z/TPF Support for OpenTelemetry

Operations and Coverage

Josh Wisniewski

2025 TPF Users Group Conference
May 4-7, Austin, TX

IBM Z

IBM

z/TPF Support for OpenTelemetry

- Transaction processing typically involves multiple systems across your enterprise. For example:

- If you monitor your transaction processing at system A, you can monitor whether service level agreements (SLAs) are being violated:
  - Is transaction processing completing in under X milliseconds?
  - Is the transaction error rate less than Y%?

- But what happens if SLAs are not being met?

z/TPF Support for OpenTelemetry

As-Is:

- An “all hands on deck” fire drill occurs to determine the cause.
- Each silo works to prove that the problem is not in their domain.
- Resolution of the issue can be slow and inefficient, which can elongate the time that the SLAs are violated and can result in financial penalties.

z/TPF Support for OpenTelemetry

To-Be:

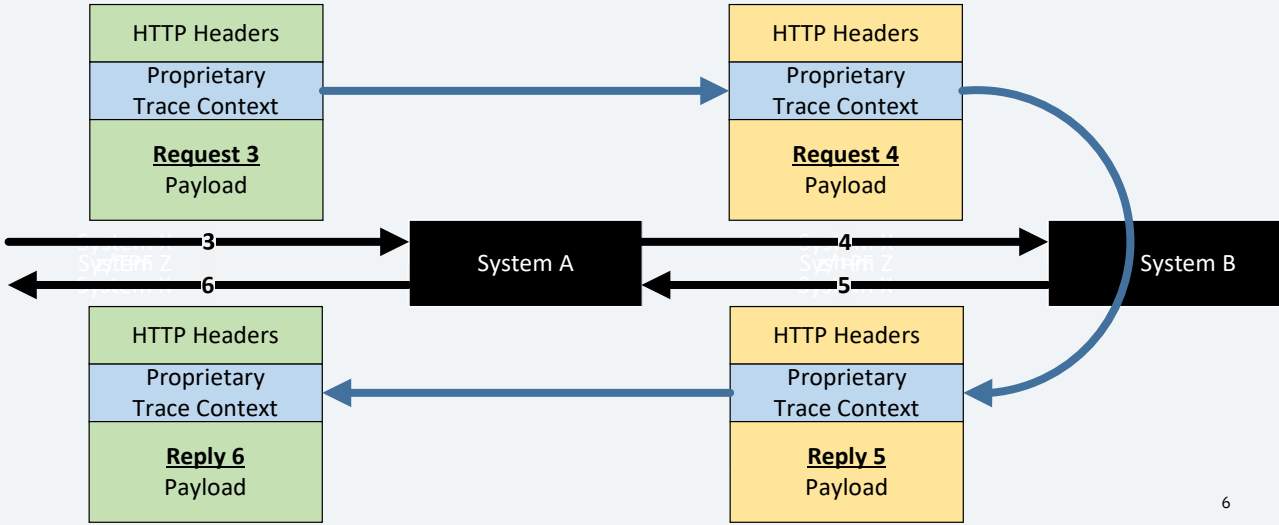
- All platforms involved in processing a transaction are monitored by a common application performance monitoring (APM) tool.
- APM tools can quickly identify where failures originate so that you can resolve the issue as quickly as possible.
- APM tools are including AI techniques that might give you the capability to detect and resolve issues before SLAs are violated.

What is OpenTelemetry?

- OpenTelemetry (Otel) is a collection of APIs, SDKs, and tools. Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software's performance and behavior. https://opentelemetry.io/
- OpenTelemetry provides a standardized open-source interface to a variety of APM tools: IBM Instana, AppDynamics, Dynatrace, Jaeger, New Relic, Sumo Logic, Splunk, and more.
- OpenTelemetry provides integration into many languages, platforms, tools and more for end-to-end enterprise monitoring.
- The goal of your APM tool is to have visibility across all platforms involved in the processing of your transactions.
- Your APM tool can analyze the system state and transactions from each of the platforms and alert you to issues before your SLAs are violated and speed the investigation and resolution of issues in your enterprise.

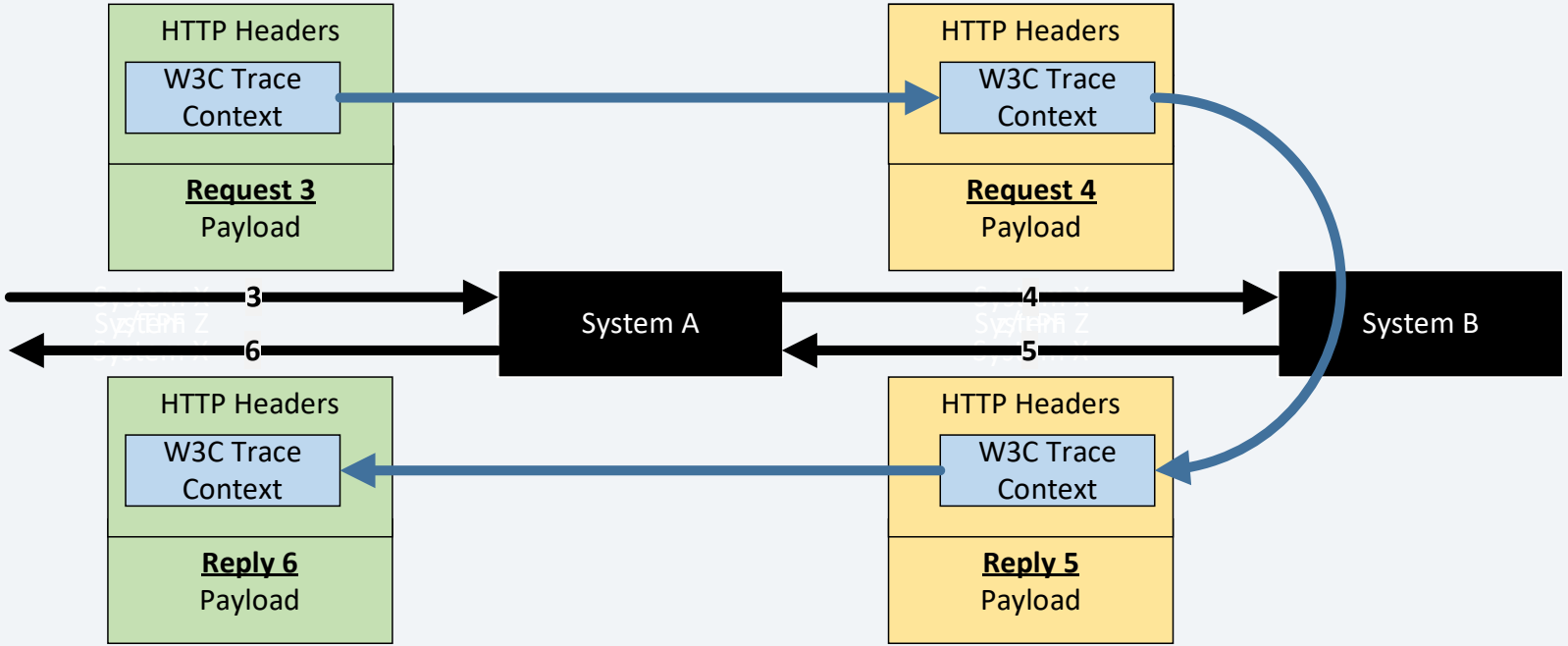
APM Tool: Trace Context

- How does an APM tool tie the metrics from the different systems together to understand a single transaction across all the platforms?
- Each request embeds a trace context with trace IDs to tie the details from different platforms together. Some standard middleware like HTTP and IBM MQ defines **where** in the message header to put the trace ID information, but **how** to locate it, insert it into the message header, and receiver interaction with it is implementation dependent.
- Before the OpenTelemetry standard, every APM tool vendor defined their own trace context format and conventions.



OpenTelemetry – APM Tool: Trace Context

- The OpenTelemetry standard uses the W3C standard trace context embedded in the standard HTTP headers.
- The W3C standard trace context defines a 16-byte trace ID to uniquely identify each transaction and tie the trace results from each platform together in the APM tool.



OpenTelemetry: Slow Network Use Case

- Suppose booking messages are taking longer and longer to complete, encroaching on your SLA terms.
- Without an APM tool, coverage for each system has visibility to the state of only their system. Each silo only sees that processing on their system is completing quickly. As a result, it takes a long time to diagnose, your SLAs are violated, and after a great deal of finger pointing, you determine that the problem is a communications delay at step 4.
- With an APM tool, which has visibility to the processing time on each system and communication times between systems, your APM tool can identify that booking messages are taking longer and alert you to the fact that you have a communications delay issue at step 4 (not in z/TPF or system C processing time). As such, you immediately know where to focus your investigation and can avoid violating your SLAs.

OpenTelemetry: High Error Rate

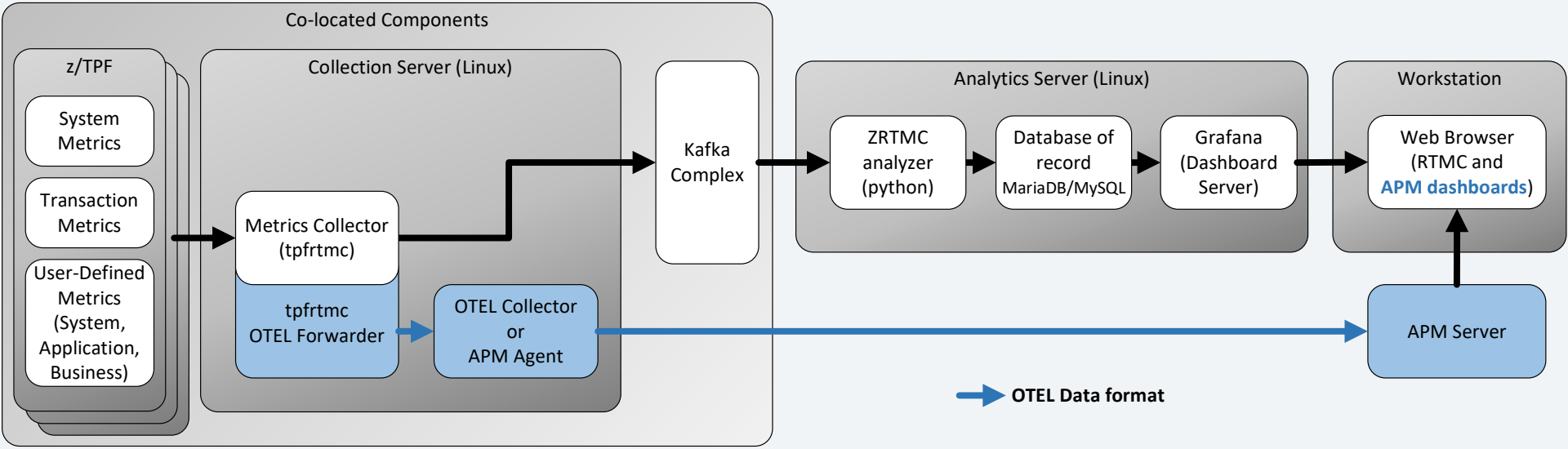
- Suppose more calls than usual are ending in error when processing your transactions.
- Without an APM tool, coverage for the various systems are all seeing more calls than usual ending in error. Coverage for each system must investigate if the errors are being propagated from called systems or if the errors are originating from their system. This requires a very knowledgeable SME to do this investigation and takes a long time.
- With an APM tool, which has visibility to the return code on each of the systems, your APM tool can alert and show you which platform those errors are originating from and propagating back in the replies. As such, you immediately know where to focus your investigation.

APAR PJ48032 (Feb 2025)
z/TPF Support for OpenTelemetry

- First deliverable includes offline runtime metrics collection changes to accelerate your adoption of z/TPF support for OpenTelemetry:
  - There are no z/TPF online code changes.
  - Name-value pair transactional data as **OpenTelemetry trace data**.
  - CDC and user-defined metrics data as **OpenTelemetry metrics data**.
- No changes to your application are required. However, you will get better results if you modify your applications to adhere to the z/TPF Otel name-value pair conventions (trace ID, return code, and so on).
- Runtime metrics collection is configurable and extensible to send results to any tool that supports OpenTelemetry.
- We primarily tested with IBM Instana and Jaeger but support the OpenTelemetry standard in support of any tool that consumes it. Your APM tool might require additional configuration, a standalone agent, or other modifications.

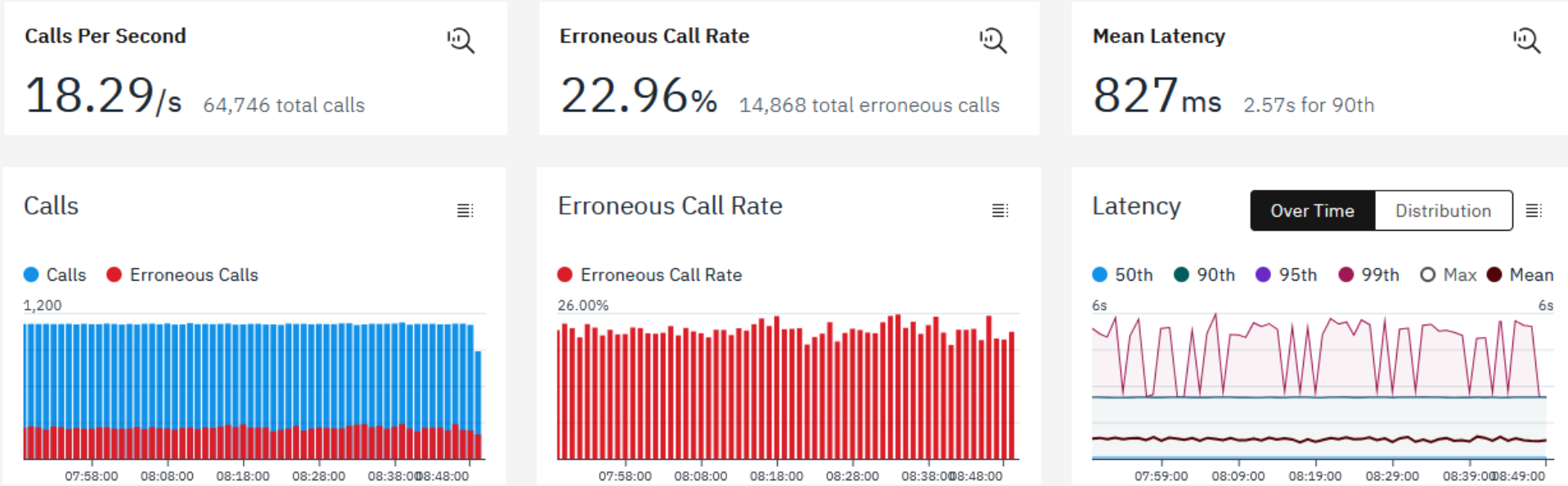
APAR PJ48032 (Feb 2025)
z/TPF Support for OpenTelemetry

- tpfrtmc OpenTelemetry forwarder is implemented as a user exit library. You can:
  - Customize which metrics to send, including user-defined metrics.
  - Customize for your APM tool, if necessary.



z/TPF Support for OpenTelemetry

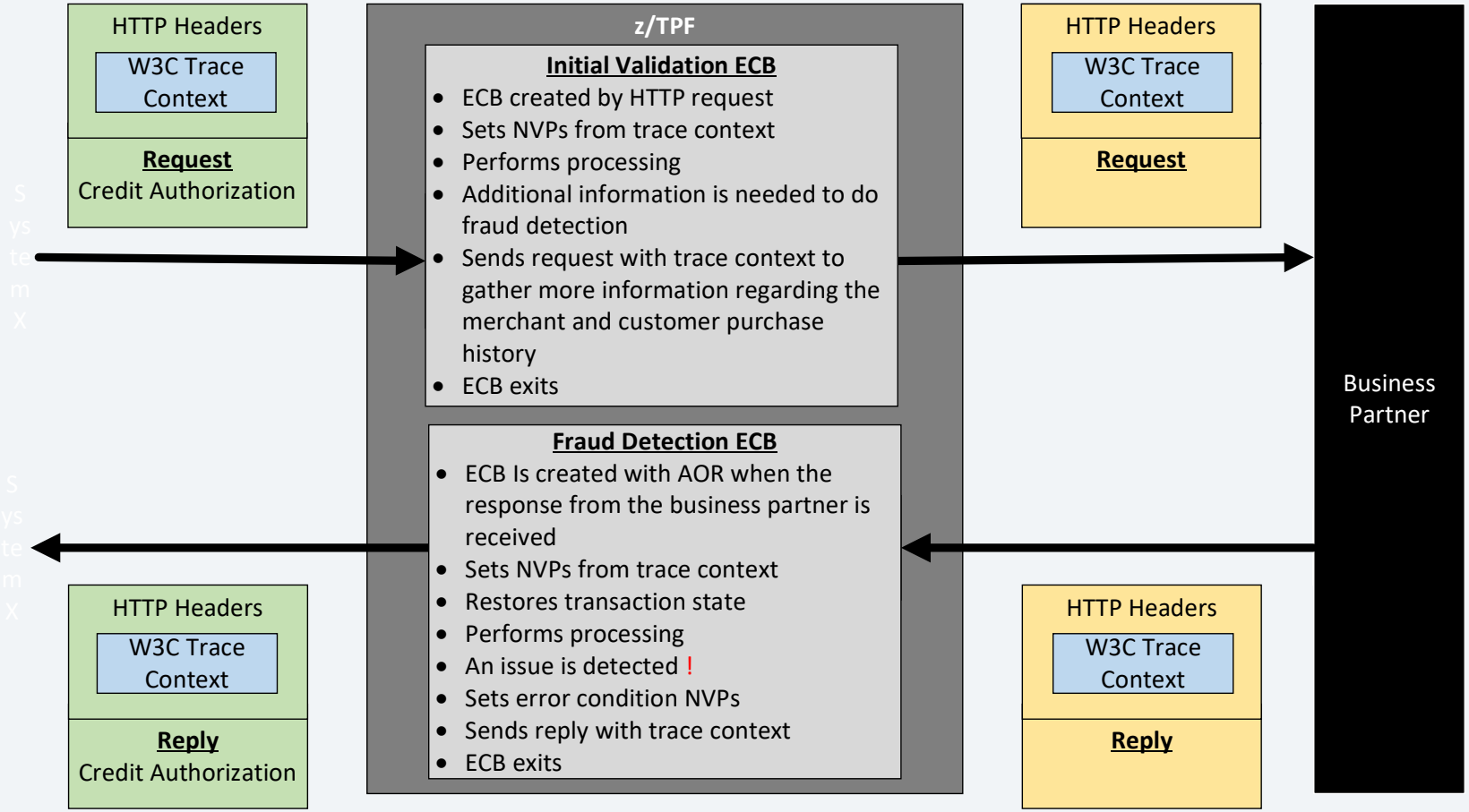
- Your APM tool can show overall or by message type metrics like the number of calls sampled per second, the error rate, and message existence time.
- For example, overall metrics as shown in IBM Instana:



z/TPF Support for OpenTelemetry

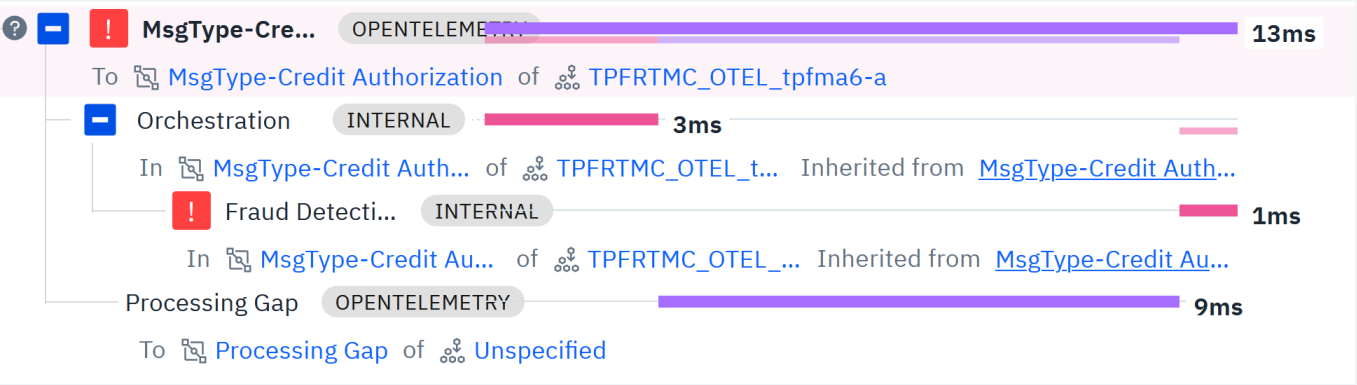
- The following slides show a message processed by z/TPF using 2 ECBs with a call to a business partner.
- In this scenario, only z/TPF is sending trace span data to the APM tool, IBM Instana.
- In the education session, I will show an example of a similar message in the context of an enterprise in IBM Instana.

z/TPF Support for OpenTelemetry



z/TPF Support for OpenTelemetry

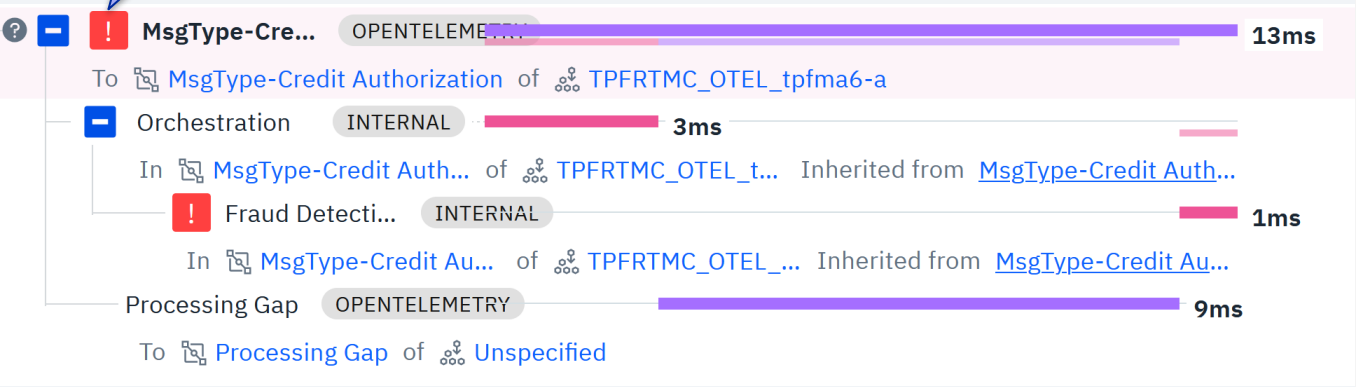
- Your APM tool can:
  - Show the relationship between systems and ECBs. In this example, the parent child relationship is shown by way of indenting the child.
  - Highlight where the failures originate.
  - Show key metrics and name-value pairs at the message and ECB level.



EXIST_TIME	3894
FIND_DASD	0
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	0

z/TPF Support for OpenTelemetry

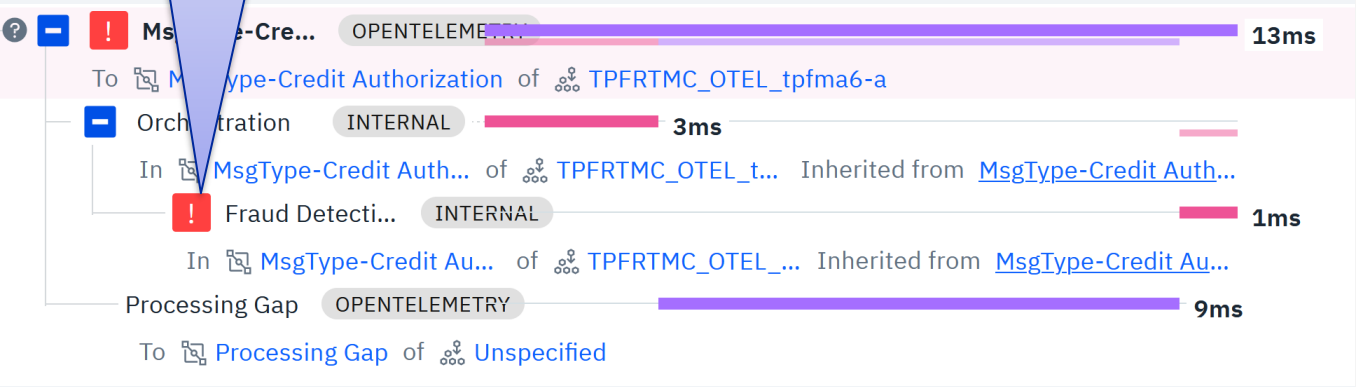
Indicates message processing ended in error.



EXIST_TIME	3894
FIND_DASD	0
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	0

z/TPF Support for OpenTelemetry

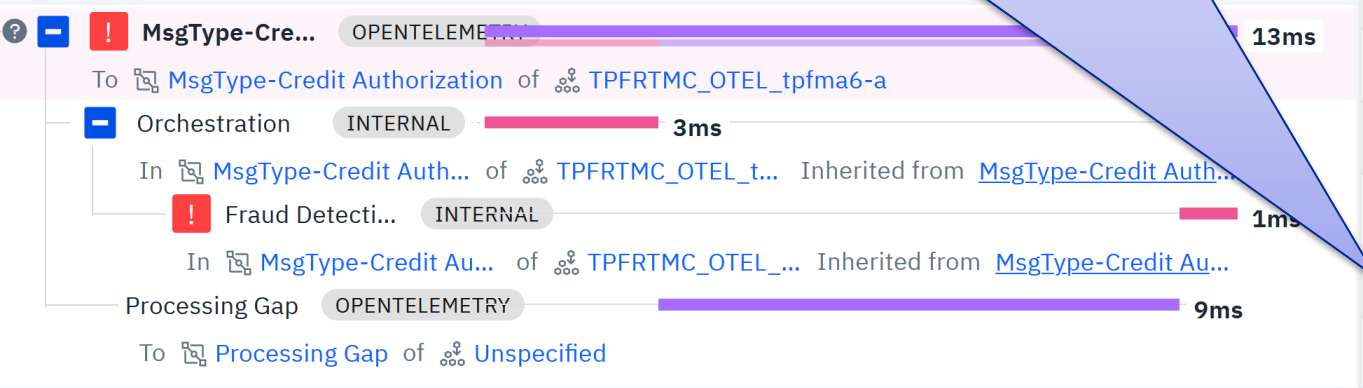
Indicates Fraud Detection ECB set an error return code



EXIST_TIME	3894
FIND_DASD	0
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	0

z/TPF Support for OpenTelemetry

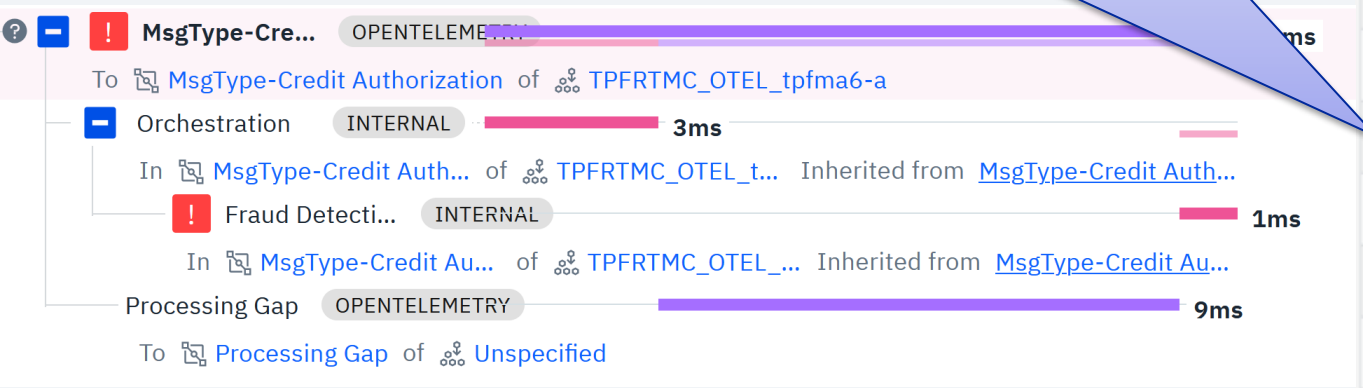
Results can include your name-value pairs like MsgType=**Credit Authorization**



EXIST_TIME	3894
FIND_DASD	24
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	11

z/TPF Support for OpenTelemetry

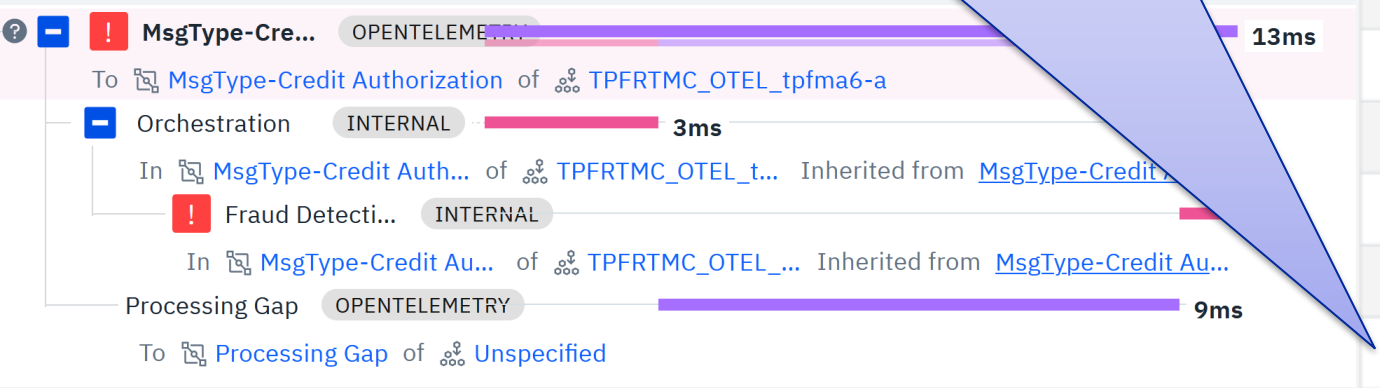
Return code name-value pair indicates to the APM tool whether the message or ECB processing resulted in success or failure.



EXIST_TIME	3894
FIND_DASD	24
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	11

z/TPF Support for OpenTelemetry

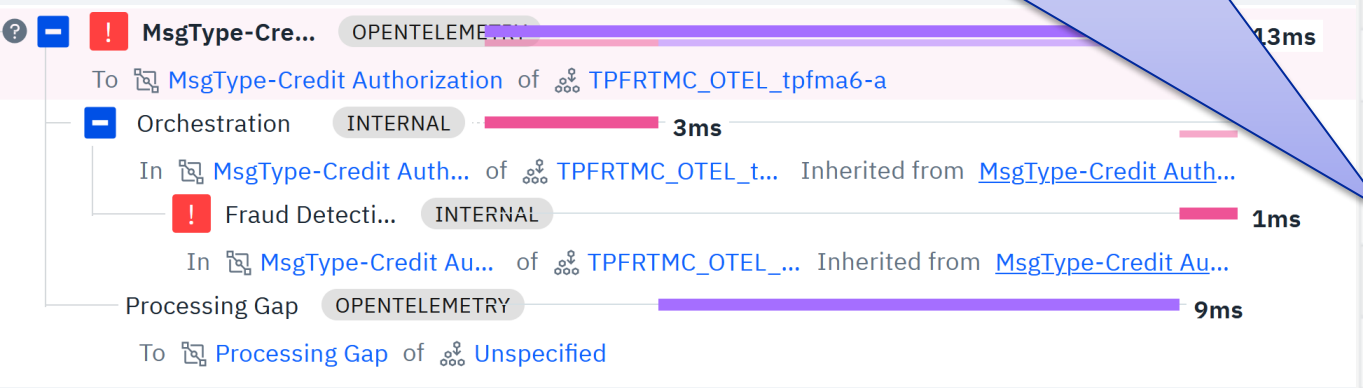
Return code message name-value pair is optional and provides additional details regarding the message or ECB processing success or failure.



EXIST_TIME	3894
FIND_DASD	24
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	11

z/TPF Support for OpenTelemetry

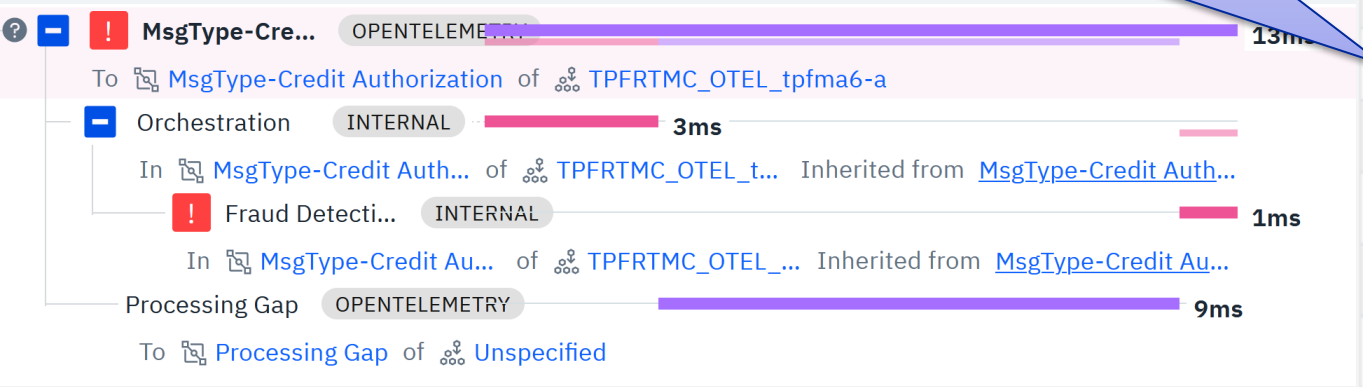
Name-value pair metrics can be included such as
CPU_USED (2.171ms) = ECB execution time
EXIST_TIME (3.894ms) = ECB existence time



EXIST_TIME	3894
FIND_DASD	24
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	11

z/TPF Support for OpenTelemetry

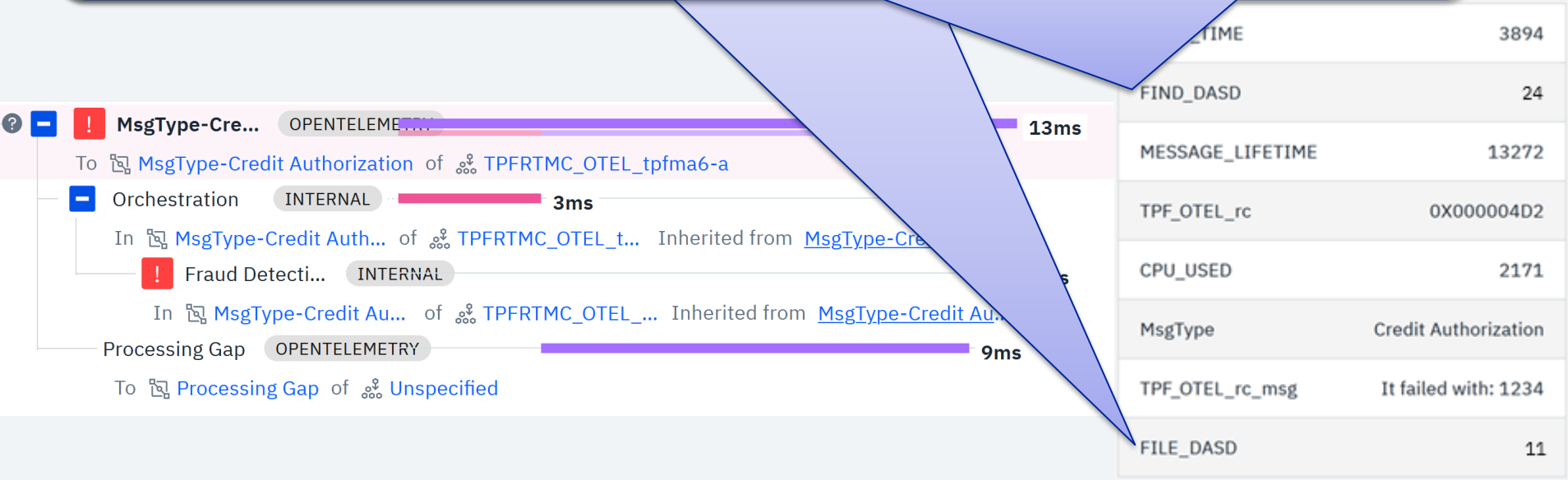
MESSAGE_LIFETIME (13.272ms) = wall clock time elapsed from the start of the first ECB to the exit of the last ECB to exit



EXIST_TIME	3894
FIND_DASD	24
MESSAGE_LIFETIME	13272
TPF_OTEL_rc	0X000004D2
CPU_USED	2171
MsgType	Credit Authorization
TPF_OTEL_rc_msg	It failed with: 1234
FILE_DASD	11

z/TPF Support for OpenTelemetry

Name-value pair metrics can be included such as:
FIND_DASD (24) = Finds of records that were satisfied from DASD
FILE_DASD (11) = Files of records that were not put in virtual file access (VFA)



z/TPF Support for OpenTelemetry

What can you do to facilitate your adoption?

- Ensure your edge servers are generating a trace context that conforms to the W3C standard which includes the unique 16-byte trace ID for each transaction and propagate the trace context through your enterprise as the transaction is processed on your various platforms.
- If necessary, modify your systems that call z/TPF with HTTP to send the trace ID in the HTTP W3C trace context header.
- Update any proprietary communications protocols to propagate the W3C trace context and parse trace IDs into name-value pairs per z/TPF conventions.
- Update requests flowing over standard middleware (such as HTTP, IBM MQ, and so on, ideally in user exit code) to parse trace IDs into name-value pairs per z/TPF conventions.
- Update your application or user exits to propagate the W3C trace context and parsed trace IDs into name-value pairs.

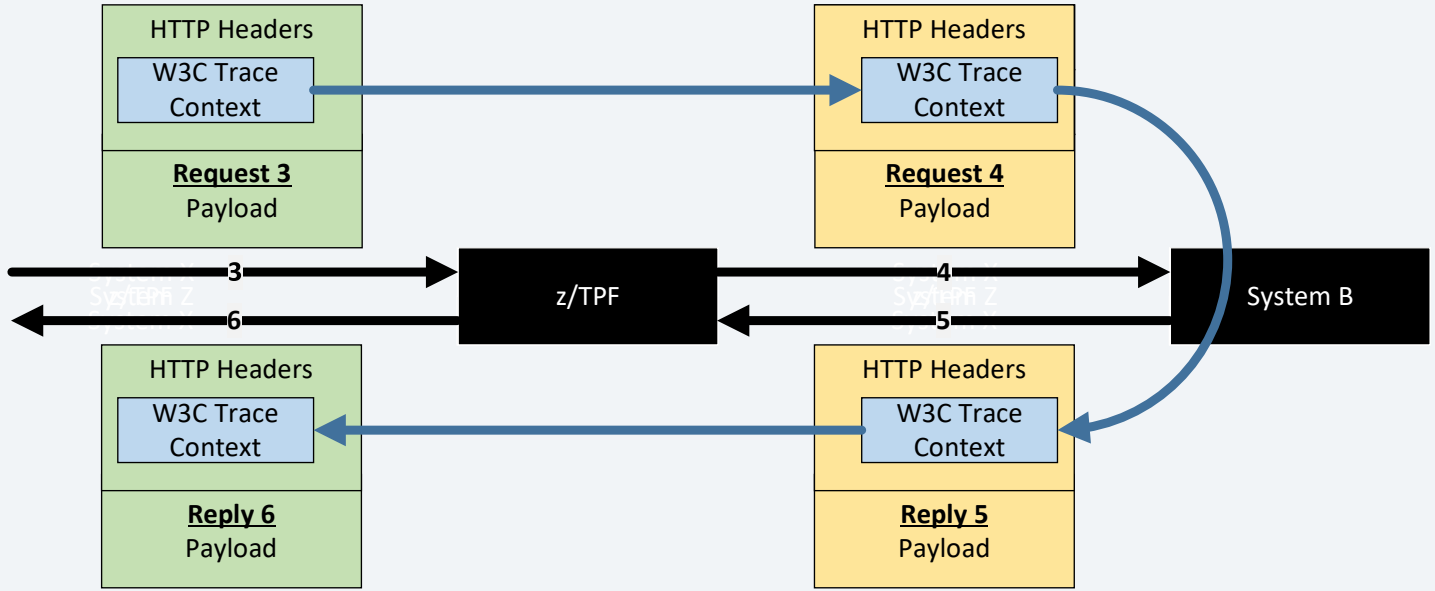
For more information, see Optimize OpenTelemetry Data.

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

z/TPF Support for OpenTelemetry

- Future deliverables look to enhance the z/TPF HTTP and IBM MQ handling to:
  - Automatically propagate the W3C trace context to requests, replies, etc.
  - Automatically create name-value pairs for z/TPF APM trace data.



Thank you