DeFi | Academy

Introduction to

Smart contracts

By Edson Alcala

Course content

You will learn:

- How smart contracts work
- How to develop smart contracts
- How to deploy smart contracts to the Ethereum Blockchain

Pre requisites

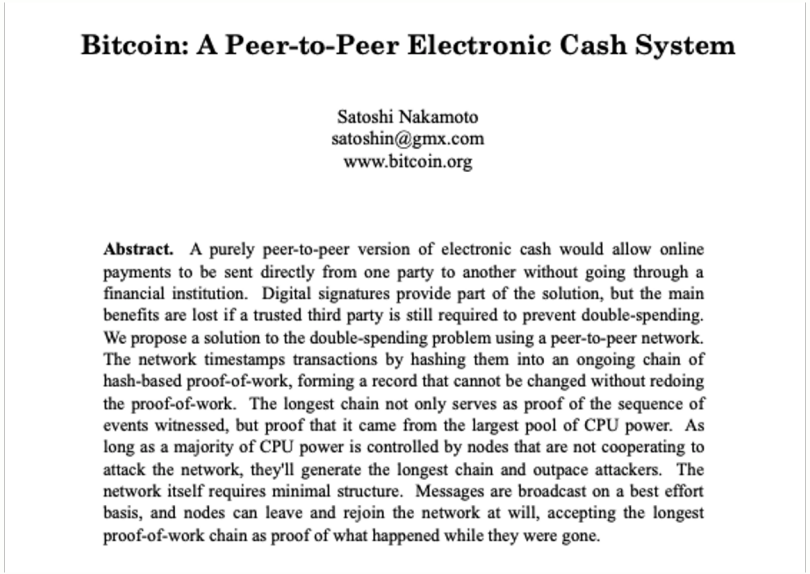
You will need

- Basic knowledge of Blockchain technologies

Introduction to Ethereum

A quick introduction to Ethereum

A short history: Bitcoin 2008



Bitcoin: A Peer-to-Peer Electronic Cash System

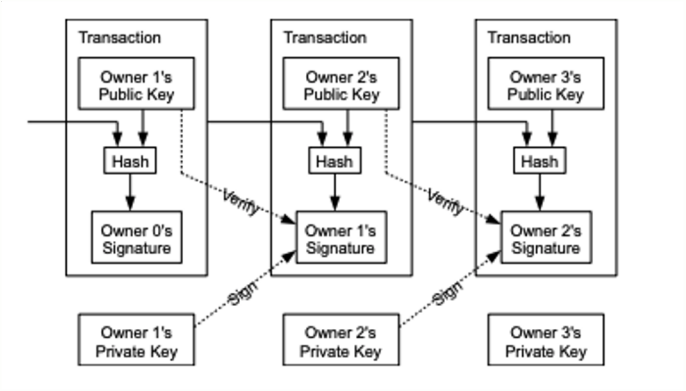
Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

source: https://bitcoin.org/bitcoin.pdf

A short history: Bitcoin 2008

- Peer to peer electronic cash system
- Cryptography (digital signatures)
- Consensus algorithms (PoW)
- Blockchain (Chain of digital signatures)



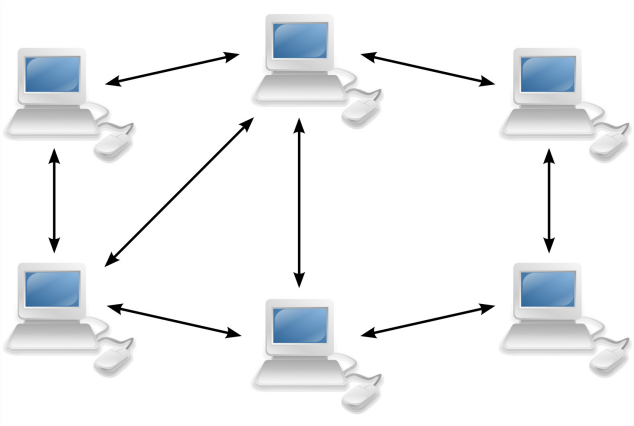
source: https://bitcoin.org/bitcoin.pdf

A short history: Bitcoin 2008

- Nodes around the world running a program

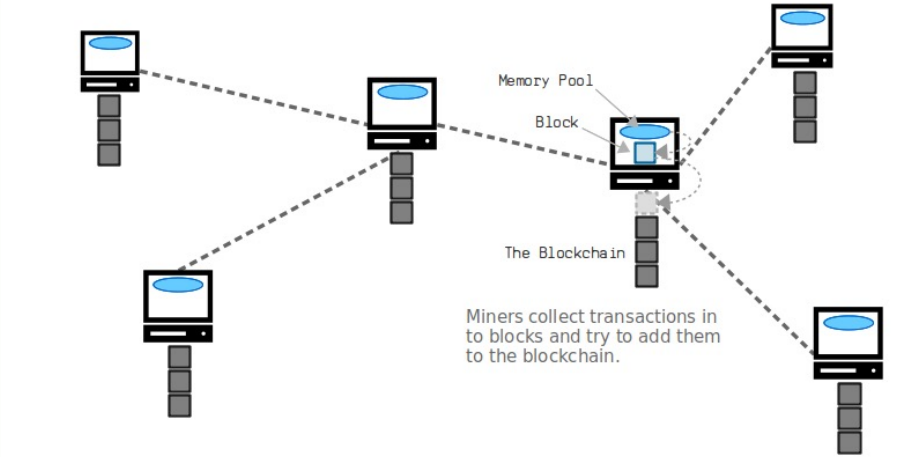
A short history: Bitcoin 2008

- Nodes are connected in a peer to peer network



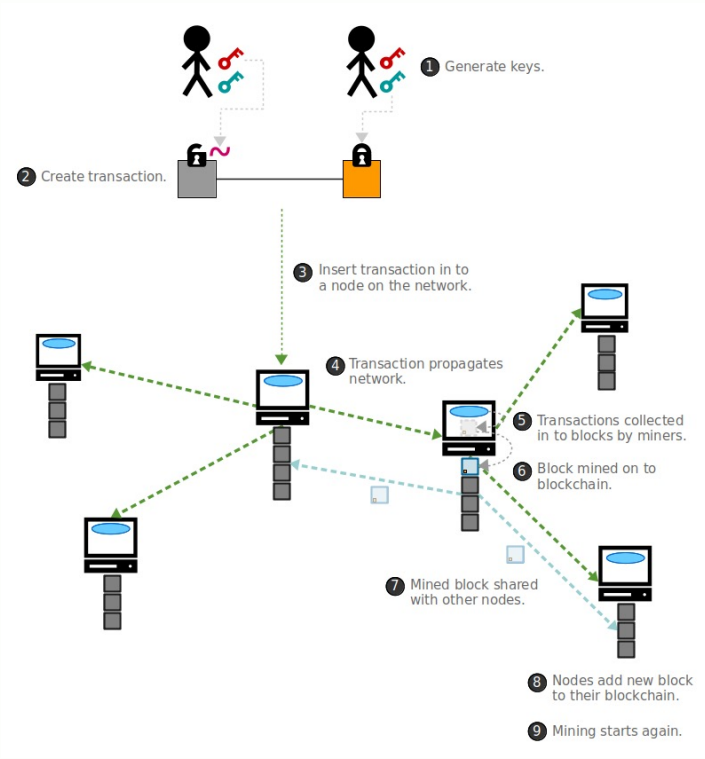
A short history: Bitcoin 2008

- Each node is collecting transactions and putting them into Blocks



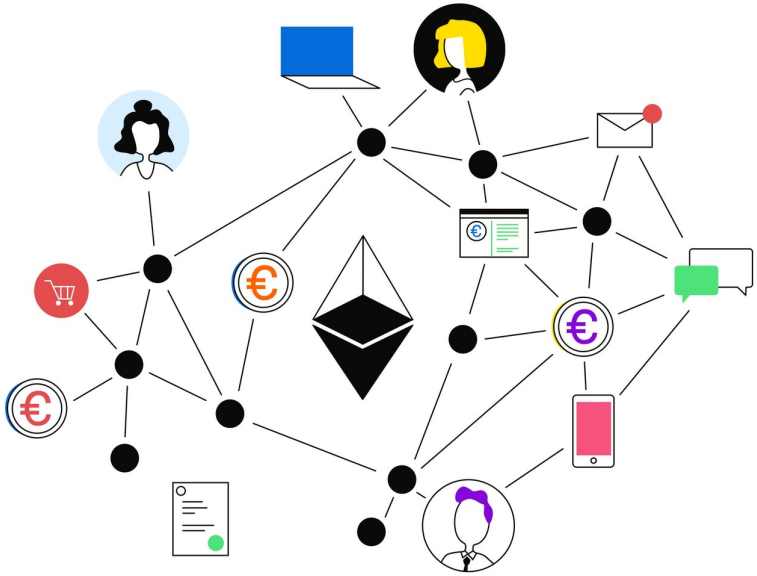
A short history: Bitcoin 2008

- Each node is collecting transactions and putting them into Blocks
- Then, each node will try to solve the PoW
- Once 1 node solves the PoW, it will broadcast to all nodes
- All nodes verify the Block
- And repeat



Ethereum

- Ethereum introduces the idea of smart contracts and decentralized applications.
- The “World Computer”



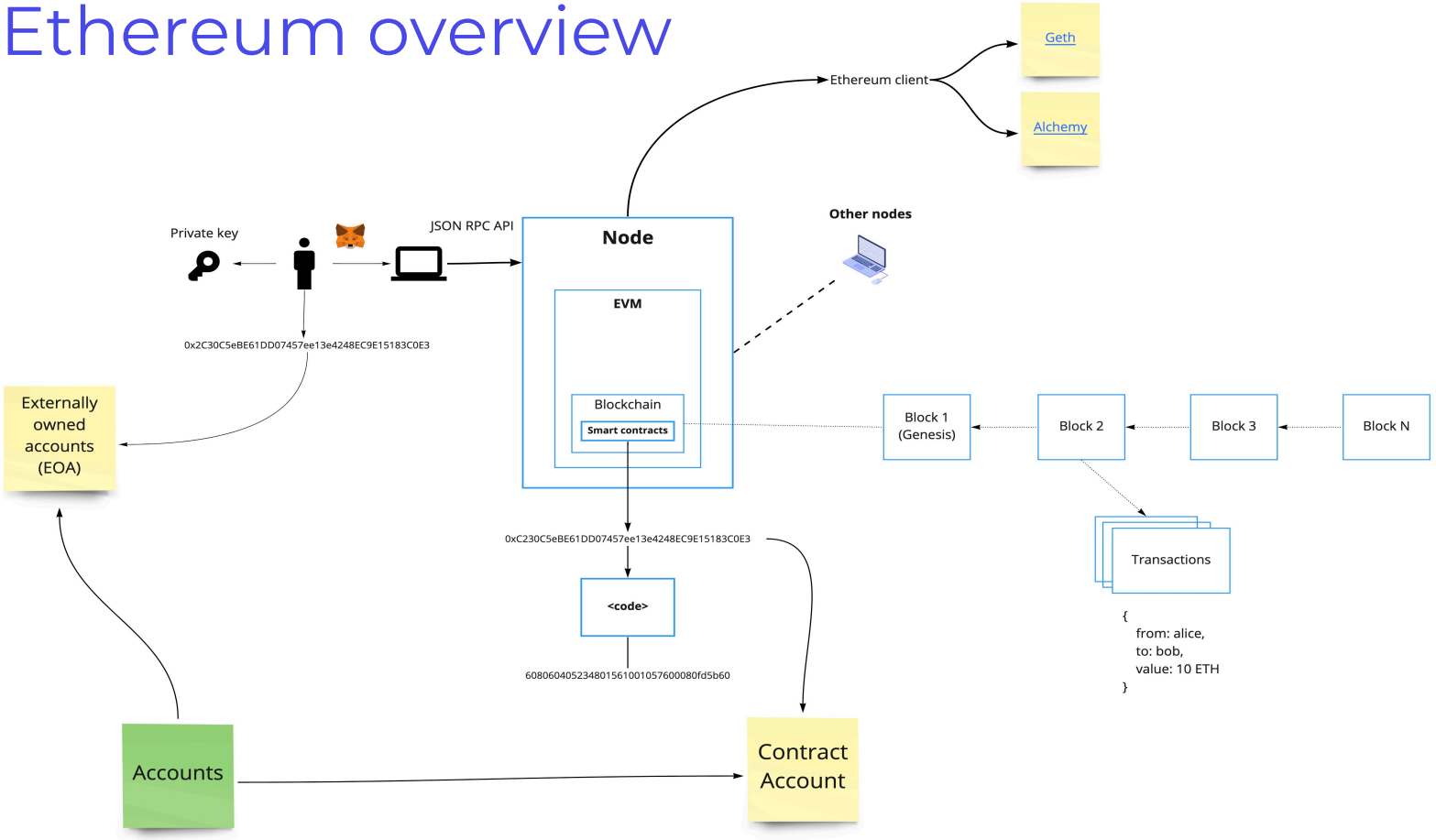
Similar concepts

- Miners
- Blocks
- Transactions

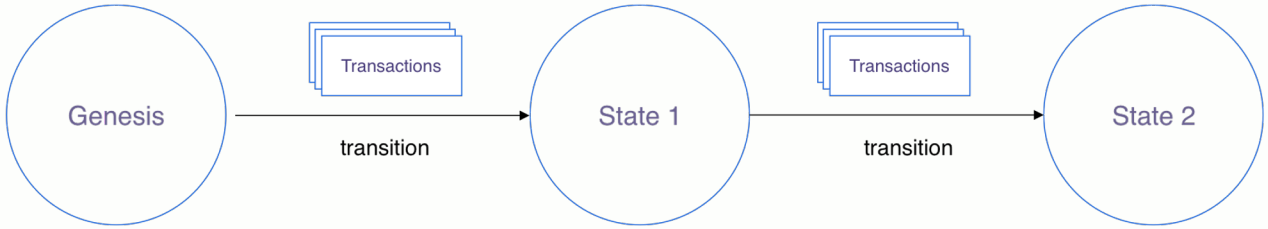
New concepts

- Ether
- Accounts
- Smart contracts
- Gas
- Ethereum Virtual Machine (EVM)

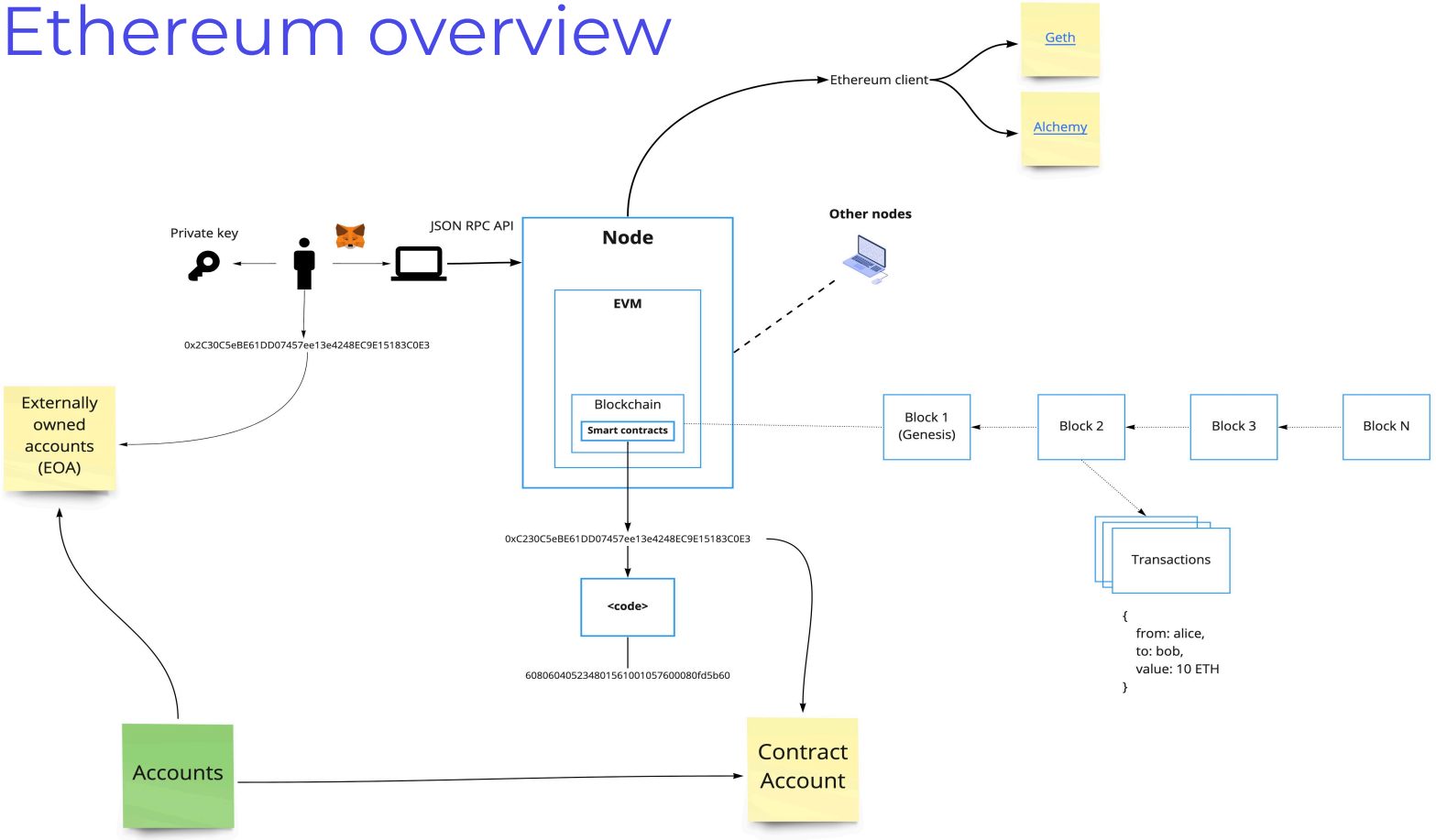
Ethereum overview



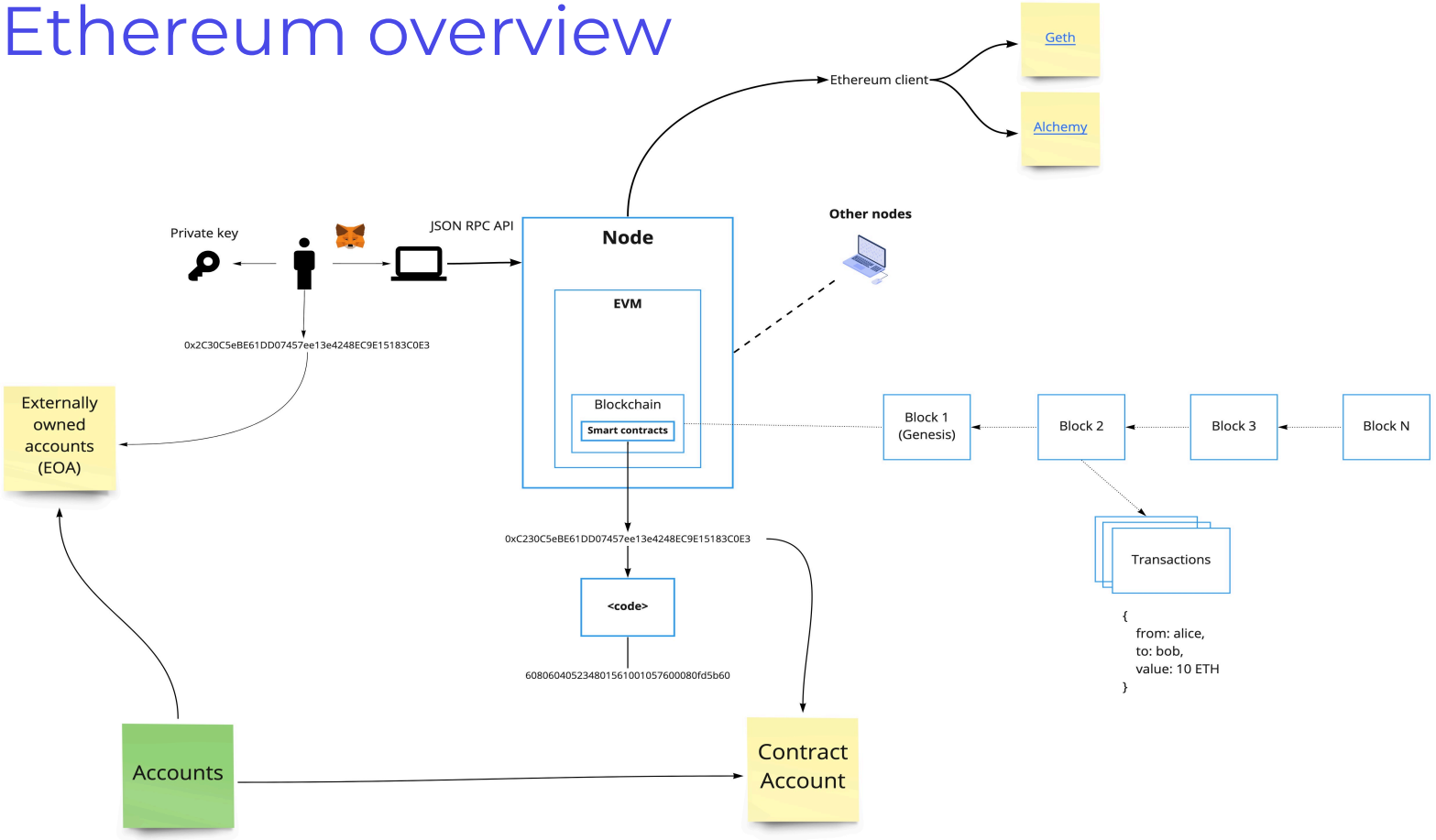
Transactions



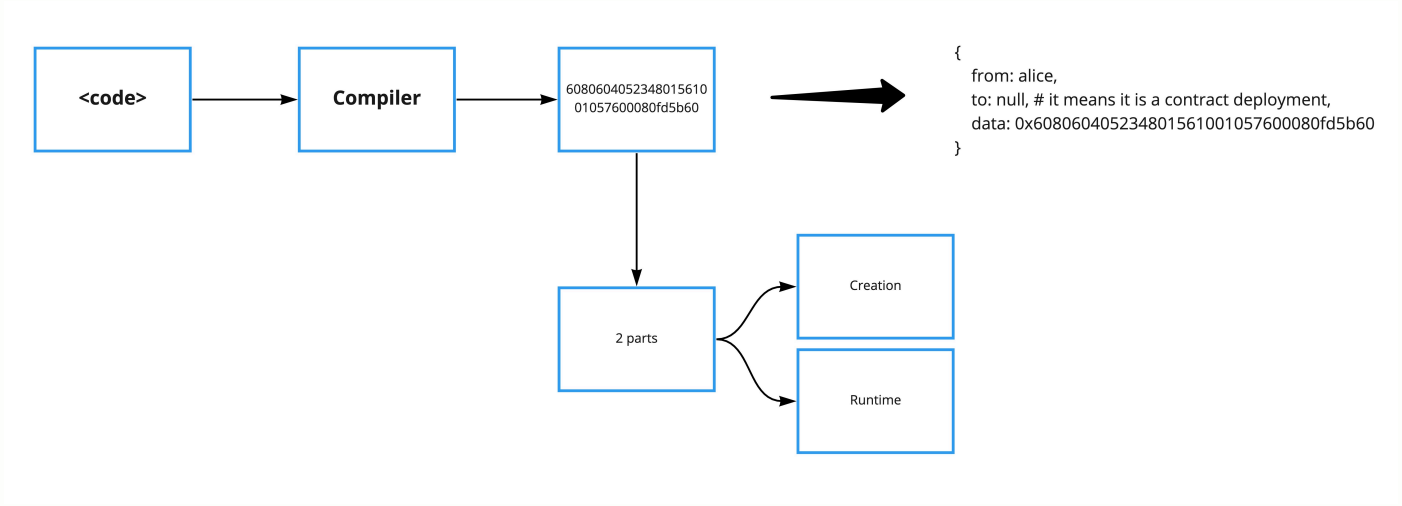
Ethereum overview



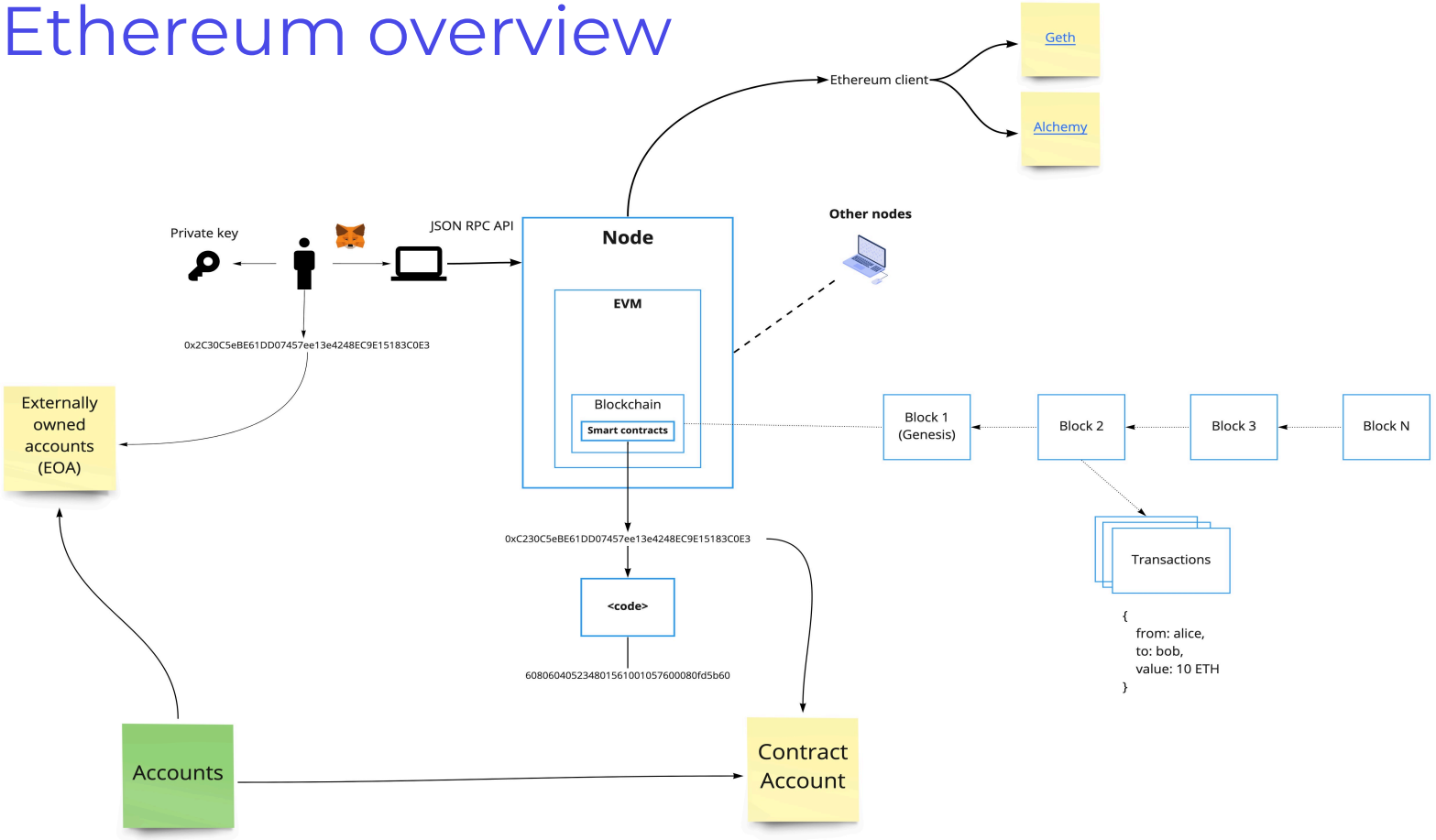
Ethereum overview



Smart contract lifecycle: Basic



Ethereum overview



The Ethereum Virtual Machine

Learning how the EVM works

Ethereum EVM illustrated
Modified

exploring some mental models and implementations

Takenobu T.

WIP

Rev. 0.01.1

Introduction to Solidity

Learning how to code smart contracts using Remix IDE

What is Solidity

- Object-oriented language for implementing smart contracts
- Influenced by C++, Python, JavaScript
- Current version (0.8.7)

Features

- Inheritance
- Libraries
- Data types
- Complex user-defined types
- Variables
- Functions
- Function modifiers
- Events

For more information: https://docs.soliditylang.org/en/v0.8.7/

How to create smart contracts?

- Create source code or write the program
- Compile
- Deploy

Smart contract frameworks

- Hardhat - https://hardhat.org (Recommended)
- Waffle - https://getwaffle.io/
- Truffle - https://www.trufflesuite.com/

Online IDE

- Remix - https://remix.ethereum.org/

Note:

We will use https://remix.defi-academy.com/ due to future incompatible updates to the IDE versions.

Hello Remix IDE

A tour of the Remix online IDE and plugins setup

https://remix.defi-academy.com

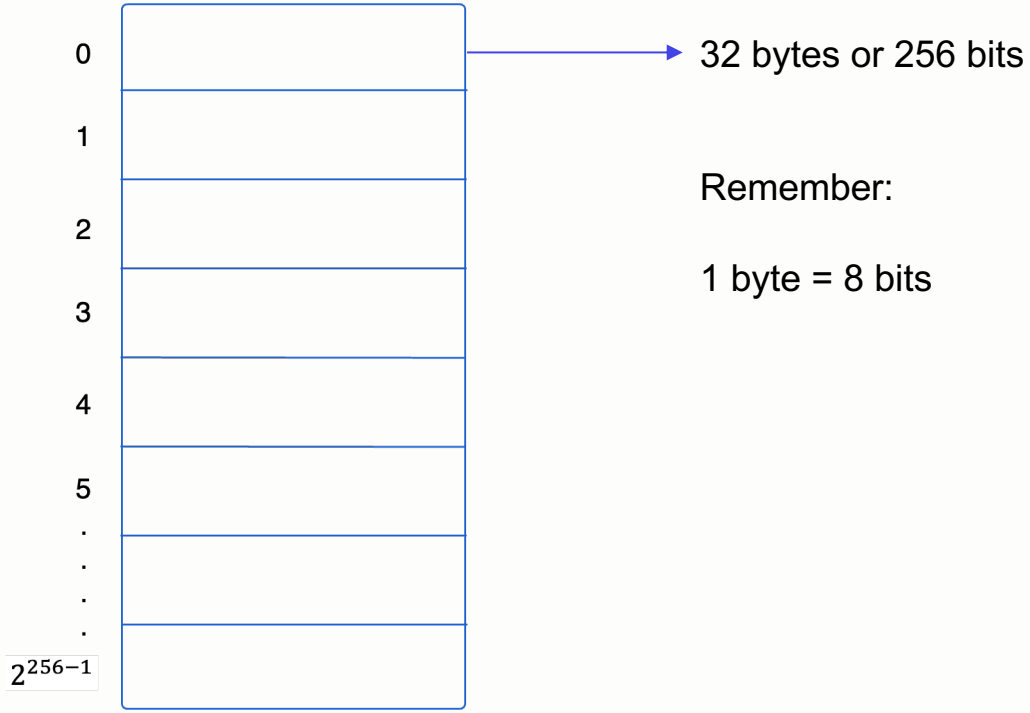
Exercise 1

- Greeting contract

Exercise 2

- Simple storage contract

How storage works



Remember:

1 byte = 8 bits

How storage works: Fixed-Sized Values

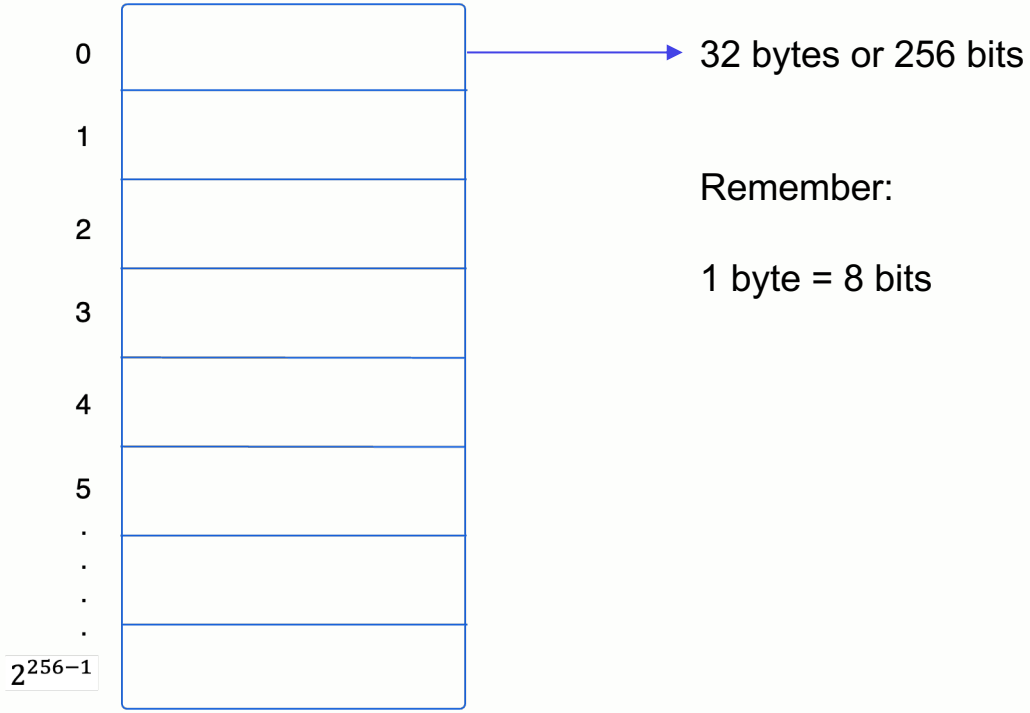
```
contract SimpleStorage {
    uint8 public age;

    constructor() {
        age = 10;
    }
}
```

Exercise 3

- Advance storage contract

How storage works



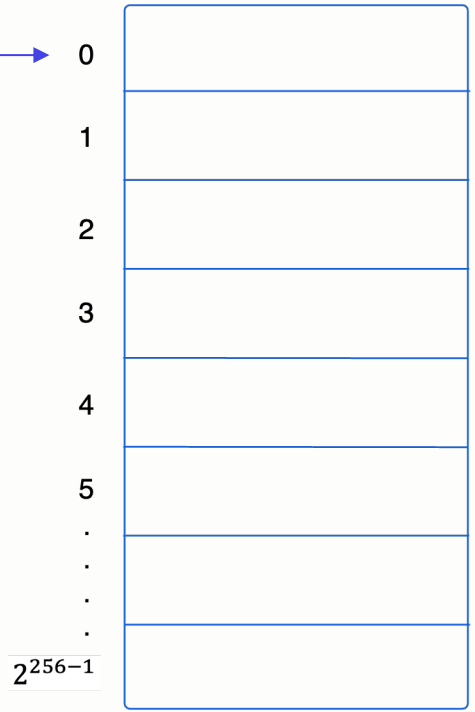
Remember:

1 byte = 8 bits

How storage works: Dynamically-Sized Values

```
contract AdvanceStorage {
    mapping(address => uint8) public ages;

    function register(uint8 _age) public returns (bool) {
        ages[msg.sender] = _age;
        return true;
    }
}
```

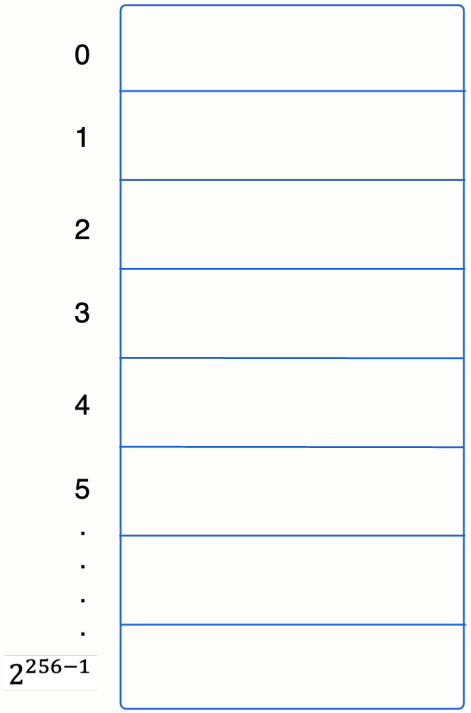


How storage works: Dynamically-Sized Values

```
contract AdvanceStorage {
    mapping(address => uint8) public ages;

    function register(uint8 _age) public returns (bool) {
        ages[msg.sender] = _age;
        return true;
    }
}
```

Value will be stored at: **Hash (key, slot)**



How storage works: Dynamically-Sized Values

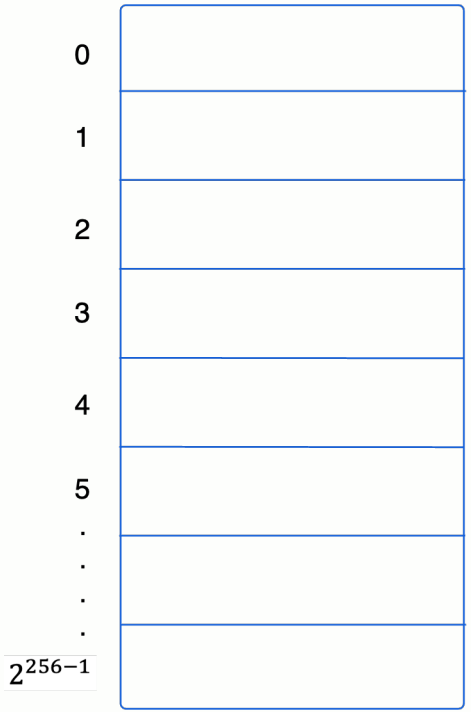
```
contract AdvanceStorage {
    mapping(address => uint8) public ages;

    function register(uint8 _age) public returns (bool) {
        ages[msg.sender] = _age;
        return true;
    }
}
```

Value will be stored at: **Hash (key, slot)**

Example

location of value of age = keccak256(msg.sender, 0)



How storage works: Dynamically-Sized Values

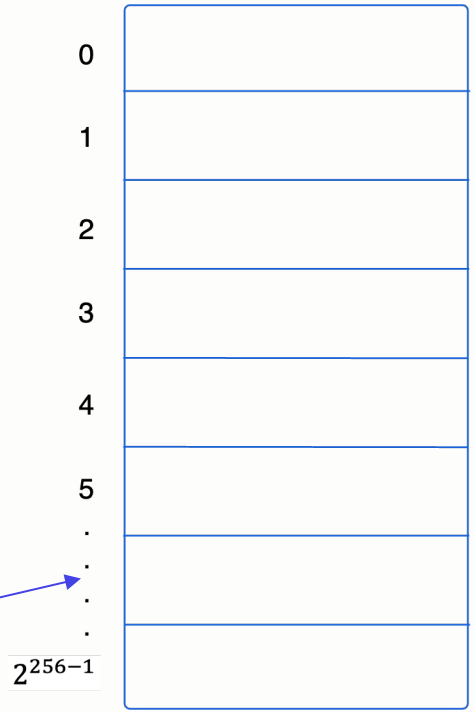
```
contract AdvanceStorage {
    mapping(address => uint8) public ages;

    function register(uint8 _age) public returns (bool) {
        ages[msg.sender] = _age;
        return true;
    }
}
```

Value will be stored at: **Hash (key, slot)**

Example

location of value of age = keccak256(msg.sender, 0)



Exercise 4

- Functions
  - Function visibility
  - Function modifiers

Exercise 5

- Inheritance

Exercise 6

- Structs

Exercise 7

- Events

Deploy smart contracts to a testnet

Deploy smart contracts to a testnet using Remix

Testnets

- Rinkeby
- Ropsten
- Goerli
- Kovan

Installing Metamask

- https://metamask.io/

Exercise

- Deploy the contract to the ropsten testnet

Thanks!

Do you have any questions?

hello@defi-academy.com