OpenTelemetry Support for z/TPF
Operations and Coverage

Josh Wisniewski

2024 TPF Users Group Conference
May 5-8, New Orleans, LA

IBM Z

IBM

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.

OpenTelemetry Support for z/TPF

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

| Request ——1——▶ | System A | ——2——▶ | System B | ——3——▶ | z/TPF | ——4——▶ | System C |
| Reply ◀——8—— | | ◀——7—— | | ◀——6—— | | ◀——5—— | |

- If you monitor your transaction processing at system A, you can monitor whether SLAs 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?

OpenTelemetry Support for z/TPF

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 penalties.

OpenTelemetry Support for z/TPF

To-Be:
- All platforms involved in processing a transaction are monitored by a common tool, Application Performance Monitor (APM).
- APMs can quickly identify where failures originate so that you can resolve the issue as quickly as possible.
- APMs 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 APMs: 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 monitoring.
- The goal of your APM is to have visibility across all platforms involved in the processing of your transactions.
- Your APM 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: Trace Context

- How does an APM 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 vendor defined their own trace context format and conventions.
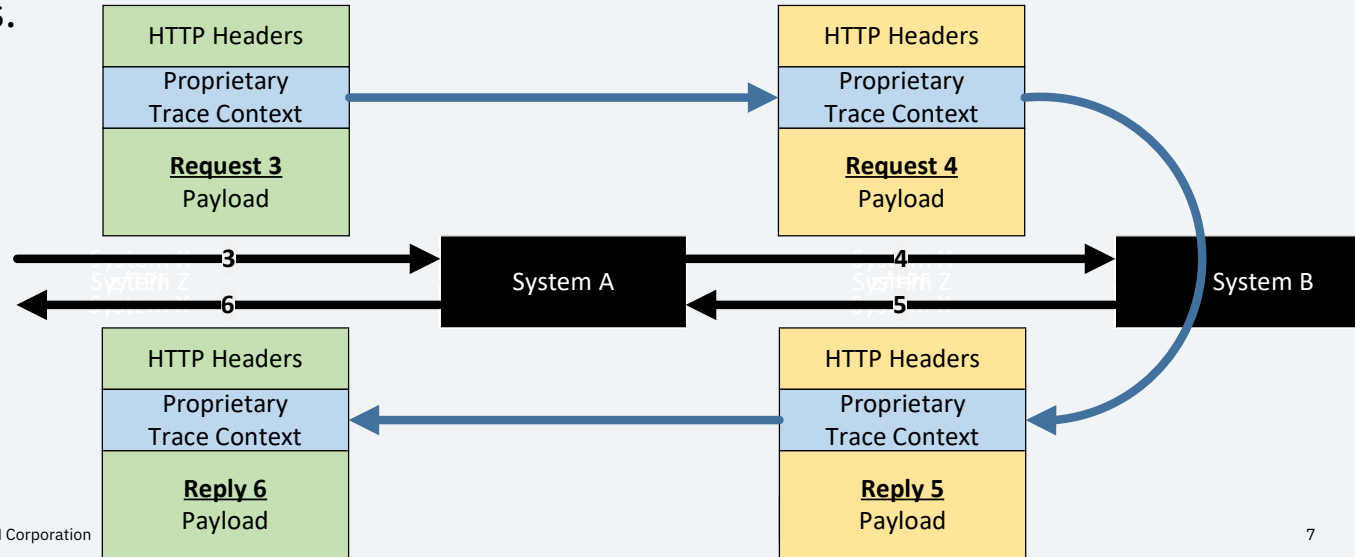
OpenTelemetry – APM: Trace Context

- 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.
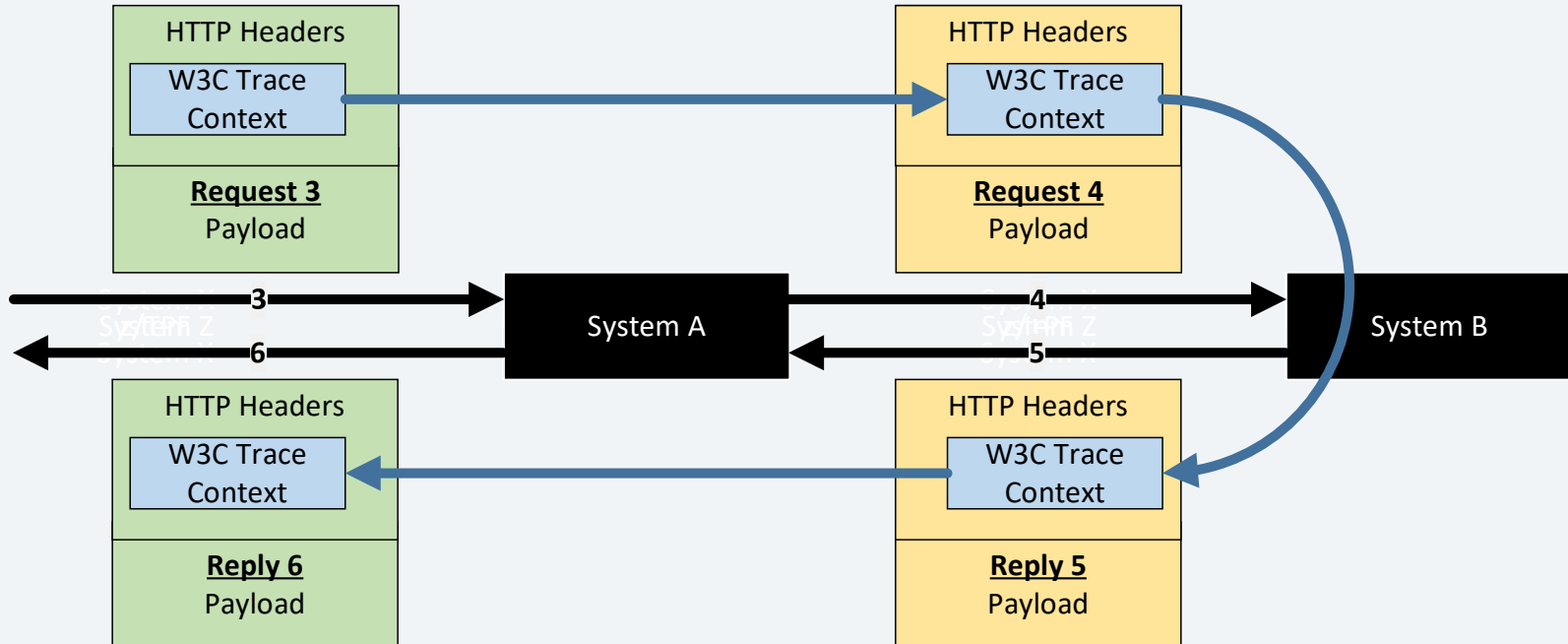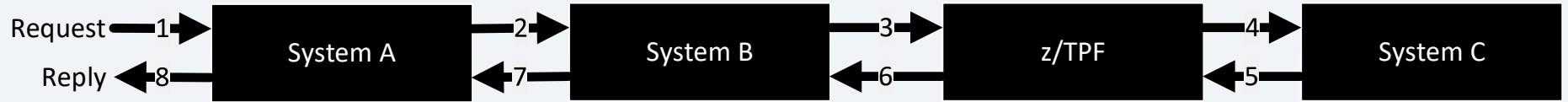
OpenTelemetry: Slow Network Use Case

Request ━1➤ **System A** ━2➤ **System B** ━3➤ **z/TPF** ━4➤ **System C**

Reply ◀8━ ◀7━ ◀6━ ◀5━
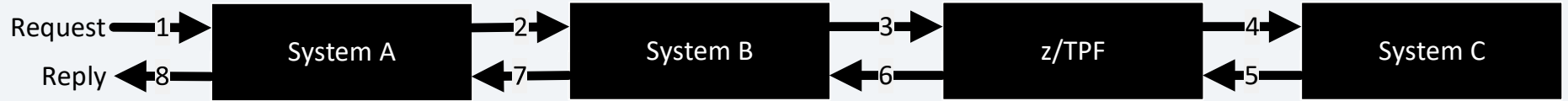
- Suppose booking messages are taking longer and longer to complete, encroaching on your SLA terms.
- Without an APM, coverage for each system only has visibility to the state of their system. Each silo only sees that processing on their system is completing quickly. As a result, it takes a long time 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, which has visibility to the processing time on each system and communication times between systems, your APM 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

```
Request ─1→  ┌─────────────┐ ─2→  ┌─────────────┐ ─3→  ┌─────────────┐ ─4→  ┌─────────────┐
             │   System A  │      │   System B  │      │    z/TPF    │      │   System C  │
 Reply  ←8─  └─────────────┘ ←7─  └─────────────┘ ←6─  └─────────────┘ ←5─  └─────────────┘
```

- Suppose more calls than usual are ending in error when processing your transactions.
- Without an APM, 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, which has visibility to the return code on each of the systems, your APM 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.

OpenTelemetry Support for z/TPF

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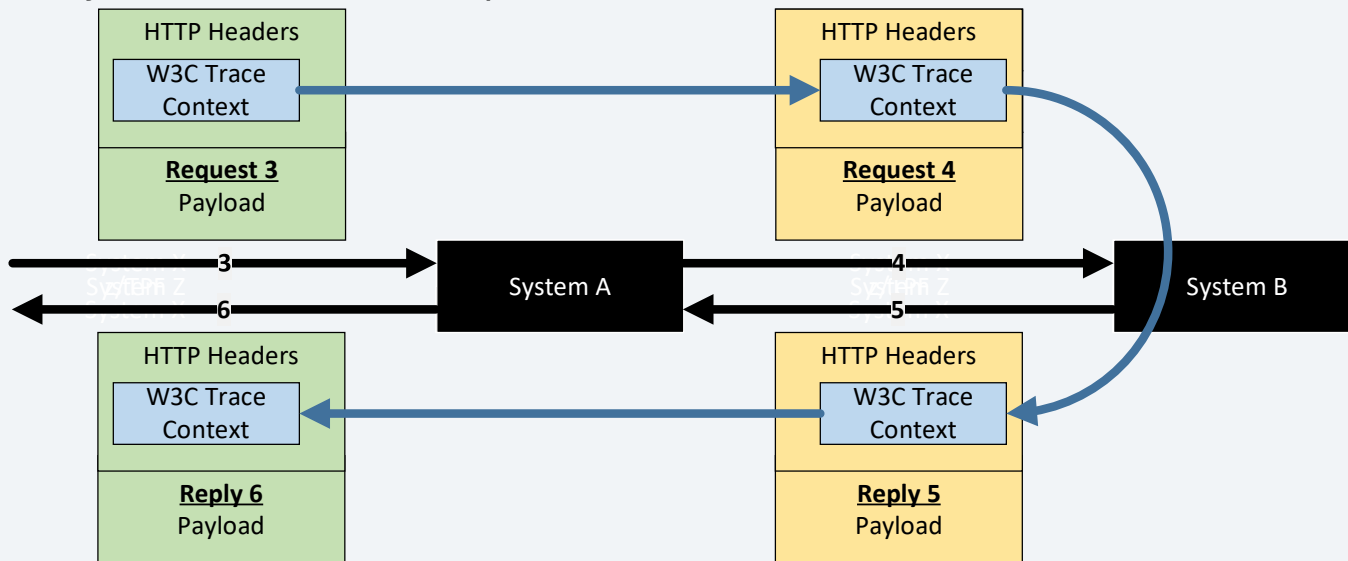
OpenTelemetry Support for z/TPF

- tpfrtmc OTEL forwarder will be implemented as a user exit library. You can:
  - Customize for your APM, if necessary.
  - Customize which metrics to send, including user-defined metrics.
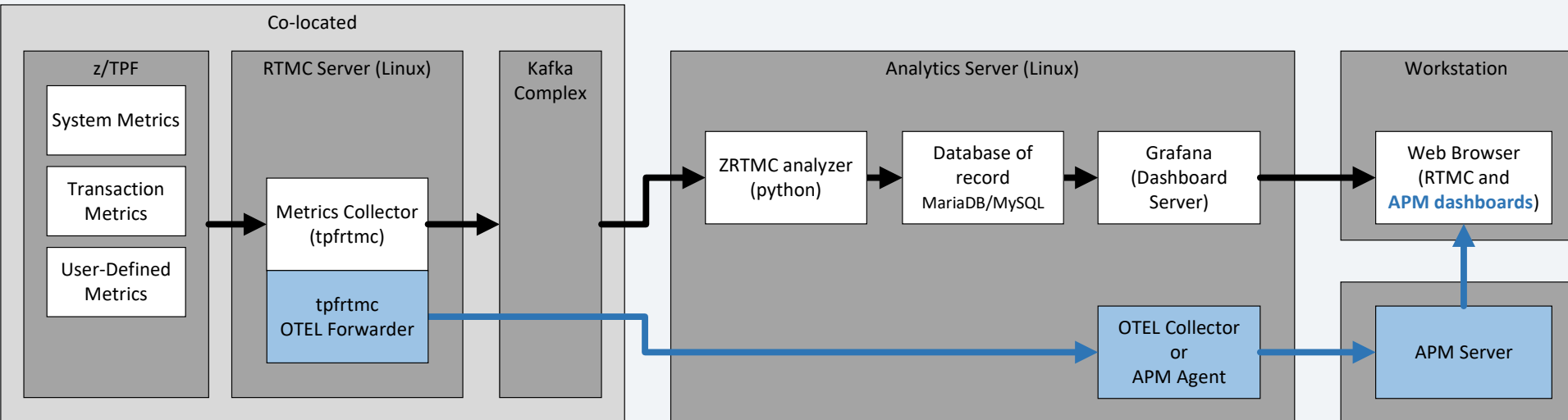
OpenTelemetry Support for z/TPF

- The initial deliverable does not include any online z/TPF code to propagate the W3C trace context or parsing trace IDs into name-value pairs. Your application or user exits can be updated to do this.
- A follow-on deliverable would 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.

OpenTelemetry Support for z/TPF

What should you do to start preparing for 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 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 (ideally in user exit code) to parse trace IDs into name-value pairs per z/TPF conventions.

Target delivery date is 4Q2024 for offline runtime metrics collection support.

Be a sponsor user

Sponsor users assist in design and implementation, and your feedback drives our development cycle.

We plan to have a customer playback soon.
- Demo.
- Discuss name-value pair conventions.
- Discuss trace span conventions.
- Discuss default metrics to send.

Target personas
- Site Reliability Engineer
- Coverage and Operations

Interested? Contact
   Josh Wisniewski (jwisniew@us.ibm.com)

Thank you

OpenTelemetry – APM: Sample Enterprise Architecture