## Module 4

**Q. Explain Gas and Ethers in detail.**

**=>**

### Gas

1. Gas refers to the unit that measures the amount of computational effort required to execute operations like transactions, smart contracts, or decentralized applications.
2. It acts like fuel , just as cars need fuel to run, Ethereum operations need gas to execute.
3. Each operation on Ethereum, such as transferring tokens or interacting with a smart contract, has a specific gas cost.
4. The gas cost depends on the complexity of the operation , simple tasks need less gas, and complex tasks need more.
5. For example, a simple Ether transfer may use around 21,000 gas, while deploying a smart contract may consume millions of gas.
6. Gas prevents network congestion by requiring users to pay only for what they use.
7. It also motivates miners (or validators) to include transactions in blocks based on the gas fees offered.
8. Gas fees act as rewards for miners who process and validate transactions.
9. Without gas, users could run infinite loops or heavy computations without any cost, which could crash the network.
10. Hence, gas acts as a control mechanism for resource management and network security.
11. The total gas fee is calculated using the formula:
    Gas Fee = Gas Limit × Gas Price.
12. The Gas Limit represents the maximum units of gas a user is willing to spend for a transaction.
13. The Gas Price is the amount of Ether the user is ready to pay per unit of gas, usually measured in Gwei.
    1 Ether = 1,000,000,000 Gwei ($10^9$ Gwei).
14. For example, if a transaction consumes 21,000 gas and the gas price is 50 Gwei, the total fee = 21,000 × 50 Gwei = 1,050,000 Gwei = 0.00105 ETH.

### Ether (ETH)

1. Ether (ETH) is the cryptocurrency used in Ethereum for paying gas fees and conducting transactions.
2. It serves both as a digital currency and as fuel to power applications on the blockchain.
3. Ether was first introduced in 2015 by Vitalik Buterin and his team when they launched the Ethereum network.
4. It is similar to Bitcoin in terms of being decentralized, but it has additional utility in running smart contracts.

5. Ether is stored in digital wallets and can be traded, sent, or received across the network.
6. It is used to reward miners (in Proof of Work) or validators (in Proof of Stake) for their computational work.
7. Ether ensures that the network remains secure and that every participant has a stake in maintaining it.
8. When users execute transactions or deploy smart contracts, Ether is automatically converted into gas.
9. Thus, Ether acts as the medium of exchange and store of value within the Ethereum ecosystem.
10. The smallest unit of Ether is called a Wei, where 1 ETH = $10^{18}$ Wei.

**Example Summary**

1. Suppose Alice wants to send 1 ETH to Bob using Ethereum.
2. She needs to pay around 21,000 gas for this simple transaction.
3. If the gas price is 30 Gwei, then the fee is 21,000 × 30 = 630,000 Gwei = 0.00063 ETH.
4. Therefore, Alice's total deduction will be 1.00063 ETH from her wallet.
5. This example shows how Ether and Gas work together to execute and pay for blockchain transactions.

**Q. Types of nodes used in Ethereum**

**=>**

1. The Ethereum blockchain is made up of multiple computers that are connected in a decentralized network.
2. Each computer that participates in maintaining the Ethereum network is called a node.
3. A node stores blockchain data, verifies transactions, and communicates with other nodes.
4. Nodes are essential for maintaining security, transparency, and decentralization in Ethereum.
5. Depending on how much data they store and what tasks they perform, there are different types of Ethereum nodes.

**Full Node**

1. A Full Node stores a complete copy of the Ethereum blockchain, including all blocks and transactions.
2. It independently verifies every transaction and smart contract execution.
3. Full nodes download the entire blockchain history from the genesis block to the latest block.
4. They ensure that all transactions follow the Ethereum rules and are valid.
5. Full nodes help maintain the network's security by checking the authenticity of new blocks.
6. Full nodes also help light nodes by sharing verified data with them.
7. Running a full node requires a powerful computer and a large amount of storage (hundreds of gigabytes).

8. Example: Geth (Go Ethereum) and OpenEthereum are popular clients that can run as full nodes.

**Light Node**

1. A Light Node is a lightweight version of a full node that does not store the entire blockchain.
2. It only downloads block headers instead of complete blocks.
3. The block header contains summaries like block hash, timestamp, and Merkle root but not full transaction details.
4. Light nodes rely on full nodes to fetch complete transaction information when needed.
5. They are suitable for users who want to access the Ethereum network without using heavy storage or high processing power.
6. Example: Mobile wallets like MetaMask or Trust Wallet use light nodes to connect quickly to the Ethereum blockchain.

Archive Node

1. An Archive Node stores everything that a full node stores plus all historical states of the Ethereum blockchain.
2. It keeps a complete record of every past account balance, smart contract state, and storage data at every block height.
3. Full nodes only store recent states, but archive nodes save all historical versions for research and analytics.
4. These nodes require very large disk space ,often multiple terabytes of storage.
5. Archive nodes are mainly used by blockchain explorers, analytics platforms, and developers who need old data.
6. Example: Etherscan uses archive node data to display complete transaction and contract history.

**Mining Node (Proof of Work Era)**

1. Before Ethereum shifted to Proof of Stake, Mining Nodes were used to create new blocks.
2. Mining nodes performed complex mathematical calculations to solve cryptographic puzzles.
3. The first miner to solve the puzzle successfully added a new block to the blockchain.
4. In return, the miner received Ether (ETH) as a reward for their computational effort.
5. Mining nodes also verified transactions and maintained the network's consensus.
6. Mining required powerful hardware such as GPUs or ASICs and large amounts of electricity.
7. Example: Ethereum miners used mining software like Ethminer or PhoenixMiner.

**Validator Node (Proof of Stake Era)**

1. In Ethereum's Proof of Stake system, Validator Nodes have replaced mining nodes.
2. Validators are responsible for proposing and confirming new blocks.
3. To become a validator, a user must stake 32 Ether (ETH) as collateral.

4. Validator nodes are randomly chosen to propose blocks and verify transactions.
5. Honest validators earn rewards, while dishonest ones lose part of their staked Ether.
6. This system saves energy and improves network scalability compared to Proof of Work.
7. Validator nodes maintain consensus by voting on block validity in each epoch.
8. Validators use software clients like Prysm, Lighthouse, or Teku to operate on the Ethereum network.
9. Example: Large staking platforms such as Lido or Coinbase run thousands of validator nodes.

Q. Explain Ethereum terms: Miner, Mining Node, Gas, Accounts, Ether, Transactions.

=>

Miner

1. A Miner is a participant in the Ethereum network who validates transactions and adds new blocks to the blockchain.
2. Miners were responsible for maintaining consensus under Ethereum's previous Proof of Work (PoW) system.
3. They used computational power to solve mathematical puzzles known as hash problems.
4. The first miner to solve the puzzle successfully proposed a new block to the network.
5. Other miners then verified the block before adding it permanently to the blockchain.
6. Miners received rewards in Ether (ETH) for their efforts, which included the block reward and transaction fees.
7. Mining ensured that only valid transactions were confirmed and stored on the blockchain.
8. For example, if Alice sent 2 ETH to Bob, miners verified the digital signature and balance before including it in a block.

Mining Node

1. A Mining Node is a computer in the Ethereum network configured to perform mining operations .
2. It maintains a full copy of the blockchain and continuously tries to create new blocks by solving cryptographic puzzles.
3. Every mining node verifies incoming transactions before adding them to a new block.
4. Once a block is successfully mined, the mining node broadcasts it to other nodes in the network.
5. Mining nodes ensure that the blockchain remains decentralized and resistant to attacks.
6. They also store important data like pending transactions (mempool), block headers, and state changes.
7. A mining node requires high computational power, stable internet, and large electricity supply.
8. Example: Geth (Go Ethereum) software could be run in "mining mode" to create a mining node.

Gas

1. Gas is a unit that measures the amount of computational effort required to perform operations on the Ethereum network.
2. Every transaction or smart contract execution consumes gas based on its complexity.
3. Users pay for gas using Ether (ETH), which compensates validators or miners for their work.
4. Gas ensures that no one can overload the network by performing unlimited computations.
5. Each operation in the Ethereum Virtual Machine (EVM) has a fixed gas cost — for example, a simple transaction costs 21,000 gas.
6. The Gas Limit defines how much gas a user is willing to spend for a transaction.

7. The Gas Price determines how much Ether is paid per unit of gas, usually measured in Gwei.
8. The total fee = Gas Used × Gas Price.
9. Example: If a transaction uses 21,000 gas at 30 Gwei per gas, the total cost = 0.00063 ETH.

Accounts

1. An Account in Ethereum represents an entity capable of holding Ether and interacting with smart contracts.
2. There are two main types of accounts: Externally Owned Accounts (EOAs) and Contract Accounts.
3. An Externally Owned Account is controlled by a user's private key and can initiate transactions.
4. A Contract Account is controlled by code (a smart contract) and executes functions when triggered by transactions.
5. Every account has an address, balance, nonce, and optional code or storage.
6. The address is a 42-character hexadecimal string starting with "0x".
7. Example: 0x123abc... is an Ethereum account address.

Ether (ETH)

1. Ether (ETH) is the native cryptocurrency of the Ethereum blockchain.
2. It is used to pay for gas fees, transaction costs, and smart contract execution.
3. Ether also acts as a reward for validators who maintain the network.
4. It is both a medium of exchange and a store of value within the Ethereum ecosystem.
5. Users must hold Ether in their accounts to perform any operation on Ethereum.
6. The smallest unit of Ether is called a Wei, where 1 ETH = $10^{18}$ Wei.
7. Ether is required to deploy contracts, send transactions, and interact with decentralized applications (dApps).
8. Example: When Alice deploys a token contract, she pays gas fees using Ether to complete the transaction.

Transactions

1. A Transaction is a signed message that changes the state of the Ethereum blockchain.
2. It can transfer Ether between accounts or call a smart contract function.
3. Every transaction includes details like sender address, receiver address, amount, gas limit, gas price, and data.
4. Transactions are created by EOAs and broadcast to the network for validation.
5. Once verified, transactions are grouped into blocks and added to the blockchain.
6. Each transaction consumes gas depending on its computational complexity.
7. Example: Sending 1 ETH from Alice to Bob or calling a function in a DeFi contract are both transactions.

## Q. List and Explain Ethereum Test Networks

1. Ethereum test networks are special blockchain environments used for **testing smart contracts and decentralized applications (DApps)** before deploying them on the **main Ethereum network**.
2. These testnets allow developers to **experiment with code safely** because they use **fake Ether (test Ether)** that has no real value.
3. Test networks help in **identifying bugs, optimizing gas usage, and verifying transactions** without risking real assets.
4. The Ethereum ecosystem provides **multiple test networks**, each having its own purpose and configuration.

***1. Ropsten Test Network***

1. The **Ropsten testnet** was one of the earliest and most popular Ethereum test networks.
2. It was designed to **closely mimic the Ethereum mainnet**, meaning it used **Proof of Work (PoW)** like the main network originally did.
3. Developers preferred Ropsten because it behaved almost exactly like the real network, making it ideal for **pre-deployment testing**.
4. However, Ropsten faced frequent **spam attacks and network instability** due to its PoW nature.
5. Example: A developer testing a new ERC-20 token contract could deploy it on Ropsten to verify that transfers and balances work correctly before deploying on the mainnet.

***Rinkeby Test Network***

1. The **Rinkeby testnet** used the **Proof of Authority (PoA)** consensus mechanism, which made it **more secure and stable** than Ropsten.
2. Only a few trusted nodes, called **authorities**, were allowed to validate blocks in this network.
3. It was mainly used for **testing DApps, wallets, and smart contracts** without worrying about spam or malicious attacks.
4. Test Ether for Rinkeby could be obtained from **Rinkeby faucets** by entering an Ethereum address.
5. Example: Developers used Rinkeby to test NFT minting smart contracts because it provided **fast block confirmation** and **predictable behavior**.
6. Rinkeby was also **deprecated in 2023** after being replaced by newer PoS-based testnets.

***Goerli Test Network***

1. The **Goerli testnet** is one of the **most widely used and currently active Ethereum test networks**.

2. It was the **first cross-client testnet**, meaning it supports multiple Ethereum clients such as **Geth, Nethermind, and Besu**.
3. Goerli uses the **Proof of Stake (PoS)** consensus mechanism, just like the Ethereum mainnet after "The Merge."
4. Developers use Goerli to **test staking, transactions, and validator setups** in a PoS environment.
5. Example: Before launching a decentralized finance (DeFi) app, a developer might test liquidity pool functions on Goerli using test Ether.
6. Goerli also helps node operators practice **validator configurations** and **slashing protection**.

***Sepolia Test Network***

1. The **Sepolia testnet** is another **active and recommended test network** for developers today.
2. It is a **lightweight and efficient** network designed to be **faster and easier to sync** compared to Goerli.
3. Sepolia also uses **Proof of Stake (PoS)** consensus, ensuring it behaves similarly to the Ethereum mainnet.
4. It provides a **stable and long-term testing environment** maintained by the Ethereum Foundation.
5. Test Ether on Sepolia can be obtained from **Sepolia faucets** through GitHub or Alchemy.
6. Example: A developer testing smart contract deployment on Remix IDE can use Sepolia as the connected network through MetaMask.
7. Sepolia is now considered the **primary testnet** for most Ethereum developers.

***Kovan Test Network (Deprecated)***

1. The **Kovan testnet** used to be another popular **Proof of Authority** network supported by Parity.
2. It was mainly used for testing Ethereum-based projects using the **Parity client**.
3. However, Kovan was **retired** due to limited client support and the shift toward PoS-based networks.
4. Example: Earlier, developers building DApps using the Parity node client preferred Kovan for reliable test transactions.

***Local Test Network (Ganache / Hardhat)***

1. Apart from public testnets, developers often create **local test networks** for faster testing.

2. Tools like **Ganache** and **Hardhat Network** simulate Ethereum environments on a developer's computer.
3. These local testnets allow instant block mining and transaction verification without internet connectivity.
4. Example: A developer can use Ganache to test hundreds of smart contract transactions instantly before pushing the code to Goerli or Sepolia.