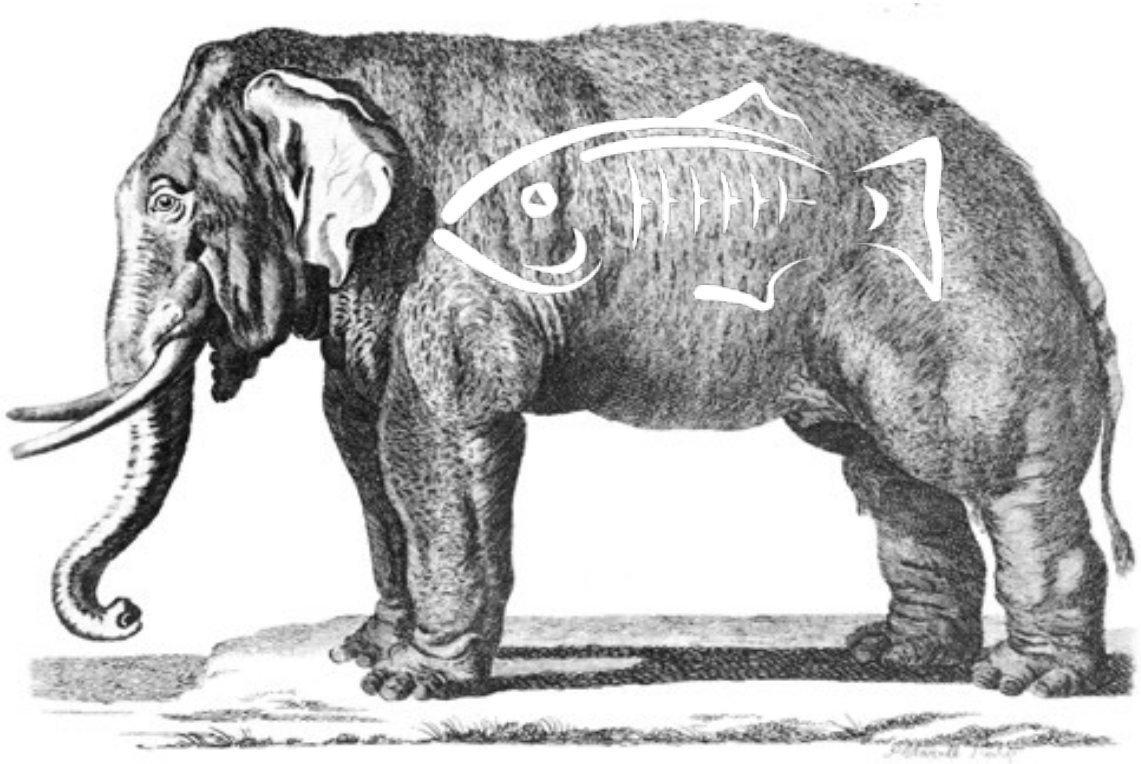
Java EE 6 - Update

Harpreet Singh –
GlassFish Portfolio Product Manager

Sun Microsystems

The Elephant In The Room



Here's what I can ...

Show

Say

Business As Usual

Business As Usual

=

Participate in an active community

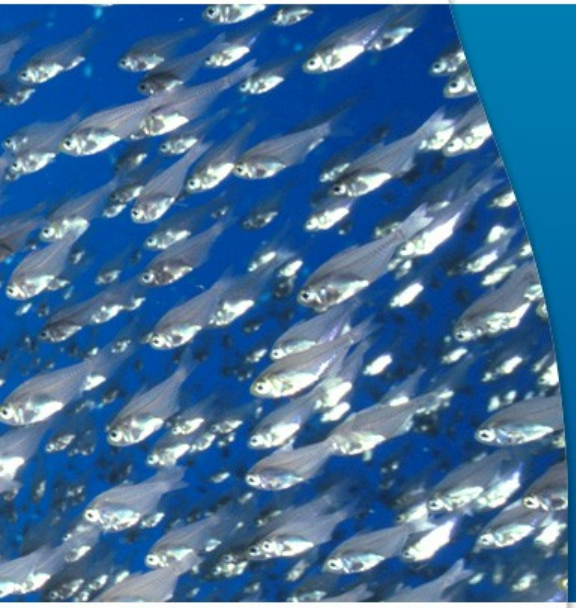
+

Create a Great Product

+

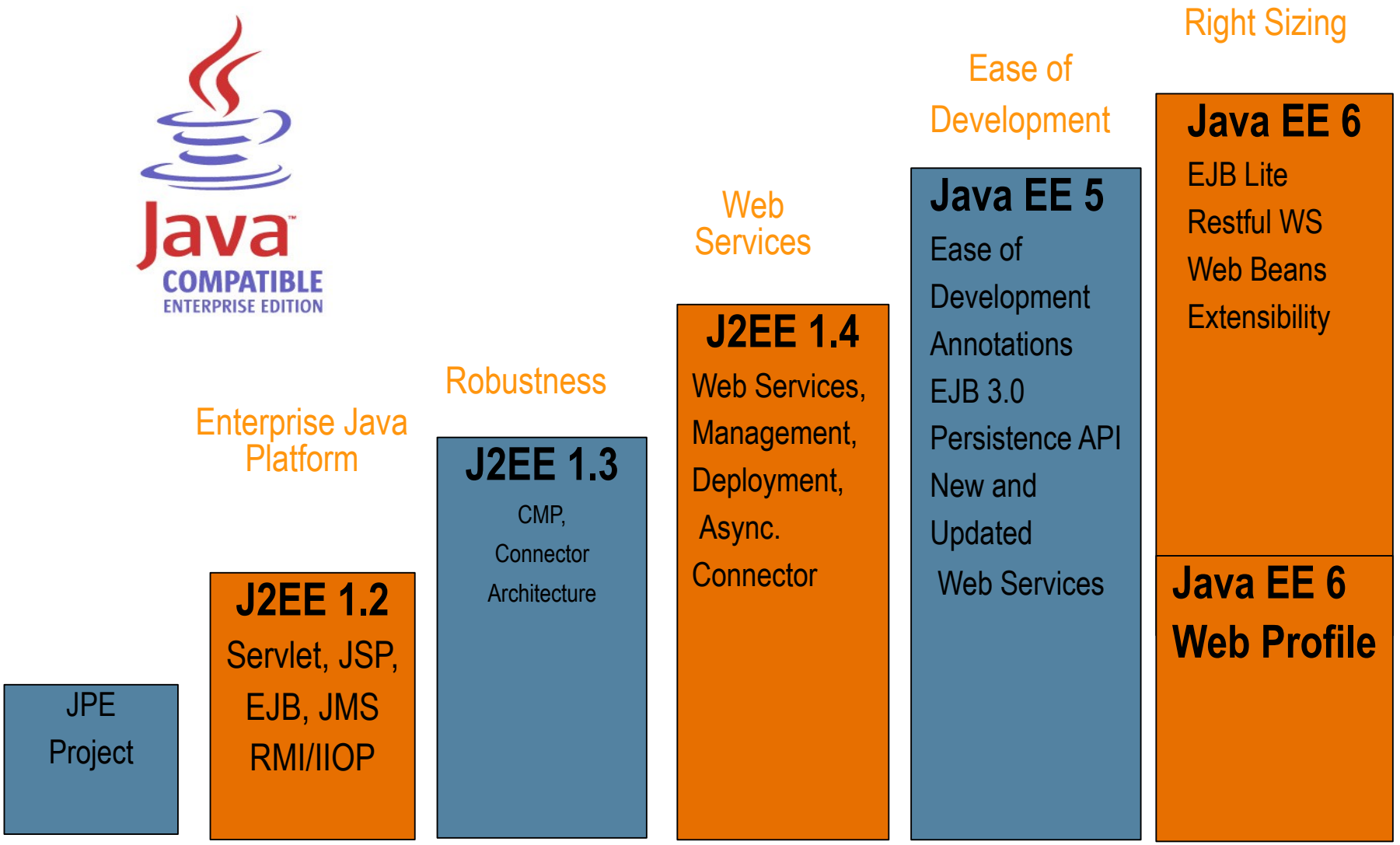
Kick Competitive Butt

Any Questions?



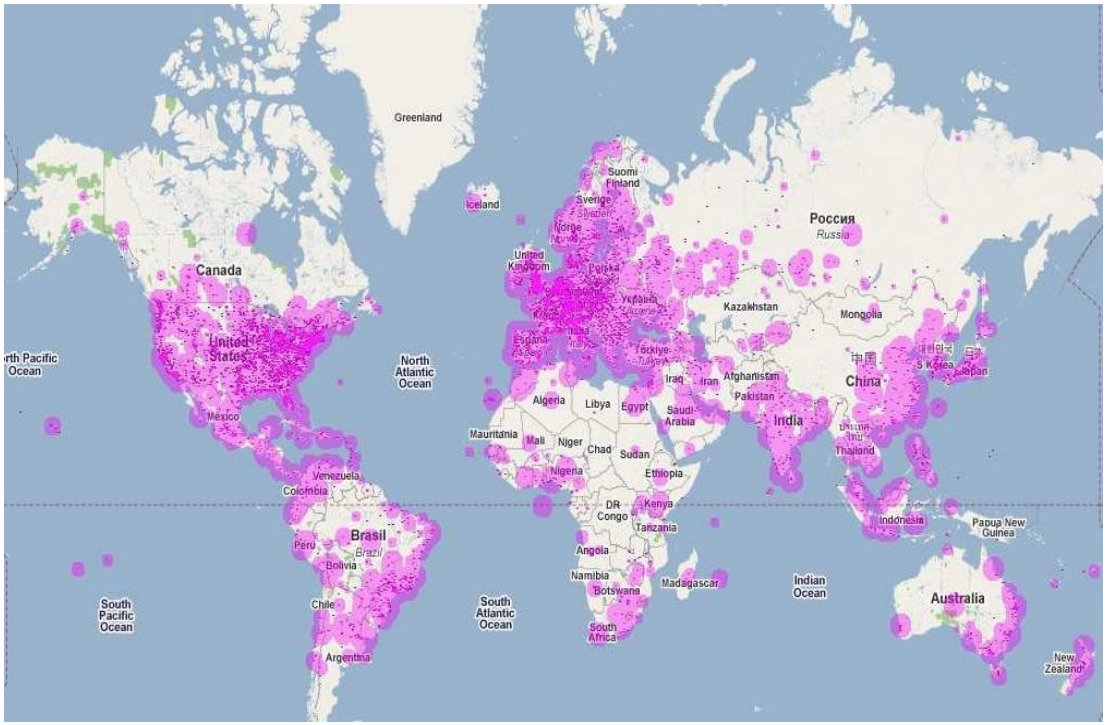
Business @ Hand
Java EE 6 Update

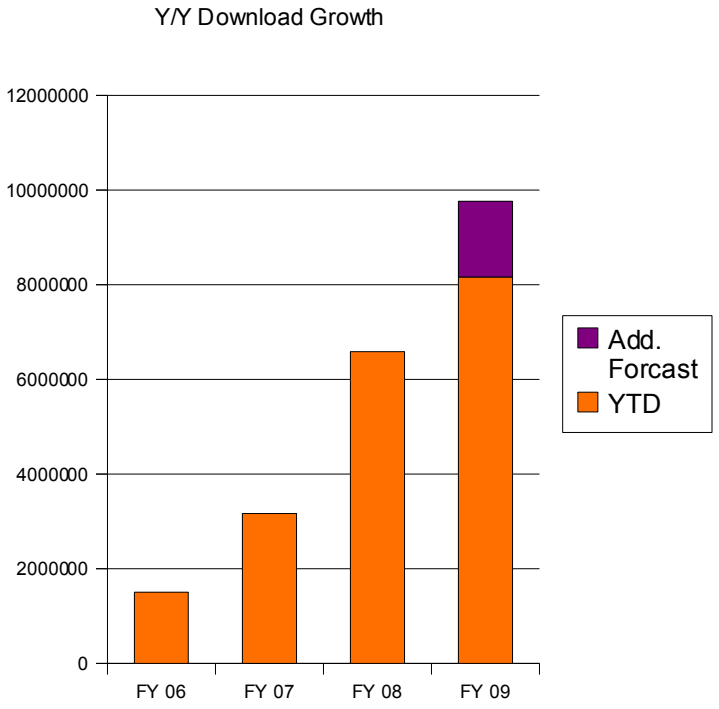
Java EE: Past & Present



Java EE Adoption

Over 18M Downloads since FY'06





Compatible Java EE 5 Implementations

Goals for the Java EE 6 Platform

- "Right Sizing"
  - Flexible, lighter weight
- Extensible
  - Embrace Open Source Frameworks
- Easier to use, develop on
  - Continue on path set by Java EE 5

Right Sizing the Platform: Profiles

- Make platform flexible
  - Decouple specifications to allow more combinations
  - Expands potential licensee ecosystem
  - Profiles
    - Targeted technology bundles
    - Defined through the JCP
    - First profile: Web Profile
      - Defined by the Java EE 6 Expert Group

Web Profile

- Fully functional mid-sized profile
  - Actively discussed in Java EE Expert Group and outside it
  - Technologies*
    - Servlet 3.0, EJB Lite 3.1, JPA 2.0, JSP 2.2, EL 1.2, JSTL 1.2, JSF 2.0, JTA 1.1, JSR-45, JSR-250

*Final set of technologies will be determined by the platform EG

Right Sizing the Platform: Pruning

- Make platform lighter
  - Makes some technologies optional
  - Pruned today, means
    - Optional in the next release
    - Deleted in the subsequent release
  - Pruned Technologies will be marked in the javadocs
  - Current pruning list
    - JAX-RPC, EJB 2.X Entity Beans, JAXR, JSR-88

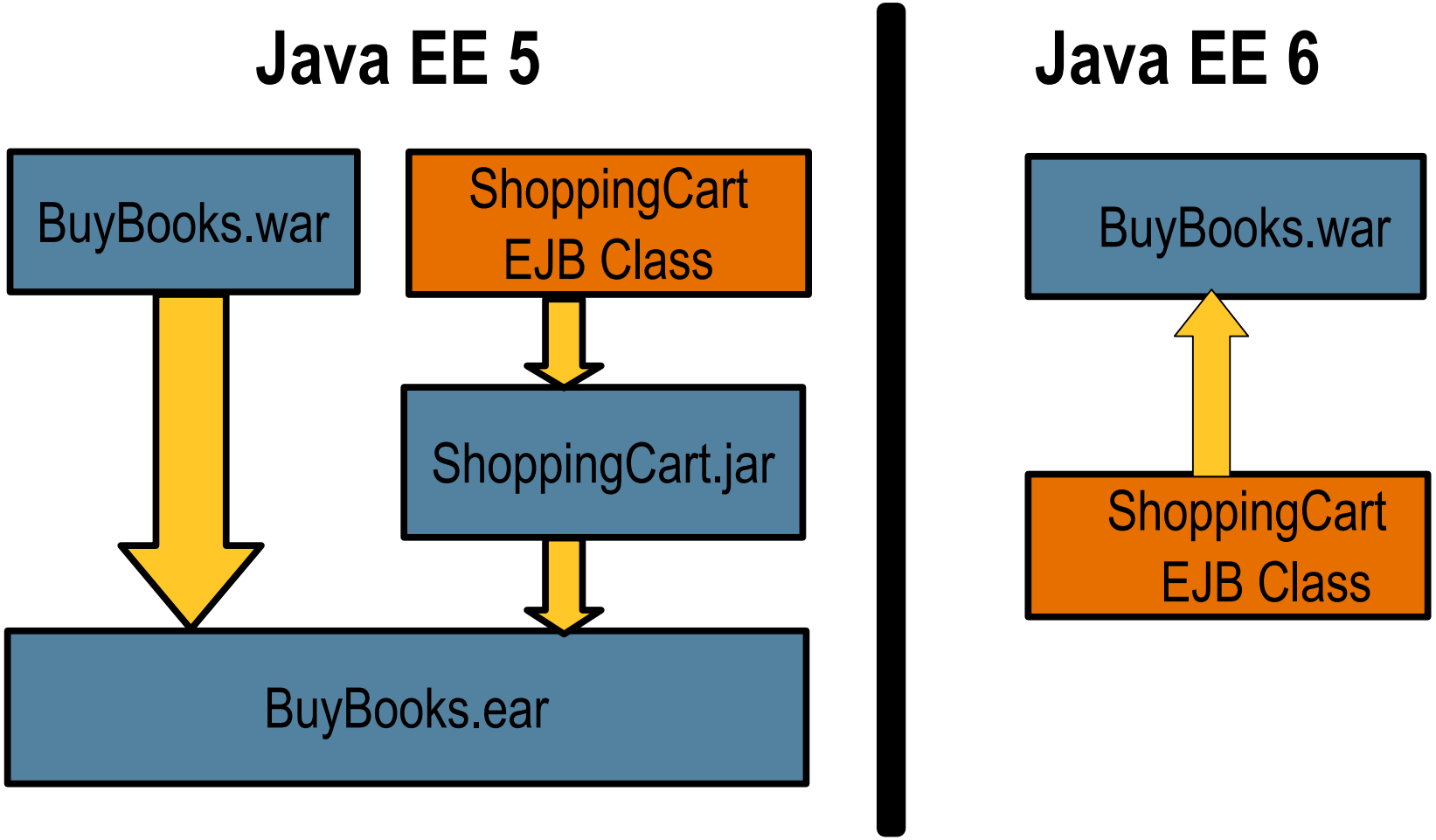
Extensibility

- Embrace open source libraries and frameworks
- Zero-configuration, drag-and-drop for web frameworks
  - Servlets, servlet filters, context listeners for a framework get discovered and registered automatically
- Plugin library jars using *web fragments*

Ease of Development

- Continue advancements of Java EE 5
- Primary focus: Web Tier
- Multiple Areas easier to use: EJB 3.1
- General principles
  - > Annotation-based programming model
  - > Reduce or eliminate need for deployment descriptors
  - > Traditional API for advanced users

Ease of Development: Adding an EJB to a Web Application



EoD Example - Annotations

Servlet in Java EE 5: Create two source files

```
<!--Deployment descriptor
   web.xml -->

<web-app>
   <servlet>
      <servlet-
        name>MyServlet
      </servlet-name>
       <servlet-class>
         com.foo.MyServlet
       </servlet-class>
    </servlet>
    <servlet-mapping>
       <servlet-
name>MyServlet
       </servlet-name>
       <url-pattern>/myApp/*
       </url-pattern>
    </servlet-mapping>
     ...
</web-app>
```

```
/* Code in Java Class */

package com.foo;
public class MyServlet
extends HttpServlet {
public void
doGet(HttpServletRequest
req,HttpServletResponse res)

{

…

}



…

}
```

EoD Example - Annotations

Servlet in Java EE 6: In many cases a single source file

```
package com.foo;
@WebServlet(name="MyServlet", urlPattern="/myApp/*")
public class MyServlet {
   public void doGet(HttpServletRequest req,
                     HttpServletResponse res)
   {
      ...
   }
```

Java EE 6 Status

- Public reviews completed
- Majority of the specs are in Proposed Final Draft
- Preview release available for JavaOne
  - Reference Implementation is GlassFish V3
  - Regular RI source drops started
- Final release in Q3
  - Targeted for September
- Making the process more open

Java EE JSR Status

- EJB 3.1 PFD
- JPA 2.0 PFD
- Servlet 3.0 PFD
- JSF 2.0 FD Submitted
- Java EE 6.0 PFD June
  - Platform
  - Managed Beans
  - Web Profile
- Connectors 1.6 PFD
- JCDI 1.0 PFD June
- Bean Validation PFD

Java EE JSR Status Continued

- JACC 1.3 MR Final
- JASPIC 1.1 MR Final
- JSR 250 1.1 MR Soon
- JAXB 2.2 MR Final
- JAX-RS 1.1 MR Final
- JAX-WS 2.2 MR Final
- EL 2.2 MR Final
- JSP 2.2 MR Final

Servlet 3.0

- Use of annotations for greater ease of development
- Optional web.xml
- Better defaults
- Enable web framework pluggability
- Support for asynchronous processing
- Alignments with JSF 2.0 and JAX-RS 1.0

Servlet 3.0

- Annotation-based programming model
  - @WebServlet @WebFilter etc.
- Modular web.xml descriptor:
  - WEB-INF/lib/mylibrary.jar -> Meta_INF/web-fragment.xml
- Annotations and web fragments are merged
- Programmatic API for dynamic registration of servlets

Servlet 3.0 Async API

- Useful for Comet, chat rooms, long waits
- Must declare @WebServlet(asyncSupported=true)
- Then call

  AsyncContext ctx = ServletRequest.startAsync(req,res);
- AsyncContext can then either:

  dispatch(String path)

  start(Runnable action)
- AsyncContext.start(Runnable) must be paired with:

  complete()

EJB 3.1

Ease-of-use improvements

- No-interface view
  - One source file per bean
- Allow EJB component classes inside a web app
  - No ejb-jar
  - Use WEB-INF/classes
  - Shared component environment

EJB 3.1

- Singleton beans: @Singleton
  - Shared state
  - One instance per bean per server JVM
  - Support for container-managed concurrency and bean-managed concurrency
    @ConcurrencyManagement
- Light-weight asynchronicity
  - Async business methods @Asynchronous
  - Methods must return void or Future<T>

EJB 3.1

- EJB Timer Service enhancements
  - Cron-like scheduling
  - Initialization-time timer creation
    @Schedule(dayOfWeek=”Mon,Wed”)
- Embeddable EJB container
  - Provides use in Java SE environments
  - Bootstrapping API
    EJBContainer

EJB 3.1 Lite

Proper, portable subset of EJB 3.1

- Session beans (stateless, stateful, singletons)
  - Local, no-interface views
- Transactions
- Security
- Interceptors
- ejb-jar.xml or annotations
- Embeddable container API: EJBContainer
- OK to bootstrap on Java SE

Web Beans 1.0

- Adds contextual services to the platform
- Unifies JSF managed beans and EJB components
- Dependency injection framework
  - @LoggedIn User user;
- Binding type to select among alternatives
- Built-in scope management (request, session, conversation, application scopes)
- Deployment types to select families of beans (e.g. mock)
- Programmatic API: Manager, Bean, Context

Removing Barriers

- EJB components can be placed in web applications
  - No need to wrap them in an EJB jar archive
- Web Beans has a unified EL resolver
  - Place @Named(“foo”) on a bean to give it a name
  - Use “#{foo}” to refer to it from JSR
- Unify @Resource-style injection with Web Beans
- Apply javax.interceptor more widely

Java Persistence API 2.0

- Expanded object/relational mapping functionality
- Additions to the Java Persistence query language
- Metamodel API
- Type-safe Criteria API
- Pessimistic locking
- Standardized caching configuration
- Support for automatic validation

JSF 2.0

- Easier to develop components
- Better leverage annotations
- Declarative renderers
- View description language (similar to Facelets, JSFTemplating, Tiles)
- Intuitive Ajax for JSF
- Additions to the render kit: calendar, tree, tab view, etc.
- Streamlined rendering process via caching
- Scripting

Summary

- Right Sizing
  - Profiles
  - Pruning
- Extensible
- Ease of use and development
  - RI and source code available today

- More Information
  - Java EE Hub
    - http://java.sun.com/javaee
  - Java EE SDK Downloads
    - http://java.sun.com/javaee/downloads
  - JCP
  - http://jcp.org/en/jsr/detail?id=316

GLASSFISH
Q&A