Grafana Labs

Prometheus, Grafana, Loki – aber wo bleibt OpenTelemetry?

Kubernetes Observability in Action

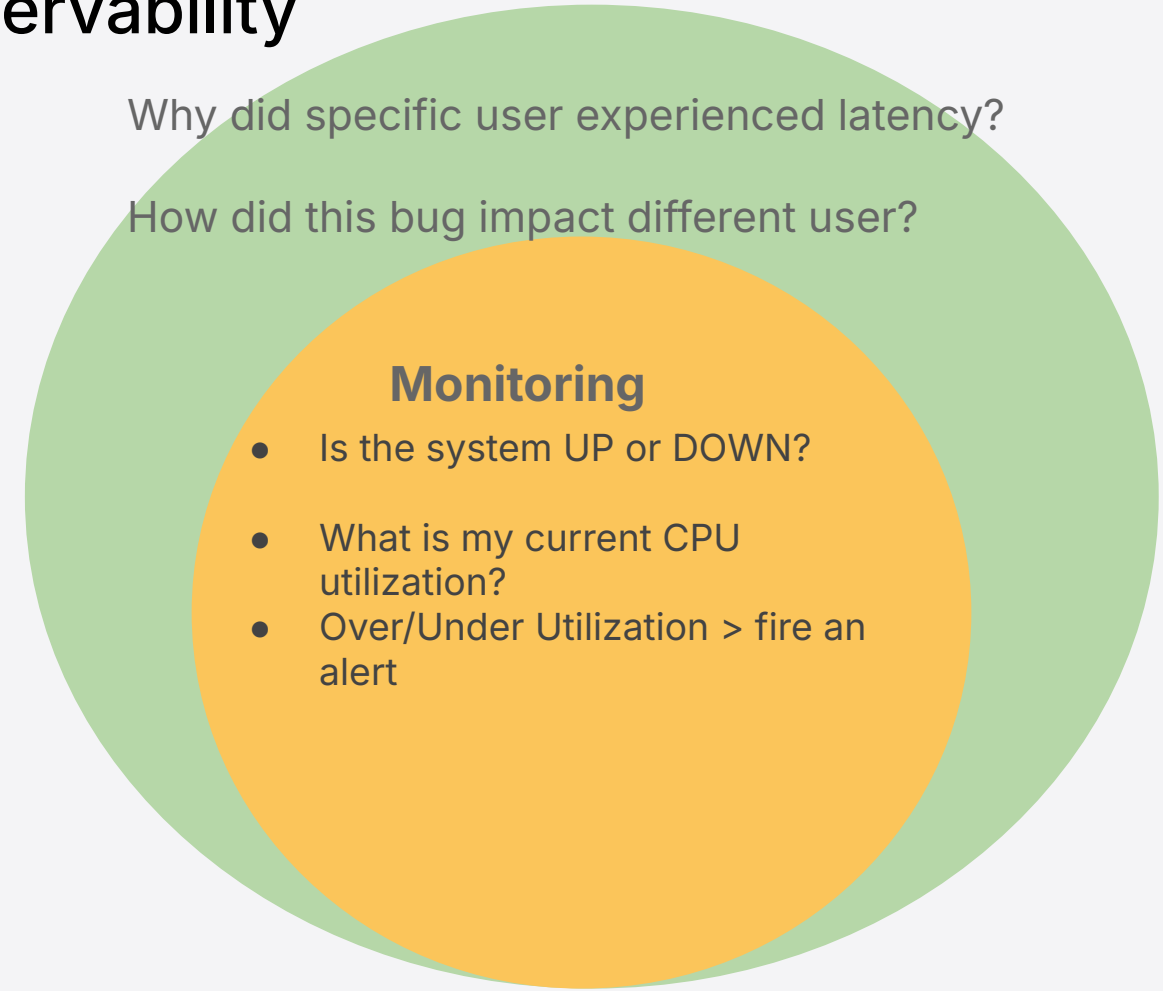
Mehmet Baykara
Sr. Solutions Architect
@ Grafana Labs

18.02.2025

AGENDA

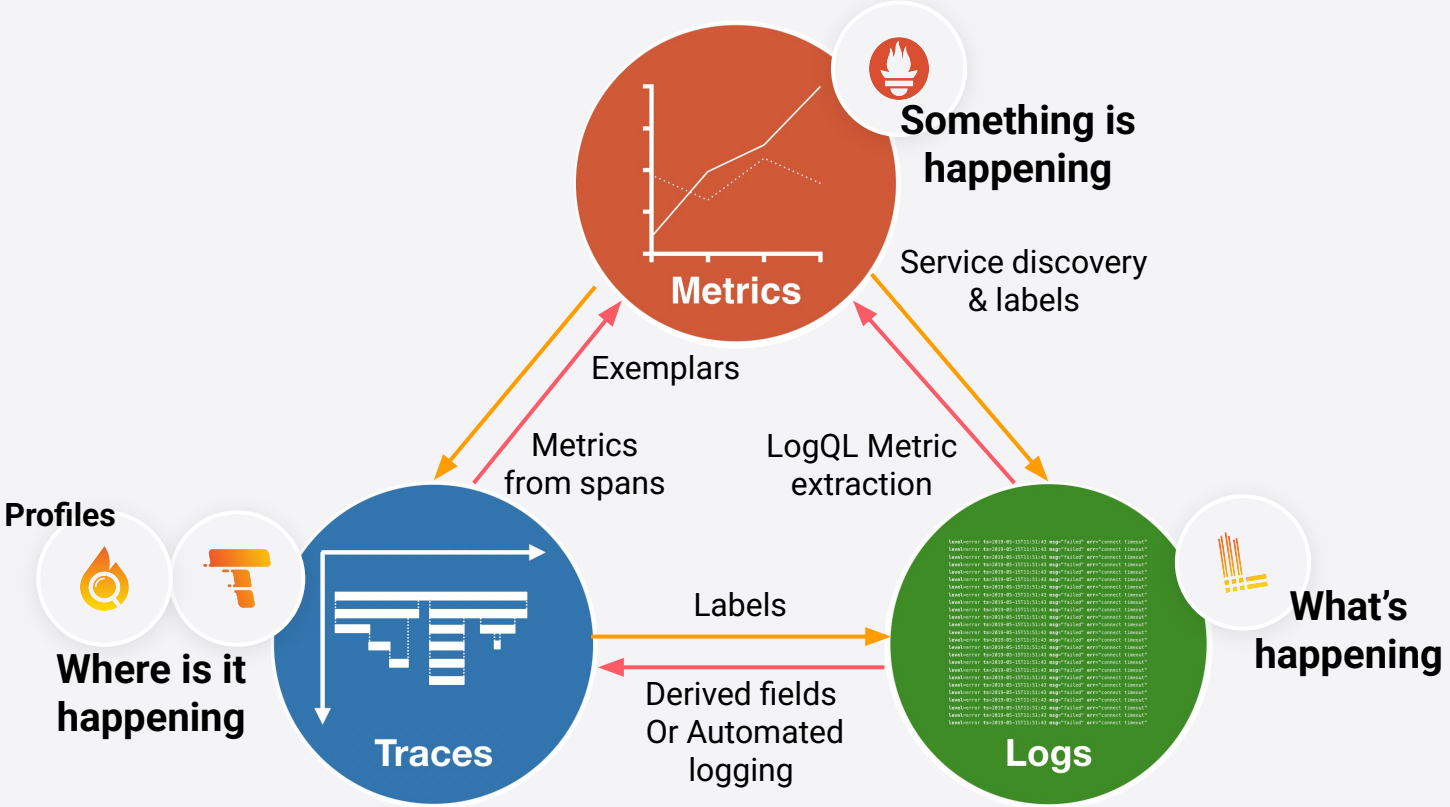
- Introduction
- Observability
- LGTM
- Introduction to OpenTelemetry
- OTel Collectors - Grafana Alloy
- K8s-monitoring Chart
- Demo
- Q&A

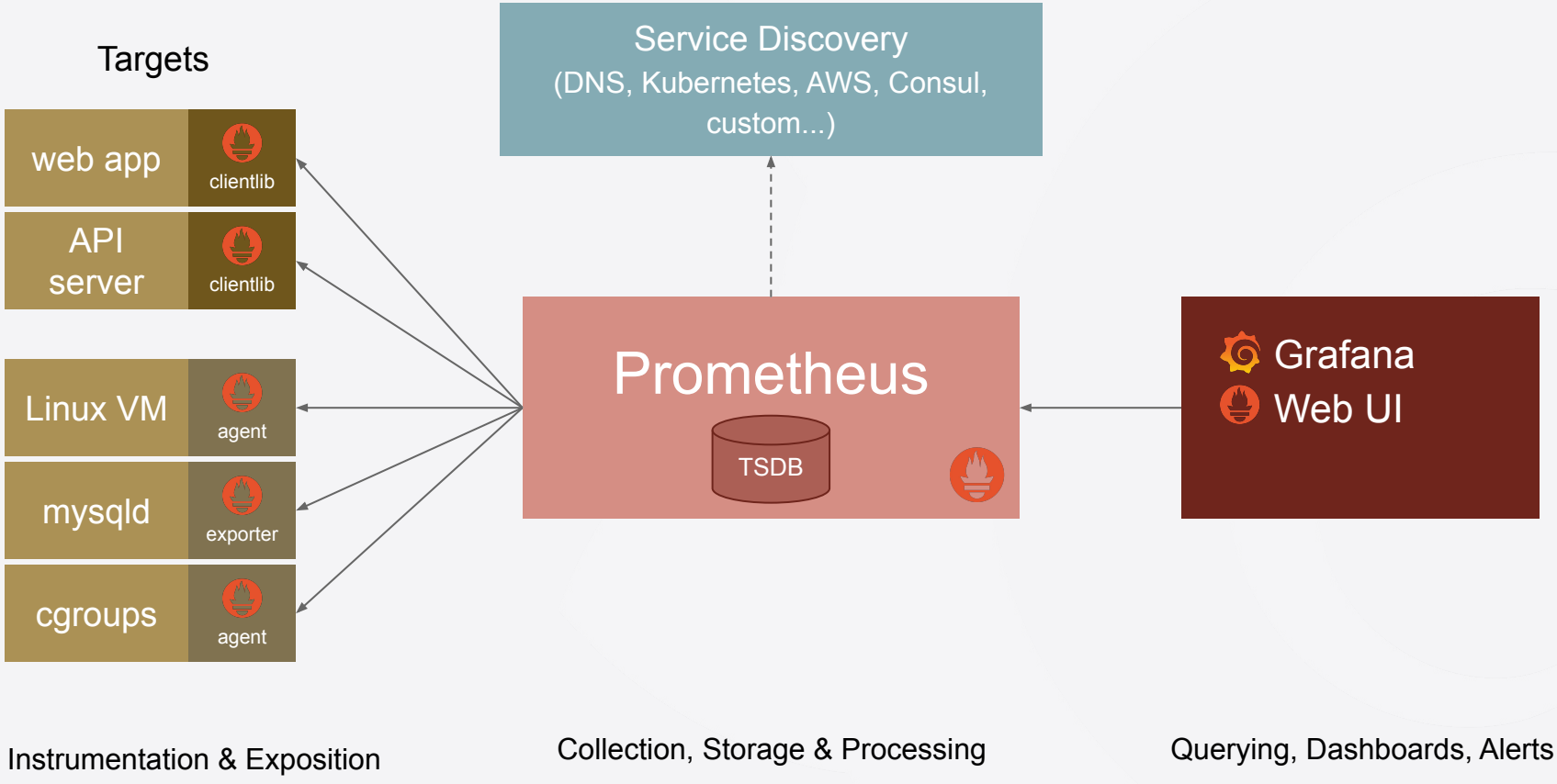
Observability



Pillars of Observability

The Pillars of Observability




Targets
web app
clientlib
API server
clientlib
Linux VM
agent
mysqld
exporter
cgroups
agent
Service Discovery
(DNS, Kubernetes, AWS, Consul, custom...)
Prometheus
TSDB
Grafana
Web UI
Instrumentation & Exposition
Collection, Storage & Processing
Querying, Dashboards, Alerts

Labels

- Labels are **key-value pairs** that define a data stream, noting where data is coming from when it is initially recorded
- **The same label** can be applied to metrics, logs, and traces, enabling you to **query and correlate across multiple data sources**

Mimir

- Open source TSDB[1] that can store metrics from Prometheus, Graphite, and more
- Part of the core Grafana stack of observability applications
- Multi-tenant
- Long-term storage
- Basis for two Grafana Labs commercial offerings
  - Grafana Enterprise Metrics (GEM)
  - Grafana Cloud Metrics

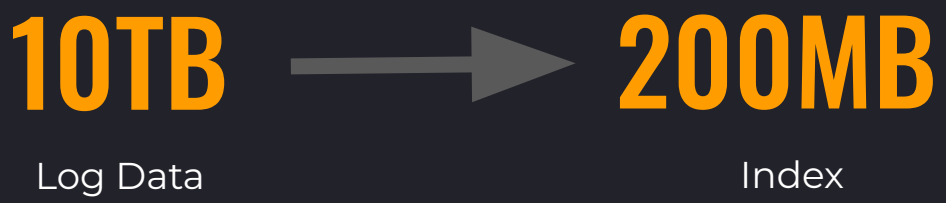
1. https://grafana.com/oss/mimir/

Loki

- Open source log aggregation system
- Allows aggregated logs to be stored in cloud storage, such as AWS S3, Google Cloud Storage, or Cassandra
- Identifies and indexes logs using labels defined by administrators
  - Makes searching for log entries by label very efficient
- Basis for two Grafana Labs commercial offerings
  - Grafana Enterprise Logs (GEL)
  - Grafana Cloud Logs

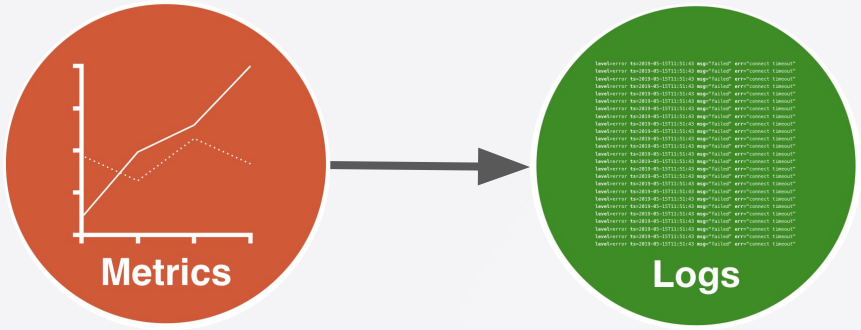
Efficient logging

Loki does not index the text of logs. Instead, entries are grouped into streams and indexed with Prometheus-style labels.



Think of it more like a table of contents than an index

Logging Info is a Great Addition…



Something is happening

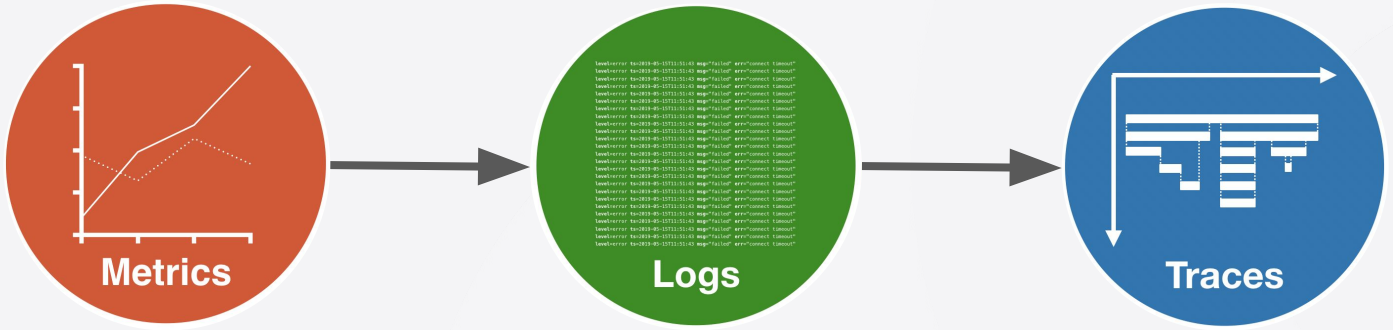
What's happening

Where is it happening

Tempo

- Open source tracing backend
- Can be used with any of the open source tracing protocols, including OpenTelemetry, Jaeger, and Zipkin
- Part of the core Grafana stack of observability applications
- Basis for two Grafana Labs commercial offerings
    - Grafana Enterprise Traces (GET)
    - Grafana Cloud Traces

Tracing Completes the Trinity



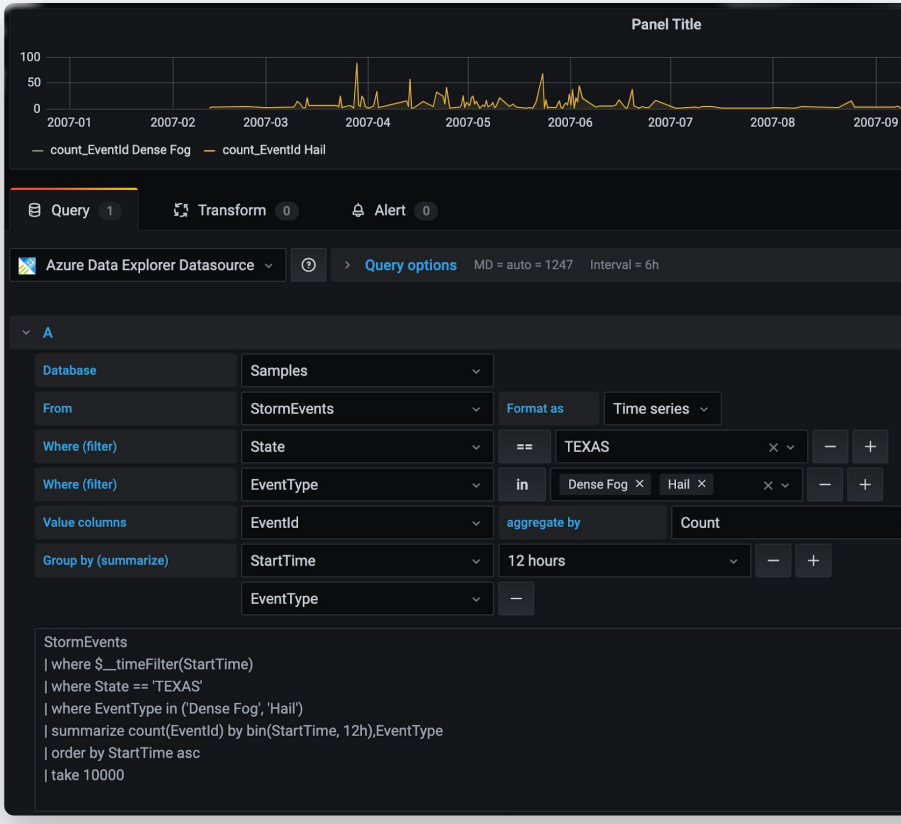
Something is happening

What's happening

Where is it happening

Grafana

- Open source single pane of glass
- Countless data sources
- Integrations
- Unified Platform
- Basis for two Grafana Labs commercial offerings
  - Grafana Cloud
  - Grafana Enterprise



OpenTelemetry

Why OpenTelemetry?

OpenTelemetry ("OTel") gives rich insights into application performance and behavior.

- **Freedom** - OpenTelemetry decouples the instrumentation from the destination. This gives you greater architectural freedom and lower vendor switching costs.
- **Standards** - OpenTelemetry provides standards for an otherwise messy ecosystem of instrumentation and semantics.
- **Open source** - OpenTelemetry development processes are transparent and driven by the open source community.
- **Vendor support** - Many vendors are invested in the OpenTelemetry project and provide support in the form of technology and people.

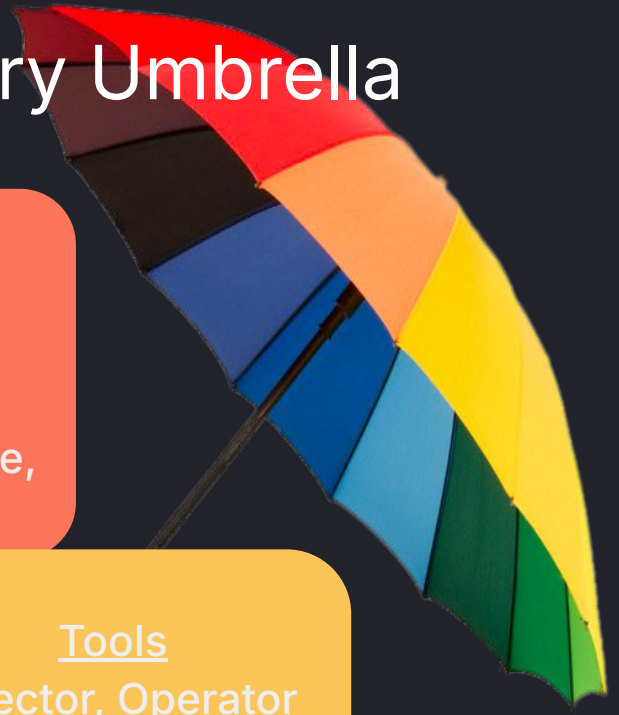
Under The OpenTelemetry Umbrella

Standards
Generic data model, line protocol, and semantic conventions for distributed tracing, metrics, and logs.

Instrumentation
Supports popular libraries and frameworks such as Spring, ASP.NET Core, Express, Quarkus.

API and SDK
API and implementation of the standards in programming languages like Go, Java, dotNet, Python, JavaScript, ...

Tools
Collector, Operator



Why OpenTelemetry Matters?

Vendor-neutral

- No vendor locking
- Freedom of Data

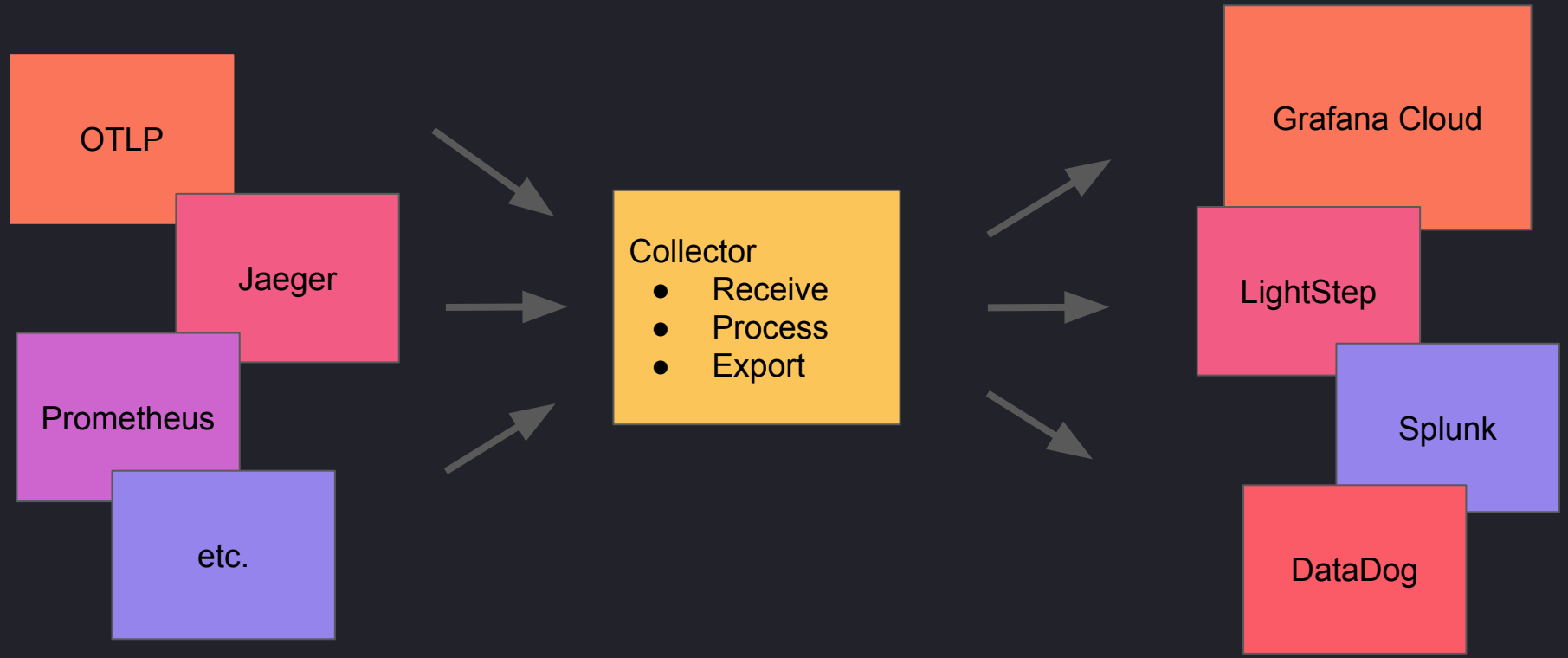
Standard for all Telemetry data

- Unified Protocol
- Seamless tracing across the microservices

Broad Ecosystem

- Open-source
- Community Driven

How the Collector Fits With the Big Tent Philosophy



Collectors ⇒ Grafana Alloy

- Vendor-neutral way of collecting telemetry data
- Ease of data processing
- Performance
- Customizable and extensible
  - Modules
  - Custom Components
- Remote Configuration and Fleet Management
- Support Metrics, Logs, Traces and Cluster Events

Alloy

Kubernetes Monitoring Helm Chart

Grafana ❤ Kubernetes

- Grafana Alloy
  - DaemonSet
  - Deployment
  - StatefulSet
- Kube State Metrics
- Node Exporter
- Open Cost

https://github.com/grafana/k8s-monitoring-helm

Kubernetes Monitoring Helm Chart v2

- Annotation AutoDiscovery
- Application Observability
- Auto Instrumentation
- Cluster Events
- Integrations
- Node Logs
- Pod Logs
- Profiling
- Prometheus Operator
- Tail Sampling

DEMO

Wrapping Up

- Prometheus, Grafana, Loki, Tempo for a solid o11y foundation.
- OpenTelemetry fills the tracing gap and more with collectors.
- K8s-monitoring Helm Chart with Grafana Alloy
- A unified telemetry
- A unified Grafana Platform
- Improves incident response.

Thank you!

btw. we're hiring!

Grafana Labs