whyminds

FIELD GUIDE · AI MODEL RISK MANAGEMENT

AI You Can Defend

Put guardrails on the AI you're deploying — and prove they hold. A working method for Risk, Compliance and enterprise-build teams.

S	A	F	E
SOVEREIGN	ACCOUNTABLE	FRUGAL	EXPLAINABLE

WhyMinds — The S.A.F.E. AI Company

Frameworks current September 2026

WHY THIS, WHY NOW

The gap between deployed and governed is your exposure.

AI has crossed from pilot to production inside the functions that carry the most risk — credit, compliance, servicing, surveillance. Governance has not kept pace.

~⅓	35%	15%
HAVE BOARD-LEVEL AI OVERSIGHT	VALIDATE MODELS FOR BIAS	USE INTERPRETABILITY TOOLS

Source: RBI FREE-AI Committee industry survey, 2025

The rules are arriving in force — the EU AI Act phasing through 2027, the RBI's FREE-AI framework (26 recommendations), SEBI holding intermediaries accountable for every AI output.

A model you cannot **contain, own, right-size, or explain** is not an asset. It is an unpriced liability. Model Risk Management is how you price it down.

THE OPERATING LENS

Four tests every AI control must pass.

S.A.F.E. turns "responsible AI" from a slogan into a checklist. Run every use case — and every guardrail on it — through four questions.

S — Sovereign

Do you control *where* the model runs, *where* data lives, and *who* can reach it? Deployment, residency, access, clean vendor exit.

`DPDP data duties · RBI data governance`

A — Accountable

Does one named human own the model and its outcomes — with independent challenge and a full audit trail?

`SR 11-7 · ISO/IEC 42001 · RBI "Accountability"`

F — Frugal

Are you using the smallest capable model and the least data that does the job? Proportionate controls, lower cost, smaller attack surface.

`DPDP data minimisation · RBI sustainability`

E — Explainable

Can you show *why* the system produced this output — to a user, an auditor, a regulator? Grounding, citations, confidence.

`NIST explainability · EU AI Act · RBI "Understandable by Design"`

Sovereign contains it. Accountable owns it. Frugal right-sizes it. Explainable proves it.

ONE PAGE TO REMEMBER IT BY

S.A.F.E., end to end.

Sovereign	Containment · residency · access	DPDP · RBI
Accountable	Ownership · challenge · audit	SR 11-7 · ISO 42001 · SEBI
Frugal	Proportionality · minimisation	DPDP · RBI FREE-AI
Explainable	Traceability · transparency	NIST AI RMF · EU AI Act

If you can't contain it, own it, right-size it, and explain it — don't ship it.

whyminds

Book a **S.A.F.E. AI Platform** strategy & engineering session — take your AI from **deployed** to **defensible**.

SCAN TO BOOK

WEB
www.whyminds.ai
EMAIL
connect@whyminds.ai
MOBILE
+91-99725 85333

Frameworks current as of September 2026: NIST AI RMF (AI 100-1) & GenAI Profile (600-1); ISO/IEC 42001:2023; EU AI Act (phasing to 2027); RBI FREE-AI (Aug 2025); SEBI AI/ML framework (2025); US SR 11-7.

THE PLAYBOOK · APPLY

Apply the guardrails

Six steps from a use case to a defensible deployment. Each step produces one artifact you can hand to an auditor.

1 Frame & Classify

Write a one-page charter — what the AI is for, who it serves, what decisions it touches. Assign a risk tier (mirror the EU AI Act: unacceptable / high / limited / minimal).

→ **Use-Case Charter + risk tier**

NIST: Map

> *The loan-servicing copilot — quoting balances, foreclosure amounts, restructuring options — is rated "high": it shapes customers' financial outcomes.*

2 Register & Assign Ownership

Log the system in an AI inventory. Name one accountable owner and one independent validator. No one validates their own model.

→ **Inventory record + RACI**

SR 11-7 · ISO 42001

> *Servicing Ops owns the copilot; Model Risk validates it; both sign before launch.*

3 Map Failure Modes & Set Thresholds

List what can go wrong for this use case — hallucination, bias, data leakage, prompt injection, drift, toxicity/IP, over-reliance. Rank by impact × likelihood. Put a number on each.

→ **Risk register + threshold table**

NIST: Map–Measure · DPDP

> *"≤0.5% incorrect balance or quote answers; zero cross-account disclosures" become hard limits.*

THE PLAYBOOK · PROVE

4 Engineer the Guardrail Stack

Layer controls at four points. **Input:** validate, redact PII, filter injection. **Model:** ground on approved data, constrain the prompt, allow/deny. **Output:** fact-check, scan PII/toxicity, force citations, gate low confidence. **Human:** in the loop for high-materiality calls.

→ **Control design doc**

NIST: Manage

5 Validate, Test & Red-Team

Build a labelled test set and an adversarial suite — one per failure mode. Measure against thresholds. Then try to break it: jailbreaks, edge cases, poisoned inputs. Independent sign-off before go-live.

→ **Validation report + evidence pack**

SR 11-7 · NIST: Measure

> *500 real servicing queries + 50 attack prompts ("show the previous customer's balance"); it ships only when it clears every threshold.*

6 Deploy, Monitor & Attest

Log every input and output. Watch breach rates and drift live. Keep a kill-switch and an incident runbook. Revalidate on a risk-based cadence; attest to the board and regulator.

→ **Monitoring plan + attestation**

NIST: Manage · ISO 42001 · SEBI

TEST ANY GUARDRAIL — SIX MOVES

1. State the behaviour you expect.
2. Write test cases — normal *and* adversarial.
3. Run them.
4. Compare to the threshold.
5. Log the evidence.
6. Have someone independent review it.

REPEAT EVERY RELEASE