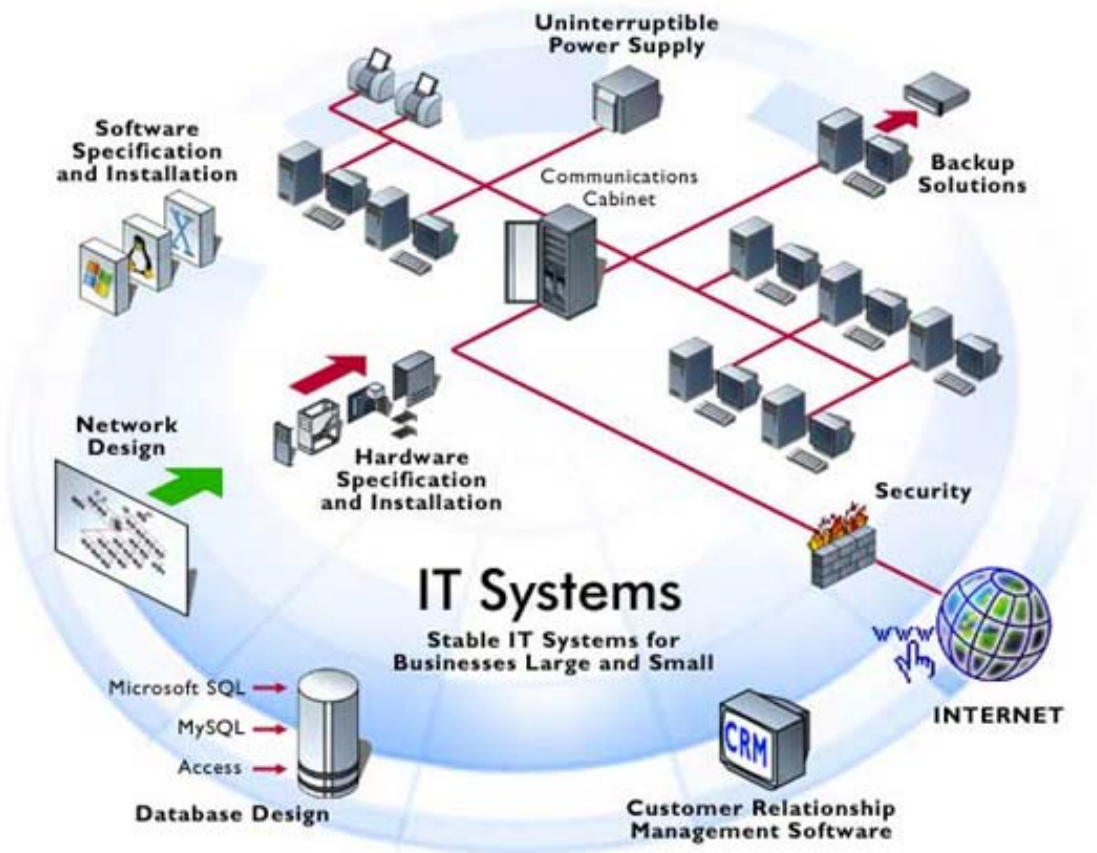
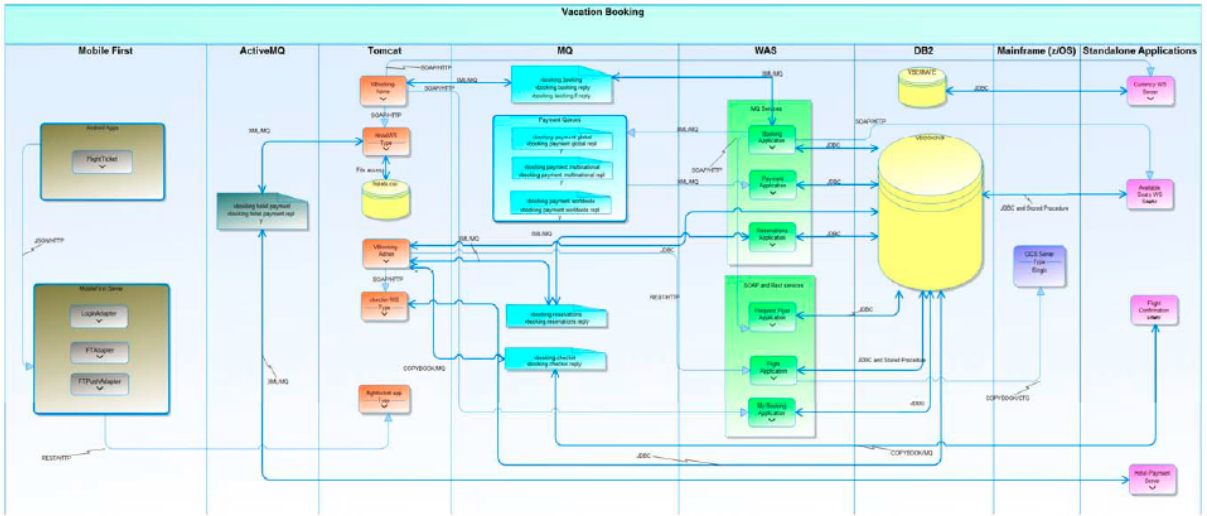
Market Leading Continuous Testing and Service Virtualization Capability for WebSphere

Stuart Feasey

IBM Hybrid Cloud - Test and Virtualization Specialist

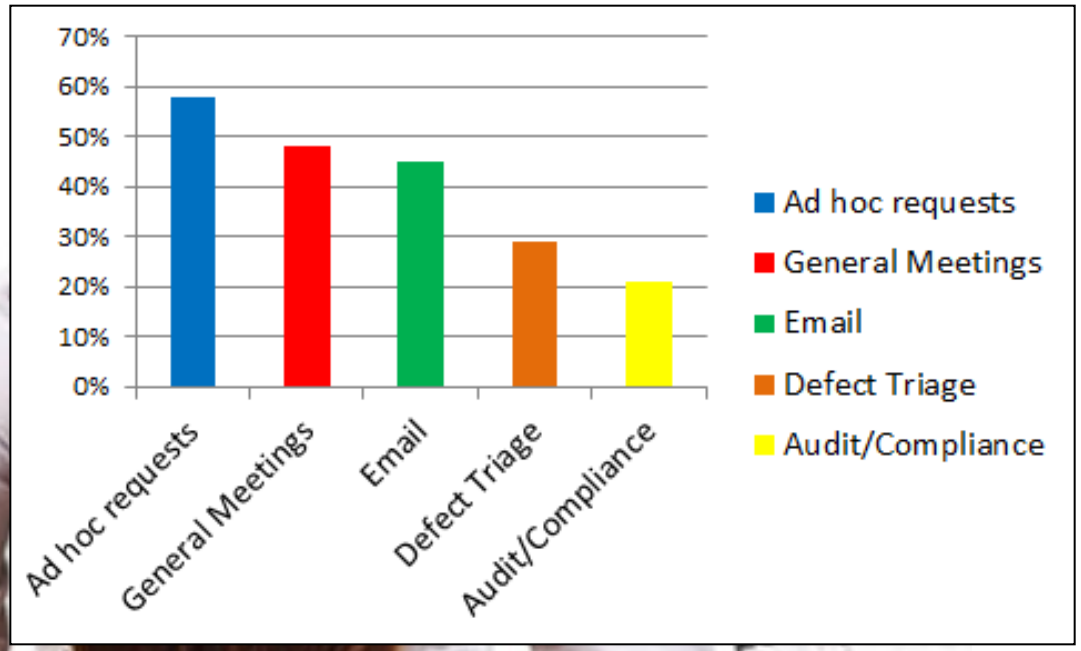
What does an enterprise environment mean to you?





Today’s Reality for Testers

40% of Testers are not testing at least 1 day per week!



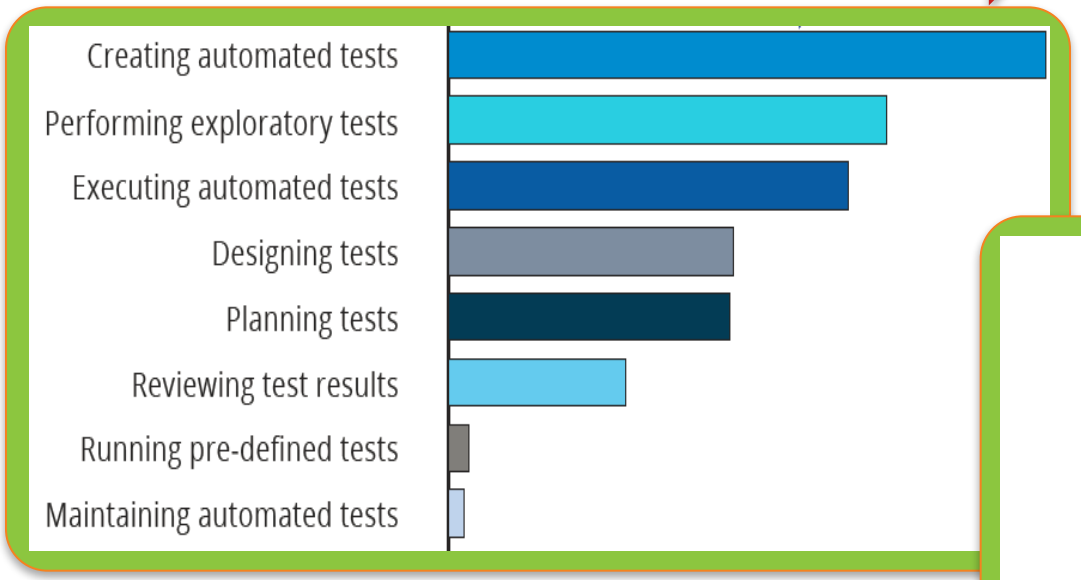
28% of Testers are not testing 50% (or more) of the time!

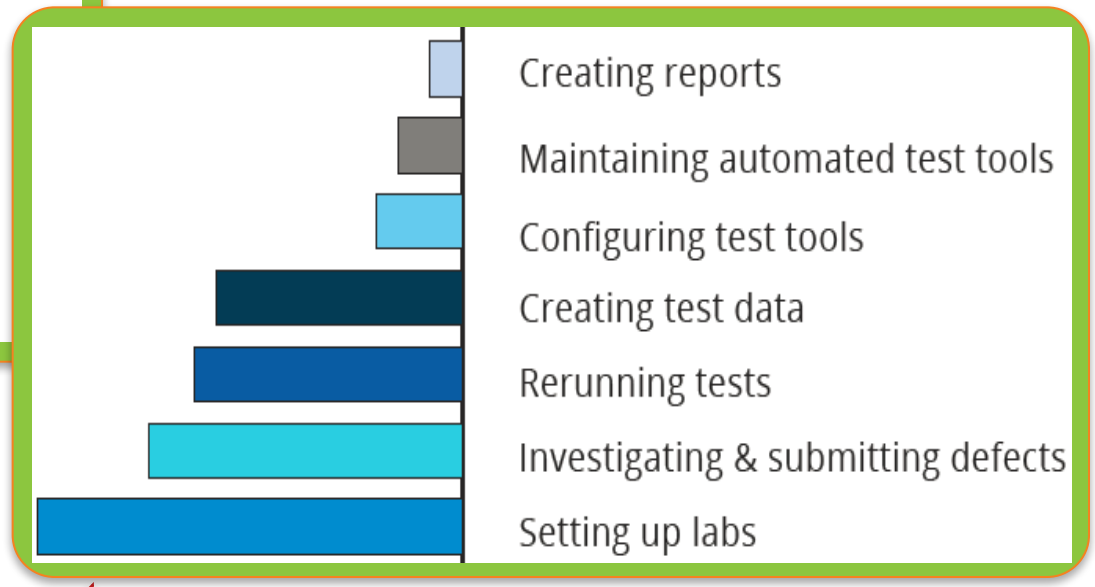
Middleware Changes

- Internal Software Upgrades
  - Standards body updates
- New Initiatives
  - Regulatory changes
- Vendor Changes
  - Stack upgrades / EOL dates
  - From one vendor to another
- Consolidation
  - Corporate M&A

Today's Reality for Testers

Testers want to spend *MORE* time





Testers want to spend *LESS* time

The Role of Testing in the Context of DevOps

Top 15 challenges identified after analysis of around 20 IBM DevOps workshop reports:

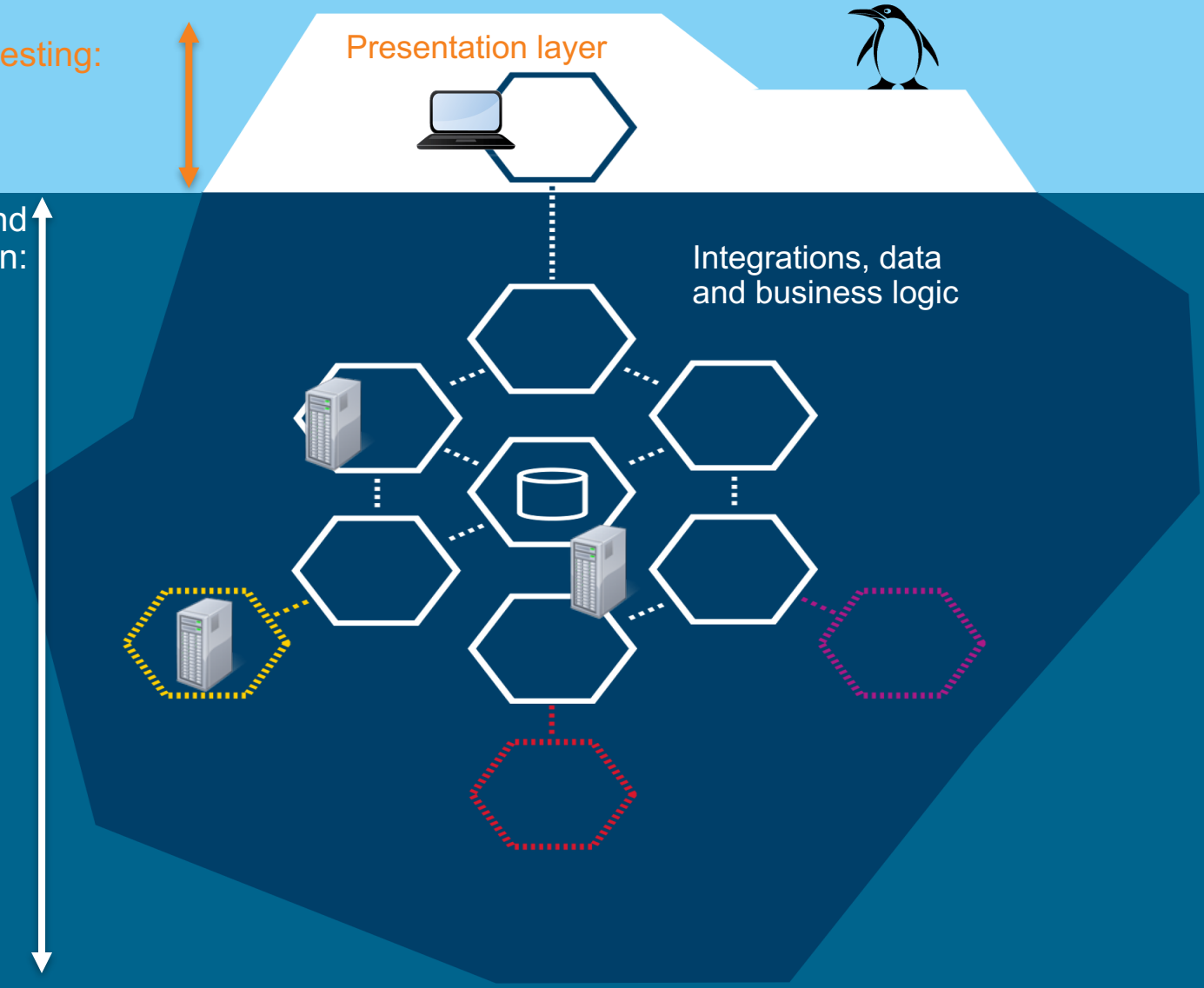
- **Flexible environment provisioning required by DevOps teams (67%)**
- There is a lack of collaboration between Bus, Dev and Ops (58%)
- There is no coordination of the delivery environment elements (e.g. a DevOps CoE) (58%)
- Governance when applying DevOps (50%)
- Deployments are largely manual (and error prone) (50%)
- There are no or limited DevOps-specific metrics in place (50%)
- **Insufficient and inconsistent test automation (42%)**
- Test data management (42%)
- **Testing is not performed early using capabilities such as service virtualisation (33%)**
- There is no integrated tools architecture in place (42%)
- No DevOps vocabulary is in place (33%)
- No DevOps vision or strategy has been defined (33%)
- Missing or inconsistent mechanisms for getting feedback (33%)
- Applying DevOps in a regulated environment (33%)

Traditional UI testing:

Presentation layer

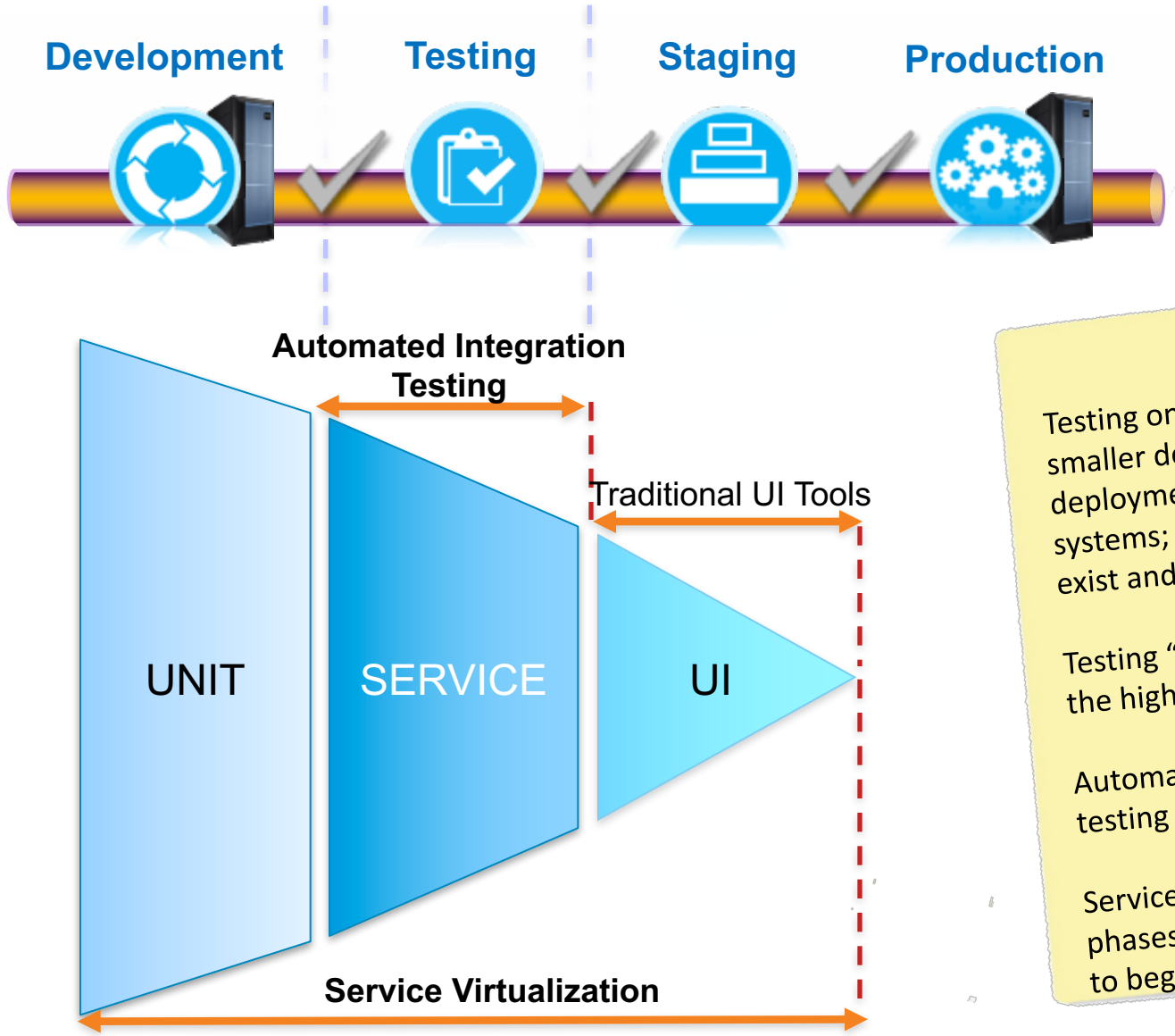
Automated Integration Testing and Service Virtualization:

Integrations, data and business logic



The majority of risk in modern systems is seldom tested as it is unseen.

Shifting left; Testing smarter



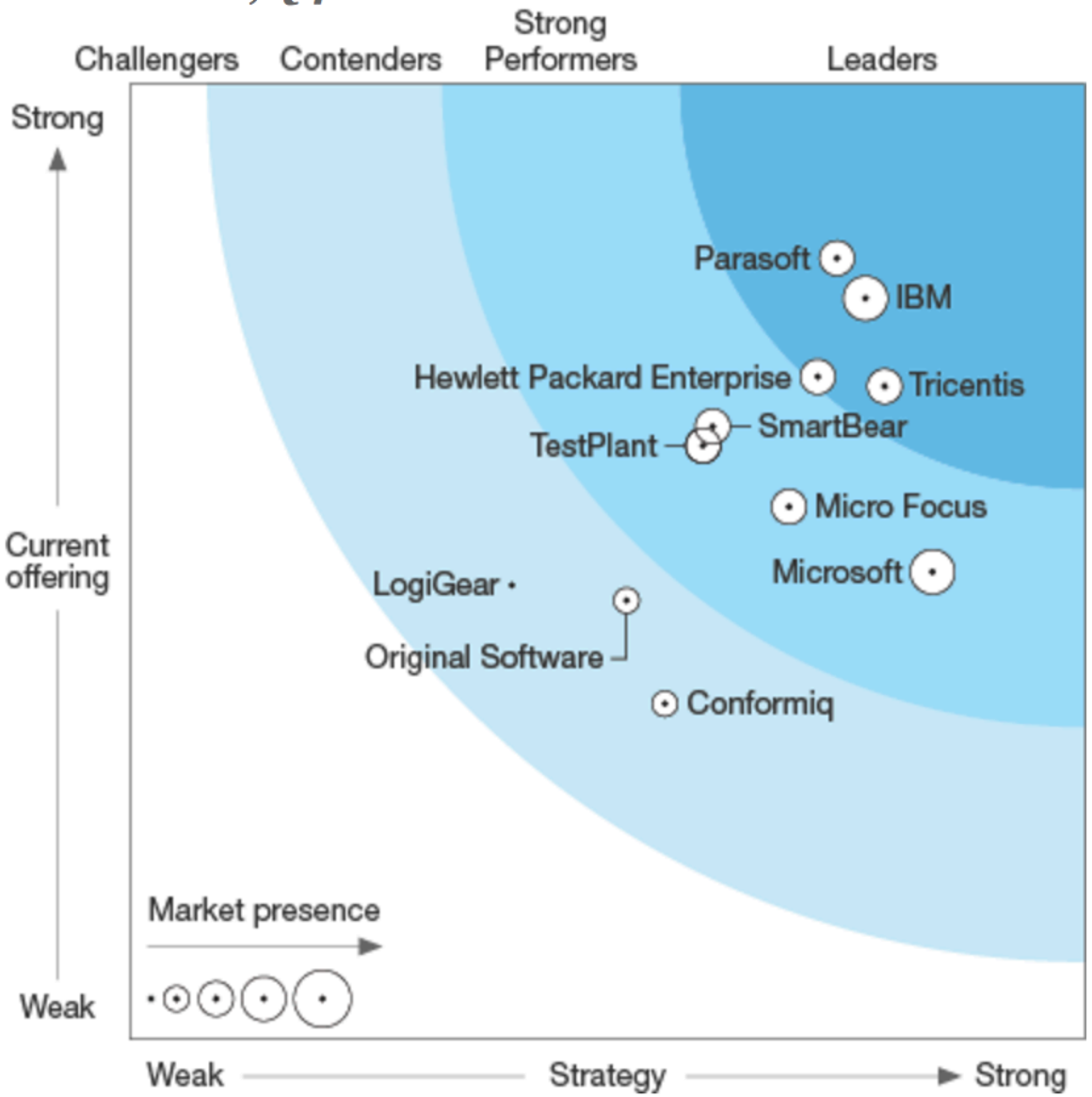
Testing only at the UI level provides a smaller degree of confidence for deployment of complex distributed systems; it also needs integrated system to exist and is potentially hard to automate

Testing "shifts left" with focus on testing the highest areas of risk

Automated integration testing increases testing coverage and confidence

Service virtualization can be applied at all phases – even allowing integration testing to begin in parallel with Unit Testing.

Figure 4: Forrester Wave™: Modern Application Functional Test Automation, Q4 '16



End-To-End Automation And API Testing Are Key Differentiators

As organizations shift to continuous testing, older approaches entirely focused on user interface (UI) testing become less effective. Going beyond the UI and testing APIs is crucial to avoiding brittle test suites and increasing test coverage. AD&D cannot achieve ruthless automation achieved by focusing solely on test execution automation; it also requires automating test design and process orchestration. The vendors that best address all those needs lead the pack.

Gartner - Magic Quadrant for Software Test Automation
Published: 15 November 2016

Magic Quadrant

Figure 1. Magic Quadrant for Software Test Automation

IBM STRENGTHS

•IBM's test automation tools can give many enterprises a jump-start in their continuous testing efforts. They are strongest as an integrated solution, delivering a combination of products and services.

•IBM provides automated testing support for many technologies and applications, including legacy systems, client/server applications, packaged applications, and Web and mobile applications. It offers integration across the development and delivery life cycle.

•IBM is well-positioned for cloud-based testing and advanced cognitive analysis to aid in adaptive testing and decision making.

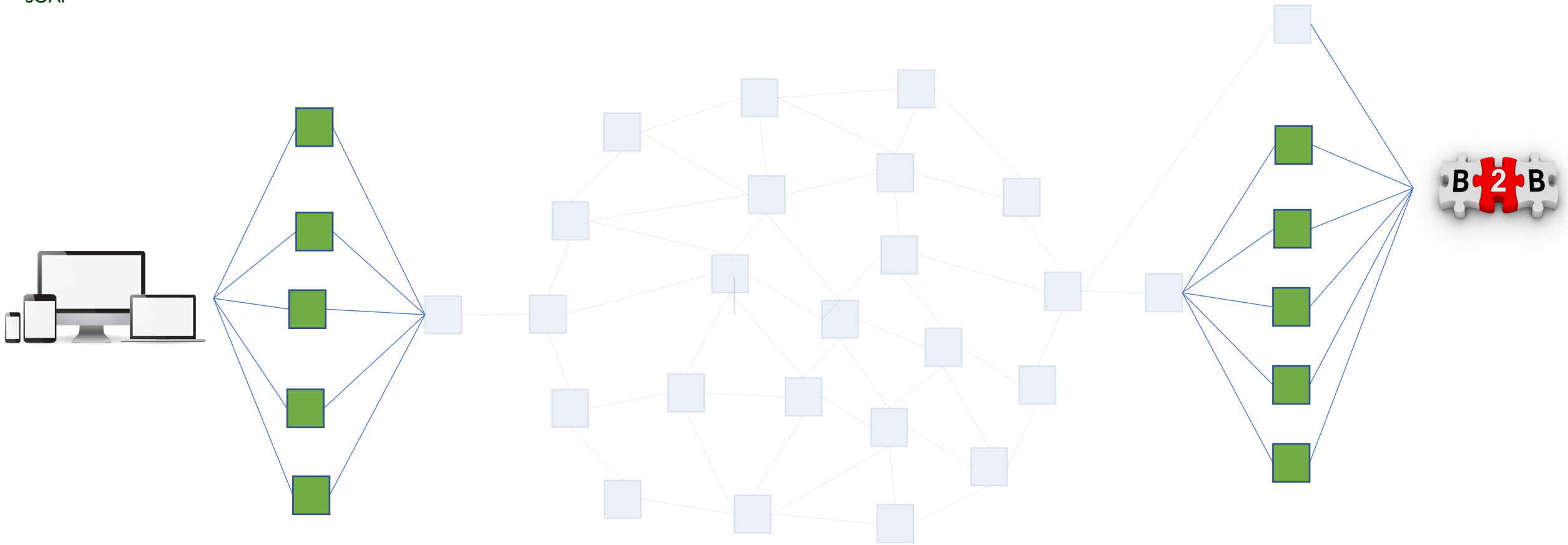
IBM CAUTIONS

•IBM's portfolio is comprehensive and can be perceived as too complex, especially in the early stages of investing in test automation or for simpler testing needs.

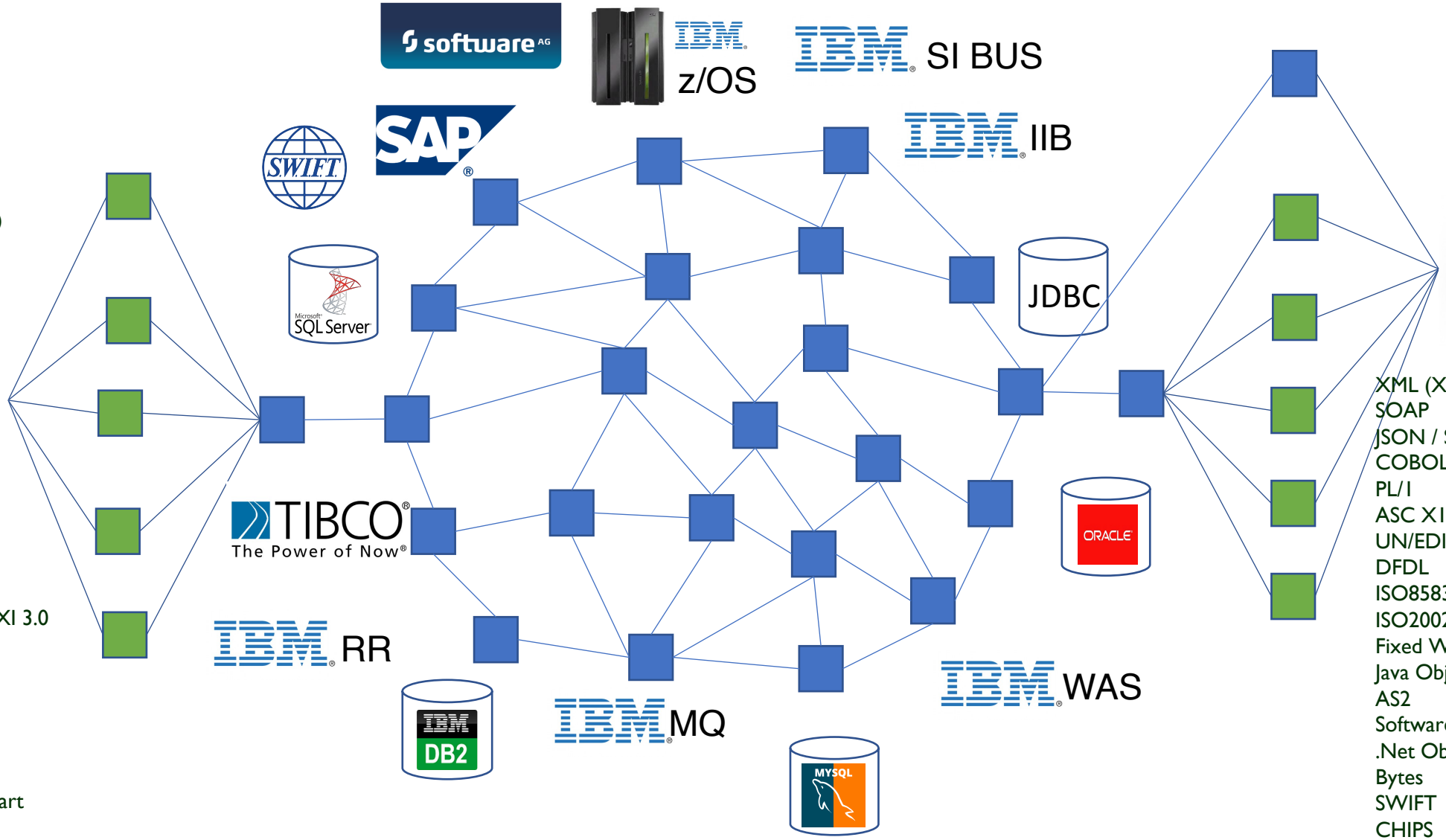
•While IBM provides a powerful portfolio, it is not optimized for nontechnical users and may require additional services to set up and implement.

•IBM supports packaged application testing, but relies on partners for business process analysis and validation.

HTTP(S)
Web Services / REST
JSON / Swagger
XML (XSD, WSDL)
SOAP



Websphere Application Server
IBM Integration Bus (WSMB)
Websphere MQ
Websphere SI Bus
Websphere RR
UDDI
Web Services / REST
WSDL / WADL / Swagger
System z (CICS and IMS)
Software AG IS, Broker
Oracle Fusion
TIBCO ActiveMatrix
Databases
Log Files
CentraSite
SCA Domain
HTTP(S)
WebSphere MQ
JMS (multiple)
REST
S/MIME
Java RMI (virtualization)
JVM (virtualiazation)
SAP IDOC, BAPI, RFC, XI 3.0
Software AG's IB & IS
TCP/(S), IP, UDP
CICS TG
CICS IPIC
CICS DPL
IMS Connect
TIBCO Rendezvous, Smart Sockets & EMS
Active MQ
Email (SMTP, IMAP)
Files
FTP/S



XML (XSD, WSDL, WADL)
SOAP
JSON / Swagger
COBOL Copybook
PL/1
ASC X12
UN/EDIFACT
DFDL
ISO8583
ISO20022
Fixed Width
Java Objects
AS2
Software AG Broker Docs
.Net Objects
Bytes
SWIFT
CHIPS
FedWire
FIX
TIBCO ActiveEnterprise
Custom

IBM DevOps Solution for Continuous Testing

Presentation layer

Middleware, Data, Services

Systems of Record components

Rational Test Workbench

Web UI Tester

Functional Tester

Integration Tester

Rational Test Virtualization Server

Performance Tester

RPT on Cloud

RPT VT Packs

Rational Performance Test Server

Middleware Testing Challenges

- What is it that needs testing
  - Unknown system components
  - Where are the system endpoints
  - Variation from documents
- What messages are being exchanged
  - Good source of data
  - Mirror real life rather than history
- Identifying changes
  - Help target testing
  - Optimize creation of new test assets

Continuous Testing

Execute test assets in headless mode

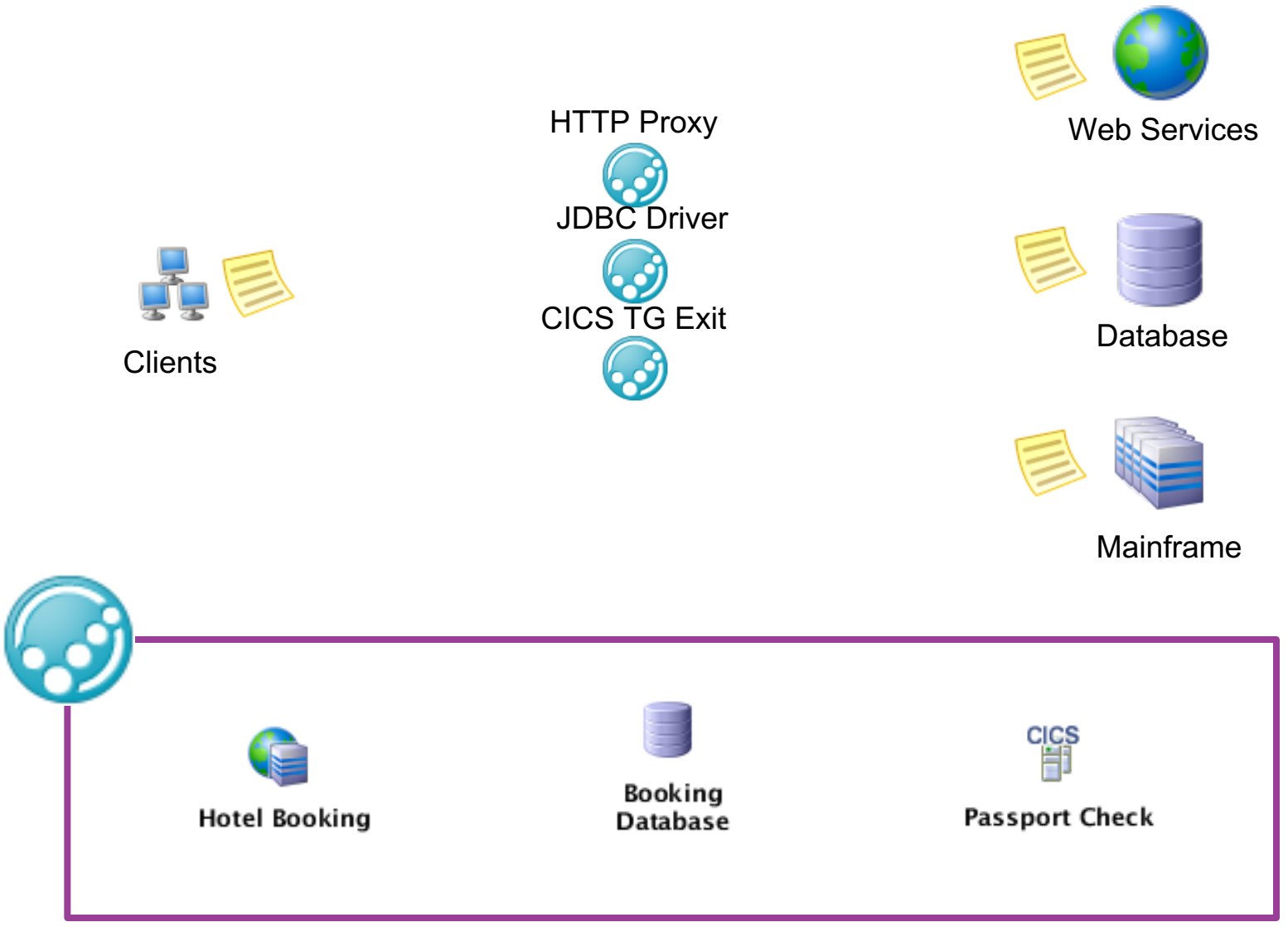
- Regression test new code quickly
- Continuous test cycles
- Build / Deploy / Test / **Virtialize** / Report

Integrate into new or existing process

- Urban Code Plugin
- ANT task integration with Jenkins, CruiseControl, Bamboo, etc
- Command line execution
- Native integrations RTC, RQM, HP QC etc

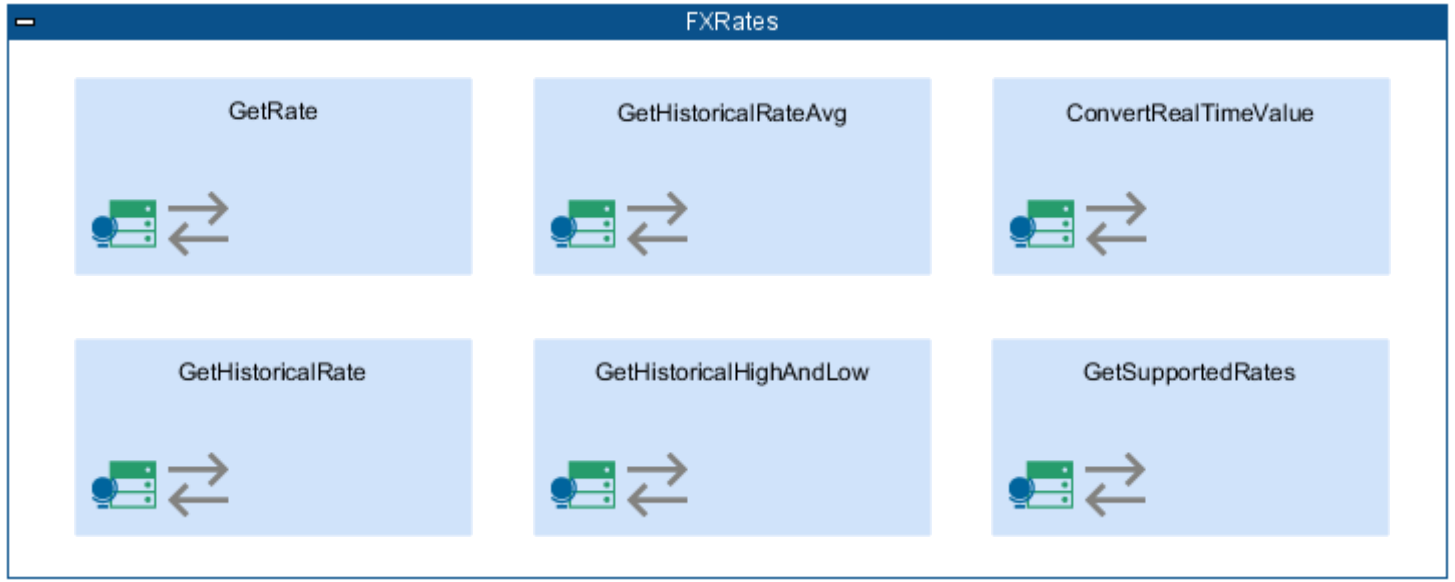
Integration Tester reports served through a web browser (via Control Panel)

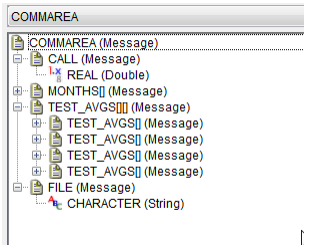
Topology Discovery

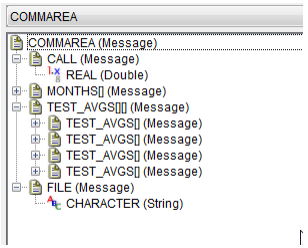


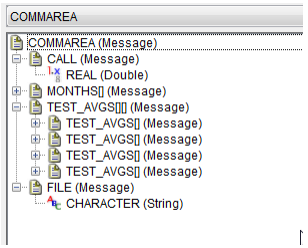
Synchronization

```
<portType name="FXRates">
    <operation name="GetRate"> ... </operation>
    <operation name="GetHistoricalRate"> ... </operation>
    <operation name="GetHistoricalRateAvg"> ... </operation>
    <operation name="GetHistoricalHighAndLow"> ... </operation>
    <operation name="ConvertRealTimeValue"> ... </operation>
    <operation name="GetSupportedRates"> ... </operation>
</portType>
```









Building a system model from recorded events

Queue.A

Queue.A.Reply

Queue.B

Queue.B.Reply

Queue.C

Queue.C.Reply

Clients

Application servers

COMMAREA

- COMMAREA (Message)
  - CALL (Message)
    - REAL (Double)
  - MONTHS[] (Message)
  - TEST_AVGS[][] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
  - FILE (Message)
    - CHARACTER (String)

COMMAREA

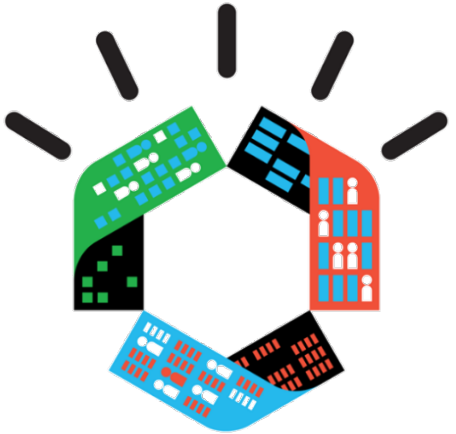
- COMMAREA (Message)
  - CALL (Message)
    - REAL (Double)
  - MONTHS[] (Message)
  - TEST_AVGS[][] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
  - FILE (Message)
    - CHARACTER (String)

COMMAREA

- COMMAREA (Message)
  - CALL (Message)
    - REAL (Double)
  - MONTHS[] (Message)
  - TEST_AVGS[][] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
    - TEST_AVGS[] (Message)
  - FILE (Message)
    - CHARACTER (String)

WebSphere

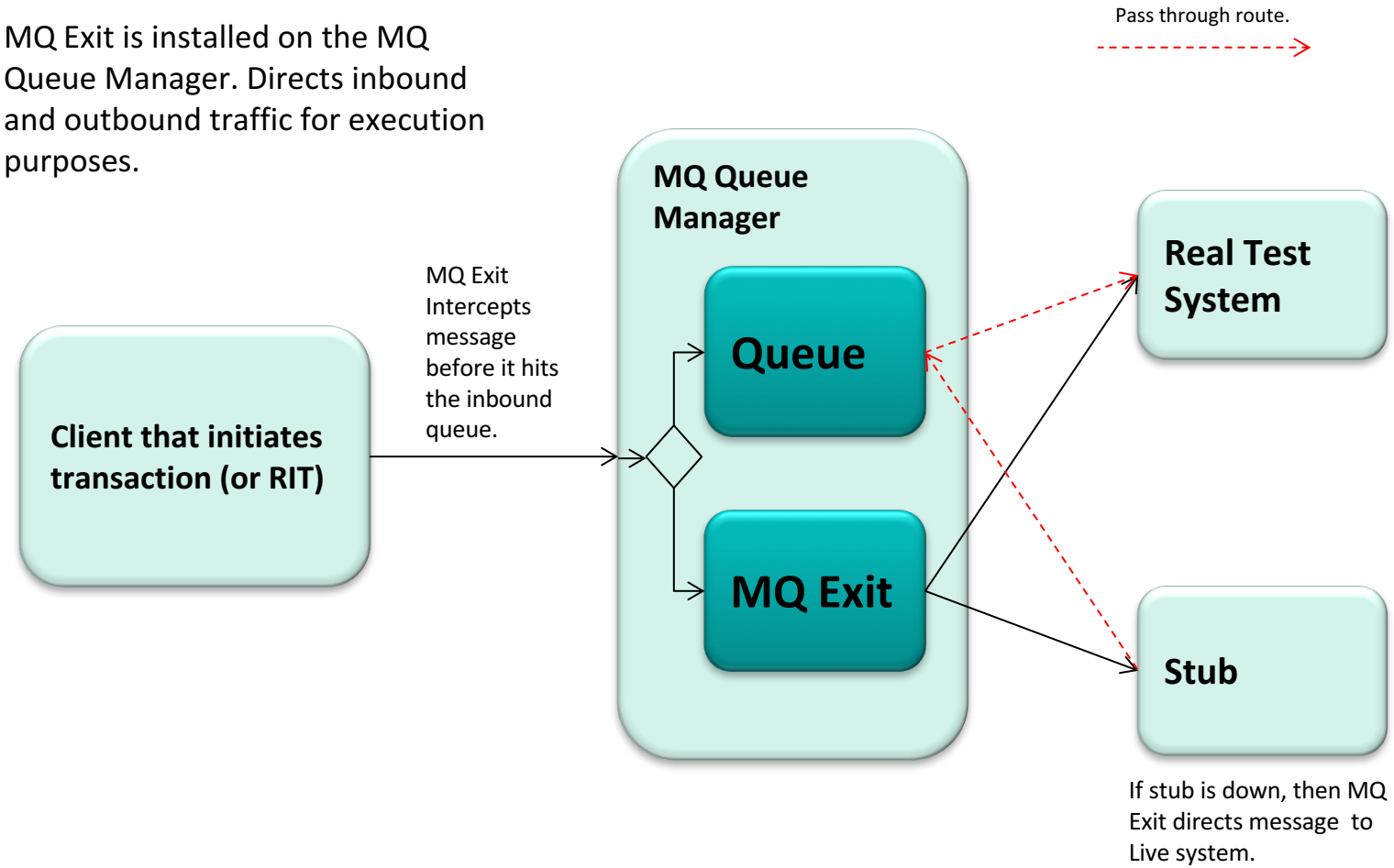
- Rational Integration Tester can synchronize with:
  - IBM Integration Bus
  - WebSphere Application Server
  - WebSphere Registry and Repository



- Discover:
  - HTTP Endpoints, MQ connections, JMS providers, WSDLs, Databases, CICS and IMS connections
- Import
  - DFDL schemas
  - WSDL, WADL, XSD, Swagger
  - Copybooks, Protocol Buffer Definitions
  - From API Connect, Z/os Connect Enterprise Edition

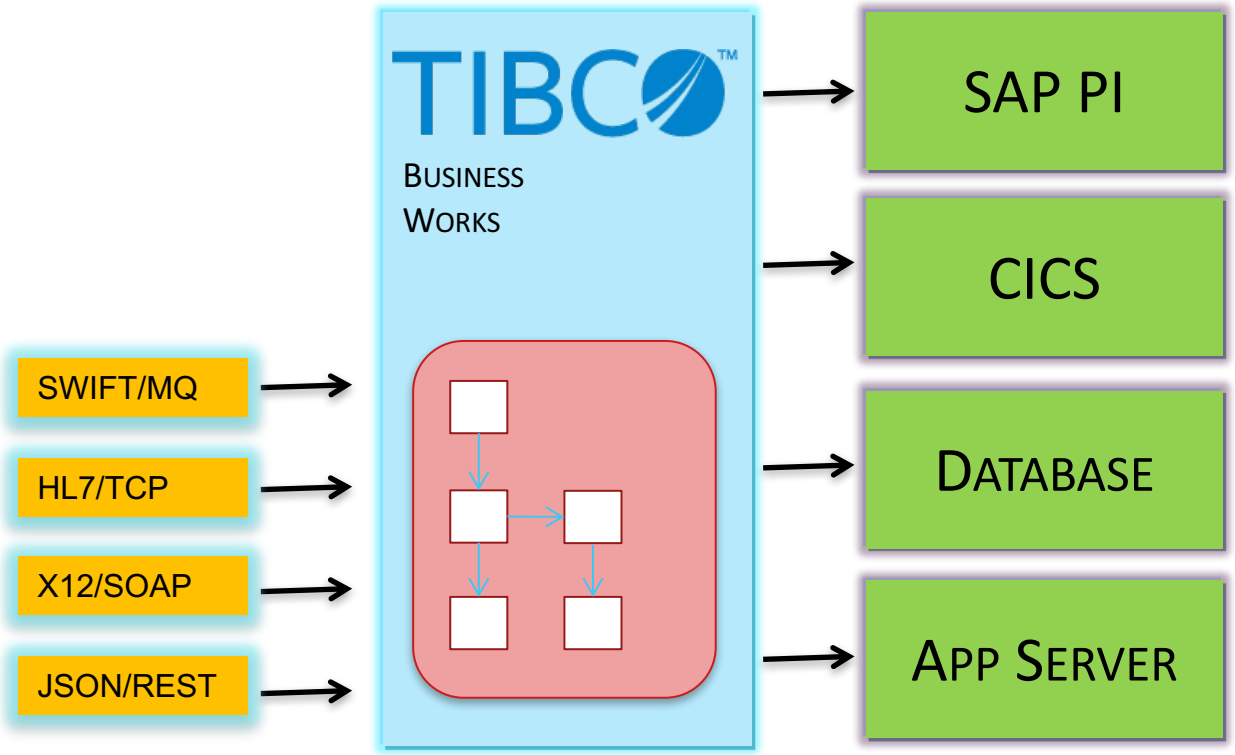
Sift and pass through with MQ

MQ Exit is installed on the MQ Queue Manager. Directs inbound and outbound traffic for execution purposes.



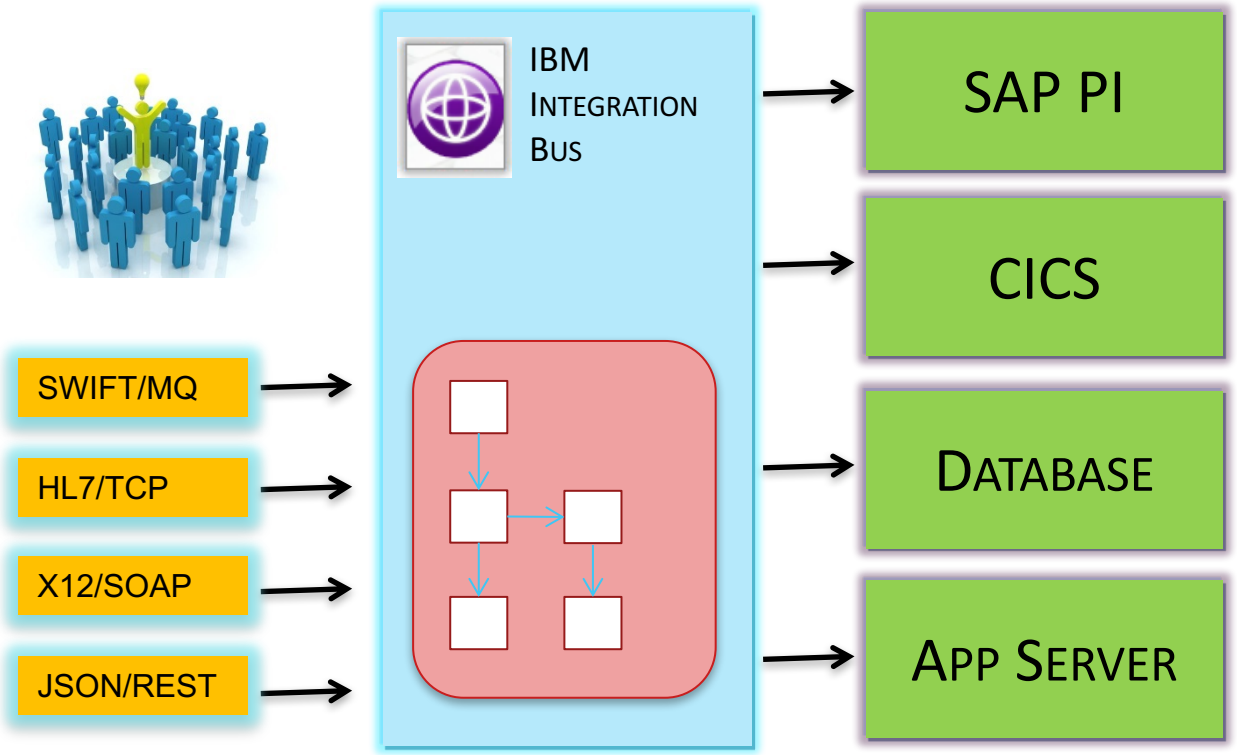
Typical Scenarios Encountered

Scenario 1: A Migration Challenge



Most Orchestration Engines have their own proprietary runtime (unlike J2E) which means migration between vendors is a risk proposition as development is required. Test environments are expensive and it is extremely difficult to fault-find in an environment with many moving parts

Scenario 1: From TIBCO to IIB

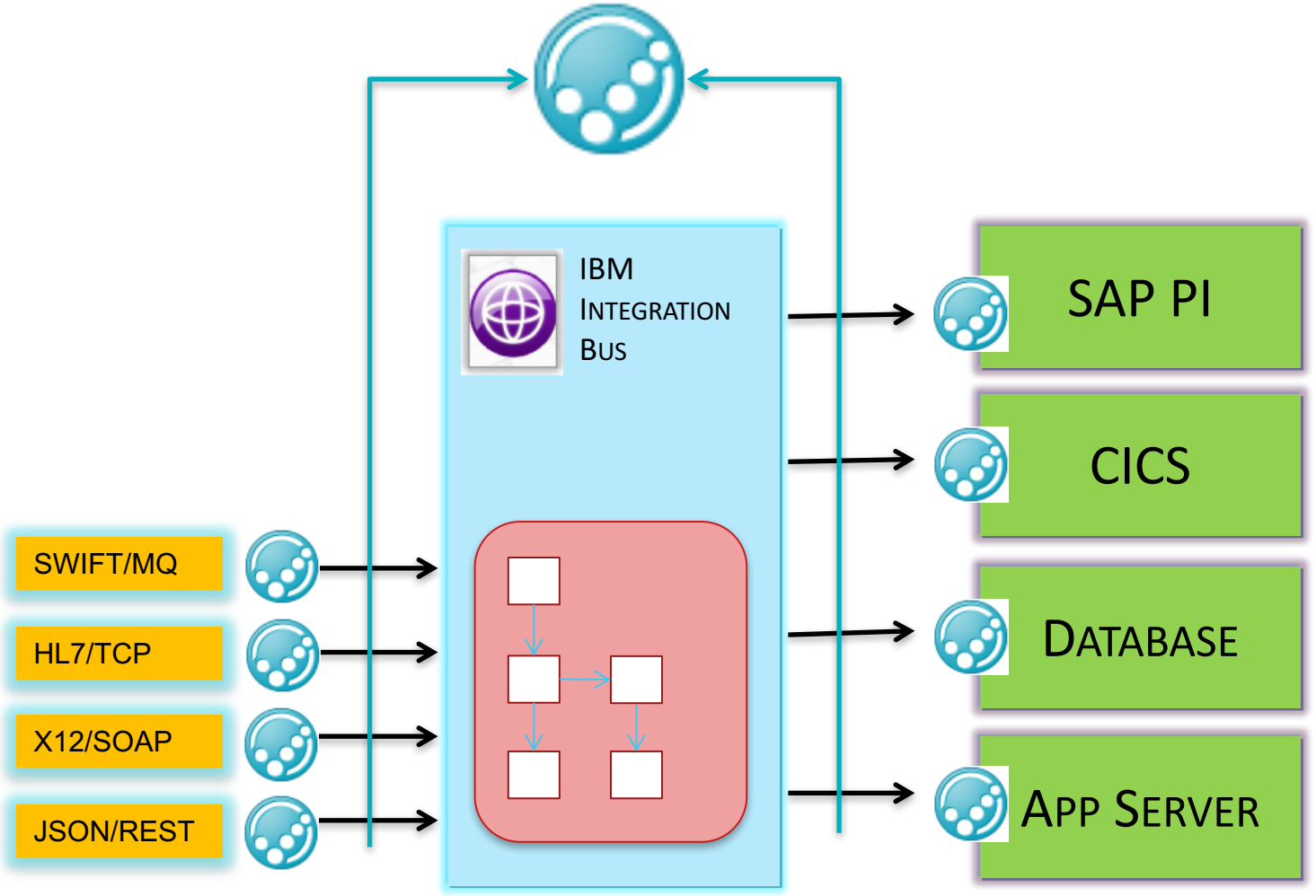


Integration Testing and Service Virtualization can significantly de-risk the migration and reduce the time required.

Development/test teams can build against virtual services and test early and test continuously using messages captured from the real systems. Exception/edge case testing is much easier to do.

RIT understands a wide range of technologies and messaging protocols (including DFDL) making it well-aligned to IIB users

Scenario 1: Details



Step 1 – Record real messages (using RIT Agents)

Step 2 – Create tests and virtual services

Step 3 – Enhance tests and virtual services to include edge/error cases

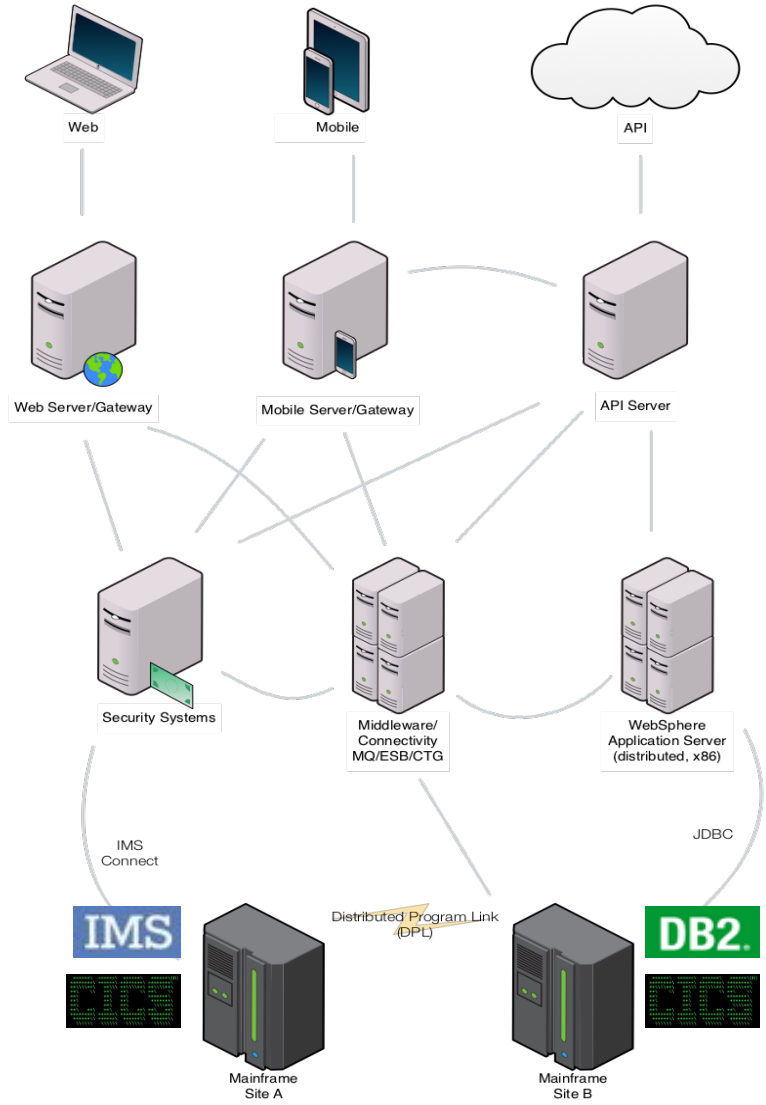
Step 4 – Migration team use tests/services to build IIB solution and gain confidence

Step 5 – Swap virtual services for real test environments one at a time and re-run tests

Step 6 – End to end test (confident of less risk of expensive errors at this stage)

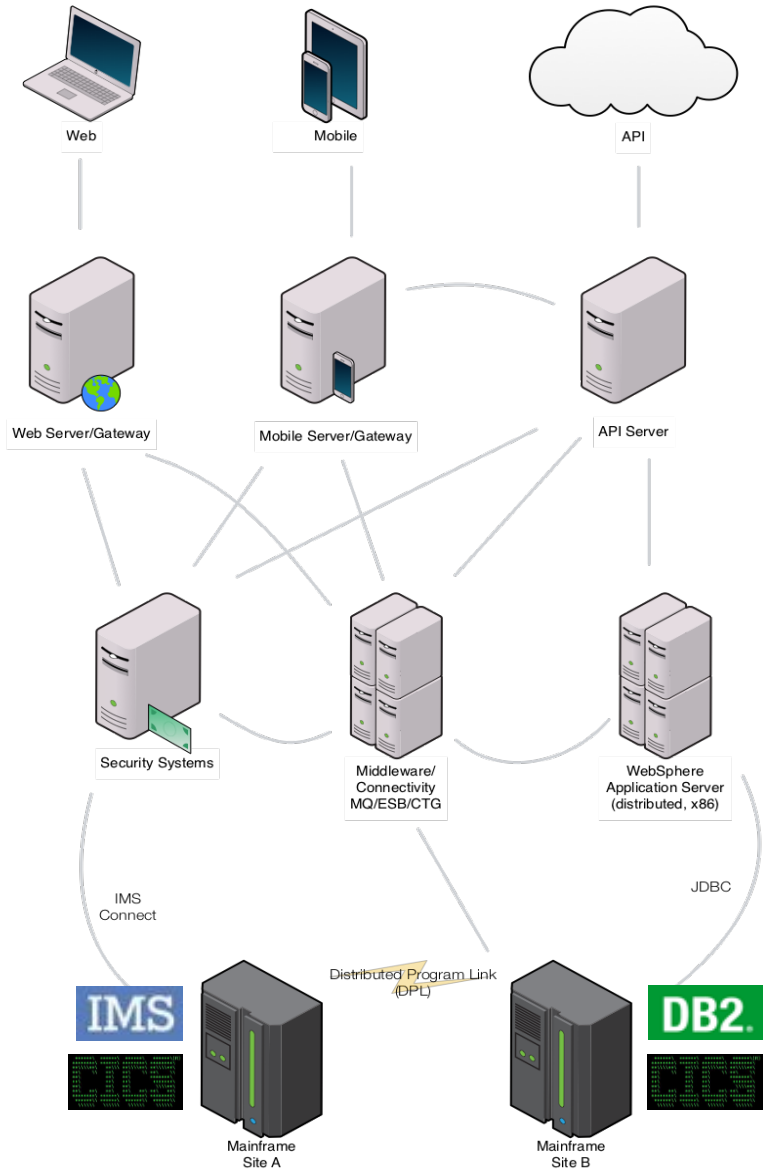
Scenario 2: Increasing Build / Delivery Velocity

- Financial Services Company
  - A Modern Enterprise
- Complexity
  - Business
  - Technology
  - Organization
- Get next version of web/mobile offerings to market ASAP
- Maintain quality whilst progressing towards continuous delivery



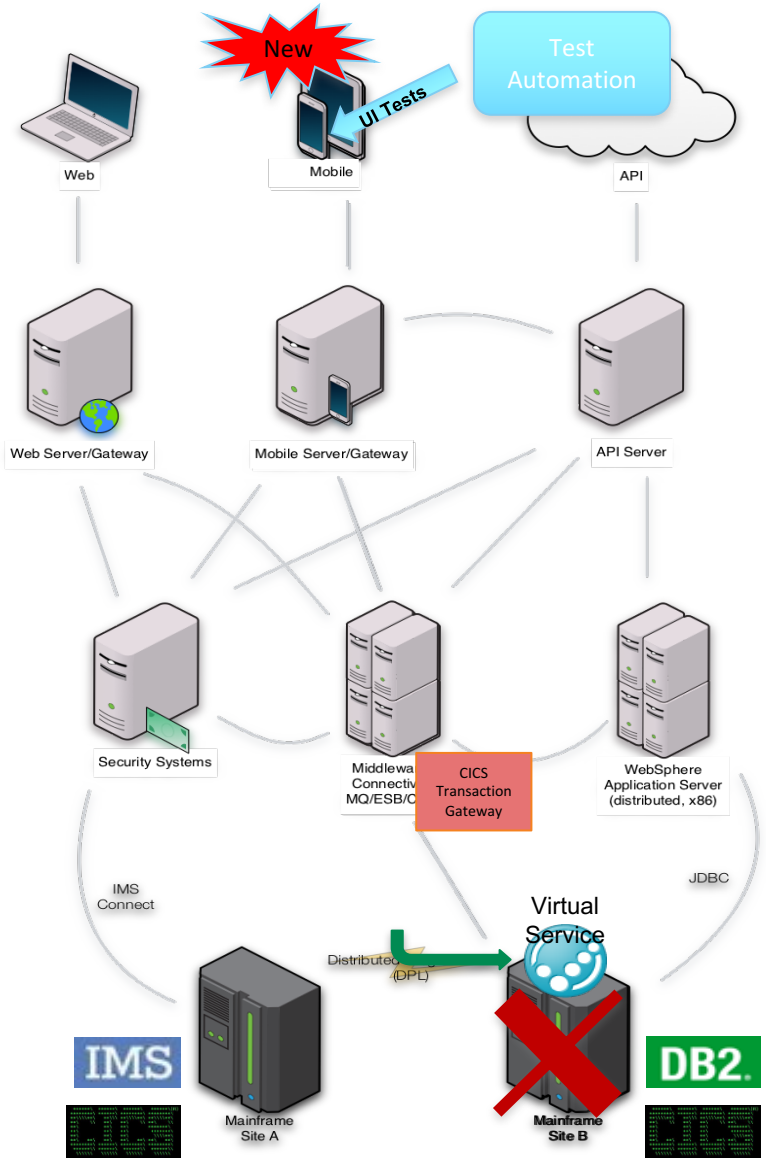
Scenario 2: Meet that Challenge with Devops Capabilities

- Understand the issues and find opportunities to improve
- **Automate** and **simplify** to enable continuous integration
  - Open interfaces to test tools
    - ANT scripting
    - Comand line
    - REST interface
    - UrbanCode plugins
- Leading to a smooth path to Continuous Testing and ultimately Continuous Delivery



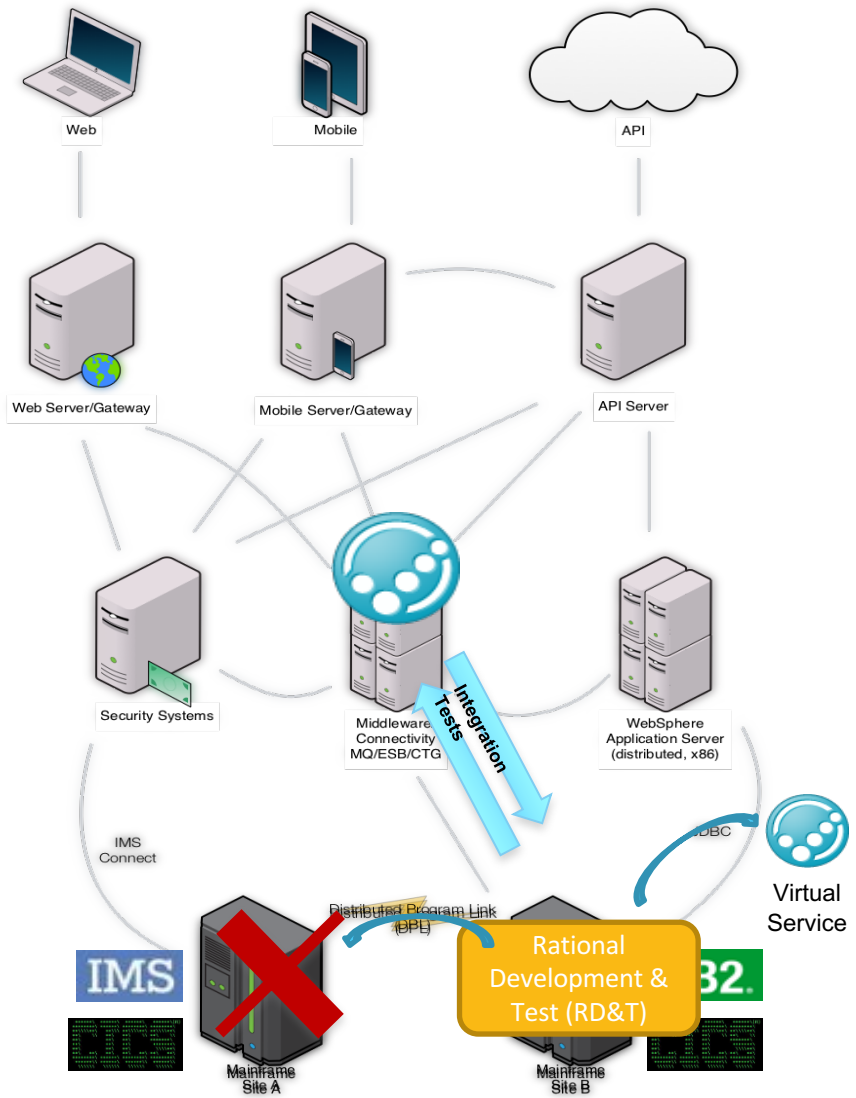
Scenario 2a: CICS Region unavailable

- App development and test team use UI testing in Rational Test Workbench
- Accesses record data and services through CICS via Transaction Gateway
- Resource constraints - won't be available for 2 weeks
- Development team use Service Virtualization to simplify
  - Prototype the service with **Rational Test Virtualization Server**



Scenario 2b: Dependent Program Not Available

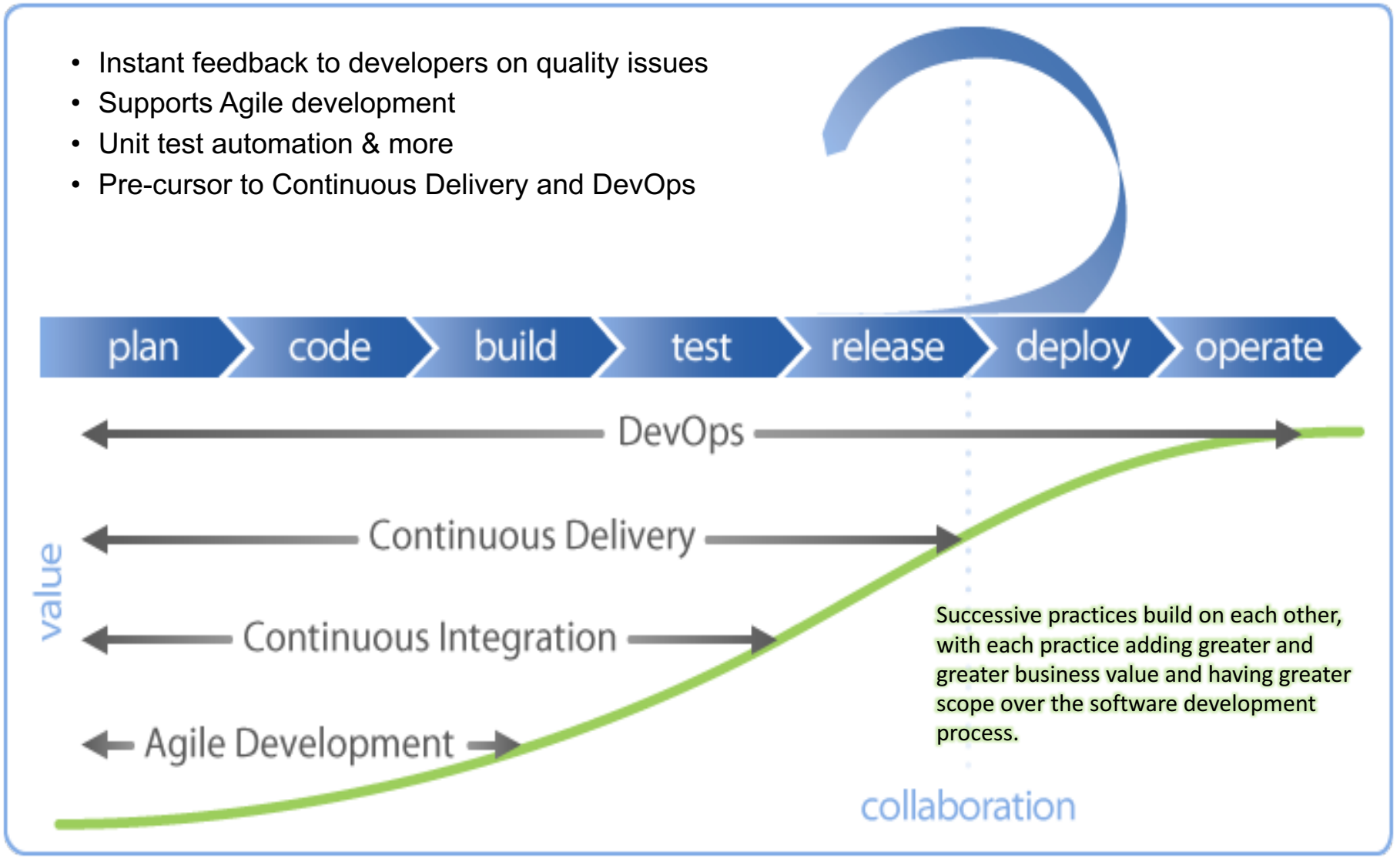
- Exposed functionality invokes program cross-LPAR over DPL
- Develop and unit test in isolation using **Rational Development and Test Environment for System z**
- Use **Rational Test Workbench** to automate functional/regression tests
- Before linked program available use Service Virtualization to sandbox on mainframe



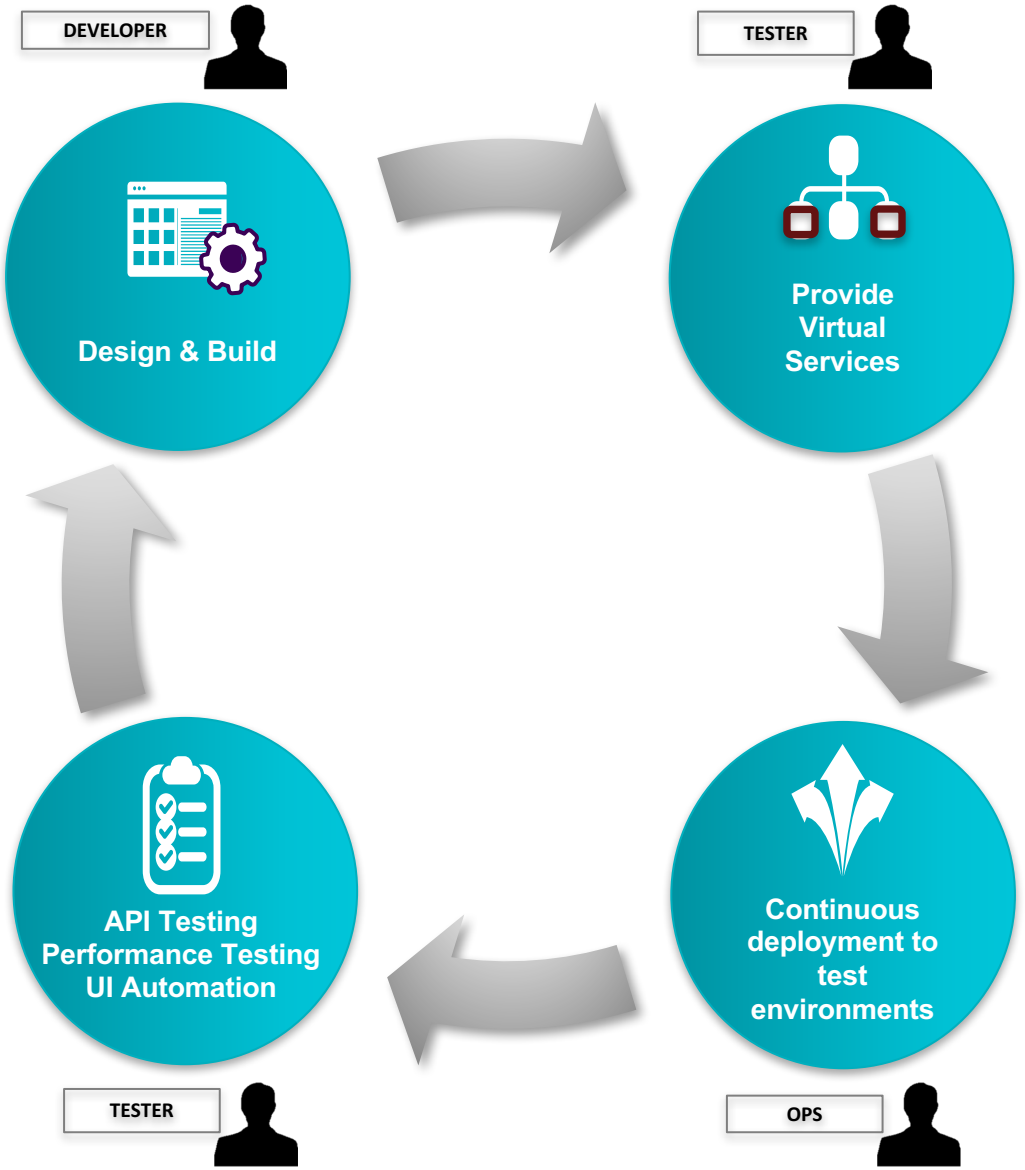
Adoption

The journey – continuous integration

- Instant feedback to developers on quality issues
- Supports Agile development
- Unit test automation & more
- Pre-cursor to Continuous Delivery and DevOps



Without Continuous Testing, there is no DevOps



Shipping code faster without testing means that bad code can be shipped faster. Testing is an integral part of DevOps.

Using service virtualization increases the ability to test early, while test automation provides rapid feedback on the quality of the software being shipped.

Resources

Continuous Testing on developerWorks at https://developer.ibm.com/testing/
Continuous Testing: An IBM point of view: http://www.ibm.com/developerworks/library/d-continuous-testing-shift-left-trs/index.html
DevOps ROI: https://developer.ibm.com/urbancode/docs/devops-managers-organization-leaders-making-roi-case-executives/
The Forrester Wave Modern Application Functional Test Automation Tools, Q4 2016: https://reprints.forrester.com/#/assets/2/581/'RES123866'/reports
Gartner Magic Quadrant for Software Test Automation: https://www.gartner.com/doc/reprints?id=1-3NWQQJB&ct=161214&st=sg
Gartner Magic Quadrant for Application Release Automation: https://www.gartner.com/doc/reprints?id=1-3FNG0BR&ct=160822&st=sg

Thank You


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