Software Group

Modern Application Architecture

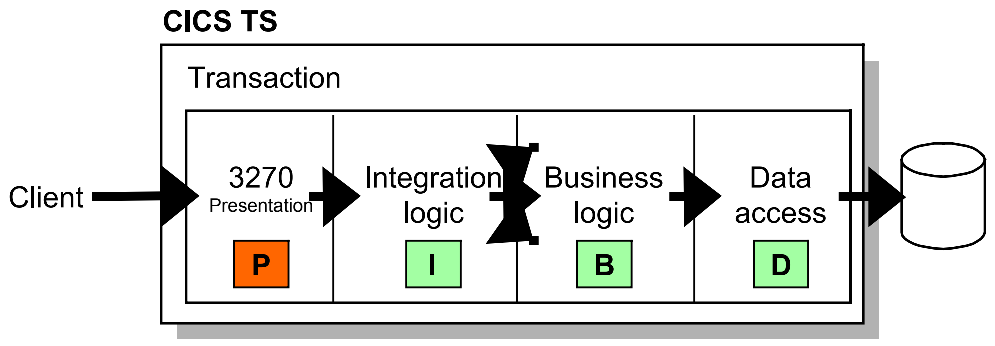
SOA : Part 2

Paolo Chieregatti
Certified IT Specialist
paolo.chieregatti@it.ibm.com

Agenda

- **Introduction**
- **Middle Tier**
  - WebSphere Application Server
    - J2EE
    - Servlets, JSP's and JSF
    - EGL
- **Client**
  - HTML
- **Connectivity**
  - Web Services, XML, SOAP, WSDL
- **Business Tier**
  - CICS
  - COBOL

Modern “CICS” architecture

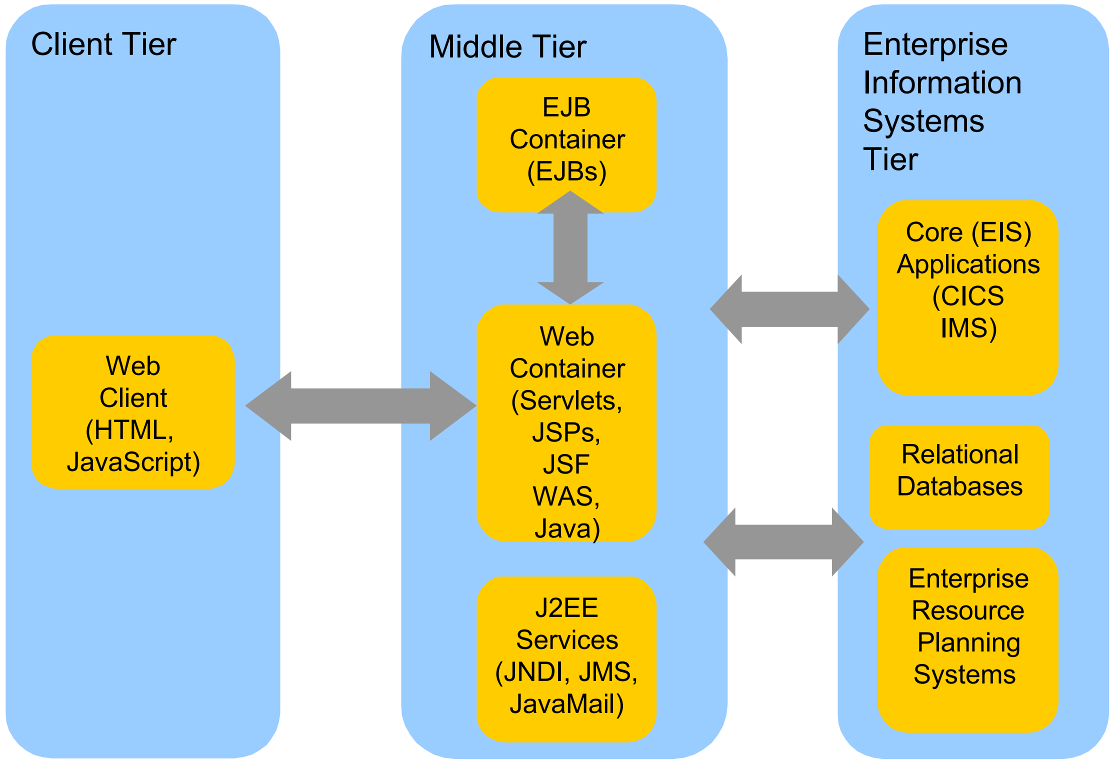


- **Best practice in CICS application design is to separate key elements of the application, in particular:**

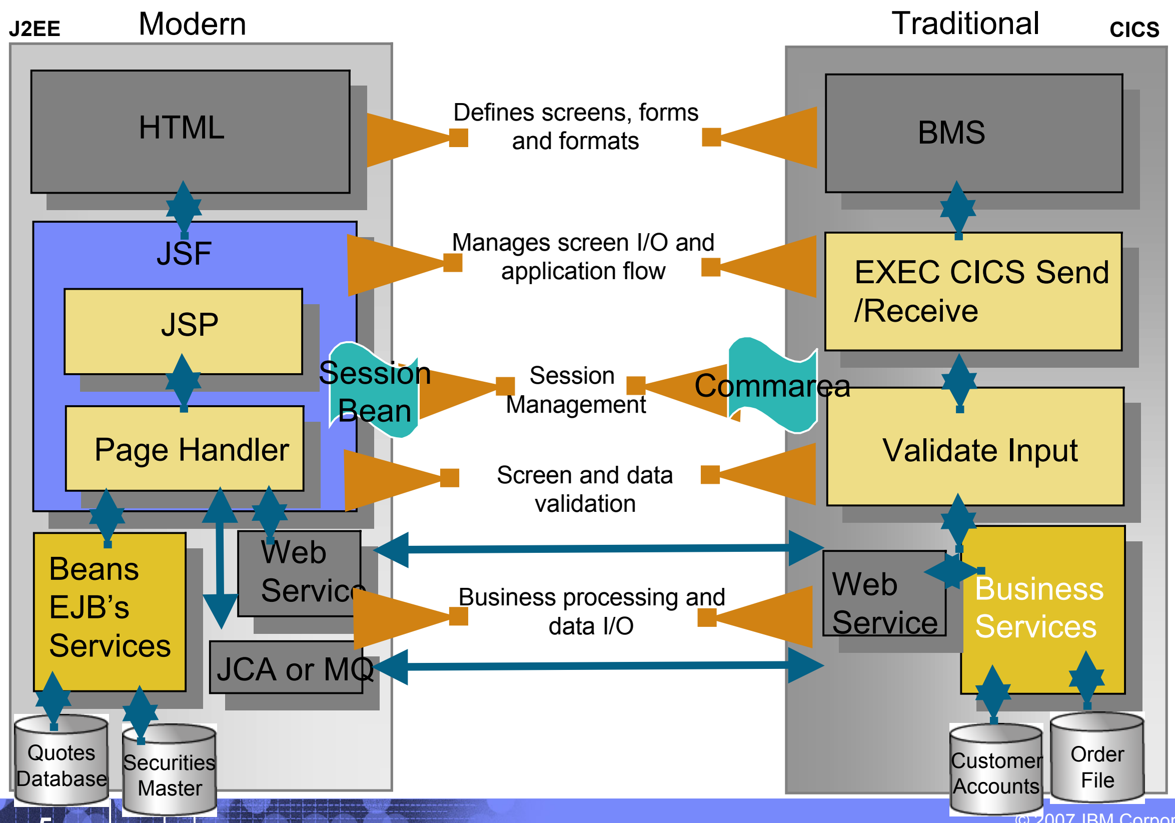
– Presentation logic	3270, HTML, XML
– Integration or aggregation logic	Menu, router, tooling
– Business logic	COBOL, PL/I, Reusable component
– Data access logic	VSAM, DB2, IMS, …

- **Provides a framework for reuse and facilitates separation of concerns, clear interfaces, ownership, and optimisation**

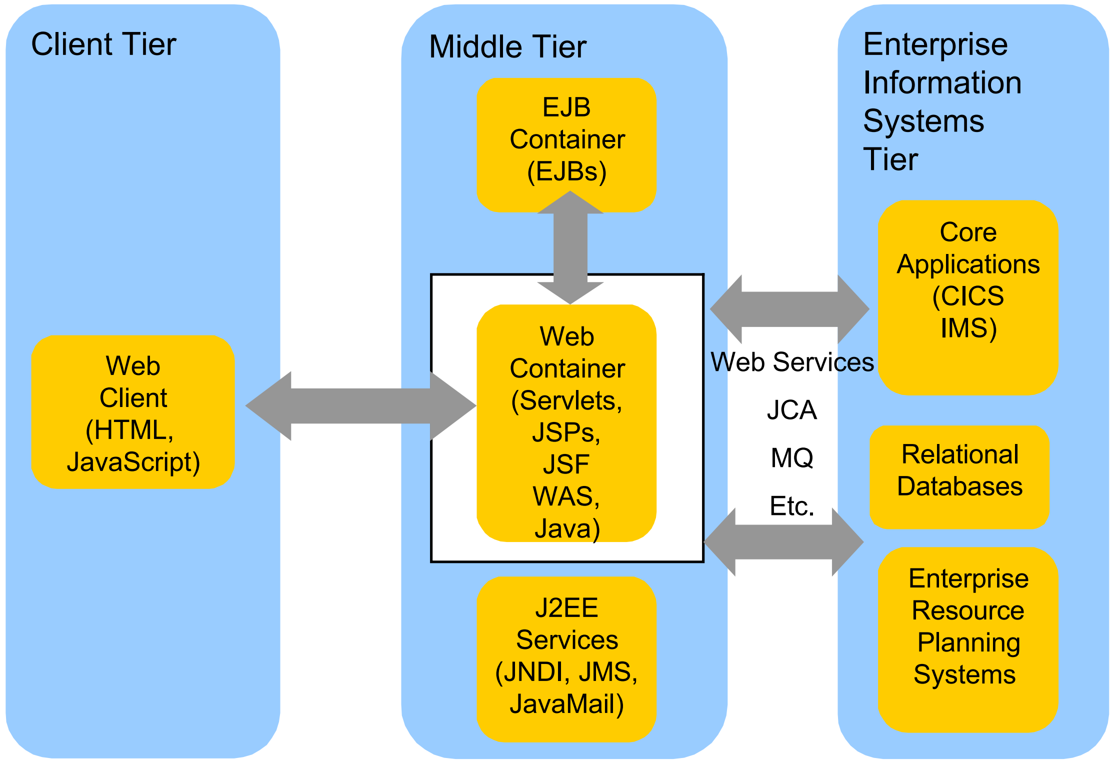
“Modern” Multitier Architecture


Client Tier
Web Client (HTML, JavaScript)
Middle Tier
EJB Container (EJBs)
Web Container (Servlets, JSPs, JSF WAS, Java)
J2EE Services (JNDI, JMS, JavaMail)
Enterprise Information Systems Tier
Core (EIS) Applications (CICS IMS)
Relational Databases
Enterprise Resource Planning Systems

It’s not that different

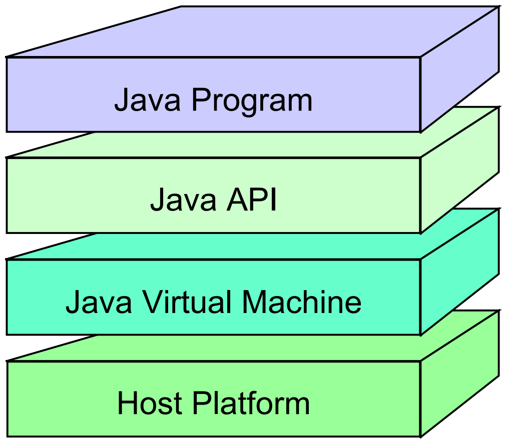


The Middle Tier


Client Tier
Web Client (HTML, JavaScript)
Middle Tier
EJB Container (EJBs)
Web Container (Servlets, JSPs, JSF WAS, Java)
J2EE Services (JNDI, JMS, JavaMail)
Web Services
JCA
MQ
Etc.
Enterprise Information Systems Tier
Core Applications (CICS IMS)
Relational Databases
Enterprise Resource Planning Systems

The Java platform

- Java is an object-oriented programming language developed by Sun Microsystems
- Java has a set of standardized class libraries that support predefined reusable functionality
- Java has a runtime environment that can be embedded in Web browsers and operating systems
- Many popular UI / Session frameworks are built on Java processing



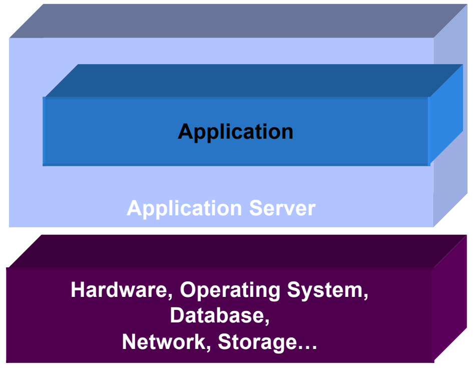
Procedural and object oriented approaches – example

- **System requirement**
  - Banking system model withdrawing money from a savings account
- **Procedural approach**
  - Identify where the data is stored
  - List the algorithmic steps necessary to perform the action
- **Object approach**
  - Identify what objects are involved; these objects will directly relate to real life objects (Bank, SavingsAccount, Teller and Transaction)
  - Show how these objects interact:
    - To enforce business rules for withdrawals
    - To modify the balance

Both have advantages in SOA – in the right place

What is an Application Server?

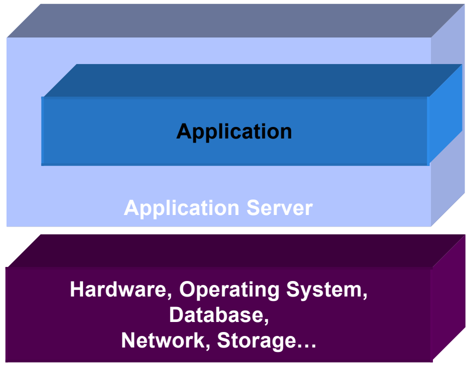
- **Provides the infrastructure for running applications that run your business**
  - **Insulates** applications from hardware, operating system, network…
  - Provides a common environment and programming model for applications
    - **Write once, run anywhere** (J2EE)
    - Platform for developing and deploying **Web Services**
  - Provides a **scalable, reliable** transaction engine for your enterprise



What is WebSphere Application Server?

- **WebSphere Application Server is a platform on which you can run Java-based business applications**
- **It is an implementation of the Java 2 Enterprise Edition (J2EE) specification**
- **It provides services (database connectivity, threading, workload management, and so forth) that can be used by the business applications**

What is J2EE?

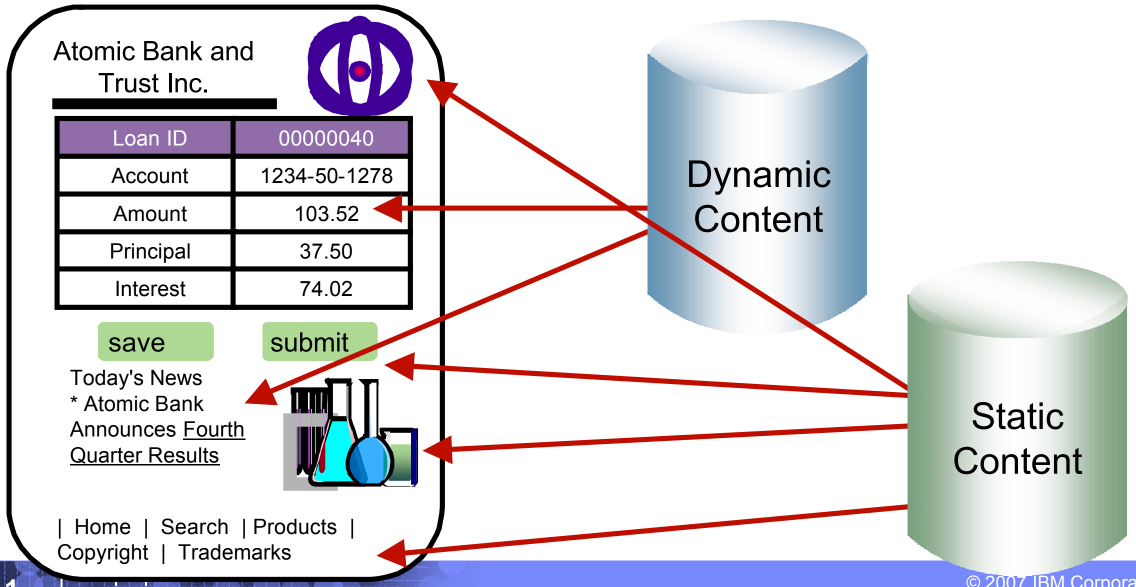


- **J2EE – Java 2 Enterprise Edition**
  - A run-time platform used for developing, deploying, and managing multitier server-centric applications on an enterprise-wide scale

- **J2EE defines four types of components which must be supported by any J2EE product**
  - Applets
    - Graphical Java components which typically execute within a browser
    - Can provide a powerful user interface for other J2EE components
  - Application client components
    - Java programs which execute on a client machine and access other J2EE components
  - Web components
    - Servlets and JavaServer Pages
    - These provide the controller and view functionality in J2EE
  - Enterprise JavaBeans
    - Distributed, transactional components for business logic and database access

Web page content

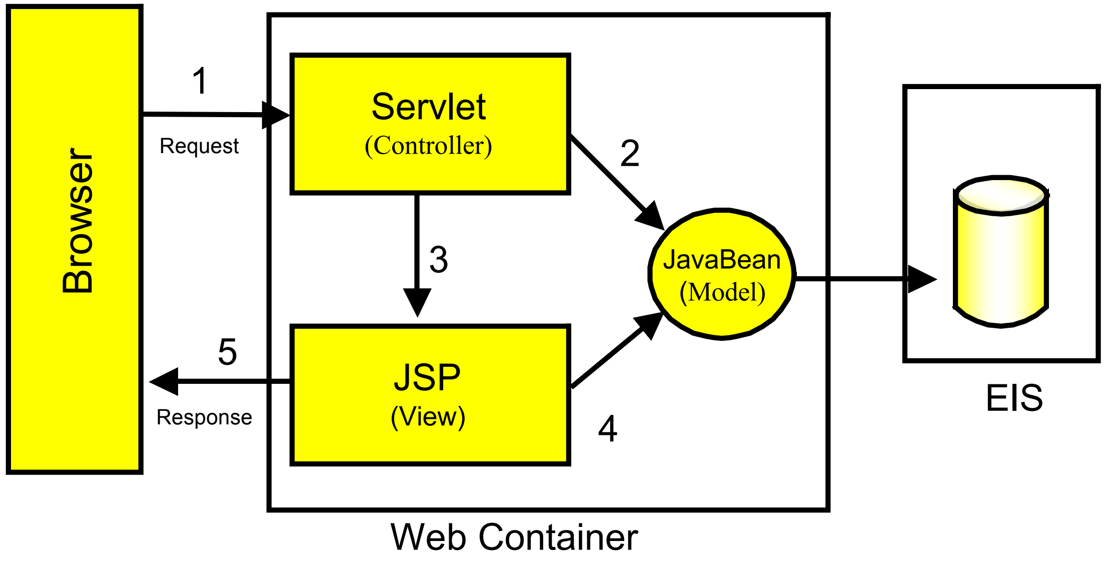
- **Content delivered to a client is composed from:**
  - Static or non-customized content
  - Customized content
- **Page layout and style are managed through HTML, XSL**



Typical J2EE Web Application Model

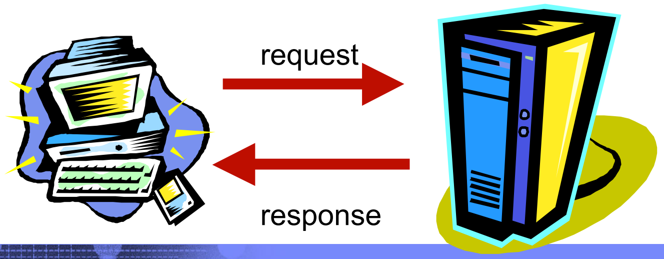
- **A request is sent to a servlet that generates dynamic content and calls a JSP page to send the content to the browser, as shown:**

MVC Design Pattern



What Is a Servlet?

- **A servlet is a standard, server-side component of a J2EE application which executes business logic on behalf of an HTTP request**
  - Runs in the server tier (and not in the client)
  - A pure Java alternative to other technologies, such as CGI scripts
  - Managed by the Web container
- **Servlets form the foundation for Web-based applications in J2EE**



A Simple Java Servlet Example

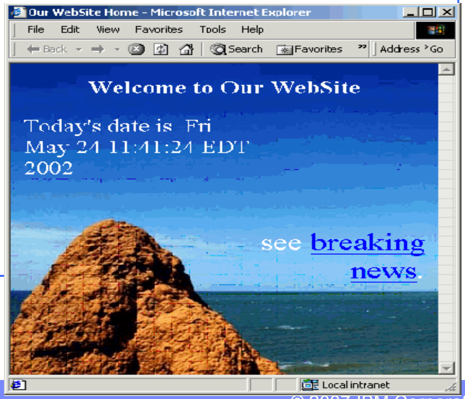
```
package com.ibm.example.servlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletException;
import java.io.IOException;
import java.io.PrintWriter;
public class VerySimpleServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
                         throws ServletException, IOException {
        String browser = request.getHeader("User-Agent");
        response.setStatus(HttpServletResponse.SC_OK);  // default
        response.setContentType("text/html");          // default
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Simple servlet");
        out.println("</TITLE></HEAD><BODY>");
        out.println ("Browser details: " + browser);
        out.println("</BODY></HTML>");
}
}
```

What is JSP (JavaServer Pages)?

- JavaServer Pages is a technology that lets you mix static HTML with dynamically generated HTML
- JSP technology allows server-side scripting
- A JSP file (has an extension of .jsp) contains any combination of:
  - JSP syntax
  - Markup tags such as HTML or XML

A simple JSP example

```
<HTML>
<HEAD><TITLE>Our WebSite Home</TITLE></HEAD>
<BODY background="image.jpg" text="#ffffff">
<TABLE>
<TR><TD>
<H1>Welcome to Our WebSite</H1>
</TD></TR><TR><TD>
<H3>Today's date is

<%= new java.util.Date() %>

</H3></TD>
<TD>see <A href="breaking.html">
breaking news</A>.
</TD></TR>
</TABLE>
</BODY>
</HTML>
```


Our WebSite Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Address Go
Local intranet

Welcome to Our WebSite

Today's date is Fri May 24 11:41:24 EDT 2002

see breaking news.

JSP or Servlet?

- **Writing HTML code in a servlet is tedious and difficult to maintain**
- **Java code embedded in a JSP is difficult to reuse and maintain**
- **Use servlets to:**
  - Determine what processing is needed to satisfy the request
  - Validate input
  - Work with business objects to access the data and perform the processing needed to satisfy the request
  - Control the flow through a Web application
- **Use JSP pages to format and displaying the content generated by your servlets**

What is JavaServer Faces?

- **JavaServer Faces (JSF) is a *framework* for developing Web-based applications.**
  - A framework is a skeleton or foundation of an application
    - Provides code, resources, concepts and best practices upon which applications are constructed
- **The main components of JavaServer Faces are:**
  - An API and reference implementation for:
    - representing UI components and managing their state
    - handling events, server side validation, and data conversion
    - defining page navigation
    - supporting internationalization and accessibility
    - providing extensibility for all of these features
  - A JavaServer Pages (JSP) custom tag library for expressing UI components within a JSP page
  - **EGL**, IBM's enterprise generation or *business* language supports JSF

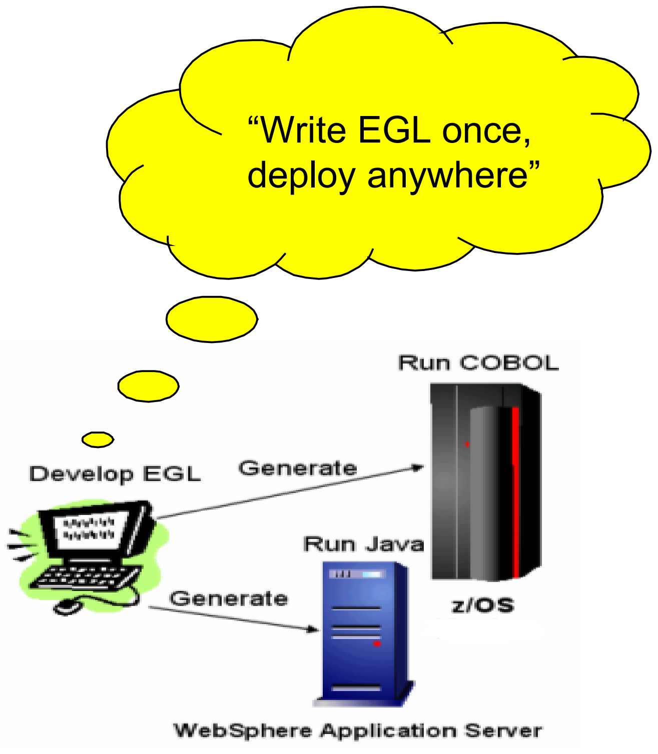
What is EGL?

Enterprise Generation Language (EGL)

- Is a development environment and programming language that lets users write full-function applications quickly
- Can be used to create text-based user interfaces for migration of existing applications
- Focus is on the business problem rather than on software technologies
- Is written independently of the target platform
- Can be generated into Java or COBOL programs
- Is well-suited to procedural programmers
- Is a high-level language which promotes iterative development and testing early in the development cycle

EGL - generation

- **Runtime code generated for appropriate platform**
  - **Java** for Windows, Linux, and so forth
  - **COBOL for z/OS**
  - Uses SQL transparently
- **EGL can be used to create "full" Web-based applications including Web UI's**
  - JavaServer Faces application is generated for runtime code
  - Runs on WebSphere Application Server


"Write EGL once, deploy anywhere"
Run COBOL
Develop EGL
Generate
Run Java
Generate
z/OS
WebSphere Application Server

EGL – key high level language abstractions

- **Data Access:**
  - Common Verbs for data access (Get, Add, Replace, Delete)
  - Abstracts access to SQL, Indexed, Relative, Serial, DL/I, MQ, Services
  - Allows complete access to SQL statement if needed
  - Common Error Handling

```
function allLoans()
      loans LoanRec[];
      get loans;
end

function loansInFlorida()
      loans LoanRec[];
      get loans with #sql{
            select *
            from LOANREC
            where state = "FL"};
end
```

- **Validation/Editing Rules**
  - Define formatting & validation rules once in common place
  - Reuse data items for Records, screens, reports

```
DataItem Password char(10) {
      validatorFunction = "passwordValidation",
      displayUse = secret,
      displayName = "Enter your Password",
      inputRequired = yes}
end
```

- **Remote Invocation**
  - Call COBOL, RPG, C, Java
  - Linkage information separated from code…simplifies development

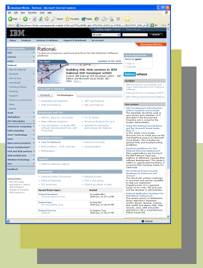
```
function callHelloWorldOniSeries()
      salutation char(30);
      call helloworld salutation;
end
```

- **Transaction Control:**
  - JDBC, CICS, IMS

```
function loansInFlorida()
      startTransaction(myLoanTransaction);
      ...
      commit();
      ...
      rollback();
end
```

EGL – simple programming model

- **Page Handlers**
  - Contain functions and data related to a .jsp
  - Should be mostly "Controller Logic"



```
PageHandler customerInfoPage {
        view = "customerInfoPage.jsp",
        title = "Customer Information",
        onPageLoadFunction = "onPageLoad"}

function onPageLoad()
        loans LoanRec[];
        get loans;
end
...
end
```

- **Report Handler**
  - Call-out to EGL "ReportHandler"
  - Open Source Reporting Framework

```
ReportHandler customerList

function afterPageInit()
...
end

end
```

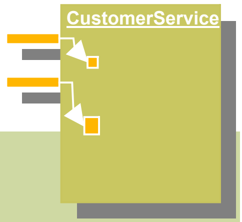
Controller Logic/User Interaction

Business Logic

- **Programs**
  - Used for single point of entry situations
  - TUI program, Batch program, GUI program

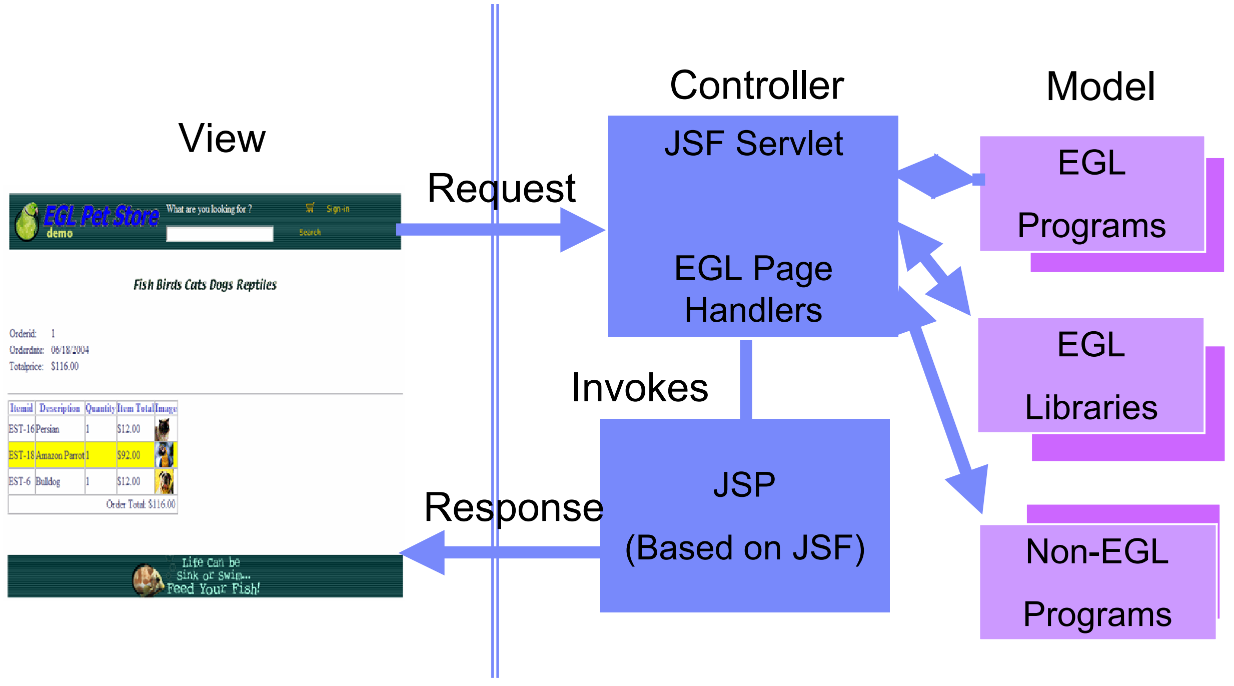
```
Program MyProgram

function myFunction()
      salutation char(30);
      call helloworld salutation;
end
```

- **Services**
  - "Business Logic" for web apps
  - Multiple entry points
  - Invoke by function



```
Service Customer

Function getAllCustomers()
end
...

end
```

EGL and JSF



Orderid: 1
Orderdate: 06/18/2004
Totalprice: $116.00

Itemid	Description	Quantity	Item Total	Image
EST-16	Persian	1	$12.00	
EST-18	Amazon Parrot	1	$92.00	
EST-6	Bulldog	1	$12.00	
			Order Total: $116.00	

JSP's / JSF / EGL and COBOL

- **JSP's are synonymous with EXEC CICS Send Map and Receive Map processing**
  - If a CICS program only processed screens – to request business processing – or work – it would need to either Link, XCTL or Calls in COBOL.
  - JSP is a similiar concept.
- **Java Server Faces provides a framework to build UI oriented forms linked with processes such as Web Services.**
  - Performs similar function as existing CICS programs which perform send/receive processing and input validation.
- **Java server faces consist of Java Server pages – which handle the build and catching of forms and user information – and page handlers which validate information and provide control calls into back end services.**

EGL Web – C.I.C.S. Programming Similarities

Page Data
~
BMS Map Fields

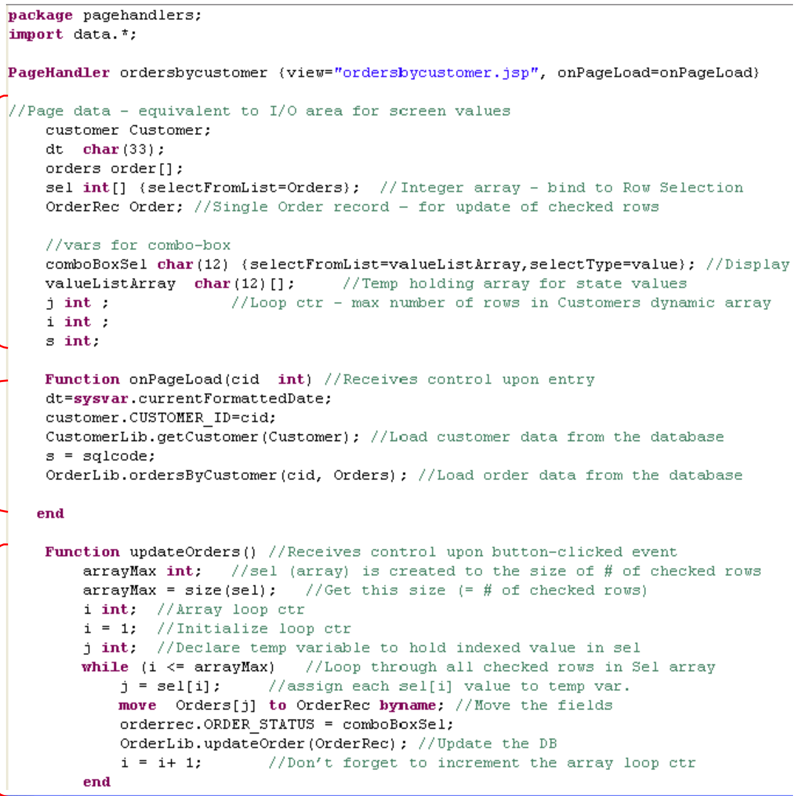
Load values from the database

“Send map”

“Receive map”

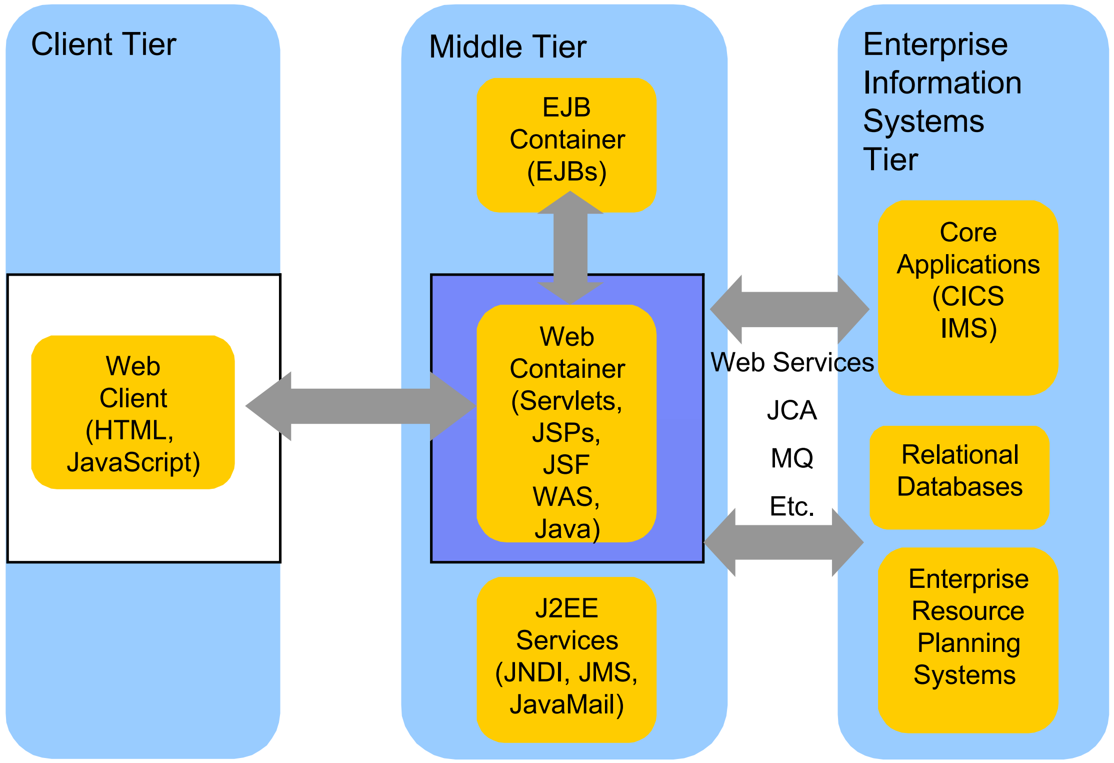
Process user-input values

Update Database



```
package pagehandlers;
import data.*;

PageHandler ordersbycustomer {view="ordersbycustomer.jsp", onPageLoad=onPageLoad}

//Page data - equivalent to I/O area for screen values
    customer Customer;
    dt   char(33);
    orders order[];
    sel int[] {selectFromList=Orders};  //Integer array - bind to Row Selection
    OrderRec Order; //Single Order record - for update of checked rows

    //vars for combo-box
    comboBoxSel char(12) {selectFromList=valueListArray,selectType=value}; //Display
    valueListArray  char(12)[];     //Temp holding array for state values
    j int ;             //Loop ctr - max number of rows in Customers dynamic array
    i int ;
    s int;

    Function onPageLoad(cid  int) //Receives control upon entry
    dt=sysvar.currentFormattedDate;
    customer.CUSTOMER_ID=cid;
    CustomerLib.getCustomer(Customer); //Load customer data from the database
    s = sqlcode;
    OrderLib.ordersByCustomer(cid, Orders); //Load order data from the database

   end

    Function updateOrders() //Receives control upon button-clicked event
        arrayMax int;   //sel (array) is created to the size of # of checked rows
        arrayMax = size(sel);   //Get this size (= # of checked rows)
        i int;  //Array loop ctr
        i = 1;  //Initialize loop ctr
        j int;  //Declare temp variable to hold indexed value in sel
        while (i <= arrayMax)   //Loop through all checked rows in Sel array
            j = sel[i];      //assign each sel[i] value to temp var.
            move  Orders[j] to OrderRec byname; //Move the fields
            orderrec.ORDER_STATUS = comboBoxSel;
            OrderLib.updateOrder(OrderRec); //Update the DB
            i = i+ 1;       //Don't forget to increment the array loop ctr
        end
```

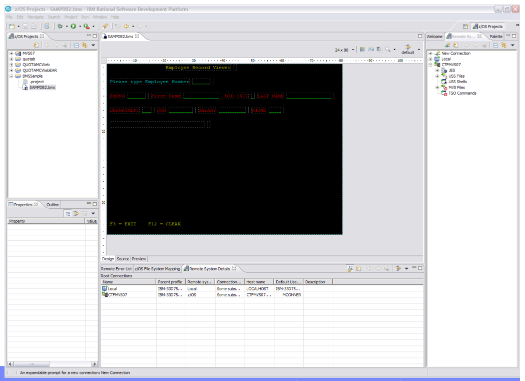
The Client


Client Tier
Web Client (HTML, JavaScript)
Middle Tier
EJB Container (EJBs)
Web Container (Servlets, JSPs, JSF WAS, Java)
J2EE Services (JNDI, JMS, JavaMail)
Web Services
JCA
MQ
Etc.
Enterprise Information Systems Tier
Core Applications (CICS IMS)
Relational Databases
Enterprise Resource Planning Systems

HTML

- **HTML performs similar processing as BMS or MFS maps. It defines the screens and fields, colors, and interactions, although the technologies and implementations of course are different.**
- **Hypertext Markup Language consists of:**
  - **Hypertext.** The way of creating web documents – and of linking multiple documents together. HTML offers support for both document as well as multimedia links.
  - Tags or controls: Pieces of code that are used to create links. All browsers let you know when you've selected an active area of the screen.
    - For example <head> marks where a heading starts and </head> marks where it ends.
    - Popular tags include:
      - Text Tags – Logical structure for content
      - Link Tags – to links such as hyperlinks, image links
      - Style sheet tags – how content is rendered
      - and many more….

- **See the green screenshot – displayed inside of the WDz BMS Map Editor, together with with the BMS Macros that are input to generate the code – that upon execution causes the "green screen" to be displayed.**
  - WDz provides similar support for HTML screens



```
                    Employee Record Viewer

Please type Employee Number

EMPNO      First Name              MID INIT   LAST NAME

DEPARTMENT     JOB          SALARY          PHONE




F3 - EXIT     F12 - CLEAR
```

Root Connections

Name	Parent profile	Remote sys...	Connection...	Host name	Default Use...	Description
Local	IBM-33D75...	Local	Some subs...	LOCALHOST	IBM-33D75...	
CTFMVS07	IBM-33D75...	z/OS	Some subs...	CTFMVS07....	MCONNER	

BMS and HTML

BMS

Name and overall format of map - Includes items such as input/output, whether keyboard should be enabled, types of terminal, colors, size etc. are defined.

```
SAMPDB2  DFHMSD
    TYPE=&SYSPARM,MODE=INOUT,LANG=COBOL,STORAGE=AUTO,
    *
        CTRL=FREEKB,EXTATT=YES,TERM=3270-2,TIOAPFX=YES,      *
        MAPATTS=(COLOR,HILIGHT,OUTLINE,PS,SOSI),             *
        DSATTS=(COLOR,HILIGHT,OUTLINE,PS,SOSI)
MAP1    DFHMDI SIZE=(24,80),                                 *
        COLUMN=1,                                            *
        LINE=1
```

Headings and text fields. Defined with DFHMDF macro. You see position, length, initial value, and field attribute below.

```
      DFHMDF POS=(3,1),LENGTH=27,                            *
        INITIAL='Please type Employee Number',               *
        ATTRB=(PROT,NORM)
```

Input Fields. Defined with DFHMDF macro. You see a name (which ultimately defines storage size (and Cobol copybook field definition), and a difference with the field defined as unprotected – information can be entered.

```
EMPONUMINPUT DFHMDF POS=(3,29),LENGTH=6,                     *
        ATTRB=(UNPROT,NORM),HILIGHT=UNDERLINE
```

HTML

Headings – Overall definition, including whether Java Server faces tags will be used, a heading, and stylesheet definition.

```
<HEAD>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ page language="java" contentType="text/html; charset=CP1252"
    pageEncoding="CP1252"%>
<META http-equiv="Content-Type" content="text/html; charset=CP1252">
<META name="GENERATOR" content="IBM Software Development Platform">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>MAP1</TITLE>
```

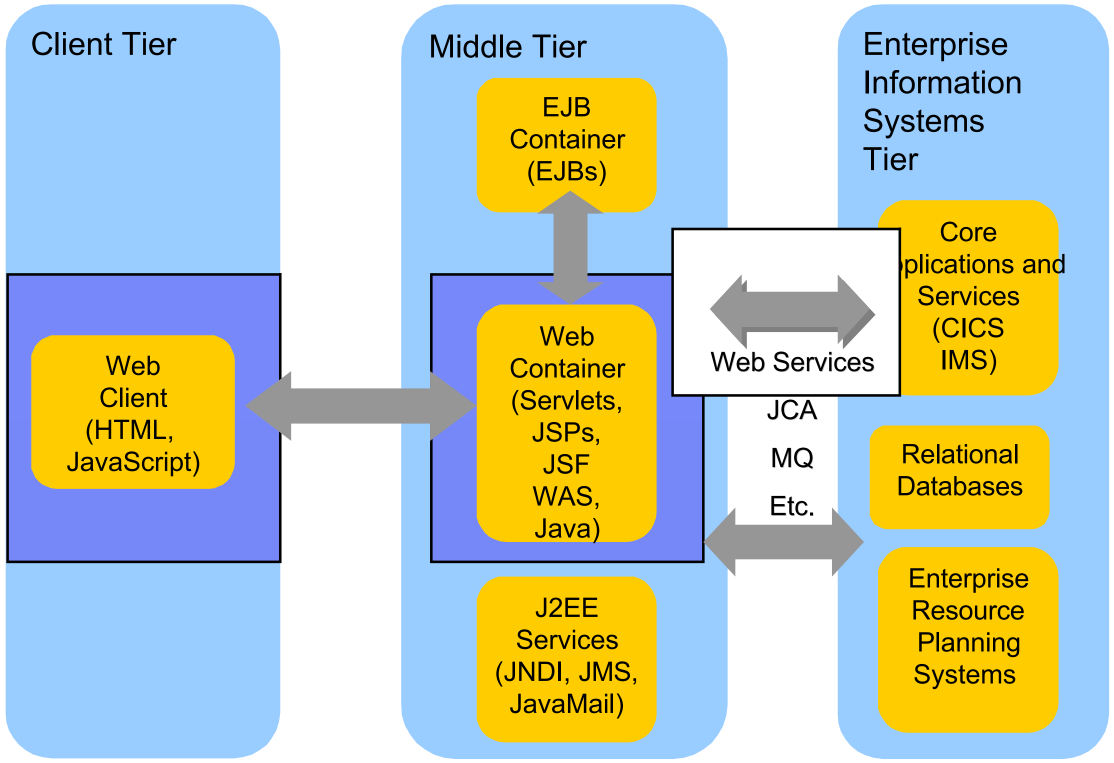
Text headings including location definition, colors, attributes, etc.

```
f:view> <BODY>
<hx:scriptCollector id="scriptCollector1"><h:form styleClass="form" dir="ltr"
    id="form1"><table><tr><td colspan="20">&nbsp;</td>
<td colspan="22" nowrap><font color="#ffff00">Employee Record Viewer</font></td>
<td>&nbsp;</td>
<td nowrap><font color="#0000ff"></font></td>
<td colspan="36">&nbsp;</td>
<tr><td colspan="80">&nbsp;</td>
<tr><td>&nbsp;</td>
<td colspan="27" nowrap><font color="#00ffff">Please type Employee Number</font></td>
```

Input fields

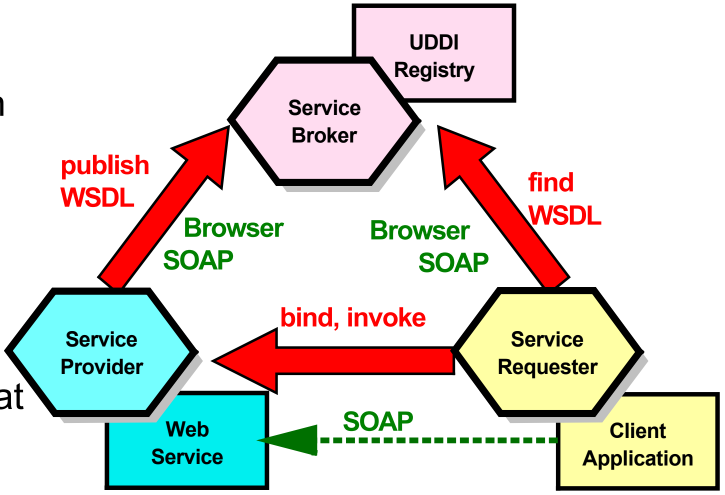
```
<td>&nbsp;</td>
<td colspan="6" nowrap><h:inputText
    value="#{pc_MAP1Page.map1Bean.emponuminput}" required="false" style="color:
    #00ff00" size="6" id="emponuminput"></h:inputText></td>
<td>&nbsp;</td>
<td colspan="44">&nbsp;</td>
<tr><td colspan="80">&nbsp;</td>
<tr><td>&nbsp;</td>
```

Connectivity


Client Tier
Web Client (HTML, JavaScript)
Middle Tier
EJB Container (EJBs)
Web Container (Servlets, JSPs, JSF WAS, Java)
J2EE Services (JNDI, JMS, JavaMail)
Web Services
JCA
MQ
Etc.
Enterprise Information Systems Tier
Core plications and Services (CICS IMS)
Relational Databases
Enterprise Resource Planning Systems

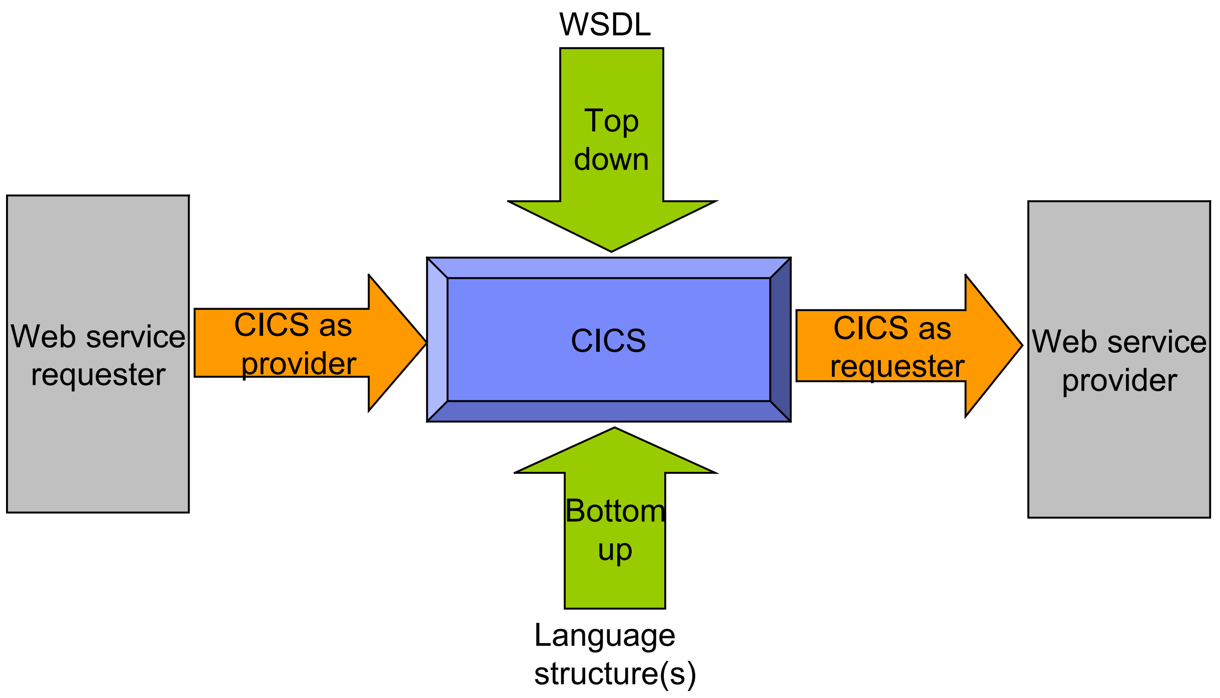
Web Services

- **Architecture for**
  - Application to application
    - Communication
    - Interoperation
- **Definition:**
  - Web Services are **software components described via WSDL** that are capable of being accessed via **standard** network protocols such as SOAP over HTTP
- **WS-I.org (Web Services Interoperablity Organization)**
  - Ensure interoperability

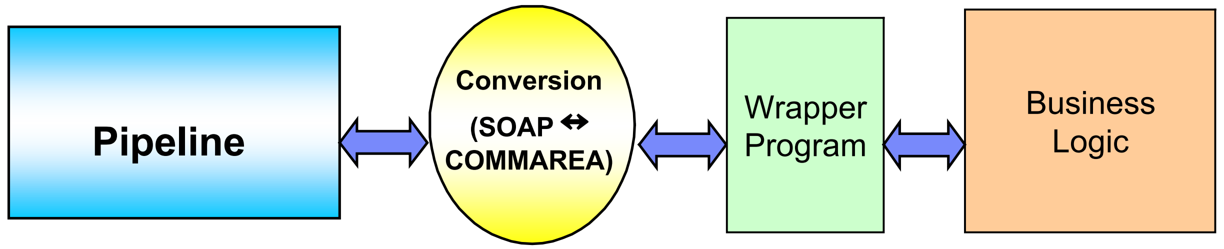


The entire industry is agreeing on one set of standards !!

Web Services Enablement Styles



Where a wrapper program fits in



XML Terminology

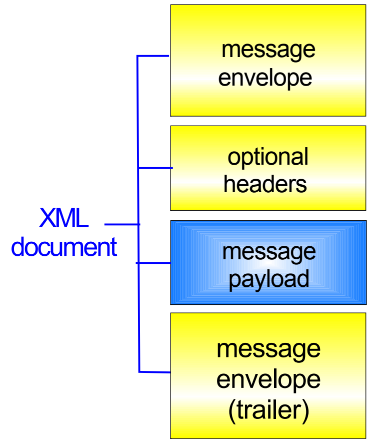
- **SOAP and WSDL are based on XML**
- **A tag / attribute based syntax**
- **Format of XML file described in**
  - **DTD** – Document Type Definition
  - **XSD** – XML Schema Definition
- **XML files are**
  - Well-formed (syntax is ok – matching tabs, etc.)
  - Valid (obeys rules in DTD or XSD) (CICS can validate)
- **Namespaces**
  - Avoids name collisions
  - A set of names (XML tags) that apply to a certain space in a document

XML – Basic Parts

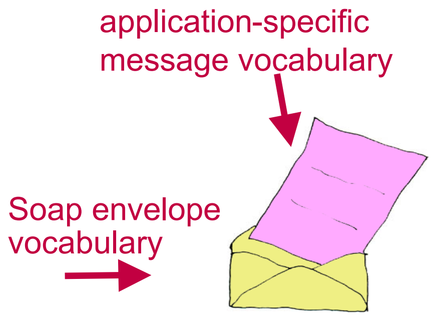
```
<?xml version="1.0" standalone="no" encoding="UTF-8" ?>          XML Declaration
<!DOCTYPE shirt SYSTEM "http://shirts.com/xml/dtds/shirt.dtd">   Document
<shirt>                                        root element      type
   <model>CICS Tee</model>             child of root             declaration
   <brand>Tommy Hilltop</brand>                end tag
                                                  start tag
   <price currency="USD“>10.95</price>         attribute
   <fabric content="70%“>cotton</fabric>       attribute
   <fabric content="30%">polyester</fabric>
   <on_sale/>
                                               empty element
   <options>
      <colorOptions>
         <color>red</color>
         <color>white</color>
      </colorOptions>
      <sizeOptions>
         <!-- Medium and large are out of stock -->          comment
         <size>small</size>
         <size>x-large</size>
      </sizeOptions>
   </options>
   <order_info>Call &phone;</order_info>
</shirt>                                           entity reference
```

Simple Object Access Protocol (SOAP)

- An XML-based protocol for exchanging of information in a decentralized, distributed environment
- An open standard whose main goal is to facilitate interoperability
- A protocol which is not tied to any operating system, transport protocol, programming language, or component technology



- ***XML Message Envelope***
  - service requested
  - routing information
  - message type
  - date/time stamp
- ***XML Message Headers***
  - authentication
  - transaction context
- ***XML Message Payload***
  - data understood by target application
- ***XML Message Trailer***
  - closing tags
  - optional message digest



SOAP spec defines how to do this!

SOAP: Request Message

```
<SOAP-ENV:Envelope
       xmlns:SOAP-ENV=
     "http://www.w3.org/2001/06/soap-envelope"
       SOAP-ENV:encodingStyle=
     "http://www.w3.org/2001/06/soap-encoding">

   <SOAP-ENV:Body>
       <m:GetLastTradePrice xmlns:m="Some-URI">
           <symbol>IBM</symbol>
       </m:GetLastTradePrice>
   </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

app-specific message

SOAP envelope

SOAP: Response Message

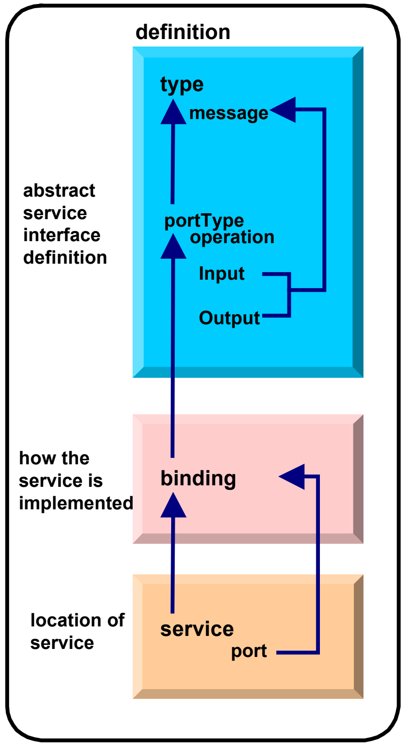
Result returned in Body

```
<SOAP-ENV:Envelope
        xmlns:SOAP-ENV=
      "http://www.w3.org/2001/06/soap-envelope"
        SOAP-ENV:encodingStyle=
      "http://www.w3.org/2001/06/soap-encoding">

   <SOAP-ENV:Body>
      <m:GetLastTradePriceResponse
         xmlns:m="Some-URI">
          <Price>134</Price>
      </m:GetLastTradePriceResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

app-specific message

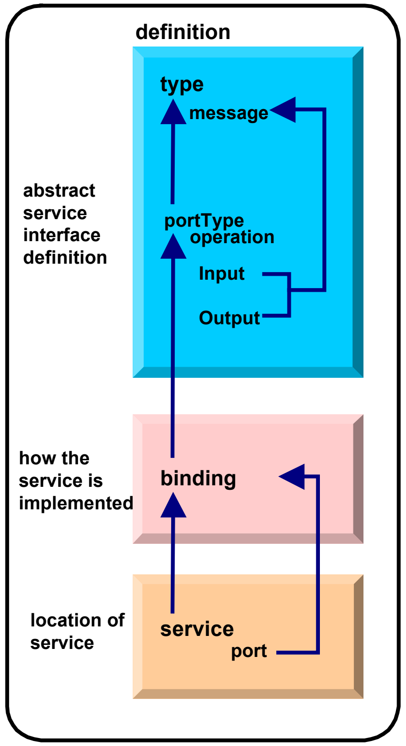
WSDL - Web Service Description Language

- Open Standard
- XML resume describing what a Web Service can do, where it resides, and how to invoke it
- Machine readable, generated, used by IDEs
- Similar in purpose to IDL, but in XML form
- Can be One or multiple documents
- Major sections are:
  - Service Interface (operations, input, output)
  - Service binding (protocol binding)
  - Service implementation (location of service)



WSDL: Logical Contents

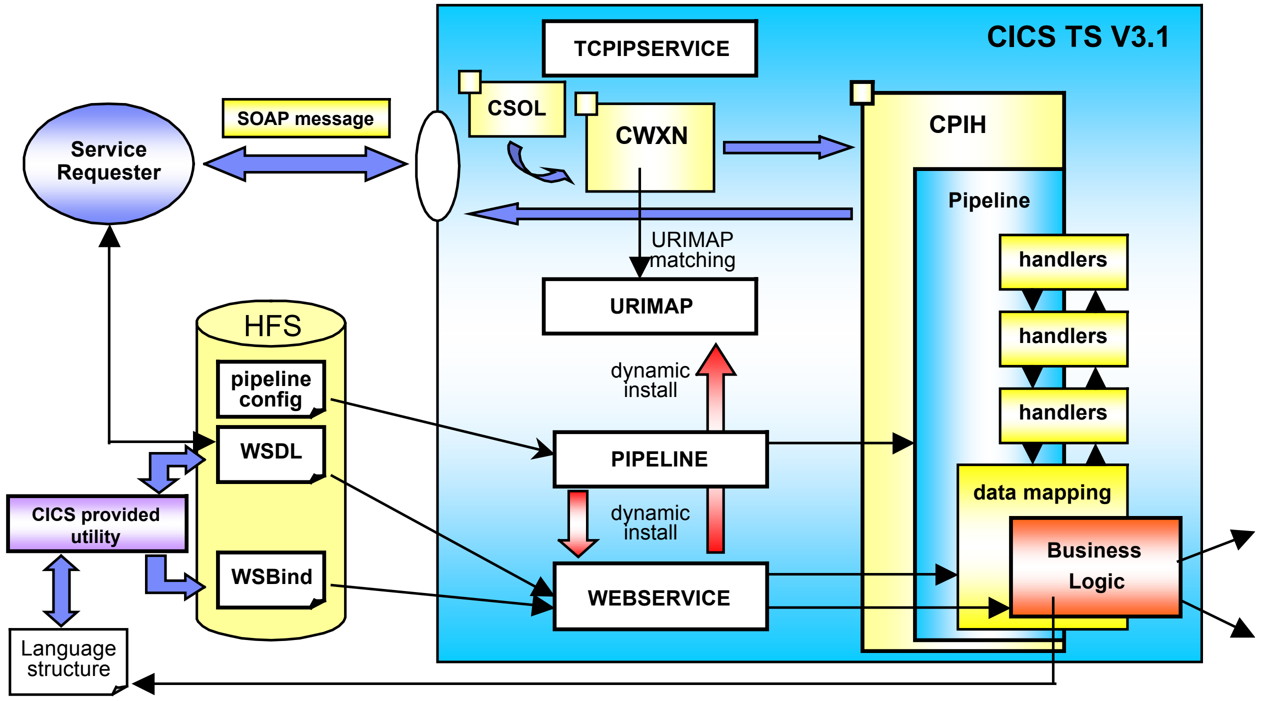
- **Service Interface**
  - Operation (business functions)
    - Input Message (0 or 1) and Output Message (0 or 1)
      - 1 or more parts
      - Parts may be simple or complex
      - Complex parts may have multiple elements
- **Service binding**
  - Definition of the physical service interface implementation
- **Service Implementation**
  - Location of the service



WSDL: Physical Contents

- **Definitions – highest level tag**
  - **types** – definition of complex parts
  - **message** – a grouping of 1 or more parts
    - **parts** – simple or complex (complex points to a type)
  - **portType** – a grouping of operations
    - **operation** – correspond to business functions
      - **input** – points to input message
      - **output** – points to output message
      - **fault** – can be returned when stuff goes wrong
  - **binding** – physical associations to operations
    - **operation** – implementation of a portType operation
  - **service** – grouping of ports
    - **port** – location of associated binding

CICS as a service provider

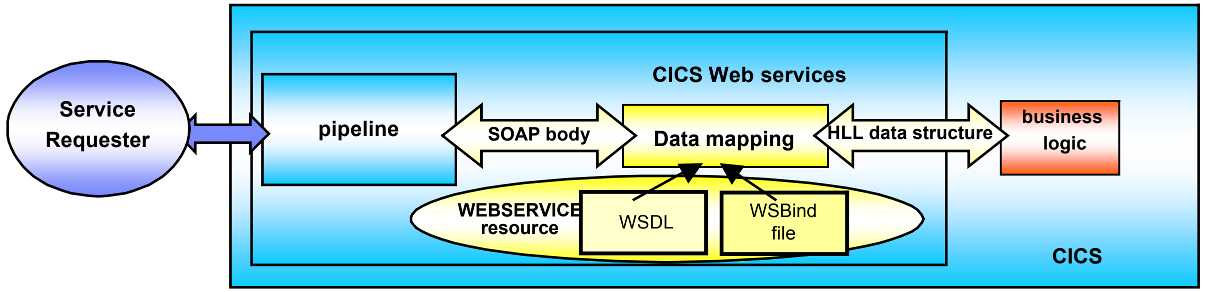


Defining the CICS Web Services Resources

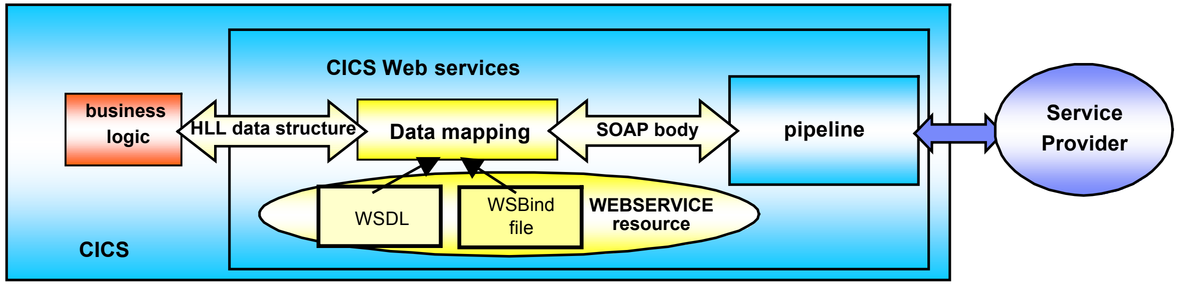
- **Define a TCPIPSERVICE (or WMQ) and a PIPELINE**
- **Then install the PIPELINE definition and issue CEMT PERFORM PIPELINE SCAN**
- **CICS uses the PIPELINE definition to**
  - Locate the WSBind file
  - From the WSBind file, CICS will dynamically create a WEBSERVICE resource
  - CICS will also dynamically create a URIMAP definition
- **Can define everything individually if preferred**

CICS usage of the WSBind file

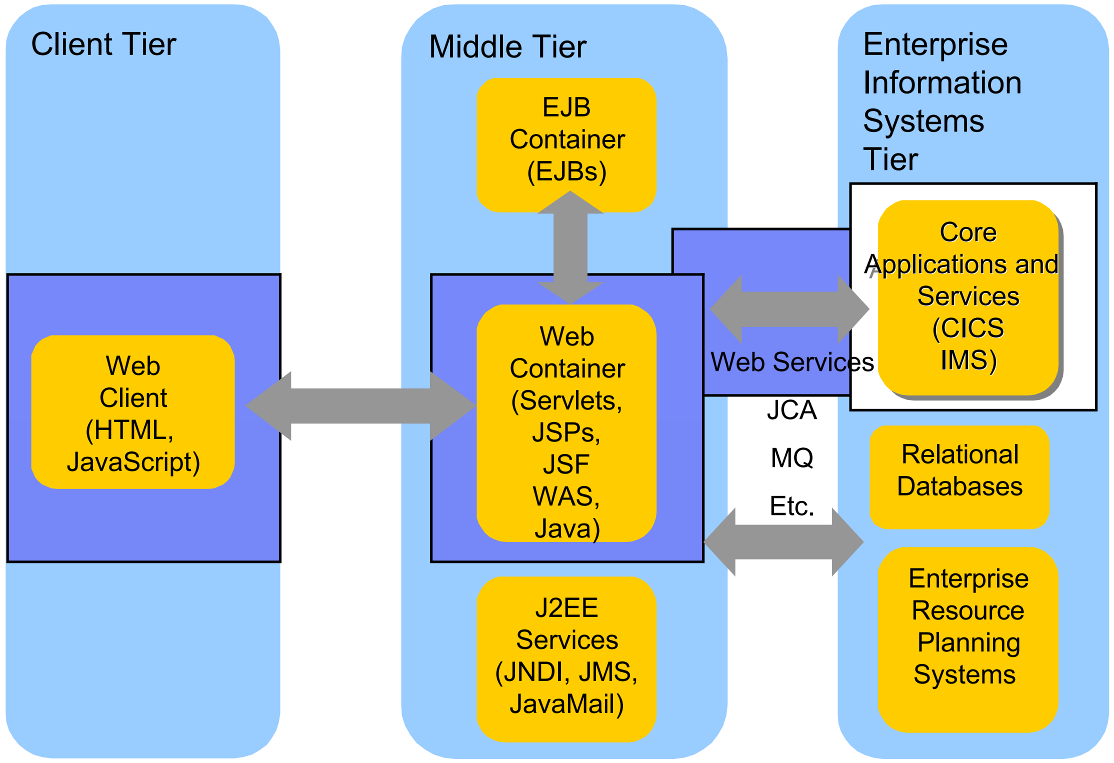
- **CICS as a service provider**



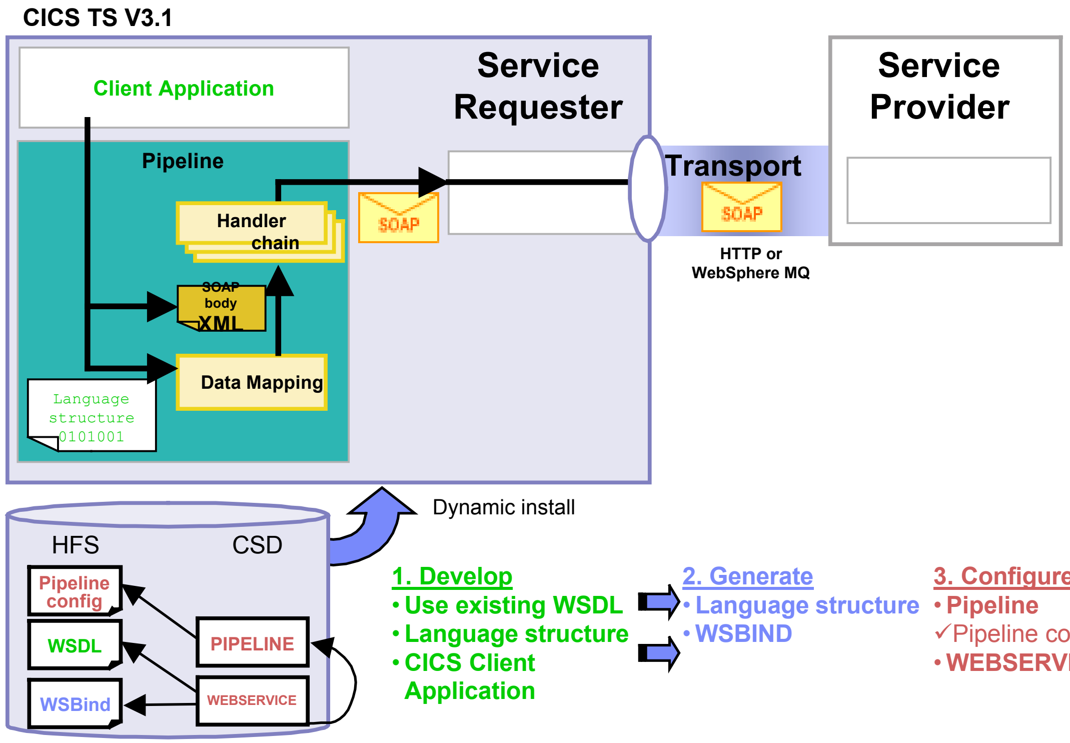
- **CICS as a service requester**



The Business Tier


Client Tier
Web Client (HTML, JavaScript)
Middle Tier
EJB Container (EJBs)
Web Container (Servlets, JSPs, JSF WAS, Java)
J2EE Services (JNDI, JMS, JavaMail)
Web Services
JCA
MQ
Etc.
Enterprise Information Systems Tier
Core Applications and Services (CICS IMS)
Relational Databases
Enterprise Resource Planning Systems

CICS as a Web service requester



1. Develop
- Use existing WSDL
- Language structure
- CICS Client Application

2. Generate
- Language structure
- WSBIND

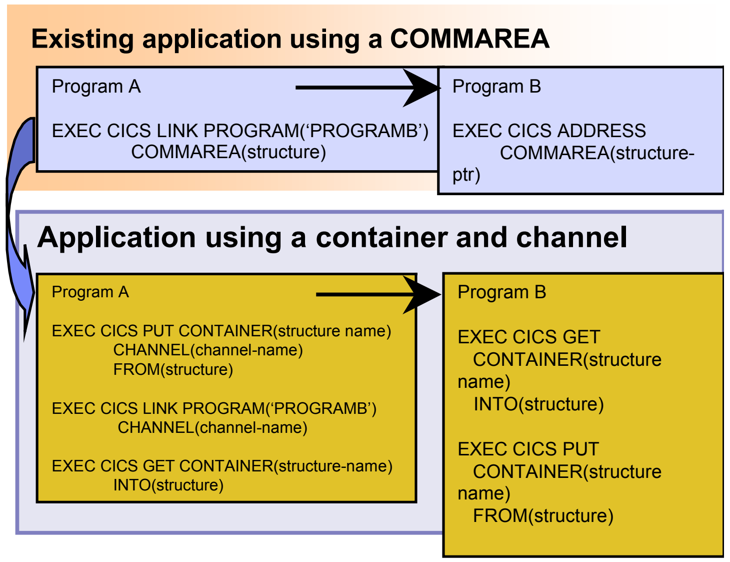
3. Configure
- Pipeline
✓Pipeline configuration
- WEBSERVICE

CICS API's

- **Invoking a Web Service from a CICS application program**
  - CICS as a service requester
    - EXEC CICS INVOKE WEBSERVICE ( ) CHANNEL ( )   URI ( ) OPERATION ( )
      - WEBSERVICE: name of the Web Service to be invoked
      - CHANNEL: name of the channel containing data to be passed to the Web Service (DFHWS-DATA container)
      - URI: Universal Resource Identifier of the Web Service (optional)
      - OPERATION: name of the operation to be invoked

Data Exchange between CICS programs with Containers and Channels

- Offers a more flexible and intuitive alternative to the COMMAREA
- Enables large amounts of data to be passed between CICS applications
  - Not subject to 32KB restriction
- Optimized and managed by CICS
- Requires minimal application changes required to use



Existing application using a COMMAREA

Program A

```
EXEC CICS LINK PROGRAM('PROGRAMB')
         COMMAREA(structure)
```

Program B

```
EXEC CICS ADDRESS
      COMMAREA(structure-
ptr)
```

Application using a container and channel

Program A

```
EXEC CICS PUT CONTAINER(structure name)
        CHANNEL(channel-name)
        FROM(structure)

EXEC CICS LINK PROGRAM('PROGRAMB')
        CHANNEL(channel-name)

EXEC CICS GET CONTAINER(structure-name)
        INTO(structure)
```

Program B

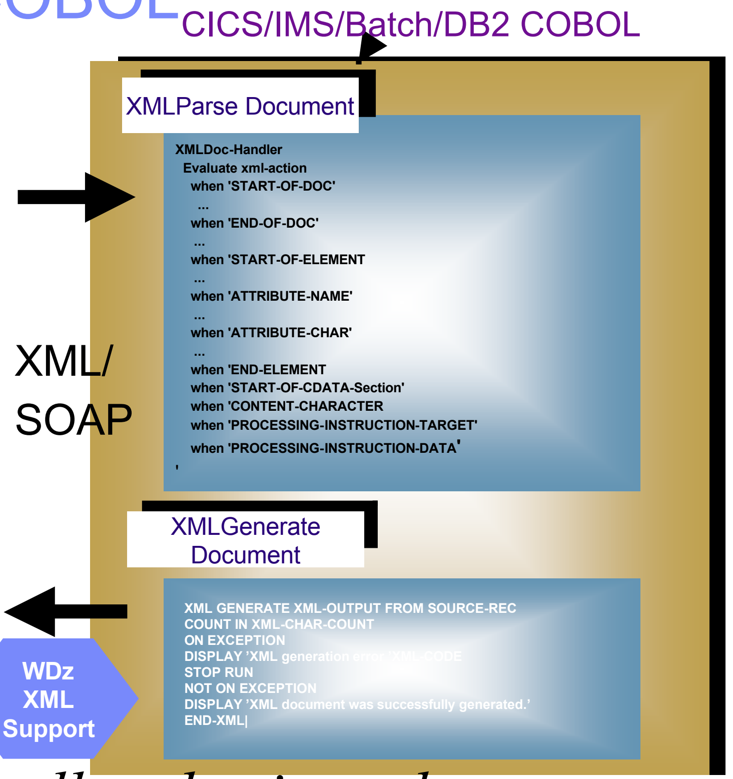
```
EXEC CICS GET
  CONTAINER(structure
name)
  INTO(structure)

EXEC CICS PUT
  CONTAINER(structure
name)
  FROM(structure)
```

IBM Enterprise COBOL

- **XML Language based generation from COBOL data structure**
  - XMLGenerate Verb
  - WebSphere EJB support
  - DB2 V8
- **High speed XML Sax based parsing**
- **Object Oriented Support for Java COBOL Interoperability**
- **Unicode support**
- **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 TIME



```
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 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|
```

COBOL is an excellent business language

Why COBOL?

- Large portfolios
- Many developers
- High performance
- Self documenting
- Proven Maintainability
- Business oriented, eases technology burden

Summary

- **MVC application model provides high levels of flexibility**
- **CICS provides leading edge support of Web Services**
  - Allows for re-use of existing business assets and new development of high QOS assets
- **Developers need "complete" application skills**
- **CICS and WebSphere Application Server are strategic middleware products that together…**
  - Interoperate - Web services, JCA, Enterprise JavaBeans
  - Exploit and complement z/OS qualities of service
  - Have high qualities of service, low cost per transaction, excellent security.

Demo

Modern Application Architecture – Building and testing a JSF/COBOL process.

- **Demo of WDz used to create a simple, understandable visual and business application process for deployment.**
- **The session shows how to build and deploy composite CICS and WebSphere applications using the IBM WebSphere Studio tooling and the Enterprise Compilers. Composite applications are applications which are assembled from independent component parts, using Web and Web Services standards.**

Software Group

Modern Application Architecture

SOA Introduction

Paolo Chieregatti
Certified IT Specialist
paolo.chieregatti@it.ibm.com

Agenda

- **IT Market : trend & directions**
- **Service Oriented Architecture**
- **COBOL & Enterprise Application : Today**
- **SOA and System z Application Lifecycle**

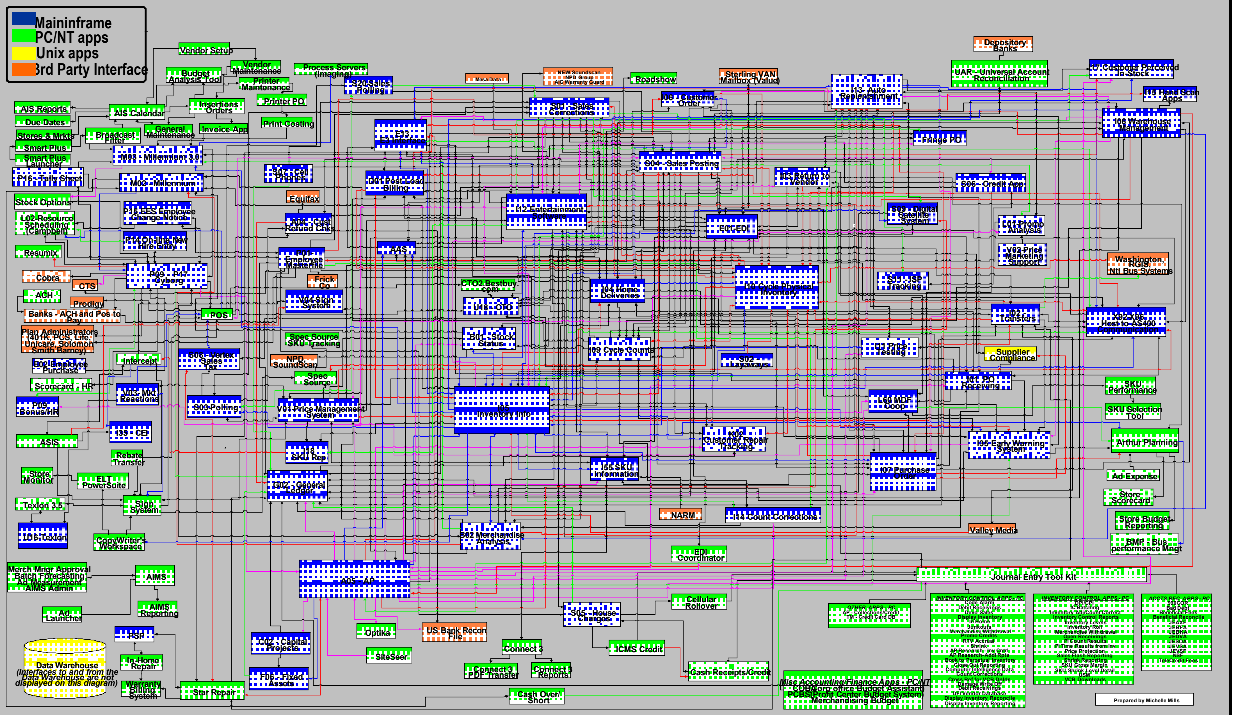
IT Market : trend & directions

- **SOA : Service Oriented Architecture**
- **Virtualization & Consolidation**
- **Web 2.0 : WOA Web Oriented Architecture**

- **Second Life : ?**

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


🟦 Maininframe
🟩 PC/NT apps
🟨 Unix apps
🟧 3rd Party Interface
Data Warehouse (Interfaces to and from the Data Warehouse are not displayed on this diagram)
Journal Entry Tool Kit
Prepared by Michelle Mills

Actual Application Architecture for Consumer Electronics Company

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?

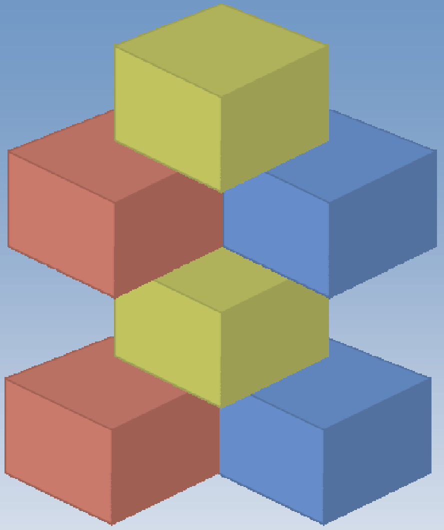
A set of **related & integrated** services that support a business process built on an SOA

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

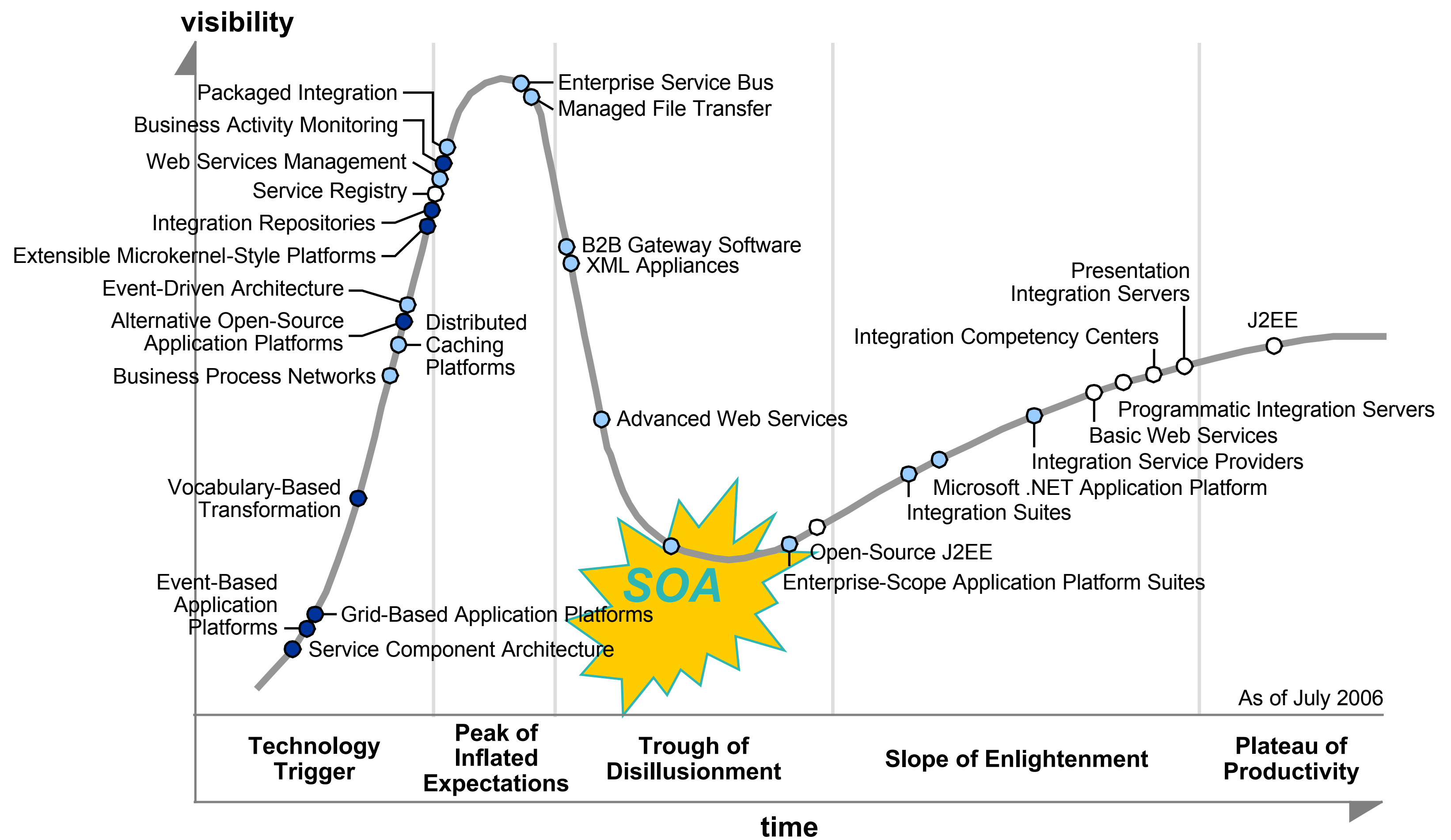
- Orchestrated flows of Services
  - Tooling
- Separated logic
  - Process flow
  - Connectivity
  - Business
- Flexible high QOS Business Functions

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

SOA in the Trough of Disillusionment: Bad News or Good News?

Application Integration & Platform Middleware Hype Cycle



Years to mainstream adoption:

○ less than 2 years ● 2 to 5 years ● 5 to 10 years △ more than 10 years ⊗ obsolete before plateau

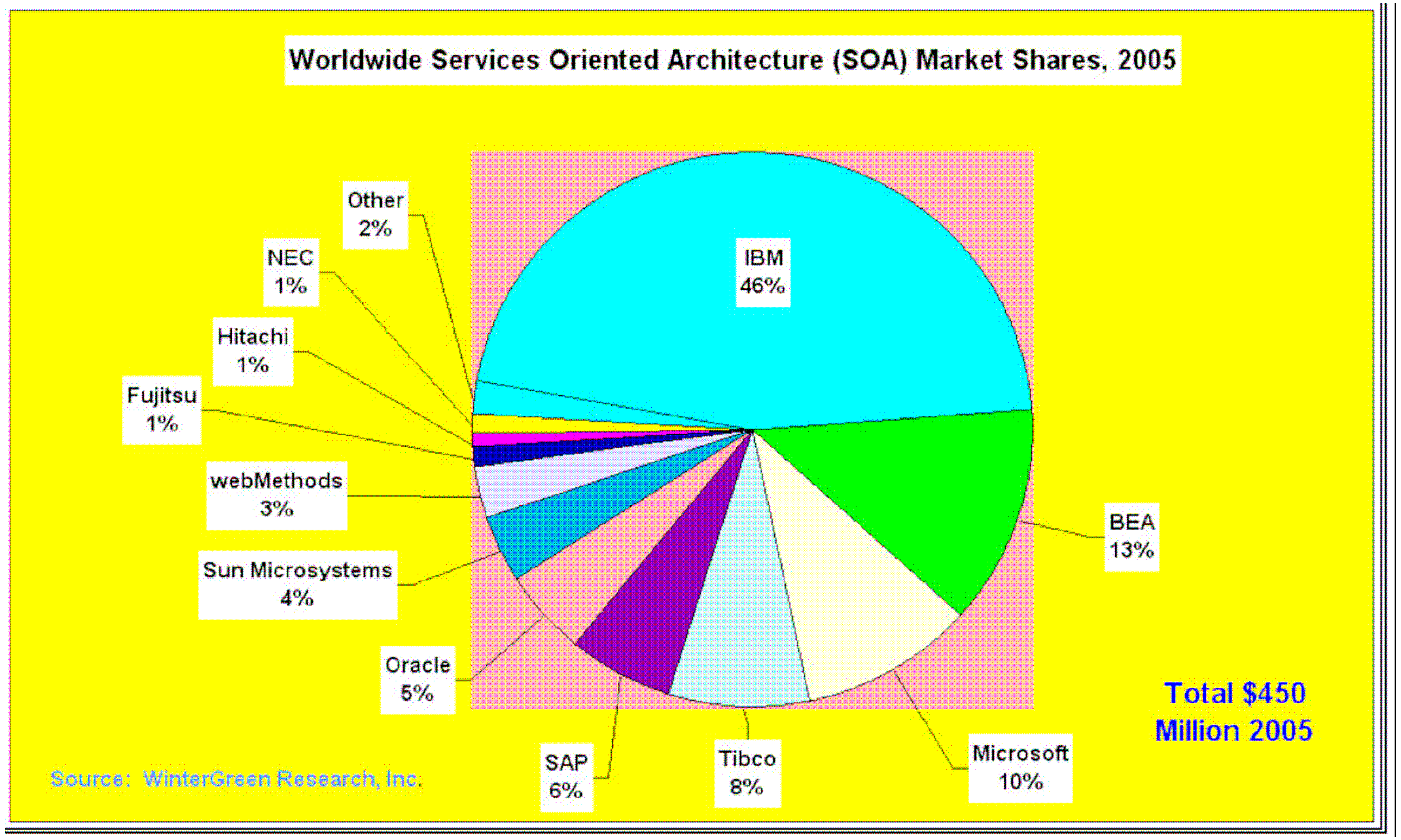
Source: Gartner

IT Market : Analysts

- **“Service oriented architecture (SOA) markets at $450 million in 2005 are expected to reach $18.4 billion by 2012. Market growth comes because SOA enables the flexible IT architecture that is needed to respond to market shifts brought by speeded product cycles and competitive challenges.”**

WinterGreen Research

…. IT Market : Analysts



Stages of SOA Adoption

	Stage 1 Introduction	Stage 2 Spreading	Stage 3 Exploitation	Stage 4 Plateau
Business Goals	Address Specific Pain (e.g., Customer Portal)	Process Integration (e.g., B2B)	Process Flexibility (e.g., Time to Market)	Continuous Adaptation & Evolution
IT Goals	Proof of Concept	Establish Technology Platform	Leverage Services Sharing	Enterprise SOA Infrastructure
Scope	Single Application	Multiple Applications (Single BU)	Multiple Applications (Cross BUs)	Virtual Enterprise
No. of Published Services*	<25	<100	<500	>500
No. of Service Consumers*	<5	<25	<50	>50
No. of Service Calls/Day*	<10,000	<100,000	<1,000,000	>1,000,000
No. of Service Developers*	<10	<20	<100	>100
Enabling Technology (cumulative)	Application Server, Portal, Adapters	ESB, WSM Integr. Suite, B2B	SOA Reg/Rep BPM Policy Mgmt.	Enterprise SOA Backplane

** =These figures represent typical scenarios, but they may vary considerably depending on the specific organization's requirements.*

Source: Gartner

Business Flexibility Is A Top Priority

·

Business Flexibility the ability to be more responsive to changing market conditions, including opportunities, customers and competitive actions.

Business Flexibility is best achieved with SOA.
Flexible business requires a flexible IT… SOA is an evolutionary approach to building flexible IT systems focused on solving business problems

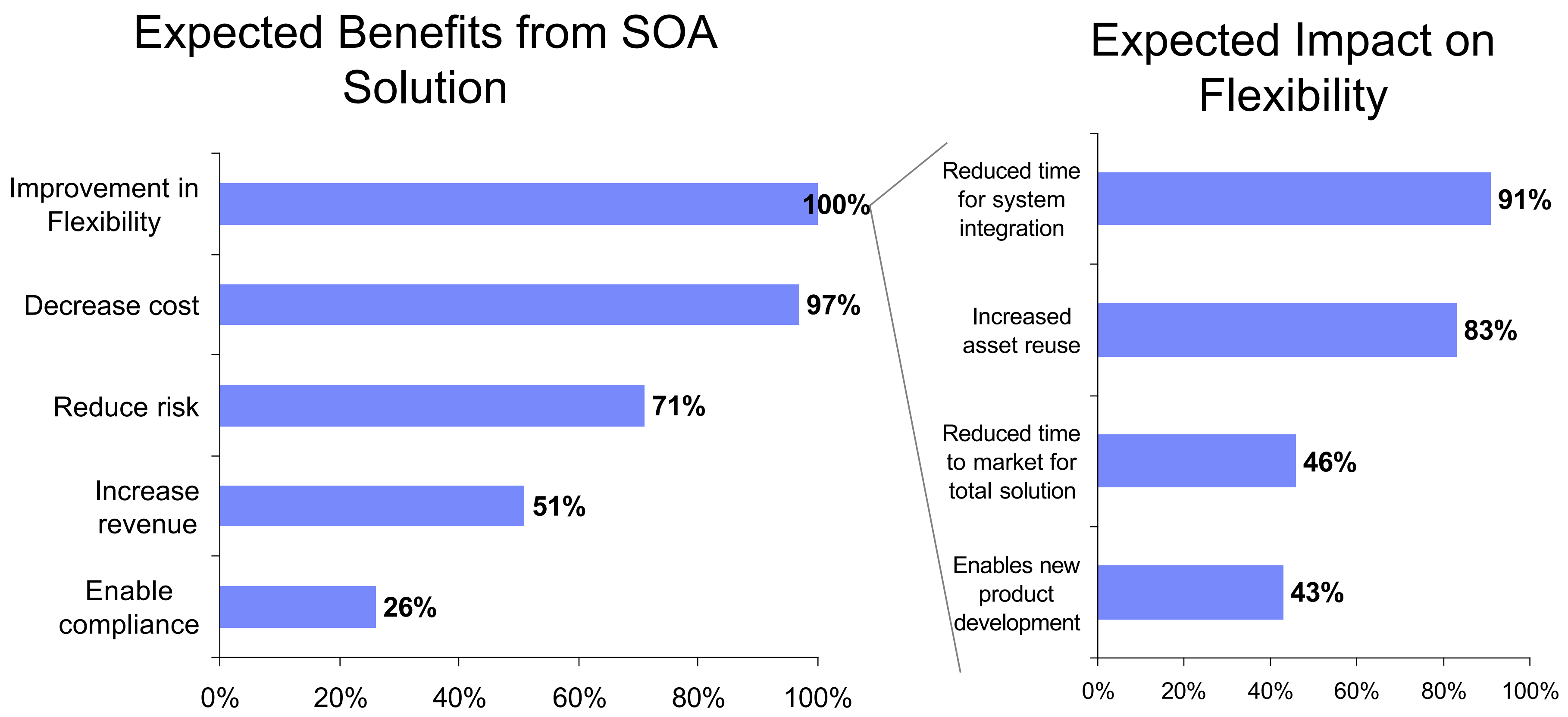
"Business flexibility is a key enabler of business innovation."
David Cearley, Gartner

"The power of business flexibility is the power of business to take advantage of new and unexpected opportunities, and this is what makes businesses competitive."
Ron Schmelzer, ZapThink

"The only way customers gain competitive advantage is through business flexibility. Without flexibility at the business and technical level, innovation isn't achievable."
Judith Hurwitz, Hurwitz & Associates

"When we can respond to rapid business changes with the flexibility to optimize our business opportunities, we will have moved to the next stage of IT."
Amy Wohl, Wohl Associates

SOA projects improved business flexibility 100% of the times



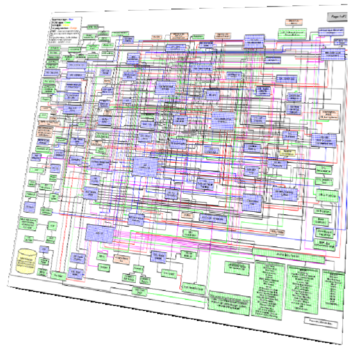
Agenda

- **IT Market : trend & directions**
- **Service Oriented Architecture**
- **COBOL & Enterprise Application : Today**
- **SOA and the System z Application Lifecycle**

SOA & Application Development

- **Reuse enterprise application**
- **Componentization**
- **Modernization**
- **Adoption of open standard**
- **New SOA application**

SOA
the next step

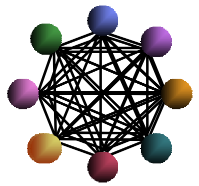


Direct Connectivity

Connectivity, mediation & additional logic

Application

Point-to-Point connection between applications



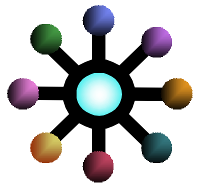
Message Queuing

Connectivity logic

Mediation & additional logic

Application

Applications via a centralized hub



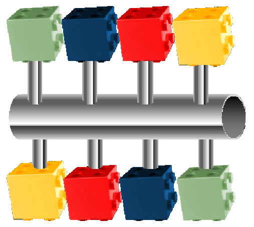
Message Brokering

Connectivity and mediation logic

Additional logic

Application

Integration and choreography of services through an Enterprise Service Bus

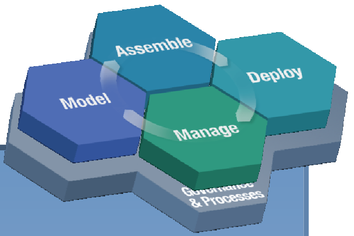


Service Orientation

Connectivity, mediation & additional logic

Application Services

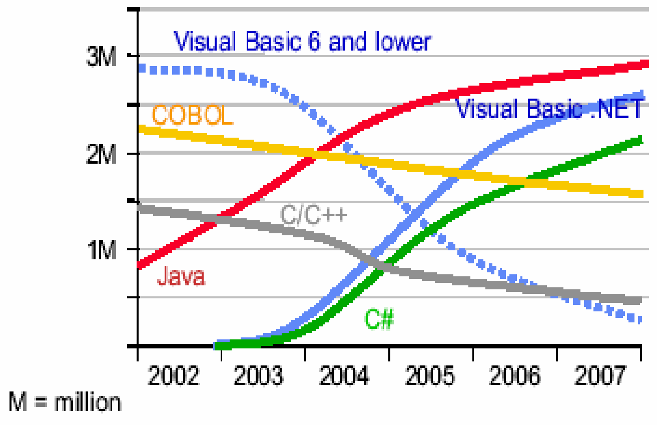
What about "before SOA"?



- **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?

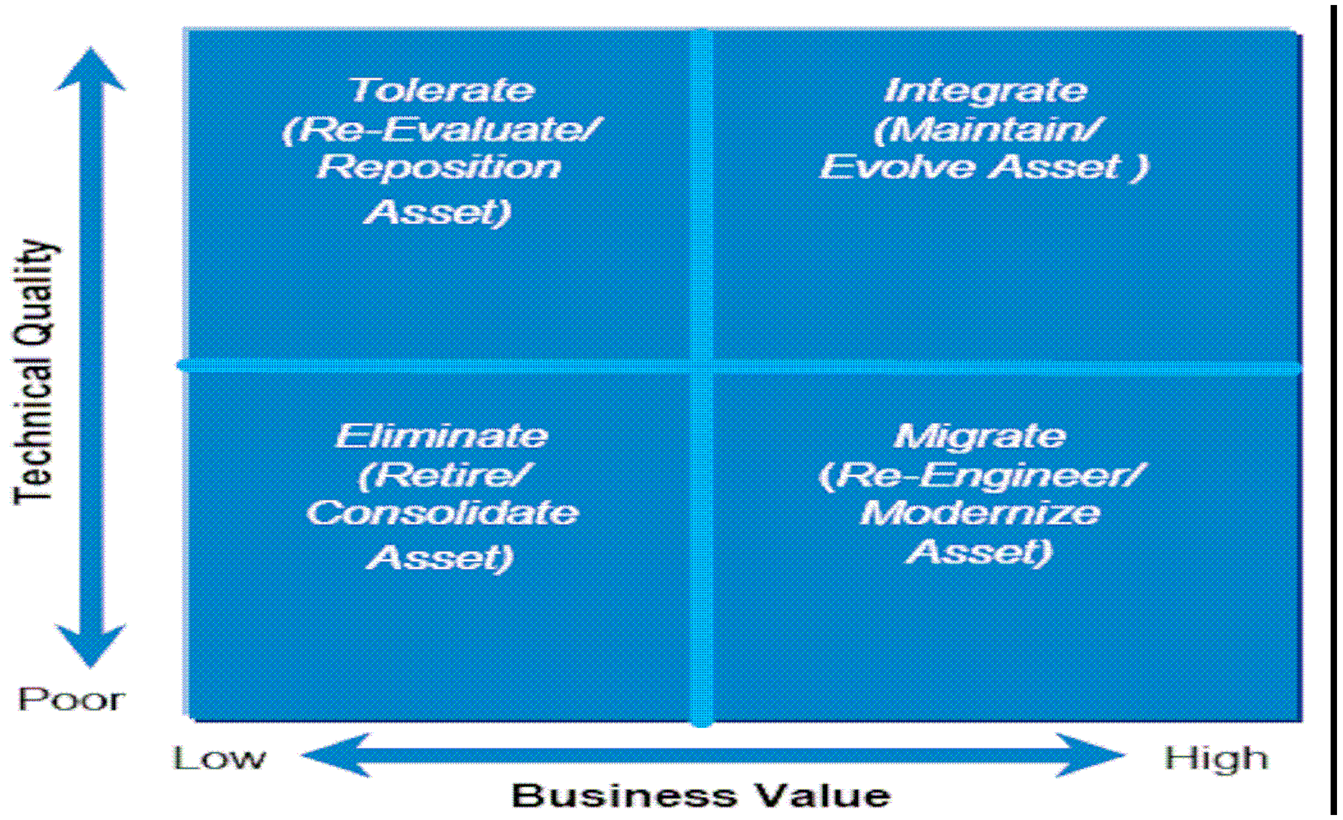
Developers
From an estimated worldwide market size of 7 million "professional" developers



M = million

Gartner

Application Portfolio Analysis

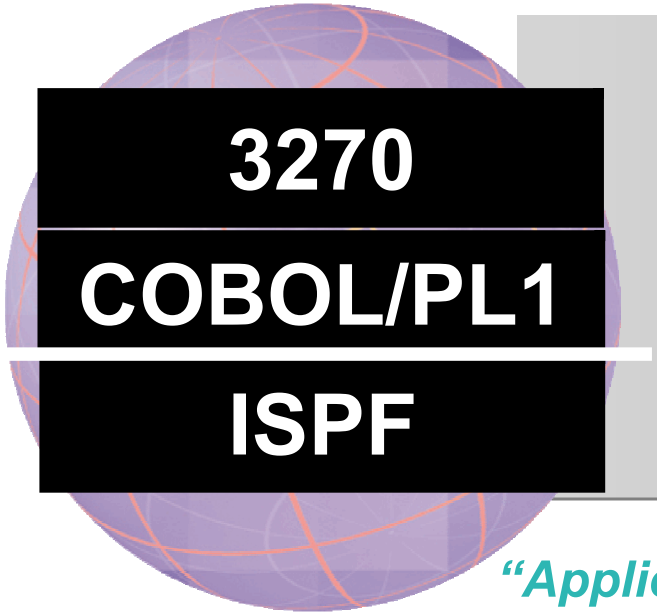

Tolerate (Re-Evaluate/ Reposition Asset)
Integrate (Maintain/ Evolve Asset)
Eliminate (Retire/ Consolidate Asset)
Migrate (Re-Engineer/ Modernize Asset)
Technical Quality
Poor
Low
High
Business Value

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.

- **COBOL : main key points**
    - Strong presence of COBOL vendors
    - 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.

Investment Challenges



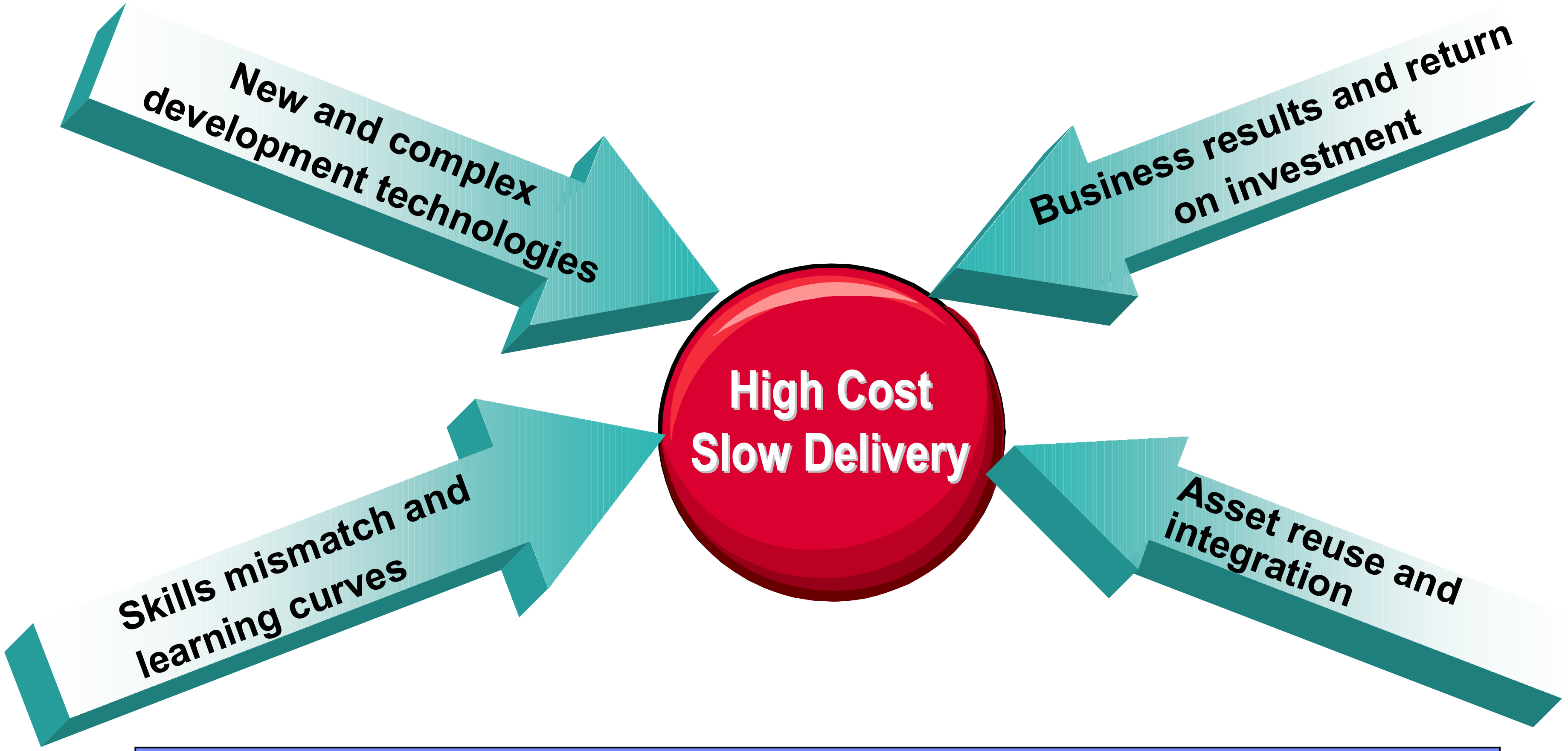
- **Many zSeries developers still:**
  - Focused on creating or enhancing 3270 applications
  - Using traditional, host-based development environment

"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

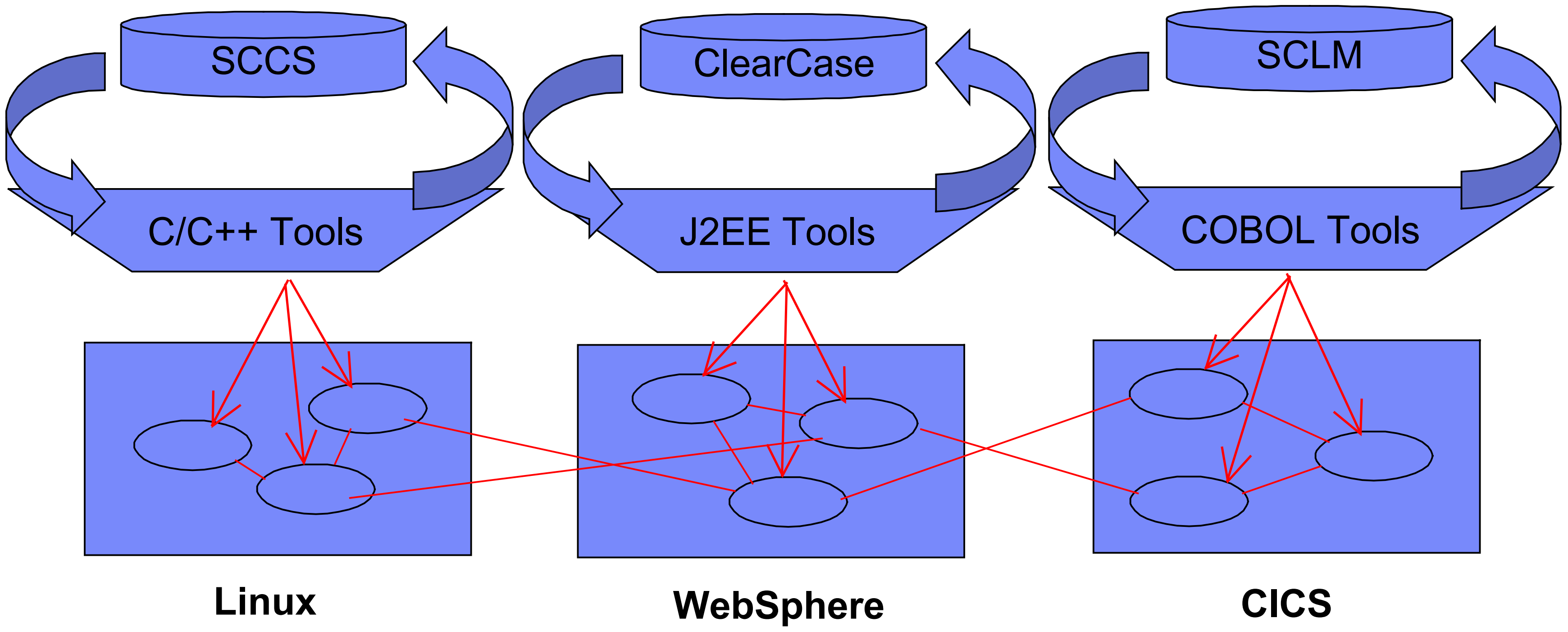

New and complex development technologies
Business results and return on investment
High Cost
Slow Delivery
Skills mismatch and learning curves
Asset reuse and integration

Issues: How do I?

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

Organizational Challenges

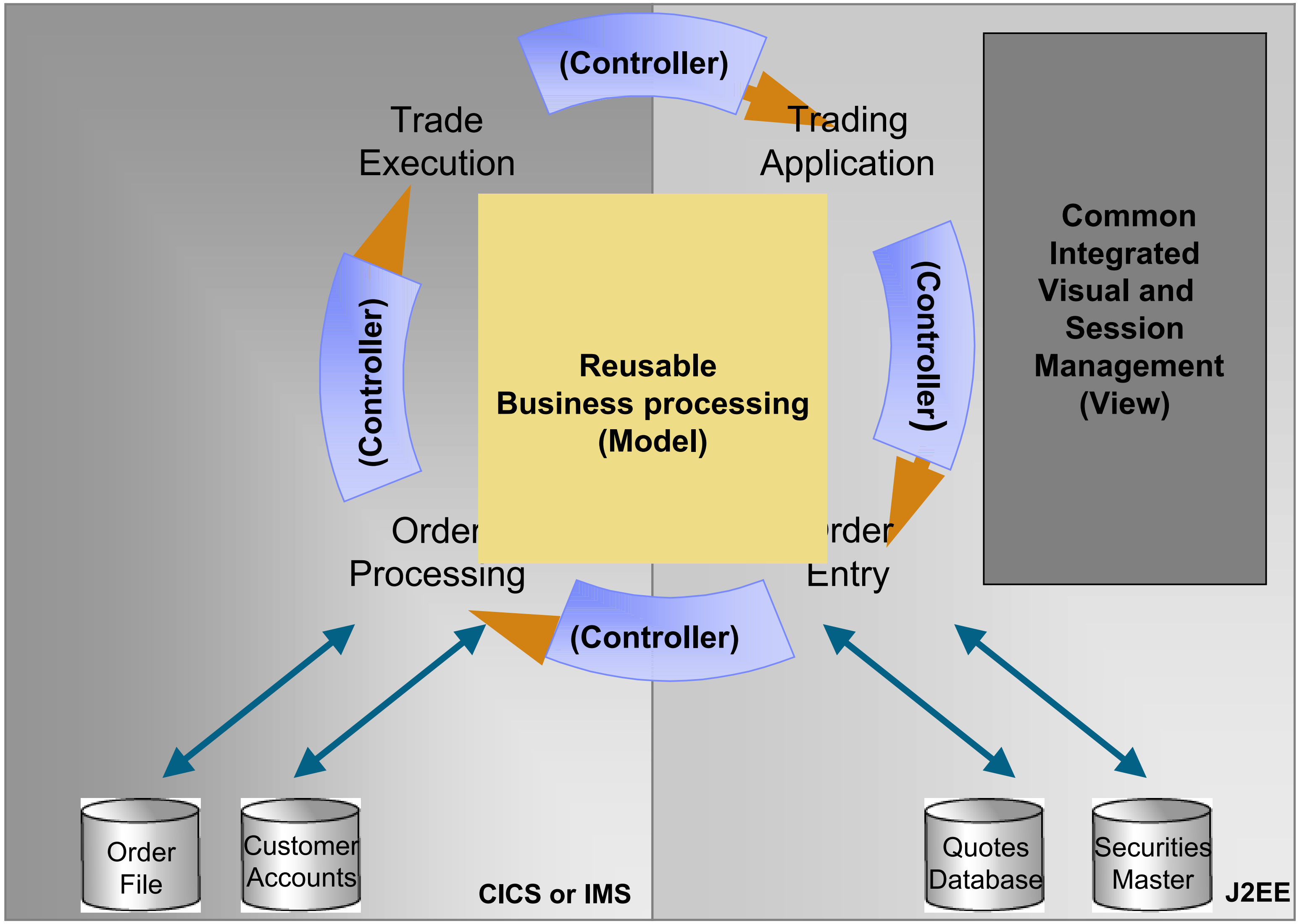
- Lack application components & skills sharing
- Ineffective / Uncoordinated development of integrated application



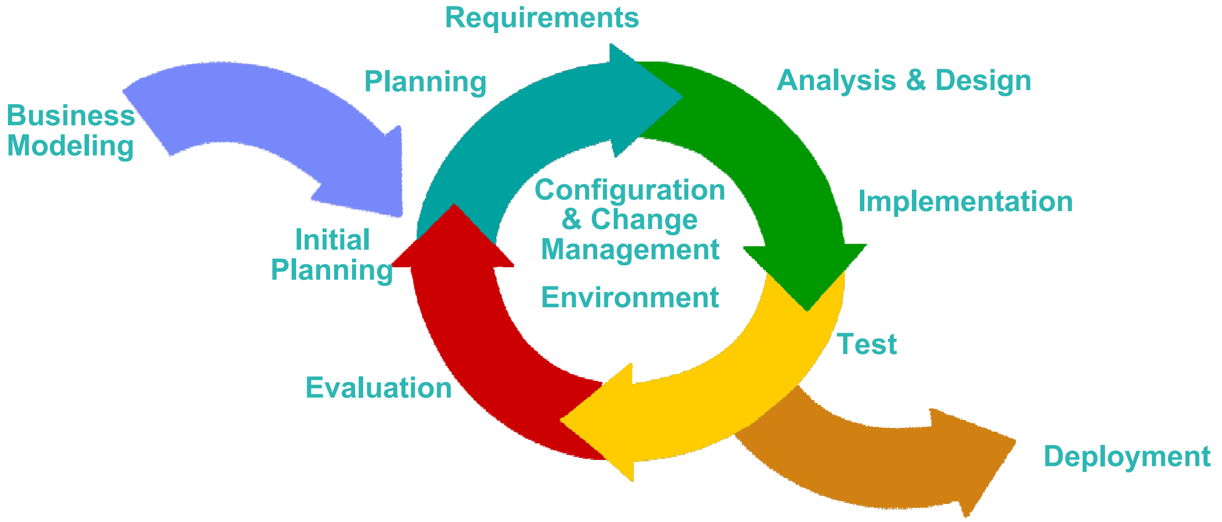
Issues: How do I?

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

Composite Workload Application Components



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

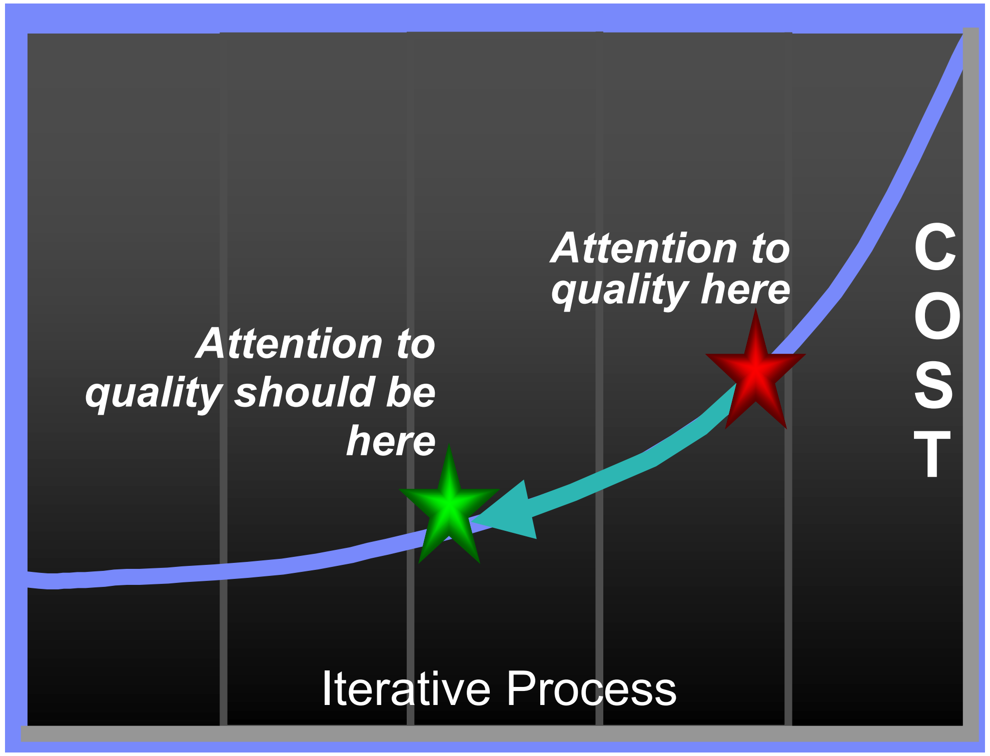


- Adopt a flexible process for both J2EE & traditional z/Series applications
- Tools integration across the lifecycle (Model and Discover, Develop and Assemble, & Deploy and Manage)
- Manage mixed workload requirements

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

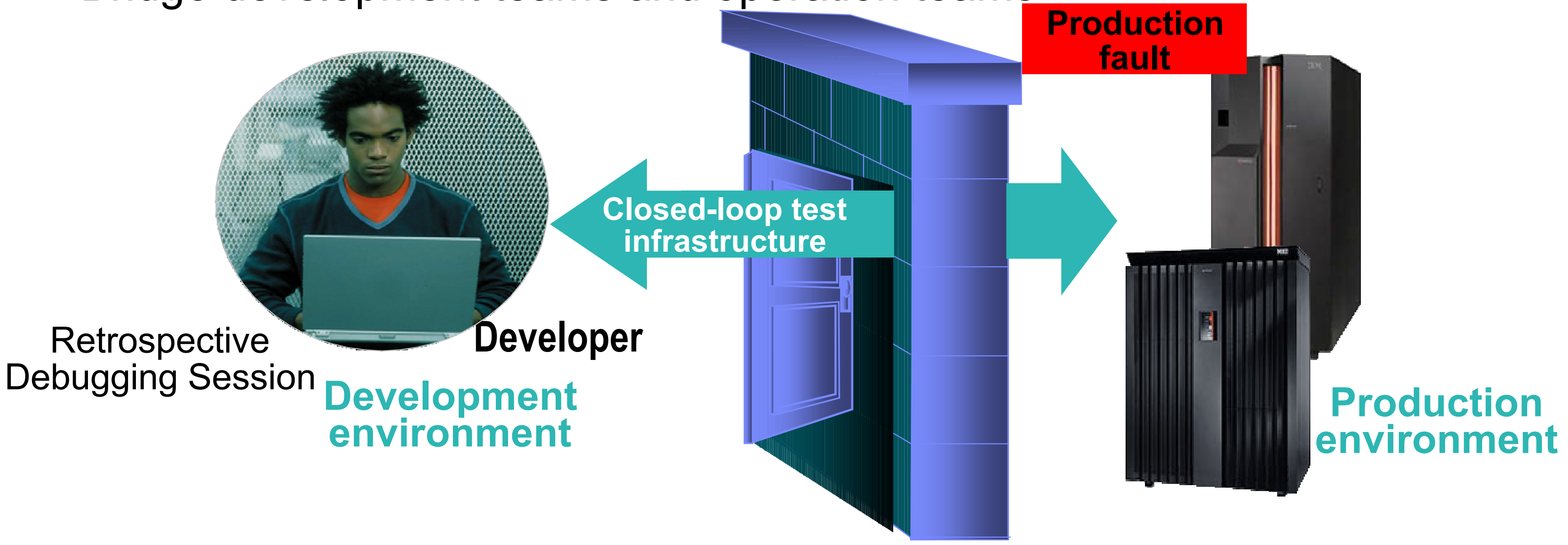


Issues: How do I?

- 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

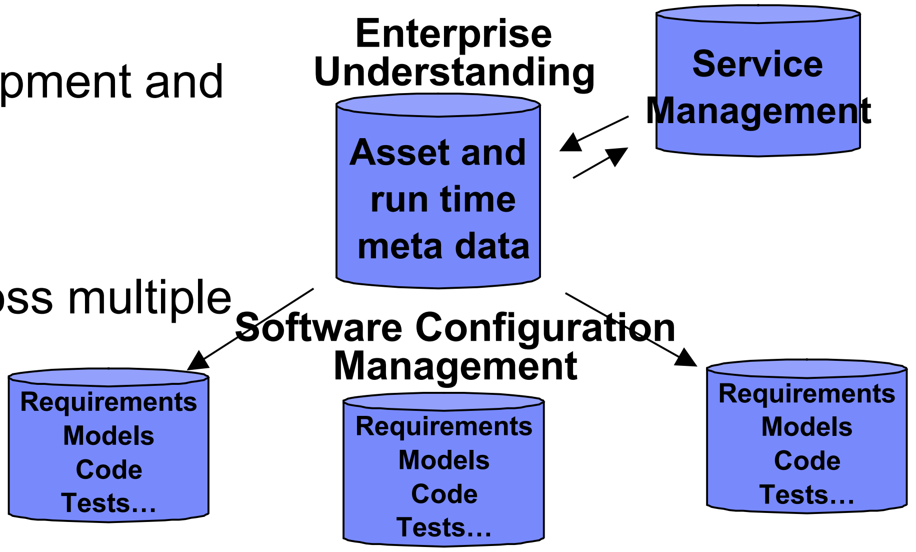


Issues: How do I?

- 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
Service Management
Asset and run time meta data
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

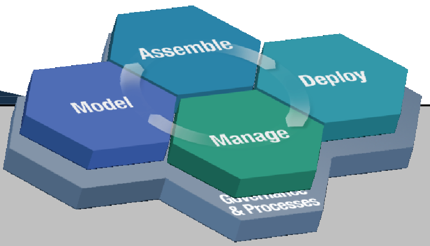
Issues: How do I?

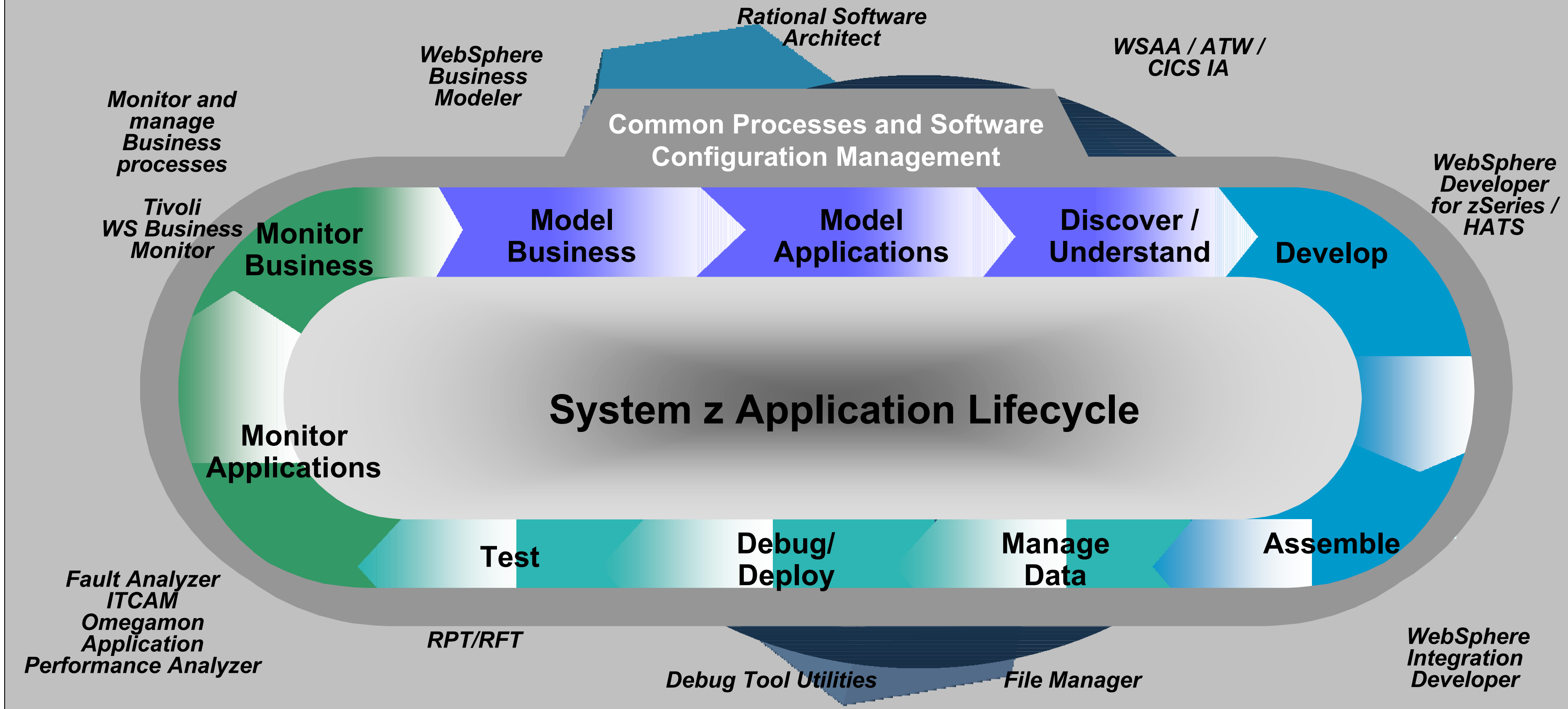
- Govern processes and enable reuse
- Track who is working on what
- Merge changes from multiple teams
- Support vastly increased numbers of artifacts across the lifecycle

Agenda

- **IT Market : trend & directions**
- **Service Oriented Architecture**
- **COBOL & Enterprise Application : Today**
- **SOA and the System z Application Lifecycle**

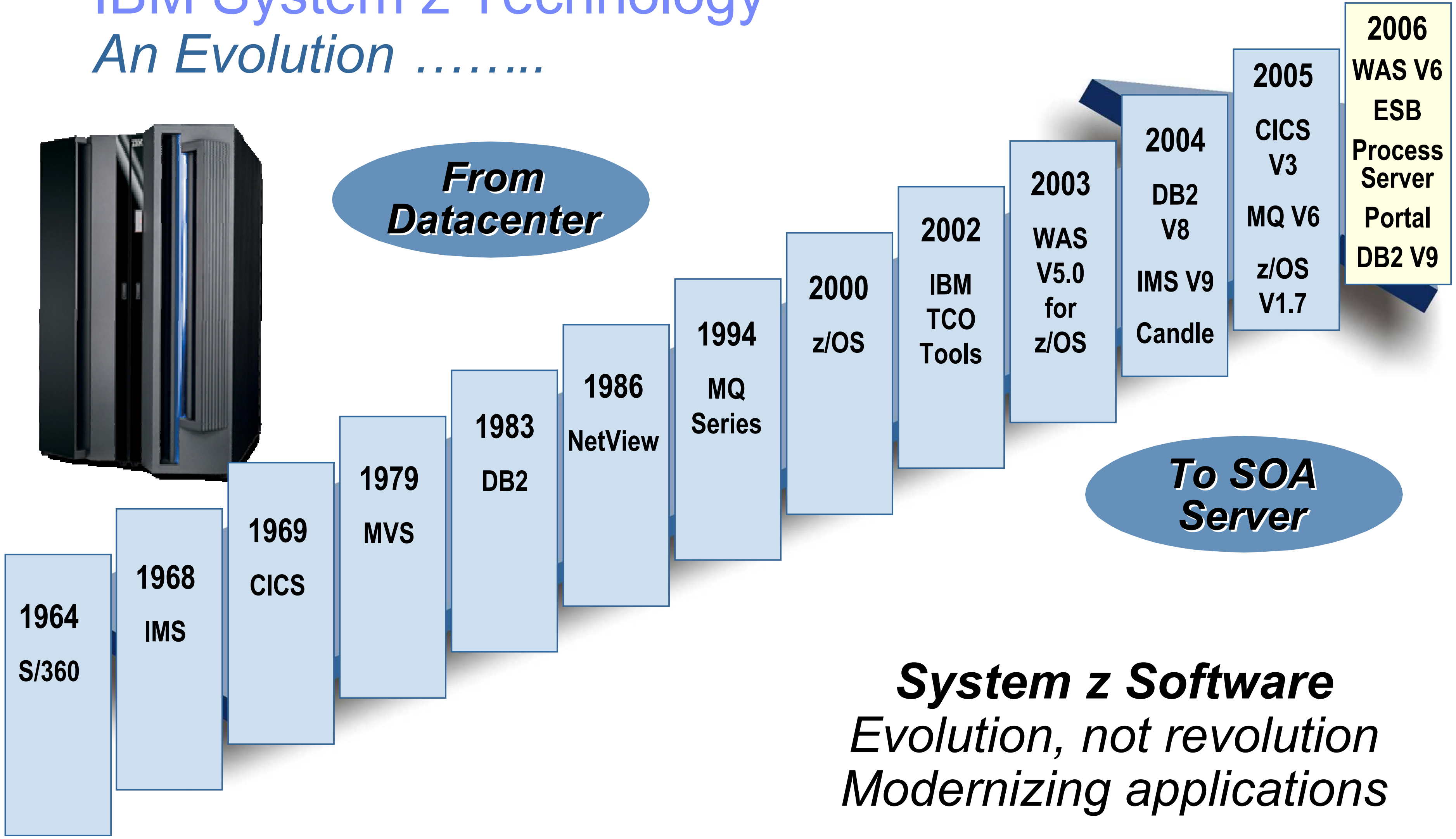
System z Application Lifecycle




Rational Software Architect
WebSphere Business Modeler
WSAA / ATW / CICS IA
Monitor and manage Business processes
Common Processes and Software Configuration Management
WebSphere Developer for zSeries / HATS
Tivoli WS Business Monitor
Monitor Business
Model Business
Model Applications
Discover / Understand
Develop
System z Application Lifecycle
Monitor Applications
Test
Debug/ Deploy
Manage Data
Assemble
Fault Analyzer
ITCAM
Omegamon
Application Performance Analyzer
RPT/RFT
Debug Tool Utilities
File Manager
WebSphere Integration Developer

IBM System z Technology

An Evolution ……..



System z Software

Evolution, not revolution

Modernizing applications

System z
Openness and Standards

Linux

UNIX ***SOA***

SAN ***Java***

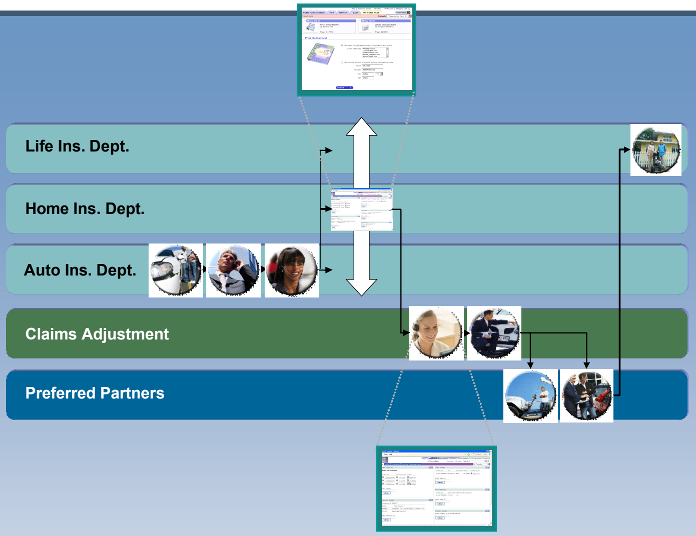
Web Services ***J2EE***

Grid & Autonomic Sys. Mgmt

Enabling a robust, flexible SOA runtime environment

While maximizing the value of existing assets

Fully SOA capable!



WebSphere Application Server V6

CICS Transaction Server V3.1

IMS Transaction and Database V9

#1 in market share for Application Server software

IBM WebSphere Application Server comes out on top

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

Tools to realize Enterprise Modernization

WebSphere Studio Asset Analyzer (WSAA)

- Application Understanding
- Impact Analysis

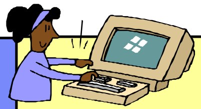
Enterprise-wide app discovery and insight; find dependencies across applications and lines of business

Architects, project leaders, DBAs, developers, system programmers

Asset Transformation Workbench (ATW)

- Application Analysis
- Business Rule Management
- Components for reuse

Project-level workbench for deep application analysis and transformation

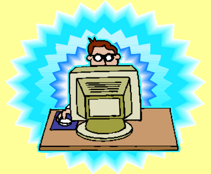


Architects, project leaders, analysts, developers

WebSphere Developer for zSeries (WDz)

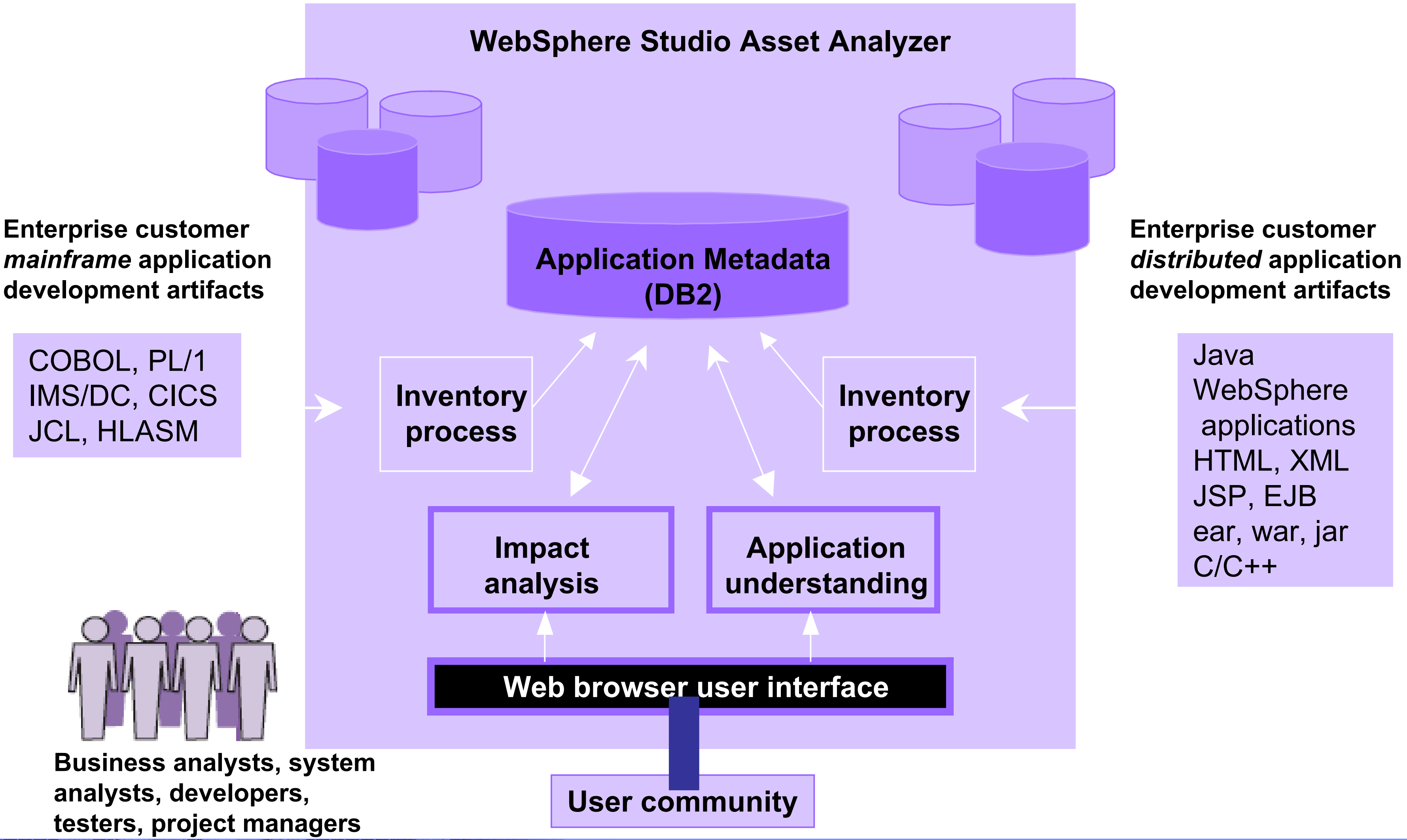
- Traditional Development
- Web Development
- Services Development

Common IDE for COBOL, PL/I, J2EE and Web services development

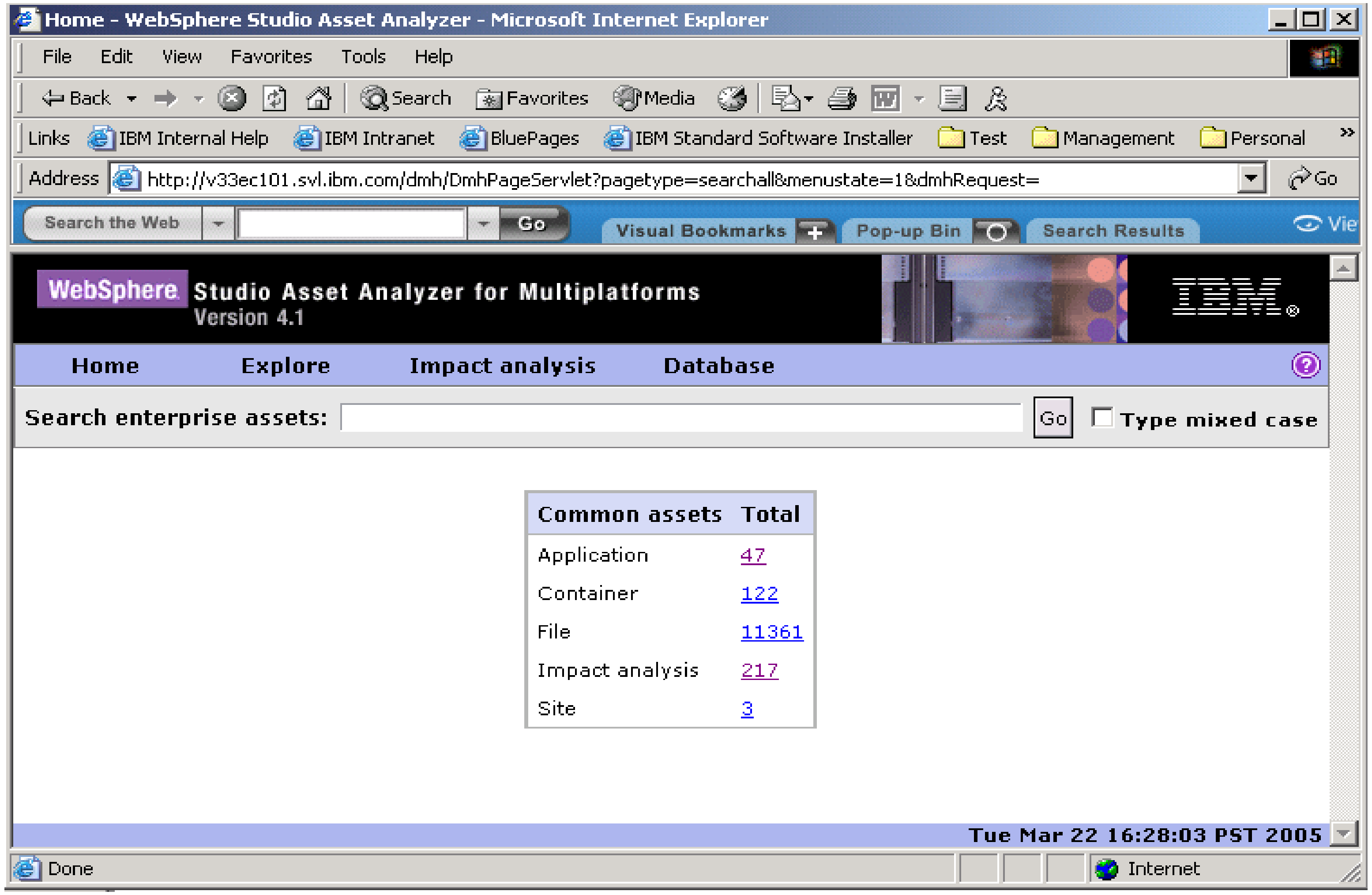


Architects, Developers

WebSphere Studio Asset Analyzer

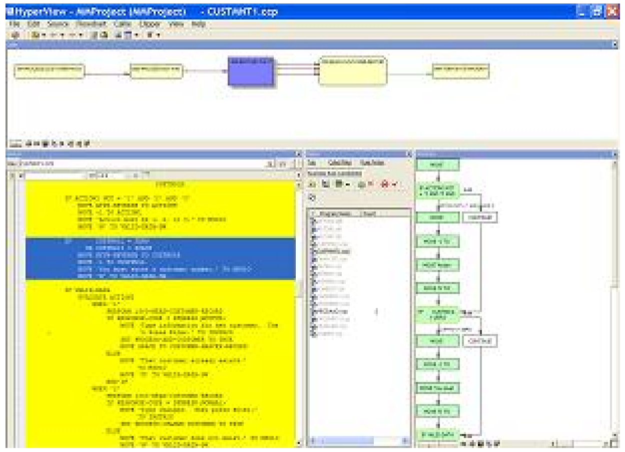

WebSphere Studio Asset Analyzer
Enterprise customer *mainframe* application development artifacts
COBOL, PL/1
IMS/DC, CICS
JCL, HLASM
Application Metadata (DB2)
Inventory process
Inventory process
Enterprise customer *distributed* application development artifacts
Java
WebSphere applications
HTML, XML
JSP, EJB
ear, war, jar
C/C++
Impact analysis
Application understanding
Web browser user interface
User community
Business analysts, system analysts, developers, testers, project managers

Searching for Application Assets

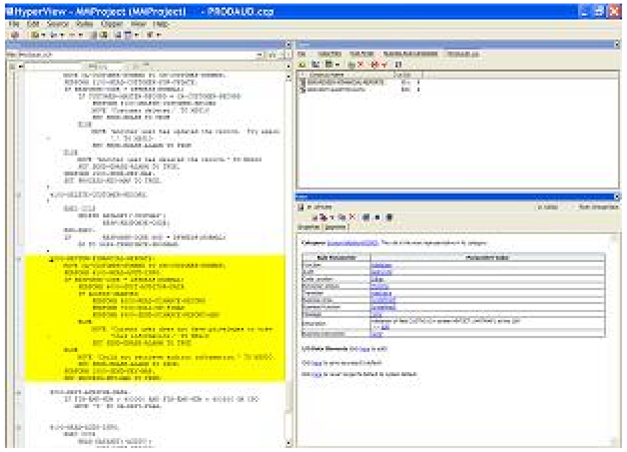

Home - WebSphere Studio Asset Analyzer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links IBM Internal Help IBM Intranet BluePages IBM Standard Software Installer Test Management Personal
Address http://v33ec101.svl.ibm.com/dmh/DmhPageServlet?pagetype=searchall&menustate=1&dmhRequest=
Go
Search the Web Go Visual Bookmarks Pop-up Bin Search Results
WebSphere Studio Asset Analyzer for Multiplatforms
Version 4.1
Home Explore Impact analysis Database
Search enterprise assets: Go Type mixed case
Tue Mar 22 16:28:03 PST 2005
Done Internet

Common assets	Total
Application	47
Container	122
File	11361
Impact analysis	217
Site	3

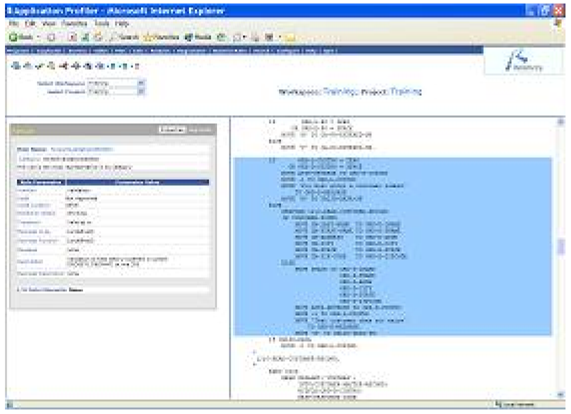
Asset Transformation Workbench



Business rule discovery

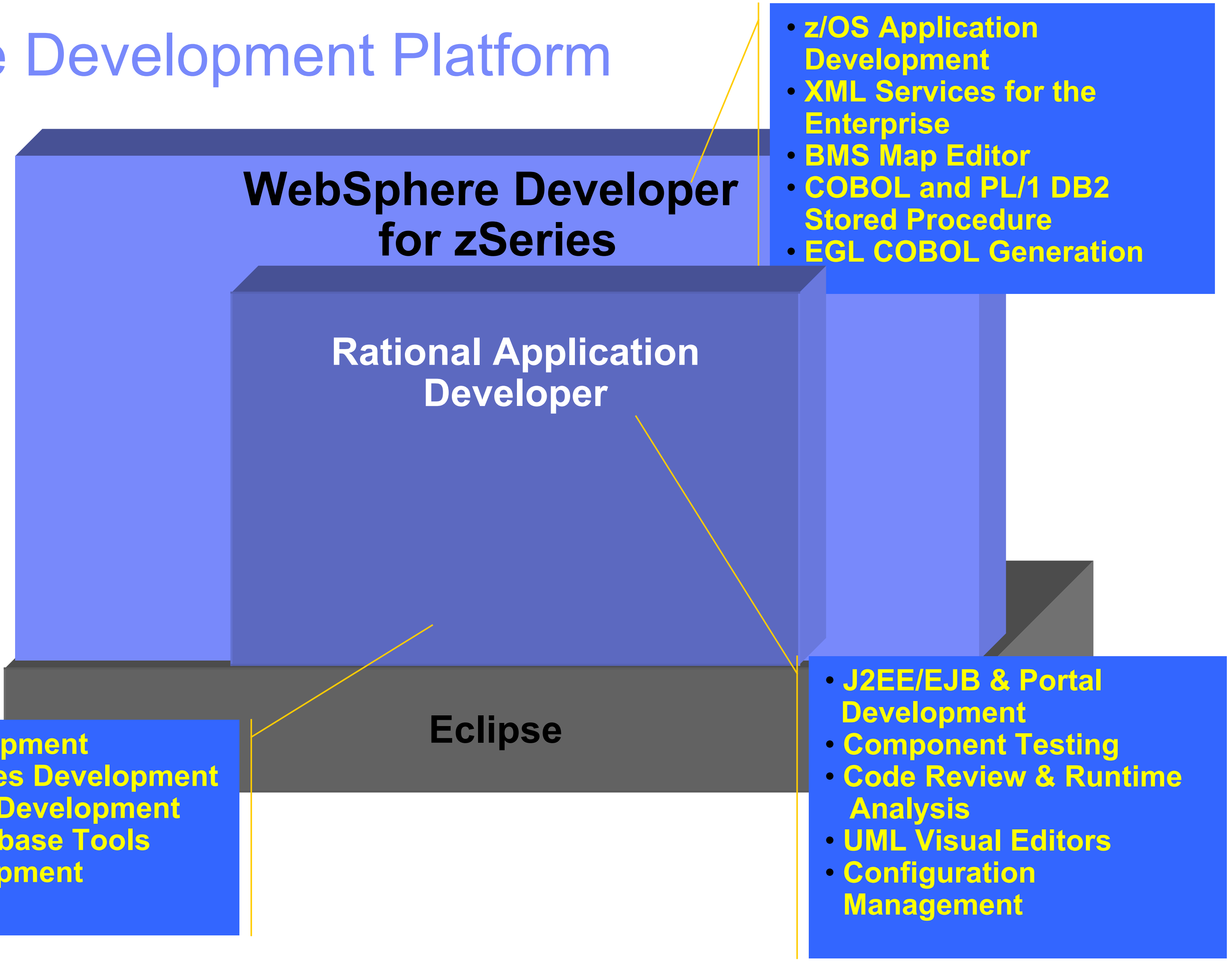


Business rule management



Integration with Application Profiler

Software Development Platform

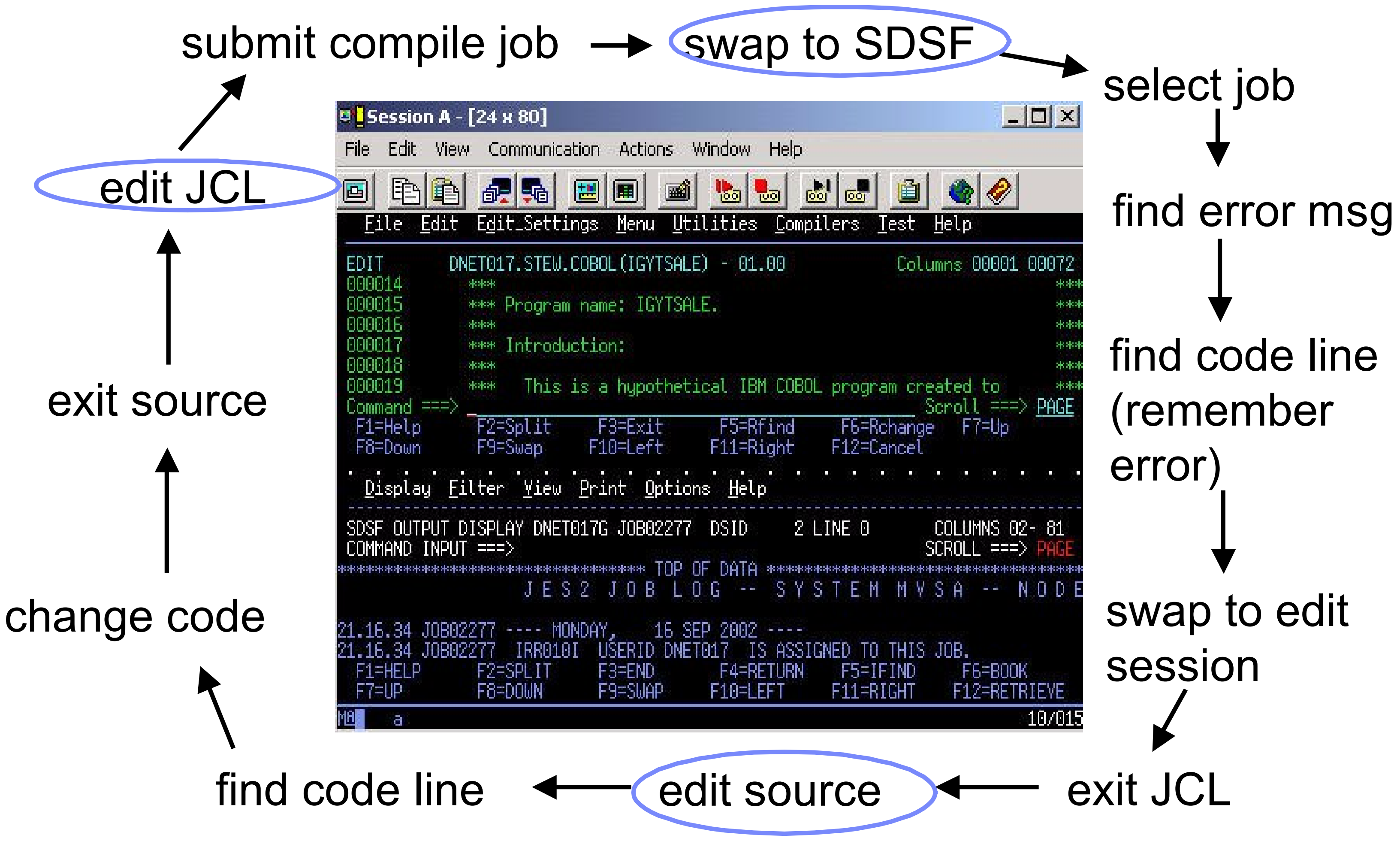

WebSphere Developer for zSeries
Rational Application Developer
Eclipse

- z/OS Application Development
- XML Services for the Enterprise
- BMS Map Editor
- COBOL and PL/1 DB2 Stored Procedure
- EGL COBOL Generation

- J2EE/EJB & Portal Development
- Component Testing
- Code Review & Runtime Analysis
- UML Visual Editors
- Configuration Management

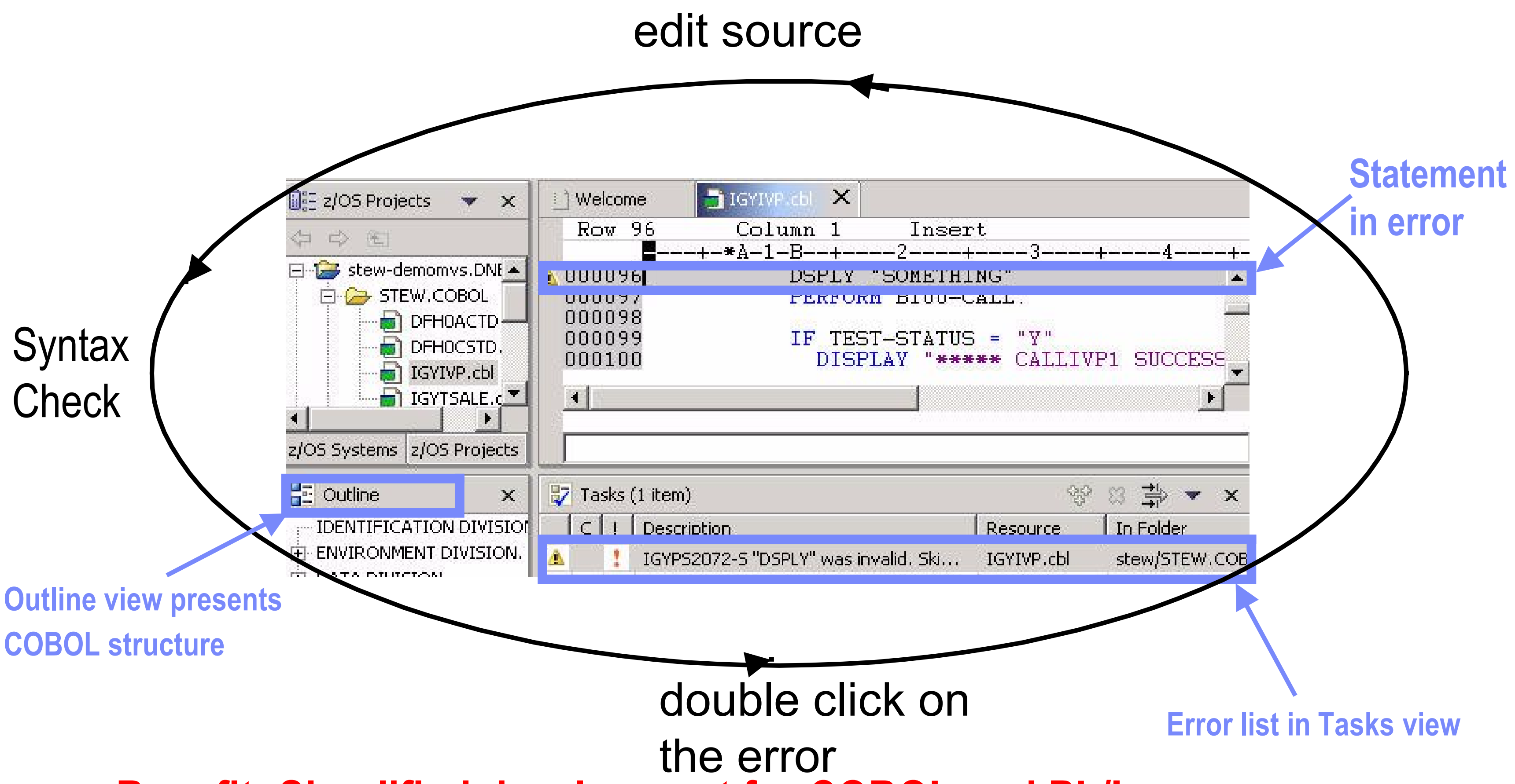
- Web Development
- Web Services Development
- Rich Client Development
- XML & Database Tools
- 4GL Development

ISPF based Development



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DNET017.STEW.COBOL(IGYTSALE) - 01.00            Columns 00001 00072
000014      ***                                                            ***
000015      *** Program name: IGYTSALE.                                    ***
000016      ***                                                            ***
000017      *** Introduction:                                              ***
000018      ***                                                            ***
000019      ***   This is a hypothetical IBM COBOL program created to      ***
Command ===> _                                                Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY DNET017G JOB02277  DSID     2 LINE 0       COLUMNS 02- 81
COMMAND INPUT ===>                                            SCROLL ===> PAGE
********************************* TOP OF DATA **********************************
                   J E S 2  J O B  L O G  --  S Y S T E M  M V S A  --  N O D E

21.16.34 JOB02277 ---- MONDAY,    16 SEP 2002 ----
21.16.34 JOB02277  IRR010I  USERID DNET017  IS ASSIGNED TO THIS JOB.
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA    a                                                                  10/015
```

WebSphere Development



```
stew-demomvs.DNE
  STEW.COBOL
    DFH0ACTD
    DFH0CSTD.
    IGYIVP.cbl
    IGYTSALE.c
```

```
 Row 96          Column 1        Insert
 ----+-*A-1-B--+----2----+----3----+----4----+-
000096               DSPLY  "SOMETHING"
000097               PERFORM B100-CALL.
000098
000099               IF TEST-STATUS = "Y"
000100                 DISPLAY "***** CALLIVP1 SUCCESS
```

```
IDENTIFICATION DIVISION
ENVIRONMENT DIVISION.
DATA DIVISION
```

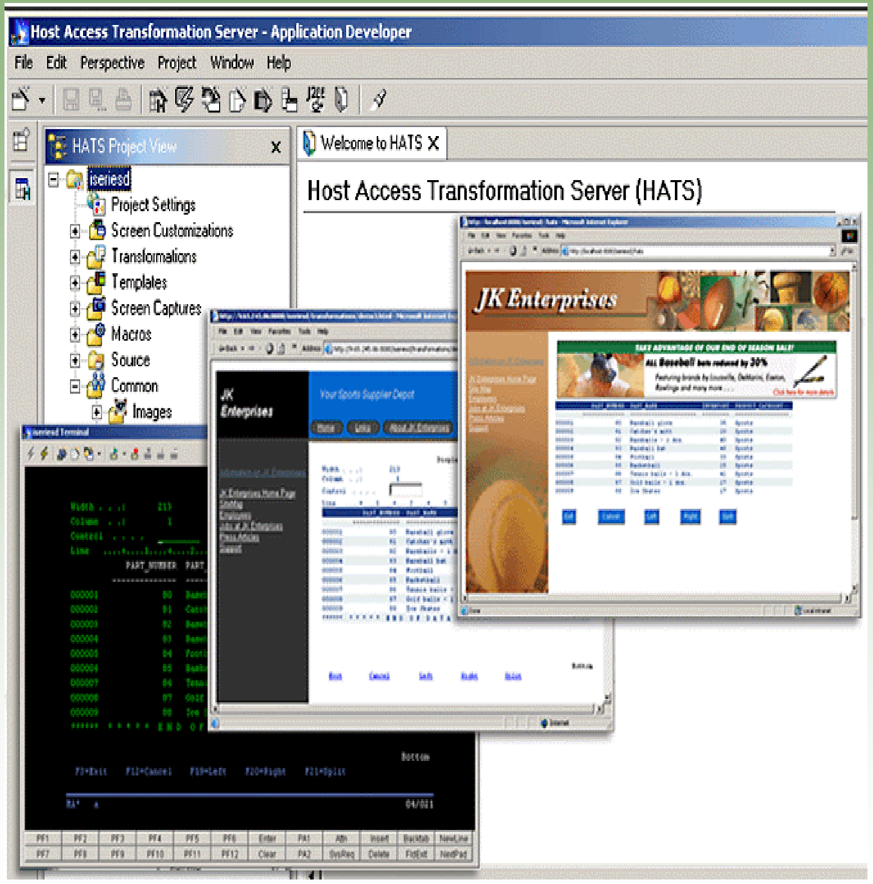
C	!	Description	Resource	In Folder
	!	IGYPS2072-S "DSPLY" was invalid. Ski...	IGYIVP.cbl	stew/STEW.COB

Benefit: Simplified development for COBOL and PL/I on a common development environment

WebSphere Host Access Transformation Server

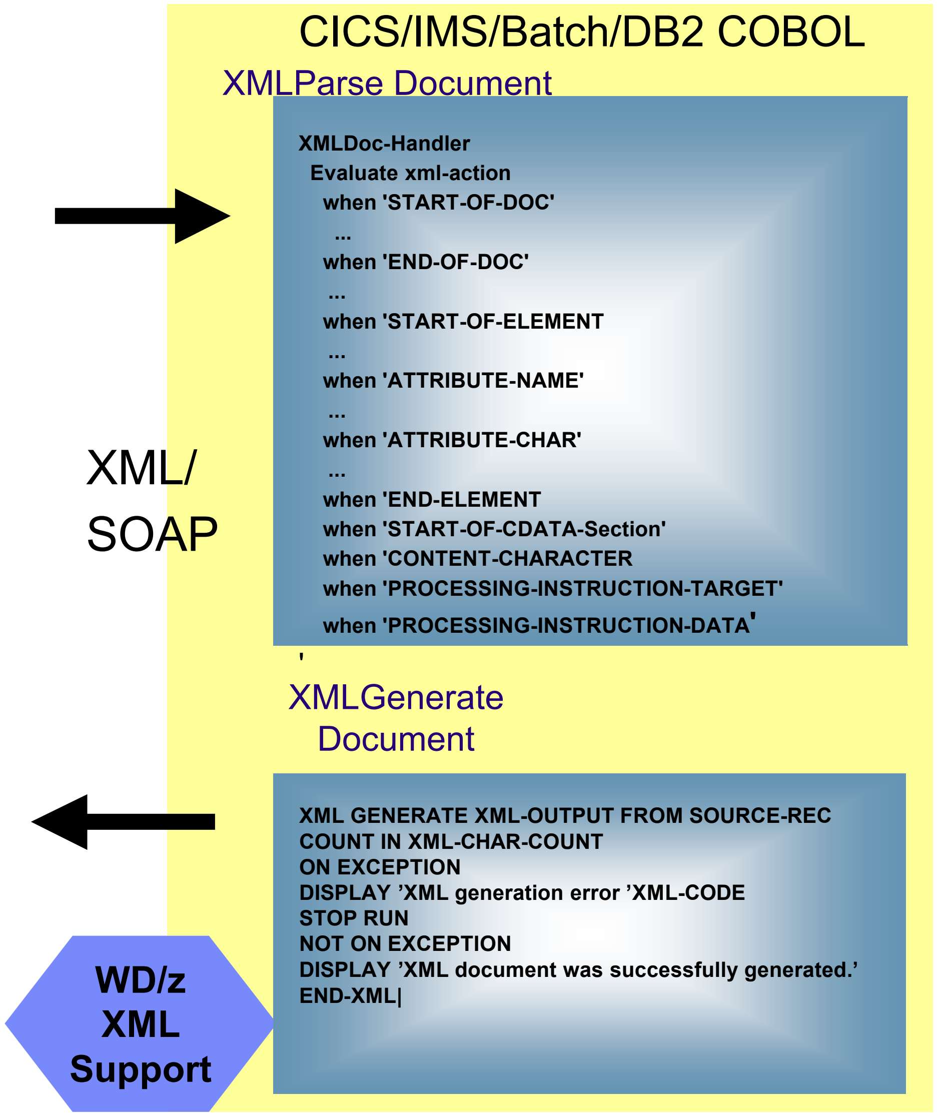
Extend business processing through existing interfaces

- 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



Using Enterprise COBOL to service-enable z/OS

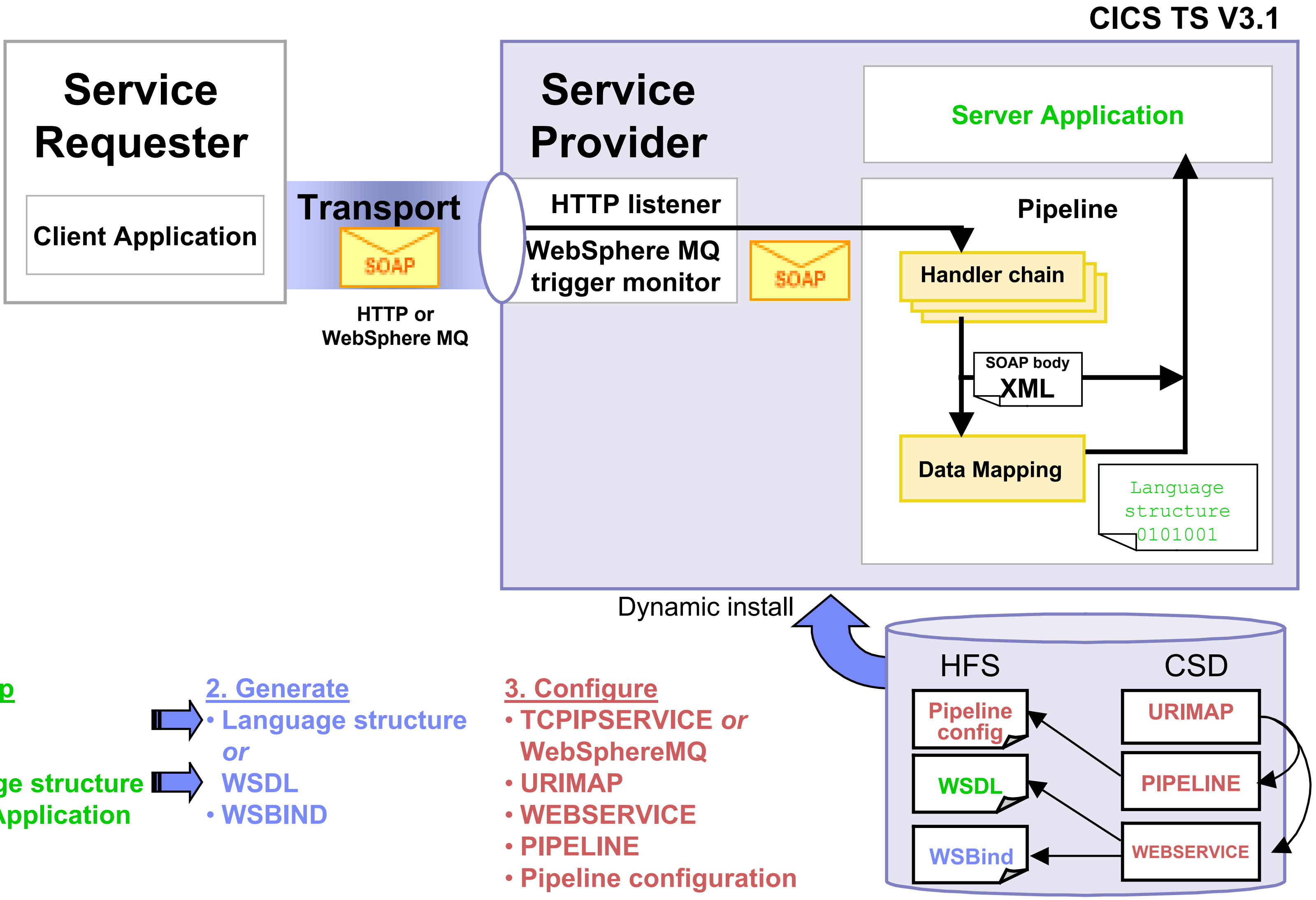
- What's the latest…
  - XML Language based generation from COBOL data structure
  - WebSphere EJB support
  - DB2 V8 preprocessor
  - CICS preprocessor
- High speed XML Sax based parsing
- Object Oriented Support for Java COBOL Interoperability
- Unicode support
- Similar XML parsing support available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



```
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 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|
```

CICS come Web service provider



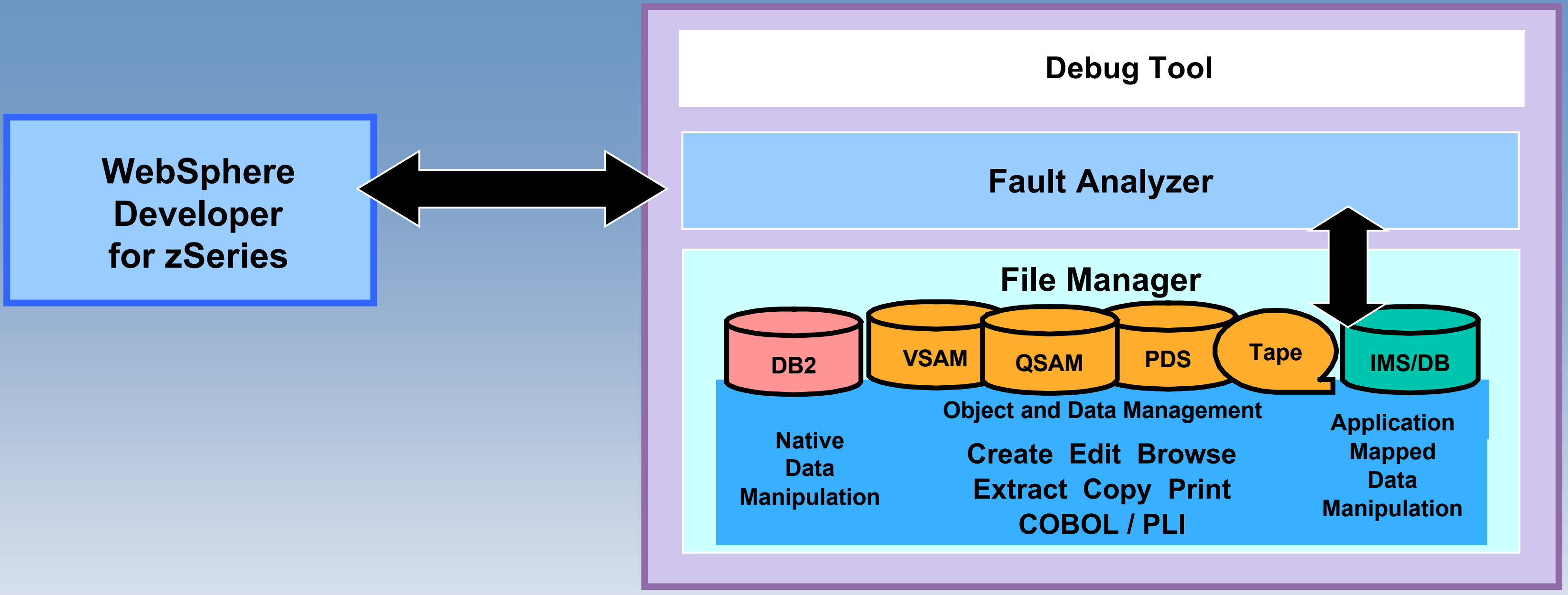
1. Develop
- WSDL
  or
  Language structure
- Server Application

2. Generate
- Language structure
  or
  WSDL
- WSBIND

3. Configure
- TCPIPSERVICE *or* WebSphereMQ
- URIMAP
- WEBSERVICE
- PIPELINE
- Pipeline configuration

Test and Problem Determination

Integration speeds time to market



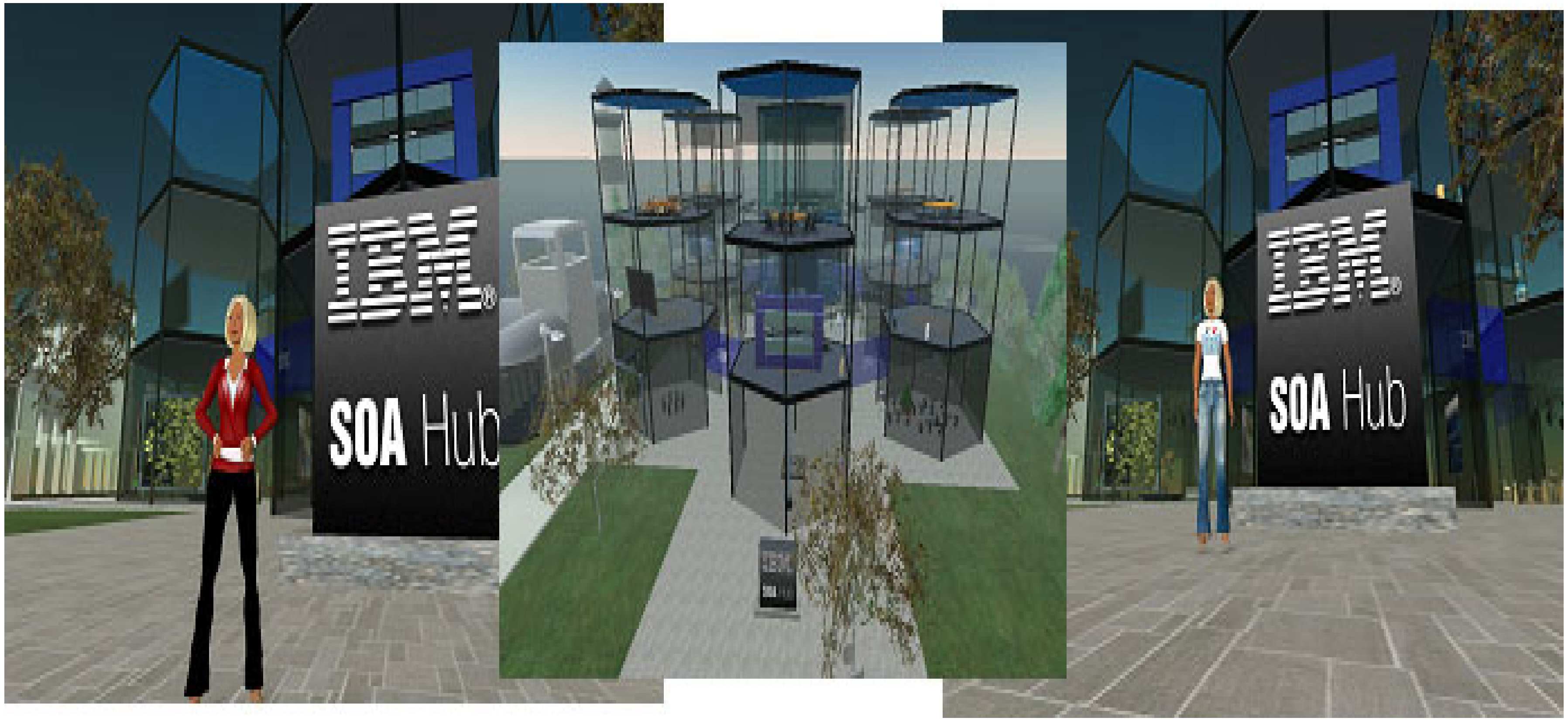
Benefits:

- **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**

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.

Second Life ?



SOA the next level : Web 2.0

Bridge between Web and Enterprise SOA

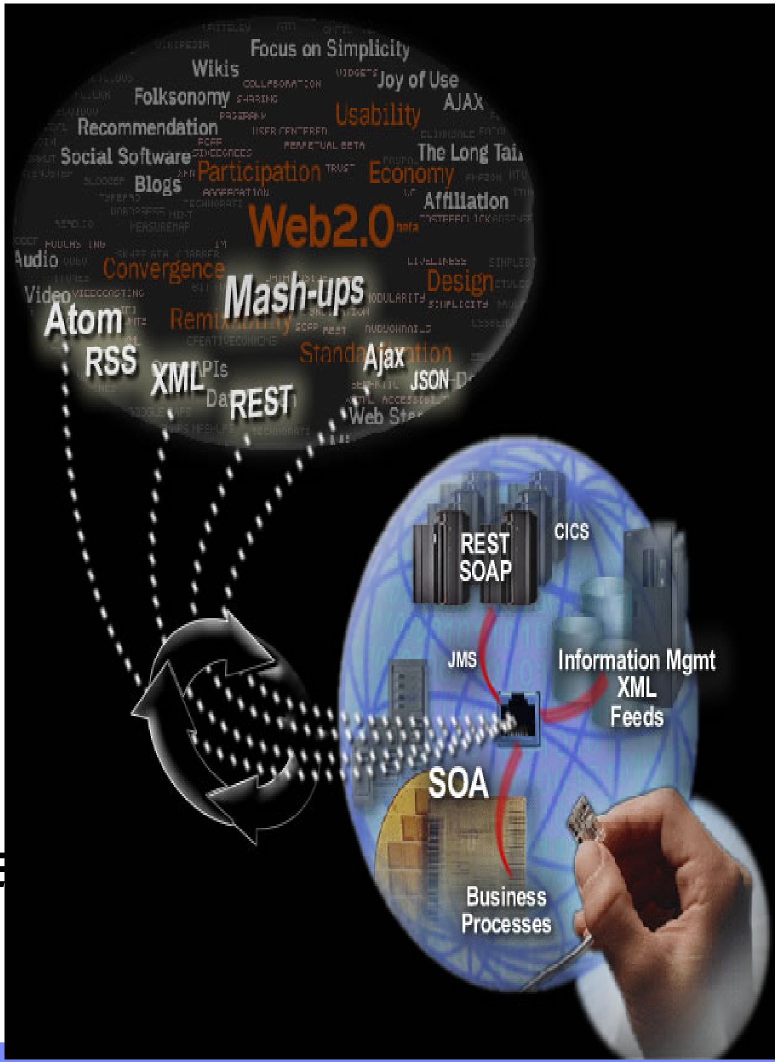
Expand SOA with Emerging Web 2.0 Technologies

- **The Web is about content :**
  - **Social Computing**
  - **Mash-Ups**
  - **Feeds**
  - **Rich User Experience (XML, AJAX, etc..)**

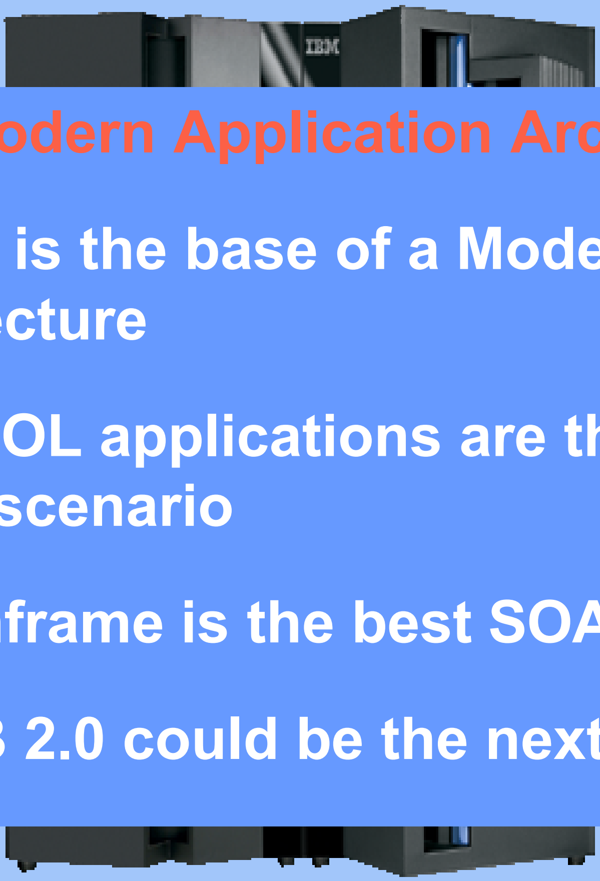
Continue Industry Web 2.0 Technology Collaborations

See video Web 2.0 on YouTube

http://www.youtube.com/watch?v=6gmP4nk0EOE



Conclusion



Modern Application Architecture

- ❑ **SOA is the base of a Modern Application Architecture**
- ❑ **COBOL applications are the key components in this scenario**
- ❑ **Mainframe is the best SOA Server**
- ❑ **WEB 2.0 could be the next step**

IBM Software Group

Rational Business Developer, EGL, and Enterprise Transformation

Rational. software

Roberto Pozzi

Advisory IT Specialist - Rational Technical Sales

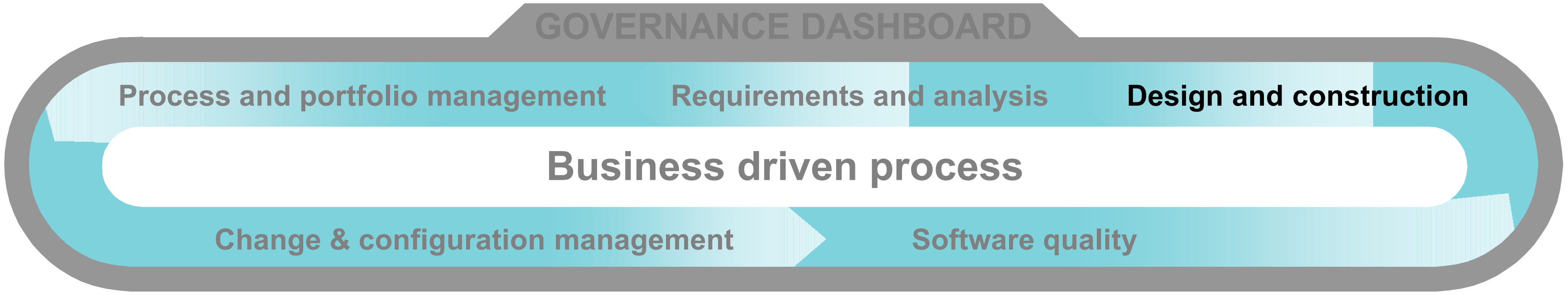
roberto_pozzi@it.ibm.com

ON DEMAND BUSINESS

Topics

- Why EGL
- EGL Overview
- EGL and Application Transformation
- EGL … What's Next
- Customer stories
- Summary

Key challenges of Design and Construction


GOVERNANCE DASHBOARD
Process and portfolio management
Requirements and analysis
Design and construction
Business driven process
Change & configuration management
Software quality

Existing Applications

- Costly to maintain
- Monolithic
- Hard to reuse in new ways

Skills

- Skills silos
- Skills mismatch
- Erosion of legacy platforms skills

Platforms / Middleware

- Proliferation
- Coexistence
- Complexity

High Costs **Slow Response**

Compromise

Why Enterprise Generation Language?

- Developing software is slow, repetitive and error prone
  - Complex low level coding bogs down programmers
- Many developers skills are “business oriented”
  - Know the business…been building business applications for years
  - RPG, COBOL, PL/I, 4GL, Visual Basic
  - … but new applications require Java/J2EE skills
- Re-training may not be an option
  - High costs
  - Business pressure may not afford time
  - Results may be sub-optimal
    - Some may not make it
    - End up with poorly written applications
- Many “legacy developers” retiring
  - … but new hires don’t know existing environments (CICS, …)
- **Solve Business problems, not technology problems**

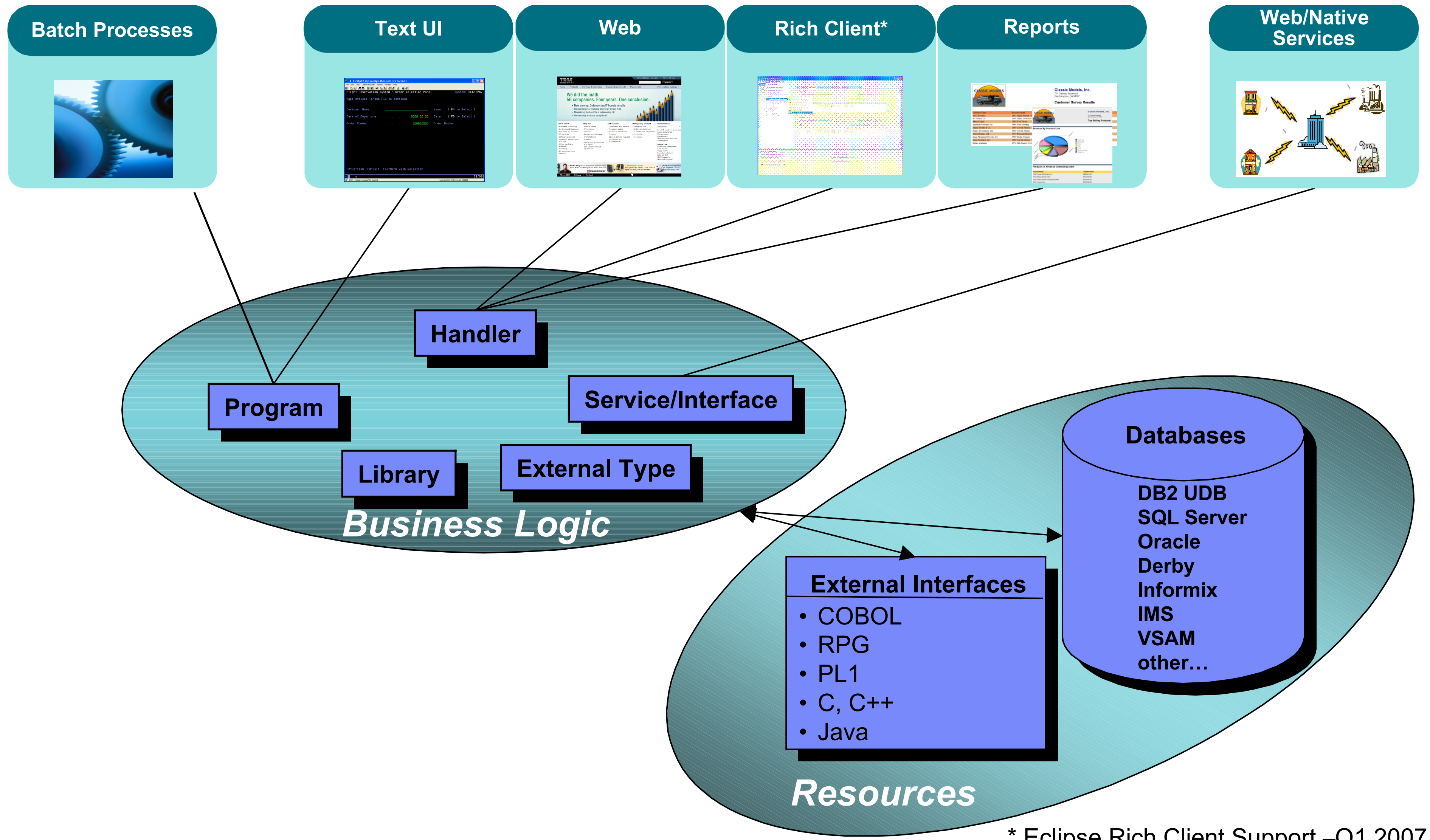
EGL Design Points

- Decouple application specification from runtimes
- Immediately useable by developers of any background
- Hide technical complexity
- Support emerging standards and technologies
- Guarantee optimal (native) deployment to any platform
  - New and traditional
- Ensure easy inter-operability with legacy
- Ensure productivity without compromising flexibility
  - Language simplicity
  - Language robustness
- Provide agile, iterative development

Topics

- Why EGL
- EGL Overview
- EGL and Application Transformation
- EGL … What's Next
- Customer stories
- Summary

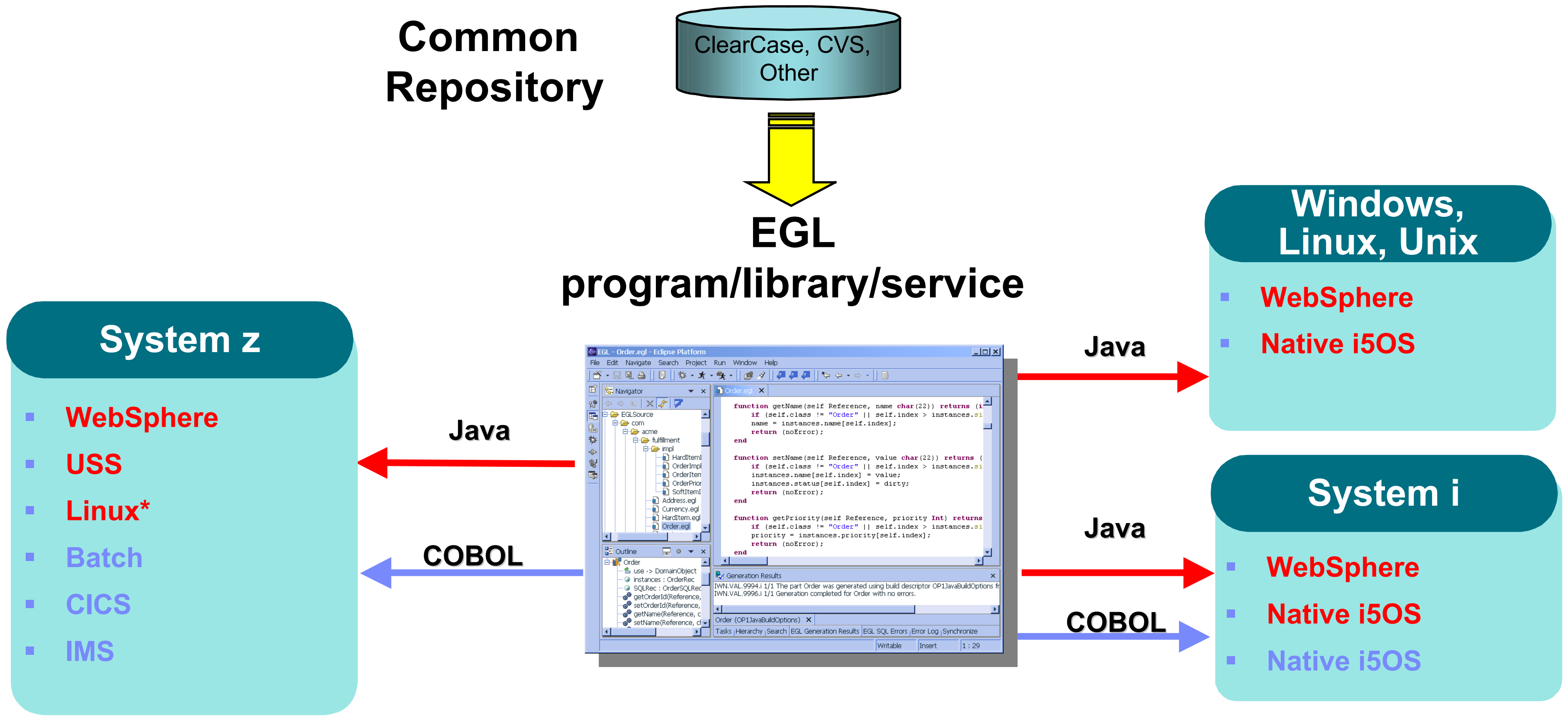
EGL…End-to-End



* Eclipse Rich Client Support –Q1 2007

EGL Platform Flexibility

Code once, deploy anywhere



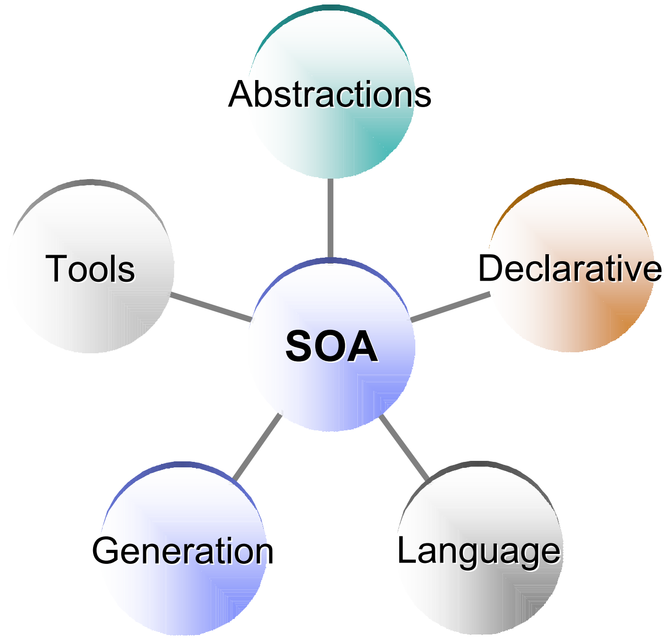
```
function getName(self Reference, name char(22)) returns (i
    if (self.class != "Order" || self.index > instances.si
    name = instances.name[self.index];
    return (noError);
end

function setName(self Reference, value char(22)) returns (
    if (self.class != "Order" || self.index > instances.si
    instances.name[self.index] = value;
    instances.status[self.index] = dirty;
    return (noError);
end

function getPriority(self Reference, priority Int) returns
    if (self.class != "Order" || self.index > instances.si
    priority = instances.priority[self.index];
    return (noError);
end
```

Generation Results

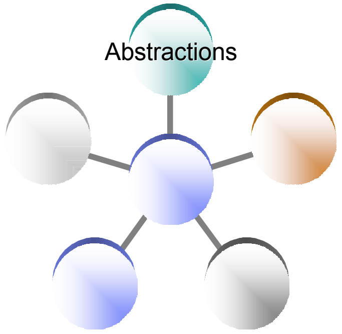
```
IWN.VAL.9994.i 1/1 The part Order was generated using build descriptor OP1JavaBuildOptions fr
IWN.VAL.9996.i 1/1 Generation completed for Order with no errors.
```

* Planned for 2007

EGL Philosophy

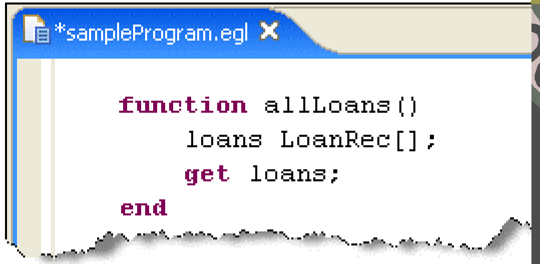


A productive, robust environment to develop business components and applications for all key business computing environments.



The power of abstractions

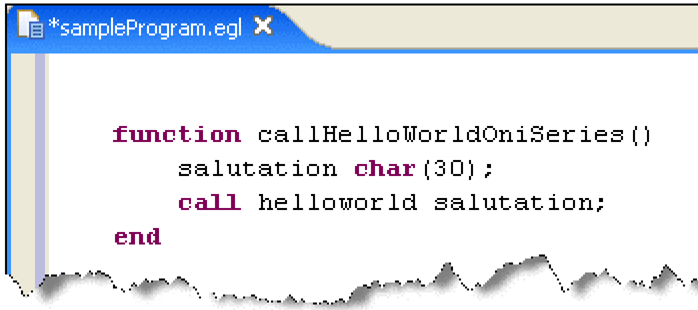
- Data access:
  - "Records" provide access to:
    - SQL, Indexed, Relative, Serial, DL/I, MQ, Service data
  - Common Verbs for data access (**Get, Add, Replace, Delete**)
  - Allows complete access to SQL statement if needed
  - Common Error Handling



*sampleProgram.egl

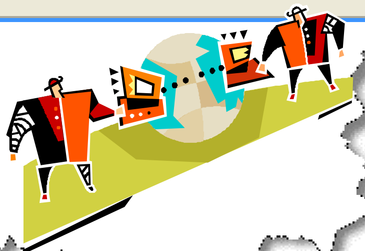
```
function allLoans()
    loans LoanRec[];
    get loans;
end
```

- Remote Invocation
  - Call COBOL, RPG, C, Java
  - Linkage information separated from code
  - Data mapping, protocol invocation all resolved at runtime, NO code necessary!



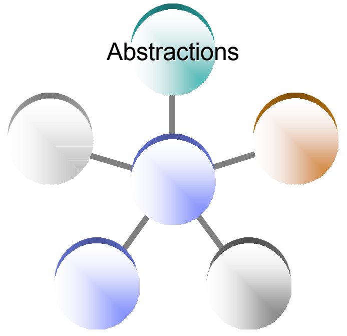
*sampleProgram.egl

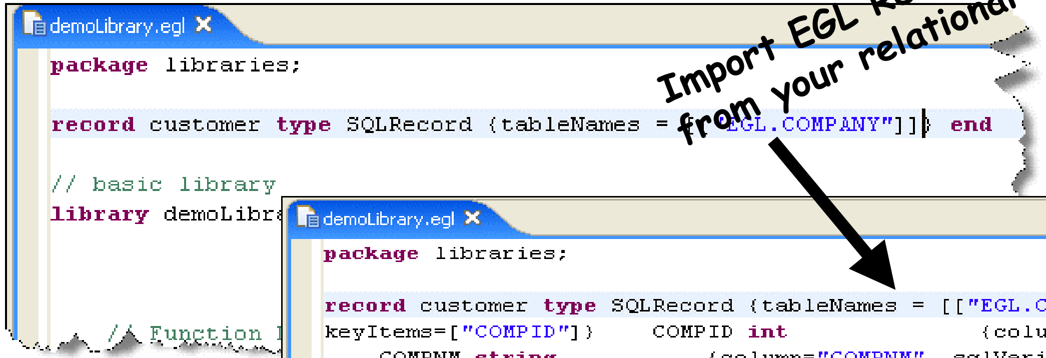
```
function callHelloWorldOniSeries()
    salutation char(30);
    call helloworld salutation;
end
```



The power of abstractions

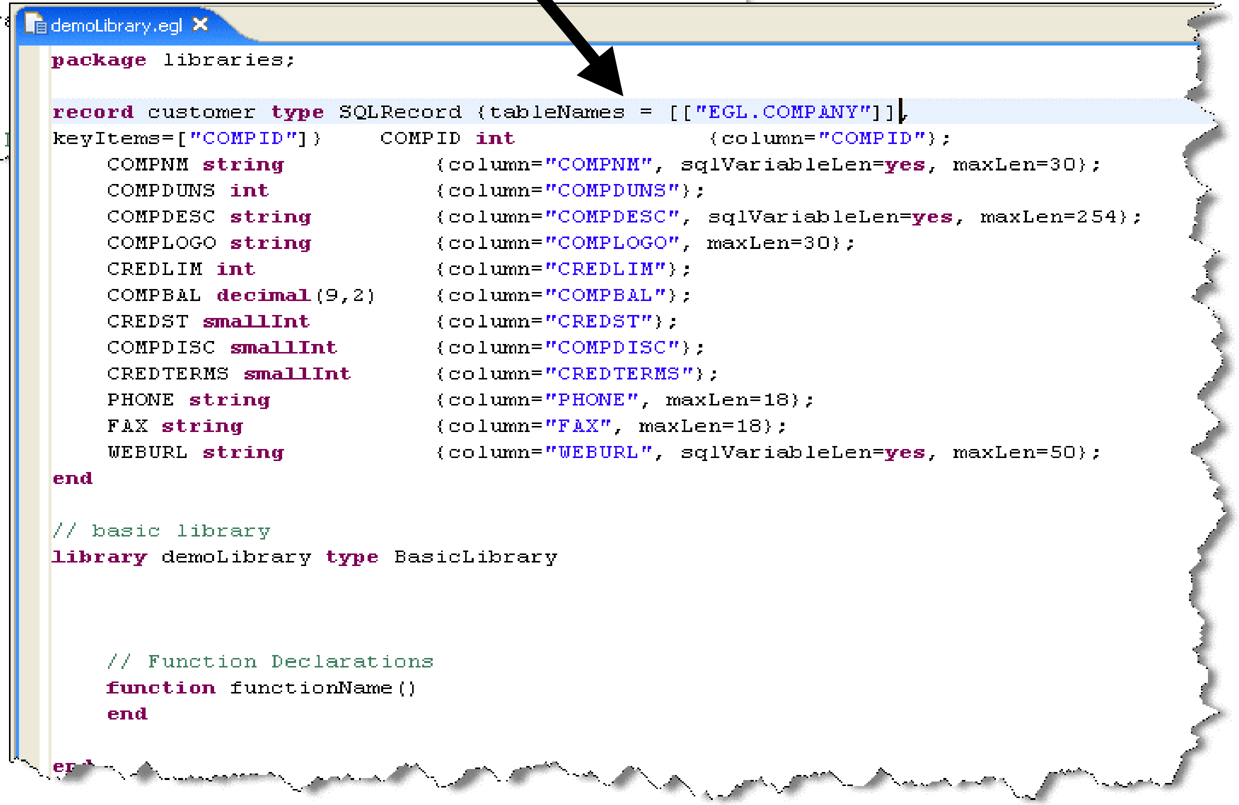
Data Driven Development





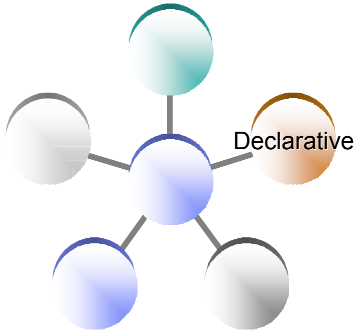
```
package libraries;

record customer type SQLRecord {tableNames = [["EGL.COMPANY"]]} end

// basic library
library demoLibr

   // Function
```

Import EGL Record definitions from your relational database

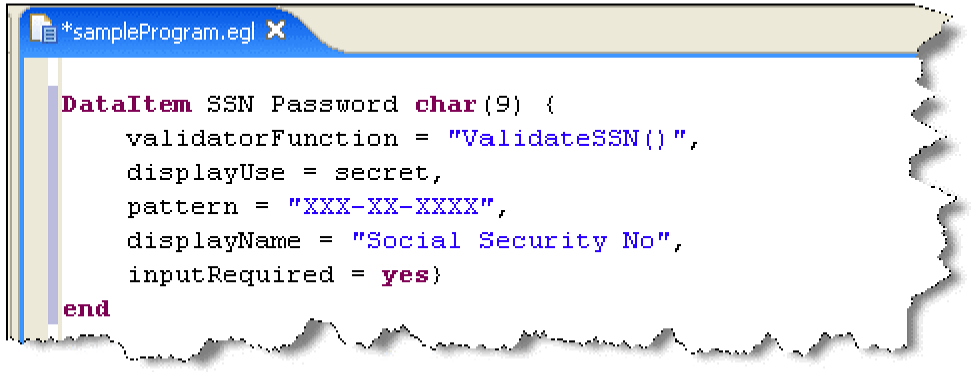


```
package libraries;

record customer type SQLRecord {tableNames = [["EGL.COMPANY"]],
keyItems=["COMPID"]}     COMPID int              {column="COMPID"};
    COMPNM string            {column="COMPNM", sqlVariableLen=yes, maxLen=30};
    COMPDUNS int             {column="COMPDUNS"};
    COMPDESC string          {column="COMPDESC", sqlVariableLen=yes, maxLen=254};
    COMPLOGO string          {column="COMPLOGO", maxLen=30};
    CREDLIM int              {column="CREDLIM"};
    COMPBAL decimal(9,2)     {column="COMPBAL"};
    CREDST smallInt          {column="CREDST"};
    COMPDISC smallInt        {column="COMPDISC"};
    CREDTERMS smallInt       {column="CREDTERMS"};
    PHONE string             {column="PHONE", maxLen=18};
    FAX string               {column="FAX", maxLen=18};
    WEBURL string            {column="WEBURL", sqlVariableLen=yes, maxLen=50};
end

// basic library
library demoLibrary type BasicLibrary


    // Function Declarations
    function functionName()
    end

end
```

The power of declarative programming



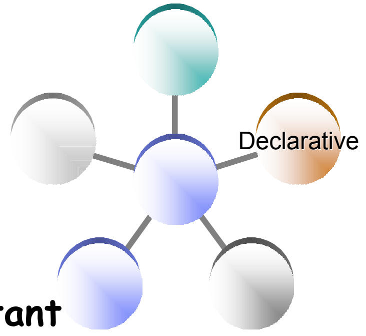
- Validation/Editing Rules
  - Via properties in “Data Items”…think Data Dictionary or “field reference file”
  - Define formatting & validation rules in a common place
  - Reuse data items for Records, screens, web pages, reports



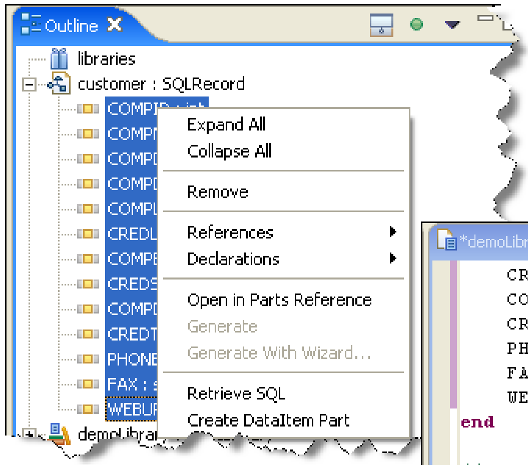
```
DataItem SSN Password char(9) {
    validatorFunction = "ValidateSSN()",
    displayUse = secret,
    pattern = "XXX-XX-XXXX",
    displayName = "Social Security No",
    inputRequired = yes}
end
```

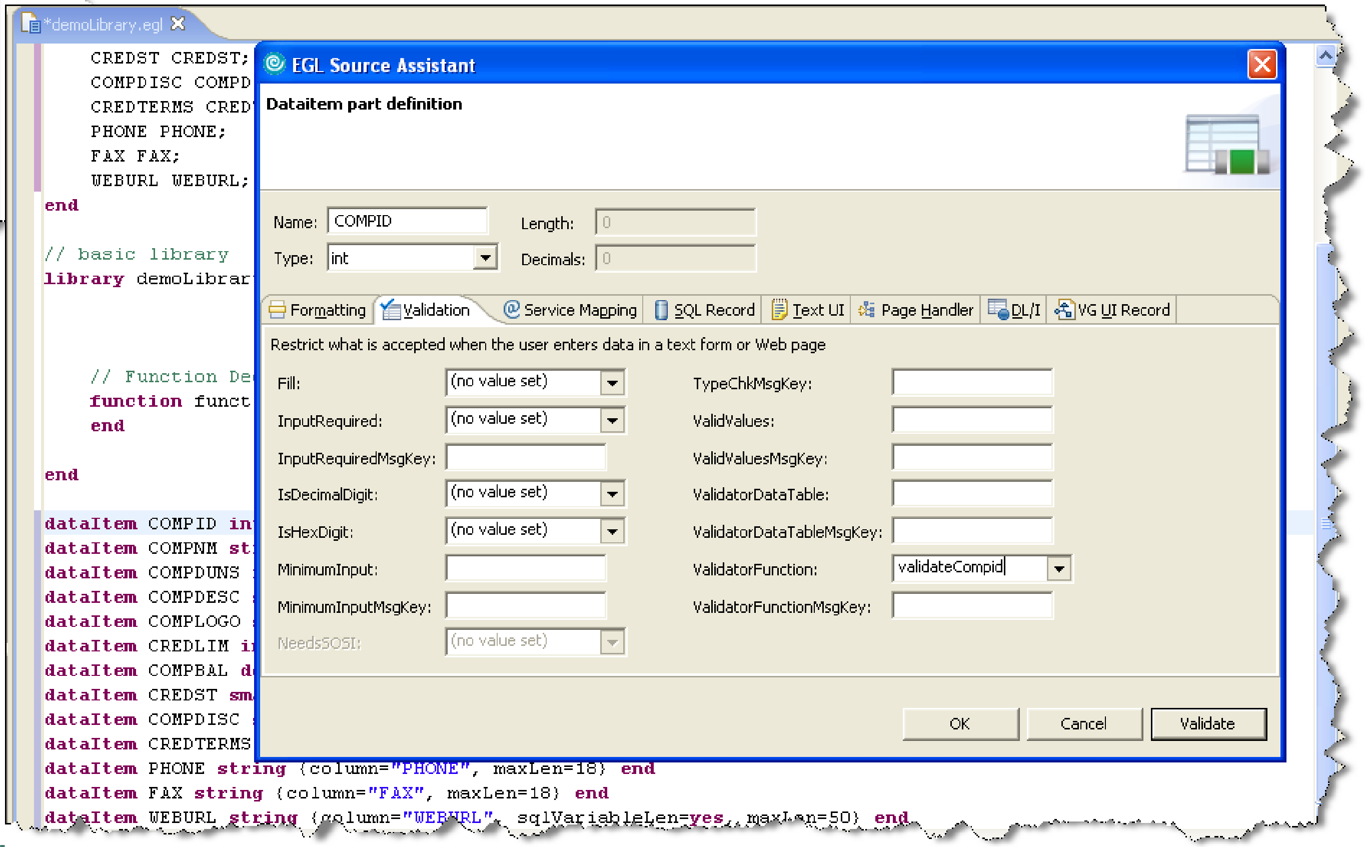
The power of declarative programming

Tools for data items



1. **Automatically create Data Items**
2. **Customize data items using the EGL Source Assistant**
3. **Specify edit, presentation and validation options**





```
    CREDST CREDST;
    COMPDISC COMPD
    CREDTERMS CRED
    PHONE PHONE;
    FAX FAX;
    WEBURL WEBURL;
end

// basic library
library demoLibrar


    // Function De
    function funct
    end

end

dataItem COMPID in
dataItem COMPNM st
dataItem COMPDUNS
dataItem COMPDESC
dataItem COMPLOGO
dataItem CREDLIM i
dataItem COMPBAL d
dataItem CREDST sm
dataItem COMPDISC
dataItem CREDTERMS
dataItem PHONE string {column="PHONE", maxLen=18} end
dataItem FAX string {column="FAX", maxLen=18} end
dataItem WEBURL string {column="WEBURL", sqlVariableLen=yes, maxLen=50} end
```

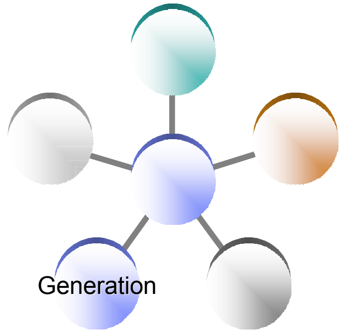
Dataitem part definition

Name:	COMPID	Length:	0
Type:	int	Decimals:	0

Restrict what is accepted when the user enters data in a text form or Web page

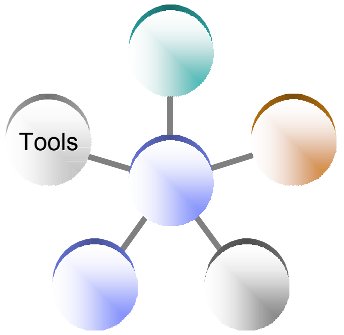
Fill:	(no value set)	TypeChkMsgKey:	
InputRequired:	(no value set)	ValidValues:	
InputRequiredMsgKey:		ValidValuesMsgKey:	
IsDecimalDigit:	(no value set)	ValidatorDataTable:	
IsHexDigit:	(no value set)	ValidatorDataTableMsgKey:	
MinimumInput:		ValidatorFunction:	validateCompid
MinimumInputMsgKey:		ValidatorFunctionMsgKey:	
NeedsSOSI:	(no value set)		

The power of generation



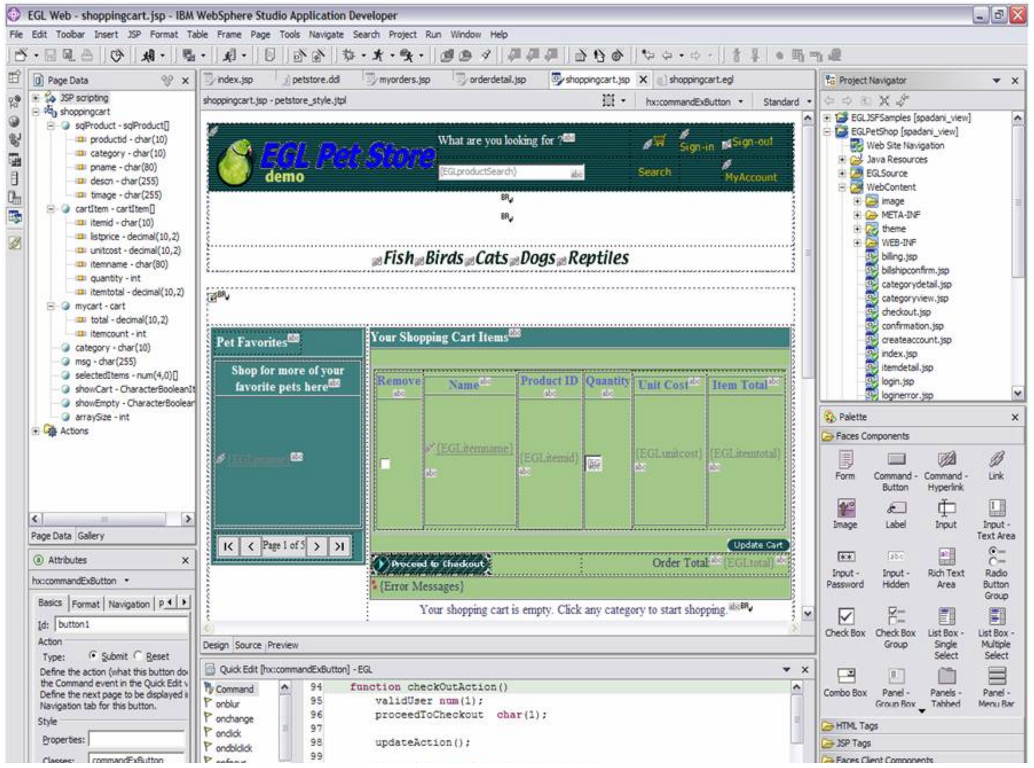
- Generate all the complex code needed to access middleware
  - MQ, DB’s, App Servers, Transaction Managers, …
  - …don’t spend creative developer time on this
- Deploy services to any platform/runtime
  - Not just application servers…inclusive of CICS, System i, IMS, …
  - …place them where they should be for optimal execution
- Deploy applications optimally to all key platforms
  - COBOL for System z CICS, IMS or Batch
  - COBOL for System i
  - Java for WAS or distributed platforms
  - …thereby breaking down “developer silos” by allowing same set of developers to build applications for all platforms

The power of tools: Robust Page Design



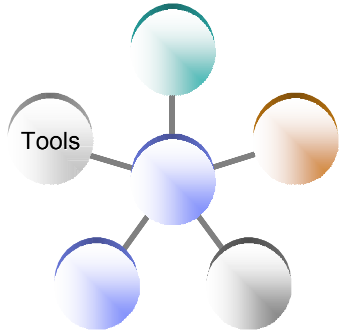
- First Class integration with Page Designer and JSF tools
  - Drop EGL data structures on JSP
    - Validation, editing, formatting rules from EGL Data Items applied
    - Appropriate UI controls rendered pre-bound to data declared in EGL Page
  - Server-side event handlers in EGL within context of page designer

- Integration is totally seamless
- No Java coding required to wire EGL data to JSF
- EGL logic can be used to handle user interaction with the JSP
- AJAX capability built in…partial refresh, etc…



```
function checkOutAction()
    validUser num(1);
    proceedToCheckout  char(1);

    updateAction();
```

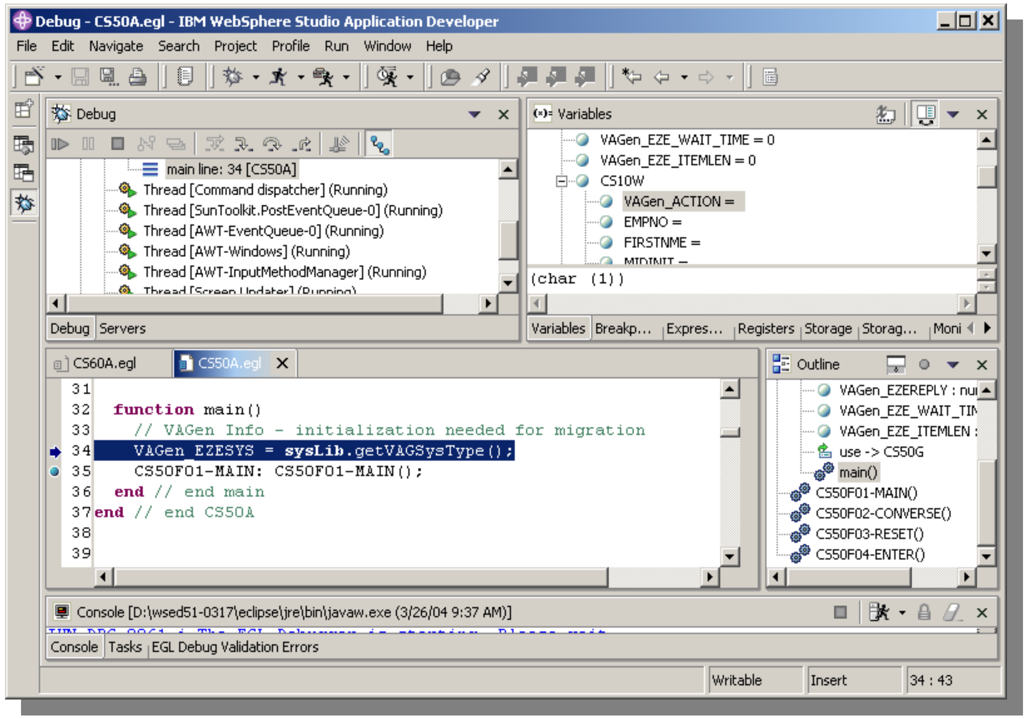
The power of tools: Debugger



- Debug entire application regardless of ultimate deployment targets
  - Transition from debugging JSP's to EGL code to Java to … and back

- EGL source debugger
  - Breakpoints
  - Watch variables
  - Change values
  - Extends base Eclipse debugger

- Great debugger = great productivity


Debug - CS50A.egl - IBM WebSphere Studio Application Developer
File Edit Navigate Search Project Profile Run Window Help
Debug
Variables
Debug Servers
Variables Breakp… Expres… Registers Storage Storag… Moni
CS60A.egl CS50A.egl
Outline
Console Tasks EGL Debug Validation Errors
Writable Insert 34 : 43

```
main line: 34 [CS50A]
Thread [Command dispatcher] (Running)
Thread [SunToolkit.PostEventQueue-0] (Running)
Thread [AWT-EventQueue-0] (Running)
Thread [AWT-Windows] (Running)
Thread [AWT-InputMethodManager] (Running)
Thread [Screen Updater] (Running)
```

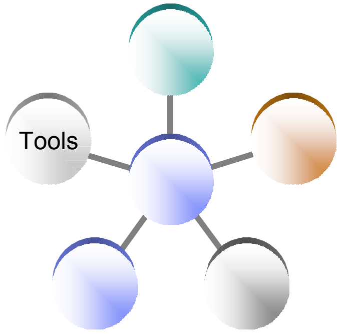
```
VAGen_EZE_WAIT_TIME = 0
VAGen_EZE_ITEMLEN = 0
CS10W
    VAGen_ACTION =
    EMPNO =
    FIRSTNME =
    MIDINIT =
(char (1))
```

```
31
32  function main()
33    // VAGen Info - initialization needed for migration
34    VAGen_EZESYS = sysLib.getVAGSysType();
35    CS50F01-MAIN: CS50F01-MAIN();
36  end // end main
37end // end CS50A
38
39
```

```
VAGen_EZEREPLY : nu
VAGen_EZE_WAIT_TIN
VAGen_EZE_ITEMLEN :
use -> CS50G
main()
CS50F01-MAIN()
CS50F02-CONVERSE()
CS50F03-RESET()
CS50F04-ENTER()
```

```
Console [D:\wsed51-0317\eclipse\jre\bin\javaw.exe (3/26/04 9:37 AM)]
[illegible]
```

The power of tools: Model Transformations
A new generation of Architected Rapid Application Development



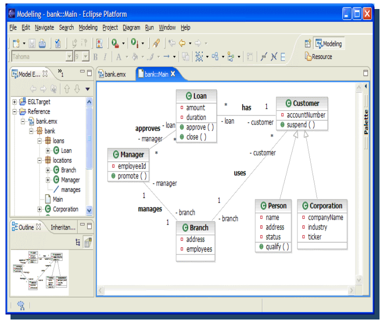
- Transform UML models to EGL
- Best way to go…

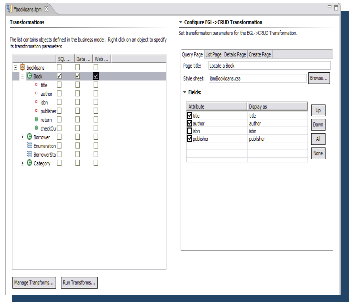
  from ***SOA models***

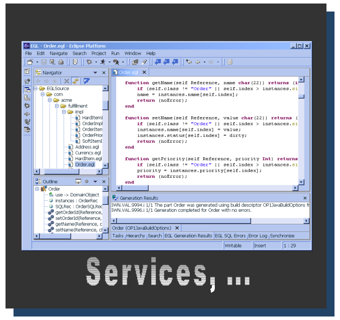
  to ***construction of services***

  to ***services***

  deployed on System z, System i, or anywhere else







1. Model

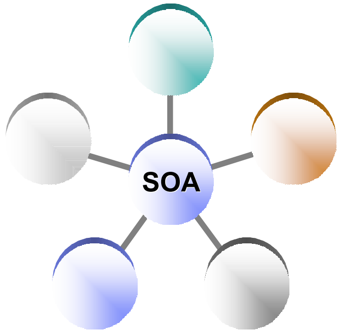
2. Define Transformation Parameters

3. Transform to EGL code

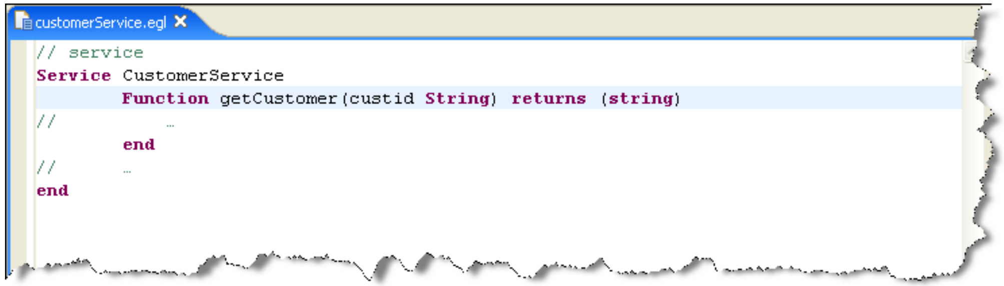
4. Deploy to platform (z, i, or distributed)

- Traceability from requirements to code
- Create your own transformations
- Transformations enriched by Transformation Parameters
- Easily build / deploy Services on host

The power of Services
Built into the language

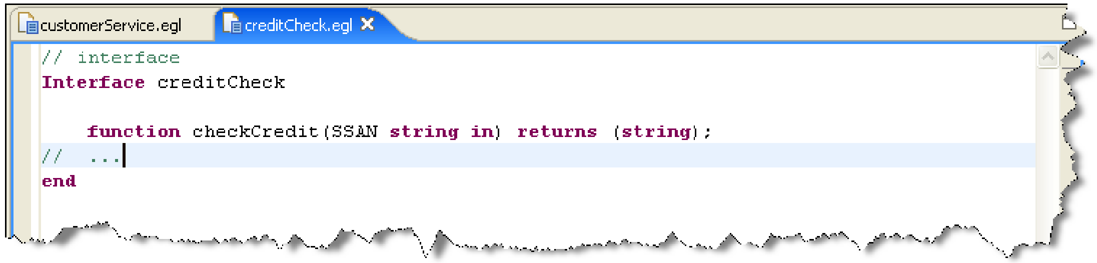


- Service part
  - a generatable part containing code that will be accessed
    - from EGL code by way of a local or TCP/IP connection (*EGL Service*)
    - from any code by way of an HTTP connection (*EGL Web service*).



```
// service
Service CustomerService
        Function getCustomer(custid String) returns (string)
//          …
        end
//      …
end
```

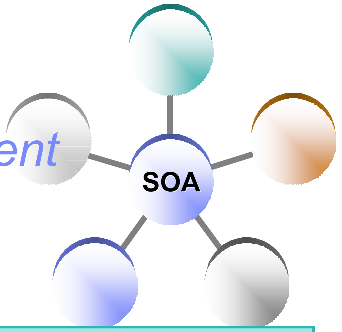
- Interface part
  - Used to access external services as EGL services or simply to provide separation of concern



```
// interface
Interface creditCheck

    function checkCredit(SSAN string in) returns (string);
//   ...
end
```

The power of Services

EGL: cross platform language for business oriented services development



At development time…

- Focus on the business logic
- Implement SOA design elements: services and interfaces
- Leverage existing business developers for new SOA development
- Ignore deployment targets/technology while coding/testing

Leverage external web services…

- EGL Interfaces
  - represent external web services
  - Are created via import from WSDL
  - Allow the EGL developer to stay within the context of the EGL programming model

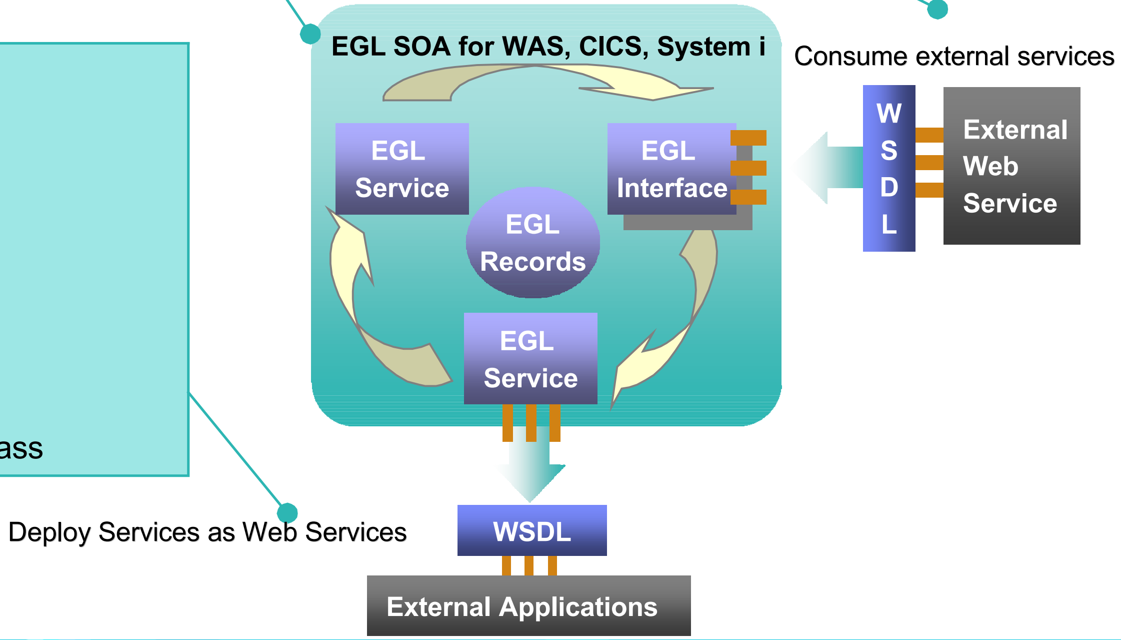
Deploy EGL services…

To any platform

- Java to WAS/Tomcat/etc.
- COBOL to CICS, iSeries (1Q 2007)
- COBOL to IMS (2H 2007)

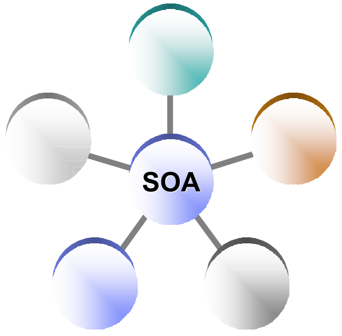
As…

- A Web service (uses SOAP)
- A private service (uses CICS ECI or TCP)
- Other SOA runtimes when they reach critical mass



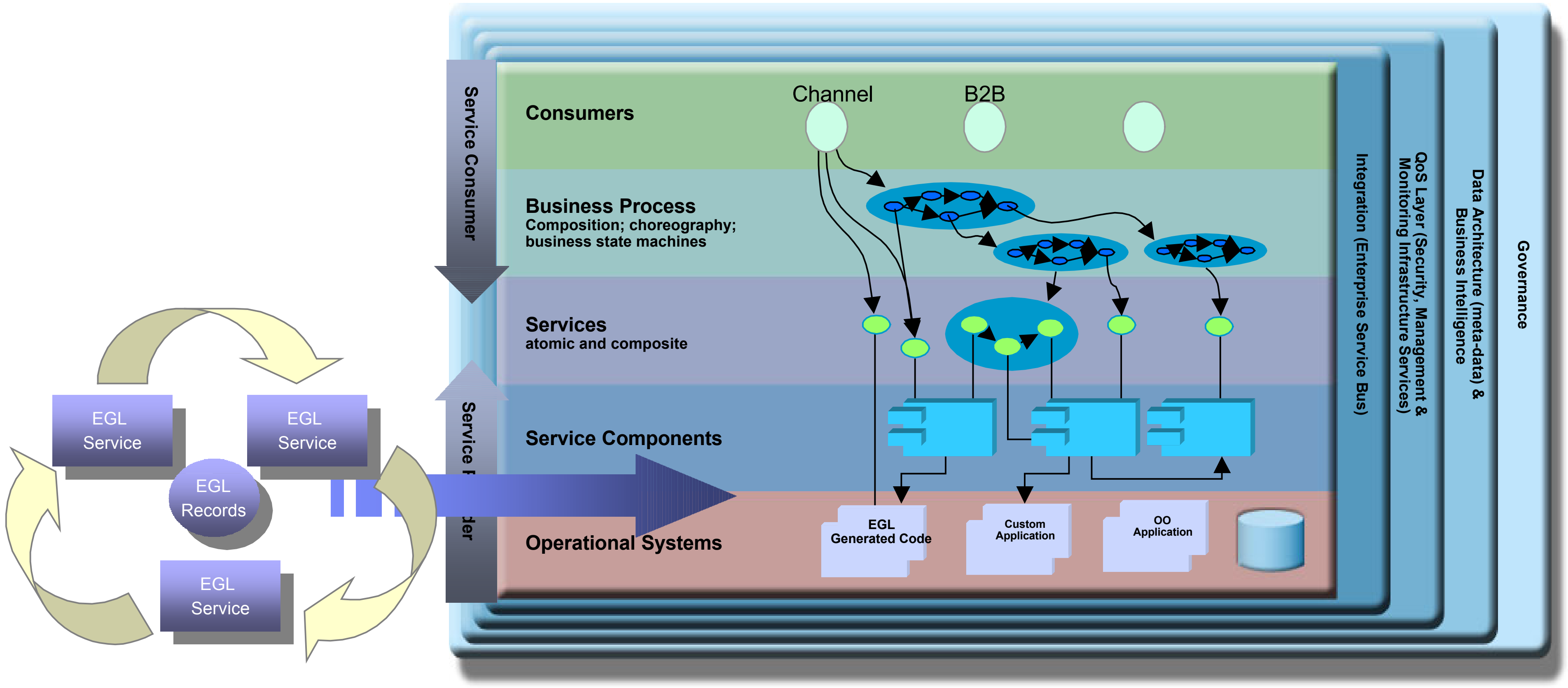
The power of Services

Seamless integration with SOA stack



EGL Services can be generated into deployable artifacts that are accessible as Web Services

EGL data appears as XML payload with no need for transformation

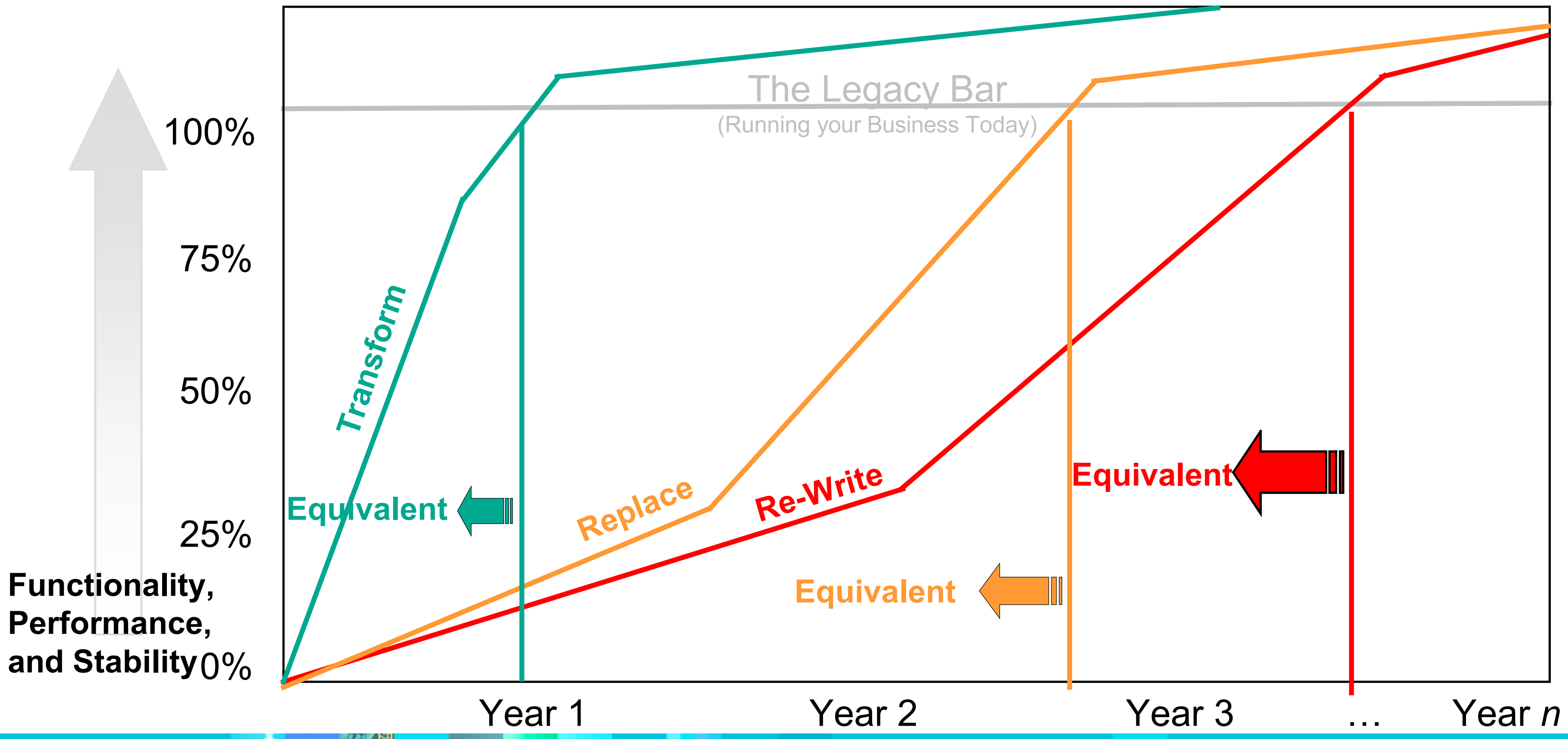


Topics

- Why EGL
- EGL Overview
- EGL and Application Transformation
- EGL … What’s Next
- Customer stories
- Summary

Leveraging EGL for Application Transformation

- Many legacy applications are valuable and reliable, but their aging architecture is an inhibitor to business growth and process change.
- Many Customers are considering the following options

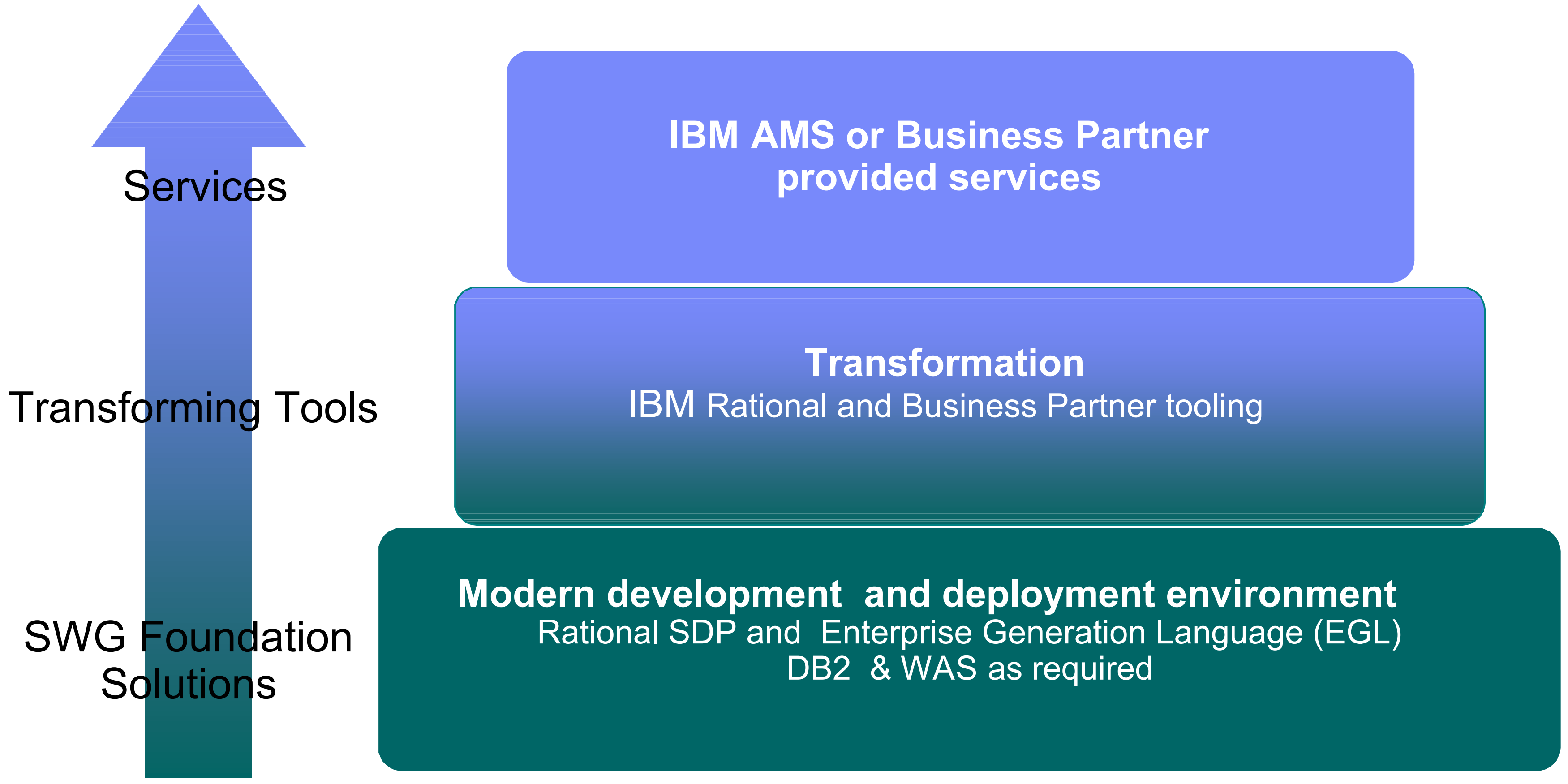


Source: ATERAS

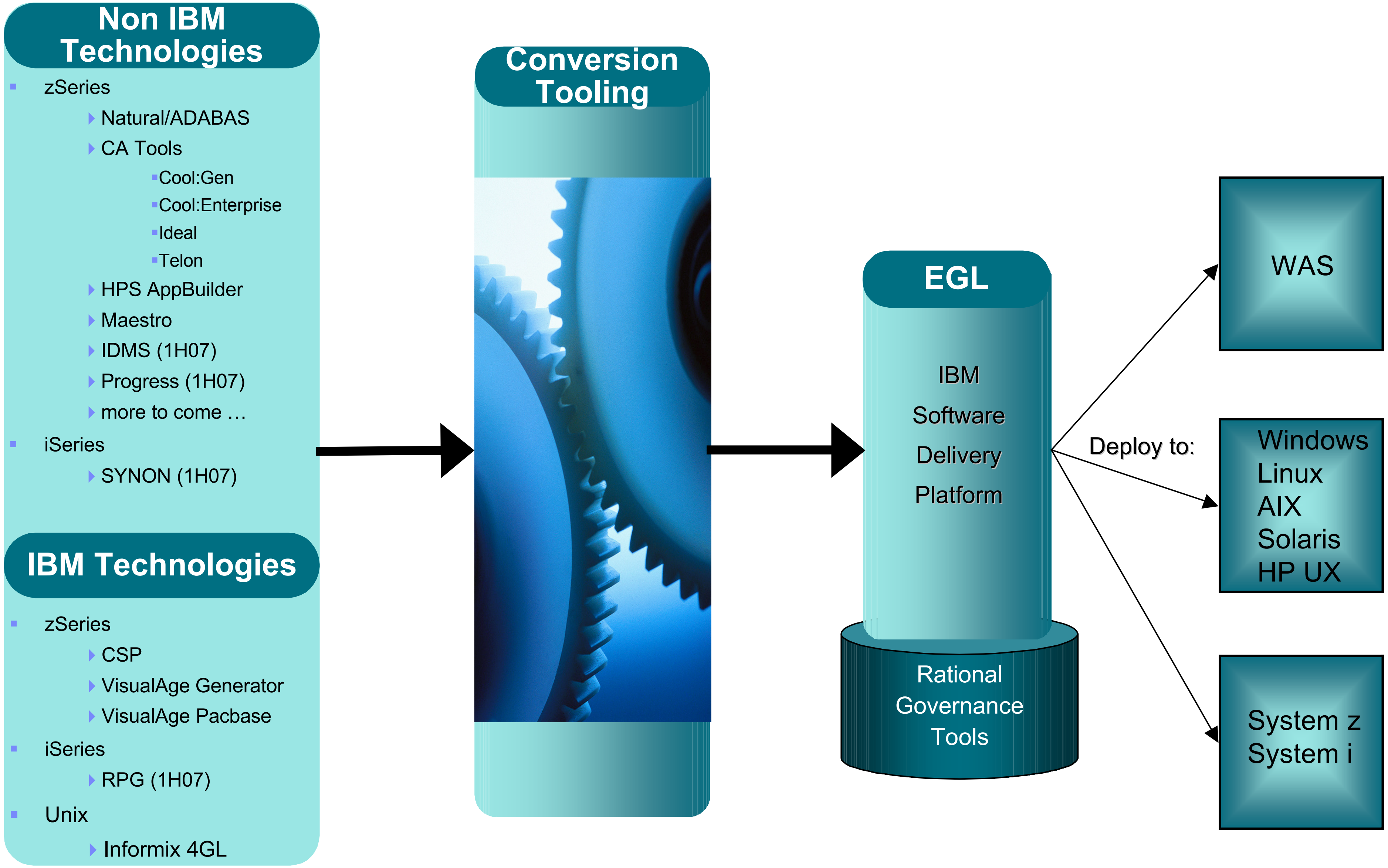
Why EGL for Application Transformation

- Platform flexibility
  - Broad choice of new target environments
- Modern, Robust, Open, SOA ready
  - Future-proof architecture allows to grow with business requirements
- Easier to learn for legacy 4GL developers
  - No need to re-staff, productive in a very short time
- Procedural nature of EGL target has greater “affinity” with original 4GL source
  - Better more natural “mapping
  - Easier to automate the transformation process
  - End result is understandable and maintainable
- Eliminate costly runtime charges

IBM Solution for Application Transformation


Services
Transforming Tools
SWG Foundation Solutions
IBM AMS or Business Partner provided services
Transformation
IBM Rational and Business Partner tooling
Modern development and deployment environment
Rational SDP and Enterprise Generation Language (EGL)
DB2 & WAS as required

The Transformation Proposition



Topics

- Why EGL
- EGL Overview
- EGL and Application Transformation
- EGL … What's Next
- Customer stories
- Summary

Version 7 Content

- Deployment of Services to System z (CICS) and System i
    - Web Services
    - Native EGL Services (No SOAP)
- Transformation from UML Models and DB2 Schemas to EGL Code
    - Includes RSA transformation engine (enabled only for EGL, not general purpose)
    - Robust transformation with additional “Transformation Parameters”
        - O/R mapping (how to handle inheritance, realize associations (key generation, foreign key, ..)
        - Data mapping (representing UML attributes etc in generated code, UML types mapping, ..)
        - Name Mapping (UML names to domain names)
        - UI (what to expose in UI, how to represent properties, stylesheets,..)
        - Data access operations (database concurrency, exceptions, ..)
        - Extensibility (open APIs to create custom transformation, templates can be customized)
- Rich Client Support - 1st of 2 Stages
    - Runtime support and ability to define RCP UI via tags (no tooling… yet… but coming soon)
- Performance
- Language Extensibility
    - Support user defined EGL attributes in language specification, definition dialogs, generation exits
    - Key to EGL standardization and open source initiatives
- Miscellaneous Debugger and Language Enhancements

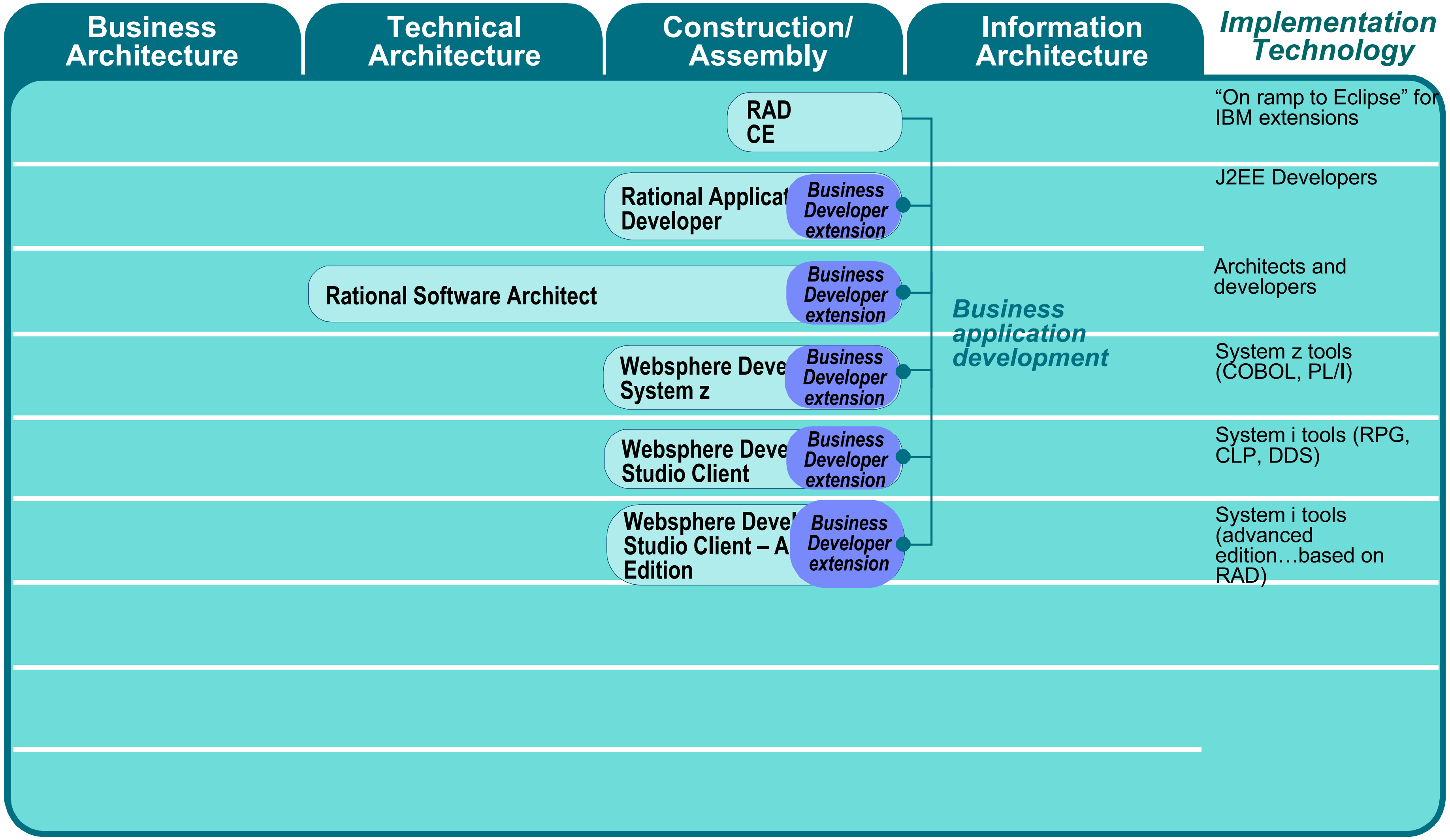
Version 7

- Target Availability
  - 2Q2007

- Repackaging
  - Overall Rational strategic move to componentization
    - Smaller, more consumable, product offerings
    - More dynamic delivery schedules
    - EGL offering is the first such “extension”
    - EGL “Component” plugs into most IBM SDP base IDEs
      - RAD, RSA, WDz, WDSC, WDSC-AE
      - RAD CE most likely will not be supported, due to lack of capability in this entry-level product
  - Rational increasing focus on “business developer” market and System z & i markets
    - EGL is a key element of this
  - Product Name:
    - Rational Business Developer extension

Analysis Design & Construction v7 Product Portfolio

Bases for Rational Business Developer extension


Business Architecture
Technical Architecture
Construction/ Assembly
Information Architecture
Implementation Technology
RAD CE
"On ramp to Eclipse" for IBM extensions
Rational Applicat Developer
Business Developer extension
J2EE Developers
Rational Software Architect
Business Developer extension
Architects and developers
Websphere Deve System z
Business Developer extension
System z tools (COBOL, PL/I)
Websphere Deve Studio Client
Business Developer extension
System i tools (RPG, CLP, DDS)
Websphere Devel Studio Client – A Edition
Business Developer extension
System i tools (advanced edition…based on RAD)
Business application development

Topics

KBC
Unifying application development across platforms (Unix and Mainframe) and transaction managers (WAS and IMS)

- **Background:**
  - Belgian Bank & Insurance company with a successful expansion in Central and Eastern Europe, 50000 employees working for 12 million clients across Belgium, Czech Republic, Slovak Republic, Poland, Hungary and Slovenia
  - Acquisition of several Bank and/or Insurance companies in Central-Europe in past years and expected to continue.
  - Striving for cost reduction through synergy and integration
- **Objectives:**
  - **Create interchangeable developers**. Shift from monolithic (3270-)applications to browser based and open systems. Brand 600 mainframe developers (KBC) as multi employable developers for Unix and mainframe
  - **Enabling component based architecture**: Shift to component based architecture, product factories and multi channel
  - **Support multi-site development**: Project teams scattered across different European countries and India.
  - **Reuse automation investments.** Business wants to create synergy by using the same solutions in Belgium and CE. Develop solutions & business components reusable within each KBC subsidiary. Support models of develop once, deploy locally or develop once, deploy centrally for different subsidiaries
- **Solution:**
  - Standardize on EGL

Topics

- Why EGL for customers
- EGL Overview
- EGL and Application Transformation
- EGL … What’s Next
- Customer stories
- Summary

Summary

- EGL is central to IBM Rational ADC strategy:
  - SOA
    - easy for business developers
    - natively for all platforms (CICS, WAS, IMS, System i)
  - Consolidation point for business developers and business applications
    - application transformations and modernization
    - natural transition for business developers
  - Time is right for tools like EGL/RBDe as customers try to figure out how to:
    - be more productive
    - leverage their business developers (COBOL, RPG, Visual Basic, …)
    - Embrace new application architectures
  - High productivity, end-to-end, cross platform (native) deployment