

# Enterprise Observability with Application Performance Monitoring Tools, OpenTelemetry and RTMC

Education Session

Josh Wisniewski

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

**IBM Z**

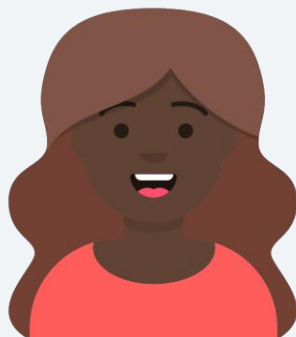


# The no disclaimer slide

- Notice that there is no disclaimer slide!
- **Everything demonstrated in this presentation is available today!**



Sarah  
site  
reliability  
engineer



Carol  
z/TPF  
coverage  
programmer



Zach  
application  
developer

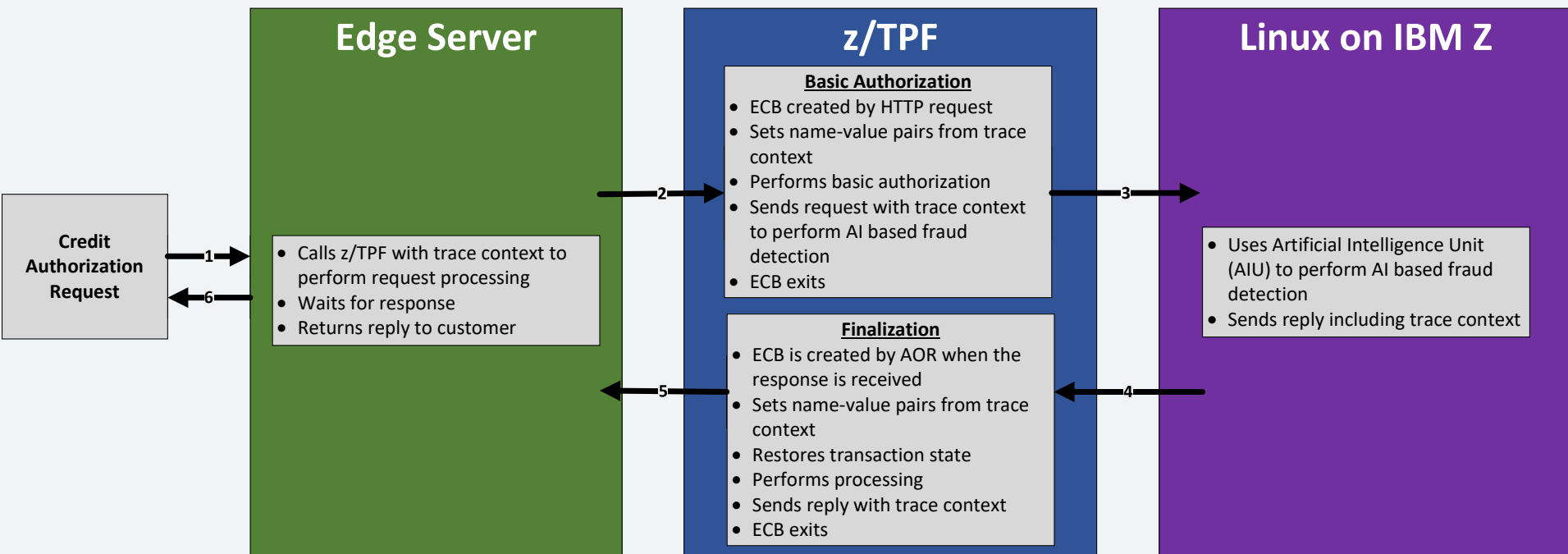
# **Credit authorization scenarios**

## **Enterprise architecture overview**

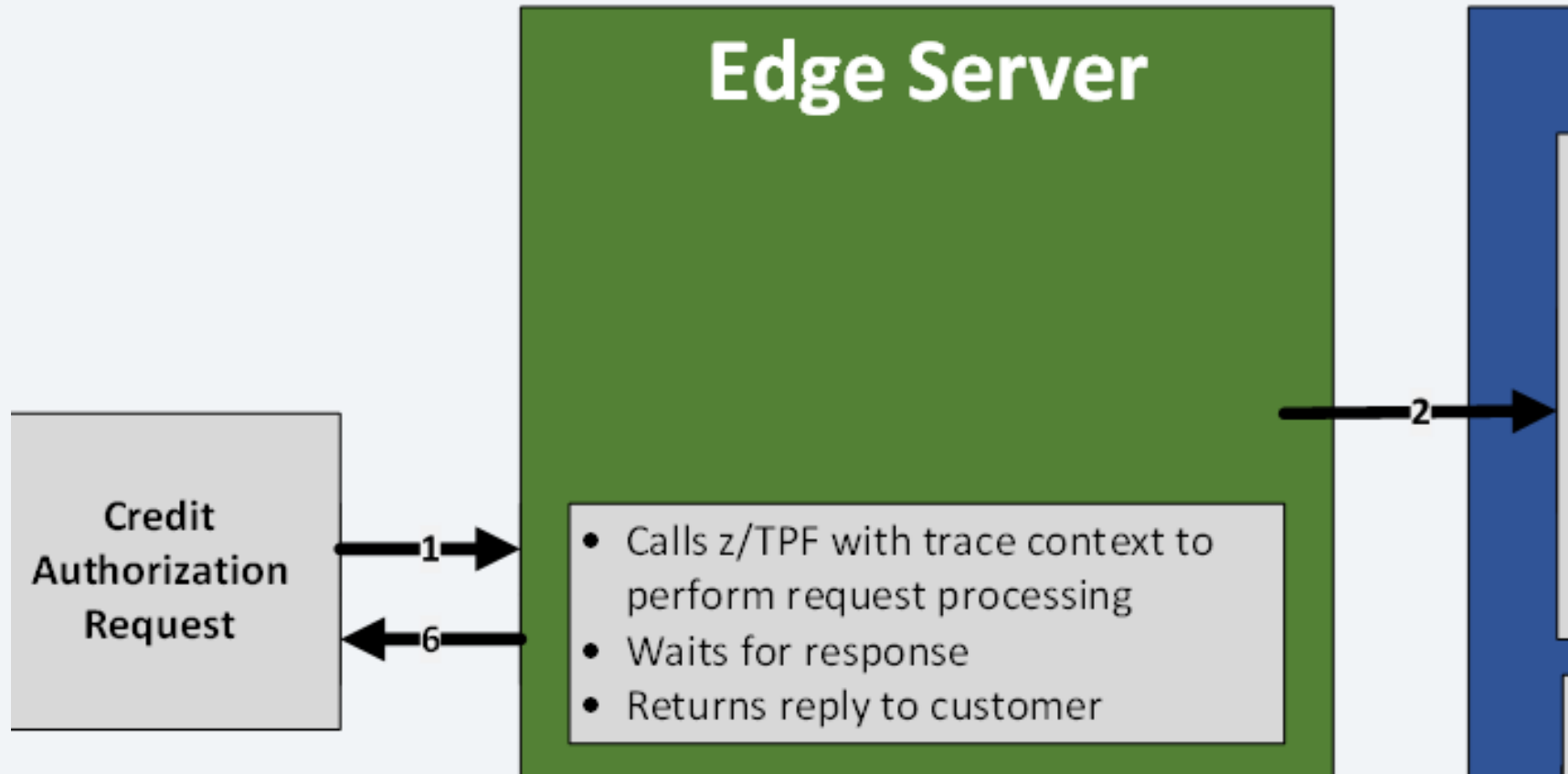
**Screen shot:** Enterprise architecture

**Story:** First let's walk through our sample enterprise architecture that is used when processing credit authorization requests.

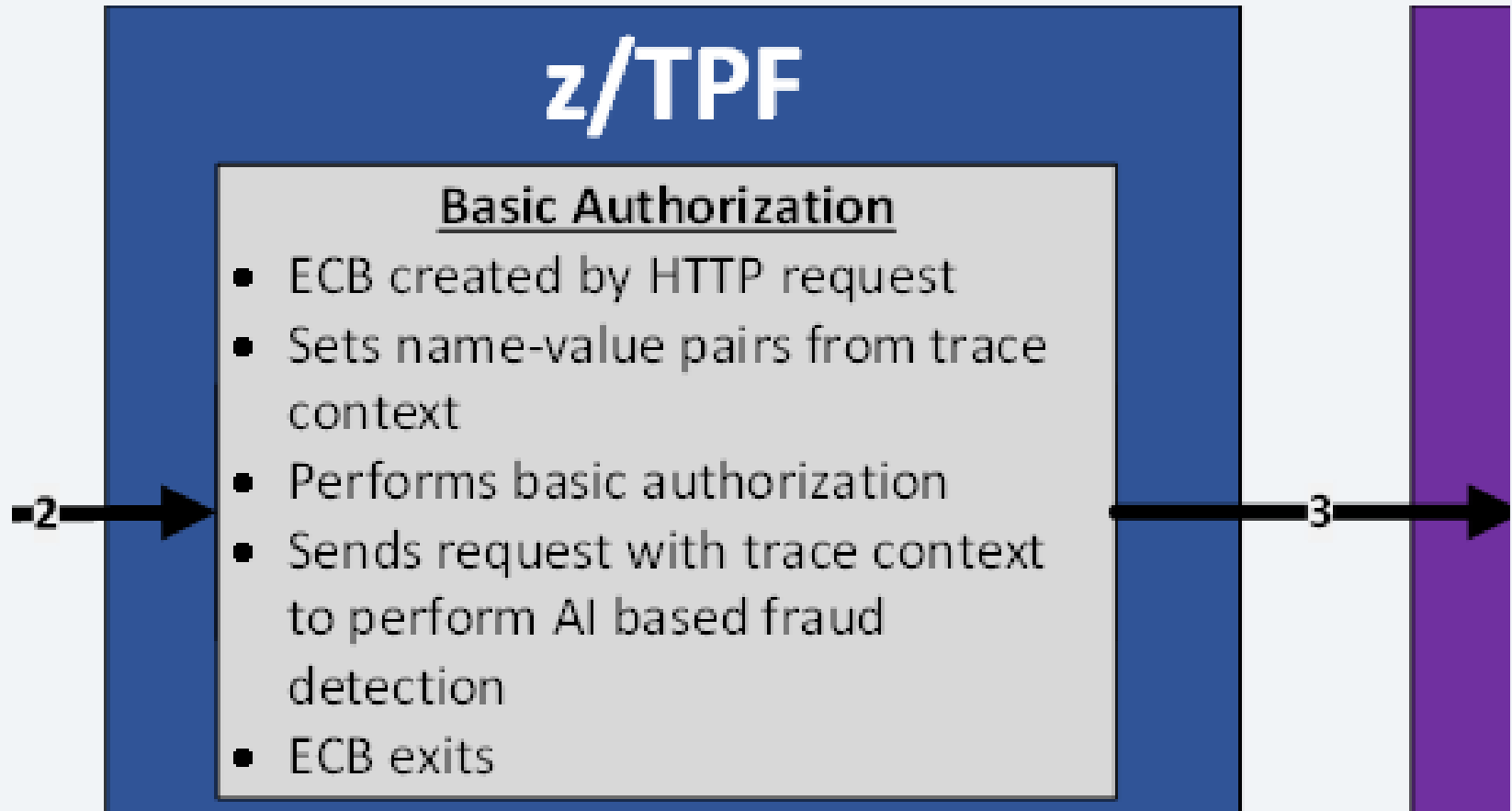
# Credit authorization enterprise architecture



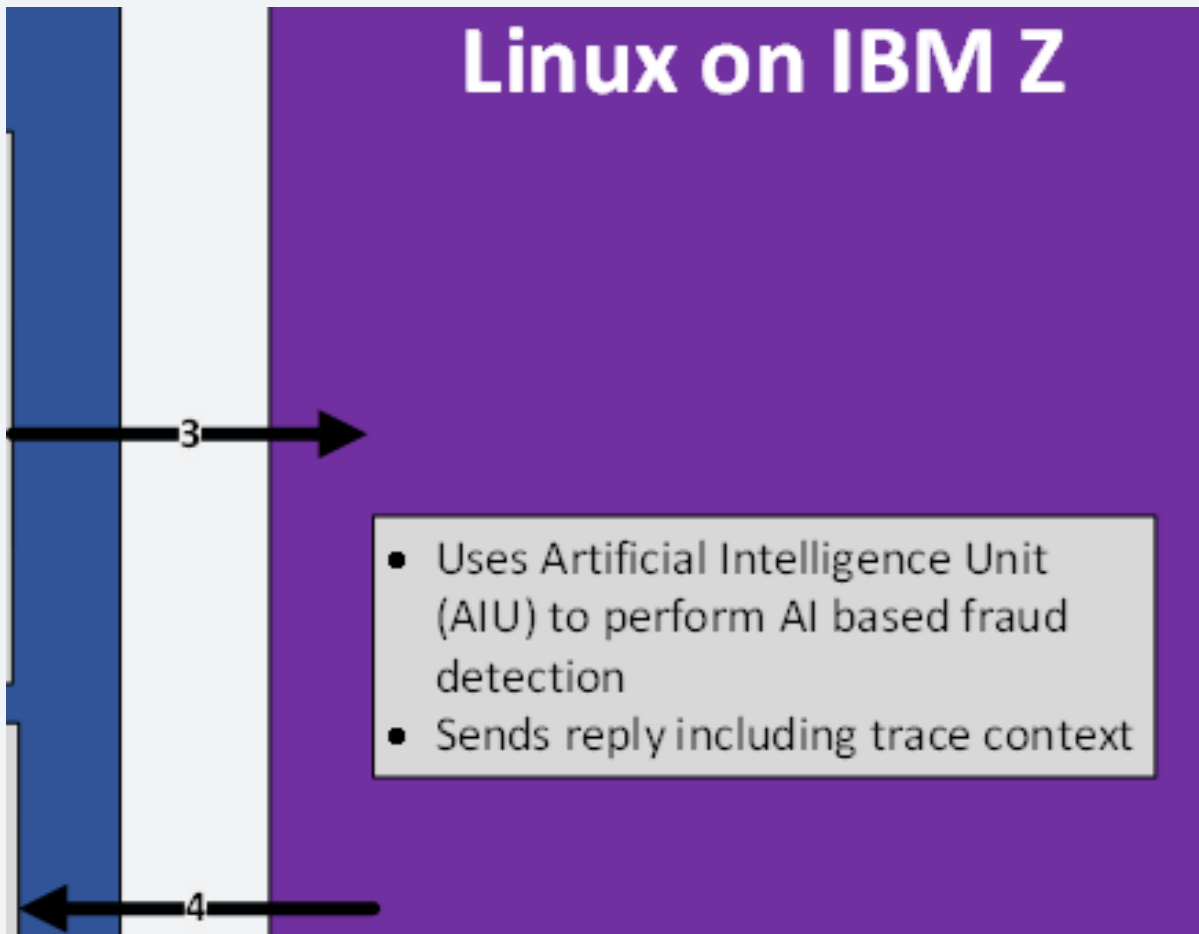
# Credit authorization enterprise architecture



# Credit authorization enterprise architecture

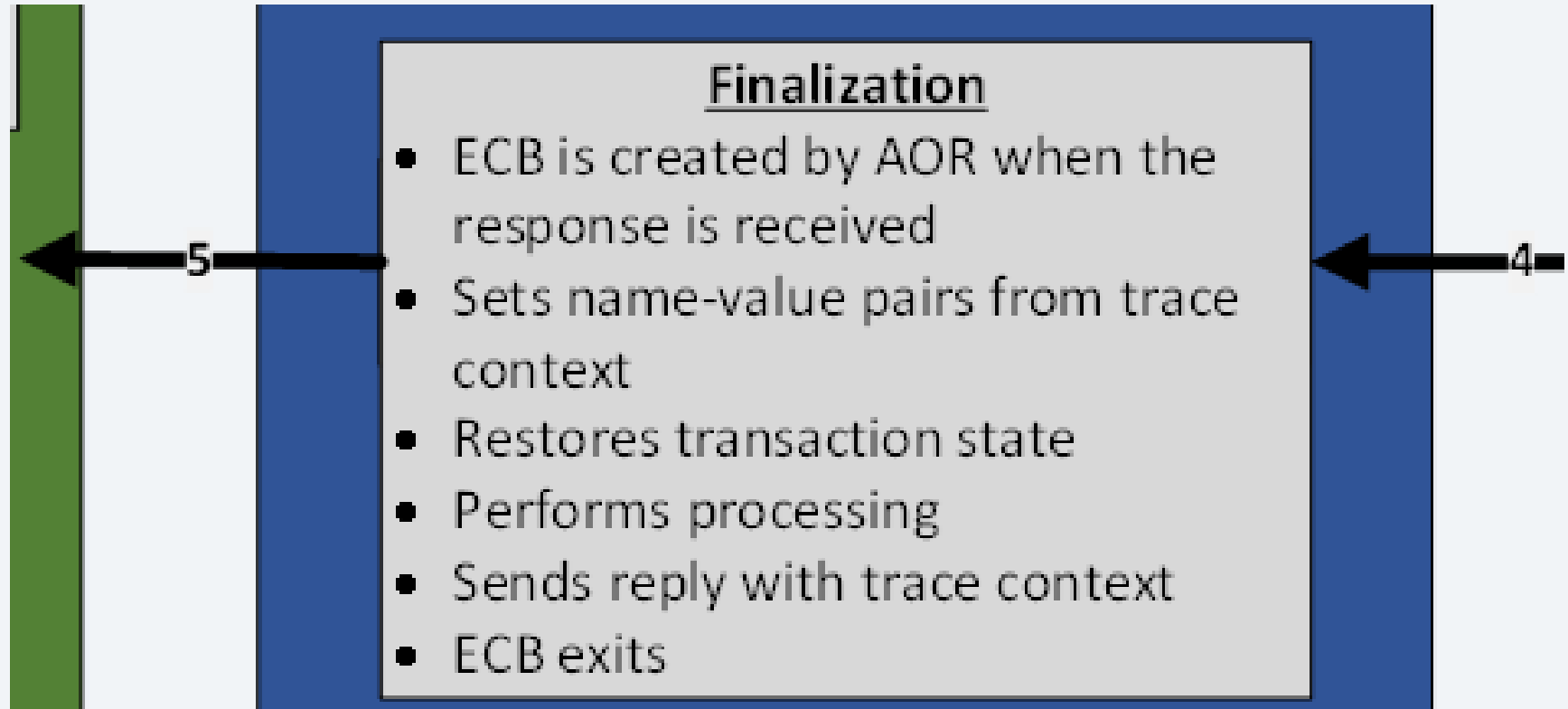


# Credit authorization enterprise architecture

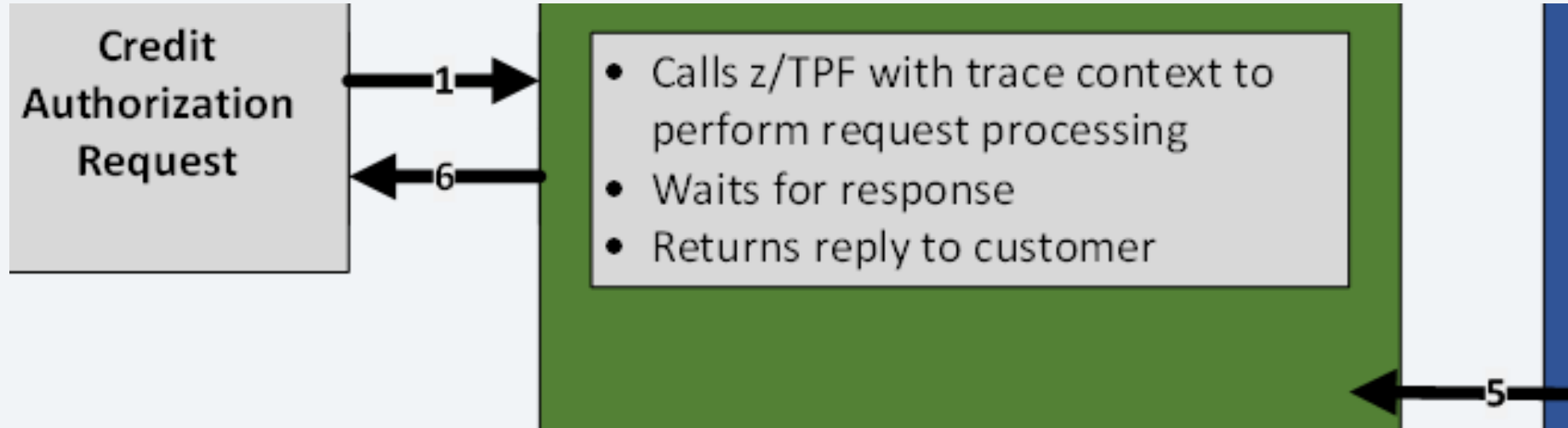




# Credit authorization enterprise architecture



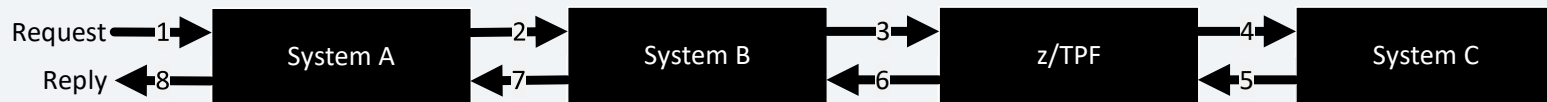
# Credit authorization enterprise architecture



# Site reliability engineer

# Site reliability engineer

- Transaction processing typically involves multiple systems across your enterprise.



- The site reliability engineer (SRE) monitors the processing of requests across the entire enterprise.
- They don't need to be experts in every system involved.
- Instead, each system can be treated as a black box that provides key metrics such as message rate, response time, and error rates.
- With z/TPF support for OpenTelemetry, z/TPF looks like any other system in your enterprise to the SRE.



Sarah  
site  
reliability  
engineer

# Site reliability engineer

- The SRE uses application performance monitoring (APM) tools, which provide **enterprise-wide observability**.
- With z/TPF support for OpenTelemetry, APM tools can receive metric and trace data from z/TPF through runtime metrics collection (RTMC).
- We're using IBM Instana in this demo, but z/TPF support for OpenTelemetry works with **any APM tool**.
- Your APM tool can identify deviations from normal and send you intelligent alerts, so you know **where to start** your investigation without involving every silo in your enterprise.



Sarah  
site  
reliability  
engineer

# **Application performance monitoring tools**

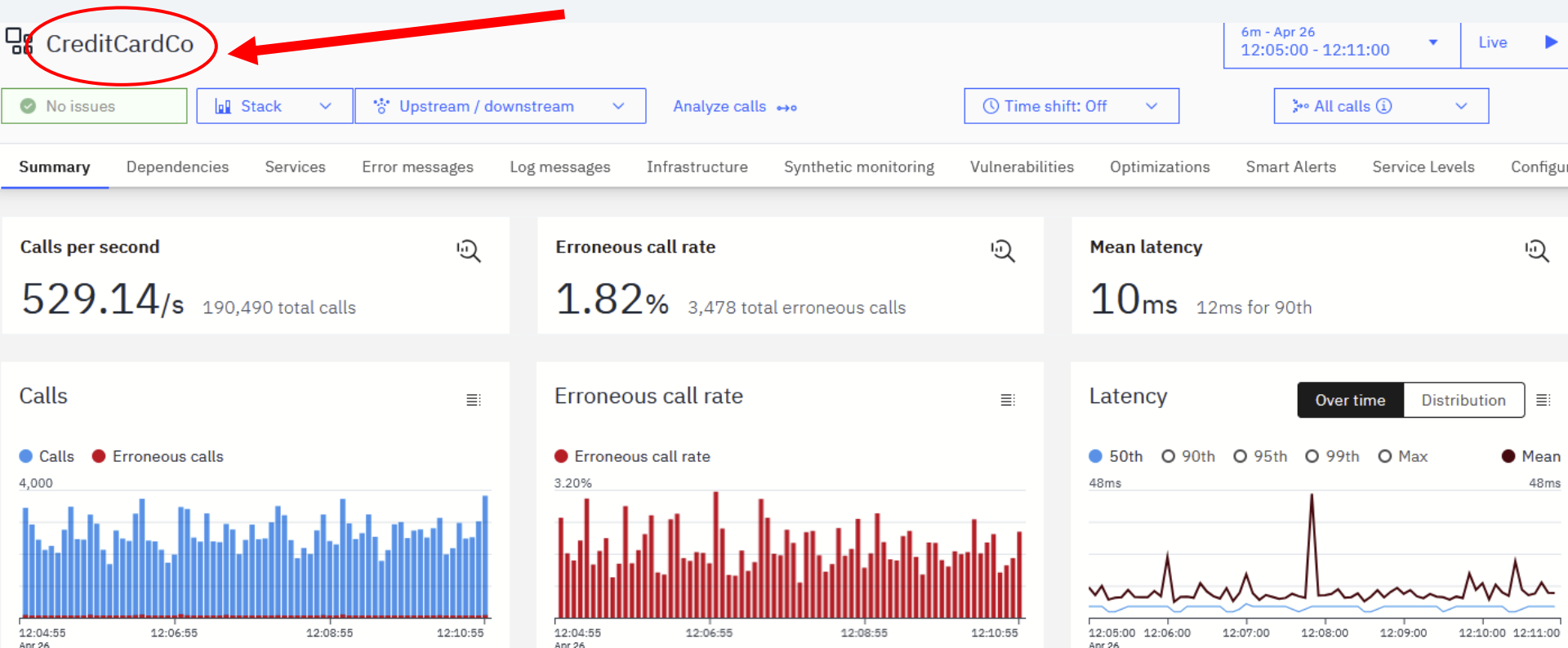
## **Credit authorizations – business as usual**

## **Screen shot:** Instana: Enterprise View – All Services

**Capabilities:** This dashboard shows us the workload across the entire enterprise. It incorporates incident reports over time. It shows the number of calls per second, the error rates and latency over time.

**Business value:** APM tools retain, display and can analyze historical data and call out incidents, trends and change points for your enterprise.

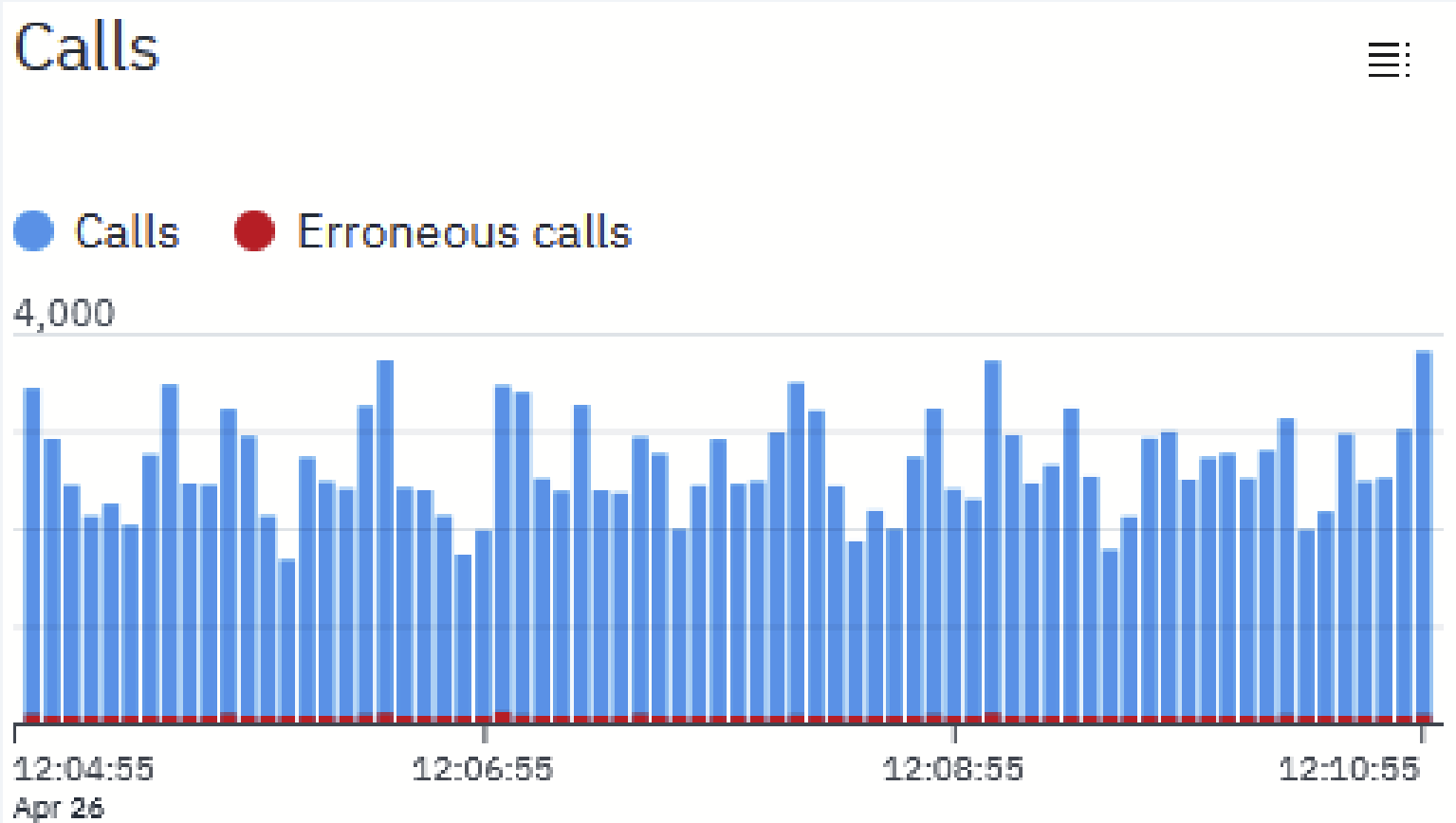
# Instana: Enterprise-wide view





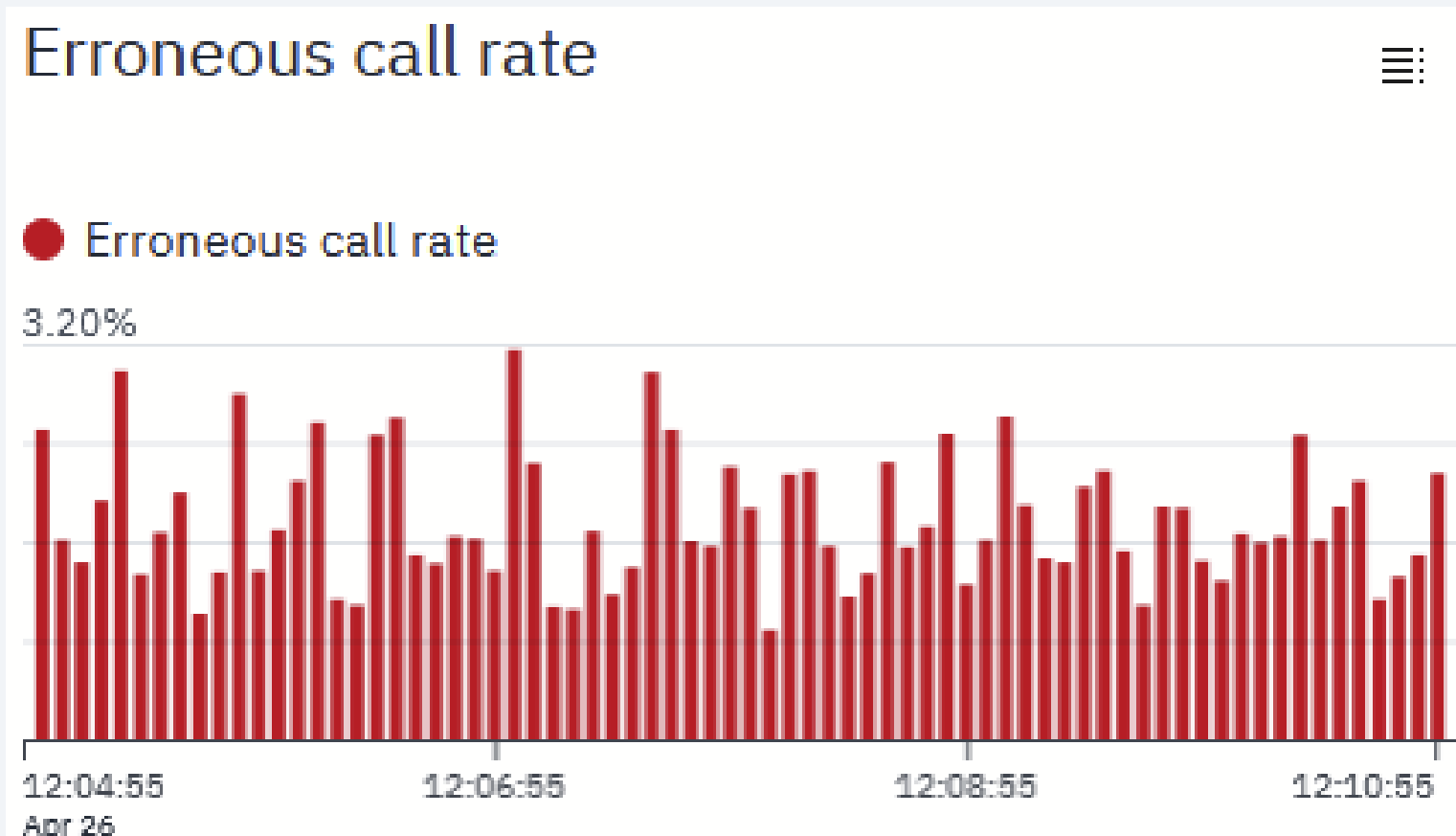
# Instana: Enterprise-wide view

## Calls graph



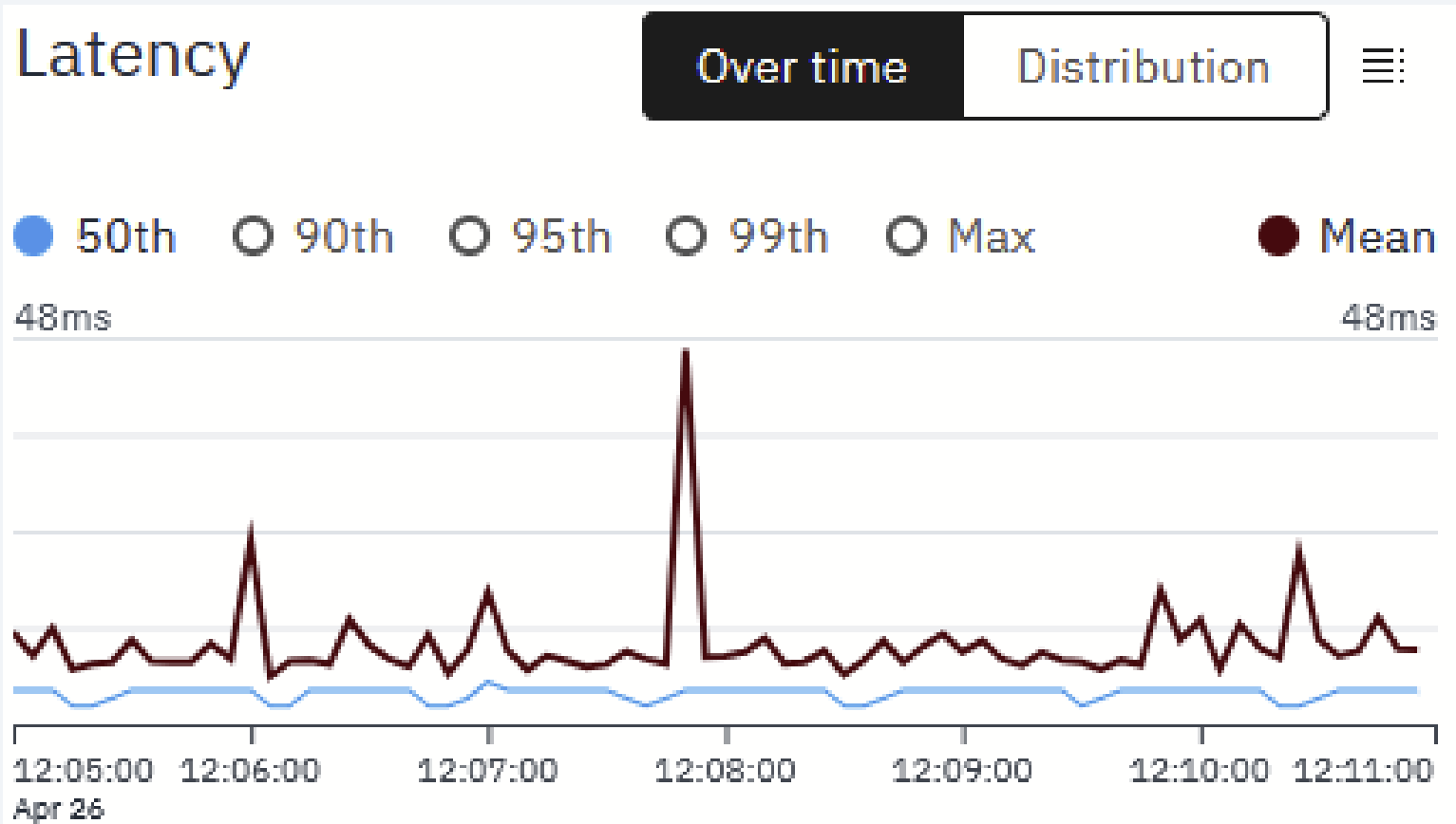
# Instana: Enterprise-wide view

## Erroneous call rate graph



# Instana: Enterprise-wide view

## Latency graph



**Screen shot:** Instana: System and user-defined metrics

**Capabilities:** Moving from the enterprise-wide views, we can drill down on a specific system, z/TPF in this case.

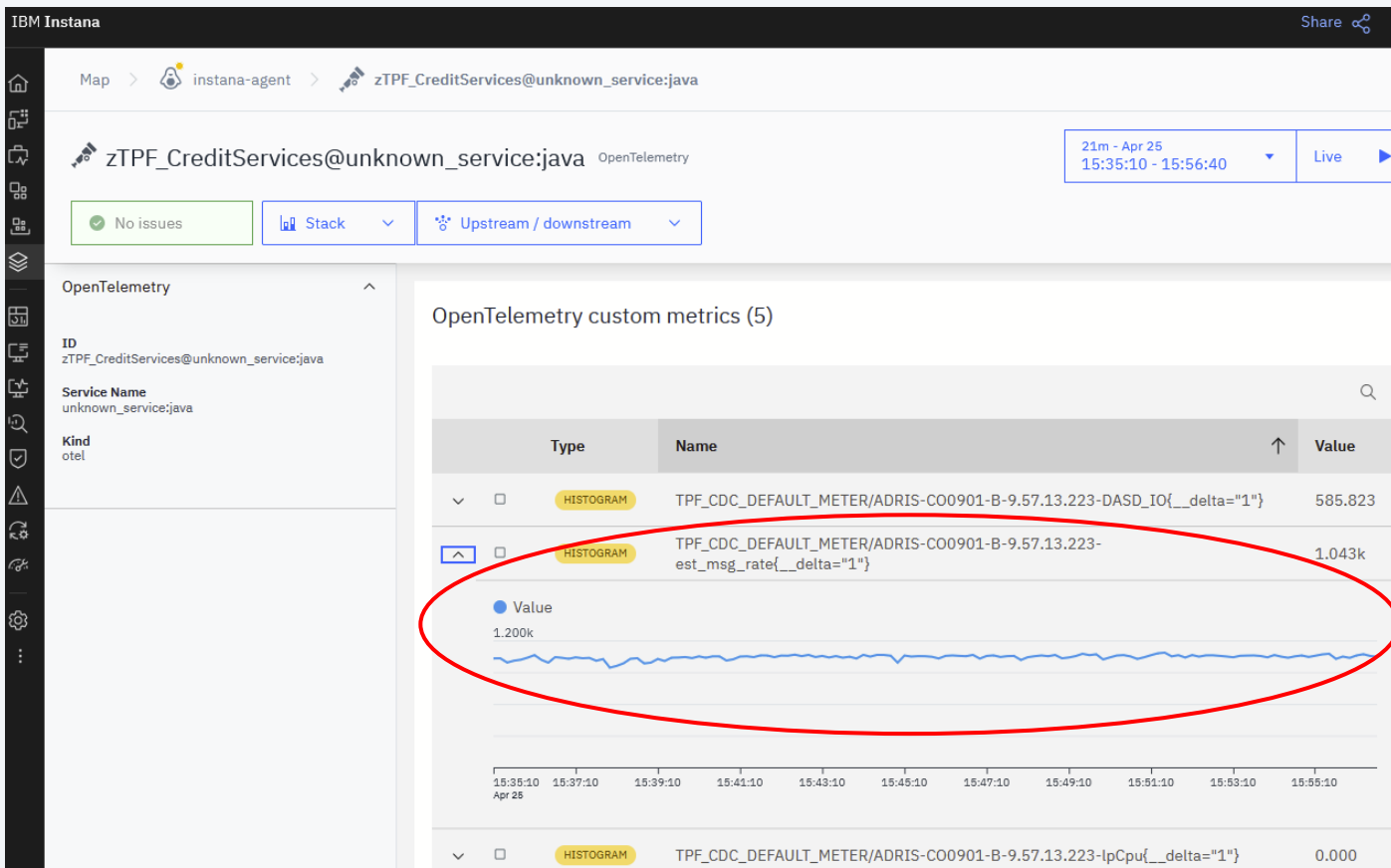
APM tools can show you key system level metrics to understand the health of your z/TPF system overall. These are our continuous data collection (CDC) metrics. The tpfrtmc OpenTelemetry forwarder by default sends processor utilization, low priority CPU utilization, workload/transactional CPU utilization, DASD I/O rates and estimated message rate which is based upon the name-value pair samples collected.

You can modify the tpfrtmc OpenTelemetry forwarder user exit to send your system, application and business user-defined metrics. You can integrate metrics specific to the health of your business into your APM tool monitoring and analysis.

**Business value:** APM tools can collect, display and analyze system level metrics including your system, application and business user-defined metrics.

# Instana: z/TPF system metrics

## Estimated message rate constant at 1000 messages per second



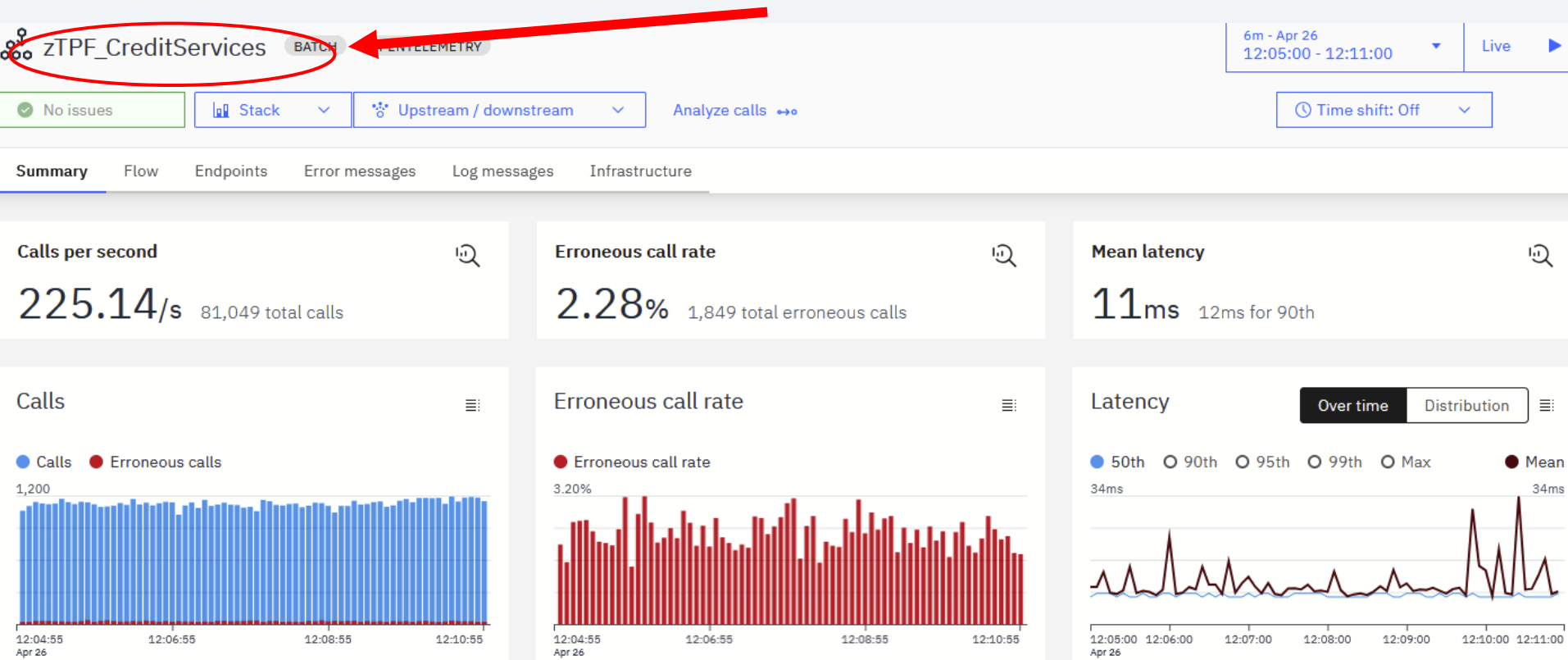
**Screen shot:** Instana: Service Dashboard – z/TPF

**Capabilities:** This dashboard shows us the workload of the TPF system from a name-value pair collection point of view. It shows history of our name-value pair sample rate, error rates for those samples and the response time (latency) of those samples.

Note that **none** of the Instana features shown have been customized for z/TPF. All of the features I'm showing are available out of the box for all systems.

**Business value:** APM tools retain, display and can analyze historical data and call out trends and change points.

# Instana: z/TPF system view



## **Screen shot:** Instana: Analyze Calls – Graphs

**Capabilities:** The analyze calls dashboard allows us to dig into the trace data. The trace data is sourced from our name-value pair collection results. Your name-value pairs can be included as attributes in the trace data.

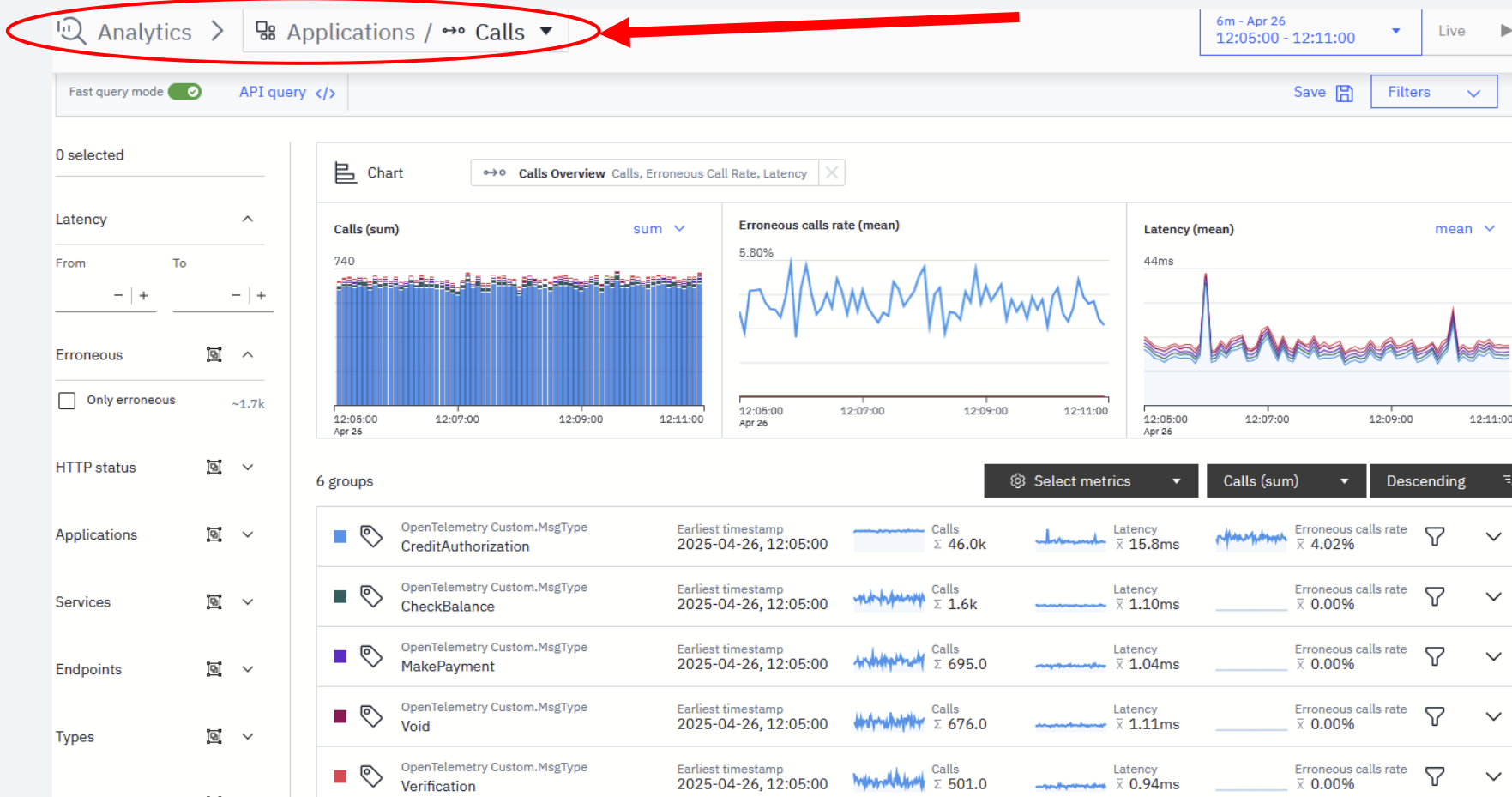
It starts at the top with a very familiar summary layout. We can see our workload broken down by message type in terms of sample rate, error rates, and response (latency time) in the summary at the top of this dashboard.

As you dig into the calls, you can filter or group by name-value pair values. The SRE does not need to know the details of your name-value pairs. They can just think of them as transaction annotations or descriptors. In this example, we're grouping by the MsgType name-value pair.

**Business value:** APM tools can show you sample rates, error rates, and response time by the type of messages processed by your system.



# Instana: z/TPF analyze calls



# Instana: z/TPF analyze calls

## Grouped by MsgType



Filter



Dest

Service

Name

=

zTPF\_CreditServices



Add filter



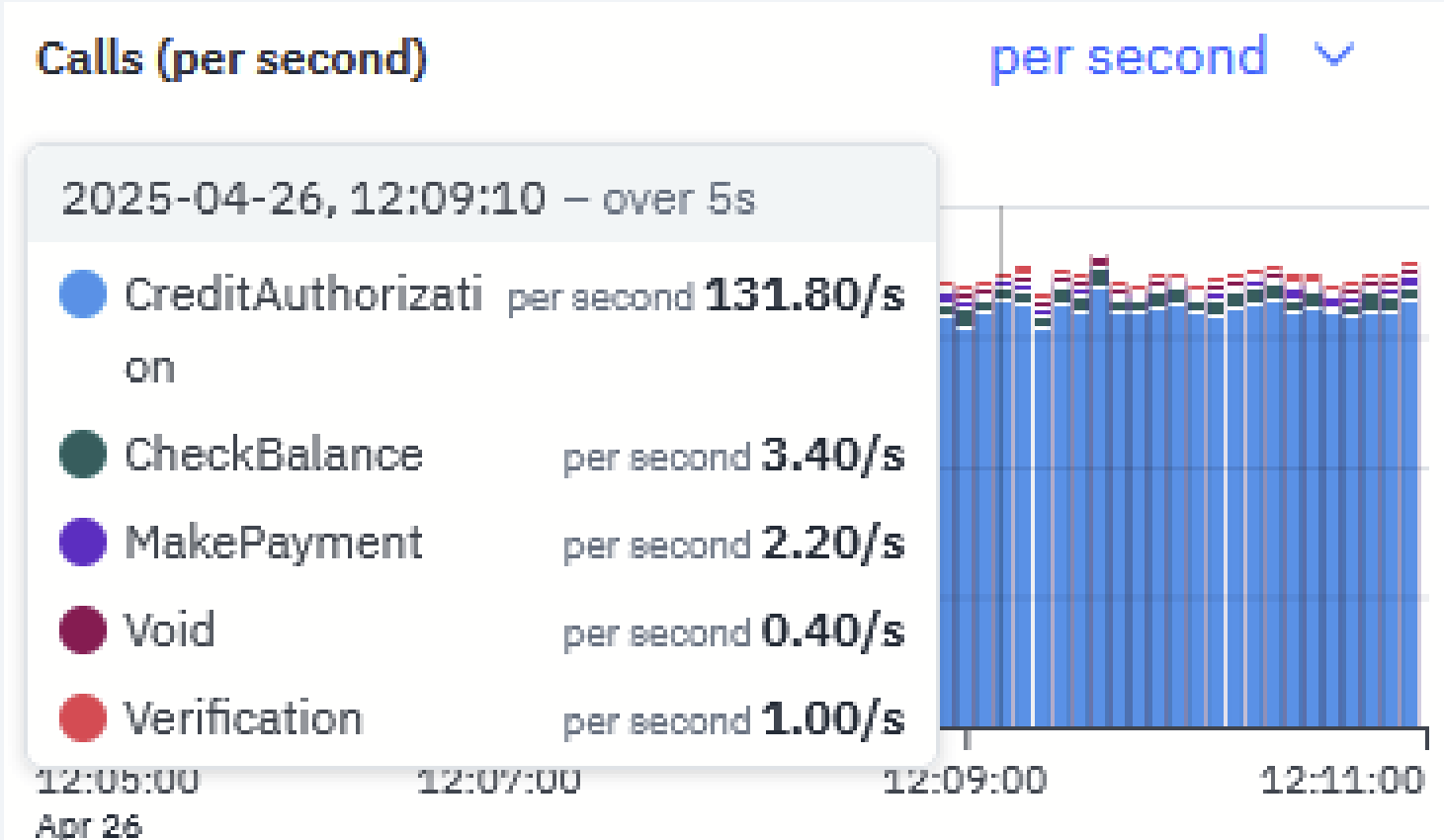
Group

OpenTelemetry Custom ▶ MsgType



# Instana: z/TPF analyze calls

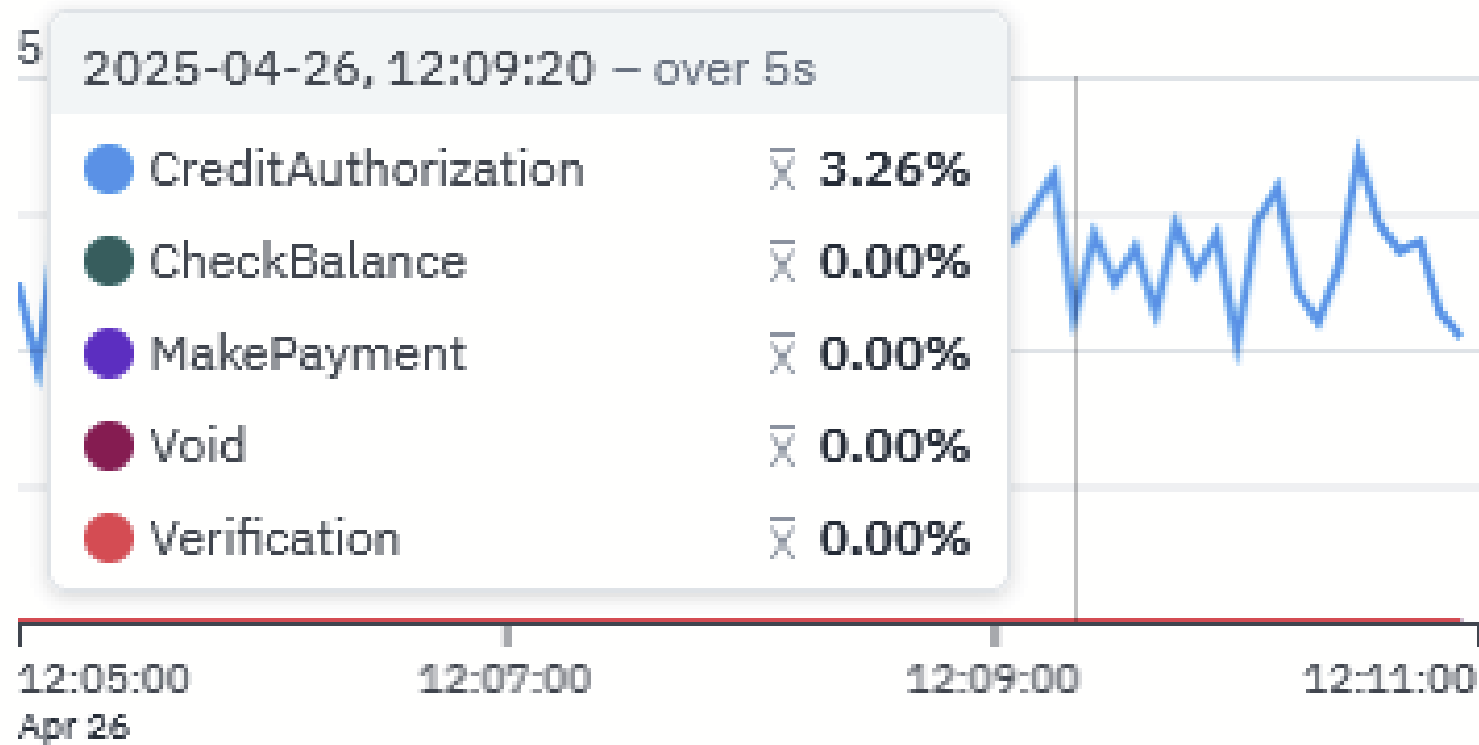
## Call graph grouped by MsgType



# Instana: z/TPF analyze calls

## Erroneous calls rate graph grouped by MsgType

Erroneous calls rate (mean)

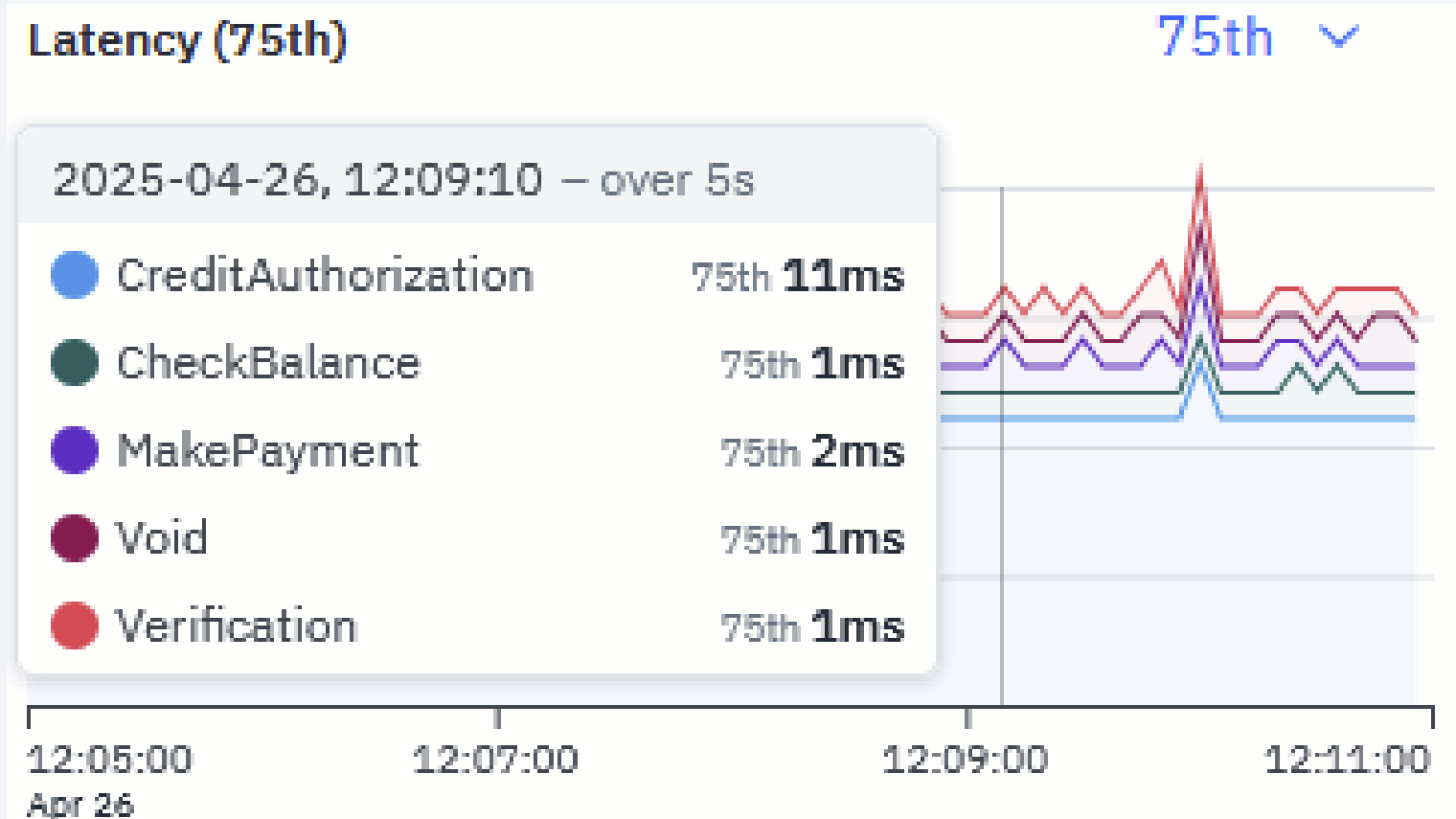


**Screen shot:** Instana: Analyze Calls – Graphs – Latency

**Capabilities:** This graph shows the latency for the 75th percentile for all traffic in z/TPF. This means that 75% or the majority of all traffic is completing in 11 milliseconds or less.

# Instana: z/TPF analyze calls

## Latency graph grouped by MsgType



**Screen shot:** Instana: Analyze Calls – List of sample credit authorizations






**Capabilities:** Your APM tool provides lists of different types of sampled messages with various details like without errors (green check) and with errors (red exclamation) along with their response time (latency). This is another way that we can compare different groupings of messages.

In this view, we can select individual messages to see the details.

**Business value:** APM tools provide lists of the sampled messages so you can compare and investigate messages of interest.

# Instana: z/TPF analyze calls list

## Filtering on credit authorization messages

<div><div></div><div>Endpoint.Name MsgType-CreditAuthorizatio</div><div>Earliest timestamp 2025-04-25, 15:44:00</div><div></div><div>Calls Σ 92.5k</div><div></div><div>Latency X 12.2ms</div><div></div><div></div></div>						
Status	Call	Service	Timestamp	↓	Latency	↑↓
✓	↔ MsgType-CreditAuth...	zTPF_CreditServices	2025-04-25, 15:55:59		9ms	
✓	↔ MsgType-CreditAuth...	zTPF_CreditServices	2025-04-25, 15:55:59		6ms	
!	↔ MsgType-CreditAuth...	zTPF_CreditServices	2025-04-25, 15:55:59		2ms	
!	↔ MsgType-CreditAuth...	zTPF_CreditServices	2025-04-25, 15:55:59		4ms	
✓	↔ MsgType-CreditAuth...	zTPF_CreditServices	2025-04-25, 15:55:59		10ms	



**Screen shot:** Instana: Call Diagram – Successful message

**Capabilities:** With your APM tool, you can inspect a sample message that is going through all of enterprise processing to understand the path.

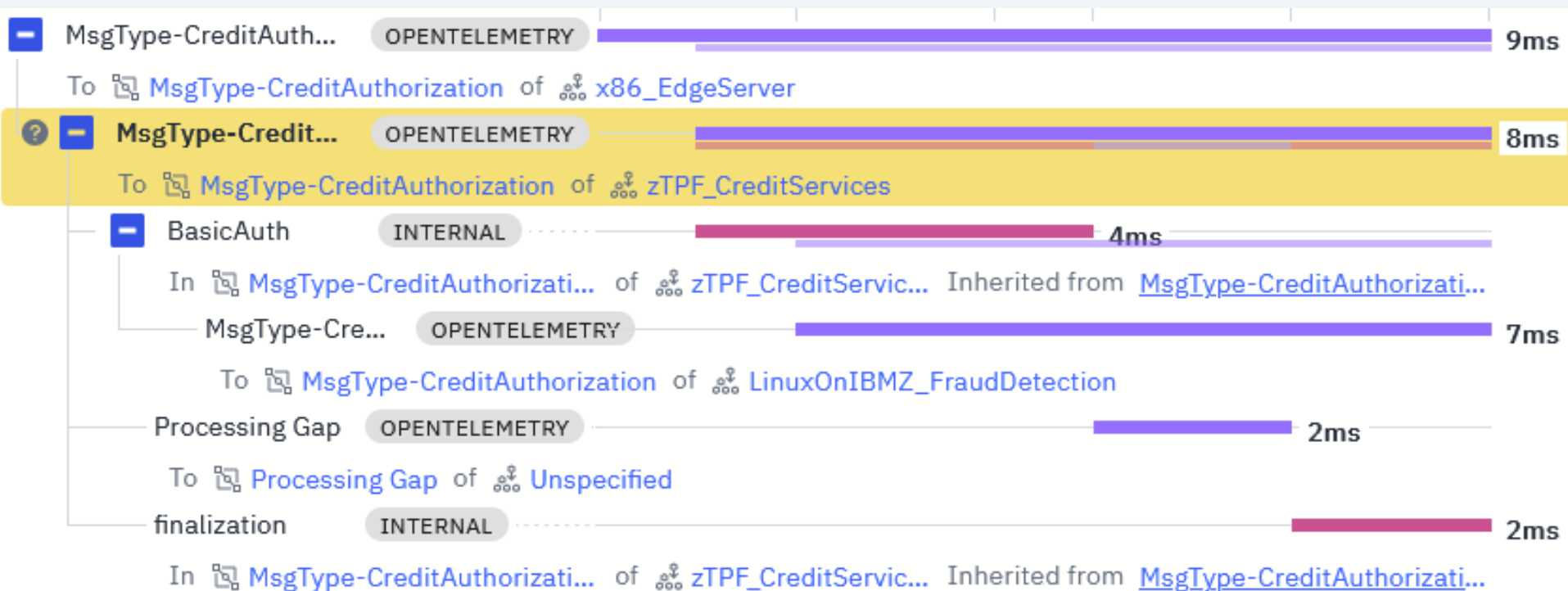
The calls diagram pictorially shows the relationship between the calls between the systems and includes the two processes on z/TPF. On the top left, you can see the edge server starting the transaction and this process exists for the life of the transaction. Moving left to right in time, we can see a call was made to z/TPF to the basic authorization processing. Continuing left to right in time, there's a call from z/TPF to a Linux on IBM Z system to perform AI fraud detection. The basic authorization process exited. Continuing left to right in time, there's a call from AI fraud detection on Linux on IBM Z to the finalization processing on z/TPF by way of an AOR. The finalization processing replies to the edge server.

We can see our message being processed by 3 different platforms in our enterprise.

**Business value:** With your APM tools, you can inspect the processing of a message throughout your entire enterprise.

# Instana: z/TPF analyze call details

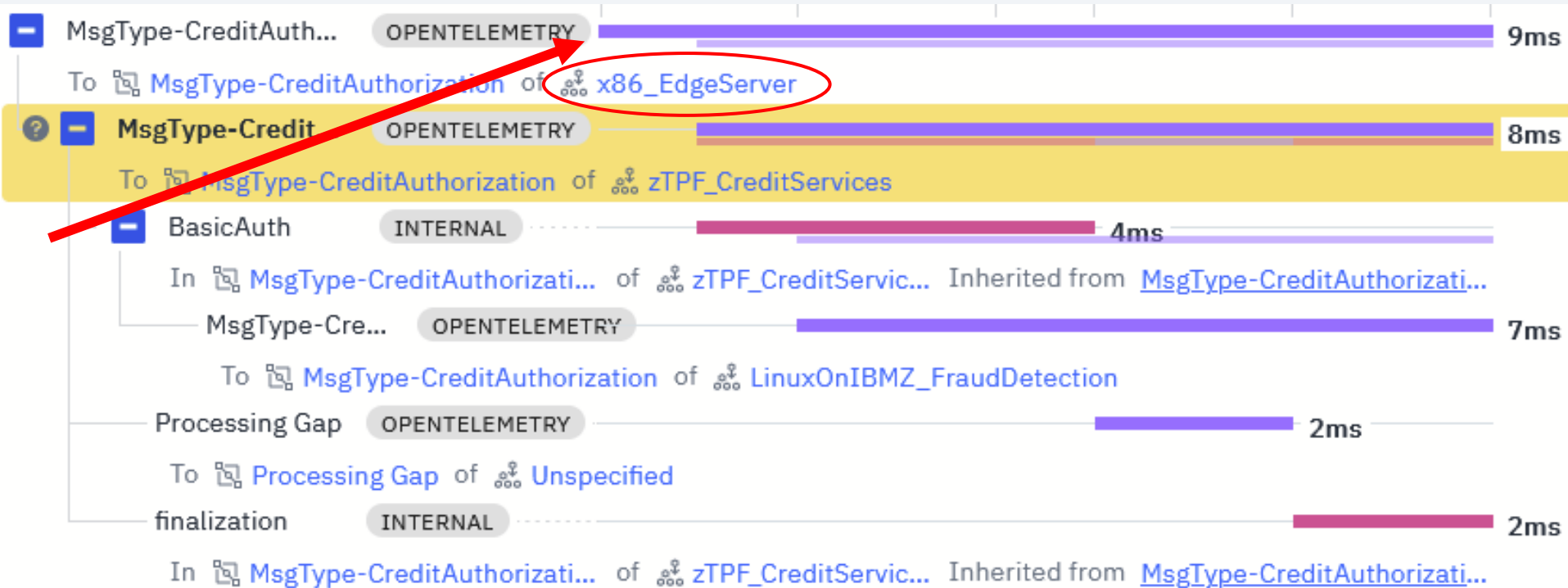
## Credit authorization message ending in success



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

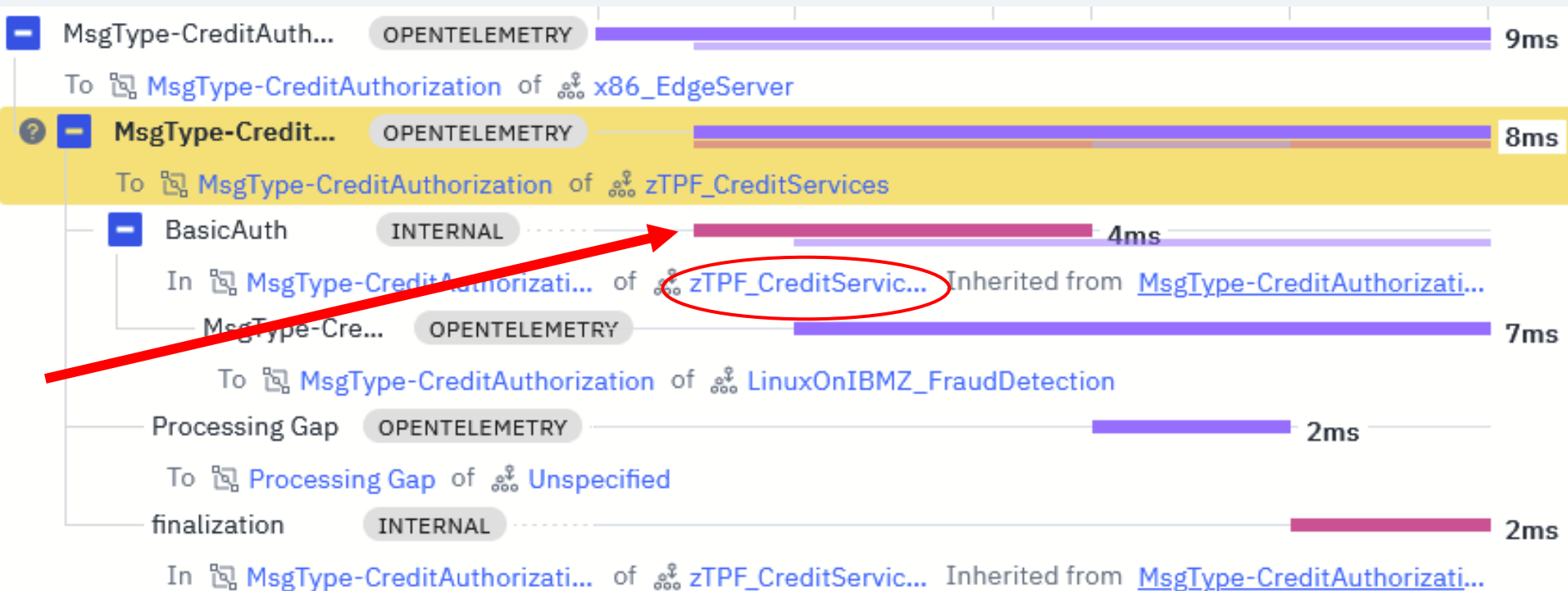
### Transaction trace starts on edge server



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

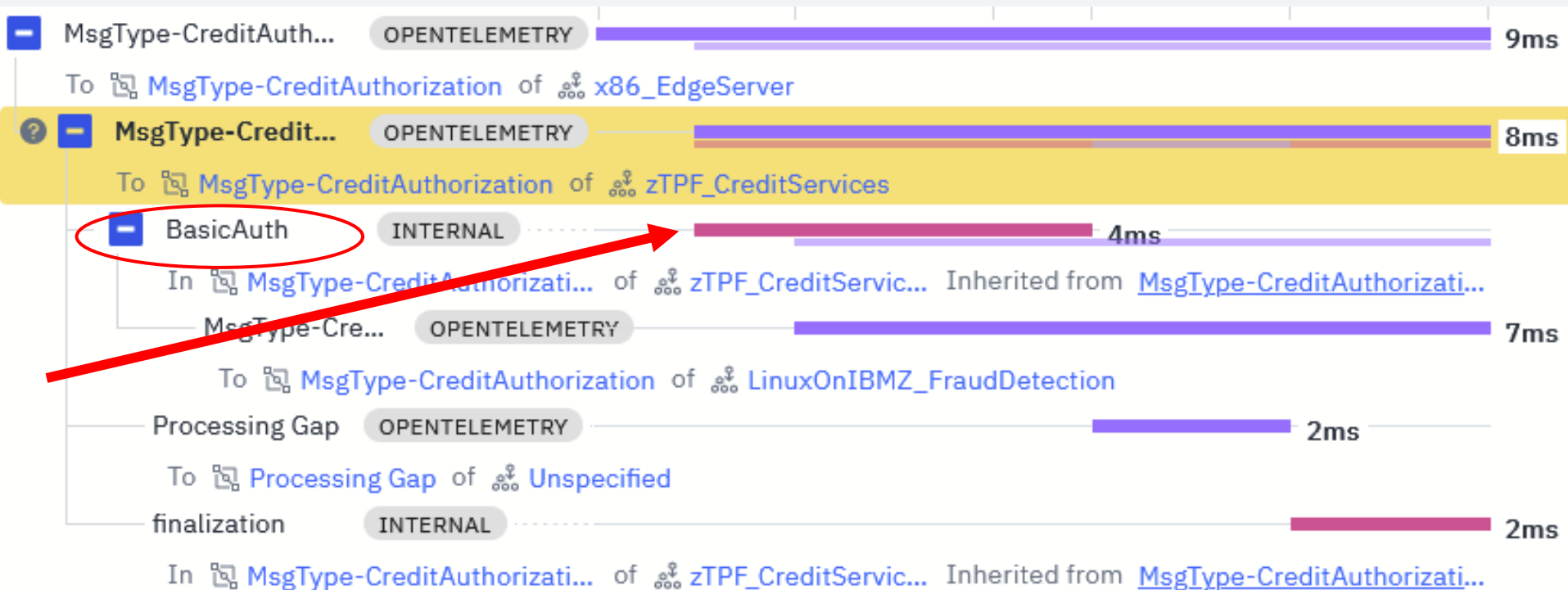
### Edge server calls z/TPF



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

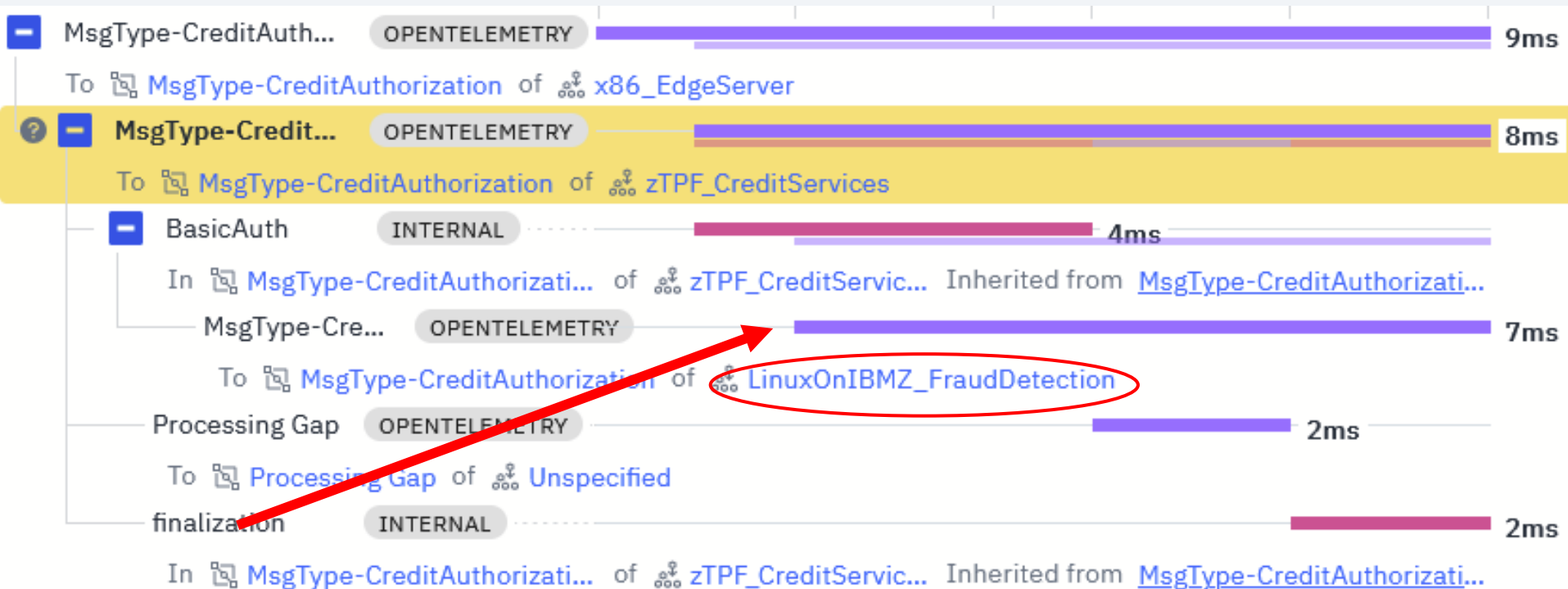
### z/TPF processing starts in basic authorization processing



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

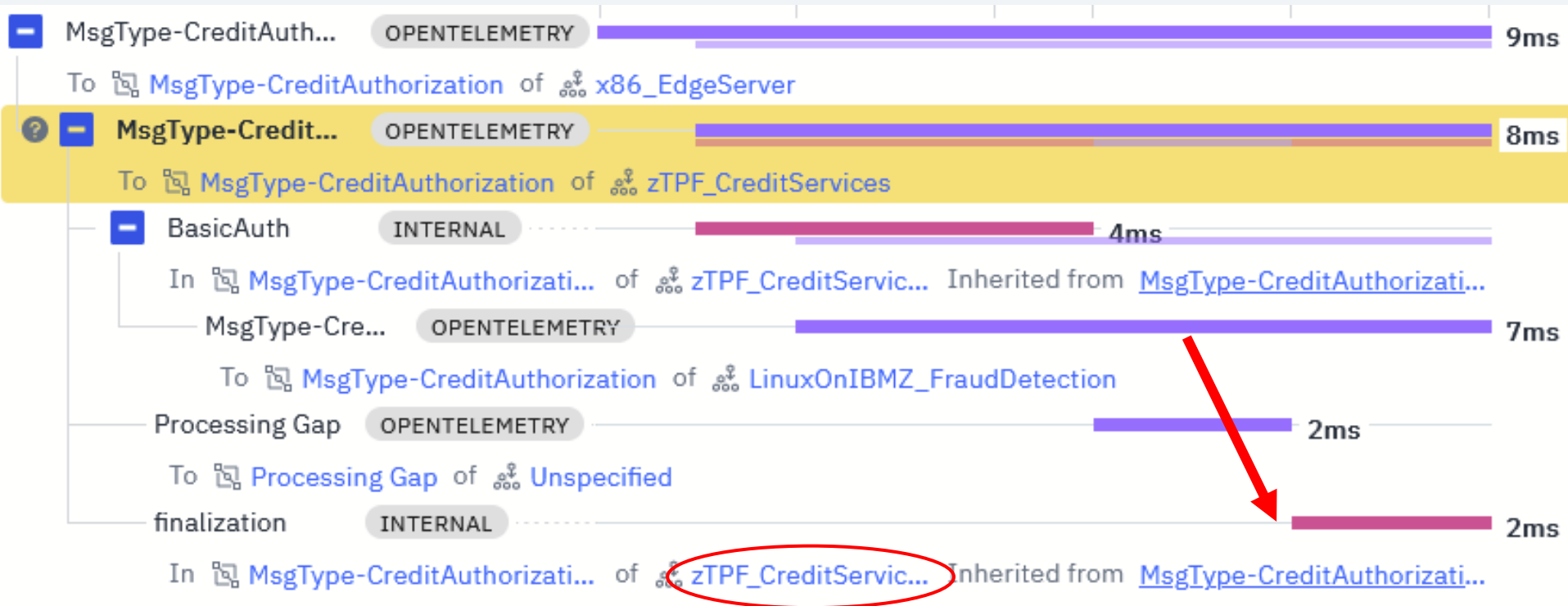
### z/TPF calls AI fraud detection on Linux on IBM Z



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

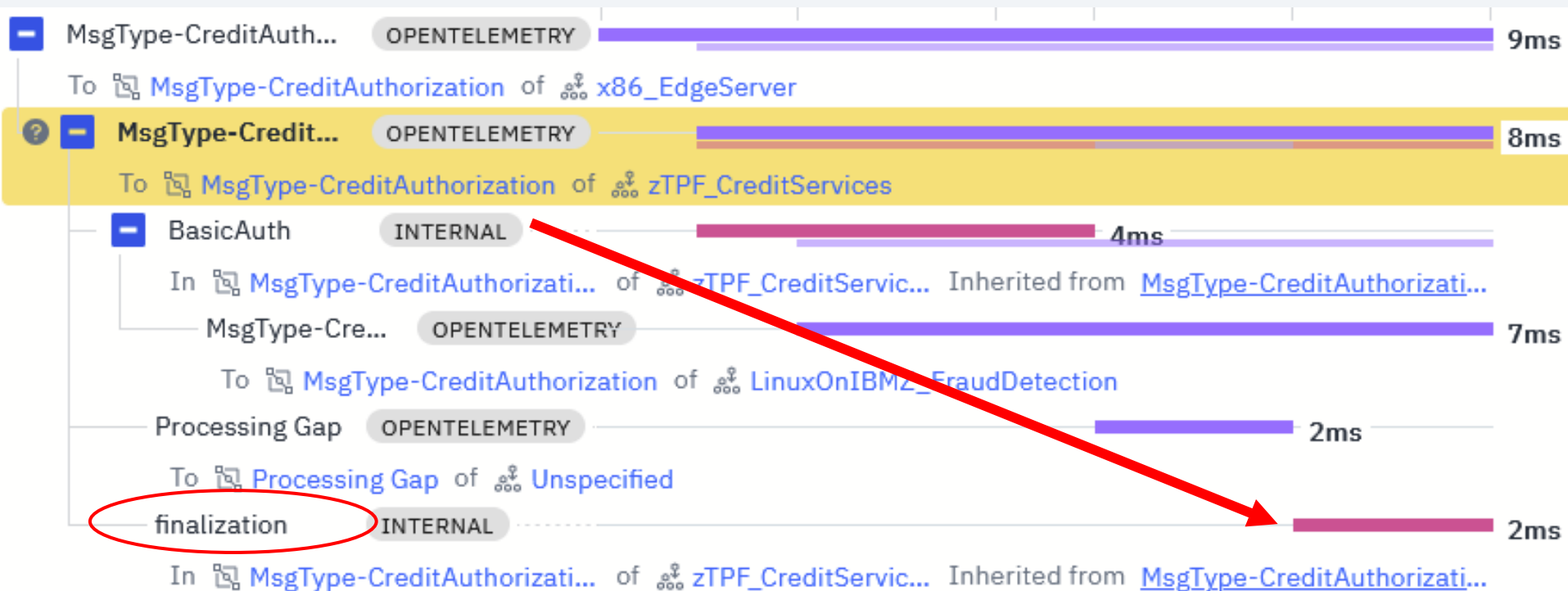
### Linux on IBM Z returns to z/TPF



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

### z/TPF processing continues in finalization processing

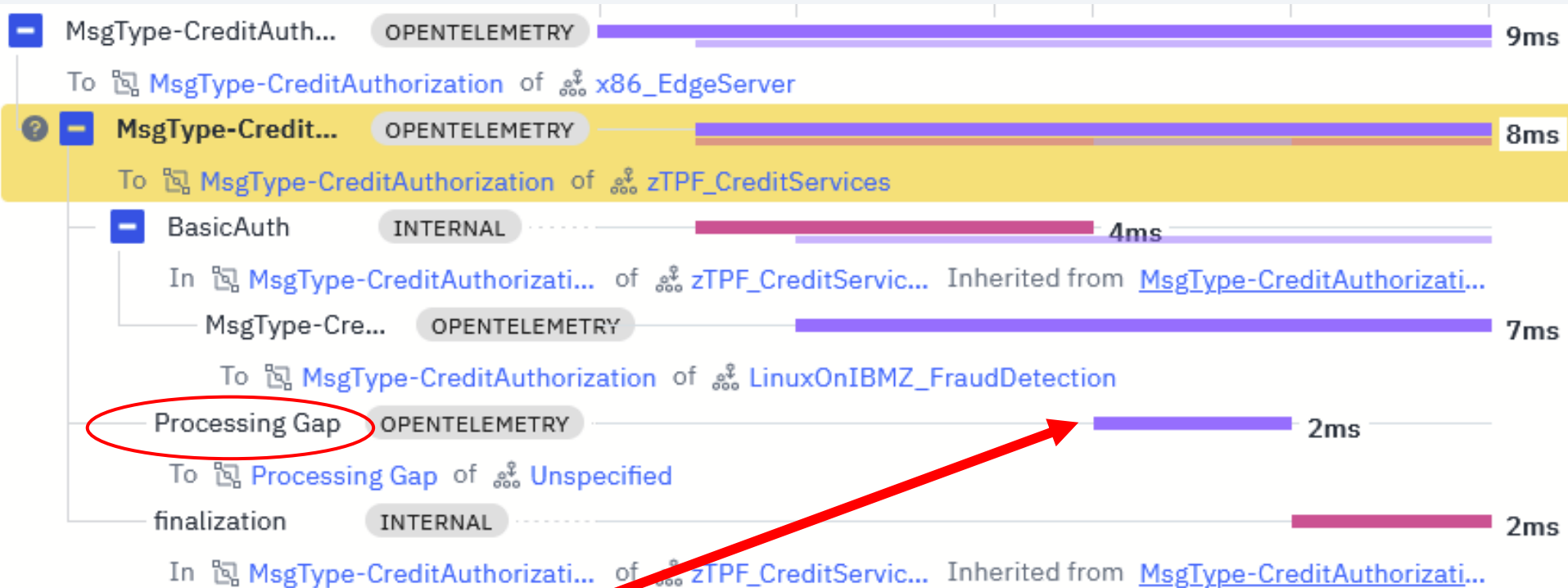




# Instana: z/TPF analyze call details

Credit authorization message ending in success

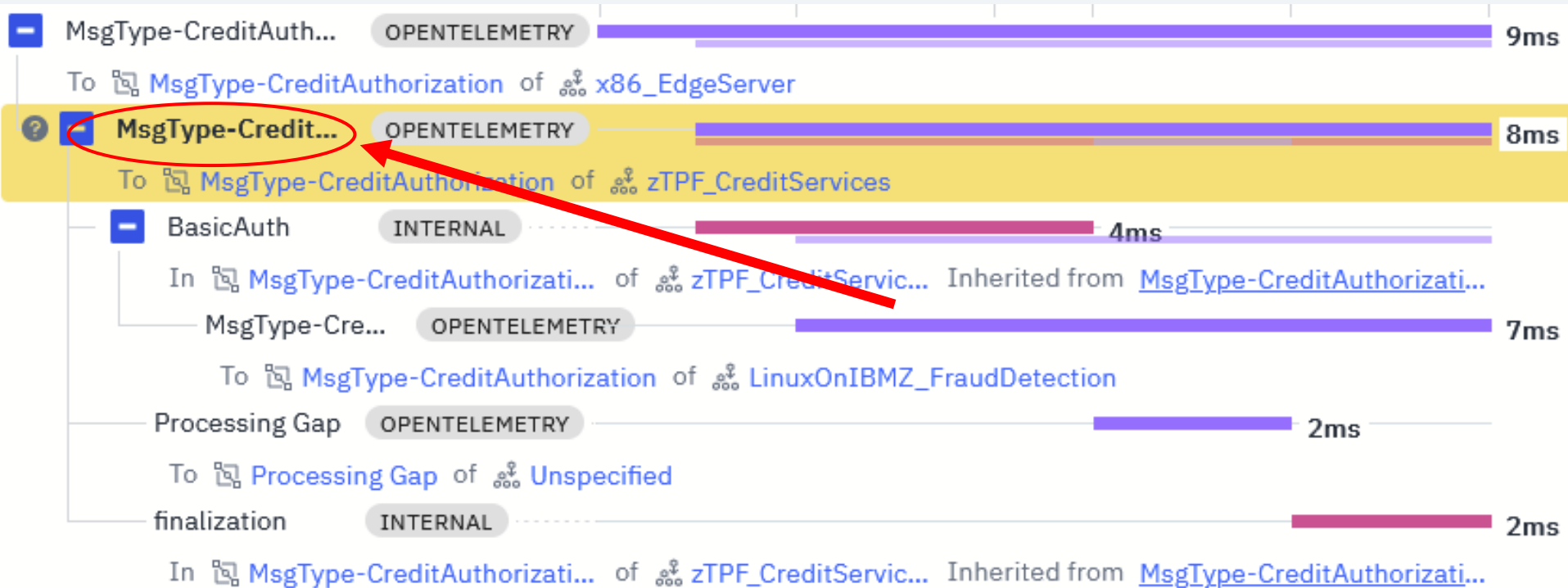
z/TPF logs the time gap between the processing on z/TPF



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

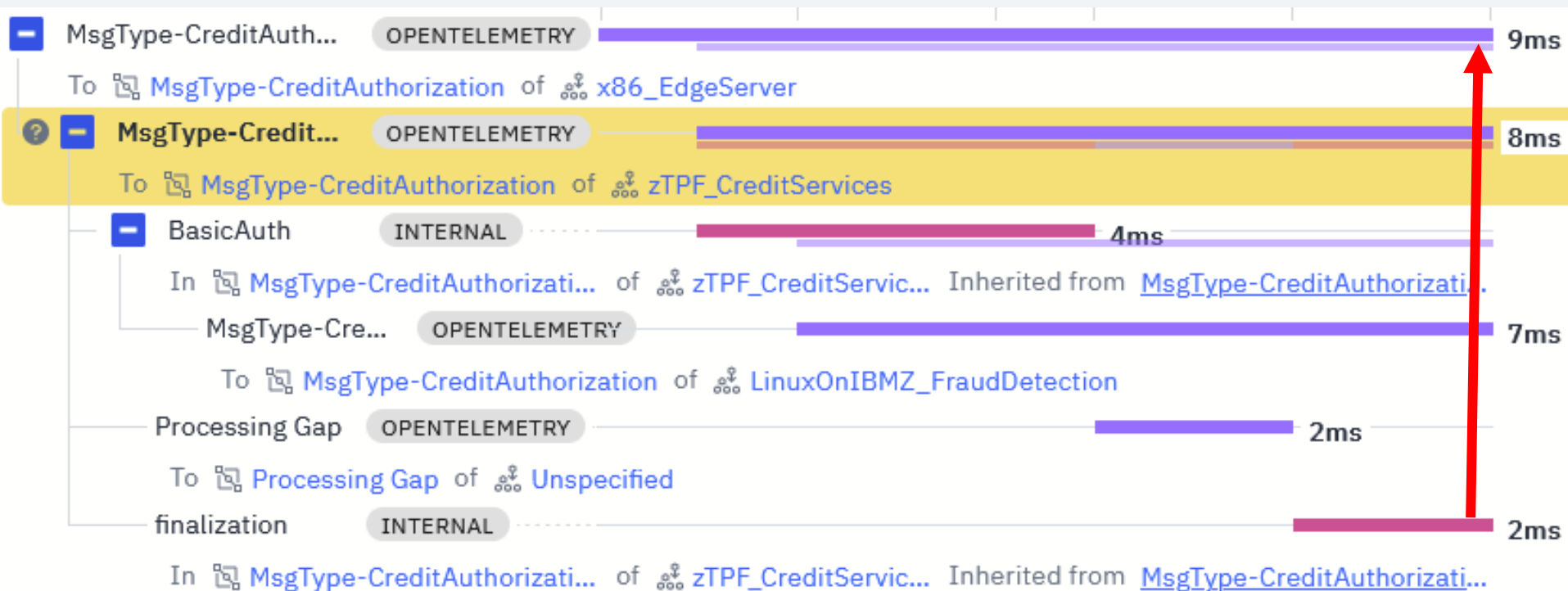
### z/TPF provides summary of work done on z/TPF



# Instana: z/TPF analyze call details

## Credit authorization message ending in success

### z/TPF returns to the edge server



**Screen shot:** Instana: Call Diagram – Successful message – Name-value pairs and metrics

**Capabilities:** In your APM tool, we can see the values for name-value pairs (like MsgType, Channel, and so on). If you implement the z/TPF OpenTelemetry name-value pair conventions, you can also see the return code, return code message, and ECB purpose to help you understand the processing.

You can also see key metrics like CPU used, existence time, and finds and files to DASD representing I/O operations.

You can modify which name-value pairs and metrics are included in the displays.

You can see these name-value pairs and metrics at the overall message level, which shows the summation of the metrics of the individual processing. Or you can see the name-value pairs and metrics for each individual process.

**Business value:** With your APM tools, you can inspect key metrics and details for each system where the message was processed.

# Instana: z/TPF analyze call details

## Credit authorization message ending in success

### Name-value pairs and metrics

Service

zTPF\_CreditServices

Operation

MsgType-CreditAuthorization

Tags

MESSAGE_LIFETIME	8321
TPF_OTEL_ecb_purpose	BasicAuth
TPF_OTEL_rc	0X00000000
IUOWID	0XC2C3D6F0F9F0F140...
FILE_DASD	0
TPF_OTEL_rc_msg	SUCCESS

# **Application performance monitor tooling – Story 1**

## **Over the credit limit errors**

## **Screen shot:** Instana: Alert

**Story:** First, we'll play the role of a site reliability engineer (SRE) in this error scenario. We will debug this together and I'll show you various features that are available today.

We received an Alert. This could be an email, text, slack message, or whatever your APM tool supports. There's an issue in our enterprise. Remember, the SRE is not a z/TPF expert.

This alert shows that the credit authorization error rate has broken the 5% error rate threshold. Glancing at this, it looks like the increase in errors is originating on z/TPF and has something to do with over the credit limit errors.

**Business value:** APM tools can alert you to conditions in your enterprise and provide insights as to which system to investigate further, possibly allowing you to remediate issues before SLAs are impacted.

# Instana: Alert!

**EventId:**

PDsh8nkDTuef2KgbmhA3Eg

**Link:**

<https://instana.fake.com/#/events:eventID=PDsh8nkDTuef2KgbmhA3Eg&incidentTo=16345989153>

**Incident started with:**

Credit Authorization message error rate exceeds 5% threshold.  
These violations are occurring continuously since 12:12:10 EST 4/26/2025.  
SLAs may be violated within the next 10 minutes.

**Detail:**

The increase is Credit Authorization message error rate is originating on z/TPF.  
Specifically, over the credit limit errors are occurring more frequently while other error types are occurring at a stable rate.

**Severity:** Critical



Sarah  
site  
reliability  
engineer



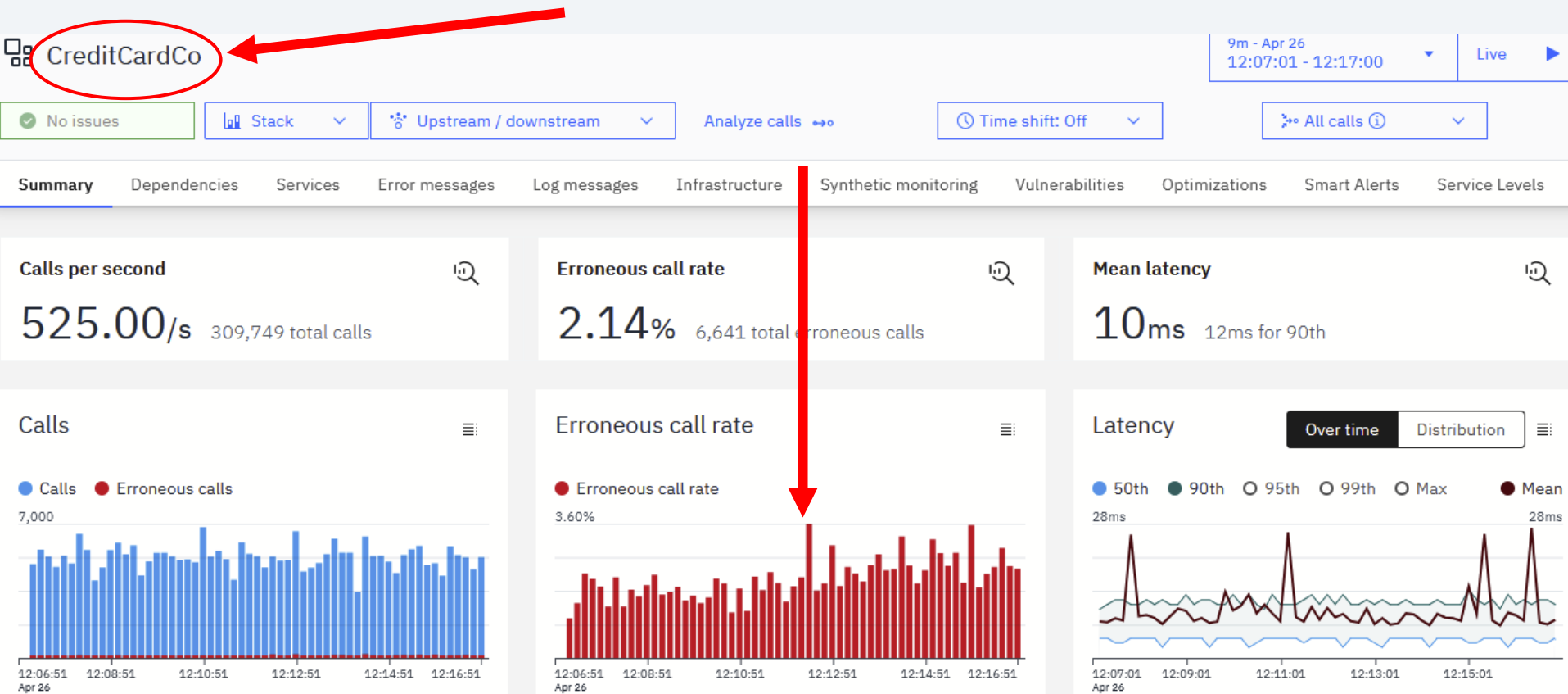
## **Screen shot:** Instana: Enterprise View – All Services

**Story:** As the SRE, we will first look at the health of the overall enterprise. In this case, our message rates are unchanged, latency is unchanged, and we're getting slightly more errors than usual, but it's actually a bit difficult to recognize. It's a good thing we got the alert. We'll keep moving.

**Business value:** APM tools retain, display, and can analyze historical data and call out trends and change points for your enterprise.

# Instana: Enterprise-wide view

## Slight increase in errors across all services in the enterprise



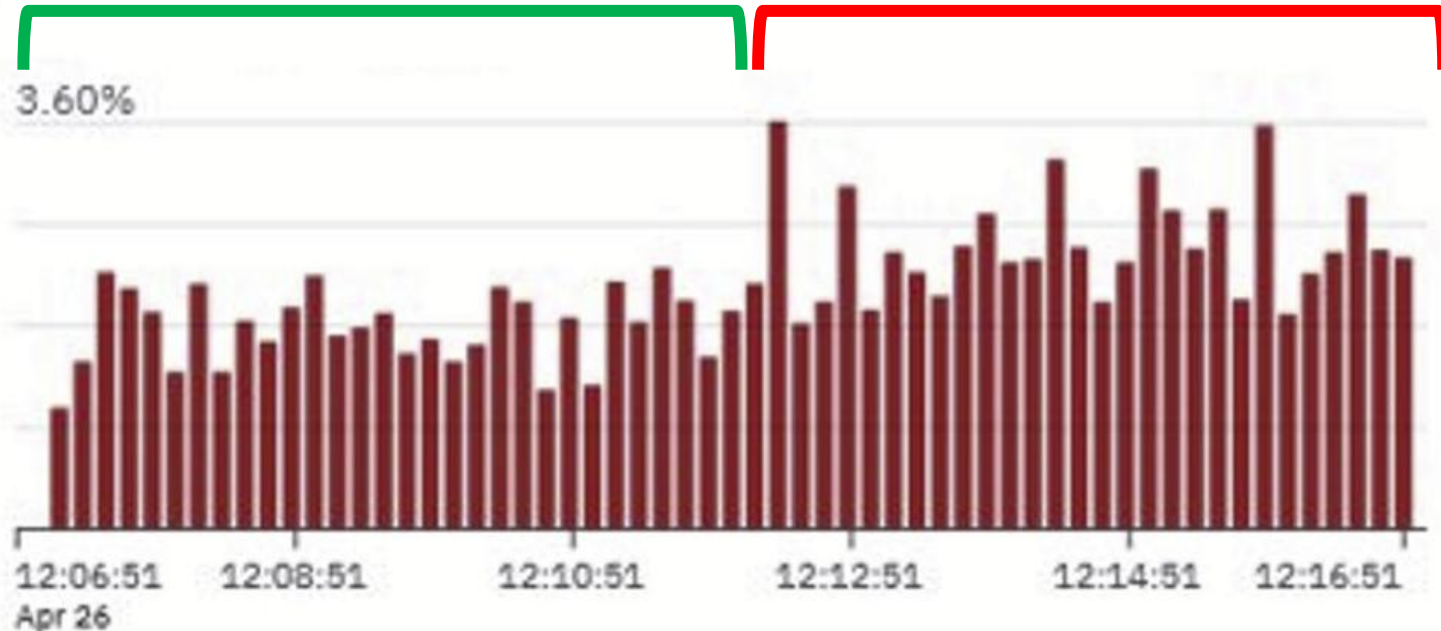
# Instana: Enterprise-wide view

Erroneous call rate graph shows a slight increase in errors

Inflection point time known from the alert

5 minutes before inflection  
point – normal traffic

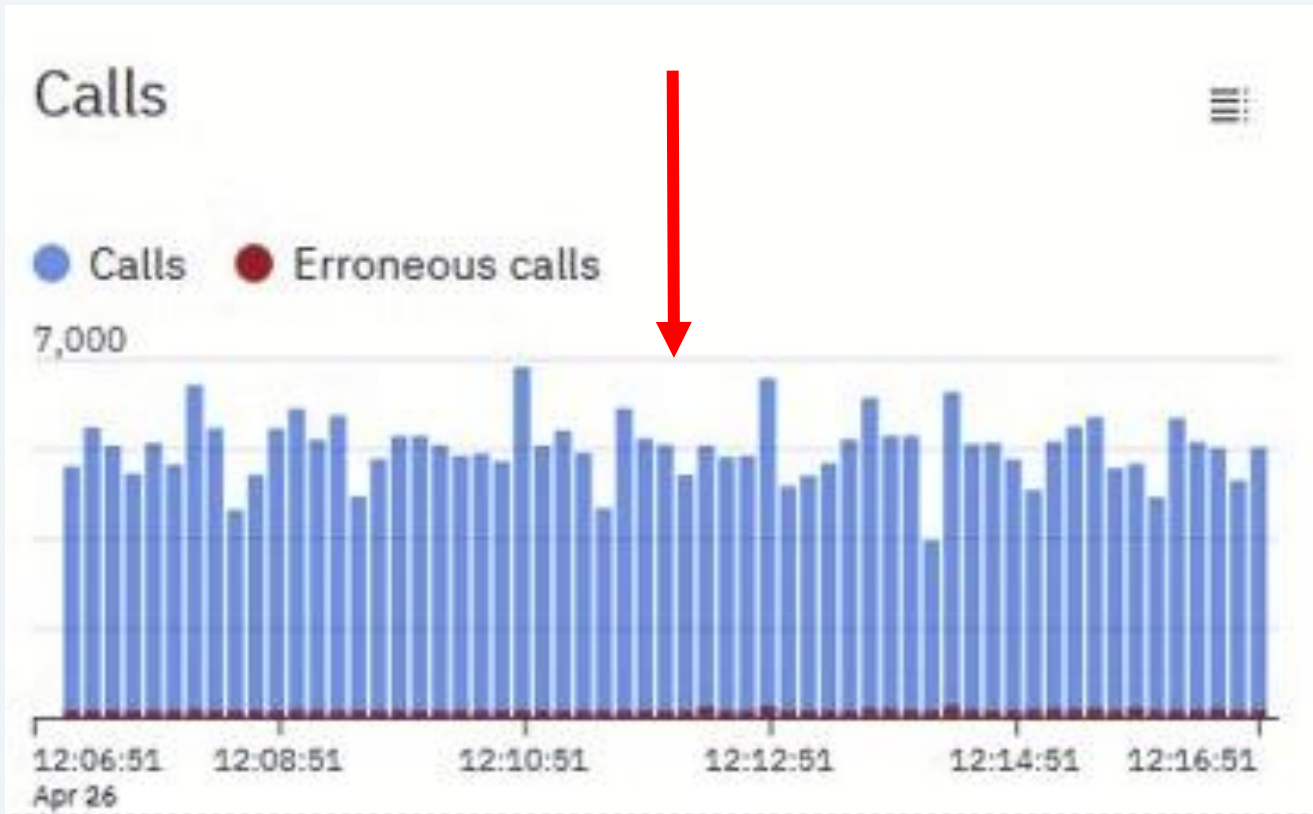
5 minutes after inflection  
point – additional errors



# Instana: Enterprise-wide view

Calls graph shows no change in the message rate

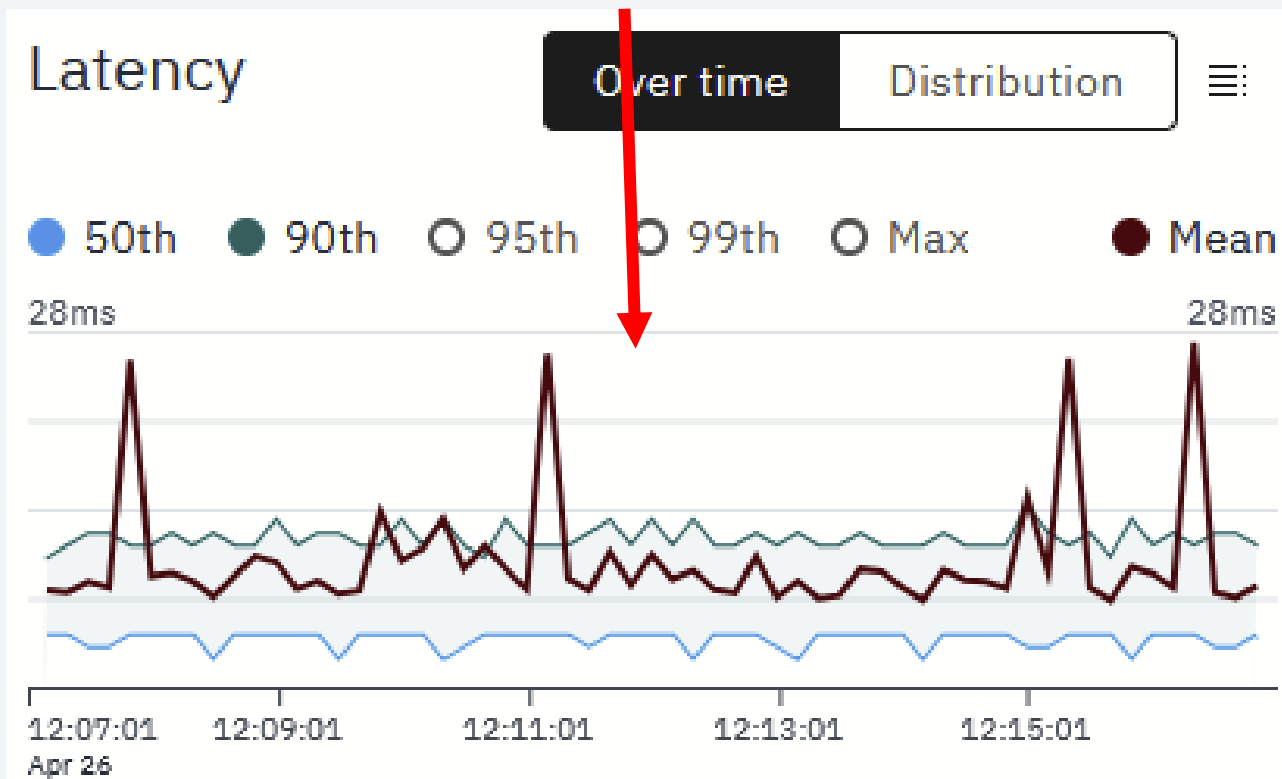
Inflection point time known from the alert



# Instana: Enterprise-wide view

Latency graph shows no change in response time

Inflection point time known from the alert



**Screen shot:** Instana: System and user-defined metrics

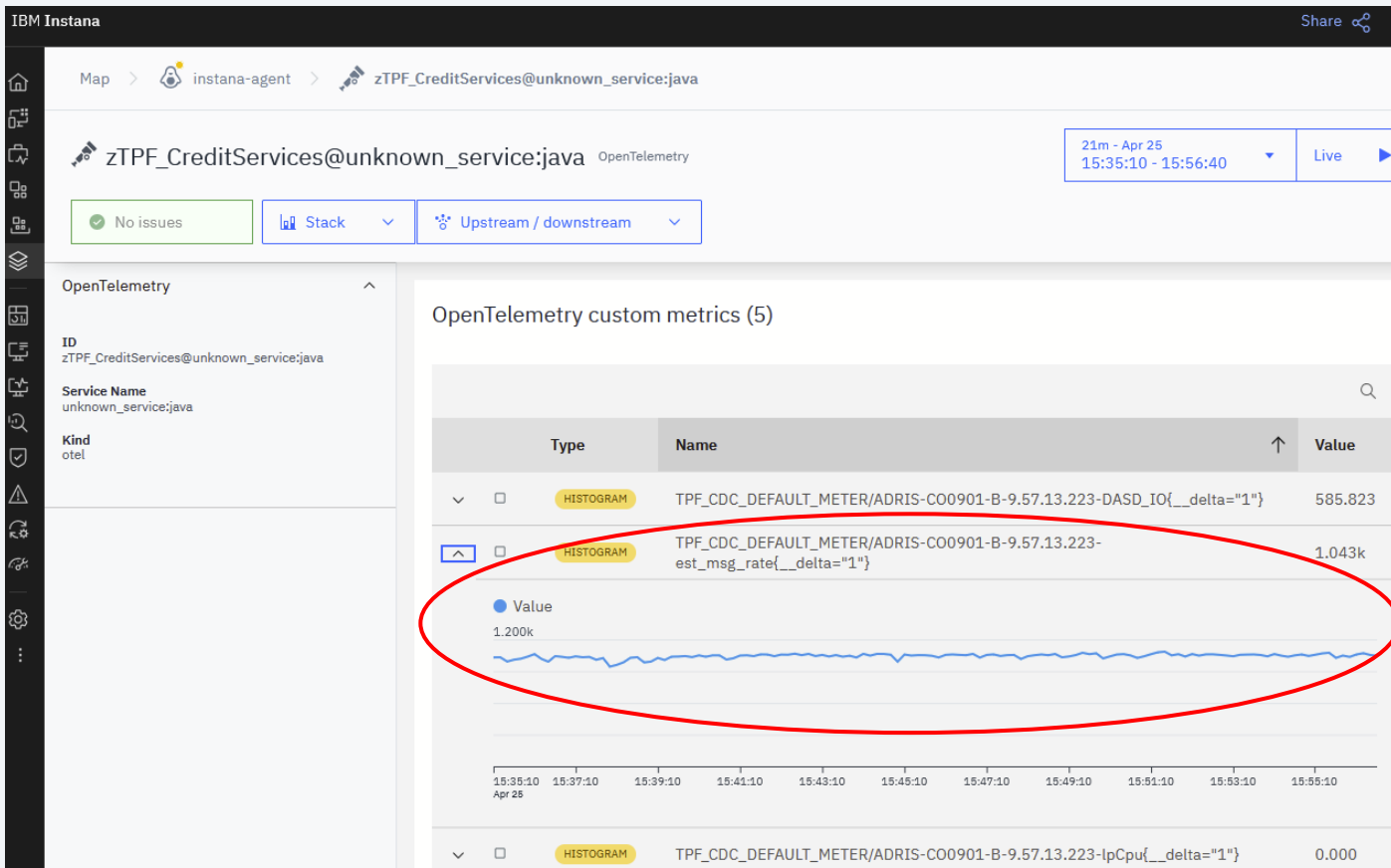
**Story:** As we saw in the normal case, we next drill down on the specific system mentioned in our alert, z/TPF in this scenario.

We can first look at key system level metrics to understand the health of the z/TPF system overall. Remember this includes the CDC system level metrics and your system, application, and business level user-defined metrics. Everything looks normal so we'll keep moving.

**Business value:** APM tools can collect, display, and analyze system level metrics including your system, application, and business user-defined metrics.

# Instana: z/TPF system metrics

## Estimated message rate constant at 1000 messages per second



## **Screen shot:** Instana: Service Dashboard – z/TPF

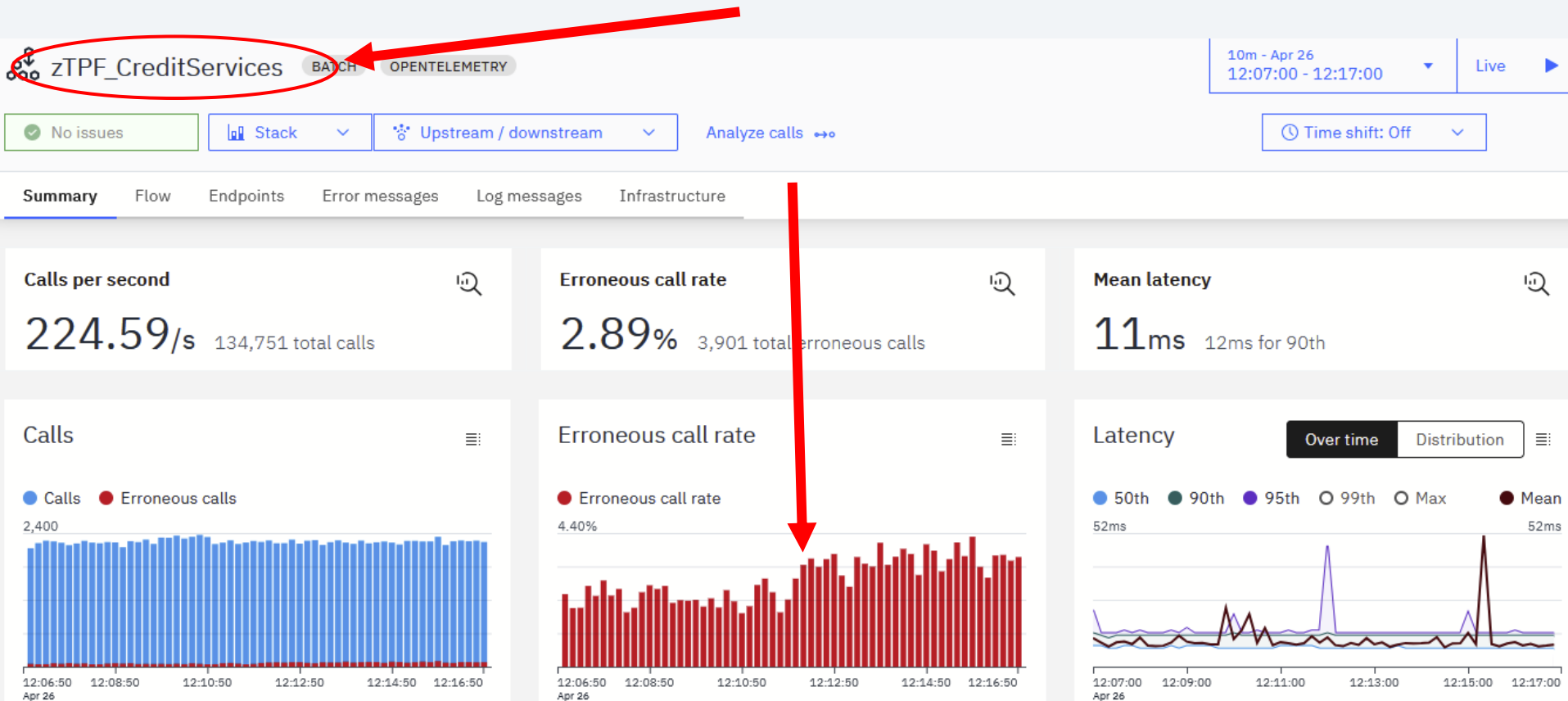
**Story:** As we did in the normal case, we look closer at the trace data on z/TPF. In this case, we can see the message rate for credit authorization is unchanged, the error rate increased 5 minutes ago, and the response time looks unchanged. The change in the error rate is a bit more obvious now that we've drilled down to just the z/TPF system. The timestamp in the alert also indicated when the increase in errors began. It looks like it was historically consistent but suddenly rose and has been consistently higher since.

**Business value:** APM tools retain, display and can analyze historical data and call out trends and change points.



# Instana: z/TPF system view

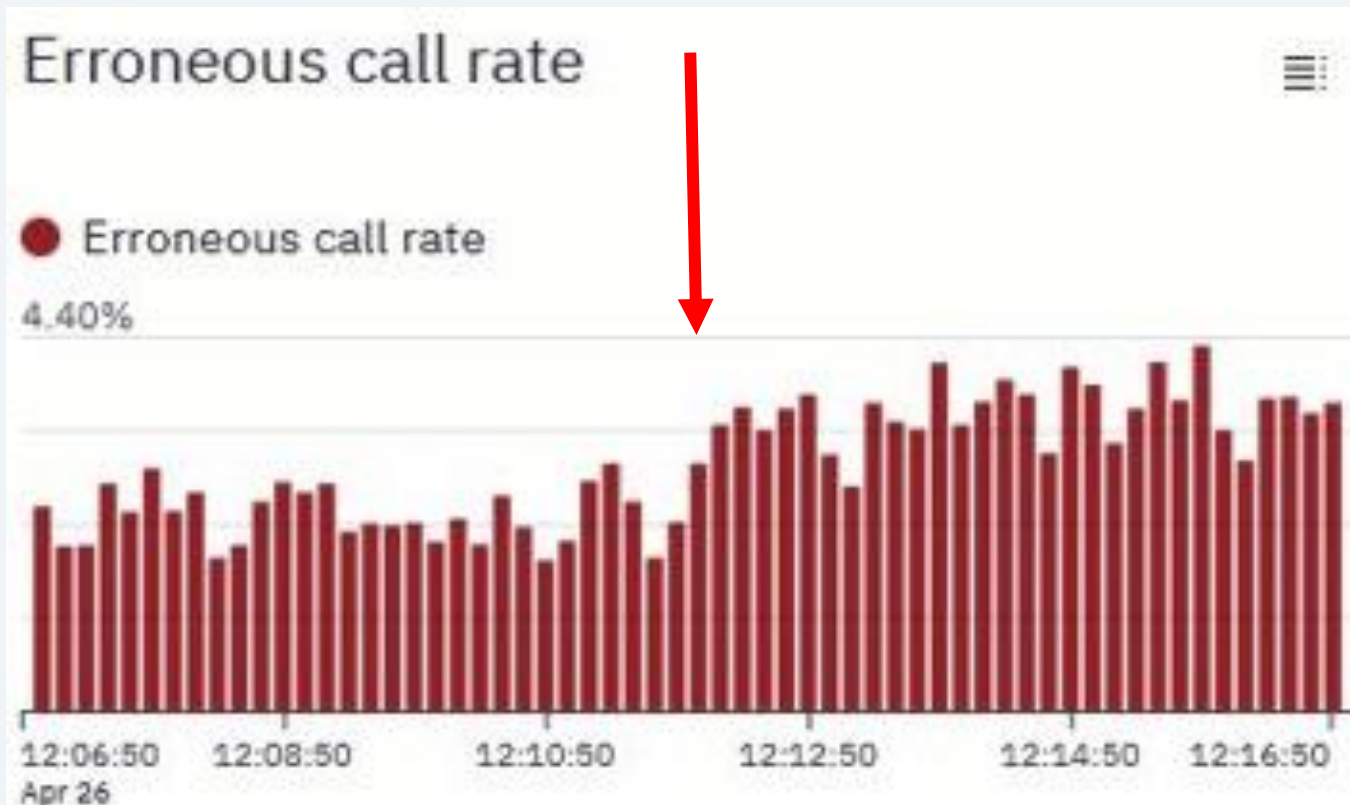
## Small increase in errors across all services in z/TPF



# Instana: z/TPF system view

Erroneous call rate graph shows a small increase in errors

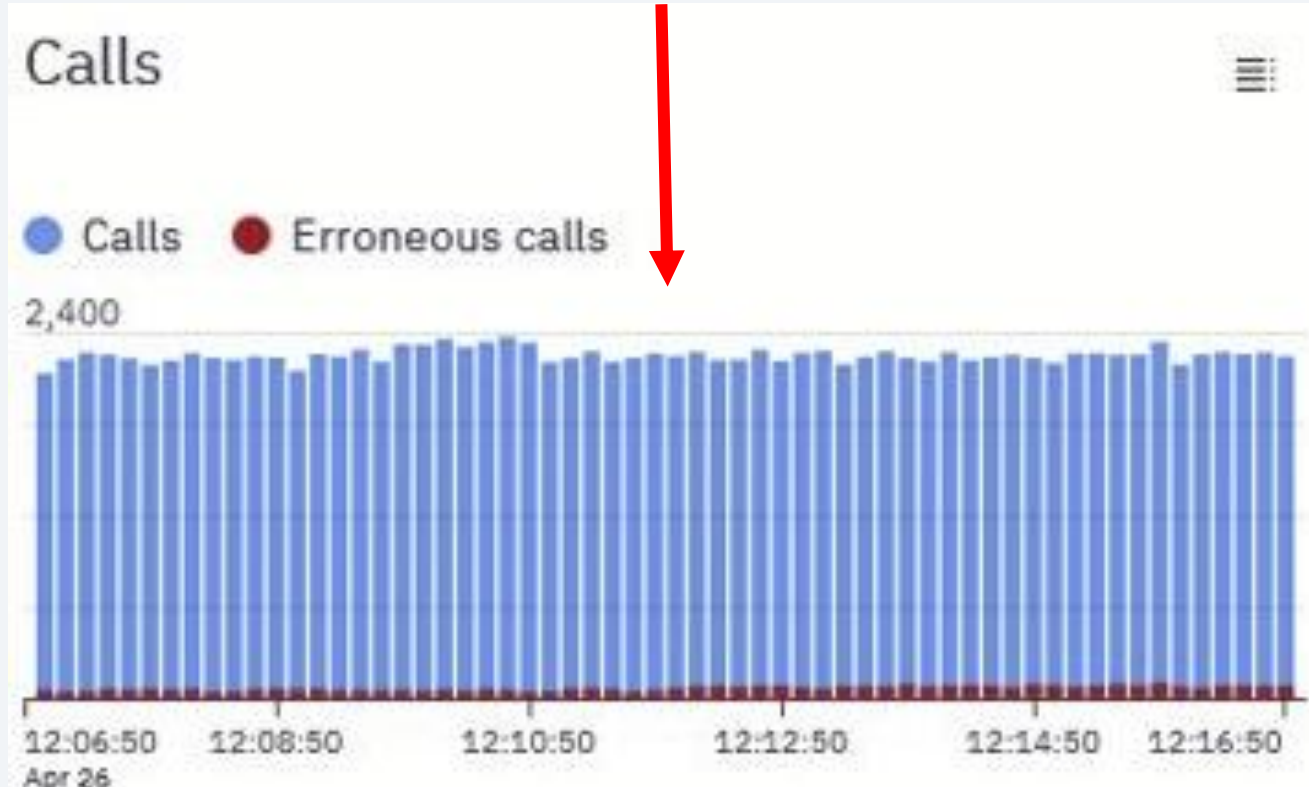
Inflection point time known from the alert



# Instana: z/TPF system view

Calls graph shows no change in the message rate

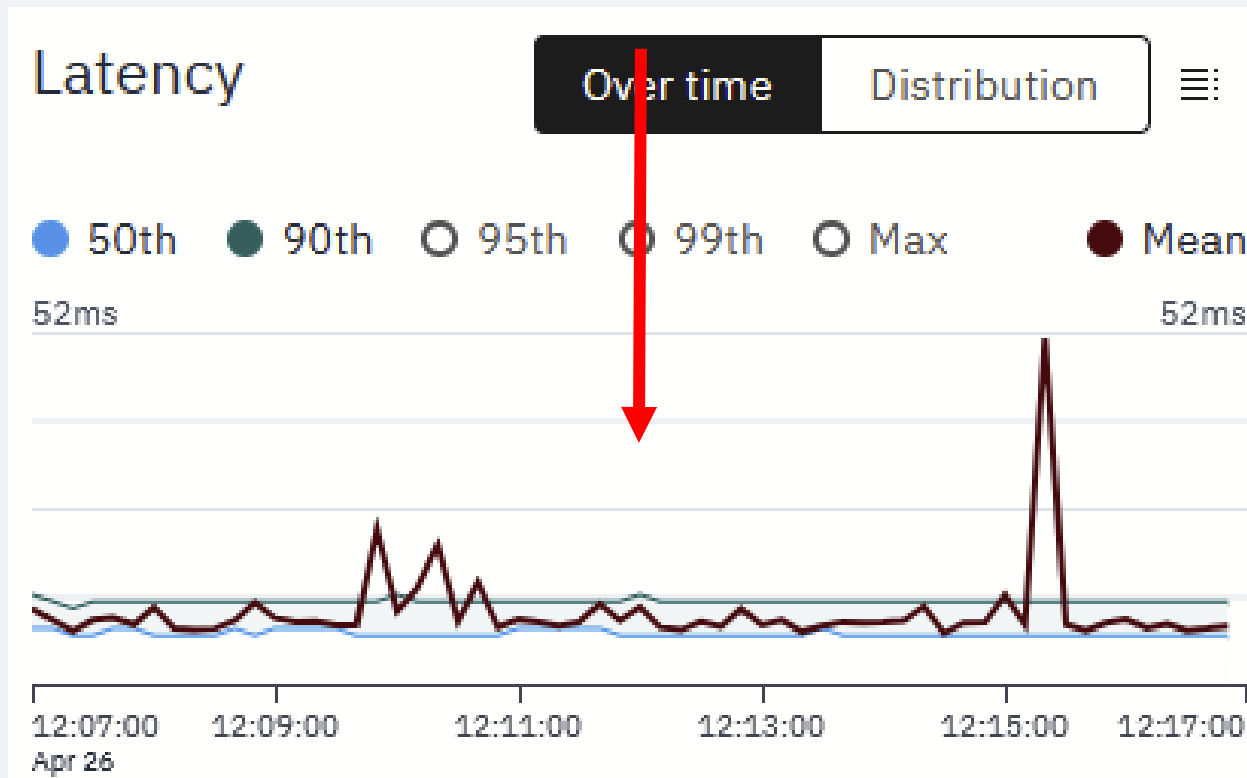
Inflection point time known from the alert



# Instana: z/TPF system view

Latency graph shows no change in response time

Inflection point time known from the alert



## **Screen shot:** Instana: Analyze Calls – Graphs

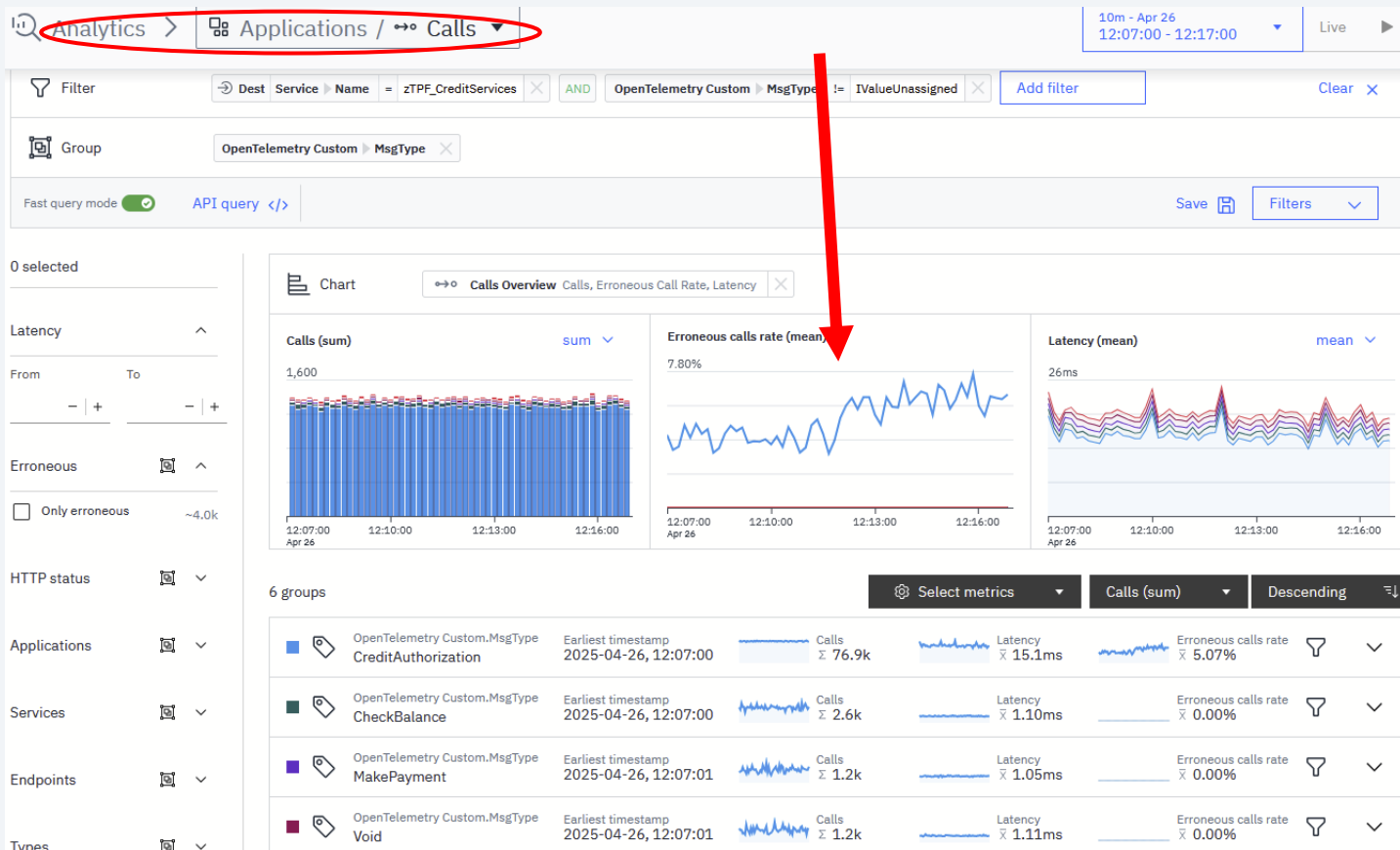
**Story:** Next, we click on the analyze calls to investigate samples of our transactions at a deeper level. Here we can see the messages grouped by the name-value pair MsgType.

Looking at error rates history for the different message types, we can see that the error rates and response times for other message types are unchanged over the same time period. We now know this looks like a problem that is isolated to credit authorization requests.

**Business value:** APM tools can show you sample rates, error rates, and response time by the type of messages processed by your system.

# Instana: z/TPF analyze calls

## Small increase in errors across all services in z/TPF



# Instana: z/TPF analyze calls

## Grouped by MsgType



Filter



Dest

Service

Name

=

zTPF\_CreditServices



Add filter



Group

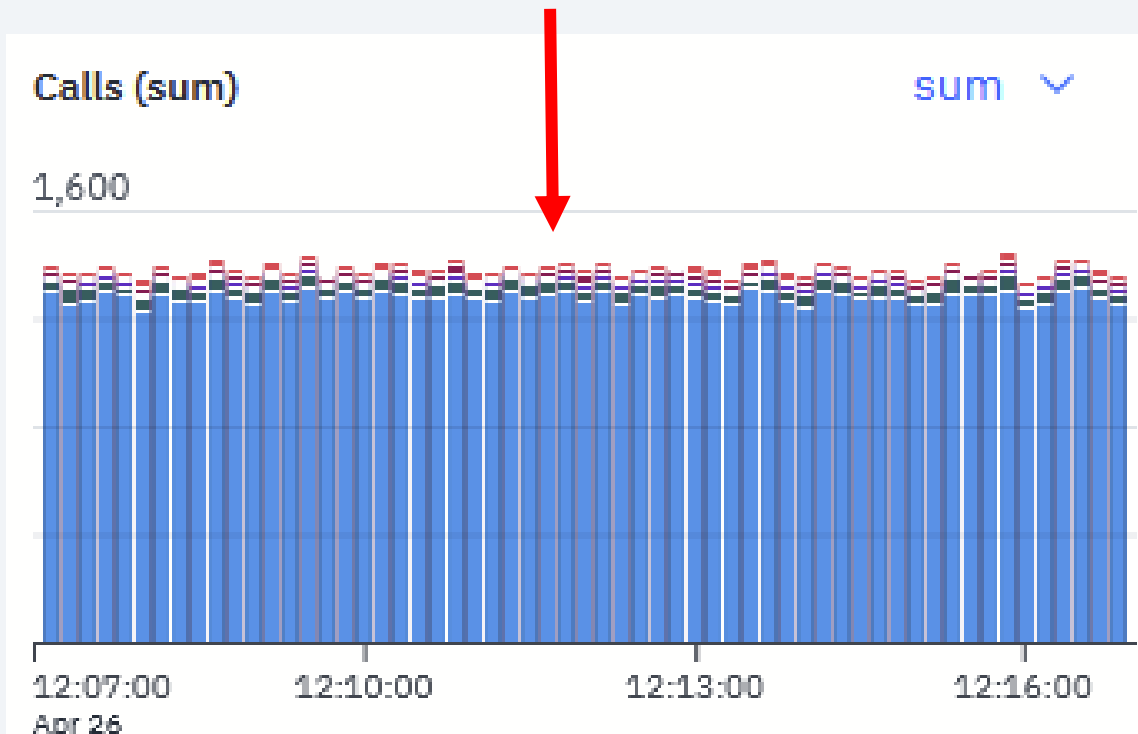
OpenTelemetry Custom ▶ MsgType



# Instana: z/TPF analyze calls

Calls graph grouped by MsgType shows no change in the message rate

Inflection point time known from the alert

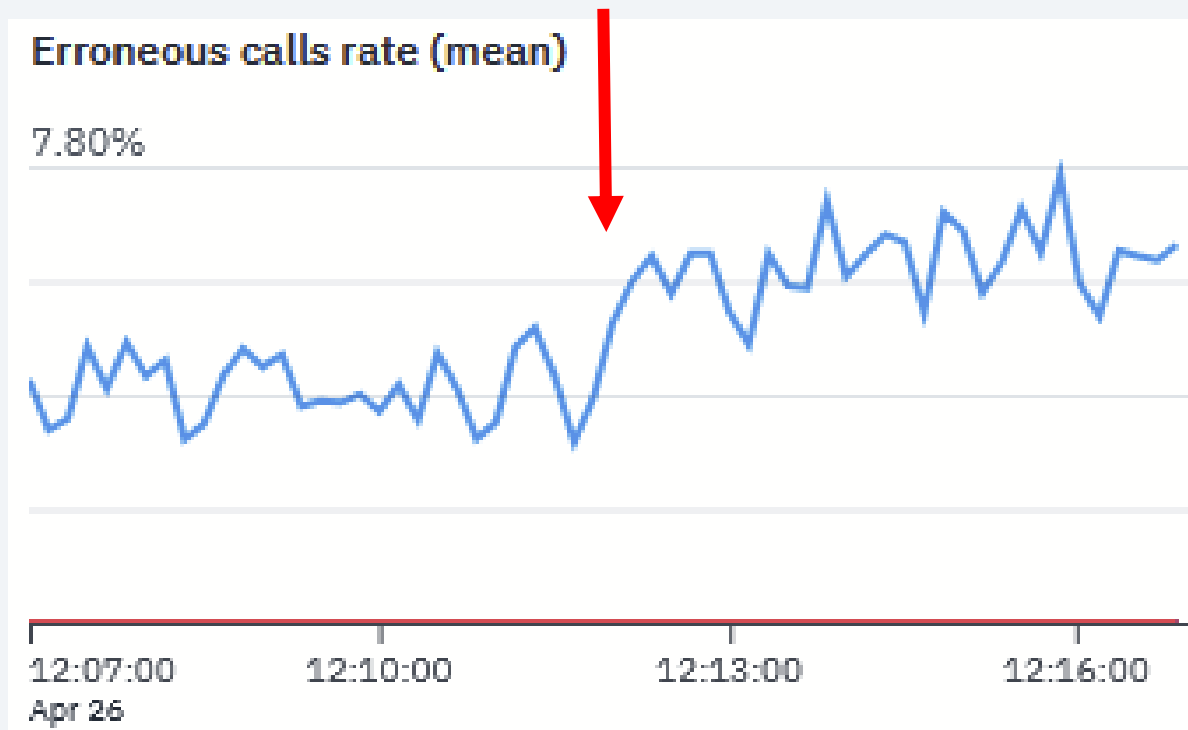




# Instana: z/TPF analyze calls

Erroneous call rate graph grouped by MsgType shows a small increase in errors

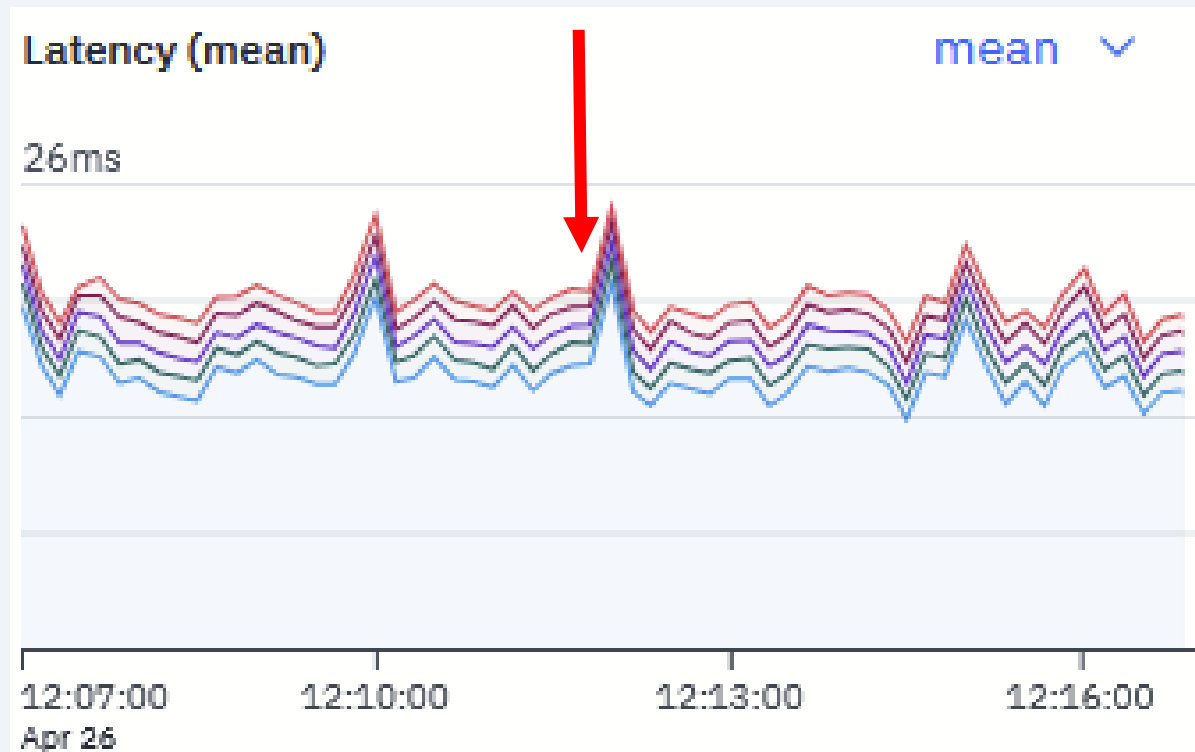
Inflection point time known from the alert



# Instana: z/TPF analyze calls

Latency graph grouped by MsgType shows no change in response time

Inflection point time known from the alert



# Instana: z/TPF analyze calls

## Error rate history by message type

### Credit authorization shows increase in error rate



OpenTelemetry Custom.MsgType CreditAuthorization	Earliest timestamp 2025-04-26, 12:07:00	 Calls Σ 76.9k	 Latency x̄ 15.1ms	 Erroneous calls rate x̄ 5.07%
OpenTelemetry Custom.MsgType CheckBalance	Earliest timestamp 2025-04-26, 12:07:00	 Calls Σ 2.6k	 Latency x̄ 1.10ms	 Erroneous calls rate x̄ 0.00%
OpenTelemetry Custom.MsgType MakePayment	Earliest timestamp 2025-04-26, 12:07:01	 Calls Σ 1.2k	 Latency x̄ 1.05ms	 Erroneous calls rate x̄ 0.00%
OpenTelemetry Custom.MsgType Void	Earliest timestamp 2025-04-26, 12:07:01	 Calls Σ 1.2k	 Latency x̄ 1.11ms	 Erroneous calls rate x̄ 0.00%

## **Screen shot:** Instana: Analyze Calls – Graphs

**Story:** The error rates are even more clear if we filter on credit authorization message type and look at the latency before and after the point of inflection.

**Business value:** APM tools can show you sample rates, error rates, and response time by the type of messages processed by your system.

# Instana: z/TPF analyze calls

## Filter on credit authorization messages



Filter



Dest

Service

Name

=

zTPF\_CreditServices



AND

OpenTelemetry Custom

MsgType

=

CreditAuthorization



Group

OpenTelemetry Custom

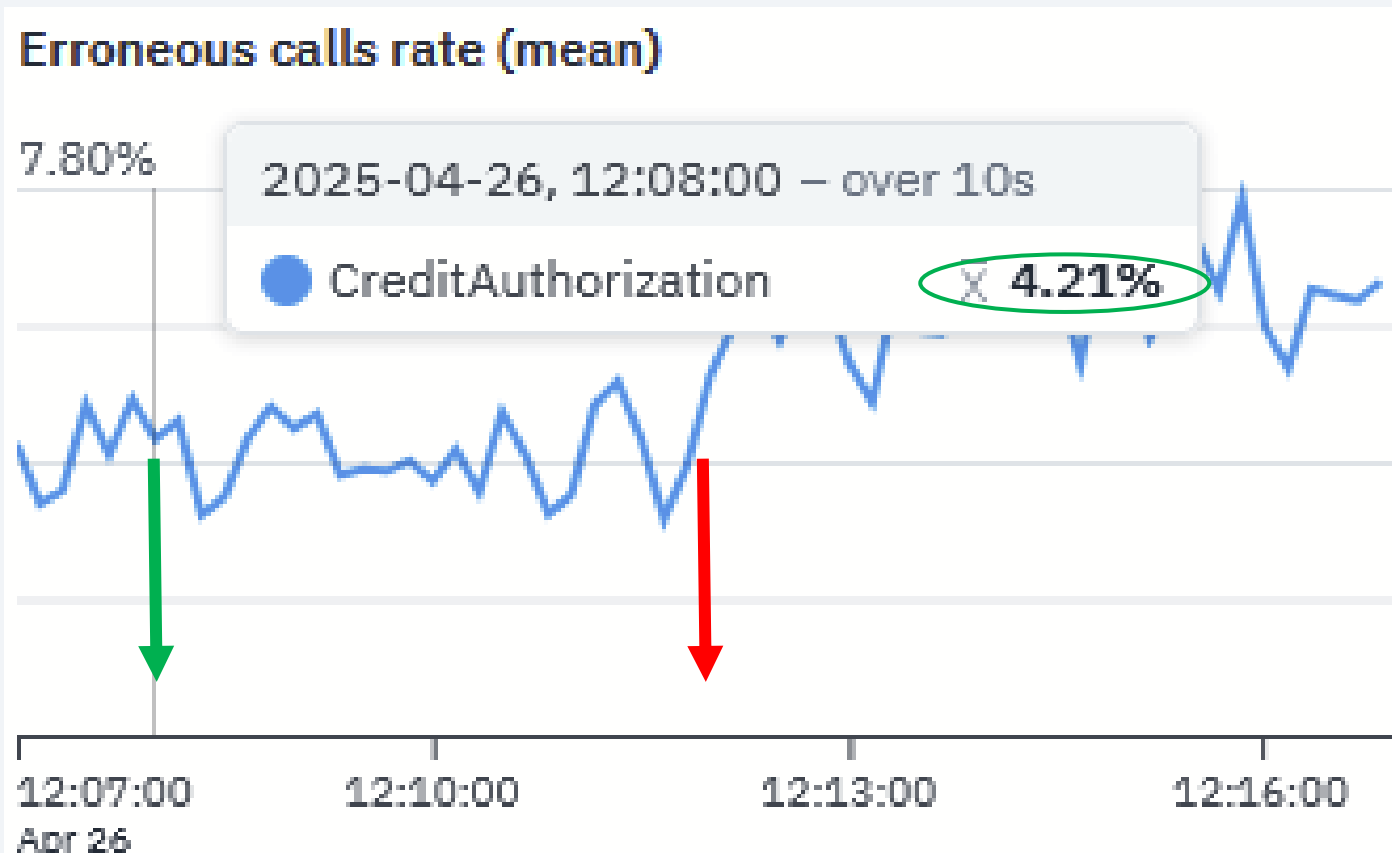
MsgType



# Instana: z/TPF analyze calls

Credit authorization error rate **before** the inflection point is 4.21%

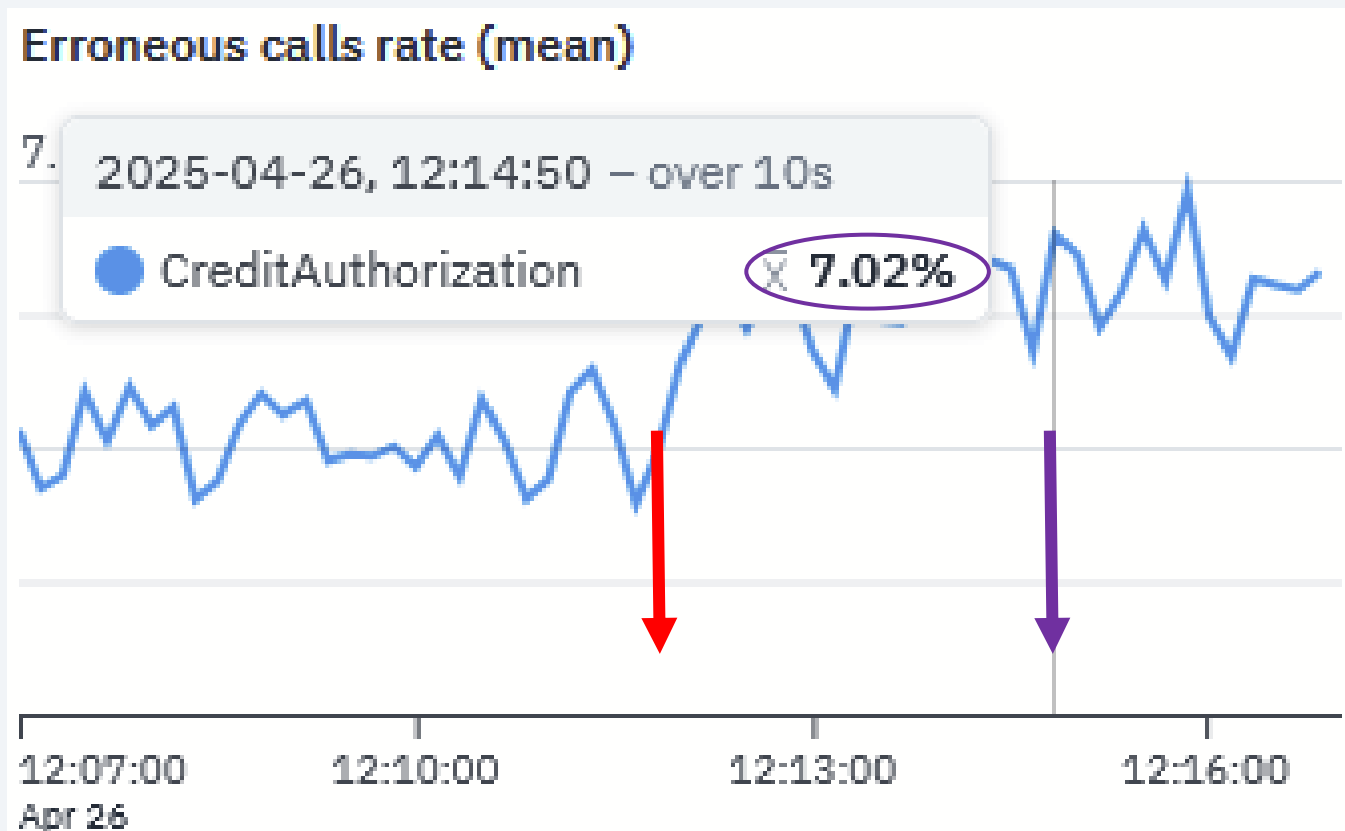
**Inflection point** time known from the alert



# Instana: z/TPF analyze calls

Credit authorization error rate after the inflection point is 7.02%

Inflection point time known from the alert



## **Screen shot:** Instana: Analyze Calls – Graphs – RC\_msg breakdown

**Capabilities:** If you have other name-value pairs in place, you could do more filtering and peering through the data. Maybe it's credit authorizations through a particular channel or bank. Maybe it's credit authorizations of a particular type.

**Story:** In this scenario, we'll filter on credit authorization messages and group by the return code message. Did the change in the error rate occur for a particular return code that we should focus on?

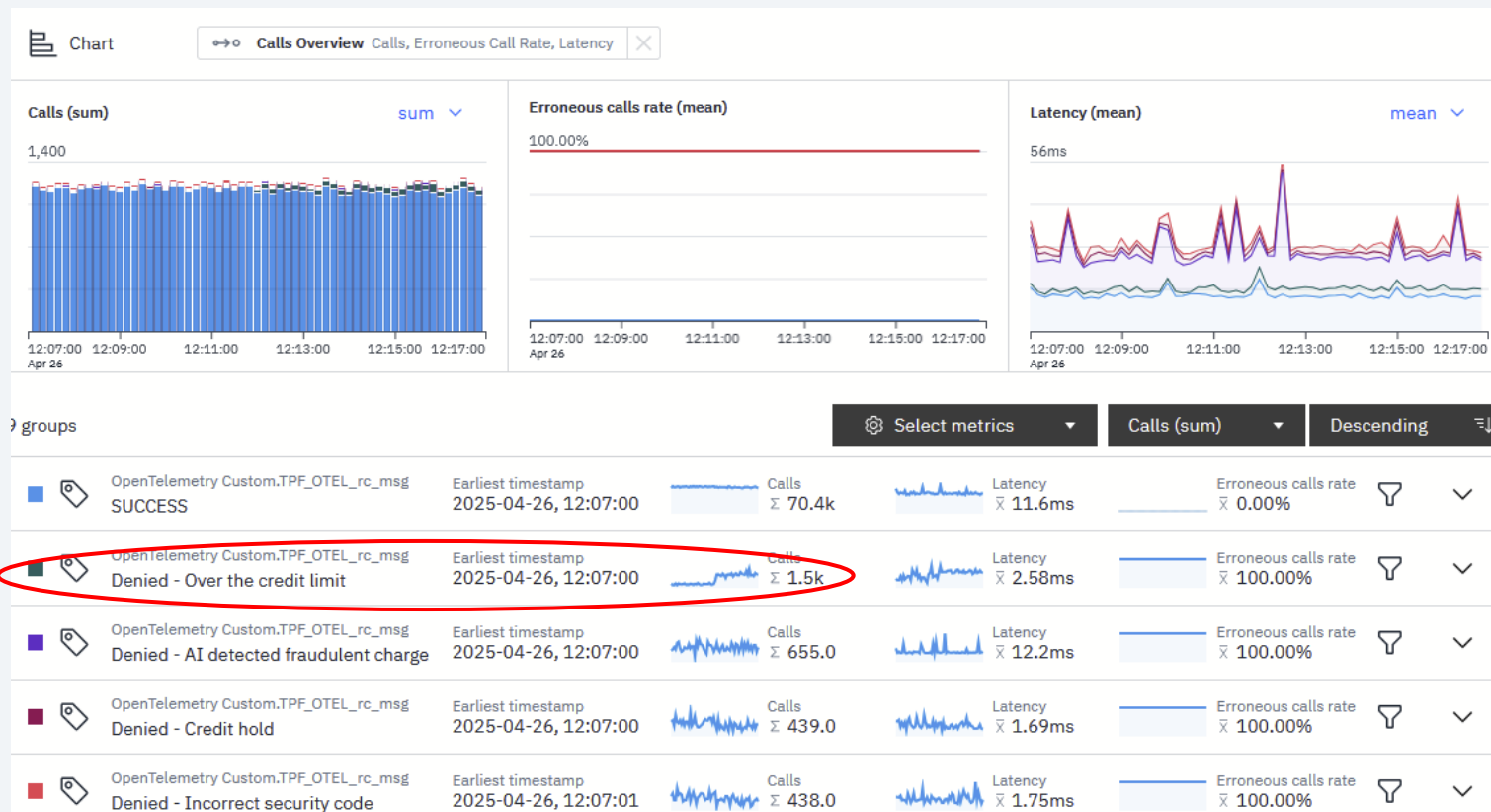
We can see that the errors that changed are for RC\_msg “Denied – Over the credit limit”. This is a great insight that helps us to focus our search on code or aspects that affect over the credit limit processing.

**Business value:** APM tools provide filter and breakdown by the name-value pairs you have implemented on your system views so you can glean insights into problematic messages.



# Instana: z/TPF analyze calls

Credit authorization messages grouped by return code message  
Over the credit limit errors show a rise in the error rate



# Instana: z/TPF analyze calls

## Filter credit authorization messages and grouped by return code message



Filter



Dest

Service

Name

=

zTPF\_CreditServices



AND

OpenTelemetry Custom

MsgType

=

CreditAuthorization



Group

OpenTelemetry Custom

TPF\_OTEL\_rc\_msg



# Instana: z/TPF analyze calls

Group by return code message name-value pair

Call graph shows increase in frequency of over the credit limit errors



OpenTelemetry Custom.TPF\_OTEL\_rc\_msg

Denied - Over the credit limit

Earliest timestamp

2025-04-26, 12:07:00



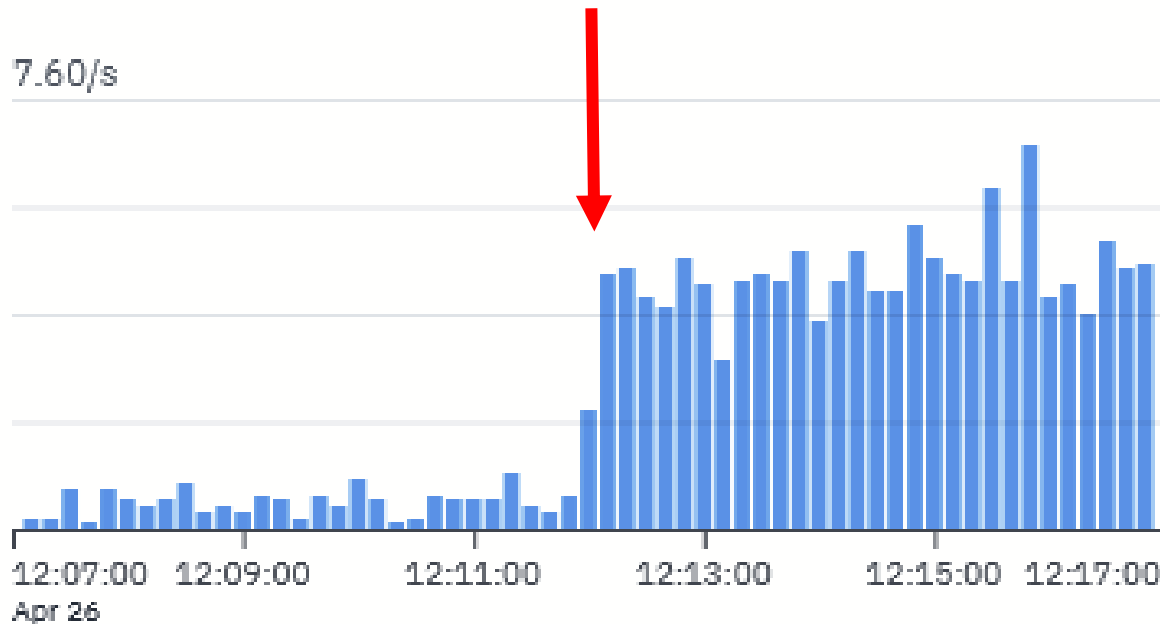
Calls

Σ 1.5k

Calls (per second)

per second ▾

7.60/s



## **Screen shot:** Instana: Analyze Calls – Graphs – Latency

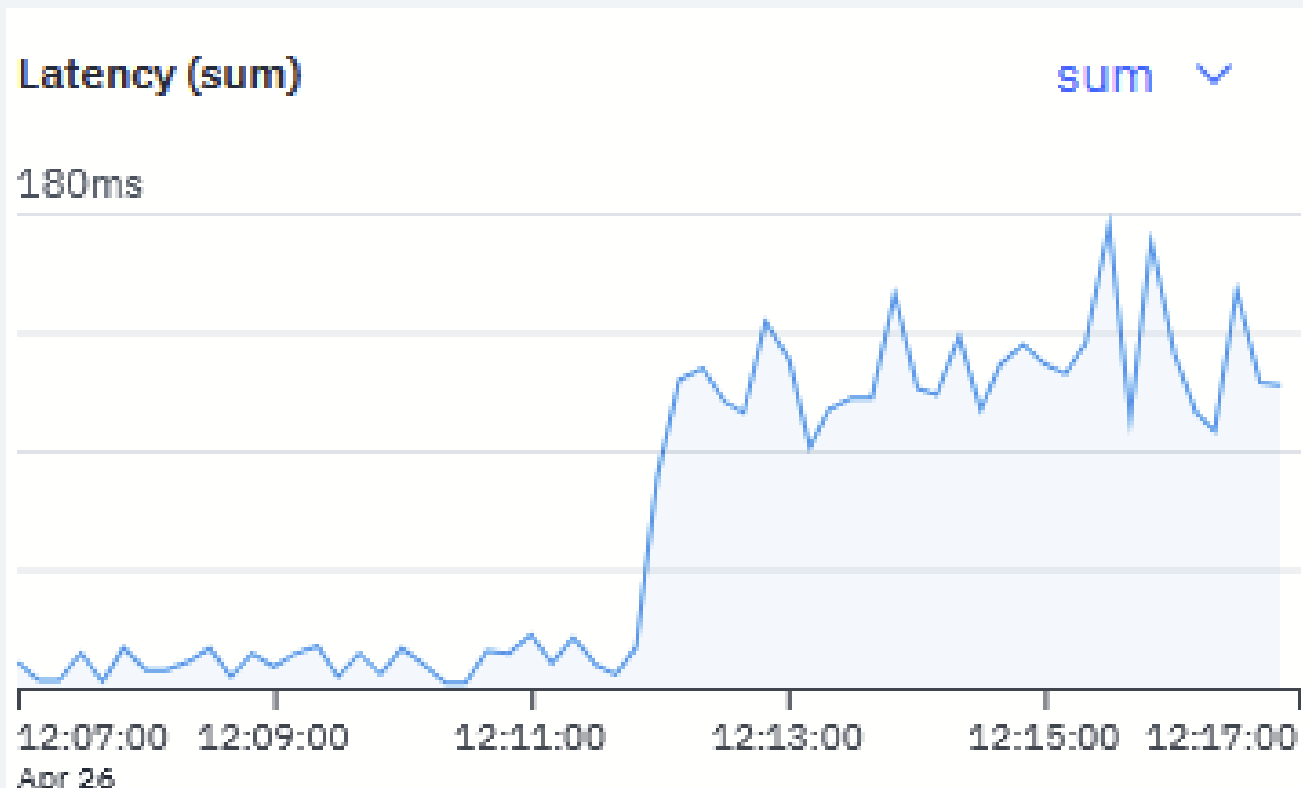
**Story:** We know that the over the credit limit errors are occurring more frequently. But there are other insights here that we can identify. Notice that the latency for the over the credit limit are larger than errors that occur before the inflection point. This is an indicator that some additional processing is occurring.

**Business value:** APM tools provide filter and breakdown by the name-value pairs you have implemented on your system views so you can glean insights into problematic messages.

# Instana: z/TPF analyze calls

Group by return code message name-value pair

Latency graph shows increase in response time of over the credit limit errors



**Screen shot:** Instana: Analyze Calls – List of samples of messages with RC message over the credit limit




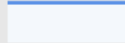





**Story:** As we saw before, with your APM tool you can see lists of different sampled messages.

In this case, we're looking at a list of samples that had the over the credit limit error. Before we look at one of these samples, let's quickly remind ourselves what a normal success message looks like to give ourselves context.

**Business value:** APM Tools provides lists of the sampled messages so you can compare and investigate messages of interest.

# Instana: z/TPF analyze calls list

## Filtering on credit authorization messages ending in over the credit limit error

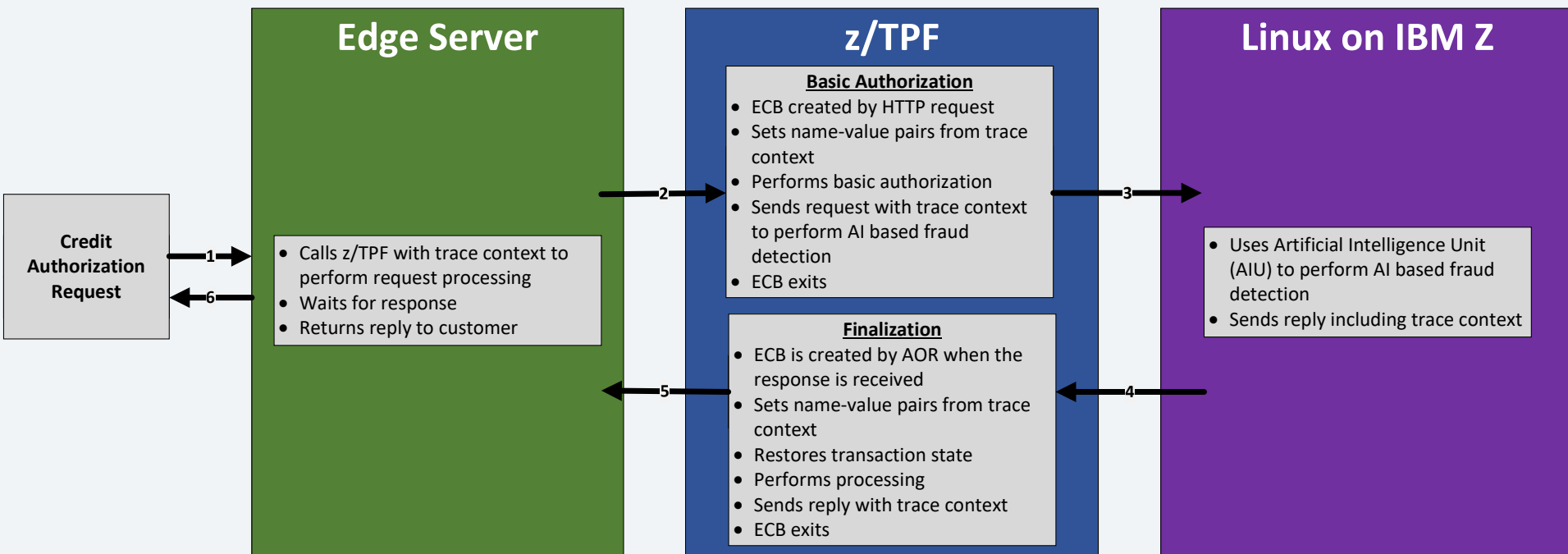
 OpenTelemetry Custom.TPF_OTEL_rc_msg Denied - Over the credit limit		Earliest timestamp 2025-04-26, 12:07:00	 Calls Σ 1.5k	 Latency X̄ 2.58ms	 Erroneous calls rate X̄ 100.00%	
Status	Call	Service	Timestamp	↓	Latency	
	↔↔ MsgType-CreditAuthorization	zTPF_CreditServices	2025-04-26, 12:16:59		4ms	
	↔↔ MsgType-CreditAuthorization	zTPF_CreditServices	2025-04-26, 12:16:59		3ms	
	↔↔ MsgType-CreditAuthorization	zTPF_CreditServices	2025-04-26, 12:16:58		1ms	
	↔↔ MsgType-CreditAuthorization	zTPF_CreditServices	2025-04-26, 12:16:58		3ms	

## **Screen shot:** Enterprise Architecture Diagram

**Story:** Remember in our enterprise architecture that we have three different systems involved in processing our message. The edge server calls z/TPF to do basic authorization tasks. z/TPF calls Linux on IBM Z to do AI fraud detection. Linux on IBM Z calls back to z/TPF to perform finalization. And lastly finalization sends a reply to the edge server which returns the reply to the customer.



# Credit authorization enterprise architecture



**Screen shot:** Instana – Call Diagram – Credit limit error

**Story:** Let's look at a request that is ending in over the credit limit error which has spiked.

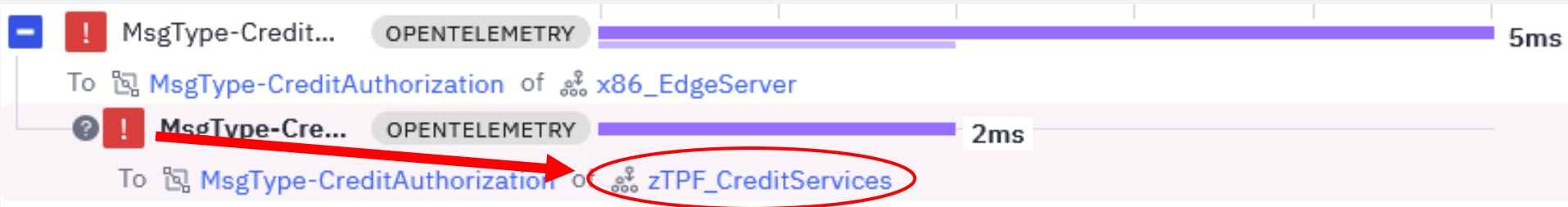
Immediately, we can see all sorts of interesting insights.

- The error is originating in the basic authorization processing and propagating up to the edge server.
- Fraud detection on Linux on IBM Z and the finalization processing on z/TPF were never called.

**Business value:** With your APM tools, you can inspect how error paths across your enterprise deviate from normal processing.

# Instana: z/TPF analyze call details

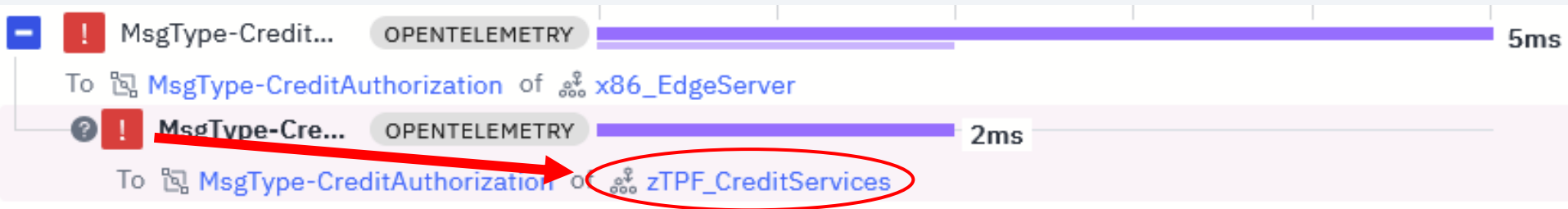
## Over the credit limit error originates on z/TPF



# Instana: z/TPF analyze call details

## Over the credit limit error originates on z/TPF

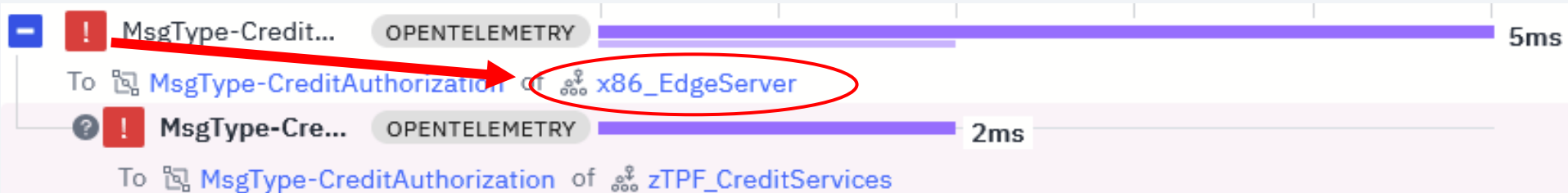
### In the basic authorization processing



MESSAGE_LIFETIME	1395
TPF_OTEL_ecb_purpose	BasicAuth
TPF_OTEL_rc	0X0000006A

# Instana: z/TPF analyze call details

## Over the credit limit error propagates to the edge server

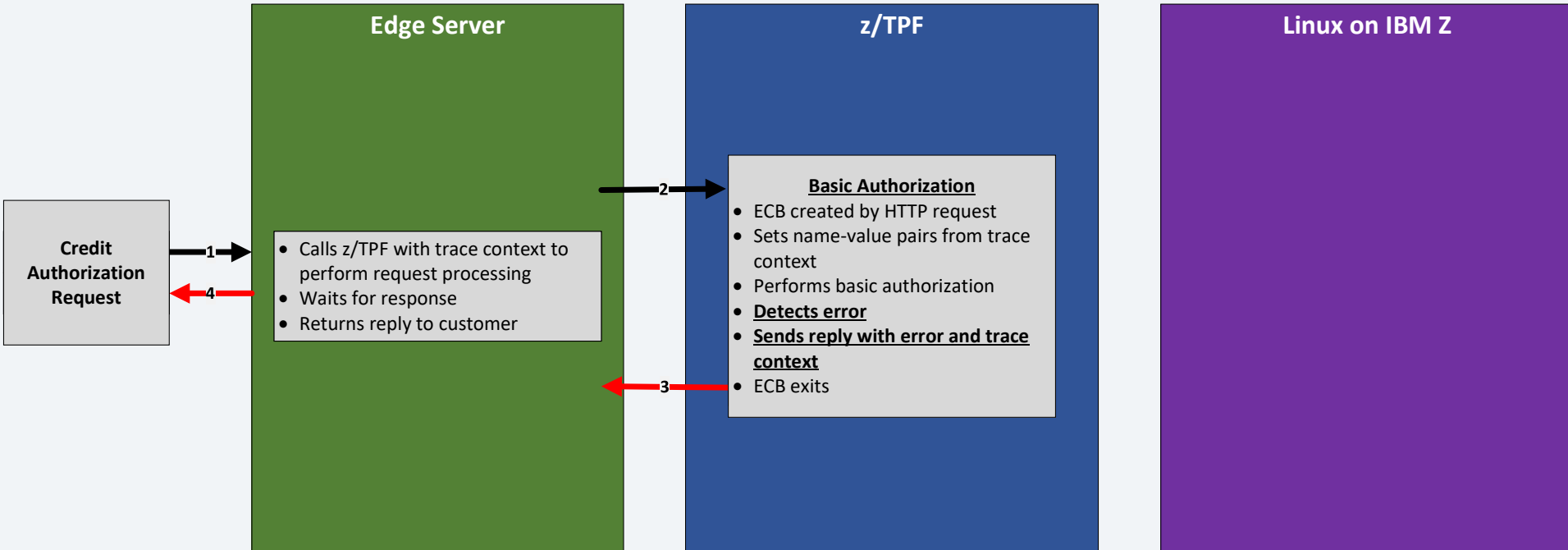


**Screen shot:** Enterprise Architecture

**Story:** We can quickly revisit our enterprise architecture diagram for this error case. This picture illustrates how the over the credit limit errors are originating from the basic authorization processing.

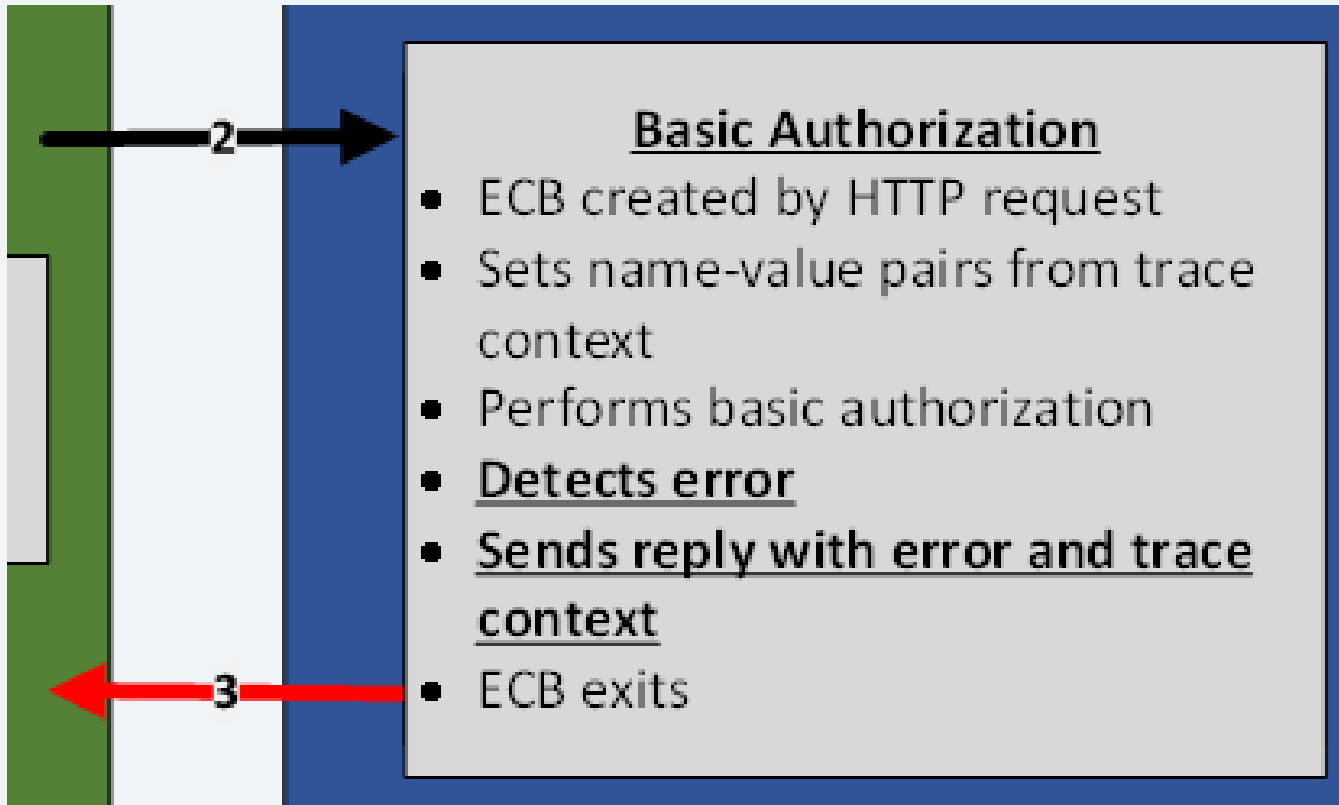
# Credit authorization enterprise architecture

## Over the credit limit error path



# Credit authorization enterprise architecture

## Over the credit limit error path





**Screen shot:** Instana: Call diagram – Credit limit error – Compare samples before and after

**Capabilities:** Your APM tool is continuously collecting samples. As such, you can do base line comparisons to the same situation before the inflection point.

**Story:** If we compare samples of over the credit limit errors before and after the inflection point, we can see that the samples after the inflection point are using more I/O. This might explain the increase in latency time we saw after our inflection point for credit authorization requests ending in over the credit limit errors.

**Business value:** With your APM tools, you can compare similar message processing paths.

# Comparison of metrics between two sample credit authorization messages with over the credit limit error

CPU_USED	545
FIND_DASD	1
EXIST_TIME	5135

Before  
inflection  
point

CPU_USED	1146
FIND_DASD	8
EXIST_TIME	3277

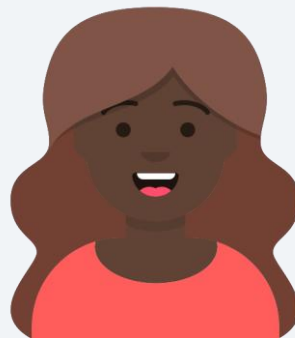
After  
inflection  
point

# From APM tool to RTMC

- The site reliability engineer needs help from a z/TPF expert to continue the investigation. (Or z/TPF coverage programmer switches from the APM tool to RTMC.)
- Notice that **not every silo** needed to be **scrambled** to investigate!
- Notice that our **SRE did not have to be a z/TPF expert** to make significant progress in the investigation!



Sarah  
site  
reliability  
engineer



Carol  
z/TPF  
coverage  
programmer

# From APM tool to RTMC

- The SRE provides the following information even though she has little knowledge of z/TPF or its processing!!!



Sarah  
site  
reliability  
engineer

- The enterprise and z/TPF systems are healthy.
- Credit authorization requests are ending in more errors, specifically over the credit limit errors.
- These errors are originating on z/TPF with greater frequency than previously seen.
- The problem started 5 minutes ago.
- The problem is originating in the basic authorization processing, early in transaction processing.
- The over limit errors are using more I/O operations than it previously did.



Carol  
z/TPF  
coverage  
programmer

# **RTMC**

## **Over the credit limit errors – Story 1 continued**

**Screen shot:** RTMC: System State dashboard

**Story:** Now we'll play the role of the z/TPF coverage programmer and use RTMC to dig deeper into the problem. The SRE gave us a ton of information to focus our investigation. That said, we'll start by doing our due diligence.

# RTMC: z/TPF insights for the coverage programmer



Carol  
z/TPF  
coverage  
programmer

## **Screen shot:** RTMC: System State dashboard

**Capabilities:** The system state dashboard shows you key metrics as to the health of your system like CPU usage, system level transformation engine (TE) and general purpose (GP) usage, in use ECBs, in use IOBs, and more. It also includes name-value pair estimated metrics like overall estimated message rate, CPU consumed per message, existence time per message and more.

Like all RTMC dashboards, the per second metrics are summarized to the minute boundary and include configurable pruning.

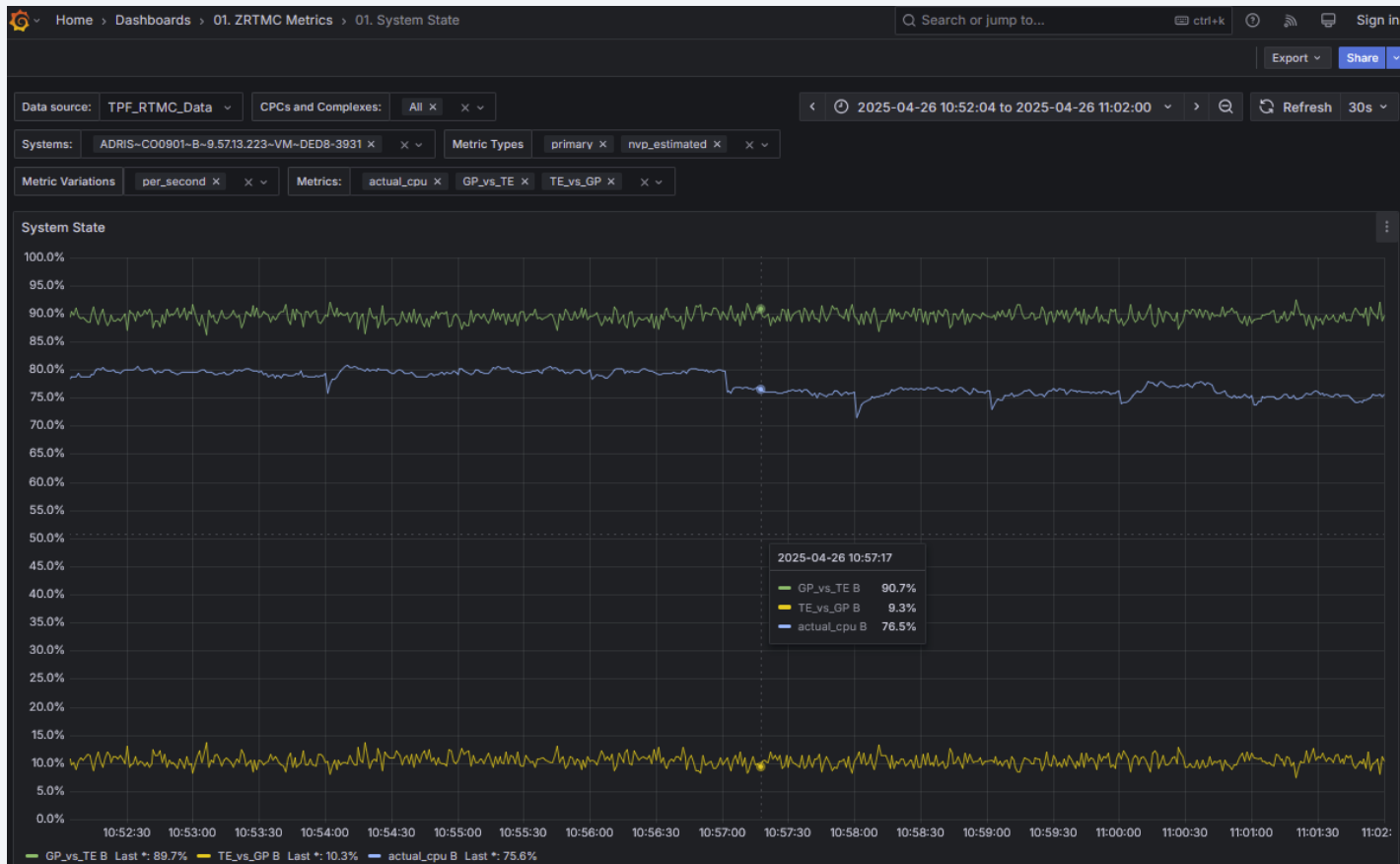
**Story:** As our SRE already determined, the system state is largely unchanged. The overall z/TPF system is healthy. So, we'll move on the name-value pair dashboards.

**Business value:** The RTMC system state dashboard provides key metrics to help you understand the health of your system.



# RTMC: System state dashboard

## Key system health metrics – nothing of interest



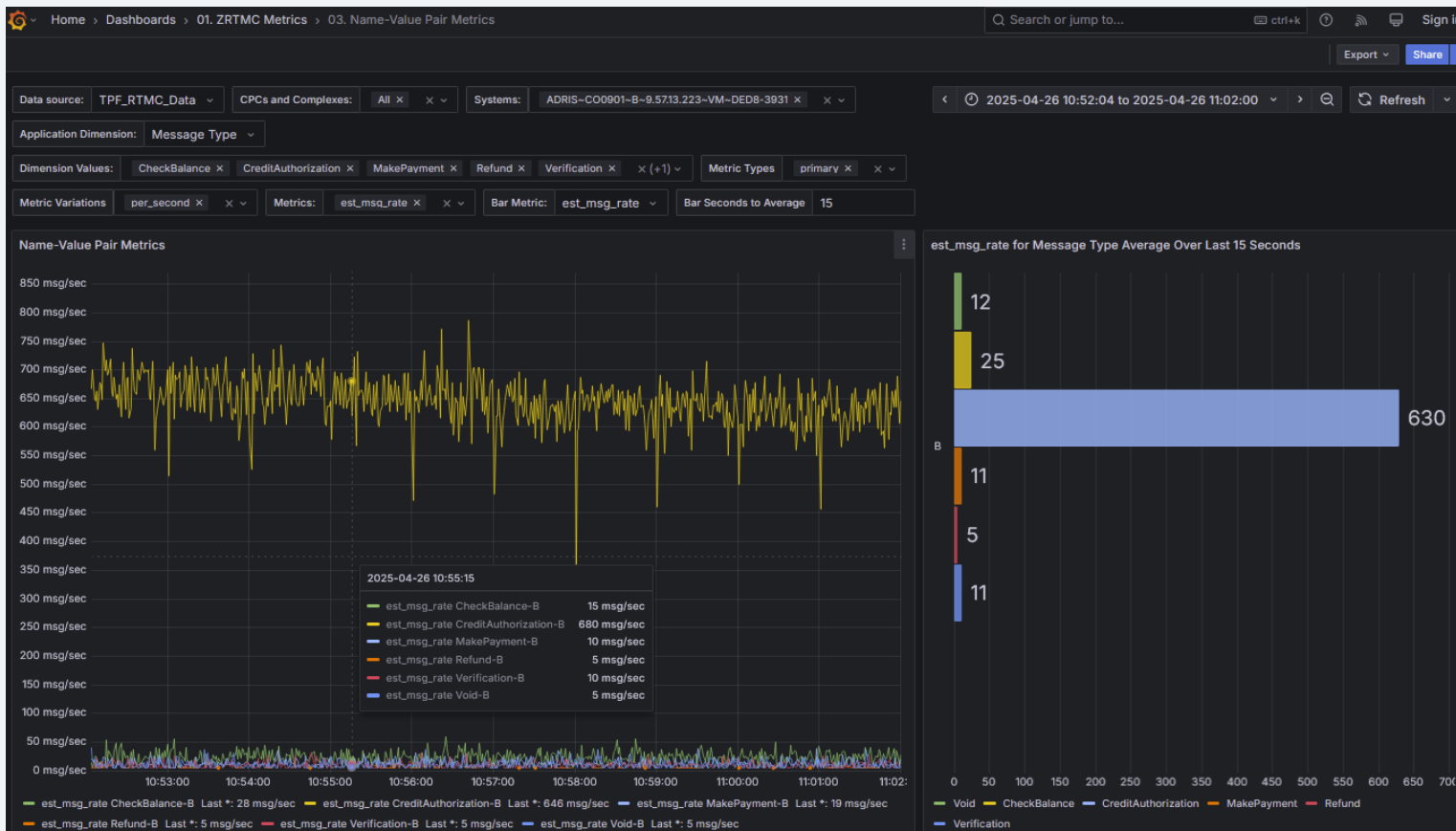
**Screen shot:** RTMC: Name-value pair metrics dashboard

**Capabilities:** With the name-value pair metrics dashboard, we can look at our messages, code packages and name-value pair combinations to understand how system resources are being used.

**Business value:** The RTMC name-value pair metrics dashboards provides resource usage metrics to help you understand your messages.

# RTMC: Name-value pair metrics dashboard

## Shows message and code package metrics



**Screen shot:** RTMC: Name-value pair metrics dashboard – Application dimensions

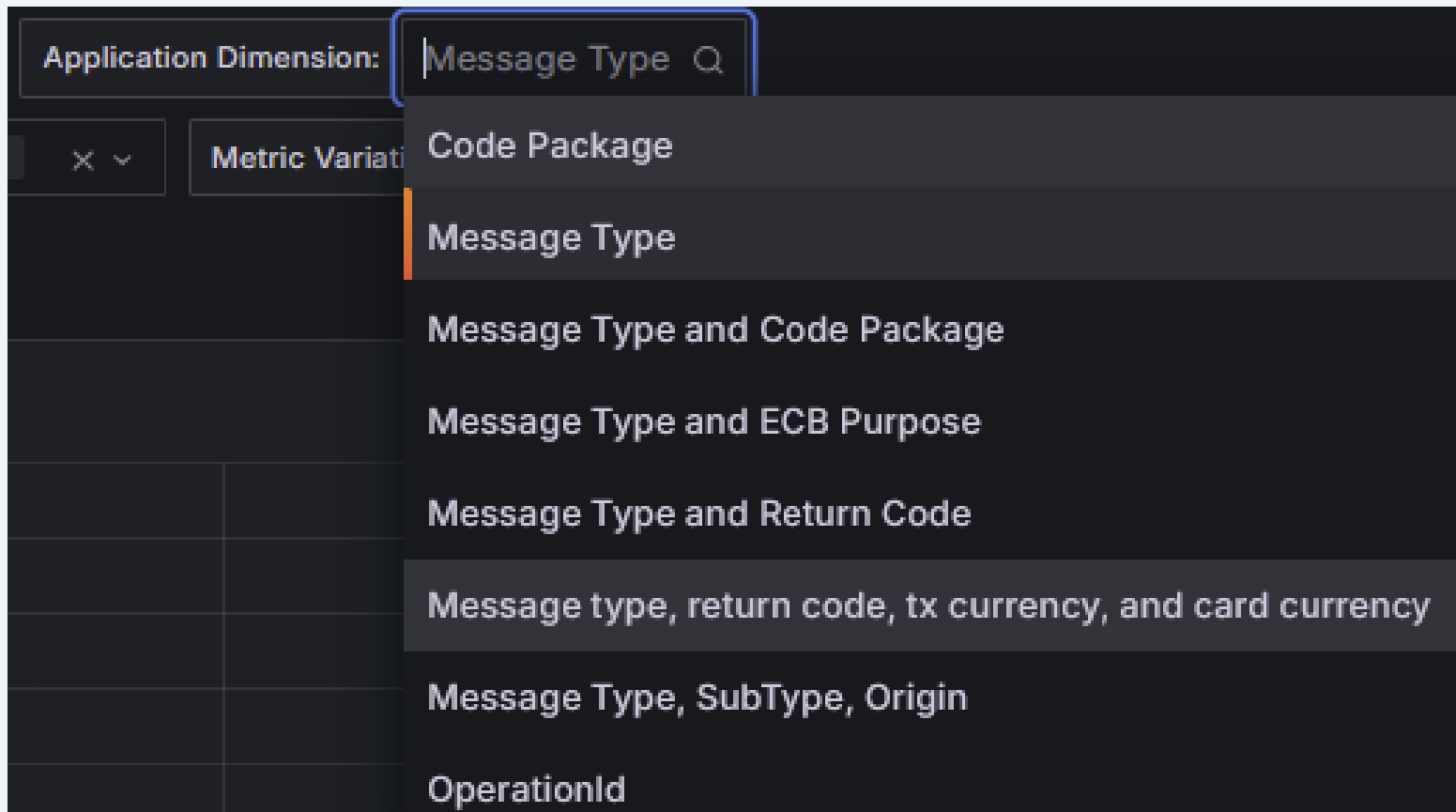
**Capabilities:** The application dimensions choose which combination of name-value pairs we are investigating. Different combinations provide all sorts of insights.

The application dimensions available are configurable in the ZRTMC analyzer for your name-value pair combinations.

**Business value:** With the RTMC name-value pair metrics application dimensions, you can understand your resource usage by your name-value pair definitions.

# RTMC: Name-value pair metrics dashboard

## Name-value pair combinations available



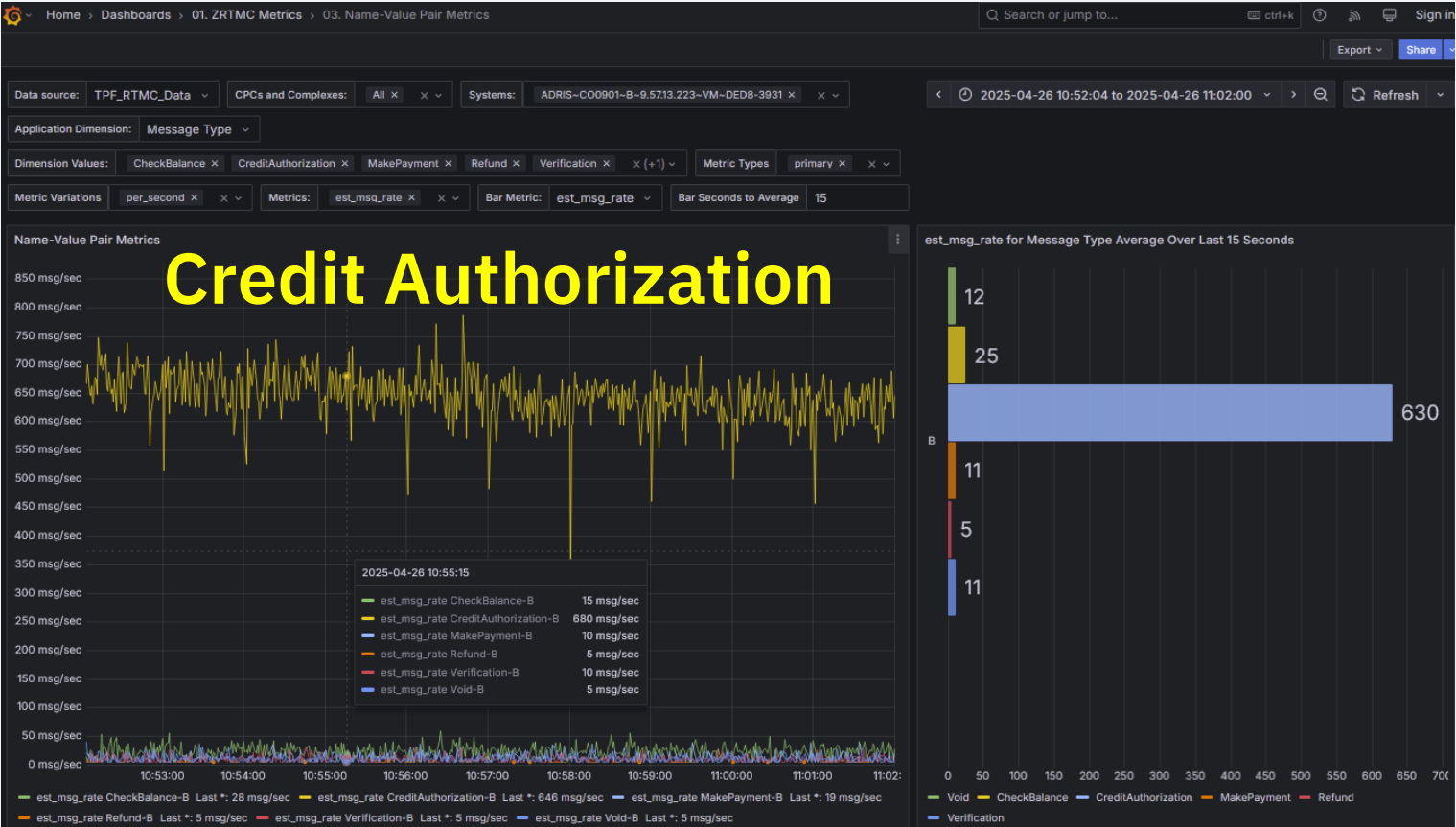
**Screen shot:** RTMC: Name-value pair metrics dashboard – Message type

**Story:** Again, following the saying “trust but verify”, we look at the MsgType application dimension to verify that only Credit Authorization messages are affected. We can see that all message types estimated message rates are holding steady.

**Business value:** The RTMC name-value pair metrics dashboards provides resource usage metrics to help you understand your messages.

# RTMC: Name-value pair metrics dashboard

## Estimated message rate of all message types is unchanged



## **Screen shot:** RTMC: Name-value pair metrics dashboard – Message type

**Story:** Next, we can filter on the credit authorization name-value pair value. We can barely see the inflection point where:

- CPU used and exist time decreases a bit for credit auth messages
- IOs rise a smidge

The impact is not dramatic, it's difficult to identify. We can also see that resources used by other message types are unchanged.

We'll look at other name-value pair combinations for credit authorization requests to dig deeper into our results. But this confirms what our SRE found that the problem is isolated to credit authorization requests.

**Business value:** The RTMC name-value pair metrics dashboards provides resource usage metrics to help you understand your messages.



# RTMC: Name-value pair metrics dashboard

## Trivial changes in metrics for credit authorization messages



**Screen shot:** RTMC: Name-value pair metrics dashboard – RC msg – Message rate

**Capabilities:** As part of the z/TPF support for OpenTelemetry, we introduced a name-value pair convention for trace IDs, span IDs and so on. We also introduced conventions for the return code and return code message. We include an application dimension by default for Message Type and Return Code.

**Story:** Filtering on credit authorization requests and showing all possible return codes, we can see the estimated message rate for success falls, over the credit limit errors rise, and all other return codes are holding constant.

**Business value:** The RTMC name-value pair metrics dashboards provides estimated message rate metrics to help you understand your message mix and how it changes over time.

# RTMC: Name-value pair metrics dashboard

Filtered on credit authorization message type showing estimated message rate for all error types

Success rate falls while over the credit limit errors rise



**Screen shot:** RTMC: Name-value pair metrics dashboard – RC – Other metrics

**Capabilities:** Name-value pair collection includes a host of other metrics we can inspect for our name-value pair combinations such as memory usage, z/TPFDF usage, copy-on-write usage and more. If more DF APIs were issued, maybe we can go to the system metrics dashboard and see if a particular z/TPFDF database is being accessed more frequently.

**Story:** Remember that the SRE determined credit authorization messages ending in over the credit limit were using more I/O operations? We've switched to look at FIND operations as well as finds for prime file address operations issued by z/TPFDF. We can see more FIND operations are used than before the inflection point but the z/TPFDF PFIND operations hold steady. This is an indication of what type of I/O operations the application is doing.

**Business value:** The RTMC name-value pair metrics dashboards provides resource usage metrics to help you understand your messages.

# RTMC: Name-value pair metrics dashboard

Filtered on credit authorization messages with over the credit limit errors

Access (FIND) to traditional z/TPF databases increased but access to z/TPFDF databases is unchanged



**Screen shot:** RTMC: Name-value pair metrics dashboard – Code package

**Capabilities:** Name-value pair collection type-sample includes vertical name-value pair results like owner name change as code package. You can use this feature to understand the resources used by different code packages in your processing.

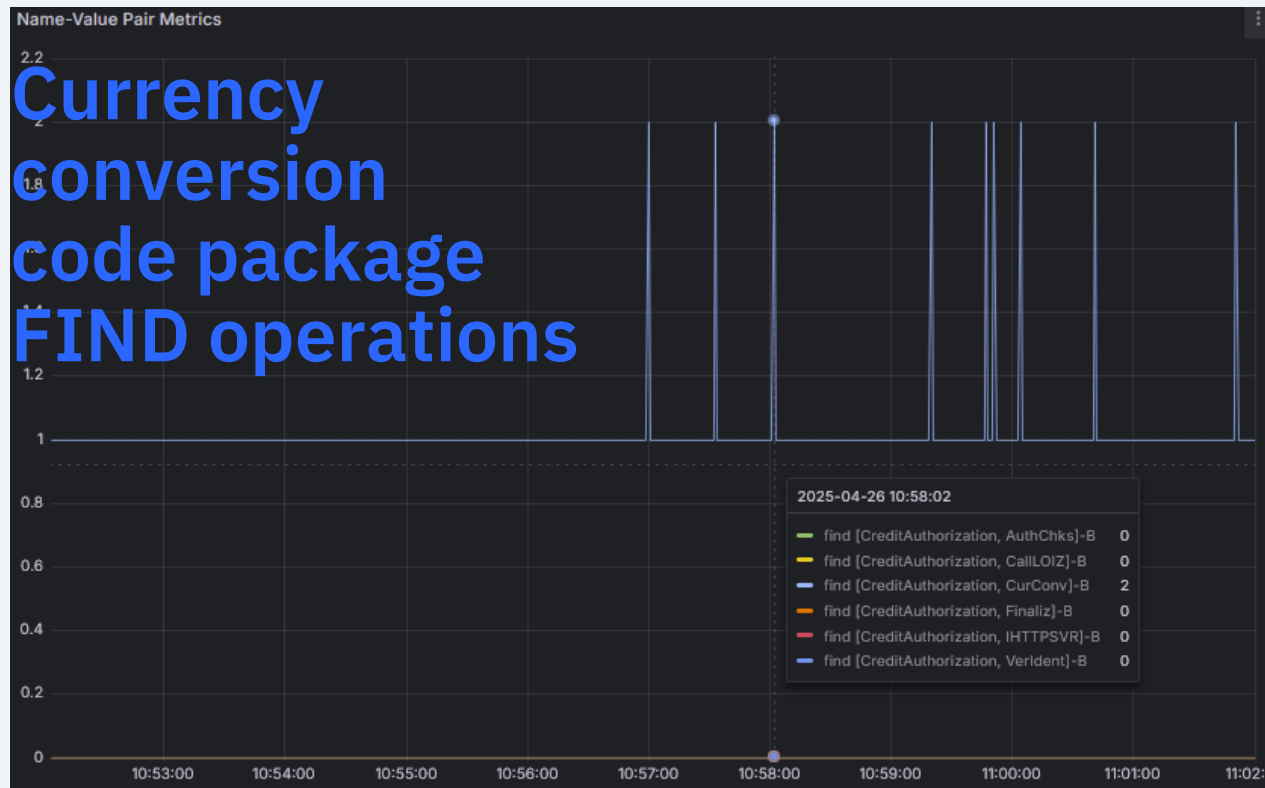
**Story:** In this case, we'll focus on the message type and code package so we can filter on the code packages used by credit authorization messages. Here we can see that the currency conversion code package is the one doing more I/O operations after our inflection point. This insight helps us to know who to contact next, the owner of the currency conversion code package.

**Business value:** The RTMC name-value pair metrics dashboards provides resource usage metrics to help you understand your messages.

# RTMC: Name-value pair metrics dashboard

Filtered on all code packages used by credit authorization messages

FIND operations rise for the currency conversion code package



**Screen shot:** RTMC: Name-value pair metrics dashboard – Message type, return code message, local currency and transaction currency

**Capability:** With the name-value pair metrics, you can define and inspect a variety of different name value pair combinations for deeper insights.

**Story:** Switching to the application dimension for message type, return code message, local currency, and transaction currency. We'll filter on our message type credit authorization and return code message for over the credit limit error. Now we can see name-value pair metrics for the various combinations of transaction and local currency. Notice that when the transaction currency is Euros and the local currency is Dollars, we see the rise in message rate while the others like Dollars and Yen remain flat.

This is one example of how you can use combinations of name-value pairs to see other insights.

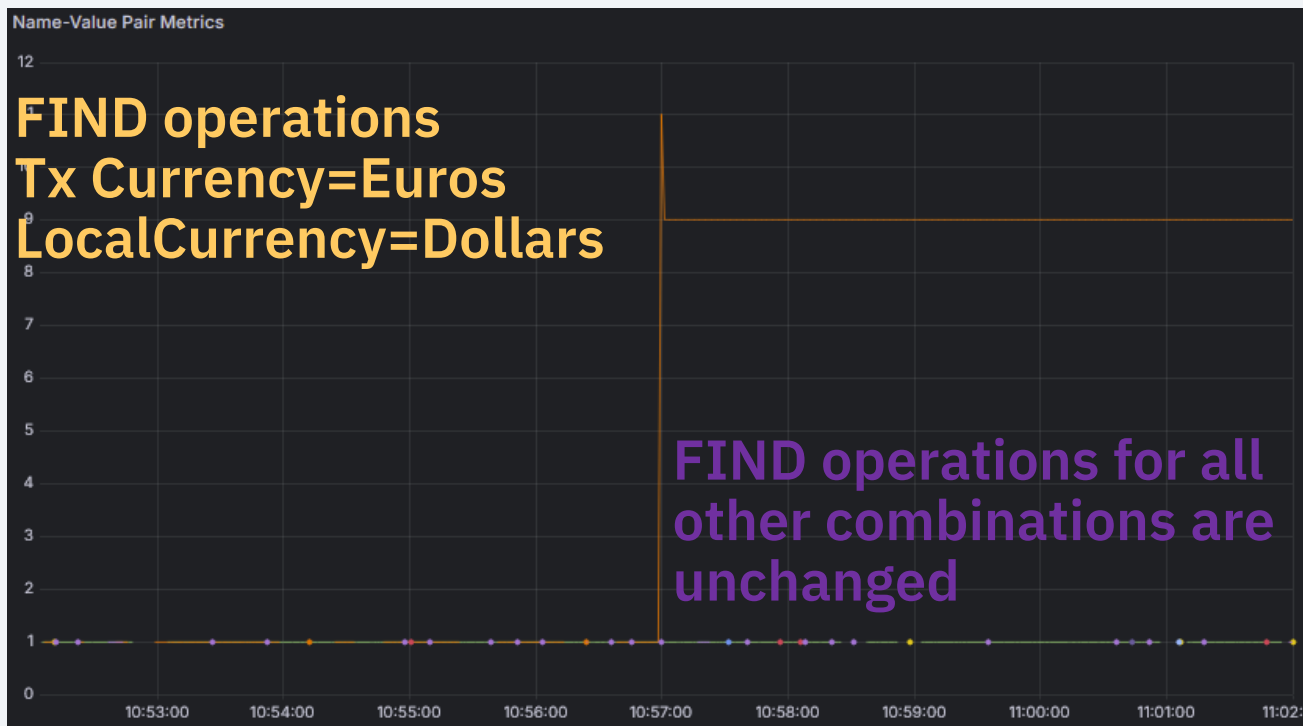
**Business value:** The RTMC name-value pair metrics dashboards provide resource usage metrics to help you understand your messages.



# RTMC: Name-value pair metrics dashboard

Filtered on credit authorization messages with over the credit limit errors  
and showing all combinations of transaction and local currency

FIND operations only rise when transaction currency is in Euros and local  
currency is in Dollars



# From RTMC to code package owner

- The z/TPF coverage programmer got a quick start from the information determined by the SRE in the APM tool.
- Using RTMC, they determined additional key insights.
- They ask the owner of the offending code package for additional insights.



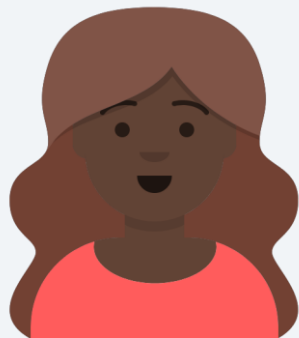
Carol  
z/TPF  
coverage  
programmer



Zach  
application  
developer

# From RTMC to code package owner

- In addition to the information discovered by the SRE, the coverage programmer provides the following details to the code package owner:
  - Most metrics are unchanged for credit authorization messages ending in the over the credit limit error when compared to messages with the same type of error before the change in error frequency occurred.
  - However, additional I/O operations are occurring. But these are not for z/TPFDF operations, they are for traditional FIND operations. The additional I/O operations are originating from the currency conversion code package.
  - The additional errors occur when the transaction currency is Euros and the card holder currency is Dollars.



Carol  
z/TPF  
coverage  
programmer



Zach  
application  
developer

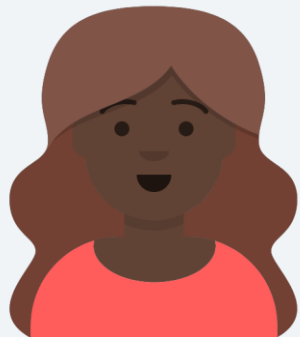
# Code package owner investigation

- The currency conversion code package has not changed recently.
- The application developer indicates that the currency conversion table in the z/TPFDF database is updated twice a day. In fact, one of the updates just happened 5 minutes ago. But this happens twice every day, why would this cause an issue?
- The application developer is intrigued by the additional FIND operations as the currency conversion tables were moved entirely into a z/TPFDF database a few years ago.
- The application developer reads through the code and remembers there's an old migration path for when they migrated the currency conversion tables to z/TPFDF. If the currency type is not in the z/TPFDF database, then the code falls back to the old currency conversion tables in a traditional z/TPF database.



Zach  
application  
developer

# Code package owner investigation



Carol  
z/TPF  
coverage  
programmer

- The application developer asks the coverage programmer to confirm that Euros are not in the z/TPFDF currency conversion table.
- The coverage programmer issues a command to display the currency conversion tables:

```
ZCURR EURO
```

```
ERROR: CURRENCY "EURO" NOT FOUND
```

- The application developer guides the coverage programmer using ZDFIL to explore the traditional database and finds that the out-of-date EURO conversion rate being used is 1.58 instead of 1.08.
- The use of the traditional z/TPF database explains the additional I/O operations.



Zach  
application  
developer

# Over the credit limit story conclusion



Carol  
z/TPF  
coverage  
programmer

- The currency conversion table is reloaded with the correct currency conversion for Euros.
- The application developer is going to follow up on removing the old currency conversion migration path and implementing better error handling for missing currency conversion entries in the z/TPFDF database.



Zach  
application  
developer

# Over the credit limit story conclusion

- As the SRE, we were able to **quickly determine the error was originating** on z/TPF and provide the coverage programmer with **numerous insights, despite limited knowledge of z/TPF**.
- As the coverage programmer, we were able to use RTMC to further understand the problem and resolve the issue quickly.



Sarah  
site  
reliability  
engineer



Carol  
z/TPF  
coverage  
programmer



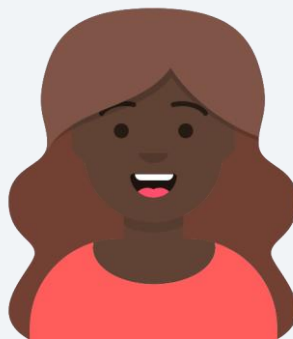
Zach  
application  
developer

# Over the credit limit story conclusion

- **Your APM tool** and z/TPF support for OpenTelemetry **helps you to engage the right silo** quickly without scrambling all of the troops.
- Your z/TPF experts can use **RTMC** to get to the **root of problems that originate on z/TPF**.



Sarah  
site  
reliability  
engineer



Carol  
z/TPF  
coverage  
programmer



Zach  
application  
developer



# **z/TPF message analysis tool – Story 1 alternate ending**

**What if the insights provided by  
the APM tool and RTMC were not  
enough to debug the problem?**

# z/TPF message analysis tool

- Why are Credit authorization messages ending in over the credit limit errors performing additional I/O operations?
- The coverage programmer knows:
  - Additional I/O operations are occurring. But these are not for z/TPFDF operations, they are for traditional FIND operations.
  - The additional I/O operations are originating from the currency conversion code package.
  - The additional errors occur when the transaction currency is Euros and the card holder currency is Dollars.



Carol  
z/TPF  
coverage  
programmer

# RTMC: Name-value pair metrics dashboard

Filtered on credit authorization messages with over the credit limit errors

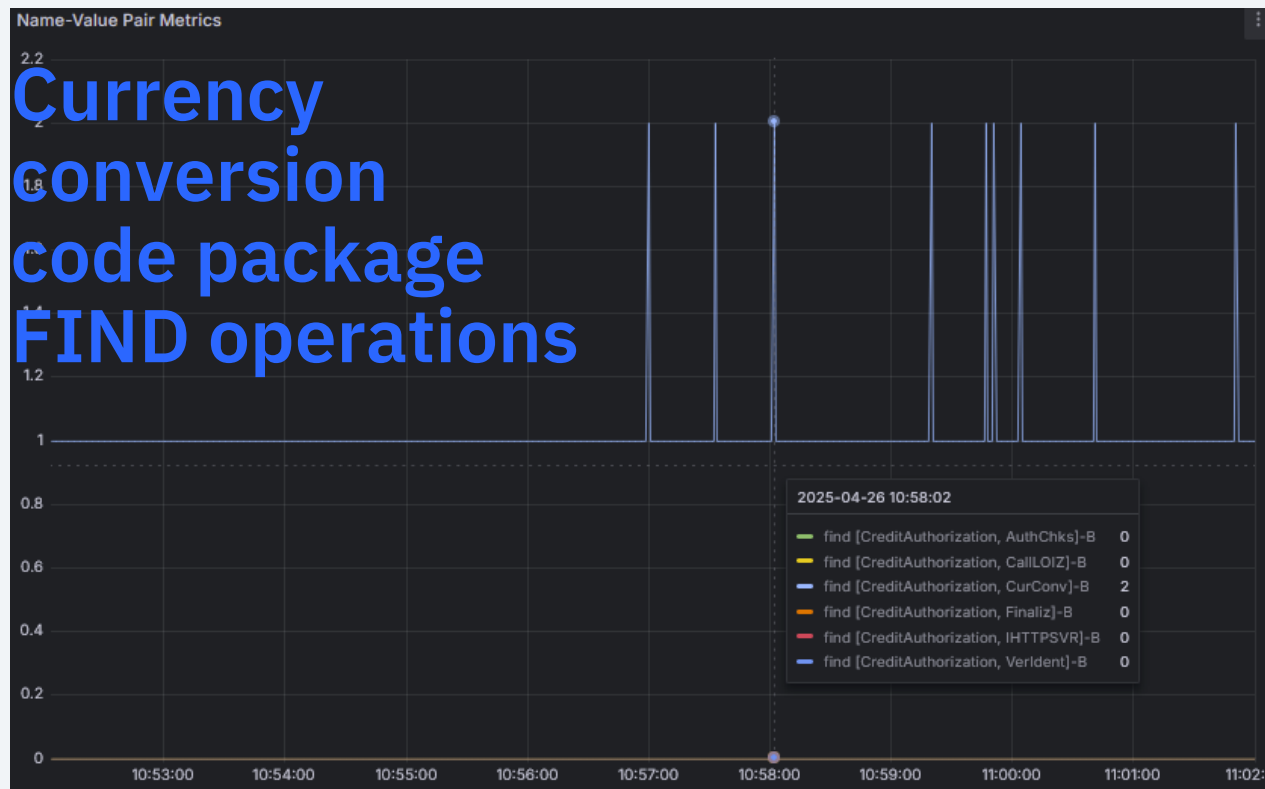
Access (FIND) to traditional z/TPF databases increased but access to z/TPFDF databases is unchanged



# RTMC: Name-value pair metrics dashboard

Filtered on all code packages used by credit authorization messages

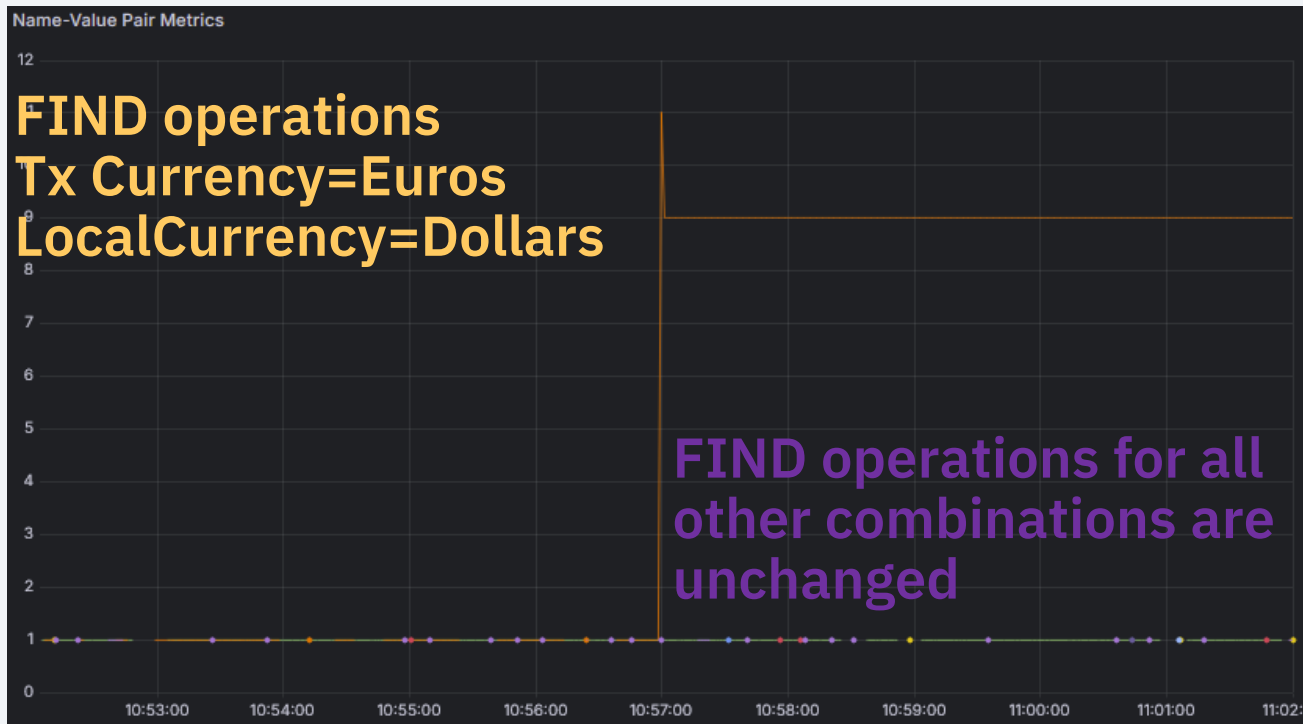
FIND operations rise for the currency conversion code package



# RTMC: Name-value pair metrics dashboard

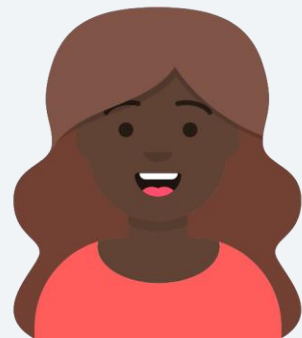
Filtered on credit authorization messages with over the credit limit errors  
and showing all combinations of transaction and local currency

FIND operations only rise when transaction currency is in Euros and local  
currency is in Dollars



# z/TPF message analysis tool

- We can use the z/TPF message analysis tool to determine where those additional I/O operations are originating in the currency conversion code package.
- You can use the z/TPF message analysis tool to dig deeper into messages at the function and macro level.
- When registering for the z/TPF message analysis tool, you can include a single name-value pair value. As such, we will register to capture messages in the production environment that call the currency conversion package when transaction currency is Euros.



Carol  
z/TPF  
coverage  
programmer

**Screen shot:** RTMC: z/TPF message analysis tool – Collection dashboard

**Capabilities:** The z/TPF message analysis tool collection dashboard shows you all the collections in your database for the time frame selected in the Grafana time picker. You can select a collection to see the messages captured for that collection. You can see a summary of key metrics used.







**Story:** We select a message with lower call count, CPU used, or existence time knowing this aligns with an error originating from the basic authorization ECB.

**Business value:** With the z/TPF message analysis tool, you can capture multiple instances of a message from production and see summaries of the resources used to know if you should investigate further.













# RTMC: z/TPF message analysis tool collection dashboard

## Shows list of collections available (top table)

### Message Analysis Collections

TARGET	DESCRIPTION	TIME	COLLECTION_ID	SUMMARY_STATUS	ANALYSIS	COMPLETE	OPT_LEVEL
	Capture CreditAuthorizati	2025-04-28 20:12:58	CO0901__B_00E0CEFF2	Done			O3
	Capture CreditAuthorizati	2025-04-28 20:10:58	CO0901__B_00E0CEFE8	Done			O3

### Target Collection Details (CO0901\_\_B\_00E0CEFF27B872AD)

TARGET	UOWID	NVPS	SUMMARY_STATUS	ANALYSIS	OPT_LEVEL	CALL_COUNT	CPU_EXIST	CPU_USED
	C2C3D6F0F9F0F14040	null	Done		O3	7226	26101860	6352636
	C2C3D6F0F9F0F14040	null	Done		O3	225	143523	143523
	C2C3D6F0F9F0F14040	null	Done		O3	225	142117	142117
	C2C3D6F0F9F0F14040	null	Done		O3	7764	28851745	6739382
	C2C3D6F0F9F0F14040	null	Done		O3	225	139387	139387
	C2C3D6F0F9F0F14040	null	Done		O3	8122	26856363	7739907















# RTMC: z/TPF message analysis tool collection dashboard

## Select a collection to see list of messages

TARGET	DESCRIPTION
<input checked="" type="checkbox"/>	Capture CreditAuthorization
<input type="checkbox"/>	Capture CreditAuthorization

# RTMC: z/TPF message analysis tool collection dashboard

Shows list of messages you can investigate

TARGET	UOWID	NVPS	SUMMARY_STATUS	ANALYSIS
	C2C3D6F0F9F0F14040	null	Done	
	C2C3D6F0F9F0F14040	null	Done	
	C2C3D6F0F9F0F14040	null	Done	
	C2C3D6F0F9F0F14040	null	Done	
	C2C3D6F0F9F0F14040	null	Done	
	C2C3D6F0F9F0F14040	null	Done	

**Screen shot:** RTMC: z/TPF message analysis tool – Summary details dashboard

**Capabilities:** From the collection dashboard, we selected a message and chose to view the summary details dashboard. The summary details dashboard shows all of the functions and macros calls, the number of times it was called, and the system resources consumed by that call such as CPU used. You can filter by ECB, module, macro and more. You can sort by the different metric columns

**Story:** In our over the credit limit error scenario, we are viewing our results by macro and can see the FINWC macro was called 15 times.

**Business value:** With the z/TPF message analysis tool, you can see a summary of the resources used by a message at the function and macro level.

# RTMC: z/TPF message analysis tool summary details dashboard

## Macro display shows FINWC called 15 times

MACRO_NAME ↓	CALL_COUNT	PCT_CALL_COUNT (%)	CPU_EXIST
FSYSC	5	<div></div>	0
FINWC	15	<div></div>	0
EXITC	2	<div></div>	0
EVNWC	2	<div></div>	0
EVNTC	2	<div></div>	0
EOWNRC	6	<div></div>	0
ENTRC->CXXC	1	<div></div> 0.0138	158578
ENTNC->QOWN	1	<div></div> 0.0138	1846491
ENTDC->QBSO	2	<div></div> 0.0277	31867

**Screen shot:** RTMC: z/TPF message analysis tool – Summary details dashboard comparison.

**Capabilities:** You can periodically capture messages from production to have a catalog of baseline z/TPF message analysis tool runs. With these historical collections, you can do a before and after comparison to see how the usage of system resources, macros and function calls has changed.

**Story:** We compare the before and after to see how the FINWC usage has changed.

**Business value:** With the z/TPF message analysis tool, you can see a summary of the resources used by a message at the function and macro level.

# RTMC: z/TPF message analysis tool summary details dashboard

Comparing over the credit limit before and after the inflection point we can see that the FINWC macro is called more after the inflection point

## Before

MACRO_NAME ↓	CALL_COUNT
FSYSC	5
FINWC	1
EXITC	2
EVNWC	2
EVNTC	2
EOWNRC	6
ENTRC->CXXC	1
ENTNC->QOWN	1
ENTDC->QBSO	2

## After

MACRO_NAME ↓	CALL_COUNT
FSYSC	5
FINWC	15
EXITC	2
EVNWC	2
EVNTC	2
EOWNRC	6
ENTRC->CXXC	1
ENTNC->QOWN	1
ENTDC->QBSO	2

**Screen shot:** RTMC: z/TPF message analysis tool – Application call path

**Capabilities:** You can query the database to inspect the application call path to understand execution of the application code.

**Story:** We read through the call path comparing it to the baseline to understand where the processing deviates. In the over the credit limit error scenario, we could trace the path through the currency conversion code package to see how the old migration code was being called.

**Business value:** With the z/TPF message analysis tool, you can see sequence of function and macro calls to understand the execution of the application code.

# RTMC: z/TPF message analysis tool

Application flow in SQL query results in database

FINWC issued by program UCCN for record ID FC11

NESTING_LEVEL	TRACE_NAME	MACRO_NAME	MACRO_DATA
16	UCCN	DETAC	(NULL)
16	UCCN	FINWC	FILE_ADDR-00000000186BF04F,DECB_ADDRESS-10C96120,REC_ID-FC11,RESIDENCY-DASD
16	UCCN	ATTAC	(NULL)
16	UCCN	RELCC	ADDRESS-10C98000,BLOCK_TYPE-0051,DATA_LEVEL-DF
16	UCCN	DETAC	(NULL)
16	UCCN	FINWC	FILE_ADDR-000000001868070B,DECB_ADDRESS-10C96120,REC_ID-FC11,RESIDENCY-DASD
16	UCCN	EHEAPC	ACTION-MALOC,HEAP_ADDR-000000001B08CF08,HEAP_SIZE-00000000000000F0
16	UCCN	ATTAC	(NULL)
16	UCCN	RELCC	ADDRESS-10C98000,BLOCK_TYPE-0051,DATA_LEVEL-DF
16	UCCN	DETAC	(NULL)
16	UCCN	EHEAPC	ACTION-MALOC,HEAP_ADDR-000000001B08A748,HEAP_SIZE-000000000000008B0
16	UCCN	FINWC	FILE_ADDR-000000001868070A,DECB_ADDRESS-10C96120,REC_ID-FC15,RESIDENCY-DASD



# z/TPF message analysis tool – conclusion

- The coverage programmer found where the additional FIND operations were occurring in the currency conversion code package: module UCCN.
- They also saw the record ID was FC11. They look up this record ID and find that it is for an old currency conversion table in a traditional z/TPF database.
- They also saw the sequence of function and macro calls leading to the additional FIND macros.
- Reading that code, they can see this is a migration code path used when a currency does not exist in the z/TPFDF database.



Carol  
z/TPF  
coverage  
programmer

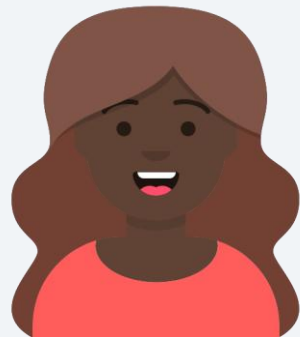
# z/TPF message analysis tool – conclusion

- The coverage programmer issues a command to display the currency conversion tables:

```
ZCURR EURO
```

```
ERROR: CURRENCY "EURO" NOT FOUND
```

- The currency conversion table is reloaded with the correct currency conversion for Euros and the over the credit limit errors return to their previous levels.



Carol  
z/TPF  
coverage  
programmer

# **Application performance monitor tooling – Story 2**

## **AI fraud detection errors – errors propagating through the entire enterprise**

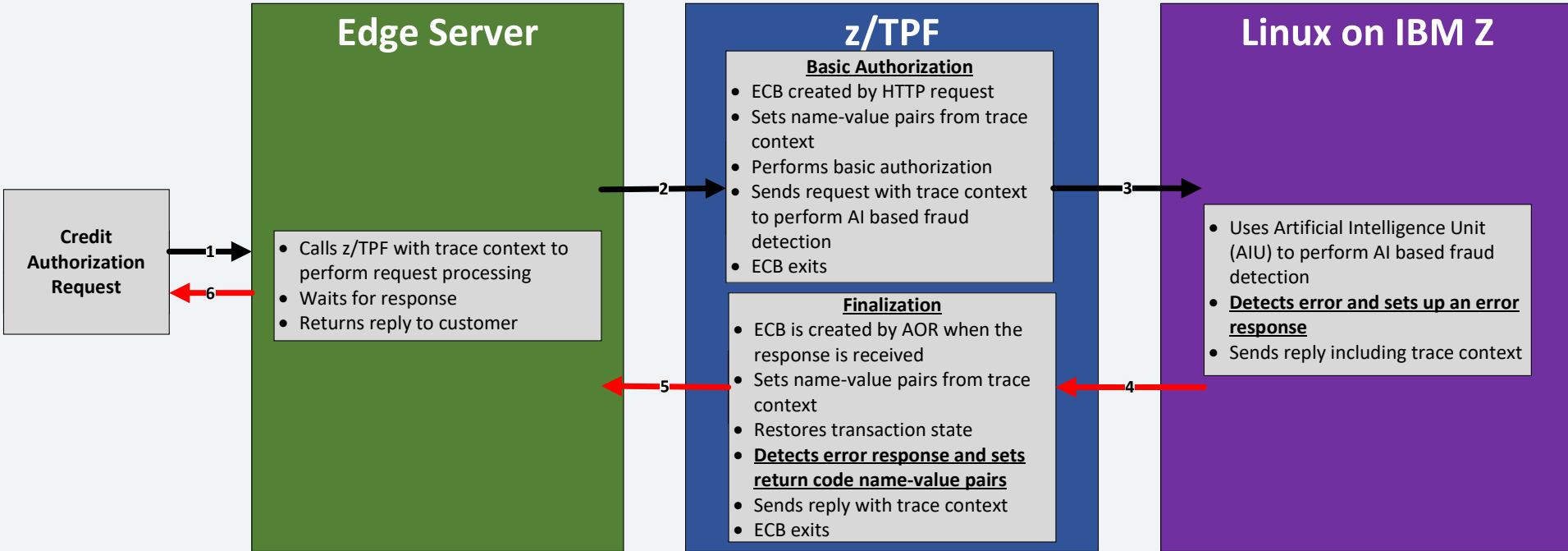
# AI fraud detection errors

## Errors propagating through the entire enterprise

- In this scenario, the error originates in the AI fraud detection in our Linux on IBM Z system such that the error propagates all the way back through the entire enterprise to the edge server.
- First, let's refresh ourselves again on the enterprise architecture.

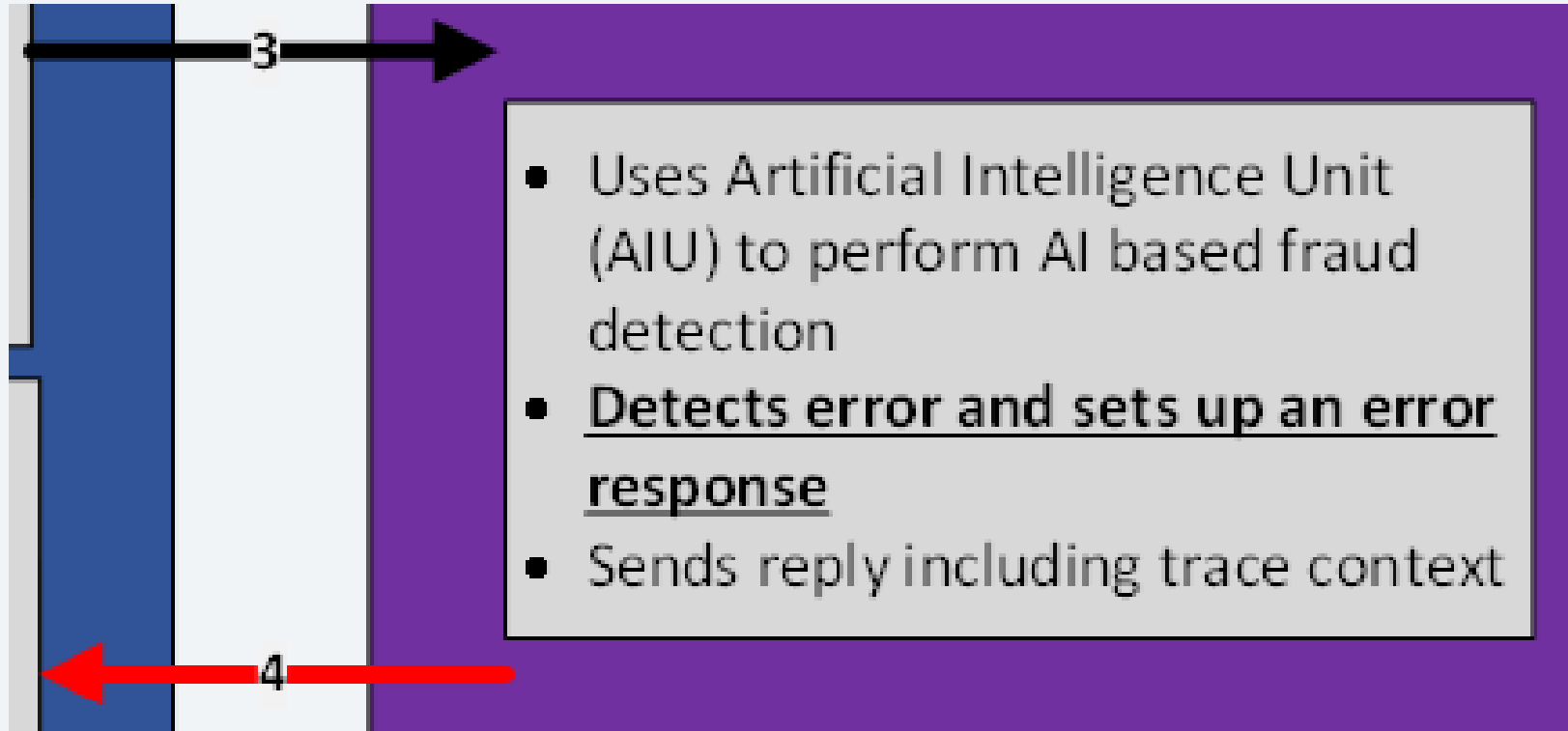
# Credit authorization enterprise architecture

## AI fraud detection error path



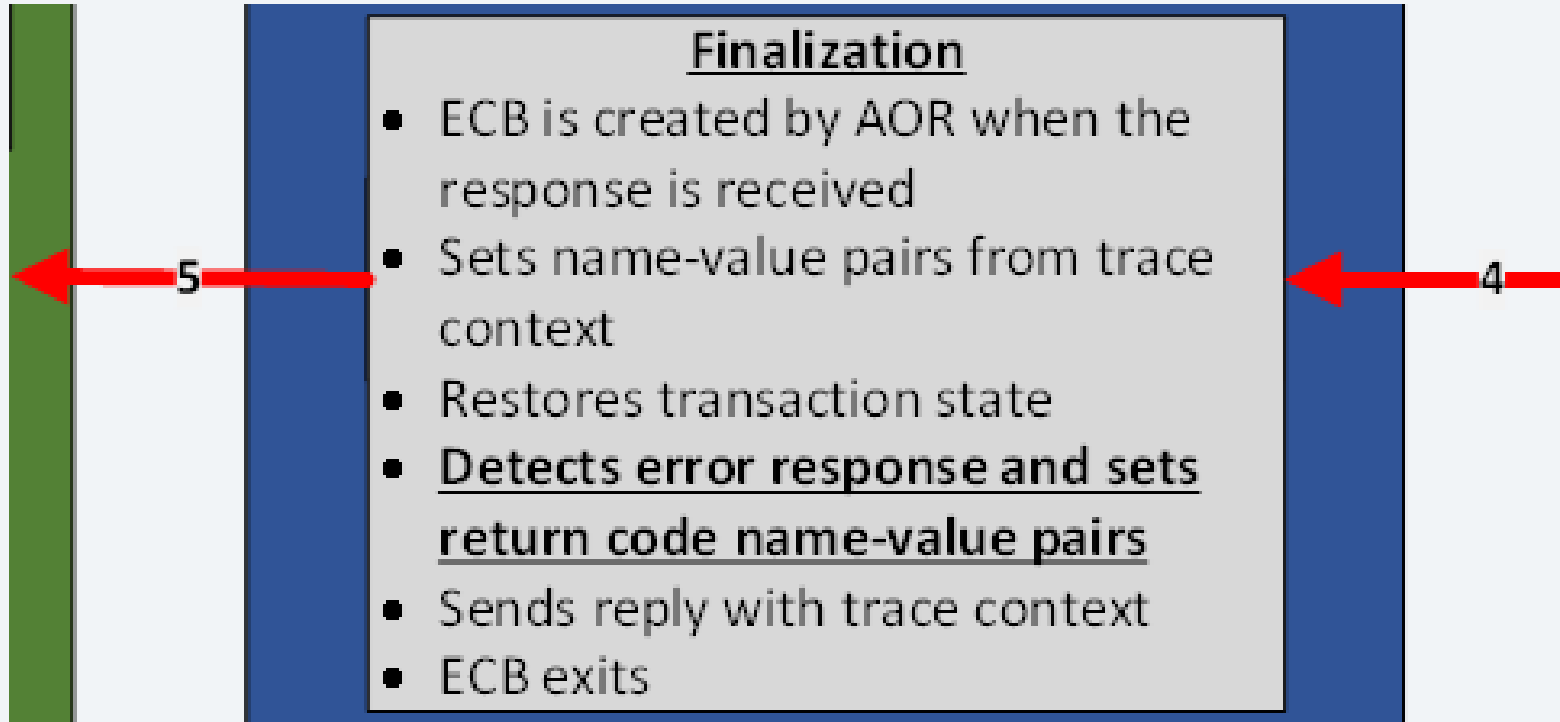
# Credit authorization enterprise architecture

## AI fraud detection error path



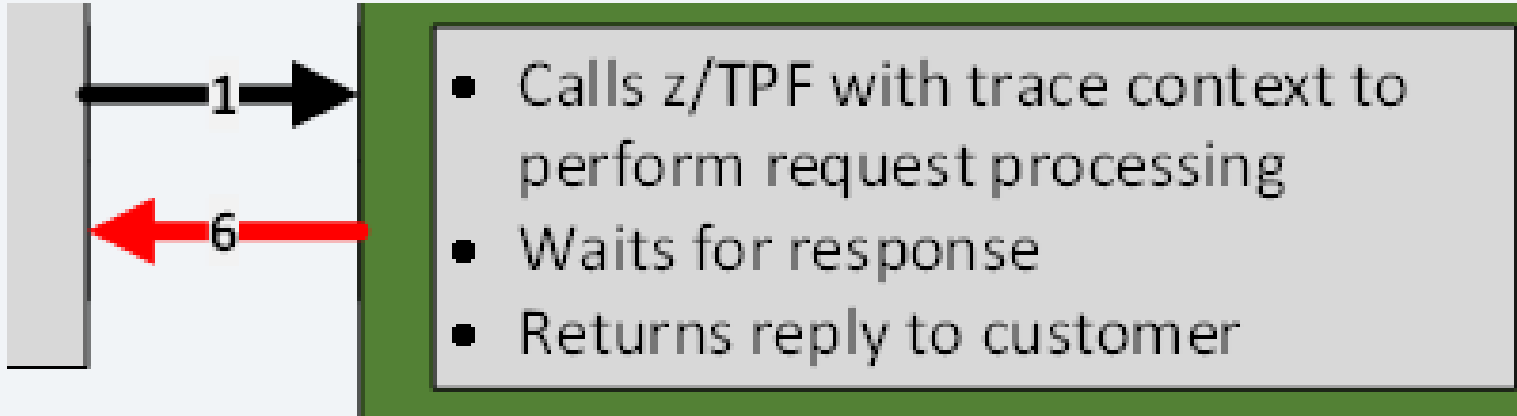
# Credit authorization enterprise architecture

## AI fraud detection error path



# Credit authorization enterprise architecture

## AI fraud detection error path





**Screen shot:** Instana: Alert

**Story:** As before, we receive an Alert. There's an issue in our enterprise.

This alert shows that the credit authorization error rate has broken the 5% error rate threshold. Glancing at this, it looks like the increase in errors is originating on Linux on IBM Z and has something to do with AI fraud detection errors.

**Business value:** APM tools can alert you to conditions in your enterprise and provide insights as to which system to investigate further, possibly allowing you to remediate issues before SLAs are impacted.

# Instana: Alert!

**EventId:**

PDsh8nkDTuef2KgbmhA3Eg

**Link:**

<https://instana.fake.com/#/events:eventID=PDsh8nkDTuef2KgbmhA3Eg&incidentTo=16345989153>

**Incident started with:**

Credit Authorization message error rate exceeds 5% threshold.  
These violations are occurring continuously since 12:12:10 EST 4/26/2025.  
SLAs may be violated within the next 10 minutes.

**Detail:**

The increase is Credit Authorization message error rate is originating on Linux on IBM Z.  
Specifically, Al fraud detection errors are occurring more frequently while other error types are occurring at a stable rate.

**Severity:** Critical



Sarah  
site  
reliability  
engineer

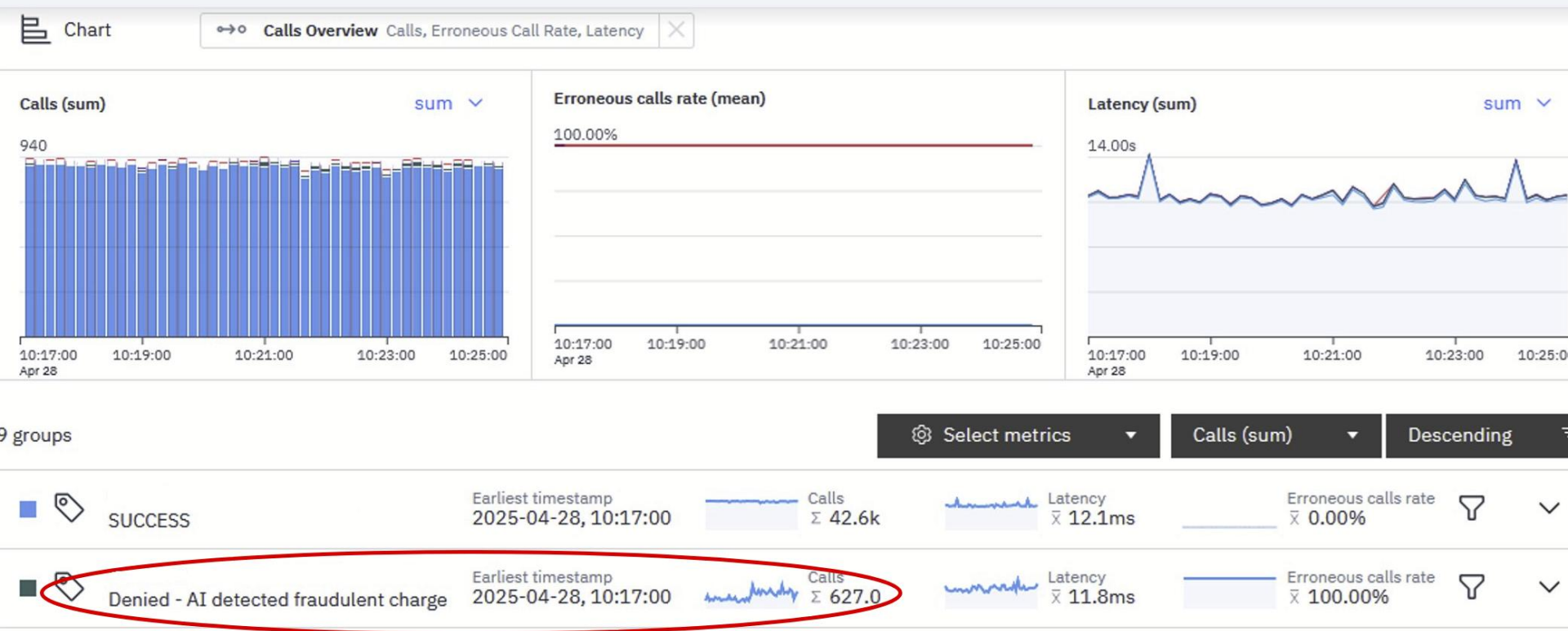
**Screen shot:** Instana: Analyze calls – Graphs – return code breakdown

**Story:** As before, we can confirm that the over all system state is unchanged. And we can see that the AI fraud detection is the only error rate that has changed.

**Business value:** APM tools provide filter and breakdown by message attributes so you can glean insights into problematic messages.

# Instana: Linux on IBM Z analyze calls

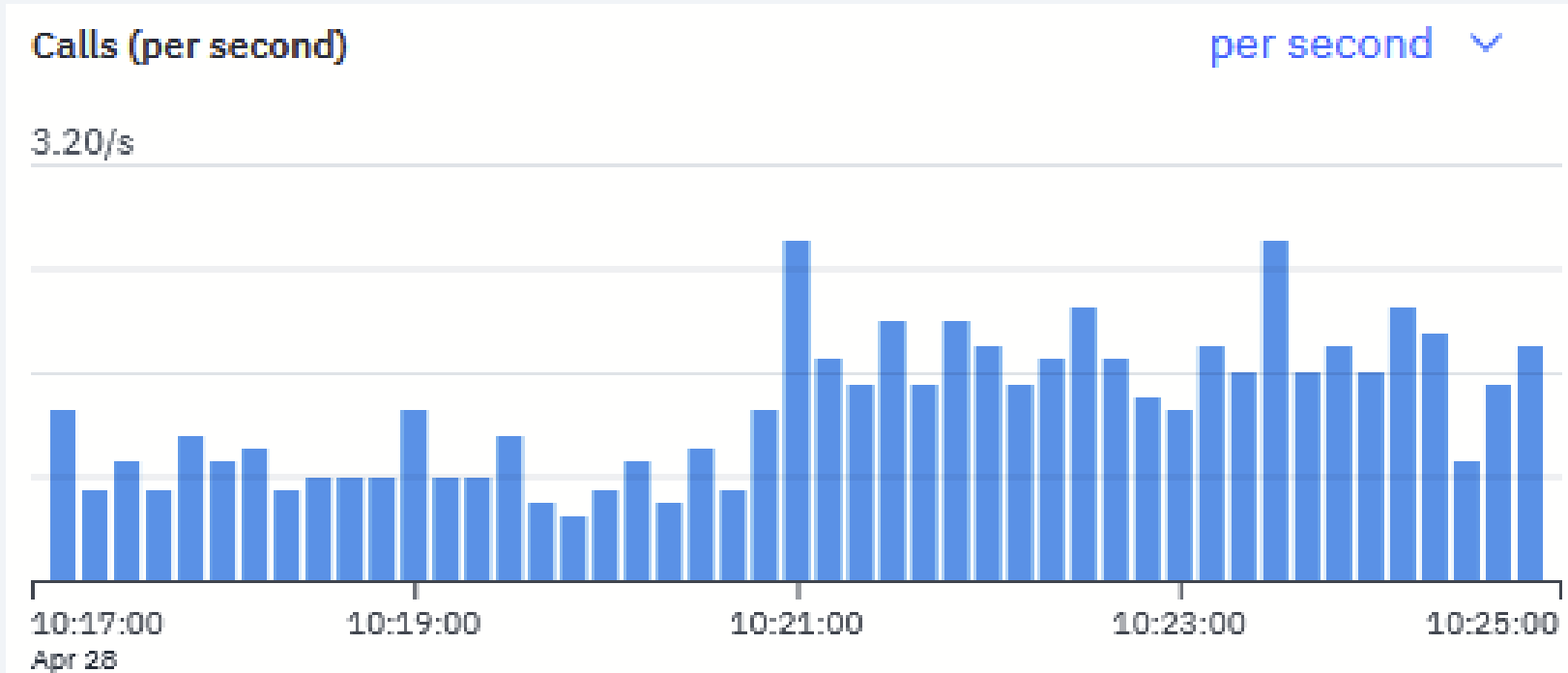
## Group by return codes



# Instana: Linux on IBM Z analyze calls

## Group by return codes

Call graph shows increase in frequency of AI fraud detection errors



**Screen shot:** Instana: Call diagram – AI fraud detection error

**Capability:** Your APM tool can show where the errors originate in your enterprise and how it propagates through the enterprise.

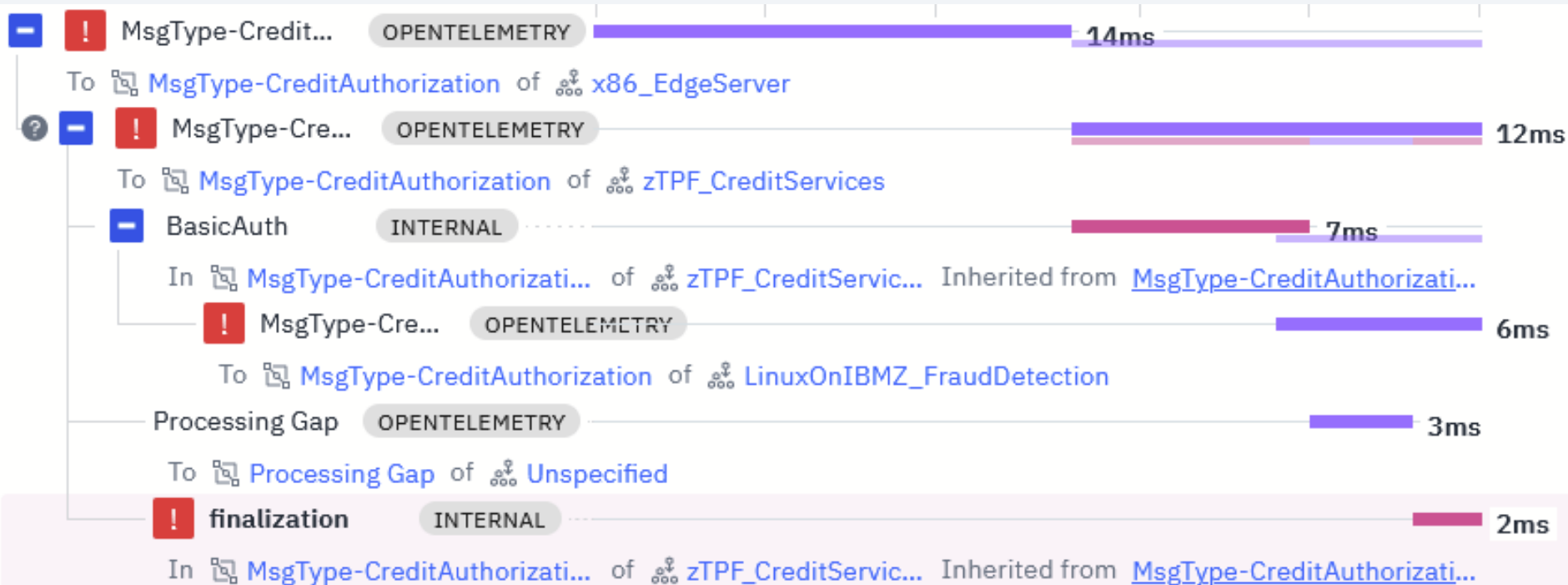
**Story:** As before, we can look at a message that is ending in error in the fraud detection on the Linux on IBM Z box. We're on our normal processing path so you can see:

- The error originates in the fraud detection processing on Linux on IBM Z.
- The error then propagates to the finalization processing on z/TPF.
- The error then propagates to the edge server.
- The overall message processing across our enterprise has ended in error.
- In this way we can see where errors originate in the processing in our enterprise.

**Business value:** With APM tools, you can see where errors originate in the processing of a message throughout your enterprise.

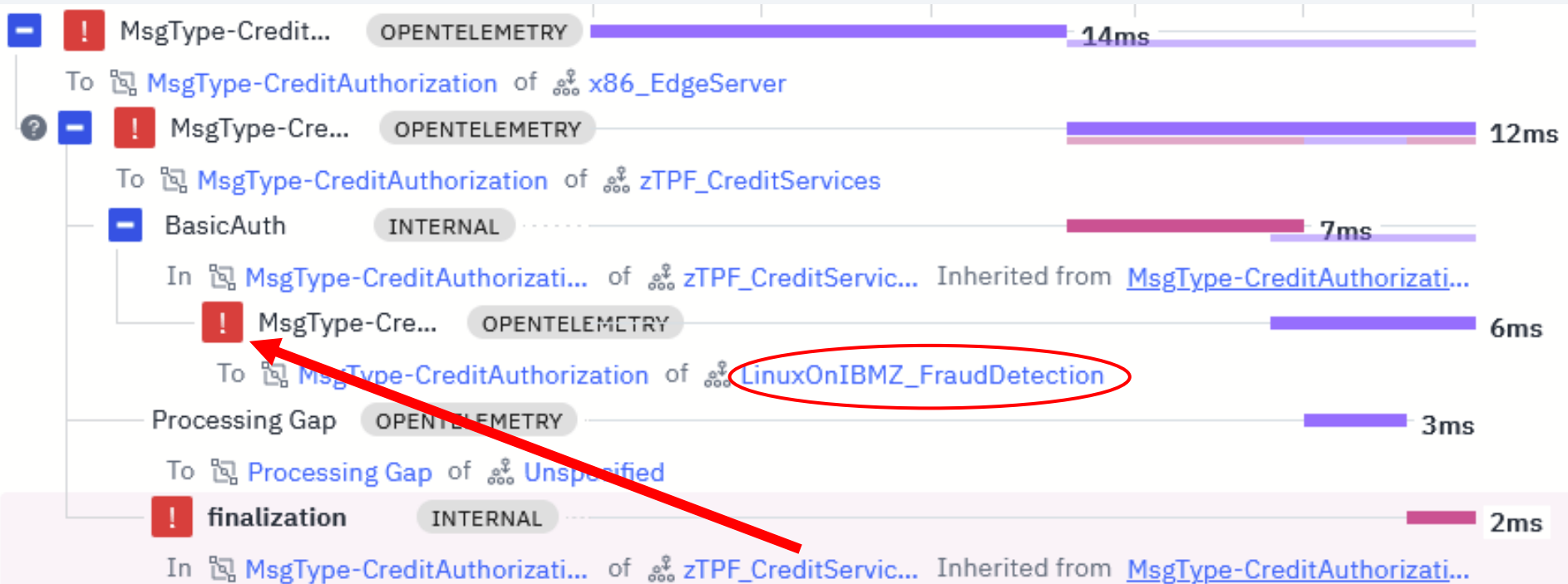
# Instana: Linux on IBM Z analyze call details

## AI fraud detection error



# Instana: Linux on IBM Z analyze call details

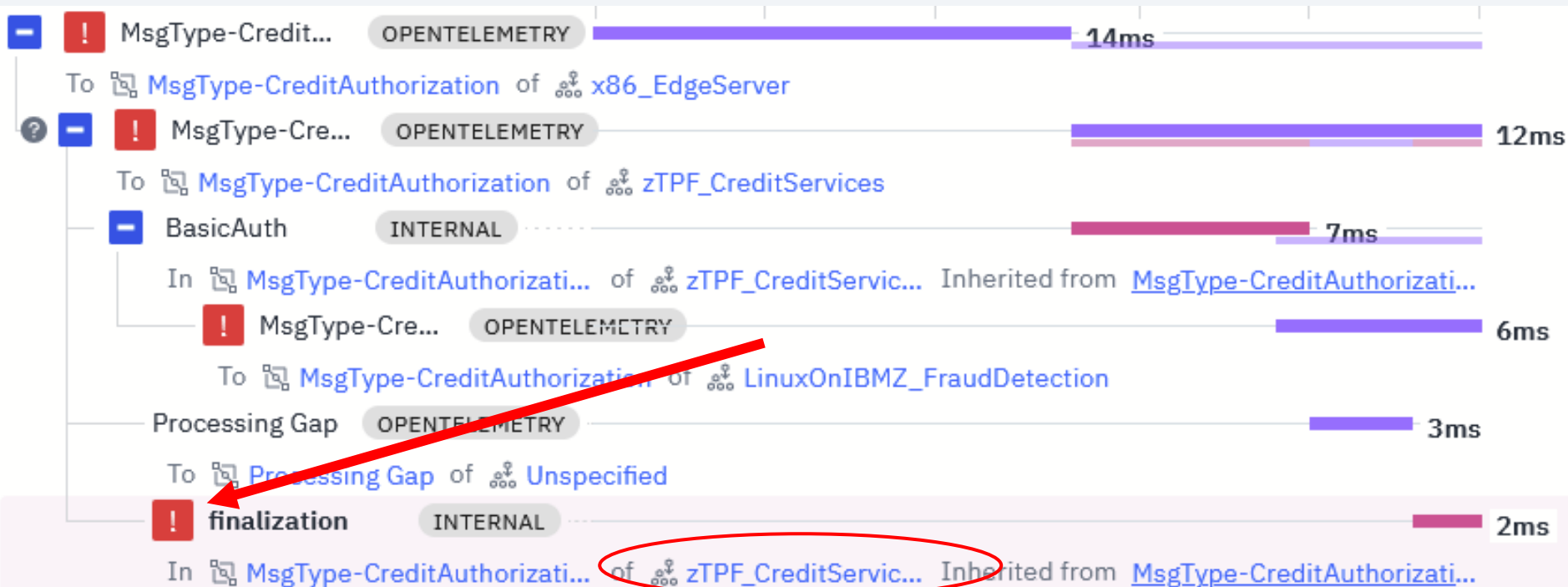
## AI fraud detection error originates on Linux on IBM Z





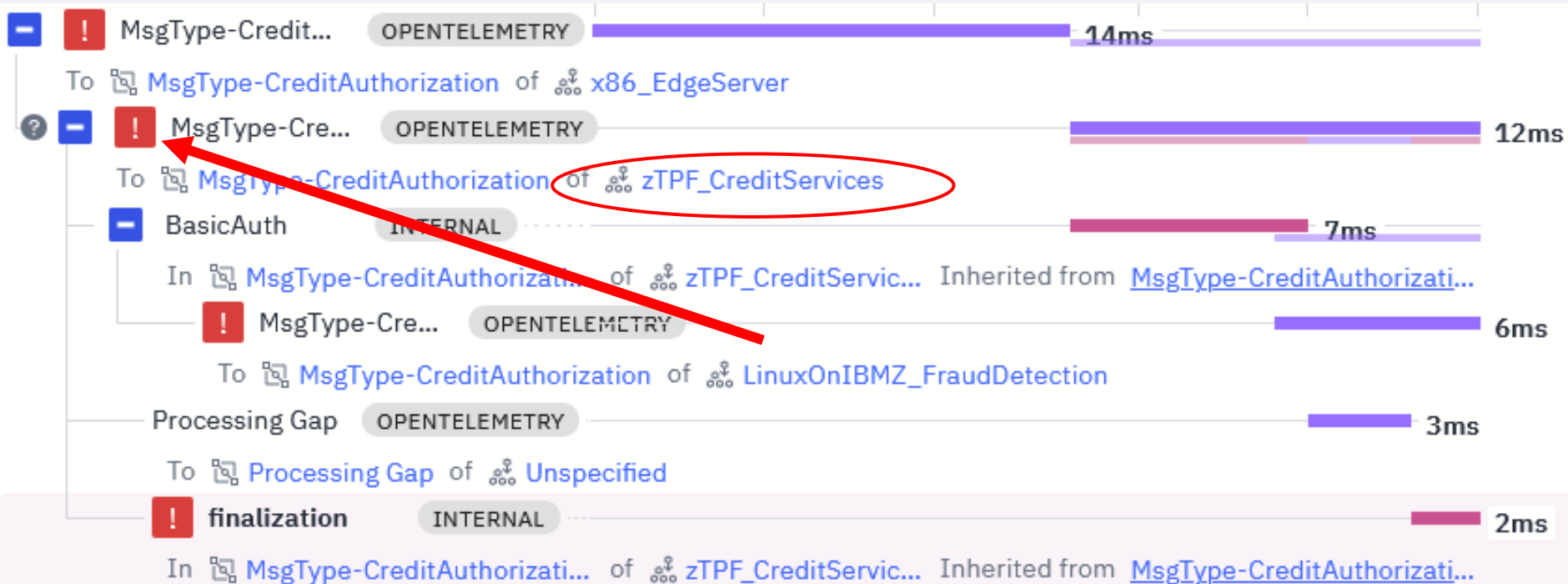
# Instana: Linux on IBM Z analyze call details

## Error propagates to z/TPF



# Instana: Linux on IBM Z analyze call details

## z/TPF summary node shows error



# Instana: Linux on IBM Z analyze call details

## Error propagates to edge server



# AI fraud detection errors – conclusion

- The site reliability engineer needs help from a **Linux on IBM Z** expert to continue the investigation.
- Notice that **the z/TPF silo was not contacted!**
- Notice that our **SRE did not have to be a z/TPF expert** to know that the **problem was not originating from z/TPF** and the **SRE did not have to be a Linux expert** to know that the **problem originated on Linux!**



Sarah  
site  
reliability  
engineer

# **RTMC**

## **Other features**

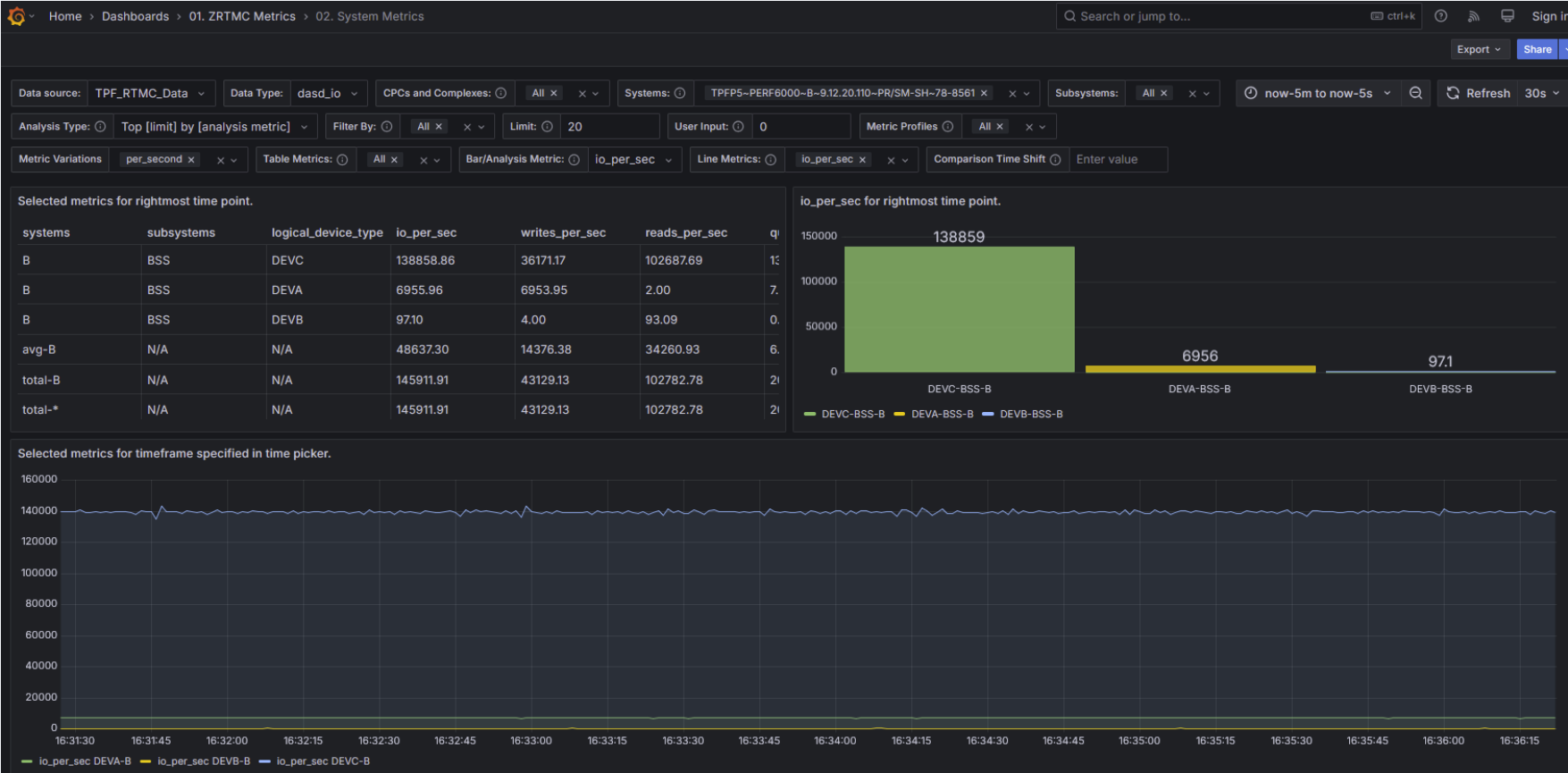
**Screen shot:** RTMC: System Metrics dashboard

**Capabilities:** The system metrics dashboard provides the deep system level metrics also known as continuous data collection (CDC). In this dashboard you can see metrics like DASD I/O operations, TCP/IP, z/TPFDF usage by database, IBM MQ, REST and much more.

**Business value:** The RTMC system metrics dashboard provides deep system level metrics to help you understand the health of your system.

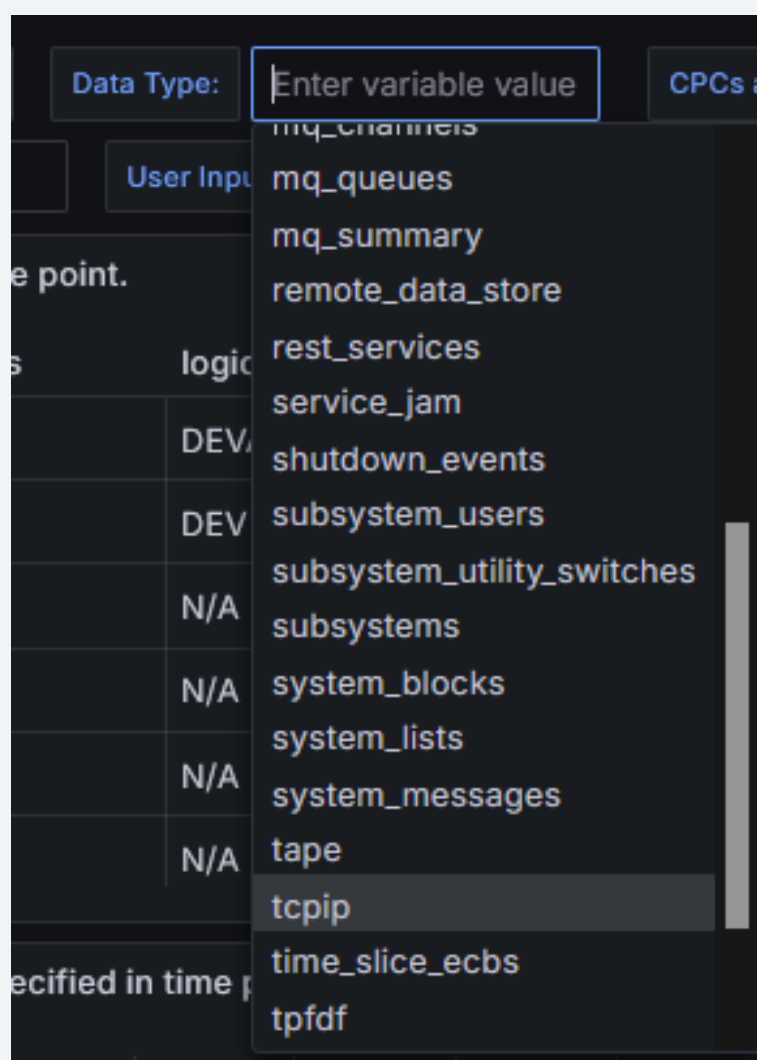
# RTMC: System metrics dashboard

## System health (CDC) metrics – the system is healthy



# RTMC: System metrics dashboard

## CDC data types list





**Screen shot:** RTMC: JVM dashboards

**Capabilities:** The JVM dashboards provide various insights for system resources used by your JVMs running on z/TPF as well as insights for lock usage by Java applications, garbage collection, application data and more.

**Business value:** The RTMC JVM metrics dashboards provides metrics for your Java on z/TPF usage to help you understand the health of your Java applications.

# RTMC: Java on z/TPF dashboards

## Name

- 01. JAM Summary
- 02. JVM Summary
- 03. JVM Drill-down
- 04. Lock Drill-down
- 05. JAM JAX-RS Drill-down
- 06. JAM JAX-RS Operation Drill-down
- 07. JVM Garbage Collection Table
- 08. JVM Static Data
- 09. JVM Command Line Arguments
- 10. JMX Application Data
- Education 1: JVM Monitoring

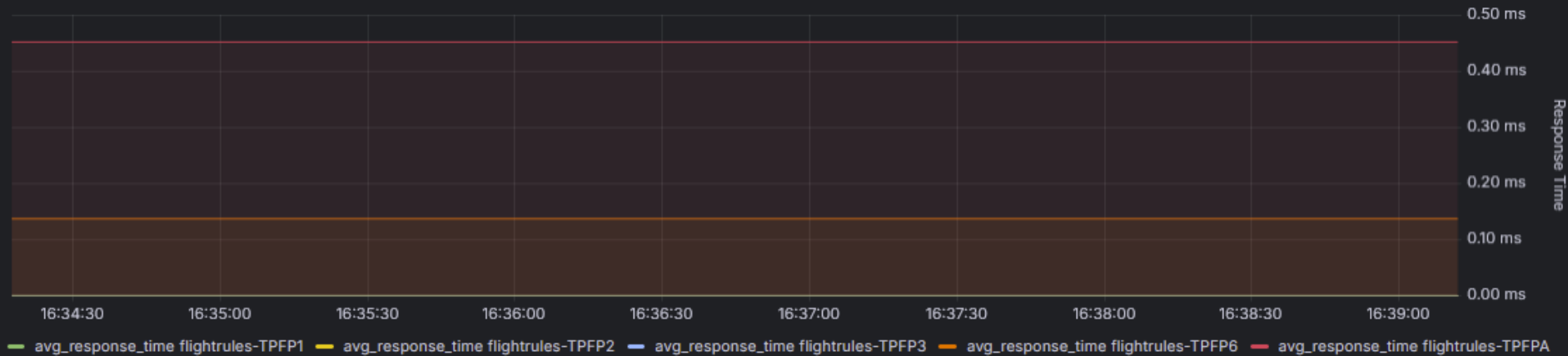
# RTMC: Java on z/TPF dashboards

## Example: JAM summary dashboard

JAM Summary Displaying Data for Rightmost Time Point ⓘ

System	Subsystem	JAM / Object	Total CPU	Avg CPU	JAM Status	# JVMs (Exp/Act)	# Threads (Tot/InUse/Max)	Avg Response Time
TPFP2~BI0001~A~9.12.2	BSS	flightrules	0.0%	0.0%	✓	1 / 1	4 / 0 / 0	0.00 ms
TPFP1~SK0001~B~9.12.2	BSS	flightrules	0.0%	0.0%	✓	1 / 1	4 / 0 / 0	0.00 ms
TPFPA~SL0001~B~9.12.2	BSS	flightrules	4.4%	4.4%	✓	1 / 1	4 / 0 / 2	0.45 ms
TPFP6~SK0001~M~9.12.2	BSS	flightrules	0.8%	0.8%	✓	1 / 1	4 / 0 / 1	0.14 ms

Select JAM/Object Metrics From Dropdown Above ⓘ



## **Screen shot:** RTMC: Name-value pair correlation analysis dashboard

**Capability:** The correlation analysis dashboard correlates system metrics to name-value pair metrics like message rate, CPU used, existence time and so on.

You could introduce machine learning, artificial intelligence, or trend analysis to identify outliers of interest. For example, if error rates exceed a normal threshold.

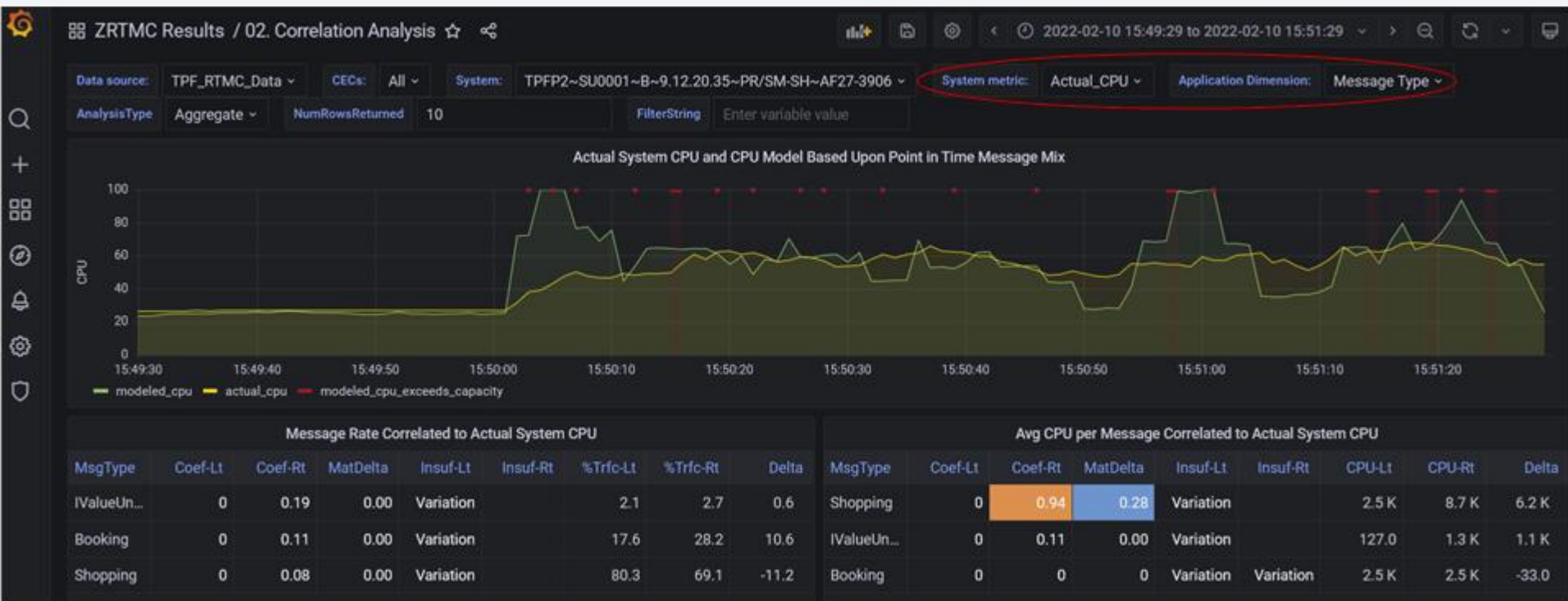
**Story:** In our scenarios, remember that our overall system metrics are stable. Maybe a tiny decrease in CPU usage and tiny increase in IO usage. For this use case, the default correlation analysis dashboard isn't going to provide us additional insights without something changing more dramatically at the system level.

In this scenario, maybe AI could immediately call out the deviations of the over the credit limit error originating in the basic authorization ECB when the transaction currency is Euros and the local currency is dollars and more IOs are used in the Currency Conversion package.

**Business value:** Data science analysis of RTMC results can provide quick insights into problem determination and can alert you before your SLAs are impacted.

# RTMC: Correlation analysis dashboard

## Correlate change in system metrics to changes in message, code package, and name-value pair combinations



**Screen shot:** RTMC: Name-value pair statistics dashboard

**Capability:** The name-value pair statistics dashboard shows min, max, avg and standard deviations for various metrics. You can also create other analysis and dashboards to view various statistics for your data. One we envision creating one day is change point detection.

**Business value:** Statistical analysis of RTMC results can provide quick insights into problem determination.

# RTMC: Name-value pair statistics dashboard

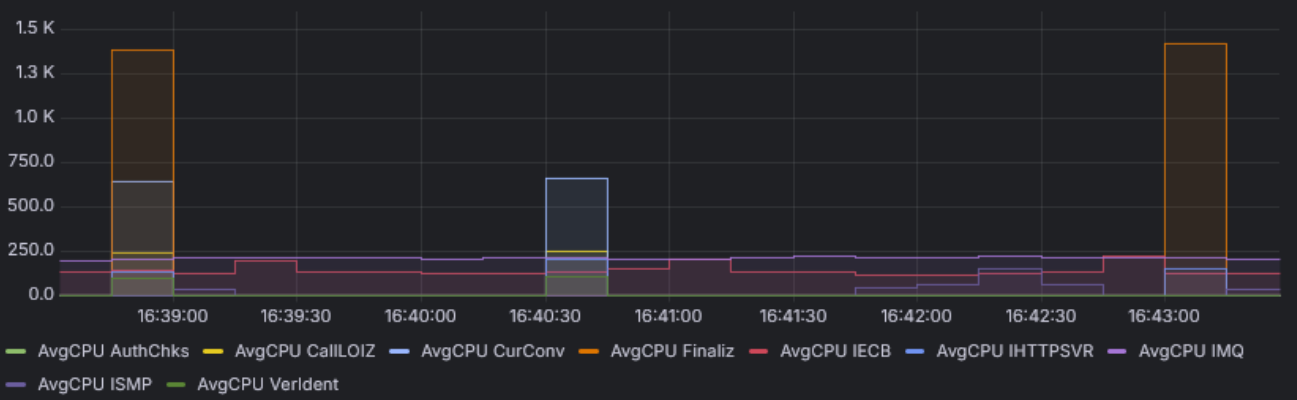
Avg CPU per Message by Code Package

Code Pk	Lt	Rt	%-Del	hltr	Min
ISMP	0.0	40.0	100.0	100.0	0.0
IMQ	200.0	209.0	4.5	4.5	200.4
IECB	132.0	128.0	-3.0	3.0	116.4
VerIdent	0.0	0.0	0.0	0.0	0.0
CallILOIZ	0.0	0.0	0.0	0.0	0.0
CurConv	0.0	0.0	0.0	0.0	0.0
Finaliz	0.0	0.0	0.0	0.0	0.0
AuthChk	0.0	0.0	0.0	0.0	0.0
IHTTPS	0.0	0.0	0.0	0.0	0.0

System Level Metrics (Including CPU Model Based Upon Point in Time Message Mix)



Avg CPU per Message by Code Package



## **Screen shot:** RTMC: User-defined metrics

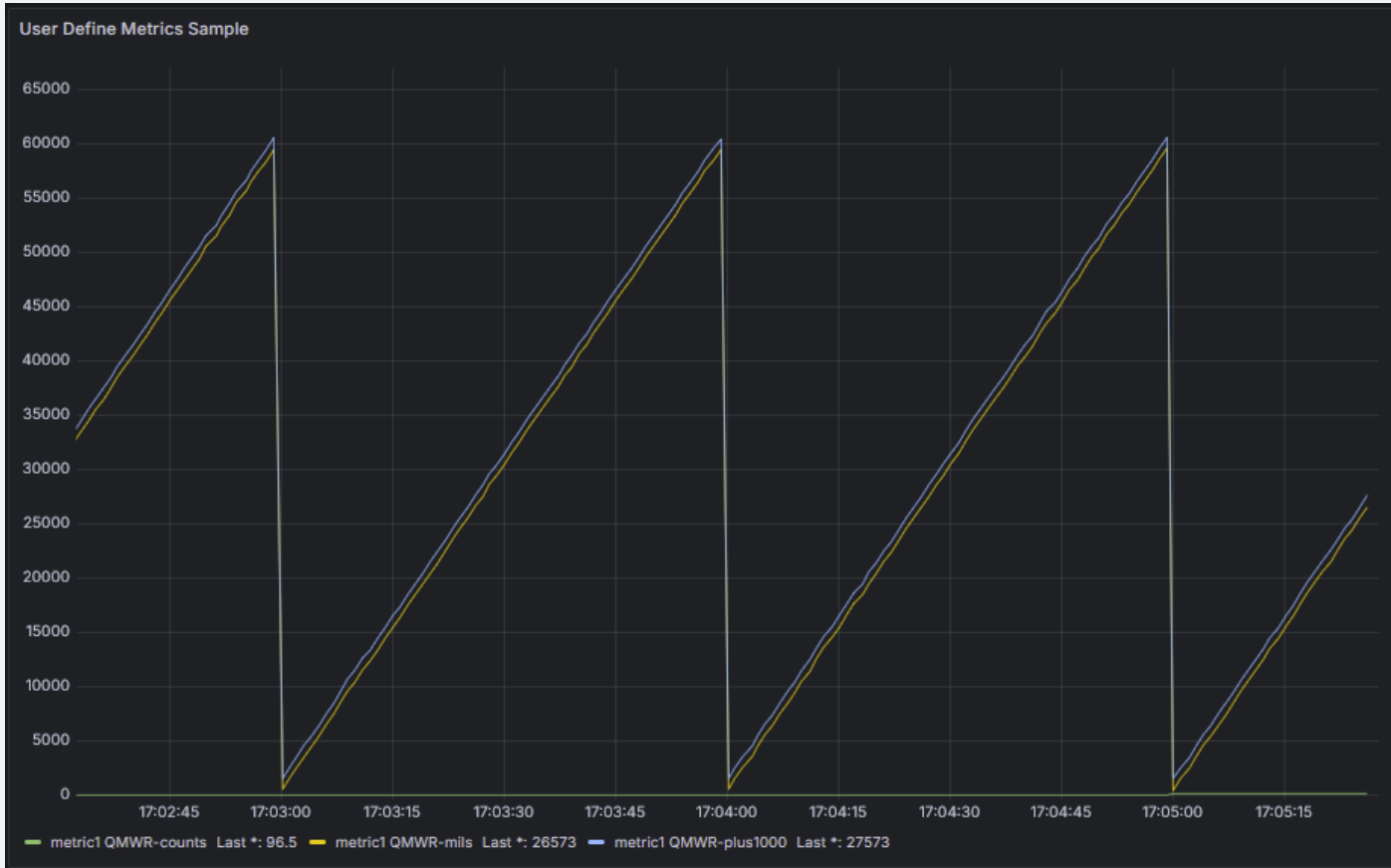
**Capability:** With the user-defined metrics, you can send system, application and business metrics through the RTMC sample analytics pipeline and perform analytics in real-time. We provide a sample dashboard, processing, tutorial, and online driver to generate a simple sawtooth pattern.

**Business value:** With the RTMC user-defined metrics, you can send system, application and business metrics through the RTMC sample analytics pipeline and perform analytics in real-time.



# RTMC: User-defined metrics

## Tutorial metrics and sample dashboard



## **Screen shot:** RTMC: User-defined metrics – Application error rates

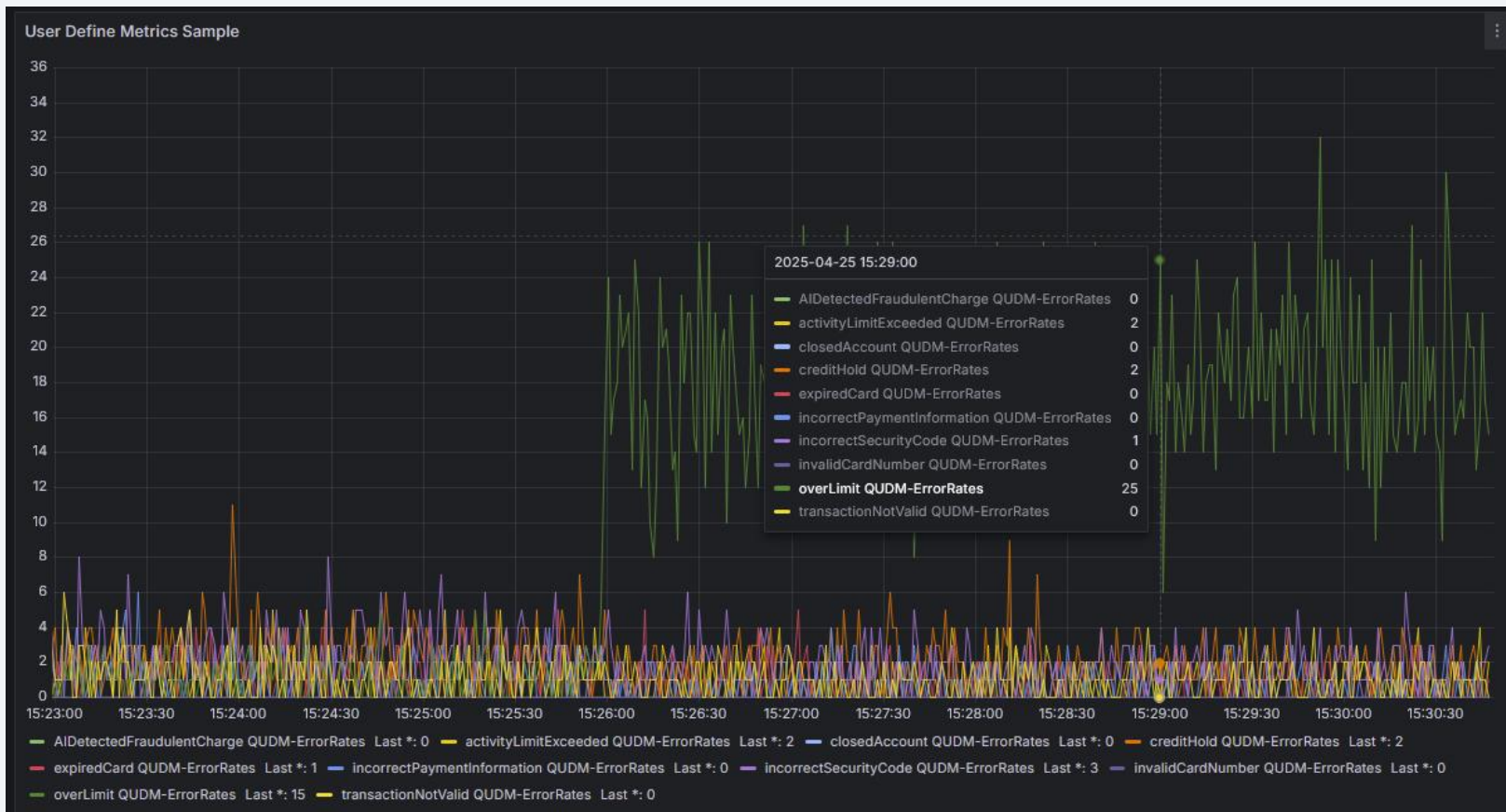
**Story:** For the over the credit limit story, we created a new user-defined metrics example. This display shows application error rates for our credit authorization messages. You can see that the error rate for the over the credit limit errors increased at the inflection point while other message types remain relatively stable.

This is just one example of the sort of thing you can do instead of having a periodic ZSTIM, screen scraping the metrics from the console at the end of the day and putting the results into a database. Instead, you can do analytics and investigations in real-time.

**Business value:** With the RTMC user-defined metrics, you can send system, application, and business metrics through the RTMC sample analytics pipeline and perform analytics in real-time.

# RTMC: User-defined metrics

## Real world application error counts dashboard example



# The no disclaimer slide

- Remember that there was no disclaimer slide!
- **Everything demonstrated is available today!**



Sarah  
site  
reliability  
engineer



Carol  
z/TPF  
coverage  
programmer



Zach  
application  
developer

# Thank you

© Copyright IBM Corporation 2025. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., 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 at [Copyright and trademark information](#).

