

J2EE v1.3 Portability

Magnus Larsson

Callista Enterprise AB

magnus.larsson@callista.se

<http://www.callista.se/enterprise>

J2EE v1.3 Portability - Agenda

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Why is portability important?

- ✍ It enables J2EE customers to move their applications from one vendors J2EE server to another vendors
- ✍ This brings fierce competition to the J2EE server market
- ✍ J2EE vendors must compete "head to head" with non J2EE-standard (but still portable) features
 - ✍ Performance
 - ✍ Robustness
 - ✍ Scalability (e.g. load balancing)
 - ✍ High Availability (e.g. fail over)
 - ✍ Ease of use (e.g. documentation)
 - ✍ Tools integration (e.g. deploy from an IDE)
 - ✍ Licensing cost (e.g. low priced "entry editions")
- ✍ Gives J2EE a competitive advantage to "other technologies"

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ **Requirements on J2EE portability**
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Requirements on J2EE portability

- ✍ It must be possible to deploy a J2EE application to any J2EE certified server without
 - ✍ Performing any code changes to the source code
 - ✍ Performing any changes to the J2EE deployment descriptors
 - ✍ Precondition: The application is J2EE spec compliant
 - "Well behaved" application code
See chapter 6 "Application Programming Interface"
 - Assembly done according to the EAR-format
See chapter 8 "Application Assembly and Deployment"

Note: Chapter numbers refer to the "J2EE v1.3 spec."

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ **Test Scope**
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Test Scope

✍ Scope

- ✍ Move a J2EE v1.3 application from one J2EE server to other J2EE servers
- ✍ Test deployment of an application that uses most of the J2EE v1.3 features.
 - Java PetStore v1.3-01 is a good candidate for this...
- ✍ Perform single-user functionality-tests on the deployed application per J2EE server

✍ Non-scope

- ✍ Testing production characteristics, e.g.
 - Performance and Robustness
 - Scalability and High Availability

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ **Introduction to news in J2EE 1.3** and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Introduction to news in J2EE v1.3

✍ Connector Architecture 1.0

✍ Servlets v2.3

✍ Filters

✍ Listeners

✍ Security

✍ JAAS 1.0

✍ **EJB 2.0**

✍ **MDB**

✍ **CMP 2.0**

✍ **CMR**

✍ **EBJ QL**

✍

Message Driven Beans

Container Managed Persistence

Container Managed Relationships

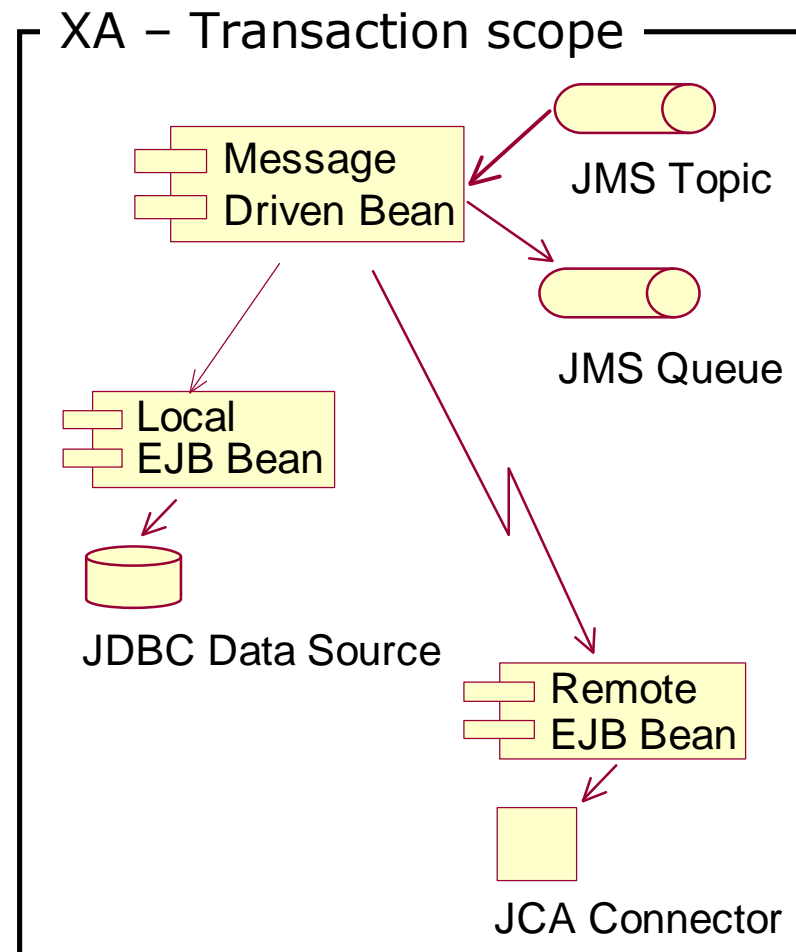
EJB Query Language

Local interfaces

News in J2EE v1.3 / EJB 2.0

Message Driven Beans (MDB)

- ✍ Enables the EJB Container to accept asynchronous calls over JMS
- ✍ No home nor remote interface
- ✍ Must implement two interfaces
 - ✍ `javax.ejb.MessageDrivenBean`
 - `ejbCreate()`
 - `ejbRemove()`
 - ✍ `javax.jms.MessageListener`
 - `onMessage()`
- ✍ Stateless by nature
- ✍ Can be XA aware



News in J2EE v1.3 / EJB 2.0 Container Managed Persistence (CMP 2.0)

- ✍ Much improved over CMP in EJB 1.1
 - ✍ Defines an abstract model
 - Enables high performing CMP implementations
 - ✍ Unknown Primary Key Class
 - The container handles technical keys

```
<prim-key-class>java.lang.Object</prim-key-class>
```

- ✍ Foundation for CMR and EJB QL (next slides)
- ✍ Preferably used together with Local Interface (next slides)

News in J2EE v1.3 / EJB 2.0

Container Managed Persistence (CMP 2.0)

Example

From the Entity bean class:

```
public abstract class CustomerEJB implements EntityBean {  
  
    // getters and setters for CMP fields  
    public abstract String getUserId();  
    public abstract void setUserId(String userId);  
}
```

From the ejb-jar.xml – file:

```
<entity>  
  <ejb-name>CustomerEJB</ejb-name>  
  <persistence-type>Container</persistence-type>  
  <cmp-version>2.x</cmp-version>  
  <abstract-schema-name>Customer</abstract-schema-name>  
  <cmp-field>  
    <field-name>userId</field-name>  
  </cmp-field>  
</entity>
```

News in J2EE v1.3 / EJB 2.0

Container Managed Relationships (CMR)

✍ Express relationships between Entity beans

✍ Cardinalities

- One-to-One
- One-to-Many
- Many-to-Many

✍ Navigability

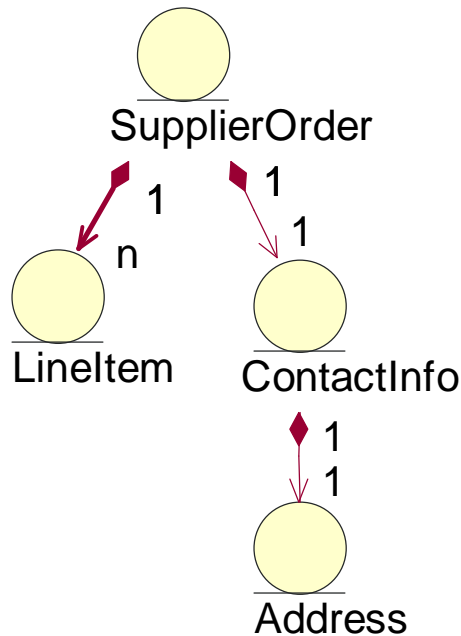
- Unidirectional
- Bi-directional

✍ Cascade Delete (Aggregates)

News in J2EE v1.3 / EJB 2.0

Container Managed Relationships (CMR)

From the ejb-jar.xml – file:



```

<ejb-relation>
  <ejb-relationship-role>
    <multiplicity>One</multiplicity>
    <relationship-role-source>
      <ejb-name>SupplierOrderEJB</ejb-name>
    </relationship-role-source>
    <cmr-field>
      <cmr-field-name>lineItems</cmr-field-name>
      <cmr-field-type>Collection</cmr-field-type>
    </cmr-field>
  </ejb-relationship-role>
  <ejb-relationship-role>
    <multiplicity>Many</multiplicity>
    <cascade-delete/>
    <relationship-role-source>
      <ejb-name>LineItemEJB</ejb-name>
    </relationship-role-source>
  </ejb-relationship-role>
</ejb-relation>
  
```

No cmr-fields for this role makes the relation unidirectional

News in J2EE v1.3 / EJB 2.0 EJB Query Language (EJB QL)

- ✍ A query language similar to SQL
 - ✍ operates on the abstract CMP/CMR - entity model

- ✍ Query methods
 - ✍ Public finder methods
 - Must return instances of the Entity Bean

 - ✍ Private select methods
 - Can return any type of Entity Bean or a single CMP field

News in J2EE v1.3 / EJB 2.0 EJB Query Language (EJB QL)

✍ To realize a public EJB QL based find-method

✍ Add a method-declaration to the home-interface

```
public Collection findOrdersByState(  
    String state, String status  
) throws FinderException;
```

✍ Define the EJB QL query in the ejb-jar.xml file

```
<ejb-ql>  
    SELECT DISTINCT OBJECT(o)  
    FROM    SupplierOrder o, IN(o.lineItems) l  
    WHERE  o.contactInfo.address.state = ?1  
    AND    l.lineItemStatus = ?2  
</ejb-ql>
```

News in J2EE v1.3 / EJB 2.0

EJB Query Language (EJB QL)

✍ Automatic EJB QL ✍ SQL translation

✍ EJB QL query

```
<ejb-ql>
  SELECT DISTINCT OBJECT(o)
  FROM   SupplierOrder o, IN(o.lineItems) l
  WHERE  o.contactInfo.address.state = ?1
  AND    l.lineItemStatus = ?2
</ejb-ql>
```

✍ SQL (WebLogic 7.0 and Oracle 8.1.7)

```
SELECT WL0.poId, WL0.poDate, WL0.poStatus
FROM   SupplierOrder WL0, contactInfo WL1,
       address        WL2, lineItem      WL3
WHERE  (((WL2.state = :1)) AND ((WL3.lineItemStatus = :2)))
AND    WL1.fk_purchaseorder = WL0.poId
AND    WL1.fk_address        = WL2.id
AND    WL3.fk_purchaseorder = WL0.poId
```

News in J2EE v1.3 / EJB 2.0

Local Interfaces

✍ The problem

- ✍ In many cases are EJB-beans called from colocated EJB-beans or JSP/Servlets
 - The implicit RMI-overhead in the EJB remote calls is very high
 - All J2EE v1.2 vendors provide a vendor specific "big white switch" that removes the RMI-overhead for colocated calls

✍ The solution

✍ Local Interfaces

- A standardized way of avoiding RMI-overhead for colocated calls

✍ Defined in parallel with home and remote interfaces

- `javax.ejb.EJBLocalHome` complements `javax.ejb.EJBHome`
- `javax.ejb.EJBLocalObject` complements `javax.ejb.EJBObject`

News in J2EE v1.3 / EJB 2.0 Local Interfaces

✍ Defined in the ejb-jar.xml file as usual

```
<entity>
  <ejb-name>CustomerEJB</ejb-name>
  <home>...CustomerHome</home>
  <remote>...CustomerRemote</remote>
  <local-home>...CustomerLocalHome</local-home>
  <local>...CustomerLocal</local>
```

✍ JNDI names are given for both the local-home and remote-home i/f

✍ No RMI also means

- ✍ No RemoteExceptions
- ✍ No PortableRemoteObject.narrow() calls
- ✍ No protecting "By Value" semantics
 - Uses "By Reference"

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ **Introduction to** news in J2EE 1.3 and **Java PetStore 1.3-01**
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Introduction to Java PetStore 1.3-01

✍ Deployable on Sun J2EE Reference Implementation v1.3.1

✍ J2EE v1.3 coverage

✍ Uses most of the J2EE v1.3 features except for

- The Connector Architecture
- JAAS based security

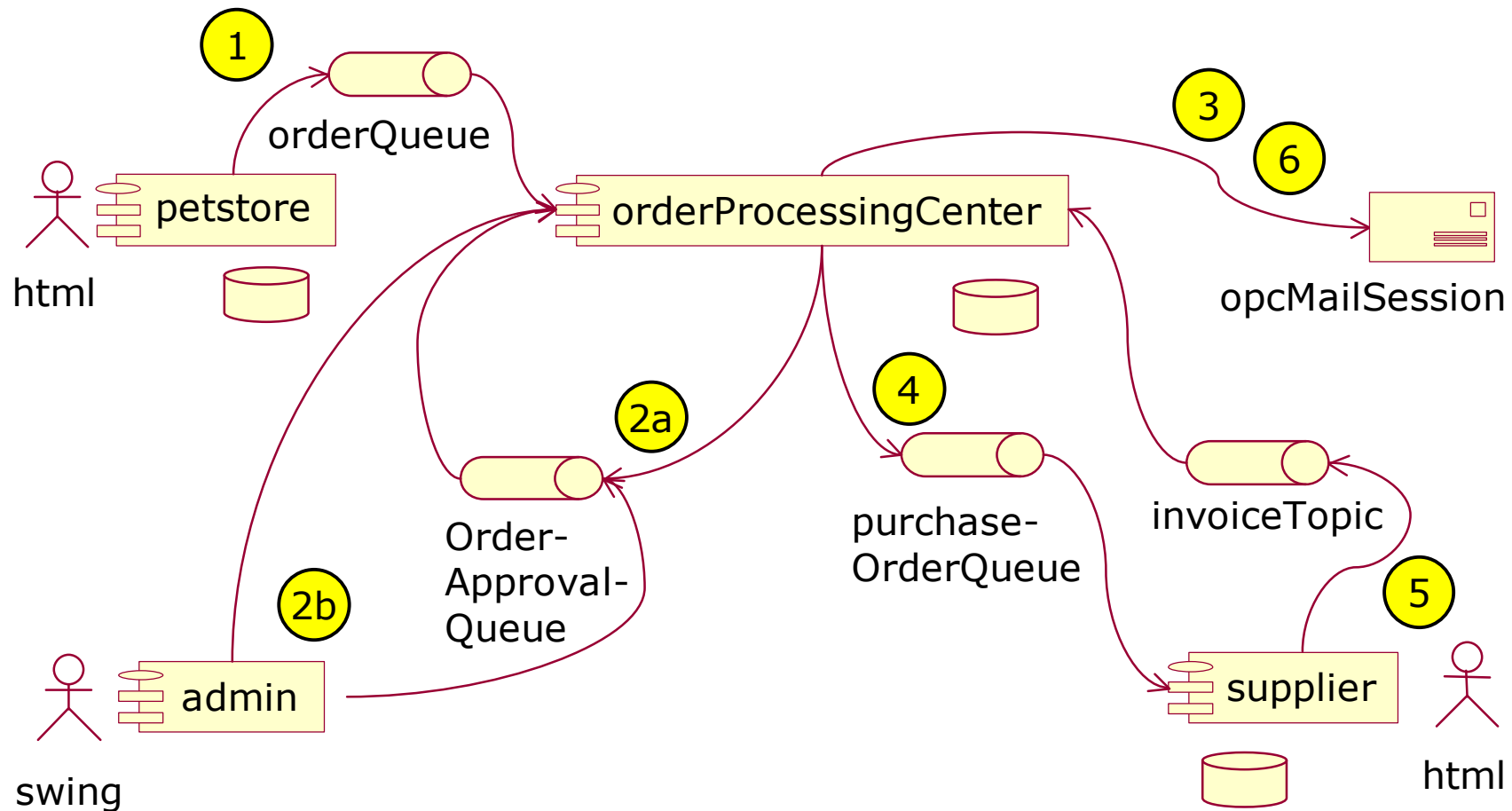
✍ Functional overview

✍ Four J2EE applications communicating over JMS

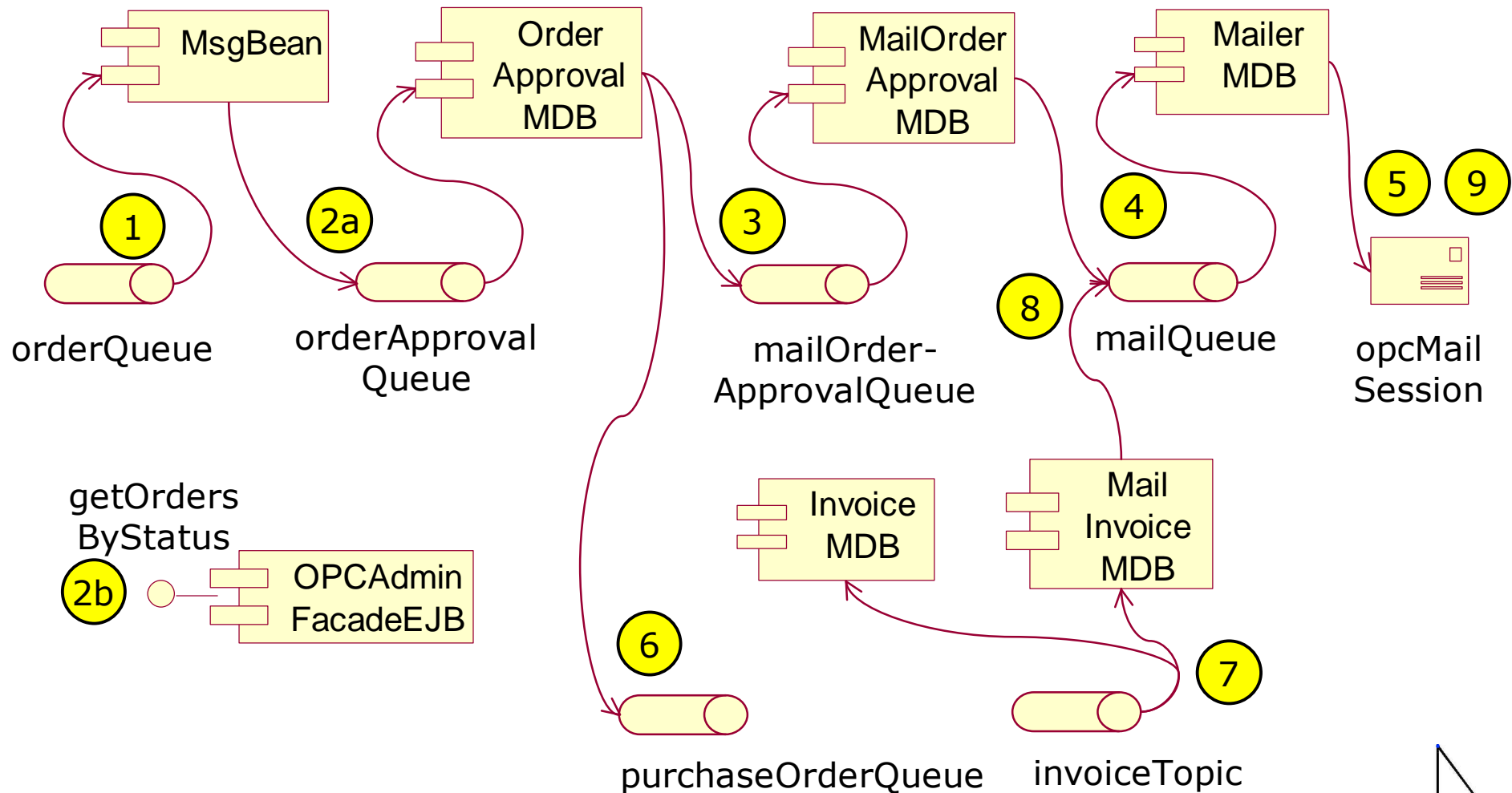
✍ Both Web and Java based GUI's

Introduction to Java PetStore 1.3-01

System Integration flows



Introduction to Java PetStore 1.3-01 orderProcessingCenter internal flows



Introduction to Java PetStore 1.3-01

✍ Components

- ✍ 4 EAR-files
- ✍ 2 Web applications
- ✍ 1 Java client application
- ✍ 13 EJB modules
- ✍ 33 EJB beans
 - 6 Stateless Session Beans
 - 3 State full Session Beans
 - 17 CMP Entity Beans
 - 7 Message Driven Beans

✍ J2EE Resources

- ✍ 3 JDBC XA Data Sources
- ✍ 4 JMS XA Connection Factories
- ✍ 5 JMS Queues
- ✍ 1 JMS Topic
- ✍ 1 Mail Session
- ✍ 2 User Groups

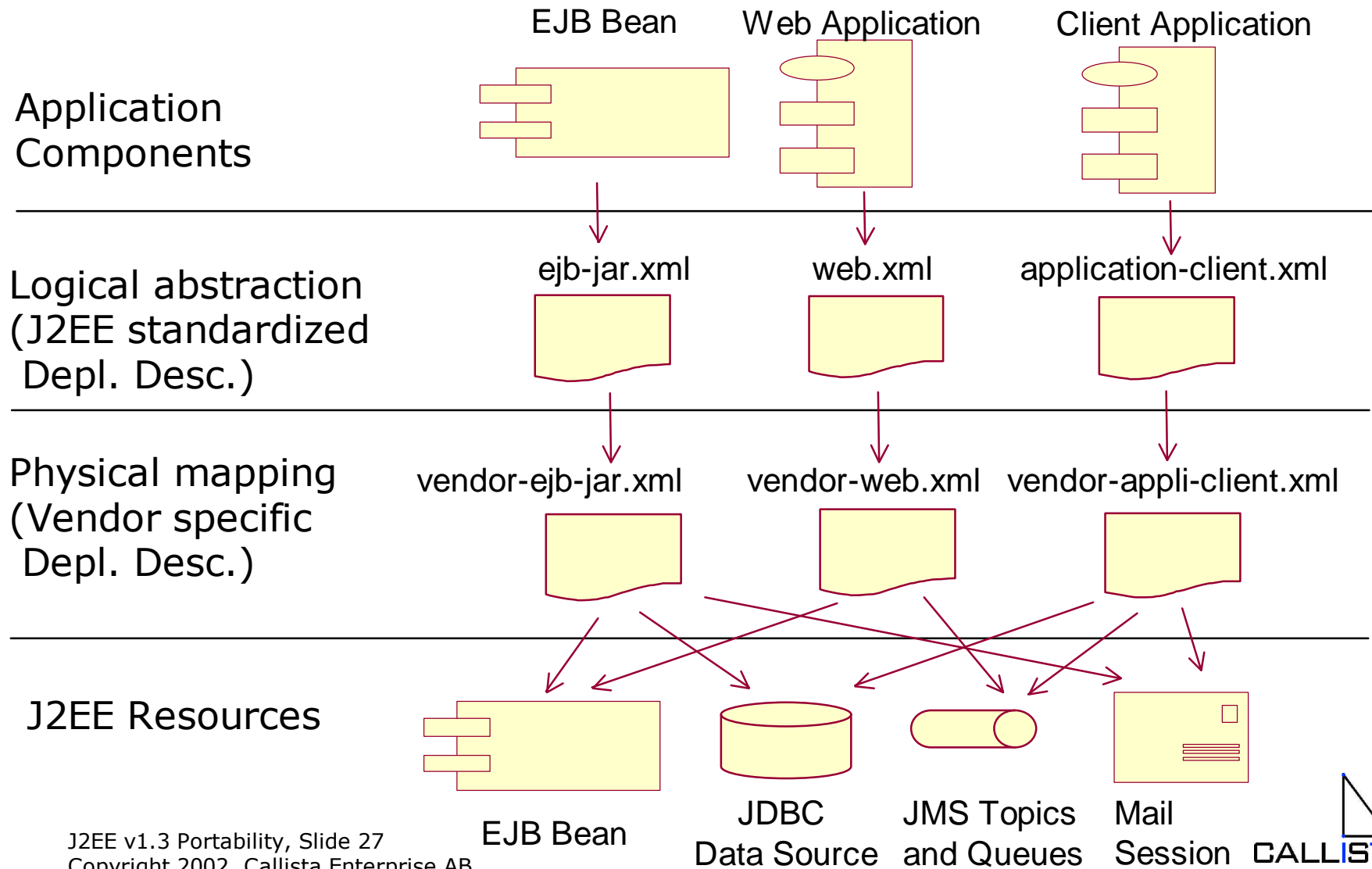
J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ **J2EE portability – theory**
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

J2EE Portability - Theory

- ✍ Porting an J2EE application to a J2EE server
 - ✍ Create required J2EE resources in the J2EE server
 - JDBC Data Sources
 - JMS Connection Factories, Queues and Topics
 - Mail-sessions
 - ✍ Create J2EE server specific deployment descriptors
 - Maps logical ejb- and resource-references used by the J2EE application to the physical JNDI-names used on this server
- ✍ Generate J2EE server specific deployment code
- ✍ Deploy the J2EE application

J2EE Portability - Theory



J2EE Portability - Theory

- ✍ What to map in the vendor specific deployment descriptors
 - ✍ JDBC DataSources
 - ✍ JMS Connection Factories, Queues and Topics
 - ✍ Mail-sessions
 - ✍ Security Roles to Users and User Groups
 - ✍ CMP mappings
 - JDBC DataSource
 - Table and Column names
 - ✍ CMR mappings
 - Primary and Foreign keys

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ **Test environment**
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Test environment

✍ J2EE v1.3 Servers

- ✍ BEA WebLogic Server v7.0 beta
- ✍ IBM WebSphere Technology Preview (v5.0...)
- ✍ Oracle OC4J v9.0.3 "1.3 Developer Preview"
 - Oracle JMS (AQ) as JMS Resource Provider

✍ Databases

- ✍ Oracle v8.1.7
 - BEA WebLogic and Oracle OC4J
- ✍ IBM Db2 v7.2
 - IBM WebSphere

✍ Operating Systems

- ✍ Windows 2000

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ **Demo**
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

Demo

- ✍ Demonstrate that Java Pet Store runs on all J2EE servers
- ✍ Perform a code change
- ✍ Auto compile, assemble, redeploy and test
 - ✍ and have a cup of coffee...
- ✍ Demonstrate that the change was applied on all J2EE servers...

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ **J2EE portability – reality**
- ✍ Findings & Recommendations
- ✍ How to get started on your own...

J2EE Portability - Reality

Test setup

- ✍ Vendor specific deployment descriptors
 - ✍ Howto create
 - ✍ Filenames
 - ✍ Adding to the EAR-files
- ✍ Adopting Cloudscape SQL to Db2 and Oracle
- ✍ Minimizing the number of ear- and jar-files
- ✍ Deferring generation of deploy code until deploy time
- ✍ Pre-compiling JSP-pages
- ✍ Automating using Ant
- ✍ Auto-testing the redeployed applications

Creating vendor specific deployment descriptors

- ✍ Each vendor provides tools to generate initial DD's
 - ✍ BEA WebLogic
 - `java weblogic.ant.taskdefs.war.DDInit`
 - `java weblogic.ant.taskdefs.ejb.DDInit`
 - ✍ IBM WebSphere
 - Predeploy-tool adds DD's to a complete ear-file
 - ✍ Oracle OC4J
 - Perform a deploy without OC4J DD's and OC4J will create default OC4J DD's at
`$(OC4J_HOME)/application-deployments/$(APP_NAME)`
- ✍ Copy the initial DD's and edit them as you please
 - ✍ The default mappings gives you good hints on how to write the correct mappings.
 - ✍ The documentation also in general describes very well what to write in the DD's.

Vendor specific deployment descriptors filenames

✍ EAR-file

- ✍ META-INF/application.xml
- ✍ META-INF/weblogic-application.xml
- ✍ META-INF/ibm-application-ext.xmi
- ✍ META-INF/orion-application.xml

✍ EJB module META-INF

- ✍ META-INF/ejb-jar.xml
- ✍ META-INF/weblogic-ejb-jar.xml
- ✍ META-INF/weblogic-cmp-rdbms-jar.xml
 - Only if cmp-entity-beans exist in the EJB module
- ✍ META-INF/ibm-ejb-jar-bnd.xmi
- ✍ META-INF/orion-ejb-jar.xml

Vendor specific deployment descriptors filenames

✍ Web application

- ✍ WEB-INF/web.xml
- ✍ WEB-INF/weblogic.xml
- ✍ WEB-INF/ibm-web-bnd.xmi
- ✍ WEB-INF/orion-web.xml

✍ Client application

- ✍ META-INF/application-client.xml
- ✍ META-INF/client-application.runtime.xml
- ✍ META-INF/orion-application-client.xml

Adding vendor specific deployment descriptors to the EAR-files

✍ Done in three steps

✍ Unpack the J2EE modules from the EAR-files

- `jar xf $(EAR_FILE)`

✍ Update the J2EE-modules with the vendors DD's

- `jar -uf $(EJB_FILE)`

- `jar -uf $(WAR_FILE)`

✍ Repack the EAR-files

- `jar -cfM $(EAR_FILE)`

✍ Update the EAR-files with the vendors DD's

- `jar -uf $(EAR_FILE)`

More on Test Setup

- ✍ Adopting Cloudscape SQL to Db2 and Oracle
 - ✍ Converted the Cloudscape SQL DDL
 - ✍ Converted the Cloudscape CatalogDAO Java Class
- ✍ Minimizing the number of ear- and jar-files
 - ✍ Tried to maximize reuse, i.e. tried to avoid to create vendor specific ear- and jar-files
 - ✍ Deferring generation of deploy code until deploy time
 - ✍ Probably not the right decision in a "production environment"
- ✍ Pre-compiling JSP-pages
 - ✍ So that you don't need to sit and wait for it to happen when you start to test the redeployment...

Automating using Ant

```
<project name="CallistaDeploy" default="all" basedir=".">

    <target name="all" depends="build, deploy, test"/>

    <target name="init">
        <property file="build.properties"/>
    </target>

    <target name="build" depends="init">
        <ant dir="../src" target="core"/>
    </target>

    <target name="deploy" depends="init">
        <ant antfile="deploy.xml" dir="src/ant"/>
    </target>

    <target name="test" depends="init">
        <ant antfile="test.xml" dir="src/ant"/>
    </target>

</project>
```

Auto-testing the redeployed applications

- ✍ Declarative auto-testing with a Ant based tool "Canoo Web Test"
- ✍ Based on HTTPUnit and JUnit

```
<target name="WebLogicTest">
  <testSpec name="Test WebLogic">
    <config host="localhost" port="7001"
      protocol="http" basepath="petstore"/>
    <steps>
      <invoke url="main.screen"/>
      <clicklink label="Birds"/>
      <clicklink label="Finch"/>
      <clicklink label="Add to Cart"/>
      <clicklink label="Proceed to Checkout"/>
      <clickbutton label="Sign In"/>
      <clickbutton label="Submit"/>
    </steps>
  </testSpec>
</target>
```

- ✍ No verify-steps for now, done by looking into the mailbox...

J2EE Portability – Reality Experiences

✍ Problems...

✍ In general...

✍ In Java Pet Store source code

✍ In BEA WebLogic

✍ In IBM WebSphere

✍ In Oracle OC4J

J2EE Portability – Reality Experiences – In general...

- ✍ Classloader related problems
 - ✍ Classloader hierarchies seems to be slightly differently implemented by all J2EE vendors
 - For WebLogic and WebSphere see <http://www.theserverside.com/resources/article.jsp?l=ClassLoading>
 - For OC4J see <http://kb.atlassian.com/content/atlassian/howto/classloaders.jsp>
- ✍ Problem with loading resources (DTD's) from jar's and from the file system...
 - ✍ Placed DTD's on a Web Server as a workaround
- ✍ Usage of hot-redeploy
 - ✍ It's cool but not 100% reliable, i.e. not fully suitable for scripting...

J2EE Portability – Reality Experiences – In Java Pet Store source code

- ✍ JPS: Custom JSP Tags not reusable
 - ✍ Member variables was not reset at endTag() calls and therefore was the state invalid when the tag-instance was reused
- ✍ JPS: JSP-pages caused compile errors on custom tag-usage
 - ✍ Tags seemed not to be used so they were removed...
- ✍ JPS: Class used by custom JSP tags missing a default constructor
 - ✍ Was added...

J2EE Portability – Reality Experiences – In Java Pet Store source code

- ✍ JPS: Calls `getPrimaryKey()` from the `ejbCreate()` method
 - ✍ Not allowed. Should be done in `ejbPostCreate()` instead
 - ✍ See §10.5.5 in the EJB2.0 spec...
- ✍ JPS: Does not close JDBC Connections
 - ✍ Missing finally clauses that ensure that the JDBC resources are guaranteed to be released
- ✍ JPS: Incorrect usage of JDBC ResultSet type `SCROLL_INSENSITIVE`
 - ✍ The type `TYPE_FORWARD_ONLY` is the only required (See §6.2.4.3 in the J2EE spec.)
- ✍ JPS: `MANIFEST.MF` in lowercase, should be in uppercase

J2EE Portability – Reality Experiences – BEA WebLogic

- ✍ WebLogic: Unknown primary key class not supported
 - ✍ See §10.8.2 in the EJB 2.0 spec.
 - ✍ Workaround described in the WebLogic doc.
 - ✍ Forces separate ejb-jar.xml-files and ejb modules and ear-files...

- ✍ WebLogic: Failed to generate deployment code at deployment time
 - ✍ Due to dependencies in the MANIFEST.MF files
 - ✍ Separate EJBC-call with correct classpath as a workaround
 - ✍ Forces separate ejb modules and ear-files...

J2EE Portability – Reality Experiences – IBM WebSphere

- ✍ WebSphere don't support CMP fields of type `java.util.Date`
 - ✍ Used `java.sql.Date` instead as a workaround
 - ✍ Typical beta related problem...
- ✍ WebSphere requires ID-attributes in all J2EE DD's to map them into the WebSphere specific DD's
 - ✍ No problem, but we need to change the J2EE DD's...
`<ejb-local-ref>`
goes
`<ejb-local-ref id="EJBLocalRef_1">`
 - ✍ Done automatically by the pre-deploy tool

J2EE Portability – Reality Experiences – Oracle OC4J

- ✍ OC4J: Bug when calling getters and setters on the dependent entity object in a 1:1 relationship
 - ✍ Well documented as a “known issue” in the OC4J doc
 - ✍ Java Pet Store source code was rewritten to avoid these types of scenarios
- ✍ Oracle JMS: Seems to “take some” time to learn for a J2EE developer...

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ **Findings & Recommendations**
- ✍ How to get started on your own...

Findings & Recommendations

✍ Findings

- ✍ Portability in J2EE v1.3 actually works!!!
 - Only some minor deviations from the portability requirements

- ✍ Not "dead easy" to write a fully J2EE compliant application
 - J2EE verifiers don't help you all the way...

- ✍ Some immaturity in the J2EE v1.3 products
 - Nothing else is however to expect from beta products

Findings & Recommendations

✍ Recommendations

- ✍ Promote portability in your J2EE applications!!!
- ✍ Use automated testing procedures to verify the portability on every build
- ✍ Use the new CMP 2.0 features in J2EE v1.3 with care for the near time...
- ✍ Keep an eye on "JSR 88 2EE Application Deployment"
 - Expected to be part of J2EE v1.4
 - <http://www.jcp.org/jsr/detail/88.jsp>

J2EE v1.3 Portability – Where are we?

- ✍ Why is portability important?
- ✍ Requirements on J2EE portability
- ✍ Test Scope
- ✍ Introduction to news in J2EE 1.3 and Java PetStore 1.3-01
- ✍ J2EE portability – theory
- ✍ Test environment
- ✍ Demo
- ✍ J2EE portability – reality
- ✍ Findings & Recommendations
- ✍ **How to get started on your own...**

How to get started on your own...

- ✍ Download J2EE tools and servers and try them out
 - ✍ J2EE software is in general free to download for evaluation or development usage!
- ✍ Expect a steep learning curve for most J2EE servers
 - ✍ Jumpstart by looking at
 - Downloadable scripts and doc. on how to setup the tests demonstrated today
 - <http://www.callista.se/enterprise/resources/>
 - Available soon... (Q2 2002)
 - 2002 J2EE™ Deployathon Online!
 - <http://developer.java.sun.com/developer/technicalArticles/J2EE/deployathon3/#jslist13>