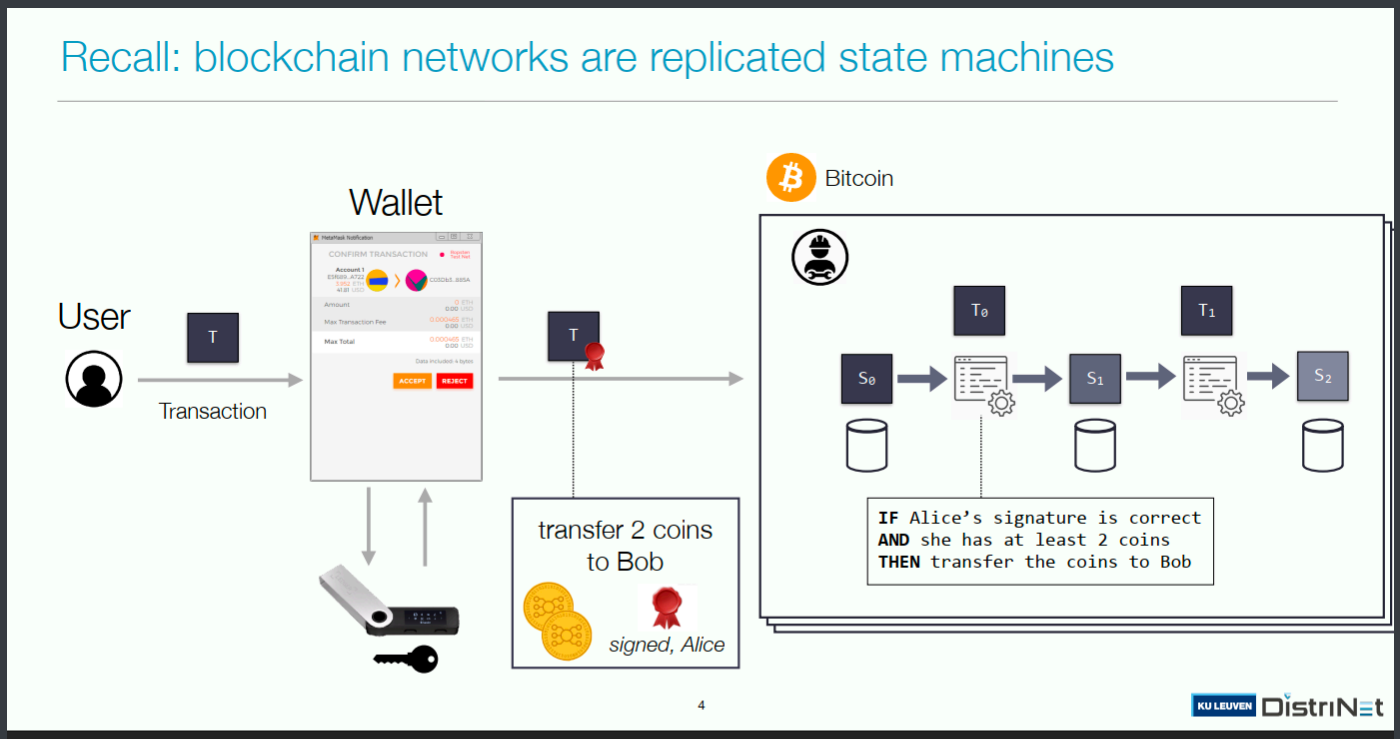
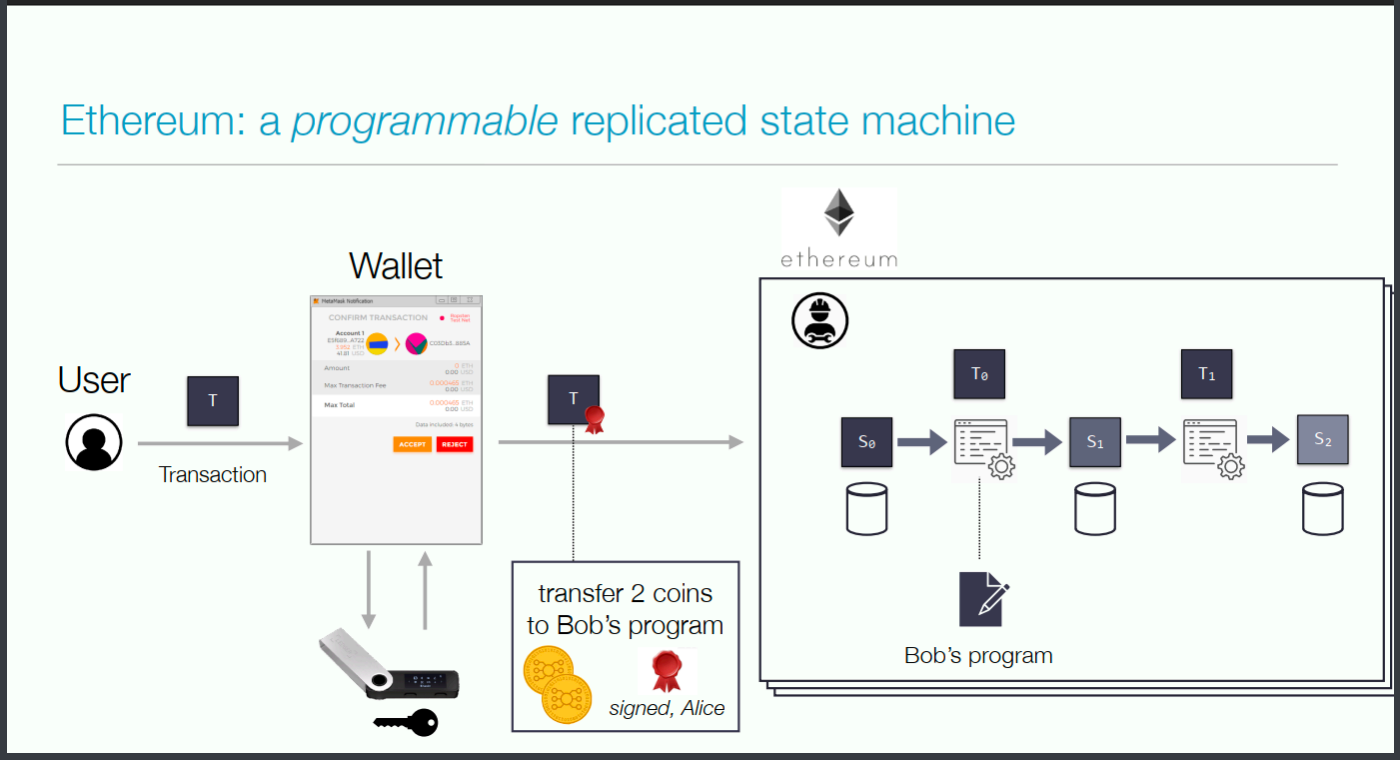
Ethereum

ethereum is a programmable blockchain





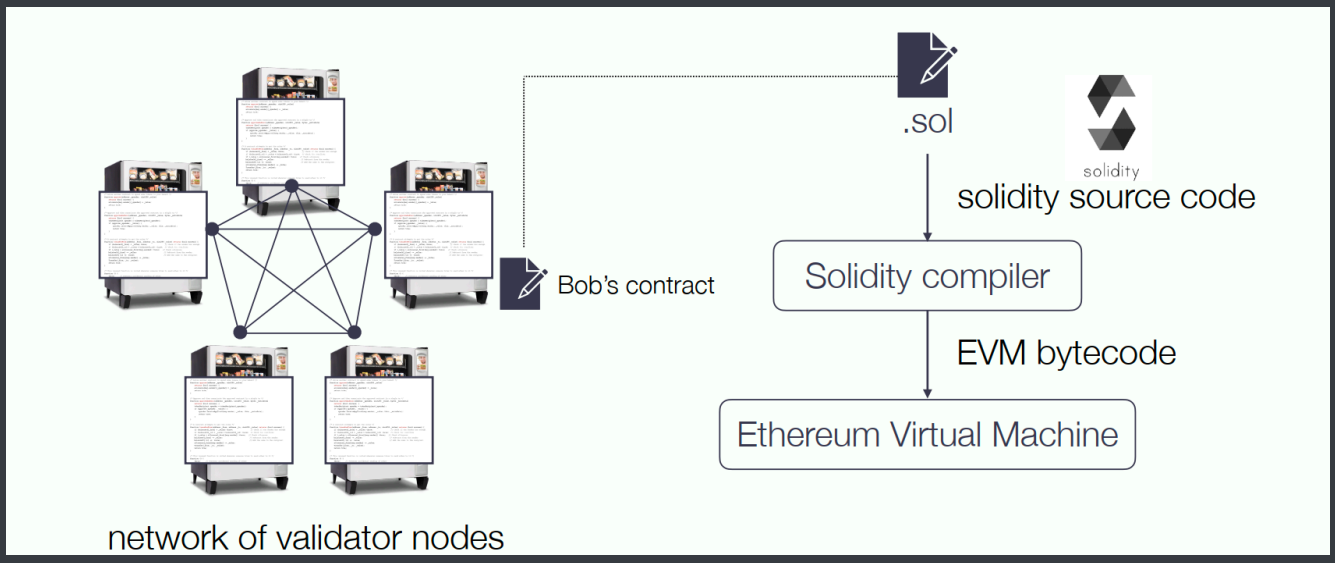
Ethereum 允许用户通过smart contract编程

smart contract是存储在区块链上的自执行代码

smart contracts

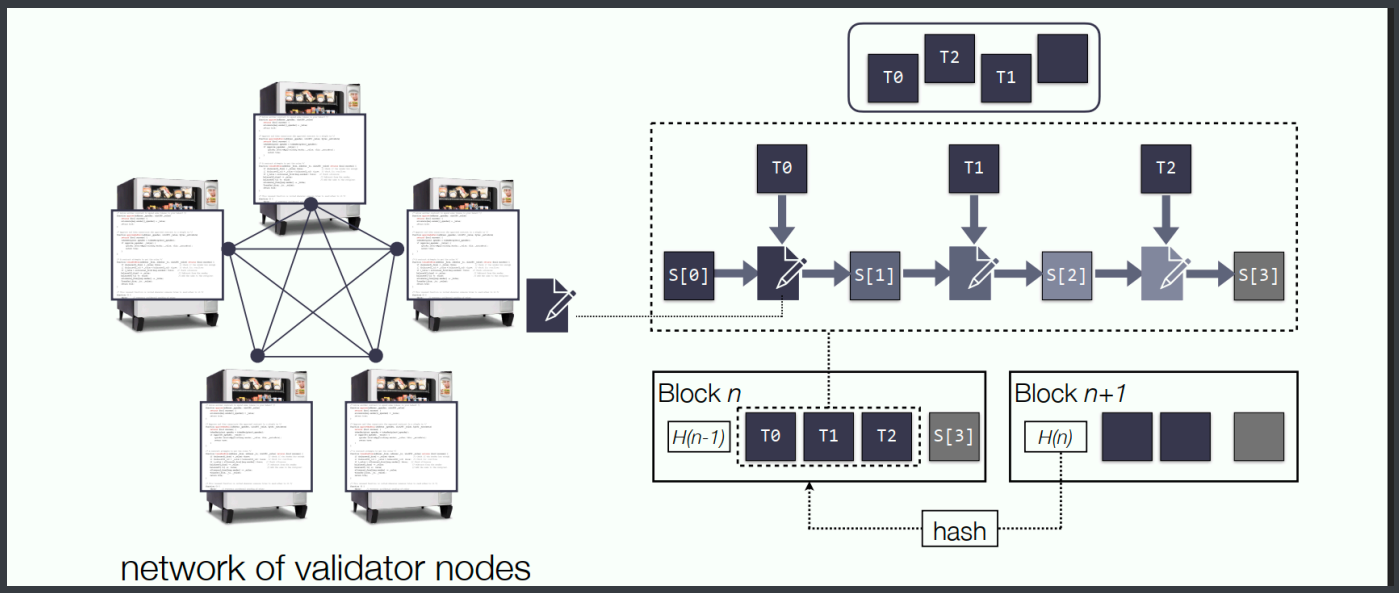
what is a smart contract

- automatically moves digital assets according to arbitrary pre-specified rules
- a program with its own bank account
- goal is to **reduce counterparty risk**
- Parties agree to the contract by transferring control of their (digital) assets to the contract thus cryptographically “locking up” their assets.
- contract 托管 assets (only can be transferred out according to the written logic)
- contracts are written in a high-level language but stored as bytecode



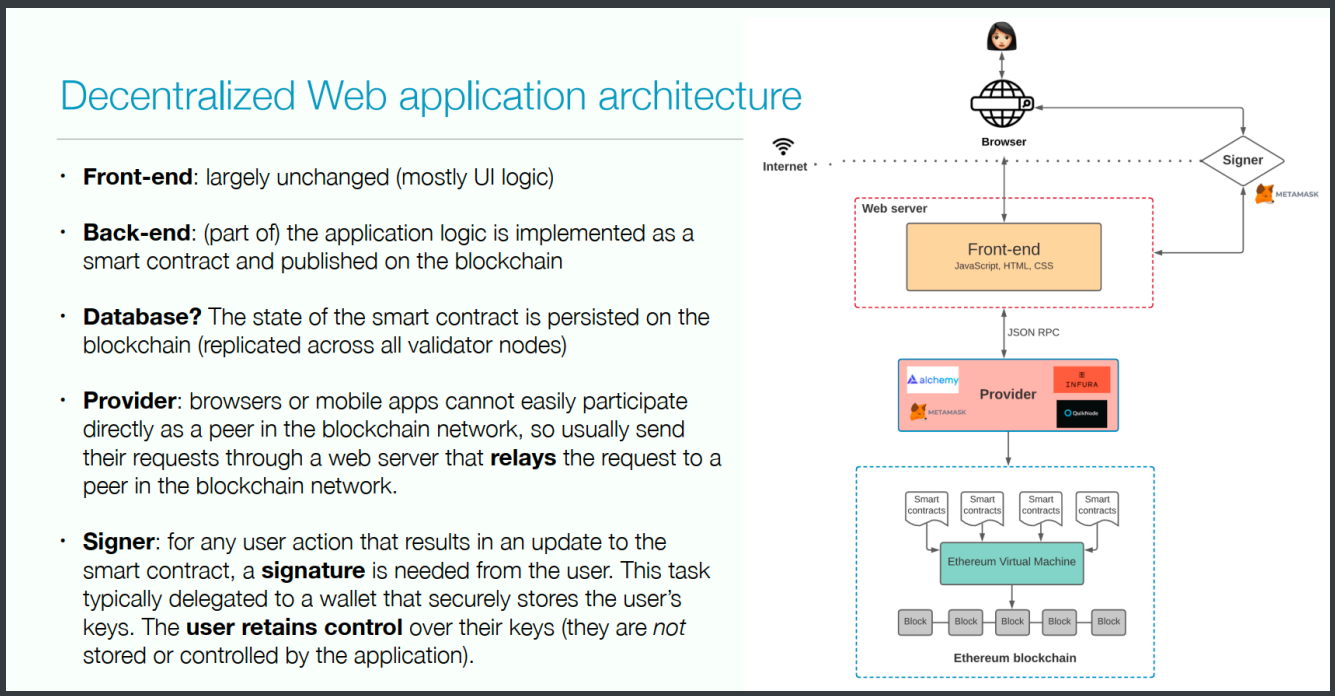
where does security lie in?

As long as the majority of network resources is controlled by honest parties, the single virtual computer is highly available and trustworthy - it is guaranteed to execute the code as described



decentralized applications (Dapps)

- parts of the software architecture are architeccturally and politically decentralized
- decentralized web application architecture



compared to centralized web application architecture:

front-end, back-end, database

Ethereum: addresses & accounts

addresses

- users are **pseudonymous**, identified by their address
- in Ethereum, addresses are 20 bytes, typically formatted as a 40-digit hex string

accounts

- generate a private - public key pair, then address is hased from the public key
- access to public key / address allows one to query the accoutn balance,

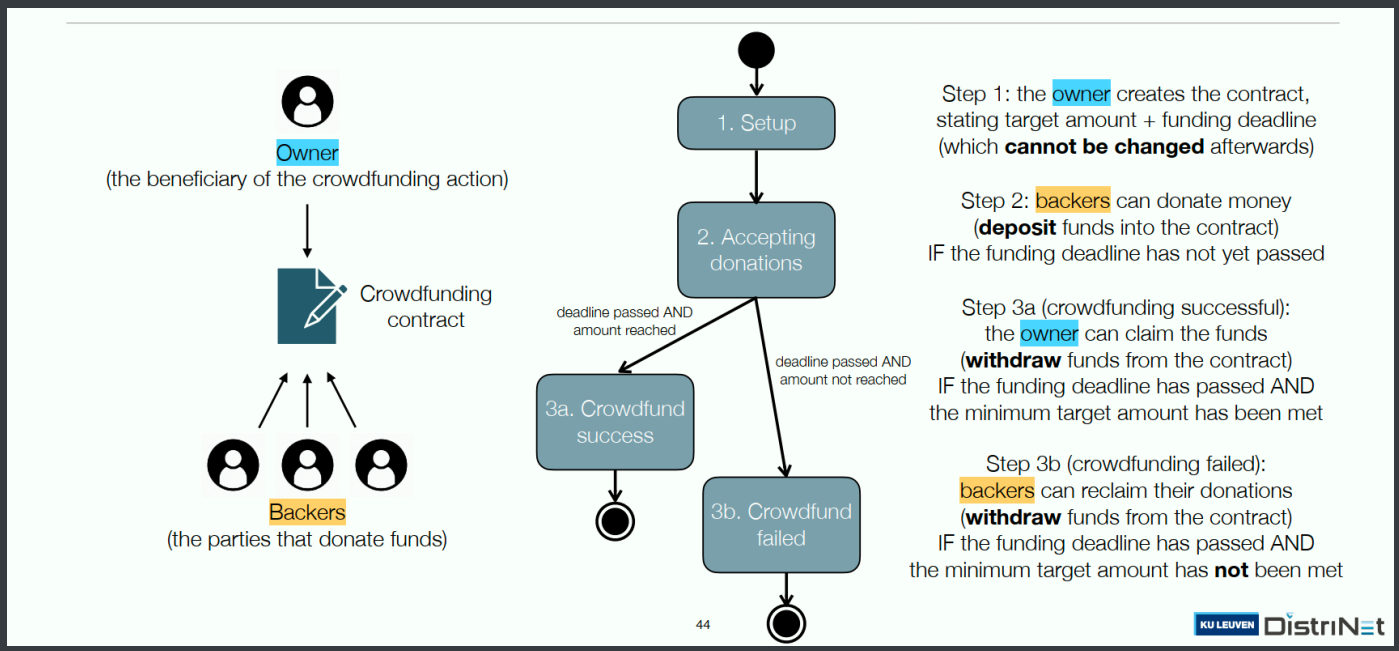
  access to private key allows one to spend the account balance (signing the transactions)

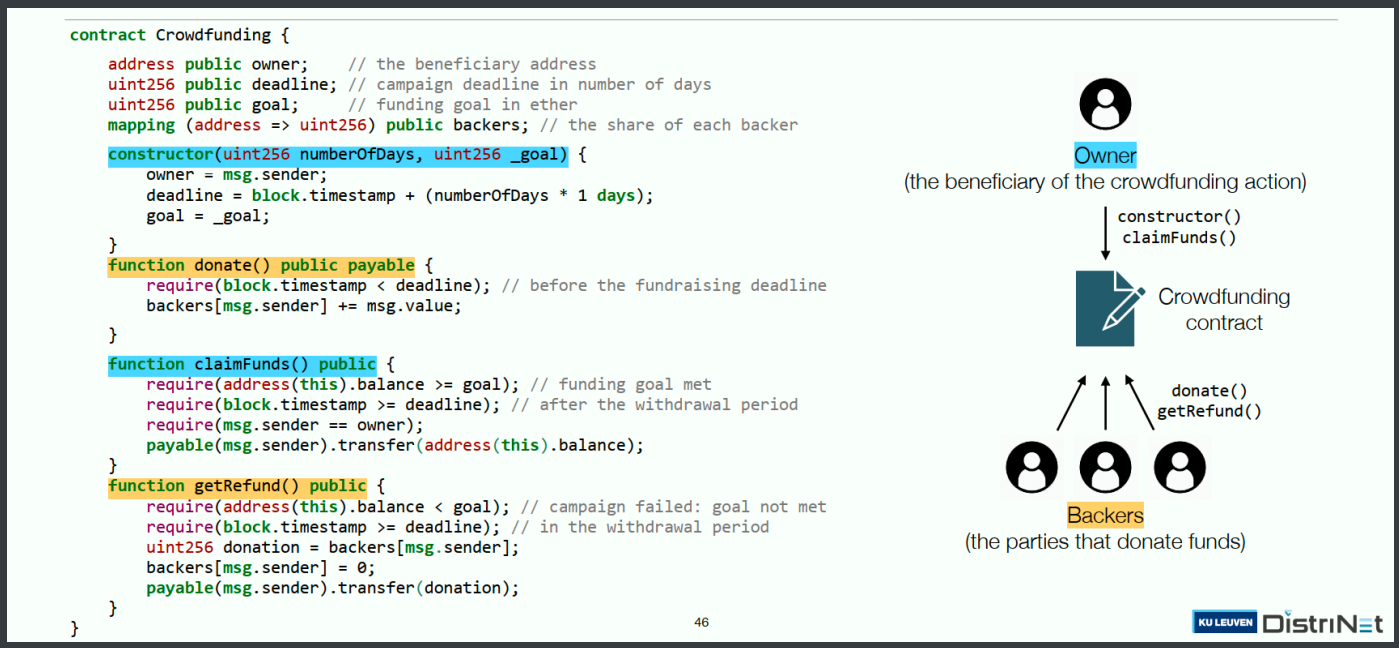
smart contracts & solidity programming language

a basic example


```
contract NameRegistry {

   mapping (string => address) public registry;

   constructor() {}

   function claimName(string name) public payable {
       require(msg.value >= 1 ether);
       if (registry[name] == address(0)) {
           registry[name] = msg.sender;
       }
   }

   function ownerOf(string name) public view {
       return registry[name];
   }
}
```

a more complete example: a crowdfunding contract





Ethereum: accounts, transactions, blocks

accounts

- externally-owned accounts

  associated with a public-private key pair

  ether balance + nonce

  account address based on hash(public key)
- contract accounts

  not associated with a public-private key pair

ether balance + nonce + storage + code

account address based on hash(sender, nonce)

transactions

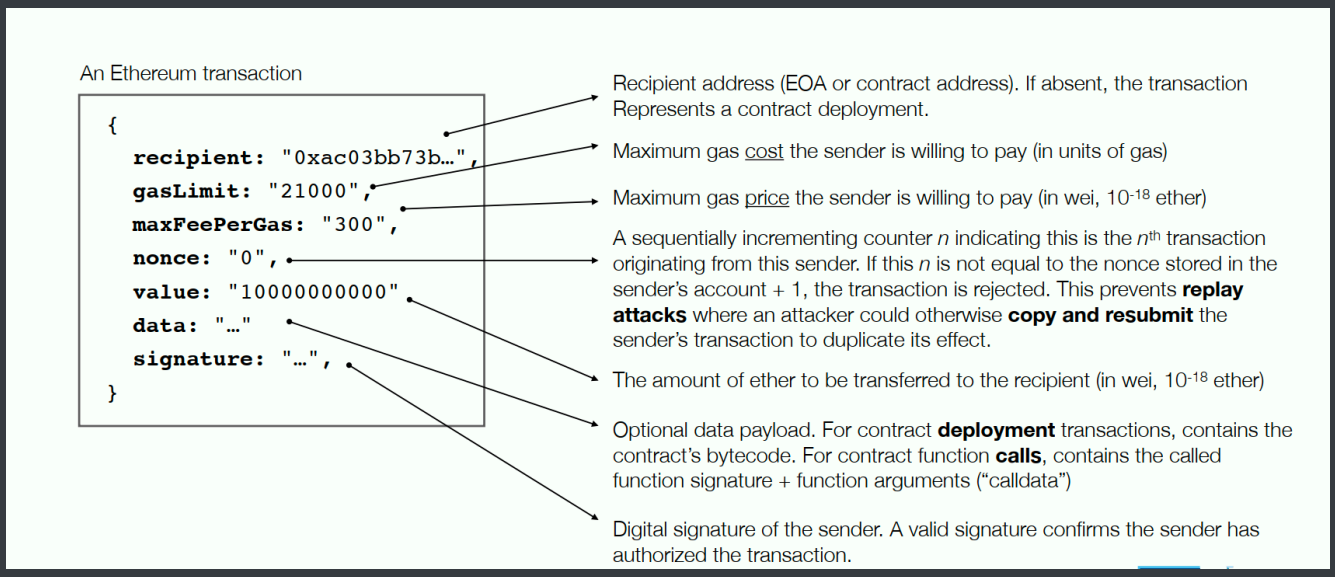
- 3 kinds of transactions:

  simple payment transactions,

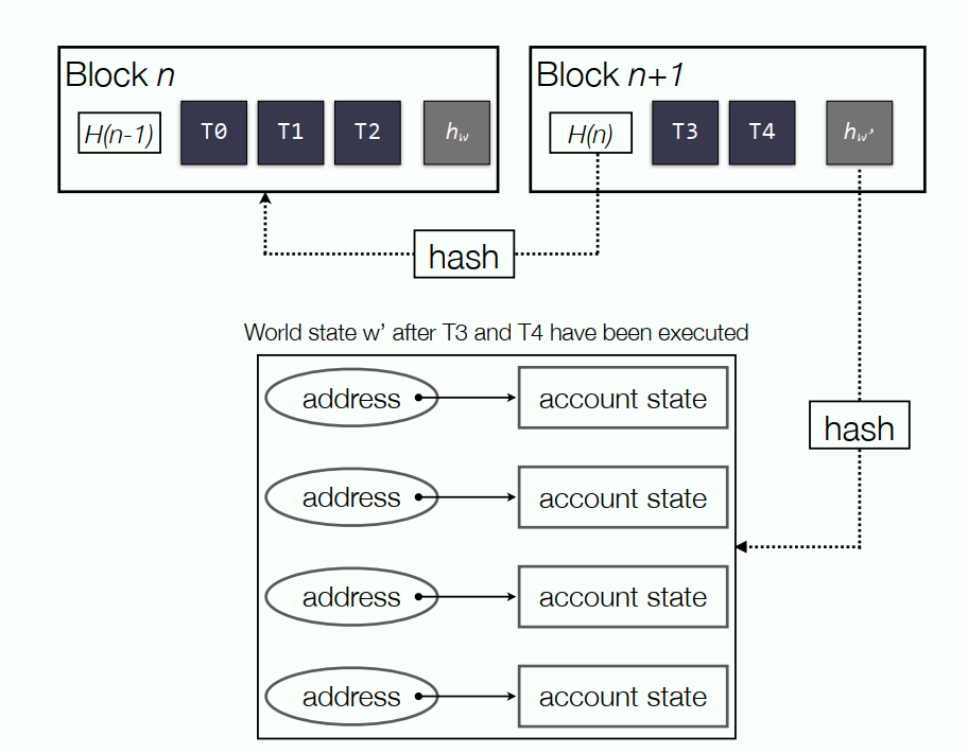
  transactions that deploy contract code to the blockchain,

  transactions that call functions on previously deployed smart contracts

- transaction format



- "world state": mapping from account addresses to account state
- hash pointer



- gas fees

  each operation code has a gas cost

  gas cost is computed in real-time

  gas limit ensures a functions call always has a finite execution time

  transaction aborted if the function call "runs out of gas"

Ethereum: Proof-of-Stake consensus

a validator's **voting power** is proportional to its **stake**

strong economic incentives to remain honest