PREPUBLICATION REVIEW | RC1

TERASYSTEMSAI LLC

TRUSTWORTHY AI TECHNICAL SERIES

Distribution Shift Assurance

Monitoring, Evidence, and Governed Response for Deployed AI Systems

TSAI-TR-2026-001

Version 1.0-RC1 | August 2026

TeraSystems Research

TeraSystemsAI LLC | Pennsylvania, USA

Independent technical guidance for trustworthy AI engineering.

Publication Notice

Publication	Distribution Shift Assurance: Monitoring, Evidence, and Governed Response for Deployed AI Systems
Series	TeraSystemsAI Trustworthy AI Technical Series
Identifier	TSAI-TR-2026-001
Version	1.0-RC1
Date	August 2026
Status	Prepublication Review Draft
Authoring team	TeraSystems Research, TeraSystemsAI LLC
Companion	Distribution Shift: Why Models Fail Quietly

Independent publication boundary

This document provides engineering and research guidance. It is not a government standard, legal opinion, regulatory certification, conformity assessment, or representation of the National Institute of Standards and Technology (NIST), the International Organization for Standardization (ISO), the European Union, or any other standards or regulatory body. References and mappings do not imply endorsement.

Purpose. This guide translates TeraSystemsAI's public trustworthy-AI principles into an operational assurance method for distribution shift and post-deployment change. It is intended to help technical and governance teams detect changed conditions, determine whether those changes matter, preserve evidence, retain accountable human authority, and respond proportionately.

Public-detail boundary. TeraSystemsAI's public research program states that

certain algorithmic formulations, calibration mechanisms, and deployment architectures are intentionally withheld to protect ongoing intellectual-property development. This report therefore focuses on assurance logic, control structure, evidence requirements, and implementation patterns rather than proprietary internals.[10]

Use of external frameworks. The NIST AI Risk Management Framework (AI RMF 1.0) is used as a structural reference because it separates foundational risk reasoning from operational functions. NIST describes the AI RMF as voluntary guidance and, as of August 2026, states that AI RMF 1.0 is being revised.[1, 2] ISO/IEC 42001:2023 and ISO/IEC 23894:2023 are referenced only at a high level. The EU AI Act discussion uses the consolidated text current to 27 July 2026.[4–6] No mapping in this report constitutes conformity, certification, or legal advice.

Prepublication status

This edition is a human-review draft. Technical claims, external-framework mappings, references, and publication language should receive final human verification before public release. The publication identifier is reserved, but this draft should not be represented as an external standard or certification.

Recommended citation after release

TeraSystems Research. (2026). *Distribution Shift Assurance: Monitoring, Evidence, and Governed Response for Deployed AI Systems* (TSAI-TR-2026-001, Version 1.0). TeraSystemsAI LLC.

Contents

List of Figures

List of Tables

Executive Summary

Distribution shift is an assurance problem, not only a monitoring problem

A deployed model operates in a world that continues to change. Populations change, upstream systems are modified, user behavior evolves, sensors are replaced, policies alter incentives, and the prevalence or meaning of outcomes can shift. These changes may be benign, material, abrupt, gradual, local to a subgroup, or difficult to observe until outcome evidence arrives. Research on dataset shift shows that predictive accuracy and uncertainty behavior can deteriorate under changing conditions. Deployment-monitoring research also shows that labels may be delayed or costly, while unlabeled drift signals alone do not establish current task performance.[7–9] This guide therefore treats distribution shift as a continuous assurance challenge rather than a one-time monitoring task. Its central proposition is:

Central proposition

A drift signal indicates that monitored conditions differ from a defined reference. It does not, by itself, establish that the model failed. Trustworthy deployment requires determining whether the observed change is consequential through evidence.

The resulting TeraSystemsAI Distribution Shift Assurance Lifecycle is:

The lifecycle is intentionally not equivalent to "drift detected $\rightarrow$ retrain." DETECT establishes a reproducible observation. ASSESS determines what changed, how credible the signal is, and whether the change is consequential to the intended use. RESPOND selects a proportionate intervention or a justified no-action decision. VERIFY tests whether the resulting operating state satisfies pre-defined acceptance criteria. DOCUMENT preserves the evidence chain, authority, uncertainty, alternatives, and residual risk.

Governance is cross-cutting. TeraSystemsAI's public accountability framework states that AI should not be the last responsible actor and that final authority

remains with humans or institutions.[12] Automated monitors may detect, summarize, and recommend, but material deployment decisions require named ownership and sufficient human competence, authority, and review capacity. The guide is designed for AI/ML engineers, model-risk teams, product owners, operations leaders, internal audit and compliance functions, and organizations deploying AI in consequential settings. It is not a universal statistical prescription. No single drift metric, threshold, cadence, reference window, or response rule is correct across all systems. Assurance design must be proportional to consequence, observability, data availability, temporal dependence, system architecture, and regulatory context.

Key outcomes

A team applying this guide should be able to answer six questions with evidence:

1. What changed? Which monitored signal, distribution, context, subgroup, or relationship departed from the reference condition?
2. Does it matter? Is there evidence that the change affects reliability, calibration, safety, fairness, security, or the intended use?
3. Who decides? Which named role owns the assessment, response, and acceptance of residual risk?
4. What was done? Was the response observation, repair, recalibration, restriction, human escalation, rollback, retraining, or another justified intervention?
5. Did it work? What verification evidence shows that the system returned to an acceptable operating state?
6. What remains unknown? Which labels, subgroups, causal mechanisms, dependencies, or time periods remain insufficiently observed?

NIST's 2026 report on deployed-AI monitoring similarly emphasizes that predeployment evaluation is not sufficient for real-world operation and that post-deployment monitoring is needed to validate expected behavior, identify unforeseen outputs, and provide visibility into unexpected consequences.[3] This

guide narrows that broad challenge to distribution shift and adds an evidence-governed response structure derived from TeraSystemsAI's public research and accountability principles.[10, 11]

How to Use This Guide

How to use this technical report

This report is written as a deployment guide rather than a compliance checklist. Organizations should adapt it to the actual risk, architecture, and evidence availability of each AI system. The lifecycle can be implemented at model level, pipeline level, application level, or portfolio level, provided responsibility and evidence boundaries remain explicit.

Table 1: Primary audiences and intended uses

Audience	Primary question	Recommended focus
AI/ML engineering	What signals and metrics should we monitor?	Sections 3, 7, 13; Appendices A-B
Model risk / validation	Is detected change consequential?	Sections 8, 10, 14
Product / operations	What action should follow an alert?	Sections 9, 12, 15
Audit / compliance	Can the decision be reconstructed?	Sections 11, 12, 17; Appendices B-D
Executives / risk owners	Who owns residual risk and escalation?	Executive Summary, Sections 5, 12, 17

Three reading paths

Engineering path. Start with Sections 3-4, then use the lifecycle Sections 7-11 and the measurement framework in Section 13. The goal is to translate sta-

tistical monitoring into operational decisions without assuming that every alert represents failure. Governance path. Start with Sections 5-6 and 12, then use the evidence sufficiency model in Section 14 and the external-framework mapping in Section 17. The goal is to ensure that automated detection never erases human authority or institutional ownership. Implementation path. Start with Section 15 and the worked example in Section 16, then adapt the appendices into operating templates. The appendices are intentionally designed to be copied into internal procedures, tickets, model cards, risk registers, or monitoring runbooks.

Important implementation rule

This report distinguishes change detection from performance verification. A statistical detector can indicate that recent data differ from a reference distribution. It cannot, by itself, prove that task performance, calibration, safety, fairness, or decision quality has deteriorated. Where outcome evidence is unavailable, delayed, selectively observed, or itself uncertain, the assurance record should state that limitation explicitly.

1 Purpose and Scope

1.1 Purpose

The purpose of Distribution Shift Assurance is to provide a disciplined method for managing uncertainty introduced by changing deployment conditions. The guide focuses on systems that use statistical or machine-learning models to support predictions, rankings, recommendations, retrieval, classification, generation, or decision support. It also applies to hybrid systems that combine models with deterministic rules, retrieval components, tools, or human review.

The guide extends a broader TeraSystemsAI research posture centered on evidence, visible uncertainty, reliability under degraded conditions, and preserved human authority. TeraSystemsAI's publication program describes trustworthiness in terms of evidence strength, uncertainty, provenance, limitations, calibration, failure modes, abstention, and accountability.[11] The company's document-intelligence work similarly argues that structured outputs

should preserve the chain between evidence and downstream decision.[14] Distribution Shift Assurance applies the same logic over time: when operating conditions change, the organization should preserve the evidence connecting the observed signal, the assessment, the authorized decision, the intervention, and the verified result.

1.2 In scope

This guide addresses:

- changes between defined reference and deployment conditions;
- batch, temporal, and sequential drift-monitoring design;
- performance, calibration, uncertainty, abstention, retrieval, and subgroup impacts;
- delayed, selectively observed, or unavailable outcome labels;
- feedback loops and deployment-induced changes in the observed data;
- escalation and response governance;
- post-intervention verification and residual-risk assessment;
- documentation, provenance, and auditability;
- implementation profiles proportional to consequence, including retrieval and agentic systems;
- high-level mapping to NIST, ISO, and EU regulatory concepts.

1.3 Out of scope

This guide does not define a universal drift metric, certify a system as safe, prescribe legal compliance, or replace domain-specific validation. It does not provide proprietary TeraSystemsAI implementation details. It also does not treat all post-deployment failures as distribution shift. Software defects, cyber incidents, data corruption, human-process failures, prompt attacks, dependency outages,

and policy violations may produce similar symptoms and require different root-cause analysis. The guide does not claim that monitoring can prove absence of failure, establish causality from observational signals alone, or guarantee safety.

1.4 Intended audience

The intended audience includes organizations that develop, deploy, operate, validate, audit, or govern AI systems. The guide is especially relevant where incorrect, unstable, or poorly understood model behavior could create operational, financial, safety, rights, or reputational consequences.

2 Foundational Assumptions

Distribution Shift Assurance begins from six assumptions.

2.1 Deployment conditions are dynamic

A development dataset captures a finite set of conditions. Deployment introduces time, feedback, external actors, new populations, evolving incentives, software changes, and operational context. A system that remains technically available can therefore become less reliable even when its code has not changed.

2.2 Change is not automatically failure

A changed distribution may be expected, harmless, or even beneficial. Seasonal traffic, an intentionally expanded customer population, a new sensor with improved precision, or a change in business mix may all produce detectable distributional differences without invalidating the model. Conversely, harmful change can occur in ways that simple input monitoring does not detect.

2.3 Observability is incomplete

Many systems do not receive immediate labels. Fraud outcomes may take weeks to confirm; healthcare outcomes may be delayed; human review may be sampled; some downstream effects may never be fully observed. Observed labels can also be selective: the system or a human reviewer may influence which cases receive verification. Monitoring must therefore distinguish direct outcome evidence from proxy signals, account for label-selection mechanisms where material, and preserve uncertainty about what remains unobserved.

2.4 Response is a governance decision

Statistical evidence can inform action, but it does not own the action. TeraSystemsAI's accountability framework defines a boundary in which AI may support pattern recognition and risk estimation while final authority remains with humans and institutions.[12] Distribution-shift controls should preserve the same boundary.

2.5 Evidence must survive the lifecycle

A useful alert is not merely a dashboard event. It should preserve enough information to reconstruct what changed, which model and monitor versions were affected, what evidence was available, what remained unknown, who decided, what intervention followed, and whether the intervention worked. This is consistent with TeraSystemsAI's broader emphasis on provenance and verifiability.[14]

2.6 Deployment can change the data-generating process

A deployed model can influence which actions are taken, which cases are reviewed, which labels become observable, and how users behave. This creates feedback and selection effects that can resemble or amplify distribution shift. Monitoring should therefore record material policy, routing, review, and automation changes alongside model and data changes.[19]

Constraint-aware posture

TeraSystemsAI's public philosophy emphasizes explicit tradeoffs rather than universal promises.[13] Distribution-shift assurance follows the same rule: tighter detection sensitivity may increase false alarms; broader coverage may increase monitoring cost; faster response may reduce time for causal diagnosis. These are design choices that should be explicit rather than hidden.

3 Distribution Shift as an Operational Risk

Let $P_{\text{ref}}(X, Y)$ denote a reference joint distribution used for development, validation, or a validated deployment period, and let $P_t(X, Y)$ denote the distribution encountered during a later deployment window t. A general distribution shift occurs when:

$$P_t(X, Y) \neq P_{\text{ref}}(X, Y). \tag{1}$$

This definition is intentionally broad. The operational consequences depend on what changed. The notation in this section is a supervised-learning abstraction, not a requirement that every deployed AI system have a single observed label Y. For retrieval, generative, multimodal, and agentic systems, teams should define analogous reference and deployment distributions over decision-relevant observables such as prompts, retrieved evidence, tool states, outputs, human interventions, and downstream outcomes.

3.1 Input shift and covariate shift

A broad input shift occurs when the marginal distribution of observed inputs changes:

$$P_t(X) \neq P_{\text{ref}}(X). \tag{2}$$

Examples include new user populations, changes in image acquisition, geographic expansion, altered transaction mix, or revised upstream data collection. The narrower term *covariate shift* is commonly used when $P(X)$ changes while the predictive conditional $P(Y \mid X)$ is assumed to remain stable. That invariance is an assumption to be examined, not inferred from an input-drift alert. Input shift may therefore be consequential, benign, or accompanied by other forms of shift.

3.2 Outcome prevalence and label shift

Outcome prevalence changes when:

$$P_t(Y) \neq P_{\text{ref}}(Y). \tag{3}$$

In the narrower label-shift setting, the label marginal changes while $P(X \mid Y)$ is assumed stable.[20] A fraud rate, disease prevalence, default rate, defect rate, or demand category may change in ways that alter calibration, decision thresholds, or workload even when ranking performance remains useful. A change in $P(Y)$ alone does not establish that the label-shift assumption holds.

3.3 Conditional or concept shift

A conditional or concept shift occurs when the predictive relationship changes:

$$P_t(Y \mid X) \neq P_{\text{ref}}(Y \mid X). \tag{4}$$

This can arise when incentives change, adversaries adapt, policies alter behavior, causal relationships shift, or the meaning of a feature evolves. Conditional shift is often harder to establish without outcome evidence because input-only monitoring can remain stable while $P(Y \mid X)$ changes. These categories are not mutually exclusive, and a drift alert does not uniquely identify the underlying mechanism. Input, prevalence, and conditional changes can occur together; with incomplete labels, several mechanisms can be observationally compatible with the same monitoring signal.

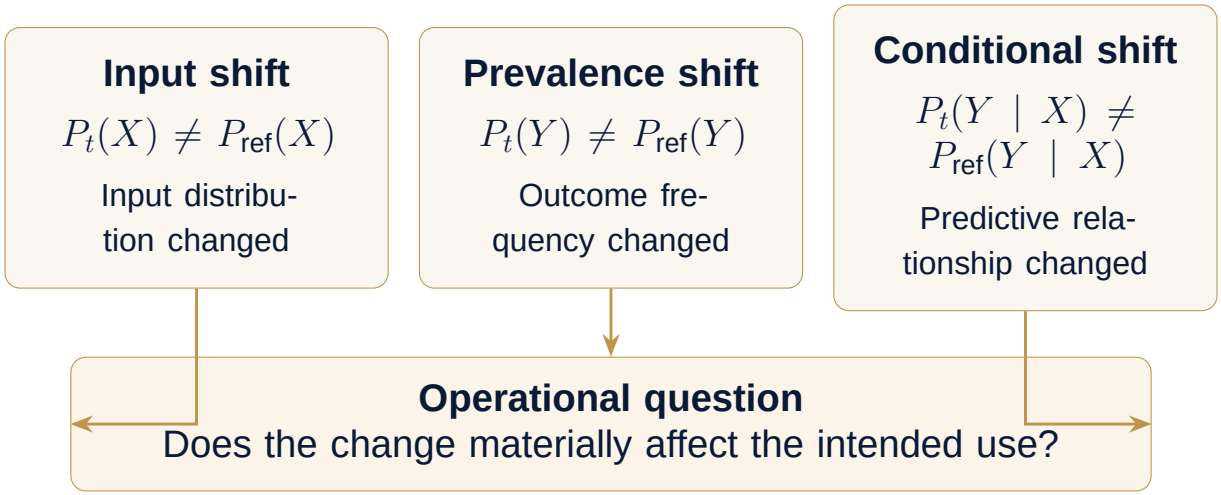

Figure 1: Distribution-shift categories are diagnostically useful, but the assurance decision depends on operational consequence.

3.4 Benign versus consequential shift

The distinction between benign and consequential shift is central to this guide. A shift is consequential when credible evidence indicates that changed conditions materially affect a requirement of the intended use, such as predictive performance, calibration, safety margin, fairness across relevant groups, system reliability, security posture, or a policy constraint. A shift can be large in statistical distance yet operationally benign, or statistically subtle yet consequential for a small but important subgroup.

4 Why Shift Can Fail Quietly

A production system can remain healthy at the infrastructure layer while becoming unhealthy at the decision layer. This separation is why distribution shift is difficult to govern.

4.1 Operational health versus predictive health

Latency, uptime, throughput, error rates, and API availability may remain normal. None of those metrics directly establish that the model is still making reliable predictions. A service can therefore satisfy conventional observability dashboards while its decision quality declines.

4.2 Delayed ground truth

Outcome evidence often arrives after the decision. Ginart, Zhang, and Zou's MLDemon work explicitly addresses deployment monitoring where labels are costly and only a limited amount can be acquired on demand.[8] This creates an evidence gap: unlabeled signals may indicate change before the organization can determine whether accuracy actually deteriorated.

4.3 Calibration and uncertainty under shift

Ovadia et al. evaluated predictive uncertainty under dataset shift and found that common methods can lose calibration as shift severity increases.[7] The operational implication is not that every shifted model becomes overconfident, but that confidence itself must be treated as a monitored property rather than an immutable guarantee.

4.4 Subgroup-local degradation

Aggregate metrics can hide localized failure. A model may remain stable overall while deteriorating for a region, product line, device type, language, age band, or other operationally relevant group. Monitoring plans should therefore define subgroup slices based on risk and use context, not only convenience.

4.5 Expected contextual variation

Cobb and Van Looveren show why context matters for drift detection: recent deployment data can legitimately differ from a historical reference because of

known contextual variables.[9] A monitor that ignores context may produce excessive alerts and train operators to disregard them. Context-aware baselines can reduce that failure mode.

4.6 Upstream changes masquerading as model problems

Schema changes, missing-value behavior, unit conversions, sensor firmware, preprocessing code, database joins, retrieval indexes, tool APIs, or provider-side model changes can alter the system's effective input or decision pathway. The correct response may be to repair a dependency or pipeline rather than retrain the predictive model. Root-cause diagnosis therefore precedes model intervention.

4.7 Feedback and selective labels can distort verification

Deployment changes what is observed. A fraud model may route only high-scoring cases to investigators; a recommendation system may influence what users see; a human reviewer may verify only uncertain cases. The resulting labels are not necessarily a representative sample of the deployment population. Apparent stability or degradation can therefore reflect the observation policy as well as the underlying task distribution.[19]

Anti-pattern: automatic retraining on drift

Retraining can encode a transient anomaly, mask a data-pipeline defect, remove a useful baseline, or introduce new regression risk. Detection alone must not automatically authorize retraining.

5 The TeraSystemsAI Assurance Principle

The current TeraSystemsAI distribution-shift article frames the essential distinction in plain language: a drift signal indicates changed conditions, not proof of

model failure. This report formalizes that statement into an assurance principle.

Assurance Principle

A distribution-shift signal is evidence that monitored conditions differ from a defined reference. It becomes evidence of model or system degradation only when additional evidence connects that difference to a material requirement of the intended use.

This principle has four consequences.

5.1 Detection is observational

A detector produces an observation about the data, model outputs, uncertainty, or operating context. The observation may be statistically strong while still being operationally ambiguous.

5.2 Assessment is decision-relevant; causal claims require stronger evidence

The assessment asks what changed, whether the change is expected, which mechanisms could plausibly connect it to model behavior, and what decision-relevant evidence is available. An operational decision can be justified without a complete causal model, but the record should not describe a mechanism as causal unless the evidence supports that claim. Correlation, temporal coincidence, and a drift alert are not causal proof.

5.3 Authority remains named

TeraSystemsAI's accountability framework asserts that AI should not be the final responsible actor.[12] In Distribution Shift Assurance, automated systems can flag, summarize, rank, and recommend. Material actions such as restricting use, rolling back, retraining, or accepting residual risk require an accountable role defined by organizational policy.

5.4 Evidence is preserved

TeraSystemsAI's document-intelligence guidance emphasizes preserving provenance between structured values and original evidence.[14] The same idea applies here: every material shift case should preserve the chain from signal to assessment to intervention to verification.

What this principle prevents

It prevents two opposite errors:

- Overreaction: treating every statistical difference as system failure and triggering unnecessary interventions.
- Underreaction: dismissing meaningful changes because aggregate accuracy has not yet visibly declined or because labels are delayed.

The assurance process exists to navigate between those errors.

6 Distribution Shift Assurance Lifecycle

The lifecycle contains five operational stages, with governance and evidence preservation spanning all five.

Table 2: Lifecycle stages, outputs, and decision gates

Stage	Primary output	Gate
DETECT	Reproducible signal with provenance	Is there credible evidence of changed conditions?
ASSESS	Impact assessment with uncertainty	Is the change consequential to the intended use?
RESPOND	Authorized intervention or justified no-action	Is the response proportionate to evidence and risk?
VERIFY	Post-intervention validation evidence	Has acceptable operation been restored or bounded?
DOCUMENT	Complete assurance record	Can the decision be independently reconstructed?

6.1 Cross-cutting governance

Governance determines who owns the monitoring policy, who investigates, who may approve an intervention, who accepts residual risk, and what evidence must be retained. This is analogous in spirit to NIST's treatment of GOVERN as a cross-cutting AI RMF function, while the TeraSystemsAI lifecycle remains independently defined.[1]

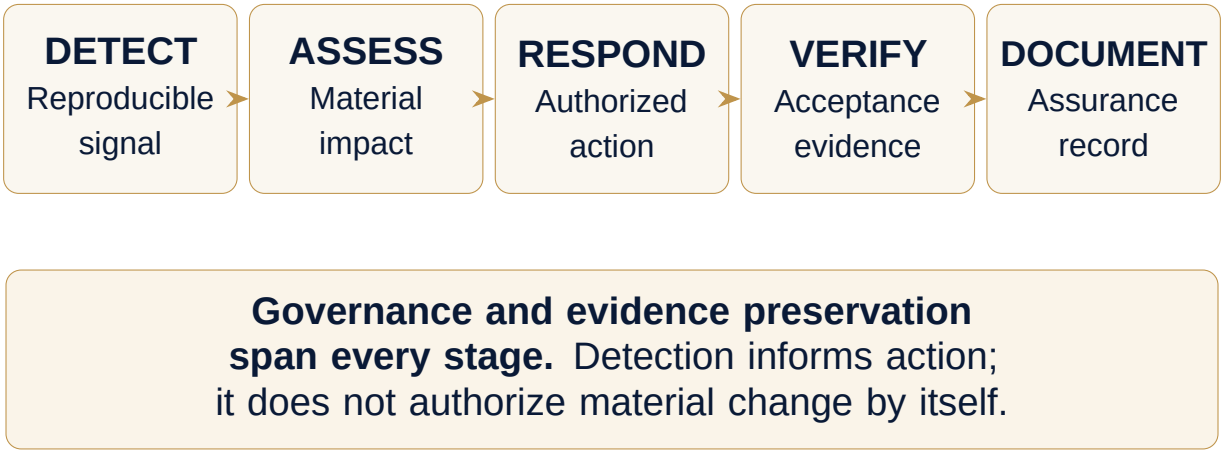

Figure 2: TeraSystemsAI Distribution Shift Assurance Lifecycle. The visual is a conceptual guide, not a statistical algorithm.

6.2 Closed-loop operation

The lifecycle is iterative. Verification can identify new evidence that changes the assessment. Documentation can reveal repeated patterns that justify better baselines or stronger controls. A response can create a new validated reference condition. The system therefore operates as a learning assurance process, not a one-time checklist.

7 DETECT

Establish Evidence of Changed Conditions

The purpose of DETECT is to identify evidence that deployment conditions differ from a defined reference condition. Detection should be reproducible, attributable to a model and data version, and designed around the system's actual risk surface.

7.1 Define and govern the reference condition

A reference condition can be the training distribution, validation set, launch cohort, fixed validated deployment window, seasonally matched peer window, or another justified operating period. The choice must be explicit because every drift result is relative to a reference. Rolling or adaptive references require additional care: if they update automatically, a degraded state can be absorbed into the baseline and become harder to detect. Any reference-update policy should therefore be versioned, reviewable, and subject to change control. A robust reference record should include:

- model and preprocessing versions;
- data-source versions and schemas;
- feature definitions and transformations;
- time window, geography, population, and relevant context;

- baseline task performance and calibration where labels exist;
- subgroup metrics where material;
- known exclusions and missing-data behavior.

7.2 Select monitored signals

Monitoring may include feature distributions, embeddings, output scores, predicted-class proportions, uncertainty measures, abstention rates, retrieval characteristics, data-quality checks, subgroup mix, system policy violations, and eventually ground-truth outcomes. TeraSystemsAI's public research program explicitly identifies drift/anomaly detection, continuous monitoring, calibration, and human escalation as research areas.[10]

7.3 Select statistical methods proportionately

Possible methods include univariate tests, population-stability indices, divergence measures, maximum mean discrepancy, classifier-based two-sample tests, embedding-distance measures, change-point detection, and model-based monitors. The report does not designate a universal best method. Choice depends on data type, dimensionality, sample size, temporal dependence, representation quality, false-alert tolerance, context, and operational consequence. High-dimensional tests can detect change without explaining which decision-relevant mechanism changed, so localization and impact assessment remain separate tasks.[21]

7.4 Define thresholds and alert logic before incidents

Thresholds should be documented before deployment or before the monitoring rule is activated. They may be static, adaptive, context-specific, or risk-tiered. The record should explain whether a threshold represents statistical significance, effect size, operational tolerance, or an escalation trigger. Repeated or sequential testing requires explicit alert logic because repeatedly applying a fixed-sample test can inflate false alerts. Window overlap, autocorrelation, and repeated comparisons should be considered in the monitoring design.[22]

7.5 Preserve alert provenance

Every alert should carry a unique identifier and enough metadata to reproduce the result. At minimum: system and model version, monitor version, reference definition, observation window, sample counts, metric or test, threshold, observed value, affected feature or slice, time generated, source-data lineage, and relevant dependency versions. Retrieval and agentic systems should also preserve corpus/index, prompt/policy, tool, and provider-model identifiers when those components can change system behavior.

Detection output is a hypothesis trigger. The alert should state what the monitor observed, not what it has not established. Prefer “feature distribution changed beyond threshold” over “model accuracy degraded” unless outcome evidence supports the latter claim.

8 ASSESS

Determine Whether the Change Matters

ASSESS converts a drift signal into a decision-relevant evidence record. The objective is to determine what changed, whether the change is expected, and whether the system remains fit for its intended use.

8.1 Confirm the signal

First determine whether the alert is reproducible and not caused by monitoring defects, duplicate data, pipeline errors, small samples, stale reference windows, or known seasonal effects. A false alert is itself useful evidence about monitoring design.

8.2 Diagnose the likely source

Candidate causes include population mix, product changes, user behavior, adversarial adaptation, upstream data processing, policy changes, sensor changes, market regimes, external events, or model feedback loops. Multiple causes can coexist.

8.3 Connect the change to the intended use

The assessment should explicitly state which requirement could be affected. Examples include sensitivity above a safety threshold, calibration within a tolerance band, false-positive burden, subgroup parity, retrieval coverage, abstention rate, latency constraint, or a policy invariant.

8.4 Acquire outcome evidence where feasible

If labels are delayed, acquire a targeted sample where the stakes justify it. MLDemon demonstrates the value of combining unlabeled monitoring with selectively acquired labels under cost constraints.[8] In practice, organizations can use sampled human review, adjudication, delayed outcomes, or other credible reference outcomes. The sampling policy matters: labels concentrated only on alerts, positives, or uncertain cases can create selection bias. Where practical, retain a probability-based audit sample or another design that supports interpretation of population-level performance.[19]

8.5 Assess calibration and uncertainty separately from accuracy

A model may retain ranking ability while confidence becomes less reliable. Conversely, a confidence distribution may change without material decision impact. Evaluate the metrics that matter to the decision policy rather than assuming a single performance number captures reliability.

8.6 Assess relevant subgroups

The assessment should include slices tied to known risk, operating context, and affected populations. If a drift signal is driven by a population change, aggregate metrics are especially likely to hide local effects.

8.7 State assessment confidence and unresolved uncertainty

The result should separate conclusion from evidence quality. “No evidence of degradation” is not equivalent to “evidence of no degradation.” The assessment should state which claims are observed, inferred, or still unresolved. When labels are absent, delayed, selectively observed, or too sparse for a required subgroup, the assurance record should state the resulting uncertainty explicitly.

9 RESPOND

Select a Proportionate Intervention

RESPOND chooses and authorizes an intervention based on the assessed consequence, confidence in the evidence, and risk of both action and inaction.

9.1 Response options

A response can include:

- continued observation with increased sampling;
- investigation of upstream data or software;
- repair of schema, preprocessing, retrieval, or data-quality defects;
- recalibration or threshold adjustment;

- human-review expansion;
- restriction to a validated population or operating range;
- abstention or fallback to a safer deterministic process;
- rollback to a prior model or configuration;
- retraining or redesign;
- temporary suspension where consequence and uncertainty justify it.

9.2 Do not default to retraining

Retraining is one possible response, not the definition of drift management. It should be used when evidence indicates that the model or decision rule should change and when candidate training data are sufficiently representative, attributable, and governed. A pipeline bug should be repaired; expected seasonality may require a contextual baseline; prevalence change may require recalibration or threshold adjustment; selective labels may require a different data-acquisition design before any retraining is defensible. For consequential systems, retraining should create a new candidate version, not silently overwrite the validated baseline.

9.3 Use response levels

Organizations can define response levels such as: These levels are illustrative. Each organization should set its own thresholds and authority matrix.

Table 3: Illustrative response levels

Level	Evidence posture	Typical action	Authority
R0	Signal within expected variation	Continue monitoring	Operator
R1	Confirmed change, low consequence	Increase observation / review	System owner
R2	Plausible material impact	Restrict, recalibrate, investigate	Risk owner + technical owner
R3	Strong evidence of harmful degradation	Rollback, suspend, retrain, redesign	Designated approval authority

9.4 Record the alternative considered

A mature assurance record should document why a selected response was preferred over reasonable alternatives. This prevents hindsight bias and makes the decision defensible after an incident.

10 VERIFY

Demonstrate That the Response Worked

Verification answers a different question from detection: after intervention, is there evidence that the system has returned to an acceptable operating state, or that the remaining risk is bounded and explicitly accepted?

10.1 Define acceptance criteria before verification

Acceptance criteria should be tied to intended-use requirements and defined before examining the final verification result. Examples include performance

bands, calibration limits, subgroup constraints, risk-coverage behavior for abstaining systems, restored data quality, no regression on protected scenarios, or successful fallback operation. Reduced alert frequency alone is not sufficient if it is achieved by weakening the detector.

10.2 Compare against more than one baseline

Where possible, compare the post-intervention system with the pre-shift validated condition, the degraded condition, and any candidate replacement. This helps determine whether apparent improvement is meaningful or simply reflects a new data mixture.

10.3 Run regression and independence checks

Any intervention can create new failures. Verification should include relevant holdout or time-separated data, stress cases, edge cases, subgroup checks, policy invariants, and operational tests. Where feasible, verification evidence should be independent of the data used to select the intervention. For systems using uncertainty, abstention, retrieval, or human escalation, verify those behaviors as first-class outputs rather than treating them as secondary diagnostics.

10.4 Assess residual risk

A response may improve performance without restoring the original state. The verification record should describe remaining uncertainty, unobserved outcomes, population gaps, and any temporary restrictions still in effect.

10.5 Independent review where warranted

Higher-consequence systems may require a reviewer separate from the person who implemented the response. Separation reduces confirmation bias and aligns with the broader TeraSystemsAI accountability posture of explicit roles and review checkpoints.[12]

Verification rule

An intervention is not complete when code is deployed. It is complete when evidence shows the resulting system satisfies defined acceptance criteria or when residual risk is explicitly accepted by the authorized role.

11 DOCUMENT

Preserve the Assurance Record

Documentation turns monitoring into accountability. A dashboard may show that an alert occurred; an assurance record explains what the alert meant, what was decided, and why.

11.1 Minimum case record

For each material case, record:

- alert ID and timestamps;
- affected application, model, and model version;
- monitor, metric/test, threshold, and observed value;
- reference definition and observation windows;
- source-data, preprocessing, retrieval, tool, and other material dependency versions;
- affected features, contexts, routes, or subgroups;
- assessment findings, alternative explanations, and known uncertainty;
- outcome evidence, label-acquisition method, and material selection limitations;
- decision authority and reviewers;

- intervention and alternatives considered;
- verification method and result;
- residual risk and follow-up actions;
- links to tickets, model cards, risk register, and incident records.

11.2 Distinguish observed, inferred, and decided

The record should separate facts observed by monitoring, interpretations inferred during assessment, and decisions made by authorized humans. This mirrors the provenance distinction in TeraSystemsAI's reliable-extraction article, where directly observed values should remain distinguishable from inferred or transformed values.[14]

11.3 Retention and accessibility

Retention periods depend on organizational, contractual, legal, and sector requirements. Regardless of duration, the evidence should remain interpretable without requiring the original operator's memory. Metadata, model identifiers, and data lineage are therefore not optional implementation details.

11.4 Version control

Every revision to monitoring policy, thresholds, reference windows, or response authority should be versioned. A post-incident reviewer must be able to determine which rule was active at the time of the decision.

12 Governance and Accountability

Distribution-shift assurance fails when alerts exist but authority is ambiguous. Governance defines who may observe, interpret, authorize, override, and ac-

cept residual risk.

12.1 Four-role model

This guide adapts TeraSystemsAI's public accountability architecture into a distribution-shift context.[12]

Table 4: Responsibility boundaries for distribution-shift assurance

Role	Assurance function	Boundary
AI / monitor	Detect patterns, estimate risk, summarize evidence	No final authority; no unilateral high-impact intervention
Human operator / reviewer	Interpret context, challenge evidence, approve routine actions	Must be able to override automated recommendation
Policy	Define thresholds, escalation rules, evidence requirements	Deterministic and versioned where feasible
Institution	Own deployment decision, residual risk, legal/accountability obligations	Cannot transfer accountability to the model or monitor

12.2 Separation of detection and authorization

Where the consequence of intervention is material, the component that detects drift should not also have unrestricted authority to retrain, redeploy, or change decision thresholds. This separation creates an intentional control point for evidence review.

12.3 Escalation design

Escalation should define both trigger and destination. "Human review" is insufficient if no role, response time, evidence package, authority, competence,

or capacity is specified. Escalation policy should state who receives the case, what evidence they receive, what they can override or stop, and when further authority is required. A nominal human-in-the-loop control that cannot meaningfully challenge the system is not an adequate assurance boundary.

12.4 Accountability under uncertainty

The absence of complete evidence does not remove responsibility. It changes the decision posture. A risk owner may choose continued operation with enhanced controls, restrict use, or suspend the system. The key requirement is that the uncertainty and acceptance decision remain explicit.

13 Measurement Framework

Measurement should answer four distinct questions: Did monitored conditions change? Did system behavior change? Did decision quality change? How certain are we about each conclusion? A single metric rarely answers all four.

13.1 Data-distribution measures

For numeric features, teams may use tests or distances such as Kolmogorov–Smirnov statistics, Wasserstein distance, Jensen–Shannon divergence, population stability index, or other methods appropriate to scale and sample size. For multivariate or embedding data, approaches can include maximum mean discrepancy, classifier-based two-sample tests, or learned representations. Rabanser et al. show that detector performance depends materially on representation and test choice; a drift detector should therefore be validated for the data regime in which it will operate.[21]

Jensen–Shannon divergence between distributions P and Q can be written as:

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M), \qquad M = \frac{P+Q}{2}. \tag{5}$$

The formula is illustrative, not a universal recommendation. Distribution dis-

tance measures differences in distributions; it does not directly measure decision harm. Practical estimates also depend on representation, support, smoothing, binning, and finite-sample behavior; these choices should be part of monitor validation.

13.2 Model-output and system-behavior measures

Useful signals include score distributions, entropy, class proportions, abstention rates, prediction-set sizes, retrieval coverage, evidence availability, tool-call failure rates, disagreement among models, and human-escalation rates. A shift in these quantities can be diagnostically useful before labels arrive, but each signal should be tied to a defined interpretation and intended use.

13.3 Outcome and task measures

Where credible labels or adjudicated outcomes exist, evaluate metrics tied to the intended use: sensitivity, specificity, precision, recall, ranking metrics, cost-weighted error, subgroup error, calibration, risk-coverage behavior, or domain-specific outcomes. Metrics should reflect real decision consequences rather than benchmark convenience. If labels are selectively observed, population-level conclusions require an explicit sampling or correction strategy.[19]

13.4 Calibration and uncertainty measures

Calibration evaluates whether stated probabilities or confidence scores correspond to observed frequencies. Expected calibration error, Brier score, log loss, reliability diagrams, and coverage metrics can be useful depending on the output type. These measures have different statistical properties and should be interpreted with sample size, binning, censoring, and label availability in mind. Ovadia et al. demonstrate why uncertainty behavior under dataset shift deserves explicit evaluation.[7]

13.5 Context, seasonality, and operating regime

A reference should account for expected contextual variation. Comparing a holiday period against an ordinary-week baseline may create predictable differences. Context-aware monitoring can reduce false alarms where observed covariates are known to influence the data distribution.[9] Context should not be used to explain away a shift without checking whether the context itself changes decision risk.

13.6 Sequential monitoring, dependence, and multiplicity

Production observations are often temporally dependent and monitors run repeatedly. Fixed-sample tests applied over overlapping windows can produce misleading alert rates. Monitoring design should document cadence, window overlap, dependence assumptions, multiple-testing policy, and reset behavior after an alert or intervention. Sequential methods can be appropriate when their assumptions and error guarantees match the data stream.[22]

13.7 Threshold design

Thresholds should consider effect size, sample size, detection delay, alert frequency, cost of missed detection, cost of false escalation, and the time needed to obtain outcome evidence. A p-value alone is not an operational threshold. Where a threshold is tuned after observing an incident, the record should state that the rule changed and should not retroactively present the new threshold as pre-specified.

Table 5: Measurement layers and what they can establish

Layer	Can support	Cannot establish alone
Input distribution	Monitored inputs differ from reference	Model or system performance deteriorated
Model outputs / behavior	Prediction or workflow behavior changed	Root cause or outcome harm
Uncertainty / calibration proxies	Confidence-related behavior changed	True calibration without credible outcomes
Outcome / adjudicated labels	Task performance changed on observed cases	Population performance if labels are selectively observed
Operational metrics	Service health and availability	Predictive or decision reliability
Human-review metrics	Review burden and escalation behavior	Independent correctness without a credible reference outcome

14 Evidence Sufficiency

A defining feature of this guide is an explicit E0–E4 evidence-sufficiency model. The model is proposed by TeraSystemsAI as a practical communication device for a specific assurance case. It is not an external standard, regulatory classification, probability scale, certification grade, or universal maturity model. Different hazards, subgroups, or claims within the same deployment can occupy different levels at the same time.

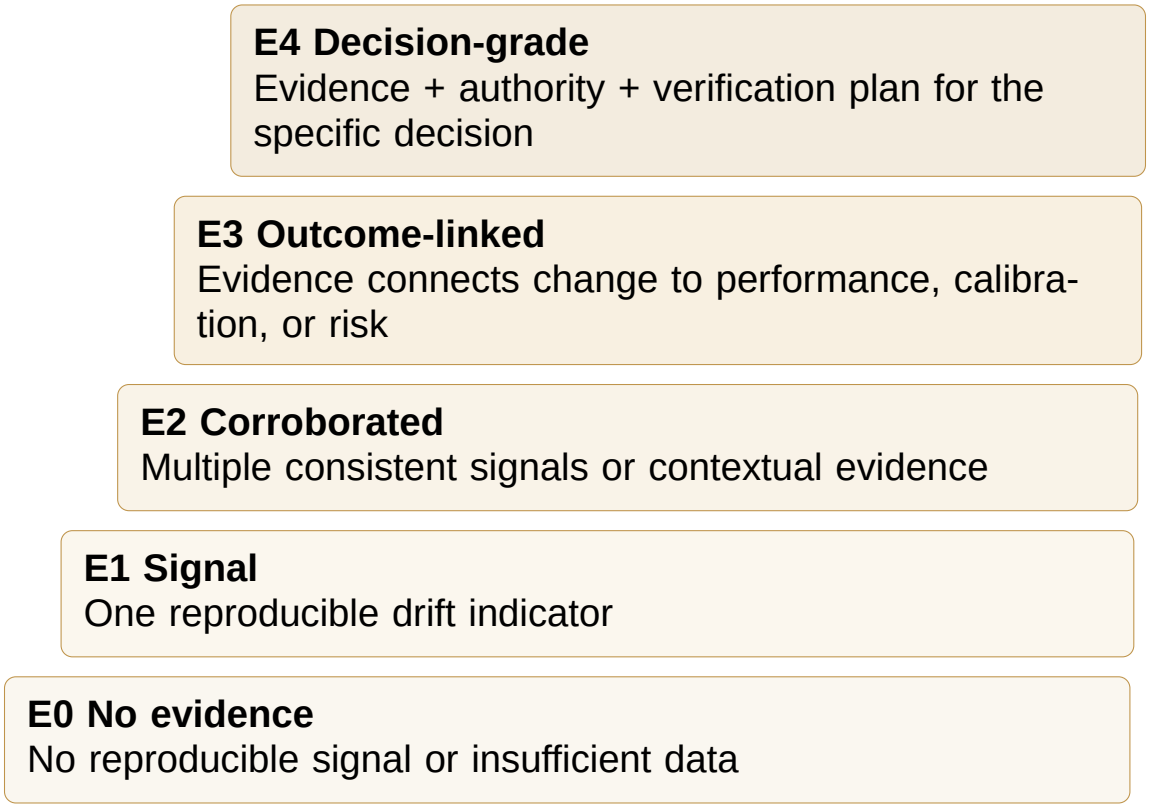

Figure 3: Proposed TeraSystemsAI E0–E4 evidence-sufficiency model. The levels communicate an assurance case; they are not probabilities or certification grades.

14.1 E0 - No usable evidence

Data are insufficient, monitoring is unavailable, or the signal cannot be reproduced. The correct conclusion is uncertainty, not stability.

14.2 E1 - Signal evidence

A detector identifies a statistically or operationally significant change. This establishes changed conditions relative to the reference, but not degradation.

14.3 E2 - Corroborated evidence

Multiple indicators agree, a known contextual event explains the shift, or independent evidence sources show consistent changes. Correlated monitors that reuse the same features, window, or representation should not automatically be counted as independent corroboration. E2 strengthens the case that a change is real or interpretable, but it does not establish task degradation.

14.4 E3 - Outcome-linked evidence

Current labels, adjudicated review, calibration checks, subgroup evaluation, or other credible evidence connect the shift to material model behavior or decision quality. The record should state how outcomes were obtained and whether selective observation, censoring, delay, or sampling limitations constrain the conclusion.

14.5 E4 - Decision-grade assurance package

E4 adds decision readiness to the evidence record. The evidence is sufficient for the consequence of the decision, the responsible authority is identified, material alternatives are considered, the intervention is justified, and verification criteria are defined. E4 does not mean certainty, zero residual risk, or that the underlying evidence is stronger in every statistical sense. It means the evidence and governance package are adequate for the specific decision.

Proportionality rule

Evidence sufficiency is claim- and consequence-dependent. An E2 record may justify increased observation in a low-consequence application but be insufficient to continue autonomous operation in a safety-critical setting. A higher E-level must never be used as a substitute for the domain-specific acceptance criteria that actually govern the deployment.

15 Implementation Profiles

Profiles adapt the lifecycle to different deployment contexts. They are starting points, not compliance classes.

Table 6: Illustrative implementation profiles

Profile	Monitoring	Evidence target	Response posture	Verification
Low-consequence automation	Key inputs/outputs; periodic outcome sample	E1-E2 for routine alerts	Observe, investigate, repair	Periodic KPI check
Enterprise decision support	Inputs, outputs, calibration, business outcomes, slices	E2-E3 for material change	Human review, recalibration, restriction	Task + business metrics
High-consequence / regulated	Multi-layer monitoring, traceable logs, subgroup and outcome evidence	E3-E4 for consequential action	Named authority; fallback; rollback; formal change control	Independent or second-line review where appropriate
Delayed-ground-truth	Proxy signals plus active label acquisition	Explicit uncertainty until labels arrive	Increase sampling; restrict automation if risk warrants	Delayed outcome reconciliation
Continuous / adaptive	Online context, model/version changes, guardrails, performance	E2-E4 depending consequence	Tight change governance; frozen rollback point	Continuous + scheduled validation
Generative / retrieval / agentic	Prompts, output characteristics, corpus/index, tools, policies, provider/-model versions	Claim-specific; outcome evidence where feasible	Restrict tools, pin versions, repair retrieval, escalate, or roll back	Scenario tests + evidence/grounding + tool-path verification

15.1 Low-consequence profile

Use lightweight monitoring where incorrect outputs are easily reversible and do not materially affect rights, safety, or essential services. The core requirement remains evidence preservation: even low-consequence alerts should identify the reference,

metric, and action.

15.2 Enterprise decision-support profile

For systems that influence human decisions, monitor both model behavior and downstream operational outcomes. Human reviewers should be able to challenge outputs and should receive drift context where it could affect reliance.

15.3 High-consequence or regulated profile

Use more conservative escalation, documented authority, subgroup analysis, change control, and independent verification where warranted. “High-consequence” is an operational risk description in this guide, not a legal classification. Legal applicability must be assessed separately by qualified counsel or compliance personnel.

15.4 Delayed-ground-truth profile

Design explicit label-acquisition or review sampling. Until outcomes arrive, avoid language that overstates evidence. Consider temporary restrictions when the potential consequence of undetected degradation is high.

15.5 Continuous or adaptive profile

Adaptive systems create an additional problem: the model can change while the environment changes. Preserve stable version identifiers, rollback points, and change records so that monitoring evidence remains attributable. If the

reference updates online, preserve the previous validated reference so gradual degradation is not silently normalized.

15.6 Generative, retrieval, and agentic profile

For generative and retrieval-based systems, the monitored “distribution” may include prompts, documents, retrieval candidates, evidence coverage, tool calls, policies, and output characteristics rather than a single fixed feature vector. For agentic systems, external tools and environment state can also change independently of the model. Preserve prompt/policy versions, retrieval corpus and index versions, tool/API versions, model-provider identifiers, and the execution path needed to reconstruct a material event. A change in output style or token distribution is not, by itself, evidence of degraded task quality; verification should use task-relevant outcomes, groundedness or evidence checks, scenario tests, and human review where appropriate.

16 Worked Example

Delayed Ground Truth

This example is illustrative and does not describe a specific TeraSystemsAI customer deployment.

16.1 Scenario

An enterprise uses a machine-learning model to prioritize potentially fraudulent transactions for human investigation. The model is decision support: investigators retain final authority. Confirmed fraud labels typically arrive days or weeks later.

16.2 Baseline

The organization records a validated operating condition including transaction mix, merchant categories, geography, score distribution, calibration, precision at the investigation budget, and subgroup false-positive rates. Monitoring runs daily, with a weekly outcome reconciliation.

16.3 DETECT

A classifier-based two-sample monitor and several feature-level checks indicate that the recent transaction distribution differs materially from the reference. The largest contribution comes from a new regional merchant network. Predicted-risk scores also shift upward. The alert is recorded as E1 evidence.

16.4 ASSESS

The team confirms that the new merchant network was intentionally onboarded. The shift is therefore real and partly expected. However, investigators report a higher volume of low-value alerts from that region. Labels are incomplete, so the team samples additional cases for expert review. Because the deployed model influences which cases are investigated, the team also retains a small probability-based audit sample from the affected region rather than relying only on model-selected cases. The combination of distribution change, score change, and review feedback raises the case to E2. The team does not claim confirmed model degradation yet.

16.5 RESPOND

Rather than retrain immediately, the risk owner authorizes three temporary controls: expand human review for the affected segment, prevent automatic downstream action for high-uncertainty cases, and increase label acquisition. The model remains in service for unaffected segments.

16.6 VERIFY

When outcome labels arrive, including the audit sample, the team finds that the model's false-positive burden increased materially in the new regional segment while performance in established segments remained stable. This provides E3 outcome-linked evidence for that segment. The team evaluates recalibration and segment-aware retraining candidates, tests them against the original and new segments, and selects the option that restores the defined acceptance criteria without unacceptable regression.

16.7 DOCUMENT

The final record preserves the alert, model and monitor versions, new merchant context, sampled labels, authority decision, temporary restrictions, candidate interventions, verification results, and residual risk. The new validated deployment condition becomes a controlled reference for future monitoring.

Lesson

The original drift alert was correct that conditions changed, but insufficient to determine the correct response. The evidence chain prevented both automatic retraining and complacent dismissal.

17 Relationship to External Frameworks

This section provides a high-level, non-authoritative mapping. It is intended to help teams place Distribution Shift Assurance within broader governance systems. It is not an official crosswalk and does not establish compliance.

17.1 NIST AI Risk Management Framework

NIST AI RMF 1.0 organizes AI risk-management outcomes around GOVERN, MAP, MEASURE, and MANAGE.[1] The TeraSystemsAI lifecycle can be conceptually related as follows: NIST's AI Resource Center notes that AI RMF 1.0

is being revised as of August 2026; organizations should verify current NIST guidance when using this mapping.[2]

Table 7: Conceptual relationship to NIST AI RMF 1.0

TeraSystemsAI stage	Related AI RMF function(s)	Relationship
DETECT	MEASURE	Monitor conditions, outputs, and evidence
ASSESS	MAP + MEASURE	Interpret context and measure material impact
RESPOND	MANAGE	Prioritize and implement risk responses
VERIFY	MEASURE + MANAGE	Evaluate response effectiveness and residual risk
DOCUMENT	GOVERN	Preserve accountability, policy, roles, and evidence
Cross-cutting governance	GOVERN	Define authority, policy, and organizational accountability

17.2 NIST post-deployment monitoring guidance

NIST AI 800-4 identifies post-deployment monitoring as important for validating real-world reliability, observing unforeseen outputs under dynamic inputs, and understanding unexpected consequences. It also reports that best practices and common terminology remain nascent.[3] Distribution Shift Assurance is compatible with that problem framing but is narrower and prescriptive about evidence flow and decision governance.

17.3 ISO/IEC 42001 and ISO/IEC 23894

ISO/IEC 42001:2023 specifies requirements and guidance for establishing, implementing, maintaining, and continually improving an AI management sys-

tem.[4] ISO/IEC 23894:2023 provides guidance on AI-specific risk management and integration of risk management into AI activities.[5] Distribution Shift Assurance can operate as a technical control within a broader management system, but this report does not reproduce ISO requirements and does not claim conformity.

17.4 EU AI Act

The EU AI Act establishes a risk-based legal framework for AI systems in the European Union.[6] For high-risk AI systems, the consolidated text addresses lifecycle risk management, logging, human oversight, accuracy and robustness, and post-market monitoring. Article 72 requires providers to establish and document a proportionate post-market monitoring system and to actively and systematically collect, document, and analyse relevant performance data across the system lifetime. The consolidated text current to 27 July 2026 reflects Regulation (EU) 2026/1744. Article 72(3) now provides that Commission guidance, including a template for the post-market monitoring plan, is to be adopted by 2 September 2027.[6] This report does not treat its lifecycle as an Article 72 template or legal crosswalk.

TeraSystemsAI's public EU AI Act field guide similarly frames compliance readiness as an engineering discipline built around evidence, oversight, transparency, and monitoring.[15]

Legal scope note

Regulatory obligations and application dates can change and depend on system classification, provider/deployer role, jurisdiction, sector, and the current consolidated law. This report is general technical guidance, not legal advice. Confirm applicable obligations against current official sources and qualified counsel immediately before publication or deployment.

18 Limitations and Open Questions

Distribution-shift assurance is not a solved discipline. Several limitations should remain visible.

18.1 No universal drift metric

Different data types, dimensions, and operational settings require different detectors. Statistical significance can be dominated by sample size, while simple distances can ignore decision relevance.

18.2 Thresholds are context-dependent

A threshold that is appropriate for a low-consequence recommender may be unacceptable for a safety-critical decision aid. Threshold design should reflect risk tolerance, false-alert cost, evidence latency, and escalation capacity.

18.3 Monitoring does not guarantee safety

A monitored system can still fail between checks, outside monitored slices, or through mechanisms the monitoring design did not anticipate. Monitoring is one control in a larger assurance architecture.

18.4 Outcome-label absence constrains certainty

Where outcomes are delayed or unavailable, organizations may need to act under uncertainty. Proxy signals can justify caution but should not be overstated as direct evidence of task performance. Where labels are selectively observed, the limitation is not only missingness but potential bias in what becomes observable.

18.5 Statistical drift does not establish causality

A correlation between distribution change and performance change may be operationally sufficient for intervention but does not necessarily identify the causal mechanism. Root-cause analysis remains important for durable remediation.

18.6 Adaptive and composed systems complicate attribution

If the model, prompt, retrieval corpus, policy, external tools, and environment all change, attributing degradation becomes difficult. Strong version control and change governance are necessary to keep the assurance record interpretable. A system-level failure can arise even when no single component exhibits a large marginal shift.

18.7 Human review has capacity and bias limits

Escalation can fail if review volume exceeds capacity. Reviewers can also be affected by automation bias, inconsistent adjudication, or selective exposure to difficult cases. Monitoring design should therefore consider the operational ability to investigate alerts, the competence and authority of reviewers, and the quality of the human-reference process. A theoretically sensitive detector that creates unmanageable review load can reduce assurance in practice.

18.8 Monitoring has blind spots

A detector can have low power for the shift that matters, especially in high-dimensional spaces, rare subgroups, or semantic changes not represented by monitored features. Failure to alert is therefore not evidence that deployment conditions are unchanged. Monitoring coverage, detector sensitivity, and known blind spots should be documented as part of the assurance case.

18.9 Open research questions

Important areas include calibrated monitoring under non-stationarity, evidence sufficiency without labels, subgroup-sensitive drift detection, sequential testing under dependence, causal diagnosis, selective-label correction, uncertainty-aware response policies, monitoring of generative and agentic systems, and governance of continuously learning systems. NIST AI 800-4 similarly identifies significant open questions and notes that validated monitoring methods and common terminology remain nascent.[3]

A Monitoring Readiness Checklist

Readiness question	Status
Is the intended use, decision boundary, and consequence of error documented?	☐
Is the reference condition defined, versioned, reproducible, and protected from silent replacement?	☐
If the reference is rolling or adaptive, is reference-update authority and validation explicitly controlled?	☐
Are monitored inputs, outputs, uncertainty measures, retrieval/tool signals, and outcomes selected based on risk?	☐
Are expected contextual changes such as seasonality, geography, policy, or population mix identified?	☐
Are subgroup and rare-event slices defined where aggregate metrics could hide material degradation?	☐
Are alert thresholds, effect-size expectations, persistence rules, and reset/escalation logic defined before incidents?	☐
Does the monitoring design account for repeated or sequential testing, overlapping windows, and temporal dependence?	☐

Readiness question	Status
Are system, model, monitor, preprocessing, retrieval, tool, and external-dependency versions linked to each material alert?	☐
Can the organization obtain credible outcome evidence or competent human adjudication when needed?	☐
If labels are selectively observed, is selection bias or censoring documented and addressed in the assessment?	☐
Is there a named assessment owner and a named authority for material intervention and residual-risk acceptance?	☐
Do human reviewers have sufficient competence, authority, information, time, and capacity to challenge or override automation?	☐
Can the system restrict, abstain, roll back, fall back, or suspend operation when evidence and consequence justify it?	☐
Are verification acceptance criteria specified before the final intervention result is known?	☐
Where feasible, is verification evidence independent of the data used to select or tune the intervention?	☐
Are material cases retained in an auditable record that separates observed, inferred, decided, and unresolved elements?	☐
Are monitoring-policy, reference, threshold, model, retrieval, and dependency changes version controlled?	☐
Is monitoring and human-review capacity realistic relative to expected alert volume and response-time requirements?	☐
Are known monitoring blind spots and unobservable outcomes documented?	☐

B Distribution Shift Evidence Record Template

Field	Record
Alert / case ID	
System / application and intended use	
Model / provider model name and version	
Monitor name and version	
Preprocessing / feature-pipeline version	
Retrieval corpus / index / prompt / tool versions, if applicable	
External dependency or API versions, if material	
Reference condition definition and version	
Observation window	
Metric / statistical test / monitor signal	
Threshold, persistence rule, and observed value	
Affected feature / subgroup / context / execution path	
Data lineage / source version	
Expected contextual change?	
Outcome evidence available?	

Field	Record
Outcome sampling / adjudication method	
Known selection, censoring, delay, or missing-label limitation	
Assessment conclusion	
Alternative explanations considered	
Evidence sufficiency level (E0–E4), by claim where needed	
Assessment confidence and unresolved uncertainty	
Decision authority and reviewers	
Response selected	
Alternatives considered and rationale	
Verification criteria defined before final result?	
Verification data / evidence source and independence note	
Verification result	
Residual risk / operating restriction	
Follow-up owner and due date	
Links to tickets / risk register / model card / incident record	

Record language

Separate **observed evidence**, **inferred interpretation**, **authorized decision**, and **unresolved uncertainty**. Do not rewrite an unlabeled drift alert as confirmed performance degradation. Record how outcomes were sampled, because selectively observed labels can bias verification.

C Illustrative Decision Matrix

Evidence	Consequence	Outcome evidence	Illustrative posture
E1	Low	Unavailable	Continue monitoring; confirm signal; no automatic model change
E1–E2	High	Unavailable	Increase competent human review; consider restriction; acquire credible outcome evidence urgently
E2	Moderate	Partial	Diagnose source; compare affected slices; examine sampling bias; define temporary controls
E3	Moderate	Available	Select proportionate repair, recalibration, restriction, or retraining based on evidence and likely mechanism
E3–E4	High	Available	Formal intervention; independent or second-line verification where warranted; explicit residual-risk approval
Any	Critical policy / safety violation	Any	Follow pre-defined incident or suspension policy; do not wait for statistical certainty if policy requires immediate control

D Model and Monitoring Change Log

Date	Version	Component	Change	Approval / evidence link
		Model / provider model		
		Preprocessing / feature pipeline		
		Retrieval corpus / index		
		Prompt / policy / system instruction		
		Tool / external dependency		
		Drift monitor / representation		
		Threshold / alert logic		
		Reference condition		
		Outcome sampling / adjudication		
		Escalation / human-review policy		
		Verification criteria		

E Terminology and Version History

Terminology

Distribution shift

A difference between a deployment joint distribution and a defined reference distribution, $P_t(X, Y) \neq P_{\text{ref}}(X, Y)$. The term alone does not imply harm, detectability, or model failure.

Input shift

A change in the marginal input distribution, $P_t(X) \neq P_{\text{ref}}(X)$. "Covariate shift" is used more narrowly in this report when $P(Y \mid X)$ is assumed stable.

Label shift

A change in $P(Y)$ under the additional assumption that $P(X \mid Y)$ remains stable. A prevalence change without that stability assumption should not automatically be labeled "label shift."

Conditional or concept shift

A change in $P(Y \mid X)$. Such a change can degrade task performance even when marginal input monitoring shows little or no shift.

Drift signal

A monitored indicator that deployment data, outputs, uncertainty, context, or system behavior differ from a defined reference. It is not, by itself, proof of performance degradation or causal mechanism.

Out-of-distribution observation

An individual observation or batch judged atypical under a chosen representation, score, or reference. An OOD observation is not synonymous with a population-level distribution shift.

Consequential shift

A change for which credible evidence indicates material impact on a requirement of the intended use, such as performance, calibration, safety, fairness, security, reliability, or policy compliance.

Reference condition

A documented, versioned baseline that includes relevant data, system, time, population, context, and dependency assumptions against which deployment conditions are compared.

Outcome reference / ground truth

Observed or adjudicated evidence used to evaluate task performance. It should not be assumed to be perfectly measured, unbiased, complete, immediate, or independent of prior model decisions.

Evidence sufficiency level

An E0–E4 communication label for the support available for a particular claim or decision. It is not a probability, certification, maturity level, or substitute for domain-specific acceptance criteria.

Assurance record

The preserved evidence chain linking detection, assessment, authority, response, verification, alternatives, residual risk, and unresolved uncertainty.

Residual risk

Risk remaining after a decision or intervention, including uncertainty caused by incomplete evidence, unobserved subgroups, delayed outcomes, and known monitoring blind spots.

Version history

Version	Date	Status	Change
1.0-RC1	Aug. 2026	Prepublication Review	Senior technical refinement of shift taxonomy, sequential monitoring, selective-label risk, evidence-sufficiency semantics, generative and agentic monitoring, verification independence, and current external-framework references.

Future revision triggers

A future version should be considered when material new evidence changes monitoring practice; NIST AI RMF revisions materially alter the external mapping; relevant ISO standards change; legal frameworks change; or TeraSystemsAI publishes validated refinements to the assurance method.

References

Primary sources and companion TeraSystemsAI materials

References to standards and regulation are contextual and should be reverified at release. Research citations support specific technical distinctions; they do not make the TeraSystemsAI lifecycle an external standard.

[1] Tabassi, E. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. National Institute of Standards and Technology. doi:10.6028/NIST.AI.100-1.

[2] National Institute of Standards and Technology. (2026). NIST AI Resource Center - AI Risk Management Framework. Accessed August 9, 2026. https:

//airc.nist.gov/airmf-resources/airmf/. NIST notes that AI RMF 1.0 is being revised.

[3] Rao, A., Keller, A., Kalra, N., Steed, R., Kwegyir-Aggrey, K., Klyman, K., Staheli, D., & Bergman, A. (2026). Challenges to the Monitoring of Deployed AI Systems. NIST Trustworthy and Responsible AI 800-4. doi:10.6028/NIST.AI.800-4.

[4] ISO/IEC. (2023). ISO/IEC 42001:2023 - Information technology - Artificial intelligence - Management system. International Organization for Standardization. https://www.iso.org/standard/42001.

[5] ISO/IEC. (2023). ISO/IEC 23894:2023 - Information technology - Artificial intelligence - Guidance on risk management. International Organization for Standardization. https://www.iso.org/standard/77304.html.

[6] European Parliament and Council of the European Union. (2024; consolidated text current to 27 July 2026). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence, as amended including Regulation (EU) 2026/1744. EUR-Lex. https://eur-lex.europa.eu/eli/reg/2024/1689/2026-07-27/eng.

[7] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., & Snoek, J. (2019). Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. Advances in Neural Information Processing Systems, 32. NeurIPS proceedings.

[8] Ginart, A. A., Zhang, M. J., & Zou, J. (2022). MLDemon: Deployment Monitoring for Machine Learning Systems. Proceedings of the 25th International Conference on Artificial Intelligence and Statistics, PMLR 151, 3962–3997. PMLR.

[9] Cobb, O., & Van Looveren, A. (2022). Context-Aware Drift Detection. Proceedings of the 39th International Conference on Machine Learning, PMLR 162, 4087–4111. PMLR.

[10] TeraSystemsAI LLC. (2026). Research - Production-Ready Uncertainty Systems. https://www.terasystems.ai/research.html. Accessed August 9, 2026.

[11] TeraSystemsAI LLC. (2026). Publications - Scientific work for trustworthy decisions. https://www.terasystems.ai/publications.html. Accessed August 9, 2026.

[12] TeraSystemsAI LLC. (2026). Accountability Framework - Human Accountability Must Survive Automation. https://www.terasystems.ai/accountability.html. Accessed August 9, 2026.

[13] TeraSystemsAI LLC. (2026). Constraint-Aware AI Engineering Manifesto. https://www.terasystems.ai/philosophy.html. Accessed August 9, 2026.

[14] TeraSystems Research. (2026). Reliable Structure Extraction from Unstructured Documents. TeraSystemsAI Insights Hub. https://www.terasystems.ai/New_Blog/28-extracting-structure.html.

[15] TeraSystems Research. (2026). The EU AI Act in Practice: A Field Guide. TeraSystemsAI Insights Hub. https://www.terasystems.ai/New_Blog/29-eu-ai-act-field-guide.html.

[16] TeraSystems Research. (2026). Distribution Shift: Why Models Fail Quietly. TeraSystemsAI Insights Hub. https://www.terasystems.ai/New_Blog/30-distribution-shift.html.

[17] Ngartera, L., Nadarajah, S., & Koina, R. (2026). Bayesian RAG: Uncertainty-Aware Retrieval for Reliable Financial Question Answering. Frontiers in Artificial Intelligence, 8:1668172. doi:10.3389/frai.2025.1668172.

[18] Ngartera, L., Nadarajah, S., Koina, R., & Gningue, Y. (2026). BRAG: Bayesian Retrieval-Augmented Generation; A Methodological Framework for Evidence-Governed Decision Support. Machine Learning and Knowledge Extraction, 8(6), 151. doi:10.3390/make8060151.

[19] Boeken, P., Zoeter, O., & Mooij, J. (2024). Evaluating and Correcting Performative Effects of Decision Support Systems via Causal Domain Shift. Proceedings of the Third Conference on Causal Learning and Reasoning, PMLR 236, 551–569.

[20] Lipton, Z. C., Wang, Y.-X., & Smola, A. J. (2018). Detecting and Correcting for Label Shift with Black Box Predictors. Proceedings of the 35th International

Conference on Machine Learning, PMLR 80, 3122–3130.

[21] Rabanser, S., Günnemann, S., & Lipton, Z. C. (2019). Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift. Advances in Neural Information Processing Systems, 32.

[22] Jang, S., Park, S., Lee, I., & Bastani, O. (2022). Sequential Covariate Shift Detection Using Classifier Two-Sample Tests. Proceedings of the 39th International Conference on Machine Learning, PMLR 162, 9845–9880.

TeraSystemsAI LLC

Research and engineering for trustworthy, evidence-grounded, and constraint-aware intelligent systems.

TSAI-TR-2026-001 | Version 1.0-RC1 | Prepublication Review | August 2026