
cisco Live!
Let's go

The bridge to possible

Unlock Intersight monitoring with OpenTelemetry

Chris Gascoigne, Principal Architect
@chrisgascoigne@hachyderm.io

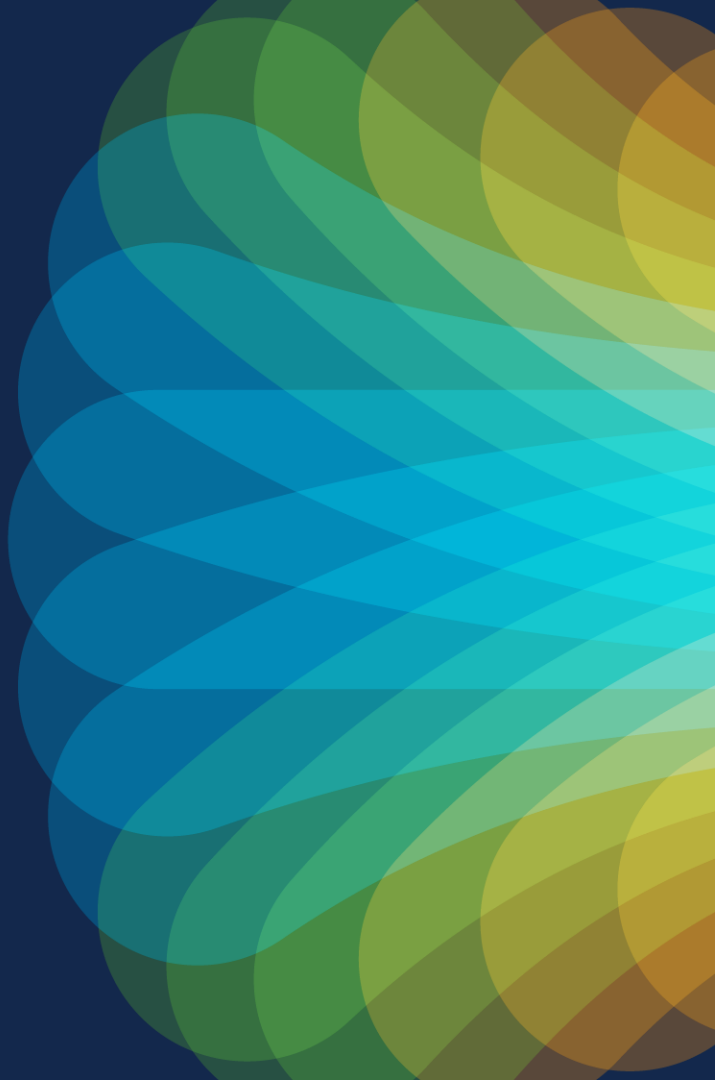
cisco Live!

DEVNET-2118

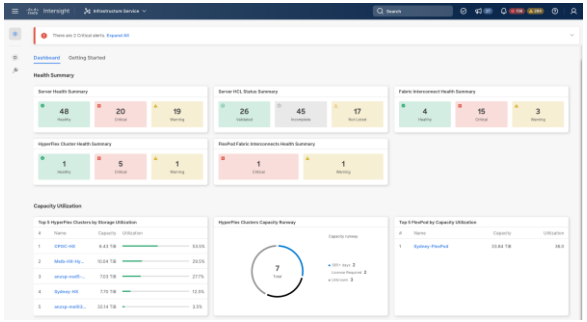
Agenda

- The Challenge
- Enter OpenTelemetry
- Intersight and OpenTelemetry?
- Key Takeaways

The Challenge



Silo’d data stores and dashboards

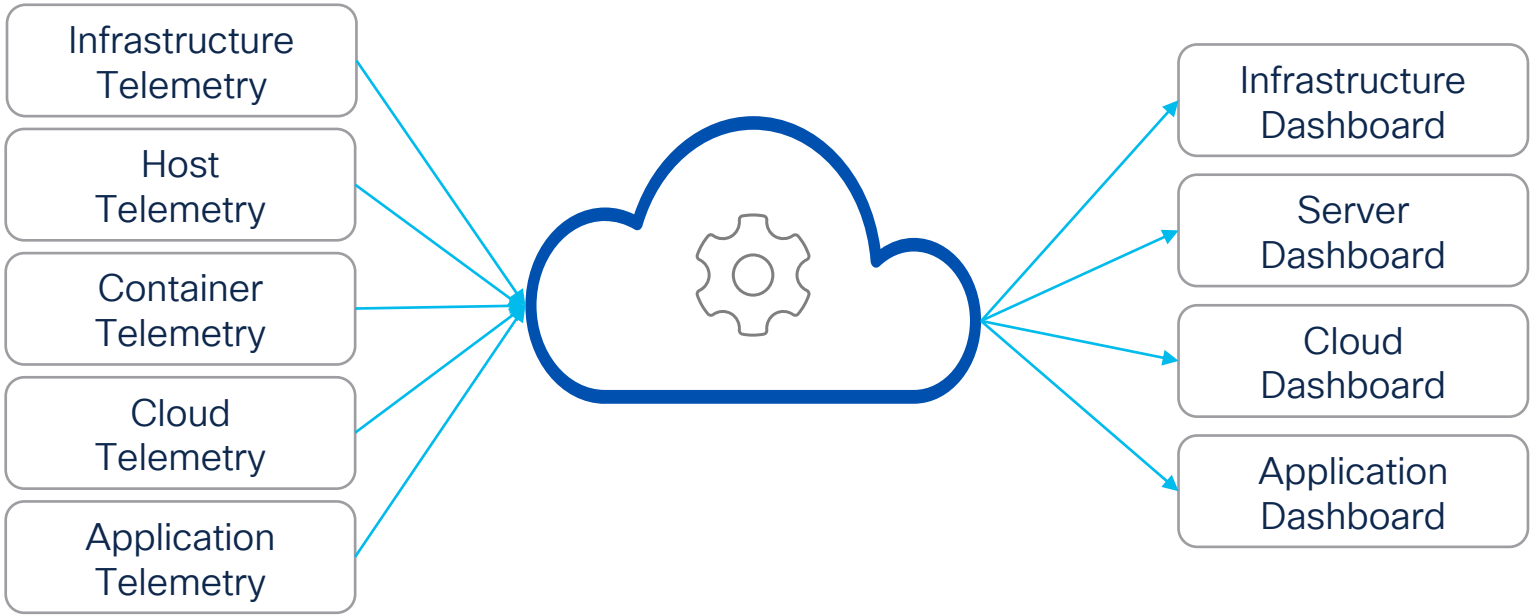


Infrastructure team

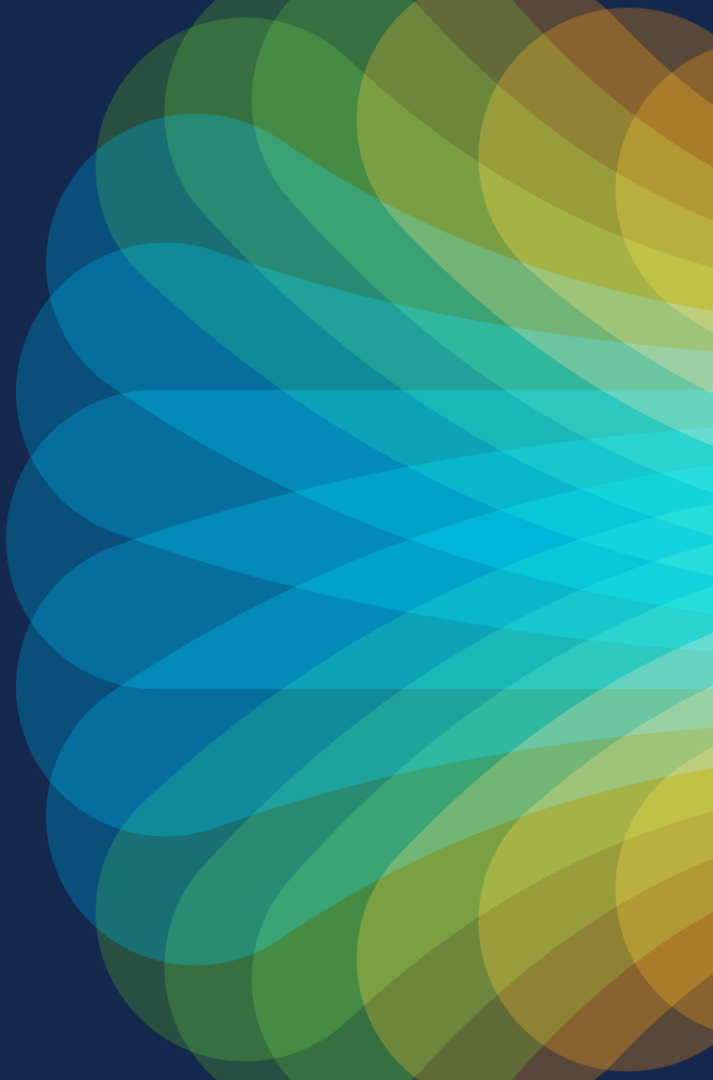
Virtualization / Systems team

Cloud team

What if there was a way to democratize that data?



Enter OpenTelemetry



What is OpenTelemetry?

“OpenTelemetry is a collection of tools, APIs, and SDKs. Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software’s performance and behavior.”

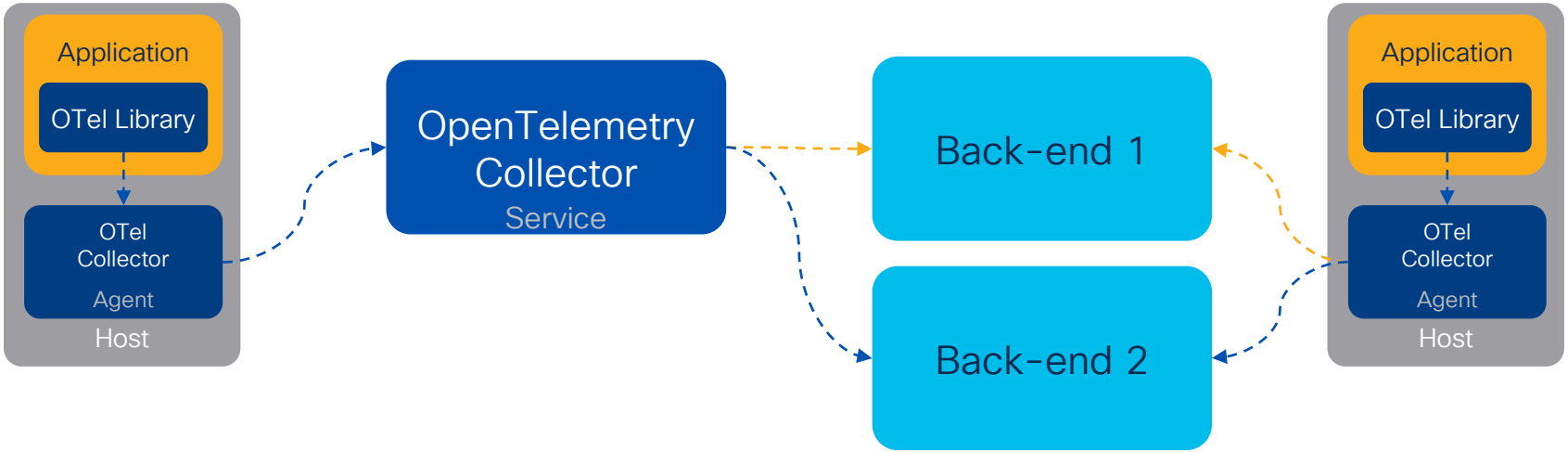
OpenTelemetry Signal Types

Metrics	a set of measurements collected at regular intervals
Logs	strings of structured or unstructured text with an associated timestamp
Traces	chains of events (or transactions) between different components in an application

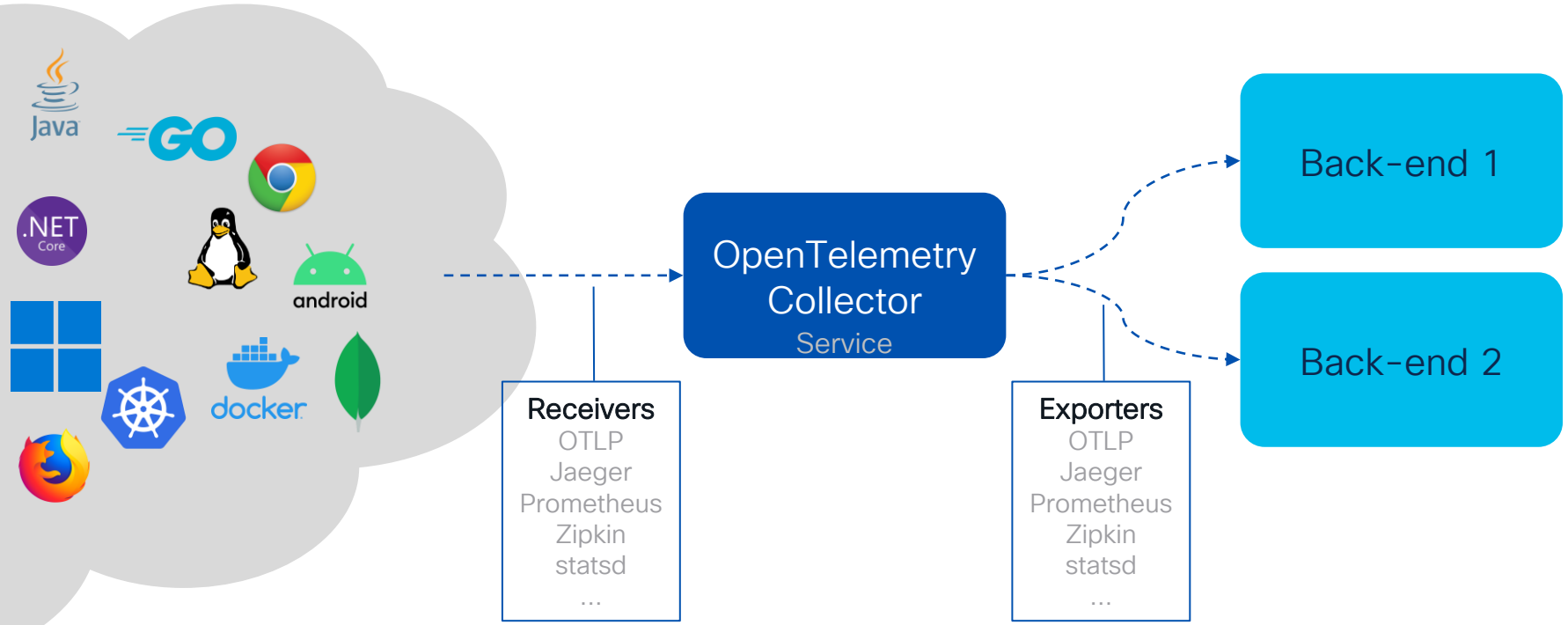
Relevant OpenTelemetry Terminology

API	In the OpenTelemetry project, used to define how telemetry data is generated
SDK	A telemetry SDK implements the OpenTelemetry API in a specific language
Semantic Conventions	Defines standard names and values of Metadata in order to provide vendor-agnostic telemetry data.
OTLP	OpenTelemetry Protocol (OTLP) defines the protocol used to exchange data between the client and the server over gRPC or HTTP

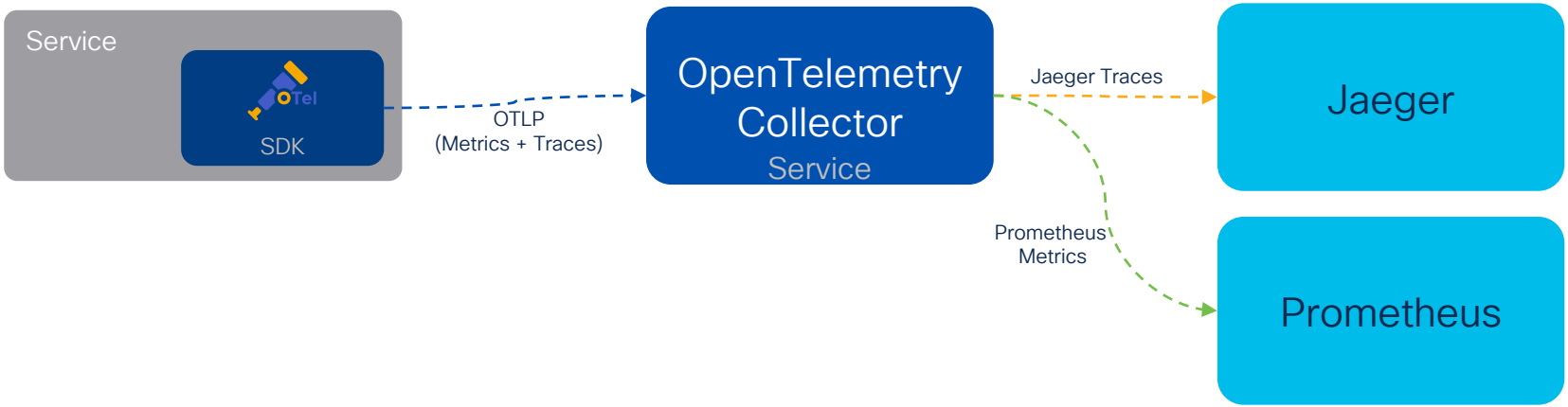
OpenTelemetry reference Architecture



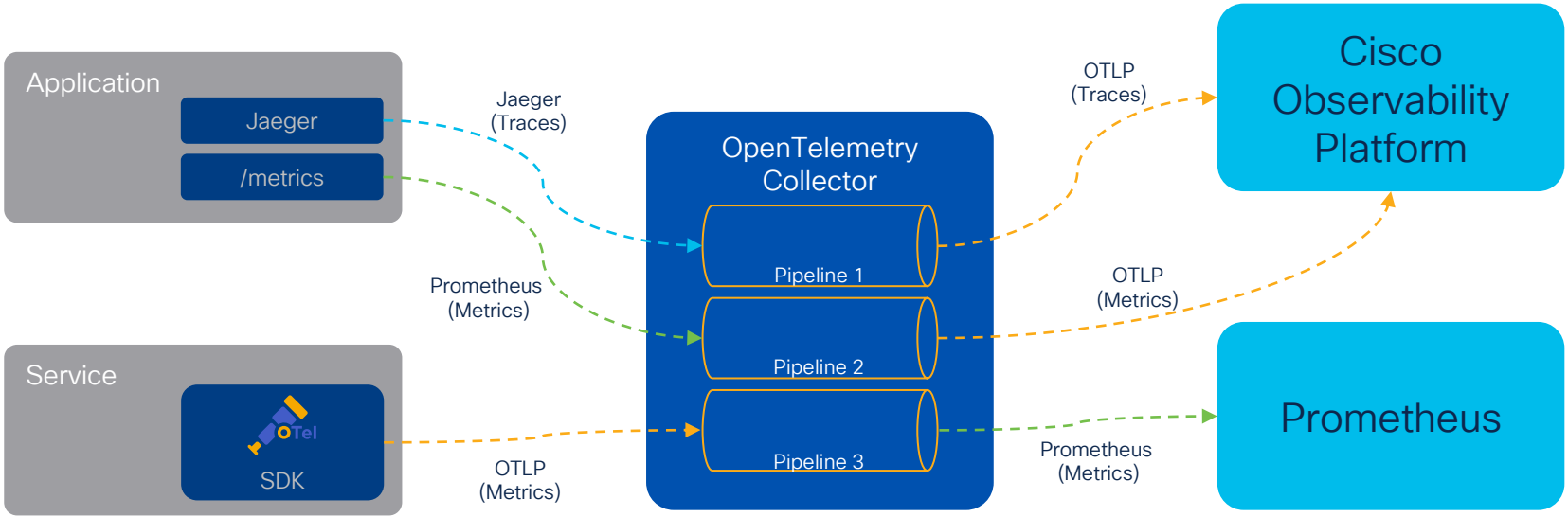
OpenTelemetry Collector as protocol translator



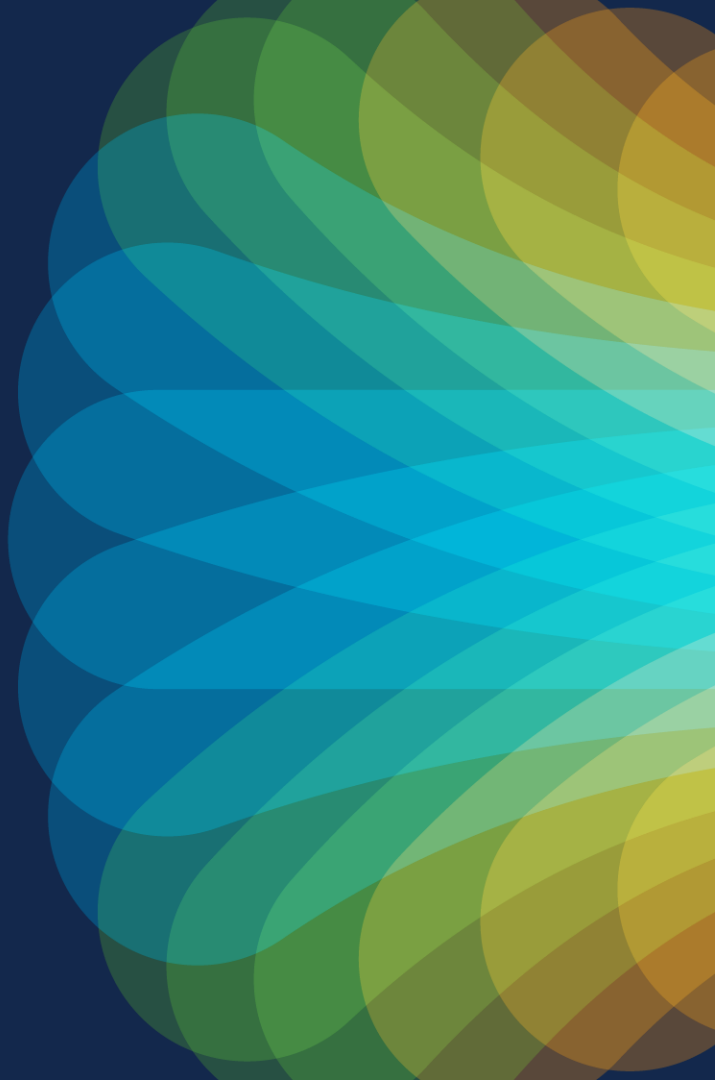
Collector Use Cases (1)



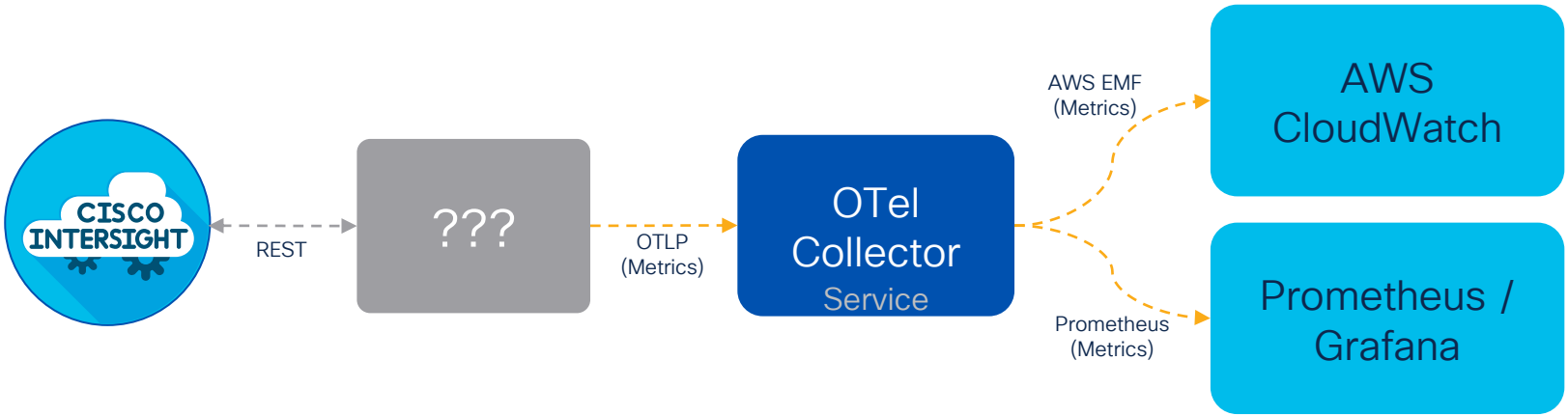
Collector Use Cases (2)



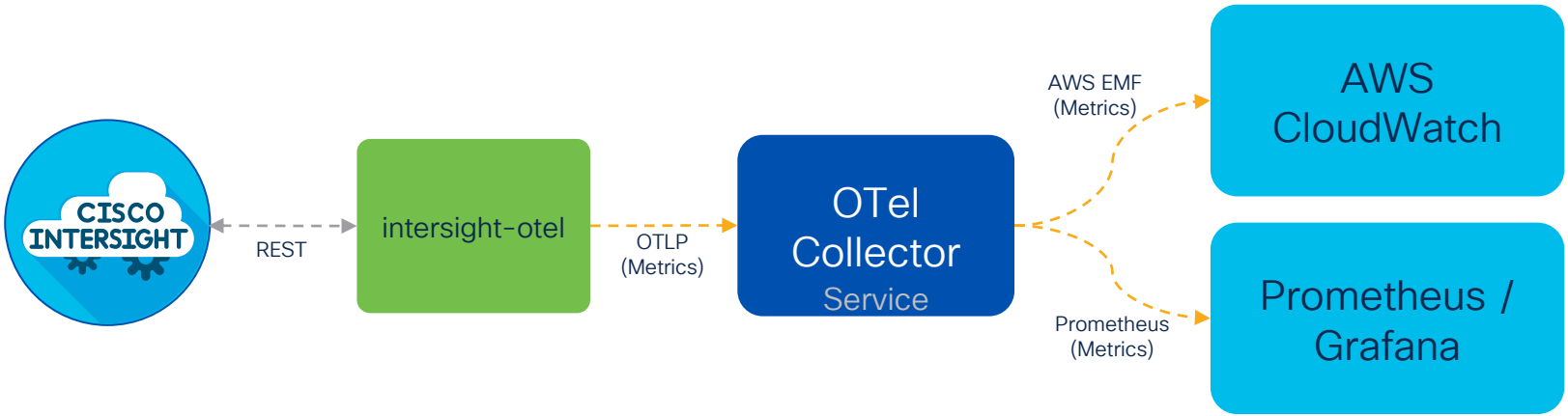
Intersight and OpenTelemetry?

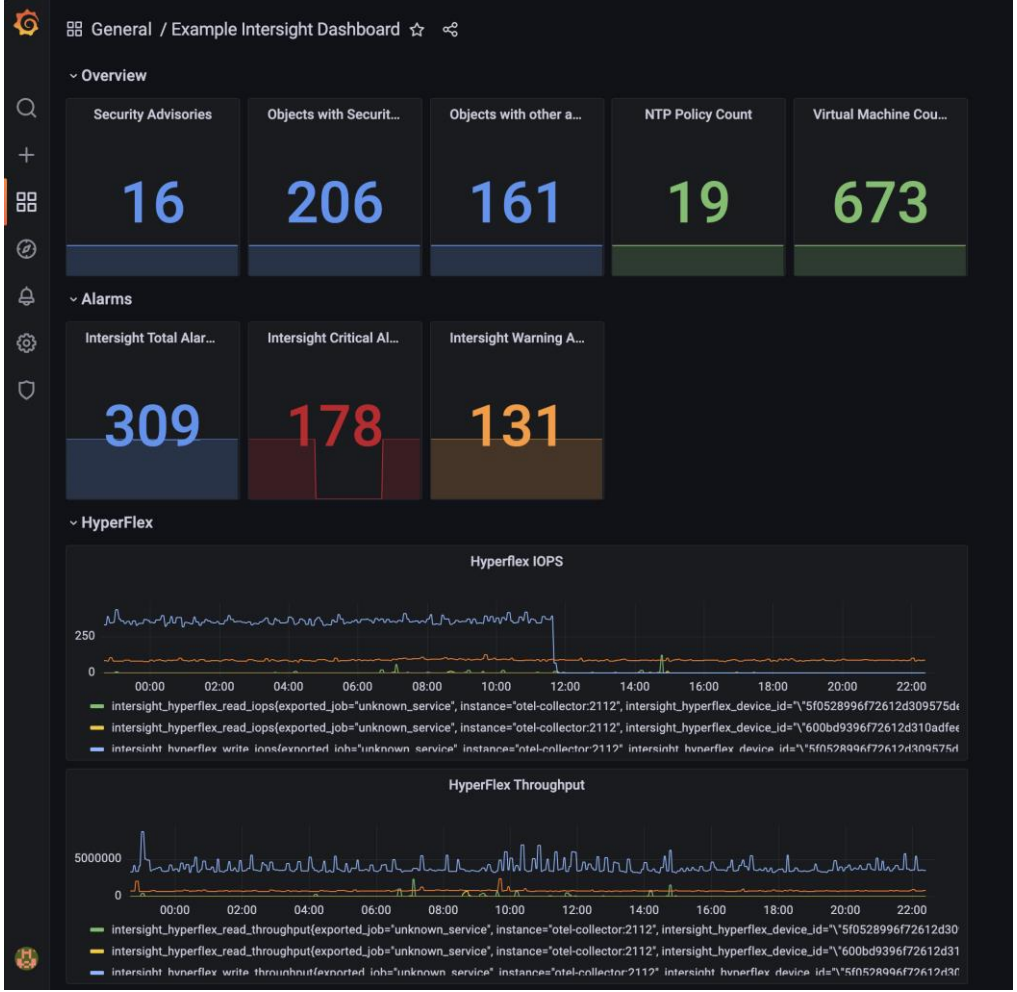


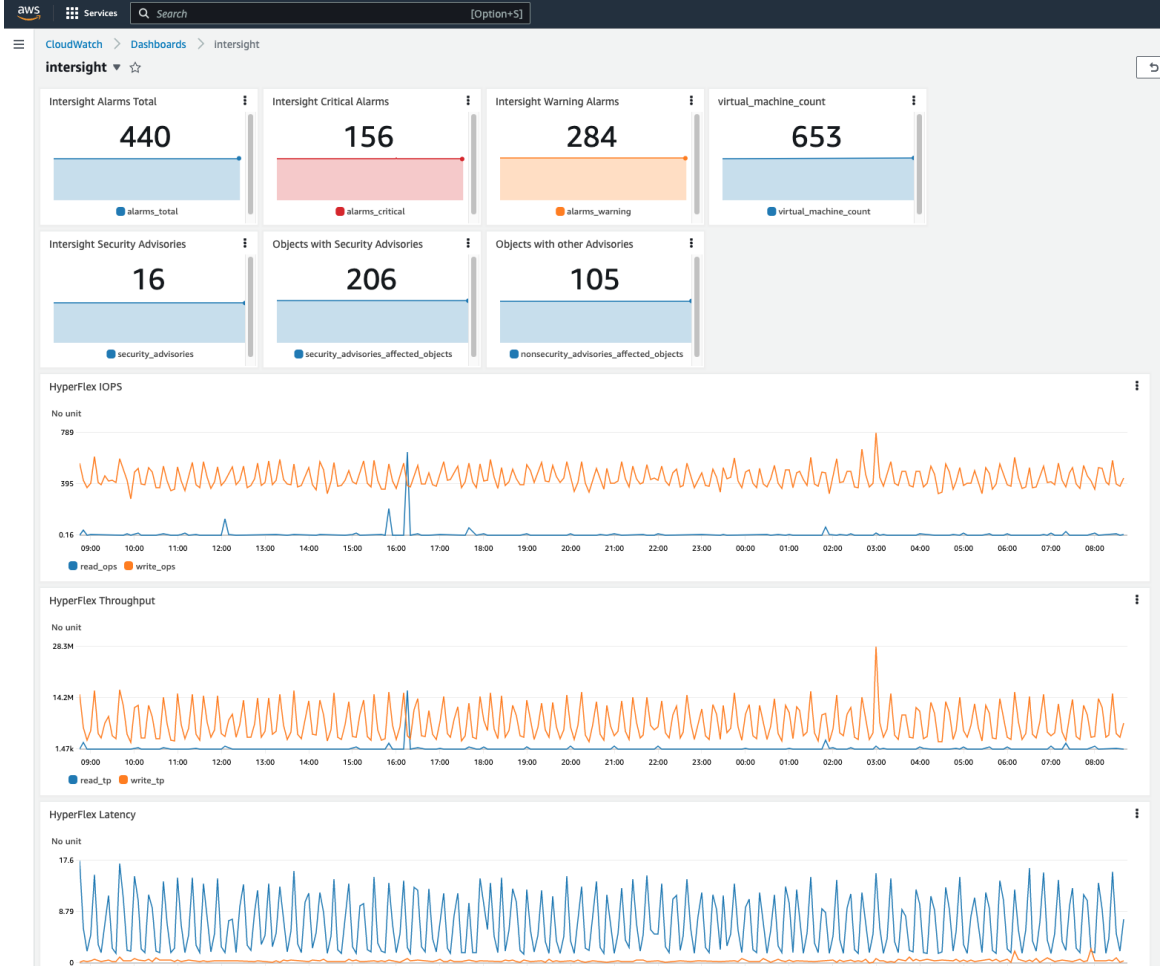
Intersight and OpenTelemetry?

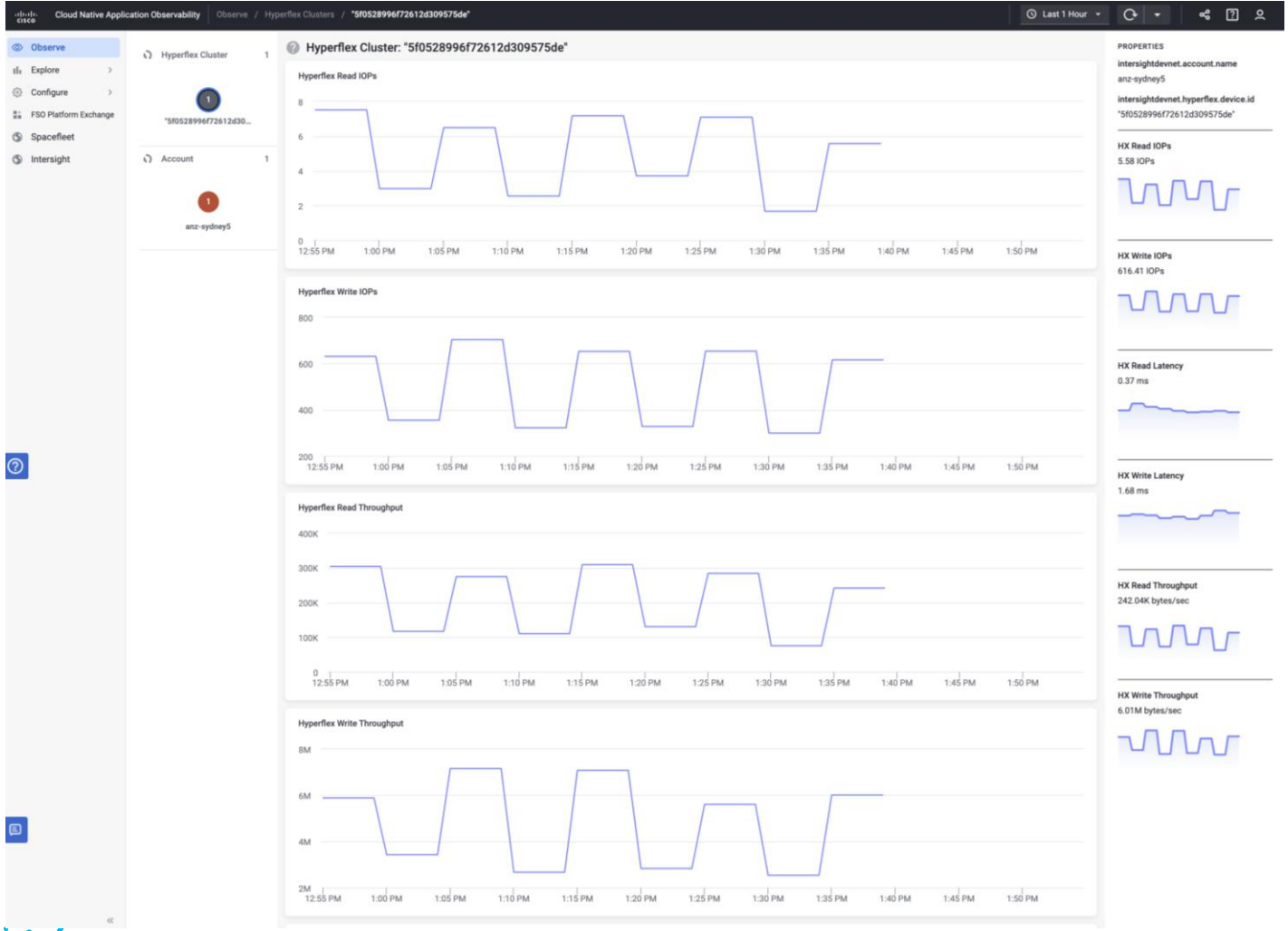


Intersight and OpenTelemetry









What is intersight-otel?

- Lightweight collector (written in Rust)
- Periodically polls Intersight API to retrieve metrics
- Extremely configurable – API requests and metrics generated are configuration not code
- Available as binaries for multiple platforms or container image to run on Docker, Kubernetes, AWS ECS, etc.
- https://github.com/cgascoig/intersight-otel

Configuration overview

```
key_file = "/etc/ismetricskey/intersight-key.pem"
key_id = "0123456789abcdef01234567/0123456789abcdef01234567/0123456789abcdef01234567"
otel_collector_endpoint = "http://otel-collector:4317"

[[pollers]]
name = "virtual_machine_count"
api_query = "api/v1/virtualization/VirtualMachines?$count=true"
aggregator = "result_count"
interval = 300

[[pollers]]
name = "ntp_policy_count"
api_query = "api/v1/ntp/Policies?$count=true"
aggregator = "result_count"
interval = 300

[[tspollers]]
name = "hx_performance"
datasource = "hx"
dimensions = ["deviceId"]
filter = { ...}
aggregations = [...]
post_aggregations = [...]
interval = 300
```

Global configuration

- Intersight auth
- Otel destination

Regular Pollers

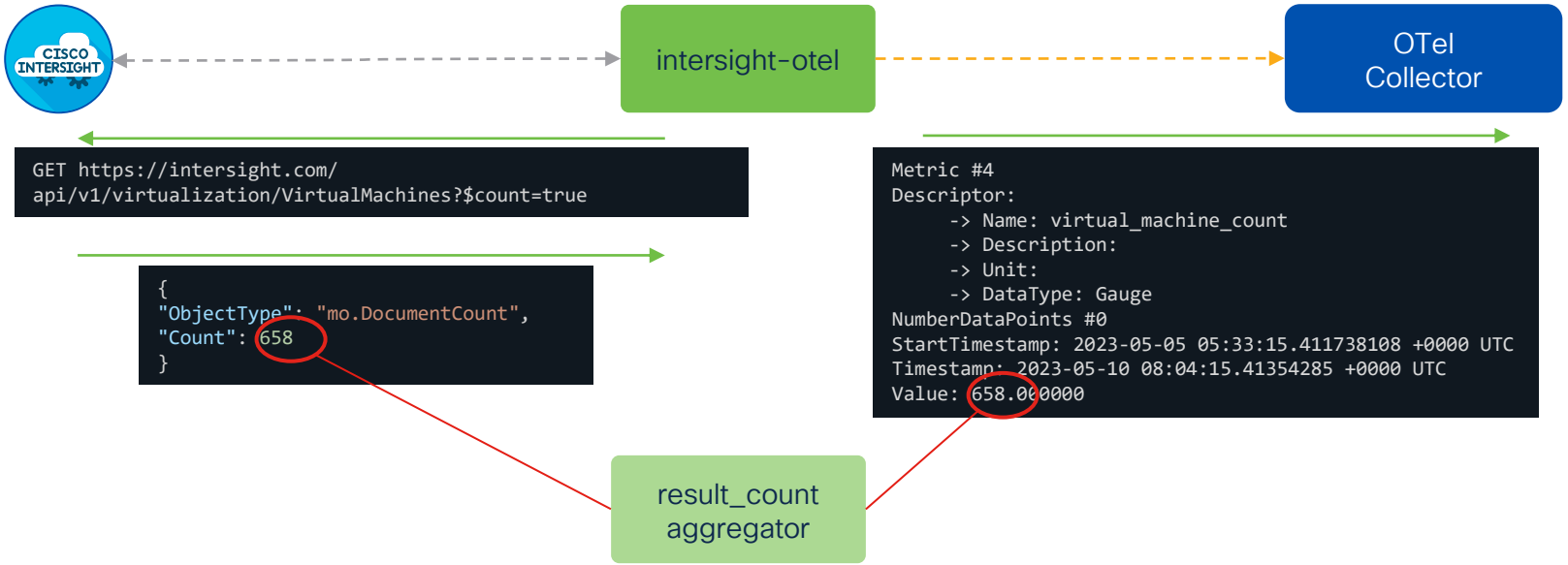
Call a regular Intersight API and extract a metric value

TimeSeries Pollers

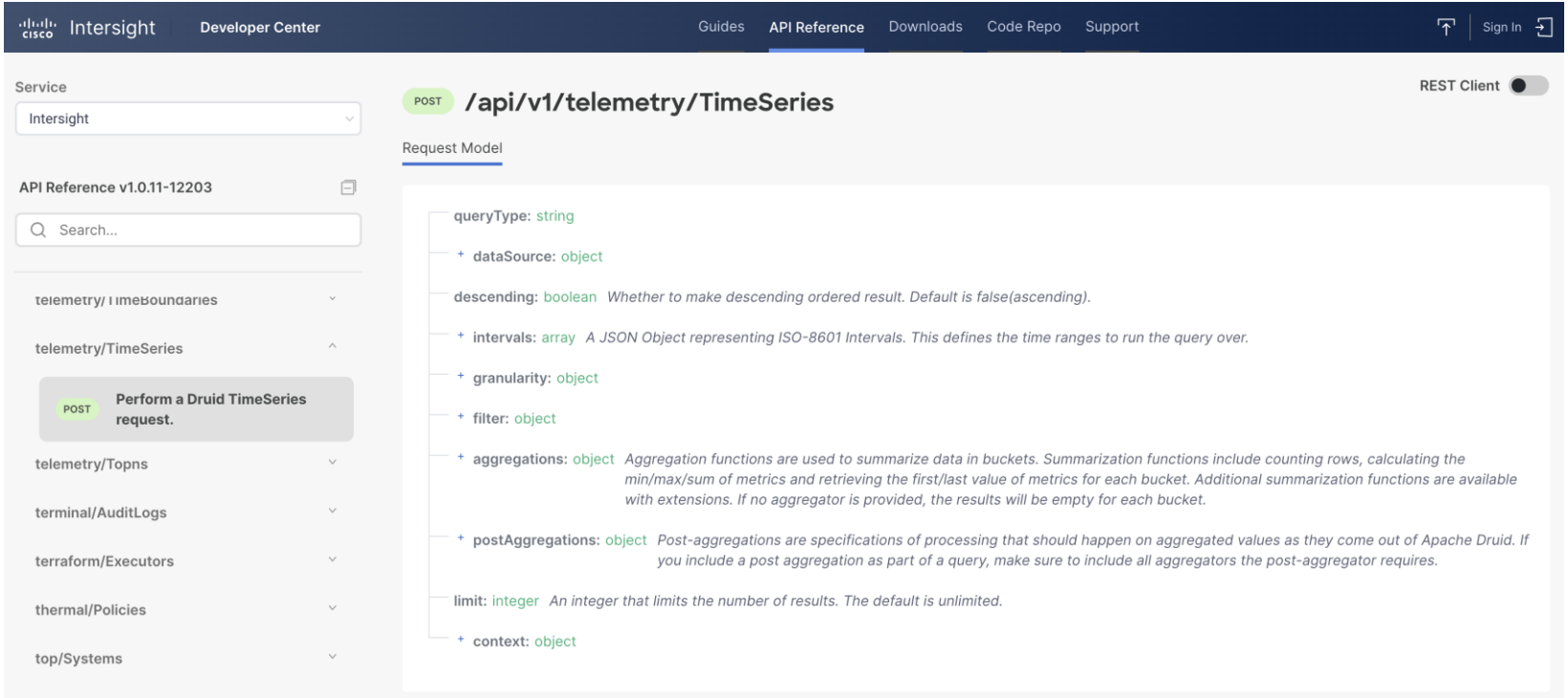
Query the Intersight “TimeSeries” API for metrics

Regular Pollers in action

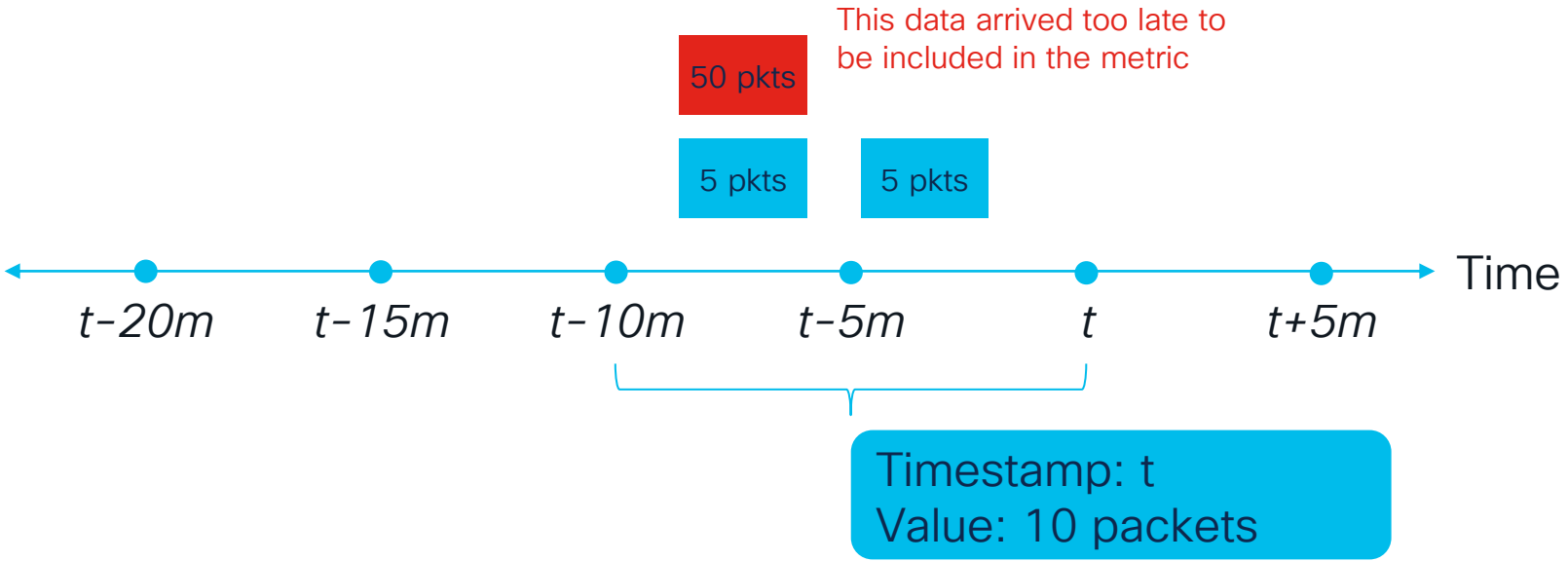
```
[[pollers]]
name = "virtual_machine_count"
api_query = "api/v1/virtualization/VirtualMachines?$count=true"
aggregator = "result_count"
interval = 300
```



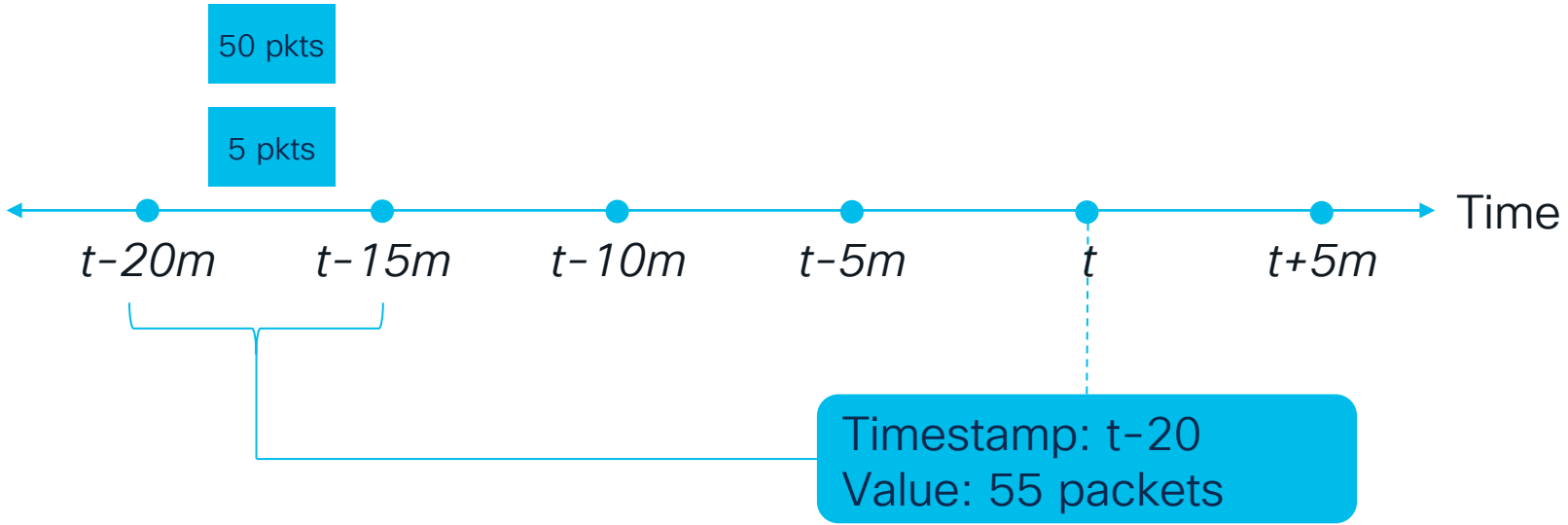
TimeSeries Pollers - API



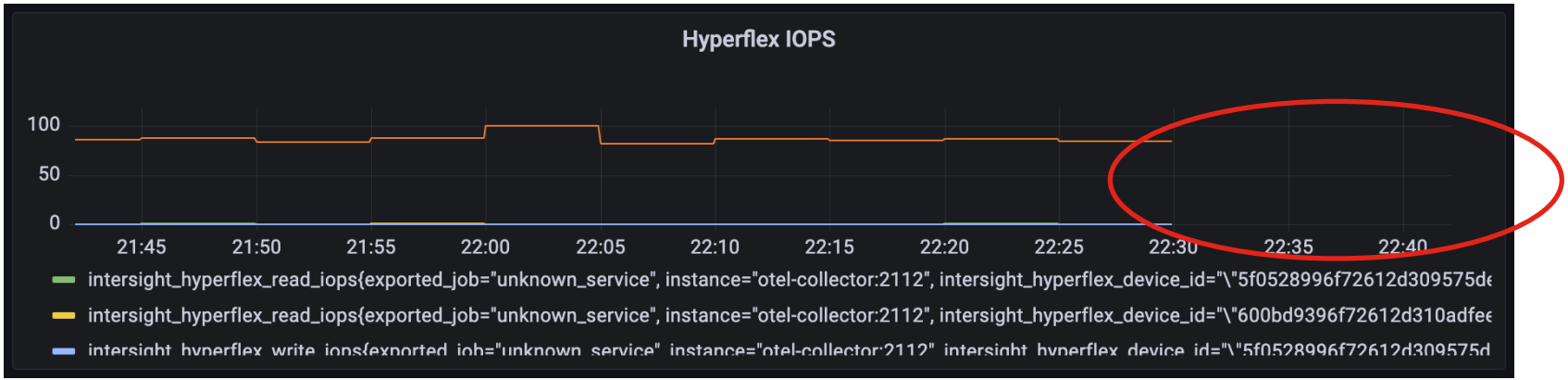
TimeSeries Pollers – Naïve strategy



TimeSeries Pollers – Actual strategy

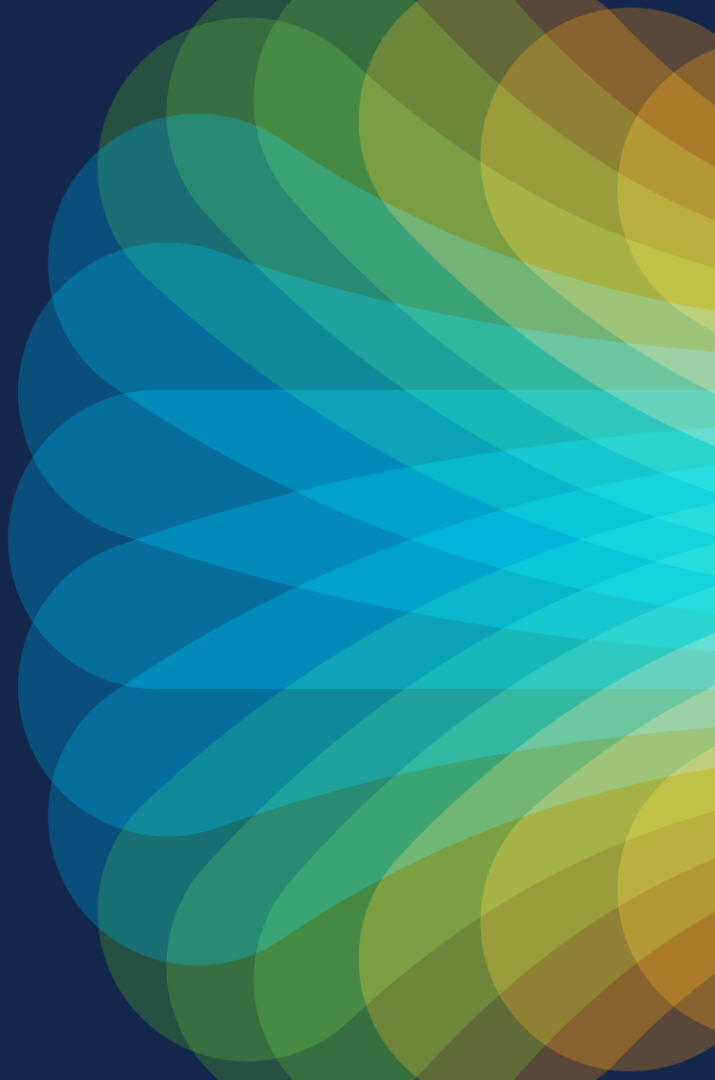


TimeSeries Pollers



Demo time!

Key Takeaways



Key Takeaways

- OpenTelemetry provides a vendor-agnostic telemetry framework
- Once in OpenTelemetry we can do almost anything with that data
- Think of OpenTelemetry Collector as a protocol translator for telemetry
- *intersight-otel* can generate OpenTelemetry metrics from almost any Intersight API call you can think of
  - Please try it out
  - Give feedback
  - “Star” / “Watch” the GitHub repository for updates


cisco
The bridge to possible
Thank you
cisco Live!


cisco Live!
Let's go