ZK Voting System

Whitepaper

Parliamentary Governance for DAOs, with Privacy as Phase 2

Version:	1.1
Author:	Tyler Delano + Dexter
Status:	Phase 1 live on Sepolia, ZK deferred to Phase 2
Frontend:	https://zk-voting-system-two.vercel.app
Live Contract:	0xF844B2B37f34Dc53b79AAd0bc657C508e628dbad

May 1, 2026

Contents

1 Executive Summary

ZK Voting System is a governance prototype for DAOs that combines a Robert's Rules-style parliamentary process with a planned zero-knowledge privacy layer.

The live hackathon demo focuses on **Phase 1** only:

- a chair-managed voter allowlist
- proposal creation and seconding
- timed voting with yes / no / abstain
- on-chain finalization and verification

The zero-knowledge portion is **not live in the current demo**. It remains part of the architecture and roadmap, but was deferred because the Circom / snarkjs proving toolchain was not stable enough to ship confidently on the hackathon timeline.

That tradeoff was intentional. The current demo shows a working governance flow first, instead of presenting a privacy system that is not reliable enough to trust.

2 What the Live Demo Actually Shows

The current live system on Sepolia demonstrates:

1. proposal creation by an eligible voter
2. seconding by another eligible voter, or chair-led fast-track for demo flow
3. timed voting with yes / no / abstain choices
4. on-chain tallying of votes
5. proposal finalization and verification through the frontend and the blockchain

The live demo does **not** currently provide:

- anonymous voting
- private voter identity proofs
- hidden vote choices
- production-grade decentralization or governance hardening

This is a real on-chain parliamentary governance prototype, but it is still a prototype.

3 Problem

Most on-chain voting systems are too simple for real governance.

They typically reduce governance to a basic token-weighted yes/no poll and ignore the procedural structure that real deliberative groups use to make decisions. They also make voting fully transparent by default, which can discourage honest participation on controversial issues.

The problem this project addresses has two parts.

3.1 Governance quality

DAOs often lack motion discipline, seconding requirements, amendment handling, reconsideration rules, and other process controls that help meetings produce legitimate outcomes.

3.2 Voter privacy

On most public chains, a vote is visible to anyone who can inspect the ledger. That makes it hard to support sensitive governance decisions where privacy matters.

4 Project Thesis

The thesis of this project is:

> DAOs need both better parliamentary process and better privacy, but the process layer should work on its own before the privacy layer is added.

That leads to a two-phase architecture:

- **Phase 1:** ship a functioning Rob's Rules-inspired governance system on-chain
- **Phase 2:** add zero-knowledge privacy to hide voter identity and vote choice while preserving eligibility and anti-double-voting guarantees

5 System Architecture

5.1 Phase 1 — RobRulesVoting (Live)

The live contract is RobRulesVoting, deployed on Sepolia.

It supports:

- chair-managed voter eligibility
- proposal creation
- seconding
- amendment submission
- opening voting windows
- fast-track voting for demo use

- division calls
- reconsideration flow
- finalization

This phase uses standard wallet-based interactions and records votes directly on-chain.

5.2 Phase 2 — ZK Privacy Layer (Planned)

The planned privacy layer replaces direct vote submission with a proof that shows:

- the voter is eligible
- the vote choice is valid
- the vote has not already been cast for that proposal

The intended result is a voting system where the chain verifies the legitimacy of the vote without learning who cast it or how they voted.

This phase is **not active in the current demo**.

6 Why ZK Is Not Live Yet

The project originally aimed to include browser-based ZK proof generation in the demo.

That was cut for a concrete technical reason, not for presentation reasons:

- the Circom 2.2.x and snarkjs toolchain introduced compatibility problems in the proving flow
- the circuit artifacts required for a stable browser proof path could not be regenerated with confidence in the available timeframe
- shipping a brittle ZK flow would have made the demo less credible, not more

So the current system keeps the ZK design in scope, but does not pretend it is already working end-to-end.

7 Parliamentary Model

The contract implements a simplified Robert's Rules-inspired flow.

7.1 Core lifecycle

Created → Seconded → Voting → Passed / Failed

7.2 Supported governance actions

Action	Purpose
createProposal()	Start a motion
secondProposal()	Require another member to support moving forward
submitAmendment()	Allow proposed changes before final vote
approveAmendment()	Chair approval for amendment flow
openVoting()	Begin timed voting after seconding
fastTrackVoting()	Chair shortcut for demo flow
castVote()	Record yes / no / abstain vote
callForDivision()	Demand a more formal recorded vote signal
reconsider()	Request reconsideration from the prevailing side
reopenVoting()	Reopen voting after reconsideration
finalizeProposal()	Close the motion on-chain

This is not a complete encoding of parliamentary law. It is a practical governance prototype designed to prove that better process can exist on-chain without reducing everything to a raw poll.

8 What Is Live vs. What Is Roadmap

8.1 Live now

- Sepolia deployment
- frontend at https://zk-voting-system-two.vercel.app
- chair dashboard
- voter portal
- proposal verification page
- Rob's Rules-inspired motion flow

8.2 Planned next

- zero-knowledge vote casting
- anonymous eligibility proofs
- hidden vote choices
- stronger production controls and governance hardening
- potentially Layer 2 deployment for cheaper execution

9 Demo Trust Model

The live demo is intentionally centralized in a few places because it is a hackathon prototype.

Current trust assumptions

- the chair manages the allowlist
- the chair has strong operational control over the live demo flow
- the system is not yet governed by a multisig, timelock, or full production review process

This is acceptable for a prototype, but it is not the final governance model.

10 Security and Production Readiness

The live build is suitable for a controlled demo, but it is **not production-ready governance infrastructure**.

Key reasons:

- the live system is still evolving quickly
- the privacy layer is not active yet
- production-grade governance controls are not fully in place
- the project has not gone through a formal external smart contract audit

Recent hardening work improved contract edge cases and test coverage, but production deployment would still require:

- a broader audit
- governance minimization or multisig controls
- operational monitoring
- stronger deployment and upgrade discipline

11 Economic Model

11.1 Phase 1

Users pay normal transaction gas costs for on-chain actions.

11.2 Phase 2

If ZK proving returns, the system may need:

- client-side proving optimization

- relayer support for better UX
- Layer 2 deployment for lower execution cost

No token economics are required for the current hackathon demo.

12 Current Live Deployment

- **Contract:** RobRulesVoting
- **Network:** Ethereum Sepolia
- **Address:** 0xF844B2B37f34Dc53b79AAd0bc657C508e628dbad
- **Frontend:** https://zk-voting-system-two.vercel.app
- **Chair Wallet:** 0x6A8C66fBAA1fE05947CfBD54b2fCF67ca3c254e0

13 Conclusion

This project does not claim to have solved private DAO voting today.

What it does show, honestly, is this:

1. a working on-chain parliamentary governance flow can be built and demonstrated now
2. the privacy layer can be designed cleanly on top of that process model
3. it is better to ship a truthful prototype than a fake end-to-end ZK story

The current demo is a functioning governance prototype with a credible path toward private voting, not a finished private voting platform.