

From Telemetry Inflation to Decisive Observability

Why proof timing—not archive volume—defines the cost floor at hyperscale line rates

- **Author:** Alain Degreffe
- **Patent status:** USPTO Track One pending

Third-party product and trademark notice: SolarWinds, Prometheus, Grafana Mimir, Monarch, Google, and other third-party names are used in this paper solely for nominative, descriptive, and comparative reference purposes. All trademarks and trade names remain the property of their respective owners. No sponsorship, endorsement, certification, affiliation, or approval by any third-party vendor is implied.

Comparative methodology notice: The quantitative comparisons in this paper are illustrative regime-level calculations derived from public documentation, published papers, and explicitly stated modeling assumptions. They are not vendor-certified sizing guidance, product benchmarks, procurement recommendations, or claims of functional equivalence.

1. Introduction

At large scale, the operational problem is rarely missing telemetry; it is delayed explanatory proof.

Most stacks optimize coverage and retention, but incident cost is often driven by how long teams wait before they can trust a causal signal.

ESC addresses this timing gap by treating observability as an evidence-activation discipline: deciding **when** deeper proof must exist, not simply **how much** data should be stored.

This discipline is built on two domain-agnostic invariants—**occurrence** and **observer-time reference**—which shift the dominant scaling variable from raw telemetry volume toward structured occurrence activity and proof-activation timing.

Observability solved collection.
ESC addresses activation timing.

In this framing, ESC shifts observability from permanent fidelity infrastructure to conditional explanatory activation. It operates upstream of logs, metrics, packet capture, and time-series infrastructure without replacing them.

At 400G/800G line rates, permanent explanatory readiness becomes economically non-linear.

The strategic risk is not telemetry absence. It is explanatory delay disguised as observability maturity.

1.1 The Two Invariants Behind ESC

ESC does not treat telemetry as the primary observable. It treats **structured occurrence behavior** as the primary operational signal.

The discipline rests on two invariants.

Occurrence

An occurrence is a bounded indication that something happened.

It does not require payload content, semantic interpretation, protocol decoding, log parsing, trace reconstruction, or measurement of system state.

In this paper, occurrence is the unit that replaces raw telemetry volume as the primary scaling variable.

Observer-Time Reference

Observer-time reference is the temporal frame in which occurrences are structured for comparison.

ESC does not require the observed system to provide a complete, globally synchronized, producer-authoritative timeline. Instead, occurrences are structured at the point of observation into ticks, cycles, or comparable temporal units.

This allows deviation-relevant behavior to be compared without requiring permanent high-fidelity capture or global clock authority as the default operational posture.

Conceptually:

```
Node A: occurrence observed within tick T
Node B: occurrence observed within tick T

ESC/SOBT: compare structured occurrence behavior under observer-time reference
```

The important distinction is not that cost becomes independent of activity. It does not.

The distinction is that ESC shifts the observability problem away from continuous interpretation of raw telemetry volume and toward bounded occurrence activity, observer-time structuring, and governed proof activation.

2. Reference Datacenter Model (Illustrative Inputs)

All comparisons use a shared parametric observability envelope:

- **Observed interfaces:** 500,000
- **Metrics per interface:** 4
- **Prometheus scrape interval:** 15 seconds
- **SolarWinds interface statistics interval:** 9 minutes
- **ESC filters per interface (upper bound):** ≤ 15
- **ESC structuring tick (default planning basis):** 250 ms
- **High availability:** assumed where relevant

The interface is used as the universal scaling unit because it matches:

- SolarWinds monitored elements;
- Prometheus time-series targets;
- Monarch telemetry streams;
- ESC occurrence vectors.

No node/interface ratio assumptions are used.

3. Comparative Decision Matrix (Illustrative)

This section replaces product-by-product narrative with a compact decision layout. Named products and systems are referenced only because their public documentation or publications provide concrete numeric anchors.

The decision question is not which telemetry product stores the most data. It is which observation regime produces trusted explanatory evidence early enough to change operational decisions.

OBSERVATION REGIME DECISION MATRIX					
Regime	Ingestion model	Primary latency driver	Primary CAPEX pressure	Primary OPEX pressure	Strategic fit for deviation-triggered proof
SolarWinds class	Polling supervision	Polling windows	Polling-engine scale-out	Delayed cross-window correlation labor	Medium (strong monitoring; proof timing remains polling-window dependent in this model)
Prometheus/Mimir class	Continuous scrape + TSDB	Scrape/rule/query pipeline	Ingest + in-memory series + replication	Query-time investigation overhead	Medium-high (strong metrics; selective activation is not the primary modeled assumption)
Monarch-class fabric	Distributed telemetry backbone	Regionalized ingest/query path	Large telemetry-fabric infrastructure	Workload-dependent explanation workflows	Reference backend class, not direct deployment comparator
ESC discipline	Deviation-triggered activation	Observer-time structuring tick	Control-plane + selective deepening logic	Trigger governance and calibration operations	High (if bounded deterministic controls are implemented)

ESC's deviation-triggered activation is grounded in the two invariants described in Section 1.1: occurrence provides the bounded operational unit, and observer-time reference provides the comparative frame. This is what allows ESC to avoid treating continuous ingestion or polling as the default source of decision authority.

3.1 Non-Substitutability Controls

This paper does not treat SolarWinds, Prometheus/Mimir, Monarch, and ESC as mutually replaceable products. They are compared only as observation regimes across load shape, timing, and proof-access behavior. The comparison is architectural and methodological, not a claim that the named products compete directly with, endorse, or can be replaced by ESC.

3.2 Infrastructure Footprint Contrast at 500,000 Interfaces

The following table intentionally pushes each observation regime into an infrastructure-footprint reading. It is not a vendor-certified deployment bill of materials, product benchmark, or purchasing guide. It is a raw regime-level stress view showing how each architecture class converts a hypothetical 500,000-interface observability problem into CPU, RAM, storage, and server pressure under the assumptions stated in this document.

The figures below are deliberately **raw baseline estimates**. They do not include HA pairs, spare capacity, multi-region replication, query acceleration layers, or operational safety margins.

Raw normalization assumptions derived from the cited public sources and explicit modeling choices:

- **SolarWinds class:** 10.4-41.7 polling engines derived from public element-per-engine guidance, then normalized using an explicit illustrative per-engine raw envelope of 16-32 CPU cores, 64-128 GB RAM, and 200-500 GB SQL/backend storage.
- **Prometheus/Mimir class:** full raw metrics-engine envelope derived from public Prometheus and Grafana Mimir sizing/storage guidance, including scrape fan-out, ingestion, in-memory series, query path, rule evaluation, compaction, index/cache pressure, and 30-day storage.
- **Monarch-class fabric:** public fleet-class reference only, based on the cited Monarch paper; not normalized to one 500k-interface deployment.
- **ESC discipline:** single modeled operator node baseline, before HA policy; storage retains deviation history and selected evidence metadata, not irrelevant occurrence events.

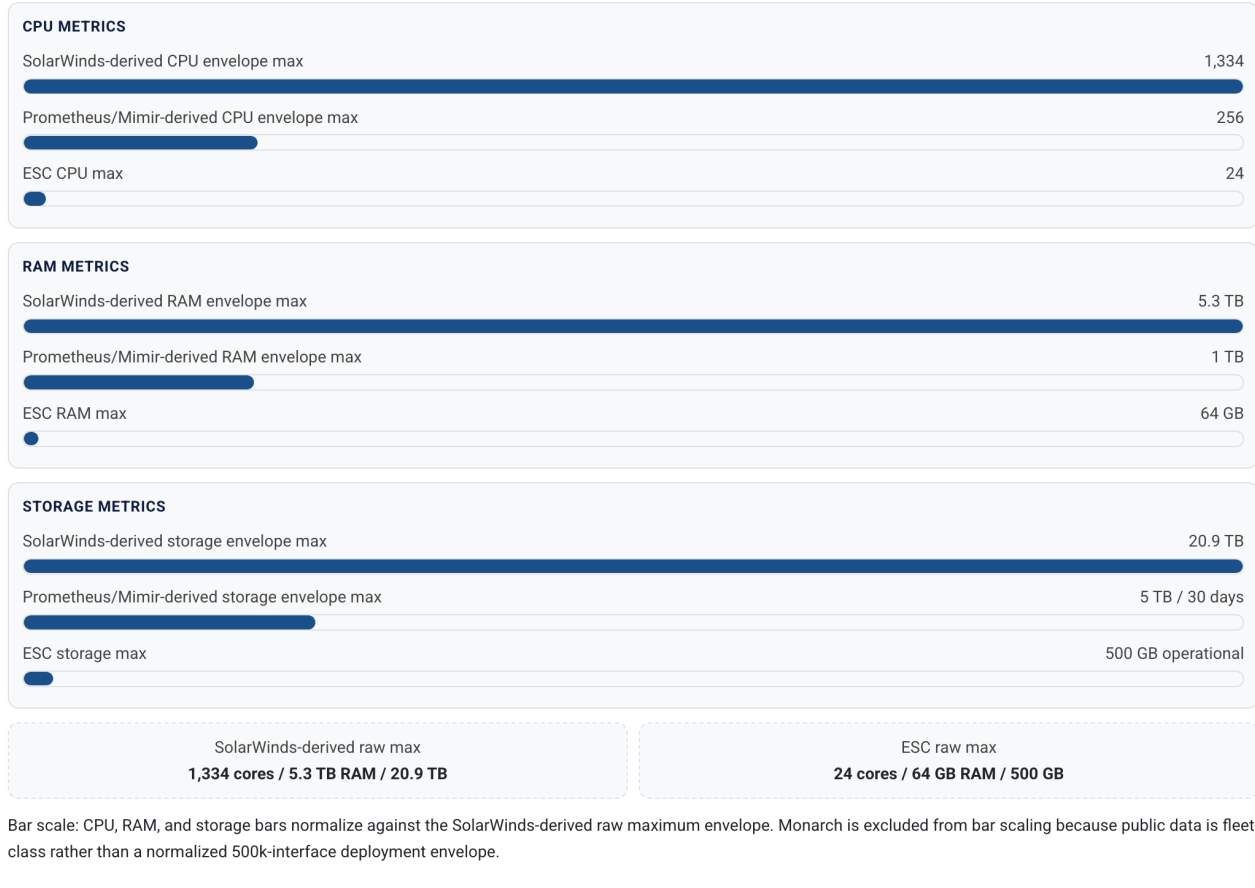
RAW INFRASTRUCTURE FOOTPRINT AT 500,000 INTERFACES					
Observation regime	CPU cores	RAM	Storage	Footprint	Dominant scaling pressure
SolarWinds class	166-1,334	666 GB-5.3 TB	2.1-20.9 TB	10.4-41.7 polling engines	Polling-engine multiplication and SQL/backend growth
Prometheus / Mimir class	64-256	256 GB-1 TB	1-5 TB / 30 days	Multi-role metrics engine	Scrape fan-out, ingest, index, query, rules, compaction
Monarch-class fabric	Not reducible from public data fleet-scale compute fabric	Close to 1 PB compressed memory	1-10 PB distributed storage	Planet-scale telemetry fabric	Fleet-class telemetry operation
ESC discipline	16-24	32-64 GB	50-500 GB	1 general-purpose server	Deviation history and selected evidence metadata

The contrast is architectural, not vendor-specific. Under the assumptions used in this model, continuous collection regimes concentrate cost around polling, ingestion, retention, and query paths. ESC shifts the modeled investment target toward compact control-plane capacity and governed activation timing. This shift is possible because occurrence and observer-time reference reduce dependence on permanent continuous high-fidelity capture as the default posture.

Figure G0 - Raw infrastructure footprint contrast at 500,000 interfaces

This figure compares raw baseline footprint only: no HA, no spare capacity, no multi-region replication, and no query-acceleration layer.

The visual contrast is intentionally architectural: polling engines and always-on ingest versus bounded activation capacity.



Methodological boundary:

- SolarWinds CPU/RAM/storage values are explicit illustrative normalization values derived from the cited public engine-count range and a stated per-engine raw envelope selected by the author for modeling purposes; they are not vendor-certified sizing.
- Prometheus/Mimir values represent a full raw metrics-engine envelope derived from cited public sizing and storage guidance, not only distributor-plus-ingester sizing. The mathematical ingest lower bound is 12 CPU cores and 22 GB RAM, but the table includes scrape fan-out, query path, rule evaluation, compaction, index/cache pressure, and operational process overhead before HA or multi-region replication. Storage uses a 1-5 TB / 30 days operational envelope, with 345-690 GB / 30 days as the mathematical raw sample floor before WAL, index, chunks, compaction, and metadata overhead.
- Monarch storage is expressed as an illustrative 1-10 PB distributed telemetry-storage envelope because the cited public paper reports close to 1 PB compressed in-memory scale, but does not disclose a per-datacenter allocation.
- ESC storage is modeled as 50-500 GB because ESC does not archive irrelevant occurrence events; it stores deviation history, selected evidence metadata, configuration/state snapshots, and bounded operational audit records.
- None of these values should be read as a statement by, certification from, or endorsement by the referenced vendors or authors.

4. SolarWinds Quantitative Anchors (Illustrative)

SOLARWINDS POLLING AND ENGINE ENVELOPE	
Item	Value
Node/interface status polling	120 s
Interface statistics polling	540 s
Node statistics polling	600 s
Interface statistics throughput (500k / 540 s)	≈ 925 polls/sec
Status polling throughput (500k / 120 s)	≈ 4,167 polls/sec
NPM scenario (12k elements/engine)	≈ 41.7 engines
NAM/SWO scenario (48k elements/engine)	≈ 10.4 engines

No single server-footprint number is asserted because SolarWinds topology depends on edition, module selection, architecture choices, deployment policy, and vendor guidance applicable to a specific customer environment.

Figure G1 - Polling-path request pressure asymmetry

Measuring polling intensity clarifies baseline supervision pressure before any deep diagnostics.

925 polls/sec on interface statistics versus 4,167 polls/sec on status polling.

Under this model, status-path polling creates higher continuous request pressure than the interface-statistics path, increasing modeled baseline polling infrastructure load.



Figure G2 - SolarWinds engine-footprint sensitivity by capacity model

Engine-count sensitivity indicates how architecture choices convert load into infrastructure footprint.

41.7 engines in the NPM scenario versus 10.4 in the NAM/SWO scenario.

The cited capacity-per-engine scenarios differ by roughly four, showing that architecture and product-scenario assumptions materially affect modeled CAPEX pressure.



5. Prometheus / Mimir Quantitative Anchors (Illustrative)

PROMETHEUS AND MIMIR CAPACITY ANCHORS	
Item	Value
Active series model	500,000 interfaces × 4 metrics = 2,000,000 series
Sample throughput (15 s scrape)	≈ 133,333 samples/sec
Distributor sizing anchor	1 core + 1 GB RAM per 25,000 samples/sec
Distributor baseline at 133k/s	≈ 5.3 cores; ≈ 5.3 GB RAM
Ingester sizing anchor	1 core + 2.5 GB RAM + 5 GB disk per 300,000 series
Ingester baseline at 2M series	≈ 6.7 cores; ≈ 16.7 GB RAM; ≈ 33.3 GB disk
Raw sample-storage lower bound	1–2 bytes/sample (Prometheus local TSDB guidance)
Raw monthly storage (pre-WAL/index/replication)	≈ 344.7–689.5 GB / 30 days

If an ingester replication factor of 3 is used, in-memory series pressure is multiplied at cluster level under the modeled assumptions.

Figure G3 - Prometheus always-on ingest baseline versus deviation-conditioned ESC load

In this model, Prometheus-class systems operate from a continuous ingest baseline, while ESC load is deviation-conditioned by design.

Prometheus ingest anchor is ~133,000 samples/sec; ESC realistic envelope is 30,000-600,000 updates/sec depending on active concurrency.

The structural constraint is economic: continuous ingestion creates fixed processing pressure even when anomalies are absent.

Prometheus continuous ingest baseline	~133,000 samples/sec
ESC realistic update envelope	30,000-600,000 updates/sec
Load posture contrast	continuous ingest vs deviation-conditioned

Figure G4 - Prometheus raw storage accumulation envelope

Storage load must be read as a range because efficiency and retention effects vary in production.

Raw throughput envelope is 133-266 KB/sec, equivalent to about 345-689 GB/month.

Even before WAL/index/replication overhead, monthly storage accumulates at substantial scale.

Raw throughput	Monthly storage
133-266 KB/s	345-689 GB/month

Figure G5 - Mimir ingest-path resource concentration (pre-replication)

In clustered TSDB designs, ingest-path roles determine where scaling pressure and recurring infrastructure cost accumulate.

Distributor baseline is 5.3 cores and 5.3 GB RAM; Ingester baseline is 6.7 cores and 16.7 GB RAM.

Under the cited sizing ratios, ingest-path pressure concentrates on ingesters before replication effects are applied.

Distributor CPU baseline	5.3 cores
Ingester CPU baseline	6.7 cores
Distributor RAM baseline	5.3 GB
Ingester RAM baseline	16.7 GB

6. Monarch-Class Telemetry Fabrics (Illustrative Role Baseline)

MONARCH PUBLIC SCALE SIGNALS	
Publicly reported characteristics	Interpretation in this paper
TB/sec ingestion capacity	hyperscale telemetry-fabric capability
millions of queries/sec	high-throughput shared backend behavior
close to 1 PB compressed in-memory	large distributed in-memory TSDB footprint

Monarch is used as a telemetry-fabric reference class based on the cited public research paper.

This paper intentionally avoids deriving per-datacenter server allocations from public data and does not treat Monarch as a directly deployable product comparator.

7. ESC Evidence-Activation Discipline

ESC FUNCTIONAL STACK (EDT/SOBT/COSAT/ESC)	
Layer	Function
EDT	occurrence detection
SOBT	observer-time structuring
COSAT	selective activation
ESC	decision layer

ESC operates upstream of telemetry materialization and governs when deeper observation activates.

Its purpose is not to increase observability volume. Its purpose is to determine when explanatory observability becomes justified.

That purpose is grounded in the two invariants introduced in Section 1.1:

- **Occurrence:** ESC reasons from bounded manifestations of activity, not raw data volume.
 - **Observer-time reference:** ESC structures those manifestations in the observer's comparative frame, avoiding dependence on a globally authoritative source timeline.
-

8. ESC Occurrence-Vector Scaling Model (Illustrative)

To keep the model operationally interpretable, ESC sizing is expressed with an explicit per-interface per-tick occurrence limit:

- **hard cap:** up to 15 occurrences/interface/tick (one per configured filter in this model);
- **recommended planning cap:** 1–2 occurrences/interface/tick sustained at fleet scale.

This yields a simple bound at 250 ms default tick:

$$\text{updates/sec} = \text{interfaces} \times \text{occurrences/interface/tick} \times 4$$

This formula follows directly from the occurrence invariant: **occurrences**, not raw bytes, packets, logs, traces, or samples, are the modeled scaling unit. The tick term follows from observer-time reference: occurrence behavior is structured in a comparative observation frame.

For 500,000 interfaces:

- 1 occurrence/interface/tick -> 2,000,000 updates/sec
- 2 occurrences/interface/tick -> 4,000,000 updates/sec
- 15 occurrences/interface/tick (hard cap) -> 30,000,000 updates/sec

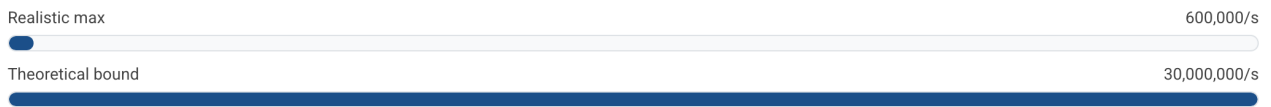
ESC UPDATE-RATE CAPACITY MATH		
Parameter	Value / Formula	Result
Counter space upper bound	500,000 interfaces × 15 filters	7,500,000 counters
Structuring tick example	250 ms	4 ticks/sec
Theoretical update bound	7,500,000 × 4	30,000,000 updates/sec
Realistic concurrency assumption	0.1% – 2% active	30,000 – 600,000 updates/sec

Figure G6 - ESC realistic envelope versus theoretical ceiling

Separating practical envelope from hard-cap arithmetic avoids planning from worst-case theoretical extremes.

Realistic range is 30,000-600,000 updates/sec, while theoretical upper bound is 30,000,000 updates/sec.

Daily planning should anchor on realistic concurrency, because hard-cap arithmetic is a resilience boundary, not an economic operating target.



8.1 Stateful Dictionary Pressure (Scaling Risk Driver)

Deployment feasibility must include both:

- update-rate arithmetic capacity;
- state-cardinality pressure in large hash-indexed key tables.

At high key counts, practical pressure includes metadata overhead, locality loss, resize transients, and disturbance-time latency variance.

9. ESC Monitoring Traffic Model (Illustrative)

ESC MONITORING TRAFFIC ENVELOPE	
Parameter	Value
Compact occurrence update payload	10–24 bytes/update
Monitoring traffic envelope (30k–600k updates/sec)	0.30–14.4 MB/sec
Scope note	before batching/compression

Figure G7 - ESC monitoring transport envelope under realistic load

Translating update cadence into transport volume makes network impact directly auditable for operators.

Monitoring traffic envelope is 0.30-14.4 MB/sec from modeled payload and realistic update rates.

This keeps modeled ESC supervision traffic structurally compact, reducing the fixed transport burden relative to continuous high-fidelity telemetry assumptions.



10. ESC Capacity Envelope (Model + Operator Planning)

10.1 Sizing in 3 Steps (Simple Rule)

Use these three inputs:

- **N** = total interfaces (here: 500,000)
- **A** = active-interface ratio at a given moment (for example: 5%, 10%, 50%)
- **O** = average occurrences per active interface per tick (recommended planning range: 1 to 2)

At 250 ms default tick (4 ticks/sec):

$$\text{updates/sec} = N \times A \times O \times 4$$

This gives an operational load number that is easy to discuss with non-math audiences.

The variable **O** is deliberately an occurrence count, not a telemetry-volume proxy. This keeps the model tied to structured occurrence activity rather than raw link throughput or archive volume.

Figure G8 - ESC update-loop CPU envelope (model-only)

The update-loop view isolates arithmetic cost from lifecycle orchestration, control-plane behavior, and high-availability policy.

Modeled update-loop envelope is approximately 0.02-1.0 cores.

The economic implication is direct: raw compute is not the dominant cost center; governance and operating architecture are.

Modeled minimum update-loop load	~0.02 cores
Modeled maximum update-loop load	~1.0 core
Scope boundary	excludes lifecycle/control-plane/HA

10.2 Scenario Table (500,000 Interfaces, 250 ms Tick)

500,000-INTERFACE SCENARIOS (250 MS TICK)					
Scenario	Active interfaces (A)		Occurrences per active interface per tick (O)	Updates/sec	Traffic
Nominal	5%		1	100,000	8.0-19.2 Mb/s
Elevated	10%		2	400,000	32.0-76.8 Mb/s
Stress	50%		2	2,000,000	160.0-384.0 Mb/s
Theoretical cap	hard	100%	15 (max filters)	30,000,000	2.4-5.76 Gb/s

10.3 What 24 Cores Means

Keeping the proposed sizing at **24 cores** (reference 3 GHz/core):

- in a **high-performance implementation path** (optimized data layout and hot-path tuning), update-loop headroom can approach ~36M updates/sec;
- in a **standard production implementation path** (robust implementation without aggressive micro-optimization), update-loop headroom can be closer to ~14.4M updates/sec;
- therefore, **24 cores is an operator-grade planning target**, not a promise of permanent worst-case saturation coverage.

Path labels used in this section:

- **Standard production implementation path (~5k cycles/update)**: production-grade Rust/Go implementation with strong correctness and maintainability, without aggressive low-level throughput tuning.
- **High-performance implementation path (~2k cycles/update)**: optimized Rust/Go implementation with cache-aware structures, reduced allocation pressure, tight batching, and tuned hot paths (including optional low-level intrinsics/assembly where justified).

The strategic interpretation is straightforward:

- nominal and elevated scenarios are comfortably within envelope;
- stress scenarios remain in-range with disciplined implementation;
- theoretical hard-cap behavior is a resilience boundary, not a normal operating target.

Position of 24 cores across the sizing range (update-loop view only):

24-CORE UTILIZATION BY LOAD POINT			
Load point (updates/sec)	Utilization at 14.4M capacity (standard production path)	Utilization at 36M capacity (high-performance path)	Reading
100,000	~0.7%	~0.3%	very low load
400,000	~2.8%	~1.1%	low load
2,000,000	~13.9%	~5.6%	moderate load
30,000,000	~208%	~83.3%	hard-cap zone; only high-performance path stays below saturation

10.4 Operator Envelope (Illustrative, Disturbance-Grade)

OPERATOR PLANNING ENVELOPE PER NODE	
Planning dimension	Envelope
CPU per node	16-24 cores
RAM per node	32-64 GB
Topology	1-3 general-purpose servers depending on HA policy

These are prudent planning envelopes pending full lifecycle validation.

Figure G11 - ESC operator deployment envelope by HA posture

Operator-grade deployment framing converts model outputs into budgetable infrastructure envelopes.

Planning range is 16-24 cores and 32-64 GB RAM per node, with 1-3 servers depending on HA posture.

The strategic constraint is governance quality: compact footprints are credible only when activation discipline remains controlled.

CPU per node	16-24 cores
RAM per node	32-64 GB
Deployment topology	1-3 general-purpose servers

10.5 Sizing Interpretation Rule

Single-number sizing claims are invalid unless assumptions are explicit for:

- active concurrency;
 - disturbance profile;
 - structuring tick and retention;
 - proof-latency SLO;
 - HA mode.
-

11. Detection vs Explanation Latency Comparison (Illustrative)

DETECTION VS EXPLANATION TIMING		
Architecture	Detection latency	Explanation latency driver
SolarWinds-derived polling model	polling-bound; typically minutes for status/statistics in this model	polling window
Prometheus-derived scrape model	scrape-bound; 15 seconds in this model	rule window and query latency
Monarch reference public-paper	not normalized as a single latency constant in cited public source	regionalized ingestion/query architecture and workload profile
ESC (modeled)	250 ms (default planning basis)	structuring tick

ESC explanatory timing is modeled as bounded by structuring interval, subject to implementation/backpressure/validation.

Figure G9 - Detection-to-explanation timing compression profile

Timing to explanatory proof, not only detection, determines operational reversibility during incidents.

Modeled anchors are ESC 250 ms, Prometheus-derived 15 s scrape timing, and SolarWinds-derived polling windows at 2/9/10 min.

Faster explanatory structuring compresses ambiguity windows, shortening escalation duration and lowering bridge OPEX exposure.



12. Correlation Pipeline Comparison (Illustrative)

CORRELATION PIPELINE SHAPES	
Regime posture	Pipeline shape
Traditional telemetry	collect -> store -> query -> infer
ESC discipline	detect -> structure -> activate -> explain

ESC moves correlation pressure upstream from query-time to observation-time.

13. Storage Model Comparison (Illustrative)

STORAGE PRESSURE DRIVERS BY REGIME	
Architecture	Storage driver
SolarWinds-derived polling model	SQL retention window
Prometheus/Mimir-derived metrics model	sample retention and TSDB replication pressure
Monarch public-paper reference	distributed in-memory telemetry fabric
ESC	occurrence vectors plus deviation history

ESC targets structured behavioral storage posture rather than exhaustive telemetry payload retention.

14. Normalized Monitoring Load Comparison (Illustrative)

NORMALIZED MONITORING LOAD BY ARCHITECTURE		
Architecture	Scaling variable	Sustained monitoring load
SolarWinds-derived polling model	elements	~925 polls/sec (interface stats path)
Prometheus-derived scrape model	active series	~133k samples/sec
Monarch public-paper reference	telemetry streams	TB/sec fleet-scale ingestion (not normalized per DC/interface)
ESC (modeled)	occurrence filters	75k–1.5M updates/sec

ESC scales primarily with occurrence density, not raw link throughput.

15. Explanation Latency as Cost Driver

ESC reduces the delay between anomaly detection and trusted explanatory evidence. That delay is the economic variable this paper treats as first-order.

Incident cost scales with:

engineers × hourly cost × explanation delay

Example:

- 8 engineers;
- \$200/hr;
- 6h vs 2h ambiguity.

Difference:

- ≈ \$6,400 per incident;
- ≈ \$77,000 annually (12 incidents);

before SLA penalties.

Figure G10 - Ambiguity-duration OPEX exposure delta

Ambiguity duration can be read as a first-order economic variable in incident operations.

Estimated delta is \$6,400 per incident and about \$77,000 annually at 12 incidents.

Reducing time-to-explanatory-proof converts directly into OPEX relief, not merely technical refinement.

Per incident	\$6,400
Annual (12 incidents)	\$77,000

CAPEX and contractual context should be read together with this OPEX mechanism:

- a heavy continuous high-fidelity posture may be budgeted as an insurance strategy and can land in an illustrative **~\$8M-\$25M** three-year TCO range for large footprints under the appendix assumptions;
- contractual exposure can add **material monthly service credits** when SLA terms are tied to minutes of violation.

In short, incident economics are jointly shaped by:

- **bridge OPEX** (concurrency x burdened hourly cost x hours to trusted causal story), and
 - **insurance-shaped CAPEX** (continuous high-fidelity overprovisioning).
-

15.1 ESC Determinism and Implementation Accountability

ESC is defined here as a **bounded, deterministic discipline**.

Its model-level behavior is not presented as probabilistic best-effort logic.

ESC specifies explicit timing, activation, and control boundaries. In this framing, ESC itself is not the source of operational uncertainty: when outcomes degrade, root causes are attributed to implementation quality, calibration governance, or deployment discipline.

Accordingly, this paper treats ESC as:

- deterministic at model level (bounded timing and activation logic);
- governable at operations level (explicit thresholds and control loops);
- accountability-driven at engineering level (implementation correctness is the decisive factor).

Implementation Accountability Matrix

IMPLEMENTATION ACCOUNTABILITY CONTROLS				
Control domain	ESC design intent		Degradation cause	Responsible layer
Trigger precision	deterministic activation	selective	threshold/calibration error	implementation and operations governance
Drift handling	controlled adaptation	baseline	missing recalibration discipline	implementation and operations governance
Backpressure behavior	bounded degradation path		unbounded queues or missing shedding logic	implementation architecture
Filter/interface alignment	deterministic semantics	placement	mapping/configuration defect	integration implementation
Correlation timing	observer-time structuring	bounded	scheduler/runtime misconfiguration	runtime implementation

In short: this paper treats adverse outcomes as **implementation-accountability events**, not as invalidation of ESC's deterministic model.

15.2 Validation Methodology

The quantitative ESC model in this paper should be validated through reproducible experiments before any production claim is made.

A reviewer-ready validation plan should include:

- synthetic occurrence-load benchmarks;
- protocol-cyclic replay tests;
- transient-deviation replay using captured lab traces;
- SOBT ingestion and windowing benchmarks;
- COSAT activation-latency measurement;
- control-plane backpressure tests;
- false-positive and false-negative analysis;
- sensitivity analysis over tick interval, filter count, and occurrence concurrency.

Minimum metrics to report:

VALIDATION METRICS CHECKLIST	
Metric	Required measurement
occurrence updates/sec	sustained and burst
SOBT CPU/update	cycles or CPU percentage
SOBT memory footprint	rolling-window and retained state
monitoring traffic	bytes/sec before and after batching
activation latency	deviation detection to observation change
proof latency	deviation emergence to high-fidelity evidence availability
false activation rate	over-trigger frequency
missed activation rate	under-trigger frequency

Until full lifecycle validation exists, deployment-scale ESC figures must remain explicitly labeled as modeled estimates. Event-detection emission behavior is partially benchmark-backed in EVE-NG lab conditions, as summarized in Section 19.

16. Empirical Validation Status (EVE-NG)

To reduce model-only bias, this strategy paper includes partial benchmark grounding from controlled EVE-NG campaigns focused on event-detection emission behavior.

Currently benchmarked and validated in lab conditions:

- traffic/event ratio envelope under synthetic and cyclic protocol activity;
- CPU pressure induced by event-detection emission paths.

These benchmark results support the feasibility of the event-detection emission layer and its resource-pressure assumptions.

Not yet claimed as fully validated in this paper:

- end-to-end production-scale activation governance across heterogeneous fleets;
- long-horizon baseline drift behavior in live environments;
- complete false-positive and false-negative operating envelopes at sustained disturbance scale.
- disturbance-grade state-cardinality behavior of large in-memory key tables at hyperscale.

Accordingly, ESC values in this document should be interpreted as:

- partially benchmark-backed for event-detection emission behavior;
- model-based for broader deployment-scale governance and lifecycle behavior.

17. Operational Interpretation

ESC is modeled as replacing permanent explanatory fidelity with conditional explanatory readiness. This is the core CAPEX/OPEX shift defended by this paper.

Deviation-triggered explanatory activation enables:

- earlier causal confirmation;
- shorter escalation bridges;
- reduced capture footprint;
- lower continuous telemetry CAPEX under the modeled assumptions;

without reducing monitoring coverage.

18. Limitations

This paper does not claim:

- vendor-internal infrastructure parity;
- replacement of observability platforms;
- traffic-independent scaling;
- full production validation of all ESC pipeline layers across diverse operational environments.

ESC values represent:

- partial benchmark-backed estimates for event-detection emission behavior (EVE-NG validation scope);
- model-based estimates for broader activation governance and deployment-scale behavior.

The cost model is also intentionally limited. The formula:

$\text{engineers} \times \text{hourly cost} \times \text{explanation delay}$

is an illustrative heuristic, not a complete financial model.

It does not include:

- blast-radius variance;
- customer opportunity cost;
- nonlinear SLA penalties;
- regulatory impact;
- reputational damage;
- repeated mitigation cost;
- business interruption beyond the technical bridge.

The model should therefore be read as a lower-bound way to reason about ambiguity cost, not as a comprehensive economic proof.

Finally, the comparative model intentionally avoids estimating Monarch's per-datacenter server footprint because the cited public Monarch paper does not disclose such allocation. Monarch is used as a hyperscale telemetry-fabric reference, not as a directly deployable comparator.

19. Conclusion

Observability architectures historically optimized:

- coverage;
- retention;
- aggregation;

but not explanatory timing.

That omission matters because detection without timely explanatory access still leaves organizations paying for ambiguity.

ESC introduces a different optimization target:

| explanatory evidence aligned with deviation relevance

rather than:

| telemetry everywhere at all times

The economic interpretation presented here is derived from modeled deployment-scale behavior combined with emission-layer benchmarks, and should be read as a strategy-supporting estimate rather than a finalized infrastructure cost model.

A conservative operator interpretation is:

| ESC can materially reduce observability infrastructure footprint relative to continuous high-fidelity collection approaches under the modeled assumptions, potentially down to a compact 1–3-node general-purpose deployment at the 500k-interface order of magnitude, subject to full lifecycle validation and explicit SLO assumptions.

At hyperscale line rates, this distinction may materially affect the economic floor of observability, subject to empirical validation.

The strategic close is direct:

| the next advantage is not more telemetry.
| it is more decisive evidence, at the moment decisions are still affordable.

At hyperscale, the cost of observability is no longer only the cost of seeing. It is the cost of seeing too late.

20. Strategic CAPEX/OPEX Thesis

The central claim is strategic, not product-comparative:

| observability economics at hyperscale are increasingly dominated by ambiguity duration, not by telemetry volume alone.

ESC therefore reframes observability policy from “collect more” to “activate proof sooner.” The objective is not less visibility; it is better-timed explanatory visibility.

In this framing:

- **CAPEX discipline** means avoiding permanent overprovisioning of high-fidelity capture paths as an insurance default where selective activation can satisfy the operational proof requirement;
- **OPEX discipline** means reducing multi-team bridge duration by improving time-to-explanatory-proof;
- **technology choice** is subordinated to proof-timing policy, not vice versa.

ESC is positioned as a decision discipline that authorizes when richer evidence should exist, while preserving existing telemetry stacks.

This strategic posture should also be read with three explicit boundaries:

- not a replacement for compliance-grade logs or flow records;
- not a generic archive optimization program;
- not a storage story dressed as innovation.

20.1 From "Apology SLAs" to "Visibility SLAs"

In many operating models, conventional SLA credits are financially necessary but operationally retrospective. An ESC-oriented posture adds a stronger proposition: faster access to trusted explanatory proof while the event still matters.

Strategic reading:

- **apology SLA logic**: compensate after impact;
- **visibility SLA logic**: shorten ambiguity while decisions are still reversible.

20.2 The Customer Also Pays for Ambiguity

Delayed proof is not only a provider-side bridge-cost issue. It also extends customer-side uncertainty through:

- delayed business decisions;
- duplicated mitigation effort;
- longer disruption handling windows;
- reduced confidence in future transient explainability.

This is why proof timing affects commercial credibility, not only internal efficiency.

20.3 Leadership-Level Implication

At hyperscale line rates, the board question is no longer "how to store more telemetry." It becomes:

how to avoid funding permanent high-fidelity insurance everywhere when governed activation can still produce decisive proof on operational time.

That is the governance shift from telemetry inflation to deviation-driven proof authorization.

At operating level, the principle is pragmatic:

the organization should not have to buy the same crisis twice - once as infrastructure, and once as calendar time.

21. Strategy Choice Framework

The practical question for leadership is whether current observability spend buys proof when decisions are still reversible, or only archives evidence for later reconstruction.

This paper defends a strategy selection logic based on five decision questions:

1. Is ambiguity duration a material cost driver in incident response?
2. Is current telemetry strong for detection but weak for timely explanation?
3. Are teams paying twice (continuous infrastructure + delayed human correlation)?
4. Can deviation-triggered activation reduce permanent high-fidelity footprint without reducing operational confidence?
5. Can the discipline be governed with measurable false-positive/false-negative boundaries?

If most answers are yes, a deviation-triggered evidence discipline is strategically justified.

21.1 CAPEX Policy Implications

A CAPEX-oriented strategy should prioritize:

- bounded continuous awareness planes;
- selectively activatable explanatory instrumentation;
- control-plane scalability for trigger-driven escalation.

This shifts investment from blanket capture capacity to evidence-authorization logic.

21.2 OPEX Policy Implications

An OPEX-oriented strategy should prioritize:

- reduction of ambiguity windows;
- shorter escalation bridges;
- faster causal narrowing under transient instability.

This shifts operating discipline from retrospective correlation to timely explanatory access.

22. Role of Quantitative Illustration

The technical comparisons in this document are intentionally illustrative and bounded.

They are used to:

- stress-test order-of-magnitude feasibility;
- compare latency and resource-pressure shapes across observation regimes;
- support strategy discussion with conservative envelopes.

They are not used to claim product substitution, vendor equivalence, benchmark supremacy, or vendor-certified deployment economics.

22.1 Board Decision Snapshot (CAPEX/OPEX)

This is the compact board-level version of the thesis:

ESC changes the funding model from permanent high-fidelity insurance to governed evidence activation. Named third-party products and systems remain reference anchors only; the board-level choice is architectural and economic.

BOARD CAPEX/OPEX CHOICE SNAPSHOT		
Decision axis	Legacy default	ESC-oriented discipline
CAPEX policy	Permanent high-fidelity capacity as insurance	Bounded continuous awareness + selective deeper activation
OPEX policy	Bridge-heavy, retrospective correlation	Ambiguity-window reduction via earlier explanatory access
SLA posture	Service credits as primary remediation	Faster explanatory access as operational confidence layer (credits remain contractual backstop)
Operating logic	Collect broadly, explain later	Detect deviation, then deepen evidence
Economic risk	Rising fixed footprint + repeated escalation labor	Trigger quality risk (must be governed and validated)
Governance requirement	Retention and coverage controls	Activation precision, drift control, and backpressure controls

This is the primary board-level choice defended by this paper.

23. Contact and NDA Validation Package

This paper is intentionally published as an open strategic document.

For organizations evaluating adoption, a deeper validation package can be discussed under NDA, including:

- benchmark protocol details and scenario matrices;
- selected raw benchmark outputs and replay traces;
- calibration logic and operating-threshold governance approach;
- implementation architecture details and integration constraints.

The objective of NDA discussion is to move from strategic fit assessment to evidence-based deployment planning.

24. Methodological and Disclosure Notes

24.1 Disclosure and Scope Boundaries

- **Disclosure note:** This document presents a strategic CAPEX/OPEX model supported by partial empirical validation. The benchmarked scope concerns event-detection emission behavior in EVE-NG lab conditions; full ESC/SOBT/COSAT lifecycle validation at hyperscale remains outside the scope of this public paper.
- **Potential conflict of interest:** The author is the named inventor/applicant for patent-pending ESC-related methodologies.
- **Third-party reference note:** SolarWinds, Prometheus, Grafana Mimir, Monarch, Google, and other third-party names are referenced solely to identify public documentation, public specifications, or published research used as numeric anchors. The references do not imply endorsement, sponsorship, affiliation, certification, or approval.
- **No vendor-certified sizing:** All derived CPU, RAM, storage, throughput, and latency values are author-side calculations from cited public material and stated assumptions. They should not be treated as official sizing guidance from any third-party vendor or project.

Detailed benchmark methodology, graphs, and interpretation are provided in the companion Vision Paper and benchmark appendix published on the ESC site. This document relies on those results only for the event-detection emission layer and does not claim full production-scale ESC validation.

24.2 Methodological Caution

SolarWinds, Prometheus/Mimir, Monarch, and ESC are not treated as substitutable products.

They are compared only as observation regimes with respect to ingestion model, storage pressure, correlation timing, and explanatory-proof latency. The comparison is not intended to disparage, rank, certify, or benchmark any third-party product.

24.3 Reading Guide

This paper is a strategic CAPEX/OPEX position paper.

Regime-level comparisons are included as illustrative technical evidence, not as the central argument.

24.4 Numeric Evidence Policy

Every externally sourced numeric constant in this document is tied to an exact public reference URL.

Derived values are deterministic arithmetic from those constants and from explicitly stated model assumptions. Where this document uses additional normalization assumptions, such as per-engine raw CPU/RAM/storage envelopes, those assumptions are identified as author-side modeling assumptions rather than vendor-published requirements.

24.5 Executive Board Summary

This paper argues for a strategy shift in observability investment policy:

- recognize that more telemetry does not automatically create earlier proof;
- distinguish collection capacity from explanatory activation timing;
- treat conditional evidence activation as a governance layer;
- treat **time-to-explanatory-proof** as a first-order economic variable;
- control CAPEX growth by avoiding permanent high-fidelity overprovisioning;
- control OPEX by reducing ambiguity windows during incidents;
- preserve existing telemetry platforms and add a deviation-triggered evidence discipline upstream;
- validate with measurable activation quality (false-positive / false-negative / latency bounds).

Modern observability infrastructures rarely fail because telemetry is absent. They fail because explanatory proof often arrives after deviation has already triggered escalation workflows.

At hyperscale line rates, transient instability lasting seconds can produce multi-hour coordination loops despite complete monitoring coverage. This reveals a structural gap between telemetry architectures optimized for archival completeness and operational requirements centered on timely causal access.

ESC is presented as a deviation-triggered explanatory instrumentation discipline operating upstream of conventional observability stacks. ESC does not replace logs, metrics, packet capture, or time-series databases. Instead, it structures when explanatory evidence should exist.

24.7 Trademark and Nominative Use Notice

SolarWinds, Prometheus, Grafana Mimir, Monarch, Google, and any other third-party names referenced in this paper are trademarks, trade names, project names, or publication names of their respective owners.

They are used solely to identify public documentation, public capacity guidance, or published research sources relevant to the comparative model.

No sponsorship, endorsement, affiliation, certification, approval, or commercial relationship is implied.

This paper does not claim that ESC is a drop-in replacement for any named product or service, nor does it claim that any named product performs according to the modeled figures in all deployments.

All comparisons are illustrative, architecture-level, and assumption-bound.

24.6 Definition of Explanatory Proof

For the purposes of this paper, **explanatory proof** means high-fidelity evidence sufficient to support, reject, or materially narrow a causal hypothesis during an operational incident window.

It does not mean:

- legal proof;
- complete packet-level reconstruction of all activity;
- permanent retention of all telemetry;
- a guarantee that causality can always be established.

The paper therefore distinguishes between **symptom detection** and **operationally useful causal evidence**.

References

- [SolarWinds Platform Scalability Engine Guidelines](#) (elements-per-engine limits; product-dependent scenarios such as NPM ~12k and NAM/SWO 48k; standard polling frequencies)
 - [SolarWinds Polling Interval Settings](#) (default 120-second node/interface status polling interval)
 - [SolarWinds Polling Statistics Interval Settings](#) (default 10-minute node statistics and 9-minute interface statistics intervals)
 - [Prometheus Storage Documentation](#) ("Prometheus stores an average of only 1-2 bytes per sample" and storage planning formula)
 - [Grafana Mimir Capacity Planning](#) (Distributor and Ingester CPU/RAM/disk sizing ratios)
 - Adams et al., [Monarch: Google's Planet-Scale In-Memory Time-Series Database \(VLDB 2020\)](#) (terabytes/sec ingestion, millions of queries/sec, close to 1 PB compressed in-memory scale indicators)
-