



USE CASE BRIEF · WHAT TEAMS USE IT FOR, AND WHAT THEY GET BACK

AI and Agents on the Data Platform

Knowing which data your models and agents reach, deciding what each request should be allowed to return, catching the sessions nobody authorized, and controlling them — from inside the platform that holds the data.

PREPARED BY	DOCUMENT	DATE
Theom	Public reference version	August 2026

The gap this closes

AI programs reach production faster than the controls around them. The question that arrives at the review — which of our data does this touch, and who is really behind the request? — is a data question, and most AI security tooling is not positioned to answer it.

- **Tools above the data cannot see the data.** A prompt filter inspects text. It cannot tell you whose data a request touched, how sensitive that data was, or whether the identity behind the request should have reached it. Those facts live in the platform, alongside classification, grants and lineage.
- **Agents inherit human credentials.** A coding assistant or analyst copilot connected with a person's credentials looks exactly like that person to the platform. Every access review you run will confirm the person had permission, and none will notice that a model was driving.
- **AI surfaces multiply faster than review cycles.** An in-store engine switched on for a business unit, a new model-serving endpoint, an MCP server exposing a schema — each is a new path to the same data, created in an afternoon and inventoried at the next audit, if at all.

In one sentence: Theom runs inside your data platform, where classification, identity, lineage and policy already sit, so **when a model or an agent reaches your data we already know the data, the identity and the policy** — and can inventory it, alert on it, or evaluate the request inline and shape what comes back.

01 See, decide, catch, control

Four questions, in the order a team faces them once AI is in front of real data.

STAGE	THE QUESTION	HOW IT IS ANSWERED
See	What AI is running against our data, and what does it reach?	An inventory built from observed activity inside the platform: which models are in use, which functions call them, which model-serving endpoints exist, and which tables, views and columns have reached an AI service — with sensitivity tags attached.
Decide	What should each request be allowed to return?	Policy is expressed against the identity and authorization behind the request and the sensitivity of the underlying data — not against the wording of the prompt. Combinations that should never be joined are treated as such.
Catch	What is happening that nobody authorized?	Sessions where an agent is executing queries under a human's credentials. External network calls made by agents on a user's behalf, with the data and volume involved. Consumption and access patterns per identity, including geography.
Control	What happens at the moment of the request?	Either the finding routes to your security team, ticketing and SIEM; or the request is evaluated inline and the response is allowed, redacted, shaped or blocked. The next sections set out exactly where Theom sits in each case.

02 How it plays out

A short scenario, and nothing in it requires anybody to behave badly.

- 1 **An engineer connects a coding assistant** to the warehouse to speed up a data task, using their own credentials because that is what the tool asks for.
- 2 **The assistant explores.** It reads schemas, samples tables and writes queries — far more of them, and far faster, than a person would.
- 3 **One query joins customer records with support transcripts.** Both are approved for the engineer to read. Together they are more sensitive than either, and no policy anticipated the combination.
- 4 **To the platform, this is the engineer working late.** The credentials are valid, the grants are real, and every access review will agree.

- 5 **Theom sees a session driven by an agent**, ties it to the classified columns it touched and the identity behind it, and raises it against named objects with the owner attached.

What happens next depends on the mode you have chosen.

Observability. The session completes. Your team receives the finding within minutes — the agent, the person, the objects, the sensitivity, the volume — and decides whether to revoke, restrict or educate. Nothing sits in any request or query path.

Data-layer enforcement. The classification Theom maintains drives your platform's own row, column and masking policies, so the assistant inherits exactly the masking the engineer would get on sensitive columns. Your platform does the enforcing.

Inline AI control. On the AI surfaces you nominate, Theom evaluates the request against identity, data sensitivity and policy before the response is returned, and allows, redacts, shapes or blocks it. Here Theom is part of the request cycle, and whether it fails open or fails closed is your configuration.

03 What teams use it for

■ Agents running under human credentials

The situation. Coding assistants, analyst copilots and connected agents authenticate as the person who set them up. The platform sees a valid session for a valid identity. Volume, timing and query patterns are the only clues, and nobody is watching for them.

What Theom does. Detects sessions where an AI agent is executing queries using a human user's credentials, from inside the platform, and joins them to what was read — objects, classifications, volumes — and to the identity behind the session.

What you get. The finding that no data platform produces on its own, usually within days of deployment: which assistants are already reaching production data, as whom, and what they read.

■ Which data is reaching AI services

The situation. Model calls are made from functions, notebooks, applications and in-store engines. Each one selects data. Nobody holds the union of those selections, so "which of our sensitive data has reached a model?" becomes an engineering survey.

What Theom does. Identifies the tables, views and columns that are being sent to AI services with sensitivity tags attached, alongside which models are in use, how often, which functions call them, who has access, who actually used them, consumption per identity, and the geography of that access.

What you get. A standing answer to the exposure question, and a cost and access picture per model and per identity that the platform's own billing view does not give you.

■ Governing what a model or agent can reach

The situation. An agent acts on behalf of many people with different entitlements. Filtering the prompt does not help: the same question is legitimate for one requester and not for another, and the difference lies in the data and the authorization, not the wording.

What Theom does. Governs what a model or agent can reach according to the identity and authorization behind each request, shapes or redacts the response using the sensitivity of the underlying data, and prevents combinations that should not be joined even where each part is permitted. For retrieval architectures the same policy applies at index time, so content the requester is not authorized to see never enters the retrieval context in the first place.

What you get. One policy that follows the data rather than the interface, applied consistently whether the requester is a person, an application, an agent or another agent in a chain — and one model serving every role, rather than a fine-tuned copy per business unit.

■ **Model, endpoint and platform assurance**

The situation. A new model-serving endpoint is a new external API surface over your lakehouse data, and it can be created in minutes. Embeddings persist copies of source content. Provider configuration drifts.

What Theom does. Monitors creation and modification of model-serving endpoints, checks endpoint configuration and embeddings storage against benchmark expectations, and tracks large-language-model provider management — mapped to the CIS benchmarks for Snowflake and Databricks and to MITRE tactics.

What you get. Notice of new AI surfaces as they appear rather than at audit, and configuration evidence in the same form as the rest of your platform posture.

■ **Response quality and grounding**

The situation. An agent that answers confidently and wrongly is a business risk even when no data leaked. Prompts drift from the standards they were written against, and nobody reads every response.

What Theom does. Checks agent prompts for conformance against your defined standards, and analyzes request-and-response pairs with a grounding check that flags fabrication, contradiction and over-extrapolation — including forward-looking claims the underlying data does not support.

What you get. Evidence about response quality in the same place as the evidence about data access, rather than in a separate tool with a separate view of the world.

04 What you receive

- **An AI and agent inventory** — models in use, functions calling them, serving endpoints, in-store engines and connected agents — refreshed from observed activity.
- **An exposure view:** the tables, views and columns reaching AI services, with sensitivity attached.
- **Shadow-AI findings:** sessions where an agent is operating under a human identity, with the objects touched and the volume transferred.
- **Compound-identity traces:** the initiating user through the application, agent or service account, to the query, to the data, and onward into downstream sharing destinations.
- **Consumption and access reporting** per model and per identity, including credits and geography.
- **Platform assurance findings** for AI surfaces — endpoint creation and configuration, embeddings storage, provider management — mapped to CIS and MITRE.

- **Response-quality findings:** prompt conformance, and grounding checks for fabrication, contradiction and over-extrapolation.
- **Control, where you want it:** tag-driven enforcement through your platform's own policies, and inline evaluation of AI requests on the surfaces you nominate.

Coverage: in-store engines including Snowflake Cortex, Databricks Genie and Salesforce Agentforce — with agent-to-agent chains — plus model providers including Anthropic and OpenAI, agents connecting in with credentials, vector stores, and MCP. In-house and custom applications are covered through OpenTelemetry instrumentation, so your own services are governed on the same terms as the ones you bought. One deployment, one policy model.

05 Three modes, one deployment

The distinction between the middle and right-hand columns matters, and it is the one most often glossed over in this market.

	OBSERVABILITY	DATA-LAYER ENFORCEMENT	INLINE AI CONTROL
Where Theom sits	Alongside. Nothing in any path.	Outside the query path. Tags drive your platform's controls.	Inline in the AI request cycle, on the surfaces you nominate.
What changes	Nothing. Findings and inventory only.	Your own row, column and masking policies act on current labels.	The response can be allowed, redacted, shaped or blocked.
Failure mode	None — no dependency.	None introduced by Theom; the platform enforces.	Theom is a dependency in the request path. Fail-open or fail-closed is your choice.
Performance	No effect.	No effect on query performance or availability.	Adds minimal latency to the AI request; we benchmark against your own configuration during the evaluation.
Who signs off	The data or AI platform team.	The platform owner as well.	The platform owner and the service owner for that AI surface.

Most teams begin in observability, because the inventory and the shadow-AI findings are what a review actually asks for and there is nothing to approve. Data-layer enforcement follows once labels are trusted, and

it introduces no new failure mode: your platform is still the thing enforcing. Inline AI control is a deliberate, per-surface decision, and the fail-open or fail-closed question belongs to you rather than to us.

06 Where it runs, and what crosses the boundary

Theom is deployed inside your account and does the classifying, observing and validating there, on your compute, through a scoped read-only identity. The control plane governs, reports, alerts and remediates, and is reached through the UI, the API or MCP.

- **Customer data and query text never leave.** Classification reads and activity analysis run where the data and the logs already are.
- **Findings stay in your datastore.** They are written to a dedicated, encrypted Theom database inside it — yours to query, audit and retain on your own terms.
- **Real object names are resolved for display, never retained.** Names cross at render time, on demand, under your own authentication, by asking your Theom deployment to resolve them. What sits outside your environment is references — identifiers standing in for schema, table and column names — and the mapping never leaves.
- **The control plane can be region-pinned,** and where the requirement is absolute it can be deployed entirely inside your own AWS or Azure account.

Deployment is the same shape on either platform: a scoped read-only identity, dedicated compute so the cost is visible to you, results held in your account, and a scheduled job that keeps the picture current. No agent on a host, and no copy of your data. The companion architecture deck sets out the object-level detail per platform.

07 What a proof of value looks like

- **One real AI surface, in production.** A named agent, engine or endpoint reaching real data — not a sandbox, because agent behavior in a sandbox tells you nothing.
- **A bounded subset of production.** One schema, catalog or workspace, so the scope stays measured in weeks.
- **Observability first.** Inventory, exposure and shadow-AI findings need no approval and no path change, so the value arrives before the architecture conversation.
- **Success criteria agreed in writing before it starts** — the specific findings that would count, so the result is judged against your expectations rather than a feature list.
- **One team to work with,** and a named owner for the findings on your side.

A reasonable first step: name one agent or one in-store engine that already reaches production data, and let us run in observability mode for two weeks. Judge the result on three things — the data it reached that you did not expect, whether any session turned out to be an agent acting as a person, and whether the inventory is the artifact your next review needs.