Progress®

OpenTelemetry Tracing and Metrics

Workshop: Observability in OpenEdge

Cameron Wright and Anil Kotha

PUG Challenge Waltham

September 2024

Your Workshop Hosts

Cameron Wright

Customer Solution Architect, Progress

Anil Kotha

Principal Software Engineer, Progress

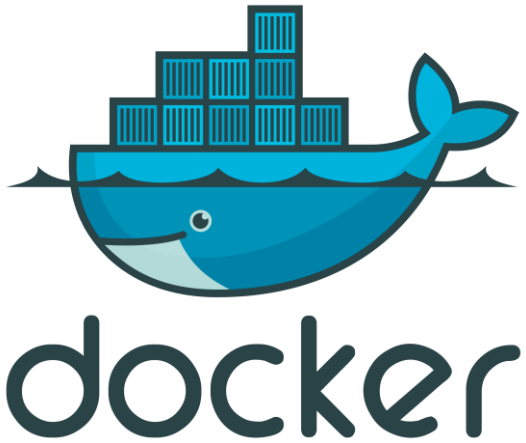
Agenda

9:00 – 9:15	**Introduction and house keeping**
9:15 – 9:30	**Observability**
9:30 – 10:00	**OpenTelemetry Tracing**
10:00 – 10:15	Break
10:15 – 11.00	**Labs**
11:00 – 11.15	**OpenTelemetry Metrics**
11:15 – 11:45	**Labs (plus extra when time left)**
11:45 – 12:00	**Round-up**

Introduction & Housekeeping

Today’s Workshop stack

DB / PASOE / OECC



Workshop Materials

- https://github.com/rwdroge/pug2024OTEL
  - Code
  - Config files
  - PDSOE Project
  - Instructions (ReadMe / PDF)

Workshop Machines

- You can use your own machine, prerequisites are noted in the ReadMe
- We have prepared some cloud VM’s for everyone with most essentials installed:
  - Connection details will be distributed

Observability

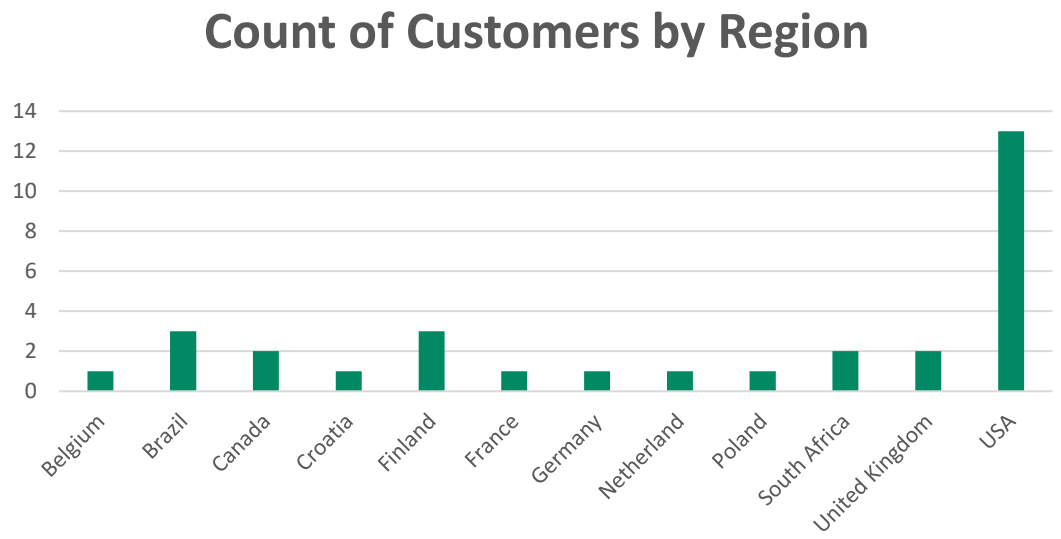
Application Monitoring Needs

OpenEdge stack is increasingly part of a Larger IT eco-system. You have complex IT portfolios end-to-end, using intelligent Enterprise monitoring Tools to help address Use Cases like -

- Performance optimization
- Outage avoidance and problem isolation
- Service level management
- Infrastructure optimization
- Capacity planning

2021 OpenEdge Survey results reveal fragmented adoption of performance monitoring tools with –

- Nagios as free tool
- Small Shops using home-grown Tools (32 out of 172 responded)
- Big Shops using market-leading APM Tools like Dynatrace, NewRelic, Datadog etc.,



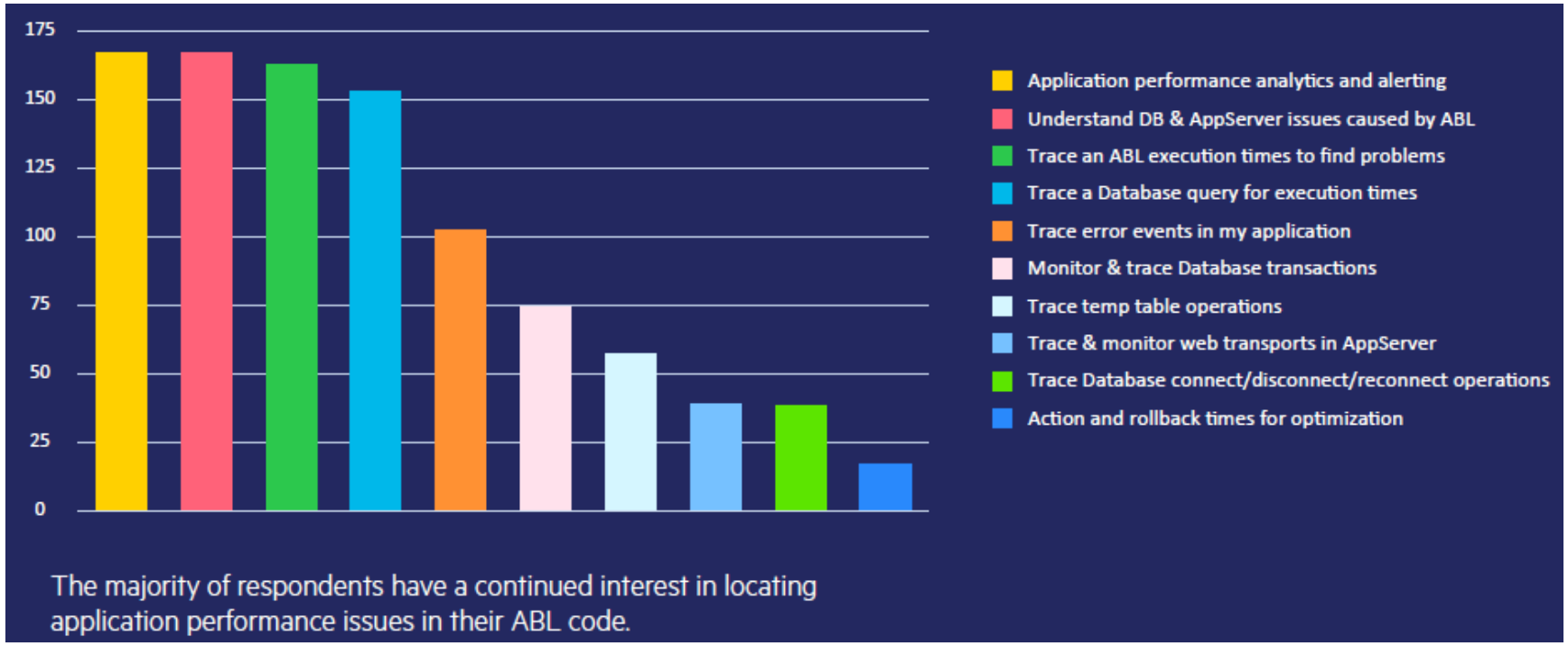
Market Problems

OpenEdge 2022 Survey Outcomes revealed that **OpenEdge Application Performance Analytics** and **Alerting, understanding DB & AppServer issues caused by ABL** are the most important for you.

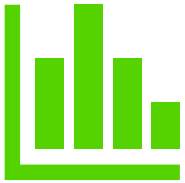
- Application outage avoidance and problem isolation - OpenEdge lacking an intelligent dashboard with email alerts facility.
- Need 4GL programs monitoring - utilization of system resources by 4GL.
- Lack of actionable insights when monitoring PASOE & Database.

Market Evidence

OpenEdge 2022 Survey Outcomes: Distributed Tracing for OpenEdge Application Monitoring



3 Pillars of Observability



Metrics

Time series of data points related to the **health of the IT Service**

Traces

Insight into **where the time is spend** delivering the IT Service

Logs

A **track record of event occurrences** meaningful to the IT Service Assurance

Gartner says **By 2025, 50% of new cloud-native application monitoring will use open-source instrumentation**.

What is OpenTelemetry?

An open-source observability framework for cloud-native software, aimed at providing standardized APIs, libraries, agents, and instrumentation for capturing and exporting telemetry data such as metrics, logs, and traces.

Why OpenTelemetry?

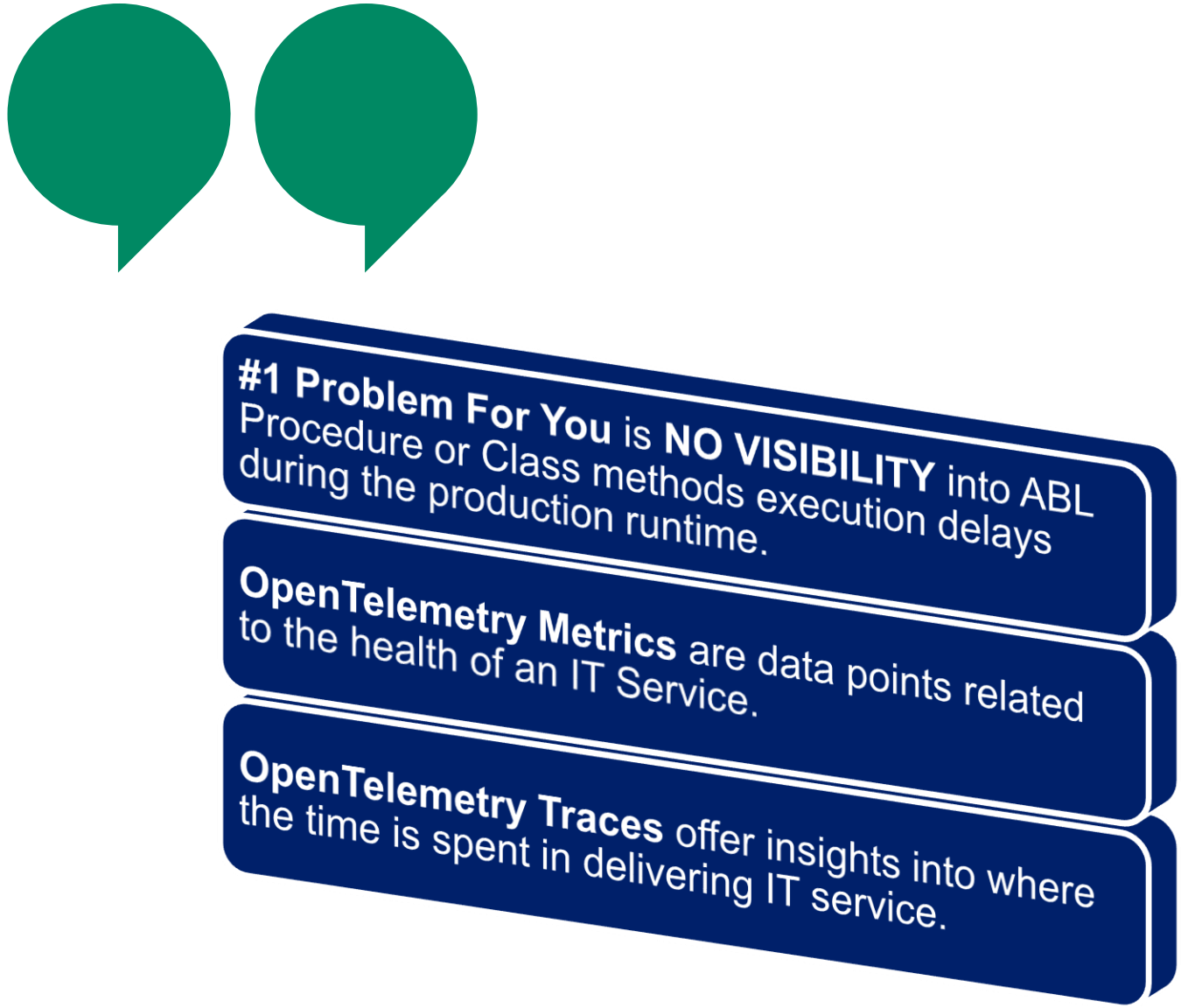
- CNCF - Merger of OpenTracing & OpenCensus Projects
- **Specification** of APIs, SDKs and Data Format.
- **Standardizes** the way telemetry data is collected & transmitted to **backend observability APM Tools**
- An **observability framework** for cloud-native software - distributed and polyglot Architecture
- Has **broad industry support** and adoption from 1000+ cloud providers, vendors and end users.
- Provides an ability to send telemetry data to multiple back-ends in parallel fashion
  - **Vendor-agnostic**
  - **Provides flexibility**

OpenTelemetry – OpenEdge Application Monitoring Vision

Offer *Production run-time observability* thru an Open Standard API i.e., OpenTelemetry Metrics, Traces, and Logs to address the Application monitoring needs of OpenEdge Customers.

Application Monitoring UseCases

- Need **Auto tracing** for ABL Procedures, Class methods.
- Need **custom metrics** at the technical system, application and business layer.
- Need **traces or observations tracked** not within PASOE/Database but the outside application service – when OpenEdge Server talking to another type of application.
- **Monitor every metric that impacts App. Performance. Support our Enterprise APM Tool.**
- Find the **slow ABL Code at Production run-time**, where my OpenEdge application is running slow in conjunction with other applications.



Progress

Progress®

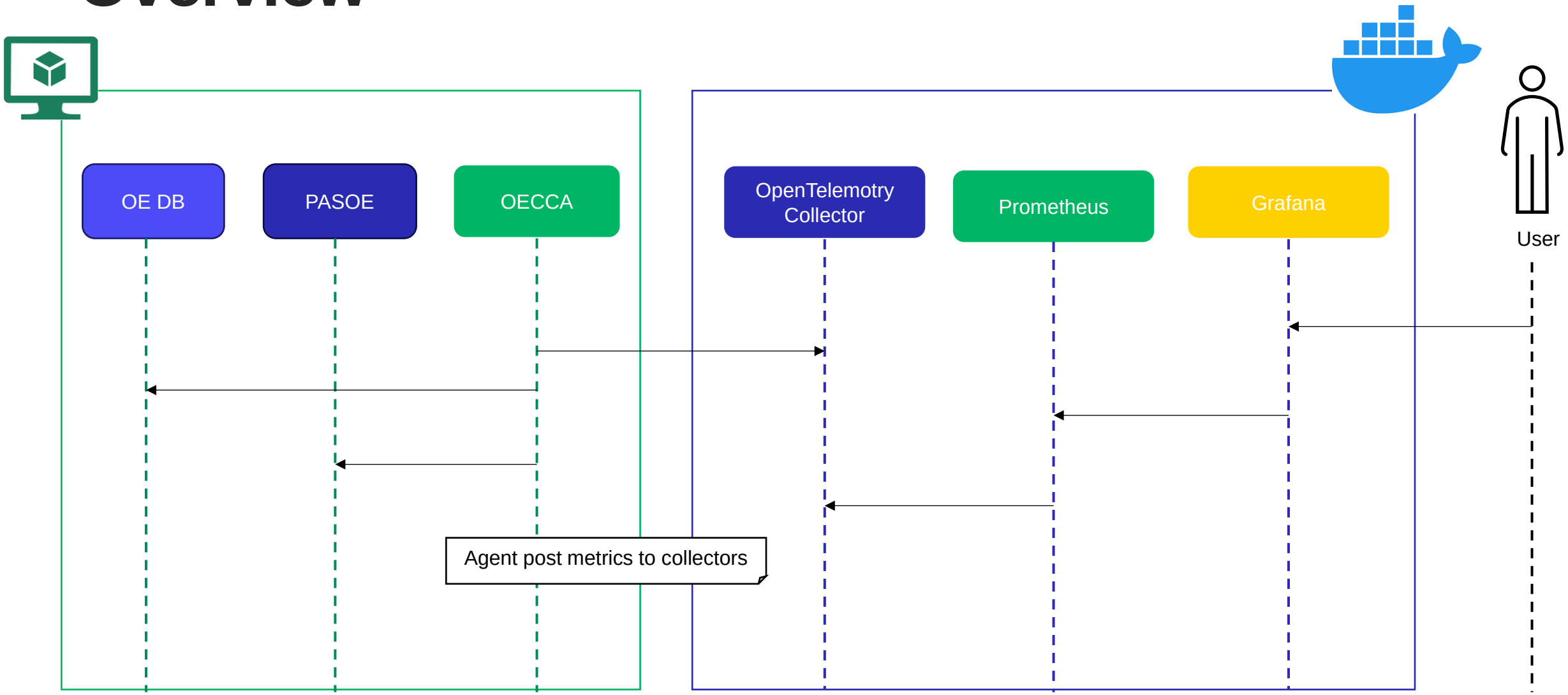
OpenTelemetry Collector

Source : https://opentelemetry.io/docs/collector/

The OpenTelemetry Collector offers a vendor-agnostic implementation of how to receive, process and export telemetry data.

OTEL COLLECTOR

Overview



OTel. Collector’s config.yaml

Receivers 

- Collect telemetry from one or more sources.
- If you specify the name of the receiver only in the receivers section, OTel will use the receiver’s the default configuration,

Processor 

- Take the data collected by receivers and modify or transform it before sending it to the exporters
- Define rules and settings: filtering, dropping, renaming, or recalculating telemetry..

Configuring a receiver or a processor does not enable it. Receivers are enabled by adding them to the appropriate pipelines within the service section.

OTel. Collector’s config.yaml

Exporter

- Generates and sends the trace data from OE resources
- OTLP endpoint exporters including OTLP/GRPC and OTLP/ HTTP

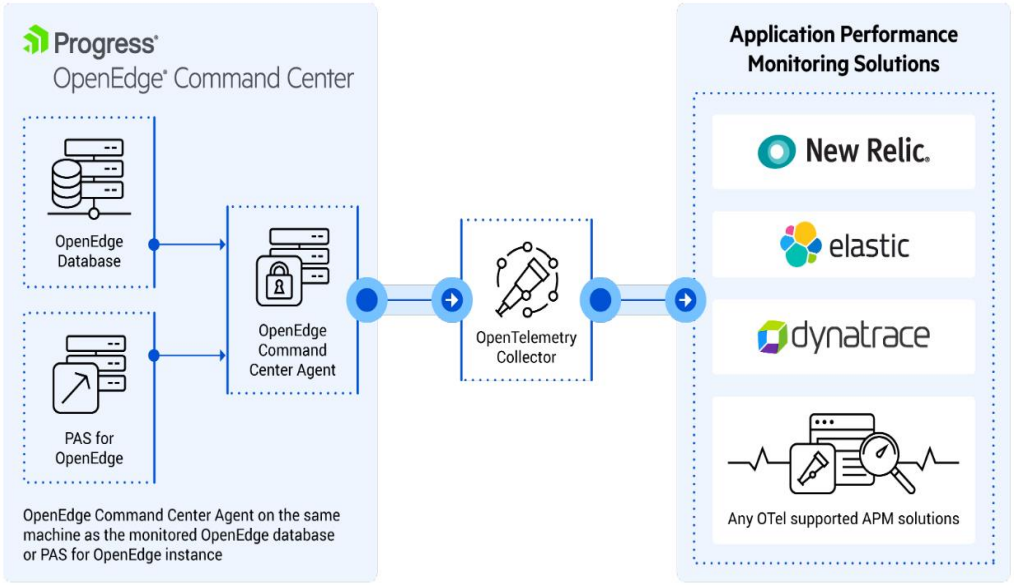
Service

- Has three subsections:
  - Extensions
  - Pipelines
  - Telemetry
- To configure what components are enabled in the Collector

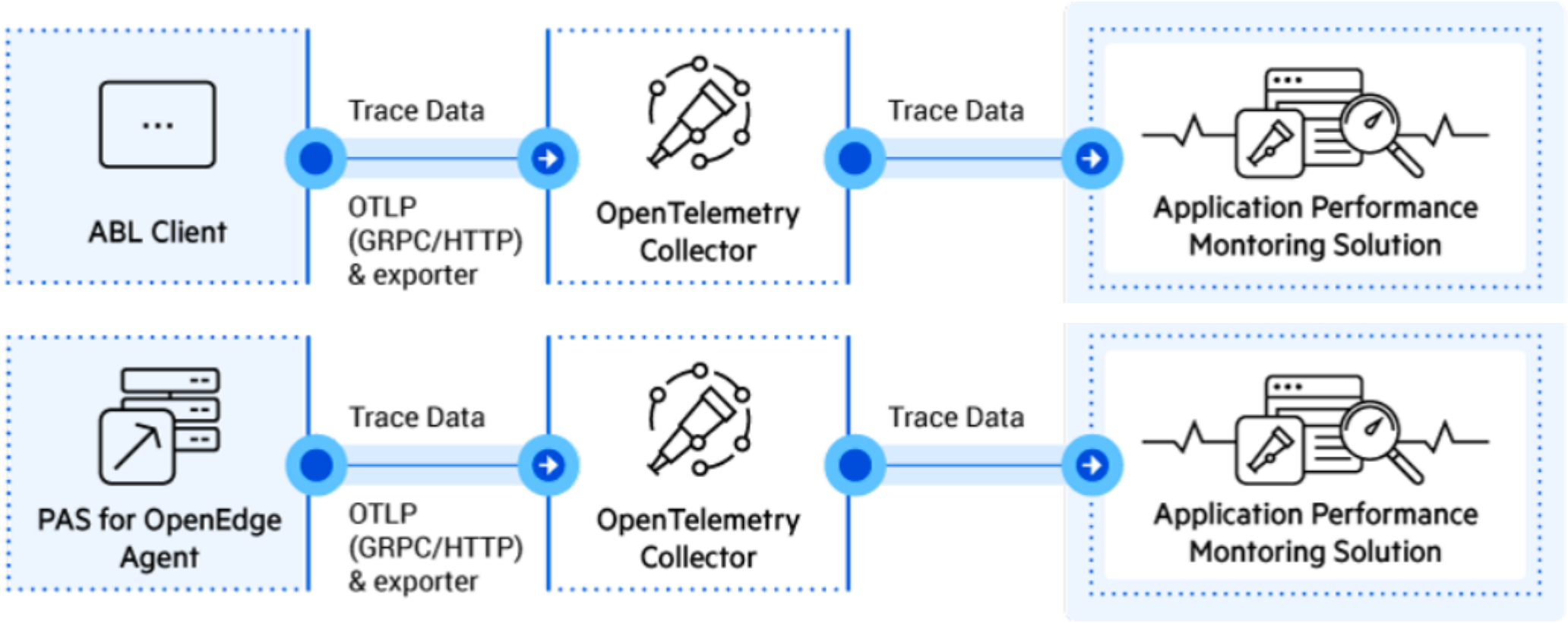
OpenEdge & OpenTelemetry Traces

OpenTelemetry Traces

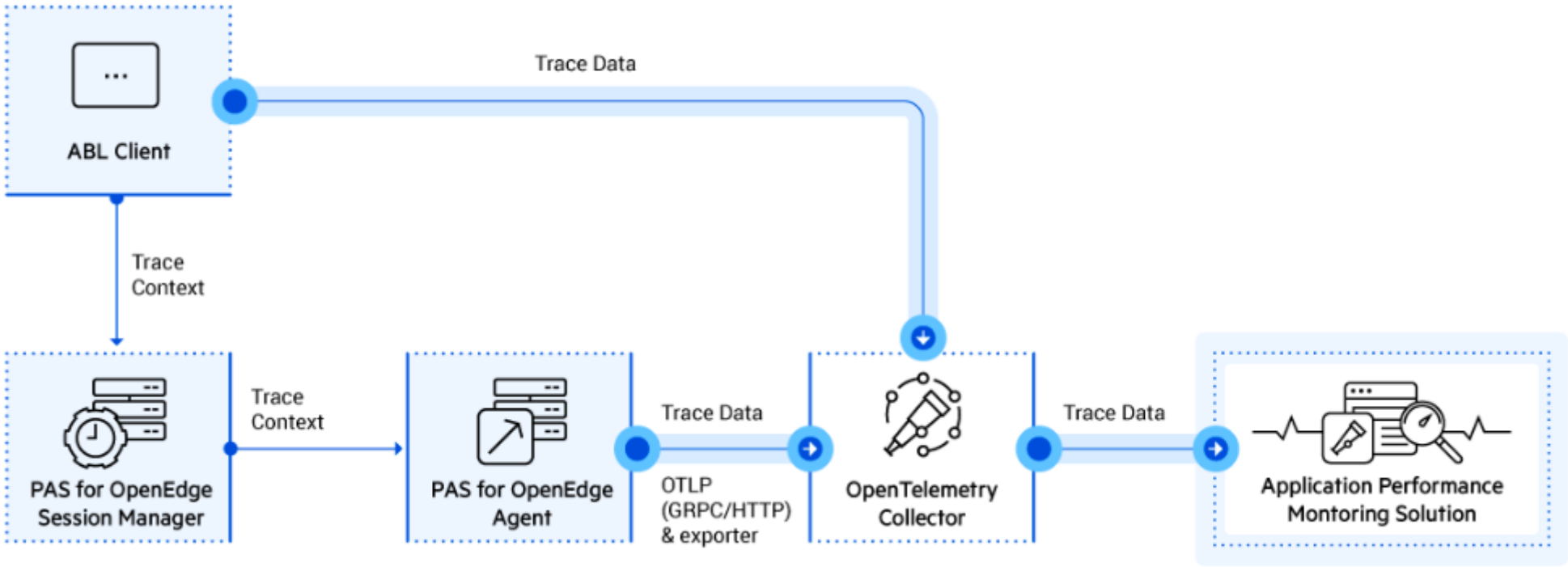
- AVM Auto Tracing
- Configurable: ABL Procedure/Class/Method begin/end first, Transaction begin/end & DB Connect/Disconnect planned
- Trace Id propagated to external PAS for OpenEdge Instances
- No changes to the application code
- New Built-in “Progress.Lang.Telemetry” Class
- Application-level integration: insert Begin/End Spans
- Add Data to current Span to propagate Trace ID for ABL & HTTP Client first, .NET & Java planned
- PAS for OpenEdge Session Manager Tracing
- Trace Request through all transports (APSV first, Web, REST, SOAP planned), through stages of request fulfilment



OpenTelemetry Tracing



1 trace for client and server

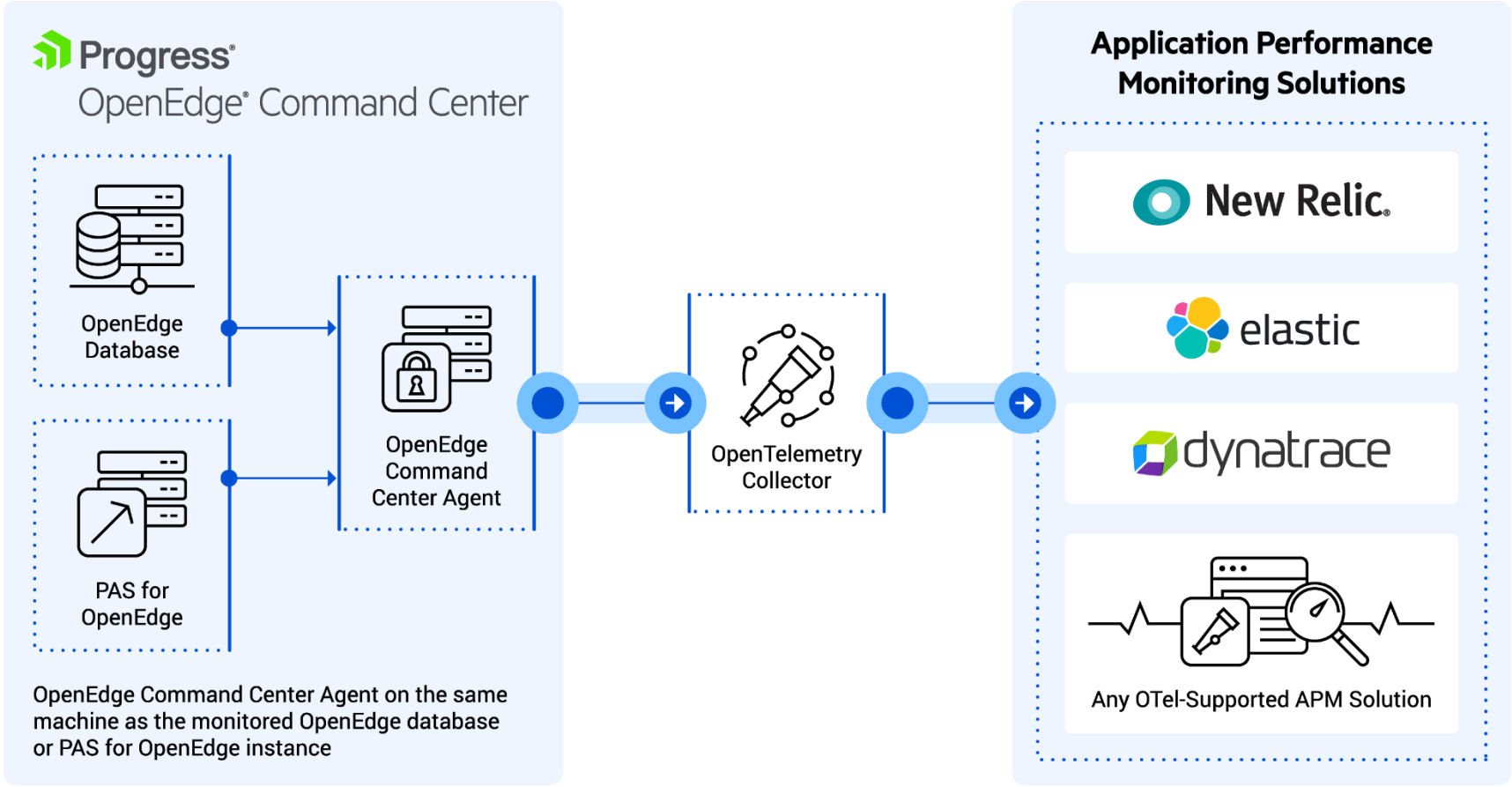


OpenTelemetry Tracing

```
{"OpenTelemetryConfiguration": {
  "exporters": {
            "otlp": {
                "grpc": [
                    {
                        "endpoint": "http://localhost:4317",
                        "span_processor": "batch",
                        "batch_processor_options": {
                            "max_queue_size": 500
                        }
                    }
            ]
        }
    }
},
"OpenEdgeTelemetryConfiguration": {
    "trace_procedures": "*",
    "trace_requires_parent": true,
    "trace_abl_transactions": true,
    "trace_request_start": true
}
}
```

OpenEdge & OpenTelemetry Metrics

OpenTelemetry Metrics



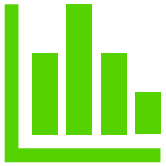
OTel. Metrics for OpenEdge 12.2 & Above

OpenEdge Database Metrics

- Activity Summary
- AI Activity
- BI Activity
- Buffer Activity
- Record Activity

PAS for OpenEdge Metrics

- Agent metrics for ABL applications.
- Request metrics for ABL applications.
- Session metrics for ABL applications.
- Connection metrics for ABL applications.
- Transport metrics for web applications associated with an ABL application (REST/SOAP/WEB/APSV)



Metrics

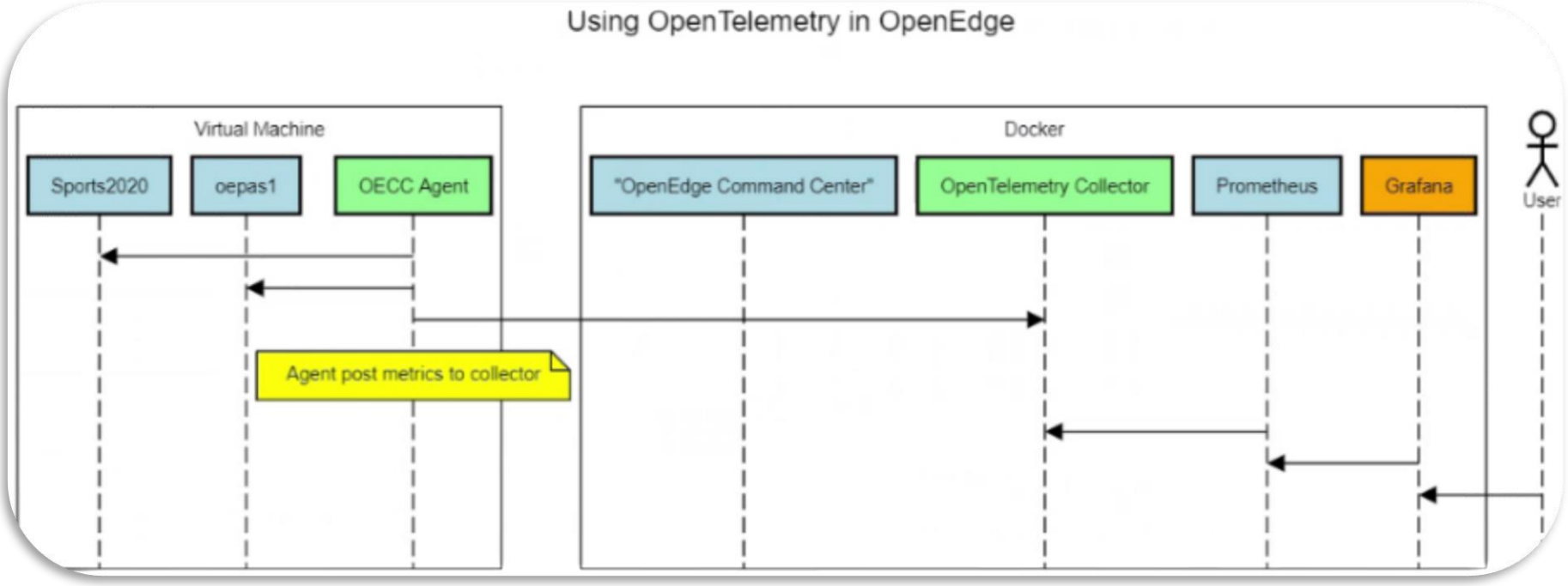
Time series of data points related to the health of the IT Service

OECC: OpenEdge Database / PASOE Performance Monitoring Metrics

Metrics	Description
Commits	The number of transactions all users have committed.
Undos	The number of transactions rolled back.
Record Updates	The number of records updated.
Record Reads	The number of records read.
Record Creates	The number of records created.
Record Deletes	The number of records deleted.
DB Writes	The number of database blocks written to disk.
DB Reads	The number of database blocks read.
BI Writes	The number of Before-Image (BI) blocks written to disk.
BI Reads	The number of BI blocks read.
AI Writes	The number of After-Image (AI) blocks written to disk.
Record Waits	The number of times users have waited to access a locked record.
Checkpoints	The number of checkpoints that have been performed.
Buffs Flushed	The number of database buffers that have been flushed to disk because they were not written by the time the checkpoint ended.
Rec Lock Waits	The percentage of record accesses that resulted in a record lock wait. A record lock wait occurs when the database engine must wait to access a locked record.
BI Buf Waits	The percentage of BI buffer waits. A BI buffer wait occurs when the database engine must wait to access a BI buffer.
AI Buf Waits	The percentage of AI buffer waits. An AI buffer wait occurs when the database engine must wait to access an AI buffer.
Writes by APW	The percentage of database blocks written to disk by the Asynchronous Page Writer (APW).
Writes by BIW	The percentage of BI blocks written to disk by the Before-Image Writer (BIW).
Writes by AIW	The percentage of AI blocks written to disk by the After-Image Writer (AIW).
Buffer Hits	The percentage of buffer hits for both the primary and alternate buffer pools. A buffer hit occurs when the database engine locates a record in the buffer pool and does not have to read the record from the disk.
Primary Hits	The percentage of buffer hits for the primary buffer pool.
Alternate Hits	The percentage of buffer hits for the alternate buffer pool.

Metrics Type	Metrics	Description
REST transport	expressionErrors	The number of expression errors.
	failedRequests	The number of failed requests.
	successfulRunRequests	The number of requests for which response was received su
	successfulRequests	The number of requests successfully sent to the PAS for Ope
	connectRequests	The number of connection requests.
	statusRequests	The number of status type requests.
	Requests	The total number of requests.
	successfulConnectRequests	The number of successful connection requests.
	serviceUnavailableRequests	The number of requests for which services were not availab
	runRequests	The number of run requests.
SOAP transport	urlNotFoundErrors	The number of errors because of incorrectly supplied URLs.
	activeRequests	The number of requests that are in the active state.
	wsdlRequests	The number of WSDL requests.
	successfulSoapRequests	The number of successful SOAP requests to the PAS for Ope instance.
	soapRequests	The total number of SOAP requests.
	methodNotAllowederrors	The number of errors caused because the requested metho authorized.
	httpRequestErrors	The number of HTTP requests that resulted in errors.
	httpRequests	The total number of HTTP requests to the PAS for OpenEdge
	soapProcessorErrors	The number of SOAP processor errors.
APSV transport	forbiddenErrors	The number of requests that failed with the 403 error code.
	disconnectErrors	The number of disconnect errors to the PAS for OpenEdge i
	connectErrors	The number of connection errors.
	disconnectRequests	The number of disconnect requests to the PAS for OpenEdg
	sessionRequests	The number of session requests to the PAS for OpenEdge in
	sessionErrors	The number of session errors.
WEB transport	headRequests	The number of HEAD requests to the PAS for OpenEdge inst
	traceRequests	The number of TRACE requests to the PAS for OpenEdge ins
	optionsRequests	The number of OPTIONS requests to the PAS for OpenEdge
	patchRequests	The number of PATCH requests to the PAS for OpenEdge ins
	getRequests	The number of GET requests to the PAS for OpenEdge instan
	servletRequests	The number of SERVLET requests to the PAS for OpenEdge i
	deleteRequests	The number of DELETE requests to the PAS for OpenEdge in
	putRequests	The number of PUT requests to the PAS for OpenEdge instan
	postRequests	The number of POST requests to the PAS for OpenEdge insta
	successfulServletRequests	The number of successful SERVLET requests to the PAS for C instance.
	headErrors	The number of HEAD requests that resulted in errors.
	traceErrors	The number of TRACE requests that resulted in errors.
	optionsErrors	The number of OPTIONS requests that resulted in errors.
	patchErrors	The number of PATCH requests that resulted in errors.
	getErrors	The number of GET requests that resulted in errors.
	deleteErrors	The number of DELETE requests that resulted in errors.
	putErrors	The number of PUT requests that resulted in errors.
	postErrors	The number of POST requests that resulted in errors.
	ablRuntimeErrors	The number of ABL runtime errors.
	ablConnectErrors	The number of ABL connections errors.
	failedServletRequests	The number of SERVLET requests that resulted in errors.

Flow of Metrics using Prometheus / Grafana



OpenTelemetry Metrics

```
exporter:
  name: "otlp"
  endpoint: "http://10.5.0.7:4317"
  protocol: "grpc"
  connectionretry: 20
  timeout: 10

oedbInstances:
  -  dbname: mttenant
     host: localhost
     port: 37010
     user: rdroge
     password: youdliketokn0w
#     metricsregex:
#     otherdbconnparams:
     dbconnectionretry: 5
     dbschedule: 30
     dbduration: seconds
```

```
exporter:
  name: "otlp"
  endpoint: "http://10.5.0.7:4317"
  protocol: "grpc"
  connectionretry: 20
  timeout: 10

pasInstances:
  -  pasdir: "/app/pas/prodpas/"
     passchedule: 30
     pasconnectionretry: 5
     pasduration: seconds
#     metricsregex:
```

Progress® OpenEdge®

News You Can Use


Progress®