INTRODUCTION TO BLOCKCHAIN

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Abhinav Sharma
Maksimjeet Chowdhary

FTX

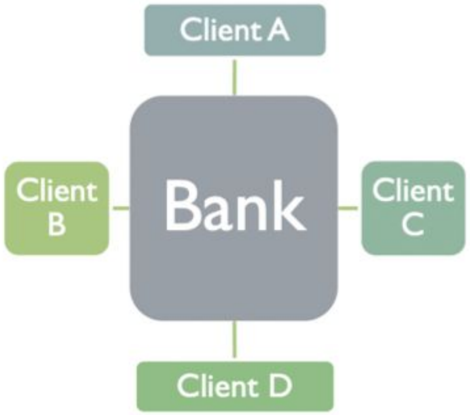
“To understand the power of blockchain systems, and the things they can do, it is important to distinguish between three things that are commonly muddled up, namely the bitcoin currency, the specific blockchain that underpins it and the idea of blockchains in general.”

The Trust Machine, THE ECONOMIST, Oct. 31, 2015

Definition

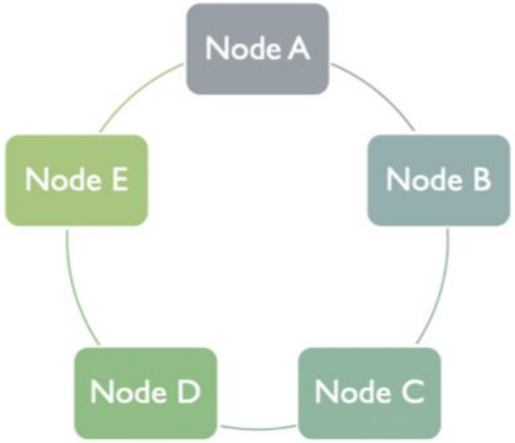
A **blockchain** is a type of distributed ledger technology (DLT) that consists of growing lists of records, called *blocks*, that are securely linked together using cryptography.

Centralized Ledger



- There are multiple ledgers, but Bank holds the "golden record"
- Client B must reconcile its own ledger against that of Bank, and must convince Bank of the "true state" of the Bank ledger if discrepancies arise

Distributed Ledger



- There is one ledger. All Nodes have some level of access to that ledger.
- All Nodes agree to a protocol that determines the "true state" of the ledger at any point in time. The application of this protocol is sometimes called "achieving consensus."

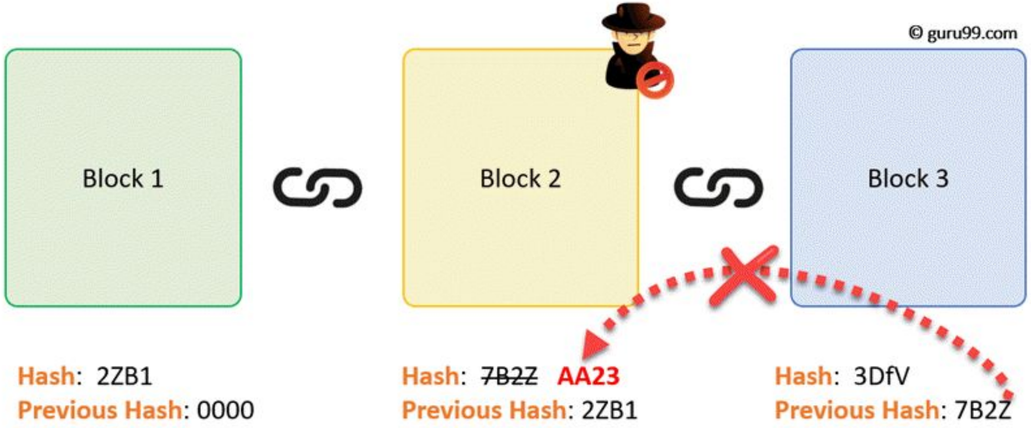
source:https://www.sec.gov/slides-nancy-lia.pdf

Definition

Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree, where data nodes are represented by leaves). The timestamp proves that the transaction data existed when the block was created.

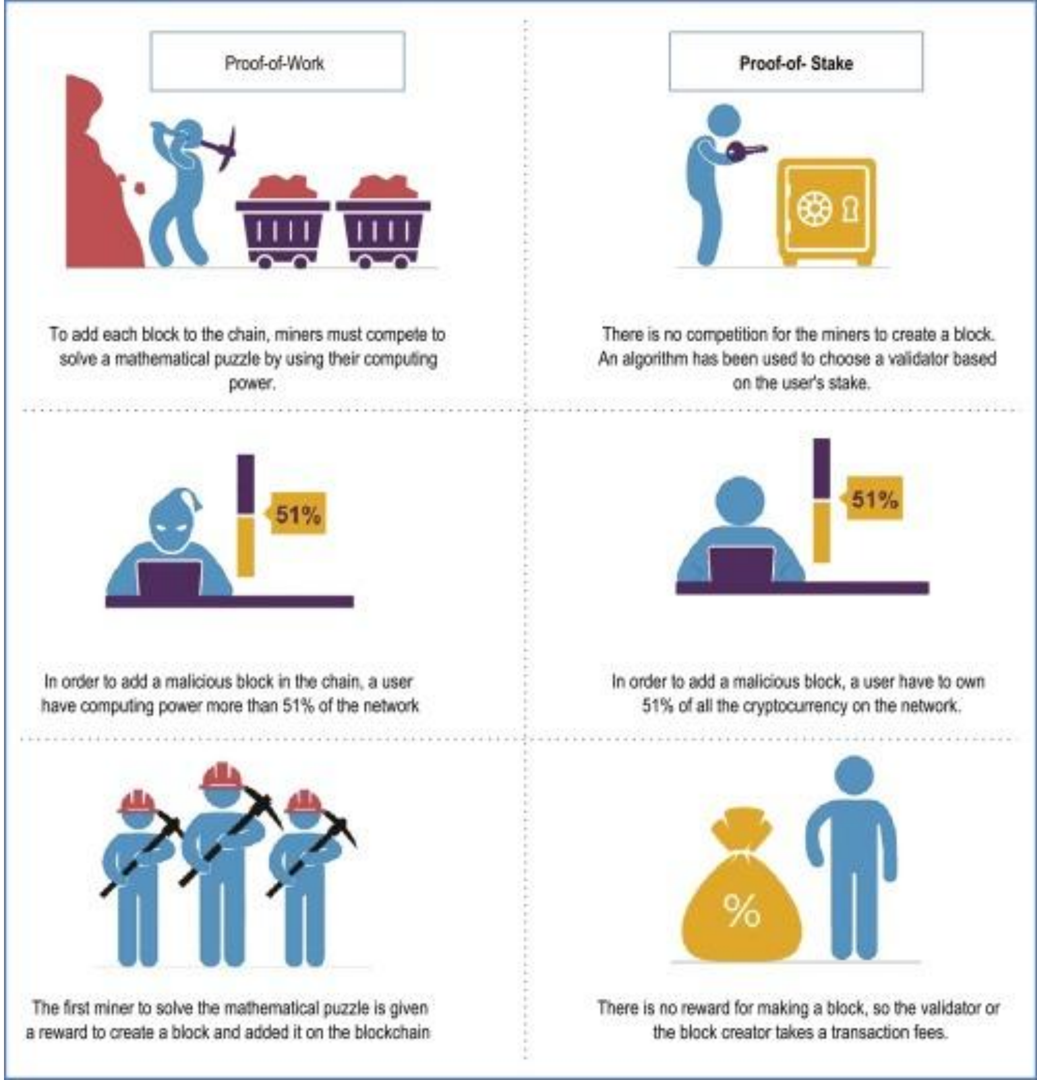
Since each block contains information about the previous block, they effectively form a *chain* (compare linked list data structure), with each additional block linking to the ones before it.

Consequently, blockchain transactions are irreversible in that, once they are recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks.



source:https://www.guru99.com/blockchain-tutorial.html

Consensus Mechanisms



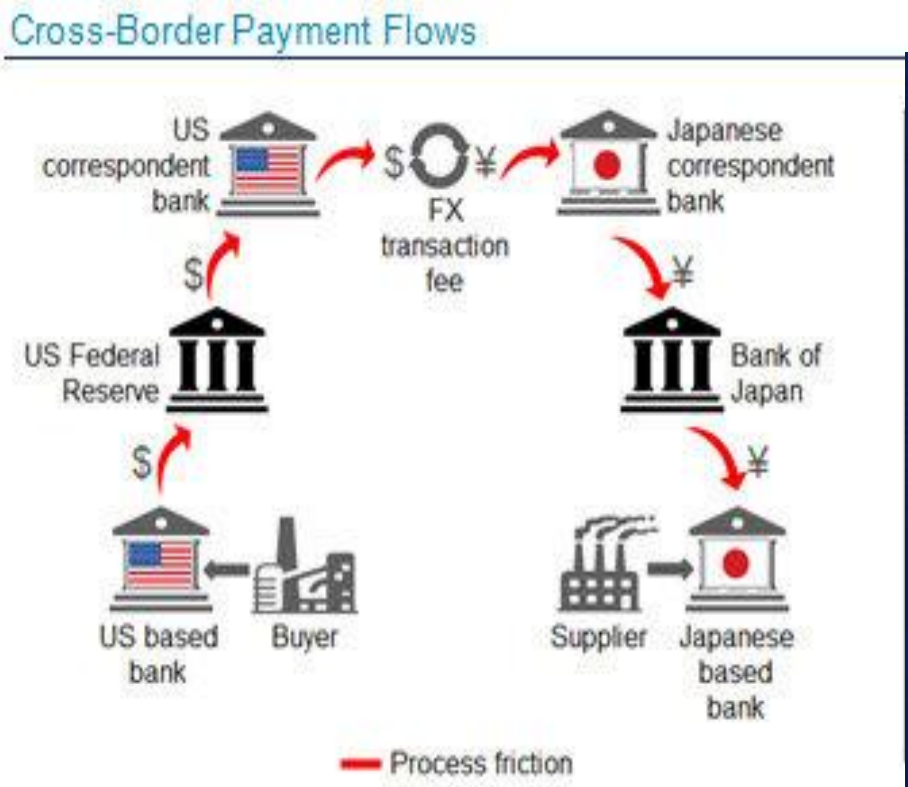
Proof-of-Work	Proof-of- Stake
To add each block to the chain, miners must compete to solve a mathematical puzzle by using their computing power.	There is no competition for the miners to create a block. An algorithm has been used to choose a validator based on the user's stake.
In order to add a malicious block in the chain, a user have computing power more than 51% of the network	In order to add a malicious block, a user have to own 51% of all the cryptocurrency on the network.
The first miner to solve the mathematical puzzle is given a reward to create a block and added it on the blockchain	There is no reward for making a block, so the validator or the block creator takes a transaction fees.

source:https://www.sciencedirect.com/science/article/abs/pii/S0065245820300668

Proof of work

- A **proof of work** is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements.
- In order for a block to be accepted by network participants, miner must complete a proof of work which covers all of the data in the block.
- The difficulty of this work is adjusted so as to limit the rate at which new blocks can be generated by the network to one every 10 minutes.
- Due to the very low probability of successful generation, this makes it unpredictable which worker computer in the network will be able to generate the next block.

Case Study


Cross-Border Payment Flows
US correspondent bank
$
¥
FX transaction fee
Japanese correspondent bank
$
¥
US Federal Reserve
Bank of Japan
$
¥
US based bank
Buyer
Supplier
Japanese based bank
Process friction

- Download Metamask.

- Solidity ERC20 code walkthrough. (Contract: https://goerli.etherscan.io/address/0x27fc6de7a2216de4b91f11506f7f9de11743f1ab#writeContract)

A smart contract contains:

- Version pragma, which specifies the version of the smart contract and ensures that the contract is compiled by the correct version
- State variables, which are used to store the state of the smart contract. They are declared outside of the functions in the smart contract and have initial values. These state variables are accessed via the self object
- Functions, which are executable units of code that are used to perform actions on the smart contract. They can be visible internally or externally, accept arguments, and return values
- Events, which are triggered and logged in the EVM's logging channel
- Interface, which is used to specify the functions that are used by the smart contract
- Structs, which define the types of state variables.

Deployed example here: 0x340339f4eDB1b7DEd9294e3F4B4F221643305eb1

UNISWAP

AAVE

https://medium.com/@maksimjeet/uniswap-v3-deep-dive-7894f7022f56