

Contexts and Dependency Injection for Java EE (CDI) Magical Mystery Tour

Reza Rahman

Expert Group Member, Java EE, EJB and JMS

Resin EJB 3.1 Lite Developer/Community Outreach Advocate

Author, EJB 3 in Action

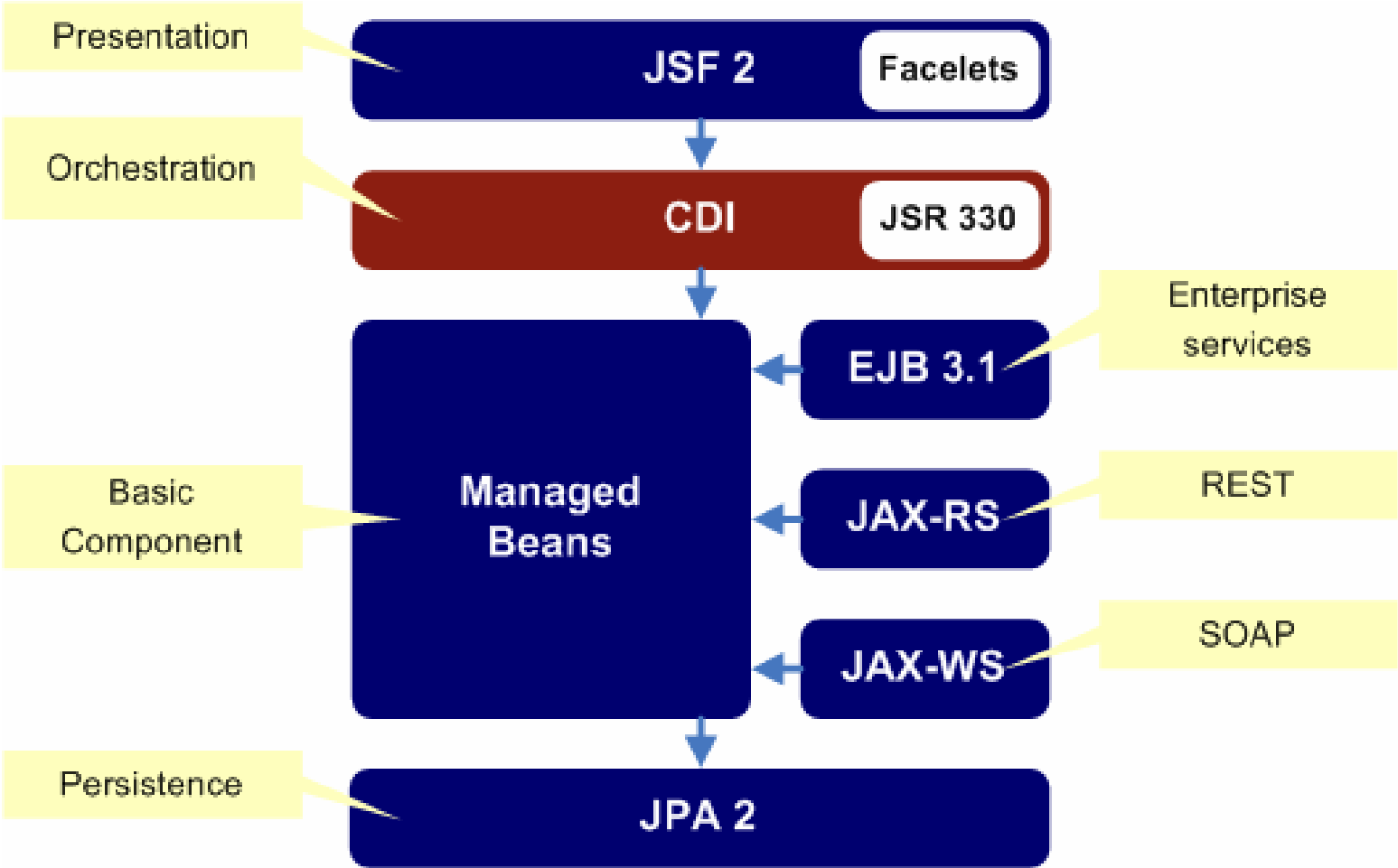
Nova/DC Java Users Group

June, 2011

Contexts and Dependency Injection for Java EE (CDI)

- **Cool new kid on dependency injection block**
- **Synthesizes best ideas from Seam 2, Guice and Spring**
- **Many innovations on its own right**
- **Loose-coupling, Java-centric type-safety, annotations, expressiveness and ease-of-use**
- **Makes Java EE much more flexible, testable, pluggable and extensible**
- **Large ecosystem rapidly evolving around CDI**

CDI and Java EE 6



CDI Features

- **Basic dependency injection**
 - **@Inject, @Qualifier, @Stereotype, @Alternative, @All, @Any**
- **Component naming**
 - **@Named**
- **Custom Object Factories**
 - **@Produces, @Disposes, InjectionPoint**
- **Context management**
 - **@RequestScoped, @SessionScoped, @ConversationScoped, @ApplicationScoped, @Scope**

CDI Features

- **Lightweight Events**
 - **Event, @Observes**
- **Interceptors/Decorators**
 - **@Interceptor, @InterceptorBinding, @AroundInvoke, InvocationContext, @Decorator, @Delegate**
- ***Portable extensions SPI!***

Weld

- **CDI reference implementation**
- **Weld reference excellent documentation for CDI**
- **Evolved from Seam 2 core**
- **Included in GlassFish and JBoss AS**
- **Can be used with Tomcat, Jetty, Java SE**
- **Forms basis for Seam 3**



CanDI

- **Caucho implementation forms core of Resin itself**
- **CanDI XML used to configure Resin**
- **Use of annotations like @TransactionAttribute, @Schedule, @Asynchronous, @RolesAllowed, @RunAs, @Lock, @Startup and @Remote outside EJB in plain CDI managed beans**
- **@PoolScoped, @TransactionScoped, @ThreadScoped**
- **Optimized embedded container based Test Framework**
- **CDI-PHP integration**
- **Partnership with CDISource**



OpenWebBeans

- **Apache implementation of CDI included in Geronimo**
- **Closely related to Apache OpenEJB**
- **Great integration with Tomcat along with OpenEJB (TomEE)**
- **Extended injection of JMS objects (sessions, connection factories)**
- **Runs in standalone Java SE**
- **Portable extensions for major Apache projects**



Apache

Demo Time!



CDI Ecosystem

Implementations



Weld



CanDI



OpenWebBeans

Portable Extensions



CODI
APACHE
MyFaces



Jersey

Struts

Runtimes



resin®



WebSphere®



jetty://



TmaxSoft



Tools



CDI Plug-ins (Portable Extensions)

Testing	JSF Enhancements	XML Configuration	EJB-style Transactions	JMS Abstractions	Security
Struts 2	Wicket	GWT	Logging	JavaMail Abstractions	REST Enhancements
JavaScript Remoting	Quartz	Internationalization	Bean Validation Enhancements	JasperReports	iText
Drools	jBPM	JBoss ESB	Scripting	Servlet Enhancements	Code Generation
ZK (RIA)	Content Management	Social Networking	Cloud API	Seam 2	Spring

Java SE Bootstrap (Weld)

```
@Singleton
public class AlertListener {
    ...
    public void init(
        @Observes ContainerInitialized event,
        @Parameters List<String> parameters) {
        showSplash();
        setupUi();
        showUi(parameters);
    }
    ...
}
```

Injecting Loggers (CODI)

```
public class DefaultAccountDao implements AccountDao {
    @Inject
    private Logger logger;

    @PersistenceContext
    private EntityManager entityManager;

    public void addAccount(Account account) {
        ...
    }
    ...
}
```

Extended Scopes (Seam + CDI)

```
@Named @RenderScoped
public class MyRenderScopedBean {
    ...
}
```

```
@Named @ViewScoped
public class MyViewScopedBean {
    ...
}
```

```
@Named @WindowScoped
public class MyWindowScopedBean {
    @Inject private Conversation conversation;
    ...
}
```

JMS Resource Injection (Seam)

```
public class TransferSender {
    ...
    @Inject
    @JmsSession(
        transacted=true,
        acknowledgementType=Session.AUTO_ACKNOWLEDGE)
    private Session session;

    @Inject
    @JmsDestination(jndiName="jms/TransferQueue")
    private MessageProducer producer;
    ...
    public void sendTransfer(Transfer transfer) {
        ...
        producer.send(session.createObjectMessage(transfer));
        ...
    }
    ...
}
```

REST(Easy) Client (Seam)

```
@Inject
@RestClient("http://services.actionbazaar.com/bid")
private BidService bidService;
...
bidService.addBid(bid);
```

Scripting (JSR-223) Integration (CODI)

```
@Inject @ScriptLanguage(JavaScript.class)
private ScriptExecutor scriptExecutor;
...
return scriptExecutor.eval("10 + 4", Double.class);
```

```
@Inject @ScriptLanguage(JavaScript.class)
private ScriptBuilder scriptBuilder;
...
return scriptBuilder.script("x + y").namedArgument(
    "x", a).namedArgument("y", b).eval(Double.class);
```

```
<h:outputText value="#{sExec.js['[a:#{bean1.result},
    b:#{bean2.result}']['a + b']}"/>
```

JavaScript Remoting (Seam)

```
@Named("bidService") @Stateless
public class DefaultBidService implements BidService {
    ...
    @WebRemote public void addBid(String bidder, String item,
        Double amount) {
        ...
    }
    ...
}

<script type="text/javascript">
    function addBid() {
        ...
        Seam.createBean("bidService").addBid(bidder,
            item, amount);
        ...
    }
</script>
...
<button onclick="javascript:addBid()">Bid!</button>
```

Servlet Life-Cycle as CDI Events (Seam)

```
public void onServletContextInitialization(  
    @Observes @Initialized ServletContext context) {  
    ...  
}
```

```
public void onSessionInitialization(  
    @Observes @Initialized HttpSession session) {  
    ...  
}
```

```
public void onRequestInitialization(  
    @Observes @Initialized @Path("/bid")  
    HttpServletRequest request) {  
    ...  
}
```

Servlet Object Injection (Seam)

```
@Inject ServletConfig configuration;  
@Inject ServletContext context;  
@Inject HttpSession session;  
@Inject HttpServletRequest request;  
@Inject @RequestParam("id") String bidId;  
@Inject @HeaderParam("User-Agent") String userAgent;  
@Inject HttpServletResponse response;  
@Inject List<Cookie> cookies;
```

Summary

- **Next generation dependency injection and context management**
- **Unifies web, transaction and persistence tiers (JSF, EJB 3 and JPA)**
- **Implementations from JBoss (Weld), Caucho (CanDI) and Apache (OpenWebBeans)**
- **Rich ecosystem of implementations, supported runtime platforms, portable extensions and tools rapidly evolving around CDI**

References

- JSR 299: Contexts and Dependency Injection for Java EE, <http://jcp.org/en/jsr/detail?id=299>
- Weld Reference, <http://docs.jboss.org/weld/reference/1.0.0/en-US/html/>
- CDI Advocacy, <http://cdisource.org/site/>
- Weld, the JBoss reference implementation for JSR 299, <http://seamframework.org/Weld>
- OpenWebBeans, Apache implementation of JSR 299, <http://openwebbeans.apache.org>
- CanDI, the JSR 299 implementation for Caucho Resin, <http://caucho.com/projects/candi/>

References

- Seam 3, <http://seamframework.org>
- Apache MyFaces CODI, <http://myfaces.apache.org/extensions/cdi/>
- Arquillian, <http://www.jboss.org/arquillian>
- Seam Forge, <http://seamframework.org/Documentation/SeamForge>