Blockchain Data Structures

Centralized Vs Decentralized

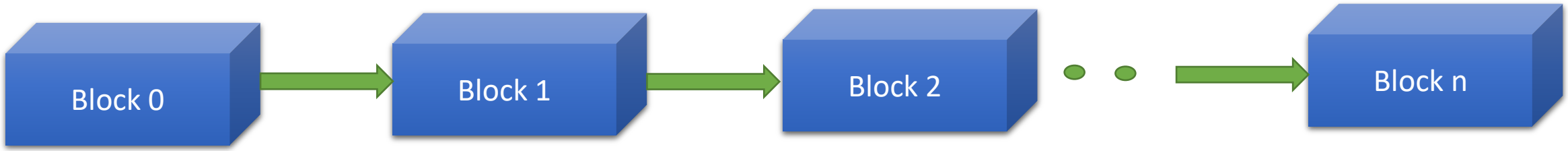
I

II

Image Source : Internet

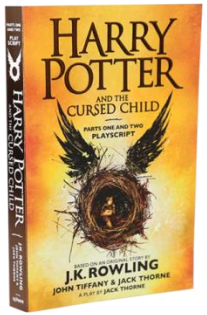
Blockchain: Definition and Features

Blockchain technology is a **distributed ledger technology** originally proposed for the crypto-currency **Bitcoin**



Blocks = Pages

Blockchain = Book



Blockchain: Definition and Features

FEATURES

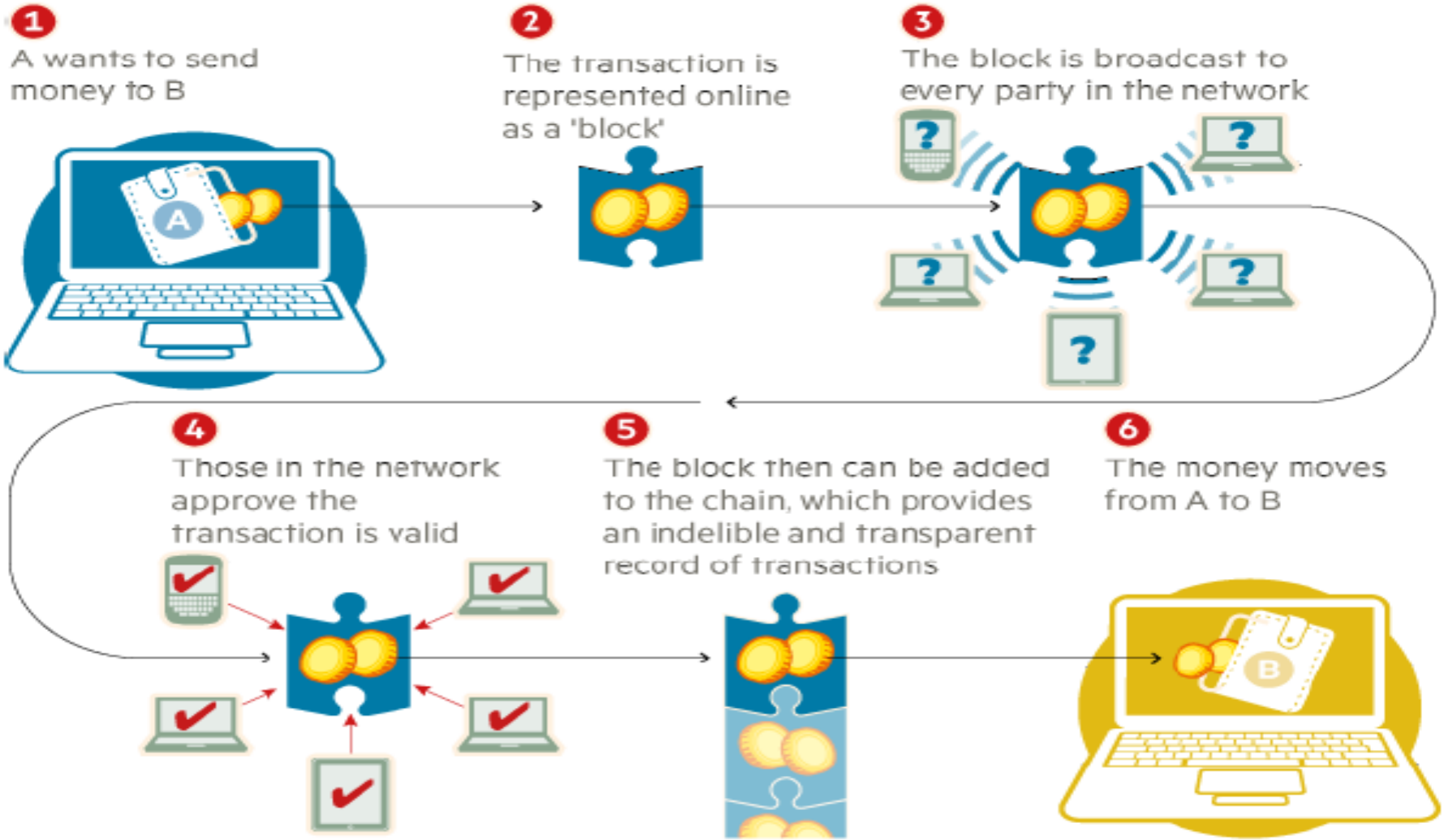
- Immutable and tamper-proof data store
- Sequential Chain with Cryptographic hashing
- Trust-free Consensus-based transactions
- Decentralized peer-to-peer network
- Distributed shared ledger

Bitcoin: The First Blockchain

- Bitcoin is the first successful electronic cash system and coincidentally, the first instance of a successful Blockchain.
- Secure, trustless, borderless
- No bank needed to authorize/process transactions
- Transactions are stored on a **distributed ledger**
- Introduced by **Satoshi Nakamoto** on **Oct. 31**, **2008**. https://bitcoin.org
- Only 21 Million coins
- First block created on January 3, 2009

*Bitcoin introduced the concept of **cryptocurrency**; decentralized digital money secured by cryptography, and used to create valuable digital assets that cannot be counterfeited.*

Bitcoin Blockchain: How it works?



Source : Financial Times

Bitcoin: How it works?

Bitcoin transactions are authorized in a peer-to-peer network.

- Each node stores the ***history*** of the chain of blocks, containing ***validated*** transactions
- Counterfeiting is ***impossible*** because if one node's history is corrupted the others stay the same, and no central authority (i.e. bank) needs to confirm; this is called **decentralization**
- Unlike previous P2P network models, members of the Bitcoin network are ***incentivized*** to participate through **cryptocurrency**.
- Specifically, the incentive is for the people who mint (create) Bitcoin, called **miners**.

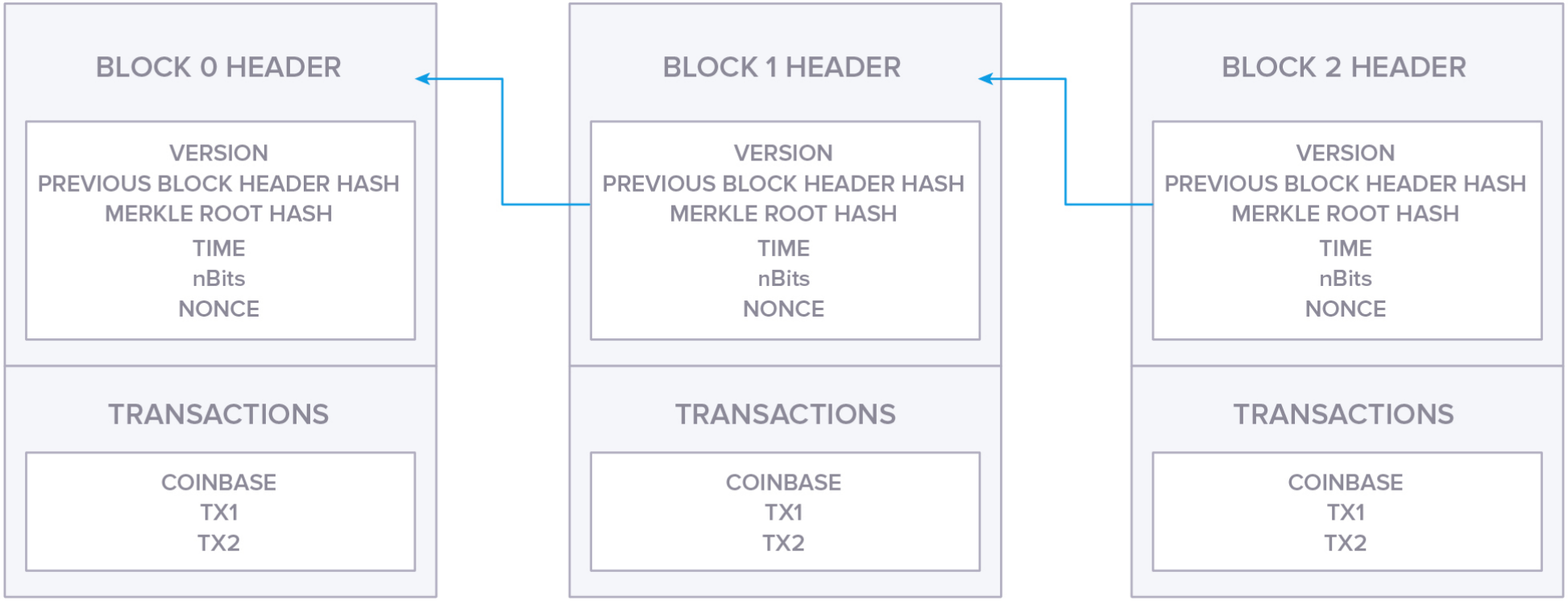
The total address space of Bitcoin is 2^160

i.e. 1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976

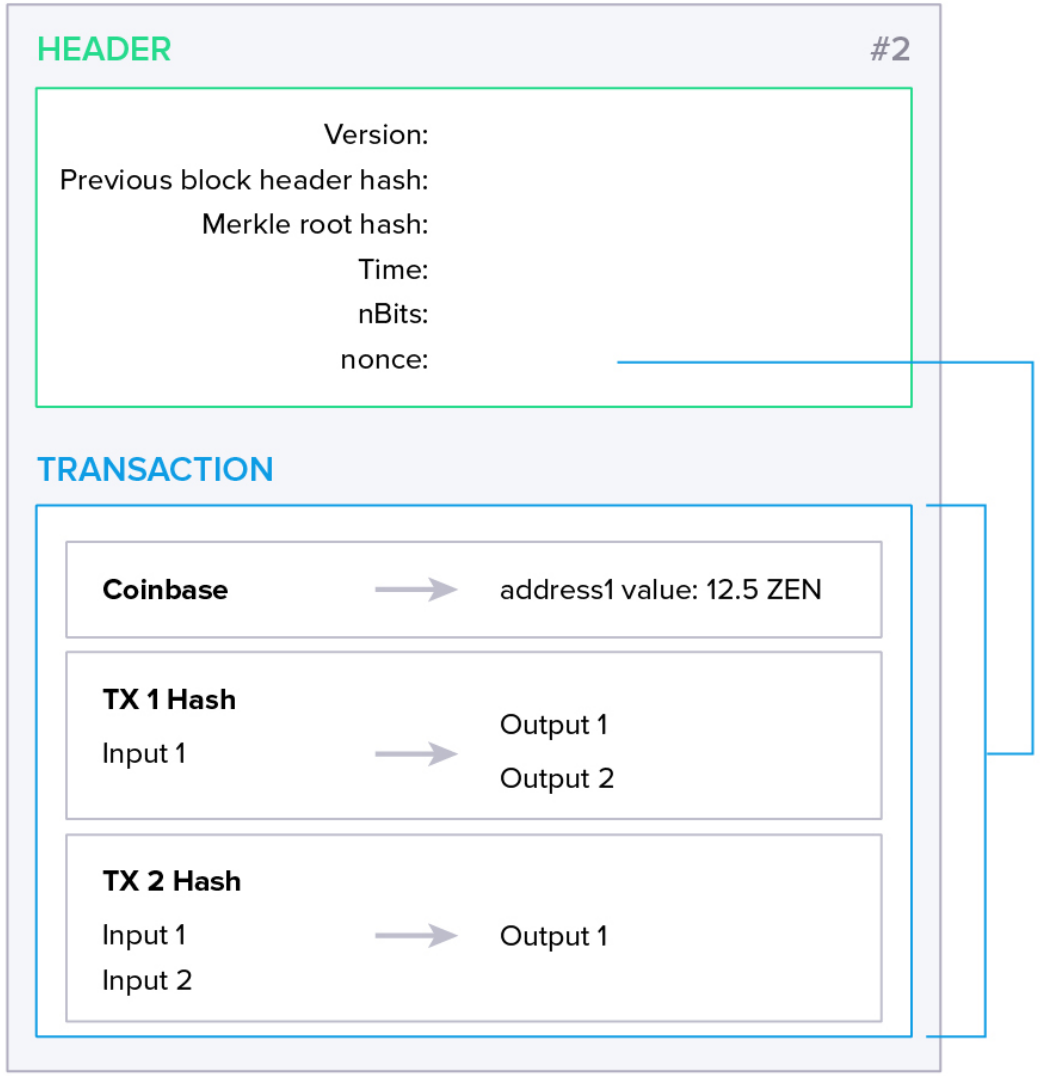
there are an estimated 2^63 grains of sand on Earth

World Population as of March 2020 - 7,800,000,000 (7.8 Billion)

Structure of a Block



Structure of a Block



Structure of a Bitcoin Block

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the Merkle-Tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The Proof-of-Work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

Block Header

- Difficulty target value (called *bits* in Bitcoin)
- **Merkle root** of all transactions in the block
- *Nonce*; a value changed in mining to find accepted blocks
- *Previous block hash (block ID)*; linking this block to the previous block in chain (this previous hash is named *parent* in figure above)
- *Timestamp*; Time of mining (creating the hash for) the block
- *Block version*; identifies support features and formats

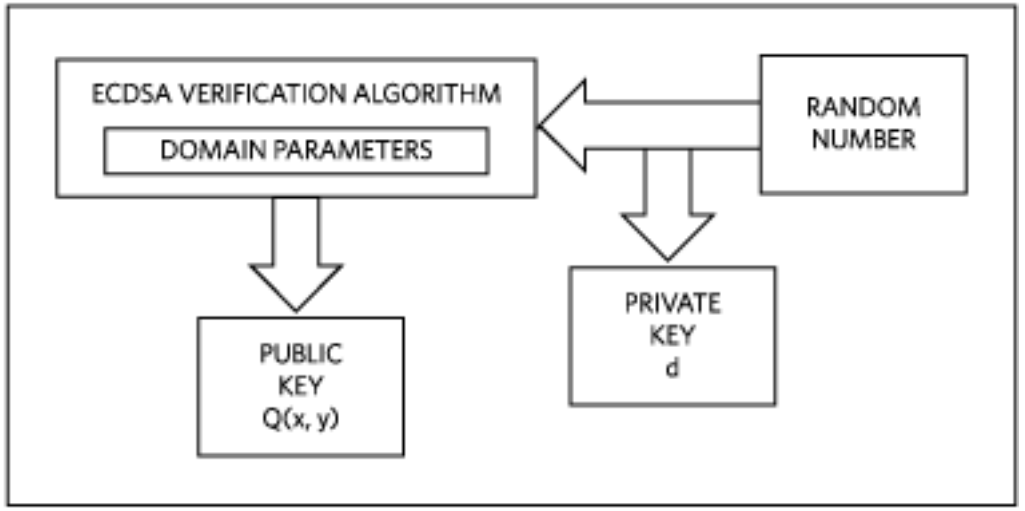
Elliptic Curve DSA (ECDSA)

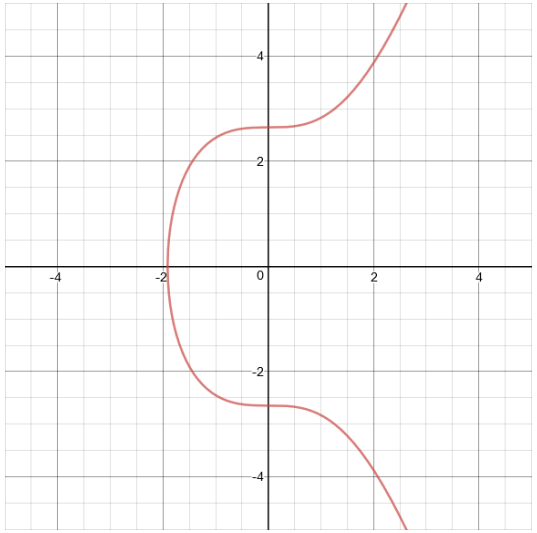
- Trapdoor Function generating A -> B

$$Y^2 = x^3 + ax + b$$

- Reverse is practically impossible, Symmetric to x-axis.
- Straight line intersecting 2 points will touch at most 1 point.
- Less key size when compared to RSA algorithm
- 256 bit key of ECDSA equals 3072 bits of RSA.
- Bitcoin and Ethereum uses **secp256k1** ECDSA

$$Y^2 = x^3 + 7$$



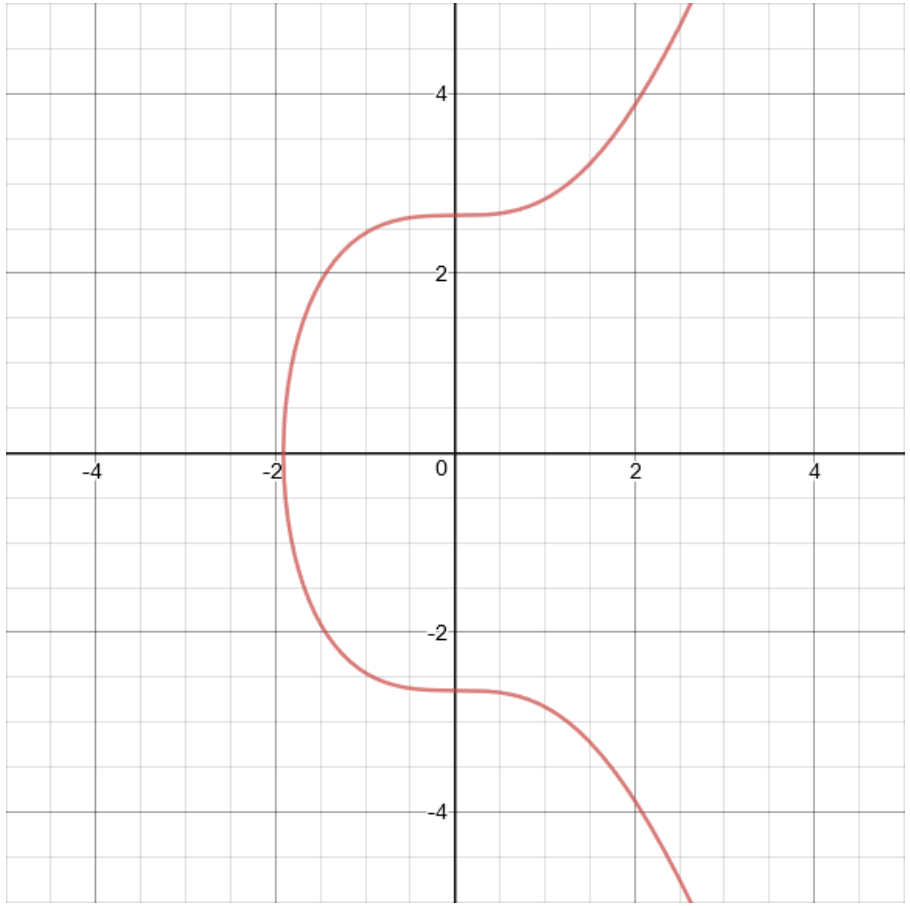


secp256k1 ECDSA

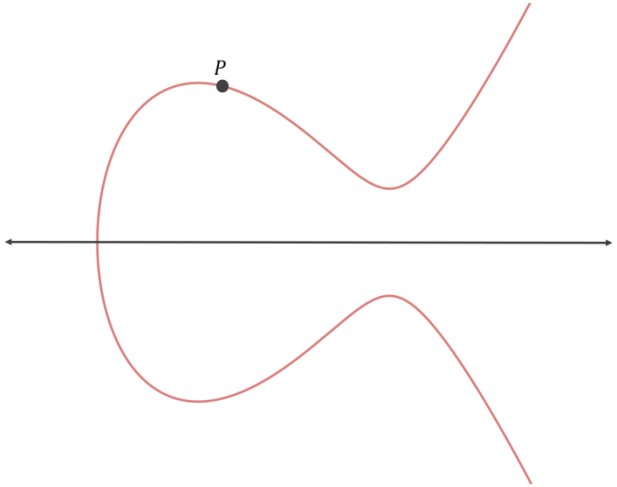
- Bitcoin and Ethereum uses **secp256k1** ECDSA

$$Y^2 = X^3 + 7$$

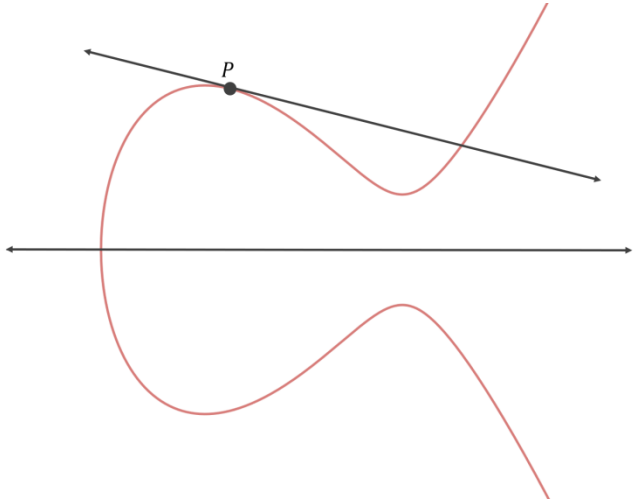
- Take a point, say P
- Draw tangent and find intersection to curve
- Flip P to x-axis and repeat..



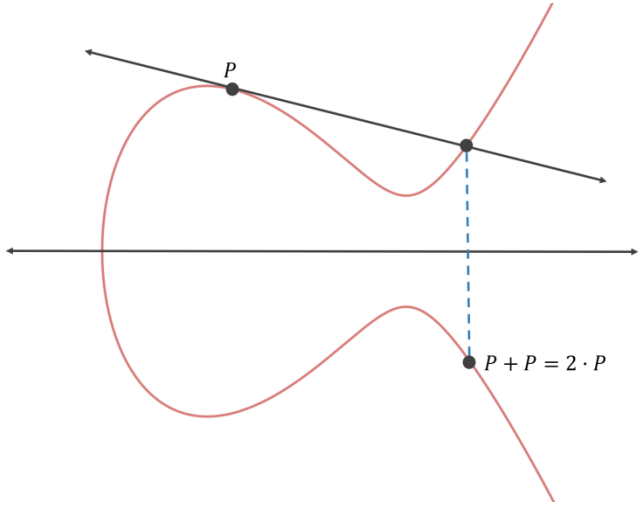
secp256k1 ECDSA



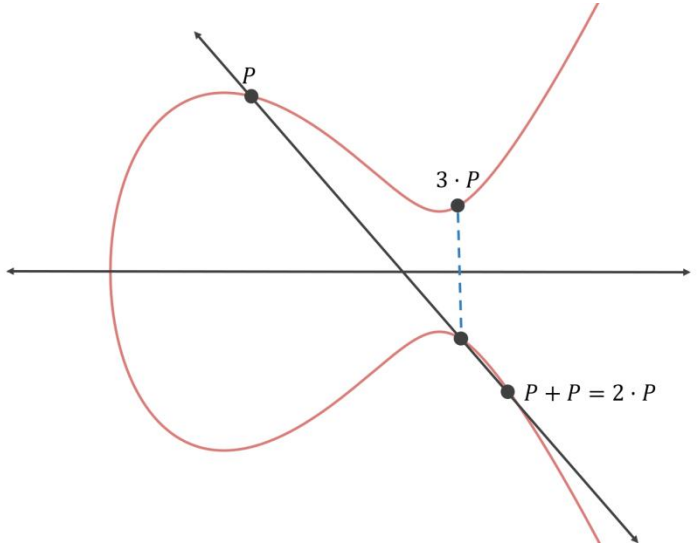
Step 1 :
Take point P.
Add P to itself



Step 2 :
Draw Tangent to P.

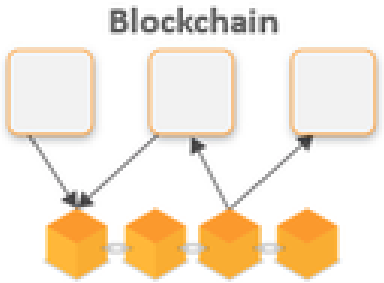


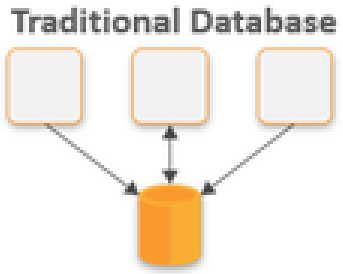
Step 3 :
line intersects and reflect it across the x-axis.



Step 4 :
Repeat process

Blockchain vs Database

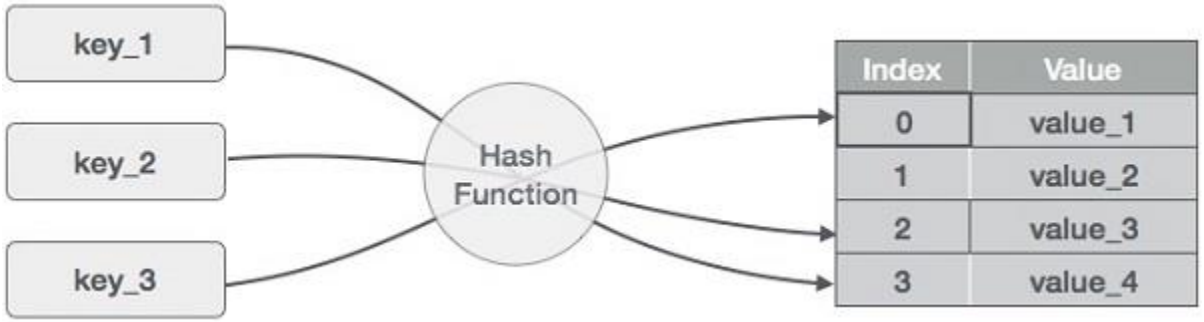




Properties	Blockchain	Traditional Database
Operations	Only Insert Operations	**Can perform C.R.U.D. Operations**
Replication	**Full Replication of block on every peer**	▪ **Master-Slave** ▪ **Multi-master**
Consensus	**Majority of peers agree on the outcome of transactions**	**Distributed transactions (2 Phase Commit)**
Invariants	**Anybody can validate transactions across the network**	**Integrity Constraints**

The Data Structure

Hash Table



- a type of “key-value store”.
- This means that for a given “key” (i.e. a vehicle type) you can save the “value” (i.e. the name of the vehicle).
- The main property of the hash table is that when you have a key, you can find the value fast, regardless of how many other items are in the hash table

- Problems?

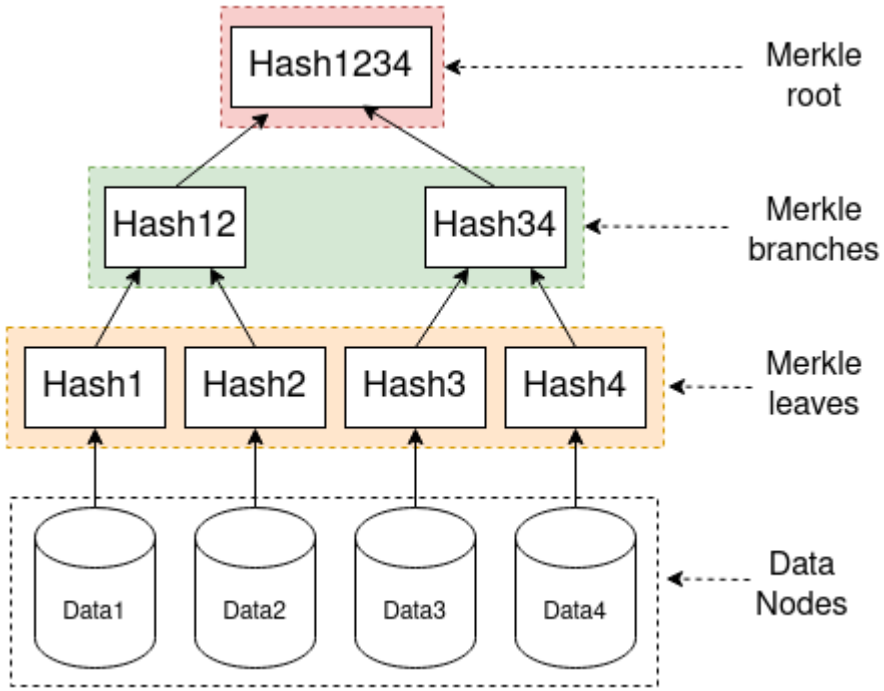
Hash table

- Consider a College with Library
  - Student have unique Roll No.
  - Library Books have Unique Code.

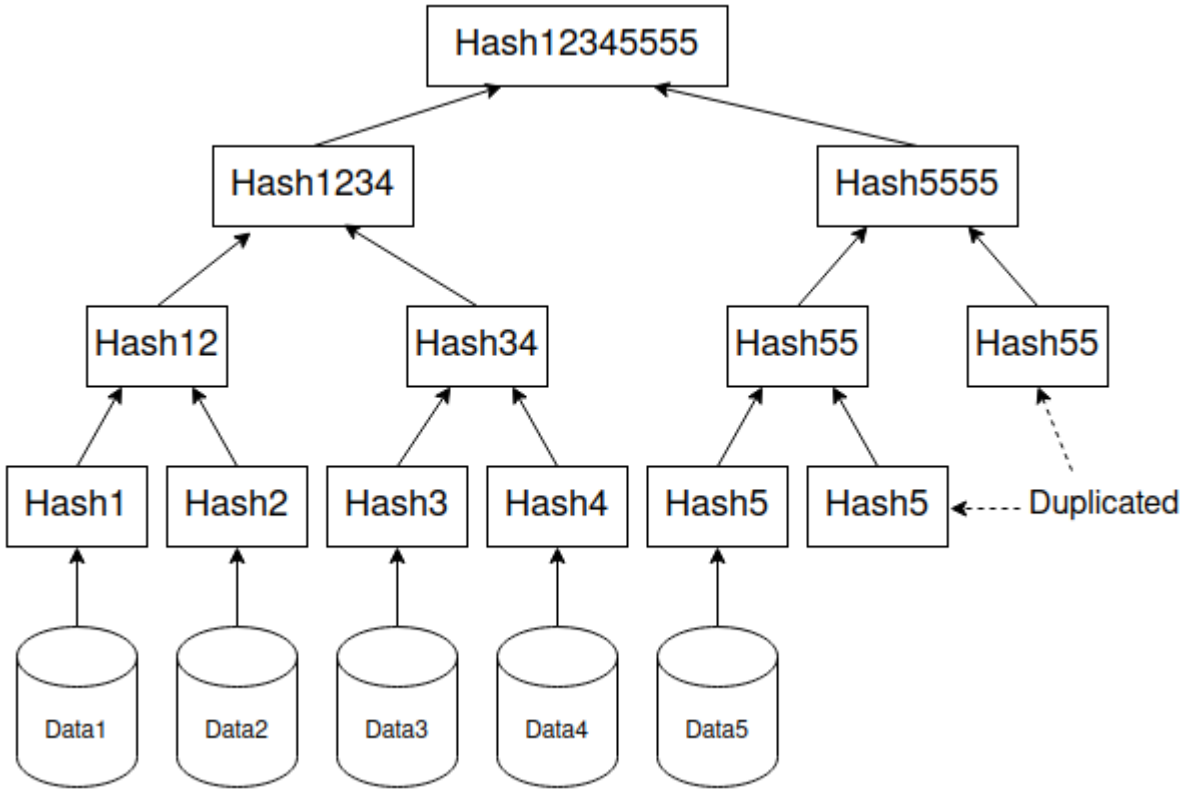
Merkle Tree (Binary Hash Tree)

- A *Merkle tree* is a data structure used within blocks.
- The transactions in a block make up the ***leaves*** of the Merkle tree.
- The resulting *Merkle root* serves as a summary of all transactions and is included in the block header.
- A Merkle tree is fundamentally just a hierarchical set of hash values,
  - building from a set of actual data (*Merkle leaf*)
  - to intermediate hashes (*Merkle branches*) and
  - up to the *Merkle root* that summarizes all the data in one hash value.

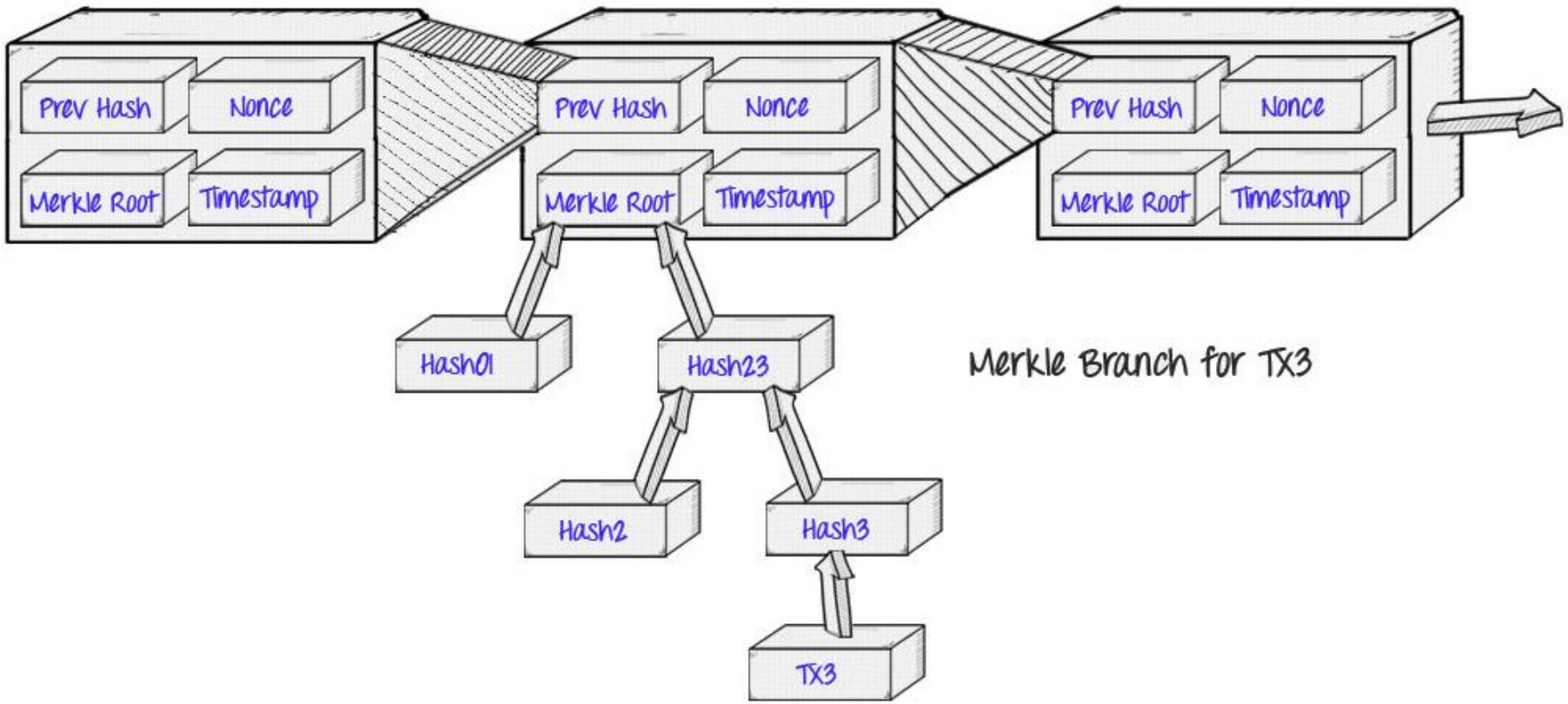
Merkle Tree


Hash1234
Merkle root
Hash12
Hash34
Merkle branches
Hash1
Hash2
Hash3
Hash4
Merkle leaves
Data1
Data2
Data3
Data4
Data Nodes

Merkle Tree - Unbalanced



Merkle Tree in Bitcoin



Why Merkle Tree?

- So what is the benefit of this strange kind of hashing algorithm?
- Why not just concatenate all the chunks together into a single big chunk and use a regular hashing algorithm on that?

Merkle Proofs..!!

- A Merkle proof consists of a chunk, the root hash of the tree, and the "branch" consisting of all of the hashes going up along the path from the chunk to the root.
- Someone reading the proof can verify that the hashing,
  - at least for that branch, is consistent going all the way up the tree, and
  - therefore that the given chunk actually is at that position in the tree.

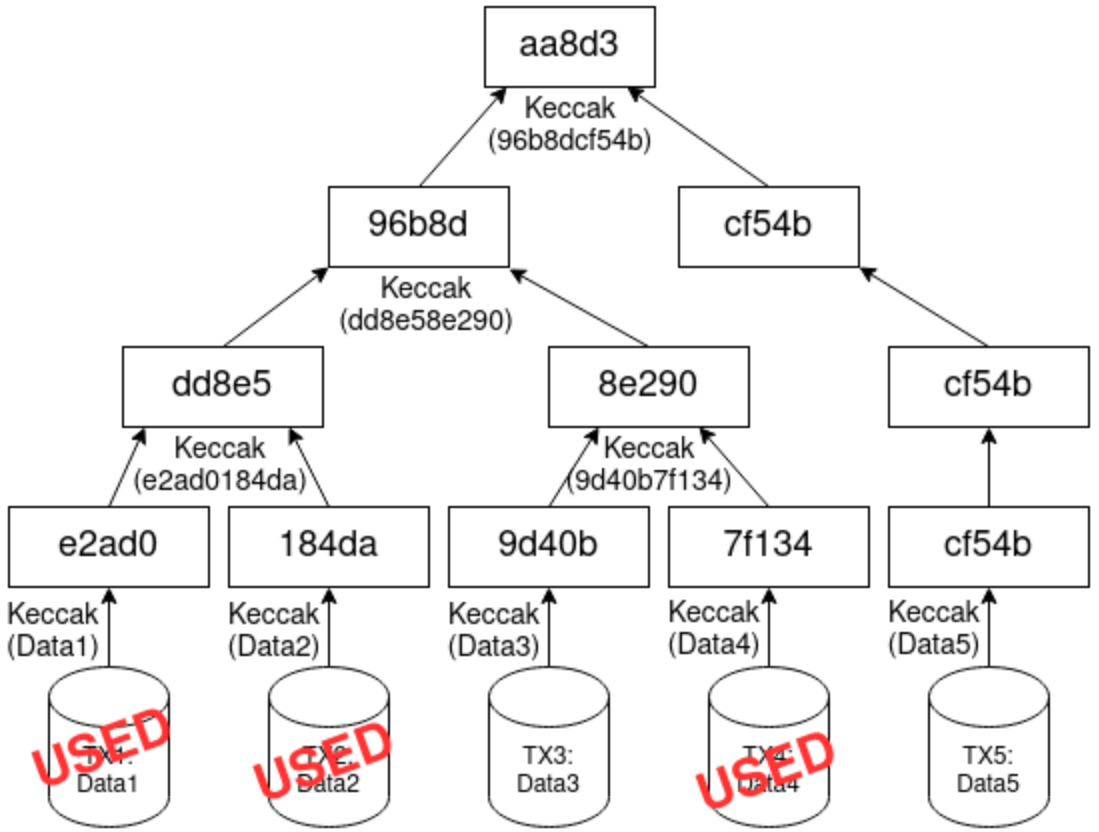
Merkle Proofs..!!

- The application is simple:
  - suppose that there is a large database, and
  - that the entire contents of the database are stored in a Merkle tree
  - where the root of the Merkle tree is publicly known and trusted (eg. it was digitally signed by enough trusted parties, or there is a lot of proof of work on it).
- Then, a user who wants to do a key-value lookup on the database (eg. “tell me the object in position 85273”) can ask for a Merkle proof, and
  - upon receiving the proof verify that it is correct,
  - and therefore that the value received *actually is* at position 85273 in the database with that particular root.
  - It allows a mechanism for authenticating a *small* amount of data, like a hash, to be extended to also authenticate *large* databases of potentially unbounded size.

Blockchain Pruning – Data management

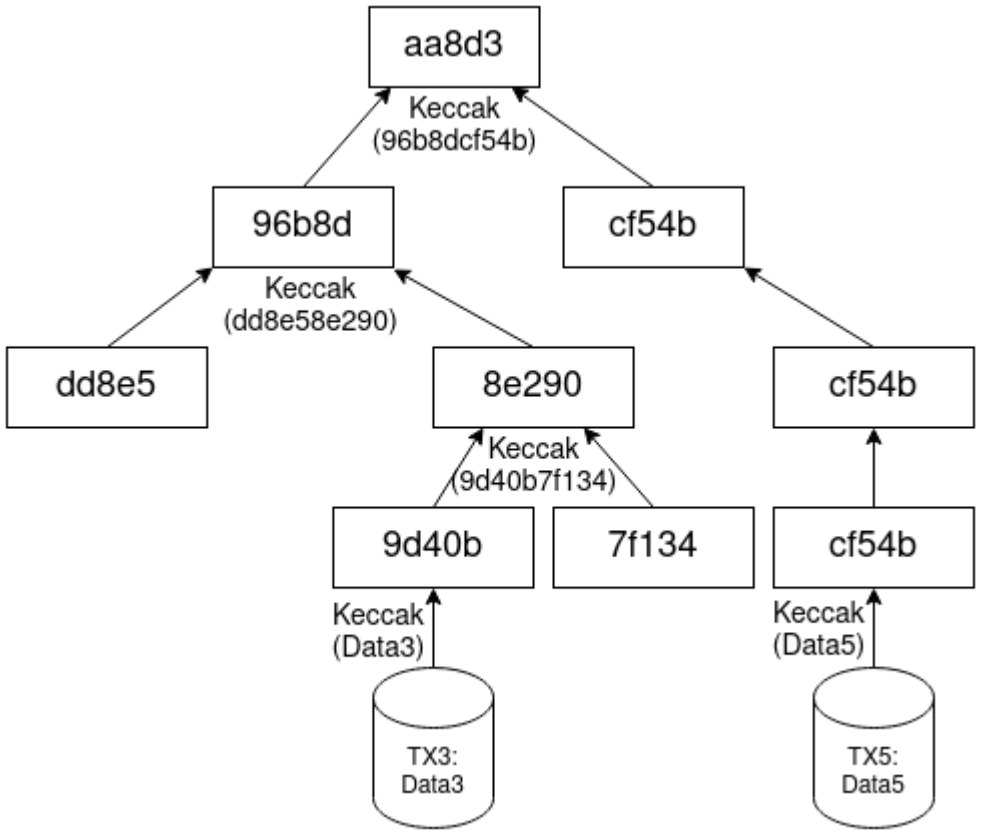
- Satoshi's proposal for data management for non-full nodes is to prune the block from the used transaction data.
- Size of Bitcoin Blockchain as of Feb 2021 is over 380 GB
- Leave the Merkle tree branches needed to verify the unspent transaction data

Blockchain Pruning – Data management



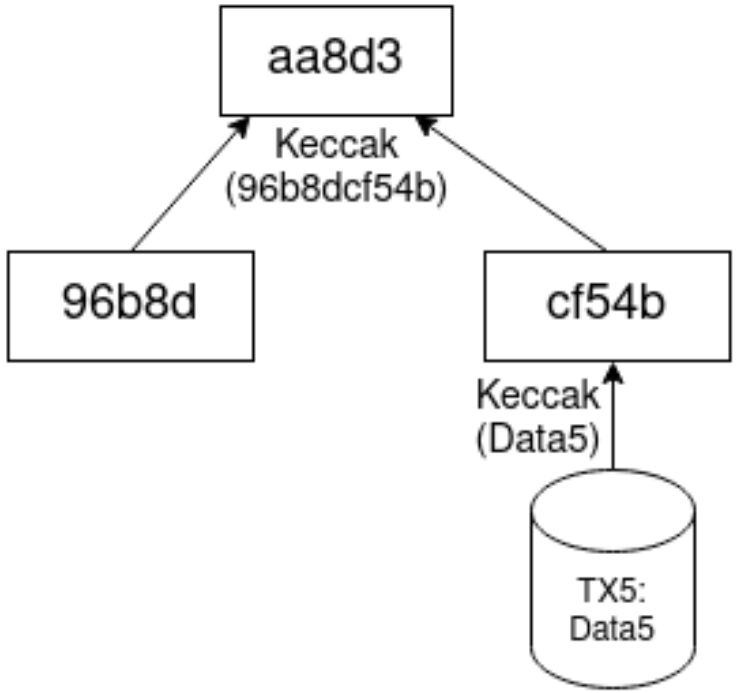
Pruned Merkle Tree

- With T3 and T5



Pruned Merkle Tree

- With T5



Directed Acyclic Graph (DAG)

- **Directed**

  The links point in the same direction with earlier transactions linked to later transactions, and so on.

- **Acyclic**

  Loops are not possible. A transaction cannot loop back on itself after linking to another transaction.

- **Graph**

  The mesh of connected transactions can be represented as nodes in a graph network, in which nodes are joined to each other by links.

Directed Acyclic Graph (DAG)

- In mathematics a DAG is a graph that travels in one direction without cycles connecting the other edges.
- This means that it is impossible to traverse the entire graph starting at one edge.
- The edges of the directed graph only go one way.
- The graph is a topological sorting, where each node is in a certain order.

DAG - IOTA

- Each new transaction in IOTA must validate at least two previous transactions before it can be validated.
- An algorithm in IOTA ensures the random selection of transactions for verification, effectively preventing network members from only validating their own transactions.

Use-case…!!

- Can we have an Online Examination System with Blockchain data structure..??

Thank You