# ETHEREUM BLOCKCHAIN AND SMART CONTRACTS

NAME : RAJA RAJESWARI.S
DEPARTMENT: EEE
CLUB ID:SVCTBCC-C-1024

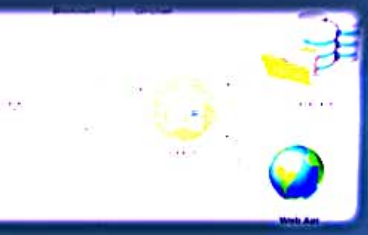
## INTRODUCTION TO ETHEREUM :

**In block chain, ether is the cryptocurrency used on the ETHEREUM platform. It's used to compensate participants who perform computations and validate transactions. It also serves as a means of payment for transaction fees and services on the network**


## ETHER:

In block chain, ether is the cryptocurrency used on the ETHEREUM platform. It's used to compensate participants who perform computations and validate transactions. It also serves as a means of payment for transaction fees and services on the network

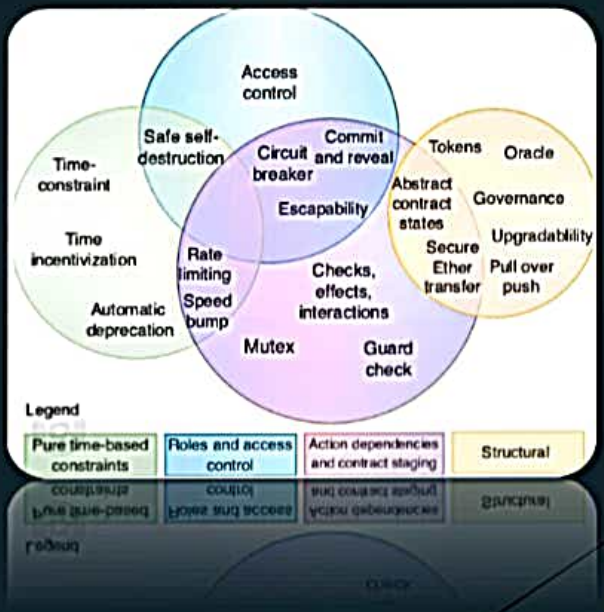
## ETHEREUM VIRTUAL MACHINE:



The Ethereum Virtual Machine (EVM) is like a computer that runs on the Ethereum blockchain. It's where smart contracts, which are small programs, run. These programs can do things like manage money or store information without needing a central authority. So, the EVM is what makes Ethereum work as a platform for all sorts of applications.

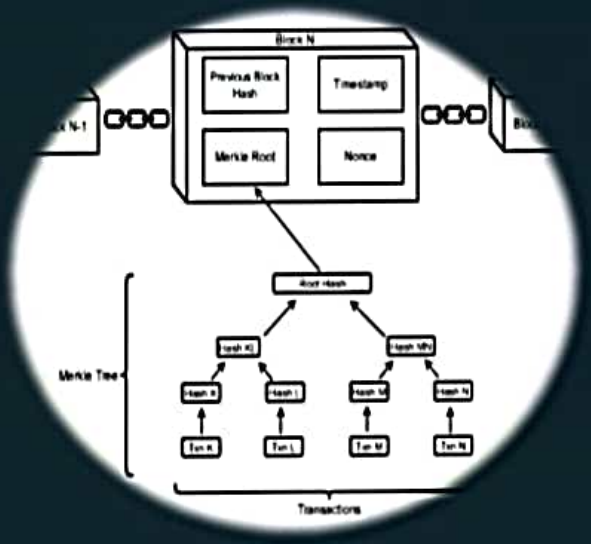
# GAS –SMART CONTRACTS IN BLOCKCHAIN:

"Gas" is like a fee you pay for doing things, like running smart contracts. Smart contracts are like automatic agreements written in code. Gas-smart contracts make sure these agreements work efficiently and fairly by managing how much resources they use on the block chain network.




# STRUCTURE OF ETHEREUM :

It like a digital ledger that records transactions. Each transaction is like a note in the ledger, and they're all linked together in a chain.



## SOLIDITY FEATURES:

Solidity is a programming language used for writing smart contracts on the ETHEREUM block chain. Some of its key features include:

## MODIFIERS

Modifiers in block chain are like rules that control how certain actions can be performed within smart contracts. They're used to specify who can do what. For example, "Only Owner" means only the contract owner can perform a certain action, while "Only Admin" limits it to administrators. They help ensure security and proper functioning of block chain applications

# ACCESS CONTROL

Access control in block chain refers to the mechanisms put in place to regulate who can interact with the block chain network and what actions they can perform. Smart contracts, cryptographic keys, and permissioned block chains are common tools used to manage access control.

# INPUT VALIDATION

Input validation in block Chain is like a bouncer at a club checking IDs before letting people in. It's about making sure that the data or transactions are legit and follow the rules before they're added to the block chain. This helps keep everything secure and running smoothly.

## STATE MANAGEMENT

State management in Block Chain is like keeping track of who owns what and what's happening in the network. It's about updating balances, smart contracts, and other important data in a way that everyone agrees on. Different Block Chains use different methods to do this, like tracking individual transactions (Bitcoin) or account balances ( ETHEREUM), and everyone in the network follows the same rules to keep everything in sync.

## GAS OPTIMIZATION

Gas optimization in block chain is like finding the most efficient way to do things on a computer. It's about using as little energy as possible while still getting the job done quickly and correctly. This helps make transactions faster and cheaper on the block chain, which is important for making the whole system work better.

# EVENTS:

## EVENT DECLARATION:

An event declaration refers to the definition of an event within a smart contract. Events are used to communicate information from the smart contract to external applications or other parts of the block chain. They're typically triggered by specific actions within the contract and provide a way for external entities to react to those actions. Event declarations include the event's name, parameters, and any relevant information that needs to be communicated when the event is emitted

# LOGGING EVENTS:

Logging events to the Block chain involves recording specific data or events onto a block chain network, creating an immutable and transparent record. This process typically involves writing transactions to the block chain, which are then validated and added to blocks by network participants through consensus mechanisms like proof of work or proof of stake. Once recorded, the data or events cannot be altered or deleted, ensuring their integrity and authenticity. This capability has applications across various industries, including finance, supply chain, healthcare, and more, where trust, transparency, and security are paramount. Using events effectively for emitting logs that can be monitored by external systems or UIs.

## ADVANCED SOLIDITY:

Advanced Solidity refers to the use of more complex features and techniques within the Solidity programming language for ETHEREUM smart contracts.


## SMART CONTRACTS DESIGN PATTERNS:

Implementing design patterns like Factory, Proxy, or Singleton to optimize gas usage or improve contract upgradability.

ORACLE INTEGRATING

Integrating external data sources or oracles into smart contracts to enable interaction with off-chain data

## UPGRADEABILITY

Designing contracts in a way that allows for upgradability while ensuring data and function integrity.

LIBRARIES AND INTERFACING:

Leveraging external libraries and interfaces to modularize code and reduce redundancy.

# CONCLUSION

Mastering these concepts requires a deep understanding of Solidity, ETHEREUM, and block chain development in general, as well as an awareness of the evolving best practices and security considerations in the space


THANK YOU ALL!