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 explain the world state concept
- Explain the Merkle Patricia Trie data structure
- Compare Ethereum's account model to Bitcoin's UTXO model

The Problem: How do we make money programmable?

The Challenge
How do we create a blockchain system that can execute arbitrary code with value at stake, where code execution is deterministic and verifiable across all nodes?

Why It Matters

- Bitcoin's scripting language only allows simple operations (send, verify signatures)
- Complex financial logic (lending, options, DAOs) requires Turing-complete computation
- Vitalik's 2013 whitepaper proposed "world computer"; Ethereum launched 2015; The DAO hack 2016 revealed execution risks

What We Need

- System design principles
- A virtual machine where all nodes execute code identically and reach consensus on state changes

The Cryptoeconomics Question
Creating systems with desired properties

Today's lesson: How Ethereum Architecture addresses this challenge

What is Ethereum?

Key Characteristics:

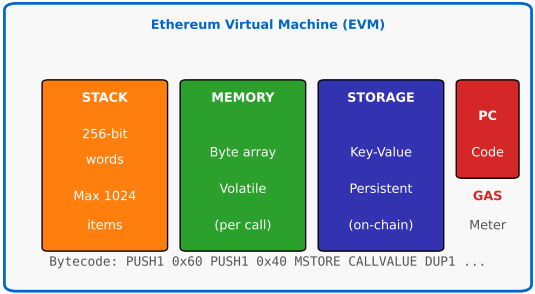
- Decentralized computation platform
- Turing-complete blockchain
- Smart contract support
- Native cryptocurrency: Ether (ETH)
- Launched July 30, 2015

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



Deterministic | Isolated | Gas-metered | 256-bit word size

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

EVM: Technical Details

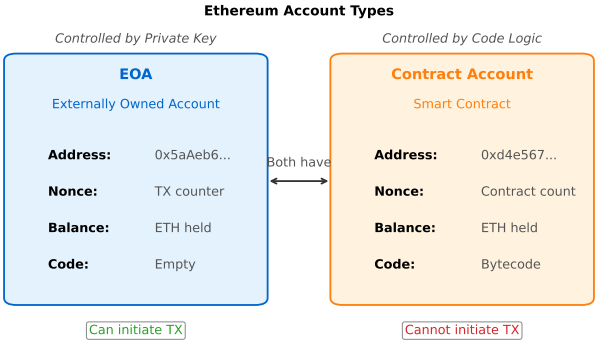
The EVM is a quasi-Turing complete state machine:

- 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



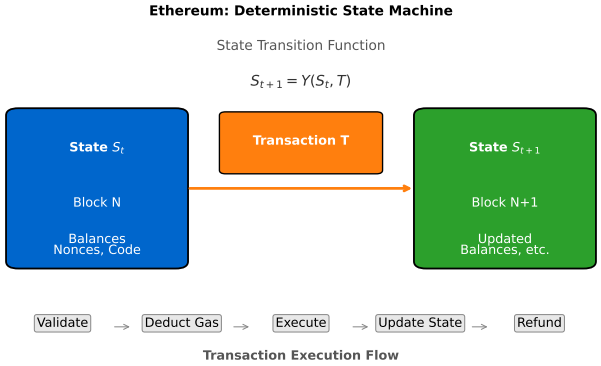
EOAs are controlled by private keys; contracts are controlled by code.

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

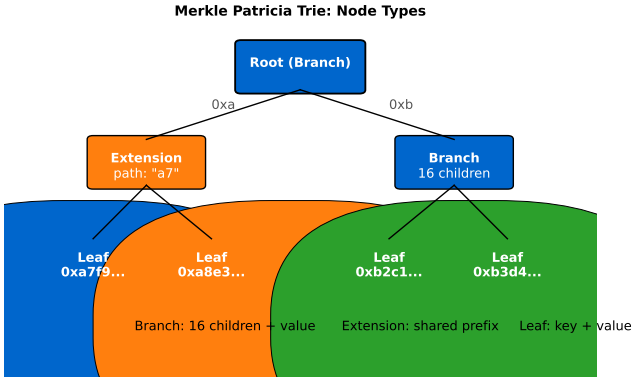
The World State is a mapping between addresses and account states:

- 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

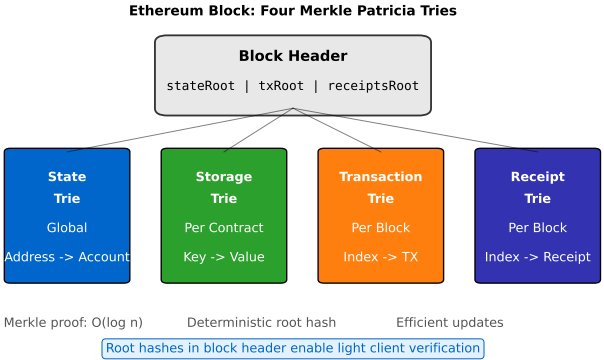
Key Properties:

- 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:

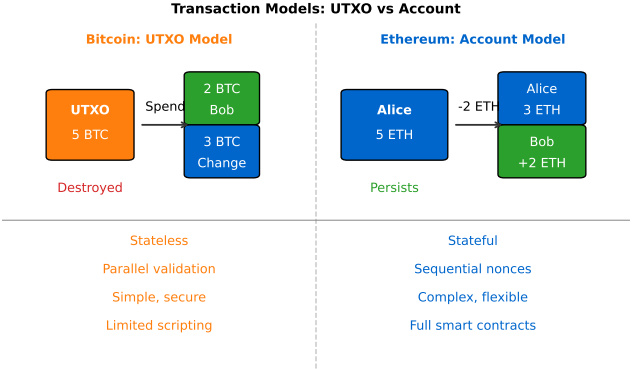
- **Branch:** 16 children (hex 0-F) + optional value
- **Extension:** Shared path prefix + next node pointer
- **Leaf:** Remaining key path + value

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:

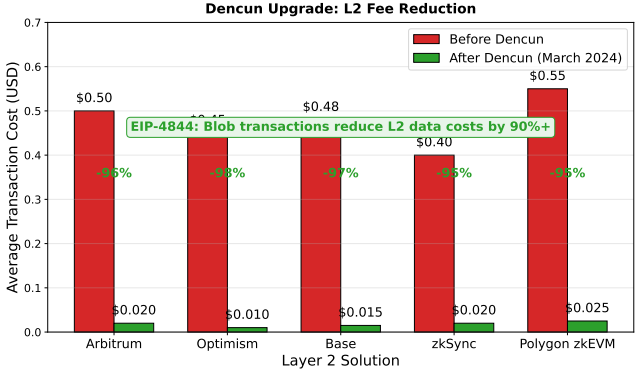
1. **Replay Prevention:** Requires nonce tracking
2. **State Growth:** All accounts stored indefinitely
3. **Privacy:** All activity linked to single address
4. **Parallelization:** Sequential nonces limit parallel validation

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)

Problem Solved? Ethereum Architecture

The Original Problem

How do we make money programmable?

How Ethereum Architecture Solves It

- **EVM executes bytecode**: All nodes run identical stack-based VM, ensuring deterministic state transitions
- **Account-based model**: Persistent storage (vs UTXO) enables complex contract logic and state management
- **World state + MPT**: Cryptographic verification of all account states at every block

Remaining Limitations

- **Execution costs**: Gas fees make computation expensive; state bloat from permanent storage
- **Single-threaded**: Sequential transaction processing (nonce ordering) limits parallelization and throughput

Open Questions

- How to scale smart contract execution (10,000+ TPS) while maintaining decentralization and security?
- Risk: State growth, MEV exploitation, smart contract bugs with irreversible financial consequences

Ethereum Architecture solves programmable money but introduces scalability and cost trade-offs

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

Quiz Questions (1/4)

Q1. What is the EVM's word size?
A) 128 bits B) 160 bits C) 256 bits D) 512 bits
Answer: C – The EVM uses 256-bit words for stack operations.

Q2. Which component makes the EVM "quasi-Turing complete" rather than fully Turing complete?
A) Stack limit B) Gas metering C) Memory size D) Bytecode length
Answer: B – Gas metering prevents infinite loops, limiting true Turing completeness.

Q3. What is the maximum stack depth in the EVM?
A) 256 items B) 512 items C) 1024 items D) 2048 items
Answer: C – The EVM stack can hold a maximum of 1024 items.

Q4. Which account type is controlled by private keys?
A) Contract accounts B) Both types C) Neither type D) Externally Owned Accounts (EOA)
Answer: D – EOAs are controlled by private keys; contracts are controlled by code.

Q5. How many Wei are in 1 ETH?
A) 10^9 B) 10^{12} C) 10^{18} D) 10^{21}
Answer: C – 1 ETH = 1,000,000,000,000,000,000 Wei (10^{18}).

Quiz Questions (2/4)

Q6. What does the nonce field track for an EOA?
A) Balance B) Transaction count sent C) Gas used D) Contract deployments
Answer: B – For EOAs, nonce counts transactions sent to prevent replay attacks.

Q7. Which field is empty for Externally Owned Accounts but populated for Contract Accounts?
A) Nonce B) Balance C) CodeHash D) All fields are populated
Answer: C – EOAs have no code, so CodeHash is empty; contracts store bytecode.

Q8. What data structure is used to store the Ethereum world state?
A) Hash table B) Binary tree C) Merkle Patricia Trie D) Bloom filter
Answer: C – The world state uses a Merkle Patricia Trie for efficient verification.

Q9. How many tries does an Ethereum block header reference?
A) One B) Two C) Three D) Four
Answer: D – State, transaction, receipt, and storage tries (four total).

Q10. What is the state root?
A) First block hash B) 256-bit hash of entire world state C) Genesis state D) Merkle tree root
Answer: B – The state root is a 256-bit hash representing the entire world state.

Quiz Questions (3/4)

Q11. Which node type in MPT has 16 children?
A) Leaf B) Extension C) Branch D) Root
Answer: C – Branch nodes have 16 children (hex 0-F) plus optional value.

Q12. What is the primary advantage of Ethereum's account model over Bitcoin's UTXO model?
A) Better privacy B) Easier parallelization C) Simpler for smart contracts D) Smaller state size
Answer: C – Account model is more natural for persistent contract storage.

Q13. What is Ethereum often called to distinguish it from Bitcoin?
A) Digital silver B) World computer C) Smart ledger D) Crypto platform
Answer: B – Ethereum is called a "world computer" for programmable logic.

Q14. When was Ethereum mainnet launched?
A) January 3, 2009 B) July 30, 2015 C) August 1, 2017 D) December 1, 2020
Answer: B – Ethereum launched July 30, 2015 (Bitcoin was 2009).

Q15. Which EVM memory type is persistent and stored on-chain?
A) Stack B) Memory C) Storage D) Cache
Answer: C – Storage is persistent on-chain (expensive); Memory is volatile per call.

Quiz Questions (4/4)

Q16. What does EIP-4844 (Dencun upgrade) introduce?
A) Proof of Stake B) Blob transactions C) Smart contracts D) Sharding
Answer: B – EIP-4844 introduces blob transactions for cheaper L2 data availability.

Q17. By approximately how much did Dencun reduce Layer 2 transaction fees?
A) 30% B) 50% C) 70% D) 90%+
Answer: D – L2 fees dropped 90%+ (e.g., \$0.50 to \$0.01) after Dencun.

Q18. What is the address size in Ethereum?
A) 128 bits B) 160 bits C) 256 bits D) 512 bits
Answer: B – Ethereum addresses are 160 bits (40 hex characters).

Q19. Which field in the account state prevents replay attacks?
A) Balance B) Nonce C) CodeHash D) StorageRoot
Answer: B – The nonce ensures each transaction is unique and sequential.

Q20. What challenge does Ethereum's account model face compared to UTXO?
A) Complexity B) Privacy limitations C) Higher storage D) No smart contracts
Answer: B – All activity is linked to a single address, reducing privacy vs UTXO.