



HTTP Pub/Sub Server[®] BEA WebLogic Server[®] 10.3

WebLogic Server Product
Management
March 2008

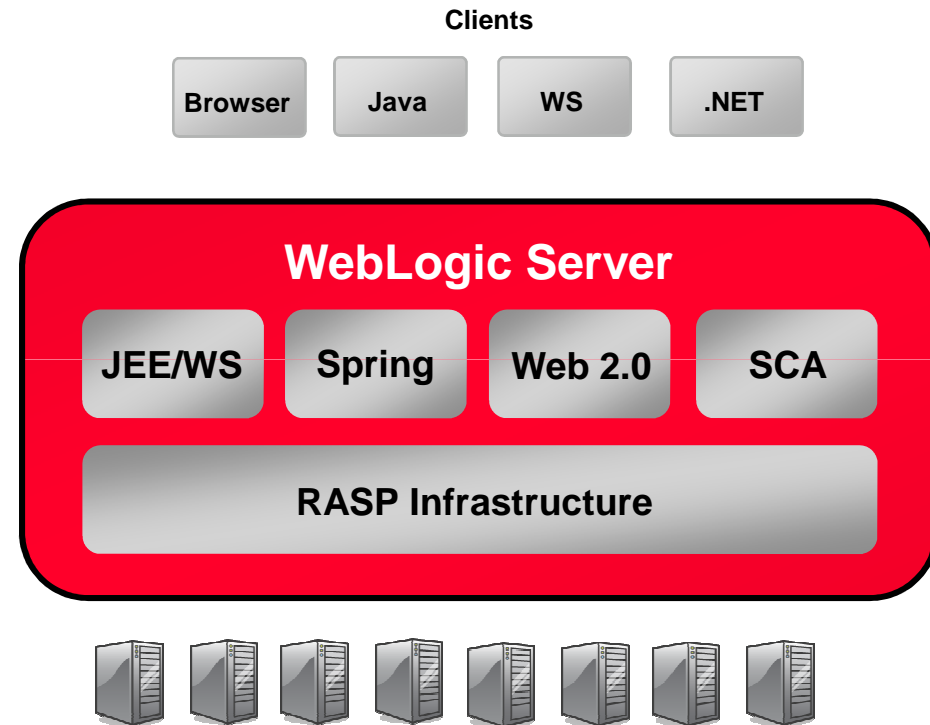
BEA WebLogic Server® 10.3

- Technology Preview Available

- ▶ Download from www.bea.com

- Themes

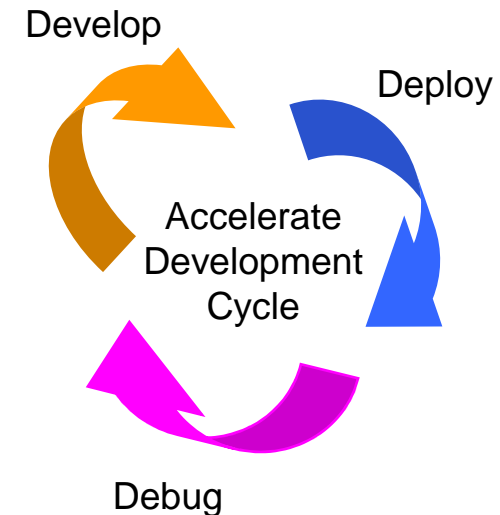
- ▶ Lightweight
- ▶ Web Services/SOA Server
- ▶ Web 2.0
- ▶ Open Source Support
- ▶ Performance
- ▶ Standards updates, technology integration



The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.

Theme: Lightweight WebLogic Server

- Improved flexibility, agility and usability
 - ▶ Accelerate iterative development cycle
- Lightweight installers
 - ▶ 150 MB for “core” WLS, add-ons (JDKs, etc) optional
 - ▶ Reduce download and install time, installation footprint
- Hot swap of classes
 - ▶ Modify classes without requiring redeployment
 - ▶ Accelerate iterative development
- Selective service startup
 - ▶ Start WebLogic Express without starting EJB, JMS services
 - ▶ Reduce resource consumption
- Improved console performance
 - ▶ Console startup and navigation – 50%+ improvements over WLS 9.2 and 10.0
 - ▶ Accelerate iterative development and console usability



Lightweight Installation

Leveraging mSA componentization for optimal footprint



JDKs are selectable, too

The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.

Runtime - Optional startup of services

- *To decrease memory footprint & startup time*
- Run server as “WLS” or “WLX”
- WLS – starting up everything
- WLX – will not startup EJB, JMS, Connector

FastSwap

- What is it?

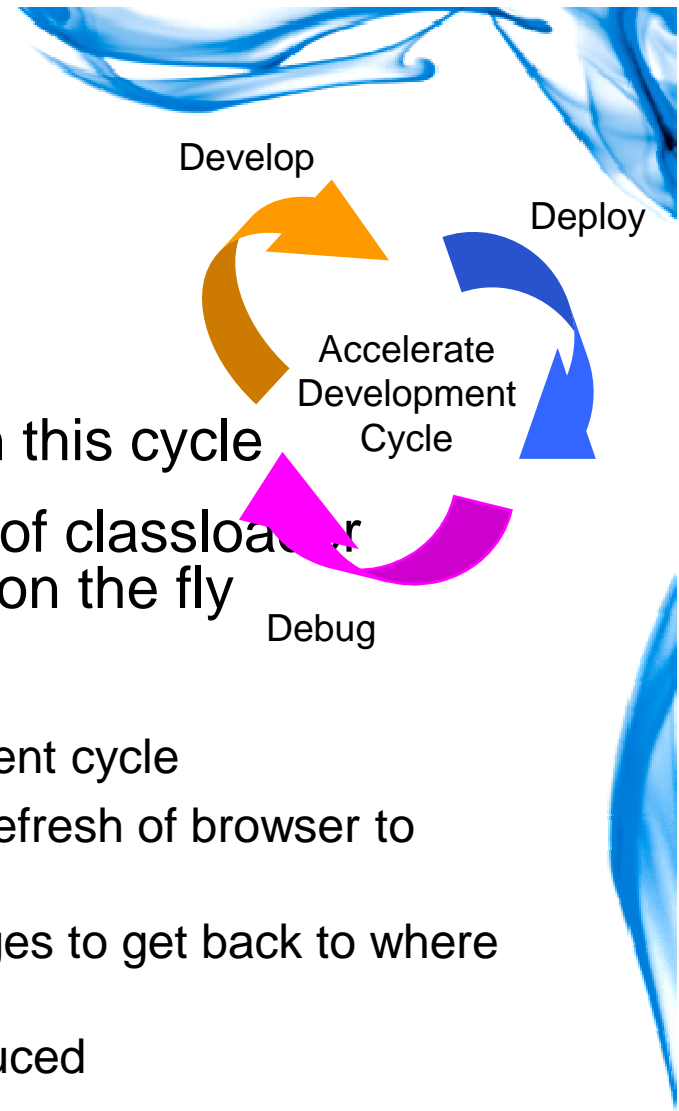
- ▶ A class redefinition technology that builds on JDK features

- What is it good for?

- ▶ Dramatically reduces iterative development time
 - Improves developer productivity
- ▶ Changes the “Edit > Save > Compile > Redeploy >Recreate State” cycle to “Edit > Save > Recompile > Refresh Browser”
- ▶ See your changes almost immediately
- ▶ No need to bounce the classloader
- ▶ *No loss of state*

FastSwap: Class Redefinition

- Typical iterative development cycle:
 - ▶ *Edit* → *Build* → *Deploy* → *Test*
- Our goal: eliminate the deployment step in this cycle
- Class redefinition facility takes advantage of classloader improvements to pickup changed classes on the fly
- The Result:
 - ▶ Eliminates the redeploy step from the development cycle
 - ▶ Allows IDE edit with auto-compile on save and refresh of browser to demonstrate change
 - ▶ State is maintained – no need to re-traverse pages to get back to where you were just to see your changes
 - ▶ Iterative development cycle is **dramatically** reduced
- Includes support for most changes, including adding, removing, changing methods
- Enabled per application; dev mode only



FastSwap – Supported Change Types

- Add, change, remove static methods (bodies, signatures)
- Add, change, remove instance methods (bodies, signatures)
- Add, change, remove static fields
- Add, change, remove instance fields
- Add, change, remove constructors
- Add, change, remove interface methods (internal interfaces only)
- Add, change, remove static, local and anonymous inner classes

FastSwap Limitations

The following are not supported:

- Java Reflection
- Changing the hierarchy of an already existing class
 - ▶ Cannot change the list of implemented interfaces of a class
 - ▶ Cannot change the superclass of a class
- Addition or Removal of Java Annotations
- Addition or Removal of methods on EJB Interfaces
- Addition or Removal of constants from Enums
- Addition or Removal of the finalize method

More: <http://e-docs.bea.com/wls/essex/TechPreview/pdf/FastSwap.pdf>

The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.

Theme: Open Source Support

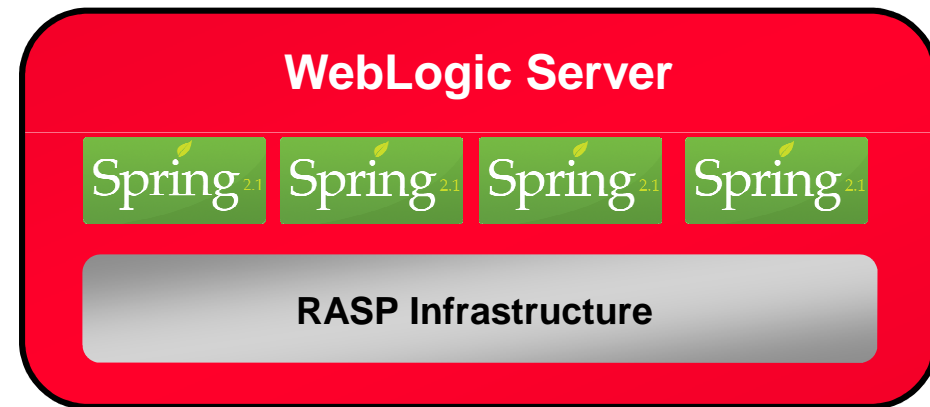
- Blended development and integration

- ▶ Spring
- ▶ Eclipse
- ▶ OpenJPA
- ▶ Glassfish JAX-WS RI
- ▶ Dojo
- ▶

- Spring enhancements

- ▶ Build on existing support
- ▶ Spring 2.5 certification
- ▶ Improved administration of Spring apps
- ▶ Spring Security/WLS security integration
- ▶ WebLogic Server as deployment container of choice for Spring apps

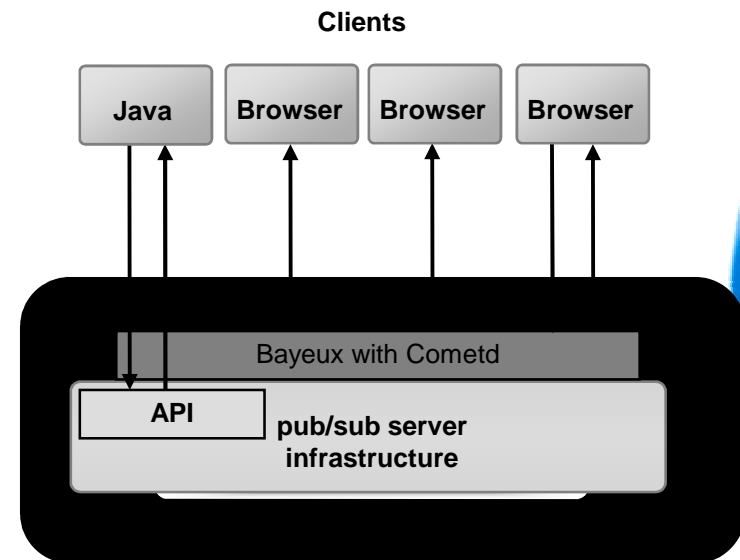
Admin and
Monitoring



The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.

Theme: Web 2.0

- Enable next generation of Web Apps
 - ▶ Rich UI for client applications
 - ▶ Collaborative apps and social computing
 - ▶ Ajax/Dojo client support
- Http pub/sub engine for collaborative apps
 - ▶ Enable data exchange within apps over persistent connections
 - ▶ Dojo client and scalable server-side side engine



Why HTTP Pub/Sub Server

● Traditional Request/Response applications

- ▶ Not dynamic – Server can't push data to clients without receiving requests
- ▶ Traditional polling for data at regular intervals can lead to high network traffic and lack of scalability

● Pub/Sub based applications

- ▶ Deliver dynamic real-time data from server
- ▶ Enables new class of applications (Stock quotes, chat rooms, auction updates)
- ▶ Develop dynamic real-time web apps with scalability

What is HTTP Pub/Sub Server?

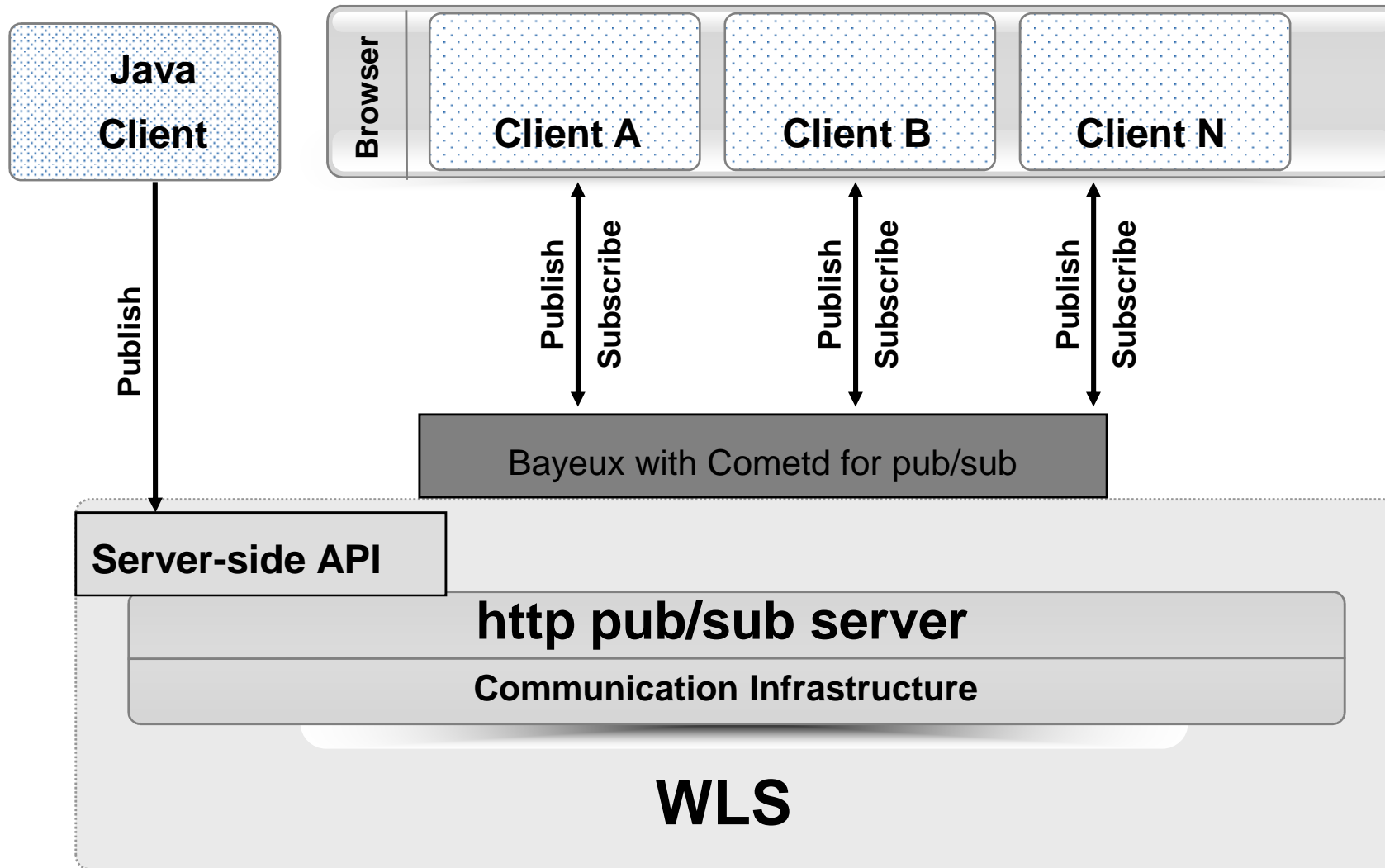
- Channels based publish/subscribe mechanism for web based clients to send and receive asynchronous messages over HTTP
- You can have multiple clients connect to a server, subscribe to channels and publish or listen for messages on these channels
- Clients connected to one server can listen to events from other servers and talk to clients connected to other servers – within a cluster.
- Delivers enterprise support features
 - ▶ Security (Authorization and Authentication, role based)
 - ▶ Message persistence
 - ▶ Cluster support for fail-over with load balancing
 - ▶ Console Management



HTTP Pub/Sub Server Drilldown

- The Pub/Sub Server is based on the Bayeux protocol proposed by the Cometd project and works with Dojo toolkit on the client side
- HTTP Pub/Sub server supports Bayeux protocol 1.0 draft1 and all released versions of Dojo including 0.43/0.9/1.0.
- Two types of message handlers
 - ▶ Local Memory (default)
 - Limited to a single managed server for all the subscribed client channels (no clustering capabilities with single point of failure)
 - ▶ WLS JMS
 - Leverages the power of WLS JMS messaging architecture on the server side (e.g. distributed destinations, JMS cluster, indexed subscribers etc.) for fast, efficient and reliable communication.
- Message persistence is configurable (stores messages on the disk)

WLS http pub/sub server



The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.

Demo(Stock Portfolio)

- A Web 2.0 application of stock portfolio based on HTTP Pub/Sub Server and Dojo Toolkit
- Multiple clients for generating simulated stock trade information and publishing the information to Pub/Sub Server
- Display trade information of stocks in portfolio as table and chart; maintenance of portfolio

Demo(Stock Portfolio)

● Use Cases

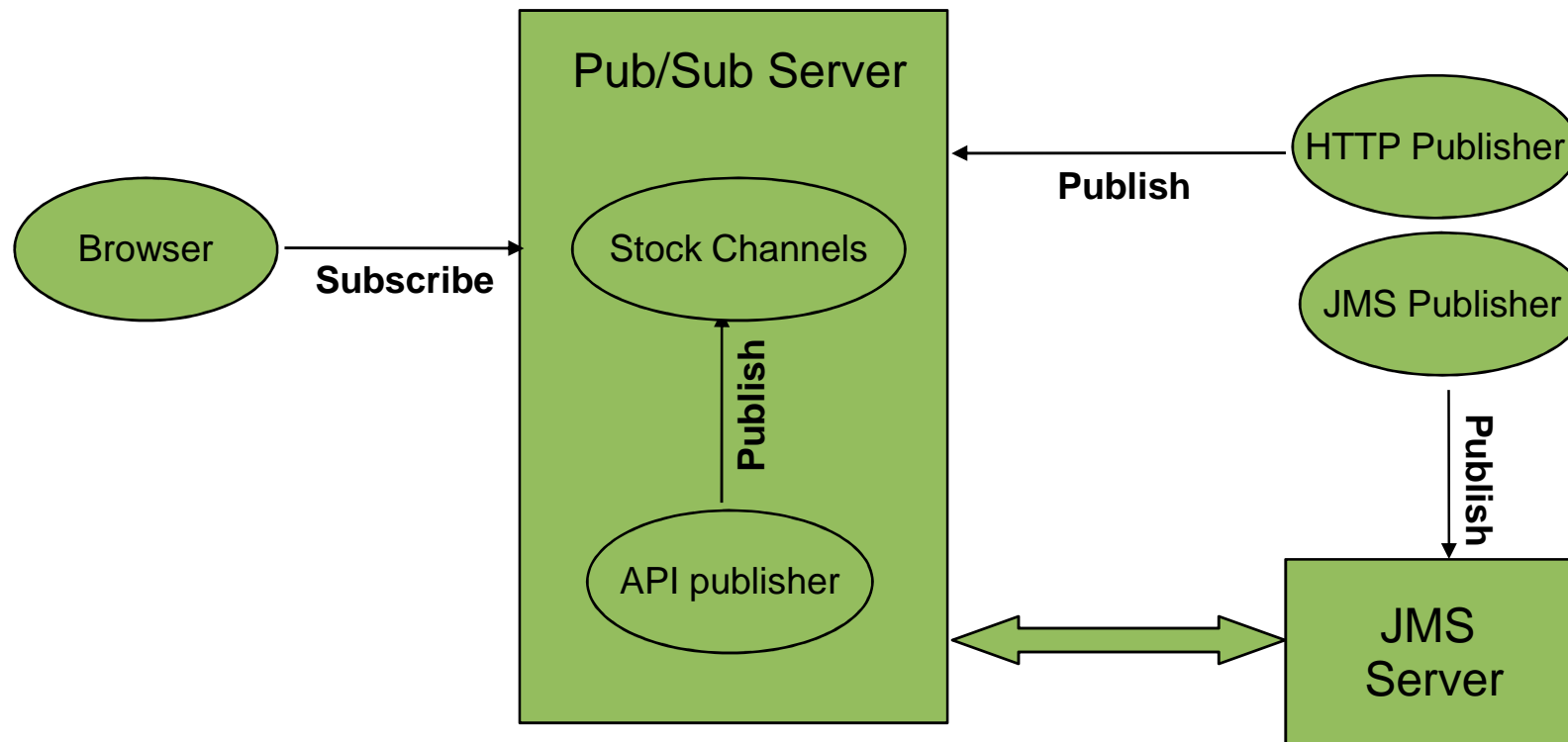
- ▶ View trade information of stock in portfolio
- ▶ Find and add a stock into portfolio
- ▶ Remove a stock from portfolio
- ▶ View price chart of selected stocks

Demo(Stock Portfolio)

- There three kinds of publishers
 - ▶ HTTP Bayeux based publisher
 - ▶ JMS based publisher
 - ▶ Server side public API based publisher

Demo(Stock Portfolio)

Only one publisher is needed at a time



The timing as well as the proposed features or functionality of any future product or service offering are subject to change at the sole discretion of BEA.



Thank you!