MODULE 4:

PUBLIC BLOCKCHAIN

Prepared By: Prof. Swapnil S. Sonawane

ETHEREUM COMPONENTS

Miner and Mining node:

A miner is responsible for writing a transaction to the Ethereum series.

Miners get two kinds of rewards: benefit for writing a block into the series and accumulative gas prices

There are two types of nodes (computers) in the Ethereum network: Mining nodes and Ethereum virtual machines.

Mining nodes refer to the nodes that belong to miners and EVM has a unique code attached to it, which is called as smart contract. Mining nodes are part of the same network where EVM is hosted.

Each mining node maintains its version of Ethereum ledger, and the ledgers could be identical. It is the job of the miners to ensure that their ledger is always upgraded with the latest blocks.

Ethereum Virtual Machine (EVM):

A blockchain system is made of numerous nodes belonging to both miners and also a few who do not want to mine but function as aids for the implementation of smart contracts. These nodes are called Ethereum Virtual Machines (EVM).

EVMs are mostly in charge of supplying a run time that could perform code compiled from smart contracts.

It does not need access to the ledger; however, it contains only limited information regarding the current transaction.

Ether:

Ether is a type of cryptocurrency used in Ethereum. To set up a smart contract using the Ethereum platform, the contract author needs to pay in the form of ether. That is done so people might write optimized codes that do not waste the Ethereum network's computing ability on unnecessary tasks.

Gas:

Gas is the internal currency of Ethereum. The execution and resource utilization costs are predetermined in Ethereum in terms of Gas units. This is also known as Gas Cost.

Transactions in Ethereum:

Ethereum stores transactions within Blocks. A transaction (Contract) is simply a set of agreements between parties; there would be an exchange of assets, products, or services in place of currency, cryptocurrency, or some other asset either in the present or in the future.

Ethereum Accounts:

Ethereum supports two kinds of accounts. Every account comes with a balance that yields the present value saved in it. Once somebody creates an externally owned account on Ethereum, a public-private key is produced.

An externally owned account may keep Ether in its balance and not have any code related to it.

Normal contract accounts are similar to externally owned accounts. They are identified by the public address and do not hold any private keys. They can carry ether very similar to externally owned accounts. However, they have a programming code for smart contracts comprising of state variables and functions.

Swarm and Whisper:

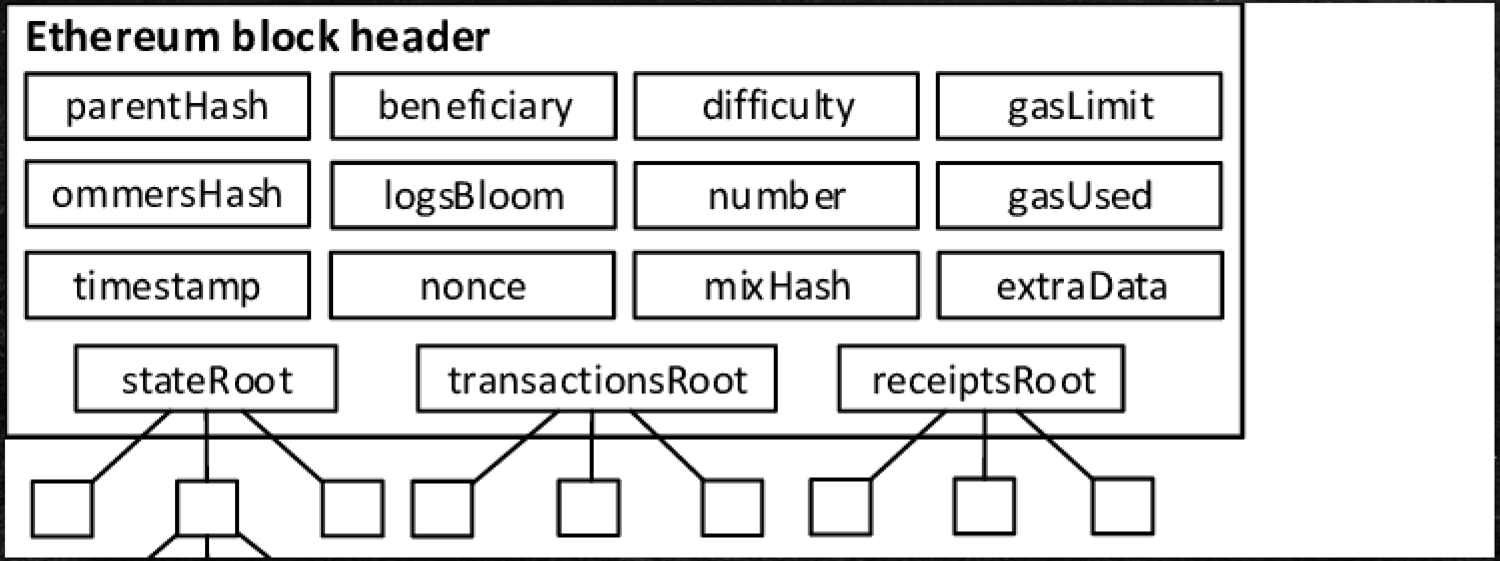
Swarm is a peer to-peer document sharing platform, similar to Bit Torrent, incentivized with micro payments of ETH File records are divided into small chunks, dispersed, and stored with volunteers. The nodes which save and also function (calculate, store, and communicate) on those pieces are all paid with ETH from people using the information. Whisper is an encrypted messaging protocol that enables nodes to send out messages directly to each other, in a secure way that additionally hides the sender and recipient identity from third-party snoopers.

Ethash

Ethereum mining used an algorithm known as Ethash. The fundamental idea of the algorithm is that a miner tries to find a nonce input using brute force computation so that the resulting hash is smaller than a threshold determined by the calculated difficulty. A miner who discovers a block that can be inserted to the blockchain receives a block reward comprising of three ethers. This gas price is credited to the miner's accounts included as proof-of-work mining procedure an additional reward for adding transactions as a portion of this block.

MINING IN ETHEREUM

The mining principle of Ethereum is different from that of bitcoin. Ethereum miners always look forward to mine new blocks and listen actively to receive new blocks from other miners. The endeavor of miners will always be to build on the block header and perform the subsequent task.



Parent Hash:

The miner need to identify hash of previous block. The hash of previous block or parent block will be added to current block header

LogsBloom:

It is the data structure that consist of log information

Ethereum Accounts:

Ethereum supports two kinds of accounts. Every account comes with a balance that yields the present value saved in it. Once somebody creates an externally owned account on Ethereum, a public-private key is produced.

Transaction, state, and receipt hash:

The miner hashes all transactions that are inside the block; the hashes are further combined in pairs to create a brand new hash.

The hash is also known as Root transaction hash or Merkle Root transaction hash. The miner, in the same manner, computes the State transaction hashes and Receipt transaction hashes and adds to the block header.

Nonce:

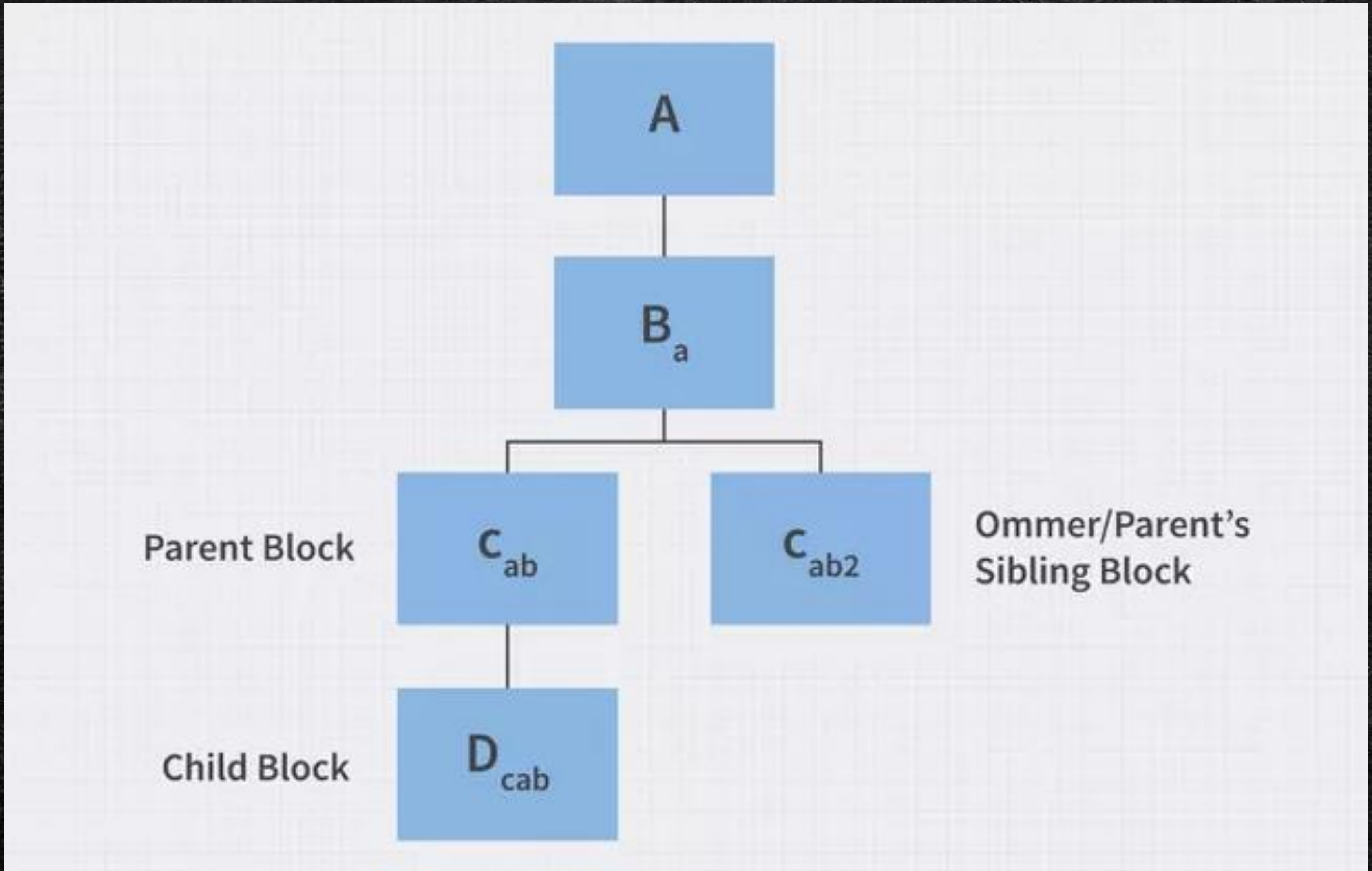
A nonce is a random number that can be used just once in the cryptographic communication.

Timestamp (Current UNIX time):

The timestamp is another field, which indicates the UNIX time. It is the seconds passed after the first of January 1970 and is a 10-digit number.

Ommer block:

Ommer blocks are created in the Ethereum blockchain when two blocks are created and submitted to the ledger at roughly the same time. Only one can enter the ledger.



A

B_a

Parent Block

C_{ab}

C_{ab2}

Ommer/Parent's Sibling Block

Child Block

D_{cab}

MixHash:

MixHash can be a hash which, when together with all nonce, demonstrates that a block has enough computation. Miners generate a mixhash until the outcome is the desired target hash.

Difficulty:

It decides the complexity of the puzzle/challenge awarded to miners of a particular block. The gas limit of the block decides the most gas allowed for the block.

Number:

The number is the sequential number of a block on the chain.

ARCHITECTURE OF ETHEREUM

In a decentralized application, we need consensus and the blockchain was that missing ingredient to reach consensus in a decentralized way. So, for consensus (Layer 0) , we need some way to agree upon all of these application level constructs

Layer 5: Dapps
Layer 4: Browsers
Layer 3: Inter operability
Layer 2: Blockchain services
Layer 1: Economic layer
Layer 0: Consensus layer

We want an economic token (Layer 1) to incentivize each one of the nodes, either to do the computation, or perform the storage operation .This is where the tokens get involved.

Layer 2, we desire a brand-new form of decentralized storage, which is peer-to-peer reviewed, along with IPFs (interplanetary file system). In addition, we also have smart contracts.

On top of that, we have interoperability (Layer 3). In this decentralized world, where we have all of these different apps, each with their tokens, how is one supposed to exchange value between these tokens?

The answer is to write a universal wrapper around all cryptocurrencies. This exchange protocol (of Layer 3) would transfer the amount or transmit value between all of these different tokens as one uses the various services.

Next, we have the browser (Layer 4) to access the decentralized applications. Some browsers are made for decentralized applications, like Mist, Omni wallet, and Maelstrom. After this layer, we build DApps (Layer 5), where we start adding application-level constructs.

WORKFLOW OF ETHEREUM

Solidity is the native coding language of Ethereum. It creates a .sol file as an outcome. Solidity Compiler (SOLC) is used to compile the solidity file (sol).

Bytecode invokes Web3js for deployment. Once it is deployed, it returns the Contract and the Application Binary Interface (ABI).

When this contract is initiated, it can invoke contract methods and signs, and pass them to Ether to perform the Operation.

Workflow for Deploying Smart Contracts:

- The first step is to download the Ethereum node.
- Then, write the solidity code. Compile it using a framework like a truffle.
- Deploy the smart contract code to the network.
- Call the deployed contract using Web3js, which is the front-end client that speaks to the Ethereum blockchain.

COMPARISON BETWEEN BITCOIN AND ETHEREUM

	Bitcoin	Ethereum
Founder	Satoshi Nakamoto	Vitalik Buterin and Team
Purpose	Cryptocurrency	Network Software
Release Date	January 2009	July 2015
Scripting language	Turing Incomplete	Turing Complete
Coin release method	Early mining	Through ICO
Average block time	Approx. 10 mins	Approx. 15 seconds
Transaction Model	UTXO	Account

	Bitcoin	**Ethereum**
Coin symbol	BTC	ETH
Tokens	Not available	Available
Monetary policy	Hardcoded	Not hardcoded
Emission rate	Halving policy followed	Occasional
Backward compatibility	Available	Not available
Block limitation	1 MB per block	NO limit

TYPES OF TEST NETWORKS

There are three test nets currently in use, and each behaves similarly to the production blockchain (where your real Ether and tokens reside).

1. Ropsten: A proof-of-work blockchain that most closely resembles Ethereum; you can easily mine faux-Ether.
2. Kovan: A proof-of-authority blockchain, started by the Parity team. Ether can’t be mined; it has to be requested.
3. Rinkeby: A proof-of-authority blockchain, started by the Geth team. Ether can’t be mined; it has to be requested.

TRANSFERRING ETHERS USING METAMASK

Step 1: Sign Into Your Coinbase Account

To initiate the transfer of ETH, you will need to start by signing into your Coinbase account to access or buy some ETH.

Step 2: Buy Ethereum (ETH)

Select buy, enter the amount of money you want to convert to ETH, double-check that you are buying Ethereum, then choose your payment method and finally click on “Buy Now”

Step 3: Click Send

Once you have some ETH in your Coinbase account, navigate to the top of your account and click the Send/Receive button. Then, choose the amount of ETH you want to send to your MetaMask wallet,

Step 4: Copy/Paste Your MetaMask Address to Coinbase

Before sending your ETH, you need to sign into your MetaMask wallet to copy your wallet address. Next, paste your Metamask wallet address into the "To" section in your Coinbase account.

Step 5: Click Continue/Send Now

After pasting your MetaMask wallet into Coinbase, click Continue to review your transfer. Once you have confirmed the transfer looks good, tap "Send Now"

Step 6: Check Your MetaMask Wallet

To confirm that you have received your ETH, log into your MetaMask wallet where you should see a front preview of your new balance.

ETHEREUM FRAMEWORKS

1. Truffle

Truffle is another framework for building decentralized applications on the Ethereum blockchain. It is the most popular and one of the pioneer frameworks. It offers the aid of compilation, deployment, and testing.

Truffle offers a development network to run and test DApps without needing to deploy to the mainnet and serves as a local development blockchain for testing.

2. Embark

Embark is another framework for the EVM blockchain that is regarded as a full-stack framework. This means the framework offers the solution of building an entire decentralized application’s frontend and backend simultaneously.

Embark watches for changes in your Solidity smart contracts and frontend (HTML and JavaScript) code, and redeploys them to the blockchain network.

3. Hardhat

Hardhat is a framework for building smart contracts that offers a development environment for professionals. This development environment lets users compile, run tests, check smart contracts for mistakes or debugging, and deploy decentralized applications.

4. OpenZeppelin

OpenZepplin is a toolkit with plugins that help build smart contracts faster. To use any of the smart contracts in OpenZeppelin, you must import them into your own smart contract. OpenZeppelin is actually distributed as an npm package, meaning you must install Node.js first.

ETHERSCAN.IO

Etherscan permits an individual to view the assets held on any public Ethereum wallet address. On Etherscan by entering any Ethereum address into the search box, one can see the current balance and transaction history of the wallet under consideration.

Etherscan displays any gas fees and smart contracts involving that address.

Etherscan can be used to:

1. Calculate Ethereum gas fees with the Etherscan gas tracker.
2. Lookup and verify smart contracts.
3. View crypto assets held in or associated with a public wallet address.

4. Observe live transactions taking place on the Ethereum blockchain.

5. Lookup a single transaction made from any Ethereum wallet.

6. Discover which smart contracts have a verified source code and security audit.

7. Keep track of how many smart contracts a user has authorized with their wallet. 8. Review and revoke access to a wallet for any decentralized applications (DApps).