


Ethereum Virtual Machine (EVM)

EVM hoạt động thế nào?

Daniel (Sơn) PHAM 

Solidity Developer Bootcamp



Ngày 20 tháng 7 năm 2023

1. The opening of the lecture
2. Ethereum virtual machine
3. Accounts on Ethereum
4. Smart contract
5. QnA

1. The opening of the lecture

- 1.1 Need for a decentralized virtual machine
- 1.2 Introduction to the Ethereum stack

2. Ethereum virtual machine

3. Accounts on Ethereum

4. Smart contract

5. QnA

Sự cần thiết của máy ảo phi tập trung



Trustless Environment

Nhằm thiết lập sự tin tưởng và loại bỏ sự phụ thuộc vào trung gian.

Decentralized virtual machine cho phép run code trong một môi trường không tin cậy, nơi validation và execution có thể được xác minh bởi mỗi bên tham gia.

Execution of Smart Contracts

Cung cấp cơ sở hạ tầng để run logic của smart contracts (smart contracts có thể hiểu như code backend).

Môi trường chạy smart contracts sẽ đảm bảo code chạy một cách nhất quán trên tất cả các nodes tham gia, bất kể môi trường riêng của chúng.

Consensus Mechanisms

Đảm bảo logic của code smart contracts trên mỗi nodes tham gia mạng lưới là hoàn toàn giống nhau.

Cũng đảm bảo trạng thái / dữ liệu của các smart contracts đều giống nhau.

Standardization

Tiêu chuẩn hóa môi trường chạy smart contract.

Tiêu chuẩn hóa này thúc đẩy tính tương thích trên các nền tảng blockchain khác nhau và cho phép nhà phát triển viết hợp đồng có thể thực thi trên bất kỳ blockchain tương thích EVM nào

Introduction to the Ethereum stack



Level 1: Ethereum virtual machine

Level 2: Smart Contract

Level 3: Ethereum nodes

Level 4: Ethereum client API

Level 5: End-user applications

Level 1: Ethereum virtual machine



Máy ảo Ethereum (EVM)

EVM (Ethereum Virtual Machine) là môi trường chạy cho smart contracts trên Ethereum.

Tất cả các smart contracts và các thay đổi trạng thái trên blockchain Ethereum được thực thi thông qua các transactions.

EVM xử lý tất cả quá trình xử lý giao dịch trên mạng lưới Ethereum.

Level 2: Smart Contract



Smart contract

- Smart contracts là các chương trình, code chạy được trên môi trường Ethereum blockchain.
- Smart contracts được viết bằng các ngôn ngữ riêng biệt (solidity, vyper), và được compile thành EVM bytecode (low-level machine instructions called opcodes).

Level 3: Ethereum nodes



Ethereum nodes

- Ethereum nodes là các máy tính / thiết bị đang chạy phần mềm / chương trình gọi là Ethereum client (geth, erigon, openethereum).
Mỗi client tham gia mạng lưới Ethereum sẽ verify tất cả giao dịch trong blocks, giữ cho mạng lưới được an toàn, và dữ liệu chính xác.

Ethereum nodes

- Full node
- Light node
- Archive node

Level 4: Ethereum client API



Client API

Các thư viện được xây dựng và bảo trì bởi cộng đồng, giúp cho các ứng dụng kết nối tới Ethereum network dễ dàng và thuận tiện.

Level 5: End-user applications



Application

At the top level of the stack are user-facing applications. These are the standard applications you regularly use and build today: primarily web and mobile apps.

1. The opening of the lecture

2. Ethereum virtual machine

- 2.1 EVM is the runtime environment
- 2.2 Smart contract code is executed on (EVM)
- 2.3 EVM is a simple stack-based architecture
- 2.4 Several resources as space
- 2.5 EVM code
- 2.6 Execution model

3. Accounts on Ethereum

4. Smart contract

5. QnA

EVM is the runtime environmen



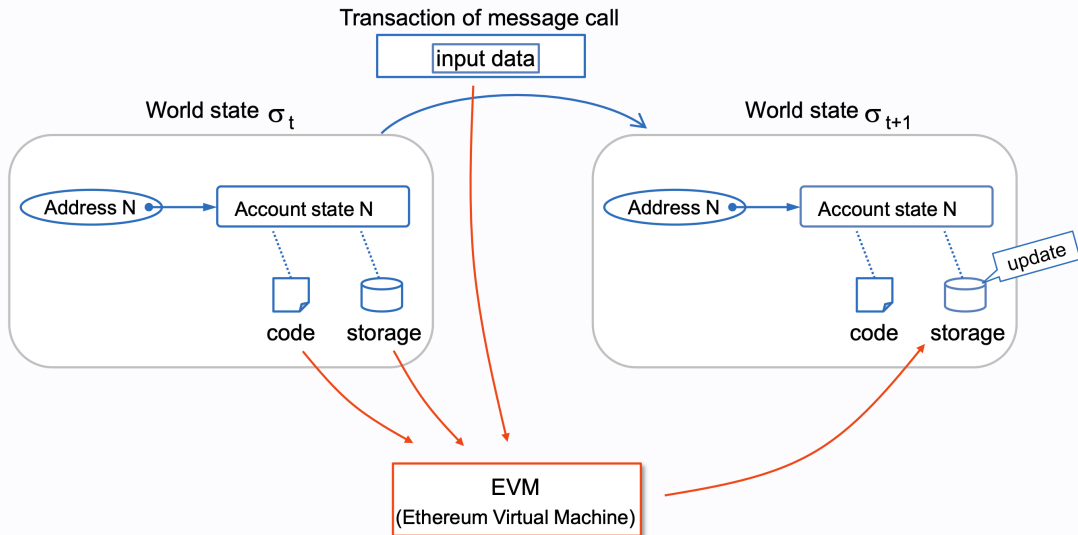
Code

EVM code

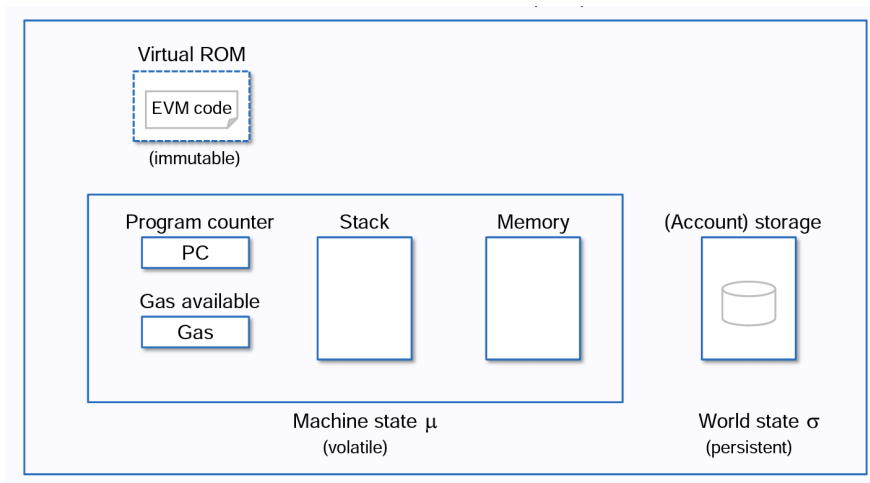
Virtual machine

EVM (Ethereum Virtual Machine)

Smart contract code is executed on (EVM)



EVM is a simple stack-based architecture



Several resources as space

Registers



Stack



stack memory

256 bits x 1024 elements

Memory



volatile memory

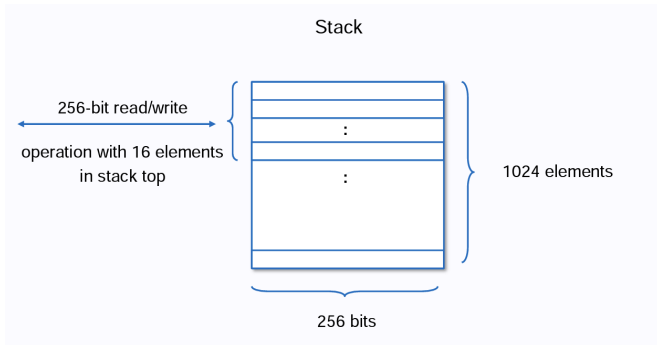
byte addressing
linear memory

(Account) storage



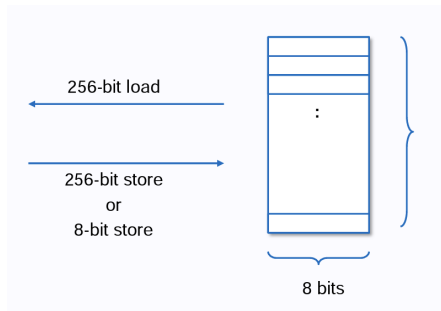
persistent memory

256 bits to 256 bits
key-value store



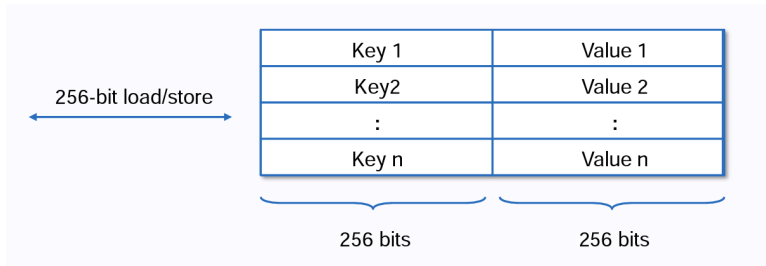
All operation are performed on the stack.

Access with many instructions such as PUSH/POP/COPY/SWAP, etc.



Memory is linear and can be addressed at byte level.
Access with MSTORE/MSTORE8/MLOAD instructions.
All locations in memory are well-defined initially as zero.

Account storage



Storage is a key-value store that maps 256-bit words to 256-bit words.

Access with SSTORE/SLOAD instructions.

All locations in storage are well-defined initially as zero

Assembly view

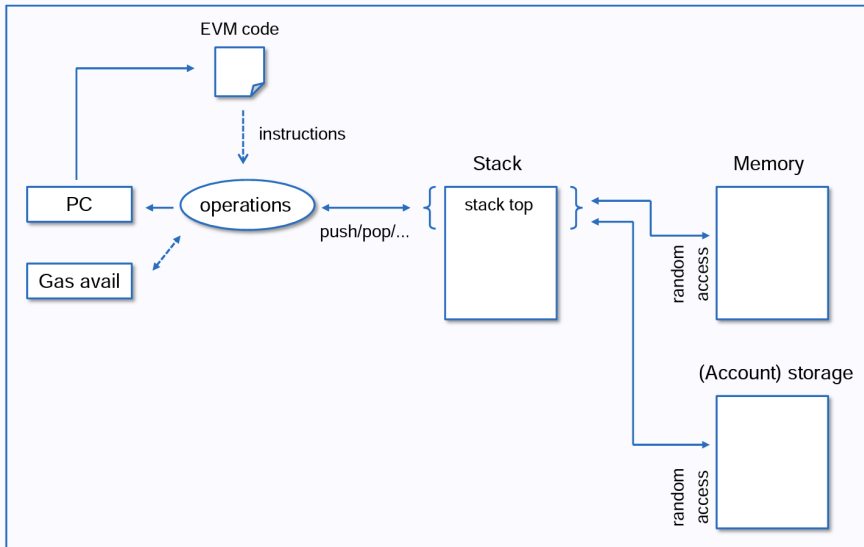
```
PUSH1 e0  
PUSH1 02  
EXP  
PUSH1 00  
CALLDATALOAD  
:
```

Bytecode view

```
0x60e060020a600035...
```

EVM Code is the bytecode that the EVM can natively execute.

Execution model



1. The opening of the lecture

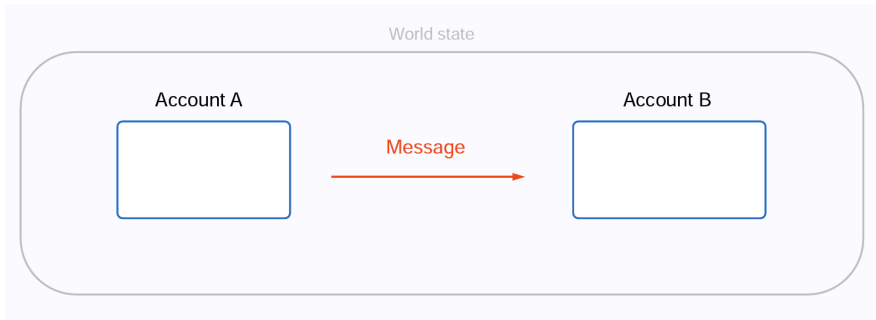
2. Ethereum virtual machine

3. Accounts on Ethereum

- 3.1 Externally-owned account (EOA)
- 3.2 Contract account
- 3.3 Get to know Account Abstraction [ERC-4337]

4. Smart contract

5. QnA



Accounts have the ability to:

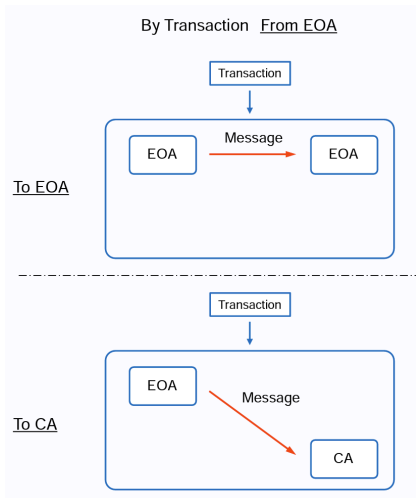
- Receive, hold, and send ETH and tokens
- Interact with deployed smart contracts

The message is passed between two Accounts.

The message is Data (as a set of bytes) and Value (specified as Ether).

EOA

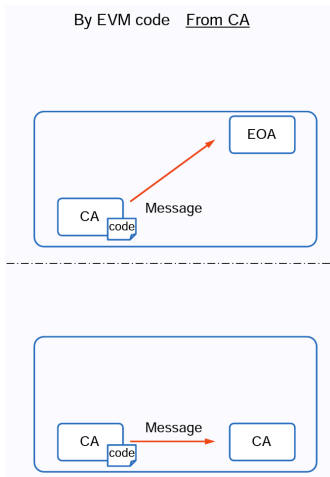
Externally-owned account (EOA) – controlled by anyone with private keys.



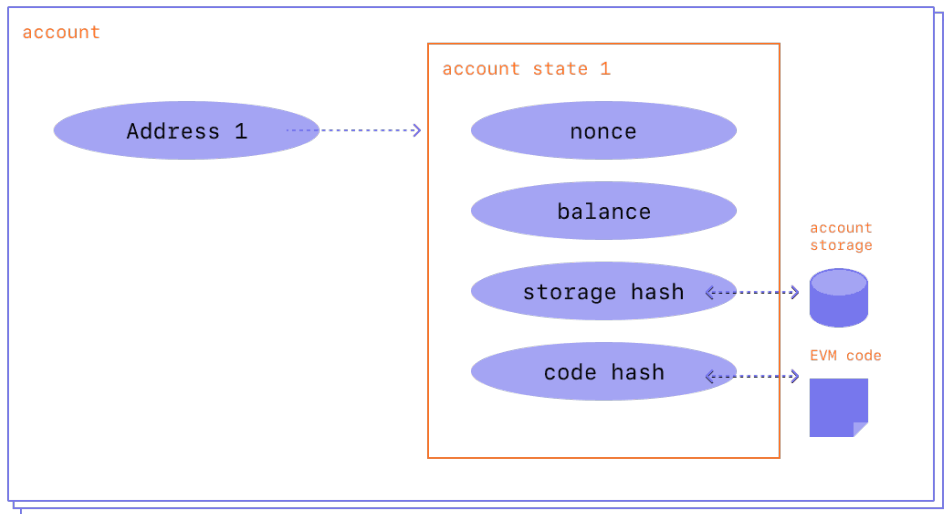
- Creating an account costs nothing
- Can initiate transactions
- Transactions between externally-owned accounts can only be ETH/token transfers
- Made up of a cryptographic pair of keys: public and private keys that control account activities

Contract account

Contract account – a smart contract deployed to the network, controlled by code.



Account fields



Contract account



- Creating a contract has a cost because you're using network storage
- Can only send transactions in response to receiving a transaction
- Transactions from an external account to a contract account can trigger code that can execute many different actions, such as transferring tokens or even creating a new contract
- Contract accounts don't have private keys. Instead, they are controlled by the logic of the smart contract code

Account abstraction refers to a concept in blockchain technology where the functionality and behavior of user accounts can be customized.

- It allows for the development of more flexible and complex smart contracts by separating the execution layer from the underlying account structure.
- With account abstraction, developers can create advanced decentralized applications with enhanced features and optimized gas usage.

1. The opening of the lecture

2. Ethereum virtual machine

3. Accounts on Ethereum

4. Smart contract

4.1 Definition

4.2 Smart contract life cycle

5. QnA

Smart contract definition



Smart contract definition

A "smart contract" is simply a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

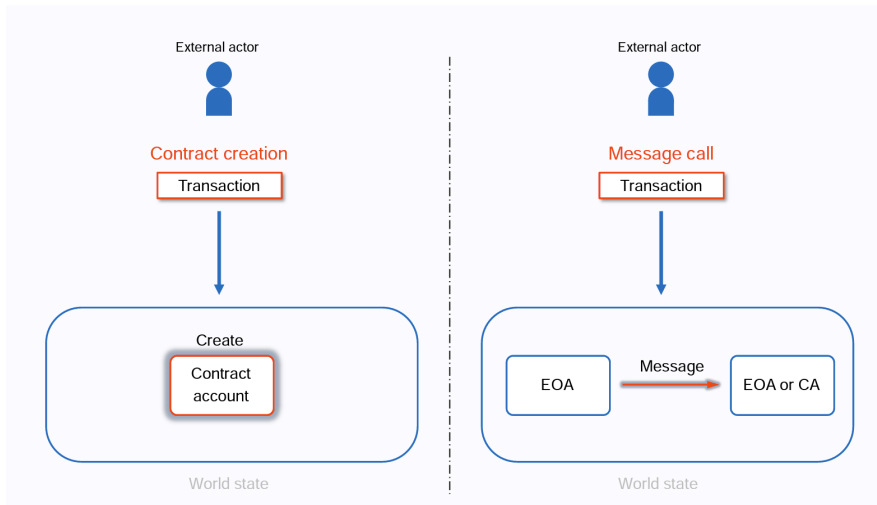
Smart contract life cycle



Smart contract life cycle

- Contract creation
- Compilation
- Deployment
- Initialization
- Activation
- Function invocation
- State updates
- Self-destruction

Contract Creation



1. The opening of the lecture
2. Ethereum virtual machine
3. Accounts on Ethereum
4. Smart contract
- 5. QnA**

