Your monitoring didn't miss the incident.

It was never designed to see it.

threshold monitoring

≠

behavior monitoring

Modern systems don't fail by crossing thresholds.

They fail by behaving differently.

WHAT YOUR STACK WATCHES FOR

- CPU spike
- Memory ceiling
- Disk full
- Process exit

HOW MODERN SYSTEMS FAIL

- Latency drifts
- Cost compounds
- Routing shifts
- Efficiency drops

You're watching:

- CPU
- Memory
- Latency (avg)
- Error rate

But failure starts somewhere else.

Behavior. Decisions. Drift.

These observability gaps
don't trigger alerts.
They trigger incidents.

Consumption Velocity

It's not how much you use.
It's how fast usage is changing.

- AI systems → token burn rate rises before cost spikes
- Microservices → request fan-out grows before latency breaks
- Pipelines → retries compound before queues back up

Distribution Drift

Averages look fine.
The tail is already failing.

02

- P95 creeps up while average stays flat
- A subset of requests gets slower and heavier
- Specific workflows deviate from normal patterns

Decision Pattern Changes

When the system starts choosing differently, it's already under stress.

03

- More requests routing to the expensive model
- Fallback paths activating more frequently
- Retries rising without corresponding error spikes

Retry Amplification

Retries don't fix failure.
At scale, they create it.

- 1 failure → 3 retries → downstream pressure
- Downstream pressure → more retries → loop
- Error rates spike only after the system saturates

Cache Miss Rate

Systems don't get slower first.
They get less efficient first.

05

- KV cache miss in LLM inference → cost increase
- Semantic cache degradation in RAG pipelines
- CDN / object cache inefficiency compounds silently

None of these trigger alerts.

But all of them predict failure.

No hard thresholds crossed	No immediate failures	No obvious incident

Together, they form a pattern:
the system is drifting out of its normal operating state.

You don't need more alerts.

You need different signals.

Full breakdown:

How modern systems fail before they break

Link in first comment →

Follow for weekly infrastructure architecture breakdowns.