Ethereum and the EVM

A Five-Minute Overview

BSc Blockchain Course

What If Money Could Think?

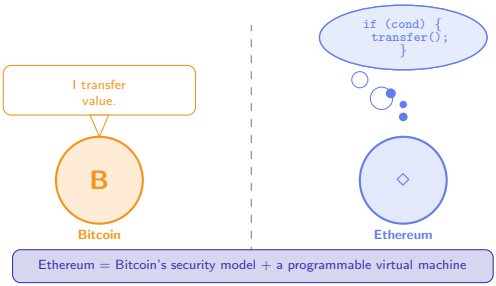
Bitcoin proved that a decentralised network could transfer value without a central bank. But it deliberately does almost nothing else.

What Bitcoin cannot do natively:

- Execute conditional logic ("pay only if deadline passed")
- Hold state between transactions
- Call other contracts or compose protocols

Vitalik Buterin's 2013 insight: add a *Turing-complete virtual machine* to the blockchain so that arbitrary programs – **smart contracts** – could run deterministically on every node.

Result: Ethereum is a *programmable state machine* whose state transitions are enforced by global consensus. Every node computes the same result; no central server needed.



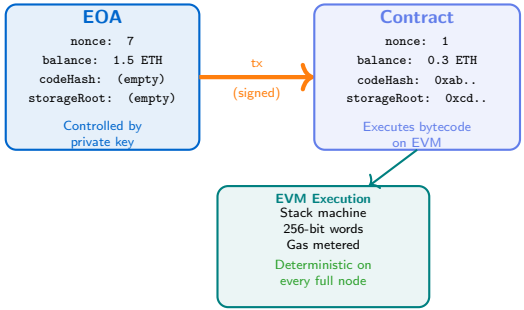
Ethereum generalises Bitcoin: instead of a single "send coins" script, any arbitrary program can run on the EVM, making Ethereum a *world computer* whose outputs are verified by all participants without trusting any one of them.

Source: Buterin, V. (2013). "Ethereum White Paper." ethereum.org/en/whitepaper

What Makes an Ethereum Account Different from a Bitcoin Address?

Ethereum's world state is a mapping from 20-byte addresses to account objects. Two kinds of account exist:

- **EOA** (Externally Owned Account) – controlled by a private key. Can initiate transactions. Has no code.
- **Contract Account** – created by deploying bytecode. Executes only when called; cannot act autonomously.



Every account stores four fields:

Field	Meaning
`nonce`	tx count (EOA) / contract-creation count
`balance`	Wei held (1 ETH = 10^{18} Wei)
`codeHash`	Keccak256 of EVM bytecode
`storageRoot`	Root of Merkle Patricia Trie

EOAs have `codeHash` = hash of empty string and `storageRoot` = empty trie root.

A transaction from an EOA triggers EVM execution inside a contract, which can read/write its storage trie, send ETH, and call other contracts – all within a single atomic transaction that either fully succeeds or fully reverts.

Source: Ethereum Yellow Paper (Wood, G., 2014, updated 2024). ethereum.github.io/yellowpaper

Why Does Every Instruction on Ethereum Cost Money?

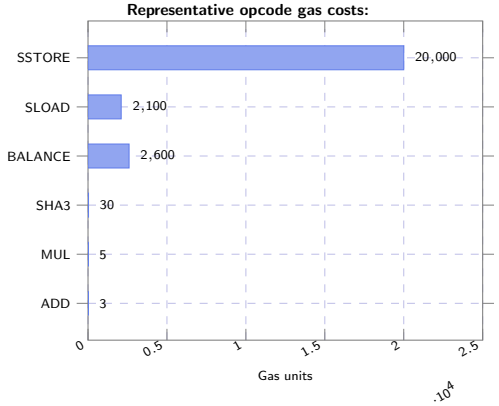
Without a cost per computation, any user could submit an infinite loop and stall every node. **Gas** is the unit of computational work on Ethereum.

How gas works:

- Every opcode has a fixed gas cost (e.g., ADD costs 3 gas)
- The sender sets a *gas limit* (max willing to spend)
- **EIP-1559** (Aug 2021): fee split into *base fee* (burned) and *priority fee* (tip to validator)
- Base fee adjusts dynamically: rises when blocks are $> 50\%$ full, falls when below

Why burning matters:
The base fee is destroyed, making ETH deflationary when demand is high. During peak usage (e.g., NFT mints), burn can exceed issuance – net supply falls.

Formula: Total fee = (base fee + priority fee) $\times$ gas used.



Storage ops (SSTORE/SLOAD) cost 1000× arithmetic – encouraging minimal on-chain storage.

Gas is Ethereum's anti-spam and resource-pricing mechanism: expensive operations (state writes) cost more than cheap ones (arithmetic), making abuse economically irrational while keeping the network usable.

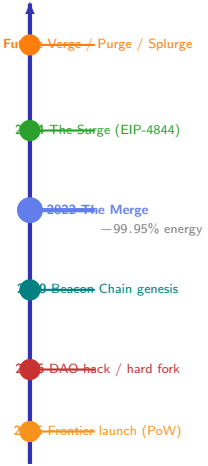
Source: EIP-1559 (Buterin et al., 2021), eips.ethereum.org/EIPS/eip-1559; Ethereum Yellow Paper Appendix G.

How Did Ethereum Switch Its Engine Without Stopping?

Ethereum launched on Proof-of-Work in 2015, burning as much electricity as a mid-sized country. The switch to Proof-of-Stake – called **The Merge** – reduced energy consumption by $\approx 99.95\%$.

Why the roadmap matters:

- **The Merge (Sep 2022):** Execution layer merged with Beacon Chain. Validators stake 32 ETH; miners retired.
- **The Surge:** EIP-4844 (proto-danksharding) – cheap data blobs for L2 rollups. Targets $>$ 100,000 TPS system-wide.
- **The Verge:** Stateless clients via Verkle trees – nodes no longer store full state.
- **The Purge:** EIP-4444 prunes old history; nodes shrink from TBs to GBs.
- **The Splurge:** EIP-7702 (account abstraction), EOF bytecode format.



The Merge is the most significant upgrade in Ethereum's history: it replaced energy-intensive mining with stake-based validation and set the stage for the Surge's L2 scaling – the path to millions of transactions per second without sacrificing decentralisation.

Source: ethereum.org/en/roadmap (2024); Ethereum Foundation Blog, "The Merge" (Sep 2022).

Five Reasons Ethereum Changes How We Build Software

Ethereum is the infrastructure layer on which most of the decentralised economy is built. Understanding it is a prerequisite for every subsequent lecture.

What Ethereum enables:

- **DeFi (L07):** Lending, trading, and derivatives with no bank – billions locked in smart contracts, open to anyone.
- **NFTs (L08):** Provable digital ownership – ERC-721 tokens that the EVM enforces without a registry.
- **DAOs (L11):** Organisations governed by code – token votes trigger on-chain treasury transfers.

Risks to understand:

- **Smart contract bugs:** Code is immutable once deployed; vulnerabilities cannot be silently patched.
- **Gas volatility:** Peak demand can price out small users (L2 rollups partially solve this).
- **Regulatory uncertainty:** Proof-of-Stake ETH may qualify as a security in some jurisdictions.

The composability principle: Any contract can call any other contract. DeFi protocols stack like Lego bricks: a yield aggregator calls a lending pool, which calls a price oracle – all in one transaction. This is impossible in traditional finance.

Bridge to the next lecture (L06 – Solidity):
Now that you understand the EVM and accounts, you can write code that runs on it. Solidity compiles to EVM bytecode; the gas costs and storage layout you saw here directly constrain how you write efficient contracts.

Key vocabulary: EVM, EOA, smart contract, gas, EIP-1559, The Merge, L2 rollup.

Source: ethereum.org; DeFiLlama (defillama.com); Ethereum Foundation Annual Report 2023.