# Modern Application Development Featuring Web 2.0 for System z

Michelle A. Cordes
Rational for System z Ecosystem Team
mcordes@us.ibm.com

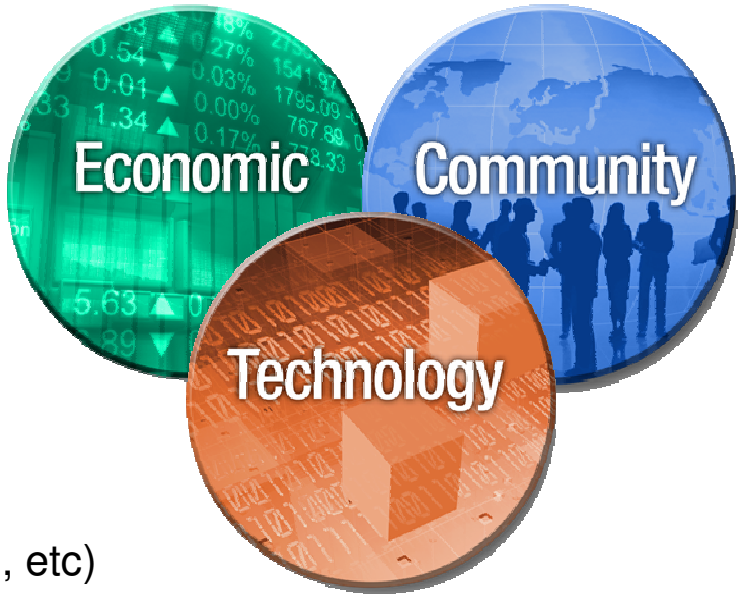
© 2009 IBM Corporation

# Agenda

- What is Web 2.0?
- Why Web 2.0 for the mainframe?
- Using IBM tooling to easily create and integrate Web 2.0 applications with mainframe applications
- Summary
- Q/A

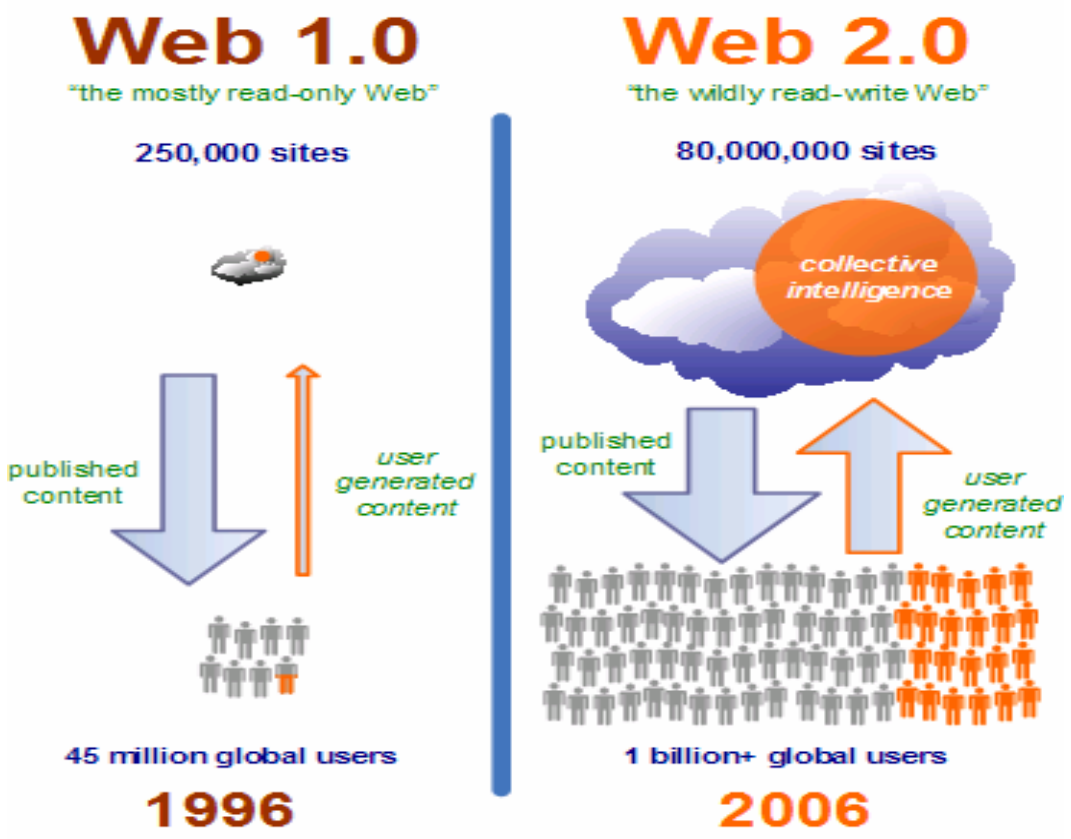
# What is Web 2.0?

- **An important trend in delivering software applications**
- **An enabler for richer web applications**
  - New business models
  - Peer-to-peer user participation
  - New technologies
  - Interactive filtering, presentation, data entry




- **A combination of core technology components**
  - Rich user experience (maps, grids, animation, D&D, etc)
  - Loose-coupling, composite applications via reuse and “mash-ups”
  - Standards (SOAP, REST, JSON, Atom, etc)

# How does Web 2.0 compare Web 1.0 (the original)?

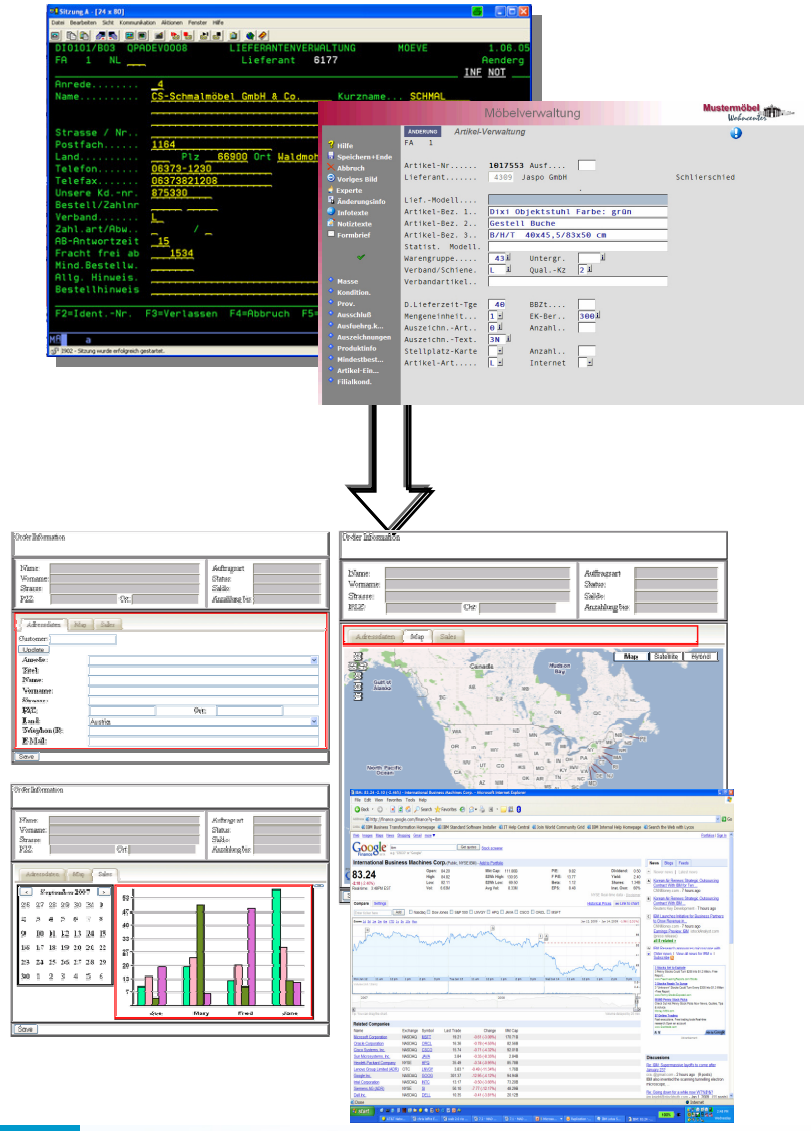



Source: http://hinchcliffe.org/img/web1vsweb2.png

| Web 1.0 | Web 2.0 |
|---|---|
| Reading | Writing |
| Companies | Communities |
| Client/Server | Peer-to-peer |
| HTML | XML |
| Home pages | Blogs |
| Britannica Online | Wikipedia |
| Lectures | Conversations |
| Ofoto | Flickr |
| Taxonomy | Folksonomy |

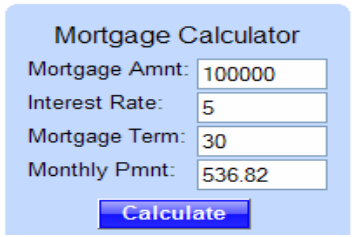
# Rich Internet Applications (RIA)

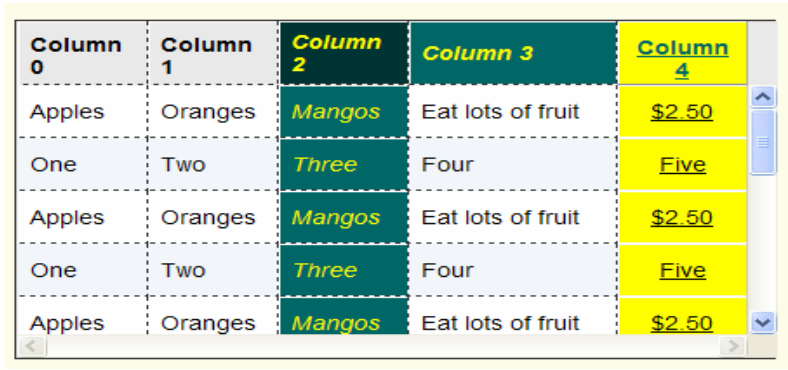
- **RIAs combine some of the best features from both traditional thin- and thick-clients**
- **RIAs are enabled by**
  - Enhancements in browser capabilities
  - Better utilization of technologies such as AJAX and JavaScript
- **Rich Internet Applications offer**
  - Desktop-like UI functionality
  - Easy deployment
  - Portability across platform
  - Mix of enterprise data and internet services
- **Examples of “really-rich” RIAs**
  - Google Maps/Gmail/Google Finance
  - Flickr, Facebook



# Terms and Concepts

- **AJAX** – Stands for Asynchronous JavaScript and XML..
- **Widget** – A widget is a graphical unit in a GUI or Internet style interface. Most widgets allow for the interaction and manipulation of data in the browser.



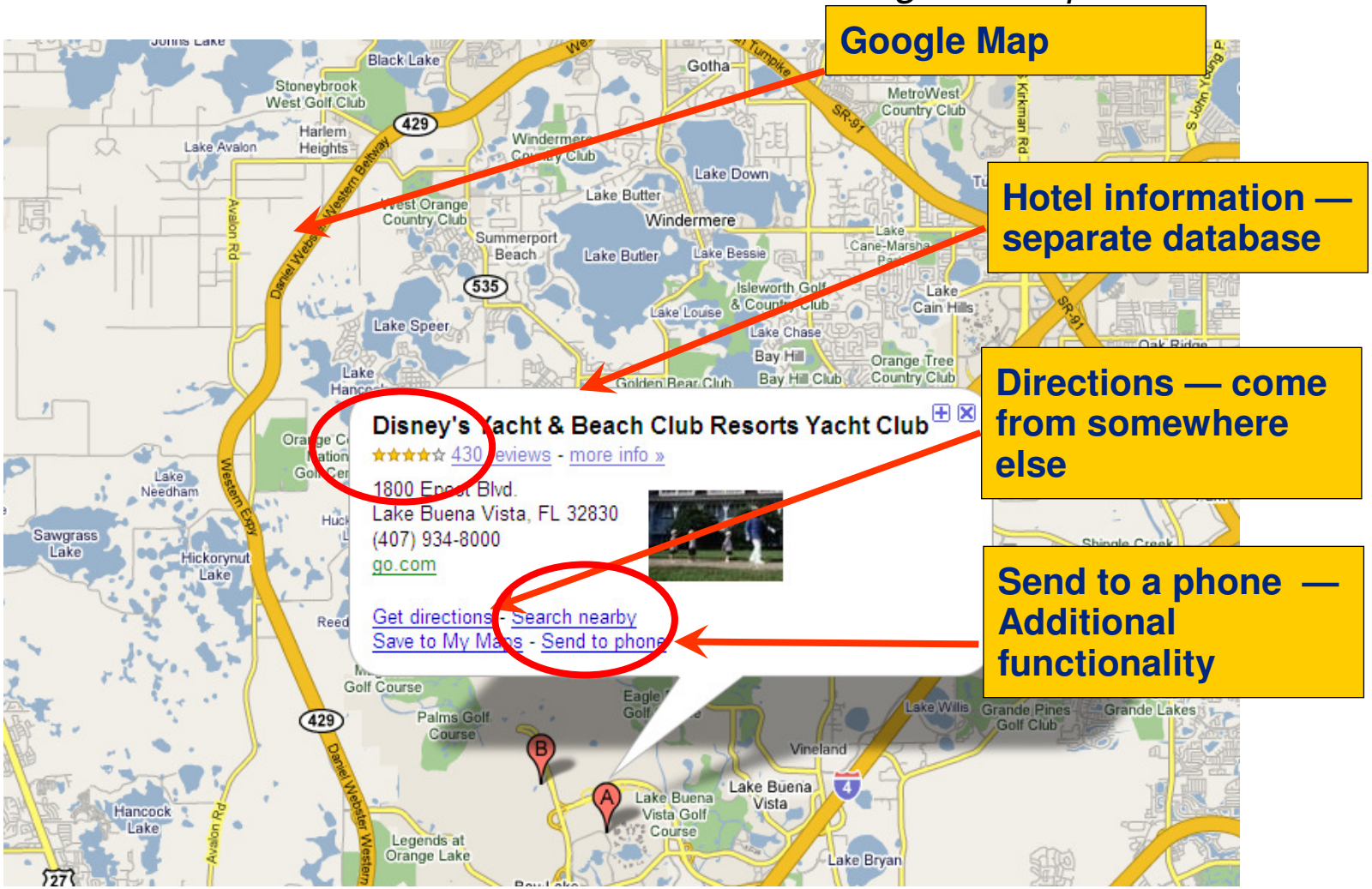



| Column 0 | Column 1 | Column 2 | Column 3 | Column 4 |
|---|---|---|---|---|
| Apples | Oranges | Mangos | Eat lots of fruit | $2.50 |
| One | Two | Three | Four | Five |
| Apples | Oranges | Mangos | Eat lots of fruit | $2.50 |
| One | Two | Three | Four | Five |
| Apples | Oranges | Mangos | Eat lots of fruit | $2.50 |

- **DOJO** – An open source JavaScript toolkit. The DOJO project sets out to create widgets using only JavaScript.

# Mashups

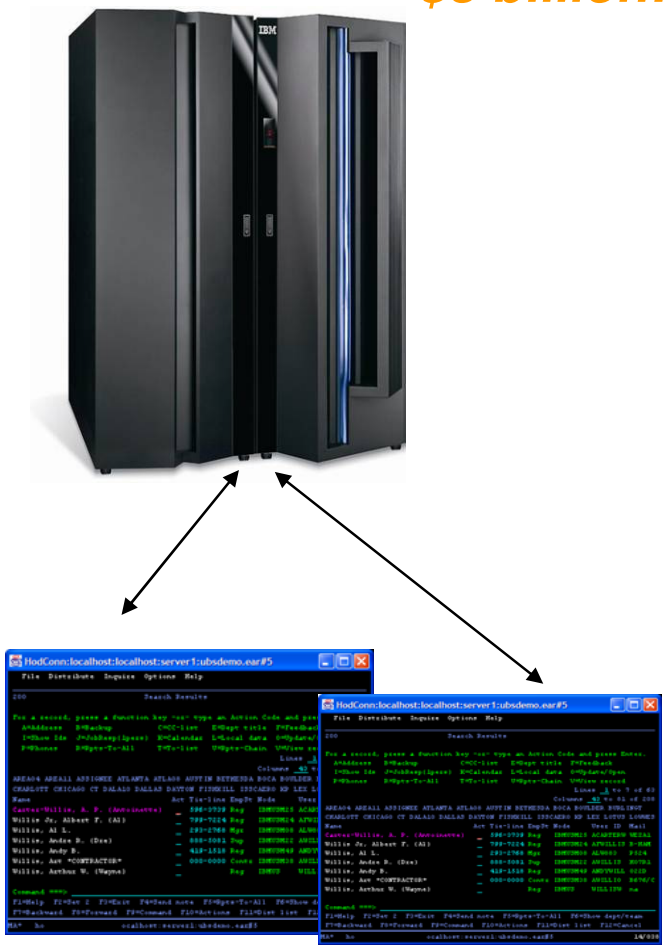
*Combine content from more than one source into an integrated experience*



# Value of Existing Applications

***30 million lines of code could be worth $3 billion!***

- **Existing applications are among the most valuable assets a company owns**
  - Fully functional applications that run the business today
  - Significant investments over the years
  - Fine-tuned for dependable optimal performance
  - Run within fine-tuned reliable, robust, scalable IT infrastructure and platforms
- **Accessed via character / text based terminals**
  - Non-intuitive, difficult to navigate user interfaces
  - Difficult to integrate in modern business processes
- **5X less expensive to reuse existing applications than to write new applications from scratch**
- **Reusing proven, time-tested applications can result in significantly lower risk and faster time to market**



# Why Web 2.0 for the Enterprise?

- **Web 2.0 is**
  - Creating new markets
  - Lowering competitive barriers
  - Encouraging creativity to come from anywhere
  - Harnessing community created media
  - Tapping into the wisdom of the crowds
  - Enhancing communications and making information more impactful
- **What does it means for the enterprise?**
  - Greater collaboration and innovation across the value chain
  - Increased levels of customer intimacy
  - Simplification of complex IT and business infrastructure
  - Business model flexibility to capitalize on new market opportunities

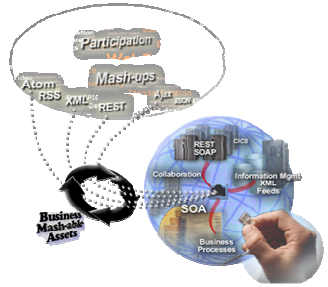
***It's fast becoming a Web 2.0 business world, but innovation never goes out of style.***

**According to Gartner:**

By 2008, the majority of Global 1000 companies will have adopted several technology-related aspects of Web 2.0

Missing out on the non-technology aspects of Web 2.0 means that many organizations will also miss out on some of the positive business benefits

# New Business Opportunities



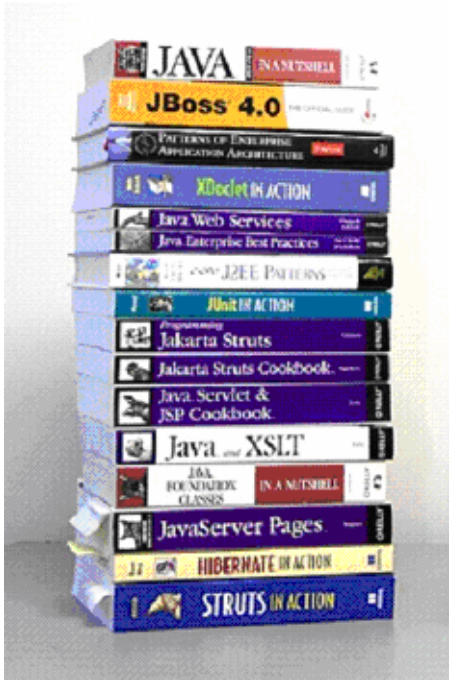

- **For example…**
  - Leverage Web 2.0 in your business critical applications
  - Investigate customization, integration and services opportunities in rolling out web 2.0-based community collaboration environments.
- **Benefits to your customers:**
  - Instant business value through rapid new product deployment, with reduced costs
  - Reduction in time to market opens new business opportunities
  - Reducing costs through customer self-service
  - Unlocking valuable business data opens new business opportunities
  - Extracting value by collecting and sharing the wisdom in the organization

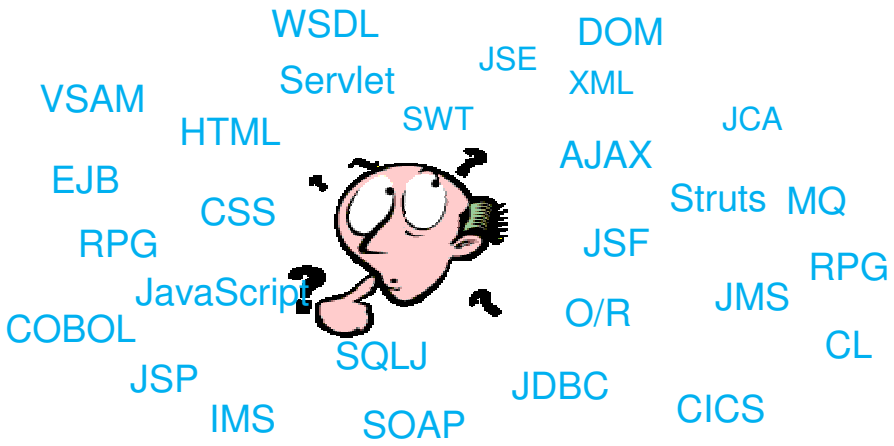
# Modernization with Web 2.0

*Reuse what you already have!*

- Build a new Web front-end for your existing order entry application, but continue to use your proven business logic.
- Create a self-service portal that provides your customers with details about past and business logic to perform a calculation.
- Expose a portion of current orders.
- Mashup information from multiple CRM systems.
- Reuse existing COBOL customer data locked away in a green screen application to another team in your organization.
- Web 2.0 is not just for Internet-accessible applications – can be used to improve productivity of internal users!

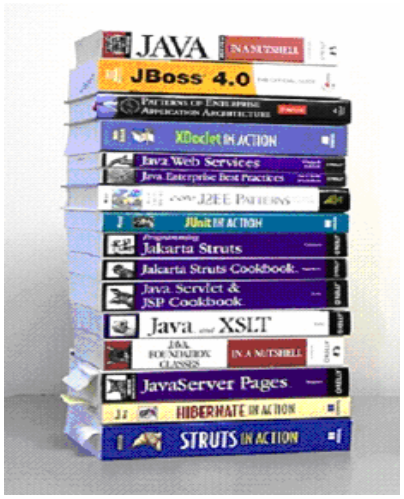
# Web 2.0 sounds pretty good, eh? But…it is hard!

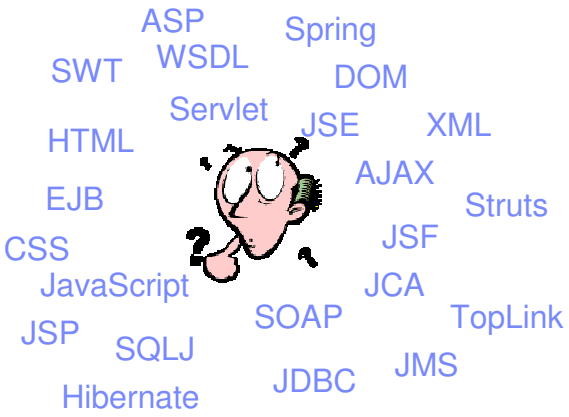



- **Technical Challenges:**
  - Developers must learn numerous different technologies
  - Cross-platform programming require totally different skills
  - Must keep up with constantly changing Web 2.0 technologies
- **Organizational Challenges**
  - Data owned by other organizations
  - Information is “hidden” in a website

# Why EGL? Because building applications today is not easy



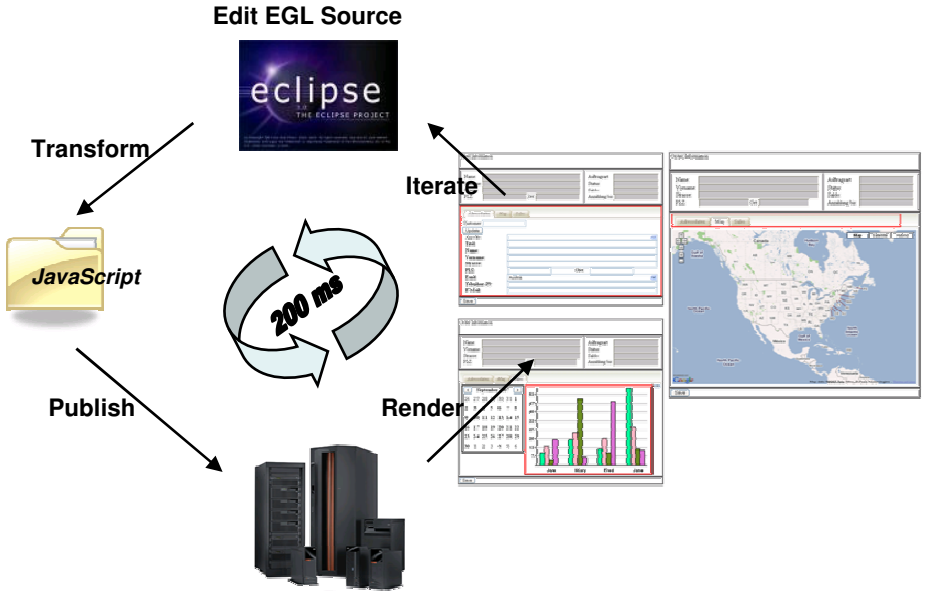
- Learn and master myriads of
  - programming languages and semantics
  - middleware interfaces
  - programming paradigms and styles
  - frameworks and libraries
- Constantly emerging new technologies

- 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
- EGL Rich UI supports creation of RIAs by providing:
  - Simple programming model
  - Productive development tools
  - Comprehensive widget library
  - Web services and Mash-ups
  - Extensibility

# EGL Rich UI – Web 2.0 made easy!

## EGL Rich UI

- Includes visual composition, libraries of RUI widgets
- Instant deploy/visualize while you code
- Fully open and extensible
- Easily integrate/consume any service (REST or SOAP)
- Single language end-to-end
- Brings legacy data and assets to Web 2.0



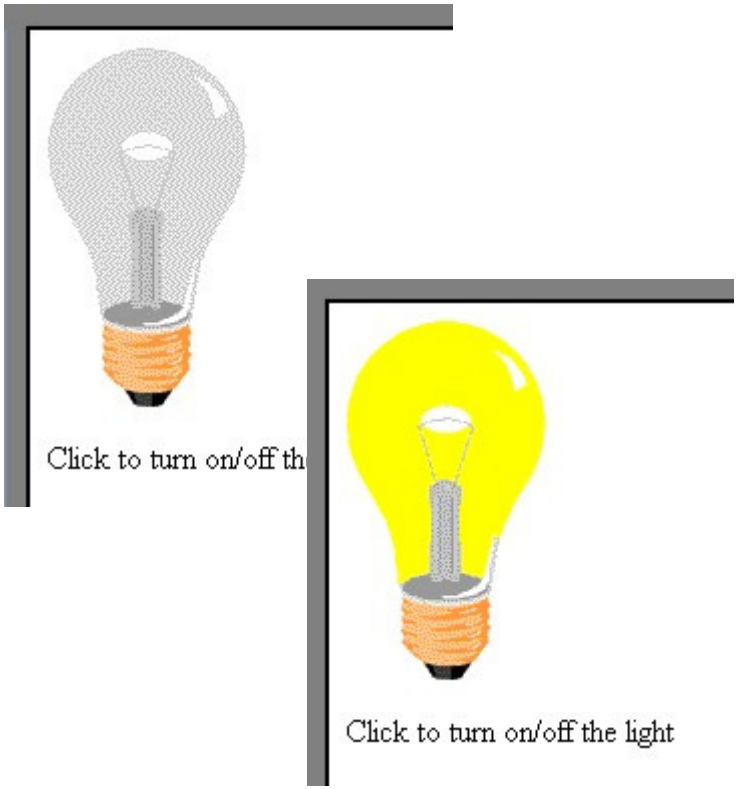
# RichUI and Client Processing

- **RichUI leverages the generation capabilities of EGL; JavaScript is the runtime**
  - JavaScript is a language that can only be run in a browser, (FireFox, Internet Explorer, Opera, etc.) – not on the server like EGL-generated COBOL or Java
- **It is JavaScript that renders your page in the browser and manipulates labels, data, graphics and controls the page's behavior**
  - No static HTML is created
  - JavaScript does all the work
- **Much of the U.I. is also implemented using leading-edge internet technologies such as:**
  - AJAX
  - DOJO
  - JSON
  - FLEX
  - Web Services

# JavaScript

- **JavaScript is a scripting language for the web**
- **JavaScript is a script, not Java**
- **Small bits of JavaScript interact with HTML and CSS**
  - provides interactivity
  - allows dynamic changes
  - completely within the browser

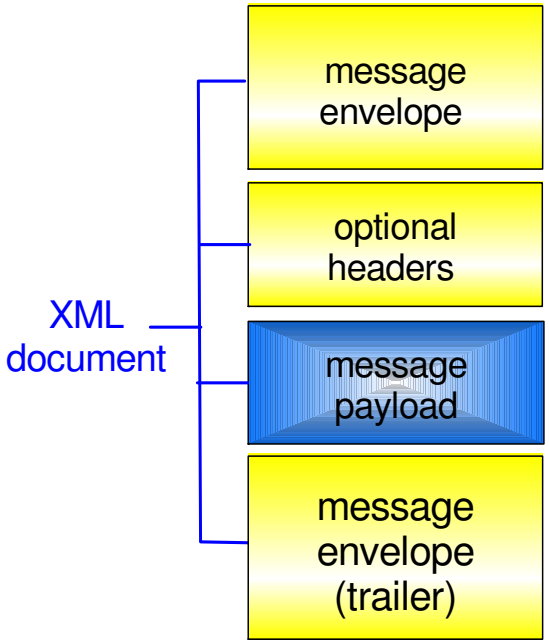
```
function changeimage() {
  if (cc==0){
    cc=1;
    document.getElementById('myimage').src="bulbon.gif";
  } else {
    cc=0;
    document.getElementById('myimage').src="bulboff.gif";
  }
}
```



# SOA and Web Services

- Rich UI makes extensive use of services, and SOA – Service Oriented Architecture, which is a way to modularize and deploy code so that it can be consumed anywhere in the world using any language.
- There are two types of Web Service calls used by Rich UI

## SOAP service calls




## RESTful service calls

- **Request…**

```
GET /mortgage/231677 HTTP/1.1
  Host: www.example.com
  Accept-Language: en
  Charset: UTF-8
```

- **Response…**

```
HTTP/1.1 200 OK
  Language: en_us
  Charset: UTF-8
<mortgage>
   <principle>238000</principle>
   <rate>3.5</rate>
   <type>5/1 ARM</type>
</mortgage>
```

- By utilizing web services, one can create a truly scalable system.

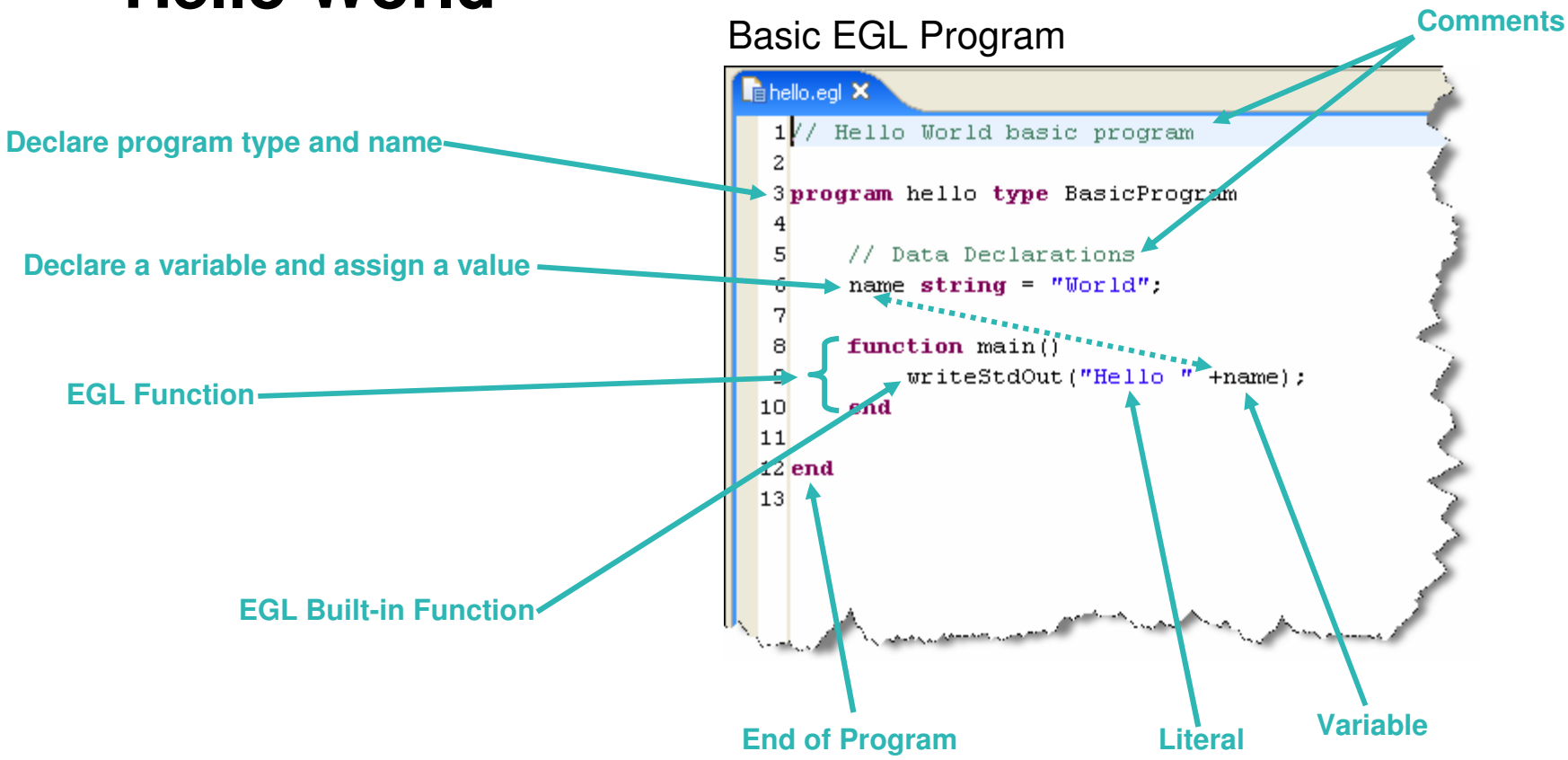
# Simplicity and Flexibility offered by EGL Rich UI

- **Data Access**
  - Create new data manipulation
    - Inspect a database and import a record
    - Tooling will generate CRUD operations
  - Use existing programs that interface data via s simple CALL
- **Services**
  - Simple declaration (using “service” keyword offered by EGL)
  - Easy deployment (WSDL and Java is generated)
  - End-to-end is especially simple
- **UIs**
  - Declarative layout, visual editor, CSS, simple event handling
  - Define/import widgets, to add to the palette

# EGL - The power of the Language

*Simple and familiar*

## Hello World



# EGL the Language

- **Rich data types**
  - Simple (int, string, boolean, etc.) or Complex (any, static arrays, dynamic arrays, dictionaries, array dictionaries, etc.)
- **Keywords**
  - Case, if-then-else, while, for loop, for loop cycling through a database result set, etc.
- **High power language capabilities**
  - Automated Casting (e.g. using AS operator)
  - Mixing data types in assignments and expressions
  - Exception handling
- **Rich libraries of built-in functions**
  - Math, string, date/time, system, etc.
- **Robust integrate with existing functions or low level APIs**
  - Call RPG, COBOL, C, etc.
  - Full Java interoperability
    - Invoke Java from EGL (map Java classes with EGL External Types)
    - Invoke EGL from Java

# 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*).

customerService.egl

```
// 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.

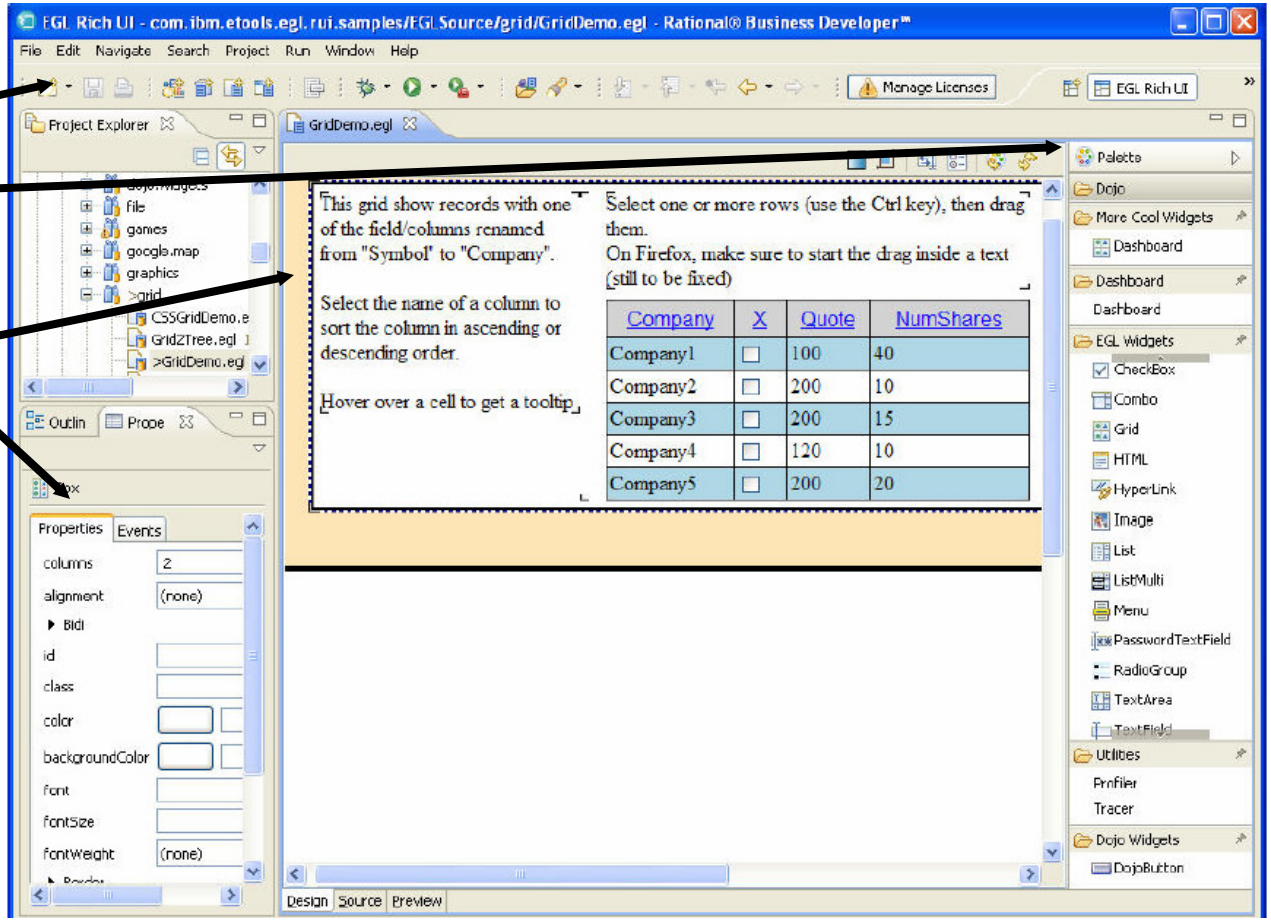
customerService.egl | creditCheck.egl

```
// interface
Interface creditCheck

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

# The power of tools: RUI Editor

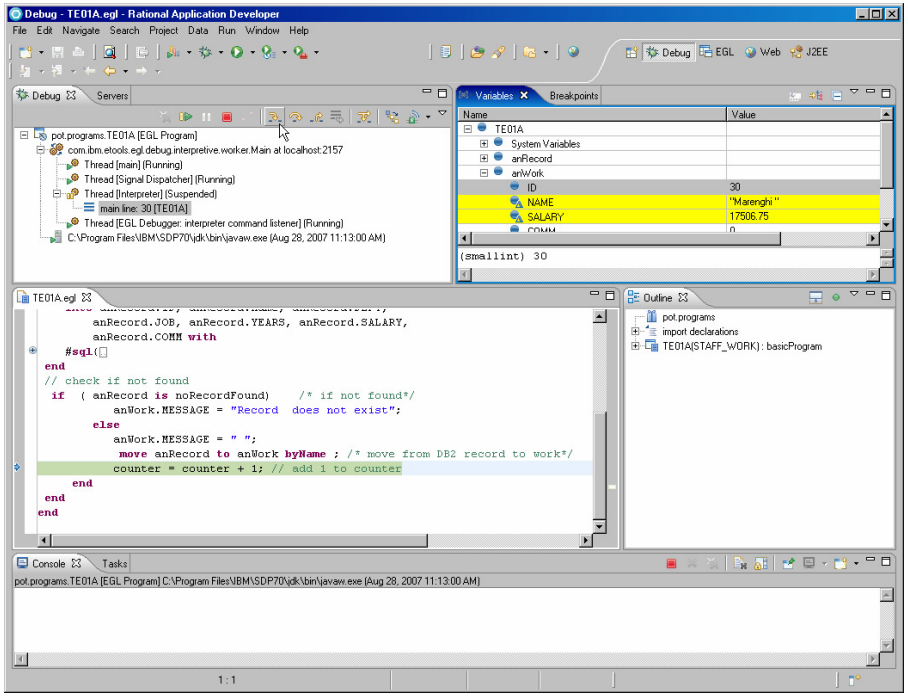
- WYSIWYG Editor
- Widget Palette
- Properties view
- Preview



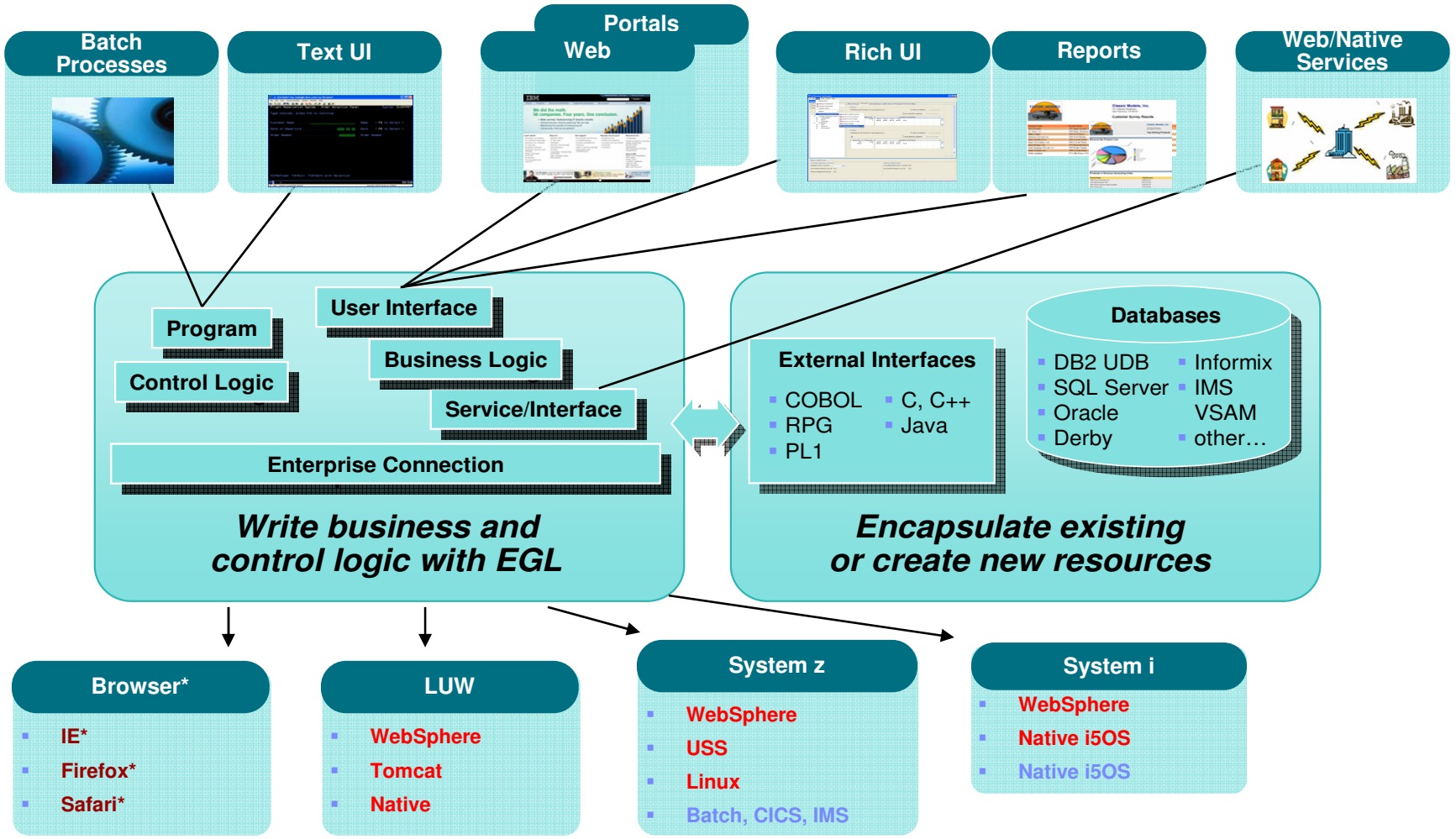
# EGL - The Power of Tools

*Integrated debug environment*

- **Debug entire application regardless of ultimate deployment targets**
  - Debug EGL, JSP, Java, etc.
- **Use features of the EGL source debugger**
  - Set breakpoints
  - Watch variables
  - Change variable values
  - Dynamic re-positioning
- **Additional EGL features**
  - Remote VSAM access

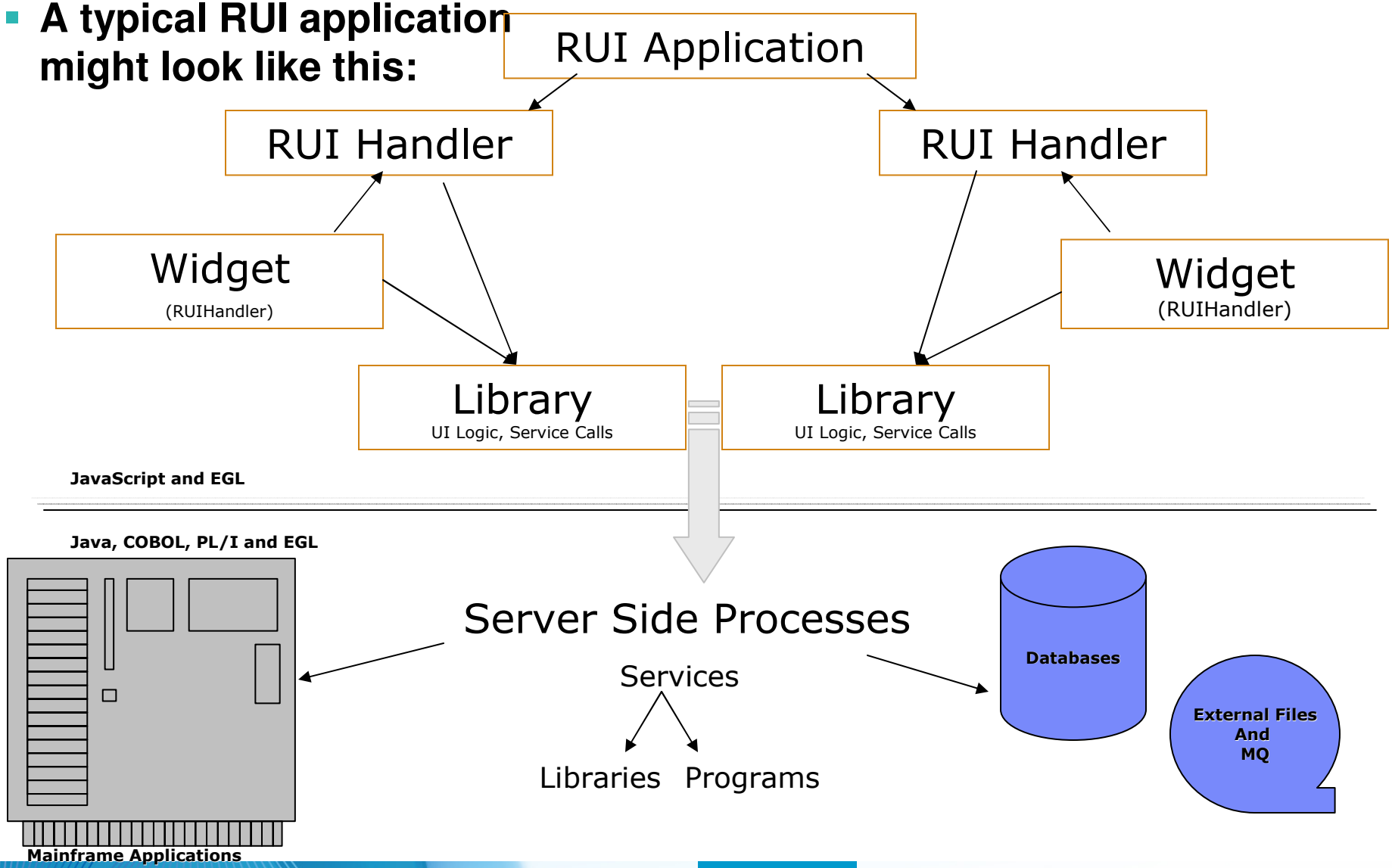


# Respond to Broadest Application Needs



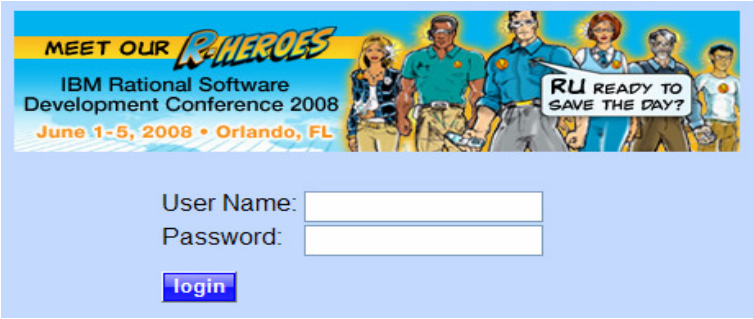
# RUI Programming – Overview

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

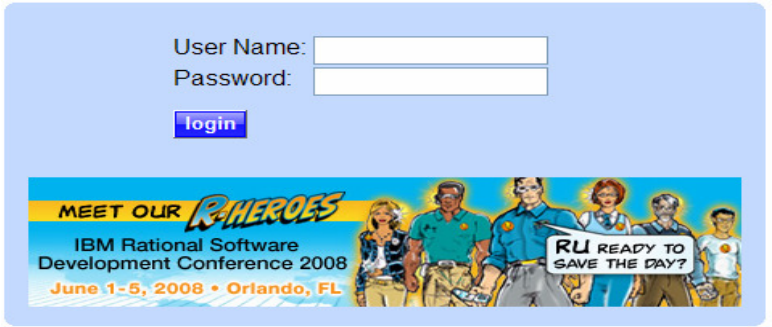


# Basic of RUI Programming - Children

- **Web pages created with RichUI are extremely flexible.**
  - For example, by simply changing the order in which children are assigned, a web page can be completely turned upside down
  - Suppose we had a page that children in this order image, loginBox, login




  - Changing the order of our children can render the following
    - [loginBox, login, image]



# Basics of RUI Programming - Widgets

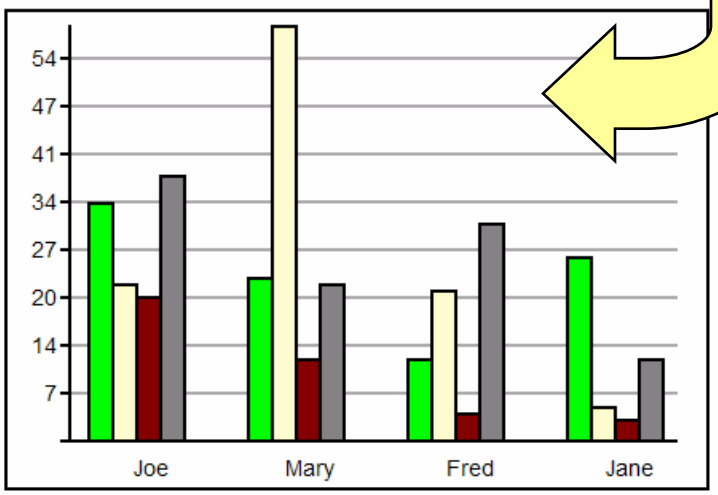
- **So what types of widgets can be used when programming in RichUI? The answer to this question is infinite!**
- **RichUI comes ships with a wide range of widgets, but also has the ability to pull in widgets from other technologies like DOJO and Silverlight. And last but not least, developers can create custom widgets using EGL RichUI!**
- **The list of widgets shipped with EGL is best found through Content Assist but here are a few:**
  - Combo Boxes
  - Radio Buttons
  - Check Boxes
  - Image Fields
  - Charts
  - Single and Multi Selection List Boxes
  - Calendars
  - This list goes on!!!

# EGL Simple Examples

## Drawing Graphics

```
Handler GraphDemo Type RUIHandler { onConstructionFunction =  myStartupFunction }
    g Graph { };

    Function myStartupFunction(parent DOMElement)
        g.init(parent);
        data int[4][4];
        data = [[35,23,21,39],[24,61,12,23],[12,22,4,32],[27,5,3,12]];
        titles String[4] = ["US","Asia","Europe","Africa"];
        labels String[4] = ["Joe","Mary","Fred","Jane"];
        g.setSeries(labels, titles, data);
    end
end
```




Graph widget was user-written in 120 lines of Javascript

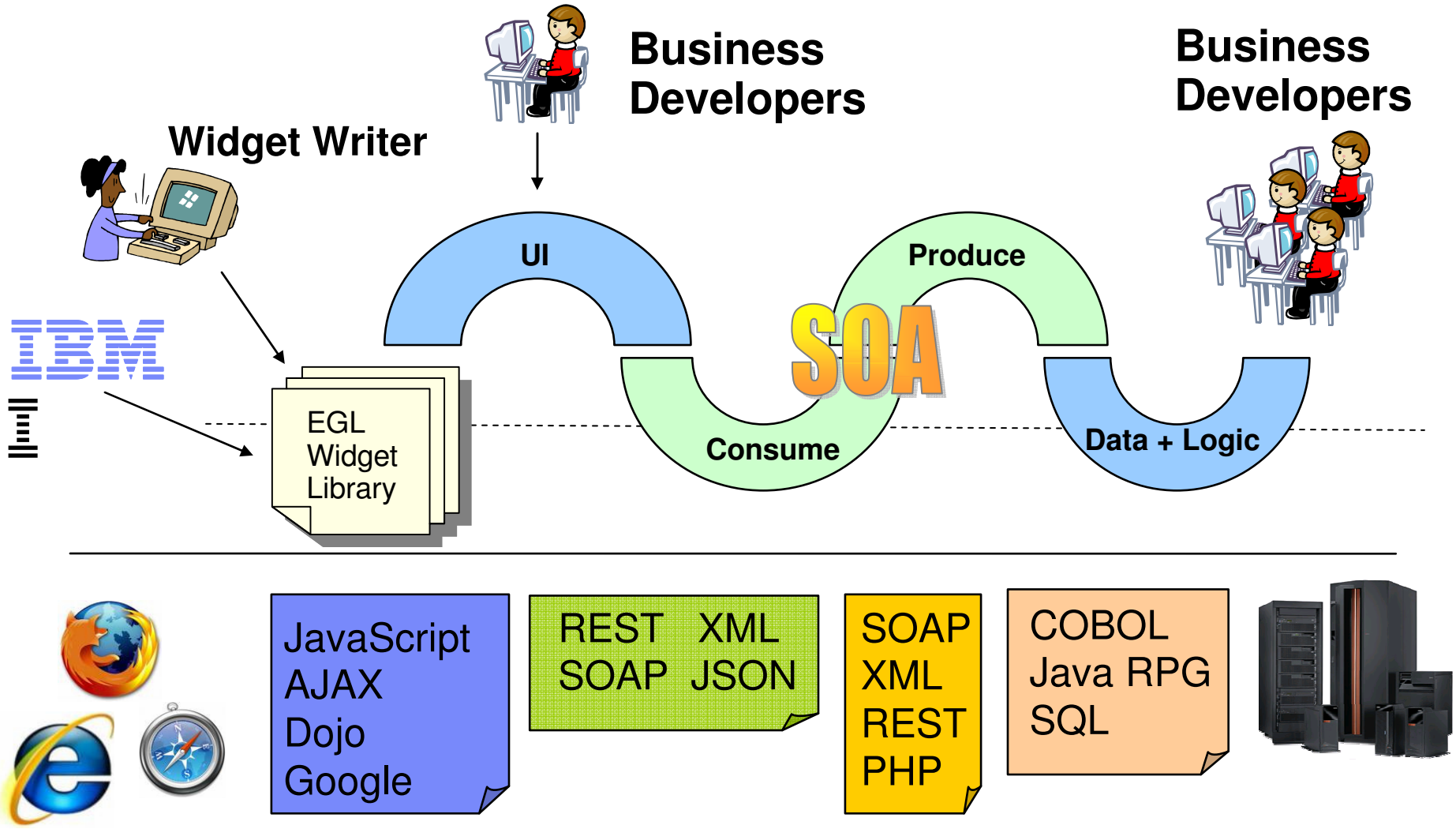
## Displaying Data

```
Handler AutoGridDemo type RUIHandler { initialUI = [ grid ]  }

    grid Grid {
        headerBehaviors ::= GridBehaviors.grayCells,
        margin = 11,
        data = [
            new Team{name = "Virginia Tech", mascot = "Hokie", division = "ACC"},
            new Team{name = "Penn State", mascot = "Nittany Lion", division = "Big Ten"},
            new Team{name = "University of Virginia", mascot = "Cavalier", division = "ACC"}
        ]
    };
end
```

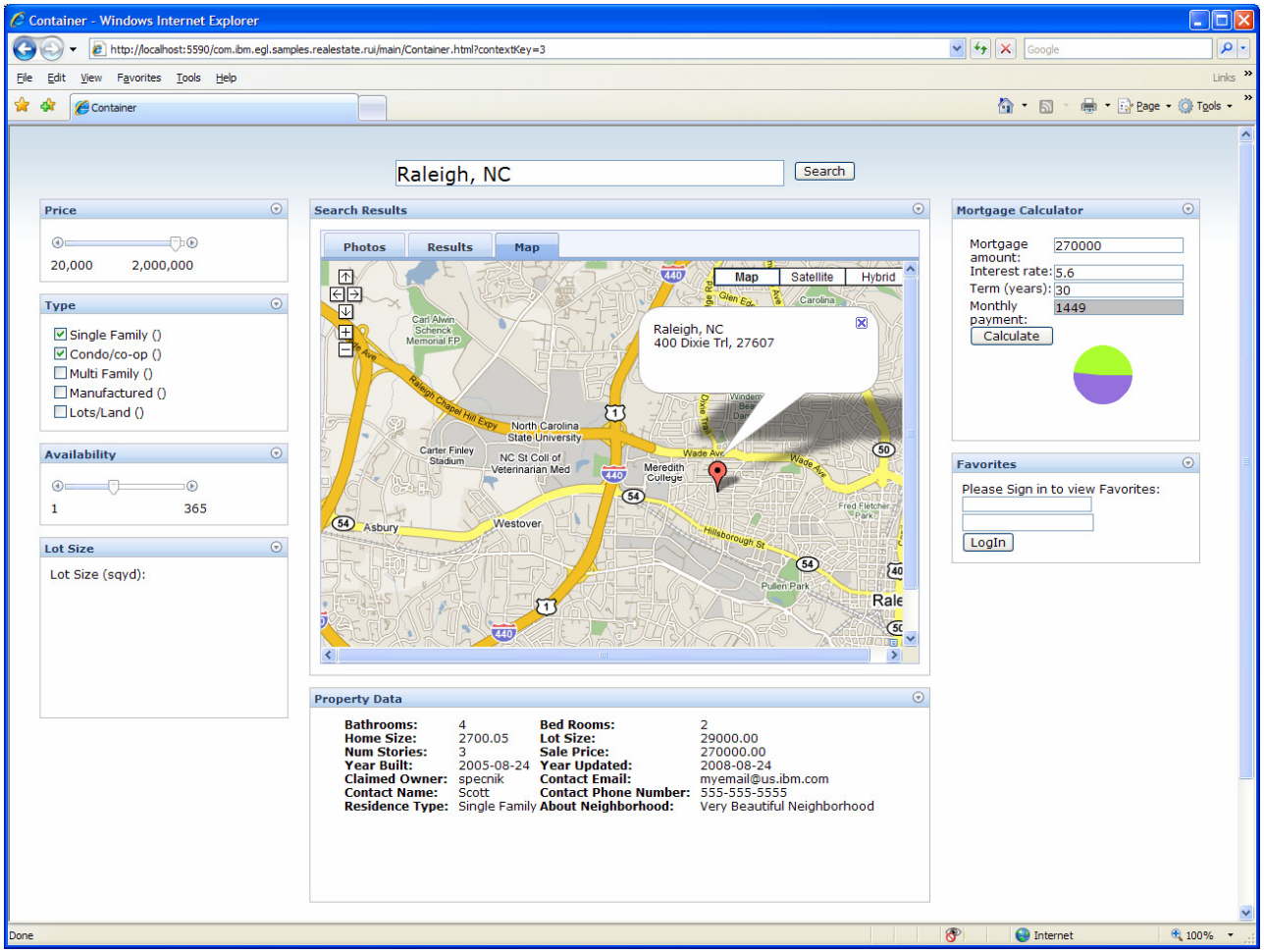
| division | mascot | name |
|---|---|---|
| ACC | Hokie | Virginia Tech |
| Big Ten | Nittany Lion | Penn State |
| ACC | Cavalier | University of Virginia |

# EGL: Shielding Complexity



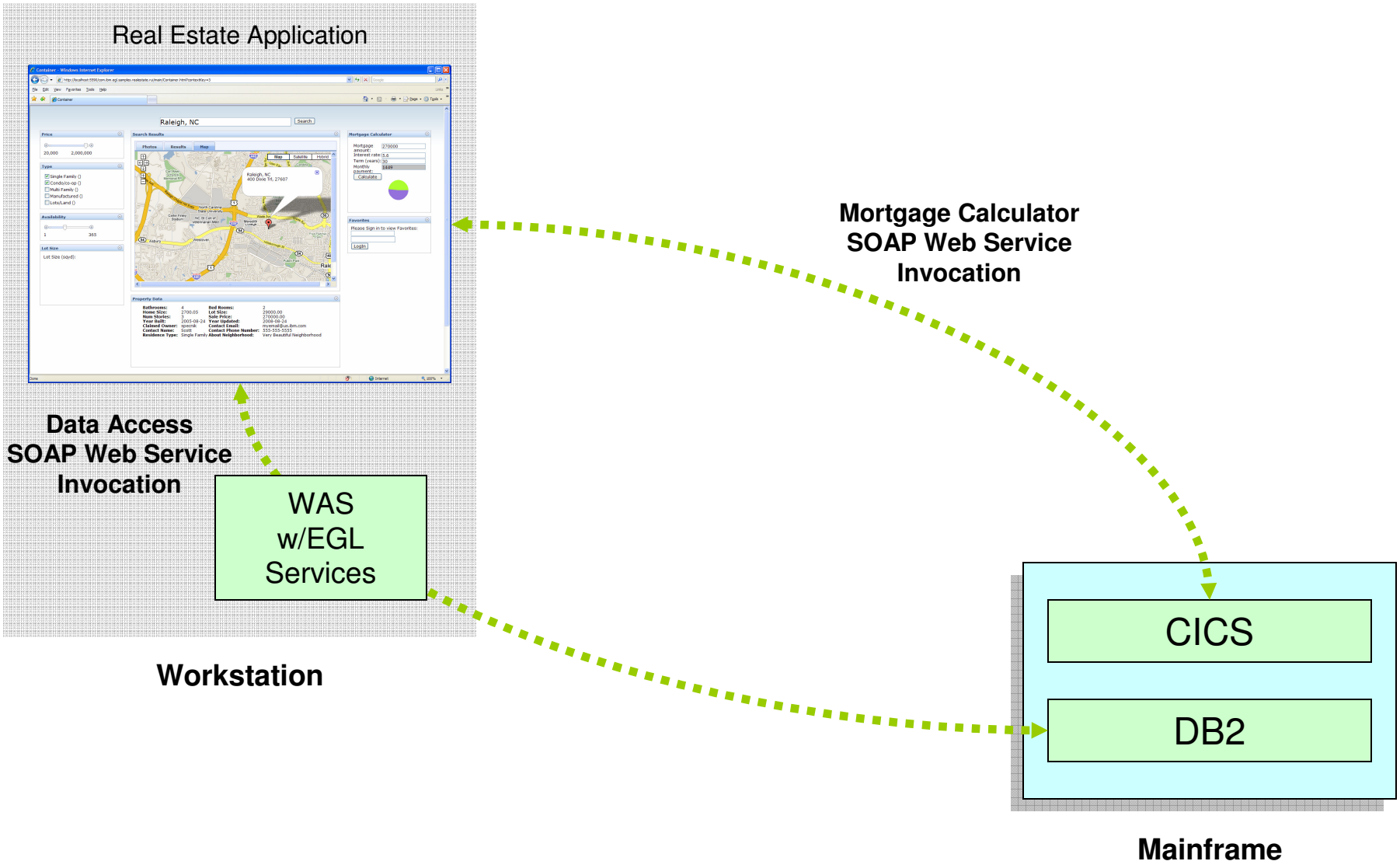
# EGL Rich UI

- **Enables development of rich, Web 2.0-style applications.**
  - Simple programming model
  - Productive development tools
  - Comprehensive widget library, including support for Dojo widgets (coming to the EGL Café soon)
  - Web services and Mash-ups
  - Extensibility




*Demo application written completely in EGL Rich UI. Uses Dojo widgets and SOAP service calls to access data in DB2 and to drive CICS transactions.*

# Real Estate Demo Application Architecture Overview



# Three Benefits of EGL and RichUI

1. 2. 3.

## Single Language

- EGL in 3 tiers:
  1. Data and Logic
  2. Soap/Rest Services
  3. Declarative UI
- Just 1 language
- Skill transfer
- All Data/Business/ Service logic held on the server

## Rich User Interfaces

- Declarative UI
- Ajax support
- Rich set of widgets
- Easily extensible
- Mashup using SOA
- Presentation logic runs on the client.

## Mobility

- Disconnect/Sync
- Easy installation
- No version pain
- Flexibility

# Drive innovation with technology advancements

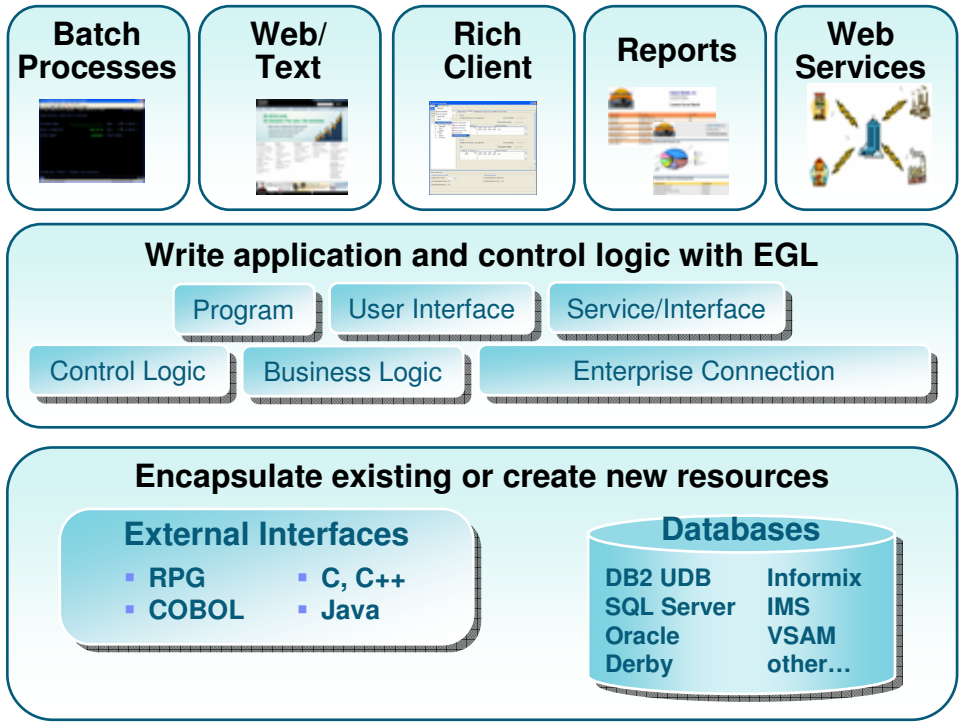
*Accelerate adoption of Web 2.0 and SOA with IBM Rational Business Developer*

**IBM Rational Business Developer (RBD)** helps business developers build cross-platform Web and service-oriented applications through "EGL", The solution helps reduce complexity of the legacy systems by accelerating the delivery of key business transactions as services. COBOL developers with business application knowledge can use **"EGL" – IBM's newest and most powerful business language**", to design, develop, deploy and test the new solutions in record time.

## Accelerate adoption of Web 2.0 and SOA technologies

- **Exploit EGL the newest and most advanced business language**
- **Update and extend mission-critical applications as services, leveraging their core value in new ways**
- **Apply existing "business-knowledgeable" and new staff on multi-platform projects**

- ✓ **Rapidly exploit latest middleware and emerging application architectures to deliver new solutions**
- ✓ **Shorten the learning curve for new and existing developers**
- ✓ **Reduce time to market and increased responsiveness**




IBM Rational Business Developer

# Drive innovation with technology advancements

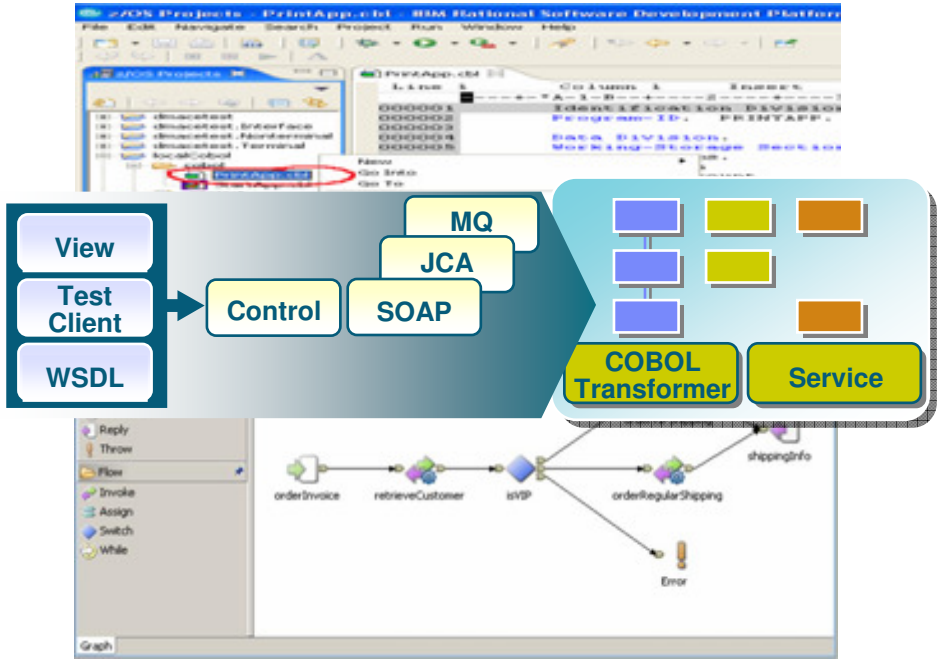
*Build traditional and composite applications with Rational Developer for System z with Java and Rational Developer for System z w/EGL*

**IBM Rational Developer for System z (RDz)** provides powerful eclipse-based tools for building COBOL, PL/I, C/C++, Java, EGL, CICS, IMS and batch applications. RDz is designed to optimize and simplify application development for traditional processes, and for the SOA you're building to address your needs today and in the future—through best practices, visual tools, templates, code generation and a comprehensive development environment.

## Achieve business innovation and transformation objectives

- **Deliver rich Web 2.0-style solutions using EGL without needing to learn or program the intricacies of low level technologies.**
- **Develop z/OS applications and CICS and IMS Web services more productively than ever**
- **Manage core and composite artifacts from z/OS**
- **Speed software delivery with integrated, automated end-to-end processes**

- ✓ **Reduce cost and time of maintenance activities**
- ✓ **Faster response to business needs resulting in better customer experience**
- ✓ **On-time and on-budget project delivery**




IBM Rational Developer for System z w/Java

IBM Rational Developer for System z w/EGL

# Resources to help you kick-start modernization

## New Educational Resources

### System z Sandbox

*Example assets and best practices providing low-risk, practical, hands-on path to leveraging IBM solutions*

**http://www.ibm.com/developerworks/downloads/emsandbox/systemz.html?S_TACT=105AGX28&S_CMP=EMSAND**

### EGL Distance Learning

*10 days of FREE instructor-led virtual class*

**http://www.ibm.com/developerworks/offers/techbriefings/details/rbdtws.html**

### Seminar Series: Web 2 Architectures for IBM Mainframe Developers

*For managers and technologists focused on extending value of mainframe applications in modern Web 2 Architectures*

**http://www-01.ibm.com/software/os/systemz/seminars/mad/**

## System z Solution Videos and Demos

*Translated solution videos and demo to help you raise awareness of Enterprise Modernization offerings*

- English
- German
- Korean
- Spanish
- Mandarin Chinese
- Japanese
- French
- Italian

http://rational.dfw.ibm.com/index.html

***Jump-start your modernization projects!***

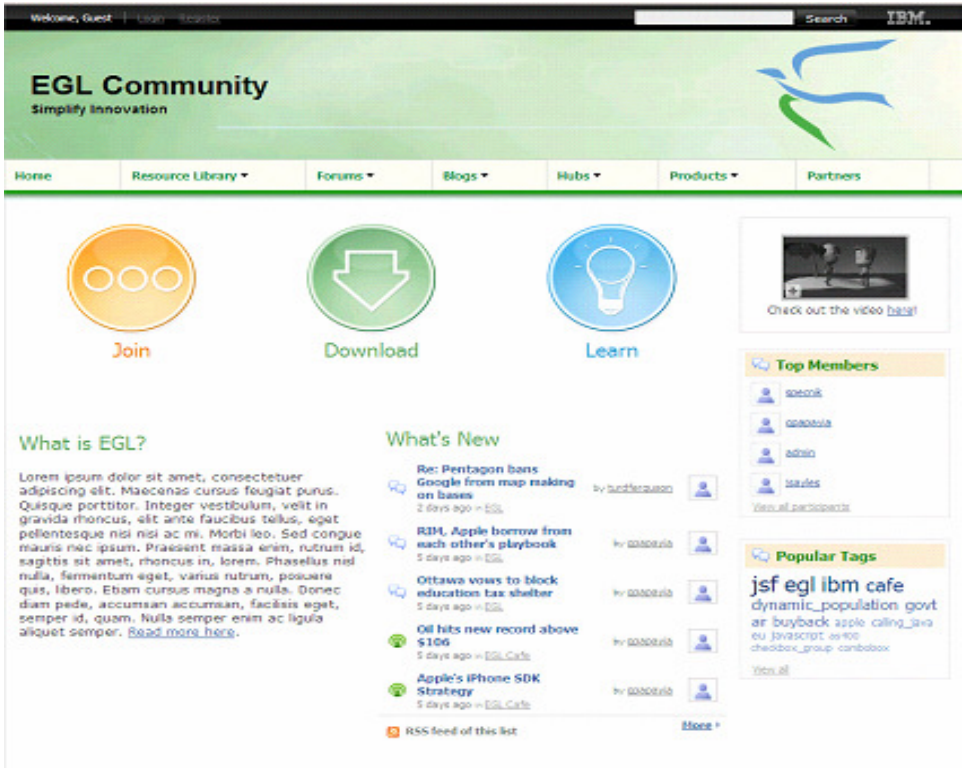
**http://www.ibm.com/rational/modernization**

# Rational Business Developer – EGL Café

## *Building a Community for Business Developers*

- Attract participation with compelling experience – Partners, Users, Prospects
- Capture “Wisdom of the Crowd” – Experience knowledge base
- Sustainable Participation – Team, Users
- Leverage re-use to reduce effort
- Extending community to CICS, IMS, COBOL, and PL/I developers

**http://ibm.com/rational/eglcafe**




***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!**

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/)

**© Copyright IBM Corporation 2009. All rights reserved.** The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

Questions