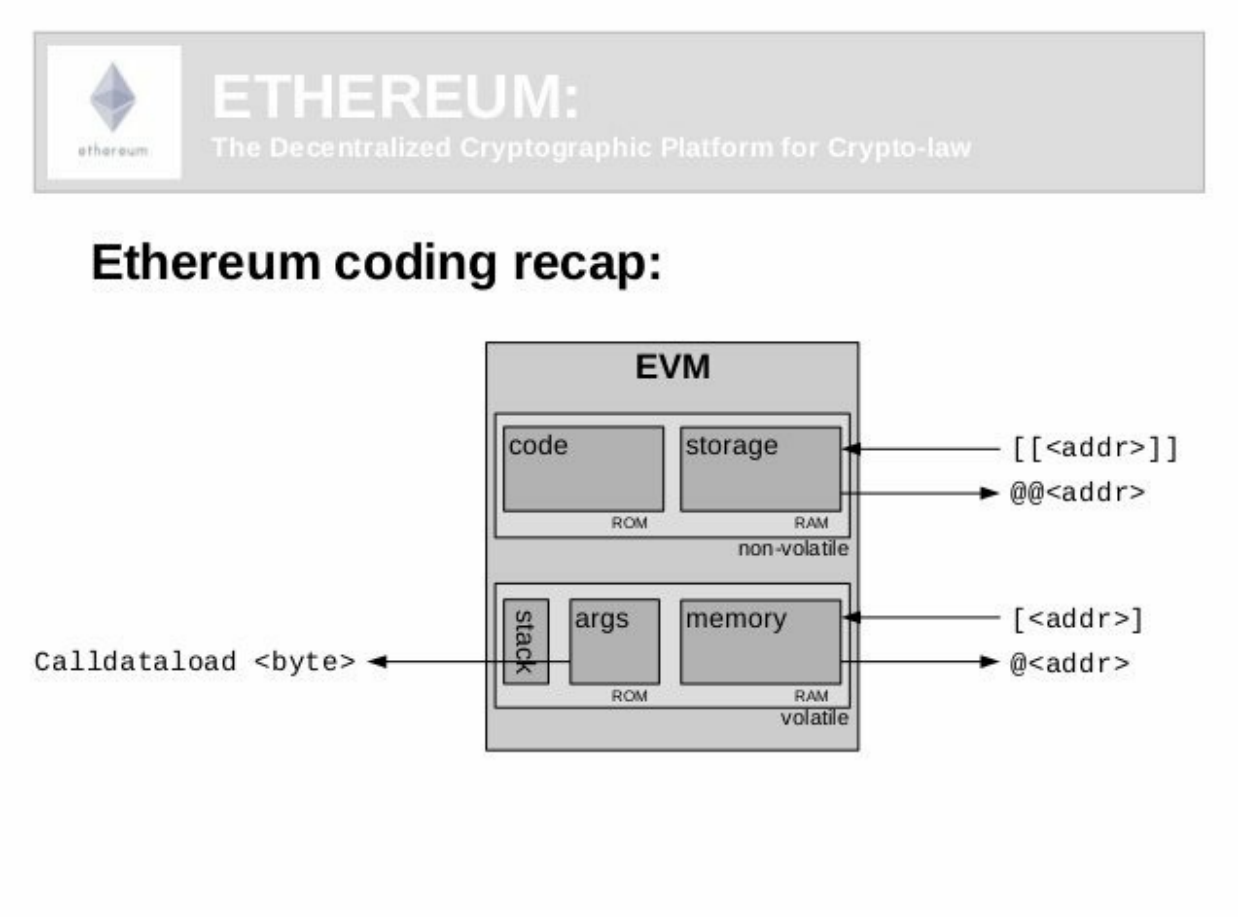
EVM(Ethereum virtual machine)

이더리움에서 스마트 컨트랙트를 실행하기 위한 실행환경 제공
The EVM is a stack-based virtual machine with a memory byte-array and key-value storage

EVM 구조



stack

A virtual stack is being used instead for operations such as parameters for the opcodes

Storage

The storage memory is the memory declared outside of the user-defined functions

and within the Contract context.

```
contract SendBalance {
    mapping ( address => uint ) userBalances ;
    bool withdrawn = false ;
(...)
}
```

Memory

The second memory area is called memory, of which
a contract obtains a freshly cleared instance for each message call.

process

간단한 contract 코드를 실행해서 어떻게 evm 에서 호출하는지 살펴 보겠습니다.

```
contract A {
    string public name;

    function setName(string _name) {
        name = _name;
    }
}
```

contract 에서 setName 이라는 함수를 호출하는 trasaction 만들어 전송합니다

```
// 호출한 method
setName("hi")
// transaction 에서 data parameter
0xc47f0027
0000000000000000000000000000000000000000000000000000000000000020
0000000000000000000000000000000000000000000000000000000000000002
6869000000000000000000000000000000000000000000000000000000000000
```

transaction 에서 data는 Message Call 영역에 셋팅됩니다.
transaction 을 실행하기 위해서 해당 contract Address 에서 code 를 가져와서
instruction set 을 구성해 opcode 를 실행합니다.

실행하는 과정에서 instruction set 순서대로 stack과 memory 필요한 데이터를
저장하고 호출(push/pop)하는 과정을 반복합니다.
(CALLDATALOAD 하면 statck 에 쌓임, CALLDATACOPY 하면 Memory에 쌓임)

Message call 은 CALL instruction 에 의해 triggered 되고 각 instruction의 arguments 와 return values 는 memory 를 통해 전달

참고

Ethereum Solidity: Memory vs Storage & How to initialize an array inside a struct