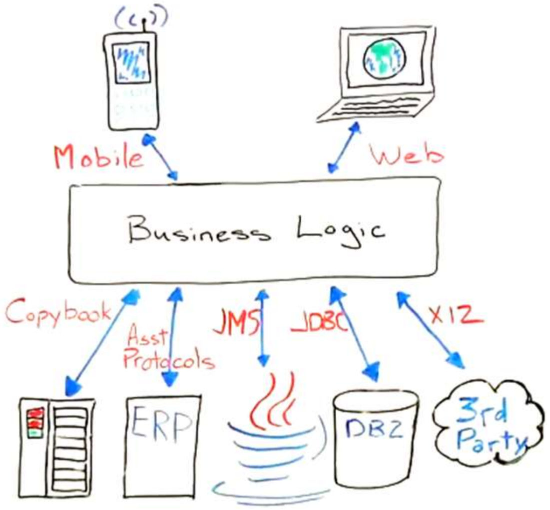
IBM
Enterprise DevOps
Continuous Testing and Virtualization

Mark Garcia

Executive IT Architect

Worldwide Devops Testing and Quality Leader

Multi-tiered Applications are becoming more complex to test



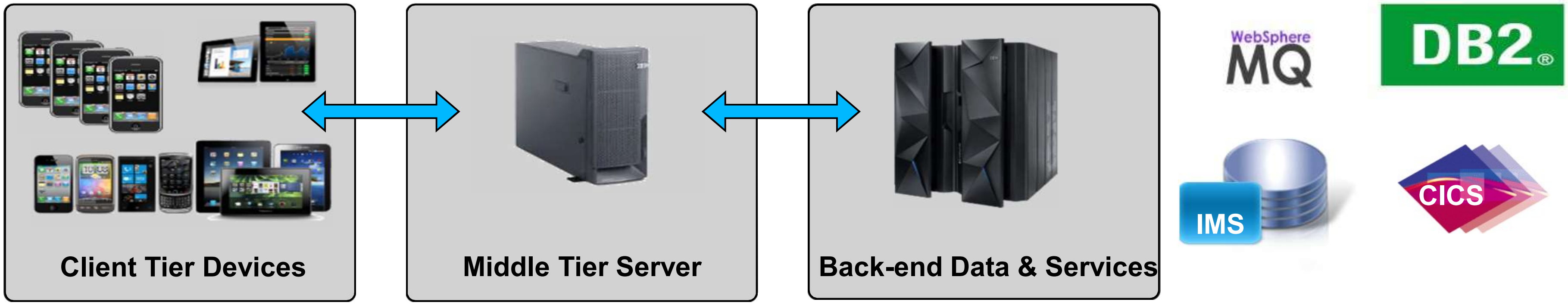
- Increasing expense of standing up complex environments
- Delays caused by lack of system availability
- Third-party services can be especially costly
- Testers spend more time setting up tests than testing

$5-30 million investment

Typically required to build a single test lab for a Fortune 500 company. Most have dozens …

30-50% average time

Testing teams spend on setting up test environments, instead of testing



Client Tier Devices

Middle Tier Server

Back-end Data & Services

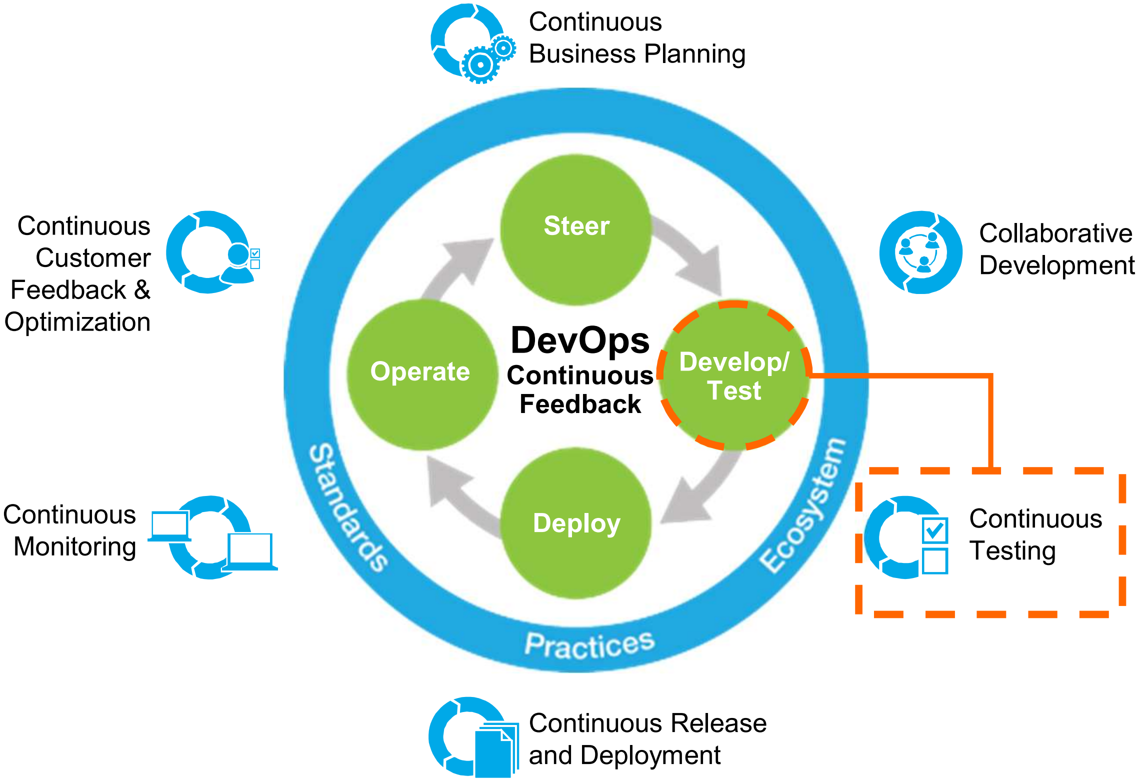
IBM DevOps – Continuous Testing

Adopting continuous testing principles provides early and constant feedback on quality and is a critical success factor for embracing a culture of quality engineering

Accelerate software delivery – *for faster time to value*

Balance speed, cost, quality and risk – *for increased capacity to innovate*

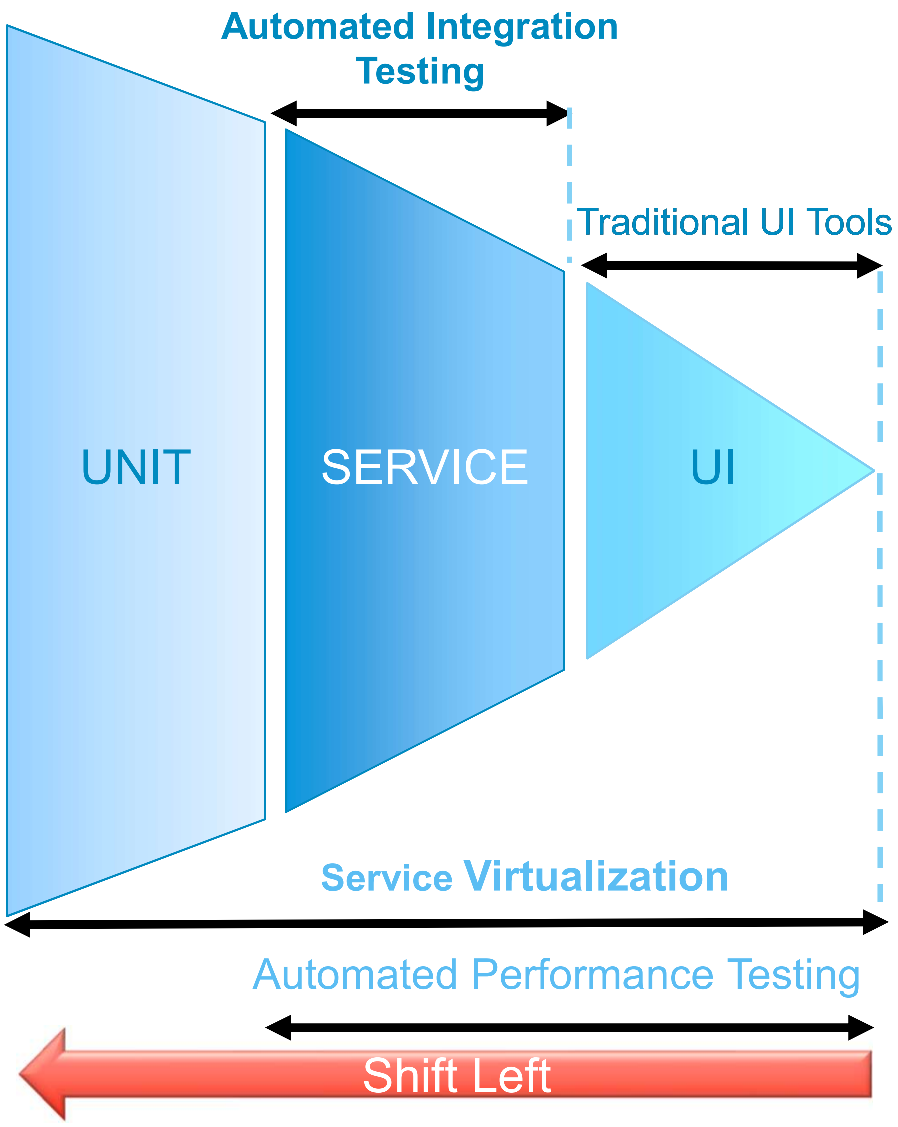
Reduce time to customer feedback – *for improved customer experience*


Continuous Business Planning
Continuous Customer Feedback & Optimization
Collaborative Development
Steer
Operate
DevOps
Continuous Feedback
Develop/ Test
Deploy
Standards
Practices
Ecosystem
Continuous Monitoring
Continuous Testing
Continuous Release and Deployment

Our DevOps POV is resonating with clients and they are delivering **measurable business outcomes with DevOps**

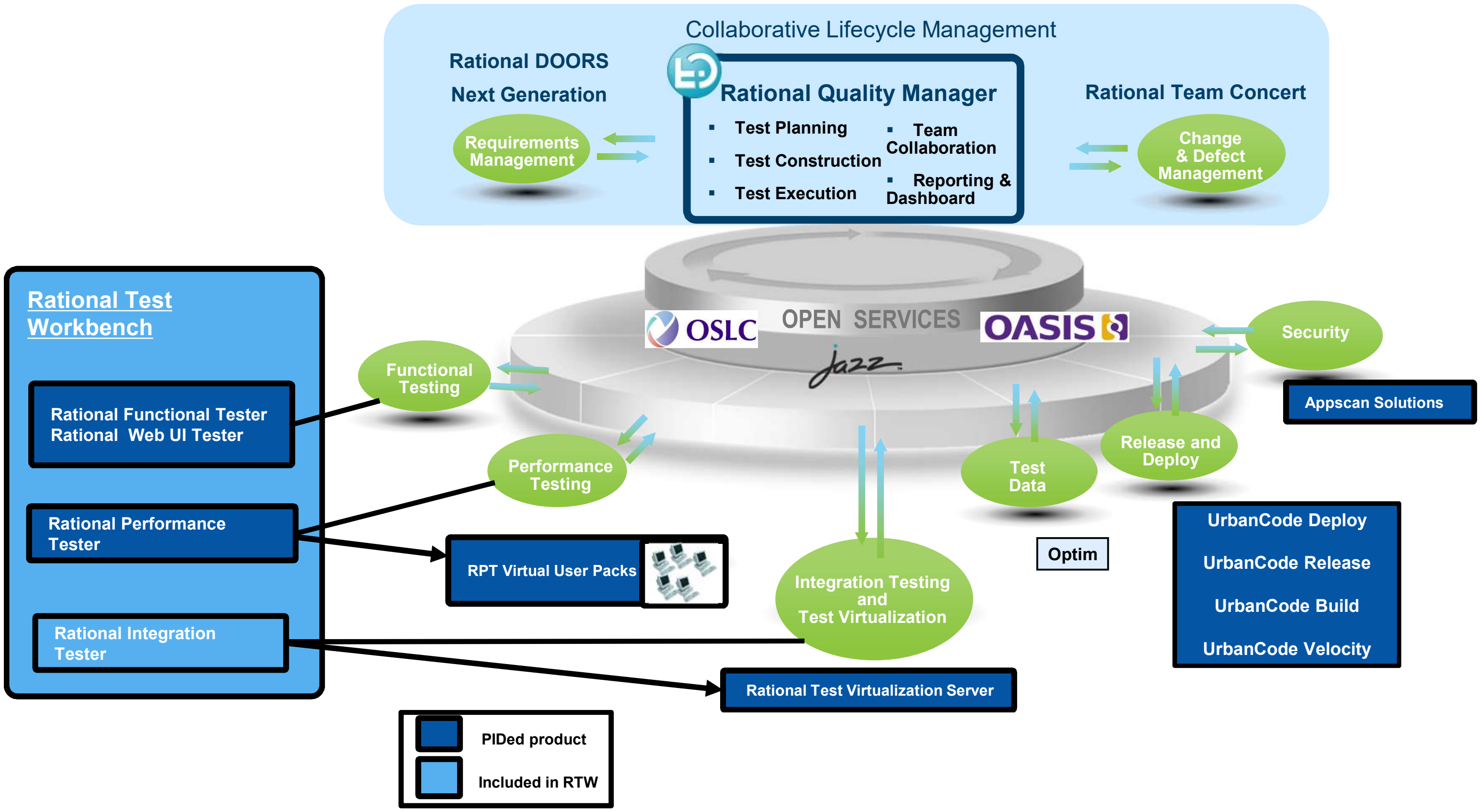
Shift Left

Is your effort directed at your risk?



- Testing only at the UI level provides little confidence for deployment of complex distributed systems
- Testing must be shifted left to focus on the highest areas of risk
- Automated integration massively increases testing coverage and confidence
- Service virtualization can be applied at all phases – even allowing integration testing to begin in parallel with Unit Testing.

IBM DevOps



Collaborative Lifecycle Management

Rational DOORS Next Generation

Requirements Management

Rational Quality Manager

- Test Planning
- Test Construction
- Test Execution
- Team Collaboration
- Reporting & Dashboard

Rational Team Concert

Change & Defect Management

OSLC OPEN SERVICES OASIS

Jazz

Rational Test Workbench

Rational Functional Tester
Rational Web UI Tester

Rational Performance Tester

Rational Integration Tester

Functional Testing

Performance Testing

RPT Virtual User Packs

Integration Testing and Test Virtualization

Rational Test Virtualization Server

Test Data

Optim

Release and Deploy

UrbanCode Deploy

UrbanCode Release

UrbanCode Build

UrbanCode Velocity

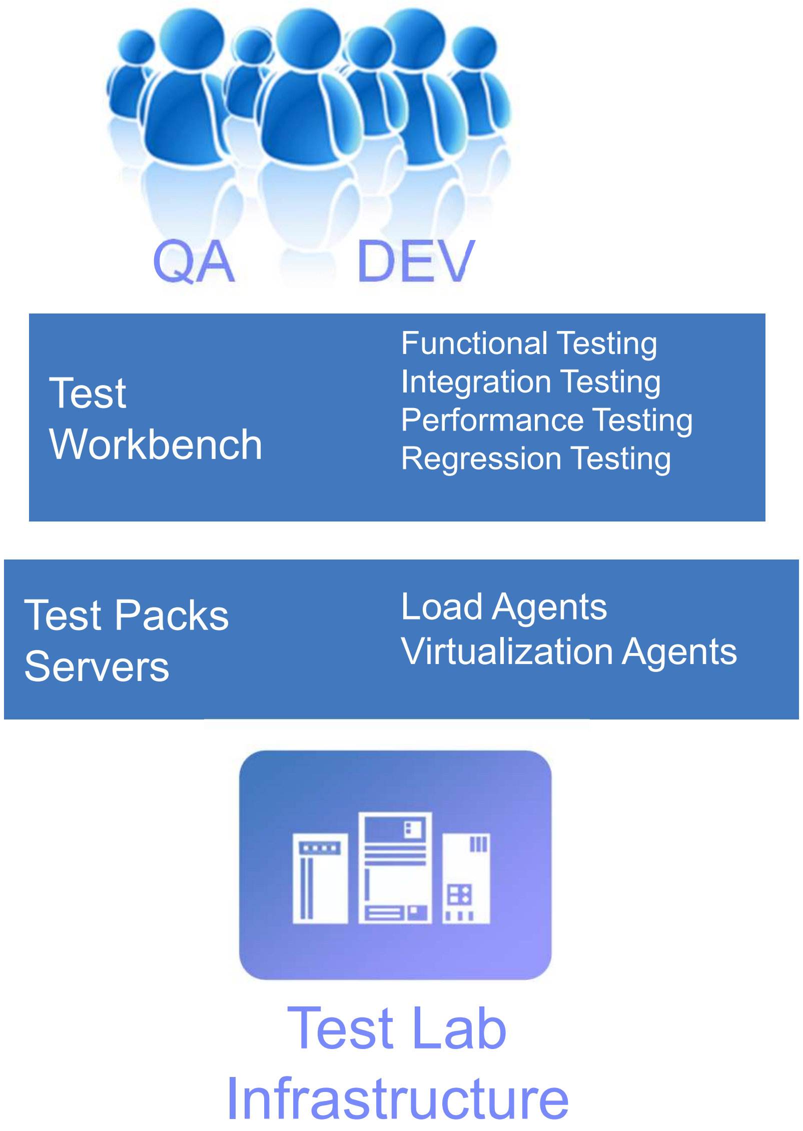
Security

Appscan Solutions

🟦	PIDed product
🟦	Included in RTW

IBM Continuous Testing Solution

- **IBM Rational Test Workbench**
  - ➢ Enable functional, performance and integration testing throughout your project lifecycle
  - ➢ With a scriptless, wizard-driven test authoring environment and support for more than 70 technologies and protocols
- **IBM Rational Test Virtualization Server**
  - ➢ Model real system behavior to eliminate test dependencies and simulate production,
  - ➢ Accelerate testing and reduce the setup and infrastructure costs of traditional testing environments.
- **IBM Rational Performance Tester Virtual Test Pack**
  - ➢ Save time and effort by reusing your functional integration tests for performance testing. New integration between RIT and RPT
  - ➢ Combine your tests to model real-world scenarios and assess the impact of load on your integrated infrastructure.



QA DEV

Test Workbench

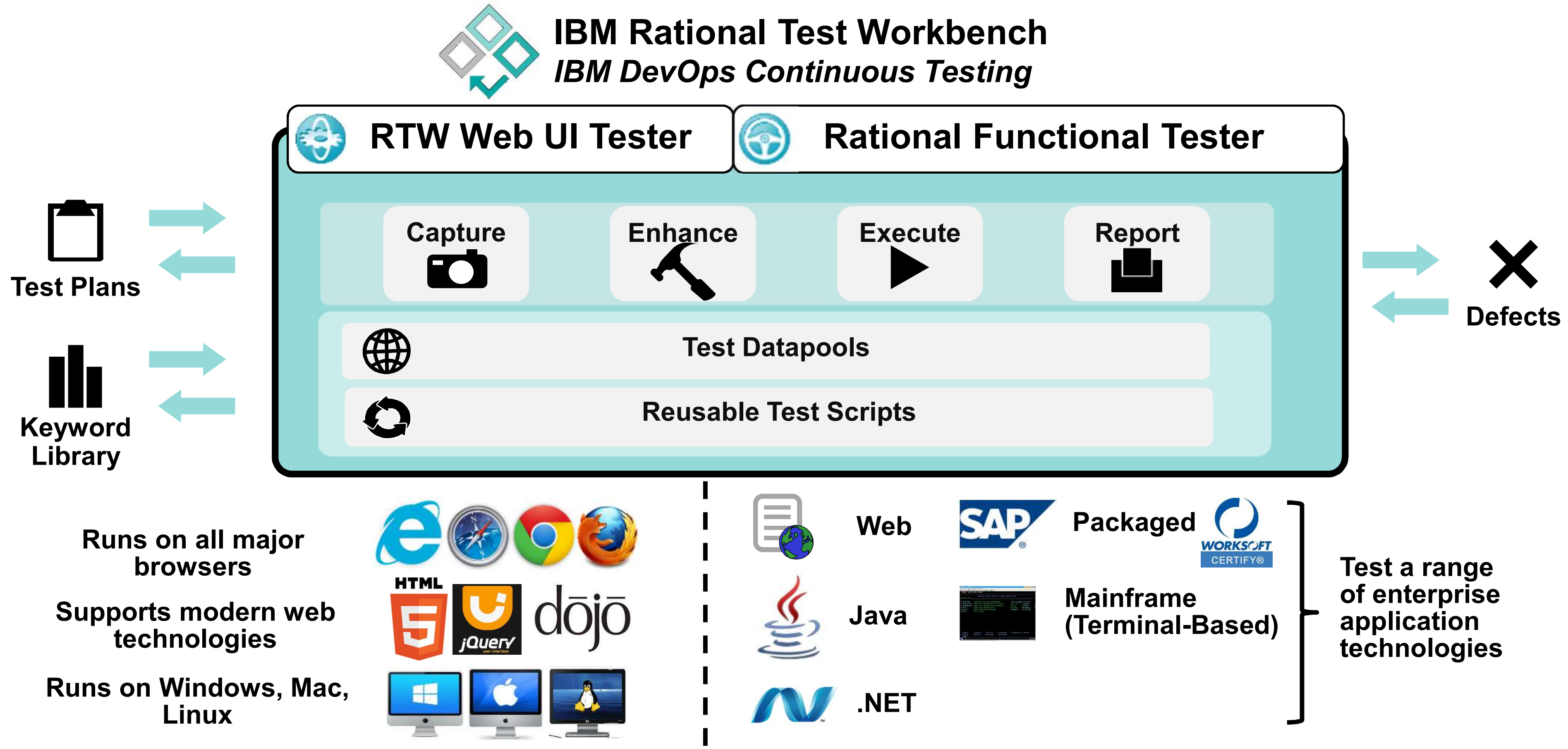
Functional Testing
Integration Testing
Performance Testing
Regression Testing

Test Packs
Servers

Load Agents
Virtualization Agents

Test Lab Infrastructure

Automated GUI Testing with IBM Rational Test Workbench



IBM Rational Test Workbench

IBM DevOps Continuous Testing

RTW Web UI Tester

Rational Functional Tester

Capture

Enhance

Execute

Report

Test Datapools

Reusable Test Scripts

Test Plans

Keyword Library

Defects

Runs on all major browsers

Supports modern web technologies

Runs on Windows, Mac, Linux

Web

Java

.NET

Packaged

Mainframe (Terminal-Based)

Test a range of enterprise application technologies

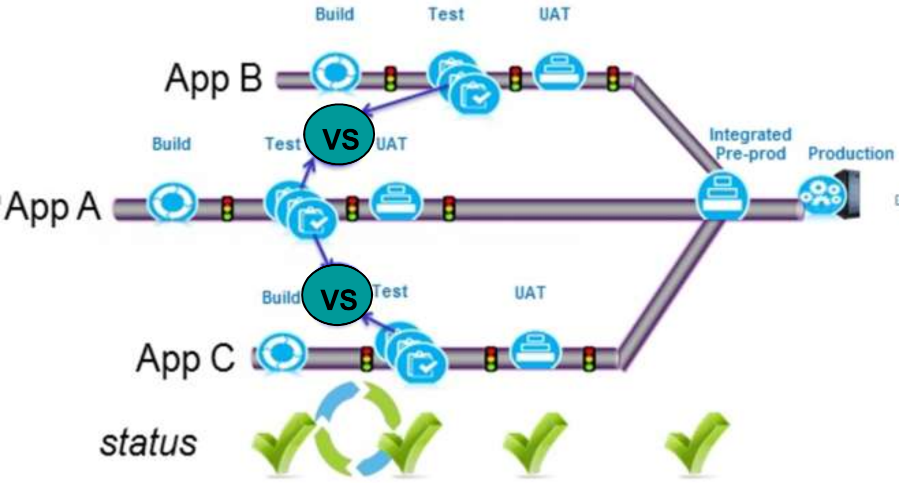
Automated Integration Testing with IBM Rational Test Workbench

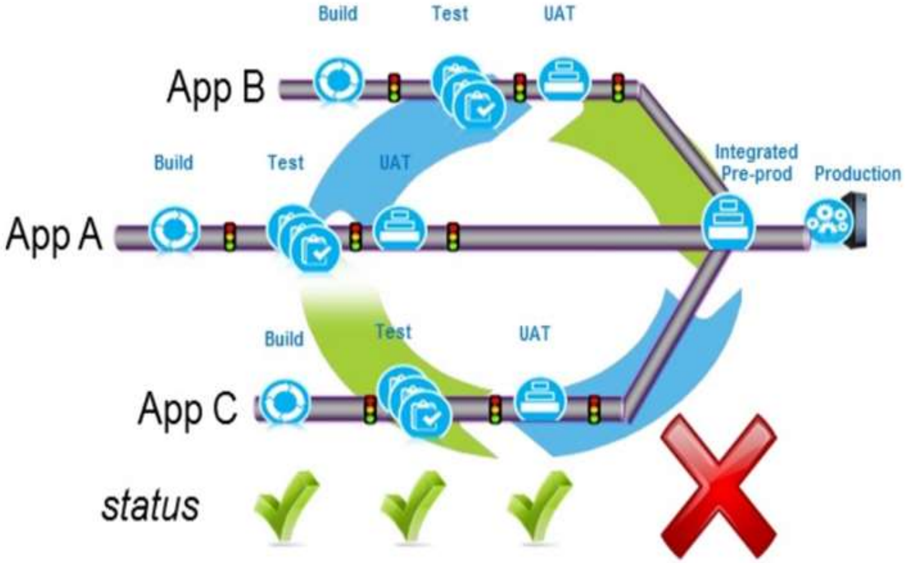
Shifts integration testing to the left!

Challenges

Dependencies between applications traditionally means that integration testing is done:

- Late, putting release schedule at risk
- Bugs detected later, costing more to fix
- Overall longer release cycle





IBM Rational Integration Tester

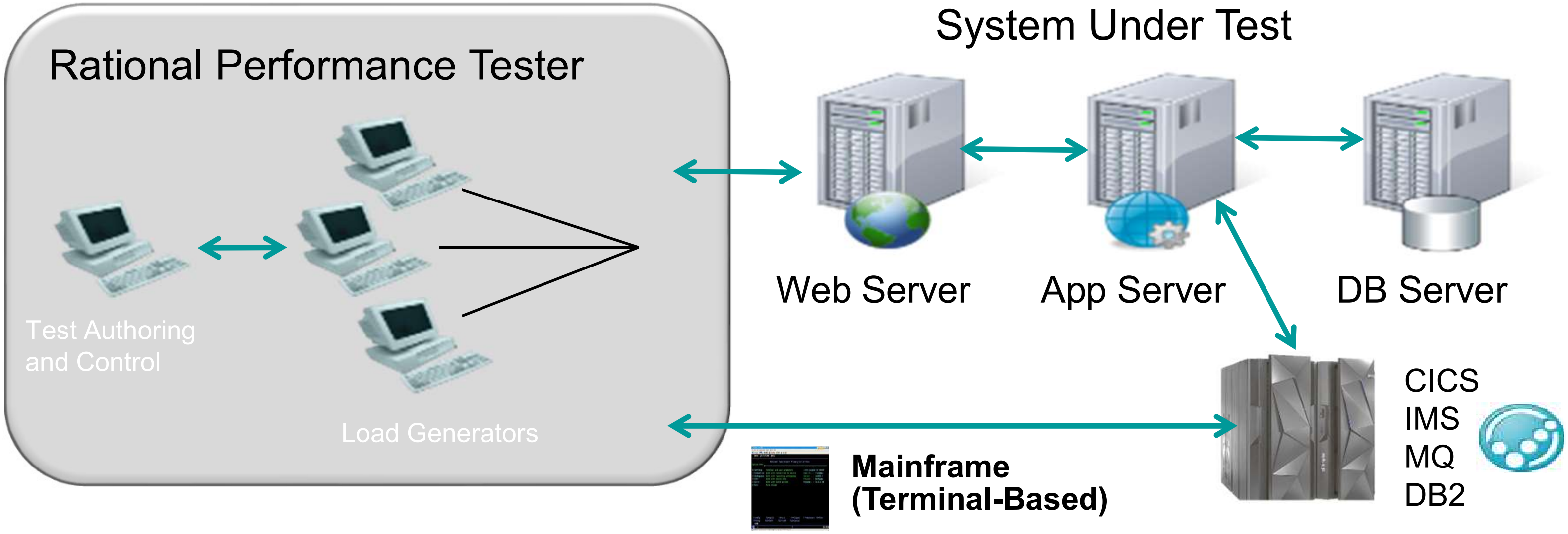
- provides a scripting-free environment for developing automated tests
- build tests by recording existing system behaviors or from requirements, by entering the data to send and the data that is expected in return
- create virtual services to remove test dependencies and enable earlier testing.

Automated Performance Testing with IBM Rational Test Workbench

Test automation for the novice and the professional

Identify and manage the risk of application performance failures

- **Capture and track** Business SLAs in your test plans
- **Create realistic user workloads** that exercise the key business transactions
- Automatically **deploy the environment and schedule execution** of your workload
- **Identify the root cause** of performance problems in hardware and software
- **Communicate results** and areas of non-compliance to the stakeholders
- includes a test recorder, automated data manipulation capabilities, flexible scheduling, reporting and a scalable execution engine.



Rational Performance Tester

Test Authoring and Control

Load Generators

System Under Test

Web Server

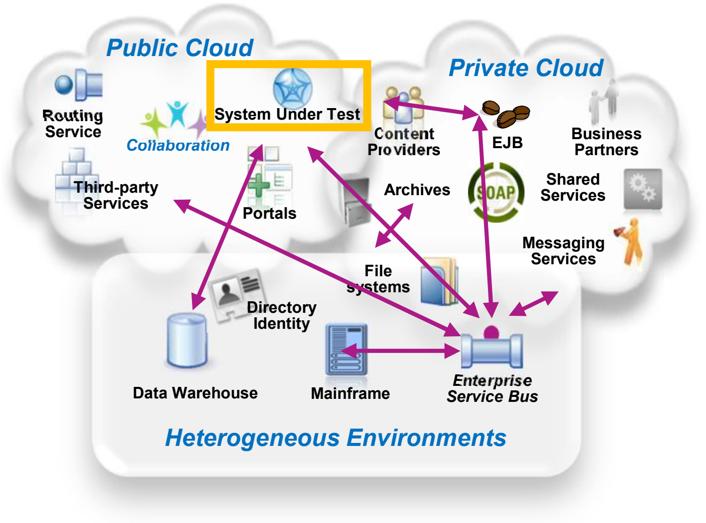
App Server

DB Server

CICS
IMS
MQ
DB2

Mainframe (Terminal-Based)

Service Virtualization : What is it? How can it help?



System *dependencies* are a key challenge in setting up test environments:

- ***Unavailable/inaccessible***: Testing is constrained due to production schedules, security restrictions, contention between teams, or because they are still under development
- ***Costly 3rd party access fees***: Developing or testing against Cloud-based or other shared services can result in costly usage fees
- ***Impractical hardware-based virtualization***: Systems are either too difficult (mainframes) or remote (third-party services) to replicate via traditional hardware-based virtualization approaches

IBM Rational Test Workbench & IBM Rational Test Virtualization Server

Key Features and Benefits

- Enables Shift Left (start testing earlier) and Continuous Testing (Developers and Testers) to improve quality at lower costs
- Reduces bottlenecks and wait times - Virtualized services can be used for testing when real services are unavailable
- Reduces infrastructure requirements – virtualized services run in a reduced footprint
- Allows testing of integration and complex scenarios earlier in the test cycle
- Virtual environments can be dynamically provisioned, allowing for high capacity, scalability testing scenarios
- Drives automation of test cases

Supported technologies the business depends on

Protocols, message formats, and technologies

Messaging Protocols

- ActiveMQ
- Email (SMTP, IMAP)
- Files
- FTP/S
- HTTP/S
- WebSphere MQ
- Java RMI
- JMS
- SAP IDoc, BAPI, RFC
- Software AG's IB & IS
- Sonic MQ
- TCP/IP
- TIBCO Rendezvous, Smart Sockets & EMS
- CICS TG
- CICS IPIC
- IMS Connect
- MongoDB
- RabbitMQ

Message Formats

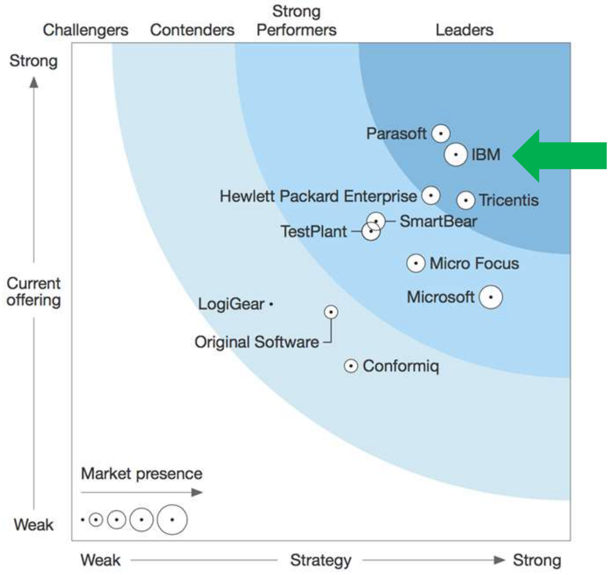
- .Net Objects
- .NET WCF
- Bytes
- COBOL Copybook
- ebXML
- ASC X12
- UN/EDIFACT
- DFDL
- Fixed Width
- Java Objects
- MIME
- SOAP
- Software AG Broker Docs
- SWIFT
- CHIPS
- FedWire
- TIBCO ActiveEnterprise
- XML (DTD, XSD, WSDL)
- ISO8583
- SWAGGER
- RAML
- Custom

Technologies

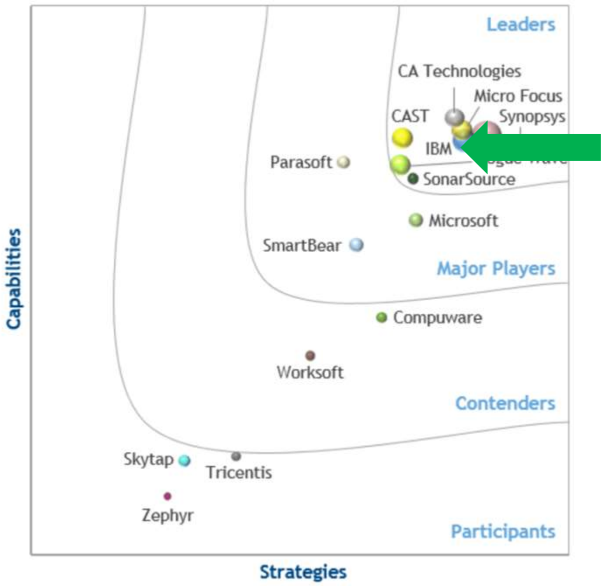
- CentraSite
- Oracle Fusion
- SCA Domain
- Software AG IS, Broker, BPMS
- WebSphere Application Server
- WebSphere Message Broker
- WebSphere MQ
- Websphere SI Bus
- Sonic ESB
- System z (CICS and IMS)
- TIBCO ActiveMatrix
- UDDI
- Web Services
- WebSphere RR
- WSDL
- BPM
- Databases
- Log Files

IBM is a Market Leader in Testing

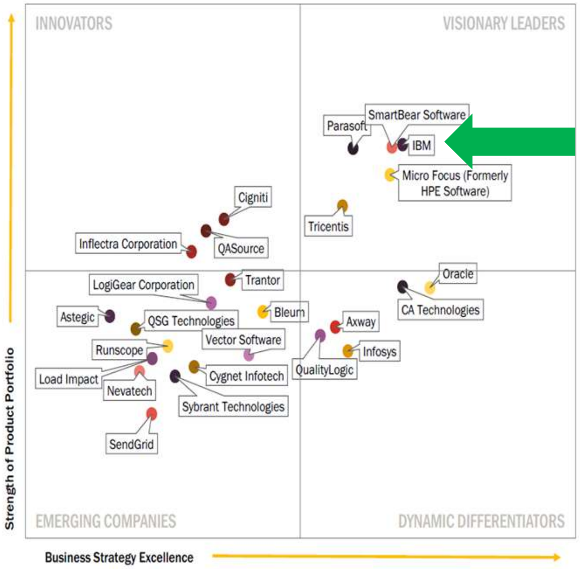
2017 Forester Wave for Modern Application Functional Test Automation Tools



IDC Worldwide Software Quality Analysis and Measurement 2017–2018 Vendor Assessment



Markets and markets API testing market Competitive leadership 2017



The ROI of Service Virtualization

The results are real!

A major European bank implemented service virtualization and automated integration testing as part of an Agile transformation to enable continuous testing. As a result of IBM capabilities, their system integration test (SIT) process was reduced from 3 weeks down to four hours!

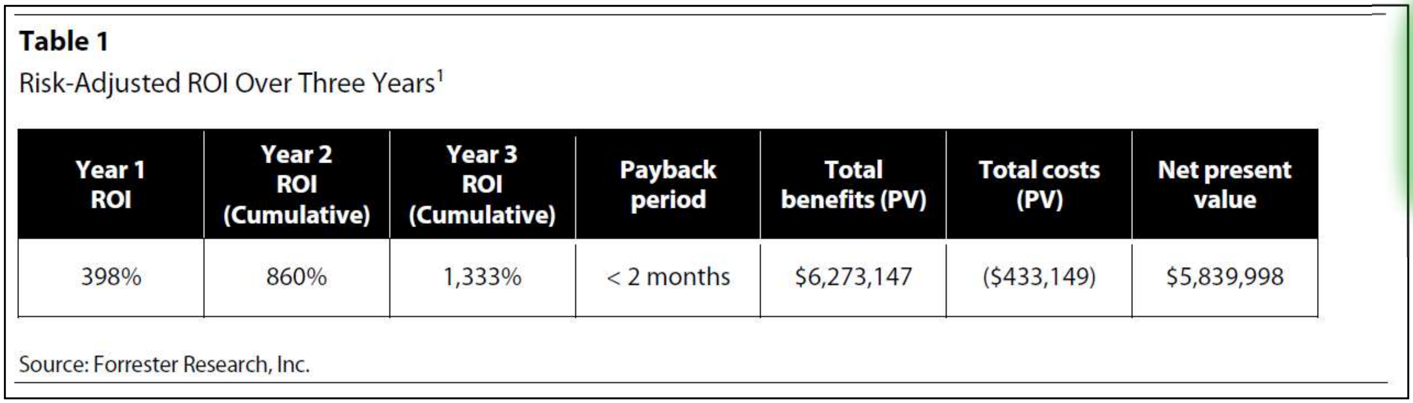


Table 1

Risk-Adjusted ROI Over Three Years[1]

Year 1 ROI	Year 2 ROI (Cumulative)	Year 3 ROI (Cumulative)	Payback period	Total benefits (PV)	Total costs (PV)	Net present value
398%	860%	1,333%	< 2 months	$6,273,147	($433,149)	$5,839,998

Source: Forrester Research, Inc.

13x ROI in 3 years!

The Total Economic Impact™ of IBM Rational Service Virtualization and Test Automation Solutions, a July 2014 commissioned study conducted by Forrester Consulting on behalf of IBM

3 Year Benefits:

- **Increased project delivery capacity 100%** (40 projects to 80 projects)
- **Avoided hiring three additional FTE's**, saving almost $600k.
- **Reduced incidents found in production from 2.5 to .3 per project**, saving $3 million
- **Identified defects earlier** and freed the team up to focus on new capabilities, saving over $560k
- **Reduced consulting and third party testing fees** with better test automation, saving $408k
- **Eliminated the need for additional hardware and labor** to set up test environments, saving $700k
- **Generated new revenue for the business** – estimated at tens of millions!

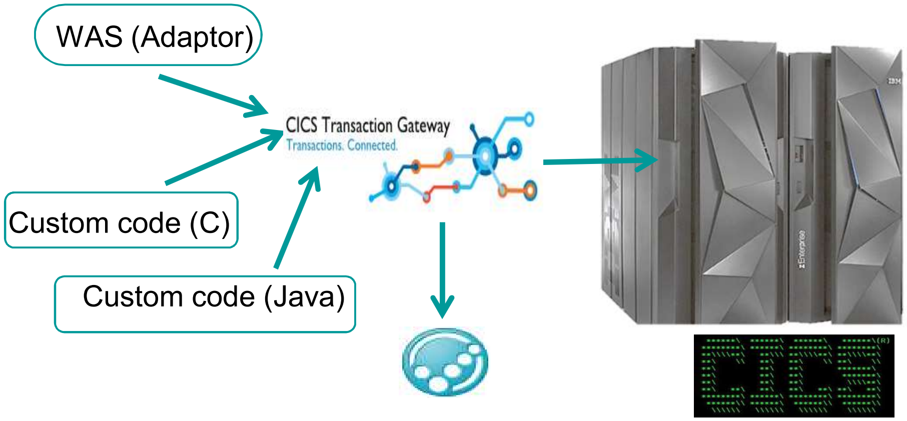
Mainframe
Test and Virtualization
Capabilities

CICS – Transaction Gateway (CTG)

Case Study - Pain Points

- Insufficient number of CICS regions to thoroughly test their programs on the mainframe
- Testing cycles were consistently delayed due to bottlenecks caused by multiple testing and development teams vying for the same CICS region
- Costs to testing mainframe applications were spiraling out of control.

Test Automation / Service Virtualization



Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

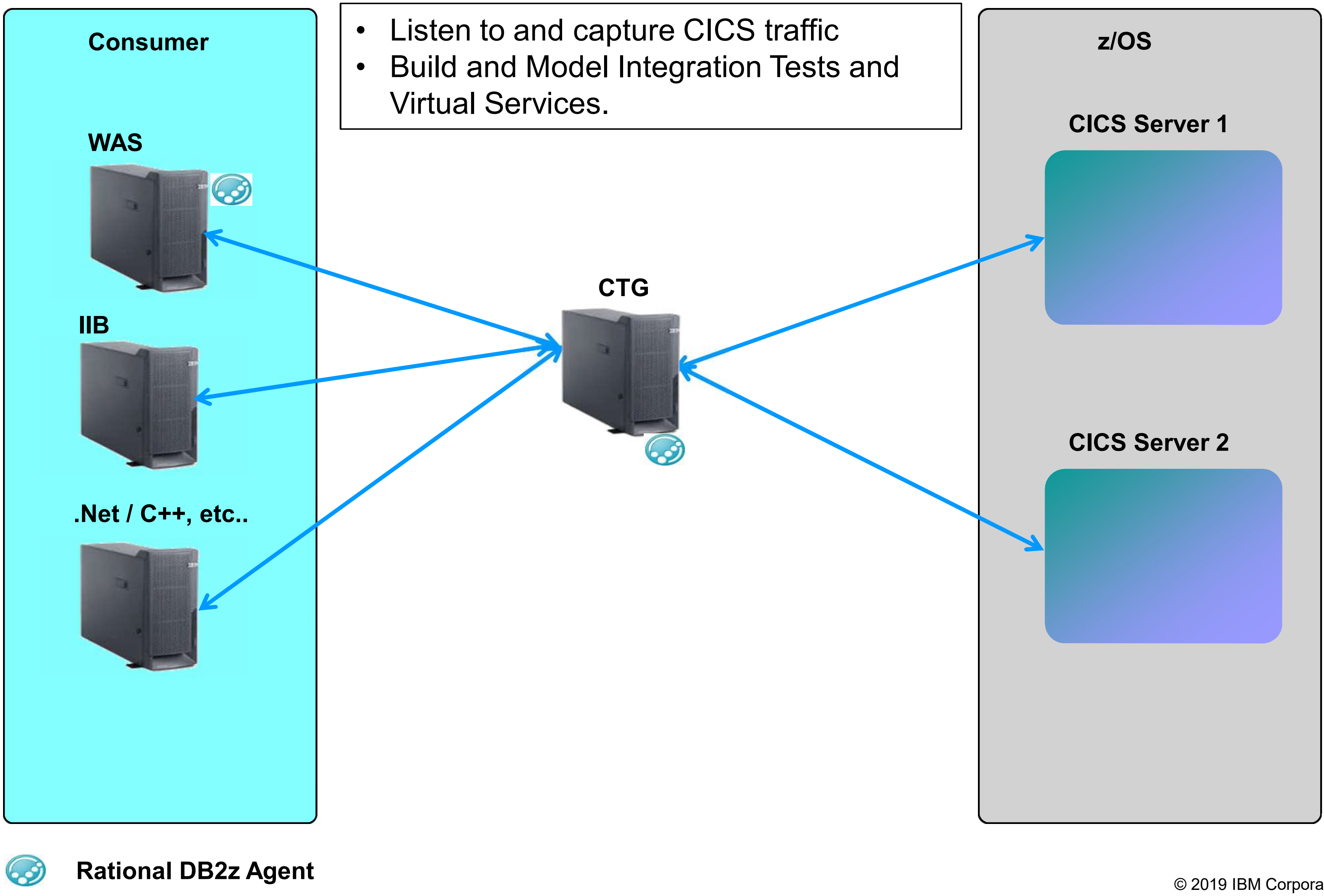
- Leverage CICS CTG virtualization capability to make automation test script and build up stubs to virtualize dependency to CICS region
- Leverage DB2/z virtualization capability to virtualize dependency to real DB2 database on Z.

Benefits / Motivation

- Reduced MIP Charges
- Reduced DASD Charges
- Decoupling from mainframe during development and test
- Reduction in scheduling constraints
- Shortening the ratio between defect identification and resolution.

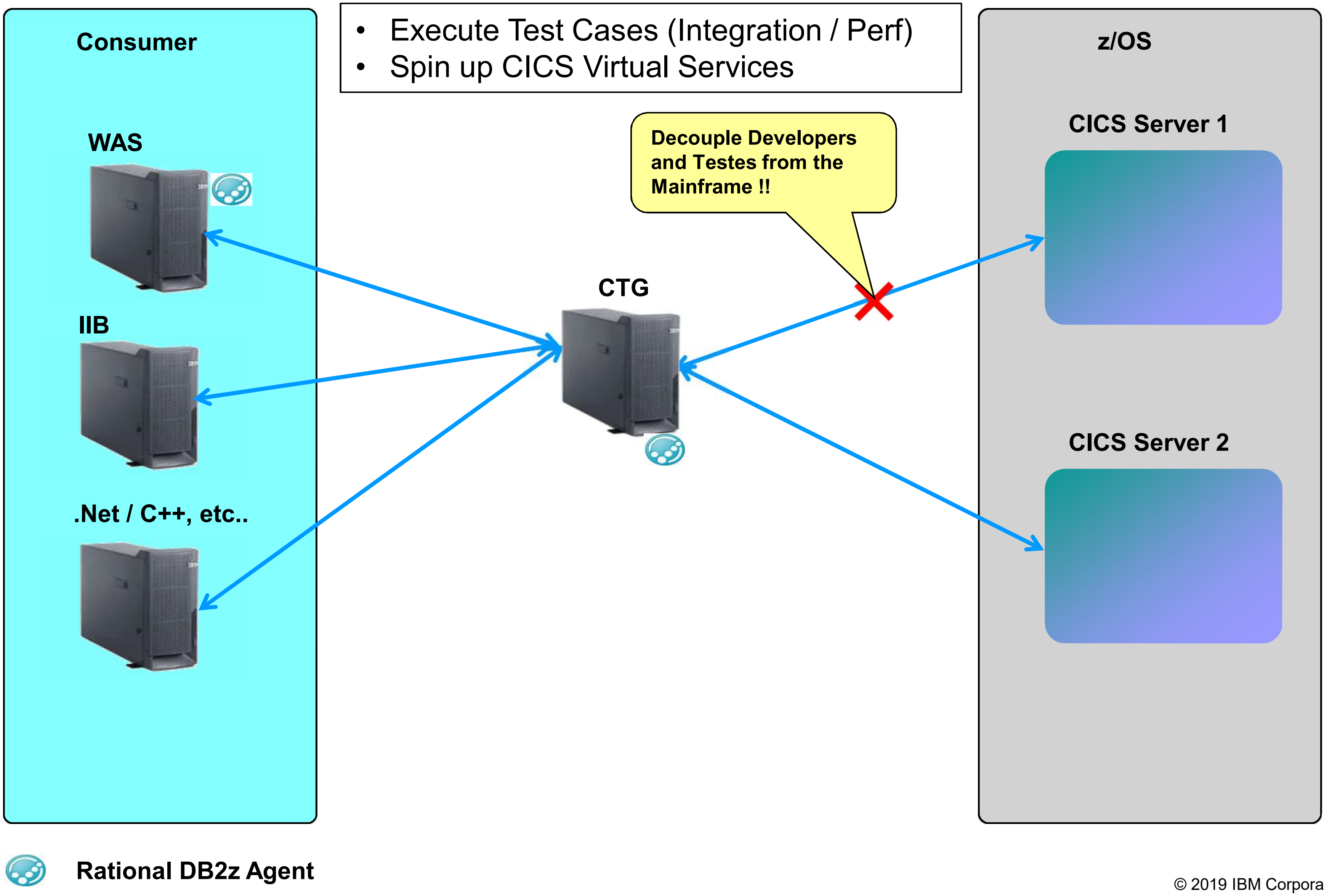
CICS CTG High Level Architecture Diagram

- Listen to and capture CICS traffic
- Build and Model Integration Tests and Virtual Services.



CICS CTG High Level Architecture Diagram

- Execute Test Cases (Integration / Perf)
- Spin up CICS Virtual Services

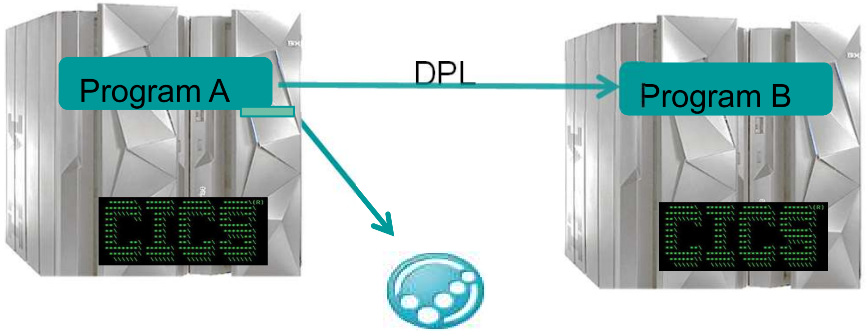


CICS – Transaction Server DPL

Test Automation / Service Virtualization

Case Study - Pain Points

- Development and testing bottlenecks due to the lack of multiple CICS regions
- Lack of resources required to test multiple versions of the code
- Testing future requirement changes are difficult due to the fact of limited region resources.



Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

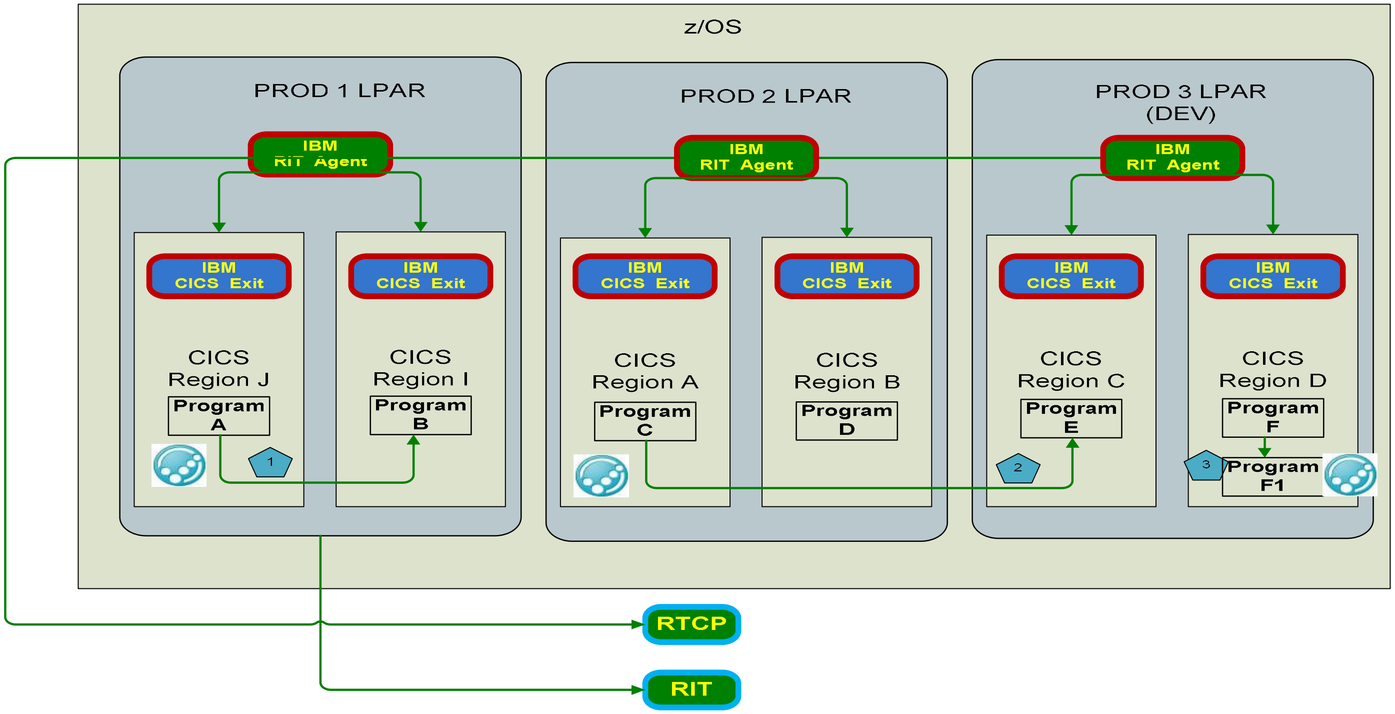
- Leverage CICS DPL virtualization capability to provision “virtual” CICS regions
- Leveraging virtual CICS transactions to model additional test scenarios.
- Leveraging virtual CICS transactions to provide parallel development and testing

Benefits / Motivation

- Provisioning virtual, simulated CICS regions at a fraction of the cost
- Reduction in scheduling constraints
- Ability to model and change linked programs (COMMAREA) on the fly
- Shortening the ratio between defect identification and resolution.

IMS – CICS – DPL Virtualization Scenarios

CICS Virtualization Distributed Program Link(DPL)



Rational CICS DPL Agent - Résides on invoking CICS Région

IMS – IMS Connect

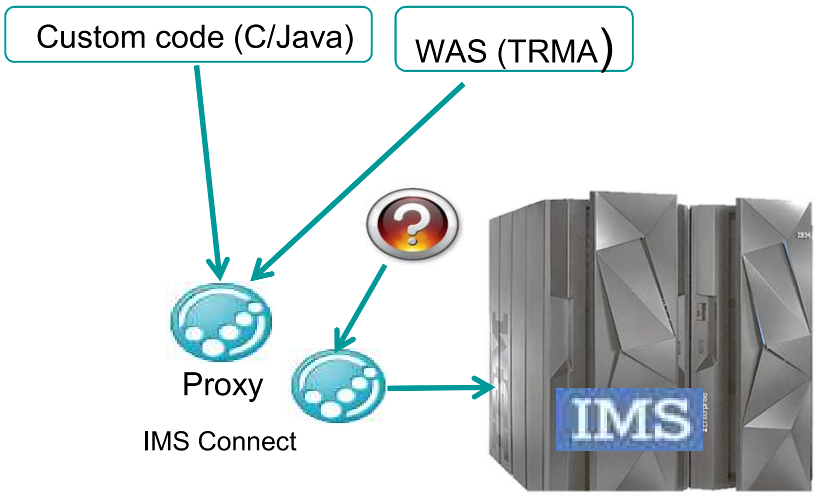
Case Study - Pain Points

- Testing cycles were consistently delayed due to bottlenecks caused by multiple testing and development teams vying for the same IMS region
- Costs to testing mainframe applications were spiraling out of control.
- Lack of quality test data to test IMS existing and future transactions
- Scheduling constraints for mainframe resources

Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

- Leverage IMS Connect recorder capability to make test scrips creation easier
- Leverage IMS Connect virtualization capability to build out additional IMS transactions
- Create "virtual" IMS regions to provide parallel development and testing for all teams

Test Automation / Service Virtualization



Custom code (C/Java)

WAS (TRMA)

Proxy

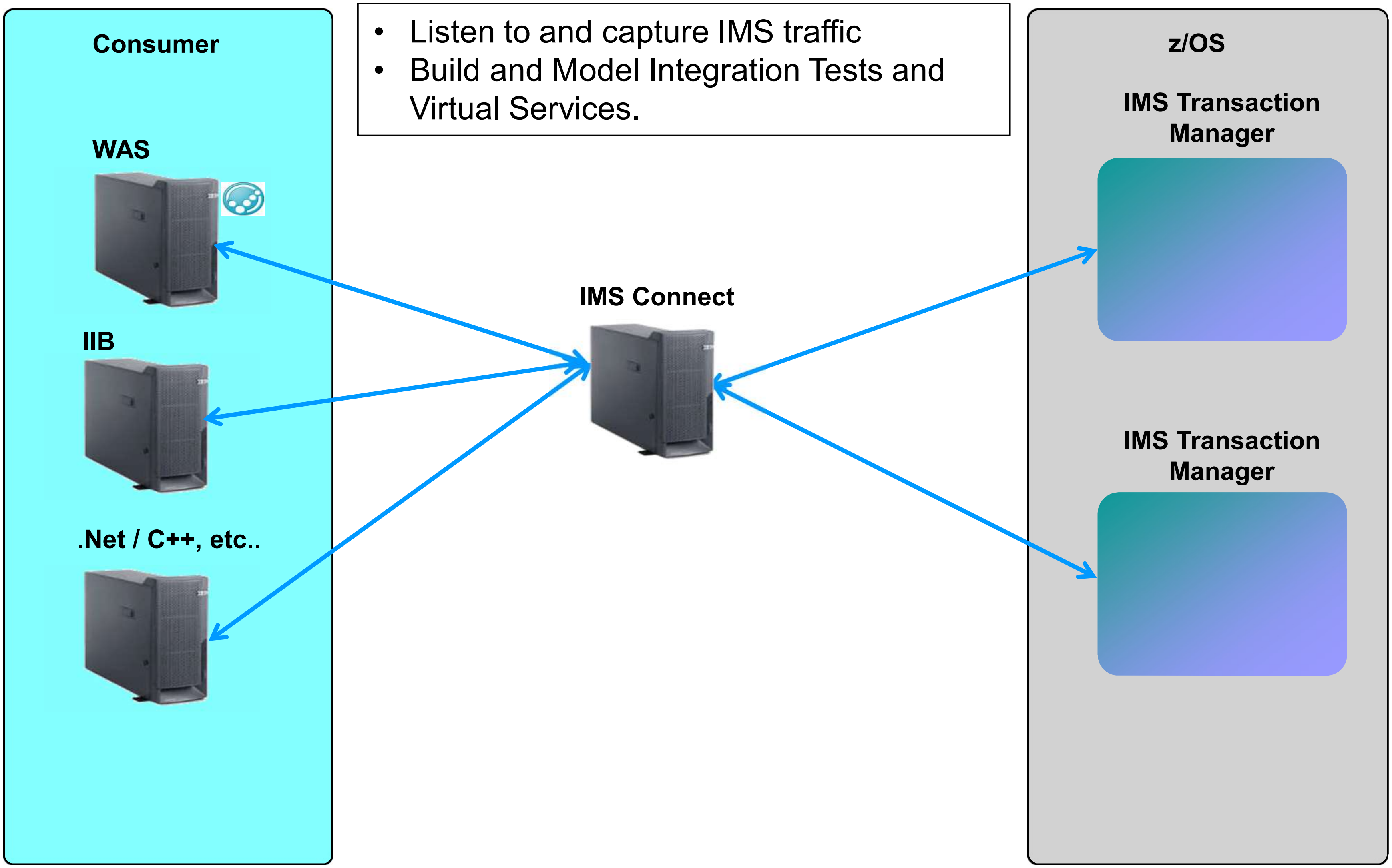
IMS Connect

Benefits / Motivation

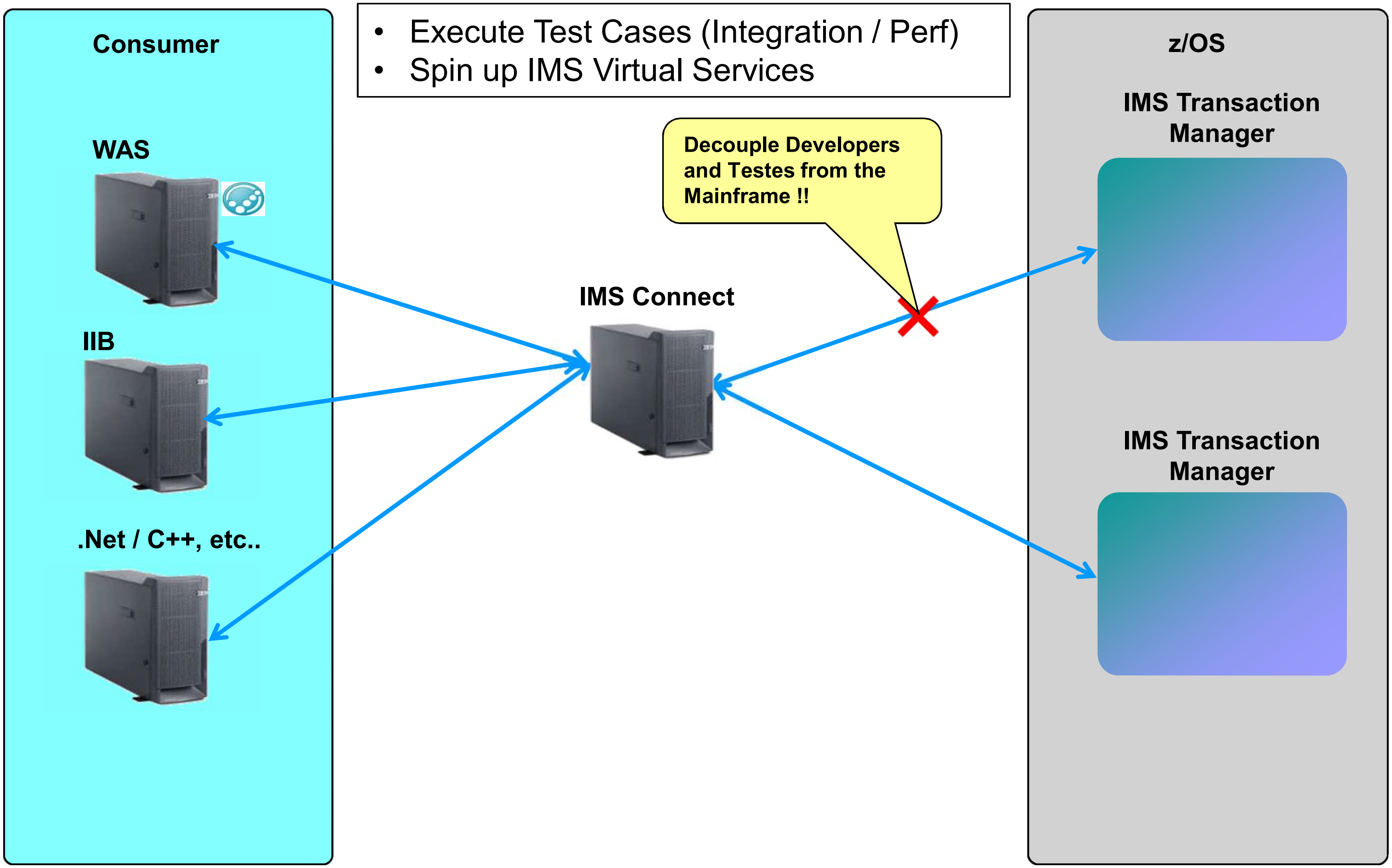
- Reduced MIP Charges
- Reduced DASD Charges
- Decoupling from mainframe during development and test
- Reduction in scheduling constraints
- Shortening the ratio between defect identification and resolution.

IMS Connect High Level Architecture Diagram

- Listen to and capture IMS traffic
- Build and Model Integration Tests and Virtual Services.



IMS Connect High Level Architecture Diagram

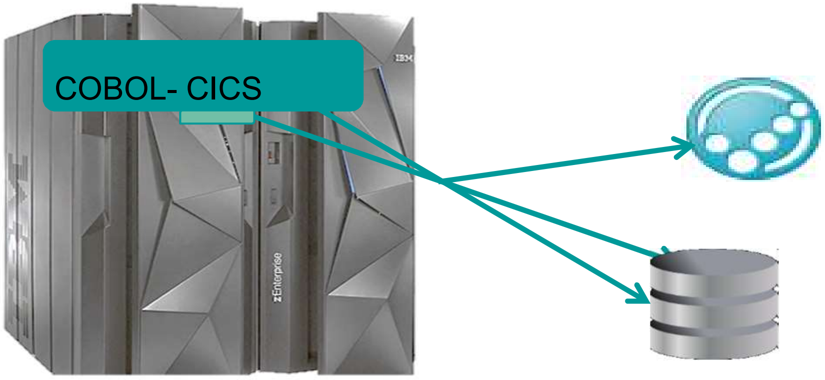


- Execute Test Cases (Integration / Perf)
- Spin up IMS Virtual Services

Decouple Developers and Testes from the Mainframe !!

COBOL – CICS – DB2

Database Virtualization



Case Study - Pain Points

- Insufficient number of CICS regions to thoroughly test their programs on the mainframe
- Testing cycles were consistently delayed due to bottlenecks caused by multiple testing and development teams vying for the same CICS region
- Lack of quality test data
- Test cases go untested, insufficient test data (creating test data is a timely tasks)

Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

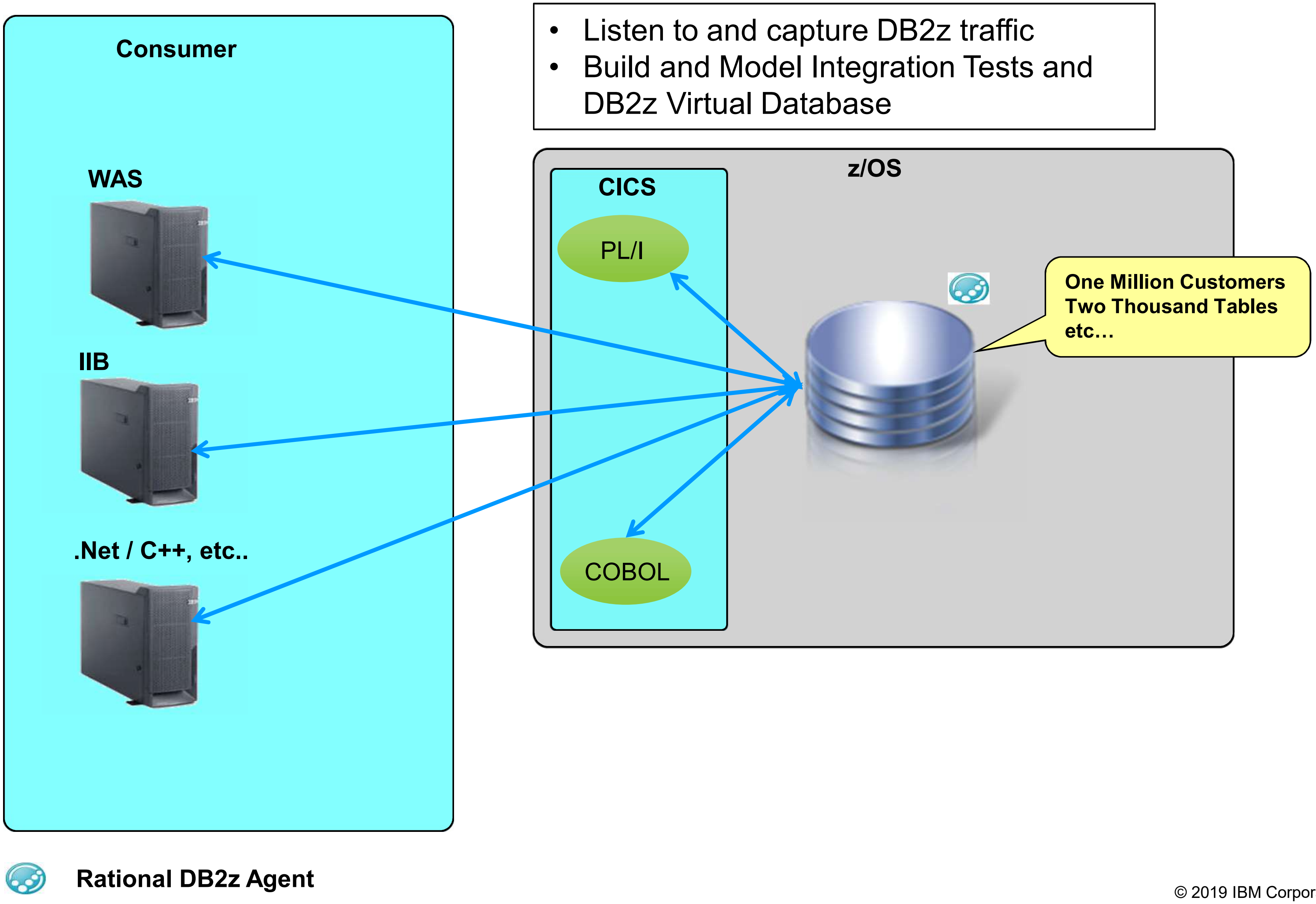
- Leverage CICS DB2z virtualization to provision virtual database for COBOL/CICS developers and testers
- Provide developers with their own test data (easy to provision and initialize
- Ability to clone data on the fly.

Benefits / Motivation

•Provide COBOL Developers with their own test data (DB2)
•Ability to test their programs more thoroughly (negative test cases)
•Reduction in scheduling constraints
•Shortening the ratio between defect identification and resolution.

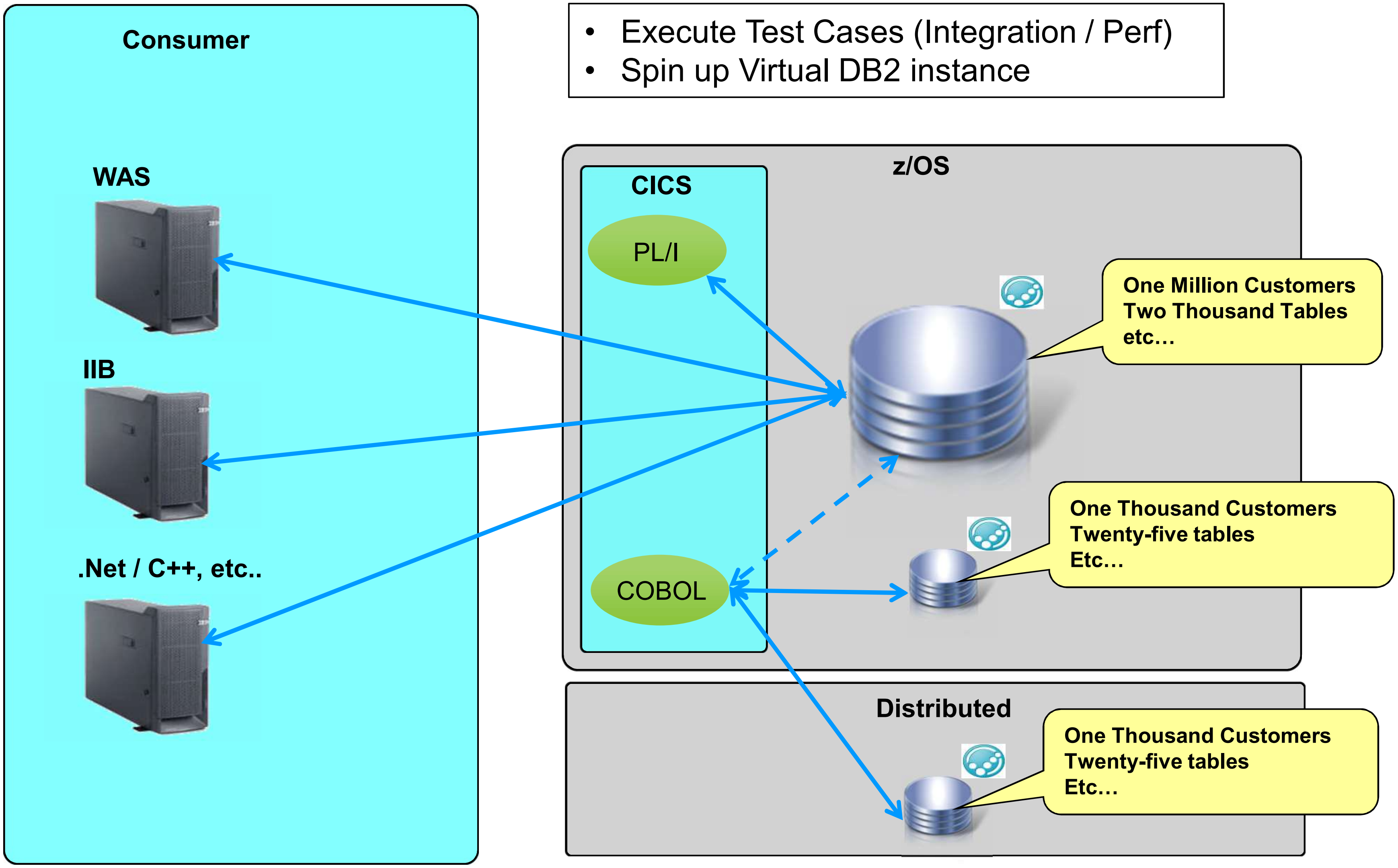
DB2z CICS Online Virtualization High Level Architecture Diagram

- Listen to and capture DB2z traffic
- Build and Model Integration Tests and DB2z Virtual Database



DB2z CICS Online Virtualization High Level Architecture Diagram

- Execute Test Cases (Integration / Perf)
- Spin up Virtual DB2 instance



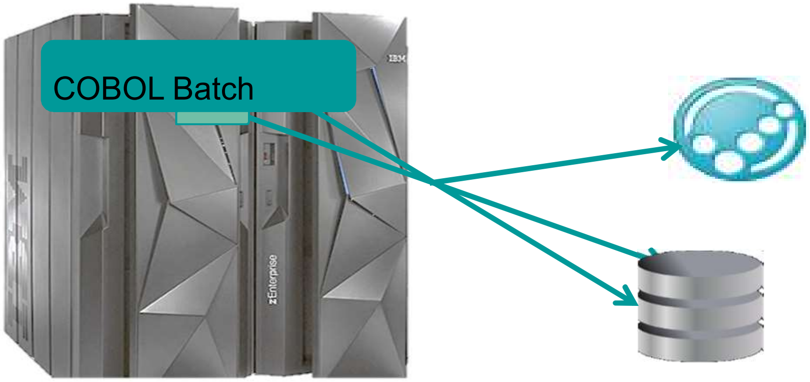
Rational DB2z Agent

COBOL – DB2 – Batch

Database Virtualization

Case Study - Pain Points

- Customer migrated many COBOL/PL1 systems from the mainframe to RD&T (Linux).
- Issues verifying the migration process (previously, all verification was done by running batch jobs manually on both systems)
- Costs were skyrocketing from manually testing migration process.
- Testing schedules were being prolonged



Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

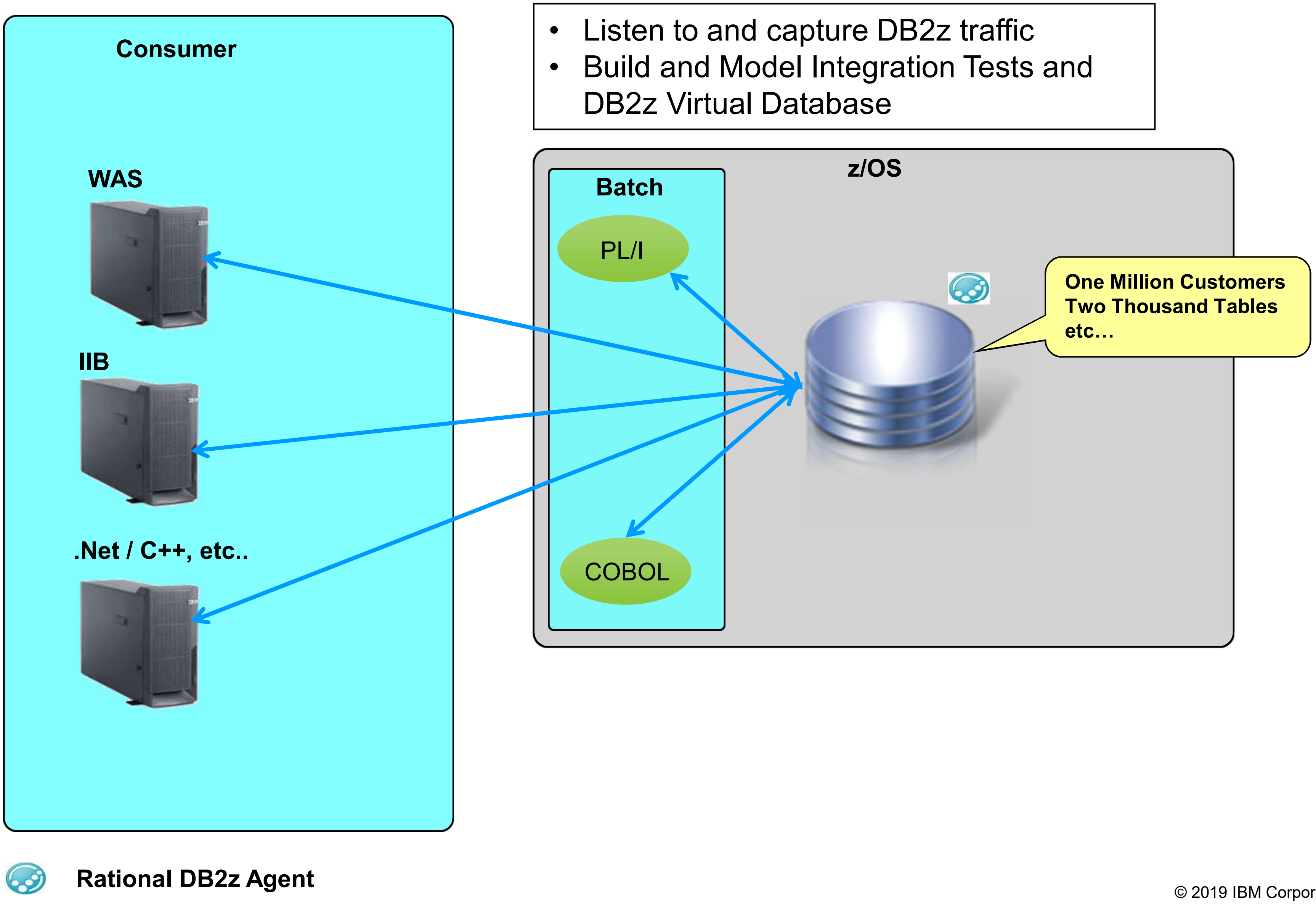
- Leverage Rational Test Workbench to execute JCL batch jobs on both zOS and RD&T (Linux)
- Compare outputs from both systems and highlight differences if they exist.
- Use existing process to automate JCL Batch testing

Benefits / Motivation

- Provide COBOL Developers with their own test data (DB2)
- Ability to test their programs more thoroughly (negative test cases)
- Reduction in scheduling constraints
- Shortening the ratio between defect identification and resolution.

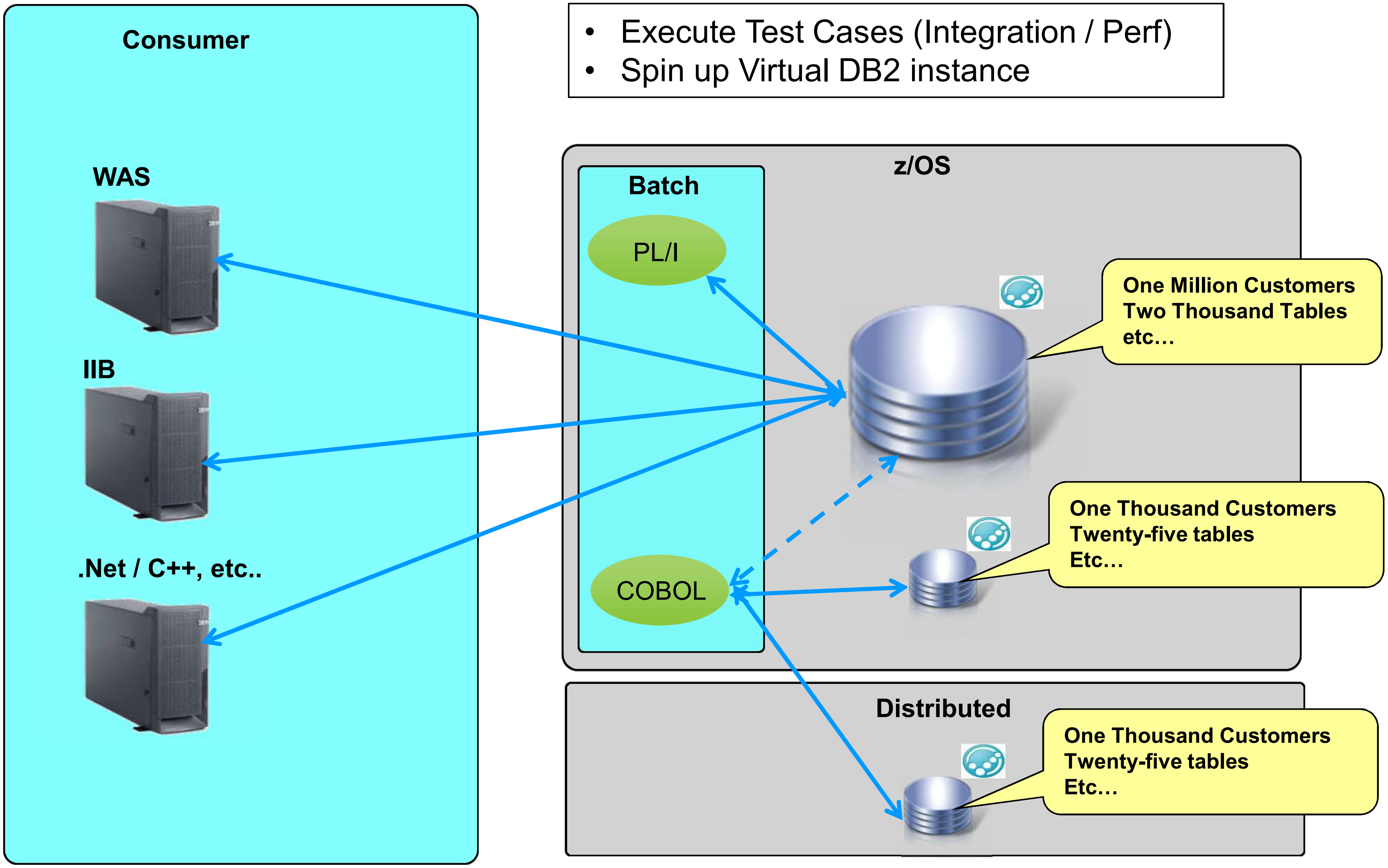
DB2z Batch Virtualization High Level Architecture Diagram

- Listen to and capture DB2z traffic
- Build and Model Integration Tests and DB2z Virtual Database



DB2z Batch Virtualization High Level Architecture Diagram

- Execute Test Cases (Integration / Perf)
- Spin up Virtual DB2 instance



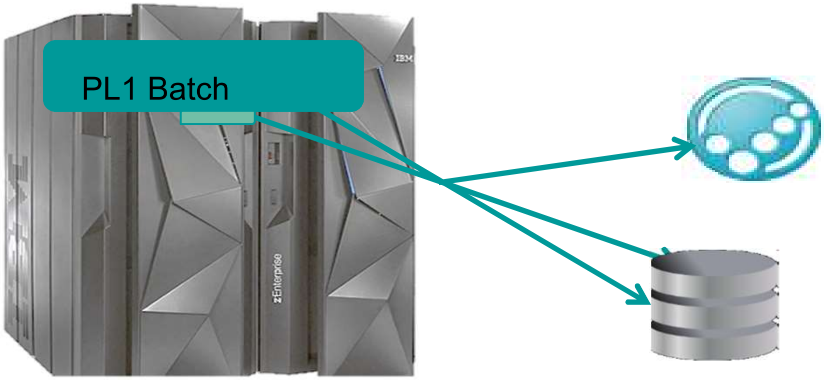
Rational DB2z Agent

PL1 – DB2 – Batch

Database Virtualization

Case Study - Pain Points

- Lack of sufficient test data is causing delays in delivery
- High percent of test cases are not executed due to lack of test data
- The majority of defects are tied back to lack of quality test data



Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

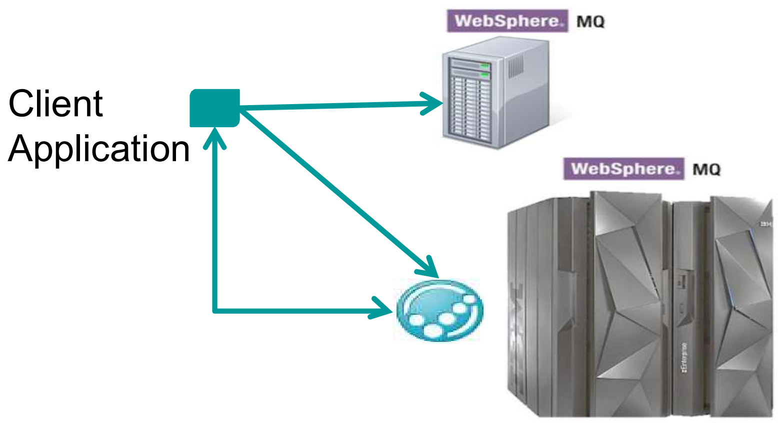
- Virtualize DB2z database to be consumed by PLI batch programmers
- Provide each PLI programmer with their own virtual database
- Additional data can be added to the virtual databases to support additional test cases

Benefits / Motivation

•Provide PL1 Developers with their own test data (DB2)
•Ability to test their programs more thoroughly (negative test cases)
•Reduction in scheduling constraints
•Shortening the ratio between defect identification and resolution.

WebSphere MQ

Test Automation / Service Virtualization



Case Study - Pain Points

- Both the distributed MQ and backend mainframe are out of the test cycle with its integration point, resources are limited.
- The majority of defects are tied back to lack of quality test data

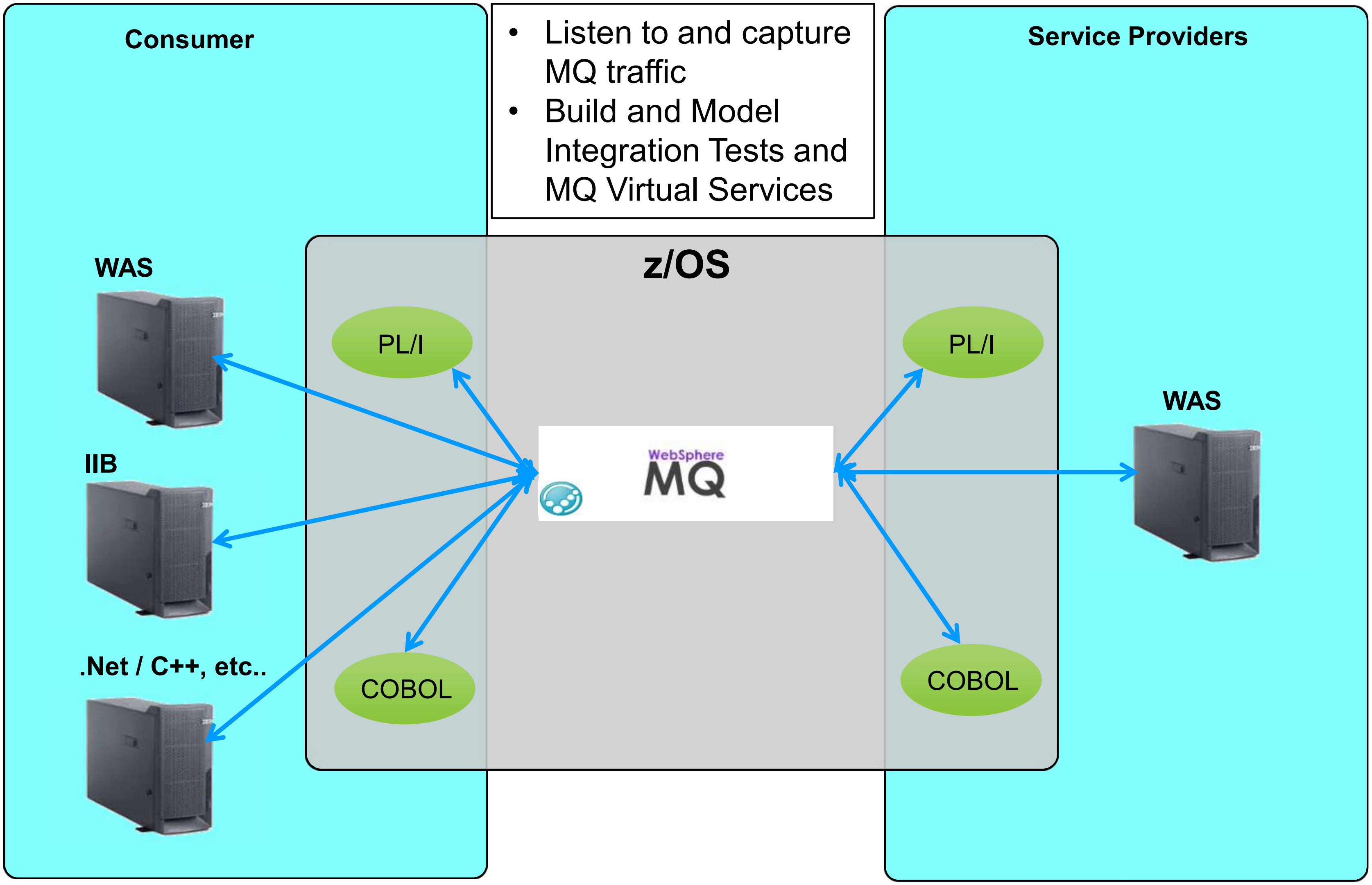
Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

- Build MQ virtual services that can be leveraged by front-end applications.
- Leverage MQ/z virtualization capability to make a stub for MQ on mainframe to serve distributed MQ requests

Benefits / Motivation

- Emulate the server application
- No changes to the client or server applications
- Distributed or Host
- Single or multiple queues
- No need to disable the server application

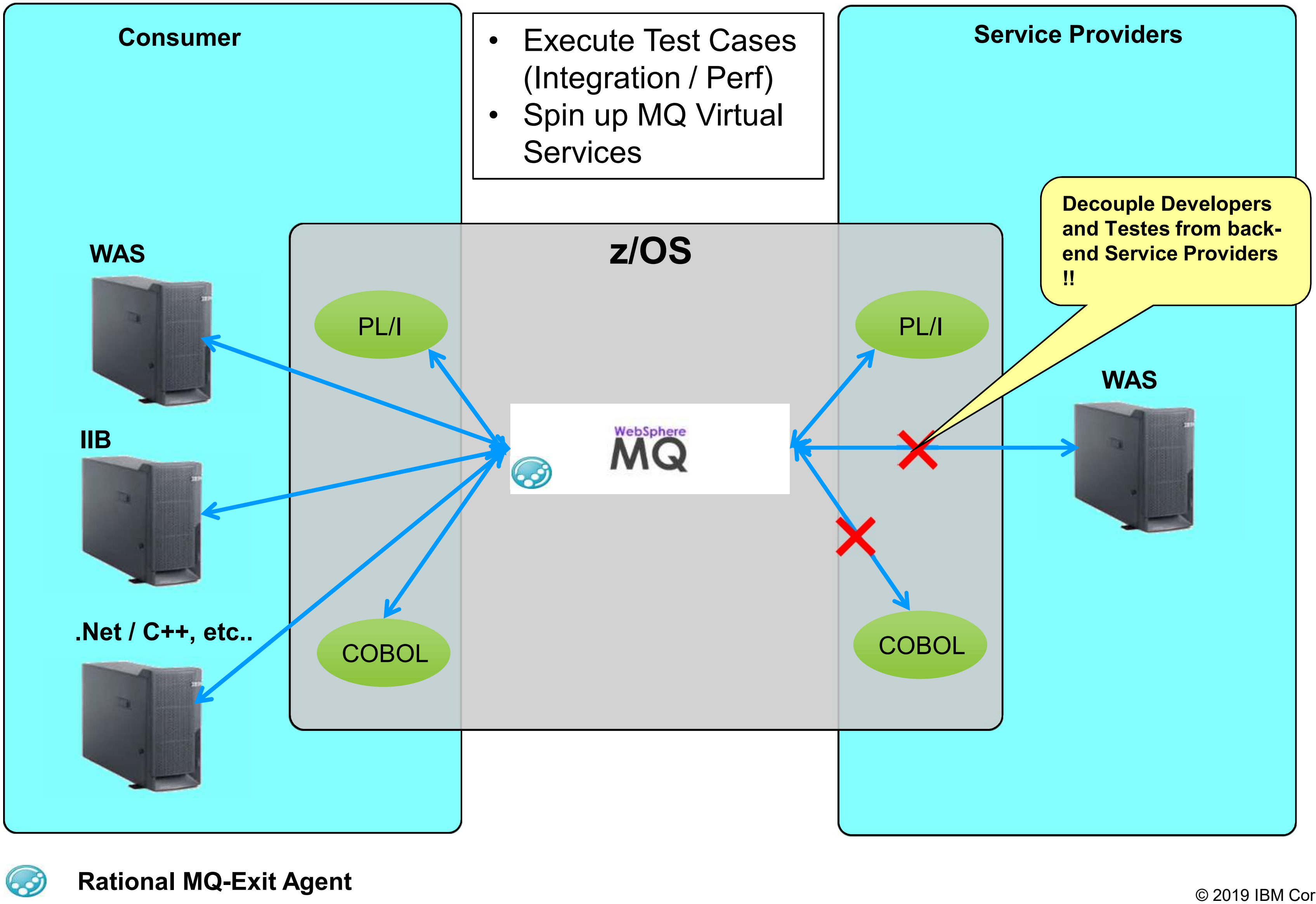
z/OS - MQ High Level Architecture Diagram



- Listen to and capture MQ traffic
- Build and Model Integration Tests and MQ Virtual Services

Rational MQ-Exit Agent

z/OS - MQ High Level Architecture Diagram



- Execute Test Cases (Integration / Perf)
- Spin up MQ Virtual Services

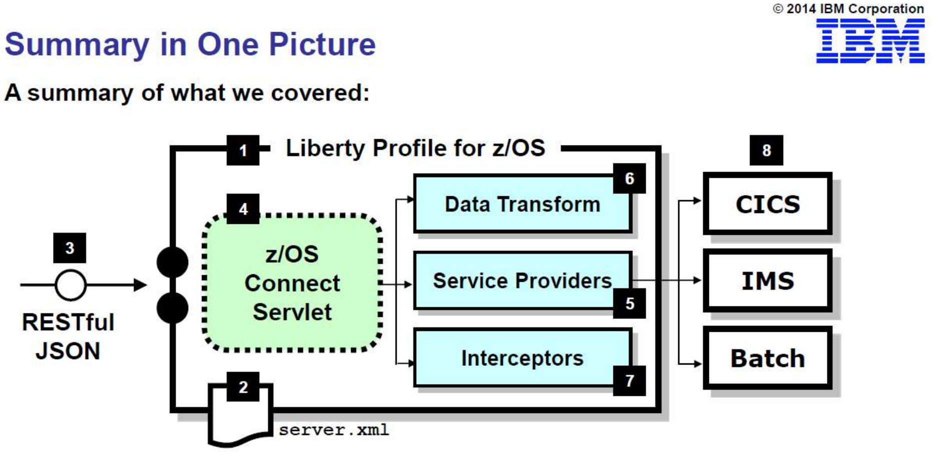
Decouple Developers and Testes from back-end Service Providers !!

z/OS Connect - Exposing Mainframe Services

Test Automation / Service Virtualization

Case Study - Pain Points

- Ability to test identified services before they are transformed – developers need the services "today"
- Ability to prop up services before they are transformed with zOS / Connect
- Lack of data to develop and test against transformed services



Summary in One Picture

A summary of what we covered:

Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

- Using Rational Integration Tester, build out unit/ integration tests that support the migration of mainframe services
- Virtualize transformed services to allow earlier testing (swap out with the "real" services are transformed.

Motivation

- Expose mainframe transactions to distributed systems (mobile, web. Etc)
- Ability to consume messages without mainframe knowledge (COBOL, CICS ..etc)
- Ability to create new transaction / data message flows.

HATS- Transformation of 3270 Back-End Services

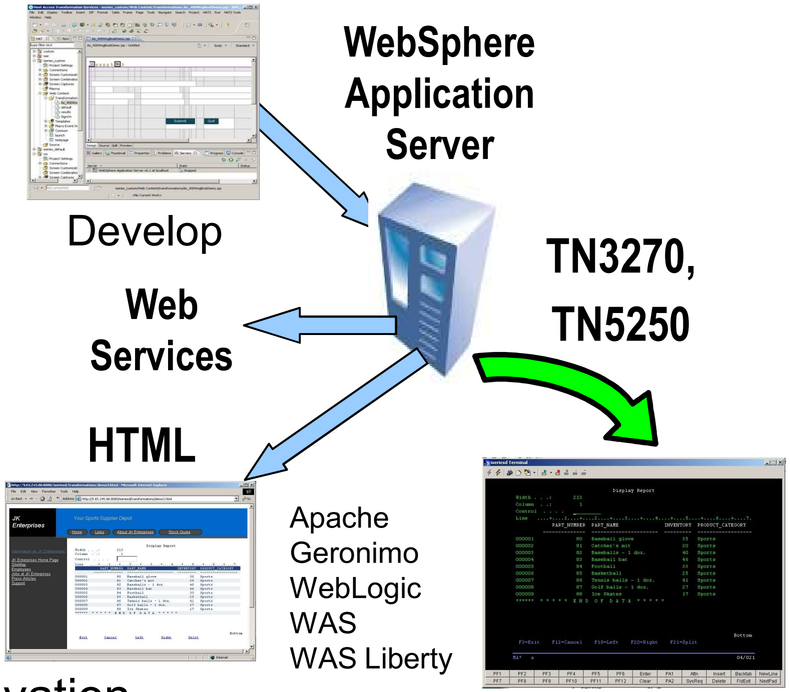
Test Automation / Service Virtualization

Case Study - Pain Points

- Manually testing back-end services through 3270 screens is not cost effect
- Lack of integration testing for 3270 work flows
- Modifying test data for 3270 applications is time consuming and error prone

Solution: **IBM Rational Test Workbench and Rational Test Virtualization Server**

- Leveraging artifacts from HATS transformation, it is now possible to create test cases and virtual services
- Bypass 3270 screen testing and use integration and unit tests to validate back-end services.
- Leverage HATS services to create virtual services that will enable shift-left development and testing
-



Develop

WebSphere Application Server

Web Services

HTML

TN3270, TN5250

Apache
Geronimo
WebLogic
WAS
WAS Liberty

Motivation

• Expose mainframe transactions to distributed systems (mobile, web. Etc)
• Ability to consume messages without mainframe knowledge (COBOL, CICS ..etc)
• Ability to create new transaction / data message flows.

Questions?

Thank You

Vinaka
köszönöm
Suksama
Maake
Asante
Shukria
Dhanyavadagalu
Dankscheen
спасибо
Kiitos
شكرا
Manana
Dankon
Suksama
Dank Je
Mauruuru
Biyan
Diolch i Chi
Gratias Tibi
Terima Kasih
நன்றி
Dziekuje
Chokrane
Grazie
Taiku
Juspaxar
Thank
Arigato
Najis Tuke
감사합니다
Bedankt
Dakujem
धन्यवाद
Gracias
Matur Nuwun
Grazas
cảm ơn bạn
Ua Tsaug Rau Koj
Di Ou Mèsi
You
Tack
Kop Khun Khap
Paldies
Obrigado
Nirringrazzjak
Děkuji
ありがとう
Rahmat
谢谢
хвала
Hvala
Eskerrik Asko
Danke
Merci
Go Raibh Maith Agat
Tingki
Salamat
ขอบขอบคุณคุณ