SRI VENKATESWARAA
COLLEGE OF TECHNOLOGY

SVCT
BLOCKCHAIN
CLUB

GITATHON

Assignment II
Ethereum Blockchain and smart contracts

By
S. HEMACHANDRAN
B.E.CSE(CYBERSECURITY)
SRI VENKATESHWARAA COLLEGE OF TECHNOLOGY
SVCTBCC-C-1012

ETHEREUM
OVERVIEW

INTRODUCTION

Ethereum is like a digital world where people can create and use smart contracts and apps. Its currency, ether, is popular and ranks second to bitcoin. It was started in 2013 by Vitalik Buterin and others, and went live in 2015. Ethereum lets people build permanent apps that no one controls. These apps include financial services called DeFi, where you can borrow or lend cryptocurrency without traditional banks. Ethereum also allows for creating and trading unique digital items called NFTs. Many other cryptocurrencies use Ethereum's technology for crowdfunding.

ETHEREUM PLATFORM

WHAT IS ETHER ?

Ether (ETH) is Ethereum's main cryptocurrency. It's used to fuel transactions and computations on the network. Users pay ETH to others to execute their code requests. Ethereum has two types of accounts: Externally Owned Accounts (EOA), which are controlled by private keys and can send transactions, and Contract Accounts, which have code that runs when they receive transactions from EOAs.

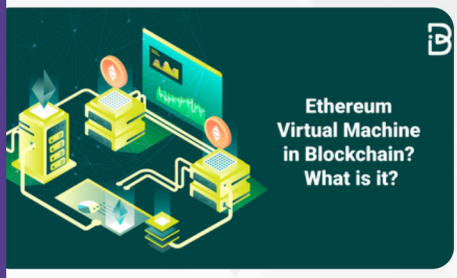
BEST ETHEREUM PLATFORM

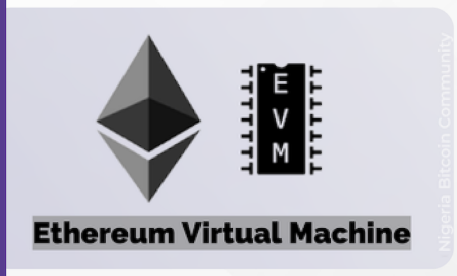
For trading Ethereum, eToro stands out as a top choice. It offers transparent fees, copy trading, and smart portfolios. Ethereum, known for its cryptocurrency ether (ETH), operates on a blockchain platform facilitating secure digital ledgers. While similar to Bitcoin, Ethereum has distinct long-term goals and transitioned to proof of stake in September 2022. It serves as the basis for various blockchain-driven innovations.

EVM

ETHEREUM VIRTUAL MACHINE

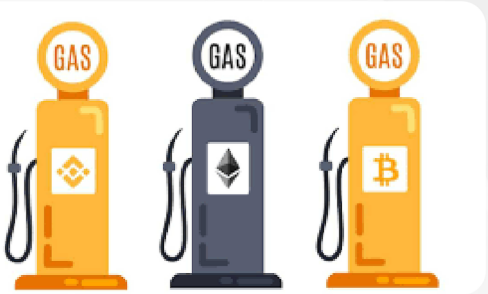
The Ethereum Virtual Machine (EVM) is crucial for Ethereum, managing the blockchain's state and enabling smart contracts. It's found within client software like Geth or Nethermind, necessary for running a node on Ethereum. The EVM sets rules for computing the network's state, executing smart contracts, processing transactions, and updating balances. Structurally, it comprises a control unit and a balloting unit connected by a cable.

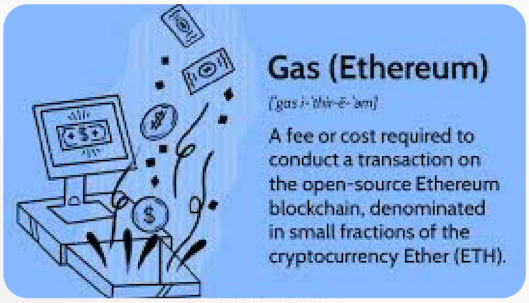




GAS SMART CONTRACT

Gas in Ethereum is like a fee you pay to do things on the network. It's needed to run transactions or smart contracts. Smart contracts can control how much gas they use when they run, helping users manage their costs on the Ethereum network.





Gas (Ethereum)

['gas i-'thir-ē-'əm]

A fee or cost required to conduct a transaction on the open-source Ethereum blockchain, denominated in small fractions of the cryptocurrency Ether (ETH).

GAS IN ETHEREUM WORKS IN SEVERAL WAYS

1. Gas Limit: Developers set a maximum gas limit for smart contracts. If the execution exceeds this limit, it fails.
2. Gas Price: Users choose the price they're willing to pay for execution. Higher prices mean faster transactions.
3. Gas Cost: This depends on how complex the smart contract is. More complex operations need more gas.
4. Gas Refund: Unused gas gets refunded at the end of execution.
5. Calculation: Total cost = Gas Used * Gas Price.
6. Gas Fees: Paid by users to incentivize miners for transaction processing.
7. Out-of-Gas Errors: If an operation exceeds the gas limit, it fails with an "out-of-gas" error.
8. Price Volatility: Gas prices can vary based on network congestion and demand, often increasing during high activity times.

WHAT IS
ETHEREUM GAS ?

Ethereum gas is what people pay to do things like transactions or use smart contracts on Ethereum. The total cost equals the gas used multiplied by the gas price.

STRUCTURE OF ETHEREUM IN SMART CONTRACTS

The structure of Ethereum smart contracts includes:

1. Contract Code: Written in languages like Solidity, it defines how the contract works.
2. State Variables: Hold persistent data for the contract.
3. Functions: Define actions the contract can take, like reading data or interacting with other contracts.
4. Events: Notify about important contract activities.
5. Modifiers: Modify how functions behave, often for access control or input validation.
6. Visibility Specifiers: Determine how functions and variables can be accessed.
7. Fallback Function: Handles unexpected transactions.
8. Constructor: Initializes the contract when deployed.
9. Immutable: Once deployed, contracts can't be changed.
10. Interoperability: Contracts can interact with each other and other platforms.
11. EVM Compatibility: Smart contracts run on the Ethereum Virtual Machine, ensuring consistency across the network.

SOLIDITY FEATURES

MODIFIERS

Modifiers in smart contracts help enforce access control, ensuring only authorized users can execute specific functions, enhancing security. They also promote code reuse by allowing common logic to be applied to multiple functions, improving readability and maintenance. Additionally, modifiers aid in input validation, verifying user-provided data to prevent unexpected behavior or vulnerabilities. They manage the contract's state before and after function execution, maintaining consistency. Modifiers also optimize gas usage, reducing transaction costs and improving scalability by eliminating redundant code.

Events, declared using the event keyword, are logged on the blockchain when triggered, providing a permanent record of contract activity for external applications.

ADVANCED SOLIDITY

1. Advanced Solidity: Involves delving deeper into Solidity, exploring concepts like inheritance, libraries, interfaces, and abstract contracts. Inheritance allows contracts to inherit properties, while libraries provide reusable code chunks, improving modularity. Interfaces define contract structures for interoperability, and abstract contracts set blueprints for consistent behavior.

2. State Variables and Constants: Solidity allows defining state variables representing a contract's state and constants whose values remain unchanged after initialization, often used for immutable parameters.

3. Payable and Non-payable Functions: Solidity functions can be payable, receiving Ether, or non-payable, not involving fund transfers.

4. Building Complex Smart Contracts: Proficient Solidity developers create complex contracts, understanding design patterns, security considerations, and gas optimization.

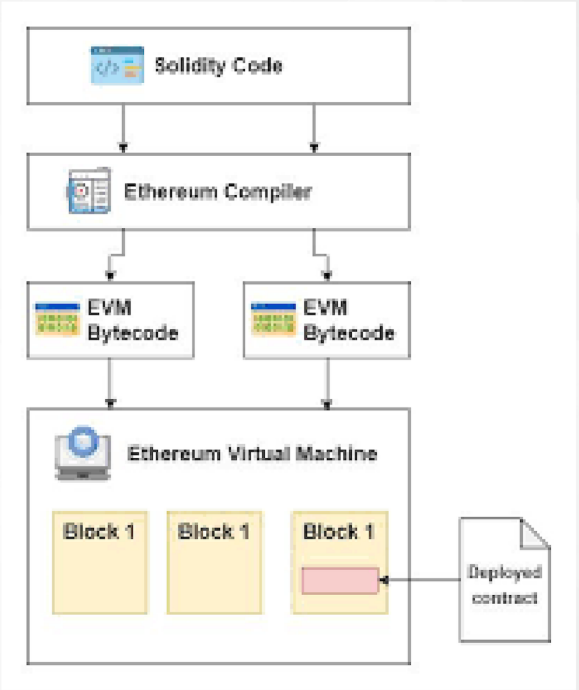
5. Best Security Practices: Security is crucial, requiring measures against common vulnerabilities like reentrancy attacks, integer overflows, and unauthorized access.

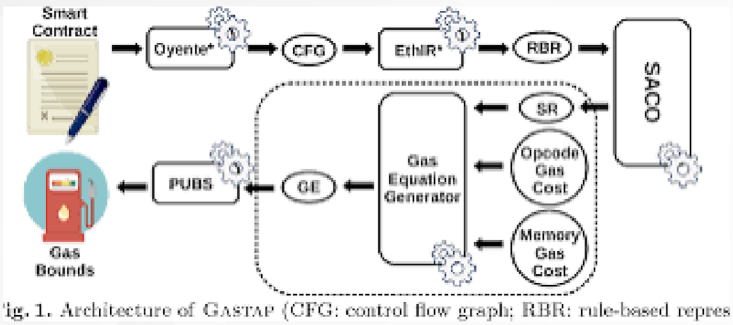
6. Introduction to Truffle Suite: A development framework for Ethereum, Truffle Suite simplifies smart contract development, testing, and deployment with tools like Truffle, Ganache, and Drizzle. Mastering these topics enables developers to build sophisticated, secure, and efficient contracts on Ethereum using Solidity and Truffle Suite.

GAS SMART CONTRACT IN
BLOCKCHAIN

Gas fees on the Ethereum network are determined by three variables:

ETHEREUM VIRTUAL MACHINE





ig. 1. Architecture of GASTAP (CFG: control flow graph; RBR: rule-based repres

GAS SMART CONTRACT

THANK YOU