The DAO Fork: The $60 Million Bug That Split a Blockchain

BSc Blockchain Course – Supplementary Lecture

Digital Finance

April 2016: A Radical Experiment Begins

ACT 1: THE DREAM

Ethereum is 10 months old. A team launches "The DAO" – a **smart contract** acting as a **decentralised venture capital fund**.

- No CEO, no board, no headquarters. Just **code** and **token holders** voting on investments.
- **Decentralised governance**: anyone holding DAO tokens can propose and vote on how funds are deployed.
- The code is deployed on Ethereum and is **immutable** – nobody can change it after launch.

The Analogy

Imagine Kickstarter, but the backers control the money – *forever*. No refunds, no management team, no override button. The **smart contract** is the CEO.

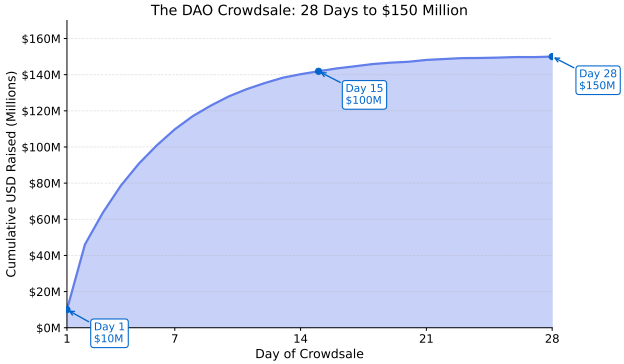
The DAO was built by Christoph Jentzsch and the Slock.it team on Ethereum.

28 Days, $150 Million, 11,000 Investors

The largest crowdfund in history (at that time):

- $150 million raised in 28 days
- 11,000 individual investors
- **14% of all ETH** locked in a single contract
- No legal entity, no terms of service
- Exchange rate: **1 ETH = 100 DAO tokens**

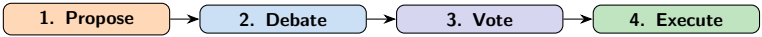
The excitement was real – but so was the risk. No legal recourse. No insurance. No audit had been completed. Just code.



At its peak, The DAO held 14% of all Ether – an unprecedented concentration of value in a single contract.

The Machine: How The DAO Made Decisions

Governance lifecycle: from idea to execution



But there was a fifth function nobody thought much about: `splitDAO()` – **the exit door**.

- If you **disagreed with the majority**, you could take your ETH and leave.
- You would create a **child DAO** – a new fork of the contract holding your proportional share.
- This was designed to *protect minority investors* from being outvoted forever.

The split function was meant to protect minority investors. It became the attack vector.

ACT 2: THE CRISIS

Security researchers identify a **reentrancy vulnerability** in the `splitDAO()` function.

- The bug is **publicly disclosed**. The community reads the report. The concern is acknowledged.
- But the contract is already deployed on Ethereum. **Smart contracts are immutable** – you cannot patch deployed code.
- $150 million sits in a contract with a known flaw, and nobody has the keys to fix it.
- Proposals are floated to drain the contract safely. None reach consensus in time.

The Dilemma

"The money sat there, behind a locked door with a known broken lock." The community could see the problem. They could not act.

The warning was published by Peter Vessenes on June 9. The attack began June 17 – eight days later.

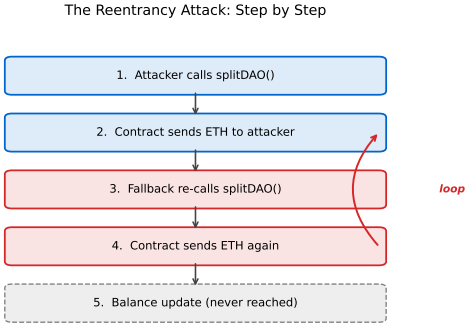
June 17, 2016: 3.6 Million ETH Vanishes

The reentrancy attack – the ATM analogy:

- Like pressing "withdraw $100" *before your balance updates* – repeatedly.
- The attacker called `splitDAO()` in a loop, draining ETH each iteration.
- **3.6 million ETH** ($60 million) moved to a child DAO.

The twist: child DAOs had a **28-day lock**. The attacker could not move the money yet.

The community had **28 days to decide**.



The bug: ETH was sent before the balance was updated. The fix: always update state before sending (Checks-Effects-Interactions pattern).

Panic, Proposals, and a Ticking Clock

A compressed timeline of community response:

- **June 17** – Attack discovered. ETH price drops 33% within hours. Exchanges halt trading.
- **June 18** – Vitalik Buterin proposes a **soft fork** to freeze the attacker's child DAO funds.
- **June 21** – The attacker publishes an open letter:
  "I followed the code's rules. This is not theft."
- **June 24** – The soft fork proposal is found to have its *own* DoS vulnerability. Abandoned.
- **June 28** – The only remaining option: a **hard fork** to reverse the transaction entirely.

In 28 days, the attacker's lock period would expire. The funds would become permanently and irrecoverably theirs.

The clock was ticking.

The attacker's open letter argued: "A contract is a contract. I made use of this feature and rightfully claimed my reward."

Code Is Law vs. The Will of the People

ACT 3: THE DILEMMA

"Code Is Law" – No Fork

- The contract executed as written. No rule was broken.
- Reversing a transaction destroys **immutability** – Ethereum's core guarantee.
- If we fork once, we fork whenever powerful people lose money.
- Sets a precedent: **blockchains are not censorship-resistant** if developers can override them.

"Intent Matters" – Fork

- The attacker exploited a bug, not a feature. **Intent** and outcome differ.
- A nascent ecosystem cannot survive a $60M theft in its first year.
- Ethereum is a social contract, not just software – **communities can self-govern**.
- The funds can be recovered with minimal collateral damage.

Neither side was wrong. Both were asking the same question: *Who has ultimate authority over a blockchain?*

This debate remains unresolved. Every blockchain community faces this question eventually.

The Community Decides: Fork or No Fork?

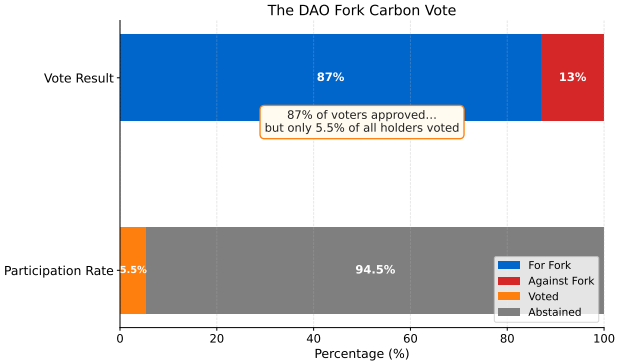
The Carbon Vote – an on-chain governance experiment:

- **87%** of voting ETH supported the hard fork.
- But only **5.5%** of all ETH holders voted.

Is 5% voter turnout a legitimate mandate?

The Governance Problem

Democracy requires participation. What happens when 95% stay home – *by design*?



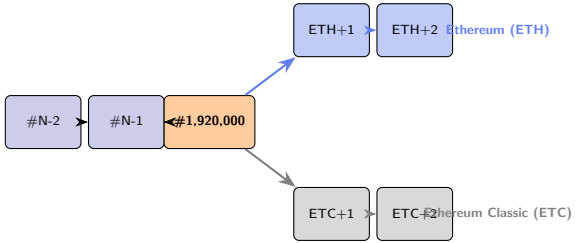
The carbon vote is often cited as evidence both for and against on-chain governance.

Block 1,920,000: Ethereum Splits in Two

On **July 20, 2016** at block 1,920,000, an **irregular state transition** was applied:

- Ethereum's developers manually moved the stolen ETH to a **refund contract**.
- Miners who ran the new software followed the forked chain.
- Miners who rejected the fork kept the original chain.
- **Both chains are "correct"** – they simply disagree on one block.

The market assigned names: **ETH** for the fork, **ETC** for the original.



An irregular state transition is a one-time manual edit to the blockchain state – the only one in Ethereum's history.

Two Chains, Two Philosophies, Two Prices

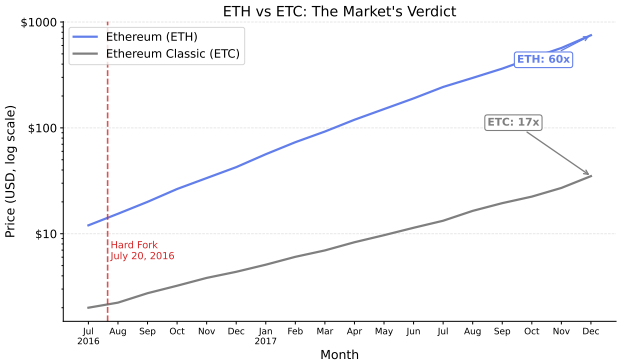
ACT 4: THE LEGACY

The market voted with its wallet:

- ETH recovered and grew to become the second-largest blockchain.
- ETC survived but was **51%-attacked three times in 2020**.

The DAO hack created the modern smart contract security industry:

- **Checks-Effects-Interactions** pattern
- Formal verification tools
- $200K–$1M audit industry standard
- **OpenZeppelin** secure contract library



Ethereum Classic's 51% attacks in 2020 are often cited as evidence that smaller PoW chains cannot maintain security.

Lessons That Apply to Every Blockchain Project

Five lessons that outlasted The DAO:

1. **Smart contracts are immutable – test before you deploy.** There is no patch, no hotfix, no support ticket once code is live.
2. **"Code is law" has limits.** Communities will intervene when the stakes are high enough. Every blockchain is ultimately a social consensus machine.
3. **Governance is the hardest problem.** Technical consensus (miners, nodes) is easier than social consensus (developers, users, investors, exchanges). The DAO fork required both – and almost failed at the social layer.
4. **Security audits are not optional.** The DAO was audited. But the audit was incomplete and the schedule was rushed. A thorough audit is not a checkbox – it is a process.
5. **Forks are the ultimate governance mechanism.** When a community cannot agree, it splits. Both outcomes can coexist. Markets determine which vision survives.

The DAO fork proved that blockchains are not just technology – they are social systems that happen to run on code.

Every concept you've learned in this course – from hashing to consensus to smart contracts – intersected in the DAO crisis.