

L13: Ethereum Architecture

Module B: Ethereum & Smart Contracts

Blockchain & Cryptocurrency Course

December 2025

By the end of this lesson, you will be able to:

- Explain the Ethereum Virtual Machine (EVM) architecture
- Distinguish between Externally Owned Accounts (EOA) and Contract Accounts
- Describe Ethereum as a state machine and understand the world state concept
- Explain the Merkle Patricia Trie data structure
- Compare Ethereum's account model to Bitcoin's UTXO model

What is Ethereum?

Key Characteristics:

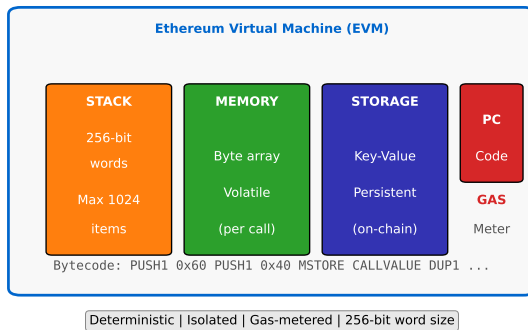
- Decentralized computation platform
- Turing-complete blockchain
- Smart contract support
- Native cryptocurrency: Ether (ETH)
- Launched July 30, 2015

Beyond Bitcoin:

- Bitcoin: Digital gold, value transfer
- Ethereum: World computer, programmable logic
- Enables decentralized applications (dApps)
- Foundation for DeFi, NFTs, DAOs

Ethereum Virtual Machine (EVM)

EVM: Stack-Based Virtual Machine



The EVM is a quasi-Turing complete, stack-based virtual machine with gas metering.

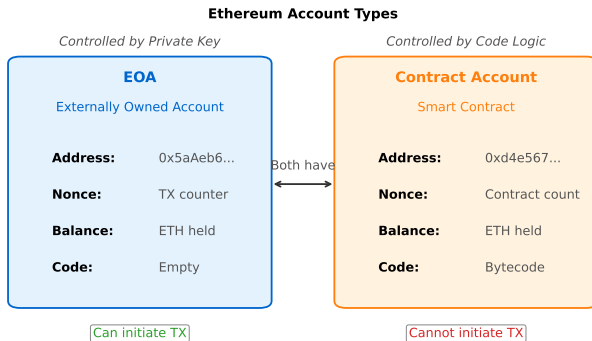
The EVM is a quasi-Turing complete state machine:

- Stack-based virtual machine (256-bit word size)
- Executes bytecode compiled from Solidity, Vyper
- Deterministic: same input always produces same output
- Gas metering prevents infinite loops
- Isolated: no network, filesystem, or process access

EVM Components:

- **Stack:** LIFO, max 1024 items, 256-bit each
- **Memory:** Byte array, volatile (per call), linear cost
- **Storage:** Key-value, persistent (on-chain), expensive
- **Program Counter:** Current bytecode position
- **Gas Counter:** Remaining computation budget

Account Types: EOA vs Contract



EOAs are controlled by private keys; contracts are controlled by code.

Every account (EOA or Contract) has four fields:

❶ **Nonce:**

- EOA: Counter of transactions sent
- Contract: Counter of contracts created
- Prevents replay attacks

❷ **Balance:**

- Amount of Wei (10^{-18} ETH) owned
- 1 ETH = 1,000,000,000,000,000,000 Wei

❸ **StorageRoot:**

- Hash of account's storage trie root
- Empty for EOAs

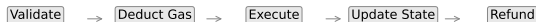
❹ **CodeHash:**

- Hash of EVM bytecode
- Empty for EOAs, immutable for contracts

Ethereum: Deterministic State Machine

State Transition Function

$$S_{t+1} = Y(S_t, T)$$



Transaction Execution Flow

State transitions are deterministic: given S_t and T , S_{t+1} is uniquely determined.

The World State is a mapping between addresses and account states:

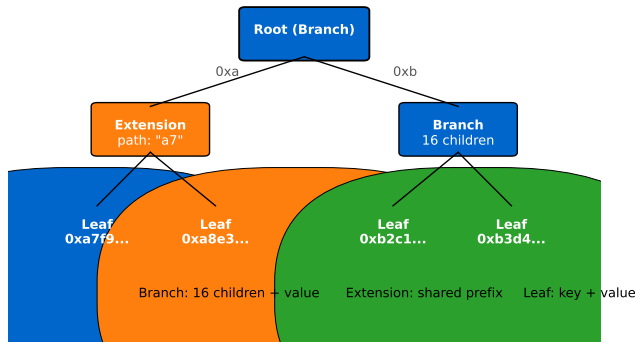
- Maps 160-bit addresses to account states
- Stored as a Merkle Patricia Trie
- Root hash included in every block header
- Allows efficient verification of account state
- Enables light clients to verify data without full state

State Root:

- 256-bit hash representing entire world state
- Changes with every block
- Uniquely identifies state at a specific block height

Merkle Patricia Trie (MPT)

Merkle Patricia Trie: Node Types



MPT combines Merkle verification, Patricia path compression, and radix optimization.

Combines three data structures:

- 1 **Merkle Tree:** Cryptographic verification via hashes
- 2 **Patricia Trie:** Efficient key-value storage with shared prefixes
- 3 **Radix Trie:** Optimized path compression

Key Properties:

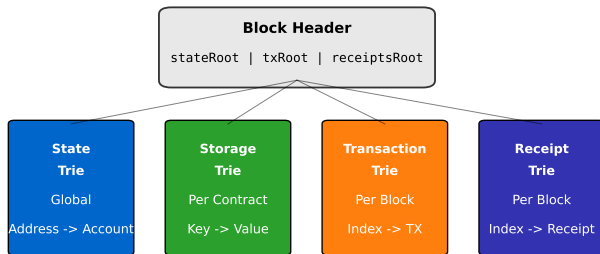
- Deterministic: Same key-value pairs \rightarrow same root hash
- Efficient verification: $O(\log n)$ proof size
- Efficient updates: Only modified paths need rehashing

Node Types:

- **Branch:** 16 children (hex 0-F) + optional value
- **Extension:** Shared path prefix + next node pointer
- **Leaf:** Remaining key path + value

Four MPT Tries in Ethereum

Ethereum Block: Four Merkle Patricia Tries



Merkle proof: $O(\log n)$

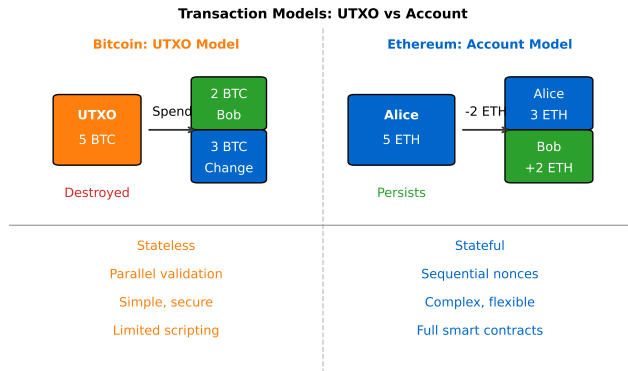
Deterministic root hash

Efficient updates

Root hashes in block header enable light client verification

Block header contains root hashes of all four tries for verification.

Ethereum vs Bitcoin: Account Model vs UTXO



Account model enables smart contracts but requires nonce tracking.

Advantages:

- ① **Simplicity:** Intuitive balance model, easier wallets
- ② **Space Efficiency:** No UTXO tracking overhead
- ③ **Fungibility:** All Ether equivalent, no dust
- ④ **Smart Contracts:** Natural fit for persistent storage

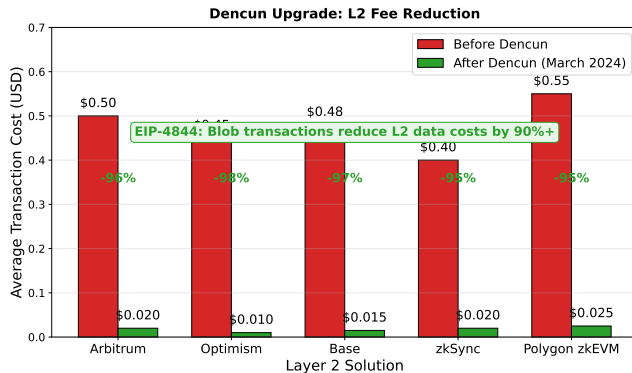
Challenges:

- ① **Replay Prevention:** Requires nonce tracking
- ② **State Growth:** All accounts stored indefinitely
- ③ **Privacy:** All activity linked to single address
- ④ **Parallelization:** Sequential nonces limit parallel validation

Ethereum block header contains:

- **parentHash:** Hash of parent block
- **beneficiary:** Address receiving block reward
- **stateRoot:** Root hash of state trie
- **transactionsRoot:** Root hash of transaction trie
- **receiptsRoot:** Root hash of receipt trie
- **logsBloom:** Bloom filter for efficient log lookup
- **number:** Block height
- **gasLimit / gasUsed:** Gas constraints
- **timestamp:** Unix timestamp
- **extraData:** Arbitrary data (max 32 bytes)

2024 Milestone: Dencun Upgrade



EIP-4844 blob transactions reduce L2 data costs by 90%+.

Major Network Upgrade (March 13, 2024):

- **EIP-4844:** Introduces “blob” transactions for Layer 2 data
- **Problem:** L2 rollups pay expensive calldata for data availability
- **Solution:** New data type (blobs) with separate, cheaper gas market

How Blobs Work:

- Blobs: 128 KB data chunks, stored for ~18 days
- Separate “blob gas” market (independent of execution gas)
- L2s post transaction batches as blobs instead of calldata

Impact:

- L2 transaction fees reduced by 90%+ (\$0.50 → \$0.01)
- Arbitrum, Optimism, Base adopted immediately
- Paves way for full Danksharding (future upgrade)

- ❶ **EVM:** Deterministic, stack-based VM enabling Turing-complete smart contracts
- ❷ **Two Account Types:** EOAs (user-controlled) and Contracts (code-controlled)
- ❸ **State Machine:** Ethereum transitions between states via transactions
- ❹ **World State:** Mapping of addresses to accounts, stored as MPT
- ❺ **MPT:** Efficient cryptographic data structure for state verification
- ❻ **Account Model:** Simpler than UTXO but with privacy/parallelization trade-offs

- ❶ Why is the EVM only “quasi”-Turing complete?
- ❷ What security benefits does the nonce provide?
- ❸ How does MPT enable light client verification?
- ❹ When might Bitcoin's UTXO model be preferable?
- ❺ How does state growth affect node operators?

Coming up next:

- Understanding gas as a computational unit
- Gas price, gas limit, and transaction cost calculation
- EIP-1559: Base fee and priority fee mechanism
- Gas costs for different EVM operations
- Optimization techniques for reducing gas consumption

Preparation:

- Review basic Ethereum transaction structure
- Familiarize yourself with Wei/Gwei/ETH units
- Browse Etherscan to see gas usage in real transactions