

Decision Rights Matrix

The Decision Rights Matrix translates the decision classes declared in c9-governance-purpose-charter into an unambiguous assignment of who proposes, who votes, who executes, who may veto, and who is held accountable for each class. A governance structure without explicit decision rights produces either paralysis (everyone must approve everything) or capture (whoever acts first wins). This matrix closes both failure modes: every decision class has exactly one accountable party, one execution authority, and a defined set of veto holders. The contract-change decision class carries special meta-governance status per C11 — it is the only authorized path to a MAJOR amendment of any substrate file or interface contract.

1. Matrix Structure

Each row covers one decision class. The five rights columns use the following role semantics:

Column	Definition
Proposer	Who may submit a formal proposal. May be constrained by eligibility rules (token threshold, role, working-group membership).
Voter	Who casts binding votes. May be a token-weighted set, a committee, or a multi-class structure.
Executor	Who carries out the passed decision. Distinct from the voter; execution authority does not default to the winning faction.
Vetoer	Who, if anyone, may block execution of a passed proposal. Veto must be exercised within the stated window and logged.
Accountable	Who is named in the C11 provenance record as responsible for the outcome, even if the decision was collective.

2. Decision Rights by Class

2a. Full Matrix

Decision Class	Proposer	Voter	Executor	Vetoer	Accountable	Approval Rule	Veto Window
treasury-allocation	Any token holder meeting minimum stake	All token holders, weighted	Treasurer role	Founding Steward Council (unanimous, within 48 hrs)	Treasurer	Simple majority; quorum required	48 hours post-pass
contract-change	Steward Council member OR working-group lead with council endorsement	All token holders, weighted	Steward designated by proposal	Founding Steward (single veto, 7-day window)	Designated Steward	Supermajority ($\geq 67\%$); quorum required; 7-day review period	7 days post-pass
working-group-character	Any token holder meeting minimum stake	All token holders, weighted	Working Group Lead	None	Working Group Lead	Simple majority; quorum required	None
membership-change	Steward Council	All token holders, weighted	Membership Steward	Founding Steward (single veto, 48 hrs)	Membership Steward	Supermajority ($\geq 67\%$); quorum required	48 hours post-pass
parameter-update	Steward Council only	All token holders, weighted	Technical Lead	Founding Steward (single veto, 7-day window)	Technical Lead	Supermajority ($\geq 67\%$); quorum required; 14-day advance notice	7 days post-pass
retroactive-reward	Working Group Lead or direct contributor (no self-vote)	All token holders excluding nominee	Treasurer	None	Treasurer	Simple majority; quorum required; no self-vote	None
platform-partnership	Founding Steward or Technical Lead	Token holders (ratification) + Board (gate)	Designated Partnership Lead	Board unanimously within 72 hrs	Founding Steward	Board must not block AND simple majority ratification	72 hours post-pass

Decision Class	Proposer	Voter	Executor	Vetoer	Accountable	Approval Rule	Veto Window
emergency-ratification	Accountable Steward only (post-action)	Steward Council	Accountable Steward (action already taken)	None	Accountable Steward	Majority of Steward Council	24 hours post-submission

2b. Voting-Weight Model

Token voting weight is linear by default: one governance token equals one vote. Alternative weighting models (quadratic, reputation-weighted, delegated) may be adopted only through a parameter-update proposal. Until such a proposal passes, linear weighting governs.

Delegation: A token holder may delegate their vote to another participant for a defined period. Delegation is recorded on-chain or in a verifiable off-chain registry before the voting period opens. A delegate may not sub-delegate without the original holder's explicit re-instruction.

3. Contract-Change Decision Class — Full Specification

The contract-change class governs all MAJOR amendments to substrate files and interface contracts (C1–C12). Its special treatment reflects the meta-governance rule in c9-governance-purpose-charter §4 and the C11 contract.

3a. What Triggers This Class

A contract-change proposal is required whenever a proposed modification meets any of the following conditions:

- Changes the decision_origin enum in glossary.md or interface-contracts.md
- Alters a C1–C12 payload field, emission rule, or change-control rule at MAJOR level
- Amends c9-governance-purpose-charter §2, §3, or §4 (decision scope, exclusions, or meta-governance rule)
- Adds or removes a decision class from this matrix
- Changes any approval rule, veto rule, or quorum threshold in this matrix
- Restructures the s3-provenance-metadata-schema shape in a breaking way

MINOR amendments (adding non-breaking fields, fixing typos without altering rules) do not require a contract-change proposal; they follow the charter maintenance rule in c9-governance-purpose-charter §6.

3b. Contract-Change Proposal Requirements

Beyond the standard proposal template in c9-proposal-design-template, a contract-change proposal **MUST** include:

1. A diff of the exact text being changed, with old and new versions side by side.
2. An impact statement covering every consumer of the affected artifact.
3. A migration plan for any in-flight governance processes that relied on the prior version.
4. Sign-off from the Technical Lead (for schema-touching changes) and Legal/Compliance reviewer (for contract-language changes).
5. A minimum 7-day review period before voting opens, following the 14-day advance notice required for parameter-update.

3c. Meta-Governance Enforcement

The Executor of a passed contract-change proposal MUST emit a C11 provenance record (via c9-execution-accountability) before committing the amendment. The s3-provenance-metadata-schema lineage chain will contain no MAJOR-version amendment without a corresponding {proposal_id, decision_class: "contract-change", outcome: "passed", decision_origin: "governance-vote"} record. Any amendment committed without this record is a governance violation and must be reversed; the Accountable Steward is liable for the correction.

4. Worked Example — PP DAO Treasury Allocation vs. Contract-Change

The following illustrates how the same PP DAO action proceeds under two different decision classes.

Scenario A: Treasury Allocation — STR Revenue Pool to AI Infrastructure

PP DAO holds a shared treasury funded by STR revenue share from Point Preserve bookings. The Technical Lead proposes allocating \$24,000 (illustrative) from the treasury to upgrade the on-site AI inference infrastructure.

Rights Field	Instantiation
Decision class	treasury-allocation
Proposer	Technical Lead (token holder meeting minimum stake)
Voter	All PP DAO token holders, linear-weighted
Executor	Treasurer (Doug Liles in the Treasurer role)
Vetoer	Founding Steward Council, unanimous within 48 hrs
Accountable	Treasurer
Approval rule	Simple majority; quorum per c9-governance-mechanisms-spec

Rights Field	Instantiation
C11 record	{proposal_id: "pp-dao-prop-2026-011", decision_class: "treasury-allocation", outcome: "passed", decision_rights_ref: "c9-decision-rights-matrix#treasury-allocation", evidence_pointer: "s3://prov/pp-dao/prop-2026-011/vote-tally.json", decision_origin: "governance-vote"}

The Treasurer executes the disbursement after the 48-hour veto window closes without exercise, then emits the C11 record before moving funds. Moving funds before emitting the C11 record is an irreversible-impact boundary violation.

Scenario B: Contract-Change — Amending the GROW Library License Terms

A DAO member proposes changing the GROW Library substrate license to add a revenue-share clause affecting all downstream operators.

Rights Field	Instantiation
Decision class	contract-change
Proposer	Steward Council member with council endorsement
Voter	All PP DAO token holders, linear-weighted
Executor	Designated Steward (named in proposal)
Vetoer	Founding Steward (single veto, 7-day window)
Accountable	Designated Steward
Approval rule	Supermajority ($\geq 67\%$); quorum required; 7-day review period
Required extras	Old/new diff, impact on all c9-governance-package consumers, Technical Lead sign-off, 14-day advance notice
C11 record	{proposal_id: "pp-dao-prop-2026-022", decision_class: "contract-change", outcome: "passed", decision_rights_ref: "c9-decision-rights-matrix#contract-change", evidence_pointer: "s3://prov/pp-dao/prop-2026-022/vote-tally.json", decision_origin: "governance-vote"}

If this proposal is rejected or vetoed, the substrate license remains at its current version. No steward may amend it unilaterally.

5. Decision Rights Across the Lifecycle

The matrix governs not just the pass/fail vote but the full lifecycle:

Proposal submitted → eligibility check (Proposer rules) →
review period (minimum per class) → voting opens (Voter rules) →
vote closes → outcome determined (Approval rule) →
veto window (Vetoer rules) → execution authorized (Executor) →
C11 record emitted → Accountable role carries ongoing obligation

Any step attempted out of sequence is a governance violation. The c9-execution-accountability playbook operationalizes the execution and C11-emission steps.

6. Failure Modes in Decision Rights

Failure Mode	Signal	Correct Behavior
Execution without a passed proposal	Tooling has no C11 record for the change	Block execution; require proposal retroactively or reverse the action
Veto exercised outside the window	Veto log timestamp is past the window	Veto is not valid; execution proceeds; the Founding Steward may raise a new contract-change proposal to reverse the action
Self-vote on retroactive-reward	Nominee address in voter set for their own reward	Disqualify the vote; re-tally without the self-vote
Delegation not registered before voting opens	Delegate casts vote without pre-registered delegation	Vote is invalid; token holder must vote directly or abstain
Wrong decision class used	A treasury-allocation vote used to amend a contract	Outcome is void; correct class proposal required

7. Cross-Reference Summary

Downstream artifact	What it reads from this matrix
c9-resource-allocation-framework	treasury-allocation row governs all budget policies
c9-proposal-design-template	Per-class review periods, eligibility, and evidence requirements
c9-governance-mechanisms-spec	Quorum, delegation, and threshold rules apply per-row
c9-execution-accountability	Executor and Accountable columns drive the C11 record

Usage Notes

Lock the decision-class rows before writing any downstream artifact — every row in this matrix is a dependency. Add new decision classes only through a contract-change proposal once the DAO is live. During initial build (pre-launch),

new rows may be added by charter owner with a version bump. Mark all token amounts, allocations, and quorum numbers (illustrative) until the c9-governance-mechanisms-spec is ratified.