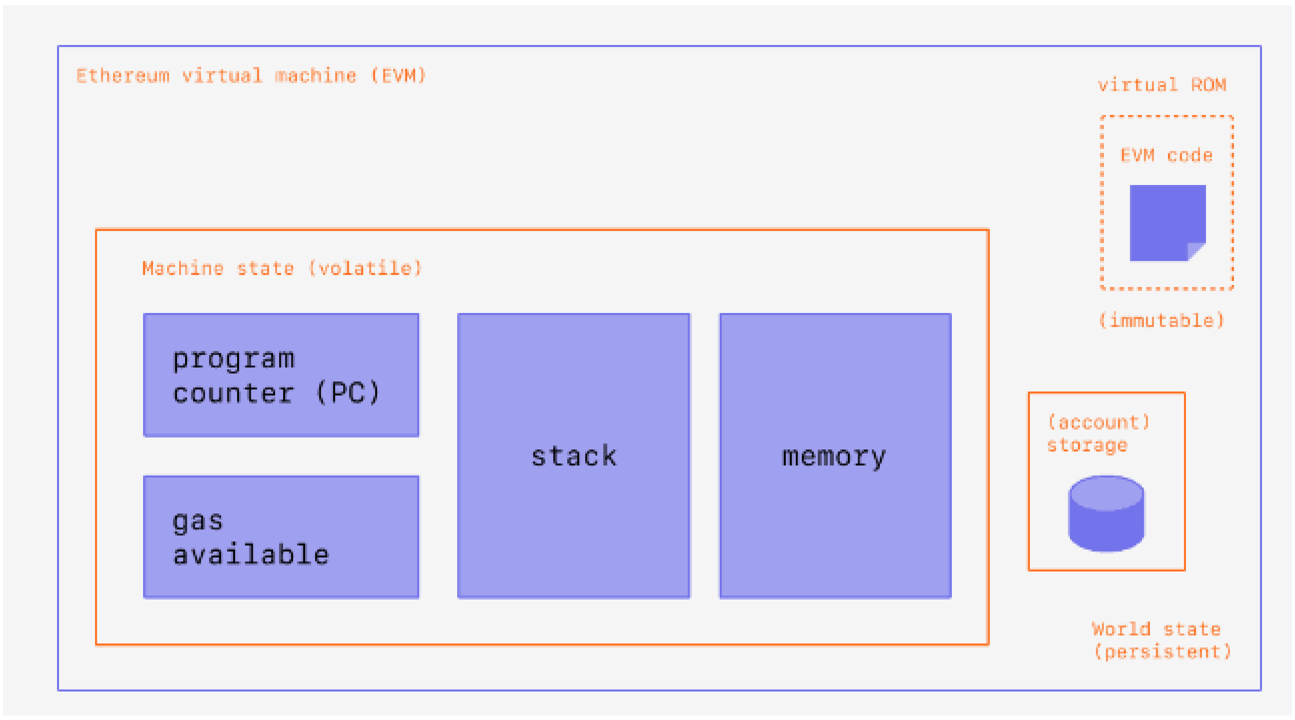
What is an Ethereum Virtual Machine, And How Does It Work?

All you need to Know about Ethereum Virtual Machine

It’s been recent that I delved into the Web3.0 domain and EVM, EVM-compatible, and EVM adaptable protocol are some of the most frequent terms I have come across.
EVM, or Ethereum Virtual Machine, is probably the most important invention of the decade. With Ethereum becoming a global settlement layer, $100bn worth of transactions are being processed daily using EVM.
So, What is the Ethereum Virtual Machine? How does it work? And What are the popular blockchains compatible with EVM?
Let’s have a look at these questions in a row.
Table of Content:

What is the Ethereum Virtual machine?
EVM lies at the heart of the Ethereum protocol that handles the deployment and execution of smart contracts on the Ethereum network.
You must be familiar with the concept of a CPU. Very simply, EVM is a virtual CPU. Although, unlike a computer, the Ethereum blockchain is a decentralized ledger, having several nodes processing a code simultaneously.



Reference: https://ethereum.org/en/developers/docs/evm/

Ethereum virtual machine runs inside a software, most commonly, Geth, installed on different nodes or computers. Geth, on the other hand, is an implementation of the Ethereum protocol.

For the Ethereum blockchain, the developers write codes using the Solidity programming language. The solidity code cannot be executed on EVM directly. Thus, they are compiled into low-level machine instructions called opcodes which are further encoded into bytecodes before being processed by the EVM.

More formally speaking, EVM is a distributed state transition function that keeps track of the current state of the blockchain. These states are changed through transactions interacting with the **Ethereum smart contracts**, which the EVM processes in the form of bytecode.

How does Ethereum Virtual Machine work?

EVM is a stack-based, quasi-Turing complete virtual machine. It executes processes to a finite number of computations based on the amount of gas available for a given Ethereum smart contracts execution.

The quasi-Turing complete characteristic of EVM solves the halting problem that might arise in the Ethereum network due to a malicious or accidental run forever.

The EVM is a simple stack-based architecture with a word size of 256-bits that has a number of addressable data components including a program code, volatile memory, and permanent storage.

Opcodes

The solidity code of an Ethereum project cannot be executed on the EVM directly. So, they are first compiled in low-level instructions called Opcodes.

The available opcodes of Ethereum can be divided into the following categories:

1. **Arithmetic Operations**: ADD, MUL, SUB, DIV, SDIV, and more.

2. **Stack Operations** (including stack, memory, and storage management instructions): POP, MLOAD, MSTORE, SLOAD, SSTORE, and more.
3. **Process Flow Operations**: STOP, JUMP, JUMPI, and more.
4. **System Operations**: LOGx, CREATE, CALL, CALL CODE, and more.
5. **Logic Operations**: LT, GT, SLT, SGT,and more
6. **Environmental Operations**: GAS, ADDRESS, BALANCE, ORIGIN, CALLER, CALL VALUE, and more.
7. **Block Operations**: BLOCK HASH, COINBASE, TIMESTAMP, NUMBER, DIFFICULTY, and GASLIMIT.

BYTECODE

Similar to a central processing unit, a high-level language is converted into bytecode instructions, and EVM executes its own bytecode instructions. Here, for Ethereum the higher-level programming used is Solidity.
Bytecode operations or the EVM operation set offers the following operations:

1. Execution context inquiries
2. Stack, memory, storage access- (Probably the most important set of operations)
3. Arithmetic and Bitwise logical operations
4. Logging, calling, and other operators
5. Control flow operations

An opcode is encoded into a Bytecode for its efficient storage, whereby each opcode is allocated a byte.

Turing Complete and gas- Cost of Interacting with the smart contracts

As has been discussed earlier, the Ethereum programming language is Turing complete. But this comes with a persistent issue of halting, i.e. a code taking forever to run.
Although, with gas, there is a fix: the EVM will stop the application from running if it hasn’t finished after a predetermined maximum amount of computation has been completed. Due to this, the EVM is a quasi-Turing-complete machine that can execute any program you feed as long as it finishes running within a certain amount of computation. In Ethereum, that cap isn’t set; you can pay to raise it to a maximum (known as the “block gas limit”), and everyone can agree to raise that cap gradually over time. However, there is a cap on the gas used at once, and transactions that use too much gas while executing are stopped.
When an EVM is required to perform a transaction, it is initially provided with a gas supply equal to the quantity indicated by its gas limit. As the EVM proceeds through the program, its gas supply is depleted since each opcode performed costs gas. The EVM determines whether there is sufficient gas before performing each operation. Execution is stopped, and the transaction is rolled back if there isn’t enough gas.

Deployment of a smart contract

As specified by the Ethereum protocol, the EVM’s role is to compute valid state transitions due to smart contract code execution to update the Ethereum state. Because external actors (account holders and miners) start state transitions by originating, accepting, and ordering transactions, this feature describes Ethereum as a transaction-based state machine.
An EVM is incorporated with all the information necessary concerning the current block being generated and the specific transaction being processed when a transaction results in executing a smart contract code.
The amount of gas purchased by the sender at the beginning of the transaction is the gas supply for this execution, which is a crucial variable.

What are the popular blockchain protocols compatible with EVM?

Because Ethereum is the most widely used settlement layer for smart contract deployment, it has reached some capacity limits. This created space for other blockchain protocols compatible with Ethereum Virtual Machine to utilize the trust of the users in Ethereum.
Following are some of the popular blockchains and Ethereum scaling solutions.

- Polygon(Matic)
- Fantom
- Avalanche
- Binance Chain and Binance Smart Chain
- Polkadot
- TRON
- Celo
- Aurora
- NEAR

And more…

Sum Up:

One of the most significant projects in the cryptocurrency industry, EVM, is a computing engine that functions as a decentralized computer executing millions of applications.
It serves as the virtual machine that underpins Ethereum’s whole organizational framework.
EVM functions as a master computer executing various kinds of transactions on the Blockchain.

References

https://medium.com/mycrypto/the-ethereum-virtual-machine-how-does-it-work-9abac2b7c9e
https://github.com/ethereumbook/ethereumbook/blob/develop/13evm.asciidoc