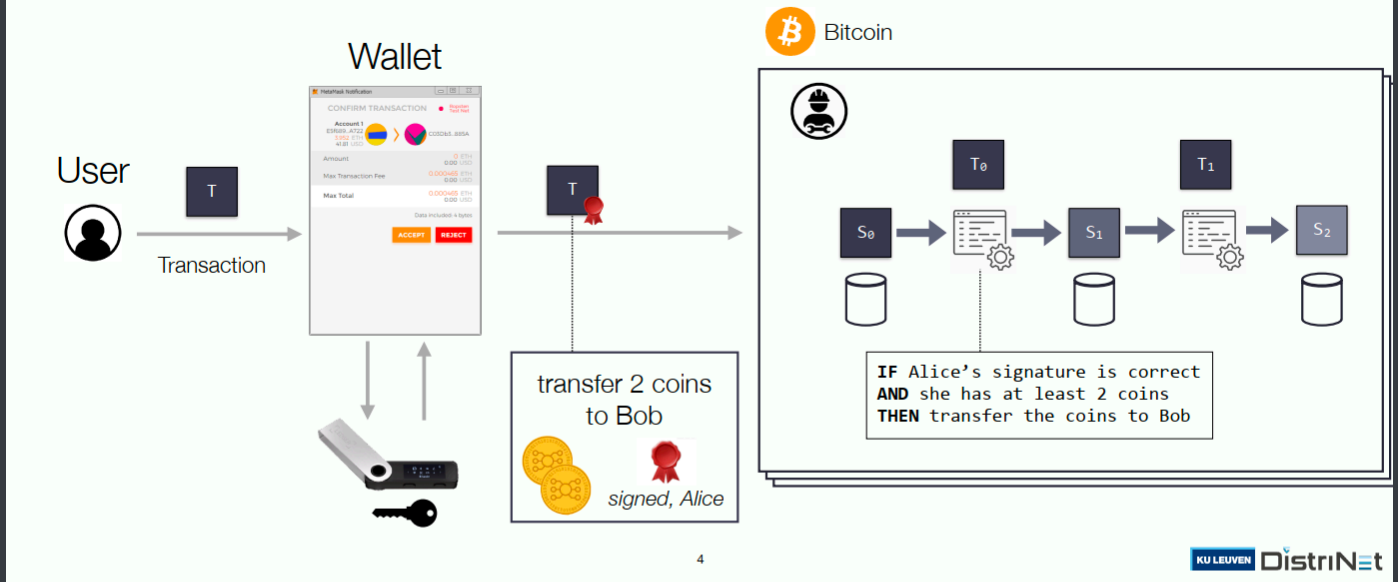


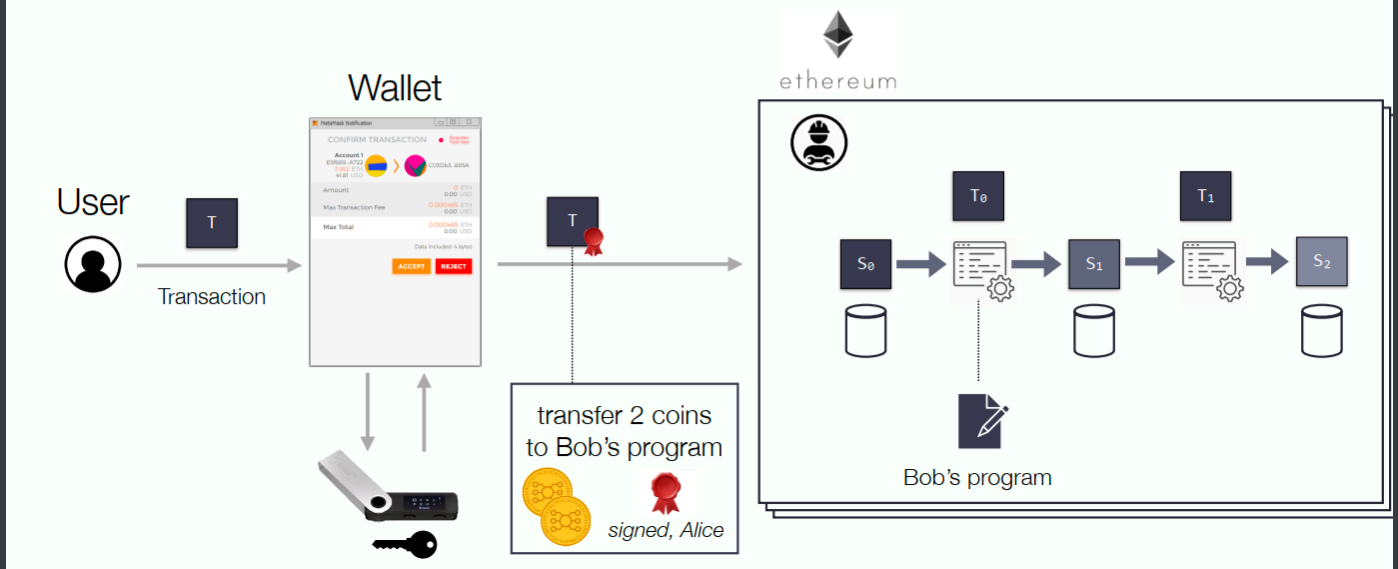
# Ethereum

ethereum is a programmable blockchain

Recall: blockchain networks are replicated state machines



Ethereum: a *programmable* replicated state machine



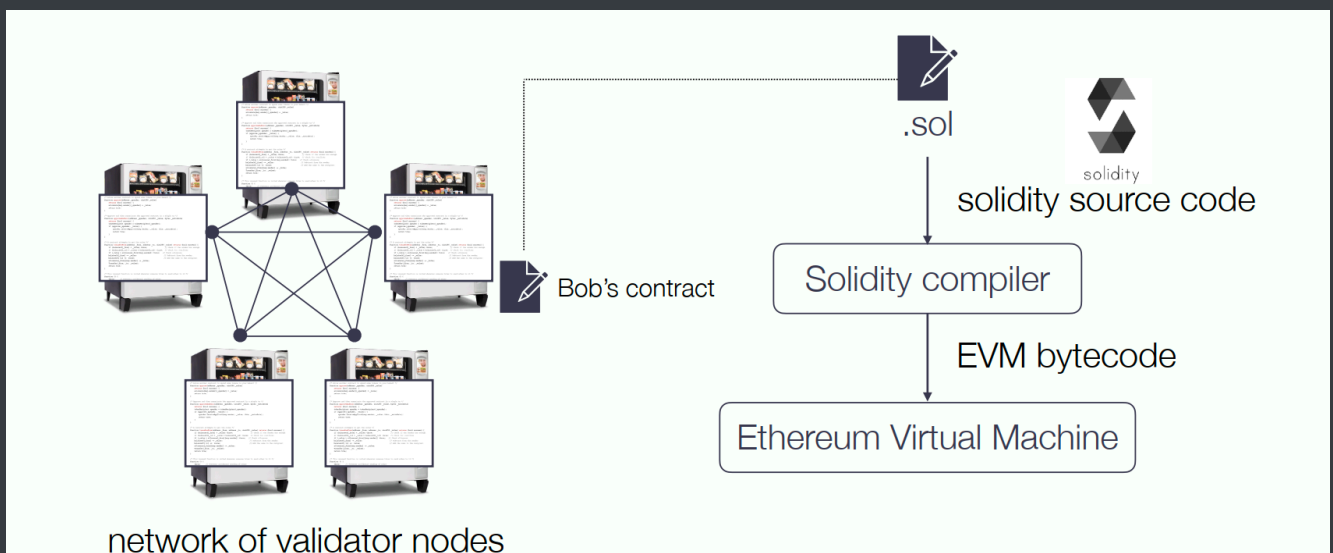
Ethereum 允许用户通过smart contract编程

smart contract是存储在区块链上的自执行代码

## smart contracts

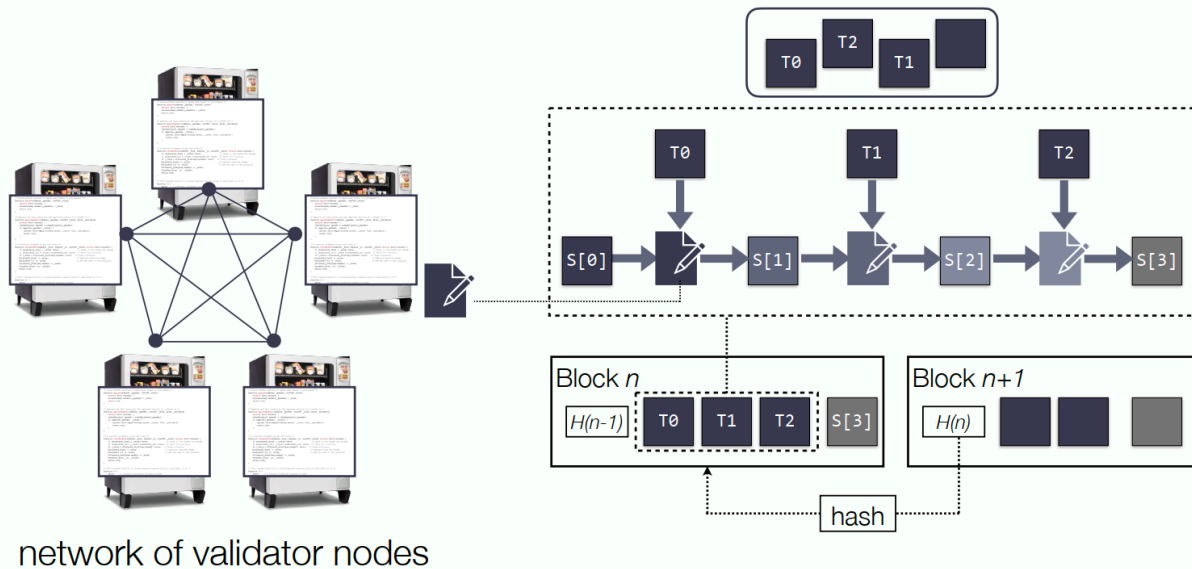
### what is a smart contract

- 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



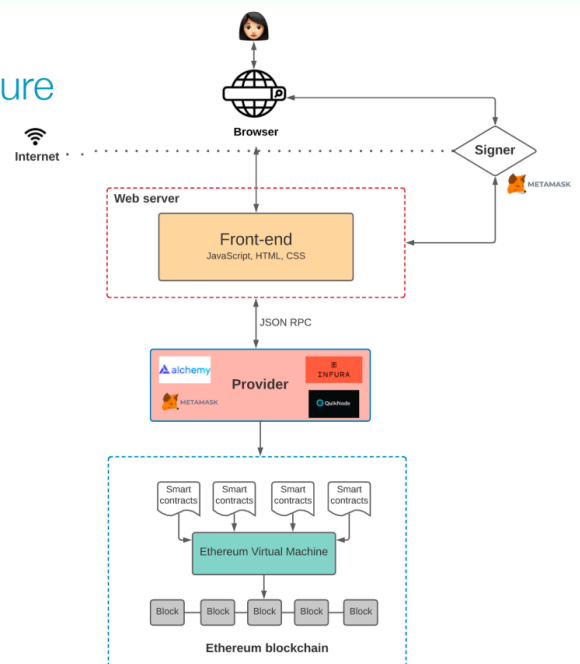
network of validator nodes

## decentralized applications (Dapps)

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

### Decentralized Web application architecture

- Front-end:** largely unchanged (mostly UI logic)
- Back-end:** (part of) the application logic is implemented as a smart contract and published on the blockchain
- Database?** The state of the smart contract is persisted on the blockchain (replicated across all validator nodes)
- Provider:** browsers or mobile apps cannot easily participate directly as a peer in the blockchain network, so usually send their requests through a web server that **relays** the request to a peer in the blockchain network.
- Signer:** for any user action that results in an update to the smart contract, a **signature** is needed from the user. This task typically delegated to a wallet that securely stores the user's keys. The **user retains control** over their keys (they are *not* stored or controlled by the application).



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 hashed from the public key
- access to public key / address allows one to query the account balance,  
access to private key allows one to spend the account balance (signing the transactions)

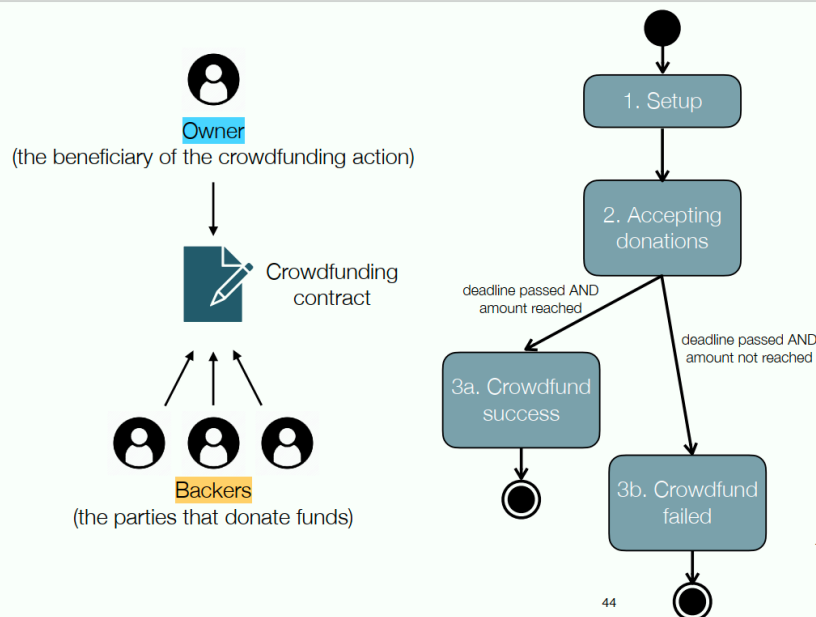
## smart contracts & solidity programming language

### 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



Step 1: the **owner** creates the contract, stating target amount + funding deadline (which **cannot be changed** afterwards)

Step 2: **backers** can donate money (**deposit** funds into the contract)  
IF the funding deadline has not yet passed

Step 3a (crowdfunding successful):  
the **owner** can claim the funds (**withdraw** funds from the contract)  
IF the funding deadline has passed AND the minimum target amount has been met

Step 3b (crowdfunding failed):  
**backers** can reclaim their donations (**withdraw** funds from the contract)  
IF the funding deadline has passed AND the minimum target amount has **not** been met

44

```

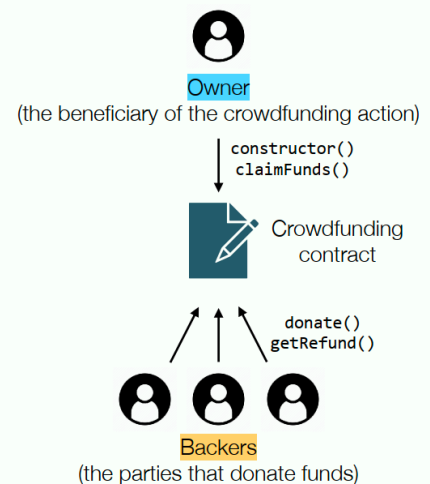
contract Crowdfunding {
    address public owner; // the beneficiary address
    uint256 public deadline; // campaign deadline in number of days
    uint256 public goal; // funding goal in ether
    mapping (address => uint256) public backers; // the share of each backer

    constructor(uint256 numberOfDays, uint256 _goal) {
        owner = msg.sender;
        deadline = block.timestamp + (numberOfDays * 1 days);
        goal = _goal;
    }

    function donate() public payable {
        require(block.timestamp < deadline); // before the fundraising deadline
        backers[msg.sender] += msg.value;
    }

    function claimFunds() public {
        require(address(this).balance >= goal); // funding goal met
        require(block.timestamp >= deadline); // after the withdrawal period
        require(msg.sender == owner);
        payable(msg.sender).transfer(address(this).balance);
    }

    function getRefund() public {
        require(address(this).balance < goal); // campaign failed: goal not met
        require(block.timestamp >= deadline); // in the withdrawal period
        uint256 donation = backers[msg.sender];
        backers[msg.sender] = 0;
        payable(msg.sender).transfer(donation);
    }
}
  
```



46

## 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

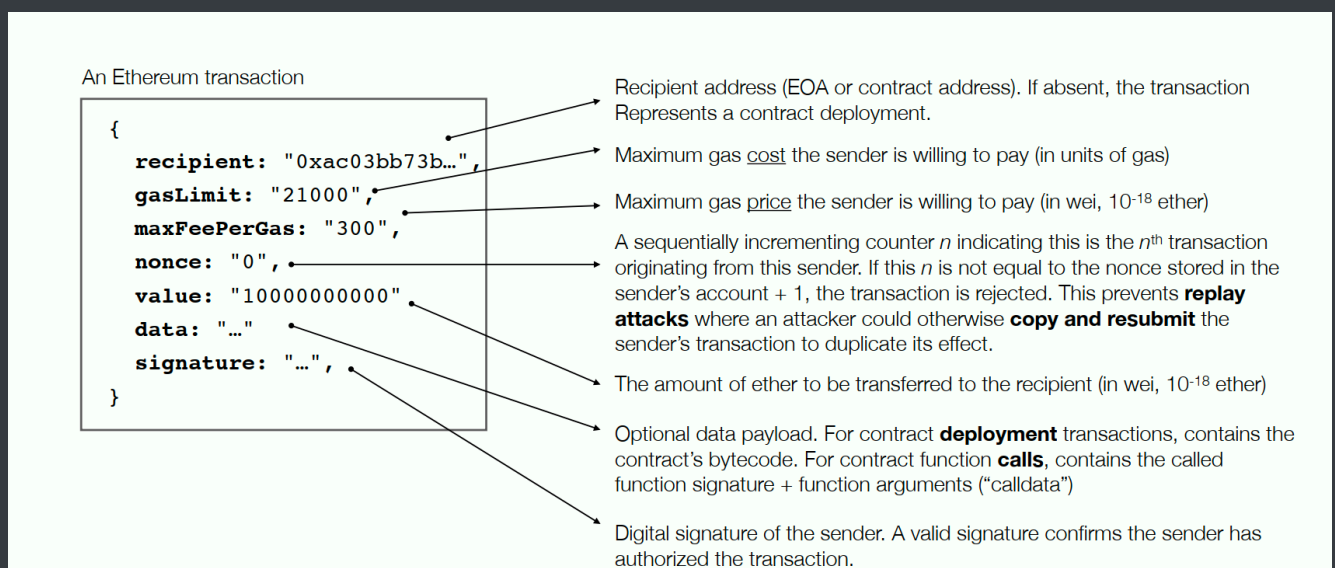
- 3 kinds of transactions:

simple payment transactions,

transactions that deploy contract code to the blockchain,

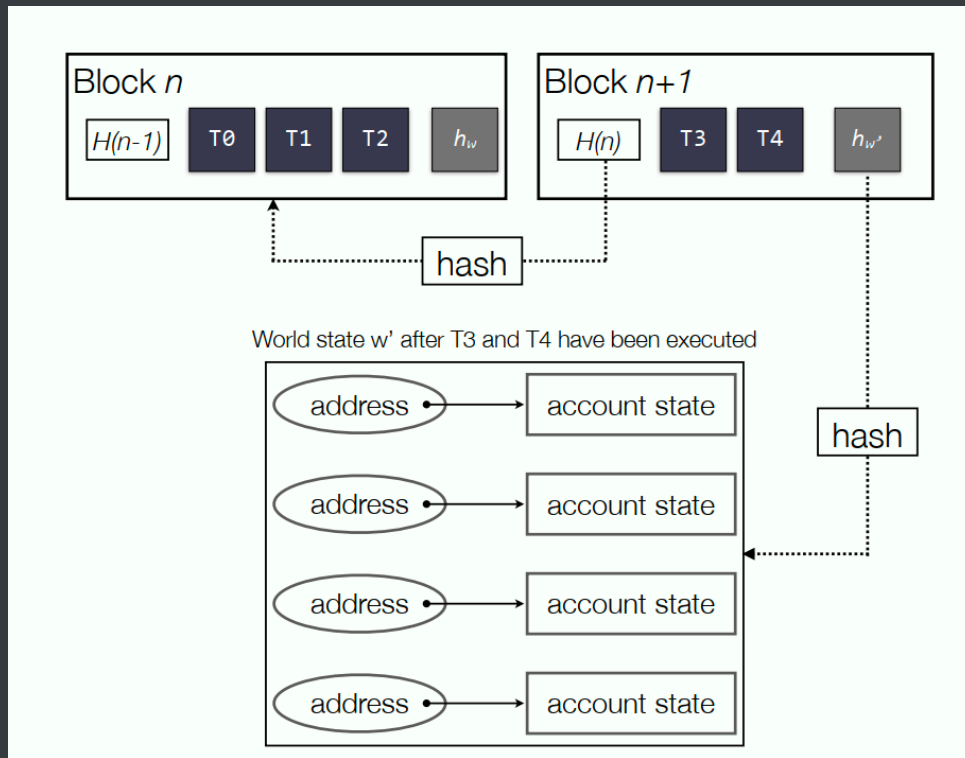
transactions that call functions on previously deployed smart contracts

- 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