Blockchain: The Trustless Trust Paradox

We built systems to eliminate middlemen — but trust doesn't disappear, it just moves

Digital Finance

Why Would You Trust a Network of Strangers with Your Money?

The Paradox

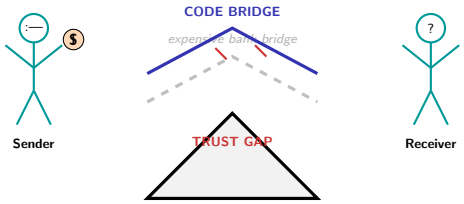
Every financial transaction requires trust. Traditionally, banks, courts, and lawyers provide it – at enormous cost. Blockchain promised to replace all of that with mathematics and code.

What makes blockchain attractive:

- No single institution controls your money
- Transactions are transparent and verifiable by anyone
- Settlement happens without waiting for a bank to approve
- Code runs exactly as written – no discretion, no bias

What blockchain cannot escape:

- You still trust the code – and code has bugs
- You still trust the miners or validators – and they have incentives
- You still trust the network – and networks can be attacked
- You still trust the oracle – and oracles are single points of failure



Every digital transaction needs trust.
The question is: who provides it?

Every digital transaction needs trust. Blockchain does not eliminate it – it moves it from institutions to algorithms.

When Was the Last Time You Thought About Who You Trust with Your Money?

Reflection Prompt

Think about what happened the last time you tapped your card or sent money through an app. Between your finger touching the screen and the money arriving, how many institutions did you trust?

Your bank, their bank, the card network, the payment processor, the app developer, the cloud provider. Did you choose to trust any of them – or did you just assume it would work?

The honest answer for most of us: we never think about it. Trust is invisible when it works.

The layers of trust you use every day:

- Your bank will not lose your money (deposit insurance, regulation)
- The card network will reverse a fraudulent charge (consumer protection)
- The payment app will deliver the money to the right person (software correctness)
- The recipient's bank will credit the right account (interbank settlement)

Blockchain's promise is to replace these intermediaries with code and cryptography. The question this lecture explores: **does "trustless" mean you trust nobody – or does it mean you trust different things?**

The answer matters. If you replace a regulated bank with an unaudited smart contract, you have not eliminated trust. You have moved it from an institution with legal obligations to a codebase with none.

You trust your bank, your payment app, and your card network every day – without thinking about it. Blockchain asks: what if you did not have to?

What Does Blockchain Actually Replace – and What Does It Keep?

Dimension	Traditional Finance	Blockchain
Trust source	Banks, courts, regulators	Code, cryptography, consensus
Settlement speed	1–3 business days (T+2)	Minutes to seconds
Transparency	Closed ledgers, audits	Public or shared ledger
Cost structure	Intermediary fees, compliance	Gas fees, validator rewards
Single point of failure	Yes (bank outage, fraud)	No (distributed network)
Regulatory status	Fully regulated	Evolving, fragmented

The pattern to notice

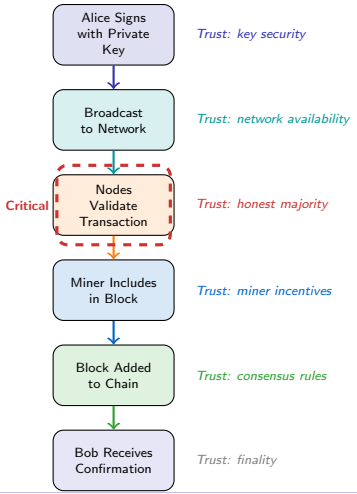
The table looks like a clear win for blockchain. But look at what is missing from the comparison:

- **Consumer protection:** When your bank makes an error, you have legal recourse. When a smart contract executes incorrectly, there is no one to call.
- **Reversibility:** Banks can reverse fraudulent transactions. Blockchain transactions are final – if you send to the wrong address, the money is gone.
- **Identity:** Banks know their customers (KYC). Public blockchains are pseudonymous – good for privacy, problematic for compliance.
- **Accountability:** When a bank fails, deposit insurance covers you. When a DeFi protocol fails, there is no safety net.

Key insight: Blockchain replaces one set of trust requirements with another. The question is which set of risks you prefer.

Blockchain replaces institutional trust with algorithmic trust – but algorithmic trust has its own failure modes.

Follow One Bitcoin from Alice to Bob – Where Does Trust Enter?



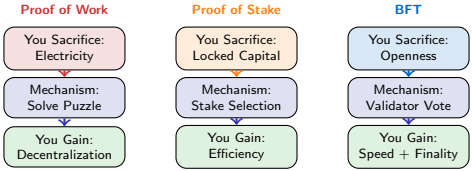
Six steps, six trust assumptions

- **Private key:** Alice trusts that her key is secure. If her wallet is compromised, the transaction is unauthorized but irreversible.
- **Network broadcast:** Alice trusts the peer-to-peer network will propagate her transaction. Network partitions can delay or isolate it.
- **Validation:** The network checks Alice's balance and signature. This is where the honest majority assumption matters – if attackers control over 50% of hash power, they can reject valid transactions.
- **Mining:** A miner selects Alice's transaction from the mempool. Miners prioritize higher fees – incentive alignment, not altruism.
- **Chain inclusion:** The block is added. But "confirmation" is probabilistic – a deeper chain could theoretically overwrite it.
- **Finality:** After six confirmations (approximately one hour), reversal becomes economically impractical. Not impossible – impractical.

Key insight: At no point is trust absent. It is distributed across cryptography, incentives, and probability.

Six steps, six trust assumptions. "Trustless" does not mean trust-free – it means the trust is distributed, not centralized.

What Do You Sacrifice to Get Consensus – Energy, Capital, or Control?



Every mechanism trades off decentralization, security, and scalability.

Three mechanisms, three currencies

- **Proof of Work:** Miners spend electricity to solve computational puzzles. Security comes from making cheating more expensive than honest participation. Cost: enormous energy consumption. Gain: maximum decentralization – anyone with hardware can participate.
- **Proof of Stake:** Validators lock capital as collateral. Misbehavior triggers slashing (lose your stake). Cost: capital lockup, concentration risk (rich get richer). Gain: 99.95% less energy than PoW.
- **BFT (Byzantine Fault Tolerance):** Known validators vote on blocks. Fast and final, but requires knowing who the validators are. Cost: centralization (limited, permissioned set). Gain: instant finality, high throughput.

The no-free-lunch principle

No mechanism excels on all dimensions. Public blockchains choose decentralization + security (sacrifice speed). Private blockchains choose speed + security (sacrifice decentralization). The design choice is a statement about which trust trade-off you accept.

No consensus mechanism is free. Each pays for security with a different currency – energy, capital, or centralization.

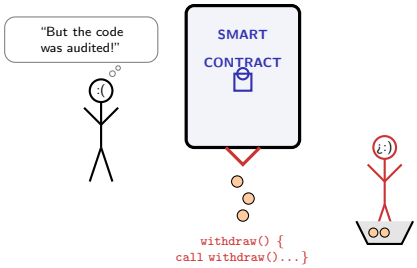
What Happens When the Code Is the Contract – and the Code Has a Bug?

When Code Is Law, Bugs Are Permanent
Smart contracts are immutable. Once deployed, they cannot be patched, updated, or rolled back. If the code contains a vulnerability, the vulnerability is permanent – and anyone can exploit it.

What can go wrong:

- **Reentrancy attacks:** A contract calls an external function that calls back into the original contract before it finishes – draining funds in a loop
- **Oracle manipulation:** Smart contracts cannot see the real world. They rely on oracles (data feeds) that can be manipulated, corrupted, or delayed
- **Governance failures:** When code fails, who decides the fix? The community that hard-forked Ethereum after a major hack split into two competing chains – neither "trustless"
- **Key management:** Lose your private key, lose your assets. No customer service, no password reset, no recourse

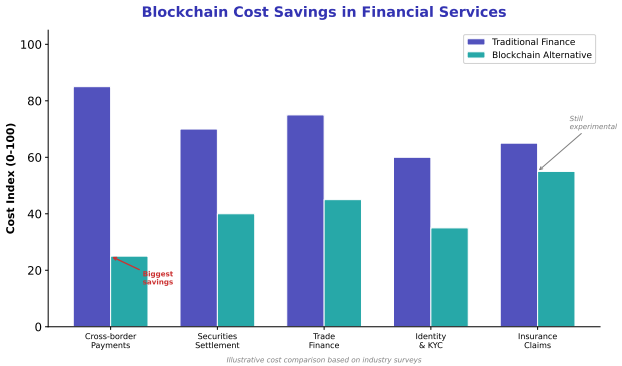
The fundamental irony: Blockchain was built to eliminate the need to trust institutions. But when things go wrong, users desperately wish for an institution that could reverse the damage.



The code worked exactly as written.
The problem was the design.

The code worked exactly as written. The problem was that "as written" and "as intended" were not the same thing.

Where Does Blockchain Actually Save Money in Finance?



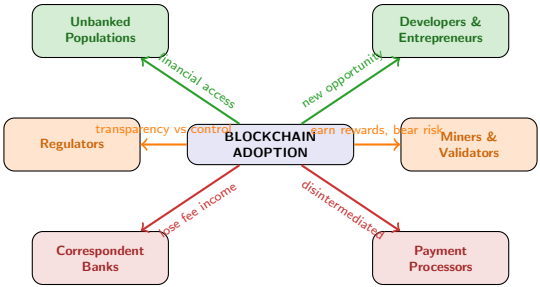
Cost savings by use case

- **Cross-border payments lead:** Correspondent banking charges 6–8% and takes 2–5 days. Direct blockchain settlement cuts intermediary layers – the biggest cost savings anywhere in finance
- **Securities settlement benefits:** Moving from T+2 to near-instant reduces counterparty risk and frees capital tied up during settlement
- **Trade finance gains:** Digitizing letters of credit on shared ledgers eliminates document handling, verification delays, and fraud risk
- **Identity/KYC improves:** Shared identity verification reduces duplication across institutions – but privacy concerns remain unresolved
- **Insurance lags behind:** Claims processing via smart contracts is mostly experimental – the oracle problem makes real-world event verification unreliable at scale

The pattern: Blockchain saves the most where trust costs are highest – cross-border transactions with multiple intermediaries. Where trust costs are low (single-institution processes), blockchain adds overhead without sufficient benefit.

Illustrative cost comparison based on industry surveys. Blockchain saves most where intermediation layers are thickest.

Who Wins and Who Loses When You Remove the Middleman?



Winners

+ **Unbanked populations:** Direct access to financial services without needing a bank account. Particularly impactful for remittances and cross-border transfers.
+ **Developers and entrepreneurs:** Open, permissionless platforms for building financial applications without negotiating with banks.

Losers

– **Correspondent banks:** Blockchain-based settlement bypasses the correspondent banking network entirely – the primary fee source for cross-border transactions.
– **Payment processors:** Peer-to-peer payments reduce the need for centralized payment rails and their associated fees.

Mixed impact

~ **Regulators:** Gain transparency (public ledgers) but lose control (pseudonymous actors, cross-border jurisdiction gaps).
~ **Miners and validators:** Earn rewards for securing the network, but face regulatory uncertainty and hardware/capital risk.

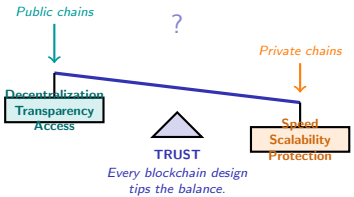
Disintermediation does not eliminate middlemen – it replaces old ones (banks) with new ones (validators, exchanges, oracle providers).

The Trust Spectrum: Every Blockchain Design Tips the Balance

The Blockchain Evaluation Framework
Before accepting or rejecting any blockchain proposal, ask:

1. **Where does the trust actually go?**
   Blockchain does not eliminate trust – it moves it. From banks to code. From regulators to validators. From legal recourse to cryptographic finality. Identify where trust lands in this specific design.
2. **What is the cost of that trust shift?**
   Every trust mechanism has a cost. PoW costs energy. PoS costs capital lockup. Immutability costs the ability to fix mistakes. Is the cost justified by the problem being solved?
3. **What happens when it fails?**
   Banks have deposit insurance, courts, and regulators. What is the fallback when a smart contract has a bug, when a validator set is compromised, or when a user loses their private key?

The trust paradox: Blockchain is most useful where trust costs are highest – but that is precisely where the consequences of trusting the wrong system are most severe.



The right question is not "Should we use blockchain?" – it is "Where does this design place trust, and are we comfortable with that?"

Your Challenge: Evaluate a Blockchain Proposal

Mini-Challenge (15 minutes)

A consortium of five European banks is considering a blockchain-based system for securities settlement. Currently, settlement takes T+2 (two business days). The consortium claims blockchain will reduce this to near-instant, saving capital costs and reducing counterparty risk.

Your deliverable: Apply the three evaluation questions from the previous slide to this proposal:

1. **Where does the trust go?**
   - Currently, trust sits with the central clearinghouse. Where would it move in the blockchain design?
   - Who are the validators? The same five banks? External parties?
   - Is this really decentralized, or is it a shared database with extra steps?
2. **What is the cost of the trust shift?**
   - What consensus mechanism would they use? What does it cost?
   - Would the blockchain be public, private, or consortium?
   - Does the speed gain justify the implementation cost?
3. **What happens when it fails?**
   - If a smart contract settles incorrectly, who reverses it?
   - What is the legal status of blockchain-based settlement?
   - What is the fallback if the system goes down?

Discuss with your neighbor: Does this proposal genuinely need a blockchain, or would a shared database achieve the same result?

The best way to understand blockchain's value is to evaluate a specific proposal – not in theory, but against the trust trade-offs it actually makes.