Decision Rules

R-SUB: Submission Rules

Submission Rules

These rules govern what constitutes a valid data submission. All three must pass before a submission enters the pending-validation queue.

R-SUB-01 SHACL Validation Required

Every submission must pass the `cth:FieldDataQuality` SHACL shape before entering the validation queue. Submissions that fail are rejected at ingestion with a machine-readable error payload.

Implementation: OPA policy `submit.rego` calls the SHACL validator endpoint synchronously. Rejection codes returned as JSON-LD ValidationReport.

R-SUB-02 FPIC Mandatory for Indigenous Territory

Any parcel whose GPS centroid intersects an officially recognised indigenous territory or Afro-Colombian collective territory requires a valid `cth:FPICCredential` signed by the Community Sovereign before submission is accepted.

Implementation: Postgis `ST_Within()` check against `territorial_boundaries` table at ingestion. Row-level security blocks write if FPIC credential not present in JWT claims.

R-SUB-03 Methodology Declaration Required

Submitter must declare which measurement methodology was used (e.g. KoboToolbox GPS v2.3, Planet NDVI mosaic 2025-Q4, manual tape survey). Methodology ID must resolve to a registered entry in `cth_methodology_registry`.

Implementation: Foreign key constraint on `submissions.methodology_id`. Validator accreditation scope is checked against methodology type.

?? These three rules compose: a submission must pass ALL of R-SUB-01, R-SUB-02, and R-SUB-03 simultaneously. Partial compliance is not accepted.

R-VAL: Validation Rules

Validation Rules

Validation is the act of a credentialed third party signing that data meets a specific standard. These rules prevent conflicts of interest and ensure regulatory-grade assurance.

R-VAL-01 Accredited Validators Only

Only principals holding a valid `cth:ValidatorCredential` may sign validation results. Validator credentials list the specific methodologies and standards the holder is accredited to validate.

Implementation: OPA `validate.rego` checks credential claims. Unsigned or self-signed validation results are rejected at ingestion.

R-VAL-02 No Self-Certification

A Data Submitter may not validate their own submission. The validator DID must differ from the submitter DID. This applies transitively — a validator who is a beneficial owner of the submitting entity is also disqualified.

Implementation: Ledger constraint: `submission.submitter_did ≠ validation.validator_did`. Beneficial ownership conflicts flagged via off-chain KYC check at validator onboarding.

R-VAL-03 CBAM and Article 6 Require Validation

EUDR DDS can be self-declared by the operator (EU Regulation 2023/1115 Article 4). However, CBAM declarations (Article 7) and Article 6.4 carbon credits require third-party validation before a Digital Conformity Credential (DCC) can be issued.

Implementation: Compliance export endpoint checks `validation_status` field. CBAM and Art.6 exports blocked if no signed DCC present.

R-CON: Consent & Benefit Rules

Consent & Benefit Rules

Consent rules govern what happens after data is collected. They protect community sovereignty and ensure benefit flows back to data contributors.

R-CON-01 Revocation Cascades Immediately

When a Community Sovereign revokes FPIC for a territory, all downstream credentials (DCCs, EUDR DDS) that relied on data from that territory are immediately flagged `status: suspended`. Third parties holding those credentials are notified via webhook within 60 seconds.

Implementation: FPIC revocation event triggers `cascade_revoke()` Postgres function. Downstream credential IDs stored in `fpic_dependencies` table. Webhook queue processes within 60s SLA.

R-CON-02 Purpose Limitation Enforced at Runtime

Data may only be used for the purposes declared in the FPIC credential and the submission metadata. An agent or API call requesting data for an undeclared purpose (e.g. using EUDR data for a carbon market without explicit consent) is rejected by OPA.

Implementation: OPA `purpose.rego` compares `request.purpose` claim against `fpic.permitted_purposes[]` array. Rejection logged as PROV-O `wasInvalidatedBy` event.

R-CON-03 Benefit-Sharing Terms in Ledger

Any commercial use of community data (carbon credits, premium certification fees, data licensing) requires a benefit-sharing agreement recorded in the governance ledger before data access is granted. Minimum 20% of net commercial value must flow to contributing community.

Implementation: Benefit-sharing contract hash stored in `benefit_agreements` table. OPA `commercial.rego` blocks commercial data access if no valid agreement present.

?? **Important:** R-CON-01 is the hardest rule in the framework. Revocation can cascade to invalidate export documents that third parties (coffee buyers, EU customs) are relying on. CTH Data Stewards must proactively manage community relationships to avoid surprise revocations.

R-AGT: Agent Rules

Agent Rules

AI agents are first-class participants in this framework. These rules ensure agents are accountable, auditable, and structurally cannot exceed the permissions of the humans who delegate to them.

R-AGT-01 All Agent Actions Logged as PROV-O

Every action taken by an AI agent — read, write, policy evaluation, credential check — must produce a W3C PROV-O provenance record. The record must name: (a) the agent DID, (b) the delegating human DID, (c) the action type, (d) the entity acted upon, (e) timestamp.

Implementation: PROV-O records written to `provenance_log` append-only table synchronously with the action. Failed PROV-O write aborts the action transaction.

R-AGT-02 Policy Check Mandatory Before Every Write

An agent MUST call `POST /policy/evaluate` before any write operation (submission, validation signature, credential issuance, ledger entry). The policy evaluation result must be `allow: true` before the write proceeds. This is enforced at the API gateway level — agents cannot bypass it.

Implementation: API gateway middleware intercepts all write requests. `X-Policy-Token` header must contain a fresh (< 30s) OPA evaluation token. Requests without valid token receive 403.

R-AGT-03 FPIC Block is Absolute

When FPIC is not granted or has been revoked for a territory, no agent action may read or write data for that territory. This is not an OPA rule — it is enforced at Postgres row-level security, below the API layer. There is no override, no emergency bypass, no admin exception.

Implementation: Postgres RLS policy: `fpic_status = 'active'` required on every row. Even CTH Steward credentials do not bypass this. The only way to lift the block is for the Community Sovereign to re-grant FPIC.

?? Agent design principle: an agent inherits the delegating human's permissions — never more. If the human cannot do something, neither can the agent acting on their behalf. Credential delegation is explicit and recorded in the JWT claims.

R-CHG: Change Governance Rules

Change Governance Rules

These rules govern how the framework itself evolves. Stability is a feature — downstream systems (EU customs, carbon registries, AI agents) depend on predictable interfaces.

R-CHG-01 ? Supermajority for Major Changes

Changes to core framework elements — governance model, role definitions, FPIC requirements, compliance mappings, standards layer — require a ? supermajority of the Governance Board. Minor changes (clarifications, bug fixes, new regulatory appendices) require simple majority.

Implementation: Change classification defined in the Framework Versioning Policy (Appendix A). Voting recorded in governance ledger with member DIDs. Results publicly verifiable.

R-CHG-02 Community Panel Structural Veto

The Community Sovereignty Panel holds a structural veto on any change that affects FPIC rules, community data rights, or benefit-sharing requirements. This veto cannot be overridden by any vote of any other body, including a unanimous Governance Board.

Implementation: Veto encoded as a constitutional constraint in the framework charter, not as an OPA rule. Legal enforceability governed by Colombian Law 70/1993 and ILO Convention 169.

R-CHG-03 Emergency Patch Protocol

Critical security or compliance fixes may be deployed by CTH Data Stewards within 48 hours without a board vote. The patch must be (a) narrowly scoped to the security/compliance issue, (b) announced to all board members immediately, (c) ratified or reversed by formal board vote within 30 days.

Implementation: Emergency patch log in governance ledger. Automatic 30-day ratification deadline enforced by scheduled task. Unratified patches auto-revert.

Version Type	Example	Approval Required	Notice Period
Major (X.0.0)	New role type, FPIC rule change	? supermajority + Community Panel	90 days

Version Type	Example	Approval Required	Notice Period
Minor (x.Y.0)	New regulatory mapping, additional standard	Simple majority	30 days
Patch (x.y.Z)	Clarification, typo, bug fix	Steward + 1 board member	7 days
Emergency	Security/compliance critical	Steward (ratify within 30d)	Immediate