



RPC Trust Failures: A Checklist

Your contracts are audited. Is the data they act on?

DeFi Security Summit
Mumbai · 2 November 2026
Jork Leonhardt · corpus.core

Every DeFi system reads the chain through RPC endpoints, and most take the answers on trust. Use these questions to review your own stack. None of them needs a code change to answer.

PART A · FOR PROTOCOL ARCHITECTS

Four questions about your stack

1 How many independent sources stand behind each critical read?

RED FLAG One endpoint, or several endpoints from the same provider or cloud.

GOOD Value-moving reads come from independent operators, and you know which path each read takes.

- ☐ List every read that can move value: prices, balances, cross-chain messages, finality.
- ☐ Name the operator, cloud and region behind each endpoint. Endpoints that share an operator or cloud count as one.

2 Is any response verified, or only trusted?

RED FLAG `eth_call` and `eth_getLogs` results are used at face value.

GOOD Responses are checked against Merkle proofs (`eth_getProof`) anchored to a header your system verified against consensus.

- ☐ Mark which value-moving reads are verified with proofs, and which are only trusted.
- ☐ Check where the anchoring block header comes from. A header from the same RPC proves nothing.
- ☐ Make sure a failed proof stops the action instead of logging a warning.

3 What happens when sources disagree or disappear?

RED FLAG The first answer wins, or the system silently fails over to whatever is still reachable.

GOOD Disagreement pauses the action and raises an alert; failover never drops below a defined quorum.

- ☐ Write down the quorum for each critical read.
- ☐ Take endpoints offline one by one in a test environment and watch what the system does.
- ☐ Alert on every failover, not only on full outages.

4 How concentrated is your provider and jurisdiction exposure?

RED FLAG One company, one cloud or one jurisdiction behind most reads, including wallet defaults.

GOOD A written map of who serves which reads, where, under which law and what they log, plus a switch that needs no deploy.

- ☐ Map providers, hosting and jurisdiction for all reads, including the defaults in the wallets your users bring.
- ☐ Read each provider's logging and retention policy.
- ☐ Make a provider change a configuration switch, not a deploy.

PART B · FOR MONITORING AND TOOLING PROVIDERS

Two gaps in today's detection

1 Response integrity: nobody checks what each component was told by its RPC

TODAY Anomaly detection watches on-chain transactions. The KelpDAO exploit looked like normal bridge activity at the transaction level.

- ☐ Compare the data a component acted on with independently verified state.
- ☐ Watch from vantage points an attacker cannot target or poison.
- ☐ Use cross-chain invariants (burns against releases) as a backstop; they still fire after the funds have moved.

2 Concentration: protocol teams cannot see how their reads depend on providers

TODAY No dashboard shows what share of reads goes through one provider, or when failover kicks in.

- ☐ Keep an inventory of RPC dependencies per component.
- ☐ Alert on failover events and loss of quorum.
- ☐ Treat attacks on a provider as a signal for every protocol that depends on it.

BACKGROUND

Four documented RPC failures

When	Failure	What happened	Lesson
Nov 2020	Availability	Infura components pinned to older Geth versions hit a consensus bug and ended up on a minority chain. MetaMask broke for many users; Binance, Bithumb and others paused ETH and ERC-20 withdrawals. The mainnet API was down for about six and a half hours.	A provider can serve a different chain, and clients cannot tell.
Mar 2022	Censorship	Infura applied new sanctions settings more broadly than intended. Users in Venezuela lost MetaMask access for several hours; sanctioned regions such as Iran stayed blocked by design.	One provider's compliance settings decide who can transact.
Nov 2022	Privacy	Consensys disclosed that Infura, MetaMask's default RPC, collects IP and wallet addresses when users send transactions. It later said it does not store them for reads, keeps them separately and would narrow retention to seven days.	Whoever broadcasts your users' transactions sees where they come from.
Apr 2026	Trust	Attackers poisoned two RPC nodes used by the single verifier (1-of-1 DVN) of KelpDAO's LayerZero bridge and knocked the remaining providers offline with a DDoS. A forged message released 116,500 rsETH, about \$292M. No smart contract bug has been identified.	Audited code acting on unverified data is an unaudited system.

Deep dive: the KelpDAO exploit

How six weeks of access, a DDoS and one forged message got past audited contracts, and why more verifiers and more RPC providers leave the trust problem in place.

[The Code Was Fine. That's the Problem. A Post-Mortem on the KelpDAO Exploit \(corpus.core\)](#)

WHAT TO DO ABOUT IT

Ways to verify instead of trust

- **State proofs.** `eth_getProof` (EIP-1186) returns Merkle proofs for account and storage values; receipt and transaction proofs cover events.
- **An independent consensus anchor.** Verify block headers against the beacon chain sync committee with a light client, never against the RPC that delivered the data.
- **Redundancy with a quorum.** Independent operators, a defined quorum, and no silent failover. This covers availability and most censorship cases; it does not remove the trust in each source.
- **Light clients** combine the first two points, as long as they verify every response they pass on. One implementation is `colibri.stateless`, a stateless client by corpus.core that verifies RPC responses with proofs ([repository.colibri-stateless.tech](#)).

SOURCES

Selected references

- LayerZero Labs, [KelpDAO Incident Report](#), 18 May 2026
- Chainalysis, [Inside the KelpDAO bridge exploit](#), April 2026
- OpenZeppelin, [Lessons From the rsETH Bridge Exploit](#), April 2026
- Messari, [The rsETH Exploit: Timeline](#), April 2026
- Infura, [Status page: ETH1 Mainnet API service degradation](#), 11 November 2020
- Decrypt, [MetaMask and Infura cut off users in certain sanctioned countries](#), 3 March 2022
- Consensys, [Data retention update](#), December 2022

Your contracts verify signatures. **Who verifies the data?**