

ENGINEERING DECISION-READY SECURITY DATA

A Technical Architecture for Normalization, Noise Reduction, and Intelligence Fusion

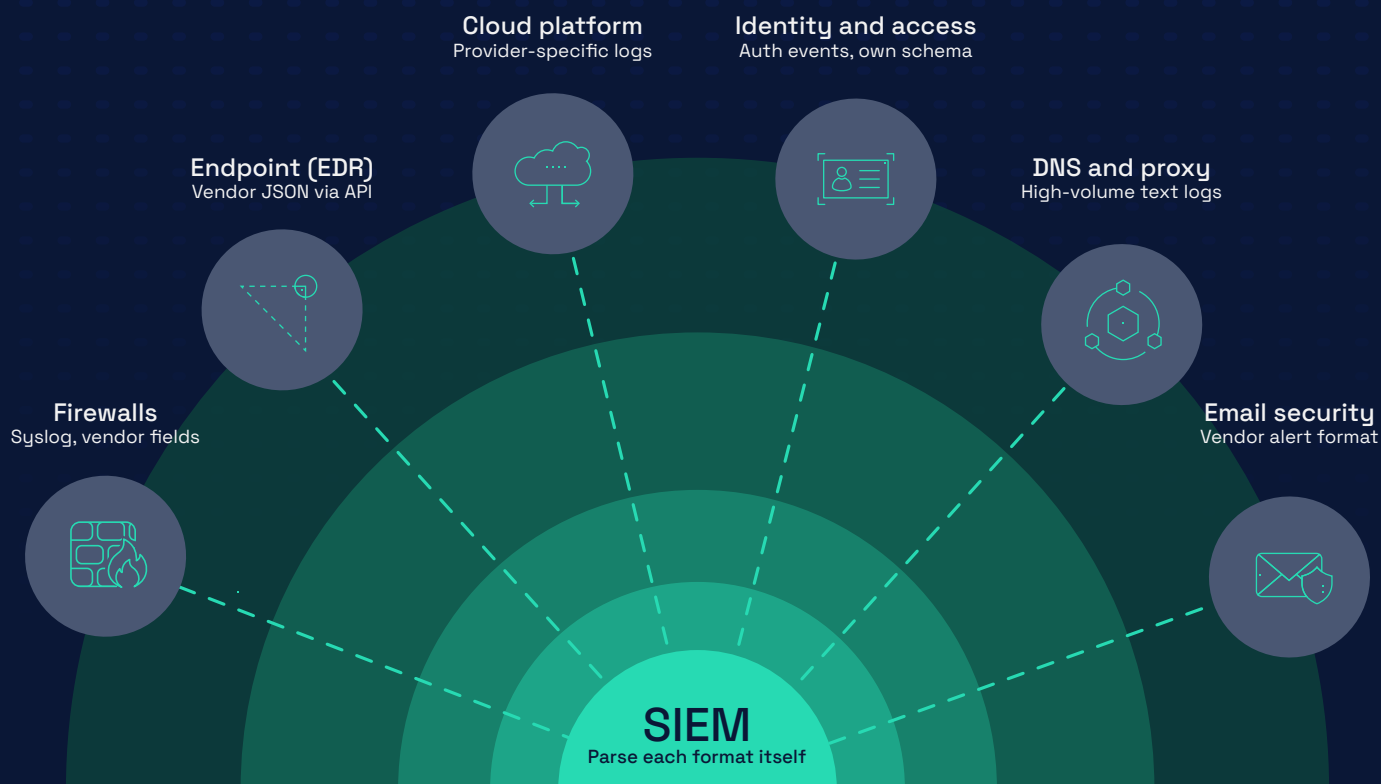
A security team running five or more source systems (EDR, email, identity, network, cloud, applications) is really running five or more different data languages at once. Every vendor defines its own schema, its own field names, its own severity scale, and none of it lines up at the point the telemetry is generated. What lands in the SIEM is duplicated in volume, fragmented in structure, and, in the case of threat intelligence, usually enriched only after an analyst has already gone looking for context that could have been attached automatically.

This paper works through a three-stage pipeline built to fix that at the source: normalization, deduplication, and intelligence fusion, applied before telemetry reaches a person or a downstream system, not after. Along the way it defines fidelity as something you can actually measure at the point data leaves that pipeline, rather than a word vendors use to mean "better." The goal is a description precise enough that a team could build against it, and specific enough to know whether it worked.

THE CURRENT DATA LANDSCAPE

Most security teams operate five or more categories of source systems: endpoint detection and response, email security, identity and access management, network and firewall infrastructure, cloud platforms, and internal applications. Each category comes from a different vendor, and each vendor defines its own event schema, its own field names, and its own severity scale. There is no shared data language at the point telemetry is generated, and there never was one to begin with; these platforms were built independently, for different jobs, by different companies.

In a legacy architecture, all of that telemetry gets routed straight into the SIEM, which then does whatever normalization, correlation, and retention the platform supports natively. SIEM pricing is typically built around ingest volume, so every additional gigabyte carries a direct cost, and long-term retention ends up competing with detection coverage inside the same budget line. Compliance regimes that require multi-year (PCI DSS 4.0, SEC disclosure rules, CMMC 2.0) push directly against that cost structure. The engineering cost shows up too: industry benchmarks put roughly 40 percent of security engineering time into data plumbing, custom parsing, and pipeline maintenance rather than into detection or response work.



WHERE THE ARCHITECTURE BREAKS DOWN

The lack of a shared schema shows up first as a correlation problem. A single user session, recorded by five different source systems, can appear as five structurally unrelated entries with no field in common:

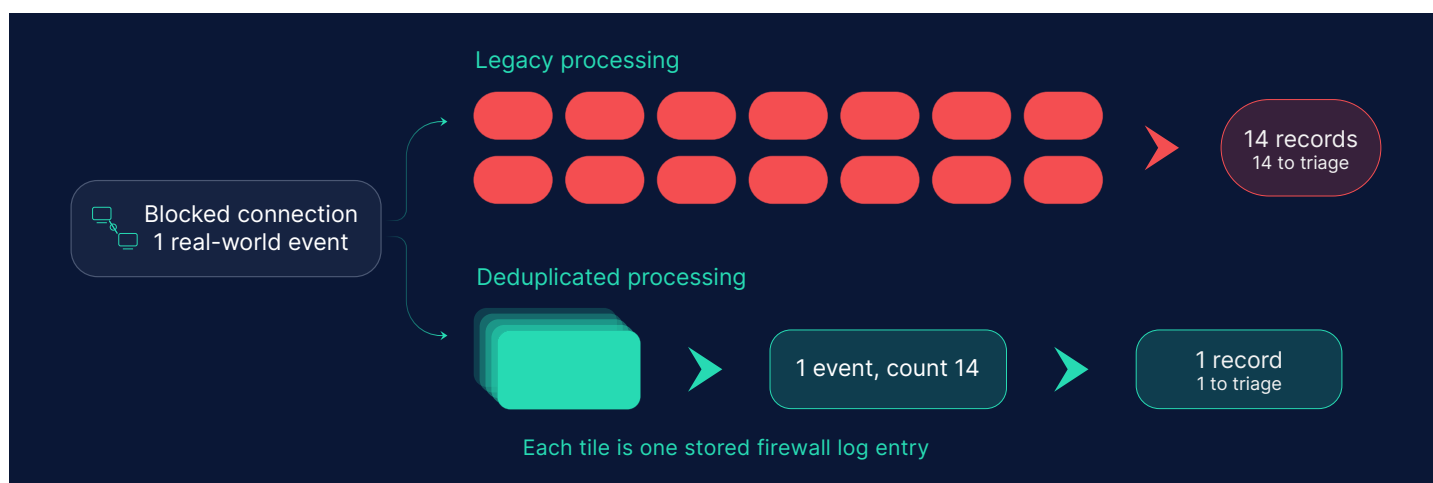
SOURCE	NATIVE LOG ENTRY
EDR	userId: john.smith · process.exe: powershell
Firewall	src_ip: 10.0.0.1 · action: deny
Identity	userId: john.smith · loginResult: fail
Cloud	principalId: arn::iam:john · eventName: auth
Email	sender: jsmith@company.com · verdict: clean

Table 1. Five native representations of a single user session, prior to normalization. No field is shared across all five sources.

Tying these five records together into one incident takes either a purpose-built correlation rule for that exact combination of sources, or a person doing it by hand. Any detection rule that needs to reason across sources has to be authored once per schema, so the same logical rule ends up written, and maintained, five or more separate times as those source schemas change underneath it.

A second problem compounds the first, and it is largely independent of it: duplication and low-value noise arriving at the same rate as genuine signal. One blocked connection attempt can generate more than a dozen near-identical firewall deny events. One brute-force attempt against a single account can produce hundreds of failed login entries inside an hour. One command-and-control beacon can throw off thousands of individual network events, all describing the same underlying activity. Industry research on SOC alert volume, including studies from UnderDefense, Tines' Voice of the SOC, and SANS-affiliated research, puts the daily load at roughly 174 alerts per analyst, with only about 22 percent of that volume rising to the level of genuine investigation. That research doesn't break the remaining 78 percent down by cause, so treat it as a description of overall alert quality, not a precise accounting of how much of it is duplication specifically.

A third problem sits at the intelligence layer, and it's a timing problem more than a data problem. Indicator-of-compromise matches have historically shown up without enrichment: no actor attribution, no confidence score, no attack-technique context. Someone has to add that manually, alert by alert, after the match has already happened. The gap between when a new indicator becomes available and when it's actually operationalized into a working detection has historically run 48 to 72 hours, which is a long time for an indicator to sit unused once it's already known.



WHAT THIS MEANS IN PRACTICE

Left alone, this architecture carries real downstream cost on both the engineering side and the operational side. Schema fragmentation isn't only a detection-authoring inconvenience: the 40 percent of engineering time cited above going to data plumbing is time that isn't going into detection coverage or response tooling. Detection logic authored per schema is also just more brittle by construction. It breaks quietly when an upstream vendor changes a field name, and every new source added to an environment adds one more rule set to maintain instead of extending something that already exists.

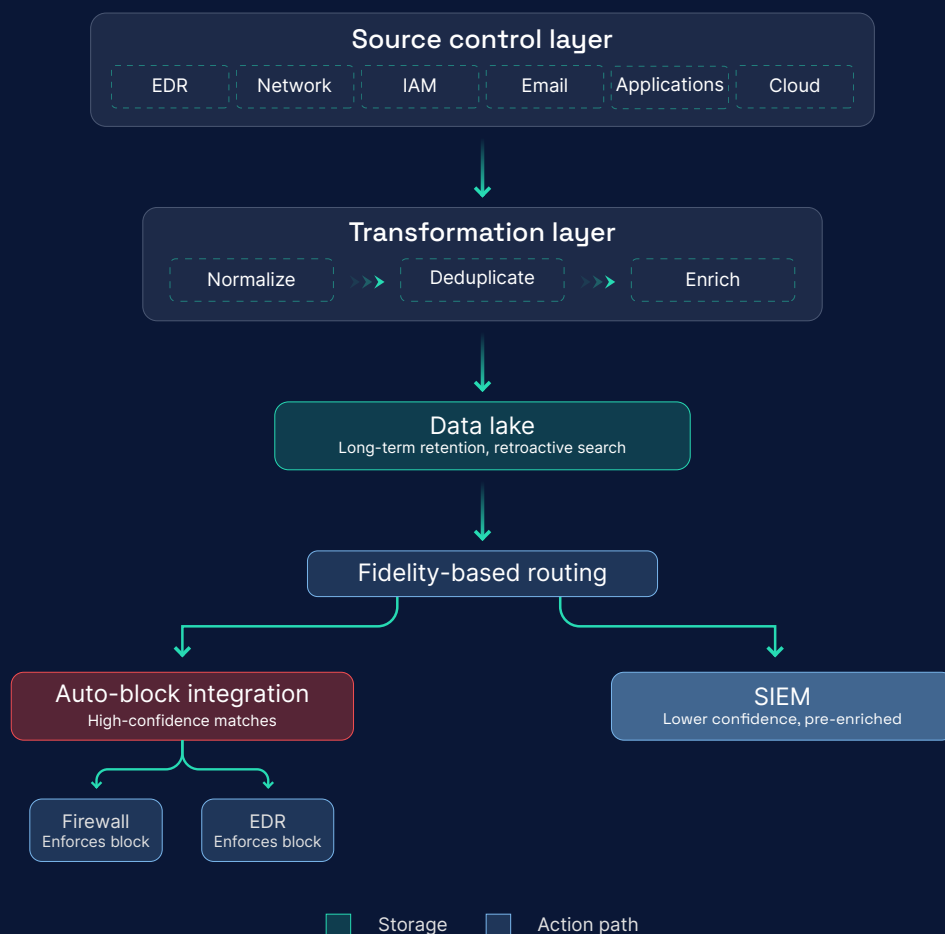
Operationally, the effect of undifferentiated noise is that investigation time goes into reconciling duplicates and rebuilding context that could already have existed at ingest, rather than into analyzing genuine threats. Because high-confidence signal sits in the same queue as low-value noise, mean time to detect and respond

goes up regardless of how good any individual analyst is. And because intelligence enrichment happens after an alert already exists, analysts frequently end up investigating something that was, in principle, already knowable the moment the indicator was published.

Note: A related but distinct set of governance and traceability concerns is emerging specifically around autonomous and agentic SOC tooling. Recent industry surveys report that a majority of organizations have already experienced confirmed agent-related security incidents, and that only a small fraction has formal security approval for agents in production. That's a forward-looking design consideration for anyone evaluating agentic capability, not a claim about how a human-staffed SOC operates today. It matters here because whatever automation gets layered onto a data pipeline later inherits whatever data foundation already exists underneath it, for better or worse.

A THREE-STAGE ARCHITECTURE

The architecture in this paper moves each of the problems above to the point of ingest, before telemetry reaches a person or a downstream system, instead of resolving them after the fact. It's organized into three sequential stages sitting between the source control layer and long-term storage: normalization, deduplication, and intelligence fusion.



DIMENSION	LEGACY ARCHITECTURE	THREE-STAGE ARCHITECTURE
Schema handling	One detection rule per source schema	One rule, correlated natively across sources
Noise handling	Duplicates and low-value events reach the analyst	Duplicates suppressed before the analyst queue
Intelligence timing	Enrichment happens manually, after the alert exists	Enrichment and confidence scoring happen at ingest
Retention economics	Ingest volume drives SIEM cost directly	Fixed hot tier plus lower-cost long-term storage

Table 2. A structural comparison between the legacy path and the architecture described in this paper. Figures for each dimension are detailed in the sections below.

NORMALIZATION

The first stage maps every source to a common schema at ingest, in this case the Open Cybersecurity Schema Framework (OCSF), an open standard with backing from AWS, Splunk, IBM, CrowdStrike, Palo Alto Networks, and more than 200 other organizations. OCSF organizes events into categories, such as Identity and Access Management, Network Activity, and Findings, each built from a defined set of classes and attributes. The Auth_Failure event referenced below, for example, falls under OCSF's Authentication class within the IAM category, which is why it carries a consistent activity_id regardless of which identity provider generated it. The same user session recorded five different ways by five different sources, shown in Table 1 above, gets mapped to one shared vocabulary:

OCSF FIELD	NORMALIZED VALUE
actor.user.name	john.smith
actor.user.uid	jsmith@company.com
device.ip	10.0.0.1
activity_id	Auth_Failure (OCSF 3002)
metadata.product	[all 5 sources correlated]
severity_id	3, Medium (normalized)

Table 3. The same session from Table 1, after normalization to a single OCSF vocabulary.

The practical result: detection logic gets authored once and correlates natively across every mapped source, instead of once per schema.

DEDUPLICATION

The second stage strips redundant and low-value telemetry out before it ever reaches a human:

WHAT ENTERS THE ENVIRONMENT	WHAT REACHES THE ANALYST
Duplicate firewall deny events (14 copies)	1 consolidated firewall event
Repeated failed login noise (200 entries/hour)	1 threshold-based login alert
Beaconing telemetry: 1 threat, 3,000 events	1 high-confidence C2 alert
Cloud API calls: same event repeated	1 deduplicated cloud audit log
EDR noise on known-good processes	Suppressed before it becomes an alert
Low-confidence IOC matches with no context	Enriched and confidence-scored before delivery

Table 4. Deduplication and noise suppression at the pipeline layer, before telemetry reaches an analyst queue.

The underlying reference material positions this layer as the main mechanism for closing the gap between the 174-alert-per-day baseline and the 22 percent that's genuinely investigation-worthy. Fair enough as a design intent, but worth being precise about: the 78 percent figure comes from separate industry research that was never broken out by cause, so this is the effect the design is aimed at, not a number that's been independently verified against it.

INTELLIGENCE FUSION

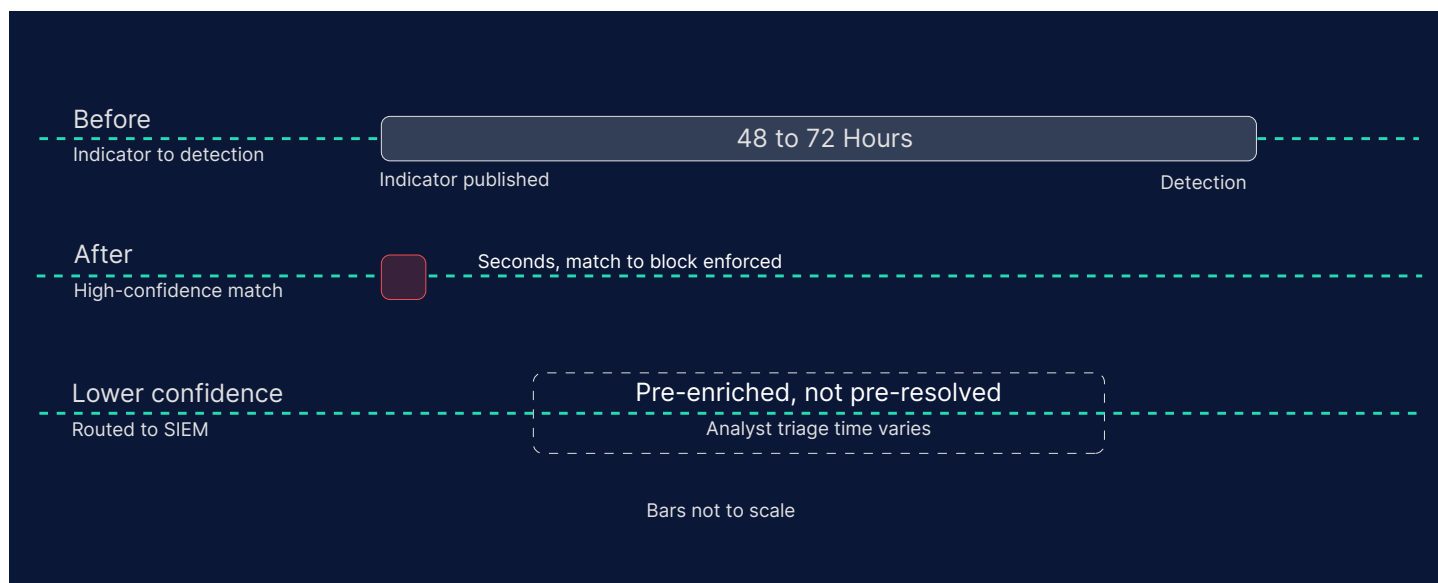
The third stage matches telemetry against threat intelligence at the moment it's ingested, rather than after an alert has already been generated. The intelligence engine draws on a large indicator repository (government feeds, commercial sources, open-source intelligence, and private sharing communities), real-time threat-actor tracking, and MITRE ATT&CK correlation, meaning each event is mapped to a specific tactic and technique, for

example T1078 (Valid Accounts) rather than left as an unlabeled anomaly. A normalized confidence score gets attached to every match, and that score is what determines what happens next. The two resulting paths are meaningfully different from each other.

High-confidence matches. A known-malicious IP address confirmed as command-and-control infrastructure, or a ransomware hash, triggers an automatic block instruction, pushed through a dedicated integration component out to firewall and EDR enforcement points in real time, with no analyst step required. This is the path where the historical 48 to 72 hour gap between indicator

discovery and operationalized detection collapses to seconds, because enrichment, confidence scoring, and the block action all happen at the same point in the pipeline.

Lower-confidence matches. Events that still need human judgment route to the SIEM instead, pre-enriched with MITRE ATT&CK mapping, actor attribution, and a confidence score already attached. An analyst working that alert starts from a contextualized event rather than a raw indicator match. This path doesn't collapse to seconds; it shortens the time spent building context before an investigation starts, without taking the analyst out of the loop.



STORAGE AND RETENTION

Long-term storage sits alongside these three stages rather than inside them. Cleansed, normalized, and deduplicated data lands in a data lake built for retroactive search at petabyte scale in seconds, rather than the much longer search times a legacy SIEM would typically need across the same volume.

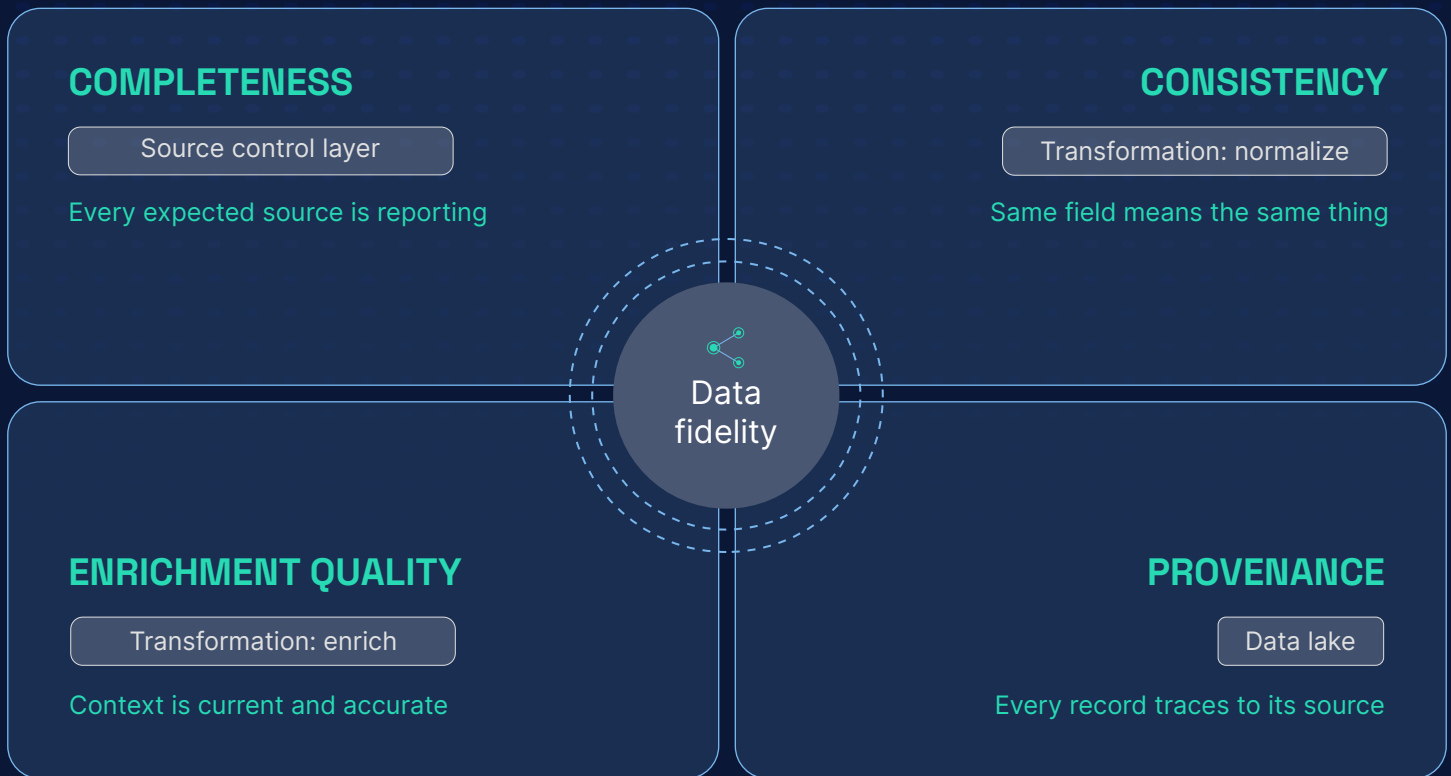
Note: The retention split referenced in the underlying reference material, a 30-day hot tier inside the SIEM with everything else retained in a separate data lake for 12 or more months, is presented there as a target architecture and cost model, explicitly labeled as a target. It isn't a description of any specific organization's current SIEM configuration, and that current baseline should be established independently before treating the 30-day figure as a design input.



Illustrative shape only, not customer data

DEFINING FIDELITY

Fidelity, throughout this paper, means four measurable properties of the data at the point it leaves this pipeline: completeness (signal isn't buried under duplicated or unsuppressed noise), consistency (every source maps to one schema, so a rule or a query means the same thing no matter where it originated), enrichment quality (confidence scores, actor attribution, and technique mapping are attached at ingest), and provenance (that enrichment traces back to the source record and the intelligence match that produced it). Each stage above maps to one or more of these properties: normalization delivers consistency, deduplication delivers completeness, and intelligence fusion delivers enrichment quality and provenance. This definition holds for the rest of the paper; it isn't redefined later.



APPLYING THIS TO YOUR ENVIRONMENT

Two starting points are useful no matter where a given environment currently sits.

STARTING POINT	WHAT TO DO	WHY IT MATTERS
Source and schema inventory	Map current telemetry sources against the four stages above (source control, transformation, storage and enrichment, routing); evaluate OCSF mapping coverage as a discrete, schedulable piece of work	Surfaces exactly where normalization gaps and duplication currently sit, rather than guessing
Measurement baseline	Before changing anything, record current alerts per analyst per day and the percentage requiring genuine investigation, using a consistent methodology	Gives any later architectural change a real before-and-after instead of an assumed one

Table 5. Two practical starting points for evaluating this architecture against a specific environment.

The highest-volume duplicate or noise source in a given environment, commonly firewall, EDR, or authentication logs, is usually the most useful first candidate for isolating the impact of a deduplication layer specifically, since its volume makes the comparison easiest to measure cleanly.

EXPECTED OUTCOMES

Applied consistently, this architecture produces outcomes that can be stated in the same measurable terms used to define fidelity above, rather than as general claims of improvement.

- Detection engineering effort per correlated event type drops from five or more schema-specific rules to one, with correlation across sources handled natively rather than manually.
- Analyst-facing alert volume is targeted specifically by the deduplication and noise-suppression layer, aimed at the gap between the 174-alert-per-day industry baseline and the 22 percent that's genuinely investigation-worthy. Treat this as the direction the underlying data points to, not an independently verified percentage reduction, since the 78 percent figure was never broken out by cause in the source research.
- Time to action for high-confidence intelligence matches drops from a historical 48 to 72 hours to seconds. This applies specifically to the automatic-block path; lower-confidence matches still require analyst review, arriving pre-enriched rather than pre-resolved.
- Retention economics shift from unbounded SIEM storage growth to a fixed, predictable cost: a defined hot-tier retention window inside the SIEM paired with 12 or more months of searchable, cleansed data in a separate data lake. As noted above, the specific figures here describe a target architecture, not a given organization's current retention.

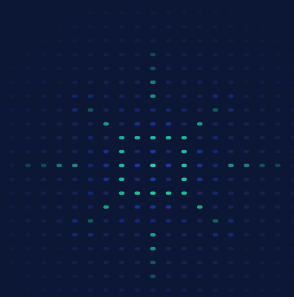
A data foundation with measurable completeness, consistency, enrichment quality, and provenance is also, by definition, the foundation any automation or agentic tooling introduced later would need. Worth stating as a technical dependency rather than a selling point: the data-layer work described in this paper is a prerequisite for that kind of initiative succeeding, independent of whether or when an organization adopts one.

SOURCES AND METHODOLOGY

The statistics cited in this paper on alert volume, analyst workload, and SOC operating conditions come from third-party industry research: UnderDefense, Tines' Voice of the SOC, SANS-affiliated research, Vectra AI, Cybersecurity Insiders, Osterman Research, Gartner's Hype Cycle for Security Operations, the World Economic Forum's Global Cybersecurity Outlook, Unit 42, Gravitee's State of AI Agent Security, and the Strata/CSA AI Agent Identity Crisis report, as cited in the underlying reference material.

NEXT STEP

Everything in this paper describes how the architecture behaves in general. Whether it behaves that way against a specific source stack, a specific schema mix, and a specific alert baseline is a narrower question, and it's answered fastest by watching it run against real telemetry.



**[See a demo](#) of the architecture described
in this paper running against representative
data from your own environment.**