Blockchain DLOC Sem VII

NMCPC62 : Cryptocurrency and Blockchain Development

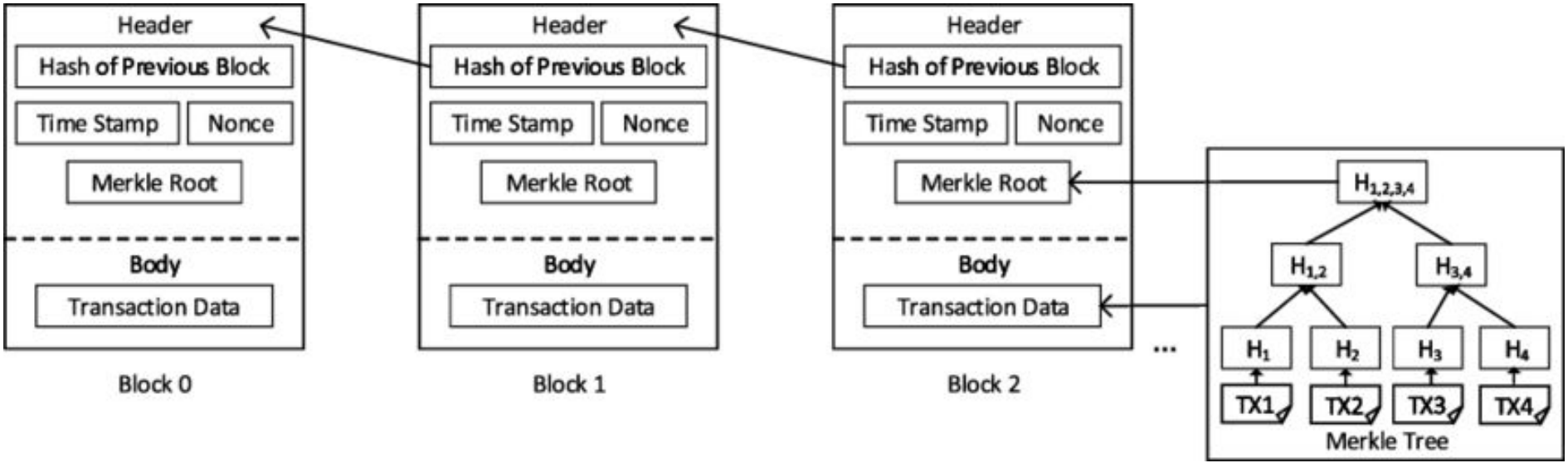
Module - 2: Blockchain Architecture and Platforms (6 Hours)

Instructors : Geocey Shejy, Lifna C S, Pradnya Raut

Topics to be covered

1. Anatomy of a Blockchain:
   - Blocks - Merkle Tree Root Hash,
   - Transactions - UTXO, Transaction fees, How wallet works, SegWit,
   - Nodes,
2. Types of Blockchains: Public, Private, Consortium, and Hybrid,
   - Comparison, Technology Requirements
3. Exploring Key Blockchain Platforms:
   - Bitcoin, Ethereum, Hyperledger, and Binance Smart Chain.
4. Smart Contracts:
   - Concept, Structure, and Applications,
5. Overview of Decentralized Applications (DApps)
6. Token Standards (ERC-20, ERC-721, etc.),
7. Security Aspects of Blockchain: Attacks, Challenges, and Mitigation Techniques

1. Anatomy of a Blockchain - Block



Courtesy : Himanshi Blog

1. Anatomy of a Blockchain - Block

Elements of a Block

- **Block Height –** It's the sequence number of the block in the chain of blocks. Block Height: 1 is the genesis block (first block in the network).
- **Block Size –** It's a 4-bytes or 32-bit field that contains the size of the block. It adds size in Bytes. Ex – Block Size: 216 Bytes.
- **Block Reward –** This field contains the amount awarded to the miner for adding a block of transactions.
- **Tx Count –** The transaction counter shows the number of transactions contained by the block. The field has a maximum size of 9 bytes.
- **Block Header –** The Block header is an 80-Byte field that contains the metadata – the data about the block.

Courtesy : Himanshi Blog

1. Anatomy of a Blockchain - Block

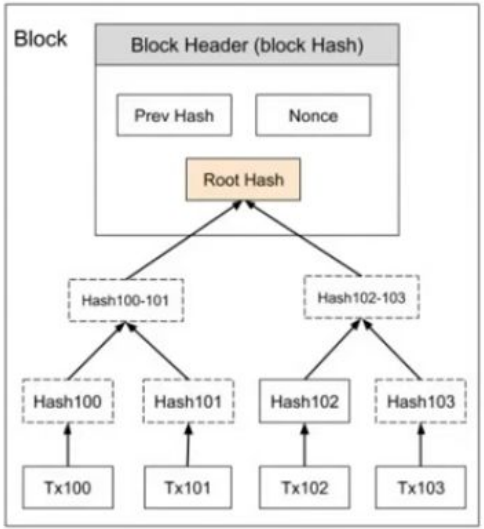
Components of the Block Header.

- **Time**
  - It's the digitally recorded *moment of time when the block has been mined.*
  - It is used *to validate the transactions*.
- **Version** –
  - It's a *4-bytes field* representing the *version number of the protocol used.*
  - Usually, for *bitcoin, it's '0x1*'.
- **Previous Block Hash** –
  - It's a *32-bytes field that contains a 256-bits hash* (created by SHA-256 cryptographic hashing) This *helps to create a linear chain of blocks.*
- **Bits** –
  - It's a *4-bytes field* that tells the *complexity to add the block*.
  - It's also known as "*difficulty bits*." According to PoW, the block hash should be less than the difficulty level.
- **Nonce** –
  - It's a *4-bytes field that contains a 32-bit number*. These are the only changeable element in a block of transactions. In PoW, miners alter nonce until they find the right block hash.
- **Merkle Root** –
  - A *32-bytes field containing a 256-bit root hash*.
  - It's *constructed hierarchically combining hashes of the individual transactions in a block.*

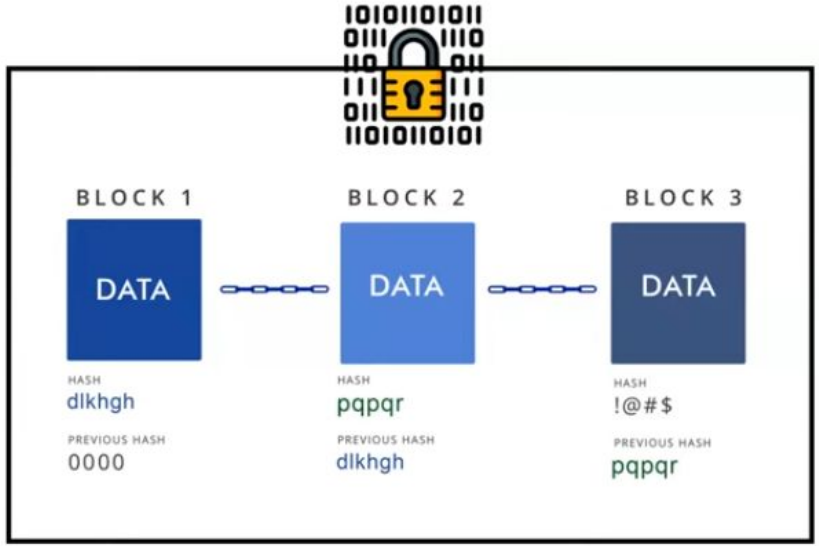
Courtesy : Himanshi Blog
https://explorer.btc.com/btc/block/1000

1. Anatomy of a Blockchain - Block (Merkle Tree)

Merkle trees are a type of data structure commonly used in computer science. They are used to encrypt blockchain data more effectively and securely in bitcoin and other cryptocurrencies.



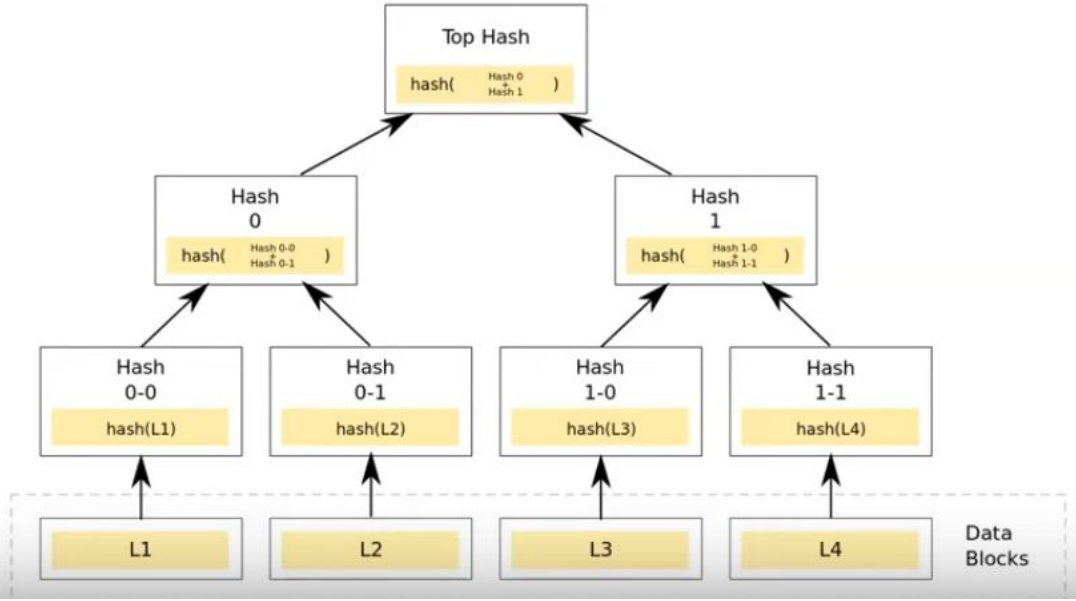
Merkle Tree



Blockchain Data encrypted Securely

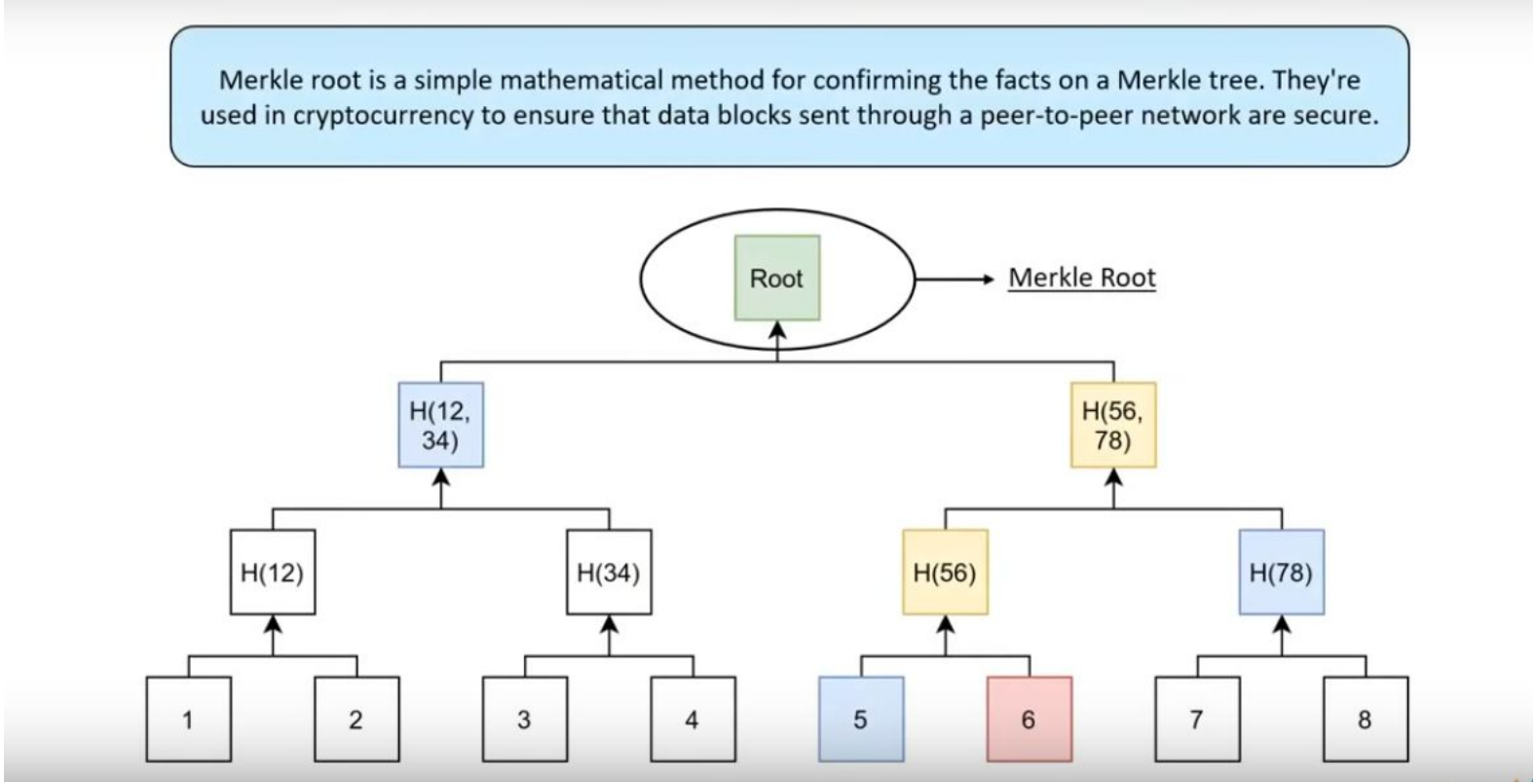
1. Anatomy of a Blockchain - Block (Merkle Tree)

It's a mathematical data structure or a method of organizing data, made up of hash number of various data blocks of transactions performed of the Blockchain Network. It acts as a summary of all the transactions.



Courtesy : Ravikiran A S Blog

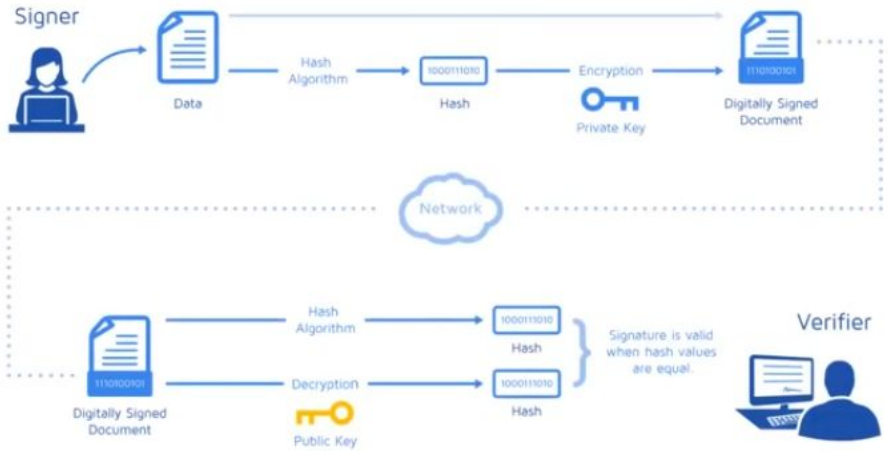
1. Anatomy of a Blockchain - Block (Merkle Tree)



1. Anatomy of a Blockchain - Block (Merkle Tree)

How does a Merkle Tree work ?

A Merkle tree totals all transactions in a block and generates a digital fingerprint of the entire set of operations, allowing the user to verify whether a transaction is included in the block.



1. Anatomy of a Blockchain - Block (Merkle Tree)

How does a Merkle Tree work ?

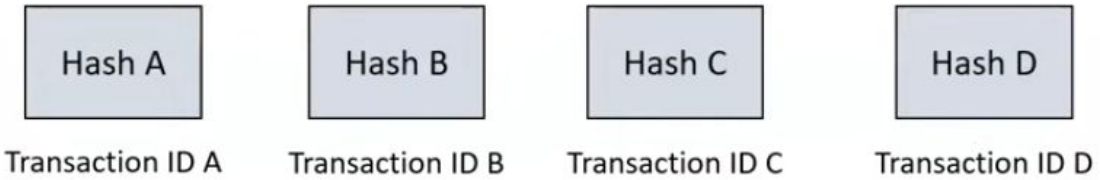
They're built from the bottom, using Transaction IDs, which are hashes of individual transactions. Consider the following scenario: A, B, C, and D are four transactions, all executed on the same block.

Transaction ID A Transaction ID B Transaction ID C Transaction ID D

1. Anatomy of a Blockchain - Block (Merkle Tree)

How does a Merkle Tree work ?

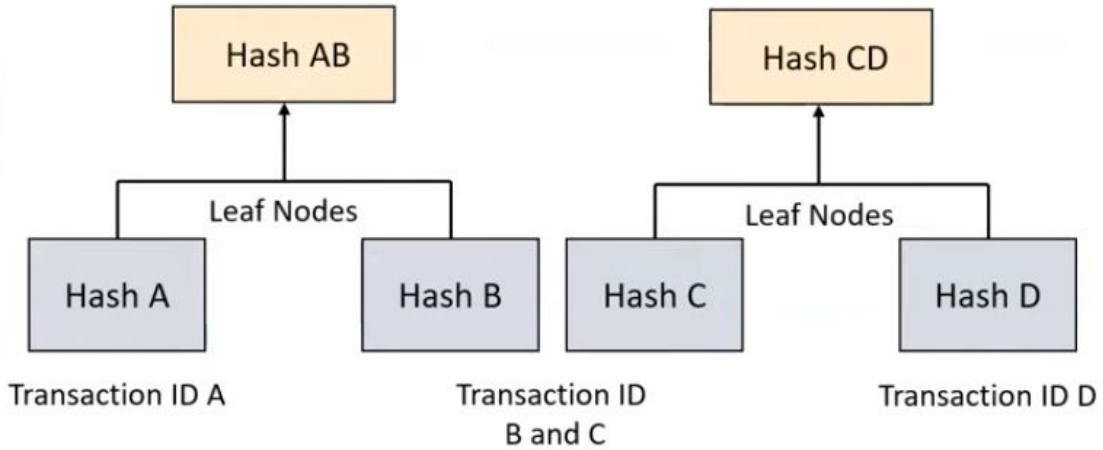
Each transaction is then hashed, leaving us with:
Hash A
Hash B
Hash C
Hash D



1. Anatomy of a Blockchain - Block (Merkle Tree)

How does a Merkle Tree work ?

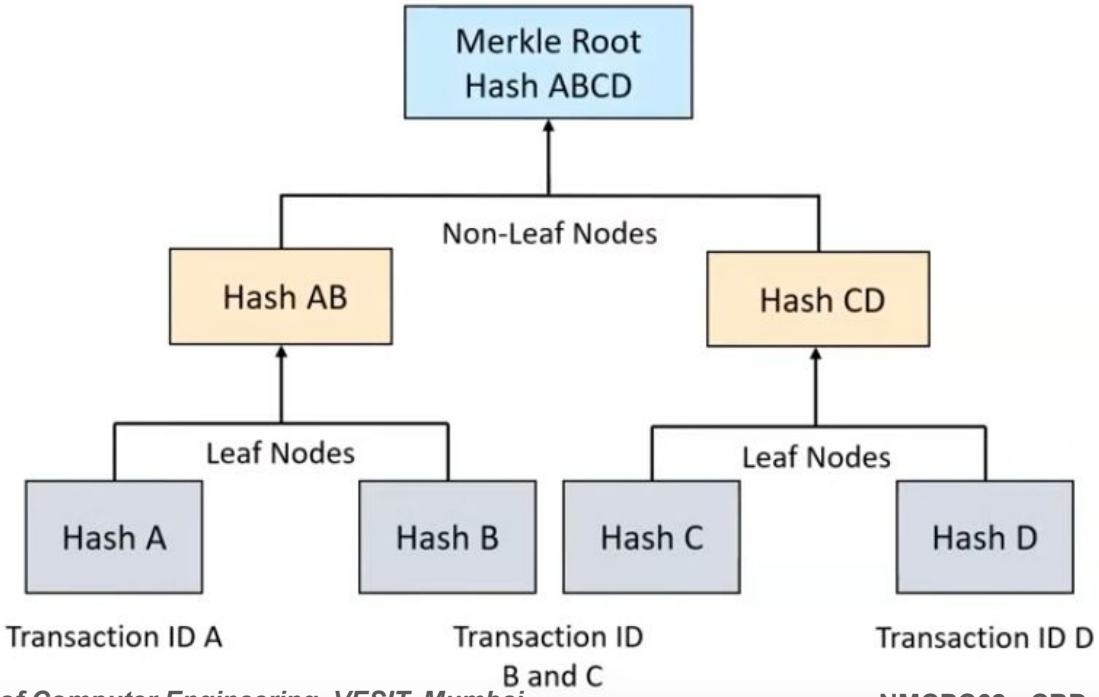
The hashes are paired together resulting in:
Hash AB
and
Hash CD
Our Merkle Root is formed by combining these two hashes: Hash ABCD.



1. Anatomy of a Blockchain - Block (Merkle Tree)

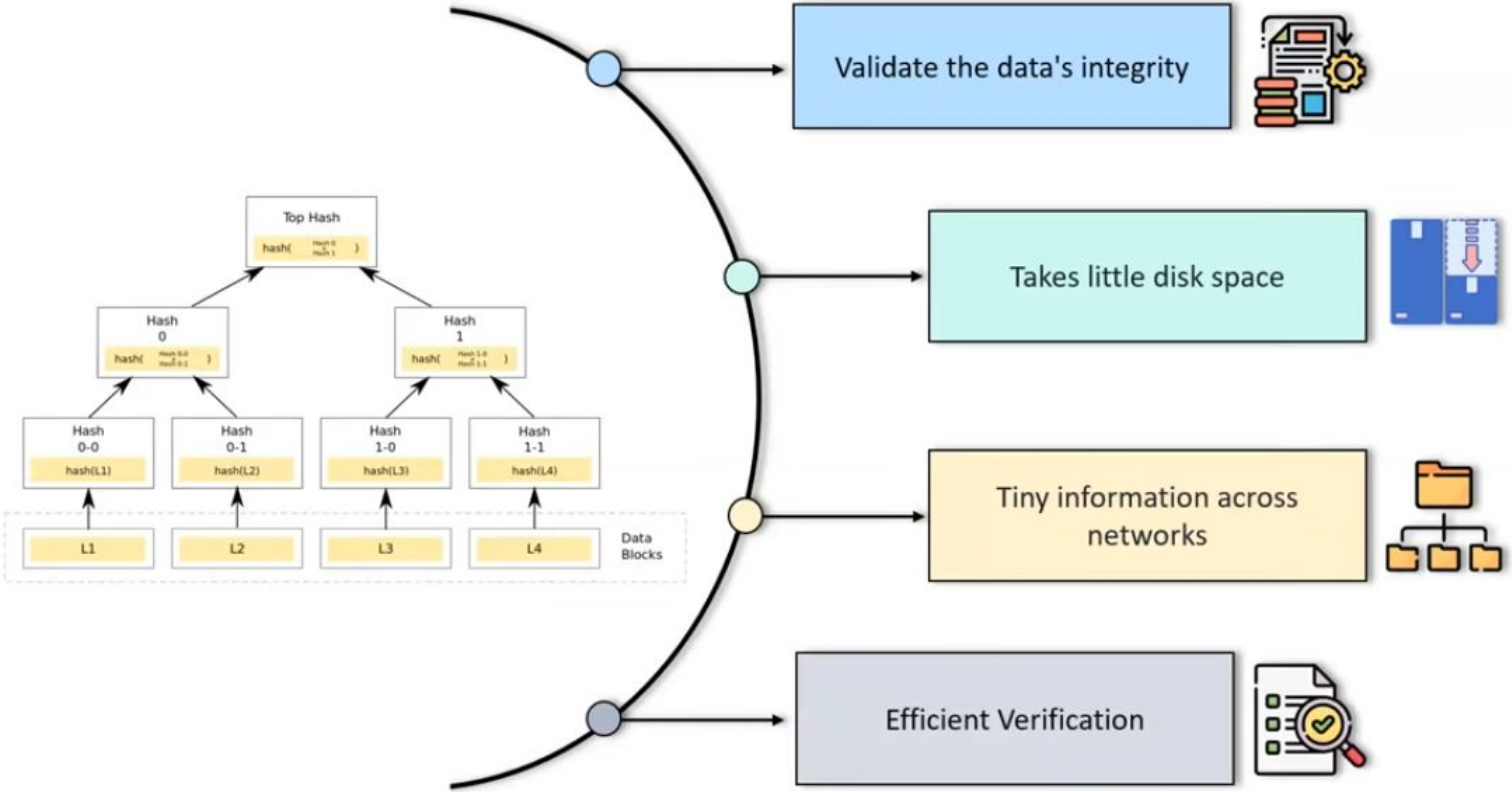
How does a Merkle Tree work ?

The hashes are paired together resulting in:
Hash AB
and
Hash CD
Our Merkle Root is formed by combining these two hashes: Hash ABCD.



1. Anatomy of a Blockchain - Block (Merkle Tree)

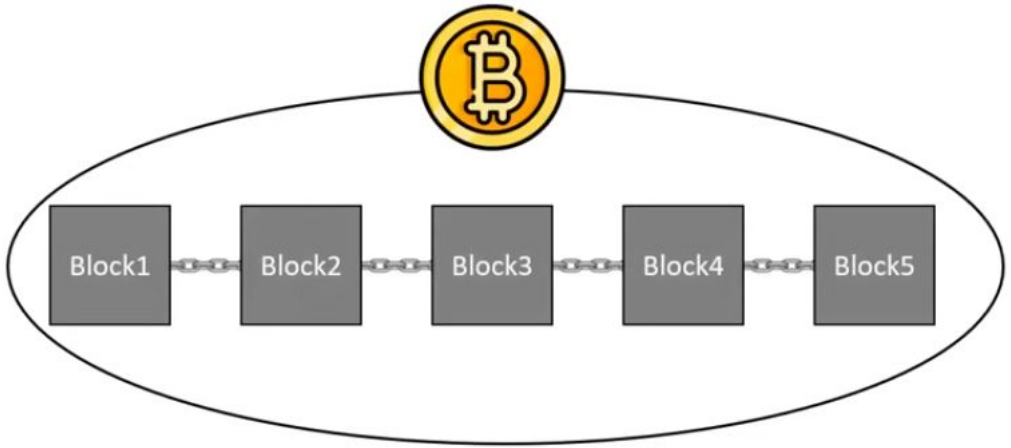
Benefits of Merkle Tree



1. Anatomy of a Blockchain - Block (Merkle Tree)

Why is it essential for Blockchain ?

Let's imagine a blockchain without Merkle Trees to get a sense of how vital they are for blockchain technology.

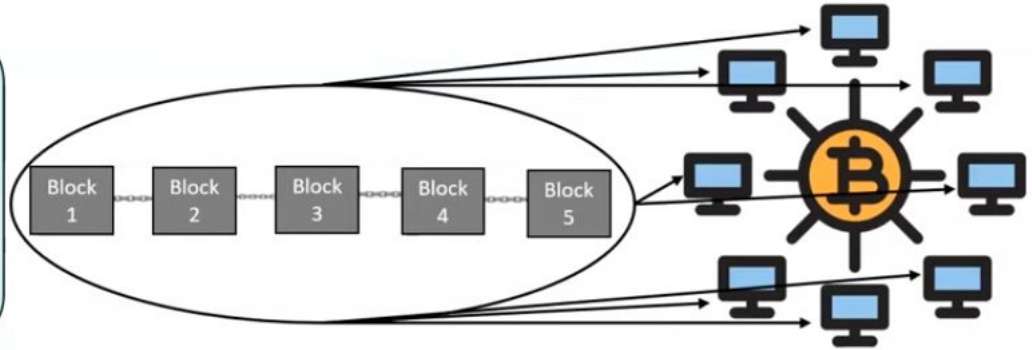


Blocks of Transactions taking place in Bitcoin Network

1. Anatomy of a Blockchain - Block (Merkle Tree)

Why is it essential for Blockchain ?

If Bitcoin didn't include Merkle Trees, for example, every node on the network would have to retain a complete copy of every single Bitcoin transaction ever made.

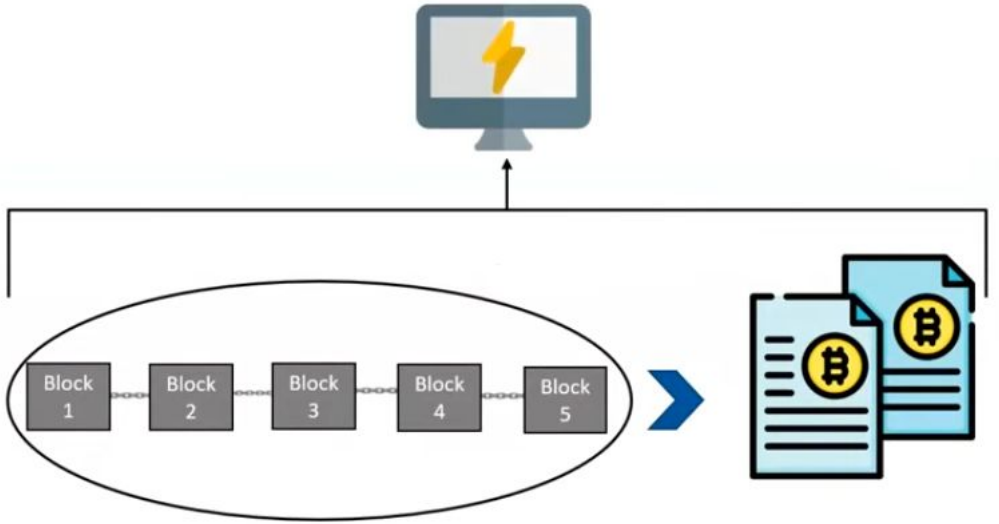


Too much information for every node to validate on their own

1. Anatomy of a Blockchain - Block (Merkle Tree)

Why is it essential for Blockchain ?

To confirm that there were no modifications, a computer used for validation would need a lot of computing power to compare ledgers.

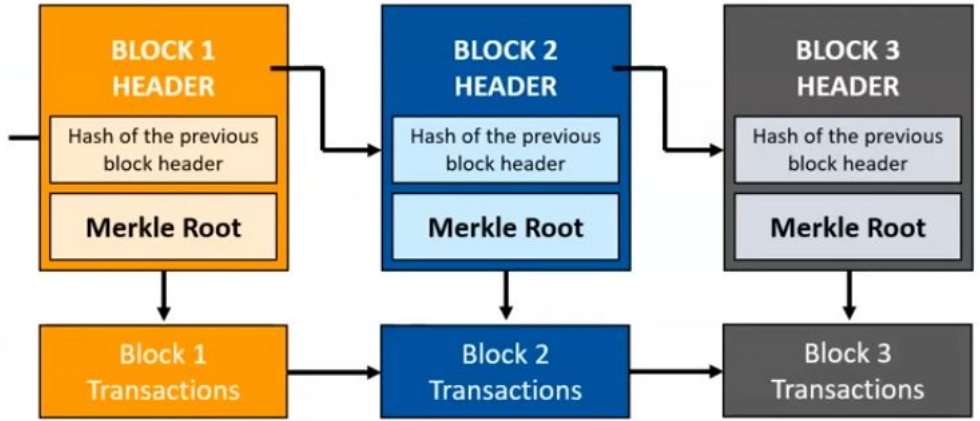


Comparing ledgers using a lot of computing power

1. Anatomy of a Blockchain - Block (Merkle Tree)

Why is it essential for Blockchain ?

Merkle Trees hash records in accounting, thereby separating the proof of data. Proving that given information across the network is all that is required for a transaction to be valid.



Merkle Tree breaking the data into tiny parts of information

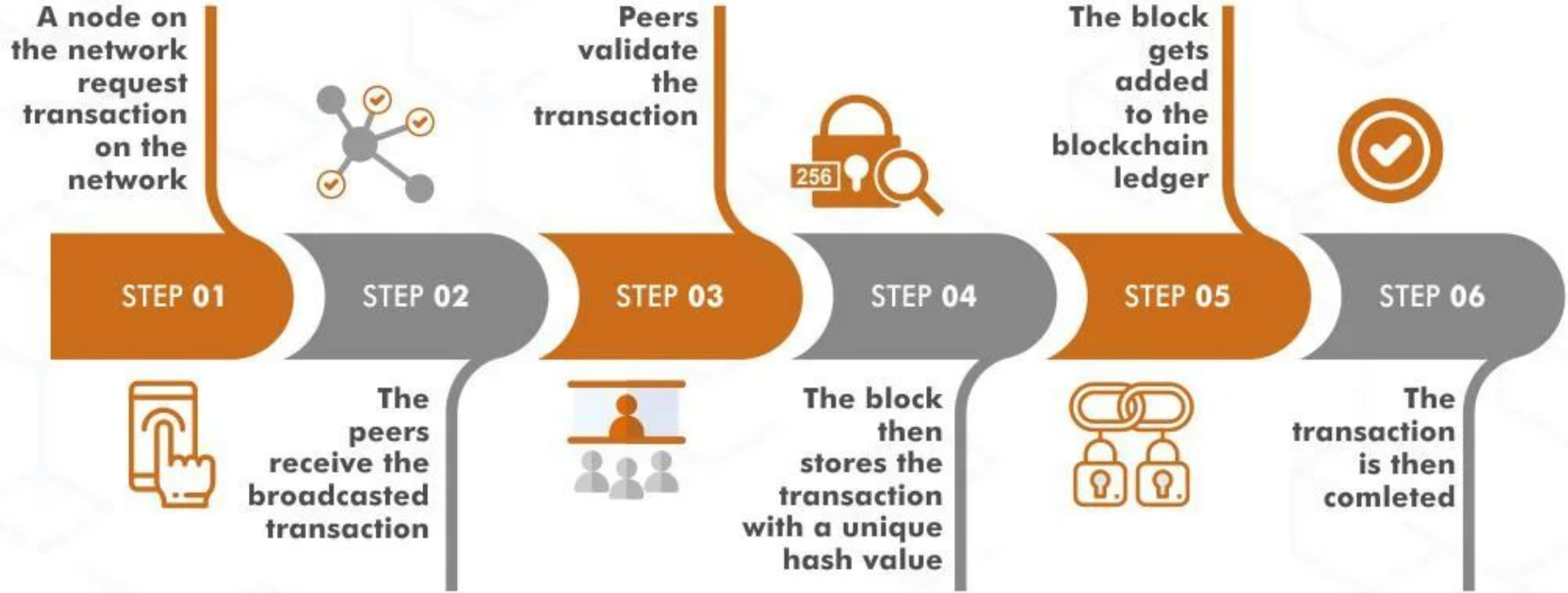
1. Anatomy of a Blockchain - Transactions

- A transaction is a **transfer of value on the blockchain.**
- A transaction is **when one person gives a designated amount of cryptocurrency they own to another person.**
- To perform transactions on the blockchain, you need a crypto wallet.
- **Each wallet is protected by a special cryptographic method** that uses a **unique pair of distinct but connected keys: a private and a public key.**
  - A **public key / blockchain address**, is a **series of letters and numbers that a user must share in order to receive funds.**
  - **Private key** must be **kept secret,** much like your bank card pin number, as it **authorizes the spending of any funds received by the associated public key.**
- **With their wallet, a user** (whoever has the private key) **can authorize or sign transactions** and thereby transfer value to a new owner.
- The transaction is then broadcast to the network to be included in the blockchain.

Courtesy : Ledger.com

1. Anatomy of a Blockchain - Transactions

Transactions in Blockchain - Life Cycle



Courtesy : CBCAmerica

1. Anatomy of a Blockchain - Transactions

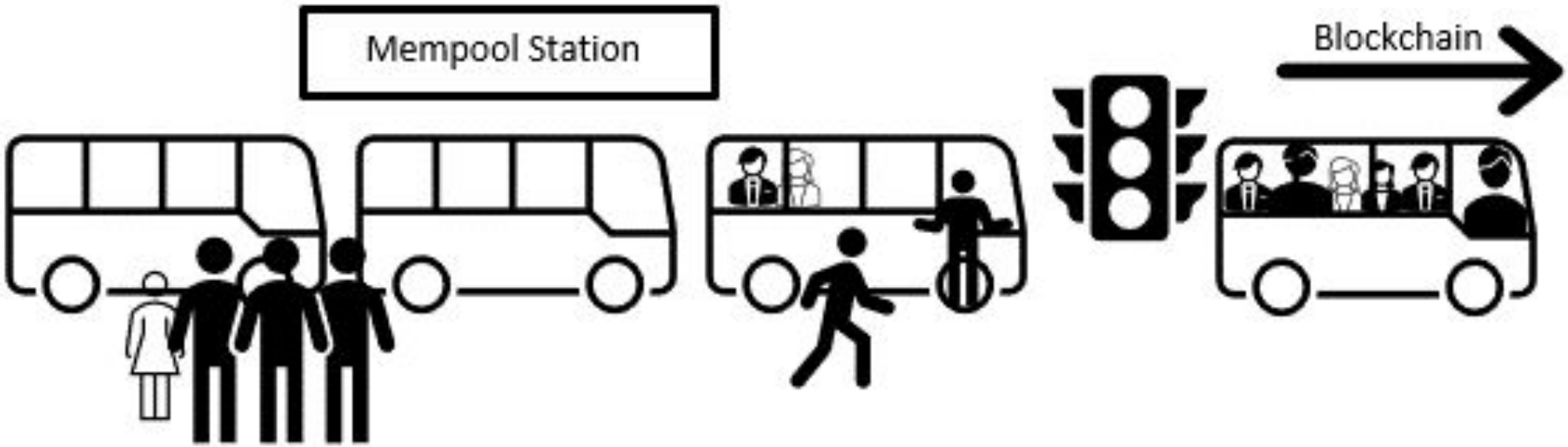
Transactions in Blockchain - Life Cycle

1. Someone requests a transaction. The transaction could involve cryptocurrency, contracts, records, or other information.
2. **Transaction is broadcast to all P2P** participation computers in the specific blockchain network. These are called **Nodes**. All transactions are published to the **Mempool** or memory pool, where they are considered 'pending'. **Gas fees** are paid by users as part of the transaction to compensate for the computing energy required to process and validate transactions on the blockchain.
3. **Miners** verify the transaction. Every computer in the network checks the transaction against some validation rules that are set by the creators of the specific blockchain network.
4. **Validated transactions** are stored into a block and are sealed with a lock referred to as the **Hash**.
5. **New block is added to the existing Blockchain**. This block becomes part of the blockchain when other computers in the network validate if the lock on the block is correct.
6. The transaction is complete. Now the transaction is part of the blockchain and cannot be altered in any way.

Courtesy : Ledger.com

1. Anatomy of a Blockchain - Transactions

Transactions in Blockchain - The Bus Station Analogy



Courtesy : TxStreet

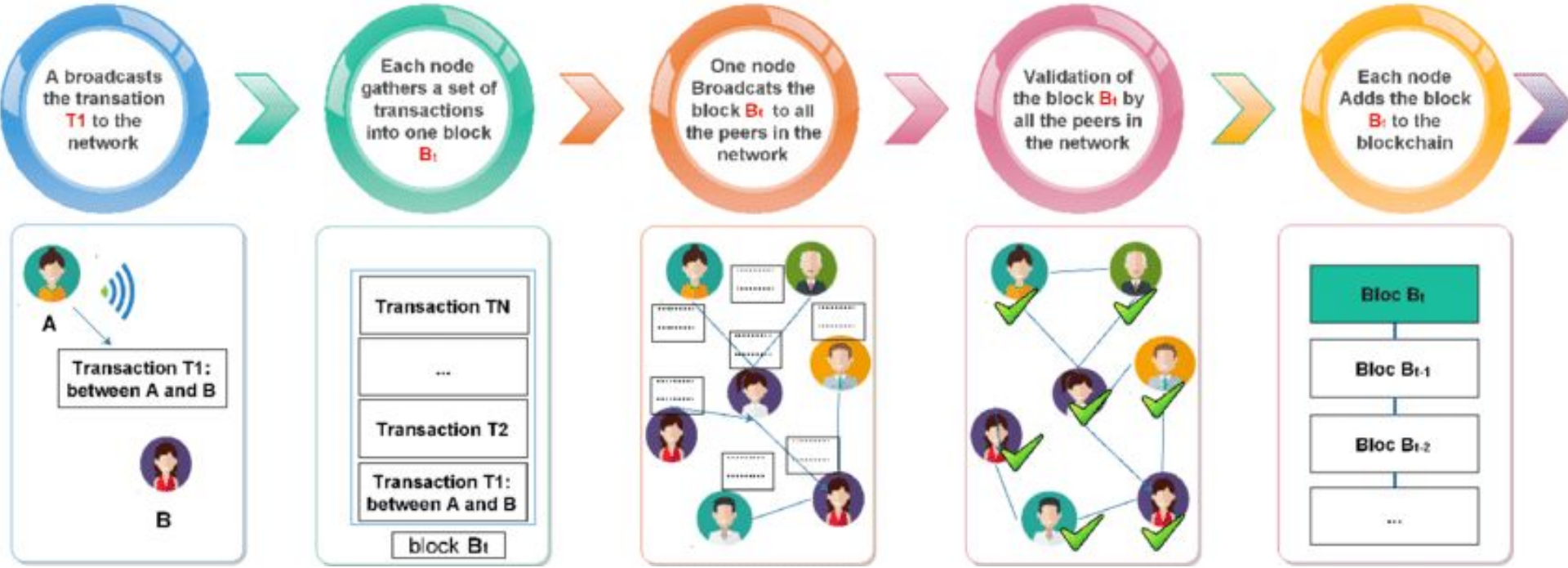
1. Anatomy of a Blockchain - Transactions

Transactions in Blockchain - Live Demo

Courtesy : TxStreet

1. Anatomy of a Blockchain - Transactions

Transactions in Blockchain - Example



Courtesy : Research Gate

1. Anatomy of a Blockchain - Transactions

Transactions in Blockchain - Example contd…

- Alice wants to send two coins to Bob.
- Each transaction has **three main parts**:
  - **The input:** Alice's private coin address, she wants to spend.
  - **The output:** Bob's public key or coin address.
  - **Amounts:** the amount of coins Alice wants to spend.

1. **Alice signs a message with the transaction details using her private key.** The message contains the input, output, and amount to be sent.
2. The **transaction is then broadcast to the network** saying the amount of coins in her account should go down by two. The amount in Bob's account should increase by two.
3. **Each computer in the network will receive the message** and apply the requested transaction to its copy of the ledger, updating the account balances.
4. **Add the transactions into the MemPool.**
5. **Miner** select few transactions from MemPool and tries to **solve Cryptographic Puzzle.**
6. On solving the Puzzle, the **Block is broadcasted to the network for validation**.
7. After adding Block into the Blockchain, **Transactions added in the Block are later removed from MemPool**

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
Hadelin	->	Me	0.3 BTC
Helen	->	Me	0.6 BTC
Susan	->	Me	0.7 BTC

} UTXOs

I want to Buy a bicycle for 0.5 BTC

TRANSACTION:

Input:
0.6 BTC from Helen

Output:
0.5 BTC to the bike shop,

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
Hadelin	->	Me	0.3 BTC
Helen	->	Me	0.6 BTC
Susan	->	Me	0.7 BTC

} UTXOs

I want to Buy a bicycle for 0.5 BTC

TRANSACTION:

Input:
0.6 BTC from Helen

}

Output:
0.5 BTC to the bike shop,
0.1 BTC back to myself

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
Hadelin	->	Me	0.3 BTC
~~Helen~~	~~->~~	~~Me~~	~~0.6 BTC~~
Susan	->	Me	0.7 BTC

} UTXOs

I want to Buy a bicycle for 0.5 BTC

TRANSACTION:

Input:
0.6 BTC from Helen

Output:
0.5 BTC to the bike shop,
0.1 BTC back to myself

UTXO
For the bike shop

UTXO
For me

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
Hadelin	->	Me	0.3 BTC
~~Helen~~	~~->~~	~~Me~~	~~0.6 BTC~~
Susan	->	Me	0.7 BTC

UTXOs

I want to Buy a bicycle for 0.5 BTC

TRANSACTION:

Input:
0.6 BTC from Helen

Output:
0.5 BTC to the bike shop,
0.1 BTC back to myself

UTXO
For the bike shop

UTXO
For me

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark -> Me 0.1 BTC
Hadelin -> Me 0.3 BTC
~~Helen -> Me 0.6 BTC~~
Susan -> Me 0.7 BTC

} UTXOs

I want to Buy a bicycle for 0.5 BTC

TRANSACTION:

Input:
0.6 BTC from Helen

Output:
0.5 BTC to the bike shop,
0.1 BTC back to myself

UTXO
For the bike shop

UTXO
For me

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC	UTXOs
Hadelin	->	Me	0.3 BTC	
Susan	->	Me	0.7 BTC	
Me	->	Me	0.1 BTC	

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC	} UTXOs
Hadelin	->	Me	0.3 BTC	
Susan	->	Me	0.7 BTC	
Me	->	Me	0.1 BTC	

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC	} UTXOs
Hadelin	->	Me	0.3 BTC	
Susan	->	Me	0.7 BTC	
Me	->	Me	0.1 BTC	

I want to Buy a 2nd bicycle for 1.1 BTC

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC	} UTXOs
Hadelin	->	Me	0.3 BTC	
Susan	->	Me	0.7 BTC	
Me	->	Me	0.1 BTC	

I want to Buy a 2nd bicycle for 1.1 BTC

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me }

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
~~Hadelin~~	~~->~~	~~Me~~	~~0.3 BTC~~
Susan	->	Me	0.7 BTC
Me	->	Me	0.1 BTC

} UTXOs

I want to Buy a 2nd bicycle for 1.1 BTC

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me

Output:
1.1 BTC to the bike shop,

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
~~Hadelin~~	~~>~~	~~Me~~	~~0.3 BTC~~
~~Susan~~	~~>~~	~~Me~~	~~0.7 BTC~~
~~Me~~	~~>~~	~~Me~~	~~0.1 BTC~~

} UTXOs

I want to Buy a 2nd bicycle for 1.1 BTC

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me

Output:
1.1 BTC to the bike shop,

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
~~Hadelin~~	~~>~~	~~Me~~	~~0.3 BTC~~
~~Susan~~	~~>~~	~~Me~~	~~0.7 BTC~~
~~Me~~	~~>~~	~~Me~~	~~0.1 BTC~~

} UTXOs

I want to Buy a 2nd bicycle for 1.1 BTC

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me

Output:
1.1 BTC to the bike shop,

UTXO
For the bike shop

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
~~Hadelin~~	~~>~~	~~Me~~	~~0.3 BTC~~
~~Susan~~	~~>~~	~~Me~~	~~0.7 BTC~~
~~Me~~	~~>~~	~~Me~~	~~0.1 BTC~~

} UTXOs

I want to Buy a 2nd bicycle for 1.1 BTC

UTXO
For the bike shop

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me

} Output:
1.1 BTC to the bike shop,

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark	->	Me	0.1 BTC
~~Hadelin~~	~~>~~	~~Me~~	~~0.3 BTC~~
~~Susan~~	~~>~~	~~Me~~	~~0.7 BTC~~
~~Me~~	~~>~~	~~Me~~	~~0.1 BTC~~

} UTXOs

I want to Buy a 2nd bicycle for 1.1 BTC

UTXO
For the bike shop

TRANSACTION:

Input:
0.3 BTC from Hadelin,
0.7 BTC from Susan,
0.1 BTC from Me

} Output:
1.1 BTC to the bike shop,

1. Anatomy of a Blockchain - Transactions

Transactions & UTXOs

Mark -> Me 0.1 BTC } UTXOs

1. Anatomy of a Blockchain - Transactions

Where do transaction fees come from?

Mark	->	Me	0.1 BTC
Sarah	->	Me	0.1 BTC
Hadelin	->	Me	0.4 BTC
Ebay	->	Me	0.3 BTC
Hadelin	->	Me	0.3 BTC

} UTXOs

1. Anatomy of a Blockchain - Transactions

Where do transaction fees come from?

Mark	->	Me	0.1 BTC
Sarah	->	Me	0.1 BTC
Hadelin	->	Me	0.4 BTC
Ebay	->	Me	0.3 BTC
Hadelin	->	Me	0.3 BTC

} UTXOs

I want to Buy a 3rd bicycle for 0.9 BTC and an apple for 0.02 BTC

TRANSACTION:

Input:
- 0.4 BTC from Hadelin,
- 0.3 BTC from Ebay
- 0.3 BTC from Hadelin

Output:
- 0.9 BTC to the bike shop,
- 0.02 BTC to the fruit shop,
- 0.06 BTC to myself

1. Anatomy of a Blockchain - Transactions

Where do transaction fees come from?

Mark	->	Me	0.1 BTC
Sarah	->	Me	0.1 BTC
~~Hadelin~~	~~->~~	~~Me~~	~~0.4 BTC~~
~~Ebay~~	~~->~~	~~Me~~	~~0.3 BTC~~
~~Hadelin~~	~~->~~	~~Me~~	~~0.3 BTC~~

UTXOs

I want to Buy a 3rd bicycle for 0.9 BTC and an apple for 0.02 BTC

TRANSACTION:

Input:
- 0.4 BTC from Hadelin,
- 0.3 BTC from Ebay
- 0.3 BTC from Hadelin

Output:
- 0.9 BTC to the bike shop,
- 0.02 BTC to the fruit shop,
- 0.06 BTC to myself

UTXO for the bike shop

UTXO for the bike shop

UTXO for me

1. Anatomy of a Blockchain - Transactions

Where do transaction fees come from?

Mark	->	Me	0.1 BTC
Sarah	->	Me	0.1 BTC
~~Hadelin~~	~~->~~	~~Me~~	~~0.4 BTC~~
~~Ebay~~	~~->~~	~~Me~~	~~0.3 BTC~~
~~Hadelin~~	~~->~~	~~Me~~	~~0.3 BTC~~

} UTXOs

I want to Buy a 3rd bicycle for 0.9 BTC and an apple for 0.02 BTC

TRANSACTION:

Input:
- 0.4 BTC from Hadelin,
- 0.3 BTC from Ebay
- 0.3 BTC from Hadelin

Output:
- 0.9 BTC to the bike shop, ← UTXO for the bike shop
- 0.02 BTC to the fruit shop, ← UTXO for the bike shop
- 0.06 BTC to myself ← UTXO for me

Fees: 0.02 BTC ← UTXO for the miner

1. Anatomy of a Blockchain - Transactions

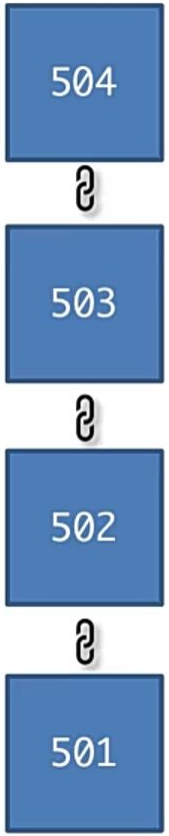
Where do transaction fees come from?

Mark	->	Me	0.1 BTC
Sarah	->	Me	0.1 BTC
Me	->	Me	0.0.6 BTC

} UTXOs

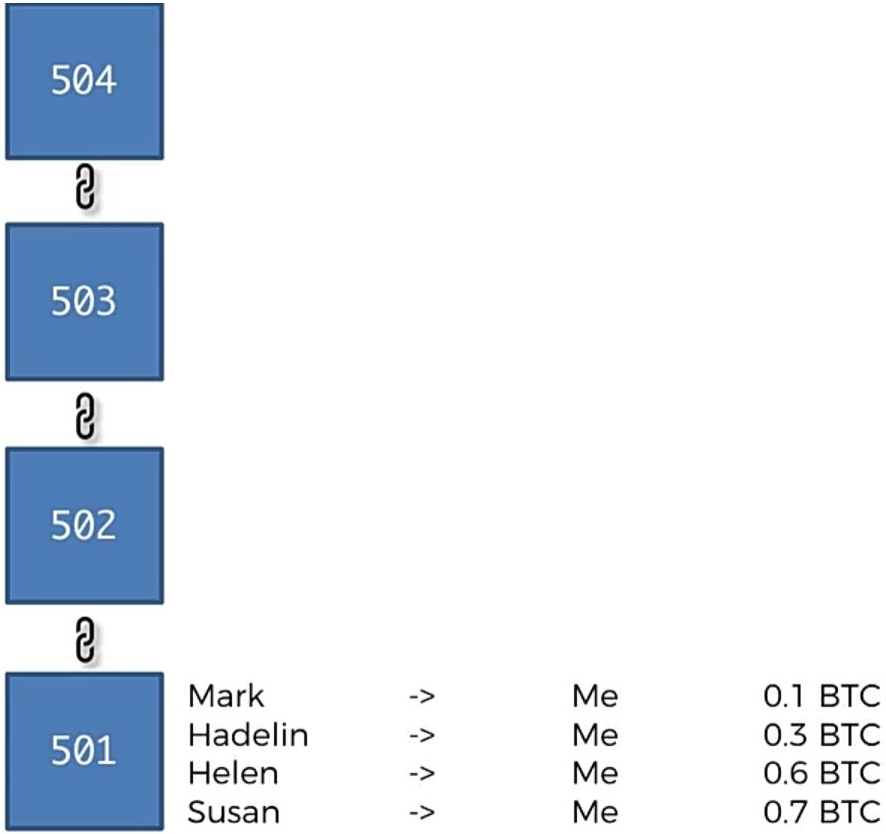
1. Anatomy of a Blockchain - Transactions

How wallets work?



1. Anatomy of a Blockchain - Transactions

How wallets work?



1. Anatomy of a Blockchain - Transactions

How wallets work?

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

Courtesy : Super Data Science

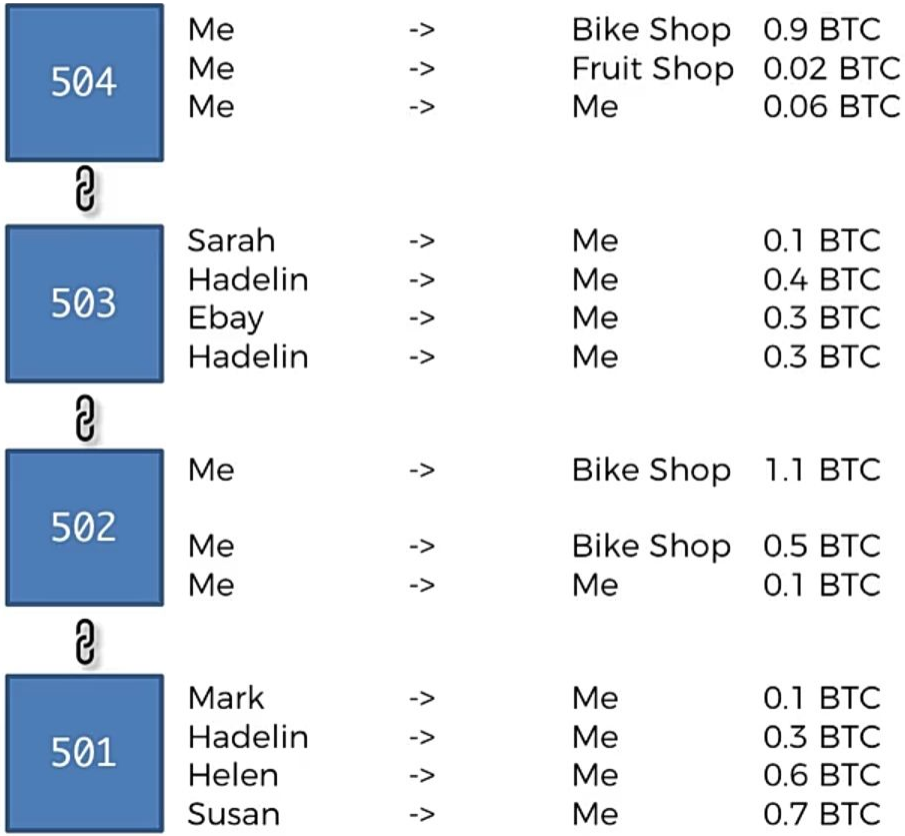
1. Anatomy of a Blockchain - Transactions

How wallets work?

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

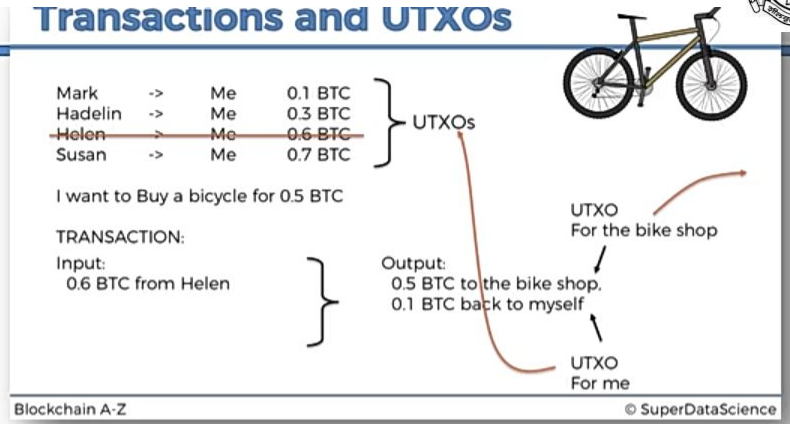


Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

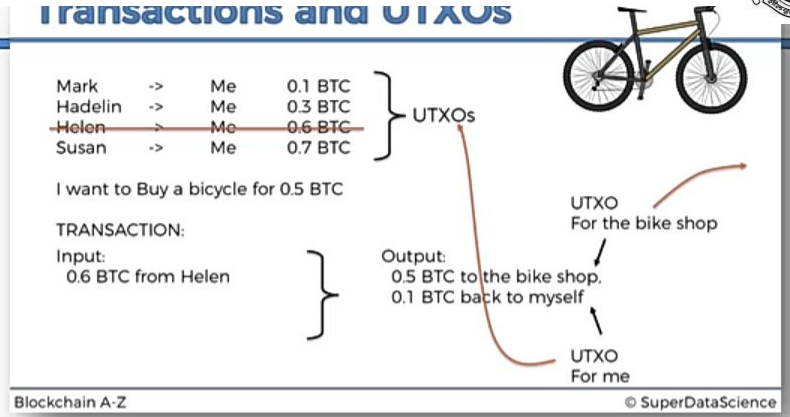


Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

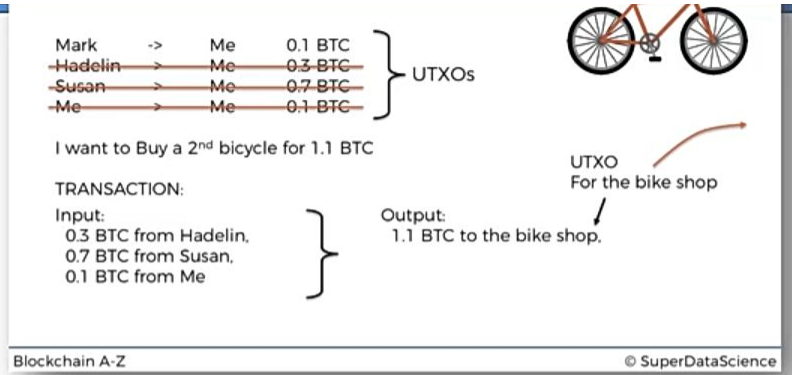


Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

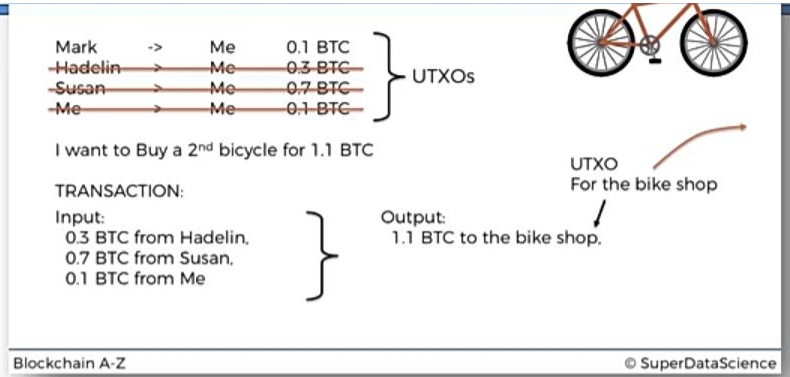


Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

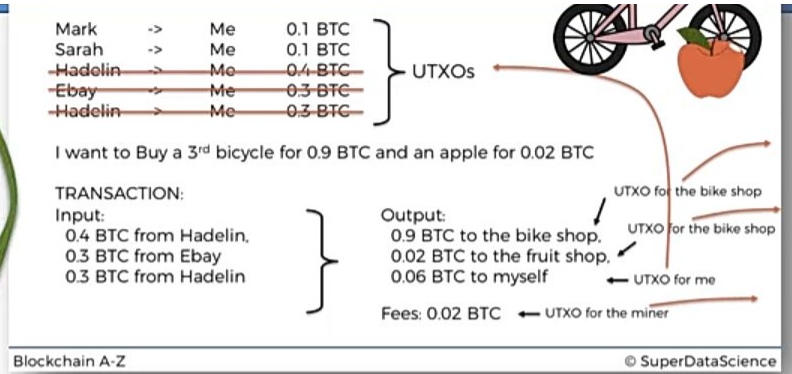


Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?

504	Me	->	Bike Shop	0.9 BTC
	Me	->	Fruit Shop	0.02 BTC
	Me	->	Me	0.06 BTC UTXO
503	Sarah	->	Me	0.1 BTC
	Hadelin	->	Me	0.4 BTC
	Ebay	->	Me	0.3 BTC
	Hadelin	->	Me	0.3 BTC
502	Me	->	Bike Shop	1.1 BTC
	Me	->	Bike Shop	0.5 BTC
	Me	->	Me	0.1 BTC
501	Mark	->	Me	0.1 BTC
	Hadelin	->	Me	0.3 BTC
	Helen	->	Me	0.6 BTC
	Susan	->	Me	0.7 BTC

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

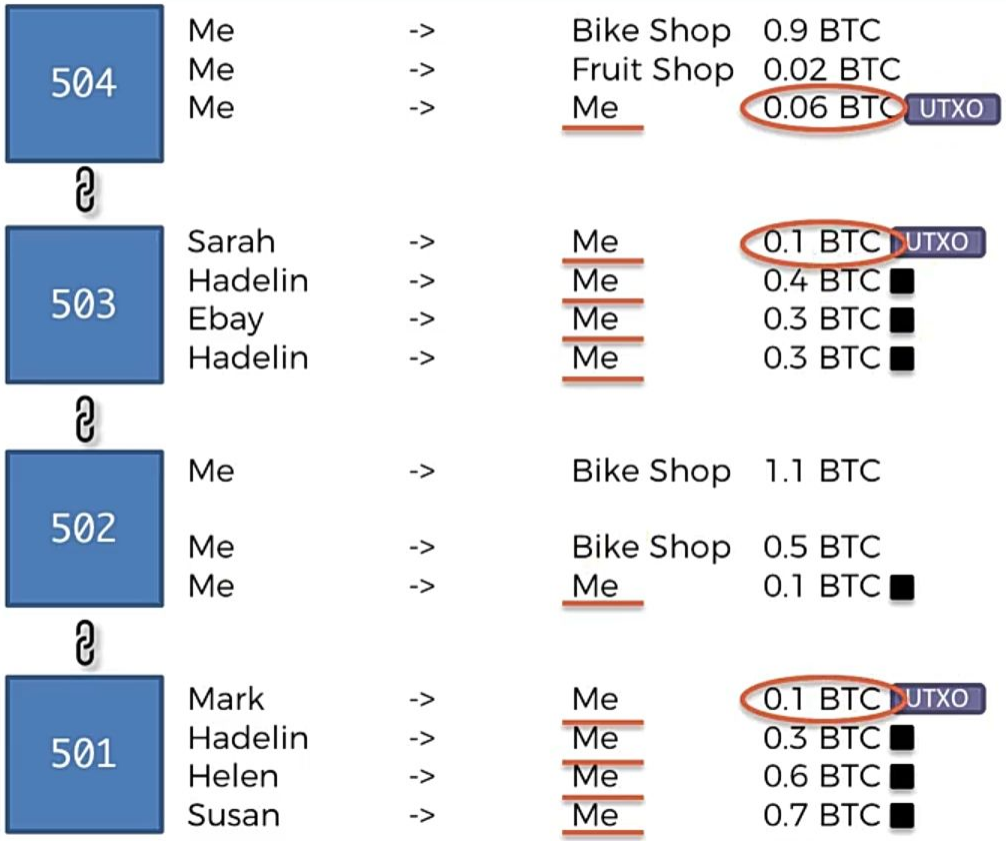
How wallets work?

504	Me	->	Bike Shop	0.9 BTC	
	Me	->	Fruit Shop	0.02 BTC	
	Me	->	Me	0.06 BTC	UTXO
503	Sarah	->	Me	0.1 BTC	UTXO
	Hadelin	->	Me	0.4 BTC	
	Ebay	->	Me	0.3 BTC	
	Hadelin	->	Me	0.3 BTC	
502	Me	->	Bike Shop	1.1 BTC	
	Me	->	Bike Shop	0.5 BTC	
	Me	->	Me	0.1 BTC	
501	Mark	->	Me	0.1 BTC	UTXO
	Hadelin	->	Me	0.3 BTC	
	Helen	->	Me	0.6 BTC	
	Susan	->	Me	0.7 BTC	

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How wallets work?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

UTXO - Unspent transaction output

- The fundamental building block of a bitcoin transaction
- Transaction outputs are **indivisible chunks of bitcoin currency,** recorded on the blockchain, and recognized as valid by the entire network.
- Bitcoin full nodes track all available and spendable outputs
- **UTXO set** - The collection of all UTXO
  - set grows as new UTXO is created
  - shrinks when UTXO is consumed.
- Every transaction represents a change (state transition) in the UTXO set.
- Eg:User receiving 1 BTC → wallet detected a UTXO that can be spent by any key in the wallet
- **Users Bitcoin balance = Sum of all UTXO** in the users wallet can detect in the network.
  1. scanning the Blockchain
  2. aggregating the **value of UTXO that wallet can spent using the keys** it stores.

1. Anatomy of a Blockchain - Transactions

UTXO - Unspent transaction output

- Have an arbitrary value denominated as a multiple of satoshis
- 1 BTC can be divided into 8 decimal places of satoshi.
- Can only be consumed in its entirety by a transaction.

Wallet Strategies to satisfy a purchase amount (done by the user wallet automatically)

- Combine several smaller units
- Finding exact change
- Using a single UTXO larger than the transaction value.

Coinbase

- Special type of transaction appears as the 1st transaction in each block.
- Placed / created by the miner as a reward for mining (**Bitcoin Money Supply is created)**
- Does not consume UTXO

1. Anatomy of a Blockchain - Transactions

UTXO - Unspent transaction output

- Every bitcoin transaction creates outputs, which are recorded on the bitcoin ledger.
- **Transaction outputs consist of two parts:**
  1. An **amount of bitcoin**, denominated in satoshis, the smallest bitcoin unit
  2. A **cryptographic puzzle** that determines the conditions required to spend the output
     - **locking script / witness script / scriptPubKey / P2PKH (Pay to Public Key Hash)**

Transaction Serialization

- Process of converting the internal representation of a data structure such that it can be transmitted one byte at a time

Transaction Deserialization / Parsing

- Process of converting byte stream representation of a transaction to internal representation of a data structure

1. Anatomy of a Blockchain - Transactions

Transaction inputs

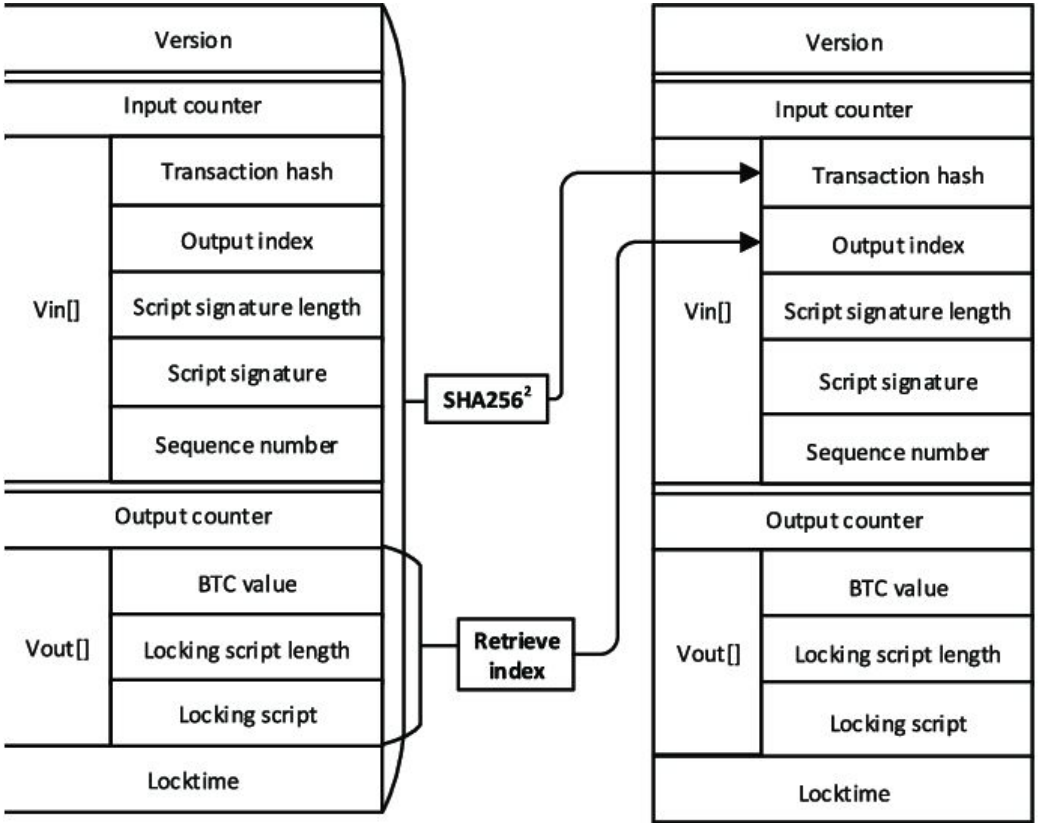
- identify (by reference) which UTXO will be consumed
- provide proof of ownership through an **unlocking script / Digital Signature**

Transaction input contains four elements:

1. A transaction ID, referencing the transaction that contains the UTXO being spent
2. An **output index** (vout), identifying which UTXO from that transaction is referenced (first one is zero)
3. A **scriptSig,** which satisfies the conditions placed on the UTXO, unlocking it for spending
4. A sequence number

Note : Once a transaction is broadcasted, *every validating node needs to retrieve the UTXO reference in the transaction inputs in order to validate the transaction.*

1. Anatomy of a Blockchain - Transactions Structure



Courtesy : Research Gate
Oreilly

1. Anatomy of a Blockchain - Transactions

Transaction Fees

- Incentives given to the miner for mining blocks
- Disincentive against abuse of the system by imposing a small cost on every transaction
- Encourages processing priority
- ***Calculated*** :
  - **Initially,** based on the siae of the transaction in KB (not on the value of the transaction)
  - **Now**, based on network capacity & transaction volume.
- **MinRelayTxFee** (Bitcoin) : Default : **0.00001 BTC**
- If **TxFee < MinRelayFee**

  ⇒ Transaction is free

  ⇒ Relayed only if there is space in mempool / Dropped
- Transaction can have different levels of Priority based on TxFees
  - High ⇒ user pays high TxFees
  - Medium or Low ⇒ user pays low TxFees

1. Anatomy of a Blockchain - Transactions

Adding Transaction Fees to Transactions

- No field in Fees in the Transaction Structure
- TxFees = Excess amount that remains after all outputs have been deducted from all inputs
  - **TxFees = Sum(Inputs) - Sum(Outputs)**
- Ideal TxFees to ensure that transactions get confirmed and verified
  - **TxFees = size of Transaction * per KB fees**

Scenario - 1 : Alice has 0.2 BTC

- 0.015 BTC ⇒ UTXO to Bob
- 0.001 BTC ⇒ UTXO to Miner (as TxFees)
- 0.184 BTC ⇒ UTXO back to Alice

Scenario - 2 : Eugenia raising funds for Children's Charity (purpose purchase school books)

- collected 50 BTC as thousands of UTXO as TxInputs
- Pay the purchaser as one UTXO
- Pay higher TxFees as there are many small TxInputs so that transaction is processed promptly.

1. Anatomy of a Blockchain - Transactions Live Bitcoin Block

Bitcoin Block 800,041

Mined on July 24, 2023 03:17:23 • All Blocks

Unknown

Coinbase Message • ▓,H>d/Foundry USA Pool #dropgold/IRq ▓▓▓▓▓▓▓▓▓

A total of 234.86 BTC ($6,978,185) were sent in the block with the average transaction being 0.0394 BTC ($1,170.67). Unknown earned a total reward of 6.25 BTC $185,703. The reward consisted of a base reward of 6.25 BTC $185,703 with an additional 0.0696 BTC ($2,067.99) reward paid as fees of the 5,967 transactions which were included in the block.

Details

Hash	00000-6bb64	Depth	1
Capacity	191.28%	Size	2,005,719
Distance	13m 21s	Version	0×20800000
BTC	234.8568	Merkle Root	79-6a
Value	$6,978,185	Difficulty	53,911,173,001,054.59
Value Today	$6,858,763	Nonce	1,134,080,607
Average Value	0.0393592782 BTC	Bits	386,218,132
Median Value	0.00000330 BTC	Weight	3,992,856 WU
Input Value	234.93 BTC	Minted	6.25 BTC
Output Value	241.18 BTC	Reward	6.31961394 BTC
Transactions	5,967	Mined on	24 Jul 2023, 15:17:23
Witness Tx's	5,915	Height	800,041
Inputs	6,274	Confirmations	1

Coinbase Txn

Last | First | ↗ Value | ↘ Value | ↗ Fee | ↘ Fee

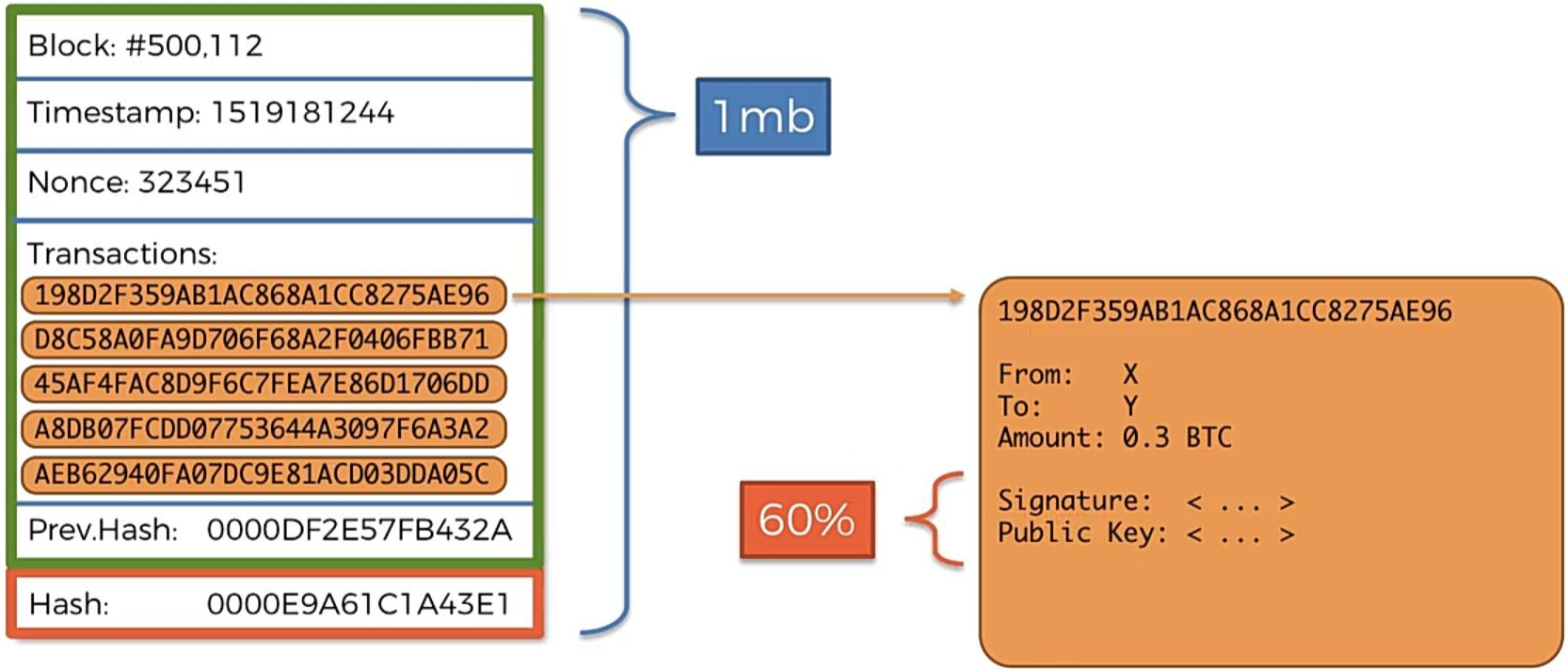
Coinbase

0	ID: ac16-528a 7/24/2023, 15:17:23	From Block Reward To 2 Outputs	6.31961394 BTC • $187,771 Fee 0 Sats • $0.00
TX	1 ID: 7528-6e68 7/24/2023, 15:16:15	From bc1q-pemf To 2 Outputs	5.83754433 BTC • $173,448 Fee 36.0K Sats • $10.70
TX	2 ID: f2e7-89d2 7/24/2023, 15:17:18	From bc1q-pemf To 3 Outputs	9.05621531 BTC • $269,082 Fee 44.5K Sats • $13.22
TX	3 ID: 13e5-a4fa 7/24/2023, 15:17:04	From bc1q-pemf To 3 Outputs	5.82139949 BTC • $172,968 Fee 44.5K Sats • $13.22
TX	4 ID: 6480-13dd 7/24/2023, 15:17:10	From 12oz-cNEA To 1MU3-m7aM	0.04040926 BTC • $1,200.66 Fee 25.8K Sats • $7.66
TX	5 ID: d69d-48fe 7/24/2023, 15:17:23	From bc1q-n6eg To bc1q-6ctp	0.00826518 BTC • $245.58 Fee 12.9K Sats • $3.83
TX	6 ID: dd90-ae7f 7/24/2023, 15:17:18	From bc1q-2pzk To bc1q-yqaj	0.00630000 BTC • $187.19 Fee 10.0K Sats • $2.97
TX	7 ID: e519-7c53 7/24/2023, 15:16:26	From bc1q-6e3z To 2 Outputs	4.09893700 BTC • $121,789 Fee 12.1K Sats • $3.60

Courtesy : Blockchain.com

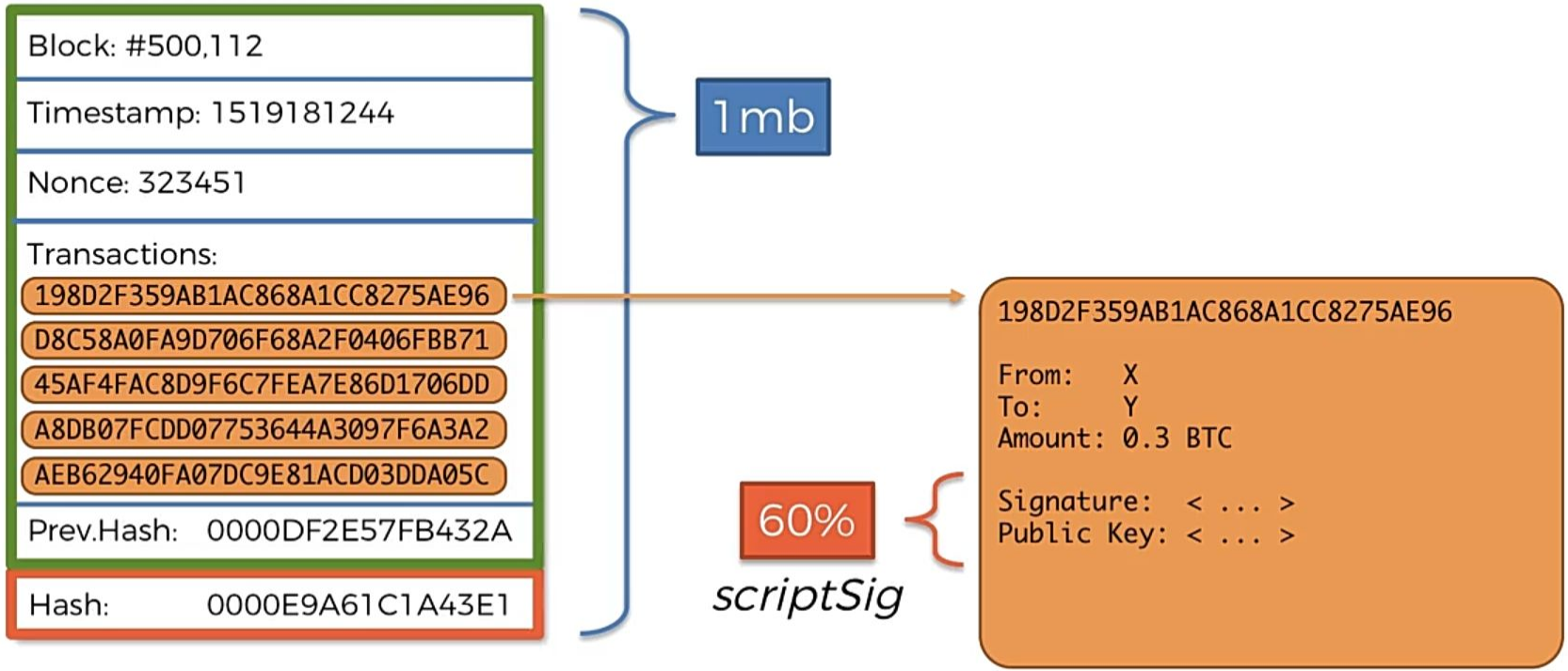
1. Anatomy of a Blockchain - Transactions

What is Segregated Witness ? (SegWit)



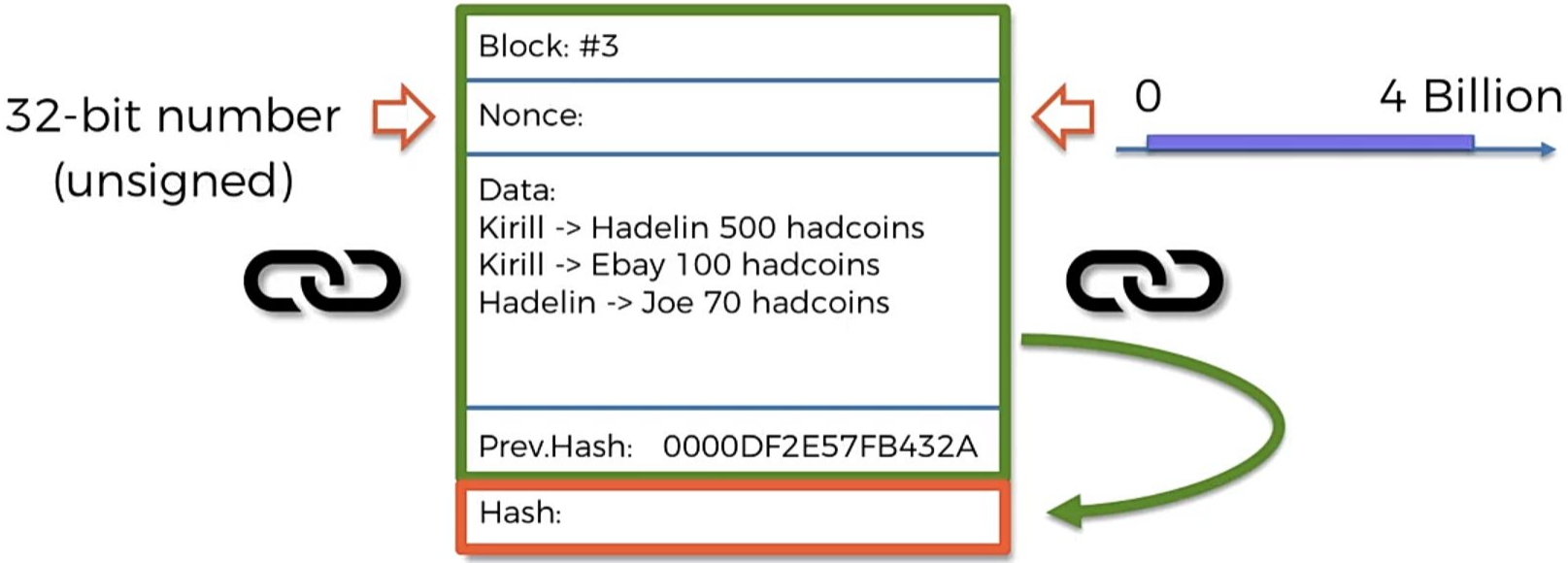
1. Anatomy of a Blockchain - Transactions

What is Segregated Witness ? (SegWit)



1. Anatomy of a Blockchain - Transactions

Nonce Range



1. Anatomy of a Blockchain - Transactions

Nonce Range

Let's do some estimations:

Difficulty:

Total possible 64-digit hexadecimal numbers: 16 x 16 x …. x 16 = $16^{64} \approx 10^{77}$

Total valid hashes (with 18 leading zeros): 16 x 16 x … x 16 = $16^{64-18} \approx 2 \times 10^{55}$

Probability that a Randomly picked hash is valid: $2 \times 10^{55} / 10^{77} = 2 \times 10^{-22} = 0.00000000000000000002\%$

Nonce:

The Nonce is a 32-bit number, the Max Nonce = $2^{32} = 4{,}294{,}967{,}296 = 4 \times 10^{9}$

Assuming no collisions, this means 4×10^{9} different hashes

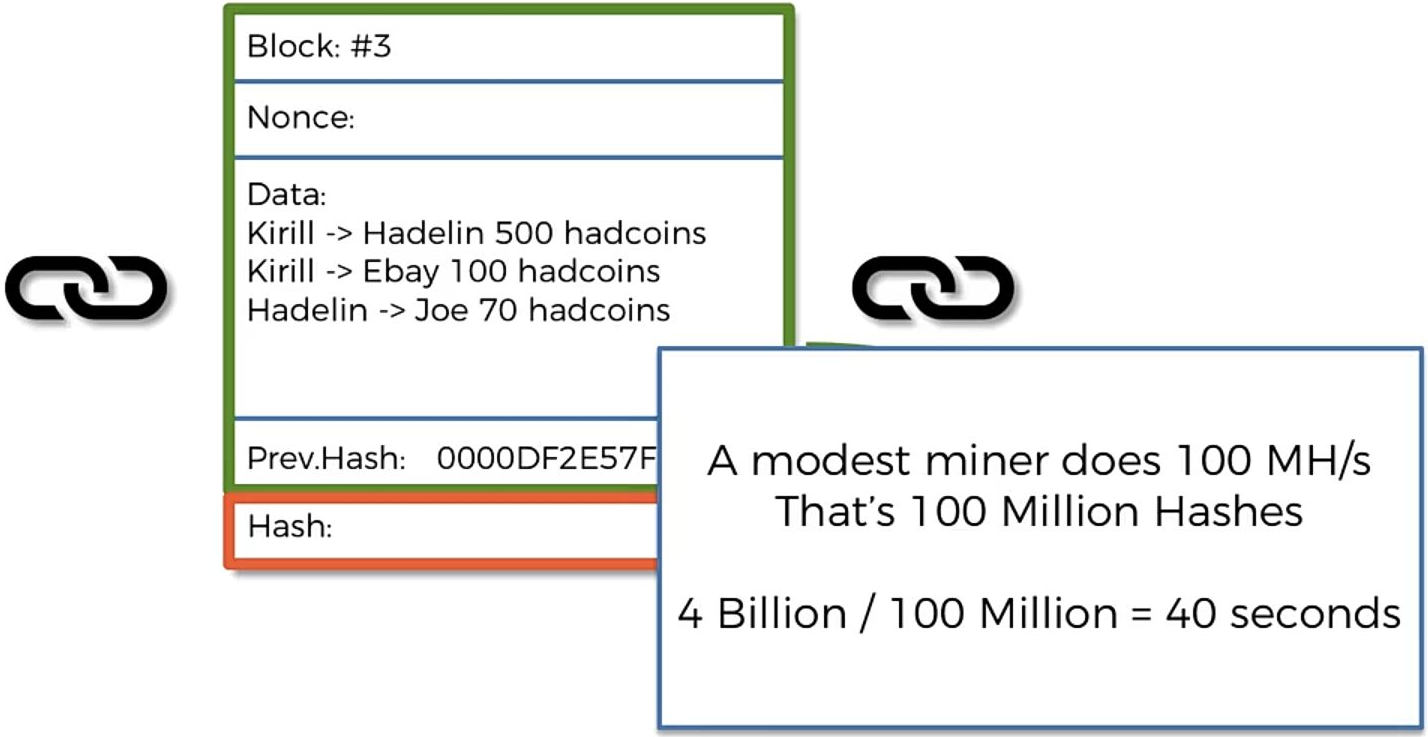
Probability that ONE of them will be valid: $4 \times 10^{9} \times 2 \times 10^{-22} = 8 \times 10^{-13} \approx 10^{-12} = 0.0000000001\%$

Conclusion: One Nonce Range is not enough

Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

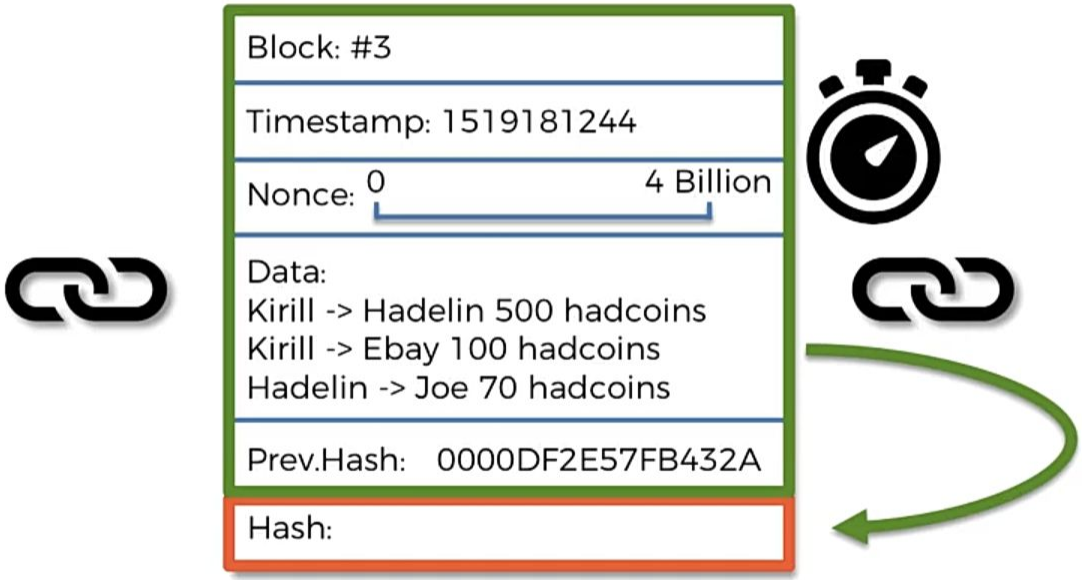
Nonce Range



Courtesy : Super Data Science

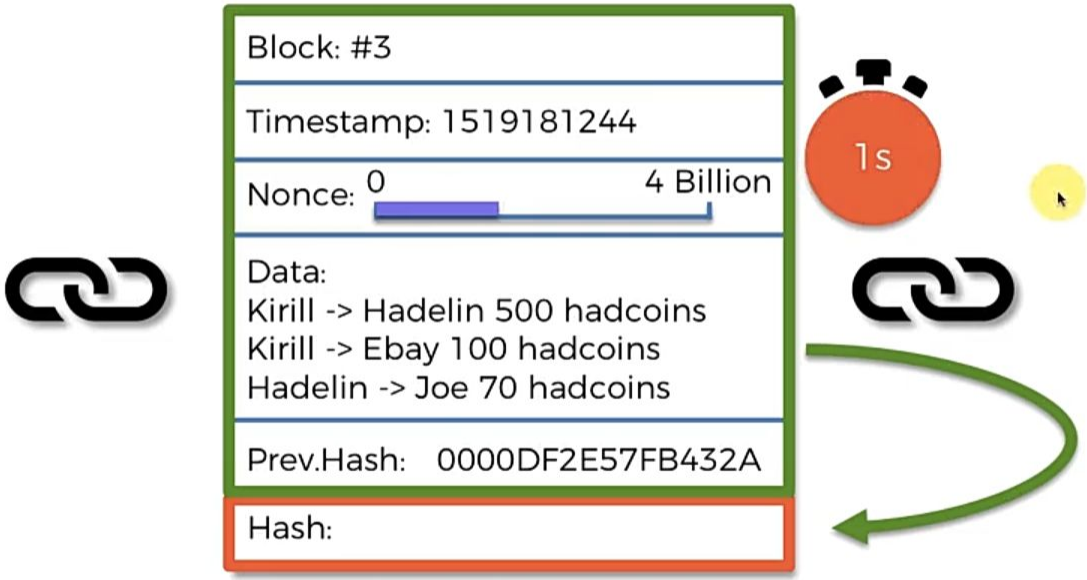
1. Anatomy of a Blockchain - Transactions

Nonce Range



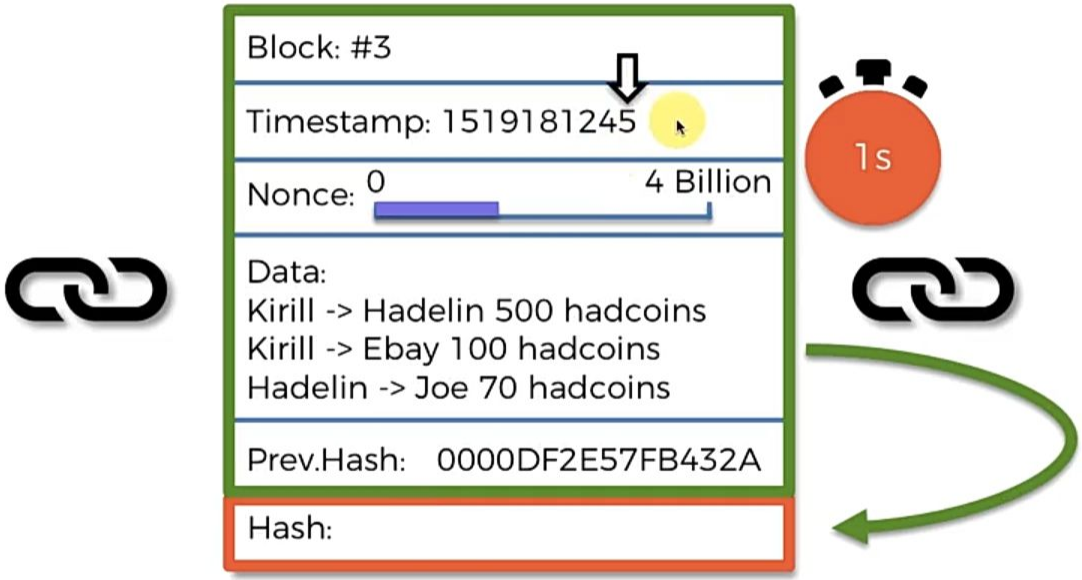
1. Anatomy of a Blockchain - Transactions

Nonce Range



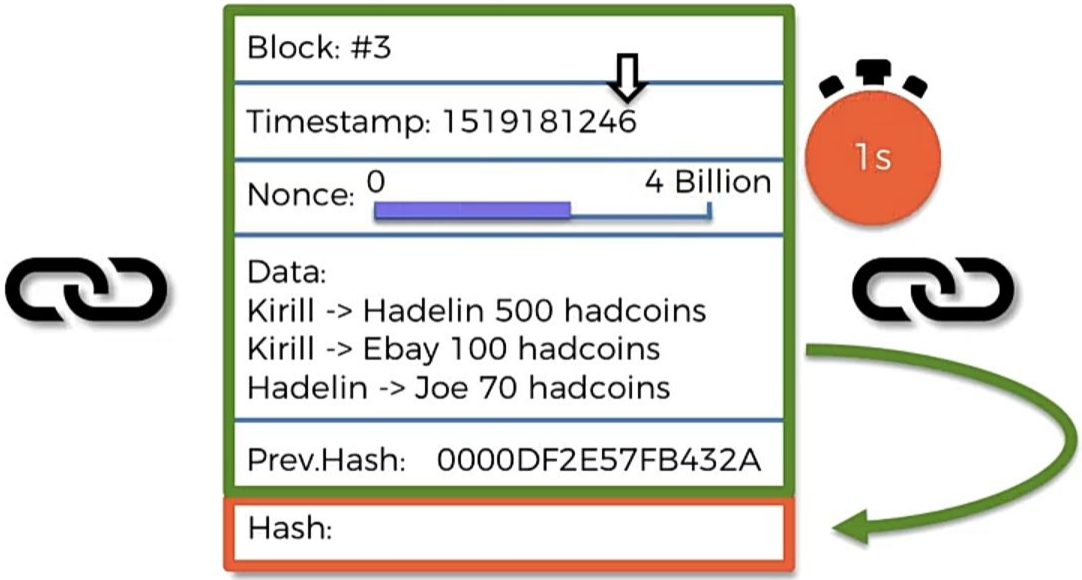
1. Anatomy of a Blockchain - Transactions

Nonce Range



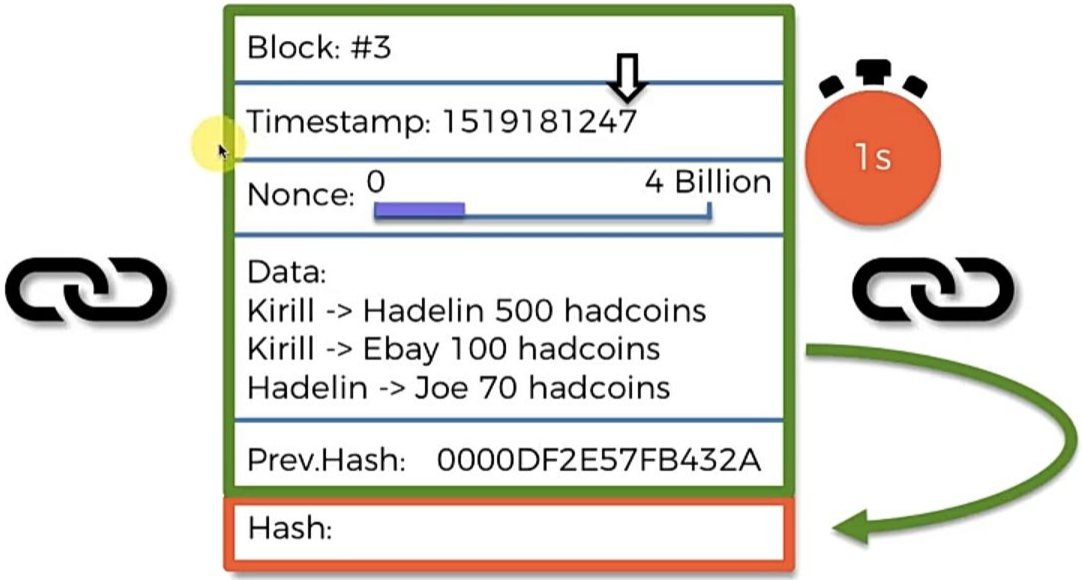
1. Anatomy of a Blockchain - Transactions

Nonce Range



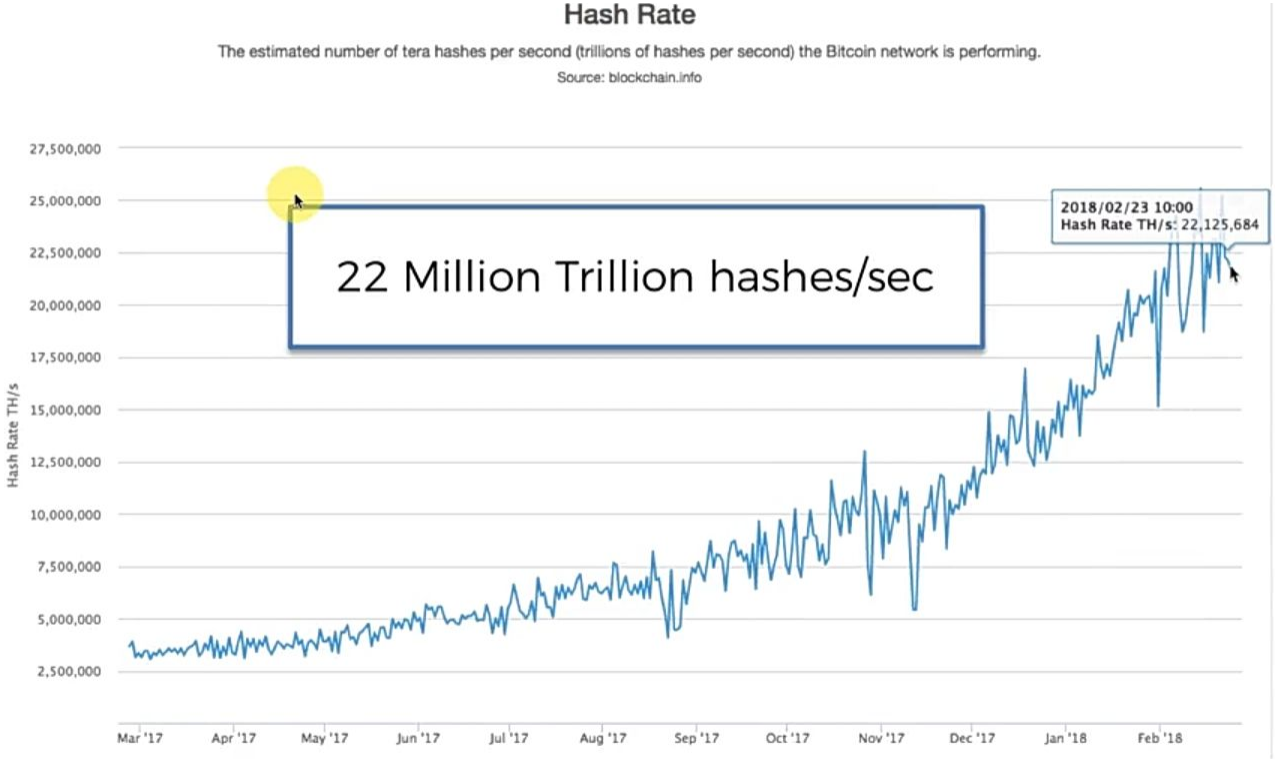
1. Anatomy of a Blockchain - Transactions

Nonce Range



1. Anatomy of a Blockchain - Transactions

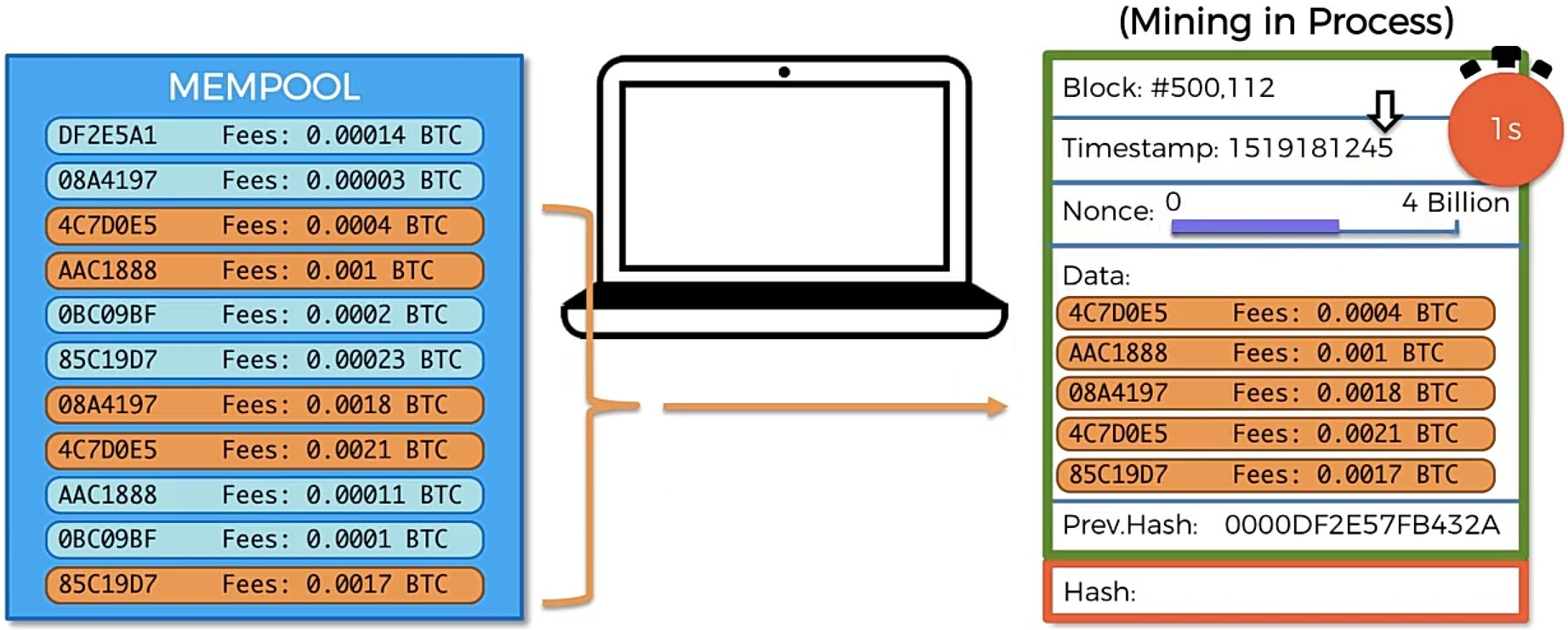
Nonce Range



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

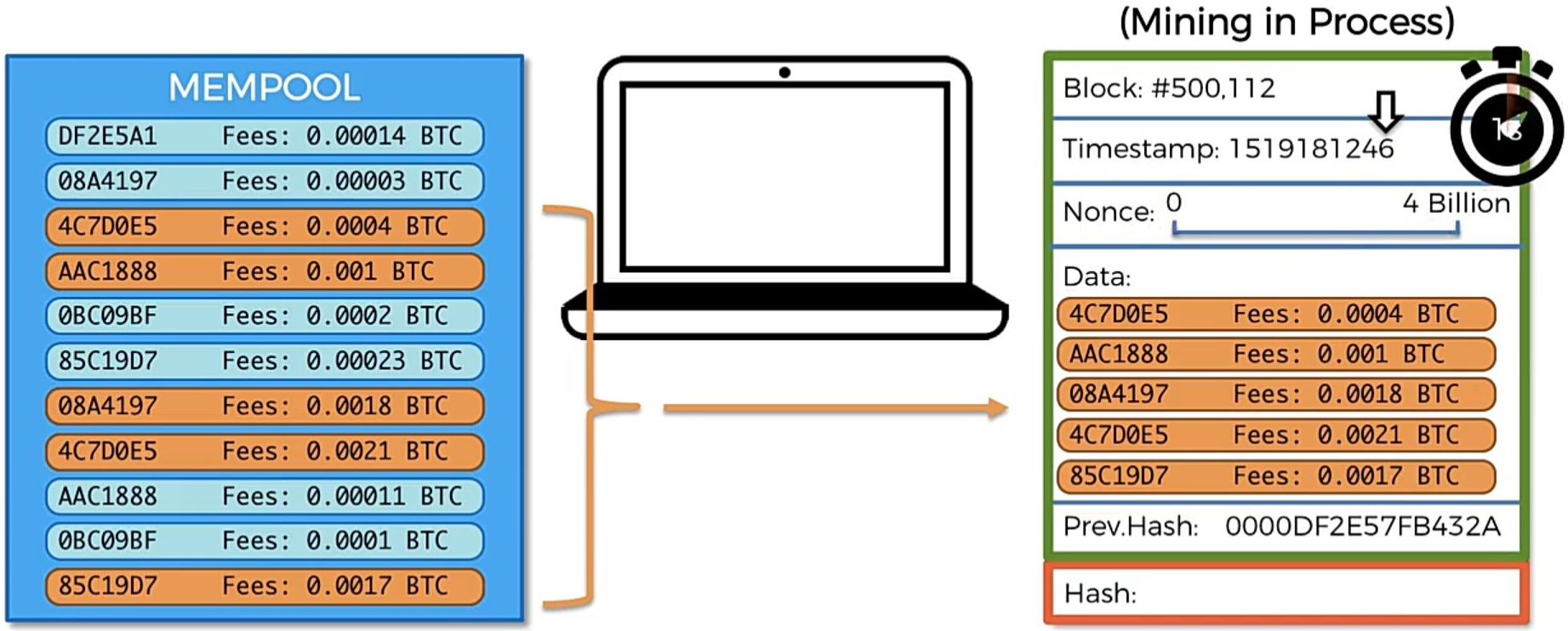
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

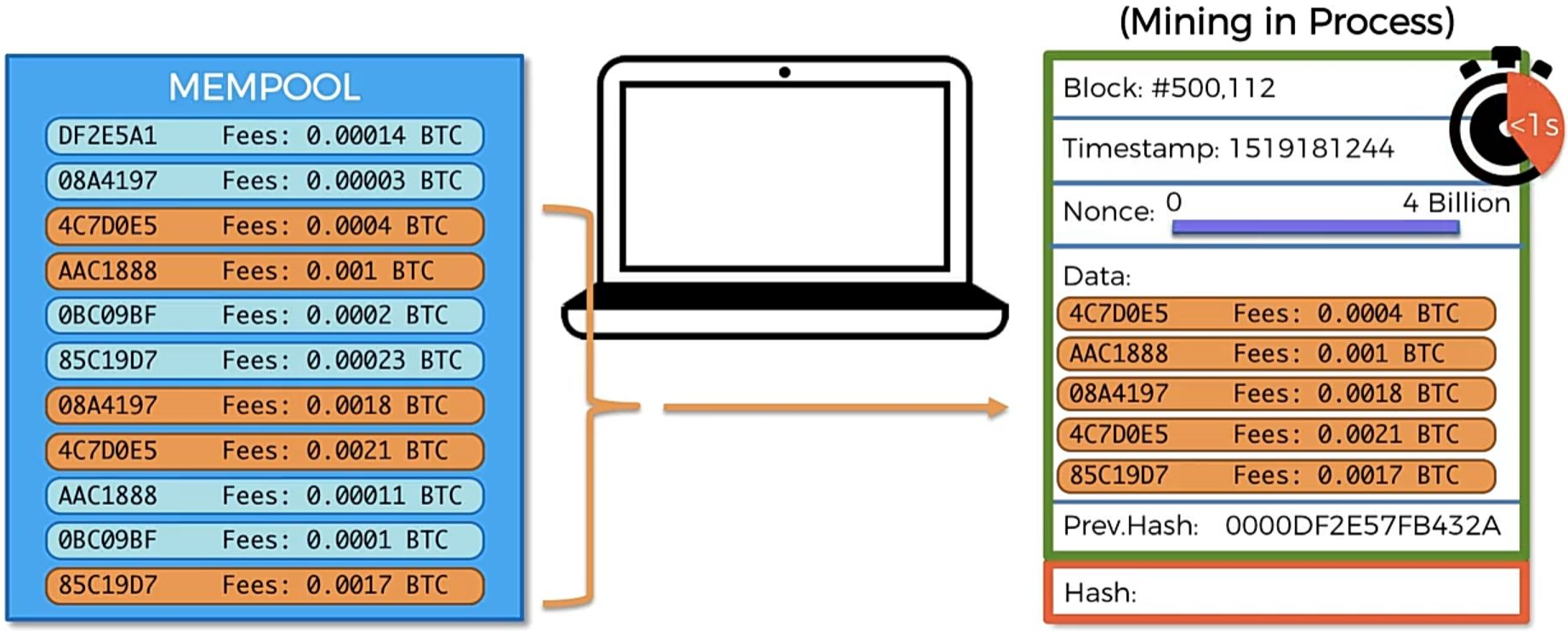
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

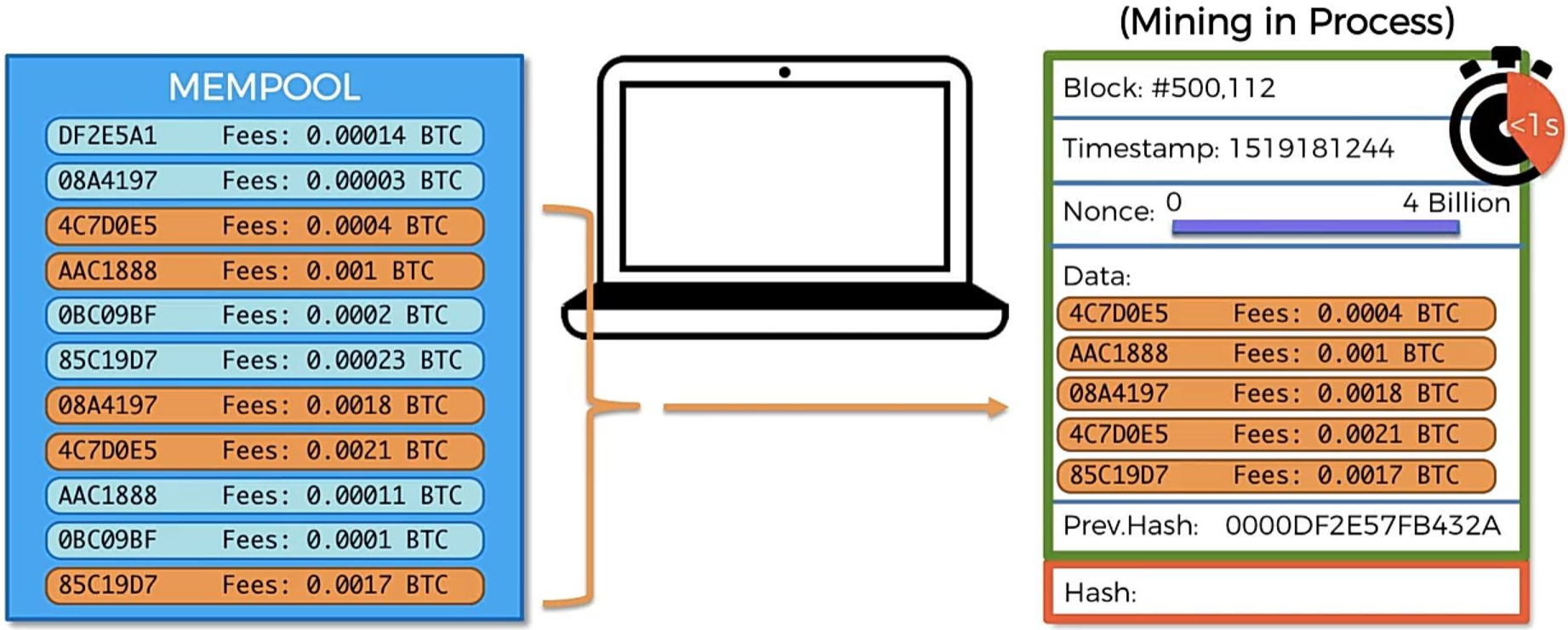
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

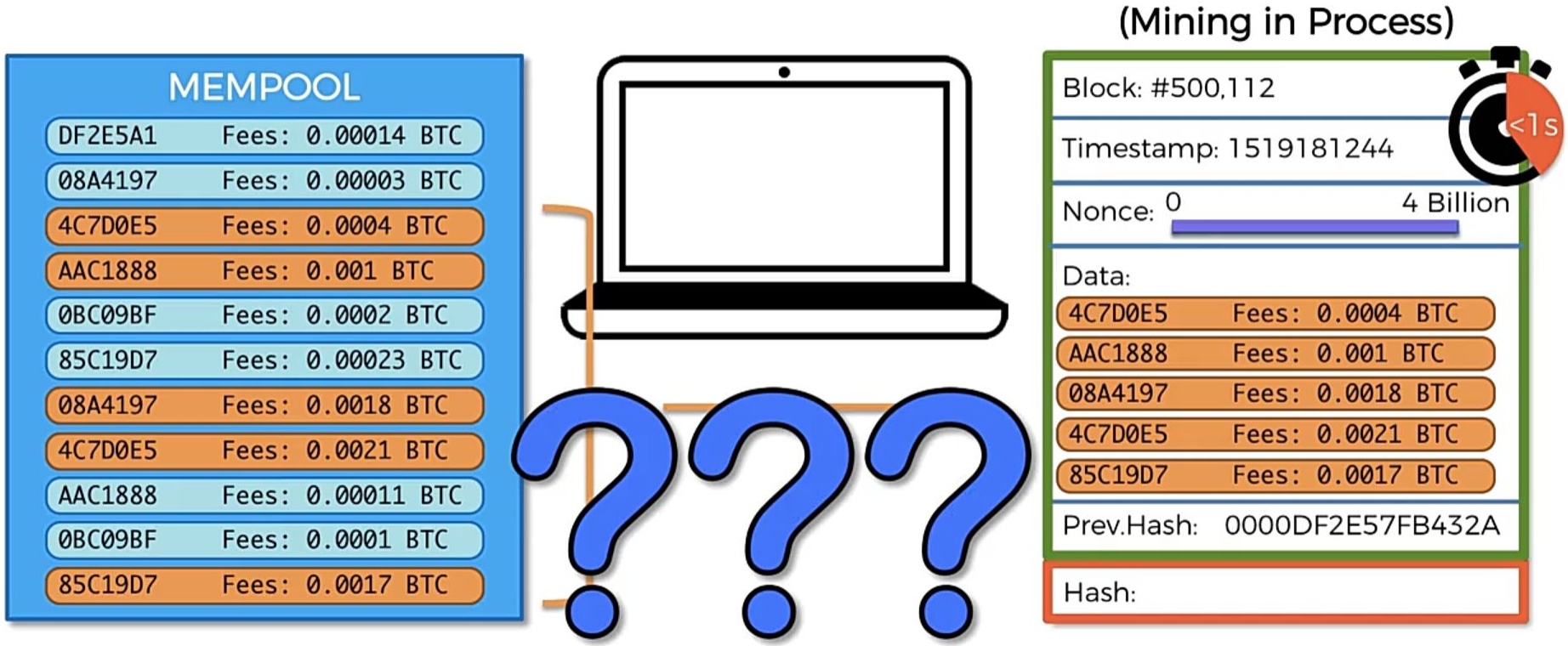
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

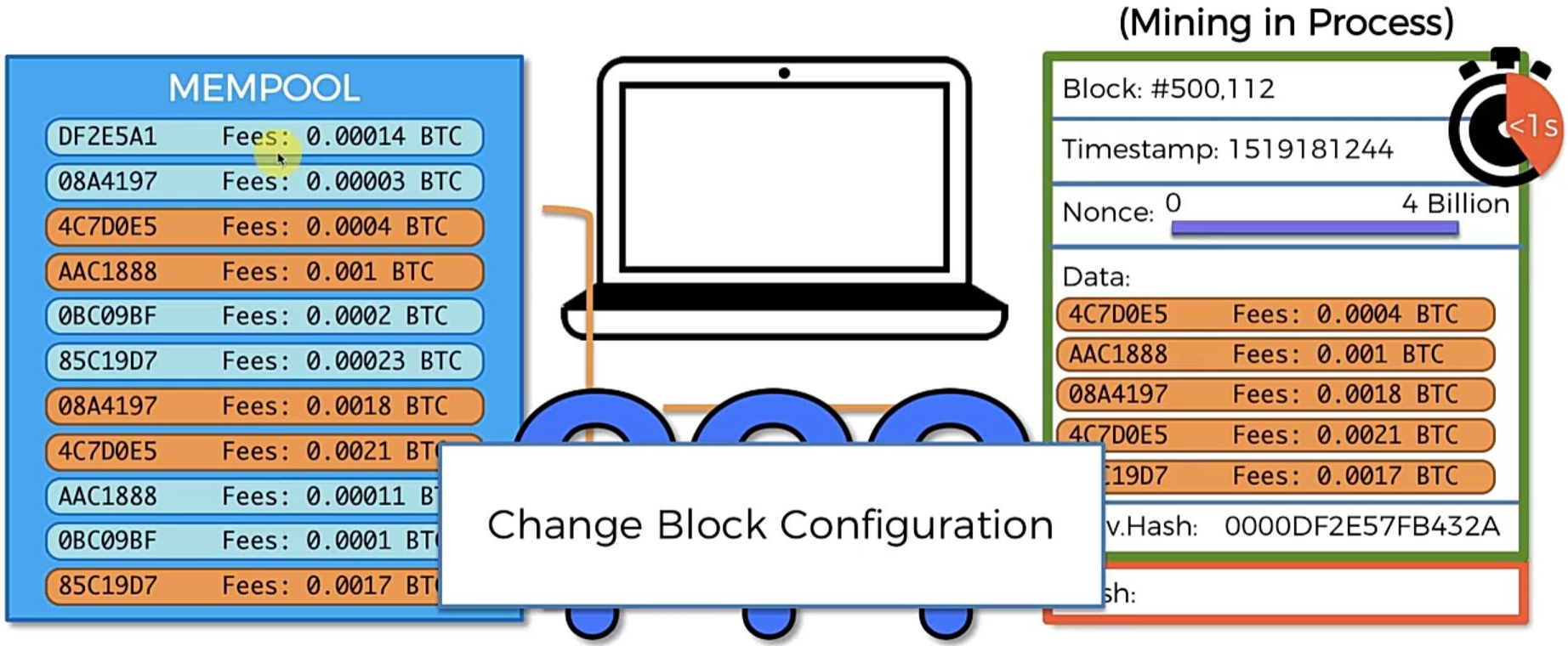
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

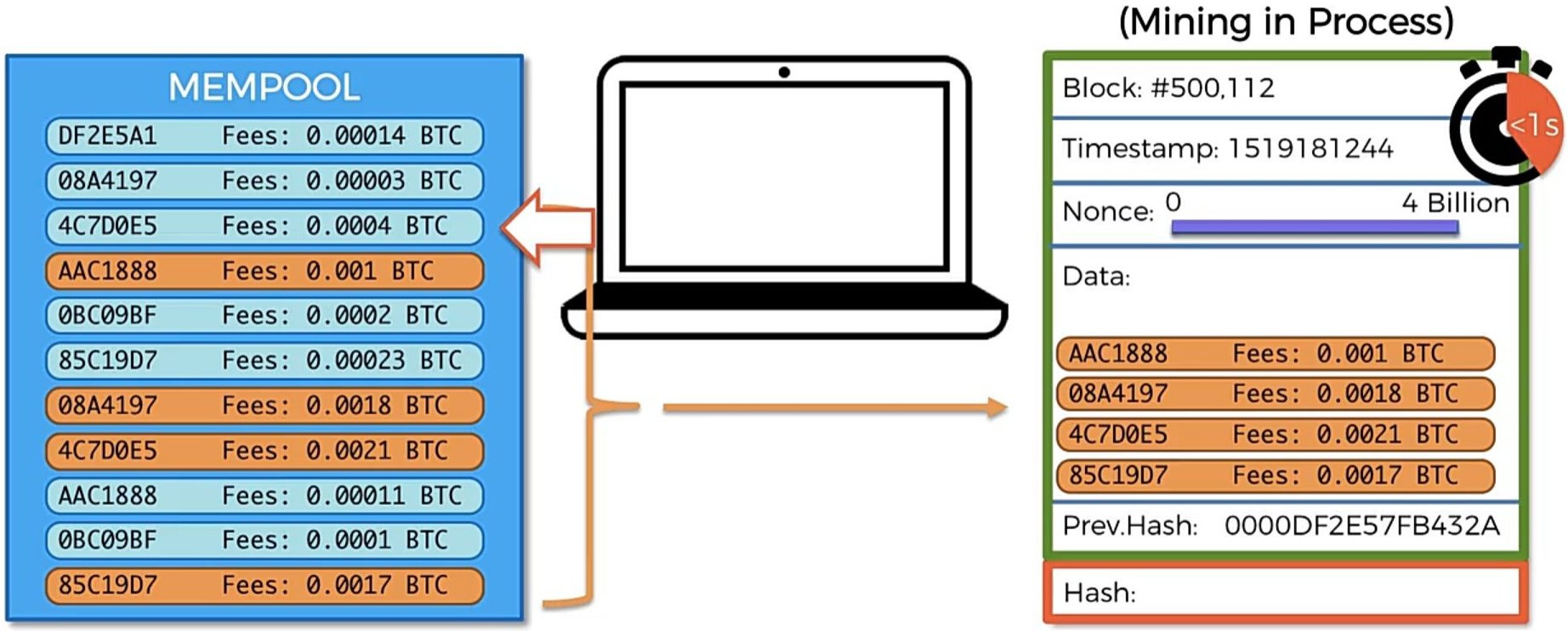
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

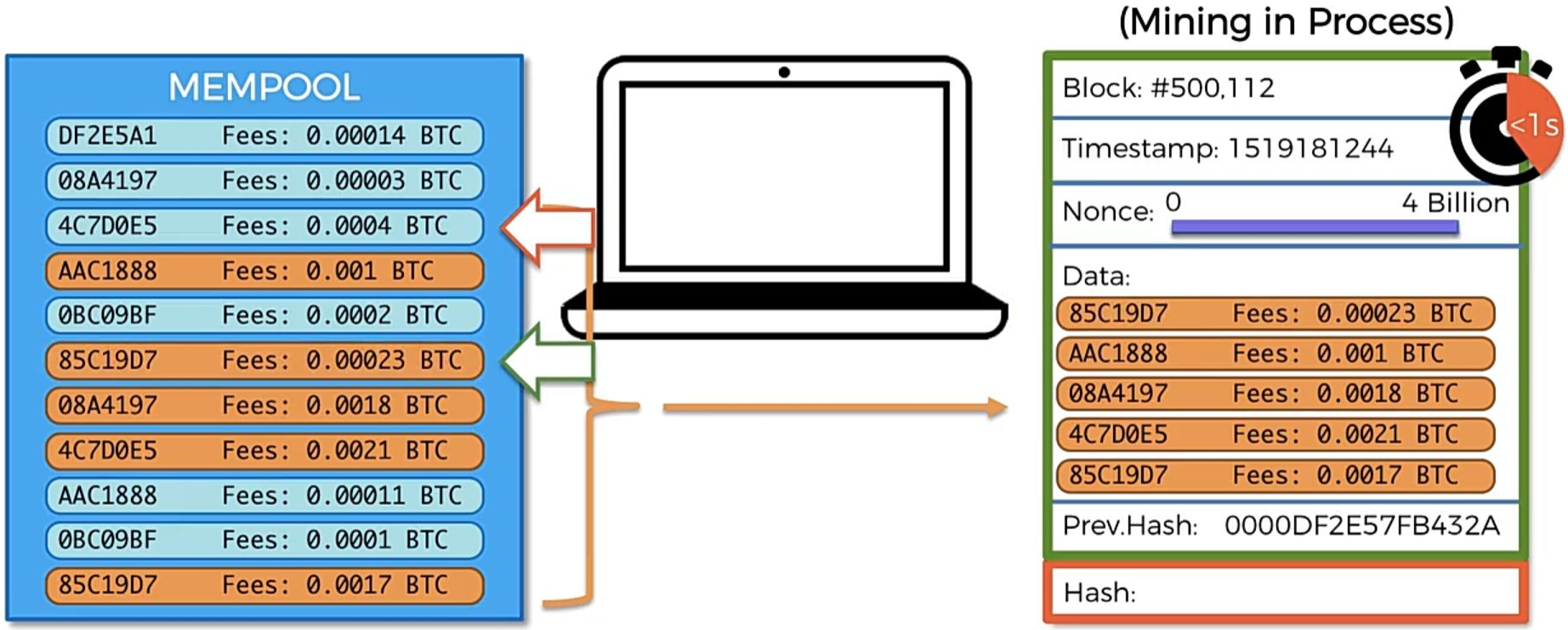
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

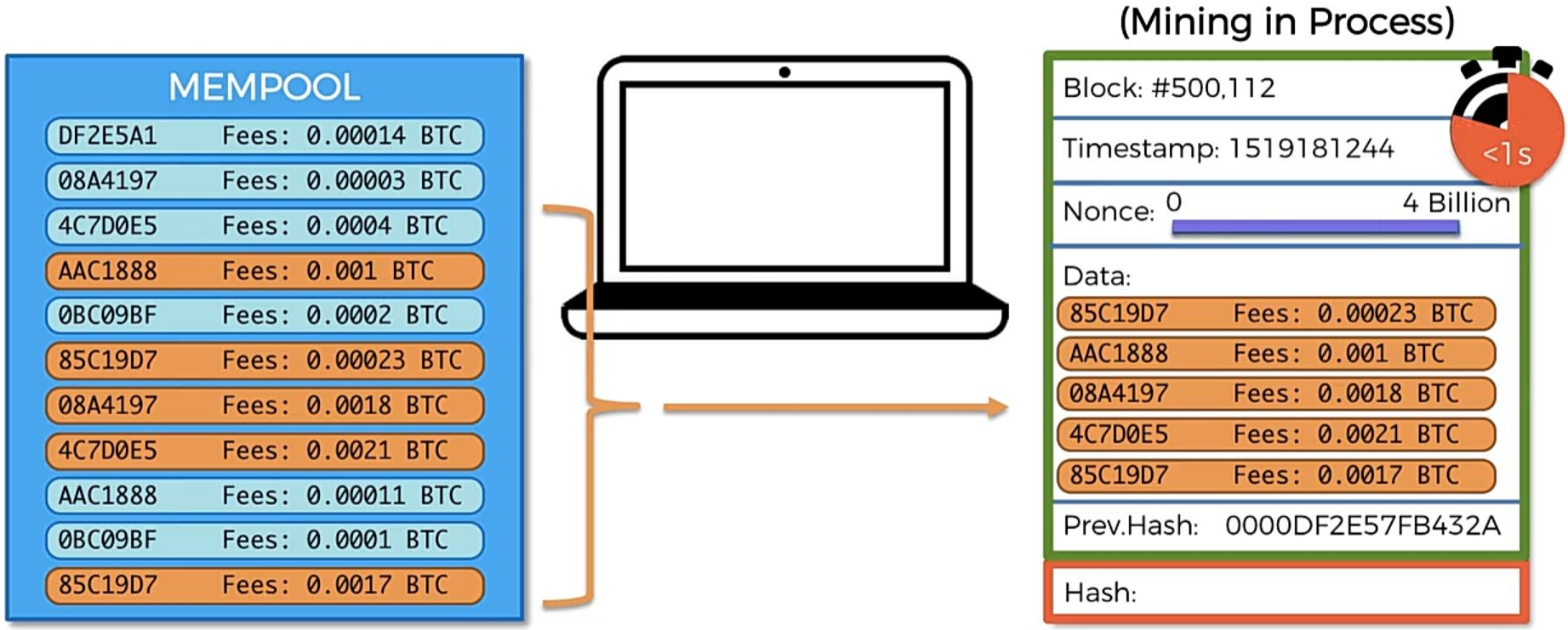
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

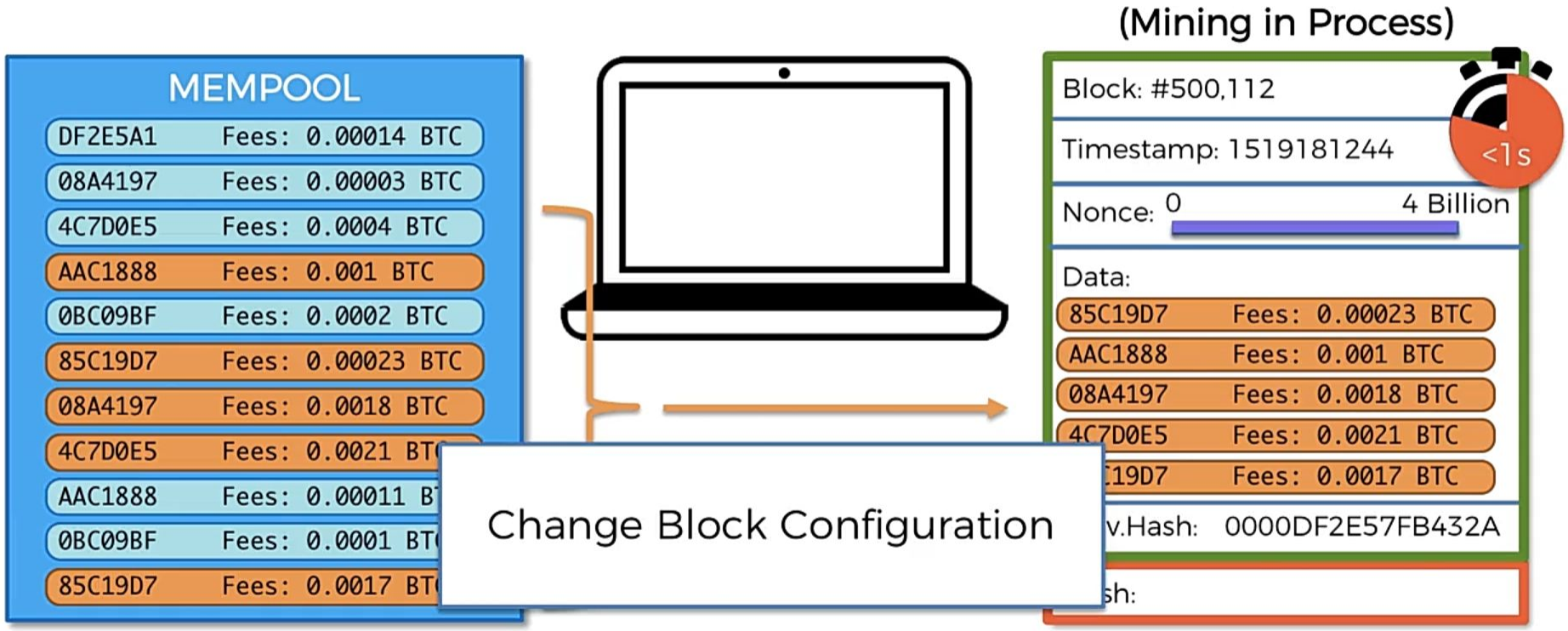
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

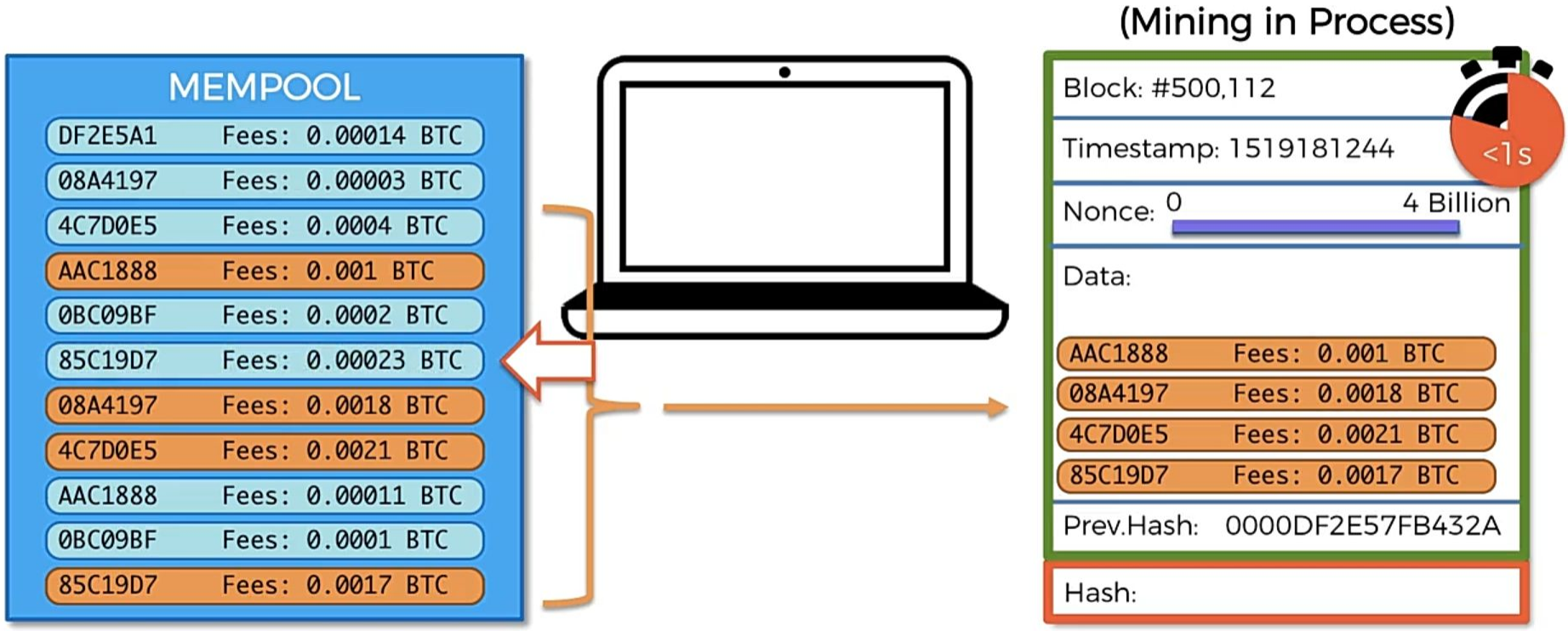
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

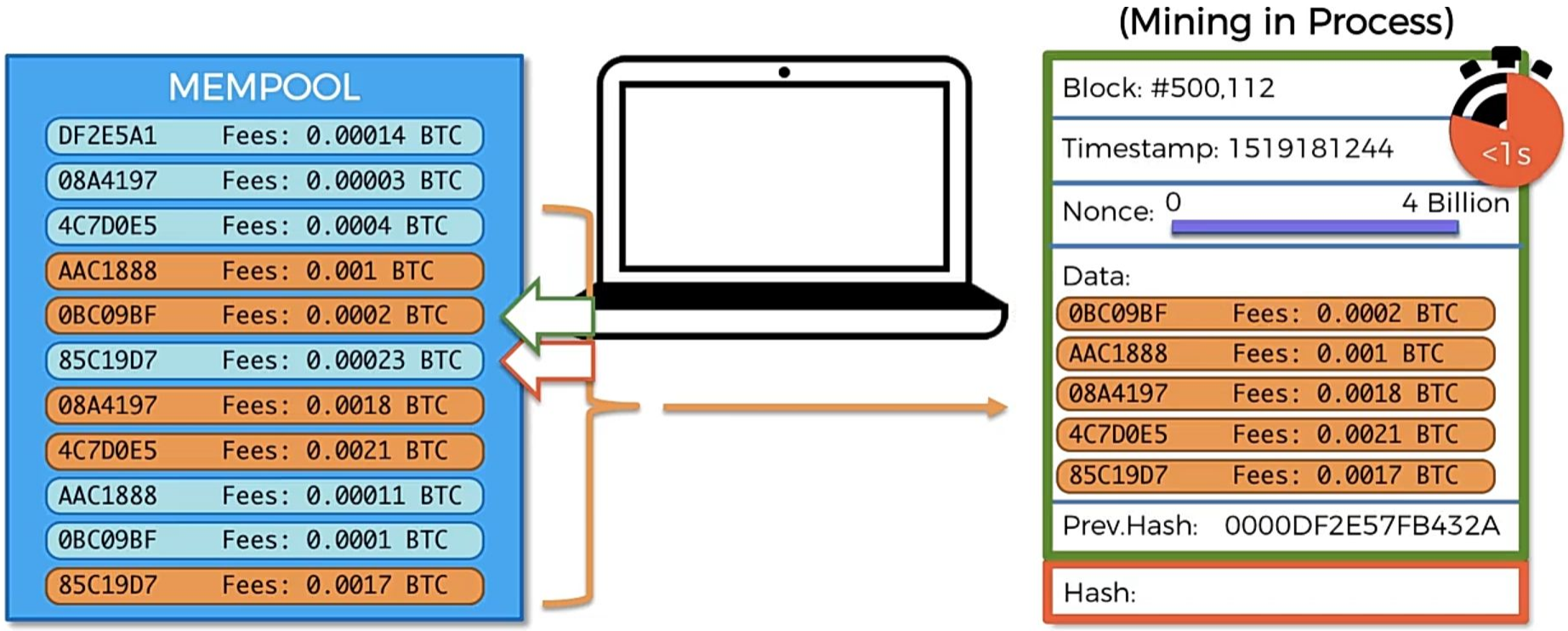
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

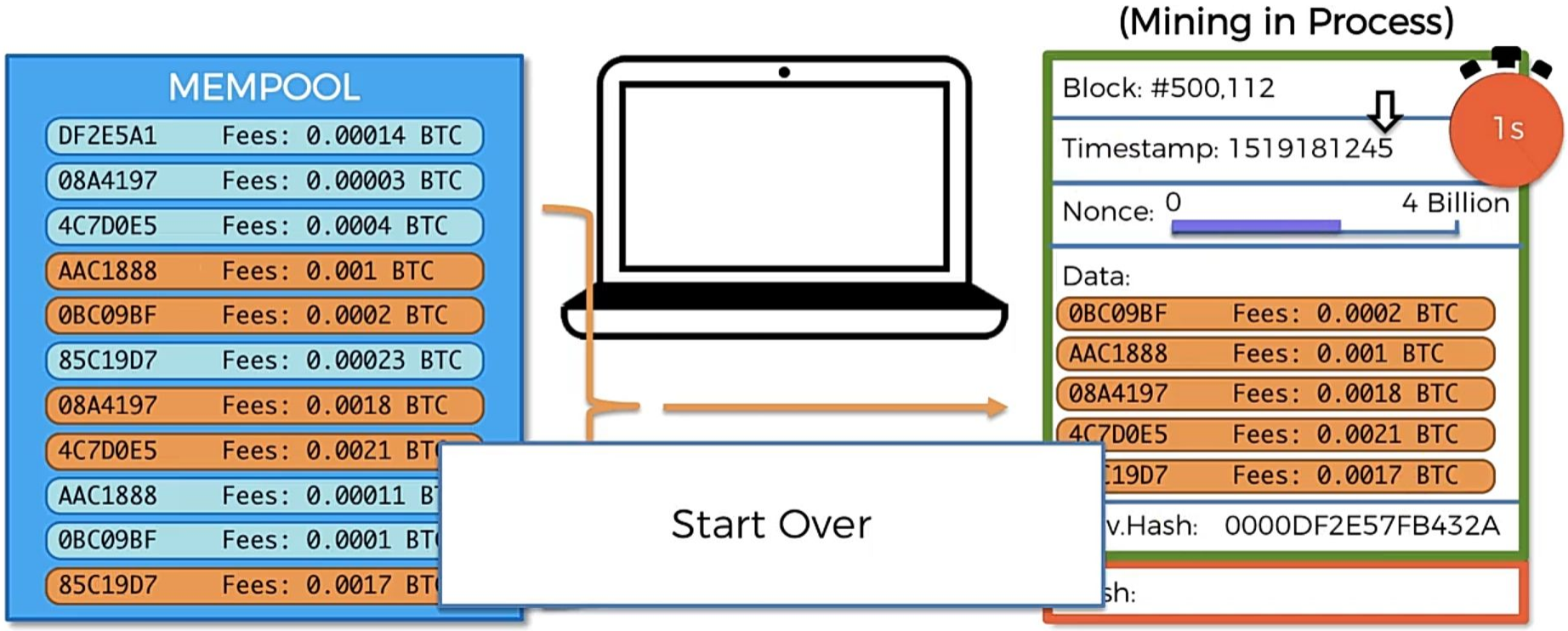
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

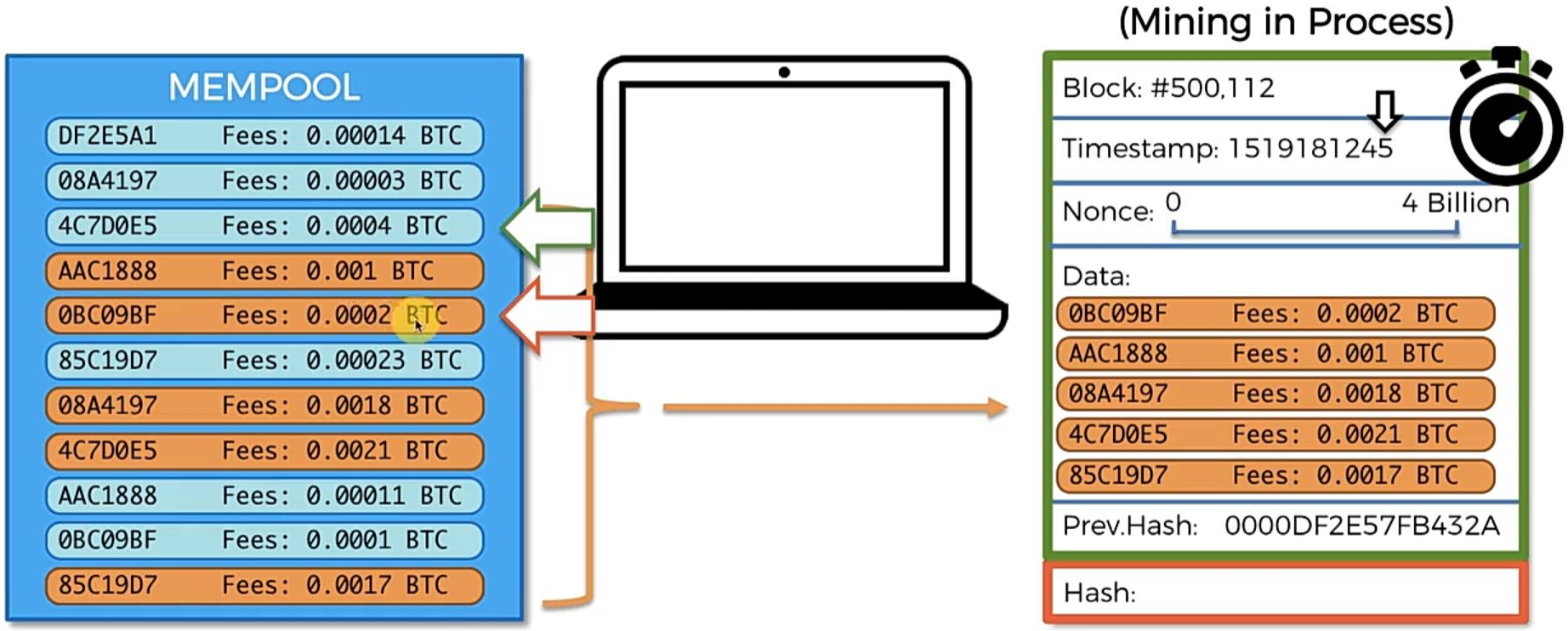
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

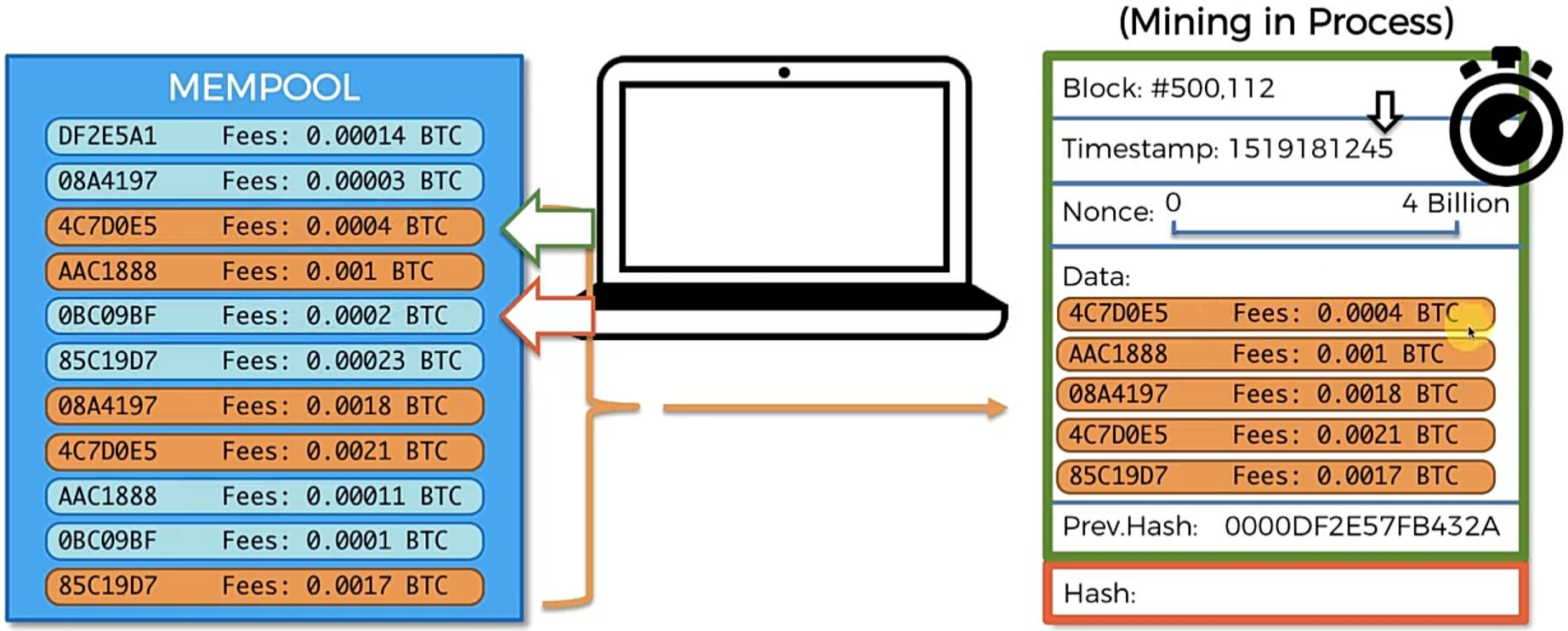
How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Transactions

How does Miner Picks transactions?



Courtesy : Super Data Science

1. Anatomy of a Blockchain - Mining Machines

CPU, GPUs and ASIC

CPU = Central Processing Unit	General	< 10 MH/s
GPU = Graphics Processing Unit	Specialized	< 1 GH/s
ASIC = Application-Specific Integrated Circuit	Totally Specialized	> 1,000 GH/s

Cloud Mining

1. Anatomy of a Blockchain - Life of Miners

ASIC (Application-specific integrated circuits) Miners

- most **effective and powerful mining hardware**
- Manufacturers build these machines with the sole purpose of mining a specific crypto algorithm
- As mining is an intensive and competitive process, it pushes these machines to their **maximum capabilities**.
- The average lifespan of a well-maintained machine : **3 to 5 years**.
- If kept in harsh or poor conditions, they can deteriorate in **a few months**.

Courtesy : Medium, YouTube

1. Anatomy of a Blockchain - Life of Miners

Common causes for ASIC damage and deterioration are:

1. **Little or no airflow**:
   - ASIC miners release a considerable amount of heat,
   - crucial to keep and run them in a well-ventilated location that keeps air moving and refreshing to avoid overheating.
2. **Humid, damp, or moist environments**:
   - internal components are damaged by humidity, which can cause rust and corrosion.
3. **Extreme temperatures**:
   - can shorten your hardware's lifespan by chipping away its internal components.
   - Cold tends to be less critical, as ASICs release heat that can counter-balance it.
   - Quick swings from below-zero to warm temperatures can cause condensation, provoking irreversible damage.

Courtesy : Medium, YouTube

1. Anatomy of a Blockchain - Life of Miners

Best practices for preserving ASIC miners

1. **Choosing a suitable location**
   - it must be a **dry** room with good, constant **airflow**, so **wide and open spaces** should be the first choice.
   - consider **installing additional fans** to keep the air moving, maintain the room dry, and avoid condensation.
2. **Mitigating the heat**
   - specialized and advanced cooling systems that reduce temperature,
   - innovative ways to **repurpose the heat produced by ASIC machines**, like heating pools or hot tubs, dehydrating fruit, or redirecting it to greenhouses for growing crops.
     - reduce or even eliminate the damage high temperatures have on miners,
     - enable **increased profitability** either by **reducing costs or adding another source of income**.
     - primary priority is **treating the excess heat** to preserve your miners.
3. **Regular maintenance and cleaning**
   - It is essential to **perform regular maintenance and clean the mining hardware**.
   - Removing accumulated dust not only prolongs lifespan, but also keeps performance high.

Courtesy : Medium, YouTube

1. Anatomy of a Blockchain - Nodes

WHAT ARE NODES?

Nodes are the electronic devices connected to the network and possessing an IP address. Generally, nodes are the communication endpoints which means that any user or application that wants to interact with the blockchain does so through nodes.

WHAT DOES A BLOCKCHAIN NODE DO?

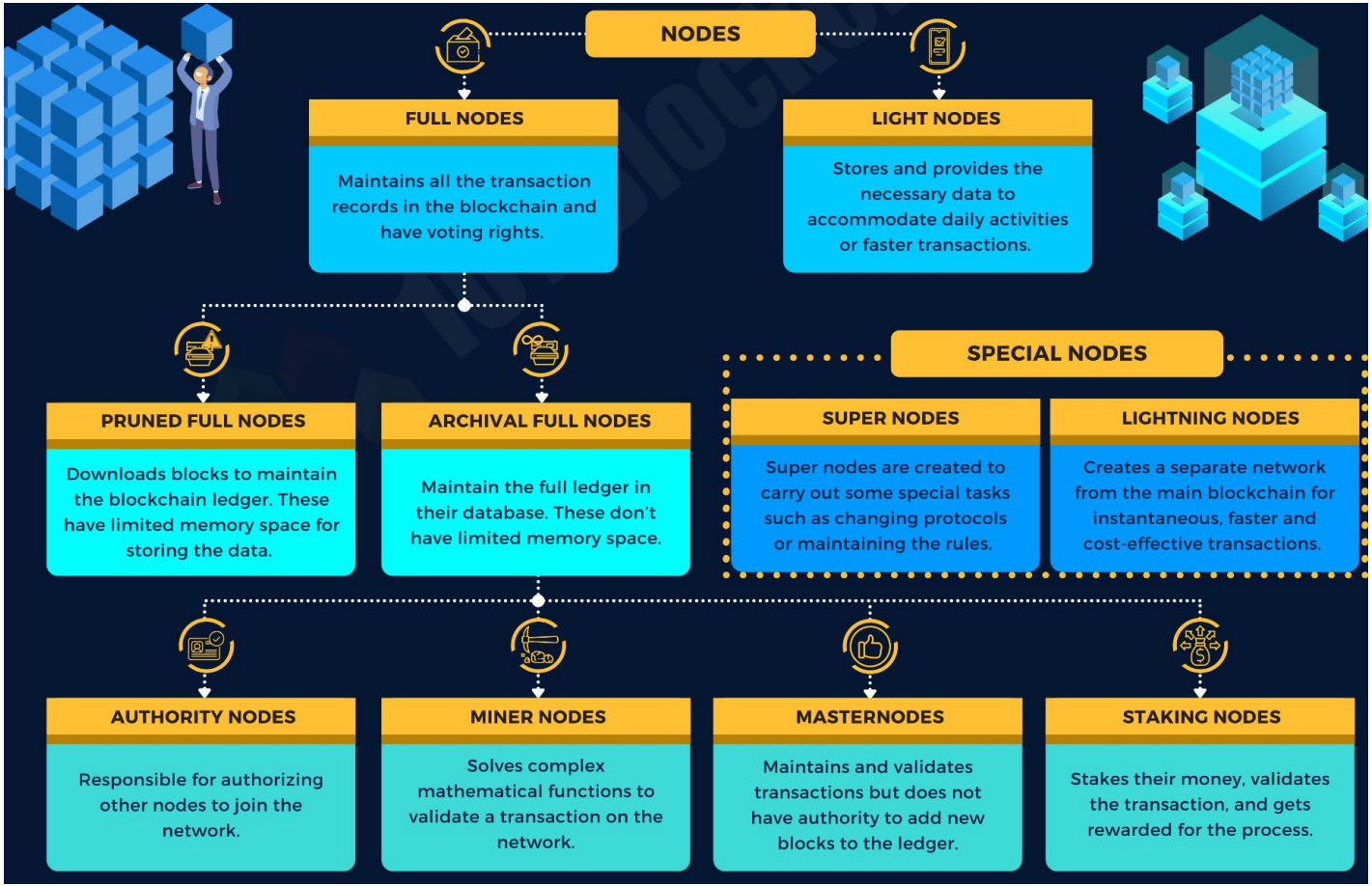
Processing A Transaction

Managing the transactions and their validity

Storing the cryptographically linked blocks

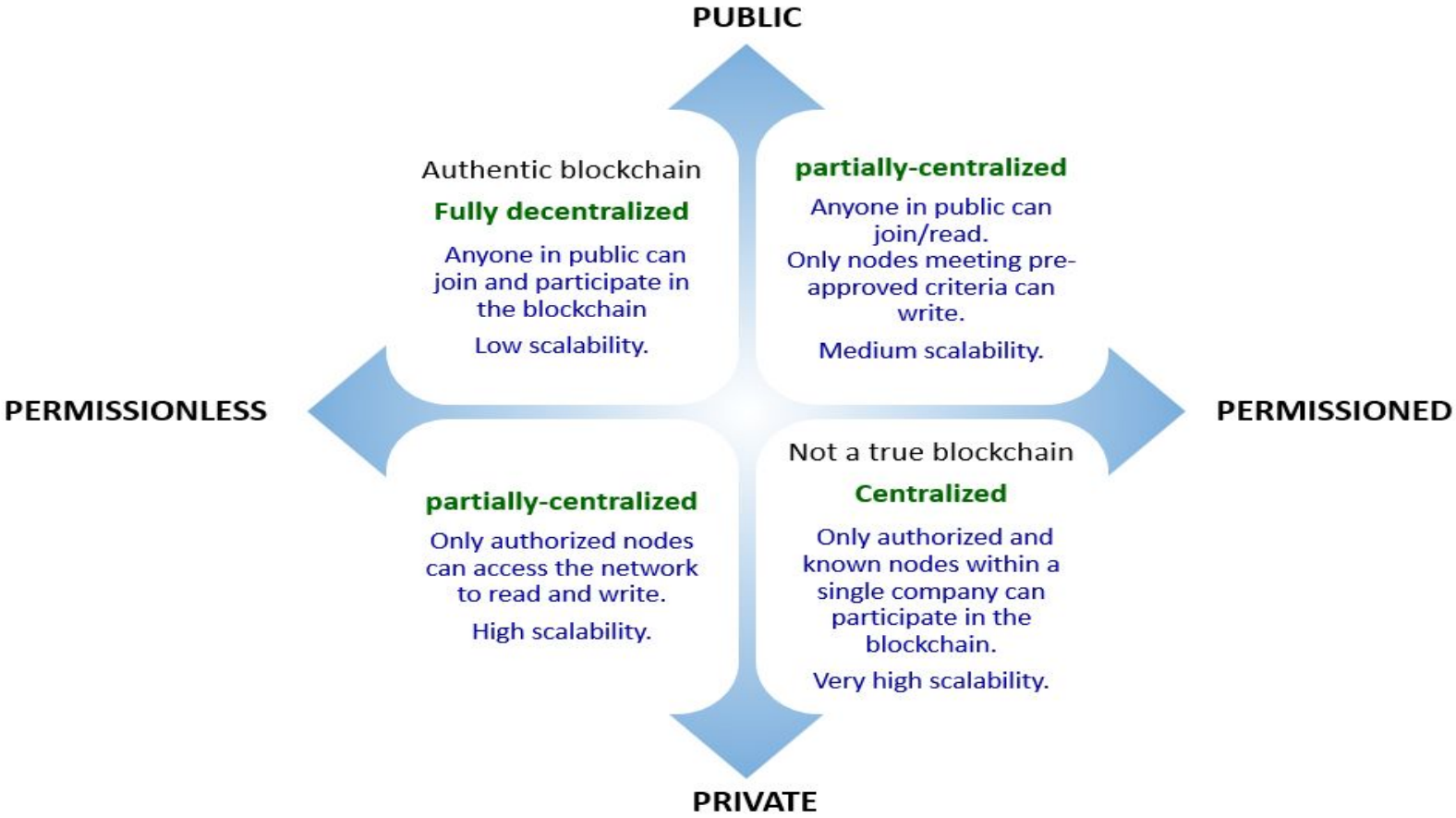
Acting as a point of communication

1. Anatomy of a Blockchain - Nodes

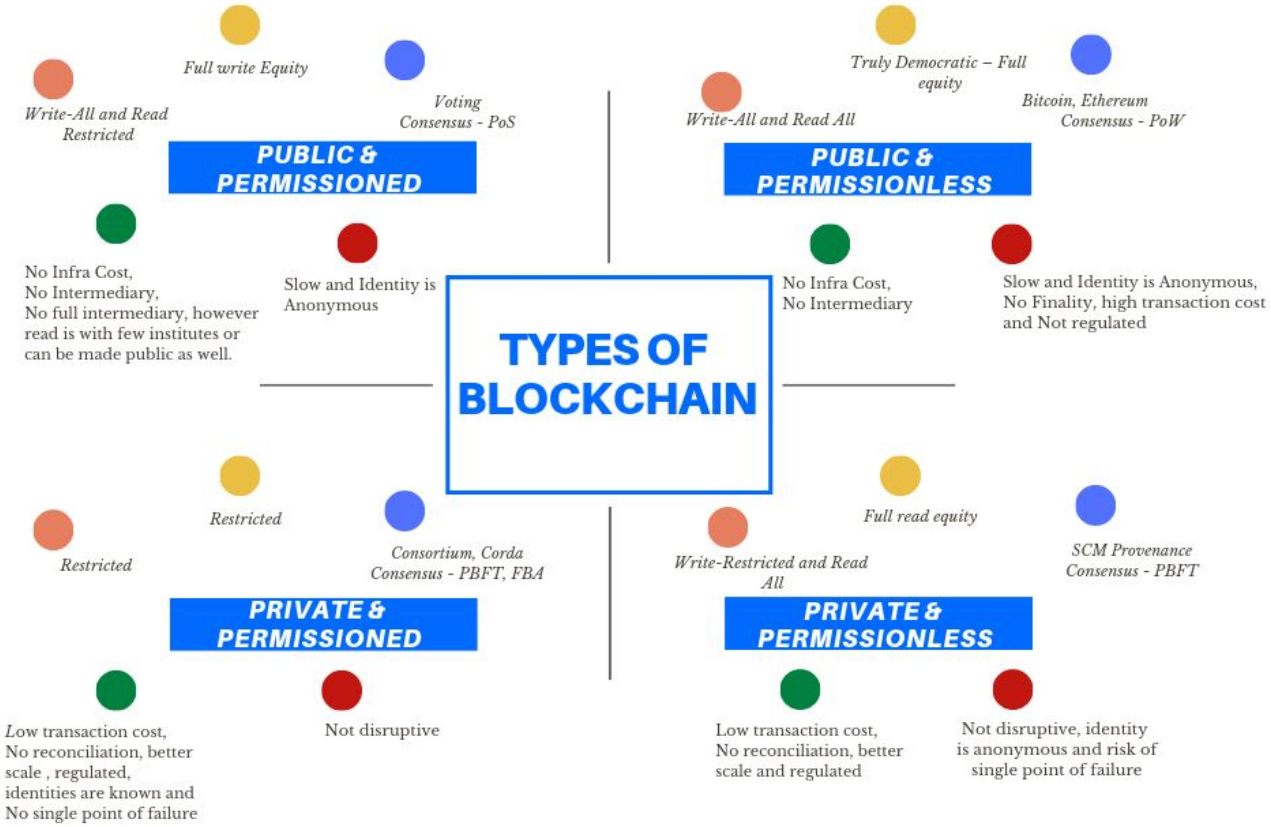


Courtesy : 101 Blockchain

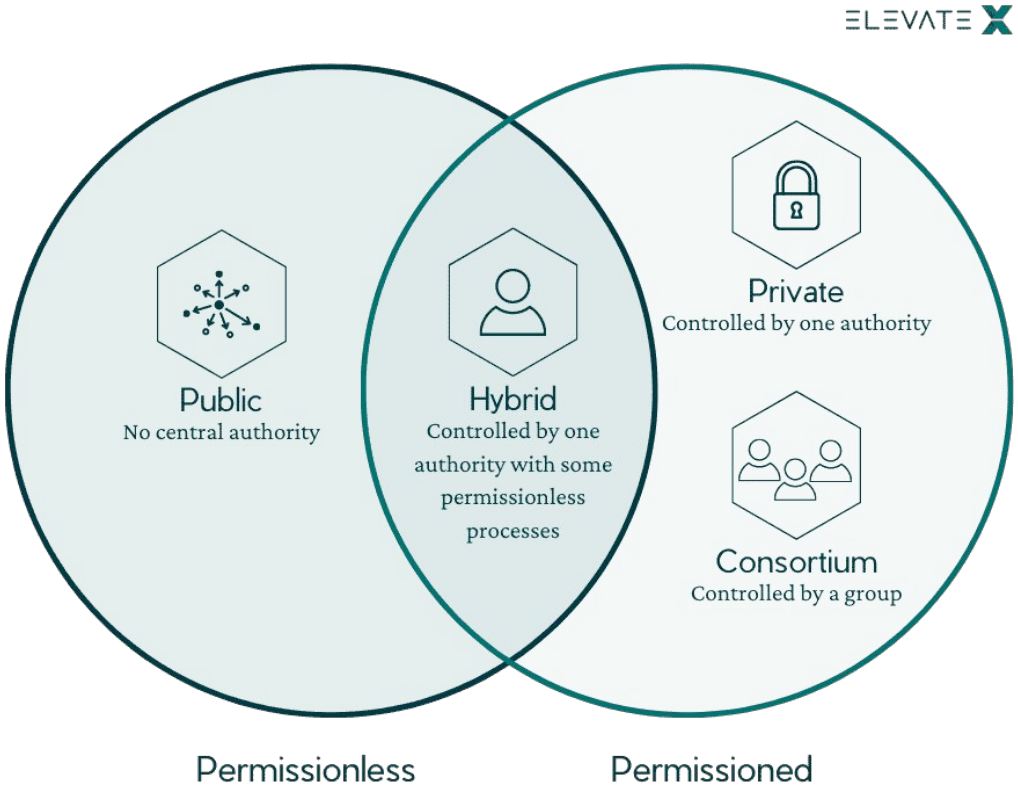
2. Types of Blockchains



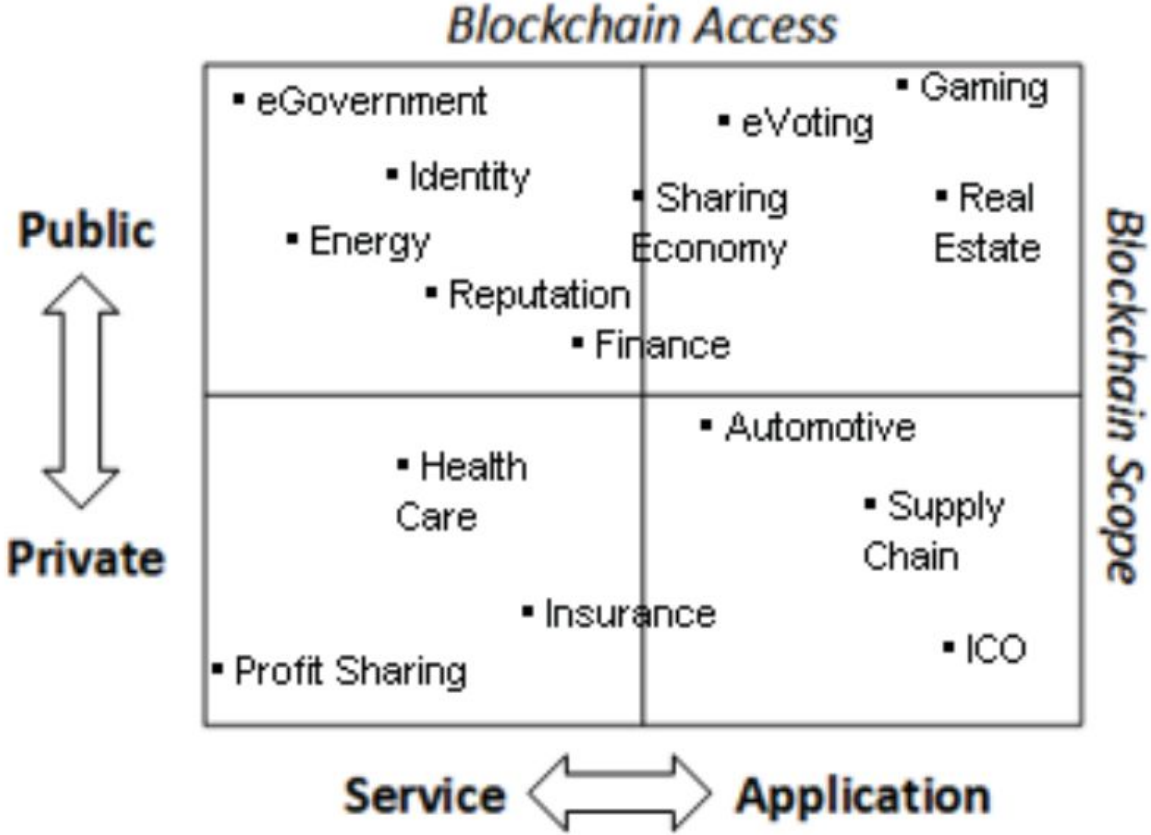
2. Types of Blockchains



2. Types of Blockchains



2. Types of Blockchains



Courtesy : Noah Fields

2. Types of Blockchains

	Public (permissionless)	**Private** (permissioned)	**Hybrid**	**Consortium**
ADVANTAGES	+ Independence + Transparency + Trust	+ Access control + Performance	+ Access control + Performance + Scalability	+ Access control + Scalability + Security
DISADVANTAGES	– Performance – Scalability – Security	– Trust – Auditability	– Transparency – Upgrading	– Transparency
USE CASES	▪ Cryptocurrency ▪ Document validation	▪ Supply chain ▪ Asset ownership	▪ Medical records ▪ Real estate	▪ Banking ▪ Research ▪ Supply chain

Courtesy : Christine Campbell

3. Exploring Key Blockchain Platforms - Need

Key aspects that make blockchain platforms appealing:

1. **Decentralization**:
   - Traditional systems often rely on a central authority, like a bank or government, to validate and record transactions.
   - In a blockchain, the ledger is distributed across a network of nodes, eliminating the need for a central authority.
2. **Security:**
   - Blockchain uses cryptographic techniques to secure transactions and control the creation of new units.
   - The decentralized and consensus-based nature of blockchain makes it resistant to tampering and fraud.
   - Once a block is added to the chain, it is extremely difficult to alter previous blocks, enhancing the overall security of the system.
3. **Transparency:**
   - All participants in a blockchain network have access to the same information.
   - This transparency helps build trust among users, as they can independently verify the transactions and data stored on the blockchain.

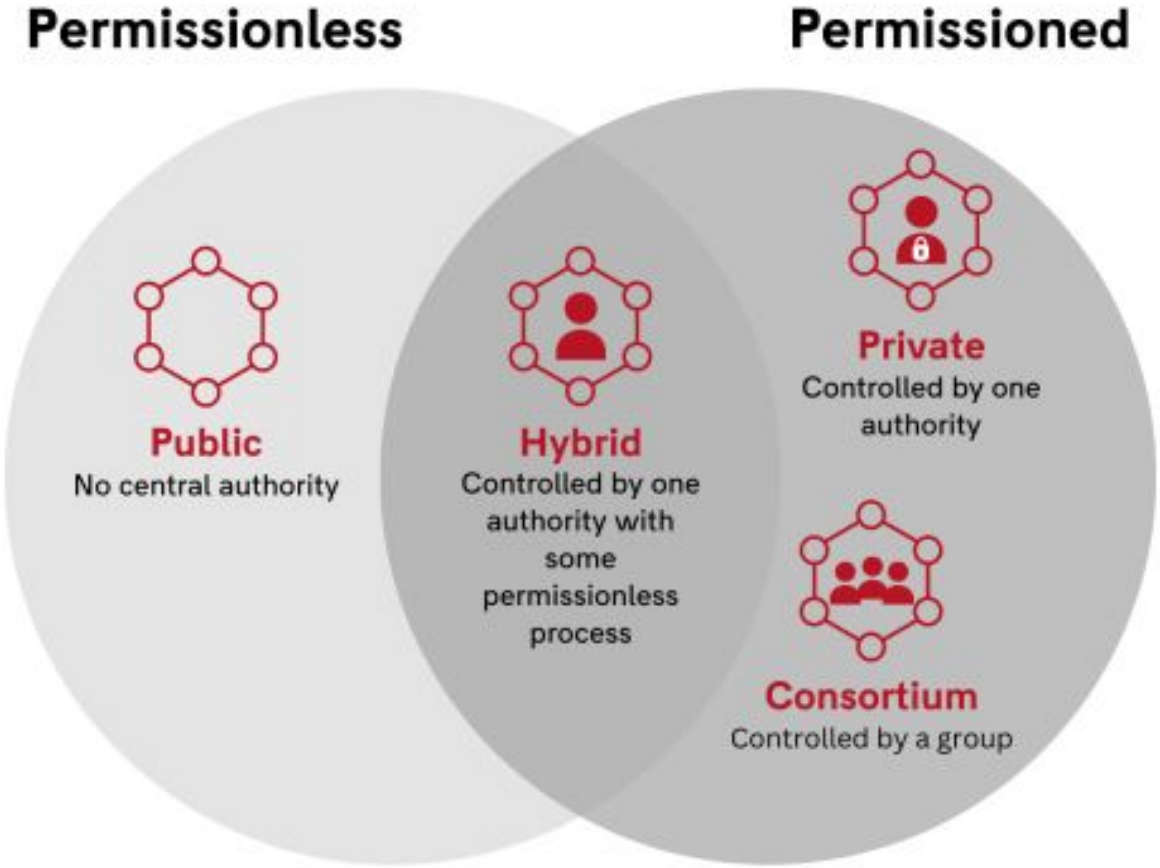
3. Exploring Key Blockchain Platforms - Need

4. **Immutability:**
   - Once a block is added to the blockchain, it becomes very difficult to change.
   - This immutability ensures that historical transactions are preserved and cannot be altered, providing a reliable and auditable record of events.
5. **Smart Contracts**:
   - Smart contracts are self-executing contracts with the terms of the agreement directly written into code.
   - They automatically execute and enforce the terms when predefined conditions are met, eliminating the need for intermediaries.
6. **Efficiency and Cost Reduction:**
   - Blockchain can streamline and automate processes, reducing the need for intermediaries and minimizing the associated costs.
   - The decentralized nature of blockchain also eliminates the single point of failure found in centralized systems, improving overall system reliability.
7. **Global Accessibility**:
   - Blockchain operates on a global network, allowing participants from different parts of the world to engage in transactions without the need for intermediaries or traditional banking systems.
   - This can be particularly beneficial for cross-border transactions.

3. Exploring Key Blockchain Platforms - Need

8. **Tokenization:**
   - Blockchain enables the creation of digital tokens, representing various assets such as real estate, art, or even shares in a company.
   - This facilitates new forms of ownership, investment, and crowdfunding.
9. **Data Integrity**:
   - Blockchain ensures data integrity by design.
   - Each block contains a unique identifier (hash) and references the previous block's hash.
   - This interconnection makes it extremely difficult for malicious actors to alter data without detection.
10. **Use Cases:**
    - Blockchain has found applications in various industries, including **finance** (cryptocurrencies like Bitcoin and decentralized finance platforms), **supply chain management, healthcare** (patient data management), **voting systems**, and more

Blockchain Platform Types



Courtesy : CFTE Education Blog

Blockchain Platform Types

	Permissioned	Permissionless
Preferred Purpose	Popular among industry-level firms and enterprises	Popular for public use
Decentralization	Limited decentralization	Broad decentralization
Development	Generally proprietary	Generally open source
Transparency	Less transparent	More transparent
Use cases	Manage supply chains Create contracts Verify payment between parties	Cryptocurrency blockchains

Courtesy : CFTE Education Blog

Blockchain Platform Types (3 / 11)

- Categorized into several types based on their **design, consensus mechanisms, and use cases**.
- Common types of blockchain platforms:
  1. **Public Blockchains:**
     - Open to anyone.
     - Decentralized and distributed ledger.
     - Examples: Bitcoin, Ethereum.
  2. **Private Blockchains:**
     - Restricted access to a specific group or organization.
     - Often used for internal purposes or within a consortium.
     - Offers more control over the network.
     - Examples: Hyperledger Fabric, Corda.
  3. **Consortium Blockchains:**
     - Semi-decentralized, involving a group of organizations.
     - Shared control among multiple entities.
     - Enhances efficiency through collaboration.
     - Examples: R3 Corda (which can be used both as private and consortium).

Blockchain Platform Types (4 / 11)

4. **Permissionless Blockchains:**
   - Anyone can join and participate.
   - Typically associated with public blockchains.
   - Examples: Bitcoin, Ethereum.
5. **Permissioned Blockchains:**
   - Participants are known and must have permission to join.
   - Offers more privacy and control.
   - Common in private and consortium blockchains.
   - Examples: Hyperledger Fabric, Quorum.
6. **Smart Contract Platforms:**
   - Allows the execution of smart contracts.
   - Automated, self-executing contracts with the terms written into code.
   - Examples: Ethereum, Binance Smart Chain.
7. **Cryptocurrency Platforms:**
   - Focused on the creation and management of digital currencies.
   - Often includes a native cryptocurrency.
   - Examples: Bitcoin, Litecoin.

Blockchain Platform Types (4 / 11)

8. **Supply Chain Blockchain:**
   - Designed for tracking and managing supply chain activities.
   - Improves transparency and traceability.
   - Examples: IBM Food Trust, VeChain.
9. **Identity Management Blockchains:**
   - Focuses on secure and decentralized identity verification.
   - Helps prevent identity theft and fraud.
   - Examples: Sovrin, uPort.
10. **Cross-Chain Platforms:**
    - Allows interoperability between different blockchains.
    - Enables the transfer of assets and data between blockchains.
    - Examples: Polkadot, Cosmos.
11. **Tokenization Platforms:**
    - Used for creating and managing digital tokens.
    - Enables the representation of real-world assets on the blockchain.
    - Examples: Ethereum (with ERC-20, ERC-721 standards), Binance Smart Chain.

Blockchain Platform Types - Public Blockchain

- completely permissionless.
- Blockchain networks are decentralised, meaning that no single organisation or individual lies at the center of it, controlling it and users can remain anonymous.
- With no permissions, anyone has the liberty to join and participate in the primary activities of a public blockchain.
- That's why public blockchains are known to be self-governed because users have complete freedom to read, write and audit activities on its network.
- **Main features of public blockchain:**
  - Permissionless
  - Largely decentralised
  - Self-governed
  - Maintain's anonimity of users
- **Examples of public blockchain:**
  - Bitcoin
  - Ethereum
  - Litecoin

Blockchain Platform Types - Private Blockchain

- it requires users on its network to be verified.
- Users can only join such a network through invitation when their identity is authenticated.
- They are also given express permission on the level of access they have and the activities they can perform. This means that private blockchains have a single central authority or owner that controls granting access to it and also has the right to override, edit, or delete entries on the blockchain when it deems necessary.
- This results in the network being partially decentralized and smaller than public blockchains because of restricted access and central control.
- However, on the upside, this also means that the networks are fast, efficient and trusted.
- **Main features of private blockchain:**
  - Permissioned
  - Partially decentralised
  - Highlly efficient
  - Trusted network
- **Example of private blockchain:**
  - Linux Foundation's Hyperledger Fabric

Blockchain Platform Types - Hybrid Blockchain

- Amalgamation of a private and public blockchain, combining the best of both networks into one.
- It has the capability of constructing a private, permissioned based system alongside a public, permissionless one.
- This creates a network that is not restricted by one blockchain's limitations
- Allows a single owner or central authority to control who has access to certain data and what is made public.
- **Main features of hybrid blockchain:**
  - Controlled by a central authority
  - A combination of permissioned and permissionless blockchain
  - Information can be accessed via smart contracts
- **Example of hybrid blockchain:**
  - XRP token

Blockchain Platform Types - Consortium Blockchain

- Known as **federated blockchain**,
- Another type of permissioned blockchain with its main point of differentiation being that it is a group of private blockchains owned by multiple individuals or entities.
- While each organisation manages their own blockchain in a consortium blockchain,
- it also allows data from several different sources to be integrated while ensuring a secure and efficient data flow.
- **Main features of consortium blockchain:**
  - Permissioned
  - Partially Decentralized
  - Controlled by a group of entities
  - Privacy and security of data is ensured
- **Examples of consortium blockchain:**
  - R3's Corda
  - ConsenSys' Quorum

Blockchain Platform Types : Comparison

	Public	Private	Hybrid	Consortium
Permissioned/ Permissionless	Permissionless	Permissioned	Permissioned & Permissionless	Permissioned
Control	No control by a central authority	Control by a central authority	Control by a central authority	Control by multiple central authorities
Main Advantages	✓ Independence ✓ Transparency	✓ Performance ✓ Scalability	✓ Performance ✓ Low Cost	✓ Performance ✓ Security
Main Disadvantages	✗ Performance ✗ Scalability Issues	✗ Security ✗ Trust	✗ Transparency ✗ Upgrading	✗ Transparency
Examples	Bitcoin Litecoin	Hyperledger Fabric	XRP token	Corda Quorum

Courtesy : CFTE Education Blog

Checklist for Choosing a Blockchain Platform

Top 5 Blockchain Platforms for 2025

1. **Ethereum :** popular for decentralized applications (dApps) and smart contracts, Layer 2 scaling solutions.
2. **Solana:** exceptional transaction speed and low fees (best for DeFi, NFTs, and high-performance applications.
3. **Hyperledger Fabric** :
   - popular permissioned, modular blockchain platform
   - designed for enterprise-grade applications (privacy and security requirements)
4. **Polkadot :** focus on **interoperability** between blockchains. Uses parachains connected to a central relay chain. blockchains that can securely communicate with each other.
5. **Binance Smart Chain (BNB Chain) :**
   - EVM-compatible and optimized for **fast, low-cost transactions**.
   - Strong DeFi adoption and easy migration from Ethereum.
   - Widely used for DApps and token ecosystems

Courtesy : Blockchain Council

Comparison on the Top Blockchain Platforms

Feature	Ethereum	Hyperledger Fabric	Solana	Polkadot	BNB Chain
Type	Public	Permissioned	Public	Public	Public
Primary Use	DeFi, NFTs, DAOs, DApps	Enterprise & Govt solutions	High-speed DeFi, NFTs, Gaming	Interoperable blockchains	Low-cost DeFi & DApps
Smart Contracts	Yes (Solidity, Vyper)	Yes (Chaincode)	Yes (Rust, C)	Yes (Substrate-based)	Yes (Solidity – EVM)
Consensus	Proof of Stake (PoS)	Raft / PBFT	PoH + PoS	Nominated PoS	Proof of Staked Authority
Permission Model	Permissionless	Permissioned	Permissionless	Permissionless	Permissionless
Transaction Speed	Moderate (Layer-2 improves)	High	Very High	High	High

Courtesy : 101 Blockchain

Comparison on the Top 5 Blockchain Platforms

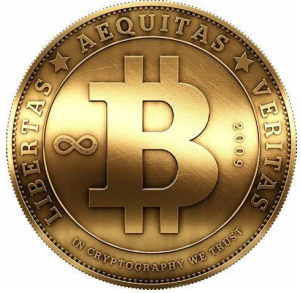
Feature	Ethereum	Hyperledger Fabric	Solana	Polkadot	BNB Chain
Transaction Cost	High / Variable	Low	Very Low	Low	Very Low
Decentralization	High	Low–Medium	Medium	High	Medium
Scalability	Via Layer-2s	Native	Native	Parachains	Native
Token Needed	ETH	No native token	SOL	DOT	BNB
Enterprise Adoption	Medium	Very High	Low–Medium	Medium	Medium
Governance	Community driven	Organization controlled	Foundation-led	On-chain governance	Binance influenced

Courtesy : 101 Blockchain

Bitcoin : Introduction

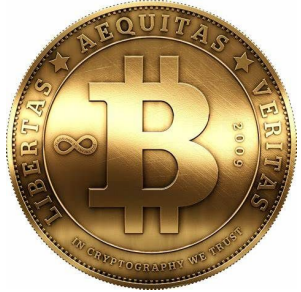
- Bitcoin is one of the **famous cryptocurrencies.**
- Bitcoin was found by **Satoshi Nakamoto.**
- Bitcoin is a **Peer-to-Peer Electronic cash system (2008)**
- Bitcoin is **not something physical**
- It is **easy to transport anywhere with low cost**
- **No third party (intermediary)**

Bitcoin : Introduction



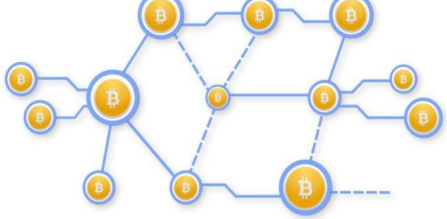
- It is a **decentralized digital currency without a central bank** that can be **sent from user to user on the peer-to-peer bitcoin network without the need for intermediaries.**
- **Transactions are verified by network nodes** through **cryptography** and **recorded in a public distributed ledger called a blockchain.**
- **Launched** in **2009**, Bitcoin is the **world's largest cryptocurrency** by market cap.

Why Bitcoin is Popular?



- It was the only cryptocurrency
- It is **decentralised currency** (Regulate by users, not any bank)
- It **allow user to do transaction anonymously**
- It is **tax free** (No matter how many bitcoins you have)
- **Transfer fund easily at very low cost**; Worldwide

Why Bitcoin is Popular?



- Bitcoin network is public - Source code is open source
- Owner uses **private key** to spend bitcoins
- **Public ledger**: All bitcoin exchanges are visible to everyone on the network
- Transfer/Spending of coins require very little fee
- Transactions are validated by miners who get rewarded

Bitcoin - denominations

DENOMINATION	ABBREVIATION	FAMILIAR NAME	VALUE IN BTC
Satoshi	SAT	Satoshi	0.00000001 BTC
Microbit	μBTC (uBTC)	Microbitcoin or Bit	0.000001 BTC
Millibit	mBTC	Millibitcoin	0.001 BTC
Centibit	cBTC	Centibitcoin	0.01 BTC
Decibit	dBTC	Decibitcoin	0.1 BTC
Bitcoin	BTC	Bitcoin	1 BTC
DecaBit	daBTC	Decabitcoin	10 BTC
Hectobit	hBTC	Hectobitcoin	100 BTC
Kilobit	kBTC	Kilobitcoin	1000 BTC
Megabit	MBTC	Megabitcoin	1000000 BTC

Bitcoin - Monetary Policy

- Digital Currency Regulation
- Demand-supply
- The Halving every 4-year or after 2,10,000 blocks.
- The Block Frequency

Bitcoin - The Halving

What is Bitcoin?

- Bitcoin is a **completely decentralized**, **peer-to-peer**, **permissionless** cryptocurrency put forth in 2009
  - **Completely decentralized:** no central party for ordering or recording anything
  - **Peer-to-peer:** software that runs on machines of all stakeholders to form the system
  - **Permissionless:** no identity; no need to signup anywhere to use; no access control – anyone can participate in any role

* Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008) (https://bitcoin.org/bitcoin.pdf)

The Permission-less Model

- Works in an open environment and over a large network of participants
- The users do not need to know the identity of the peers, and hence the users do not need to reveal their identity to others
- Good for financial applications like banking using cryptocurrency

Privacy - Security

- The system is tamper-proof – it is "**extremely hard**" to make a change in the blockchain
  - Tampering the system becomes harder as the chain grows

- For Bitcoin, the transactions are pseudo-anonymous
  - Transactions are sent to public key addresses,- cryptographically generated addresses, computed by the wallet applications

Peer Addresses

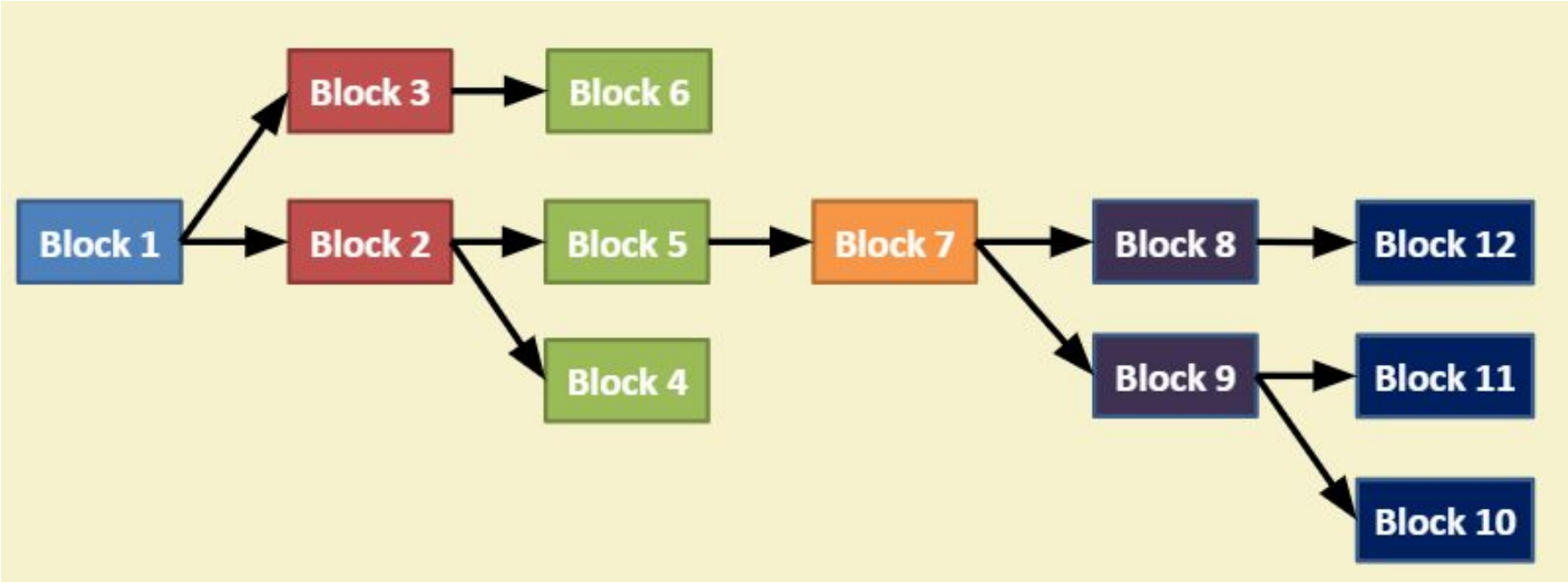
- "**Address**" in bitcoin is synonymous to an **"Account"** in a bank.

- The wallet listens for transactions addressed to an account,
  - Encrypts the transactions by the public key of the target address
  - Only the target node can decrypt the transaction and accept it

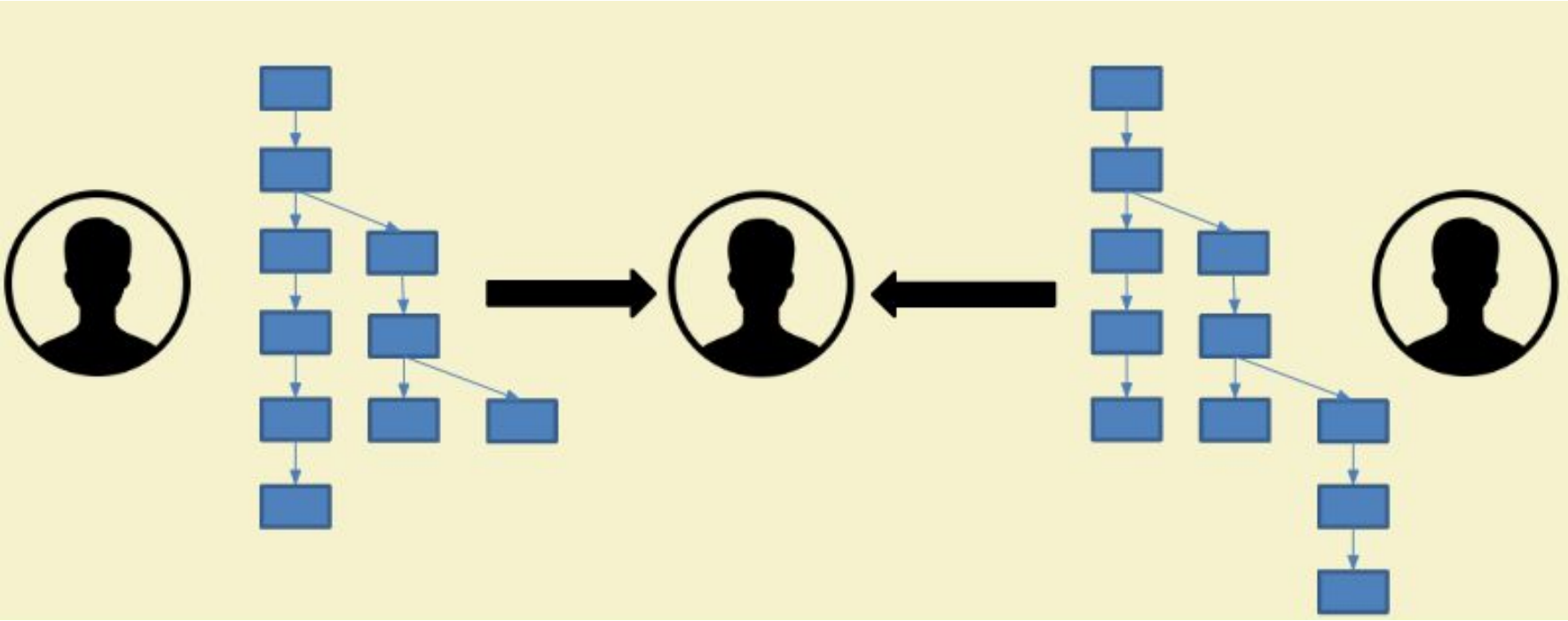
- However, the actual transaction amount is open to all for validation

Blockchain (at Permission-less Model) as a Tree

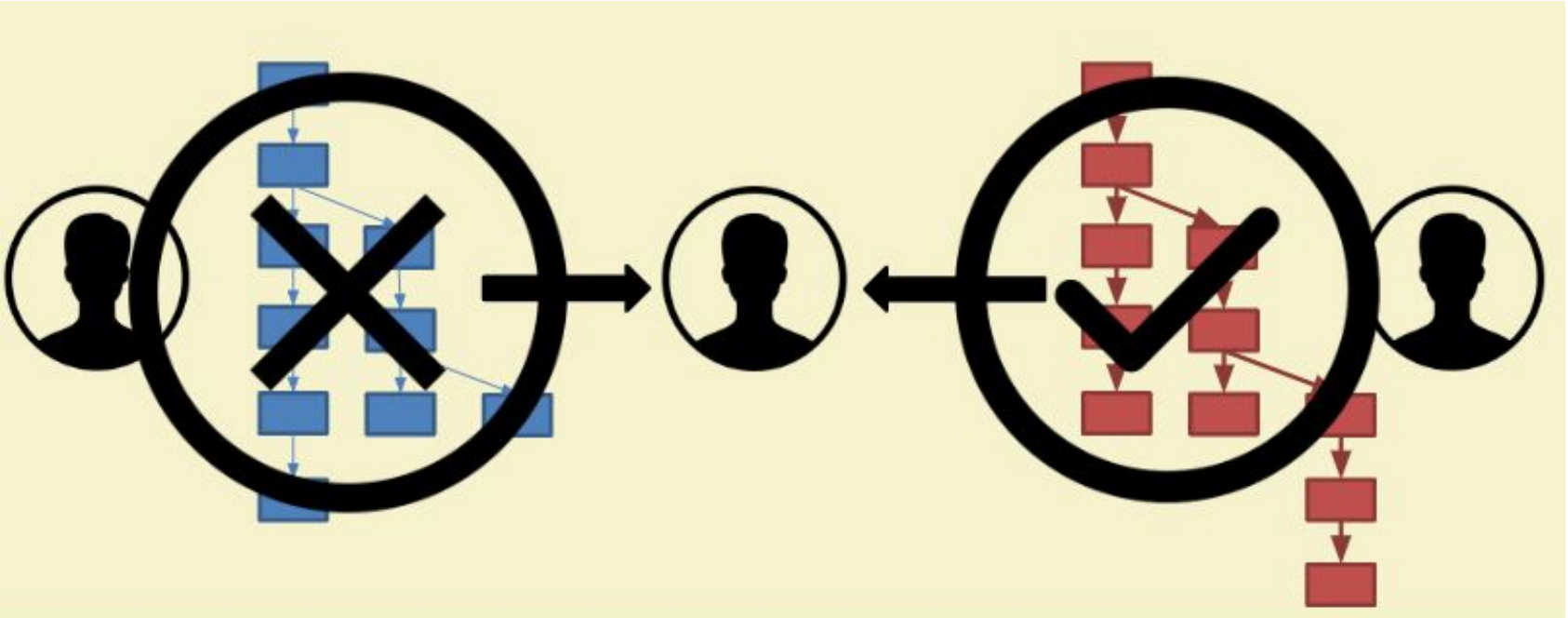
- The longest chain is the accepted chain



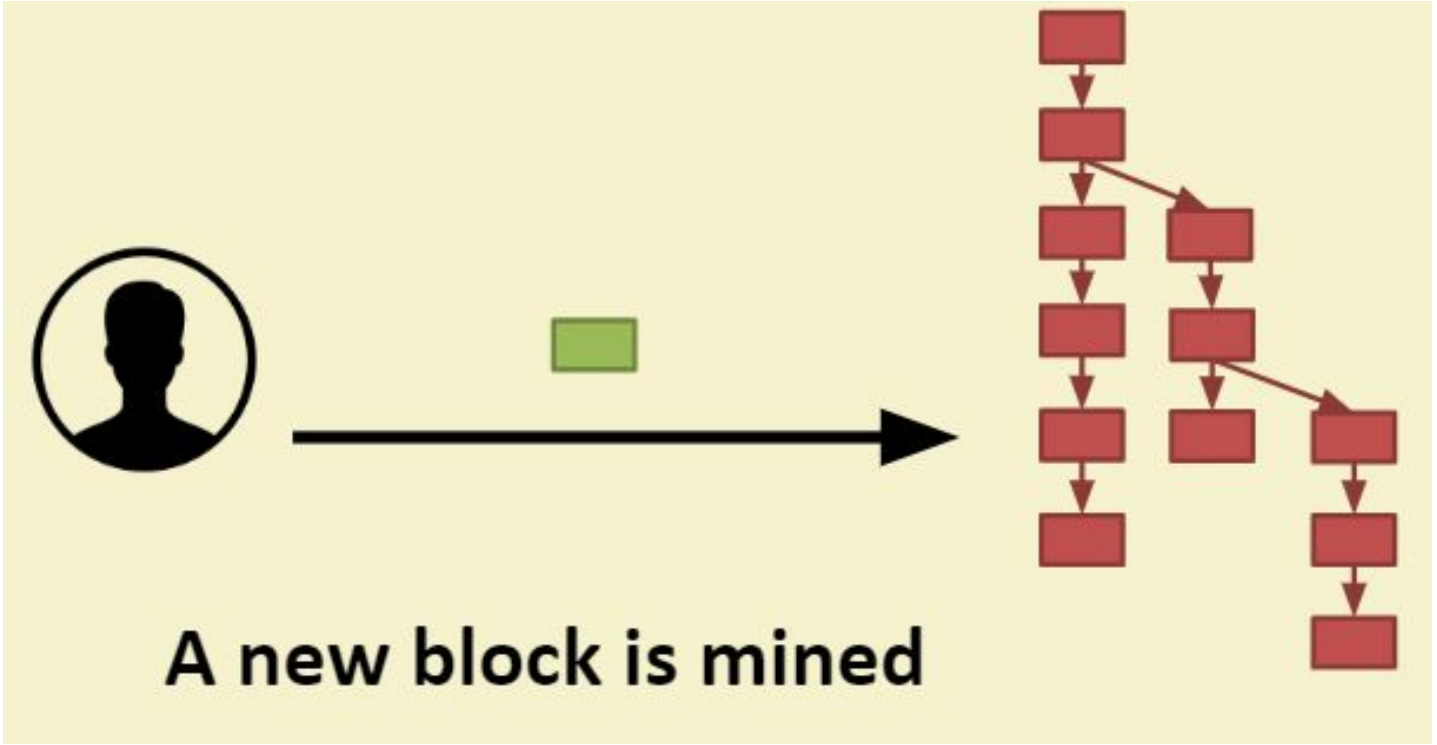
Accepting the Longest Chain



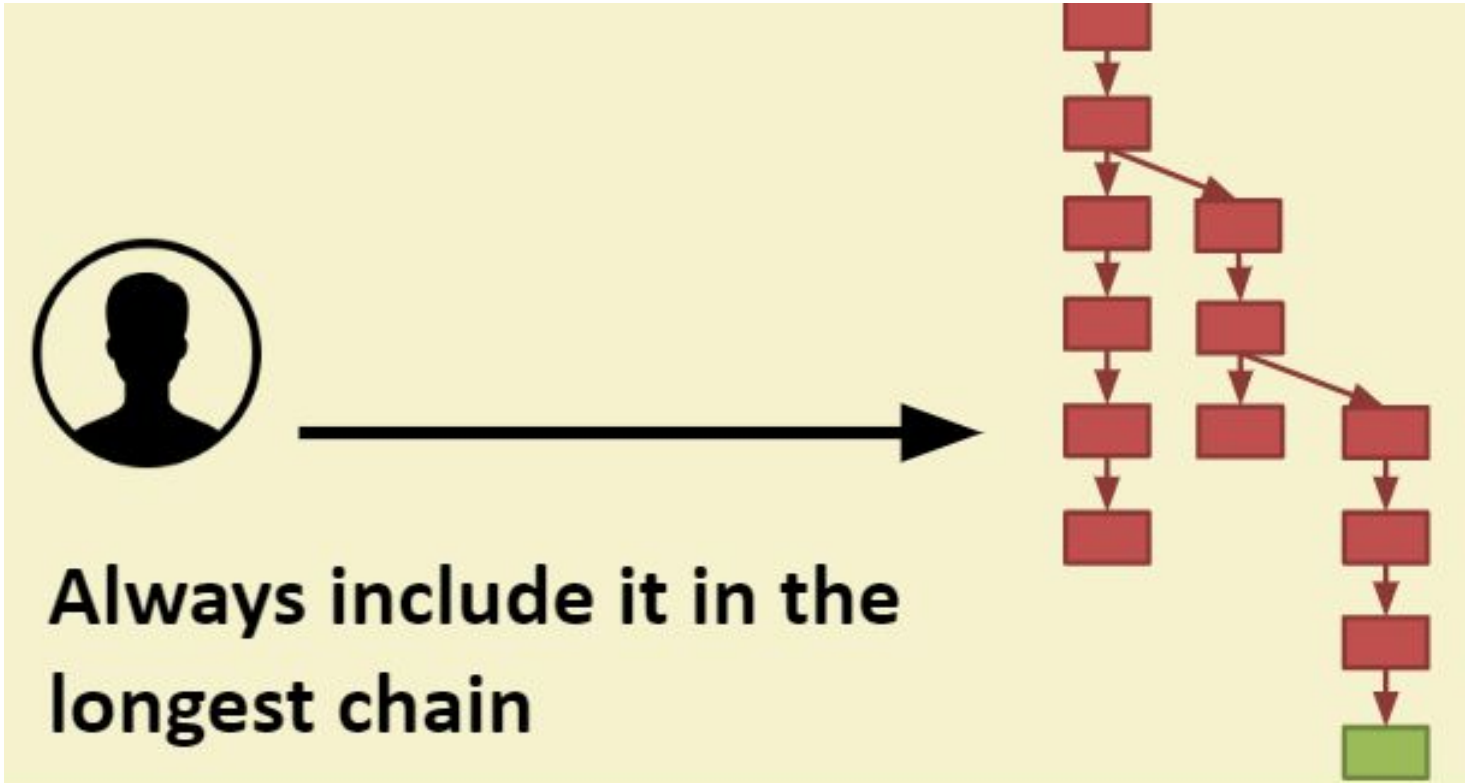
Accepting the Longest Chain



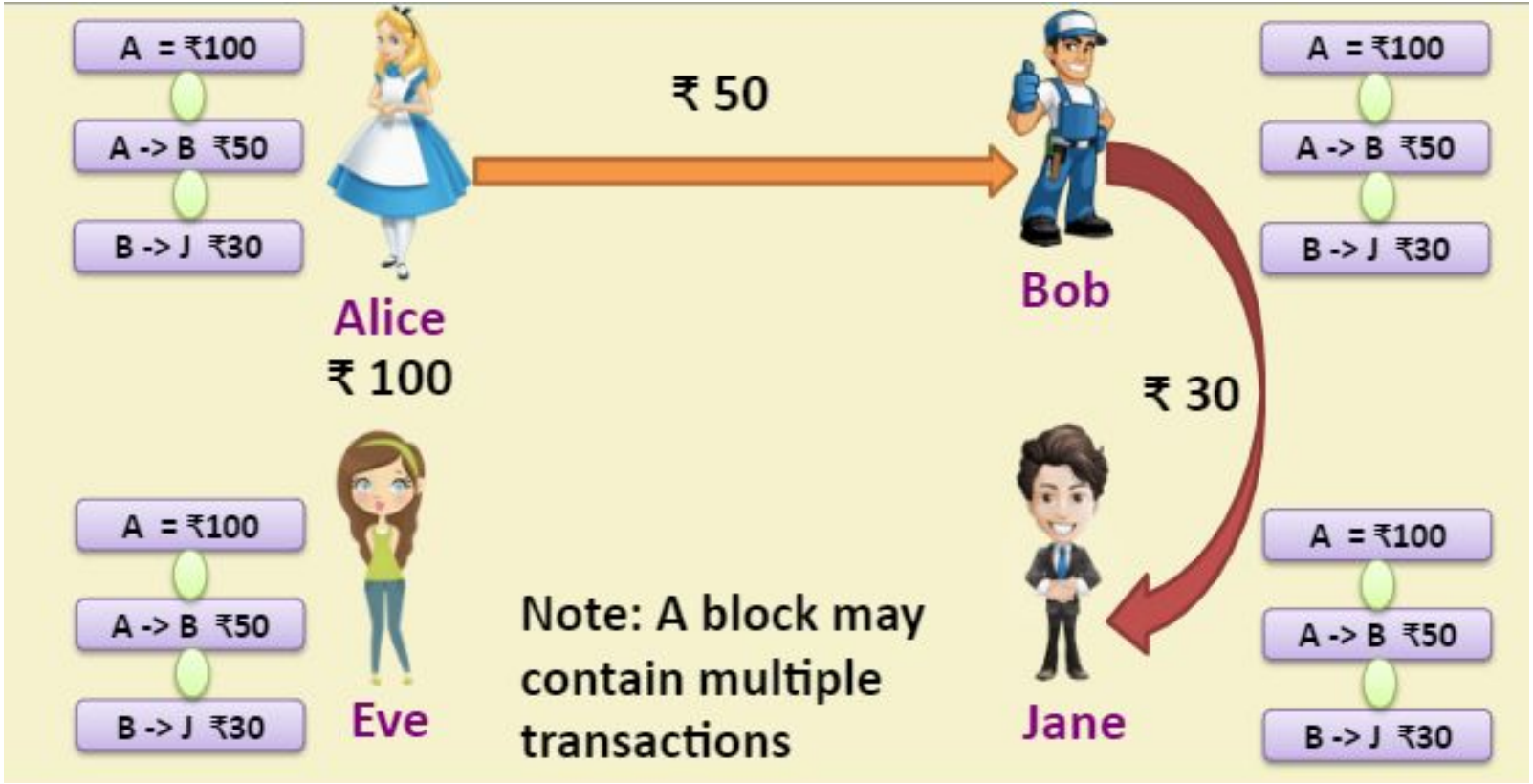
Accepting the Longest Chain



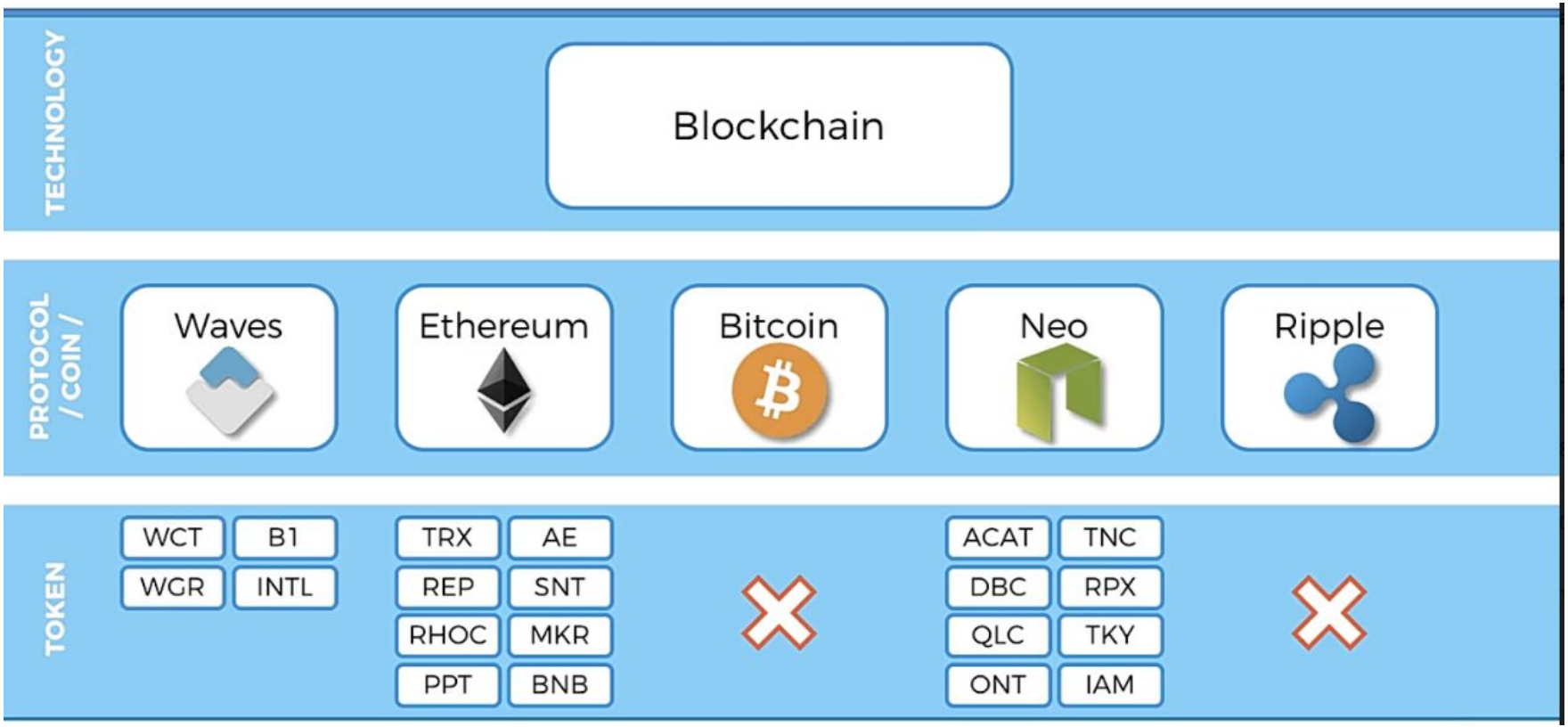
Accepting the Longest Chain



Technology behind Bitcoin - Blockchain

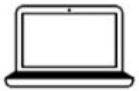
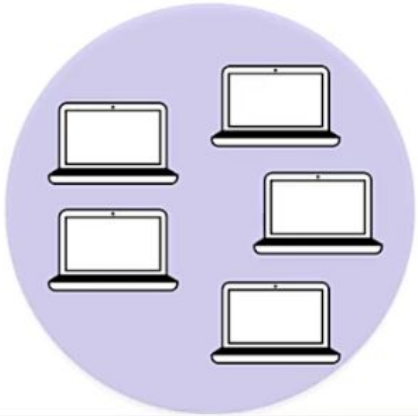


What is Bitcoin?



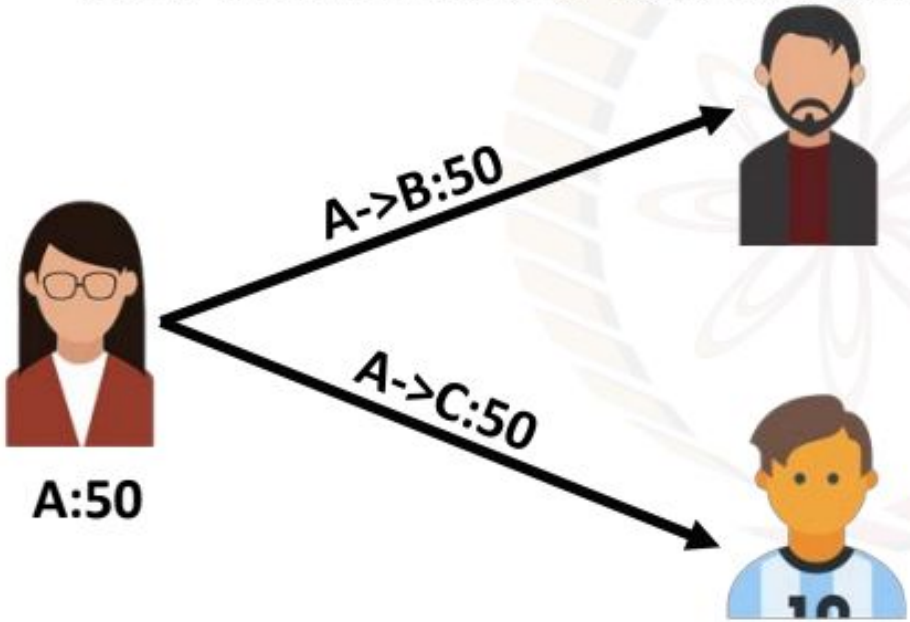
What is Bitcoin?

The Bitcoin Ecosystem:

- Nodes 
- Large Mines 
- Miners 
- Mining Pools 

Bitcoin Basics - Double Spending

- Same bitcoin is used for more than one transactions



- In a centralized system, the bank prevents double spending
- **How can we prevent double spending in a decentralized network?**

Bitcoin Basics - Handling Double Spending

- Details about the transaction are sent and forwarded to all or as many other computers as possible
- Use Blockchain – a constantly growing chain of blocks that contain a record of all transactions
- The blockchain is maintained by all peers in the Bitcoin network – everyone has a copy of the blockchain

Bitcoin Basics - Handling Double Spending

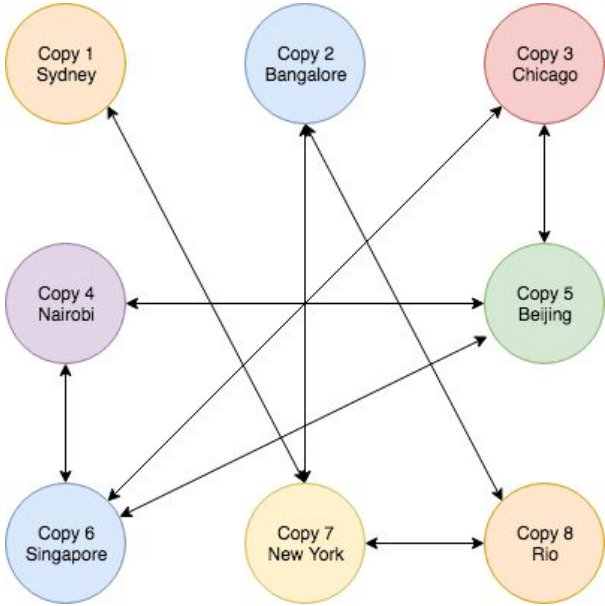
- To be accepted in the chain, transaction blocks must be valid and must include proof of work – a computationally difficult hash generated by the mining procedure
- Blockchain ensures that, if any of the block is modified, all following blocks will have to be recomputed

Bitcoin Basics - Handling Double Spending

- When multiple valid continuation to this chain appear, only the longest such branch is accepted and it is then extended further (longest chain)
- Once a transaction is committed in the blockchain, everyone in the network can validate all the transactions by using Alice's public address
- The validation prevents double spending in bitcoin

What is Ethereum?

- a **public, blockchain based distributed computing platform.**
- as **one big computer** made up of small computers around the world.
- Eg: One can write applications and run them on this global computer.
- The platform **guarantees that your application will always run without any downtime, censorship, fraud or third-party interference.**
- Ethereum blockchain can **also transfer money between 2 parties without a central authority**



What is Ethereum?

- A Blockchain network that introduced a built-in **Turing-complete programming language** that can be **used for creating** various decentralized applications(also called **Dapps**).
- Ethereum network is fueled by its own **cryptocurrency called 'ether'.**
- Allow the implementation of **smart contracts.** Smart contracts can be thought of as 'cryptographic bank lockers' which contain certain values.
- These cryptographic lockers can only be unlocked when certain conditions are met.
- Ethereum is a network that can be applied to various other sectors.
- called **Blockchain 2.0** → it proved that blockchain can be used beyond the financial sector.
- The consensus mechanism used in Ethereum is **Proof of Stakes(PoS)**, which is **more energy efficient** than, Proof of Work(PoW).
- PoS depends on the **amount of stake a node holds**.

History of Ethereum

2013

- Ethereum was **first described in Vitalik Buterin's white paper**
- Goal of **developing decentralized applications.**

2014:

- **EVM was specified in a paper by Gavin Wood**, and the formal development of the software also began.

2015:

- Ethereum **created its genesis block** marking the **official launch of the platform.**

2018:

- Ethereum **took second place** in Bitcoin **in terms of market capitalization.**

2021:

- a major network upgrade named **London included Ethereum improvement proposal 1559**
- **introduced a mechanism** for **reducing transaction fee volatility.**

2022:

- Ethereum has **shifted from PoW(Proof-of-Work) to PoS(Proof-of-State)**
- Also known as **Ethereum Merge**.
- It has **reduced Ethereum's energy consumption by ~ 99.95%.**

Features of Ethereum

1. Smart contracts:
   a. Ethereum allows the creation and deployment of smart contracts.
   b. Smart contracts are created mainly using a programming language called **solidity.**
2. Ethereum Virtual Machine (EVM):
   a. **a runtime environment for compiling and deploying Ethereum-based smart contracts.**
3. Ether:
   a. **cryptocurrency of the Ethereum network**.
   b. **only acceptable form of payment for transaction fees** on the Ethereum network.
4. Decentralized applications (Daaps):
   a. **has its backend code running on a decentralized peer-to-peer network**.
   b. Has a frontend and UI to make calls and query data from its backend.
   c. They **operate on Ethereum** and **perform the same function irrespective of the environment** in which they get executed.
5. Decentralized autonomous organizations (DAOs):
   a. **works in a democratic and decentralized** fashion.
   b. **relies on smart contracts for decision-making** within the organization.

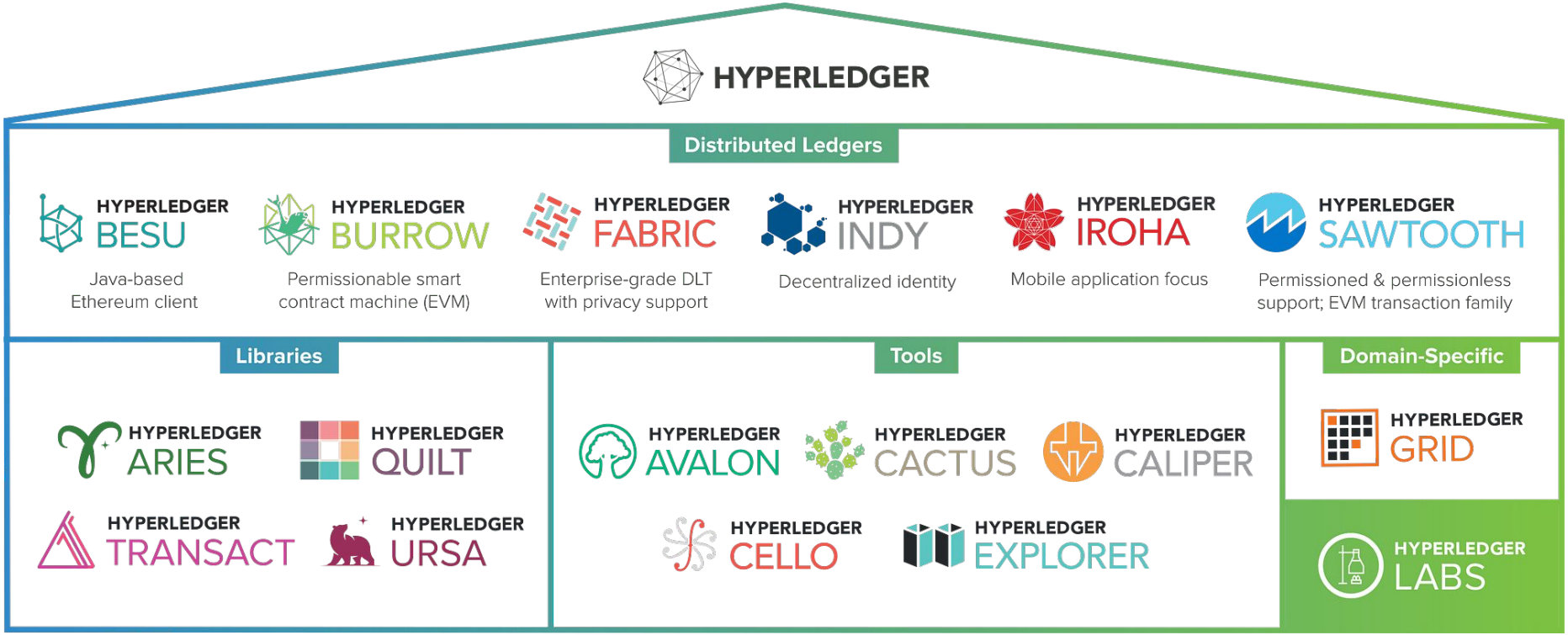
Real-World Applications of Ethereum

1. **Voting**:
   a. Voting systems are **adopting Ethereum.**
   b. The **results of polls are available publicly, ensuring a transparent fair system thus eliminating voting malpractices.**
2. **Agreements**:
   a. With Ethereum smart contracts, agreements and contracts **can be maintained and executed without any alteration.**
   b. Ethereum can be **used for creating smart contracts and for digitally recording transactions based on them.**
3. **Banking systems**:
   a. **Due to the decentralized nature** of the Ethereum blockchain it becomes **challenging for hackers to gain unauthorized access to the network**.
   b. **Makes payments on the Ethereum network secure**
4. **Shipping**:
   a. **Ethereum provides a tracking framework** that helps with the **tracking of cargo and prevents goods from being misplaced.**
5. **Crowdfunding**:
   a. **helps to increase trust and information symmetry**.
   b. It **creates many possibilities for startups by raising funds** to create their own digital cryptocurrency.

Hyperledger

- An open-source collaborative effort aimed at advancing cross-industry blockchain technologies.
- Hosted by The Linux Foundation, Hyperledger brings together a diverse community of developers and organizations to collaborate on the development of modular frameworks and tools for enterprise-grade blockchain solutions.
- The project was launched in December 2015 and has since grown to include a variety of blockchain frameworks, libraries, and tools.
- **Key components and features of Hyperledger**
  1. **Collaborative Ecosystem:**
     - Hyperledger fosters a collaborative and vendor-neutral environment, bringing together contributors from various industries, including finance, healthcare, supply chain, and more. It encourages open participation and contributions from individuals and organizations.

Hyperledger



Hyperledger

2. **Modular Frameworks:**
   - Hyperledger provides a range of modular blockchain frameworks, each designed to meet specific enterprise use cases. Notable frameworks include:
     - Hyperledger Fabric: A permissioned blockchain framework with a modular architecture, supporting plug-and-play components for consensus algorithms, membership services, and more.
     - Hyperledger Sawtooth: A modular and scalable framework with a unique consensus algorithm called Proof of Elapsed Time (PoET).
     - Hyperledger Besu: An Ethereum-compatible client designed for enterprise use within the Hyperledger consortium.
     - Hyperledger Indy: Focused on decentralized identity, enabling the creation of self-sovereign identities.

Hyperledger

3. **Permissioned Blockchains:**
    - Hyperledger frameworks are generally designed for permissioned blockchains, where participants are known entities and have defined roles and permissions. This approach is well-suited for enterprise use cases, emphasizing privacy, scalability, and performance.
4. **Smart Contracts and Chaincode**:
    - Hyperledger Fabric supports smart contracts, referred to as "chaincode," which are written in languages like Go, JavaScript, or Java. Chaincode allows the execution of business logic on the blockchain.
5. **Consensus Mechanisms:**
    - Hyperledger frameworks offer flexibility in choosing consensus mechanisms. Hyperledger Fabric, for example, supports various consensus algorithms, including Practical Byzantine Fault Tolerance (PBFT), Raft, and others.

Hyperledger

6. **Interoperability and Standards:**
   - Hyperledger emphasizes interoperability and collaboration between different blockchain networks. It aims to establish common standards to facilitate seamless integration between various Hyperledger frameworks and other blockchain technologies.
7. **Identity and Privacy:**
   - Hyperledger Indy focuses specifically on decentralized identity solutions, providing tools and libraries for creating and managing self-sovereign identities. Privacy and identity protection are critical considerations in Hyperledger projects.
8. **Open Source and Apache 2.0 License:**
   - All Hyperledger projects are released under the open-source Apache 2.0 license, promoting transparency, accessibility, and collaboration within the community.

Hyperledger

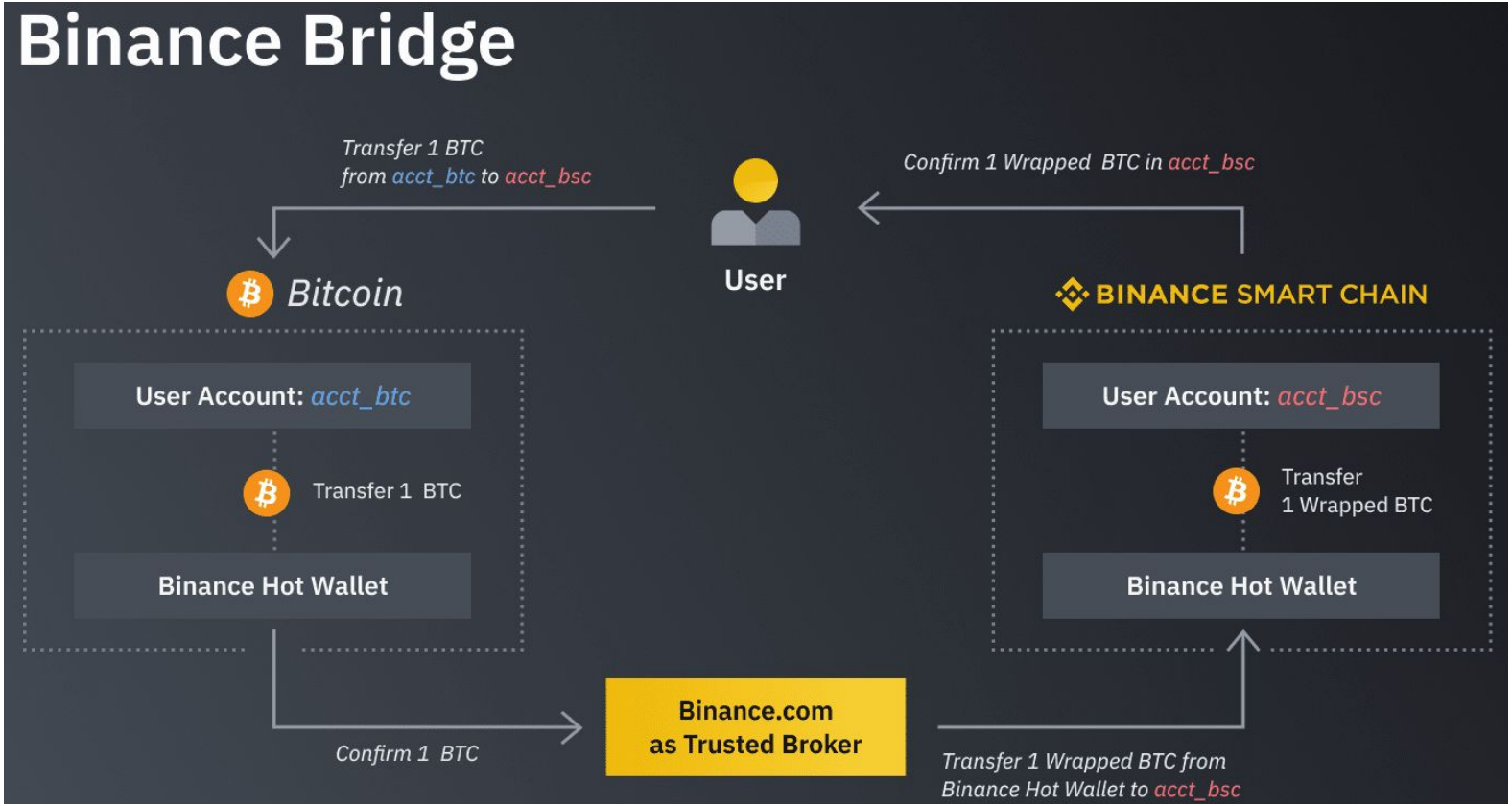
9. **Enterprise Adoption:**
    - Hyperledger frameworks are designed with enterprise requirements in mind, such as scalability, performance, and compliance with regulatory standards. This focus has led to widespread adoption in industries seeking to leverage blockchain technology for various use cases.
10. **Education and Community Support:**
    - Hyperledger offers educational resources, including documentation, tutorials, and training programs, to help developers and organizations understand and implement blockchain solutions using Hyperledger frameworks.

Introduction to Binance Smart Chain

- Launched by Binance in September 2020
- Parallel blockchain to Binance Chain
- Supports smart contracts and dApps
- EVM-compatible (Ethereum Virtual Machine)
- Designed for high performance and low fees
- High transaction speed (~3 second block time)
- Low transaction fees
- Ethereum compatibility
- Growing DeFi and NFT ecosystem
- Strong exchange backing (Binance)

Courtesy : BNB Chain

Introduction to Binance Smart Chain



Courtesy : BNB Chain

Binance Smart Chain - Ecosystem

- Dual-chain architecture: Binance Chain + BSC
- Binance Chain: Fast trading
- BSC: Smart contracts and DeFi
- Cross-chain compatibility
- Proof of Staked Authority (PoSA) consensus

Binance Smart Chain - Consensus Mechanism : PoSA

- Combination of Delegated Proof of Stake and Proof of Authori
- 21 active validators
- Validators selected based on staking and reputation
- Fast block confirmation
- Energy-efficient compared to PoW

Binance Smart Chain

BNB Token Utility in BSC

- Used for transaction fees (gas)
- Staking and validator participation
- Governance participation
- Utility in DeFi, NFTs, and gaming

BSC - Key Features

- EVM compatibility
- Smart contract support (Solidity)
- Low gas fees
- Cross-chain asset transfers
- High throughput

Binance Smart Chain - Usecases

- **Decentralized Finance (DeFi):** BSC hosts a massive DeFi ecosystem, including PancakeSwap, Venus, and various yield farming protocols that thrive on low fees.
- **Non-Fungible Tokens (NFTs):** Popular for minting and trading NFTs due to cost-effective transactions.
- **Blockchain Gaming (GameFi):** High speed and low latency make it ideal for building decentralized games and in-game asset economies.
- **Token Launches (IDOs):** Utilized for Initial DEX Offerings (IDOs) and token sales through platforms like Binance Launchpad.
- **Stablecoins and Assets:** Supports pegged tokens, bringing assets like BTC and ETH onto the BSC network.
- **Governance and Staking:** BNB holders can stake tokens to secure the network, act as validators, and participate in community governance

BSC Vs Ethereum

Parameter	Binance Smart Chain (BSC)	Ethereum
Launch Year	2020	2015
Founder	Binance	Vitalik Buterin
Native Token	BNB	ETH
Consensus Mechanism	Proof of Staked Authority (PoSA)	Proof of Stake (PoS)
Number of Validators	~21 validators	Thousands of validators
Level of Decentralization	Moderate (more centralized)	High (highly decentralized)
Block Time	~3 seconds	~12 seconds
Transaction Speed (TPS)	Higher	Lower (Layer 1)
Gas Fees	Very low	Higher (depends on network congestion)

Courtesy : BNB Chain

BSC Vs Ethereum

Parameter	Binance Smart Chain (BSC)	Ethereum
Smart Contract Support	EVM-compatible	EVM-native
Security Level	Good but relatively lower due to validator concentration	Very high due to large validator network
Ecosystem Size	Growing DeFi ecosystem	Largest DeFi & NFT ecosystem
Popular dApps	PancakeSwap, Venus	Uniswap, Aave, OpenSea
Scalability Approach	High base-layer throughput	Layer 2 solutions (Arbitrum, Optimism, etc.)
Enterprise Adoption	Limited	Strong institutional and enterprise adoption
Best Suited For	Low-cost retail DeFi & high-volume transactions	Secure, large-scale, institutional-grade applications

Courtesy : BNB Chain

Binance Smart Chain

Advantages

- Scalability and speed
- Cost efficiency
- Developer-friendly (Solidity support)
- Large ecosystem support
- Strong liquidity via Binance

Challenges & Criticism

- Concerns about centralization
- Security vulnerabilities in DeFi projects
- Regulatory scrutiny
- Validator concentration risk

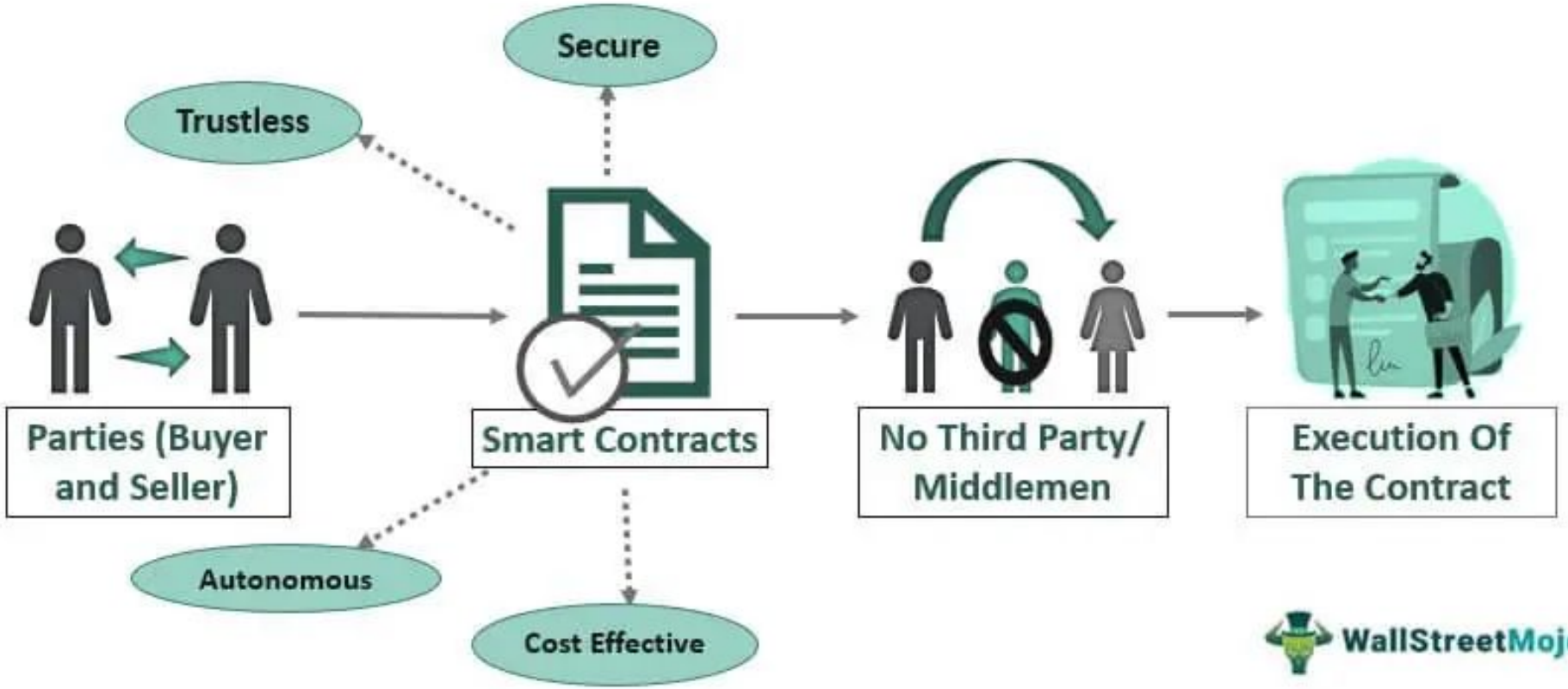
Courtesy : BNB Chain

4. Smart Contract

- A **self-executing program** that automates the actions required in an agreement or contract. Once completed, the transactions are trackable and irreversible.
- Smart contracts **permit trusted transactions and agreements to be carried out among disparate, anonymous parties** without the need for a central authority, legal system, or external enforcement mechanism.
- **do not contain legal language**, terms, or agreements
- **Only code that executes actions when specified conditions** are met.
- Nick Szabo, an American computer scientist
  - invented a virtual currency called **"Bit Gold"** in 1998,
  - defined smart contracts as computerized transaction protocols that execute the terms of a contract

Courtesy : Investopedia

What is a Smart Contract ?



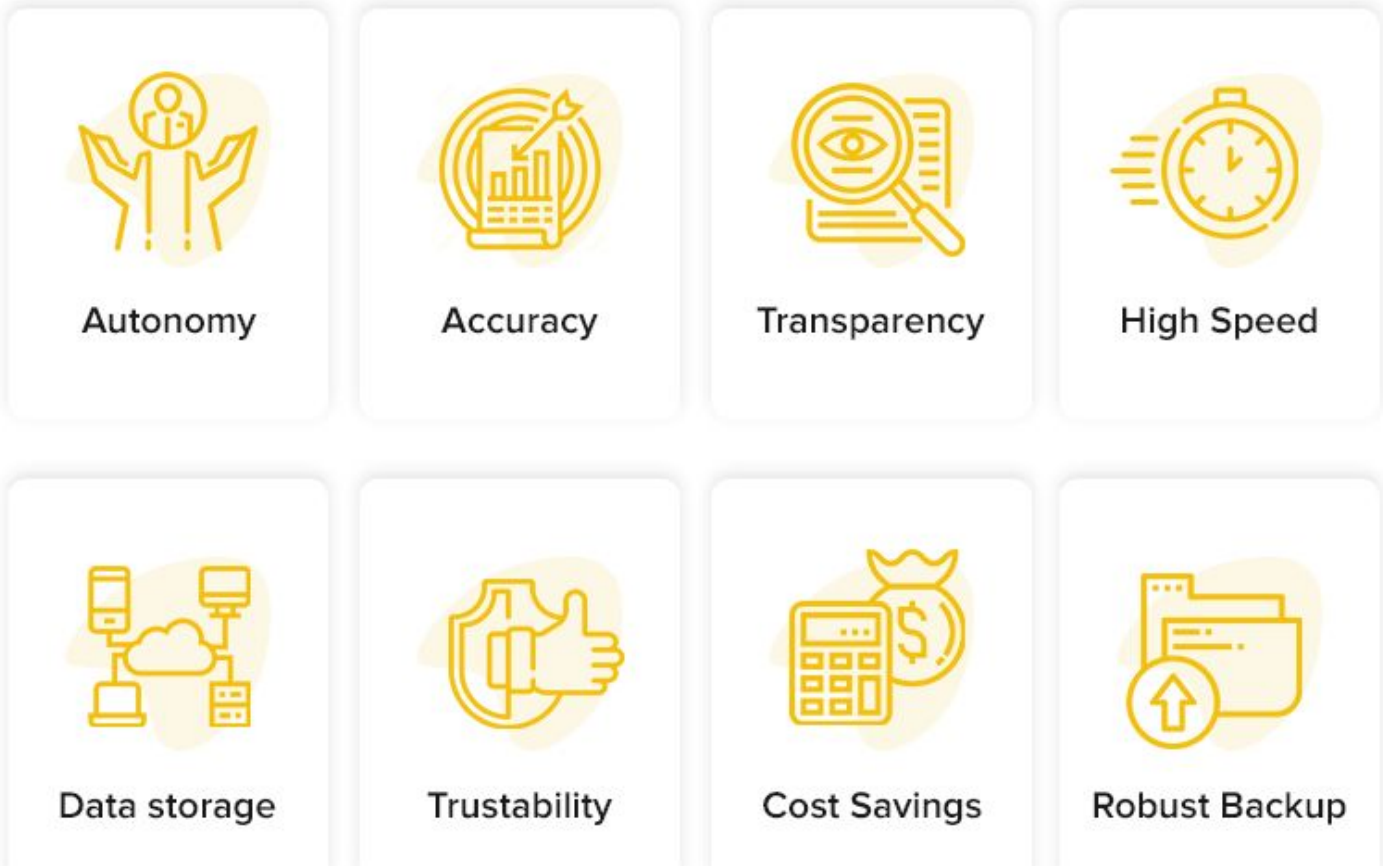
Courtesy : WallStreetMojo

Features of Smart Contracts

1. **Distributed:** Everyone on the network is guaranteed to have a copy of all the conditions of the smart contract and they cannot be changed by one of the parties. A smart contract is replicated and distributed by all the nodes connected to the network.
2. **Deterministic**: Smart contracts can only perform functions for which they are designed only when the required conditions are met. The final outcome will not vary, no matter who executes the smart contract.
3. **Immutable:** Once deployed smart contract cannot be changed, it can only be removed as long as the functionality is implemented previously.
4. **Autonomy:** There is no third party involved. The contract is made by you and shared between the parties. No intermediaries are involved which minimizes bullying and grants full authority to the dealing parties. Also, the smart contract is maintained and executed by all the nodes on the network, thus removing all the controlling power from any one party's hand.
5. **Customizable**: Smart contracts have the ability for modification or we can say customization before being launched to do what the user wants it to do.
6. **Transparent:** Smart contracts are always stored on a public distributed ledger called blockchain due to which the code is visible to everyone, whether or not they are participants in the smart contract.
7. **Trustless:** These are not required by third parties to verify the integrity of the process or to check whether the required conditions are met.
8. **Self-verifying:** These are self-verifying due to automated possibilities.
9. **Self-enforcing:** These are self-enforcing when the conditions and rules are met at all stages.

Courtesy : GeeksforGeeks

Benefits of Smart Contracts



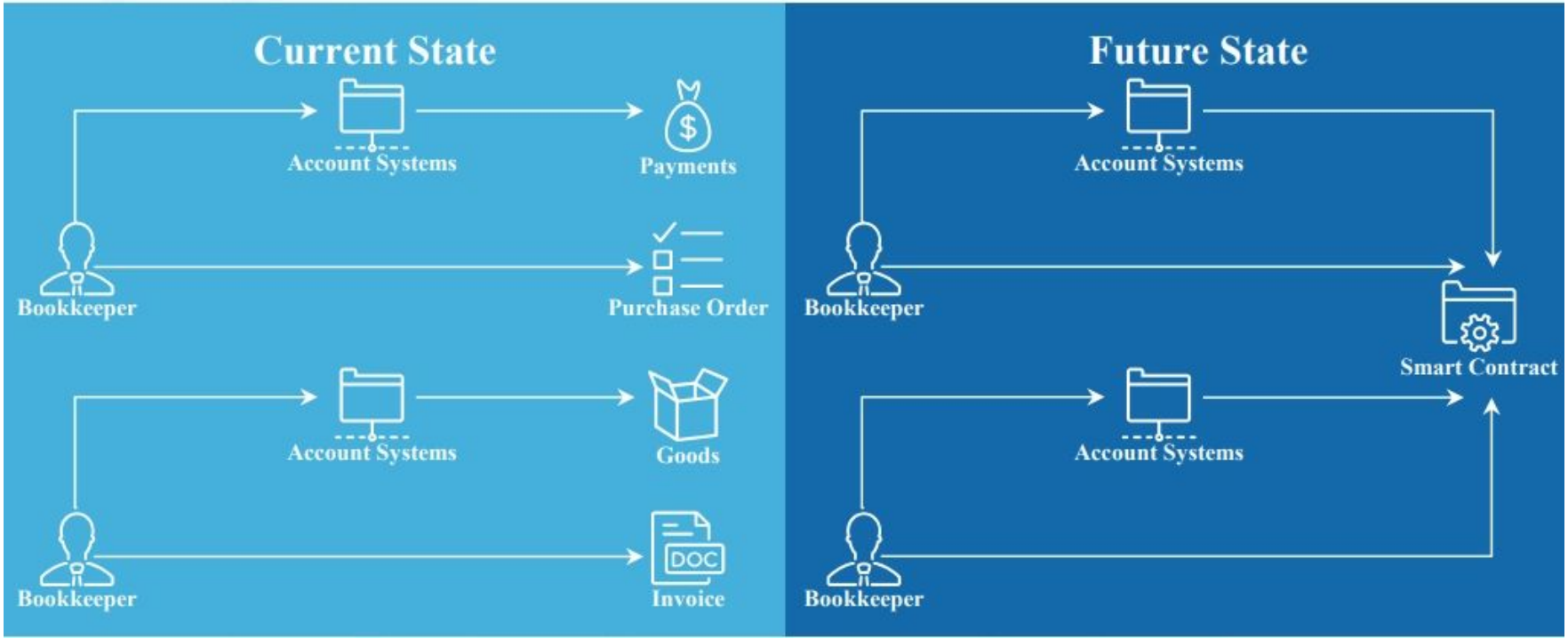
Courtesy : Appinventiv

Types of Smart Contract

1. **Smart Legal Contracts**
   - Legally enforceable
   - Require the parties to fulfill their contractual obligations.
   - Failure to do so may result in strict legal actions against them.
2. **Decentralized Autonomous Organizations (DAO)**
   - These are blockchain communities that are bound to specific rules coded into blockchain contracts combined with governance mechanisms.
   - Any action taken by the community members gets replaced by a self-enforcing code.
3. **Application Logic Contracts (ALC)**
   - contain an application-based code that remains in sync with other blockchain contracts.
   - It enables communication across different devices, such as the merger of the Internet of Things with blockchain technology.

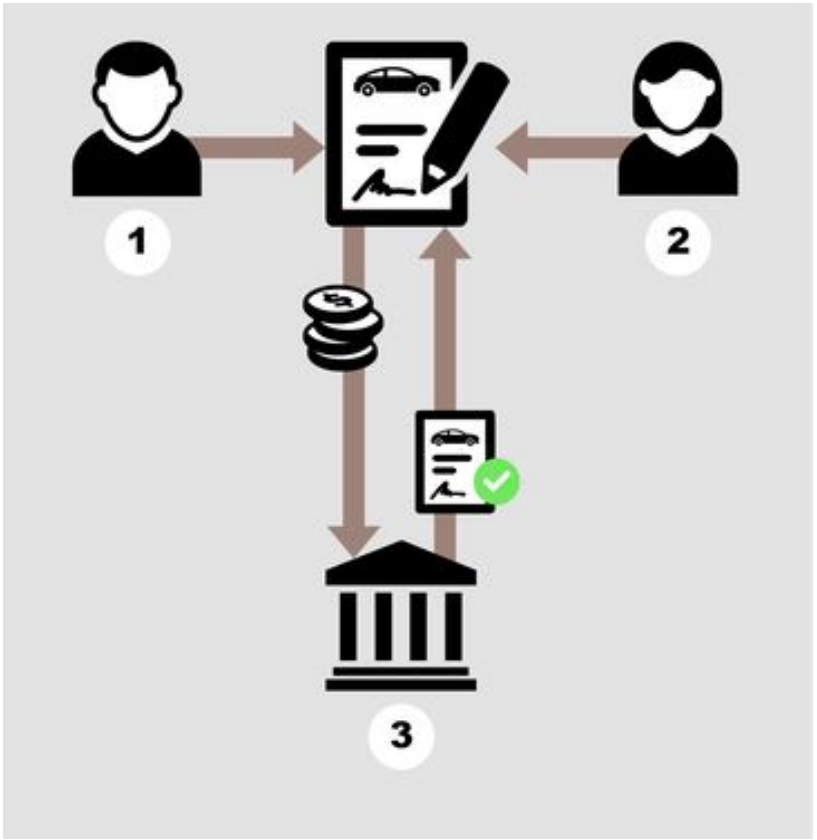
Courtesy : WallStreetMojo

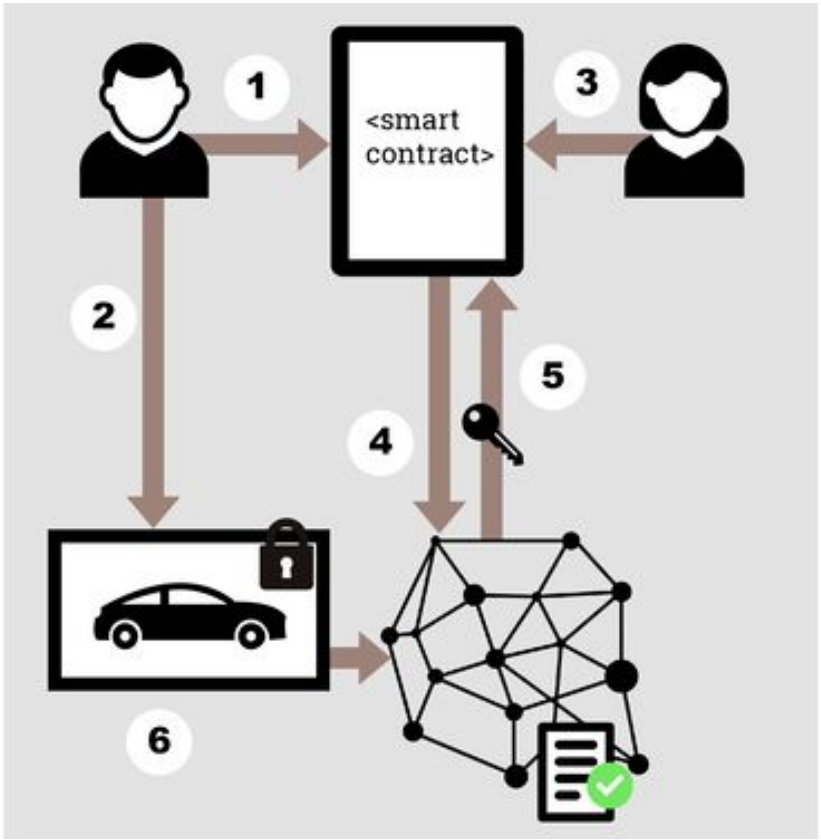
Future of Financial Reporting



Courtesy : AI Multiple

Traditional Contracts Vs Smart Contracts





Courtesy : LinkedIn - Marvin Fedler

Traditional Contracts Vs Smart Contracts - Explained

Traditional Contracts

1. Bob would like to sell his car.
2. Alice would like to buy a car.
3. A third party (intermediary) enables the trust that is needed in order to transfer the ownership of the car. Mostly different intermediaries are needed: motor vehicle registration authority, notary, insurance company. All middlemen take fees.

Courtesy : LinkedIn - Marvin Fedler

Traditional Contracts Vs Smart Contracts - Explained

Smart Contracts

1. Bob would like to sell his car. He defines in a smart contract the conditions by which he will sell the car and signs the contract with his private key.
2. Bob leaves his car locked with a smart lock in his garage. The car has its own blockchain address and the smart lock is controlled by a smart contract.
3. Alice would like to buy a car. She finds Bobs car on an internet platform and signs the smart Bob's contract with her private key. She adds the agreed amount from her blockchain address to Bob's blockchain address.
4. As soon as the smart contract is executed the whole blockchain network will check if Bob is the real owner of the car and if Alice has enough money to buy the car.
5. If all peers in the blockchain network agree on the same state, it means that all conditions in the smart contract are met. The access code for the smart lock will be transferred to Alice and the blockchain address of the car will be registered to Alice. Bob will get the defined amount of money in his blockchain address.
6. Alice will be able to open the smart lock with her private key.

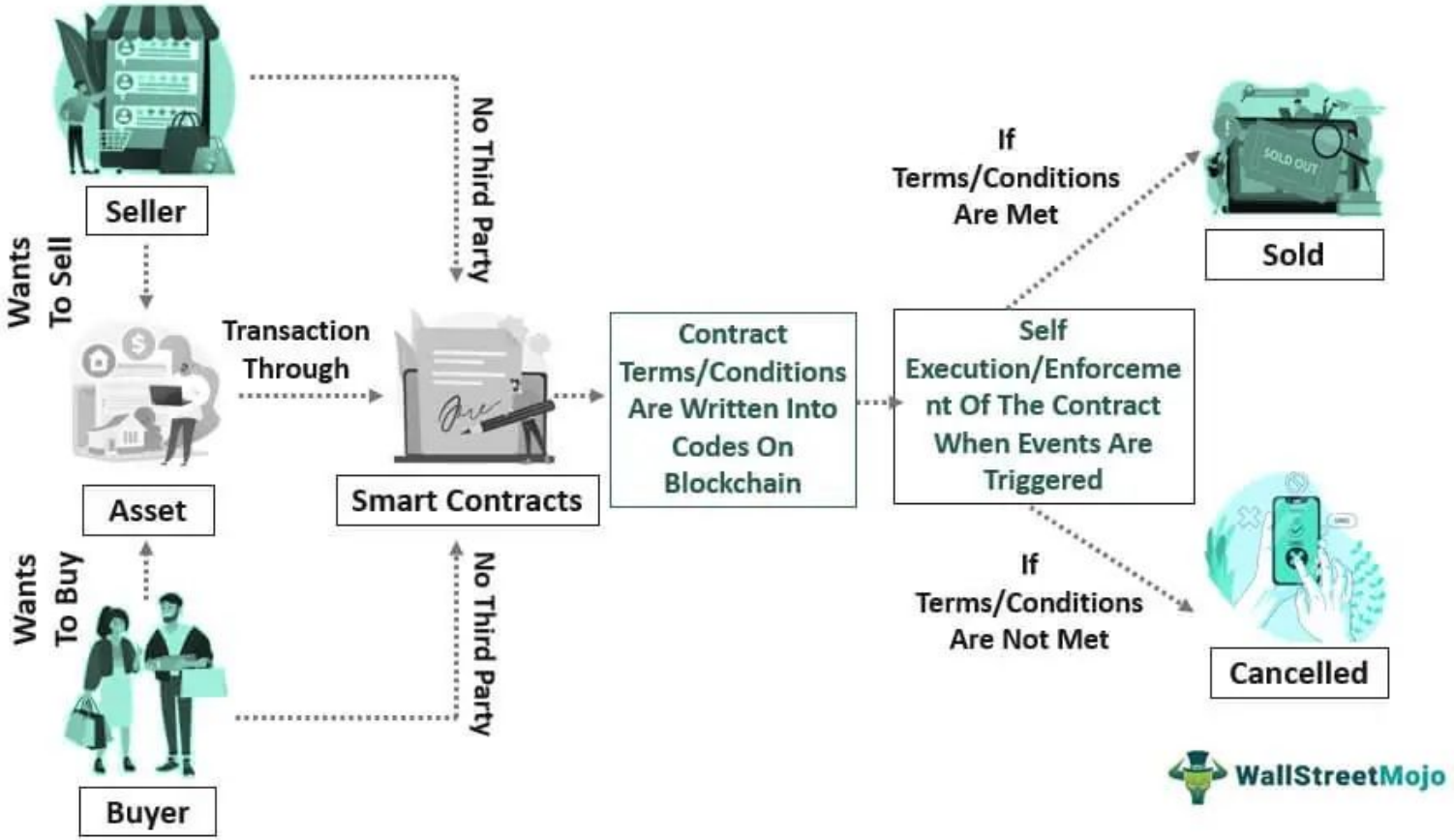
Courtesy : LinkedIn - Marvin Fedler

Future of Financial Reporting

	Written contract	Smart contract
Language/Code	Human language	Machine computer code
Automation	All parts of the agreement	Only transactions to be automated
Recording	Conditions can be written on a paper by the concerned parties	Embedded into blockchain or another ledger
Modification of an internal state	Subject to interpretation	Generally immutable

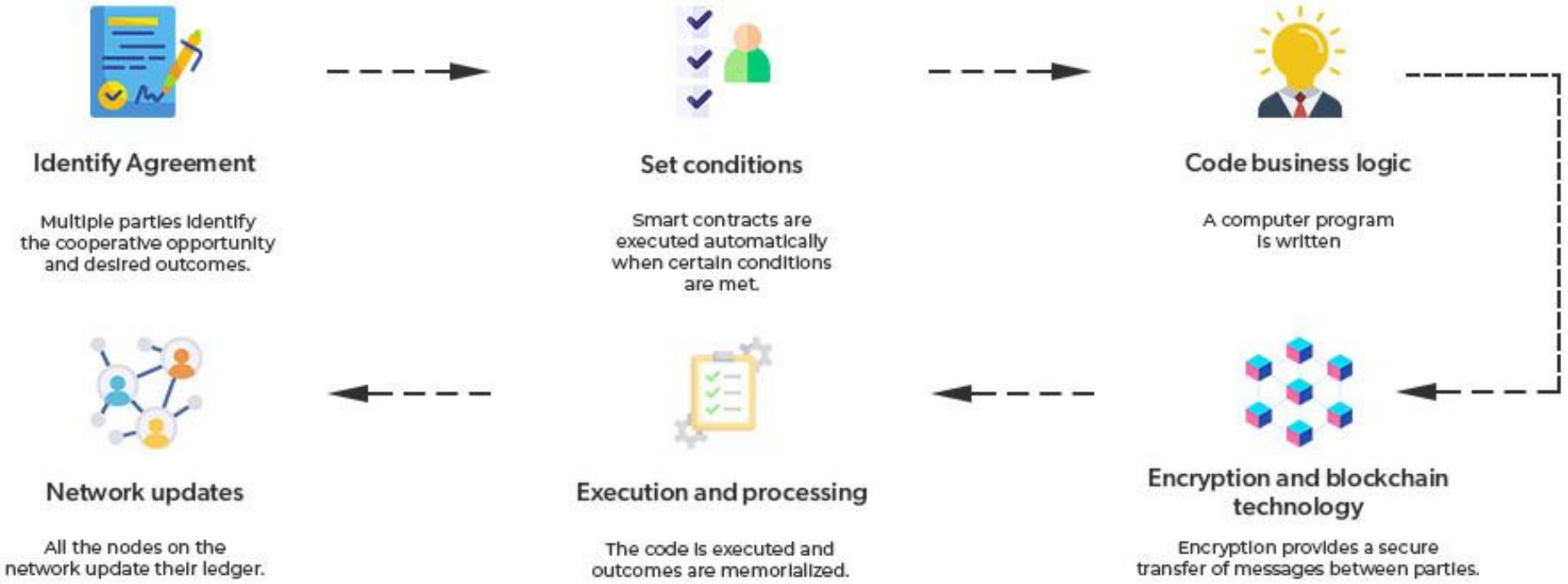
Courtesy : Cointelegraph

Smart Contract Functioning



Courtesy : WallStreetMojo

How do smart contracts work ?



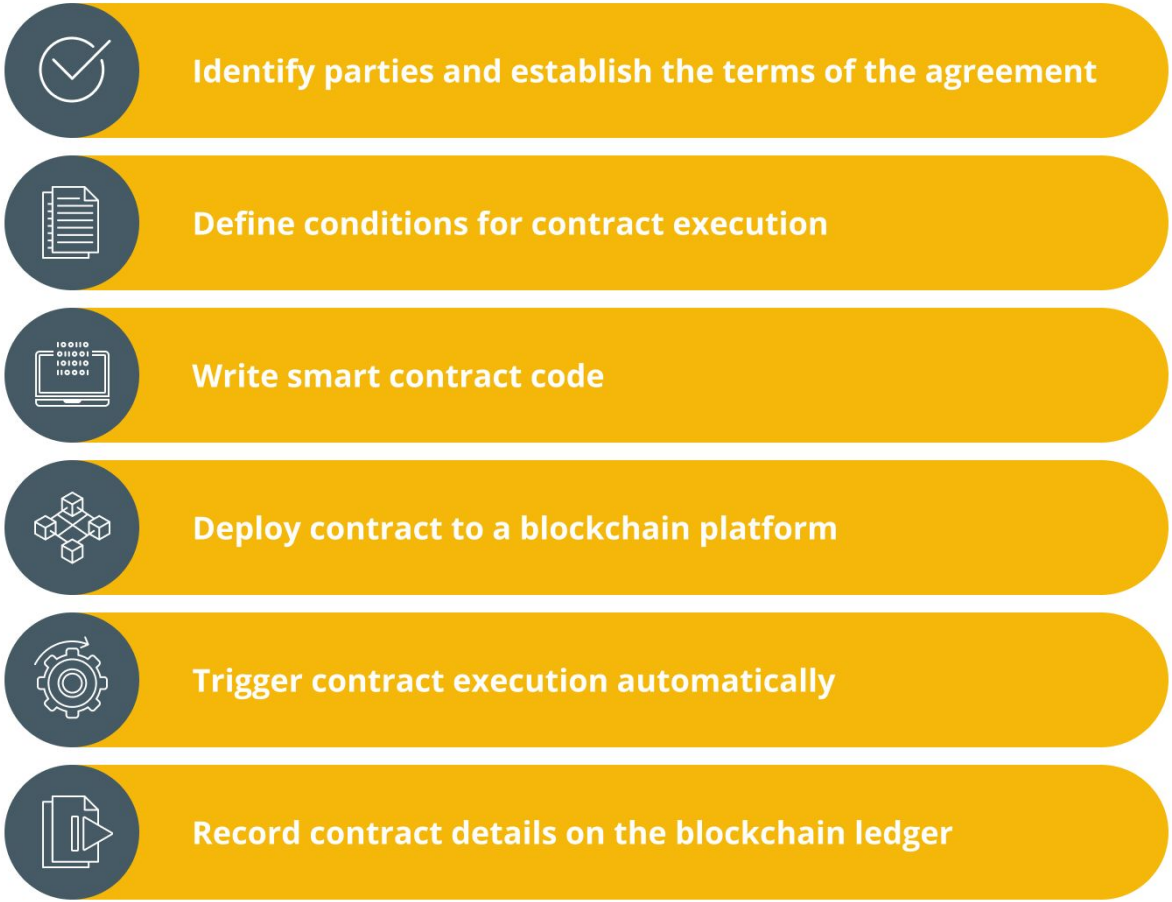
Courtesy : GeeksforGeeks

How do smart contracts work ?

1. **Identify Agreement:** Multiple parties identify the cooperative opportunity and desired outcomes and agreements could include business processes, asset swaps, etc.
2. **Set conditions:** Smart contracts could be initiated by parties themselves or when certain conditions are met like financial market indices, events like GPS locations, etc.
3. **Code business logic:** A computer program is written that will be executed automatically when the conditional parameters are met.
4. **Encryption and blockchain technology:** Encryption provides secure authentication and transfer of messages between parties relating to smart contracts.
5. **Execution and processing:** In blockchain iteration, whenever consensus is reached between the parties regarding authentication and verification then the code is executed and the outcomes are memorialized for compliance and verification.
6. **Network updates:** After smart contracts are executed, all the nodes on the network update their ledger to reflect the new state. Once the record is posted and verified on the blockchain network, it cannot be modified, it is in append mode only.

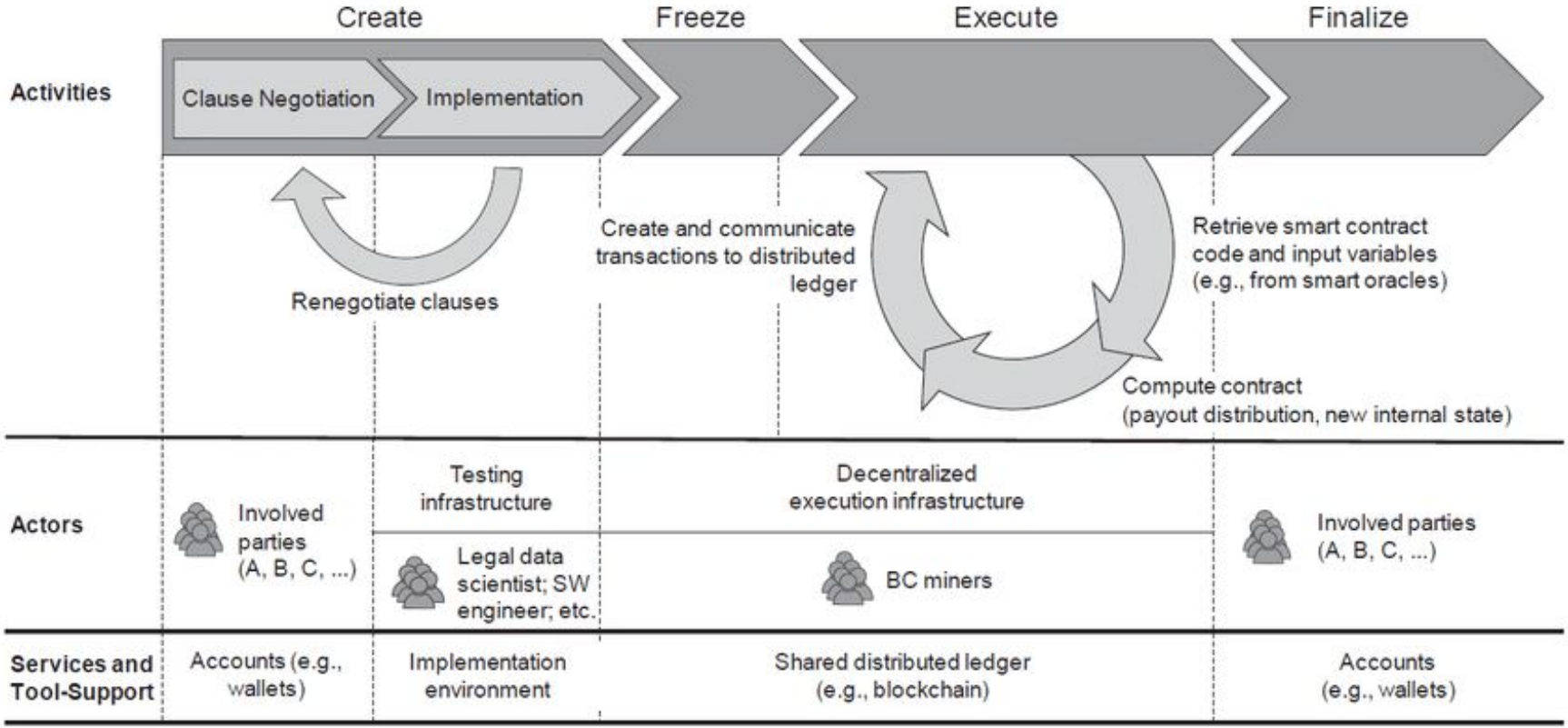
Courtesy : GeeksforGeeks

Steps involved in the Smart Contracts



Courtesy : Cointelegraph

Life Cycle of a Smart Contract



Courtesy : ResearchGate Paper

Life Cycle of a Smart Contract

1. **Creation Phase:**

- consists of iterative contract negotiation and an implementation phase.
- First, the parties must agree on the broad content and goals of the contract.
- Similar to typical contract negotiations and can be conducted online or in person contracts
- During this phase, the following tasks are completed:
    1. Multiple-party bargaining.
    2. Design, implementation, and validation of smart

2. **Freeze:**

- The validation of transactions on a blockchain is performed by a network of computers known as nodes. The blockchain miners are these nodes.
- To prevent the ecosystem from being swamped with smart contracts, miners must be paid a tiny fee in exchange for this service.

Courtesy : Medium

Life Cycle of a Smart Contract

3. **Execution:**

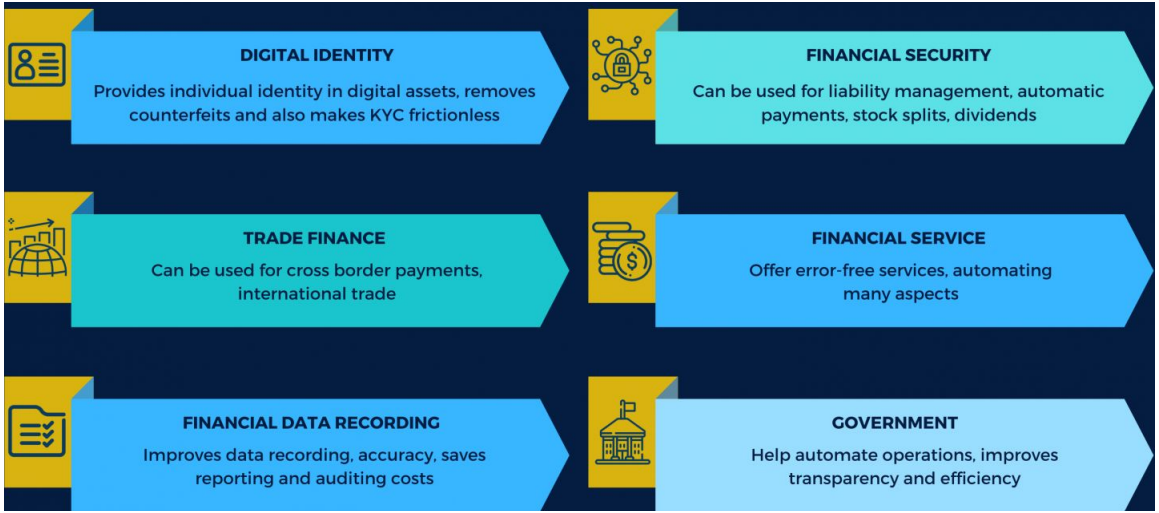
- Contracts placed on the distributed ledger are read by participating nodes.
- The authentication nodes validate the integrity of a smart contract,
- the code is performed by the smart contract interference engine (or by the compiler).
- When one party's inputs for execution are received in the form of coins (commitment to goods via coins), the interference engine **generates a transaction triggered by the met criterion**.
- The execution of the smart contract **results in a new set of transactions and a new state for the smart contract.**
- The **discoveries and new state information** are entered into the distributed ledger and validated using the consensus procedure.

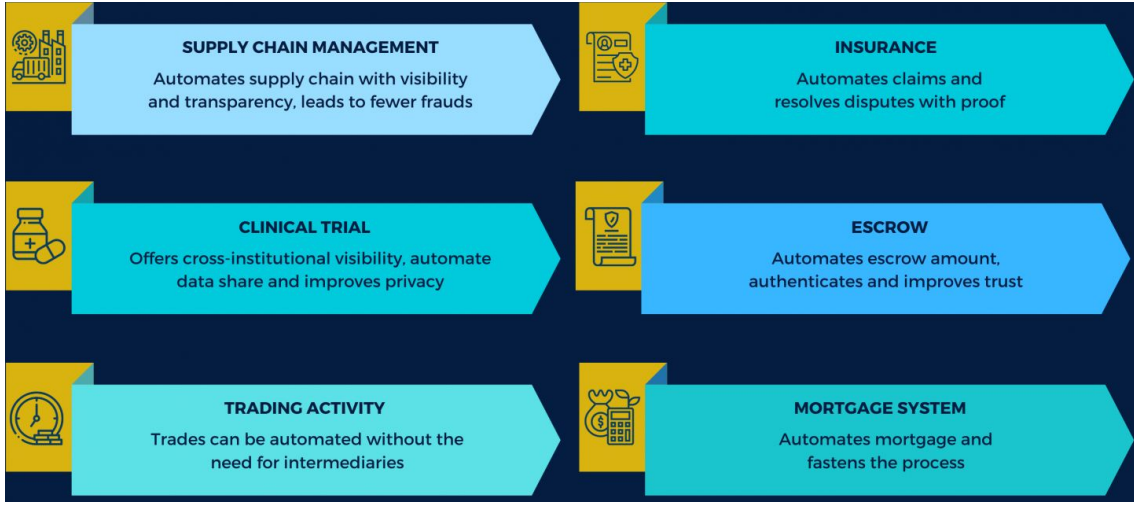
4. **Finalize:**

- the resulting transactions and updated state information are recorded in the distributed ledger and confirmed via the consensus process.
- The previously pledged digital assets are transferred (assets are unfrozen), and the contract is signed to confirm all transactions.

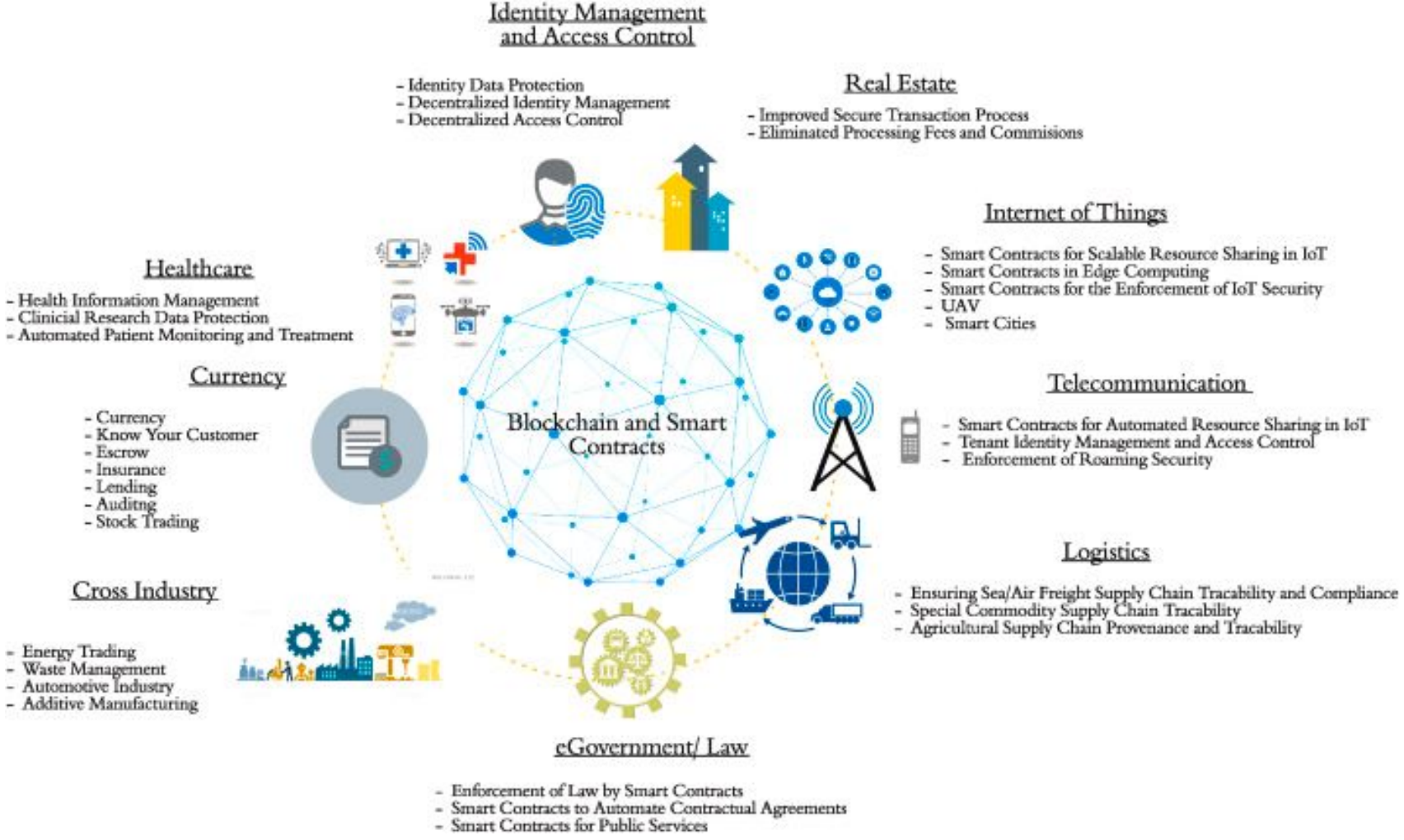
Courtesy : Medium

Use cases of Smart Contracts



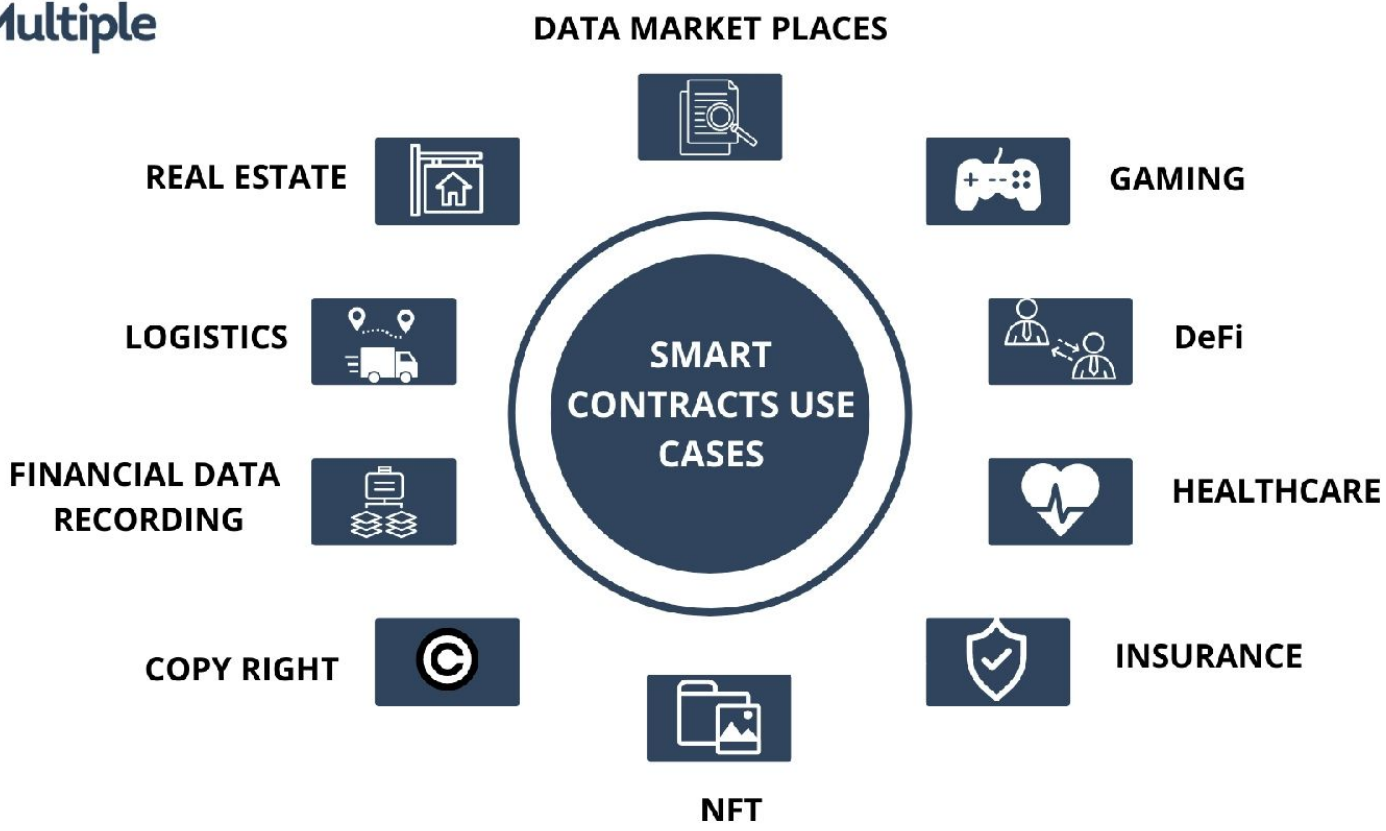


Courtesy : 101 Blockchains



Courtesy : Science Direct Paper

Top 10 Use cases of Smart Contract 2023



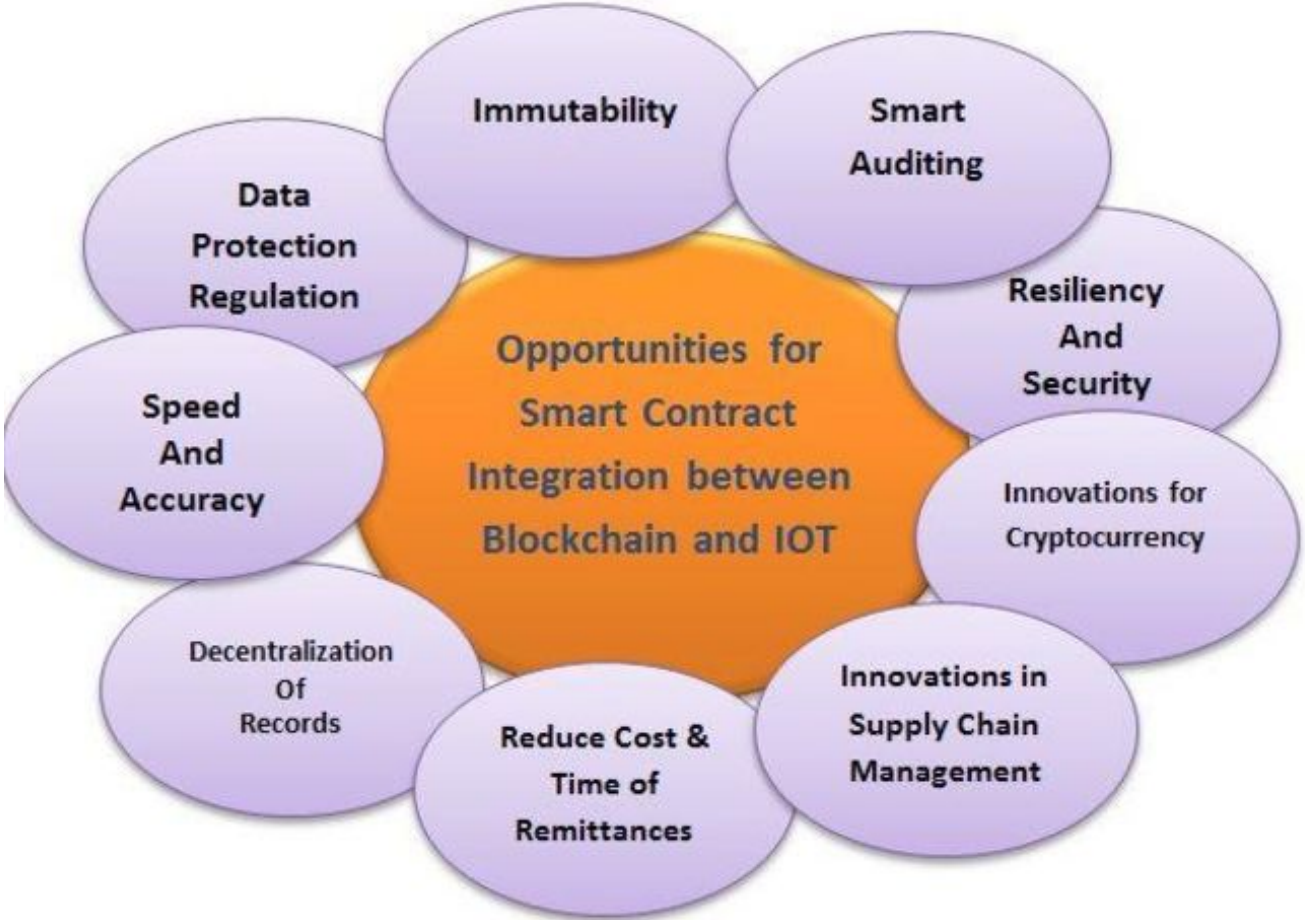
Courtesy : AI Multiple

Common Use cases of Smart Contract

- **DeFi**
  - Cryptocurrencies and smart contracts have allowed decentralized finance platforms to provide financial services without a need for a middleman.
  - DeFi had a total value locked of $94 billion by end of 2021.
  - DeFi has evolved to be more than just peer-to-peer transactions.
  - Smart contracts have enabled sophisticated transactions such as lending, borrowing, and derivative transactions on DeFi platforms.
  - For example:
    - AAVE is a DeFi platform that allows borrowing and lending in different cryptocurrencies.
    - Opyn is a DeFi derivative trading platform that utilizes smart contracts for options trading.
- **NFTs**
  - $17 billion worth of Non-fungible tokens (NFTs) were traded in 2021, making it one of the most impactful smart contract use cases.
  - Even though the market has cooled down in the 2nd quarter of 2022, NFTs have real-life use cases which can lead to long-term use of NFTs.
  - Smart contracts have enabled the creation of non-fungible tokens (NFTs) by allocating ownership and managing the transferability of NFTs. These contracts can also be modified to include additional features such as royalty payments and access rights to a platform or software.

Courtesy : AI Multiple

Smart Contract Opportunities



Courtesy : Semantics Scholar Paper

Smart Contract Opportunities

1. **Trust and Transparency:** Smart contracts operate on decentralized blockchain networks, ensuring trust and transparency in transactions. Parties involved can rely on the code's execution rather than trusting a centralized authority.
2. **Reduced Intermediaries:** By automating contract execution, smart contracts eliminate the need for intermediaries like banks, lawyers, or notaries, reducing costs and delays associated with traditional contracts.
3. **Efficiency:** Smart contracts execute automatically once predefined conditions are met. This reduces manual errors, speeds up processes, and lowers administrative costs.
4. **Security:** Blockchain technology provides a high level of security through encryption and consensus mechanisms. Once data is on the blockchain, it is challenging to alter, enhancing data integrity.
5. **Global Reach:** Smart contracts can be accessed and executed globally, enabling businesses to interact with international partners and customers seamlessly.
6. **Immutable Records:** Transactions recorded on a blockchain are tamper-resistant and cannot be altered once confirmed, ensuring a reliable and permanent record

Smart Contract Challenges

Courtesy : Semantics Scholar Paper

Smart Contract Risks

1. **Coding Errors:** Smart contracts are only as good as the code written to create them. Coding errors or vulnerabilities can lead to significant financial losses or breaches of privacy. The code must be thoroughly audited and tested.
2. **Regulatory Uncertainty:** The legal and regulatory framework for smart contracts varies by jurisdiction. The lack of clarity in some areas can lead to legal challenges or compliance issues.
3. **Irreversible Transactions:** Once a smart contract is deployed, its actions are irreversible. If there's a mistake or dispute, it can be challenging to resolve without external intervention.
4. **Oracles and External Data:** Smart contracts often rely on external data sources (oracles) to trigger actions. If these sources provide incorrect or manipulated data, it can lead to undesirable outcomes.
5. **Scalability:** Blockchain networks, especially Ethereum, face scalability challenges, resulting in slow transaction times and high fees during periods of high demand.
6. **Privacy Concerns:** While blockchain provides transparency, it may not be suitable for contracts that require complete privacy or confidentiality.
7. **Human Element:** Smart contracts can't account for all real-world scenarios. Human intervention may still be needed for complex negotiations or unforeseen circumstances.
8. **Upgrades and Forks:** Blockchain networks can undergo upgrades or forks, potentially impacting the functionality or compatibility of smart contracts.
9. **Cost of Deployment:** Deploying smart contracts on some blockchain networks can be costly due to gas fees (transaction costs) and development expenses.

What Smart Contracts does not promise to do ?

Ease of Correction

Cases of Loophole

Third Party Elimination

Legal Unclarity

Management of Vague Terms & Conditions

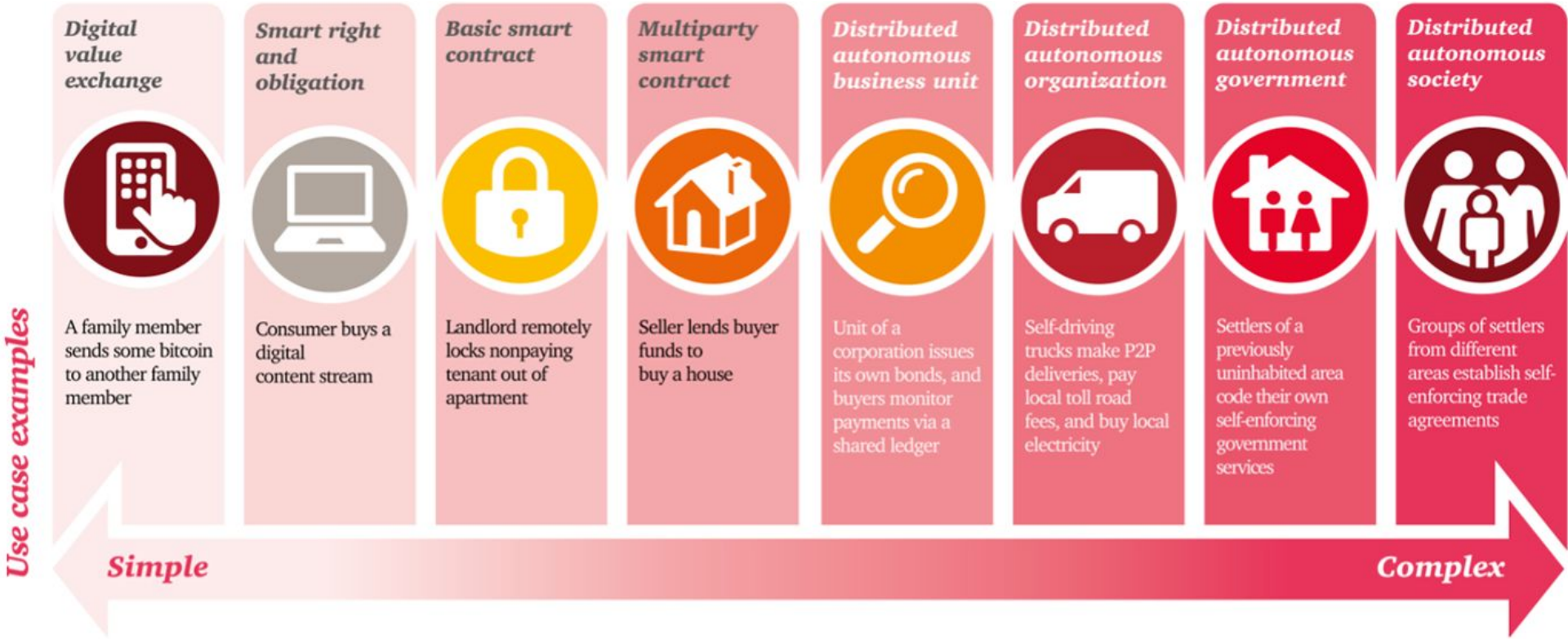
Courtesy : Appinventiv

Blockchain based Smart Contract Integration Platforms

Platform	Blockchain	Smart Contract Language	Consensus Protocol	Cryptocurrency	System Complexity	Scalability
BitCoin	Public & Private	IVY for Bitcoin Language	PoW	BTC	Medium	Block size 3-7 Tx/Sec
Ethereum	Public & Private	Solidity	PoS	Ether (ETH)	High	Block size 5-20 Tx/Sec
HyperLedger Fabric	Private	Java, Node.js and Go	PBFT/SIEVE	None	High	Block size 100-3000 Tx/Sec
NEM	Public	Java and Node.js	PoI	XEM	Medium	Block size 1,000-10,000 Tx/Sec
Stellar	Public	Stellar SDK & Go	PBFT / FBA	Lumens (XLM)	High	Block size 1,000-1,500 Tx/Sec
Waves	Public	RIDE	LPoS	WAVES	Medium	Block size 100 Tx/Sec
Lisk	Public	Lisk JScript	DPoS	LSK	Medium	Block size 25Tx/Sec
NXT	Public	TC Script	PoS	NXT	Medium	Block size 5-20 Tx/Sec
Monax	Private	Monax SDK and Solidity	PoS	MultiAsset	High	-
Qtum	Public	QSCL and Solidity	PoS	QTUM	Medium	Block size 70-140Tx/Sec

Courtesy : Semantics Scholar Paper

Smart Contracts by Complexity Level



Courtesy : LinkedIn - Marvin Fedler

5. Overview of Decentralized Applications (DApps)

- A software application that operates on a decentralized network, typically using blockchain technology.
- Compared to traditional applications that rely on centralized servers, Dapps leverage the security, transparency, and trustlessness of blockchain to function.
- Salient Features of DApps
  1. **Decentralization:**
     - Dapps run on a decentralized network of computers, often referred to as a blockchain.
     - This network is distributed across many nodes, making it resistant to single points of failure and censorship.
  2. **Smart Contracts**:
     - Dapps typically use smart contracts, which are self-executing contracts with the terms of the agreement directly written into code.
     - Smart contracts automatically execute actions when specific conditions are met, without the need for intermediaries.
  3. **Transparency:**
     - All transactions and actions within a Dapp are recorded on a public ledger (the blockchain).
     - This transparency ensures that anyone can verify transactions and data, enhancing trust in the application.

Courtesy : Preeti - DApp

Introduction to Dapp (Decentralized Application)

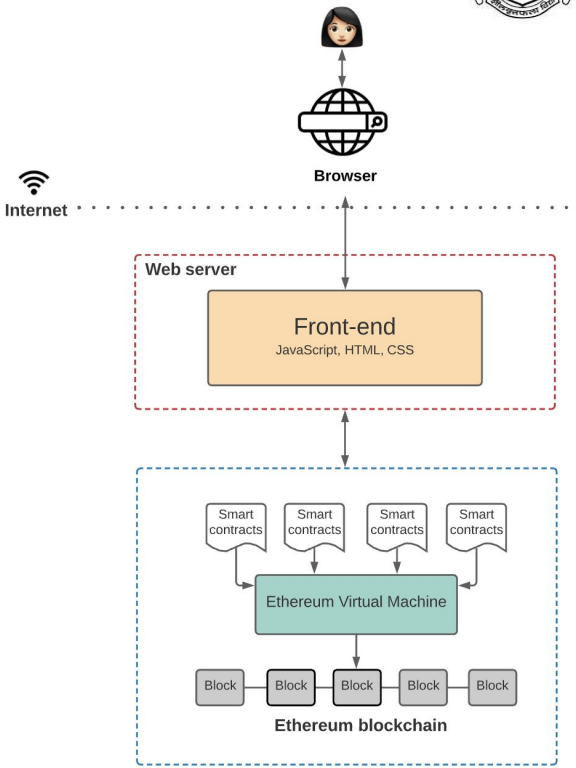
4. **Security:**
   - Blockchain technology, with its cryptographic techniques, provides a high level of security.
   - Transactions are immutable, meaning once recorded on the blockchain, they cannot be altered.
   - This makes Dapps resilient to fraud and hacking.
5. **Trustlessness:**
   - Dapps aim to operate without relying on a central authority or intermediary.
   - Users can interact with the application and each other directly without needing to trust a third party.
6. **Token Economy:**
   - Many Dapps have their own native tokens or use established cryptocurrencies like Ethereum's Ether.
   - These tokens can be used for various purposes within the application, such as paying for services, participating in governance, or earning rewards.
7. **Use Cases:**
   - Dapps have a wide range of use cases, including decentralized finance (DeFi), non-fungible tokens (NFTs), supply chain management, voting systems, gaming, and more.
   - Each Dapp is designed to solve specific problems or provide new functionalities in a decentralized manner.

Introduction to Dapp (Decentralized Application)

8. **User Interface:**
   - Dapps often have user interfaces (UIs) similar to traditional apps or websites, making them accessible to a broader user base.
   - Users may not even be aware that they are interacting with blockchain technology.
9. **Challenges:**
   - While Dapps offer numerous advantages, they also face challenges, including scalability issues, user adoption barriers, and regulatory compliance.
   - These challenges are actively being addressed by the blockchain community.
10. **Development:**
   - Building a Dapp typically involves programming smart contracts using languages like Solidity (for Ethereum) and developing a frontend interface.
   - Popular Dapp development frameworks like Truffle and web3.js make the development process more accessible.

Dapp Architecture

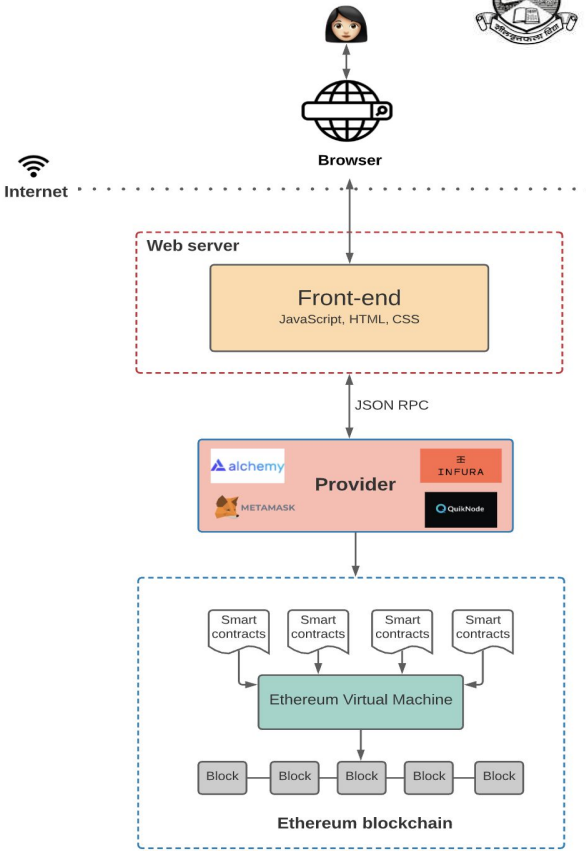
1. **Ethereum blockchain:**
   - Securely executes and verifies application code, called smart contracts.
   - DApps use the Ethereum blockchain for data storage.
2. **Smart Contracts:**
   - DApps use smart contracts to define the state changes happening on the blockchain.
   - A smart contract is a collection of code and data that resides at a specific address on the Ethereum Blockchain and runs on the Ethereum blockchain.
3. **Ethereum Virtual Machine(EVM):**
   - Global virtual computer that executes the logic defined in the smart contracts and processes the state changes that happen on this Ethereum network.
4. **Front-end:**
   - The part of the DApps, that users can see and interact with such as the graphical user interface(GUI),
   - Communicates with the application logic defined in smart contracts.



Courtesy : Preeti - DApp

DApp - How Frontend Code Communicate ?

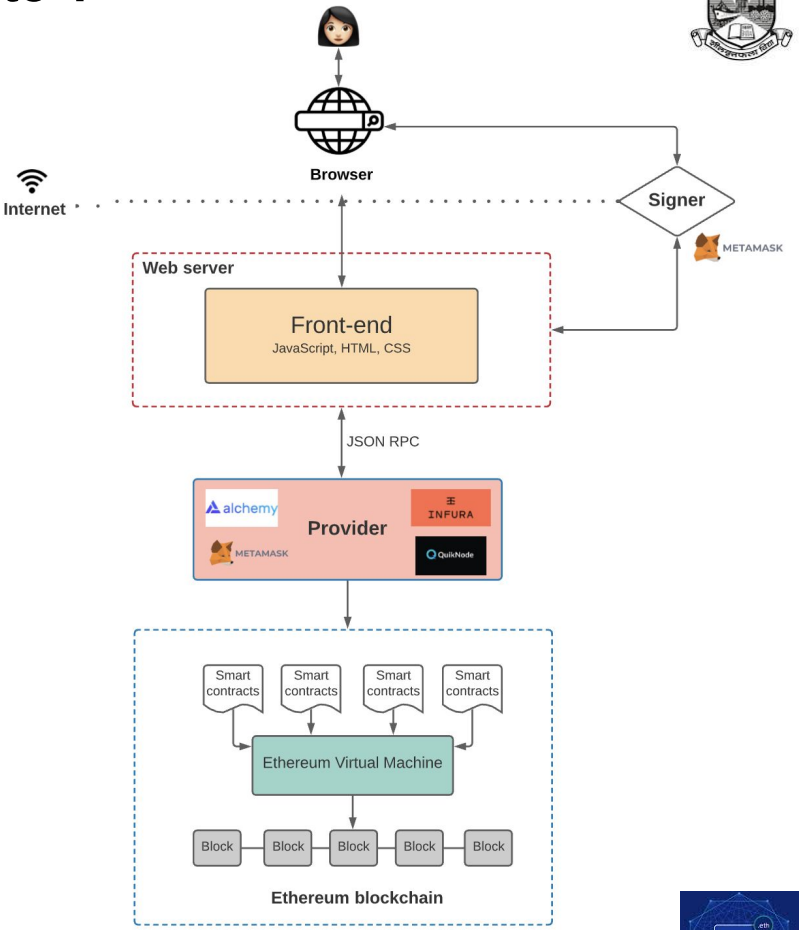
- When we need to interact with the data and code on a blockchain, we need to interact with one of these nodes.
- As any node can broadcast a request for a transaction to be executed on the EVM.
- A miner will then execute the transaction and propagate the resulting state change to the rest of the network.
- There are two ways to broadcast a new transaction:
  1. Set up your own node which runs the Ethereum blockchain software
  2. Use nodes provided by third-party services like Infura, Alchemy, and Quicknode
- Every Ethereum client (i.e. provider) implements a JSON-RPC specification. This ensures that there's a uniform set of methods when frontend applications want to interact with the blockchain.



Courtesy : Preeti - DApp

DApp - How Frontend Code Communicate ?

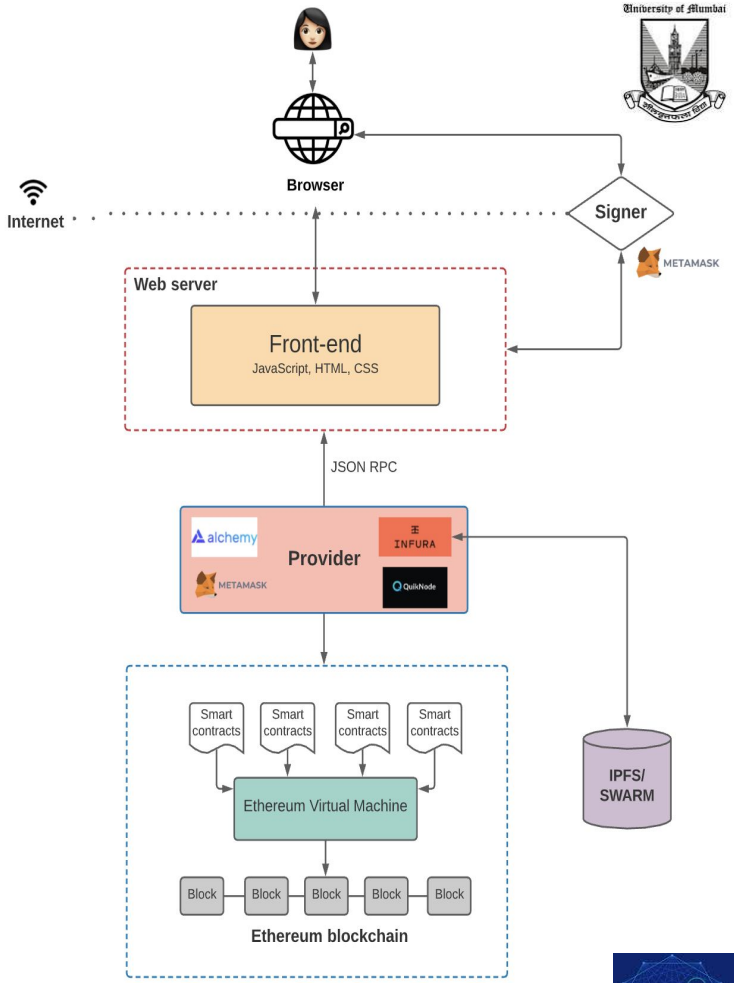
- When a user wants to publish a new post onto the chain, our DApp would ask the user to “sign” the transaction using their private key — only then would the DApp relay the transaction to the blockchain.
- Otherwise, the nodes wouldn’t accept the transaction.
- This “signing” of transactions is where Metamask typically comes in.



Courtesy : Preeti - DApp

DApp - Storage on the Blockchain

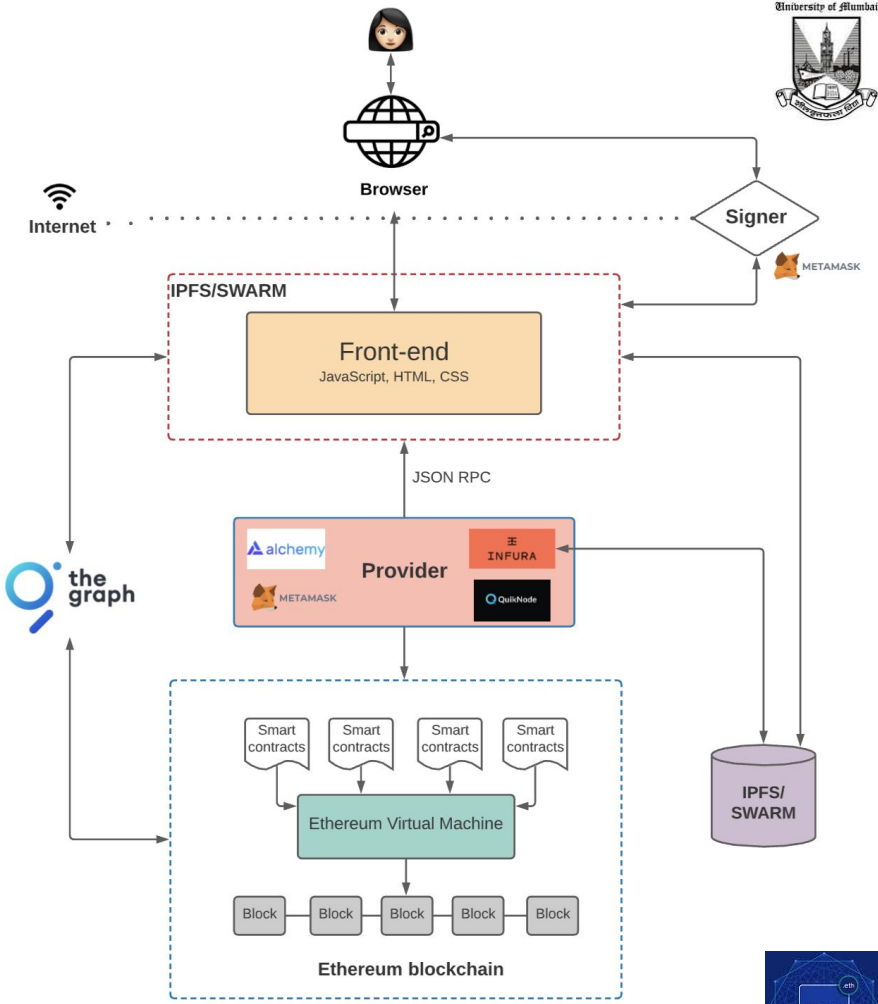
- Storing everything on the blockchain gets really expensive.
- Users to pay extra for using your DApp every time their transaction requires adding a new state is not the best user experience.
- One way to combat this is to **use a decentralized off-chain storage solution**, like IPFS or Swarm.
- IPFS : Distributed file system for storing and accessing data.
  - distributes and stores the data in a peer-to-peer network.
  - Has an incentive layer known as “Filecoin.” This layer incentivizes nodes around the world to store and retrieve this data.
  - IPFS providers
    - **Infura** (which provides you with an IPFS node) or
    - Pinata (“pin” your files to IPFS and take the IPFS hash and store that on the blockchain).
- **Swarm** : A decentralized storage network,
  - **Difference :** Swarm’s incentive system is built-in and enforced through smart contracts on the Ethereum blockchain for storing and retrieving data.



Courtesy : Preeti - DApp

DApp Querying the Blockchain

1. **Smart Contract Events :**
   - Use the Web3.js library to query and listen for smart contract events.
   - Here, we need to listen to specific events and specify a callback every time the event is fired.
   - using callbacks to handle various UI logic gets very complex
   - Issue: when you deploy a smart contract and later realize you need an event emitted that you didn't originally include.
2. **The Graph**
   - An off-chain indexing solution that makes it easier to query data on the Ethereum blockchain.
   - Allows to define which smart contracts to index, which events and function calls to listen to, and how to transform incoming events into entities that the frontend logic can consume.
   - It uses **GraphQL** as a query language,
   - By indexing blockchain data, The Graph lets us query on-chain data in our application logic with low latency

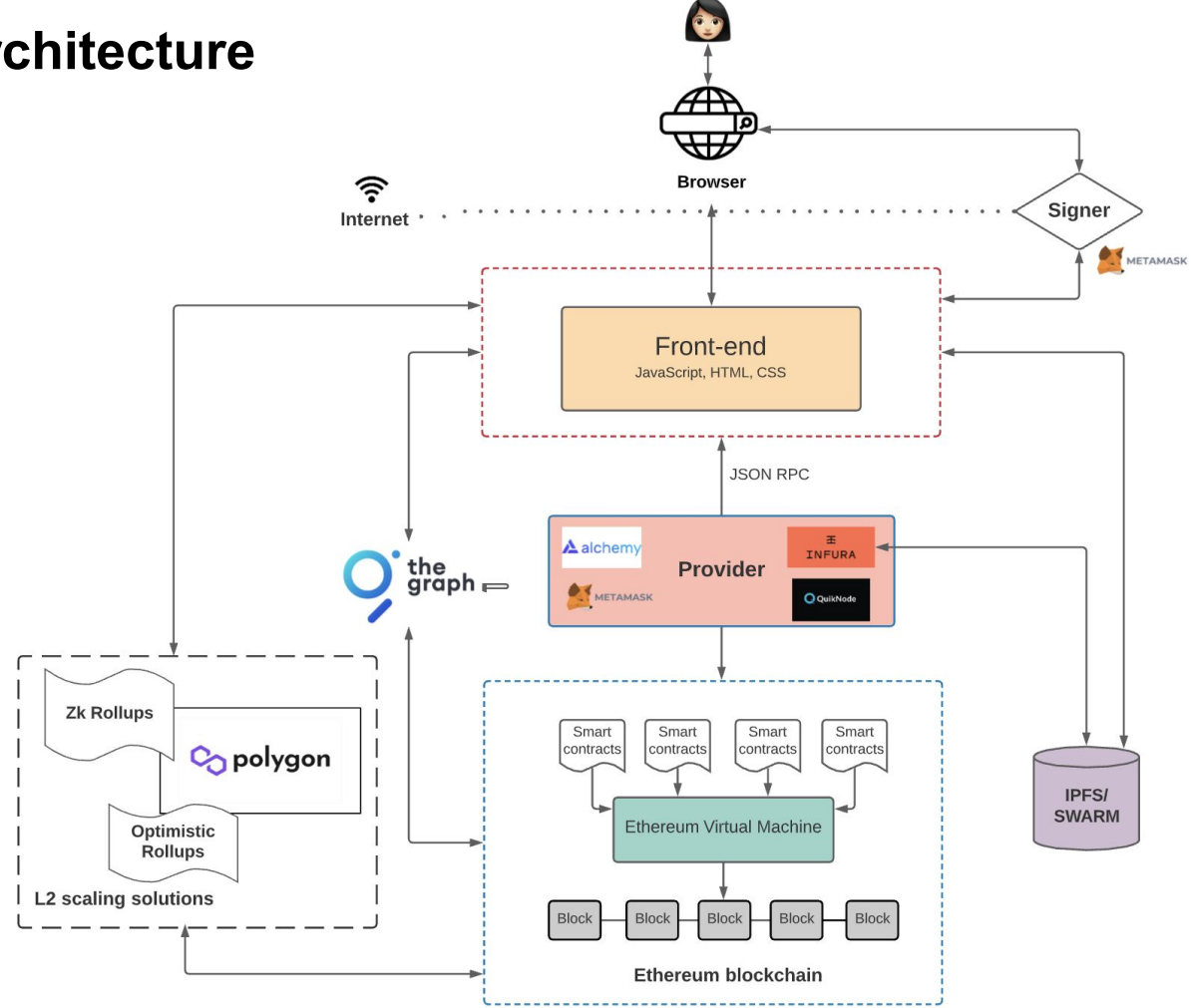


Courtesy : Preeti - DApp

Overall DApp Architecture

Addressing

- Storage,
- Querying
- Scalability



Courtesy : Preeti - DApp

6. Token Standards

WHAT ARE ERC STANDARDS?

Ethereum Request for Comments (ERC) is a document that smart contract programmers are using in the Ethereum blockchain platform to write. These are rules that the Ethereum-based tokens must comply with.

CREATION PROCESS

1. Developers submit new ERC policies through Ethereum Improvement Proposal
2. Ethereum community reviews these documents
3. The community may ask for revision or comment in favor or oppose
4. After acceptation, the developers implement it

Courtesy : 101 Blockchain - ERC Standards

ERC20

ERC Standards

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.

Courtesy : 101 Blockchain - ERC Standards

ERC20

ERC Standards

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.

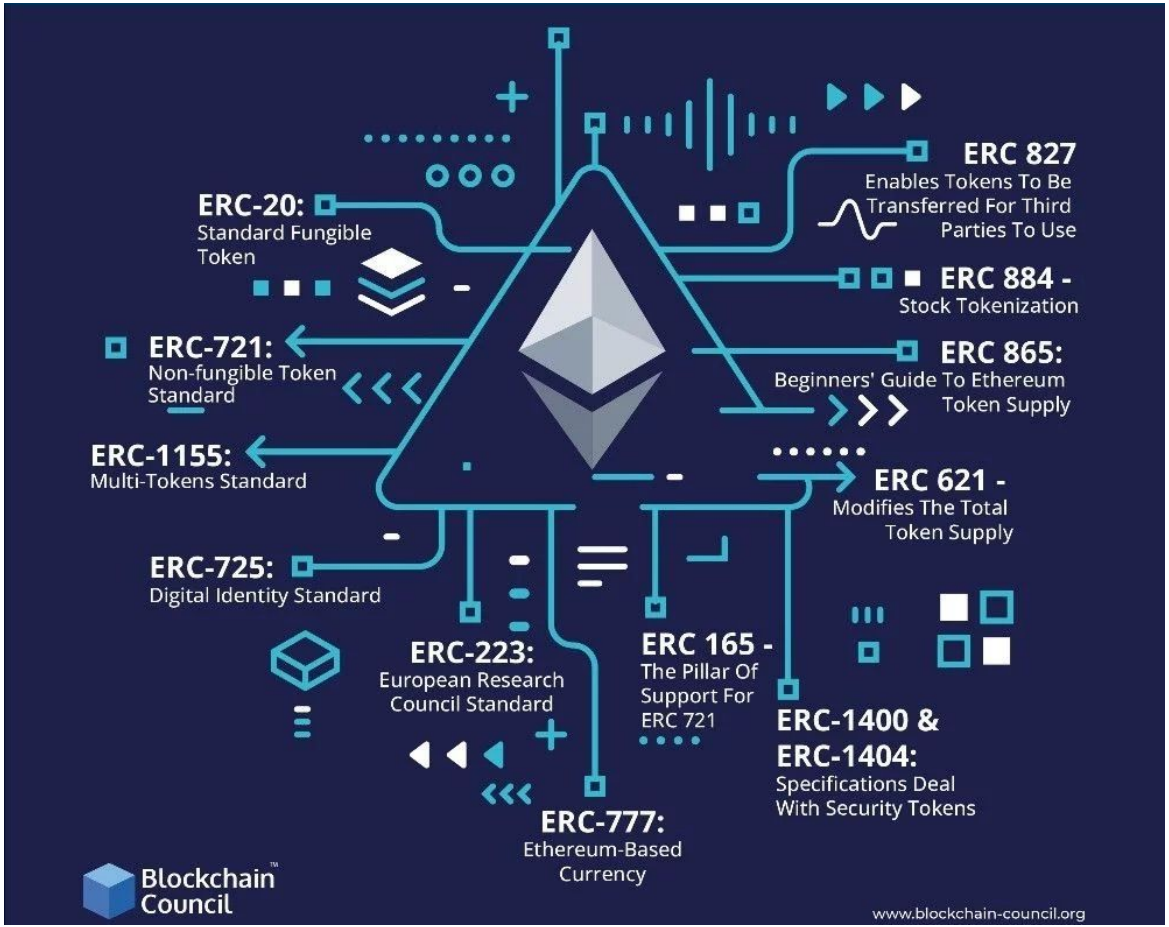
ERC 884

- 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.

Created by 101blockchains.com

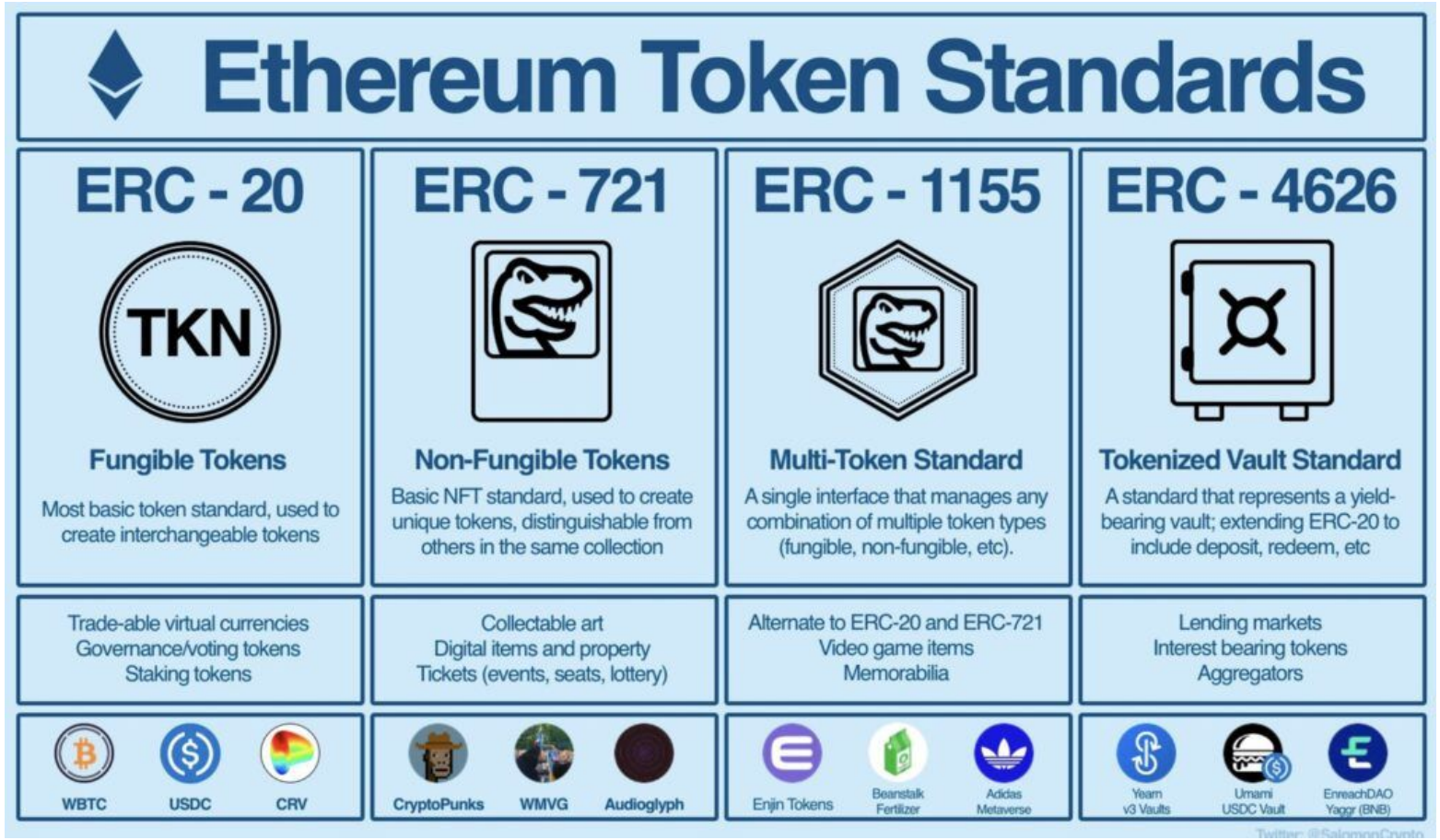
Courtesy : 101 Blockchain - ERC Standards

ERC Token Standards List



Courtesy : [Blockchain Council - ERC Token Standards]

ERC Token Standards List



Courtesy : [Salomon - ERC Token Standards]

ERC20 Token

- ERC-20 is the **technical standard for fungible tokens created using the Ethereum blockchain.**
- A **fungible token is interchangeable with another token**—where the well-known non-fungible tokens (NFTs) are not interchangeable.
- ERC-20 allows developers **to create smart-contract-enabled tokens that can be used with other products and services.**
- These **tokens are a representation of an asset, right, ownership, access, cryptocurrency, or anything else that is not unique in and of itself but can be transferred.**
- A **list of functions and events** that must be implemented into a token for it to be considered ERC-20 compliant.
    - **functions** describe **what must be included in the smart-contract-enabled token,**
    - **events** describe **an action.**
- ERC-20 guides the creation of new tokens on the Ethereum blockchain so that they are interchangeable with other smart contract tokens.

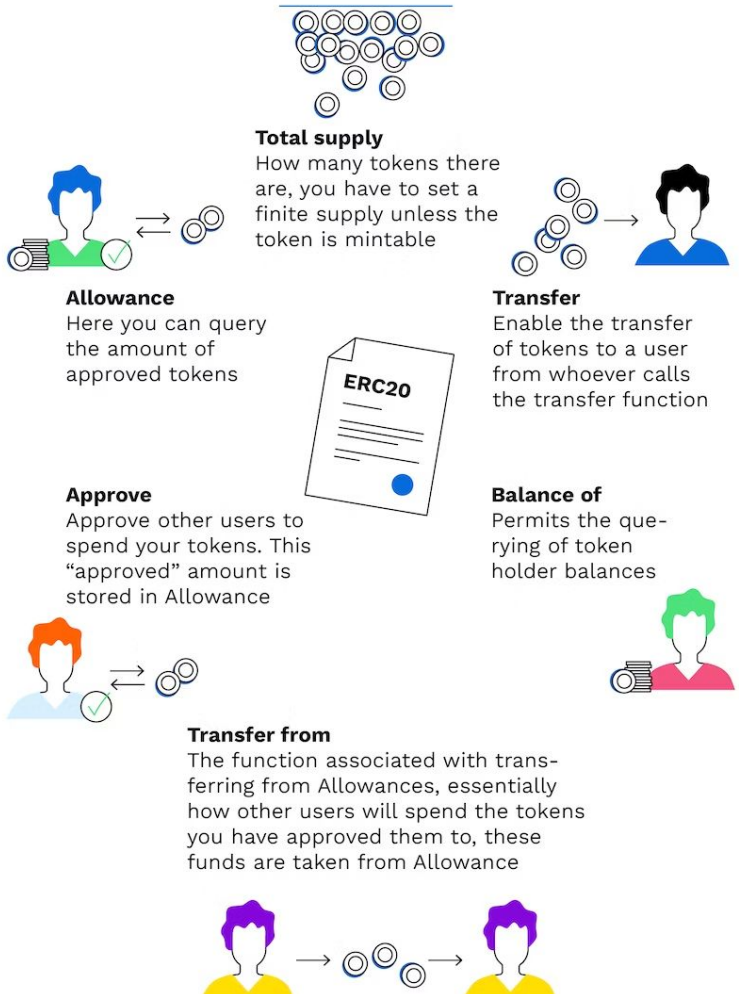
Courtesy : Investopedia - ERC 20

ERC20 Token Features

Courtesy : [Coinsclone - ERC 20]

ERC20 Token Functions

1. **TotalSupply:** The total number of tokens that will ever be issued
2. **BalanceOf:** The account balance of a token owner's account
3. **Transfer:** Automatically executes transfers of a specified number of tokens to a specified address for transactions using the token
4. **TransferFrom:** Automatically executes transfers of a specified number of tokens from a specified address using the token
5. **Approve:** Allows a spender to withdraw a set number of tokens from a specified account, up to a specific amount
6. **Allowance:** Returns a set number of tokens from a spender to the owner



Courtesy : Investopedia - ERC 20
Bitpanda - ERC 20

ERC20 Token

The events that must be included in the token are:

1. **Transfer:** An event triggered when a transfer is successful
2. **Approval:** A log of an approved event (an event)

The following functions are optional and are not required to be included, but they enhance the token's usability:

1. Token's name (optional)
2. Its symbol (optional)
3. Decimal points to use (optional)

Courtesy : Investopedia - ERC 20

ERC20 Token

What is the significance of ERC-20 tokens?

ERC-20 tokens have become the standard for creating new tokens on the Ethereum Blockchain. This standardization makes it easier for developers to create new tokens and for users to interact with them, as they all follow the same set of rules.

Additionally, ERC-20 tokens have enabled the creation of decentralized applications (dApps) on the Ethereum network, which has the potential to revolutionize industries.

What are the risks associated with ERC-20 tokens?

Like all cryptocurrencies, ERC-20 tokens are volatile and their value can fluctuate rapidly. Additionally, there is a risk of fraud, scams, and theft. It's important to do your research and be cautious when investing in ERC-20 tokens or any other cryptocurrency.

ERC20 Token

How do ERC-20 tokens differ from other cryptocurrencies?

ERC-20 tokens are built on the Ethereum Blockchain and follow a standardized set of rules and guidelines. This makes it easier for developers to create new tokens and for users to interact with them. Other cryptocurrencies may have different rules and guidelines, making them more complex to work with.

How can I use ERC-20 tokens?

- ERC-20 tokens can be used for a variety of purposes, including investment, trading, and participating in decentralized applications.
- Some ERC-20 tokens can also be used for purchasing goods and services, although this is less common.

Courtesy : Blockchain Council

ERC20 Token

Popular Digital currencies use the ERC-20 standard.

- Tether USD (USDT)
- USD Coin (USDC)
- Shiba Inu (SHIB)
- Binance USD (BUSD)
- BNB (BNB)
- DAI Stablecoin (DAI)
- HEX (HEX)
- Bitfinex LEO (LEO)
- MAKER (MKR)

Courtesy : Investopedia - ERC 20
Blockchain Council

ERC20

ERC20 Token - Advantages

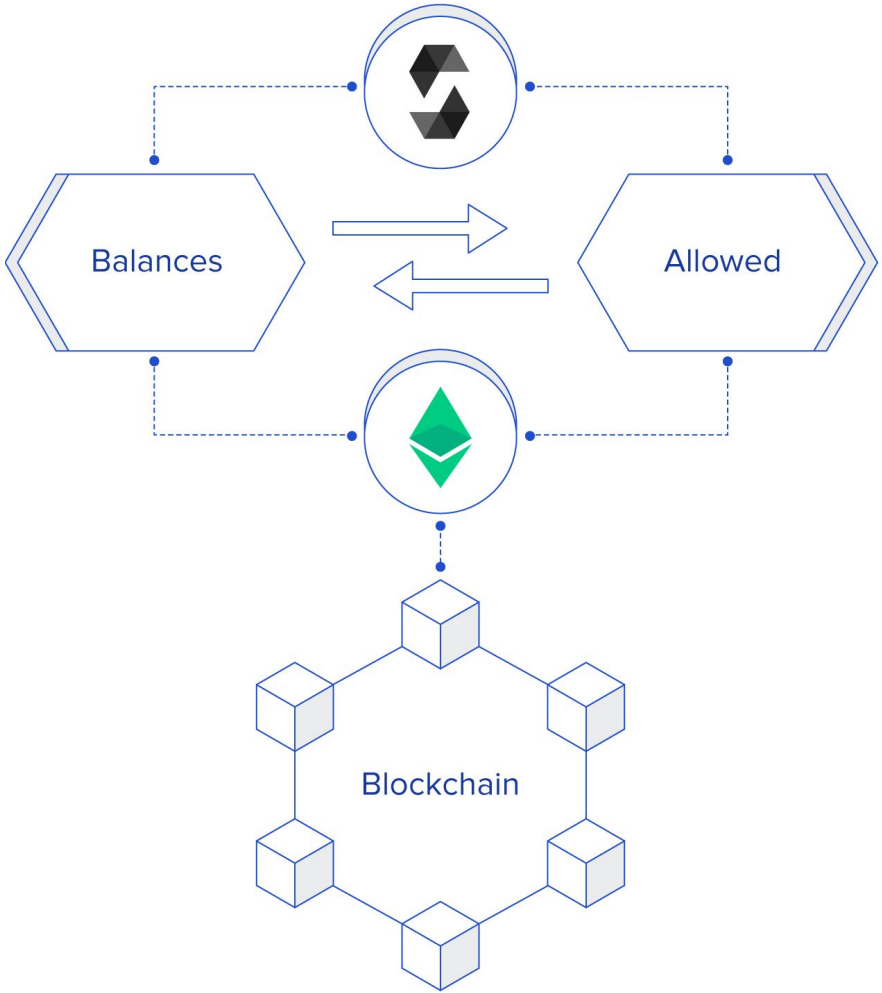
1. **Interoperability:** ERC-20 tokens adhere to a standardized set of rules, making them compatible with a wide range of wallets, exchanges, and applications. This interoperability fosters easy integration and widespread use across different platforms.
2. **Ease of Development:** Creating ERC-20 tokens is relatively straightforward due to the well-defined standard. This simplicity has lowered the barrier for developers to create their own tokens and launch initial coin offerings (ICOs) or token sales.
3. **Widespread Adoption:** The ERC-20 standard has gained widespread adoption within the Ethereum ecosystem. Many tokens, including some of the most well-known cryptocurrencies and utility tokens, are based on the ERC-20 standard.
4. **Liquidity and Trading:** ERC-20 tokens can be easily traded on various decentralized and centralized exchanges. Their widespread availability has contributed to liquidity and trading opportunities for token holders.
5. **Token Integration:** ERC-20 tokens can be integrated into various decentralized applications (dApps), allowing developers to build applications that utilize tokens for specific use cases, such as voting, rewards, governance, and more.
6. **Token Standards Evolution:** The success of ERC-20 paved the way for the development of other token standards, such as ERC-721 (non-fungible tokens) and ERC-1155 (multi-token standard), catering to different token use cases and enhancing the Ethereum ecosystem's versatility.

ERC20 Token - Disadvantages

1. **Limited Functionality:** ERC-20 tokens are primarily suited for fungible assets. If a project requires more complex features, such as non-fungible tokens (NFTs) or unique attributes for each token, ERC-20 might not be the ideal choice.
2. **Scalability and Gas Fees:** The Ethereum blockchain has faced scalability challenges, leading to network congestion and high gas fees during periods of heavy usage. This can impact the cost and speed of transactions involving ERC-20 tokens.
3. **Lack of Native Support for Advanced Features:** ERC-20 tokens lack built-in support for certain features, such as royalties, more intricate tokenomics, and on-chain interactions that other token standards might offer.
4. **Security Risks:** Although the ERC-20 standard provides a framework for secure token development, improper coding practices or vulnerabilities can still expose tokens to potential security risks and vulnerabilities.
5. **Lost Tokens:** ERC-20 tokens can be accidentally sent to the wrong addresses or lost due to mistakes in handling private keys. Unlike centralized systems, there's no straightforward way to recover lost tokens.
6. **Regulatory Concerns:** Due to the ease of creating tokens using the ERC-20 standard, there have been instances of fraudulent token sales or scams. This has led to regulatory scrutiny in some cases.

ERC20 Token

Writing an ERC20 Token in Solidity



Courtesy : Toptal - ERC 20

ERC 721 Token

- An ERC-721 token is a **type of non-fungible token (NFT) standard** on the Ethereum blockchain.
- ERC-721 tokens are **unique and distinct from each other.**
- Each ERC-721 token has **individual properties, attributes, or characteristics,** making them ideal for representing ownership of one-of-a-kind digital or physical assets.
- The term "ERC-721" stands for "**Ethereum Request for Comment 721**," and
- it refers to the **standard that defines the rules and functions for creating and managing non-fungible tokens on the Ethereum platform.**
- This standard was introduced **to enable the creation of digital assets that can represent collectibles, virtual items, artwork, real estate, in-game items,** and more, where each token holds unique value.

ERC 721 Token Characteristics

1. **Uniqueness:**
   - Each ERC-721 token has a distinct identifier, and no two tokens are the same. This allows for the representation of rare or unique items that hold individual value.
2. **Properties and Metadata:**
   - ERC-721 tokens can include metadata, attributes, and additional information that provide context and value to the token. For example, a digital artwork NFT could include the artist's name, creation date, and other relevant details.
3. **Ownership and Transfers:**
   - Ownership of an ERC-721 token is recorded on the blockchain. Tokens can be transferred between addresses using the standard transferFrom function. Ownership changes are transparent and verifiable on the blockchain.
4. **Use Cases:**
   - ERC-721 tokens have a wide range of use cases, including digital art ownership, virtual collectibles, unique in-game items, domain names, real estate representation, identity verification, and more.

ERC 721 Token Characteristics

5. **Standard Interfaces:**
   - ERC-721 defines a set of standard functions for interacting with tokens, including querying ownership, transferring ownership, and querying token metadata.
6. **Decentralized Applications (dApps):**
   - Developers can build decentralized applications that utilize ERC-721 tokens. These dApps can create marketplaces, games, platforms for trading collectibles, and other innovative services that leverage the uniqueness of NFTs.
7. **Ecosystem Growth:**
   - The popularity of ERC-721 tokens has led to the growth of NFT marketplaces and platforms, where users can buy, sell, and trade NFTs. These tokens have gained significant attention, especially in the realms of art, entertainment, and gaming.
8. **Gas Costs and Scalability:**
   - Transacting with ERC-721 tokens can be more gas-intensive than ERC-20 tokens due to the uniqueness and additional metadata associated with each token. Scalability remains a consideration, especially during periods of high network congestion.

Comparison between ERC20 & ERC721

Criteria	ERC-20	ERC-721
Fungibility	Fungible in nature.	Non-fungible in nature.
Token Identity	There is no specific disparity among the different tokens.	Each token has a specific identity and could be easily distinguished.
Collecting Tokens	ERC-20 tokens are not collectible.	You can collect ERC-721 tokens like fiat currency.
Value Fluctuation	The value of ERC-20 tokens remains the same.	The value of ERC-721 tokens fluctuates according to rarity and uniqueness.
Adoption	Commonly adopted.	Limited levels of acceptance.
Substitutes	Easier for substitution.	No scope for substitution.
Divisibility	Can be divisible into decimals.	ERC-721 tokens are not divisible.
Ownership Functions	No special ownership functions are allocated.	ERC-721 tokens can enable special ownership functions.

Courtesy : 101 Blockchains

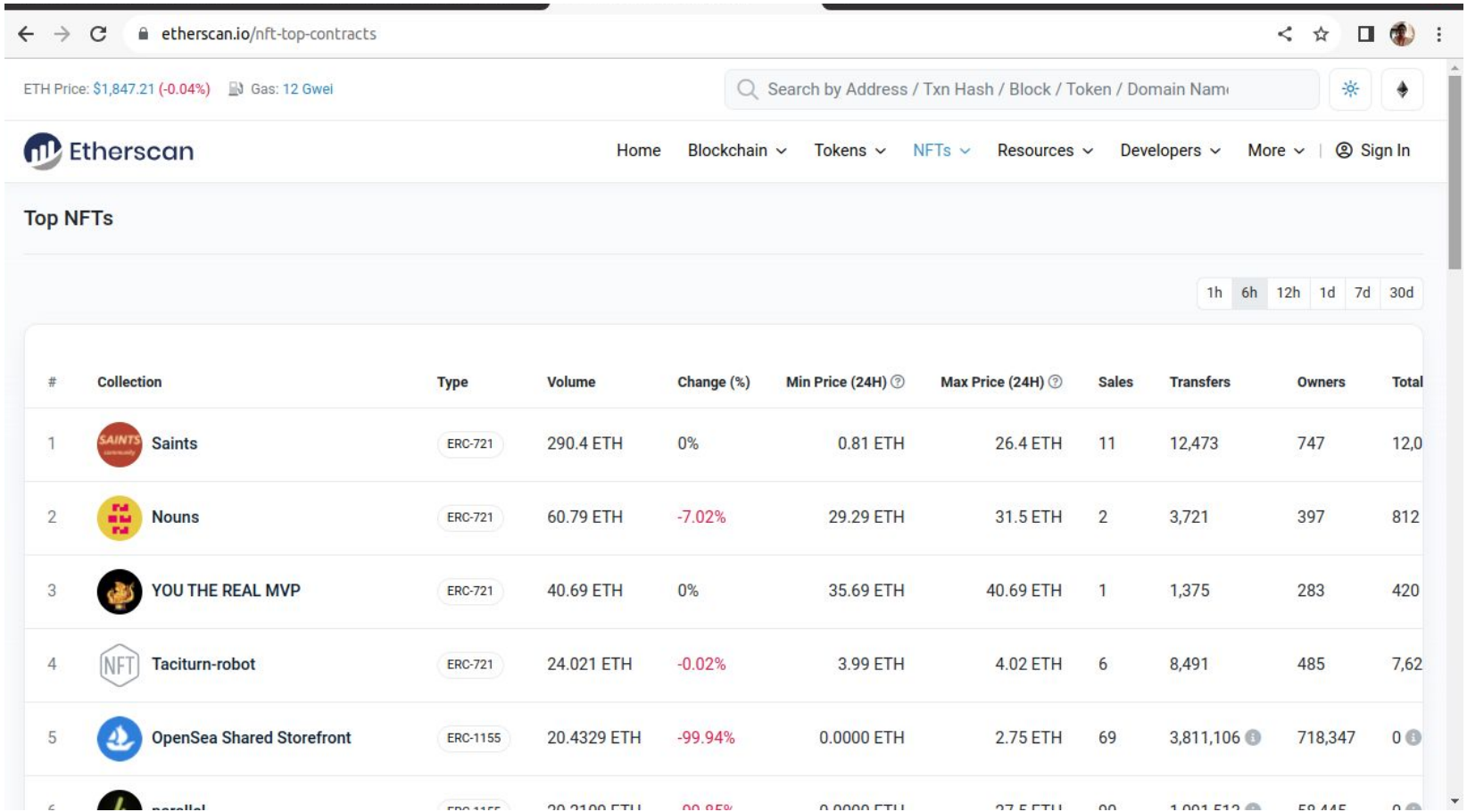
Comparison between ERC20 & ERC721

Aspect	ERC-20 Tokens	ERC-721 Tokens
Type	Fungible tokens (identical and interchangeable)	Non-fungible tokens (unique and distinct)
Individuality	Tokens are identical, no unique attributes	Each token has unique attributes or properties
Use Cases	Currency, utility tokens, rewards, ICOs	Digital collectibles, gaming items, unique assets
Transfers	Can be transferred on a one-to-one basis	Transfers may involve individual token properties
Standard	Single standard (ERC-20)	Single standard (ERC-721)
Functions	Basic functions: transfer, balanceOf, approve, etc.	Advanced functions for managing unique tokens
Properties	No inherent properties or metadata	Tokens can have metadata, name, and attributes

Comparison between ERC20 & ERC721

Aspect	ERC-20 Tokens	ERC-721 Tokens
Interoperability	High interoperability across platforms	May require specialized interfaces for certain apps
Token Identifiers	Token IDs are not unique across contracts	Token IDs are unique within a contract
Example	Ethereum (ETH), Chainlink (LINK)	CryptoKitties, Decentraland LAND
Scalability	Scalable for large quantities of tokens	Potentially less scalable due to unique properties
Gas Costs	Lower gas costs due to standardized operations	Higher gas costs for more complex transfers
Complexity	Simpler to implement	More complex due to individual token attributes
Registries	Not required; tokens can exist independently	Often use token registries for unique IDs

Exploring etherscan.io - top NFT's



#	Collection	Type	Volume	Change (%)	Min Price (24H)	Max Price (24H)	Sales	Transfers	Owners	Total
1	Saints	ERC-721	290.4 ETH	0%	0.81 ETH	26.4 ETH	11	12,473	747	12,0
2	Nouns	ERC-721	60.79 ETH	-7.02%	29.29 ETH	31.5 ETH	2	3,721	397	812
3	YOU THE REAL MVP	ERC-721	40.69 ETH	0%	35.69 ETH	40.69 ETH	1	1,375	283	420
4	Taciturn-robot	ERC-721	24.021 ETH	-0.02%	3.99 ETH	4.02 ETH	6	8,491	485	7,62
5	OpenSea Shared Storefront	ERC-1155	20.4329 ETH	-99.94%	0.0000 ETH	2.75 ETH	69	3,811,106	718,347	0

Courtesy : Etherscan.io

7. Security Aspects of Blockchain

Blockchain ensures trust in decentralized systems

- Provides immutability, transparency, and cryptographic security
- However, it is not immune to attacks and vulnerabilities

Core Security Features of Blockchain

- Cryptographic Hash Functions
- Public-Key Cryptography
- Consensus Mechanisms (PoW, PoS)
- Decentralization & Distributed Ledger

Courtesy :

7. Security Aspects of Blockchain

51% Attack

- Occurs when a single entity controls majority of network hash power
- Can lead to double spending
- Threat mostly to smaller blockchain networks

Double Spending Attack

- Spending the same cryptocurrency twice
- Exploits transaction confirmation delays
- Mitigated by waiting for multiple confirmations

Sybil Attack

- Attacker creates multiple fake identities
  - Attempts to influence consensus
  - Mitigated using Proof-of-Work / Proof-of-Stake

Courtesy :

7. Security Aspects of Blockchain

Smart Contract Vulnerabilities

- Reentrancy attacks
  - Integer overflow/underflow
  - Poor access control
  - Example: Exploitation of vulnerable contract logic

Private Key & Phishing Attacks

- Users targeted to reveal private keys
  - Wallet malware attacks
  - Irreversible loss of digital assets

Network-Level Attacks

- Eclipse attacks
  - Routing attacks (BGP hijacking)
  - Denial of Service (DoS)

7. Security Aspects of Blockchain

Key Security Challenges

- Scalability vs Security trade-off
  - Energy consumption (PoW)
  - Regulatory uncertainty
  - Smart contract audit complexity

Best Practices for Blockchain Security

- Use hardware wallets
  - Enable two-factor authentication (2FA)
  - Regular code audits
  - Network monitoring and anomaly detection

Mitigation Techniques

- Decentralized and robust consensus mechanisms
  - Smart contract audits and formal verification
  - Multi-signature wallets
  - Cold storage for private keys
  - Regular security testing