

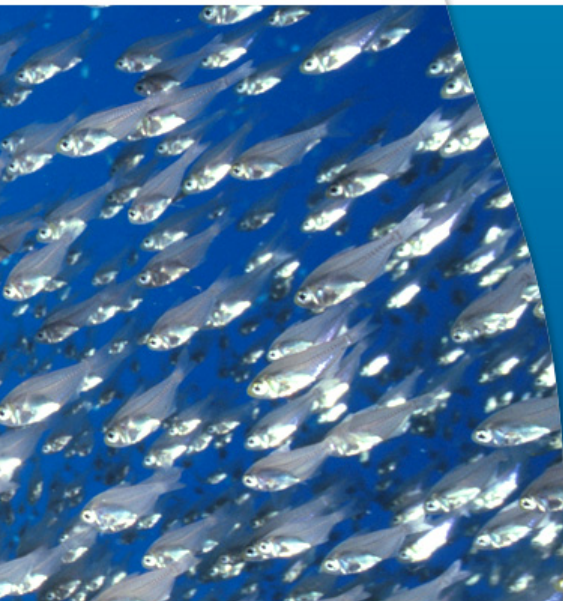


Java EE 6 Overview

Roberto Chinnici

Java EE 6 Specification Lead





Java EE 6

Final Release

December 10, 2009

Java EE 6 Platform

What's New?

- Several new APIs
- Web Profile
- Pluggability/extensibility
- Dependency injection
- Many improvements to individual APIs



Java EE 6 Platform

New Technologies

- JAX-RS 1.1
- Bean Validation 1.0
- DI 1.0
- CDI 1.0
- Managed Beans 1.0



Java EE 6 Platform

Updated Technologies

- EJB 3.1
- JPA 2.0
- Servlet 3.0
- JSF 2.0
- Connectors 1.6
- Interceptors 1.1
- JAX-WS 2.2
- JSR-109 1.3
- JSP 2.2
- EL 2.2
- JSR-250 1.1
- JACC 1.4
- JASPIC 1.0



Java EE: Past & Present



JPE
Project

Enterprise Java
Platform

J2EE 1.2

Servlet

JSP

EJB

JMS

RMI/IIOP

Robustness

J2EE 1.3

CMP

Connector
Architecture

Web
Services

J2EE 1.4

Web
Services
Management

Deployment

Async.
Connector

Ease of
Development

Java EE 5

Ease of
Development

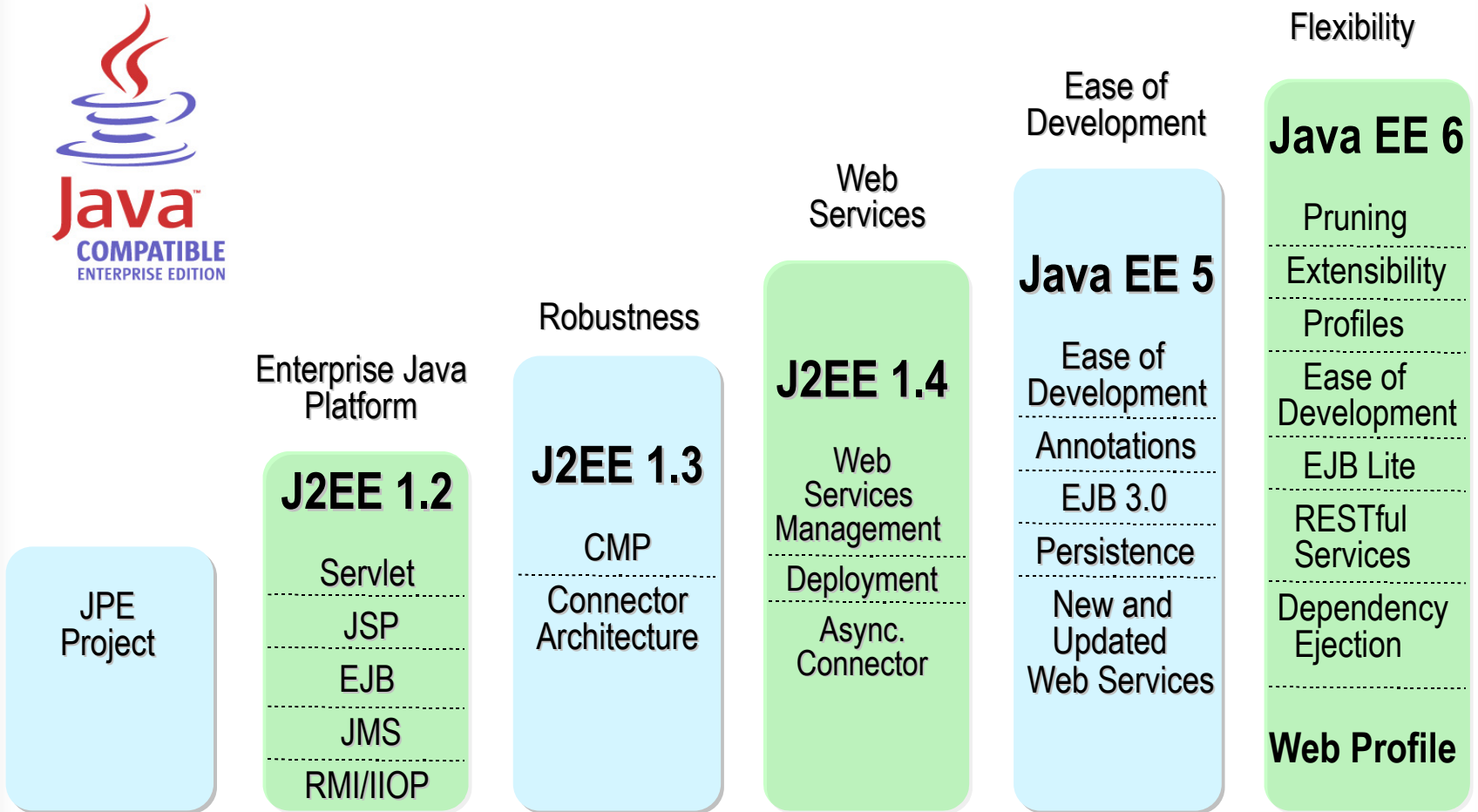
Annotations

EJB 3.0

Persistence

New and
Updated
Web Services

Java EE: Past & Present



Profiles

Increase Flexibility and Focus

- Bundles of technologies targeting specific classes of applications
- Decoupled from each other
- With compatibility guarantees



Web Profile

A Modern Java Web Application Stack

- First profile to be defined
- Mid-size, fully functional
- Complete, but not the kitchen sink



Web Profile

Contents

- Servlet
- JSP / EL
- JSTL
- JSF
- Bean Validation
- EJB Lite
- JPA
- JTA
- DI
- CDI
- Managed Beans
- Interceptors
- JSR-250



Pruning

Two-step Process

- Addresses bloat concern
- These technologies will become optional in the next release:
 - > JAX-RPC
 - > EJB 2.x Entity Beans
 - > JAXR
 - > JSR-88



Pluggability / Extensibility

Goals

- Focus on the web tier
- Simplify packaging of web apps
 - > Zero-configuration
- Create a level playing field for third party libraries and frameworks



Modular Web Applications

Servlet 3.0

- Libraries can contain a **web-fragment.xml** descriptor
- **web.xml** is now optional
- Ability to serve resources out of jars with **/META-INF/resources**



Web Fragment

META-INF/web-fragment.xml

```
<web-fragment
  version="3.0"
  xmlns="http://java.sun.com/xml/ns/javaee">
  <servlet>
    <servlet-name>welcome</servlet-name>
    <servlet-class>WelcomeServlet</servlet-class>
  </servlet>
  <listener>
    <listener-class>RequestListener</listener-class>
  </listener>
</web-fragment>
```

Pluggability

Servlet 3.0

- **ServletContainerInitializer** interface implemented by extensions
- **@HandlesTypes** to declare interest in one or more annotation types
- **ServletContext** now contains method to dynamically register servlets and filters



Servlet Container Extension

Registered using META-INF/services

```
@HandlesTypes(WebService.class)
public class WSInitializer
    implements ServletContainerInitializer {
    public void onStartUp(Set<Class<?>> c,
                          ServletContext ctx) {
        ServletRegistration r = ctx.addServlet(...);
        r.addServletMapping(...);
    }
}
```


Asynchronous Processing

Servlet 3.0

- New programming model for async request processing
 - > e.g. Comet, chat, push apps
- Opt-in model

`@WebServlet(asyncSupported=true)`

- Threads are managed by the container



EJB 3.1

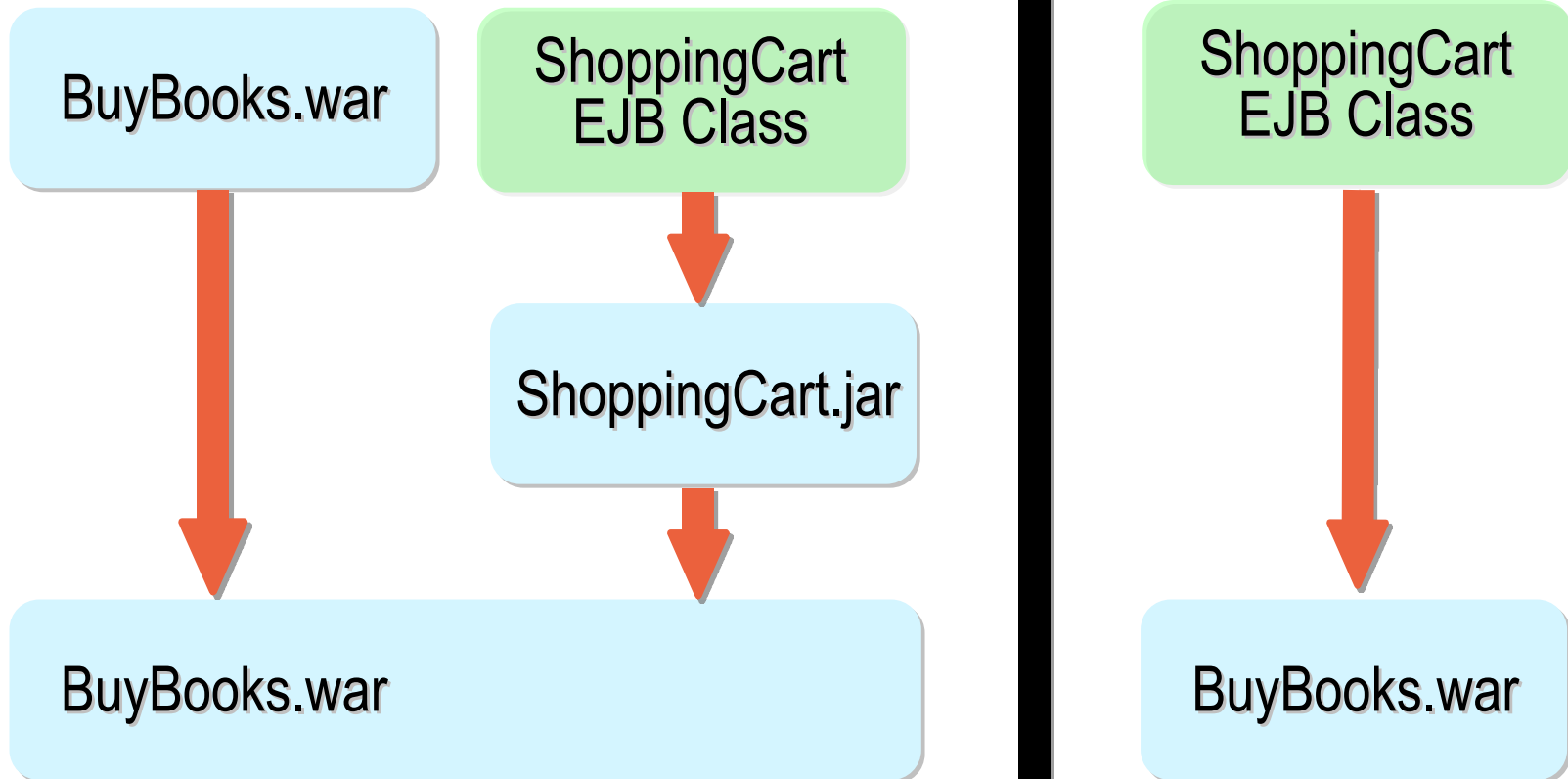
Ease of development

- **@Singleton** beans
- **@Startup** beans
- **@Asynchronous** invocations
- No interface view
- Define EJBs directly inside a web application
- **EJBContainer** API works on Java SE



Simplified Packaging

Adding an EJB to a Web Application



EJB 3.1 Lite

A subset of EJB 3.1

- All types of session beans
 - > stateful, stateless, singleton
- Declarative transactions and security
- Interceptors
- **ejb.xml** descriptor allowed



Dependency Injection

DI 1.0 / CDI 1.0

- New `@Inject` annotation
`@Inject @LoggedIn User user;`
- Beans autodiscovered at startup
- Injection metamodel (`BeanManager` API)
- `@Resource` still available for container resources



RESTful Web Services

JAX-RS 1.1

- Already widely adopted
- Really a high-level HTTP API
- Annotation-based programming model
- Programmatic API when needed



JAX-RS Resource Class

Identified by the `@Path` annotation

```
@Path("widgets/{id}")
@Produces("application/widgets+xml")
public class WidgetResource {
    public WidgetResource(@PathParam("id")
                          String id) { ... }

    @GET
    Widget getWidget() { ... }
}
```

Standard Validation API

Bean Validation 1.0

- Integrated with JSF, JPA
- Constraints represented by annotations

@NotNull

@Size(max=40) String address;

- Fully extensible

@Email String recipient;



Persistence

JPA 2.0

- Support for collections of basic types and embeddable objects
- JPQL enhancements
 - > e.g. CASE WHEN, NULLIF
- Pessimistic locking
- Criteria API for dynamic query construction



Sample Query Using the Criteria API

Uses the *canonical metamodel* classes

```
EntityManager em = ... ;
CriteriaBuilder cb = em.getCriteriaBuilder() ;
CriteriaQuery<Employee> cq =
    cb.createQuery(Employee.class) ;
Root<Employee> emp = cq.from(Employee.class) ;
cq.select(emp) ;
cq.where(cb.equal(emp.get(Employee_.lastName),
    "Smith")) ;
TypedQuery<Employee> query = em.createQuery(cq) ;
List<Employee> rows = query.getResultList() ;
```

Connectors

Java Connector Architecture 1.6

- Annotation-based programming model for resource adapters
- Generic work context
 - > QoS across EIS
- Security contract for RAs

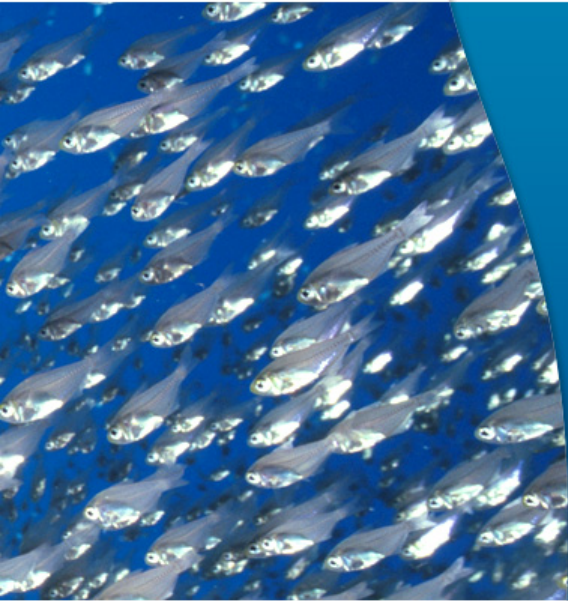


Ease of Development

Some of the strategies we applied

- Capture common patterns
- Fix inconsistencies
- Adopt what works
- Make APIs work better together
- Reducing packaging overhead
- Be transparent





Demo

Java EE 6

Summary

Java EE 6 Platform

- More powerful
- More flexible
- More extensible
- Easier to use



Resources

Java EE 6 and GlassFish v3

- **Java EE 6 Home**

java.sun.com/javaee

- **Java EE 6 Downloads**

java.sun.com/javaee/downloads

- **Upcoming Training**

java.sun.com/javaee/support/training

Java EE 6

- **Sun GlassFish Enterprise Server v3 Home**

www.sun.com/glassfishv3

- **Community Page**

glassfish.org

- **The Aquarium Blog**

blogs.sun.com/theaquarium

- **White Papers/Webinars**

<http://www.sun.com/glassfish/resources>

GlassFish



Thank You!

roberto.chinnici@sun.com

