VIKRAM JHA - DEFENSIBLE AI
SCHEMA

AGENT INVENTORY SCHEMA

Version 1.0 - 24 July 2026
https://vikramjha.work/artifacts/agent-model-inventory-schema

The model-risk inventory record, rewritten for a system that acts rather than one that scores.

WHY IT EXISTS

Model risk guidance was written for models that score and predict, and the inventory field set follows from that: a model has inputs, a methodology, an owner and a validation date. An agent has none of those cleanly. It chains tools, prompts and sub-models at runtime, its behavior changes the day a provider ships a new version with no code change on your side, and the thing that creates exposure is not the score but the action taken on it. Most MRM inventories cannot represent that, so agents get recorded as one row called 'the assistant' and the inventory stops being a control. These are the fields that make the row governable, and what each one is actually for.

WHAT IT CONTAINS

1. Nineteen fields, each with the failure it exists to prevent
2. Why the equivalent classic MRM field breaks when the system acts rather than predicts
3. The evidence that makes each field true rather than merely populated
4. Risk tiering driven by action class and reversibility, not by model size or spend
5. The revalidation trigger matrix - including the provider-side changes that fire no internal event

INVENTORY FIELDS

AGENT IDENTIFIER

What it records: Stable ID, distinct from the application and the model.
Why the classic MRM field fails here: Model IDs get reused across agents; application IDs cover many agents.
Evidence that makes it true: The same ID appears on the action records.

ACCOUNTABLE HUMAN

What it records: A named person, not a team or a product.
Why the classic MRM field fails here: MRM names a model owner; nobody owns the agent's authority.
Evidence that makes it true: A signed acceptance naming the individual and the date.

PURPOSE

What it records: What it is permitted to do, in business terms.

Why the classic MRM field fails here: Model purpose describes prediction; agents act across purposes.
Evidence that makes it true: Purpose statement that the enforcement layer can actually test against.

ACTION CLASSES
What it records: Every class it can reach, per the action-class taxonomy.
Why the classic MRM field fails here: No equivalent field exists in classic MRM. This is the field the exposure follows.
Evidence that makes it true: Capability list reconciled against the tools it can call.

AUTHORITY GRANT REFERENCE
What it records: The grant that authorized each action class, and who issued it.
Why the classic MRM field fails here: MRM has no concept of delegated authority.
Evidence that makes it true: A machine-readable grant, scoped and time-boxed.

SCOPE LIMITS
What it records: Resource, value, rate, data-classification and jurisdiction bounds.
Why the classic MRM field fails here: Model limits are about input ranges, not about how much the system may spend.
Evidence that makes it true: Limits enforced at the decision point, evidenced by a blocked action.

EXPIRY AND REVIEW
What it records: When authority lapses and what re-approval requires.
Why the classic MRM field fails here: Models are revalidated annually; grants routinely never expire.
Evidence that makes it true: Validity window with a recorded re-approval event.

TOOLS AND INTEGRATIONS
What it records: Every tool, API and system of record it can invoke.
Why the classic MRM field fails here: MRM records data inputs, not action surfaces.
Evidence that makes it true: Reconciled against runtime, not against the design document.

MODELS AND VERSIONS
What it records: Every model, with an immutable version - never a moving alias.
Why the classic MRM field fails here: 'GPT-class model' is not a version. A provider alias changes underneath you.
Evidence that makes it true: Pinned identifiers appearing on individual action records.

PROMPT AND CONFIGURATION VERSION
What it records: The prompt, system message, tool schema and settings in force.

Why the classic MRM field fails here: No MRM analog. In agents this is where most behavior lives.
Evidence that makes it true: Version identifier resolvable to the exact text at a past date.

DATA DOMAINS
What it records: Classifications it can reach, including anything reachable transitively.
Why the classic MRM field fails here: Model input schemas are fixed; agent retrieval is open-ended.
Evidence that makes it true: Access evidence at the volume the agent actually generates.

JURISDICTIONS
What it records: Where the data subjects are and where inference physically runs.
Why the classic MRM field fails here: Rarely recorded at all, and provider routing crosses borders invisibly.
Evidence that makes it true: A traced request path, not a contractual assurance.

ENFORCEMENT POINT
What it records: Where policy can refuse the action, between intent and system of record.
Why the classic MRM field fails here: Classic controls run at authentication and never again.
Evidence that makes it true: An allow and a deny test, both captured.

EVIDENCE STORE AND RETENTION
What it records: Where action evidence lands and how long it survives.
Why the classic MRM field fails here: Application log retention is assumed sufficient and almost never is.
Evidence that makes it true: Retention schedule mapped per evidence class to the binding regime.

VALIDATION STATUS
What it records: What was validated, when, by whom, and its limitations.
Why the classic MRM field fails here: Validating the model is not validating the agent's authority.
Evidence that makes it true: An independent report covering authority, tools and containment.

CHANGE TRIGGERS
What it records: What forces revalidation.
Why the classic MRM field fails here: Provider updates fire no internal event, so nothing triggers.
Evidence that makes it true: A trigger matrix with evidence it has fired at least once.

MONITORING THRESHOLDS
What it records: Metrics tied to action risk, not to generic output quality.
Why the classic MRM field fails here: Model drift is slow; agent behavior changes overnight.
Evidence that makes it true: Threshold definitions with recorded alert dispositions.

CONTAINMENT PATH
What it records: How this agent is stopped mid-incident.
Why the classic MRM field fails here: The MRM answer is decommissioning, which takes a release.
Evidence that makes it true: A tested kill-path exercise with a recorded time-to-stop.

LAST RECONSTRUCTION TEST
What it records: Date an independent party rebuilt why one action was permitted.
Why the classic MRM field fails here: No MRM analog, and it is the field that predicts examination outcome.
Evidence that makes it true: A completed sample with reviewer sign-off.

RISK TIERING
Tier on what the agent can do and how reversible it is - not on model size, spend, or how advanced it sounds. Reversibility is the axis most schemes omit and the one that matters during an incident.

TIER 1
Trigger: Reaches Transact, Configure or Deny/terminate without a human in the path.
What the tier should compel: Independent validation, enforced limits, tested containment, quarterly reconstruction sampling.

TIER 2
Trigger: Reaches Commit, Mutate record or Communicate externally.
What the tier should compel: Validation of authority and evidence, monitored thresholds, annual reconstruction test.

TIER 3
Trigger: Assert or Recommend only, with evidenced human review.
What the tier should compel: Scoped entitlements, oversight evidence, change-trigger monitoring.

TIER 4
Trigger: Retrieve only, inside one classification, no external output.
What the tier should compel: Entitlement scoping and access evidence. Re-tier the moment a tool is added.

REVALIDATION TRIGGERS

The first three fire outside your change control. If nothing in your process detects them, the inventory is stale by default rather than by exception.

PROVIDER SHIPS A NEW MODEL VERSION OR RETIRES ONE
Why it matters: Behavior changes with no code change and often no notice you have subscribed to.

A MOVING ALIAS RESOLVES TO SOMETHING NEW
Why it matters: You did not change the version; the version changed under the alias.

A TOOL OR DOWNSTREAM API CHANGES ITS CONTRACT
Why it matters: The agent's reachable action set changes without any deployment on your side.

A PROMPT, SYSTEM MESSAGE OR TOOL SCHEMA IS EDITED
Why it matters: In agents this is a behavioural change of the same order as a retrain.

A NEW TOOL OR INTEGRATION IS ADDED
Why it matters: Can move the agent two action classes and a whole risk tier.

SCOPE, LIMIT OR ENTITLEMENT IS WIDENED
Why it matters: The grant no longer matches what was validated.

VOLUME CROSSES A BAND
Why it matters: Controls calibrated for hundreds of actions rarely hold at hundreds of thousands.

A REGIME THE AGENT TOUCHES IS AMENDED
Why it matters: The evidence you owe changes even though the system did not.

HOW TO USE IT
1. Populate one agent completely - intake through evidence - before rolling the schema out. The first row surfaces most of the structural problems.
2. Fill the evidence column with links, not descriptions. A field describing evidence you cannot produce is a gap recorded as a control.
3. Reconcile tools and models against runtime rather than against the design document - the two diverge quickly.
4. Tier on action class and reversibility, then let the tier set the validation and monitoring obligations.
5. Wire the trigger matrix to something that actually fires. A trigger nobody receives is not a control.

Built from public standards and general practice. No client information appears anywhere in this document. Regulatory instruments move - check anything cited here against its primary source on the day you rely on it.

Free to use and adapt; attribution welcome, not required.