



IBM Software Group

Building Rich Internet Applications Using JavaScript

IBM Software Group Emerging Technology

Dan Gisolfi
Laurent Hasson



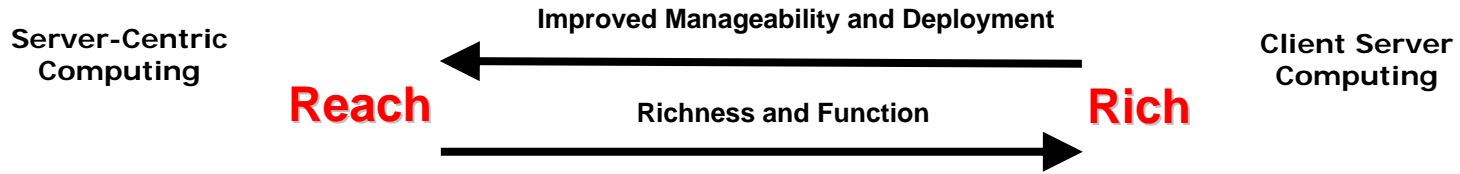
@business on demand.

Key Messages

- IBM has been working with RIA based technology for over 3yrs and has assets and experience with Ajax-like communication patterns
- A robust RIA framework should implement a full MVC client-side architecture including Data and Event models as well as flexibility with respect to backend infrastructure.
- One benefit of a client-side MVC architecture is that it allows widgets from multiple UI component libraries to communicate and share a common data model.
- IBM is driving the industry towards reusable and accessible JavaScript widgets
- An open source RIA framework could benefit from contributions derived from a number of exiting efforts such as Laszlo, JSL, Rico, etc.

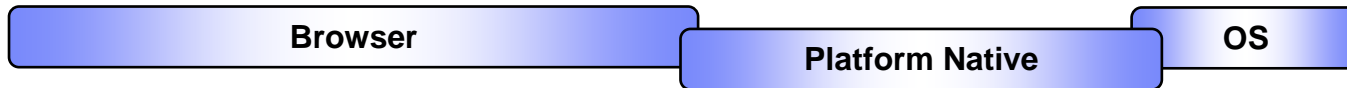


User Experience Technology Spectrum



Common Browser	Rich Browser	Enhanced Browser	Locally Managed Application	Managed Native Application	Native Application
Rich Internet Applications					
Traditional Device Dependent Web Browsers	Web Browsers without Plug-ins	Dynamically supplemented Web Browsers	Locally deployed Service Applications	Remotely deployed Service Applications	Traditional Client Applications
HTML XHTML WML	ECMA-Script DOM DHTML	Applets Plug-in Flash	OSGi Bundles Quicken MS-Money	OSGi Bundles	No Standard deployment model

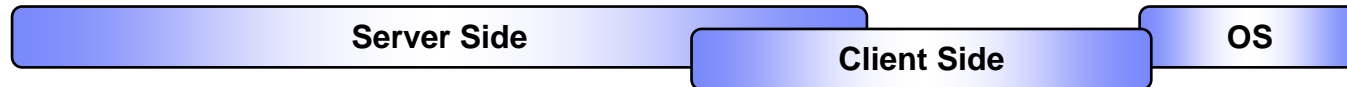
Application Container



UI Rendering

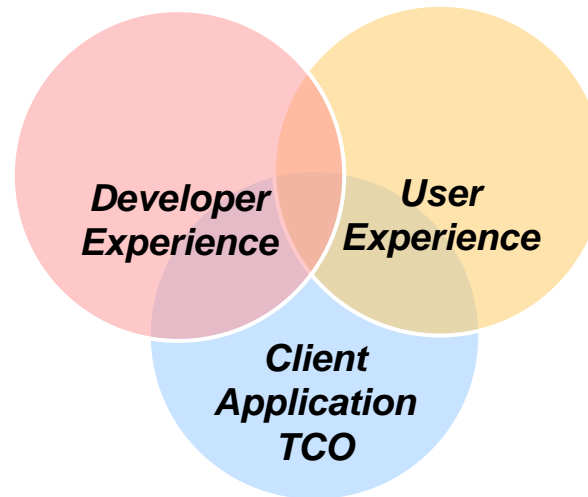


Programming Model



Observations of RIA Framework Requirements

- Consistent programming model
- Simplification of technical details based on roles
- Common tooling environment, regardless of role to reduce cost and learning curve for developers
- Support for open standards and long term vendor commitment
- Flexible tooling to support common MVC architecture



- Real-time Data Monitoring
- Rich Web Application UI Experience (Rich or Ultra-Rich)
- Eliminate page refreshes
- Improve the nature of user interactions with Web Applications

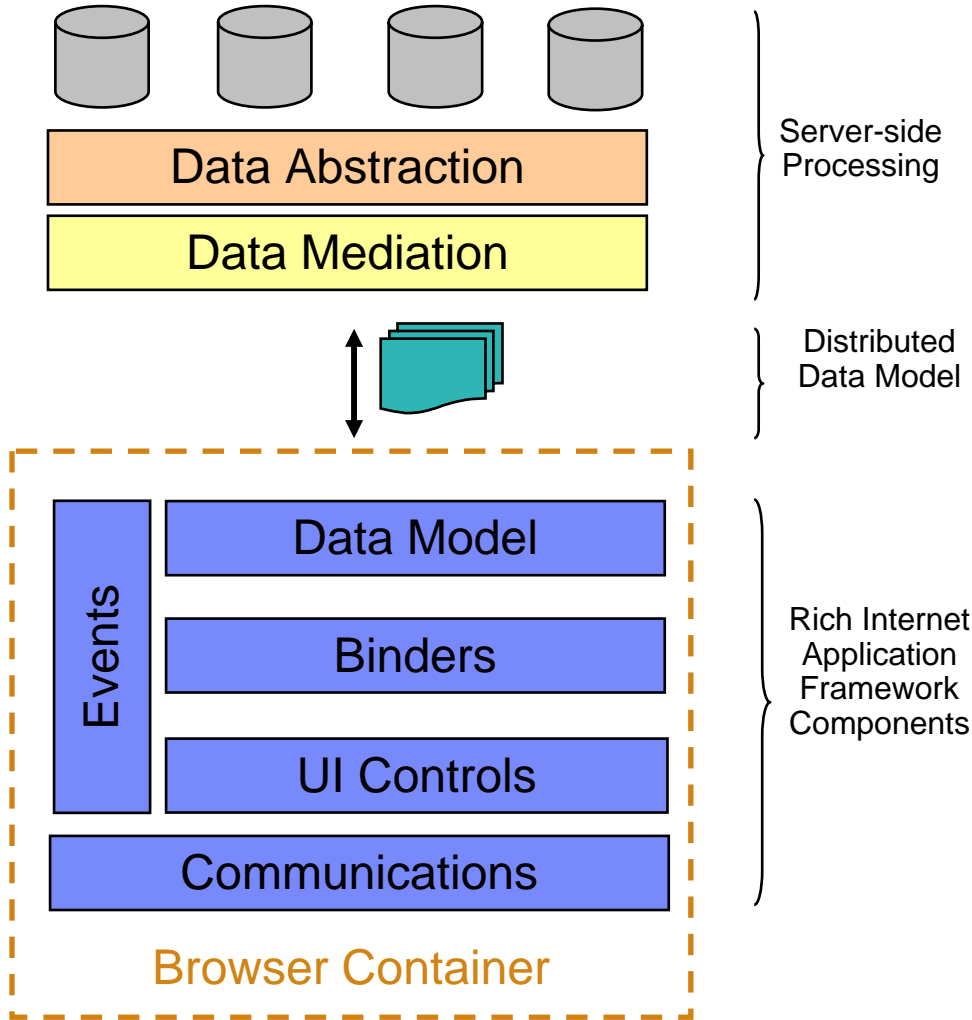
- Reduce Network Traffic (less round trips)
- Increase Web Application Manageability
- Increase Web Application Reach (broader platforms)
- Offload computations from server to client (reduce processing drag)

In order to enable the full potential of rich internet applications we need to establish a zero-footprint JavaScript-based client-side MVC architecture.

- JavaScript
 - ▶ JavaScript is found on over 50% of all web sites today,
 - ▶ It is the common enabling technology for UI Widget development regardless of scripting language(PHP, JSP, ASP, etc)
 - ▶ New web applications are being created to render custom widgets modeling rich desktop componentry based on JavaScript
 - ▶ JavaScript is a broadly supported by browsers
- Rich Internet Applications
 - ▶ Requires best qualities of desktop, web and communication software whereby the following activities are all addressed independently of the server-side infrastructure
 - Computations are performed client-side,
 - Data is sent and retrieved in the background asynchronously with respect to user interactivity
 - Redraws sections of a screen with refreshing page
 - Uses audio and video in a tightly integrated manner
- MVC Architecture
 - ▶ Web applications typically implement a struts-like server-centric Model-2 MVC architectures pattern
 - ▶ Current browsers do not contain the necessary MVC pattern components to enable RIAs
- Accessibility
 - ▶ HTML does not provide adequate markup to support accessible dynamic content.
 - ▶ Current negative position of W3C WAI on JavaScript has dramatically affected the ability for persons with disabilities to access web content in spite of the pervasiveness of JavaScript.



RIA Frameworks: Conceptual Model



Server-side Processing

- Ability to connect to wide variety of data services
- Leverage server-side abstraction layers for data-store independence
- Manage transformation between inbound/outbound data format on the wire

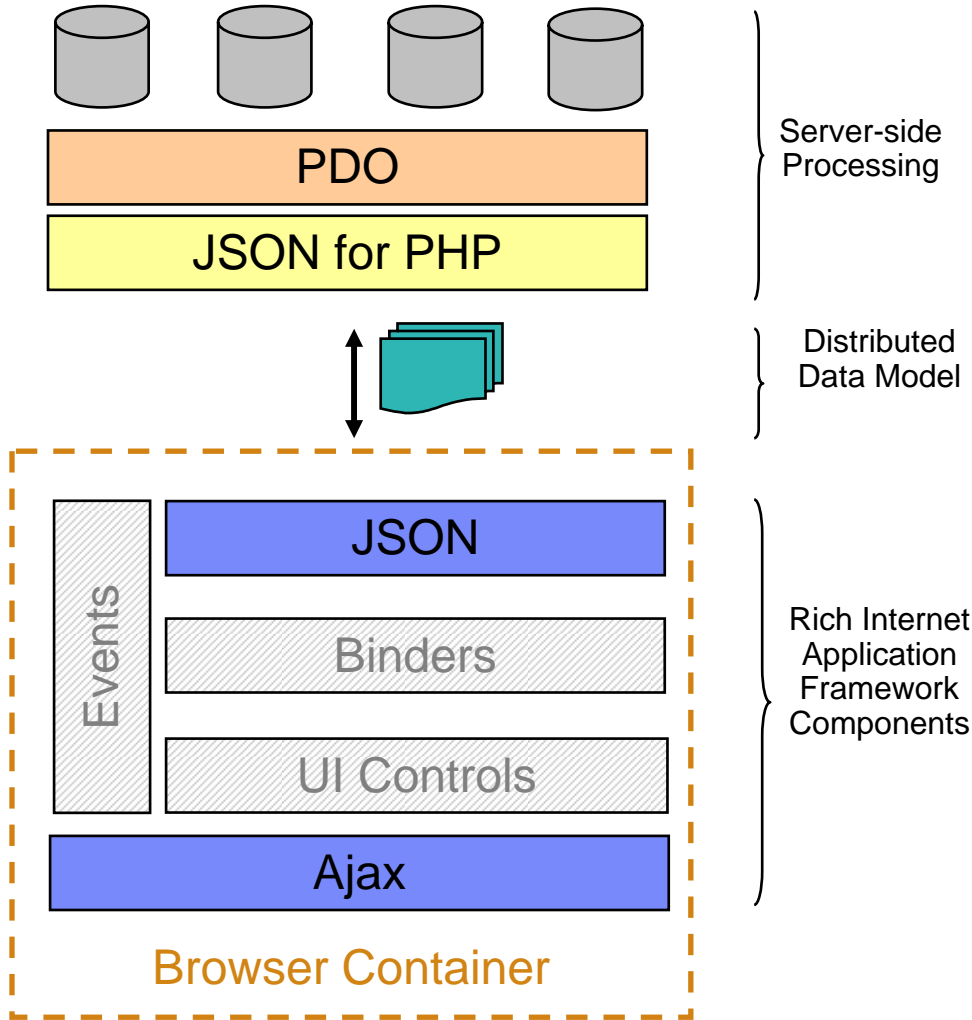
Distributed Data Model

- Expected data format agreed upon by server and client side processing models

Client-side framework components that collectively comprise an MVC architecture

- Event Model
 - Supporting client side processing of user interactions and remote data updates
- Data Model
 - Client-side data model and distributed data API
- Binders
 - Client-side adapters for model, view and controls
- UI Controls
 - Accessible and extendable UI widget library that is bound to a specific client-side data model
- Communications
 - HTTP Back Channel
 - Web Services
 - Web Messaging (Real-time Push)

RIA Frameworks: Example Open Source Component Technologies



Server-side Processing

- PDO is an example data abstraction layer for PHP5
- JSON for PHP provides a PHP adapter API for generating and processing JSON data streams

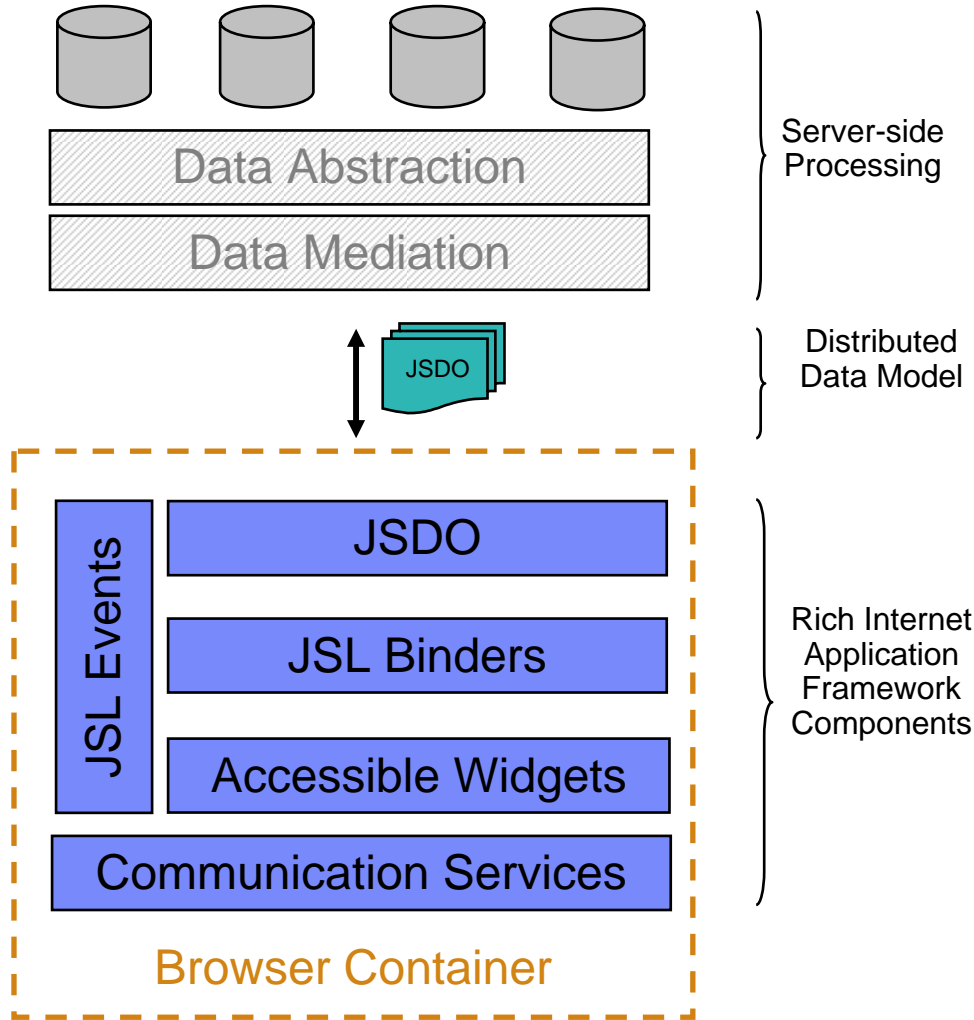
Distributed Data Model

- Expected data format agreed upon by server and client side processing models

Client-side framework components that collectively comprise an MVC architecture

- Data Model
 - Instead of using XML Data Island, JavaScript Object Notation offers an example of a language neutral data model
- Communications
 - Ajax: The Asynchronous JavaScript+XML communication pattern (aka: XML-RPC) has gained recent popularity

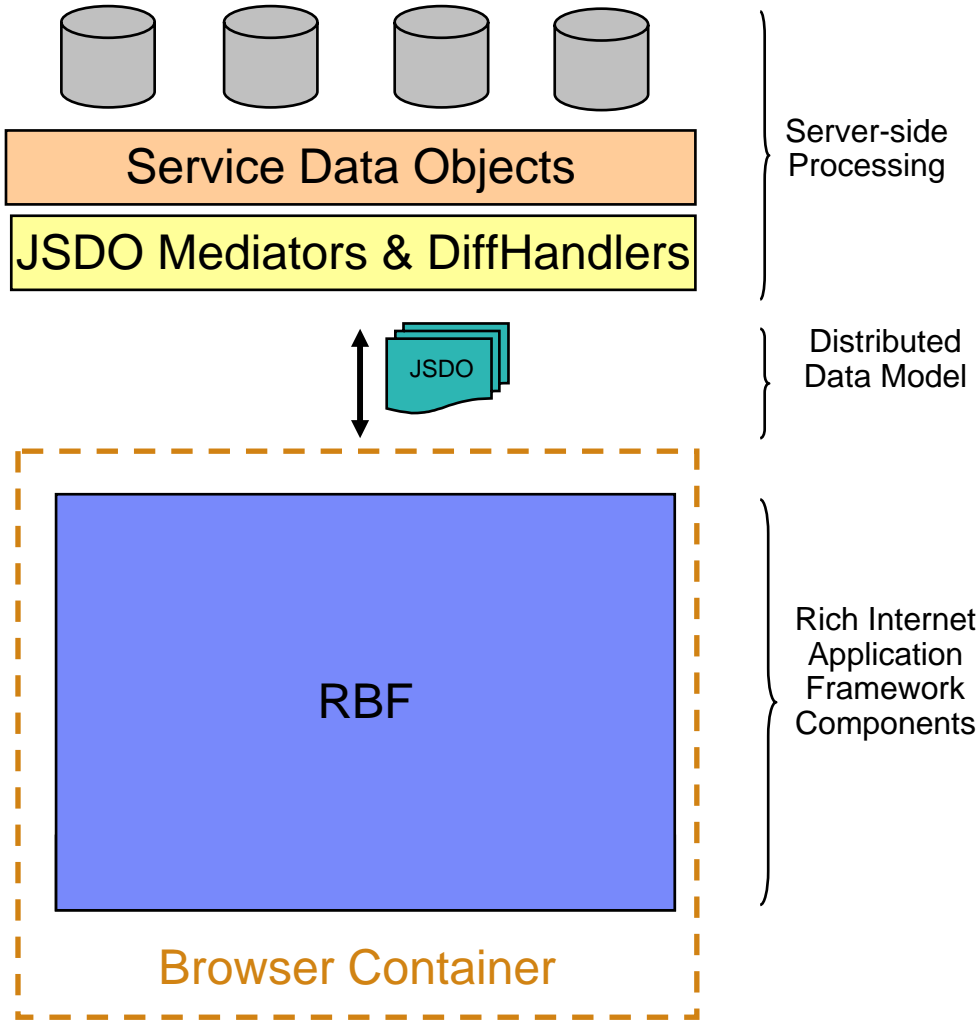
RIA Frameworks: IBM's Rich Browser Framework (RBF)



Server-side Independent JavaScript-based RIA Framework

- **Event Model**
 - Customized DOM Event extensions to supporting client side processing of user interactions and remote data updates
- **Data Model**
 - JavaScript Data Object format expected for all inbound/outbound wire formats.
- **Binders**
 - Client-side adapters for model, view and controls. These binders establish a tight coupling between the components of the framework in order to achieve a robust client-side MVC architecture.
- **UI Controls**
 - Accessible and extendable UI widget library that are bound to a specific client-side data model. This widget library can be extended to provide components that enrich the user experience
- **Communications**
 - HTML Back Channel (XML-RPC)
 - Web Services communications through the ubiquitous Flash conduit
 - Web Messaging (Real-time Push)

RBF: Server-side Independence



Data Abstraction

- Service Data Objects
 - An open source specification that provided a data programming architecture which unifies data programming across data source types, provides robust support for common application patterns, and enable applications, tools, and frameworks to more easily query, view, bind, update, and introspect data.
 - SDO has been implemented in:
 - Java by IBM
 - PHP by IBM & Zend (target PHP5.1)

Data Mediation & Change Management

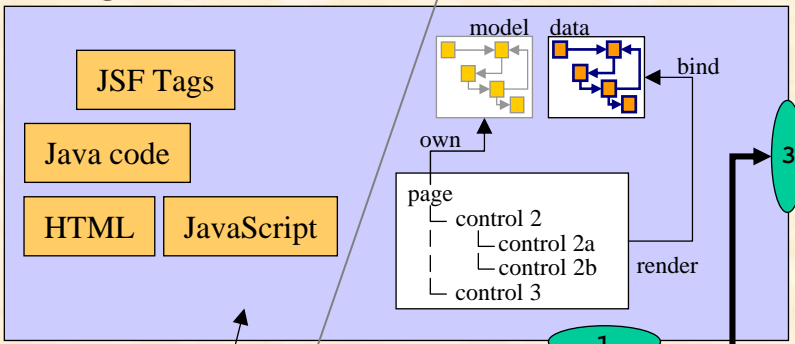
- Mediators
 - These programmatic adapters insure that the wire format of a inbound or outbound transaction from/to the client is in JSDO format.
 - Supported Languages
 - Java
 - PHP (IBM target PHP5.1)
- DiffHandlers
 - This server-side processing mechanism manages the synchronization of updates between the server and client data models
 - Supported Languages
 - Java
 - PHP (IBM target PHP5.1)

JavaScript Library (JSL): MVC Projection

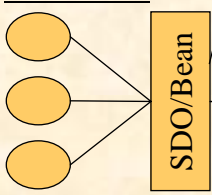
server

browser

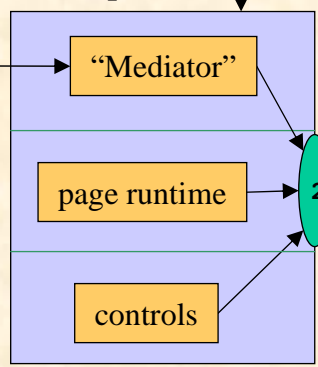
JSF Page



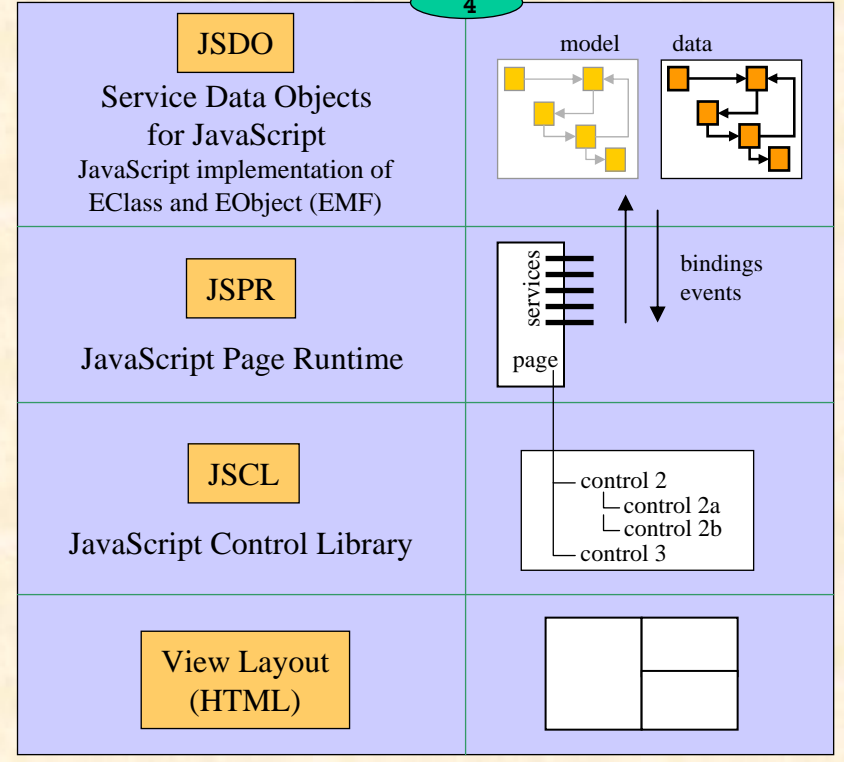
data source



OBF JSF Components



JavaScript Library (JSL)



1

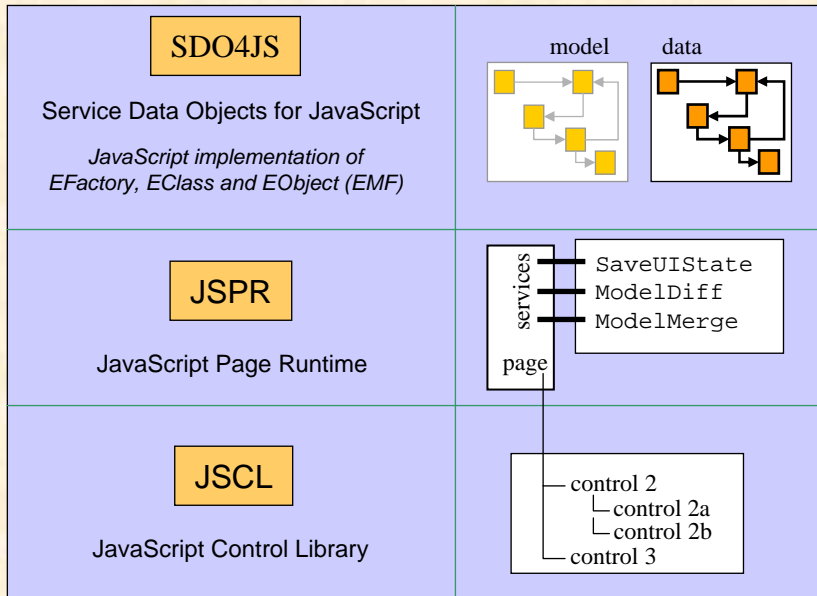
3

4

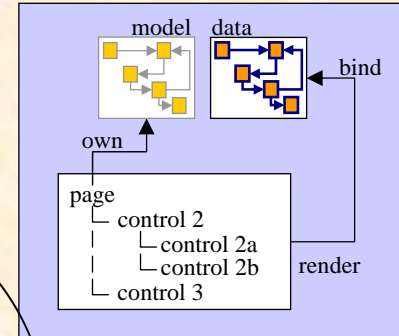
JavaScript Library (JSL): Server Integration

browser

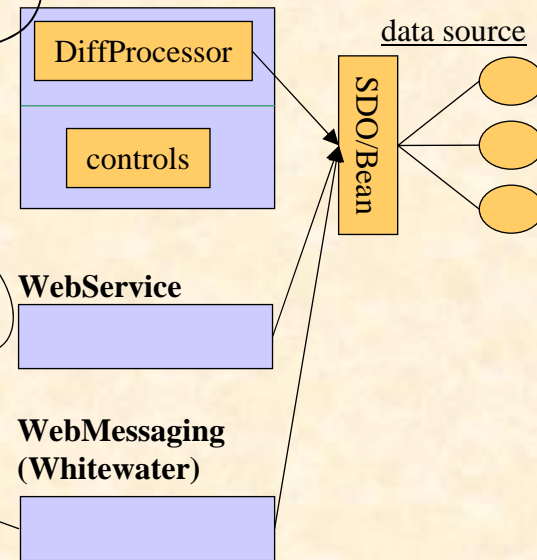
JavaScript Library (JSL)



JSF Page



OBF JSF Components



New page

A Diff, UI State
HTTP

B MergeData

WebServices
Data or Diff

C MergeData
HTTP

Demo: Toto.html

- Standalone Code demonstration
 - ▶ Browser Agnostic Demo of IBM's JavaScript Framework (JSL) with hard coded client-side data model
 - ▶ Key Messages
 - Simple JavaScript based programming model
 - Rich Data Model Definition
 - Rich Component Definition
 - MVC architectures ties together components and data through JavaScript events

- *Code Deep Dive!*



Demo: stockportfolio.php

- JSL & PHP Integration
 - ▶ PHP page used instead of standalone HTML page
 - ▶ PHP data mediator backend implemented against MySQL database
 - ▶ JSL framework deployed on server as plain .js files off web root
 - ▶ *Note: DiffHandler not yet implemented*

- Key Messages
 - JSL is back-end technology neutral (JSP, ASP, PHP, Ruby, Perl, etc)
 - Client-side JSDO generated and synced with any backend data abstraction

- *Code Deep Dive!*



Demos: Laszlo / JSL Integration

- Servlet Example
 - ▶ Collaboration of Laszlo and JSL controls on the same page sharing common data and event models

- Portal Example
 - ▶ Same collaboration within a J2EE portal infrastructure

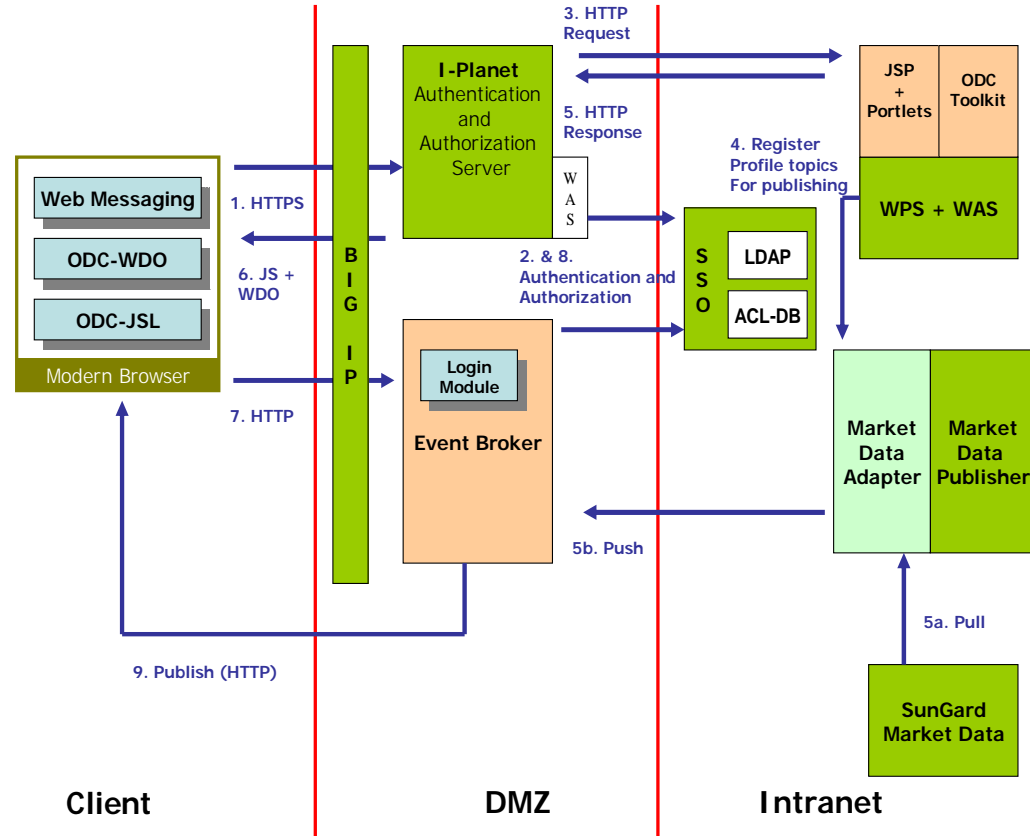
- Key Messages
 - Generic JSL component API allows developers to use any browser friendly technology to implement UI widgets.



Demos: CGMI

- Web Messaging

- Delivering real-time subscription based financial market data to a broad range of consumers who are members of Smith Barney's Access Retail Brokerage Web Site
- Implemented using HTTP Tunneling
- Note: Demo code is based on Applet, current code is part of JavaScript framework



- Key Messages

- Ajax is just one type of RIA communications (XMLHttpRequest). JSL also implements (supports) Web Services (via Flash Remoting) and Web Messaging.



Demos: RAD Tooling

- Visual Programming with JSL
 - ▶ Rational Application Developer v6 implements JSF. IBM's implementation of JSF embeds our JSL

- Key Messages
 - ▶ RAD is a sophisticated visual development IDE that demonstrates the end-2-end toolability of the framework.

- Trial Version of RAD is available @
 - ▶ <http://www-128.ibm.com/developerworks/downloads/r/rad>



Conclusion

- A complete end-2-end RIA framework is important for the industry.
- Such a framework needs:
 - ▶ Strong browser support (JS, HTML, CSS are key technologies)
 - ▶ To be backend agnostic (Java, PHP, ASP, Perl, etc)
 - ▶ To support any browser based technology for widget development (Laszlo, XUL, Flash, DHTML, etc)
 - ▶ To support Rich Backend connectivity
 - XMLHttpRequest
 - Web Service
 - Web Messaging
 - ▶ To be visually toolable
 - ▶ To be available in the open source community so that it can be extended
- OpenRBF (Open Rich Browser Framework) offers these benefits ... BUT
 - ▶ We need support to open source it
 - ▶ We help to build PHP backend for it
 - ▶ We need help to visually tool PHP backend



Feedback

- If you have requirements desires pertaining to RIA frameworks please send us an email --- we really need your feedback
 - ▶ Dan Gisolfi: gisolfi@us.ibm.com
 - ▶ Laurent Hasson: ldhasson@us.ibm.com

