Blockchain and Ethereum Virtual Machine (EVM): Summary

Lucas Kim
Loyola Marymount University
Los Angeles, US

Abstract

This summary introduces blockchain and the Ethereum Virtual Machine (EVM) in simple terms. It explains how blockchain works, what smart contracts are, and why gas and computation limits matter. Blockchain is not magic—it is verifiable computation shared by thousands of nodes..

Blockchain Driving Principle: "Trust Without a Boss"

Imagine a shared notebook: Everyone has a copy. To add a page, everyone must agree and write the same thing. No single person controls it.

How it works:

(1) **Transactions:** "Alice sends 1 coin to Bob."
(2) **Block:** Group 100 transactions into one page.
(3) **Hash link:** Each page includes a fingerprint (hash) of the previous page. Change one letter → all later fingerprints break.
(4) **Consensus:** Validators stake coins or miners solve puzzles to add the next block. The longest valid chain wins.

Result: No central bank needed. Anyone can verify the entire history. Tampering is impossible without rewriting the whole chain faster than the network.

Ethereum Virtual Machine (EVM): The Global Computer Inside Blockchain

Analogy: Think of the EVM as a calculator inside every Ethereum node. Everyone runs the same calculator on the same inputs → same output.

EVM Driving Principle:

(1) **Bytecode:** Smart contracts compile to EVM instructions (like assembly code).
(2) **Deterministic:** Same code + same input = same result on every computer.
(3) **Stack-based:** Uses a 1024-item stack (256-bit words). Opcodes: ADD, MUL, JUMP, etc.
(4) **Stateful:** Contracts have permanent storage that survives years.

Key: The EVM turns blockchain from "digital money" into "programmable money."

Smart Contract: "Self-Executing Agreement"

Example: Crowdfunding.

```
if (raised >= goal) {
  sendToProject();
} else {
  refundAll();
}
```

Definition:

- Code deployed on blockchain.
- Runs automatically when called.
- Cannot be changed after deployment (immutable).
- Anyone can read and verify the rules.

No middleman: No lawyer, no bank. The code *is* the contract.

Gas Fee: "Pay-Per-Calculation"

Analogy: Using a payphone. You insert coins per second of talk time.

Why gas exists:

(1) **Prevent spam:** Infinite loops would crash the network.
(2) **Pay nodes:** Validators run your code; they deserve compensation.
(3) **Measure work:** Simple ADD costs 3 gas, complex math costs more.

Formula:

$$\text{Total Fee} = \text{Gas Used} \times \text{Gas Price (in ETH)}$$

EIP-1559: Base fee (burned) + tip (to validator). Makes fees predictable.

Why Computational Complexity Matters

Dangerous code = network crash.

- Loop 1 billion times → every node freezes.
- Gas caps this: run out → transaction fails safely.

Real attacks:

- **2016 DAO Hack:** Reentrancy drained $50M.
- **Denial-of-Service:** Fill block with expensive ops.

Best practice: Write efficient code. Test gas usage. Use secure libraries (OpenZeppelin).

Key Takeaway

Blockchain = shared truth. EVM = shared computer. Gas = fuel. Smart contracts = unbreakable rules.

References

[1] Ethereum.org: https://ethereum.org/en/what-is-ethereum/
[2] G. Wood. Ethereum Yellow Paper. https://ethereum.github.io/yellowpaper/paper.pdf