Introduction to Blockchain

A Standalone Mini-Course

BSc Blockchain Course

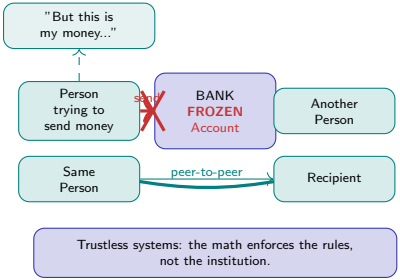
Why Would Anyone Trust a Network of Strangers with Their Money?

Your bank account can be frozen without warning. A payment to another country can take three days and cost a small percentage of the amount. A government can devalue the currency overnight. These are not hypothetical stories – millions of people experience them every year.

The question blockchain asks:

- What if no single institution could freeze your funds?
- What if a payment settled in seconds, not days?
- What if the rules were written in code – visible to everyone and changeable by no one?

The answer was not to find a more trustworthy bank. The answer was to remove the bank from the equation entirely – and replace trust in institutions with trust in mathematics.



Blockchain does not make people more trustworthy – it makes trust unnecessary by encoding the rules in publicly verifiable mathematics.

Blockchain = replace institutional trust with mathematical proof.

Think About the Last Time You Sent Money – Who Did You Have to Trust?

You have done this many times – paid a friend back, sent money to a family member, bought something online. You probably did it in seconds without thinking. But underneath that tap or click was a chain of institutions, each one a potential point of delay, failure, or refusal.

Quick Exercise – Think Before We Continue

1. **List every party involved the last time you sent or received money.** Start with your phone or card. Who processed the request? Which networks carried the message? Which institutions held the funds at each end? Write down every name.
2. **How long did it actually take – and what did it cost?** Was it instant, or did the funds arrive a day later? Was there a fee, a conversion spread, or a minimum amount? Could you have sent one euro to someone in another continent for free?
3. **What if one institution in that chain had refused?** Could your account have been frozen? Could the payment have been reversed by someone other than you? Who, if anyone, could you have called?

Bring your answers to class. We will use your real examples as running cases throughout this session.

Every payment you have ever made relied on institutions you never chose and rarely questioned.

What Makes Blockchain Different from Every Other Database?

Five database types – same data, very different rules:

Type	Control	Mutable?	Transparent?	Trust model
Traditional DB	Single owner	Fully	Private	Owner
Cloud DB	Cloud provider	Fully	Private	Provider
Distributed DB	Consortium	Fully	Partial	Members
Permissioned BC	Consortium	Append-only	Partial	Members
Public Blockchain	No owner	Append-only	Fully public	Math + code

The pattern to notice: Read across the last row. Every property that requires trust in the other rows is replaced by a technical guarantee. No owner – no single point of coercion. Append-only – no one can quietly rewrite history. Fully public – anyone can audit. Trust the math, not the manager.

What "append-only" really means:

- **Traditional DB:** row can be updated or deleted silently. Yesterday's record is gone.
- **Blockchain:** every record is cryptographically chained (linked by code fingerprints). To change block 100, you must redo block 101, 102, 103. . . and outpace every other participant simultaneously.

What "no owner" costs:

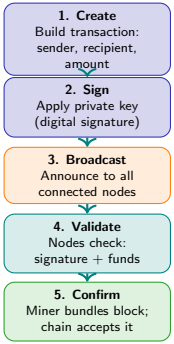
- Slower to write (many must agree)
- Harder to fix errors
- No customer support

These trade-offs are deliberate – not bugs.

Blockchain is not a better database – it is a different contract: you give up speed and flexibility in exchange for a history that no single party can revise.

Append-only + no single owner + public auditability = the blockchain guarantee.

Follow One Bitcoin Transaction from Send to Confirmed



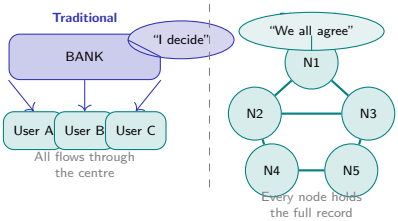
Step by step – in plain language:

1. Your wallet software builds a message: "I want to send 0.01 coins from address A to address B." Nothing has moved yet.
2. Your wallet uses your private key (a secret number only you hold) to produce a mathematical signature. Anyone can verify the signature is yours – but no one can forge it without your key.
3. Your signed message is broadcast to the peer-to-peer network. Within seconds, thousands of nodes have a copy.
4. Each node independently checks: is the signature valid? Does address A have enough funds? If yes, the transaction enters the waiting pool.
5. A miner collects waiting transactions, groups them into a block, and solves a computational puzzle. Once solved, the block – and your transaction – is permanent.

The entire journey from your device to permanent record takes roughly ten minutes and involves no bank, no intermediary, and no permission.

Five steps: create, sign, broadcast, validate, confirm – each enforced by code, not people.

Who Validates Your Transaction – A Bank, A Miner, or the Crowd?



How consensus replaces central authority:

- **Traditional system:** one institution keeps the ledger. If it says you have no funds, the conversation ends. You have no copy, no appeal, no alternative.
- **Blockchain:** thousands of nodes each hold the complete ledger. To change a record, you would need to simultaneously overpower the majority of the entire network.

The consensus mechanism (proof of work):

- Nodes compete to add the next block by solving a puzzle that requires enormous computation
- The winner is rewarded; all others verify the answer
- The longest chain is the agreed truth

Changing history requires more computing power than the rest of the network combined – economically impractical.

Consensus is the mechanism by which a crowd of strangers – with no shared trust – arrives at the same truth without a referee.

Consensus = every node agrees, every node verifies – no single point of authority.

The Code Worked Perfectly – So Why Did Everyone Lose Sixty Million Dollars?

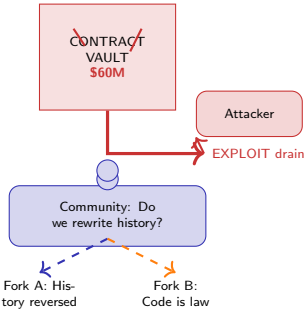
The DAO Exploit (2016) In 2016, a smart contract (a piece of self-executing code) was launched to pool funds and vote on investments. Hundreds of millions of dollars flowed in from thousands of participants worldwide.

An attacker found a flaw: the contract sent funds *before* recording that it had done so. By repeatedly triggering a withdrawal before the balance was updated, the attacker drained roughly sixty million dollars – all within the rules as the code understood them.

The hard fork dilemma:

- The code performed exactly as written
- "Code is law" – should the exploit stand?
- The community voted to rewrite history – a hard fork reversed the transactions
- A minority refused; the original chain lived on as a separate currency

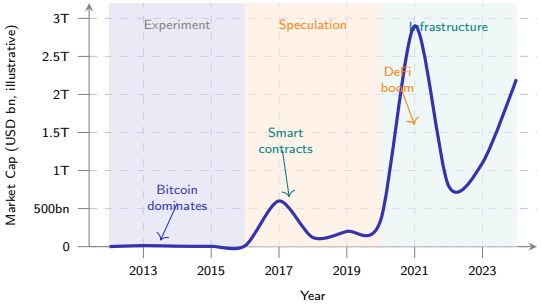
Lesson: immutability is a feature until it is a catastrophe.



The code worked exactly as written. The problem was the design – and the community's choice to override immutability proved that "code is law" is a philosophy, not a physics law.

Smart contract exploits: the code did what it said, not what the authors intended.

Why Has the Crypto Market Grown from Zero to Three Trillion Dollars?



Illustrative trend based on published indices. Not investment advice.

Three phases, three different stories:

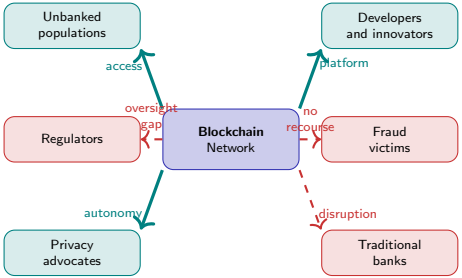
- **Experiment (early years):** Almost no one knew the technology existed. A tiny community of cryptographers and libertarians exchanged tokens worth fractions of a cent. Market cap was smaller than a mid-sized restaurant.
- **Speculation (mid period):** Retail investors joined in waves. Prices surged and crashed repeatedly. Most gains were driven by narrative, not utility. Many projects raised money and delivered nothing.
- **Infrastructure (recent years):** Institutional capital entered. Smart-contract platforms hosted real financial products. Governments began regulating. The market became too large for serious institutions to ignore.

Each phase was necessary for the next – the speculation funded the infrastructure.

Three trillion dollars is not the measure of blockchain's utility – it is the measure of how many people believe in its future utility. Those are very different things.

Market size follows belief cycles: experiment, speculation, infrastructure – the pattern repeats across technologies.

Who Benefits When Banks Are No Longer Required – And Who Gets Hurt?



The same system – different views:

Unbanked: Over a billion adults worldwide hold no bank account. A phone and internet access are now sufficient to store and transfer value across borders.

Developers: Open, permissionless platforms let anyone build financial products without regulatory approval or corporate partnership.

Privacy: Pseudonymous addresses allow transactions without sharing identity with a central authority.

Banks: Payment revenue and lending margins face pressure from systems that replicate their functions without their cost structure.

Regulators: Anonymous, cross-border, 24/7 transactions challenge existing frameworks built around national institutions.

Fraud victims: No central authority can reverse a confirmed transaction. Scam victims have no one to call.

The same property – no central authority – is liberation for the unbanked and impunity for the fraudster. Technology is neutral; outcomes depend on who uses it and why.

Decentralization redistributes power – not always to those who need it most.

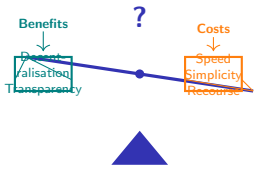
Three Questions That Reveal Whether a System Truly Needs Blockchain

Before any organisation adopts blockchain, ask:

Q1 **Do multiple parties need to share a record – and do they distrust each other?** If a single trusted party could hold the database, a conventional system is simpler, faster, and cheaper. Blockchain only earns its cost when distrust between participants is genuine and persistent.

Q2 **Is immutability genuinely required – or just convenient?** Immutability means errors, fraud, and changed circumstances are permanently embedded in the record. Many applications require the ability to correct mistakes. If you can live without the ability to rewrite history, immutability is an asset; if you cannot, it is a liability.

Q3 **Are the participants willing to accept the speed, cost, and complexity trade-offs?** Public blockchains process far fewer transactions per second than a bank's core system. Every participant must run or trust a node. Operational complexity is an order of magnitude higher. If performance requirements exceed what the network provides, blockchain is the wrong tool.



Every design choice tips the balance.
Only Q1–Q3 tell you which way.

Quick diagnostic:

- All three YES → blockchain is a strong candidate
- Two YES, one NO → evaluate the trade-off carefully
- One YES or fewer → a shared database is probably better

Blockchain is not a universal upgrade. It is a solution to one specific problem: coordinating records between parties who cannot agree on a single trusted keeper.

Apply Q1–Q3 to every blockchain proposal you encounter. Most do not pass all three.

Your Challenge

Read the case below, then apply the three questions from the previous slide.

Case: Multi-Clinic Patient Record Sharing

Situation: A hospital network wants to share patient records across five clinics operated by different organisations. No single clinic is willing to let another manage the central database – they do not trust each other with data ownership, and each has its own IT infrastructure. Patients move between clinics, and delays in accessing records have caused repeated treatment errors.

Apply the three questions. Fill in the table below:

Question	Your answer (Yes / No / Partial)	Reasoning (one sentence)
Q1: Multiple distrusting parties sharing a record?	.....	.....
Q2: Immutability genuinely required?	.....	.....
Q3: Trade-offs acceptable (speed, cost, complexity)?	.....	.....

Discuss with your neighbour (3 minutes): Does this case need blockchain – or would a shared database with strict access controls and an independent audit log be enough? Where do you disagree? What additional information would change your answer?

The hardest skill in blockchain is knowing when NOT to use it. That is what Q1–Q3 trains.