The Constellate Architecture

Will Blair
constellate.science/whitepaper
v0.1 · May 25, 2026

Abstract

Constellate replaces the read-only artifact substrate of scientific publication with a writable, governed state substrate. Findings, failures, corrections, dependencies, and reviewer judgments become signed, replayable state transitions over a per-frontier event log; two independent reducer implementations materialize that log into byte-identical state. The history is the canonical object, not any paper or dataset that triggered it.

The substrate is necessary because scientific generation has outpaced institutional absorption. Agentic systems already produce 10^3 to 10^4 acceptable-quality proposals per month per active frontier-lab pipeline; the late-decade upper bound is 10^5 to 10^6. No human review process integrates work at that volume into a record the next decision can read. The cost is borne by patients whose treatments arrive a decade late because a failed trial's lesson never travels, by early-career researchers reconstructing maps that should have been inherited, and by the next generation of scientific intelligence whose memory is private only to the labs that pay for it.

The architecture rests on three commitments: replay determinism, signed state transitions as the canonical object, and structural forkability such that capture at any orchestration layer fails to entrap the underlying state. The protocol — Vela — ships at v0.500+ under dual Apache-2.0 / MIT licensing. The first deployment is the blood-brain-barrier amyloid corridor in Alzheimer's translation, with named host, lab, and regulator candidates specified in the companion *First Corridor Pilot Plan v0.1*. The *Constellate trilogy — Constellations of Borrowed Light*, *The Discovery Engine*, and *The Terafactory Age* — makes the moral and institutional argument. This document specifies the protocol. §13 names how to participate.

1 The Problem

The amyloid hypothesis of Alzheimer's disease entered Phase III with bapineuzumab in 2008, solanezumab in 2009, semagacestat in 2010, verubecestat in 2013, and lanabecestat, atabecestat, and several others before lecanemab and donanemab partially redeemed the program in the early 2020s. Each Phase III failure unwound years of enrollment and represented patients and families who had organized their lives around a hypothesis that ended at interim analysis. The cost across the program ran past ten billion dollars in trial spend and a generation of clinical labor. The lessons the failures should have taught the field — about APOE4 stratification, about cerebrovascular comorbidities, about biomarker-cognition decoupling — were learned,

slowly, by the small set of investigators who happened to read the right post-hoc analyses. They did not enter a record the next trial designer could inherit. The 2025 AD pipeline (Cummings et al., 2025) shows the absorption gap as current data: 138 drugs in 182 active trials across 4,529 sites worldwide, 50,109 participants enrolled or being enrolled, 80% of sites participating in only one trial. Cross-trial learning at that scale is not happening anywhere a downstream trial designer can read.

Begley and Ellis published *Nature* in 2012 reporting that 47 of 53 "landmark" preclinical cancer findings could not be reproduced by Amgen's internal labs. The Reproducibility Project: Cancer Biology in 2021 reported that 46% of effects in 50 published cancer papers were less than half the original effect size. The 2015 Open Science Collaboration psychology replication found 36% of 100 effects replicated. None of these failures of inheritance entered a record that downstream trials, funders, or models could read. They became citations in the next reproducibility paper. The originals continued to accumulate uncorrected citation counts.

Scientific generation has become cheaper than institutional absorption. The cost of that gap is not borne by the institutions producing closed knowledge. It is borne by the patients waiting for treatments that arrive a decade late because a failed trial's lesson never traveled. It is borne by the taxpayers funding the duplicated experiments no one knew were duplicated. It is borne by the early-career researchers reconstructing maps that should have been inherited from their advisors. It is borne by the next generation of scientific intelligence, whose models train on whatever record their predecessors leave behind — and that record is, at present, the private memory of closed labs and the fluent-but-discardable context windows of agents reading whatever public corpus remains.

Models propose candidate experiments faster than wet labs can test them. Autonomous platforms run closed-loop synthesis at tempos that did not exist a decade earlier. Agentic systems extract findings, draft critiques, and chain tools at scales no graduate cohort can match. The bottleneck has moved. Once it was producing the next candidate; now it is integrating what has already been produced into a record the next decision can read.

The integrating layer does not exist. A paper records a claim and the author's reasoning, not what changed in the field's working knowledge or what depends on it. Datasets record observations without the claims they bear on. Repositories record code without the experiments it produced. Registries record artifacts without the state those artifacts changed. Each existing layer carries something true and useful. None carries the change.

AI sharpens the gap. Generation now produces public activity at a rate where private absorption is structurally insufficient. An agent synthesizing the literature produces a fluent summary no other agent inherits, no record carries forward, and no reviewer can read as a change. When models train on the corpus, they inherit whatever record exists — and if that record is private, incomplete, or controlled by the wrong incentives, AI reproduces the distortions at higher speed and lower friction.

The realistic deposit-rate envelope today is roughly 10^3 to 10^4 acceptable-quality agent-generated proposals per month per active frontier-lab pipeline, bottlenecked on tool-use reliability and grounding rather than generation speed. By the late decade the plausible upper bound rises to 10^5 to 10^6 per month per pipeline, conditional on improvements in agent reliability that are an open empirical question. Either rate exceeds the human-review capacity of any

single corridor by orders of magnitude, and the institutional layer that integrates them is the binding constraint Constellate is built to address.

The risk is not that science will fail to produce work, and it is not that closed labs will ship a competing protocol. Frontier labs already run their own internal state layers; those layers do not look like Vela, and they will not be made to. The risk is that the work compounds entirely *outside* any public substrate — that the next generation of scientific intelligence inherits the memory closed labs maintain for themselves, and the public record fills only with what the closed stacks chose to publish. The competitor is non-deposit, not closed-Vela.

External recognition of the absorption gap is visible at policy-portfolio scale. IFP's *Launch Sequence* (2025) proposes thirteen vertical AI-for-science infrastructure programs — Replication Engine, Advancing Biotech Commercialization, TELOS (evaluation), X-Labs (autonomous experimentation), Lost Archive (dark-data rescue), Scaling Pathogen Detection with Metagenomics, and others — with a collective ask in the range of roughly five to ten billion dollars in proposed federal and philanthropic spending across the portfolio. Each piece names an institutional shape for a single vertical of the problem this paper addresses; the collection's existence at that funding scale is independent evidence that the absorption gap is widely recognized by serious operators in 2025–2026. The substrate is the horizontal coordination layer those verticals would compose against. Section 11.4 returns to this with the technical positioning.

Existing scientific infrastructure shows what coordinated layers can do. The Protein Data Bank has held experimentally determined structures across competing institutions for over five decades and made AlphaFold's training data possible. Crossref has operated DOI, citation, and retraction infrastructure across competing scholarly publishers since 2000 under non-profit governance. ClinicalTrials.gov has held trial registrations across sponsors and jurisdictions since 2000, with imperfect enforcement but durable coverage. Nextstrain renders global pathogen evolution as a shared phylogeny across competing national agencies. GISAID carries genomic deposits across the same boundaries. UK Biobank and All of Us sustain participant cohorts, phenotyping, samples, and longitudinal data across institutions and decades.

Each of these works. None of them is a state layer.

They coordinate artifacts. The Protein Data Bank coordinates structures, not the claims structures bear on. Crossref coordinates DOIs, not the state of the work the DOIs identify. ClinicalTrials.gov coordinates trial registrations, not the state changes the trial outcomes should produce. The missing primitive is the reviewed change to what a field currently believes or can act on, with evidence, scope, provenance, confidence movement, signer, and downstream effects attached.

The precedent the Constellate architecture looks to is not the scientific information system but the content-addressed, append-only ledger. Git and Bitcoin coordinate state transitions over code and currency respectively; their value is that the history of changes is the canonical object, not any individual artifact. Constellate applies the same primitive to scientific claims: the reviewed, signed, replayable transition is what compounds across institutions and time, not the paper or dataset that triggered it.

Constellate is not a proposal to replace peer review. It is an architectural complement to whatever evaluation regime a corridor adopts. The protocol commits only to the operational primitive — a signed transition under a credentialed actor — and leaves the questions

of credentialing, lifecycle timing, and signature thresholds to the host foundation. eLife's review-only model, traditional pre-publication review, and post-publication review (PubPeer, Retraction Watch, F1000Research, ASAPbio, Review Commons) all record identically. The merge-authority debate is not resolved by Constellate; it is made legible. Signed transitions make the reviewer, scope, evidence, and decision rationale part of the public record where journal pipelines hide them.

1.1 A Morning in the Corridor

A Tuesday in March, two years after the first signed deposit. A reviewer at a participating academic medical center opens her queue at seven thirty. The substrate has accumulated forty-two proposed transitions overnight — most from agentic platforms running the BBB amyloid frontier, several from human researchers at affiliated labs, two from a clinical trial's interim DSMB rapporteur. Her queue is filtered to transitions touching findings she has signing authority on and whose confidence has shifted enough to warrant human attention. Twelve transitions today.

She opens the first. A perturbation experiment proposes that a cytokine signature previously attached to early-stage cognitive decline should split into two subgroup-specific signatures stratified on APOE genotype. The evidence packet pulls the perturbation data, a human-cohort cross-reference, a failed APP/PS1 cerebrovascular replication from a contract-research partner, and the agent's reasoning chain. She queries a clinical statistician on the stratification rationale, reviews fifteen minutes, signs.

The protocol routes the consequences. A Phase II trial whose endpoint depends on the original unscoped signature queues an APOE-stratified protocol-amendment pathway for its DSMB. A foundation portfolio review scheduled for Thursday now reflects the narrower scope. A draft review article being authored at another institution sees the change in its dependency graph and flags two paragraphs. A model that scrapes the substrate for training inherits the narrower claim from the next sync. None of this makes headlines.

She signs eleven more transitions before her coffee gets cold. Most are uncontested narrowings, two are flagged conflicts that route to a second signer, one she rejects with a citation. By noon, the substrate has integrated more state movement than her institution's email-and-PDF workflow would have moved in two months.

This is the operating image. The protocol is not a publication system. It is the layer where a field's working knowledge becomes something a reviewer reads, signs, and routes; something a downstream decision inherits without anyone having to reconstruct it; something the next generation of scientific intelligence can train against without inheriting only what closed labs chose to publish.

The protocol scales absorption, not generation. AI makes candidate science cheap; Constellate makes accepted science cumulative.

2 The Scientific State Transition

A scientific state transition is a reviewed update to what a field currently believes or can act on, with evidence, scope, provenance, confidence movement, and downstream effects attached. It is the basic operating unit of Constellate.

In the protocol, a state transition has seven parts:

1. **Prior state.** The state of the affected finding(s) at the moment the transition is proposed, identified by content hash.
2. **Proposed change.** The kind of update (asserted, reviewed, confidence revised, scope narrowed, retracted, replicated, contradicted) and its target.
3. **Evidence packet.** The source artifacts, measurements, datasets, code, and trajectory bearing on the change.
4. **Context and scope.** The population, model system, assay, comparator, endpoint, and sample size to which the change applies.
5. **Provenance.** The actor proposing, the actor reviewing, the institutional context, and the chain back to source.
6. **Review decision.** A signed acceptance, rejection, or qualified merge under governance.
7. **Replay effect.** The hash of the state after the transition, the affected dependencies, and the canonical event recorded in the append-only log.

A transition is the minimum object that can carry a real epistemic update across institutions, agents, and time. It is small enough that an ordinary scientist can read it. It is structured enough that another system can act on it. It is signed enough that downstream work can trace why a decision was made.

State transitions form an append-only log per frontier. The current state of any finding is the deterministic replay of its event history. Corrections enter as new events. History is preserved. A signed transition is the smallest object an agent can produce that the field can inherit.

This is the distinction that holds the rest of the architecture together. A dataset, a model prediction, a robotic run, a review comment, an agent's literature scan: none is state. Each becomes state only when it is converted into a proposed change, reviewed against a frontier, attested under governance, and recorded as an event future work can read.

3 The Constellate Ontology

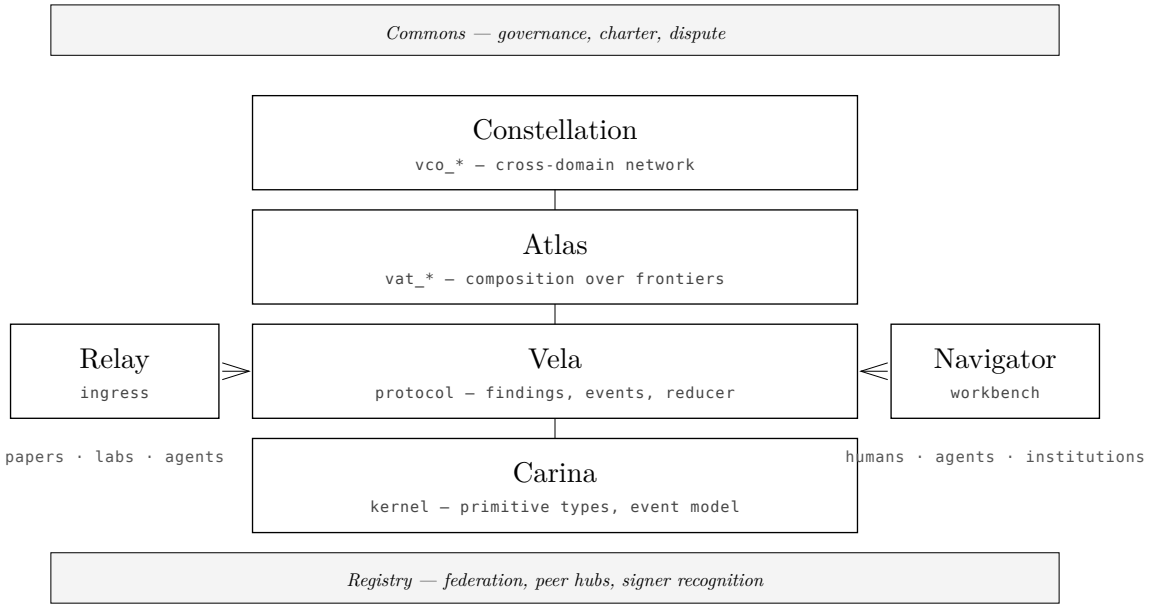

Figure 1: *The Constellate stack. Vertical layers compose; horizontal layers cross-cut. Each layer can be implemented multiple times, and each layer can be forked, with history and signer graph traveling along.*

Constellate is a stack of layers that separate so no single layer can fuse the system into a captured platform.

- **Vela** — the protocol. Finding bundles, proposals, canonical events, reducers, and the rules under which a state transition becomes part of a frontier's history. The substrate that must be open, governed, and forkable.
- **Carina** — the kernel. The type system and event model Vela enforces. Findings, evidence atoms, conditions, trajectories, negative results, replications, predictions, resolutions, artifacts, datasets, code artifacts, diff packs, conjectures, and proof packets. Versioned (current: v0.2 with selected v0.4–v0.5 extensions) and pinned per-frontier.
- **Atlas** — the composition layer. A living, versioned map over one or more frontiers with explicit bridges between them. Read-only over the frontiers it composes; type `vat_*`.
- **Constellation** — the cross-domain network. A graph of Atlases spanning scientific domains; type `vco_*` reserved in Carina v0.5, no instance ships yet.
- **Relay** — the adapter layer. Modules translating external scientific activity (papers, preprints, lab notebooks, model outputs, agent runs) into protocol proposals.
- **Navigator** — the workbench. The human and agent surface for reading frontier state, proposing transitions, reviewing queued proposals, signing acceptances. The product layer; multiple implementations against the same protocol.
- **Registry** — the federation layer. The canonical record of frontiers, signer identities, peer hubs, and inter-frontier dependencies. Currently point-to-point through peer hubs declared in each frontier's manifest; canonical Registry reserved for a future layer.

- **Commons** — the governance stewardship. A federated structure with nested enterprises (local frontier, federated hub, protocol consortium).

Each layer can be implemented multiple times. Each layer can be forked, with history and signer graph traveling along. The architecture's coherence comes from the separations, not from a single body holding all of them.

4 Protocol Mechanics

This section describes Vela's mechanics at the architectural level. The formal specification lives in the Vela protocol documentation (`docs/PROTOCOL.md`, `docs/STATE_TRANSITION_SPEC.md`). What follows is the part that has to be understood to evaluate the architecture; details of serialization, canonical JSON, and exact byte layouts are deferred to the spec.

4.1 The Finding Bundle

The finding bundle is the primary object Vela holds. Each bundle carries a bounded scientific claim with the evidence and provenance that support it. A bundle has a stable identifier of the form `vf_` followed by the full hex encoding of a SHA-256 digest computed over the normalized assertion text, assertion type, and provenance identifier — 64 hex characters, 256 bits of collision resistance. Surface contexts (UI, references in prose, conflict-detected event payloads) typically display the leading 16 hex characters as an abbreviation, but the canonical identifier is the full digest. The provenance identifier prefers DOI, then PubMed ID, then source title. Identifiers are therefore content-addressed: two systems that observe the same claim from the same source produce the same canonical identifier; abbreviations are display-only and resolve to the canonical form. The full-width choice follows Git's evolution from 7-character to 12-character to full-40-character commit identifiers as the codebase scaled, and ensures that a substrate intended to outlast decades plus agent-rate proposals retains a collision budget well above the birthday-attack threshold.

A finding bundle holds:

- The assertion (typed by Carina; e.g. mechanism, association, intervention, prediction)
- The supporting evidence atoms (`vea_*`)
- The condition records (`vcnd_*`) describing scope (species, assay, comparator, endpoint, sample size)
- The trajectory (`vtr_*`) describing how the finding was produced, where applicable
- The current confidence score (bounded scalar in [0.0, 1.0], interpreted as a subjective probability that the assertion holds under the recorded condition record, conditioned on the supporting evidence; the protocol does not commit to a frequentist interpretation, and the downstream reputation and calibration machinery in §5.6 and §11.6 treats the value accordingly; credal sets — interval-valued confidence per Walley's imprecise-probability framework — are future work)
- The dependency edges to other findings (within the frontier and to external frontiers)
- The signature graph (which actors have signed the finding under what scope)
- The flags (signature thresholds, access tier, status)

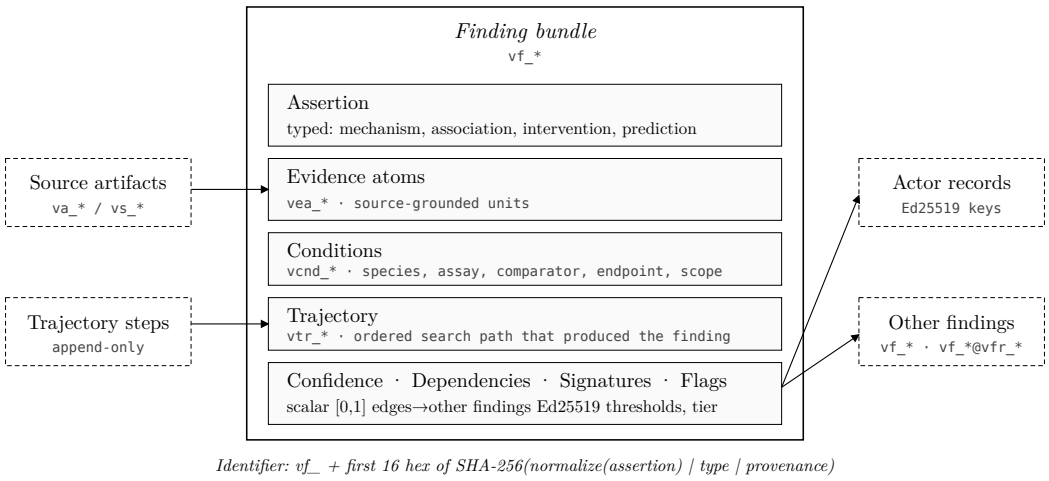

Figure 2: *The finding bundle. The bundle is the unit a reviewer reads; its identifier is content-addressed over assertion, type, and provenance, so two systems that observe the same claim from the same source produce the same identifier.*

The bundle is the unit a reviewer reads. A frontier is a set of bundles plus the event log that produced them.

4.2 Frontiers

A frontier is a bounded, reviewable scientific question. Examples: blood-brain-barrier dysfunction in early Alzheimer's disease; perturbation-response signatures in pediatric high-grade glioma; pathogen surveillance across a regional wastewater network; the synthesis of a defined inorganic crystal class.

Each frontier has a stable identifier `vfr_*` derived from the SHA-256 of its genesis event. A frontier holds its findings, its event log, its governance policy, its peer declarations, and its dependency pointers to other frontiers. Frontier identity is content-addressed: two frontiers cannot collide because their genesis events differ.

A frontier is the unit of replay determinism. Given an event log and a pinned Carina kernel digest, two independent reducer implementations must produce byte-identical finding-state digests. This invariant is load-bearing: it lets federation, mirroring, and forking work without trust in the original operator.

4.3 Proposals, Events, and the Reducer

Vela holds a strict state-transition discipline: a proposed change is reviewed, accepted as a canonical event, applied by a deterministic reducer, and rendered as frontier state that feeds back into the context for the next proposal.

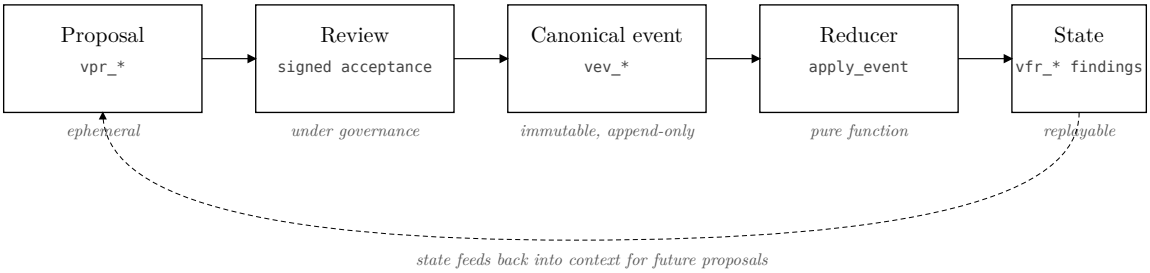

Figure 3: *The state-transition lifecycle. Proposals are ephemeral working objects; only reviewed, signed events enter the canonical log; the reducer applies them deterministically.*

A **proposal** (`vpr_*`) is a working object, not yet canonical. It carries an actor, a kind (e.g. `finding.asserted`, `finding.reviewed`, `finding.rejected`, `negative_result.asserted`, `replication.deposited`), a target reference (a finding ID, an artifact ID, a bridge ID), a timestamp, a reason, and a kind-specific payload. Proposals are ephemeral until accepted as events.

A **canonical event** (`vev_*`) is the immutable record that an accepted proposal has been merged into the frontier. Events carry: the kind, the target, the actor, the timestamp, the hash of the frontier state before, the hash of the frontier state after, the reviewer reason, and the kind-specific payload. Events are content-addressed and form an append-only log.

The **reducer** is a pure function from `(state, event) → state`. It dispatches on the event's kind and mutates the frontier state deterministically. As of Carina kernel v0.6 (Vela v0.500), 26 mutation kinds are defined in the `REDUCER_MUTATION_KINDS` constant at `crates/vela-protocol/src/reducer.rs`. A small number of additional event kinds (notably the federation-observation kinds and `attestation.recorded`) are dispatched by `apply_event` but are intentionally excluded from `REDUCER_MUTATION_KINDS` because they do not mutate finding, negative-result, trajectory, artifact, or evidence-atom state. Each kind has a documented semantics in `docs/STATE_TRANSITION_SPEC.md`.

The reducer's determinism is what lets the protocol survive federation. Independent implementations in different languages must produce byte-identical state from the same event log. Currently, the Rust reducer is reference-grade; the Python reducer hydrates most kinds but does not yet cover the full v0.55+ trajectory and evidence-atom materializers; the TypeScript reducer covers a subset against fixture data. Cross-implementation reducer parity is an ongoing engineering commitment, not a finished property.

4.4 Replay and the State Hash

The current state of a frontier is the deterministic replay of its event log under the pinned Carina kernel. Each event's `before_hash` must equal the preceding event's `after_hash` for that target. If the chain breaks, replay fails — the protocol refuses to materialize state from an inconsistent log.

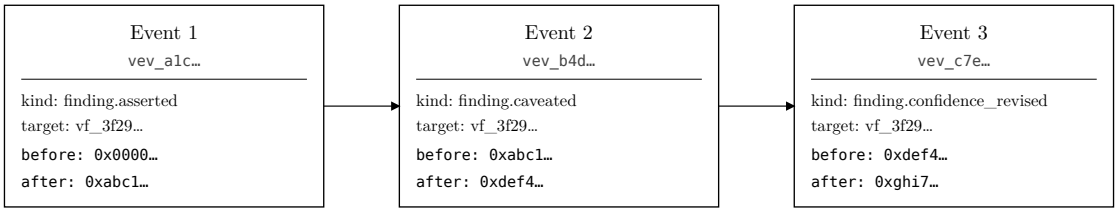

Figure 4: *The append-only event chain. Two independent reducer implementations on the same event log produce byte-identical finding-state digests; this is what lets federation, mirroring, and forking work without trust in the original operator.*

The state hash is computed over the canonical JSON of the frontier's findings, sorted by ID, with all fields ordered deterministically per the canonicalization rules specified in *The Vela Protocol Specification* §10. The protocol adopts RFC 8785 (JCS, JSON Canonicalization Scheme) as its canonicalization standard: lexicographic key ordering, UTF-8 NFC string normalization, IEEE 754 double-precision number serialization with no trailing zeros and no exponent for integer-valued floats, escaped control characters, and explicit treatment of absent versus null fields. Two implementations agreeing on the kernel version and the event log under JCS canonicalization must agree on the hash. Disagreement is a protocol bug or an implementation defect. The cross-implementation parity gaps reported in the specification's §9 trace primarily to incomplete JCS conformance in the Python and TypeScript reducers — the canonicalization specification is the precondition for the 95% conformance threshold.

This determinism is the federation primitive. A mirror does not have to trust the original operator to replay the same state. A fork carries its history because the history is content-addressed. A regulator inspecting a state history does not need the original signing keys to verify the chain; only the public keys of the signers and the pinned kernel.

4.5 Cross-Frontier References

A finding in one frontier may bear on a finding in another. References use the form `vf_<id>@vfr_<id>`. Resolution requires the dependent frontier to be declared in the host frontier's manifest with a pinned snapshot hash for integrity. Pinning is mandatory under strict mode and strongly recommended elsewhere; unpinned cross-frontier references are the open door for dependency-forgery attacks and the protocol prefers to refuse resolution rather than silently follow an unverified pointer.

Cross-frontier references are the seed of Atlas composition and of the Constellation layer. They are also the surface where conflicts can arise: two frontiers may carry inconsistent views of the same underlying object. The protocol records this as a structural state rather than collapsing to one canonical view (see §7.3, Conflict Resolution).

4.6 Fork Choice

The reducer is deterministic per event log: given the same log and the same Carina kernel pin, two implementations produce byte-identical state. The question is what happens when two

peers in a federated frontier accept different proposals on the same target finding at the same chain position. Both events are valid in isolation; both extend a chain whose prior head was the same after-hash; the two chains have now diverged.

Vela's fork-choice rule is **first-canonical-wins under maintainer arbitration with explicit conflict events on the losing chain**. Mechanically: each peer maintains its own canonical view, accepting events in the order it observes them and recording any conflicting event from a peer as `frontier.conflict_detected` (see §7.3). The frontier's maintainer quorum (governed by the Registry Governance Policy, §7.1) holds canonical-merge authority and resolves the divergence by signing a `frontier.conflict_resolved` event that designates which event takes canonical position; the other event remains in the log as evidence of the divergence and is reachable through replay but does not contribute to canonical state. Peers re-converge on the maintainer-signed resolution.

This is closer to Git's explicit-merge model than to Bitcoin's longest-chain or Ethereum's LMD-GHOST. The protocol does not assume an honest majority of compute or stake; it assumes a governance quorum that can be held accountable through forkability. Where the quorum itself is contested, the protocol does not collapse to a single view: per §7.3, plural canonical views are a recordable state, and a frontier that cannot reach maintainer consensus persists as a divergent state that downstream consumers see and decide how to interpret. Convergence is not assumed; legibility is.

Two consequences. First, fork choice is a governance problem, not a consensus problem — the protocol provides the mechanism for recording, arbitrating, and surfacing divergence, but the decision lives with the credentialed reviewers a frontier has named. Where the maintainer quorum itself is compromised or captured, fork-choice degrades to whatever the captured quorum signs; the §7.1 signer-independence assumption is the structural premise that this degradation does not happen invisibly, and the corridor's forkability commitment (§7.5) is the recourse when it does. Second, the cost of an adversarial peer that signs conflicting events is bounded: the events are recorded, the peer's signing record carries the conflict, the conflict propagates into the reputation composite of §11.6, and revocation (§6.4) closes the prospective trust. The protocol commits to a structural penalty stronger than reputation drift alone: a verified equivocation by a credentialed signer triggers automatic credential suspension on the corridor, recoverable only through quorum re-attestation by a *fresh* maintainer set. Cryptographically, "fresh" means the re-attesting quorum must overlap the suspending quorum by no more than $\lfloor k/2 \rfloor$ members (where k is the threshold size), the re-attestation must itself cross the corridor's transparency-log witness (§7.1.1), and the re-attestation event must be signed by at least one institutional actor that was not part of the suspending quorum. Equivocation cannot be prevented cryptographically, but it can be made expensive, inspectable, and operationally costly to repeat.

Worked example — partition-induced divergence. A corridor with maintainer quorum `{M1, M2, M3, M4, M5}` is partitioned by a network event: `{M1, M2, M3}` can communicate with each other and with peer hubs in region A; `{M4, M5}` can communicate with each other and with peer hubs in region B. A finding `vf_3f29…` at chain position N has after_hash `0xabc1`. While partitioned, region A's maintainers sign `finding.reviewed` (kind X) on `vf_3f29…`, producing event `vev_p1` with after_hash `0xdef4`. Region B's maintainers sign `finding.caveated` (kind Y) on the same target at the same prior position, producing event `vev_p2` with after_hash `0xfed3`. When the partition heals, peer hubs in both regions observe the divergence: each peer's local

log carries one of the two events; the other event arrives via cross-peer state-hash exchange and raises `frontier.conflict_detected` events at all peers. The full quorum (now reachable) reviews the divergence and signs a `frontier.conflict_resolved` event designating one of the two events as canonical at position N+1; the other remains in the log as evidence of the divergence and is reachable through replay but does not contribute to canonical state. Downstream consumers reading the frontier after position N+1 see the resolution and the conflict simultaneously. If the partition is adversarial (one set of maintainers equivocated rather than partitioned), the equivocators' credentials are automatically suspended per the rule above; the recovery path is a fresh-maintainer-set re-attestation.

5 Evidence and Provenance

A state transition is only as inheritable as the evidence it carries. Vela's evidence model is structured so that proposed changes can be inspected back to source, replicated against the original observations, and challenged when the evidence does not support the scope of the claim.

5.1 The Source–Artifact–Atom Chain

Evidence enters the protocol through four layered primitives.

Sources (`vs_*`) are the upstream origins of scientific assertions: papers, preprints, lab notebooks, model outputs, expert assertions, database records, data releases, researcher notes. A source is registered, identified, and pointed to. The protocol does not host sources; it references them and depends on them being durable.

Artifacts (`va_*`) are content-addressed commitments to bytes or pointers: protocols, trial records, supplements, notebooks, code, model outputs, datasets, source files. Artifacts carry a kind, a content hash (SHA-256), a storage mode, a locator, the findings they target, and an access tier (public, restricted, or classified).

Datasets (`vd_*`) and **code artifacts** (`vc_*`) are specialized artifact types that anchor empirical claims to the bytes that produced them. A dataset is a versioned, content-addressed reference to data distinct from the paper that reports it. A code artifact is a pointer to a specific region of source code at a specific git commit — not a repository, a specific commit. These let claims be tied to the data and code that produced them rather than to the prose summary; a subsequent reader can re-execute the analysis.

Evidence atoms (`vea_*`) are materialized, source-grounded units of evidence that bear on a specific finding. An atom is a span of text, a measurement, a table cell, a figure, or an analytic output, anchored to its source and locatable for inspection. Atoms are the unit a reviewer reads when deciding whether a finding's evidence supports its scope.

The chain runs source → artifact → atom → finding. A finding can be traced from its current state to the canonical event that produced it, to the atom that supplied the evidence, to the artifact that carries the atom, to the source the artifact references.

5.2 Conditions and Scope

A finding is not a sentence; it is a scoped claim. **Condition records** (`vcnd_*`) define the boundaries within which a finding holds: species, assay, comparator, endpoint, sample size,

population stratification, model system. Two findings that look identical in their assertion text may be distinguished by their conditions.

The condition model is what lets corrections travel narrowly. A finding asserted across a broad population can be amended to apply only to a subgroup without retracting the underlying assertion. The protocol records the scope change as an event (`finding.caveated` or `finding.entity_resolved`) and the downstream findings inherit the narrowed scope.

5.3 Trajectories

A **trajectory** (`vtr_*`) is the ordered search path that produced a finding — the sequence of intermediate steps, decisions, and discarded branches that led to the result. Trajectories are append-only: steps are added via `trajectory.step_appended` events; the trajectory itself has `created`, `reviewed`, and `retracted` events.

Trajectories carry information that papers usually omit. The branches that did not work, the analytic choices that were considered and rejected, the parameter sweeps, the assumptions tested and discarded. In a paper system, trajectory information lives in a researcher's notebook or in the tacit experience of a lab; in the protocol, it is structured and inspectable.

Trajectories are not required. A finding can be asserted without one. But trajectories are how negative-result information enters the record at the granularity that lets the next researcher avoid repeating the failure.

5.4 Negative Results and Replications

A **negative result** (`vnr_*`) is a first-class object: an experiment or trial that did not support its hypothesis. The protocol supports two flavors — registered trials (with power, effect-size bounds, and pre-specified analysis) and exploratory experiments (with reagent, condition, and outcome) — and gives both the same lifecycle events (`negative_result.asserted`, `.reviewed`, `.retracted`) and the same review machinery as positive findings. Negative results are the deposits the file-drawer problem suppresses; the protocol makes them evidence rather than moral obligation.

A **replication** (`vrep_*`) is a specific replication attempt at a target finding under named conditions. It carries the target finding, the actor, the conditions, and the outcome — support, contradict, or partial confirmation. Multiple replications under different conditions coexist as a structural set rather than collapsing to a replicated/not-replicated boolean. Replication is too structured to be a flag; it is a relationship between a finding and a specific attempt under specific conditions.

5.5 Forward-Pointed Claims

A **prediction** (`vpred_*`) is a falsifiable claim about a future observation, scoped to existing findings and tied to a registered actor. A **resolution** (`vres_*`) closes a prediction by recording the actual outcome. Together they form the protocol's epistemic accountability ledger: a model that proposes interventions can be scored on its predictions; its calibration record becomes part of its actor record. Calibration is a first-class signal, not an aggregate computed elsewhere.

A **conjecture** (`vcj_*`) is a signed forward institutional claim with a structurally enforceable falsification path. The falsification path is expressed as a typed predicate over future protocol events — a Boolean composition of conditions on resolutions, replications, negative results,

and confidence revisions, evaluated by the reducer against the conjecture's specified evidence scope. A conjecture might commit, for example, that "if three independent replications under condition records of class C resolve negative against the dependent finding within 24 months, the conjecture is falsified." The predicate language is **quantifier-free first-order logic over typed event-pattern atoms with bounded temporal scope**, defined in `crates/vela-protocol/src/conjecture/predicate.rs`. The restriction guarantees decidability and termination: every predicate is evaluable in time polynomial in the size of the conjecture's evidence scope, and the reducer's evaluation is deterministic across implementations. This expressivity bound distinguishes Constellate's conjectures from the "performative falsification" pattern that mimics rigor without enabling it.

A **proof packet** (`pp_*`) is a hash-stable, signature-verifiable receipt for external verification of frontier state. A proof packet seals one frontier's state at one point in time — the artifact a regulator, funder, or downstream institution receives when they need to inspect what a frontier currently knows without taking on the operational burden of replaying the full event log. State crosses institutional boundaries without forcing the receiving institution to adopt the full protocol stack.

5.6 Diff Packs

A **diff pack** (`vsd_*`) is a reviewable set of frontier state changes grouped for human adjudication. When multiple proposed changes are best reviewed together — because they share an underlying source, an operation class, or an affected dependency cluster — they enter the protocol as a single pack. The pack records the member proposals, the operation class, the affected findings, the validation results, and a pack-level review decision.

Diff packs are the unit that scales human reviewer attention. A reviewer adjudicates coherent batches of related transitions rather than every transition in isolation. The pack-level decision is recorded as an event (`diff_pack.released`, `diff_pack.reviewed`) so the structure of the batched review is itself inspectable.

5.7 Relationship to Prior Structured-Assertion Work

Several existing systems propose structured representations of scientific assertions; the comparison below clarifies what Constellate inherits, what it diverges from, and where the integration points sit.

System	Primary unit	Carries signed state changes?	Replay determinism	Forkable	Closest Constellate analogue
Nanopublications (Groth/Gibson/ Velterop 2010; Kuhn et al. 2013, 2021)	Atomic citable assertion with attribution and provenance	No (immutable assertions; revisions are new nanopubs)	N/A (no event log)	Partial (republication-with-attribution via Trusty URIs and Nanopub Network mirrors, but no shared event chain — forks share content addressing without sharing history)	Finding bundle (`vf_*`) plus a signed `finding.asserted` event
W3C PROV-O / PROV-DM	Entity / Activity / Agent triples for provenance	No (descriptive provenance only)	N/A	No	Source–artifact–atom chain (§5.1); PROV-O subset can be derived from event log
Research Objects (Bechhofer et al. 2013; RO-Crate)	Aggregations of artifacts with manifests	No (package state, not claim state)	No	No	Closest to Atlas composition (`vat_*`) over artifacts, but read-only
ORKG triples (Open Research Knowledge Graph)	Structured claim triples extracted from papers	No (graph relationships; no event log)	No	No	Cross-frontier references (§4.5), without signatures or replay
Constellate	Reviewed, signed, replayable state transition	**Yes** (canonical event log with reducer)	**Yes** (per-frontier byte-identical)	**Yes** (history travels with fork)	—

The integration points matter as much as the differences. A Vela finding bundle can be exported as a PROV-O triple (Activity = the canonical event that produced the bundle; Entity = the bundle itself; Agent = the signing actor) for consumers operating in PROV-O-native systems; a nanopublication can be ingested through the Relay layer (§3) as a source artifact and become an evidence atom anchored to a finding; an ORKG triple becomes a candidate dependency edge under review; a Research Object package can compose into an Atlas as a read-only artifact bundle. The architectural commitment Constellate adds across all four predecessors is the signed state-transition event and the replay invariant — the move from "structured assertion" to "structured assertion under a governed, replayable, forkable history."

6 Identity and Signing

State only inherits if signatures inherit. Vela's identity model is built around Ed25519 keypairs tied to registered actor records, with optional ORCID anchoring, access tiers, and revocation.

6.1 Actor Records

An actor record carries:

- A stable namespaced identifier (e.g. `reviewer:will-blair`, `lab:haverford-chemistry`, `agent:scientist-v0.2`)

- A public key (hex-encoded Ed25519, 32 bytes)
- The signing algorithm (currently `ed25519`)
- A creation timestamp
- An optional tier (e.g. `auto-notes` for lightweight auto-apply)
- An optional ORCID identifier
- An optional access clearance (Public, Restricted, Classified)
- An optional revocation timestamp and reason

Actor records are themselves recorded in the protocol. Registering an actor is a state transition that other actors can sign. The Registry layer (see §7) holds the federated record of actors across hubs.

6.2 Signatures

A signature is an Ed25519 signature over the canonical JSON of the object being signed (deterministic, sorted-key serialization). Signatures carry: the object's identifier, the signer's public key, the signed-at timestamp, the algorithm, and the signature bytes.

The protocol verifies signatures against the registered public key. Once an actor is registered, any canonical event referencing that actor under strict mode must carry a verifiable signature. This is the protocol's defense against signature forgery: a signature that does not verify is not a signature.

Every signed object — actor records, governance events, finding bundles, canonical events, proof packets — binds to its containing frontier identifier (`vfr_*`) as part of the signed payload. A signature produced over an actor-record update in frontier A cannot be replayed against frontier B because the frontier ID is part of the canonicalized JSON the signature covers. This binding is mandatory; signatures over payloads that omit the frontier ID are rejected at signature-verification time. The defense is structural rather than procedural: cross-frontier signature replay is not addressed through detection alone but through making the replayed signature cryptographically invalid in the target frontier.

6.3 Signature Thresholds

A finding can declare a signature threshold via `flags.signature_threshold = k`, requiring k distinct valid signatures from registered actors before the finding is accepted into canonical state. Thresholds are how the protocol expresses "this finding requires multi-reviewer consensus" at the object level, without requiring multi-signature on the event itself.

This is structurally important. Some findings — clinical guidelines, safety-relevant transitions, high-dependency hubs — should not be merged on a single reviewer's signature. The threshold mechanism makes the requirement explicit in the object's flags, inspectable by any downstream reader, and enforceable by any independent implementation.

6.4 Revocation

Keys can be compromised. Actors can be deregistered. Institutions can lose trust in past signers.

The protocol records revocation as an event: an actor's record is amended with a `revoked_at` timestamp and a `revoked_reason`. Signatures produced before the revocation timestamp remain valid for the historical record; signatures produced after the revocation timestamp are rejected. The historical state is not rewritten — only the prospective trust changes.

This is the protocol's compromise between correction and history. A compromised signer's past acceptances are not retroactively voided; they are inspectable, and downstream consumers can decide whether to re-review claims that depend on that signer. A revoked signer can no longer mint new acceptances.

6.5 Key Rotation

Revocation (§6.4) handles the case of a compromised or retired actor. Rotation handles the everyday case: a researcher replaces a laptop, an institutional HSM is cycled on its annual policy, a postdoc finishes a fellowship and moves to a new lab. The protocol distinguishes the two cleanly.

An actor's identity is the stable namespaced identifier (e.g., `reviewer:will-blair`), not any particular keypair. The actor record carries a list of authorized signing keys, each with its own `created_at` and optional `retired_at` timestamp. Adding a new key is a state transition signed by an existing authorized key on the same actor (or, for first-time actors, by an institutional sponsor whose key already appears in the corridor's credentialed pool). Retiring an old key sets its `retired_at` timestamp; signatures produced before retirement remain valid, signatures produced after are rejected.

The mechanism is structurally similar to Sigstore's transparency-log key bindings and to WebAuthn's credential rotation. It preserves three properties: an actor's reputation graph (signed events, calibration record, signer history) is identity-scoped, not key-scoped, and survives any number of key rotations; rotation requires authorization from an existing key on the same actor, so a stolen single key cannot rotate the actor into the attacker's control; and the rotation chain itself is recorded as protocol events, so a downstream consumer can audit when each key was added and retired.

Institutional key custody handles the harder case where the actor is a lab or organization rather than a researcher. A lab's actor record can authorize multiple subkeys held by the PI, lab manager, and one or more current postdocs, each with its own retirement timestamp. A postdoc leaving the lab triggers a `key.retired` event on their subkey; the lab's actor record continues uninterrupted.

Lost or compromised institutional keys without a recoverable predecessor signature require a governance-layer recovery path. The default policy a corridor inherits unless it overrides specifies: a recovery attestation requires three signatures from a five-person recovery panel composed of (i) one institutional officer of the affected actor's home institution (e.g., research dean or office of research integrity), (ii) two corridor-credentialed reviewers outside the affected actor's lab and outside the home institution, and (iii) up to two additional officers from peer institutions in the credentialed pool. The attestation must be posted to the corridor's transparency log (§7.1.1) and a public-notice window — defined as **14 business days excluding standard institutional recess periods (mid-June through August, mid-December through mid-January, and any recess period the participating institutions publish in their academic calendars)** — must elapse before the recovery takes effect, during which any credentialed reviewer in the corridor may file a `recovery.contested` event that pauses the recovery pending maintainer-quorum review. The business-day definition addresses the academic-calendar reality that field rotations, conferences, sabbaticals, and standard recess periods can leave eligible contesters unreachable on naive calendar timing. The recovery does not retroactively re-validate signatures from the lost key; the historical

record remains under the old keys with their original validity intervals. A corridor's RegistryGovernancePolicy (§7.1) may strengthen these defaults but cannot weaken them.

6.6 Access Tiers

Not every finding is fit for public deposit. Clinical records carry patient-level information; some protein designs and synthesis routes are dual-use; certain pathogen sequences raise biosecurity concerns. The protocol carries an access-tier dimension on artifacts and events.

Three tiers are defined: Public (default), Restricted (trusted-reviewer rooms), and Classified (regulator escrow only). The state transition itself remains inspectable — its hash, signer, and dependency movement are public. The underlying evidence may be tiered: hashes and signatures live in the open record, full evidence content lives behind the appropriate access tier.

This is the protocol's commitment to "public because the state change is inspectable, not because every underlying file is globally visible." Open by default; tiered where required; the decision auditable in either case.

6.7 The Agent-Attestation Tier

Agent operators are a first-class actor category, but the agent-attestation tier was named in earlier sections (§11.1, §12.3) without being specified as a Carina primitive. This subsection commits the specification.

An agent actor record extends the actor schema (§6.1) with three additional fields. A `tier` field set to `agent` distinguishes it from `human` or `institution` tiers at the schema level, not just by string convention; downstream consumers and reducer arms can branch on the tier without parsing identifiers. An `operator` field carries the stable identifier of the accountable human or organization — the named principal who is legally and reputationally answerable for the agent's deposits. The operator must themselves be a credentialed actor in the corridor; an agent without a named operator cannot be registered. A `stack_manifest` field carries a content-addressed reference to the agent's stack description — model identifiers and versions, tool suite, prompt templates, training-data cutoffs, and any retrieval-augmented sources. The manifest does not have to be public (it may be tiered as Restricted), but its hash must be inspectable so two deposits from "the same agent" are verifiably from the same configured stack.

Agent deposits use a distinct event kind, `agent_attestation.deposited`, separate from `finding.asserted`. The semantic difference is canonical-merge authority: an `agent_attestation.deposited` event enters a pending-review state regardless of the agent's reputation; canonical merge into `finding.asserted` requires a human or institutional signature under §6.3 signature thresholds. The agent's deposit is durable and citable as an attestation but does not, on its own, move canonical state.

A submission stake is required for agent deposits and forfeited on rule-based rejection. The stake is denominated in the agent's reputation graph (acceptance rate, calibration score) rather than in currency; an agent that floods the substrate with low-quality deposits sees its acceptance rate fall, its routing priority drop, and eventually its registration suspended through governance.

A new operator with no reputation cannot bootstrap on stake alone — the cold-start problem requires explicit sponsorship. An uninvited registration is accepted only when at least two credentialed reviewers in the corridor (each with their own non-trivial reputation history and no shared institutional affiliation with the new operator) sponsor the registration. Sponsoring

reviewers stake their own reputation against the new operator's first 100 deposits: if those deposits are rejected at a rate substantially above the corridor median, the sponsors incur a reputation penalty proportional to the rejection volume. Sponsors also receive a positive return — a small share of the sponsored operator's accepted-deposit reputation credit (specifically, 15% of the sponsored operator's calibration-weighted contribution score during the probationary window) accrues to the sponsor's own reputation graph. The asymmetric incentive (liability proportional to rejection volume, gain proportional to acceptance-weighted-by-calibration quality) is designed so sponsorship is a continuously-priced bet rather than pure liability. Without this positive return, the equilibrium degenerates to "no one sponsors anyone new" and the operator set calcifies around founders. New operators enter a 90-day probationary period during which their deposits are subject to elevated rule-based screening and human-review sampling. If a sponsor is later revoked or suspended, their sponsored operators are not automatically expelled but lose their sponsorship cushion and must either secure a replacement sponsor within 30 days or enter a remedial review pool.

Multi-agent deposit pipelines — a retrieval agent feeding an extraction LLM feeding a tool-use verifier feeding a human reviewer — collapse in this specification to one accountable operator and one stack manifest at the surface, but the protocol preserves the pipeline structure through trajectory steps (`vtr_*`). The trajectory primitive carries two structurally distinct event kinds: `trajectory.step_appended` for *scientific search-path* steps (branches tried, parameter sweeps, analytic choices considered and rejected — the original semantics) and `trajectory.pipeline_step` for *agent-handoff* steps (agent A produced this intermediate output and handed it to agent B — the pipeline-provenance semantics). Both kinds attach to the same `vtr_*` trajectory object but the reducer treats them differently: scientific steps contribute to the finding's epistemic history; pipeline steps contribute to the deposit's accountability chain. Each pipeline step records the contributing agent, its sub-stack-manifest hash, the intermediate output's content hash, and the handoff to the next agent in the pipeline. The canonical deposit carries one operator and one composite stack manifest; the trajectory carries both the inspectable scientific history and the inspectable pipeline composition. This lets a regulator or auditor examine which agent produced which intermediate output without exposing the protocol's canonical surface to multi-actor accountability ambiguity, and without conflating "what we tried experimentally" with "which model produced what output."

Different agent roles have different protocol surfaces. A **proposer agent** that submits `finding.asserted` candidates differs from a **reviewer agent** that signs adjudications and from a **bridge-detector agent** that proposes cross-frontier links. The agent-attestation tier carries an additional `role` field with these values; role-specific stake levels, sponsorship requirements, and merge-authority rules are defined by the corridor's RegistryGovernancePolicy. The protocol does not collapse all agents into one tier.

A reference `RegistryGovernancePolicy` template ships with v0.500 covering default sub-tier stake levels, sponsorship thresholds, and merge-authority rules. The template is a *default*, not a normative spec — corridors may fork it and tune values, but the template ensures that early corridors do not have to invent these parameters from scratch and that variations across corridors are inspectable as deltas from a known starting point. The protocol commits to publishing **stake-velocity** (how quickly an operator's stake recovers after a rejection penalty) and **reputation-anomaly metrics** (deviation from corridor-median acceptance patterns, sponsor-cluster concentration, suspicious-deposit-velocity flags) as canonical state surfaces,

so adversarial-detection signals are themselves inspectable rather than living in a Registry operator's private dashboard.

This specification is provisional. Production deployment will surface operational requirements — telemetry standards, audit logs, model-update notification, agent-handoff between operators, sub-tier merge-authority economics — that this paper does not yet address. The Vela protocol crates ship the actor-record extension fields at v0.500+ (Carina v0.6); the event-kind specification, stake mechanism, sponsorship rules, and trajectory wiring are in active design.

6.8 The Agent Read Surface

The agent-attestation tier specifies how agents write to the substrate. Production agent integration also requires a read surface — how an agent queries frontier state, walks dependencies, retrieves evidence atoms, and discovers candidate transitions to propose.

The read surface ships in two layers. **Synchronous query** is a typed GraphQL-style API exposed by every Vela instance: an agent can request a finding's current state by ID, walk its dependency edges within a depth limit, retrieve the evidence atoms backing the finding (subject to the agent's access tier — Public tier by default; Restricted tier requires the agent's operator to be credentialed at that tier), inspect the signature graph, and read the conjecture register attached to the finding. Queries are read-only and idempotent; they do not produce protocol events.

Asynchronous frontier subscription is a streaming interface: an agent subscribes to a frontier (or a subset of findings within a frontier) and receives canonical events as they enter the log, filtered by event-kind and access-tier eligibility. Subscriptions are signed by the agent's operator, rate-limited per corridor governance, and serve as the integration point for agents that maintain their own internal state in sync with frontier canonical state.

Access tiers govern read as well as write. A public-tier agent sees hashes, signers, dependency movement, and Public-tier evidence content; a Restricted-tier agent additionally sees Restricted-tier evidence content for findings within the agent's operator's clearance; Classified-tier evidence is never exposed through the agent read surface and requires a separate regulator-escrow access flow. The agent's reads are themselves recorded as `agent.read` events for audit purposes, with the rate of recording calibrated to balance audit completeness against query cost.

The read surface is specified in `crates/vela-protocol/src/agent_read.rs` and ships with the v0.500+ release. SDK bindings for Python and TypeScript are part of the v0.500 milestone.

7 Governance and Federation

A protocol that holds scientific state has to hold it under governance. The protocol layer enforces what is structurally enforceable — signature verification, replay determinism, append-only chains, content addressing. Everything else lives in the governance layer above it.

7.1 Registry Governance Policy

Each frontier declares a `RegistryGovernancePolicy` (`vgp_*`). The policy specifies:

- The frontier the policy applies to
- The current owner epoch (the generation of authorized owners)
- The bootstrap epoch (the founding configuration)

- A rotate quorum: threshold, eligible actors, role constraints, attestation TTL
- Optional emergency and policy-update quorums

Owner rotation is governed: a sufficient quorum of eligible actors must attest to the rotation within the TTL window. Theorem 16, formally proved in Lean (`lean/Vela/GovernedQuorumSoundness.lean`), states informally:

> *If `verify_quorum(policy, proposal, attestations)` returns `true`, then there exist at least `policy.threshold` distinct actors `{a₁, …, aₖ}` such that each `aᵢ` (i) appears in `policy.eligible_actors`, (ii) is not revoked at `proposal.timestamp`, (iii) satisfies `policy.role_constraints` if any, and (iv) has produced a valid Ed25519 signature `attestations[aᵢ]` over the canonicalized proposal that verifies against the public key registered to `aᵢ` at `proposal.timestamp`.*

The theorem's hypotheses are: the Carina kernel digest pinned at proposal time is honored, the actor registry's revocation state at proposal time is accurate, and the signer-independence assumption (next paragraph) holds. The theorem does not, and cannot, prove the independence hypothesis itself.

The honest limit: no protocol survives compromise of the threshold authority set. Defense is in policy design — large enough quorums, role diversity, institutional stewards, short attestation TTLs — not in any single technical mechanism.

Theorem 16 establishes that `verify_quorum` accepts only if the formal predicate is satisfied; it does not, and cannot, prove **signer independence**. The protocol assumes that the eligible actors named in a `rotate_quorum` are genuinely distinct decision-makers, not three keys held by the same person or three labs in the same captured network. This independence assumption is the load-bearing premise behind every quorum-based defense and the protocol cannot enforce it through cryptography alone. Structural checks approximate independence: institutional diversity (eligible actors named across distinct organizations), geographic diversity (signers in distinct jurisdictions where applicable), role diversity (mixing PI-class signers with statistician-class and provenance-auditor-class), and external observation through the transparency-log witness specified in §7.1.1. Where any of these structural checks fail, the protocol's governance guarantees degrade gracefully to the strength of the smallest independent decision unit.

7.1.1 7.1.1 Transparency-Log Witness

Every governance event — actor registrations, key rotations, quorum proposals, owner-rotation attestations, kernel-pin updates, peer-hub declarations, recovery panel attestations — must be cross-posted to a transparency log before downstream consumers honor it under strict mode. The witness gives the broader community a tamper-evident record outside the corridor's own infrastructure, so a captured maintainer quorum cannot silently rewrite governance state without leaving an inspectable trail.

The protocol commits to Sigstore Rekor v2 as the default transparency-log implementation, with Sigsum-style witness cosigning as the assumed architecture: each cross-posted event is signed by the corridor's governance quorum, submitted to the log, and counter-signed by a set of independent witnesses (each running a verifiable copy of the log) within a defined freshness window. **The freshness window is set per event class and bounded by the protocol's default policy**: 4 hours for routine governance events (actor registrations, key rotations, peer-

hub declarations); 24 hours for kernel-pin updates and major governance-policy revisions; 7 days for actor registration onboarding flows; 1 hour for emergency-quorum events. A reducer running under strict mode accepts a governance event only if (i) the event verifies against the corridor's governance policy, (ii) the event's inclusion proof in the transparency log verifies, and (iii) at least two independent witness cosignatures are present, verify against the witnesses' registered public keys, and were produced within the event-class freshness window.

Candidate witness operators for the first corridor (each in active conversation, none under signed commitment as of v0.1): a university-operated witness through one of the Trustworthy AI consortium partners; an open-infrastructure witness through the Internet Archive's TimeMachine project or Sigsum's existing public witness pool; a foundation-aligned witness through Code for Science & Society or the Open Source Security Foundation; a civil-society witness through an entity like EFF or the OpenStack Foundation's emerging audit infrastructure. The corridor's initial witness set is selected by the steward in consultation with the corridor maintainer quorum, requires at least two independently-operated entities in distinct legal jurisdictions, and is published as a governance event before the first proof packet is signed.

The schema of witness entries follows the Sigstore Rekor v2 entry format: a hashed leaf containing the canonicalized governance event payload, a Merkle tree inclusion proof, the corridor's governance signature, the witness cosignatures, and a freshness timestamp. The corridor's RegistryGovernancePolicy declares which witnesses count as eligible and the minimum number required; the protocol's default policy mandates at least two witnesses operated by distinct legal entities in distinct jurisdictions, refreshed on a rolling annual basis.

An auditor verifying a corridor's governance history fetches the transparency log entries, checks inclusion proofs and witness signatures, and verifies that every governance event in the corridor's local log has a corresponding transparency-log entry. Missing entries indicate either log misconfiguration (investigate operationally) or governance tampering (escalate to maintainer-quorum review and consider corridor fork). The auditor procedure is mechanical and runnable by any third party; the audit tooling ships alongside the Vela CLI.

Transparency-log centralization is itself a threat surface (see §12.10). The witness-diversity requirement is the protocol's defense — no single log operator can rewrite history without colluding with at least one independent witness — but it is not absolute. A determined adversary with resources to compromise both the log and a sufficient witness set defeats this defense. The fallback is corridor forkability: a corridor whose transparency-log infrastructure is compromised can be forked, and the fork can declare a new witness set without losing the per-frontier event history.

7.1.2 7.1.2 Witness-Set Governance

The witness set is itself a governed object whose membership rotates over time. The protocol commits to the following governance rules.

Admission. A new witness operator is admitted by a `witness.admitted` governance event signed by the corridor's maintainer quorum (per the standard threshold) and cross-posted to the existing witness set. The admission event carries the candidate's identifier, public key, jurisdiction, operating-organization details, and a `witness_set_version` increment. Existing witnesses must counter-sign within the freshness window for the admission to take effect.

Removal. A witness is removed by a `witness.removed` event under the same governance procedure. Removal does not retroactively invalidate signatures the witness produced during

its tenure; it stops the witness from being counted toward the quorum-witness requirement going forward. A witness that goes offline for longer than 30 days without explicit cause is automatically marked `witness.stale` and excluded from quorum counts until it returns or is removed.

Minimum-quorum-maintained invariant. At every moment in a corridor's history, the active witness set must contain at least two operationally-independent witnesses meeting the §7.1.1 diversity criteria. If admission or removal would temporarily drop the count below two, the operation pauses until a replacement is admitted. A corridor that cannot maintain this invariant pauses all governance-event acceptance until it is restored.

Witness-set version pinning under replay. Each governance event carries a `witness_set_version` field; the reducer accepts an event only if it carries witness cosignatures from the witness set active at the version named in the event. Historical events under prior witness sets remain valid forever — replay does not require the historical witness set to still be operating, only that the historical cosignatures were valid under the witness set active at the time. The `witness_set_version` field is therefore a per-event-epoch pin, not a moving target.

Witness-operator obligations. Each registered witness commits operationally to: (i) verifiably mirror the transparency log; (ii) produce cosignatures within the corridor's freshness window for valid governance events; (iii) refuse to cosign events that do not verify against the corridor's stated governance policy; (iv) publish the witness's own operational status, software version, and hardware-root-of-trust details on a regular schedule. Failure to meet any of these obligations is grounds for `witness.removed` proceedings under maintainer-quorum review.

7.2 Federation through Peer Hubs

Frontiers federate by declaring peer hubs in their manifest. A peer declaration carries:

- The peer's identifier
- The peer's HTTPS URL
- The peer's Ed25519 public key

Federation is currently point-to-point: each frontier knows its peers. The canonical Registry layer, which would hold the federated record of frontiers and signer identities globally, is reserved but not yet shipped. Until it ships, federation depends on each frontier's manifest being current and each peer being reachable.

Peers exchange events, signatures, and state hashes. A peer cannot rewrite another peer's history; it can only mirror, observe, and propose. Disagreements between peers about the state of a shared object are recorded as events (see §7.3).

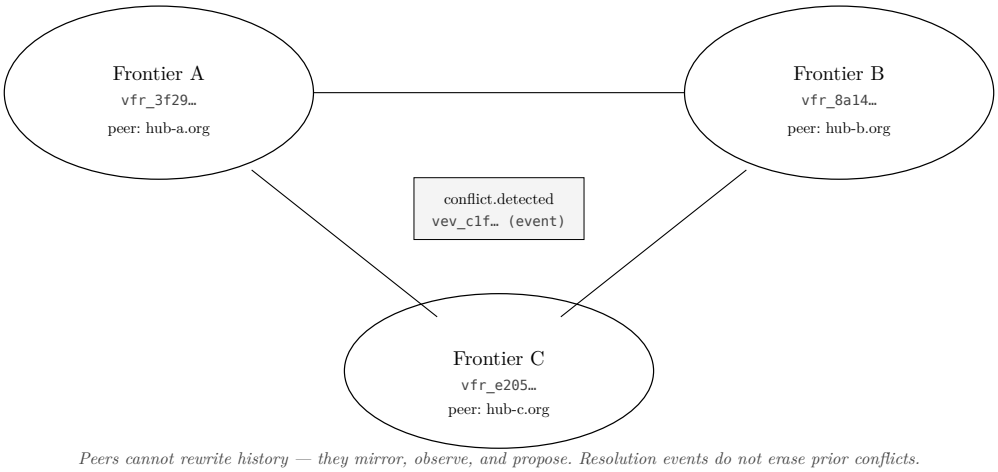

Figure 5: *Federation through peer hubs. Frontiers federate by declaration; peers mirror events and verify signatures; conflict between peers is recorded as state of its own, and resolution does not erase the original disagreement.*

7.3 Conflict Detection and Resolution

When two peers' views of a finding diverge, the protocol records a `frontier.conflict_detected` event. The conflict is not silently resolved; it is recorded as a state of its own.

Resolution is an explicit event: `frontier.conflict_resolved` carries the resolution decision, the affected findings, the reasoning, and the signing actors. The original conflict event remains in the log — the protocol does not erase the disagreement when the disagreement is resolved.

This is the protocol's commitment to maintaining disagreement as a recordable state rather than forcing premature consensus. Two replicated results that disagree under conditions X and Y are not noise to be averaged away; they are a structural feature of the field's current knowledge, and the protocol records them as such.

When the underlying disagreement cannot be resolved — when two scientifically defensible camps reach different conclusions — plural canonical views can coexist on the same finding. The protocol records the dispute itself as state, with each defensible position carrying its own scope, evidence packet, and signing actors, so a downstream reader inherits the structure of the disagreement rather than an averaged compromise.

7.4 Bridges and Cross-Frontier Links

A **bridge** (`vbr_*`) is a candidate cross-frontier link: two findings in different frontiers that share an entity, mechanism, or evidence base. Bridges are detected automatically (current implementation filters out too-generic biological terms to reduce noise) and confirmed or refuted by review.

A `bridge.reviewed` event records the reviewer verdict: `confirmed` (the bridge holds) or `refuted` (the bridge does not hold). The reducer does not mutate the underlying findings; the verdict projects onto the bridge's status on read.

Bridges are the seed of Atlas composition. An Atlas declares which frontiers it composes and which bridges between them have been confirmed. The Constellation layer, when built, will compose Atlases across domains using bridges between Atlases as the cross-domain primitive.

7.5 Forkability

A protocol is captured if it cannot be forked. Vela's forkability is structural: because identities are content-addressed and history is signed, a fork carries its full history and signer graph. The fork inherits everything except the canonical Registry pointer; the forked frontier declares its own peer set.

A fork is a sanction in Ostrom's sense — the ultimate sanction available when governance fails. Lesser sanctions exist as design goals (reviewer credential suspension, registry visibility reduction, frontier quarantine) and remain partly to be designed. The fork is the constitutional commitment: any frontier can be forked at any time, and the fork inherits the history rather than starting from blank state.

The fork path must be real before it is needed. Designing forkability as a v1 commitment, with running tooling, is what makes the threat credible during normal operation.

7.6 Kernel-Retraction Semantics

A released Carina kernel version published to the transparency log can subsequently be retracted by the steward — when a defect is identified, when a security vulnerability is found in the canonicalization or signature-verification rules, or when the kernel is determined to violate the protocol's governance constraints. The protocol commits to the following semantics.

A `kernel.retracted` event signed by the steward (with appropriate quorum) marks a kernel version as retracted and posts the retraction to the transparency-log witness set. The retraction is *prospective for new events*: a reducer encountering a fresh event under a retracted kernel rejects the event under strict mode. The retraction is *informational for historical events*: events already in the log under the retracted kernel remain in the log with their original signatures intact, but downstream consumers see a `kernel.retracted` annotation on the affected frontier's metadata indicating that the historical evidence should be re-evaluated against the corrected kernel before it is treated as load-bearing.

If a retraction is severe enough that historical events under the retracted kernel must be considered unsound — for example, a signature-verification rule defect that allowed forged signatures to verify — the corridor's maintainer quorum may file a `frontier.kernel_remediation` event proposing migration of the frontier's history to a corrected kernel version. The migration is itself a series of governance-reviewed events: each historical event is re-validated under the new kernel, signatures are re-verified, and any event that fails re-validation under the corrected kernel is marked as quarantined pending review. Historical events that pass re-validation retain their original timestamps and signers; quarantined events are inspectable but excluded from canonical state. This is the protocol's recourse against fundamental kernel defects discovered late, and it explicitly trades off some replay-determinism guarantees for the ability to correct a load-bearing flaw without abandoning a corridor's accumulated history.

8 Composition: Atlases and Constellations

The frontier-level invariants — per-frontier event log, replay determinism, content-addressed identity — are load-bearing. Composition layers read from frontier state without rewriting it. Replay determinism is per-frontier; append-only chains work because a target finding lives in exactly one frontier; federation moves state at frontier granularity. The substrate (Vela) guarantees per-frontier replay; composition reads from substrate state.

An **Atlas** (`vat_*`) is a living, versioned map composed of one or more frontiers with explicit bridges between them. It declares its composing frontiers (each by `vfr_*`, optionally pinned to a snapshot hash), confirmed and refuted bridges, maintainers, and review policy. An Atlas is read-only over its frontiers — the operator chooses which frontiers to include and which bridges to confirm or refute, but cannot rewrite history. Atlas-level proof packets concatenate the composing frontiers' proof packets plus the bridge attestations.

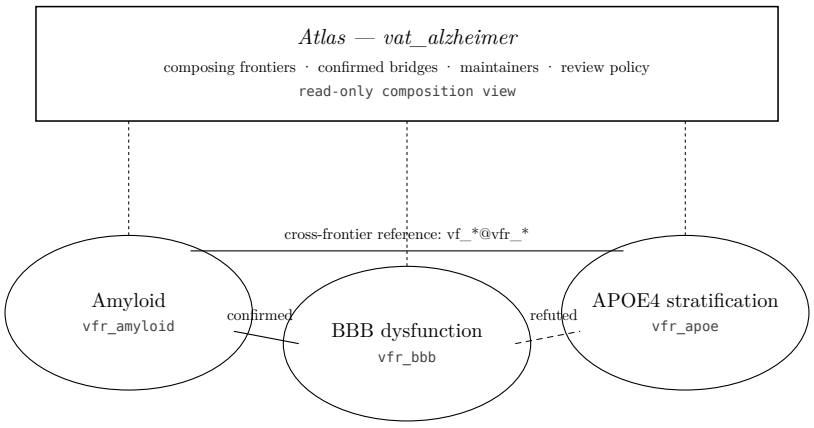

Figure 6: *Atlas composition. Bridges between frontiers are confirmed or refuted by review; the Atlas presents a composition view over read-only frontier state.*

A **Constellation** (`vco_*`) is a cross-domain network: a graph of Atlases spanning scientific domains. The primitive is reserved in Carina v0.5; no instance ships yet. The hard problem is bridge granularity at the cross-domain scale, and the engineering commitment is that this layer ships only when its primitives are stable.

9 The Adoption Path

The protocol is built to be deployed in stages. The first deployment is a single bounded frontier; the long-arc deployment is a federated commons across scientific domains. The path is not a forecast; it is the sequencing argument for how an open public substrate becomes the default before closed alternatives consolidate.

9.1 The First Corridor

The first proof is a single writable frontier. One question, one community, one correction that travels farther than it would have in the paper system. The pilot passes only if an

accepted correction changes a downstream funding, review, lab, or regulator-facing decision that otherwise would have repeated the old assumption.

The minimum viable configuration:

- One bounded scientific frontier (a specific disease question, a specific materials class, a specific surveillance corridor)
- A patient-led foundation or focused research organization as host
- Three to five participating labs
- A weekly frontier export (what changed, what should stop, what should be tested, which assumptions are now too fragile to buy)
- Signed negative-result and correction deposits as a milestone-funding condition
- One regulator-readable proof packet produced from the corridor

The first corridor is engineered to close the loop once. Closing the loop means: a lab deposits a result, a reviewer signs the proposed change, the canonical event records the state transition, the downstream consumer (foundation, reviewer, regulator) inspects the change, and the next decision is made differently because of it.

The first corridor is not a generic platform pilot. It is a corridor with a name, a question, named participating institutions, and a named host. The point of the pilot is not to demonstrate the protocol's elegance; it is to demonstrate that signed state movement changes a real decision. The corridor's named candidates, budget envelope, regulator counterpart, reviewer credentialing committee composition, conditional-deposit grant clause, gating commitments, sandbox-demo commitment, IRB/IACUC procedures, and pivot criterion are specified in the companion document *The First Corridor Pilot Plan v0.1*. The cross-implementation conformance suite that the first corridor will run against is described in *The Vela Protocol Specification* §9. As of v0.1, the host, lab, and regulator counterparties for the first corridor are in active conversation but not yet under signed letters of intent; the companion plan names the candidates and the gating commitments explicitly so funders can read the proposal against real institutional categories rather than abstractions.

9.2 Capital and Incentive Structure

The protocol's adoption depends on the incentive stack matching the work it asks of participants.

Labs deposit failed protocols and negative results because milestone capital is conditional on signed deposits. Reviewers sign transitions because the host foundation underwrites reviewer labor per signed event. Foundations adopt because they get a weekly frontier export they could not produce alone — a view of which assumptions are now too fragile to fund, which experiments are most discriminating, which corrections should pause active programs. Regulators engage because the protocol produces inspectable provenance suitable for IND, CMC, real-world-evidence, DSMB, IRB, or IACUC submissions, without requiring agency adoption of the protocol itself. Hospitals join because the corridor produces a recommendation surface their clinicians can read against the current frontier rather than the last textbook.

The substrate does not have to win every actor's incentive analysis at once. It has to win the first coalition's. The first coalition is: one patient-led foundation or FRO host, three to five labs willing to deposit failures, a small reviewer pool credentialed for the corridor's safety class, and

one downstream consumer (funder, reviewer, or regulator-facing team) whose decisions inspect the state.

The protocol recognizes several actor capacities — labs, reviewers, maintainers, hosts, agent operators, regulators, translators, funders — each with a corresponding actor record and a corresponding contribution path. Per-role participation paths are specified directly in §13. These are protocol-recognized capacities, not job titles; the same institution typically holds several. The protocol does not enforce separation between roles; it makes the actions under each one inspectable.

9.3 Sequencing

The deployment runs in phases. The boundaries are approximate; the order is load-bearing. Each phase has a load-bearing milestone, a capital source matched to the work, and a failure mode that decides whether the next phase becomes possible.

The **first corridor** runs over years one and two. The substrate goes live in one bounded frontier with three to five labs, one host, signed deposits, a weekly export, and one regulator-readable proof packet. Capital comes from a patient-led foundation or FRO seed grant. The phase fails if no corridor changes a real decision — the substrate stays theoretical.

Adjacency runs years two through four. Second and third corridors adopt the same protocol because rebuilding the infrastructure costs more than joining. Reviewer credentialing professionalizes, foundation pools grow, and regulators begin accepting state histories as inspectable support for submissions. Capital comes from foundation pools, early federal grants, and FRO follow-ons. The candidate second corridor under active discussion is **pediatric high-grade glioma perturbation-response state**, hosted by a coalition between The Pediatric Brain Tumor Foundation, Alex's Lemonade Stand, and the Children's Brain Tumor Network — chosen because it has clean perturbation-response datasets at scale, an established and tight clinical-trial community, and a high-information-density failure record (multiple recent immunotherapy and targeted-agent trials with negative readouts). The pediatric-cancer foundation ecosystem is also better positioned than the AD foundation ecosystem to adopt the conditional-deposit grant clause (see the companion *First Corridor Pilot Plan*) in a meaningful share of its grant portfolio. The phase fails if the protocol stays niche; parallel closed substrates emerge to fill the gap.

Engine maturity runs years three through six. Task contracts ship, agent-attestation tiers stabilize, a public scientific compute pool forms, and an ARPA-H-style Frontier Infrastructure BAA writes the first envelope combining translational program management, BARDA-shaped surge capacity, and FRO-built bottleneck teams. The phase fails if closed-lab agent stacks define the category before open alternatives become credible.

Body at scale runs years five through ten. The first scientific execution facility at industrial scale comes online: federated synthesis halls running autonomous protocol execution against shared frontier state, accredited review surfaces, GMP-grade manufacturing handoff for downstream translation, and writeback into the public substrate, all under one operational roof at a physical footprint comparable to a battery factory or a semiconductor fab. This is what *gigafactory-class scientific execution* names — the term is meant literally, in the same sense as Tesla Gigafactory Nevada or TSMC's advanced-node fabs, applied to scientific work rather than to cells or chips. The institutional argument for the category is made in *The Terafactory Age*;

here the relevant property is scale matched to the agent-rate envelope from §1 and §11.1. An open compiler exists and integrates with the federated identity layer. The body clause stacks across foundation capital, federal grants, regulator-readable inspection, and hospital enrollment compacts. Capital comes from real-asset infrastructure funds (the asset class that finances battery factories and semiconductor fabs), sovereign mandates, and federal cost-share. The phase fails if closed bodies become default infrastructure and the open public version negotiates from a weaker position.

The **second decade** runs years ten through twenty. Substrate-native credit is supplementary to the existing citation economy for at least three full tenure cycles — fifteen years at a minimum, longer in fields with slower committee turnover. Search committees and study sections begin reading substrate contribution records alongside publication lists, not in place of them. Lab capacity standardizes around shared infrastructure. Translation professionalizes at the state-to-decision boundary. None of this happens if the protocol layer has not already survived a few real adversarial events. The phase fails quietly: substrate-native credit stalls and the protocol survives without compounding into the next paradigm.

The failure modes across phases are institutional, not technical. The protocol layer's job is to make each move possible without making any of them mandatory. A phase that does not arrive does not break the protocol; it leaves the next phase harder to start.

9.4 The Second Decade

The first decade builds the wedge. The second decade is when the cultural and institutional transitions happen — and these are the hard ones.

The credit transition is gated by the slowest institution. A scientist whose work moves canonical state but produces no high-impact papers should be promotable; until tenure committees and study sections accept substrate-native credit, the system runs on dual rails (legacy citation credit and protocol-native contribution credit). The wedge has to be designed so that legacy credit and protocol-native credit are legible in the same packet for at least a generation.

The funding transition is gated by capital structure. Most current research funding follows narrative grant proposals on annual or multi-year cycles. Substrate-native funding pays partly on signed state movement: milestone payouts on deposits, retrospective bounties on closed branches, foundation-set rewards for high-leverage corrections. Both regimes coexist. The principle: payment follows the unit of work the system needs more of.

The institutional transition is gated by the speed at which new institutional forms can form. FROs, patient-led foundations with milestone-funding authority, federated reviewer pools, translation studios, and infrastructure-class capital vehicles are emerging but not yet at scale. The body essay's gigafactory and terafactory institutions are second-decade categories: they exist after the substrate has cleared its first adversarial events and the political case for public scientific infrastructure has been made by visible benefit.

The geopolitical question becomes load-bearing in the second decade. The protocol's forkability and federated identity partly handle plural jurisdictions, but clinical and pathogen data crossing borders raise legal and political questions that no technical design resolves on its own. The jurisdictional compact described in the Terafactory essay — sample-sovereignty terms, data-localization boundaries, export-control review, dual-use committee jurisdiction, benefit-sharing

conditions, regulator-readable evidence packets — is a precondition for cross-border state movement, not a downstream consequence.

10 Implementation Status

The protocol is implemented. The implementation is partial.

The documentation stack. This paper is the architecture. *The Vela Protocol Specification* (v0.5) is the bit-level technical companion — type primitives, canonical event shape, reducer mutation kinds, canonicalization rules, and conformance criteria. *The First Corridor Pilot Plan* (v0.1) is the operational companion — named candidates, budget envelope, regulator counterpart, gating commitments. The Vela repository's `docs/` directory carries the operational specification (`PROTOCOL.md`, `CARINA.md`, `REGISTRY_GOVERNANCE.md`, `STATE_TRANSITION_SPEC.md`, `MISSION_ATLAS.md`). Formal proofs live in `lean/Vela/`. Implementers work primarily against the specification; funders and hosts work primarily against the pilot plan; this paper carries the architectural argument that frames the others.

10.1 What Ships

The Vela protocol crates ship as a Rust workspace under dual Apache-2.0 / MIT licensing. The relevant crates:

- `vela-protocol`: core protocol — finding bundles, events, proposals, reducers, proof, signing
- `vela-protocol-core`: minimal-dependency core types for WASM and embedded targets
- `vela-protocol-wasm`: WebAssembly bindings
- `vela-cli`: user-facing CLI (`vela` binary), published to crates.io
- `vela-atlas`: Atlas composition layer
- `vela-relay`: adapter layer for translating external activity into protocol proposals
- `vela-search`: build-time index over registered frontiers (derived view, not authority)

The Carina kernel ships at v0.2 with selected v0.4–v0.5 extensions for Atlas and Constellation primitives. Kernel digests are pinned per-frontier; integration tests enforce consistency between the JSON schema and the documentation in `docs/CARINA.md`.

Working protocol features at v0.338:

- Finding bundles with content-addressed identifiers
- Append-only event logs with full reducer dispatch over 26 mutation kinds
- Deterministic replay (Rust reducer reference-grade; Python and TypeScript partial)
- Ed25519 signing with actor records and revocation
- Multi-signature governance policies for owner rotation, with Lean-proved soundness (Theorem 16)
- Federation through declared peer hubs
- Conflict detection and resolution events
- Bridge detection and confirmation between frontiers
- Atlas composition (read-only) over multiple frontiers
- Negative results, trajectories, replications, predictions, resolutions, datasets, code artifacts, diff packs, conjectures, proof packets

10.2 What Is Reserved

The following components have reserved namespaces and Carina primitive definitions but no shipping implementation:

- **Constellation layer (`vco_*`)**: cross-domain network, Carina v0.5 primitive defined
- **Navigator**: workbench UI; namespace reserved
- **Registry**: canonical federation record; current federation is point-to-point via peer hubs
- **Commons**: governance stewardship; current governance is per-frontier `RegistryGovernancePolicy`
- **Constellate Studio** and **Constellate Commons**: namespace reserved

The decision to reserve namespaces and primitives without shipping implementations is deliberate. The primitives need to be stable in Carina before implementations build against them; shipping implementations before the primitives stabilize would lock in choices that later need migration. The architecture is honest about which layers ship and which are scaffolded.

10.3 What Is Partial

Several components ship in partial form:

- **Cross-implementation reducer parity**: Rust is reference-grade. Python hydrates most kinds but does not yet cover v0.55+ trajectory and evidence-atom materializers in full. TypeScript covers a subset for fixture validation.
- **Bridge detection**: ships, but with conservative filters that exclude too-generic biological terms. The trade-off: false-positive bridges are suppressed; some real cross-frontier links (e.g., shared blood-brain-barrier physiology across anti-amyloid and brain-tumor frontiers) are missed because the only shared entity tag is too generic. Improving this requires more granular entity tagging in the source frontiers.
- **Confidence model**: scalar bounded score in [0.0, 1.0]. Credal sets (interval-valued confidence with explicit lower and upper probability) are future work.
- **Federation**: peer-hub model works but requires manifest currency. The canonical Registry, which would hold the federated record of frontiers and signers, is not yet built.

10.4 What Has No External Steward

The Vela protocol is open-source under Apache-2.0 / MIT, publicly released at v0.48.0 on 2026-05-02. No external steward is yet named. Naming one is a precondition for the protocol becoming infrastructure rather than a project; the stewardship structure is part of the Commons layer that has not yet been built.

Within 90 days of v0.1's release, the project will either name an external steward (a non-profit, fiscally sponsored entity, or chartered consortium) or stand up fiscal sponsorship through an established host such as Code for Science & Society, Open Collective Foundation, or NumFOCUS. The steward will hold the Vela trademark and crates.io publication rights, govern the Carina kernel evolution process, and own dispute resolution for governance-level decisions. Until the steward is named, the project declines consortium-scale commitments and limits engagement to pilot-grade collaborations. Funders, regulators, and institutional partners should treat the absence of a named steward as a deliberate gate, not an oversight.

The long-term steward sustainability question — how the steward stays funded after the initial pilot — is answered through a layered model: foundation membership dues from participating hosts (analogous to Crossref's member-publisher model), federal cost-share on infrastructure-class scientific infrastructure (NIH Common Fund, NSF Mid-Scale RI-2, or ARPA-H-style program envelopes), kernel-services-as-public-utility fees from corridors using premium services (priority review queue scheduling, custom witness operation), and a long-term endowment goal building from foundation contributions toward an operating-cost reserve of three to five years. The xz, Log4j, OpenSSL, and NumPy maintainer-burnout literature is the cautionary corpus the steward will be designed against: no single individual carries the protocol's continuity, no single funder carries the steward's payroll, and the steward's operating budget is published quarterly so the community can monitor sustainability and intervene through governance if it drifts toward fragility.

Honest framing: the protocol is mature enough to deploy in a first corridor; the institutional structure around it is not yet mature enough to support broad adoption. The work of the next 12–24 months is forming the institutional structure (steward, governance consortium, charter) at the pace adoption requires.

11 Open Problems

Three known unresolved problems, named rather than deferred.

11.1 Scaling Reviewer Authority at Agent Rates

Agent proposal rates can exceed human review rates by orders of magnitude. At hundreds to low thousands of signed events per month, the current architecture — scarce merge authority, credentialed reviewer pools paid per event, deduplication and rule-based rejection for low-stakes transitions — works. At millions per month, the regime mature agent infrastructure will produce, the model changes. Agent-attestation has to become canonical for low-stakes transitions, with humans concentrated on contested merges, safety-relevant updates, and high-dependency hubs. The protocol's structural answer is the agent-attestation tier (§6.7) with operator accountability, content-addressed stack manifest, sponsor liability, and reputation-denominated stake. The economics and safety properties of agent-canonical merges at scale have not yet been demonstrated.

11.2 Paradigm Shifts and the Tacit Boundary

The protocol assumes the unit of state — the finding, the evidence atom, the condition — is stable. Across normal science, this holds. Across paradigm shifts, what counts as a finding under one paradigm is not what counts under another. Carina versioning, kernel pinning, and migration events handle incremental schema evolution but not the case where a new paradigm requires a fundamentally different schema with the migration itself contested. The design goal is migration legibility (recorded as events, reviewed under governance, forkable if disputed) rather than prediction of the next paradigm's schema.

A related limit is Polanyi's observation that "we can know more than we can tell." Tacit knowledge — in hands, in calibrated instruments, in trained intuition — sets a coverage limit on what the protocol structures into events. The architectural response is explicit infrastructure of a different kind: apprenticeship rotations, mentorship contracts, transfer events recording

who trained whom at what site. The protocol does not carry tacit content; it makes the structure of tacit transmission inspectable. How much of what currently appears tacit is actually structurable as condition records, trajectory steps, and calibration logs is an empirical boundary that will move as the protocol matures.

11.3 Capture, Credit, and Cross-Jurisdictional State

The protocol can remain open while the institutions around it consolidate. Git remained open; GitHub did not. The scientific analogue is the protocol staying open while orchestration tooling, calibration registries, reviewer credentialing, lab capacity allocation, or signer recognition consolidate into a closed stack. The architectural defense is the separations principle — each layer ships as a separable protocol or institution, with multiple implementations, forkability, and identity sovereignty — and the legal architecture (charter commitments, antitrust posture, licensing terms) is part of the design rather than a downstream consequence. The capture vector cannot be closed once and for all; defense requires continuous institutional work.

The substrate's reputation system must ship in v1, not as a v2 retrofit. Participants who would most benefit from substrate-native credit (early-career researchers, contributors at non-elite institutions, agent operators) will be the slowest to adopt if the protocol records signed contributions but does not surface them as inspectable reputation. The intended reputation signal is not a raw proposal-to-acceptance ratio (gameable through easy-proposal capability hacking) but a composite: Brier score on resolved predictions, novelty-adjusted acceptance rate, and downstream-citation weight. All three are computed from canonical state and independently re-verifiable. Building the composite surface is the work of the next 12–24 months.

Clinical data, patient identifiers, pathogen sequences, and dual-use biological information do not move freely across jurisdictions. The access-tier model handles part of this; legal and political constraints on cross-border data flow exceed what any technical mechanism resolves. The architectural commitment is that the transition's existence and provenance can be public even when underlying evidence is tiered to specific jurisdictions. The deeper jurisdictional compact — shared definitions of what travels, sample sovereignty, dual-use review, benefit-sharing — is still ahead.

11.4 The Knowledge-Insufficiency Boundary

The protocol coordinates the absorption of validated findings into shared scientific state. It does not generate scientific knowledge that does not already exist in some form in the corpus the substrate absorbs from. Working scientists, including Lowe (2025) in the AI-drug-discovery context, have correctly argued that machine learning rearranges existing evidence through pattern-matching and does not fill in the parts of the puzzle the field has not yet reached through reality-contact. The protocol operates inside that boundary, not outside it.

This matters operationally. A frontier model of Alzheimer's disease assembled by absorbing the public corpus cannot answer questions the public corpus does not contain. Pharma-archived failed-program data, patient-level cohort records, and trial-internal subgroup analyses sit outside the public corpus today; the IFP "Lost Archive" proposal (2025) is one institutional mechanism to surface a fraction of that data into accessible form. The protocol's access-tier model can carry such material under controlled review once it becomes available. New experimental evidence still has to be generated by labs, trials, and observations at the pace biology and ethics allow.

The Lost Archive sits inside IFP's broader 2025 *Launch Sequence* collection, a coordinated set of thirteen vertical AI-for-science infrastructure proposals (Replication Engine, ABC, TELOS, X-Labs, Lost Archive, Scaling Pathogen Detection with Metagenomics, and adjacent verticals) covering replication, evaluation, lab automation, dark-data rescue, biosurveillance, and federal data infrastructure. The collection's collective ask sits in the range of roughly five to ten billion dollars in proposed federal and philanthropic spending. Each piece names an institutional shape and a funding envelope; none names the substrate primitives the verticals would need to compose. The protocol triangulates against that collection: where each Launch Sequence proposal solves a vertical problem, Constellate is the horizontal coordination layer beneath them. Several of the proposals (Replication Engine, Lost Archive, Pathogen Detection) are difficult to operate without a shared scientific-state primitive of the kind this document specifies, and the absorption-gap thesis the collection collectively triangulates on is the same thesis this protocol's §1 names.

The protocol's honest claim is bounded. It does not promise to end disease. It promises to make the cumulative validated work that does end diseases, when it happens, durable, signed, replayable, and propagated. The cardiovascular-disease mortality decline of 1950 to today (a roughly 75% reduction in age-standardized US deaths) shows that many compounding layered interventions across drugs, devices, surgery, diagnostics, emergency response, and lifestyle absorbed over seventy years can produce civilizational results. The protocol's contribution is to make that absorption layer exist as infrastructure rather than as decades of cultural practice, so the next equivalent declines can compress. The boundary at which the protocol stops being useful is the boundary at which there is no new validated reality-contact to absorb. The protocol does not push that boundary outward. Experiments, trials, and observations do.

11.5 Evaluation and Benchmarking

The protocol records reviewed scientific-state transitions. It does not currently ship a public evaluation regime for the quality of those transitions, the calibration of the reviewers who sign them, or the downstream predictive value of the resulting frontier state. Three evaluation surfaces are implicit in the architecture and unresolved as deliverables. The first is reviewer calibration: a Brier score on resolved predictions per credentialed reviewer, computed from `vpred_*` resolution records, is structurally available in canonical state but is not surfaced as a published metric in v0.1. The second is frontier predictive value: the proportion of canonical findings that survive future reality-contact, measured against the contradiction and retraction record, is computable from the DAG but not packaged as a benchmark. The third is agent-output quality: the acceptance, contradiction, and downstream-citation distributions for agent-attested deposits are recorded but not published as an agent-reliability ranking.

IFP's TELOS proposal (2025) is the institutional analogue for what FrontierBench-shape work would look like at scale: an evaluation institution that scores AI-for-science systems on operationally meaningful tasks rather than benchmark proxies. TELOS proposes the institution and the funding shape; it does not specify the primitive types the evaluations would write into, the signing rules for evaluation results, or the replayability requirements for the evaluation artifacts. The substrate provides those primitives. A future FrontierBench specification (out of scope for v0.1) would define `vbench_*` and `veval_*` types, the canonical hash discipline for benchmark task definitions, the signature thresholds for benchmark-result deposits, and the cross-frontier reference grammar that lets evaluation results inspect the canonical state they evaluate against. Whether Constellate ships that specification or hosts an external one is open.

What the architecture commits to is that the primitives are present and a future benchmark layer is implementable on top of them rather than requiring a parallel substrate.

12 Threat Model

The protocol carries scientific state across institutions and over time. The state's value to the field is also its value to actors who would prefer to bend it. This section names the adversaries the architecture expects, the surfaces they attack, the defenses the protocol provides, and the residual risks no protocol design eliminates on its own. The threat model is a companion to §11 (Open Problems): open problems are unresolved questions, where threats are expected adversaries with known surfaces. Two principles run across the section. First, the protocol prefers detection and revocation over prevention: an attack that succeeds locally should leave a trail. Second, the protocol prefers structural defense over policy defense: a separation that cannot be undone by any single actor's decision is stronger than a rule that depends on that actor's good faith.

12.1 Identity-Based Attacks

Where reputation accrues, attackers attempt to forge it. The first identity-based attack is the **sybil**: forging multiple identities to inflate apparent reputation. The protocol's first defense is identity anchoring — actor records are tied to Ed25519 keys, optionally to ORCID identifiers and institutional affiliations. Reputation accrues against an actor's signed history, which is expensive to fabricate at scale. Signature thresholds and credentialed reviewer pools concentrate canonical-merge authority on actors whose credentials trace back to institutional credentialing. Low-volume pseudonymous sybils remain feasible at small scale; what they cannot do is forge a years-long signing history under multiple identities and have all of them admitted into credentialed pools. State-actor-scale sybils with significant resources to accumulate fake event histories are not stopped by any protocol mechanism alone; defense against that adversary requires governance-layer vigilance — credentialing audits, suspicious-pattern detection, and revocation when abuse is identified.

The second identity-based attack is the **compromised signer**: an Ed25519 key stolen by insiders, lost through endpoint compromise, or coerced. The protocol records revocation as an event: an actor's record is amended with a `revoked_at` timestamp and a reason. Signatures produced before the revocation timestamp remain valid for the historical record; new signatures from the revoked key are rejected. Past acceptances under a compromised key are not retroactively voided — they remain inspectable, and downstream consumers can decide whether to re-review claims that depend on the compromised actor. Signature thresholds on high-stakes findings ensure that no single compromised key can move canonical state alone. Cross-frontier signature replay is closed structurally by the frontier-ID binding specified in §6.2.

12.2 Orchestration Capture

The protocol can remain open while the institutions around it consolidate. The same pattern played out in software: Git remained open while the collaboration tooling above it (issues, pull requests, Actions, review history) consolidated into one company's platform. The scientific analogue is the protocol staying open while orchestration tooling, calibration registries, reviewer credentialing, lab capacity allocation, or signer recognition consolidate into a closed stack. The architectural defense is the separations principle: each layer of the stack ships as a separable

protocol or institution, with multiple implementations, forkability, and identity sovereignty. The legal architecture — charter commitments, antitrust posture, licensing terms — is part of the design rather than a downstream consequence. Capture pressure is continuous, and the defense requires continuous institutional work.

12.3 Malicious Agent Floods

The substrate is designed to receive proposals from humans, labs, and agents. Agent proposal rates can exceed human review rates by orders of magnitude, and a flood of low-quality or strategically-biased proposals can drown the reviewer queue. The protocol's defenses are scarce merge authority, deduplication, rule-based rejection of malformed deposits, and the agent-attestation tier (§6.7) with operator accountability, content-addressed stack manifests, sponsor liability, and reputation-denominated submission stakes. At the volumes mature agent infrastructure will produce, this architecture works only if reviewer-pool capacity scales with grant funding and if low-stakes transitions accept agent-canonical merge under sampling and audit. The full economics have not been demonstrated; a sophisticated adversary that cultivates sponsor relationships and accumulates a clean first-100 record before flooding remains the residual risk this defense addresses imperfectly.

12.4 Schema and Consensus Threats

Schema poisoning. Carina evolves. New event kinds, new evidence types, new field semantics enter through kernel version bumps. An attacker who can influence the schema-evolution process can embed primitives that admit transitions the field would reject if visible. The defense is schema evolution under multi-party governance: kernel digests are pinned per frontier, migration events are explicit and inspectable through the transparency-log witness (§7.1.1), and implementations are obliged to reject events that do not match the pinned version. A coordinated attack against the governance quorum that decides schema updates remains possible; that risk is the same risk as governance capture more broadly.

Premature consensus and dependency forgery. Two failure modes sit on opposite sides of the same line. Premature consensus forces a single canonical view where legitimate disagreement should be recorded; dependency forgery falsely attaches a finding to a high-confidence anchor it does not deserve. The protocol's defense against the first is the explicit treatment of disagreement as a recordable state (§7.3); plural canonical views can coexist on contested findings, and the dispute itself is part of the record. The defense against the second is pinned snapshot hashes on cross-frontier references (§4.5, mandatory under strict mode) and signed dependency edges; consumers must verify the chain. Detection lags adoption, and a sufficiently coordinated reviewer group can still narrow what gets proposed — the protocol records, but does not police, the culture around it.

12.5 Trade-Secret Leakage Through State

Proprietary search topology — the routes a closed lab tried and abandoned — is sometimes worth more than the eventual finding. A naive deposit regime can leak that topology by aggregation: enough public state transitions can reconstruct the rejected branches, even when each individual deposit looks innocuous. Conditions records (species, assay, comparator, endpoint, sample size) are exactly the fields that fingerprint a lab's pipeline; an adversarial

competitor reading the public condition record on twelve negative deposits over eight months can reconstruct a lab's screen topology far better than the lab would like.

The protocol's defense layers four mechanisms. First, the access-tier model (§6.6): public transitions carry hashes, signers, and dependency movement, while the full evidence content can live behind Restricted (trusted-reviewer escrow) or Classified (regulator escrow). Second, a **corridor-maturity-dependent k-anonymity floor on condition records** in the Public tier: condition records that uniquely identify a lab's pipeline must be aggregated to at least k *non-colluding* depositors before they appear in the Public surface, where non-colluding is verified by requiring the depositors to span distinct institutional affiliations and at least one to be outside the corridor's named host's funding portfolio. The floor scales with corridor maturity: k=2 during the first 12 months of a corridor's operation (when participating-lab counts are still small and any larger floor would force all deposits into Restricted tier), k=5 once the corridor has at least eight distinct active depositors, and k=10 for corridors handling competitive-research deposits in domains where adversarial inference is high-value (drug discovery, materials, agriculture). Below the active floor, records hold in Restricted tier with release conditions defined at deposit time. The k-1 collusion attack remains a real concern at any floor; the non-colluding-depositor requirement raises its cost but does not eliminate it, and a corridor whose adversary has placed a depositor inside its credentialed pool has problems the privacy-floor mechanism cannot solve. Third, **deposit batching and timing jitter**: a lab can elect to deposit weeks of trajectory updates in a single signed batch with timestamps coarsened to the week, so adversarial inference about the lab's experimental cadence is blunted. Fourth, a **per-depositor aggregation-leak budget**: each depositor's cumulative public condition-record surface across a rolling twelve-month window has a budget defined by their host's corridor governance; exceeding the budget triggers either coarsening to Restricted tier or an explicit policy review.

These defenses raise the cost of aggregation deanonymization without eliminating it. A determined adversary with months of patient observation across many deposits can still infer significant pipeline structure, and the protocol explicitly acknowledges this. The mitigation is not perfect privacy; it is to make the leak rate slow enough that the strategic value to the depositor (faster downstream decisions, calibration records, reviewer attention) outweighs the leak.

12.6 Selective Non-Deposition

A closed lab can publish what flatters its pipeline and withhold what does not. The protocol cannot compel deposit from actors outside its coalition. The first defense is incentive design: milestone capital, grant conditions, hospital enrollment compacts, and regulator-readable inspection stack until non-deposit is too expensive for serious public-facing actors. Foundation capital and federal grants are the most effective levers; non-grantee labs are not bound by these conditions, and coalition coverage of any given field is partial.

The second defense converts absence into recordable state. A `finding.non_deposit_suspected` event class lets a credentialed actor (typically a corridor maintainer or foundation program officer) sign a suspicion event when external evidence — a regulatory filing referencing undeposited data, a conference abstract describing unreported negative trials, a patent filing whose enabling work has no deposited evidence, a news report of a discontinued program — suggests a closed actor has produced findings that should be in the public record under

existing grant conditions or community norms. The event carries the suspected actor's identifier, external-evidence reference, rationale, and corridor's policy basis. It does not assert that a violation occurred; it asserts that the suspicion has been recorded. A pattern of such events around a particular actor accumulates into a public signal that funders, regulators, and peer institutions can read — converting "absence is silent" into "absence is structured state."

The defense against false-positive suspicion is the signer's own reputation graph: malicious or sloppy suspicion events accrue against the signer's calibration record, and the suspected actor may file a `non_deposit_contested` event with countervailing evidence. The defense against the *harassment surface* (a credentialed but malicious reviewer issuing suspicion events against rival labs at scale) is a structural rate limit: the protocol's default policy caps `finding.non_deposit_suspected` events at two per signer per suspected actor per rolling 90-day window. Exceeding the limit converts subsequent events into Restricted-tier deposits pending maintainer-quorum review.

12.7 Trust-Substrate Threats

The substrate is itself attackable at three distinct layers — the kernel pin, the federation peer set, and the cryptographic algorithms — each with its own defense.

Kernel-pin and manifest MITM. Each frontier pins a Carina kernel digest in its manifest and federation depends on every implementation reducing the event log against the same kernel. An adversary who controls the manifest server, a CDN, or DNS can serve a benign manifest to auditors while feeding a tampered kernel pin to one or more reducer instances; the reducers materialize divergent state, and the divergence is invisible until a peer-state-hash comparison surfaces it. The defense is out-of-band attestation through diverse transparency logs (§7.1.1): every released kernel version is published to at least two independent transparency logs (Sigstore Rekor v2 by default, plus at least one log in a distinct jurisdiction) with mutual witness cosigning, and kernel-pin changes are recorded as protocol events under corridor governance rather than as silent manifest updates. A reducer that observes a kernel-pin change without a corresponding signed event in the required transparency logs refuses to reduce against the new pin.

Peer-discovery eclipse. A frontier federates by declaring peer hubs in its manifest. An eclipse attack — partitioning a frontier from honest peers by controlling the manifest server, DNS, or the peer infrastructure itself — would let the adversary serve a divergent history that the partitioned frontier cannot detect. The defense layers three mechanisms: strict-mode frontiers must declare ≥3 independently-operated peer hubs (distinct legal entities, distinct hosting providers, distinct jurisdictions); peers exchange state hashes periodically and raise `frontier.divergence_detected` events on disagreement; the authoritative peer set is published to the same transparency log holding kernel pins. The deepest version of the attack — a state-actor adversary controlling multiple peers, the log, and the manifest simultaneously — remains undefended by any technical mechanism; the recourse is corridor forkability.

Transitive transparency-log trust. Cross-frontier references (§4.5) introduce transitive trust: a finding in frontier A that pins a snapshot of frontier B trusts B's witness set transitively. The structural defense is that pinned cross-frontier snapshots include the target frontier's `witness_set_version` at pin time, so a consumer can audit dependent witness sets independently. The corridor's launch documentation must include a written transitive-trust

analysis enumerating which referenced frontiers' witness sets the corridor's correctness depends on.

Long-range cryptographic compromise. Ed25519 is the current algorithm. Over the decades the protocol is built to outlast, that may not hold. Actor records carry an explicit `algorithm` field; the protocol can admit new algorithms as they mature, deprecating old ones through revocation timestamps. Historical signatures remain verifiable under the algorithm of their era; downstream consumers judge era-appropriate trust. Migration is a governance decision recorded through the transparency-log witness, not a silent rewrite.

12.8 What No Protocol Eliminates

Capture, premature consensus, and state-actor sybils require institutional defenses — credentialing, audit, charter commitments, regulator alliances — that exceed what any signing scheme provides. The protocol's job is to make these defenses possible, inspectable, and forkable. Whether they hold is a question of stewardship, not of code. The honest position is that §11 (Open Problems) and this section overlap on capture and sybils because they describe the same surface from two angles: what is unresolved, and what we expect to face.

13 How to Participate

A substrate is not a paper. It is a coordination object that exists only insofar as actors with real institutional weight choose to coordinate around it. The protocol the rest of this document specifies will be load-bearing if the right small set of people read the next paragraphs and do the work the paragraphs describe. The call is direct.

Reducer authors. The Rust reducer is reference-grade at v0.500. The Python reducer covers ~78% of mutation kinds; the TypeScript reducer covers ~41%. The 95% cross-implementation conformance threshold gates first-corridor deployment (see *The Vela Protocol Specification* §9). The path is mechanical: pick a missing kind from the specification's §6, study the Rust arm in `crates/vela-protocol/src/reducer.rs`, write the equivalent in the target language, validate against the test vector suite at `crates/vela-protocol/tests/vectors/`. Contributing instructions live at `github.com/vela-science/vela/CONTRIBUTING.md`. The fastest path to the protocol becoming canonical is independent reducer implementations agreeing on byte-identical state.

Corridor hosts. A corridor is the first piece of substrate any community needs. The pilot plan names candidate hosts for the first corridor (Cure Alzheimer's Fund, the Alzheimer's Drug Discovery Foundation, BrightFocus, the Chan Zuckerberg Neurodegeneration Initiative, or an FRO chartered for this purpose); the second is in active conversation with pediatric high-grade glioma foundations. If your foundation runs a research portfolio with three to five committed labs in a coherent frontier, you can host the third corridor. The host commits to convening the lab coalition, underwriting reviewer labor at the corridor's defined rate, adopting the conditional-deposit grant clause (Appendix F of the pilot plan), and owning corridor-level governance through the pilot period. Initial conversations: `corridors@constellate.science`.

Depositors and reviewers. If you run a lab, draft a clinical trial, or chair a study section in a frontier where a corridor exists, the protocol's value to you is reciprocal. You deposit signed findings and reviewer adjudications; you read the current frontier state when you make funding, trial-design, or clinical decisions. The lift is integrating one deposit step into existing

lab workflow. The corridor's host coordinates onboarding and reviewer credentialing. Express interest by writing to the corridor host or to `participants@constellate.science`.

Witnesses. §7.1.1 specifies a transparency-log witness regime under which independent witnesses cosign canonical state transitions for governance events. Witness candidates are universities, foundations, scientific societies, and standards bodies with established infrastructure operations. A witness cosigns governance-tier events from a published key, retains an append-only log of cosigned events, and participates in the witness-set governance process specified in §7.1.2. Witness onboarding documentation is at `docs/WITNESSES.md` in the Vela repository. Initial conversations: `witnesses@constellate.science`.

Stewards. The protocol commits to naming an external steward within 90 days of v0.1's release (§10.4). The steward holds the Vela trademark and crates.io publication rights, governs the Carina kernel evolution process, and owns dispute resolution for governance-level decisions. The candidate forms are a chartered non-profit, fiscal sponsorship through Code for Science & Society or NumFOCUS, or a member-governed consortium of the Crossref pattern. If you represent an institution willing to host the steward — a university, foundation, scientific society, or standards body — the project will work with you on the charter. Inquiries: `stewards@constellate.science`.

Researchers, journalists, policy writers. Read the trilogy. Read this paper. Tell your students, your readers, your funders. Cite the work when you write about cumulative science, the AI-science compounding question, peer review reform, or the institutional response to the absorption gap. The protocol does not need promotion; it needs to be understood by the people who decide what science's next infrastructure looks like.

The substrate becomes canonical because enough actors with public legitimacy coordinate to make withholding state visible, expensive, and institutionally awkward. That coalition does not assemble itself. It is assembled.

APPENDIX A References and Lineage

The architecture's intellectual lineage runs through several traditions. The load-bearing influences, not a complete bibliography.

A.0.1 A.1 The Constellate Trilogy

The case for a scientific state layer is argued in three essays:

- *Constellations of Borrowed Light* — the moral and epistemic case for a shared scientific record
- *The Discovery Engine* — the architecture of governed state transitions
- *The Terafactory Age* — the institutional and physical consequence at industrial scale

The trilogy is the argument for why. This document specifies what is built.

A.0.2 A.2 Institutional and Governance Lineage

Wu's *The Master Switch* supplies the Separations Principle and the structural argument against vertical consolidation in information industries. Ostrom's *Governing the Commons* gives the design principles for sustainable commons under collective governance — nested enterprises, graduated sanctions, monitoring rules. Lessig's *Code and Other Laws of Cyberspace* supplies the code-as-law frame and the legal architecture of protocol design. Nielsen's *Reinventing Discovery* anchors the collective-intelligence and designed-serendipity arguments. Bowker and Star's *Sorting Things Out* names the political consequences of classification systems the protocol's type kernel inherits. Strevens's *The Knowledge Machine* supplies the iron rule of evidence-only argument that grounds the canonical-merge norm. Galison's *Image and Logic* gives trading zones as the mechanism for cross-community coordination without unified ontology — the structural metaphor for cross-frontier links.

A.0.3 A.3 Scientific Epistemology

Polanyi's *The Tacit Dimension* marks the boundary between explicit and tacit knowledge that §11.4 acknowledges. Kuhn's *Structure of Scientific Revolutions* supplies the paradigm-shift question §11.2 addresses through schema evolution. Halpern's *Reasoning About Uncertainty* (MIT, 2003) and Walley's *Statistical Reasoning with Imprecise Probabilities* (Chapman & Hall, 1991) provide the formal frameworks for epistemic confidence and the credal-set future-work foundation referenced in §4.1. Groth, Gibson, Velterop (2010) and Kuhn et al. (2013, 2021) define nanopublications, the closest extant structured-assertion prior art (see §5.7). Bechhofer et al. (2013) and RO-Crate define Research Objects. W3C PROV-O / PROV-DM provides the provenance ontology FAIR infrastructure uses; integration with Vela's source–artifact–atom chain is discussed in §5.7. Casadevall and Fang's mBio editorials from 2009 onward, eLife's 2022 review-only model transition, and the work of PubPeer, Retraction Watch, F1000Research, ASAPbio, and Review Commons are the peer-review reform discourse §1 positions Constellate as a substrate complement to. Begley and Ellis (2012, *Nature*) is the preclinical reproducibility audit that motivates replication as a first-class object. Wilkinson et al. (2016, *Scientific Data*) is the FAIR principles paper. Bush (1945), *Science: The Endless Frontier*, is the institutional argument for federally funded basic research that informs the public-good framing throughout §1 and §9. Engelbart (1962), *Augmenting Human Intellect*, is the framing of tools as cognitive amplifiers.

A.0.4 A.4 Operational Precedents

The Protein Data Bank has sustained shared open-deposition infrastructure across competing institutions for over five decades. Crossref operates non-profit infrastructure across competing scholarly publishers under member governance. ClinicalTrials.gov coordinates trial registrations across sponsors and jurisdictions. Nextstrain and GISAID carry open phylogenetic and genomic deposition across competing national agencies. UK Biobank and All of Us sustain population-scale cohort infrastructure across institutions. The RECOVERY Trial demonstrates shared-protocol, low-overhead clinical-trial coordination at national scale. Polymath shows modular open-collaboration on research-frontier mathematics. Foldit and Galaxy Zoo demonstrate designed-serendipity collective intelligence at scale.

A.0.5 A.5 Protocol Precedents

TCP/IP and the IETF RFC process supply the open protocol governance pattern under standards-body consensus. Bitcoin (Nakamoto, 2008) is the canonical example of content-addressed, append-only ledgers with deterministic state. Ethereum (Buterin et al., 2014; Wood, 2014) demonstrates the pattern of protocol whitepaper plus yellow-paper formal spec — a structure Constellate inherits. IPFS supplies content-addressed distributed storage and federation. Git supplies content-addressed version control with forkability as a constitutional commitment. W3C PROV is the provenance ontology for scientific data. The Open Research Knowledge Graph supplies structured representation of scientific contributions.

A.0.6 A.6 The Vela Codebase

The Vela protocol, Carina kernel, Atlas composition, and supporting tooling ship as open source at `github.com/vela-science/vela` under dual Apache-2.0 / MIT licensing. Public release: v0.48.0, 2026-05-02. Current development version: v0.338. Protocol specifications, governance documents, and Lean proofs of the governance soundness theorems are in the repository's `docs/` and `lean/Vela/` directories.

APPENDIX B Why This Is Not the Previous Attempts

Open scientific infrastructure has a graveyard. The Open Science Framework launched in 2013 to host preregistrations, projects, and artifacts; the same period saw PLOS normalize open-access publishing, FORCE11 publish the FAIR principles and Scholarly Commons recommendations, ResearchEquals attempt modular publication units, the Open Research Knowledge Graph attempt structured assertion extraction, and Sci-Hub make access politically unavoidable. Each was a real intervention. None coordinates the layer Constellate names.

OSF built a durable artifact-coordination layer with preregistration, project structure, and persistent identifiers, but does not carry the state of the claims those artifacts contain. A preregistration is a frozen snapshot, not a live, signed, replayable transition. PLOS moved publication economics toward open access and made article-processing-charge funding plausible at scale, but did not change the unit — the paper remains the canonical object, and "what changed in the field" remains implicit in citation graphs. FORCE11 and FAIR named the principles (findable, accessible, interoperable, reusable) and moved policy at NIH, OSTP, and EU funders; the principles cover data, metadata, software, and workflows but explicitly do not cover the changes those objects make to a field's working knowledge. ResearchEquals built modular publication primitives — observations, hypotheses, methods, results, conclusions — as separately citable units; it carries finer-grained artifacts, not the reviewed transitions that turn

artifacts into claim-state updates. ORKG structures assertions from papers into a queryable graph and is the closest prior work to Constellate's evidence model; it runs into the limit Constellate's protocol is designed to address — a knowledge graph stores relationships but does not, by itself, decide what should change, who can propose it, who can merge it, or what action follows. Sci-Hub made the political point that access is contestable but did not propose a different layer.

The architectural commitment that distinguishes Constellate is the combination of three properties simultaneously: replay determinism, signed state transitions as the canonical object, and structural forkability. None of the predecessors made these three commitments simultaneously, and none ships running cross-implementation reducers as proof. The wager is that the layer the predecessors did not coordinate is the layer where compounding actually happens.

A second class of predecessor has emerged since 2020. AI and ML systems are building scientific capability directly. AlphaFold and AlphaFold 3 turned protein structure prediction into a queryable artifact, with the AlphaFold DB holding over 200 million predicted structures. Evo and Evo 2 scale a biological foundation model across genomes. ESM, Geneformer, and scGPT learn representations over proteins, cells, and single-cell expression. RFdiffusion designs proteins generatively. AlphaMissense classifies variant pathogenicity. GNoME proposed 2.2 million candidate crystal structures and identified roughly 380,000 stable ones. A-Lab ran 353 closed-loop synthesis experiments in 17 days. Coscientist, Google's AI co-scientist, and the FutureHouse agents act as scientific operators that search, plan, critique, and propose. Open Targets and PrimeKG aggregate target-disease evidence at scale.

Each of these systems is doing real scientific work. None of them is a state layer. AlphaFold is a prediction artifact: it does not absorb new validated evidence or carry forward the field's accepted scope on structure questions. Evo is a sequence foundation model: it generalizes across genomes without maintaining a position on what is currently believed about any particular mechanism. A-Lab closes a synthesis loop within a narrow materials regime; its decisions do not propagate to dependent claims outside the loop. Open Targets aggregates evidence under a scoring framework that supports prioritization but does not record reviewed change. The AI co-scientists generate hypotheses and proposals in session, but the proposals do not enter a record subsequent agents inherit.

The shape of the gap is the same as in the open-science predecessors. Each system optimizes a local scientific capability. The coordination layer where validated changes accumulate across systems, institutions, and time is unbuilt. Constellate is built to be that layer rather than to replace any of the systems above. AlphaFold's predictions, Evo's sequences, A-Lab's syntheses, and a co-scientist's proposals all enter the substrate as proposed transitions, get reviewed under their host frontier's governance, and become signed state if accepted. The systems compound through Constellate; they do not compete with it.

The recent pattern is sharper than "AI is doing science." The recent pattern is that scientific world models scale where validated data substrates already exist, and stall where they don't. AlphaFold scaled because the Protein Data Bank existed: fifty years of structural biologists depositing experimentally determined structures into one shared, machine-readable record. Evo scaled because GenBank, Ensembl, and the metagenomic atlases existed. GNoME scaled because the Materials Project and decades of computed-property databases existed. Open Targets aggregates evidence because ChEMBL, OMIM, and the disease-association literature

were structured first. The model is the visible artifact. The substrate is the prerequisite that made the model trainable.

The domains where comparable models have not appeared map onto the domains where comparable substrates do not exist. Disease progression, patient stratification, mechanism causality, trial-design assumptions, treatment-response heterogeneity, failed protocols, reviewed contradictions, negative results: each is data-rich in private form and substrate-poor in public form. There is no PDB-equivalent for what the field currently believes about Alzheimer's vascular biology in APOE4 carriers. The data exists. It sits in trial reports, supplementary tables, advisor memory, and the reviewer comments journals do not surface. A model trained on that distribution today inherits the distortions of the closed labs that hold the curated version.

The next decade of scientific world models will scale fastest in the domains where someone builds the substrate first. The protocol's job is to make that substrate cumulative and credit-bearing: a signed transition is the depositing primitive, the host frontier is the curated scope, the reviewer record is the validation layer, and the federation is the cross-institutional carrier. The AlphaFold-equivalent for disease progression, or for trial-design assumptions, or for mechanism causality, has the same prerequisite shape as the AlphaFold that was. The substrate has to exist before the model can train on it. The opportunity in 2026 is to build the substrates that the next generation of scientific world models will be trained against, before the closed labs build them privately and ship the models alone.

Two caveats keep this argument honest. First, scaling rates vary by domain. Protein structure scaled fast because verification was cheap and the data substrate had been built for half a century. Theorem-proving, materials simulation, and chemistry search will continue scaling fast for the same reason. Clinical medicine, ecology, social systems, and climate intervention will scale slower regardless of substrate quality, because reality-contact in those domains is slow, expensive, and ethically constrained. The protocol does not pretend Alzheimer's will scale like protein structure in the next decade. It commits only to making the substrate exist so that whatever pace the domain admits, the validated work compounds rather than scatters.

Second, the scientific world model for any non-trivial domain is plural. A serious Alzheimer's modeling stack will include a progression model, several patient-subtype models, mechanism-specific causal models, biomarker-trajectory models, treatment-response models, trial-simulation models, and model-system-transfer models. They specialize. They train against shared frontier state. They write back into it. The single-artifact framing that worked for next-token prediction does not transfer cleanly; the substrate-prerequisite argument does. Each specialized model needs the same thing the AlphaFold-equivalent for protein structure needed: a curated, machine-readable, validated substrate it can train against, query, and propose updates to. Constellate's protocol commitment is to that substrate, not to the models trained on it.

Three inputs determine the rate at which a scientific frontier advances: reality-contact (the cohorts, labs, simulators, and trials that produce new evidence), model capacity (the world models, simulators, and LLM agents that reason over evidence), and absorption (the validated state changes that let evidence compound across institutions). Each currently scales on a different curve. Reality-contact is bounded by biology, ethics, and capital, and stays that way regardless of compute. Model capacity is scaling with AI compute and architecture; the foundation models in protein structure, sequence biology, and materials are early evidence. Absorption is not yet a scaling object at all; it sits where institutional friction lives, and is the slowest of the three by orders of magnitude.

Constellate is the absorption layer. Reality-contact is the work of labs, clinics, and patients; model capacity is the work of AI labs and modeling groups; absorption is the work that nobody currently owns. The protocol commitment is narrower than "scale the model" or "scale the lab" because owning a narrower layer well is the leverage point. A working absorption layer makes reality-contact more reusable across institutions and makes model capacity trainable on data that compounds rather than scatters. The bet is not that absorption is more important than the other two inputs. It is that the other two are scaling already, and the missing input is the one that lets them compound together.

This is v0.1 of the Constellate architecture paper. Revisions will be made as the protocol evolves, the first corridor closes its loop, and the open problems in §11 are addressed.

How to cite. Blair, W. (2026). *The Constellate Architecture* (v0.1). constellate.science/whitepaper.