SOLA SECURITY RESEARCH | WHITE PAPER | OCTOBER 2026

Resolving the Enterprise: Continuous Identity and Device Mesh Discovery in the Sola Security Brain

Sola Security Research

EXECUTIVE SUMMARY

Security work is moving to agents. An agent asked to investigate, triage or prioritise does not reason inside one tool. It reasons across the enterprise: from an alert on a laptop, to the person behind it, to everything that person can reach in every system they exist in. Each of those steps is an inference over entities, and in an enterprise the entities are fragmented, one record per tool, with no tool designed to agree on what the others describe. When a link is missing, the agent produces something shaped like it.

We describe how the Sola Security Brain resolves identities and hosts across the enterprise *ahead of inference*, so that every agentic step reads a verified link rather than guessing one. Two entity resolution layers, Identity Mesh and Device Mesh, run continuously on every sync across all connected systems, link records on identifiers the vendors already expose, and produce the graph that agentic inference, risk scoring and blast radius all read from. We demonstrate the capabilities the mesh unlocks on a set of detections that are indistinguishable at the alert level. Reasoning over the resolved enterprise, the agent separates them by who is behind each host and what that identity can reach, ranks them by real organisational risk rather than technical severity, and surfaces coverage gaps that no single alert or tool contains.

Keywords Entity Resolution · Agentic Inference · Identity Security · Device Inventory · Security Context Graph

1 Agentic inference needs a resolved enterprise

An investigating agent works in a loop. It plans, queries, follows what it finds and refines. In security, almost every step of that loop crosses a system boundary. The alert lives in the EDR, the owner lives in the identity provider, the privilege lives in the cloud, the business function lives in HR, and the data lives somewhere else again. The quality of the final answer is bounded by the weakest join along that chain.

The Sola Security Brain is a context layer built for exactly this: a graph of the organisation's security-relevant state, sitting on top of the raw asset inventory, so that any question about the environment can be answered from it, including questions no policy anticipated [1,2]. Its defining choice is where the work happens. Resolving entities across sources, inferring the relationships

between them and computing what reaches what are carried out *before* inference, continuously and over the whole population. The agent still investigates, but it does so over a resolved enterprise rather than against raw APIs.

Everything that layer offers an agent rests on one prior step. Before an agent can infer what an identity can reach, the graph has to know which records describe that identity. Before it can infer whose laptop raised an alert, the graph has to know which records describe that laptop. This paper is about that step, and why it has to be done for the agent rather than by it.

2 Where inference breaks: entities that do not connect

Consider one employee laptop. The EDR knows it as a host with a sensor. The MDM knows it as an enrolled device. The identity provider knows it as a registered device assigned to a user. The Zero Trust agent knows it as a device posture record. Four systems, four rows, and no key that any of them was built to publish to the others.

Identities fragment the same way, and faster. One person is a login in the identity provider, an employee record in the HR system, a member in the chat tool, a committer in source control, a principal in the cloud and an administrator in some SaaS tenant nobody remembers granting. Contractors, service accounts and the agents now acting on behalf of all of them multiply the count, across multi-cloud, hybrid and SaaS estates that were never designed to interoperate.

Deciding which of these records describe the same thing is *entity resolution*. For an agent, every unresolved pair is a hop it cannot take. Reachability, blast radius and ownership are only as complete as the links beneath them, and a path the graph cannot join is a path no agent can infer.

The failure this produces is quiet. A model returns the most likely continuation of its input. Asked whose host this is, or what this identity can reach, and missing the join, it produces something shaped like the join. In the output, a guessed link and a verified one are the same sentence.

> *An invented link might be caught on review. A missing one rarely is.*

Asking the agent to resolve entities itself at question time does not fix this. It moves the most expensive part of the investigation into the inference loop, repeats it on every question, and runs it on whatever sample of records the budget allows. The two usual shortcuts fare no better at enterprise scale: matching on a shared field name assumes a column means the same thing in two vendors, and hand-building an integration for every vendor pair grows with the square of the stack.

3 Resolving ahead of inference

Sola runs entity resolution across every connected system as two mesh layers, part of the modelling layer of the Sola Security Brain (Table 1).

Table 1. The two mesh layers.

	Device Mesh	Identity Mesh
Resolves	Host and endpoint records to the same physical or virtual machine.	Account records to the same person or identity.
Example	A CrowdStrike-protected laptop resolves to its record in the identity provider, the cloud or the MDM, without a hand-built integration for that vendor pair.	A login in the identity provider and a principal in the cloud resolve to the same identity, across every system that identity appears in.
Spans	Endpoint protection, device management, identity-provider device inventories, cloud virtual machines and servers, observability agents, Zero Trust clients and AI coding tools.	Identity providers, cloud IAM and identity centres, HR and finance, collaboration and source control, security tooling, AI platforms, Zero Trust access and business applications.

Both layers match on the identifiers vendors already expose for the purpose, such as a serial number or an assigned-user field, and verify each candidate link against live data rather than assuming it from a shared field name.

3.1 How the mesh discovers its patterns

The matching logic is scalable and automated. Every relationship pattern, the rule that says when a record in one system describes the same entity as a record in another, or when one entity has access to another, is discovered and validated in three stages before it is allowed to run.

1. **Agentic investigation.** An agent explores the schemas and data of the connected systems and proposes candidate patterns: which identifiers one vendor exposes that correspond to identifiers in another, and under what conditions two records should be linked. This is where an agent's breadth pays off. It can survey data shapes across the whole stack far faster than a team could map them by hand.
2. **Data hit validation.** Each candidate is run against live data. A pattern earns its place only if it actually resolves records, and the links it produces hold up when checked against the records on both sides. A pattern that looks right on the schema but finds nothing, or finds contradictions, is discarded or sent back for another round of investigation.
3. **Human in the loop.** New patterns that pass validation are reviewed by a person before they reach production. The reviewer confirms that the link means what the pattern claims, which is the one judgement a hit rate cannot make: two fields can agree across many records and still describe different things.

Once approved, a pattern becomes part of the mesh and runs on every sync, so every match is a link in the graph before any agent asks. The agent's role in resolution is to propose, not to decide. What reaches the graph has been checked against the data and signed off by a person, which is why the agent that later reasons over it can treat each link as a fact rather than a guess.

What is prepared in advance is the substrate, the entities, their links and their scores, not the answer. The agent still investigates on top of it, backs every relationship it reports with a link it retrieved, and

says a connection cannot be established rather than infer one. That is what makes agentic output auditable: each claim names its source, and an absence is reported as an absence.

3.2 At inference: what the agent reads from the Security Brain

Everything above happens before a question is asked. When one arrives, the agent runs its investigation on top of the Security Brain. What it reads there is broader than the resolved entities themselves, but every part of it is anchored by the mesh: the mesh decides which host or which person each fact belongs to.

- **Resolved entities and their metadata.** Which records are the same host or the same person, together with what each source knows about it: ownership, device posture, enrolment state, roles and group memberships. Resolution is what lets metadata scattered across many tools be read as one profile rather than as fragments.
- **Related entities.** Not only the records that are identical, but the entities connected to them: the groups an identity belongs to, the roles it can assume, the workloads and data it can reach, and the devices it signs in from. Identity links and access relationships sit side by side in the same graph.
- **Findings discovered in advance.** Failed controls, blast radius and attack vectors are computed ahead of the question and over the whole population [1]. Blast radius, for example, is built from predefined paths that combine identity links with access relationships: an identity-provider account resolving to its cloud identity, then on through the groups, permission sets and roles that identity can assume, to what those roles can reach.
- **Org context.** Facts no connected system holds, such as who the executives are or which controls every company device must run, entered once and available to the agent alongside everything else.

The mesh is what makes these inputs composable. A failed control on a laptop, an attack vector to a cloud role and an admin grant in a SaaS tenant are three findings in three tools until the mesh says they belong to the same person. Without resolution, each is evaluated alone and ranked by its own technical severity. With it, they add up to one entity's exposure. At inference, the agent does not reconstruct any of this. It reads it.

4 Across the whole enterprise

Neither mesh is scoped to a vendor or a category. Every connected system that holds a record of a machine or a person becomes part of the mesh, and every new integration joins it through the identifiers it already exposes (Figure 1). Each added source is another place an identity or host can be recognised, and another set of links agentic inference can follow.

The consequence for an agent is that no question has to know where to look. It asks about a host or a person, and the graph already holds every system that host or person appears in.

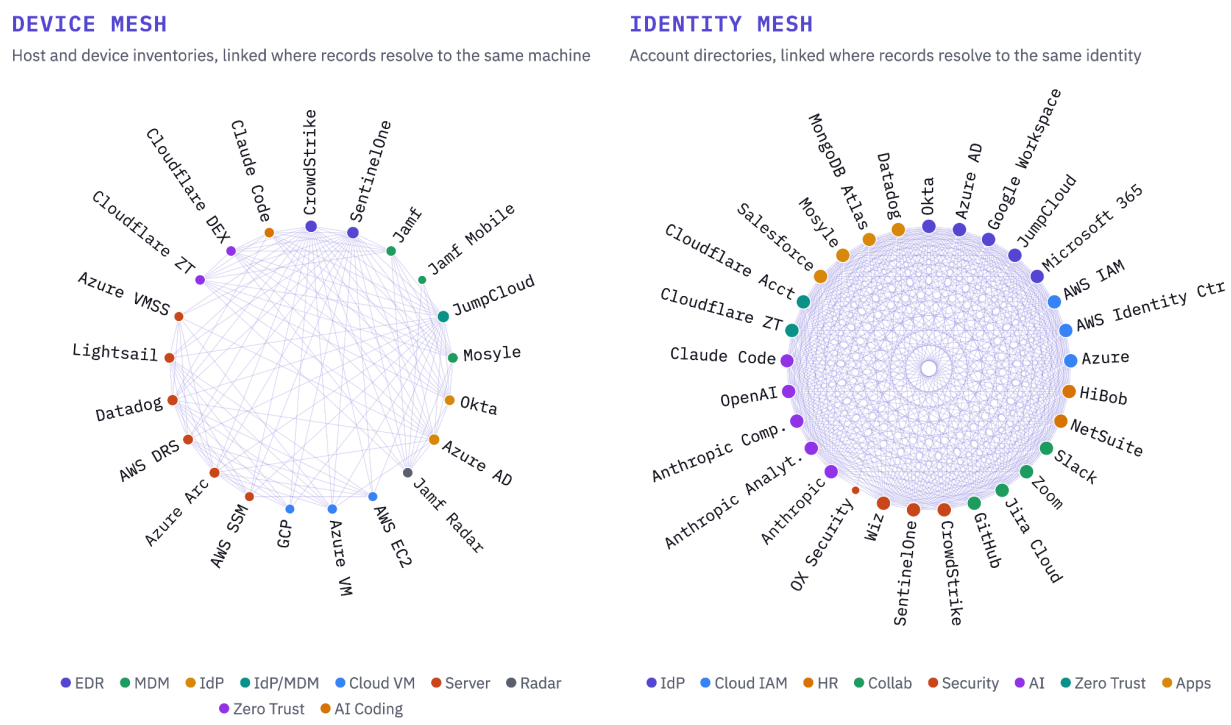


Figure 1. Device Mesh and Identity Mesh: connected inventories, linked where records resolve to the same machine or the same identity.

5 What a resolved enterprise enables for agents

A resolved graph is not a feature in itself. It is what four agentic capabilities depend on.

- **Grounded inference.** The agent reasons from links it retrieved, and says so when a link does not exist.
- **Cross-stack reasoning.** The agent follows a laptop to a person to every system that person can act in, without being told where to join.
- **Contextual prioritisation.** Rankings move with business function, data held, reach and org context, not just the finding.
- **Reasoning about absence.** A record the graph expects and cannot find becomes something the agent can report.

The demonstration below shows all four on a case where the alerts themselves give the agent nothing to work with.

6 Demonstration

To see what the mesh unlocks, we staged a case in which the alerts give the agent nothing to work with. Eleven employees each downloaded a well-known privilege-escalation assessment script to their own laptop, some of them more than once. The result was fifteen real EDR detections with the

same name, the same severity and the same command line. On the alert queue, they are indistinguishable.

We then asked the Sola agent three questions in plain language, with no hint of how to join the data: identify the identity behind each host and every system where either appears; find where else they show up and what they can access there; and rank the incidents by risk to the organisation. The demo video shows the run in full (Figure 2).



WATCH THE DEMO VIDEO

The live run in Sola chat (anonymised)

Figure 2. Demo video: the live run in Sola chat (anonymised).

On the device side, the agent read the Device Mesh rather than rebuilding it. Most laptops resolved to their owners through serial numbers in the identity provider's device inventory, and one that the identity provider had never seen resolved through the Zero Trust agent instead. The same step exposed a gap no alert contained: none of the laptops could be confirmed in the connected MDM, although org context requires every company device to run one.

On the identity side, following each resolved person across the Identity Mesh changed the picture entirely. Behind identical alerts sat administrators of both identity planes, owners of cloud subscriptions, source control, collaboration and AI platforms, a CRM administrator, and an executive whose cloud access reached decrypt permissions on the keys protecting stored data.

The ranking is where entity resolution and org context meet. Resolution attaches each alert to a person and everything they hold; org context, a few facts no system holds, such as who the executives are and which controls every device must run, says what matters. Org context set the floor, marking the executive and the IT owner critical regardless of technical severity. Resolved privilege set the

order: identity-plane control, cloud ownership and business-application admin lifted a detection, and a security engineer reached the critical tier on privilege alone. Fifteen identical detections became three critical, three high and five lower priority, without a rule written for the incident.

7 Key takeaways

- **Inference reads; it does not reconstruct.** Resolution is paid for once, on sync, and reused by every question. An agent rebuilding the joins from raw APIs pays again on every alert, over a sample.
- **One resolved enterprise serves every agentic question.** The links that ranked these detections also answer who owns a workload or what a compromised account could touch. No workflow has to rebuild them.
- **Org context turns resolution into priority.** The mesh says who is behind an alert and what they hold; org context says how much that matters. Together they rank by organisational risk, not technical severity.
- **An absence is a finding.** The mesh expected an MDM record for each laptop, could not confirm one, and org context said one was required. No single tool could report that.
- **Resolution is bounded by what is connected.** It does not manufacture coverage; it completes the coverage that exists and makes its edges visible. Where relationships are resolved ahead of time, agentic answers are decisively more complete [1,3–5].

When an agent reports a connection, it is reading one. When it cannot, it says so.

All hosts, identities and roles in the demonstration are anonymised. The detections were triggered deliberately on employee laptops.

References

[1] L. Goldberg, G. Engelberg, E. Yavin, E. Elouz, A. Zadok, K. Koutsyi. Coding Agents Aren't Enough! Evaluating an Enterprise Security Brain for Agentic Cloud Investigations. arXiv:2609.30345, 2026.

[2] L. Goldberg. Context is the Architecture: Rethinking Security Platforms in the LLM Era. Sola Security, July 2026.

[3] G. Engelberg, L. Goldberg, K. Koutsyi, B. Plotnikov, T. Gilat, B. Benhemo. AI Native Asset Intelligence. arXiv:2605.09115, 2026.

[4] G. Engelberg, K. Koutsyi, L. Goldberg, R. Elezra, I. Pinto, T. Moalem, S. Cohen, Y. Weintrob. Sola-Visibility-ISPM: Benchmarking Agentic AI for Identity Security Posture Management Visibility. arXiv:2601.07880, 2026.

[5] E. Yavin, G. Engelberg, K. Koutsyi, L. Goldberg, G. Baron. Cross-Vendor Sola ISPM Benchmark: Evaluating Agentic AI for Federated Identity Security Reasoning. arXiv:2606.02674, 2026.