Modern Application Development Featuring Web 2.0 for System z

Rational Business Developer and EGL

Rational. software

Key architecture management challenges

SAP HR system

Custom .Net applications

Back-office legacy systems

Home-grown line of business apps.

Oracle Siebel CRM

Windows

.Net

AIX

Linux

Java

COBOL/zOS

5iOs

Existing Applications

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

Skills

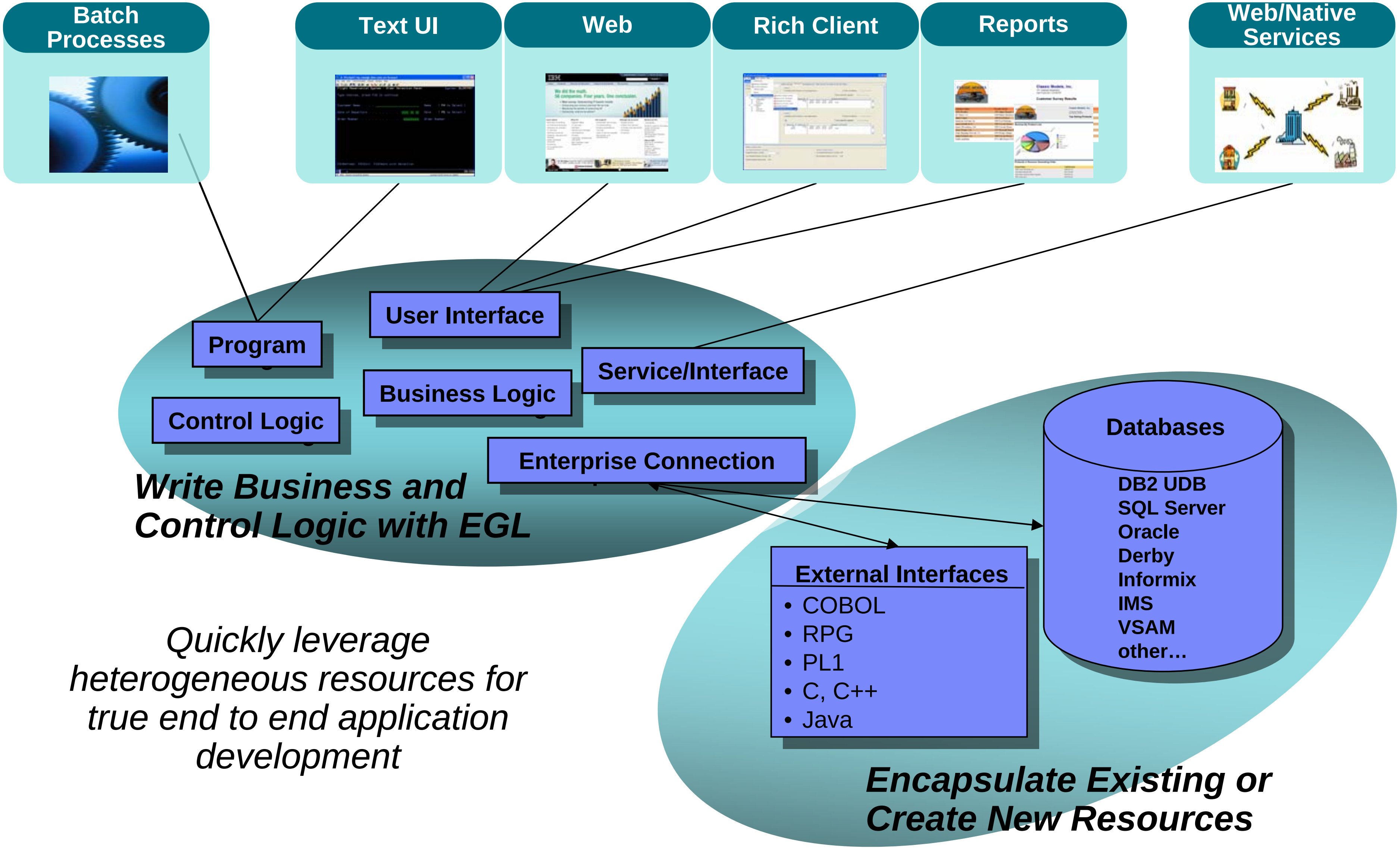
- Skills silos
- Skills mismatch
- Erosion of legacy platforms skills

Platforms / Middleware

- Proliferation
- Coexistence
- Complexity

High costs **Compromise** **Slow response**

Solution: **Enterprise Generation Language**



Batch Processes

Text UI

Web

Rich Client

Reports

Web/Native Services

User Interface

Program

Service/Interface

Business Logic

Control Logic

Enterprise Connection

Write Business and Control Logic with EGL

Databases

DB2 UDB
SQL Server
Oracle
Derby
Informix
IMS
VSAM
other…

External Interfaces

- COBOL
- RPG
- PL1
- C, C++
- Java

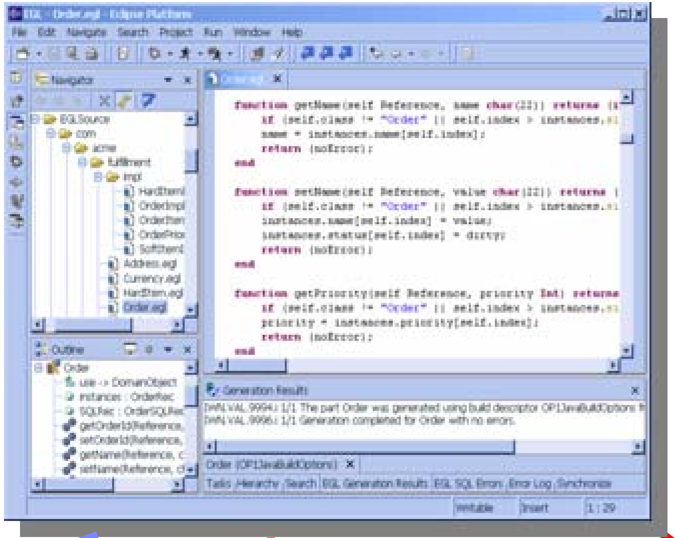
Encapsulate Existing or Create New Resources

Quickly leverage heterogeneous resources for true end to end application development

Solution: **Simplify and Accelerate cross-platform development**

Build once, deploy anywhere

IBM Rational Business Developer



```
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
```

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

Java

COBOL

Java

COBOL

Java

System z

- WebSphere
- USS
- Linux
- Batch
- CICS

System i

- WebSphere
- Native i5OS
- Native i5OS

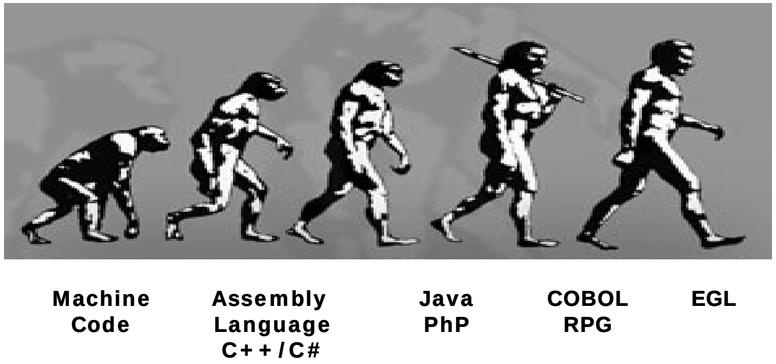
Windows, Linux, Unix

- WebSphere
- Tomcat

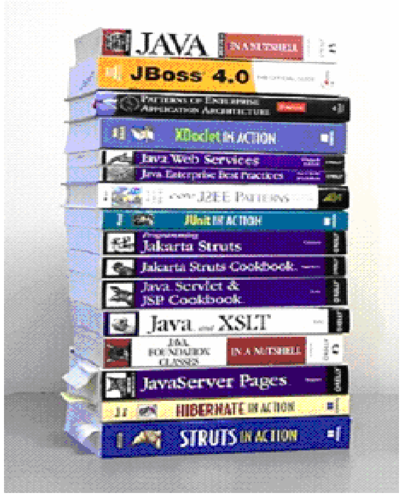
Platform Flexibility with IBM Rational Business Developer

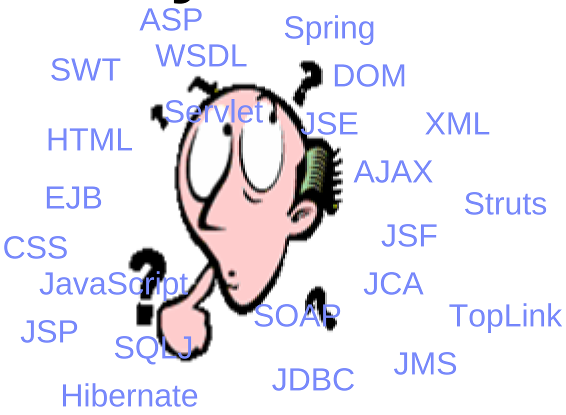
Introduction to EGL

- EGL (Enterprise Generation Language) is IBM’s newest business language
- EGL is a true cross-platform, cross-tier language targeted at all types of developers
  - Traditional, procedural, object-oriented, Web, SOA, etc.
- The goal of EGL is to shield developers from complexities that are unnecessary for building business applications.
- EGL is an excellent target language for migration of existing, traditional applications.



Why EGL? Because building applications today is not easy



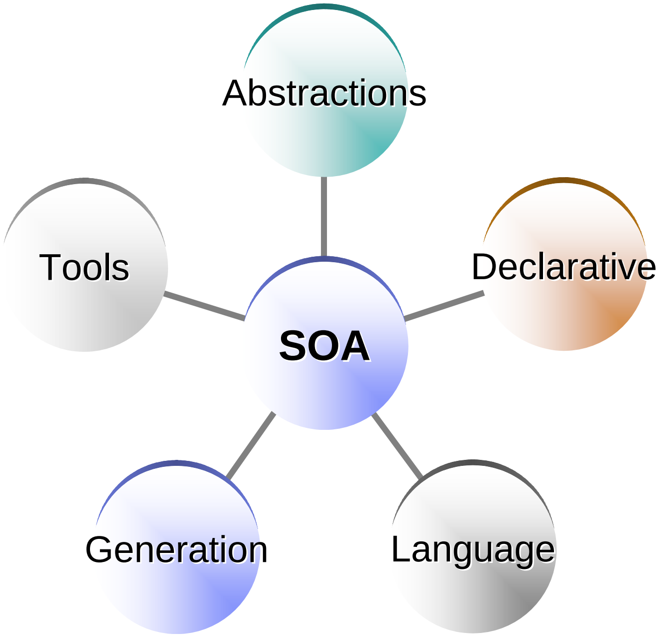


ASP Spring SWT WSDL DOM Servlet HTML JSE XML AJAX EJB Struts CSS JSF JavaScript JCA SOAP TopLink JSP SQLJ JMS JDBC Hibernate

- Learn and master myriads of
  - programming languages, semantics, paradigms and styles
  - middleware interfaces, frameworks and libraries
- Constantly emerging new technologies
- 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 and business pressure may not afford time
  - Results may be sub-optimal and some may not make it
  - End up with poorly written applications

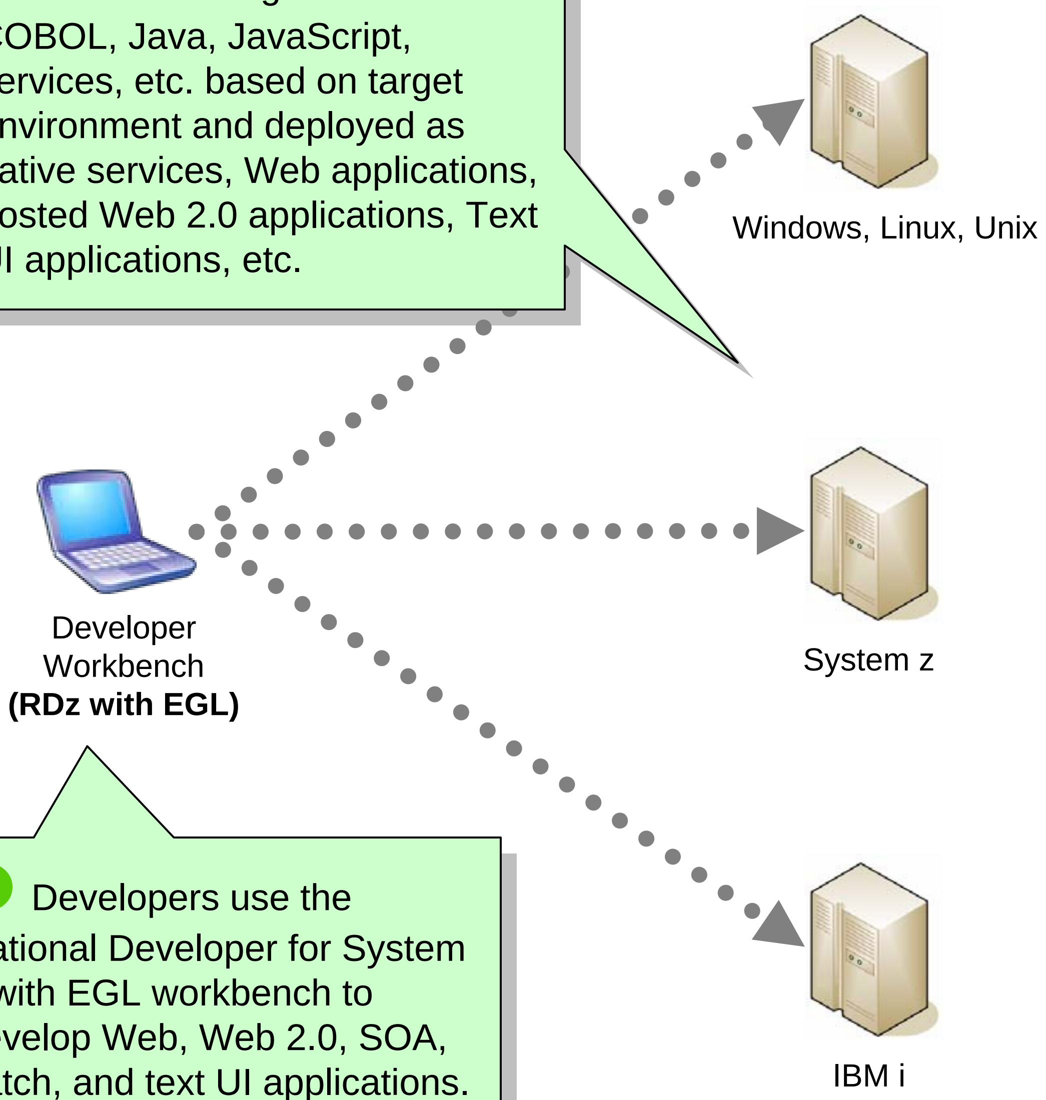
- Code at a more abstract and simpler level
- Easy to learn, modern and comprehensive language
- Keeps up with emerging technologies
- Inter-operates seamlessly with legacy

The EGL Philosophy



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

❷ EGL code is generated as COBOL, Java, JavaScript, services, etc. based on target environment and deployed as native services, Web applications, hosted Web 2.0 applications, Text UI applications, etc.



Java and JavaScript generation
- WebSphere Application Server
- Apache Tomcat
- Java Runtime Environment

Java generation
- WebSphere Application Server
- USS
- z/Linux

JavaScript generation
- WebSphere Application Server

COBOL generation
- Batch
- CICS
- IMS

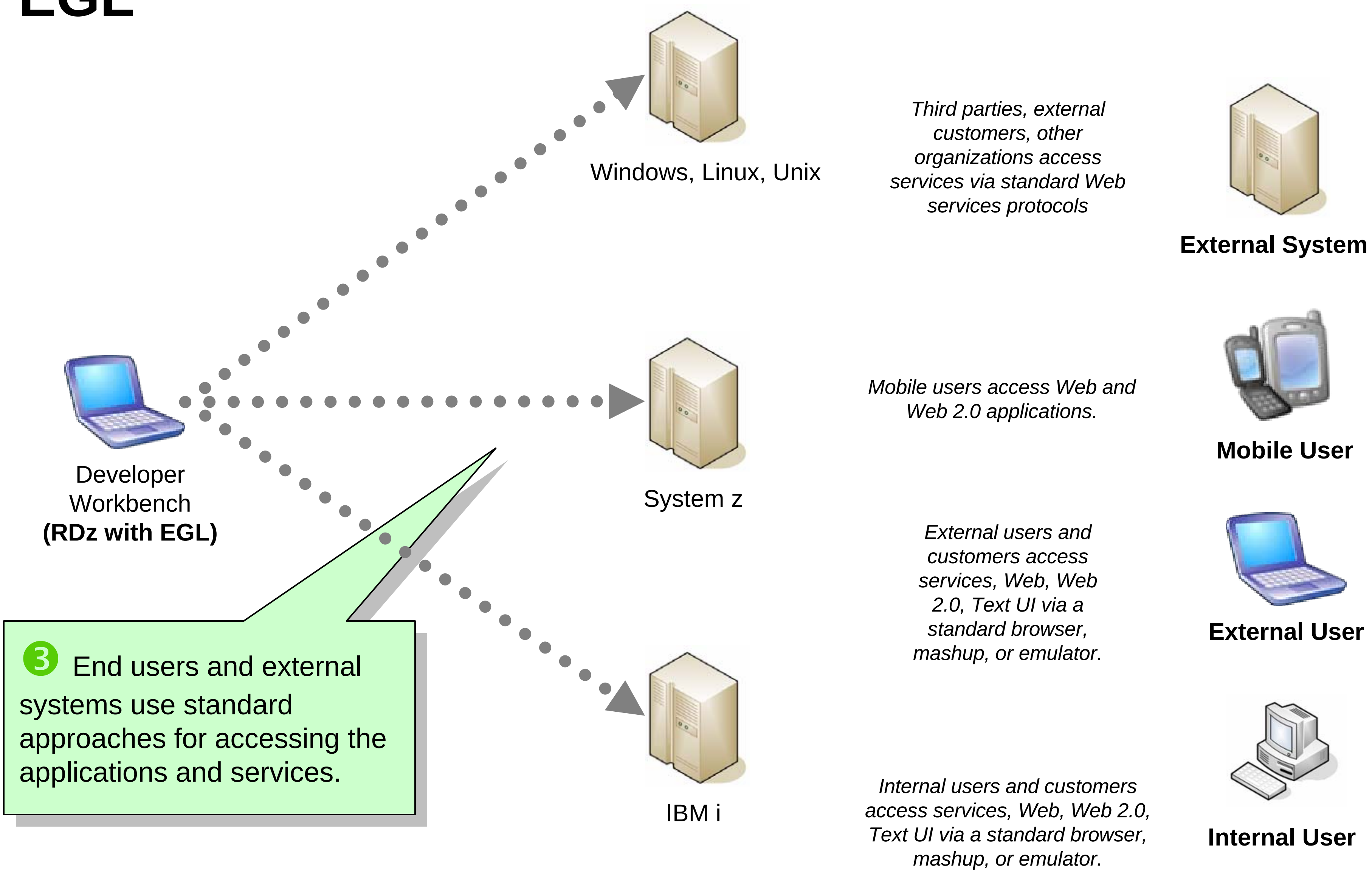
Java generation
- WebSphere Application Server
- Native IBM I

JavaScript generation
- WebSphere Application Server

COBOL generation
- IBM i

❶ Developers use the Rational Developer for System z with EGL workbench to develop Web, Web 2.0, SOA, batch, and text UI applications.

EGL



Third parties, external customers, other organizations access services via standard Web services protocols

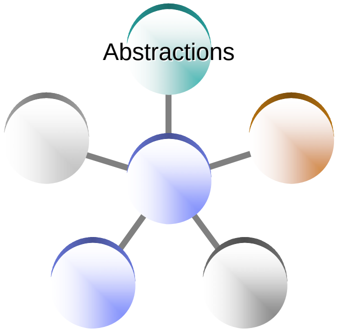
Mobile users access Web and Web 2.0 applications.

External users and customers access services, Web, Web 2.0, Text UI via a standard browser, mashup, or emulator.

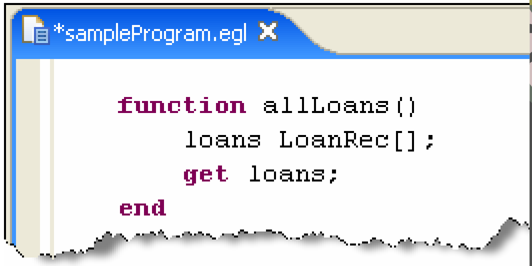
Internal users and customers access services, Web, Web 2.0, Text UI via a standard browser, mashup, or emulator.

❸ End users and external systems use standard approaches for accessing the applications and services.

The power of abstractions

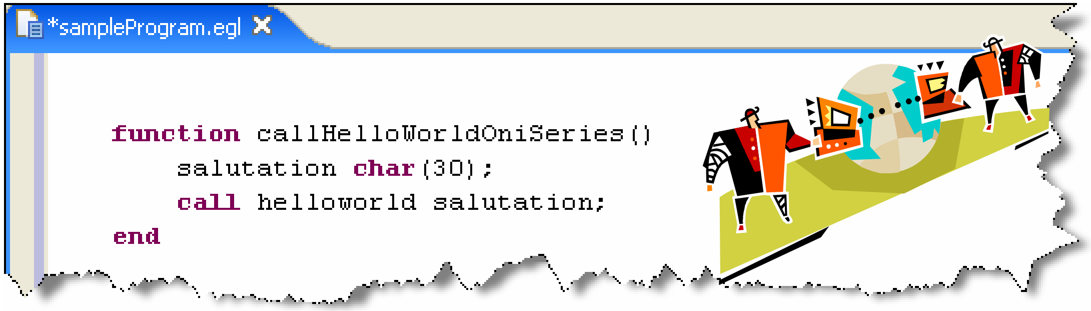


- **Data access:**
  - “Records” provide access to:
    - SQL, Indexed, Relative, Serial, DL/I, MQ, Service data
  - Common Verbs for data access (**Get, Add, Replace, Delete**)
  - Allows complete access to SQL statement if needed
  - Common Error Handling



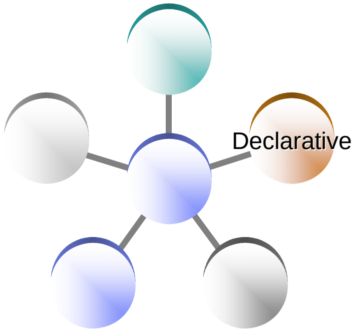
```
function allLoans()
     loans LoanRec[];
     get  loans;
end
```

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

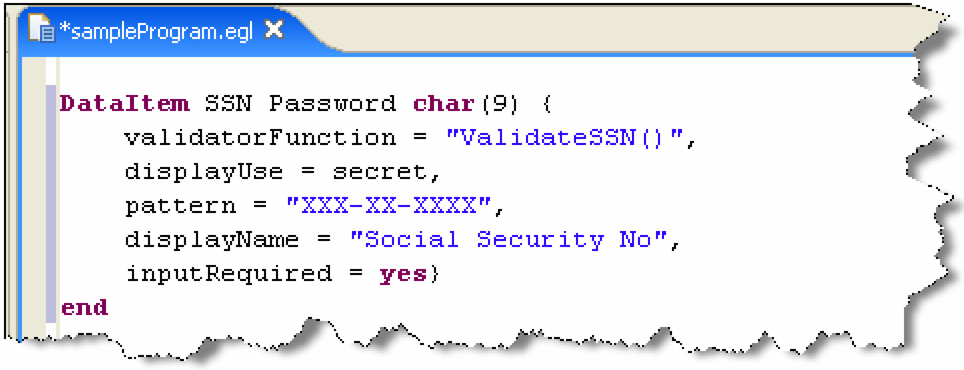


```
function callHelloWorldOniSeries()
     salutation char(30);
     call helloworld salutation;
end
```

The power of declarative programming



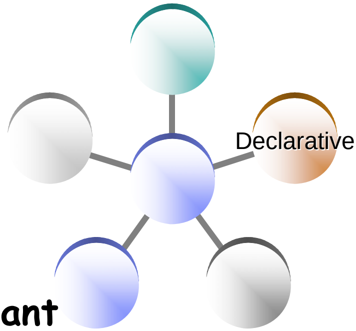
- **Validation/Editing Rules**
  - Via properties in “Data Items”…think Data Dictionary or “field reference file”
  - Define formatting & validation rules in a common place
  - Reuse data items for Records, screens, web pages, reports



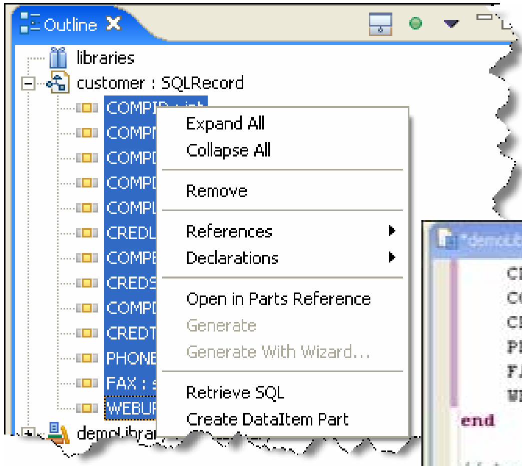
```
DataItem SSN Password char(9) {
    validatorFunction = "ValidateSSN()",
    displayUse = secret,
    pattern = "XXX-XX-XXXX",
    displayName = "Social Security No",
    inputRequired = yes}
end
```

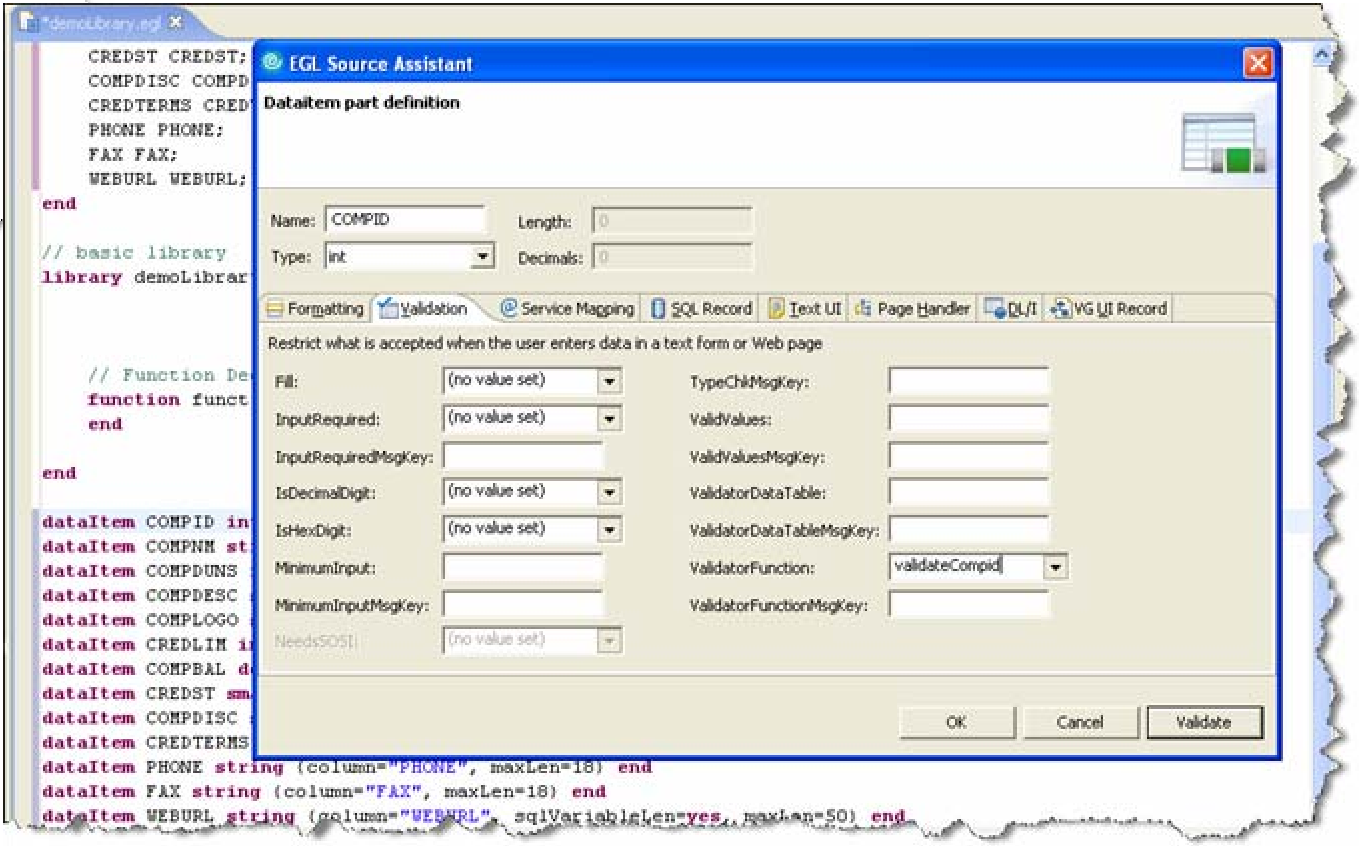
The power of declarative programming

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

The Power of The Language

EGL Data Model

Reuse Existing Databases and Data Model Definitions:

Import UML Class Model from Rational Rose/XDE

Import Data model from DB Schema

Create Record Definitions using SQL Retrieve

Central Data Dictionary:

Define a central data definition dictionary across multiple applications.

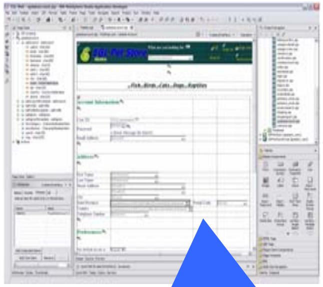
Define display, formatting, validation for data items

Define dynamic tables for lookup, error handling etc.

Reuse DataItem definitions to create application Record definitions

Data access from multiple sources:

Create Record definitions to access data from Relational, MQ, CICS, IMS, XML* etc.*



Reuse Record and DataItems definitions across multiple pages for consistency:

Automatically enforce data access, display, formatting, and validations rules defined by record and data items

```
// Customer SQL Record
Record Customer SQLRecord
    { tableNames=("DB2ADMIN.CUSTOMER"),
    keyItems=("customerId") }

    customerId CustomerId;
    firstName FirstName;
    lastName LastName;
    …
    state State;
end
```

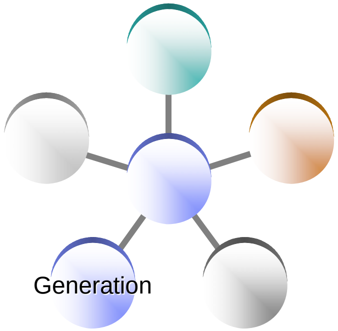
```
// StoreLocation SQL Record
Record Store SQLRecord

    …
    state State;
end
```

```
DataItem CustomerId int
    { column=CUSTOMER_ID,
    range=(1, 1000),
    displayName="Customer Number",
    format= "#######" }
end
```

```
DataItem State char(2)
    {displayName="State Abbr.",
    format= "AA",
    validatorTable=StateTable }
end
```

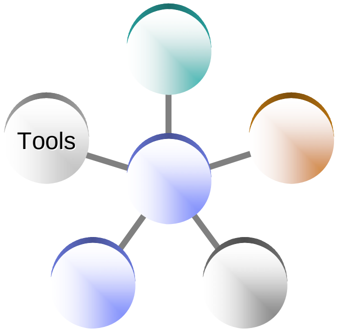
```
DataTable StateTable type matchValidTable
    StateAbbreviation   char(3);
    {contents = [["NC"], ["MN"], ["TX"], ["VA "] ...]}
end
```

The power of generation



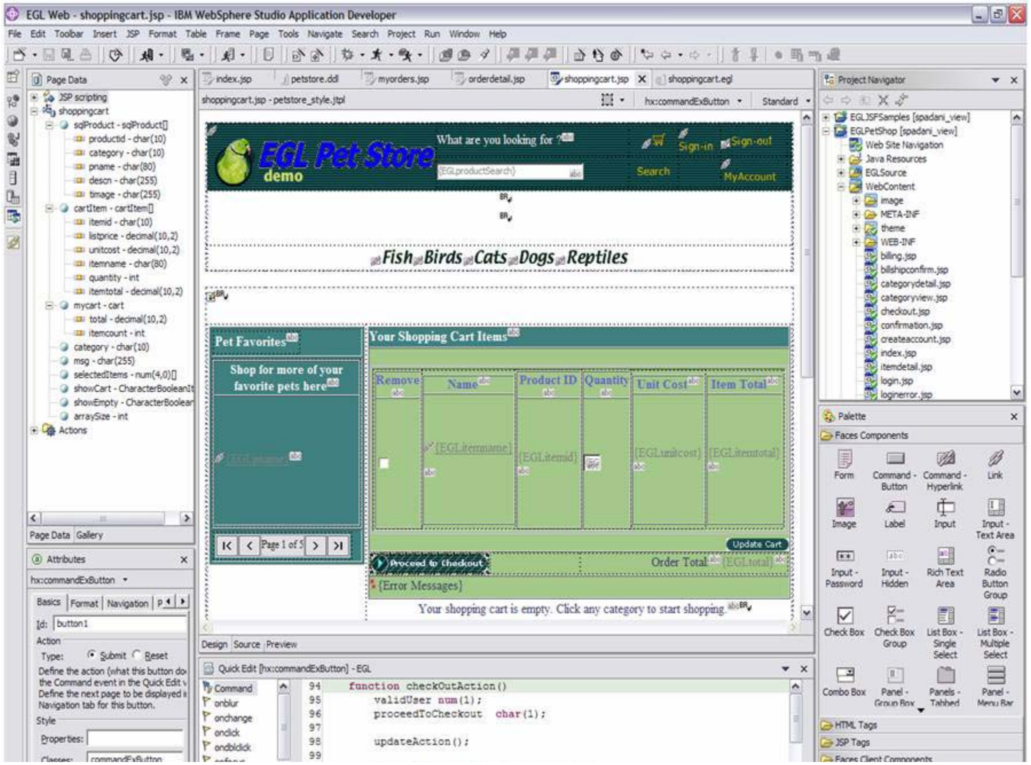
- **Generate all the complex code needed to access middleware**
  - MQ, DB’s, App Servers, Transaction Managers, …
  - …don’t spend creative developer time on this
- **Deploy services to any platform/runtime**
  - Not just application servers…inclusive of CICS, System i, IMS, …
  - …place them where 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



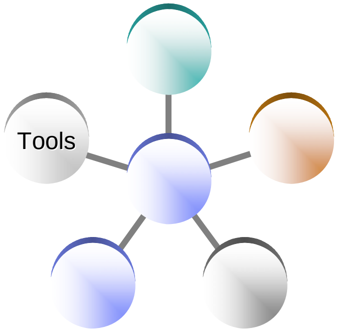
- **First Class integration with Page Designer and JSF tools**
  - Drop EGL data structures on JSP
    - Validation, editing, formatting rules from EGL Data Items applied
    - Appropriate UI controls rendered pre-bound to data declared in EGL Page
  - Server-side event handlers in EGL within context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

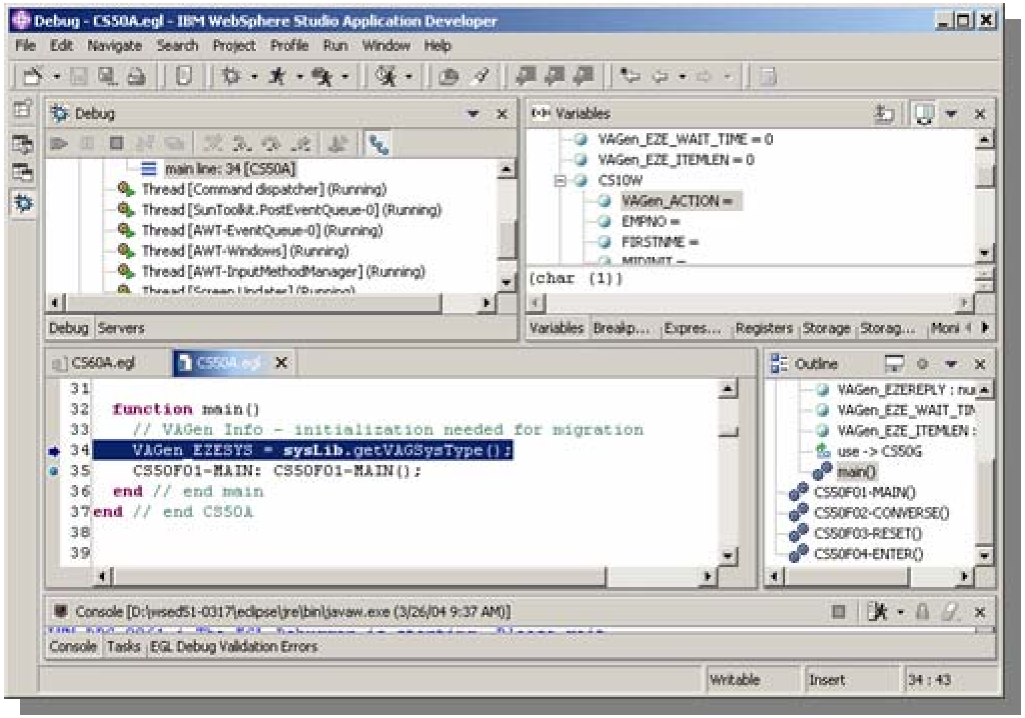
The power of tools: Debugger



- **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**



```
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))
```

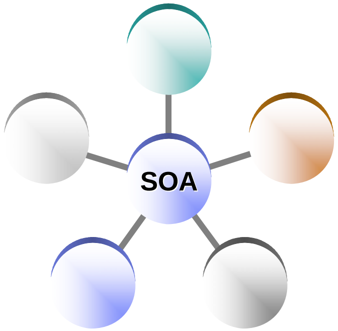
```
  function main()
    // VAGen Info - initialization needed for migration
    VAGen_EZESYS = sysLib.getVAGSysType();
    CS50F01-MAIN: CS50F01-MAIN();
  end // end main
end // end CS50A
```

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

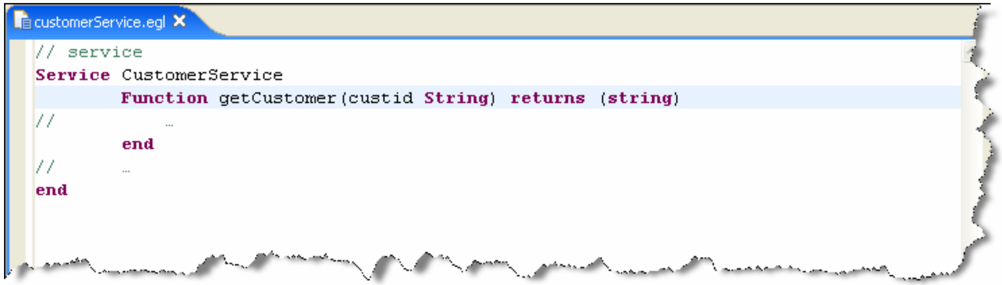
Console [D:\wsed51-0317\eclipse\jre\bin\javaw.exe (3/26/04 9:37 AM)]

The power of Services

Built into the language

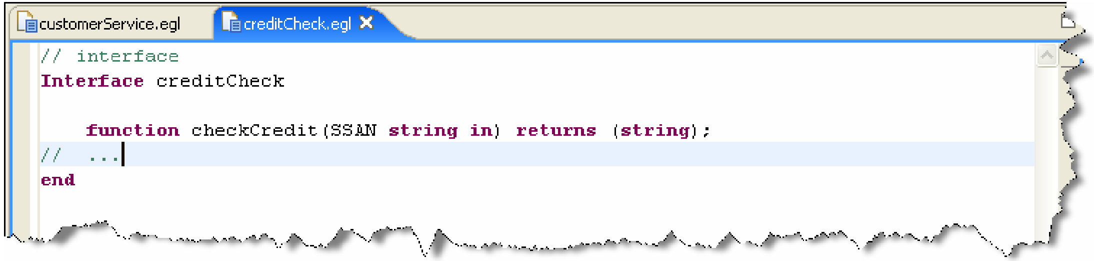


- **Service part**
  - a generatable part containing code that will be accessed
    - from EGL code by way of a local or TCP/IP connection (*EGL Service*)
    - from any code by way of an HTTP connection (*EGL Web service*).



```
// 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

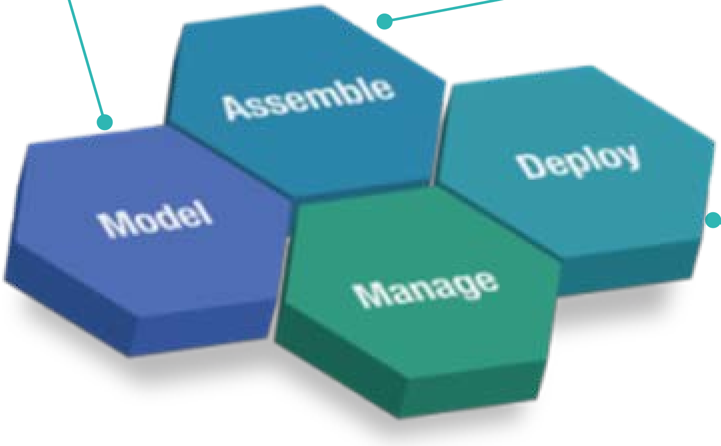
EGL in IBM Rational Business Developer: cross platform language for business oriented services development

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

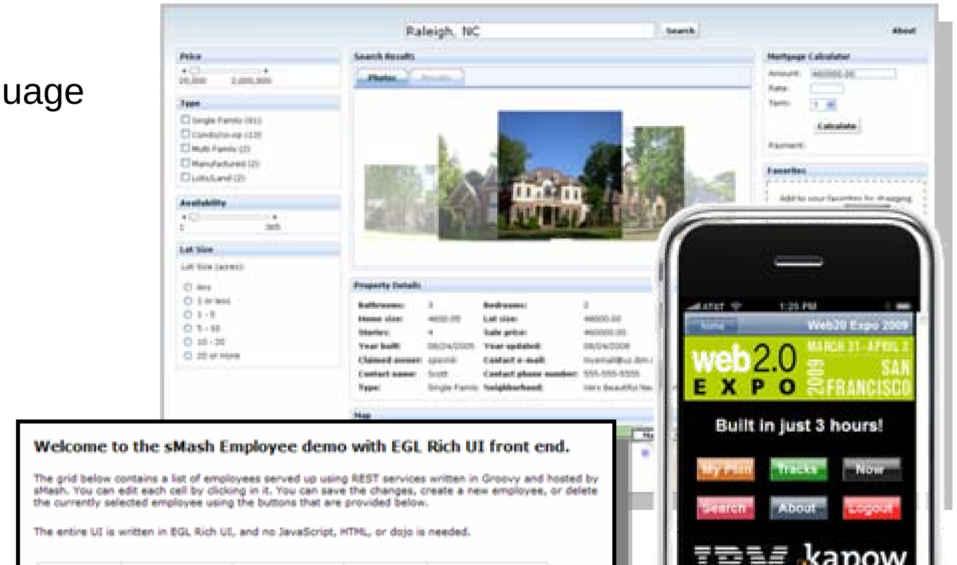
- Java to WAS/Tomcat/etc.
- COBOL to CICS, iSeries (1Q 2007)
- COBOL to IMS (2H 2007)

As…

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

Rich User Interfaces with EGL

- Build end-to-end Web 2.0 quickly with a single language
- Fully extensible
- Use a language that is easy to learn
- Benefit from productive development
- Use Ajax without any of its complexities
- Use a rich, extensible widget library
- Debug web applications in an Eclipse IDE
- Consume any type of web service



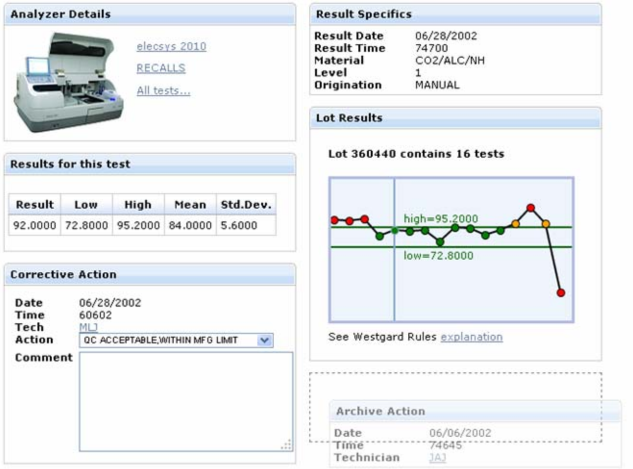
Built in just 3 hours!

Welcome to the sMash Employee demo with EGL Rich UI front end.

The grid below contains a list of employees served up using REST services written in Groovy and hosted by sMash. You can edit each cell by clicking in it. You can save the changes, create a new employee, or delete the currently selected employee using the buttons that are provided below.

The entire UI is written in EGL Rich UI, and no JavaScript, HTML, or dojo is needed.

Username	First name	Last name	Location	Phone Number
Chris	Chris	Laffra	East	555-3345
Saul	Ben	Margolis	East	555-9932
		dhajkdhaksj	West	555-3321
		Ramsey	West	555-3321



Analyzer Details

elecsys 2010
RECALLS
All tests...

Result Specifics

Result Date	06/28/2002
Result Time	74700
Material	CO2/ALC/NH
Level	1
Origination	MANUAL

Results for this test

Result	Low	High	Mean	Std.Dev.
92.0000	72.8000	95.2000	84.0000	5.6000

Lot Results

Lot 360440 contains 16 tests

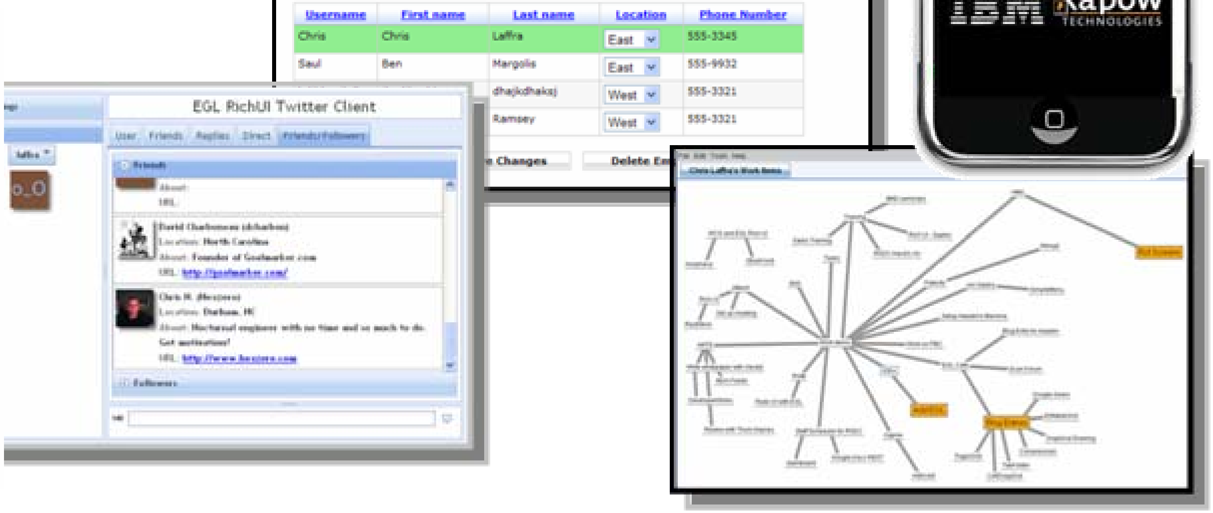
See Westgard Rules explanation

Corrective Action

Date	06/28/2002
Time	60602
Tech	MLJ
Action	QC ACCEPTABLE,WITHIN MFG LIMIT
Comment	

Archive Action

Date	06/06/2002
Time	74645
Technician	JAJ



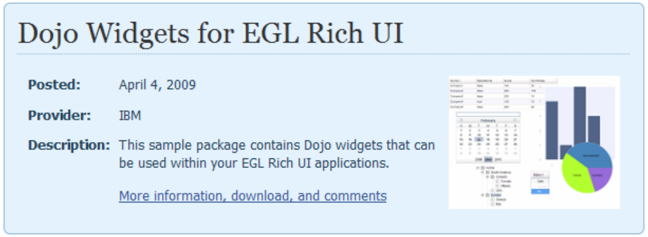
EGL RichUI Twitter Client

Why EGL Rich UI?

- **Web 2.0 / RIA development is complicated**
  - No standard tools or frameworks in place – “everybody is building a RIA framework”
  - JavaScript is notoriously tedious to code
  - Technology has not fundamentally changed – people are doing unnatural things
- **IT organizations understand the difficulties in building true multi-tier, cross-platform applications**
  - Difficult to keep up with technology
- **Most solutions today are hybrids of existing Web 1.0 technologies**
  - PHP RIA support is a server-side solution that follows the same Web 1.0 pattern
- **EGL takes a dramatically different approach to Web 2.0 / RIA development**
  - Programming model not based on HTML and traditional Web page construction

EGL Rich UI - Open and Extensible

- **Rich UI based on Web Standards**
  - REST, SOAP, JSON, OpenAjax, Dojo, JavaScript, etc.
- **Use available frameworks/runtimes**
  - WebSphere IBM i, CICS, IMS, Apache Tomcat
  - JEE, JSF, TUI, BIRT, etc.
- **UI Libraries at the EGL Café**
  - Download third-party libraries
  - Write your own and upload them
  - Import into the visual editor palette
- **Rich UI built 100% using standard EGL mechanisms**
  - External types
  - Annotations
- **Complete extensible (you are never blocked from doing what you need to do)**
- **Plans for open implementation**
  - Allow third parties to extend EGL, develop their own version



Dojo Widgets for EGL Rich UI

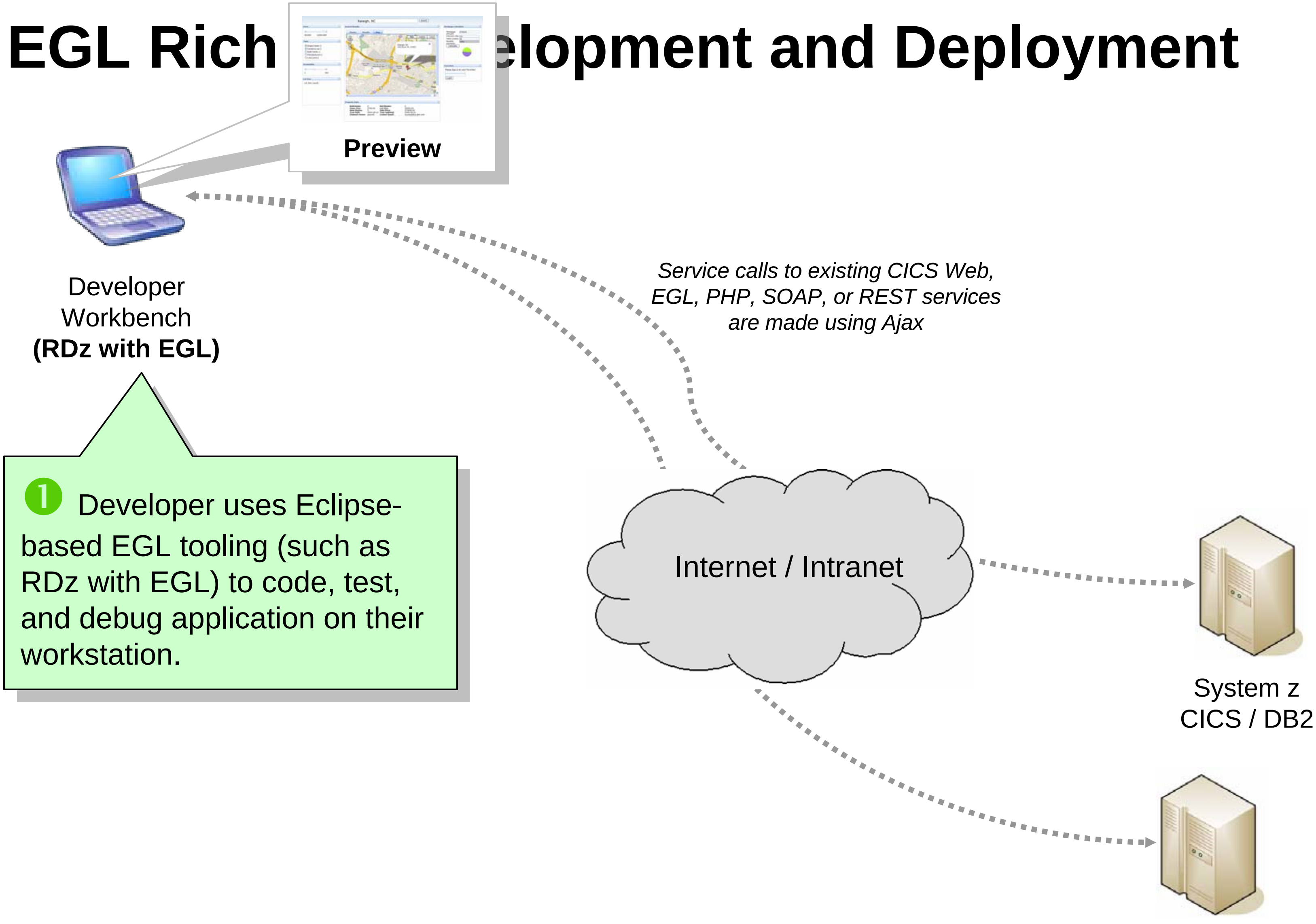
Posted: April 4, 2009

Provider: IBM

Description: This sample package contains Dojo widgets that can be used within your EGL Rich UI applications.

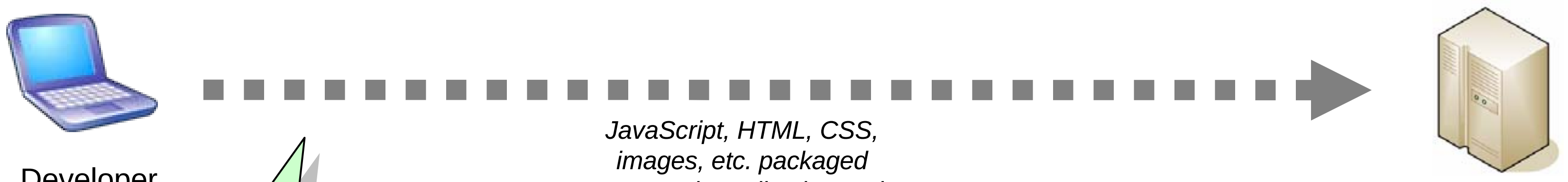
More information, download, and comments

EGL Rich elopment and Deployment



➊ Developer uses Eclipse-based EGL tooling (such as RDz with EGL) to code, test, and debug application on their workstation.

EGL Rich UI Development and Deployment



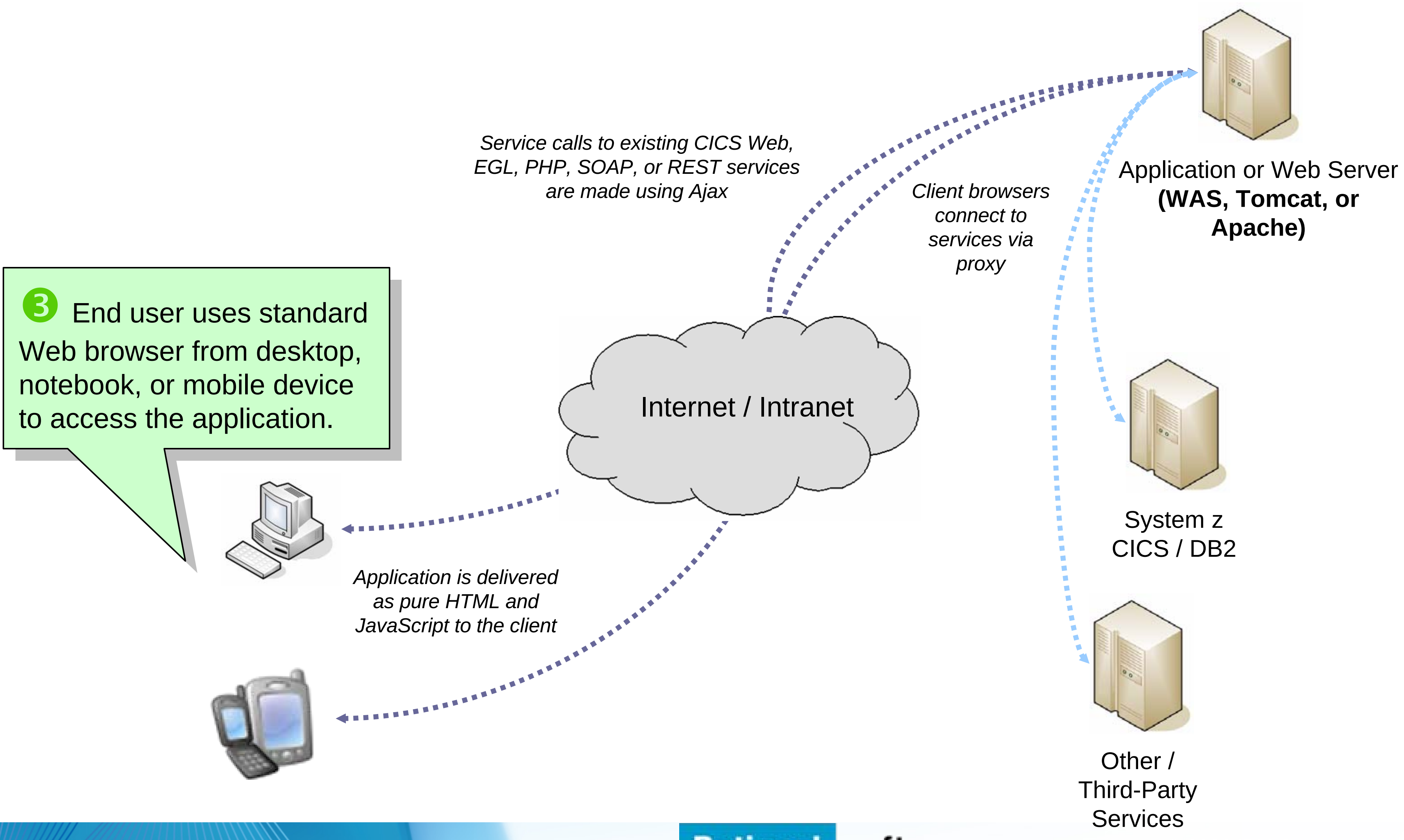
Developer
Workbench
(RDz with EGL)

JavaScript, HTML, CSS, images, etc. packaged as a Web application and deployed on the server

Application or Web Server
(WAS, Tomcat, or Apache)

❷ Developer runs Deployment wizard to create a Web application containing HTML and JavaScript (generated from EGL code). Application is deployed to an application server.

EGL Rich UI Development and Deployment



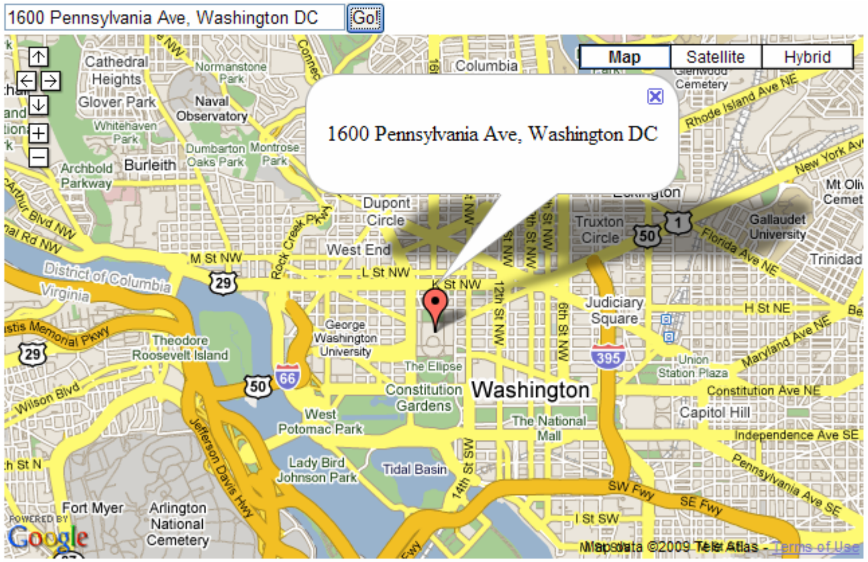
❸ End user uses standard Web browser from desktop, notebook, or mobile device to access the application.

Service calls to existing CICS Web, EGL, PHP, SOAP, or REST services are made using Ajax
Client browsers connect to services via proxy
Application or Web Server **(WAS, Tomcat, or Apache)**
Internet / Intranet
System z CICS / DB2
Application is delivered as pure HTML and JavaScript to the client
Other / Third-Party Services

EGL in Action (Side-by-Side Comparison)

EGL Rich UI

```
handler MyRuiHandler type RUIhandler { initialUI = [ addressForm,
map ] }

 addressField TextField { text = "1600 Pennsylvania Ave, Washington
DC", width = 250 };

 goButton Button { text = "Go!", onClick ::= goButton_clicked };
 addressForm Box { children = [ addressField, goButton ] };

 map GoogleMap { width = "500px", height = "300px" };

  function goButton_clicked (e Event in)
    addresses String[] = [ addressField.text ];
    map.showAddresses(addresses, addresses);
  end
end
```



1600 Pennsylvania Ave, Washington DC

HTML and JavaScript

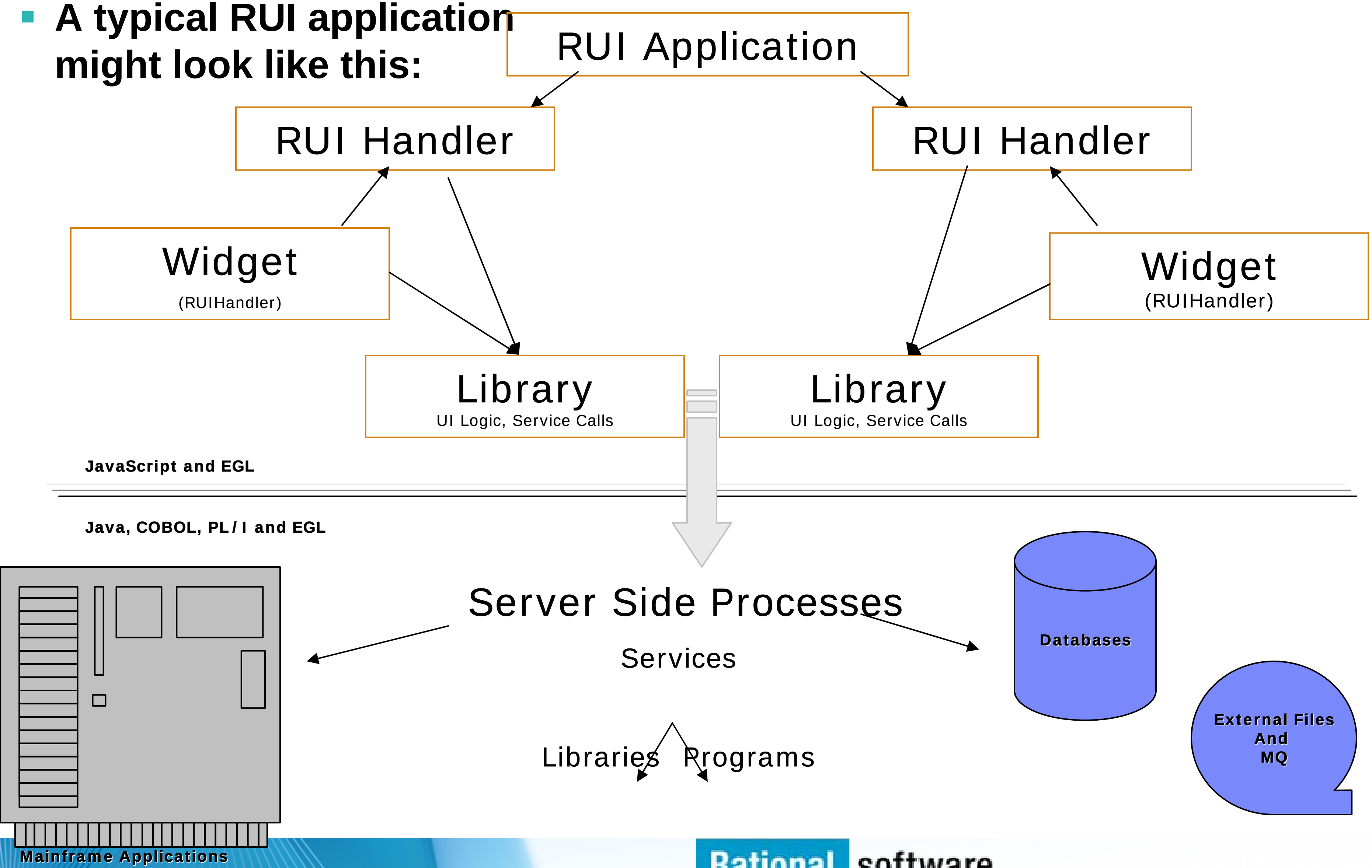
```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
com:vml">
  <head>
    <meta http-                                    html; charset=UTF-8"/>
    <title>Goog                                    ng</title>
    <script src                                    api&amp;v=2.x
    <script typ


    var geocode

    function initialize() {
      if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(37.4419, -122.1419), 13);
        geocoder =
      }



      if (g
        geocoder.
          address
          functio
            map.s
            var m
            map.a
            marke
          }
        );
      }
    }
    </script>
  </head>

  <body onload="initialize()" onunload="GUnload()">
    <form action="#" onsubmit="showAddress(this.address.value); return
false">
      <p>
        <input type="text" size="60" name="address" value="1600 Pennsylvania
Ave, Washington DC" />
        <input type="submit" value="Go!" />
      </p>
      <div id="map_canvas" style="width: 500px; height: 300px"></div>
    </form>
  </body>
</html>
```

All code, including UI and controller logic, is written completely in EGL.

The complexity of the Google Map APIs are hidden from the developer, so the developer can focus on the actual business requirement and not technical complexities.

RUI Programming – Overview

- **A typical RUI application might look like this:**



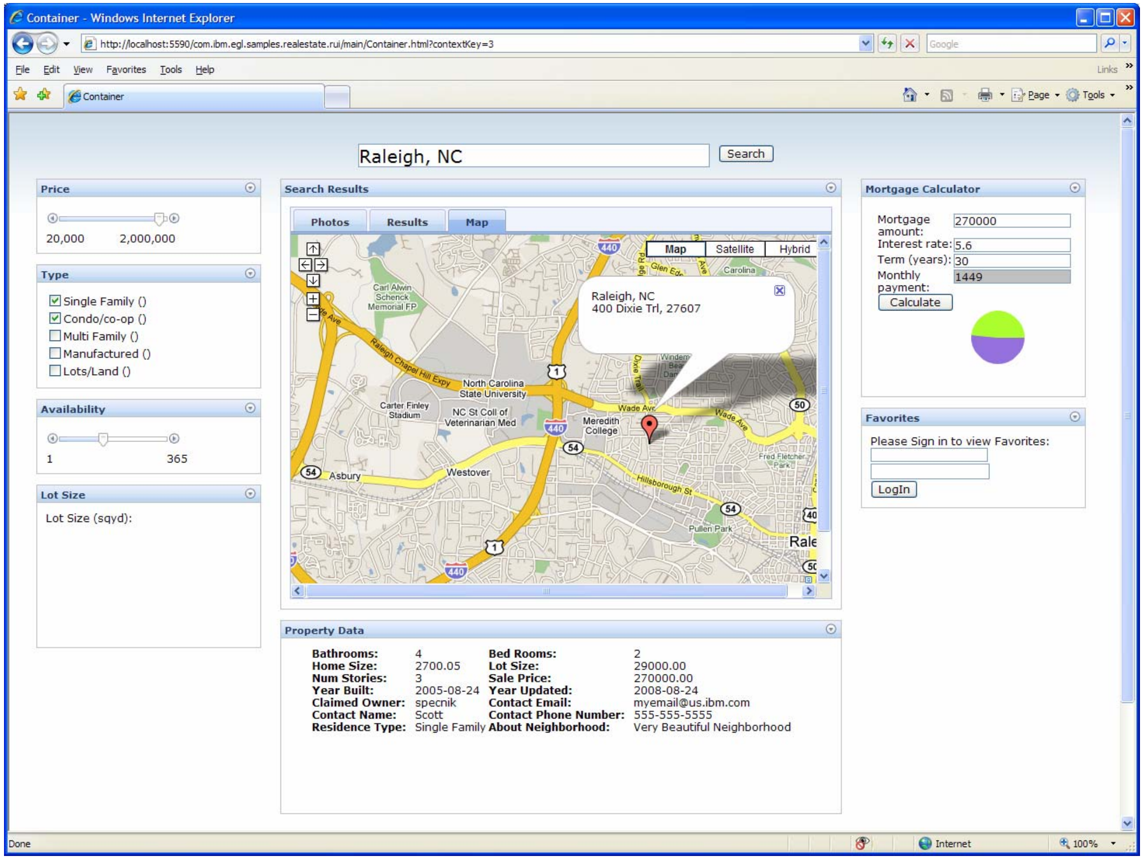
Modern Application Development Featuring Web 2.0 for System z

Real Estate Demo Application Overview

Rational. software

Overview

- We will be working with a sample real estate application that has been created for this seminar. This application shows how existing data and applications can be easily exposed in a new Web 2.0-style application.



Container - Windows Internet Explorer

http://localhost:5590/com.ibm.egl.samples.realestate.rui/main/Container.html?contextKey=3

Raleigh, NC

Price

20,000 2,000,000

Type

- ☑ Single Family ()
- ☑ Condo/co-op ()
- □ Multi Family ()
- □ Manufactured ()
- □ Lots/Land ()

Availability

1 365

Lot Size

Lot Size (sqyd):

Search Results

Raleigh, NC
400 Dixie Trl, 27607

Mortgage Calculator

Mortgage amount:	270000
Interest rate:	5.6
Term (years):	30
Monthly payment:	1449

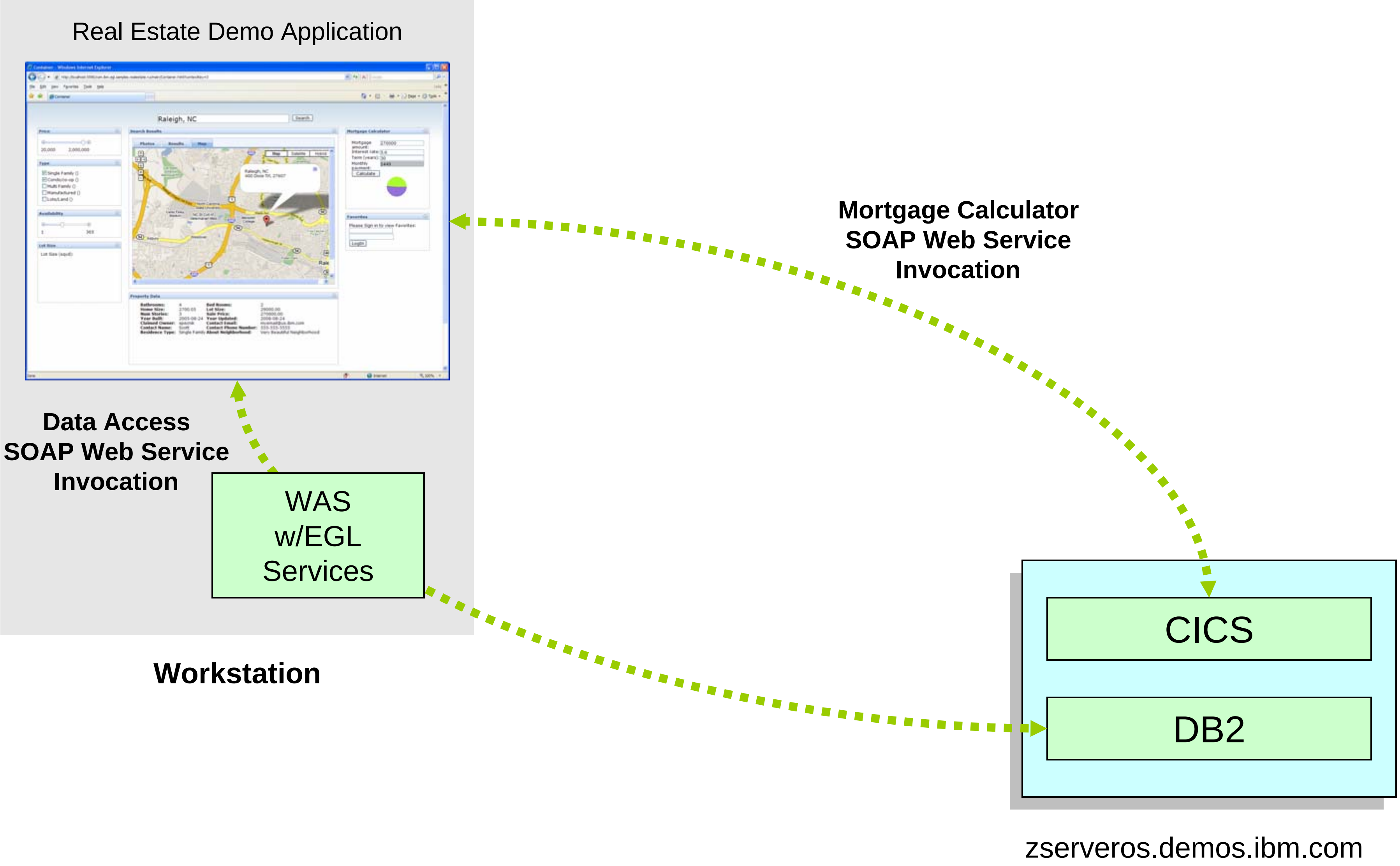
Favorites

Please Sign in to view Favorites:

Property Data

Bathrooms:	4	**Bed Rooms:**	2
Home Size:	2700.05	**Lot Size:**	29000.00
Num Stories:	3	**Sale Price:**	270000.00
Year Built:	2005-08-24	**Year Updated:**	2008-08-24
Claimed Owner:	specnik	**Contact Email:**	myemail@us.ibm.com
Contact Name:	Scott	**Contact Phone Number:**	555-555-5555
Residence Type:	Single Family	**About Neighborhood:**	Very Beautiful Neighborhood

Architecture Overview

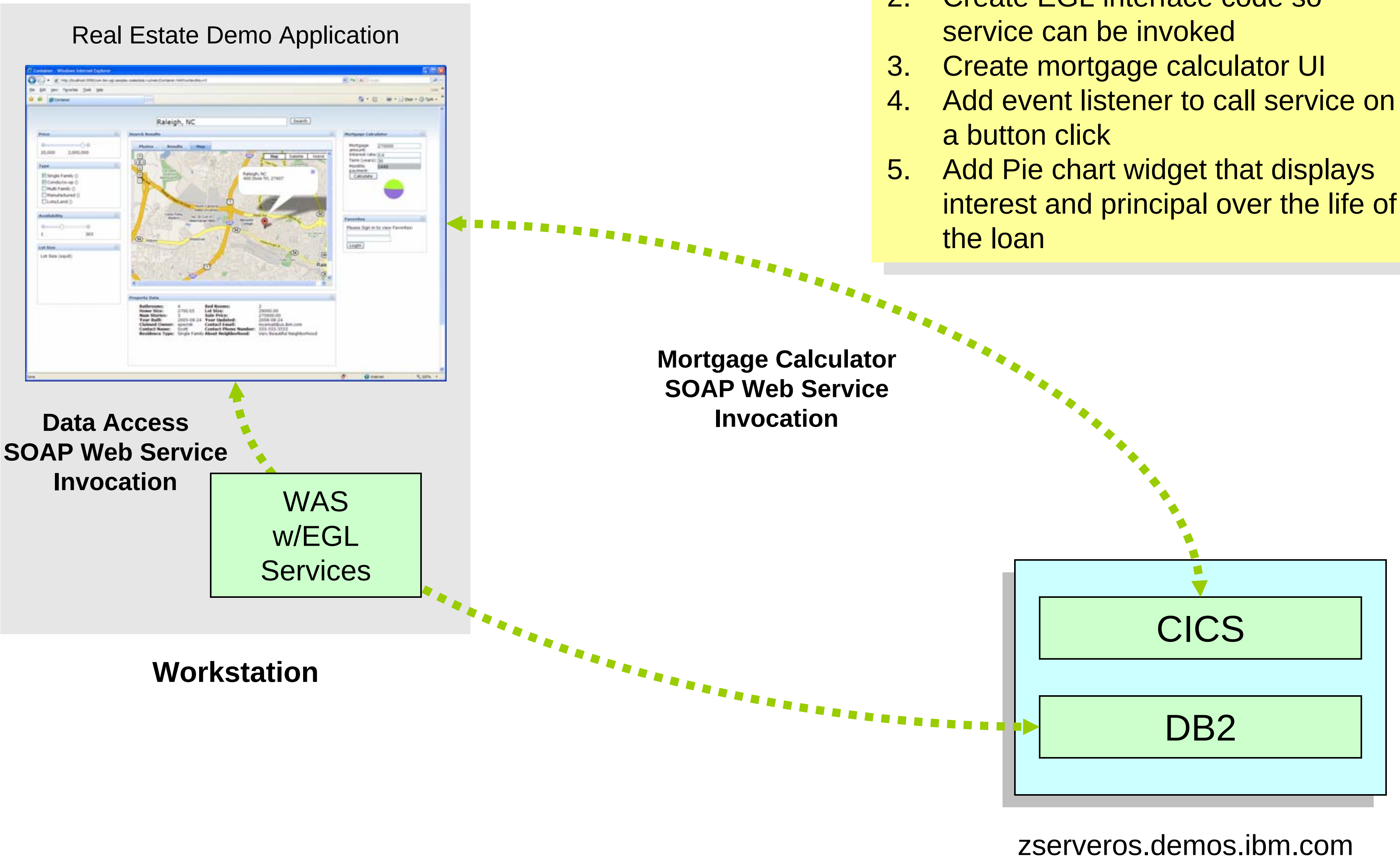


Real Estate Demo Application

Demo Tasks

Task 1 - Adding Mortgage Calculator

1. Consume CICS Web service in EGL Rich UI project
2. Create EGL interface code so service can be invoked
3. Create mortgage calculator UI
4. Add event listener to call service on a button click
5. Add Pie chart widget that displays interest and principal over the life of the loan

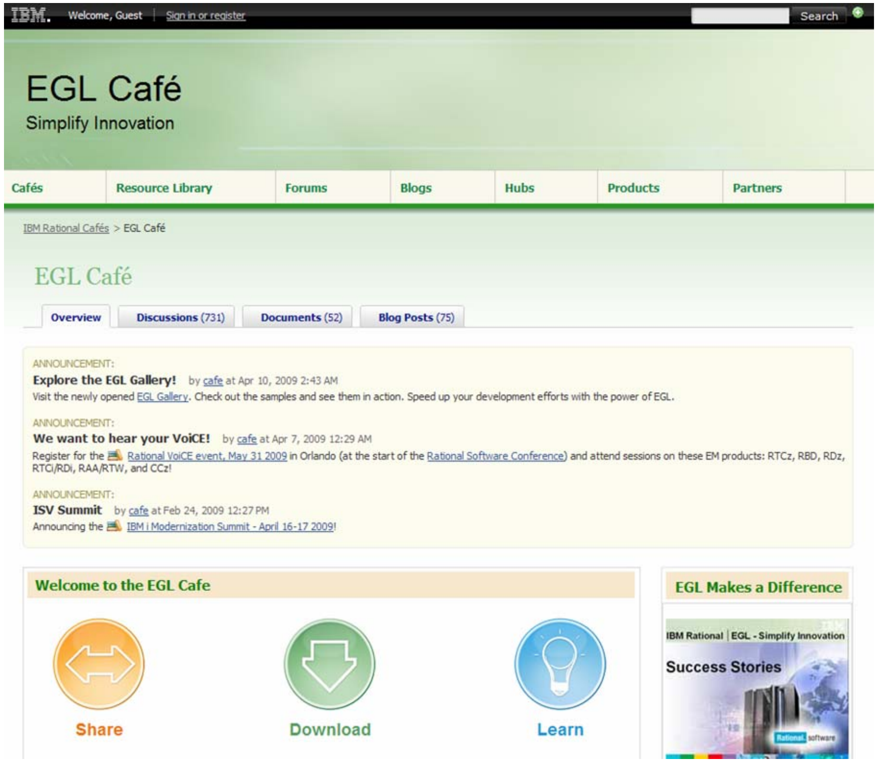

Real Estate Demo Application
Raleigh, NC
Mortgage Calculator
Data Access SOAP Web Service Invocation
WAS w/EGL Services
Workstation
Mortgage Calculator SOAP Web Service Invocation
CICS
DB2
zserveros.demos.ibm.com

EGL Rich UI

Mortgage Calculator, Data Table, and Google Map

EGL Café

- Online community for EGL developers, partners, and clients
- Discussion forums
- Gallery of sample applications and widgets
- Presentations, videos, and articles
- Blogs by IBMers and partners
- Success stories
- Become part of the community today!
  http://ibm.com/rational/eglcafe



ANNOUNCEMENT:
Explore the EGL Gallery! by cafe at Apr 10, 2009 2:43 AM
Visit the newly opened EGL Gallery. Check out the samples and see them in action. Speed up your development efforts with the power of EGL.

ANNOUNCEMENT:
We want to hear your VoiCE! by cafe at Apr 7, 2009 12:29 AM
Register for the Rational VoiCE event, May 31 2009 in Orlando (at the start of the Rational Software Conference) and attend sessions on these EM products: RTCz, RBD, RDz, RTCi/RDi, RAA/RTW, and CCz!

ANNOUNCEMENT:
ISV Summit by cafe at Feb 24, 2009 12:27 PM
Announcing the IBM i Modernization Summit - April 16-17 2009!

Real Estate Demo

Posted: April 9, 2009
Provider: IBM
Description: Sample real estate search Web application built with EGL Rich UI. Shows SOAP, REST, database access, drag-and-drop, Dojo widgets, and more.
More information and comments
Try Now!

Dojo Widgets for EGL Rich UI

Posted: April 4, 2009
Provider: IBM
Description: This sample package contains Dojo widgets that can be used within your EGL Rich UI applications.
More information, download, and comments

Resources: Download, Learn, Presentations, Video/viewlet, Sample Code
Community: Clients, Partners, Influencers, Press, News and Events
Collaboration: Blogs, Forums, Tips and Techniques Comments, Ratings
Testimonials: Case Studies, Celebrations!

EGL Distance Learning 2009

- Learn EGL in 2 weeks with IBM's no charge (free) remote course.
- Explore the emerging cross-platform rapid development environment from IBM that enables you to build Web, Web 2.0, and SOA solutions using EGL.
- Discover how quickly you can learn EGL and build state-of-the-art applications, all without getting bogged down in the technical complexities of middleware and runtime platforms.
- **Week 1:**
  - Foundations of EGL technical workshop
- **Week 2:**
  - Web 2.0 development with EGL Rich UI technical workshop

2009 Dates

- May 4
- June 15
- July 27 (part 2 only)
- September 14
- November 30

2 hours of instruction per day

3 - 5 hours of independent hands-on per day

Register now!

http://www.ibm.com/software/rational/cafe/docs/DOC-3004

Learn more at: IBM Enterprise Modernization Solution
(http://www.ibm.com/software/info/developer/solutions/em/)

- IBM Rational software
(http://www.ibm.com/software/rational/)
- IBM Rational Software Delivery Platform
(http://www.ibm.com/software/info/developer/index.html)
- Process and portfolio management
(http://www.ibm.com/software/rational/offerings/ppm/)
- Change and release management
(http://www.ibm.com/software/rational/offerings/crm/)

Other links for information and learning

Quality management
(http://www.ibm.com/software/rational/offerings/quality/)

Architecture management
(http://www.ibm.com/software/rational/offerings/architecture/)

Rational trial downloads
(http://www.ibm.com/developerworks/rational/downloads/?S_TACT=105AGX23&S_CMP=RCD)

Leading Innovation Web site
(http://www.ibm.com/software/rational/leadership/leaders/)

developerWorks Rational (http://www.ibm.com/developerworks/rational)

IBM Rational Business Partners
(http://www.ibm.com/software/rational/partners/)