KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

OTel Blueprints

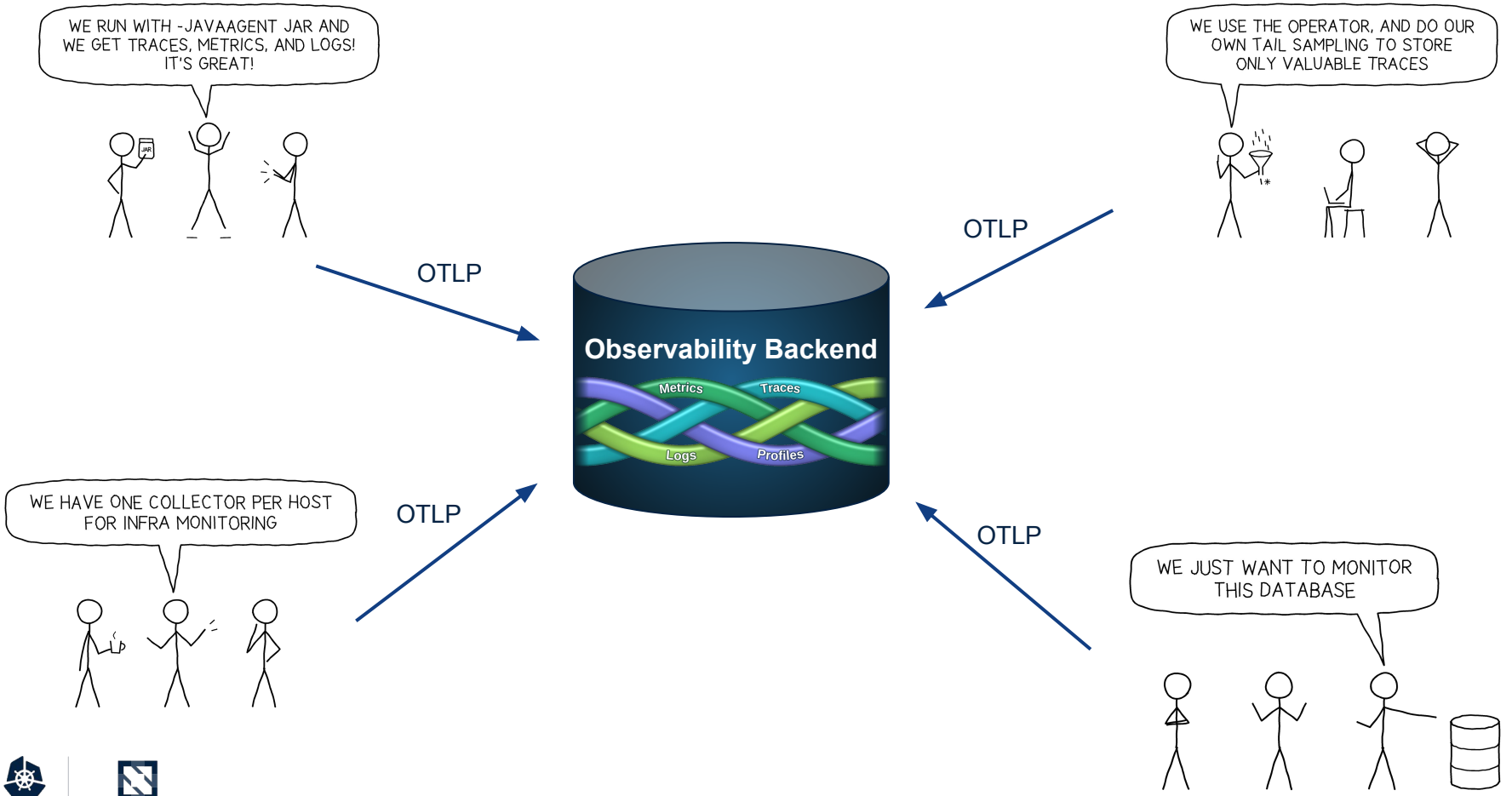
Prescriptive Patterns for Enterprise Observability

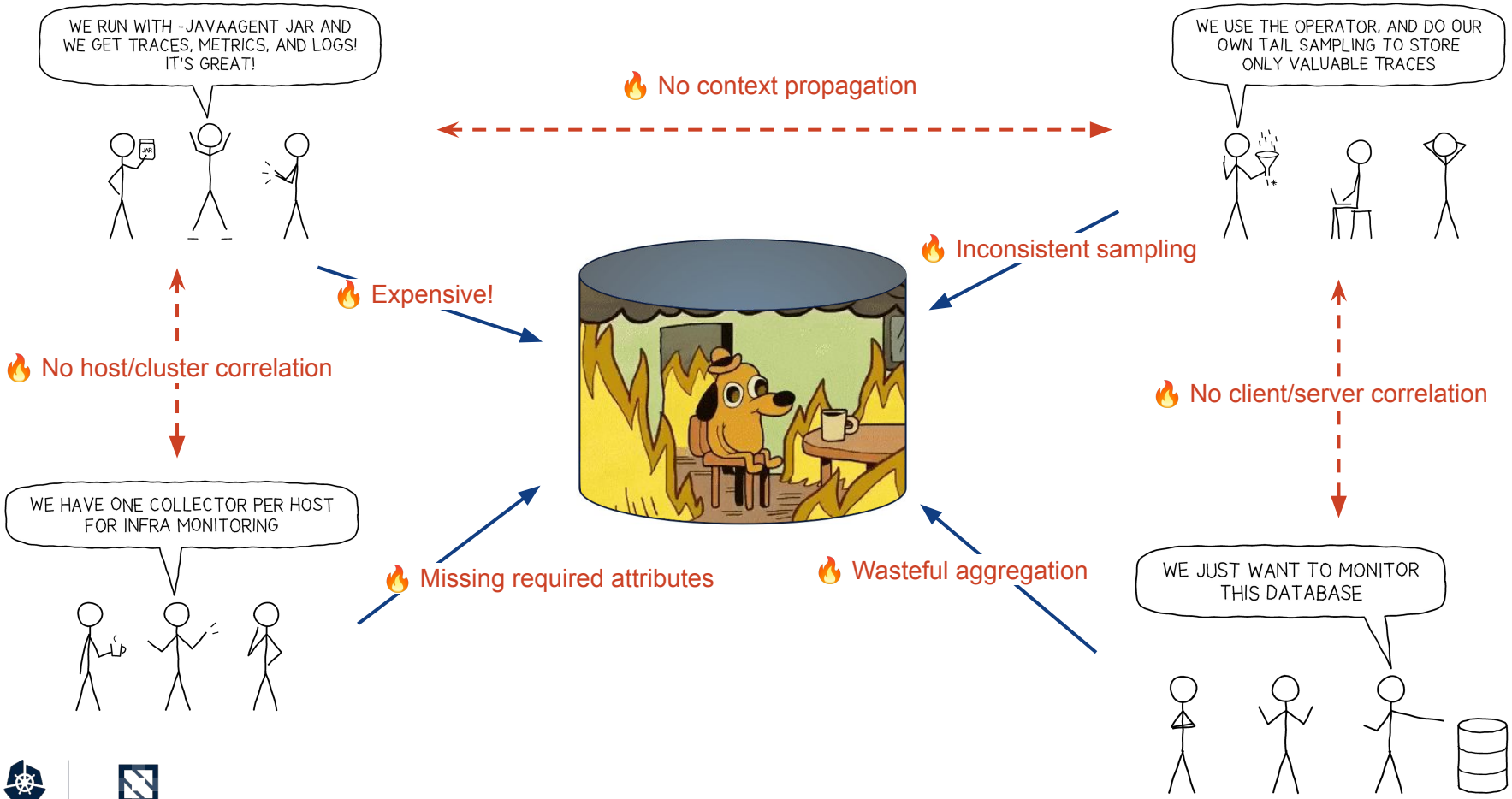
Dan Gomez Blanco
Principal Observability Architect (New Relic)

Lukasz Ciukaj
Observability Solutions Architect (Splunk by Cisco)




KubeCon
CloudNativeCon
North America 2026
“We’re adopting OpenTelemetry”
“Awesome! But what does that mean?”


WE RUN WITH -JAVAAGENT JAR AND WE GET TRACES, METRICS, AND LOGS! IT'S GREAT!
JAR
WE USE THE OPERATOR, AND DO OUR OWN TAIL SAMPLING TO STORE ONLY VALUABLE TRACES
OTLP
OTLP
Observability Backend
Metrics
Traces
Logs
Profiles
WE HAVE ONE COLLECTOR PER HOST FOR INFRA MONITORING
OTLP
OTLP
WE JUST WANT TO MONITOR THIS DATABASE


WE RUN WITH -JAVAAGENT JAR AND WE GET TRACES, METRICS, AND LOGS! IT'S GREAT!
WE USE THE OPERATOR, AND DO OUR OWN TAIL SAMPLING TO STORE ONLY VALUABLE TRACES
No context propagation
Expensive!
Inconsistent sampling
No host/cluster correlation
No client/server correlation
WE HAVE ONE COLLECTOR PER HOST FOR INFRA MONITORING
Missing required attributes
Wasteful aggregation
WE JUST WANT TO MONITOR THIS DATABASE

Conway's Law of OpenTelemetry

Organizations which ~~design systems~~ adopt OpenTelemetry are constrained to produce ~~designs~~ telemetry streams which are copies of the communication structures of these organizations.

—*Melvin E. Conway, How Do Committees Invent?* *(with some creative freedom)*

Who we are



Dan Gomez Blanco

Principal Observability Architect
New Relic

- Governance Committee Member
- End-User SIG Maintainer
- Author of Practical OpenTelemetry: Adopting Observability Standards Across Your Organization

Fun fact: Real passion is drumming 🥁🥁🥁

@danielgblanco linkedin.com/in/danielgblanco86/



Lukasz Ciukaj

Observability Solutions Architect
Splunk by Cisco

- Contributing to OpenTelemetry since 2022
- Blueprints Co-lead & Approver (End-User SIG)
- Founder & Lead, OpenTelemetry Industrial
- Founder & Co-organizer, Cloud Native RTP

Fun fact: Dad of two girls 🦄🌈

@lukaszciukaj · @luke6Lh43 linkedin.com/in/lciukaj

Agenda

KubeCon CloudNativeCon North America 2026

01 **How we got here**
Why the project was born

02 **What is a blueprint?**
Strategy, not docs

03 **Current blueprints**
Published guidance

04 **Reference implementations**
Self-serve, real-world evidence

05 **Getting involved**
Submit proposals and contribute your story

06 **Q&A**
Your questions, our answers

01

How we got here

Why the project was born,
and what we've shipped so far



End users kept asking for the same thing

> **End users want a prescriptive, opinionated way of "deploying OpenTelemetry".**
>
> *OTel Blueprints announcement, opentelemetry.io blog (May 2026)*

What we heard

- Why does adoption feel so complex?
- Which Collector topology should we run?
- How do get dozens of teams to stay consistent?
- How do I justify it to leadership? What's the benefit?

End-User SIG + Developer Experience SIG

Two kinds of complexity

KubeCon CloudNativeCon North America 2026

Essential complexity

Inherent, it can't be removed

- OpenTelemetry's breadth: every signal, language and layer of the stack
- Cross-cutting by nature, it touches every team and component
- A large configuration surface, all of it is needed

Accidental complexity

Organizational, it can be reduced

- Teams adopting independently, without a shared strategy
- Inconsistent SDK and Collector configuration
- Broken context propagation and lack of semconv → disjointed telemetry

There's no silver bullet*, blueprints target accidental complexity.

*Inspired by "No Silver Bullet Essence and Accidents of Software Engineering", Fred Brooks

It takes a village (or a community)

From listening to end users to published guidance



Listen

2024 – 2025

Feedback via the End-User SIG and hallway track conversations

Bootstrap

2025 H2

Cross-cutting OpenTelemetry initiative backed by leadership

Define

2026 H1

Strategy, templates, and contribution process

Launch

May 2026

Intro blog post on opentelemetry.io

Publish

2026 Q2 – Q3

Initial blueprints & reference implementations

Grow

2026 Q4 +

Wrap up bootstrap initiative and scale contributions



02

What is a blueprint?

Strategic thinking, not “new documentation”

A blueprint, defined



Living documents that solve common adoption and implementation challenges in a given environment.

opentelemetry.io/docs/guidance



Strategic

Ties components, solutions and best practices into one holistic approach.



Scoped

Picks the most critical challenges in an environment, and solves those alone.



Opinionated

Recommends a path. Not a menu of every possible configuration.

Anatomy of a blueprint

1

Summary

Target audience and the environments it applies to

2

Common challenges

The problems in scope, and explicitly what's out

3

General guidelines

Design patterns and best practices, with architecture diagrams

4

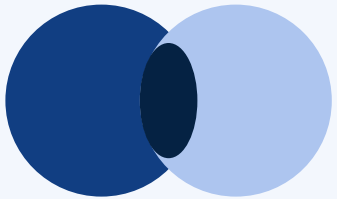
Implementation

Actionable steps, linking to existing docs

Same structure* every time → easy to compare, review and keep current

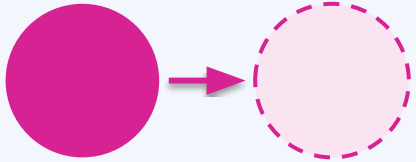
*Inspired by "Good Strategy Bad Strategy: The Difference and Why It Matters", Richard Rumelt

Blueprints build on each other



Overlap

Share identical design patterns, like a Collector DaemonSet deployment.



Extension

One blueprint solves problems another declared out of scope.

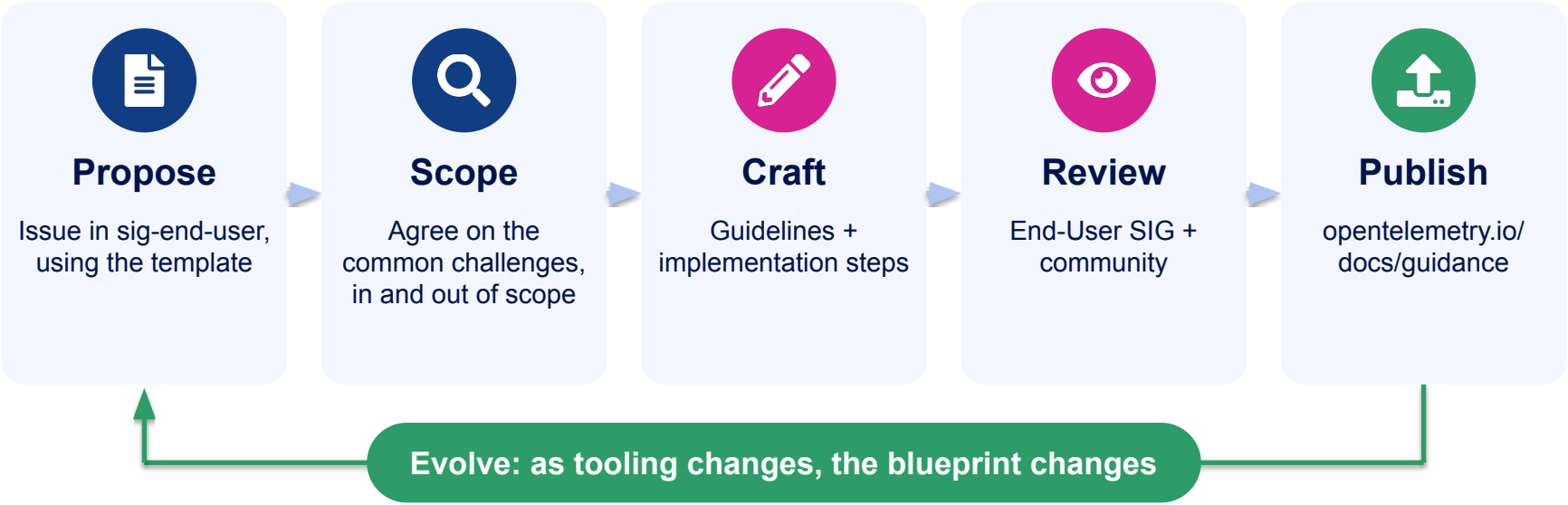


Relation

One blueprint assumes the patterns of another are already in place.

Need more context? Every blueprint links to related blueprints and docs.

From proposal to living guidance



Proposals are approved in sig-end-user before a PR is opened on opentelemetry.io



03

Current blueprints

Three environments, one consistent strategy



Blueprints today



Published

Managed telemetry platforms for Kubernetes workloads

Platform teams offering telemetry-as-a-service



Published

Infrastructure & processes in non-K8s environments

VMs, bare metal, on-prem, standalone containers



Published

Kubernetes cluster observability

Workloads + critical cluster components

Infrastructure & processes in non-K8s environments

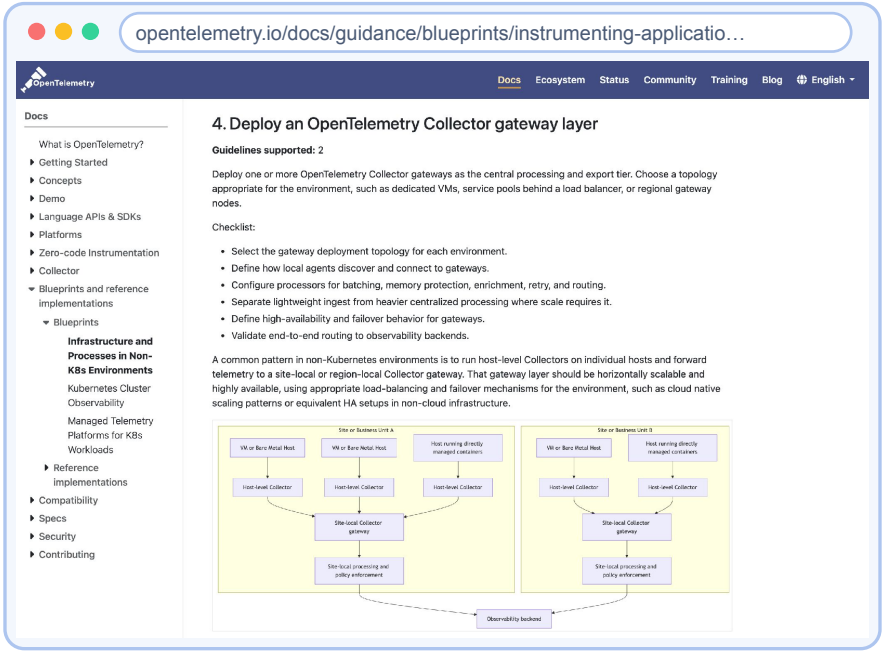
Published *For platform & SRE teams running VMs, bare metal, on-prem and standalone containers*

What it solves

- Manual, host by host agent rollout
- Fragmented instrumentation
- Siloed processing and export

What we recommend

- Manage agents centrally
- Add a Collector gateway layer
- Standardize resource attributes



Managed telemetry platforms for Kubernetes workloads

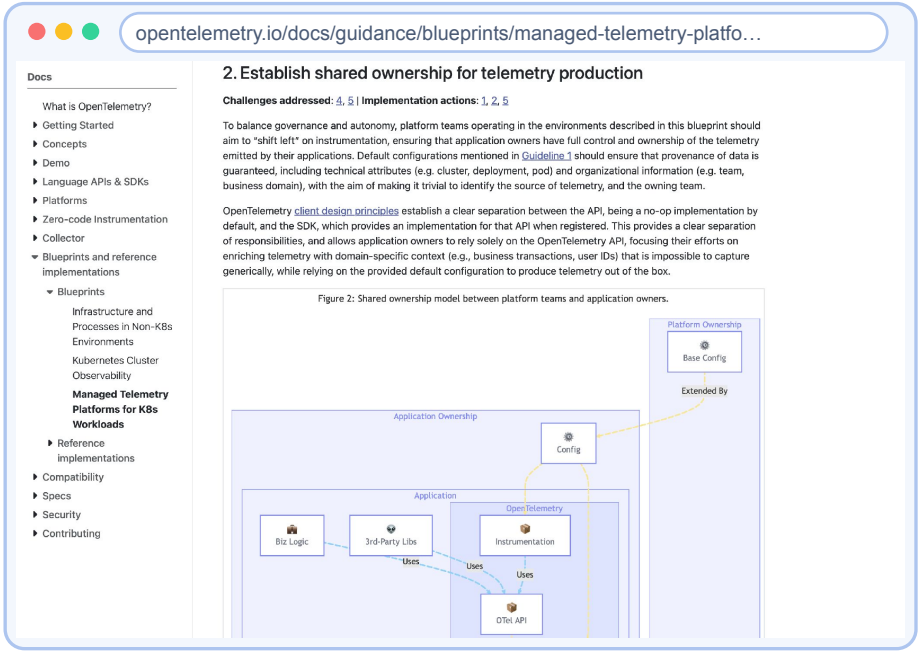
Published *For platform teams offering telemetry as a service to product teams*

What it solves

- Inconsistent SDK configuration
- Collector config sprawl
- Unoptimized, costly pipelines

What we recommend

- Central defaults teams can extend
- Shared telemetry ownership
- Managed gateways + tail sampling



Kubernetes observability

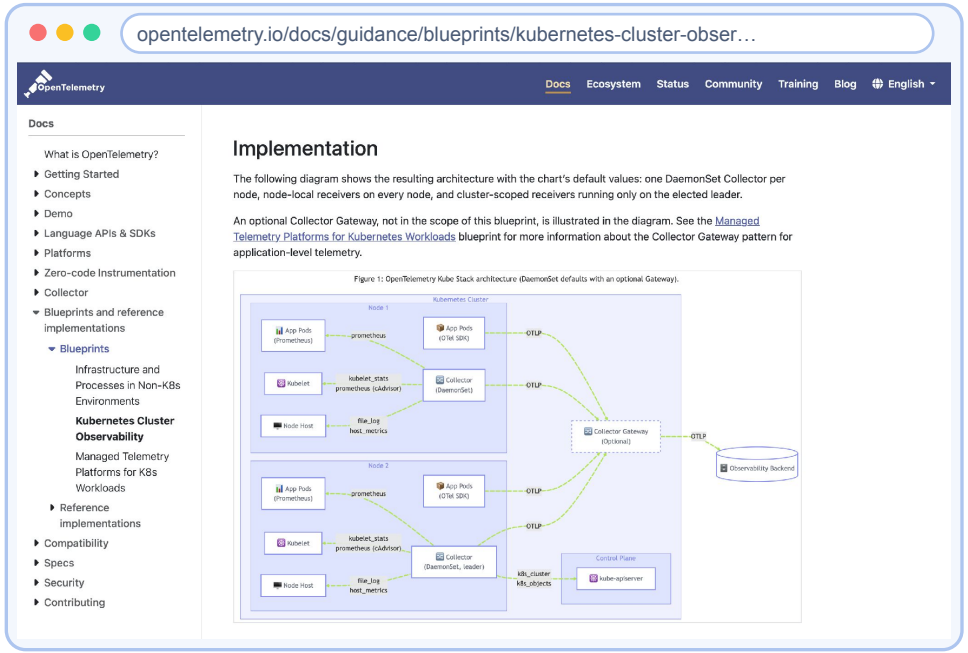
Published *For teams observing workloads and critical cluster components*

What it solves

- Missing OOM, throttling, probe signals
- Inconsistent Kubernetes metadata
- Blind spots: CoreDNS, CNI, ingress

What we recommend

- OTel-native Kubernetes receivers
- Semantic conventions everywhere
- Operator + node-local scraping



Patterns that keep showing up

KubeCon CloudNativeCon North America 2026



Paved road + escape hatches

Central defaults that teams can extend



Agent → gateway

Collect locally, process centrally



One resource identity

Resource attributes & semantic conventions everywhere



Config as a managed artifact

Operator & Helm on K8s; OpAMP elsewhere



Observe the observer

SDK & Collector self-telemetry

Same ideas in every blueprint: that's the "overlap" relationship



04

Reference implementations

Self-serve, real-world evidence



What is a reference implementation?



Snapshots in time that show how real-world organizations use OpenTelemetry.

	Blueprint	Reference implementation
Nature	Living document, maintained	Snapshot in time
Voice	Prescriptive (“here’s what to do”)	Descriptive (“here’s what they did”)
Scope	An environment and set of challenges	One organization
Role	The recommendation to follow	The evidence it works

mastodon

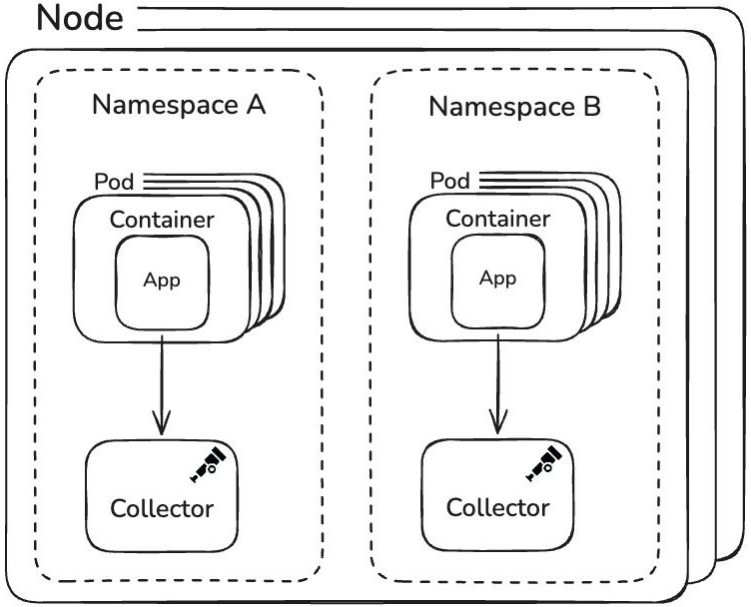
1

Collector per namespace

- ~10M requests/min on mastodon.social
- One Collector per namespace with Operator + Argo
- Tail sampling: ~0.1% of successful, 100% of errors

> ***Keep it simple. Rely on Operators. Upgrade frequently.***

↔ Centrally managed Collectors



Adobe

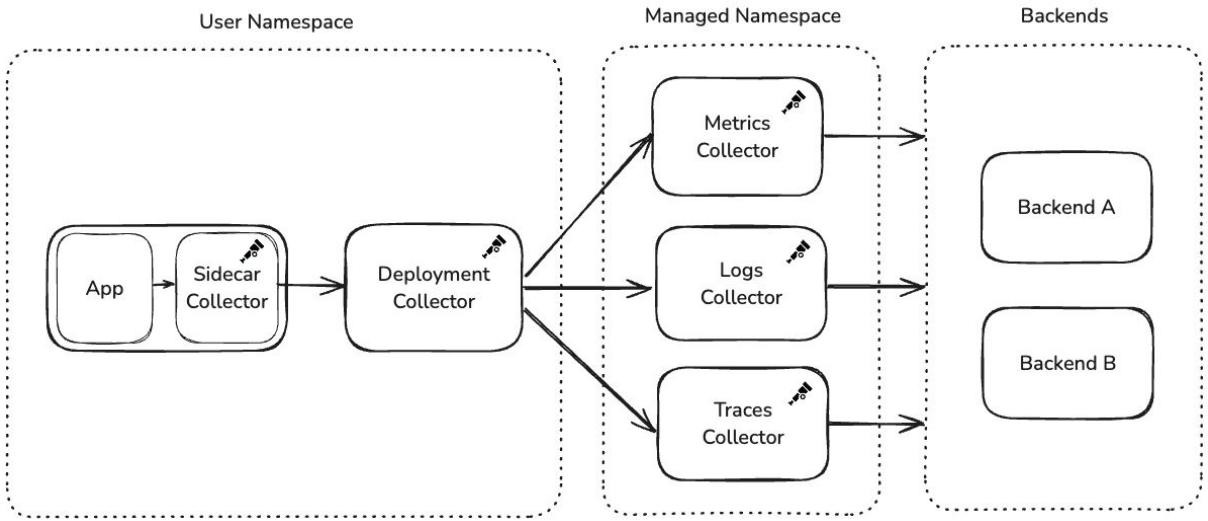
1000+

of Collectors per signal

- 3-tier topology: user Helm chart, managed per-signal Collectors, observability backends
- Operator auto-instrumentation via pod/namespace annotations
- Minimal custom distro with circuit-breaker extension

> ***Treat OpenTelemetry as a platform to build on. Design for user simplicity.***

↔ **Centralized, extensible defaults**



Skyscanner

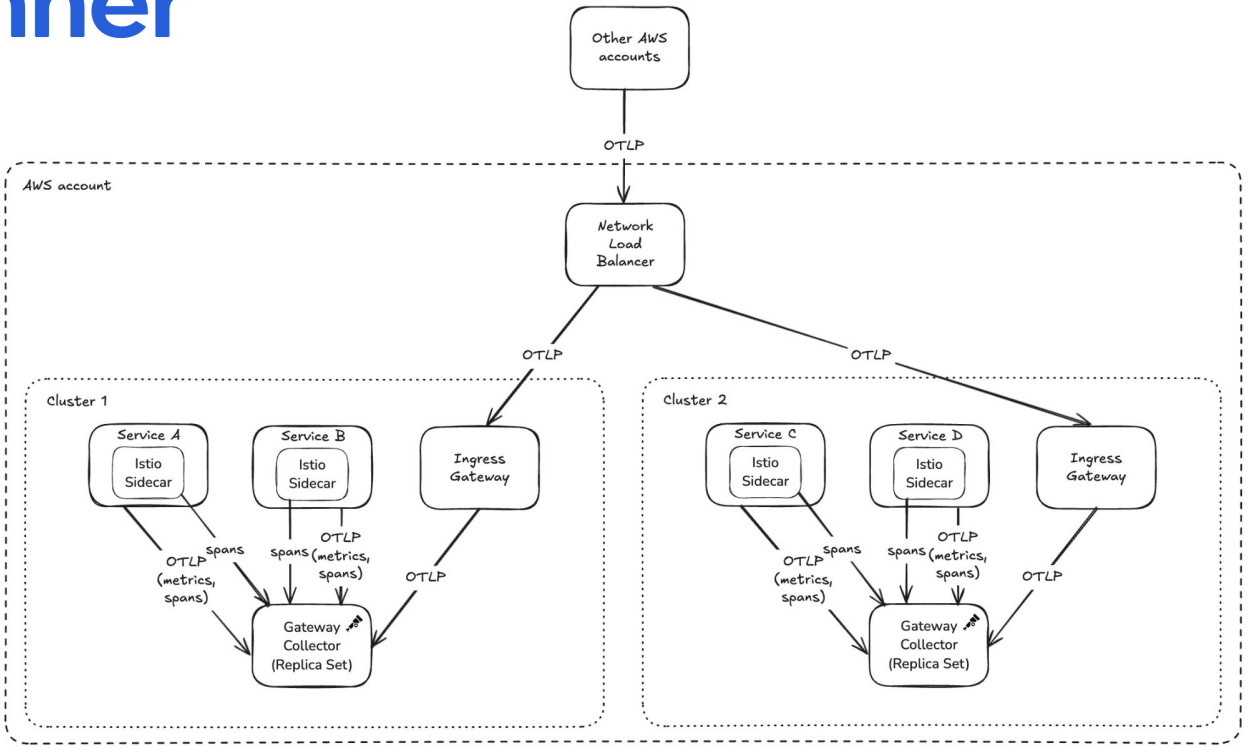
24+

clusters in a service mesh

- 1,000+ microservices, 6-person platform team
- Gateways + DaemonSet agents, Istio routing + span metrics
- Opinionated auto-instrumentation, progressive rollouts

> **Start simple. Filter early. Don't over-engineer telemetry resilience.**

↔ **Layer your aggregation & sampling**



KubeCon CloudNativeCon
North America 2026

05

Getting involved

Submit your story, or help shape the next blueprint



Two ways to get involved

For both blueprints and reference implementations

Propose
new ideas

1. Open an issue on the `sig-end-user` repo using our templates
2. Check out our guidelines to craft an excellent blueprint or reference implementation
3. The End User SIG is here to support you!

Contribute
on existing work

1. Browse open issues in `sig-end-user` repo
2. Pick one where help is needed
3. Write, review, and improve blueprints and reference implementations

Join us: #otel-blueprints on CNCF Slack

Get started today

KubeCon CloudNativeCon North America 2026

Check out current blueprints and reference implementations

opentelemetry.io/docs/guidance

Give us feedback about this talk on Sessionize!

KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Thank you!

Questions?

