OpenTelemetry tracing with CICS TS 6.3

Mark Cocker
mark_cocker@uk.ibm.com
Product Manager, CICS TS

Notices and disclaimers

This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.

Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM.

Not all offerings are available in every country in which IBM operates.

Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at: www.ibm.com/legal/copytrade.shtml.

Certain comments made in this presentation may be characterized as forward looking under the Private Securities Litigation Reform Act of 1995.

Forward-looking statements are based on the company’s current assumptions regarding future business and financial performance. Those statements by their nature address matters that are uncertain to different degrees and involve a number of factors that could cause actual results to differ materially. Additional information concerning these factors is contained in the Company’s filings with the SEC.
Copies are available from the SEC, from the IBM website, or from IBM Investor Relations.

Any forward-looking statement made during this presentation speaks only as of the date on which it is made. The company assumes no obligation to update or revise any forward-looking statements except as required by law; these charts and the associated remarks and comments are integrally related and are intended to be presented and understood together.

Agenda

Observability and IBM Z

- Open, cross-platform observability standards
- Native OpenTelemetry tracing on z/OS
- IBM Z Observability Connect

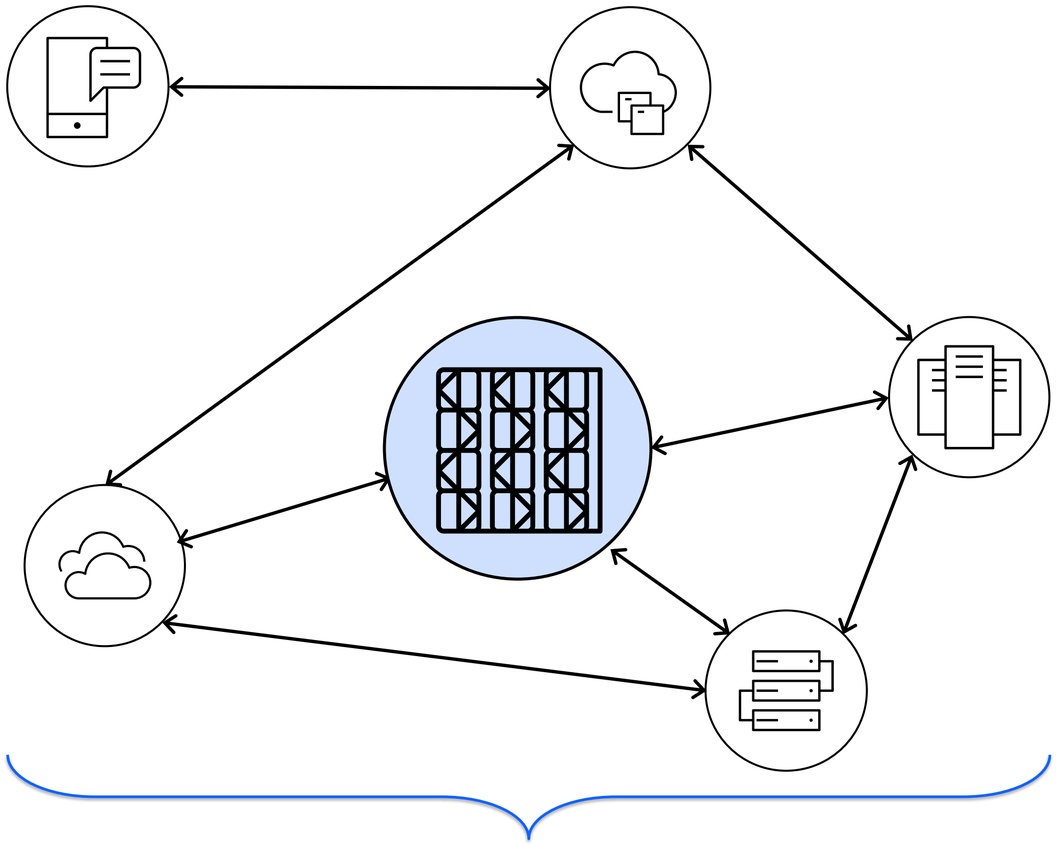
CICS TS 6.3

- What is OpenTelemetry?
- How are spans connected
- OTel trace propagation in CICS
- CICS configuration
- Trace span for a CICS task
- Examples with HTTP, LINK, MQ, Db2, Liberty

CICS TS open beta

- RETURN TRANSID() command
- SEND/CONVERSE command with DTP over MRO

Observability and IBM Z



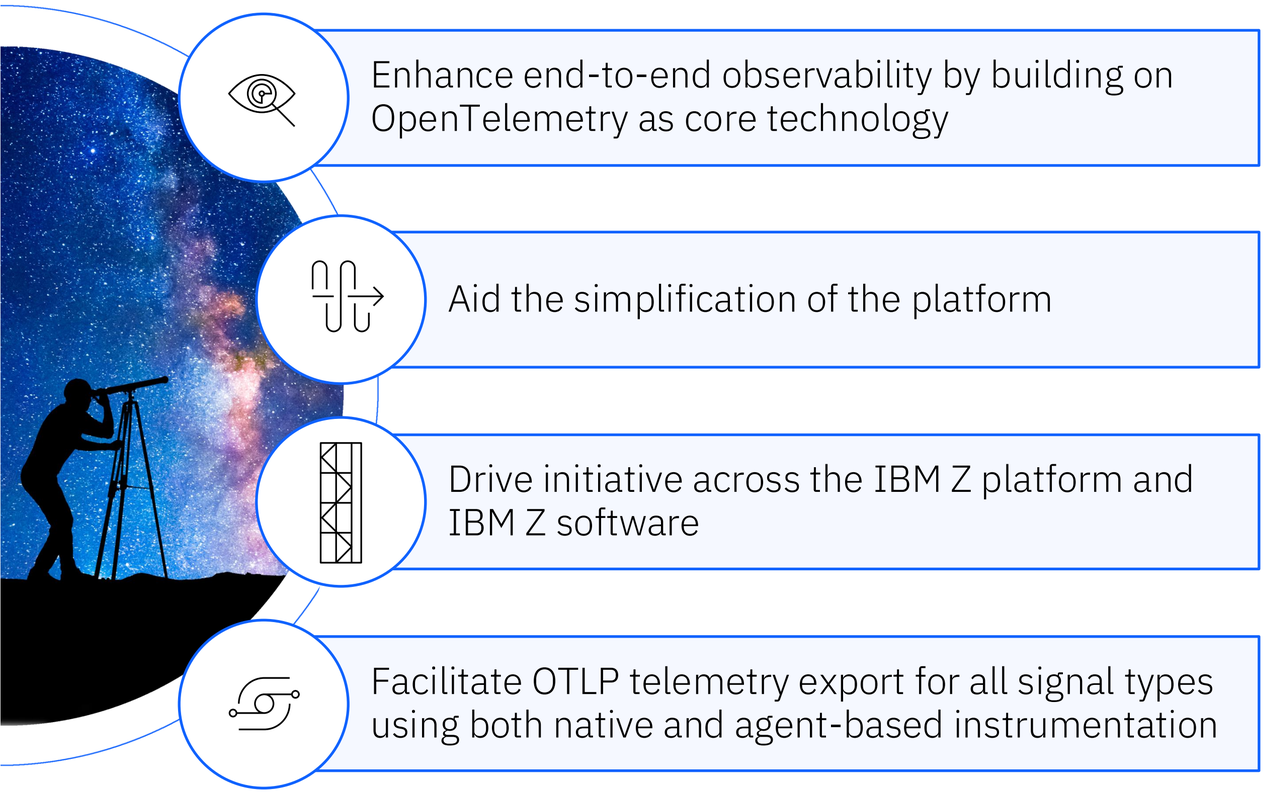
End-to-end observability for hybrid applications and underlying infrastructure

The mainframe is essential to the successful operations and business workflows of major enterprises…

Yet the majority of organizations lack integration of IBM Z into their enterprise-wide observability strategy

IBM’s strategy for Observability on IBM Z and IBM LinuxONE: Adopting open, cross-platform observability standards

Objectives:

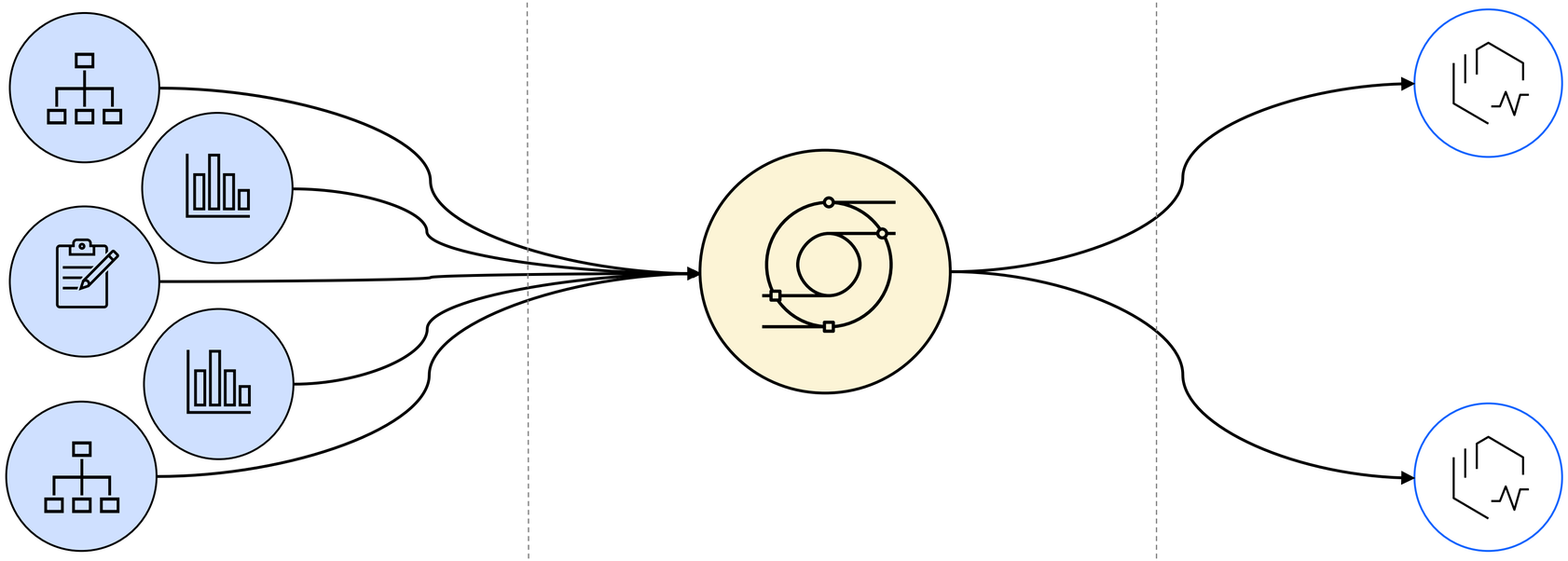


- Enhance end-to-end observability by building on OpenTelemetry as core technology
- Aid the simplification of the platform
- Drive initiative across the IBM Z platform and IBM Z software
- Facilitate OTLP telemetry export for all signal types using both native and agent-based instrumentation

Vision:

Ensure IBM Z is a first-class participant in enterprise-wide observability, supporting clients through the adoption and exploitation of standards like OpenTelemetry

Simplified adoption of OpenTelemetry on IBM Z from source to consumption

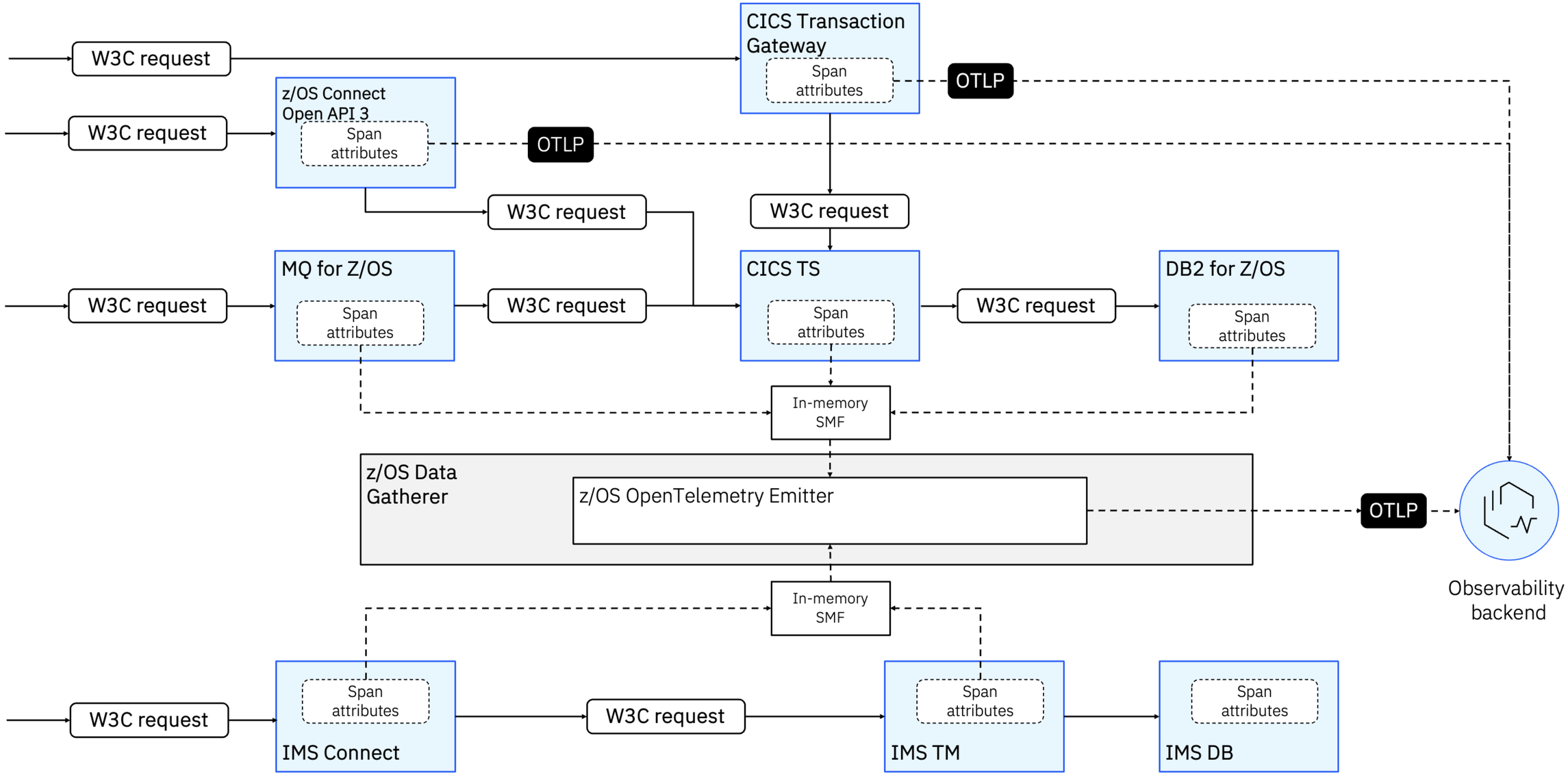


Data Producers
Deliver source telemetry signals (metrics, trace, logs)

Data Aggregation & Management
Act as gateway for signals from IBM Z to consumers.

Data consumers
Leverage signals to derive insight and analysis

Native OpenTelemetry distributed tracing on z/OS



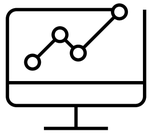
Native OpenTelemetry distributed tracing

Required product releases and updates	IBM z/OS Data Gatherer PTF for z/OS OpenTelemetry Emitter on z/OS 3.1 and 3.2
	CICS Transaction Server for z/OS 6.3
	CICS Transaction Gateway 10.1 PTF
	IBM MQ for z/OS 9.4.3 and potentially 9.4.4. (additional functionality)
	z/OS Connect (OpenAPI 3) V3.0.96 (September 2025) and above
	Db2 V13 PTF
	IMS 15.5 PTF
	WebSphere Liberty – support already built-in with Telemetry MicroProfile and support of OpenTelemetry Java Agent

Introducing **IBM Z Observability Connect**

Formerly IBM Z APM Connect

Demand for OpenTelemetry on IBM Z is growing, success is dependent simplicity of adoption

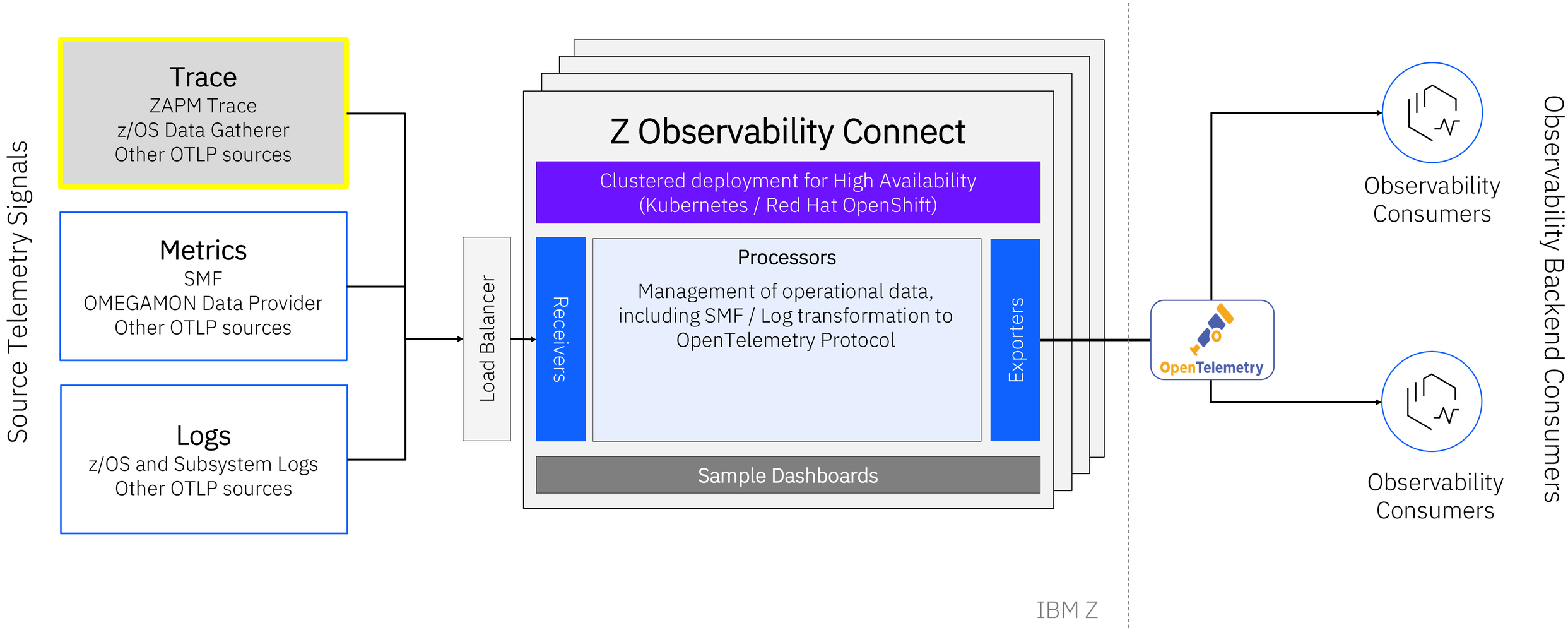


Observability platforms look to integrate trace, metrics and log signal types to provide end-to-end visibility

Clients are concerned about cost and management of enabling OpenTelemetry across critical z/OS resources

IBM Z Observability Connect is positioned to be the single data aggregator for mainframe operational data and deliver it to multiple consumers for observability of application and resources. It adds value by managing, filtering and enriching the source data for interchangeable consumers

Introducing Z Observability Connect



IBM CICS Transaction Server for z/OS 6.3

OpenTelemetry

Enable CICS to capture, propagate, and emit tracing data using OpenTelemetry standards so can observe and diagnose problems across hybrid cloud applications and infrastructure.

Modernization

Develop CICS applications in VS Code with enhanced editors from IBM and Zowe.

Use declarative YAML files to configure CICS regions with cicsconfig and start it.

AI agent ready

Quickly answer questions about CICS regions and configuration using AI assistants in natural language. CICS includes an MCP server to provide accurate information to AI agents during app development and problem diagnosis.

Core

Manage CICS with new CICS policy rules of APPC connection and JVM server status. Enhancement to statistics, XPI and API - many were raised as Ideas by clients!

Security

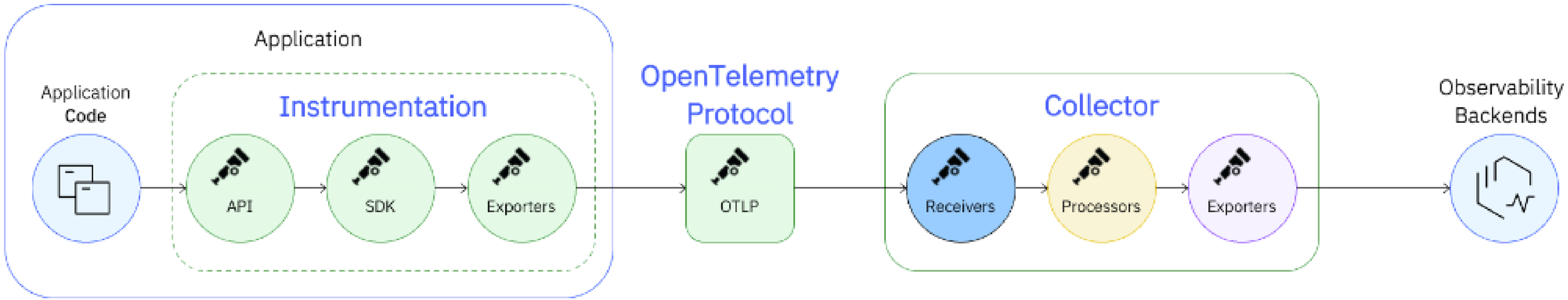
Simplify the adoption of Zero Trust principles. AT-TLS can be used to secure IPIC and outbound HTTPS connections. and security best practices. Adopt best-practices with enhanced workflow in Security Discovery editor.

Modern languages

Enable developers to use the latest Java and Node.js support, including Java 21, Jakarta EE 10, and Spring Boot 3. CICS Java APIs have been further enhanced.

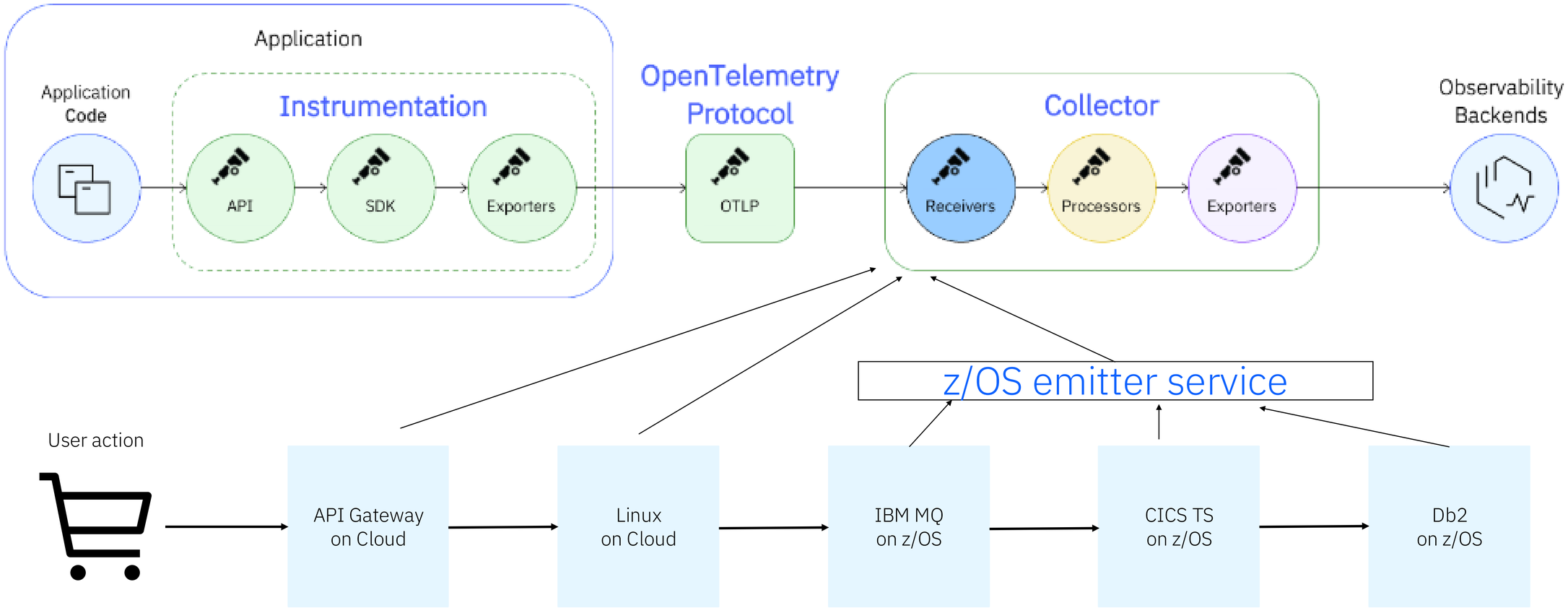
What is OpenTelemetry?

OpenTelemetry (OTel) is a vendor-agnostic *observability framework* that assists in generating, processing and distributing telemetry data such as metrics, logs and **traces**.



Source https://opentelemetry.io/docs/

Instrumentation

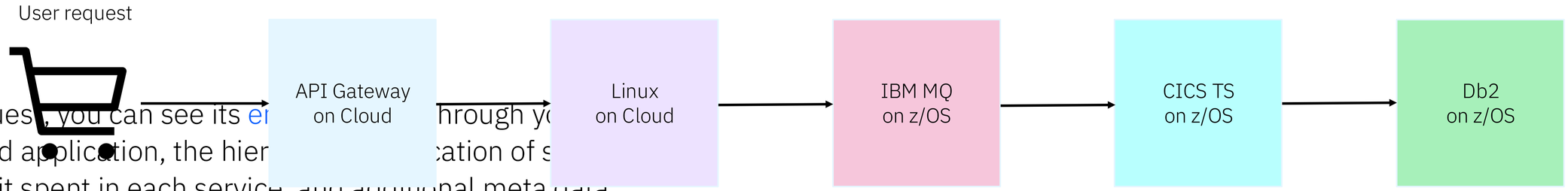


Runtimes are instrumented to emit OTel trace data corresponding to application request events

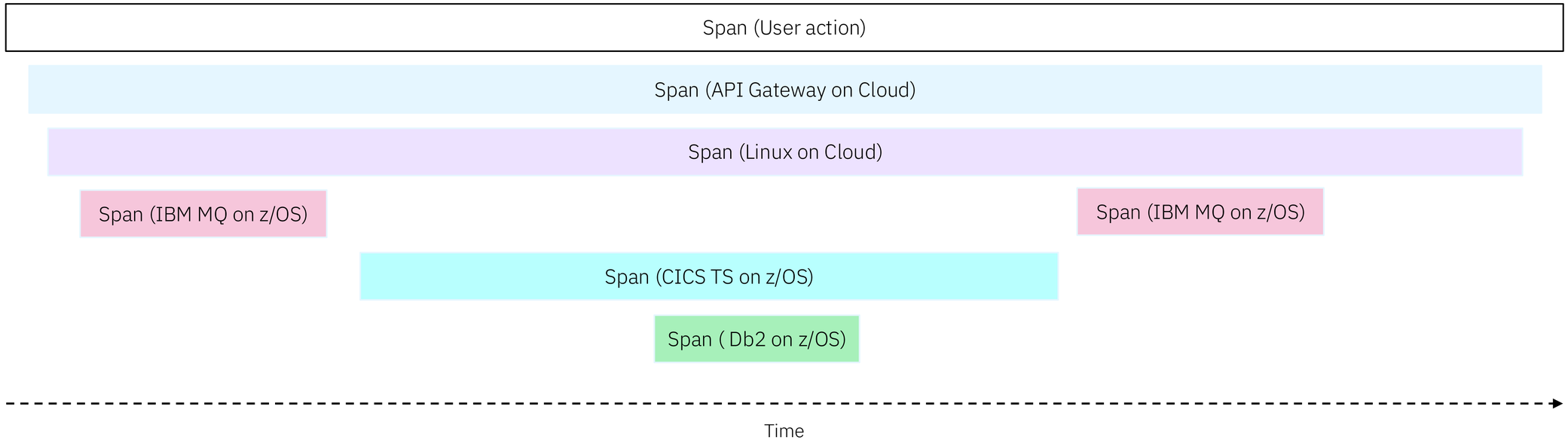
CICS TS 6.3 introduces a zero-code solution for emitting OpenTelemetry trace data.

Trace data is aggregated and can be explored using an Observability Backend

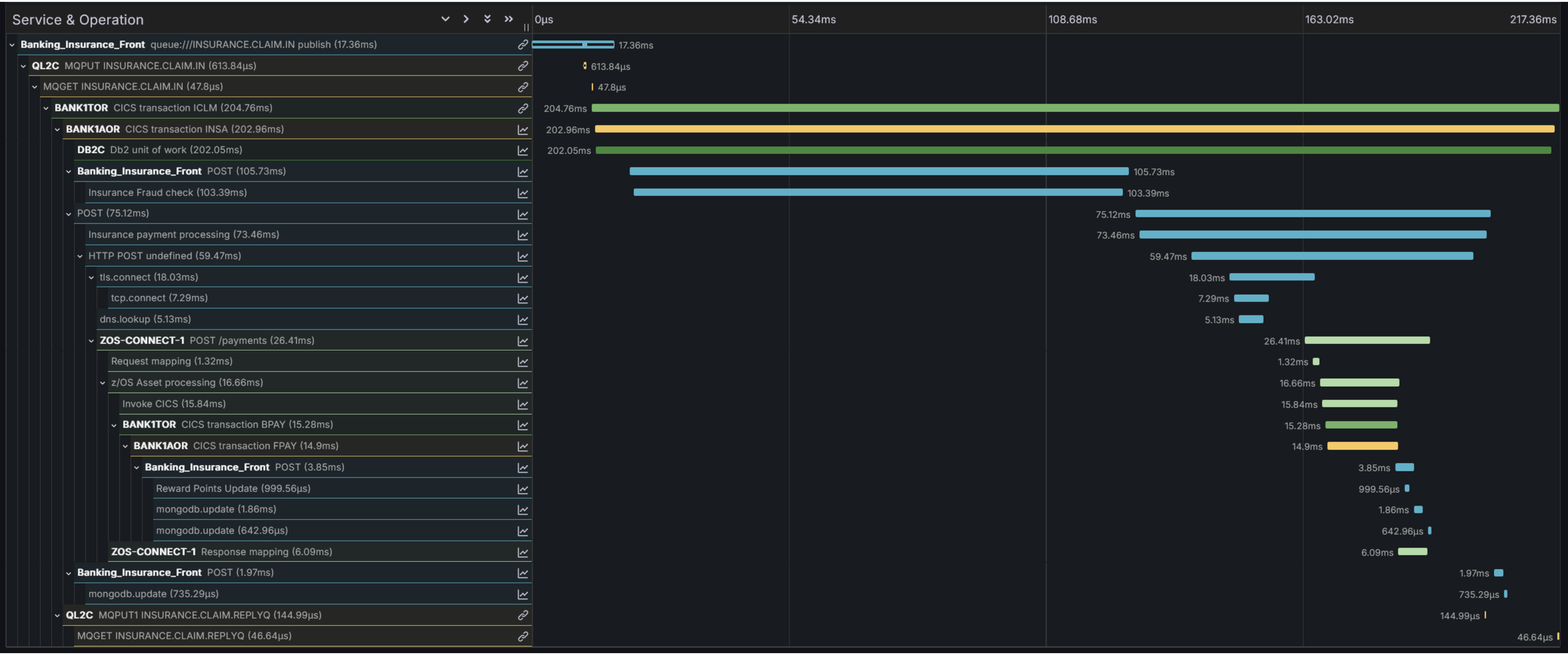
Spans



For a reques[illegible], you can see its e[illegible] through y[illegible] distributed application, the hier[illegible]cation of s[illegible] how long it spent in each service, and additional meta data associated with each service.



This OTel tracing shows the flow of processing for one insurance claim request

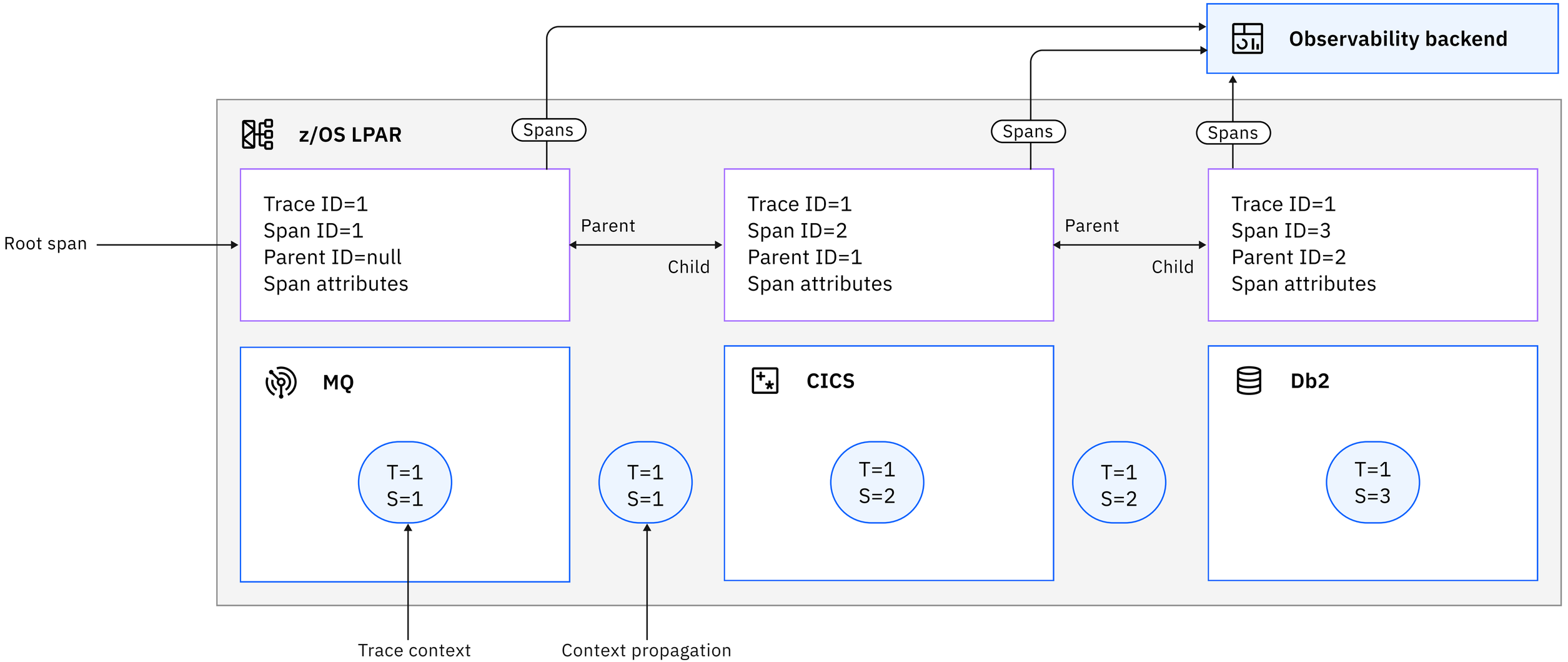


How are spans connected together?

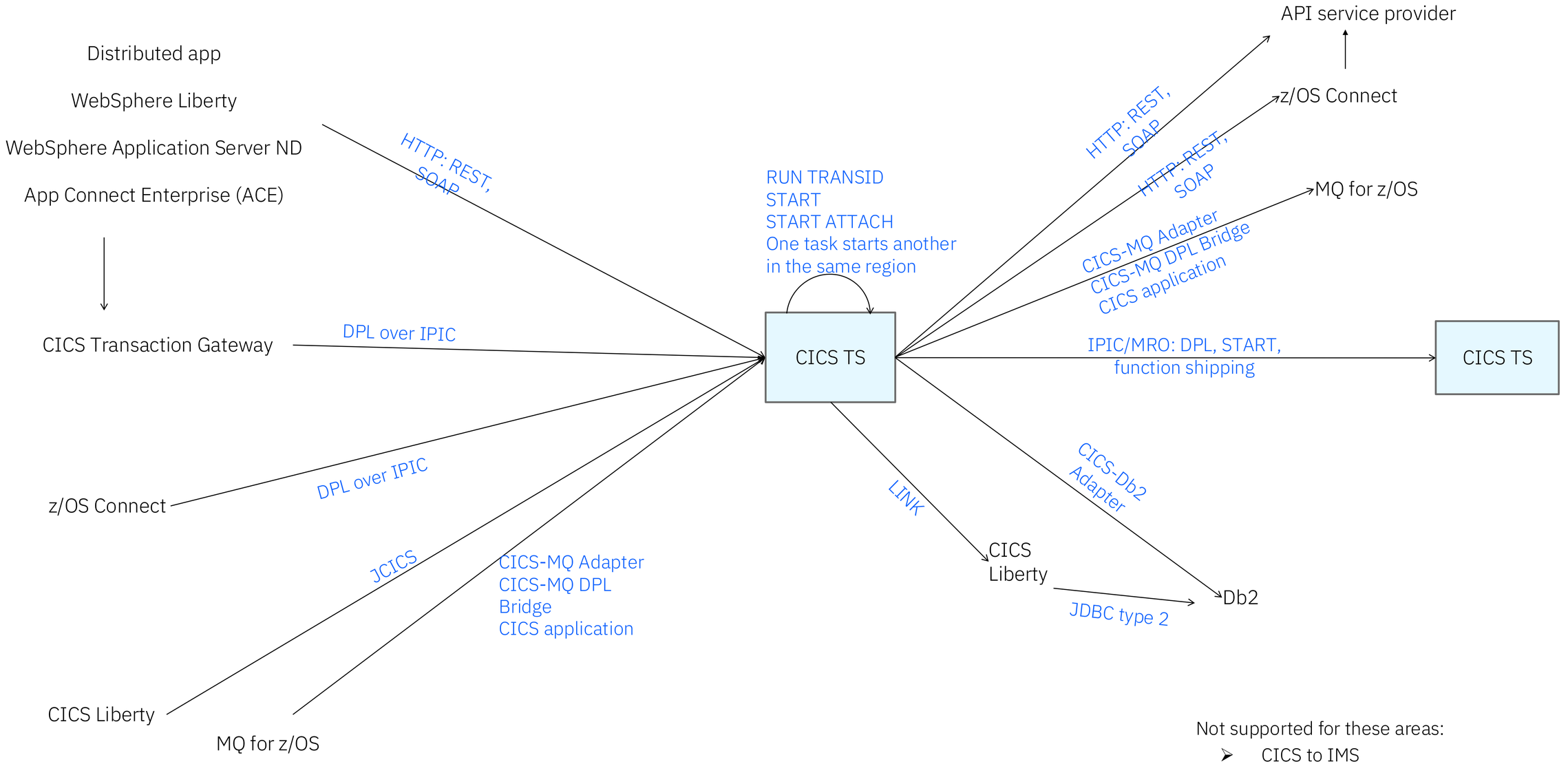
Each tracing participant propagates **traceparent** to the next system

- example 00-0af7651916cd43dd8448eb211c80319c-b7ad6b7169203331-01
- **traceparent** is contains
  - version – 1 byte
  - the trace-id - 16 bytes
    - uniquely identify a request with a distributed trace through a system
    - The trace-id is used by the OTel collector to identify spans which belong to the same trace
  - the parent-id – 8 bytes
    - ID of this request as known by the caller
  - the trace-flags – 1 byte
    - The right-most bit is a flag called sampled. When set, denotes that the caller may have recorded trace data

How are spans connected together?



OTel trace propagation – CICS TS 6.3



Not supported for these areas:

- CICS to IMS
- Between CICS regions over APPC and LU61
- DTP (distributed transaction processing)
- EXEC CICS RETURN TRANSID

Propagating OTel trace to and from CICS TS

CICS can propagate OTel traceparent in and out for:

- ✓ With-in the same region tasks

- ✓ Over HTTP, SOAP, REST
- ✓ EXEC CICS START over IPIC and MRO
- ✓ EXEC CICS LINK PROGRAM over IPIC and MRO
  - This also enable callers such as z/OS Connect EE and CTG
- ✓ Function-shipping requests over IPIC and MRO
- ✓ LINK3270
- ✓ Web service transaction routing (within a region and over MRO)

- ✓ CICS JVM
- ✓ CICS to/from MQ
- ✓ CICS to Db2

Propagation to these are not supported:

- IMS
- Over APPC and LU61
- DTP (distributed transaction processing)
- EXEC CICS RETURN TRANSID

Configure OTel tracing at the region level

SIT parameter OTELTRACE=(YES|NO)

Default is NO

- With NO - tasks in CICS region do not participate OTEL tracing, regardless of the transaction setting. This provides for a predictable upgrade experience.
- With YES – non-system tasks in the CICS region are eligible to participate OTEL tracing, subject to its transaction setting.

The region level setting is cataloged so its value is preserved over CICS warm start.

The region level setting can be inquired and set dynamically:

- INQUIRE OTEL TRACE
- SET OTEL TRACE(NOOTELTRACE|OTELTRACE)
  - Setting from OTELTRACE to NOOTELTRACE, any accumulated span data for tasks already completed are sent immediately to SMF; any active task span data will not be accumulated.
- CICS also sends out span data every 5 seconds.

Configure OTel tracing at the transaction level

Transaction new attributes:

OTELPROP=(Noprop| Proponly)

OTELEMIT=(Noemit | Taskend)

- The settings are cataloged so its value are preserved over CICS warm start.
- They can be inquired and set dynamically.

```
CEDA  View TRANSaction( OTI2 )
 RESSec          : Yes                  No Yes
 CMdsec          : Yes                  No Yes
 Extsec          : No                   No | Yes
 TRANSec         : 01                   1-64
 RSl             : 00                   0-24 | Public
OTEL
 OTELProp        : Proponly             Noprop | Proponly
 OTELEmit        : Taskend              Noemit | Taskend
DEFINITION SIGNATURE
 DEFinetime      : 24/06/24 09:26:44
 CHANGETime      : 05/09/24 11:26:31
 CHANGEUsrid     : CICSUSER
 CHANGEAGEnt     : CSDApi               CSDApi | CSDBatch
 CHANGEAGRel     : 0760
```

Configure OTel tracing at the transaction level - 2

Trace propagation

- OTELPROP=(Noprop| Proponly)
- Proponly: if received traceparent, CICS will process it for the task, generate a span id for this task, and pass to the next task or next system.
- Noprop: CICS will not process incoming traceparent, so nothing will be passed to the next task/system.

Span data generation

- OTELEMIT=(Noemit | Taskend)
- Taskend: if task has OTel trace and the OTel trace flag is set to sampling, CICS will generate span data to SMF type 1159 when the task ends.
- Noemit: if task has OTel trace, CICS will not generate span data to SMF type 1159. Propagation to the next one still happens.

Summary of behaviour

Region OTELTRACE	OTELProp	OTELEmit	OTel trace propagated?	Span data generated?
YES	Noprop	Noemit	No	No
	Noprop	Taskend	No	No *
	Proponly	Noemit	Yes	No
	Proponly	Taskend	Yes	Yes when sampling

* For span data to be generated, traceparent must be received inbound.

Scenarios of using CICS OTel trace configurations

Migrate from lower CICS release

- Default SIT OTELTRACE is NO
- This means no OTEL tracing so no impact when migrating this release, regardless transaction setting.

Join corporate-wide OTel tracing

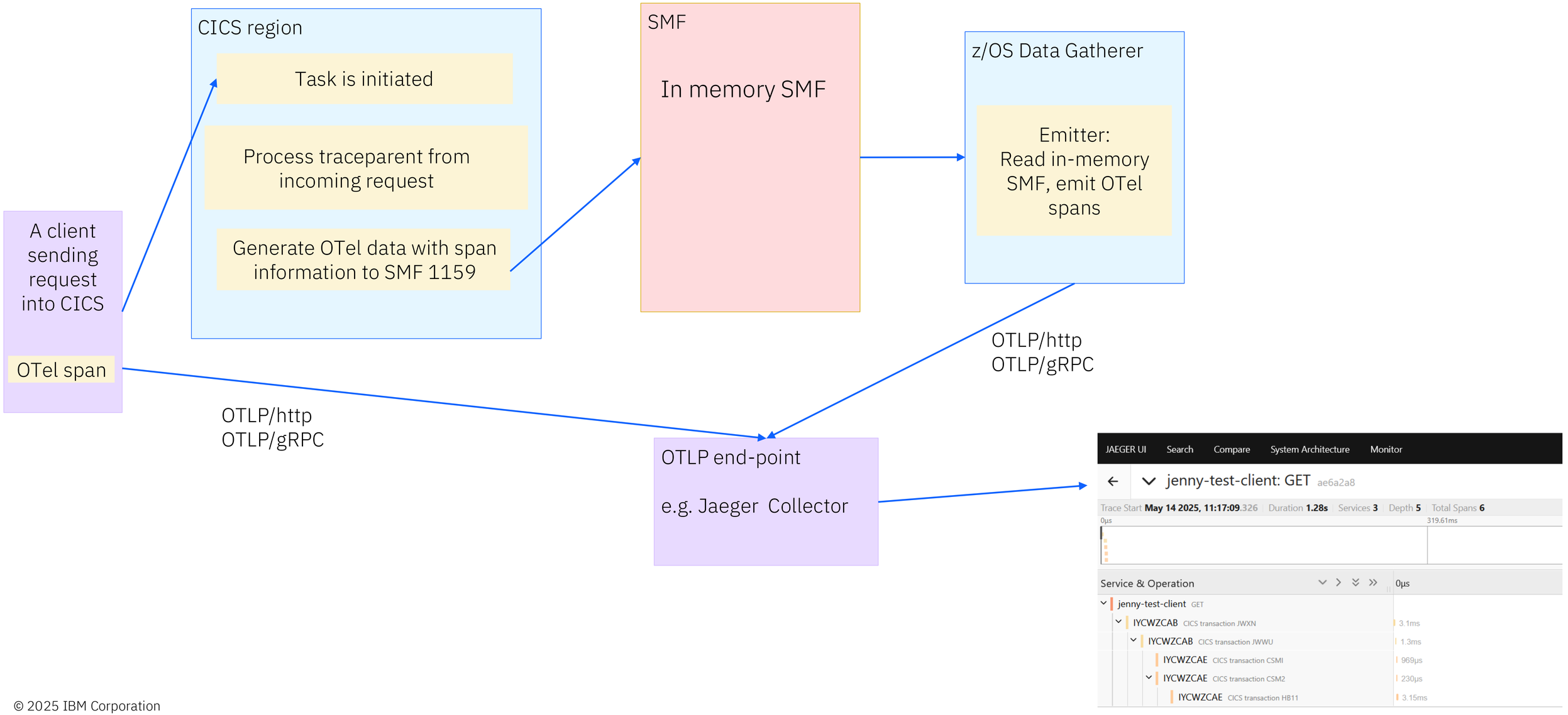
- Keep the default transaction attribute set to Proponly , Taskend
- Only need to change SIT OTELTRACE to YES
- OTel trace will be propagated by CICS from those requests carrying them, and span data will be emitted if sampling flag is set from client.
- Not impacting those requests not carrying OTel trace.

Limit the volume of SMF data

- SIT OTELTRACE = YES
- 3 levels of control:
  - Noprop - No OTel trace will be propagated and no span data will be emitted.
    - This means this task and its following-on will not show up in observability tool.
  - Proponly, Noemit - OTel trace will be passed only if received. No span data will be emitted.
    - This means this task will not show up in observability tool.
  - Proponly, Taskend - Client trace will be passed only if received. Span data is emitted unless:
    - If sampling flag is set to 0, no otel trace show up in observability tool.

Consistent user-experience with other zOS subsystems and OpenTelemetry SDK

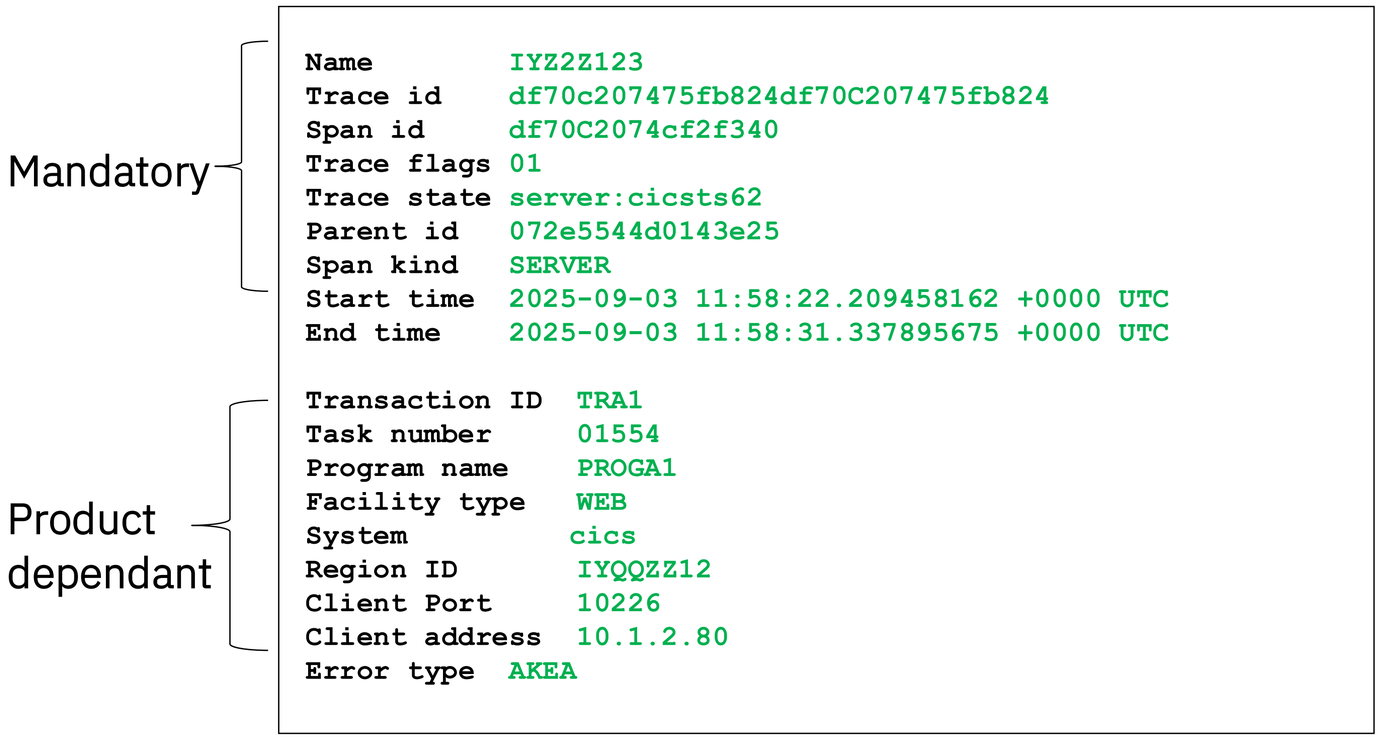
What’s involved to generate OTel trace for CICS region



OTel span for a CICS task

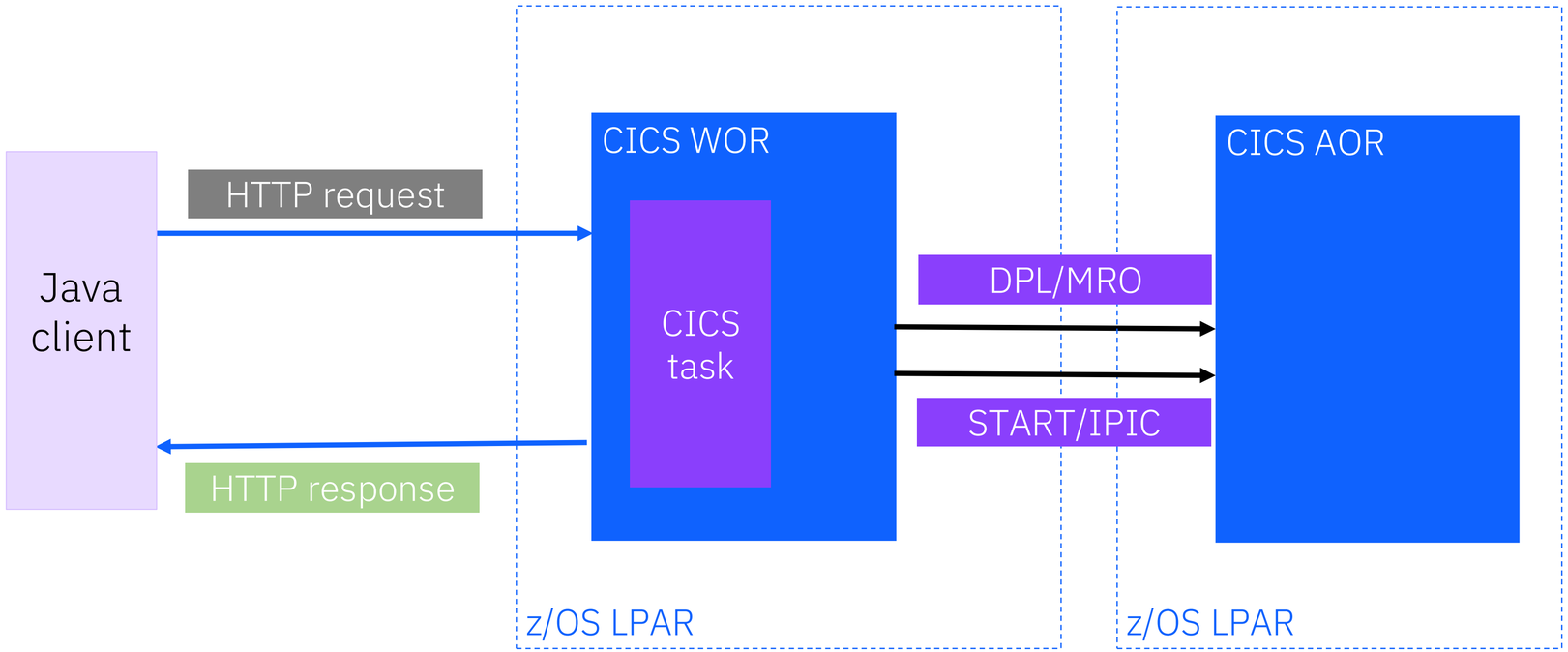
Spans must contain certain “mandatory” items

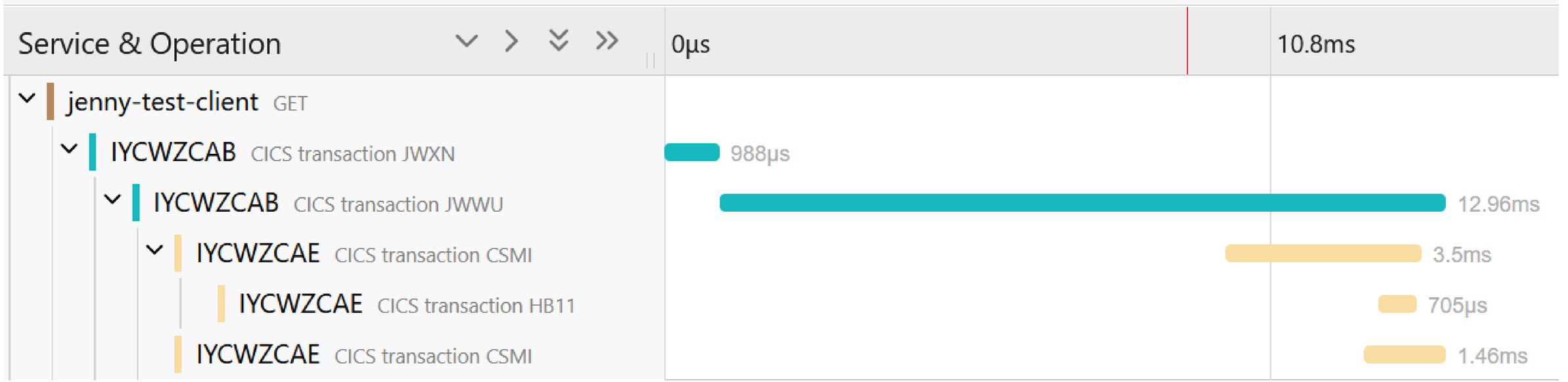
Spans can contain product determined items



For the latest span fields see CICS Doc

OTel Tracing – HTTP request with cross-region scenario





CICS transaction JWXN

Tags

client.address	9.69.254.228
client.port	59890
otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.facility.type	SOCKET
tps.program.name	DFHWBXN
tps.region.id	IYCWZCAB
tps.system	cics
tps.task.id	00065
tps.transaction.id	JWXN

CICS transaction JWWU

Tags

client.address	9.69.254.228
client.port	59890
otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.facility.type	WEB
tps.program.name	DFHWBA
tps.region.id	IYCWZCAB
tps.system	cics
tps.task.id	00066
tps.transaction.id	JWWU

CICS transaction CSMI

Tags

client.address	9.20.5.0
client.port	63527
otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.connection.name	IB2A
tps.facility.type	IPIC
tps.program.name	DFHMIRS
tps.region.id	IYCWZCAE
tps.system	cics
tps.task.id	00059
tps.transaction.id	CSMI

CICS transaction HB11

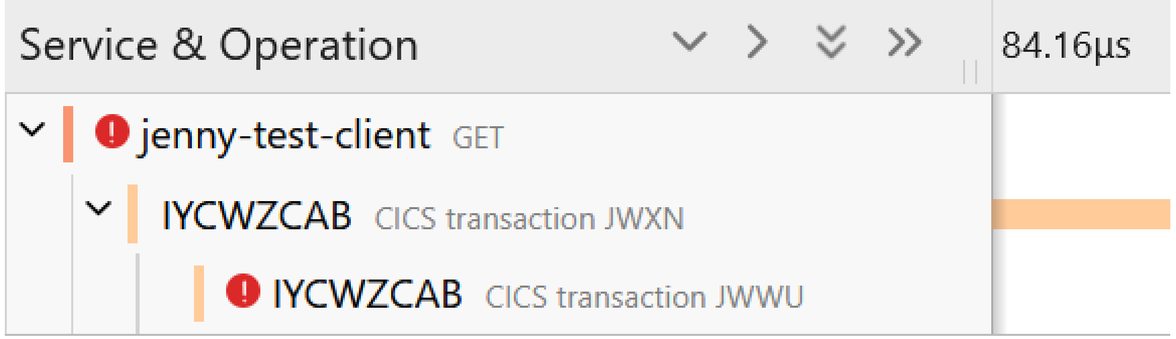
Tags

otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.facility.type	START
tps.program.name	NULLPROG
tps.region.id	IYCWZCAE
tps.system	cics
tps.task.id	00061
tps.transaction.id	HB11

CICS transaction CSMI

Tags

otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.connection.name	MB2A
tps.facility.type	MRO
tps.program.name	DFHMIRS
tps.region.id	IYCWZCAE
tps.system	cics
tps.task.id	00060
tps.transaction.id	CSMI



GET

Service: **jenny-test-client** | Dur

Tags

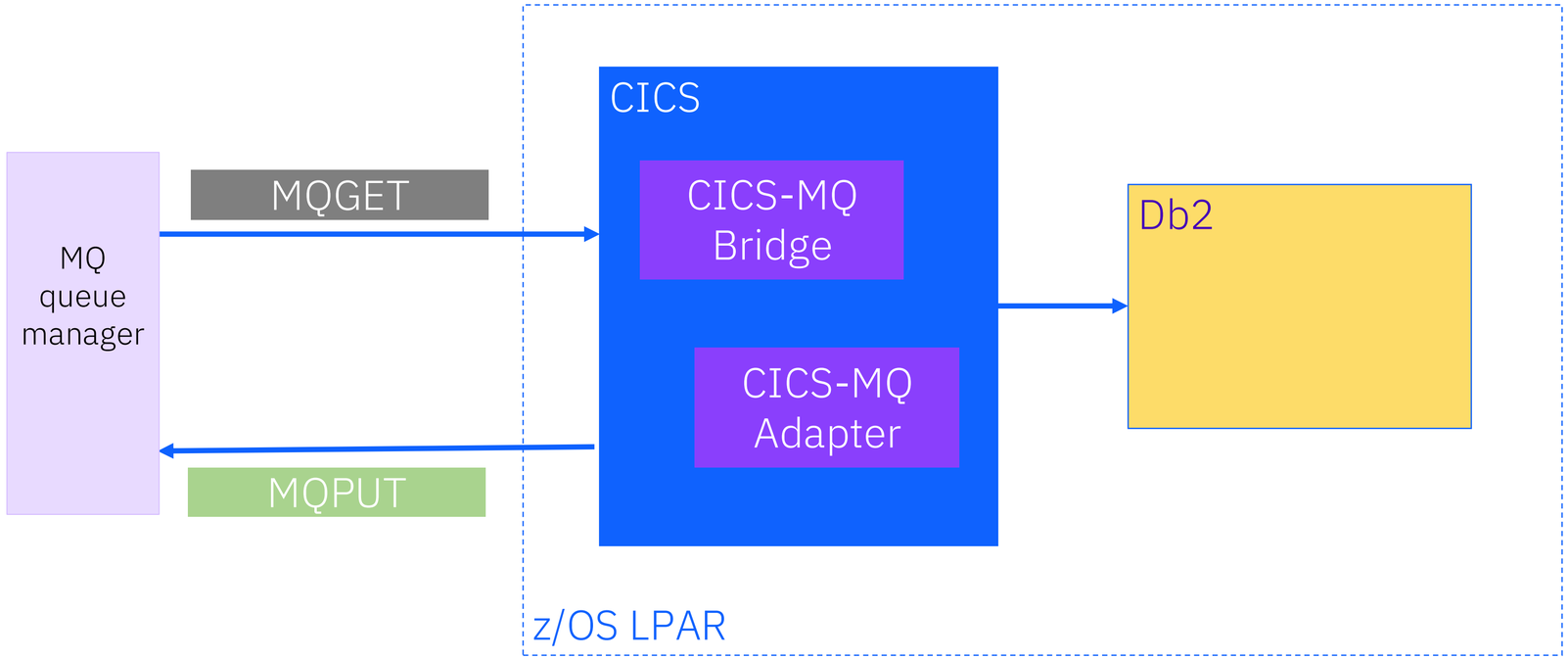
error	true
error.type	404
http.request.method	GET
http.response.status_code	404
network.protocol.version	1.1
otel.scope.name	io.opentelemetry.java-http-client
otel.scope.version	2.7.0-alpha
otel.status_code	ERROR
server.address	winmvs2c.hursley.ibm.com
server.port	10136
span.kind	client
thread.id	1
thread.name	main
url.full	http://winmvs2c.hursley.ibm.com:10136/OTEWEBP3/

CICS transaction JWWU

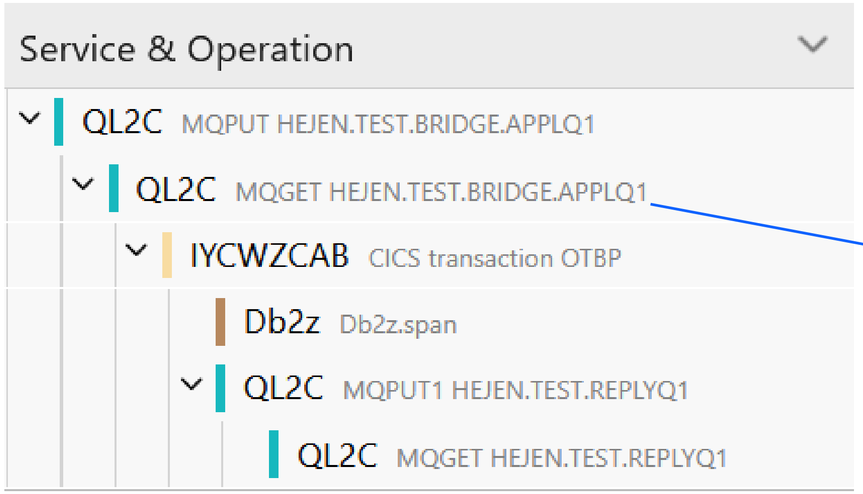
Tags

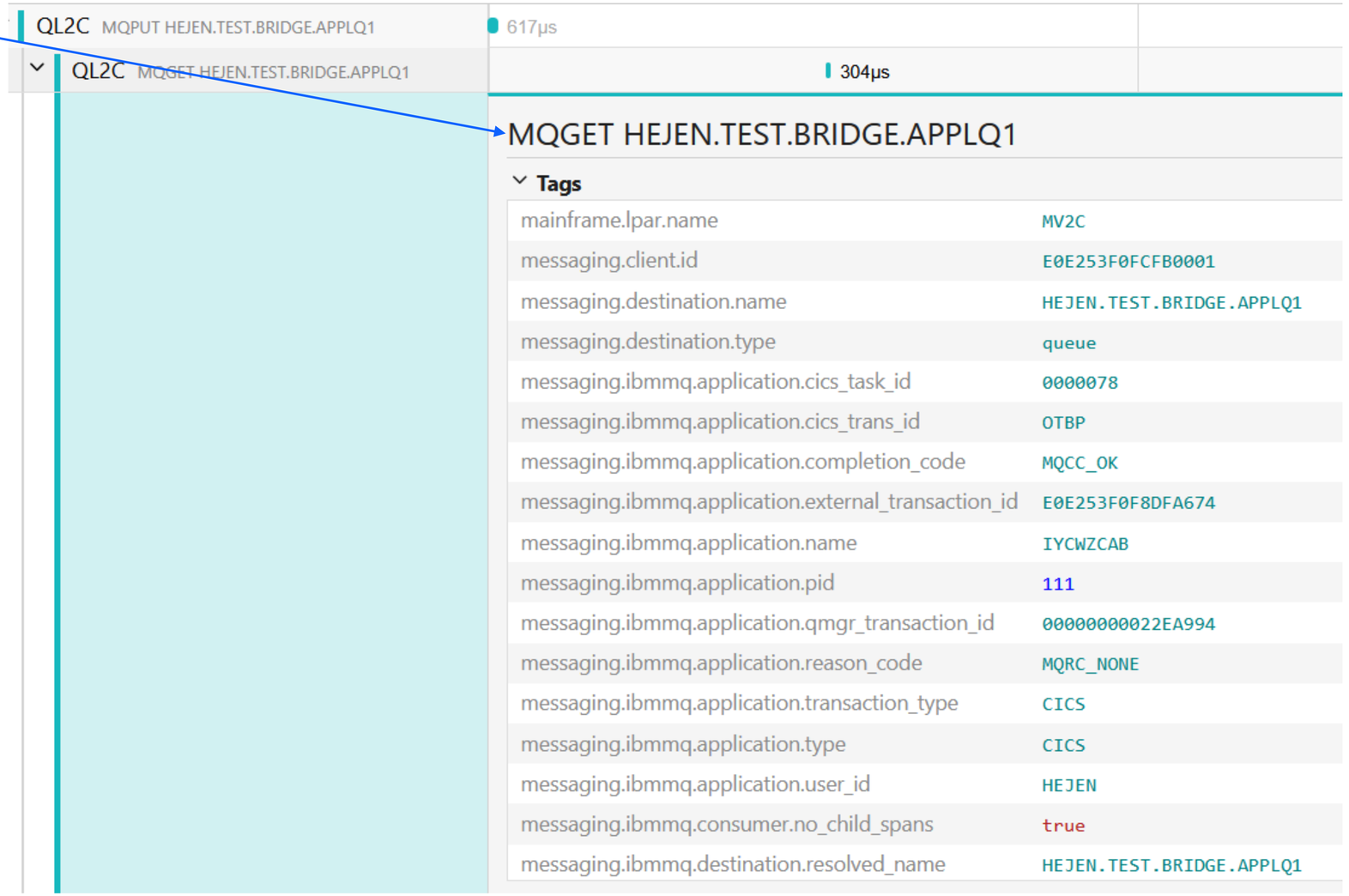
client.address	9.69.254.228
client.port	60377
error	true
error.type	AWBM
otel.scope.name	z/OS OTel
otel.scope.version	1.4.0-beta25
otel.status_code	ERROR
span.kind	server
tps.facility.type	WEB
tps.program.name	DFHWBA
tps.region.id	IYCWZCAB
tps.system	cics
tps.task.id	00083
tps.transaction.id	JWWU

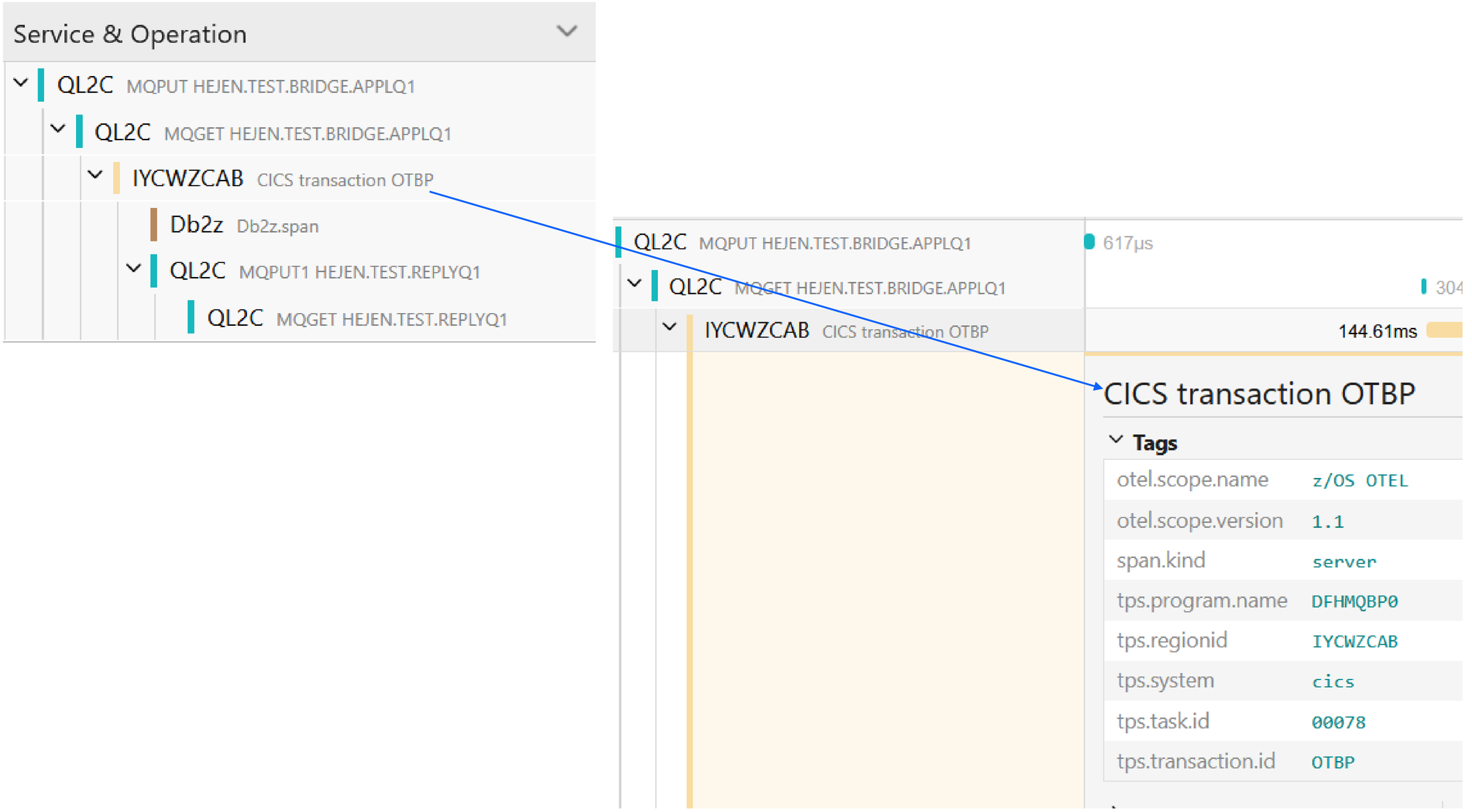
OTel Tracing – CICS, MQ and Db2 scenario

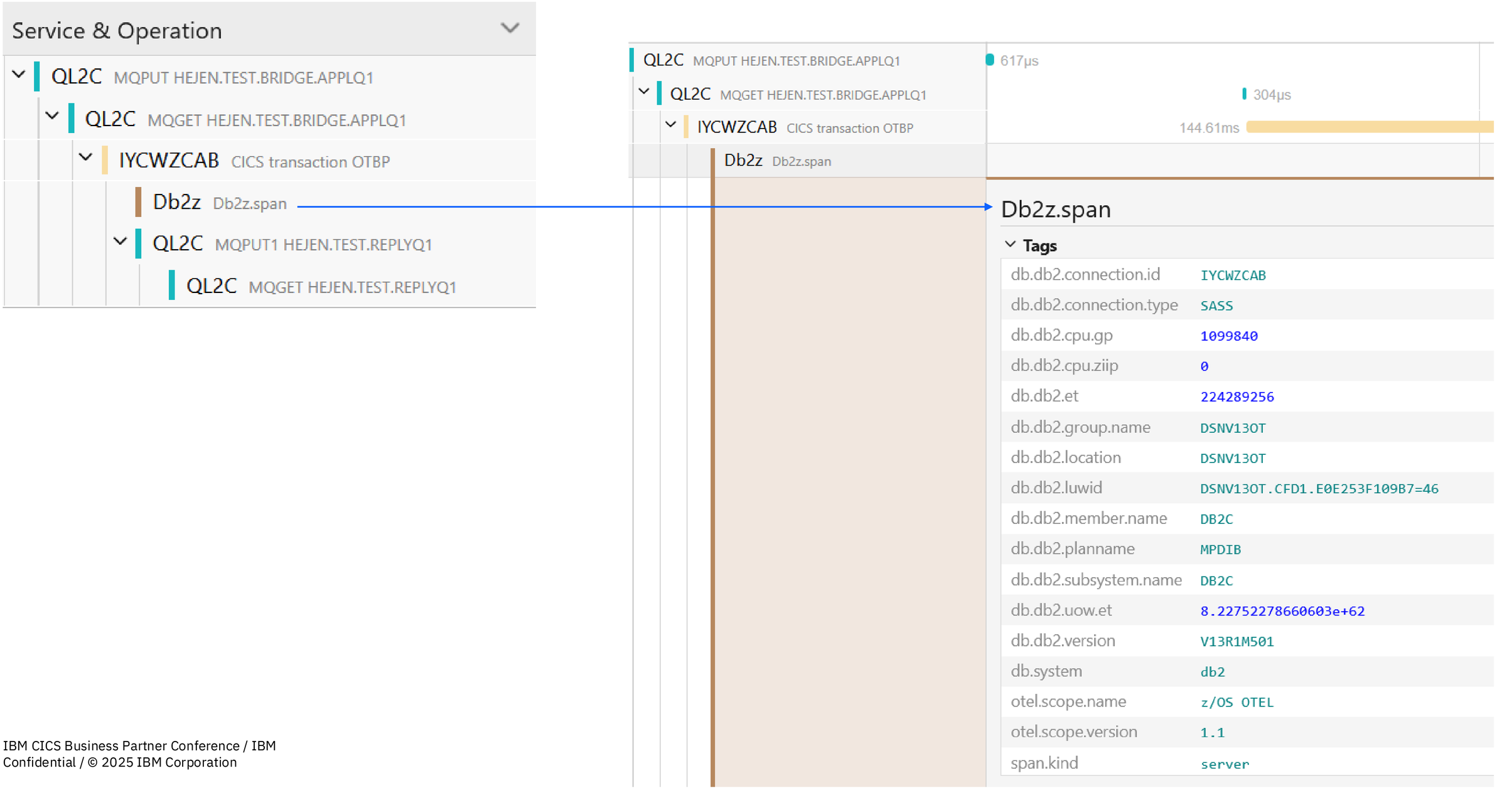


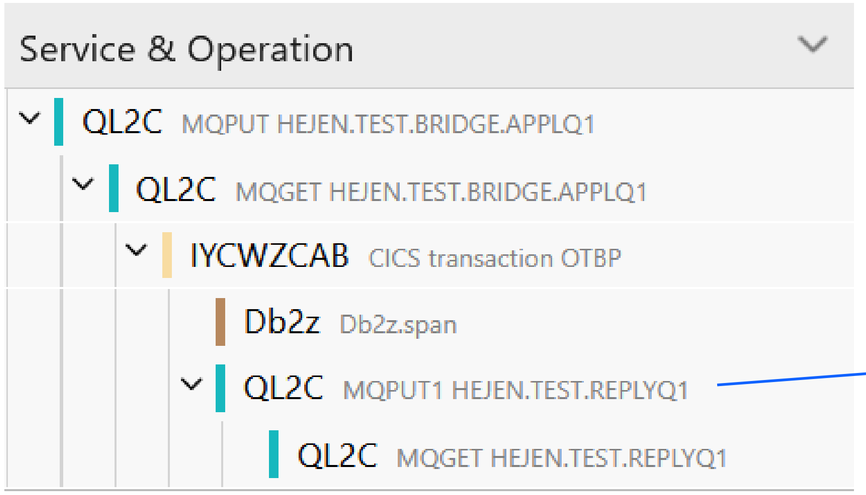
OTel Tracing – CICS, MQ and Db2 scenario

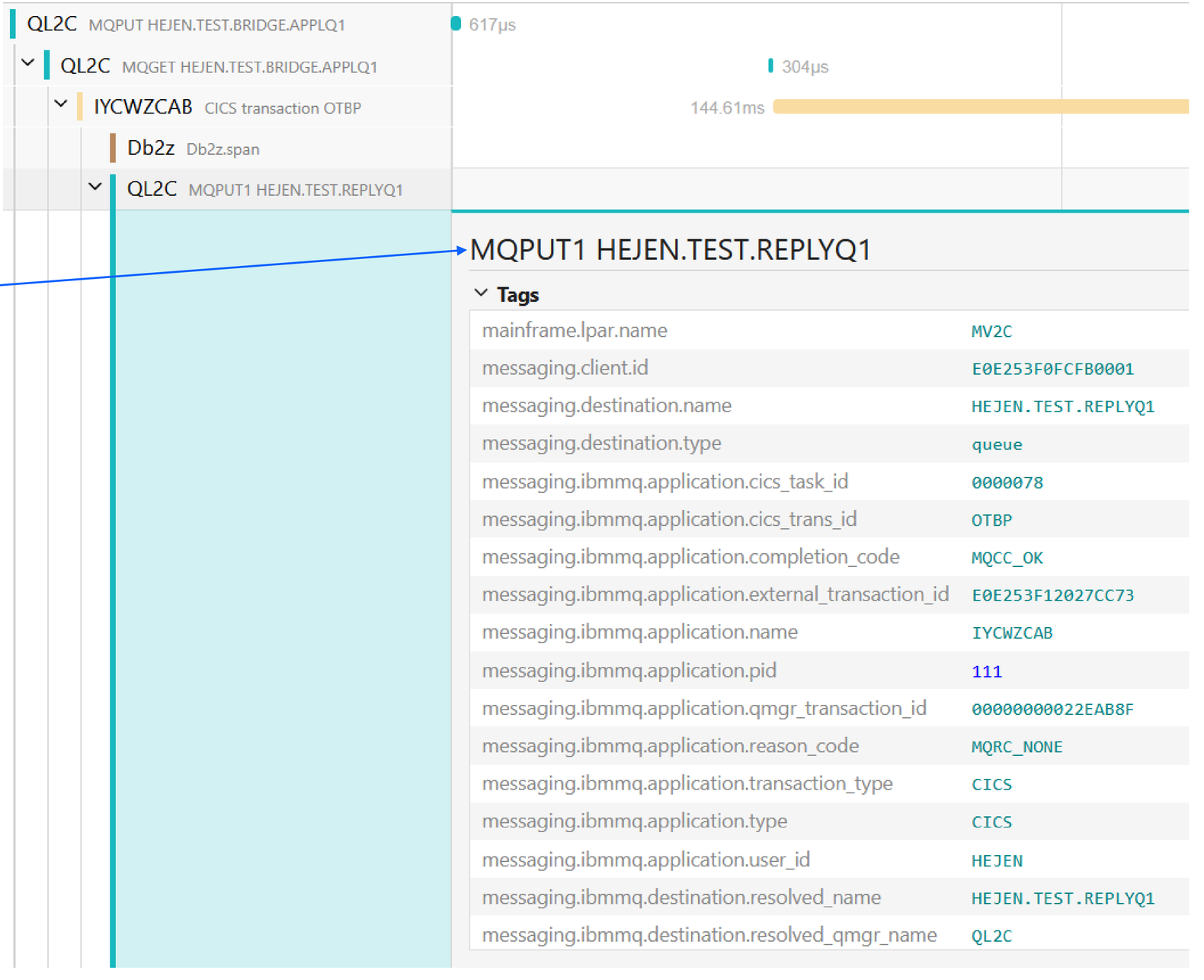





Service & Operation
QL2C MQPUT HEJEN.TEST.BRIDGE.APPLQ1
QL2C MQGET HEJEN.TEST.BRIDGE.APPLQ1
IYCWZCAB CICS transaction OTBP
Db2z Db2z.span
QL2C MQPUT1 HEJEN.TEST.REPLYQ1
QL2C MQGET HEJEN.TEST.REPLYQ1
QL2C MQPUT HEJEN.TEST.BRIDGE.APPLQ1
617µs
QL2C MQGET HEJEN.TEST.BRIDGE.APPLQ1
304
IYCWZCAB CICS transaction OTBP
144.61ms
CICS transaction OTBP
Tags
otel.scope.name z/OS OTEL
otel.scope.version 1.1
span.kind server
tps.program.name DFHMQBP0
tps.regionid IYCWZCAB
tps.system cics
tps.task.id 00078
tps.transaction.id OTBP


Service & Operation
QL2C MQPUT HEJEN.TEST.BRIDGE.APPLQ1
QL2C MQGET HEJEN.TEST.BRIDGE.APPLQ1
IYCWZCAB CICS transaction OTBP
Db2z Db2z.span
QL2C MQPUT1 HEJEN.TEST.REPLYQ1
QL2C MQGET HEJEN.TEST.REPLYQ1
QL2C MQPUT HEJEN.TEST.BRIDGE.APPLQ1
617µs
QL2C MQGET HEJEN.TEST.BRIDGE.APPLQ1
304µs
IYCWZCAB CICS transaction OTBP
144.61ms
Db2z Db2z.span
Db2z.span
Tags
db.db2.connection.id IYCWZCAB
db.db2.connection.type SASS
db.db2.cpu.gp 1099840
db.db2.cpu.ziip 0
db.db2.et 224289256
db.db2.group.name DSNV13OT
db.db2.location DSNV13OT
db.db2.luwid DSNV13OT.CFD1.E0E253F109B7=46
db.db2.member.name DB2C
db.db2.planname MPDIB
db.db2.subsystem.name DB2C
db.db2.uow.et 8.22752278660603e+62
db.db2.version V13R1M501
db.system db2
otel.scope.name z/OS OTEL
otel.scope.version 1.1
span.kind server





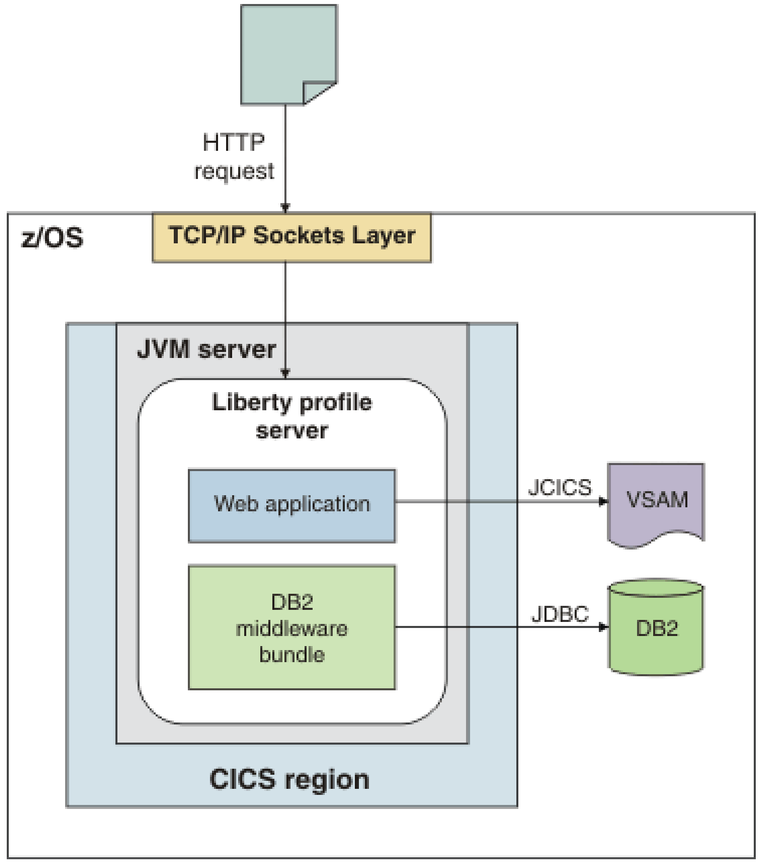
OpenTelemetry for CICS JVM server

OpenTelemetry for application running in CICS JVM server

As a mixed-language environment, more and more java application are hosted in CICS.

Benefits of OTel in CICS JVM servers:

- end-to-end distributed tracing across both CICS transactions and Java applications running in CICS JVM.
- gain visibility into how Java code contributes to overall transaction processing, including calls to external web services or databases.
- diagnose performance bottlenecks, understanding service dependencies, to improve reliability in complex, mixed-language CICS environments

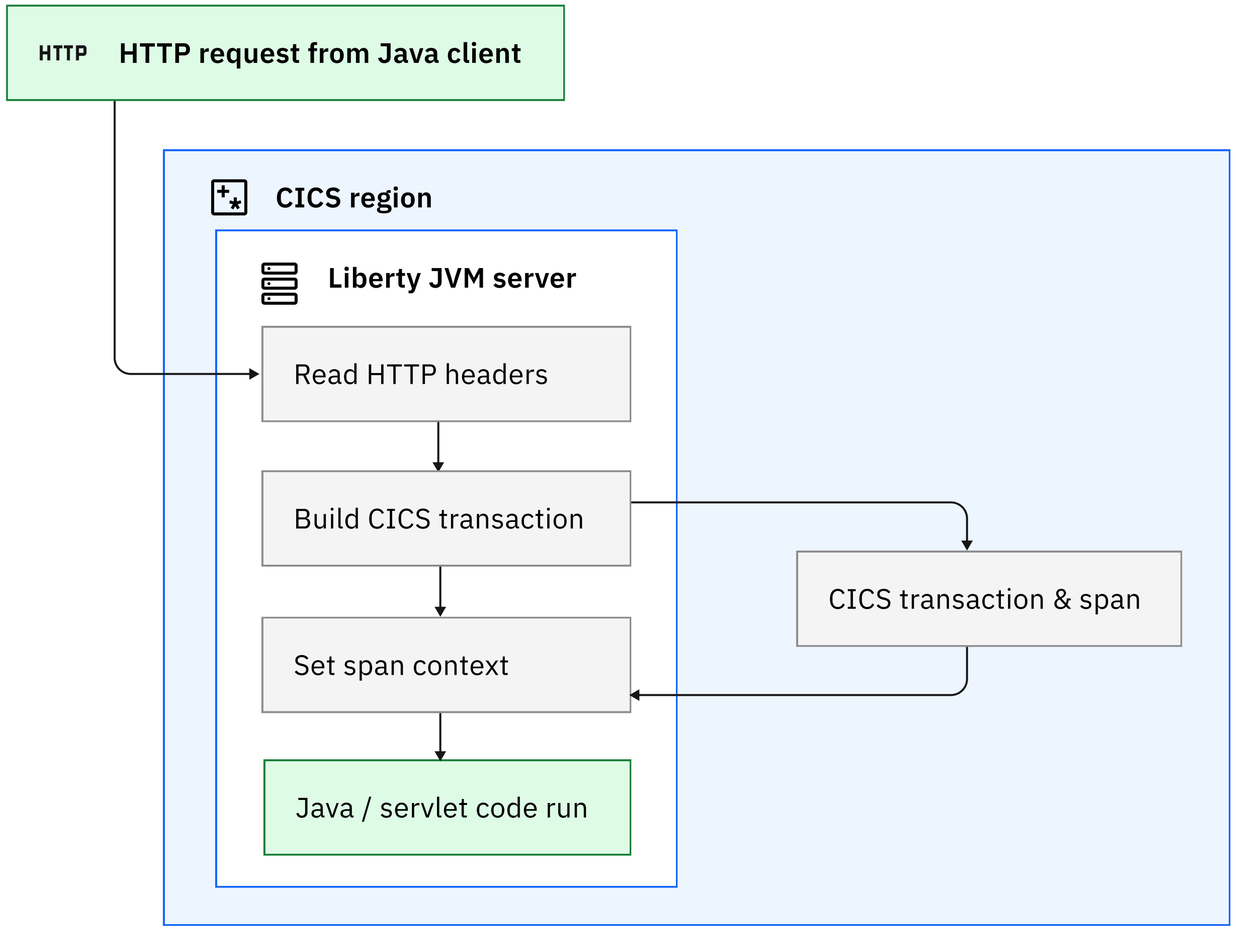


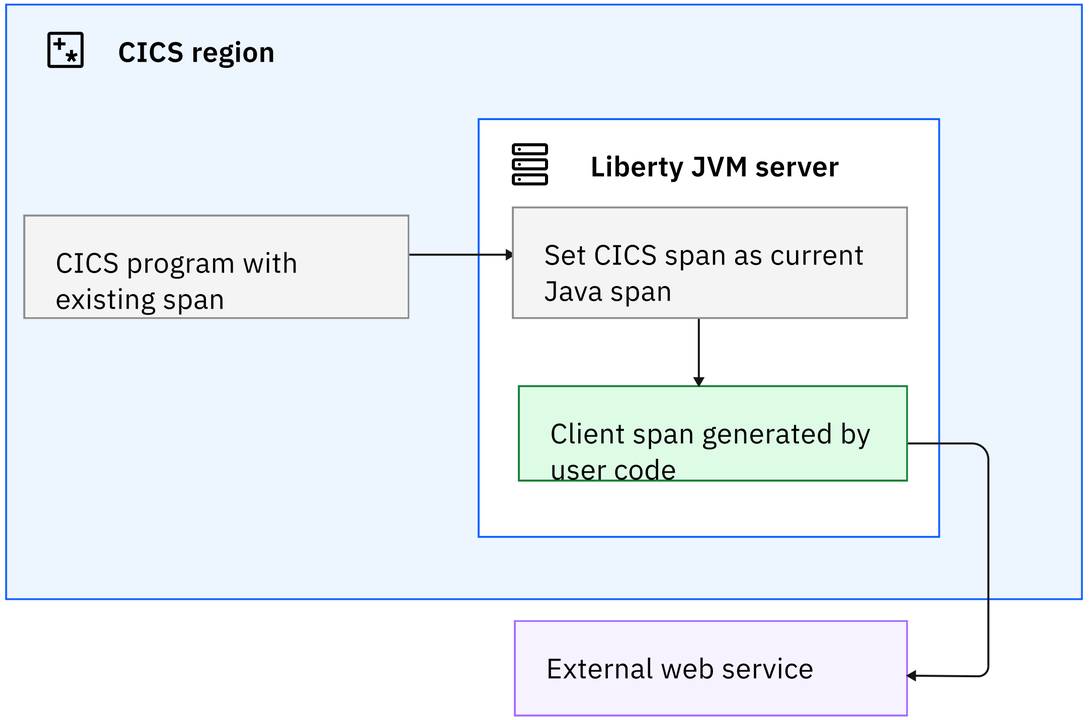
OpenTelemetry for application running in CICS JVM server

- Automatic propagate in/out from request into Java application to CICS task if the hosting CICS region participate in OpenTelemetry

- When region OTELTRACE level is ON,
  - CICS installs the OpenTelemetry Java APIs and SDK into all enabled JVM servers
  - Liberty JVM servers will then process the traceparent HTTP header on incoming request and propagating this context into the CICS transaction
  - CICS also adds extensions to the OTel Java API to get the current Java context from the CICS transaction, ready for outgoing requests

Automatic propagate trace context between hosted Liberty/OSGi JVM server and CICS task





Instrument JVM application for OTel

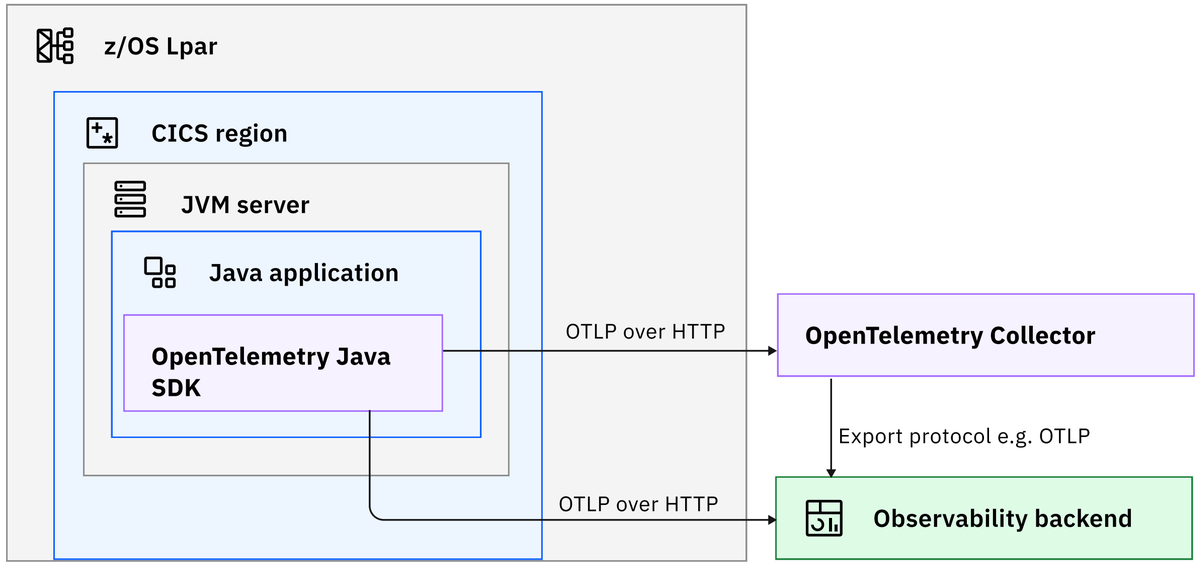
- Now CICS task has OTel trace, what about the work running on a java thread?
- These different ways can be used to instrument JVM application for OTel to create span
  1. Java Agent: zero-code, automatic, rich support with JDBC, JMS, HTTP requests etc.
     - Use this for Liberty JVM server
     - spans created by it do not use the CICS span as parent which breaks correlation of spans
  2. Manual: flexible control.
     - This is the best option in OSGi JVM servers as it inherits CICS task as parent
  3. MicroProfile Telemetry
     - For Liberty only.
     - spans created by it does not use the CICS span as parent which breaks correlation of spans

Example of manual instrumentation in JVM application

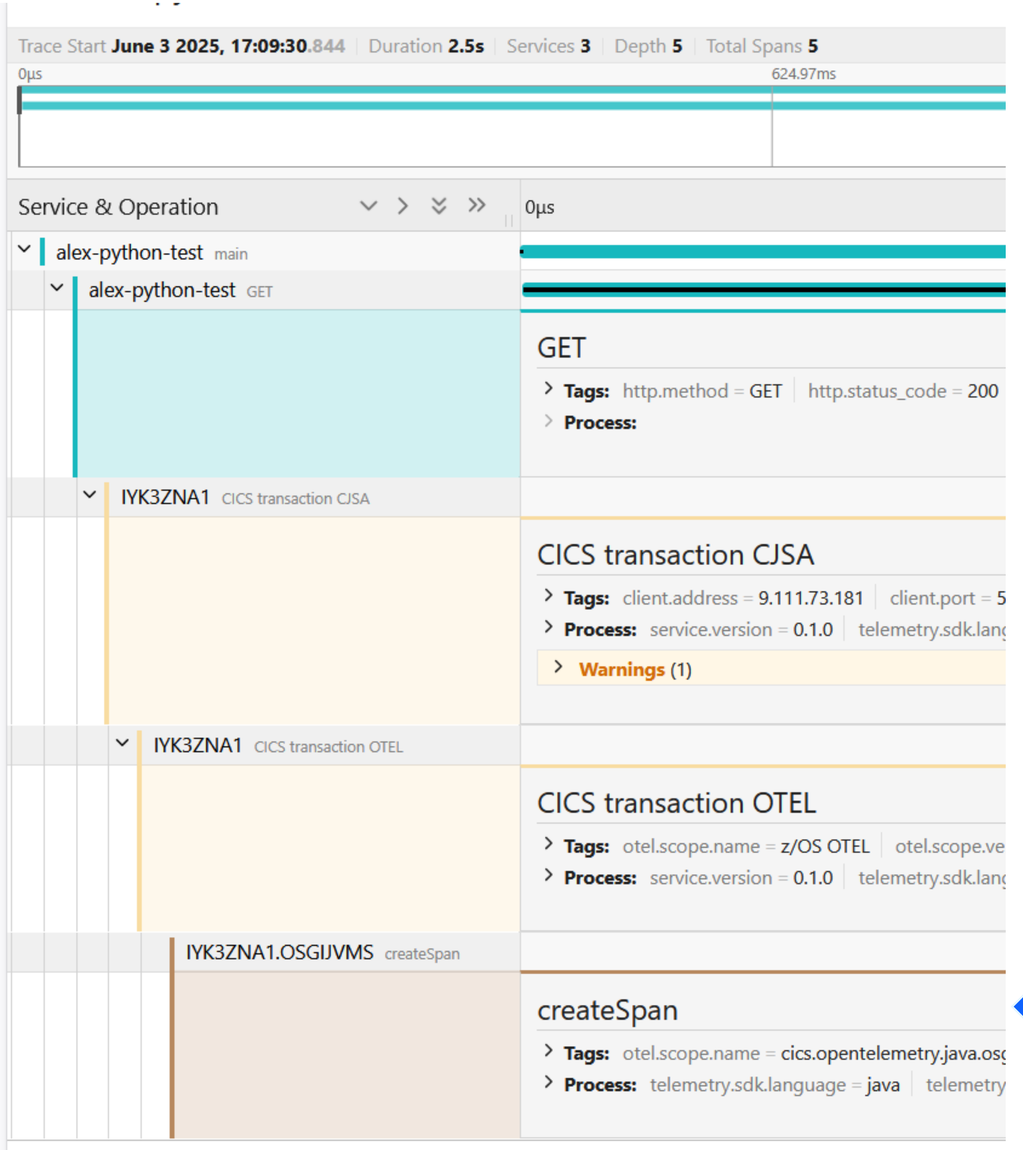
```
public class CICSJavaOTel
[
    public static final OpenTelemetry OPEN_TELEMETRY = AutoConfiguredOpenTelemetrySdk.initialize()
        .getOpenTelemetrySdk();
    private static final String SCOPE_NAME = "com.example.CICSJavaOTel";

    public void usage()
    {
        Tracer tracer = OPEN_TELEMETRY.getTracer(SCOPE_NAME);

        Span span = tracer.spanBuilder("usage").startSpan();
        try(Scope scope = span.makeCurrent())
        {
            // Traced business logic here.
        }
    }
}
```

Example configuration in JVM profile

```
# Configure the OpenTelemetry Java Agent
-javaagent:&USSHOME;/lib/opentelemmetry/opentelemetry-java-agent.jar

# Configure for sending OTel spans:
# Set the service name for the JVM server
-Dotel.service.name=MyCICSJavaApp

# Export traces using OTLP
-Dotel.traces.exporter=otlp

# Export with OTLP HTTP
-Dotel.exporter.otlp.protocol=http/protobuf

# The OTLP collector to export to
-Dotel.exporter.otlp.endpoint=http://otel.example.com:4318
```



Scenario: Python client (HTTP to)--> CICS Liberty (Link to)--> OSGi program issue AsyncService (RUN TRANSID) --> OTEL transaction is started which is a OSGi program that creates a span in Java



CICS transaction CJSA

Tags

client.address	9.111.73.181
client.port	58294
otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.facility.type	JVMSERVER
tps.program.name	DFHSJTHP
tps.region.id	IYK3ZNA1
tps.system	cics
tps.task.id	00277
tps.transaction.id	CJSA

CICS transaction OTEL

Tags

otel.scope.name	z/OS OTEL
otel.scope.version	1.1
span.kind	server
tps.facility.type	ASRUNTRAN
tps.program.name	NEWSPAN
tps.region.id	IYK3ZNA1
tps.system	cics
tps.task.id	00278
tps.transaction.id	OTEL

createSpan

Tags

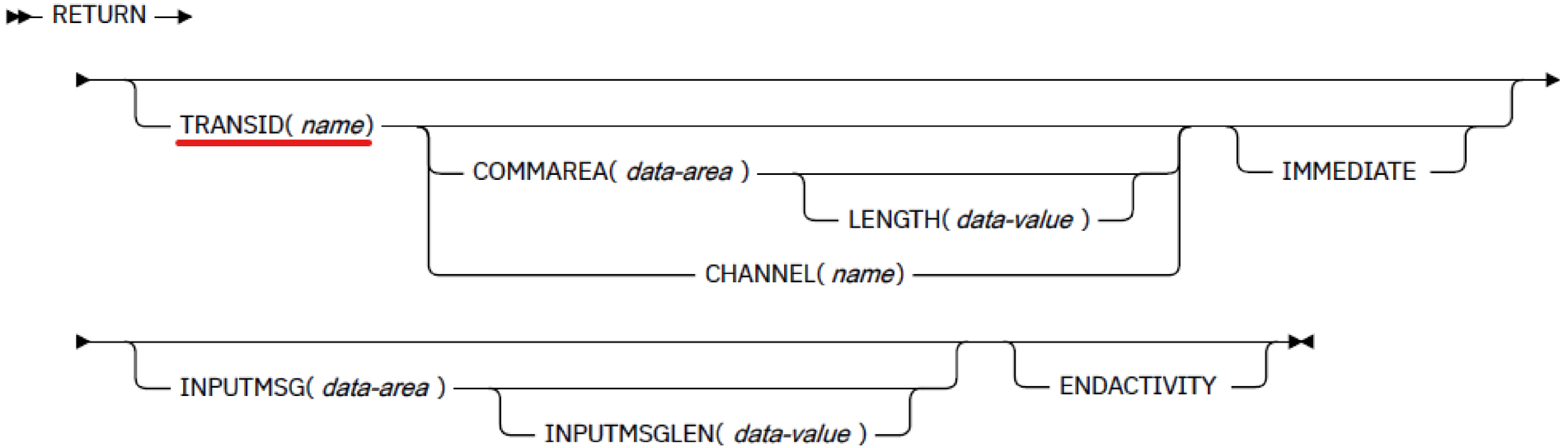
otel.scope.name	cics.opentelemetry.java.osgi.App
span.kind	internal

CICS TS open beta

RETURN TRANSID() command

- EXEC CICS RETURN
  - Most common usage, return control to the next higher logical level.
  - Within same task boundary, it generally has minimal impact on OTel.
  - But, this API provides many variations and options beyond this basic usage.

RETURN

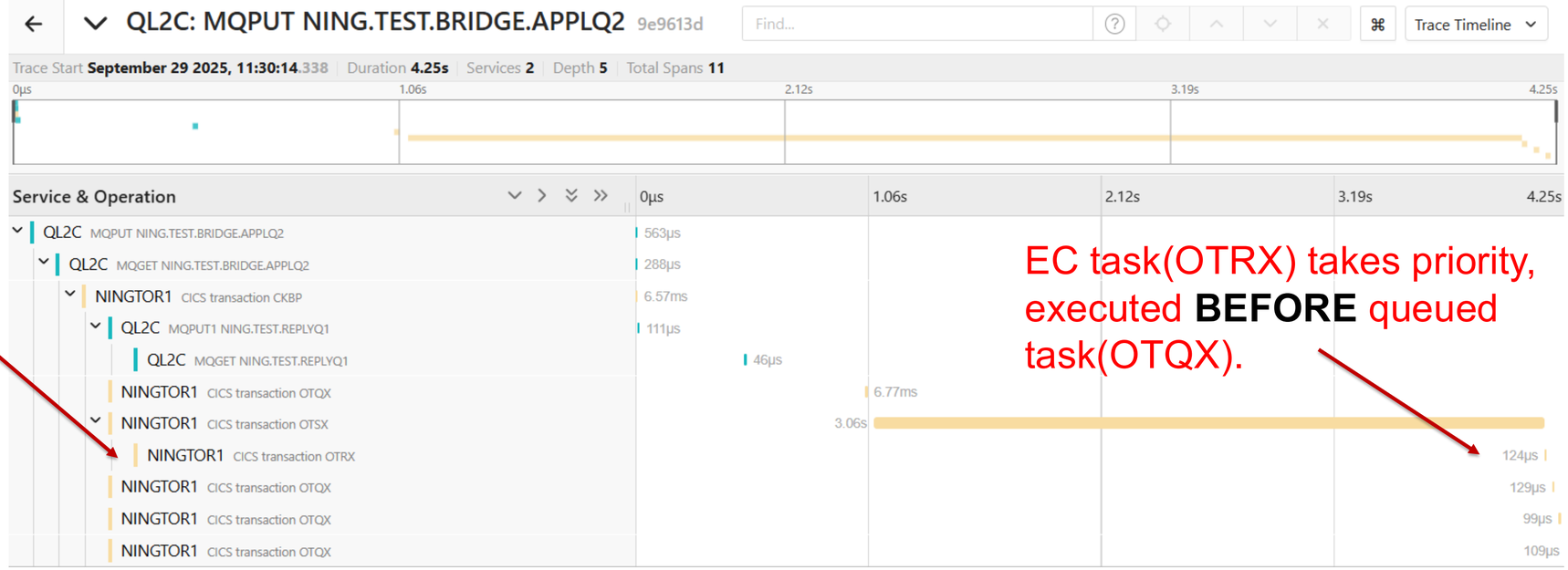


- EXEC CICS RETURN **TRANSID** ….
  - The task returns control to CICS
  - CICS starts a **new task** – TRANSID on the terminal (of calling task)
  - Pseudo conversation between tasks – passing data from one task to the EXEC CICS RETURN task with inputmsg, commarea, or channel

No behavioural changes, OTel context propagation has been added

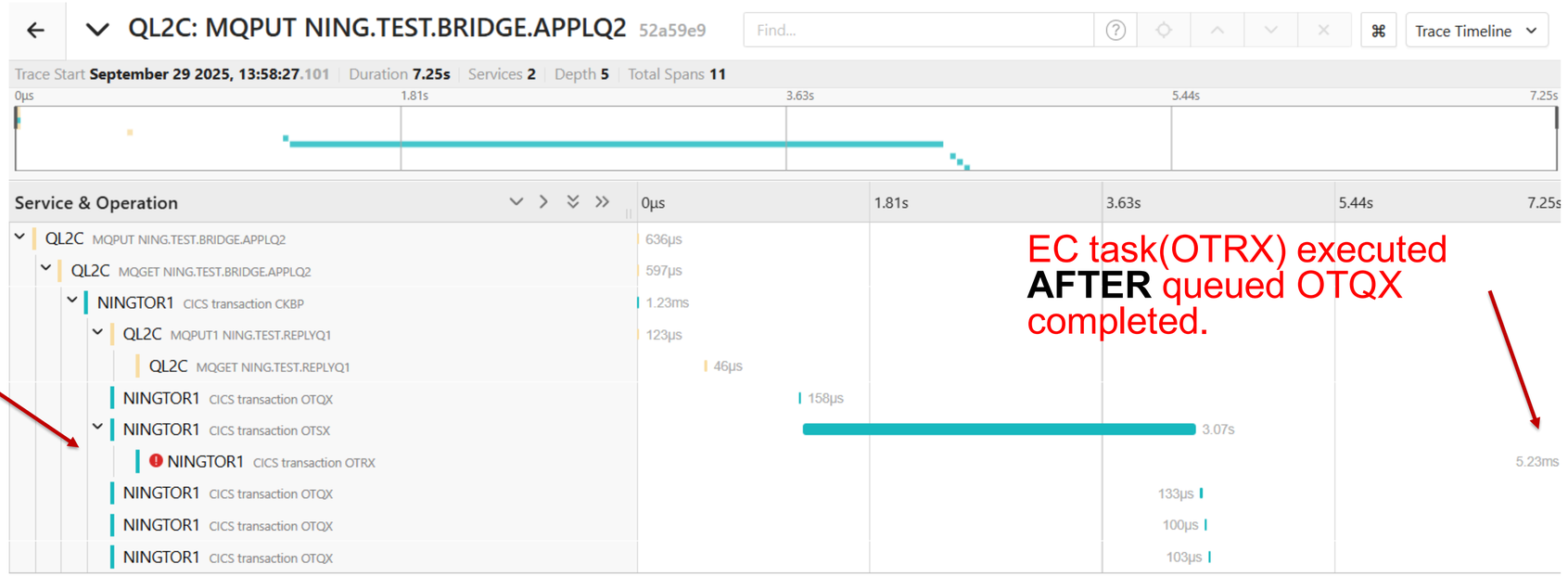
- From calling task to newly started task
- Propagated to remote region if connection is MRO or IPIC

EC return task(OTRX) – IMMEDIATE specified



EC task(OTRX) takes priority, executed **BEFORE** queued task(OTQX).

EC return task(OTRX) – IMMEDIATE NOT specified

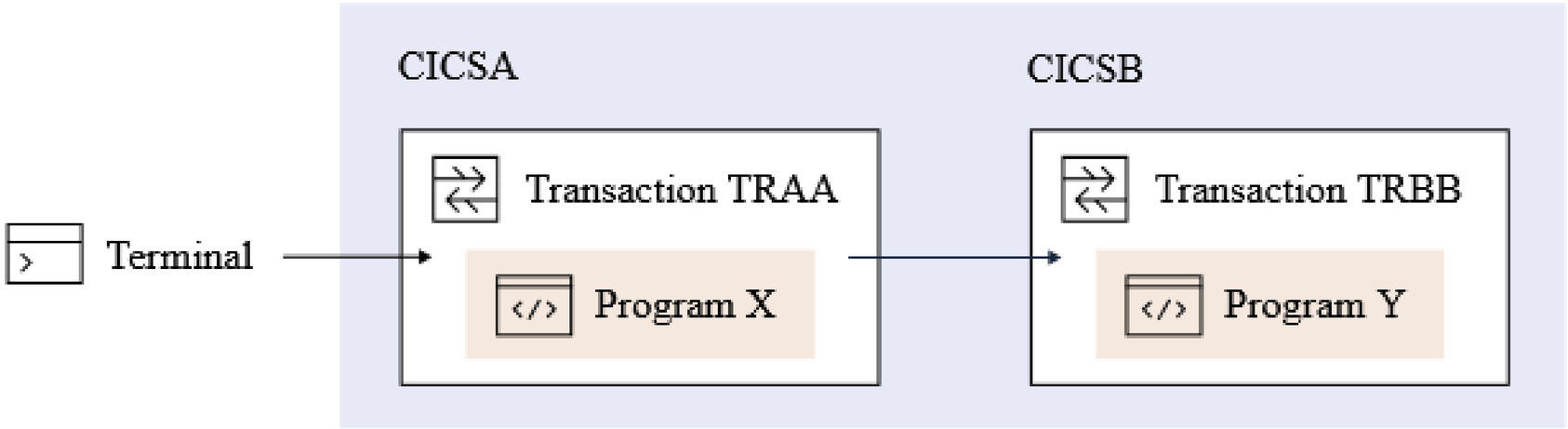


EC task(OTRX) executed **AFTER** queued OTQX completed.

CICS TS open beta

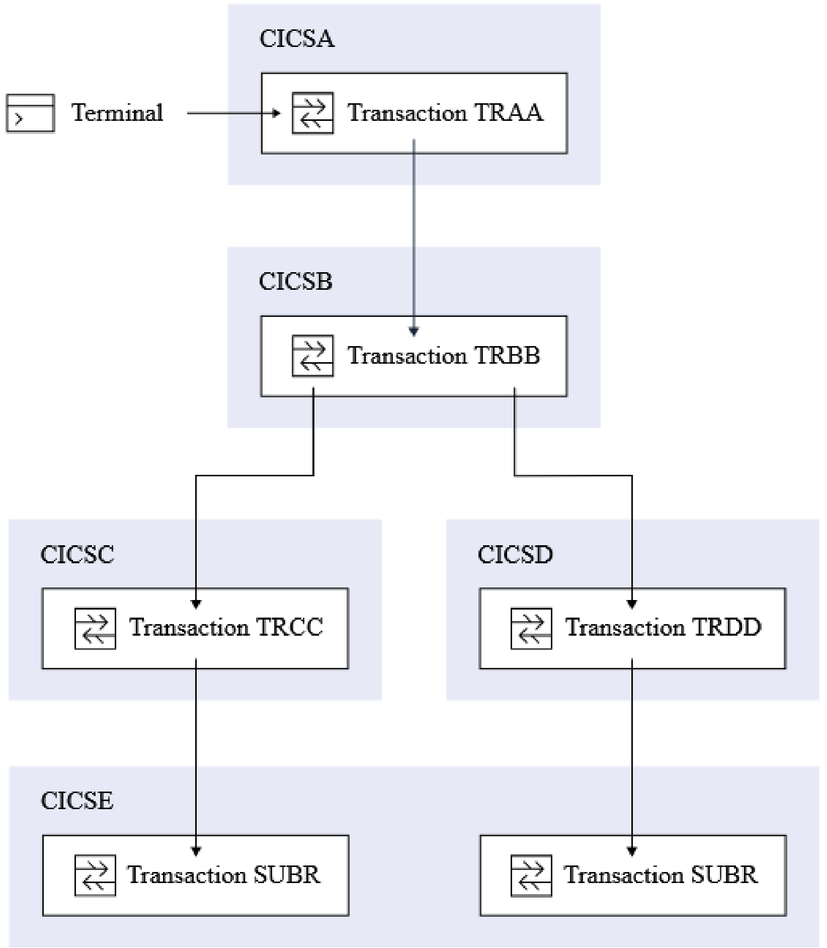
SEND/CONVERSE command with
DTP over MRO

What is Distributed Transaction Processing (DTP)?

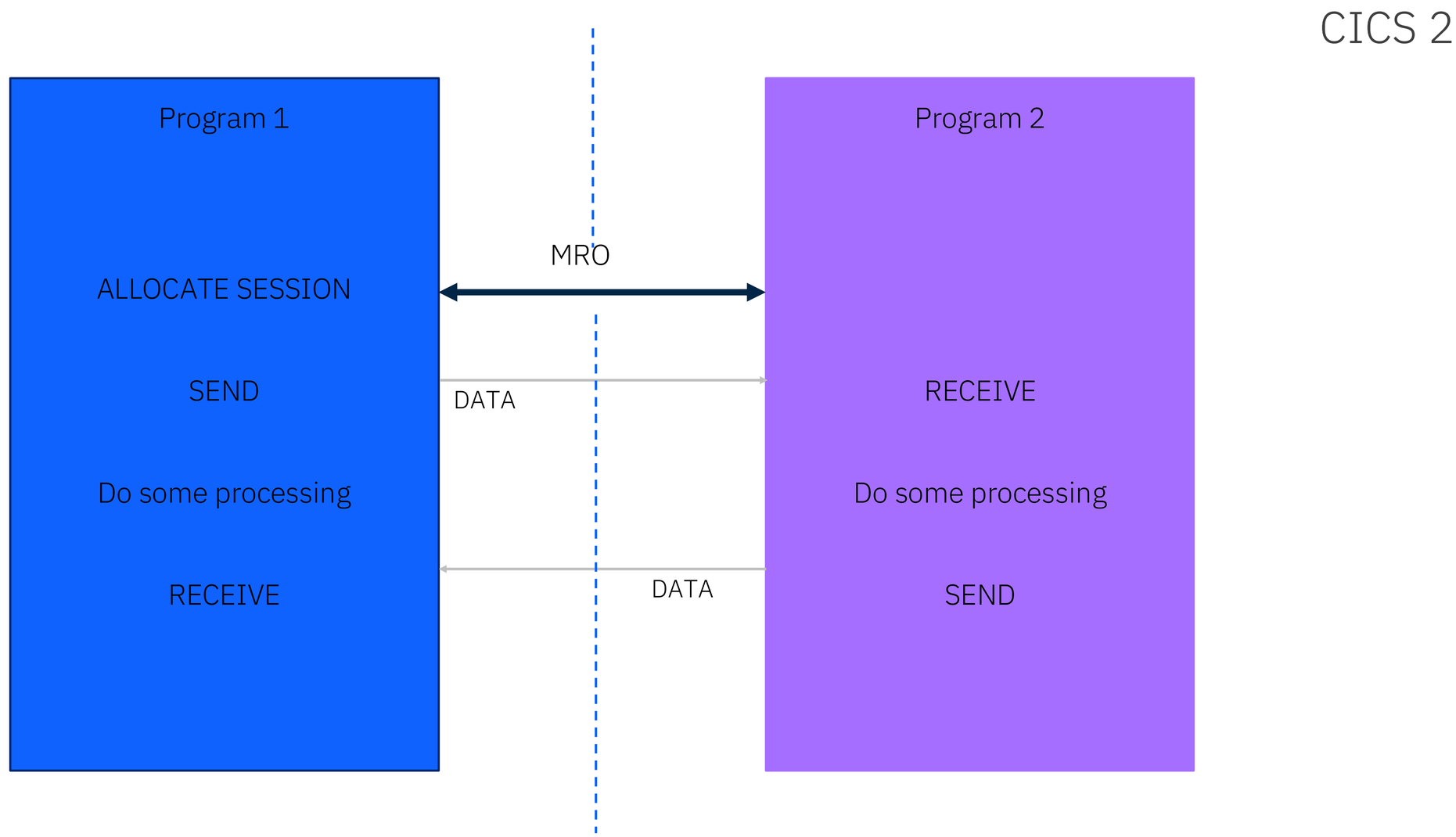


Distributed transaction processing

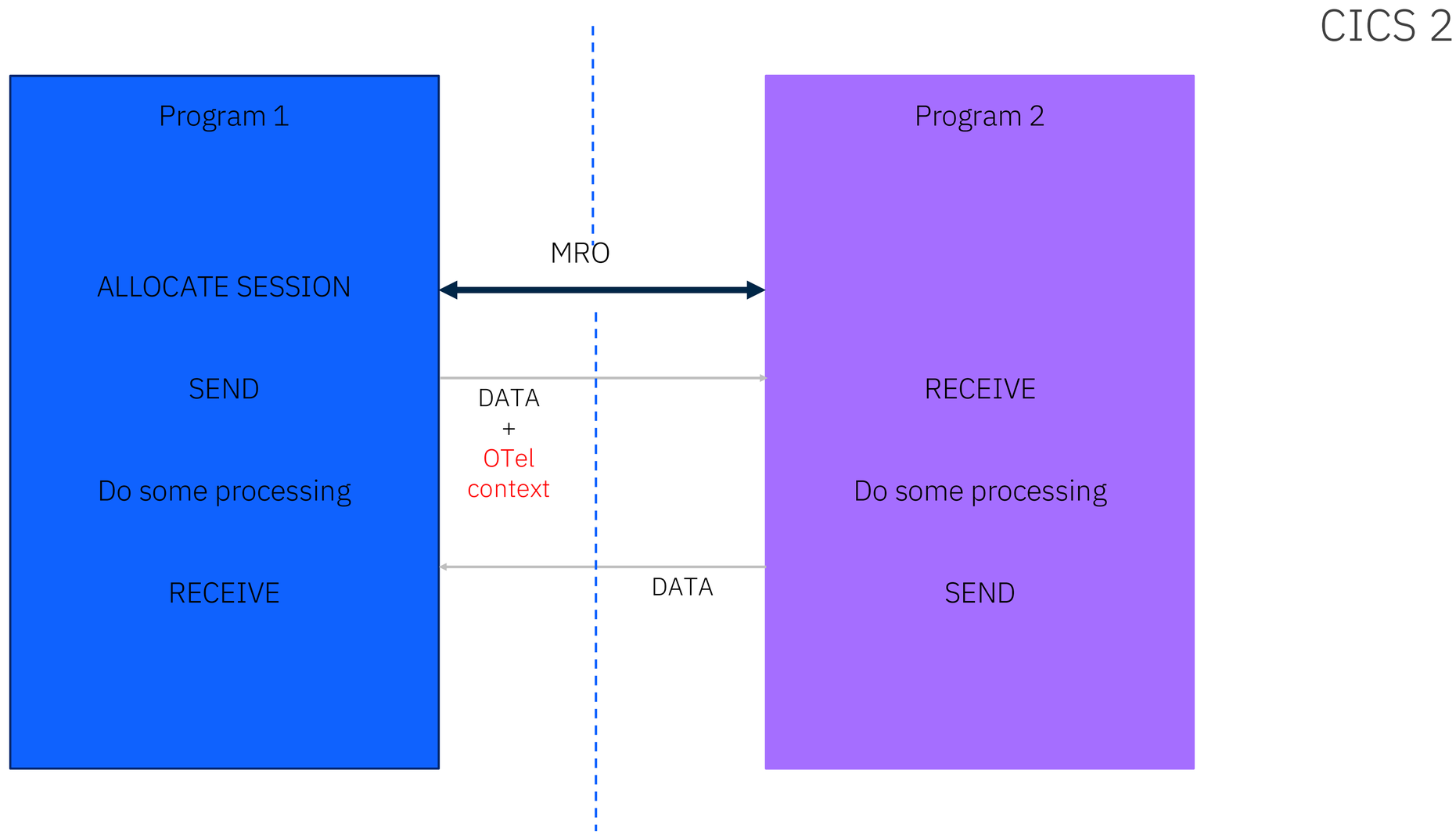
DTP scenarios can be complex



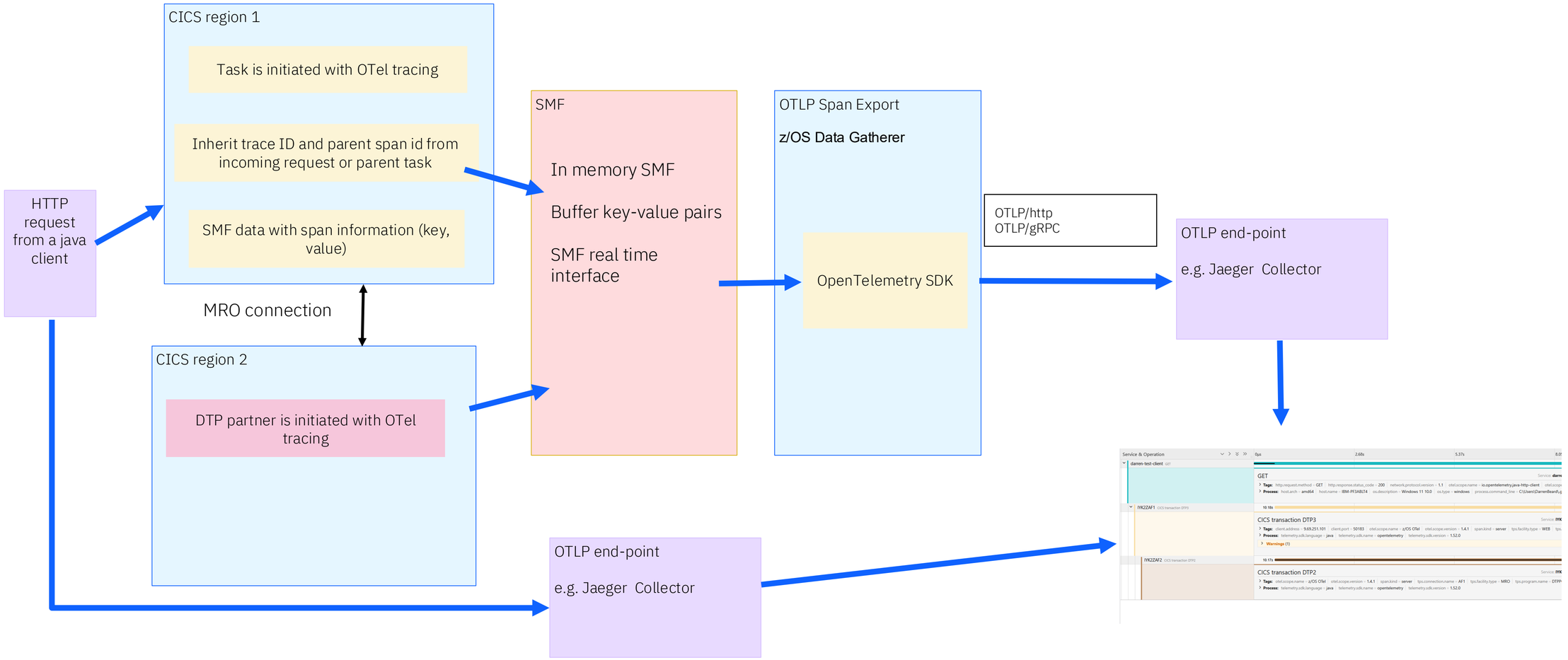
Program flows without OTel support



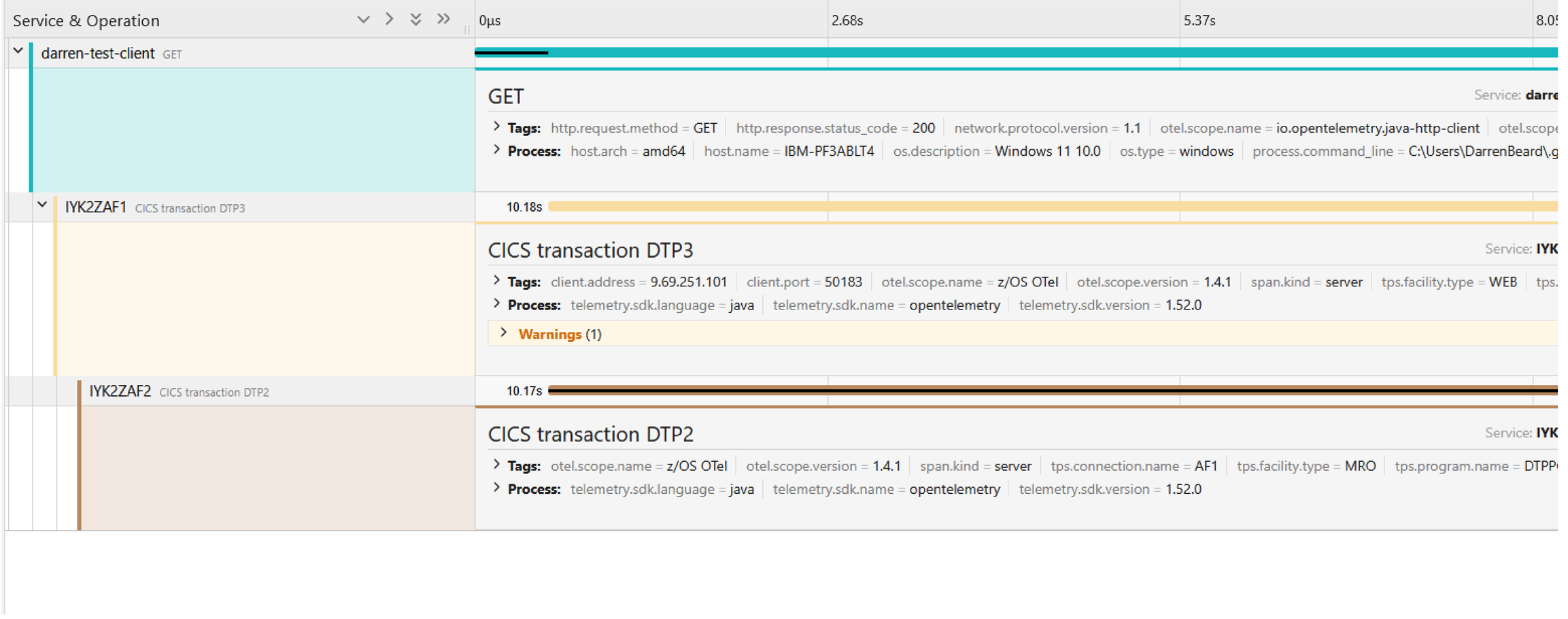
Program flows with OTel support



CICS OTel trace propagation and span data generation



OTel trace obtained



The OTel context is sent only once per DTP conversation

- A check is made that both CICS regions contain the DTP support before the OTel data are sent
- When the context is sent, it is only included on the first communication between the connected CICS regions
- Subsequent communications are unchanged

Your feedback is important!

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation