The DevOps approach: Develop and test – demo of the Continuous Integration Solution for System z development

Speaker Name and Title

Building a Continuous Integration for System z (CIz) Infrastructure for a DevOps Solution

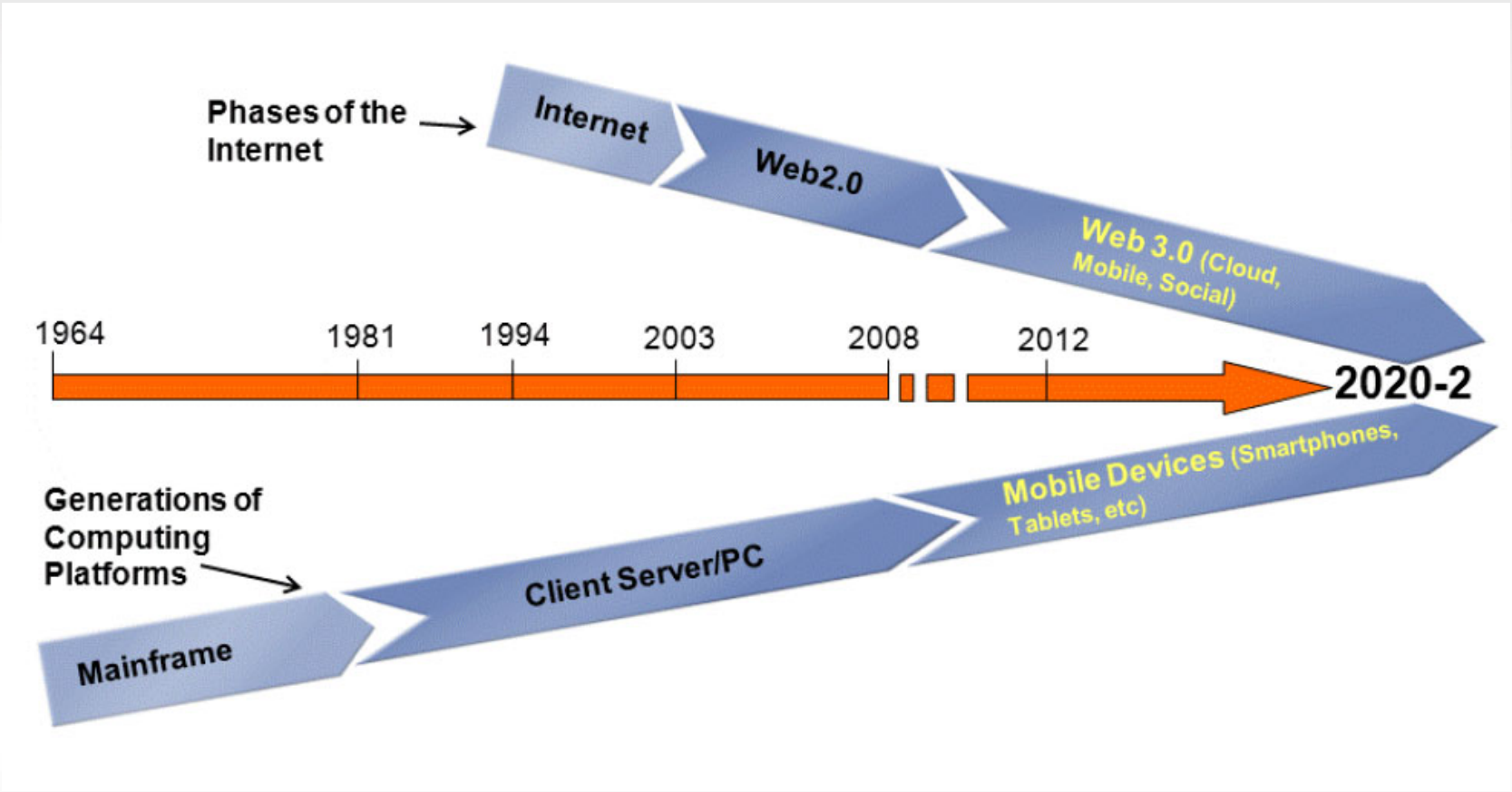
Session #5

In Session #4, we demonstrated how easy an automated continuous testing, a DevOps solution, can be in the System z environment. We showed how using continuous development and test can reduce Time-to-Market, Lower Defect Rate and Reduce Cost. Three major components of Continuous Integration for System z (CIz) are: Increasing System z resources to support unit testing, automated tooling and agility. In this session, we are going to focus on increasing System z resources through the use of Rational Development and Test (RD&T) and the virtualization of required resources through Rational Test Workbench (RTW). We will demonstrate the virtualization of a CICS DPL (Distributed Program Link) transaction in our CICS mortgage application.

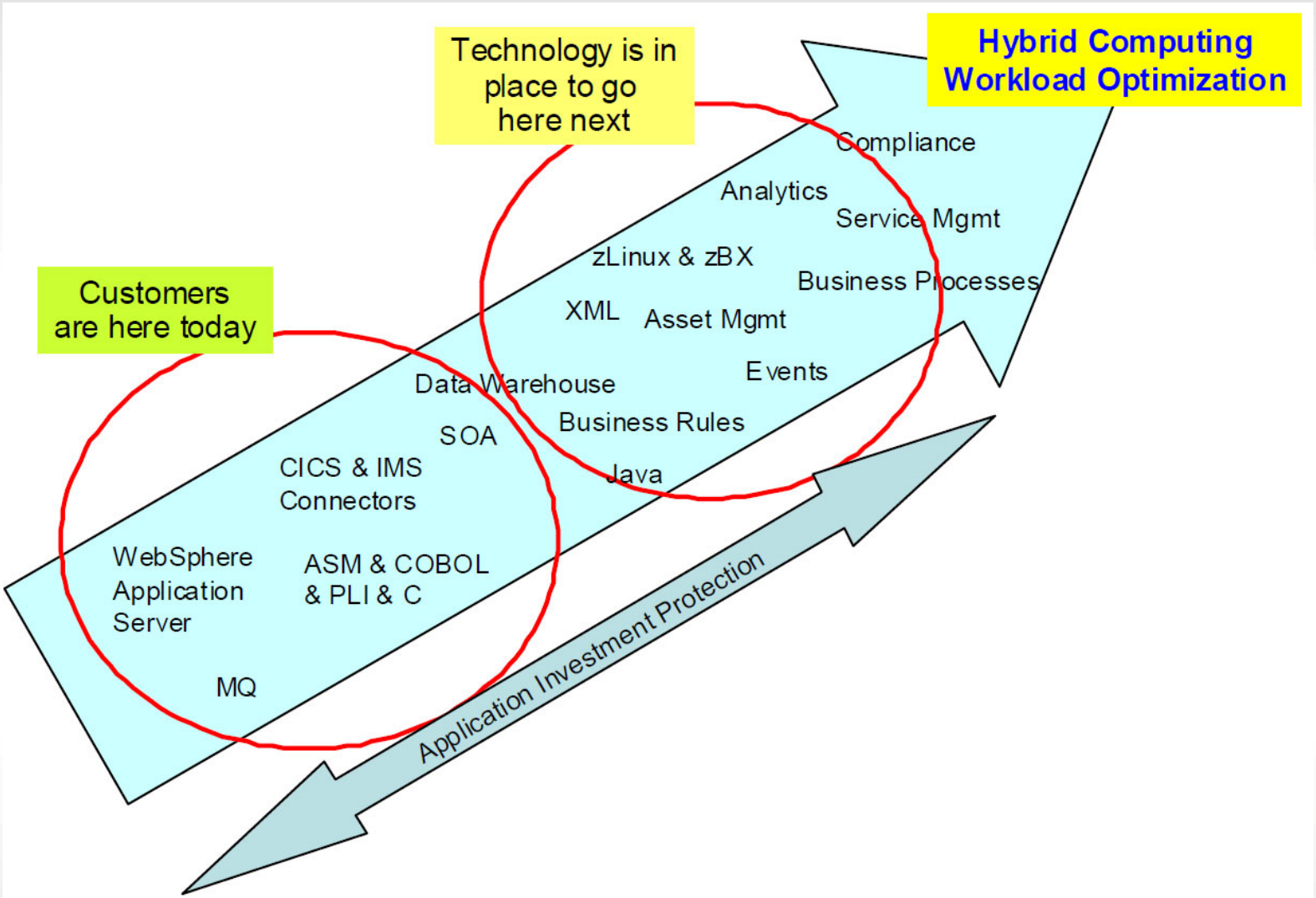
Agenda

- Trigger Driving Transformation and System z Technology
- DevOps Concepts and Increasing System z Resources
- Overview of a Modern Enterprise Data Center Architecture
- CICS Integration Overview
- CIz and Rational Test Virtualization Server (RTVS)
- Demo of CICS DPL Virtualization using RD&T
  - Architecture of Mortgage Application
  - Demo
- Questions

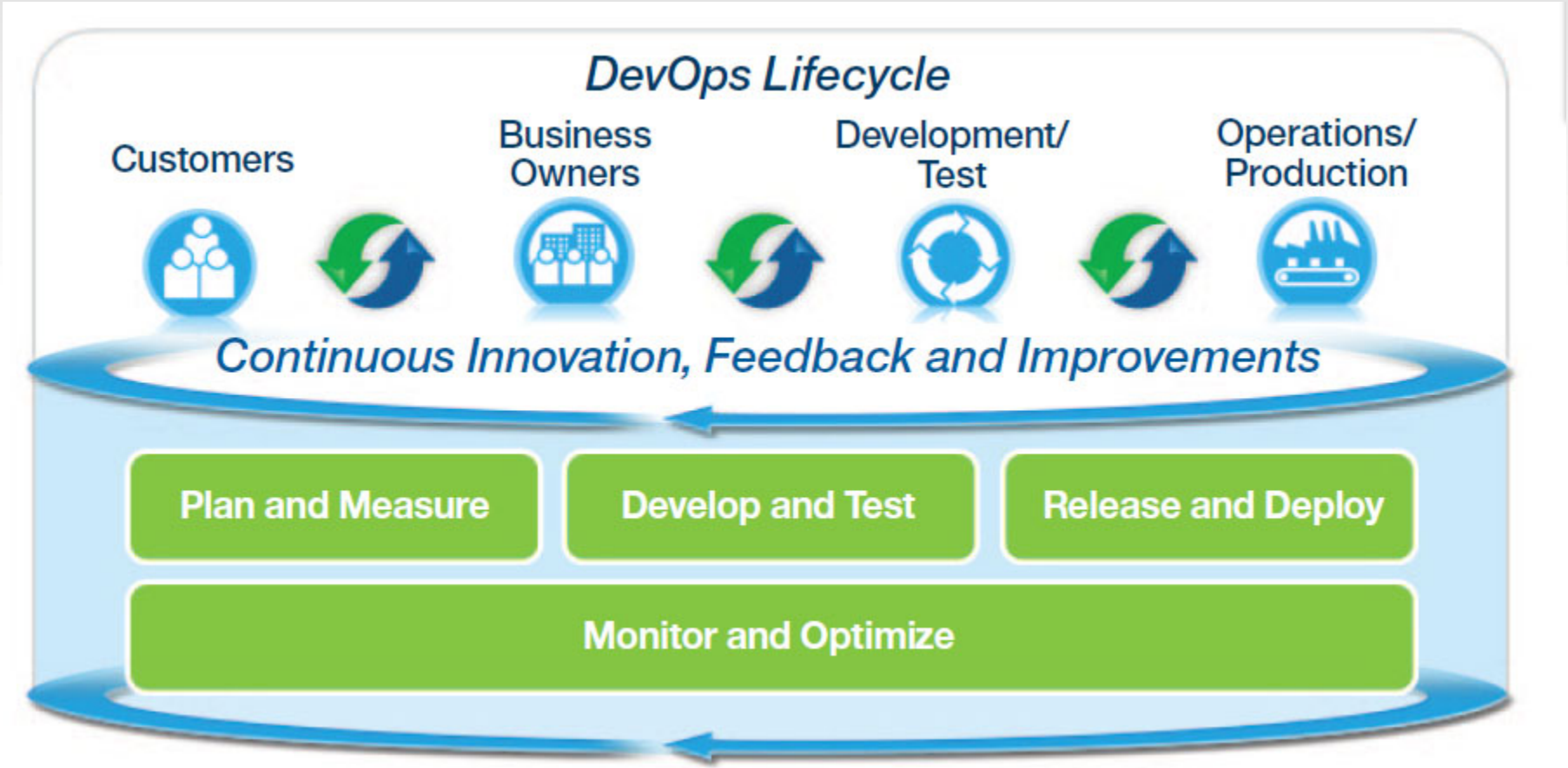
Trigger Driving Transformation



Technology is in Place on System z for Business Application Modernization



DevOps Reference Architecture



Challenges Meeting Business Time Pressures with Quality Software

*34% of all new IT Projects deploy late**

41% experience delays in integration, configuration and testing of applications

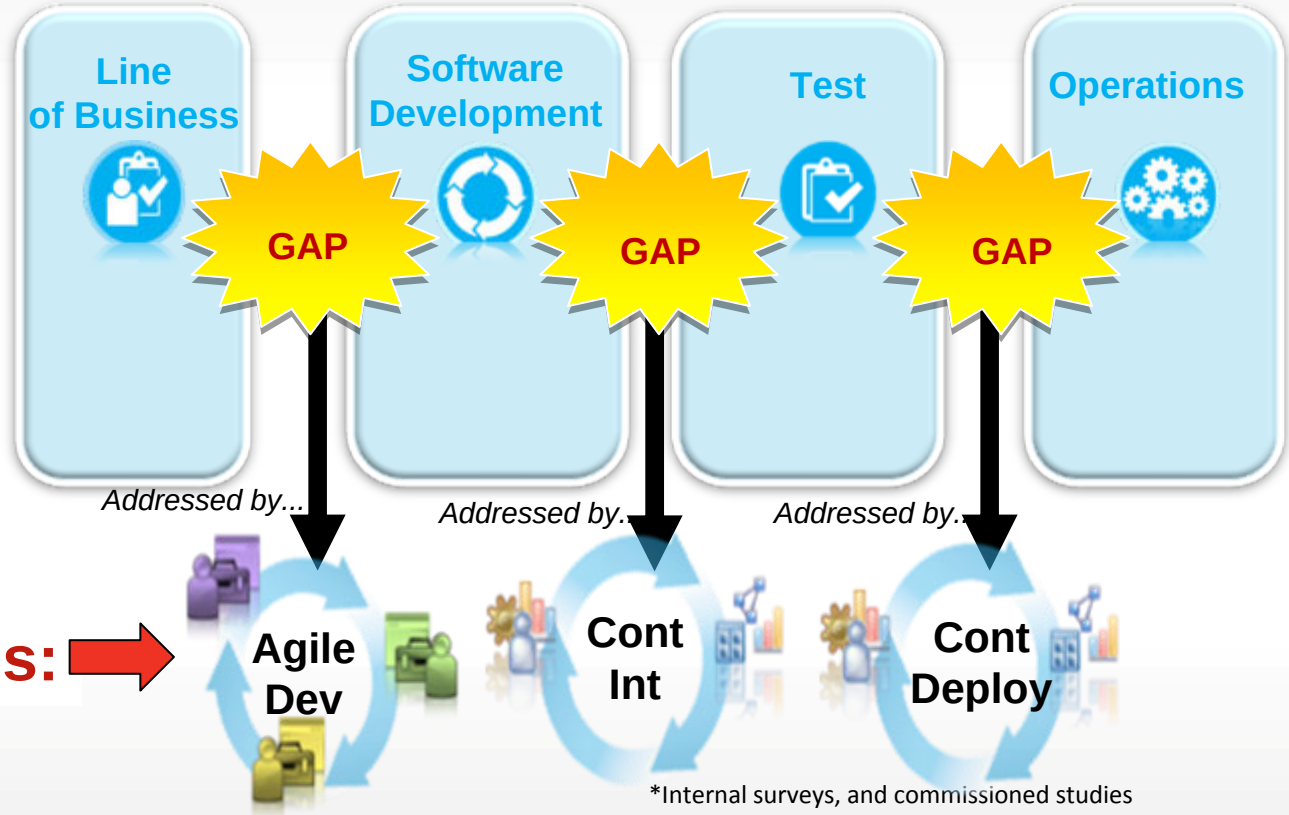
51% applications rolled back due to quality issues escaping into production

45% experience delays due to troubleshooting and fine-tuning issues in production

Up to

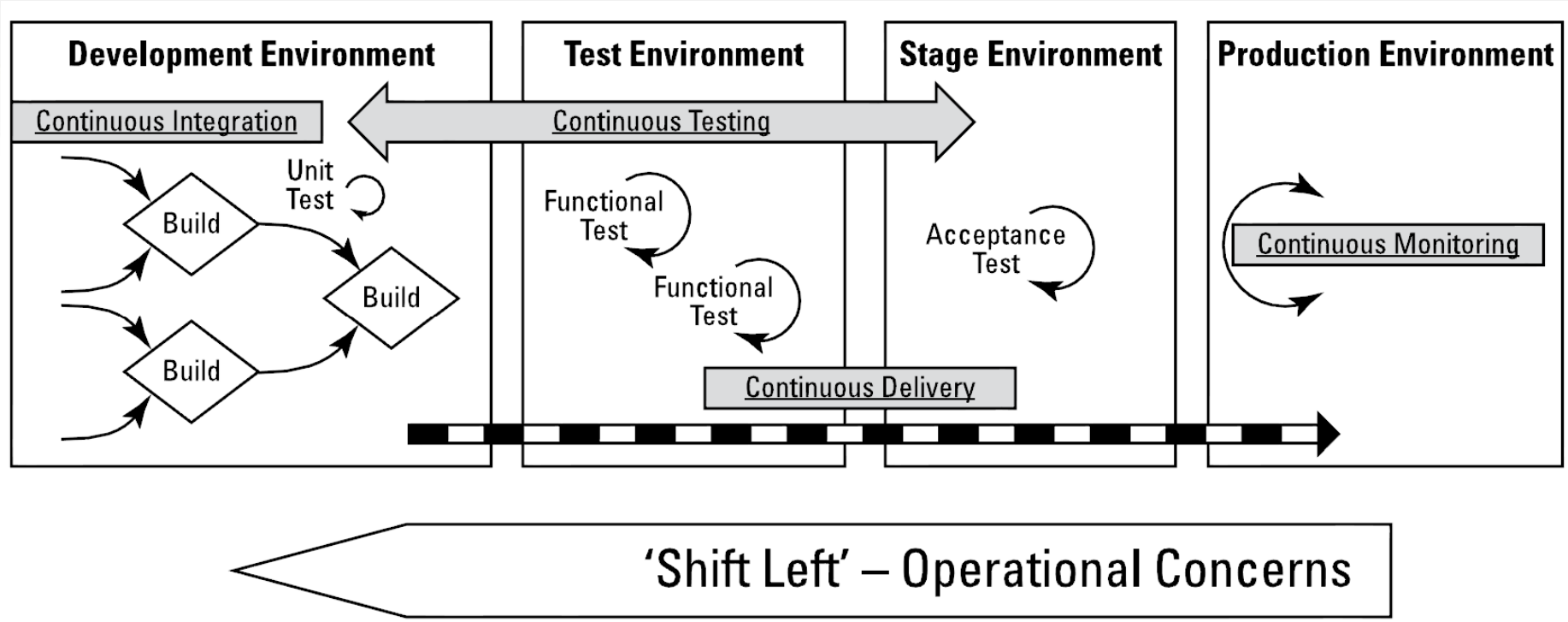
4-6 Weeks

to deliver a simple change

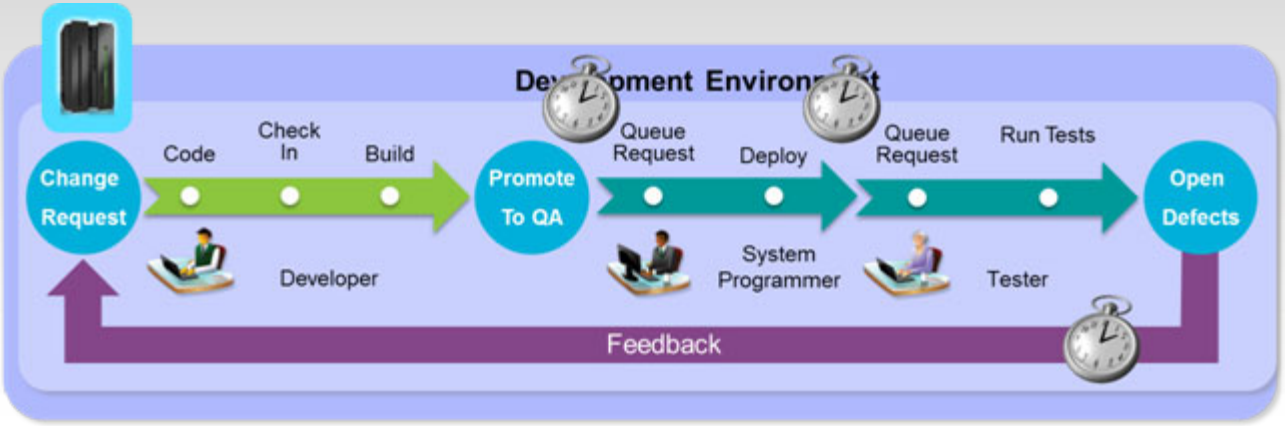


*Internal surveys, and commissioned studies

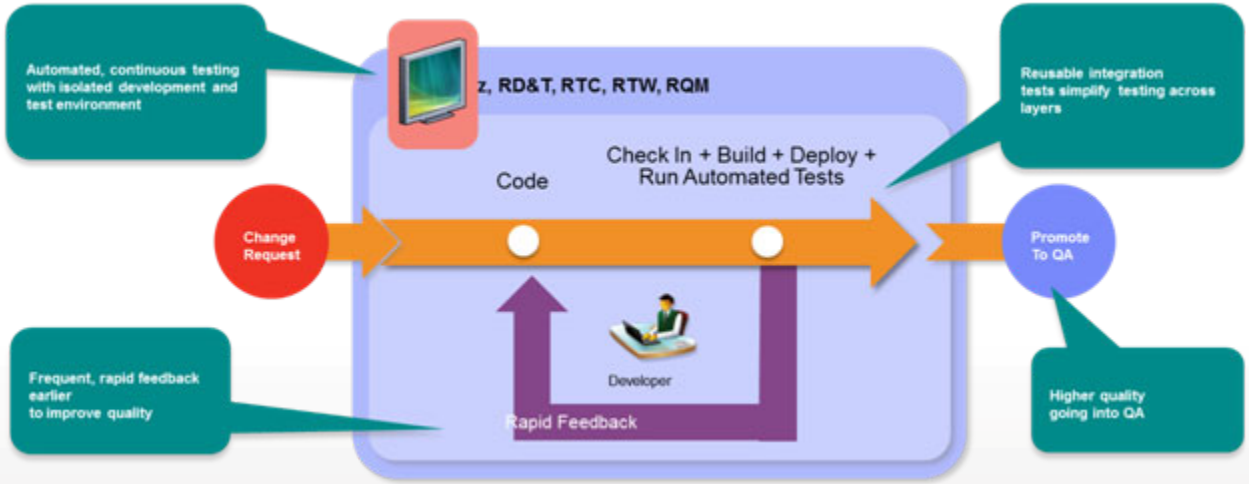
Dev Ops shift-left concept moves operations earlier in the development life cycle



Net Results of Clz



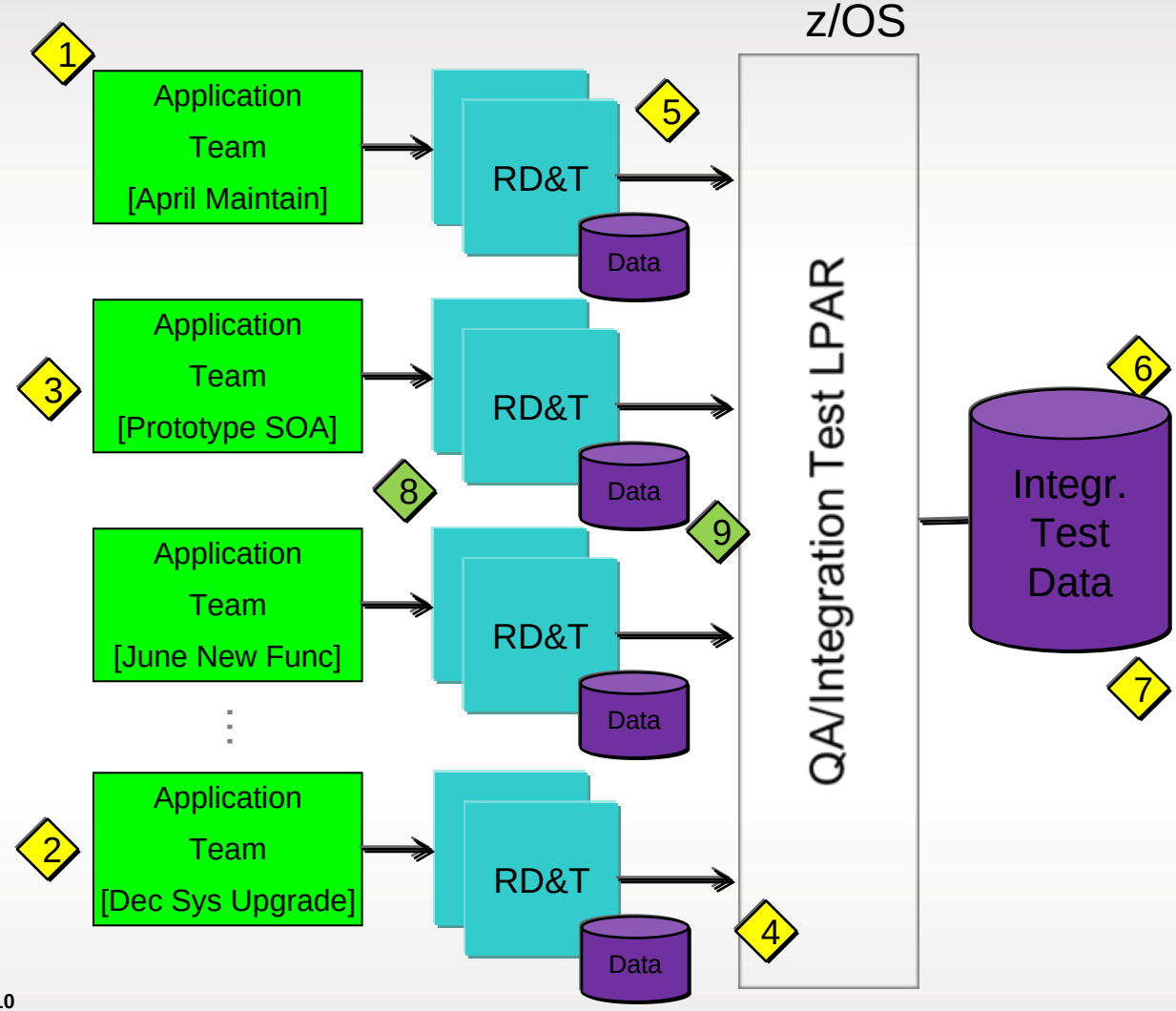
Waterfall development becomes Clz



Increase System z Resources

Organized Testing for Flexibility and Quick Delivery

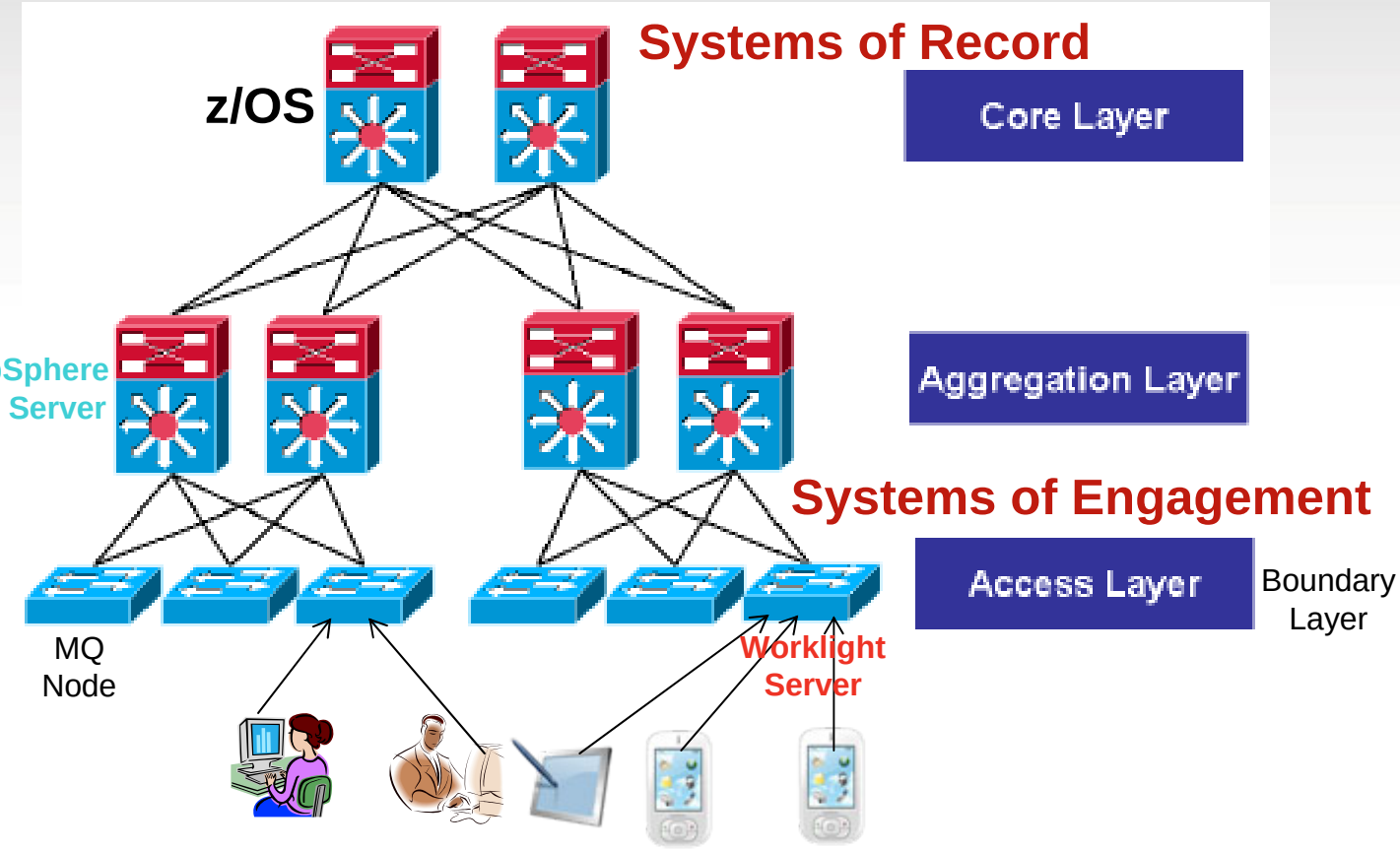
Organized by application team, horizontally sliced, dedicated resources, highly automated



Problems Encountered

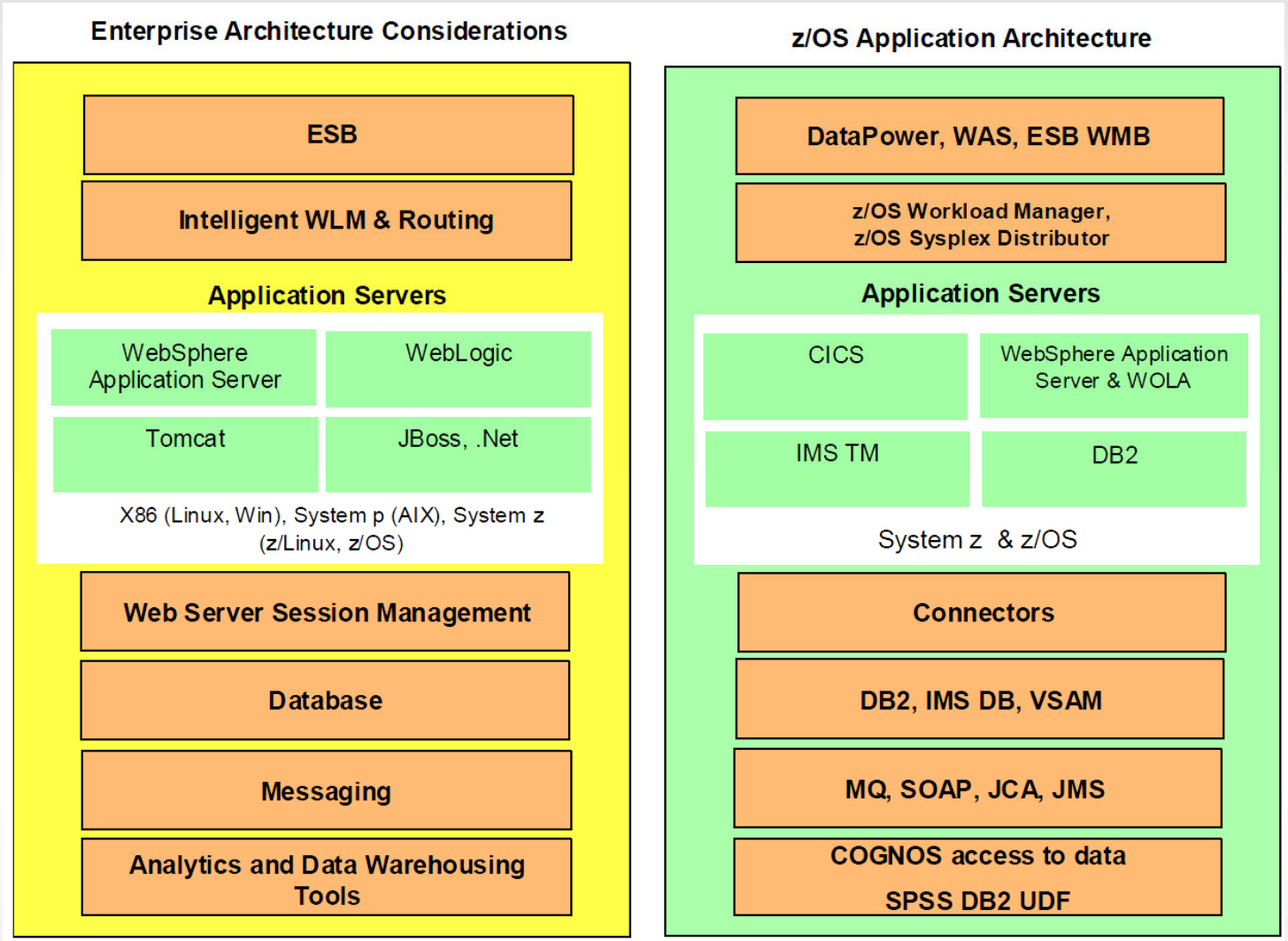
1. Teams compete for resources
2. Teams schedules cause overhead in prioritizing work
3. Lack of sandbox environment inhibits innovation and slows initial code delivery
4. Coordination of environmental changes causes bottlenecks at LPAR management
5. Coordination of release components leads to more rework and fewer releases
6. Shared test data takes time to coordinate.
7. Data is one-size-fits-all leading to over-testing

Today's Enterprise Computing Environment

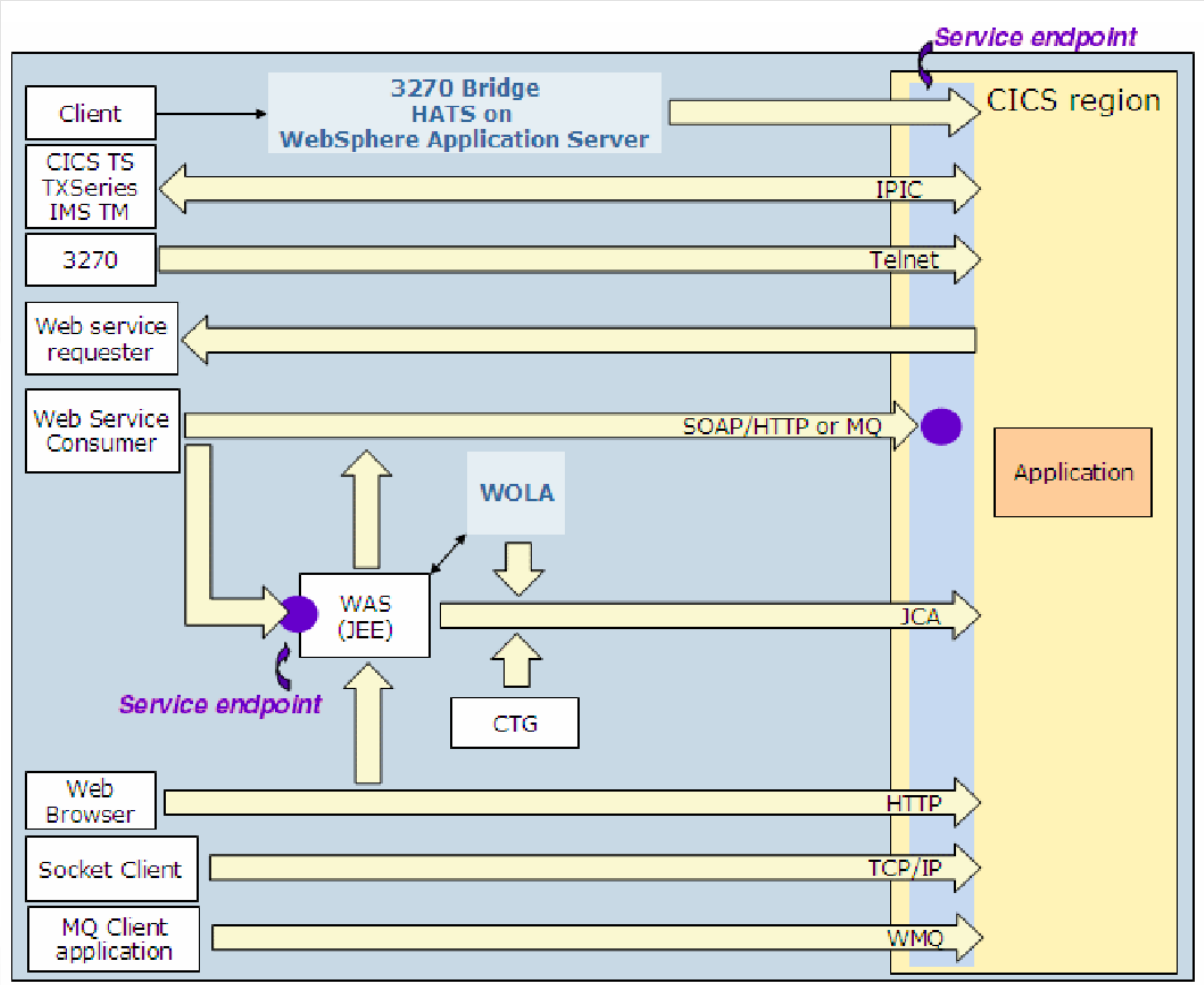


*** DevOps is the continuous integration, test and delivery in this hybrid environment

Enterprise Architecture Components Mapped to Application Support Facilities

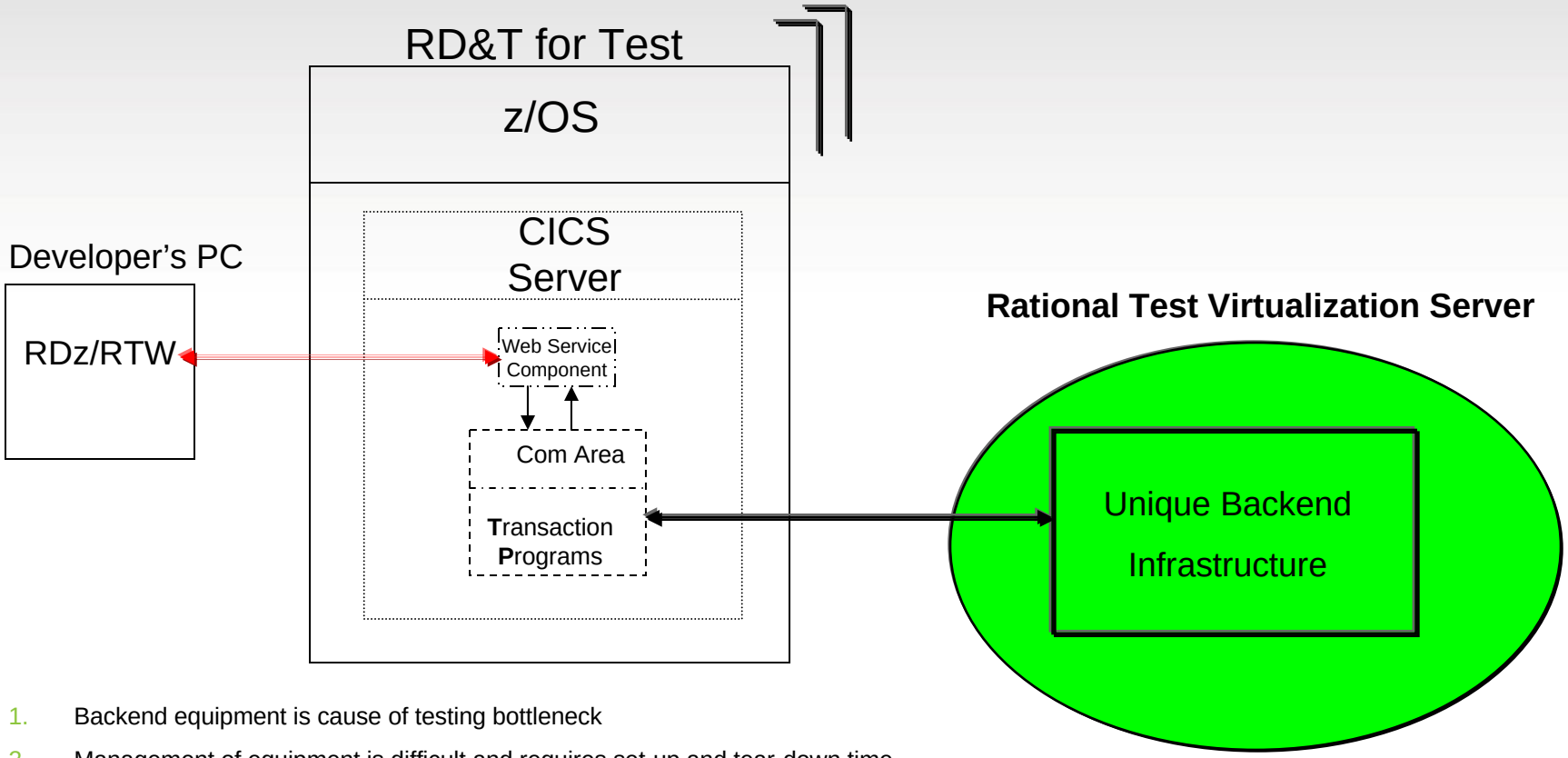


CICS Integration Overview



Need to support many different types connections for unit testing

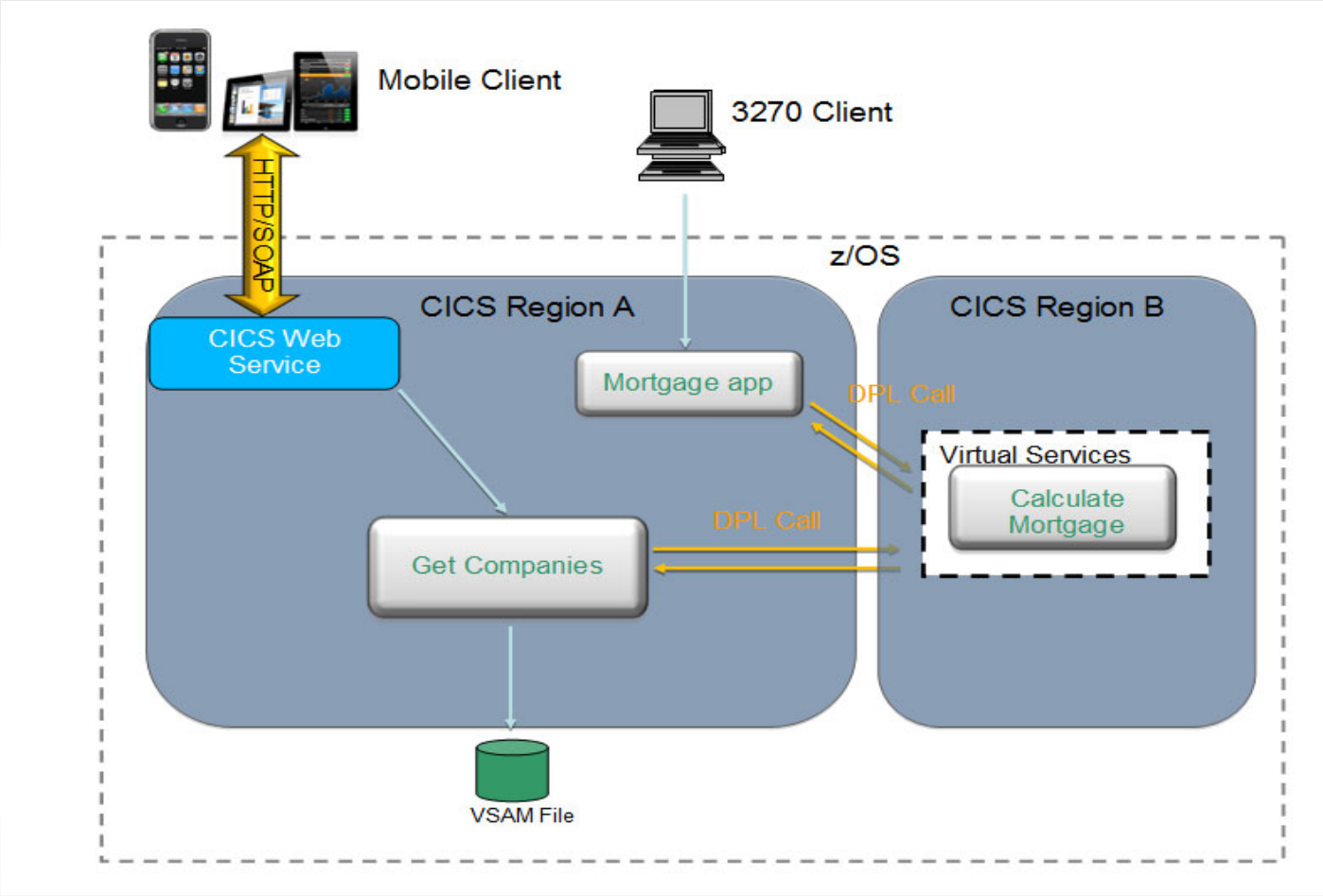
CIz and Rational Test Virtualization Server



1. Backend equipment is cause of testing bottleneck
2. Management of equipment is difficult and requires set-up and tear-down time
3. This can be an inhibitor to testing
4. Rational Test Virtualization Server can resolve infrastructure issues


DEMO

A Sample of Using CICS DPL Virtualization



Questions