Ethereum Blockchain

Dayanand Ambawade

Overview

- Introduction to Ethereum Blockchain ,
- Types Network, Nodes, Accounts,
- Smart Contract, DApps,
- EVM,
- GAS,GAS Price,GAS Limit,
- DAO,DAO Attack,
- Hard Fork and Soft Fork,
- ICO, Sharding,
- Ethereum 2.0,
- Alt coins

Review of Blockchain Technology

- What is a blockchain?
- How does a blockchain work?
- Why use a blockchain?

What is a blockchain?

- Shared database consisting of ledger of transactions
- Every stake holder keeps a copy of the ledger and can verify all transactions that are put in the ledger
- Reading/writing on the ledger is completely decentralized and secure
- Fault tolerance
- Independent verification by anyone interested : disintermediation (anyone can audit)

How does Blockchain work?

- Nodes, Transactions, Blocks
- Mining through solving hard problems (solving Byzantine fault-tolerant consensus)
- Hashing for integrity
- Digital Signature for Authenticity and/or authorization
- Permanence (Tamper resistance)

Why use Blockchain?

• Blockchains are used when multiple parties, perhaps located across the world, need to share data and transfer value without trusting each other.

• The financial world describes this trust as the counterparty risk:

  • the risk that the other party won't hold up their end of the bargain.

• Blockchains attempt remove the counterparty risk through a clever usage of mathematics, cryptography, and peer-to-peer networking.

DRAWBACK's OF BITCOIN (ADVANTAGES OF ETHEREUM)

1) Lack of turing complete.

2) Value Blindness.

3) Lack of state.

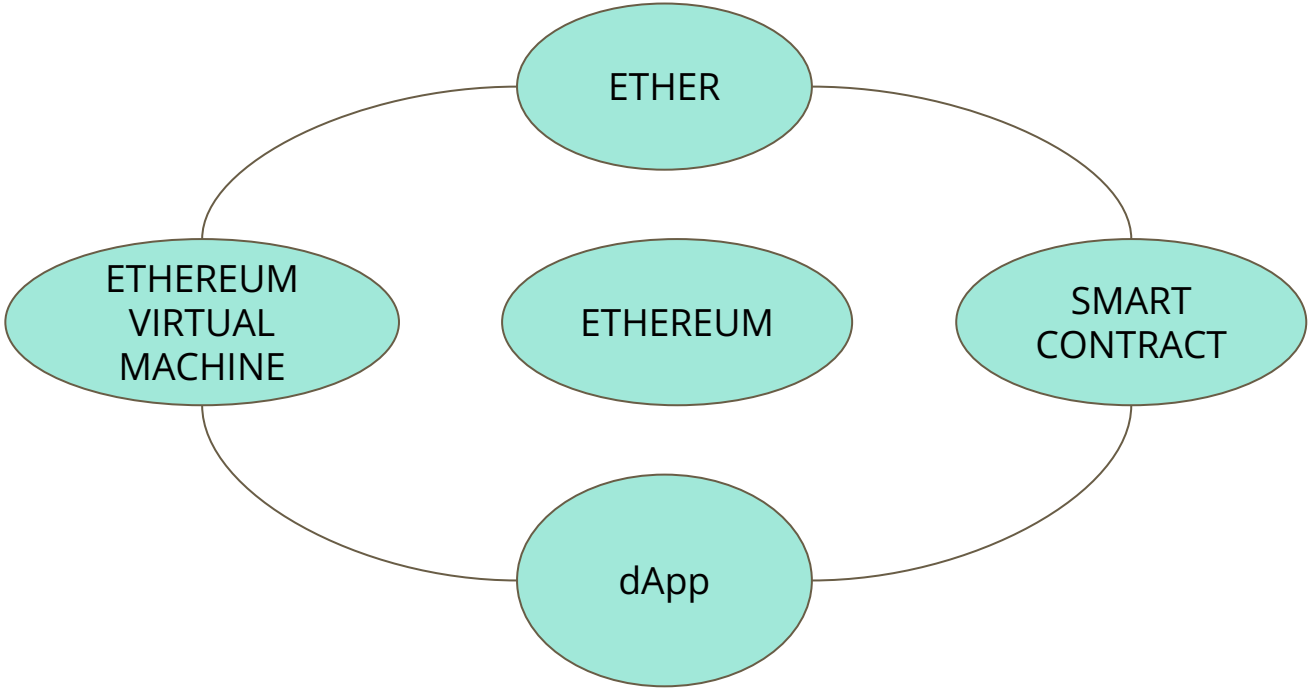
4) Blockchain Blindness.

Creator of Ethereum Blockchain

Introduction to Ethereum Blockchain

- Open source
- Distributed
- Decentralised
- Ledger
- Developed by Vitalik Buterin

Features of Ethereum



What is Ethereum?

• Ethereum is a blockchain that allows you to run programs in its trusted environment, contrasts with the Bitcoin blockchain, which only allows you to manage cryptocurrency.

• Ethereum has a virtual machine -- Ethereum Virtual Machine (EVM).

• The EVM allows code to be verified and executed on the blockchain, which providing guarantees it will be run the same way on everyone's machine.

• This code is contained in "smart contracts"

• Ethereum maintains the state of the EVM on the blockchain.

• All nodes process smart contracts to verify the integrity of the contracts and their outputs.

Types of Ethereum Blockchain

- Public and Permissionless (Ether)
- Private Permissioned

EVM

- A software that can execute Ethereum Bytecode
  - Follows the EVM specifications *(Ethereum protocol)*
  - Runs as a process on a computer/server

  - EVM implemented in multiple languages

Ethereum Virtual Machine (EVM)

- Provides a layer of **abstraction** between the code and the machine
- Also makes the code **portable** across different machines
- The EVM has **140 opcodes** which allow it to be **Turing complete**
- Each opcode takes 1 byte of storage space.
- The EVM also uses a 256 bit register stack which holds 1024 items.
- It also has a contract memory (non-persistent) for complicated operations
- For storing data indefinitely, storage is used.
- Reading from storage is free, but writing to storage is extremely expensive.

Ethereum and Smart Contract

Solidity

Vyper

Remix IDE

What is a smart contract?

- A smart contract is code that runs on the EVM.
- Smart contracts can accept and store ether, data, or a combination of both.
- Using the logic programmed into the contract,
    - it can distribute that ether to other accounts or even other smart contracts.
- Example:
    - Alice wants to hire Bob to build her a patio
    - they are using an escrow contract (a place to store money until a condition is fulfilled) to store their ether before the final transaction.

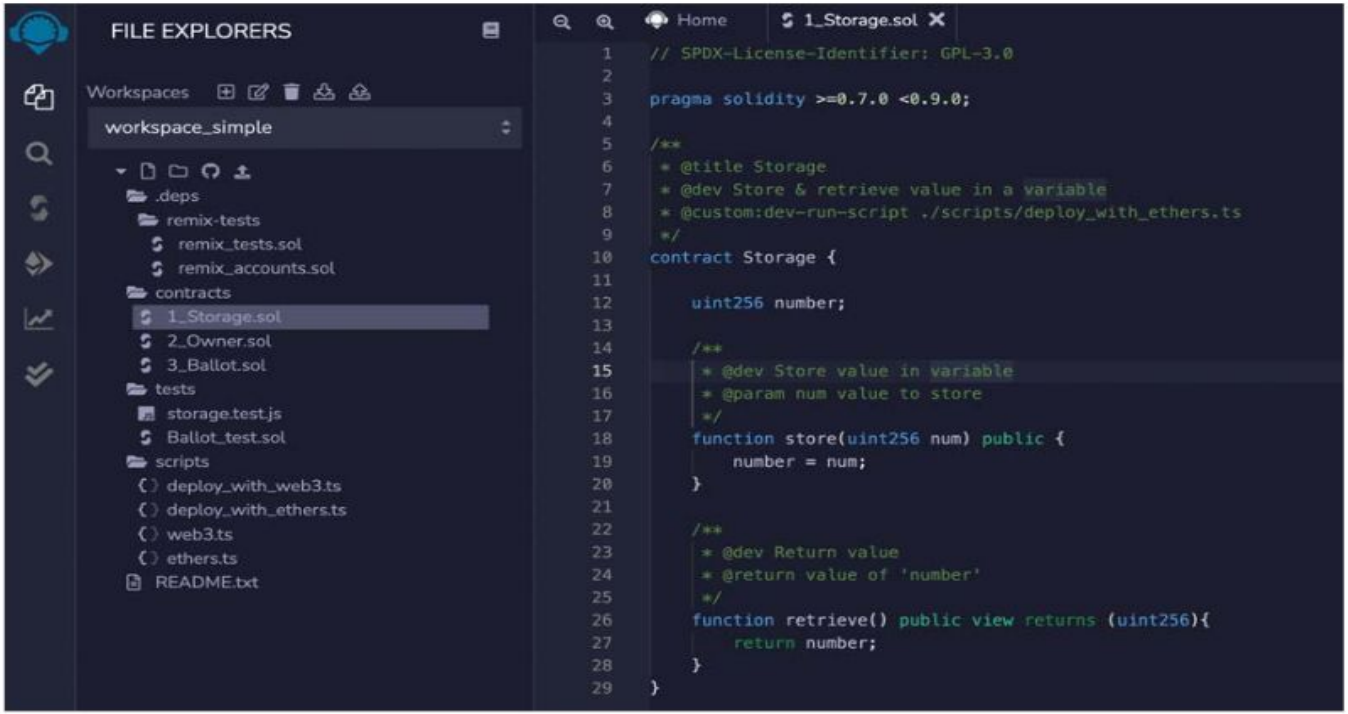
Language of Smart Contracts in Ethereum

- Smart Contracts for Ethereum are written in Solidity
  - Solidity is statically typed
  - supports inheritance, libraries, and complex user-defined types
  - Similar to Javascript syntactically
- To learn solidity go to https://remix.ethereum.org and you can start programming smart contracts without having to create your own Ethereum network

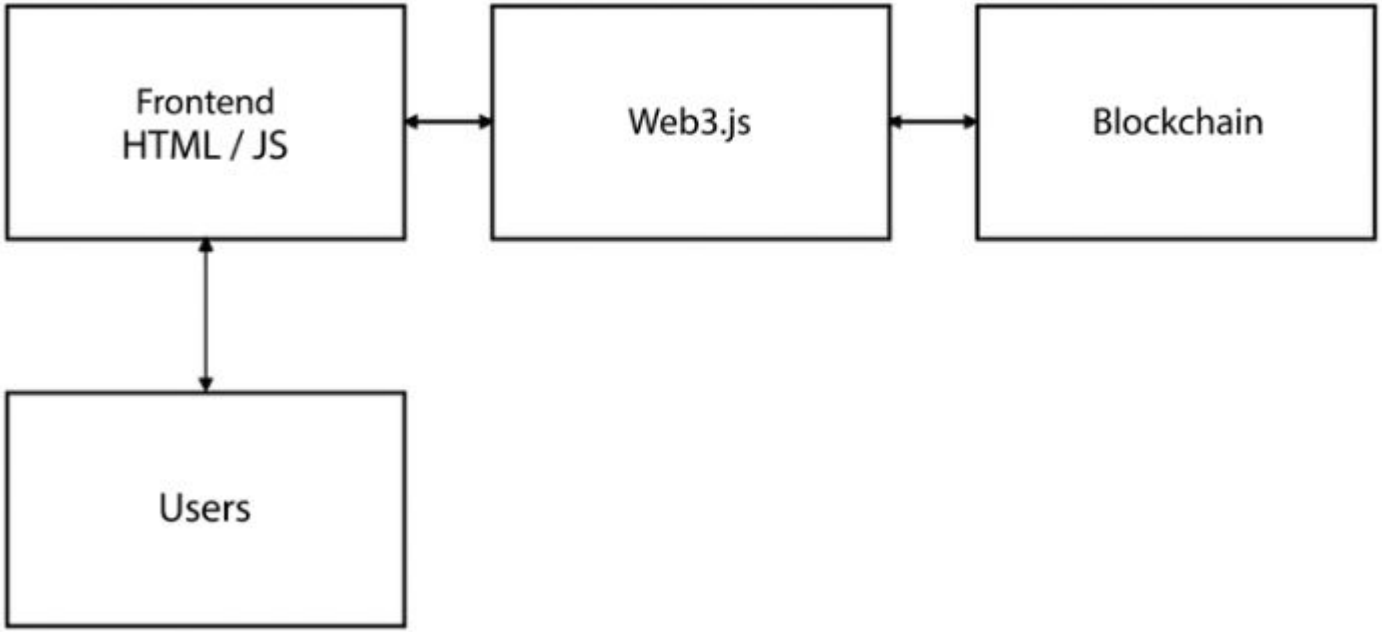
Solidity

- Solidity is the most popular ***programming language*** for Ethereum Smart Contracts
- It is similar to Javascript and C++
- Other experimental languages like Vyper and Bamboo are not much in use
- Solidity Compiler ***(solc)*** converts the source code to EVM bytecode

Remix IDE


FILE EXPLORERS
Workspaces
workspace_simple
.deps
remix-tests
remix_tests.sol
remix_accounts.sol
contracts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
tests
storage.test.js
Ballot_test.sol
scripts
deploy_with_web3.ts
deploy_with_ethers.ts
web3.ts
ethers.ts
README.txt
Home
1_Storage.sol
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
/**
* @title Storage
* @dev Store & retrieve value in a variable
* @custom:dev-run-script ./scripts/deploy_with_ethers.ts
*/
contract Storage {
uint256 number;
/**
* @dev Store value in variable
* @param num value to store
*/
function store(uint256 num) public {
number = num;
}
/**
* @dev Return value
* @return value of 'number'
*/
function retrieve() public view returns (uint256){
return number;
}
}

web3.js, frontend, and blockchain interaction architecture



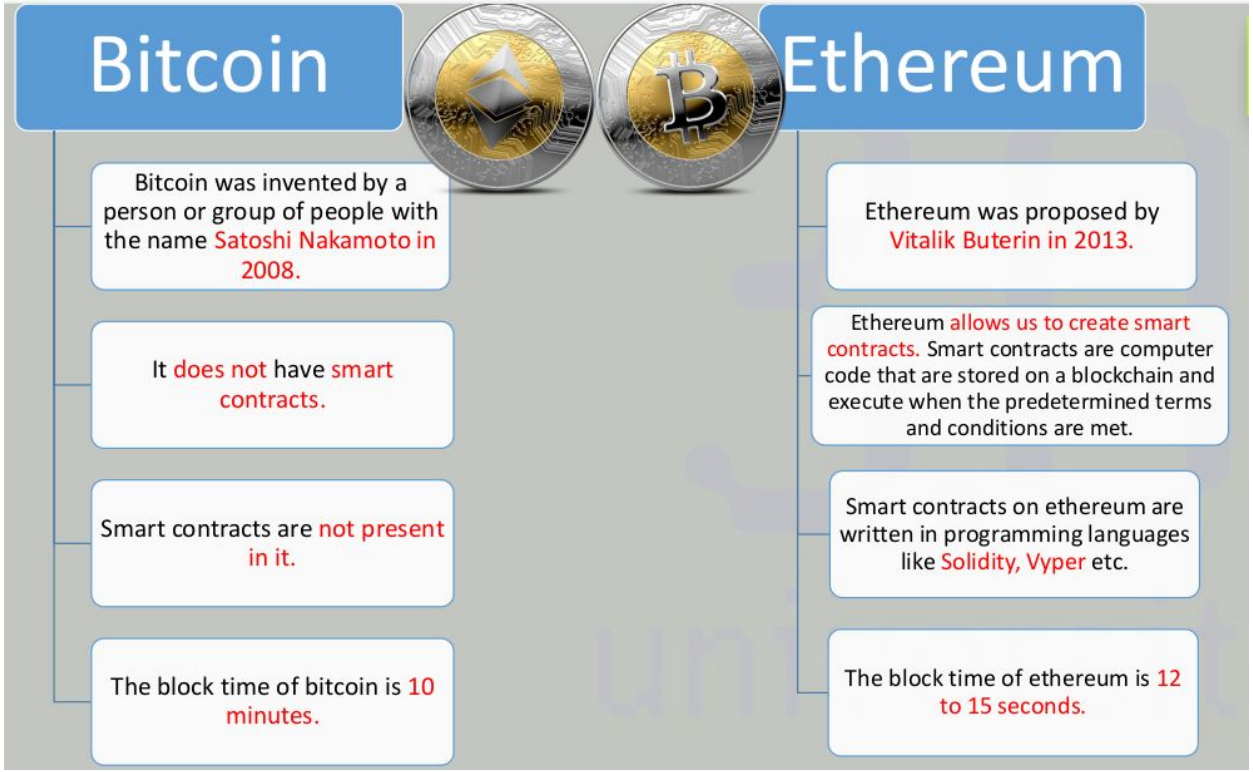
Why Should I use Smart Contracts?

- Accuracy
- Efficiency
- Trust and Transparency
- Security
- Savings
- Guaranteed Outcomes

Examples where smart contracts can be used.

- Digital voting during political elections
- Smart contracts can also be used in the organizations as a communication and workflow management tool.
- Smart contracts can be used effectively in the real estate transactions.
- Smart contracts can also be used in healthcare industry

Bitcoin vs Ethereum



The Ethereum Network

The mainnet:

The mainnet is the current live network of Ethereum. Its network ID is 1 and its chain ID is also 1.

The network and chain IDs are used to identify the network.

A block explorer that shows detailed information about blocks and other relevant metrics is available at

https://etherscan.io .

This can be used to explore the Ethereum blockchain.

The Ethereum Network

Testnets

There is a number of testnets available for Ethereum testing.

The aim of these test blockchains is to provide a testing environment for smart contracts and DApps before being deployed to the production live blockchain.

A being test networks, they also allow **experimentation and research.** The main testnet is called **Ropsten**, which contains all the features of other smaller and special-purpose testnets that were created for specific releases. For example, other testnets include **Kovan and Rinkeby,** which were developed for testing Byzantium releases. The changes that were implemented on these smaller testnets have also been implemented in Ropsten. Now the Ropsten test network contains all properties of Kovan and Rinkeby.

The Ethereum Network

Private nets:

The private networks that can be created by generating **a new genesis block.** This is usually the case in private blockchain networks, where a private group of entities start their blockchain network and use it as a permissioned or consortium blockchain.

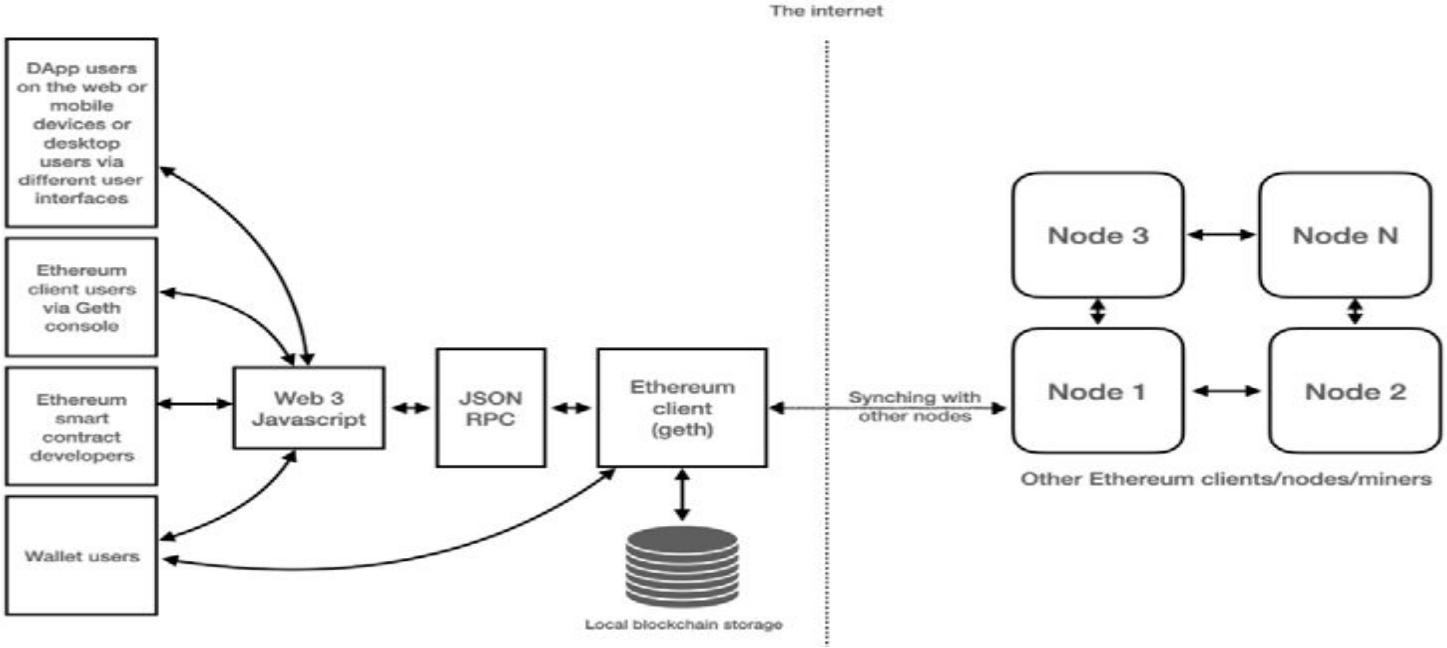
Other Ethereum Networks

- The Ethereum blockchain can be simulated locally for development.
- Local test networks process transactions instantly and Ether can be distributed as desired.
- An array of Ethereum simulators exist;
    - **Ganache**
- Developers use public test networks (or testnets) to test Ethereum applications before final deployment to the main network.
- Ether on these networks is used for testing purposes only and has no value.

Private/Enterprise Networks

- Private Ethereum networks allow parties to share data without making it publicly accessible.
- A private blockchain is a good choice for:
    - Sharing of sensitive data, such as health care records
    - Scaling to handle higher read/write throughput, due to the smaller network size
- An example of a private enterprise blockchain is Quorum, originally written by J.P. Morgan.

Components of Ethereum Ecosystem



Components of Ethereum Blockchain

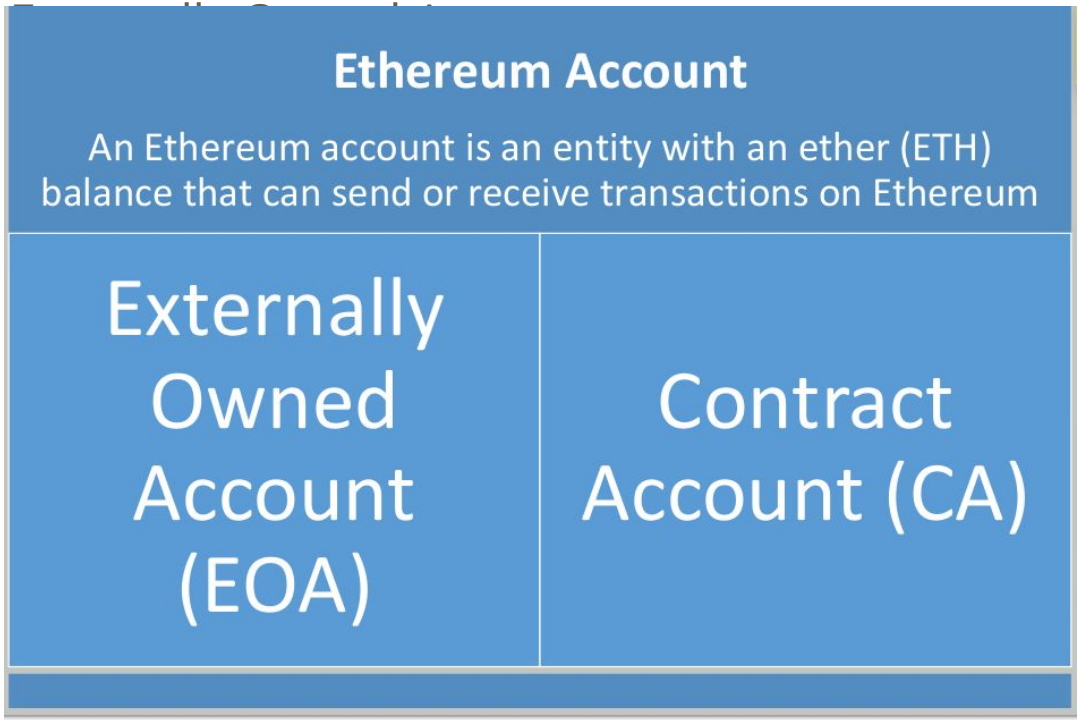
A list of elements present in the Ethereum blockchain is:

- Keys and addresses
- Accounts
- Transactions and messages
- Ether cryptocurrency/tokens
- The EVM
- Smart contracts and native contracts

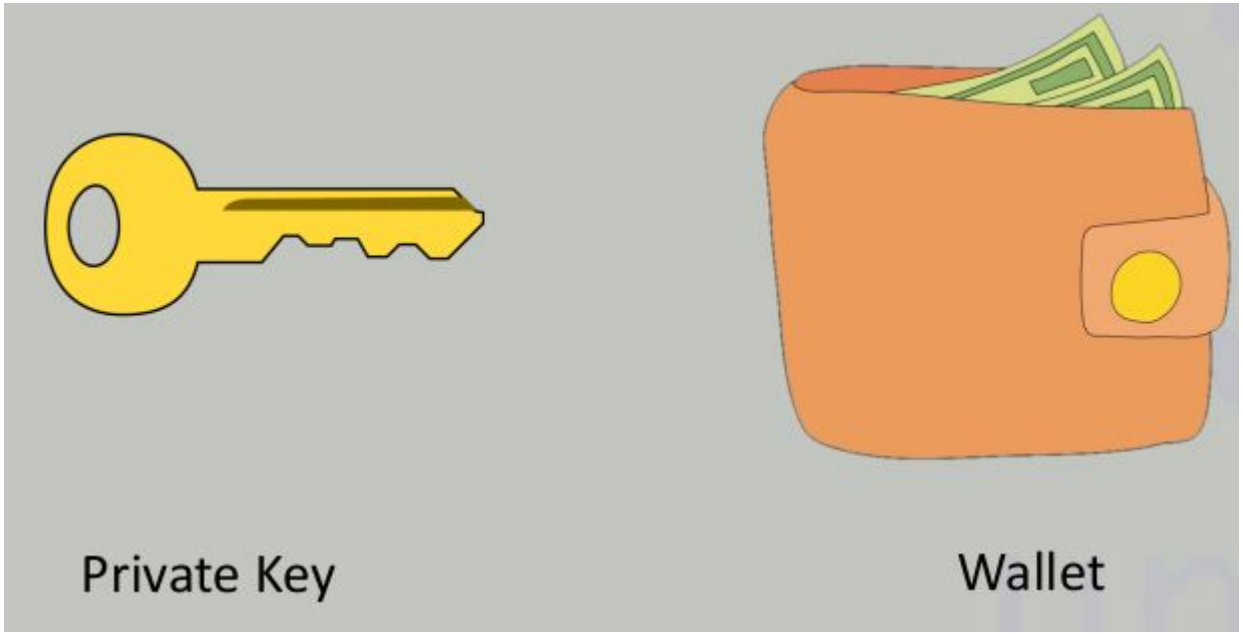
Ethereum Nodes

- Full Node
- Lite Node
- Archive Node

Ethereum Accounts



Ethereum Accounts



EOA vs CA

EOA	CA
Private Key is needed	No private or public key is needed
Controlled by Human	Controlled by Contract code
Has a unique address	Has a unique address
Holds ETH balance	Holds ETH balance

Distributed Applications (Dapps)

• Applications using smart contracts for their processing are called "distributed applications", or "dapps".

• The user interfaces for these dapps consist of familiar languages such as HTML, CSS, and JavaScript.

• The application itself can be hosted on a traditional web server or on a decentralized file service such as Swarm or IPFS.

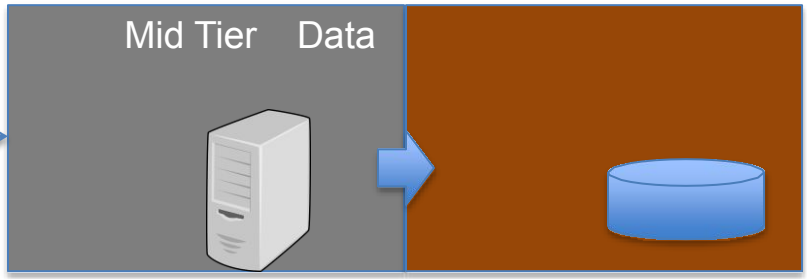
• Dapps based solution available for:

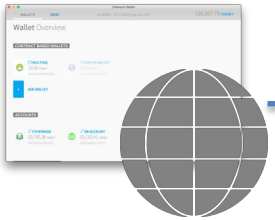
- Record keeping
- Finance
- Supply chains
- Real estate
- Marketplaces

Web App → DAPP

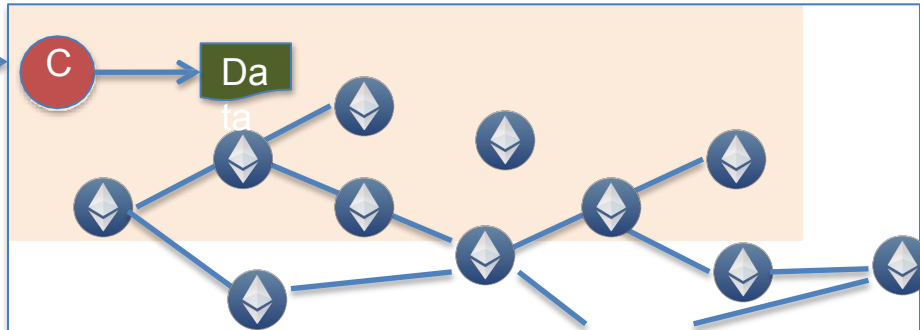
Front end apps

Centralized Resources
Owned by the organization





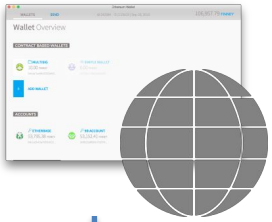
Decentralized Resources
Public domain



Working of Dapp

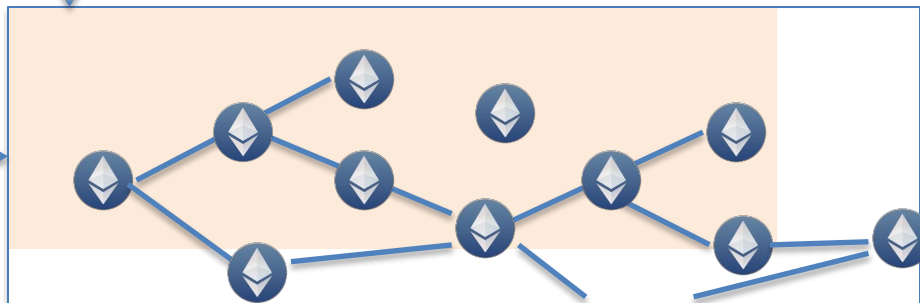
Invoke Contract

- App user pays *gas/fee*



- Manage funds
- Invoke Contracts

- Miner collects
- Transaction validated/mined
- Recorded in ledger



Example DAPP
Bidding

Ship
Goods

Seller Application

Event: BidReceived

Buyer Application

withdrawFunds()

Bid()

Contract

DAPP Technology Stack

web3

OLIDITY

Serpent

Lisp Like Language

Blockchain explorer

- Websites (or webapps) that show information on
  - Transactions
  - Blocks
  - Accounts

https://etherscan.io/

https://testnet.etherscan.io/

https://live.ether.camp/

Etherchain.org

https://etherchain.org/

Ethereum Ether, GAS

Ether

GAS

GAS Price

GAS Limit

Ether and Gas

Ether is the native cryptocurrency of the Ethereum Network

Gas is a unit to measure the computational work done.

- Introduced as it would have been unfair to base the transaction fees on just the transaction length or keep it constant
- Each operation has a **gas cost**
- Every transaction mentions the **gas price**
- The two together give the transaction fees in Ether
- Two new scenarios -
    - Transaction running out of gas
    - Gas price too low/high

Different units of Ether Gas

Denominations of Ether		
Unit Name	Wei Value	Number of Wei
Wei (wei)	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
Twei (szabo)	1e12 wei	1,000,000,000,000
Pwei (finney)	1e15 wei	1,000,000,000,000,000
Ether (buterin)	1e18 wei	1,000,000,000,000,000,000

Decentralized Autonomous Organization(DAO)

Software acting as an organization with predefined rules and procedures using smart contracts on blockchain.

Decentralized : No central authority, governed entirely by its individual members, each and every member partake for decision making.

• **Autonomous :** Having the freedom to govern itself or control its own affairs.

Organization : Group of people with a particular purpose Community-led entity with no central authority.

Smart contracts lay the foundational rules, execute when agreed upon decisions, and at any point, proposals, voting, and even the very code itself can be publicly audited

Decentralised Autonomous Organization (DAO)

What is DAO term?

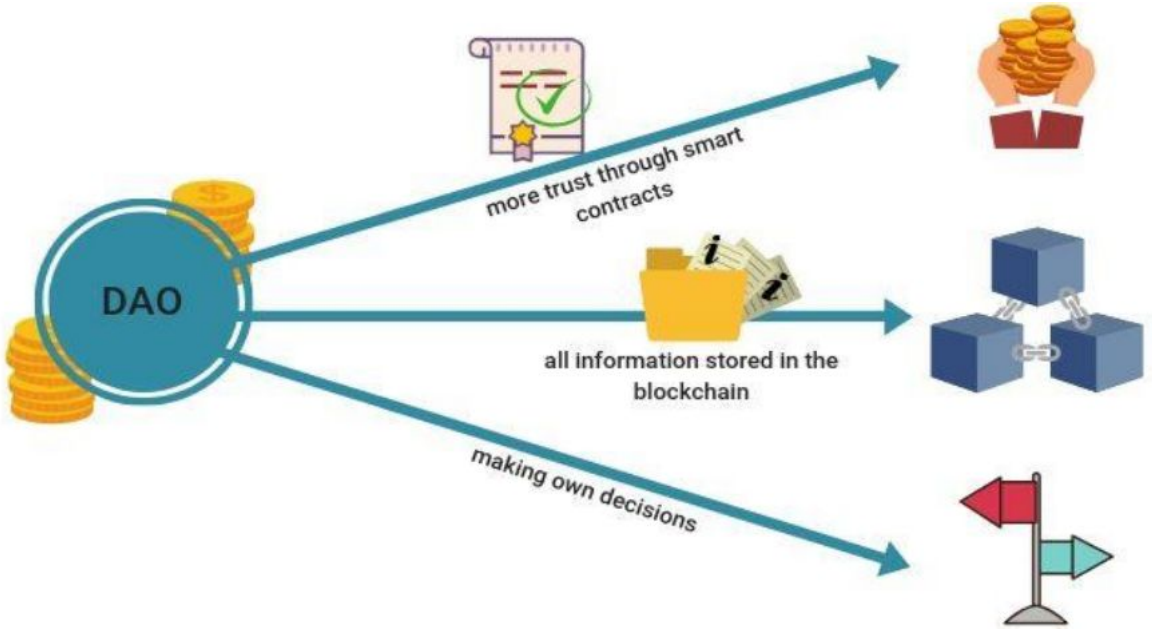
The term DAO stands for “decentralized autonomous organization” and can be described as **an open-source blockchain protocol governed by a set of rules, created by its elected members, that automatically execute certain actions without the need for intermediaries**. [Wikipedia]

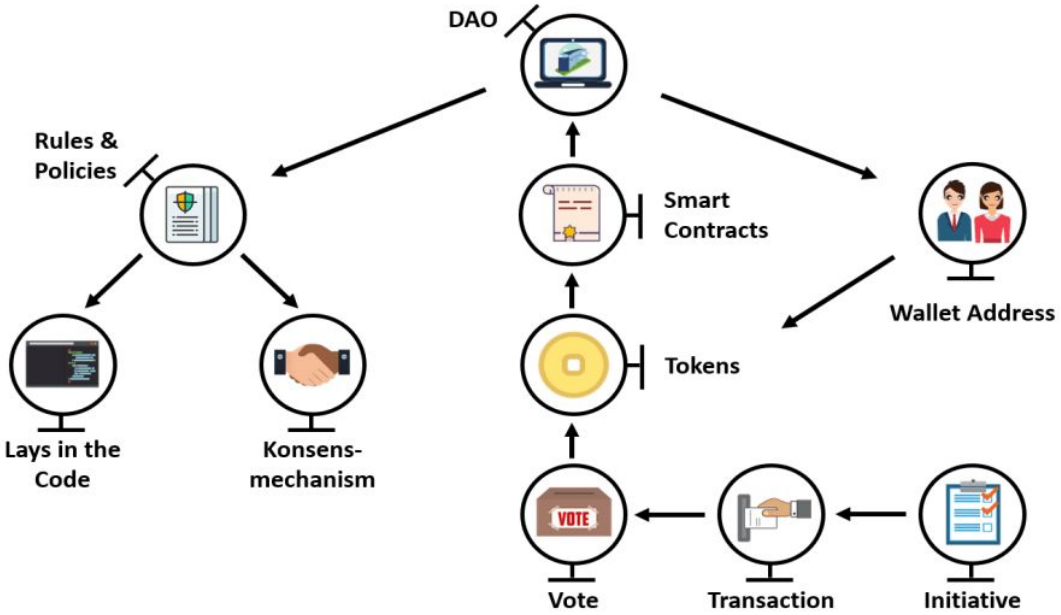
What is a DAO in blockchain?

A DAO is **a way of organizing people and their interests on the internet using the blockchain**. The blockchain is a public ledger system that exists only on the internet. It uses a complex cryptography system to ensure that everything written to it (“blocks”) is verifiable.

What is DAO and how it works?
DAO stands for decentralized autonomous organization, which is a term for **a group of people who agree to abide by certain rules for a common purpose**.
Those rules are written into the code of the organization via smart contracts—algorithms that run when certain criteria are met.

Functionality of DAO

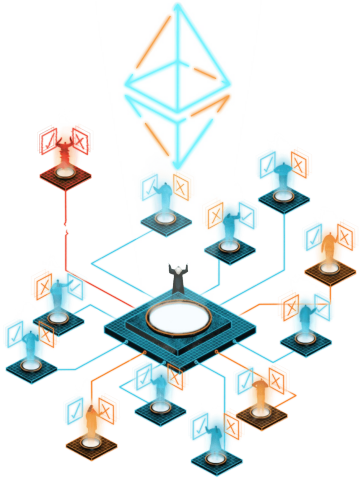



DAO
Rules &
Policies
Smart
Contracts
Wallet Address
Tokens
Lays in the
Code
Konsens-
mechanism
VOTE
Vote
Transaction
Initiative

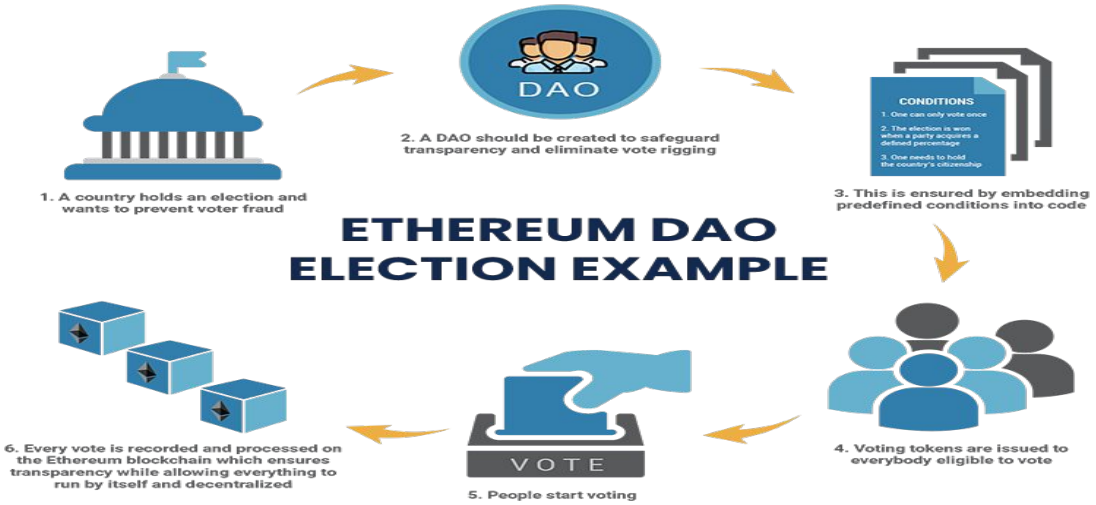
DAOs

What is the purpose of a DAO?

Fueled by ether, the DAO was designed to **allow investors to send money from anywhere in the world anonymously**. The DAO would then provide those owners tokens, allowing them voting rights on possible projects.



Ethereum DAO Example



Ethereum Improvement Proposals (EIPs)

Dayanand Ambawade

Ethereum Improvement Proposals

EIP stands for Ethereum Improvement Proposal.

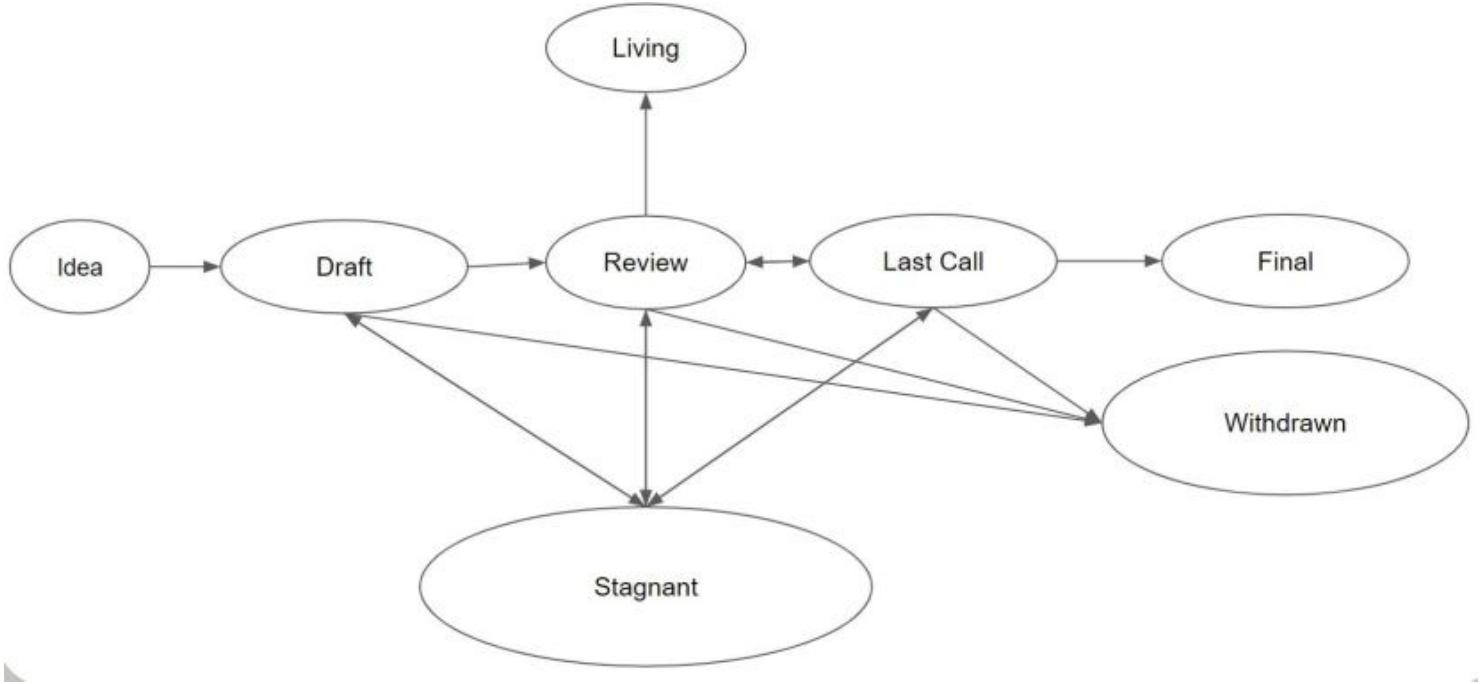
• An EIP is a design document providing information to the Ethereum community or describing a new feature for Ethereum or its processes or environment.

• The EIP should provide a concise technical specification of the feature and a rationale for the feature.

• The EIP author is responsible for building consensus within the community and documenting dissenting opinions.

• Ethereum Improvement Proposals (EIPs) describe standards for the Ethereum platform, including core protocol specifications, client APIs, and contract standards. Network upgrades are discussed separately in the Ethereum Project Management repository.

EIP Process


Living
Idea
Draft
Review
Last Call
Final
Withdrawn
Stagnant

EIP Process Term

Idea - An idea that is pre-draft. This is not tracked within the EIP Repository.

Draft - The first formally tracked stage of an EIP in development. An EIP is merged by an EIP Editor into the EIP repository when properly formatted.

Review - An EIP Author marks an EIP as ready for and requesting Peer Review.

Last Call - This is the final review window for an EIP before moving to FINAL. An EIP editor will assign Last Call status and set a review end date (`last-call-deadline`), typically 14 days later. If this period results in necessary normative changes it will revert the EIP to Review.

Final - This EIP represents the final standard. A Final EIP exists in a state of finality and should only be updated to correct errata and add non-normative clarifications.

Stagnant - Any EIP in Draft or Review if inactive for a period of 6 months or greater is moved to Stagnant. An EIP may be resurrected from this state by Authors or EIP Editors through moving it back to Draft.

Withdrawn - The EIP Author(s) have withdrawn the proposed EIP. This state has finality and can no longer be resurrected using this EIP number. If the idea is pursued at later date it is considered a new proposal.

Living - A special status for EIPs that are designed to be continually updated and not reach a state of finality. This includes most notably EIP-1.

EIP Types

EIPs are separated into a number of types, and each has its own list of EIPs.

- Standard Track
- Core
- Networking
- Interface
- ERC
- Meta
- Informational

Hard Fork & Soft Fork In Blockchain.

What is Fork in Blockchain:

A fork is a change to the protocol, or a divergence from the previous version of

the Blockchain. When a new, alternative block is generated by a rogue miner, the

system reaches consensus that this block is not valid, and this is very soon

abandoned by the other miners.

Hard Fork

A Hard Fork is a rule change such that the software validating according to the old rules will

see the blocks produced according to the new rules as invalid. In case of a hard fork, all

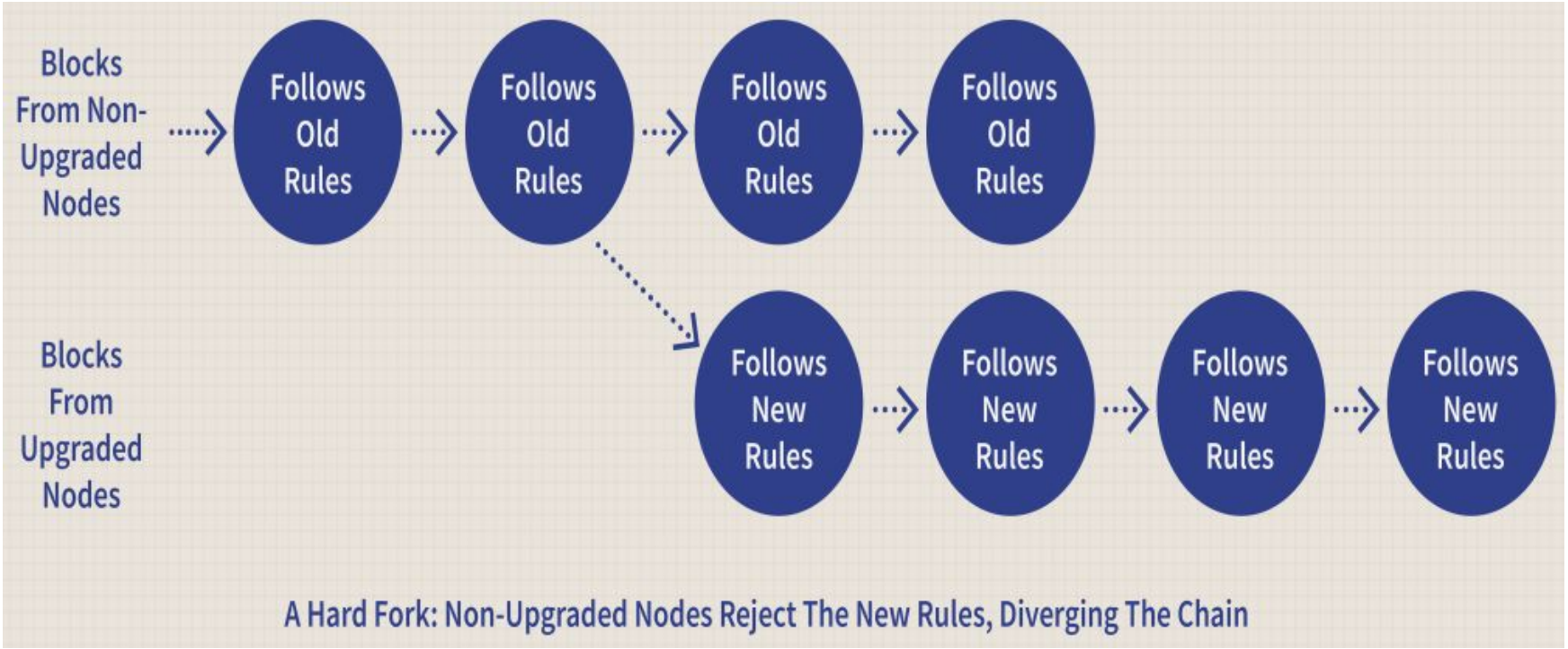
nodes meant to work in accordance with the new rules need to upgrade their software. If one

group of nodes continues to use the old software while the other nodes use the new software,

a permanent split can occur.

• Hard fork is reversible in nature.

Hard Fork



A Hard Fork: Non-Upgraded Nodes Reject The New Rules, Diverging The Chain

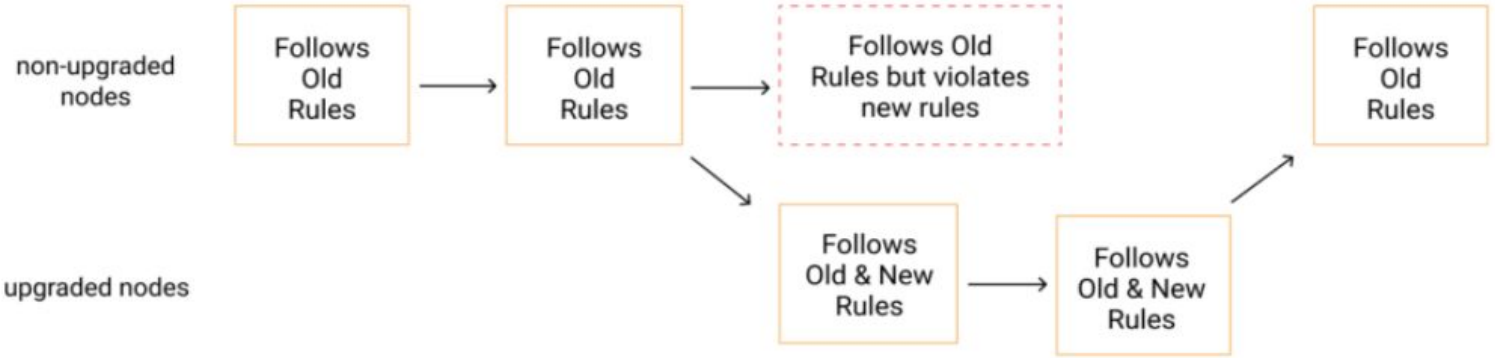
Soft Fork

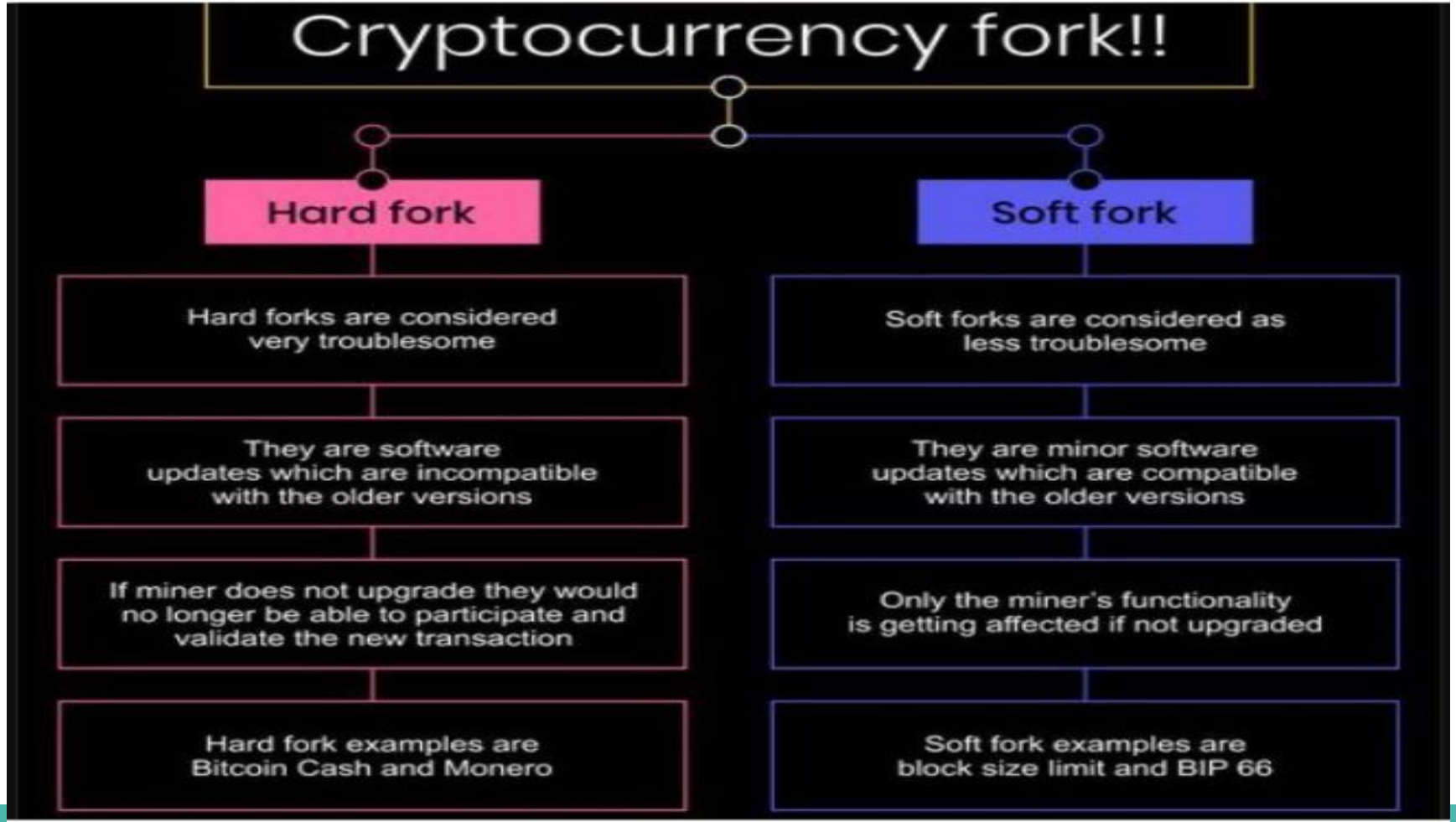
• A soft fork or a soft-forking change is described as a fork in the Blockchain which can occur when old network nodes do not follow a rule followed by the newly upgraded nodes. This could cause old nodes to accept data that appear invalid to the new nodes, or become out of sync without the user noticing.

• Soft fork is Irreversible.

Soft Fork

Blocks Violationg New Rules are made stale by the upgraded mining majority




Cryptocurrency fork!!
Hard fork
Soft fork
Hard forks are considered very troublesome
They are software updates which are incompatible with the older versions
If miner does not upgrade they would no longer be able to participate and validate the new transaction
Hard fork examples are Bitcoin Cash and Monero
Soft forks are considered as less troublesome
They are minor software updates which are compatible with the older versions
Only the miner's functionality is getting affected if not upgraded
Soft fork examples are block size limit and BIP 66

Hardfork and Softfork

What is Forking?

As the blockchain progresses (more blocks are added to the blockchain) governed by the consensus mechanism, on occasion, the blockchain can split into two. This phenomenon is called **forking.**

Hardfork

Softfork

Initial Coin Offerings (ICOs)

ICOs

Crowdfunding

Conclusion

We cover history of Ethereum, the motivation behind Ethereum development, and Ethereum clients.

we introduced the core concepts of the Ethereum blockchain, such as the state machine model, the world and machine states, accounts, and types of accounts.

A detailed introduction to the core components of the EVM

.....

Introduction to Blockchain Technology:

Definition of Blockchain,Elements of Blockchain Technology,The architecture of Blockchain Technology

[1] What is Blockchain? Simple Explanation

https://youtu.be/SSo_EIwHSd4?list=RDLVSSo_EIwHSd4

[2] What is BLOCKCHAIN? The best explanation of blockchain technology

https://youtu.be/3xGLc-zz9cA?list=RDLVSSo_EIwHSd4

[3] Blockchain 101- A Visual Demo by Anders Brownworth

https://youtu.be/SSo_EIwHSd4?list=RDLVSSo_EIwHSd4

[4] Blockchain 101 - Part 2 - Public / Private Keys and Signing

https://youtu.be/xIDL_akeras?list=RDCMUCx7YSpyYtgaYik1Wen9jYSA

[5] Bitcoin mining with 15 lines of python code | Python Bitcoin Tutorial

https://youtu.be/ZhnJ1bkIWWk

References

[1] The Bitcoin whitepaper was published in 2008 under the pseudonym Satoshi Nakamoto.

https://bitcoin.org/bitcoin.pdf

[2] Know About The Best Blockchain Open Source Projects

https://101blockchains.com/blockchain-open-source/

[3] Understand the Blockchain in Two Minutes

https://youtu.be/r43LhSUUGTQ

[4] But how does bitcoin actually work?

https://youtu.be/bBC-nXj3Ng4

[5] The Blockchain-Enabled Future of Telecommunications by Dennis Carroll

https://slideplayer.com/slide/17245257/