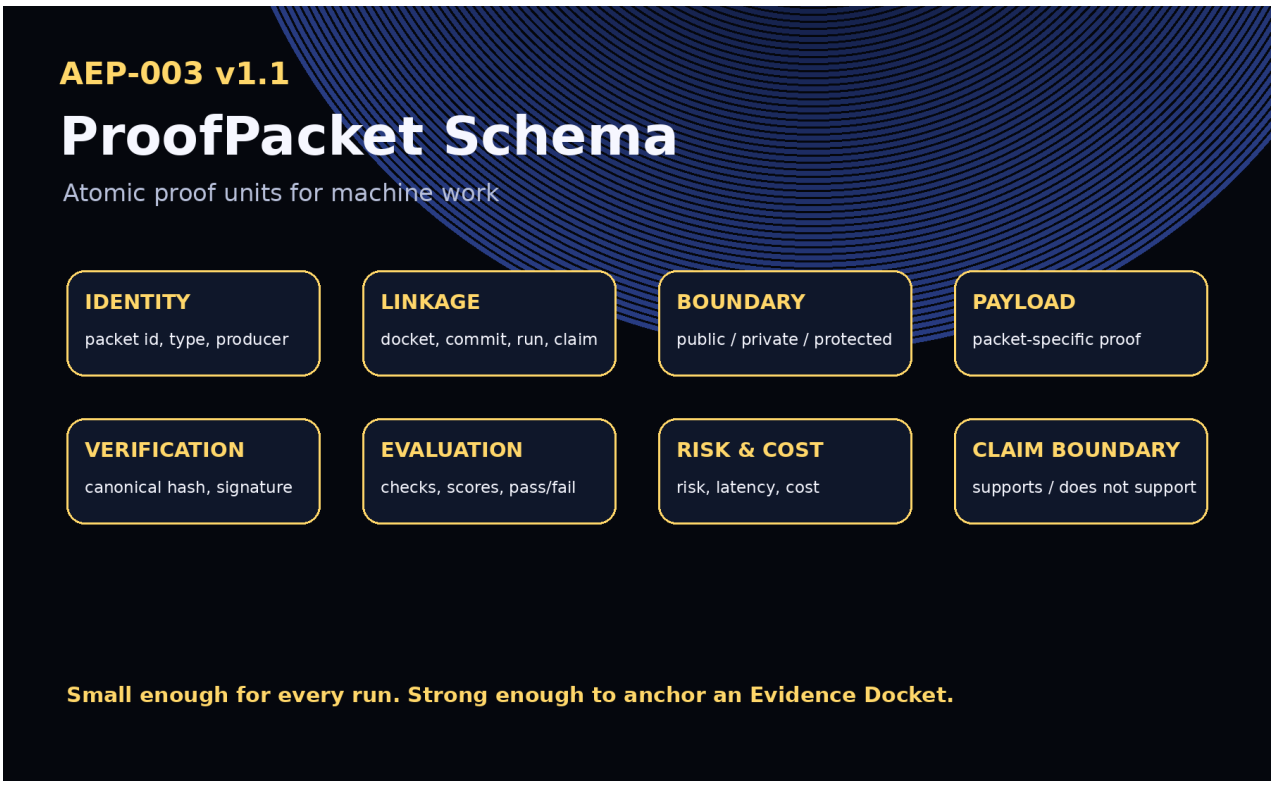
AEP-003

ProofPacket Schema

Atomic proof units for machine work

Version 1.1 Institutional Edition · Proof Gradient / Agent Evolution Protocol



AEP-001 defines the protocol. AEP-002 defines the docket. AEP-003 defines the packet.

1. Abstract

AEP-003 defines the ProofPacket: the atomic evidence unit emitted by an AI-agent run, machine-work step, tool call, evaluation, policy decision, selection decision, rollout event, rollback event, or public-safe report event.

AEP-001 defines the constitutional protocol. AEP-002 defines the Evidence Docket. AEP-003 defines the packet-level schema that makes machine work traceable, portable, tamper-evident, and composable.

2. Canonical Law

No proof, no evolution.

No eval, no propagation.

No rollback, no release.

3. v1.1 Implementation Upgrade

- Canonicalization profile
- Attestation and bundle schemas
- Full examples for all packet types
- Hash, validation, conformance, and chain-verification tools
- Website, conformance CI, and release workflows

4. Design Principles

- Atomic
- Hashable
- Composable
- Boundary-aware
- Claim-bounded
- Evolvable
- Rollback-aware

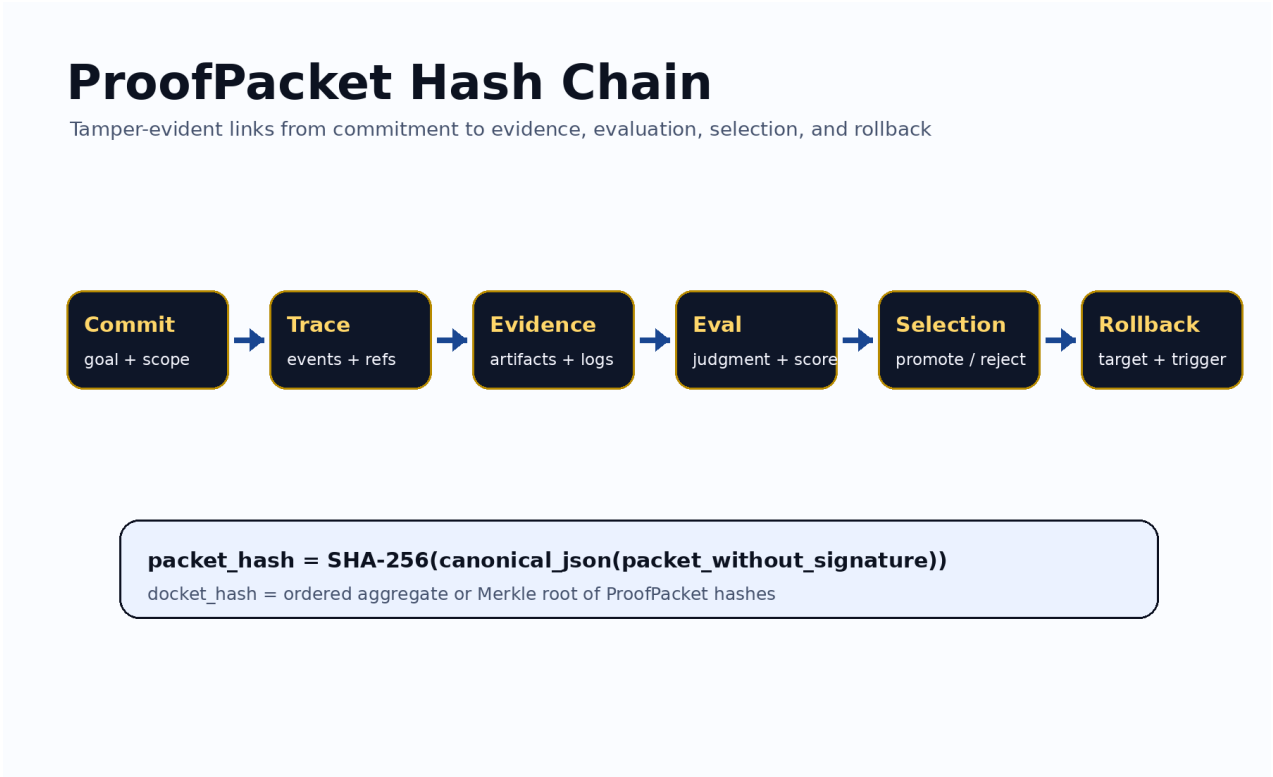
5. Required Fields

Field	Purpose
packet_id	Unique packet identifier.
schema	Must be AEP-003.
schema_version	Schema version.
packet_type	commit, trace_event, tool_call, eval_result, rollback_event, etc.
created_at	Creation timestamp.
producer	Agent, system, human, organization, workflow, or validator.
docket_id	Related AEP-002 Evidence Docket.
commitment_id	Related commitment.
run_id	Related machine-work run.
claim_refs	Claims supported or bounded by the packet.
evidence_refs	Evidence references.
boundary	Public/private/protected visibility boundary.
payload	Packet-type-specific content.
hash	Canonical packet hash.
claim_boundary	Supported and unsupported claims.

6. Packet Types

Packet Type	Purpose
commit	Mission, authorization, constraints, success criteria, failure criteria.
trace_event	Relevant execution event without full trace exposure.
tool_call	Tool use, permission class, approval state, result summary.
policy_decision	Policy decision, denial, escalation, or approval requirement.
approval	Human or institutional authorization.
eval_result	Evaluation, test, score, check, or evaluator attestation.
evidence_ref	Reference to evidence stored elsewhere.
risk_event	Risk, incident, mitigation, residual concern.
cost_event	Cost, latency, token count, compute time, human-review burden.
selection_decision	Selection Gate decision.
rollout_event	Canary scope, monitoring plan, propagation limit.
rollback_event	Rollback trigger, target, action, owner, recovery.
public_report	Public-safe report publication event.

7. Canonical JSON and Hashing



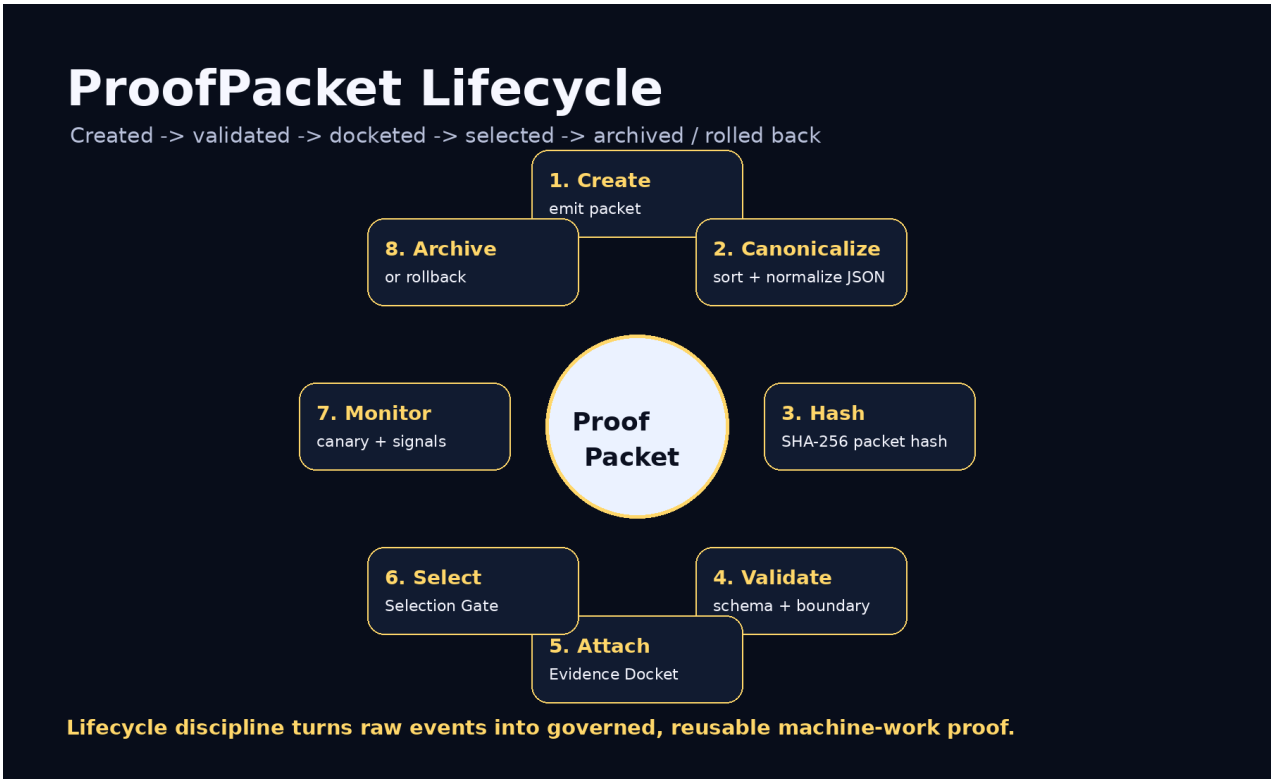
A ProofPacket should be hashed using deterministic canonical JSON: remove mutable signature fields, set hash to a zero-placeholder, sort object keys, encode as UTF-8, and hash using SHA-256 unless another approved algorithm is specified.

packet_hash = SHA256(canonical_json(packet_without_signature))

8. Boundary Block

- access_class: public, private, protected, or restricted
- public_safe: true or false
- contains_sensitive_data: true or false
- publication_allowed: true or false
- jurisdiction and retention policy

9. Lifecycle


ProofPacket Lifecycle
Created -> validated -> docketed -> selected -> archived / rolled back
1. Create
emit packet
2. Canonicalize
sort + normalize JSON
3. Hash
SHA-256 packet hash
4. Validate
schema + boundary
5. Attach
Evidence Docket
6. Select
Selection Gate
7. Monitor
canary + signals
8. Archive
or rollback
Proof Packet
Lifecycle discipline turns raw events into governed, reusable machine-work proof.

10. Conformance Levels

Level	Requirement
Level 0	Informal human-readable proof fragment.
Level 1	Valid JSON packet.
Level 2	Hashable packet with valid canonical hash.
Level 3	Docket-linked packet.
Level 4	Institutional packet with policy, eval, risk, cost, boundary, attestation, and Selection Gate references.
Level 5	Sovereign / regulated packet with jurisdiction, retention, protected-boundary classification, signer, timestamp authority, and restricted appendix controls.

11. Security and Privacy Requirements

Do not place secrets, personal data, regulated data, private prompts, full sensitive traces, credentials, confidential tool outputs, or protected operational details in a public ProofPacket. Use references, hashes, and access-class markers instead.

12. Claim Boundary

AEP-003 does not claim achieved AGI, achieved ASI, perfect safety, legal compliance certification, financial or legal advice, guaranteed ROI, production readiness, government endorsement, or national-security readiness. AEP-003 defines an atomic proof schema.

GoalOS makes machine work provable one packet at a time.