The Resilience Newsletter

THALAMUS ADVISORY

ISSUE 02 SEPTEMBER 2026 READING TIME · 12 MIN SERIES · FOUNDATIONS

FAILURE MECHANISMS

Probe Windows Anchored to the Wrong Hour

Your automated investigation queried the last sixty minutes. The incident started four hours ago. Every signal came back flat, the agent reported no anomaly, and it was right — about an hour in which nothing happened.

THALAMUS ADVISORY • ENTERPRISE RESILIENCE PRACTICE

01 — THE SIGNAL

An investigation that finds nothing

There is a particular kind of incident report that should worry you more than a failed one. It arrives complete. It says traffic was normal, error rates were nominal, latency was within bounds, and no anomaly was detected. It concludes — reasonably, given what it examined — that the reported symptom is inconsistent with a service degradation and is probably a monitoring artefact.

Then a customer calls, and the outage is four hours old.

The agent did not malfunction. It queried telemetry over a sixty-minute window ending at the moment the query ran. The incident began at 11:20. The investigation started at 15:40. The window it examined — 14:40 to 15:40 — is a period during which the system had already stabilised, or failed over, or simply moved on. Every number it returned was accurate. None of them described the event.

11:20

Onset. Connection pool saturates. Error rate climbs to 40%.

11:55

Automatic failover. Errors subside. The system looks healthy again.

15:40

A customer escalation finally reaches engineering. Investigation opens.

15:40

The agent queries "the last 60 minutes" — 14:40 to 15:40. It sees a healthy system and says so.

This is not an exotic edge case. It is the default behaviour of almost every observability query, dashboard and diagnostic tool ever built, because they were all designed around a human who was *already looking*.

02 — THE MECHANISM

A convention that stopped being safe

"Last 15 minutes," "last hour," "last 24 hours." Relative time ranges are the universal default. For a human at a dashboard during an active incident, they are exactly right — the interesting thing is happening now, and a relative window follows you as you work.

The convention encodes an assumption: **that the observer is present at the time of the event.** Break that assumption and the convention silently inverts from helpful to actively misleading.

Three things break it, all of which are now normal:

- **Detection latency.** The gap between onset and anyone noticing. For customer-reported issues this is routinely hours.
- **Queue latency.** Problems land in a queue and are dispatched in priority order. A SEV3 raised at 11:20 may be investigated at 16:00.
- **Self-healing.** Retries, failover and autoscaling resolve the symptom while leaving the cause. The window in which evidence exists is narrow and closes on its own.

> *A relative time window assumes the observer was there. Automation is specifically the practice of not being there.*

Why this is uniquely hard to notice

Ordinary bugs announce themselves. This one produces a well-formed report full of accurate numbers. Nothing errors. No query fails. Your monitoring of the monitoring is green.

Worse, the failure is *self-confirming*. The report says no anomaly was found. The natural inference is that the alert was noisy — so the team tunes the alert threshold up, which reduces future detection, which increases the gap between onset and investigation, which makes the next misaligned window more likely. A team can reason its way into blindness one sensible decision at a time.

> THE TWO-MINUTE DIAGNOSTIC
>
> Take any automated investigation report from the last month. Find the incident's onset timestamp. Find the time window each probe actually examined.
>
> **Does the window contain the onset?** In most implementations we have reviewed, for any incident investigated more than an hour after it began, it does not.

03 — THE ENTERPRISE PROBLEM

Where the cost accumulates

CONSEQUENCE	HOW IT COMPOUNDS
False all-clears	An incident is closed as unreproducible. The cause persists and recurs — often repeatedly — before anyone connects the occurrences.
Alert tuning in the wrong direction	Repeated “no anomaly found” results are read as noise. Thresholds are relaxed. Detection latency grows, which makes the original problem worse.
Change correlation breaks	“What changed before this started?” becomes “what changed before I asked?” The deploy that caused a 04:00 failure is invisible to a 09:00 investigation looking at a rolling window.
Retention becomes a silent ceiling	If traces are kept 3 days and a pattern is noticed on day 4, the investigation cannot be performed at all — and nothing reports this as a failure.
Postmortems built on the wrong hour	The permanent record describes a window that did not contain the event. Action items are raised against systems that were healthy.

The deepest cost is epistemic. An organisation can accumulate hundreds of investigations that found nothing and conclude its estate is in better shape than it is. Confidence rises as visibility falls.

04 — THE THREAT LANDSCAPE

Why adversaries benefit from this

A defender who can only examine the recent past has told every attacker exactly how long they need to wait.

Dwell time already exceeds the window by orders of magnitude

Intrusions are routinely measured in days to weeks between initial access and detection. Set that against a diagnostic function anchored to “the last hour” and the asymmetry is absolute: by the time anyone investigates, the evidence window has moved on repeatedly, and the only tooling available is pointed at the present.

AI has made low-and-slow cheap

Patience used to cost an adversary something — attention, coordination, skilled operator time. Automation has collapsed that cost. Reconnaissance can be spread thin across weeks without a human maintaining the campaign, precisely because the marginal cost of waiting has gone to nearly zero.

The defensive consequence is direct: **detection latency is increasing at the same time as the attacks worth detecting are getting slower.** Both trends widen the gap between when something happened and when anyone looks, and that gap is exactly what a relative window cannot span.

THE PATTERN TO WATCH FOR

An adversary generates a modest, plausible anomaly — enough to trigger an alert, not enough to page a human. It is queued. By the time automated investigation runs, the window has moved past both the decoy *and* the real activity it was covering.

The investigation reports no anomaly. The alert is marked noisy. The threshold is raised. **The attacker has just used your diagnostic tooling to negotiate a larger operating envelope**, and every step of that process looked like good hygiene from the inside.

Forensics has the same problem, with legal consequences

Breach investigation is time-anchored by definition: what happened, when, and what was accessed. If the investigation tooling cannot address a window that closed weeks ago — because the interface assumes “recent” or the retention expired — the organisation cannot answer the questions regulators ask under GDPR, SEC disclosure rules or DORA. “We were unable to determine the scope” is not a neutral finding. It is assumed to be the worst case.

05 — THE REMEDY

Anchor to the event, not the observer

1. Make onset a first-class property

Every incident record must carry the timestamp of when the condition *began*, not when it was detected, reported or queued. These are four different times and most systems conflate them.

Onset must survive the whole pipeline. In estates we have reviewed it is typically discarded at one of three points: a webhook receiver stamping records with receipt time; a queue table storing only `received_at`; or a service-health object carrying no timestamp at all. Any one of these silently

reintroduces the bug downstream.

2. Every probe window ends at onset

The window should be `[onset - N, onset]`, not `[now - N, now]`. The run-up is what explains a breach. Extending past onset mixes the fault together with its own blast radius and makes attribution harder, not easier.

3. Make the window visible in the output

A probe should report *"the 60 minutes before 2026-09-10T11:20Z"*, not *"the last 60 minutes."* The two are indistinguishable in a report and mean completely different things. Stating the window makes a misaligned investigation obvious to any reader at a glance — which turns an invisible failure into a visible one.

4. Align retention with investigation latency, not budget

Plot the distribution of time-from-onset-to-investigation. Compare it to your retention window. The overlap is the set of incidents you are capable of diagnosing; everything beyond it is undiagnosable by construction.

Tiered retention resolves most of this economically: keep error traces materially longer than successful ones. The cost difference is large, because errors are a small fraction of traffic, and the diagnostic value is asymmetric in the opposite direction.

5. Anchor downstream systems too

Log-regression baselines, change correlation and anomaly detection all need the same treatment. A log-group regression comparing "recent" against "baseline" must be able to take an explicit anchor, or it re-anchors to its own clock and quietly reintroduces the bug in a system you already fixed.

06 — IN PRACTICE

How the Thalamus products address this

Thalamus SRE — onset-anchored investigation

Every specialist probe closes its window at the incident's onset rather than the wall clock. The onset is carried from the source problem through the queue to the investigation, so a run dispatched hours later still examines the hour that matters.

The Change Correlation specialist ranks change records by proximity to *onset* — the distinction that makes "what changed just before this started" answerable the next morning rather than only during the event.

Probe descriptions state their window explicitly. A specialist reports examining *"the 60m before 2026-09-10T11:20Z"*, so a misaligned investigation is visible in the transcript instead of hidden inside an accurate-looking summary.

> WHERE THIS CAME FROM
>
> This issue is not theoretical for us. We shipped this bug. Our investigation agents were anchoring probes to query time, and the symptom was exactly as described above: confident reports of flat metrics and no anomaly.
>
> What made it findable was that the specialists reported *"probe unavailable, 0 rps"* and the pipeline concluded the evidence was **consistent with a telemetry gap rather than a service degradation**, at low confidence, and escalated to a human instead of inventing a cause. The reasoning was sound; the window was wrong. Fixing it meant threading onset through the webhook, the queue and the health object — the three places it had been dropped.

ThalamusTrace — telemetry that accepts an anchor

Span timeseries, failure breakdowns and critical-path aggregation all take explicit `start` and `end` bounds rather than only a rolling window. Log-group regression accepts an explicit anchor, so the recent-versus-baseline comparison is made against the incident's hour rather than the query's.

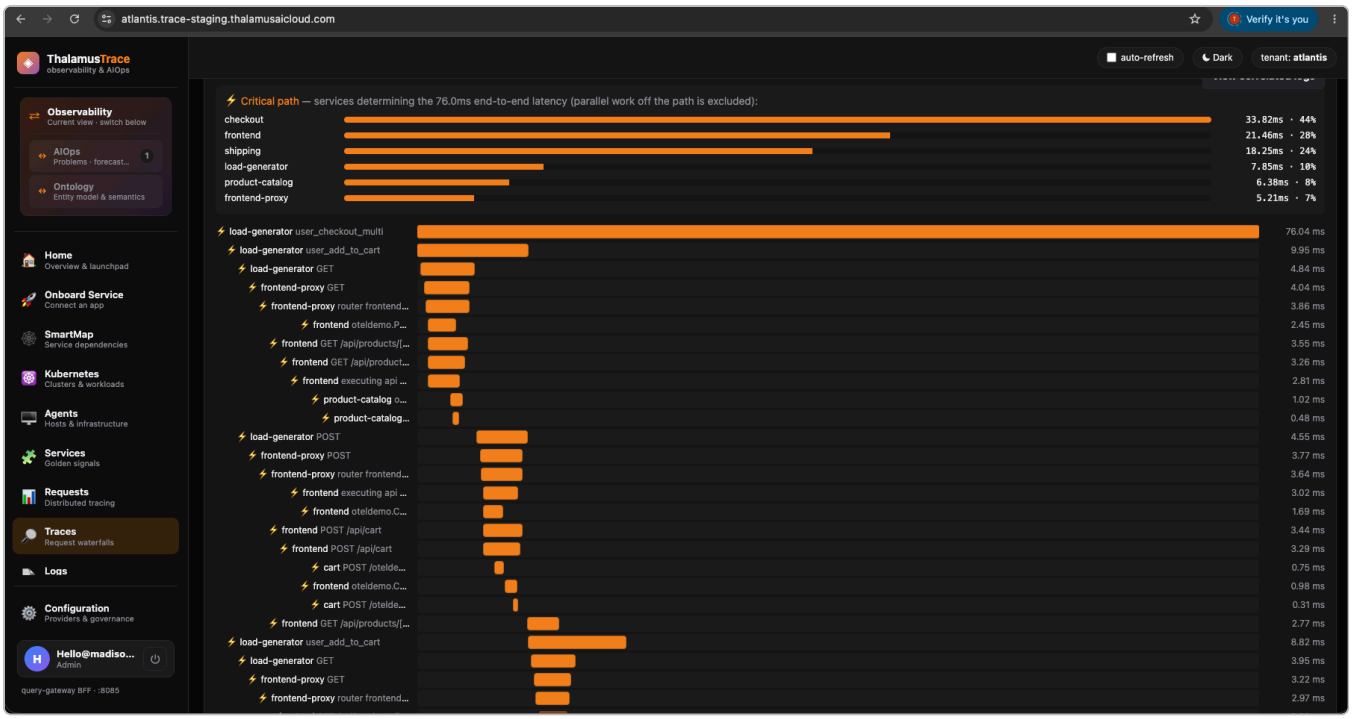

FIGURE 2 — THALAMUSTRACE · CRITICAL PATH

Which hop actually held the latency. Critical-path aggregation attributes the 76.0ms end-to-end across the services that determined it — checkout 44%, frontend 28%, shipping 24% — explicitly excluding parallel work off the path. Because the analysis takes explicit start and end bounds rather than a rolling window, the same attribution can be produced for a window that closed days ago.

Retention is tiered: error traces are kept materially longer than normal ones, so the traces most likely to be needed by a late investigation are the ones still present.

Problems carry `startedAtUnixNano` — the onset — and it is that value, not receipt time, that flows downstream through signed webhooks to whatever consumes them.

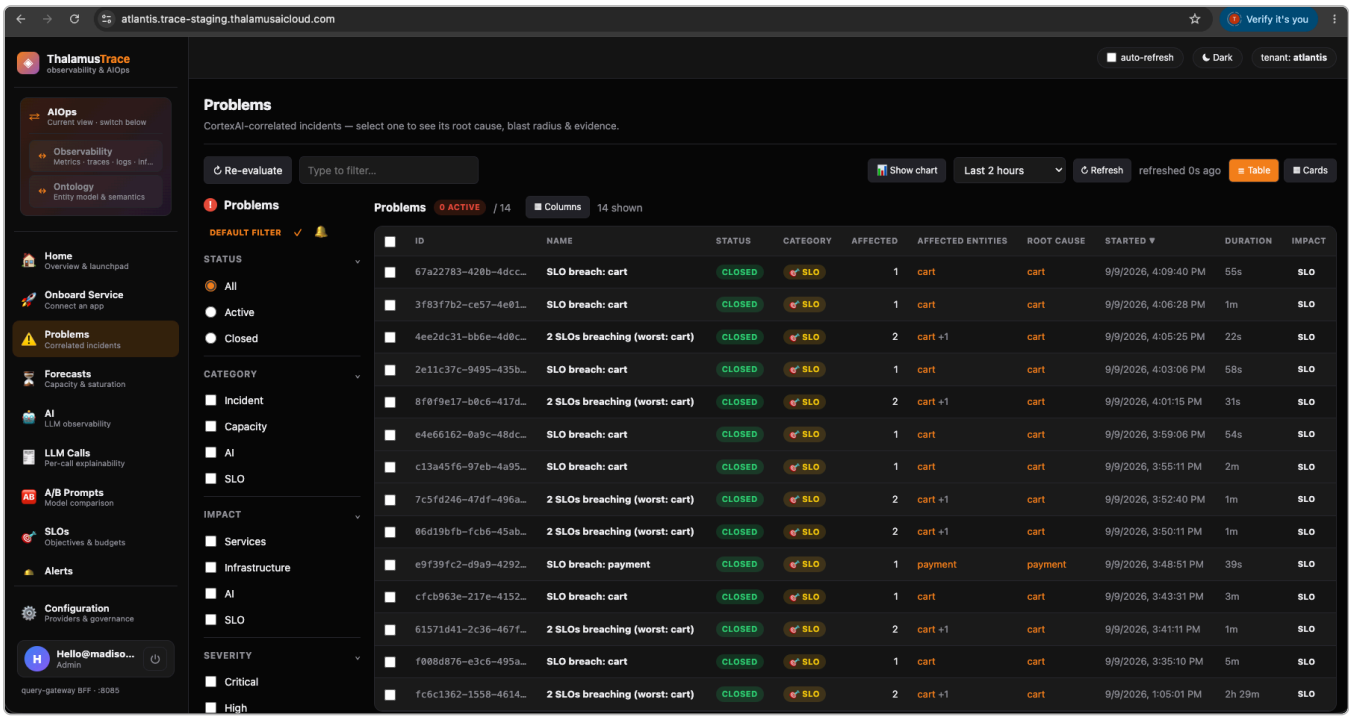

FIGURE 1 — THALAMUSTRACE · PROBLEMS

Onset as a first-class column. Every correlated problem carries *Started* and *Duration* beside its root-cause entity and affected entities — so an investigation opened hours later still knows which hour to examine. Note the correlation at work: *"2 SLOs breaching (worst: cart)"* across two affected entities is one problem with one named cause, not two pages to two teams.

07 — GETTING THERE

How Thalamus Advisory helps

ENGAGEMENT	WHAT IT PRODUCES
Window Alignment Review 2 weeks	We take a sample of your automated investigations and check, mechanically, whether the window examined contained the onset. Deliverable: the proportion that did not, and where onset is being dropped in your pipeline.
Retention Economics 2–3 weeks	Your time-from-onset-to-investigation distribution plotted against current retention, with a tiered-retention model sized to your actual error ratio. Usually reduces cost while increasing diagnosable coverage.
Incident Archaeology 3–4 weeks	We re-investigate a set of incidents previously closed as unreproducible, using correctly anchored windows. Typically recovers several genuine root causes and provides the business case for the rest of the programme.
Detection Latency Reduction 6–8 weeks	Attacks the upstream cause. SLO design so conditions are detected by objective breach rather than customer report, closing the gap the misalignment exploits.
Forensic Readiness 4 weeks	For regulated clients: whether you could reconstruct a defensible timeline for an event discovered 30 days later, assessed against the reporting deadlines that actually apply to you.

We deploy self-hosted where the security boundary requires it, including classified and regulated enclaves.

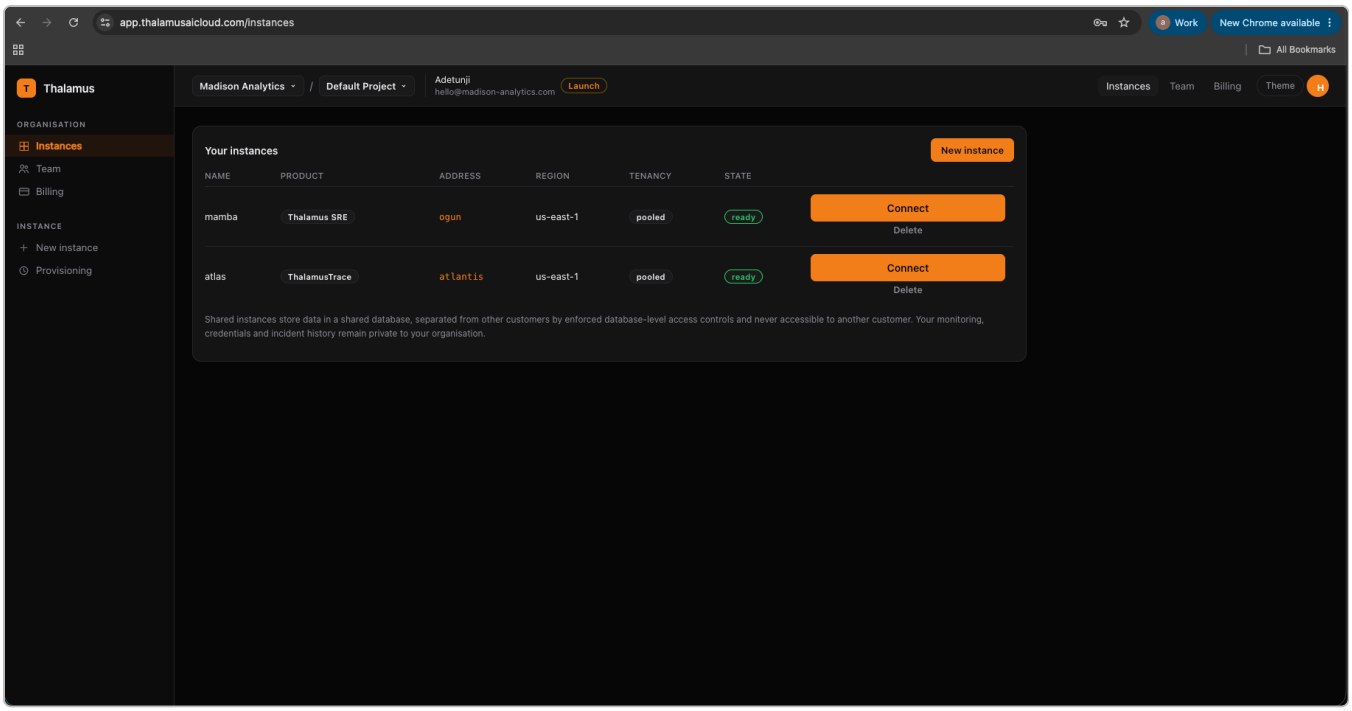

FIGURE 3 — THALAMUS AI CLOUD

From nothing to instrumented in an afternoon. ThalamusTrace and Thalamus SRE run as managed instances, so an Incident Archaeology engagement begins against live telemetry rather than a procurement cycle. OTLP-native ingest means the instrumentation is the OpenTelemetry standard rather than a vendor agent — which is also what keeps the data portable back out again.

THE COUNTER-ARGUMENT, MADE HONESTLY

Onset is not always knowable. For gradual degradation there may be no clean moment when the condition began, and a precise-looking onset timestamp can be a false precision that misleads in its own way.

Longer retention also costs real money, and for a high-volume estate the figure is not trivial. An organisation whose incidents are overwhelmingly detected and investigated within minutes genuinely does not have this problem, and should not spend against it.

The honest test is the distribution, not the anecdote: **measure your time-from-onset-to-investigation before deciding.** If the ninetieth percentile is under twenty minutes, close this issue and spend the money elsewhere. In most enterprises we have measured, it is measured in hours.

ONE NUMBER

Four hours and twenty minutes. The gap between onset and investigation in the worked example above — and a wholly unremarkable figure for a customer-reported issue in a large enterprise.

Against a sixty-minute rolling window, that gap does not degrade the investigation. It makes it examine an hour with no relationship to the event, and report the result with total confidence.

Where to start on Monday

Open your last three incidents that were closed as "could not reproduce" or "no anomaly found." For each, write down two timestamps: when the condition began, and what window the investigation actually examined.

If those do not overlap, you have not had three unreproducible incidents. You have had three unexamined ones — and the causes are still in your estate.

That check takes twenty minutes. Re-investigating them properly is the Incident Archaeology engagement above, and it is usually where the business case writes itself.

Thalamus Advisory

Enterprise resilience for the age of autonomous systems. We help organisations build operational practice that stays trustworthy when the systems making decisions are no longer entirely human.

THE RESILIENCE NEWSLETTER · ISSUE 02 · SEPTEMBER 2026
Every figure we print carries its assumptions. Where we model, we say so.
Written for engineering leaders. Forward it to one.