

Production Case Study — Fixing a Critical Blockchain Transaction

Problem

In a live fintech system serving more than one million users, blockchain transactions started failing randomly. The backend APIs were stable, RPC provider health looked normal, and the network seemed healthy, but the transactions still failed intermittently.

Root Cause

After improving logs, the actual issue became clear: nonce mismatch, nonce too low, and replacement transaction errors. The root cause was nonce desynchronization in the blockchain wallet system.

What Happened

Multiple background workers, retry systems, and admin cancellation flows were processing transactions at the same time. This caused race conditions and conflicting nonces. In simple terms, several transactions attempted to use the same sequence number and the blockchain rejected the conflict.

Fix

The fix was a centralized Redis-based nonce manager. Redis became the single source of truth, with atomic increment behavior ensuring each worker received a safe, unique nonce. The flow became worker -> Redis -> RPC -> blockchain.

Guiding Conclusion

In distributed fintech systems, even a concept as simple as a sequence number can become a critical production risk if state is not managed centrally. Logging and centralized state control are essential to resilient engineering.