Observability v .NET pomocí OpenTelemetry

Tomáš Jecha

Head of Engineering at cbData

LinkedIn /in/jechtom | X @jechtom

Observability Signals

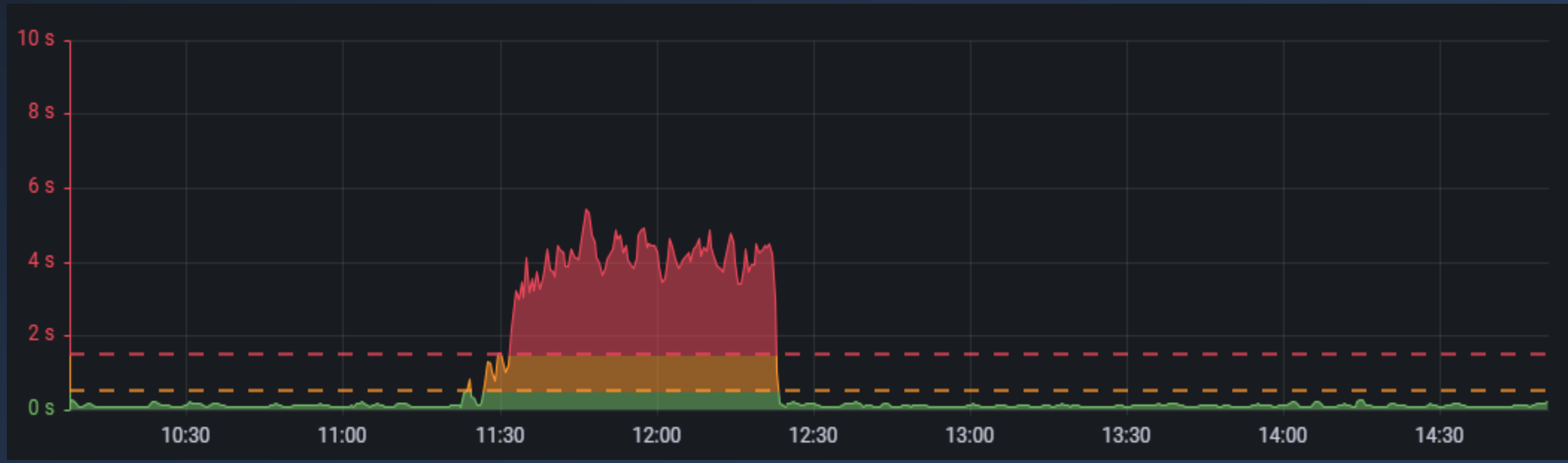
- Logs
- Metrics
- Tracing

Observability Signals – Logs

03 Mar 2024 17:21:29.094	OTelDemo.Web	Privacy page visited
03 Mar 2024 17:21:28.550	OTelDemo.Web	Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeout='30'] SEL...
03 Mar 2024 17:21:28.550	OTelDemo.Web	received-first-response
03 Mar 2024 17:21:28.548	OTelDemo.Web	End processing HTTP request after 64.2179ms - 200
03 Mar 2024 17:21:28.548	OTelDemo.Web	Received HTTP response headers after 64.0337ms - 200
03 Mar 2024 17:21:28.537	OtelDemo.Backend	Got weather forecast
03 Mar 2024 17:21:28.484	OTelDemo.Web	Sending HTTP request GET http://localhost:4006/WeatherForecast
03 Mar 2024 17:21:28.484	OTelDemo.Web	Start processing HTTP request GET http://localhost:4006/WeatherForecast
03 Mar 2024 17:21:28.484	OTelDemo.Web	Done
03 Mar 2024 17:21:28.477	OtelDemo.Backend	Getting weather forecast
03 Mar 2024 17:21:28.437	OTelDemo.Web	Part way there
03 Mar 2024 17:21:28.381	OTelDemo.Web	Index page visited

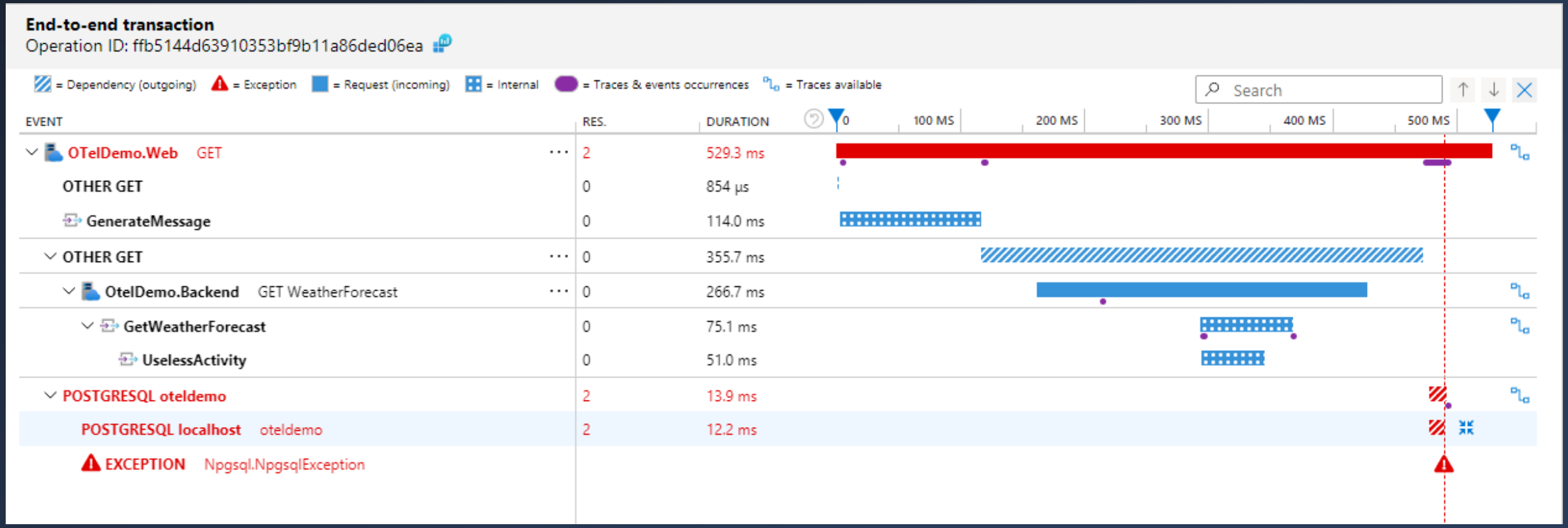
SEQ logging UI (proprietary)

Observability Signals – Metrics



Grafana dashboard with Prometheus data (open source)

Observability Signals – Tracing



Azure Application Insights (proprietary)

What to Choose for Your Observability Stack?

Prometheus	Azure Monitor	DynaTrace	DataDog	Seq	Grafana	ElasticStack
Jaeger	Zipkin	Splunk	New Relic	Logstash	Fluentd	AWS CloudWatch
Google Cloud Ops	Betterstack Logs	Telegraf	Sentry	Loggly	Honeycomb	Rollbar
Loki	Graphite	InfluxDB	Tempo	OpenTSDB	Graylog	…..

What is OpenTelemetry?

- Protocol – serialization and transport (metrics, logs, tracing)
- Semantic rules – naming and well known attributes
- APIs – interfaces for instrumentation
- SDKs – implementation of the APIs that are used to configure and operate the telemetry collection
- Ecosystem of libraries – instrumentation, exporters, ...
- Massive support by the industry – see https://opentelemetry.io/ecosystem/vendors/

OpenTelemetry Protocol (OTLP)

- OTLP/gRPC (Protobuf) or OTLP/HTTP (JSON)
- Protocol specs and protobuf definitions at https://github.com/open-telemetry/opentelemetry-proto
- Defines services:
  - Logs collector
  - Metrics collector
  - Trace collector

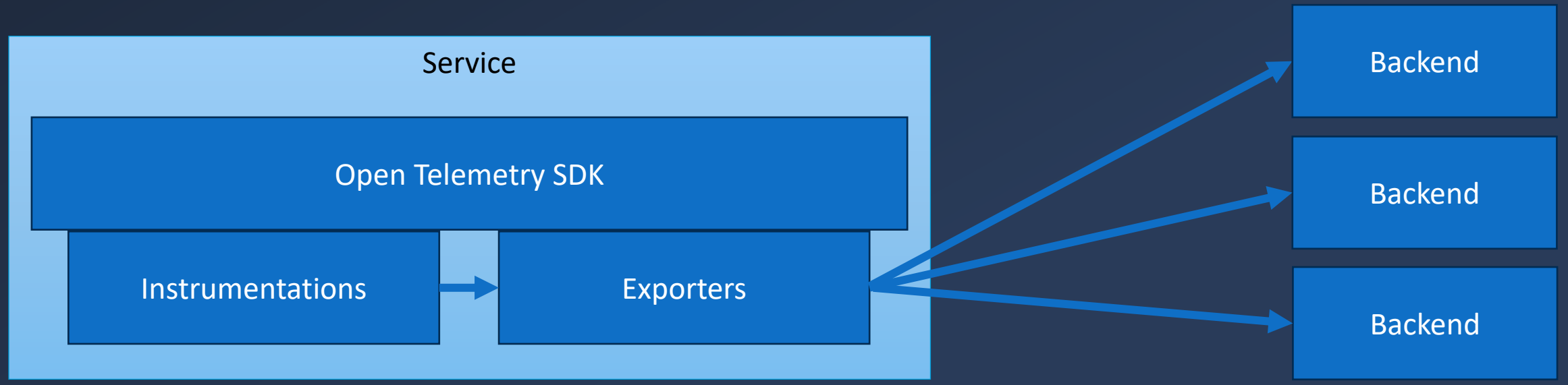

DEMO
Under the Hood: OTLP + Serilog

OpenTelemetry API & .NET

- Logs: Connects to Microsoft.Extensions.Logging
  - ✅ Widely used and integrates well with OpenTelemetry
- Metrics: Subscribes to System.Diagnostics.Metrics
  - ✅ Since .NET6, designed to integrate well with OpenTelemetry
  - Replaces EventCounters (since .NET Core 3) and PerformanceCounters (Win only) – see https://learn.microsoft.com/en-us/dotnet/core/diagnostics/compare-metric-apis
- Traces: Integrates System.Diagnostics.Activity
  - ✅ Widely used and integrated directly to OpenTelemetry
  - W3C Trace Context HTTP headers – already implemented in .NET – see https://learn.microsoft.com/en-us/dotnet/core/diagnostics/distributed-tracing-concepts

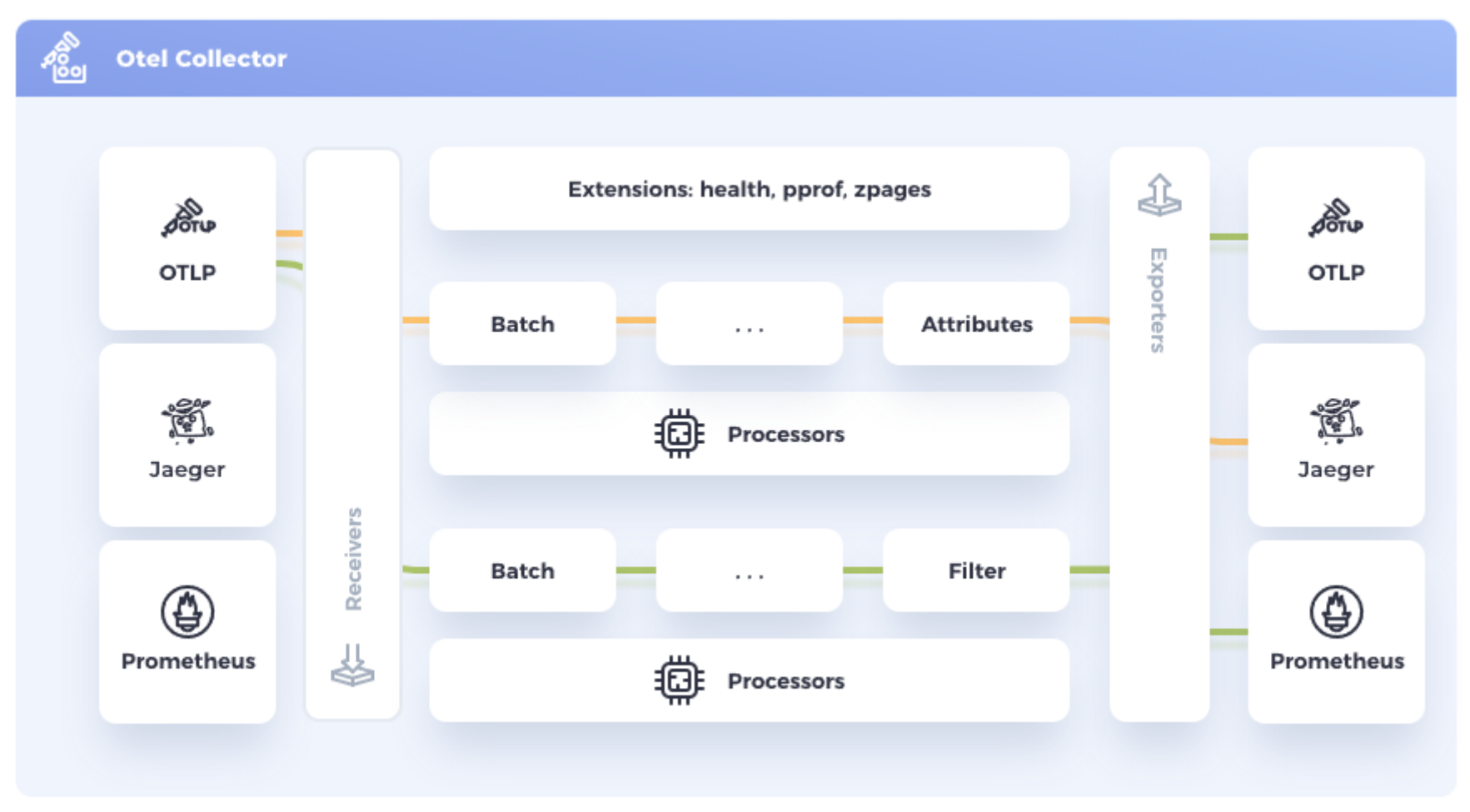
OpenTelemetry Libraries

- Registry: https://opentelemetry.io/ecosystem/registry
- Instrumentation libraries – generates relevant telemetry data
- Exporter libraries – sends telemetry (via OTLP or other protocols)




DEMO
OpenTelemetry SDK:
Configuration, Instrumentation, Exporters

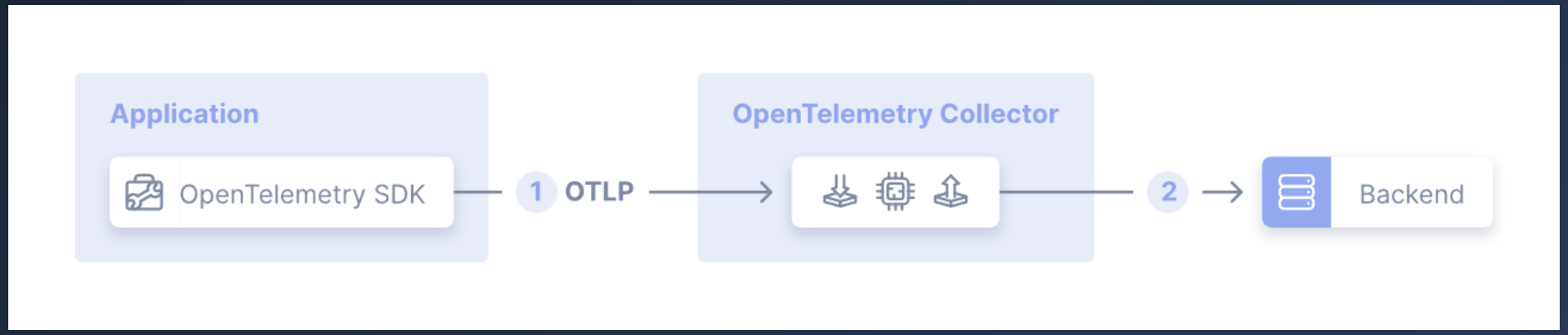
OpenTelemetry Collector

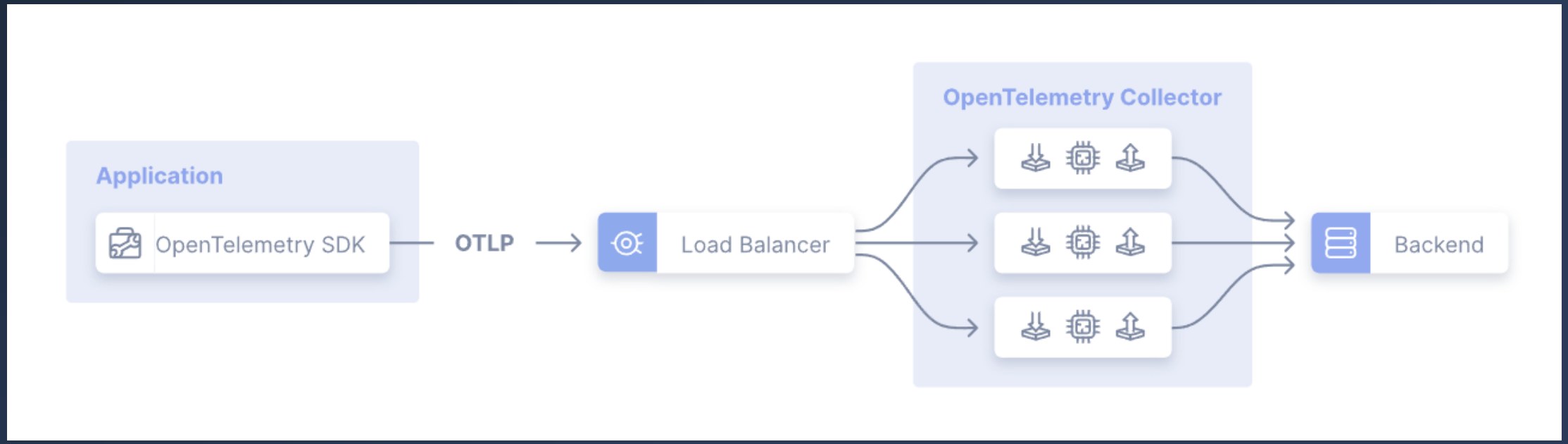


OpenTelemetry Collector

- Receive, process and export telemetry data
- https://opentelemetry.io/docs/collector/
- Alternatives: Logstash, Fluentd, Telegraf (InfluxDB), …
- Built-in receivers/processors/exporters/connectors:
  - Core: https://github.com/open-telemetry/opentelemetry-collector
  - Contrib: https://github.com/open-telemetry/opentelemetry-collector-contrib

Collector Deployment Models






DEMO
OpenTelemetry Collector

Zero-code Instrumentation for .NET

- https://opentelemetry.io/docs/languages/net/automatic/
- Steps:
  1. Install auto-instrumentation (once)
  2. Run `.otel-dotnet-auto/instrument.sh`
  3. Configure with env variables (OTEL_EXPORTER_OTLP_ENDPOINT, etc.)
  4. Run your app/service

- Works like *magic** 🦄 🌈

**magic* is limited to .NET 6+


DEMO
Zero Code Instrumentation

Pros and Cons

😔 Still relatively young
- Some packages are still in preview and relatively untested
- Many features are in the experimental phase

😔 Very broad scope (SDKs and libraries)
- Not everything will be perfect
- It will be difficult to keep the project on track

😊 First vendor agnostic single-wire telemetry protocol

😊 Intensive progress made in the last 2 years
- Strong support from the community and solution providers
- We will start testing it in production soon

https://github.com/jechtom/demo-2024-open-telemetry

Tomáš Jecha

LinkedIn /in/jechtom | X @jechtom