Honest Audit: How Decentralized Is This Really?

What the Server ACTUALLY Does (The "Dumb Server" Test)

Server Role	Current Status	Verdict
Decides truth?	No. Math formula decides: `Score = (Σw_true) / (Σw_true + Σw_false) × 100`	✅ PASS
Can admin change scores?	No. Finalized scores are INSERT-only, never UPDATE. Reputation is a pure function of vote history.	✅ PASS
Can admin delete rumors?	No. Only the creator (proven via Ed25519 signature) can delete.	✅ PASS
Can admin ban users?	No. There's no ban mechanism at all. No admin endpoints.	✅ PASS
Can admin forge votes?	No. Every vote requires a valid Ed25519 signature from the voter's private key, which only the user has.	✅ PASS
Is truth auditable?	Yes. Public `/api/audit/log` with cryptographic hashes of every action. Anyone can verify.	✅ PASS

The Prompt Requirements — Line by Line

Requirement	Status	How
No central server/admin controls truth	✅ PASS	Server executes deterministic math. No admin endpoints exist. Score = weighted vote ratio.
Anonymous students verify/dispute	✅ PASS	Ed25519 keypairs, no PII, votes are TRUE/FALSE
Trust scores through methods YOU design	✅ PASS	Reputation-weighted voting with exponential growth/decay + recency weighting + tanh normalization
Prevent same person voting multiple times WITHOUT collecting identities	✅ PASS	`PRIMARY KEY(rumor_id, voter_public_key)` — one vote per key. PoW + probation make account creation expensive.
Popular false rumors shouldn't auto-win	✅ PASS	Votes weighted by reputation, not count. 10 new bots with rep=0 each get weight=1. One veteran with rep=50 outweighs all 10.
Verified facts changing scores mysteriously	✅ PASS	Once deadline passes → `finalized_scores` table (INSERT only, never updated). Scores are frozen forever.
Bot accounts manipulating votes	✅ PASS	3-layer defense: Puzzle CAPTCHA → PoW (difficulty 4, escalates to 5) → probation period → rep starts at 0
Deleted rumors affecting trust scores	✅ PASS	Reputation only uses `JOIN finalized_scores` — deleted rumors have no finalized_scores entry, so they're automatically excluded. Cache invalidated on deletion.
Prove system can't be gamed by coordinated liars	⚠️ PARTIAL	Mathematically resistant (see below) but no formal proof in code/docs.
Can't centrally control who participates	✅ PASS	No ban/block mechanism. Anyone with compute power can join.

Game-Theory Resistance (The Math Argument)

The system resists coordinated attacks through three interlocking mechanisms:

1. Cost of Entry

Each account requires PoW computation (~30-60 seconds). Creating n bot accounts costs **O(n · 2^d)** where d = difficulty.

2. Reputation Bootstrapping Problem

- New accounts have `rep = 0`, so their vote weight = 1 (minimum)
- An attacker controlling k fresh accounts gets total weight = k
- Meanwhile, one honest veteran with reputation R gets weight R
- **The attacker needs k > R accounts to outweigh a single veteran**

3. Self-Correcting Feedback

- If bots vote incorrectly (against eventual outcome), their reputation drops to `rep × 0.85` per wrong vote
- Honest users gain `rep × 1.15 + 0.15 × recency`
- **Over time, the gap widens exponentially**

However, there's currently no formal mathematical proof written in the docs. The README describes the challenge but doesn't include the proof. This should be added.

What's Still Centralized (Honest Gaps)

Gap	Severity	Notes
Server is single point of failure	Medium	If server goes down, everything stops. True decentralization would need P2P or federation — but the prompt says "without blockchain", and the server is execution-only, not decision-making.
Database is centralized storage	Medium	Same — but data is auditable and the server never makes subjective decisions. It's a "dumb relay + calculator".
IP hash stored for rate limiting	Low	Rotated daily, SHA-256 hashed, only first 16 chars. Can't be reversed. Used only for bot detection, not for banning.
Server clock trusted for deadlines (FR8.3)	Low	FR8.3 calls for "distributed consensus time". We use server `NOW()`. In practice this is fine since server can't benefit from clock manipulation — deadlines are set at creation time and are immutable.

Bottom Line

The server is genuinely a **"dumb executor of transparent rules."** It can't:

- Decide what's true or false
- Ban anyone
- Change scores
- Forge votes
- Override deadlines
- Delete anyone else's content

Every action is **cryptographically signed** by the user and logged in the public audit trail. The algorithm is **deterministic** — given the same vote data, anyone can independently verify every trust score and reputation value.

Final Assessment

Is it as decentralized as a blockchain?

No — it uses a central server for storage and computation.

But that's exactly the point of the prompt:

Prove you can achieve these properties WITHOUT blockchain.

The answer is:

Yes, through:

- Cryptographic signatures
- Deterministic algorithms
- Public auditability
- Zero admin control

The system successfully demonstrates **"decentralized truth-finding"** without blockchain infrastructure. The server is simply infrastructure — like using GitHub for code hosting doesn't centralize decision-making in open-source projects. The control is cryptographic, not administrative.