



TECHNICAL DEEP DIVE

Blockchains' Biggest Technical Challenges

And How Teams Work Around Them

Blockchain technology promises a decentralised future, yet beneath the surface lies a web of engineering hurdles that continue to test even the most seasoned development teams. From throughput bottlenecks to the delicate balance of security and privacy, this presentation unpacks the core challenges slowing decentralised networks – and the clever workarounds engineers are deploying to keep the chain moving.

The "Invisible Bottleneck" Behind Most Failures

When blockchain networks fail to scale, the root cause is rarely a single flaw. Instead, three interlocking bottlenecks conspire to throttle performance just as adoption begins to accelerate. Understanding where congestion originates is the first step toward building systems that can actually keep pace with global demand.

Consensus

Every node must agree on the state of the chain. As transaction volume rises, the computational and communication overhead of reaching agreement grows, creating a fundamental ceiling on how fast the network can finalise blocks.

Execution

Each node must re-execute every transaction to verify correctness. Complex smart contract computations consume gas and time, serialising work that could otherwise proceed in parallel, limiting real-world throughput.

Storage

The ledger grows monotonically. Storing years of historical state data imposes ever-increasing hardware burdens on validators, raising the barrier to participation and gradually eroding the network's decentralisation.

CHALLENGE 1

Scalability: When Throughput Collides With Decentralisation

The most cited challenge in blockchain engineering is the gap between decentralised throughput and what the world actually demands. Bitcoin's roughly 7 transactions per second is dwarfed by Visa's capacity for thousands, a gap that becomes painfully visible whenever network demand surges.

The Trilemma Effect

As demand increases, latency rises and fees climb. More transactions competing for limited block space means users bid up gas prices, pricing out smaller participants and degrading the user experience precisely when the network needs to perform.

Why It's Hard

Consensus layer work grows in lockstep with transaction load. Every node must process, validate, and store each transaction – so simply adding more participants does not increase throughput. In fact, a larger network often processes transactions more slowly, because coordination overhead per block rises with the validator count.

Performance: Congestion Becomes a Fee War

When a blockchain gets busy, users discover a harsh reality: block space is auctioned in real time. Peak-time congestion on Ethereum dApps has repeatedly slowed confirmations and sent fees soaring, turning what should be a seamless experience into a bidding war where only the wealthiest transactions make it into the next block.

Confirmation Latency

During periods of high demand, transactions wait in the mempool longer for inclusion.

For time-sensitive dApps such as decentralised exchanges or on-chain gaming, delayed finality can mean missed opportunities or arbitrage losses.

Fee Volatility

Gas prices swing dramatically with network load. The same operation might cost cents one hour and dollars the next, making it nearly impossible for developers to offer users predictable costs or for businesses to budget around on-chain interactions.

The Dual Front

Scaling efforts must fight on two fronts simultaneously: improving confirmation speed so transactions settle quickly, and increasing throughput so more transactions fit per block. Progress on one axis without the other still leaves the network vulnerable to congestion-driven meltdowns.

Security & Privacy: The DAO Moment and What Followed

The 2016 DAO hack – in which roughly \$50 million was siphoned through a re-entrancy flaw in a smart contract – remains a defining cautionary tale. It proved that code running on an immutable ledger carries existential risk: once deployed, bugs cannot simply be patched away without contentious hard forks.

Smart Contract Vulnerabilities

A single overlooked flaw can drain millions. Formal verification and rigorous auditing are essential, yet even well-audited code can harbour edge cases that adversaries exploit under real-world conditions.

Adversarial Networks

Validators operate in hostile environments. They must handle worst-case scheduling, Byzantine actors, and network partitions – all while maintaining safety guarantees. Consensus protocols are only as robust as their weakest assumption under attack.

The Privacy Paradox

Blockchains demand transparency for trust, yet users need confidentiality. Zero-knowledge proofs offer a path forward, but integrating ZK features adds significant cryptographic complexity and increases the surface area for subtle implementation errors.

The Engineering Playbook: Rollups, Sharding, and Validator Limits

Engineers have developed a layered toolkit to push past these bottlenecks. Each approach makes a deliberate trade-off, accepting centralisation in one dimension to gain performance in another – while preserving enough decentralisation to keep the network trustworthy.

1

Constrain Validator Sets

Limiting the number of validators reduces consensus overhead and speeds finality, but concentrate control among fewer participants, eroding censorship resistance.

2

Sharding

Splitting the chain into parallel partitions allows each shard to process transactions independently, multiplying aggregate throughput – but cross-shard communication remains a famously hard problem.

3

Rollups

Rollups piggyback on Layer 1 for data availability and liveness while centralising execution to a sequencer. This achieves high throughput without dumping security onto a separate, weaker consensus.

- ❑ **The Core Trade-off:** Every scaling decision is a negotiation among scalability, security, and decentralisation. No single solution eliminates the tension – the art lies in choosing which corner of the triangle to compromise and by how much.