Components of the Ethereum ecosystem

Ethereum

A **blockchain-based platform** for building secure, decentralized applications (**dApps**).

Key Features:

•Smart Contracts: Self-executing digital agreements.
•Decentralized: No central authority—powered by a global network.
•Secure: Data on the blockchain is tamper-proof.

Uses of Ethereum:

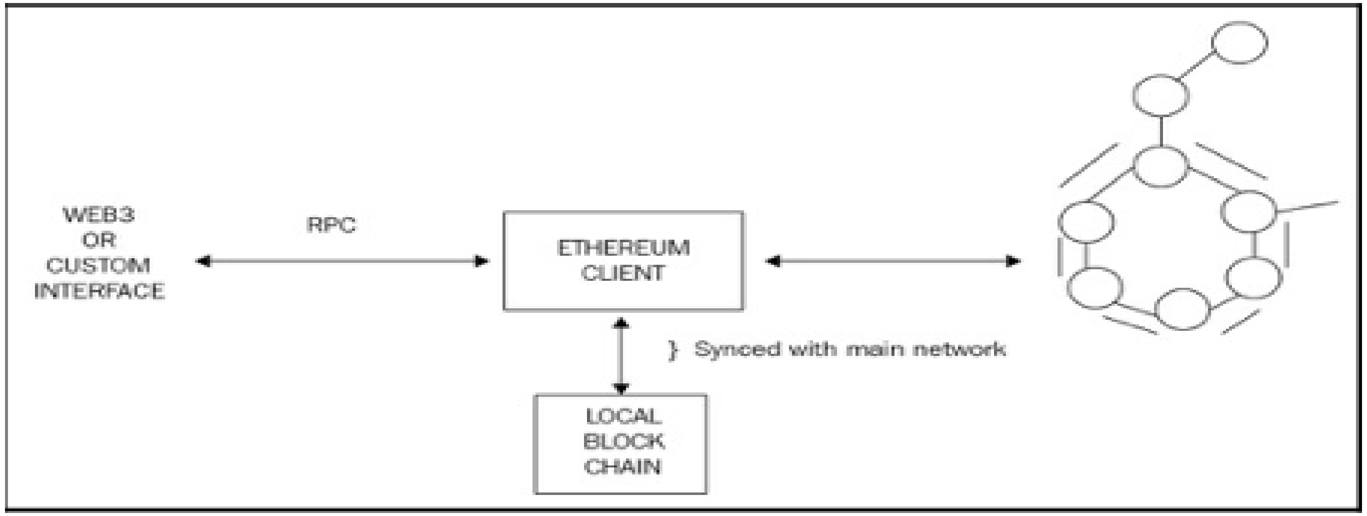
•Finance (DeFi): Secure, fast transactions without banks.
•NFTs: Digital ownership of art, collectibles, and more.
•Transparent product tracking.

Currency:

Ether (ETH) is used to power transactions and apps.

Components of Ethereum

- The **Ethereum Blockchain** records all transactions and smart contracts.
- The **Ethereum Network** connects nodes to share and update blockchain data.
- The **Ethereum Client (Geth)** helps nodes sync, mine, and manage accounts.
- Each node has a **Local Copy** of the blockchain, regularly updated.
- **web3.js** enables apps to interact with the blockchain via RPC



Elements of the Ethereum Blockchain

- Keys and addresses
- Accounts
- Transactions and messages
- Ether cryptocurrency/tokens

Keys and addresses

•Keys and addresses represent ownership and transfer of Ether on the Ethereum blockchain.

Keys:

•Private Key: A secret, randomly generated 256-bit number that proves ownership of Ether.
•Public Key: Derived from the private key using ECDSA (Elliptic Curve Digital Signature Algorithm).

Address Generation Process:

1. Private Key Generation: A random 256-bit number is generated based on the secp256k1 elliptic curve.
2. Public Key Derivation: The public key is created from the private key using ECDSA.
3. Address Derivation: The address is the last 20 bytes of the Keccak-256 hash of the public key.

Keys and addresses

An example of how keys and addresses look like in Ethereum:

1.Private Key:
b51928c22782e97cca95c490eb958b06fab7a70b9512c38c36974f47b954ffc4

2.Public Key:
3aa5b8eefd12bdc2d26f1ae348e5f383480877bda6f9e1a47f6a4afb35cf998ab847f1e3948b1173622dafc6b4ac198c97b18fe1d79f90c9093ab2ff9ad99260

3.Address:
0x77b4b5699827c5c49f73bd16fd5ce3d828c36f32

Transactions and Messages

In Ethereum, transactions are how data and Ether are transferred between accounts or to create new smart contracts. Here's a breakdown of how transactions work:

Types of Transactions:

1. **Message Call Transactions**: These transactions send a message from one contract account to another.
2. **Contract Creation Transactions**: These create new smart contracts on the Ethereum blockchain.

Transactions and messages

Key Components of a Transaction:

1. **Nonce**: A unique number assigned to each transaction from the sender. It ensures the transaction is only processed once.
2. **Gas Price**: The amount of Ether you’re willing to pay for the transaction to be processed.
3. **Gas Limit**: The maximum amount of gas that can be used for a transaction.
4. **To**: The address of the recipient (or the new contract, in case of contract creation).
5. **Value**: The amount of Ether to be transferred.
6. **Signature**: A digital signature that proves the sender's identity and protects the transaction from tampering.
7. **Init (for Contract Creation)**: Contains the code to initialize a contract when it is created. This code is executed once when the contract is created.
8. **Data (for Message Calls)**: The data sent along with the message in a contract call, typically used to trigger specific functions within the contract.

Transactions and messages

1. Contract Creation Transaction

This type of transaction is used to **create a new contract** on the Ethereum blockchain. It involves the following steps:

- **Sender:** The person or account initiating the transaction.
- **Gas:** A certain amount of gas is required to execute the transaction.
- **Endowment:** The amount of ether (ETH) that the sender allocates to the contract (this can be zero).
- **Initialization Code:** This is the code that will be executed to set up the contract once it's created.

When you create a contract, Ethereum generates an **address** for the new contract. This address is derived from the sender's address and the transaction nonce (a number that ensures the uniqueness of transactions). Specifically, the new address is the last 160 bits of the Keccak hash of the sender’s address and nonce.

If there is any issue during contract creation, such as running out of gas, **the contract won’t be created**, and no changes are made to the blockchain state. If everything goes well, the contract is created, and the sender's balance is reduced by the amount of ether allocated to the contract.

Transactions and messages

2. Message Call Transaction

A **message call** is a transaction where one account calls a function or executes code in another account (typically, this would be another contract). The main parameters for a message call are:

- **Sender:** The account sending the message.
- **Recipient:** The account that will receive and process the message.
- **Gas & Gas Price:** Gas is needed to perform the transaction, and the sender pays the gas fee.
- **Value:** The amount of ether being sent with the transaction (can be zero).
- **Input Data:** This is any data required for the function call.
- **Depth:** This refers to how deep the call is within the execution stack (i.e., how many function calls have been made).

Transactions and messages

How it Works:

- When the message call is made, it **executes the function** or code in the recipient contract.
- If the function **changes the blockchain state** (e.g., updates balances or transfers ether), those changes are recorded permanently.
- If the function **returns data**, the result is sent back to the sender or calling program.

Transactions and messages

1. Transaction Process:

- When someone wants to make a transaction (like sending Ether or interacting with a smart contract), the transaction is first sent to a **transaction pool**. This is like a waiting area where transactions sit before they are processed.
- In the transaction pool, transactions are **not yet confirmed**; they are just waiting to be validated by the network.

2. Mining Process:

- **Miners** are responsible for picking the transactions from the pool. They typically pick the ones that offer the highest fees (because miners are rewarded for their work).
- Miners then **process the transactions**, which means they check the transaction details and add them to a new **block**.
- After adding the transactions, miners need to **solve a cryptographic puzzle** (kind of like a math problem) to confirm that the block is valid.
- Once the miner solves the puzzle and the block is verified, the block is **broadcasted** (sent out) to the rest of the network, where other nodes check if everything is correct.

Messages

Messages are like **transactions** but differ in origin. Transactions are created by external entities (Externally Owned Accounts), while **messages** are produced by **contracts**.

- A **message** is essentially a data packet containing:
  - **Sender's address**: Where the message is coming from (contract or account).
  - **Recipient's address**: The target of the message (another contract or account).
  - **Amount of Wei**: The value being transferred with the message.
  - **Data field** (optional): This contains the actual data or input parameters required by the recipient contract.
  - **Gas**: The maximum amount of computational resources (gas) allocated to process the message.

Accounts

In Ethereum, accounts are where transactions happen, and the state of the blockchain changes whenever a transaction occurs between them. Here’s how a transaction works step by step:

1. **Check the Transaction:**
   The system checks if the transaction is valid (e.g., if the signature is correct and if it's not a replay).

2. **Calculate the Fee:**
   The system calculates how much the transaction will cost (called the gas fee). It checks if the sender has enough money to pay for the fee.

3. **Gas for the Transaction:**
   The sender needs to provide enough Ether (gas) to pay for the transaction. Gas is based on how big the transaction is.

Accounts

4. Transfer the Ether:
The Ether is moved from the sender’s account to the receiver’s account. If the receiver doesn’t exist, a new account is created automatically.

5. Smart Contract Execution:
If the receiver is a smart contract, the contract’s code is run. If there’s not enough gas, the contract will stop before it’s finished.

6. If Something Goes Wrong:
If there’s an issue (like not enough money or gas), the transaction is canceled, but the transaction fee is still paid to the miners.

7. Return the Remaining Fee:
If there’s any leftover fee, it’s returned to the sender as change.

Types of Accounts

Externally Owned Accounts (EOAs):

1. EOAs hold Ether and can send transactions.
2. They are controlled by private keys and are linked to a user.
3. EOAs don’t have any code associated with them.

Contract Accounts (CAs):

1. CAs also hold Ether but have code stored on the blockchain.
2. They can execute code when triggered by a transaction or other contracts.
3. CAs can maintain their own state and interact with other contracts.

Ether cryptocurrency

Ether (ETH) is the native cryptocurrency of the Ethereum blockchain. It is used to pay for transaction fees and smart contract execution within the Ethereum network.

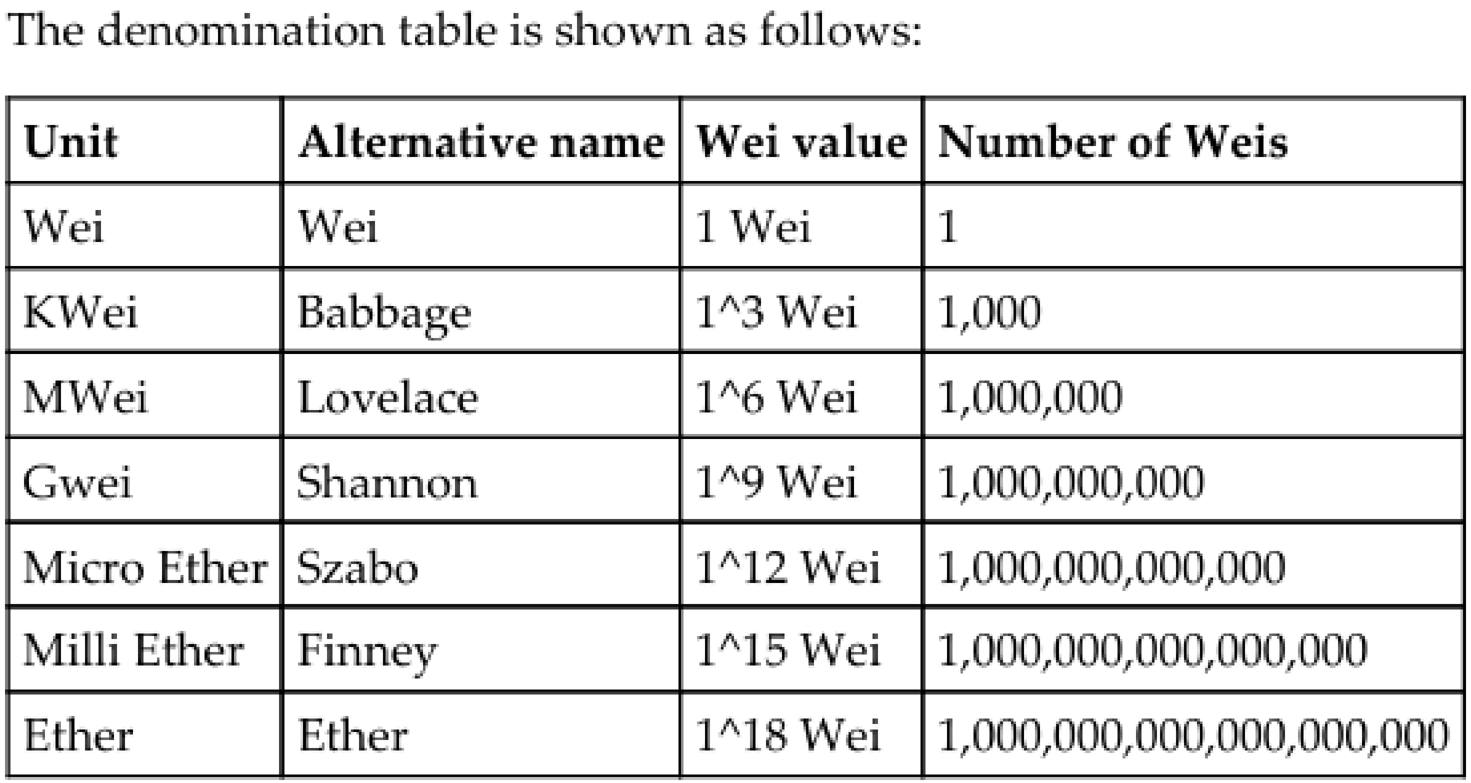
Mining & Rewards:

- Miners validate transactions and secure the Ethereum network. In return for their computational work, they are rewarded with Ether (ETH).

- **Ethereum Forks:**
  - **ETH**: The current version of Ethereum after a hard fork following the DAO hack. This version is active and undergoing continuous development.
  - **ETC**: Ethereum Classic is the original version of Ethereum that was not altered post-DAO hack and has its own community and development.

- **Usage of Ether:**
  - **Gas**: Ether is used to pay for "gas," which is required to perform computations and execute smart contracts on the Ethereum blockchain. Gas acts as the network’s fuel.

Ether Cryptocurrency



The denomination table is shown as follows:

Unit	Alternative name	Wei value	Number of Weis
Wei	Wei	1 Wei	1
KWei	Babbage	1^3 Wei	1,000
MWei	Lovelace	1^6 Wei	1,000,000
Gwei	Shannon	1^9 Wei	1,000,000,000
Micro Ether	Szabo	1^12 Wei	1,000,000,000,000
Milli Ether	Finney	1^15 Wei	1,000,000,000,000,000
Ether	Ether	1^18 Wei	1,000,000,000,000,000,000

THANK YOU…