Different classes may make the interface abstraction concrete

in different ways, but they still have the abstraction that is captured by the interface in

common. That Java has a way of expressing what these different implementations have

in common is a strength of the language, not a weakness

you can trust that

the other programmers on your team are using the same interface definition because

all the classes that implement a particular interface will share a common definition

Ejb

Client --- ejb object---- services----enterprise bean----db

Client never talks directly to bean

Services is wat u pay for

3 types of beans

Entity bean- represents row in table

Msgdrienbean – only if u need JMS consumer, clients must send a msg to messaging service to call this bean….no ejb object here..

Entity bean are nouns….session beans are verbs….like process….

Session bean…..stateful………stateless

Stateless beans are more scalable than stateful …but user stateful if situation arises

Class advicebean implements sessionbean

Interface Advice extends ejbobject

Interface advicehome extends ejbhome

Legally the bean class can implement the componenet interface, but spec doesnot recommend it.

Return types, arguents…must be primitives, serializable objects, array or collection of serialization objects or a remote object

In ordiary local method calls java passes the object reference by value, the object itself is never passed.

For remote calls java passes the object copy not referece copy

All the collection implementations in j2se api are serialzable.

Don’t rely on collection returned by map’s values() method to be serializable.

With collections and objects, if any of the objects inside are’nt serializable, the whole serialization fials

If parent is serializable ..then child too. Pg 77

Wen using remote method calls, make sure the arg type class exist, else it doesnot getdeserialized.

Plain rmi

Client—biz interface—stub---biz interface—remote object---db

Ejb rmi

Client—biz interface—stub---biz interface—remote object which is ejbobject –services---enterprise bean---db

Interface hierarchy

Remote🡨----ejbobject🡨cust remote component interface

Enterprisebean🡨 session bean 🡨 bean class

Who writes wat

stub---biz interface—remote object which is ejbobject –services---enterprise bean---db

con you container you

home object makes the ejbobject and sends back the stub.

There is only one home object on server and each client will get his own dedicated bean and ejbobject

Msg driven beans don’t have (remote or local) that expose methods to the client

Client ----JMS----services------container get one of msgdriven bean out of pool and call event handler

Isequals() compares on same heap….isidentical compares on remote objects on server

Beaness has the privileges like the ability to get security info abou the client, look up the special deployment properties in jndi, get jdbc conection from pool managed by container

By callind create() method , and object converts to bean,

A bean will get reference to

Session context

Jndi context

Resource manager like database

Not all kinds of beans have all of these privileges.

Stateful beans- requested by client- have sec info

Stateless beans- created by container- no sec into

Bean managed transaction

Your bean class must have 3 types of methods,

Stuff form home

Business methods from component interface

4 session bean methods

Ejb persistence

Trade o when choosing between using a home finder method that return ejb object reference (especially if the reference are remote ) vs a home business methods that returns value objects

It’s the container’s job to keep etity and bea in synch