

Interface Contracts

The GROW library is wired by twelve locked contracts. Each contract defines a producer artifact, a consumer artifact or domain, the exact payload schema crossing the boundary, and the change-control rule that governs schema evolution. Sub-agents building S1 through S10 code to these contracts; deviation fails the integration check in the builders evaluation gate.

Contracts are versioned by this artifact's semver. A breaking change to any payload is a MAJOR bump and forces re-review of every producer and consumer.

C1 — Reliability → Evaluation

Producer: s1-failure-mode-register (S1, register).

Consumer: s2-eval-test-seed-matrix (S2, matrix).

Payload schema (one row per failure mode):

Field	Type	Allowed values				
failure_id	string	kebab-case, globally unique within the register				
severity	enum	critical \	high \	medium \	low \	info
detectable_signal	string	free text; must describe an observable signal (log line, metric, tool-call pattern, output property)				
expected_behavior	string	free text; the correct behavior the agent should exhibit instead				

Derivation rule: Every row in the s2-eval-test-seed-matrix carries a seeded_from: <failure_id> pointer. The matrix may expand a single failure mode into multiple test seeds, but no test seed may exist without a seeded_from value resolvable in the register.

Change-control: Adding a column is MINOR. Renaming or removing a column, changing the severity enum, or changing failure_id semantics is MAJOR. Severity demotions must cite an incident or eval result; promotions require S1 owner sign-off.

C2 — Reliability → Provenance

Producer: s1-hitl-review-policy (S1, policy) — emits events on overrides, escalations, fallback fires.

Consumer: s3-provenance-metadata-schema (S3, schema) — stores the events.

Payload schema (one record per event):

Field	Type	Allowed values			
event_id	UUID v7	UUID v7 string, unique within the store			
timestamp	ISO-8601 datetime with timezone	UTC required for storage			
agent_id	string	stable identifier of the agent instance			
decision_origin	enum	agent \	human-override \	fallback \	escalation
evidence_pointer	string	URI or storage key resolving to retained evidence (log slice, tool trace, document hash)			
rationale	string	free text; human-readable reason captured at decision time			

Emission rule: Every override, escalation, and fallback fire produces exactly one provenance record. Drops are gate-failing.

Change-control: Adding fields is MINOR. Changing the decision_origin enum, or changing event_id/timestamp semantics, is MAJOR. Field deprecation requires a one-version overlap.

C3 — Evaluation → Provenance

Producer: s2-audit-trail-schema (S2, spec) — every eval run.

Consumer: s3-provenance-metadata-schema (S3, schema) — eval-run log table.

Payload schema (one record per eval run):

Field	Type	Notes
inputs	object	full input payload to the agent, including system prompt hash
outputs	object	full agent output, including refusals and partials
tool_calls	list[object]	each: {name, arguments, returns, latency_ms} in call order
retrieval_sources	list[object]	each: {source_id, source_confidence, excerpt_hash}
decision_trace	list[object]	ordered: {step, decision_origin, rationale} where decision_origin matches C2 enum

Storage rule: Records are append-only. decision_trace is required even when length is 1.

Change-control: Adding fields is MINOR. Changing list item shapes, or removing required fields, is MAJOR. decision_origin enum is governed by C2 — changes propagate.

C4 — Provenance → Evaluation

Producer: s3-source-inventory-template (S3, register).

Consumer: s2-scoring-system (S2, spec).

Payload schema (per source):

Field	Type	Allowed values			
source_id	string	kebab-case			
confidence_band	enum	high \	medium \	low \	unknown
band_rationale	string	one-sentence justification			

Usage rule: Eval rubrics cap the score of any rubric criterion grounded in retrieval by the source_confidence (alias of confidence_band) of the lowest-confidence source the answer relies on. Caps are explicit:

- high — no cap. Criterion may score up to the rubric maximum (2 of 2 on a binary-plus-partial scale).

- medium — capped at 1 of 2 (half credit; explicit partial pass).
- low — capped at 0 (cannot pass).
- unknown — capped at 0 (cannot pass) AND the run is marked for review.

These caps apply per criterion, before any aggregation. A run citing one unknown source and one high source caps only the criteria grounded in the unknown source; criteria grounded solely in the high source are uncapped.

Change-control: confidence_band enum changes are MAJOR. Cap-table changes (the four high/medium/low/unknown cap values in the usage rule) are MAJOR. Band re-assignments are MINOR but logged to provenance as their own events.

Cluster Extension Contracts (C5–C12)

C1–C4 wire the Systems cluster and are frozen. C5–C12 (added in v0.3.0) wire the Execution, Trust, and Coordination clusters back into Systems. Every extension contract names a course-prefixed producer (e4/e5/e6/t7/t8/c9/c10), a Systems consumer, the payload crossing the boundary, and change-control. All extension contracts adopt v0.2.0 conventions: course-prefixed ids, explicit cap behavior where scoring is involved, and the {ref, hash, field} hash-and-ref pattern for any heavy evidence payload (so the C3 split-storage model is preserved — full content in the S2 audit record, pointers in the S3 provenance record).

Decision-origin dependency. C5, C6, C9, C10, and C11 emit one of the five extended decision_origin values locked in glossary.md v0.2.0 (workflow-step, sensor-triggered, compliance-determination, permission-decision, governance-vote). Because those values are an additive superset of the base C2 enum, the Systems consumers (s3-provenance-metadata-schema, s2-audit-trail-schema) accept them without a breaking change; a base-only reader maps any extended value to agent for control purposes. Adding a sixth extended value is MINOR; renaming or removing one, or changing a base value, remains MAJOR.

C5 — Workflow → Provenance

Producer: e4-workflow-artifacts (S4, template) — the process-map executor emits one record per executed workflow step.

Consumer: s3-provenance-metadata-schema (S3, schema).

Payload schema (one record per step):

Field	Type	Allowed values
step_id	string	kebab-case, unique within the process map
step_inputs_ref	object	{ref, hash, field} pointer into the S2 audit record (heavy payloads are NOT inlined)

Field	Type	Allowed values
step_outputs_ref	object	{ref, hash, field} pointer into the S2 audit record
branch_taken	string	the branch/edge id selected at this step, or none for linear steps
decision_origin	enum	workflow-step for autonomous steps; human-override/escalation/fallback when a gate fired (reuses C2 enum)

Emission rule: Every step that produces an output or selects a branch emits exactly one C5 record. Steps gated by an HITL checkpoint also emit a C2 event from S1 (the two records share a correlation_id). Drops are gate-failing.

Change-control: Adding step metadata is MINOR. Removing a field or changing decision_origin semantics is MAJOR. Inlining heavy payloads instead of hash-and-ref is forbidden (violates C3 split-storage).

C6 — Sensor Fusion → Reliability

Producer: e5-alert-design-spec (S5, spec) — the fused-telemetry alert stream.

Consumer: s1-fallback-architecture-blueprint and s1-threshold-escalation-spec (S1).

Payload schema (one record per alert):

Field	Type	Allowed values				
sensor_id	string	kebab-case; resolves to a row in e5-telemetry-source-map				
signal	string	the normalized, fused measurement that crossed the threshold				
severity	enum	critical \	high \	medium \	low \	info (reuses S1/glossary enum)
confidence_band	enum	high \	medium \	low \	unknown (reuses C4 enum); MAY carry an optional numeric confidence_score (0.0–1.0)	
suggested_fallback	string	the declared S1 fallback the alert recommends, or none				

Routing rule: A critical or high alert routes into S1's threshold-escalation spec exactly as an internal low-confidence signal would; the S1 state machine — not the sensor layer — owns the fallback decision. Telemetry alerts MUST obey S1 fallbacks; a sensor layer that silently overrides the S1 state machine is gate-failing.

Change-control: Adding the numeric confidence_score alongside the band is MINOR (the band remains authoritative for control). Changing severity or confidence_band enums is MAJOR (governed by S1/C4).

C7 — Compute Economics → Evaluation

Producer: e6-routing-policy (S7, policy) — the routing-decision logger.

Consumer: s2-audit-trail-schema (S2, schema), appended to each tool_calls[] entry.

Payload schema (appended to each tool/model call in the audit record):

Field	Type	Notes			
model_tier	enum	premium \	standard \	cheap \	deterministic
routing_rationale	string	one-sentence reason the tier was selected			
est_cost_usd	number	estimated marginal cost of the call			
est_latency_ms	number	estimated/observed latency			
fallback_tier_if_any	string	tier used if the primary tier failed or exceeded budget, else none			

Usage rule: Cost/routing metadata is observability appended to the existing tool_calls[] shape; it does NOT change C3 storage (still split-storage). The S2 scoring system MAY read model_tier to flag a cost/quality regression but MUST NOT let tier alone change a pass/fail verdict — safety and quality scoring are tier-agnostic.

Change-control: Adding cost/latency metadata is MINOR. Renaming a field or changing the model_tier enum is MAJOR.

C8 — Compute Economics → Reliability

Producer: e6-routing-policy (S7, policy) — the budget gate.

Consumer: s1-hitl-review-policy (S1, policy).

Payload schema: a budget-exceedance event that reuses the C2 six-field event shape verbatim (event_id UUID v7, timestamp, agent_id, decision_origin, evidence_pointer, rationale), with decision_origin: escalation and rationale naming the exceeded budget (cost, token, rate-limit, or latency).

Trigger rule: Crossing a hard budget ceiling defined in e6-cost-model raises an HITL gate via S1 rather than silently degrading or stopping; the human decides continue/downgrade/abort. Soft-budget warnings do NOT gate — they log to C7 only.

Change-control: Inherits C2 change-control. Budget-class additions (storage, egress, ...) are MINOR.

C9 — GovTech Compliance → Provenance

Producer: t7-compliance-workflow (S6, spec) — the compliance-determination logger.

Consumer: s3-provenance-metadata-schema (S3, schema).

Payload schema (one record per determination):

Field	Type	Allowed values			
rule_id	string	resolves to a row in t7-jurisdiction-rule-map			
jurisdiction_id	string	kebab-case (e.g., fl-walton-county, fl-state, us-federal)			
determination	enum	compliant \	non-compliant \	needs-review \	not-applicable
evidence_pointer	string	URI/key resolving to the cited rule text and the input it was applied to			
decision_origin	enum	compliance-determination (autonomous) or escalation (routed to expert)			
expert_review_required	boolean	true whenever the determination crosses the expert-review boundary			

Routing rule: Any determination with expert_review_required: true MUST also raise a C2/S1 HITL event and MUST NOT be surfaced to an applicant as final. The "not legal advice" boundary is enforced here, not in prose.

Change-control: Adding a determination value is MINOR; removing one or changing decision_origin is MAJOR.

C10 — Security Zero Trust → Reliability + Provenance

Producer: t8-tool-use-control-list (S8, matrix) — the permission-decision point for every tool invocation.

Consumers: s1-hitl-review-policy (S1, policy) and s3-provenance-metadata-schema (S3, schema).

Payload schema (one record per tool-invocation attempt):

Field	Type	Allowed values					
tool_id	string	resolves to a row in t8-tool-use-control-list					
principal	string	the agent/role attempting the action					
action_class	enum	read \	external-write \	financial \	comms \	admin \	destructive
permission_decision	enum	auto \	confirm \	human-approval \	blocked		
decision_origin	enum	permission-decision (autonomous policy eval) or human-override (a human changed the decision)					
evidence_pointer	string	pointer to the policy rule and the requested arguments					

Gate rule: human-approval and any financial/destructive action class raise a C2/S1 HITL gate; blocked is terminal and emits a provenance record but no action. A tool invocation with no matching control-list row defaults to blocked (least privilege).

Change-control: Adding an action_class or permission_decision value is MINOR; removing one is MAJOR. The default-deny rule is MAJOR-locked.

C11 — DAO Governance → Provenance

Producer: c9-execution-accountability (S9, playbook) — the governance-event logger.

Consumer: s3-provenance-metadata-schema (S3, schema).

Payload schema (one record per executed governance decision):

Field	Type	Allowed values			
proposal_id	string	kebab-case; resolves to a c9-proposal-design-template instance			
decision_class	string	the class from c9-decision-rights-matrix (e.g., treasury-allocation, contract-change)			
outcome	enum	passed \	rejected \	vetoed \	quorum-failed
decision_rights_ref	string	pointer to the decision-rights row that authorized the vote			
evidence_pointer	string	pointer to the tallied votes and the proposal evidence pack			
decision_origin	enum	governance-vote			

Meta-governance rule: A decision_class: contract-change outcome of passed is the ONLY mechanism authorized to amend a C1–C12 contract or a substrate file at a MAJOR version. This closes the loop: the stack's own change-control is itself governed. Each such change emits a C11 record.

Change-control: Adding an outcome value is MINOR; changing decision_origin is MAJOR.

C12 — Commercialization → Evaluation + Compute Economics

Producer: c10-revenue-model-design (S10, spec) — pricing and offer architecture read proof-of-value and cost.

Consumers (read-only): s2-scoring-system (S2) and e6-unit-economics-worksheet (S6/Course 6).

Payload schema (per offer, read-only — Commercialization consumes, it does not write run-time events):

Field	Type	Notes
offer_id	string	kebab-case
pov_metric_refs	list[string]	rubric-criterion ids in s2-scoring-system that define the offer's proof-of-value
unit_cost_ref	string	pointer to the e6-unit-economics-worksheet row for this offer's delivery cost
min_margin_pct	number	the gross-margin floor below which the offer is not viable

Usage rule: A pricing hypothesis is invalid unless every `pov_metric_ref` resolves to a real, currently-passing rubric criterion and `unit_cost_ref` resolves to a populated unit-economics row. Commercialization cannot claim value the evaluation system has not scored, nor a margin the cost model does not support. This is the anti-vaporware contract.

Change-control: Adding fields is MINOR. Changing the resolution rule (that refs must point to live, passing criteria) is MAJOR.

Change log

- v0.3.1 (2026-05-29): Aligned the C3 `retrieval_sources` row to the shipped s2-audit-trail-schema producer, which emits `{source_id, source_confidence, excerpt_hash}` (replacing the prior mismatched per-item shape). C1/C2/C4–C12 unchanged.
- v0.3.0 (2026-05-29): Added cluster extension contracts C5–C12 wiring Execution (C5 Workflow→Provenance, C6 SensorFusion→Reliability, C7 ComputeEcon→Evaluation, C8 ComputeEcon→Reliability), Trust (C9 GovTech→Provenance, C10 Security→Reliability+Provenance), and Coordination (C11 Governance→Provenance, C12 Commercialization→Evaluation+ComputeEcon) back into Systems. C1–C4 unchanged and frozen. Relies on the additive `decision_origin` superset locked in glossary v0.2.0. C11 establishes meta-governance: a passed contract-change proposal is the only authorized path to a MAJOR contract/substrate amendment.
- v0.2.0 (2026-05-28): Reconciled producer/consumer ids to course-prefixed convention. Fixed C2 `event_id` type. Made C4 usage rule explicit (cap behavior by `confidence_band`). Aligned C1 consumer to s2-eval-test-seed-matrix.
- v0.1.0 (2026-05-28): Initial substrate contracts.