Software Group

Modern Application Architectures for COBOL Developers - An Introduction

Agenda

- **COBOL Today**
- **Service Oriented Architecture**
  - Introduction
  - Challenges for System z Customers
  - Strategies
- **SOA and the System z Application Lifecycle**

COBOL Today and the future

- **COBOL (COmmon Business Oriented Language)**
  - The predominant programming language of business applications for over 40 years
  - Specifically designed for business applications
    - Two million programmers write up to 5 Billion lines of COBOL code every year.

- **The following factors are some of the reasons that COBOL continues to maintain its reign as the predominant programming language for commercial business applications.**
    - Strong presence of COBOL vendors
    - Modern COBOL extensions to existing COBOL applications
    - COBOL's ease of use and ease of comprehension reduces documentation and learning costs.
    - Continues to be popular and its use is growing
    - IBM continues to deliver value in its COBOL compiler products.
    - COBOL is easy to learn and maintain over time, with or without formal training.
    - The mainframe delivers superior operational efficiency due to its centralized design.
      - Offloaded applications would increase the costs of operations
      - Effort of offloading applications off the mainframe is risky and expensive.
      - Migrating COBOL off the mainframe can cost $25 per line of code (Network World Oct 20, 2003).

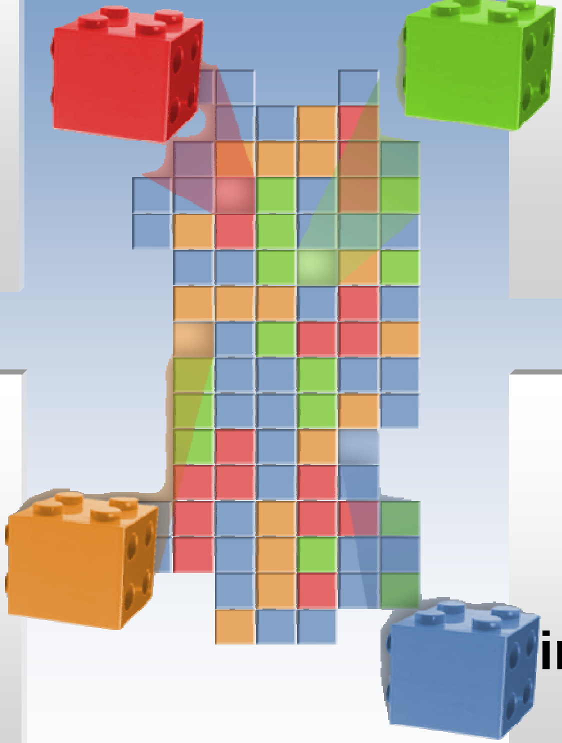
What is Service Oriented Architecture (SOA)?

… a service?

A **repeatable business task** – e.g., check customer credit; open new account

… service orientation?

A way of integrating your **business as linked services** and the outcomes that they bring



… service oriented architecture (SOA)?

An IT **architectural style** that supports service orientation

… a composite application?

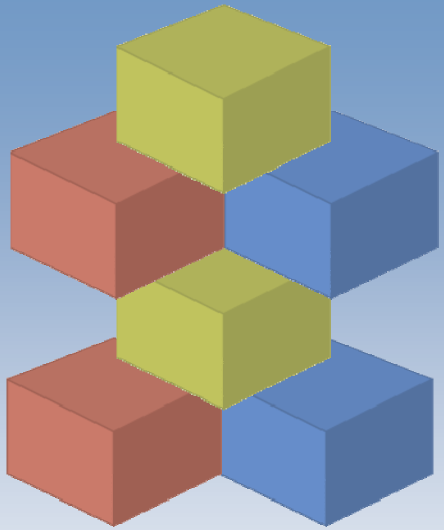
A set of **related & integrated** services that support a business process built on an SOA

SOA: The focus is on Flexibility and Reuse

Business Perspective

Modern UI’s linked with Business Process

- Orchestrated sequence of
- Activities
- Separated elements
  - Activity sequence
  - Activity hand-off
  - Activity content



IT Perspective

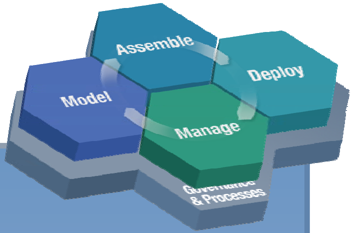
Web User Interfaces and Composite Application

- Orchestrated flows of Services
  - Tooling
- Separated logic
  - Process flow
  - Connectivity
  - Business
- Flexible high QOS Business Functions

Why Service Oriented Architecture? …

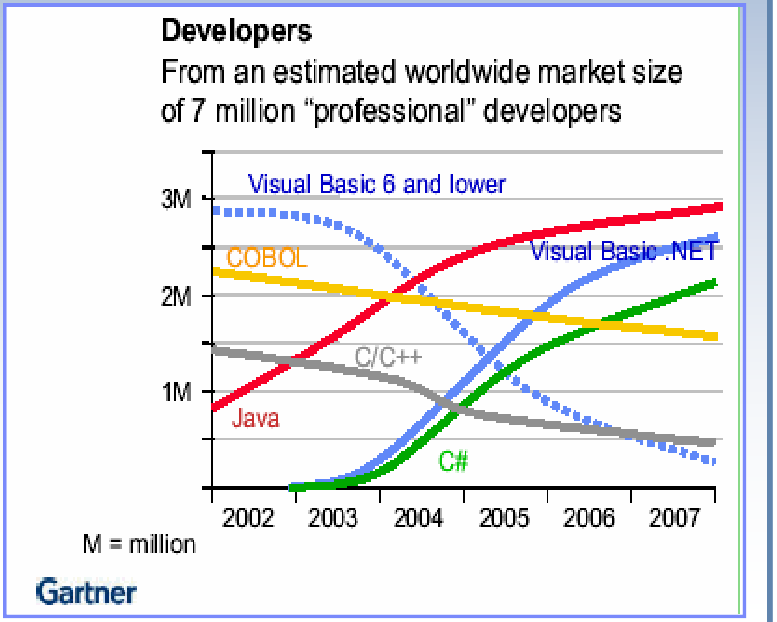
- Enables re-use of existing assets
- Enhances system flexibility through logic isolation
- Supports simplified integration of new assets with existing assets

What about “before SOA”?



- **Significant business intelligence exists in core systems**
  - "200 Billion lines of COBOL code in existence" ***eWeek***
  - "5 Billion lines of COBOL code added yearly“ ***Bill Ulrich, TSG Inc.***
  - “2 Million COBOL developers" ***Gartner***
  - "Majority of customer data still on mainframes“ ***Computerworld***
  - “Replacement costs $20 Trillion” ***eWeek***

- **Rewriting - is it an option.....**
  - How long will it take? (lose strategic benefit)
  - Who will do it? (who has the business knowledge?)
  - How much will it cost?
  - Risk?



Developers

From an estimated worldwide market size of 7 million “professional” developers

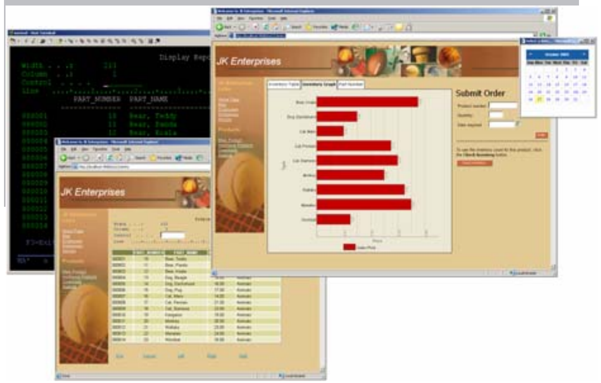
M = million

Gartner

Three Styles of Application Transformation

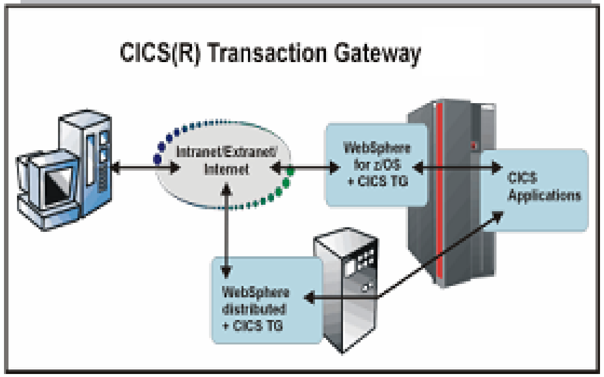
Transform User Experience

Enhance user interface and workflow for quick return on investment



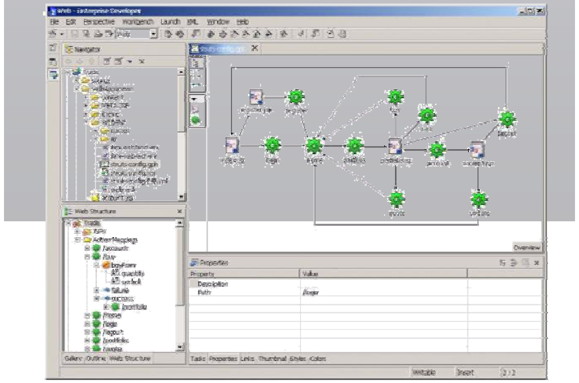
Transform Application Connectivity

Improve business processes and develop customer, partner and supplier relationships using Web services and Java connectors


CICS(R) Transaction Gateway
Intranet/Extranet/ Internet
WebSphere for z/OS + CICS TG
CICS Applications
WebSphere distributed + CICS TG

Transform Application Architecture

Update and extend mission-critical applications as services, leveraging their core value in new ways



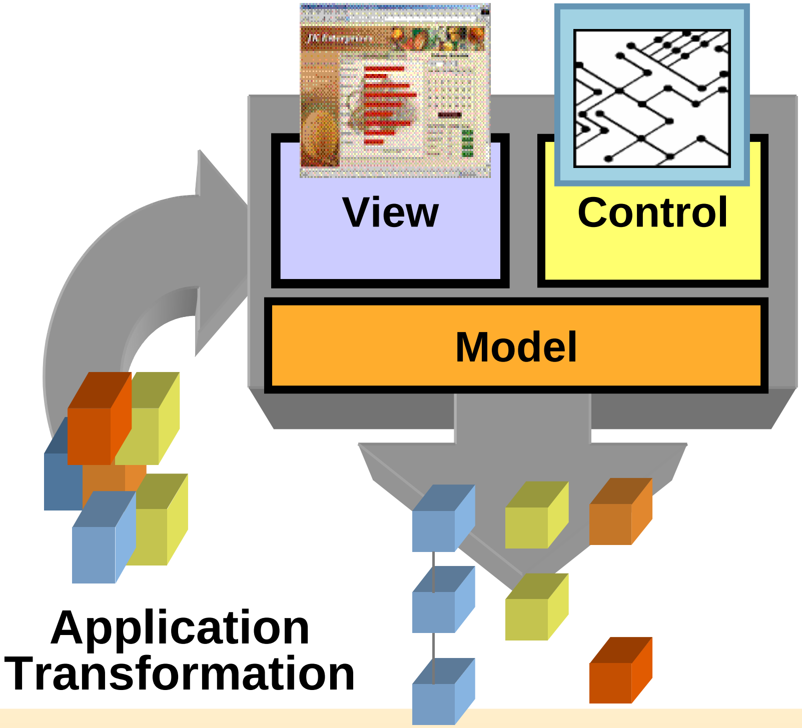
Single integrated delivery vehicle across application transformation styles

Three styles of Application Transformation

Transform User Experience

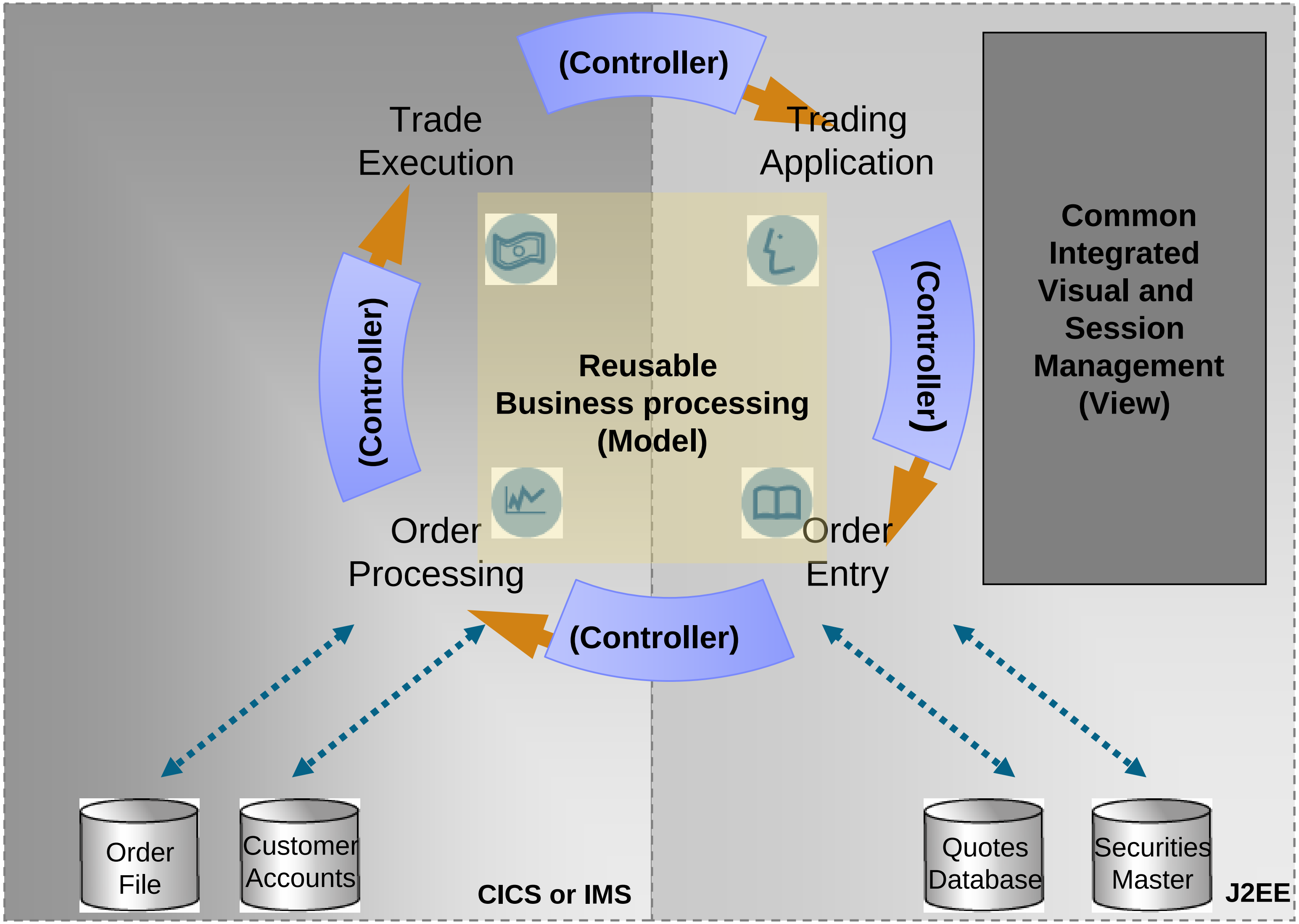
Transform Application Connectivity

Transform Application Architecture

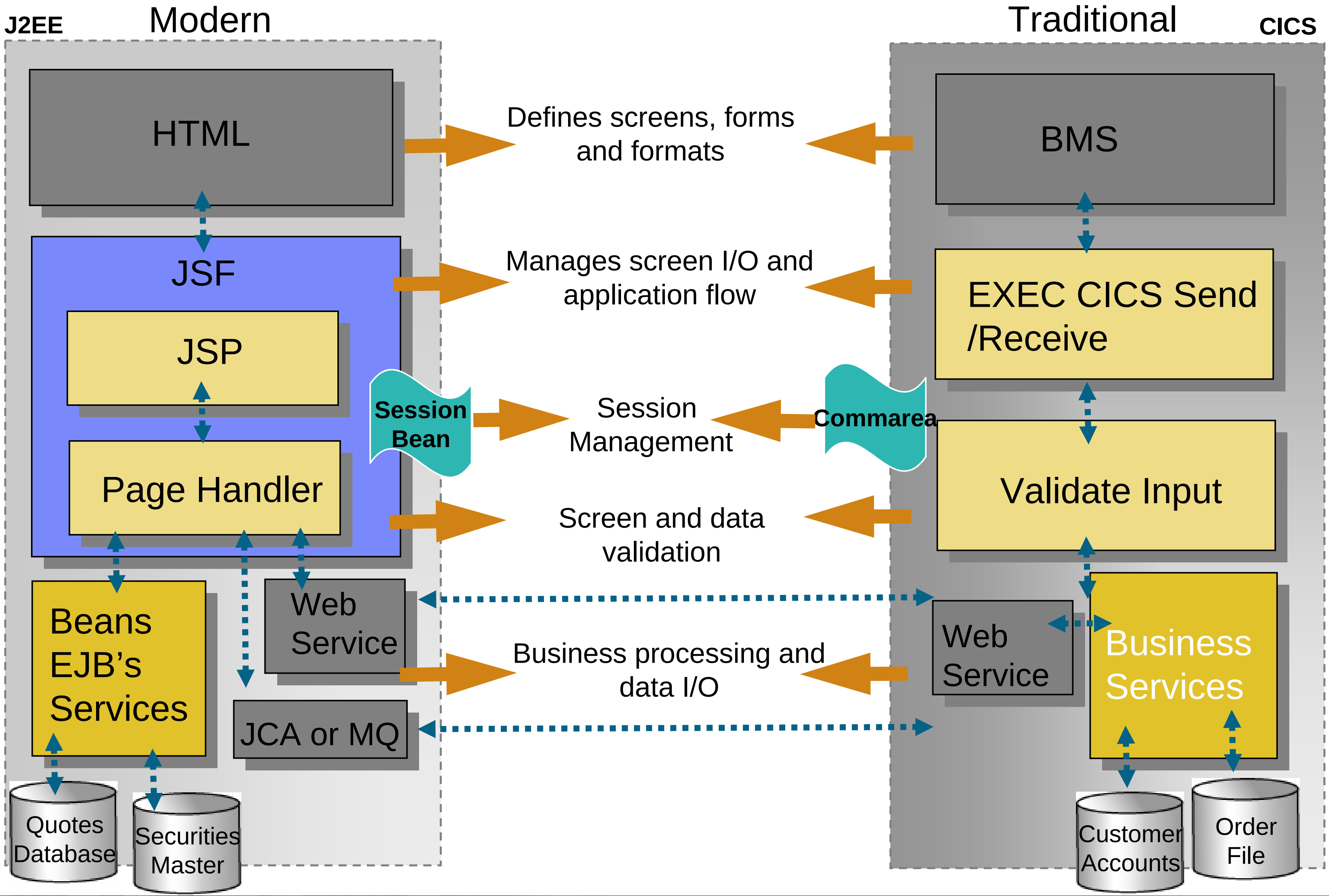


Single integrated delivery vehicle across application transformation styles

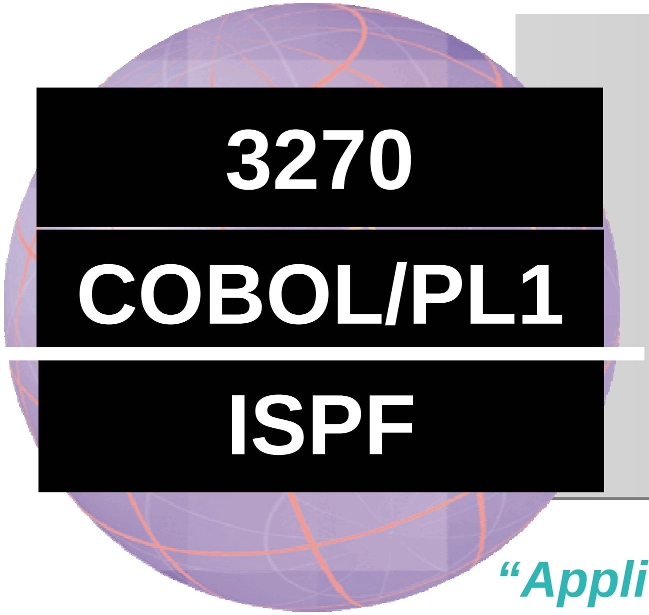
Composite Workload Application Components



It’s not that different



Investment Challenges



3270

COBOL/PL1

ISPF

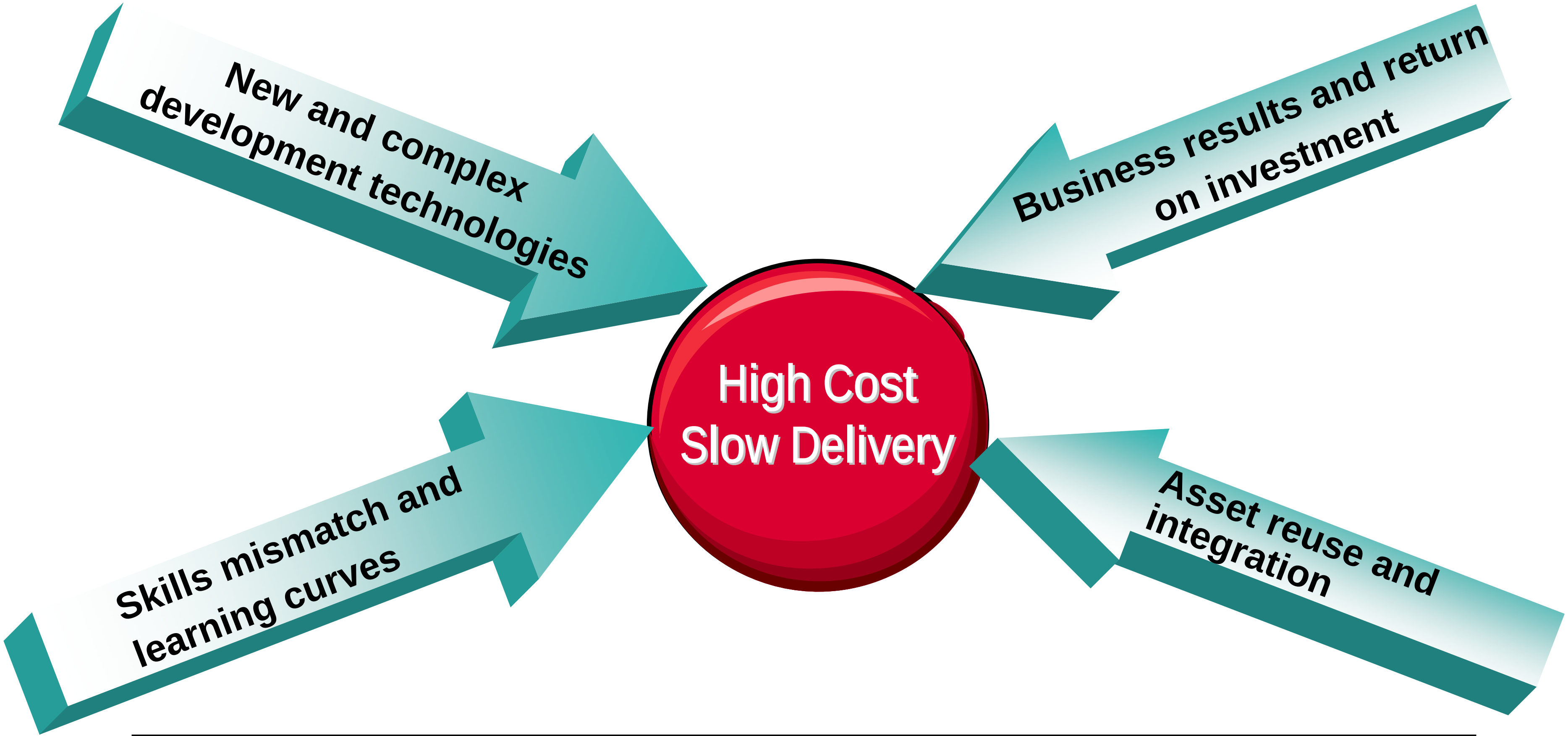
- **Many zSeries developers still:**
  - Focused on creating or enhancing 3270 applications
  - Using traditional, host-based development environment

"Application maintenance consumes between 60 – 80 percent of IT budgets" - Phil Murphy, Forrester

Issues: How do I?

- Increase productivity of business developers working on traditional applications
- Enabling broad business developer community in SOA and Web Based infrastructures
- Improve Time to market and IT responsiveness

Technology Challenges



New and complex development technologies

Business results and return on investment

High Cost Slow Delivery

Skills mismatch and learning curves

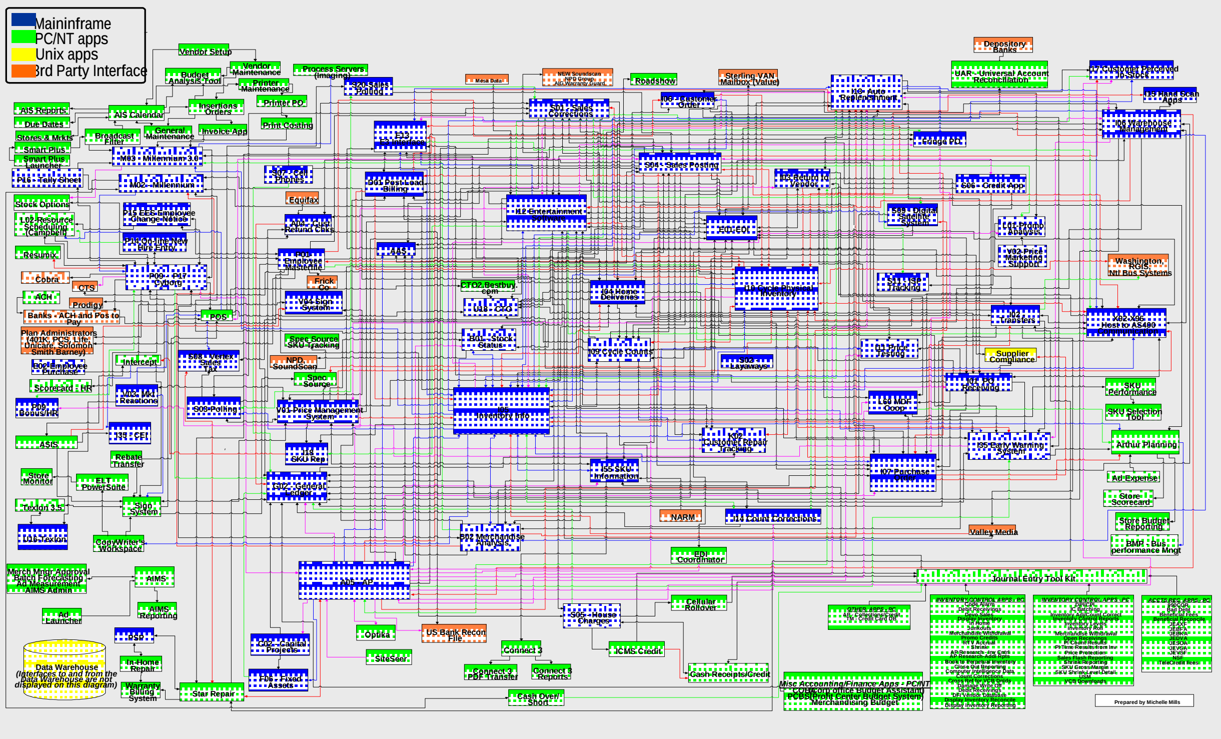
Asset reuse and integration

Issues: How do I?

- Enable experts on Core Applications in modern technologies
- Leverage business skills
- Create the SOA infrastructure without throwing everything else away

Architectural Challenges

- Application dependencies are extraordinarily complex, and exist at multiple levels
- Dependencies cross technologies and environments
- Need to support application maintenance, development and test
- Need to support application integration and service / component creation



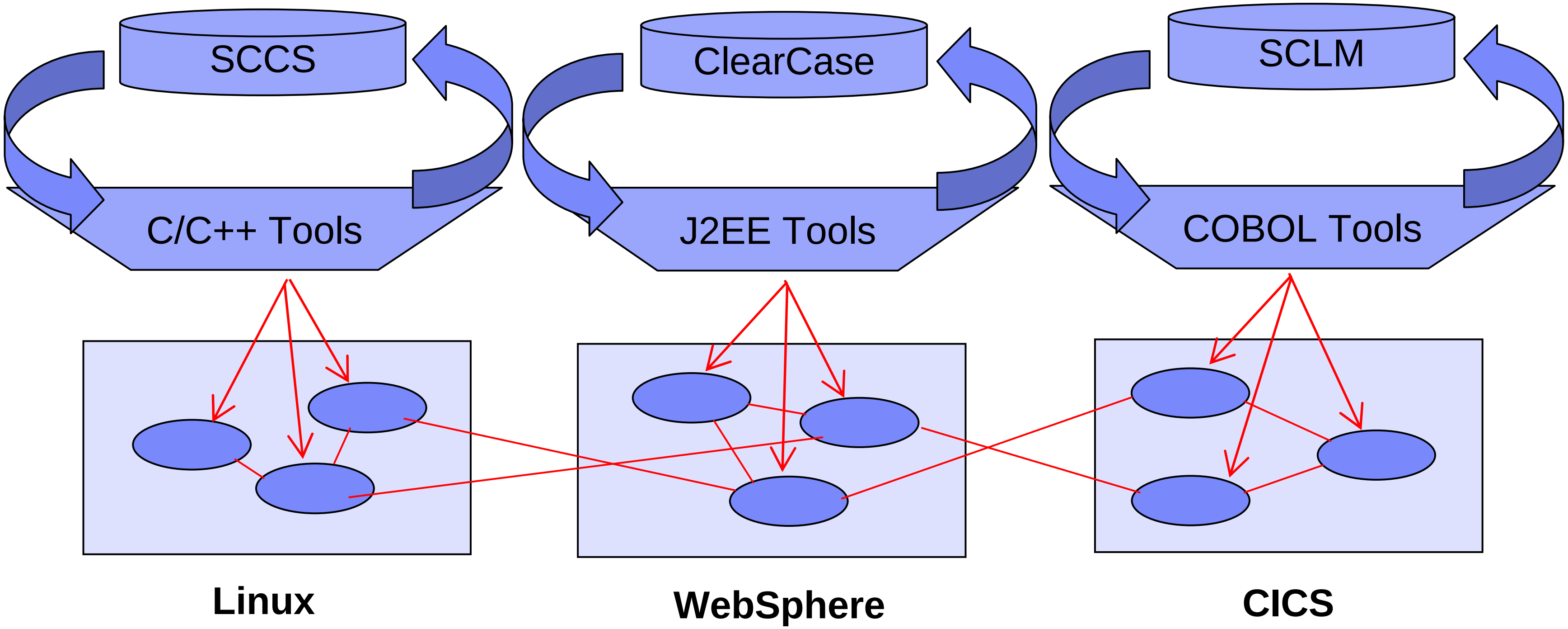
Actual Application Architecture for Consumer Electronics Company

Issues: How do I?

- Improve application backlog and throughput of requirements
- Avoid unplanned impacts – manage quality - during change cycles
- Enable rapid reuse

Organizational Challenges

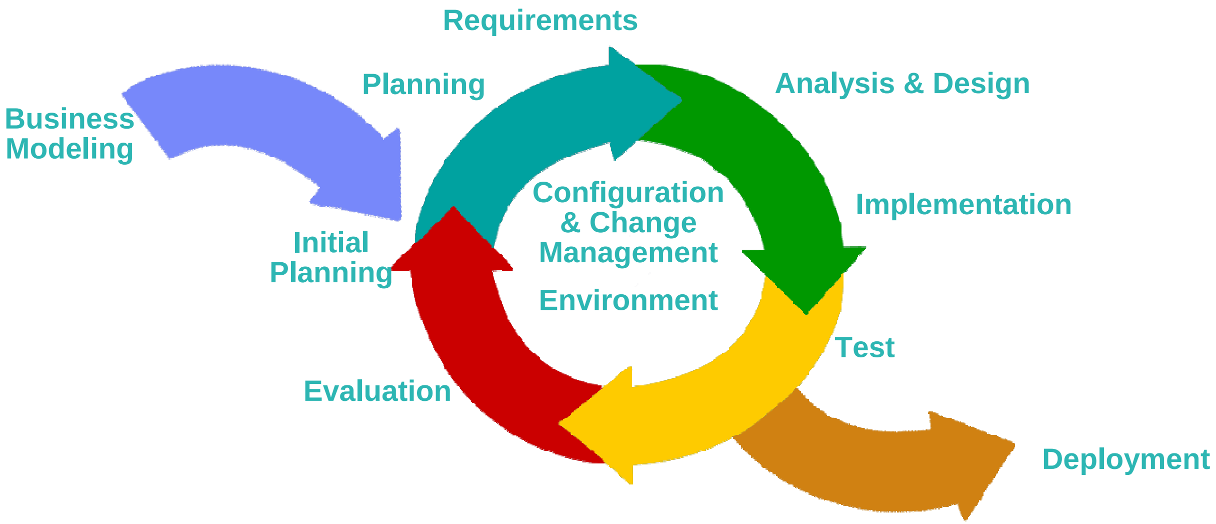
- Lack application components & skills sharing
- Ineffective / Uncoordinated development of integrated application



Issues: How do I?

- Manage change across geographically distributed development teams
- Communicate available services and resources
- Leverage existing code – and process – at the same time improving quality

Strategy 1 - Bring iterative model driven development paradigms to composite applications



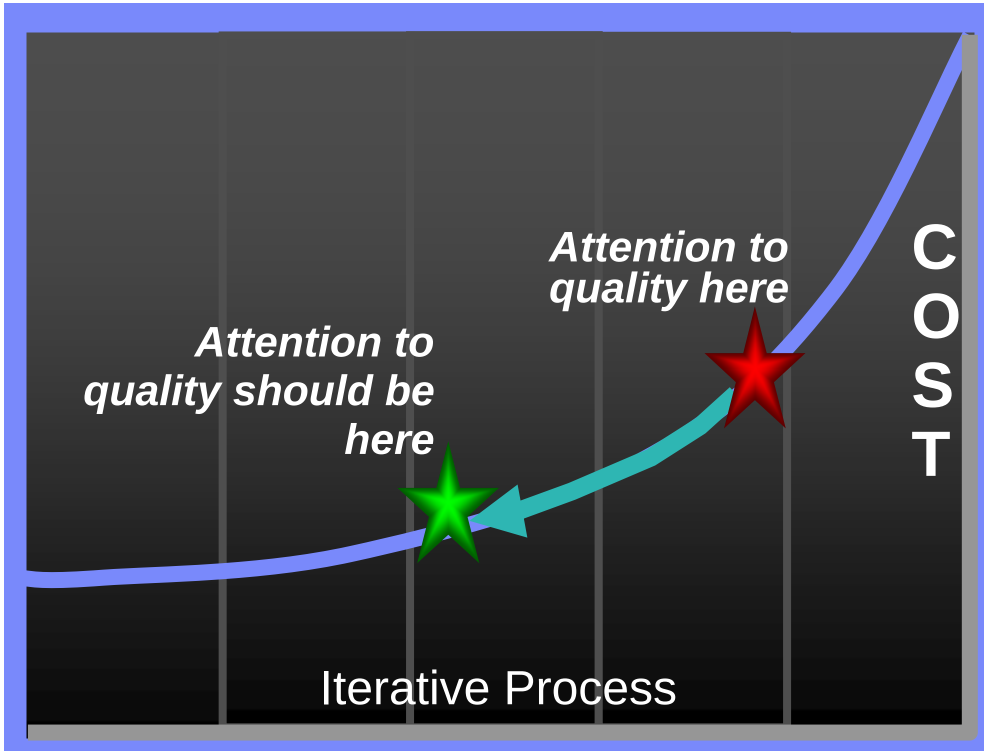
- Adopt a flexible process for both J2EE & traditional z/Series applications
- Tools integration across the lifecycle (Model and Discover, Develop and Assemble, & Deploy and Manage)
- Manage mixed workload requirements

Issues: How do I?

- Leverage modern development techniques across broad developer organizations
- Generate complex SOA architectures, versus hand coding
- Improve documentation and speed the development to test cycle

Strategy 2 -Prevent, detect, diagnose and remove defects

- Improve application quality and test process
- Provide early warnings of activities susceptible to failure
- Analyze across disciplines to understand root causes

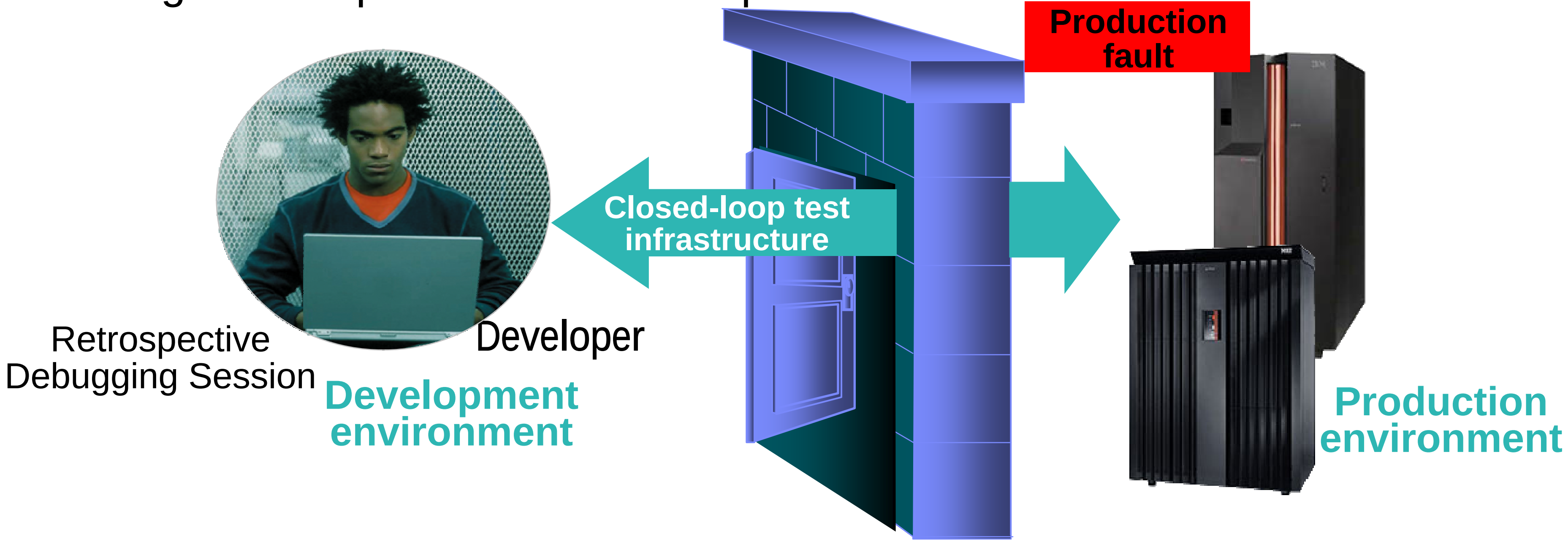


Issues: How do I?

- Find problems in development, before system test and production
- Debug SOA applications cross programs, platforms, languages, etc.
- Perform risk analysis on quality of deliverables

Strategy 3 - Reduce application downtime

- Find and fix errors post-deployment quickly
- Speed application rebuild and redeploy
- Bridge development teams and operation teams

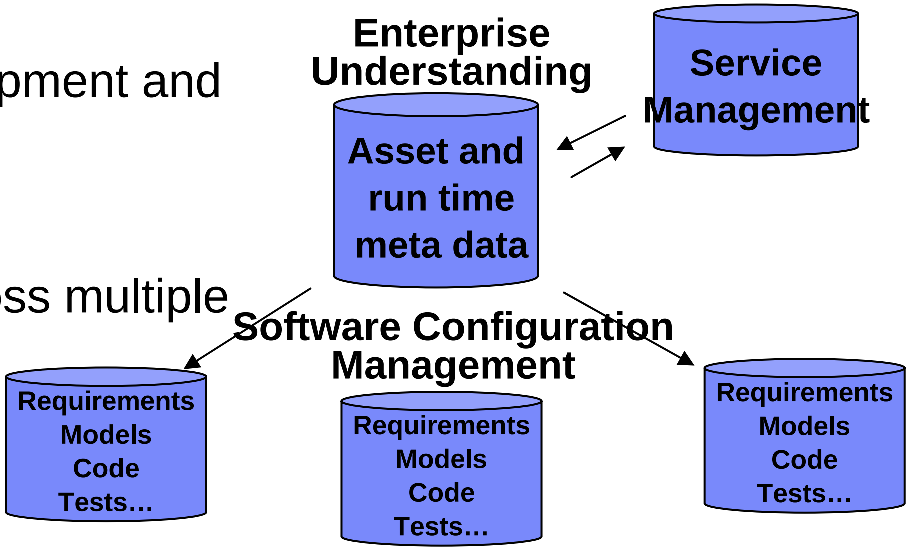


Issues: How do I?

- Manage quality in a SOA environment
- Solve applciation faults when multiple runtimes are involved
- Leverage business knowledge during problem determination process – i.e., common skills across developer bases

Strategy 4 - Manage change and assets as services

- Manage change across multiple development and operational environments
- Manage diverse assets
- Automate and accelerate workflow across multiple development teams


Enterprise Understanding
Asset and run time meta data
Service Management
Software Configuration Management
Requirements Models Code Tests…
Requirements Models Code Tests…
Requirements Models Code Tests…

Business Benefits

- Quickly respond to change
- Develop anytime, anywhere, in parallel
- Enable reuse and protect assets

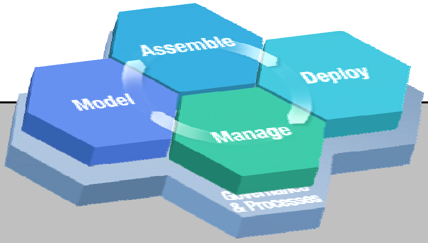
Technology Benefits

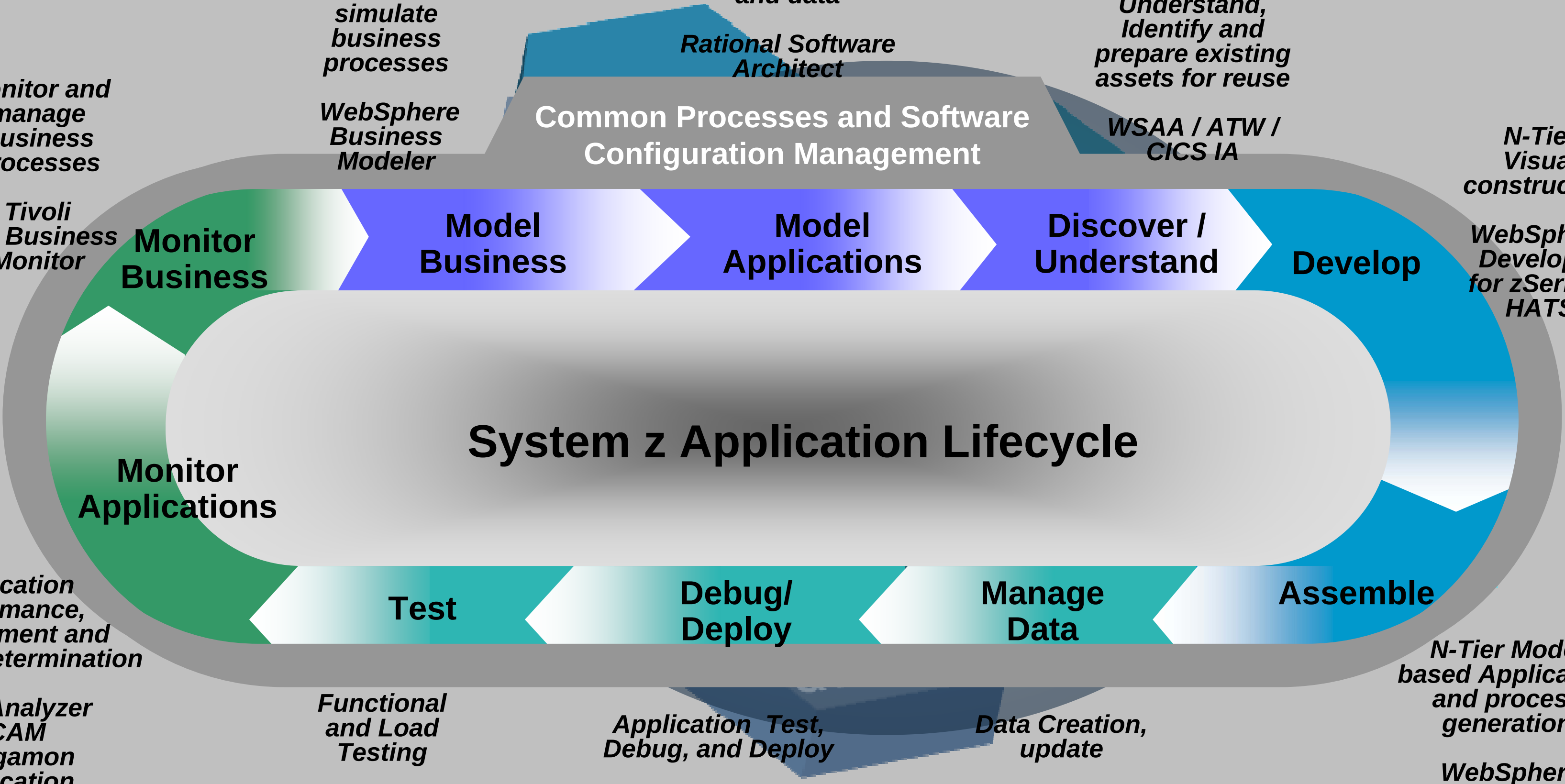
- Flexible workflow and process support
- Distributed team management
- Traceability across the lifecycle

Issues: How do I?

- Govern processes and enable reuse
- Track who is working on what
- Merge changes from multiple teams
- Support vastly increased numbers of artifacts across the lifecycle

System z Application Lifecycle





Common Processes and Software Configuration Management

Monitor Business

Model Business

Model Applications

Discover / Understand

Develop

Assemble

Manage Data

Debug/ Deploy

Test

Monitor Applications

System z Application Lifecycle

Monitor and manage Business processes

Tivoli WS Business Monitor

Model and simulate business processes

WebSphere Business Modeler

Model applications and data

Rational Software Architect

Understand, Identify and prepare existing assets for reuse

WSAA / ATW / CICS IA

N-Tier Visual construction

WebSphere Developer for zSeries / HATS

N-Tier Model based Application and process generation

WebSphere Integration Developer

Data Creation, update

File Manager

Application Test, Debug, and Deploy

Debug Tool Utilities

Functional and Load Testing

RPT/RFT

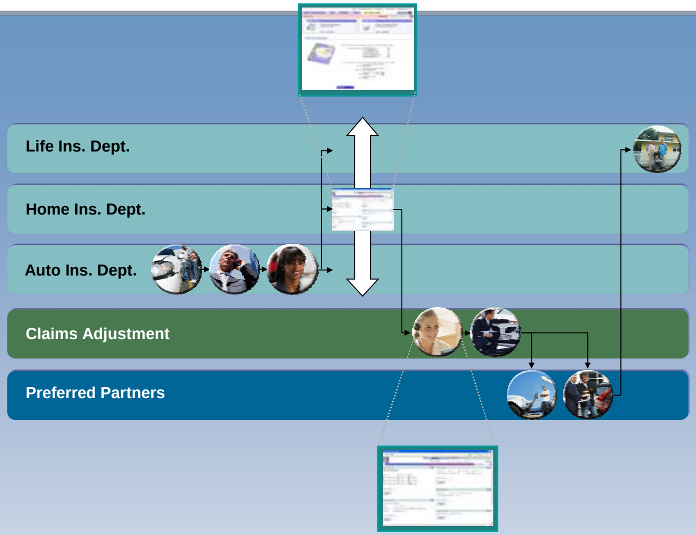
Application performance, management and problem determination

Fault Analyzer
ITCAM
Omegamon
Application Performance Analyzer

Enabling a robust, flexible SOA runtime environment

While maximizing the value of existing assets

Fully SOA capable!



WebSphere Application Server V6

- Extend existing Java assets with support for Web Services standards and standards-based messaging
- Help ensure 24x7 availability of business-critical applications with clustering and high availability
- Build and deploy Web Services quickly and easily with rapid development and deployment features

CICS Transaction Server V3.1

- Exploit provider/requestor Web service support for CICS assets, based on full Web service standards
- Extend the value of CICS transactions in a mixed language environment
- Build Web services from CICS transactions with no change to existing applications.

IMS Transaction and Database V9

- Exploit Web service support for IMS assets, based on full Web service standards
- Extend the value of IMS transactions in a mixed language environment
- Build Web services from IMS transactions with no change to existing applications

#1 in market share for Application Server software

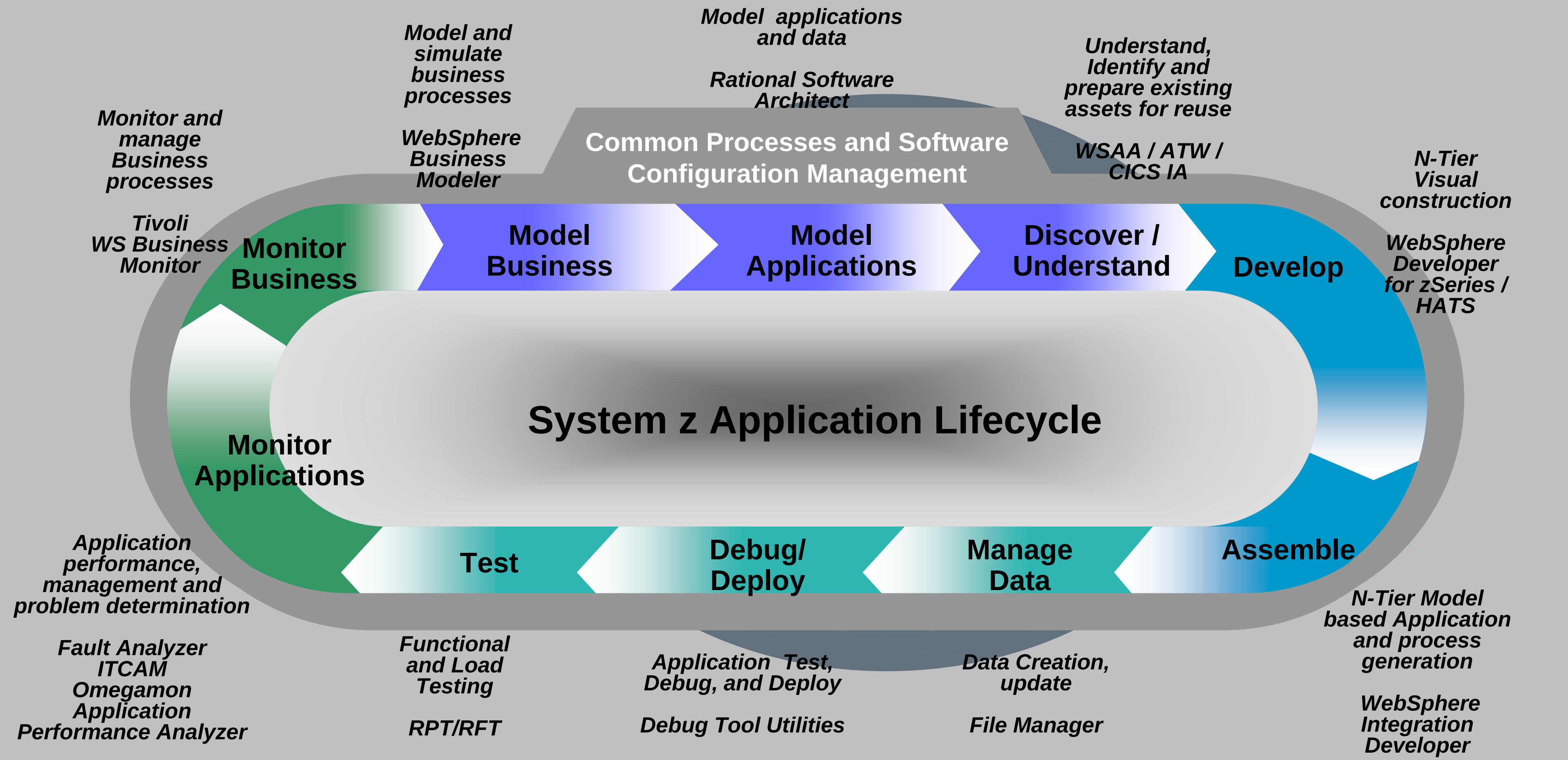
Network Computing

WebSphere tops Network Computing App Server Reviews

IBM WebSphere Application Server comes out on top

35+ years of maturity and innovation in transaction and data systems

Model and Discover



Monitor and manage Business processes

Tivoli WS Business Monitor

Model and simulate business processes

WebSphere Business Modeler

Model applications and data

Rational Software Architect

Understand, Identify and prepare existing assets for reuse

WSAA / ATW / CICS IA

N-Tier Visual construction

WebSphere Developer for zSeries / HATS

Application performance, management and problem determination

Fault Analyzer ITCAM Omegamon Application Performance Analyzer

Functional and Load Testing

RPT/RFT

Application Test, Debug, and Deploy

Debug Tool Utilities

Data Creation, update

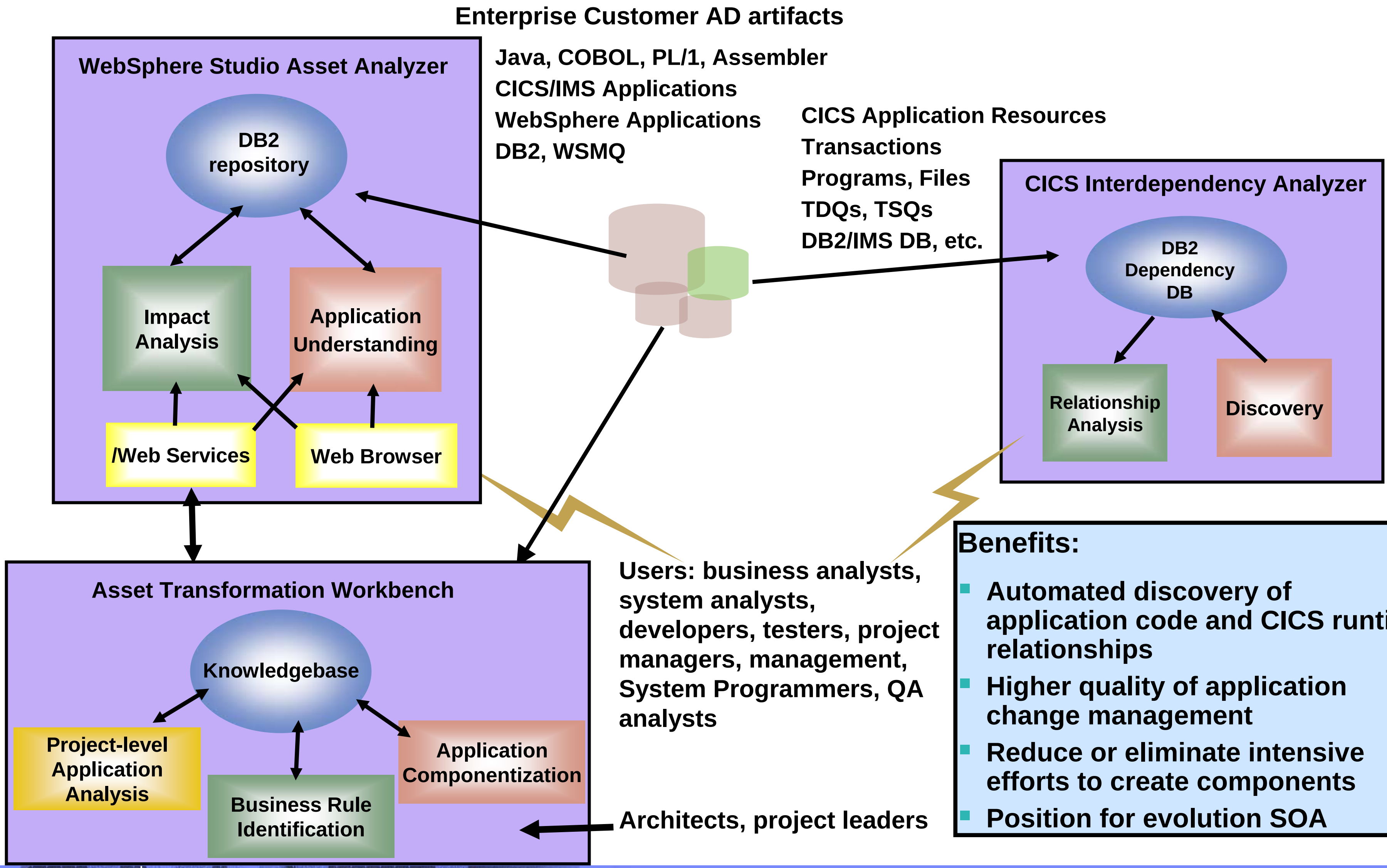
File Manager

N-Tier Model based Application and process generation

WebSphere Integration Developer

Enterprise Access to Assets

Speed application discovery, understanding and asset reuse



Enterprise Customer AD artifacts

Java, COBOL, PL/1, Assembler
CICS/IMS Applications
WebSphere Applications
DB2, WSMQ

CICS Application Resources
Transactions
Programs, Files
TDQs, TSQs
DB2/IMS DB, etc.

WebSphere Studio Asset Analyzer
DB2 repository
Impact Analysis
Application Understanding
/Web Services
Web Browser
CICS Interdependency Analyzer
DB2 Dependency DB
Relationship Analysis
Discovery
Asset Transformation Workbench
Knowledgebase
Project-level Application Analysis
Business Rule Identification
Application Componentization

Users: business analysts, system analysts, developers, testers, project managers, management, System Programmers, QA analysts

Architects, project leaders

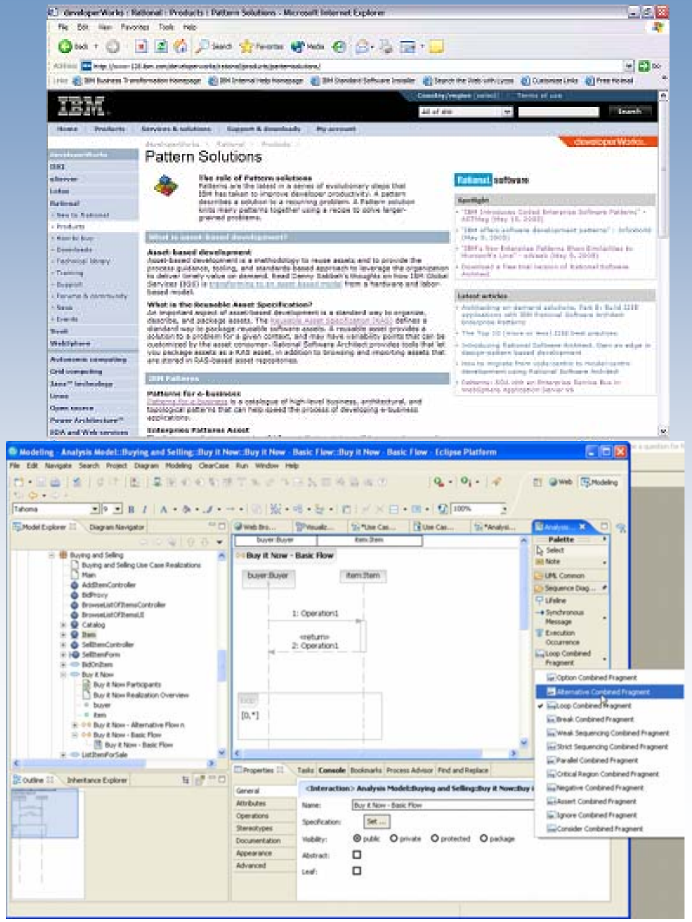
Benefits:

- Automated discovery of application code and CICS runtime relationships
- Higher quality of application change management
- Reduce or eliminate intensive efforts to create components
- Position for evolution SOA

Model - For The IT Architect and Developer

Using patterns to speed up the process

Model using industry standard UML 2, integrating the architecture into development


developerWorks : Rational : Products : Pattern Solutions - Microsoft Internet Explorer
Pattern Solutions
The role of Pattern solutions
What is asset-based development?
Asset-based development
What is the Reusable Asset Specification?
IBM Patterns
Patterns for e-business
Enterprise Patterns Asset
Rational software
Spotlight
Latest articles
Modeling - Analysis Model::Buying and Selling::Buy it Now::Buy it Now - Basic Flow::Buy it Now - Basic Flow - Eclipse Platform
File Edit Navigate Search Project Diagram Modeling ClearCase Run Window Help
Model Explorer
Diagram Navigator
Buy it Now - Basic Flow
buyer:Buyer
item:Item
1: Operation1
<return>
2: Operation1
Palette
Option Combined Fragment
Alternative Combined Fragment
Loop Combined Fragment
Break Combined Fragment
Weak Sequencing Combined Fragment
Strict Sequencing Combined Fragment
Parallel Combined Fragment
Critical Region Combined Fragment
Negative Combined Fragment
Assert Combined Fragment
Ignore Combined Fragment
Consider Combined Fragment
Properties
Tasks Console Bookmarks Process Advisor Find and Replace
General
Attributes
Operations
Stereotypes
Documentation
Appearance
Advanced
Name: Buy it Now - Basic Flow
Visibility: ● public ○ private ○ protected ○ package
Abstract: □
Leaf: □

Rational Software Architect V6.0.1

- Model in UML and transform to Web service
- Use patterns to help automate development of applications and promote reuse
- Use Process and best practices ensure repeatable success
- Integrates with business process modeling to ensure business needs drive development

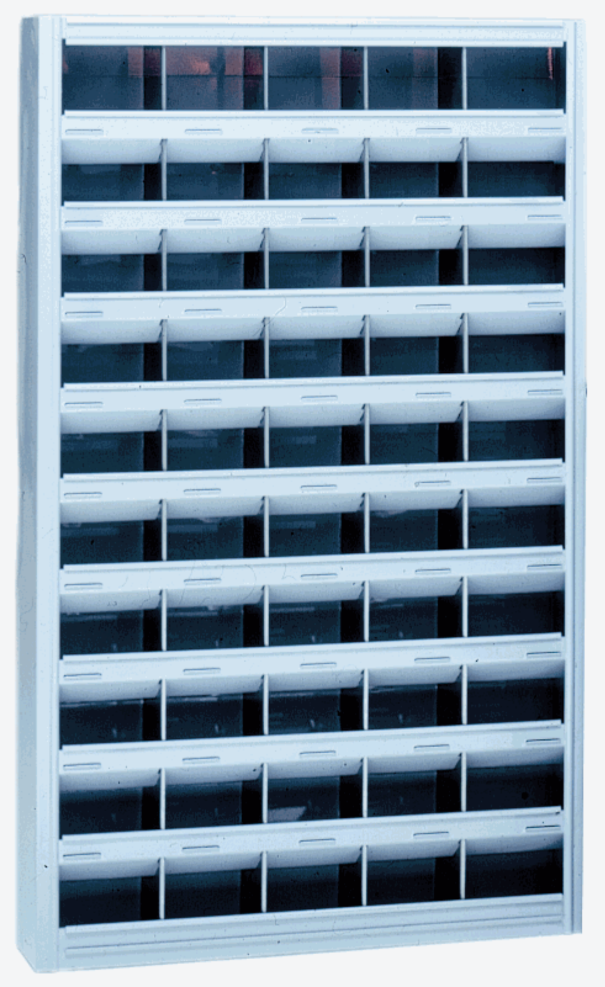
Rational Software Architect Pattern Solutions

- Improve productivity with reusable assets
- Rapidly build and configure the Enterprise Service Bus (ESB) with the WebSphere Platform Messaging Patterns

Identify Assets

WebSphere Service Registry and Repository

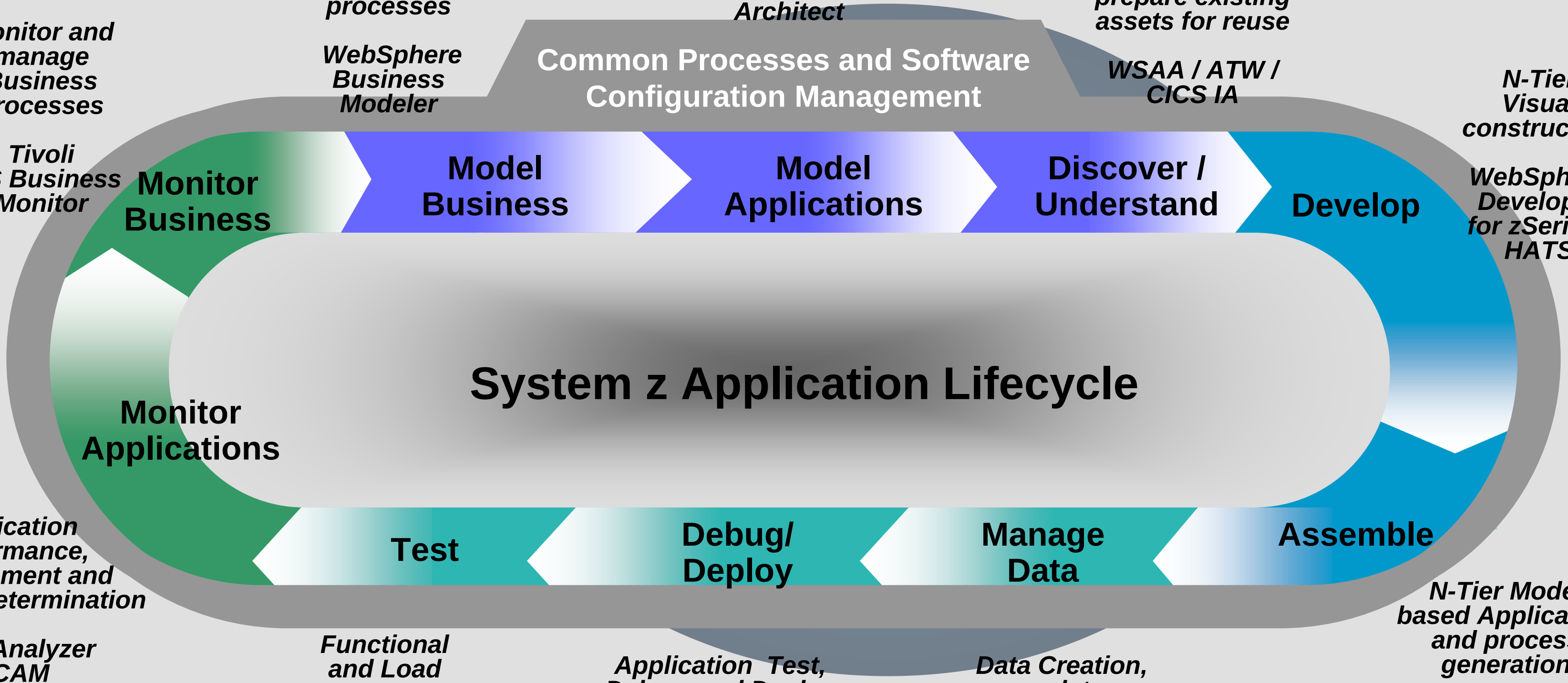
An enterprise-wide service registry and repository improves visibility, reusability, adaptability, and manageability of services



The WebSphere Service Registry and Repository …

- A repository for service metadata
  - for example, WSDL and XSD
- For publication of services
  - to advertise their capabilities
- For finding suitable services
  - for reuse and runtime agility
- For capturing service dependencies
  - to support change management
- An extensible framework
  - to support validation and notification

Develop and Assemble



Monitor and manage Business processes

Tivoli WS Business Monitor

Model and simulate business processes

WebSphere Business Modeler

Model applications and data

Rational Software Architect

Understand, Identify and prepare existing assets for reuse

WSAA / ATW / CICS IA

N-Tier Visual construction

WebSphere Developer for zSeries / HATS

Application performance, management and problem determination

Fault Analyzer ITCAM Omegamon Application Performance Analyzer

Functional and Load Testing

RPT/RFT

Application Test, Debug, and Deploy

Debug Tool Utilities

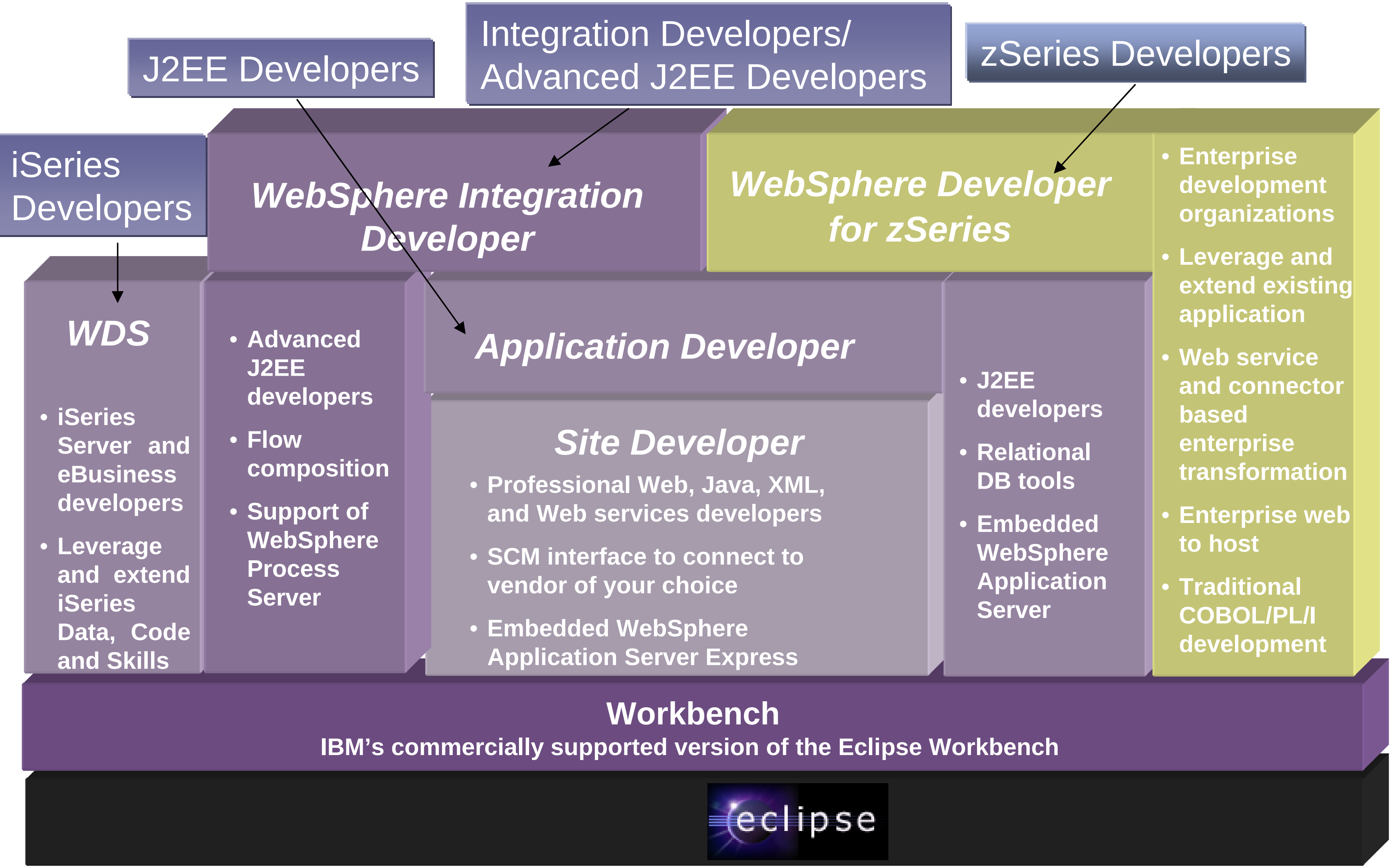
Data Creation, update

File Manager

N-Tier Model based Application and process generation

WebSphere Integration Developer

WebSphere/Rational Development Family



J2EE Developers

Integration Developers/ Advanced J2EE Developers

zSeries Developers

iSeries Developers

WebSphere Integration Developer

WebSphere Developer for zSeries

- Enterprise development organizations
- Leverage and extend existing application
- Web service and connector based enterprise transformation
- Enterprise web to host
- Traditional COBOL/PL/I development

WDS

- iSeries Server and eBusiness developers
- Leverage and extend iSeries Data, Code and Skills

- Advanced J2EE developers
- Flow composition
- Support of WebSphere Process Server

Application Developer

Site Developer

- Professional Web, Java, XML, and Web services developers
- SCM interface to connect to vendor of your choice
- Embedded WebSphere Application Server Express

- J2EE developers
- Relational DB tools
- Embedded WebSphere Application Server

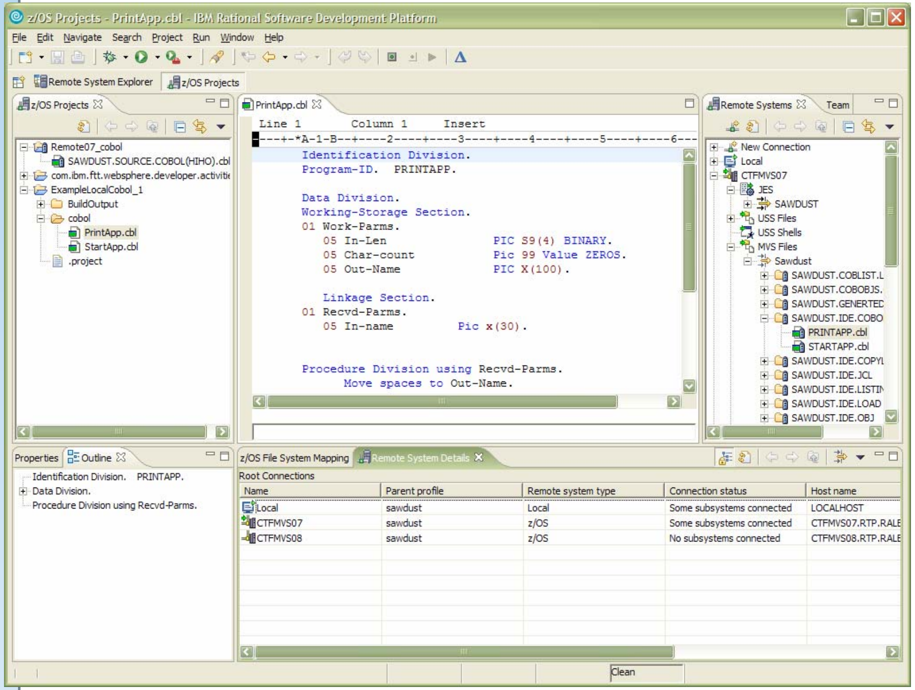
Workbench

IBM's commercially supported version of the Eclipse Workbench

WebSphere Developer for zSeries

Eclipse-based integrated development environment for developing enterprise-level, multi-tier applications (composite applications)

- **Builds core stack zOS applications**
  - COBOL, PLI, HLASM
  - TSO/Batch, CICS, IMS, DB2
  - DB2 Stored Procedures – COBOL, PLI, Java, SQL
- **Creates COBOL/CICS/JSF/Java/J2EE Multi-tier apps**
  - Built on Rational Application Developer
    - Includes all of the J2EE web development tools
  - Generate JSF/EGL/J2EE web front ends
  - COBOL backends running on zSeries
- **Enables CICS and IMS applications for Web services and SOA**
  - Provides tooling to make it easy to integrate existing applications into an SOA
- **Supports the full application lifecycle**
  - Model, Architect, Develop, Test, Deploy, and Manage



```
Line 1          Column 1       Insert
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6---
       Identification Division.
       Program-ID.  PRINTAPP.

       Data Division.
       Working-Storage Section.
       01 Work-Parms.
          05 In-Len                  PIC S9(4) BINARY.
          05 Char-count              Pic 99 Value ZEROS.
          05 Out-Name                PIC X(100).

          Linkage Section.
       01 Recvd-Parms.
          05 In-name            Pic x(30).


       Procedure Division using Recvd-Parms.
             Move spaces to Out-Name.
```

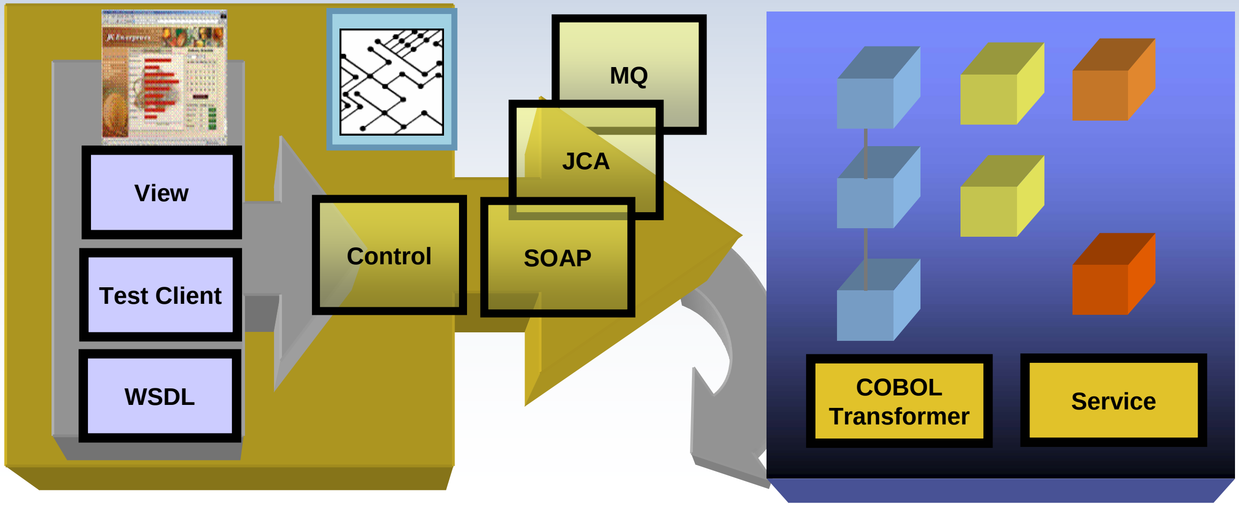
Root Connections

Name	Parent profile	Remote system type	Connection status	Host name
Local	sawdust	Local	Some subsystems connected	LOCALHOST
CTFMVS07	sawdust	z/OS	Some subsystems connected	CTFMVS07.RTP.RALE
CTFMVS08	sawdust	z/OS	No subsystems connected	CTFMVS08.RTP.RALE

z/OS Composite Development tools

Transition of Traditional environments to Web and Composite applications

- SOA / SOAP / XML / Enablement
- JCA Support
- Service Flow Modeler
- HATS
- Enterprise Generation Language (EGL) / JSF
  - COBOL/CICS generation
  - Java generation

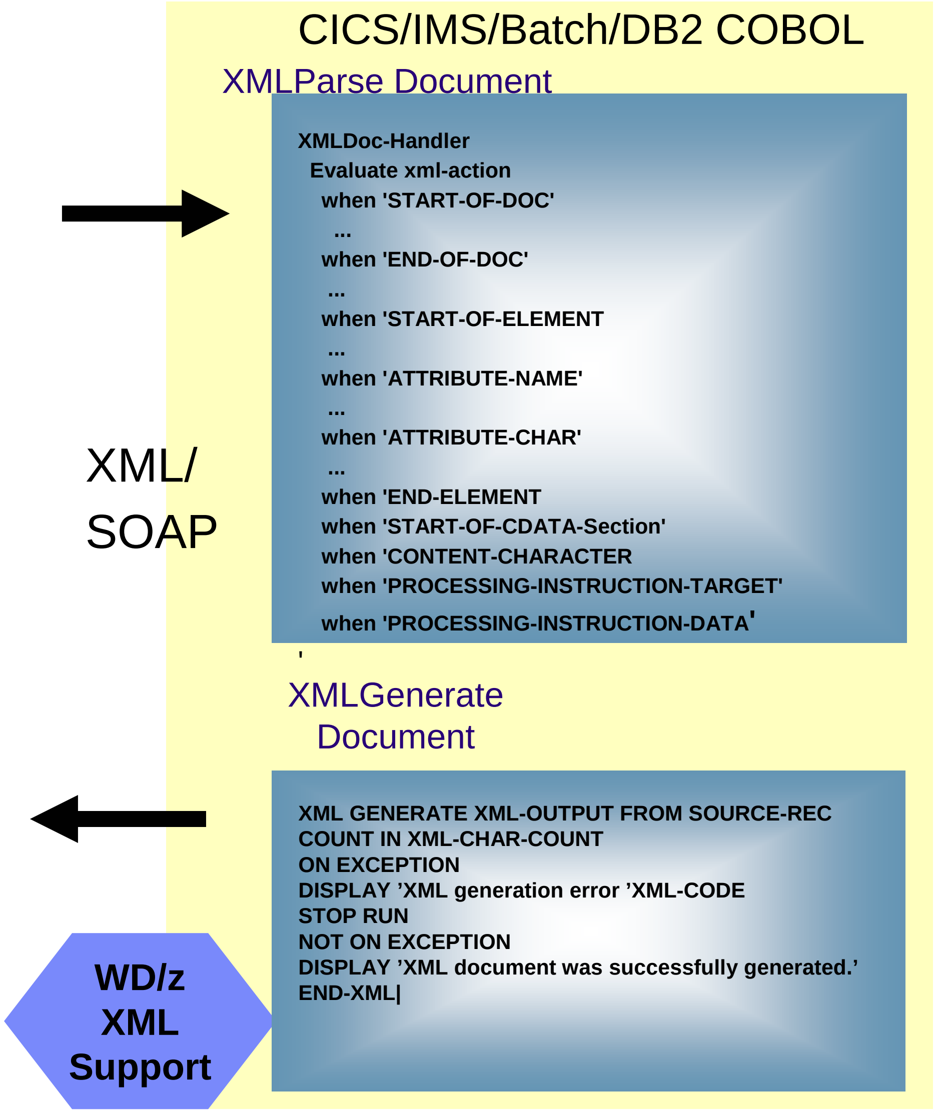


Using Enterprise COBOL to service-enable z/OS

- What's the latest…
  - XML Language based generation from COBOL data structure
  - WebSphere EJB support
  - DB2 V8 preprocessor
  - CICS preprocessor
- High speed XML Sax based parsing
- Object Oriented Support for Java COBOL Interoperability
- Unicode support
- Similar XML parsing support available in Enterprise PL/I
- CICS and DB2 integrated preprocessor
- Raise 16Mb COBOL data size limit
  - Picture clause replication:

    01 A PIC X(134217727).
  - OCCURS::

    05 V PIC X OCCURS 134217727 TIMES.



CICS/IMS/Batch/DB2 COBOL

XMLParse Document

```
XMLDoc-Handler
 Evaluate xml-action
  when 'START-OF-DOC'
    ...
  when 'END-OF-DOC'
   ...
  when 'START-OF-ELEMENT
   ...
  when 'ATTRIBUTE-NAME'
   ...
  when 'ATTRIBUTE-CHAR'
   ...
  when 'END-ELEMENT
  when 'START-OF-CDATA-Section'
  when 'CONTENT-CHARACTER
  when 'PROCESSING-INSTRUCTION-TARGET'
  when 'PROCESSING-INSTRUCTION-DATA'
'
```

XML/ SOAP

XMLGenerate Document

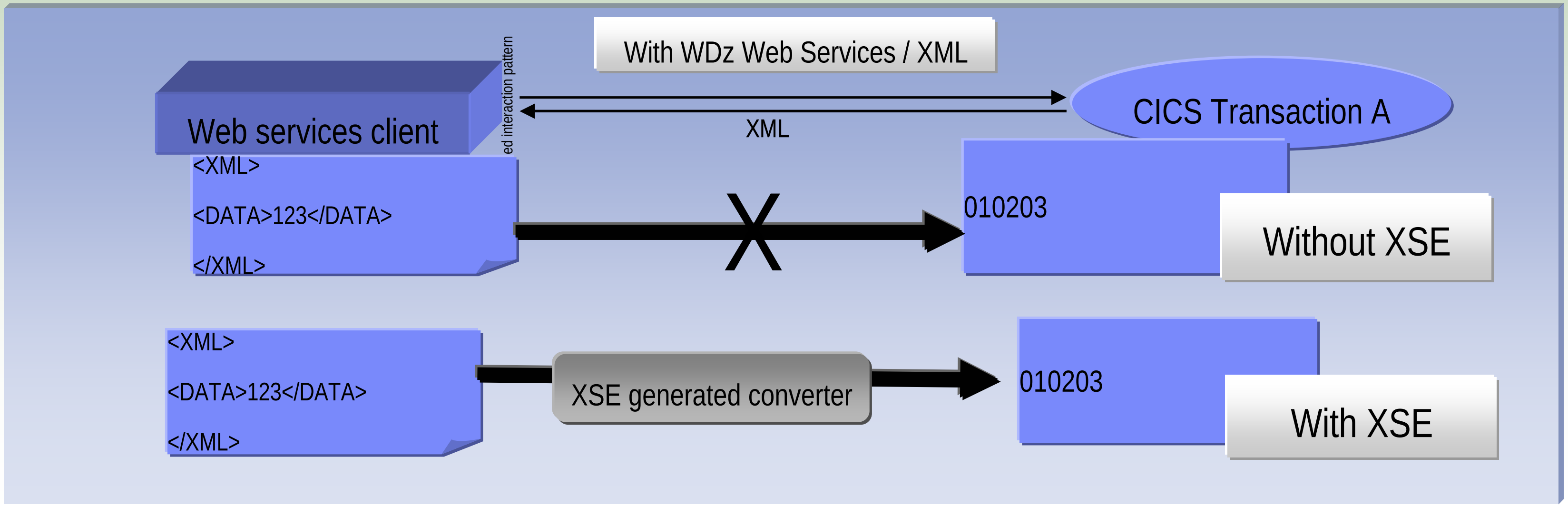
```
XML GENERATE XML-OUTPUT FROM SOURCE-REC
COUNT IN XML-CHAR-COUNT
ON EXCEPTION
DISPLAY ’XML generation error ’XML-CODE
STOP RUN
NOT ON EXCEPTION
DISPLAY ’XML document was successfully generated.’
END-XML|
```

WD/z XML Support

WDz SOA Tools – Part 1

XML Services for the Enterprise (XSE) in WDz

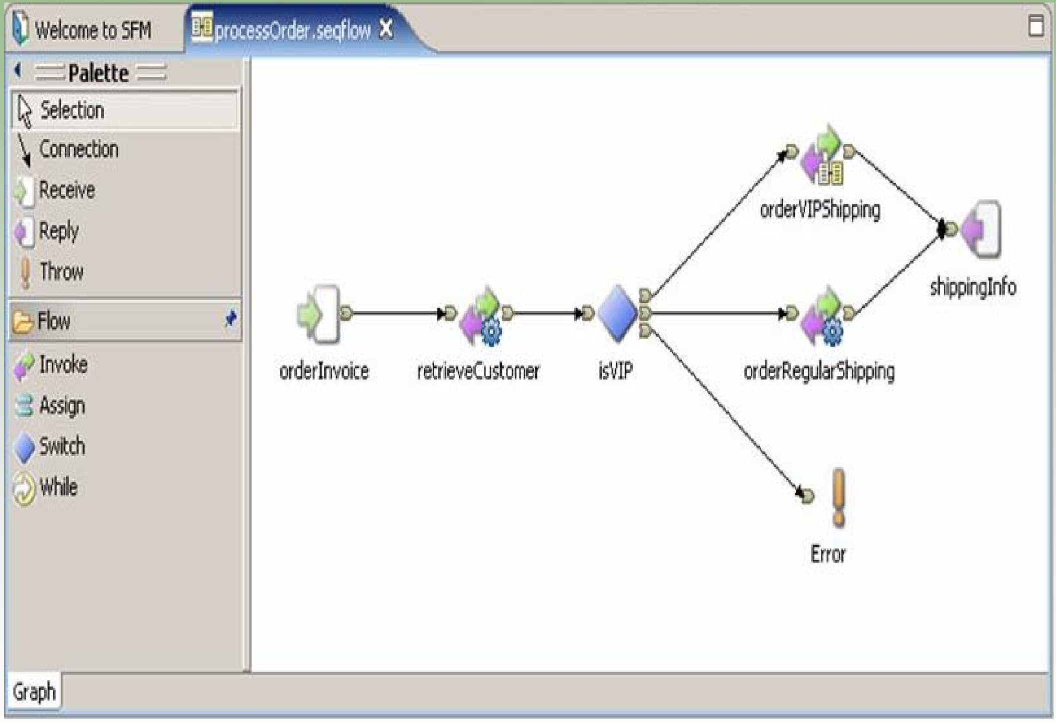
- Most rapid building of Web services from existing CICS applications
  - Single CICS and IMS transactions enabled for Web Services
  - Supports IMS Message Queue, CICS Commarea and new Channels/Container based applications
  - Rapid generation of WSDL, CICS WSBind, and Adapter generation eliminating complex hand coding of XML to/from language conversions
  - Includes complete Web Services Test and Java generation environment



WDz SOA Tools – Part 2

Service Flow Modeler in WebSphere Developer for zSeries

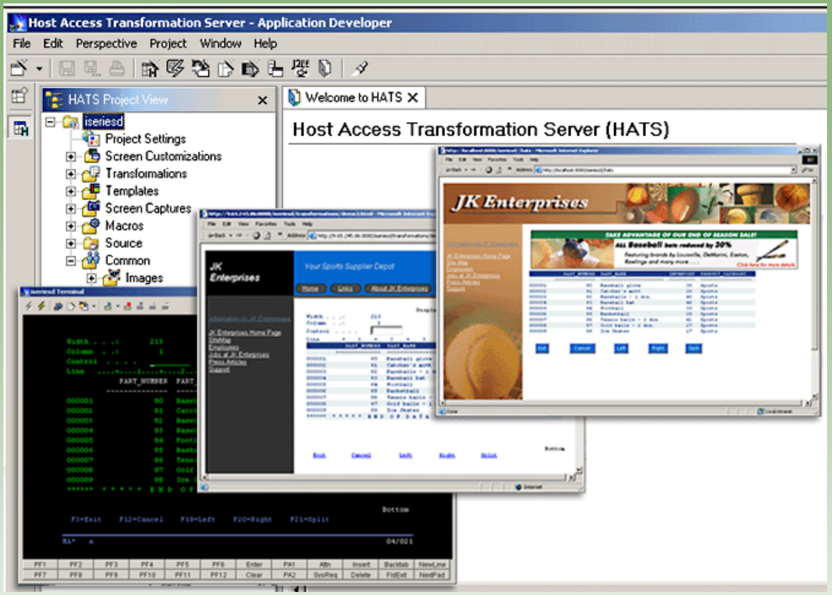
- Builds Web services from existing CICS applications
  - Aggregates multiple CICS transactions into high-level business processes through visual modeling
  - Supports CICS BMS (terminal-based) applications & CICS commarea applications
  - Highly optimized CICS runtime supporting Web services and XML interfaces



WebSphere Host Access Transformation Server

Extend business processing through existing interfaces

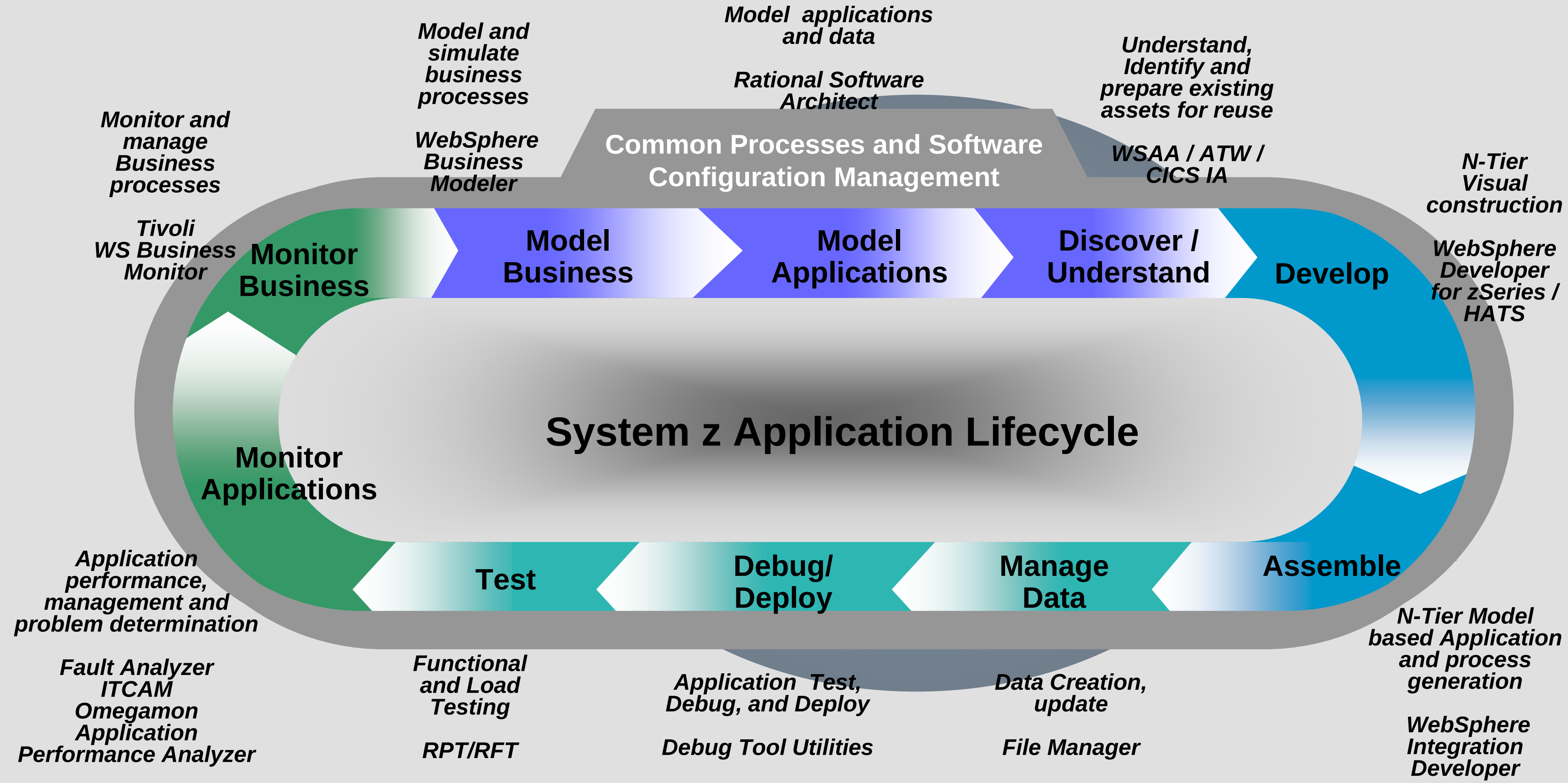
- Automatically transforms 3270 & 5250 green screen applications into HTML interfaces
- Extends terminal applications as Web Services
- Low skills requirement – no zSeries skills required
- Rules-based, highly customizable
- Iterative, eclipse-based development environment



Benefits:

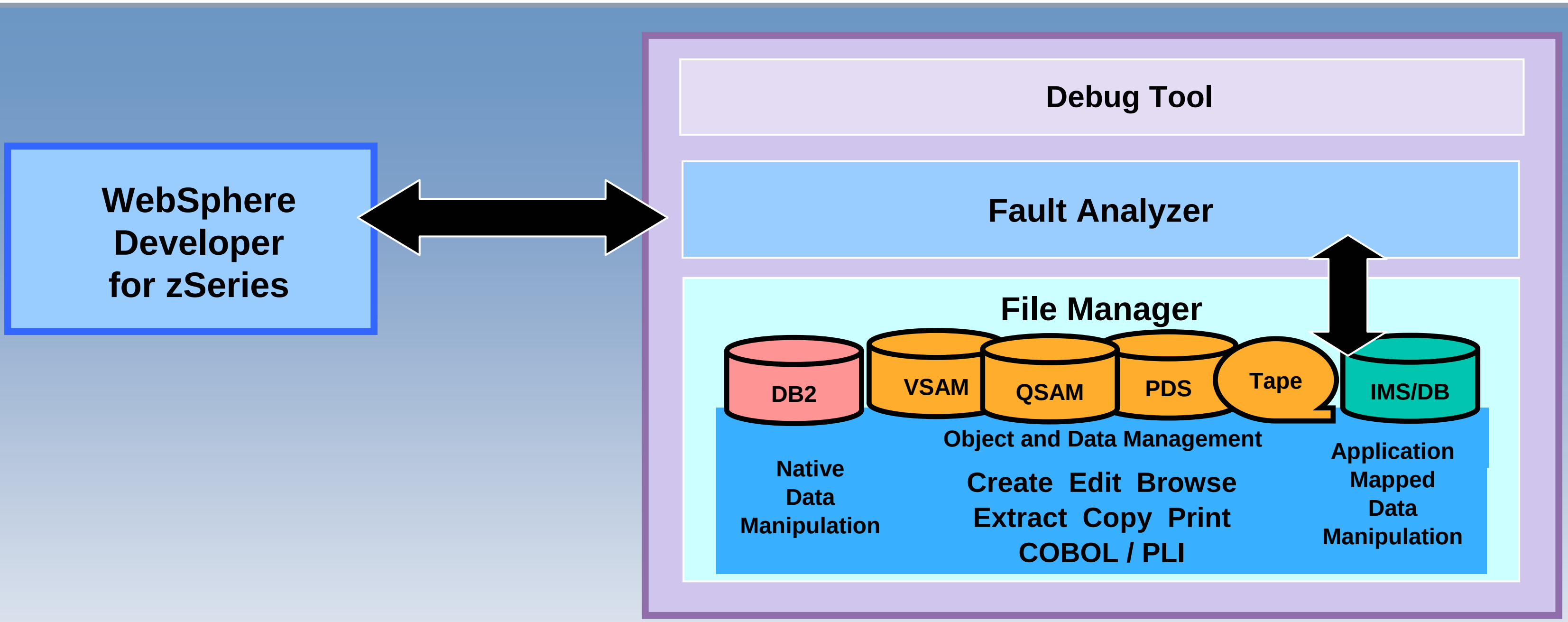
- Increase productivity and reduce training costs.
- Extend existing applications to new users
- Integrate traditional applications into enterprise portals
- Reduce development costs by avoiding rewrite of legacy applications.

Deploy and Manage


Model applications and data
Rational Software Architect
Model and simulate business processes
WebSphere Business Modeler
Understand, Identify and prepare existing assets for reuse
WSAA / ATW / CICS IA
Monitor and manage Business processes
Tivoli
WS Business Monitor
N-Tier Visual construction
WebSphere Developer for zSeries / HATS
Common Processes and Software Configuration Management
Monitor Business
Model Business
Model Applications
Discover / Understand
Develop
System z Application Lifecycle
Monitor Applications
Test
Debug/ Deploy
Manage Data
Assemble
Application performance, management and problem determination
Fault Analyzer
ITCAM
Omegamon
Application Performance Analyzer
Functional and Load Testing
RPT/RFT
Application Test, Debug, and Deploy
Debug Tool Utilities
Data Creation, update
File Manager
N-Tier Model based Application and process generation
WebSphere Integration Developer

Test and Problem Determination

Integration speeds time to market



Benefits:

- Simplify development of zSeries test cases
  - Data creation for DB2, IMS/DB, VSAM, and QSAM
  - Extract and load
- Reduced deployment complexity
  - Production data validation and creation
- Common environment
  - Reuse of skills across e-bus and traditional applications

End To End Monitoring

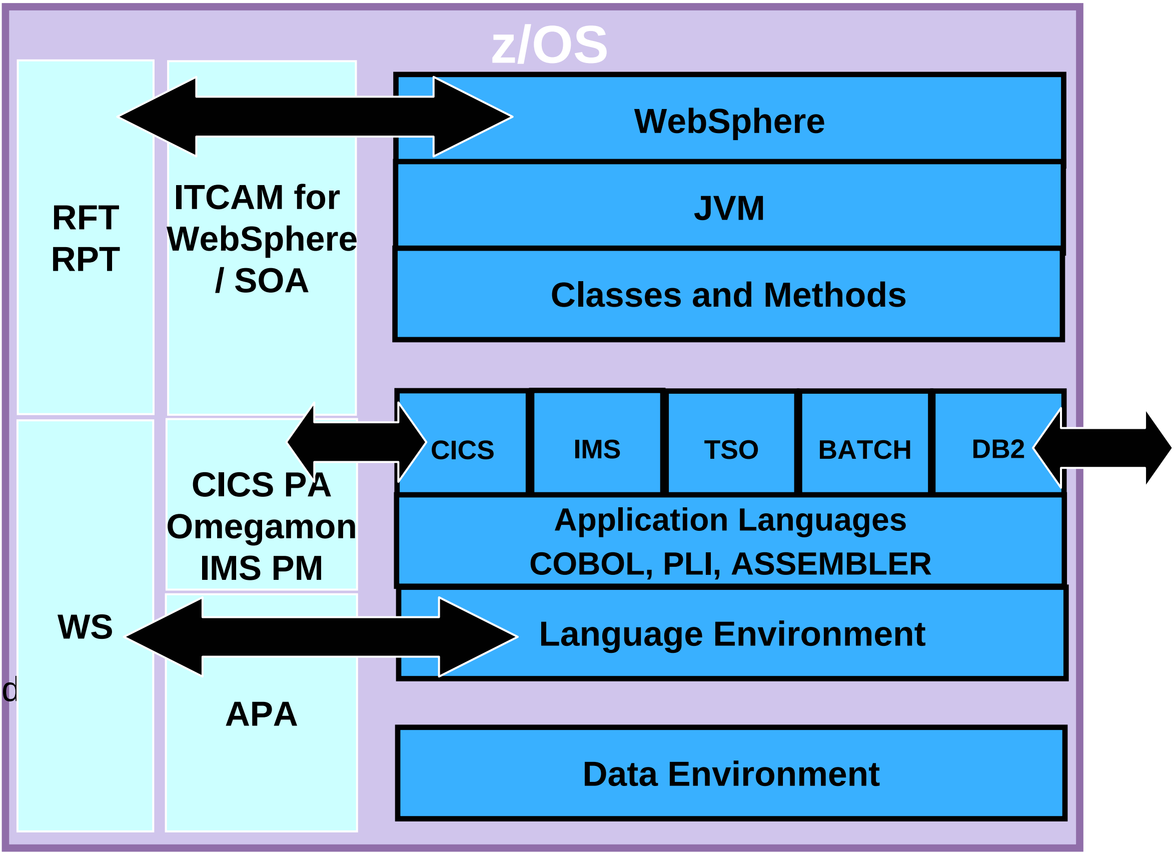
Enables highest QOS and maintainability of composite applications

Benefits:

- RPT, ITCAM used to drive and monitor J2EE performance on both WAS and traditional servers enabling rapid problem determination and reduced downtime

- CICS PA /OMEGAMON provide CICS and IMS resource monitoring enabling rapid response to problems

- System z WS and PA are used to drive and monitor CICS transactions and DB2 performance for COBOL / PLI applications enabling high throughput in System z environments



DB2 PM provides DB2 resource monitoring

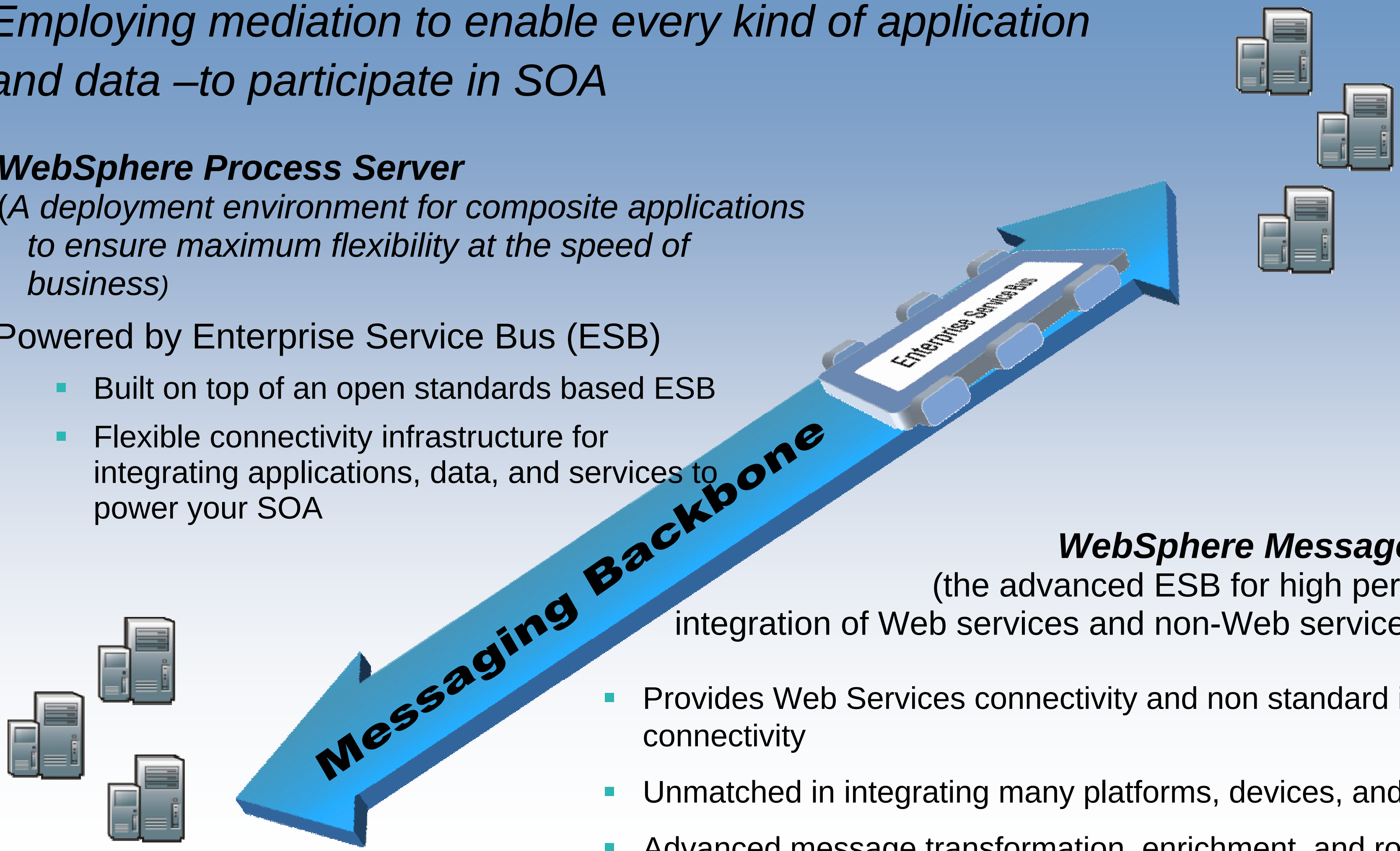
Deploying processes on a flexible, robust SOA integration platform

Employing mediation to enable every kind of application and data –to participate in SOA

WebSphere Process Server
(A deployment environment for composite applications to ensure maximum flexibility at the speed of business)

Powered by Enterprise Service Bus (ESB)

- Built on top of an open standards based ESB
- Flexible connectivity infrastructure for integrating applications, data, and services to power your SOA



WebSphere Message Broker
(the advanced ESB for high performance integration of Web services and non-Web services assets)

- Provides Web Services connectivity and non standard interface connectivity
- Unmatched in integrating many platforms, devices, and APIs
- Advanced message transformation, enrichment, and routing

Gartner: Best Practices for Mainframe SOA

- Act tactical, think strategic
- Evaluate tools that provide good microflow orchestration
- Create services that utilize function from across existing application boundaries.
- Build a reuse culture and technology infrastructure.
- Work with operations to create management/performance-monitoring support.
- Use code understanding/inventory/restructuring tools to improve service granularity.
- Define the role of the mainframe in future application architecture.