GoalOS: The Validated Skill Graph

Reusable Capability, Chronicle-Gated Memory, and Proof-Carrying Skill Accumulation

Montreal.AI Research

Abstract

This paper introduces the Validated Skill Graph: a proof-carrying memory substrate for autonomous AI work. Conventional AI memory stores text, embeddings, prompts, or traces. Such memory is easy to accumulate but difficult to govern; it may preserve persuasive output without evidence, replay, risk context, or admission policy. The Validated Skill Graph treats reusable capability as a typed, scoped, versioned, evidence-backed object. A candidate skill may originate from an LLM draft, an expert intervention, an AGI Job, a reviewer critique, or a repeated mission pattern, but it cannot influence future missions until it passes a Chronicle admission gate. Admission requires an Evidence Docket, returned ProofBundle artifacts, proof-level-dependent acceptance checks, risk posture, replay or reconstruction path, and a rollback condition. Accepted nodes become reusable capability; rejected or partial nodes become repair paths or custom AGI Jobs. We formalize the skill node object model, graph operations, proof-level policy ladder, worth-keeping utility, threat model, and evaluation program. The central claim is architectural: compounding value in a proof economy does not come from larger prompt libraries; it comes from validated skill graphs that convert proof-bearing work into governed, reusable, and auditable capability.

Thesis. GoalOS creates proof; the Validated Skill Graph makes proof reusable. A model may draft, an agent may act, and a reviewer may validate, but only a Chronicle-admitted skill node may influence future missions.

1 Introduction

The first wave of AI systems made generation abundant. They produced text, plans, code, reports, search results, and agent traces at low marginal cost. Abundance, however, is not the same as institutional trust. An organization does not merely need to remember what an agent said; it needs to know which outputs deserve reuse, payment, propagation, training, settlement, and rollback authority.

The Validated Skill Graph addresses this missing layer. It is not a prompt repository, vector database, note archive, or transcript store. It is a governed capability graph. Nodes are validated skills; edges describe how those skills can be reused, composed, taught, packaged, audited, retired, or applied to future missions. The graph is populated only through Chronicle admission: a policy gate that checks evidence, replay, validation, risk, and scope. This transforms memory from an informal convenience into an institutional control surface.

The design follows the GoalOS proof loop: an objective becomes a mission contract; unsupported claims become proof debt; proof debt becomes custom AGI Jobs; jobs return ProofBundle artifacts; artifacts enter an Evidence Docket; the Chronicle gate decides what becomes durable memory; admitted nodes become reusable capability. The Validated Skill Graph is therefore the compounding layer of the proof economy.

2 Problem Statement: Memory Without Proof

Ungated AI memory creates four recurring failures.

1. **Contamination.** A persuasive but unsupported answer can shape future behavior if it is stored as memory.
2. **Drift.** Instructions, heuristics, or tactics copied across runs lose context and gradually exceed their original scope.
3. **Irreplayability.** A remembered result may not include inputs, methods, versions, reviewers, or artifacts sufficient to reconstruct it.
4. **Authority leakage.** A local draft may be mistaken for validated evidence, settlement readiness, or production permission.

These failures are not solved by larger context windows or better summaries. They require a memory object with admission policy, provenance, versioning, evidence linkage, and rollback. The Validated Skill Graph supplies this object.

3 System Positioning

The Validated Skill Graph is one layer in a larger proof economy. It receives candidates from human experts, LLMs, AGI Agents, reviewers, and mission traces, but it is not controlled by any one of them. Its authority comes from the Chronicle gate.

- A **prompt repository** stores instruction fragments. A **validated skill graph** stores evidence-backed capability.
- A prompt can be copied. A validated skill carries provenance, acceptance tests, scope, and rollback.
- A prompt can drift. A skill node has versioned constraints and an explicit admissible-use envelope.
- A prompt may persuade. A skill node must survive proof.

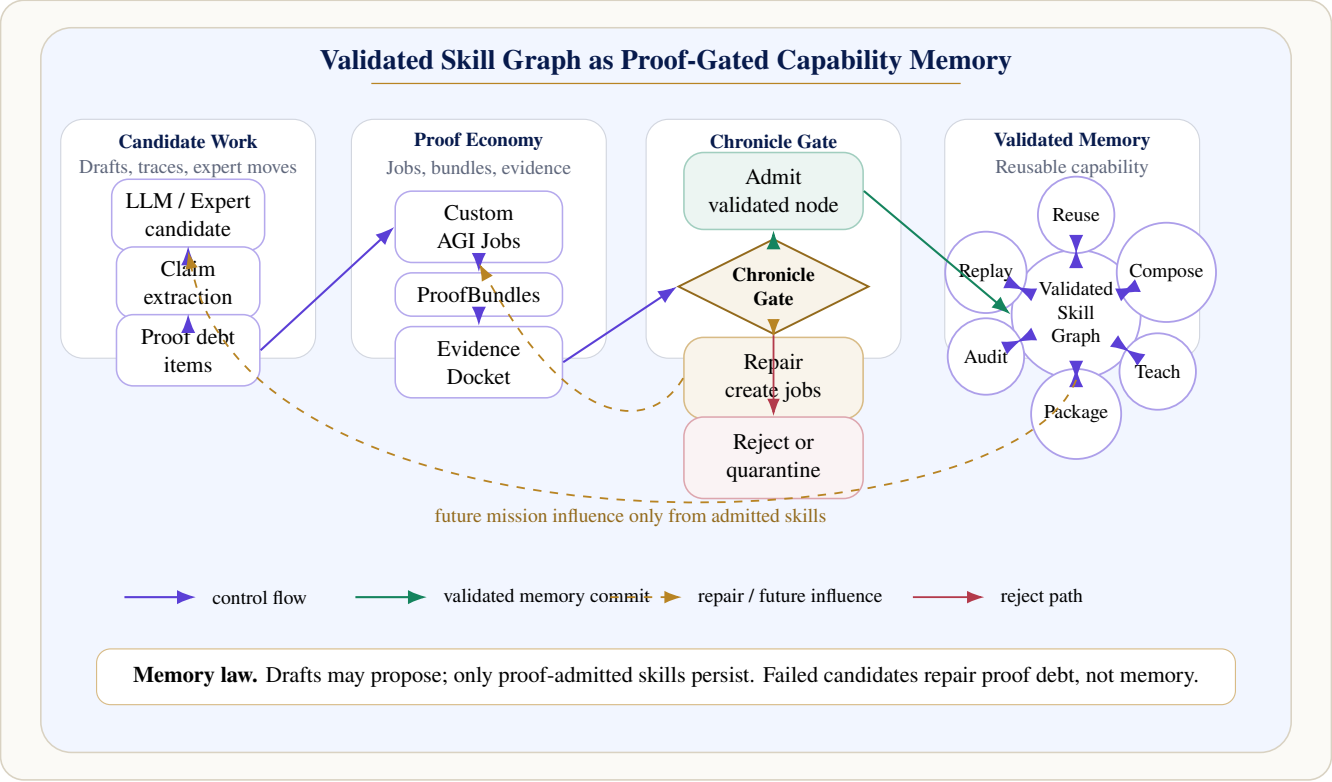

Figure 1. The Validated Skill Graph as a governed memory substrate. Candidate work becomes durable capability only after proof, evidence aggregation, and Chronicle admission.

Figure 1 positions the Validated Skill Graph as a memory substrate rather than a convenience layer. Drafts, traces, and expert moves remain outside durable memory until they become evidence-backed skill nodes.

4 Formal Object Model

A skill node is a scoped capability record. It contains not only a description of what worked, but the conditions under which the capability may be used again. Formally, a skill node is:

$$s = \langle id, type, scope, method, evidence, tests, validators, risk, policy, version, rollback, utility \rangle.$$

The fields have the following interpretation.

`id`	A stable identifier bound to mission lineage, version, and content hash.
`type`	The role of the skill: vision-to-proof, domain research, reviewer judgment, job design, replay, boundary audit, or composite skill.
`scope`	The admissible-use envelope: domain, assumptions, proof level, jurisdiction, tools, and excluded uses.
`method`	The procedure, heuristics, toolchain, and sequencing that produced the result.
`evidence`	Links to ProofBundle artifacts, Evidence Docket entries, replay manifests, reviewer notes, and metrics.
`tests`	Acceptance tests required before reuse or composition.
`validators`	Human, AGI Node, council, or hybrid validators, including review state and quorum.
`risk`	Risk posture, failure modes, open debt, and repair conditions.
`policy`	Admission level, reuse permissions, retention period, and mainnet handoff boundary.
`version`	Version number, parent nodes, supersession path, and changelog.
`rollback`	Conditions for demotion, quarantine, or retirement.
`utility`	Estimated and measured contribution to future mission success, cost reduction, speed, trust, or safety.

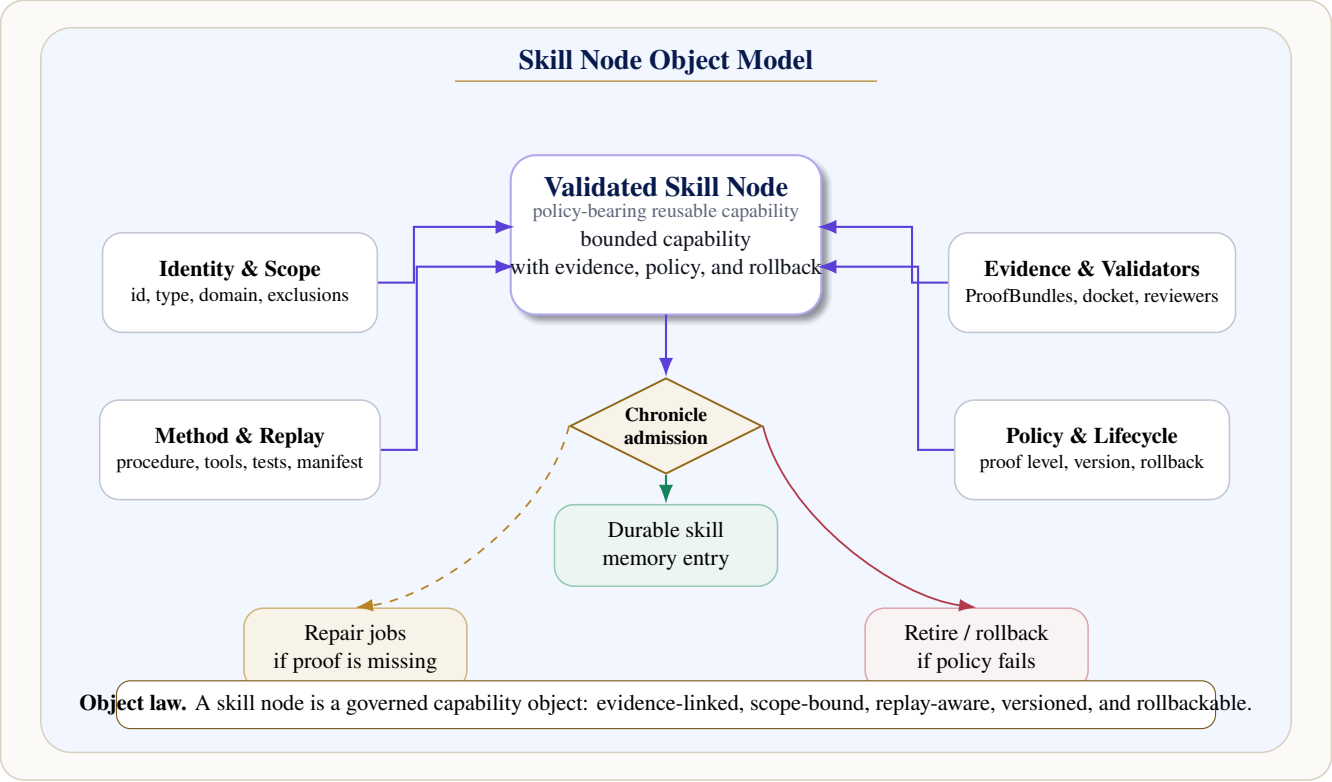

Figure 2. A validated skill node contains operational semantics, evidence, policy, utility, and rollback. The node is a capability record, not a prompt fragment.

5 Chronicle Admission Policy

Chronicle is the gate between evidence and memory. It is not a passive database write. A candidate skill enters the graph only if the configured proof level is satisfied. Let c be a candidate skill and ℓ a proof level. The admission function is:

$$A_\ell(c) = \mathbf{1}\big[E(c) \geq \tau_\ell^E \ \wedge \ R(c) \leq \tau_\ell^R \ \wedge \ V(c) \geq \tau_\ell^V \ \wedge \ P(c) = 1 \ \wedge \ B(c) = 1\big],$$

where E measures evidence strength, R risk posture, V validation coverage, P replay or reconstruction readiness, and B boundary compliance. If $A_\ell(c) = 1$, the skill is admitted. If not, the system routes it to repair, rejection, quarantine, or further AGI Jobs.

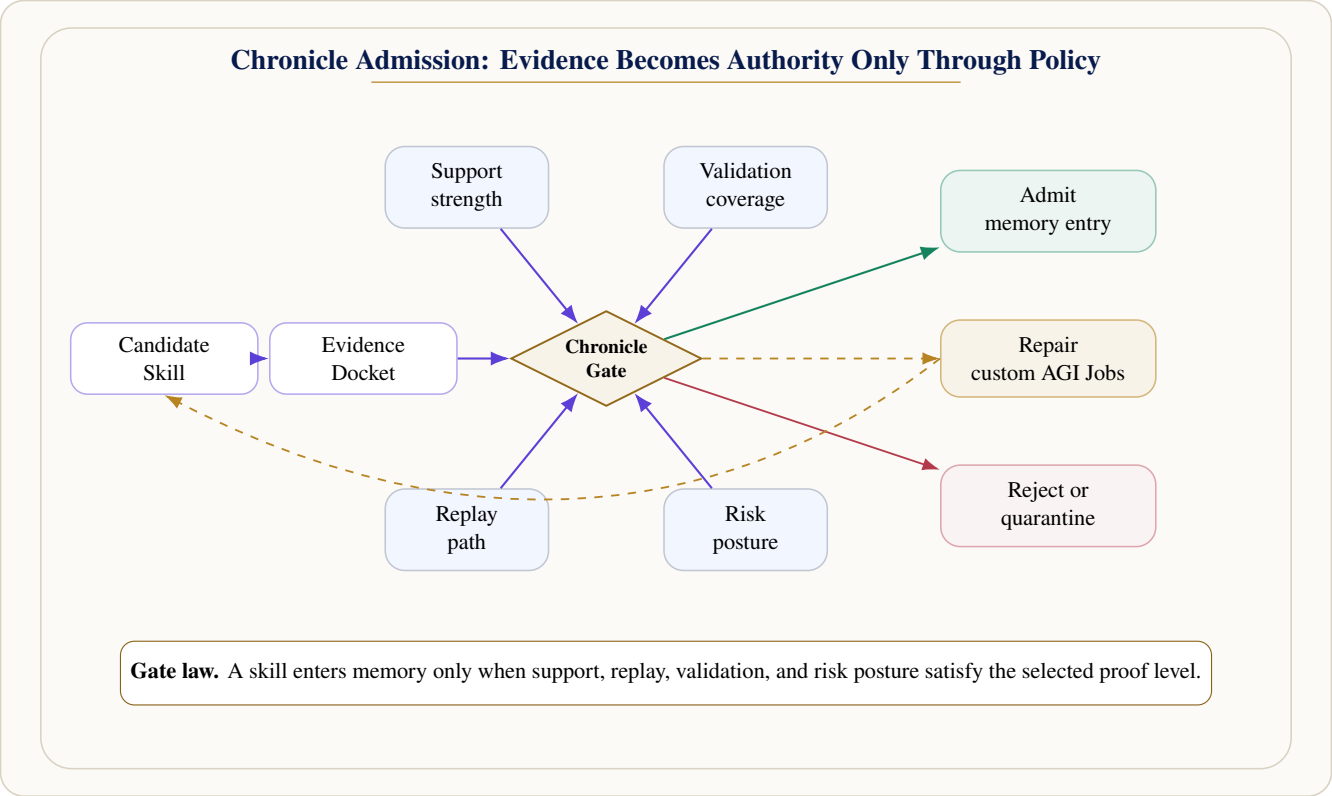

Figure 3. Chronicle admission policy. The gate transforms evidence into authority only when configured proof requirements are satisfied.

6 Proof-Level Dependent Admission

Not every mission requires the same level of proof. A brainstorming mission, internal review, public claim, mainnet handoff, and production-grade memory entry should not share the same threshold. GoalOS therefore treats proof level as a mission parameter.

Table 1. Proof levels for validated skill admission.

Level	Name	Required evidence posture	Memory effect
L1	Structural / Demo	Mission structure, explicit proof debt, local artifact coherence.	No durable Chronicle influence.
L2	Internal Review	Operator or internal reviewer confirms usefulness, scope, and obvious safety.	Temporary reuse, clearly labeled.
L3	External Review	Cross-source support or independent reviewer / AGI Node validation.	Scoped reusable skill.
L4	Mainnet Handoff	Handoff packet with validated evidence, job specification, and settlement boundary.	May inform external execution preparation.
L5	Chronicle / Formal	Multi-validator evidence, replay, baseline comparison, risk gate, rollback condition.	Durable skill with future mission influence.

Raising the level increases validator burden, evidence strictness, and reuse authority. Lowering the level accelerates ideation but prevents strong claims from compounding. The selected proof level is immutable within a mission so that agents, validators, and operators know which admission policy governs the run.

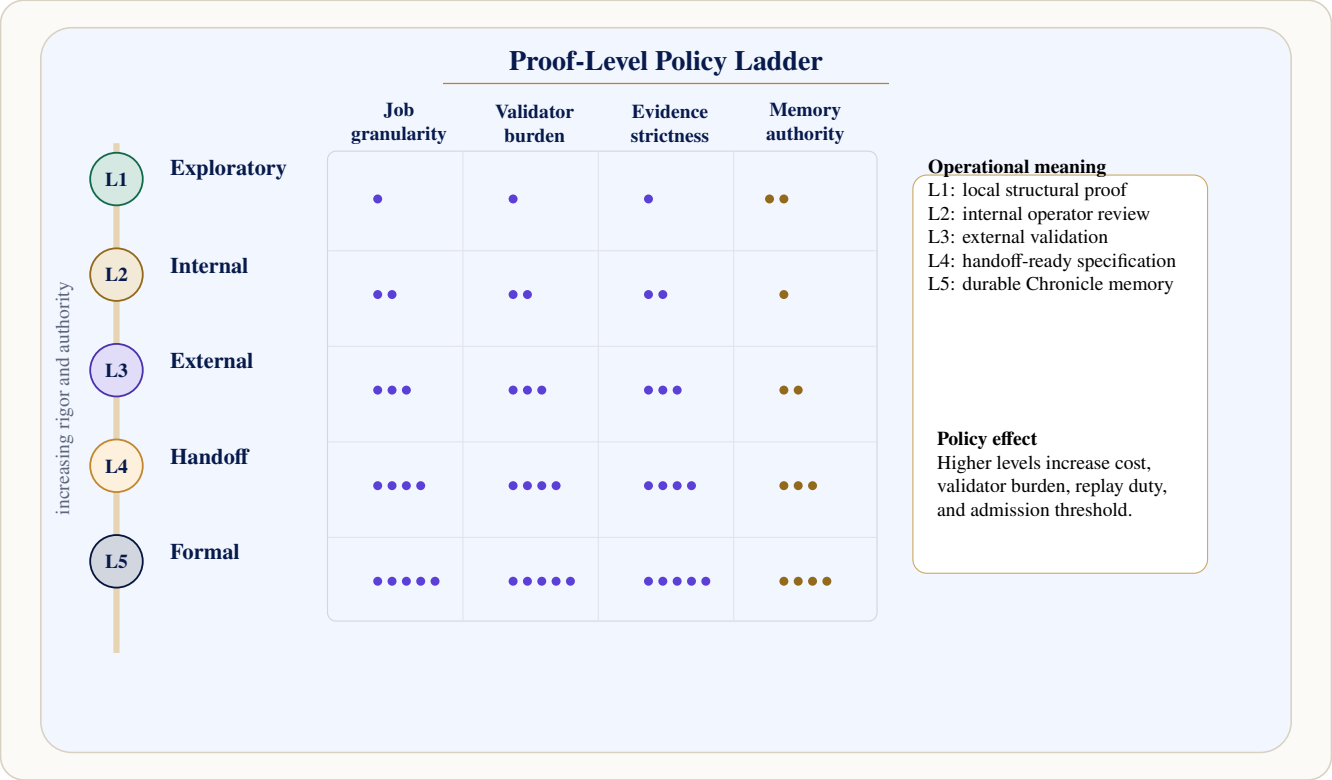

Figure 4. Flexible proof levels calibrate the authority of skill admission. The selected level changes job decomposition, validator density, evidence strictness, and memory effect.

7 Graph Semantics and Operations

The Validated Skill Graph supports six core operations. Each operation is evidence-preserving and policy-aware.

1. **Reuse.** Apply a validated skill to a new mission within its scope.
2. **Compose.** Combine multiple validated skills into a higher-order capability.
3. **Teach.** Derive training material, checklists, or masterclass lessons from a skill node.
4. **Package.** Emit a reusable artifact: proof room, playbook, job template, or handoff packet.
5. **Audit.** Re-open a skill for review when evidence, policy, or context changes.
6. **Retire.** Demote or quarantine a skill that fails replay, drifts out of scope, or becomes unsafe.

Let $G = (S, E)$ be the skill graph. Each edge $e = (s_i, s_j, \tau, w, p)$ connects two skill nodes with relation type τ, utility weight w, and policy condition p. A future mission may use an edge only if both endpoint skills are admitted under the mission's required proof level and the edge policy matches the mission scope.

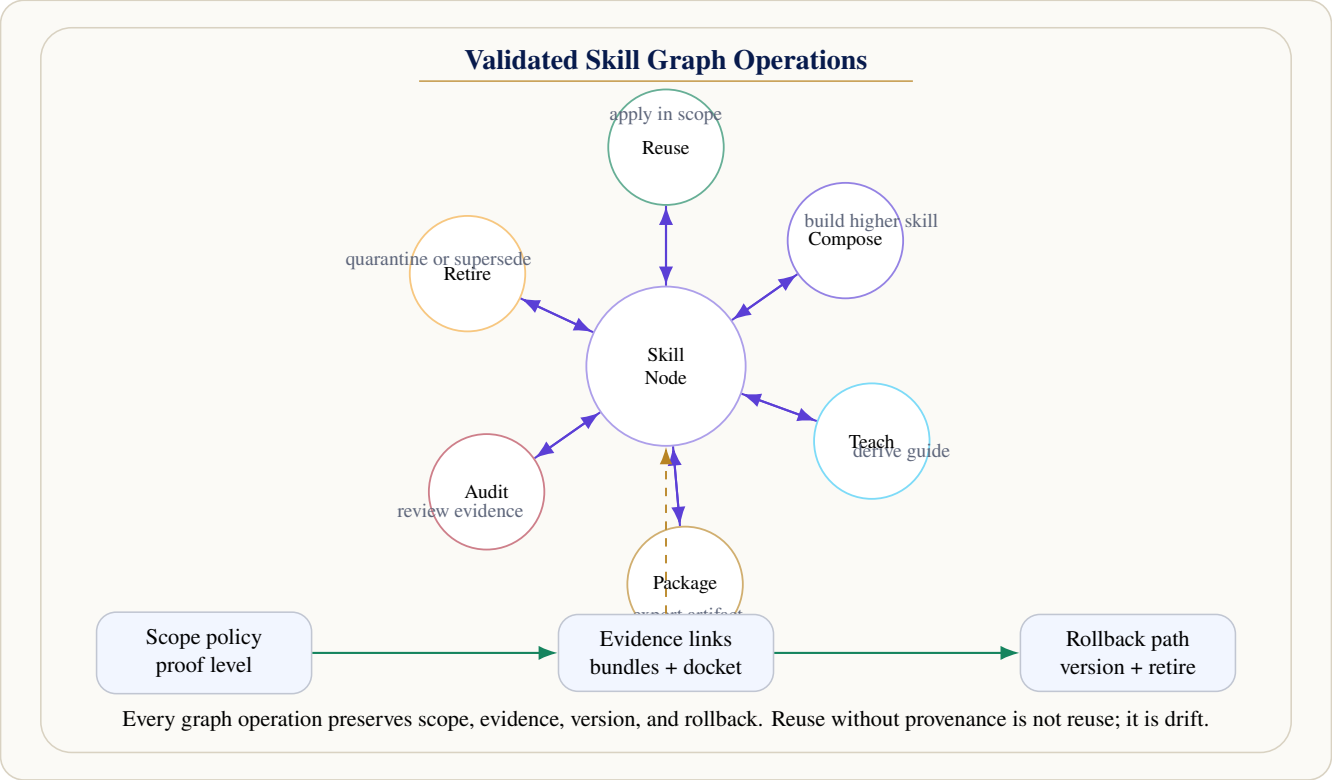

Figure 5. Core operations of the Validated Skill Graph. Operations are allowed only when the relevant evidence, scope, and policy conditions hold.

8 Worth-Keeping Utility

Not every validated artifact should become durable memory. The graph needs a retention policy that prevents memory bloat. We define a worth-keeping function:

$$W(s) = \alpha \Delta Q + \beta \Delta T^{-1} + \gamma \Delta C^{-1} + \delta \Delta R^{-1} + \eta U - \lambda_1 K - \lambda_2 D - \lambda_3 B,$$

where ΔQ is quality improvement, ΔT^{-1} time reduction, ΔC^{-1} cost reduction, ΔR^{-1} risk reduction, U expected reuse, K maintenance cost, D drift risk, and B boundary complexity. A skill is retained when $A_\ell(s) = 1$ and $W(s)$ exceeds the retention threshold.

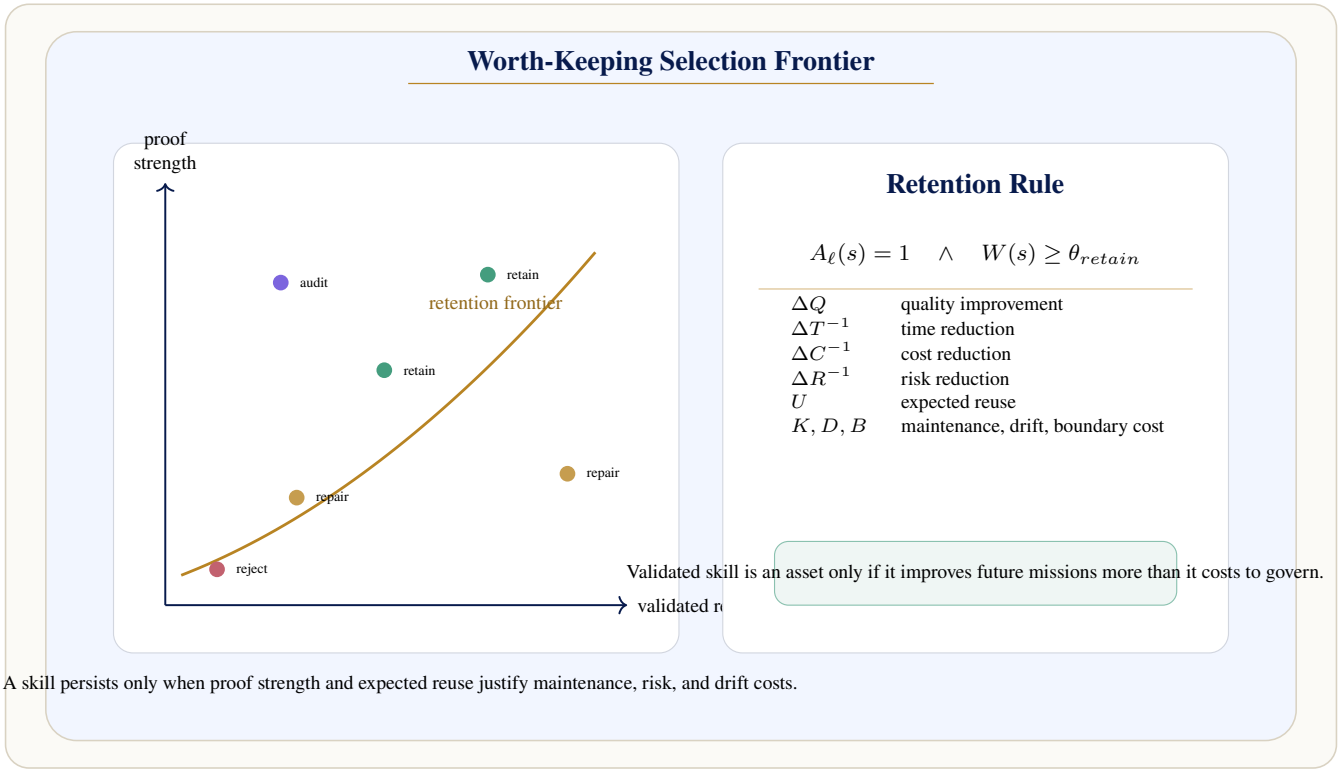

Figure 6. Worth-keeping utility prevents memory bloat. Chronicle admission is necessary but not sufficient: persistent skills must also justify reuse, maintenance, and governance cost.

9 Skill Capture Studio

The Skill Capture Studio converts expert judgment, meeting notes, mission traces, and reviewer critiques into candidate skill nodes. It is deliberately not a general memory importer. It extracts skill episodes: bounded reusable moves with a trigger, method, evidence requirements, validator rubric, and safe reuse envelope.

A skill episode has the form:

$$\epsilon = \langle trigger, decision, method, artifact, validator, failure, reuse\_condition \rangle.$$

For example, an expert move such as "convert a visionary claim into a proof-gated mission architecture" becomes a candidate skill only after it is scoped, tested, reviewed, and admitted. The result may be reused in future product missions, but only within its accepted proof level.

10 Capability Composition

Skill nodes can be composed. Composition is powerful and dangerous: two individually valid skills can become unsafe when combined. GoalOS therefore treats composition as a new candidate skill requiring its own proof packet.

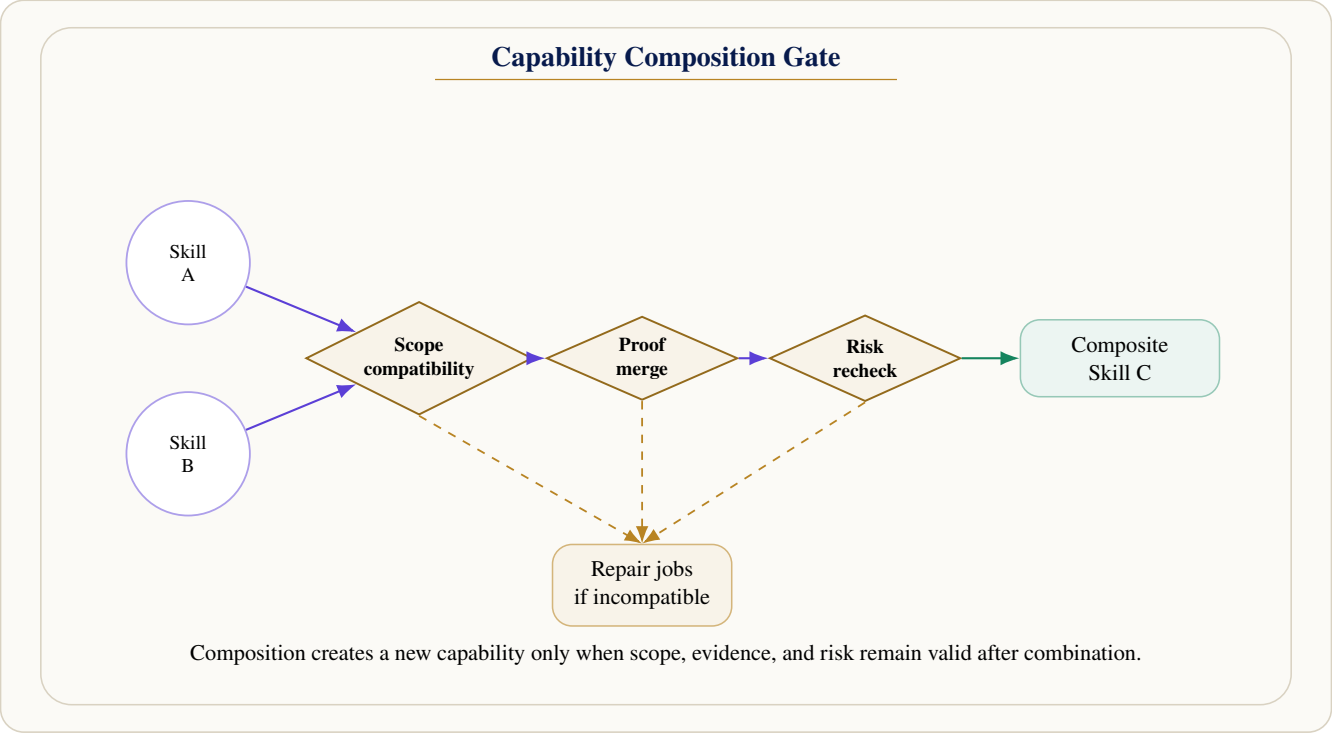

Figure 7. Capability composition requires a new gate. Valid skills do not automatically yield a valid composite.

11 Custom AGI Jobs for Skill Repair and Extension

A missing proof item becomes a custom AGI Job. The job factory starts empty. It receives the mission objective, selected proof level, skill candidate, blocked claim, and repair target. It then creates the minimal work needed to retire the proof debt. A generated job contains:

- source skill or candidate;
- proof gap and blocked claim;
- worker role and required tools;
- validator role and acceptance tests;
- expected ProofBundle structure;
- failure mode and repair branch;
- optional mainnet handoff metadata.

Because jobs are custom and contextual, the Validated Skill Graph avoids the misleading appearance that a static catalog of jobs exists before the user defines the objective and proof level.

12 Mainnet Handoff Boundary

The Validated Skill Graph can prepare handoff packets for external execution or settlement, but the local graph does not itself post to a chain, move funds, escrow payouts, sign transactions, or create on-chain records. The local system creates evidence-bound packets: mission scope, skill candidates, job specs, proof state, validator requirements, and content hashes. Live posting and settlement remain outside local mode.

This boundary preserves both safety and clarity. A skill may be internally validated, handoff-ready, or settlement-ready, but these are distinct states. External finality requires explicit user action, wallet signing, protocol execution, and external validation rules.

13 Evaluation Program

The Validated Skill Graph should be evaluated by its effect on future work, not by the number of nodes it stores. The key metrics are:

- **Chronicle precision:** fraction of admitted skills that later pass replay or reuse tests.
- **Reuse yield:** measurable improvement in future mission speed, quality, cost, or risk posture.
- **Drift resistance:** rate at which skill nodes remain within scope under repeated reuse.
- **Repair efficiency:** proportion of failed candidates that are converted into useful AGI Jobs.
- **Composition validity:** success rate of composites versus atomic skills.
- **Boundary correctness:** absence of local-live confusion, unsupported authority leakage, or unapproved settlement claims.

14 Threat Model

The graph is designed against threats that arise when memory becomes authority.

Table 2. Threats and controls for validated skill accumulation.

Threat	Failure mode	Control
Prompt contamination	Persuasive text enters memory as capability.	Chronicle gate; proof-level admission; no durable memory without evidence.
Reviewer capture	Weak skill admitted through bad review.	Multi-validator policies; challenge windows; replay; reputation.
Scope drift	Skill used outside admitted envelope.	Scope tags; policy checks; runtime warnings; rollback.
Memory bloat	Low-value nodes accumulate.	Worth-keeping utility and retirement policy.
Replay failure	Skill cannot be reconstructed.	Replay manifest, artifact hashes, baseline records.
Settlement confusion	Local skill treated as on-chain finality.	Handoff boundary and explicit external signing.

15 Commercial Implications

The Validated Skill Graph is the compounding asset behind a proof mission business. Missions create jobs. Jobs create evidence. Evidence creates validated skills. Validated skills reduce future cost, improve quality, improve buyer trust, and enable specialized product lines such as AI Trust Rooms, Enterprise Proof OS, and validator marketplaces.

The commercial unit is therefore not access to an output generator. It is the ability to create, validate, reuse, and govern skill. The skill graph makes expertise portable without making unsupported claims contagious.

16 Falsification Tests

The Validated Skill Graph claim should fail if any of the following hold under fair evaluation:

1. Admitted skills do not improve future missions compared with ordinary notes or prompt libraries.
2. Chronicle admission cannot prevent unsupported claims from entering future mission influence.
3. Skill composition increases failure or drift more than it increases leverage.

4. The worth-keeping policy admits low-value nodes or rejects high-value validated nodes at high rates.

5. The mainnet handoff boundary cannot be made clear to users or reviewers.

These falsification tests make the system empirical: the graph must earn its authority by improving future work without increasing ungrounded authority.

17 Conclusion

The Validated Skill Graph is the memory layer of a proof economy. It changes the unit of accumulation from text to validated capability. A skill is not a prompt, note, or transcript. It is a bounded, evidence-backed, replay-aware, policy-bearing, rollbackable object that can influence future missions only after Chronicle admission.

This design converts proof-bearing work into durable capability. It also supplies the missing bridge between autonomous work and institutional trust: agents may explore, reviewers may validate, and jobs may generate evidence, but only the validated skill graph compounds.

> **Final statement.** The durable asset is not the prompt. It is the proof-carrying skill node. The durable advantage is not more output. It is the validated graph of reusable capability.

A Canonical Skill Node JSON


```
{
  "skill_id": "vsg-skill-vision-to-proof-001",
  "type": "vision_to_proof",
  "scope": {
    "domains": ["AI Trust Rooms", "proof missions"],
    "proof_level": "L3_external_review",
    "excluded_uses": ["legal certification", "investment advice", "production authority"]
  },
  "method": {
    "summary": "Convert a high-level claim into a bounded mission, proof debt, AGI Jobs, and reviewer
        room.",
    "inputs": ["objective", "claim", "constraints"],
    "outputs": ["mission_contract", "proof_debt", "agi_jobs", "evidence_docket"]
  },
  "evidence": {
    "proofbundles": ["bundle://example/vision-to-proof"],
    "evidence_docket": "docket://example/61",
    "reviewers": ["human_reviewer", "agi_node_validator"]
  },
  "tests": [
    "claim_boundary_explicit",
    "proof_debt_nonempty_when_claims_are_unsupported",
    "reviewer_room_generated",
    "blocked_claims_visible"
  ],
  "policy": {
    "chronicle_admission": "requires_external_review",
    "reuse_allowed": true,
    "mainnet_handoff_allowed": false
  },
  "version": {
    "number": "1.0.0",
    "parents": [],
    "superseded_by": null
```

```
  },
  "rollback": {
    "conditions": ["review_failure", "scope_violation", "replay_failure"],
    "action": "quarantine_and_create_repair_jobs"
  },
  "utility": {
    "expected_reuse": 0.72,
    "measured_time_reduction": null,
    "risk_reduction": "pending_measurement"
  }
}
```

B References

[1] Montreal.AI Research. *GoalOS Mission OS: The Proof OS for Autonomous AI Work*. Publication-safe claim-bounded edition, 2026.

[2] Montreal.AI Research. *GoalOS: The Proof-of-Evolution Constitution, AEP-001*. Institutional standard edition, 2026.

[3] Montreal.AI Research. *AGI ALPHA: A Scalable Substrate for Intelligence Organizations*. Systems and implementation-architecture paper, 2026.

[4] Montreal.AI Research. *AGIJobManager: Protocol Terms and Smart Contract Boundary*. Mainnet handoff and settlement risk disclosure, 2026.