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

Sensitive Data in the Cloud Data Platform

Knowing what is there, deciding who should reach it, catching the access that should not have happened, and acting on it — without data leaving your platform.

PREPARED BY	DOCUMENT	DATE
Theom	Public reference version	August 2026

The gap this closes

Cloud data platforms have good access controls. What they do not have is an answer to the question an auditor, a regulator or an incident actually asks: was this use appropriate? Three things make that hard to answer in a large estate.

- **Entitlements describe possibility, not behaviour.** An access review proves an identity could reach a table. It says nothing about whether the reads that followed made sense — particularly where a shared service account, a BI tool or an agent sits between the person and the data.
- **Classification decays the moment it finishes.** Pipelines land new columns weekly. A quarterly scan describes an estate that no longer exists, and the labels the platform enforces on are only as current as the last pass.
- **Grants outlive the reason they were issued.** A broad grant is still the fastest way to unblock a team on a Friday, and almost nothing in the platform notices when the data underneath that grant changes.

> **In one sentence:** Theom runs inside your data platform, keeps classification current as the data changes, reconciles it against who can actually reach it, watches what those identities do, and then either **enforces through the platform's own controls or alerts on what it sees** — your choice, and the same deployment either way.

01 Know, decide, catch, act

Four questions, in the order a security team has to answer them. Most tools answer one.

STAGE	THE QUESTION	HOW IT IS ANSWERED
Know	What sensitive data is in the platform, where did it land, and who owns it?	Classification runs inside the platform on your own compute, on a schedule, through a dedicated read-only identity. Results are written back as tags in your catalog, so the labels the platform enforces on are current.
Decide	Who should be able to reach it, and who actually can?	Grants, roles, group membership and ownership are reconciled against the classification. Over-broad and inherited grants, dormant privileges and shared identities surface as findings against named objects, not as a score.
Catch	What did those identities actually do?	Observed activity is baselined per identity and joined to classification and ownership. Alerts fire on first access to a sensitive object, on volume and pattern change, and on reads by identities with no business relationship to the data.
Act	What happens next, and who has to be in the room?	Either the tag drives a native platform control — row and column policies, masking — or the finding goes to your security team, your ticketing system and your SIEM.

02 How it plays out

The shortest true story we can tell. Nothing in it requires anybody to behave badly.

1. **A table is created** in a working schema, by a team that owns its own space.
2. **A broad grant is issued** to unblock a downstream job. Reasonable at the time. Nobody revisits it.
3. **A pipeline lands company-confidential invoice data** into that table. The data changed; the grant did not. The platform has no reason to object.
4. **Theom classifies the new columns on the next run** and reconciles them against the grant. Confidential content, world-readable access — a mismatch, raised against the object, with the owner attached.

What happens next depends on the mode you chose.

> **Enforcement.** The tag Theom wrote drives a native policy. Row and column access controls restrict what the intern's role can return, and the sensitive rows are quarantined before anyone reads them. The platform does the blocking; we supply the label it blocks on.

> **Observability.** The grant stands and the intern queries the table. Theom records the access, matches it to the classification, the identity and the absent business relationship, and alerts your data security team with the query context attached — in minutes, not at the next review.

> **The detection is identical.** Both paths depend on the same classification and the same reconciliation. The only difference is whether Theom's finding is allowed to change what the platform returns — a policy decision rather than a product one.

03 What teams use it for

Sensitive data nobody knew was there

The situation. Pipelines land new columns weekly, working copies of production tables accumulate, and unstructured content is parsed into structured tables by jobs whose output nobody re-reviews. The last classification exercise described an estate that has since moved on.

What Theom does. Classifies in place and on a schedule, across structured tables and the outputs of unstructured processing, and writes the results back as tags in your platform's catalog. Where sensitive content is copied or transformed, the classification follows it.

What you get. A current inventory of sensitive data by object, schema and owner — including the tables that no team claims — and labels your platform can act on rather than a report that ages.

Access that was permitted but not appropriate

The situation. Entitlement reviews confirm that an identity had permission. They cannot tell you whether the use that followed made sense, and the picture blurs further when a BI tool, a service account or an agent sits between the person and the data.

What Theom does. Baselines observed activity per identity, resolves the compound path — user through BI tool or agent, through to the object — and joins it to classification, ownership and policy. Alerts carry the query context, not just an event ID.

What you get. A defensible answer to "who read this, and should they have", within minutes of it happening, and a per-identity baseline that makes the second occurrence obvious.

Grants that outlived their reason

The situation. Access was granted broadly to unblock a team, inherited through a role hierarchy nobody has unwound, or issued to a service identity that has since been reused. The data underneath it changed; the grant did not.

What Theom does. Continuously reconciles grants, roles, group membership and ownership against what the data actually contains, and separates the privileges that are exercised from the ones that merely exist.

What you get. A ranked, named list of grant-to-content mismatches with owners attached — the input to a revocation you can defend, rather than an access review that removes nothing.

Evidence that survives an audit

The situation. The questions are always the same: what was sensitive, who could reach it, who did, what changed, and when. Assembling that by hand each cycle consumes the team that should be reducing the risk.

What Theom does. Keeps the record continuously, from the same graph the alerts come from, so the reporting and the detection cannot disagree with each other.

What you get. Reporting produced from evidence rather than reconstructed from memory, and a shorter path through the next regulator question or internal audit.

AI and agent access to sensitive data

The situation. Agents and AI workflows read from the same tables as everyone else, usually under a shared identity, and often on behalf of a person who never appears in the log.

What Theom does. Treats an agent as an identity like any other: classified content, observed reads, baselined behaviour, and the compound path back to whoever initiated it.

What you get. The same access assurance for agent traffic as for human traffic — before it becomes the exception your policy did not anticipate.

04 What you receive

- **A sensitive-data inventory** by object, schema and owner, refreshed on your schedule rather than at audit time.
- **Tags written into your platform's catalog**, so classification is usable by the controls and tools you already run.
- **Findings against named objects** — over-broad grants, dormant privileges, ownership gaps, classification and policy mismatches — ranked, with owners attached.
- **Alerts with context**, delivered into your SIEM and ticketing systems, describing what was read, by which identity, under which grant, and why it stood out.
- **Reporting for audit and regulatory questions**, drawn from the same continuous record.
- **Enforcement, when you want it** — the platform's own row, column and masking controls, driven by tags Theom keeps current.

> **Four categories, one deployment:** activity monitoring, access governance, classification and posture are usually four products, each with its own agent, its own copy of your metadata and its own console. Theom answers all four from inside the platform, which is why the outcome is generally simpler operations as well as better coverage.

05 Two modes, one deployment

	OBSERVABILITY	ENFORCEMENT
What changes	Nothing in the data path. Tags are written; access is unchanged.	Tags drive the platform's own row, column and masking policies.
If a label is wrong	A false alert, triaged and tuned.	A query returns less than it should, and someone is blocked.
Who signs off	The data security team.	The platform owner as well — this changes what production returns.
Typical use	The whole estate, from week one.	The classes where the label is trusted and the cost of over-blocking is low.

These are not competing products or a licensing tier. Most teams run observability first — it has no blast radius, and a few weeks of findings is how classification earns the trust that enforcement requires. Enforcement is then enabled class by class rather than estate-wide. **Theom is never in the query path:** enforcement is your platform applying its own controls to labels we maintain, so nothing we do can affect query performance or availability.

06 Where it runs, and what leaves

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

- **Customer data never leaves the platform.** Classification reads happen in your account, against your storage.
- **Query text never leaves.** Activity analysis runs where 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.
- **What crosses is pseudo keys.** Schema, table and column names are replaced by identifiers before anything leaves, and the mapping back to the real names stays in your environment. When the interface shows you a real name, it has asked your own deployment to resolve it, under your authentication. Theom's platform services are stateless, and the metadata retained outside your environment is non-identifying by construction.
- **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 Snowflake and on Databricks: 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, no host to manage, and no copy of your data. The companion architecture deck sets out the object-level detail for each platform.

07 What a proof of value looks like

- **A bounded subset of production.** Production, so the findings reflect real data and real usage. Bounded to a schema or catalog, so the scope stays measured in weeks.
- **About an hour to deploy.** Objects created, identity granted, first classification running.
- **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:** pick one production schema that concerns you, run in observability mode for two weeks, and judge the result on what comes back — the sensitive data you did not know was there, who could reach it, and who did.