ADI

Application Delivery Intelligence 2016

Josh Wisniewski
Tooling Architect

IBM **z/TPF**

April 11, 2016

10 Minutes	**What is ADI?**
18 Minutes	**Name-value pair collection**
2 Minutes	**More information**

Agenda

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.

What is ADI?

- In early 2015, several customers gave the feedback that the most significant strategic problems facing customers today are:
  - Large code bases of legacy code dating back to the 1960s. Subject matter experts (SMEs) no longer available.
  - Smaller work force. Impending baby-boomer retirements will result in fewer SMEs.
  - Systems are highly complex. Difficult to train up new SMEs.
  - Heavy reliance on a dwindling number of SMEs to convey tribal knowledge (how things work, impacts of making a change, how to test, and etc).

- Throughout 2015 and 2016, IBM engaged with various customers in design thinking discussions to define a set of tooling that can aid in the preservation of SME knowledge, generation of new insights into code bases and more.
- The goal can be summarized with the following statement:

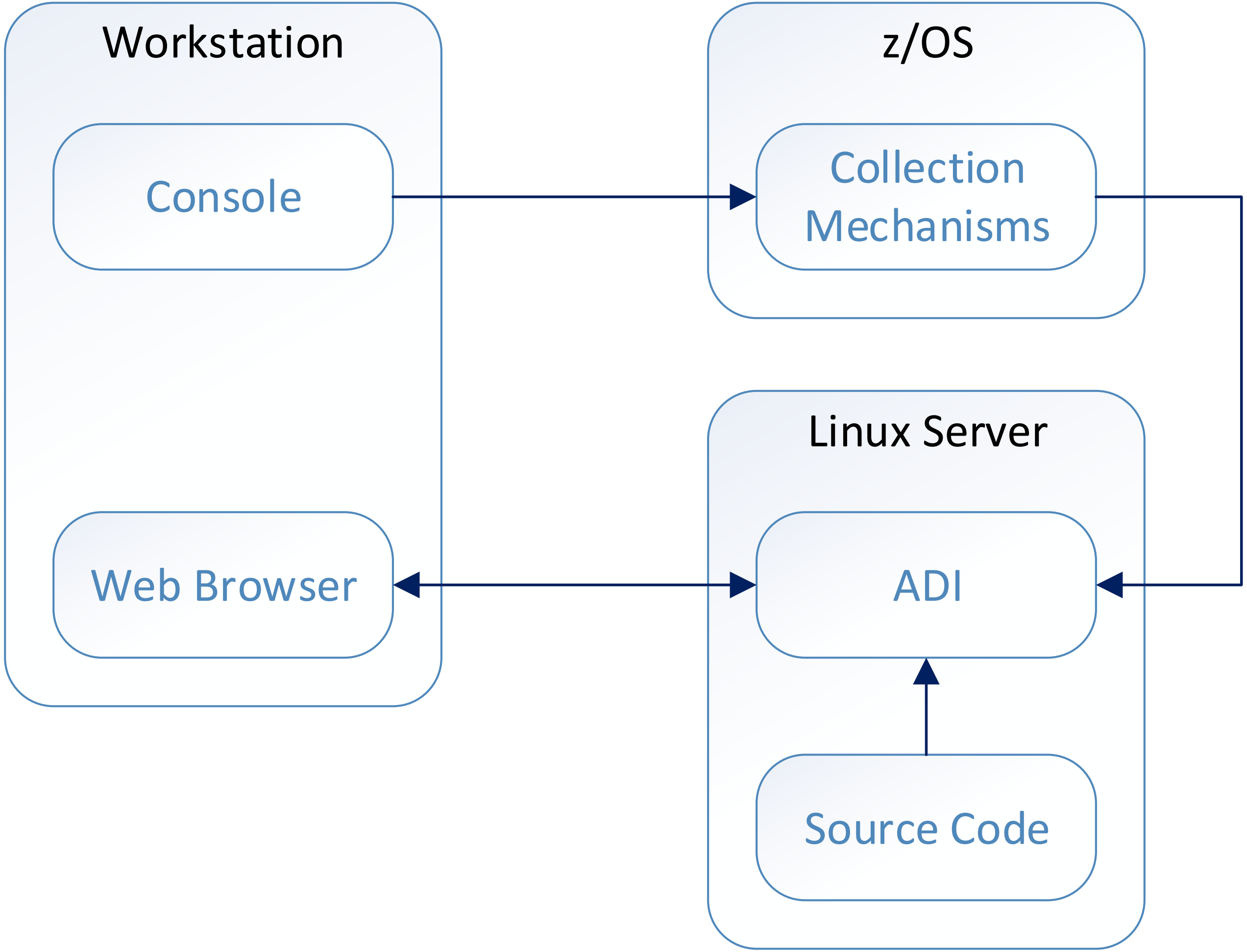
  - A user can **view integrated code analysis, metrics and user documentation** to more **effectively develop code, diagnose problems and make strategic decisions**.

- The tooling is intended to eventually provide:
  - Static code analysis
  - Execution flow
  - Data flow
  - Integration of run time metrics
  - Documentation
  - Modeling
  - Testing knowledge
  - And more...

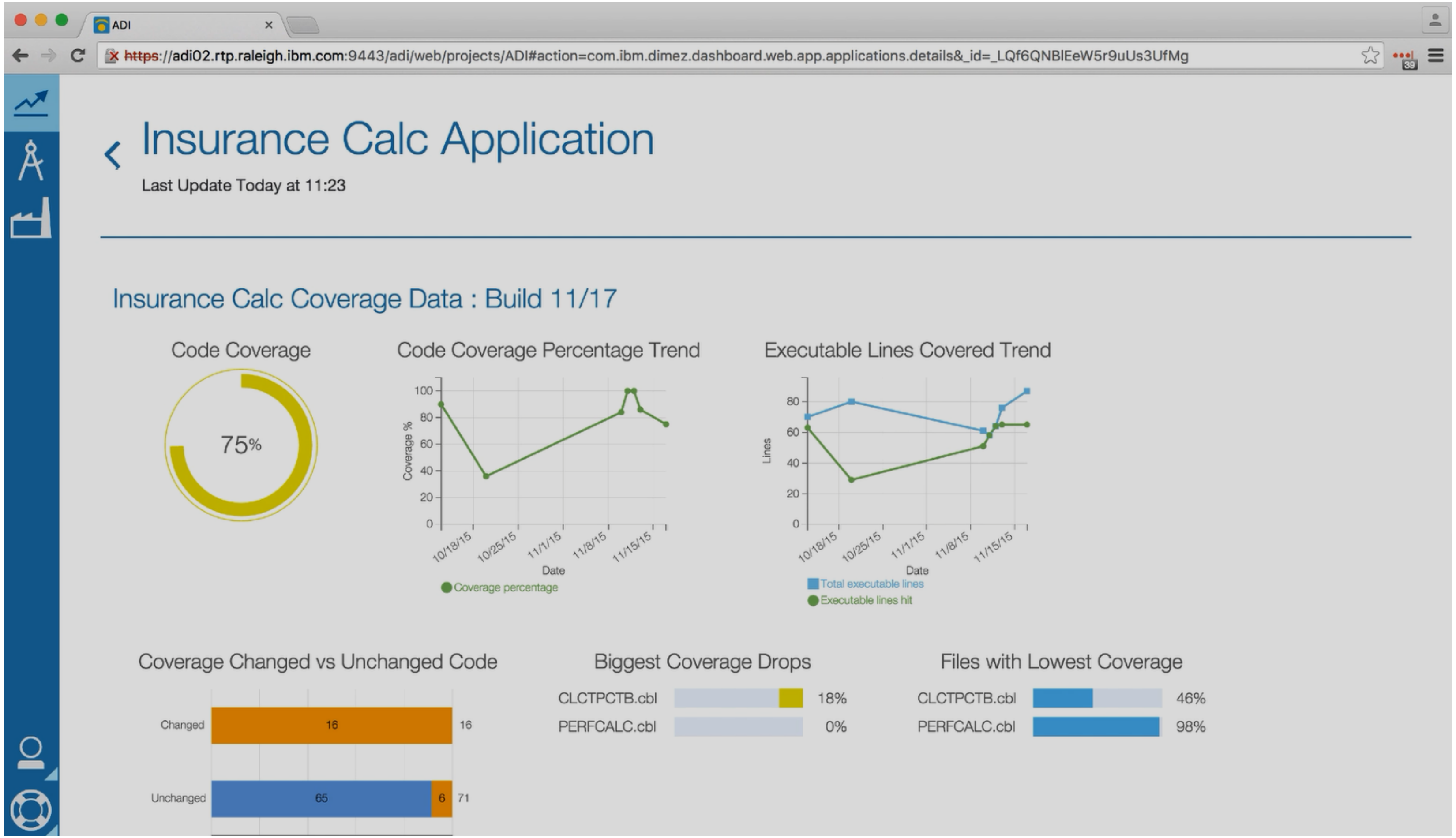
- Example usage: A developer looks at function ABCD and can quickly see:
  - Who calls function ABCD? What request types cause ABCD to be called?
  - What data is passed in? Passed out? Accessed inside?
  - What system resources are consumed by calling ABCD? What resources are consumed by functions and macros called by ABCD?
  - Where does ABCD fit in the application design? Documentation on how ABCD works.

- This tooling is planned to be delivered in a new product: “IBM Application Delivery Intelligence (ADI)”.
- ADI is an Enterprise Continuous Integration analytics dashboard and optimization software with focus on application understanding, performance and quality metrics and trends.
  - Web-based centralized solution running on a Linux server optimized to work with existing IBM Enterprise Continuous Integration tools.
  - Shift from Displaying Data to Providing Insight and trends.
  - Make Continuous Improvement real by allowing timely feedback for IBM z Systems customers.

What is ADI?


Workstation
Console
Web Browser
z/OS
Collection Mechanisms
Linux Server
ADI
Source Code

- The first release of ADI focuses on Test Optimization Analysis
  - Reduce time and effort in running regression test by suggesting the list of tests to run to achieve maximum code coverage
  - Provide insight into the applications which do not meet the set thresholds for code coverage
- The first release of ADI was 3/2016 with support for z/OS. z/TPF is not supported in the first release of ADI.
- Plan to add support to ADI for z/TPF in 2016.



ADI

https://adi02.rtp.raleigh.ibm.com:9443/adi/web/projects/ADI#action=com.ibm.dimez.dashboard.web.app.applications.details&_id=_LQf6QNBlEeW5r9uUs3UfMg

Insurance Calc Application

Last Update Today at 11:23

Insurance Calc Coverage Data : Build 11/17

Code Coverage

Code Coverage Percentage Trend

Executable Lines Covered Trend

Coverage Changed vs Unchanged Code

Biggest Coverage Drops

CLCTPCTB.cbl	18%
PERFCALC.cbl	0%

Files with Lowest Coverage

CLCTPCTB.cbl	46%
PERFCALC.cbl	98%

What is ADI?

- See the following links for more information about ADI:
  - http://www-03.ibm.com/software/products/en/ibm-application-delivery-intelligence
  - https://www.youtube.com/watch?v=1eLMurepPPg

- The TPF team will add platform independent features to ADI such that the features added can be used by other platforms supported by ADI (such as z/OS).

Name-value pair collection

- The TPF team plans to add the name-value pair collection feature to ADI in 2016.
- Name-value pair collection can be summarized in the following Hill statements:
  - A coverage programmer can use name-value pair collection to gain new insights into system resource usage and identify the source of problems in as little as 1/20 of the time previously required.

- Sub-Hill 1: A coverage programmer can use name-value pair collection to identify factors that are causing a significant increase in resource usage based upon new metrics such as message type, end user, and code package.
- Sub-Hill 2: A capacity planner can use name-value pair collection to determine the additional physical assets required for the expected message rate growth of a specific service given new resource usage metrics.
- Sub-Hill 3: An application architect can use name-value pair collection to identify inefficient code packages that can be refactored to improve system performance.

- What are name-value pairs?
  - Existing mechanism that allows you to make data available anywhere it is required in an application without making interface changes.
  - For example, name-value pairs can be used to pass credentials (user id, passwords, and etc).

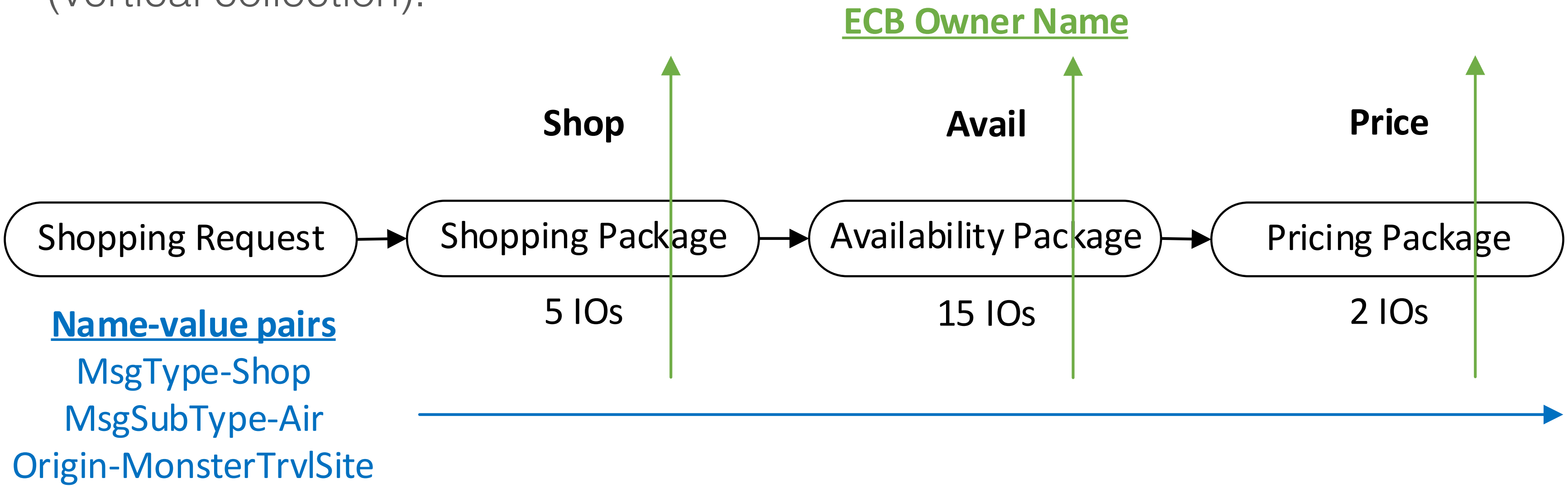
- Name-value pair collection leverages the name-value pair mechanism.
- Customers will update their code to set name-value pairs such as
  - Describing the request to be processed (message type, message subtype, input parameters, etc)
  - Describing the origin of the request (customer, geography, etc)
  - Providing unique ids that tie ECBs, cross system calls, requests, transactions and etc into units of work.

- What are ECB Owner Names?
  - Existing mechanism that allows you to tag an ECB with up to 3 textual qualifiers.
  - ECB owner names are leveraged by owner name collection (ZMOWN) as way to categorize resource usage by specific applications.
  - ECB owner names are also used as block owner names for some block types, for display purposes in some z commands, and etc.

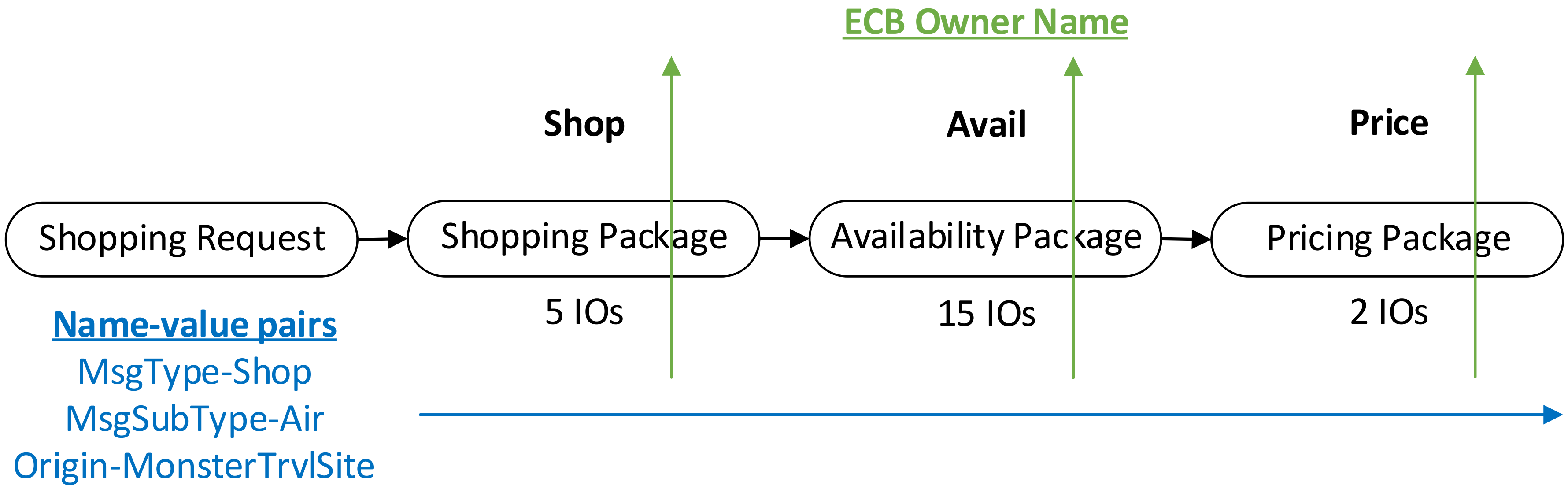
- Name-value pair collection also leverages the owner name mechanism.
- Customers will update their code to set owner names at the primary interface points to code packages to
  - Describe the code package currently being executed.

Name-value pair collection

- Name-value pairs will be used to denote resource usage by details of the request (horizontal collection).
- ECB owner name will be used to denote resource usage by code package (vertical collection).



Name-value pair collection



ECB Owner Name

Shop

Avail

Price

Shopping Request

Shopping Package

Availability Package

Pricing Package

5 IOs

15 IOs

2 IOs

Name-value pairs

MsgType-Shop

MsgSubType-Air

Origin-MonsterTrvlSite

Name-value pair	IOs
MsgType-Shop MsgSubType-Air Origin-MonsterTrvlSite	22

ECB Owner Name	IOs
Shop	5
Avail	15
Price	2

Name-value pair collection

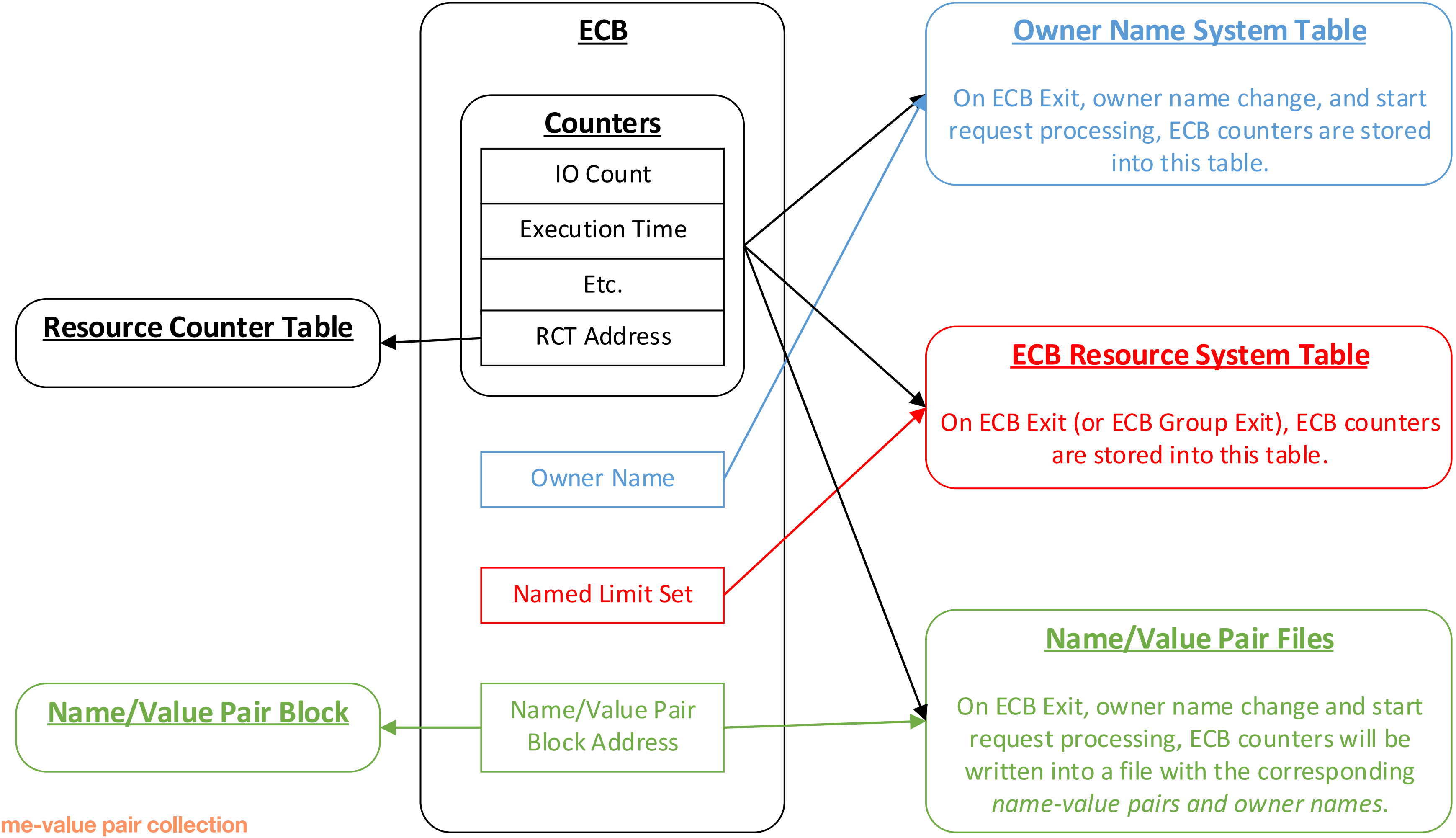
- One of the advantages of name-value pair collection and ADI analysis, is that the results will be available in both the horizontal and vertical.
- In the table below, we have a Shop-Air request from MonsterTrvlSite that performs 22 IOs on average. The column on the right shows where (in which code packages) the IOs are occurring for this type of request.

Name-Value Pair	IOs	ECB Owner Name Breakdown	
MsgType-Shop MsgSubType-Air Origin-MonsterTrvlSite	22	Shop	5
		Avail	15
		Price	2

- And based upon the breakdowns of the data you require.
- In the table below, we have a code package Avail that performs 15 IOs on average when it is called. The column on the right shows who (which types of requests) are calling the Avail code package and how many IOs are used on average when those requests are made.

ECB Owner Name	IOs	Name-Value Pair Breakdown by MsgType	
Avail	15	Avail	17
		Shop	15

- Key features of name-value pair collection
  - Your shop defines the name-value pairs and owner names that will be used to describe a unit of work.
  - Collection will occur at owner name change and ECB exit.
  - Name-value pair collection will report the same counters as ECB owner name and named limit set collections along with the name-value pairs and ECB owner names active at that time.



On ECB Exit, owner name change, and start request processing, ECB counters are stored into this table.

On ECB Exit (or ECB Group Exit), ECB counters are stored into this table.

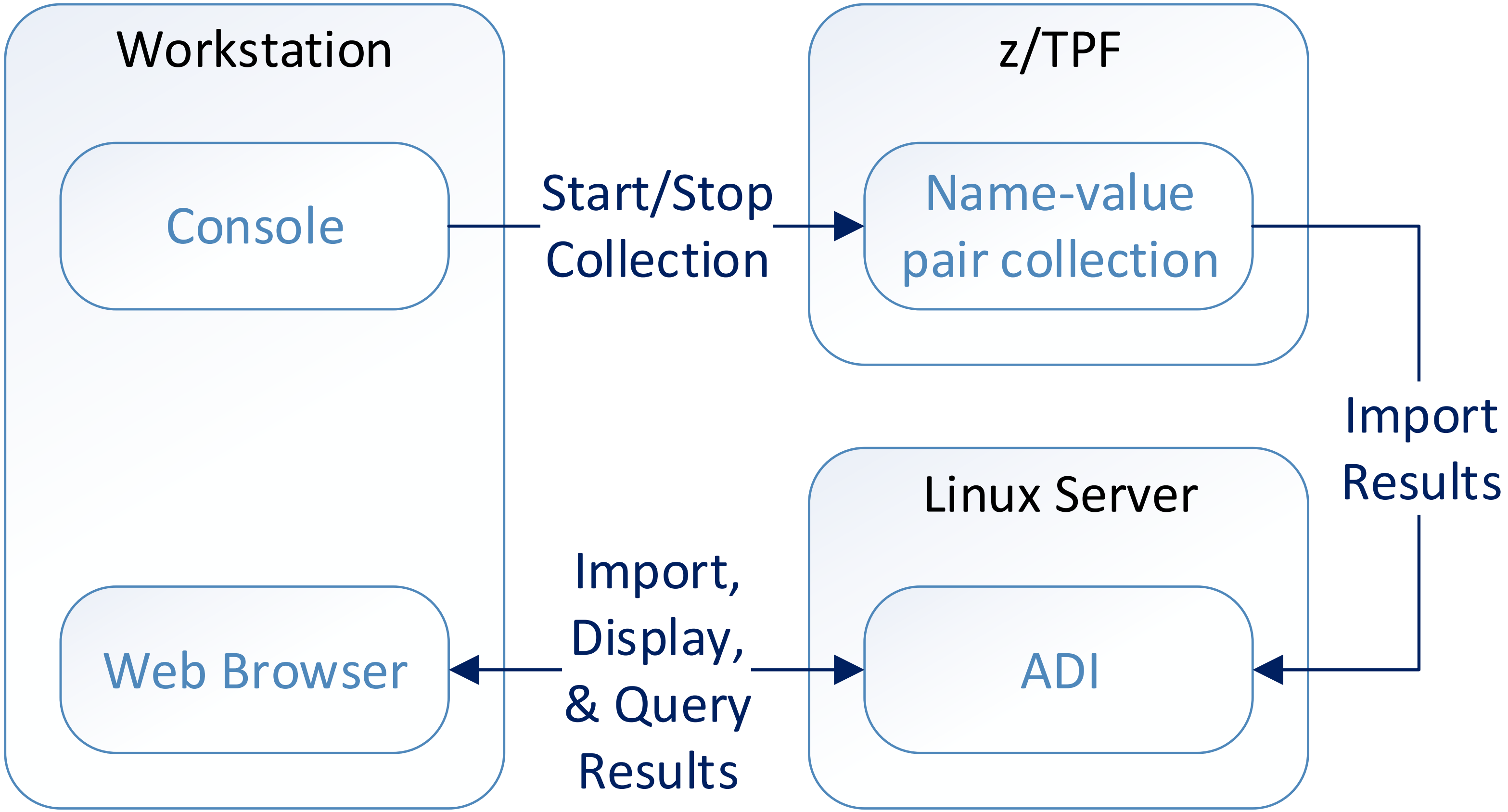
On ECB Exit, owner name change and start request processing, ECB counters will be written into a file with the corresponding *name-value pairs and owner names.*

Name-value pair collection

- Name-value pair collection will be started and stopped from a new utility which can be run during peak.
- Name-value pair collection will be configurable:
  - Write every Nth name-value pair collection entry.
  - Write only for ECBs with the matching name-value pairs.
  - Write only on ECB Exit for lower cost collection mechanism.

- Name-value pair collection results will be imported into the ADI server offline.
- ADI analysis of the name-value pair collection results will build a set of user requested tables to make the data consumable.

Name-value pair collection



Workstation

Console

Web Browser

Start/Stop Collection

z/TPF

Name-value pair collection

Import Results

Linux Server

ADI

Import, Display, & Query Results

Name-value pair collection

Use Case

- Problem statement:
  - The number of IOs has unexpectedly increased by 10% from one day’s peak to the next day’s peak. The increase in IOs can not be attributed to any code change, weather event or etc.

- As-is scenario
  - With the existing tools (such as ZMOWN, software profiler and etc), the increase in IOs has been attributed to the shopping package. However, the current tools do not indicate what has caused the change.
  - Given experience, it is believed that a given customer has changed the input to their requests. However, it is difficult to identify which customer has caused the problem.

- To Be Scenario
  - Name-value pair collection is captured and imported into ADI.
  - Today’s collection is compared to previous collections.
  - Since name-value pair collection results are broken down by customer, it quickly becomes clear which customer has caused the increase in IO usage.
  - The combined view of name-value pair and owner name collection data allows you to also see which code package is using more resources.

- For example, assume a user has defined a report that is broken down by message type, message sub type, and customer for CPU utilization, IOs and heap usage.
- Assume highlighting shows the likely culprit of IO increase given historical data.

Name-value pair collection – Use Case

Msg Type	% of Traffic	Sub Type	% of Msg Type	Customer	% of Sub Type	CPU Util (avg - mils)	IOs (avg)	Δ (avg)	%Δ (avg)	Heap (avg KB)
+Avail	11					8	40	0	0	.5
+Book	4					12	60	0	0	100
-Shop	50					30	12.06	3.24	37	2000
		-Air	90			31	12.60	3.60	40	2000
				MonsterSite	60	35	9	6	200	1900
				BigTravel	35	29	12	0	0	2400
				Mom&Pop	5	18	6	0	0	3500
		-Hotel	8			20	8	0	0	500
				MonsterSite	60	20	8	0	0	500
...										

- Note that the format shown is conceptual.

Name-value pair collection – Use Case

- The user clicks on the highlighted entry and chooses to view a breakdown based on owner names to understand which code package has increased it’s IO usage in light of the historical data.

Msg Type	% of Traffic	Sub Type	% of Msg Type	Customer	% of Sub Type	CPU Util (avg - mils)	IOs (avg)	Δ (avg)	%Δ (avg)	Heap (avg KB)
Shop	50	Air	90	MonsterSite	60	35	9	6	200	1900

Owner Hi	Owner Mid	Owner Low	CPU Util (avg - mils)	IOs (avg)	Δ (avg)	%Δ (avg)	Heap (avg KB)
Avail	Air	Domestic	7	1	0	0	50
Pricing	Air	Domestic	4	0	0	0	100
Shopping	Air	coOrd	3	7	6	600	50
Shopping	Air	Domestic	21	1	0	0	1700

- Note that the format shown is conceptual.

Business Value

- IBM Application Delivery Intelligence (ADI) is an Enterprise continuous integration analytics dashboard and optimization software with focus on application understanding, performance and quality metrics and trends.
- The name-value pair collection feature will be implemented in ADI so that:
  - A coverage programmer can use name-value pair collection to gain new insights into system resource usage and identify the source of problems in as little as 1/20 of the time previously required.

Call to action

- Update your code today
  - to set name-value pairs when a request is received to describe the request to be processed, the origin of the request, and provide unique ids that identify units of work.
  - to set owner names at the primary interface points to code packages.
  - to leverage the immediate benefits of using name-value pairs to pass data, owner name collection (ZMOWN) and etc.
  - to prepare to leverage name-value pair collection once it is available.

More Information

- Future discussions of the 2016 development efforts will continue with Sponsor Users.
- Please let us know if you are interested in being a Sponsor User.

- Education session on Wednesday will further discuss
  - Existing collection mechanisms.
  - Collection mechanism usage guidelines to provide guidelines and examples of how to code ECB owner names and name value pairs in preparation for name-value pair collection and other future changes anticipated for the ECB owner name mechanism.

More Information

Thank you!

Questions or comments?

Trademarks

- IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines 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 on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Notes

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.