



'Boss your way to SCEA!

..... Or "Understanding Enterprise J2EE Applications"

© JBoss Inc. 2006

What is 'SCEA'?


- Why should I try to 'Boss my way to it?

2

What is the SCEA exam?


- Sun Certified Enterprise Architect
- 3 Part exam
- Part 1 is a multiple choice, taken at ProMetric.
- Part 2 is an application design assignment, usually about 100 hours.
- Part 3 is a 4 question essay exam, taken as you submit part 2 (again at ProMetric).



3

We'll concentrate on Part 1



- With hints for Parts 2 and 3 later.



4

Why should I use JBoss to reinforce SCEA concepts?


- It's free.
- Tutorials and how-tos are easily found.
- It's well-known in the industry.
- It's the leading J2EE application server. ☺

5

Part 1

- Concepts
- Common Architectures
- Legacy Connectivity
- EJB
- EJB container model
- Protocols
- Applicability of J2EE
- Design Patterns
- Messaging
- Internationalization
- Security



6

Our approach

- The exam is too big to walk in one sitting, so we'll take a high-level overview and delve down into areas that lend themselves to further study.

Concepts

Business requirements vs. Service level requirements

- Business requirements tell what the system has to do rules-wise. They are unique to your type of application.
- Service level requirements (Quality of Service requirements) address concerns common to software applications.

Concepts

Example Business Requirements

- The system should increment a frequent-user account balance with every purchase.
- The system should calculate days remaining using a business calendar.
- The system should allow extra time for users outside this zip code.

Concepts

Service-level Requirements

- Performance – response time, throughput
- Scalability – Make it bigger
- Reliability – is it consistent under load
- Availability – is it accessible
- Extensibility – add functionality
- Maintainability – correct flaws
- Manageability – manage continued health
- Security – cannot be compromised

Concepts

QoS requirements require tradeoffs

- More performance might mean less maintainability or extensibility.
- More security might mean less performance.
- Etc. etc.

Concepts

Scaling

- Vertical (Add machine resources to existing servers)
- Horizontal (Add servers)

'Boss your way to horizontal scaling!

- JBoss offers clustering features
- Now available with 4.0.4 installer!
- Based on JGroups, JBossCache
- Note application design considerations– session information can be kept in different places. (On the client, on the web tier, on the ejb tier, in the database)



13



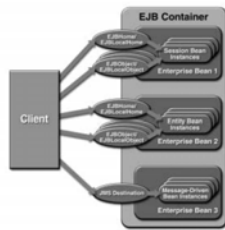
EJBs

- Contain business logic
- Created and managed by EJB container
- Client access is mediated by container
- Metadata for security and transactions *CAN* be external (new annotations)
- If your application doesn't require security or transaction support, may not need EJB

14



EJBs, as depicted by Sun



15



EJBs

What kind of bean do you need?

- Session Beans (handle actions, verbs)
- Entity Beans (handle data, nouns)
- Message Driven Beans (Asynchronous)

16



EJBs – Session Beans

Contain business rules, workflow

- ✓ Stateless
- ✓ Stateful
- ✓ Statefulness deals with 'memory' in between client invocations. Will the bean remember your state variables?
- ✓ Statefulness declared with XML or annotation.

17



Example Session Bean

```
package trail.slsb;

import javax.ejb.*;

@Stateless
@Remote (RemoteCalculator.class)
public class CalculatorBean2 implements Calculator, RemoteCalculator {

    public double calculate (int start, int end, double growthrate, double
    saving) {
        double tmp = Math.pow(1. + growthrate / 12., 12. * (end - start) + 1);
        return saving * 12. * (tmp - 1) / growthrate;
    }

    public String getServerInfo () {
        return "This is the JBoss EJB 3.0 Trail Blazer";
    }
}
```

18



Example Entity Bean

```
@Entity
@Table(name = "fund")

public class Fund implements Serializable {
    private int id;
    private String name;
    private double growthrate;

    public Fund () {
    }

    public Fund (String name, double growthrate) {
        this.name = name;
        this.growthrate = growthrate;
    }

    @Id(generator = GenerationType.AUTO)
    public int getId () {
        return id;
    }

    public void setId (int id) {
        this.id = id;
    }
    .....
}
```

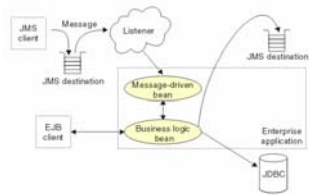
19

About that Entity Bean....

- It knows which data source to use through persistence.xml, an xml config file.
- So it's not all annotations, all the time. JBoss wants to give you a choice for everything, though.
- Annotations seem to be an optional path. You can still use the xml configuration files if you prefer them.

20

MDB with Business Delegate



21

Example MDB

```
@MessageDriven(activationConfig =
{
    @ActivationConfigProperty(propertyName="destinationType",
        propertyValue="javax.jms.Queue"),
    @ActivationConfigProperty(propertyName="destination",
        propertyValue="queue/mdb")
})
public class CalculatorBean implements MessageListener {

    public void onMessage (Message msg) {
        try {
            TextMessage tmsg = (TextMessage) msg;
            .....
        }
    }
}
```

22

BMP or CMP (And what about DAOs?)

- Container managed persistence means your logic knows nothing about the database. (It's solid, well tested, etc.)
- Bean managed persistence might offer better performance, though.
- Sun (and others) recommend using DAOs if you use BMP. That way it's easier to upgrade when tools negate performance advantage.

23

Bean Pooling

The application server can provide better performance by pooling Stateless beans and Entity Beans.

It makes sense.... Stateful beans can't be pooled, they have state for only 1 client...

24

Bean Passivation

- Means the bean can be brought 'out of memory' and written to disk, later re-hydrated.
- Works for Stateful beans and Entity beans.

- Makes sense, why would you want to passivate a shared, stateless thing like a Stateless bean?

What about transactions?

- Know the ACID principals:
 - ✓ Atomic
 - ✓ Consistent
 - ✓ Isolated
 - ✓ Durable

Transactions in EJB

- CMT vs. BMT
- (Container managed vs. Bean managed)
- Session beans can use either.
- Entity beans can use only CMT.

Container Managed Transactions

- Declare the transactional attributes and let the container worry about it.
- Works at the 'method' level.
- i.e. The method 'transferFunds()' might be made transactional, but 'inquireBalance()' might not be.
- Requires scope attribute: Required, RequiresNew, NotSupported, Supports, Mandatory, Never

Bean Managed Transactions

- Can be JDBC or JTA
- Java Transaction API (JTA) is part of J2EE, allows you to work with multiple databases from different vendors.

'Boss your way to EJB!

- You can learn more about Passivation by putting diagnostics in the `ejbPassivate()` and `ejbActivate()` methods.
- You can learn more Transactions by putting diagnostics in `afterBegin()`, `beforeCompletion()`, and `afterCompletion()`.
- Also `PrePersist()`, `PostPersist()`, `PreRemove()`, `PostRemove()`, `PreUpdate()`, `PostUpdate()`, `PostLoad()`

“Boss your way to EJB, special bonus page!

- Use the JBoss ‘TrailBlazer’ series.
- They’ll walk you through various flavors, all in an interactive manner.
- Web based, free training.
- Source code, build scripts, etc. all nicely bundled for you to download.



Legacy Connectivity

- Data-level integration
- Application (business)/presentation-level integration
- B2B integration

Data-level Integration

- Use of JDBC to read/write data shared with other applications.

Application (business)/presentation-level integration

- JMS (MOM)
- Java IDL (CORBA)
- SOAP
- JCA (can be transactional)
- JNI (call your other-language code and be called by it)
- Object mapping tools (wrap other-environment objects with Java)

Application (business)/presentation-level integration, continued

- Off-board server (acts as a proxy for legacy system, talks to it with legacy protocols)
- Screen scraper (Intercepts character based streams, acts as a proxy)

B2B Integration

- Spoke model (Someone else owns hub)
- Exchange model (clearinghouse)
- Hub model (You own the hub)

Protocols

- HTTP
- HTTPS
- IIOP
- JRMP

Protocols HTTP

Simple to implement
Ubiquitous standard
Most firewalls don't block it

Stateless*

Insecure

*Mitigate with Cookies, URL re-writing

Protocols HTTPS

- ✓ Secure
- ✓ Can be used to maintain state
- ✓ Firewall support

- ✓ Slower, more bandwidth than HTTP
- ✓ Client code is more difficult

Protocols IIOP

- Part of CORBA
- Allows different languages to talk
- Via Java IDL or RMI/IIOP

- Not the best performance

Protocols Java Remote Method Protocol

- JRMP– see JBoss naming in swing clients
- Connection based, stateful
- Built in support for HTTP tunneling
- Simple, High performance

- Not good firewall support
- Only the naming service available (vs. CORBA)
- Works only for Java

'Boss your way to Protocols!

- Build Hello World Applications on JBoss
- Monitor them with Apache's Tunneller
- Monitor them with Ethereal – consult local network law enforcement first!



What Tunneller will show you

TCP Tunnel Monitor: Tunneling localhost:81 to localhost:8080

From localhost:81

```
GET /airline/controller HTTP/1.1
Host: localhost:81
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.6) Gecko/20080705 Firefox/2.0.0.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=80DC9CFC9E2E0C1DA0A07EF012E28C0C0
```

From localhost:8080

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet/2.4,JBoss-4.0.4RC1 (build: CVSTag=JBoss-4.0.4RC1)
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 427
Date: Mon, 20 Feb 2006 18:52:35 GMT

<html>
<h1>
Flight Search
```

Listening for connections on port 81 ...

JBoss World
LAS VEGAS

43

What Ethereal will show you

Ethereal

Time: 0.000000000

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.2.15	TCP	60	64800 → 8080 [RST] Seq=4111329444 Win=0 Len=0
2	0.000000000	10.0.2.15	10.0.2.15	TCP	60	8080 → 64800 [RST] Seq=2479728728 Win=0 Len=0
3	0.000000000	10.0.2.15	10.0.2.15	TCP	60	64800 → 8080 [RST] Seq=4111329444 Win=0 Len=0
4	0.000000000	10.0.2.15	10.0.2.15	TCP	60	8080 → 64800 [RST] Seq=2479728728 Win=0 Len=0

Packet 21: 0.000000000

```
GET /airline/controller HTTP/1.1
Host: localhost:81
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.6) Gecko/20080705 Firefox/2.0.0.12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=80DC9CFC9E2E0C1DA0A07EF012E28C0C0
```

Packet 22: 0.000000000

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet/2.4,JBoss-4.0.4RC1 (build: CVSTag=JBoss-4.0.4RC1)
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 427
Date: Mon, 20 Feb 2006 18:52:35 GMT

<html>
<h1>
Flight Search
```

JBoss World
LAS VEGAS

44

Design Patterns and UML

- Don't 'Boss your way with these – it'd be tedious
 - Study books and web sites
 - These are big parts of the exam, but don't lend themselves to study with an application server.
 - You might peruse the JBoss source code for Patterns, but you'll have better luck trolling internet sites for UML examples.
- JBoss World
LAS VEGAS
- 45

Messaging

- Synchronous and Asynchronous
 - Point-to-Point and Publish-Subscribe
- JBoss World
LAS VEGAS
- 46

Messaging - Synchronous

- Caller blocks until response comes
 - Caller knows if everything went well.
 - Message order is known.
 - Reliable.
- JBoss World
LAS VEGAS
- 47

Messaging - Asynchronous

- Caller doesn't need to wait for response.
 - Messages can be queued.
 - Sender/receiver need not always be up.
 - Loose coupling between caller / callee.
 - Uses hardware efficiently.
- JBoss World
LAS VEGAS
- 48

Messaging – Point-to-Point

- Sender puts message in queue.
- Queue is monitored by one receiver.
- Usually FIFO, but can be changed.

Messaging- Publish-Subscribe

- Publishers publish to a 'topic'.
- Multiple listeners on topic.
- Subscribers must be 'up' to get message. (Unless it's a 'durable subscription'– then it's delivered when listener comes back up.)

'Boss your way to Messaging!

- Use JBoss JMS and MDBs to learn Messaging. (Non-standard–MDPojos)
- MDBs are easy
- Use a servlet to tickle process from a browser
- Use `System.out.println` in `onMessage()` method in MDB

Internationalization

- I18n
- Means to write the program so it can render output in different languages
- Contrast to 'Localization'

Localization

- Means to adapt your application to specific localities
- Can mean text, pictures, algorithms (i.e. tax calculations)

How it's done...

- ResourceBundle – like a Hashmap
- Key is common among bundles, i.e. HELLO_MSG
- Value is specific, i.e. Hola!
- Different flavors ResourceBundle, JDK gives you:
 - ✓ PropertyResourceBundle
 - ✓ ListResourceBundle

ResourceBundle examples

- MyResource_en_US // USA text, etc.
- MyResource_fr_FR // French
- MyResource_fr_CA // Canadian style
- Contents of MyResource_en_US
Greeting="Hello World!"
Colloquial_Greeting="Hiya"
- Contents of MyResource_cs_CZ
Greeting="Nazdar Spousta!"
Colloquial_Greeting="Jak se vam dari?"

55



Getting a ResourceBundle

```
ResourceBundle.getBundle  
("MyResource", new  
Locale("fr", "CA")) ;  
// Note the 'Locale' object
```

BTW, it will search a 'path' for you if it doesn't get an immediate hit. (Probably en_US as default). It can return a MissingResourceException if nothing...

56



Getting something useful

```
ResourceBundle res;  
res = ResourceBundle.getBundle  
("MyResource", new  
Locale("en", "US")) ;  
  
String labelOK =  
(String) res.getObject("OK_TEXT");  
// Convenience methods available to  
avoid cast
```

57



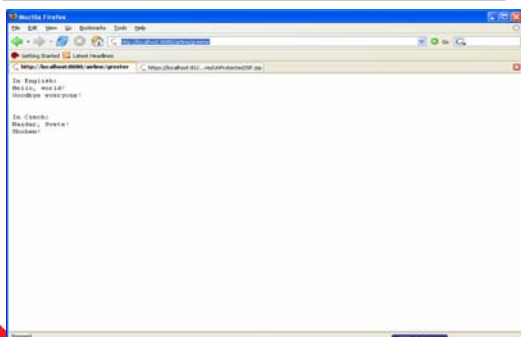
Example I18n

```
Locale locale = new Locale("en", "US");  
Locale localecz = new Locale("cs", "CZ");  
ResourceBundle resourceBundle =  
ResourceBundle.getBundle("com.flyingdog.airline.web.int  
ernationalization.MyResource", locale) ;  
  
String labelHello = (String)  
resourceBundle.getObject("HELLO_TEXT");  
String labelGoodbye =  
(String)resourceBundle.getObject("GOODBYE_TEXT");  
  
PrintWriter out = res.getWriter();  
out.println("In English:");  
out.println(labelHello);  
out.println(labelGoodbye);
```

58



Example I18n, continued



59



'Boss your way to I18n!

- Build a servlet that uses ResourceBundles
- Put your Resource files in WEB-INF/classes (or use ListRB classes)
- Change Browser preferences to see different greetings



60



Security

- Provided in layers
 - ✓ Applet security
 - ✓ JDK security
 - ✓ Web tier security
 - ✓ EJB tier security

61

Security – Fundamentals

- Principal – person, role, or system
- Authentication – the means to prove someone is who they say they are
- Authorization – the means to restrict access to specified resources
- Data Integrity – assure information has not been altered while in transit
- Confidentiality – make information accessible only to the right users

62

Security - Cryptography

- Symmetric – common key for encrypting/decrypting
- Uses less resources
- Common key must be shared. If it's compromised, mechanism is no good

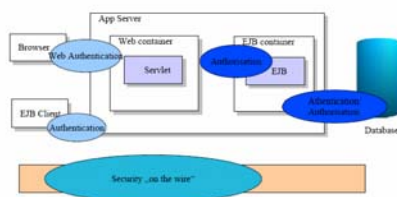
63

Security - Cryptography

- Asymmetric – Two different, but related keys are used. Messages encrypted with one key can only be read with the other.
- Eliminates need to share keys.
- Is computationally expensive

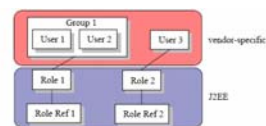
64

A picture's worth 1000 words...



65

Another thousand....



66

Security

- Applet Security
 - ✓ Can't make network connections, except back to host
 - ✓ Can't read/write file system
 - ✓ Can't start other programs, load libraries

These can be overridden!

Classes that are on classpath don't obey these.

67



Security – Web security

- Managed with Deployment Descriptor
- security-constraint tag
 - ✓ user-data-constraint / transport-guarantee for SSL
 - ✓ auth-constraint tells which roles can access the protected resource
- ✓ Different authentication methods: Basic, Form, Digest (like basic, except password is hashed)

68



Web Security snippet

```
<security-constraint>
<display-name>Restrict access to JSP pages</display-name>
<web-resource-collection>
  <web-resource-name>Protected JSPs</web-resource-name>
  <url-pattern>/secured/*</url-pattern>
</web-resource-collection>
<auth-constraint>
  <description>
    Allow PooBahs
  </description>
  <role-name>PooBah</role-name>
</auth-constraint>
</security-constraint>
```

69



Security – EJB tier

- Can be implemented by Application Server (not EJB container) for non-web clients
- Can be 'pass through' Authentication from Web Tier (common)
- Authentication done by container
- Authorization can be declarative or programmatic

70



Security - EJB

- Authorization is done at method level
- Example, with annotations (vs. descriptor)

```
@Stateless
@SecurityDomain("other")
public class SecureCalculator implements Calculator {

  @PersistenceContext
  protected EntityManager em;

  @RolesAllowed({"AdminUser"})
  public void addFund (String name, double growthrate) {
    Fund fund = new Fund (name, growthrate);
    em.persist (fund);
  }
}
```

71



JBoss Boss your way to Security!

- Protect an HTTP resource, then use Tunneller to view the traffic.
- <security-constraint>
 - <web-resource-collection>
 - <web-resource-name>Protected Page</web-resource-name>
 - <url-pattern>/ProtectedJSP.jsp</url-pattern>
 - <http-method>GET</http-method>
 - <http-method>POST</http-method>
 - </web-resource-collection>
 - <user-data-constraint>
 - <transport-guarantee>confidential</transport-guarantee>
 - </user-data-constraint>
 - </security-constraint>

72



What we see when we go to the unprotected resource

```

TCP Tunnel Monitor: Tunneling localhost:81 to localhost:8080
From localhost81                                From localhost8080
GET /airline/UnProtectedJSP.jsp HTTP/1.1        -HTTP/1.1 200 OK
Accept: image/gif, image/jpeg, image/png, application/xml
Accept-Language: en-us                         Server: Apache-Coyote/1.1
Accept-Encoding: gzip, deflate                 X-Powered-By: Servlet/2.4, JBoss/4.0.4RC1 (build: CV8Tag-JBoss_4_0_4_f
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; NETC   )
Host: localhost81                             Set-Cookie: JSESSIONID=4117FA14856F26677E50B3D062E59A3; Path=/
Connector: Keep-Alive                          Content-Type: text/html;charset=ISO-8859-1
                                               Content-Length: 50
                                               Date: Wed, 22 Feb 2006 19:24:40 GMT
                                               <html>
                                               <h1>
                                               <h1>UnProtected JSP
                                               </h1>
                                               This JSP is out in the open and naked.
                                               </html>
Listening for connections on port 81 ...
    
```

What we see when we go to the protected resource

```

TCP Tunnel Monitor: Tunneling localhost:81 to localhost:8443
From localhost81                                From localhost8443
GET /airline/ProtectedJSP.jsp HTTP/1.1        404 Not Found
Accept: image/gif, image/jpeg, image/png, application/xml
Accept-Language: en-us                         Server: Apache-Coyote/1.1
Accept-Encoding: gzip, deflate                 X-Powered-By: Servlet/2.4, JBoss/4.0.4RC1 (build: CV8Tag-JBoss_4_0_4_f
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; NETC   )
Host: localhost81                             Set-Cookie: JSESSIONID=4117FA14856F26677E50B3D062E59A3; Path=/
Connector: Keep-Alive                          Content-Type: text/html;charset=ISO-8859-1
                                               Content-Length: 50
                                               Date: Wed, 22 Feb 2006 19:24:40 GMT
                                               <html>
                                               <h1>
                                               <h1>Protected JSP
                                               </h1>
                                               This JSP is protected.
                                               </html>
Listening for connections on port 81 ...
    
```

‘Boss your way to Security! part 2

- Apply declarative security to a bean, then access it from a different clients with different credentials.

Part 2 – what’s it all about?

- You are given an application to build
- It’s somewhat ambiguous in places
- It takes about 100 hours to finish
- There are lots of ways to go about the assignment. See the SCEA discussion board at JavaRanch for some ideas.

Part 3 – what’s it about?

- A few questions about why you did things as you did in part 2.
- Many people say no study is necessary, if you’ve done a good job on your part 2 assignment.

Really, really good books to study

- Cade Study guide– by the author of the test
- Core J2EE book– especially useful for part 2