Trustible

AI Monitoring for Governance Teams

A Practical Guide to Monitoring AI Systems After Deployment

trustible.ai

Trustible

Table of Contents

Trustible

Introduction

AI monitoring means different things to different people. To a machine learning engineer, it means tracking model drift and inference latency. To a CISO, it means watching for adversarial attacks and unauthorized access. To a lawyer, it means staying ahead of regulatory changes that could shift the legal exposure of a deployed system. To a business leader, it means knowing whether an AI investment is still delivering value.

All of these are valid. And all of them are necessary. The problem is that most organizations treat AI monitoring as a single technical discipline when it's actually a cross-functional responsibility that spans engineering, security, legal, and business teams. The result is predictable: teams invest in the monitoring they know best and leave gaps in the areas they don't. Engineers build great model performance dashboards. Governance teams scan the news for regulatory changes. Nobody connects the two.

Post-market monitoring is an explicit requirement for high-risk AI systems under the EU AI Act (Article 72). ISO 42001 requires ongoing risk monitoring as part of AI management system maintenance. The NIST AI Risk Management Framework calls for continuous monitoring across its GOVERN and MANAGE functions. OMB M-25-21 mandates continuous monitoring for federal AI systems. The regulatory consensus is clear: deploying an AI system isn't a one-time event. It requires ongoing oversight.

In March 2026, NIST published AI 800-4, "Challenges to the Monitoring of Deployed AI Systems." It's the first major government report to systematically catalog what post-deployment AI monitoring involves and where the gaps are. Drawing on three workshops with over 250 practitioners and a review of 87 published papers, the report identifies six categories of monitoring and surfaces dozens of open challenges. This whitepaper builds on that framework.

Our core argument: AI monitoring is ultimately about ensuring that your organization captures the signals that should prompt governance action, and has the structures in place to respond. Monitoring generates signals. Signals become governance triggers. Triggers initiate governance activities. That pipeline is the operational backbone of ongoing AI oversight. But it only works if you're monitoring broadly enough to catch the signals that matter. Most organizations aren't.

The NIST AI 800-4 Monitoring Framework

NIST AI 800-4 proposes six categories of post-deployment AI system monitoring. These categories were derived from practitioner workshops and a literature review, and they represent the most structured vocabulary currently available for discussing what AI monitoring should cover. For each category, NIST provides a definition, documents current practices, and catalogs gaps, barriers, and open questions.

Functionality Monitoring
Does the system continue to work as intended? This covers performance baselines, drift detection, model comparison, and degradation over time. Key challenges include the lack of ground-truth datasets in production settings, the overhead of longitudinal tracking, and the difficulty of establishing meaningful performance thresholds. Drift detection in particular is only useful when calibrated to the specific use case, something that requires governance context established before deployment.

Operational Monitoring
Does the system maintain consistent service? This covers uptime, latency, compute costs, and distributed logging. Key challenges include fragmented logging across distributed infrastructure (especially relevant for agent-based systems) and tracking indirect costs beyond raw compute, like human labor and opportunity costs.

Human Factors Monitoring
Is the system transparent to humans and high quality? This covers user feedback, human-AI interaction patterns, and dark design patterns like sycophancy and anthropomorphization. NIST's data shows something notable here: workshop practitioners raised human factors issues far more frequently than the published literature addresses. That gap suggests this is the most underserved monitoring category in the field today.

Security Monitoring
Is the system secure against attacks and misuse? This covers adversarial attacks, jailbreaks, data poisoning, and deceptive model behavior. A notable challenge: AI systems may behave differently when they detect they're being evaluated, making test-time monitoring unreliable as a sole source of truth. NIST also flags the difficulty of detecting whether a model is actively attempting to subvert its monitoring setup.

Compliance Monitoring
Does the system adhere to relevant regulations and directives? This covers regulatory tracking, terms-of-service enforcement, and navigating a shifting policy environment across jurisdictions. NIST notes that compliance monitoring is focused primarily on knowing what to monitor for, rather than how to do it. That's significant: the regulatory environment is changing faster than most organizations can track, and the cost of missing a new requirement can be severe.

Large-Scale Impacts Monitoring
Does the system promote human flourishing? This covers societal impact, downstream effects on stakeholders, and measuring beneficial outcomes. Key challenges include the absence of standard metrics for positive impact, the difficulty of capturing distributed harm through incident logging, and the near-impossibility of tracking downstream effects once open-weight models are released.

Cross-Cutting Challenges

Across all six categories, NIST identifies five themes that recur as barriers to effective monitoring: a lack of trusted methods and standards; visibility and transparency gaps across the AI value chain; the pace of change in technology and regulation; misaligned organizational incentives; and the resource burden of monitoring itself. These aren't theoretical concerns. They reflect the daily reality of teams trying to maintain oversight of AI systems that were deployed faster than governance structures could keep up.

NIST AI 800-4

"Stakeholders across the AI ecosystem agree on the need for post-deployment monitoring; however, best practices, validated methodologies, and common terminology is nascent."

The report calls for further guidance on monitoring methods ranging from field studies to incident monitoring.

Trustible

AI Deployment Patterns and Monitoring Implications

Before diving into how to organize monitoring, it's worth understanding the different ways AI systems get deployed. Each deployment pattern determines what monitoring data is accessible and what tools are needed to collect it. Most organizations will have AI systems across multiple patterns, which is part of what makes monitoring hard.

Local AI Model
The AI model runs on a portable device like a laptop or phone. On-device AI is increasingly common (Apple Intelligence, local code assistants), but monitoring is challenging because there's no centralized place to observe the system's behavior. Any monitoring requires the ability to send telemetry from individual devices to aggregated data stores, which raises both technical and privacy concerns.

Self-Hosted Model
The organization hosts the AI model directly in its own cloud environment with full access to all runtime components. Open-weight models that organizations self-host fall into this category. The key advantage for monitoring: all inputs, inference logs, output streams, and system metrics are fully accessible. This is the easiest deployment pattern to instrument for internal monitoring.

API-Only Model
The organization accesses an AI system purely through an API or abstract interface. They may have a dedicated cloud endpoint, but that endpoint acts as its own abstraction layer, and direct access to logs may only be available in limited form. Proprietary models from OpenAI or Anthropic accessed through their platforms fit this pattern. Monitoring here depends heavily on what the provider chooses to expose.

Vendor AI SaaS
The AI system is fully hidden behind a vendor's application layer. The critical distinction: there's often a lot of opaque system logic sitting between the user and the AI model itself. Accessing monitoring data is the most restricted here, and typically depends on whatever enterprise features (audit logs, compliance endpoints, admin consoles) the vendor offers. Many organizations have dozens of SaaS tools with AI features enabled, each with its own level of observability.

Trustible

A key implication of these patterns: any approach to internal monitoring needs to be adapted per deployment type. Self-hosted models give you full access; SaaS tools might give you almost nothing. But the concepts we'll introduce next, particularly external and use-case-level monitoring, apply consistently across all four patterns. That makes them especially valuable for organizations with diverse AI portfolios.

Trustible

Two Levels of Monitoring: Models and Use Cases

The NIST framework tells us what to monitor for. The next question is: monitor what, exactly? The AI monitoring conversation typically defaults to model-level concerns. Is this endpoint performing within spec? Has output distribution shifted? Those are important questions. But they aren't the questions governance teams need answered.

Organizations don't govern models. They govern use cases. A single use case ("AI-assisted customer support") might involve multiple model endpoints, a retrieval-augmented generation pipeline, a guardrail layer, a human escalation path, and a vendor SaaS interface. Model monitoring tells you about each component in isolation. Use case monitoring tells you about the whole system and whether it's meeting its goals.

Model Monitoring

Model monitoring tracks the performance and behavior of individual model endpoints: accuracy, latency, token consumption, input/output distribution shifts, and guardrail trigger rates. It's the domain of MLOps platforms like Datadog, Arize, Evidently, and WhyLabs.

The tooling here is increasingly mature and automatable. But there are structural limitations. Model monitoring tells you about the model, not the outcome. A model can be performing perfectly on its own metrics while the use case it supports is failing its users. And if you've got a dozen model endpoints serving one use case, those signals need to be aggregated into a use-case-level view before governance teams can make decisions.

The two levels connect through the monitoring pipeline. Drift detected on a model endpoint is a data point. Whether that drift is significant enough to change the risk profile of the use case, and what to do about it, is a governance question. Model-level signals become inputs to use-case-level governance decisions.

Use Case Monitoring

Use case monitoring tracks whether the AI system, taken as a whole, is meeting its intended goals, operating within its risk tolerance, and staying compliant with applicable requirements. Is the system causing harm? Has the regulatory environment changed? Is the vendor ecosystem stable? Has the risk profile shifted enough to warrant reassessment?

Use case monitoring spans all six NIST categories. It doesn't depend on deep technical access to model internals. But it's harder to automate and requires structured governance processes: a clear AI inventory, documented risk profiles, and defined thresholds for when signals should trigger action.

Why the Distinction Matters

Monitoring requires context to be useful. This is a point that's easy to underestimate. Model drift isn't a universal absolute. A 3% shift in output distribution might be catastrophic for a medical triage system and completely irrelevant for an internal document summarizer. Defining what constitutes meaningful drift, what thresholds matter, and what counts as acceptable performance requires governance context that lives at the use case level. Risk assessments, documented goals, defined tolerances: this governance work upfront is a prerequisite for effective monitoring, not a separate activity.

The two levels connect through the monitoring pipeline. Drift detected on a model endpoint is a data point. Whether that drift is significant enough to change the risk profile of the use case, and what to do about it, is a governance question. Model-level signals become inputs to use-case-level governance decisions.

Trustible

Internal and External Monitoring

Regardless of whether you're monitoring at the model or use case level, monitoring activities fall into two orientations based on where the signal comes from.

Internal Monitoring

Internal monitoring looks inward at the system itself. Data sources include system logs, inference logs, input/output data streams, software performance metrics, automated product analytics, and network traffic. The primary audience is ML engineering, DevOps, platform teams, and security operations.

Within the NIST framework, internal monitoring provides the strongest coverage for Functionality and Operational monitoring. It partially covers Security (access logs, guardrail triggers). It has limited reach into Human Factors, Compliance, and Large-Scale Impacts. For most organizations, internal monitoring is the more familiar discipline. The tooling is more mature, the workflows are better established, and the teams responsible for it have been doing analogous work in DevOps and cybersecurity for years.

External Monitoring

External monitoring looks outward from the system. Data sources include AI incident reports, structured user feedback, academic research on AI risks, regulatory and legal updates, vendor product announcements and changelogs, and industry benchmarks like transparency indexes.

The primary audience is governance teams, risk managers, compliance officers, legal teams, and business owners. Within the NIST framework, external monitoring is strongest for Compliance, Large-Scale Impacts, and Human Factors. It also provides important supplementary signals for Functionality (user-reported quality issues) and Security (published vulnerabilities, peer incidents).

External monitoring is where most organizations have the biggest gap. It's less familiar, less tooled, and often nobody's explicit responsibility. But it's also where many of the highest-impact governance triggers originate: a new regulation, a vendor deprecation, a published incident affecting a model you depend on.

How They Work Together

Internal and external monitoring aren't competing approaches. They're complementary. Internal monitoring catches technical signals: a model endpoint drifting, latency spiking, a guardrail triggering more often. External monitoring catches environmental signals: a regulatory change, a vendor update, a user reporting harm. Both feed into the same governance pipeline.

The difference in deployment pattern access matters here. Internal monitoring is constrained by deployment type: you get full visibility into a self-hosted model and almost nothing from a vendor SaaS tool. External monitoring applies equally across all deployment patterns. Whether your AI runs on a laptop or behind a vendor's API, you still need to track regulatory changes, vendor updates, published incidents, and user feedback. That consistency makes external monitoring especially valuable for organizations managing a diverse AI portfolio.

The Agent Dimension

AI agents complicate both orientations in ways the existing tooling hasn't caught up with.
Internally, agent monitoring is harder because agents chain multiple model calls, invoke tools, and operate across distributed infrastructure. There's no universal standard for logging agent actions or identifying agent-initiated requests. Protocols like MCP and Google's A2A are still evolving, and monitoring frameworks haven't adapted yet.

Agents also blur the internal/external boundary. When an agent books a meeting, submits a form, writes code, or delegates to another agent, it's taking actions with real-world consequences beyond model inference. Monitoring the model call is internal. Monitoring whether the agent's action was appropriate, authorized, and correct is external. And in multi-agent systems where Agent A delegates to Agent B, which then calls Agent C's API, there may not be a clear "right answer" for what the optimal chain of actions should have been. Defining expected behavior for these systems requires the same kind of use-case-level governance context that makes model drift meaningful.

Shadow agents are an emerging blind spot. Browser-based agent tools, personal API keys connected to automation platforms, and autonomous workflows running outside IT's visibility create monitoring gaps that purely internal approaches can't address. Shadow agents are becoming the next generation of shadow AI.

Monitoring User Stories

With the conceptual framework in place, we can now outline the typical monitoring scenarios that organizations want to capture and respond to. These user stories represent the practical "what are we actually watching for?" of AI monitoring. Each maps to one or more NIST categories and falls on the internal or external side of the monitoring spectrum.

For each user story, we describe what it involves, give a concrete example, and note key implementation considerations. These aren't exhaustive, but they cover the scenarios that come up most frequently in governance conversations.

External Monitoring User Stories

External monitoring user stories focus on signals that originate outside the AI system's technical stack. They tend to be less tool-dependent and more process-dependent, which makes them accessible to governance teams without deep MLOps expertise.

Incident alert
NIST categories: Functionality, Security, Large-Scale Impacts

This is about triggering an internal review based on a newly reported AI failure, vulnerability, or published incident affecting a system, model, or vendor the organization uses. For example, a major AI vendor releases a public statement about a critical vulnerability in their latest model, prompting an urgent internal review of all systems using that model. Implementation depends on monitoring AI incident databases (like the OECD AIM or PAI AIID), vendor security bulletins, and news sources. The data then needs to be compared against your internal AI inventory to determine relevance.

Vendor change monitoring
NIST categories: Operational, Compliance, Functionality

Tracking vendor product announcements, deprecations, security patches, model updates, and terms-of-service changes that may affect the organization's risk profile. For example, a cloud provider announces an unexpected end-of-life date for an API powering a core AI feature, requiring a migration plan. The main data sources are vendor changelogs, release notes, and TOS updates. This data needs to be mapped against an internal inventory of which systems depend on which vendors.

User feedback analysis
NIST categories: Human Factors, Functionality

Collecting and categorizing feedback submitted by end-users through support channels, surveys, or public forums to identify quality or safety concerns. For example, a customer submits a help ticket reporting that an AI chatbot gave them factually incorrect and potentially harmful information. This requires collecting raw feedback, categorizing it (positive interactions, improvement requests, system failures), and tracking trends over time. A surge in specific types of negative feedback can serve as an early warning signal.

ROI assessment
NIST categories: Functionality, Large-Scale Impacts

Evaluating whether an AI system continues to meet its intended business goals and whether costs are justified by realized benefits. For example, an organization conducts an internal user survey about an AI tool up for renewal to decide if positive ROI justifies the cost. This often can't be fully automated. It may require combining user feedback with usage analytics, cost reports, and business KPIs, then making a judgment call about whether the system is still worth it.

Regulatory update mapping
NIST categories: Compliance

Mapping new regulations, judicial rulings, enforcement actions, or standards updates to the existing AI inventory to identify compliance gaps. For example, a regulator releases new compliance guidance recommending specific controls for a type of AI system your organization deploys. This requires a strategy for tracking relevant policy updates and a process for mapping those updates against your AI inventory. Once affected systems are identified, some form of gap assessment or audit determines what additional work is required.

Stakeholder impact review
NIST categories: Large-Scale Impacts, Human Factors

Periodic assessment of how an AI system affects distinct stakeholder groups, including intended users, affected populations, and downstream dependents. This is less about detecting a specific alert and more about scheduled governance reviews. It may involve stakeholder interviews, demographic analysis, and structured impact assessment frameworks. Several regulations require this for high-risk systems.

Third-party model evaluation
NIST categories: Security, Functionality
Assessing upstream model provider transparency, risk disclosures, red-teaming results, and published evaluation data. For example, reviewing a vendor's newly published model card or safety report to determine whether previously unknown risks affect your use case. This is especially relevant when upstream providers release new model versions or when third-party audits surface new findings.

Agent activity review
NIST categories: Human Factors, Security, Functionality
Reviewing actions taken by AI agents to determine whether tool calls, delegations to other agents, and real-world actions were appropriate and authorized. For example, an AI agent autonomously sends an email to a client that contains inaccurate pricing, triggering a review of agent permissions and output validation. This is an emerging user story with immature tooling. Agent action logs, tool-call traces, and outcome validation are the primary data sources, but standards for these are still in development.

Internal Monitoring User Stories

Internal monitoring user stories focus on signals from within the AI system's technical stack. They tend to be more automatable but require deeper technical access, which varies by deployment pattern.

User behavior analysis
NIST categories: Human Factors, Security
Observing regular patterns of user behavior and flagging deviations that may indicate harmful or unauthorized use. For example, a user who normally interacts with an internal AI assistant about work topics begins discussing sensitive personal topics. This requires input datastreams and user identity data, plus a way of classifying usage into relevant categories. Defining what behavior to watch for and what thresholds to use is itself a complex task that often uses AI.

Bias monitoring
NIST categories: Functionality, Large-Scale Impacts

Detecting unfair or biased outcomes across demographic or protected groups. For example, a job application filtering system consistently scores candidates from one demographic lower than others with identical qualifications. This involves complex analysis of input and output streams, sometimes combined with demographic data about users. Defining what to look for often requires input from legal teams and subject matter experts in fairness and equity.

Performance degradation
NIST categories: Functionality

Monitoring key statistical performance metrics in production (accuracy, recall, latency, false positive rates) for unacceptable drops in quality. For example, a fraud detection system's false positive rate suddenly doubles, flagging legitimate transactions. Some degradation can be detected from software metrics like inference latency, but most will require comparing the output stream with user feedback to identify meaningful quality declines.

Agent tool-use monitoring
NIST categories: Security, Functionality, Operational

Logging and tracing agent tool calls, multi-agent delegations, and autonomous actions. This is a new and rapidly evolving user story. When an agent chains multiple tool calls or delegates to other agents, the monitoring challenge is both technical (how to capture the trace) and conceptual (how to determine if the chain of actions was correct). Protocols like MCP and A2A are still being standardized, and monitoring tooling for agentic systems is immature.

Challenges and Open Questions

AI monitoring is a hard problem. Several features of the current environment make it especially so.

Generative AI Resists Simple Metrics
Traditional classification models had structured inputs and outputs designed for a single purpose, which made defining monitoring metrics relatively tractable. Generative AI systems accept and produce unstructured content, can be used for a wide range of tasks, and have no single "correct" output. Defining what to monito r and what counts as a problem is much harder. Often it requires a subject matter expert, not just a statistical test.

The Agent Monitoring Gap
Agent frameworks and protocols are changing monthly. MCP, A2A, and competing approaches are still being standardized. There's no universal agent identity scheme, no standard for logging tool calls across systems, and no established method for tracing multi-agent delegation chains. When Agent A asks Agent B to perform a task, and Agent B calls three different tools, determining whether the outcome was correct requires reasoning about intent and context that current monitoring tools can't provide.

Context Is a Prerequisite
Effective monitoring requires governance context established before monitoring can be useful. Drift is meaningless without a defined baseline and thresholds calibrated to the specific use case. Alert criteria require documented risk tolerances. Incident triage requires understanding the intended users and potential harms. Organizations that skip the governance work and jump straight to monitoring infrastructure end up with dashboards nobody can interpret and alerts nobody knows how to act on.

Privacy vs. Observability
Many monitoring techniques require access to prompt streams and model outputs, which most organizations consider highly sensitive. NIST AI 800-4 identifies this as a cross-cutting challenge, noting the difficulty of balancing user privacy with system transparency. Privacy-preserving techniques exist (aggregated analytics, differential privacy), but they reduce the granularity of available signals.

Shadow AI and AI Sprawl
Organizations can only effectively monitor what they know about. Shadow AI (unauthorized tools, personal accounts, browser-based agents) creates blind spots. Even for known systems, each deployment pattern requires its own integration approach, which multiplies the engineering effort for internal monitoring across a diverse portfolio.

Alert Fatigue and Actionability
Cybersecurity teams have long struggled with alert fatigue from excessive false positives. AI monitoring faces the same risk, plus an additional complication: the "right response" to an AI alert is often unclear. If drift is detected, is that enough to pause the system? If a user reports a bad output, is that an incident or normal variance? AI-specific governance playbooks are still immature, and without them, monitoring produces noise rather than signal.

The Expertise Gap
Teams setting up and responding to monitoring need expertise spanning AI technology, risk management, and regulatory requirements. Many governance teams lack the technical depth to interpret model-level signals. Many engineering teams lack the governance context to know which signals matter. Bridging that gap is an organizational challenge, not a tooling challenge.

Conclusion and Recommendations

AI monitoring isn't one discipline owned by one team. It's a cross-functional practice that spans engineering, security, legal, and business stakeholders. The NIST AI 800-4 framework provides a structured vocabulary for what monitoring covers. But knowing the categories isn't enough. Organizations need to decide what level they're monitoring at (models vs. use cases), where the signals come from (internal vs. external), and what happens when those signals indicate a problem.

Most organizations have invested heavily in internal, model-level monitoring while leaving external, use-case-level monitoring underdeveloped. That's the gap this whitepaper is designed to highlight and help close. The user stories in Section 6 provide a practical starting point for thinking about what your organization should be watching for.

Our recommendations:

1. **Build and maintain an AI inventory.** You can't monitor what you don't know about. Discovery is the first step of any monitoring program, and it needs to account for shadow AI and agent-based tools.
2. **Do the governance work first.** Risk assessments, documented goals, defined thresholds. Without this context, monitoring data is uninterpretable. Drift means nothing without a baseline calibrated to the use case.
3. **Distinguish model monitoring from use case monitoring.** Both matter, but governance decisions happen at the use case level. Make sure your monitoring aggregates model-level signals into use-case-level views.
4. **Use the NIST monitoring categories as a planning framework.** For each AI system, identify which categories are relevant and which data sources, both internal and external, feed each one.
5. **Invest in external monitoring.** Regulatory tracking, vendor monitoring, incident scanning, and structured user feedback are high-impact activities that work across all deployment patterns without deep technical integration.
6. **Connect monitoring to governance action.** The goal isn't more dashboards. It's ensuring that when monitoring reveals something important, there's a clear owner, a defined response, and a process for following through.
7. **Plan for agents now.** Agent monitoring is an emerging requirement. Build visibility into agent usage, establish logging expectations, and develop governance processes before adoption outpaces your oversight.

The organizations that get AI monitoring right won't be the ones with the most advanced MLOps pipelines. They'll be the ones that connect technical signals to governance decisions, track the external environment as carefully as they track their own systems, and build the organizational discipline to act on what they find.

ABOUT TRUSTIBLE

Trustible is a purpose-built AI governance platform that helps organizations inventory, assess, and oversee their AI systems. The platform connects AI inventory management, automated risk assessment, compliance tracking, vendor evaluation, and governance workflows into a single system of record for AI governance. Learn more at trustible.ai.

Trustible

Appendix: NIST Categories and Monitoring Orientation

This table maps each NIST AI 800-4 monitoring category to internal and external monitoring activities, governance triggers, and regulatory requirements.

NIST Category	Internal Monitoring	External Monitoring	Governance Triggers	Regulatory Hooks
Functionality	Drift detection, performance metrics, automated evals	User-reported quality issues, SME output review, ROI assessment	Drift threshold crossed, degradation, goals unmet	NIST AI RMF MEASURE; EU AI Act Annex IV
Operational	Uptime, latency, compute costs, system logging	Vendor SLA compliance, indirect cost tracking, total cost of ownership	SLA breach, cost overrun, service disruption	Standard IT operations frameworks
Human Factors	Telemetry, product analytics, click tracking	User feedback, stakeholder assessment, dark pattern detection	User-reported harm, feedback surge, accessibility issues	EU AI Act transparency; NIST AI RMF MAP
Security	Access logs, guardrails, vulnerability scanning	Published CVEs, peer incidents, deceptive behavior reports	Adversarial attack, new vulnerability, incident reported	NIST AI 100-2; EU AI Act cyber requirements
Compliance	Automated policy enforcement, guardrail trigger rates	Regulatory tracking, standards updates, TOS monitoring	New regulation, investigation launched, takedown notice	EU AI Act Art. 72; ISO 42001; OMB M-25-21
Large-Scale Impacts	Bias metrics, fairness evaluations	Stakeholder feedback, societal impact assessment, industry patterns	Breakthrough tech, incident pattern, impact identified	EU AI Act FRIA; state algorithmic impact laws

Trustible

References

NIST AI 800-4: Rao AK, Keller AJ, Kalra N, Steed R, Kwegyir-Aggrey K, Klyman K, Staheli D, Bergman AS (2026). Challenges to the Monitoring of Deployed AI Systems. National Institute of Standards and Technology, NIST AI 800-4. https://doi.org/10.6028/NIST.AI.800-4

EU AI Act: Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence.

ISO/IEC 42001:2023: Information technology. Artificial intelligence. Management system.

NIST AI Risk Management Framework (AI RMF 1.0): National Institute of Standards and Technology, NIST AI 100-1, January 2023.

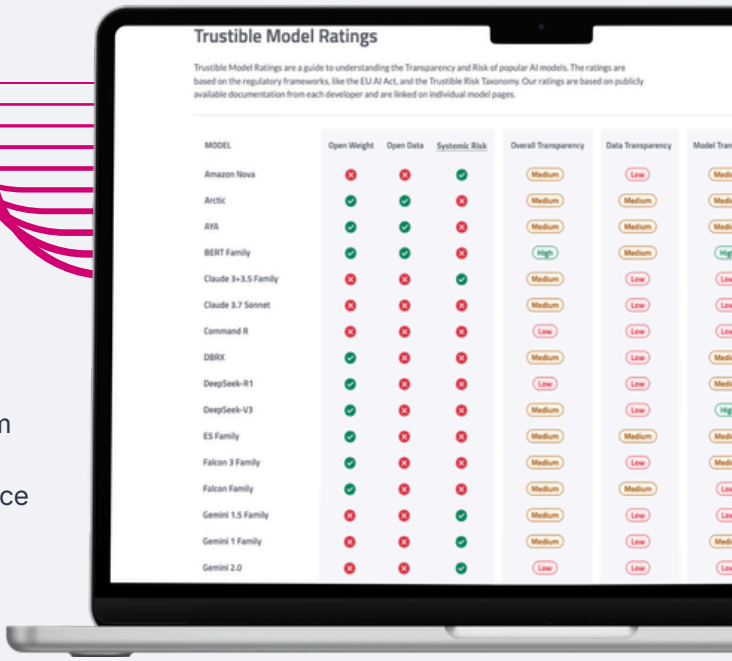
OMB M-25-21: Accelerating Federal Use of AI through Innovation, Governance, and Public Trust. Office of Management and Budget, 2025.

NIST AI 100-2: Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. March 2025.

Trustible. AI Governance Triggers: When to Act and Why It Matters. 2025. Available at trustible.ai.

Trustible. AI Governance Activities. 2025. Available at trustible.ai.

About Trustible



Who We Are

Trustible is a purpose-built AI governance platform that helps organizations inventory, assess, and oversee their AI systems. The platform connects AI inventory management, automated risk assessment, compliance tracking, vendor evaluation, and governance workflows into a single system of record. Trustible is headquartered in Arlington, Virginia.

What We Do

AI Governance Platform for Risk and Compliance Teams
Trustible gives governance teams the structure, intelligence, and automation to govern AI with confidence. From intake to ongoing oversight, the platform connects inventory, risk assessment, compliance frameworks, vendor evaluation, and reporting into a single system of record.

Built for second-line professionals. Ready for the scale of enterprise AI.

AI Inventory
Centralized record of all AI use cases, models, agents, and third-party vendors.

Risk Management
Automatic risk scoring, risk registers, and mitigation recommendations.

AI Compliance Frameworks
EU AI Act, NIST AI RMF, ISO 42001, and more. Document once, comply at scale.

Automated Workflows
Intake reviews, impact assessments, and periodic oversight orchestrated across your team.

Insights Taxonomies
Continuously updated risk, incident, and regulatory intelligence built into every workflow.

Agentic Governance
Purpose-built AI agents for intake automation, intelligent triage, document analysis, and workflow orchestration.

Leading organizations choose Trustible

leidos Guardian OLYMPUS MOLSON COORS beverage company evertec nuix KROLL

Trustible

Website	www.trustible.ai
E-mail	contact@trustible.ai
HQ address	1201 Wilson Blvd, Floor 25 Arlington, VA, USA 22209