Unit - 5 BT.

Page No.
Date

* What is Ethereum.

* decentralized, Peer to peer, Cryptography, Info Security, ~~dis~~ dLT.
* Block chain Technology.
* By developers for developers.
* currency - Ether (ETH)
* Create and publish PApp.
* It has EVM.
* Turing complete language.
* Can serve as general purpose computer.
* we can create app without creating new blockchain network.

* Ethereum 2.0 - new version. - PoS.

Ethereum has a Memory that Stores both data & code and they track how Memory is changing over time.

It has ability to load the code in ~~machine~~ State Machine a genesis state is transformed into a final state.

* Components of Ethereum

1. P2P network.
2. Consensus regulation. Yellow Paper Contain
3. Transactions.

4. State Machine. EVM.
5. Data structure - Merkle patricia Tree
6. Consensus algorithm.
7. Economic Security
8. Client.
9. Gas
10. Account
11. ether
12. node.

* Gas - Internal currency, we need gas to run an App or Smart contract.
  - So, Every instruction have some fixed price.
  - EVM will terminate the computation if gas exceeds.

* DApps. - Decentralized App.
  - Open, decentralized, peer to peer
    front end - user interface
    Back end - Smart Contract.
  - Whisper - Messaging protocol
  - Swarm - Strage

* Ether - currency, ETH.

* key and addresses. - used for transfer and Ownership of ether.

1) First private key chosen from Ecc
2) then public key is drived from private key
3) And address is drived from public key

* Types of Account.

(1) EOA - Externally owned Account
→ It has private key to make transaction.
→ no associated code
→ human user.

(2) Contract Account
→ code and storage associated
→ Not have private key so, they can't initiate trans^n.
→ logic of the smart contract control this account

* Ethereum Node → peer to peer machine
  ① Mining node
  ② EVM

* Transaction. - agreement blw sender & receiver
  EOA → CA
  EOA → EOA
  CA → CA
  CA → EOA.

It contain.

from	blocknumber	v r s
To	gas	trans^n
value	gasPrice	Index.
Input	hash	
block hash	nonce	

* Types of Ethereum Network

Ethereum network -> environment

(1) public Network.

Anyone can access and add, view, transaction

(a) Ethereum Mainnet

-> primary public network when actual transn occur.

-> It has real value, real ETH and real consequences.

(b) Ethereum Testnets

-> public, used by the developers and test contract code before release in mainnet.

-> uses proof of authority

-> no real value

Adv.

1. require less data to Syn & Store.
2. In a few hours, it can Syn.
3. public

Disadv.

1. gas is free
2. test ether is used, no real value.

Private Network.

- If nodes is not connected to public network.
-> Isolated.

Adv.

1. only you are there, No other users
2. No requirement of test.
3. only your contract is there.
4. limit acess.

Disadv.

1. less decentralized & Secure

* Ethereum Virtual Machine : EVM

-> DApp developers in ethereum write their smart contract in high level languages like solidity,

- Compile them into byte code.

-> then upload them for execution

-> Ethereum smart contract run time environment is called EVM.

-> It has own Storage

-> limited access into of current transn

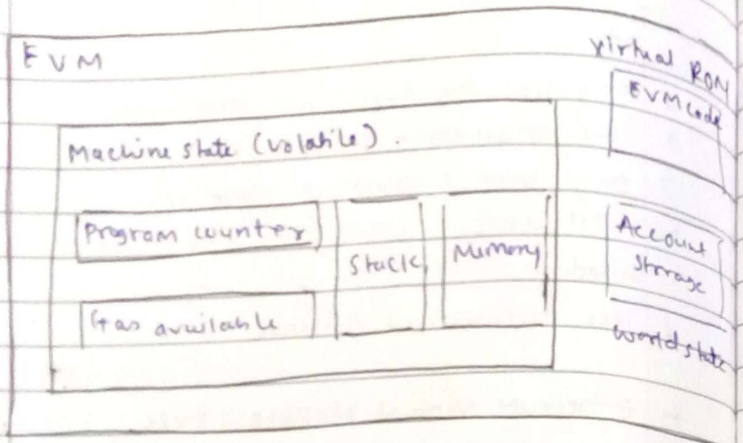
-> It run Smart contract's code line by line.

-> amount of gas required.

-> Ethereum protocol is their for maintain the uninterrupted, continuous, & immutable operation of this unique State Machine.

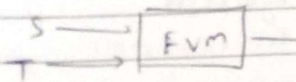
→ code → bytecode → execute

→ It is distributed State Machine and state is data structure that contain all accounts and balances.



Ethereum State transition function.

$Y(S, T) = S'$ S - State, T - transaction.



State uses modified Merkle Patricia tree.

Transaction: instruction from account that have been cryptographically signed.
(1) message call
(2) Contract.

Evm execute around 140 operation
eg XOR, AND, ADD, SUB, ADDRESS, ...

Smart Contract

→ Computer program
→ Immutable.
→ Deterministic
→ Evm Context - can access only limit info of recent block
→ Decentralized world computer -

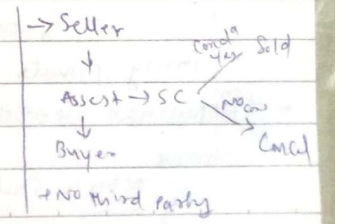
Entire task is completed using Smart Contract

Life cycle of a smart contract.

1. Creation - using high level lang. eg. Solidity and compile into low level lang. eg. bytecode
2. Deployment - on Ethereum network.
3. Execution - evaluation of contract, execution
4. Completion: State is change, successful termination

Purpose of Smart Contract

→ Security,
→ transparency
→ prevent data manipulation
→ track transaction
→ identity kept secret



Types of Smart Contract

1. Smart Legal Contracts → legally enforceable and require parties to satisfy their contractual obligation.

2. Decentralized Autonomous Organization. →
The contract is bound to specific rule that are coded into blockchain Contract Hend with governance mechanism

3. Application Logic Contract
   - application based code
   → allow device to function securely
   → eg. IOT

* Implementing and deploying Smart Contract using Solidity

Code.

```
pragma solidity ^0.5.0;        → min. version of compiler

contract Helloworld → contract object, similar to class in Java
{                  → scope
    string private statv = "Helloworld";   → variable
                  → access modifier
    function GetHC) public view return (string memory)   → function
    {
        return statv;
    }
                  only for read
```

Remix - IDE

Icon panel
Side panel.
Main panel
Terminal Panel.

1. create new workspace + icon.
2. Create new file. and name. ("Hello.sol")
3. Type code
4. click on compiler icon and select compiler version
5. Deploy the smart contract.
6. click on button show Greet as function and it give message 0: string: Helloworld

→ ABI - Application Binary interface
→ Bytecode

Swarm : Decentralized Storage platform

- It is giant DHT (Distributed Hash Table) which act as a database where nodes can discover where the data is stored.
- Many protocol uses this as storage solution
- Decentralized permissionless Communication platform

DISC: Distributed immutable storing chunks
- It is underlying storage Model for Swarm.
→ Strong protection
→ Data integrity

Swarm Storage structure

Chunk - Unit of storage used in Swarm - 4kB

- It allow dApp developer to store & broadcast the data among the network.

- BZZ are rewarded to who contribute in resource.

- ~~Kademlia~~ KadeMlia is DHT. where self organized Computer System can communicate.

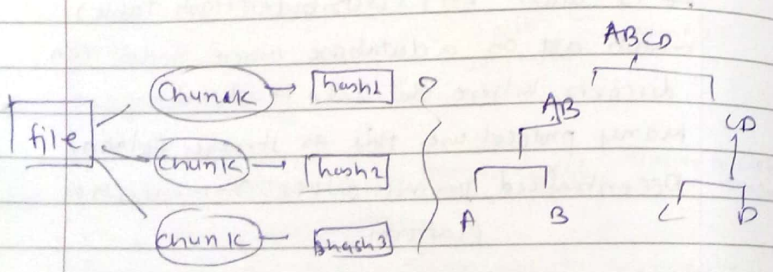
→ Not good for huge database.

① Global pinning : User store data on their drive : free

② Postage Stamp : nodes can issue BZZ token for storing content.

working with Swarm

Merkle Tree



Manifest file : Collection of document, contain Path, content, hash,

→ uses devp2p / RLPx protocol.

Whisper : Decentralized messaging platform.

→ used for communicate

→ Simple API to send & receive msg

→ Government can track our Movement, So, Whisper is secure communication.

1) Delivering Message in Darkness

No evidence of communication

(1) inaccessible to interceptor

(2) nodes are not easily identified.

2) Protocol in whisper

RLPx protocol serve as base layer foundation of this system.

→ Max size of message 64 kb

Scenario

If A want to send message to B, it will first send msg to the Decentralized Network, here whisper nodes will send msg to all other whisper nodes and the forget receipt will able to pick up the msg from any node.

+ whisper encrypt msg using Symmetric key.

→ It seal into an envelope. (doesn't contain info about receipt)

Envelope contain.

1. Version : use for decryption format.
2. Data : content
3. TTL (Time to live) time left for msg.
4. Expiry : Expiration date.
5. Topic field
6. EnvNonce : a no. that help POW method to determine whether the work has already been approved by the system
7. AFSNonce : Unique no. that improve network security.

- SHA 256 is used for encryption.