



IBM Rational Software Conference 2009
As Real as It Gets!



Develop **Rich Internet Applications** using
IBM Rational Application Developer
and the
IBM WebSphere Web 2.0 Feature Pack

Tim Francis

Distinguished Engineer, RAD Chief Architect

Sarika Sinha

Senior Staff Software Engineer, RAD Development

Rational. software

MAC27

Agenda

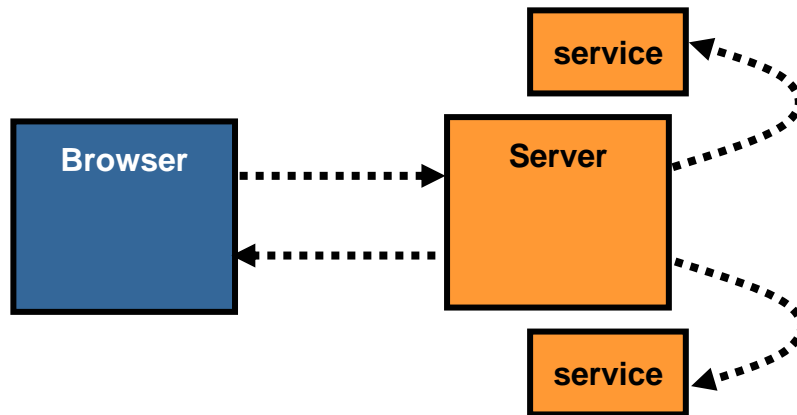
- Web 2.0 – Overview & Concepts
- WebSphere Web 2.0 Feature Pack
- Developing Web 2.0 Apps in RAD
- Demo
- Questions

Today's Agenda

- Web 2.0 – Overview & Concepts
- WebSphere Web 2.0 Feature Pack
- Developing Web 2.0 Apps in RAD
- Demo
- Questions

Web Applications and SOA

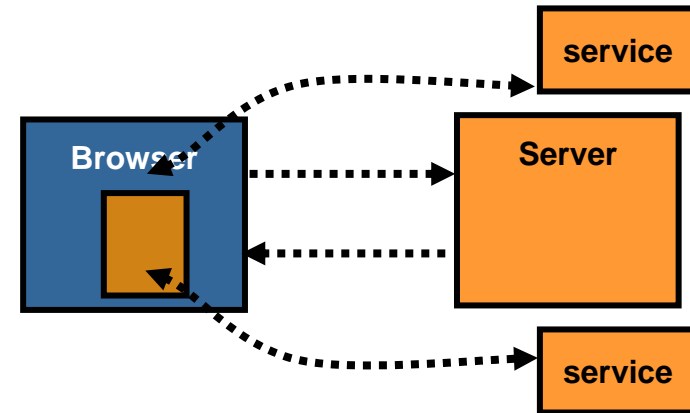
Classic web application patterns promoted server side service access



Existing web services standards (WS_*, WSDL, SOAP) are typically used in this model

Focus is on access from multiple programming languages using a variety of communication protocols

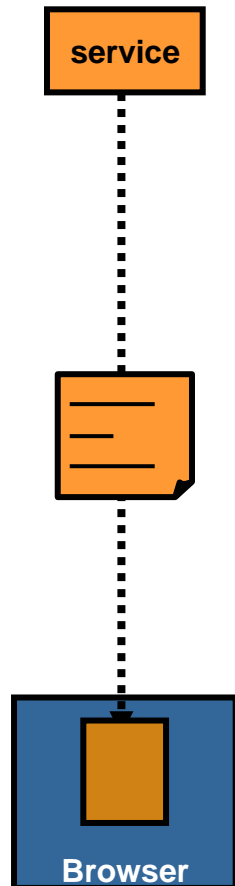
Web 2.0 application patterns promote client side (browser) access to services



Emerging web services approaches leveraging REST and JSON are typically used in this model

Optimized for *single language* (JavaScript), *single protocol* (HTTP) access via Ajax in a browser

Web 2.0 style services – key concepts



REST

- ▶ **RE**presentational **S**tate **T**ransfer
- ▶ Server side architectural style relying on HTTP semantics to access services or resources
- ▶ Easily invoked by browsers via Ajax

JSON

- ▶ **J**ava**S**cript **O**bject **N**otation
- ▶ Data format used to exchange information between browser and a service
- ▶ Directly consumable by JavaScript clients

Ajax

- ▶ **A**synchronous **J**ava**S**cript **A**nd **X**ML
- ▶ Browser based technology to provide highly interactive and responsive web pages
- ▶ Enables the browser to invoke services directly from the client

Agenda

- Web 2.0 – Overview & Concepts
- **WebSphere Web 2.0 Feature Pack**
- Developing Web 2.0 Apps in RAD
- Demo
- Questions

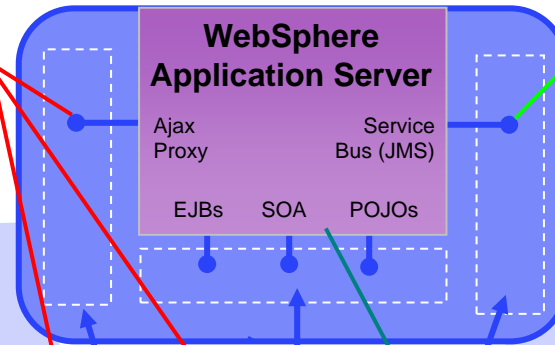
WebSphere Feature Pack for Web 2.0 Highlights

Web 2.0 to SOA Connectivity

For enabling connectivity from Ajax clients to SOA services and other JEE assets. Extends enterprise data to customers and partners through web feeds.

AJAX Messaging

For connecting Ajax clients to real-time updated data like stock quotes or instant messaging.



External Web Services



Event-Driven Data

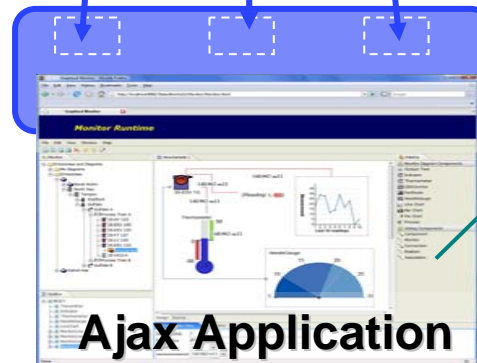
IBM \$125.25 +\$2.50... MSFT \$43.75 -\$1.50 ...



Web Feeds

Ajax Development

Based on RAD 7.5 and Dojo. Reduces time to market and helps lower Ajax adoption and development costs.

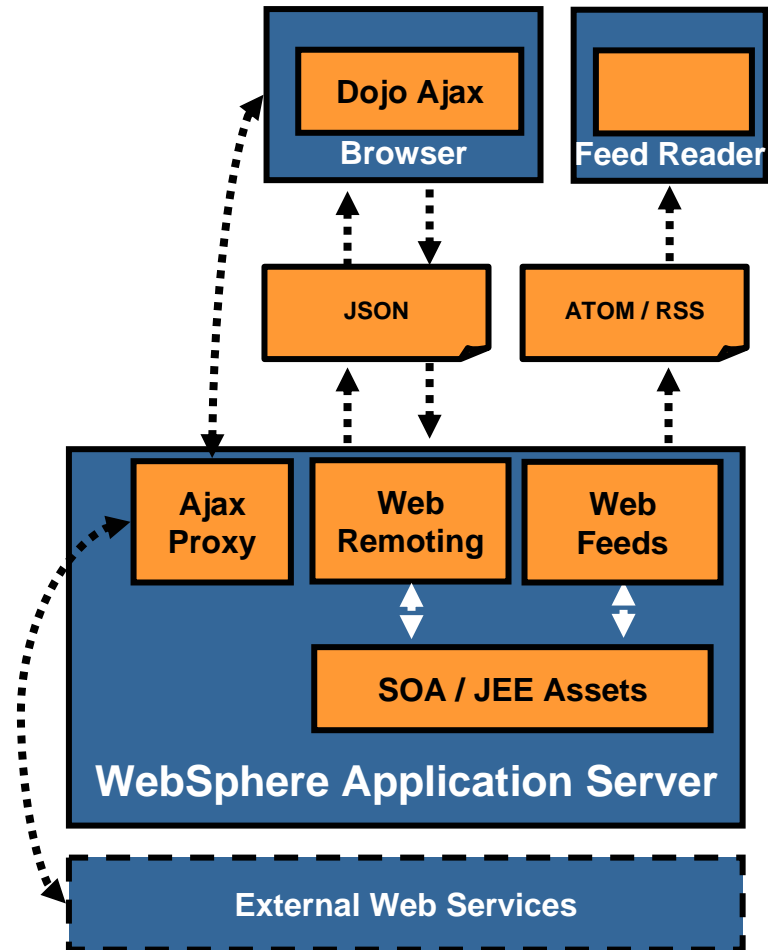


Ajax Application

Installs on: WAS 6.0, 6.1, 7.0

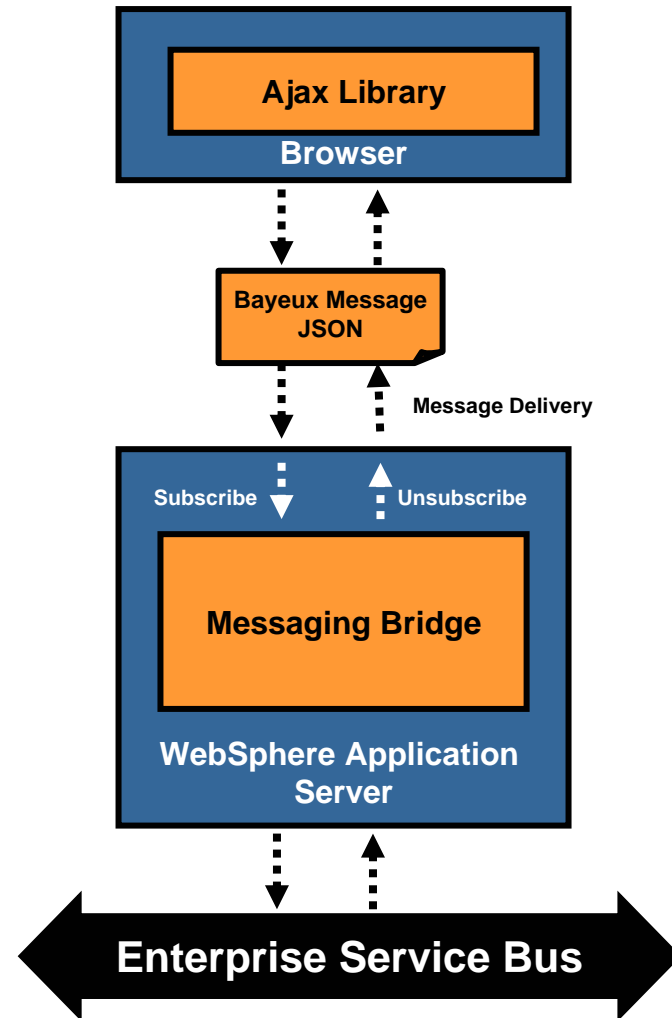
Web 2.0 to SOA Connectivity – Overview

- **Web 2.0 to SOA Connectivity** – For enabling connectivity from Ajax clients to external web services, internal SOA services, and other JEE assets.
- Extends enterprise data to customers and partners through web feeds.



Ajax Messaging - Overview

- **Ajax Messaging** – For connecting Ajax clients to real-time updated data like stock quotes or instant messaging.
- Connects the browser to a service integration bus for pushing server-side events to the browser.



Dojo Toolkit v1.0



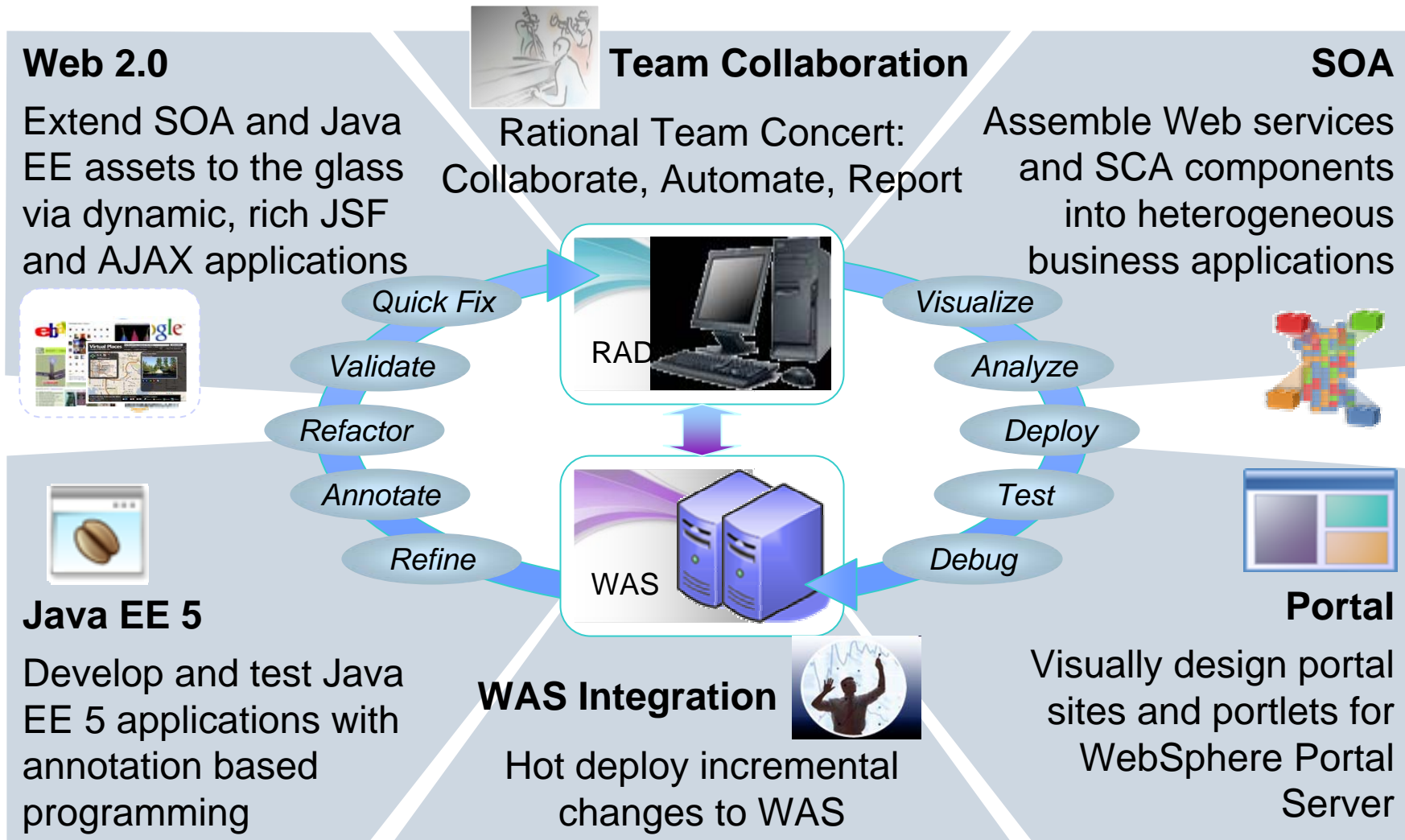
- A JavaScript toolkit for developing Ajax applications with rich user interfaces
- Key capabilities
 - ▶ Works well across most modern browsers
 - ▶ Small footprint, high function
 - ▶ Features
 - Dojo Core
 - Utility routines
 - Event handling system
 - Ajax support
 - Drag and Drop
 - Language utilities & support for localization
 - Data access
 - Dojo Widgets
 - Accessible
 - High-quality neutral default theme (replaceable)
 - Extensive layout & form capabilities
 - Data bound widgets
 - Many community provided “extension” modules



Agenda


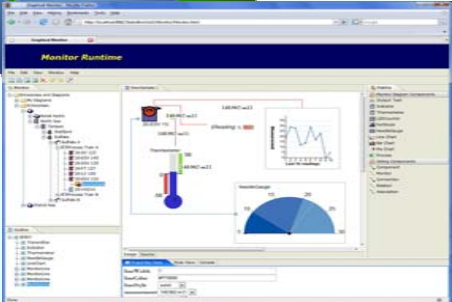
- Web 2.0 – Overview & Concepts
- WebSphere Web 2.0 Feature Pack
- **Developing Web 2.0 Apps in RAD**
- Demo
- Questions

RAD Accelerates SOA, Java EE, Web 2.0, Portal Development for IBM Middleware



Use RAD to build Enterprise Java IT applications

1st Tier



Web Client

- JavaScript
- JSF

Portal Client

Rich Java GUI Client

2nd Tier

REST style Services

Web Services

EJB
JPA
POJO






WebSphere Application Server

Tomcat JBoss

InfoSphere MashupHub

WebSphere Portal Server

3rd Tier

- IBM CICS Transaction Server
- IBM InfoSphere DB2
- Oracle
- IBM InfoSphere IMS
- SAP
- PeopleSoft
- ...

Web 2.0 Development in RAD

- JavaScript development
- DOJO page development
- WebSphere Web 2.0 Feature Pack support

JavaScript Development

- Basic language Tools
 - ▶ Full-featured JavaScript editor
 - Syntax highlighting
 - Content assist
 - Mark occurrences
 - Customizable formatting
 - Bracket auto-completion, indentation
 - ▶ Outline view
 - ▶ Validation
 - ▶ Refactoring

JavaScript Debugging

- Firebug
 - ▶ Shipping with RAD

- Eclipse integration in RAD
 - ▶ Automatic installation of Firebug
 - ▶ Launch page into Firebug
 - ▶ Editor synchronization
 - ▶ Problems and Console views synchronization

DOJO Page Development

- DOJO project setup
- DOJO-specific *JavaScript* content assist, validation, refactoring
 - ▶ `dojo.declare("dijit.Editor", dijit._editor.RichText, {...});`
- DOJO-specific *markup* content assist, validation, palette items, property views



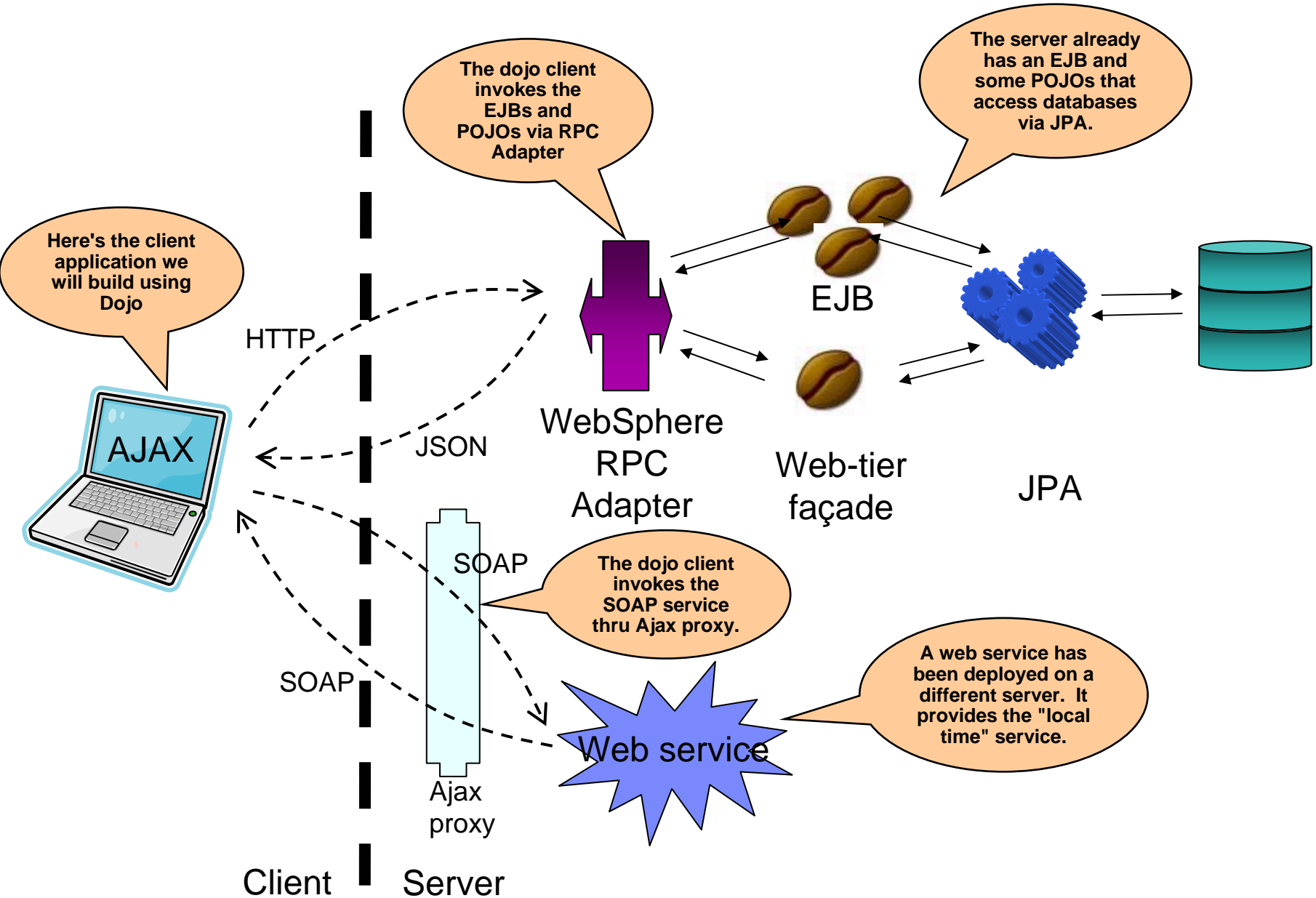
- ▶ Based on OpenAJAX metadata spec

WebSphere Web 2.0 Feature Pack Support

- Setting up Dojo projects with FeP content (“Blue Dojo” + IBM extensions)
- Visual Tools for Web Remoting (HTTP RPC)
- Visual Tools for AJAX Proxy
- Application libraries
 - ▶ Apache Abdera for ATOM and RSS feeds
 - ▶ JSON4J
 - ▶ Web Messaging (pub/sub, WebSphere SIB integration)
- Samples gallery



DEMO



Client | Server

A Tab layout for the project center

A Split layout for the master vs. details

A Grid to display all projects

#	Name	Start Date	End Date	Department	Manager
AD3100	ADMIN SERVICES	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D01	000010
AD3110	GENERAL ADMIN SYSTEMS	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D21	000070
AD3111	PAYROLL PROGRAMMING	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D21	000230
AD3112	PERSONNEL PROGRAMMING	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D21	000250
AD3113	ACCOUNT PROGRAMMING	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D21	000270
IF1000	QUERY SERVICES	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	C01	000030
IF2000	USER EDUCATION	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	C01	000030
MA2100	WELD LINE AUTOMATION	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D01	000010
MA2110	W L PROGRAMMING	1982-01-01T00:00:00-0500	1983-02-01T00:00:00-0500	D11	
MA2111	W L PROGRAM DESIGN	1982-12-01T00:00:00-0500	1982-12-01T00:00:00-0500		
MA2112	W L ROBOT DESIGN	1982-12-01T00:00:00-0500	1982-12-01T00:00:00-0500		
MA2113	W L PROD CONT PROGS	1982-02-15T00:00:00-0500	1982-12-01T00:00:00-0500		

On click of a project, its Department and project manager data are retrieved

Project Manager data is retrieved all at once during page load, and saved in Dojo date store. It's then retrieved using Dojo queries

Department		Project Manager	
Name	ADMINISTRATION SYSTEMS	Name	PEREZ, MARIA
ID	D21	ID	000270
Location		Phone	9001
Manager	000070	Department	D21
		Salary	27380.00

Department data is retrieved from a separate request each time

Salary data is retrieved from an external SOAP web service.

Additional Resources

- v7.5.2 is now available:
 - ▶ <http://www-01.ibm.com/software/awdtools/developer/application/>
- developerWorks articles on web 2.0 development using RAD:
 - ▶ http://www.ibm.com/developerworks/rational/library/08/0930_stein/
 - ▶ http://www.ibm.com/developerworks/rational/library/08/1118_endres/
 - ▶ http://www.ibm.com/developerworks/rational/library/08/0819_mutdosch/index.html
- Overview of Web 2.0 Feature Pack:
 - ▶ http://www.ibm.com/developerworks/websphere/techjournal/0802_haverlock/0802_haverlock.html
- Documentation:
 - ▶ <http://publib.boulder.ibm.com/infocenter/radhelp/v7r5/index.jsp>

Questions

Thank You

© 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, Telelogic, the Telelogic 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.



Backup

Web 2.0 style services – what problem are we trying to solve?

■ Browser access to SOA web services can be complex

- ▶ Browsers must translate from SOAP web services formats and protocols into browser-friendly JavaScript or HTML
- ▶ More complex browser side logic means more testing and debug of individual clients (Microsoft IE, Firefox, Safari, Opera, etc.) due to browser compatibility issues

Web 2.0 style services focus on simplified browser access

- HTTP centric patterns on the server (REST)
- JavaScript friendly data formats (JSON)
- Browsers invoking services with minimal overhead (Ajax)

Other issues with Web 2.0 applications

- The use of Ajax on the browser enables a richer web application experience, but it does pose additional challenges for application developers:
 - ▶ Browser and server event handling
 - Classic web applications are server driven; Ajax applications have a combined client/server model
 - Events can be generated from the client to the server, or from the server back to the client
 - Ajax applications often use a *publish and subscribe engine* to handle the flow of events between client and server
 - ▶ JavaScript development challenges
 - ▶ Hundreds of proprietary and open source Ajax toolkits to choose from
 - ▶ Ajax security concerns
 - The proliferation of JavaScript code in browsers has led to multiple security issues, notably cross-site scripting attacks
 - Browser security restrictions pose problems for programmers legitimately seeking to use cross-site functions in building their applications

Ajax applications can benefit from server side support to address key issues such as event handling and proxy support

Web 2.0 to SOA Connectivity – Feature List (1 of 2)

■ **Web Remoting**

- ▶ Provides a lightweight web endpoint which can expose methods of Java EE assets (EJB's, PoJo's, web service proxies)
- ▶ Can be easily invoked from Ajax applications using JSON or XML formats
- ▶ Supports HTTP GET/POST mapping for methods
- ▶ Enabled through simple configuration options without rewriting the original Java objects, EJBs or web services

■ **The JSON4J library**

- ▶ An implementation of a set of JavaScript™ Object Notation (JSON) handling classes for use within Java environments
- ▶ JSON has gained much popularity within Ajax applications as a lightweight data interchange format (defined at <http://www.json.org>)

Customer Pain Points with RIA

- Confusion over Rich Internet Application (RIA) technology options
 - Hundreds of proprietary and **open source Ajax toolkits** to choose from
 - Concern over vendor lock-in with **proprietary RIA Frameworks**
 - Risks of going with OS Ajax Frameworks

- Ajax frameworks
 - T
 - S
 - E

legitimately seeking solutions for applications

•RAD provides visual and source tools to aid in JavaScript and Dojo development

•Dojo provides a lot of enterprise features like localization, accessibility

Dojo abstracts out low level details of JavaScript and handles differences between browsers

•RAD provides an integrated debug environment with Firebug

Dojo is part of the OpenAJAX

•AJAX Proxy provides secure access to mashup cross site services

•RPC Adapter can have validators on parameters

RAD and the Web 2.0 FEP lets you enhance your existing applications with HTTP RPC services to target J2EE and SOA artifacts

Dojo provides a lot of enterprise features like localization, accessibility

•AJAX Proxy provides secure access to mashup cross site services

•RPC Adapter can have validators on parameters

- JavaScript development challenges
 - Javascript is extremely flexible, the
 - How to deal with **browser compatibility**
 - Need to leverage **current investment** in J2EE & developer skills

Web 2.0 to SOA Connectivity – Feature List (2 of 2)

■ **Ajax Proxy – Browser access to cross-site services**

- ▶ A lightweight Proxy to enable browser based access to cross-site services in a Web 2.0 fashion
- ▶ Proxy can run embedded within another JEE application or standalone
- ▶ Uses JEE application level security for proxy access control
- ▶ Support for white-listing policies for filtering on criteria of incoming requests such as: Cookies, MIME types, HTTP Headers, HTTP Verbs (GET, POST, PUT, etc)

■ **Web Feeds**

- ▶ ATOM and RSS libraries for exposing JEE resources as web 2.0 style “data feeds” - pushing updates to clients when data changes

Ajax Messaging – Feature List

- A publish/subscribe messaging implementation
 - ▶ Connects the browser to the WebSphere Application Server service integration bus, for pushing server-side events to the browser
- Ajax Messaging implements the “Comet” server model
 - ▶ Simulated long running connections
 - ▶ Topic-based publish/subscribe mechanism
 - ▶ JEE Applications use standard JMS APIs to access messages routed to/from Ajax clients, rather than augmented servlet APIs
 - ▶ Uses the Bayeux protocol for client/server communication
- Client side support is provided by the Dojo Toolkit
- Security
 - ▶ Restricts cross-site access to non-authorized domains.

What is REST ?

- REST is the acronym for “**Representational State Transfer“**
- It is the architectural model the World Wide Web is based on
- Principles of REST
 - ▶ Resource centric approach
 - ▶ All relevant resources are addressable via URIs
 - ▶ Uniform access via HTTP – GET, POST, PUT, DELETE
 - ▶ Content type negotiation allows retrieving alternative representations from same URI
- REST style services
 - ▶ are easy to access from code running in web browsers, any other client or servers
 - ▶ can serve multiple representations of the same resource
- More info:
<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

What is JSON ?

- The vast majority of the clients of a RESTful service will be written in JavaScript
- In recognition of that, JSON (JavaScript Object Notation) allows for rapid exchange of JavaScript objects, but also in a simple, human-readable format
- JSON is built up from a collection of name-value pairs and ordered lists of values

```
{  
    "customer" : {  
        "name" : "Jane Doe",  
        "company" : "Acme Enterprises"  
    }  
}
```

What is AJAX ?

- AJAX is the acronym for **A**synchronous **J**avaScript **A**nd **X**ML
- The purpose is to create more dynamic and responsive web pages
- It is also about building web clients in a Service Oriented Architecture that can connect to any kind of server: JEE, PHP, ASP.Net, Ruby on Rails, etc.
- AJAX involves existing technology & standards: JavaScript and XML
- AJAX enables major improvements in responsiveness and performance of web applications, e.g. used at Yahoo! Mail, Google Maps, live.com, and others
- AJAX is **NOT** hype – it is very real and very useful for highly interactive applications

Ajax Messaging - Example Applications

- Stock Quotes / Front Office Applications
- SIP Applications – Inbound Call Consoles
- Collaborative Document Authoring
- Chat & Instant Messaging
- Sporting Event Scores
- Browser Based Email
- Management Consoles / Monitoring & Alerts

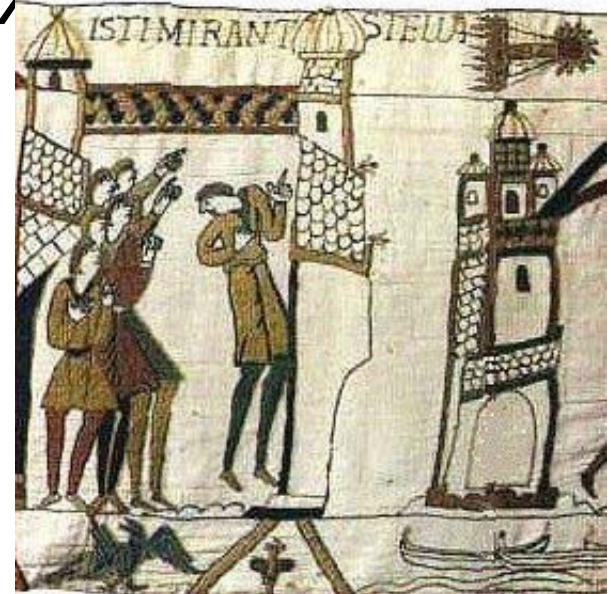
...any application where delivery of events to the client in a timely manner is important

Bayeux Protocol

- Bayeux is a JSON-based protocol for clients to subscribe to events and for servers to deliver them more timely than possible with AJAX

■ Goals

- ▶ Make event delivery fast
- ▶ Keep it simple
- ▶ Allow for the protocol to be extended



Extract from the Bayeux tapestry showing the arrival of Halley's comet