z/TPF Message Analysis Tool

2022 TPF Users Group Conference
March 27-30, Dallas, TX
Operations and Coverage

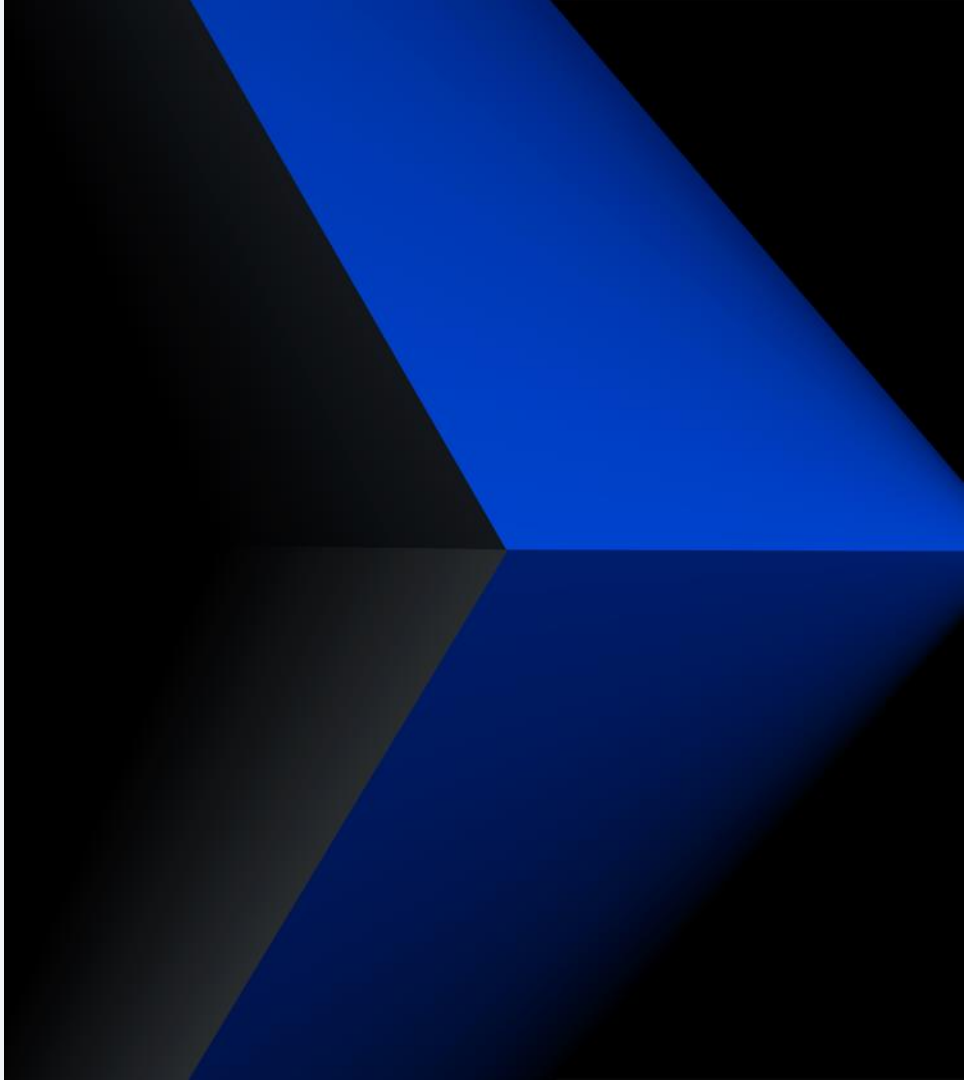
Gabriel Nieves

IBM Z

IBM

Disclaimer

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



Problem Statement

- How do you understand where changes in resource usage (CPU, IO, memory, etc.) occur due to application changes during development?
- How do you identify code refactoring opportunities with the largest ROI?
- How do you educate new application developers about the code flow of real transactions in production?

Overview

z/TPF message analysis tool provides you with the capability to capture and analyze the functions and macros that are used when the system processes a message.

APAR PJ46308 (12/2021) and the latest z/TPF real-time insights dashboard starter kit are now available.

Message Analysis Collection

- New ZMATC command manages the collection of ECB trace log data for a message based on a set of user-defined criteria.
- Message is one or more ECBs with the same unit-of-work ID (UOWID).
- When an ECB enters the target program, it must match the rest of the selection criteria to have its UOWID selected for a new collection.
- Collections can be done on a production system with minimum impact.
- This support leverages the new post-processing program option from the trace log facility delivered with APAR PJ46309 (11/2021).
- Data is sent to the Message Analysis Tool using high-speed connector.

Message Analysis Collection

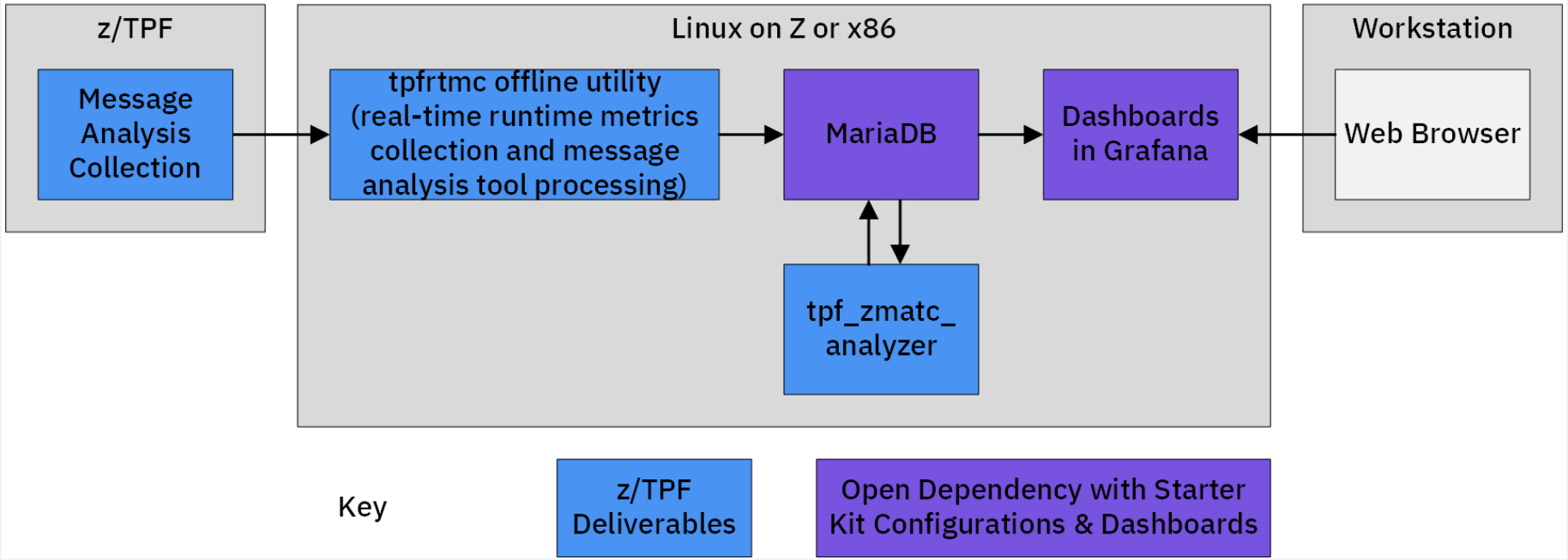
- A JSON-based configuration file is used to define the selection criteria:
  - Timeout values for individual messages and full collection.
  - Limit number of messages for a full collection.
  - Select messages from a specific terminal address (LNIATA).
  - Select messages from the program of a selectively-activated loadset.
  - Select messages with a specified name-value pair assigned.
  - Option to include/exclude children ECBs of the selected message.

Message Analysis Tool

- Web-based GUI dashboard for viewing collections.
- The tpfrtmc offline utility is built on open-source solutions such as MariaDB, Grafana, and Docker.
- Sample starter kit deploys all the components on Linux through a Dockerfile.
- Customizable and extendable by the user.
- Capable of analyzing real-time collections from a production environment.
- Provided at no additional costs or products to purchase
- Available now with the latest z/TPF real-time insights dashboard starter kit: https://www.ibm.com/support/pages/ztpf-real-time-insights-dashboard-starter-kit

Message Analysis Tool

- The message analysis tool uses the same processing chain as real-time runtime metrics collection.
- One instance of the tpfrtmc offline utility can service both the message analysis tool collection and real-time runtime metrics collect.



Value Statement

- Andrew, an application developer, can identify earlier in the development cycle where excessive system resources are consumed by an application due to their code changes thereby reducing the time and costs required for development and test. ***(Delivered – APAR PJ46308)***
- Arthur, an application architect, can analyze data collected from production traffic to identify where real-world inefficiencies exist in applications in more in-depth details than was possible using the TPF Toolkit 4.2 Performance Analyzer. ***(Delivered – APAR PJ46308)***
- Zach, a new application developer, can use application flow data captured from production systems during onboarding to learn how an application works and become productive more quickly. ***(Future)***

As-Is User Story

- Andrew modified an existing application.
- They used real-time runtime metrics collection to determine **if** there was a problematic difference in resources used by the modified application when compared to the base code.
- Suppose they found that the modified application is now using 30% more CPU than was anticipated by their acceptance criteria specified by the application architects.
- Now Andrew needs to determine **where** that additional CPU is being used.

Andrew

Application Developer

To-Be User Story

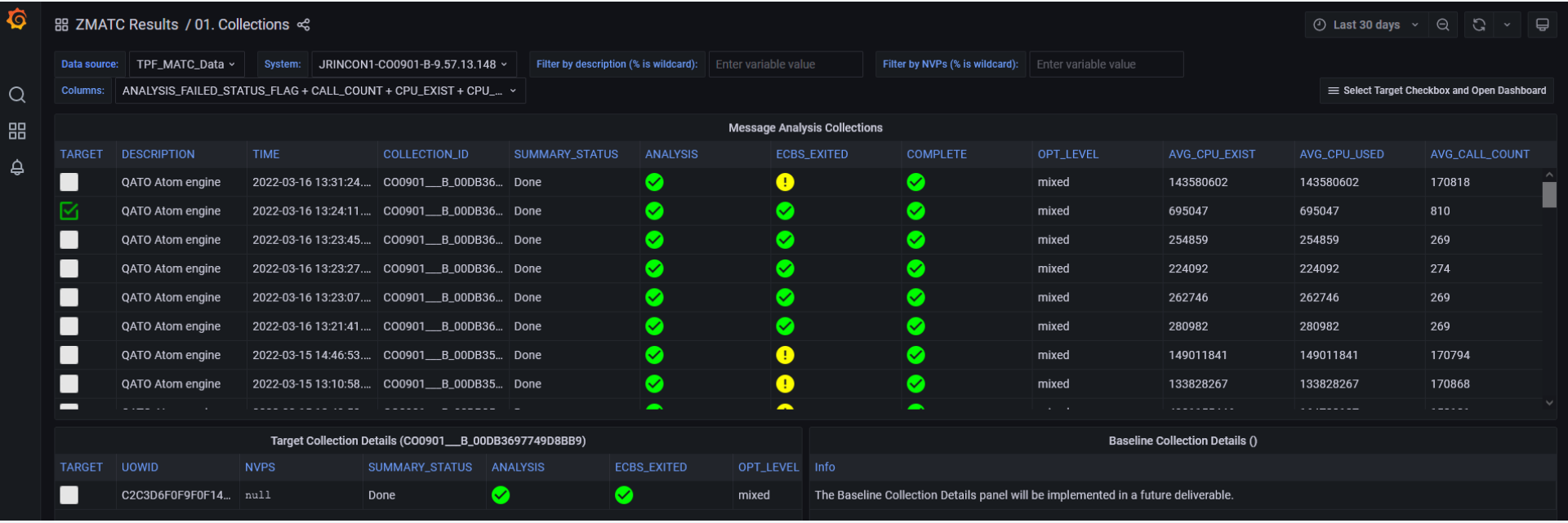
- Andrew uses the z/TPF message analysis tool to discover where those resources are being used.
- A collection is performed on both versions of the application with the original code base and new development changes.
- The Collections and Summary Details dashboards are used to find the average CPU usage of and number of calls to each function within the application code path.
- Andrew identifies that their application changes cause a module that they did NOT modify to use substantially more resources.

Andrew

Application Developer

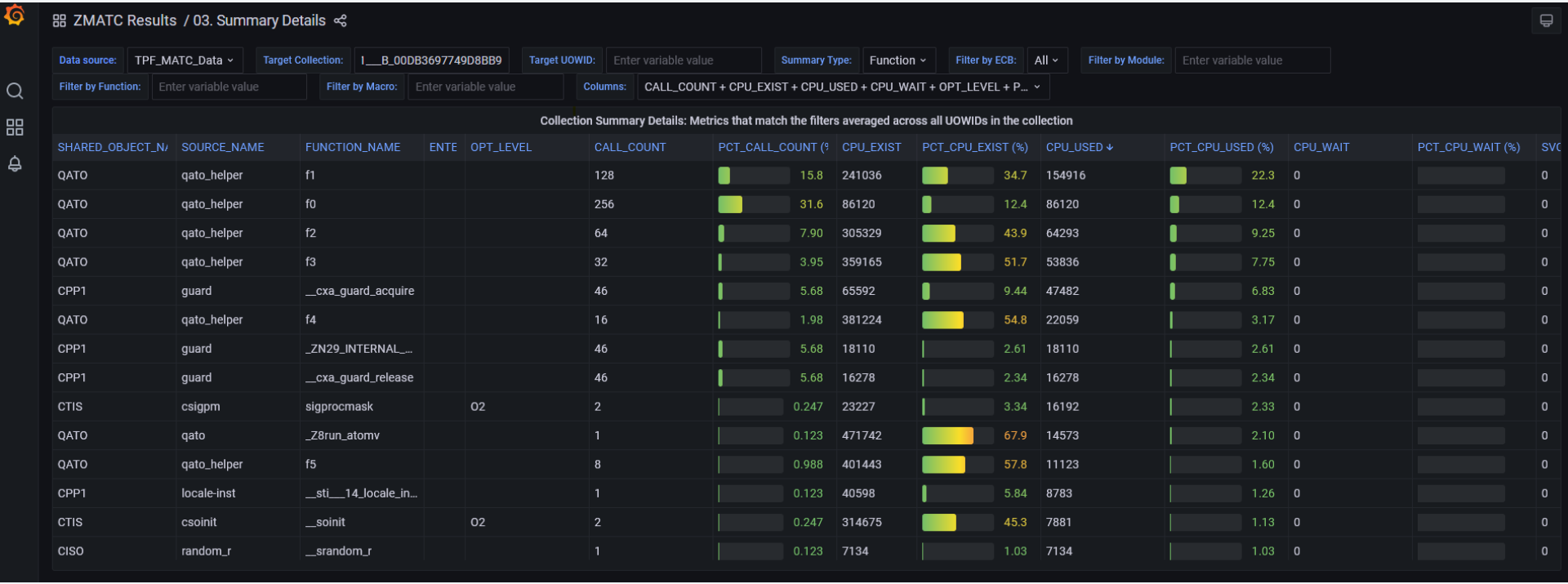
To-Be User Story

- The Collections dashboard allows you to see the list of available collections.



To-Be User Story

- The Summary Details dashboard shows summarized metrics for functions, modules, etc.



SHARED_OBJECT_N/	SOURCE_NAME	FUNCTION_NAME	ENTE	OPT_LEVEL	CALL_COUNT	PCT_CALL_COUNT (%	CPU_EXIST	PCT_CPU_EXIST (%)	CPU_USED ↓	PCT_CPU_USED (%)	CPU_WAIT	PCT_CPU_WAIT (%)	SVC
QATO	qato_helper	f1			128	15.8	241036	34.7	154916	22.3	0		0
QATO	qato_helper	f0			256	31.6	86120	12.4	86120	12.4	0		0
QATO	qato_helper	f2			64	7.90	305329	43.9	64293	9.25	0		0
QATO	qato_helper	f3			32	3.95	359165	51.7	53836	7.75	0		0
CPP1	guard	__cxa_guard_acquire			46	5.68	65592	9.44	47482	6.83	0		0
QATO	qato_helper	f4			16	1.98	381224	54.8	22059	3.17	0		0
CPP1	guard	_ZN29_INTERNAL_...			46	5.68	18110	2.61	18110	2.61	0		0
CPP1	guard	__cxa_guard_release			46	5.68	16278	2.34	16278	2.34	0		0
CTIS	csigpm	sigprocmask		O2	2	0.247	23227	3.34	16192	2.33	0		0
QATO	qato	_Z8run_atomv			1	0.123	471742	67.9	14573	2.10	0		0
QATO	qato_helper	f5			8	0.988	401443	57.8	11123	1.60	0		0
CPP1	locale-inst	__sti___14_locale_in...			1	0.123	40598	5.84	8783	1.26	0		0
CTIS	csoinit	__soinit		O2	2	0.247	314675	45.3	7881	1.13	0		0
CISO	random_r	__srandom_r			1	0.123	7134	1.03	7134	1.03	0		0

To-Be User Story

- They spent an hour to diagnose the issue early in development instead of the As-Is scenario of spending a week late during the testing phase.
- The development team refactors the design and modifies the code early in the project when change is easier to absorb.
- As Andrew completes their work, they create a report from the z/TPF message analysis tool results to document that their application changes are within their acceptance criteria specified by the application architects.

Andrew

Application Developer

TPF Toolkit Performance Analyzer Comparison

- Benefits of the z/TPF Message Analysis Tool:
  - Perform collections on a production environment.
  - SVC macros usage are included in the analyzed trace data.
  - Child ECBs of the same UOWID are part of the collection.
  - Summary Details dashboard gives results across multiple collections in a single view.

New Features Roadmap (In-Progress)

- Application Code Flow – shows the sequence of function calls throughout the life of the transaction.
- ECB Troubleshoot Dashboard – assists in identifying problems from the function caller's perspective.
- Results Comparison Dashboard – compares the aggregated trace entries across different collections.

We want sponsor users!

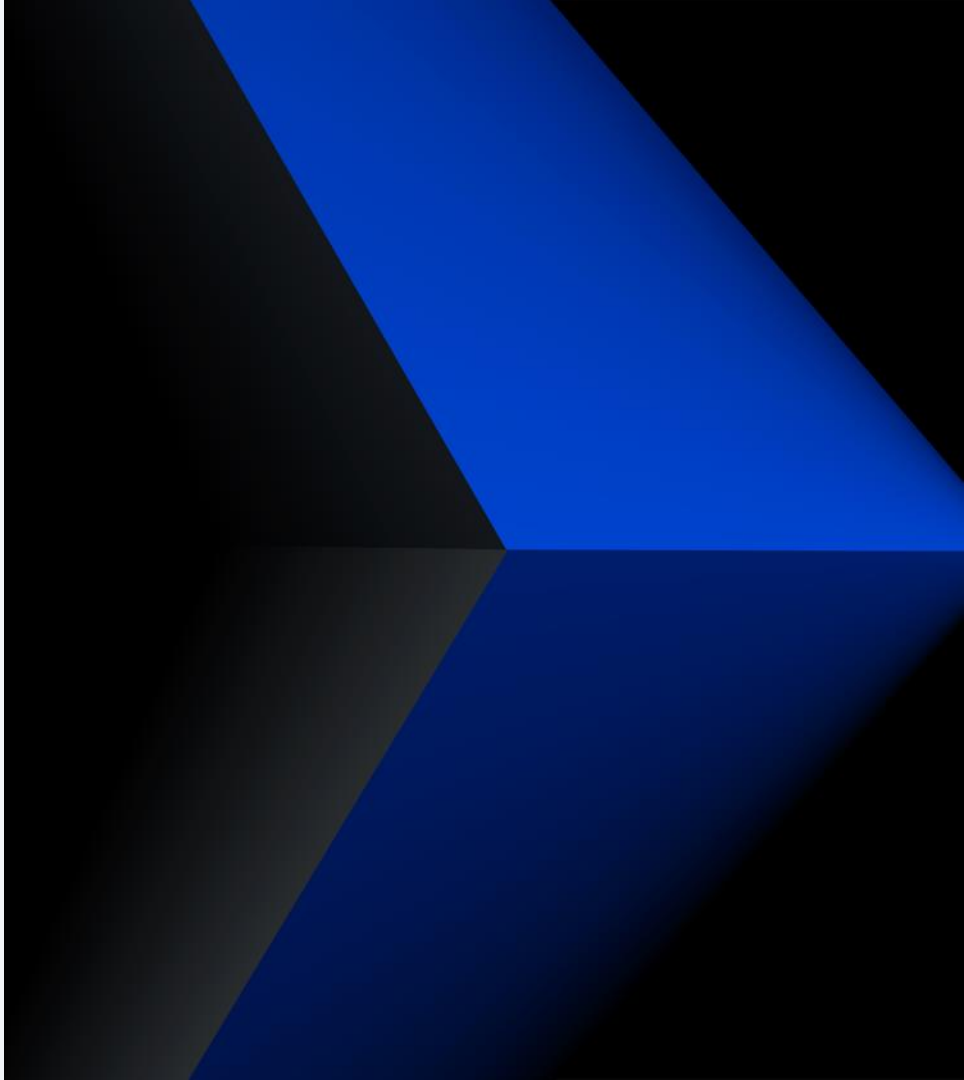
Our development cycle is driven by your feedback.

We are looking for sponsor users to assist in design and implementation, targeting the following personas:

- Application Developer
- Coverage Programmer
- Operator

If you are interested in participating as a sponsor user, please contact:

Josh Wisniewski (jwisniew@us.ibm.com)



Thank you