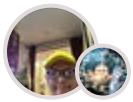
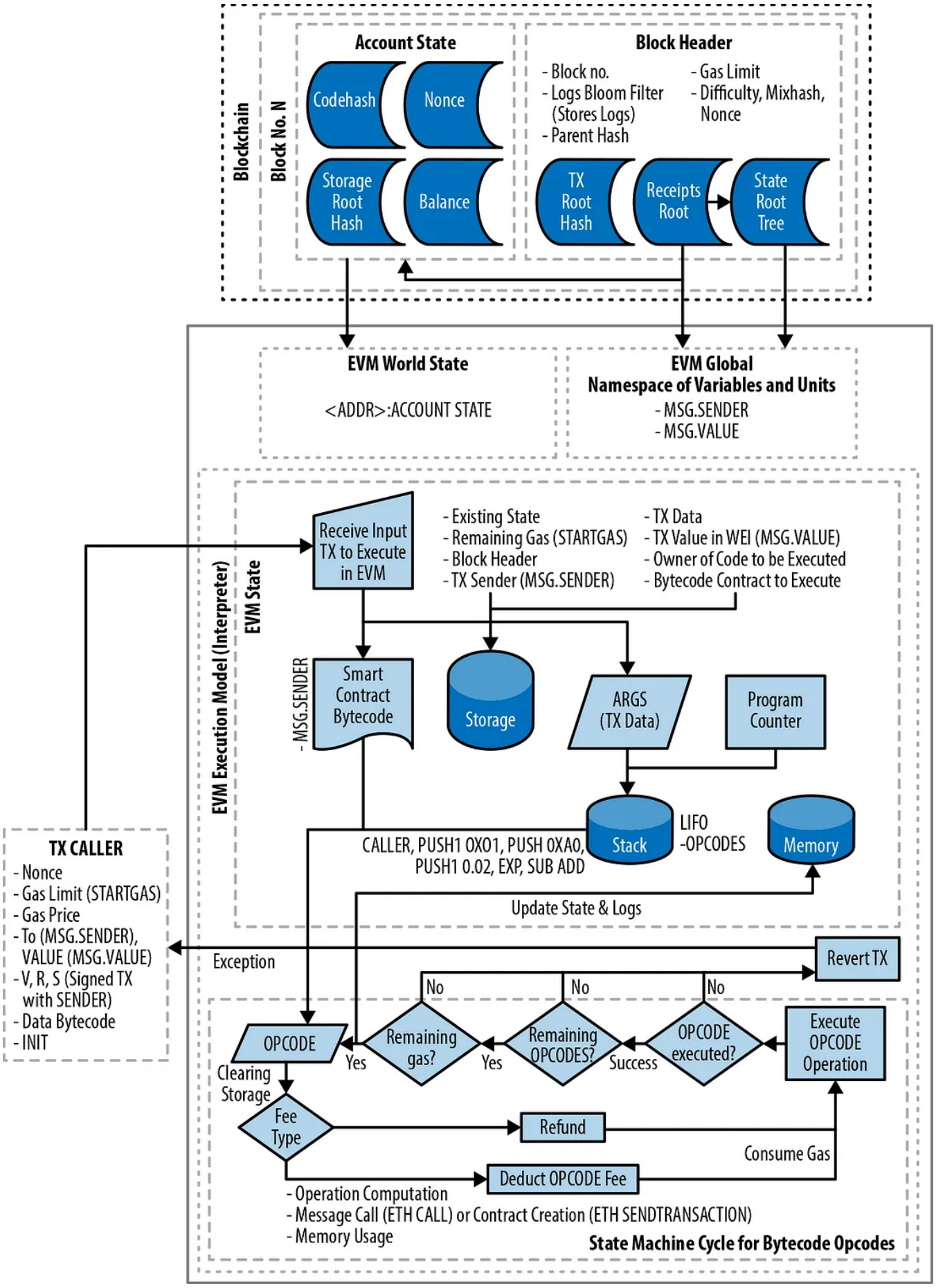
Understanding the Ethereum Virtual Machine (EVM)



Nova Novriansyah · Follow
Published in Novai-Blockchain 101
4 min read · May 7, 2024


Blockchain
Block No. N
Account State
Codehash
Nonce
Storage Root Hash
Balance
Block Header
- Block no.
- Logs Bloom Filter (Stores Logs)
- Parent Hash
- Gas Limit
- Difficulty, Mixhash, Nonce
TX Root Hash
Receipts Root
State Root Tree
EVM World State
<ADDR>:ACCOUNT STATE
EVM Global Namespace of Variables and Units
- MSG.SENDER
- MSG.VALUE
EVM Execution Model (Interpreter)
EVM State
Receive Input TX to Execute in EVM
- Existing State
- Remaining Gas (STARTGAS)
- Block Header
- TX Sender (MSG.SENDER)
- TX Data
- TX Value in WEI (MSG.VALUE)
- Owner of Code to be Executed
- Bytecode Contract to Execute
- MSG.SENDER
Smart Contract Bytecode
Storage
ARGS (TX Data)
Program Counter
Stack
LIFO
-OPCODES
Memory
CALLER, PUSH1 0X01, PUSH 0XA0, PUSH1 0.02, EXP, SUB ADD
Update State & Logs
TX CALLER
- Nonce
- Gas Limit (STARTGAS)
- Gas Price
- To (MSG.SENDER), VALUE (MSG.VALUE)
- V, R, S (Signed TX with SENDER)
- Data Bytecode
- INIT
Exception
Revert TX
No
No
No
OPCODE
Remaining gas?
Remaining OPCODES?
OPCODE executed?
Execute OPCODE Operation
Yes
Yes
Success
Clearing Storage
Fee Type
Refund
Consume Gas
Deduct OPCODE Fee
- Operation Computation
- Message Call (ETH CALL) or Contract Creation (ETH SENDTRANSACTION)
- Memory Usage
State Machine Cycle for Bytecode Opcodes

Ethereum Virtual Machine

The Ethereum Virtual Machine (EVM) is like the engine powering the Ethereum network. It's not a physical thing you can point to; rather, it exists as a concept maintained by thousands of computers connected to the Ethereum network.

What is the EVM?

The EVM is a special state machine where all Ethereum accounts and smart contracts live. It ensures the continuous, uninterrupted, and unchangeable operation of the Ethereum network. At any given point in time, Ethereum has only one 'canonical' state, and the EVM defines the rules for computing a new valid state from block to block.

Prerequisites

To understand the EVM, it's helpful to be familiar with basic computer science terms like bytes, memory, and stack, as well as concepts in cryptography and blockchain like hash functions and Merkle trees. Please read my other atricles related with this.

From Ledger to State Machine

While Bitcoin operates primarily as a distributed ledger, *Ethereum goes further by enabling smart contracts*. Instead of just recording transactions, Ethereum functions as a *distributed state machine*. This means it maintains a state that can change over time according to predefined rules, and it can execute arbitrary machine code.

While Ethereum has its own cryptocurrency (Ether), it also enables a much more powerful function: smart contract.

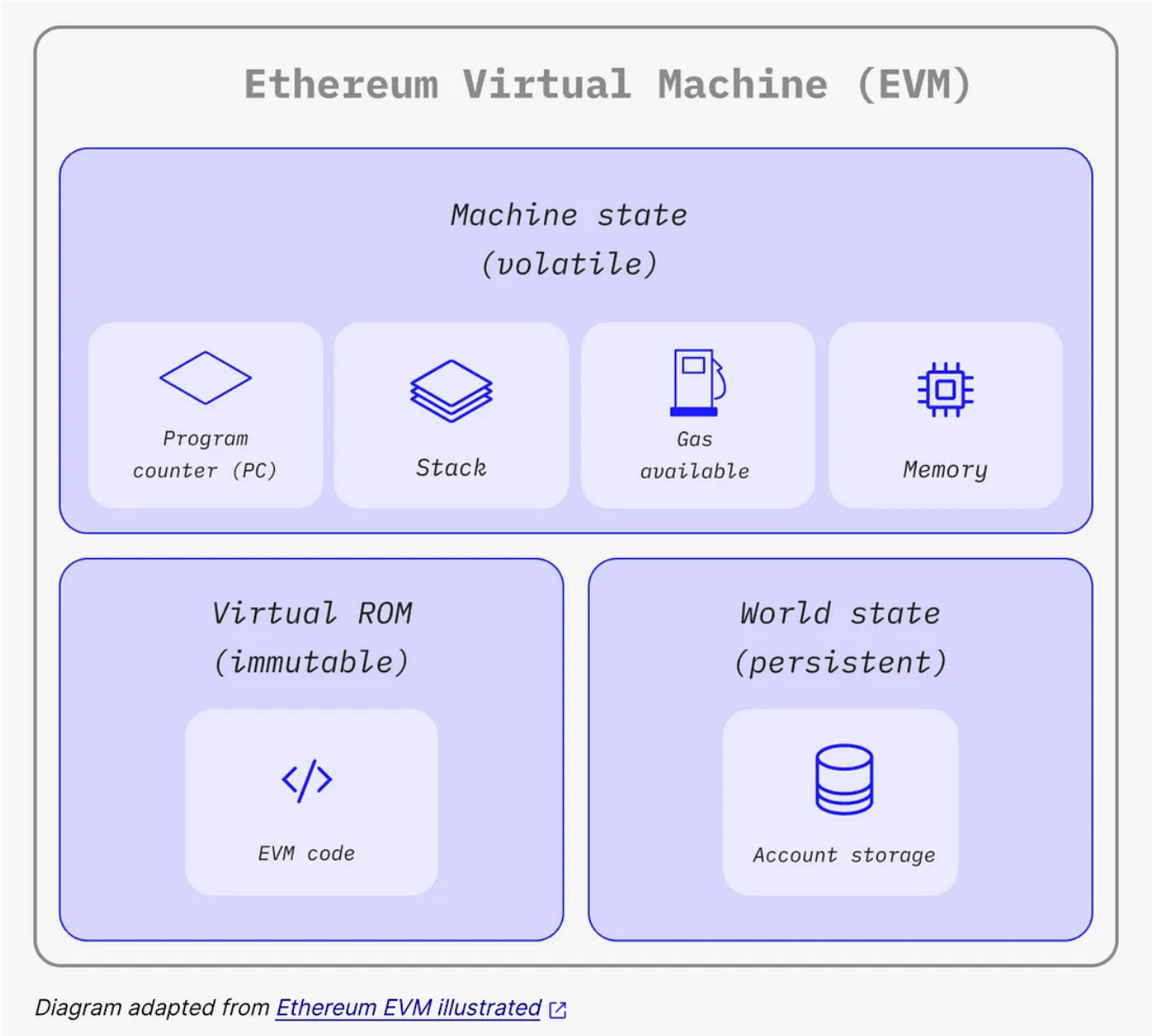


Diagram adapted from Ethereum EVM illustrated

The Ethereum State Transition Function

The EVM operates like a mathematical function: given an input, it produces a deterministic output. This function, known as the Ethereum state transition function, takes an old valid state and a set of transactions as input and produces a new valid output state.

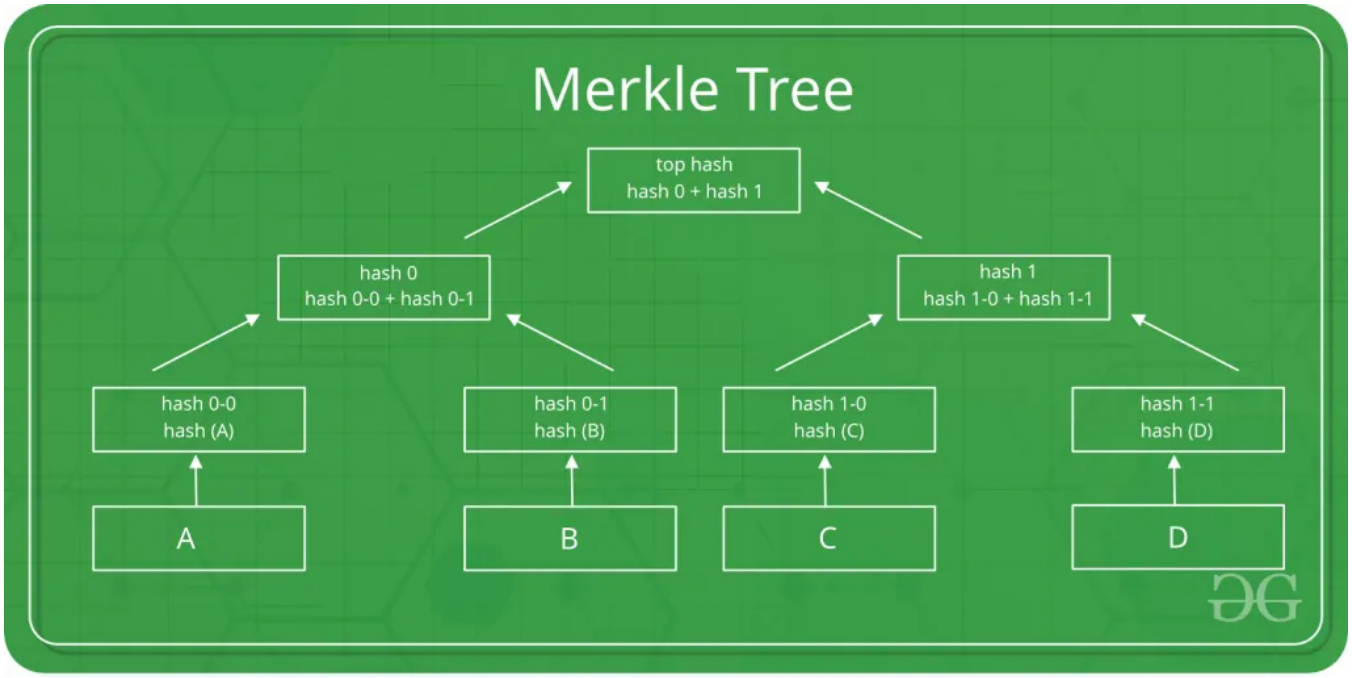
```
Y(S, T)= S
```

State and Transactions

In Ethereum, the state is a massive data structure called a modified Merkle Patricia Trie. It contains all accounts and balances, along with machine state information.

Transactions are cryptographically signed instructions from accounts. There are two types of transactions: those which result in message calls and those which result in contract creation.

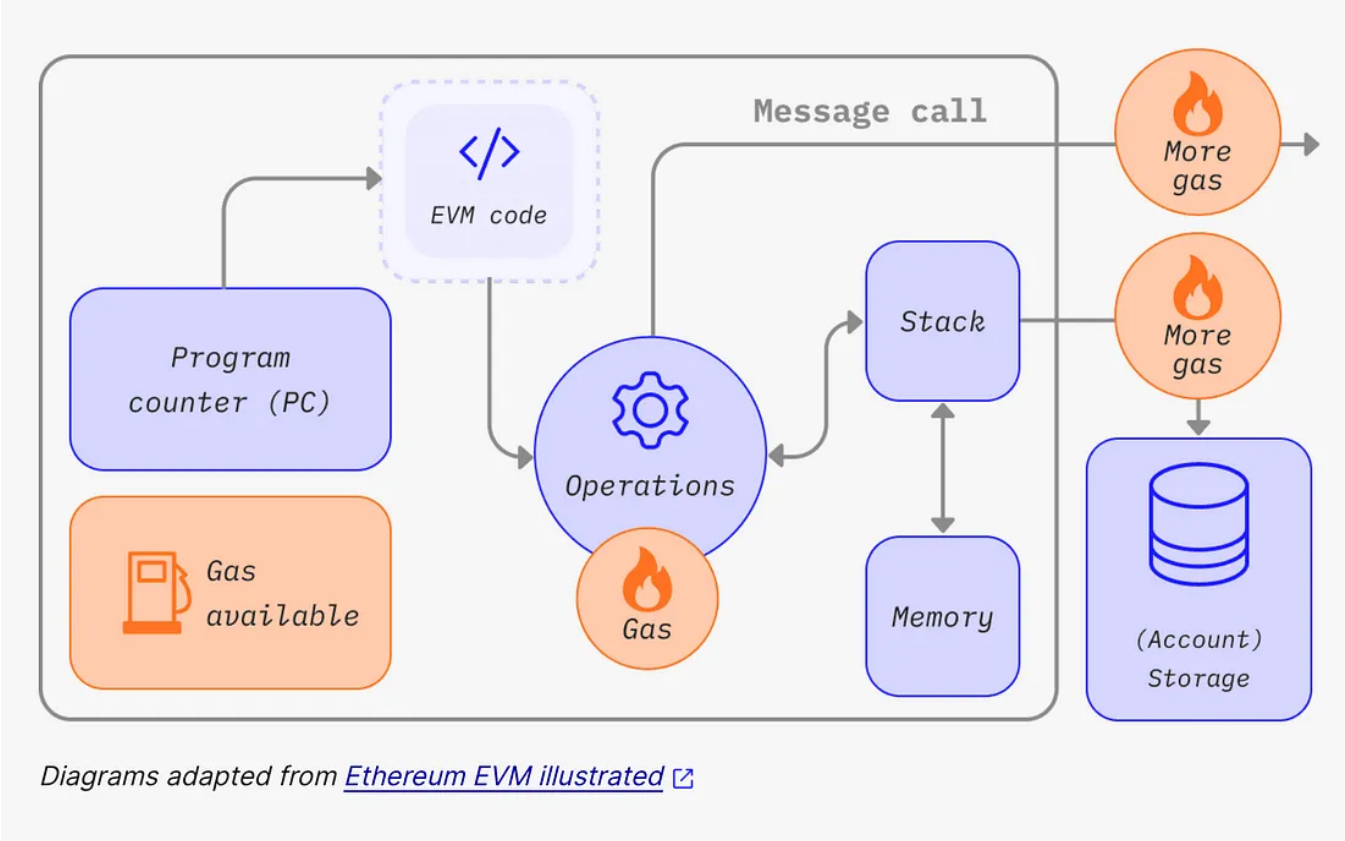
Contract creation results in the creation of a new contract account containing compiled bytecode. Whenever another account makes a message call to that contract, it executes its bytecode.



Merkle Tree

EVM Instructions

The EVM executes as a stack machine with a depth of 1024 items. Each item is a 256-bit word, chosen for compatibility with 256-bit cryptography. Contracts contain bytecode that executes as a series of EVM opcodes, performing operations like XOR, ADD, SUB, etc.



Diagrams adapted from Ethereum EVM illustrated

The EVM Instruction Set (Bytecode Operations)

The EVM instruction set offers most of the operations you might expect, including:

- Arithmetic and bitwise logic operations
- Execution context inquiries
- Stack, memory, and storage access
- Control flow operations
- Logging, calling, and other operators

In addition to the typical bytecode operations, the EVM also has access to account information (e.g., address and balance) and block information (e.g., block number and current gas price).

Arithmetic opcode instructions:

```
ADD         //Add the top two stack items
MUL         //Multiply the top two stack items
SUB         //Subtract the top two stack items
DIV         //Integer division
SDIV        //Signed integer division
MOD         //Modulo (remainder) operation
SMOD        //Signed modulo operation
ADDMOD      //Addition modulo any number
MULMOD      //Multiplication modulo any number
EXP         //Exponential operation
SIGNEXTEND //Extend the length of a two's complement signed integer
SHA3        //Compute the Keccak-256 hash of a block of memory
```

A complete list of opcodes and their corresponding gas cost can be found in [evm_opcodes].

EVM Implementations

The EVM specification is outlined in the Ethereum Yellowpaper. Over time, it has undergone revisions, and there are implementations in various programming languages including Python, C++, JavaScript, and Rust.

Understanding the EVM is crucial for grasping how Ethereum works behind the scenes. It's the backbone of the Ethereum network, enabling its decentralized and programmable nature.

Compiling Solidity to EVM Bytecode

To transform a Solidity source file into EVM bytecode, various methods are available. In [intro_chapter], we utilized the online Remix compiler. Here, we'll employ the solc executable through the command line. To explore available options, use:

```
$ solc --help
```

Generating the raw opcode stream of a Solidity source file is straightforward using the — opcodes command-line option. While this opcode stream might lack some details (the — asm option provides comprehensive information), it suffices for our purposes. For instance, to compile a Solidity file named Example.sol and direct the opcode output to a directory named BytecodeDir, use:

```
$ solc -o BytecodeDir --opcodes Example.sol
```

or:

```
$ solc -o BytecodeDir --asm Example.sol
```

To obtain the bytecode binary for our example program, execute:

```
$ solc -o BytecodeDir --bin Example.sol
```

The resulting opcode files depend on the contracts within the Solidity source file. For our simple Example.sol file containing only one contract named “example”:

```
pragma solidity ^0.4.19;

contract example {

  address contractOwner;

  function example() {
    contractOwner = msg.sender;
  }
}
```

This contract maintains a single persistent state variable, storing the address of the last account to execute the contract.

Checking the BytecodeDir directory reveals the opcode file example.opcode, housing the EVM opcode instructions of the example contract. Opening this file in a text editor reveals the opcode stream.

Ethereum Blockchain

Follow

Published in Novai-Blockchain 101

1 Follower · Last published Jun 2, 2024

Welcome to our blockchain channel, where we unravel the mysteries of decentralized technology. Delve into the concepts of public and private blockchains, exploring their unique features, applications, and potential impact on various industries. Whether you're a blockchain novice

Follow

Written by Nova Novriansyah

109 Followers · 34 Following

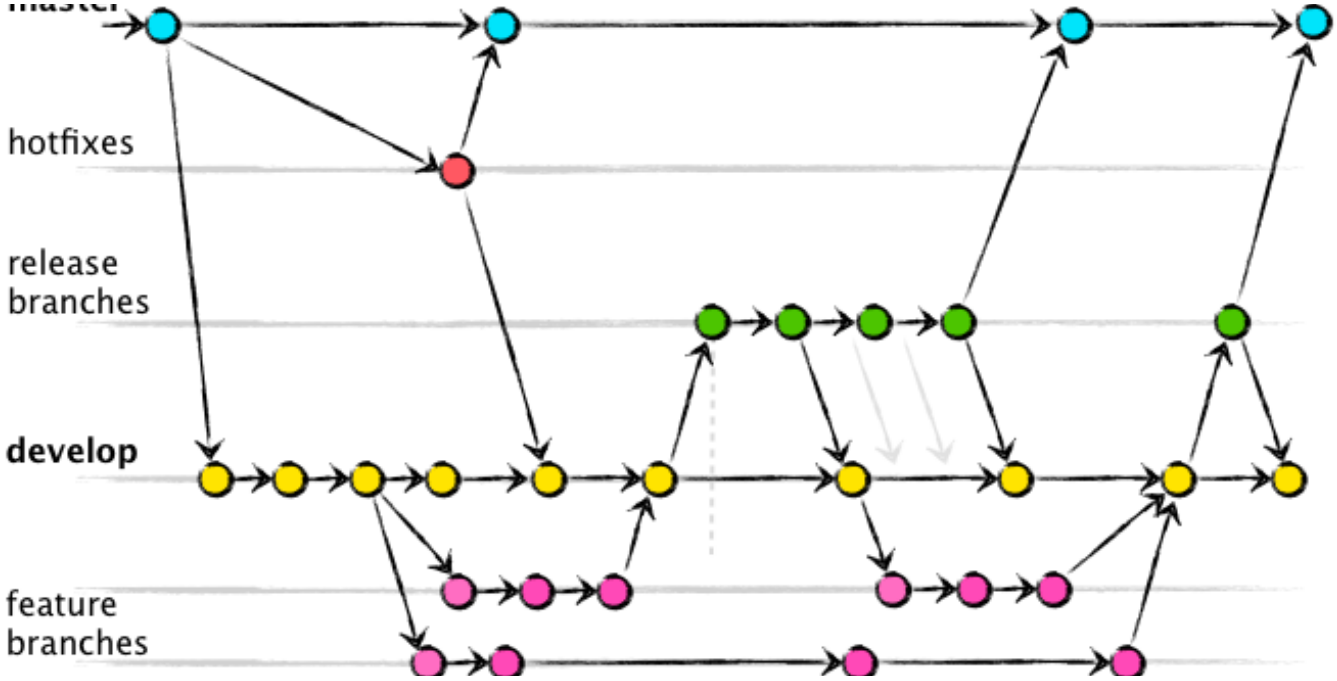
C|CISO, CEH, CC, CVA,CertBlockchainPractitioner, Google Machine Learning , Tensorflow, Unity Cert, Arduino Cert, AWS Arch Cert. CTO, IT leaders. Platform owners

No responses yet

What are your thoughts?

Respond

More from Nova Novriansyah and Novai-Blockchain 101

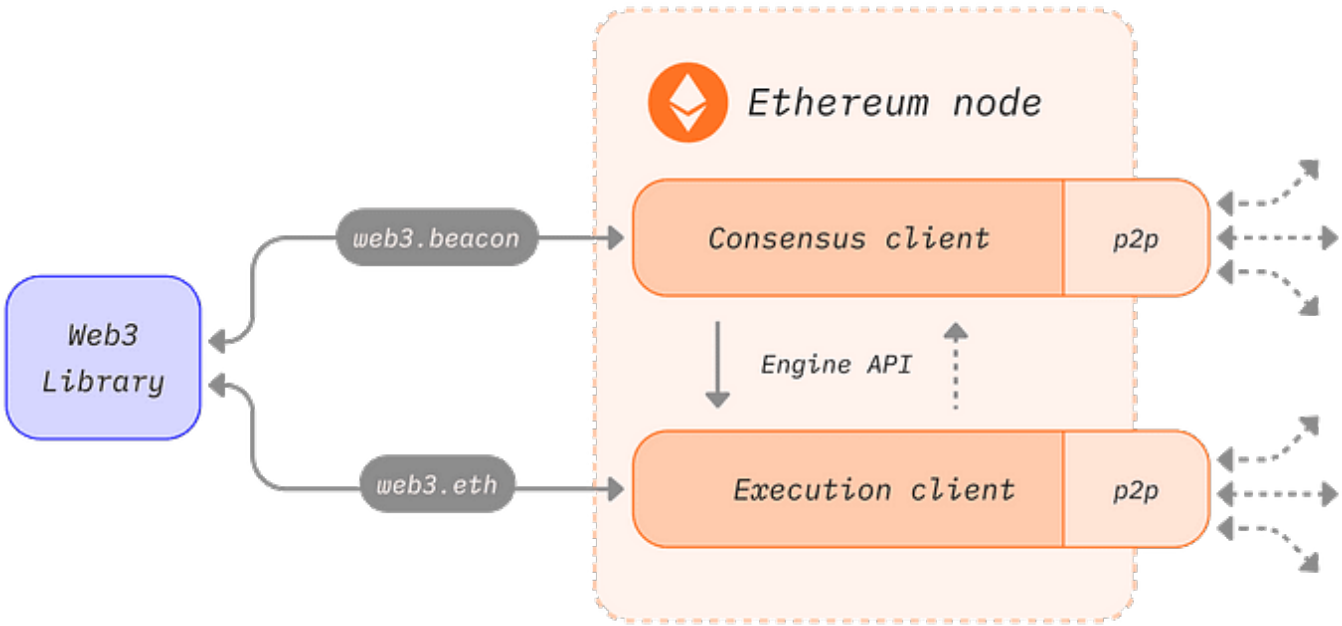


In NovAI- Agile & DevOPS 101 by Nova Novriansyah

Top 4 Branching Strategies and Their Comparison: A Guide with Recommendations

Branching strategies are critical in version control, helping teams manage and organize code changes efficiently. Choosing the right...

Aug 15 14

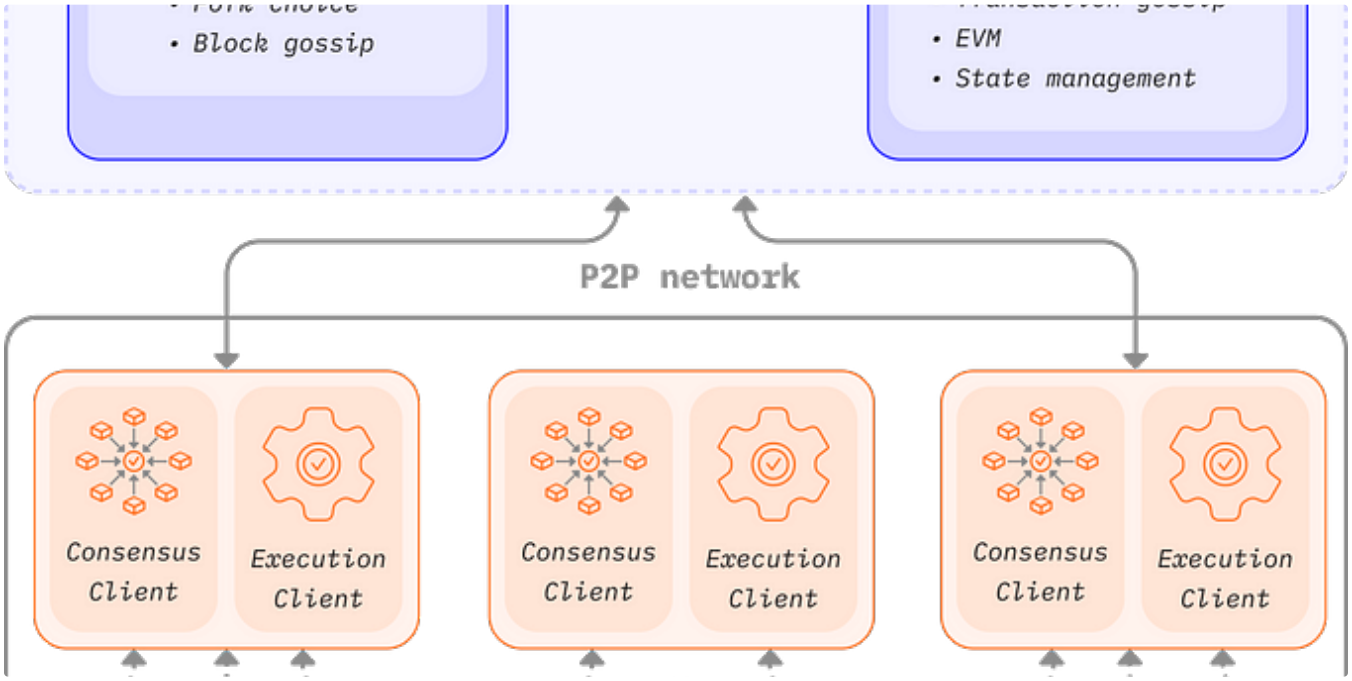


In Novai-Blockchain 101 by Nova Novriansyah

Understanding Nodes and Clients in Ethereum

In the realm of Ethereum, nodes and clients play crucial roles in maintaining the network's integrity and facilitating transactions. Let's...

May 7 2



In Novai-Blockchain 101 by Nova Novriansyah

Understanding Ethereum Node Architecture

Ethereum, the groundbreaking blockchain platform, operates through a complex network of nodes. These nodes play crucial roles in executing...

May 7 2

In NovAI Cloud Computing — GCP by Nova Novriansyah

How to Install Google Cloud CLI (Command-Line Interface) on Mac, Windows, and Linux

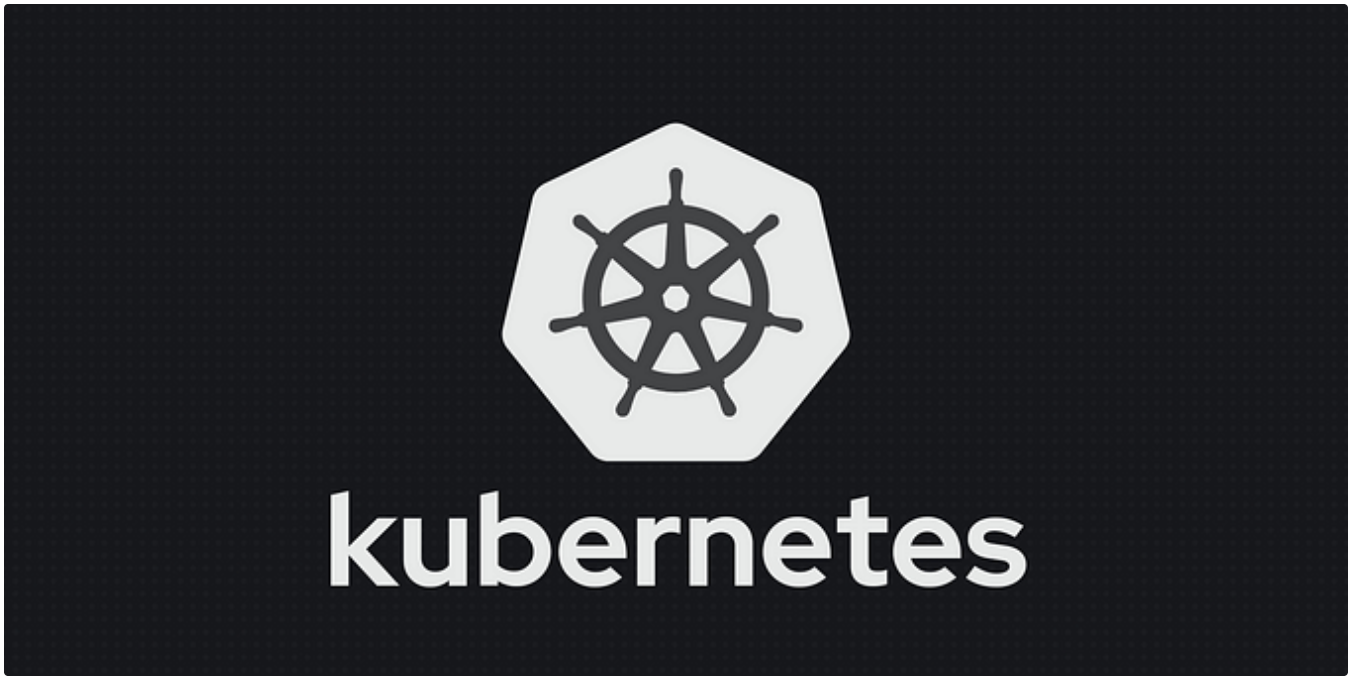
Google Cloud CLI, known as gcloud, is an essential tool for managing Google Cloud Platform (GCP) resources from the command line...

Jun 21 1 1

See all from Nova Novriansyah

See all from Novai-Blockchain 101

Recommended from Medium



In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

Nov 19 3.7K 111

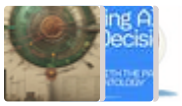
Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free

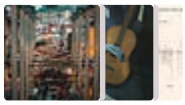
Oct 26 6.2K 89

Lists



data science and AI

40 stories · 296 saves

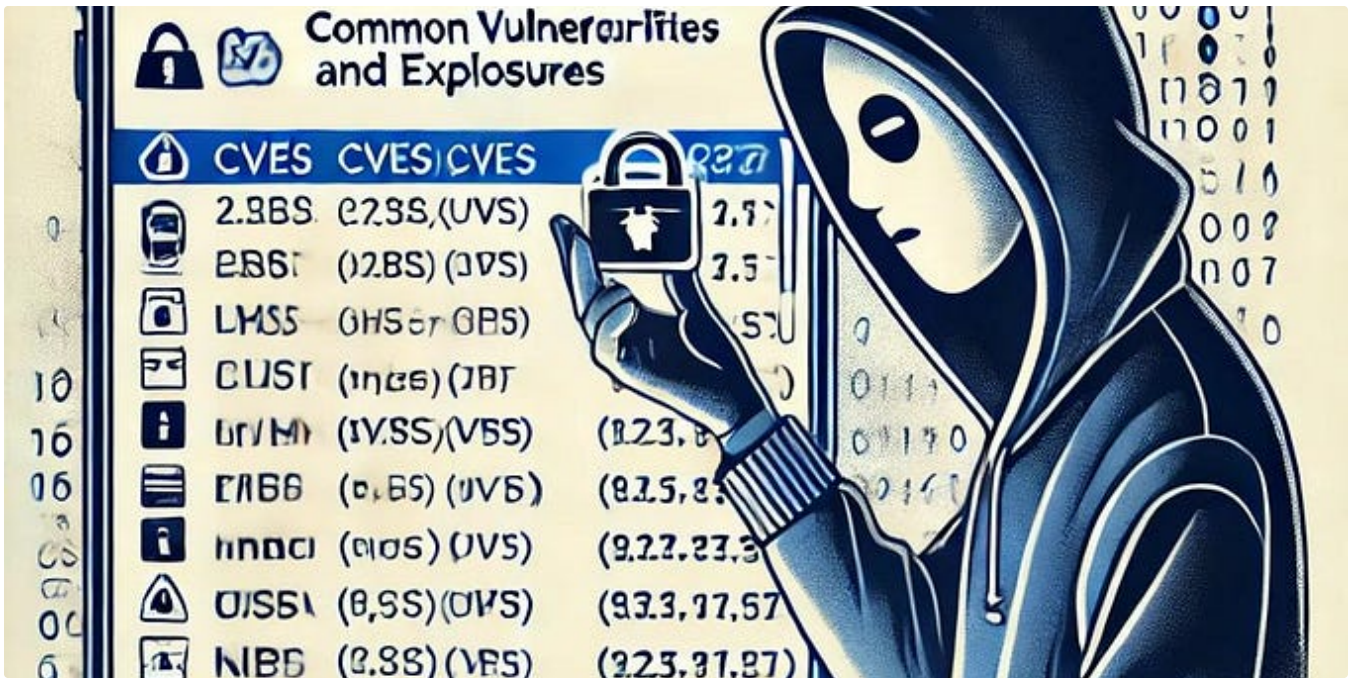


My Kind Of Medium (All-Time Faves)

102 stories · 598 saves

MODERN MARKETING

199 stories · 948 saves



Jonathan Mondaut

How ChatGPT Turned Me into a Hacker

Jessica Stillman

Jeff Bezos Says the 1-Hour Rule Makes Him Smarter. New Neuroscience Says He's Right

Jeff Bezos's morning routine has long included the one-hour rule. New neuroscience says yours probably should too.

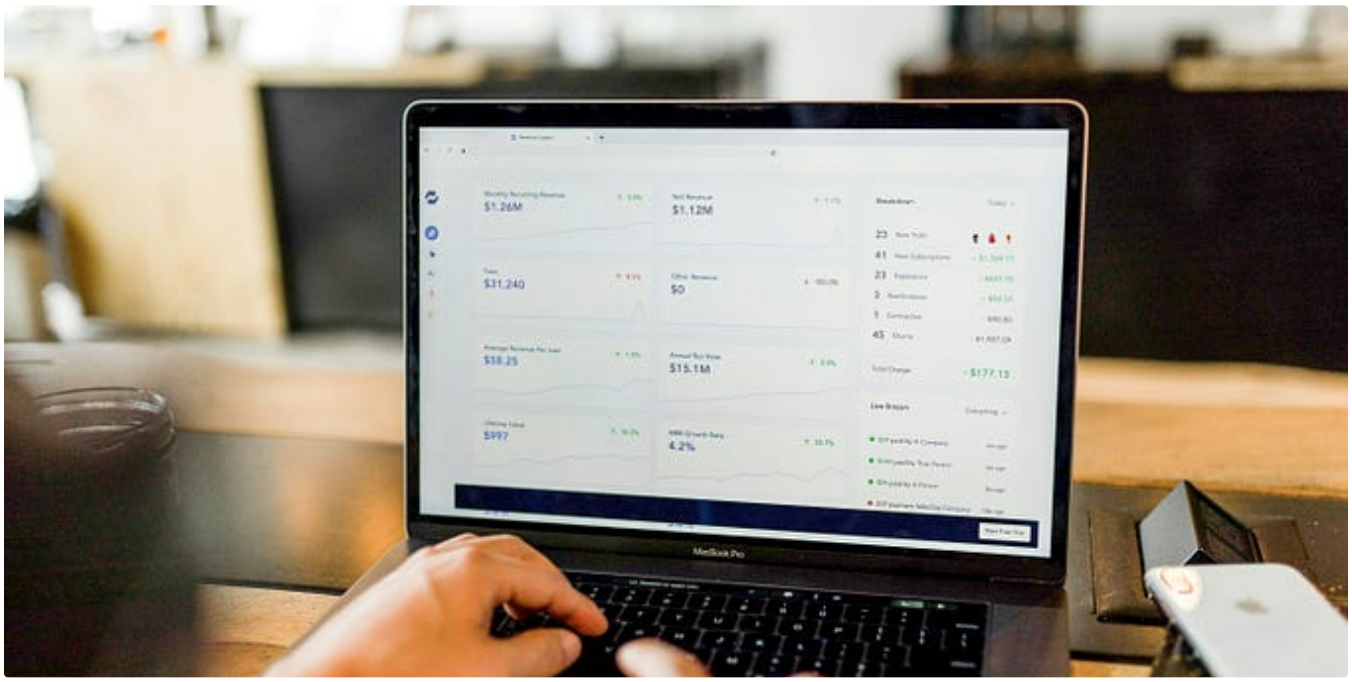
Oct 30 13.9K 325

In Web3 Labs by George Ţebrean

Hyperledger Web3j: Truly decode support for dynamic Solidity structs

In Solidity, dynamic structs are complex data types that can store multiple elements of varying sizes, such as arrays, mappings, or other...

Aug 16 47



Taofikat Titilayo Adeleke

Ethereum Layer Two Solutions: Scaling Blockchain

Ever since its start, Ethereum has become a key player in the blockchain world. It's the second most valuable platform, next to Bitcoin...

Jul 18

See more recommendations