IBM Software Group

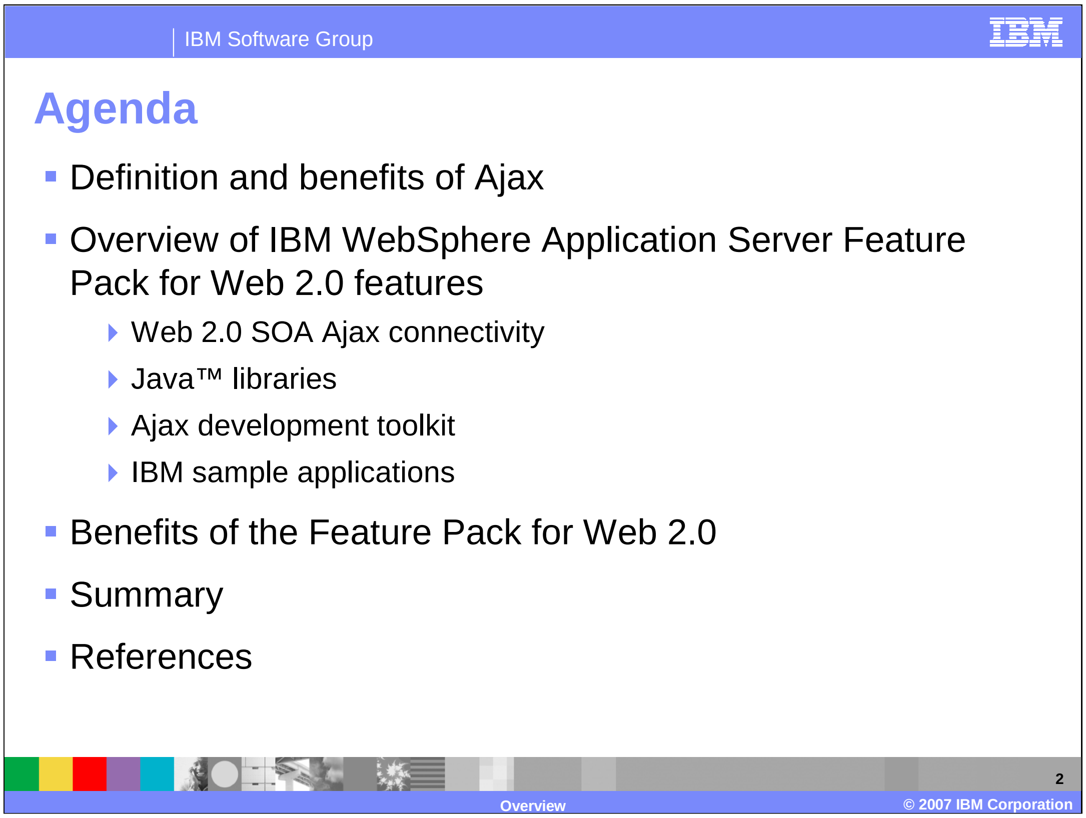
# IBM WebSphere® Application Server Feature Pack for Web 2.0

## *Overview*

business on demand.

Updated December 18, 2007

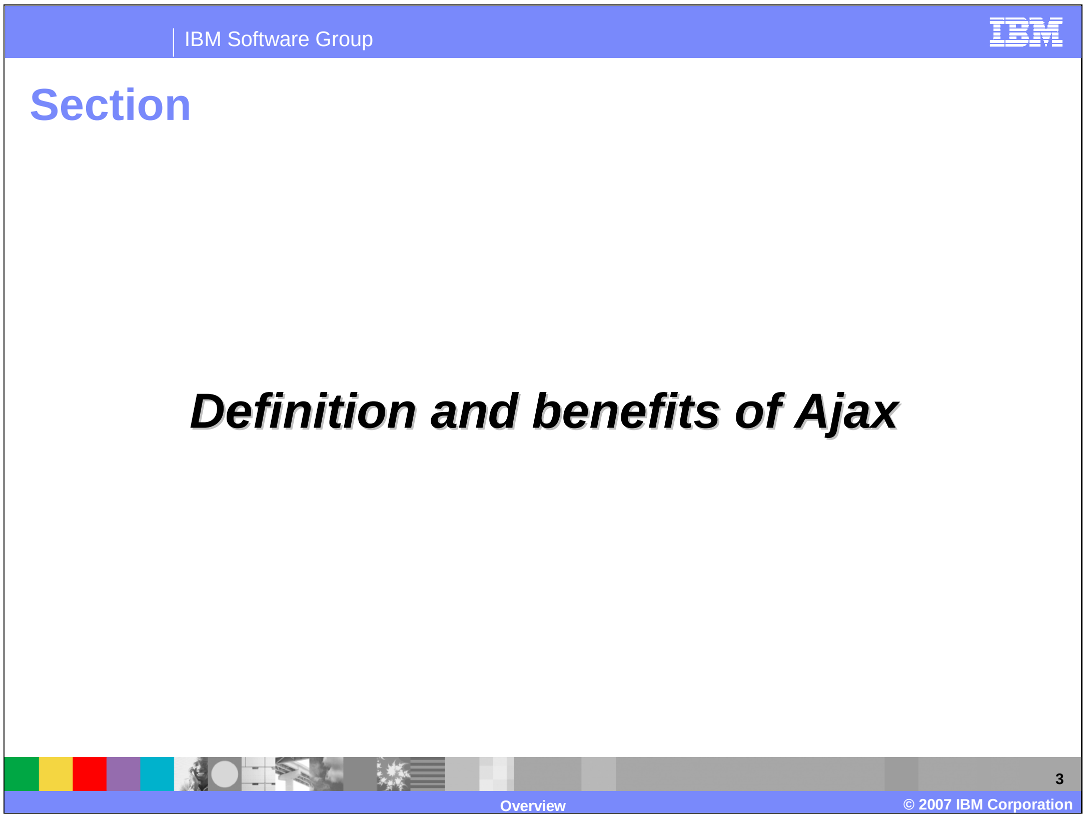
This presentation is an overview of IBM WebSphere Application Server Feature Pack for Web 2.0



## Agenda

- Definition and benefits of Ajax
- Overview of IBM WebSphere Application Server Feature Pack for Web 2.0 features
  - Web 2.0 SOA Ajax connectivity
  - Java™ libraries
  - Ajax development toolkit
  - IBM sample applications
- Benefits of the Feature Pack for Web 2.0
- Summary
- References

The agenda for this presentation entails a recap to what Ajax is and its benefits. It will then be followed by the overview of IBM WebSphere Application Server Feature Pack for Web 2.0 features, Dojo Toolkit and the benefits of the feature pack.



IBM Software Group

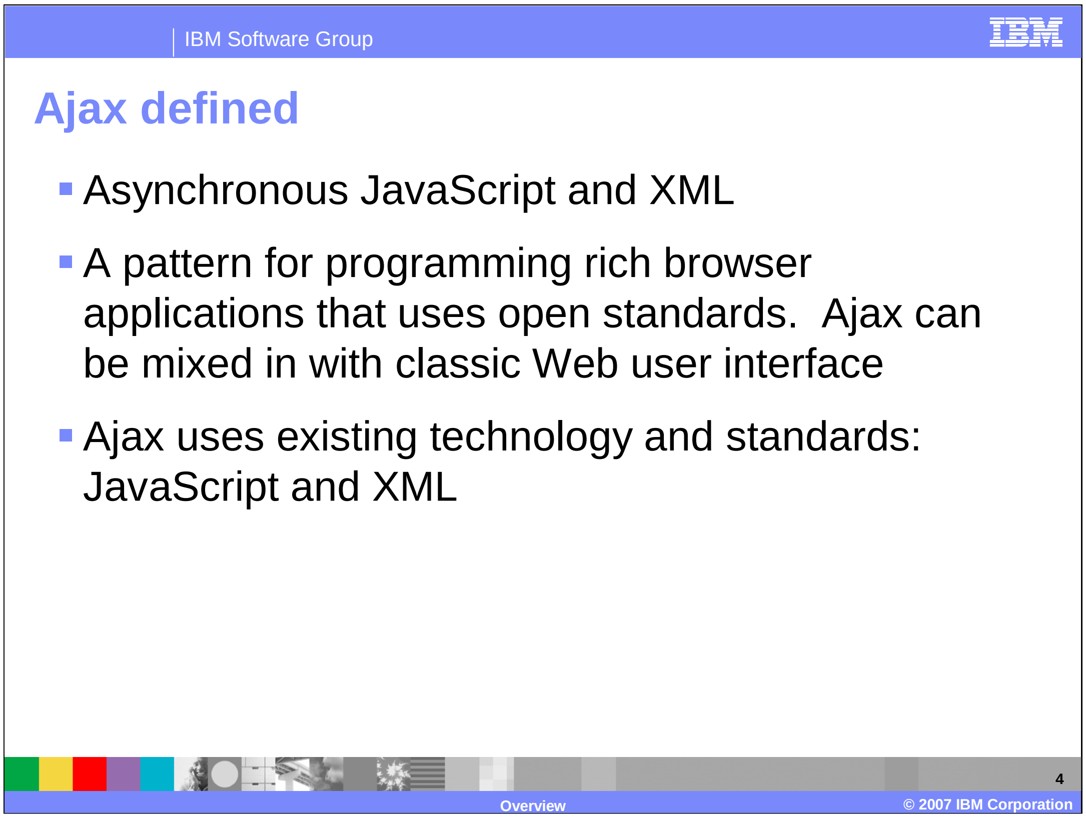
## Section

### *Definition and benefits of Ajax*

3

Overview

This section describes Ajax and its benefits.



IBM Software Group

## Ajax defined

- Asynchronous JavaScript and XML
- A pattern for programming rich browser applications that uses open standards.  Ajax can be mixed in with classic Web user interface
- Ajax uses existing technology and standards: JavaScript and XML

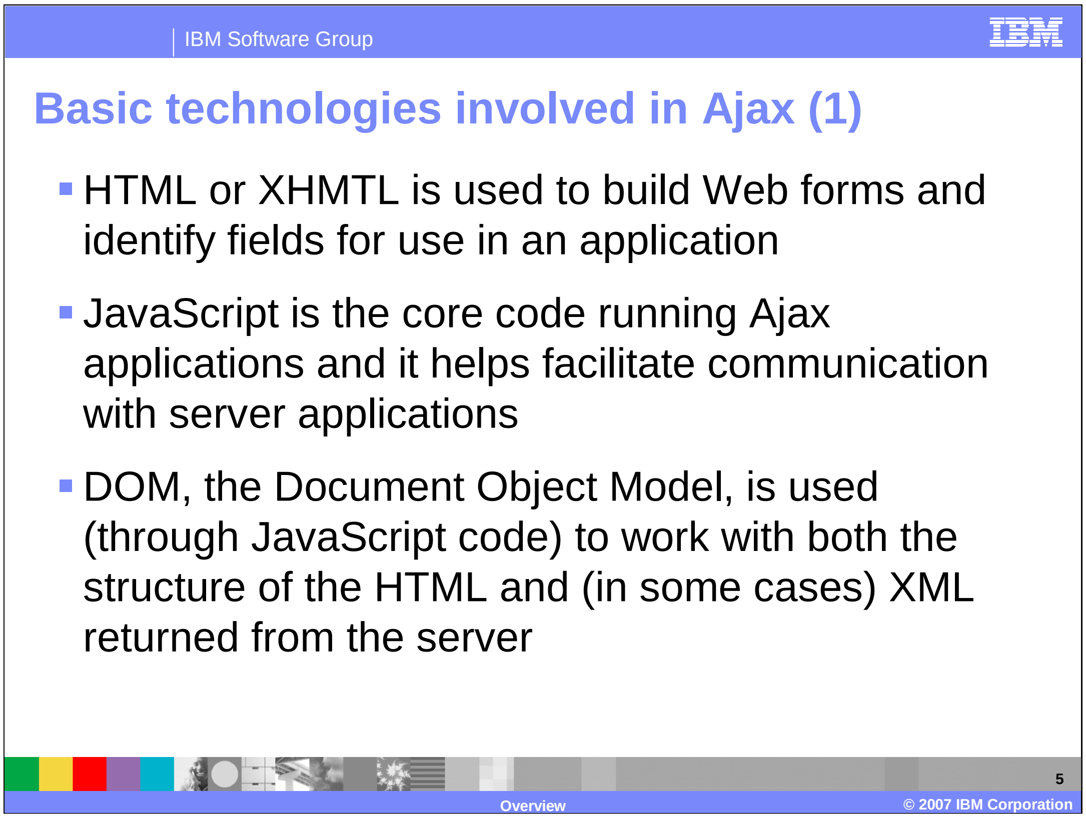
4

Overview

Ajax means for Asynchronous JavaScript and XML. It is a Web development technique for creating interactive Web applications that make Web pages more responsive. Exchanging small amounts of data with the server means that the entire Web page does not have to be reloaded each time you make a change. This increases the Web page's interactivity, speed, and usability. In short, Ajax enables a dynamic, asynchronous Web experience without the need for page refreshes.

In Ajax programming, HTML - or XHTML - and CSS are used for marking up and styling information. Dynamic display and interaction use JavaScript and the Document Object Model, or DOM.

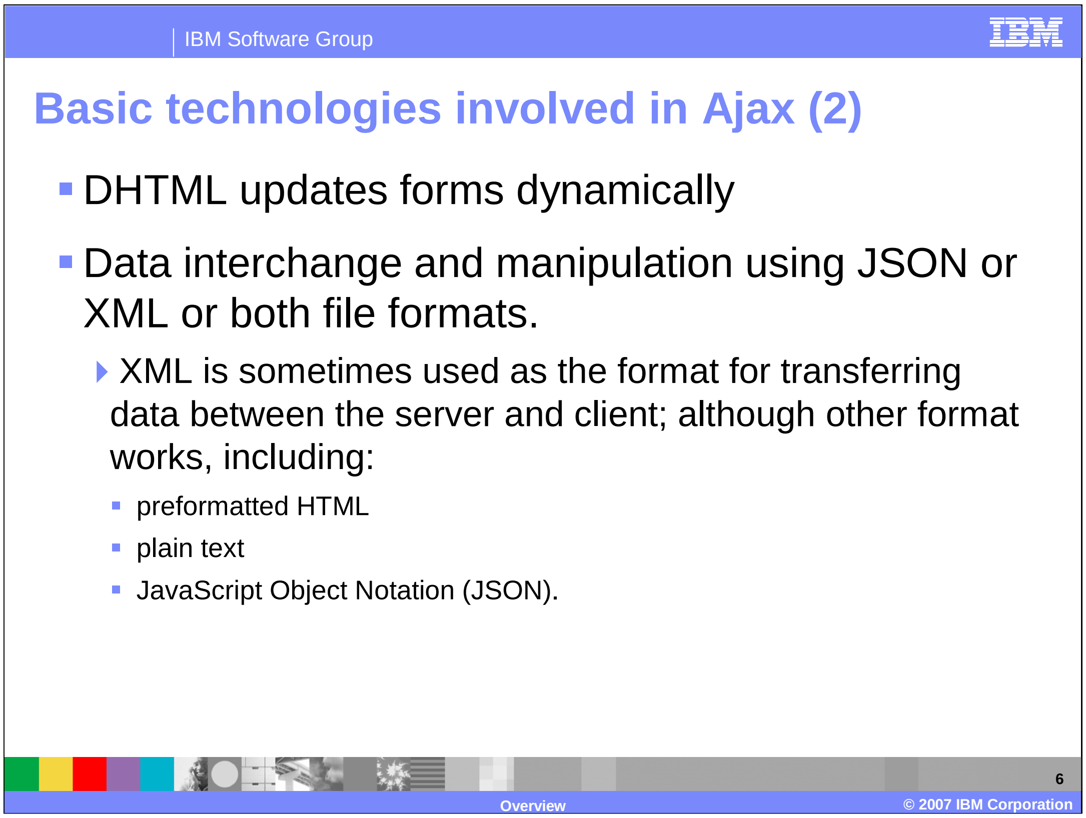
Data interchange and manipulation are accomplished using JSON or XML or both file formats. The XMLHttpRequest object can be used to exchange data asynchronously with the Web server.



## Basic technologies involved in Ajax (1)

- HTML or XHMTL is used to build Web forms and identify fields for use in an application
- JavaScript is the core code running Ajax applications and it helps facilitate communication with server applications
- DOM, the Document Object Model, is used (through JavaScript code) to work with both the structure of the HTML and (in some cases) XML returned from the server

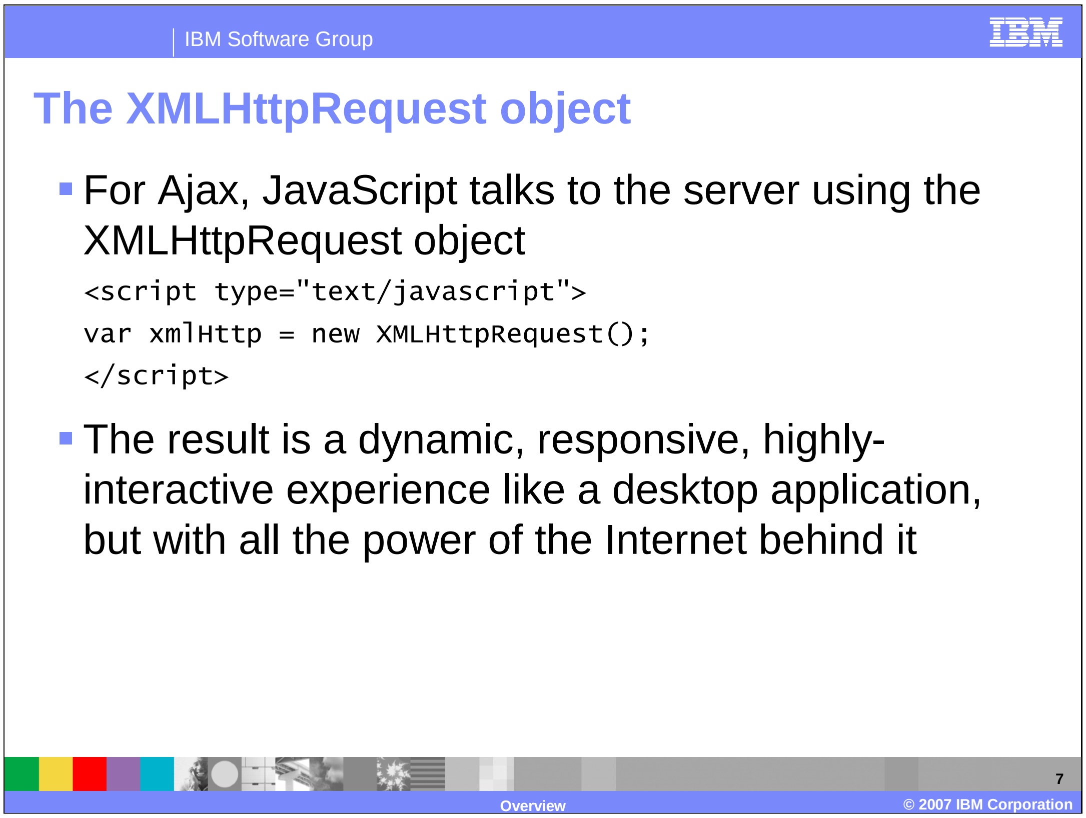
The basic technologies in Ajax include XHTML and CSS which provide a standards-based presentation. JavaScript is also one of the technologies and it binds everything together. The DOM provides dynamic display and interaction. XML, XSLT and JSON provide data interchange and manipulation.



## Basic technologies involved in Ajax (2)

- DHTML updates forms dynamically
- Data interchange and manipulation using JSON or XML or both file formats.
  - XML is sometimes used as the format for transferring data between the server and client; although other format works, including:
    - preformatted HTML
    - plain text
    - JavaScript Object Notation (JSON).

Also included in the Ajax technologies is Dynamic HTML, or DHTML, which updates forms dynamically. XML can be used as the format for transferring data between the server and client; although several formats work, including preformatted HTML, plain text, and JavaScript Object Notation, or JSON.



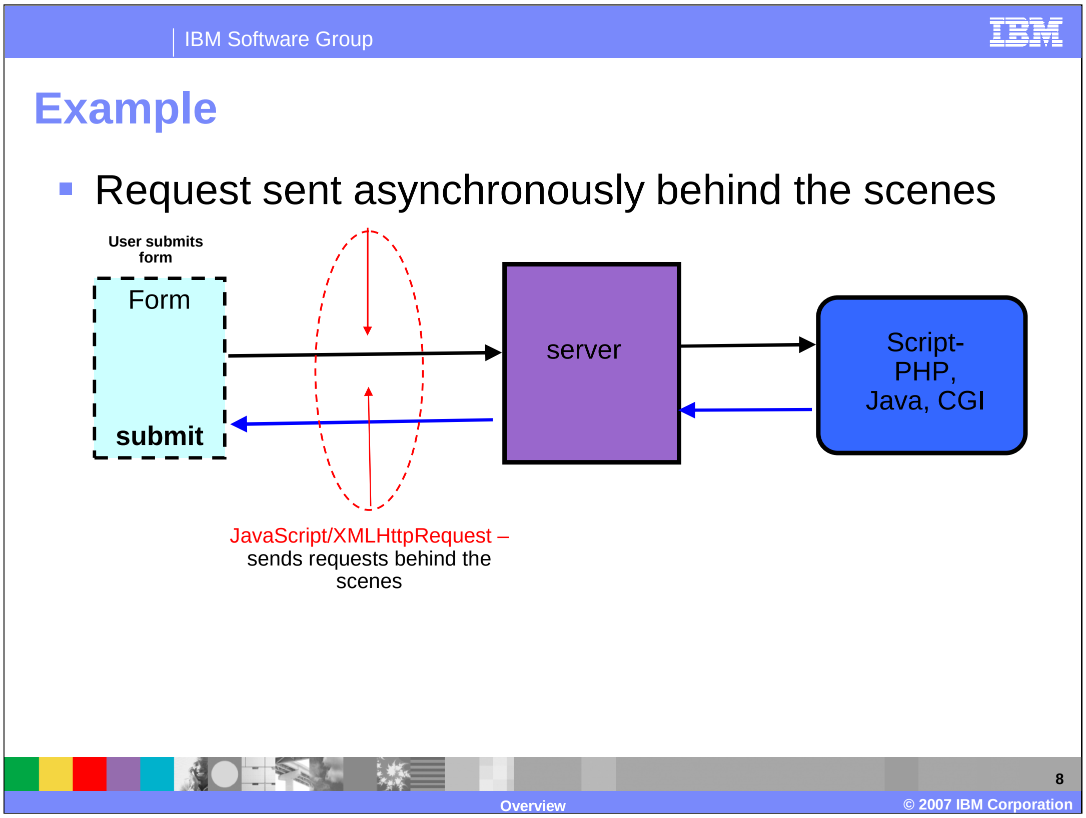
## The XMLHttpRequest object

- For Ajax, JavaScript talks to the server using the XMLHttpRequest object

```
<script type="text/javascript">
var xmlHttp = new XMLHttpRequest();
</script>
```

- The result is a dynamic, responsive, highly-interactive experience like a desktop application, but with all the power of the Internet behind it

The core of Ajax technology is a JavaScript object XMLHttpRequest . With an XMLHttpRequest object, you can use JavaScript to make a request to the server and process the response. Examples of applications that use this technology include Google's Gmail service, Google's Suggest dynamic lookup interface, and the MapQuest dynamic map interface. It is the JavaScript technology through the XMLHttpRequest object that talks to the server which is not the normal application flow and it is where Ajax gets much of its magic.



IBM Software Group

## Example

- Request sent asynchronously behind the scenes

User submits form

Form

submit

server

Script-<br>PHP,<br>Java, CGI

JavaScript/XMLHttpRequest – sends requests behind the scenes

8

Overview

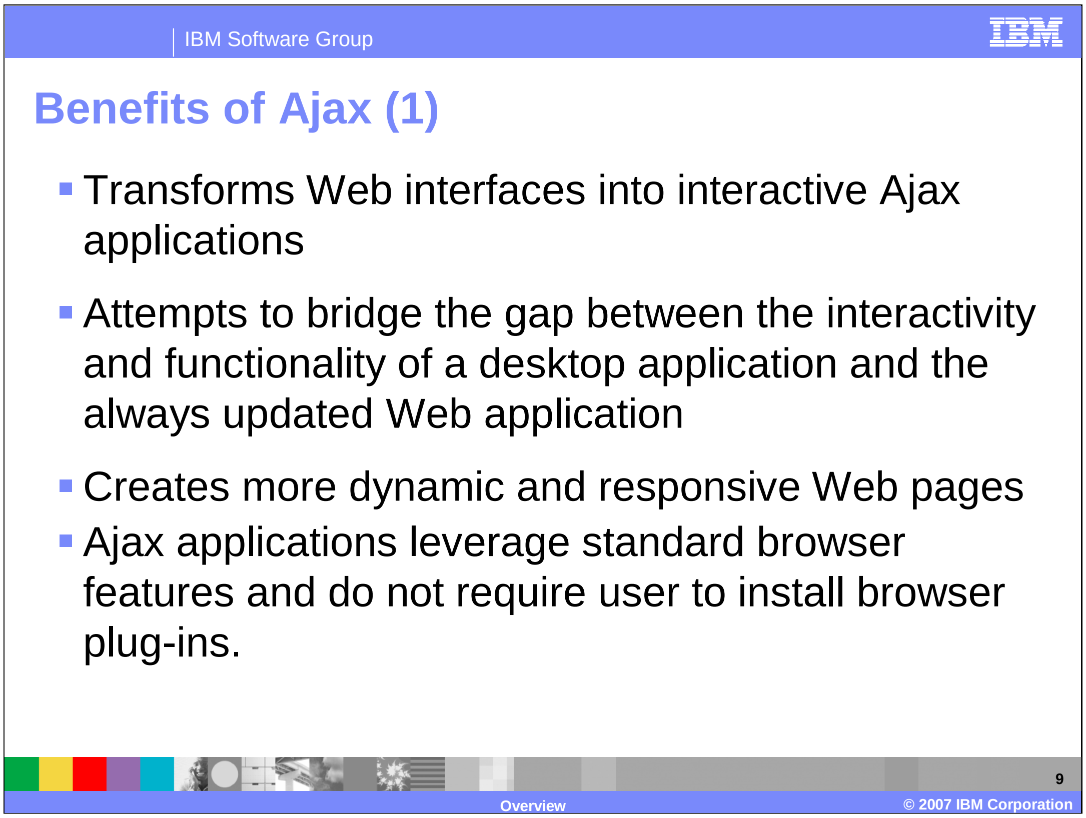
© 2007 IBM Corporation

This example shows how a request is sent asynchronously with Ajax. In a normal Web application, when you fill out a form and submit it, the entire form is sent to the server. The server then passes on processing to a script (typically PHP or Java or maybe a CGI process or something similar), and when the script is done, it sends back a completely new page.

Ajax puts JavaScript technology and the XMLHttpRequest object between the Web form and the server so when you fill out a form, that data is sent to some JavaScript code and not directly to the server.

The server then sends data back to the JavaScript code (still standing in for the Web form) which decides what to do with that data. It can update form fields on the fly, giving that immediate feeling to your application.

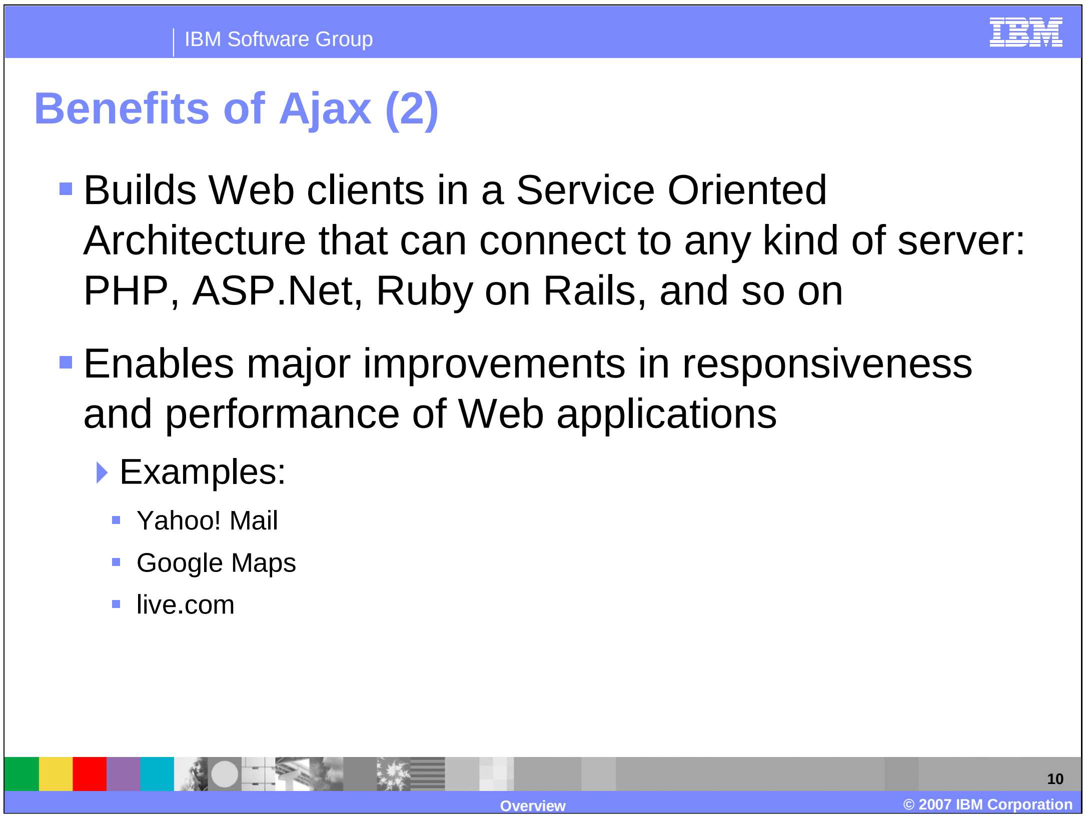
You are getting new data without the form being submitted or refreshed. The JavaScript code can get the data, perform some calculations, and send another request, all without user intervention! This is the power of XMLHttpRequest. It can talk back and forth with a server all it wants, without you ever knowing about what's really going on. The result is a dynamic, responsive, highly-interactive experience like a desktop application, but with all the power of the Internet behind it.



## Benefits of Ajax (1)

- Transforms Web interfaces into interactive Ajax applications
- Attempts to bridge the gap between the interactivity and functionality of a desktop application and the always updated Web application
- Creates more dynamic and responsive Web pages
- Ajax applications leverage standard browser features and do not require user to install browser plug-ins.

Ajax bridges the gap between the desktop applications and the Web applications. Web pages are more responsive by seamlessly exchanging small amounts of data with the server. The entire Web page does not have to be reloaded each time a change is made thus increasing Web page interactivity, speed, and usability.  Ajax applications leverage standard browser features and do not require you to install browser plug-ins.



## Benefits of Ajax (2)

- Builds Web clients in a Service Oriented Architecture that can connect to any kind of server: PHP, ASP.Net, Ruby on Rails, and so on
- Enables major improvements in responsiveness and performance of Web applications
  - Examples:
    - Yahoo! Mail
    - Google Maps
    - live.com

Another benefit of Ajax is that it builds Web clients in a Service Oriented Architecture that can connect to any kind of server: PHP, ASP. Net, Ruby on Rails, and so on. It also enables major improvements in responsiveness and performance of Web applications.

IBM Software Group

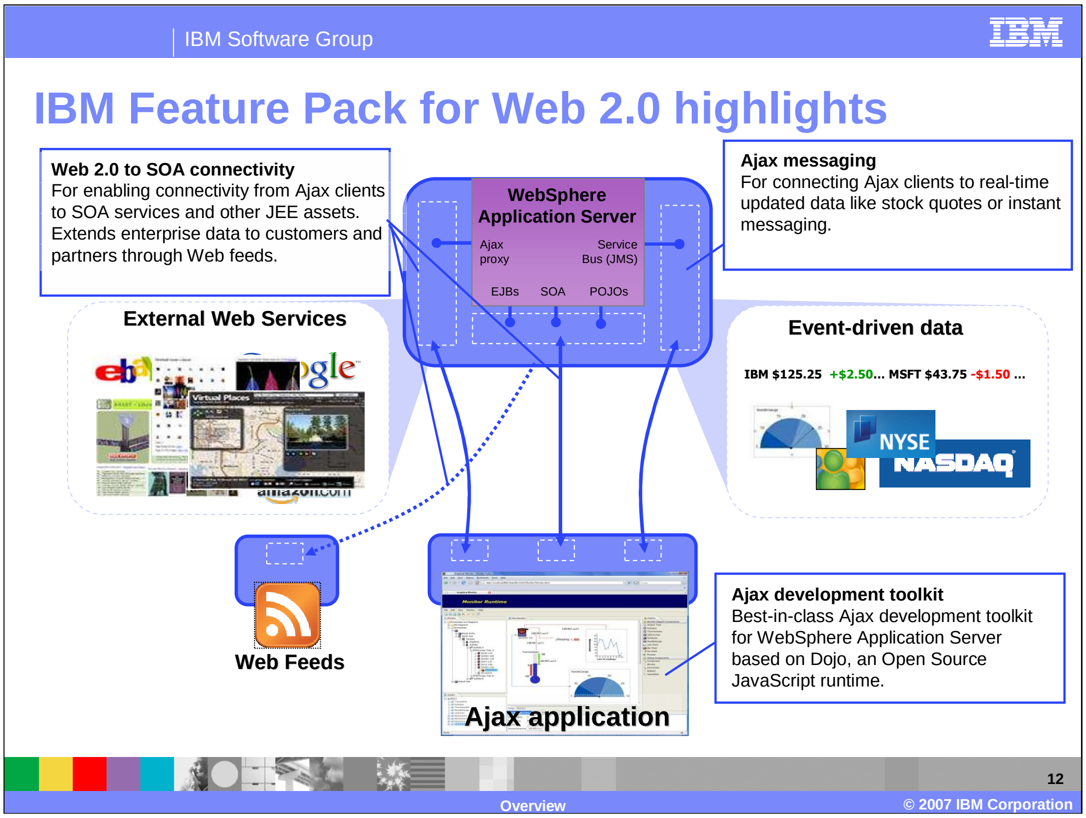
## Section

***Overview of the IBM WebSphere Application Server Feature Pack for Web 2.0 features***

11

Overview

This section covers the features of the IBM WebSphere Application Server Feature Pack for Web 2.0



## IBM 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.

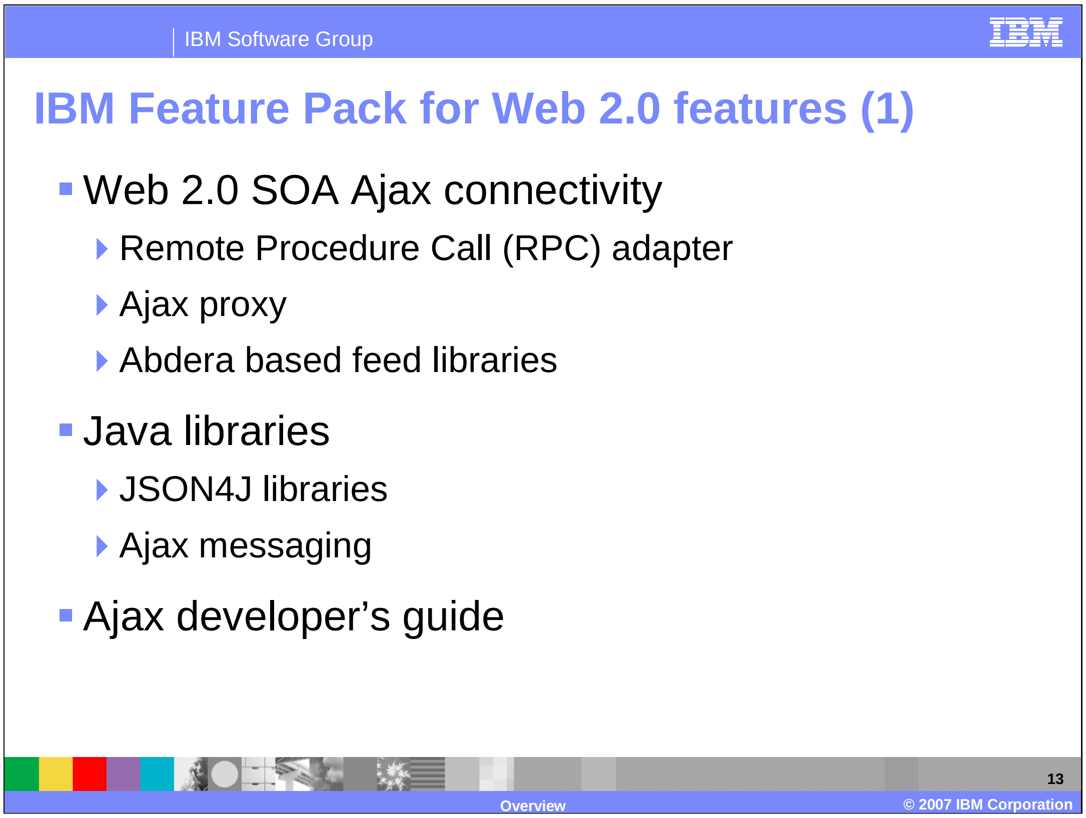
**Ajax development toolkit**
Best-in-class Ajax development toolkit for WebSphere Application Server based on Dojo, an Open Source JavaScript runtime.

This picture shows the feature pack highlights and how the pieces of the feature pack fit together.

There is the **Web 2.0 to SOA connectivity** for enabling connectivity from Ajax clients to SOA services and other J2EE assets. It extends enterprise data to customers and partners through Web feeds.

Then there is the **Ajax messaging** for connecting Ajax clients to real-time updated data like stock quotes or instant messaging.

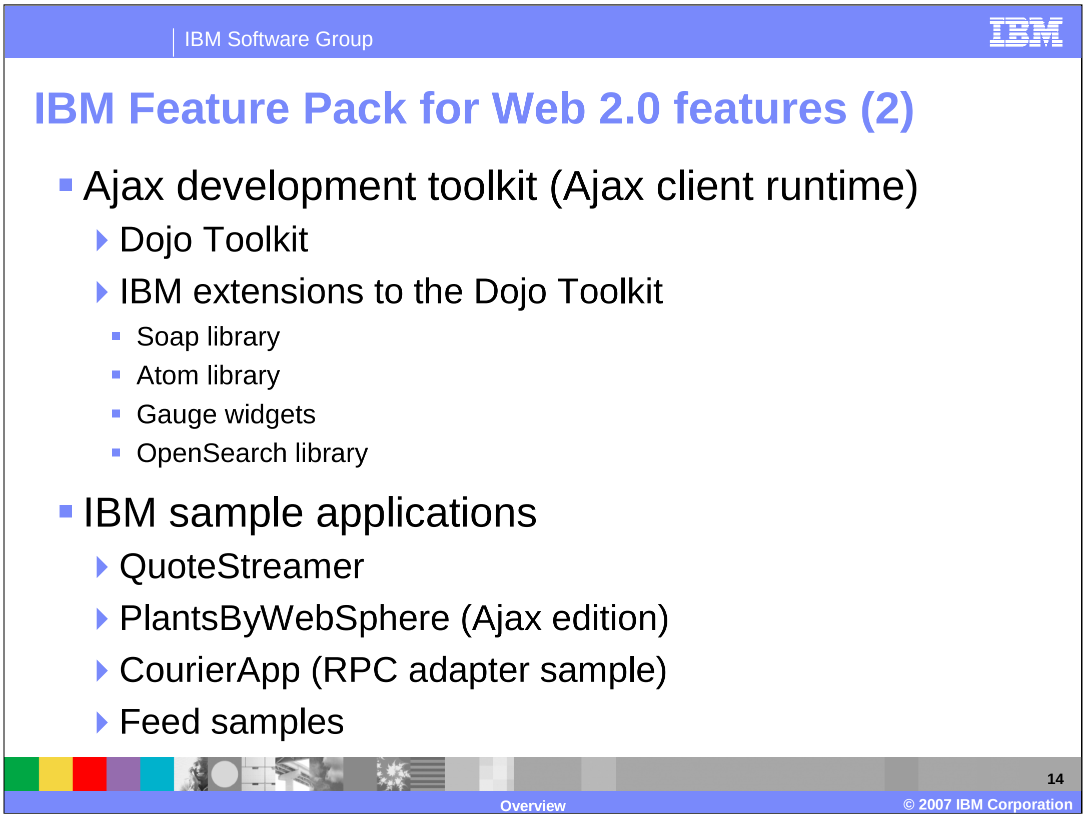
The **Ajax development toolkit** for WebSphere Application Server is based on Dojo, and is an open source JavaScript runtime.



## IBM Feature Pack for Web 2.0 features (1)

- Web 2.0 SOA Ajax connectivity
  - Remote Procedure Call (RPC) adapter
  - Ajax proxy
  - Abdera based feed libraries
- Java libraries
  - JSON4J libraries
  - Ajax messaging
- Ajax developer’s guide

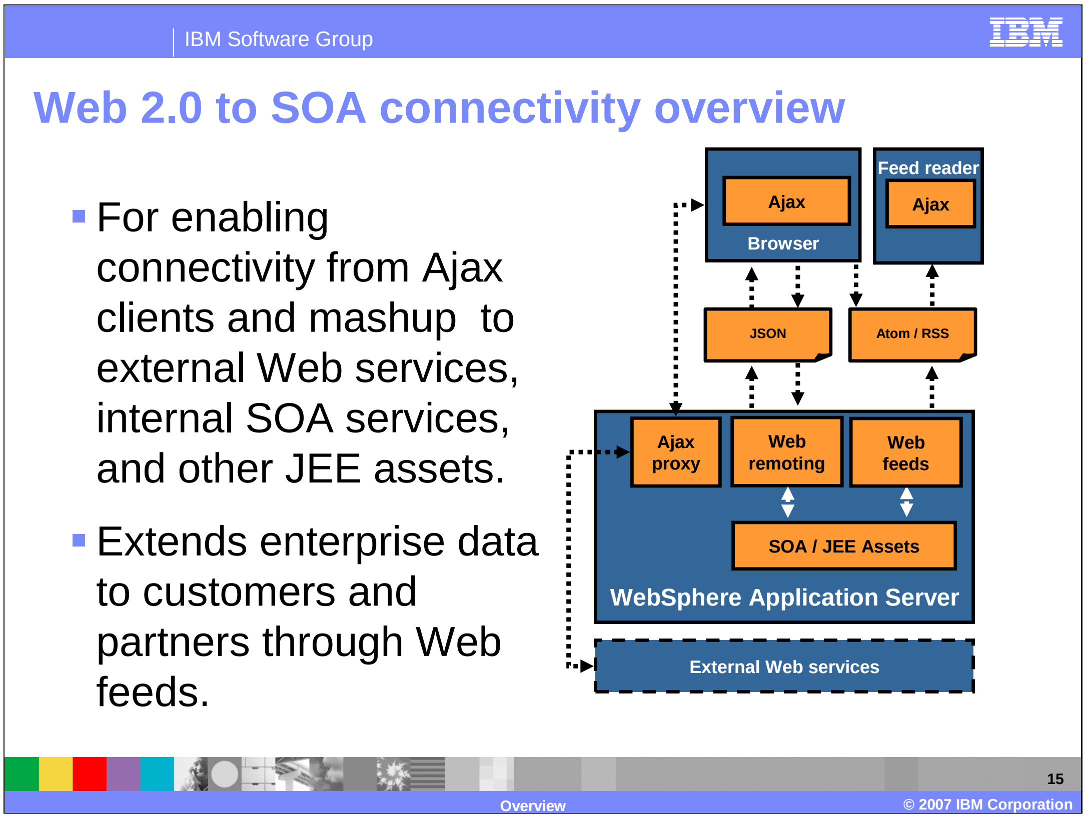
Features in the feature pack include Web 2.0 SOA Ajax connectivity, JSON4J and Ajax messaging Java libraries, and the Ajax developer’s guide.



## IBM Feature Pack for Web 2.0 features (2)

- Ajax development toolkit (Ajax client runtime)
  - Dojo Toolkit
  - IBM extensions to the Dojo Toolkit
    - Soap library
    - Atom library
    - Gauge widgets
    - OpenSearch library
- IBM sample applications
  - QuoteStreamer
  - PlantsByWebSphere (Ajax edition)
  - CourierApp (RPC adapter sample)
  - Feed samples

The Ajax development toolkit consists of the Dojo Toolkit, and IBM extensions to the Dojo Toolkit, like the Atom Library, Gauge Widgets, OpenSearch Library, and the Soap Library. Sample applications are listed here.



IBM Software Group

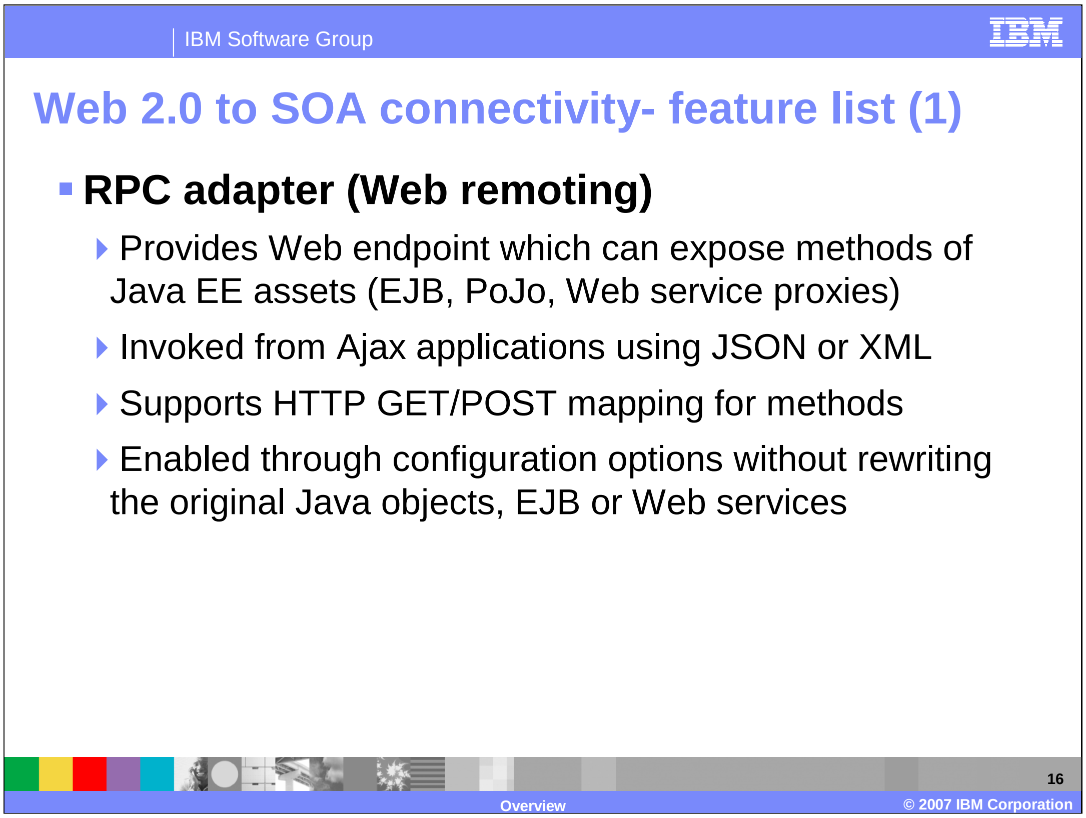
# Web 2.0 to SOA connectivity overview

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

15

Overview

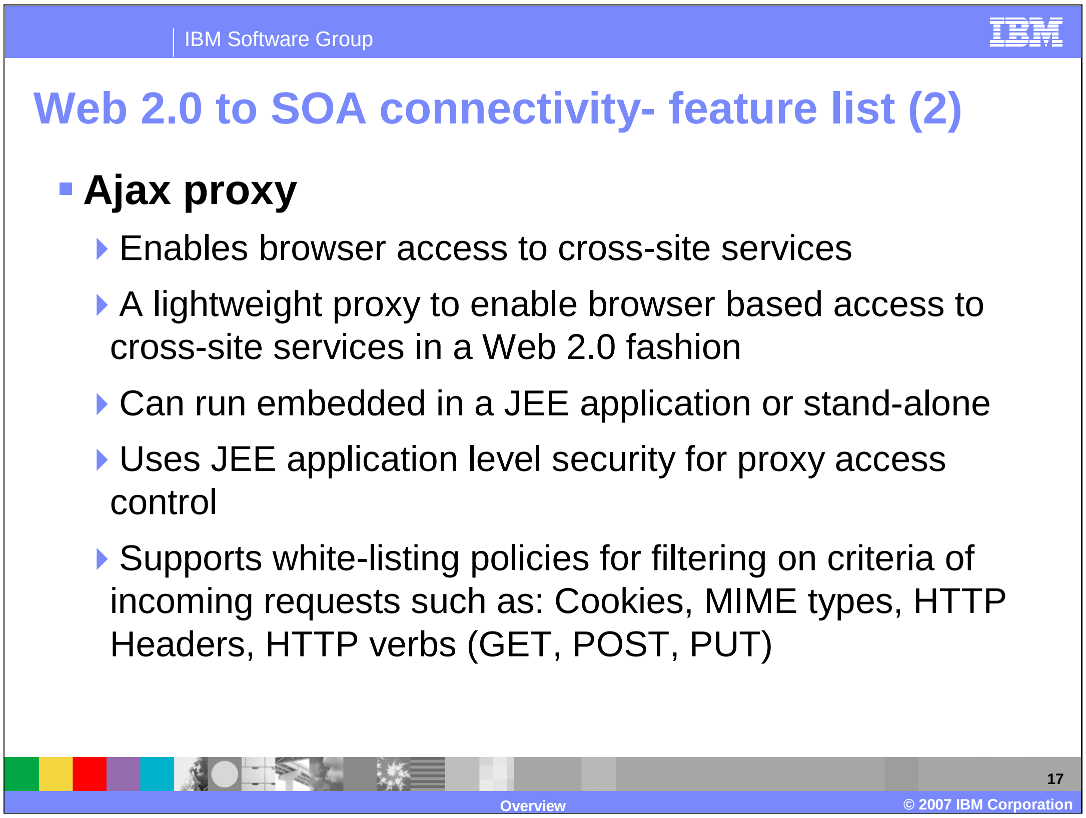
The Web 2.0 to SOA connectivity lets you connect from Ajax clients to external Web services, internal SOA services, and other J2EE assets. It also extends enterprise data to customers and partners through Web feeds.



## Web 2.0 to SOA connectivity- feature list (1)

- **RPC adapter (Web remoting)**
  - Provides Web endpoint which can expose methods of Java EE assets (EJB, PoJo, Web service proxies)
  - Invoked from Ajax applications using JSON or XML
  - Supports HTTP GET/POST mapping for methods
  - Enabled through configuration options without rewriting the original Java objects, EJB or Web services

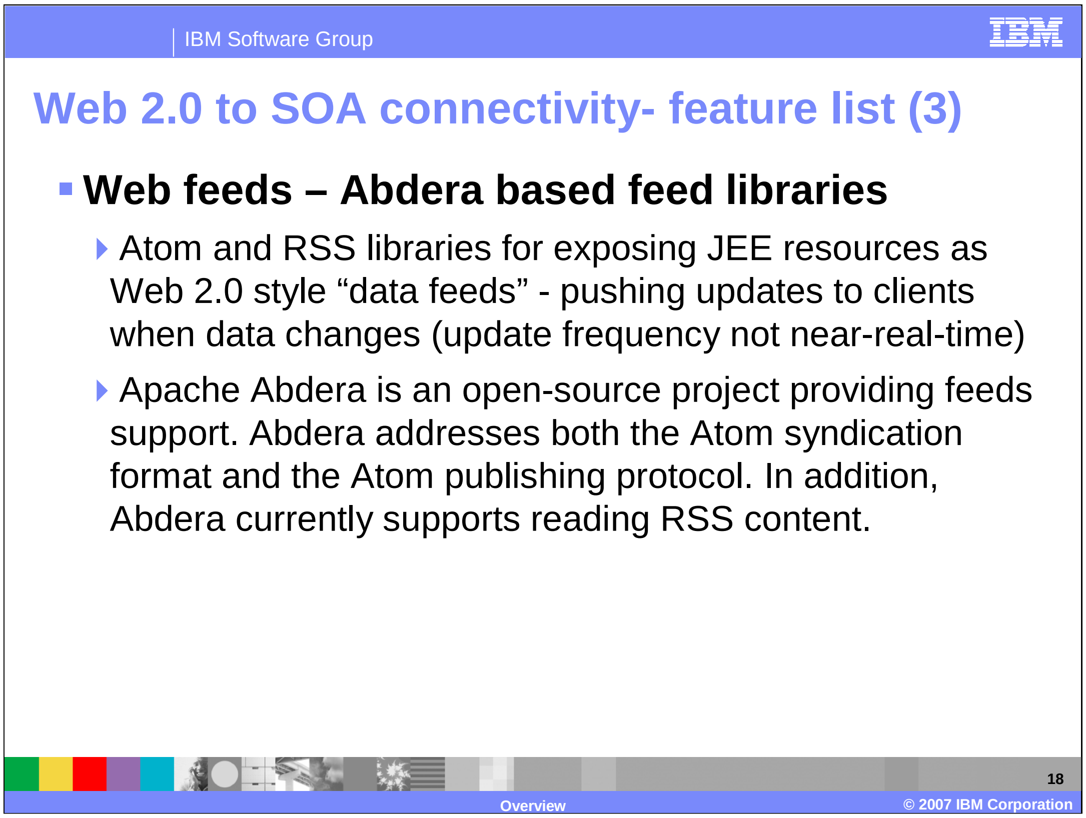
Web remoting is one of the features for Web 2.0 to SOA connectivity. Web remoting provides a lightweight Web endpoint which can expose methods of Java EE assets. Web remoting can also be easily invoked from Ajax applications using JSON or XML formats and supports HTTP GET/POST mapping for methods. It is enabled through simple configuration options so you do not need to rewrite the original Java objects, EJB or Web services.



## Web 2.0 to SOA connectivity- feature list (2)

- **Ajax proxy**
  - Enables browser access to cross-site services
  - A lightweight proxy to enable browser based access to cross-site services in a Web 2.0 fashion
  - Can run embedded in a JEE application or stand-alone
  - Uses JEE application level security for proxy access control
  - Supports white-listing policies for filtering on criteria of incoming requests such as: Cookies, MIME types, HTTP Headers, HTTP verbs (GET, POST, PUT)

The Ajax proxy is a lightweight proxy to enable browser based access to cross-site services. The proxy can run embedded within a J2EE application or stand-alone and it uses JEE application level security for proxy access control. The Ajax proxy also has support for white-listing policies for filtering on criteria of incoming requests such as: Cookies, MIME types, HTTP Headers, HTTP verbs.



IBM Software Group

## Web 2.0 to SOA connectivity- feature list (3)

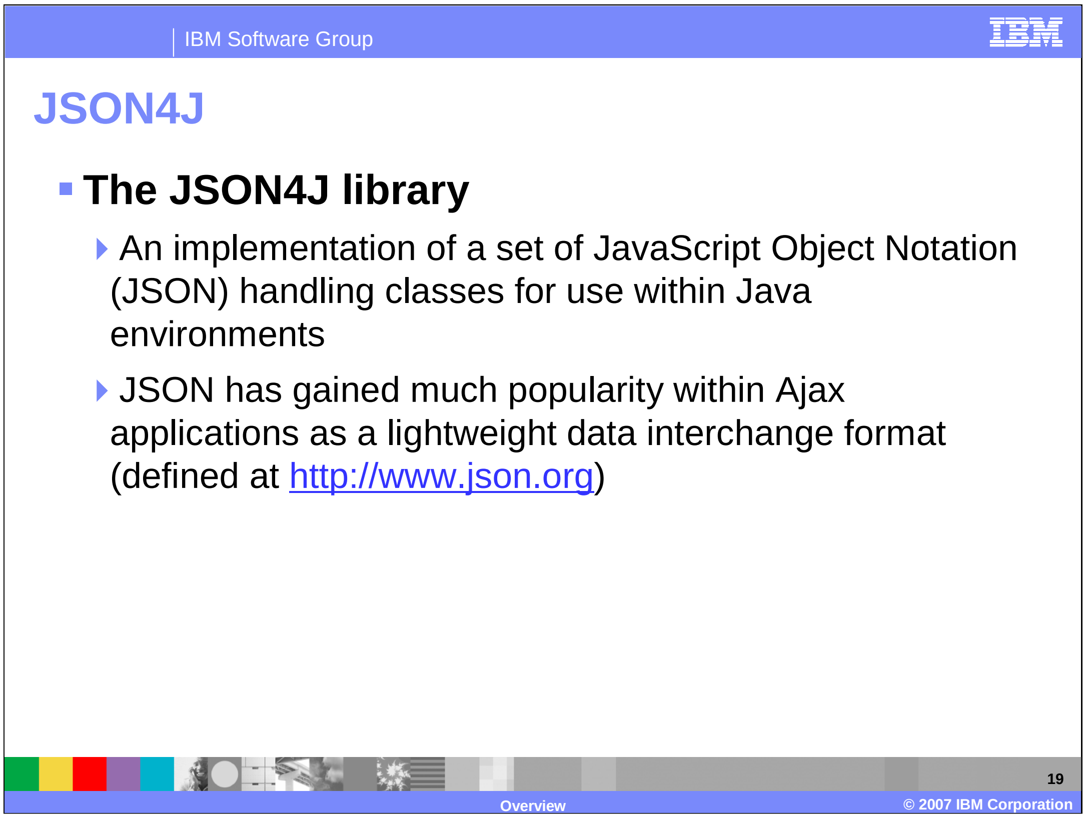
- **Web feeds – Abdera based feed libraries**
  - Atom and RSS libraries for exposing JEE resources as Web 2.0 style “data feeds” - pushing updates to clients when data changes (update frequency not near-real-time)
  - Apache Abdera is an open-source project providing feeds support. Abdera addresses both the Atom syndication format and the Atom publishing protocol. In addition, Abdera currently supports reading RSS content.

18

Overview

© 2007 IBM Corporation

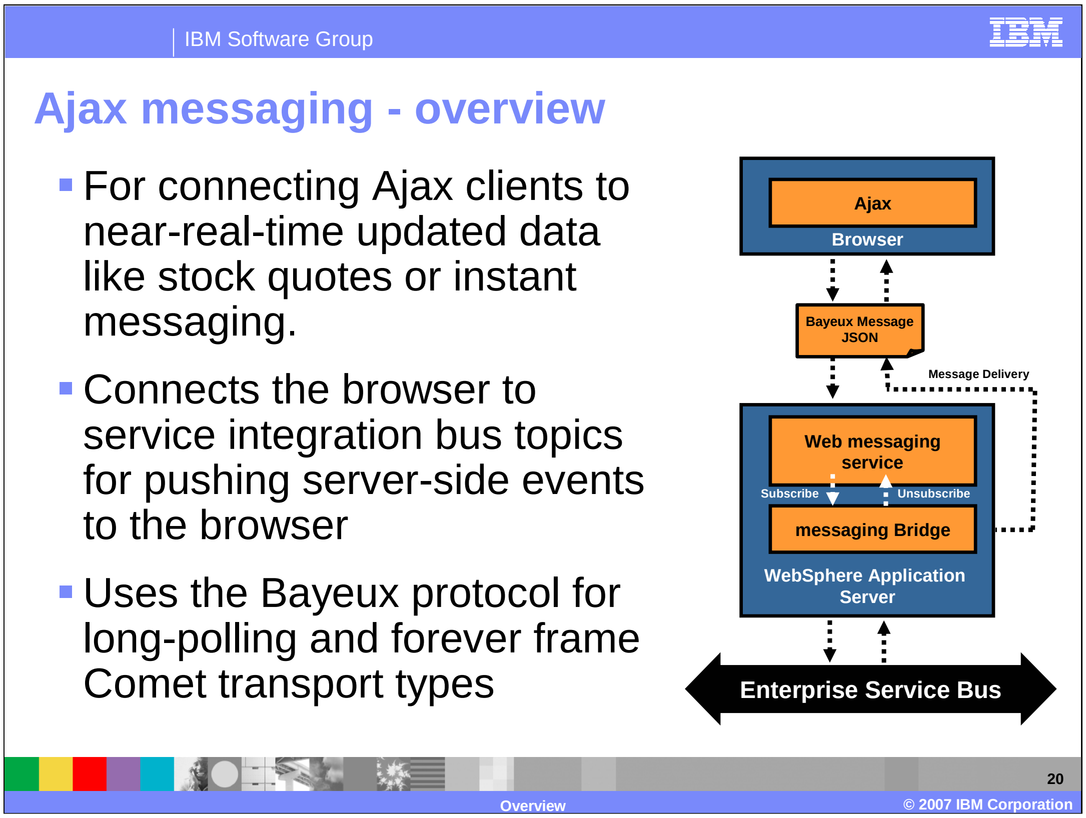
Web feeds are also part of the Web 2.0 to SOA connectivity. Atom and RSS libraries allow you to expose JEE resources as Web 2.0 style “data feeds.” This lets clients subscribe to updates and get notified when the data changes, as long as the update frequency is not near-real-time. Apache Abdera is an open-source project providing feeds support. Abdera addresses both the Atom syndication format and the Atom publishing protocol. In addition, Abdera currently supports reading RSS content.



## JSON4J

- **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)

Another feature for Web 2.0 to SOA connectivity is the JSON4J library. It is 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.

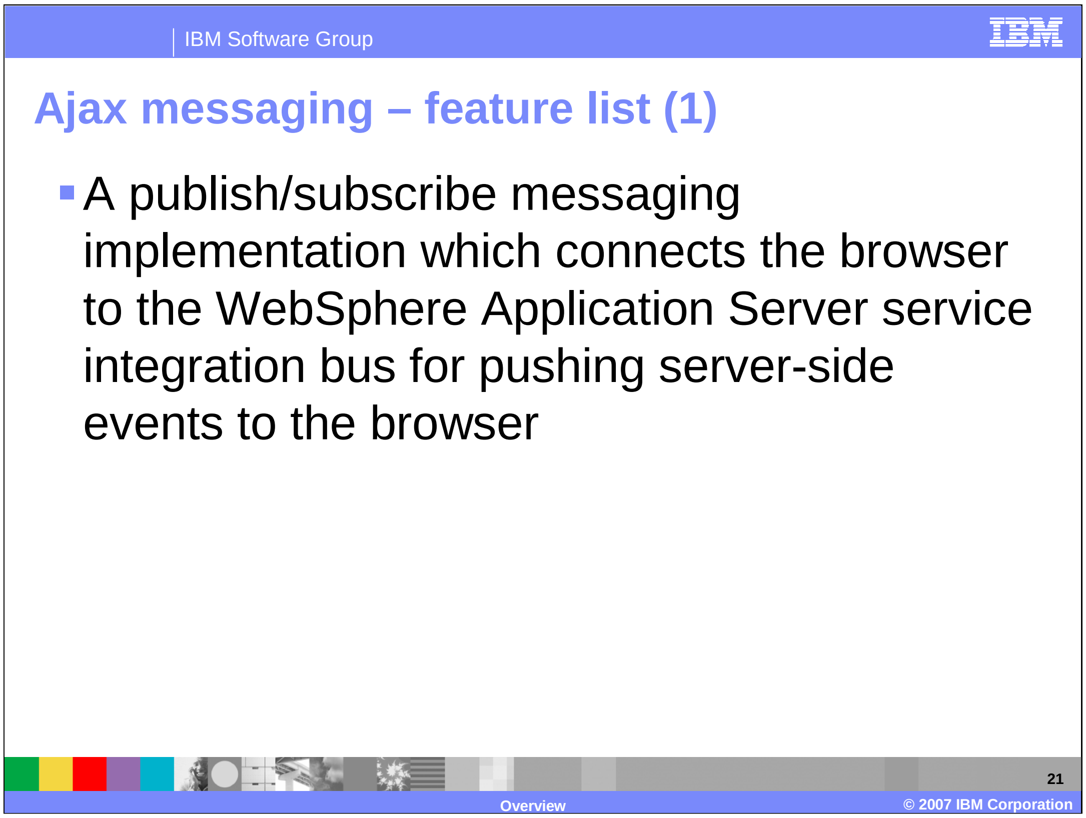


## Ajax messaging - overview

- For connecting Ajax clients to near-real-time updated data like stock quotes or instant messaging.
- Connects the browser to service integration bus topics for pushing server-side events to the browser
- Uses the Bayeux protocol for long-polling and forever frame Comet transport types

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

It also connects the browser to service integration bus topics for pushing server-side events to the browser. It uses the Bayeux protocol for long-polling and forever frame Comet transport types.



IBM Software Group

## Ajax messaging – feature list (1)

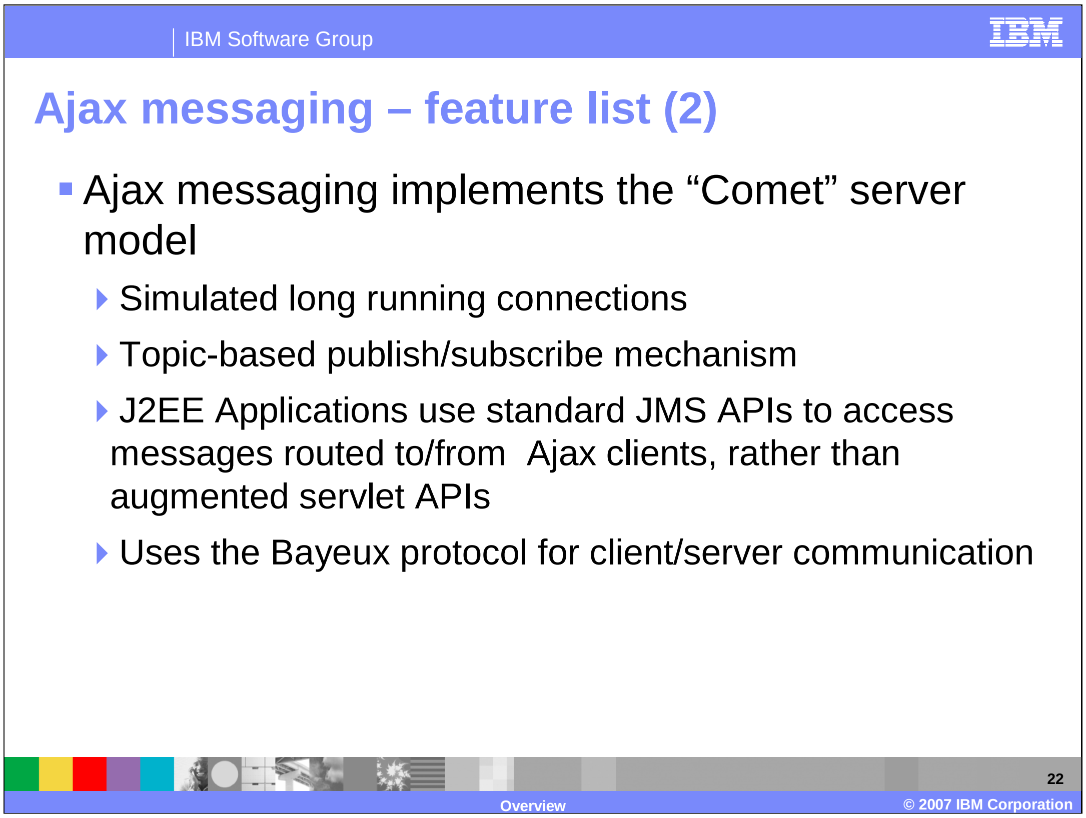
- A publish/subscribe messaging implementation which connects the browser to the WebSphere Application Server service integration bus for pushing server-side events to the browser

21

Overview

The Web messaging service bridges browser clients to the service integration bus allowing a Web service or any other item connected to the bus to publish events to Web-based clients.

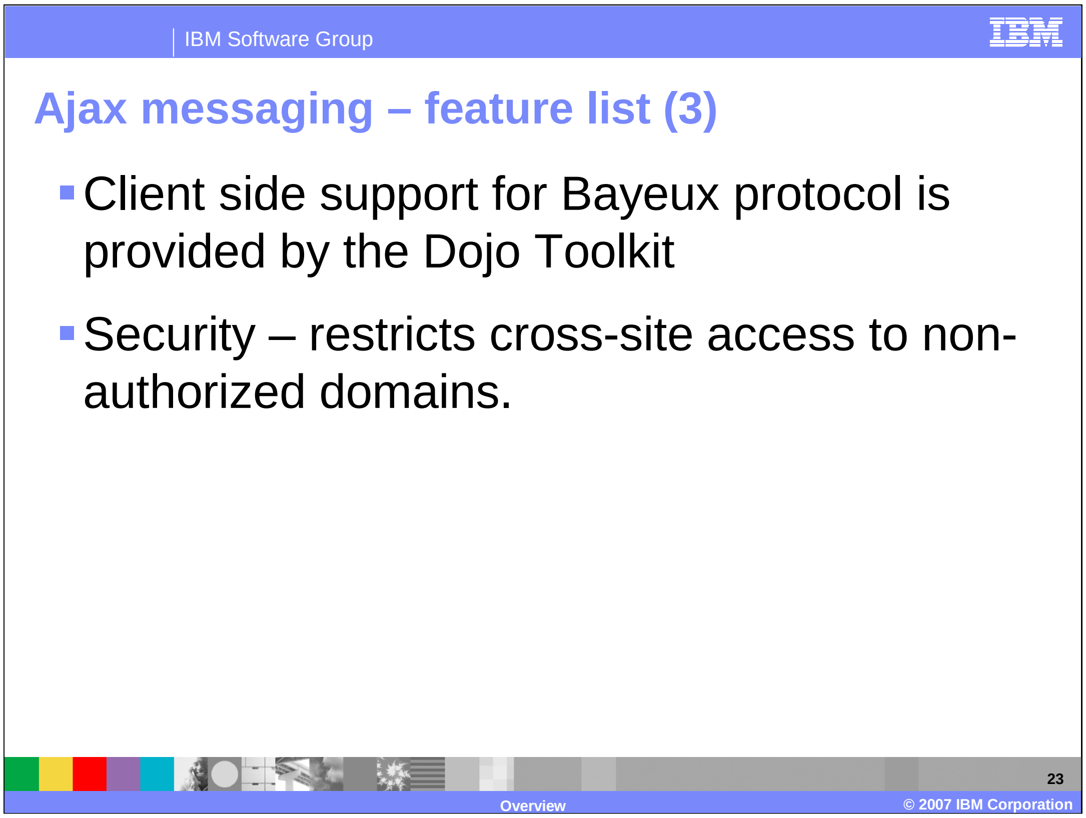
You can use the Web messaging Service in a new or existing application by placing a utility file library - a .jar file - in an application Web module, setting up a simple configuration file, and configuring servlet mappings. In addition, the service integration bus can be connected to WebSphere MQ through MQ link.  Through the service integration bus you should be able to bridge MQ to Web clients.



## Ajax messaging – feature list (2)

- Ajax messaging implements the “Comet” server model
  - Simulated long running connections
  - Topic-based publish/subscribe mechanism
  - J2EE 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

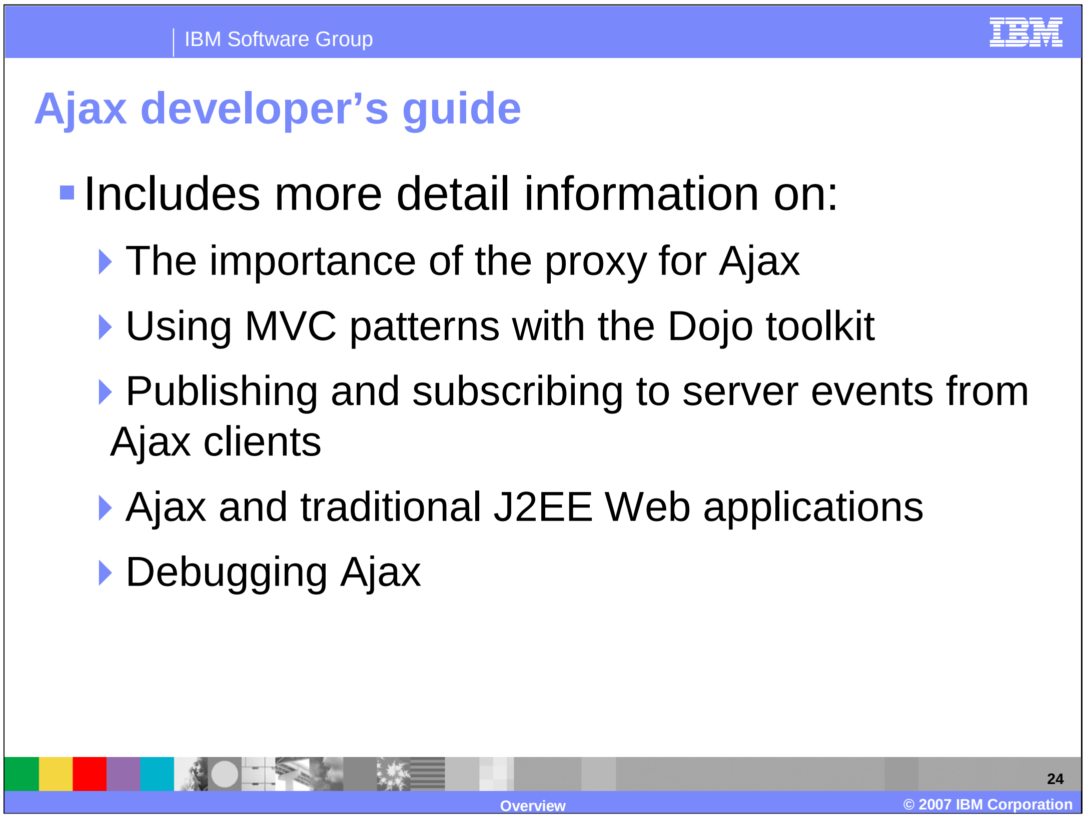
Web messaging implements the “Comet” server model. It has simulated long running connections and a topic-based publish/subscribe mechanism. J2EE Applications use standard JMS APIs to access messages routed to or from  Ajax clients, rather than an augmented servlet API. Ajax messaging uses the Bayeux protocol for client-server communication. Bayeux is a JSON-based protocol for clients to subscribe to events and for servers to deliver them more timely than possible with Ajax-based polling. The goals of messaging are to make event delivery fast and simple. Ajax messaging also allows for the protocol to be extended.



## Ajax messaging – feature list (3)

- Client side support for Bayeux protocol is provided by the Dojo Toolkit
- Security – restricts cross-site access to non-authorized domains.

For Web messaging, client side support for Bayeux protocol is provided by the Dojo Toolkit. Security restricts cross-site access to non-authorized domains.



IBM Software Group

## Ajax developer’s guide

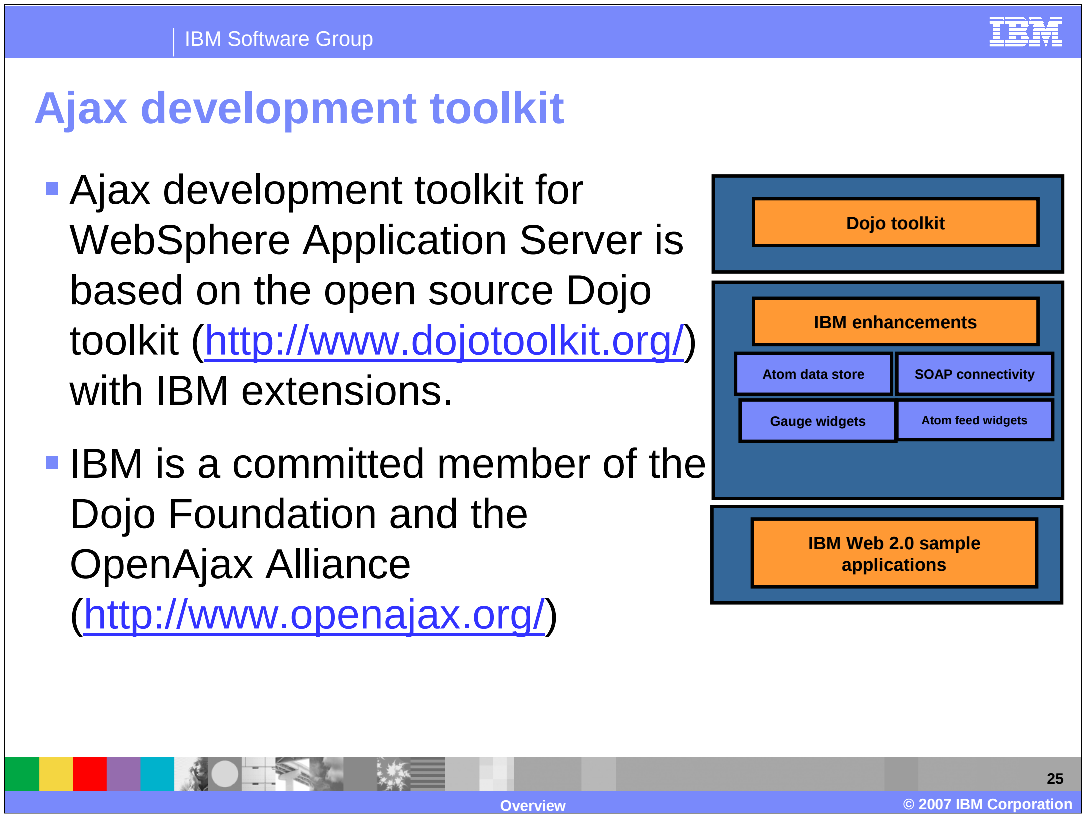
- Includes more detail information on:
  - The importance of the proxy for Ajax
  - Using MVC patterns with the Dojo toolkit
  - Publishing and subscribing to server events from Ajax clients
  - Ajax and traditional J2EE Web applications
  - Debugging Ajax

24

Overview

© 2007 IBM Corporation

The Ajax Developer’s Guide includes information on the importance of the proxy for Ajax, using MVC patterns with the Dojo toolkit, publishing and subscribing to server events from Ajax clients, Ajax and traditional J2EE Web applications, and debugging Ajax.

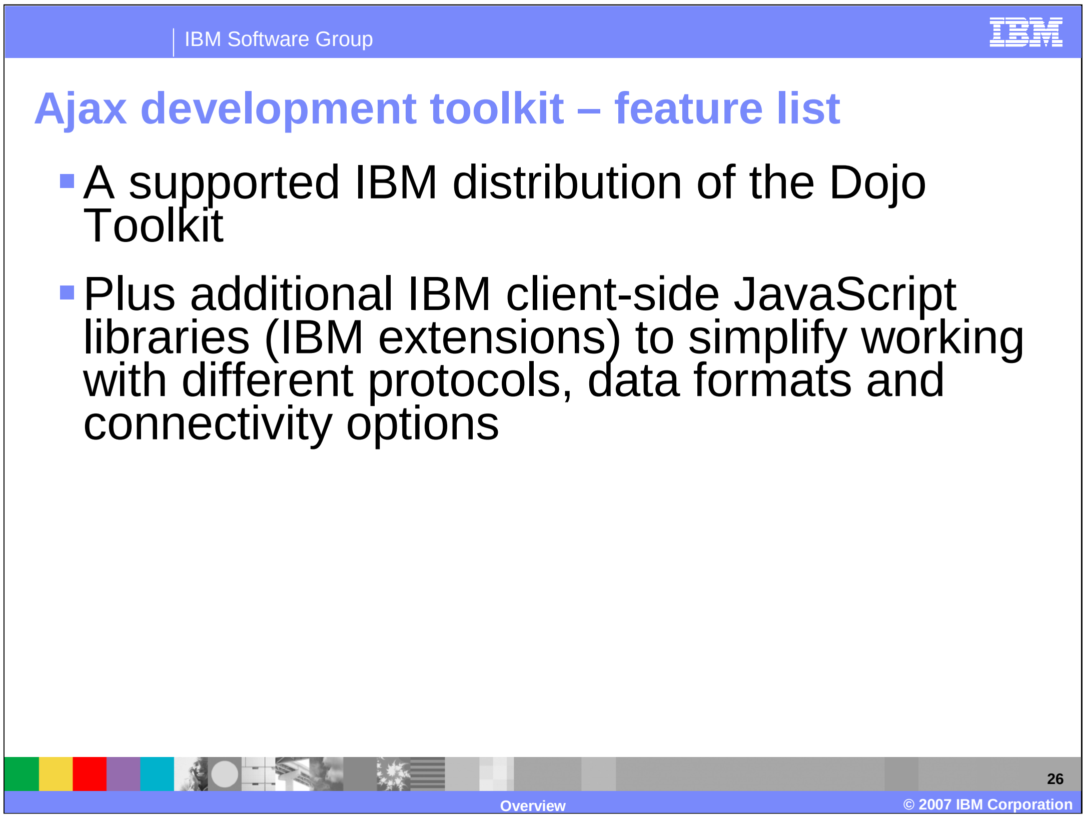


# Ajax development toolkit

- Ajax development toolkit for WebSphere Application Server is based on the open source Dojo toolkit (http://www.dojotoolkit.org/) with IBM extensions.
- IBM is a committed member of the Dojo Foundation and the OpenAjax Alliance (http://www.openajax.org/)

Dojo toolkit
IBM enhancements
Atom data store
SOAP connectivity
Gauge widgets
Atom feed widgets
IBM Web 2.0 sample applications

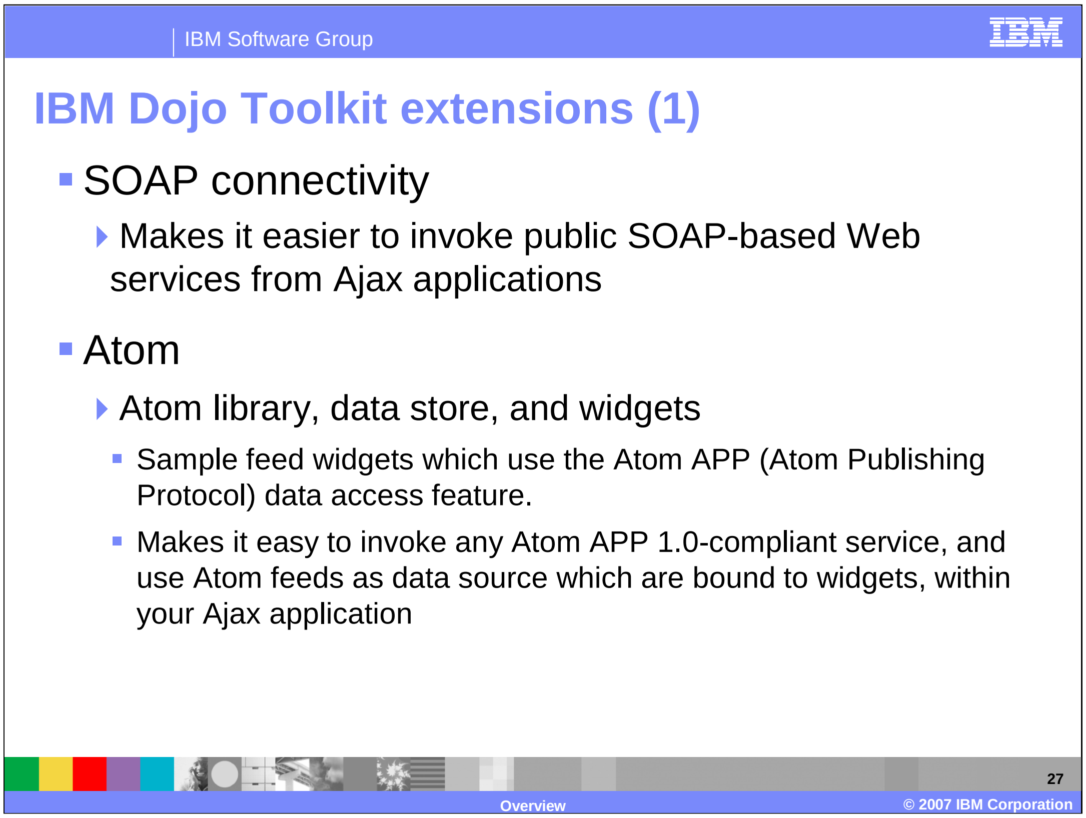
The Ajax development toolkit for WebSphere Application Server is based on the open source Dojo toolkit. IBM is a committed member of the Dojo Foundation and the OpenAjax Alliance.



## Ajax development toolkit – feature list

- A supported IBM distribution of the Dojo Toolkit
- Plus additional IBM client-side JavaScript libraries (IBM extensions) to simplify working with different protocols, data formats and connectivity options

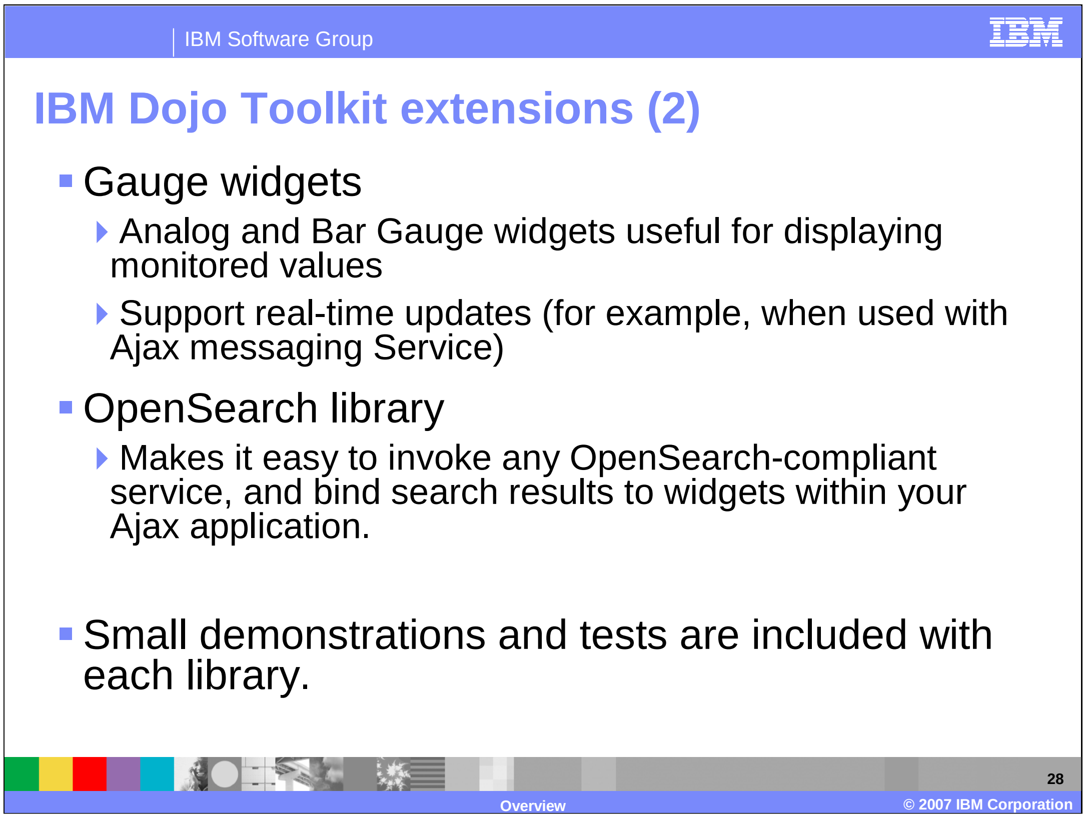
The Ajax development toolkit is a supported IBM distribution of the Dojo Toolkit plus additional IBM client-side JavaScript libraries (IBM extensions) to simplify working with different protocols, data formats and connectivity options



## IBM Dojo Toolkit extensions (1)

- SOAP connectivity
  - Makes it easier to invoke public SOAP-based Web services from Ajax applications
- Atom
  - Atom library, data store, and widgets
    - Sample feed widgets which use the Atom APP (Atom Publishing Protocol) data access feature.
    - Makes it easy to invoke any Atom APP 1.0-compliant service, and use Atom feeds as data source which are bound to widgets, within your Ajax application

The IBM Dojo Toolkit extensions include Atom, SOAP connectivity, Gauge widgets, and the OpenSearch Library. SOAP connectivity makes it easier to invoke public SOAP-based Web services from Ajax applications. The Atom library, data store, and widgets are included in the Dojo toolkit extensions.  The Atom publishing protocol (APP) data access makes it easy to invoke any Atom APP 1.0-compliant service, and uses Atom feeds as data sources which are bound to widgets, within your Ajax application. Atom feed widgets are sample feed widgets which use the Atom APP data access feature.



IBM Software Group

## IBM Dojo Toolkit extensions (2)

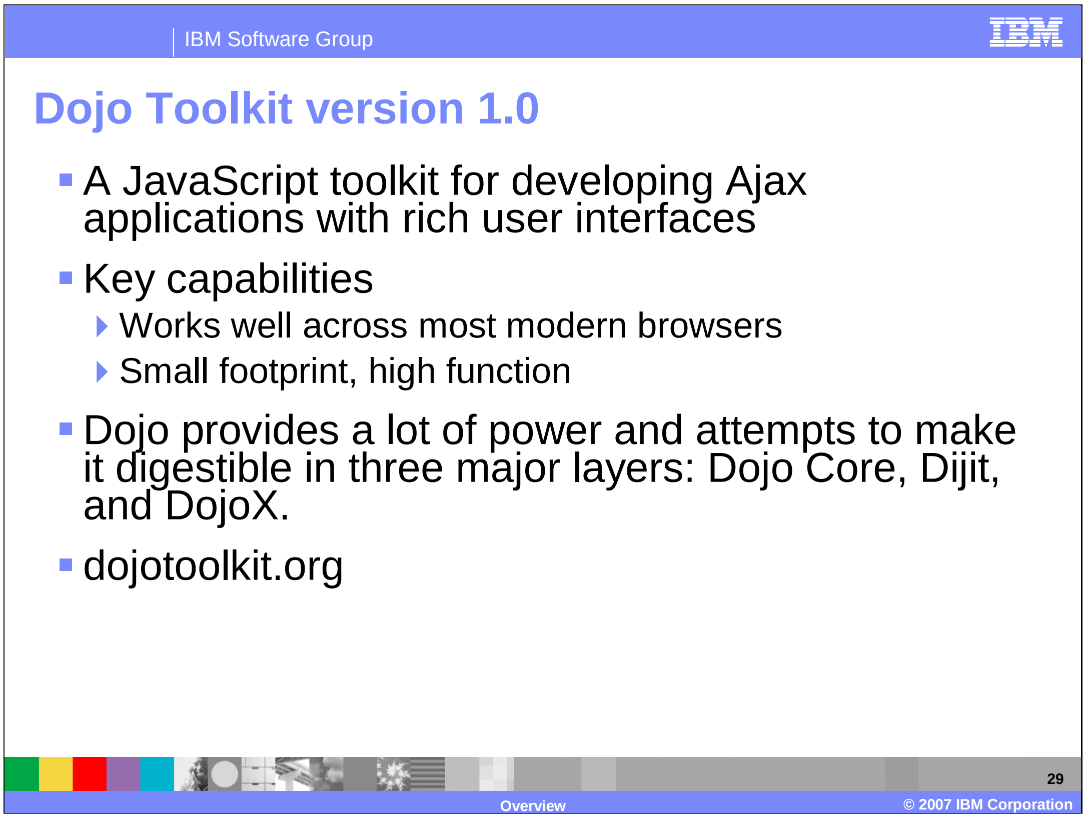
- Gauge widgets
  - Analog and Bar Gauge widgets useful for displaying monitored values
  - Support real-time updates (for example, when used with Ajax messaging Service)
- OpenSearch library
  - Makes it easy to invoke any OpenSearch-compliant service, and bind search results to widgets within your Ajax application.
- Small demonstrations and tests are included with each library.

28

Overview

© 2007 IBM Corporation

Analog and Bar Gauge widgets are useful for displaying monitored values and they support real-time updates. The OpenSearch Library makes it easy to invoke any OpenSearch-compliant service and bind search results to widgets within your Ajax application. Small demonstrations and tests are included with each library.



IBM Software Group

## Dojo Toolkit version 1.0

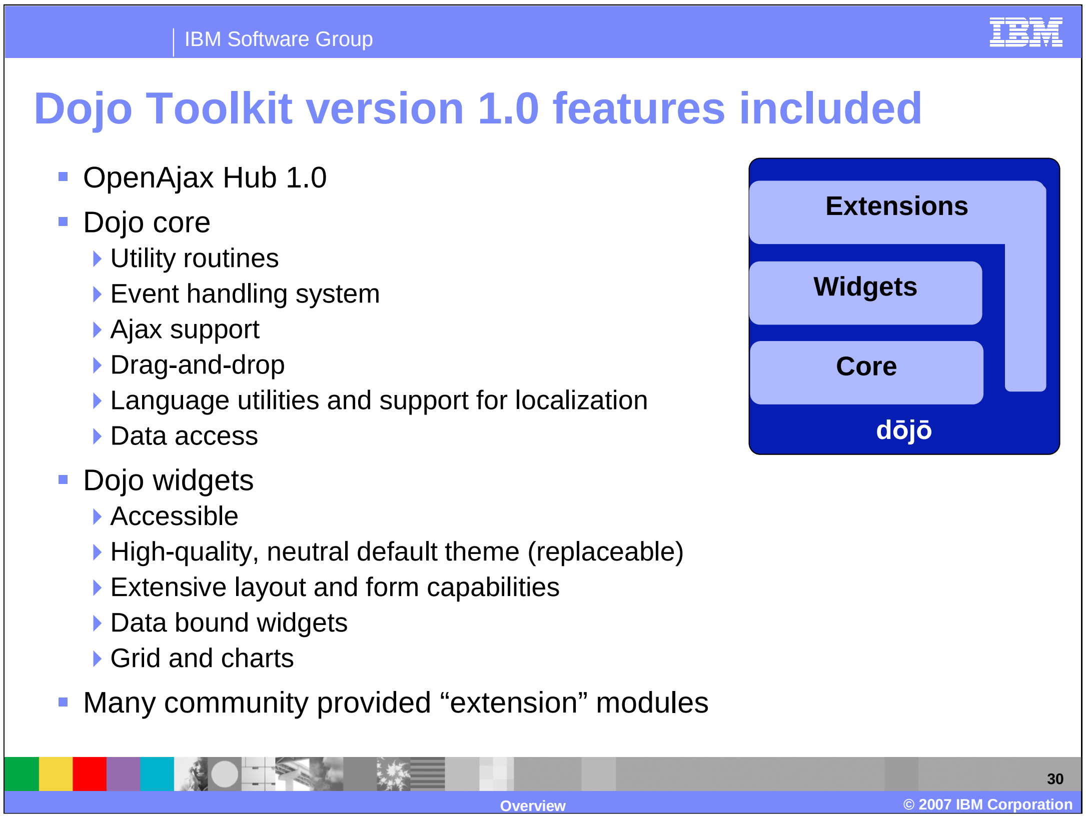
- A JavaScript toolkit for developing Ajax applications with rich user interfaces
- Key capabilities
  - Works well across most modern browsers
  - Small footprint, high function
- Dojo provides a lot of power and attempts to make it digestible in three major layers: Dojo Core, Dijit, and DojoX.
- dojotoolkit.org

29

Overview

© 2007 IBM Corporation

Dojo is an open source DHTML toolkit written in the JavaScript language. Dojo allows you to build dynamic capabilities into Web pages.  You can use the components that Dojo provides to make your Web sites more usable, responsive, and functional. Dojo builds on several contributing code bases. Dojo toolkit works well across most modern browsers and provides a lot of power and attempts to make it digestible in three major layers: Dojo Core, Dijit, and DojoX.



## Dojo Toolkit version 1.0 features included

- OpenAjax Hub 1.0
- Dojo core
  - Utility routines
  - Event handling system
  - Ajax support
  - Drag-and-drop
  - Language utilities and support for localization
  - Data access
- Dojo widgets
  - Accessible
  - High-quality, neutral default theme (replaceable)
  - Extensive layout and form capabilities
  - Data bound widgets
  - Grid and charts
- Many community provided “extension” modules

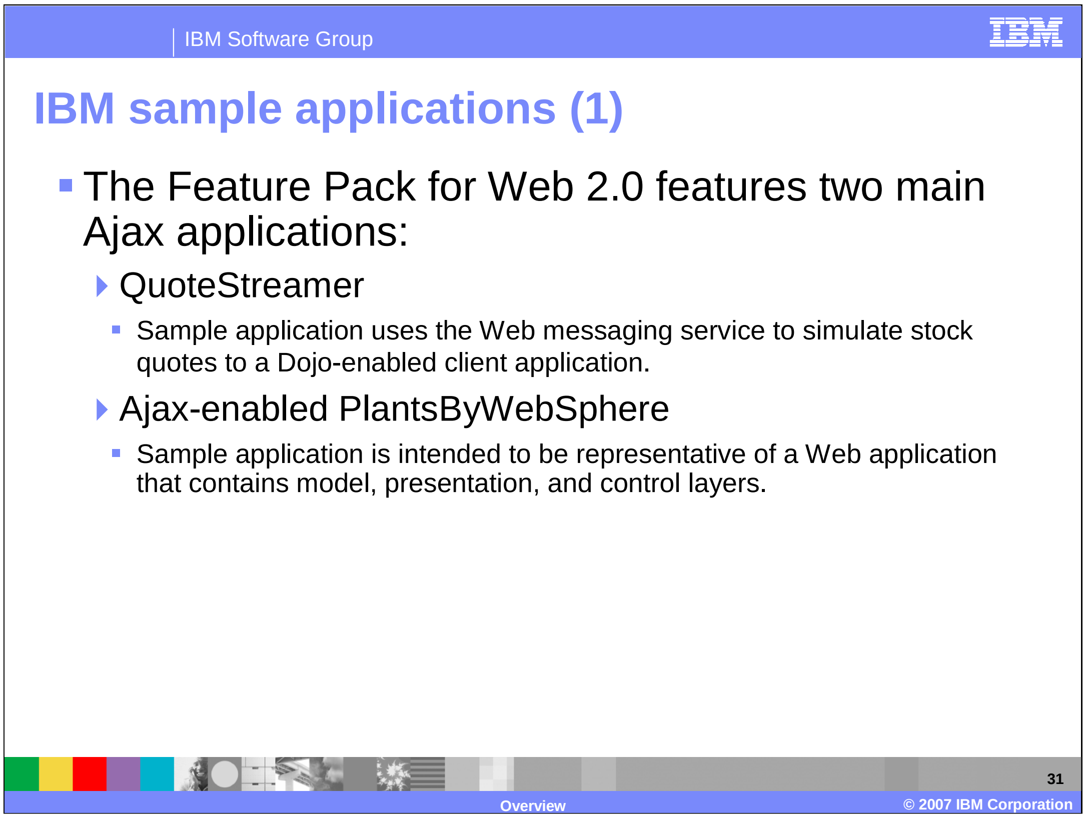
Extensions

Widgets

Core

dōjō

The features included in the Dojo Toolkit version 1.0 include Dojo Core and Dojo widgets. There are also many community-provided extension modules.



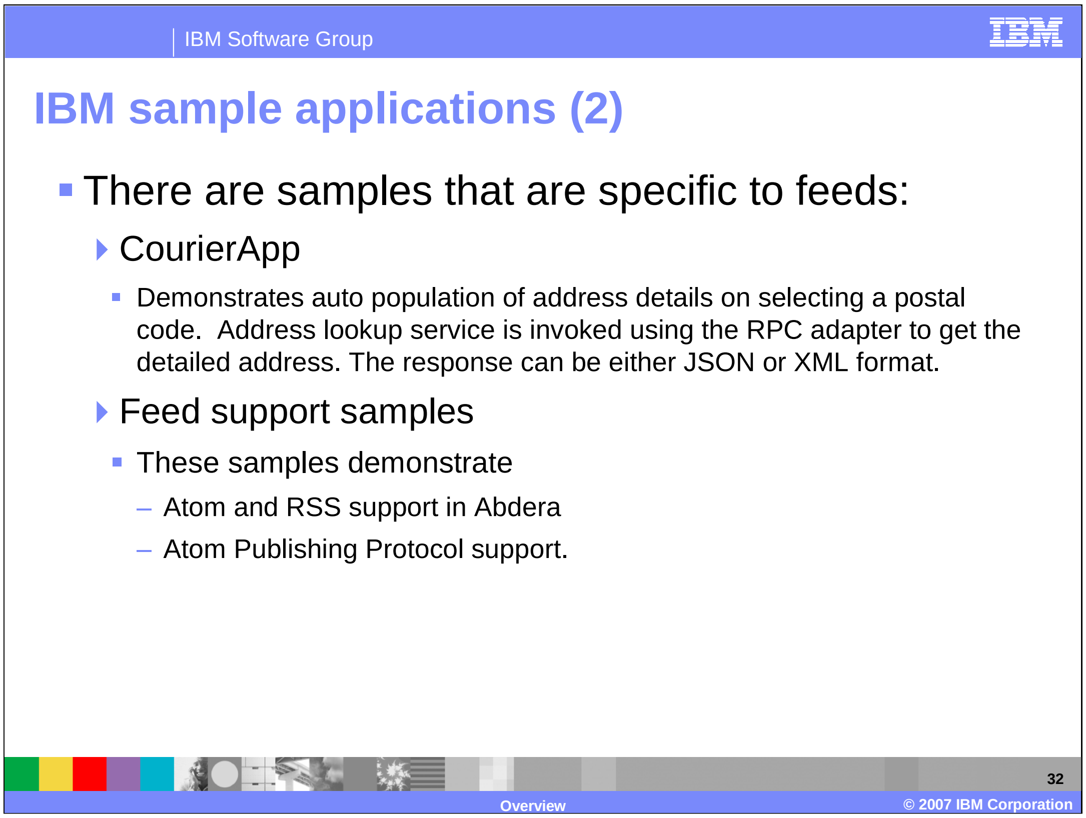
## IBM sample applications (1)

- The Feature Pack for Web 2.0 features two main Ajax applications:
  - QuoteStreamer
    - Sample application uses the Web messaging service to simulate stock quotes to a Dojo-enabled client application.
  - Ajax-enabled PlantsByWebSphere
    - Sample application is intended to be representative of a Web application that contains model, presentation, and control layers.

The Feature Pack for Web 2.0 features two main Ajax applications.

The QuoteStreamer sample application uses the Web messaging service to simulate stock quotes to a Dojo-enabled client application.

The Ajax-enabled PlantsByWebSphere application is intended to be representative of a Web application that contains model, presentation, and control layers.



## IBM sample applications (2)

- There are samples that are specific to feeds:
  - CourierApp
    - Demonstrates auto population of address details on selecting a postal code.  Address lookup service is invoked using the RPC adapter to get the detailed address. The response can be either JSON or XML format.
  - Feed support samples
    - These samples demonstrate
      - Atom and RSS support in Abdera
      - Atom Publishing Protocol support.

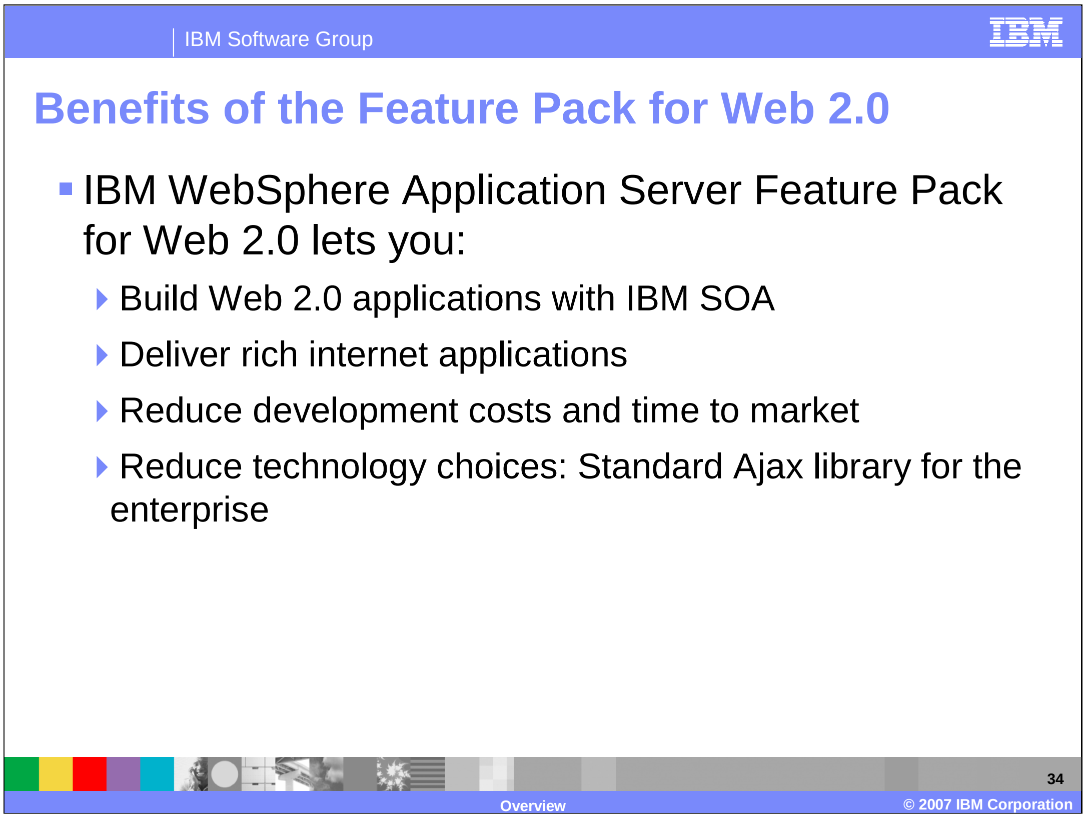
The Feature Pack for Web 2.0 contains code snippets and small test samples and four larger sample applications.

There are two samples that are specific to feeds: The CourierApp Sample Web application demonstrates auto population of address details on selecting a postal code.  The address lookup service is invoked using the RPC adapter to get the detailed address. The response can be either JSON or XML format. The feed samples demonstrate the Atom and RSS support in Abdera and the Atom Publishing Protocol support.

## Section

***Benefits of IBM WebSphere Application Server Feature Pack for Web 2.0***

This section addresses the benefits of the feature pack for Web 2.0.



IBM Software Group

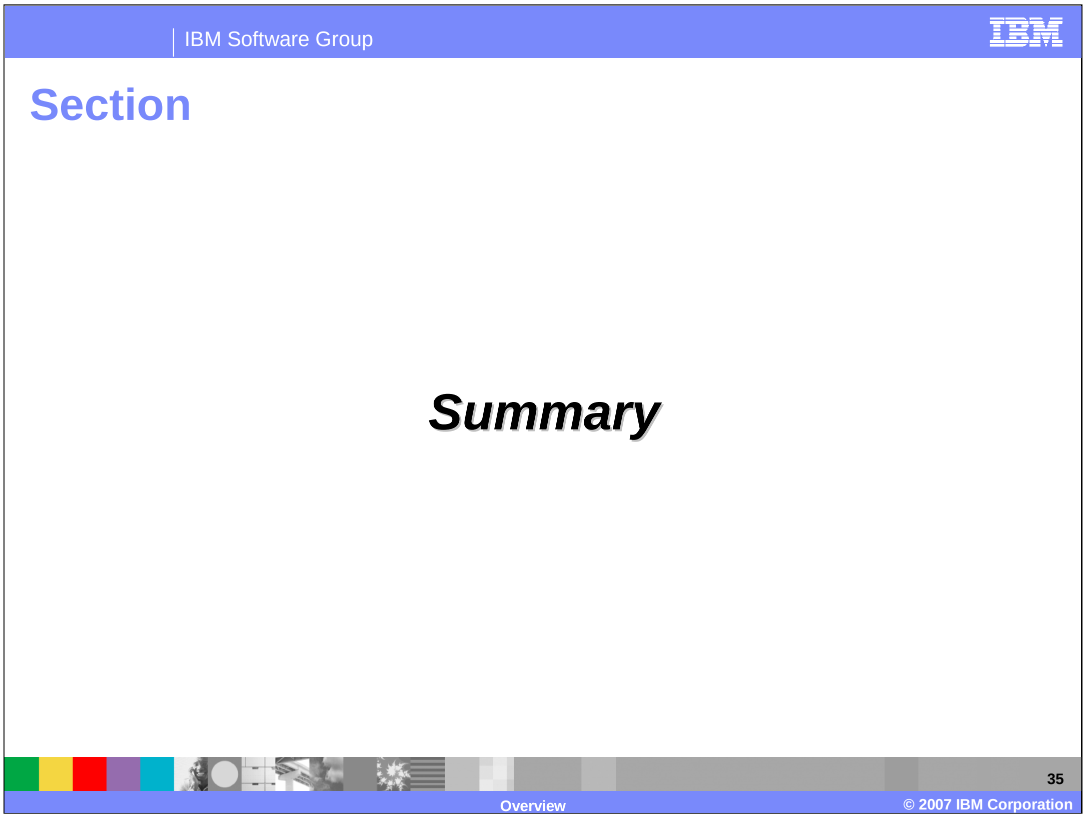
## Benefits of the Feature Pack for Web 2.0

- IBM WebSphere Application Server Feature Pack for Web 2.0 lets you:
  - Build Web 2.0 applications with IBM SOA
  - Deliver rich internet applications
  - Reduce development costs and time to market
  - Reduce technology choices: Standard Ajax library for the enterprise

34

Overview

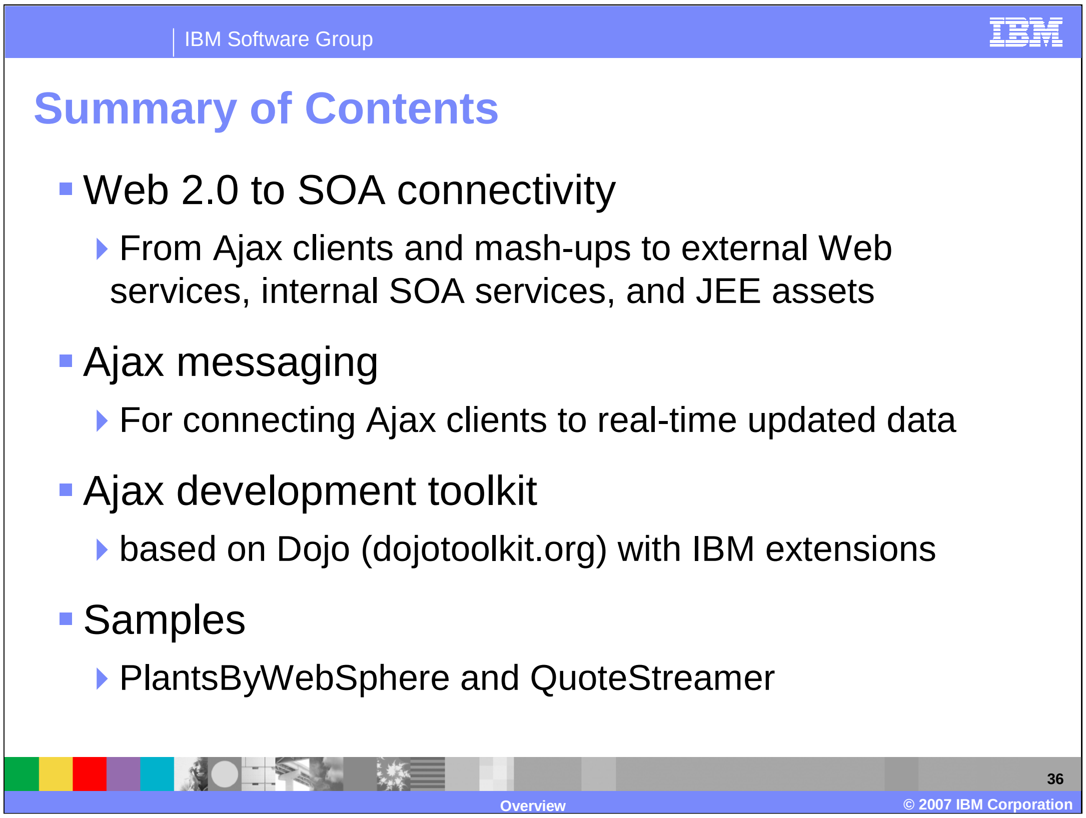
The feature pack lets you build and deliver rich Web 2.0 internet applications with IBM SOA, reduce development costs and time to market, and reduce technology choices by providing a standard Ajax library for the enterprise.



## Section

***Summary***

This section is the summary.



IBM Software Group

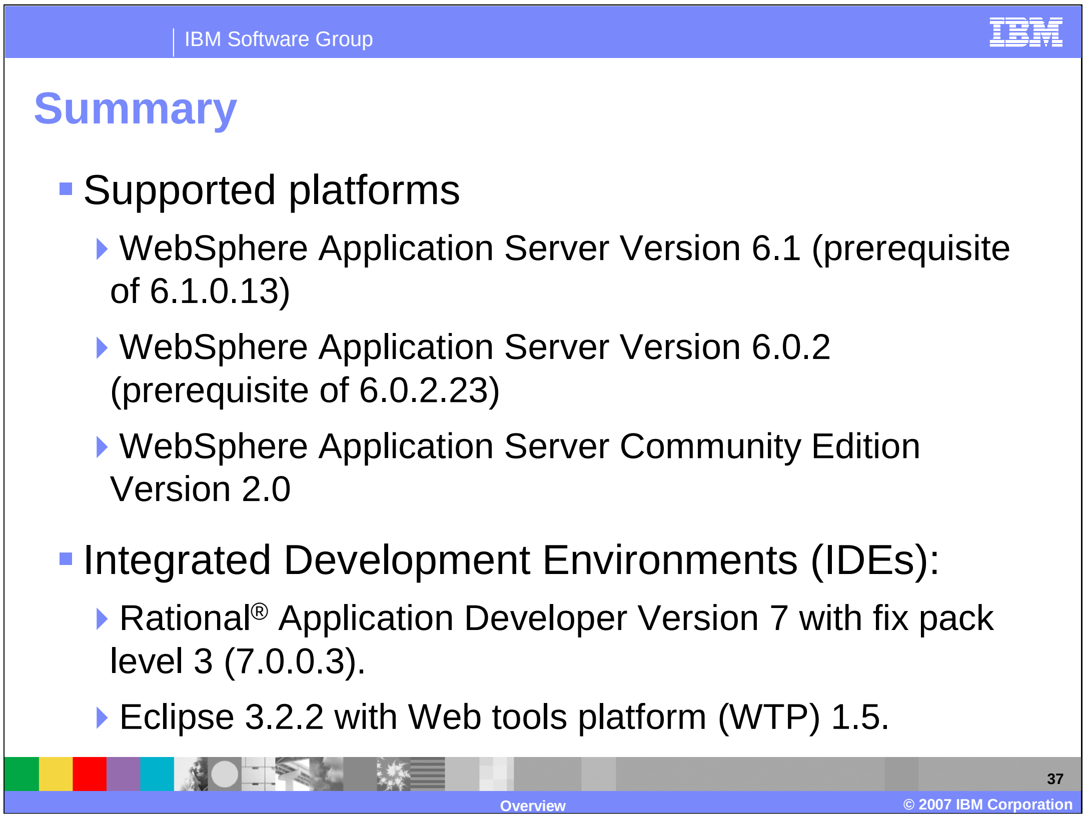
## Summary of Contents

- Web 2.0 to SOA connectivity
  - From Ajax clients and mash-ups to external Web services, internal SOA services, and JEE assets
- Ajax messaging
  - For connecting Ajax clients to real-time updated data
- Ajax development toolkit
  - based on Dojo (dojotoolkit.org) with IBM extensions
- Samples
  - PlantsByWebSphere and QuoteStreamer

36

Overview

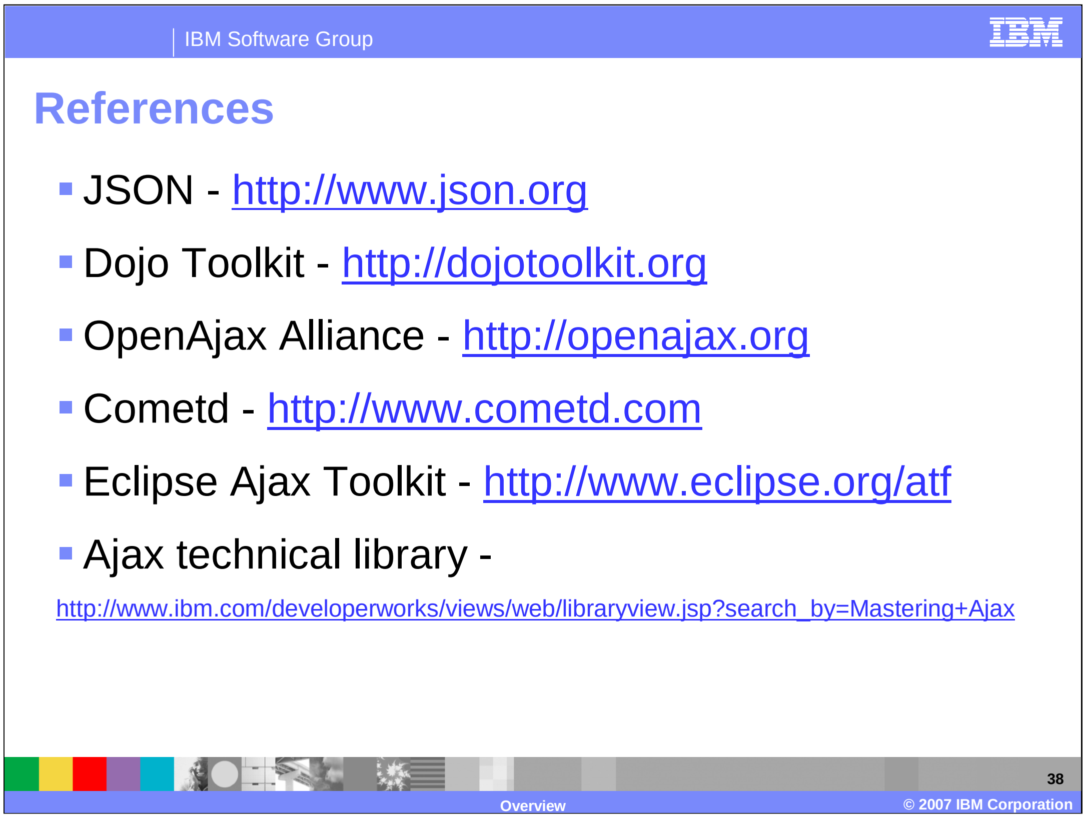
Web 2.0 to SOA connectivity enables connectivity from Ajax clients and mash-ups to external Web services, internal SOA services, and JEE assets. Ajax messaging connects Ajax clients to real-time updated data like stock quotes or instant messaging. The Ajax development toolkit for WebSphere Application Server is based on Dojo with IBM extensions, and the samples include Ajax-based versions of PlantsByWebSphere and QuoteStreamer.



## Summary

- Supported platforms
  - WebSphere Application Server Version 6.1 (prerequisite of 6.1.0.13)
  - WebSphere Application Server Version 6.0.2 (prerequisite of 6.0.2.23)
  - WebSphere Application Server Community Edition Version 2.0
- Integrated Development Environments (IDEs):
  - Rational® Application Developer Version 7 with fix pack level 3 (7.0.0.3).
  - Eclipse 3.2.2 with Web tools platform (WTP) 1.5.

The supported platforms and development environments are listed here for your reference.

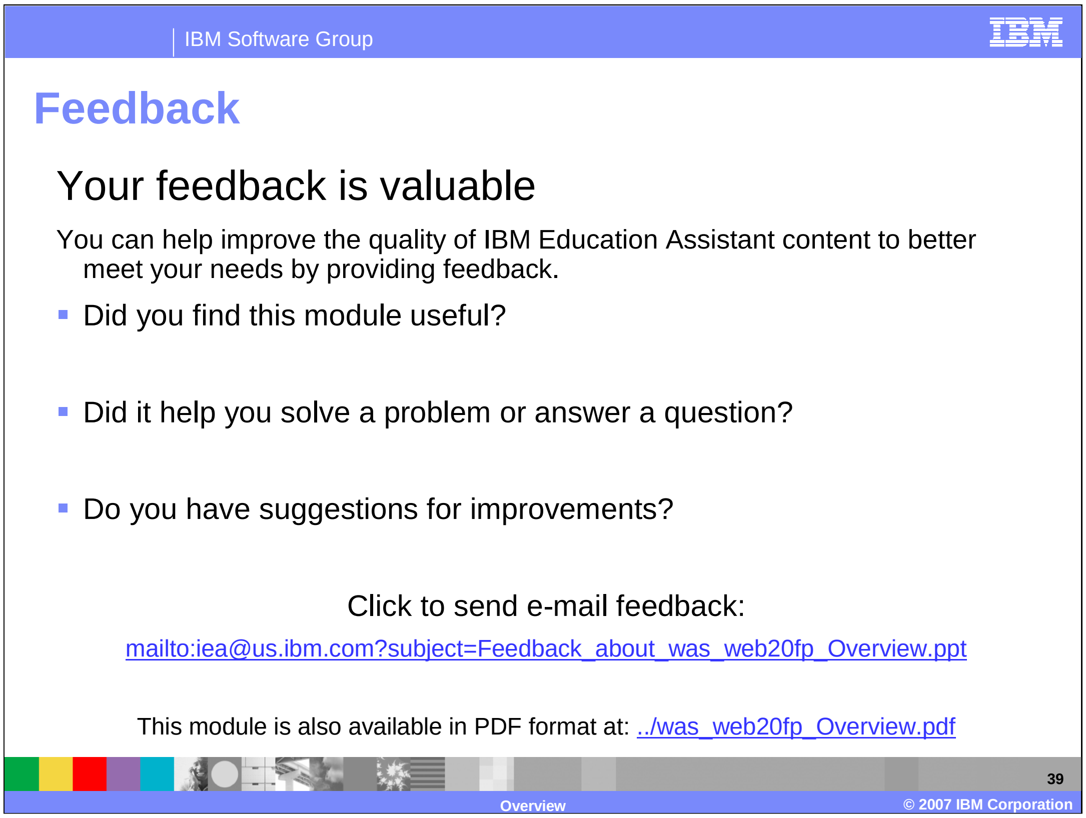


## References

- JSON - http://www.json.org
- Dojo Toolkit - http://dojotoolkit.org
- OpenAjax Alliance - http://openajax.org
- Cometd - http://www.cometd.com
- Eclipse Ajax Toolkit - http://www.eclipse.org/atf
- Ajax technical library -

http://www.ibm.com/developerworks/views/web/libraryview.jsp?search_by=Mastering+Ajax

These references are provided for more information on each of these topics.



## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

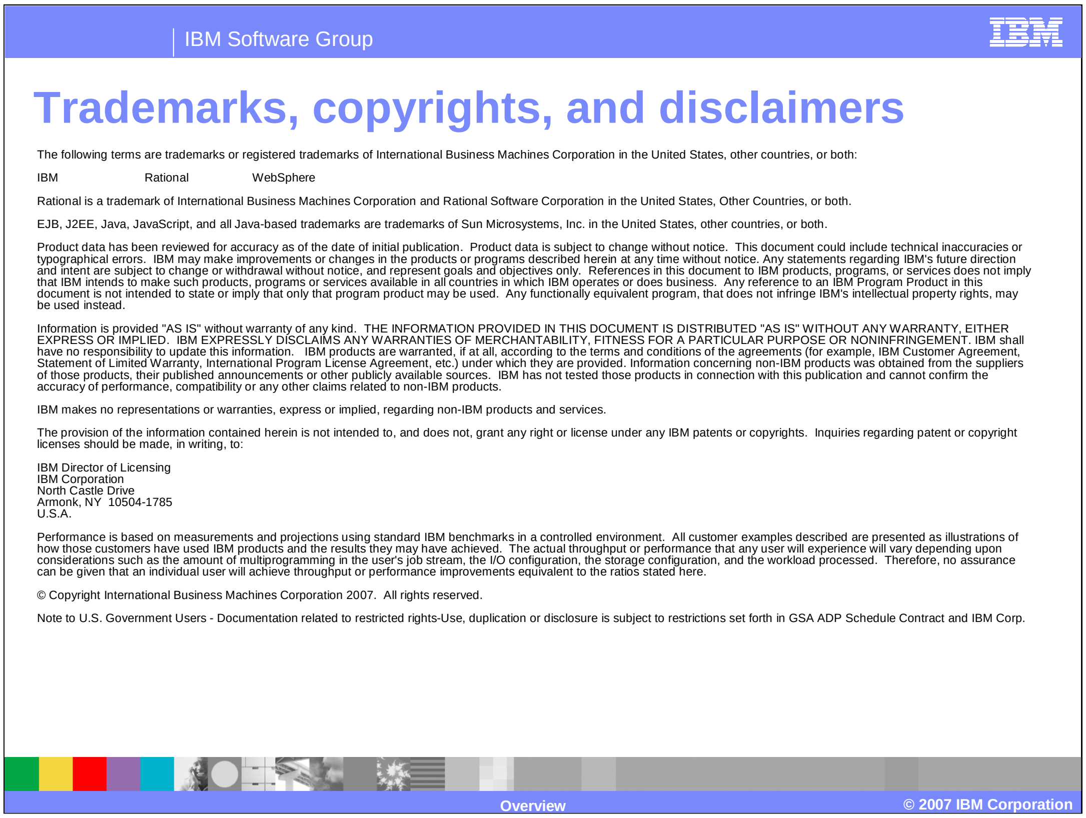
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_was_web20fp_Overview.ppt

This module is also available in PDF format at: ../was_web20fp_Overview.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



# Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM Rational WebSphere

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

EJB, J2EE, Java, JavaScript, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.