Svelto Contextual Assurance: Methodological & Cryptographic Open Manifest

Version: 1.0
Publication Date: 2026-03-12
Canonical URL: https://docs.svelto.io/methodology/v1
Status: Public - Approved for External Audit Review (Founding Partner Phase)

---

1. Purpose and Scope

This document exists for one reason: to allow an external auditor to evaluate whether the evidence Svelto generates is credible, independently verifiable, and correctly scoped - without access to the Svelto platform itself.

It describes, at a conceptual level:

- How policy is translated into test scenarios.
- What human review occurs before any scenario reaches an employee.
- How governed manual dispatch is controlled and disclosed when the standard scheduler is bypassed.
- How pass/fail is calculated and what the numbers mean.
- How evidence integrity is cryptographically anchored by a neutral third party.
- How the complete evidence record can leave Svelto's custody and be stored in a customer-controlled location.
- The precise boundary of what Svelto attests and what it explicitly does not attest.

Versioning: Document version increments with every material change.

Publication Status: This document is publicly available at https://docs.svelto.io/methodology/v1. No confidentiality restrictions apply. Auditors may reference, cite, or distribute this document without permission.

> **Note:** This document does not describe technical implementation details. It describes the methodology - the rules, constraints, and human obligations the system enforces.

---

2. Policy Ingestion: Dual-Vector Semantic Constriction

Svelto does not generate generic security awareness content. Every micro-simulation is bound by a strict Two-Phase Retrieval-Augmented Generation (RAG) architecture that deconstructs your internal policy into verifiable compliance metadata.

2.1 Dual-Vector Disqualification (The Semantic Guardrail)

During setup, policies are mapped strictly to the organization's configured framework (e.g., ISO 27001:2022). Every framework control in Svelto's database is stored with two distinct mathematical embeddings:

- **The Inclusion Vector (`embedding`):** Represents what the control is.
- **The Exclusion Vector (`not_for_embedding`):** Represents what the control is not, utilizing a systematically validated, versioned list of explicitly excluded technical concepts maintained by Svelto and published in the Exclusion Taxonomy.

When analyzing policy text, the system computes similarity against both vectors. If the text is mathematically more similar to the exclusion vector, the control is programmatically disqualified:

```
not_for_similarity > similarity
```

Example: A policy discussing the "Integrity of Books and Records" will not be falsely mapped to an IT security control because the exclusion vector recognizes the financial context and strips it out.

Exclusion Taxonomy: The complete, versioned Exclusion Taxonomy - listing every explicitly excluded concept, the rationale for each exclusion, and the date each entry was added - is published in the Exclusion Taxonomy. This taxonomy is systematically validated through a multi-layer review process combining automated semantic verification and human expert spot-checking. Each entry is versioned and auditable. Any update to the taxonomy increments both the taxonomy version and this document's version.

2.2 Observability and Deterministic Fallbacks

Svelto enforces strict limits (maximum 3 primary controls, 5 related controls per chunk). If the AI engine exceeds these boundaries or fails confidence thresholds, the system triggers a deterministic raw-text fallback and logs a critical observability event. The AI has zero final authority; all control mappings remain unverified drafts until a named Administrator formally attests the fully assembled scenario at the Admin Attestation Gate (§4.1). Authorization occurs at the scenario level, not the mapping level - the Administrator evaluates a complete, deployable situation with its correct answer and policy reference, not intermediate retrieval artifacts.

2.3 RAG Retrieval Confidence Thresholds

When retrieving policy chunks to generate scenarios, the system enforces fixed confidence thresholds:

- **Primary retrieval:** confidence >= 0.70 (chunks with strong semantic match to the target control)

- **Fallback retrieval:** confidence >= 0.50 (expanded search if primary retrieval yields insufficient results)

These thresholds are hardcoded system constants, not customer-configurable parameters. They were established during system design to balance precision (avoiding false-positive mappings) against recall (ensuring sufficient content is available for scenario generation). If retrieval at both thresholds yields zero results, scenario generation for that control is aborted rather than proceeding with low-confidence content.

---

3. Deterministic Scenario Generation Pipeline

To ensure the "Closed Loop" is broken, Svelto forces the AI through a proprietary three-phase generation pipeline. The AI cannot invent scenarios; it acts as a constrained parser of your uploaded evidence. Every scenario is generated just-in-time (JIT) at the point of delivery - scenarios are not pre-generated or cached, which means the same employee cannot recognize and answer a scenario from prior memory across compliance periods.

3.0 Methodological Basis

The scenario design rules in this section are not arbitrary. They are grounded in two governing NIST publications that establish the federal standards for security awareness and training program design:

NIST Special Publication 800-16 (Information Technology Security Training Requirements: A Role- and Performance-Based Model)

This publication defines the Awareness -> Training -> Education learning continuum and establishes the valid test instruments for each level. Svelto's scenario format operates within the Training band of this continuum, as defined in SP 800-16 Exhibit 2-2:

Level	Test Measure	Svelto Implementation
Awareness	True/False, Multiple Choice (identify learning)	Not used at this cognitive level
Training	Problem Solving: Recognition and Resolution (apply learning)	Svelto's scenario format - active judgment
Education	Essay (interpret learning)	Outside scope - theoretical competency

Note: Multiple choice is the delivery format across levels. The distinguishing

factor is cognitive demand - Awareness-level multiple choice tests passive recognition ("have you seen this rule?"), while Training-level multiple choice tests applied judgment ("given this situation, what is the correct action under policy?"). Svelto scenarios are constructed to require the latter.

Svelto does not operate at the Awareness level (passive information reception) nor at the Education level (theoretical interpretation). It operates explicitly within the Training level, requiring employees to apply policy knowledge to recognize and resolve realistic situations. This is the correct test instrument for the stated objective: verifying that employees can execute policy-driven decisions under operational conditions.

The difficulty stratification in §3.1 maps to the progression described in SP 800-16 §2.1: from basic "Security Awareness" (recognition) through "Security Basics and Literacy" (foundational application) to "Roles and Responsibilities Relative to IT Systems" (contextual judgment).

NIST Special Publication 800-50 Revision 1 (Building a Cybersecurity and Privacy Learning Program)

This 2024 update establishes the modern requirements for an integrated Cybersecurity and Privacy Learning Program (CPLP). It shifts the focus toward managing organizational risk through measurable behavior change and continuous monitoring. Svelto's operational rules implement the following requirements from SP 800-50r1:

- **§4.3 Program Strategy & Data-Driven Design** - requires that learning activities be informed by organizational risk and performance data. Svelto's Policy Ingestion engine ensures that scenarios are not generic, but are derived directly from the user's uploaded evidence, fulfilling the Rev. 1 requirement for "contextually relevant learning."
- **§6.1 Monitoring & Strategy Evaluation** - mandates the use of automated tools to monitor program effectiveness and collect "learning metrics" for stakeholders. Svelto's xAPI-based Human Compliance Telemetry provides the "unfiltered visibility" into workforce risk drift recommended by the update.
- **§6.2 Continuous Improvement (Corrective Action)** - requires a feedback loop to address identified competency gaps. Svelto's Remediation Lifecycle (§4.5) automates this by triggering "Calibration Micro-Sims" based on real-time failure data, directly implementing the Rev. 1 "Iterative Improvement" mandate.
- **Privacy & Cyber Integration** - Rev. 1 requires the integration of privacy into the learning lifecycle. Svelto's pipeline treats Privacy (e.g., PII handling, Data Minimization) as a first-class control family, ensuring evidence artifacts satisfy both Security and Privacy audit requirements.

These publications are available at https://csrc.nist.gov. Auditors are encouraged to verify that Svelto's "Human Static Analysis" approach provides the quantitative impact data required by the 2024 CPLP framework.

3.1 Hardcoded Difficulty Stratification

Subjective difficulty and predictable patterns allow for manipulated pass rates. Svelto prevents this programmatically during the initial Blueprinting phase by enforcing mandatory system distributions.

These distributions implement the role-based learning progression model described in NIST SP 800-16 §2.1:

Difficulty Distribution

- **60% Basic** - Single-variable identification (e.g., identifying a clear PII violation in a message). Maps to the "Security Awareness" level of the NIST continuum.
- **35% Intermediate** - Contextual dependency (e.g., determining whether data shared with a Guest vs. Internal user violates the "Need to Know" clause). Maps to the "Security Basics and Literacy" level.
- **5% Advanced** - Policy conflict / nuance (e.g., navigating "Emergency Access" exceptions against standard MFA requirements). Maps to the "Roles and Responsibilities" level.

Signal-to-Noise Ratio

- **80% Violation Scenarios**
- **20% Legitimate/Benign Scenarios**

Rationale for Proportional Distribution

The 60/35/5 distribution reflects the sequential learning progression defined in NIST SP 800-16 §2.1. The majority of scenarios are Basic because foundational recognition competency must be established before higher-order judgment can be meaningfully tested - consistent with the SP 800-16 model that treats each level as a prerequisite for the next. The Advanced tier is deliberately narrow (5%) because contextual judgment under competing policy constraints cannot be fairly assessed in a population that has not yet demonstrated baseline adherence. Assigning a higher proportion of Advanced scenarios to an unprimed workforce would produce failure artifacts that reflect test design error, not genuine behavioral risk.

The 20% benign baseline directly addresses the acclimation phenomenon described in NIST SP 800-16 §2.2.1, which warns that repeated identical stimuli cause learners to selectively ignore them - producing Pavlovian response patterns rather than genuine policy evaluation. The benign baseline ensures employees must actively evaluate each scenario against the policy context, achieving what SP 800-16 calls **assimilation**: incorporating new learning into conscious decision-making rather than reflexive pattern-matching.

The Administrator cannot alter these distributions to inflate competence metrics. The audit artifact evidence summary explicitly states the benign scenario count for every compliance run (e.g., "15 scenarios delivered: 12 violation, 3 be-

nign"), allowing auditors to independently verify the distribution was honored without platform access.

Auditor's Yardstick: The rubric above gives an auditor the means to evaluate the substantive difficulty of any scenario in the evidence record without platform access. Each scenario's `difficultyLevel` field in the xAPI statement maps directly to one of the three categories above and can be independently assessed against the NIST SP 800-16 Exhibit 2-2 framework. The complete xAPI field definitions are published in the xAPI Profile.

3.2 Independent Retrieval and Audit Citations

Scenarios are generated via an independent RAG query specifically tuned to the topic blueprint.

- **Navigation Defense:** The AI is mathematically warned against confusing document navigation references (e.g., "Section II.4") with Compliance Control IDs.
- **Traceability:** Policy-grounded scenarios preserve strict `sourceCitations` and may include a `policyReference`, proving the clause tested when organization training materials are used.

3.3 The Validation Guard and Audit Anchor

Svelto employs a 3-attempt validation loop. If the AI suggests an invalid control ID, the system rejects it. Before reaching a human, the `primaryControlId` is forcefully anchored to the framework's designated Awareness/Training control (e.g., NIST AT-2, ISO A.6.3, SOC 2 CC2.2), regardless of what the AI generated.

3.4 Immutable Scoring Logic

Svelto enforces a strict separation between Operational Context and Compliance Logic.

- **Administrative Constraint:** Administrators cannot manually edit, override, or "force-pass" any scenario's scoring rubric or control mapping.
- **Conflict Resolution:** If an Administrator disputes the AI's mapping or correctness logic, the system prohibits manual correction. The Administrator must either (a) update the source policy to clarify the rule, or (b) reject the scenario, which triggers the Anti-Fishing Protocol (§3.5).
- **Audit Value:** This ensures the pass/fail result is an objective product of the policy and the engine, not administrative discretion.

3.5 The Anti-Fishing Protocol (Integrity Guard)

To prevent "grading on a curve," Svelto prevents administrators from cycling scenarios to find easier versions for their employees.

- **Fixed-Variable Regeneration:** If a scenario is rejected by an Admin, the engine is programmatically locked to regenerate a replacement for the identical Control ID at the identical Difficulty Level.
- **Negative Evidence Logging:** Every rejection event - including the discarded scenario and the Admin's stated reason - is hashed and exported to the external TSA and the Customer SIEM.
- **Audit Value:** Every rejection is a permanent, hashed record in the audit trail. An auditor can review the full rejection history for any scenario in the evidence record without access to the Svelto platform.

3.6 Governed Manual Dispatch

The standard delivery path is scheduler-driven. Svelto also permits a narrowly governed manual dispatch path for operational testing and time-bound compliance operations. This path invokes the same micro-simulation pipeline used by the scheduler and does not bypass scenario-generation constraints, human approval requirements, or scoring rules.

Manual dispatch is restricted to Super Administrators and limited to a maximum of two dispatches per room within a rolling 7-day window. Each manual dispatch creates a distinct provenance record containing the dispatching administrator identity, room reference, dispatch timestamp, and optional reason. That provenance record is protected by a SHA-256 integrity hash and an RFC 3161 trusted timestamp.

When manual dispatch is used, the audit artifact includes a dedicated Manual Dispatch Events section for the reporting period. The raw xAPI export flags statements originating from manually dispatched tasks, allowing auditors to distinguish governed out-of-band dispatches from scheduler-driven delivery without platform access.

---

4. Human Oversight, Scoring, and Remediation

Svelto relies on behavioral telemetry and deterministic scoring rules, not AI judgment at response time.

4.1 The Admin Attestation Gate

Every scenario must pass a mandatory Admin Attestation Gate before deployment. Attestation is an all-or-nothing judgment: the Administrator either approves the scenario as a complete unit or rejects it. There is no partial approval, no per-field override, and no ability to edit any component of the scenario.

The Administrator reviews the fully assembled scenario, including control mapping, difficulty level, narrative, policy reference, source citations, and answer

alternatives, and attests that the scenario is valid for deployment in the organization's environment. The attestation event itself is recorded as a discrete, timestamped audit-trail event under the named identity of the attesting Administrator.

The Administrator cannot modify the scoring rubric, correct answer, difficulty level, control mapping, or other system-governed parameters. Those elements are fixed by the methodology and preserved as part of the scenario artifact presented for attestation. If the Administrator cannot attest to the scenario as presented, the only available action is rejection, which triggers the Anti-Fishing Protocol (§3.5).

For rejected scenarios that are regenerated, Svelto preserves the original evidence and constrains the replacement workflow so that the replacement must remain within the same locked methodological envelope defined by the system-governed parameters applicable to that scenario type. The Administrator therefore cannot use regeneration to rewrite the scoring model or relax the scenario into a different methodological class.

Administrator Roles and Identity Each organization has one Super Administrator who holds primary compliance authority and is responsible for program configuration and attestation obligations. The Super Administrator may provision additional Administrators, who share attestation authority within the same system constraints. All attestation events are logged under the individual Administrator's named identity, regardless of role level, ensuring every approval and rejection is attributable to a specific person. No anonymous or role-level attestations are permitted.

4.2 Objective Pass/Fail Evaluation

Once attested, the scenario becomes a deterministic artifact for scoring purposes. When an employee answers, the system records a discrete xAPI statement (`verb = "answered"`). No AI evaluation occurs at response time. Pass/fail is determined by a binary comparison of the employee's submitted response against the pre-attested correct answer recorded at scenario creation.

Scenarios use structured multiple-choice responses delivered through the configured communication channel. Pass/fail is an exact-match comparison between the employee's selected option and the pre-attested correct option. No fuzzy matching, partial credit, or AI interpretation occurs during evaluation.

The complete xAPI Profile, including field definitions, verb vocabulary, extension keys, and example statements, is publicly available. Auditors are encouraged to review the profile to independently verify that the fields referenced throughout this document, including `difficultyLevel`, `scenarioType`, `sourceCitations`, and, when present, `policyReference`, carry the meaning described here.

4.3 Configurable Compliance Metrics with System-Enforced Floors

Svelto computes compliance status for each employee and reporting period from a set of metrics configured by the Administrator at program setup. These threshold values are **locked at room creation**: the exact configuration in force when a simulation room is opened is recorded as an immutable snapshot bound to that room for its entire lifecycle. Subsequent administrator changes to thresholds apply only to rooms created from that point forward and never retroactively alter evidence already in progress. Each room's locked thresholds - including the effective date and the identity of the administrator who set them - are recorded in every audit artifact, allowing auditors to verify the exact thresholds that governed each room's pass/fail outcomes without platform access.

The following metrics are Administrator-configurable, subject to system-enforced constraints:

Metric	Description	System Constraint
`accuracyAuditReadyPercent`	Minimum pass rate required to achieve Audit Ready status	Floor: 60%
`accuracyCriticalDriftPercent`	Pass rate at or below which Critical Drift is triggered	Must be lower than `accuracyAuditReadyPercent`
`velocityAuditReadySeconds`	Maximum median response time for Audit Ready status	Organization-defined
`velocityCriticalDriftSeconds`	Response time at or above which Critical Drift is triggered	Must be greater than `velocityAuditReadySeconds`
`pendingResponseHours`	Hours before an unanswered scenario is flagged overdue	Maximum: 72 hours
`integrityAccuracyWeight`	Weighting of accuracy in the composite integrity score	Accuracy and velocity weights must sum to 1.0
`integrityVelocityWeight`	Weighting of response velocity in the composite integrity score	Accuracy and velocity weights must sum to 1.0
`baselineSeconds`	Expected median response time for a well-prepared employee	Used as velocity benchmark

A threshold set at the system floor of 60% is itself a visible policy choice in the audit artifact and may warrant examiner scrutiny. Setting

`pendingResponseHours` at or near the 72-hour ceiling is similarly visible to the examiner. Because thresholds are locked at room creation, any change made after a room opens does not affect that room's evidence record; the original locked values remain the sole governing criteria for that room.

Velocity as a Compliance Signal Response velocity, defined as median time from scenario delivery to employee response, is treated as a behavioral signal rather than a hidden scoring heuristic. The methodology is based on the premise that unusually slow response times under realistic operational conditions may indicate policy uncertainty even when the final answer is correct. An employee who answers correctly but only after significant delay may understand the policy conceptually while lacking operational fluency to apply it confidently under time pressure.

Velocity is therefore configurable rather than mandatory in the composite integrity score. Organizations that do not accept this model may set `integrityVelocityWeight = 0`, but because the integrity weights must sum to `1.0`, this only works when `integrityAccuracyWeight = 1.0`. That configuration removes velocity from the composite score calculation only; velocity thresholds still affect audit-ready status, critical-drift detection, and velocity-specific analytics. The selected weights are stored in versioned settings and locked to rooms at creation time, but current generated audit reports do not render those weights directly.

Composite Integrity Score The composite integrity score combines accuracy and velocity into a single weighted metric:

```
Integrity Score = (Accuracy Rate * integrityAccuracyWeight) + (Velocity Performance * integ
```

Both `Accuracy Rate` and `Velocity Performance` are expressed on a 0-100 scale. Velocity Performance is calculated as:

```
100, if median response time <= velocityAuditReadySeconds
0, if median response time >= velocityCriticalDriftSeconds
otherwise: 100 - (((median response time - velocityAuditReadySeconds) / (velocityCriticalDri
```

A score of 100 represents perfect accuracy with response performance at or better than the configured audit-ready threshold. The composite score is surfaced in dashboard analytics together with `accuracyRate`, `velocityScore`, and the configured weights. Current generated audit reports instead emphasize the room's locked operational thresholds, such as pass threshold and pending-response window. An organization that sets `integrityAccuracyWeight = 1.0` and `integrityVelocityWeight = 0.0` is explicitly declaring that only correctness contributes to the composite score, while velocity remains part of the broader operational classification model.

4.4 Administrator Approval Rate Transparency

To provide transparency into scenario curation behavior, Svelto tracks and reports the Administrator approval rate as the percentage of scenarios approved relative to the total number of scenarios presented for attestation. This metric is included in the audit artifact as a transparency signal only.

A high rejection rate may reflect legitimate policy misalignment, quality-control scrutiny, or selective curation behavior. A low rejection rate may reflect operational alignment, permissive approval behavior, or a stable generation context. Approval-rate data therefore does not, by itself, prove the absence of bias or confirm the quality of administrative judgment.

For that reason, approval-rate reporting is not used to determine compliance status. Instead, it provides auditors with a visible summary of attestation behavior that can be cross-checked against the raw xAPI evidence. Auditors can independently verify the reported figures, named approvers, rejection reasons, timestamps, and approval/rejection chains for individual scenarios against the cryptographically anchored xAPI export without requiring platform access.

No cross-customer benchmarking is implied or required for audit validity. The purpose of this metric is traceability and examiner visibility, not normative scoring.

4.5 Remediation Lifecycle and Corrective Action Evidence

When an employee's aggregate performance triggers a compliance gap, the Administrator may initiate a targeted remediation cycle. Remediation is not a reset - it is an additive evidence layer on top of the employee's existing xAPI record.

Remediation Content Generation

Remediation scenarios are generated using the same RAG pipeline described in §3, but targeted specifically at the control ID where the gap was identified. The system retrieves policy chunks using a three-pass strategy (control-based at confidence $>= 0.70$, relaxed confidence fallback at $>= 0.50$, semantic keyword fallback using hardcoded control vocabulary) and generates scenarios constrained to that control's policy context. The same 3-attempt validation loop, control anchoring, and Admin Attestation Gate apply.

Difficulty in Remediation

Remediation does not impose a fixed easy difficulty. The same 60/35/5 distribution target applies across the employee's entire session history - regular and remediation sessions combined. Because an employee in remediation typically has an underfill at Basic level, the system naturally steers toward foundational content. However, this is a gap-analysis outcome, not a hardcoded override. The difficulty is always determined by what the historical distribution requires, not by the remediation state itself.

Verification Threshold for a Remediation Cycle

In the current implementation, each remediation tracking record is created with a fixed verification requirement of `3` scored remediation simulations. Each correct answer increments `correctCount`; each incorrect answer increments `failureCount`. The tracking record remains in active monitoring until three total verification attempts have been recorded. At that point, the outcome is determined by simple majority: `RESOLVED` if the employee answered at least `2 of 3` correctly, and `FAILED` if the employee answered `2 of 3` incorrectly. This threshold is currently hardcoded in the application, not administrator-configurable.

Remediation Does Not Automatically Restore Compliance

This is a deliberate design decision, explicitly encoded in the system. Completing remediation proves that targeted corrective training was delivered and verified. It does not clear the employee's compliance status. Audit Ready status is restored only when the employee's rolling aggregate xAPI record - across all sessions in the reporting period - meets the configured `accuracyAuditReadyPercent` and velocity thresholds again. Remediation is the mechanism; ongoing performance is the proof.

Terminal States and Mandatory Administrator Acknowledgment

The system is designed so that nothing closes automatically. Both terminal remediation states require explicit administrator action, and both generate permanent audit trail entries:

- **RESOLVED (employee passed remediation):** In the current implementation, this means the employee achieved a majority-correct outcome for that remediation tracking record, which today is `2 of 3` correct scored remediation simulations. The tracking record remains in RESOLVED / pending acknowledgment state and appears in the audit artifact under `completedRemediationEvidence`. It only moves to acknowledged status when the Administrator explicitly clears it - an action that records their email, timestamp, and archive date permanently in the audit trail.
- **FAILED (employee did not pass remediation):** In the current implementation, this means the employee reached a majority-incorrect outcome for that remediation tracking record, which today is `2 of 3` incorrect scored remediation simulations. The tracking record enters FAILED / escalation suggested state. The system does not auto-escalate. The audit artifact surfaces the finding with explicit guidance: (a) escalate to human intervention (manager, HR, or compliance officer), or (b) clear the failed status and initiate a new remediation cycle. Until the Administrator takes one of these actions, the system blocks any new remediation generation for that employee/control pair. The resolution action - whichever path is taken - is permanently recorded with admin email, timestamp, and outcome.

Audit Value: The remediation section of every audit artifact distinguishes

between acknowledged instances and pending acknowledgment instances. An auditor can see, for any compliance period: how many remediations were initiated, how many resolved, how many failed, and whether each outcome received timely administrator acknowledgment. Unacknowledged RESOLVED or FAILED records sitting open at period close are themselves a visible finding.

4.6 Employee Notification and Disclosure Policy

Svelto does not mandate a specific employee disclosure posture. Administrators are responsible for ensuring their deployment complies with applicable employment law and data protection regulations in their jurisdiction, including but not limited to GDPR (EU), LGPD (Brazil), and CCPA (United States).

The configured disclosure posture - whether employees are notified of the compliance program, when, and in what detail - should be documented in the organization's own compliance records as context for the Svelto evidence package. This is a customer policy decision, not a platform constraint.

---

5. The Audit Artifact

5.1 Format and Generation

Svelto generates a PDF audit artifact on demand, scoped to a customer-selected compliance period (1, 3, 6, or 12 months). The PDF is the primary deliverable for audit submission.

For auditors or compliance systems requiring structured data, the xAPI JSON export of the full evidence record is available via direct API access to the administrator. The export mechanism is authenticated via the administrator's existing platform credentials and delivered as a cryptographically anchored JSON file within 60 seconds of request. The export includes each statement's integrity hash, RFC 3161 token, and the TSA provider metadata recorded with that statement, plus a separate export-level RFC 3161 token over the export bundle itself. For scored micro-simulation interactions, the raw xAPI record also preserves the verbatim question text, scenario context, selected answer, canonical correct answer, and related scoring context. No manual approval or third-party mediation is required.

When evidence escrow is configured, underlying telemetry is continuously routed to the customer's designated SIEM or storage endpoint (§7.1), meaning the customer holds an independent copy of the raw evidence outside Svelto infrastructure.

5.2 Methodology Version Locking

The methodology version in effect at the time of artifact generation is recorded as an immutable field in every audit artifact. If the live methodology document

is updated after a compliance period closes, the artifact retains the version that governed that period. Auditors can retrieve the exact methodology version referenced in any artifact from `docs.svelto.io/methodology/v[version]`.

5.3 Evidence Summary Fields

Every audit artifact includes the following summary fields. Fields derived from behavioral telemetry are independently computable by the auditor from the raw xAPI export without platform access. Threshold provenance fields (locked values, effective date, configuring administrator) are drawn from the audit artifact itself, which is the authoritative record for that compliance period:

- Total scenarios delivered; breakdown by difficulty level (Basic / Intermediate / Advanced)
- Total scenarios delivered; breakdown by type (Violation / Benign)
- Administrator approval rate (scenarios presented vs. attested)
- Manual dispatch disclosures, when applicable: event count and per-event disclosure of dispatch time, room, dispatching Super Administrator, and stated reason; underlying provenance records are retained in the evidence escrow payload and termination export, and related xAPI statements are flagged in the raw export
- Per-room locked compliance thresholds: pass accuracy threshold (`accuracyAuditReadyPercent`) and pending response window (`pendingResponseHours`) - the two values that directly gate each user's PASS / FAIL / PENDING verdict - each accompanied by the settings version effective date and the identity of the administrator who configured them; a disclosure paragraph is included when rooms in the report were governed by more than one settings version. The remaining six governed fields (velocity thresholds, integrity weights, baseline) are recorded in the locked settings version and visible in the admin portal but are not repeated per-room in the PDF because they do not individually alter pass/fail outcomes.
- Aggregate pass rate and composite integrity score for the compliance period; individual-level breakdown for employees identified as Critical Drift (up to 15 highest-risk actors, ranked by drift factor), including per-control failure detail, reaction time, and remediation status for each surfaced actor
- Compliance status distribution across the population (Audit Ready / In Training / Critical Drift) as cohort counts and percentages; individual compliance status shown only for Critical Drift actors surfaced in the exception log
- Remediation summary: initiated, resolved, failed, acknowledged, pending acknowledgment
- Zero-Trust Audit Trail summary describing statement-level RFC 3161 anchoring and the availability of independently verifiable raw xAPI export data; detailed TSA provenance is retained in the underlying xAPI export and report-generation records rather than enumerated exhaustively in the

PDF body
- Methodology version governing this period

5.4 Data Retention Policy

Svelto retains raw telemetry (xAPI statements, TSA tokens, scenario metadata) for up to 12 months after successful delivery to the customer's designated SIEM or storage endpoint, matching the longest supported compliance period. After 12 months, the customer's copy is the sole authoritative record. Customers requiring longer retention within Svelto infrastructure may request extended retention as a contract-level service term.

During subscription termination, an authorized administrator can generate a complete evidence export (xAPI statements, TSA tokens, audit artifacts, remediation records, and manual dispatch provenance records). In MVP dedicated-instance deployments, infrastructure may be decommissioned before 12 months once transfer is confirmed through one of the following paths: (a) endpoint delivery receives HTTP 2xx, or (b) manual download is explicitly acknowledged by an authorized customer administrator. Without transfer confirmation, retention remains available up to the 12-month maximum. This process is designed to support data portability obligations under GDPR, LGPD, and comparable data protection regulations.

---

6. Independent Cryptographic Anchoring (RFC 3161)

Generating a SHA-256 hash on a PDF you printed yourself is a self-referential act. Svelto breaks this closed loop by anchoring telemetry to an independent external authority.

6.1 External Timestamp Authority (TSA)

Svelto utilizes **RFC 3161 Trusted Timestamping** at two layers. First, every xAPI statement is anchored at the moment of employee execution. Administrators configure the active TSA provider from Svelto's published list of supported authorities, and the resolved provider ID and URL are persisted with each individual xAPI statement. If the administrator changes TSA settings over time, a single compliance period may contain statements anchored by different providers. Second, audit-report generation and xAPI-export generation each record a separate RFC 3161 anchor over the artifact integrity hash at generation time. The PDF is therefore derived from already anchored telemetry and accompanied by its own generation-time provenance record, even though the visible PDF body does not enumerate every underlying statement-level provider.

Supported TSA providers:

Provider	Type	Recommended For
DigiCert Timestamp Authority	Commercial, enterprise-grade	Regulated industries (FedRAMP, PCI-DSS, HIPAA)
Sectigo (formerly Comodo)	Commercial, enterprise-grade	Regulated industries
Freetsa.org	Public, community-operated	Non-regulated environments only

Organizations operating under regulated compliance regimes are strongly recommended to select a commercial TSA provider with a published SLA and legal standing appropriate to their compliance environment. Freetsa.org is a community-operated service with no uptime SLA and is not recommended for organizations subject to regulatory audit.

The anchoring process operates as follows:

1. At the moment an employee executes a decision, a hash of the xAPI statement is submitted to the configured TSA.
2. The database write is strictly blocked until a valid `.tst` token is returned.
3. If the TSA is unreachable, the anchoring attempt fails immediately with a hard error and the scenario response is not recorded. The employee receives an error state and must re-engage with the scenario. No unanchored responses enter the evidence record under any failure condition.

6.2 Independent Verification

An auditor can validate the RFC 3161 tokens against the TSA's public certificate chain using standard open-source tooling:

```
openssl ts -verify -in <token.tst> -data <xapi_statement.json> -CAfile ca.pem
```

For statement-level verification, the auditor uses the TSA provider recorded with the specific xAPI statement in the raw export. For audit-report or export-generation verification, the auditor uses the TSA provider recorded with that report-generation record. Because TSA settings are administrator-configurable and may change over time, a single compliance period must not be assumed to have one universal provider. The CA chain must be obtained from the provider associated with the exact token under review. This verification requires no access to Svelto infrastructure and cryptographically proves that Svelto did not backdate, alter, or fabricate execution telemetry after the fact.

---

7. Decentralized Evidence Escrow

7.1 Automated Cryptographic Batch Routing

Evidence is routed as cryptographically signed batch payloads to the administrator's designated AWS WORM S3 bucket, configured SIEM ingest endpoint, or both, on a hardcoded 24-hour maximum interval. In v1.0, the supported escrow destinations are AWS S3 WORM and Splunk HEC. This interval is not customer-configurable. Svelto monitors delivery pipelines and alerts the configured workspace administrator via Slack notification immediately if delivery fails, ensuring zero silent failures in the chain of custody.

7.2 Zero-Trust Custody

Svelto's escrow integration is one-way. For AWS S3 WORM, the recommended customer configuration grants `PutObject` only. For SIEM ingest, Svelto submits append-only delivery requests to the configured endpoint. It does not use read or delete operations against the customer's escrow copies. Once delivered, the customer's copy is the authoritative copy.

---

8. The Attestation Boundary

Svelto provides forensic execution telemetry. It does not attest to technical configuration.

8.1 What Svelto Proves (Human Control Telemetry)

Svelto generates primary, standalone audit evidence for:

- **NIST 800-53 Rev5:** AT-2 (Literacy Training and Awareness)
- **ISO 27001:2022:** A.6.3 (Information security awareness, education and training)
- **SOC 2 (2017):** CC2.2 (COSO Principle 14: Internal Communication)

Svelto operates at the applied knowledge layer of the security training continuum. Rather than recording that a policy was delivered or acknowledged, it tests whether employees can apply policy knowledge to recognize and resolve realistic operational situations - producing verifiable evidence that training has produced measurable behavioral competence.

8.2 What Svelto Explicitly Does Not Prove

Svelto tests the human layer of technical controls. It does not verify technical implementation:

- **Access Control:** Proves employees identify violations; does not prove IAM/RBAC configuration.

- **Authentication:** Proves employees understand rules; does not prove MFA is technically enforced.
- **Data Protection:** Proves employees recognize handling requirements; does not prove encryption is deployed.

---

9. Published Companion Documents

The following documents are published alongside this methodology and form part of the complete audit evidence package. Each is independently versioned; updates to any companion document that affect the methodology described here will increment this document's version.

Document	URL	Purpose
Exclusion Taxonomy v1.0	https://docs.svelto.io/methodology/exclusion-taxonomy/v1	Complete list of explicitly excluded concepts used by the Dual-Vector Disqualification engine (§2.1)
xAPI Profile v1.0	https://docs.svelto.io/methodology/xapi-profile/v1	Complete field definitions, verb vocabulary, extension keys, and example statements for all xAPI evidence records

---

10. Immutable Versioning

Field	Value
Document Version	1.0
Publication Date	2026-03-12
Canonical URL	docs.svelto.io/methodology/v1
Exclusion Taxonomy Version	1.0
xAPI Profile Version	1.0
Governing Publications	NIST SP 800-50 Rev. 1; NIST SP 800-16
TSA Provider	Administrator-configured - see §6.1 for supported providers
Minimum Pass Threshold Floor	60% (system-enforced, non-configurable)

Field	Value
Maximum Pending Response Window	72 hours (system-enforced, non-configurable)
Evidence Batch Routing Interval	24 hours maximum (system-enforced, non-configurable)
RAG Retrieval Confidence Thresholds	Primary: 0.70 / Fallback: 0.50 (system-enforced, non-configurable)
Data Retention Period	Up to 12 months post-delivery; dedicated instance can be decommissioned earlier after confirmed termination export transfer
Scenario Generation	Just-in-time (JIT) - no pre-generation or caching
Status	Public - Approved for External Audit Review (Founding Partner Phase)

---

11. Contact / Discrepancy Reporting Mechanism

To allow independent audit scrutiny, Svelto maintains a public discrepancy reporting channel for inconsistencies between this methodology and any generated audit artifact.

Auditors can report discrepancies by email at contact@svelto.io.

To accelerate triage, each report should include at minimum:

- Artifact identifier (for example, artifact ID or canonical reference).
- Methodology URL and version being referenced.
- A concise description of the observed inconsistency.

Service expectations: Svelto acknowledges receipt within 2 business days and provides an initial follow-up within 5 business days.

Immutable Engineering Attestation

Field	Value
Document Version	1.0 (LATEST)
Status	Approved for External Audit
Canonical ID	S-METH-1.0-2026-03-12