

Decision Provenance Standard™ v1.1 – Companion D: Diagrams

Status: Companion to the Decision Provenance Standard v1.1 (rev. 9). Explanatory, **non-normative**.

These figures are explanatory aids, not normative. The text of the Standard is the contract; every figure here *illustrates* a section of that text. What is shown in" / "illustrated by," never "as required by." Every state name, transition label, field name, enum value, signal name, and controlled-vocabulary term, including hyphens, and underscores.

Notation Key

The figure set uses one consistent visual language. Learn it once; it holds across all eight figures.

Notation	Meaning
Rounded rectangle	A state or a step
Double border	Terminal / irreversible state (e.g., Charter closed , record affirmed + sealed)
Solid arrow	A transition
Dashed arrow	A conditional / edge-case transition, or a one-way pointer
Accent (a single desaturated blue, plus heavier stroke)	The one load-bearing element of a figure – comprehension does not depend on color
Monospace text	A literal field name, enum value, or signal token (e.g., mode_declaration , disclosure)

Color is never the only signal. Every distinction is carried by shape, border style, stroke weight, or label, so the figures remain fully legible in grayscale.

List of Figures

Figure	Title
Figure 3-1	Charter Lifecycle State Machine
Figure 3-2	Mode Dispatch Grammar and the Embedded-Mode-2 Edge Case
Figure 4-1	Disclosure Block Flow
Figure 4-2	Mode-Drift Four-Layer Composed Mitigation
Figure 4-3	Emission-Cadence by Semantic Class
Figure 5-1	Decision-Record State Machine: Two Distinct State Families
Figure 6-1	Artifact-Set Relationship Map
Figure 7-1	The Three Conformance Levels: Cumulative Criteria

Figure 3-1 – Charter Lifecycle State Machine

```
stateDiagram-v2
    direction TB

    state "open" as open
    state "mode-declared" as mode_declared
```

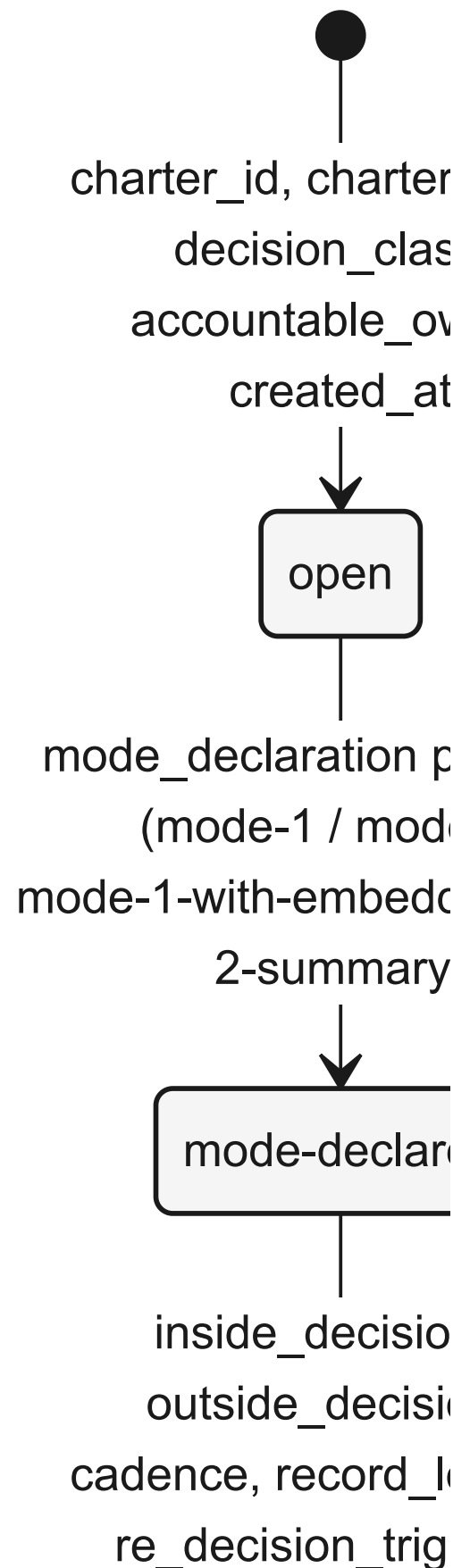
```

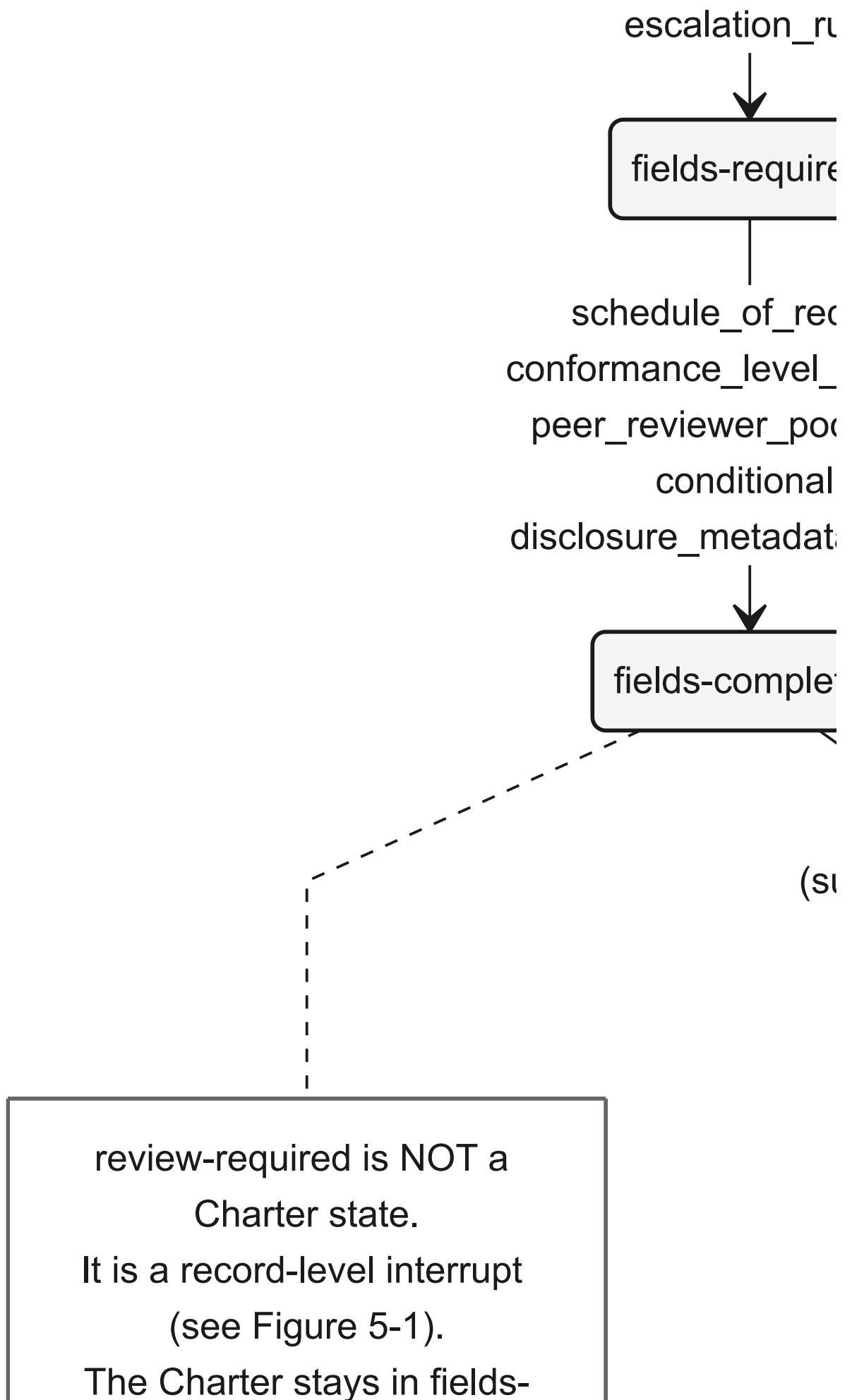
state "fields-required" as fields_required
state "fields-completed" as fields_completed
state "closed" as closed
state "new Charter v2 (prior_charter_ref)" as newCharter

[*] → open : charter_id, charter_name,<br/>decision_class, accountable_owner,<br/>created_at
open → mode_declared : mode_declaration populated<br/>(mode-1 / mode-2 /<br/>mode-1-with-embedded-mode-2-summary)
mode_declared → fields_required : inside_decisions, outside_decisions,<br/>cadence, record_location,<br/>re_decision_triggers, escalation.
fields_required → fields_completed : schedule_of_records,<br/>conformance_level_declared,<br/>peer_reviewer_pool (≥3),<br/>conditional d:
fields_completed → closed : closed_at set<br/>(subsumption / dissolution / retirement)
closed → [*]
closed → newCharter : reactivation =<br/>NEW Charter (dashed)

```

Figure 3-1 — Charter Lifecycle





completed while
individual records interrupt.



Charter lifecycle state machine: a forward-only sequence open, mode-declared, fields-required, fields-completed, closed, with a dashed off-ramp from closed to a new Chart

Figure 3-1 — Charter Lifecycle State Machine. A forward-only state machine in which each transition is gated by a named set of Charter fields; c

Full description. The Charter lifecycle is a forward-only state machine with five states: open, mode-declared, fields-required, fields-complete creation into open requires charter_id, charter_name, decision_class, accountable_owner, and created_at; open to mode-declared requires declared to fields-required requires inside_decisions, outside_decisions, cadence, record_location, re_decision_triggers (at least one o schedule_of_records, conformance_level_declared, a peer_reviewer_pool of at least three named reviewers, and a conditional disclosure_meta closed requires closed_at to be set on subsumption, dissolution, or explicit retirement. The closed state is terminal; a closed Charter cannot ret dashed off-ramp from closed to a separate "new Charter v2" node, not a back-edge into any prior state. The review-required interrupt is deliberat

Illustrates §3.3.

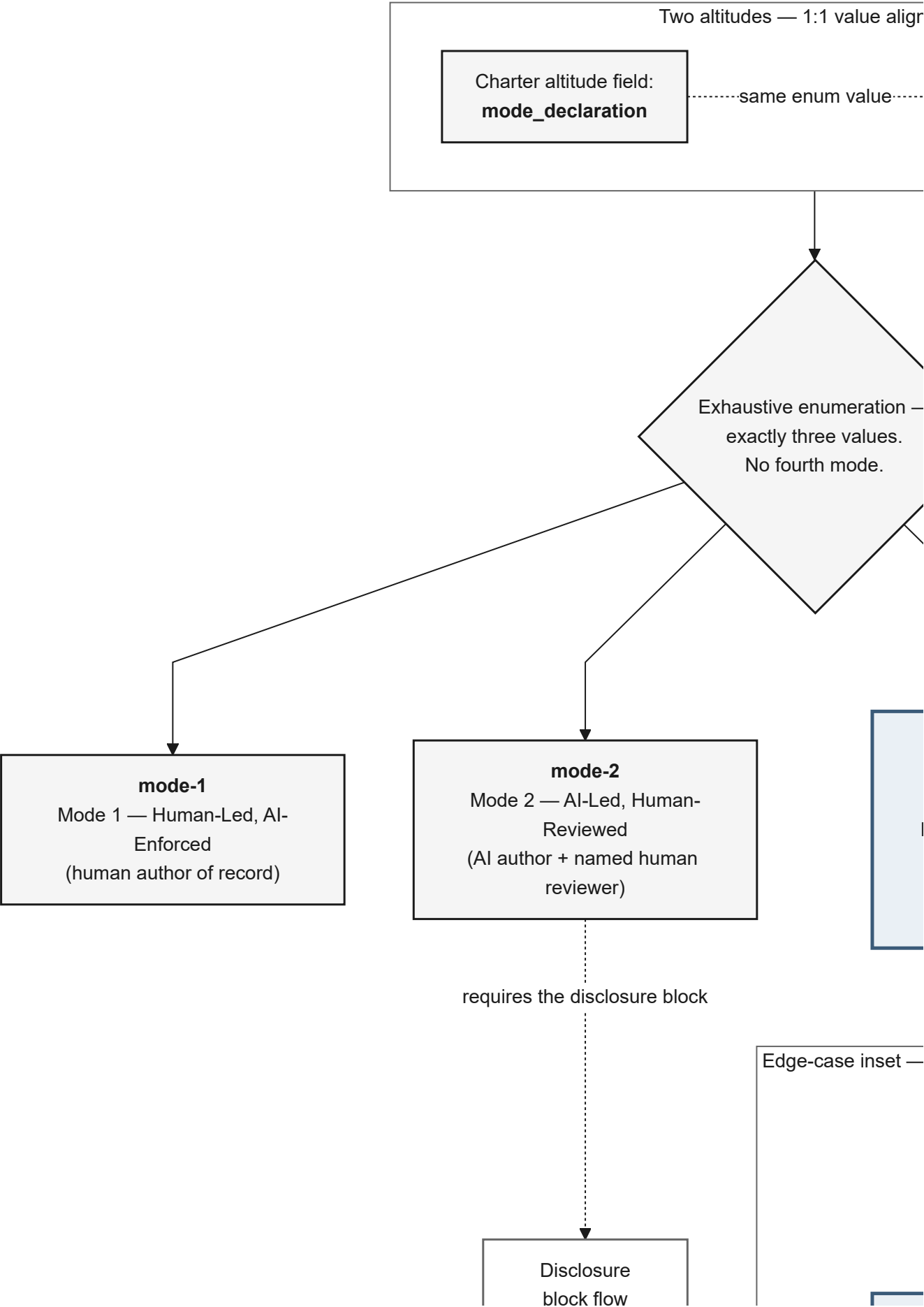
Figure 3-2 — Mode Dispatch Grammar and the Embedded-Mode-2 Edge Case

```
flowchart TB
    subgraph altitudes["Two altitudes - 1:1 value alignment"]
        direction LR
        charterField["Charter altitude field:<br/><b>mode_declaration</b>"]
        recordField["Record altitude field:<br/><b>dispatch_mode</b>"]
        charterField -. "same enum value" .-> recordField
    end
    altitudes --> enumGate["Exhaustive enumeration -<br/>exactly three values.<br/>No fourth mode."]
    enumGate --> m1["<b>mode-1</b><br/>Mode 1 - Human-Led, AI-Enforced"]
    enumGate --> m2["<b>mode-2</b><br/>Mode 2 - AI-Led, Human-Reviewed"]
    enumGate --> m1e["<b>mode-1-with-embedded-mode-2-summary</b><br/>human-authored container,<br/>AI-authored embedded sub-outputs"]
    m2 -. "requires the disclosure block" .-> a50["Disclosure block flow<br/>(see Figure 4-1)"]
    m1e --> edgeInset
    subgraph edgeInset["Edge-case inset - authorship is content-level, not container-level"]
        direction TB
        container["mode-1 container<br/>(human-authored)"]
        embedded["embedded AI-generated span<br/>carries its OWN disclosure block:<br/><b>mode_1_edge_case_flag = true</b>"]
        container --> embedded
    end
```

```
end
```

```
enumGate -. "no runtime fourth mode" .-> fourth["A fourth mode routes through the<br/>section 4.10 amendment process,<br/>NEVER a runtime w
```

Figure 3-2 — Mode Dispatch Grammar and the E



(see Figure 4-1)

Mode dispatch grammar: two aligned altitude fields `mode_declaration` and `dispatch_mode` feeding an exhaustive three-value enumeration, with an inset showing content-le

Figure 3-2 — Mode Dispatch Grammar and the Embedded-Mode-2 Edge Case. Mode is declared at two altitudes that align one-to-one, drawn from :

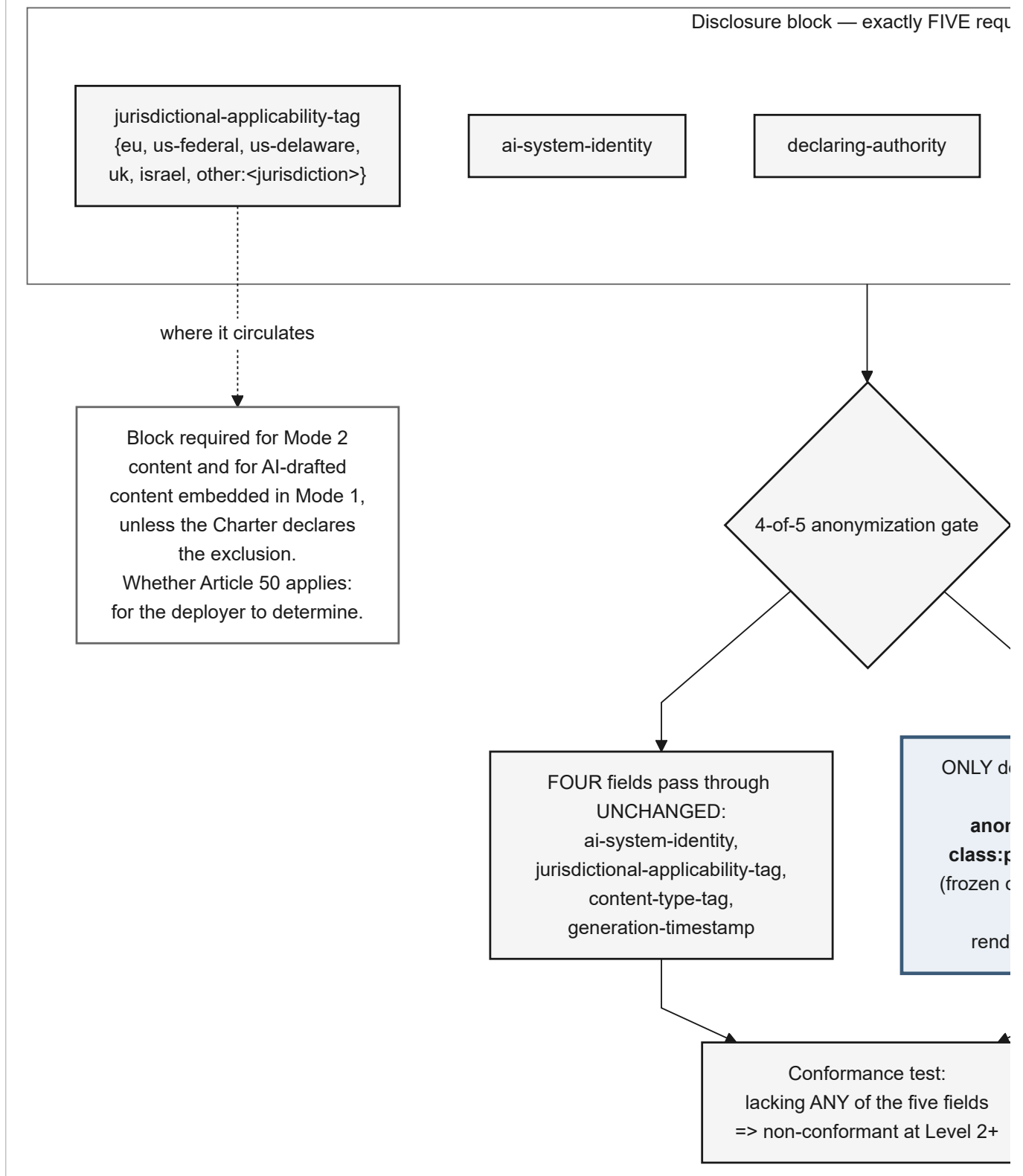
Full description. Mode is declared at two altitudes that align one-to-one: the Charter field `mode_declaration` and the decision-record field `dispatch_mode` (Mode 2 — AI-Led, Human-Reviewed), and `mode-1-with-embedded-mode-2-summary` (Mode 1 with the embedded-Mode-2-summary edge case). The `mode-1` container holds an AI-authored embedded span; authorship is judged at the content level, not the container level, so that no fourth mode exists; adding one routes through the §4.10 amendment process, never a runtime workaround. The three token families are distinct: `mode-1` / `mode-2` / `mode-1-with-embedded-mode-2-summary` are enum values.

Illustrates §3.4, §4.4, §4.7.

Figure 4-1 — Disclosure Block Flow

```
flowchart TB
    subgraph block["Disclosure block — exactly FIVE required fields"]
        direction TB
        f1["declaring-authority"]
        f2["ai-system-identity"]
        f3["jurisdictional-applicability-tag<br/>{eu, us-federal, us-delaware, uk, israel, other:&lt;jurisdiction&gt;}"]
        f4["content-type-tag<br/>{decision-summary, recommendation, decision-aid,<br/>draft, classification, synthetic-media, other:&lt;type&gt;}"]
        f5["generation-timestamp"]
    end
    block --> anon["4-of-5 anonymization gate"]
    anon --> pass["FOUR fields pass through UNCHANGED:<br/>ai-system-identity, jurisdictional-applicability-tag,<br/>content-type-tag, generation-timestamp"]
    anon --> transform["ONLY declaring-authority may transform to:<br/><b>anonymized-deployer-class:product-organization</b>"]
    pass --> test["Conformance test: lacking ANY of the five fields<br/>≥&gt; non-conformant at Level 2+"]
    transform --> test
```

Figure 4-1 — Disclosure Block



Disclosure block flow: five required fields feeding a four-of-five anonymization gate where four fields pass unchanged and only declaring-authority may transform to the value

Figure 4-1 — Disclosure Block Flow. The disclosure block has exactly five required fields; under the 4-of-5 anonymization rule, four survive unchanged.

Full description. The disclosure block has exactly five required fields: `declaring-authority`, `ai-system-identity`, `jurisdictional-applicability-tag`, `jurisdictional-applicability-tag` takes a value from {eu, us-federal, us-delaware, uk, israel, other:<jurisdiction>}, recording where the draft, classification, synthetic-media, other:<type>}. Under the four-of-five anonymization rule, four fields survive anonymization unchanged.

organization . That value is a frozen controlled-vocabulary enum token and is rendered byte-identical; it contains the substring "product-organizat
note states that the block is required for Mode 2 content and for AI-drafted content embedded in Mode 1, unless the Charter declares the exclusion,
Illustrates §4.6.2, §4.6.3.

Figure 4-2 – Mode-Drift Four-Layer Composed Mitigation

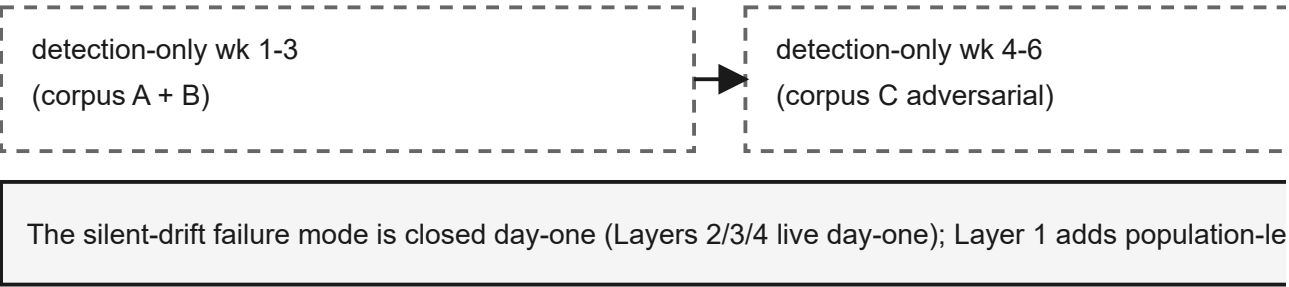
Figure 4-2 — Mode-Drift Four-Layer Composed Mitigation

Structural answer to silent Mode 1 to Mode 2 drift. Orthogonal composition: no two layers share

Layer	Actor (independence axis)	Detection moment (independence axis)
Layer 1 — statistical detection	independent classifier (disjoint corpus)	post-close population sampling
Layer 2 — in-flow audit hook	original declaring authority	record-close hard gate
Layer 3 — Mode-Confirmation Audit primitive	designated peer reviewer peer_reviewer_pool (≥3 named; not original author)	review-required interrupt
Layer 4 — Named Human-Attestation Fallback (double border = binding)	named attestor	record-close binding

Two composition axes: actor-independence and detection-moment-independence (no two layers share)
Safety property: combined false-negative bounded by the product of layer rates; false-positive bounded

Phased Layer 1 rollout



Explanatory figure (non-normative). The text is the contract; this figure illustrates it.

Mode-drift mitigation as a four-row matrix of orthogonal layers by Layer, Actor, Detection moment, and Coverage, plus a phased Layer 1 rollout timeline strip.

Figure 4-2 — Mode-Drift Four-Layer Composed Mitigation. Four orthogonal mitigation layers — no two sharing an actor or a detection moment —

Full description. The Mode-Drift mitigation composes four orthogonal layers, no two sharing an actor or a detection moment. Layer 1, statistical de coverage is a 15% baseline, up to 100% on the first hundred records, and 30% on embedded-summary Charters, and it routes drift candidates above authority as a record-close hard gate, covering 100 percent of Mode-1 records through a four-question Substantive-Authorship Challenge (Q1–Q4); Y 3 and blocks closed . Layer 3, the Mode-Confirmation Audit primitive, is run by a designated peer reviewer drawn from the peer_reviewer_pool (ε per \$6.4 and is the only layer with authority to change Mode classification. Layer 4, the Named Human-Attestation Fallback, is bound at record close composition axes are actor-independence and detection-moment-independence; the combined false-negative is bounded by the product of layer ra 3 (corpus A + B), detection-only in weeks 4–6 (corpus C adversarial), and enforcement from week 7 onward with full firing authority for no_silent_r day one; Layer 1 adds population-level signal from week 7.

Illustrates §4.8.1.

Figure 4-3 — Emission-Cadence by Semantic Class

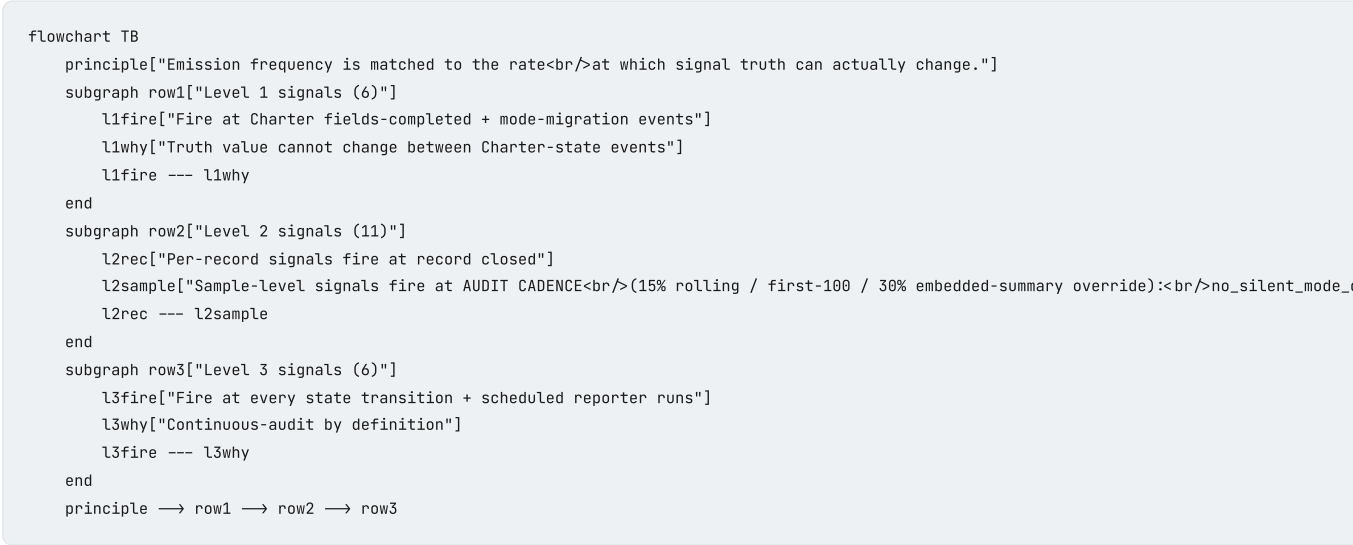
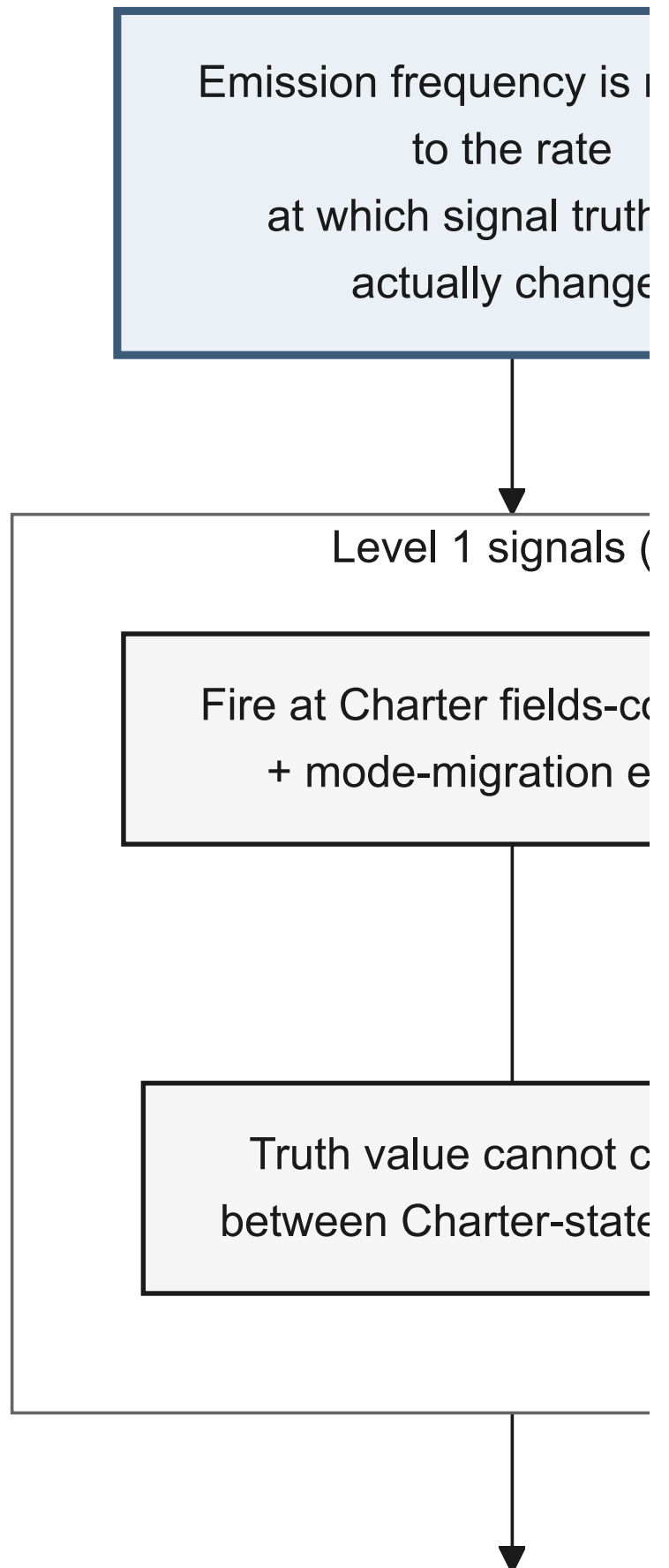


Figure 4-3 — Emission-Cadence



Level 2 signals (

Per-record signals fire at
closed

Sample-level signals fire at AL
(15% rolling / first-100 / 30%
summary override
no_silent_mode_drift_in
no_passive_promotion_to_affir
redaction-attestation, altitu

Level 3 signals (

Fire at every state tra
+ scheduled reporte

Continuous-audit by d

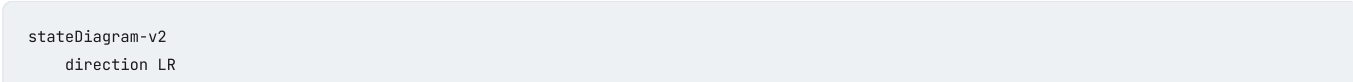
Emission-cadence as a three-row temporal strip: Level 1 signals fire at Charter-state events, Level 2 signals split into per-record and audit-cadence firing, Level 3 signals fire c

Figure 4-3 — Emission-Cadence by Semantic Class. Signal emission frequency is matched to the rate at which each signal's truth can actually chan

Full description. Conformance signals fire on a cadence matched to how fast their truth can change. The six Level 1 signals fire at Charter `fields-i`. The eleven Level 2 signals split into per-record signals that fire at record `closed` and sample-level signals — `no_silent_mode_drift_in_sample`, `no_` cadence of 15% rolling with first-100 and 30%-embedded-summary overrides. The six Level 3 signals fire at every state transition and on scheduled Level 3.

Illustrates §4.8.2.

Figure 5-1 — Decision-Record State Machine: Two Distinct State Families



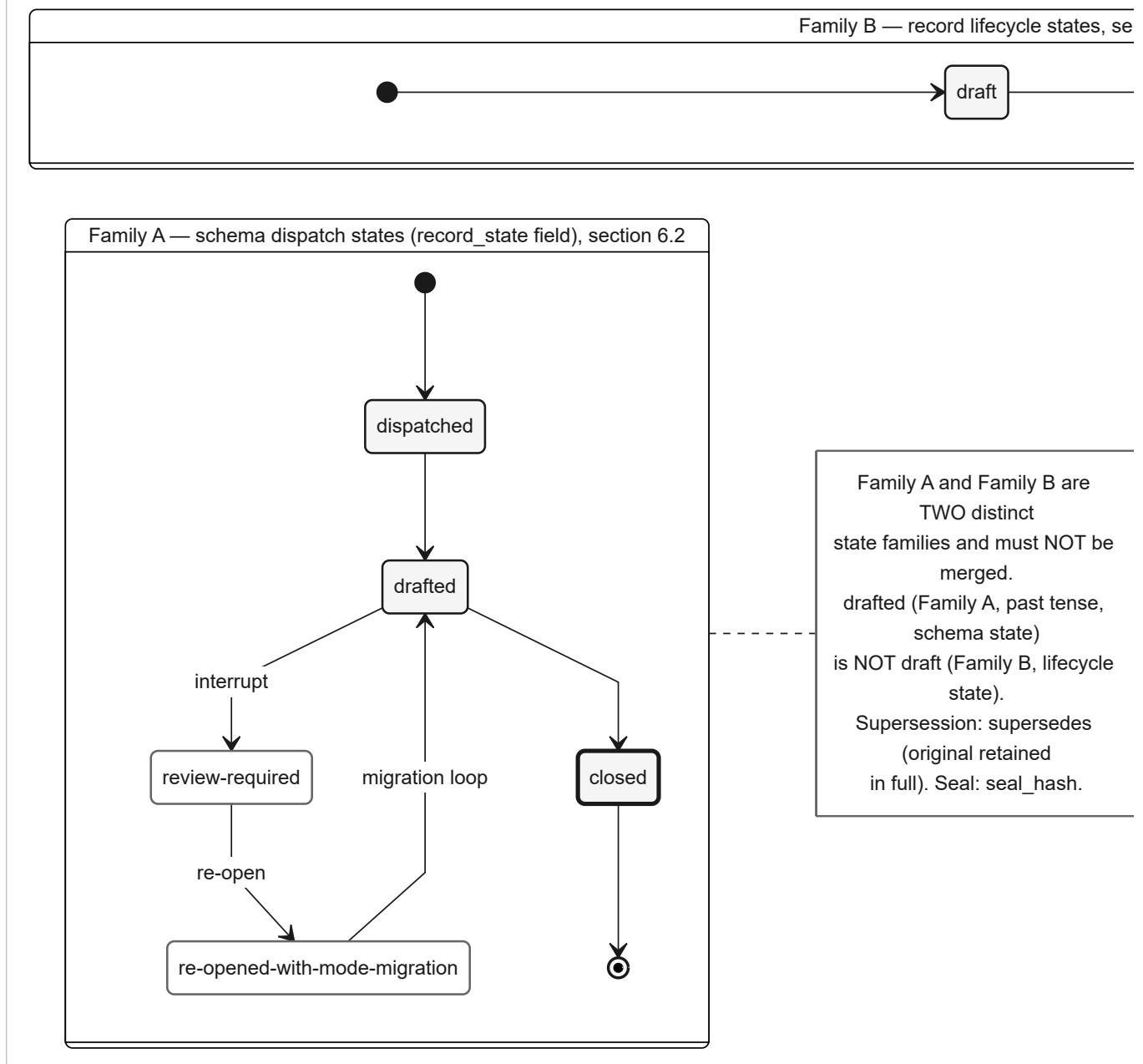
```

state "Family A – schema dispatch states (record_state field), section 6.2" as FamilyA {
    direction TB
    state "dispatched" as dispatched
    state "drafted" as drafted
    state "closed" as closedA
    state "review-required" as reviewRequired
    state "re-opened-with-mode-migration" as reopened
    [*] → dispatched
    dispatched → drafted
    drafted → closedA
    drafted → reviewRequired : interrupt
    reviewRequired → reopened : re-open
    reopened → drafted : migration loop
    closedA → [*]
}

state "Family B – record lifecycle states, section 5.1" as FamilyB {
    direction TB
    state "draft" as draftB
    state "reviewed" as reviewed
    state "affirmed" as affirmed
    [*] → draftB
    draftB → reviewed
    reviewed → affirmed
    affirmed → [*]
}

```

Figure 5-1 — Decision-Record State Machine: T



Decision-record state machine drawn as two visually separate lanes: Family A schema dispatch states dispatched, drafted, closed with a review-required interrupt and re-open

Figure 5-1 – Decision-Record State Machine: Two Distinct State Families. A decision record carries two deliberately distinct state families — schemata

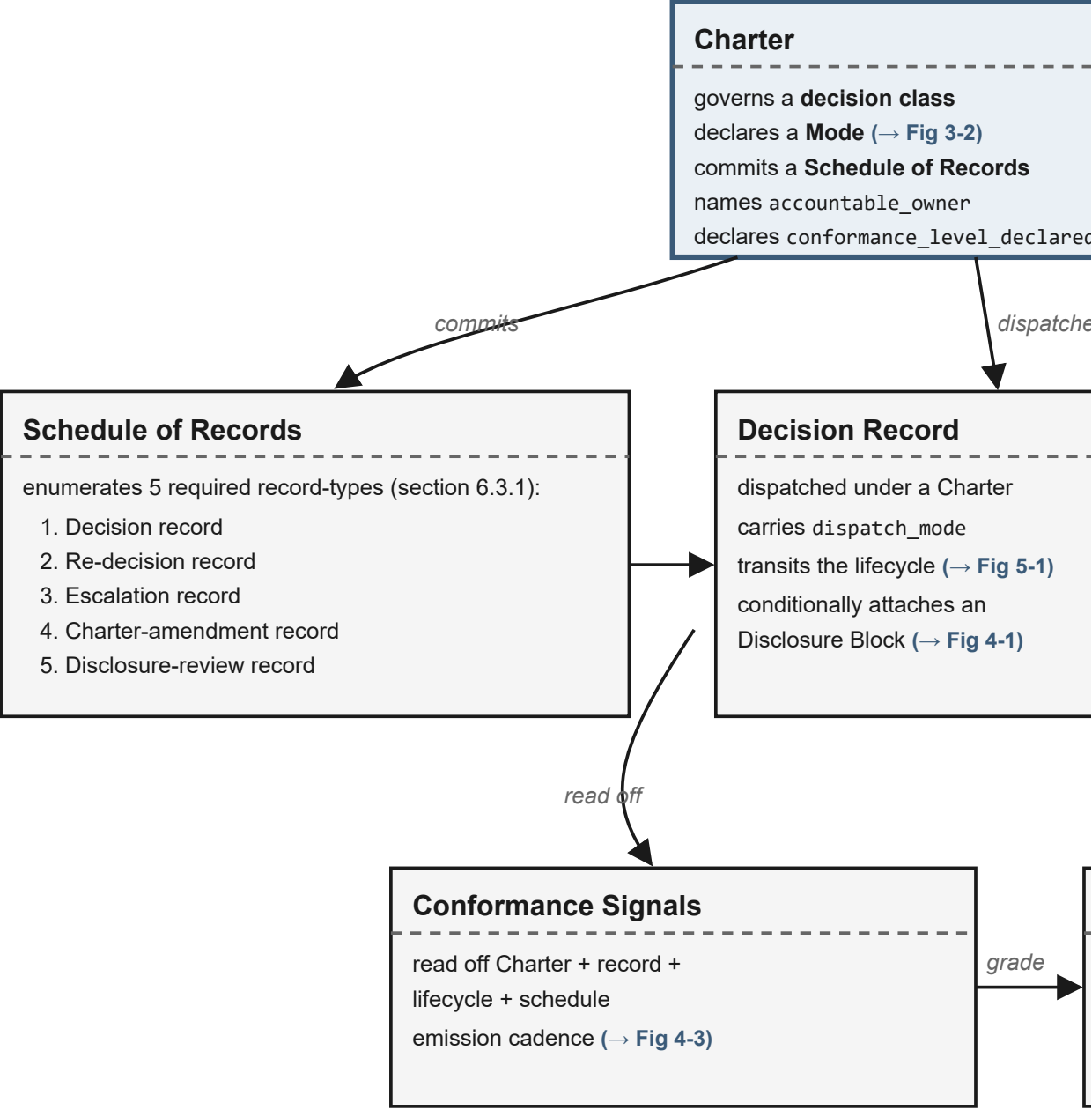
Full description. A decision record carries two deliberately distinct state families that must not be conflated, drawn as two separate lanes. Family A with a `review-required` interrupt branching off `drafted` and a `re-opened-with-mode-migration` path that loops from `review-required` back to `drafted`. Family B is the record lifecycle, rendered as a separate track: `draft`, then `reviewed`, then `affirmed`. Family B is not a separate enum field — `affirmed` simultaneously. The past-tense schema state `drafted` (Family A) is distinct from the lifecycle state `draft` (Family B); conflating them is a conformance error.

Illustrates §5.1, §6.2.

Figure 6-1 – Artifact-Set Relationship Map

Figure 6-1 — Artifact-Set Relationship Map

One-picture orientation: how the standard's parts compose before the normative prose. Arrows in p



Explanatory figure (non-normative). The standard points one-way to the deploying organization's m

Artifact-set relationship map: a Charter governs a decision class, declares a Mode, commits a Schedule of Records, names `accountable_owner`, and declares `conformance_level_declared`; Conformance Signals grade Conformance Levels; a one-way dashed pointer leads to the deploying organization's methodology.

Figure 6-1 — Artifact-Set Relationship Map. A one-picture orientation showing how the standard's artifacts compose before the reader reaches the **Full description**. This orientation map shows how the standard's artifacts compose. A Charter governs a decision class, declares a Mode (Figure 3-2). The Schedule of Records enumerates five required record types: Decision record, Re-decision record, Escalation record, Charter-amendment record, and Disclosure-review record. The Decision Record carries `dispatch_mode`, transits the record lifecycle (Figure 5-1), and conditionally attaches a Disclosure Block (Figure 4-1) on Mode 2 and Mode-1 edge-cases. Conformance Signals grade Conformance Levels (Figure 7-1). The standard points one-way, via a dashed neutral pointer, to the deploying organization's methodology; it does not

Figure 7-1 – The Three Conformance Levels: Cumulative Criteria

```
flowchart TB
  subgraph L3["Conformance Level 3 – Continuously Auditable (section 7.4.1) – subsumes Level 2"]
    l3a["1. Re-decision triggers fire and produce records on the declared cadence"]
    l3b["2. Escalation rule produces records when invoked"]
    l3c["3. Disclosure metadata blocks carry current last_reviewed_at"]
    l3d["4. Schedule of records is queryable and exportable on demand"]
  end
  subgraph L2["Conformance Level 2 – Mode-Disambiguated (section 7.3.1) – subsumes Level 1"]
    l2a["1. Every decision record carries its dispatch_mode field"]
    l2b["2. Every Mode 2 record carries a complete disclosure block (FIVE fields – see Figure 4-1)"]
    l2c["3. Every Mode 1 edge-case record carries a per-record disclosure block (reads mode_1_edge_case_flag)"]
    l2d["4. Schedule sample audit returns no silent drift (no_silent_mode_drift_in_sample)"]
  end
  subgraph L1["Conformance Level 1 – Charter-Conformant (section 7.2.1)"]
    l1a["1. Charter has reached the fields-completed lifecycle state"]
    l1b["2. mode_declaration populated with one of three values"]
    l1c["3. Schedule of records committed and non-empty"]
    l1d["4. record_location resolves to a durable surface"]
    l1e["5. accountable_owner names one human"]
    l1f["6. re_decision_triggers meets two-class minimum"]
    l1g["7. escalation_rule populated with a named, exact trigger"]
  end
  end
end
L3 → rule["Cumulative rule: a Charter that does not grade at Level 1 cannot grade at Level 2 or 3."]
rule → footer["self-declared; the Steward does not certify, grade, or audit."]
```

Figure 7-1 — The Three Conformance Lev

Conformance Level 3 — Continuously Auditable (section 7.3)

1. Re-decision triggers fire and produce records on the Charter's declared cadence

2. Escalation rule produces records when invoked

3. Disclosure metadata blocks carry current last_reviewed_at per the review cadence

Conformance Level 2 — Model

Three conformance levels drawn as nested bands: Level 1 Charter-Conformant with seven criteria innermost, Level 2 Mode-Disambiguated with four criteria, Level 3 Continuo

Figure 7-1 — The Three Conformance Levels: Cumulative Criteria. Three cumulative conformance levels drawn as nested bands; all are self-declai

Full description. The three conformance levels are cumulative, drawn as nested bands with Level 1 innermost. Level 1, Charter-Conformant (§7.2.1 is populated with one of the three enumerated values (`mode-1` , `mode-2` , `mode-1-with-embedded-mode-2-summary`); the schedule of records is commi `re_decision_triggers` meets the two-class minimum; and `escalation_rule` is populated with a named, exact trigger. Level 2, Mode-Disambiguate `dispatch_mode` field; every Mode 2 record carries a complete disclosure block of five fields (Figure 4-1); every Mode 1 edge-case record carries a per sample audit returns no silent drift findings (`no_silent_mode_drift_in_sample`). Level 3, Continuously Auditable (§7.4.1), subsumes Level 2 and add produces records when invoked; disclosure metadata blocks carry a current `last_reviewed_at` per the review cadence; and the schedule of record: cannot grade at Level 2 or 3. All levels are self-declared; the Steward does not certify, grade, or audit.

Illustrates §7.2, §7.3, §7.4.

End of Companion D.

The records are input, not evidence. The Standard informs frameworks without satisfying them. Conformance is self-declared; no body certifies it. It is not legal advice and Decision Provenance Standard™ v1.1 — Reading Edition (rev. 9). Open standard under CC-BY 4.0. Not a certified product. Founding Steward: Yohay Etsion; institutional Steward: Ets

Conformance Level 1 — Charter-Conforma

1. Charter has reached the fields-completed lifecycle sta
2. `mode_declaration` populated with one of three values (`mode-1`, `mode-2`, `mode-1-wi embedded-mode-2-summar`
3. Schedule of records committed and non-empty

4. Schedule of records is queryable and exportable for counsel and auditor on demand

4. record_location resolves to durable surface

5. accountable_owner name one human

6. re_decision_triggers meet two-class minimum

7. escalation_rule populated with a named, exact trigger

Cumulative rule: a Charter that does not grade at Level 1 cannot grade a

Level 2 or 3.



self-declared; the Steward
does not certify, grade, or au

