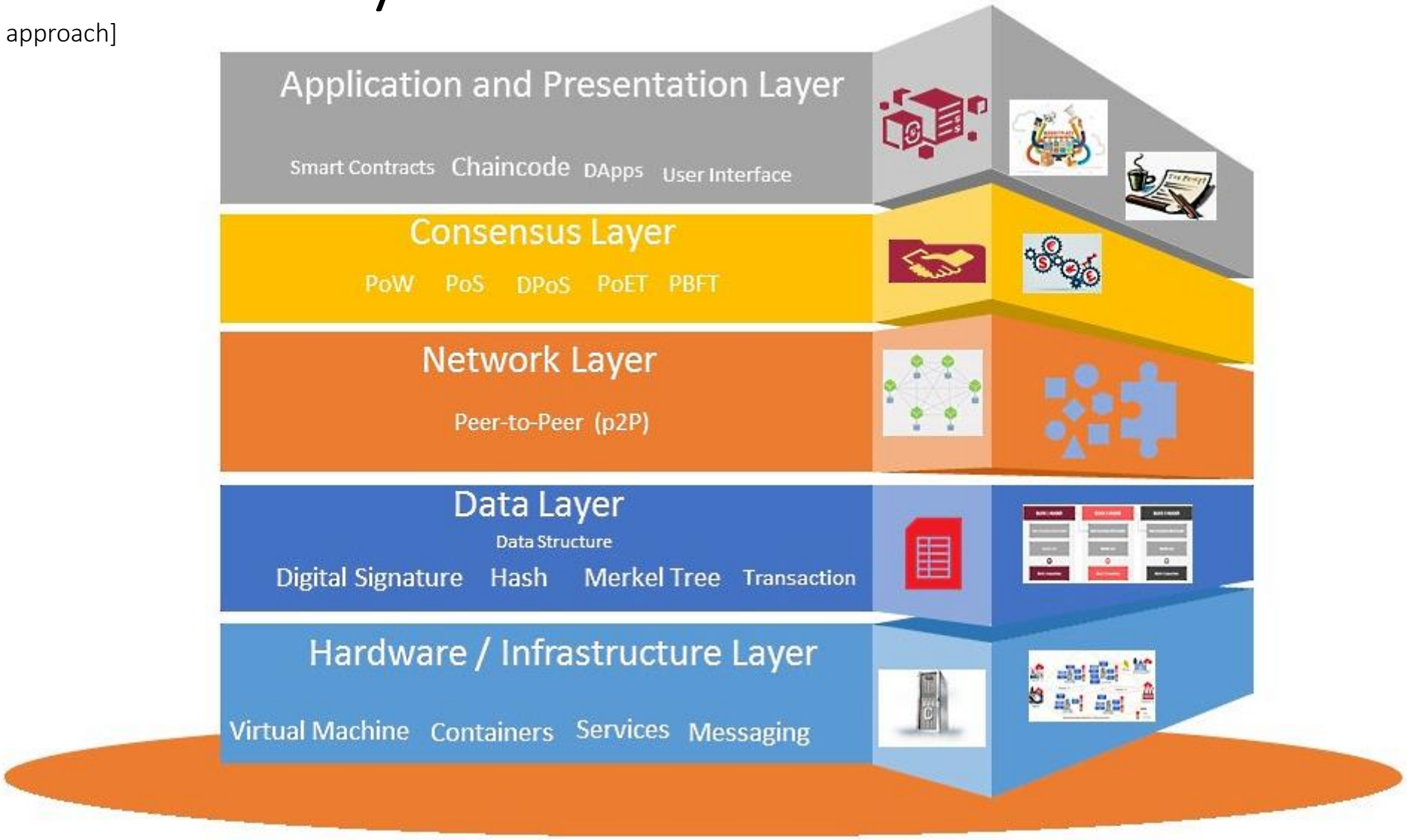
Network view of a Blockchain



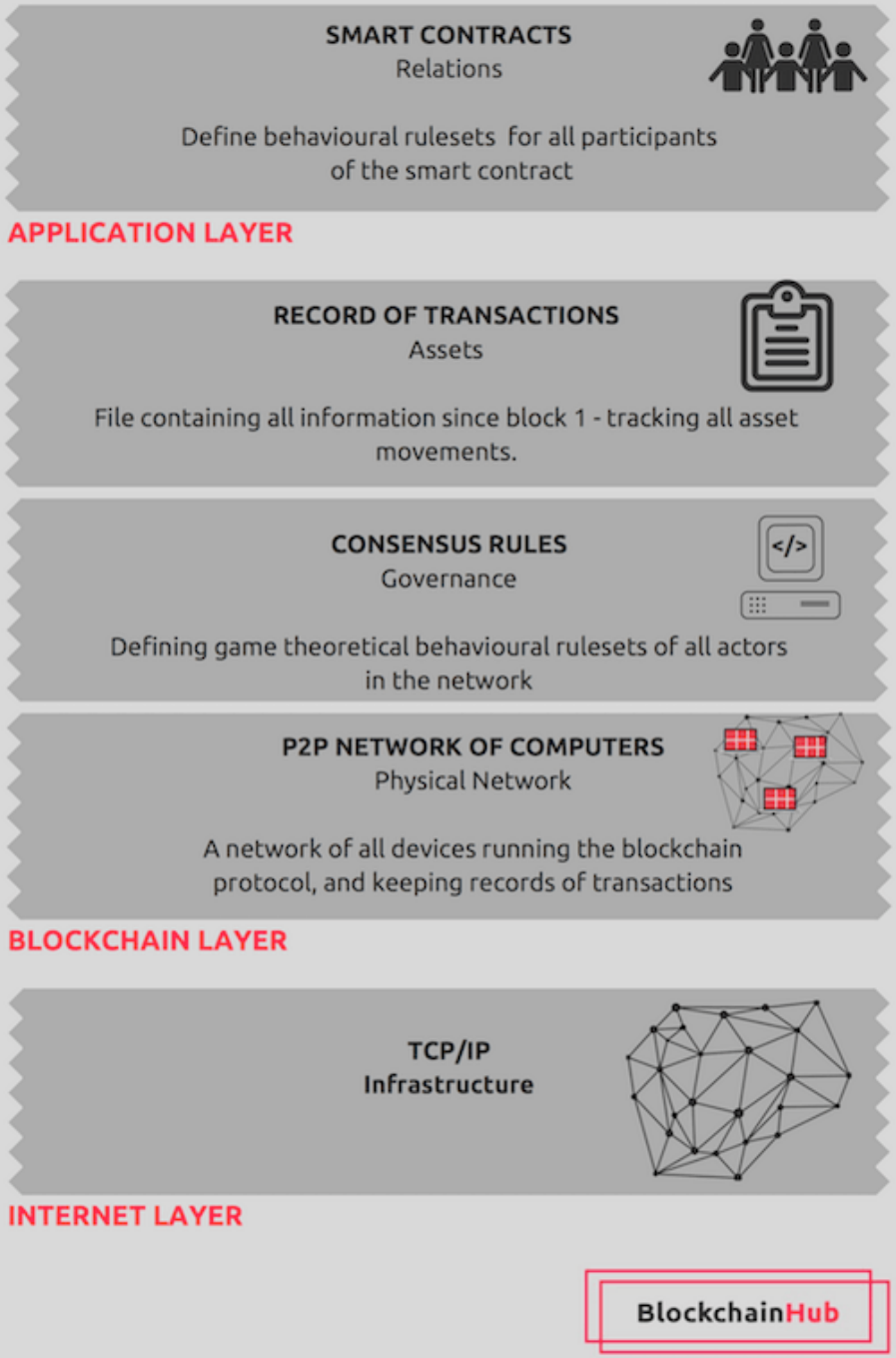
Bashir, I., 2017. *Mastering blockchain*. Packt Publishing Ltd.

Blockchain Layers

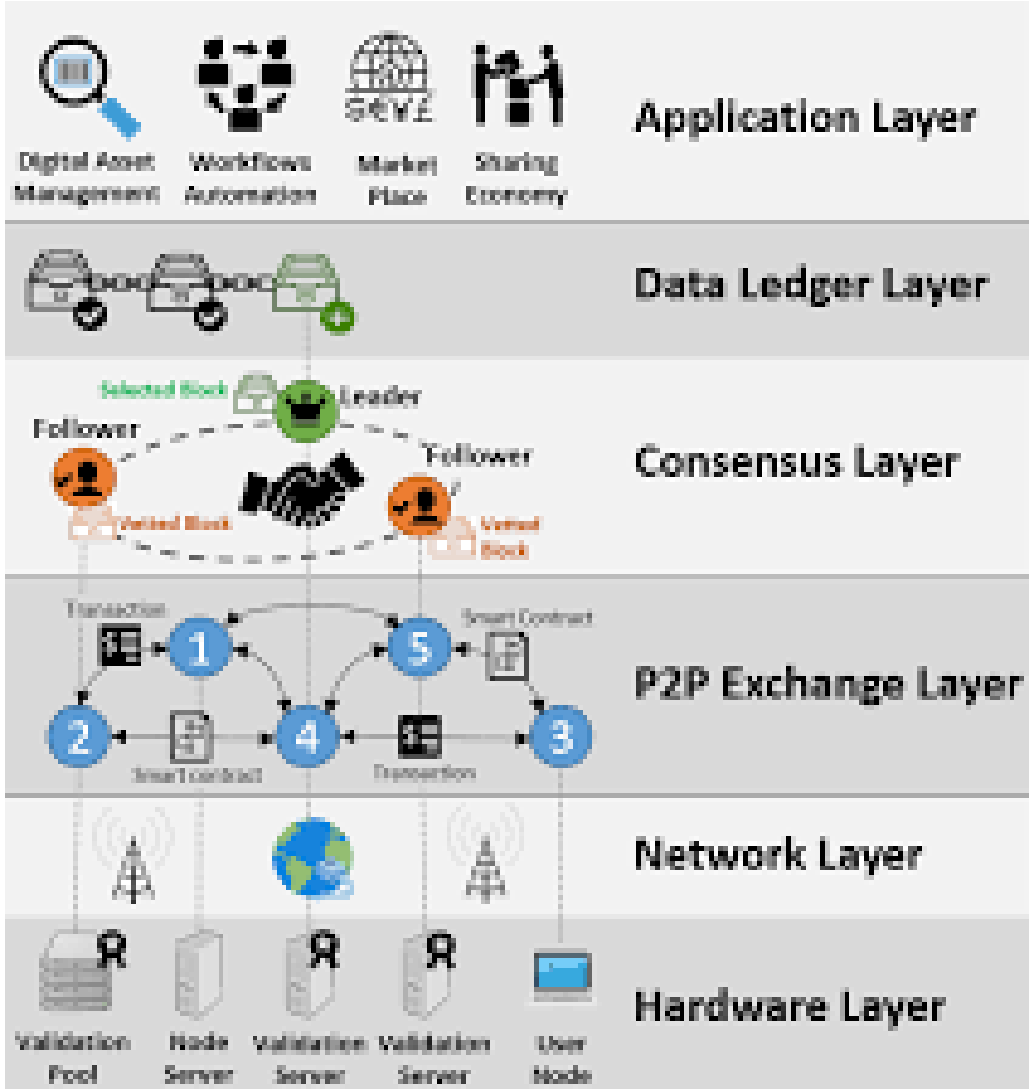
[different approach]



More Layers…

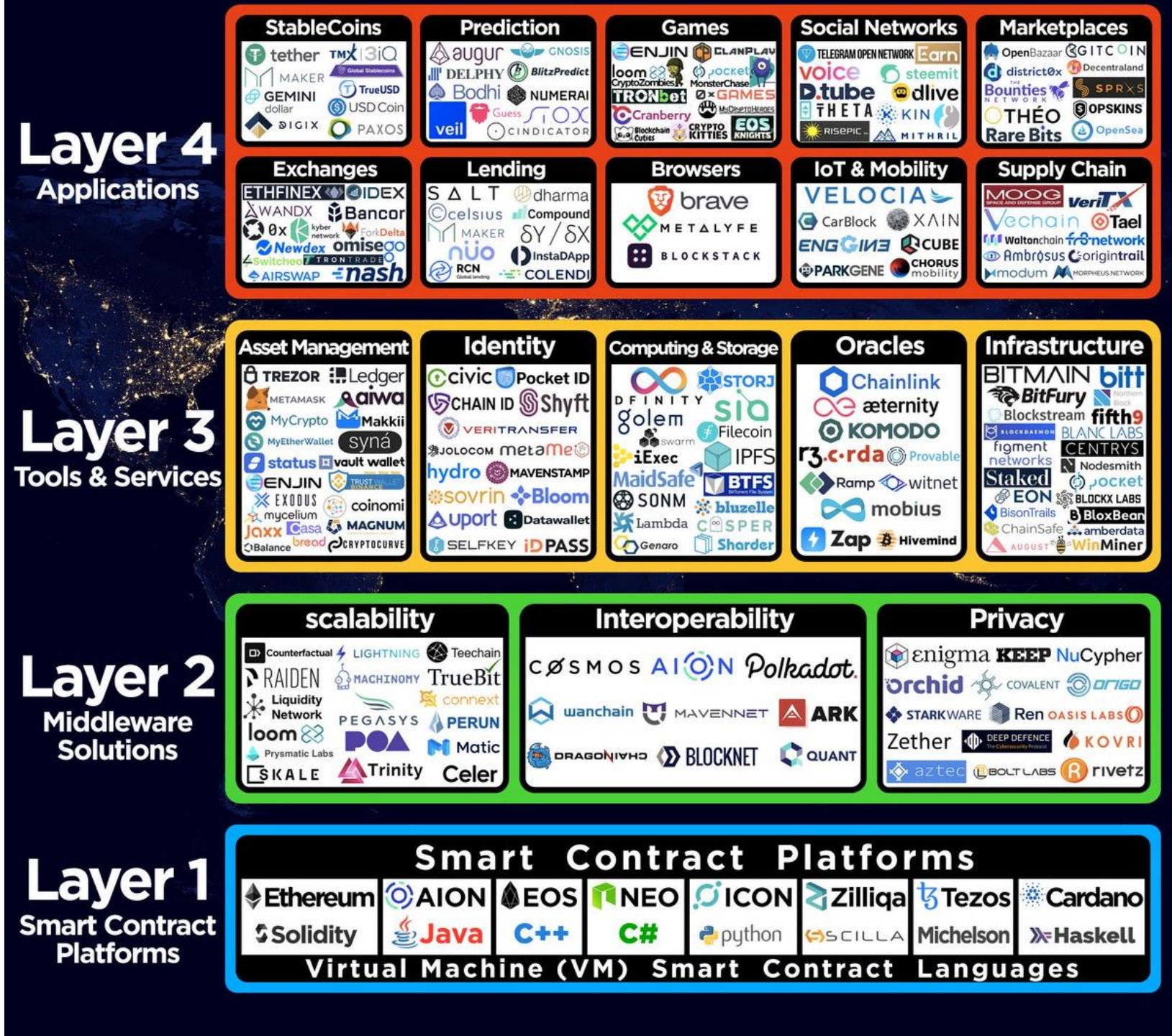

SMART CONTRACTS
Relations
Define behavioural rulesets for all participants of the smart contract
APPLICATION LAYER
RECORD OF TRANSACTIONS
Assets
File containing all information since block 1 - tracking all asset movements.
CONSENSUS RULES
Governance
Defining game theoretical behavioural rulesets of all actors in the network
P2P NETWORK OF COMPUTERS
Physical Network
A network of all devices running the blockchain protocol, and keeping records of transactions
BLOCKCHAIN LAYER
TCP/IP
Infrastructure
INTERNET LAYER
BlockchainHub

More Layers…

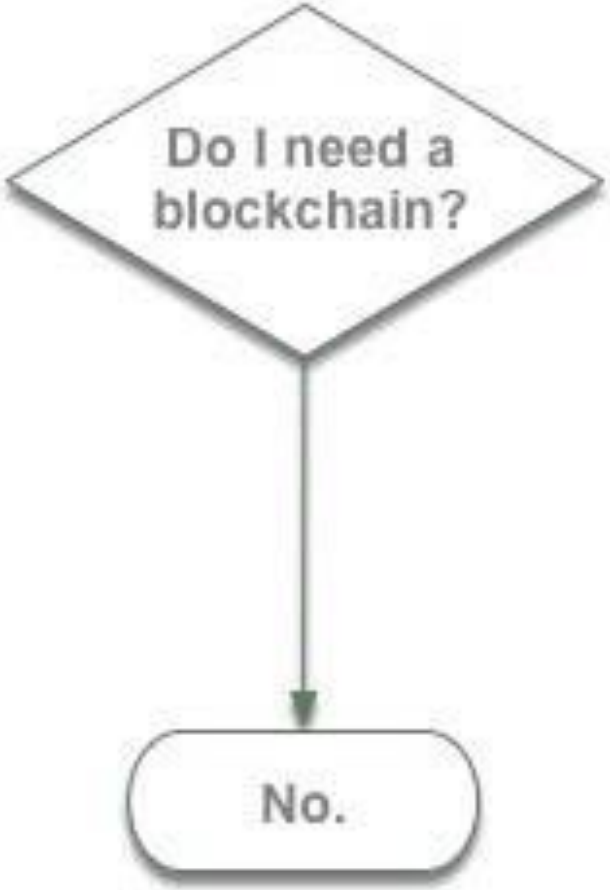

Digital Asset Management
Workflows Automation
Market Place
Sharing Economy
Application Layer
Data Ledger Layer
Selected Block
Leader
Follower
Follower
Voted Block
Voted Block
Consensus Layer
Transaction
Smart Contract
1
5
2
4
3
Smart Contract
Transaction
P2P Exchange Layer
Network Layer
Validation Pool
Node Server
Validation Server
Validation Server
User Node
Hardware Layer

Blockchain ecosystem

[blockchain technology stack]



When do you need a blockchain?



Blockchain Strategy Framework

[KEY CRITERIA - Oxford]

Answer these questions to determine whether blockchain could help solve your use case:

1. Is there a predictable, reputable process that lends itself well to automation?
2. Is there an ongoing or long-running transaction or process, rather than a process that occurs only once?
3. Are there multiple stakeholders in this process or value chain?
4. Is the role or reconciling disparate data usually played by one party or a limited number of parties?
5. Remembering that value is not only monetary, is there an element of value transfer?
6. Is there value in an immutable record? Or is an immutable record a requirement?

If you answered YES to any of the key criteria, consider the following questions to start planning how you would use blockchain:

Blockchain Framework

Additional set of questions that ask for specific relating to blockchain

Who are the users?	Do they want something that they could not have done without a blockchain? ◆ Are they primarily concerned that heir data is at risk? ◆ Are they anticipating reduced cost? ◆ What performance expectations, like transaction latency, do they expect? ◆ Will users have access restrictions to the blockchain?
What data do users input?	Do users enter data sets or transactional data with pointers that are stored elsewhere?
Who are the peers?	Who can become a peer? ◆ How many peers are necessary? ◆Who are initial peers? ◆ How much trust is there amongst the peers? ◆ Will peers have access restrictions to the blockchain? ◆ Are some peers also peers in other blockchain networks?
How do peers create blocks?	How many transactions or contract or whatever go into a block? ◆ When are blocks formed? ◆ Do blocks have size limitations?
What do peers validate?	Do peers validate that the data satisfies certain conditions aside from syntactical (syntax) constraints and/or proof-of-something? ◆ Is validation deterministic or do the peers sometimes have to make judgement calls?
How do peers reach consensus?	What information do peers need in order to validate? ◆ Do they need the blockchain to be able to validate? ◆ Do they need the entire blockchain?
How are peers incentivized?	Do peers compete for rewards? ◆ Do users compete to get peers to service their inputs?

The Value of Blockchain

Automation and trust through Smart Contracts	Reduce friction and improve efficiency by automatically executing business logic agreed upon by the participants
Provenance	Guarantees quality and lineage vie the immutable ledger
Privacy	Participants have visibility into transactions that is commensurate to their role in the network
Growing the Business Network	New participants of different types can join the network and provide value and new possibilities

Quotes

"There are no experts in blockchain, just fools at different levels of discovery"

— Samson Williams

"What happens on blockchain, stays on blockchain"

— Kris Bennett

Short video clips…

- What is blockchain? (6min) [LINK]

❖Blockchain USE CASES…

- Cross-border Supply Chain, Logistics (IBM & Maersk | Hyperledger) [LINK]
- Seafood Supply Chain, Provenance (Hyperledger Sawtooth) [LINK]
- Walmart & IBM (3min) [LINK]
- Money/crypto (3min) [LINK]
- Financial Institutions/Business+ (6min) [LINK]
- Estonia Government: e-residency [LINK]; e-health (5min) [LINK]
- Dimond's (1.5min) [LINK]
- Royal Mint (2min) [LINK]

Data and blockchains

- Blockchains are **not designed or meant for data storage** and storing large documents would be very expensive.
- What blockchain hold is the **hash of the data**, and uses it as a **reference to the data (fingerprint)**.
- Blockchains might be used to maintain a Distributed Hash Table (DHT), which contains hashes of data files stored off-chain.
- Blockchain-like solutions designed just to store data were developed recently:
  - STORJ.IO
  - SIA-coin
  - FILECOIN
  - IPFS
  - MaidSafe
  - Sharder
  - BigchainDB

Split into groups

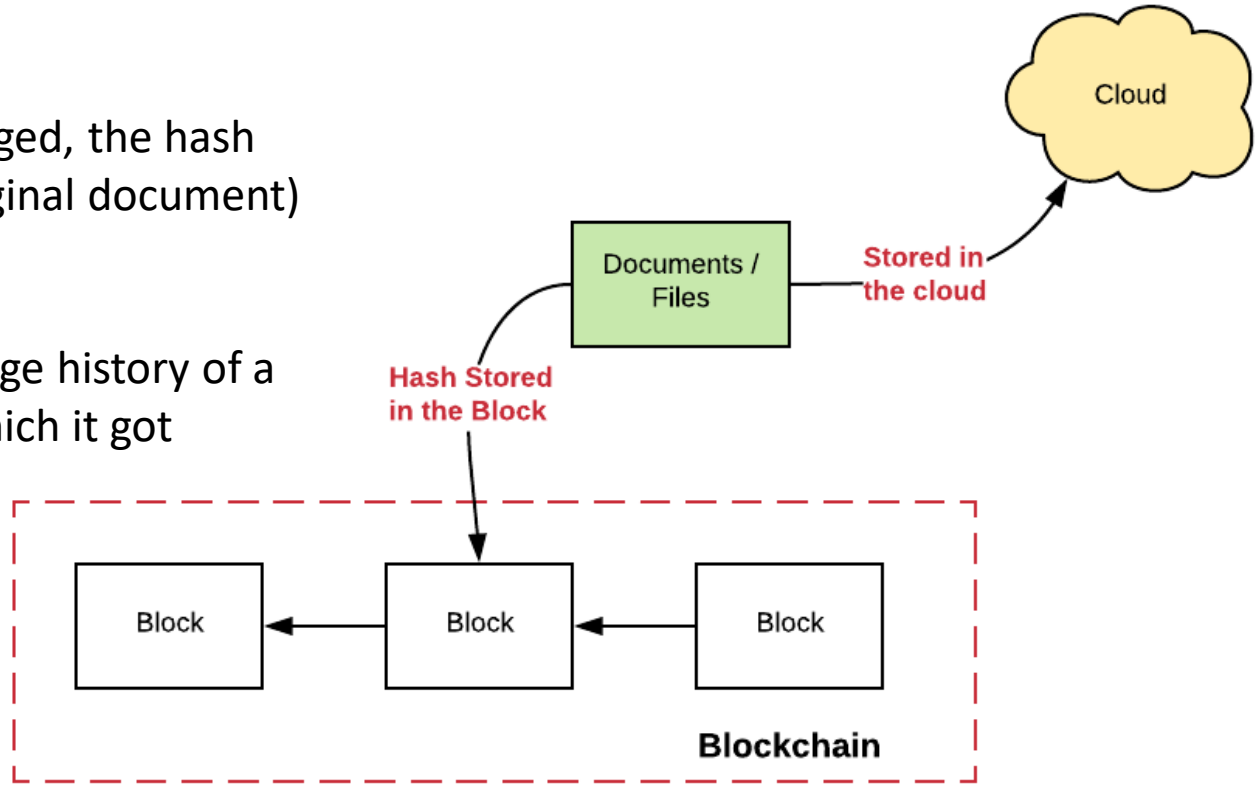
Benefits of storing hashes on blockchain

- Brings the general benefit of the Blockchain in relation to the document, such as **integrity, non-repudiation, authentication** etc.

- Store Document related information: As part of the transaction, information such as **who created the document, version, timestamp** etc. also gets recorded.

- **Minimize the fraud**: In case the document is changed, the hash value won't match with the hash value (of the original document) stored in the blockchain.

- **Document history using the timestamp**: The change history of a particular document vis-a-vis the date/time on which it got changed gets recorded as the digital timestamp.



Side-Chains/Off-Chain

- Emerging mechanism that allow tokens and other digital assets to **move from one blockchain (main chain) and to be securely used in a separate blockchain** and then moved back to the original blockchain if needed.
- A side chain is a **separate similar to blockchain** tech that is attached to its parent blockchain using a **two-way peg**. The two-way peg enables interchangeability of assets at a predetermined rate between the parent blockchain and the sidechain.

Cost of Data

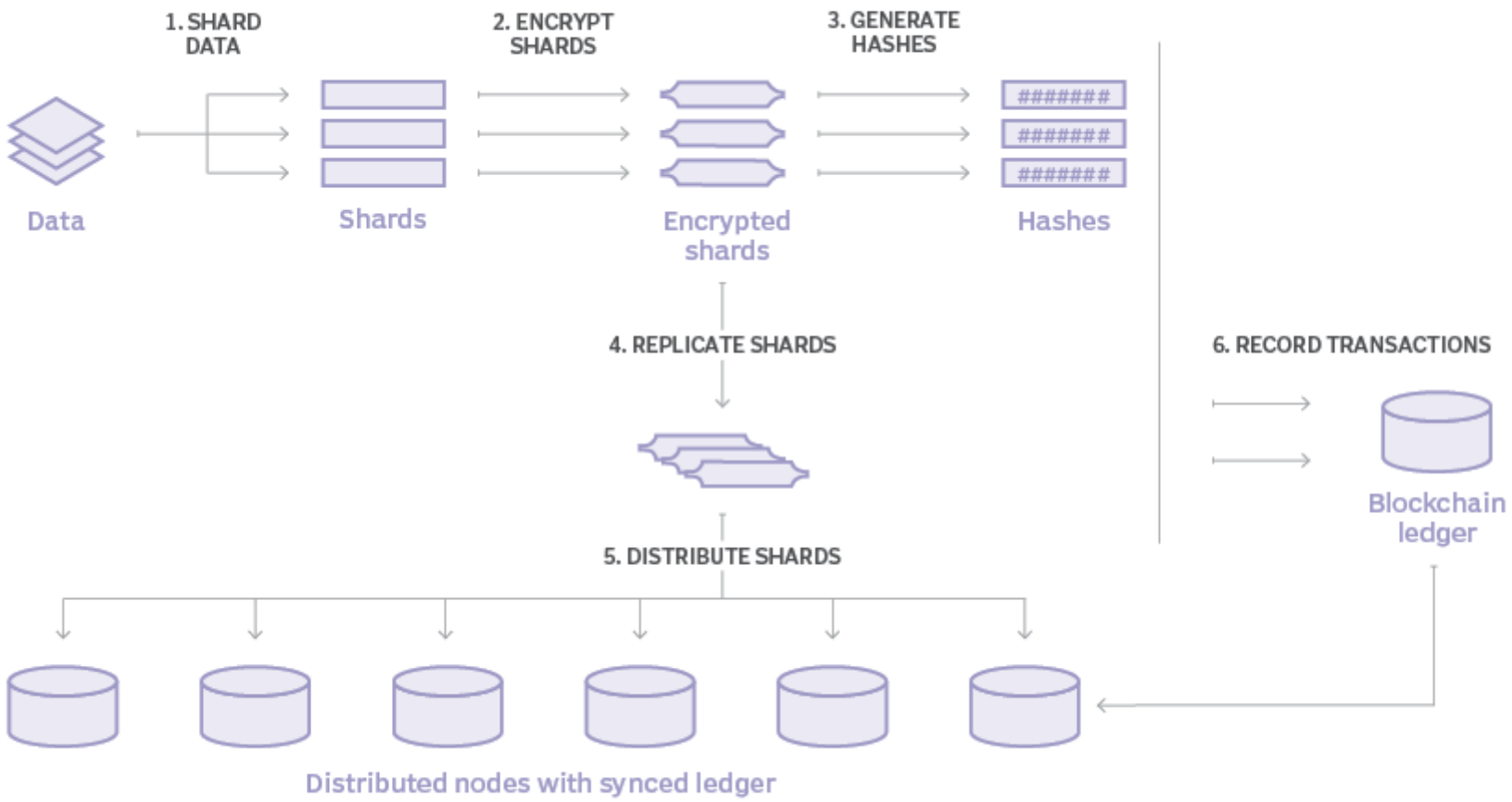
Year 2016:

- <u>Storage costs:</u>
  - This is Solidity code for creating a contract with 1 Kilo Byte of data.

```
contract Storage { byte[1024] data; function Storage() { for (uint i = 0; i < 1024; i++) { data[i] = 'A';}}}
```

  - If we run this code in the online compiler, we get an estimated transaction cost of 5925085 gas. The gas price today is 23731285772 Wei (10^18 Wei = 1 Ether = **$15**). So, **storing 1 Kilo Byte of data in the public blockchain**, as per conversion rates on is approximately **$2.11**.
- <u>Reading costs:</u>
  - Similarly, r**eading 1 Kilo Byte** of data costs 284396 gas, which is approximately **$0.1**.
  - This **price might increase if there is an increase in the Gas value or Ether value**.

How blockchain storage works



Break & Quick Recap

Public Blockchain

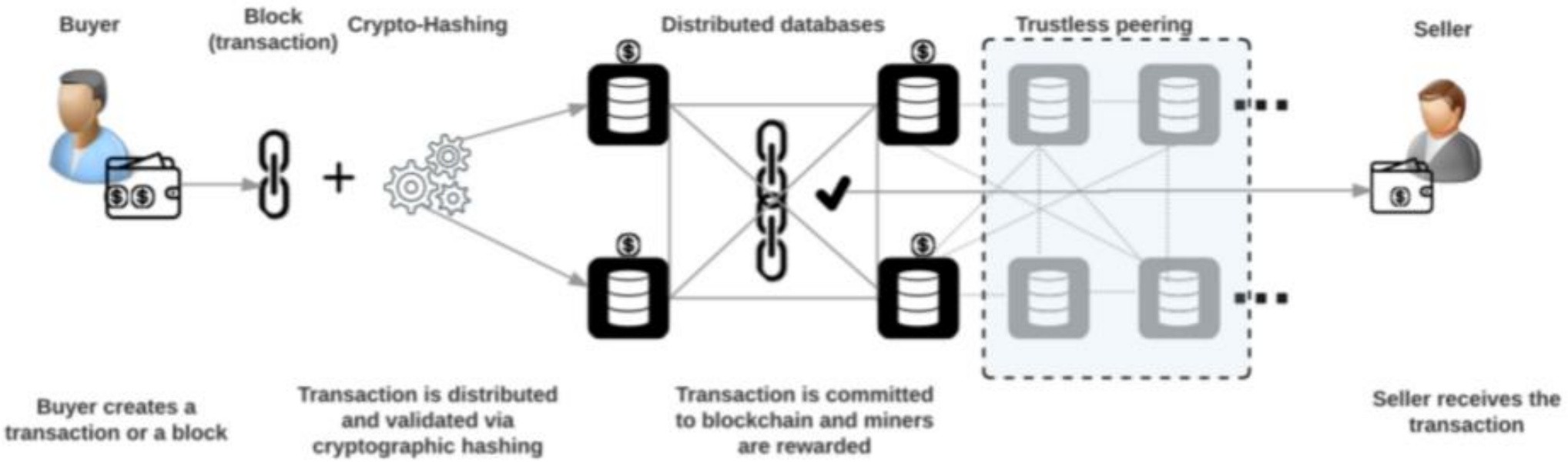
[permission-less]

- **Anyone is allowed to participate** in the network
- Anyone can indulge in the network as a node and can conduct transactions. **Ledger are not maintained by anyone alone** and is completely opened to anyone.
- **Network is completely decentralized**, user can download the software and track the record.
- **Data is shared among participants**, and anyone can see and participate in the ongoing transactions.
- However, the **distributed consensus mechanism is being maintained** in order to reach the final decision.
- Transactions are carried **without the need of a 3rd party**, therefore eliminates the third party;
- **Each transaction need to be verified by thousands of participants**, therefore the verification process becomes **time-consuming**.

Source: beingcrypto.com

Public Blockchain

[permission-less]



Source: beingcrypto.com

Private Blockchain

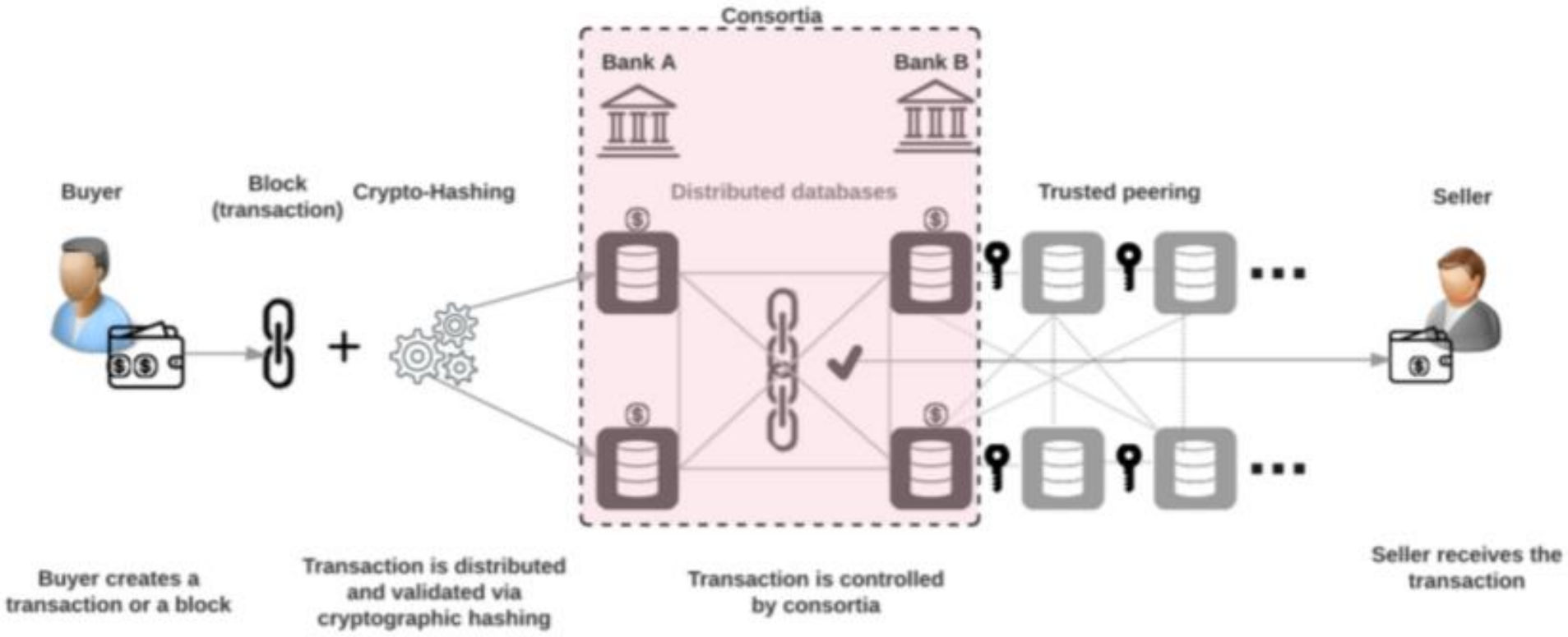
[permissioned]

- **Opened to a group of ‘entities’** or consortium that has **agreed to share the ledger** with each other.
- Mutually agreed to **trust the validators** in the network (trusted and certified).
- Organization that runs the blockchain is given the authority to allow participants to **read a particular transaction** as all transaction are not necessary for participants to be read.
- Private network gives **more privacy** to the members.

Source: beingcrypto.com

Private Blockchain

[permissioned]



Source: beingcrypto.com

Consortium Blockchain

[Hybrid Model]

	PUBLIC Blockchain	PRIVATE Blockchain	CONSORTIUM/ENTERPRISE/ FEDERATED/HYBRID Blockchain
ACCESS	• **Anyone**	• **Single** Organization	• **Multiple** selected organizations
PARTICIPANTS	• Permissionless • Anonymous	• Permissioned • Known Identities	• Permissioned • Known identities
SEURITY	• Consensus mechanism • Proof of Work/ • Proof of Stake	• Pre-approved participants • Voting/multi-party consensus	• Pre-approved participants • Voting/multi-party consensus
TRANSACTION SPEED	• Slow	• Lighter and faster	• Lighter and faster

Private vs Public – Final thoughts

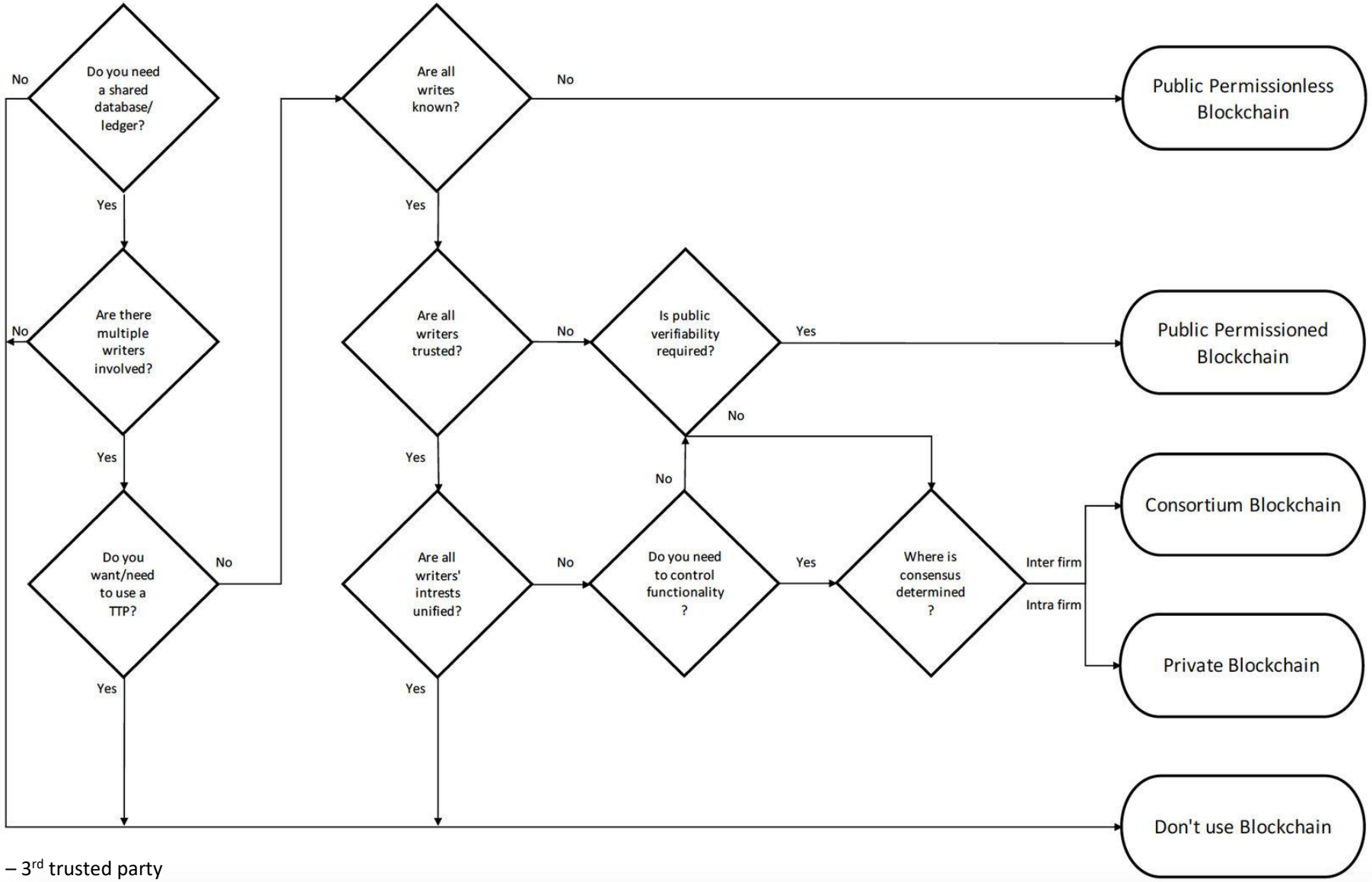
- Public has **miners**; Private has **validators**;
- Public miners are rewarded, validators maybe;
- Private users are known (**KYC**), while in public users are **anonymous** – question of identity and understanding who the user is and to what information should they have access to;
- **Consensus Mechanism** is public is energy consuming (expensive);
- Private is much **quicker and secure** as only limited by number of people are taking part in the network.

- *There's a perception that public blockchain platforms like Ethereum can't be used to build permission scenarios or to control access to data.*
  The truth is that they can, they just don't give you all the built-in tools that you can find on a private or permission blockchain platform. But basically, you can always use these open public platforms to build a permission solution, you just need to be aware that it's upon you, your architects and your developers to create that permissioning model, and that all starts with some kind of identity management system.
- *A lot of people get the impression that they compete with one another, but they really don't. They just serve to provide different types of solutions.*

Source: medium.com

Private vs Public – Final thoughts

[DIAGRAM]



*TTP – 3rd trusted party

© Exterkate & Wagenaar 2018

Blockchain Components

Ledger	contains the current world state of the ledger and a Blockchain of transaction invocations
Smart Contract	encapsulates business network transactions in code. transaction invocations result in gets and sets of ledger state
Consensus Network	a collection of network data and processing peers forming a Blockchain network. Responsible for maintaining a consistently replicated ledger
Membership	manages identity and transaction certificates, as well as other aspects of permissioned access
Events	creates notifications of significant operations on the Blockchain (e.g. a new block), as well as notifications related to smart contracts. Does not include event distribution.
Systems Management	provides the ability to create, change and monitor Blockchain components
Wallet	securely manages a user's security credentials
Systems Integration	responsible for integrating Blockchain bi-directionally with external systems. Not part of Blockchain, but used with it.

Source: beingcrypto.com

Blockchain Components

[MORE]

COMPONENT	DETAILS
LEDGER	• Contains all the record of the transaction and participants. • 2 types of ledgers (public and private).
SMART CONTRACT	• Like a traditional document that defines rules and regulations (events) but it automatically enforces those obligations. (i.e. vending machine) • Reduces fraud and illegal activities.
CONSENSUS MECHANISAM	• Consensus refers to aggregate decision or agreement between different people over a group of rules and law which run a blockchain.
MEMBERSHIP	• All participants on the blockchain network are given permission to use the it by assigning a unique identity.
EVENTS	• Useful as it gives notifications of all the important operations on the blockchain network.
SYSTEM MANAGEMENT	• A system management or manger define, create modify or remove the rules (broad responsibility). • Give authority to make sure that error-free operations happen on the blockchain.
WALLET	• Each member of the blockchain is being given a wallet which securely maintains all records of the user. • Types of wallets: hardware, paper, desktop, mobile, and web.
SYSTEM INTEGRATION	• Oracle, AI, IoT, BaaS, etc.

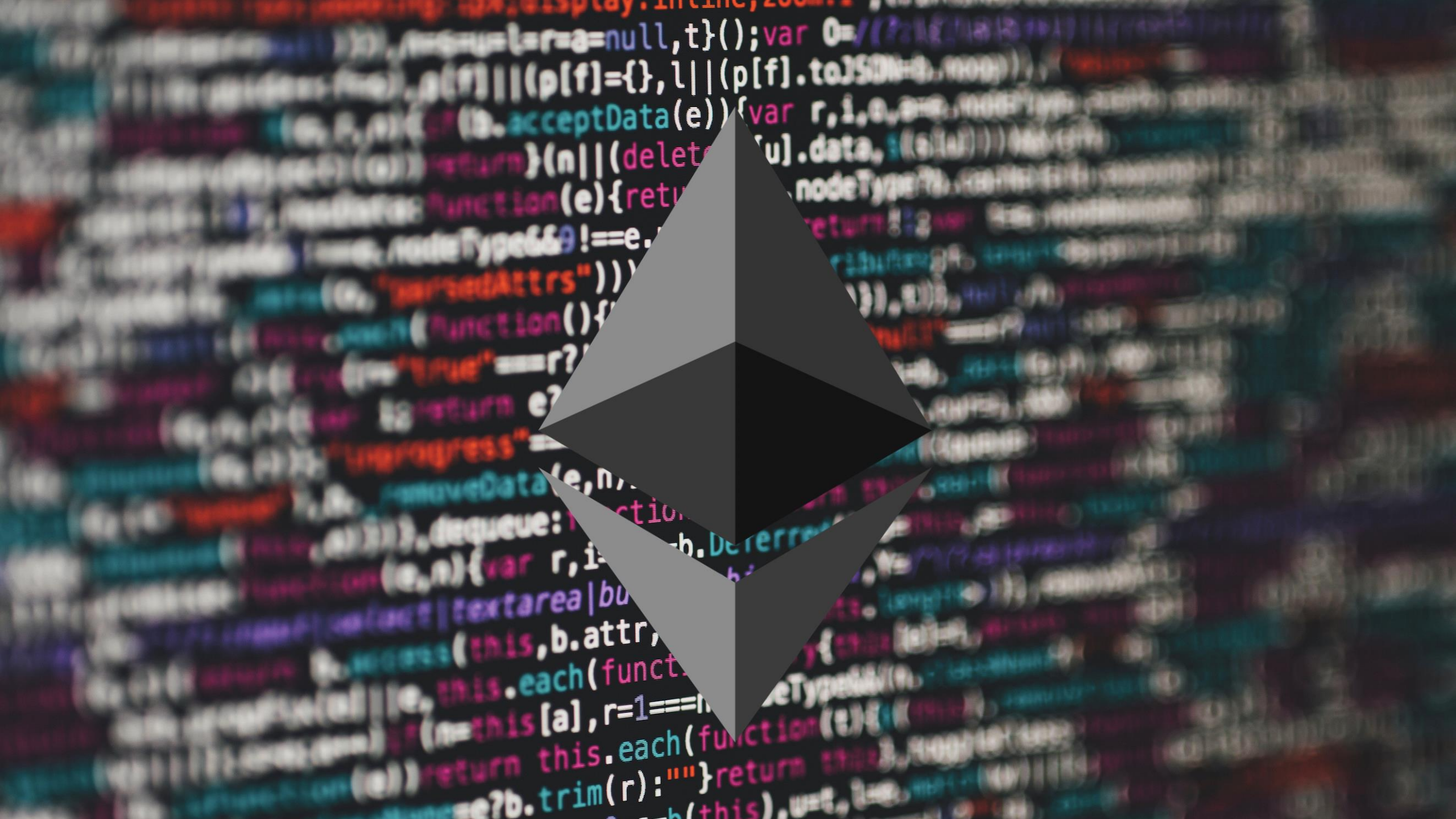
Source: beingcrypto.com

Additional Material

- Do you need blockchain? (paper) – [LINK]

- BBC Panorama – Just how do cryptocurrencies work (public, PoW) (3min) [LINK]
- BBC Newsnight - How does bitcoin mining work? (7min) [LINK]

- Proof-of-Work vs Proof-of-Stake vs Proof-of-…
  - Consensus: Proof-of-…. [LINK]
  - PoW vs PoS by Vitalik [LINK]
  - Understanding Consensus Mechanisms [LINK]
  - Practical Byzantine Fault Tolerance (PBFT) used by Hyperledger (will cover later)



Ethereum

[Blockchain 2.0]

- What is Ethereum
- Features of Ethereum
- Smart Contracts
- Decentralized Autonomous Organizations (DAO)

What is Ethereum

- Ethereum is an **open-source, public, blockchain-based distributed computing platform** featuring smart contract functionality. “A planetary scale computer built on blockchain technology”
- It is **not limited to crypto-currencies**; a programable blockchain using **smart contracts**.
- A **decentralized platform that runs smart contracts**: application that run exactly as programmed **without any possibility of downtime, censorship, fraud, or third party interference**.”
- Ethereum Virtual Machine (**EVM**) – runs on Ether, pay per computation step (concept of **gas**).
- Turing-complete - any system or programming language able to compute anything computable given enough resources.

What is Ethereum

- How is it different from bitcoin

	Bitcoin	Ethereum
Concept/Usage	Digital Money – be a global decentralized payment system	Smart Contract – be a decentralized super computer to power dApps
Scripting Language	Turing incomplete	Turing complete
Release date	Jan 2009	July 2015
Coin Release Method	Early Mining (Genesis Block Mined)	Through ICO (Presale)
Average Block time	~10 minutes (6/hr)	~12-15 seconds (215.83/hr)
Throughput	7-8 tps	15-20 tps
Consensus	PoW	PoW (PoS)
Block Reward	6.25 BTC	2 ETH

Ethereum Features

Key factors making Ethereum popular and effective:

- **Crypto-currency**: blockchain implementation and therefore requires its internal crypto called Ether (ETH);
- **Smart contracts:** application that runs on blockchain and process all the information on the blockchain;
- **Decentralized Autonomous Organizations**: decentralized organization that does not require governing;
- **Smart Property**: digitize your property and transform without the hustles for exchanging the property or validating the documents

Ether

- **Token** of the Ethereum blockchain, listed as **ETH** on exchanges;
- **Used to pay for transaction fees and computational services** on the Ethereum network;
- Every time a contract is executed, **Ethereum consumes tokens** (termed as **gas**) to run the computations;
- Providing to little gas will result in failed transaction;
- ETH GAS STATION, [link]

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

ERC standard tokens

DIFFERENT ETHEREUM REQUEST FOR COMMENTS

ERC 20

- Most popular token standard.
- Used in most of the ICOs.
- Fungible token standard.
- Allows the implementation of standard API within a smart contract.

ERC 721

- A standard for non-fungible tokens.
- Allows the implementation of standard API for NFTs within a smart contract.
- Offers functionality to transfer and track NFTs.

ERC 165

- A standard for a method, instead of tokens.
- Covers how interfaces are identified.
- States how any contract can publish the interfaces after the implementation.
- States how to detect when a contract implements ERC-165.
- Covers the way to detect when a smart contract uses any given interface.

ERC 777

- Reduces friction in crypto transactions.
- Not in use, still in EIP phase.
- Gets rid of the double transaction verification of ERC 20.
- Lowers transaction overhead.
- Allows users to reject incoming tokens from a blacklisted address.

ERC 223

- Prevents accidental burns of tokens, a bug in ERC 20.
- Developers can either accept or decline tokens arriving at their smart contract addresses.
- Rejected transactions will fail but won't burn the tokens.
- Not in use, still in EIP phase.

ERC 827

- An extension of ERC 20.
- Wallets and exchanges can reuse tokens.
- Token holders can transfer token while also approving a 3rd party to spend it.
- Not in use, still in EIP phase.

ERC 621

- An extension to ERC 20 standard
- Uses two functions -'increaseSupply' and 'decreaseSupply.'
- Can increase or decrease the token supply.
- Not in use, still in EIP phase.

ERC 884

- Allows companies to use blockchain to maintain share registries.
- Identity verification and mandatory whitelisting of token holders.
- Only whole value of tokens, i.e., no partial value.
- Recording of information regulators mandate.
- Not in use, still in EIP phase.

source: 101blockchains.com

ERC tokens

- **ERC 721** - is a non-fungible token standard that enables the development of unique tokens from the same contract address. The ERC 721 standard is popularly known as the standard for creating NFTs (non-fungible tokens) .
- **ERC 1410** - These tokens correspond to the requirement of securities to contain differing metadata yet allow a level of fungibility. To this effect, the ERC 1410 token standard enables partitioning of the token into a fungible and a non-fungible component.
- **ERC 1643** - Traditional securities are associated with several documents that relate to ownership, rights, and obligations. The ERC 1643 standard takes care of this requirement by enabling a mechanism of simplified transfer of documents for a speedy transfer of legal rights and obligations associated with the ownership of that security token.

Smart Contracts

- Idea from 1990s by **Nick Szabo**;
- Computerized transaction protocol that executes the terms of a contract;
- Most popular scripting language – **Solidity**, Serpent;
- Tools for writing & deploying smart contracts:
  - ✓ **Mist Browser** – It is a tool to browse and use dApps. It is a separate browser that can be used to browse dApps and interact with them.
  - ✓ **Truffle Framework** – Truffle is a popular development framework for Ethereum. It has built-in smart contract compilation, linking, deployment, and binary management.
  - ✓ **Metamask – MetaMask** is a bridge that allows one to visit the distributed web of tomorrow in their browser today. It allows users to run Ethereum dApps right in their browser without running a full Ethereum node.
  - ✓ **Remix Browser** – Remix is a web browser based IDE that allows users to write Solidity smart contracts, then deploy and run the smart contract.

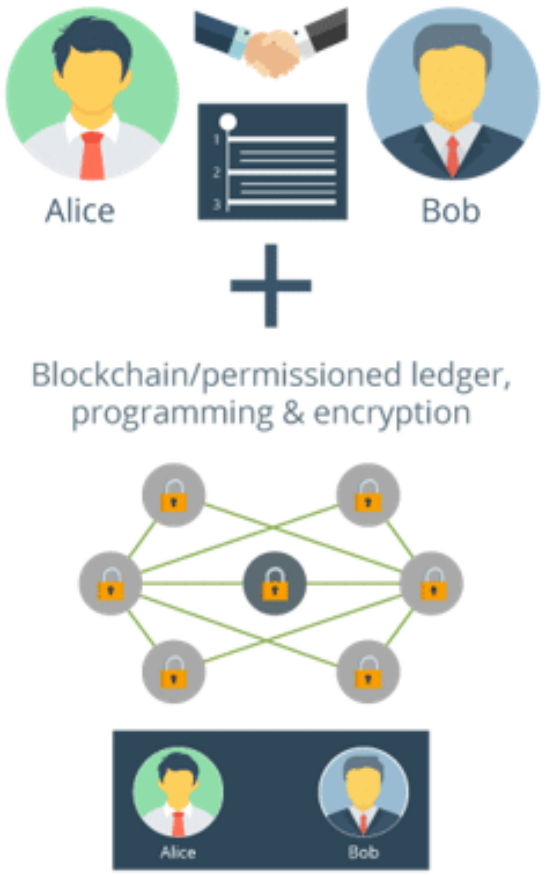
Smart Contracts

Traditional contracts	Smart contracts
1-3 Days	Minutes
Manual remittance	Automatic remittance
Escrow necessary	Escrow may not be necessary
Expensive	Fraction of the cost
Physical presence (wet signature)	Virtual presence (digital signature)
Lawyers necessary	Lawyers may not be necessary

source: edureka.com

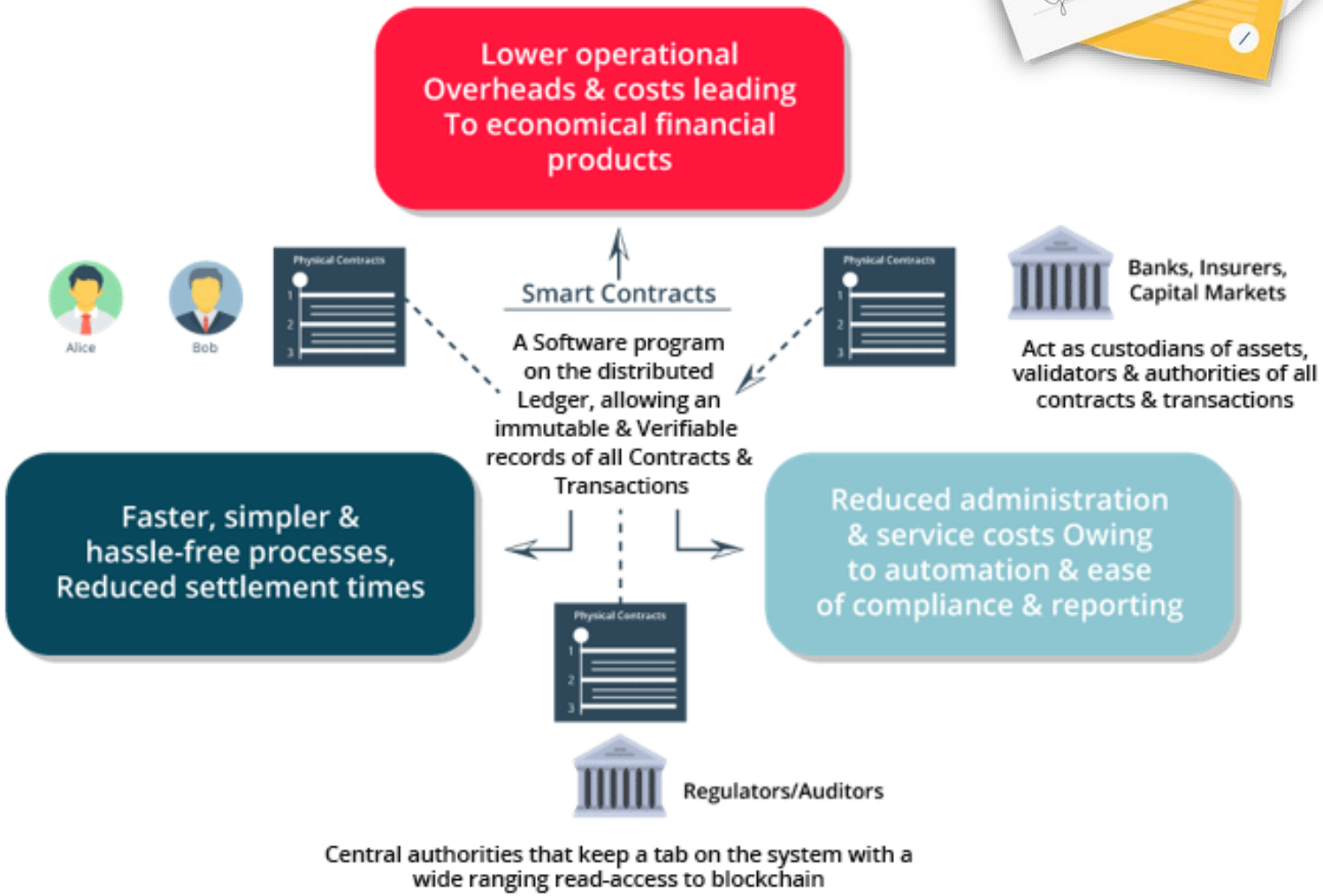
Smart Contracts

Physical Contracts



=

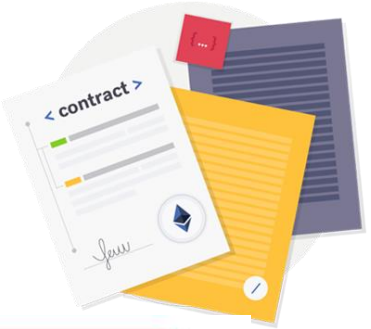
Smart Contracts



source: edureka.com

Smart Contracts

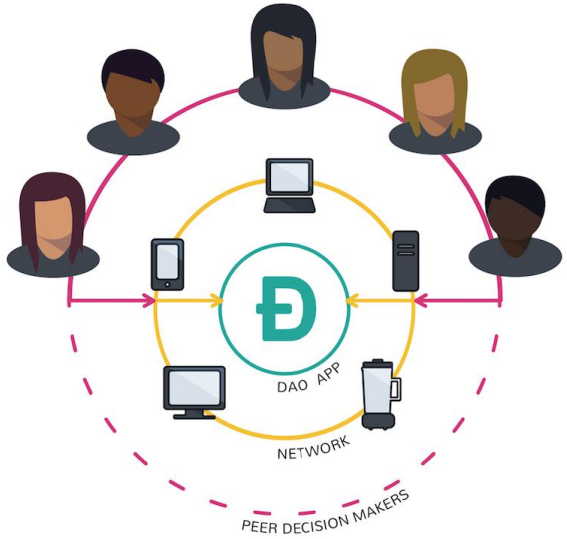
[FROM SIMPLE TO COMPLEX]



source: pwc.com

DAO - Decentralizes Autonomous Organizations

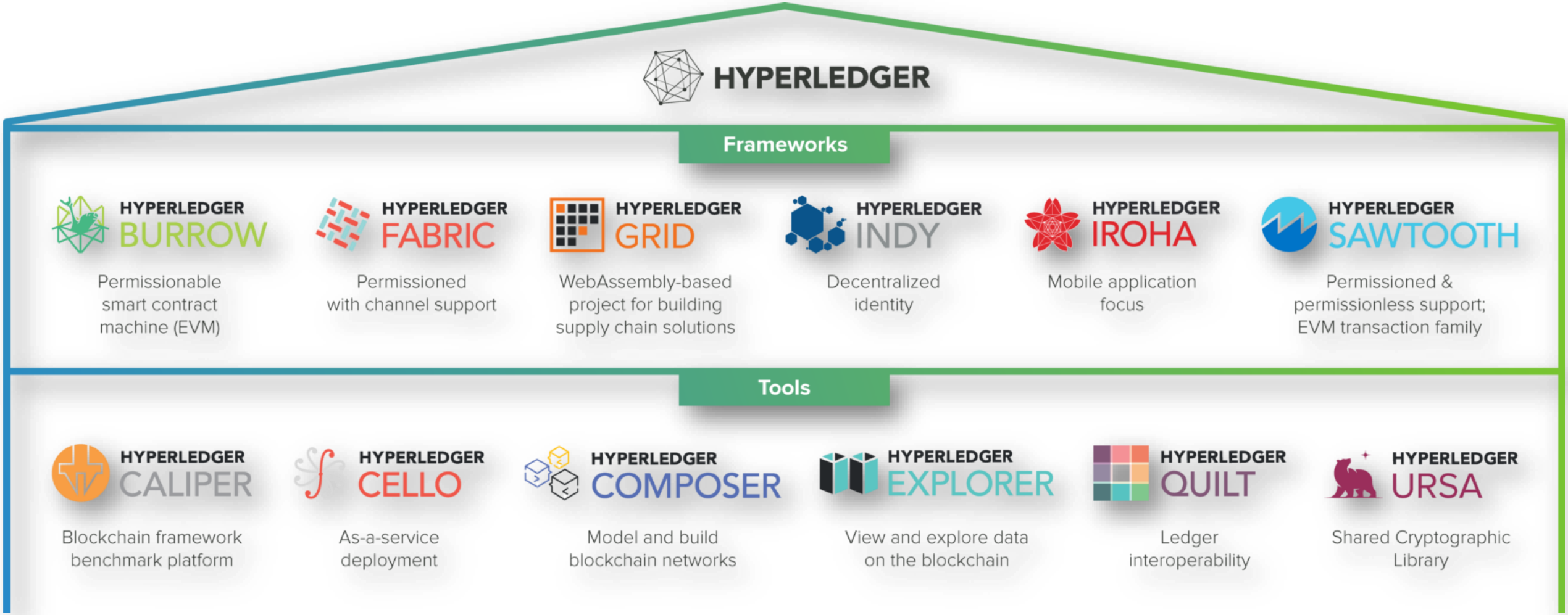
- Organizations that exists **autonomously** on a blockchain and are **governed by its protocol**;
- It is a corporation whose **bylaws are written entirely in code**;
- **No central control or authority**;
- Humans are only playing a part of a bug/code fixes, otherwise they are just participants;



HYPERLEDGER

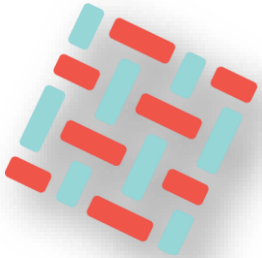
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

THE LINUX FOUNDATION


HYPERLEDGER
Frameworks
HYPERLEDGER BURROW
Permissionable smart contract machine (EVM)
HYPERLEDGER FABRIC
Permissioned with channel support
HYPERLEDGER GRID
WebAssembly-based project for building supply chain solutions
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

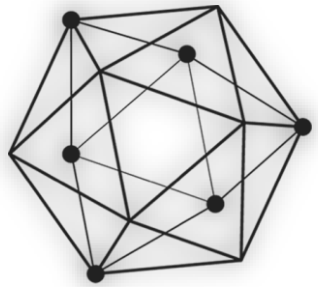
The blockchain for Enterprises/Business



HYPERLEDGER
FABRIC

HYPERLEDGER
SAWTOOTH

HYPERLEDGER PROJECT



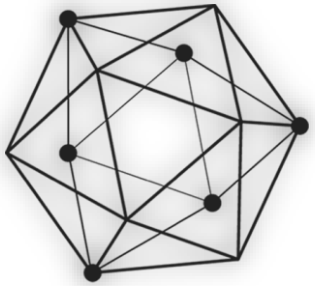
The Hyperledger model is designed to allow organizations to build and run industry-specific blockchain applications, platforms, and hardware systems that support their own business transactions.

Solutions lean more towards a enterprise models of blockchains.

Who are your business partners?

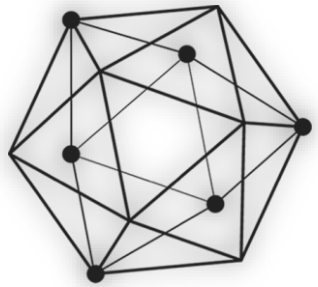
Can you agree and work together with others towards a common goal.

Who-What is HYPERLEDGER PROJECT?



- The Hyperledger project is **led by Linux foundation**, the trade association that oversees the development of open-sourced Linux operating systems;
- Combining previous work done by: IBM Open Blockchain, Digital Asset Holding, and Blockstream's libconsensus.
- The goal is to **advance blockchain technology's use in business** by developing both a **cross-industry open standard** and an **open-sourced development library** that would allow business to build custom-distributed solutions;
- It has **no native currency**;

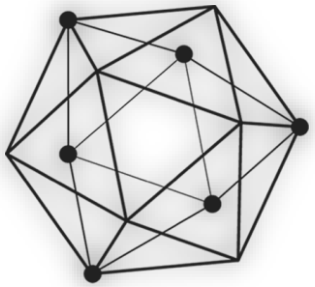
So, what is HYPERLEDGER PROJECT?



- Hyperledger in an **umbrella for multiple Blockchain related projects** (Incubator approach);
- Hyperledger has **members around the world**, with focus on interoperability of components and bringing blockchain community under one arm;
- **Open source** – duplication of work is avoided as tech community collaborates;
- Consistent **standards and best practices**;
- **Incentives are trust and security** instead of cryptocurrencies or digital tokens;
- Can be used across **all sectors and industries**;
- **Think of it as a operating system for marketplaces, data sharing networks, micro-currencies, and decentralized digital communities.**

source: transaformationworx.com

HYPERLEDGER PROJECTS

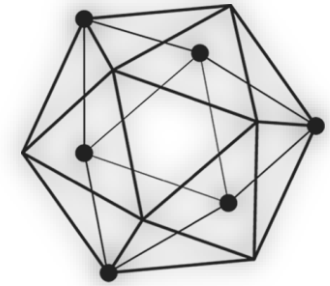


SAWTOOTH	Modular platform
IROHA	Infrastructure framework (i.e. IoT)
FABRIC	Application development framework
GRID	Supply Chain Solutions
BURROW	Permission-able smart contract machine
INDY	Decentralized Identity
CELLO	Blockchain as a service toolkit (BaaS)
COMPOSER	Collaboration tools, IDE
EXPLORER	Interactive web application service
QUILT	Interoperability between ledger systems (payments)

HYPERLEDGER FABRIC

HYPERLEDGER SAWTOOTH

source: transaformationworx.com



Infrastructure

Technical, Legal, Marketing, Organizational

Ecosystems that accelerate open development and commercial adoption

THE LINUX FOUNDATION

CloudFoundry | Node.js | **Hyperledger** | Open Container Initiative

Frameworks

Meaningfully differentiated approaches to business blockchain frameworks developed by a growing community of communities

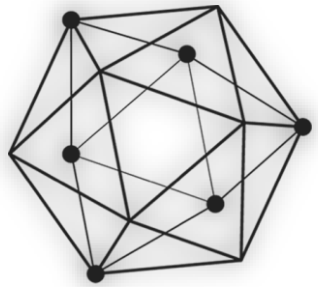
Hyperledger **Indy** | Hyperledger **Fabric** | Hyperledger **Iroha** | Hyperledger **Sawtooth** | Hyperledger **Burrow**

Modules

Typically built for one framework, and through common license and community of communities approach, ported to other frameworks

Hyperledger **Composer** | Hyperledger **Explorer** | Hyperledger **Cello**

HYPERLEDGER PROJECTS



	BITCOIN	ETHEREUM	HYPERLEDGER FRAMEWORKS
Cryptocurrency based	YES	YES	NO (but can be implemented)
Permissioned	NO	NO	YES (in general)*
Auditable	YES	YES	YES
Immutable ledger	YES	YES	YES
Modularity	NO	NO	YES
Smart Contracts	NO	YES	YES**
Consensus Protocol	PoW (SHA256)	PoW(Ethash)	Various*** (PBFT)
Language	C++	Golang, Python	Golang, Java

* Sawtooh – can be programed to be permission-less
** In Hyperledger referred to as Chaincode
** Key Hyperledger consensus protocols are **PoET in Sawtooth**, RBFT in Indy, Tendermint in Burrow; **Fabric** is built on pluggable architecture such that developers can configure with the consensus module that best suits their needs. Initial release offers three consensus mechanisms: 'no-op' – consensus ignored; '**classic PBFT**', and 'SIEVE' – enhanced version of PBFT.

source: medium.com

Hyperledger Fabric Capabilities

IDENTITY MANAGEMENT	Enables creation of permissioned networks by authenticating every network participant. Enables access control lists to provide granular role based access
PRIVACY	All data is encrypted and signed to provide need based access that allows enterprise application states
EFFICIENT PROCESSING	Enables concurrency and parallelism by separating transaction execution from transaction ordering thereby reducing the overhead and increasing throughout
CHAINCODE FUNCTIONALITY	Enables logic to define asset ownership and maintain distinct transaction validation
MODULAR DESIGN	Plug and Play modular design enables to adapt the system to any possible use case by incorporating only those components that are needed for the use case

source: transaformationworx.com

COMPONENTS of Hyperledger Framework

A business blockchain solution used to build for enterprise blockchain or for consortium of organizations.

- **Append-only** distributed ledger.
- A **consensus algorithm** for agreeing to changes in the ledger.
- **Privacy** of transactions through permissioned access.
- **Smart Contracts** to process transaction requests.

CHAINCODE

[smart contract]

- **Chaincode is a programme** that is written to read and update the ledger state. All the business logic is inside the chaincode.
- Written in Go or Node.JS
- Enables users to **create transactions** in the network's shared ledger and **updated the state of the assets**.
- Developers use chaincode to: **develop business contracts**, **asset definition**, and collectively **mange decentralized applications**.
- It **manages the ledger state** through transactions invoked by applications.
- **Applications interact with the blockchain via chaincode.**

CHAINCODE

[smart contract]

Quick high level overview

- Peer can be part of one or more channel.
- Every channel has a separate ledger.
- Every Channel has one or more chain codes.
- Every Chain code has a different endorsement policy.
- Chaincode must be part of a channel. As the ledger is part of a channel. One channel can have as many chain-codes as possible.
- Chaincode must be installed in each peer that is part of the channel and instantiated.
- When a Chaincode gets instantiated a policy (endorsing) has to be defined. [consensus: before a transaction can be recorded in the ledger only if a rule is met]

PRIVATE CHANNELS

Fabric's distributed ledger and smart contract platform allow for private channels. If you have a large blockchain network and you want to **share data only with certain parties**, you can create a private channel with just those participants.

Moreover, **not every transaction can be seen by every user of the network. Fabric allows private transactions — something that isn't possible in Ethereum**, which fosters transparency. For certain highly regulated industries, such as healthcare, this is most definitely a pro.

Use Case - TUNA

- With the Tuna 2020 Traceability Declaration (WEF) in mind, our goal is to eliminate illegal, unreported, and unregulated fishing. We will use Hyperledger Fabric to bring transparency and clarity to a real-world example: the supply chain of tuna fishing.

- Two things to keep in mind:
  1. There are **many actors within the network**, and you will see how these actors **interact**, and how a transaction is conducted.
  2. **Private channels** allow Sarah and Miriam to privately agree on the terms of their interaction, while still **maintaining transparency**, so other actors can corroborate and confirm their transaction.

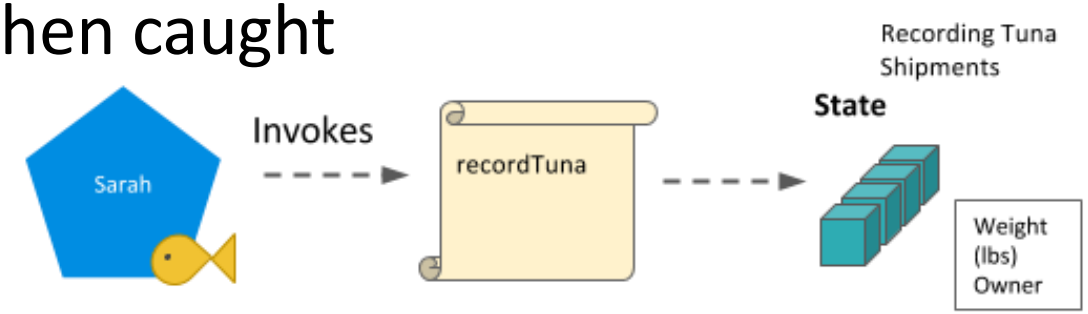
- Using private channels, regulators and restaurateurs can confirm whether a particular shipment of tuna was sustainably and legally sourced, without needing to see the details of the entire journey.

Use Case - TUNA

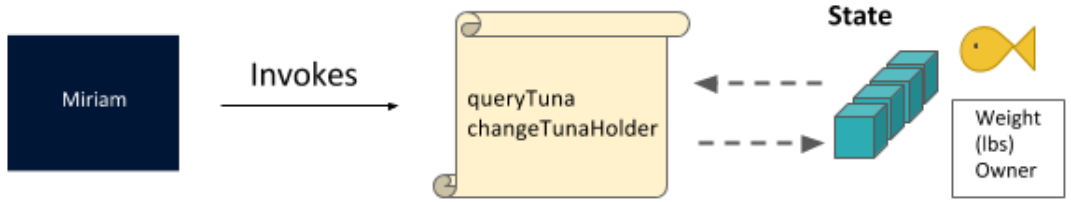
- Stakeholders
  - Sarah - is the fisherman who sustainably and legally catches tuna.
  - Regulators – verify that the tuna has been legally and sustainably caught.
  - Miriam – restaurant owner that serves to the end users, who buys Sarah's tuna.
  - Carl – another restaurant owner, who Sarah sells tuna as well.

Use Case - TUNA

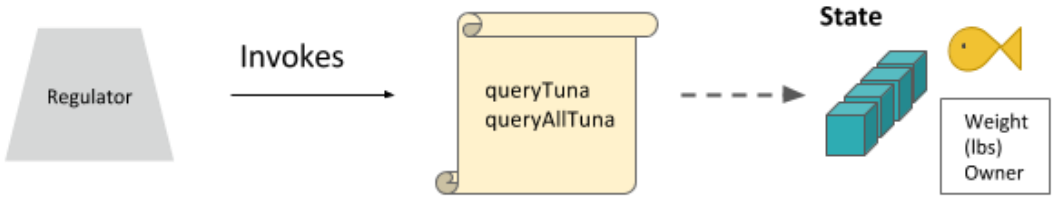
- Sarah fisherman will record the tuna when caught



- Miriam can confirm when she received the tuna, and can also view the state of a particular tuna.

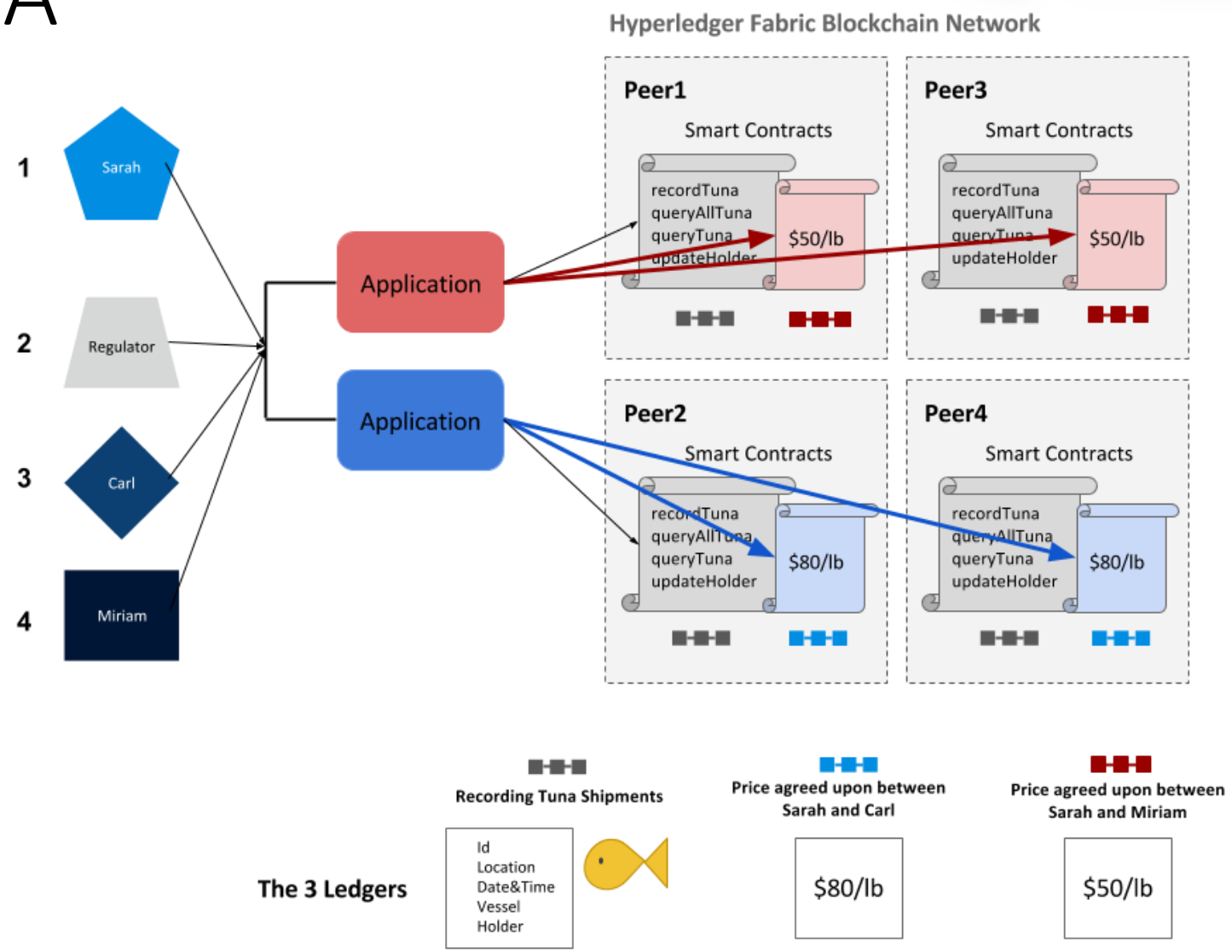


- Regulators can verify and check for sustainability of the supply chain.



Use Case - TUNA

- The BIG picture
- Tuna attributes that will be recorded on the ledger:
  - Vessel;
  - Location;
  - Date and Time;
  - Holder;



Hyperledger Fabric - ELEMENTS

1. **Channels** are data partitioning mechanisms that allow transaction visibility for stakeholders only. Each channel is an independent chain of transaction blocks containing only transactions for that particular channel.
2. **Chaincode /(Smart Contract)** | It encapsulates both the asset definitions and the business logic (or transactions) for modifying those assets. Transaction invocations result in changes to the ledger.
3. **Ledger** contains the current world state of the network and a chain of transaction invocations. A shared, permissioned ledger is an append-only system of records and serves as a single source of truth.
4. **Network** is the collection of data processing peers that form a blockchain network. The network is responsible for maintaining a consistently replicated ledger.
5. **Ordering Service** is a collection of nodes that orders transactions into a block.
6. **World State** reflects the current data about all the assets in the network. This data is stored in a database for efficient access. Current supported databases are LevelDB and CouchDB.
7. **Membership Service Provider** (MSP) manages identity and permissioned access for clients and peers.

Advantages

1. **Modular Architecture** – developers can create plug-in components (i.e. easy to include custom identity management).
2. **Build Permissioned Blockchain** – there are no anonymous or pseudo-anonymous users; identity is known; 'Membership Service Provider' (MSP) is for issuing and validating certifications and for user identification. You can assign different roles to nodes: 'client' – invokes transactions, 'order' – updates transaction data, 'peer' – receives updates from order and commits orders to the ledger, 'endorser' – a kind of peer that validates transaction authentically.
3. **Performance and scalability** – there is no PoW and crypto mining, delivers fast transactions and scalability;
4. **'Channels' for portioning data** – Fabric provides a data-portioning capability that achieves physical separation of sensitive data. (i.e. Banks that need to keep data for 10 years, and each year they need to upgrade the system to keep up with the computational power of hackers).
5. **Hardware-based protection of digital keys** – 'Hardware Security Model' (HMS) help in managing and safeguarding digital keys used for authentication, such as identity management;
6. **Rich community support** – some of the members are Oracle, Deloitte, Huawei, JP Morgan, etc.

Disadvantages

- An inadequate number of **skilled programmers** able to use it.
- Lack of proven use cases.

Considering that the Fabric was released in July 2017, the above disadvantages are expected. The project team is aware and challenges take time.

HYPERLEDGER FABRIC

"If you have a large blockchain network and you want to **share data with only certain parties**, you can **create a private channel with just those participants**. It is the most distinctive thing about Fabric right now."

- Brian Behlendorf, Executive Director of Hyperledger, The Linux Foundation

Other material

- Blockchain Expert Explains One Concept in 5 Levels of Difficulty, 18 min [LINK] – which level do you think you are at?

- IBM and Maersk Demo Cross Border Supply Chain Solution, 4 min [LINK]
- Hyperledger Fabric Explainer, 3 min [LINK]
- IBM Blockchain Car Lease Demo, 3 min [LINK]
- Jerry Cuomo answers the top questions about Hyperledger Fabric and v2.0, 13 min [LINK]

Hyperledger SAWTOOTH

- **Open-source** blockchain project under the enterprise blockchain platform Hyperledger.
- Perfect solution for developing networks and **distributed ledger applications, (dApps).**
- **Application domain is separated from the core system**.
- You can use **any programing language to develop the application**. What does this mean? When you are on the application layer developing dApps, it will take up resources but others using the core system won't be affected by it. Thus, **no slowing down or low-efficiency issues**.
- Mostly **suited for consortia** or similar environments. It is a private blockchain but you **can also gear it for a public environment**. For internal networking and for customer interaction as well.

source: 101blockchains.com

Hyperledger SAWTOOTH

- Having a public setup you **can pre-select the nodes that can enter the platform.**
- One of the best features of Sawtooth is the **Ethereum Contracts compatibility using Seth**. The Seth or Sawtooth-Ethereum Integration Project will create a **link between the two platforms**. Thus, you can deploy a smart contract within Hyperledger Sawtooth environment.
- **Pluggable consensus methods**, it will allow you to **run multiple consensuses at the same time in the platform**. At the time of setting up the network, you can easily select the one you wish to use. Moreover**, you can change it later and even while transacting;** this kind of process is called **Dynamic consensus**.

source: 101blockchains.com

Sawtooth Example: How It Works?

- Firstly the fisherman catches the seafood and then tags them with IoT sensors.
- Next, all the sensors start transmitting information back to the network with the location of the seafood.
- It goes through many distribution channels and the platform track each one of the sites.
- Before the buyer buys the seafood, he/she can use the platform to check the fish's provenance.
- Use Case – Restaurant [LINK]

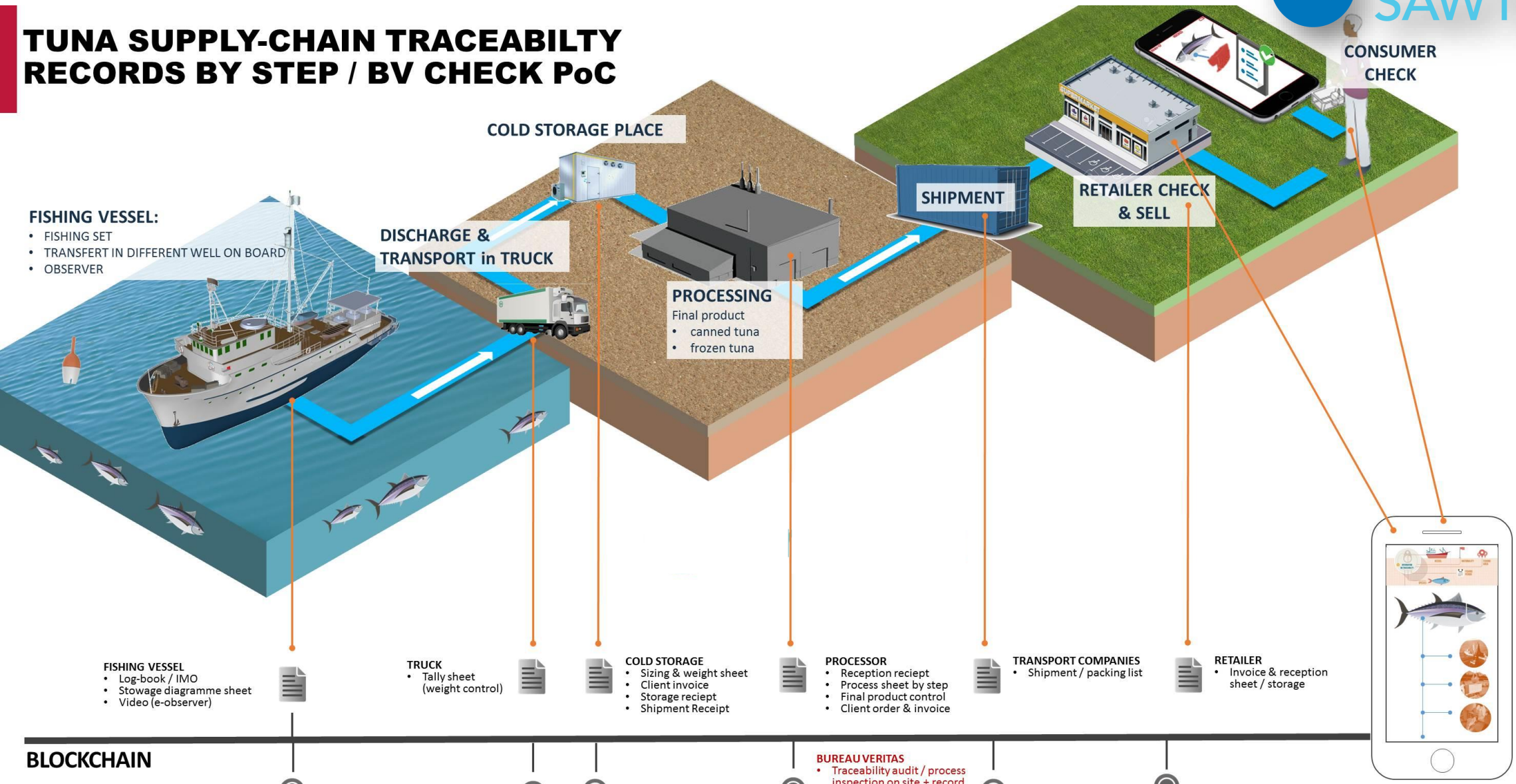
source: 101blockchains.com

Advantage

- **Extreme transparency** runs throughout the **whole process of supply chain.**
- Automated process **saves a lot of costs and time.**
- **Gets rid of man-made errors** and creates a **streamline of correct information.**
- Connects with better suppliers and **promotes consumer trust**.

TUNA SUPPLY-CHAIN TRACEABILTY RECORDS BY STEP / BV CHECK PoC

HYPERLEDGER SAWTOOTH



BLOCKCHAIN

BUREAU VERITAS
- Inspection during fishing trip
- Records « mass balance check
- Video check

BUREAU VERITAS
Unloading inspection

BUREAU VERITAS
- Cold storage inspection & audit

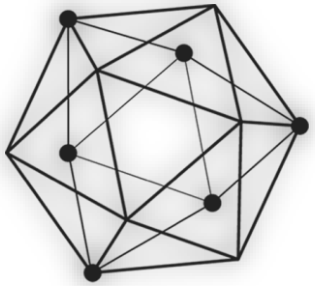
BUREAU VERITAS
- Traceability audit / process inspection on site + record check
- Lab testing (DNA) on final product

BUREAU VERITAS
- FRI (Final Random inspection)
- PSI (pre-shipment inspection)

BUREAU VERITAS
- FRI (Final Random inspection) quality + weight & tracability

Significant Differences

[some of many]



- **Hyperledger Fabric** is restricted in terms of participation, whereas Sawtooth is flexible as it is a blockchain network which is open to join. **Hyperledger Sawtooth** can be programmed to be permission-less.
- **Hyperledger Fabric** has a better and tighter governance framework as it uses MSPs (Membership Service Provider) and CAs (Certificate Authority).
- **Hyperledger Fabric** offers the unique concept of channels. It provides fine-grained privacy control over transactions. It supports B2B scenarios and many others where it can be applied.
- Due to its **Seth** framework, **Hyperledger Sawtooth** supports **Ethereum** solidity-based smart contracts. **Hyperledger Fabric** does not support this.
- In **Fabric**, you'll see that all the processing of the transaction is done using the endorsing peers and ordering services. In reality, these two types of nodes are in charge of processing all the transactions. But in **Hyperledger Sawtooth** the issue changes drastically. Here, there's no ordering service or endorsing peers but rather validators that look after the process.

HYPERLEDGER FABRIC

HYPERLEDGER SAWTOOTH

source: blockchaincouncil.com

Significant Differences

[some of many]

	HYPERLEDGER FABRIC	HYPERLEDGER SAWTOOTH
PERMISSION LEVEL	Permissioned	Permissioned and Permissionless
CONSENSUS ALGORITHM	Mostly uses Kafka, supports other algorithms as well.	Mostly uses Proof of Elapsed Time, supports other algorithms as well.
BFT SUPPORT	No, only supports Crash Fault Tolerance	Yes, supports Byzantine Fault Tolerance
SMART CONTRACT TECHNOLOGY	Chaincode	Transaction Families
SMART CONTRACT TYPE	Installed	On-chain and Installed
SUPPORTED SMART CONTRACT LANGUAGE	Golang or JavaScript	Rust, JavaScript, Go, or Python; Supports Solidity using Seth
TRANSACTION PROCESSING	Endorsing Peers and Ordering Services	Validators
STATE STORAGE	CouchDB or leveldb	Central lmdb database
EVM SUPPORT	No	Yes (through Seth)

source: blockchaincouncil.com

HYPERLEDGER DESIGN ARCHITECTURE

EXTREMELY SECURE PLATFORM

Hyperledger ensures enhanced security measures in the design architecture of the network to resist any online attacks.

CRYPTOCURRENCY-AGNOSTIC

The company doesn't have any cryptocurrency of its own. However, you will be able to create and manage it as the design architecture permits it.

MODULAR DESIGN

The modular structure ensures the development of individual components without affecting the overall nature of the network.

INTEROPERABLE

Interoperability will ensure that every blockchain network can communicate with each other. Hyperledger believes this factor will streamline the globalization effectively.

HIGH END API SUPPORT

Hyperledger provides feature-rich APIs for other clients or applications to communicate with the core DLT infrastructure.

CORE HYPERLEDGER USE CASES

BANKING

- Offers a transformative identity solution.
- Provides integration of smart contracts for loans.
- Can provide security for the membership system.

FINANCIAL SERVICES

- Streamline post-trade processing.
- Offer confidentiality and privacy for sensitive data.
- No delay, no bottlenecks in transactions.

HEALTHCARE

- Streamlining credentialing physicians.
- Security to fight off counterfeit drugs.
- Offer insurance verification for patients.

INFORMATION TECHNOLOGY

- Offers trustless environment.
- Promotes self-sovereign identity.
- Offers industrial grade security for Information Technology.

SUPPLY CHAIN MANAGEMENT

- Provides global authority for tracing.
- Offers logistic based processes.
- Can quickly identify corrupted personal.

GOVERNMENT

- Security for citizen rights.
- Offers transparent election system.
- Provide digital identity for citizens.

AUTOMOTIVE

- Offer secure, seamless mobility services.
- Provide micropayment options.
- Better interaction with ride-sharing applications.

MEDIA AND ENTERTAINMENT

- Streamlining entertainment artists and channels.
- Offer better artist value.
- Security for intellectual properties.

Video

- Blockchain and Music industry [LINK]

Break & Quick Recap

Smart Contracts

- A smart contract is software code that **runs on a blockchain** network such as Ethereum and **performs actions** or tasks **based on certain events.**
- Such an example is sending Ether; If everything checks out, the network will ‘transfer’ the funds to the receiver.
- But what if I wanted to **create a decentralized application** that **needed external data** such as the current weather temperature, the price of a stock or even the results of the FIFA World Cup finals, NBA, Tennis, Olympics, etc.? How does a smart contract, or, in other words, a piece of code on a blockchain, get this information?

Oracles and Blockchain

So where does the blockchain get the ‘truth’ from the **outside world**?

Smart Contracts and Oracles

- Oracles are **trusted data sources** or entities that provide information or sign claims about the **state of the world** for smart contracts. They are the **link between real world events and the digital world** of blockchain platforms. They **don't make predictions** about the future but **report events from the past**.
- The **most powerful oracle architecture offers a decentralized network of oracles that connects directly to smart contracts**, and **feeds the data back in a secure manner**. With both **software and hardware security measures**, this ensures **security and tamper-proof data that can be trusted.**

Smart Contracts, Oracles and Trust

- The whole point of the blockchain and its decentralized network was to **remove the need to trust any intermediary and remove any single point of failure so obviously a smart contract that requires trusting a single outside data source is a bit of a contradiction but this can be mitigated by having multiple independent oracles to form a consensus.**

Types of Oracle

SOFTWARE	Handles data sourced online
HARDWARE	Replays offline data from physical world
INBOUND	Passes external data to smart contrast
OUTBOUND	Communicates smart contracts data to outside world
CONSENSUS OF ORACLES	Increased data validity confirmation from several oracles

Blockchain-Oracle projects

AUGUR

GNOSIS

Chain Link

æternity BLOCKCHAIN

Smart Contract – Characteristics

[smart contract are not a technical issue but a business one]

1. **Smart contracts can call other contracts**, if needed. This means that if there are **two parts** of a contract, wherein the **first part is concerned with checking a requirement**, then after the condition gets validated the **second part of the contract where the task might be to carry out a transaction**, will receive the output of the first part and the transaction will take place successfully.
2. **Smart contracts act as accounts** that need multiple signatures. Unless a required number of people sign, funds won't be transferred.
3. Smart contracts **help in managing the agreement between two or more users**, for example if one person sells an insurance to the other.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

Smart Contract - Characteristics

4. **You don’t need to rely on any intermediaries to carry out or confirm your agreements**. This **removes the risk of a third party manipulation**, as execution is done by the network in an **automated** way.

5. All the documents are encrypted in a shared ledger so nobody can claim to have lost something of yours.

6. Everything is backed up in the blockchain, as all the nodes have a copy of it so you can’t really worry about losing data.

7. The smart contracts are built on a blockchain environment so everything is well encrypted so it would take a really smart hacker to tamper with data.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

Smart Contract - Characteristics

8. Smart **contracts save your time** as all tasks are automated due to them written in software code. This also **removes the paperwork** and endless settlements and other legalities.

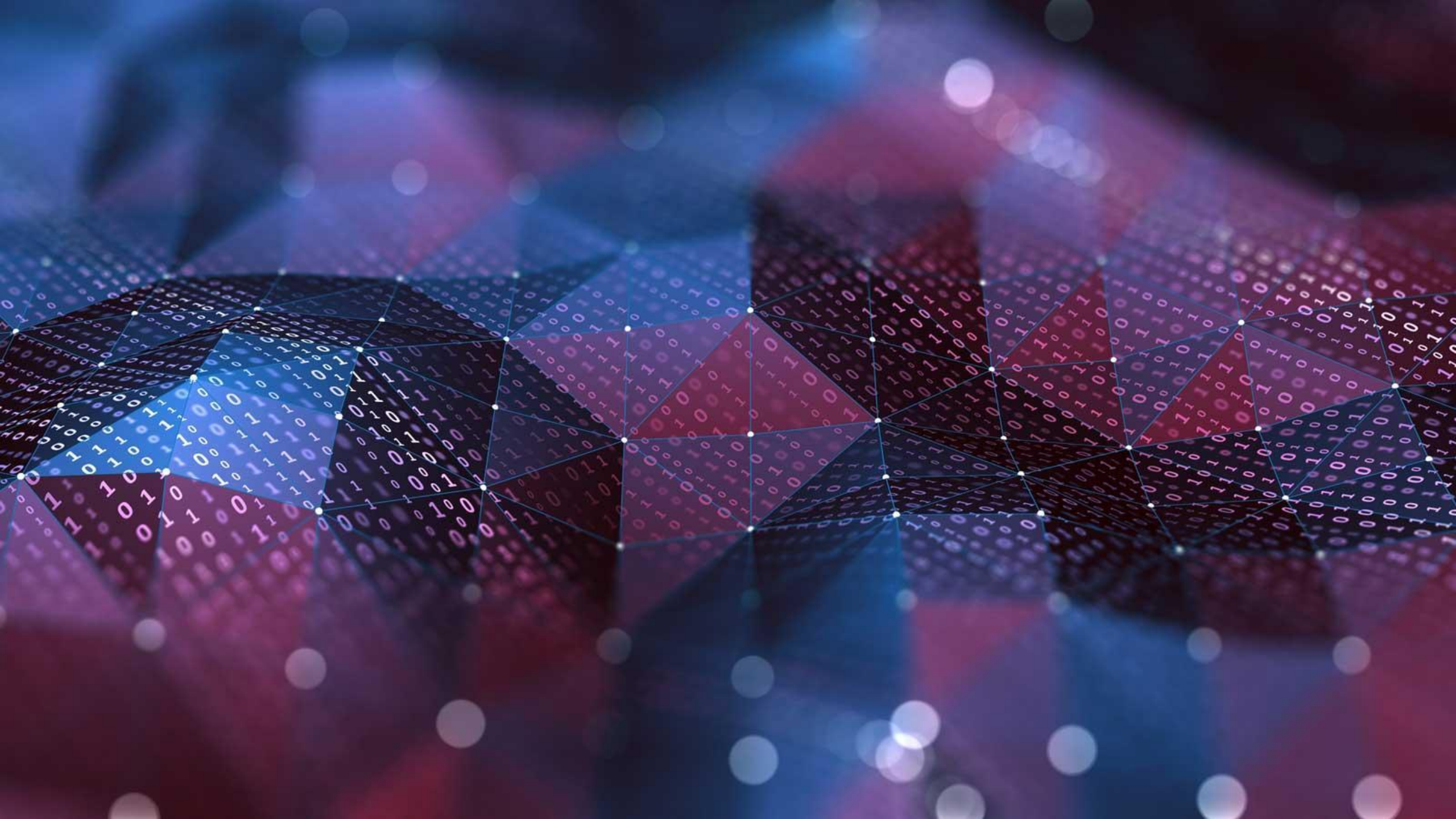
9. Smart contracts also **save you money** since **no middleman** is being paid here, thanks to the blockchain environment!

10. Smart contracts are pretty accurate as they are automated, which removes the possibility of human errors that may occur in filling out heaps of forms.

source: https://hackernoon.com/smart-contracts-a-time-saving-primer-b3060e3e5667

Future

The use of blockchain combined with smartphones, RFID, IoT networks, and mobility applications could create (by 2030) or make almost every if not all industries with near-perfect records.



USE CASE | Manufacturing Supply Chain

[quick demo i.e. laptop]

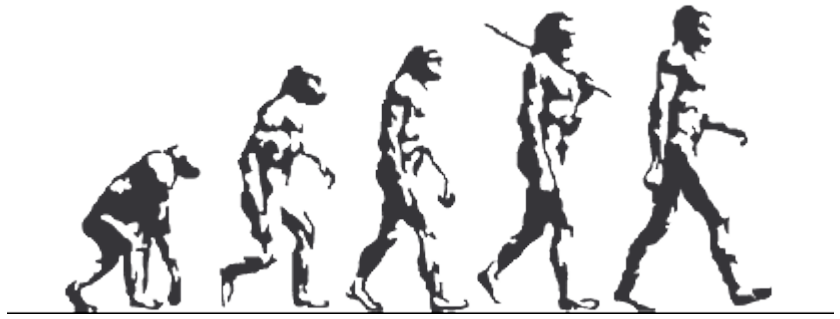
- A computer company decides to produce some amount of laptops for the following year. They will make an order to the laptop manufacturer.
- Manufacturer will contact its hardware suppliers (chips, monitors, HD, ram, video cards, etc.) in order to produce the final product.
- Once all parts are assembled together to produce the final product (laptop) they will be shipped via logistic companies to a distribution center.

source: transformationworx.com

USE CASE | Manufacturing Supply Chain

[quick demo i.e. laptop]

- Problem statement:
  - Where there is an issue (problem) with a laptop and its called back it is difficult to know were it originated and if it's a single laptop issue or the whole series of them. The investigations can last for months.
  - Inability or very expensive means to identify point of original source. What came from where?
  - Root cause opaque* in the system and individual components are not connected one with each other in a reliable manner. (digital does not mean reliable)



(Not able to see through, not transparent, unclear, uncertain)

source: transformationworx.com

USE CASE | Manufacturing Supply Chain

[quick demo i.e. laptop – current process]

Supplier 01

ABC | Manufacturer | Supplier 02 | Assembly | Logistics | Distribution

Supplier 03

Placing Master Order

order A →

← quote for A

PO for A →

Placing Sub-Orders

order A.1 →

order A.2 →

order A.3 →

← quote for A.1

← quote for A.2

← quote for A.3

*each supplier has access to their information and can not be seen by other suppliers

source: transformationworx.com

USE CASE | Manufacturing Supply Chain

[quick demo i.e. laptop]

	ABC	Manufacturer	Supplier 01	Supplier 02	Supplier 03	Assembly	Logistics	Distribution
Place Master Order	✓	✓						
Place sub-orders		✓	✓*	✓*	✓*			
Parts from suppliers								
Final Product								
Logistics – Location								
Logistics – Time								
Logistics - Status								

source: transformationworx.com

USE CASE | Manufacturing Supply Chain

[quick demo i.e. laptop]

	ABC Company	Manufacturer	Supplier 01	Supplier 02	Supplier 03	Assembly	Logistics	Distribution
Place Master Order	✓	✓						
Place sub-orders		✓	✓*	✓*	✓*			
Parts from suppliers		✓	✓*	✓*	✓*			
Final Product	✓	✓				✓		✓
Logistics – Location	✓	✓	✓	✓	✓	✓	✓	✓
Logistics – Time	✓	✓	✓	✓	✓	✓	✓	✓
Logistics - Status	✓	✓	✓	✓	✓	✓	✓	✓

source: transformationworx.com

USE CASE | On-chain vs Off-chain

[quick demo i.e. laptop]

Information seen only by the **SOME** players thus requires discussion if it has to be on-chain or off-chain

	ABC	Manufacturer	Supplier 01	Supplier 02	Supplier 03	Assembly	Logistics	Distribution
Place Master Order	✓	✓						
Place sub-orders		✓	✓*	✓*	✓*			
Parts from suppliers		✓	✓*	✓*	✓*			
Final Product	✓	✓				✓		✓
Logistics – Location	✓	✓	✓	✓	✓	✓	✓	✓
Logistics – Time	✓	✓	✓	✓	✓	✓	✓	✓
Logistics - Status	✓	✓	✓	✓	✓	✓	✓	✓

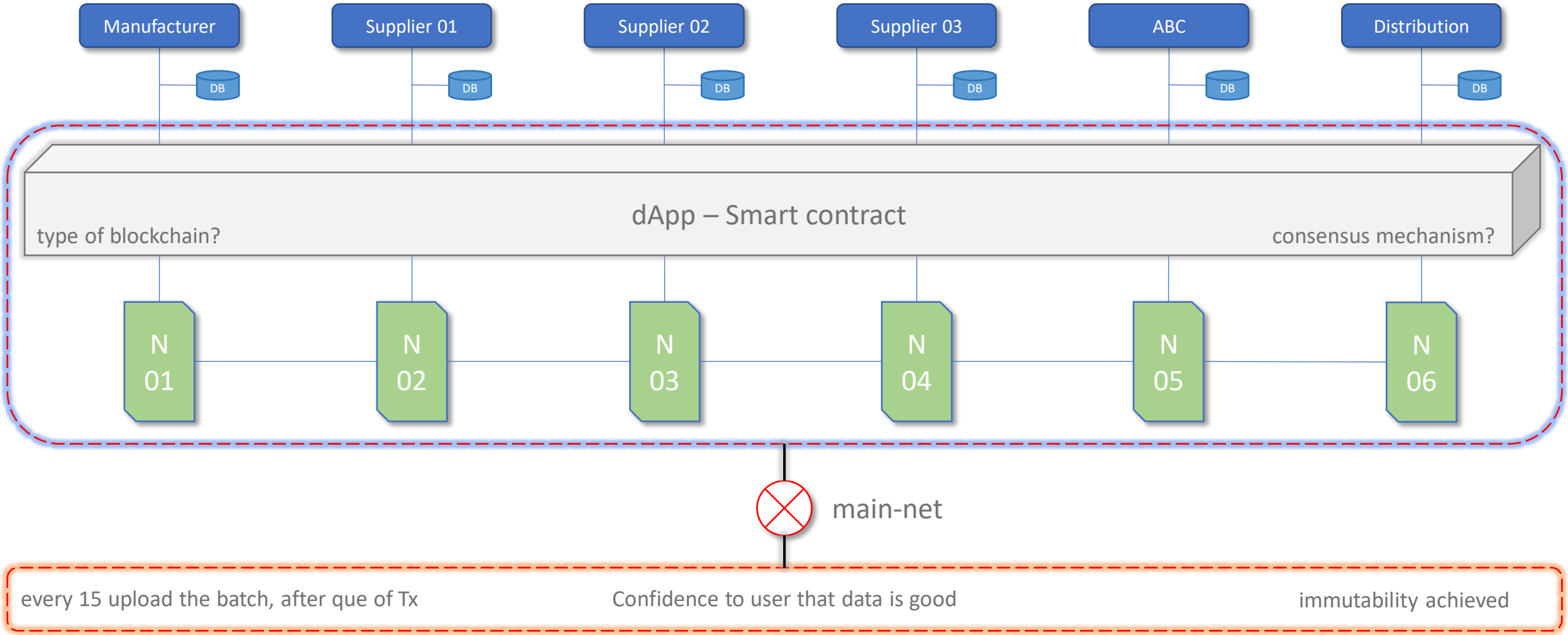
Information seen only by **ALL** players thus requires discussion if it has to be on-chain or off-chain

Manufacture is a contender that has access to everything on the ledger – contender for the blockchain – they are the signers or decision makers that make a decision if the information moves onto the blockchain or no.

source: transformationworx.com

USE CASE | Graphic Design Solution

[quick demo i.e. laptop]



every 15 upload the batch, after que of Tx

Confidence to user that data is good

immutability achieved

ETHEREUM

v2.0

https://www.guru99.com/ethereum-tutorial.html

So far what we know…

- Unlike bitcoin were we don’t know who is Satoshi Nakamoto, the big name that stands behind the creation of **Ethereum is Vitalik Buterin**.
- Vitalik profile – Russian origin who grow up in Toronto-Canada, excellent mathematician, went to Waterloo and studied Computer Science and dropped out from school to work on Ethereum (with his father and contributors around the world);
- Ethereum famous for the **idea of Smart Contracts** and the platform is being used **to create decentralized apps** for a broad range of services and industries;
- **Many issues** with the platform in the beginning such as the DAO hack;
- Updates have been made to the platform since the creation…

What is Ethereum

- ETHEREUM is an **open-source operating system** which offers **smart contract** functionality.
- It is a **distributed computing platform** that supports developing decentralized digital applications(**DApps**) using blockchain technology.
- Ethereum provides a **decentralized virtual machine called Ethereum Virtual Machine(EVM)** that can run the scripts with the **use of an international network of public nodes**.

Why do you need Ethereum?

- **Centralized systems** are one of the most widespread models for software applications.
- This system directly **controls the operation of the individual** units and the flow of information **from a single center**. In this kind of system, **individuals are depended on the central power** to send & receive information.

- However, there are **issues** with the centralized system are:
  - Single point of control & failure
  - It can be corrupt easily
  - Performance bottleneck
  - Silo effect

The solution is Decentralized applications

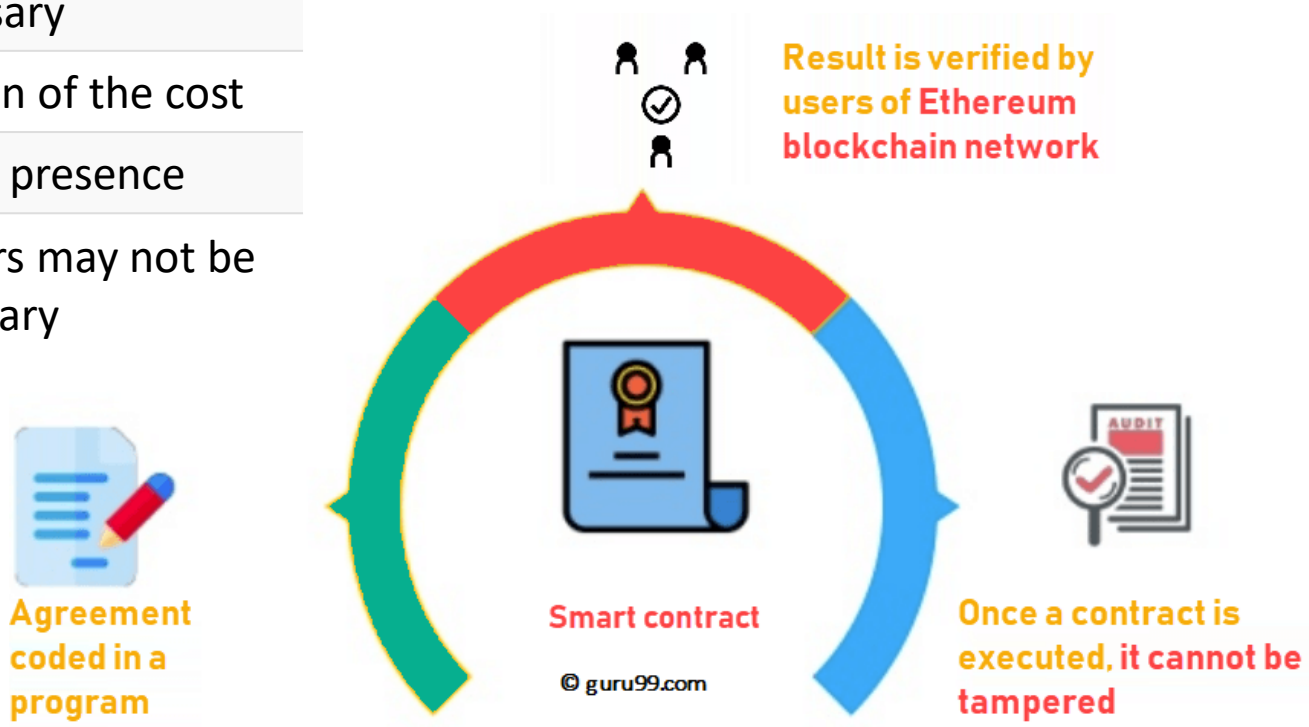
The term **DApp** is a combination of **two words**- **decentralized applications**. In simple words, it is an application, tools, or programs that work on the decentralized Ethereum Blockchain.

History of Ethereum

- **2013**: Vitalik Buterin, a developer who was involved in Bitcoins, and he was first to describe on paper
- **2014**: A Swiss firm Ethereum Switzerland Gmbh developed the first Ethereum software project
- **2015**: Frontier, the first version of Ethereum was launched.
- On March 14, **2016**: A planned protocol Homestead becomes second biggest version upgrade of the ethereum network.
- On May **2016**: Ethereum gets the most extensive media coverage when the DAO raised a record $150 million in crowd sale.
- On July **2016**: The network branched into two broad categories: Ethereum (ETH) and Ethereum Classic (ETC).
- June **2017**: Ethereum rallies above $400 recording a 5001% rise since Jan 1st, 2017
- May **2017-** Ethereum will eventually overtake the success of Bitcoins
- June **2018-** The DAO was hacked by an anonymous group claiming $50 worth of ETH.

What is Smart Contact?

Parameter	Traditional	Smart contracts
Duration	103 Days	Minutes
Remittance	Manual	Automatic
Escrow	Necessary	Necessary
Cost	Expensive	Fraction of the cost
Presence	Physical presence	Virtual presence
Lawyers	Lawyers are important	Lawyers may not be necessary



Gas on Ethereum

- To perform a transaction on the Ethereum network, a user requires to **make a payment (to the miner)** Ether via an intermediary token called **'Gas**.' It is a unit which allows you to **measures the computational work required for running a smart contract** or other transactions.

- In Ethereum, the **transactions fee is calculated in Ether**, which is given as

- Ether = Tx Fees= Gas Limit * Gas Price
- Where,
  - **Gas Limit**= Refers to the amount of gas that is used for the computation
  - **Gas Price**= The amount of Ether a user is required to pay

Ethereum vs. Bitcoin

Parameter	Bitcoin	Ethereum
Definition	Bitcoin is a digital money	Ethereum is a world computer.
Founder	Satoshi Nakamoto	Vitalik Butarrn
Hashing algorithms	Bitcoin used SHA-256 algorithm.	Ethereum uses Etash algorithm.
Average Block time	10 minutes	10-15 sec
Release Date	9 Jan 2008	30 July 2015
Release Method	Genesis Block Mind	Prasala
Blockchain	Proof of work	Proof of work (Planning for POS)
Usage	Digital Currency	Smart Contracts Digital Currency
Cryptocurrency Used	Bitcoin(Satoshi)	Ether
Blocks Time	10 Minutes	12-14 Seconds
Mining	ASIC miners	GPUs
Scalable	Not now	Yes
Concept	Digital money	World Computer
Cryptocurrency Token	BTC	Ether
Turing	Turing incomplete	Turing complete
Coin Release Method	Early mining	Through ICO
Protocol	Bitcoin still employs the pool mining concept.	It uses a Ghost Protocol.

Advantages of Ethereum

- Allows you to upload and request programs to be executed
- 100% uptime and DDOS resistant
- Ethereum helps you to create a tradable token that you can use as a new currency or virtual share.
- Persistent and permanent data storage
- Build virtual organizations
- Helps you to develop decentralized applications
- Ethereum helps you to build fault-tolerant and highly secure decentralized apps.

Disadvantages of Ethereum

- The Ethereum Virtual Machine is slow, so you can't use it for large computations
- Storage on the blockchain is expensive.
- Swarm Scalability is an issue, so there is a trade-off with decentralization Private block chains are likely to proliferate.
- Fixing bugs or updating Apps is a tough task because every peer in the network need to update their node software.
- Some applications require verification of user identity, and as there is no central authority to verify the user identity.

So what is Ethereum 2.0 - **Serenity**?

- Ethereum 2.0 marks a **long-anticipated upgrade** to the Ethereum public mainnet.
- Ethereum 2.0 is not a new idea in the Ethereum community. A shift in Ethereum's underlying consensus mechanism to address the restrictions of a Proof of Work blockchain has existed since the blockchain's genesis.
- The launch of Ethereum 2.0 is especially significant compared to past upgrades because of the **implementation of a Proof of Stake** consensus mechanism, moving the network away from its existing Proof of Work architecture.

In brief

- Ethereum, the **second-largest cryptocurrency** by market cap, is planning a **major update to its network**.
- The Ethereum 2.0 upgrade **aims to address the network's scalability and security.**
- The first phase of the upgrade, due in 2020, will see Ethereum switch to a proof of stake consensus mechanism.

Previous Upgrades

- **Homestead**, March 2016
- Metropolis: **Byzantium**, October 2017
- Metropolis: **Constantinople**, February 2019
- **Istanbul**, December 2019

Proof of Stake

- **Proof of Stake (PoS)** is the most significant change in Ethereum 2.0, because it **reforms the crypto-economic incentive structure for validating the blockchain**.
- In Proof of Work, miners run nodes and expend computational energy to solve complex mathematical problems in a competition to mine the next block.
- The time and money that miners need to run hardware and expend electricity on PoW chains is validated by block rewards, which are distributed to miners who successfully mine a block into existence. **PoW chains are extremely secure**; the combined computational power required for an individual to compromise a well-established PoW blockchain like Bitcoin or Ethereum would cost an extraordinary amount of money, and **may not even exist**.
- **Although extremely secure, PoW blockchains suffer from scalability and accessibility issues**.

Proof of Stake

- **Proof of Stake replaces the two primary components of PoW (miners & electricity) with validators and stake on Ethereum 2.0.** Broadly speaking, **validators replace miners as the individuals who maintain the agreed-upon state of the network and receive rewards** for randomly selecting the next block of data. Unlike in **PoW, in which miners expend physical energy (called hash power) by burning electricity to confirm blocks, validators in a PoS system commit 32 ETH as 'skin in the game.'**

- On Ethereum 2.0, validators will **stake at least 32 ETH by depositing** the funds into the official deposit contract that has been developed by the Ethereum Foundation. Validators will download and run Ethereum 2.0 client software. While running client software, they will be randomly selected to propose and attest to blocks on the Ethereum 2.0 blockchain. **Validators who correctly propose and attest to blocks will receive a reward of ETH as a percentage of their stake**.

Proof of Stake

- The **crypto-economic incentives against malicious actors are stronger on a PoS network**. Whereas in PoW, the economic risk of a failed attack is equal to the cost of electricity to achieve the required hash power, in PoS the cost to launch an attack is equal to the amount of staked ETH, and therefore so is the penalty.
- The chances are higher that an individual is warier of directly risking money in the form of staked crypto to launch an attack compared to the more abstract cost of electricity.
- **Whereas a failed PoW attack results in the loss of electricity costs, slashing a validator's stake is the PoS equivalent of a miner burning down an entire PoW server farm in a failed attack**. In short, the economic incentive to act in the **network's best interest is stronger**.

Conclusion

- Ethereum 2.0 has been long-anticipated and much discussed in the blockchain ecosystem. Proof of Stake and **sharding** will bring considerable **improvements to scalability, security, and accessibility**.

- Ethereum 2 will also introduce a **new structure**: **rather than being one chain it will consist of a beacon chain that communicates with several sharded blockchains ("shards").** This allows transactions to be carried out in parallel, increasing the transaction throughput and capacity of the network.

- Ethereum 2 will be **rolled out in several phases**. Phase 0, the launch of the beacon chain, is expected in late 2020.

Ethereum 2.0 (“Serenity”) is an upgrade to the Ethereum Network which improves the speed, efficiency, and scalability of the network

Further Info - How many **phases** are expected in Ethereum 2?

- How many phases are expected in Ethereum 2? Ethereum 2 will be rolled out in three main phases:

  - **Phase 0**: the beacon chain. Launched on 1 December 2020. This is the core chain that makes sure the shards across the network are kept in sync. It achieves this using validators.

  - **Phase 1**: the shard chains. Expected 2021. Ethereum's data and transactions will be run on shard chains. These are proof-of-stake blockchains that use validators to confirm transactions, create blocks and communicate with the beacon chain. It is expected that Ethereum 2 will begin with 64 shards. At this point, they will not be supporting smart contracts or accounts.

  - **Phase 1.5**: the legacy Ethereum chain becomes a shard on the new Ethereum blockchain. Expected 2021. In this phase, the legacy Ethereum network will transition to a proof-of-stake network and be connected to the main Ethereum 2 chain.

  - **Phase 2**: shards fully functioning. Expected 2021+. The shard chains will be fully functioning and able to communicate with each other and run smart contracts.

Why is Ethereum being **upgraded** to Ethereum 2?

- The **legacy Ethereum network uses a proof-of-work** block validation process, similar to Bitcoin. Ethereum 2 proposes a **number of architectural changes**, whose aim is to introduce a number of **improvements to Ethereum** – helping **reduce the energy** needed to run the chain, **increase scalability** and **transaction throughput** while also **increasing the network's attack resistance**.
- Ethereum 2.0 will involve sharding to **drastically increase network bandwidth and reduce gas costs, making it cheaper to send Ethereum, tokens, and interact with smart contracts**.

What are “**validators**” on Ethereum 2?

- With a proof-of-stake block validation process, Ethereum 2 will use **validators** to secure the network.

- A validator on Ethereum 2 **runs computer software that confirms transactions on the shards, adds them to the next block in the chain and communicates with the beacon chain**. A minimum of 32 ETH is required to be a validator, along with some technical skills.

- Validators are **rewarded for validating blocks and securing the network** and are **fined (“slashed”) if they behave badly or maliciously**. This economic incentive helps ensure “**good behaviour**” and keeps the network operating in a well-coordinated manner.

- What does “**beacon chain**” mean in Ethereum 2?
  - Rather than just being one blockchain, Ethereum 2 will be **made up of several chains: a core beacon chain connected to several shard chains (“shards”).** The beacon chain in Ethereum 2 will be launched in Phase 0.

What is “**staking**” on Ethereum 2?

- Staking on Ethereum 2 is the act of participating in the validation of blocks on the network by committing funds – staking them – and serving as a validator.

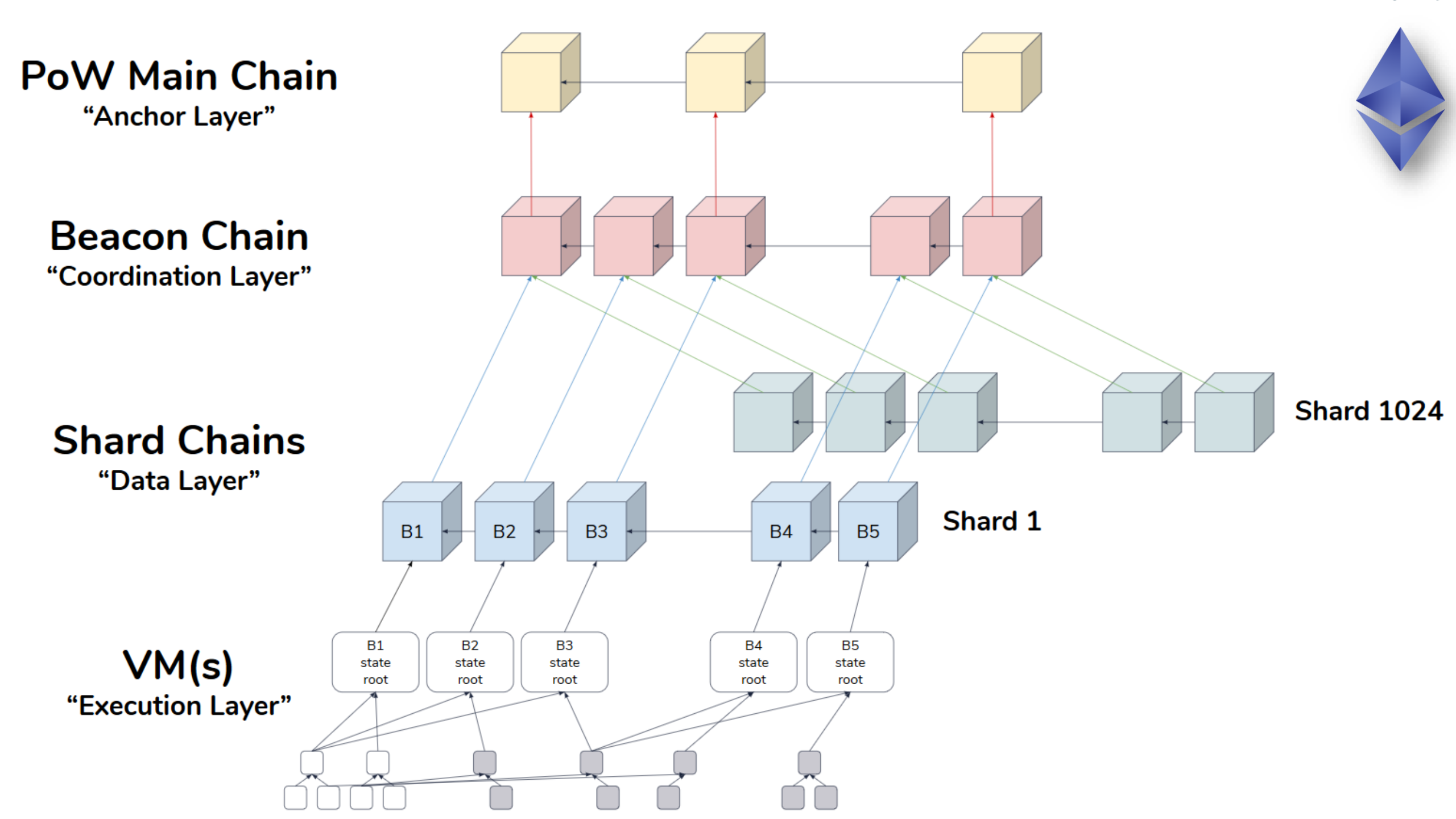
- There are **two ways to stake**: become a **validator**, or join a **staking pool**.

- **Validators**: A validator on the Ethereum network runs computer software that confirms transactions on the network, adds them to the next block in the chain and communicates between the shard and beacon chain. A minimum stake of 32 ETH is required to be a validator, along with software and some technical skills.
- **Staking pools**: If you have less than 32 ETH or you do not wish to run a validator yourself, you can participate through a staking pool or staking service.

What is a “**shard**” in Ethereum 2?

- The current Ethereum set-up has a blockchain consisting of a single chain with consecutive blocks. This is secure, but very slow and not efficient. With the introduction of shard chains, this blockchain is split up, enabling transactions to be handled in parallel chains instead of consecutive ones. This speeds up the network, and can scale more easily.

- Shards, sharding, or shard chains refer to parallel blockchains existing inside the new Ethereum 2 structure. While previously Ethereum existed as one blockchain, the new structure consists of a core beacon chain connected to different shard chains (expected to be 64 initially). This allows parallel processing of transactions across different shards. The transactions in each shard will be verified by validators and confirmation passed to the beacon chain, thus maintaining consensus throughout the network.

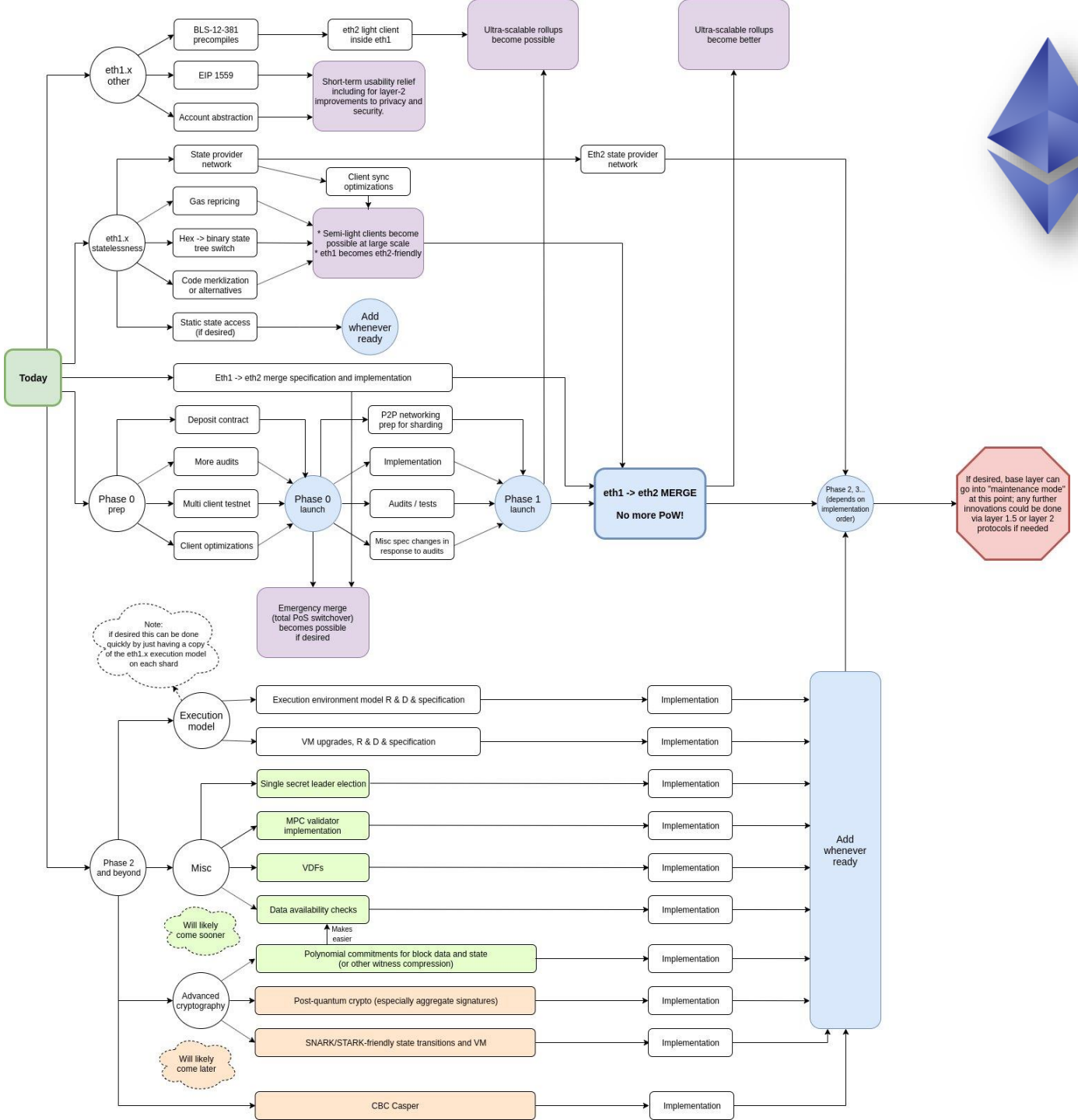

PoW Main Chain
"Anchor Layer"
Beacon Chain
"Coordination Layer"
Shard Chains
"Data Layer"
VM(s)
"Execution Layer"
Shard 1024
Shard 1
B1
B2
B3
B4
B5
B1 state root
B2 state root
B3 state root
B4 state root
B5 state root

Is Ethereum 2 a **fork**?

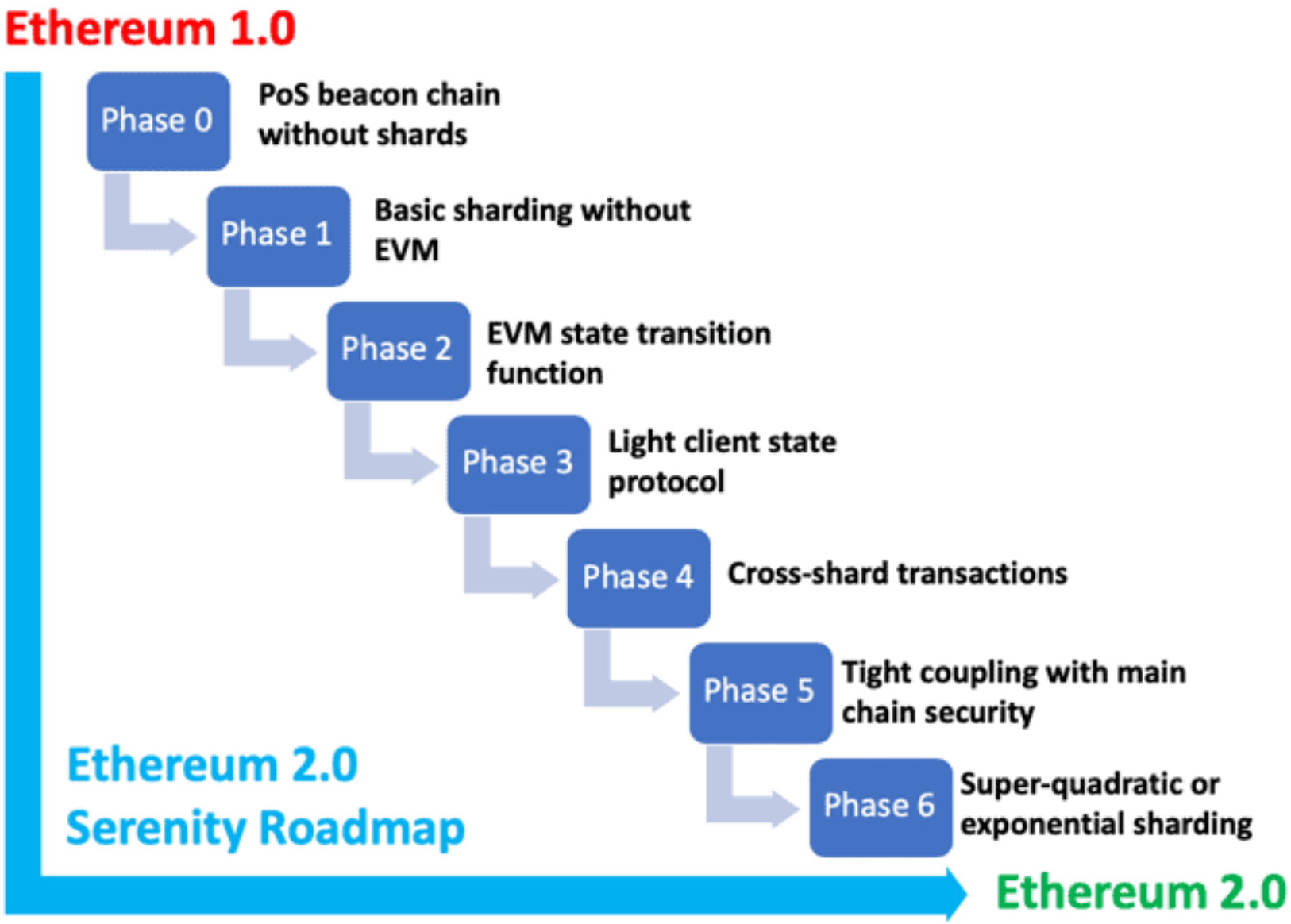
- **No**. Since its inception, Ethereum has undergone several upgrades in the form of hard forks. With Ethereum 2, rather than forking, a whole new blockchain based on proof-of-stake is being launched. This will initially run in parallel to the legacy chain and will be rolled out in several phases.

Roadmap

- Tweet [link]

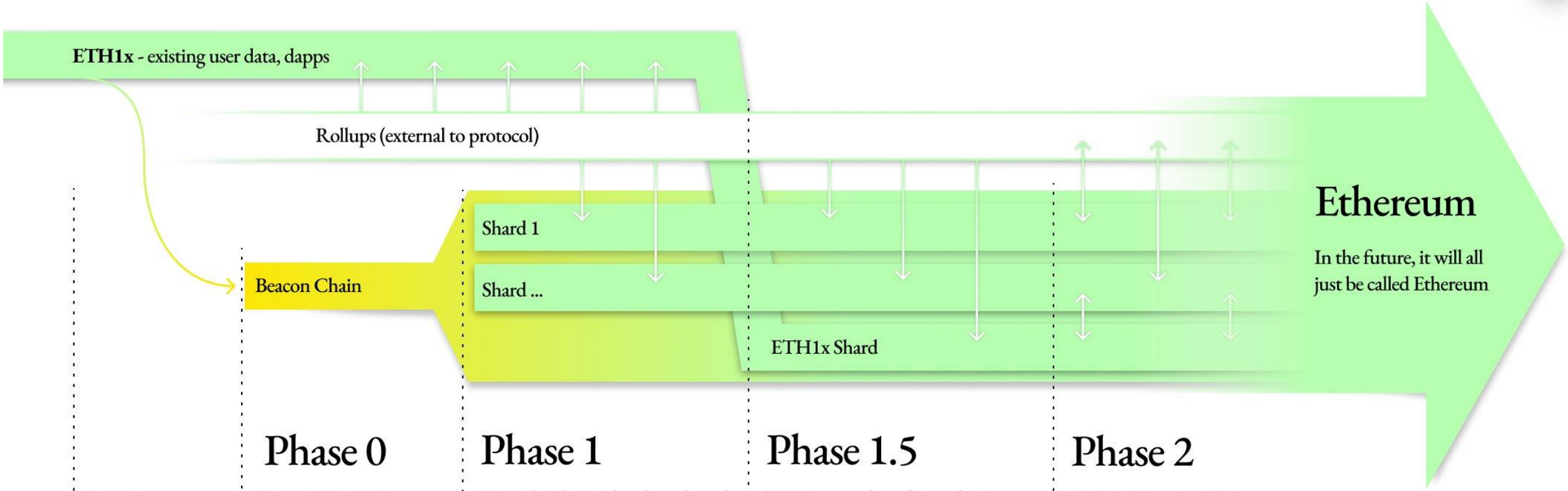


Roadmap


Ethereum 1.0
Phase 0
PoS beacon chain without shards
Phase 1
Basic sharding without EVM
Phase 2
EVM state transition function
Phase 3
Light client state protocol
Phase 4
Cross-shard transactions
Phase 5
Tight coupling with main chain security
Phase 6
Super-quadratic or exponential sharding
Ethereum 2.0
Serenity Roadmap
Ethereum 2.0

How they relate: ETH1x, ETH2, & Rollups

Phases not shown to scale - graphic by @trent_vanepps



Deposit contract state bootstraps Phase 0 Genesis.

Phase 0

Proof of Stake Beacon Chain starts and lives on its own for a bit. Rollups can leverage ETH1x for Data Availability ↑

Phase 1

Data Sharding - Shards anchored to the Beacon Chain, but without native transaction execution. Rollups / L2 can now use Shards for Data Availability ↑ very useful for offchain scaling.

Phase 1.5

ETH1x gets slotted into the Beacon Chain architecture as a single shard. This connects existing ETH1x user state to PoS consensus, replacing miners and PoW.

Phase 2

Existing Shards gain the ability to execute transactions natively ↕. Research continues on the optimal integration of this feature.