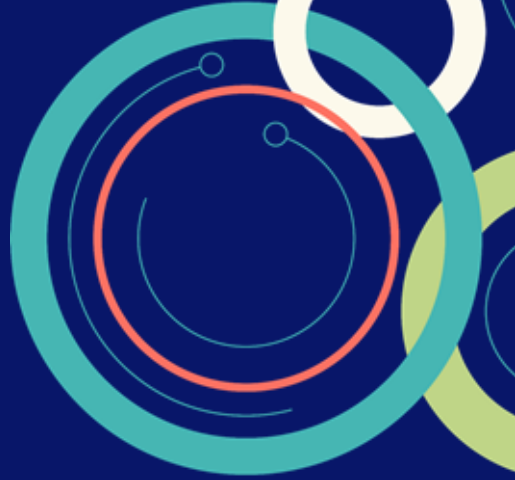


Observability Debt

When Telemetry Stops Telling the Truth

Spoorthi Palakshaiah

The Promise We Made To Ourselves

Dashboards

We'll always know what's happening in production

Alerts

We'll be first to know when something breaks

Metrics & SLOs

We'll measure what matters to users

Confidence

Observability gives us the power to ask any question

What is Observability Debt?

Not missing telemetry.

Telemetry whose **meaning has drifted** because the system changed but the observability artifacts didn't.

Dashboards

appear healthy

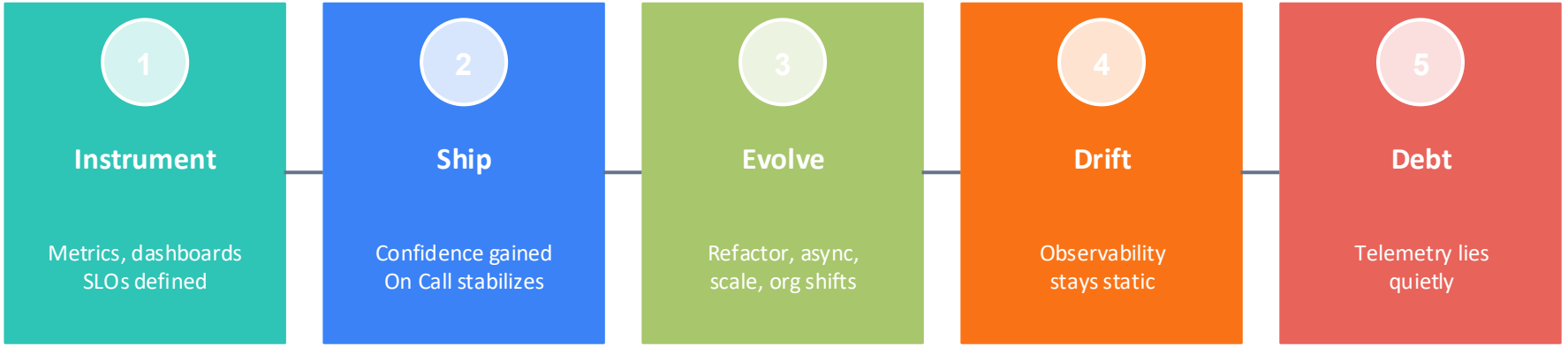
Alerts

lack context

Incidents

take longer to resolve

How Debt Accumulates



Nobody deletes the old dashboard.

Nobody invalidates the old SLO.
Nobody files a ticket called "update metrics for new arch."

Common triggers

- Sync → async pipeline added
- New caching/LB layer
- Service split to microservices
- Team ownership changes

A Minimal System: Low-Debt State

`http_request_duration → measures complete user wait time ✓`

1 metric = 1 journey

No hidden async work

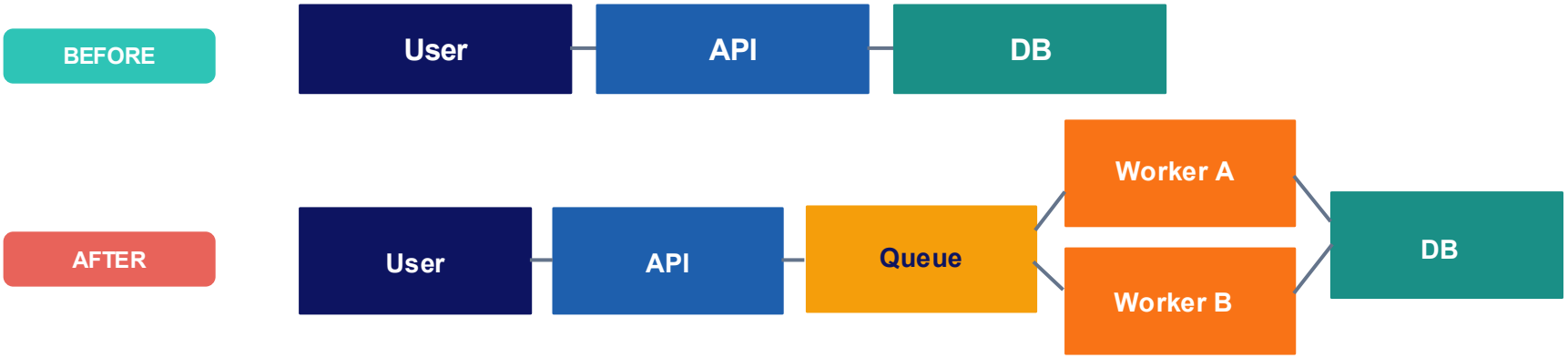
p99 = real user wait

Direct causal link

Error rate = user failures

No silent background jobs

The System Evolves. The Telemetry Doesn't.



What the old metric misses after this change:

- API returns 200 immediately, work happens asynchronously in workers
- p99 latency dropped 64% (looks like a win!) but user wait time actually increased
- Worker B failures are silent and never surface to the HTTP error rate metric

DASHBOARD SAYS

All Green ✓

API Latency p99

89ms

▼ *64% vs last week*

HTTP Error Rate

0.01%

Within SLO ✓

SLO Compliance

99.97%

Budget healthy ✓

REALITY

Users Suffering

4-8s

User wait time

Queue backlog not instrumented

15%

Jobs silently failing

Worker B errors never surface

SLO is a lie

Measures API, not job completion

Patterns of Telemetry Drift

1 The Aggregation Lie

Averaging across instances hides the one node failing. p99 stays green; 20% of users hit errors.

2 The Sync-to-Async Illusion

API latency drops after async refactor, SLO improves. But user-perceived time now includes queue wait.

3 The Cached Success

Cache hit rate looks great. The miss path is 10× slower with its own failure mode never instrumented.

4 Ownership Amnesia

The team that built the dashboards left. New team trusts them. Thresholds tuned for v1 traffic patterns.

The Cost: Slower Incident Understanding

3×

longer MTTD

Mean time to detect

40%

of incidents

start with a misleading signal

Hours

lost debugging

telemetry vs. reality mismatch

A Typical Incident with High Observability Debt

T+0	User reports slowness
T+20m	Dashboard shows green, dismissed
T+45m	Dig deeper, discover queue backlog
T+90m	Root cause: Worker B silent errors
T+2h	Resolved. Dashboards still wrong.

Paying Down Observability Debt

Detect

- Run telemetry archaeology after any architectural change
- Ask: does this metric still mean what we think it means?
- Diff dashboards against your current system diagram

Prevent

- Treat observability as part of the definition of done
- Require telemetry review in architecture RFC process
- Add SLI/SLO validity check to post-incident reviews

Fix

- Instrument the user journey, not just service boundaries
- Trace async work end-to-end with correlation IDs
- Version your SLOs alongside your architecture

Key Takeaways

01

Observability debt is silent

Dashboards don't tell you when they stop being relevant. You have to check.

02

The problem is meaning, not data

You have all the metrics you need. You don't know what they're measuring anymore.

03

Architecture changes must trigger telemetry reviews

Every async refactor, every new layer, every team handoff is a debt creation event.

04

Instrument user journeys, not service boundaries

Start from what the user experiences. Work backwards to what you measure.

Observability Summit North America

Telemetry that once told the truth can quietly stop doing so.

The question isn't whether you have observability.
It's whether you still trust it and whether you should.

After this talk: audit one dashboard.

Does it still mean what you think it means?

Observability Summit North America

Thank you!

www.linkedin.com/in/spoorthi-p/