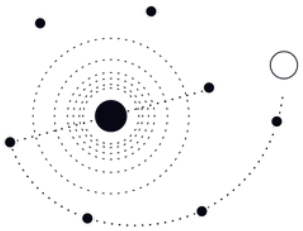
# Zen and the Art of Partial SIEM Migration

Better Coverage.
Lower Costs.
Modern Architecture.

ANVILOGIC™



# The Path from Dark Data to Enlightenment

The term "SIEM migration" usually conjures visions of large-scale upheaval: massive data transfers, multi-year platform shifts, and high-stakes bets on a new ecosystem. But that path isn't right—or necessary—for every team.

Partial SIEM migration isn't a short cut or just a tactical workaround. It's a pragmatic path to sustainable, high-performance detection engineering. By offloading high-volume logs to more cost-effective compute (like Snowflake or Databricks), aligning retention with access needs, and deploying detections closer to the data, teams can achieve measurable savings without sacrificing visibility or capability.

This approach recognizes a basic truth: most modern security teams are already hybrid. They rely on more than one detection surface, more than one query engine, more than one store of truth. Partial migration turns that reality into a strategy.

And it’s not just theory. Forward-looking security organizations are already adopting this model to:

- Move high-volume or low-value logs out of expensive ingest pipelines
- Run detections in data lakes without duplicating effort
- Retain flexibility in tooling without starting over

For their efforts, they’ve unlocked millions of dollars saved in SIEM licensing and storage costs, with better fidelity and more responsive detection coverage. These teams don’t look at this hybrid architecture as a compromise, but as a true step forward.

This journey begins not with a tool, but with awareness.

*Not everything needs fixing. But with clarity, the path forward reveals itself—quietly, and without disruption.*

# Begin with Awareness

*Not everything that can be indexed should be indexed.*

Data growth is inevitable. Industry averages estimate that security-relevant data grows at approximately 30% year over year. That's context, telemetry, events, CMDB data (asset and config info), alerts, and signals, in addition to log volume. Every new application, endpoint, or cloud workload contributes. Security teams are dealing not just with more data, but with more types of data.

For some faced with this reality, the natural reaction is to try and ingest everything "just in case", to treat the SIEM like a vacuum, pulling in as much data as possible for fear of missing the one vital signal. But this instinct, while understandable, is not just counterproductive, but untenable. You need all of your security-relevant data, but most cannot truly afford to store and index all of it. And even if they could, the sheer volume would overwhelm their ability to write detection rules that are both thorough and precise.

## Scaling SIEMs and the Mirage of Full Migration

SIEMs were originally built to centralize, but centralization has a cost—and SIEM costs scale linearly with ingestion, whereas value does not. Teams quickly reach a point where the marginal cost of new data far exceeds its marginal value. The result is a painful paradox: dark data. This is data you know exists, that you know could be useful for detection, hunting, or investigation, but that you cannot afford to bring into the SIEM. It's there, but it's inaccessible. And so for many uses, it may as well not exist at all.

Meanwhile, the specter of a full SIEM migration looms. Migrations are disruptive, expensive, and often regress existing capabilities. They require retraining, reimplementation, and significant time and cost. Too often, the savings they promise fail to materialize and teams are left disillusioned, burdened by new workflows that offer less functionality, flexibility, or reliability while demanding the same or more budget, effort, and compromise.

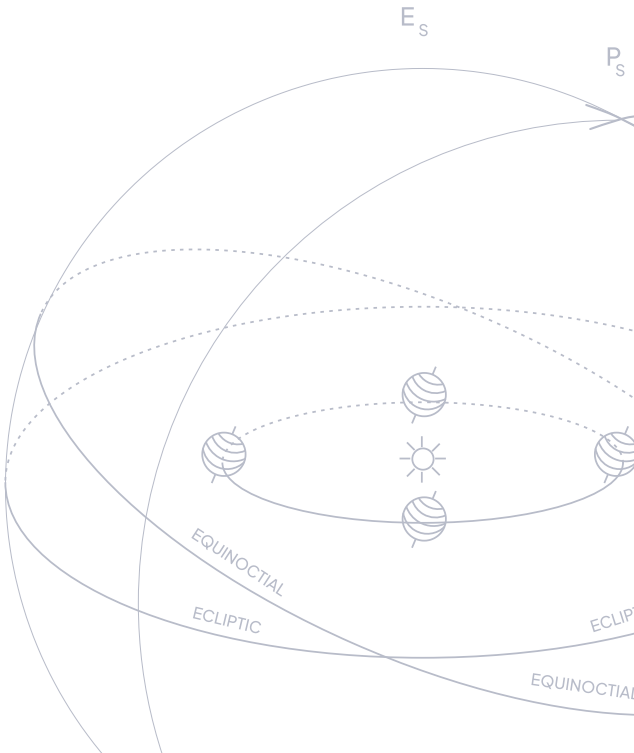
Another path starts smaller. Not with movement, but with observation.

## Evolve Intentionally

Start by seeing clearly.

1. Which detections consistently deliver value?
2. Which ones generate noise?
3. Where does your data actually live?
4. Where does it need to live?

Clarity is your starting point—not configuration.



# Honor the Foundation

*"Garbage in, garbage out" is still the law of the land."*

In an era obsessed with AI, it's tempting to believe that intelligence alone can solve our biggest problems. But no matter how sophisticated your algorithms are, they can't overcome bad inputs. AI cannot fix foundation—only amplify it.

Security teams are being asked to do more with less: more detections, faster triage, better insights. And increasingly, they're being asked to infuse AI into every part of that process. But here's the truth: without clean, structured, meaningful data, AI simply accelerates confusion.

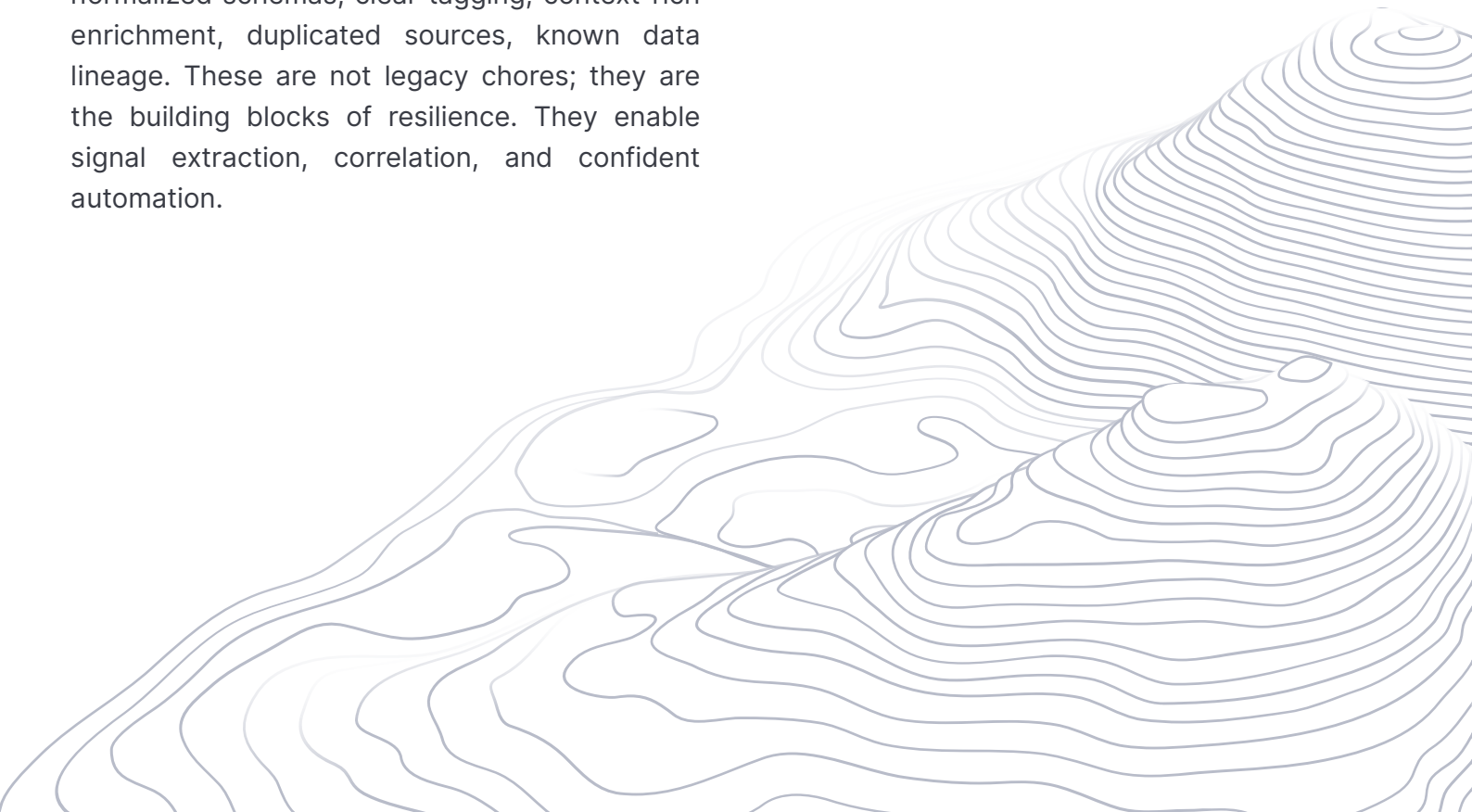
Think of your data architecture like a garden. AI is not the gardener, it's the fertilizer. If what's planted is tangled, inconsistent, or choked with weeds, AI won't produce clarity. It'll produce chaos, just faster and at greater scale.

This is why strong fundamentals matter: normalized schemas, clear tagging, context-rich enrichment, duplicated sources, known data lineage. These are not legacy chores; they are the building blocks of resilience. They enable signal extraction, correlation, and confident automation.

*And they enable trust*. Analysts trust what they understand. Engineers trust what they can trace. Leadership trusts what delivers consistent results.

The move toward partial SIEM migration doesn't start with technology but instead with discipline. If your logs are unclear, your pipeline opaque, or your enrichment brittle, then no platform swap will save you. You will carry confusion with you.

The foundation is not glamorous, but it is transformative, and it's what makes everything else—the AI, the analytics, the automation—actually work.



# Karma from the Cloud

*Wisdom has been chasing us; now we are slowing down to let it catch up."*

Every security team now lives in the cloud—whether by strategy or surrender. But how we got here matters, and in that story lie lessons worth remembering.

The early days of cloud migration were marked by urgency. Teams were told to move fast, modernize, and "lift and shift" their infrastructure with minimal disruption. Applications, data, and workflows were uprooted and replanted in cloud environments—often without rethinking architecture, cost, or control.

The results were predictable: bloated bills, fragmented visibility, and brittle systems that weren't built for scale or agility. We learned, sometimes painfully, that moving to the cloud is not the same as adapting to it.

Security teams today face an eerily similar moment with their data pipelines. The pressure is on to modernize, to divest from legacy SIEMs, and to embrace new architectures. But the instinct to "lift and shift" detection logic and ingest habits—without reimagining their purpose or place—is a trap.

Here's what we've learned from infrastructure teams that have already walked this path:

01 **Cost is architectural, not just contractual.**

You don't control spend with vendor negotiations; you control it with design decisions.

02 **Replatforming without redesign just replicates the past.**

Don't move inefficient workflows to a new system—optimize them.

03 **Modularity wins**

Teams that embraced microservices, containers, and decoupled systems gained agility. The same applies to unbundling the SIEM.

04 **Visibility must be intentional.**

Cloud observability wasn't a given—it had to be designed. Security visibility is no different.

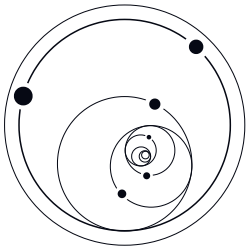
05 **It's not either/or.**

You don't need to move everything to the cloud, or rip out your SIEM overnight. Hybrid architectures work. Transitional states are valid. Have both and use each for what it's good for.

We are not starting from scratch. The cloud's growing pains have already played out. The wise move now is not to repeat them, but to learn from them.

Modernizing your security data stack is just as much a karmic effort as it is a technical one. The design choices you make today will echo into cost, coverage, and capability tomorrow.

So now we can move with intention because the cloud already taught us how.



# Unbundling the Stack

*The whole works best when its parts can move.*

Traditional SIEMs were designed during a very different era of security. They assumed that log volumes were modest, infrastructure was mostly on-prem, and detection logic lived inside a single, centralized system. Back then, bundling everything into one platform made sense.

Today, that model strains under modern demands. Security teams face explosive data growth, hybrid architectures, and the need for fast, iterative detection development. But legacy SIEMs still hinge on one model: ingest, index, and pay. As a result, teams are forced to route all data through the same funnel—whether or not it belongs there.

While vendor lock-in remains a challenge, the deeper problem is architectural: these systems were not built for petabyte-scale telemetry, cloud-native logs, or distributed analytics.

While vendor lock-in remains a challenge, the deeper problem is architectural: these systems were not built for petabyte-scale telemetry, cloud-native logs, or distributed analytics.

Unbundling is the answer, and for many SOCs it’s already underway.

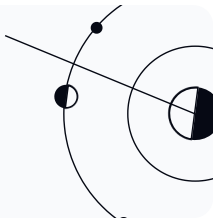
## The Modern Alternative

Rather than funneling everything through a single vendor's constraints, modern security teams are embracing modularity. This isn't just a preference, but a necessity in a world where speed, scale, and flexibility are non-negotiable.

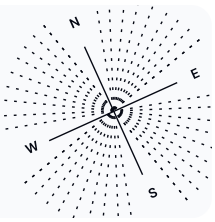
Modular architectures separate the stack into discrete, composable layers.

| | Tools | Tooltip |
|---|---|---|
| Ingest & Stream | Cribl, Fluentbit, Apache NiFi | “Route and filter raw data from any source” |
| Store & Compute | Snowflake, Databricks, Azure, S3 | “Efficient, scalable storage and compute engines” |
| Normalize & Detect | Anvilogic, custom pipelines | “Enrich, parse, and apply detection logic” |
| Search, Hunt & Triage | Anvilogic, open federation, cloud-native search | “Query and investigate across distributed data” |
| Case & Response | Any SOAR, any case management | “Automated response and investigation workflows” |

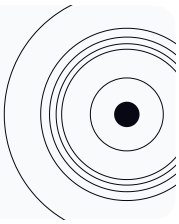
> “Each layer can be replaced, scaled, or skipped independently.”



> “Unbundled = no vendor lock-in.”




> “Pay for only what you use, at the layer you need it.”



> “Accelerate innovation without waiting on monolithic SIEM releases.”

Each tool does what it does best. And critically, each can be evolved independently. No more waiting for a bundled SIEM to support a feature already native to the broader ecosystem.

This modular model enables not just flexibility—but efficiency. You pay for each layer based on what it needs to do, not what a bundled platform dictates. You gain the freedom to replace, scale, or skip layers depending on use case.

This shift mirrors what happened with infrastructure in the cloud era: tightly coupled platforms gave way to services that could evolve independently. Why shouldn’t your security stack work the same way?

## Why It Works

### Optionality

Swap out a single layer without redoing the stack

### Cost Control

Route logs and compute where it’s most efficient

### Innovation

Adopt new tools without waiting for vendor roadmaps

### Resilience

Failure in one layer doesn’t bring down the whole system

The modern SIEM isn’t a tool. It’s an approach. One that favors interoperability over centralization, lets you evolve without disruption, and that is built for the real-world complexity of today’s environments.

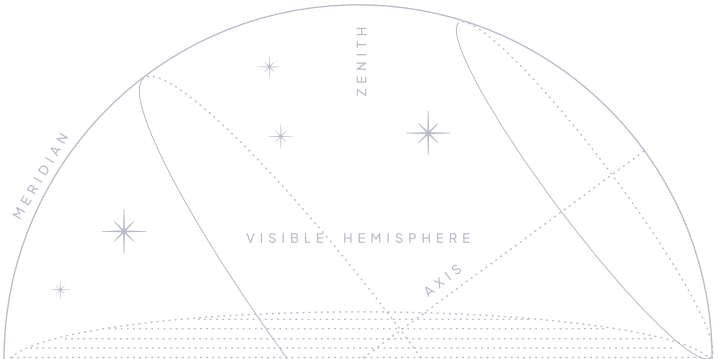
# Data Ingestion & Cost Primer

Understanding how data tiers map directly to cost empowers teams to make informed onboarding decisions. In modern cloud data platforms—Snowflake, Azure Data Explorer, Databricks—each transformation impacts both storage and compute charges. For example:

| Raw logs / Machine data | Parsed, normalized, joined | Security insights & BI dashboards |
|---|---|---|
| **Bronze Tier** | **Silver Tier** | **Bronze Tier** |
| Storage : ~$23/TB/month | Storage : ~$23/TB/month | Storage : ~$23/TB/month |
| Compute : Minimal (ingest only) | Compute : ~$8/hr (medium warehouse) | Compute : ~$16/hr (high-perf warehouse) |
| 100 GB/day ≈ ~$2.30/month in storage | 2 hr/day = ~$480/month in compute | 8 hr/day = ~$3,840/month in compute |

"Legacy SIEM: $0.15–$0.30/GB ingested = $4,500–$9,000/month for same 100 GB/day."

While exact rates differ slightly by vendor, these figures are broadly representative across Snowflake, Databricks, and Azure Data Explorer — all of which employ similar TB-per-month storage pricing and credit-based compute billing models.



# Flow Where the Data Goes

*Start where you are. Every step builds the next.*

Modernizing your security data stack doesn't have to be a giant leap, it can be a steady current. Like water, your architecture should flow around constraints, not crash into them. That's the philosophy behind incremental divestment.

This isn't a drastic break from tradition. It's a chance to modernize without disruption. You don't need to rip out your SIEM and you don't need to ingest everything into a lake. You just need to move intentionally—one phase at a time.

01

## Phase 1: Awareness

Map your current data estate:

- What log sources are feeding your SIEM?
- What's being dropped or never ingested?
- What's hot vs. cold? Real-time vs. archival?

This is your opportunity to take inventory—not just of data, but of pain points and missed opportunities.

02

## Phase 2: Shadow Search

Before migrating data, validate its value:

- Run detections outside your SIEM (e.g., Snowflake, Databricks)
- Compare performance, coverage, and false positive rates
- Prove that insights don't have to come from indexed data

This builds confidence without risk.

03

## Phase 3: Selective Offloading

Shift high-volume log-sets:

- Move them from expensive SIEM ingest to cheap, queryable storage
- Retain access through federated search or parallel triage

The goal: realize large cost savings without losing signal.

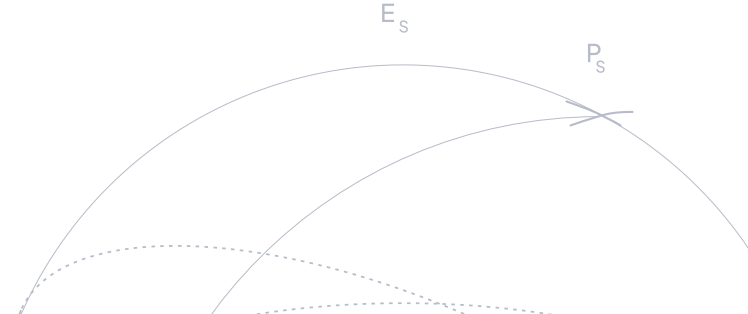
04

## Phase 4: Retention Recalibration

Not all logs need equal time-to-live:

- Align retention policies with access patterns
- Keep hot what you query; cold what you rarely touch
- Meet regulatory requirements without overpaying

This reduces storage costs and improves search performance.



01

## Phase 5: SIEM Slimming

Now you're ready to narrow the focus:

- Ingest only what your SIEM does best: alerting, correlation, high-value detections
- Redirect everything else to cheaper, more flexible platforms

At this point, you're reducing clutter, tightening your focus on what delivers the most value.

Each phase supports the next. No leaps. No rewrites. Just momentum.

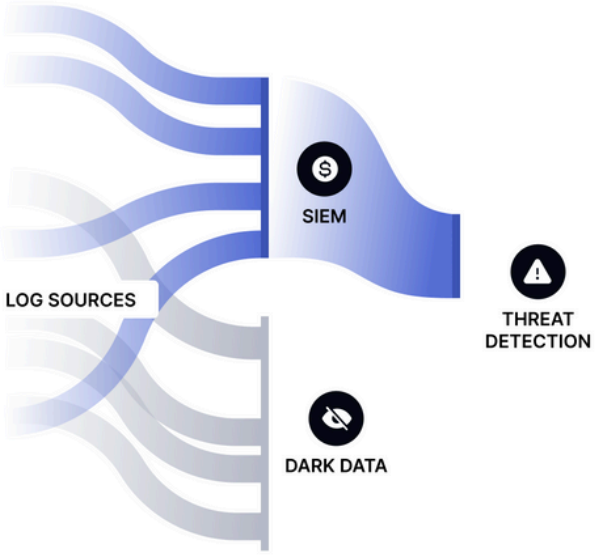
Divestment doesn't mean leaving something behind. It means designing your system around what works best today. Flow where the data goes and bring your architecture with you.

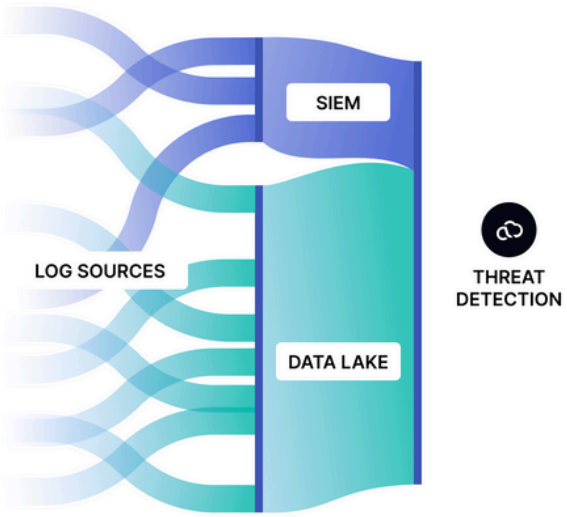
### SIEM Only Strategy

- ⊗ Data silos
- ⊗ Missed signals
- ⊗ $$$ cost for ingestion

### Partial SIEM Strategy

- ✓ All logs available
- ✓ Detection coverage improves
- ✓ Lower cost per GB



# The Wisdom of Where

*Not all logs are created equal.*

Security data isn't binary, it exists along many axes. Some logs are critical, others contextual. Some are urgent, others archival. The instinct to treat all data the same to ingest it all, retain it equally, and make it all instantly searchable is a fast path to overspending and underutilizing.

Instead, think dimensionally. Logs vary in:

01 **Purpose** – Detection? Investigation? Compliance?

02 **Timing** – Real-time? Post-incident?

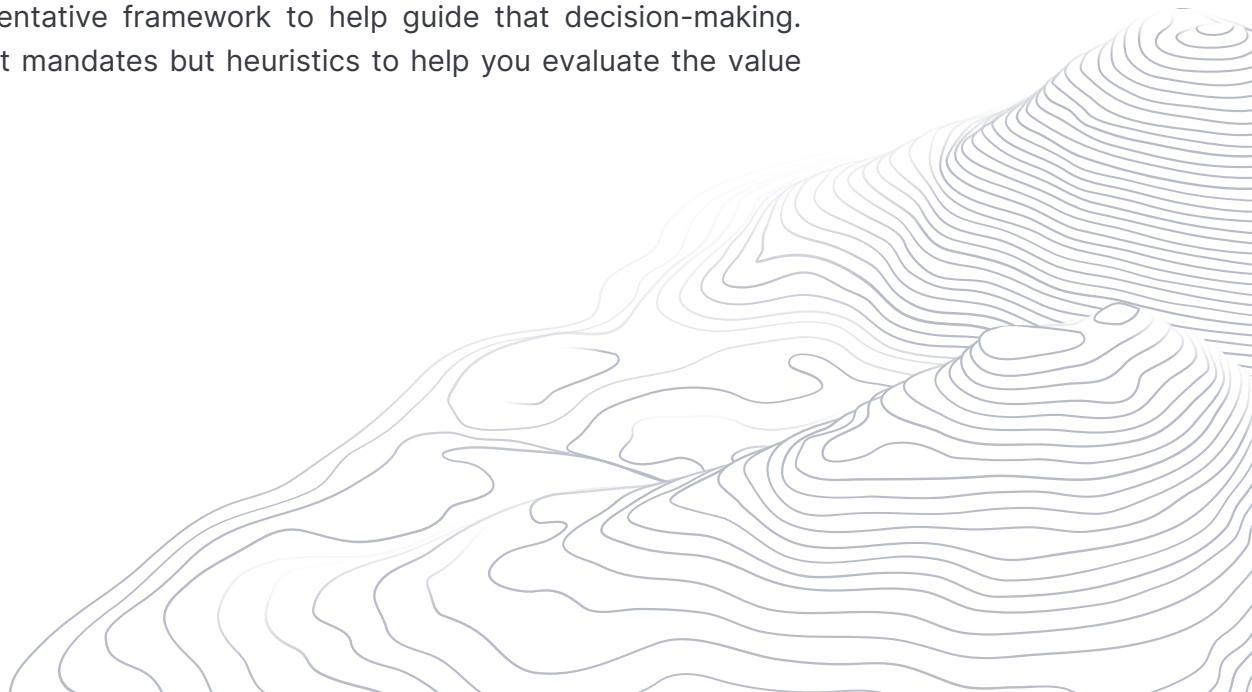
03 **Access** – Frequent? Rare?

04 **Sensitivity** – Confidential? Public?

05 **Volume** – High-noise? Signal-dense?

Your architecture should reflect those differences. Assign logs to storage and processing tiers based on what they do, not just where they come from.

Below is a representative framework to help guide that decision-making. These tiers are not mandates but heuristics to help you evaluate the value of data in context:



| Priority | Data source type | Log Examples | Assets to Protect |
|---|---|---|---|
| Higher priority | Endpoint, Network, Cloud, Security tools | Firewall syslog, Windows endpoint logs, Threat intel | IP assets, critical apps, customer data, financials |
| | Authentication & Management tools | AWS CloudTrail, Azure AD, Okta logs | User accounts, API keys, config files, admin access |
| | Infrastructure & Application tools | Syslog, Database logs, IT tool telemetry | Servers, databases, dev environments, internal systems |
| Lower priority | Monitoring & Visualization tools | Performance dashboards, usage metrics | Observability, operational insight |

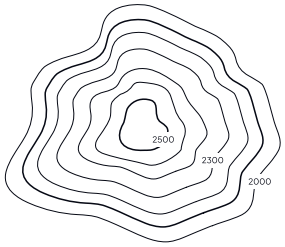
These tiers aren't rigid they're reflective. What's priority 2 in one context may be priority 0 in another. But using a model like this helps teams:

- Align log collection with actual security outcomes
- Prioritize ingestion based on value, not fear
- Clarify what logs benefit from real-time access or correlation in the SIEM and which can be staged, searched, or triaged through other means
- Start designing a system that flexes with change

Importantly, criticality doesn't always correlate with ingest volume. High-priority sources like EDR or CloudTrail often benefit from cheaper storage and targeted detection.

Treat this matrix not as a verdict on where logs should go, but as a conversation starter between security goals and system design.

In a well-balanced system, every log has its plac and not every place is the SIEM. When logs are placed with intention, your detection surface improves, your costs go down, and your team gains confidence in what's covered and what isn't.




# The Hidden Cost of Ingest-Only Thinking

The instinct to ingest everything into the SIEM "just in case" is deeply ingrained. But it comes with quiet, compounding costs:

### Dollars

You pay to ingest, store, and search; often for data you'll never query.

### Latency

Centralized indexing slows detection and investigation when data volumes spike.

### Confidence

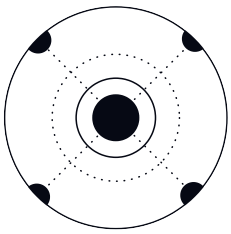
Analysts don't trust a system they know is incomplete; or too slow to use.

## Much of this data is :

- **Cold:** rarely if ever queried
- **Duplicative:** already available elsewhere
- **Unenriched:** hard to act on without context

## The result :

You spend more, see less, and still feel behind. Move past the reflex. Ingest what matters and architect for access, not accumulation.



# Flow Over Friction

*In migration, the rarest resource isn't storage—it's attention.*

Many teams already know which logs matter; what they lack is human bandwidth to move them. High-value data remains in SIEMs not from indecision, but from queue length. Inertia—not ignorance—is the obstacle.

The question shifts from "Which logs deserve SIEM storage?" to "Where will limited engineering time create the next, safest win?" Small, intentional moves beat heroic cutovers.

## Begin with Constraints

| Constraint | How it shows up | Hiddene cost |
|---|---|---|
| **Staff cycles** | Detection engineers can't pause triage to rewrite parsers | Aging alerts, growing backlog |
| **Rule entanglement** | Macros and dashboards assume on-box data | Refactors snowball |
| **Change risk** | Auditors bristle if critical logs disappear overnight | Dual environments double license cost |
| **Tool sprawl** | New pipelines (e.g., Kinesis → Snowflake) need care and feeding | More run books, more on-call pages |

Seeing the friction clearly lets you plan a backlog driven by effort, not theory.

## Clean Up Rules First

Before moving feeds, start by cleaning up and rationalizing the applicable feed rules in in your SIEM. Apply the classic “garbage in, garbage out” principle: validate what detections currently depend on the feed, remove stale or redundant rules, and document active coverage. This ensures that when feeds are migrated, you understand their present role and avoid transferring unnecessary noise.

## Sequencing Over Sorting

Logs move in the order of energy required, not importance on paper. Start with the lanes that yield impact with minimal disruption. When in doubt, chase volume; GB/day drives cost, so moving the highest-volume sources provides the fastest relief.

| Fast Win | Opportunistic | Legacy / Low Noise |
|---|---|---|
| **Action:**<br>Mirror to the lake now (keep SIEM copy for short overlap) | **Action:**<br>Detect in place, query via join | **Action:**<br>Leave in SIEM until retirement |
| **Candidate Logs**<br>CloudTrail, VPC Flow, ELB access | **Candidate Logs**<br>AD/Okta auth, EDR telemetry | **Candidate Logs**<br>On-prem app logs, print-server syslog |
| **Why this first**<br>High-volume JSON sources map cleanly → e.g. Snowflake; immediate storage savings (GB/day) and big cost delta | **Why this first**<br>Medium volume overall; prioritize high-GB/day auth streams first; rules migrate with small macro tweaks | **Why this first**<br>Low volume; moving saves pennies, costs hours |
| **Exit Criteria**<br>**14–30 days** of clean dual-write checks (**7–21** for top-volume sources; tune per **Set the Overlap Window** below) | **Exit Criteria**<br>Touch the rule for any other change | **Exit Criteria**<br>Application decommissioned |

Sometimes the wisest move is to *wait*—until the effort-to-benefit ratio improves.

Rule of thumb: if a source sits in your top three by GB/day, mirror or move it early; if it's bottom-quartile volume, defer unless it's detection-critical.

**Set the Overlap Window (Tune to Taste)**

**Starting point (cost-aware):** 14–30 days of dual-write/dual-read for new pipelines and fast-win sources. For your top-volume sources, cap overlap at 7–21 days unless audit requires more. Shorten or extend based on risk, comfort, and budget.

**Shorten to ~7–21 days when:**

- Parsers are mature and schema-stable (JSON, strongly typed events)
- Detections are portable with minimal macro changes
- Data is non-regulated and audit needs are light
- You've validated parity on a representative sample (e.g., last 14 days)

**Extend to ~30–60+ days when:**

- Regulated workloads or strict audit report cycles apply
- Parser refactors or vendor upgrades are in flight
- Seasonality affects detections (e.g., quarterly access reviews)
- Incident tempo is high and you want more side-by-side confidence

**Exit checks (pick 2–3):**

- Parser/ingest error rate ≈ 0% for the overlap period
- Alert volume/fidelity within ±10–15% between stores (expected deltas explained)
- No MTTR regression for alerts tied to the moved source
- Stakeholders sign-off (SecOps + Compliance, if applicable)

**Budget guardrail:** At enterprise ingest rates, 30 days of dual-write on top-volume sources can add up quickly.

**Estimate before you commit:**

I*ncremental overlap cost ≈ (GB/day) × (destination $/GB) × (overlap days) + pipeline egress*

If the estimate exceeds your weekly budget, reduce overlap days or scope (see knobs below).

**Cost-cutting knobs (use singly or in combination):**

- **Scope the mirror:** Mirror only a sample (10–25%) or specific event classes needed for parity checks.
- **Aggregate during overlap:** Send summaries/aggregates to the SIEM while storing full fidelity in the lake.
- **Backfill instead of always-on:** Run nightly batch backfills to the lake for the overlap window rather than real-time streaming.
- **Lifecycle & compression:** Apply shorter retention and compression on the destination during overlap (e.g., 7–14 days hot, then auto-expire).
- **Turn off duplicate alerting:** Keep alerts firing from one system during overlap; use the other for validation to avoid operational noise.

## Validation Phase: Confirming Coverage

Once feeds are migrated, use the cleaned-up baseline to validate that you haven't lost detection coverage. Compare scenarios, threat intelligence triggers, and rule outcomes between your SIEM and your data lake. The goal is not just to maintain fidelity but to increase coverage by leveraging new joins, expanded data availability, and refined rule sets.

## Unify the View, Relax the Timeline

An analytics layer that queries both your data lake **and** your SIEM removes finish-line pressure. Analysts can:

- write a single rule that targets both stores,
- compare alert fidelity during overlap windows,
- phase out the SIEM copy only when confidence is high.

Prioritization becomes a function of people capacity, not a months-long design exercise.

Log value still matters, but migration sequencing should track staff availability and risk tolerance. Begin with feed cleanup, then validate that coverage strengthens after migration. Use the tiers above to chip away at high-volume offenders while keeping critical detections live. Over time, the center of gravity shifts naturally to the lake, without a heroic, all-at-once effort.

# What You Gain by Letting Go

*Clarity isn't what you add. It's what you stop holding onto.*

Partial migration isn't just about solving problems, it's about unlocking value. Every step you take toward intentional architecture gives something back.

## Lower Cost, Higher Clarity

When you offload noisy or redundant data, you don't just cut spend, you sharpen your focus. Analysts waste less time on false positives. Engineers spend less time firefighting ingestion failures. Budgets stretch further.

## Better Detection, Faster Response

Running detections closer to the data unlocks new use cases. Correlation improves when you're not restricted to what's been ingested. Triage gets faster when alerts are enriched with context that was previously out of reach.

## Greater Agility

With an unbundled stack, you can change one part of your pipeline without breaking the rest. Want to trial a new enrichment layer or lake backend? You don't need a migration plan, you need a config update.

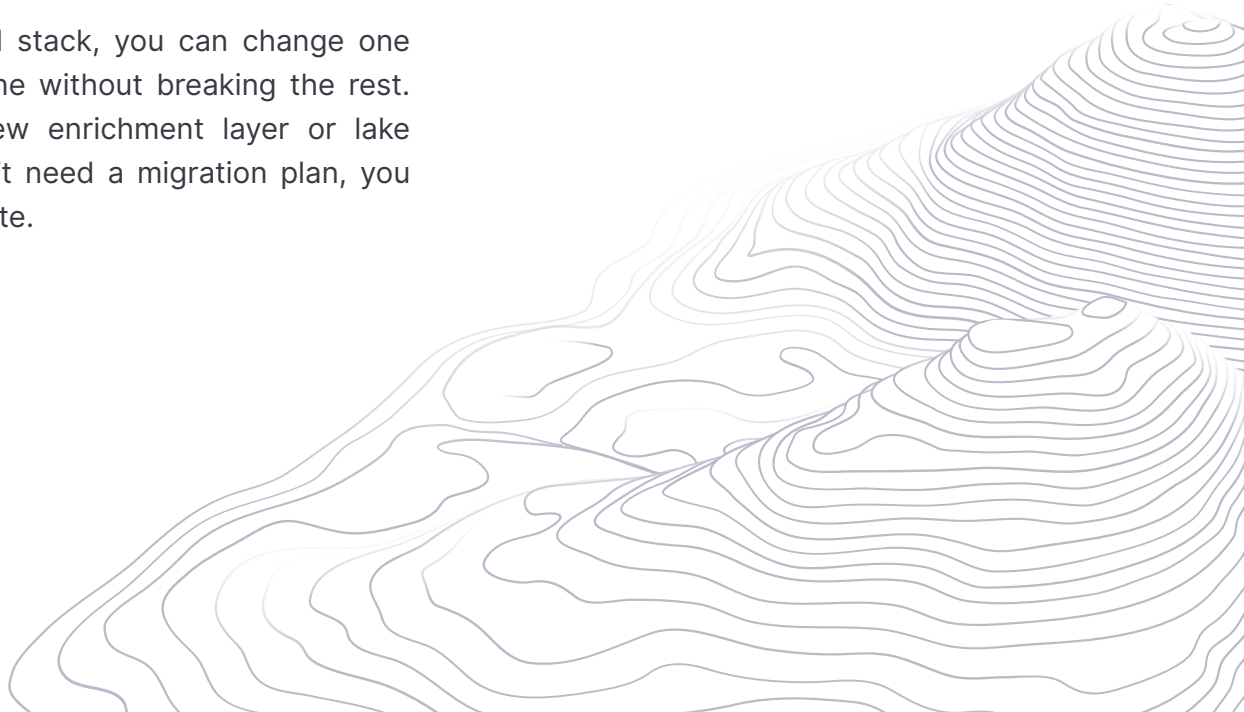
## Optionality at Every Layer

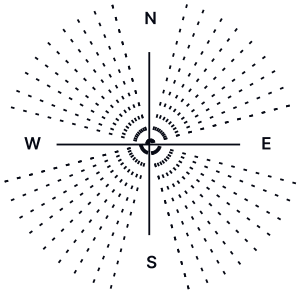
Your architecture reflects your priorities. Modern, modular design means you can:

- Store more, search smarter
- Detect where it's cheapest
- Alert where it's fastest
- Retain what matters most

You're not locked into a single vendor's roadmap. You're not betting your future on a single platform.

You keep what works. You upgrade what doesn't. And you do it all at your pace.






# Closing Reflection

*You don’t need to rip and replace. You need a platform that respects your path.*

You don’t have to start over to make progress. The path to a more modern, more sustainable architecture doesn’t begin with destruction, it begins with clarity.

Modernization isn’t a platform choice. It’s a design discipline. It means asking: What data do we truly need? What signals matter most? Where should they live, and how should we access them?

This guide hasn’t advocated for a single destination, but instead offered a method; a quieter, more observant way to evolve. Less disruption, more continuity. Less noise, more intention.

You may still use your SIEM. You may still rely on core workflows. That’s not a failure, it’s maturity. The goal isn’t to abandon what’s familiar, but to build around it, improve upon it, and let your architecture reflect the way your team already works.

There’s no award for ripping and replacing. But there is real progress in tuning your system to serve you better.

Modern security isn’t loud. It doesn’t announce itself. It shows up in workflows that just work, costs that make sense, and data that’s where you need it, when you need it.

Look ahead. The horizon is clear.

# Epilogue: A Path You Don’t Have to Walk Alone

This guide is more of a reflection than a prescription. A way to think more clearly about your architecture, your telemetry, and your time.

Maybe you’re still deep in a traditional SIEM. Maybe nothing has moved yet. That’s okay.

You don’t have to overhaul everything to begin. You don’t even have to offload a single log. What matters is clarity and from there, movement becomes easier.

There are platforms designed to meet you exactly where you are. To make detections stronger, more connected, and easier to manage long before anything is migrated. To help you explore adjacent data sources without losing fidelity. And to support you with workflows that grow more autonomous and more aligned with every step.

When you’re ready to unify across environments, the architecture is already in place. And when you’re not, nothing breaks; you keep moving forward.

This is the philosophy that guides our platform designed to support your journey, not dictate it.

The journey isn’t measured by how quickly you divest but by how confidently you detect.

## Ready to take the next step?

Let’s explore what clarity looks like for your team.

Book a conversation →