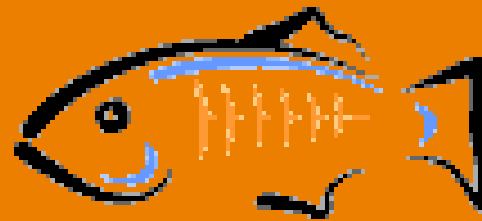




Java EE 6: What's Coming ?



Arun Gupta, GlassFish Guy
<http://blogs.sun.com/arungupta>
Sun Microsystems, Inc.



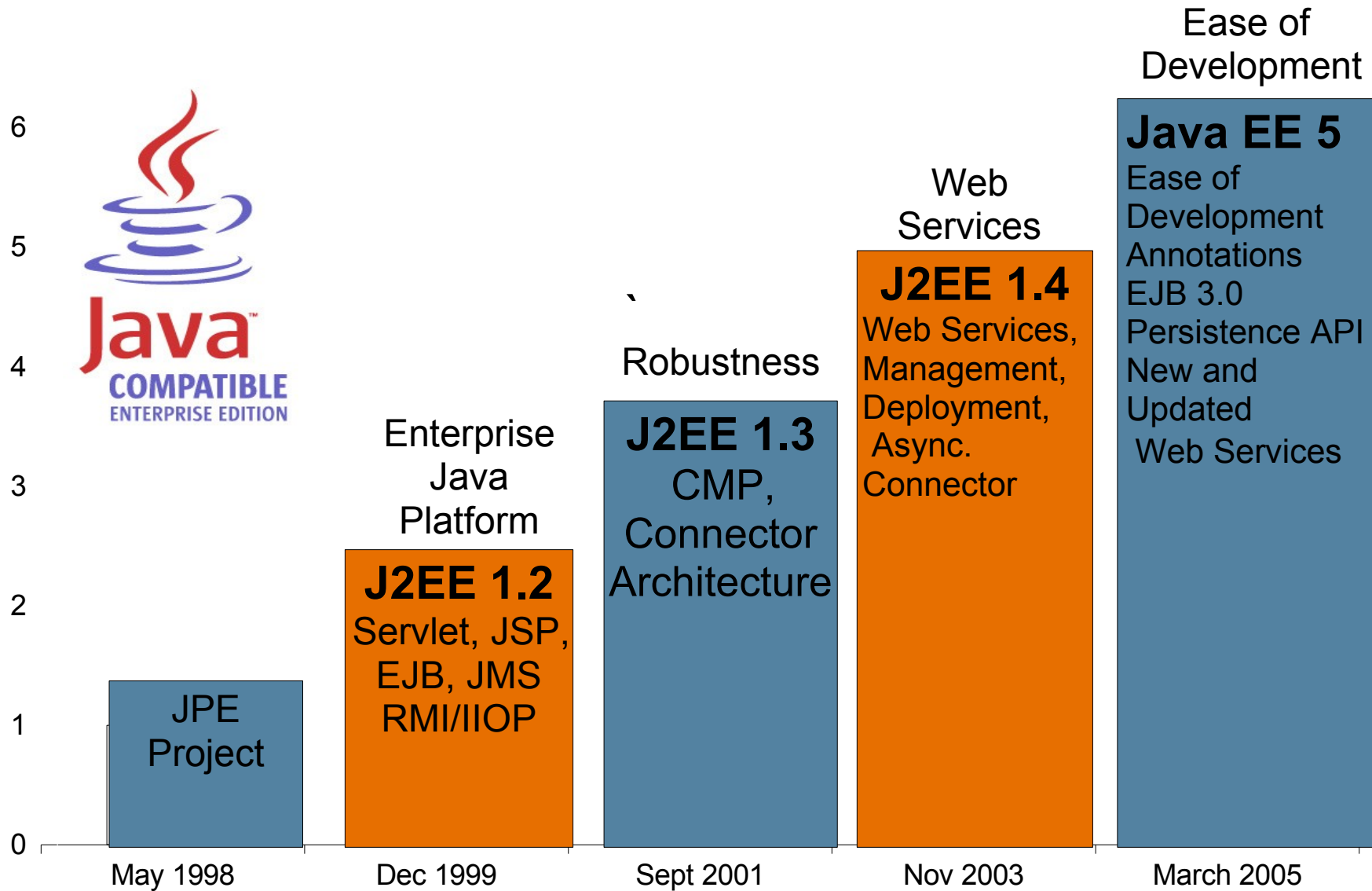
Who is Arun ?

- GlassFish Guy
- With Sun for over 10 years
- Specifications, Engineering, Standards, Interoperability, ...
 - > blogs.sun.com/arungupta
 - > twitter.com/arungupta



[ajax](#) [ajaxworld](#) [brazil](#) [conf](#)
[eclipse](#) [fitness](#) [gem](#)
[glassfish](#) [india](#) [indigo](#)
[interoperability](#) [javaone](#)
[javaone2008](#) [javaserverfaces](#)
[jax-ws](#) [jersey](#) [jmaki](#) [jpa](#)
[jruby](#) [jug](#) [lotd](#) [marathon](#)
[metro](#) [microsoft](#) [mysql](#)
[netbeans](#) [photography](#) [presos](#)
[railsconf](#) [results](#) [ruby](#)
[rubyonrails](#) [running](#)
[runninglog](#) [runsfm](#) [screencast](#)
[spotlight](#) [students](#) [sun](#)
[suntechdays](#) [svmarathon](#) [tango](#)
[theserverside](#) [totd](#) [training](#)
[traveltips](#) [v3](#) [vista](#) [wcf](#) [web2.0](#)
[webservices](#) [webtier](#) [windows](#)
[wsaddressing](#) [wsit](#) [\(all\)](#)

History of Enterprise Java



Java EE 6 - Goals

- More powerful
- More flexible
- Easier to learn and use
- Easier to evolve going forward

Java EE 6 – How ?

- **Profiles:** Proper subsets/supersets
 - > Web profile defined
- **Pruning:** Make some components optional
- **Extensibility:** Embrace open source libraries & frameworks

Java EE 6 - Specs

- EJB 3.1 (JSR 318)
- JPA 2.0 (JSR 317)
- Servlet 3.0 (JSR 315)
- JSF 2.1 (JSR 314)
- JAX-RS 1.1 (JSR 311)
- Connector Architecture 1.6 (JSR 322)
- Bean Validation 1.0 (JSR 303)

Java EE 6 – Other Updates

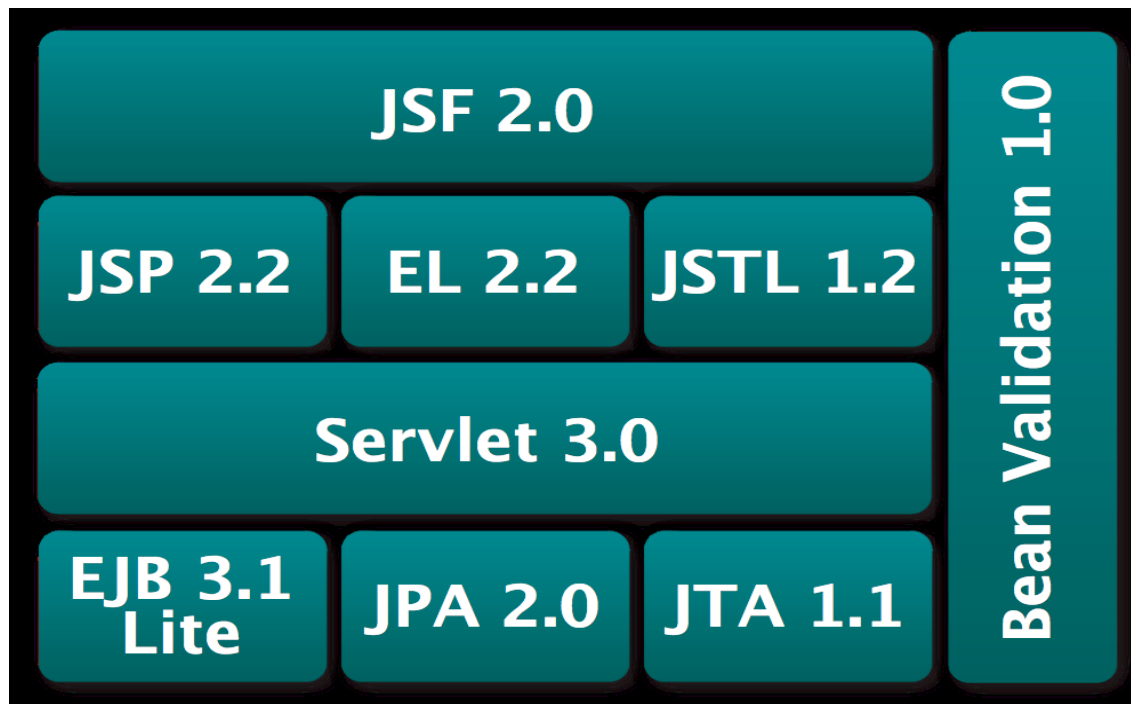
- JAXB 2.2
- JAX-WS 2.2
- JSR-109 1.3
- EL 1.2 / JSP 1.2
- JASPIC 1.1 (JSR-196)
- Common Annotations 1.1 (JSR-250)
- Java SE 6 underneath

Java EE 6 - Profiles

- A profile is a targeted bundle of technologies
- Rules set by the Java EE Platform spec
- Profiles can be subsets, supersets, overlapping
- Define immediately a Web Profile
- JCP process for future profiles

Java EE 6 - Web Profile

- A fully functional mid-sized profile for web apps
- Extensible via Servlet 3.0 pluggability contracts



Servlet 3.0

- New revision of the Servlet specification
- Adding pluggability, async support and security enhancements
- “web.xml” descriptor will be optional
- Use of annotations and other newer language features
- Ability to login and logout
- File upload support

Servlet 3.0 – Pluggable Extension Libraries

- Libraries in WEB-INF can self register
- “web.xml” fragments
- Powerful dynamic registration

Java Server Faces 2.0 – Top 5 Goals

- Make custom components much easier to develop
 - > Use Facelets/templating to build complex composite components
- Ajax support
 - > The best ideas from RichFaces, ICEFaces, ADFFaces
- Page description language (PDL)
 - > Based on Facelets
- Reduce the configuration burden
 - > Leverage Annotations
- Better compatibility between JSF component libraries
 - > Especially Ajax Components

Mojarra – JSF Implementation

- Most up to date implementation of the JSF 2.0 specification
- Open source
 - > <https://javaserverfaces.dev.java.net>
- Mojarra users
 - > <https://javaserverfaces.dev.java.net/customers/expert-group.html>

EJB 3.1

- Lots of usability improvements
 - > Works inside web applications
 - > Optional local business interface (@Local)
 - > Singleton Session Beans
 - > Application Initialization and Shutdown Events
 - > Timer service enhancements
 - > “Lite” standalone version
 - > Small subset of EJB 3.1 for use in Web profile
 - > Same application can be deployed to Web and Full profile

“Lite” vs Full functionality

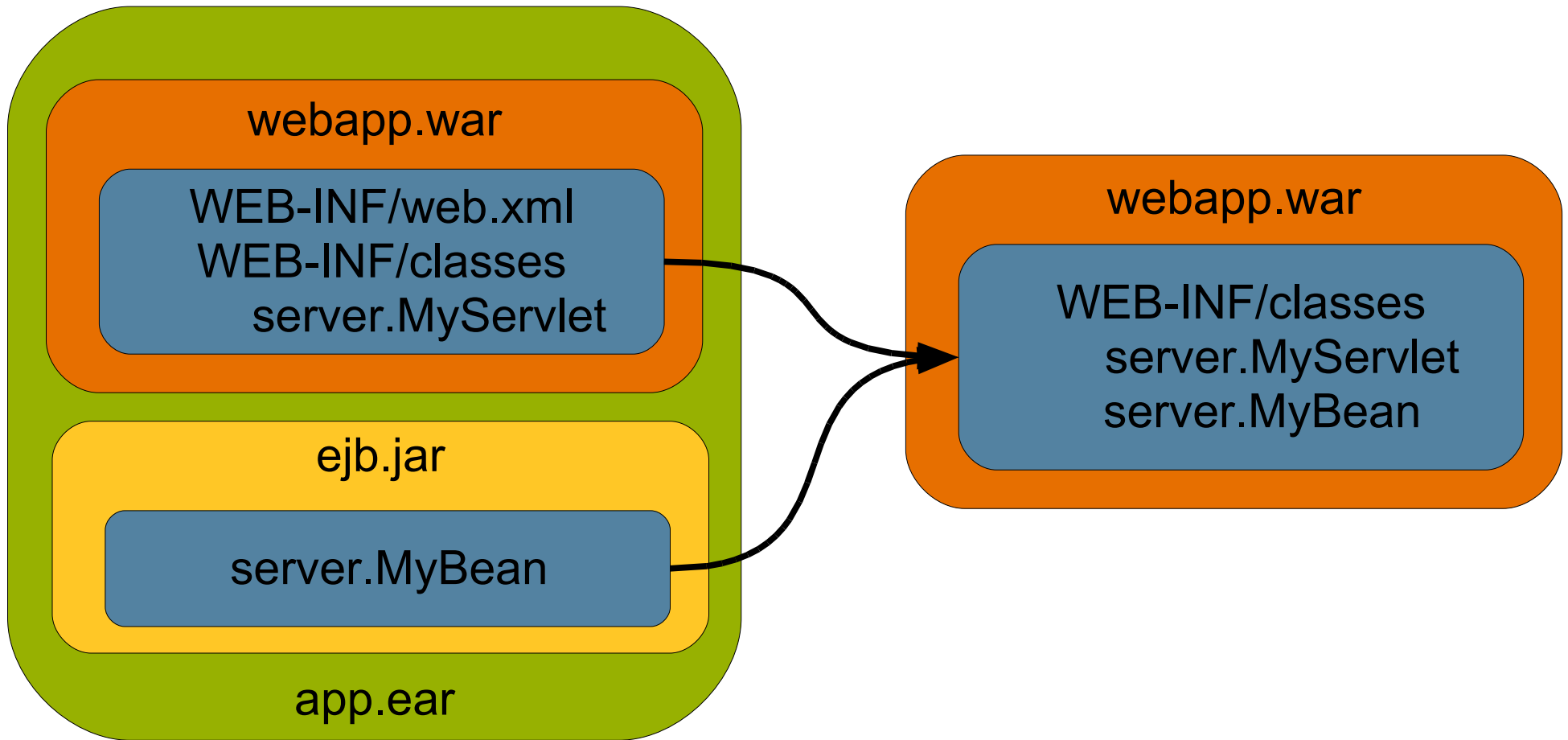
“Lite”

- Local Session Bean
- Annotations/ejb-jar.xml
- CMT/BMT
- Declarative Security
- Interceptors
- (Also requires JPA 2.0 API / JTA 1.1 API)

Full = Lite + the following:

- Message Driven Beans
- EJB Web Service Endpoints
- RMI-IIOP interoperability
- 2.x/3.x Remote View
- 2.x Local View
- Timer Service
- CMP / BMP

EJB 3.1 Packaging



EJB 3.1 Sample Code

```
@Startup @Singleton
public class CacheBean {
    private Cache cache;

    @PostConstruct
    void init() { ... }

    @Schedule(minute="*/5", hour="*")
    void refresh() { ... }

    @Lock(READ)
    public Object get(String key) { ... }
}
```

Bean Validation (JSR 303)

- Standard set of constraints
- Works with JSF, JPA, JAX-RS
- Comprehensive validation API

Bean Validation Sample Code

```
@Entity
public class Customer
    implements Serializable {

    private Long id;

    @NotNull @Length(max=80)
    private String name;

    @NotNull @Valid
    private Address address;

    ...
}
```

JAX-RS 1.1

- Java API for RESTful Web Services
- Annotation-based server-side API
- HTTP centric
- Server-side only
- Servlet or SE deployment
- Five open-source implementations already
 - > Jersey, JBoss RESTEasy, Restlet, Apache CXF, Triaxrs

JAX-RS Example

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

    private Widget w;

    public WidgetResource(@PathParam("id") String id) {
        this.w = locateRecord(id);
    }

    @GET
    Widget getWidget() {
        return w;
    }

    @PUT
    Widget updateWidget(Widget update) {
        w = processUpdate(update);
        return w;
    }
}
```

GlassFish v3

- Modular, Extensible, OSGi-based
- Developer-friendly
- Scales from embedded to carrier-grade

GlassFish v3 – Fast and Easy

- Java EE 6 Standard
- NetBeans and Eclipse integration
- Deploy on Save
- Session Preservation

Java EE 6 - Schedule

- Public review of all specs completed
- Proposed final draft in progress (almost all done)
- Final release in September 2009

Java EE 6 Platform

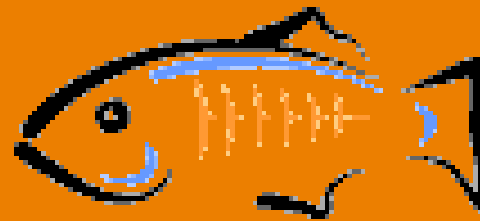
Coming Sep '09

References

- <http://glassfish.org>
- Portuguese Aquarium:
http://blogs.sun.com/theaquarium_br/



Obrigado!



Arun Gupta, GlassFish Guy
<http://blogs.sun.com/arungupta>
Sun Microsystems, Inc.

