



Golden Paths for Async Workflows: Dapr Meets OpenTelemetry

Mauricio “Salaboy” Salatino
Kasper Borg Nissen



KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA

Who?



Mauricio Salatino

Ecosystem Engineer &
Passionate about Open Source
Platform Engineering on Kubernetes
Author



Kasper Borg Nissen

Principal Developer Advocate at Dash0
Former KubeCon Co-Chair NA/EU
CNCF Ambassador
Golden Kubestronaut
CNCG Aarhus
Cloud Native Denmark
Cloud Native Nordics



TLTW (too long to watch)

If your applications and infrastructure grows in complexity you must have the right tools to understand what is going on at all times.





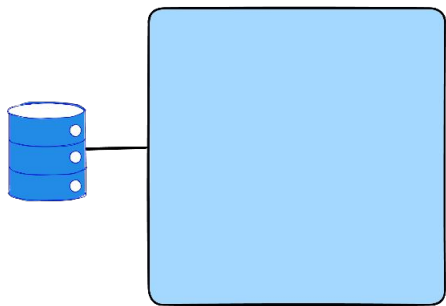
Why Async is Powerful



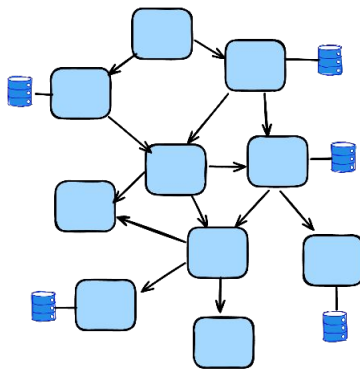
KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA



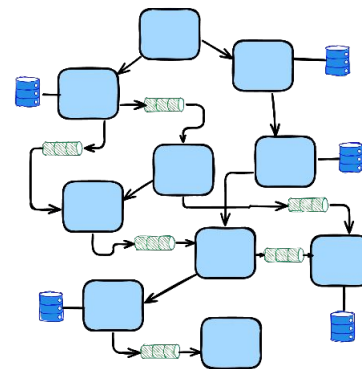
The Rise of Async Microservices



Monoliths



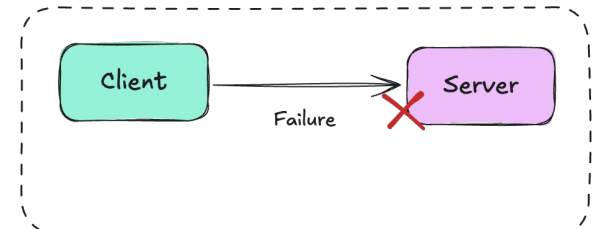
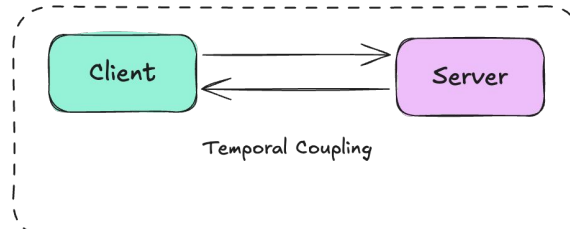
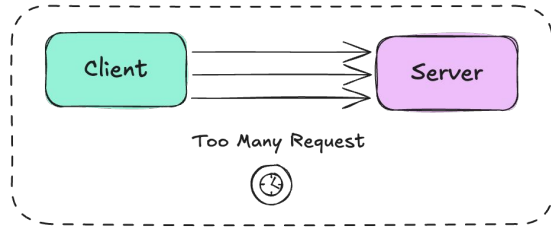
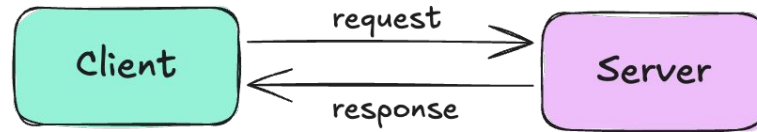
Microservices



Event-driven
Microservices

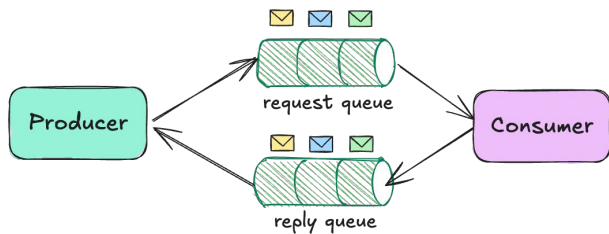


Limitations with synchronous communication

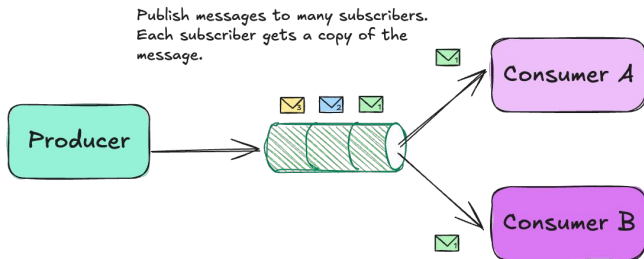


Async?

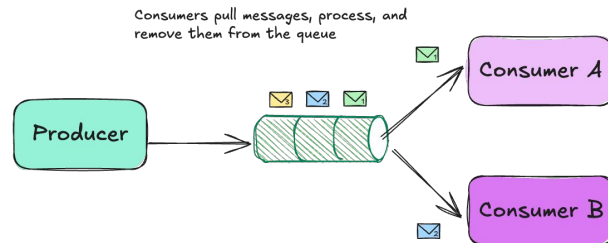
Request Reply Pattern



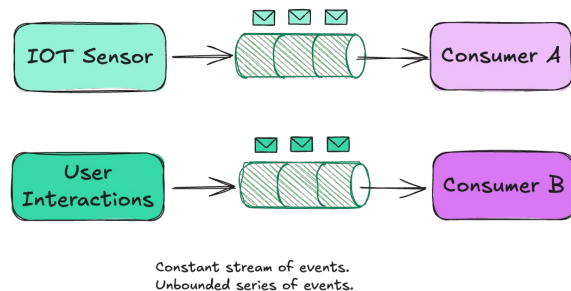
Publish/Subscribe



Queues



Streams



**Synchronous communication is familiar,
Asynchronous communication is powerful,
and in real systems you need both to work
seamlessly together.**





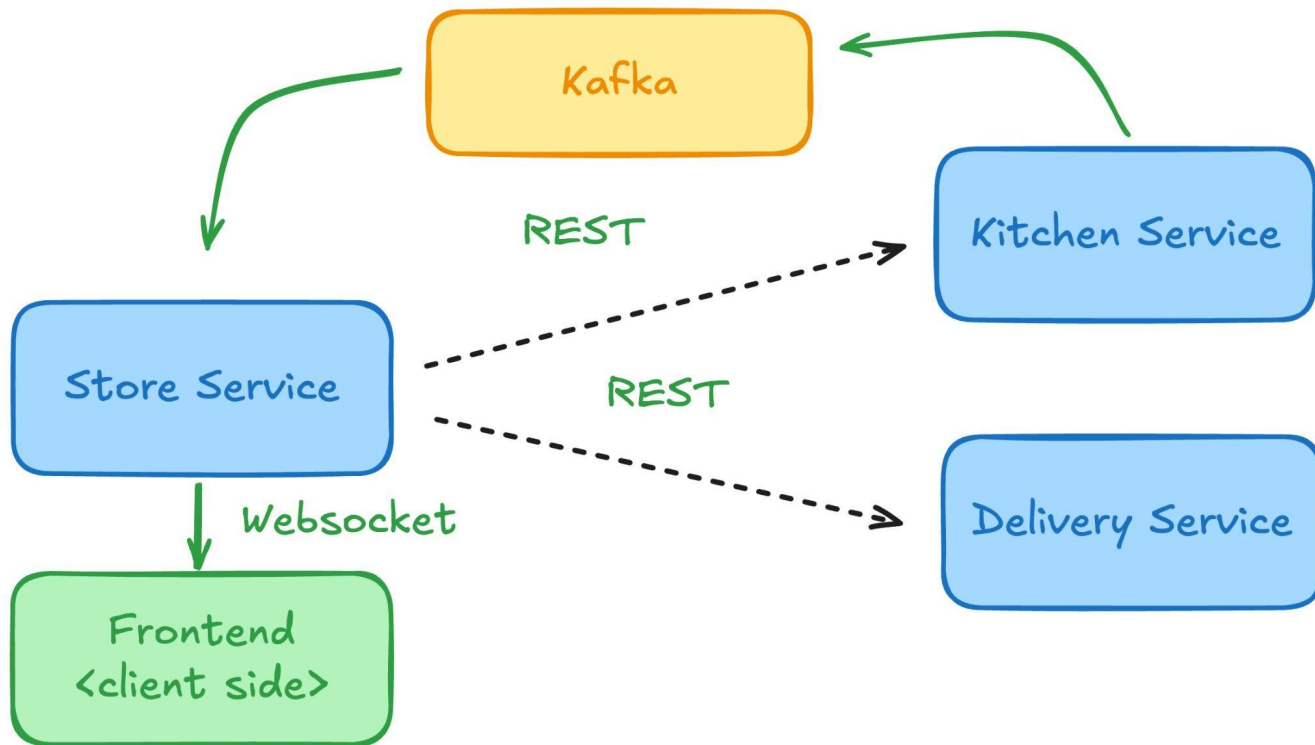
Demo #1



KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA



Recap





The Golden Path - Dapr + OpenTelemetry



KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA



The CNCF Opportunity: Shared Abstractions



dap^r



Dapr Building Blocks

- APIs to help developers build scalable and resilient distributed applications
- PubSub, Workflows, Secrets/Configs, Conversation (LLMs), etc
- All these APIs, behind the covers, implement cross-cutting concerns
 - Security
 - Resilience
 - **Observability**

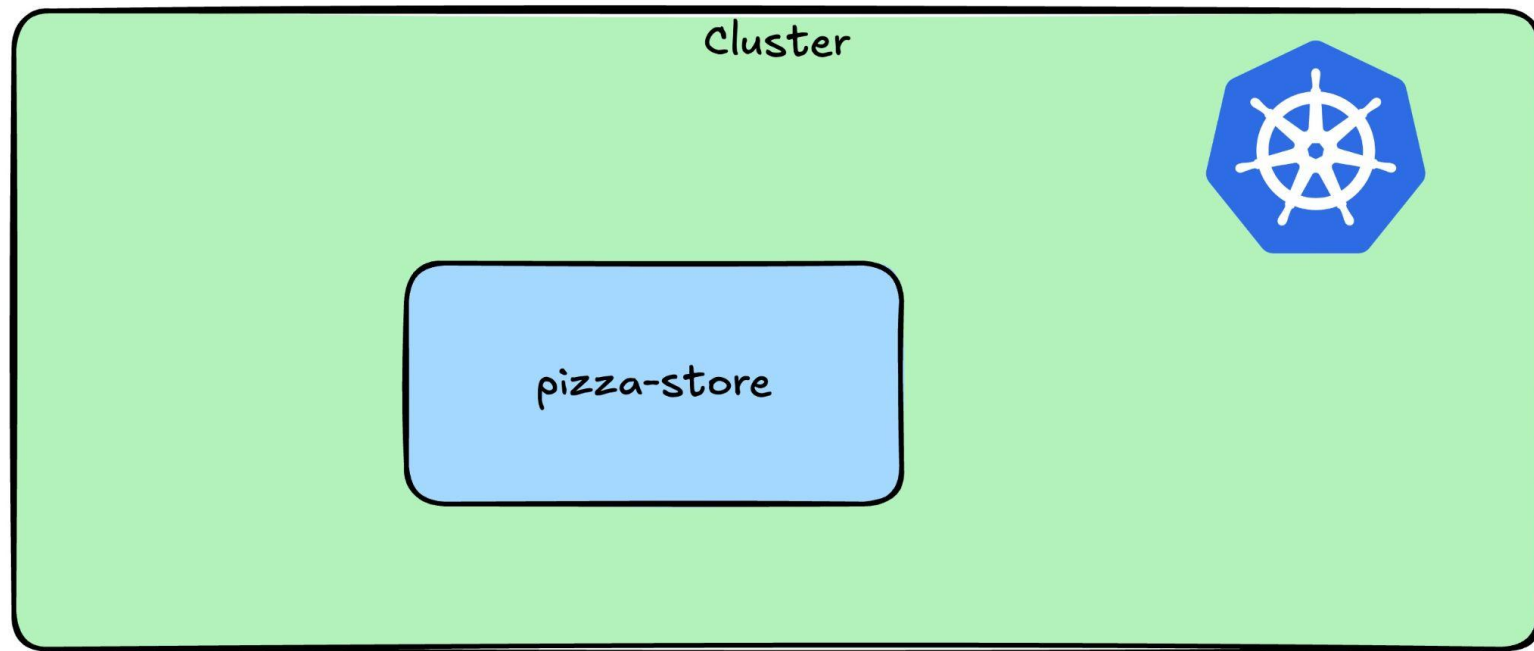


How does it work?

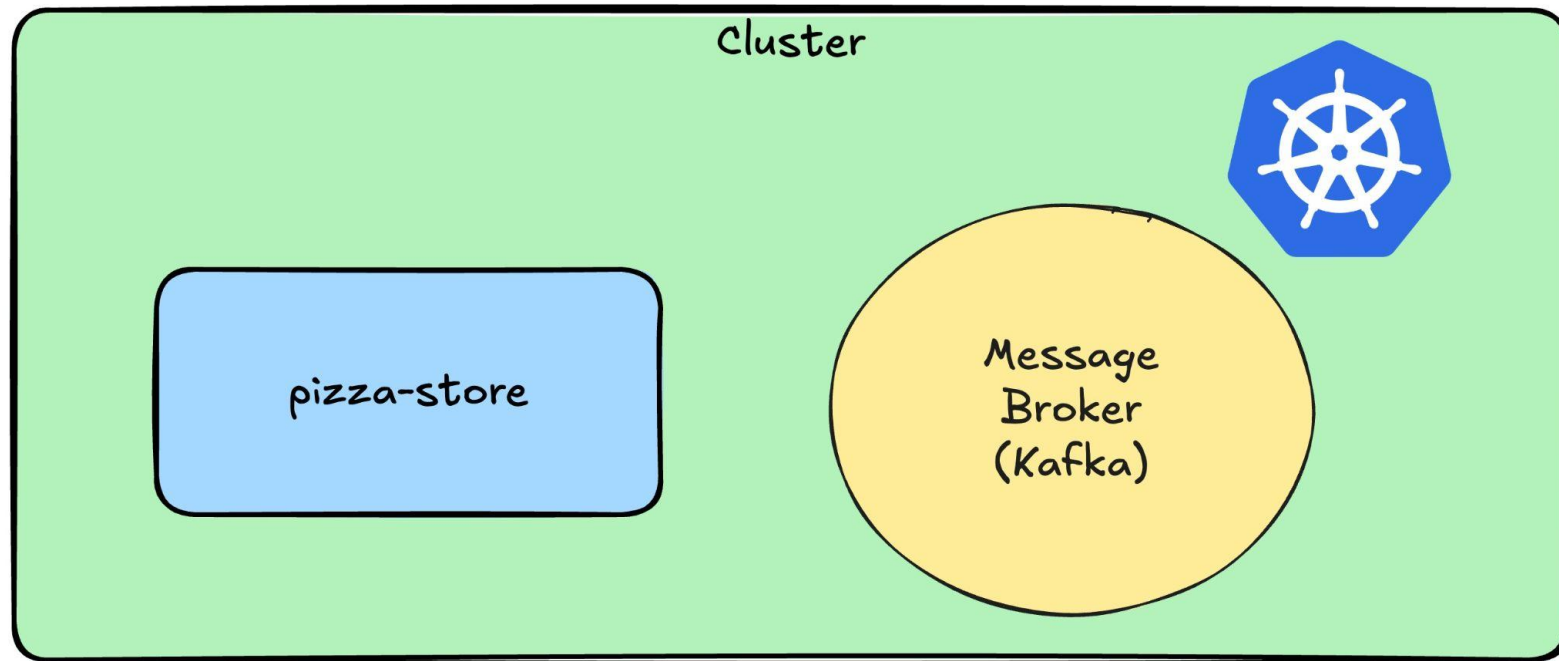
Cluster



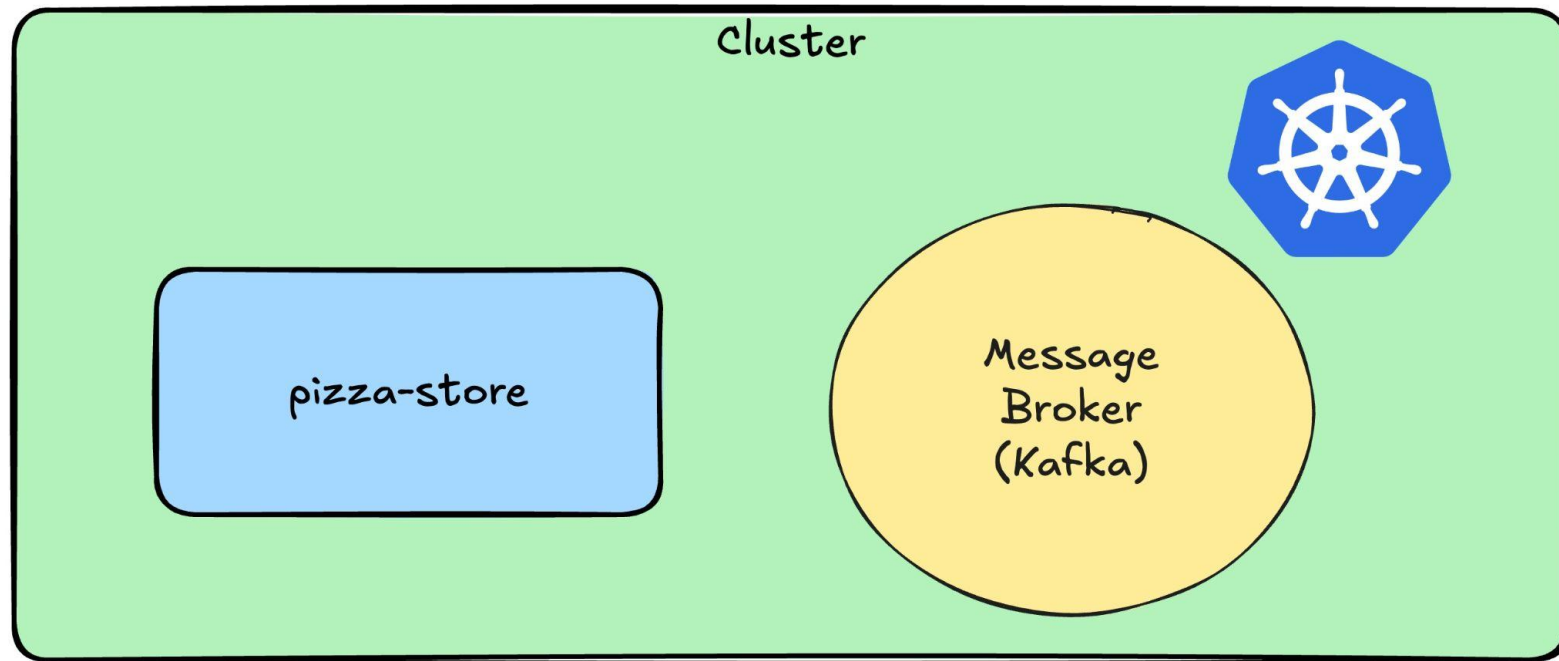
How does it work?



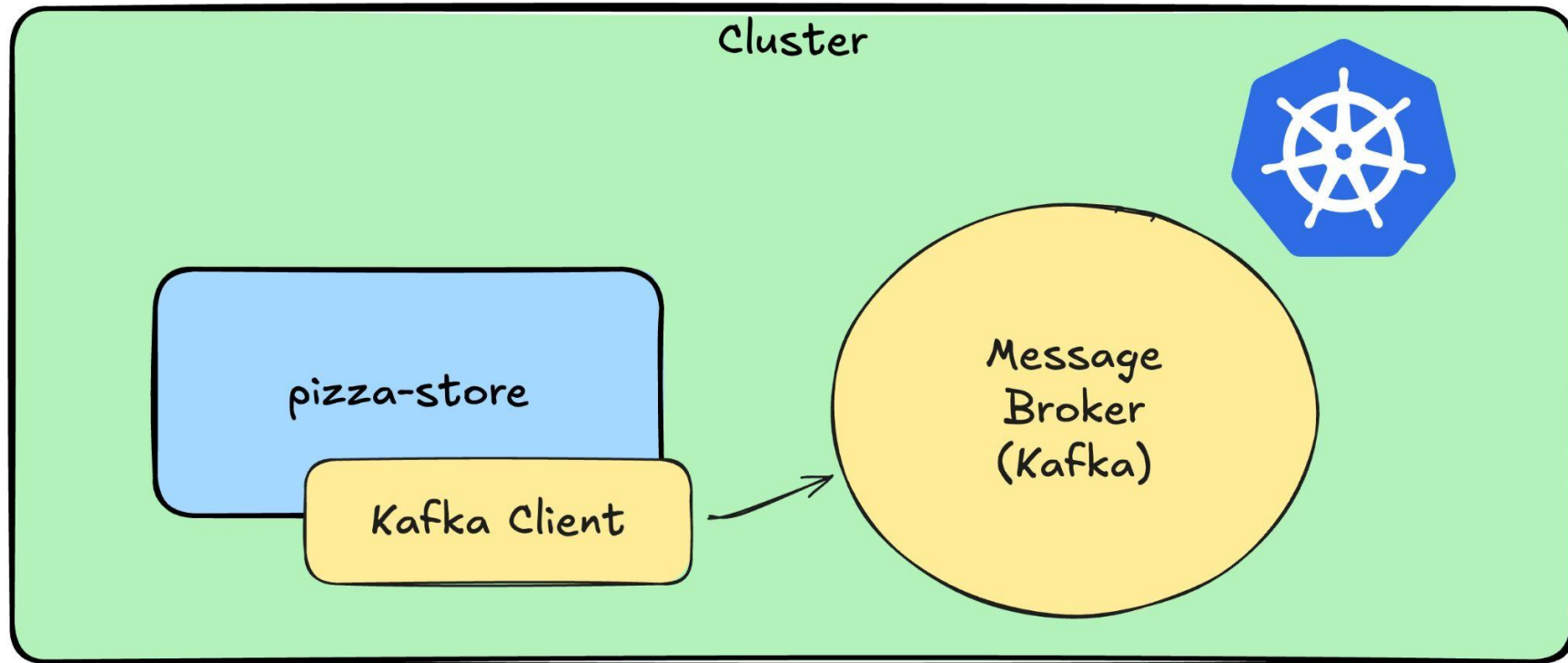
How does it work?



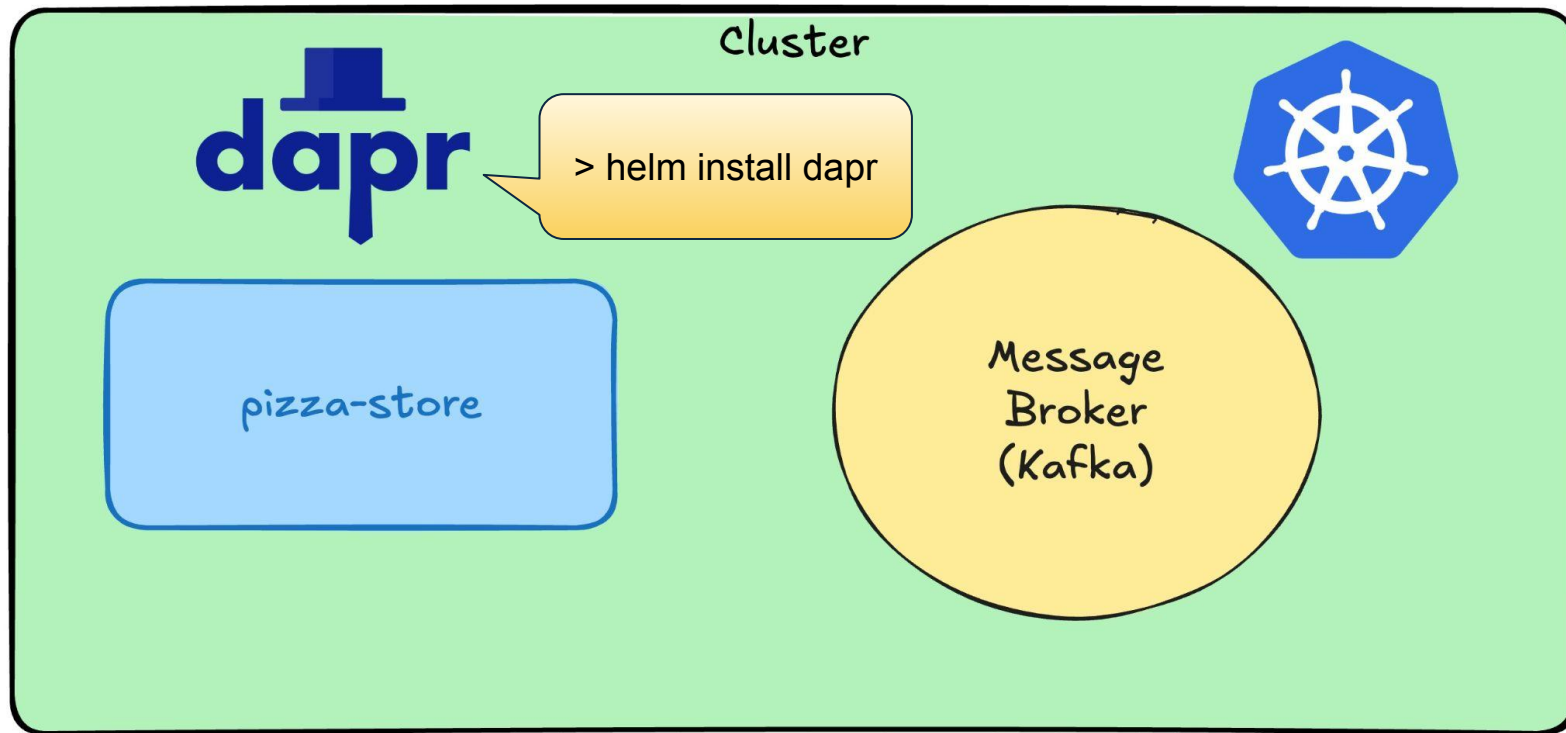
How does it work?



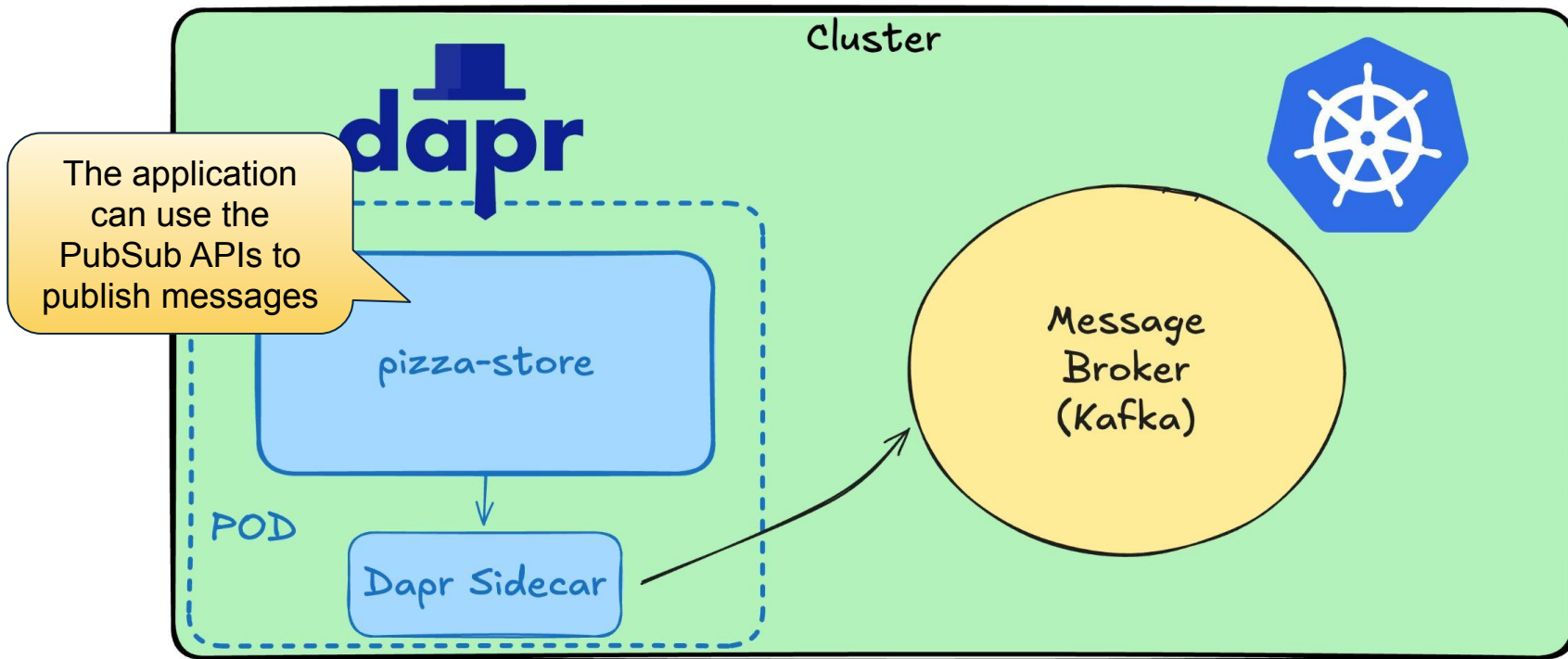
How does it work?



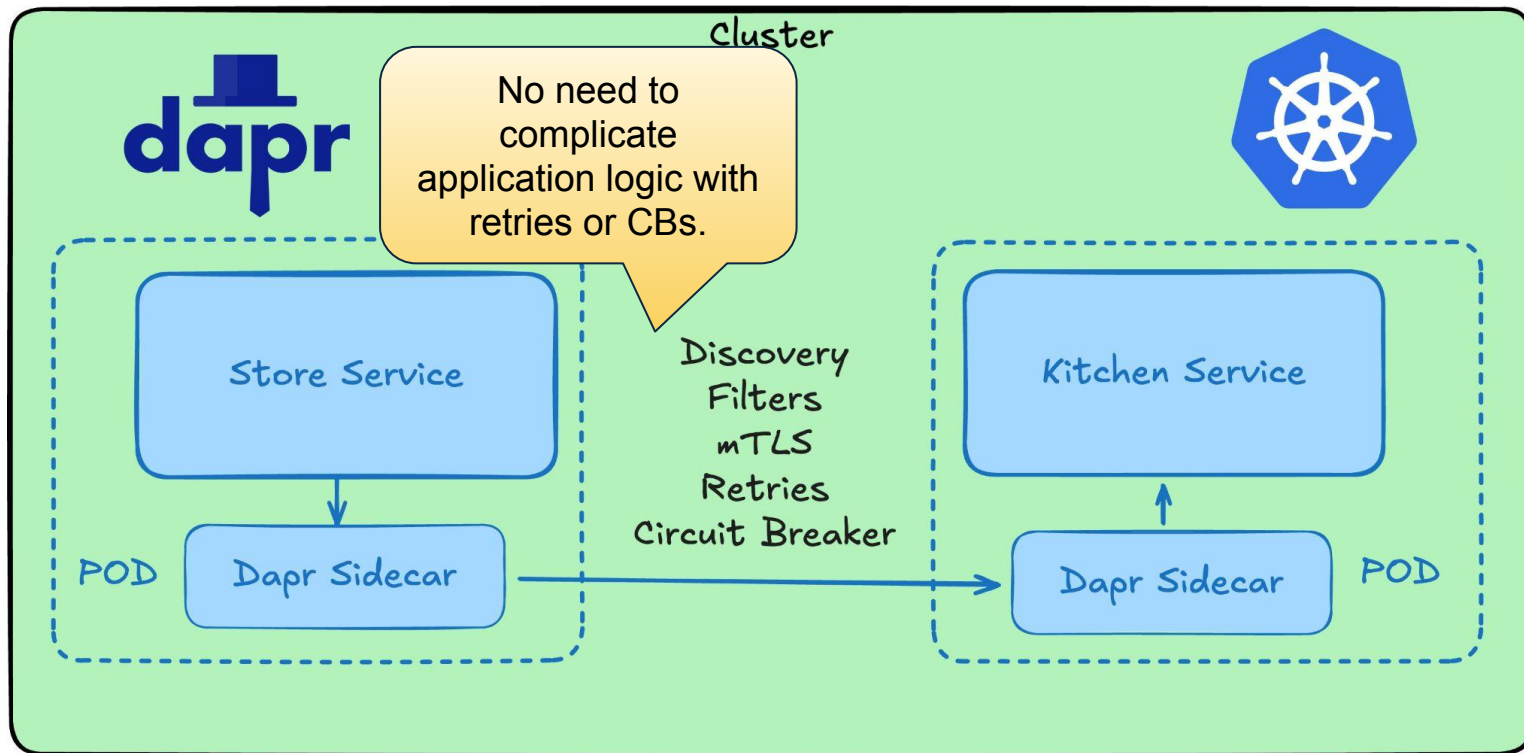
How does it work?



Sidecars to the rescue



Service To Service Invocation API



Ok, but what happen when things go wrong?

- What happens if the kitchen service is down and the retries are exhausted?
- What happens if Kafka is down?
- We cannot leave our pizza customers without their pizzas!

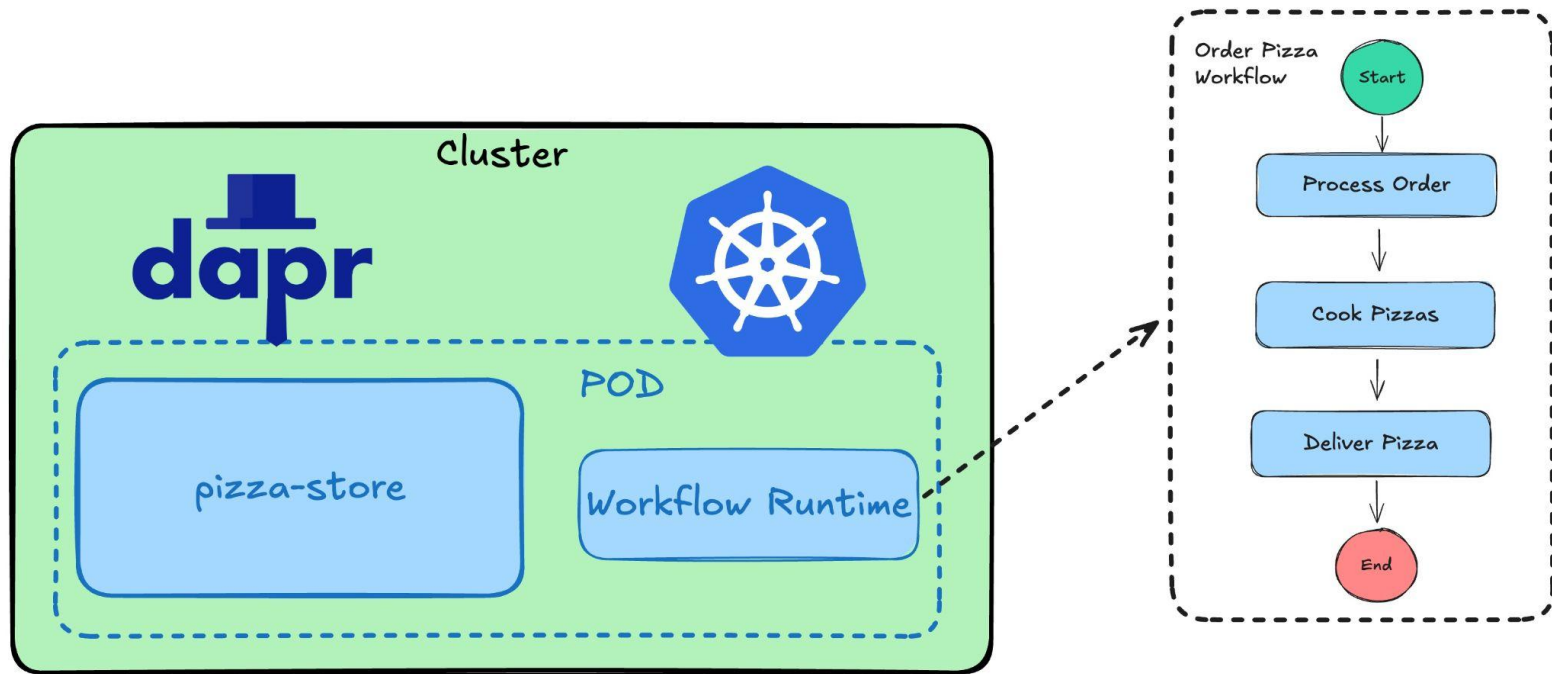


Dapr Workflows: Resilient orchestrations

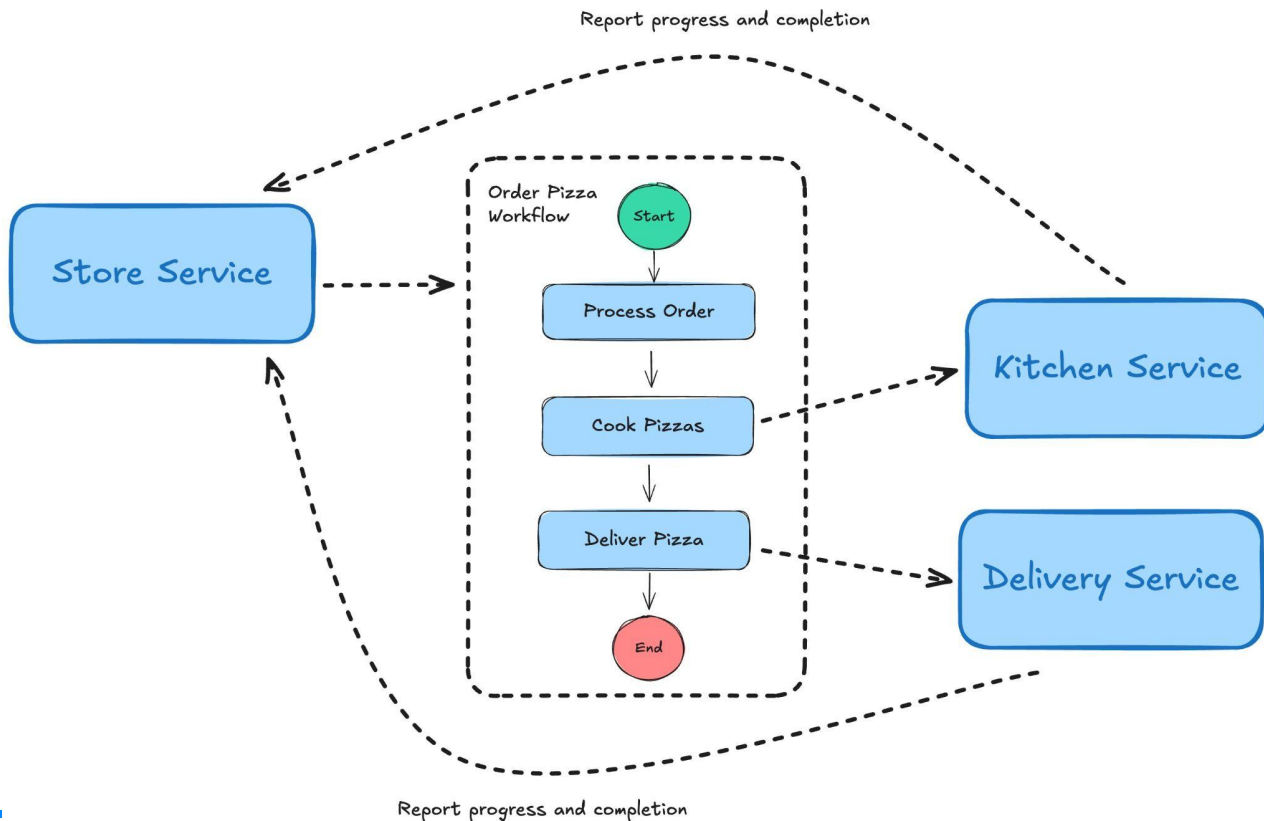
- Workflows are defined in code, executed by the Dapr sidecar
- Durable, long-running state management
- Retries, timers, wait-for-events all included
- No single point of failure or SaaS service needed
- Workflows will keep trying no matter what goes down! (even the workflow runtime!!!)



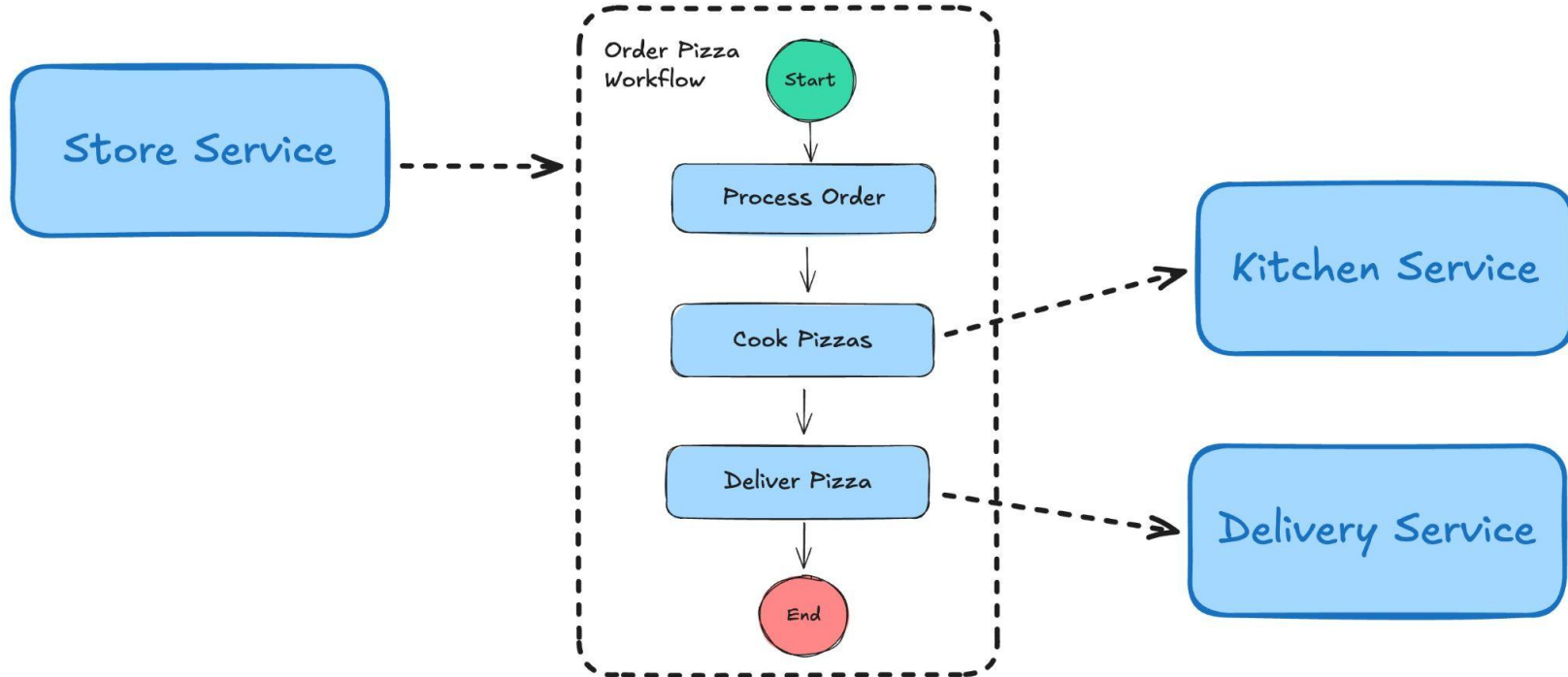
How does it work?



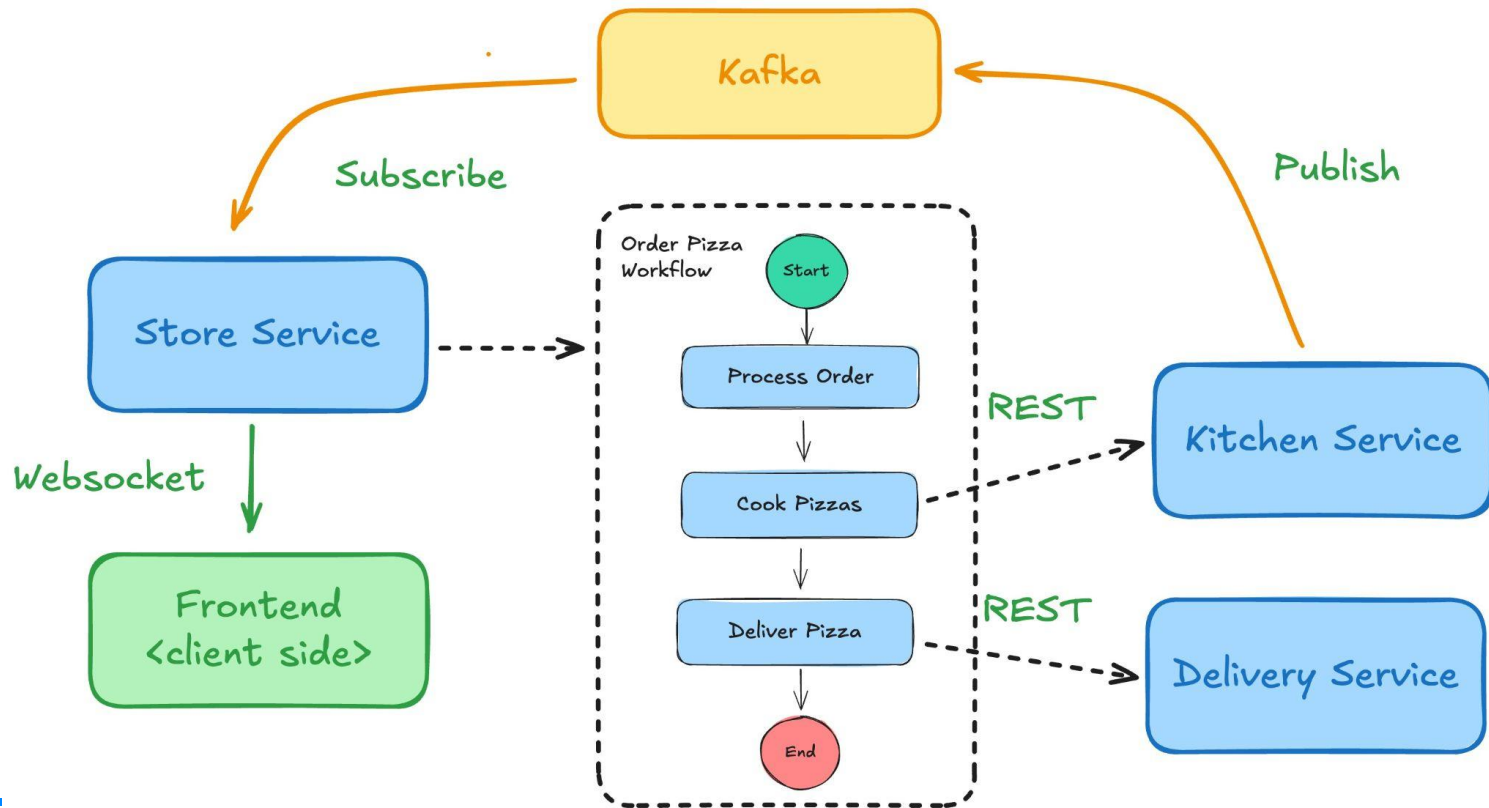
It is not that simple

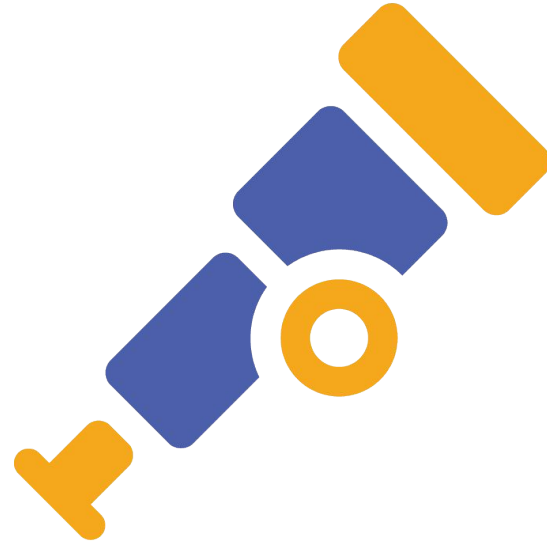


Pizza Orchestration



It is not that simple ++





OpenTelemetry



KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA



OpenTelemetry

OpenTelemetry (OTel) is an open source project designed to provide standardized tools and APIs for generating, collecting, and exporting telemetry data such as traces, metrics, and logs

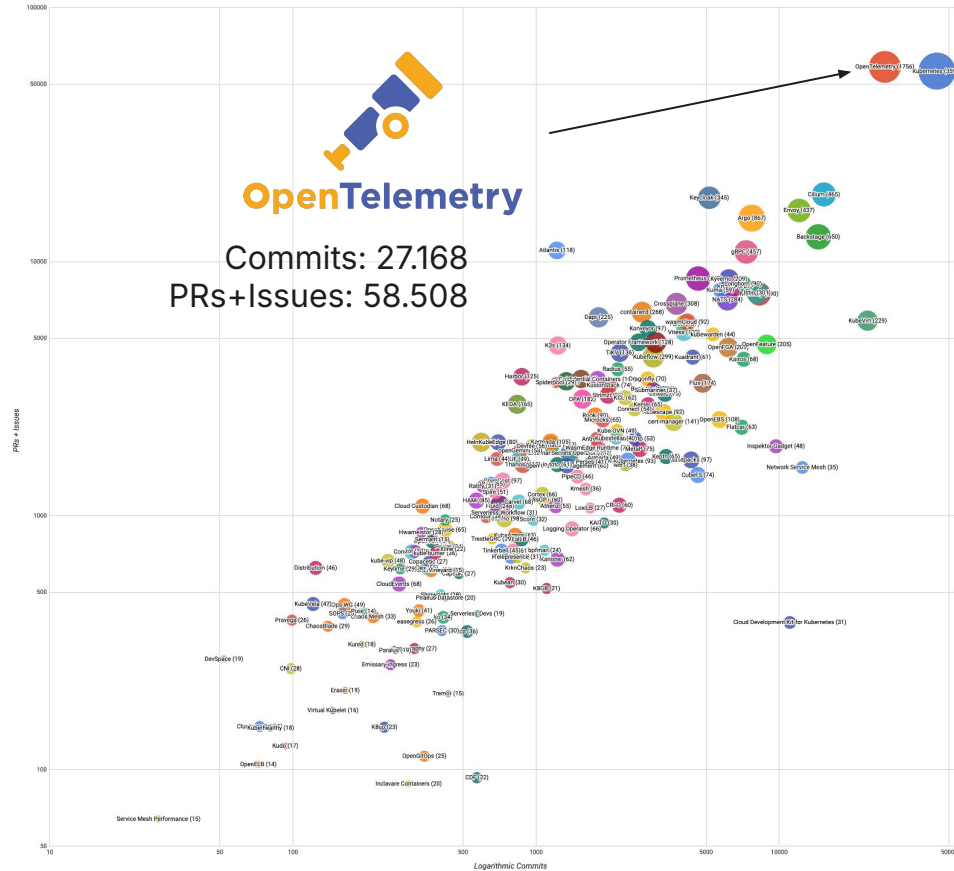
The de-facto standard for distributed tracing, supports metrics, logs, RUM, and profiling (experimental)

The main goals of the project are:

- Unified telemetry
- Vendor-neutrality
- Cross-platform



1/1/2024-1/1/2025



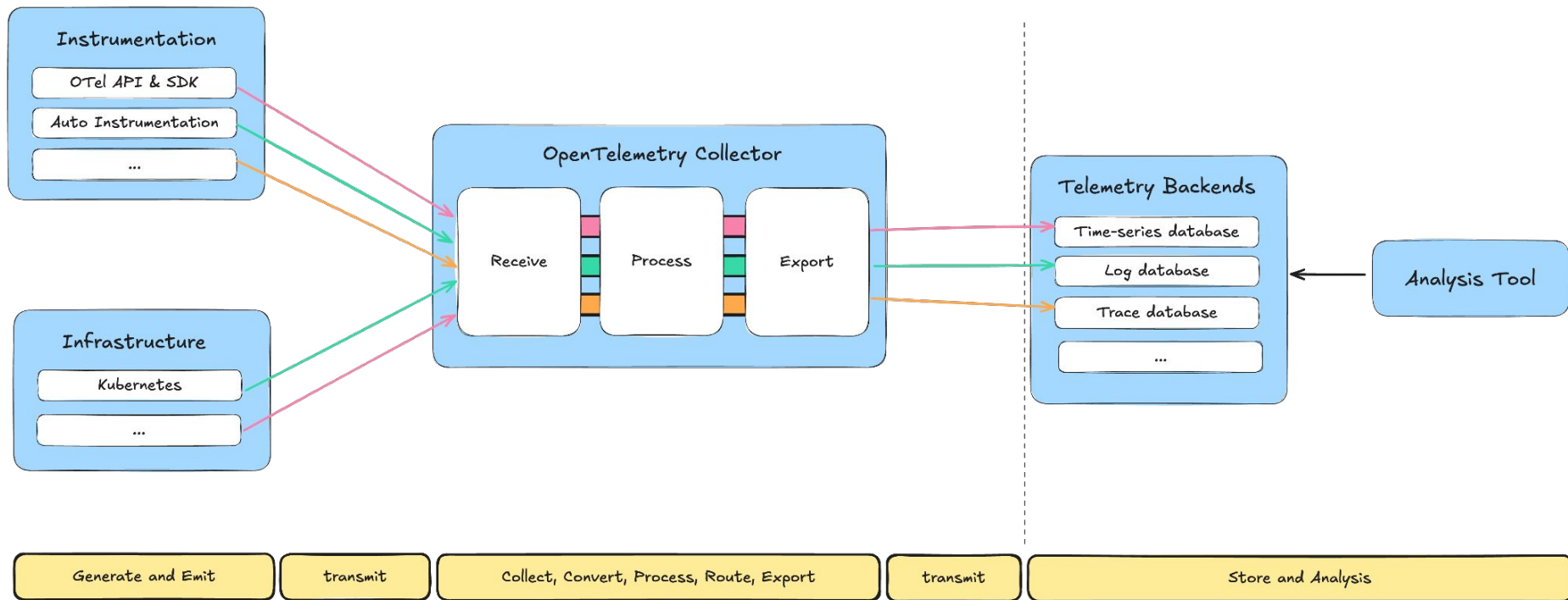
- Kubernetes (kubernetes.io) 3990 authors
- OpenTelemetry (opentelemetry.io) 1756 authors
- Argo (argoproj.github.io) 867 authors
- Backstage (backstage.io) 650 authors
- Prometheus (prometheus.io) 608 authors
- Charm (charm.io) 465 authors
- gRPC (grpc.io) 437 authors
- Envoy (www.envoyproxy.io) 437 authors
- Meshery (meshery.io) 390 authors
- Keycloak (keycloak.org) 345 authors
- Consul (consul.io) 338 authors
- Isito (isito.io) 301 authors
- KubeFlow (kubeflow.org) 299 authors
- NATS (nats.io) 284 authors
- container (container.io) 268 authors
- FluentD (fluentd.org) 248 authors
- Flask (flask.palletsproject.com) 248 authors
- KubeVirt (kubevirt.io) 229 authors
- Depr (depr.io) 225 authors
- Kyverno (kyverno.io) 209 authors
- Open GA (open.ga.dev) 208 authors
- OpenFeature (openfeature.dev) 205 authors
- Kusion (kusion.dev) 187 authors
- OPA (openpolicyagent.org) 182 authors
- Flux (fluxcd.io) 174 authors
- Falco (falco.org) 167 authors
- External Secrets Operator (external-secrets.io) 167 authors
- HAProxy (haproxy.org) 166 authors
- Helm (helm.sh) 158 authors
- etcd (etcd.io) 152 authors
- Jaeger (jaegertracing.io) 150 authors
- cert-manager (cert-manager.io) 141 authors
- TKV (tkv.org) 138 authors
- Wix (wix.com) 134 authors
- Operator Framework (operatorframework.io) 128 authors
- Harbor (harbor.io) 125 authors
- Atlantis (iuniverse.io) 118 authors
- Thanos (thanos.io) 110 authors
- OpenBS (openbs.io) 108 authors
- LinuxChaos (linuxchaos.io) 107 authors
- Karmada (karmada.io) 100 authors
- Veeva (veeva.io) 103 authors
- Confidential Containers (github.com/confidential-containers)
- Volcano (volcano.sh) 98 authors
- Budibase (budibase.com) 97 authors
- OpenCost (opencost.com) 97 authors
- Konveyor (konveyor.io) 97 authors
- Strimzi (strimzi.io) 95 authors
- OVN-Kubernetes (ovn-kubernetes.io) 93 authors
- seamCloud (seamcloud.com) 92 authors
- Kubeescape (github.com/kubeescape) 92 authors
- Longhorn (longhorn.io) 90 authors
- Book (book.io) 90 authors
- Spiffe (spiffe.io) 87 authors
- HAMA (projecthama.io) 85 authors
- K8sPT (k8spt.io) 80 authors
- KubeEdge (kubedge.io) 80 authors
- WasmEdge Runtime (wasmedge.org) 79 authors
- Linkerd (linkerd.io) 76 authors
- Meta3 (meta3.io) 75 authors
- CubeFS (cubefs.io) 74 authors
- KusionStack (kusionstack.io) 74 authors
- Ingenuiti (ingenuiti.io) 70 authors
- Kame (kame.io) 68 authors
- Cloud Custodian (cloudcustodian.io) 68 authors
- 158 more



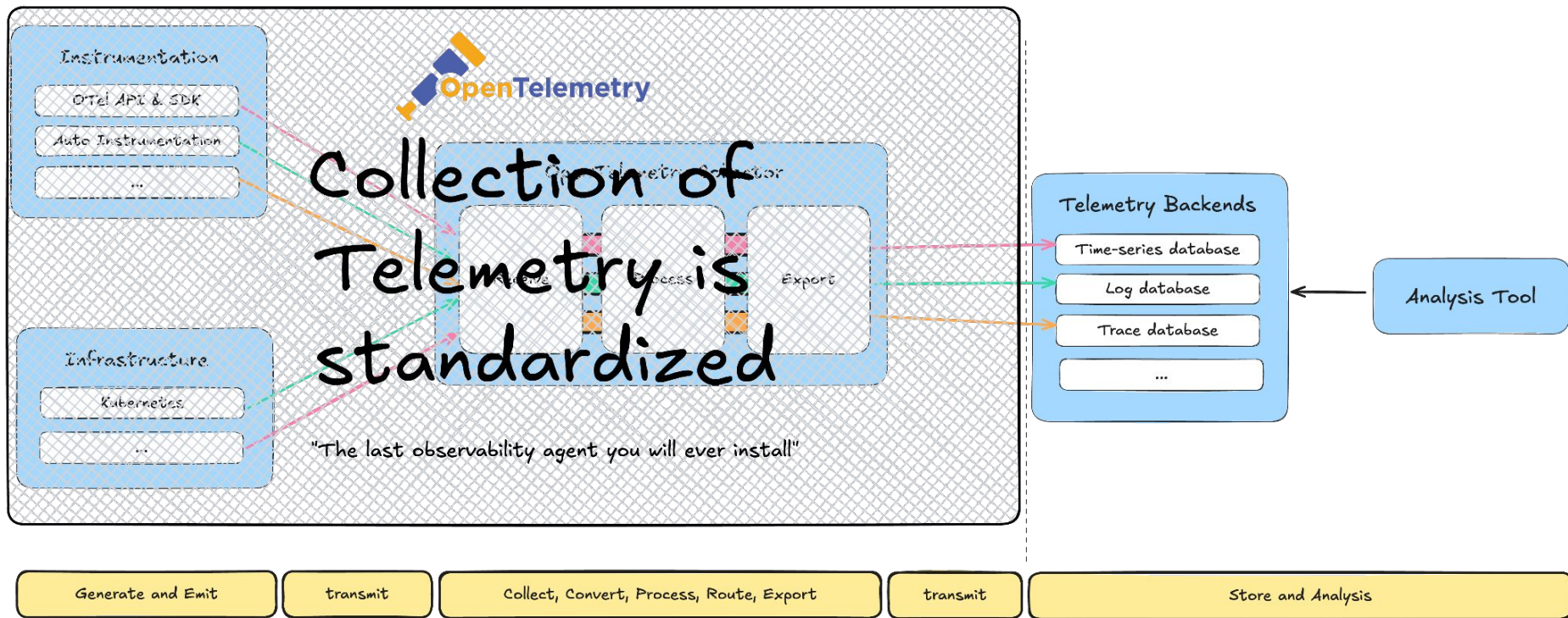
kubernetes

Commits: 44.486
PRs+Issues: 56.299Source: [CNCF Velocity](#)

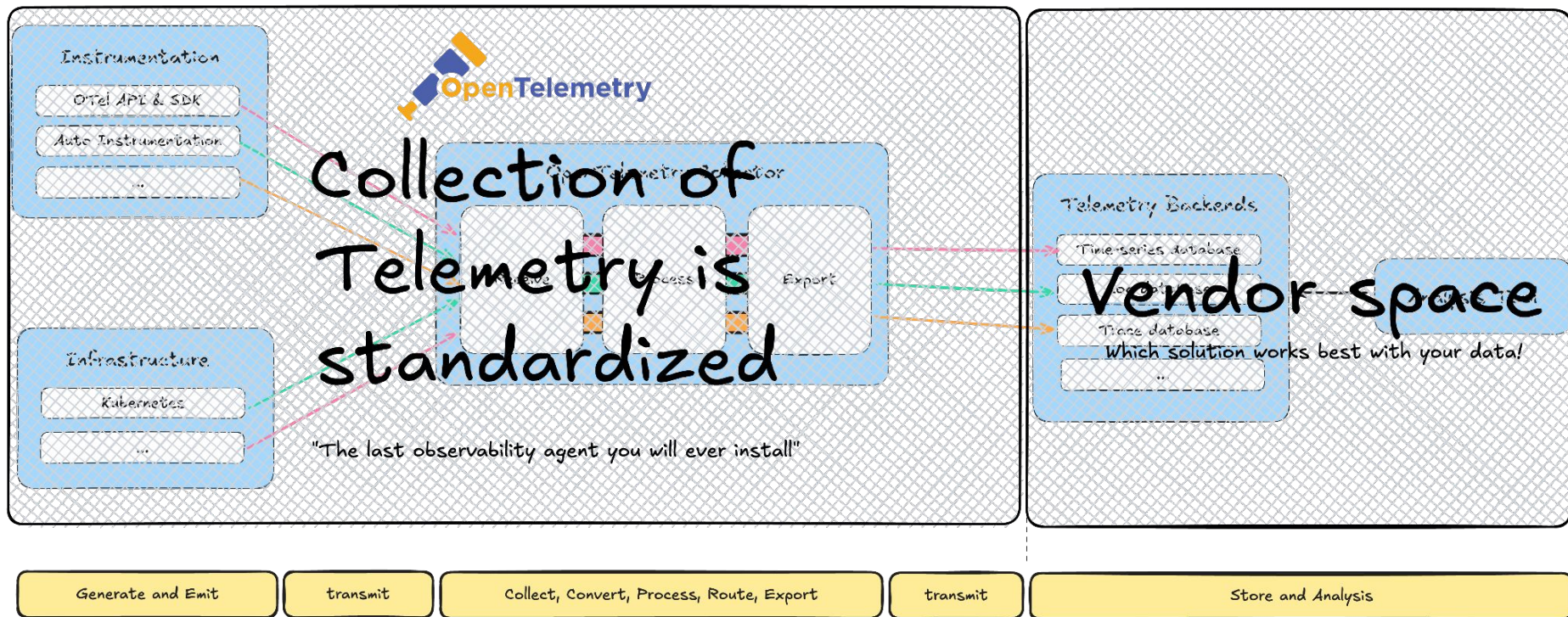
OpenTelemetry Collector



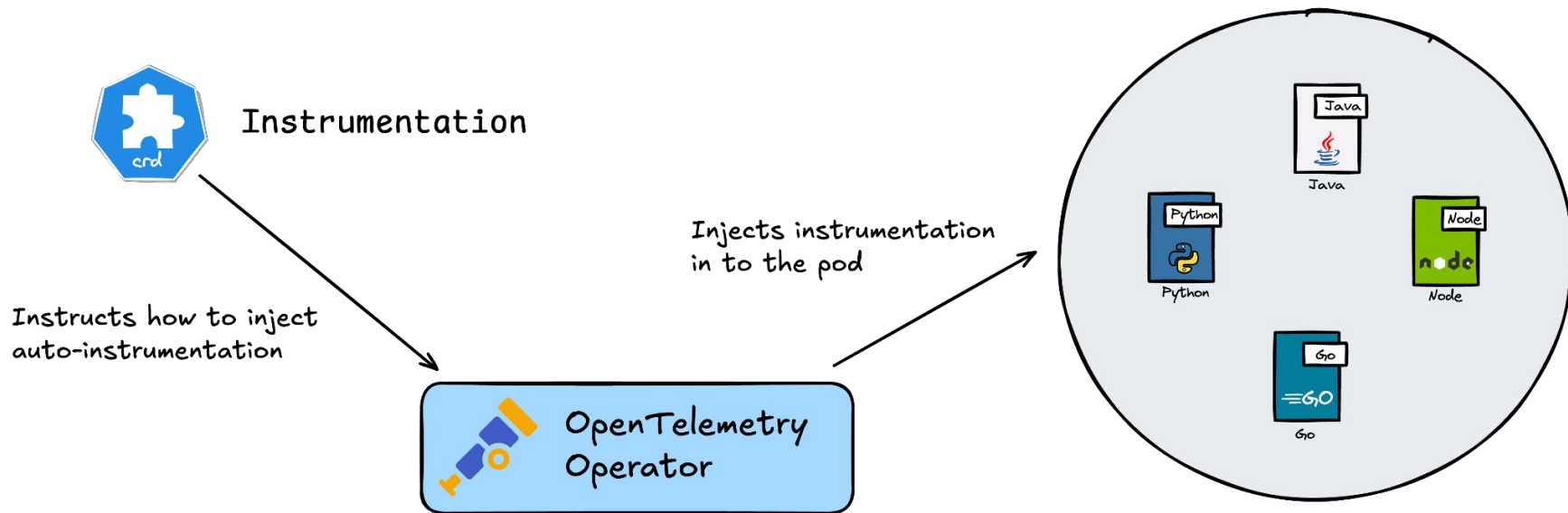
OpenTelemetry Collector



OpenTelemetry Collector



OpenTelemetry Operator

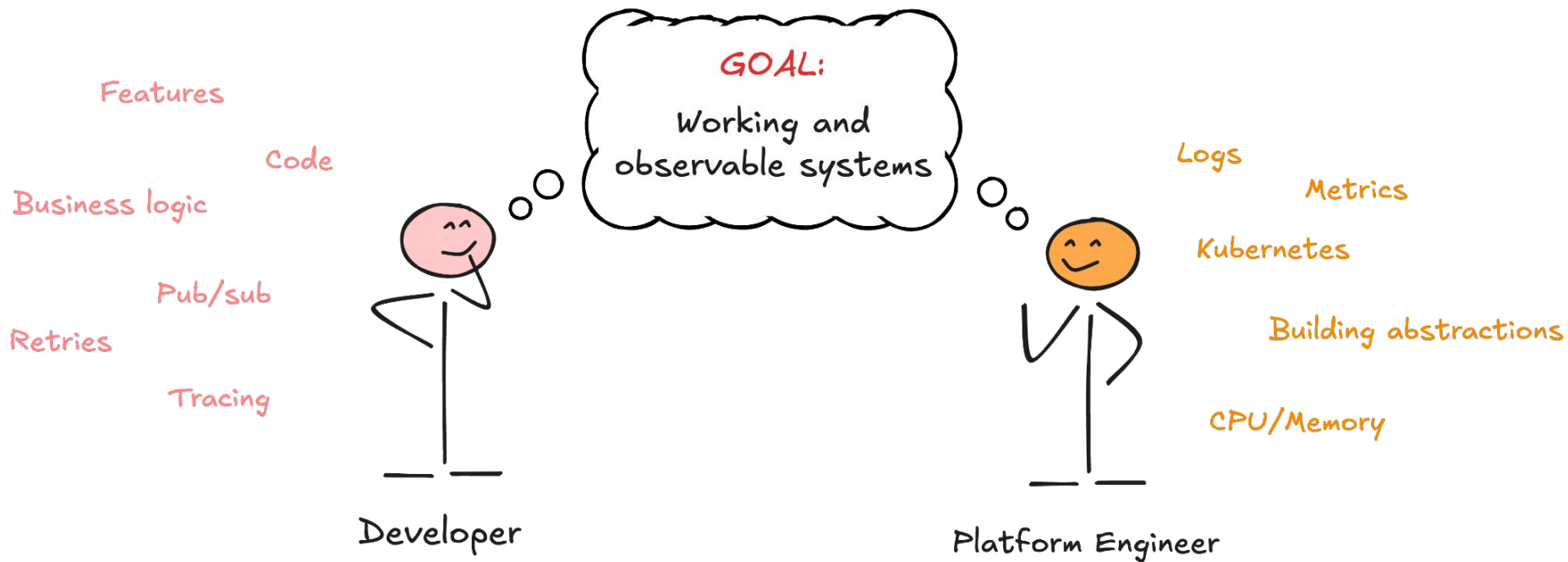




Making sense of all the complexity

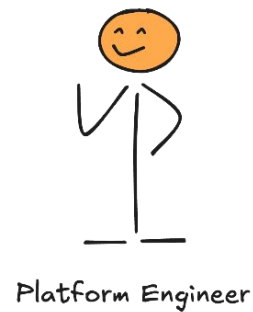
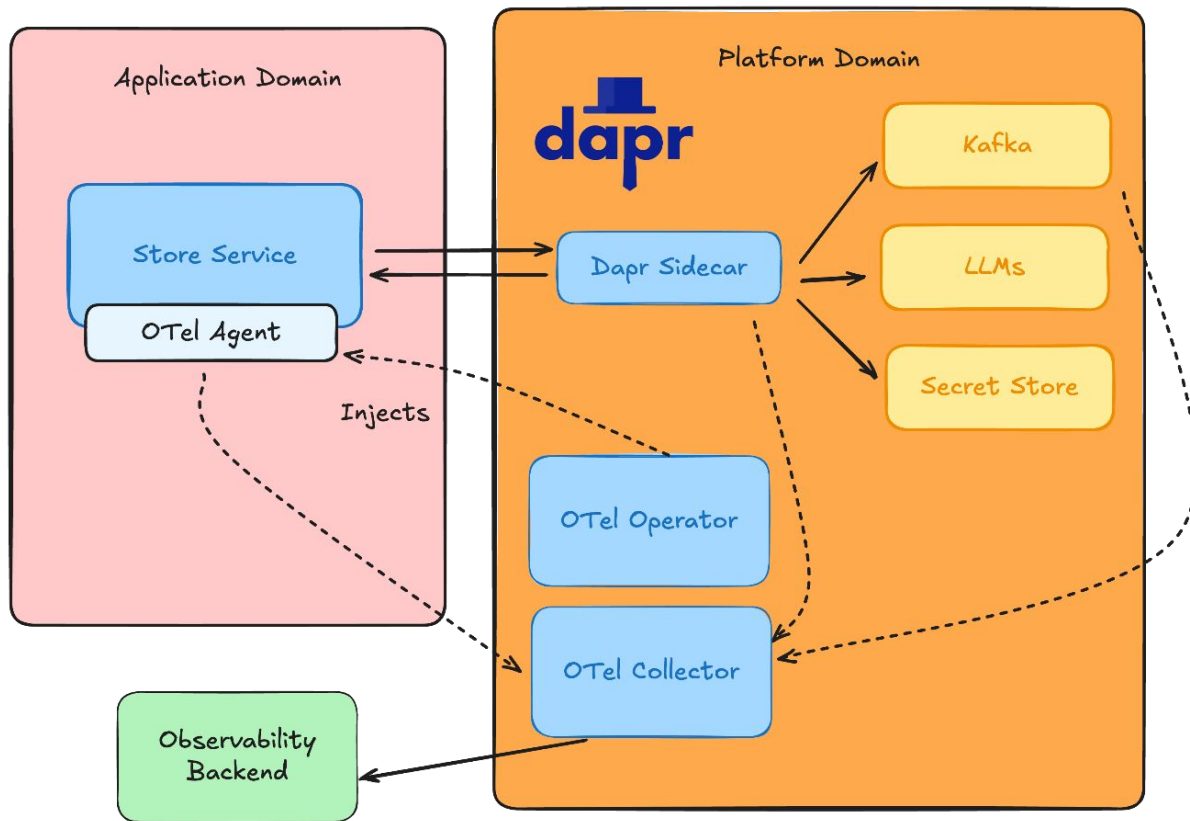


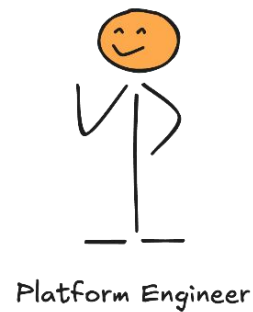
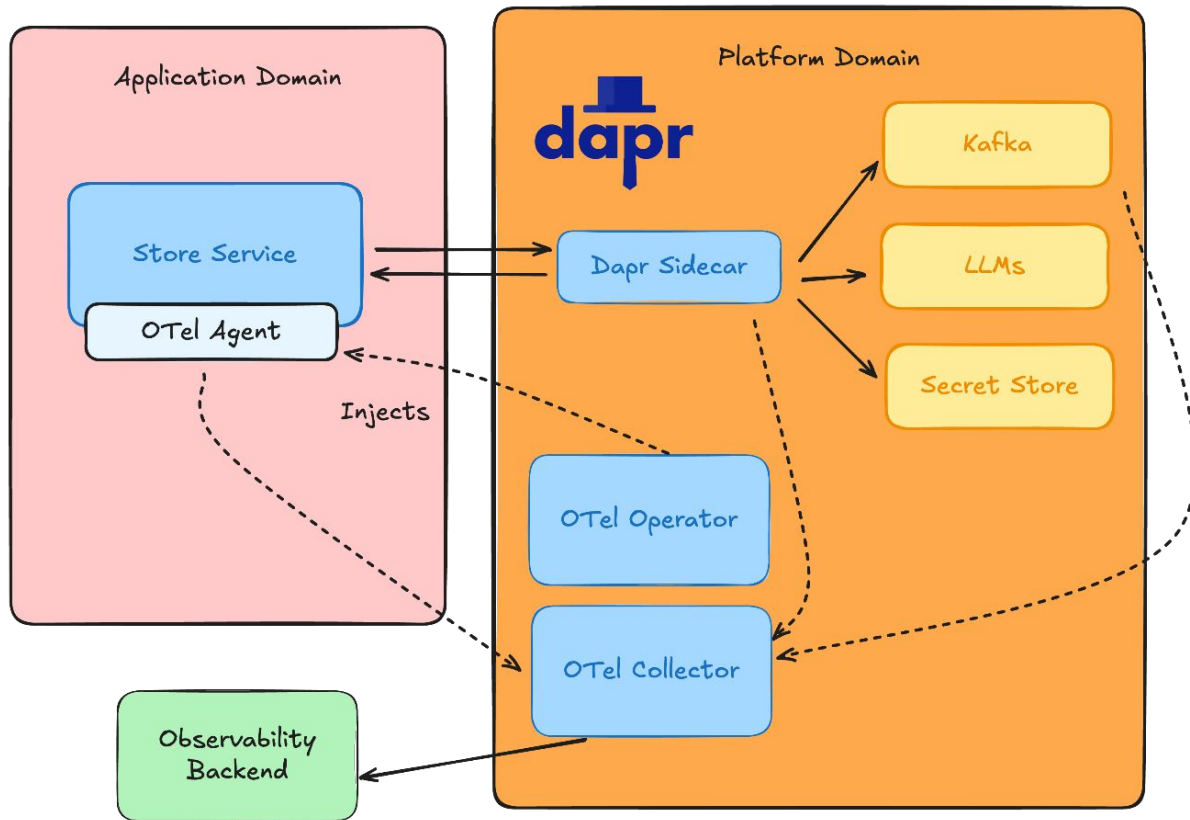
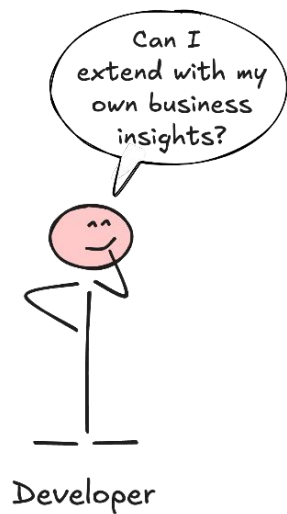
Two Perspectives, One Goal



Who Owns Tracing? A Hidden Conundrum



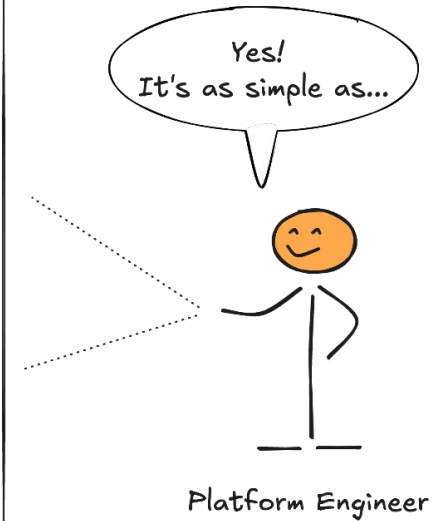




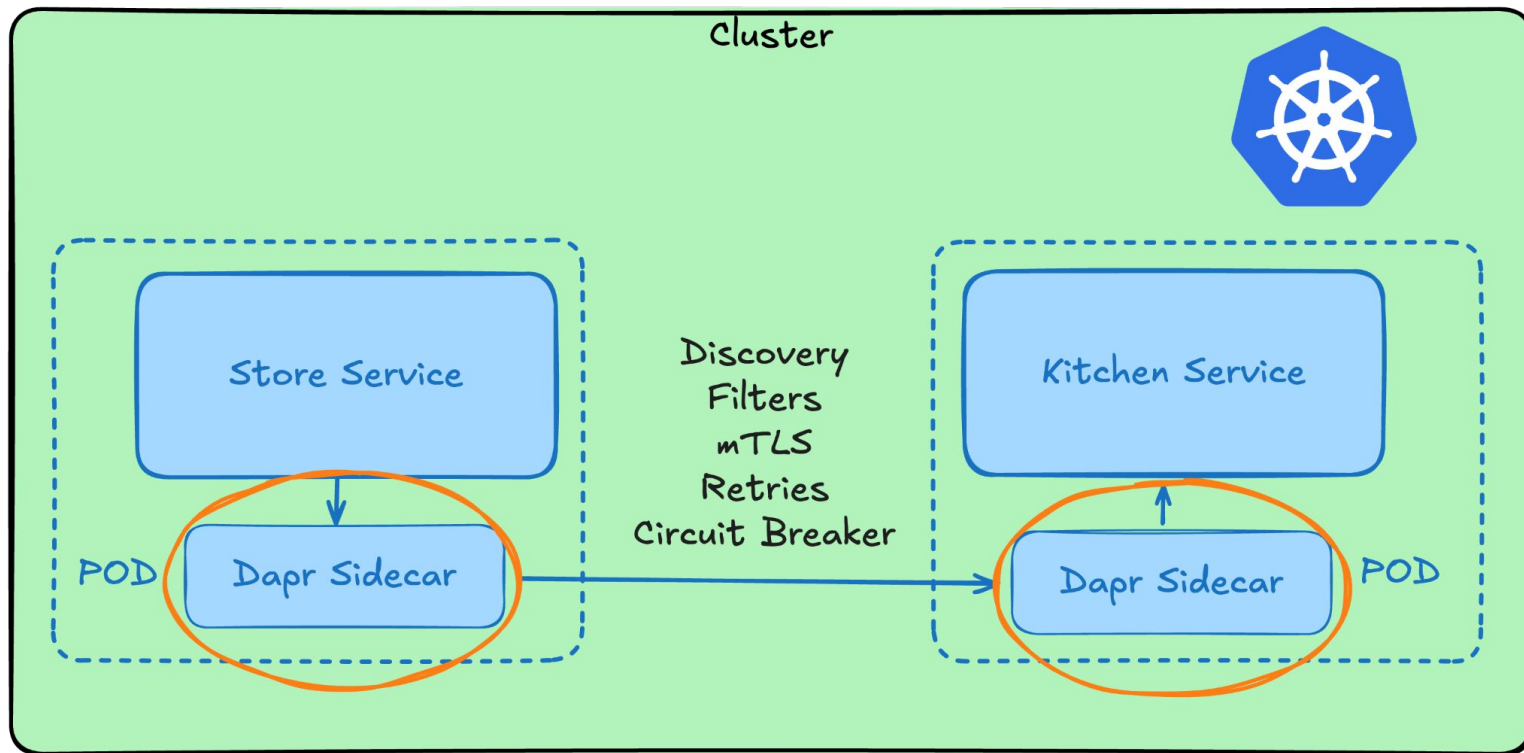
```
import io.opentelemetry.api.GlobalOpenTelemetry;
import io.opentelemetry.api.trace.Tracer;

Tracer tracer = GlobalOpenTelemetry.getTracer("application");

Span span = tracer.spanBuilder("doWork").startSpan();
...
span.end();
```



Why Observability is Critical with Dapr



Why Async Is Hard to Observe

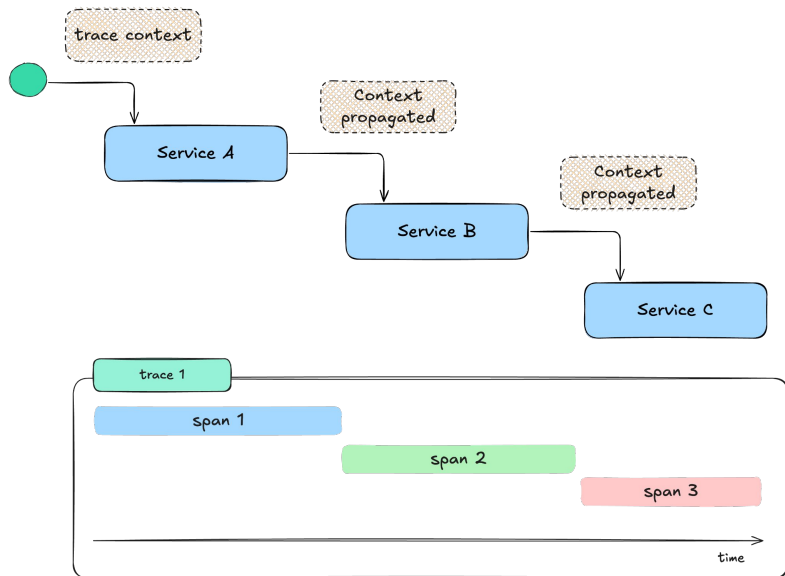
Traditional Tracing
assumes synchronous
request/response

Context often
survive between steps

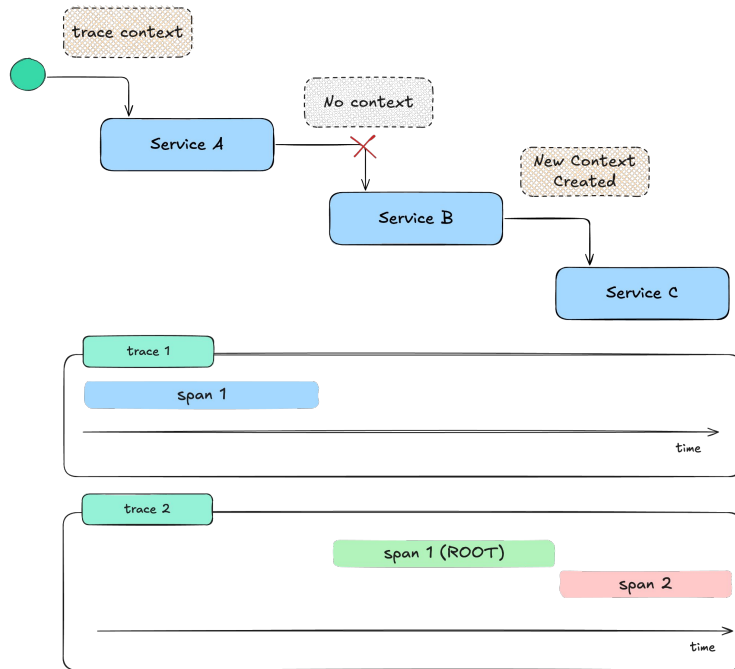


A Shared Pain: Context Gets Lost

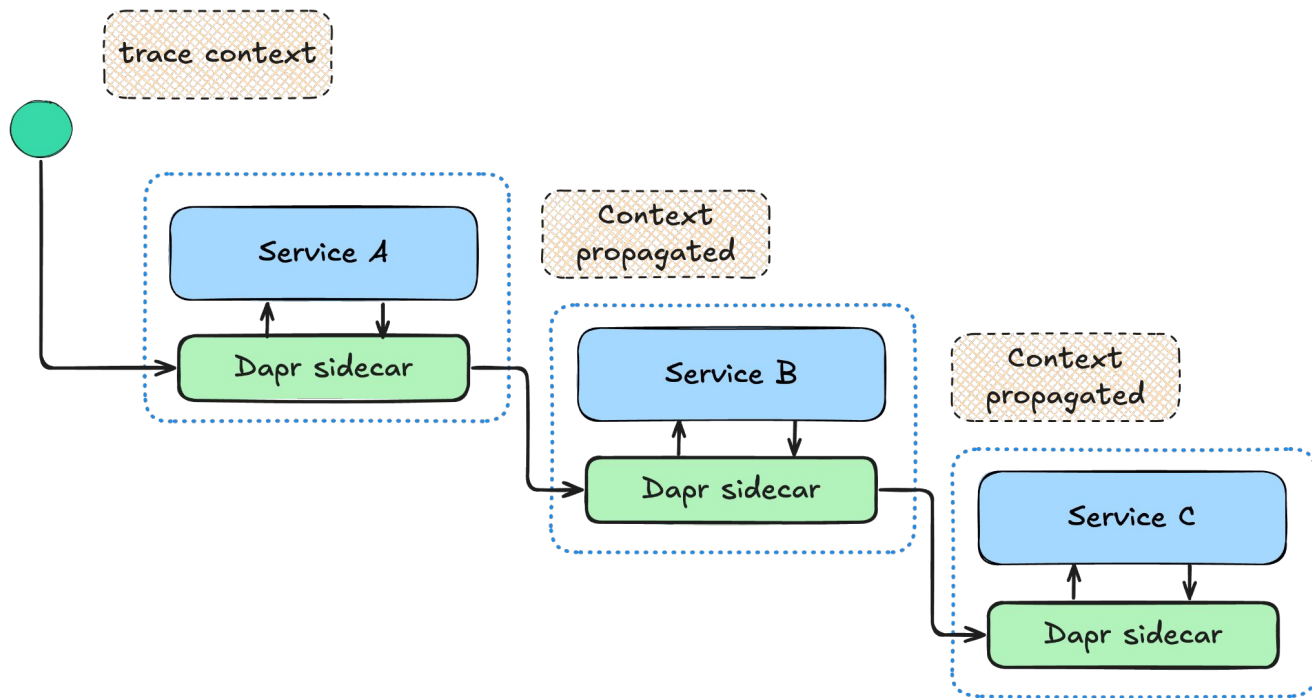
Context Propagation



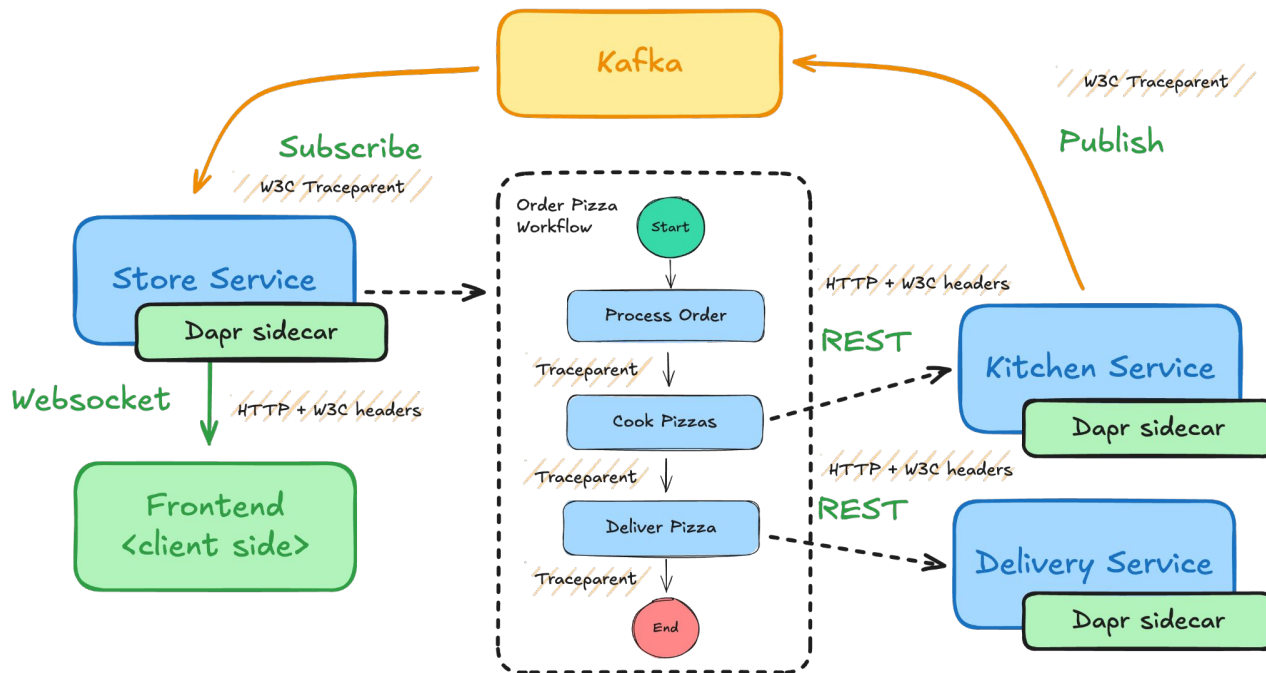
Broken Context Propagation



Trace propagation with Dapr



Context Propagation for Async Workflows



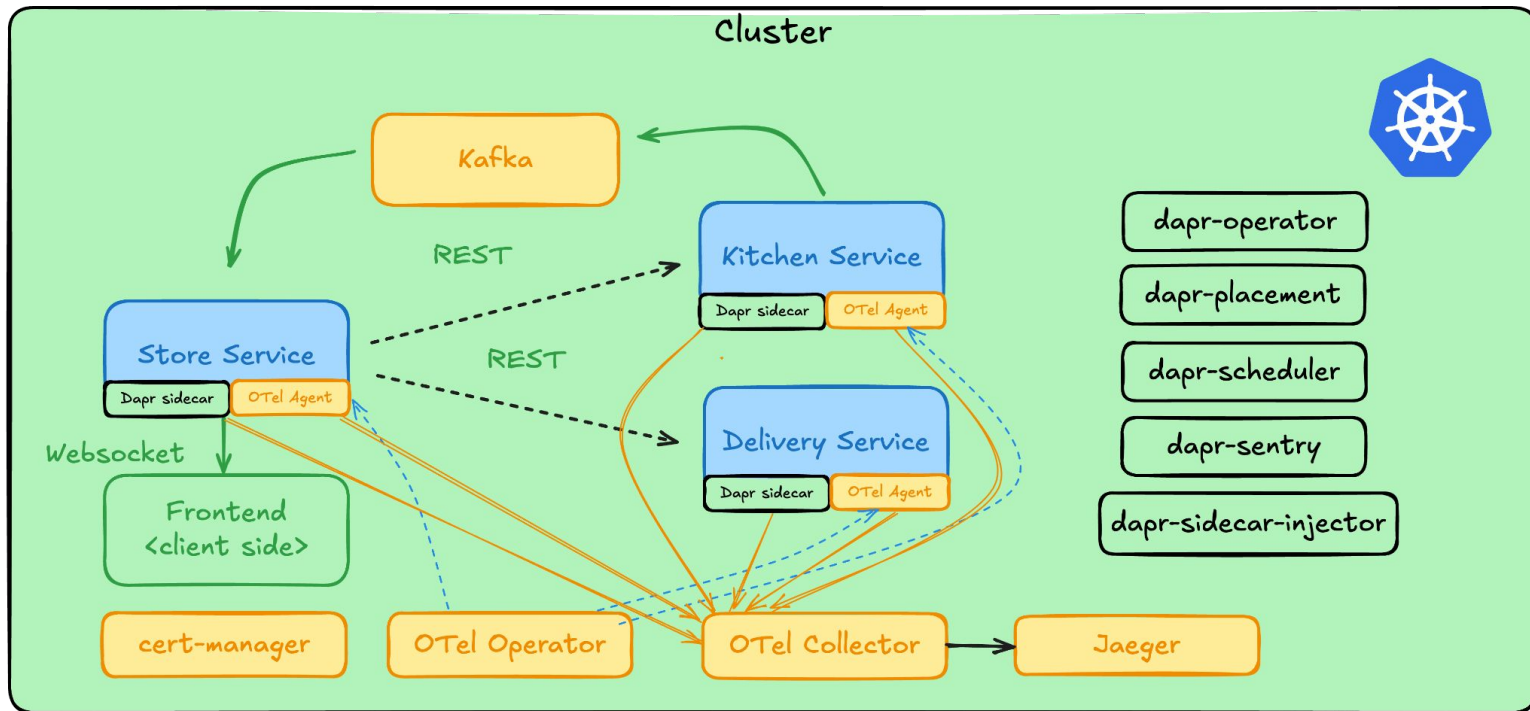
W3C Trace Context

traceparent: <version>-<trace_id>-<parent_id>-<trace_flags>

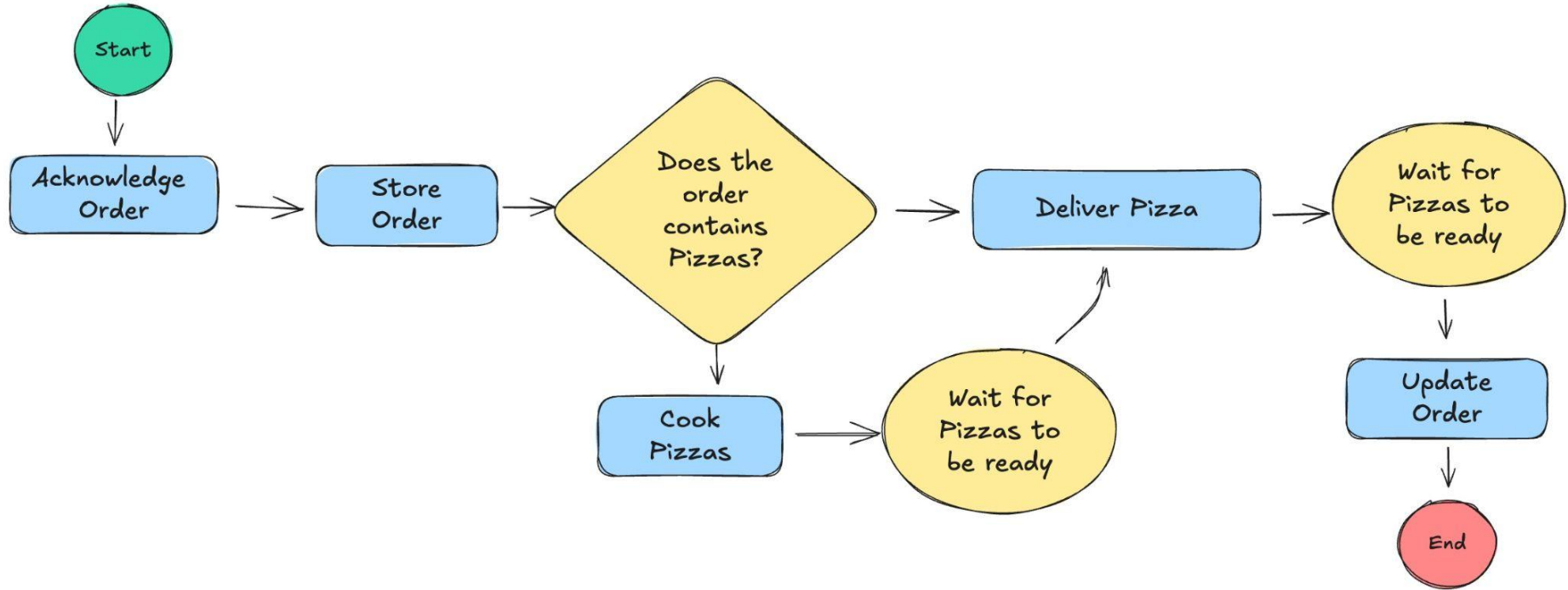
tracestate: <vendor_name>=<value>,<vendor_name>=<value>



Demo setup



Workflow Details





Demo #2

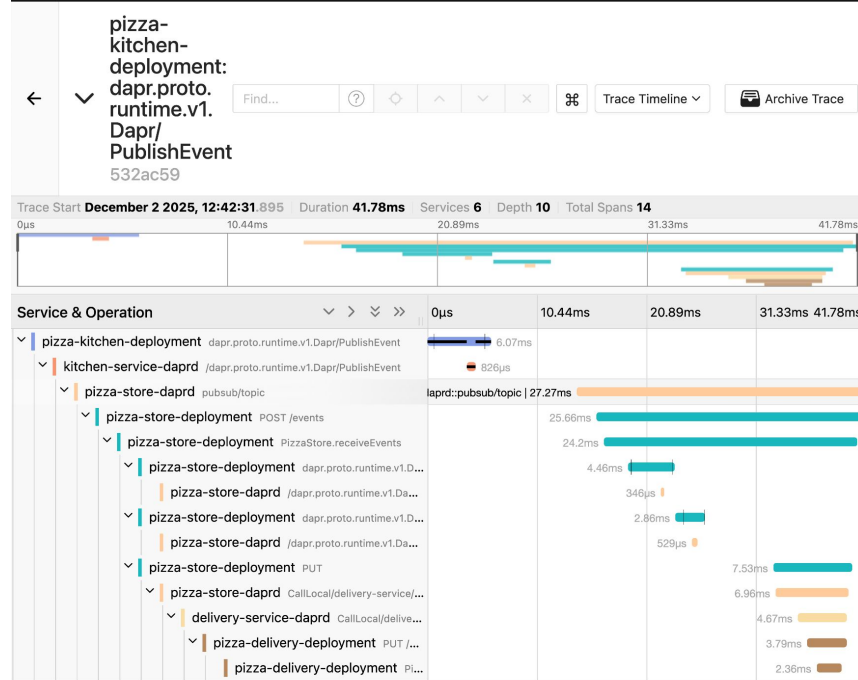
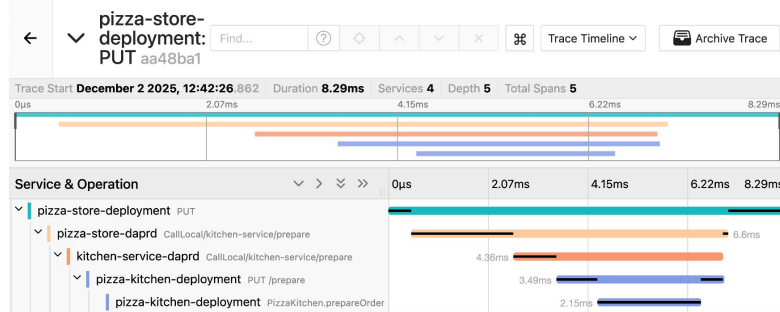
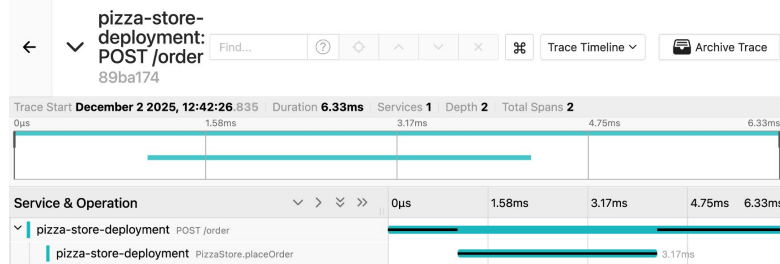


What Works Today

- Dapr supports OpenTelemetry out of the box
- Sidecar emits spans for pub/sub, service invocation, and workflows
- OpenTelemetry Operator enables auto-instrumentation
- OpenTelemetry Collector handles ingestion, processing, export



What Works Today

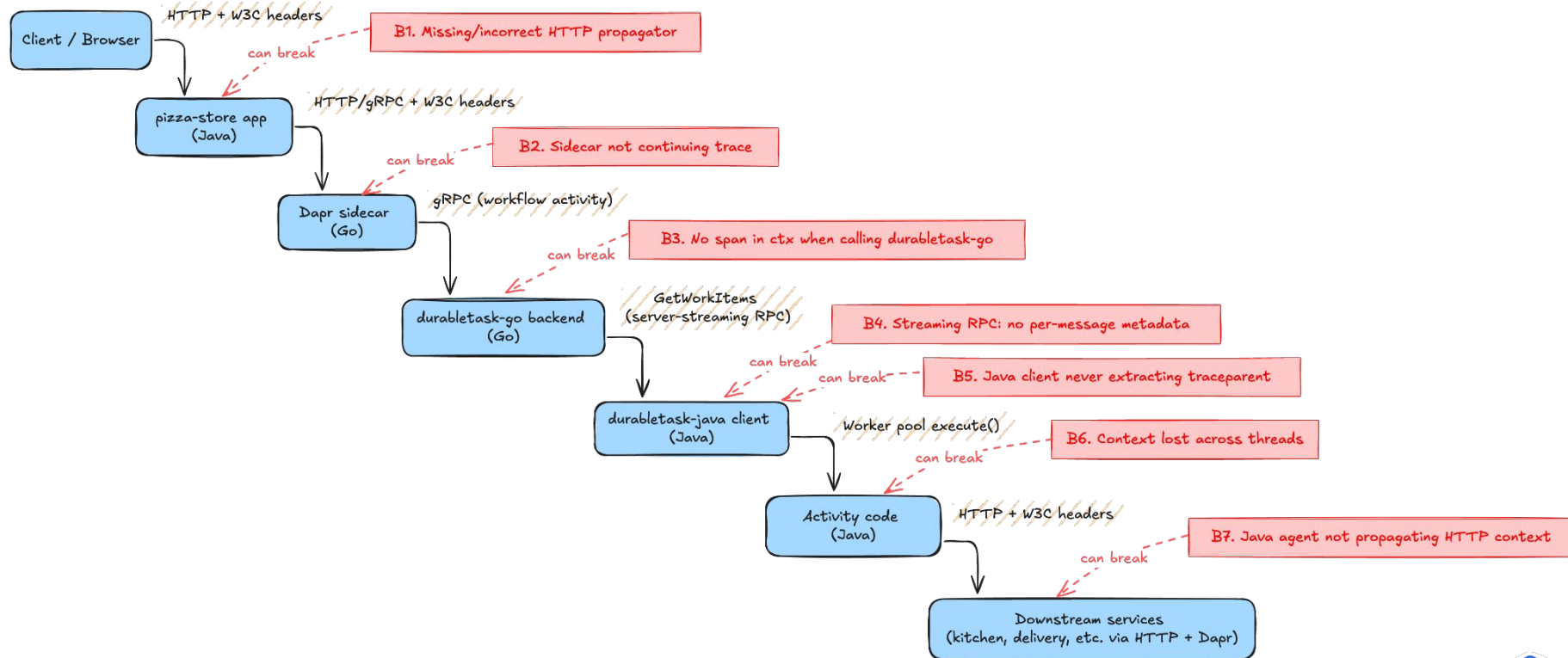


Challenges

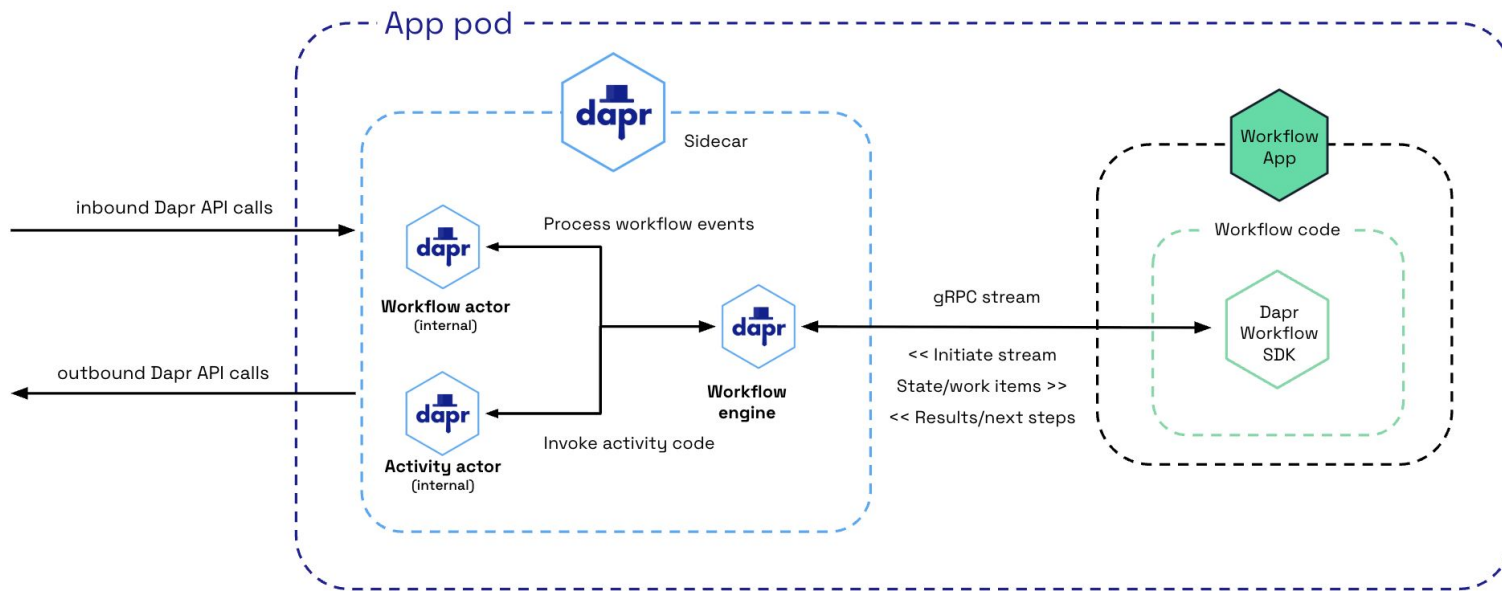
- Async boundaries break context
- Sidecars add additional hops
- Workflow engines introduce thread + process separation



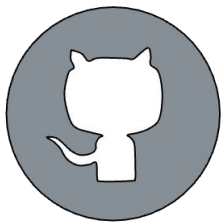
Context Propagation for Async Workflows



Challenges with gRPC streaming

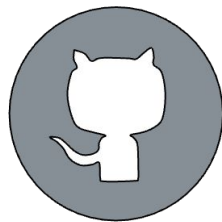


Gaps and Fixes in Dapr & OTel



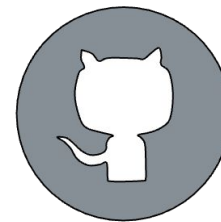
PR #57

Trace context,
SemConv



PR #9213

Trace context,
SemConv, Pub/Sub
Span kind

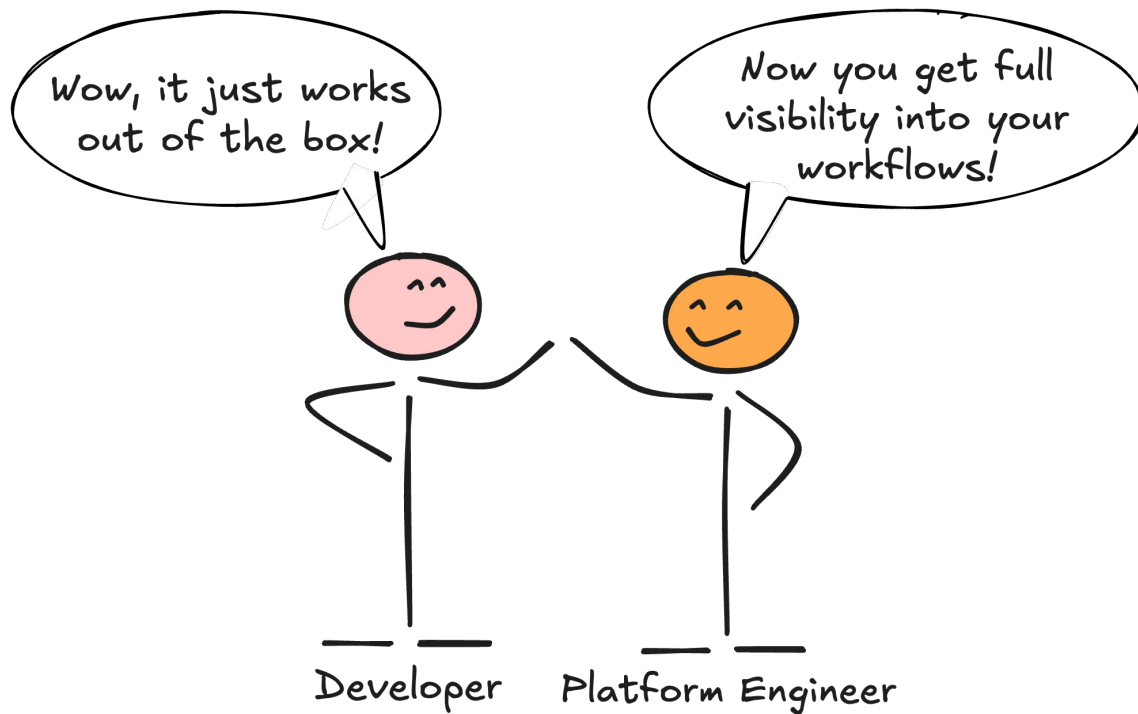


PR #46

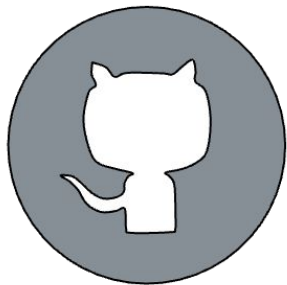
Propagating context
to executors client
side



Enabling the Golden Path...



Get Involved



Github

github.com/open-telemetry
github.com/dapr



CNCF Slack

#dapr
#opentelemetry, otel-*



KCD Suisse Romande 
DECEMBER 4 & 5, 2025 – CERN, GENEVA





Thank you!



Abstract

Async workflows power modern microservices, but they can be notoriously hard to observe. In this talk, we show how two CNCF projects - Dapr, for developer-friendly building blocks, and OpenTelemetry, for unified observability - create a golden path that bridges developer productivity and platform reliability.

We'll start by using Dapr Workflows and Pub/Sub to connect and orchestrate services without boilerplate. Then we'll add the OpenTelemetry Operator for no-touch instrumentation, instantly delivering traces, metrics, and logs - even across asynchronous boundaries.

You'll see current OpenTelemetry capabilities for tracking async requests end-to-end, where the gaps are today, and practical ways to correlate events in complex workflows. Through a live demo, we'll prove that with the right abstractions, shipping features fast and observing systems deeply can go hand in hand.

