The Ethereum Virtual Machine (EVM) is a virtual machine that runs on the Ethereum network. It is responsible for executing smart contracts written in the Ethereum programming language, Solidity. Smart contracts are comprised of programmed logic that is executed by transactions that users or other smart contracts call against a contract. A contract is typically an agreement between two or more parties. Here, the reference to smart contract, means that these contracts will enforce these contractual agreements without intermediaries.

The EVM is a deterministic, stack-based virtual machine that is capable of executing complex computations. It uses a specialized programming language called EVM bytecode, which is a low-level, machine-readable language that is compiled from Solidity code.

State

The EVM is responsible for managing the state of the Ethereum network, which includes the balances of all user accounts and the state of all smart contracts. It also manages the execution of transactions and ensures that all nodes on the network have the same copy of the blockchain. This state is maintained within 3 Patricia Merkle Trie structures.

World State Trie: Contains User and Contract state (smart contract code)

Receipts Trie: Contains Smart Contract Events

Transactions Trie: Contains all Transaction

Architecture

A simple stack-based architecture. The word size of the machine (and thus size of stack items) is 256-bit. This was chosen to facilitate the Keccak256 hash scheme and elliptic-curve computations. The memory model is a simple word-addressed byte array. The stack has a maximum size of 1024.

Gas

With the EVM, the concept of gas is foundational to the way code is executed, and to the design, patterns used to program smart contracts. If a transaction’s resources are exhausted, or there is some logical execution error when the contract code runs, the EVM will revert all changes that the transaction affected, and the state will revert to the state prior to the code which ran as a result of the offending transaction.

Computational Resources

The EVM manages the execution of code on the Ethereum network, which is a global, public network. As it is public, and open to anyone, the resources allocated to the EVM must be constrained. This is one key difference between Bitcoin and Ethereum, with Solidity, which is a Turing complete language.

EVM vs Bitcoin

Ethereum allows for more complex execution than the Bitcoin scripting language, such as looping. Because of this expanded feature, the architects of Ethereum needed a way to limit spam, and unlimited execution which could consume the entire network's resources and create a denial of service if, for instance, an infinite loop was included in a smart contract.

Halting Problem

The Halting problem is a famous computer science problem that states given a Turing Machine, it is not possible to tell whether it will halt or run forever. In a decentralized system, there must be an enforcement mechanism to ensure all programs halt or else the entire system could fall to its knees.

To protect against such scenarios (whether intentional or unintentional), the concept of gas was introduced and implemented on the EVM