



When Prometheus met OpenTelemetry

Fosdem 2024

Pavol Loffay
Principal Software Engineer



About



- Pavol Loffay
- Principal Software Engineer @ Red Hat
- OpenTelemetry operator, Tempo operator and Jaeger maintainer



@ploffay



CNCF Slack

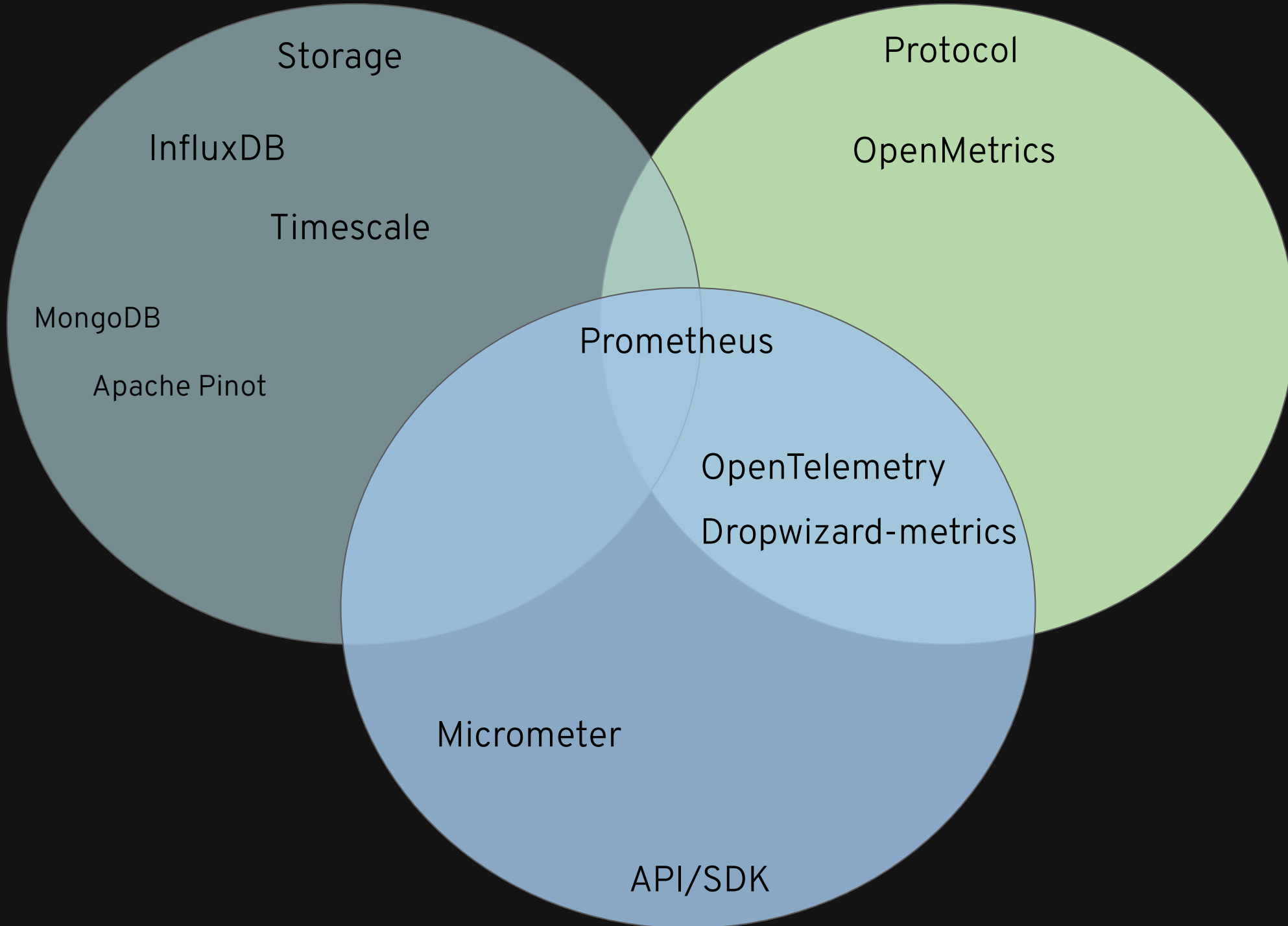


Agenda

- OSS metrics ecosystem
- Data model differences
- Prometheus in the OpenTelemetry ecosystem
 - API/SDK & Collector
- Collecting Prometheus metrics on Kubernetes
 - OpenTelemetry target allocator
- Wrap up

Why? (OSS metrics ecosystem)

Shall we create another “standard”?



Data model differences

Prometheus data model vs OpenTelemetry

Protocol and Data model differences

- Pull vs Push
- Cumulative vs delta temporality [#12763](#)
- ~~Native vs exponential histograms (with min/max)~~
- target_info vs resource attributes [#1262](#)
- [float vs int](#)
- Utf8 support for label and metric names [#13095](#)

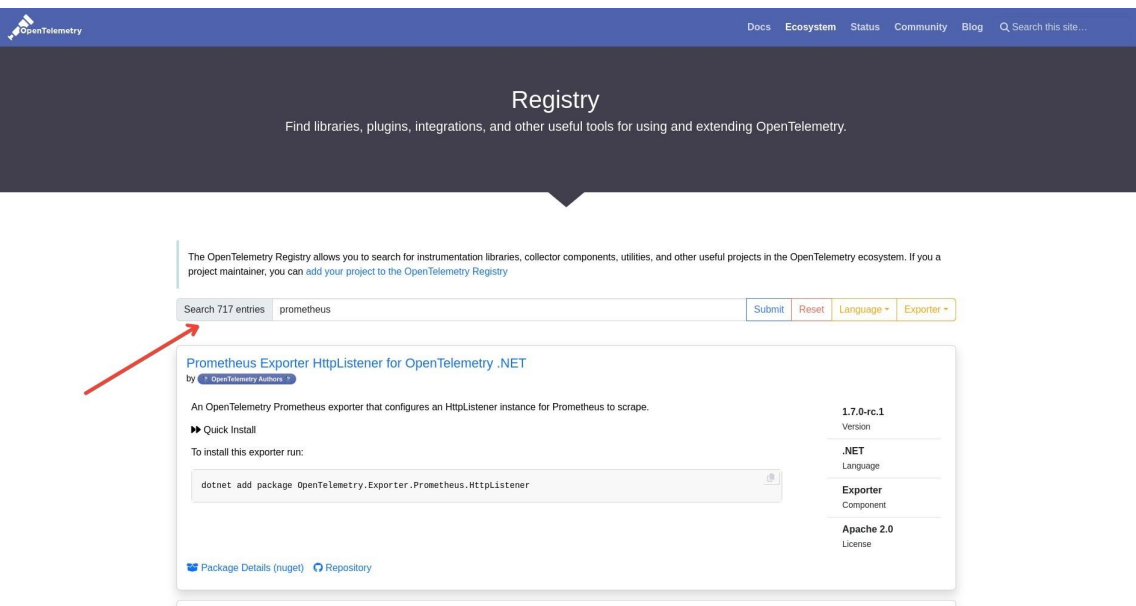


Temporality

- OTEL_EXPORTER_OTLP_METRICS_TEMPORALITY_PREFERENCE
 - Cumulative, Delta, LowMemory
- Prometheus exporter
 - Collector does delta2cumulative
 - SDKs might not support delta2cumulative

Prometheus in OpenTelemetry ecosystem

Prometheus receiver/exporter, target allocator



The screenshot shows the OpenTelemetry Registry website. The header includes the OpenTelemetry logo and navigation links: Docs, Ecosystem, Status, Community, Blog, and a search bar. The main heading is "Registry" with the subtitle "Find libraries, plugins, integrations, and other useful tools for using and extending OpenTelemetry." Below this, a search bar contains the text "Search 717 entries" and "prometheus". A red arrow points to the search bar. The search results show a card for "Prometheus Exporter HttpListener for OpenTelemetry .NET" by "OpenTelemetry Authors". The card includes a description: "An OpenTelemetry Prometheus exporter that configures an HttpListener instance for Prometheus to scrape." and a "Quick Install" section with the command: `dotnet add package OpenTelemetry.Exporter.Prometheus.HttpListener`. The card also lists the version "1.7.0-rc.1", language ".NET", component "Exporter", and license "Apache 2.0".

OpenTelemetry SDKs

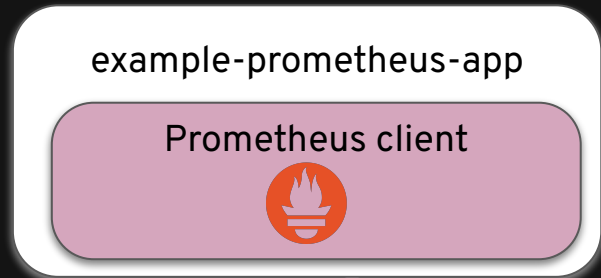
- Prometheus exporter
 - delta temporality not well supported

OpenTelemetry Collector

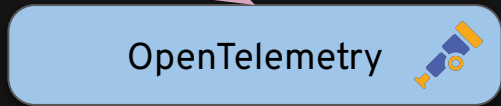
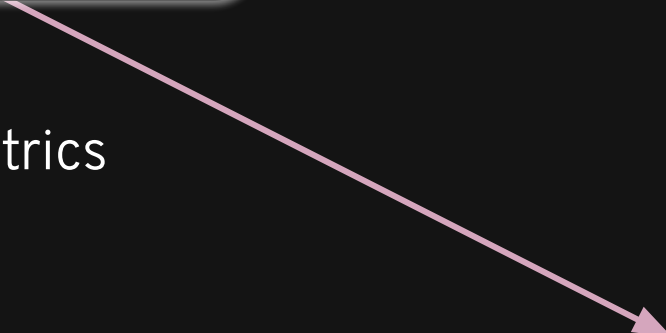
- Prometheus receiver
- Prometheus exporter
 - should support delta2cumulative
- Prometheus Remote Write Exporter
 - delta temporality not well supported

Demo

OpenTelemetry collector with Prometheus



/metrics



/metrics
debug exporter

Prometheus receiver

Manually configure prometheus receiver

Prometheus receiver - static config

```
apiVersion: opentelemetry.io/v1alpha1
kind: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: deployment
  config: |
    receivers:
      prometheus/static:
        config:
          scrape_configs:
            - job_name: 'app'
              scrape_interval: 10s
              static_configs:
                - targets: [ 'prometheus-example-app:80' ]
```

Prometheus receiver service discovery

Via OpenTelemetry collector

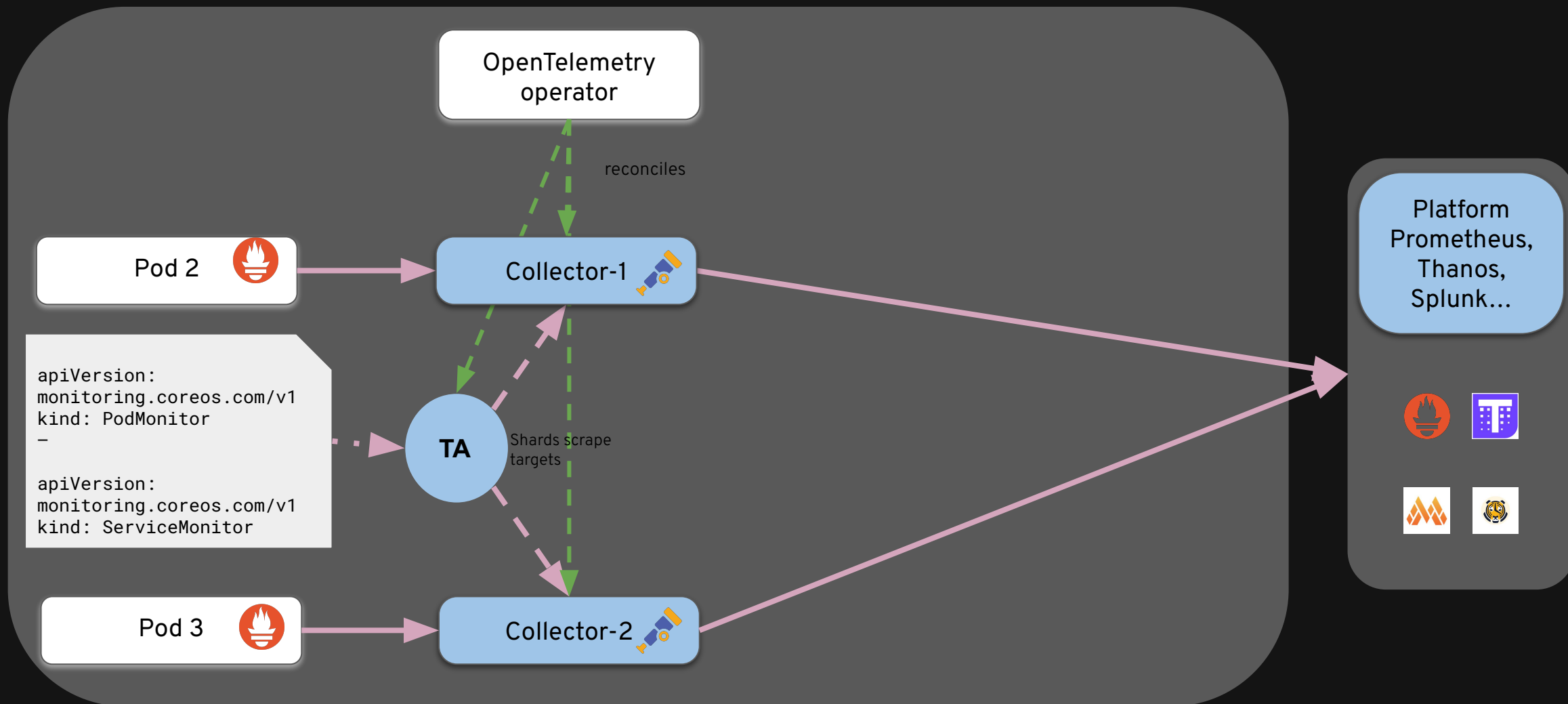
Prometheus receiver with service discovery

```
apiVersion: opentelemetry.io/v1alpha1
kind: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: deployment
  config: |
    receivers:
      prometheus/k8s:
        config:
          scrape_configs:
            - job_name: k8s
              scrape_interval: 10s
              kubernetes_sd_configs:
                - role: pod
                  selectors:
                    - role: pod
                      label: "app.kubernetes.io/name=prometheus-example-app"
```

Prometheus receiver and Service/Pod monitors

OpenTelemetry collector and target allocator

OpenTelemetry collector with target allocator

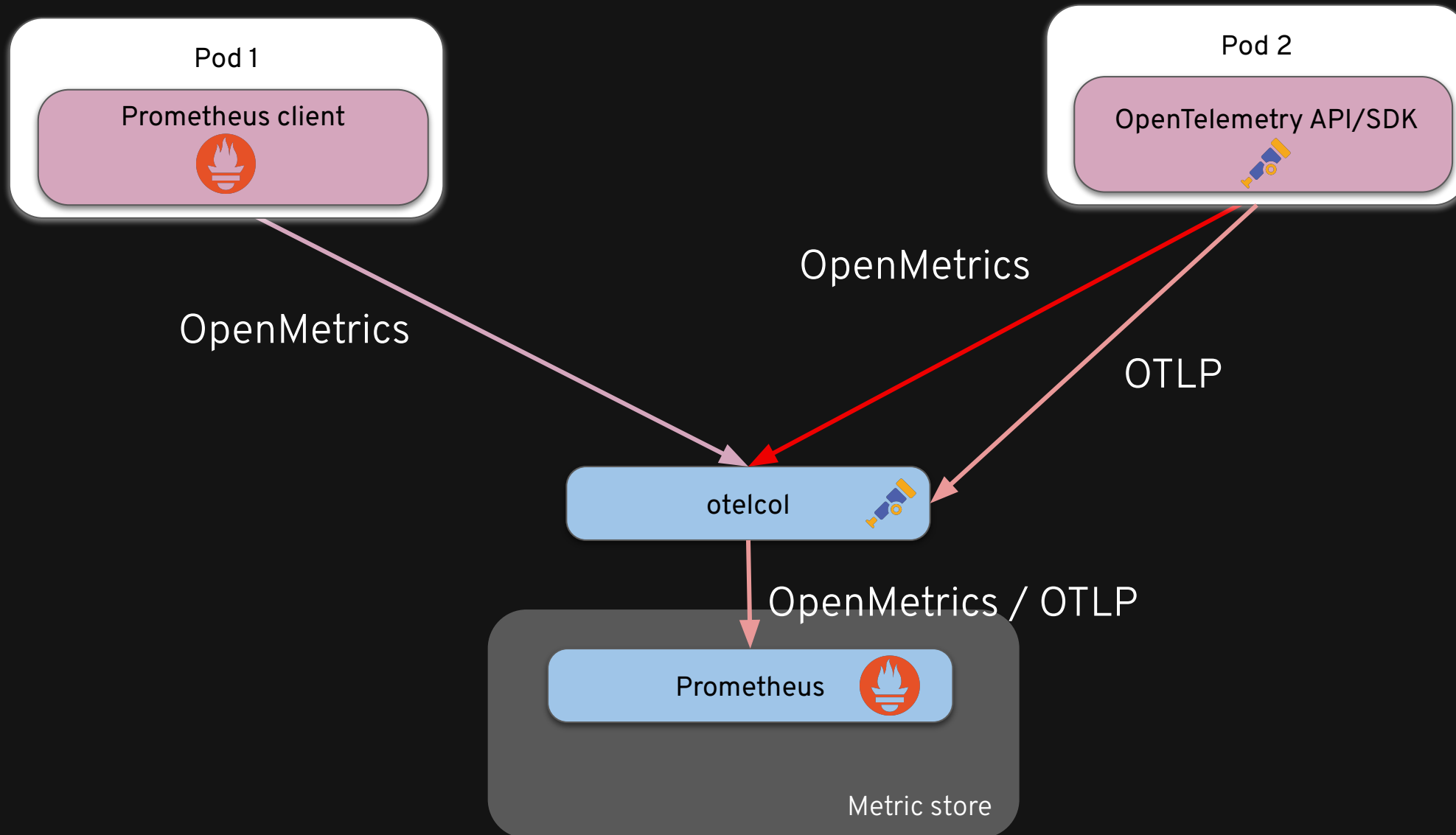


Prometheus receiver with target allocator

```
apiVersion: opentelemetry.io/v1alpha1
kind: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: statefulset
  targetAllocator:
    enabled: true
  prometheusCR:
    enabled: true
    scrapeInterval: 5s
    serviceMonitorSelector:
      name: prometheus-example-app
  config: |
    receivers:
      prometheus:
        config:
          scrape_configs: []
```

Wrap-up

We have learned some OpenTelemetry magic tricks



Wrap-up

- Be careful with OTEL instrumented services (delta temporality)
- Prometheus receiver
 - escape \$ with (\$\$)
- Probe and ScrapeConfig CRDs are not supported
- Target allocator
 - TLS
 - Other options from the Prometheus spec are not supported [#1934](#)

Prometheus call to action

- Prometheus 3.0 OTLP
- [prometheus/projects/9/views/8](https://github.com/prometheus/projects/9/views/8)

A screenshot of the GitHub project page for Prometheus 3.0. The page shows a list of issues filtered by the label "help wanted". The issues are organized into milestones: "Native Histograms", "OTEL Support", "Prometheus 3.0", and "No Milestone". A section titled "OTLP" contains 8 issues, and a section titled "Feature Flags" contains 2 issues. The issues are listed with their titles, IDs, assignees, and status (e.g., "Assigned", "Todo").

Milestone	Title	ID	Assignee	Status
Native Histograms	Native Histograms: support exemplars in pure native histograms	#1120	rafaelcnp	Assigned
OTEL Support	Aggregations like avg, min and max produce incorrect results for native histograms	#11973	beorn7	Todo
Prometheus 3.0	Update tsdb/docs/... for native histograms and out-of-order ingestion	#12005		Todo
OTLP (8)				
	Store metric metadata in TSDB	#12608		Todo
	Performance: when ingesting Otel metrics, use a cache for metric and labels nor...	#12627		Todo
	Handle OTEL resource attributes	#12629		Todo
	Mark out-of-order as stable	#12631		Todo
	Discussion: Enable 5 minutes of out-of-order by default in Prometheus	#12632		Todo
	Document OTLP ingestion endpoint	#12642		Todo
	OTLP ingestion creates too many metrics	#12655		Todo
	Support Otel delta temporality	#12763		Todo
Feature Flags (2)				
	remove the remote-write-receiver flag from enable-feature	#13206		Assigned

OpenTelemetry call to action

- OpenTelemetry collector v1alpha2 CRD
- Target Allocator CRD
- Prometheus receiver/exporter in the collector



The image shows two screenshots of GitHub pages. The left screenshot displays the 'v1alpha2 CRD release' milestone page for the 'opentelemetry-operator' repository. It features a progress bar at 36% completion, a due date of February 01, 2024, and a list of 21 open issues. The right screenshot shows the 'opentelemetry-collector-contrib' repository's issue list, filtered for 'is:open is:issue label:exporter/prometheus'. It displays a list of 13 open issues, including 'Collector agent does not return spanmetrics', 'prometheusexporter: metrics plummeted without prior notification', and 'prometheusexporter's "metric_expiration" parameter is ignored when used with spanmetrics connector'.

Thank you

Pavol Loffay
@ploffay