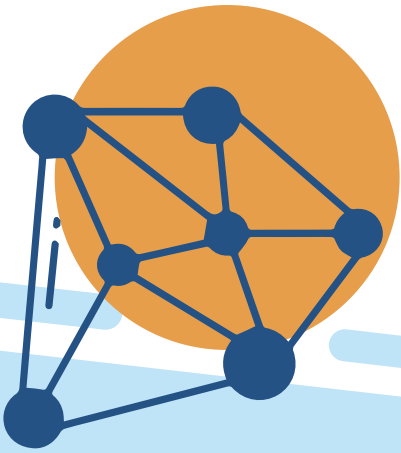
Ethereum Virtual Machine (EVM):

A special state machine maintained by thousands of connected computers running an Ethereum client. It defines the rules for computing a new valid state from block to block in the Ethereum blockchain.



State Transition Function:

A mathematical function of the EVM that, given an old valid state and a new set of valid transactions, produces a new valid output state. It ensures the continuous operation and evolution of the Ethereum blockchain.

State:

In Ethereum, the state refers to an enormous data structure called a modified Merkle Patricia Trie, which holds all accounts linked by hashes and reducible to a single root hash stored on the blockchain.

Transactions:

Instructions from accounts that are cryptographically signed. Transactions can result in message calls or contract creation, where a new contract account containing compiled smart contract bytecode is created.

EVM Instructions:

Commands executed by the Ethereum Virtual Machine during the processing of smart contracts. These instructions include standard stack operations like XOR, AND, ADD, SUB, as well as blockchain-specific operations like ADDRESS, BALANCE, BLOCKHASH, etc.

EVM Implementations:

Different software implementations of the Ethereum Virtual Machine in various programming languages. These implementations must adhere to the specification described in the Ethereum Yellow Paper and ensure compatibility with the Ethereum blockchain.

Stack Machine:

The execution model of the EVM, where operations are performed on a stack with a depth of 1024 items. Each item is a 256-bit word, chosen for compatibility with 256-bit cryptography. During execution, the EVM maintains transient memory and uses a Merkle Patricia storage trie for contracts.