LEARNING HYPERLEDGER

1. Introduction to H.L Blockchain Technologies

i) Distributed Ledger Technology (DLT)

DL → a data structure residing over multiple computer devices, can be spread across locations or regions

3 components:

1. Data Model to capture current state of ledger
2. Language of transactions to change ledger state
3. Protocol to build consensus on which txns. will be accepted, in what order, by the ledger

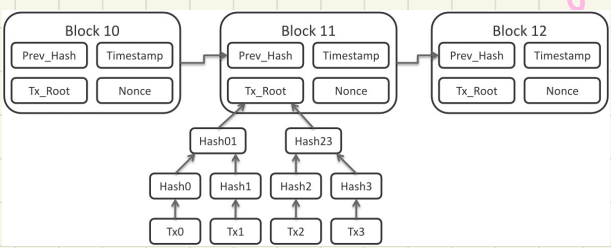
examples of DLTs → block chain, chain core, Corda, quorum, IOTA

- Blockchain → P2P DL, forged by consensus, combined with a system for smart contracts & other assistive technologies
- Smart Contracts → Programs which execute predefined actions when certain conditions within the system are met.
- Consensus → System ensuring that parties agree to a certain state of the system as the true state.
- Block → a set of txns bundled together, added to the chain at the same time
- Timestamping → Each block is timestamped, every new block refers to the previous block. With timestamps + cryptographic hashes → the chain provides an immutable record of all txns, from the 1st (genesis) block.

Blockchain → miner nodes bundle valid + unconfirmed txns into a block. Each block contains a certain no of txns. On Bitcoin, miners solve a cryptographic challenge to propose the next block → This is PROOF OF WORK

Bitcoin Blockchain Block:
- reference to previous block
- Timestamp
- PoW or nonce
- Merklee tree root for txns in the block

- Merkle Tree / Binary Hash Tree
  → a data str. used to store hashes of individual data in large datasets, & allow efficient verification of the dataset. It's an anti tamper mechanism to ensure the dataset hasn't been changed



- Transactions → Records of events, cryptographically secure with a digital signature, which is verified, ordered & bundled together into blocks.
  On bitcoin b.chain → txns = transfer of bitcoins
  Generally, txns = transfer of any asset or record of a service being rendered.

- Cryptography → Allows secure communication b/w diff. parties & ensures authenticity & immutability of data. For b.chains:
  - it proves that a txn was created by the right person,
  - links txns into a block immutably
  - creates link b/w blocks i.e. the b.chain

- Blockchains vs Databases

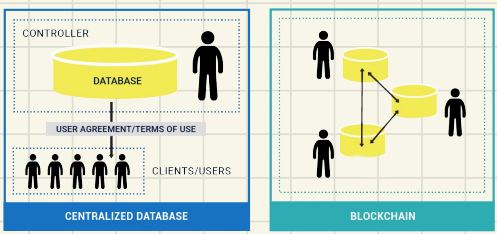
→ Write only structure. New entries only get added to the ledger. New block is linked to the previous block's hash. No editing is allowed.

→ Used in decentralized apps

→ Data can be modified/deleted

→ Used in centralized apps

CENTRALIZED DATABASES VS. BLOCKCHAIN



- Types of blockchains:

1) Permissionless/Public
↳ Anyone can join the n/w

→ Useful for apps, where parties can transact, w/o verifying each others identities
e.g. bitcoin, cryptocurrencies

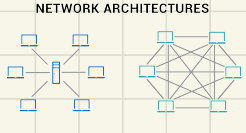
2) Permissioned / Private
↳ Requires pre-verification of participating parties, which are known to each other

↳ Useful when parties need to be vetted before they can transact, like in supply chain management.
e.g. Hyperledger Blockchains

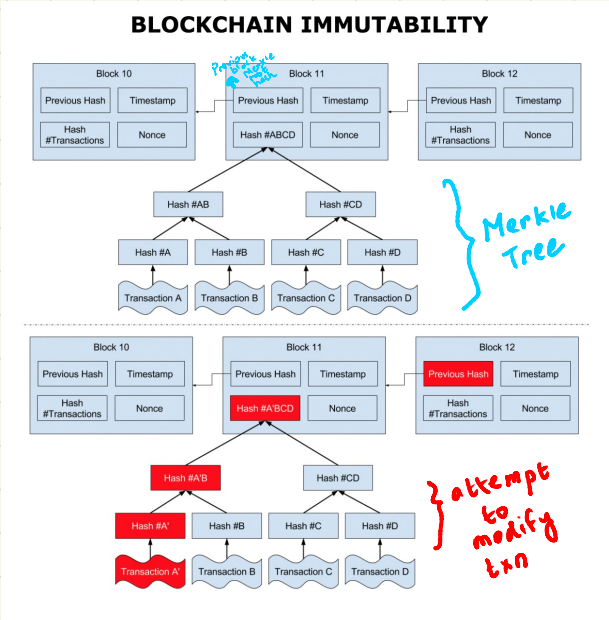
- Peer-to-Peer Network Architecture
  - → Computers which are directly connected to each other w/o a central server. Peers contribute to the compute & storage required for the n/w.
  - → more secure than traditional centralized n/ws which have a single point of failure → the central server is.

Permissionless P2P → Do not require a min no. of online peers, hence slower

Permissioned P2P → Requires peers to be online, to ensure uptime & quality of service



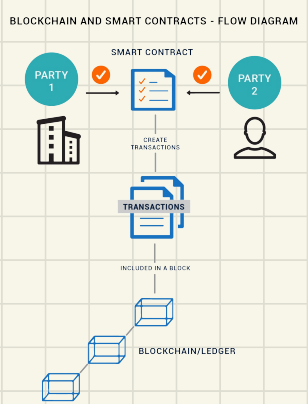
- IMMUTABILITY



Once a txn is written onto the b.c, its difficult to change it.
If there is an attempt to collude, its easy to detect
⇒ If a txn is changed, its own hash will change + the prev. hash of next block has to be updated for proof of work ⇒ which would be a red flag

- SMART CONTRACT

↳ Programs which execute predefined actions when certain conditions are met. SCs provide the language of transactions that allow a ledger's state to be modified



ii) Bitcoin & Ethereum Blockchains

- Bitcoin → P2P cash system / a global decentralized payment n/w with a distributed & publicly owned infrastructure, operating as a permissionless system

Permissionless b.chains are open for anyone. The key is to incentivize good behavior, so:

i) Malicious actors can't take over the n/w through an attack
ii) Malicious actors can't collude to undertake an organized majority attack
iii) Payoffs of securing > cost of attacking the n/w
iv) Cost of attacking the n/w is prohibitively high

- Ethereum → Open b.c. platform that lets anyone build & use d.apps, facilitates scripting functionality → smart contracts which are run through the nodes in the n/w.
Unlike bitcoin, Ethereum not only tracks txns, but also programs them.

- Dapps → Apps developed on decentralized consensus based n/w s.
Malicious actors can't secretly tamper with a dapp by changing the app's code & compromising all users.
→ They don't run on central servers at one place, instead the computing power that runs the n/w is contributed by nodes spread across the globe. Hence, dapps have zero downtime

- Eth Smart Contracts

A hypothetical example of an Ethereum-based smart contract may involve the following transaction: in an equity raise, transfer amount X from the investor to the company upon receiving the given shares from the company. The monetary amount X, which was pre-validated by the company for the transaction (much like in a credit card purchase), is held in escrow by the smart contract, until the shares have been received by the investor. Any kind of arbitrary sophisticated business logic can be committed to the blockchain. The Ethereum blockchain only encodes these "rules of the games". The actual payoffs occur by interacting with the blockchain.

The illustration below describes this process. The smart contract encodes the agreement between the company raising funds and its investors (Panel 1). The smart contract sits on the Ethereum public blockchain, and is run on the Ethereum Virtual Machine (EVM). Once hitting a triggering event, like an expiration date or a strike price that has been pre-coded, the smart contract automatically executes as per the business logic (Panel 2). As an added benefit, regulators are able to scrutinize the market activity on an ongoing basis, without compromising the identity of specific players in a permissionless public blockchain, as Ethereum (Panel 3).

iii) Permisionless Blockchains

- **Consensus Algorithms**

Consensus → Process of achieving agreement amongst the n/w participants as to the correct state of data on the system.

Consensus Algorithm → i) Ensures that the data on the ledger is same for all nodes in the n/w.

ii) Prevents malicious actors from manipulating the data.

Examples of such algos → Proof of work (bitcoin), Proof of stake, Proof of burn, Proof of Capacity, Proof of elapsed time

1) Proof Of Work

→ Solving a challenging computational puzzle in order to create new blocks in the Bitcoin b.c. The process is called MINING, & nodes engaged in mining are called MINERS.

Incentive for mining fxns → competing miners are rewarded with 12.5 bitcoins & a txn fee.

PoW is the outcome of a successful mining process, i.e. hard to create, easy to verify

Explainer article on Medium Making a POW blockchain Another impl Pow blockchain in Go Ethereum POW (used till 2022)

Drawbacks: → Algorithm is computationally heavy, so a huge amount of energy has to be expended

→ High latency of txn validation

→ Concentration of mining power in countries with cheap electricity.

→ Succeptible to 51% attack (attack on b.c by a group of miners controlling >50% of the nw's compute.

2) PROOF OF STAKE → generalization of PoW

- Nodes are called validators, and instead of mining the b.c, they validate the txns to earn a txn fee.
- No mining needs to be done, since all coins exist from the start.

- Nodes are randomly selected to validate blocks.

  More the stake (coins) owned by a node, more is its likelihood to be called upon to validate a block of txns.

Benefits → • saves expensive computational resources that would've been used for mining in PoW

Pos blockchain in Go Ethereum POS POS Research Paper

3) Proof of Elapsed Time (PoET)

Emulates Bitcoin's PoW. Instead of competing to solve a cryptographic challenge & mine the next block, in PoET, there's a hybrid of random lottery & first come first serve basis. Each validator is given a random wait time. The one with the shortest wait time is the leader, who gets to create the next block on the chain. e.g. Hyperledger Sawtooth

todo: study Sawtooth's architecture

4) Simplified Byzantine Fault Tolerance (SBFT)

→ Single validator who bundles proposed txns & forms a new block. Validator is called party. Consensus is achieved when a min. no. of other nodes ratify the new block.

For BFT, 2f+1 nodes must reach consensus in a system containing 3f+1 nodes. eg Bytcoin Medium Explainer Paper

5) Proof of Authority

→ Set of authorities which are designated nodes that are allowed to create new blocks & secure the ledger. Sign-off by majority of authorities is required to create a block

Paper PBFT vs PoA

iv) Hyperledger → Collection of b.chain projects, hosted by the Linux Foundation. Makes enterprise ready b.chain solutions for finance, banking, IOT, supply chain, healthcare, etc.

- Permissioned blockchains, parties that join are authenticated & authorized to participate on the n/w.
  Since its permissioned, it offers more efficient txn performance, is highly scalable & has a well defined governance structure

Todo: Revisit this course & study the Hyperledger Projects & literature around it, for now I'm moving to the white papers & actual technical implementations

HYPERLEDGER ARCHITECTURE

Whitepaper Link →