L13: Ethereum Architecture

Module B: Ethereum & Smart Contracts

Blockchain & Cryptocurrency Course

December 2025

Learning Objectives

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?

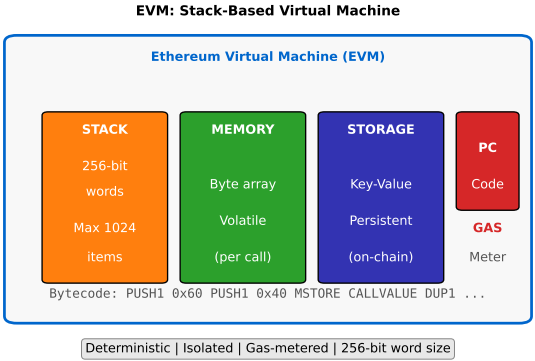
Key Characteristics:

- 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)



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

EVM: Technical Details

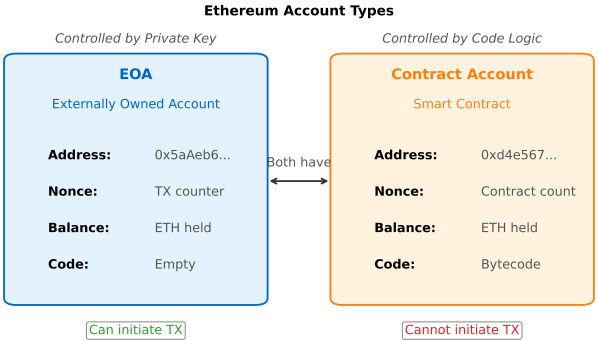
The EVM is a quasi-Turing complete state machine:

- 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



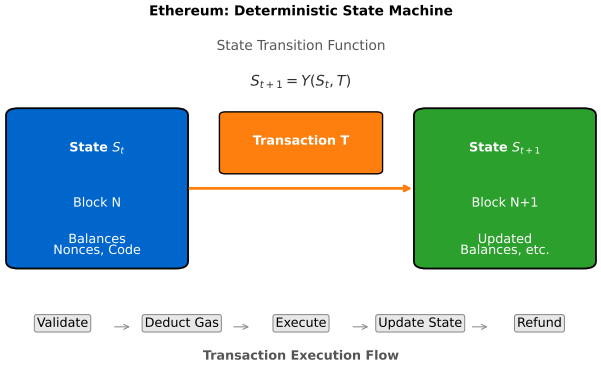
EOAs are controlled by private keys; contracts are controlled by code.

Account State Components

Every account (EOA or Contract) has four fields:

1. **Nonce:**
   - EOA: Counter of transactions sent
   - Contract: Counter of contracts created
   - Prevents replay attacks
2. **Balance:**
   - Amount of Wei (10^{-18} ETH) owned
   - 1 ETH = 1,000,000,000,000,000,000 Wei
3. **StorageRoot:**
   - Hash of account's storage trie root
   - Empty for EOAs
4. **CodeHash:**
   - Hash of EVM bytecode
   - Empty for EOAs, immutable for contracts

Ethereum as a State Machine



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

World State

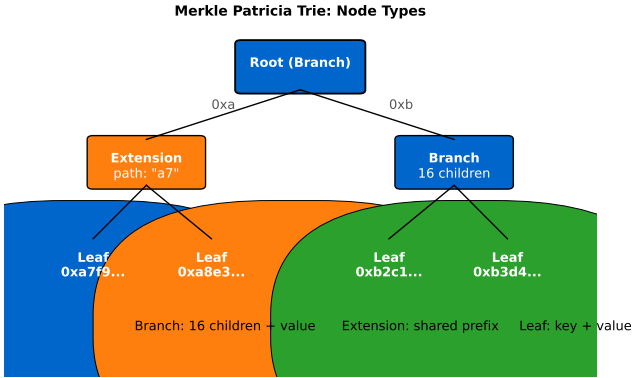
The World State is a mapping between addresses and account states:

- 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)



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

MPT Properties

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

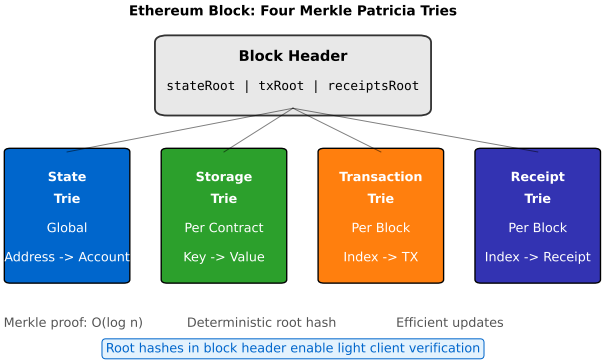
Key Properties:

- 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:

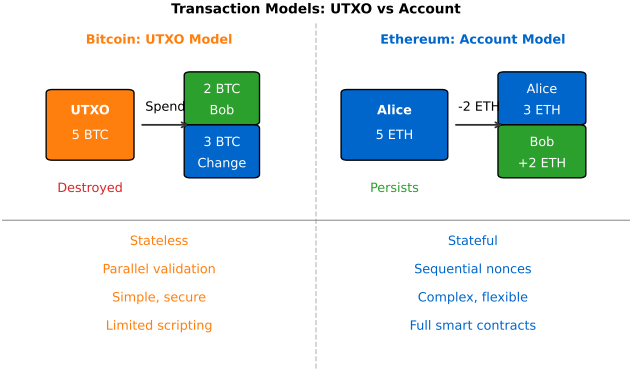
- **Branch:** 16 children (hex 0-F) + optional value
- **Extension:** Shared path prefix + next node pointer
- **Leaf:** Remaining key path + value

Four MPT Tries in Ethereum



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.

Account Model: Trade-offs

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:

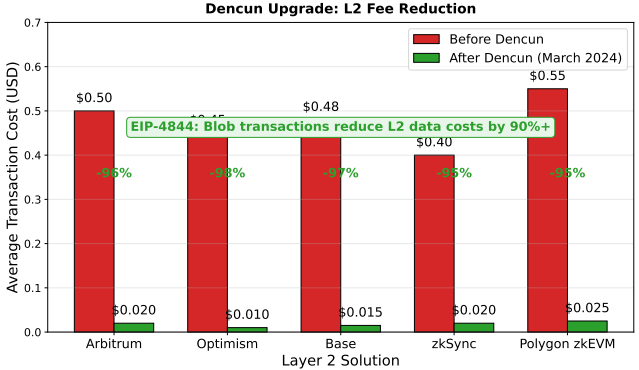
1. **Replay Prevention:** Requires nonce tracking
2. **State Growth:** All accounts stored indefinitely
3. **Privacy:** All activity linked to single address
4. **Parallelization:** Sequential nonces limit parallel validation

Block Structure

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%+.

Dencun: Proto-Danksharding

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 $\sim$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 $\rightarrow$ \$0.01)
- Arbitrum, Optimism, Base adopted immediately
- Paves way for full Danksharding (future upgrade)

Key Takeaways

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

Discussion Questions

1. Why is the EVM only “quasi”-Turing complete?
2. What security benefits does the nonce provide?
3. How does MPT enable light client verification?
4. When might Bitcoin's UTXO model be preferable?
5. How does state growth affect node operators?

Next Lesson Preview: L14 - Gas Mechanics

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