

PALO taxonomy and ontologies for agentic governance

Technical deep dive for the follow-up discussion with the European Commission

Fabrizio Degni · PALO Framework · Evidence snapshot 12 October 2026

PALO provides a versioned vocabulary for connecting lifecycle governance, delegated authority, controls, evidence and accountable decisions. Its semantic implementation comprises a released foundation, structured agentic contracts and a draft extension for dependencies and policy impact. These layers have different purposes and maturity. The central contribution is the explicit separation of what a system is, what an agent is authorised to attempt, what actually happened, what evidence supports that account and who accepts the resulting decision.

The two sections below explain **taxonomy and controlled dictionaries**, followed by **ontologies and their operational grounding**. The accompanying catalogues expose every schema entry and enumeration in the inspected schema set, together with the actual RDF classes, properties and shapes. They make the analysis inspectable beyond the narrative.

The scope is semantic design and evidence contracts. PALO-AI remains a developer-preview reference runtime. A coherent vocabulary, a valid record or a passing fixture does not establish legal conformity, deployment admission or operating effectiveness in an adopting organisation. The positive meeting response reported by the author is not used as evidence of institutional endorsement.

Sintesi italiana

PALO dispone di una base semantica concreta: un registro di concetti e relazioni, definizioni versionate del ciclo di vita, schemi per le evidenze e le decisioni, un'ontologia RDF/OWL e vincoli SHACL. La governance agentic aggiunge contratti operativi per identità, autorità delegata, richieste di azione, approvazioni, capacità esecutive, effetti attesi, ricevute, verifica degli esiti, idoneità dei dati e divulgazione delle informazioni.

Il punto da presentare alla Commissione è la continuità tra significato, decisione e prova: una regola di governance è collegata al caso concreto, alla versione applicabile, alle evidenze e all'autorità che decide. Occorre però distinguere ciò che è già rappresentato in OWL da ciò che oggi è formalizzato soprattutto in JSON Schema e nel runtime. L'estensione del grafo dello stack è dichiarata draft e non va presentata come una seconda ontologia già stabilizzata.

Le verifiche su questo snapshot hanno confermato 65 schemi leggibili, 61 fixture JSON con esito atteso, 40 corrispondenze su 40 tra file e digest del manifest, una fixture RDF valida e una invalida correttamente respinta. Sono controlli dei contratti acquisiti, non una nuova qualificazione dell'intero runtime. La riconciliazione aggiunge una corrispondenza esplicita fra le cinque attività storiche e le sei fasi canoniche, distingue i due significati precedentemente associati a CECR e uniforma l'esito di routing a "Redesign required". Tre ulteriori schemi PolicyWatcher completano il corpus. Le proposte di estensione ontologica restano separate dalle modifiche editoriali effettive.

Reading the implementation status

Status used here	Meaning
Released foundation	Artifact present in the PALO 3.1 semantic release and, where listed, bound by its manifest.
Executable reference contract	JSON Schema and associated reference code or fixtures exist; deployment effectiveness requires separate evidence.
Draft extension	The artifact explicitly identifies itself as an additive draft or preview.
Documented design	A narrative or UI definition exists, without assuming a corresponding OWL class or runtime guarantee.
Proposed next step	Analysis in this dossier; it has not amended the canonical PALO model.
TBFV	To Be Verified: an assertion requiring additional evidence before stronger use.

Verified inventory

Artifact family	Snapshot result	Interpretation
Semantic spine 3.1.0	46 nodes, 48 links, 8 node types	A framework and navigation graph, not a count of all domain concepts.
Lifecycle	6 phase definitions, 6 gate definitions	Frame, Classify, Assess, Control, Measure, Prove.
Control plane	12 domains, 31 controls, 38 indicators	Tailoring contracts; thresholds and control effectiveness remain contextual.
Mapping registry	409 atomic mappings	101 conditional and 308 supporting; no current mapping asserts total coverage.
Core OWL vocabulary	27 classes, 18 object properties, 18 datatype properties, 163 triples	Lightweight formal vocabulary.
Core SHACL	12 node shapes, 231 triples	Constraints on selected records and inherited semantic categories.
Stack model 0.1.0	25 entity types, 31 relationship types	Draft model reusing 7 foundation entity types.
Stack RDF and SHACL	18 declared classes, 25 object properties; 4 shapes	New declarations, not 25 additional classes or 31 additional properties.
JSON Schema corpus	65 schema files; 2,408 schema entries; 204 enum declarations	Entries include objects, branches and definitions; counts are not distinct business concepts.
Verification	61/61 JSON fixture expectations; 40/40 release item digests	Scoped checks described in Section 2.11.

Sources: [P01], [P02], [P03], [P04], [P05], [P06], [P07], [P08], [P09], [P10], [V01].

1 Taxonomy and controlled dictionaries

1.1 What PALO classifies and why

For this deep dive, **taxonomy** means the organised classification of governance concepts; a **controlled dictionary** specifies the permitted terms, identifiers, values and intended meanings; an **ontology** states formal classes and relationships; a **schema** constrains a record's structure; and an **executable policy** evaluates an action against current rules and inputs. These are complementary mechanisms. An allowed value has a defined lexical meaning, but only a valid decision process can give that value authority for a particular action.

PALO's taxonomy is best described as a set of connected facets. The lifecycle answers when governance work occurs. The agentic modality asks who acts, under whose authority, on which resources and with what autonomy. The evidence model asks which assertion is being made, what supports it and how it was evaluated. The stack extension adds technical dependency and policy-change context. No single tree captures all these dimensions without collapsing material distinctions. [P01], [P02], [P10], [P11]

The present implementation distributes its dictionaries across JSON registries, JSON Schema enumerations, RDF terms, JSON-LD contexts, source records and UI labels. It does not contain a single reviewed SKOS concept scheme unifying all these terms. A SKOS export is therefore an interoperability proposal, not a description of an existing PALO deliverable. [P01], [P07], [P08], [W01]

1.2 Framework layers and their authority

The December 2025 founding document supplies the conceptual roots: fairness/non-discrimination; transparency/explainability; accountability/responsibility; privacy/data governance; safety/robustness; human agency/oversight; and societal/environmental well-being. It evaluates use cases across ethical integrity, technical robustness/safety, business value/feasibility, legal/regulatory conformity and socio-environmental impact. These are normative evaluation dimensions and objectives, not assertions that a system has achieved them. The later semantic spine and agentic contracts turn parts of this conceptual structure into versioned identifiers and evidence requirements. [H01]

Layer	Governance subject	Concrete representation	Boundary
PALO	Lifecycle, ownership, assessment, controls, measurement and proof	Spine, gates, control packs, evidence schemas	Framework definitions do not authorise a deployment.
PALO-AM	Delegated action and human agency	Five operational object cards, authority/action-space matrix, simulator	A routing model is not a legal risk classification.
PALO-AI	Selected checks at the action boundary	Profiles, Action Claims, policy evaluation, approval, capabilities and receipts	Developer preview; deployment admission remains separate.
Stack governance extension	Dependencies from providers and services to systems, controls and cases	Typed dependency graph, RDF vocabulary and impact query	Draft extension; graph impact produces review work.
PolicyWatcher input	Observed changes in external public policies	Source-bound signals and applicability review	A change signal cannot independently decide applicability.

This distinction is part of the semantic design, not just product naming. The core ontology explicitly separates `RegimeDefinition`, `GovernanceFrameworkDefinition` and `EnforcementAdapterDefinition`, with pairwise disjointness

axioms. However, declaring classes does not automatically classify every UI label or component as an instance of them. [P07], [P10], [P11], [P12]

The spine has four evidence/authority classes: canonical-definition, source-backed-context, illustrative-local-preview and human-review-required. The first is authoritative **as a PALO definition**. None of the four, merely by classification, grants legal authority or changes the state of a case. The authoritative: true flag on canonical definitions must always be read within this scope. [P01], [P02]

The public semantic catalog, case workspace and assurance review are three workspaces. “Executive” and “Technical” are presentation lenses, not authenticated identities or access-control roles. Delivery maturity is another axis: the capability matrix distinguishes specified, prototype, implemented and production-ready. These values must not be merged with evidence authority or legal status. [P01], [P02], [P13]

1.3 The lifecycle taxonomy

Canonical phase and gate	Main question	Typical output	Agentic interpretation
Frame	What problem, purpose, population and accountable owner define the case?	Use-case brief	Justify delegated action and identify alternatives with less authority.
Classify	Which route and scrutiny apply?	Risk route	Classify action space, autonomy, data sensitivity and relevant obligations.
Assess	Which impacts and failure modes matter?	Impact assessment	Evaluate authority misuse, tool errors, delegation, human oversight and downstream effects.
Control	Which owned safeguards address the findings?	Control plan	Define allow-lists, policy checks, approval conditions, effect contracts and safe states.
Measure	How will performance and exposure be observed?	KPI/KRI register	Measure errors, interventions, drift, disclosure, unresolved outcomes and control performance.
Prove	What evidence supports the decision and next review?	Evidence bundle and gate decision	Reconstruct authority, execution, observation, exceptions and accountable acceptance.

Gate definitions list proceed, proceed-with-conditions, hold, redesign and stop. The Gate Decision Record additionally permits unresolved, preserving the difference between a concluded routing choice and a recorded undecided state. A conditional decision requires conditions; decisions after sequence 1 require the preceding decision identifier and digest. [P03], [P14]

The PALO-AM page also retains a five-phase historical presentation: ideation and screening; assessment and planning; development and validation; ethical deployment and monitoring; continuous improvement and decommissioning. This is not the same list as the canonical six-phase lifecycle. The editorial glossary and agentic page publish the following explanatory correspondence: historical phase 1 mainly informs Frame/Classify; phase 2 informs Classify/Assess/Control; phase 3 informs Control/Measure; phase 4 informs Measure/Prove; phase 5 informs Prove and the next cycle. That correspondence is a versioned interpretation for readers, not a one-to-one migration mapping or an executable gate rule. [P35] Canonical identifiers should remain the six current phases. [P03], [P11]

1.4 The five PALO-AM operational objects

Agent Identity Profile describes the agent and its provenance, owner and operating context. The narrative includes credential lifecycle, topology, clearance and revocation. The concrete Agent Profile schema represents `agentId`, `profileVersion`, `role`, `lineage`, `base model` and `systemPromptHash`, with optional `temperature`. Status can be `active`, `suspended` or `retired`. A prompt hash binds bytes; it does not prove that the model followed the prompt or retained all conversational requirements. [P11], [P15]

Agent Authority Profile describes permitted activity. The schema supplies tool allow-lists, operation types, read/write scopes, external-network permission, optional host constraints, argument schemas and a Vibe Gate requirement. Identity and authority coexist in the Agent Profile but remain conceptually distinct. A recognised agent is not automatically authorised to use every tool. [P15]

Agentic Risk Matrix routes a configuration using action-space impact and autonomy, modified by reversibility, sensitivity and potential impact. Narrative considerations additionally include third-party dependency, speed/volume and multi-agent complexity. Those additional considerations are not independent inputs to the inspected five-field simulator. [P11], [P16]

Agentic Control Layer describes the technical and organisational controls surrounding model proposals: policy evaluation, tool schemas, scoped credentials, approval checkpoints, observability, circuit breakers and change management. The presence of a control in this taxonomy is a governance expectation. Whether it is implemented and non-bypassable must be established for the actual connector and deployment. [P11], [P12]

Agentic Evidence Layer records proposals, tool calls, structured inputs/outputs, policy decisions, approvals, incidents, measures and retirement evidence. It does not require disclosure of a model's private chain of thought. The object cards are useful operational domains; they are not five correspondingly named OWL classes in the core vocabulary. [P07], [P11]

1.5 Action space, autonomy and the exact routing matrix

Facet	Controlled values in the simulator	Meaning
Action space	read, internal, cross-system, critical	Read only; internal write; cross-system action; financial, safety, legal or physical action.
Autonomy	supervised, low, medium, high	Human executes; agent proposes and human approves; agent acts within bounded rules; agent selects and executes independently.
Reversibility	reversible, partial, irreversible	Fully reversible; partially reversible; irreversible or hard to remedy.
Data sensitivity	public, internal, sensitive, restricted	Public/synthetic; internal; personal/commercially sensitive; special-category, secret or highly restricted.
Potential impact	low, moderate, high, severe	Localised through consequential rights, safety, livelihood or systemic impact.

The implemented base matrix is reproduced below. Lower numbered positive tiers demand stronger controls. 0 means redesign required in the simulator. [P16]

Autonomy	Read	Internal write	Cross-system	Critical
Supervised	4	4	3	3
Low	3	3	2	2
Medium	3	2	1	1
High	2	1	1	0

Tier 1 is “Maximum controls”; Tier 2 “Controlled”; Tier 3 “Supervised”; Tier 4 “Monitored”. If at least one of irreversible action, restricted data or severe impact is present, a positive base tier is escalated by **one tier only**, with a floor of Tier 1. The three escalation triggers do not accumulate into three tier changes. High autonomy plus critical action remains 0. Partial reversibility changes recommended controls/evidence but does not itself trigger the tier escalation. These details matter if a third party reimplements the taxonomy. [P16]

The reconciled static matrix and simulator both label the high-autonomy/critical combination “Redesign required”. The former static label “PROHIBITED” is retained only as a deprecated display alias in the editorial glossary. This is a PALO design rule and must not be equated with a prohibited practice under the EU AI Act. [P35] A legal classification would require a separate source-specific analysis. “Supervised” also appears both as an autonomy input and a tier label; these are different fields and should be qualified when exported. [P11], [P16]

1.6 Identity, authority and delegation dictionaries

The Agent Profile permits six operation values: `read`, `create`, `update`, `delete`, `execute`, `delegate`. Action Claims use the same operation vocabulary, while network intent is separately classified as `none`, `read`, `write` or `bidirectional`. An operation token is intentionally broad. The exact tool, resource, canonical path and argument schema determine what the action actually means. [P15], [P17]

Delegation uses a maximum depth, maximum subagent count, permitted subagent roles and a human-validation requirement. The profile schema allows a depth from 0 to 20 and a maximum subagent count from 0 to 1,000. These are schema bounds, not recommended safe operating settings. An adopting organisation must choose substantially narrower limits where appropriate and test enforcement. [P15]

Action Claim authority context distinguishes the human principal, agent identity, workload identity, delegation chain and tenant. Workload proof types include `mtls`, `dpop`, `http-message-signature` and `spiffe-svid`; the enumeration lists accepted representation categories and does not establish universal interoperability with every implementation of those technologies. ANS integration is upstream identity/status evidence and does not itself establish permission, data fitness or successful effect. [P17], [P18]

The swarm mandate extends the scope to a named owner, objective, coordinator, members, tenant/case, validity interval, budget and concurrency limit. Budget rules bind tool, operation, resource and path; the prototype’s narrative requires one unambiguous matching rule, a non-negative integer charge and a per-action maximum. Membership changes carry an expected membership digest and reason. A natural-language objective records an accountable intention; it is not a machine proof of goal alignment. [P19]

This yields a useful delegation taxonomy: principal versus agent versus workload; profile versus one-action authority; parent versus descendant; individual limit versus shared exposure; permission revocation versus confirmed interruption. Revoking future authority must not be described as undoing an action already executed or proving cancellation of an in-flight external effect. [P18], [P19]

1.7 The action and assurance dictionary

Term	Concrete meaning and key binding	Common misinterpretation to avoid
Agent Profile	Versioned identity, authority, delegation and evidence configuration	A profile is not a permit for a particular action.
Action Claim	A specific proposed action bound to agent, case, arguments, scopes, time, replay controls and, for newer versions, authority/effect/data contracts	A proposal does not establish permission.
Policy Decision	allowed, denied or pending_approval, with reasons, versions and claim digest	allowed does not mean executed or compliant.
Approval	Human decision concerning the exact claim digest and an expiry	Approval of changed arguments is a different decision.
Execution Capability	Short-lived, single-use reference authority bound to claim, decision, executor, verifier and target	It is distinct from an architectural capability in the stack graph.
Effect Contract	Resource selector, preconditions, expected and forbidden effects, verification and recovery policy	It verifies declared predicates, not every unstated expectation.
Execution Receipt	Signed reference record of execution status and request/result/pre-state digests	succeeded does not prove the intended business outcome.
Outcome Attestation	Verifier result against the effect contract and observed state	verified is scoped to these predicates and evidence.
Assurance Incident	Record connecting an adverse or unresolved assurance result to response/review	It is not automatically a legally reportable serious incident.
Compensating action	A new action proposed to address a previous effect	It needs its own authority cycle and is not a magical rollback.

The reference sequence is **Propose** → **Authorize** → **Approve when required** → **Capability** → **Execute** → **Receipt** → **Observe** → **Verify** → **Escalate when required**. This is a governance/assurance sequence, not a replacement for the six lifecycle phases. Approval is conditional; it should not be depicted as necessarily occurring in every action. [P12], [P17], [P20]

Action Claim versions 1.1.0 through 1.4.0 are represented in the schema. Version-specific conditions matter: 1.3 introduces the identity-bound authority context; 1.4 additionally binds the Data Fitness Decision and Disclosure Contract by identifier and digest, subject and purpose. An integration must evaluate the actual schema branch rather than treating schemaVersion as decorative metadata. [P17], [P21]

1.8 State vocabularies must remain separate

Record or field	Exact values in the inspected contracts
Profile status	active, suspended, retired
Policy decision status	allowed, denied, pending_approval
Approval status	pending, approved, denied, expired, cancelled
Execution capability status	issued, consumed, expired, revoked
Execution receipt status	succeeded, failed, unknown
Outcome attestation status	verified, mismatch, inconclusive
Predicate check	pass, fail, unknown
Data fitness decision	allowed, denied, review_required
Disclosure contract	active, revoked
Disclosure receipt	verified, mismatch, inconclusive
Evidence claim	asserted, under-review, accepted, rejected, superseded
Evidence evaluation	sufficient, partial, insufficient, unverified
Mapping approval	draft, in-review, approved, rejected, superseded
Mapping coverage	total, partial, conditional, supporting
Gate decision outcome	proceed, proceed-with-conditions, hold, redesign, stop, unresolved

These values are not interchangeable booleans. An approved mapping is a publication/review assertion; an approved action is a particular human approval; an accepted evidence claim has still a distinct evaluation record. Likewise, `unknown`, `unverified`, `inconclusive` and `unresolved` preserve different kinds of missing knowledge. The catalogue keeps each enum attached to its schema and field path instead of deduplicating words that only look similar. [P14], [P17], [P20], [P21], [P22], [P23], [C01]

A notable implementation detail is the effect evaluator's aggregate precedence: a check with `unknown` makes the aggregate `inconclusive` before the code considers failing checks. Therefore a record may contain both a failed predicate and an inconclusive aggregate. Consumers must retain and inspect per-predicate results rather than hiding an observed failure behind the aggregate label. This is an observation of the inspected reference code, not a proposed alteration. [P24]

1.9 Data and information disclosure taxonomy

The data-assurance contracts classify the governed subject as `dataset`, `data-product`, `model`, `ai-system`, `agent`, `tool`, `access-entitlement` or `other`. They connect the subject and tenant to a purpose, source references, quality and lineage assertions, access state, source authority, approval, freshness and incident context. External metadata is evidence to be evaluated; importing it does not independently attest its truth. [P21], [P25]

Data fitness means suitability for a stated use under a particular policy and current evidence. The reference evaluator supports quality thresholds, age limits, source authority, ownership/approval, lineage, access, incidents, prohibited classifications and permitted purposes. Missing evidence has explicit deny/review handling. Conflicting

current evidence is evaluated conservatively rather than selecting the most favourable assertion. This remains a rule-based fitness decision, not a universal numerical quality score. [P21]

Disclosure is separately governed because appropriate access does not imply appropriate transmission. The Disclosure Contract classifies egress as zero-row, aggregated or row-level; trace modes as disabled, metadata-only or redacted-payload; and binds fields, row limits, redactions, destinations, provider/model/region/host, retention, export and output schema. Aggregation does not automatically establish anonymity. [P21], [P25]

The lawful-basis field permits consent, contract, legal-obligation, vital-interests, public-task, legitimate-interests, not-applicable and pending-review. These tokens record a classification supplied to the governance process. They do not determine whether a basis is legally available. The documented active-contract rule excludes pending-review and requires an approver; that runtime condition is stronger than simply observing that the string exists in a schema enum. [P21], [P25]

An observation records actual fields/rows read and disclosed, sensitive categories, redactions, destination, trace behaviour and digests. The Disclosure Receipt compares that observation with the contract. Missing trustworthy observation remains inconclusive. Because the reference executor and signing material share an in-process boundary, its signed observation is not independent proof that all external network activity was observed. [P21]

1.10 Controls, indicators and applicability dictionaries

The 12 governance-control domains are fairness and subgroup testing; system cards and explanations; notice, appeal, review and remedy; Article 50 transparency; data lifecycle; GPAI and systemic risk; serious incidents and decommissioning; accessibility; environmental performance; AI-literacy effectiveness; ISO/IEC 42001 AIMS overlay; and PALO-AI production boundary. Each domain names its objective, applicability questions, controls, indicators, gates, evidence contracts, completion criteria, stop conditions and residual boundary. [P04]

The control library classifies controls as directive, preventive, detective or corrective. It includes purpose boundaries, provenance, meaningful review, supplier due diligence, adversarial testing, incident response, material change gates, decision traceability and agentic authority, alongside the domain-specific additions. “Not applicable” must be supported by a reasoned accountable decision; omission is not an applicability outcome. [P04], [P05]

The 38 registered KPI/KRI definitions carry formula, cadence, direction, thresholds, linked controls/gates/sources and caveats. Their source explicitly labels thresholds as illustrative. A metric can indicate documentation coverage while saying little about substantive correctness. For example, provenance coverage does not establish permission or fitness, and completed reviews do not establish that reviewers had enough time or independence. [P06]

The agentic page adds 11 named indicators: HOR, RTD, ABI, TCER, AISI, MACR, MTTI, CECR, AIRT, TDS and PVR. These are not 11 automatically additional canonical registry entries: some overlap with registry concepts, and others remain narrative definitions. ABI and TDS lack a fully specified numerical protocol in that page. The earlier CECR collision is resolved in the editorial glossary: `palo-am:cascading-error-containment-rate` retains CECR, while `palo-am:simulator-control-effectiveness-coverage` names the separate simulator recommendation without that abbreviation. The latter still needs a numerical protocol before it can be treated as an operational metric. The glossary does not add either concept to the canonical KPI/KRI registry. [P35] [P11], [P16]

Oversight and bias crosswalks in the briefing are separate research artefacts. Their relationship to the core should be expressed as versioned, source-scoped mappings, retaining draft-source status, clause interpretation and reviewer responsibility. They should not be counted among the 409 core mappings merely because they also use the word crosswalk. Private draft source documents are not required for the public dictionary export. [P26]

1.11 Evidence, mappings and versioned meaning

The atomic evidence model distinguishes an **artifact**, an **evidence claim**, an **evaluation** and a **bundle manifest**. The artifact identifies content and provenance. The claim states a falsifiable assertion about one target in one case scope. The evaluation records the reviewer, assessment-run context, outcome, rationale and quality dimensions. The bundle composes digest-bound records. This prevents a document's presence from being mistaken for successful evaluation of the proposition it is meant to support. [P22], [P23], [P27]

Evidence Evaluation quality values separately cover relevance, integrity, recency and independence, each in the range 0–1. The schema bounds the values but does not supply a universally valid scoring method. A consumer should preserve the method and rationale and avoid treating a mean of four values as proof of assurance. [P23]

Each core mapping is atomic: one source version, one target version, mapping type, coverage, conditions, rationale, basis, approval and provenance. The 409 records comprise 101 control-to-gate, 55 control-to-indicator, 72 control-to-source, 11 gate-to-source, 106 indicator-to-gate and 64 indicator-to-source relationships. Every control-to-gate record is conditional; all other current records are supporting. Although the schema permits **total** and **partial**, those values do not occur in this inspected registry. [P09]

These mapping counts measure authored relationships, not coverage of 409 legal obligations. The mapped source version is a PALO registry reference, not a statement that every clause of an external standard is represented. A mapping's **approved** publication state must not be transformed into a conformity assertion. [P09], [P28]

1.12 A reviewable dictionary entry and maintenance process

For an interoperable exchange, a dictionary entry should contain a stable concept IRI, preferred label, definition, scope note, aliases/acronyms, broader/related concepts, schema and field paths, allowed values, version, status, owner/reviewer, provenance and change history. Language labels should use explicit tags. Semantic identity should remain stable across translation; an English label and its Italian translation are not different control concepts.

PALO already supplies substantial ingredients, especially identifiers, versions, source references, authority boundaries, schemas and mappings. The accompanying extraction preserves these machine-level facts. Additional bilingual definitions, ownership and synonym resolution are a **proposed curated layer**, not fields assumed to be populated in every existing artifact. [P01], [P02], [C01]

The maintenance sequence should be: propose term; identify existing meanings; choose reuse or a new identifier; attach source and scope; review consequences for schemas, policies and user interfaces; test representative and adversarial records; approve a version; publish a digest-bound release; preserve old identifiers or explicit migration. Major meaning or cardinality changes require the existing change-impact and migration process. A wording clarification should not silently change what actions a policy permits. [P28], [P29]

1.13 PolicyWatcher transport and review dictionaries

Three additional JSON contracts make external-policy signals more precise. They complement the individual PolicyWatcher signal contract and the draft dependency graph; none grants policy applicability or action authority. [P34]

Contract	Exact controlled vocabulary and bindings	Governance interpretation
Signal Batch	mode = complete-active-snapshot; locale = en or it; count and limit up to 25; hasMore, nullable nextCursor, individual signal schema reference	A page is one part of a snapshot. Completeness cannot be inferred from the first page or a successful HTTP response.
Operational Signal Registry	Transport state: not-synchronized, healthy, degraded, unavailable; complete-traversal and stale flags; last attempted and successful sync; collection and signal digests; active and revoked entries	Transport health, freshness and signal status remain separate. A valid or accepted import is not an accepted governance finding.
Browser-local Review Ledger	localOnly = true; source-registry and signal digests; transport status active or revoked; review state pending-human-review, in-review, routed-to-case-file, reviewed-no-action, rejected	A local review state records triage. Routing to a Case File is not a signed gate decision or a production approval.

The registry separately names error codes `network-error`, `http-error`, `schema-error`, `response-too-large`, `pagination-error` and `capacity-exceeded`. Those classifications preserve the reason a snapshot cannot be relied on as complete. Registry statistics count received, accepted, rejected, changed, active and revoked signals; the schema fields must accompany the number, because “accepted” here describes the import process.

The review ledger binds an assessment to the imported signal digest and source-registry digest. If the transport later reports revocation or a different digest, the prior local review must remain attributable to the version actually seen. The JSON contract supplies those bindings; an adopting implementation must enforce reconciliation, reviewer identity and persistence. Browser-local data must not be represented as an organisation-wide authoritative review log.

The complete schema catalogue now includes the union of the inspected agentic and semantic contracts, these three transport/review schemas, and three explicitly draft or local extension schemas. Equal counts in two repositories do not imply identical memberships. The published field dictionary therefore includes the schema filename, path and constraints for every entry rather than presenting an undifferentiated list of terms.

2 Ontologies and their operational grounding

2.1 Architecture from concepts to action records

PALO's semantic architecture has four cooperating layers. The semantic spine and lifecycle registries organise framework meaning. The core ontology supplies a vocabulary for definitions, runtime records, immutable records, decisions, evidence, mappings and release identity. JSON Schema supplies richer record contracts, including many agentic objects not declared in the core OWL file. Reference code then evaluates temporal, cryptographic, policy and state-transition conditions that a static ontology alone cannot enforce. [P01], [P07], [P08], [P17], [P24]

The draft stack ontology extends the vocabulary to agents, tools, models, providers, services, policy versions and applicability assessments. Its JSON graph model supplies typed edge rules and operational query context. RDF and JSON contracts must therefore be described together. Saying "PALO has an ontology" is supported; saying "all agentic governance is already represented and enforced by OWL reasoning" is not. [P10], [P30]

2.2 The complete core class taxonomy

All 27 classes declared in the core file are listed below. The grouping follows actual subclass relations, with three additional canonical-definition subclasses declared near the end of the file. [P07], [C02]

Class or family	Declared classes	Intended role
Root	SemanticEntity	Shared semantic category.
Canonical definitions	CanonicalDefinition; LifecycleDefinition, PhaseDefinition, GateDefinition, AuthorityClass, WorkspaceDefinition, RegimeDefinition, GovernanceFrameworkDefinition, EnforcementAdapterDefinition	Versioned framework meaning and authority separation.
Runtime records	RuntimeRecord; GateInstance, EvidenceArtifact, ControlImplementation, GovernanceCase	Case-scoped occurrences or operational records.
Immutable records	ImmutableRecord; EvidenceClaim, EvidenceEvaluation, EvidenceBundleManifest, EvidenceBundleItem, MappingRecord, SemanticReleaseManifest, SemanticReleaseItem, MetricObservation, AssessmentRun	Records whose historical identity should be preserved.
Decisions	DecisionRecord under ImmutableRecord; GateDecisionRecord under DecisionRecord	Attributable decisions tied to input snapshots.

This is intentionally a lightweight hierarchy. For example, `EvidenceArtifact` is a subclass of `RuntimeRecord`, not `ImmutableRecord`, even though its documentation emphasises immutable content identity. The description of immutable evidence and the precise OWL inheritance must not be conflated. Likewise, only the three regime/framework/adaptor classes have the explicit pairwise disjointness described above; the file does not make every conceptual separation an OWL disjointness axiom. [P07]

2.3 Core relationships and cardinalities

The core declares 18 object properties. The complete set is `hasPhaseDefinition`, `hasGateDefinition`, `instantiatesGate`, `gateInstanceCase`, `decisionForGate`, `decisionActor`, `usesArtifact`, `supportsTarget`, `claimScope`, `evaluatesClaim`, `containsBundleItem`, `bundleItemReferences`, `mapsSourceVersion`, `mapsTargetVersion`, `pinnedToReleaseManifest`, `containsReleaseItem`, `releaseItemReferences` and `supersedes`. [P07]

Relationship	Declared domain and range where present	SHACL or operational qualification
Lifecycle → phase/gate definitions	LifecycleDefinition → PhaseDefinition/GateDefinition	At least one of each in the generic SHACL shape; the released registry has six.
Gate instance → definition and case	GateInstance → GateDefinition/GovernanceCase	Exactly one definition and one case, plus one string caseId.
Gate decision → gate	GateDecisionRecord → GateInstance	Exactly one gate; actor, authority basis, rationale and input snapshot required.
Claim → artifact/target/case	EvidenceClaim → EvidenceArtifact/SemanticEntity/GovernanceCase	Exactly one of each, statement and integrity reference.
Evaluation → claim	EvidenceEvaluation → EvidenceClaim	Exactly one claim and one allowed evaluation outcome.
Bundle → items/release	EvidenceBundleManifest → EvidenceBundleItem; release link range SemanticReleaseManifest	At least one item and exactly one pinned release.
Mapping → source and target versions	MappingRecord → CanonicalDefinition	Exactly one source and one target, coverage, rationale and approval state.
Record → superseded record	ImmutableRecord → ImmutableRecord	Directional historical link; storage immutability is a separate concern.

OWL domain and range declarations support logical inference; they are not equivalent to rejecting malformed records. If a triple uses a property whose domain is `EvidenceClaim`, a reasoner can infer that type. A validation engine must separately check the intended input constraints. This distinction is essential when discussing “enforcement” with ontology specialists. [P07], [W02], [W03]

The core’s 18 datatype properties are `stableId`, `definitionVersion`, `caseId`, `recordedAt`, `appendOnly`, `contentDigest`, `recordDigest`, `inputSnapshotHash`, `authorityBasis`, `rationale`, `claimStatement`, `integrityReference`, `evaluationOutcome`, `coverageType`, `approvalStatus`, `manifestDigest`, `bundleDigest` and `ordinal`. Some other terms used in contexts or shapes are not explicitly declared as datatype properties in the ontology. RDF permits vocabulary use without declaration; it still warrants documentation in a curated export. [P07], [P08], [C02]

2.4 What the core SHACL profile actually checks

The 12 shapes cover `SemanticEntity`, `CanonicalDefinition`, `ImmutableRecord`, `LifecycleDefinition`, `GateInstance`, `GateDecisionRecord`, `EvidenceArtifact`, `EvidenceClaim`, `EvidenceEvaluation`, `EvidenceBundleManifest`, `MappingRecord` and `SemanticReleaseManifest`. They use property cardinality, datatype, pattern, class, minimum length, fixed value and enumeration constraints. [P08]

Every targeted semantic entity needs exactly one non-empty `stableId`; every canonical definition one dotted three-part version. An immutable record needs a date-time, a SHA-256-shaped record digest and `appendOnly` true. Digests are checked for lexical format here; a string matching the pattern is not proof that it was computed from the correct payload. The release verifier’s byte-hash comparison is a distinct check. [P08], [P28]

Superclass shapes matter. In the supplementary test, a `GateDefinition` with no stable ID or version produces two violations when the ontology is supplied, even without enabling an additional inference regime. A consumer that omits the class hierarchy or uses a validator with different target handling can obtain weaker results. Validation configuration belongs in the exchange profile. [V01], [W03]

The profile does not declare every node closed, does not provide a bespoke shape for every class and does not enforce all operational invariants. For example, naming something `ImmutableRecord` or requiring `appendOnly` true cannot stop a database administrator rewriting history. Schema and shape checks need persistence controls, version

governance and audit procedures to realise that intention. [P07], [P08]

The published JavaScript semantic validator uses N3 parsing and an in-repository implementation of the constraint subset used by the profile. It checks required class presence, namespace agreement, the invariant registry and valid/invalid fixtures. It is not evidence of conformance to the complete SHACL specification. This review adds independent pySHACL checks of the existing fixture pair, with the ontology supplied, no remote imports and meta-SHACL enabled. Those checks strengthen evidence for these fixtures, not for all possible graphs or the whole runtime. [P31], [V01]

2.5 Definition, case, evidence and decision as different objects

The distinction between `GateDefinition`, `GateInstance` and `GateDecisionRecord` separates three questions: what the gate means; how it is instantiated for this case; and which decision was made against which evidence at a particular time. Changing a definition should not retroactively rewrite earlier case decisions. A later decision can supersede a prior one while preserving both. [P03], [P07], [P14]

Similarly, the evidence chain is **artifact** → **claim** → **evaluation** → **decision**, not “file uploaded → approved”. A security-test report can support a claim that one connector rejected a particular unauthorised operation. Its scope is that connector, test, version, method and case. It does not automatically support a claim that all connectors are non-bypassable, nor that the deployment is legally compliant. [P22], [P23], [P27]

A Semantic Release Manifest binds the exact definition and schema bytes supporting the exchange. A bundle manifest pins the release and constituent records. This makes reconstruction possible: a reviewer can identify the vocabulary and record contracts that applied when a decision was produced. The manifest does not contain the truth of the world; it establishes the identity of the files being used. [P28]

2.6 The draft stack ontology and the 31 typed relations

The stack model reuses seven core classes: `GovernanceCase`, `ControlImplementation`, `EvidenceArtifact`, `EvidenceClaim`, `EvidenceEvaluation`, `GateInstance` and `GateDecisionRecord`. It declares 18 additional classes: `Layer`, `Capability`, `GovernancePlane`, `ComponentVersion`, `AISystemVersion`, `Agent`, `ModelVersion`, `Tool`, `DataAsset`, `Provider`, `Service`, `PublicPolicy`, `PolicyVersion`, `ChangeSignal`, `ApplicabilityAssessment`, `RequirementVersion`, `ControlDefinition` and `Actor`. Together these form the model’s 25 entity types. [P10], [P30]

Relationship family	Exact graph relationship types	Question supported
Architectural capability	PROVIDES_CAPABILITY, BELONGS_TO_LAYER, GOVERNS_LAYER, INFORMS	Which components and governance planes provide or govern capabilities?
Provider and technical dependency	OFFERS, USES_SERVICE, DEPENDS_ON, USES_MODEL, CALLS_TOOL, READS	Which systems depend on a changed service, model, tool or data asset?
Public-policy versioning	HAS_VERSION, FROM_VERSION, TO_VERSION, AFFECTS_SERVICE	Which two policy versions are compared and which service mapping is accepted?
Case and applicability	GOVERNS_VERSION, ASSESSES_SIGNAL, ASSESSES_SYSTEM, IN_CASE, CONSIDERS_REQUIREMENT, REVIEWED_BY	Which case, system, requirement and reviewer contextualise the change?
Control implementation	IMPLEMENTS, ADDRESSES_REQUIREMENT, APPLIES_TO_SYSTEM	Which deployed implementation addresses which control/requirement for which system?
Evidence	USES_ARTIFACT, SUPPORTS_TARGET, EVALUATES_CLAIM	What assertion and evaluation support the review?

Relationship family	Exact graph relationship types	Question supported
Gate and decision history	REVIEWS_SYSTEM, DECISION_FOR_GATE, DECIDED_BY, BASED_ON_ASSESSMENT, SUPERSEDES	What decision was made, by whom, on what assessment, replacing which earlier decision?

The JSON model constrains source and target types for each edge. The RDF extension declares 25 new object properties and reuses six from the core. Its new property declarations principally contain explanatory comments; they do not reproduce all JSON endpoint rules as OWL domain/range axioms. Therefore “31 typed relations” describes the graph model, not 31 fully axiomatized new OWL properties. [P10], [P30], [C02]

The architecture view has layers L0 infrastructure/deployment, L1 evaluation/observability, L2 models/inference, L3 orchestration/tools, L4 knowledge/storage, L5 representation/retrieval, L6 acquisition/provenance, L7 context/memory and L8 runtime security/controls, plus applications and human interaction. This is a capability map, not a compulsory sequential execution pipeline. Governance and evaluation cross layers; components and capabilities can participate in more than one layer. [P10]

The four stack shapes cover policy version, change signal, applicability assessment and requirement. Requirement kind is law, standard, internal-rule or contractual-term. `PublicPolicy` and `RequirementVersion` are explicitly disjoint: a provider statement and a legally or organisationally applicable requirement are different objects. The applicability assessment connects the signal to the specific system and case, with rationale and purpose. [P30]

2.7 Time, knowledge and policy-impact queries

The graph distinguishes **valid time**, the period for which an assertion applies, from **recorded time**, when it became known. `validFrom` and optional `validTo` use a half-open interval. Query parameters `asOf` and `knownAt` allow a reviewer to ask both “what applied then?” and “what could have been known then?”. Historical analysis should not import later evidence without making that change visible. [P10]

The impact query starts from accepted current service mappings and follows reverse dependencies through `DEPENDS_ON`, `USES_SERVICE`, `USES_MODEL`, `CALLS_TOOL` and `READS`. It returns witness paths and relevant cases, controls, gates, decisions and evidence. Provisional mappings are excluded from accepted traversal and reported as incomplete coverage. A missing path does not prove that a system is unaffected. [P10]

The statuses `review-required`, `unmapped`, `source-versions-unavailable`, `signal-unavailable-at-query-time`, `baseline-established` and `no-mapped-system-in-scope` describe different epistemic conditions. None is a legal approval or a guarantee of no impact. The documented query profile limits graph size and traversal depth; a depth-limit warning is evidence of incomplete traversal, not a safe result. [P10]

This graph query must also be distinguished from the reference runtime’s **continuous-assurance signal**. The latter can invalidate purpose-bound Data Fitness Decisions and revoke unconsumed capabilities for the matching tenant/subject. The public-policy impact query itself does not automatically rewrite controls, reopen gates or revoke capabilities. These are separate mechanisms and authority paths. [P10], [P21]

2.8 Where agentic semantics are operational today

The core OWL vocabulary does not declare `ActionClaim`, `ExecutionCapability`, `EffectContract`, `ExecutionReceipt`, `OutcomeAttestation`, `DataFitnessDecision`, `DisclosureContract` or `SwarmMandate` as classes. These are nevertheless substantial PALO semantic contracts: their meanings are operationalised through JSON schemas, exact bindings and reference runtime code. Treating absence from OWL as absence from the project would be as inaccurate as claiming complete RDF coverage. [P07], [P17], [P20], [P21], [P24]

An Effect Contract groups predicates as preconditions, expected effects and forbidden effects. The schema’s operator set includes `exists`, `notExists`, `equals`, `unchanged`, `changedTo`, `deltaWithin`, `numberWithin`, `containsAll` and `typels`, with version-specific legacy restrictions. Resource selectors bind the target and can bind tenant context.

Verification specifies timing and inconclusive handling; recovery specifies hold/review or a proposed compensating action. [P20]

At the action boundary, the inspected code checks schema semantics, identity and tenant consistency, policy, current data governance, claim digests and approval binding. Capabilities tie together the exact claim, decision, executor, verifier and target. Receipts and attestations preserve separate execution and verification outcomes. These checks support reference demonstrations; they do not establish that external connectors cannot be bypassed in an arbitrary deployed architecture. [P12], [P21], [P24]

Several action-assurance contracts use the reference HMAC-SHA256 mechanism; the runtime also contains a separate Ed25519 evidence-envelope path. These mechanisms must not be described as interchangeable. A valid HMAC establishes integrity/authenticity relative to possession of the shared secret; it does not provide public-key non-repudiation between mutually untrusted parties. Neither a signature nor an unkeyed digest proves that an input observation was truthful. Key custody, observer trust, external anchoring and independent validation require explicit deployment evidence. [P15], [P21], [P24]

2.9 Worked example connecting taxonomy, ontology and execution

Consider a **synthetic** refund agent for a support case. A human owner authorises refunds within a defined resource, tenant, per-action limit and shared budget. This example illustrates the contracts and makes no claim about an actual financial deployment. [P12], [P19]

1. **Frame and classify.** Record the business purpose, affected customer, accountable owner and whether an agent is necessary. Classify the action as financially consequential; classify autonomy separately. A simulator tier informs scrutiny but does not authorise the refund.
2. **Define authority.** Register the agent profile and delegated scopes. If several agents share exposure, bind them to the same mandate and budget. A permission for each agent must not multiply the total approved budget.
3. **Establish data fitness.** Identify the order/customer-data subject, purpose and current evidence. A fitness decision for another tenant or purpose cannot be reused.
4. **Bound disclosure.** Permit only the needed fields and destinations. A refund may succeed while still violating the disclosure contract, so the two outcomes remain separately observable.
5. **Construct the Action Claim.** Bind tool, operation, arguments, target, nonce, sequence, idempotency key, expiry, authority context and effect/data contracts. The proposed amount is part of the content being authorised.
6. **Evaluate and approve.** Evaluate current policy and obtain an exact-claim approval when required. Editing the amount or beneficiary after approval changes the claim and requires a new evaluation/approval path.
7. **Issue and consume the capability.** Bind executor and verifier, reserve exposure and execute through the governed connector. A lost response must not by itself justify an ungoverned duplicate refund.
8. **Record execution and observe effect.** Preserve receipt digests; obtain the authoritative post-state needed to evaluate the contract. A transport success is not enough.
9. **Evaluate outcome and disclosure.** Expected predicates might include the intended refund amount and account change; forbidden predicates might include an unintended beneficiary or disallowed data transmission. Missing observation remains inconclusive.
10. **Build the evidence chain.** An artifact contains the bounded execution evidence; a claim states exactly what it supports; an evaluation judges it; a gate decision records accountable acceptance, conditions or hold. Pin the semantic release and preserve later supersession.

Adverse variant	Required semantic distinction	Evidence the reviewer should request
Data permission withdrawn after initial approval	Historical permission versus current fitness/capability validity	Signal, invalidation time, revocation and attempted execution record.
Budget available to two agents but insufficient for a third	Individual permit versus aggregate exposure	Reservation/commit history and proof the denied connector did not run.
Refund API reports success but ledger cannot be read	Execution succeeded versus outcome inconclusive	Receipt, failed/missing observation and held incident.
Wrong field disclosed while refund effect is correct	Business effect versus disclosure conformity	Disclosure observation, mismatch checks and response record.
Coordinator revoked while an action is in flight	Future authority revoked versus confirmed external cancellation	Lease/cancellation acknowledgement and eventual effect observation.
New evidence contradicts the previous evaluation	Original decision versus superseding decision	Both records, input hashes, rationale and time/knowledge context.

2.10 Competency questions and interoperability

An ontology's usefulness can be assessed through **competency questions**: specific questions its representation should answer with evidence. The following questions distinguish existing representation from additional work.

Question	Existing support	Remaining boundary
Which definition/version governed this gate?	Gate Instance, lifecycle definition and release manifest	Consumer must resolve the pinned artifacts.
Which artifact supports this claim and how was it evaluated?	Artifact/claim/evaluation relations and schemas	Truth, method quality and independence require review.
Who decided and what inputs were considered?	Gate Decision Record actor, authority basis, rationale and snapshot hash	Strings do not independently authenticate authority.
Which systems depend on a changed provider service?	Draft graph, accepted mappings and witness-path query	Inventory/mapping completeness and currentness.
Was a proposed action approved exactly as executed?	Claim digest, approval, capability and receipt contracts	End-to-end connector evidence and deployment controls.
Was authority still current at execution?	Runtime revalidation, expiries and revocation mechanisms	Race conditions and distributed boundary qualification.
Which data could leave and which did leave?	Disclosure contract, observation and receipt	Trustworthiness and completeness of the observer.
What was known at an earlier decision time?	Valid/recorded time and snapshot query	Availability of complete historical snapshots.
Can another organisation reconstruct the same graph and validation result?	Public vocabulary and schemas	Complete JSON-to-RDF mapping and conformance profile remain to be established.

Question	Existing support	Remaining boundary
Can every agentic contract be queried in RDF?	Partial vocabulary and rich JSON contracts	A complete agentic RDF module and conversion are proposed work.

Interoperability should preserve semantic boundaries rather than maximise the number of vocabulary links. The following are candidate alignment directions; no implemented PALO mapping is asserted solely from conceptual similarity.

Vocabulary	Useful alignment target	Required care
SKOS	Concept schemes, preferred/alternative labels, broader/narrower relations, mapping annotations	Distinguish related or close meanings from exact equivalence; preserve local scope.
PROV-O	Entities, activities, agents and derivation/association of evidence	A business “agent” and <code>prov:Agent</code> do not establish authorised delegation by themselves.
ODRL	Permissions, prohibitions, duties, parties and constraints	PALO action approval and effect verification need explicit operational semantics beyond a policy description.
DPV	Data processing purposes, personal-data concepts, actors, risks and measures	Preserve jurisdiction, evidence and reviewer decisions; vocabulary reuse does not establish a lawful basis.
DCAT	External datasets, distributions, data services and catalog metadata	PALO’s evidence reference and AI-system inventory need not become a duplicate data catalog.
JSON-LD	Stable IRIs and typed links for exchange	A context alone is not a complete serializer for an arbitrary graph envelope.
OWL and SHACL	Conceptual axioms and record validation	Keep inference, validation, authorisation and effect observation separate.

Primary specifications: [W01], [W02], [W03], [W04], [W05], [W06], [W07], [W08]. These alignments require term-by-term mappings, counterexamples, directionality, versions, scope and round-trip tests. In particular, avoid blanket `owl:equivalentClass`, `owl:sameAs` or `skos:exactMatch` assertions for concepts whose authority, time or case scope differs.

2.11 Reproducible verification and what it establishes

The source snapshot combines resident working-tree files, identified Git objects where accessible and exact public artifacts recovered for cloud-only files. Provenance and SHA-256 values are retained in the research manifests. The 40 core release items all match the inspected semantic manifest, establishing integrity of the selected foundation snapshot. Other working-tree and draft-extension material has separate version history; a matching digest does not prove current deployment or current source review. [V01]

The reproducible review script parsed all four Turtle files, enumerated classes/properties/shapes, checked 65 JSON schemas structurally, extracted 2,408 schema entries and 204 enum declarations, and validated 61 named valid/invalid JSON fixtures against their matched contracts with format checking enabled. All 61 produced the expected valid/invalid classification. This is not a claim that every schema has a complete negative-test suite. [V01]

The independent RDF run used the supplied core ontology and shapes: the valid fixture conformed; the deliberately invalid fixture produced 10 SHACL results. An incomplete `GateDefinition` probe was rejected for missing stable identity and version. Meta-SHACL validation was enabled. The stack vocabulary and shapes were parsed and

inventoried; no stack-instance SHACL conformance suite was run in this review. [V01]

The 14 named invariants provide a useful control map: stable identity; canonical versions; preservation of released definitions; definition/instance separation; gate definition/case binding; attributable immutable decisions; atomic claim scope; separate evaluation; manifest-based bundles; version-specific mappings; release pinning; distinct delivery/authority classifications; no creation of legal or deployment authority by framework guidance; and accountable review of non-authoritative signals. Their stated enforcement mechanisms range from schemas and generators to documentation and module contracts. They must not be presented as 14 universally machine-proven properties. [P32]

2.12 Gaps, ambiguity and a bounded next extension

Finding	Current interpretation	Recommended disposition
Five historical activities and six canonical phases	Editorial correspondence 1.0 is now explicit	Retain six-phase canonical IDs; treat the teaching correspondence as many-to-many.
CECR acronym previously had two meanings	Separate editorial concept IDs and simulator wording now distinguish them	Define a numerical protocol for control effectiveness and coverage before operational use.
Routing and legal prohibition	Both UI views now use “Redesign required”; old display alias is deprecated	Preserve independent legal classification.
Agentic objects primarily in JSON	Operational contracts are richer than the OWL vocabulary	Add a focused agentic ontology module tied to existing contract IDs.
JSON-LD contexts do not fully map all envelopes/fields	Vocabulary portability is partial	Define explicit transforms and round-trip loss tests.
Some classes have no dedicated shape; some terms lack declarations	Lightweight profile, not exhaustive closed validation	Publish shape coverage and intended open/closed boundaries.
14 invariants mix automated and procedural controls	A registry entry does not prove enforcement	Attach each invariant to code/tests/operational evidence and residual limits.
Signatures and append-only flags have limited scope	Integrity and storage assertions are not truth or independent assurance	Add deployment-specific key, observer, persistence and audit evidence.
Bilingual labels and synonym governance are distributed	No unified reviewed dictionary release	Curate a bilingual concept registry and acronym policy.
Draft-source oversight/bias mappings have a separate lifecycle	Research alignment is not adopted-standard conformity	Preserve exact source edition, interpretation, reviewer and revision status.

The smallest useful next semantic release would cover one complete chain: **Agent** → **delegated mandate/authority** → **Action Claim** → **policy decision** → **optional approval** → **execution capability** → **execution** → **receipt** → **outcome attestation** → **evidence claim/evaluation** → **gate decision**. Data-fitness and disclosure links should be included because Action Claim 1.4 already depends on them. It should reuse existing identifiers and contracts instead of renaming the runtime around a new ontology. This is a proposed extension, not an implementation delivered by this dossier.

A bounded acceptance package would include: a bilingual concept register; explicit proposed OWL classes and properties; JSON-to-RDF transformations; SHACL coverage tied to schemas; positive and negative synthetic examples; competency queries; temporal/revocation/case/tenant counterexamples; lossless round-trip checks for the

agreed profile; and an immutable release manifest. Test missing observations, expired approvals, changed arguments, stale fitness, conflicting evidence, shared budgets and a revocation during execution. Keep broader legal applicability and production admission outside the semantic pass/fail criterion.

The useful question for the Commission is therefore concrete: **which minimal vocabulary and evidence-exchange profile would allow independent parties to reconstruct and challenge a delegated-action decision consistently?** PALO already supplies a substantial foundation for that exercise. The next step is to agree the bounded profile, evidence and acceptance criteria, then demonstrate interchange and failure behaviour against it.

2.13 Additional insights from GLM and their evidence review

A GLM-5.3-Flash consultation on 12 October 2026 challenged the public artifact summary. Its response is an analytical contribution based on that supplied summary, not an independent audit of the repository. The following insights were retained after comparison with PALO artifacts and primary specifications.

Authority of statements and authority to act are separate. The four spine classes describe the status of framework information; an agent's authority depends on its principal, profile, delegation, policy and exact action context. A unified dictionary should give these axes different property names and scope notes. It should not infer an execution permit from a canonical-definition label.

Verification obligations deserve explicit closure. GLM proposed tracking executions whose required attestation has not arrived by a declared deadline. PALO already implements verification tasks, expiry windows, retries and `input_required` states; these mechanisms must be recognised before calling the capability missing. A useful extension is a documented completeness query and metric: executions requiring verification with neither a recorded attestation nor an explicitly assigned unresolved incident by the agreed deadline. This proposed measure needs a defined clock, denominator, exclusions, retry semantics and recovery protocol. An inconclusive attestation closes the obligation to record an assessment but does not resolve the underlying business uncertainty. [P24]

Evidence disagreement should remain visible. A future conflict record could identify the claims/evaluations in conflict, their scope and time, resolution authority and decision. This should complement the existing conservative data-fitness evaluator, not replace it with a rule that always prefers newer or signed evidence. Integrity, independence, relevance and truth are separate dimensions. PALO's current Evidence Evaluation already keeps four quality dimensions separate. [P21], [P23]

Cross-format integrity requires an explicit algorithm profile. The inspected core runtime recursively sorts JSON object keys for one canonicalisation function; arrays retain order. The separate evidence-envelope version 2 explicitly declares RFC8785, and `assurance-foundation.js` implements a JCS-labelled canonicaliser with finite-number and Unicode-surrogate checks for Ed25519 signing/verification. The release manifest instead hashes exact file bytes. These procedures have different scopes. Independent RFC conformance is not established by this review, and no RDF Dataset Canonicalization claim follows from deterministic JSON output. An exchange profile should name the precise path and test vectors, including number handling, Unicode and graph serialisation. [P24], [P28], [P33]

Several GLM suggestions were deliberately not adopted as facts: a linear ladder that places cryptographic anchoring above independent evaluation; an assumption that all 409 mappings are direct regulatory crosswalks; identity binding or temporal handling described as wholly absent; and categorical claims that SHACL cannot express policy-relevant constraints. The precise boundary is that validation constraints can support a policy process, while a shape evaluation alone does not grant operational authority or enforce external effects. GLM's older DPV/DCAT references were replaced with the primary publications checked for this dossier. [P09], [P17], [P24], [W03], [W06], [W07]

2.14 Publication provenance and maintenance

The semantic model, the editorial dossier, the runtime source and a hosted website are different release surfaces. This dossier retains the exact source inventory and verification results; the website exposes a separate publication inventory for the two detailed sections, technical catalogue, glossary and PDF. Updating the editorial material does not raise the maturity of the runtime or the draft stack ontology.

The source-registry reconciliation marks overdue entries [review-due](#) while retaining their original [checkedAt](#) and [nextReviewAt](#). Six entries require substantive review at this snapshot date. The semantic release manifest and Knowledge Reader manifest are updated together so that the metadata transition cannot silently break their integrity pins. This is a review-status transition, not a new validation of the referenced regulations or standards.

The publication process records source identity, artifact hashes, target and verification time. GitHub Pages and the separately hosted canonical domain require distinct deployment evidence. A sitemap date comes from the editorial modification record, while component release dates and source verification dates retain their own meaning. A [noindex](#) briefing page is deliberately excluded from the search sitemap; that does not exempt its content from release tracking. [P36]

Sources and accompanying catalogues

PALO artifact links resolve to public source locations where available. The research snapshot and validation records preserve the inspected bytes. Linked living pages may subsequently change.

- [P01] [PALO semantic foundation](#).
- [H01] Fabrizio Degni, *The PALO Framework A Paradigm for Principled AI Lifecycle Orchestration in Business*, December 2025, 67-page local founding document under [framework/](#). Core principles: pp. 13–15; five-phase lifecycle: pp. 18–22; evaluation dimensions: pp. 22–23. Page numbers refer to the PDF page sequence. Extract retained in the research record.
- [P02] [Semantic spine](#).
- [P03] [Lifecycle definitions](#).
- [P04] [Governance control packs](#).
- [P05] [Control library](#).
- [P06] [KPI and KRI registry](#).
- [P07] [Core ontology](#).
- [P08] [Core SHACL profile](#).
- [P09] [Semantic mappings](#).
- [P10] [Stack and policy impact graph and stack model](#).
- [P11] [PALO-AM operational model](#).
- [P12] [Full-cycle assurance](#).
- [P13] [Agentic capability matrix](#).
- [P14] [Gate Decision Record schema](#).
- [P15] [Agent Profile schema](#).
- [P16] [PALO-AM simulator source](#).
- [P17] [Action Claim schema](#).
- [P18] [ANS integration](#).
- [P19] [Swarm prototype, distributed extension and mandate schema](#).
- [P20] [Effect Contract, Execution Capability, Execution Receipt, Outcome Attestation, Policy Decision, Approval](#).
- [P21] [Data assurance control plane](#).
- [P22] [Evidence Claim schema](#).
- [P23] [Evidence Evaluation schema](#).

- [P24] Inspected local source packages/palo-mcp-server/core.js, preserved with digest in the research snapshot; public project [PALOframework](#). Local working-tree code is not assumed to equal the published Git HEAD.
- [P25] [Disclosure Contract](#), [Disclosure Observation](#), [Disclosure Receipt](#), [Data Fitness Decision](#).
- [P26] [Oversight crosswalk](#) and [bias crosswalk](#).
- [P27] [Evidence Artifact](#) and [Evidence Bundle Manifest](#).
- [P28] [Semantic Release Manifest](#) and pinned public release generator.
- [P29] [Semantic change-impact register](#).
- [P30] [Stack ontology](#), [stack SHACL](#), [stack JSON-LD context](#) and [core JSON-LD context](#).
- [P31] [Pinned public semantic validator](#).
- [P32] [Semantic invariants](#).
- [P33] [Evidence-envelope schema](#) and pinned assurance foundation implementation.
- [P34] [PolicyWatcher Signal Batch](#), [Operational Signal Registry](#) and [Browser-local Review Ledger](#). Acquired from Git commit 42d17508ddc836ae37076d9320cbbc672ea268cc.
- [P35] [Agentic editorial glossary and historical lifecycle correspondence](#), version 1.0.0. Separate from the canonical ontology and KPI/KRI registry.
- [P36] [Semantic publication inventory](#). This records the editorial release; deployment evidence is a separate receipt.
- [W01] [W3C SKOS vocabulary](#).
- [W02] [W3C OWL 2 Primer](#).
- [W03] [W3C SHACL Recommendation](#).
- [W04] [W3C PROV-O](#).
- [W05] [W3C ODRL Information Model](#).
- [W06] [Data Privacy Vocabulary](#). DPV 2.3 identifies itself as a Final Community Group Report and explicitly states that it is not a W3C Standard or on the W3C Standards Track.
- [W07] [W3C DCAT 3](#).
- [W08] [W3C JSON-LD 1.1](#).
- [C01] Accompanying catalogues/schema-index.json, field-dictionary.json, field-dictionary.csv and controlled-vocabularies.json.
- [C02] Accompanying catalogues/rdf-inventory.json and human-readable technical catalogue.
- [V01] Accompanying research/audit_semantics.py, validation-results.json, source manifests and SHACL reports. Verification date: 12 October 2026.