

Companion to the Decision Provenance Standard v1.2; tracks core revision rev. 10.

This Companion provides the Standard's own operational guidance for installing the Standard: the install sequence, the Charter-authoring role pattern, the common implementation pitfalls, and the compressed pattern for smaller organizations. It is self-contained: a reader with only the Standard and its companions can install the Standard end-to-end. This Companion's cross-references to the Standard's core sections (§1–§7, §10.7, §11) resolve against the core Reading Edition (rev. 10); its references to Companion A and Appendix G resolve against those documents.

Note on the install-≠-compliance non-claim. The §10.7 "Explicit non-claim" (install does not constitute compliance) is **retained in the core Standard**, not in this Companion. The §C.7 anchor is reserved and unused; references to the install-≠-compliance non-claim resolve to core §10.7.

Companion C – Implementation Guidance

 **Use of the Standard.** See top of the core Standard for the full notice and the Jurisdiction Assumed declaration.

C.1 Purpose

This Companion provides operational guidance for an organization installing the Decision Provenance Standard™. It describes the sequencing of Charter authoring, the role engagement pattern for Charter authoring and decision-record maintenance, the common implementation pitfalls a deployer should anticipate, the compressed install pattern available to organizations earlier than Series C, the operational pre-conditions an installation depends on, and the explicit non-claim that installation does not constitute compliance with any regulatory framework (retained in the core Standard at §10.7).

Companion C is install guidance, not a compliance program. The Standard structures how a Charter is authored, how decision records are maintained, how records progress through the §5 lifecycle to **affirmed** -and-sealed status, and how the schedule of records becomes findable.

Whether a deployer's installation conforms to the Standard at conformance level 1, 2, or 3 is a structural reading per Section 7. Whether the deployer's organization is compliant with any external regulatory framework is for the deployer to determine, against its actual implementation, not against this Companion. The two questions are distinct, and Companion C commits only to the first.

A deployer reading Companion C is reading the operational arc by which the structural requirements of Sections 2 through 7 and Companion A (Regulatory Cross-References) land inside their organization. The arc has a sequence. The sequence has roles. The roles have failure modes. Companion C names them.

C.2 Sequencing of Charter authoring

C.2.1 The 90-day install sequence

The Standard defines a 90-day install sequence. The sequence is the operational spine of the install: six moves across three windows. It is run by a newly-seated accountable leader (typically a CPO, VP Product, or Director of Product Operations) who has read Sections 2 through 7 of the Standard and has the Charter template open. The windows below are the deployer's operating manual; the operational detail (per-window tactics, common stumbles) is described inline.

Week One — Read the landscape before replacing it (Move 1). The deployer changes nothing in the first week. They map what is already installed, memorized as ritual, and missing, across existing cadences, decision records, the commitment register, and in-flight bets. They close Week One with a one-page state-of-the-decision-system note shared with their CEO and the standing leadership forum. The note is a read, not a plan. No Charter authored in Week One.

Days 1–30 — Author one Charter and seat the standing forum (Moves 2 and 3). The deployer authors one Charter on one decision. Not two. Not four. One. The first-Charter candidates are launch readiness, pricing exceptions, and platform intake. The Charter is filled out per the Standard's Section 3 (Charter Mechanism) and Section 6 (Required Artifact Set) field requirements: every field populated, the `mode_declaration` field set to one of the three enum values (Mode 1 — Human-Led, AI-Enforced; Mode 2 — AI-Led, Human-Reviewed; or Mode 1 with the embedded-Mode-2-summary edge case), the schedule of records committed, the accountable owner named as a single human, and at minimum one outcome-evidence and one market-evidence re-decision trigger written down. In the same window, the deployer seats the standing leadership forum (seven default seats: Product Management, Product Marketing, Business Development, Competitive Intelligence, Business Operations, Product Operations, and Customer

Success with Value Realization). Each seat carries a director-altitude owner who delivers gating inputs as evidence, not as narrated slides. The Charter is taken to the forum as a commit, not a proposal.

Days 30–60 — Install the cadence and the CPO-counterparty interfaces (Moves 4 and 6). The deployer installs three cadences in order: the quarterly Portfolio Review (90 minutes; Business Operations pre-read and Competitive Intelligence market read as gating inputs), the monthly Product-line Review, and the Launch Readiness Review on a T-6 / T-14 rhythm for material launches. In parallel, the deployer names the four CPO-counterparty interfaces at altitude: CPO to CTO on the platform envelope, CPO to CRO on commercial motion mix, CPO to CFO on the capital envelope, CPO to CMO on the brand-calendar boundary. Each interface is recorded at the altitude it decides; an unnamed interface narrates around itself by ad-hoc escalation. The five operating-layer observability signals (decision latency, commitment drift, cadence adherence, charter decay, re-decision integrity) are wired here; Product Operations operates them, the CPO consumes them at Portfolio Review.

Days 60–90 — Convert one Decision to a Commitment and run the first re-decision (Move 5 plus first trigger fire). The deployer hardens one Decision from the first Charter into a Commitment by attaching a resource: a hire released, funding moved, sequencing locked, or an external promise made. A roadmap item that triggered hiring is a Commitment; a slide in a deck is not. The deployer expands the Charter footprint to two or three additional in-flight bets — typically those with the most ambiguous ownership, the most-promised outcomes, or the largest capital commitment. Somewhere in this window, the first re-decision trigger should fire on an existing bet. The deployer runs that re-decision as a named event against the documented trigger, with the required inputs in the named forum. The first clean re-decision is where the cadence installed in Days 30–60 earns its license. By the end of Quarter 1 the Portfolio Review runs for the first time under full discipline.

C.2.2 What this sequence produces — and what it does not

A Charter authored against the Standard's Section 3 field requirements, reaching Section 7's `fields-completed` lifecycle state, with its schedule of records committed and its mode declared, **produces a Standard-conformant Charter at conformance level [N], where N is the level Section 7 grades against the deployer's structural completeness.** The Standard's conformance levels (1, 2, 3) describe structural conformance. The grade is a structural fact: it records that fields are populated, that signals are present, and that the schedule of records is queryable per Section 6.

The 90-day sequence does **not** produce a "compliant Charter." Compliance with any external framework — SOX 404, EU AI Act, GDPR, HIPAA, NIST AI RMF, ISO/IEC 42001, or any other regulatory or standards corpus — is for the deployer to determine against its actual

implementation. The Standard structures the install; the determination is not the Standard's. A deployer who reads Companion C as a path to compliance is reading the wrong section. Companion A (Regulatory Cross-References) names the frameworks the Standard is designed to be useful to as input; counsel and auditors convert the audit-ready decision provenance the Standard structures into whatever evidence the deployer determines those frameworks call for.

This distinction is load-bearing. **The Standard structures the install; whether the install complies with any framework is for the deployer to determine.** Companion C commits to the first half of that sentence and explicitly does not commit to the second.

C.2.3 Migration narrative – Mode 1 to Mode 2

When a deployer installs the Standard, the first Charters are typically authored as Mode 1 — Human-Led, AI-Enforced. The human author writes the decision; the AI worker checks the human's work against the Charter's required fields. As the deployer's organization grows confident in the discipline of the Standard's mode-declaration mechanism (Section 3) and the disclosure-block schema (Section 4), some Charters may be re-declared to Mode 2 — AI-Led, Human-Reviewed, where the AI worker authors the decision and a named human reviewer reviews before action.

The Standard does not prescribe when this migration should happen. The migration narrative the Standard adopts is the following:

Most enterprises start at Mode 1 and migrate to Mode 2 on their own clock; the Standard does not prescribe the pace.

A deployer migrating a Charter from Mode 1 to Mode 2 amends the Charter's `mode_declaration` field through a named decision that is itself recorded under the Charter (Section 4 governs the mode-migration decision-record requirements; the seam between the Charter state model and the dispatch state machine is enumerated in Section 4). The migration is not a runtime override; it is a Charter amendment that produces its own decision record. A deployer that tries to migrate a Charter without amending the Charter has not migrated the Charter — they have produced a silent Mode 1 → Mode 2 drift, which is the dispatch state machine's most exposed failure mode (see §C.4 below).

C.2.4 Compressed pattern – sub-Series-C organizations

A deployer whose organization is earlier than Series C — pre-product-market fit, Series A, or early Series B — typically lacks the seats to run the full 90-day sequence. The Standard supports this

stage through a compressed install pattern (described below).

The compressed pattern is not a different Standard. It is the same Standard, run on a smaller surface. Three compressions apply:

1. **Charter shape.** The compressed Charter is the same artifact as the Standard's Section 3 Charter, with the required-inputs list collapsed from three tiers (core / empowerment / execution) to a single list of three to five named individuals. Mode declaration is still required; the schedule of records is still committed; one outcome-evidence and one market-evidence re-decision trigger are still written down.
2. **Standing forum.** The seven-seat standing leadership forum is replaced by whatever standing leadership forum the company already has — typically the founder, the head of product, and one or two adjacent leaders. The forum is the authoring and reviewing surface for the Charter; the full seven-seat forum is the surface a sub-Series-C deployer grows into, not a precondition for installing the Standard.
3. **Install order.** The install order runs in three moves rather than six. Move 1: pick one decision the team is making badly (launch readiness is the most common right first install) and author the compressed Charter for it. Move 2: run the next instance of that decision against the Charter as a commit, not a proposal. Move 3: when the Charter has run twice cleanly, author the second Charter. The expansion to seven seats and the full six-move sequence waits until the company has the seats to hold it.

The compressed pattern produces a **Standard-conformant Charter at conformance level [N]** at the same structural altitudes Section 7 grades against — the level grade does not depend on company stage, only on structural completeness. A clean compressed Charter at level 1 is a more honest conformance state than a half-installed full Charter at level 0. **The mistake at every sub-Series-C stage is the same: installing the decision system the blueprint describes before the company has the seats to run it produces decoration, not architecture.**

The deferred practices at sub-Series-C scale (the full seven-seat leadership forum; per-function Charters across the Section 4 set; the five-signal observability layer for the operating system itself; the sensor-to-decision-compulsion protocol as a standalone discipline) are revisited at the stage that earns the right to revisit each, as the company's seats fill.

C.3 Role engagement pattern

The Standard distributes responsibility across three named roles. The roles are operational; they are not job titles. A deployer's organization may staff each role differently, but the role itself is binding: every Charter has each of the three roles named.

C.3.1 The accountable owner (Charter-level)

A single named human is the accountable owner of each Charter, per Section 3's `accountable_owner` field requirement. The accountable owner authors the Charter, holds the Charter's `mode_declaration` decision, signs off on the Charter's transition through its lifecycle states (`open` → `mode-declared` → `fields-required` → `fields-completed` → `closed`), and is the named human counsel and auditors meet with when reviewing the Charter. The accountable owner is one and only one person. A Charter without an accountable owner is structurally incomplete and cannot reach `fields-completed` ; a conformance-level reporter will not grade it past level 0.

In practice, the accountable owner is most often a director-altitude or VP-altitude product leader. For first-Charter installs the accountable owner is frequently the deployer themselves. For Charters authored later in the install arc, the accountable owner is the seat whose decision class the Charter governs (the Product Marketing director for the positioning Charter, the Product Operations director for the launch readiness Charter, and so on; Companion B carries worked examples across these functional surfaces).

C.3.2 The declaring authority (mode-declaration and disclosure block)

The declaring authority for any Charter declared as Mode 2 — AI-Led, Human-Reviewed, or as Mode 1 with the embedded-Mode-2-summary edge case (per Section 4 §4.6) is the person who prepares the disclosure block for AI-authored outputs delivered under the Charter, named together with the organization they act for (Section 4 §4.6.2). The declaring authority is named in the disclosure metadata block; it is not the Charter and not the AI system. Whether Article 50 of the EU AI Act applies to those outputs, and who carries any legal duty, is for the deployer to determine.

The accountable owner and the declaring authority **may** be the same person, but they are not required to be. A deployer authoring a Mode 2 Charter where the accountable owner sits on the operating side of the organization and the declaring authority sits on the regulatory or privacy side has named two people on the Charter, both human, both individually accountable for distinct aspects of the Charter's operation.

C.3.3 The schedule-of-records maintainer (Section 6 surface)

A single named role — typically Product Operations — maintains the Charter's schedule of records per Section 6. The maintainer is responsible for the schedule's findability (the Standard's "reconstructable in 30 seconds" findability rule), for the schedule's queryability (Section 6's export commitments), for the conformance-level reporter's access to the schedule (Section 7 grading depends on it), and for the schedule's integrity over time (no silent record removal; no record renaming without back-pointers). The maintainer is the operational counterpart to the accountable owner: the accountable owner authors decisions; the maintainer ensures the records of those decisions remain findable and exportable for counsel and auditor review.

At sub-Series-C scale the schedule-of-records maintainer is often the same person as the accountable owner. At Series C and beyond the role separates, typically into a Product Operations seat. The role does not require a Product Operations title; it requires named accountability for the schedule of records.

C.3.4 Engagement cadence

The three roles engage on the cadence the Charter declares. The accountable owner runs the Charter's standing forum on the declared cadence (weekly, monthly, quarterly, or on-trigger). The declaring authority reviews and reaffirms the disclosure block on the cadence the Charter declares for disclosure review (recorded as a `last_reviewed_at` date or a disclosure-review record; §7.4.1). The schedule-of-records maintainer runs the conformance-level reporter on the cadence the Charter declares for conformance reporting. The three cadences may be the same or different; the Charter declares each one.

C.4 Common implementation pitfalls

Five pitfalls catch first-time deployers (§C.4.1–§C.4.5); three further integrity and edge-case pitfalls (§C.4.6–§C.4.8) surface as an installation matures. A deployer who reads Companion C before installing the Standard is reading these pitfalls in advance; a deployer who reads them only after experiencing one is reading them too late.

C.4.1 Install-everywhere-at-once

The deployer authors Charters on launch readiness, platform intake, pricing exceptions, and portfolio sequencing simultaneously in the first sixty days. Organizational attention runs out by Day 45. The deployer ends the quarter with four half-installed Charters, none of which has run a clean re-decision.

The structural mitigation is the sequencing rule in §C.2: one Charter in the first thirty days, three by Day 90. A deployer tempted to install more is tempted to install decoration.

C.4.2 Mode 1 / Mode 2 confusion at the Charter level

The deployer authors a Charter without populating the `mode_declaration` field, or populates it ambiguously. The Charter cannot transit out of `open`. If the deployer overrides the lifecycle gate and dispatches decisions under an unmoded Charter, every decision record produced under it is structurally non-conformant — and the first decision that incorporates an AI-authored sub-output carries no disclosure metadata block.

The structural mitigation is the Charter state model itself (Section 3): the `mode_declaration` field is required at any state past `open`, the field is enumerated (not free-text), and the dispatch state machine of a tool that implements the Standard refuses to dispatch decisions under an unmoded Charter. The deployer's task is to declare the mode honestly. Mode 1 is the default. Mode 2 is a deliberate choice with disclosure consequences.

C.4.3 Silent Mode 1 → Mode 2 drift

A Charter is declared as Mode 1. The human author of decisions under the Charter leans on AI worker output for options framing, draft language, or recommendation phrasing. The resulting decision record is materially Mode 2 — AI-authored substantive content — but bears Mode 1 metadata. The Charter's `mode_declaration` is unchanged. The disclosure metadata block does not attach. The schedule of records carries the decision as Mode 1. The disclosure the Standard requires is silently skipped.

This is the dispatch state machine's most exposed failure mode (Section 4). The structural mitigation is the Mode 1 → Mode 2 migration trigger (Section 4): when a decision under a Mode 1 Charter incorporates a substantive AI-authored sub-output that exceeds the embedded-summary scope, the Charter's accountable owner reviews the `mode_declaration` and either rolls back the AI sub-output (keeping the Charter as Mode 1) or amends the Charter to Mode 2. The trigger fires only if the AI worker correctly self-classifies its sub-output. **A deployer training their organization on this Standard should train accountable owners and declaring authorities to scrutinize their own use of AI worker output against the Charter's declared mode.**

C.4.4 Schedule-of-records discoverability decay

The deployer authors a Charter and commits a schedule of records. The first decision records land in a known location. Six months later, an organizational reorganization moves the wiki, the file storage, or the decision-record tooling. Some records migrate; some do not. The schedule of

records is no longer findable in 30 seconds. The Charter's conformance level drops from 3 to 1 not because the Charter changed but because the records the Charter committed to maintain are no longer queryable.

The structural mitigation is the schedule-of-records maintainer role (§C.3.3), the `record_location` field on the Charter (Section 3), and the Section 7 conformance signals that grade against schedule queryability (`schedule_of_records_queryable` , `record_location_resolvable`). The deployer's task is to treat the schedule of records as a load-bearing artifact, not as a folder. When the organization reorganizes, the schedule of records is migrated with the Charter, the `record_location` field is updated, and the conformance-level reporter is run to confirm the records remain queryable.

C.4.5 Re-decision-trigger neglect (§4.7 capture-intent vs anonymization-floor seam)

The deployer authors re-decision triggers as boilerplate. Quantitative thresholds are written down without grounding in the deployer's actual operating context. When a re-decision trigger should fire, the deployer either does not notice (the threshold was never instrumented) or notices and re-confirms rather than re-decides. The Charter's re-decision triggers become decorative.

A subtler variant of this pitfall is the **§4.7 capture-intent vs anonymization-floor seam**. Whoever leads the install, whether staff or an outside adviser, instinctively captures re-decision-trigger metrics specifically, proud of them and eager to show they instrumented quantitative thresholds. Meanwhile the anonymization protocol applies a generic floor that crushes the captured specificity. The install runs cleanly; the captured metrics surface later as anonymization-protocol failures.

The structural mitigation is twofold. First, on the deployer side, Section 4 worked-example template guidance bakes order-of-magnitude banding for quantitative success criteria — re-decision triggers are written down at the order of magnitude that survives anonymization without losing operational meaning. Second, on the install-lead side, install-lead training teaches the §4.7 KEEP / REDACT / REVIEW pre-tag rule with worked examples drawn from the worked-example template's order-of-magnitude banding. **The training schedule itself lives outside Companion C**; Companion C references the seam as a deployer-facing pitfall, and the operational treatment is the install-lead training named at §C.6.1.

C.4.6 Passive promotion to `affirmed`

A deployer authors Charters and dispatches decisions; records reach the `reviewed` state; the deployer's tooling — or the deployer's process — silently promotes records to `affirmed` after a

window of inactivity, after a default-approval timer fires, after no objection is received within a threshold, or via any other passive signal. The accountable owner has not affirmed; no affirmation event is recorded; the seal is computed on a record the named owner did not actively affirm.

This violates Section 5.2's load-bearing affirmation requirement. The structural mitigation is twofold. First, deployer tooling **MUST** refuse to compute a seal or to set the lifecycle state to `affirmed` without a populated `affirmation_record` field carrying timestamp, actor identity, and method per §5.1(3). Second, the Section 7 Level 2 conformance signal `no_passive_promotion_to_affirmed_in_sample` audits a sample of `affirmed` records and surfaces any record whose affirmation method appears to be a passive signal; a Charter cannot grade at Level 2 if the audit returns findings. The deployer's task is to design the install so that affirmation is an affirmative act by the named owner, not a state the record drifts into.

C.4.7 Regulated-function dispatch without licensed-personnel affirmation

Where a Charter binds decisions in a regulated function — legal practice, medical practice, investment advice, accounting attestations, engineering sign-off, and equivalent regimes — Mode 2 dispatch **MUST** require affirmation by a person licensed in the relevant jurisdiction and function. The §5.2 affirmation requirement does not by itself ensure the named affirmer is licensed; the Charter (§3) **MUST** name the licensure requirement explicitly in the `accountable_owner` declaration, and the affirmation event records the affirmer's licensure status as a structural fact at the moment of affirmation. The structural mitigation is twofold. First, deployer tooling **SHOULD** refuse to set the lifecycle state to `affirmed` on a regulated-function record without a populated licensure-status field (the deployer's HR/identity surface determines the field's source of truth; the Standard does not specify the source). Second, the deployer confirms at authoring that the licensure declaration is correct for the regulated function. Deployers operating in regulated functions are advised to consult licensed counsel before authoring Charters that bind regulated decisions; the Appendix G §G.11.3 voluntary-adoption posture and the §1.4.2 "not legal advice" non-claim apply with equal force in regulated-function contexts. A deployer who dispatches regulated-function decisions under Mode 2 without licensed-personnel affirmation has departed from the Standard's audit-trail integrity posture. The Standard makes no claim about anyone's personal liability.

C.4.8 Reading the conformance reporter during the Layer 1 detection-only window

A deployer installs the Standard and begins reading their own conformance-level reporter in the first weeks of operation. During the Layer 1 phased rollout (detection-only weeks 1-3 while the

Layer A+B corpus is assembled; detection-only weeks 4-6 while the Layer C adversarial corpus is constructed; enforcement-mode week 7+), the Level 2 signal `no_silent_mode_drift_in_sample` emits in a known-conservative posture: false negatives are possible because the adversarial corpus is incomplete. A deployer who reads a clean drift signal during weeks 1-6 and concludes the Charter is free of silent Mode 1 → Mode 2 drift is over-reading the signal. This is a required deployer-facing disclosure (core §4.8.5 and §7.3.3): deployer tooling MUST surface the rollout-window status alongside the `no_silent_mode_drift_in_sample` reading so a deployer reading the reporter during weeks 1-6 is told the signal is in conservative posture and that full firing authority begins at week 7. Burying the window status in internal documentation recreates the trust problem the disclosure exists to prevent. The classifier-version question this window raises (whether a classifier increment is "clarifying" or "substantive") is governed by the minor-release non-break commitment at Appendix G §G.7.5.

C.5 Compressed pattern – pointer

The compressed install pattern for sub-Series-C organizations is described in full at §C.2.4 above: the three compressions (Charter shape, standing forum, install order) and the three-move install order are the operational manual. A deployer at sub-Series-C scale runs the compressed Charter and the three-move install order from §C.2.4, holds the deferred practices §C.2.4 enumerates, and grows into the full 90-day sequence as the seats fill.

C.6 Operational pre-conditions

The Standard's install depends on four operational pre-conditions. A deployer attempting the install without them produces decoration. The pre-conditions are not part of the Standard's structural conformance grading — Section 7 does not grade against pre-conditions — but they are the operational substrate the Standard needs to land.

C.6.1 Install-lead training

Whoever leads the deployer's install, whether staff or an outside adviser, is trained on the Standard's full structural set (Sections 2 through 7), the worked-example template's order-of-magnitude banding for quantitative re-decision triggers, the §4.7 KEEP / REDACT / REVIEW pre-tag rule for the anonymization seam, and the Mode 1 / Mode 2 dispatch state machine (Section 4).

The training schedule itself is set by the deployer and is referenced here as a pre-condition; it is not in Companion C's scope.

C.6.2 Tooling minimums

The Standard does not prescribe a tool stack. The deployer is free to use any combination of document storage, decision-record tooling, conformance-level reporter implementation, and schedule-of-records exporter that satisfies Section 6's structural commitments. The minimum is the following: the schedule of records is queryable within 30 seconds by a person not in the room when the decision was made; the conformance-level reporter can read the schedule and emit a conformance-level grade per Section 7; the disclosure block (when applicable) is attached to the Charter or decision record per Section 4 and is reviewable by the declaring authority on the Charter's declared review cadence.

Many implementations of these tooling minimums are possible. The Standard does not certify any tool implementation; it specifies the structural requirements an implementation must satisfy for a Charter or decision record to reach a given conformance level.

C.6.3 Standing forum availability

The deployer has a standing forum — at sub-Series-C scale, the leadership forum the company already has; at Series C and beyond, the seven-seat standing leadership forum (Product Management, Product Marketing, Business Development, Competitive Intelligence, Business Operations, Product Operations, and Customer Success with Value Realization) — that meets on a cadence the Charter declares. A deployer without a standing forum cannot author a Charter that runs; the Charter would have no forum to land in. The standing forum is the operational pre-condition the Standard cannot supply on its own.

C.6.4 IT, security, and data-governance functions engaged before Charter authoring

A deployer authoring a Charter under this Standard **SHOULD** engage the deployer's IT, security, and data-governance functions before Charter authoring begins. Charter authoring without these functions engaged risks committing the deployer to a schedule of records the deployer's actual systems cannot maintain (stable-ID continuity per §6.4; machine-readable schema export per §6.3; seal-hash continuity through migrations per §6.4; supersedes-graph integrity per §5.1(3); RBAC for record write/read/affirm/export consistent with the deployer's identity provider; redaction layer for cross-altitude review per the per-altitude consent posture of Appendix G §G.11.3). Engaging IT, security, and data-governance before Charter authoring is the operational substrate that lets the Charter's commitments survive contact with the deployer's actual systems.

A Charter authored without this engagement may grade at Level 1 against §7 but cannot reliably grade at Level 2 or Level 3 without retroactive system work.

§C.7 — RESERVED. The Implementation Guidance "Explicit non-claim" (install ≠ compliance) is retained in the core Standard at **§10.7**, not in this Companion. This anchor is reserved and unused; do not author content here. Any reference to the install-≠-compliance non-claim resolves to core §10.7.

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 not a regulatory substitute.

Decision Provenance Standard™ v1.2 — Reading Edition (rev. 10). Open standard under CC-BY 4.0. Not a certified product. Founding Steward: Yohay Etsion; institutional Steward: Etsion Brands Ltd.