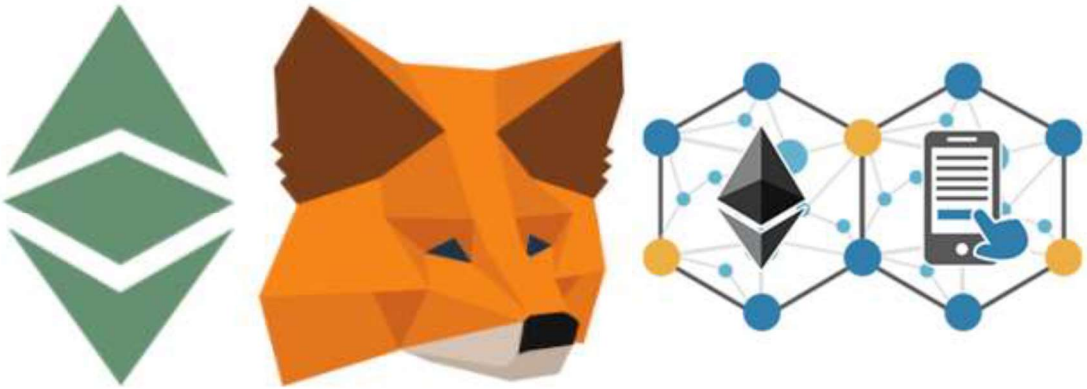
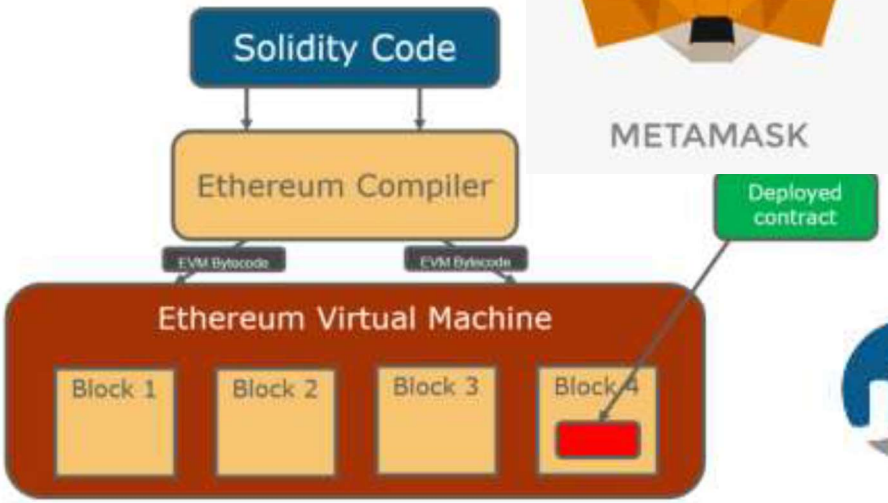
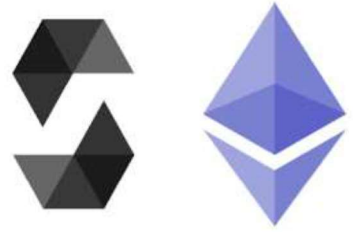
रा.इ.सू.प्रौ.सं
NIELIT



National Institute of Electronics and Information Technology

Types of Ethereum networks, Installation & Deployment
Module 2-Setup and Configuration of BlockChain





Some Basic Concepts

1. The **main objective of Ethereum** is to accept transactions from accounts, update their state, and maintain this state as current till another transaction updates it again.
2. The whole process of accepting, executing, and writing transactions can be divided into two phases in Ethereum. There is a decoupling between when a transaction is accepted by Ethereum and when the transaction is executed and written to the ledger.
3. **Cryptography** is the science of converting plain simple text into secret, hidden, meaningful text, and vice-versa. There are the following two types of cryptography in computing:
   - **Symmetric cryptography** refers to the process of using a single key for both encryption and decryption.
   - **Asymmetric cryptography** refers to the process of using two keys (public & private) for encryption and decryption. Any key can be used for encryption and decryption. Message encryption with a public key can be decrypted using a private key and vice versa.
4. **Hashing** is the process of **transforming any input data** into fixed length random character data, and it is not possible to regenerate or identify the original data from the resultant string data.
   - Hashes are also known as **fingerprint of input data**. It is next to impossible to derive input data based on its hash value.
   - Hashing ensures that even a slight change in input data will completely change the output data.
   - Ethereum uses **Keccak256** as its **hashing algorithm**.
5. **Digital signatures** are used to sign transaction data by the owner of the asset or cryptocurrency, such as Ether.

Some Basic Concepts

6. **Ether** is the **currency of Ethereum**. Every activity on Ethereum that modifies its state costs Ether as a fee, and miners who are successful in generating and writing a block in a chain are also rewarded Ether.
7. Ethereum has a **metric system of denominations** used as units of Ether. The smallest denomination or base unit of Ether is called **wei**.
8. **Gas** is the **internal currency** of Ethereum. The execution and **resource utilization cost** is predetermined in Ethereum in terms of **gas units**. This is also known as **gas cost**.
9. **Nodes** represent the computers that are connected using a **peer-to-peer protocol** to form an **Ethereum network**. There are **two types of nodes** in Ethereum: **Ethereum virtual machine (EVM)** and **Mining nodes**. In most scenarios, there is no dedicated EVM. Instead, all nodes act as miners as well as EVM nodes.
10. A **Merkle root** is the hash of all the hashes of all the transactions that are part of a block in a blockchain network.
11. Ethereum has a concept of the **genesis block** also known as **first block**. This block is created automatically when the chain is first initiated. You can say that a chain is initiated with the first block known as the **Genesis Block** and the formation of this block is driven through the **genesis.json** file.
12. Each block has an **upper gas limit** and each **transaction** needs a certain amount of gas to be consumed as part of its execution. The cumulative gas from all transactions that are not yet written in a ledger **cannot surpass the block gas limit**.

Some Basic Concepts

13. EVMs are the **execution components** in Ethereum. The purpose of an EVM is to execute the code in a smart contract line by line.
14. However, when a transaction is **submitted**, the transaction is not executed immediately. Instead it is pooled in a **transaction pool**.
15. Each **miner** maintains an **instance of ledger**. A ledger contains all blocks in the chain. The miners **synchronize their blocks** on an on-going basis to ensure that every miner's ledger instance is the same as the other.
16. A **miner's job** is very similar to that of an **accountant**. As an accountant is responsible for writing and maintaining the ledger; similarly, a miner is solely responsible for **writing a transaction** to an **Ethereum ledger**.
17. **Only one miner** can write the block to the ledger. The miner responsible for writing the block is determined by way of a **puzzle**. The miner who **solves the puzzle first** writes the block containing transactions to his own ledger and sends the block and **nonce value** to other miners for verification. Once verified and accepted, the new block is written to all ledgers belonging to miners.
18. Nonce is the **number which can be used only once**. Miners test and discard millions of Nonce per second until they find that **Golden Nonce** which is valid. In order to complete the verification faster than other miners, miners compete with each other using their **computer hashing power**. Once the Golden Nonce is found, they can complete the Block and add it to the Block Chain and there by receive the Block reward.

Proof of Work

1. At some point, the **miner collects all transactions** from the **transaction pool** and constructs a new block & adds all transactions to it. It will check if any of the transactions are not already written in a block that it might receive from other miners. If so, it will **discard those transactions**.
2. The next task for a miner is to **generate the block header** and perform the following tasks:
   i. The miner takes hashes of two transactions at a time to generate a new hash till he gets a single hash from all transactions. The hash is referred to as a root transaction hash or Merkle root transaction hash. This hash is added to the block header.
   ii. The miner also identifies the hash of the previous block. The previous block will become parent to the current block and its hash will also be added to the block header.
   iii. The miner calculates the state and receipts of transaction root hashes and adds them to the block header.
   iv. A nonce and timestamp is also added to the block header.
   v. A block hash consisting of both block header and body is generated.
   vi. The mining process starts where the miner keeps changing the nonce value and tries to find a hash that will satisfy as an answer to the given puzzle. It is to be kept in mind that everything mentioned here is executed by every miner in the network.
   vii. Eventually, one of the miners will be able to solve the puzzle and advertise the same to other miners in the network. The other miners will verify the answer and, if found correct, will further verify every transaction, accept the block, and append the same to their ledger instance.
3. This entire process is also known as **Proof of Work (PoW)** wherein a miner provides proof that it is has worked on computing the final answer that could satisfy as solution to the puzzle.

Ethereum Accounts

1. **Accounts** are the **main building blocks** for the **Ethereum ecosystem**. It is an **interaction between accounts** that Ethereum wants to **store as transactions** in its ledger.
2. There are **two types of accounts** available in Ethereum—**externally owned accounts** and **contract accounts**.
3. Each account, by default, has a **property named balance** that helps in querying the current balance of Ether.
4. **Externally owned accounts** are accounts that are owned by people on Ethereum. When an externally owned account is created on Ethereum by an individual, **a public/private key is generated**.
   - The private key is kept safe with the individual while the public key becomes the identity of this externally owned account.
   - This public key is generally of 256 characters, however, Ethereum uses the **first 160 characters** to represent **the identity of an account**.
   - An externally owned account **can hold Ether in its balance** and **does not have any code associated with it.**
   - It can execute transactions with other externally owned accounts and it can also execute transactions by invoking functions within contracts.
5. **Contract accounts** are very similar to externally owned accounts.
   - They are identified using their public address.
   - They do not have a private key.
   - They can hold Ether similar to externally owned accounts; however, they contain code—code for smart contracts consisting of functions and state variables.

Transactions

1. A transaction is **an agreement** between a buyer and a seller, a supplier and a consumer, or a provider and a consumer that there will be **an exchange of assets, products or services** for **currency, cryptocurrency or some other asset**, either in the present or in the future.
2. **Ethereum helps** in **executing the transaction**.
3. Following are the **three types of transactions** that can be **executed in Ethereum**:

   i. **Transfer of Ether from one account to another:** The accounts can be externally owned accounts or contract accounts. Following are the possible cases:

   - An externally owned account sending Ether to another externally owned account in a transaction
   - An externally owned account sending Ether to a contract account in a transaction
   - A contract account sending Ether to another contract account in a transaction
   - A contract account sending Ether to an externally owned account in a transaction

   ii. **Deployment of a smart contract:** An externally owned account can deploy a contract using a transaction in EVM.

   iii. **Using or invoking a function within a contract:** Executing a function in a contract that changes state is considered a transaction in Ethereum. If executing a function does not change a state, it does not require a transaction.

Important Properties Related to a Transaction

1. The **from** account property **denotes the account** that is **originating the transaction** and represents an account that is ready to send some gas or Ether. The **from account** can be externally owned or a contract account.
2. The **to** account property refers to an account that is receiving Ether or benefits in lieu of an exchange. For **transactions** related to **deployment of contract**, the to field is **empty**. It can be externally owned or a contract account.
3. The **value** account property refers to the amount of Ether that is transferred from one account to another in wei.
4. The **input** account property refers to the **compiled contract bytecode** and is used during **contract deployment** in EVM. It is also used for **storing data** related to **smart contract function calls** along with its parameters.
5. The **blockHash** account property refers to the **hash of block** to which this transaction belongs.
6. The **blockNumber** account property is the block in which this transaction belongs.
7. The **gas** account property refers to the **amount of gas supplied** by the **sender** who is executing this transaction.
8. The **gasPrice** account property refers to the **price per gas** the **sender** was willing to **pay in wei**. **Total gas** is computed at **gas * gasPrice**.
9. The **hash** account property refers to the hash of the transaction.
10. The **nonce** account property refers to the **number of transactions** made by the **sender** prior to the current transaction.

Important Properties Related to a Transaction

11. The **transactionIndex** account property refers to the **serial number** of the **current transactions** in the block.
12. The **v, r and s** account properties relate to **digital signatures** and the signing of the transaction.

A typical transaction in Ethereum where a contract function is invoked


```
{ blockHash: '0xba93a91df520c7565e80347346e47b83a41d473a33352d1cf7e689c30b305ba5',
  blockNumber: 70,
  from: '0xa57de277ede9c1521f51f6989ed2497a5b9c1926',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0x6b65b86462e6aa89d5f9469ce03b9ea21e8bf72f8c11aa72de2978f9b7a5b9fd',
  input: '0xc8aaea4000000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000000000000000000000000000',
  nonce: 1,
  to: '0x6b90c690b23af11c9575c2b9b8e26d47d84b4f8b',
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0x42',
  r: '0xd4f5adb0f1739105668afa5aba700ae4f031e88e9f21bdb7ac787a3af156baf7',
  s: '0x7f89cda2ae325948f73bcd8da93e2cd6cf9917b22e32d93c3f8caedc0edcd1' }
```

Blocks

1. Blocks are an important concept in Ethereum.
2. Blocks are containers for a transaction.
3. A block contains multiple transactions.
4. Each block has a different number of transactions based on gas limit and block size.
5. The blocks are chained together to form a blockchain.
6. Each block has a parent block and it stores the hash of the parent block in its header.
7. Only the first block, known as the genesis block, does not have a parent.