



Architecture des Systèmes d' Information

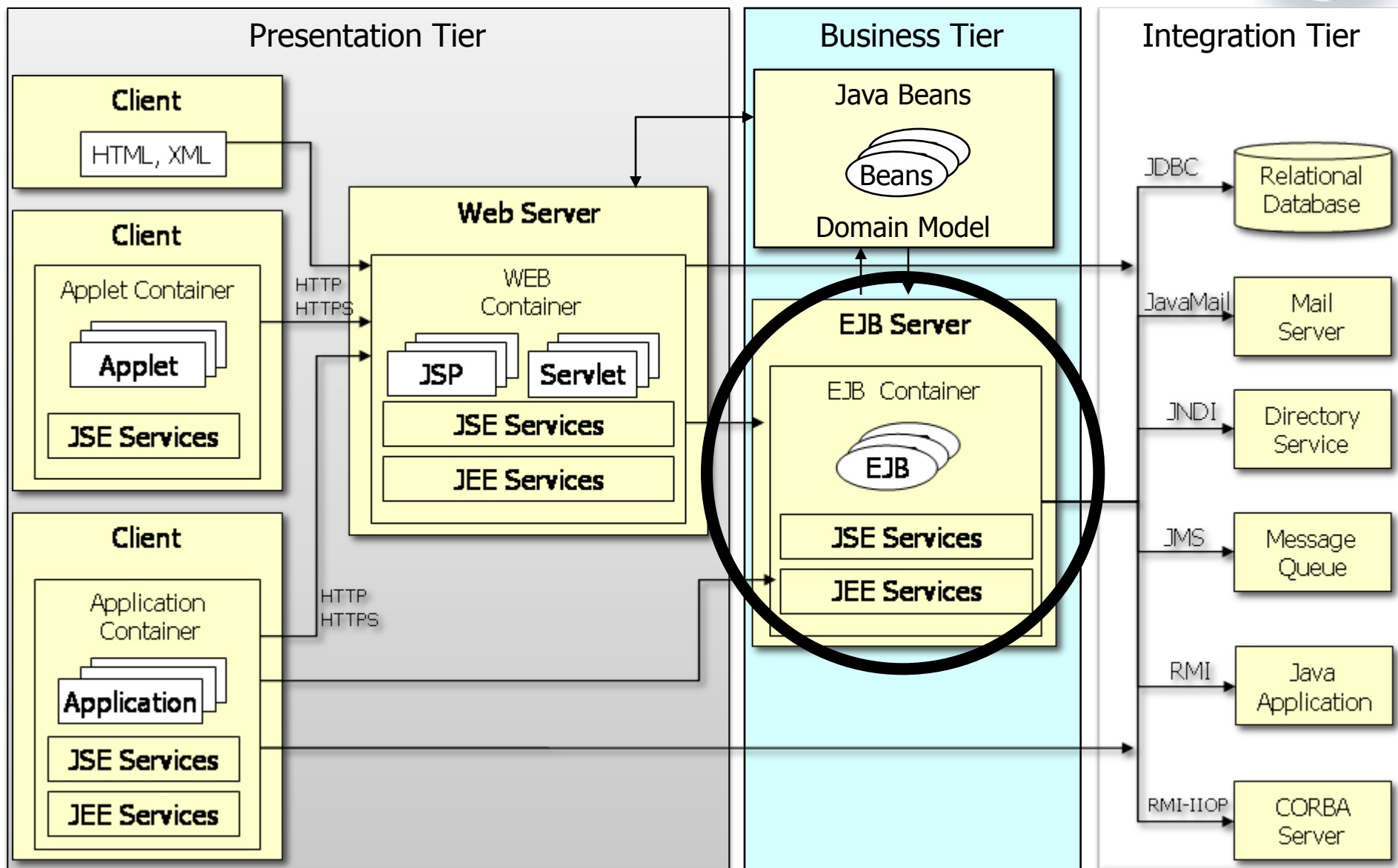
EJB interfaces



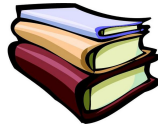
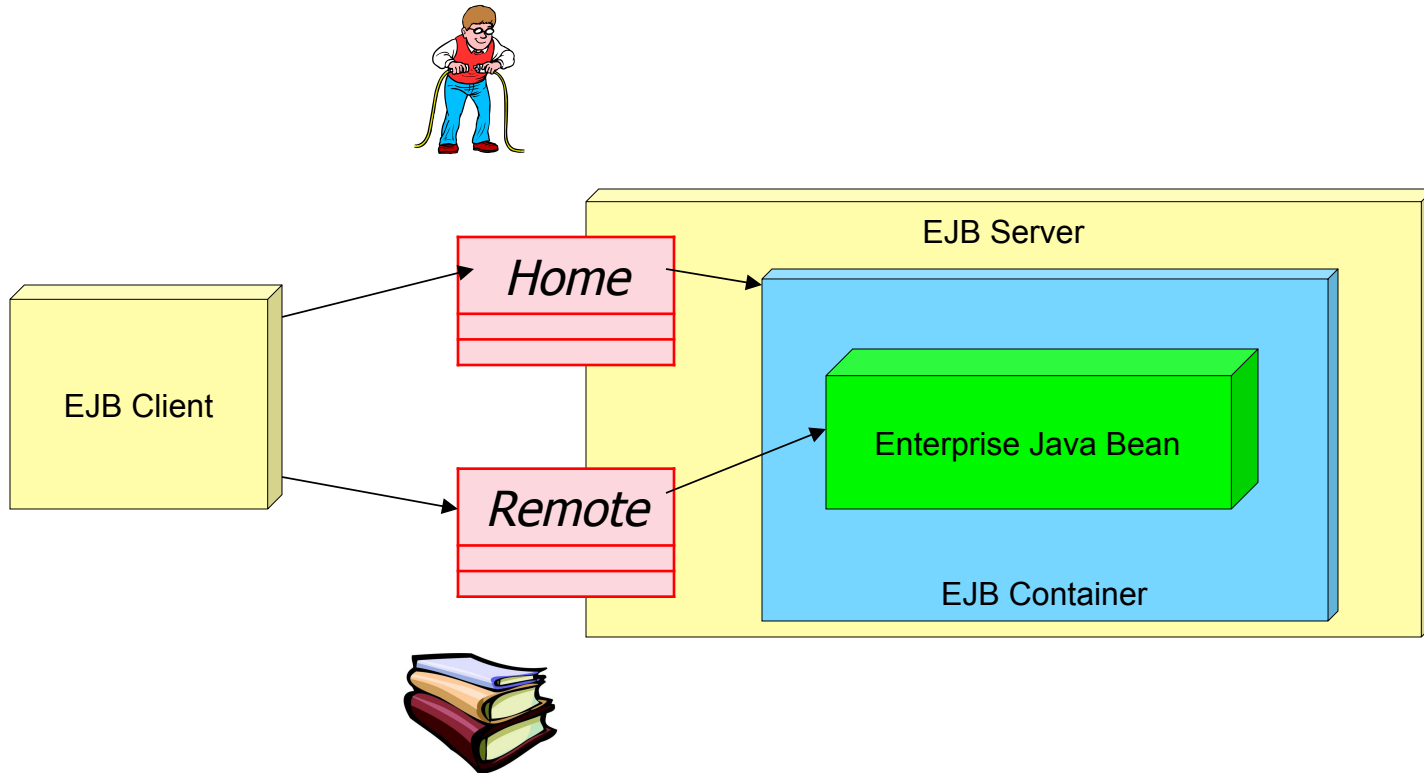
Traduction en cours



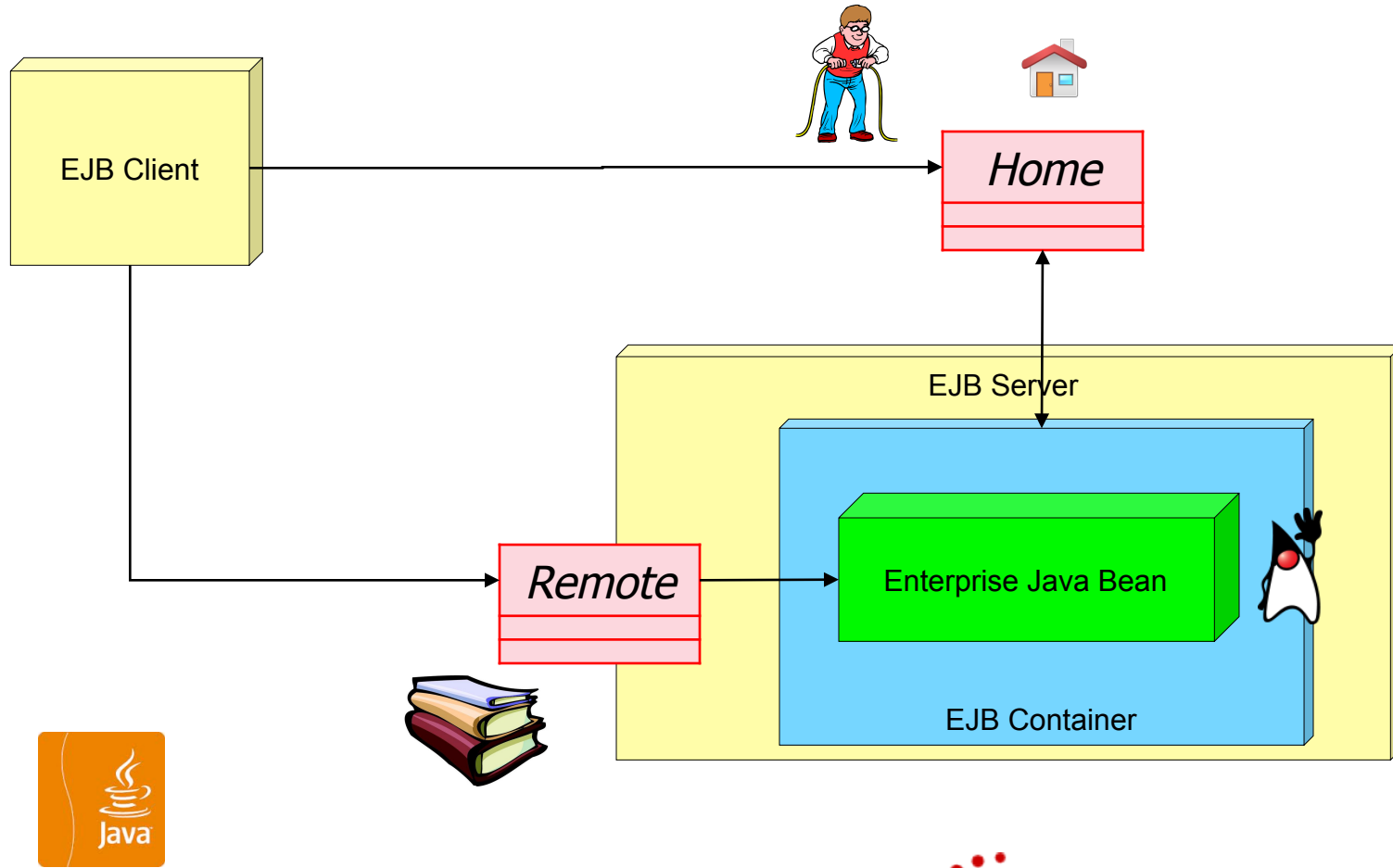
JEE JSE and EJB



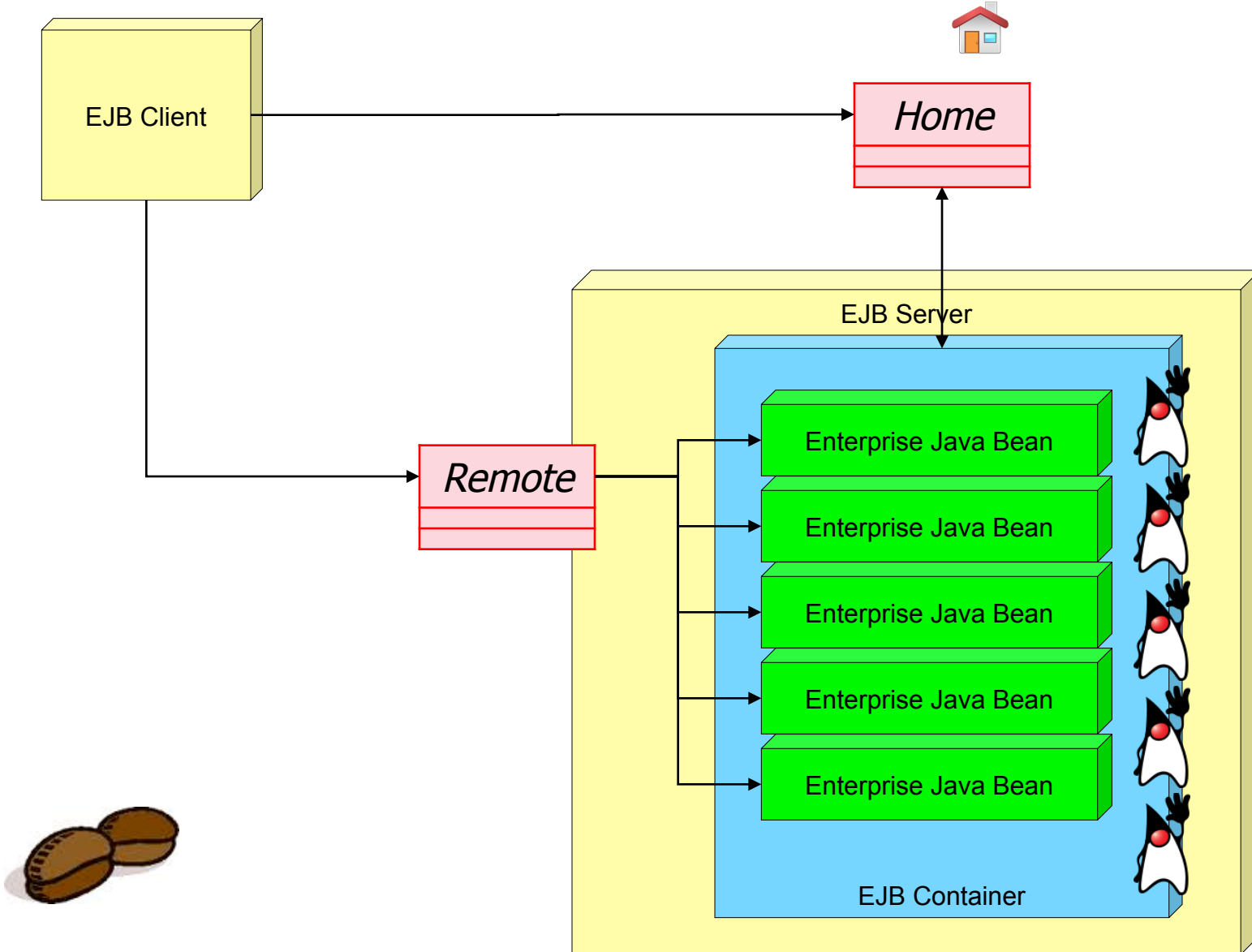
EJB IDL : Java Interface



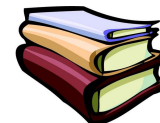
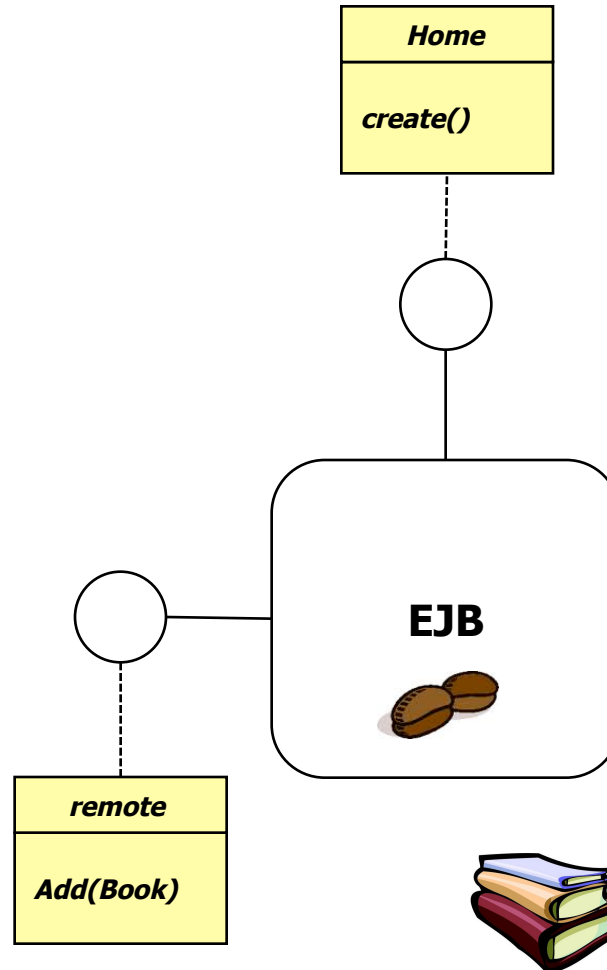
EJB IDL : Java Interface



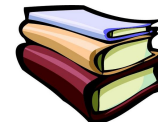
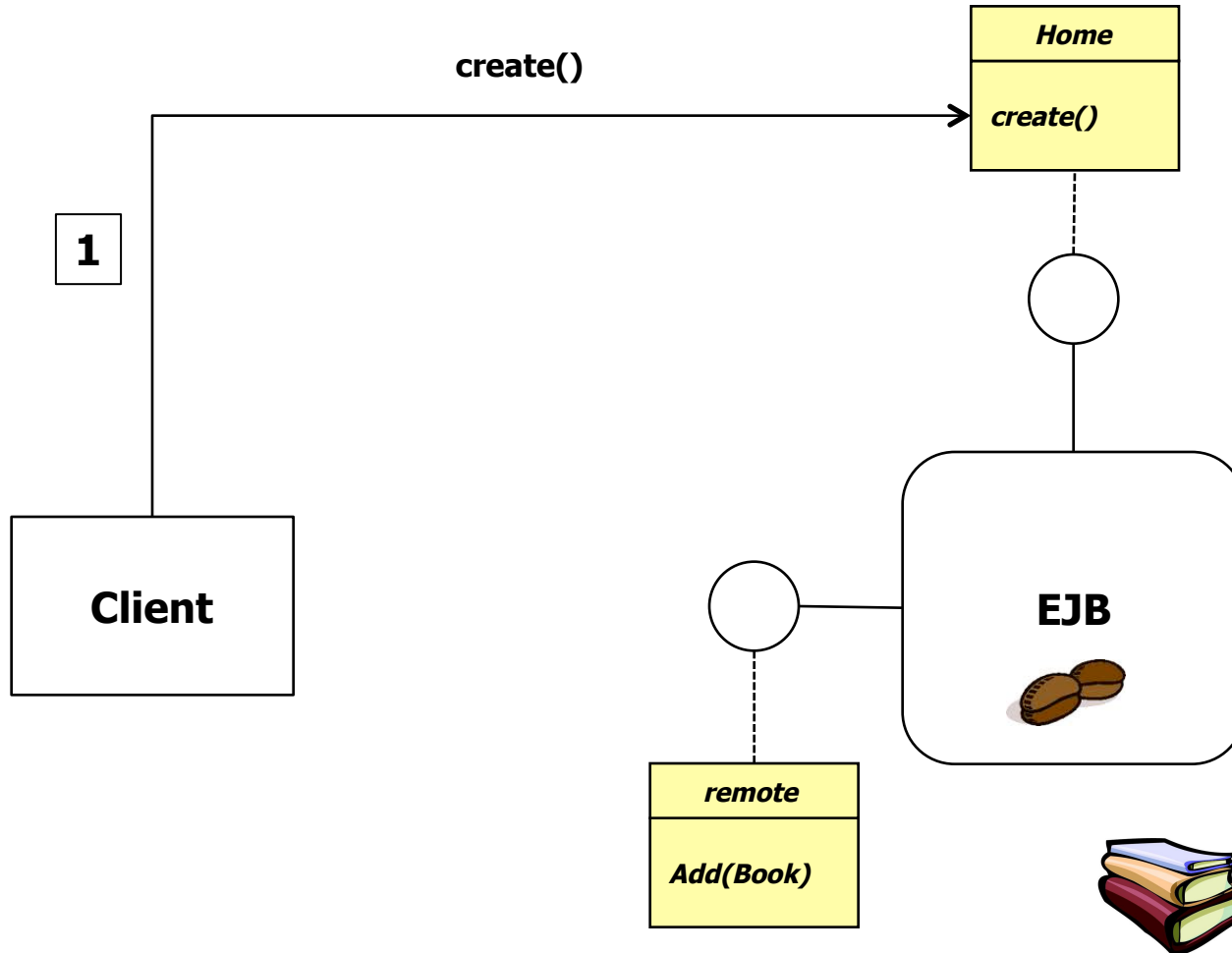
EJB IDL : Java Interface



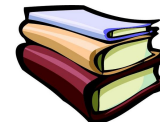
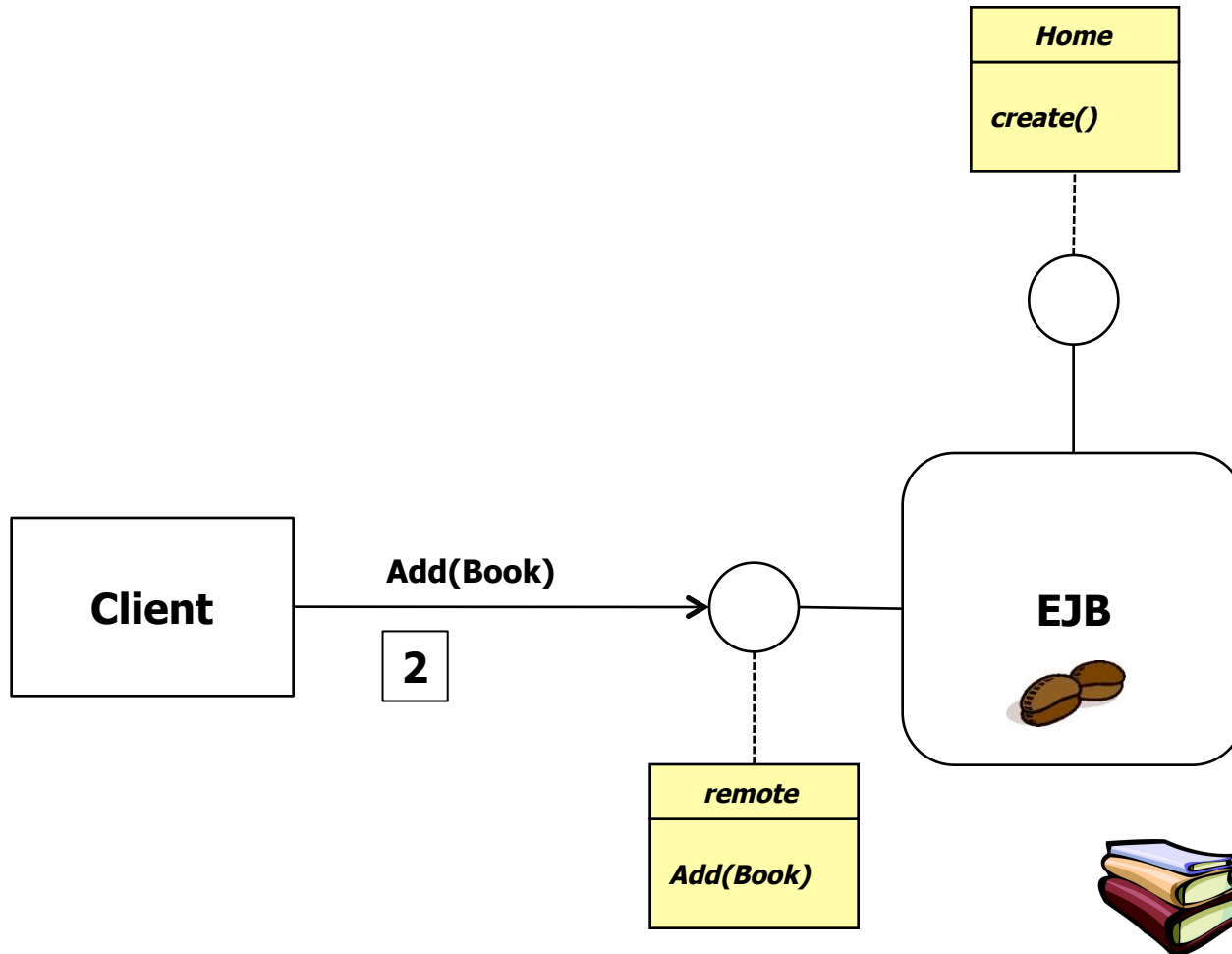
EJB 2.1 remote interface



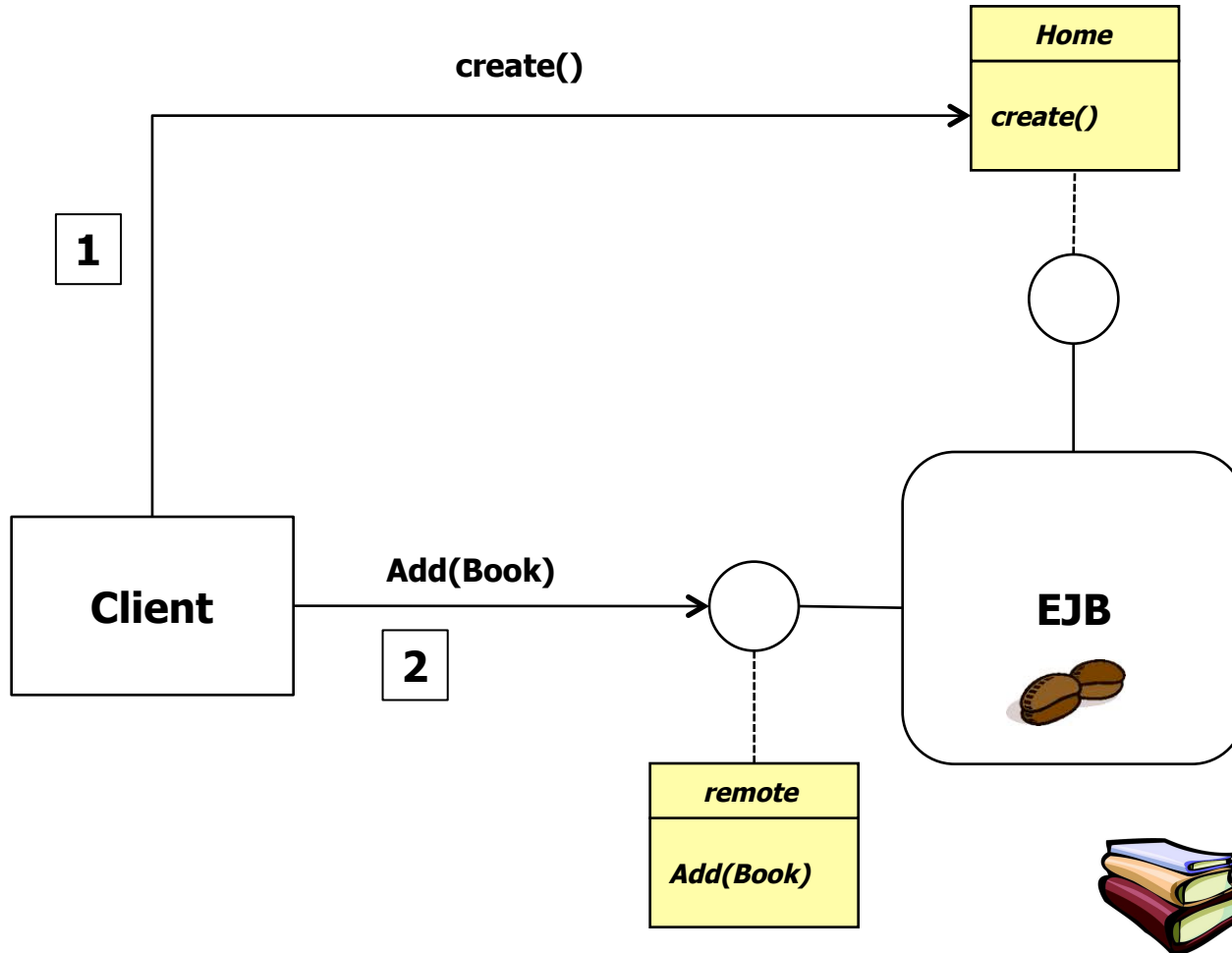
EJB 2.1 remote interface



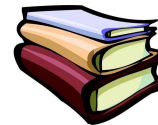
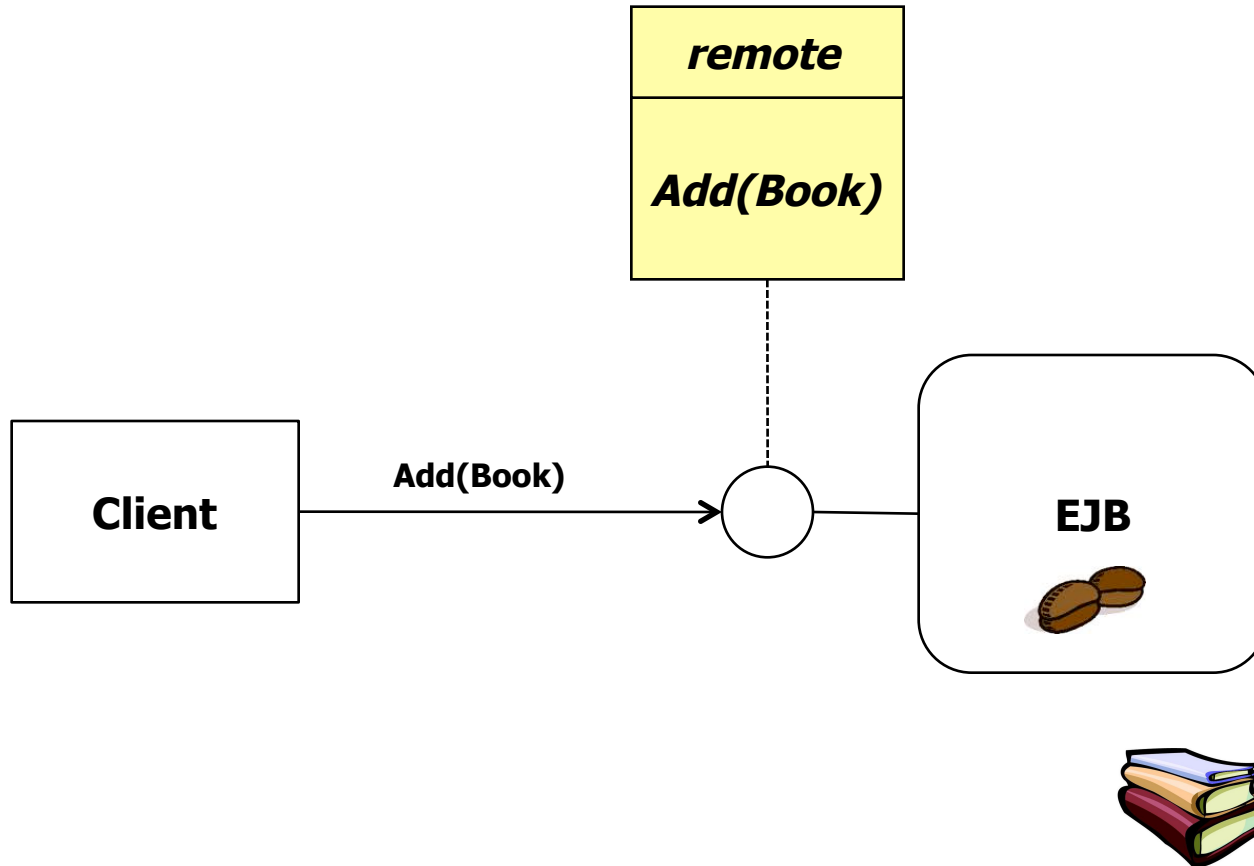
EJB 2.1 remote interface



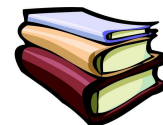
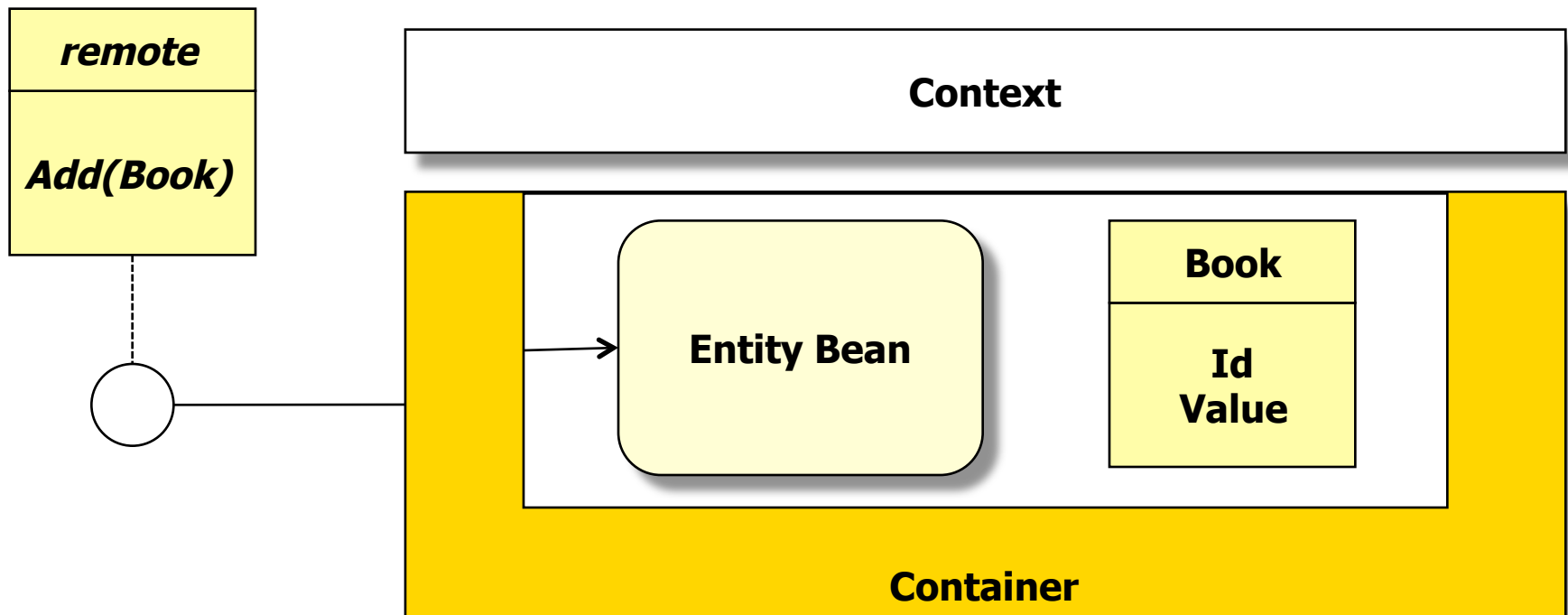
EJB 2.1 remote interface



EJB 3.x remote interface



Container and Context (local interface)



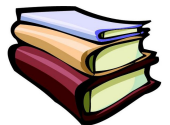
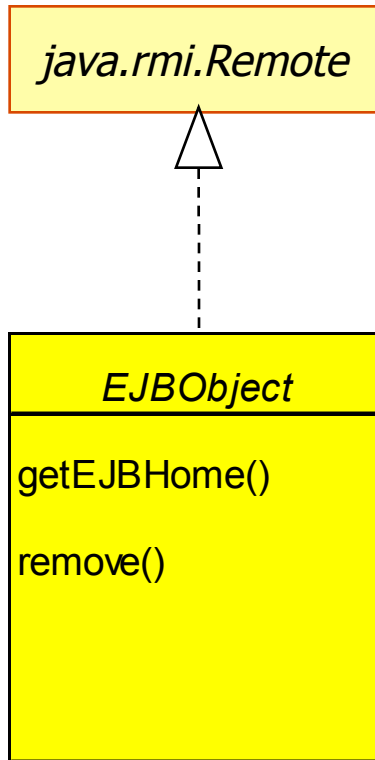
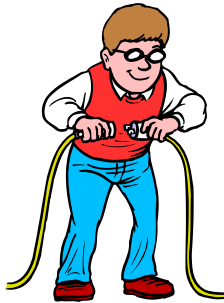
EJB javaDoc Remote and Local Interfaces



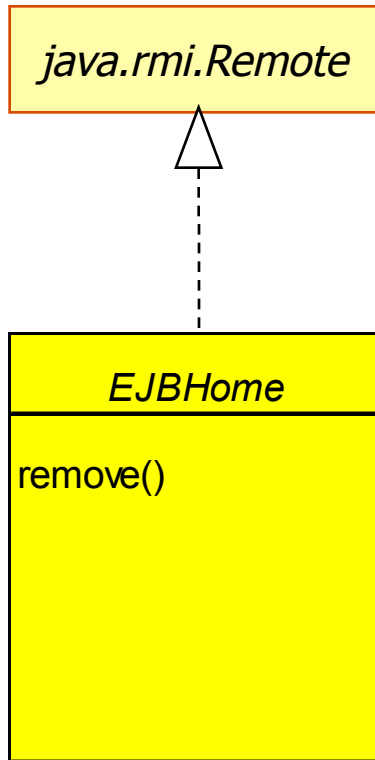
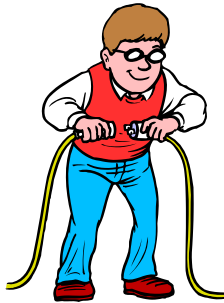
EJBObject	The EJBObject interface is extended by all enterprise Beans' remote interfaces.
EJBLocalObject	The EJBLocalObject interface must be extended by all enterprise Beans' local interfaces.
EJBHome	The EJBHome interface must be extended by all enterprise Beans' remote home interfaces.
EJBLocalHome	The EJBLocalHome interface must be extended by all enterprise Beans' local home interfaces.



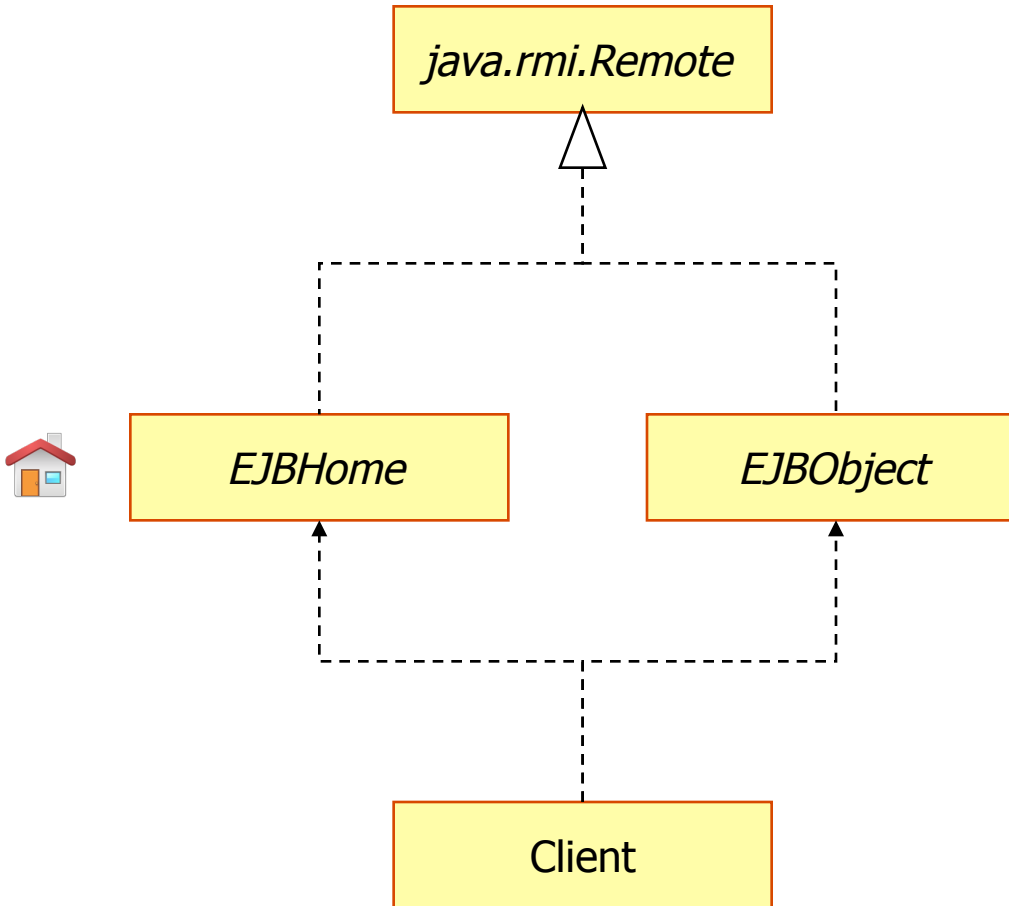
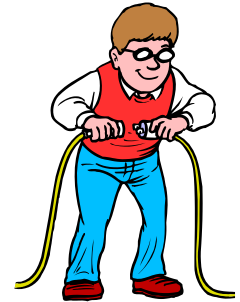
EJBObject (Remote Interfaces)



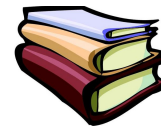
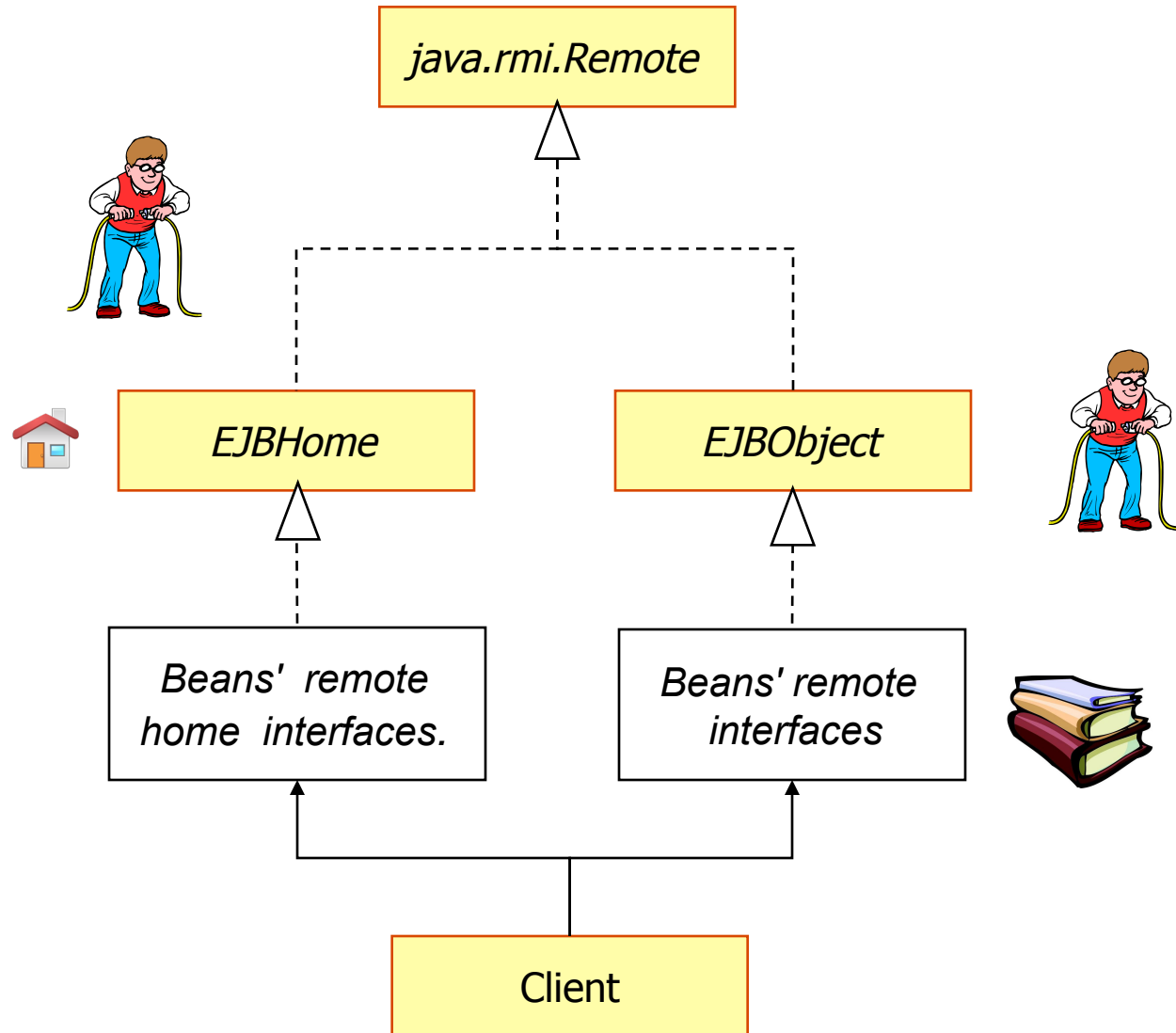
EJBHome (Remote Interfaces)



EJB remote interfaces



EJB remote interfaces



EJBObjects (from the EJB specifications)



- A client never directly accesses instances of the session bean's class.
- A remote or local client always uses the session bean's component interface to access a session bean instance.
- The class that implements the session bean's component interface is provided by the container.
- Instances of a session bean's remote interface are called session EJBObjects.



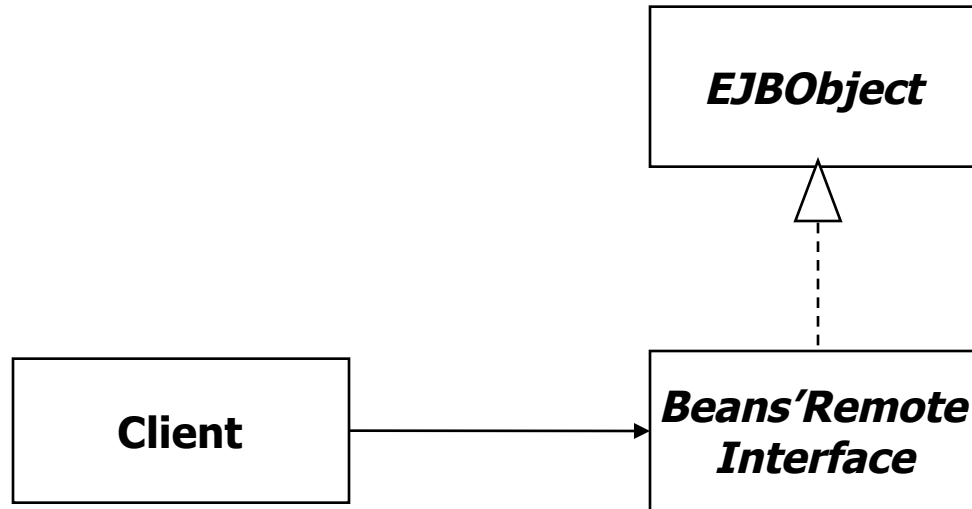
EJBObject (Remote Interfaces)



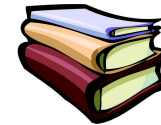
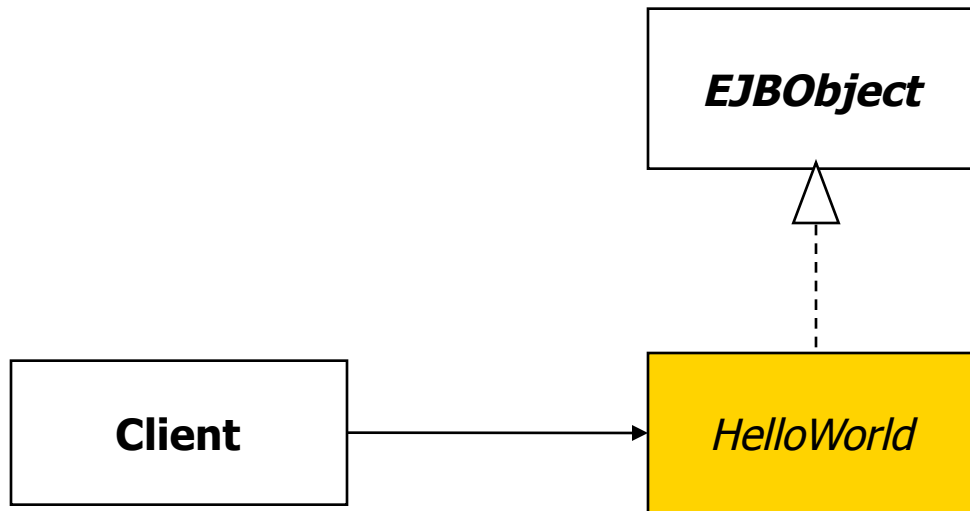
- A session EJBObject supports:
 - The business logic methods of the object.
 - The session EJBObject delegates invocation of a business method to the session bean instance.
 - The methods of the `javax.ejb.EJBObject` interface. These methods allow the client to:
 - Get the session object's remote home interface.
 - Get the session object's handle.
 - Test if the session object is identical with another session object.
 - Remove the session object.



Beans' remote interfaces.



Beans' remote interfaces.



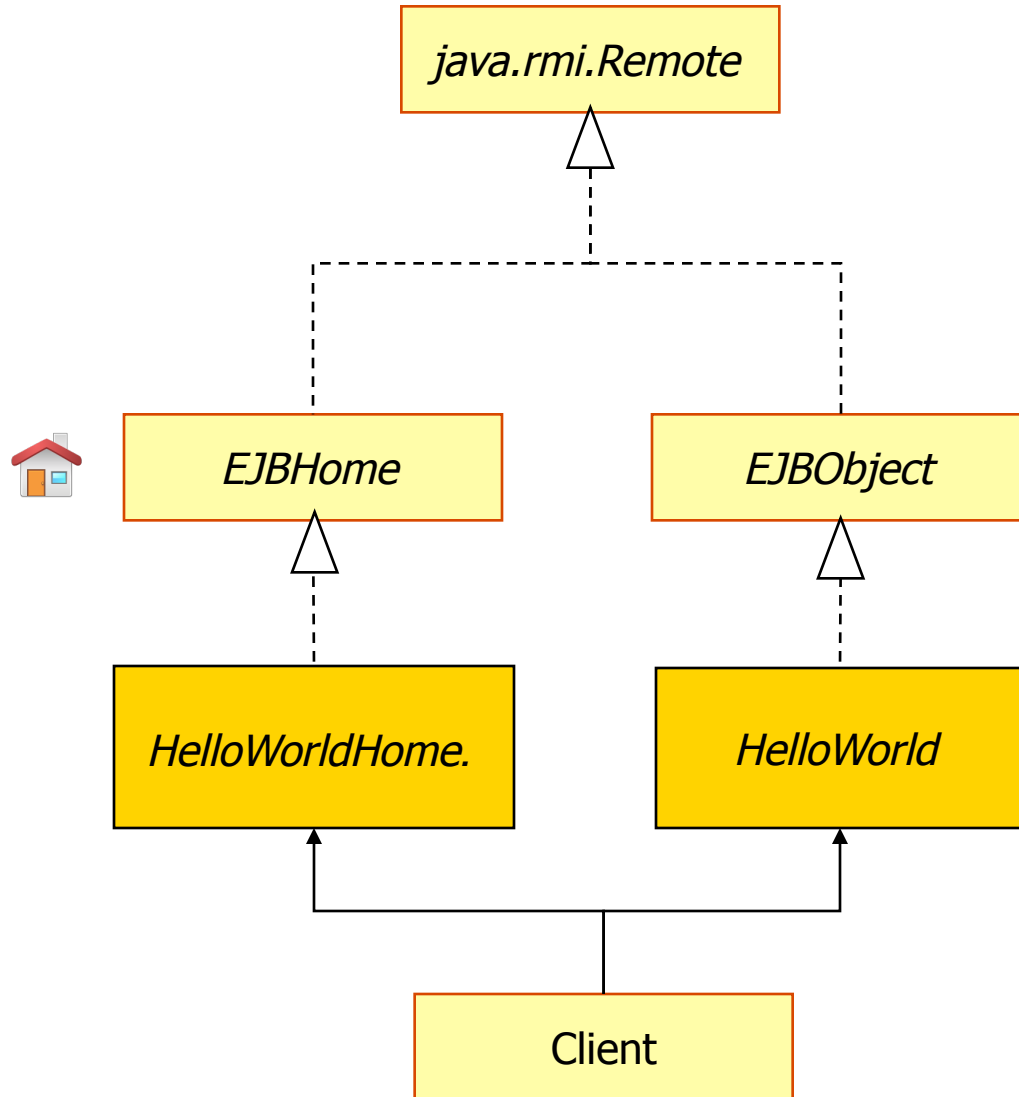
Beans' remote interfaces.



```
package helloWorld;  
  
import java.rmi.RemoteException;  
import javax.ejb.EJBObject;  
  
public interface HelloWorld extends EJBObject{  
  
    public String sayHello() throws RemoteException;  
  
}
```



EJB remote interfaces



EJB Home (Remote Interfaces)



```
package helloWorld;  
  
import java.rmi.RemoteException;  
  
import javax.ejb.CreateException;  
import javax.ejb.EJBHome;  
  
public interface HelloWorldHome extends EJBHome {  
  
    public HelloWorld create() throws CreateException, RemoteException;  
  
}
```

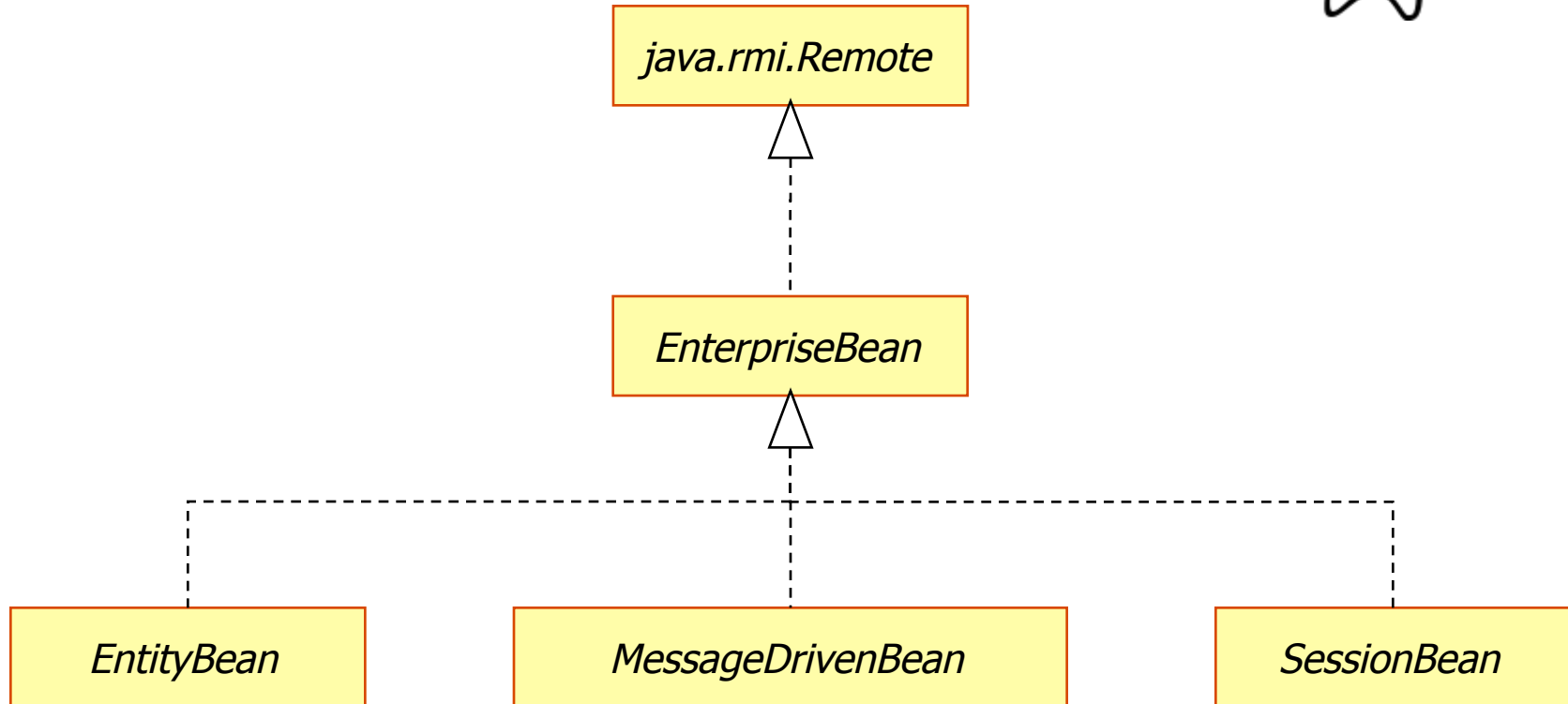


EJB Beans' Interface (local)

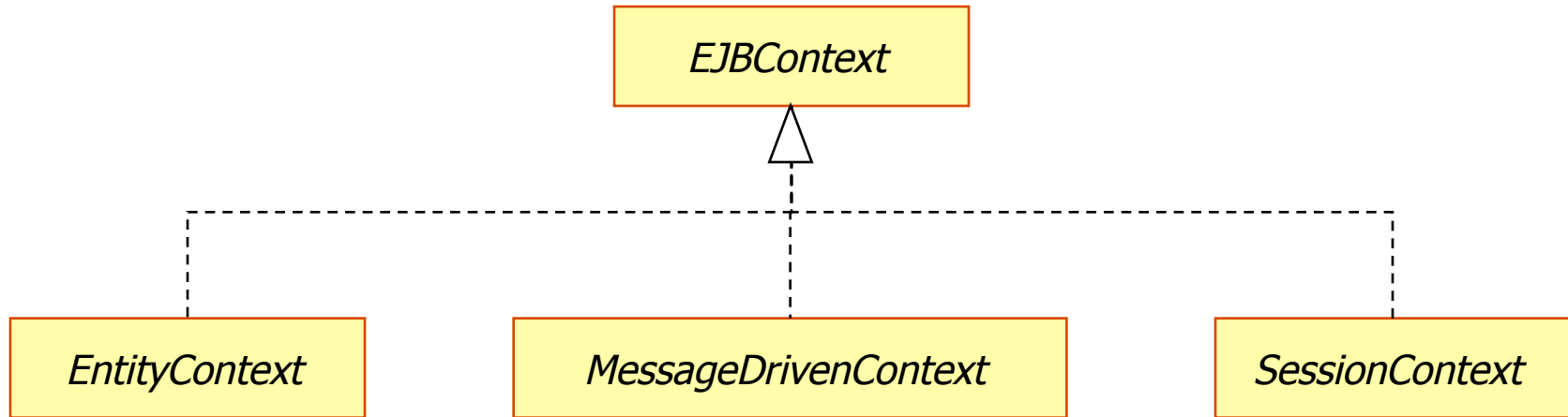


EnterpriseBean	The EnterpriseBean interface must be implemented by every enterprise Bean class.
EJBContext	The EJBContext interface provides an instance with access to the container-provided runtime context of an enterprise Bean instance.
EntityContext	The EntityContext interface provides an instance with access to the container-provided runtime context of an entity enterprise Bean instance.
SessionContext	The SessionContext interface provides access to the runtime session context that the container provides for a session enterprise Bean instance.
MessageDrivenContext	The MessageDrivenContext interface provides access to the runtime message-driven context that the container provides for a message-driven enterprise Bean instance.

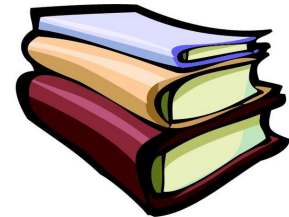
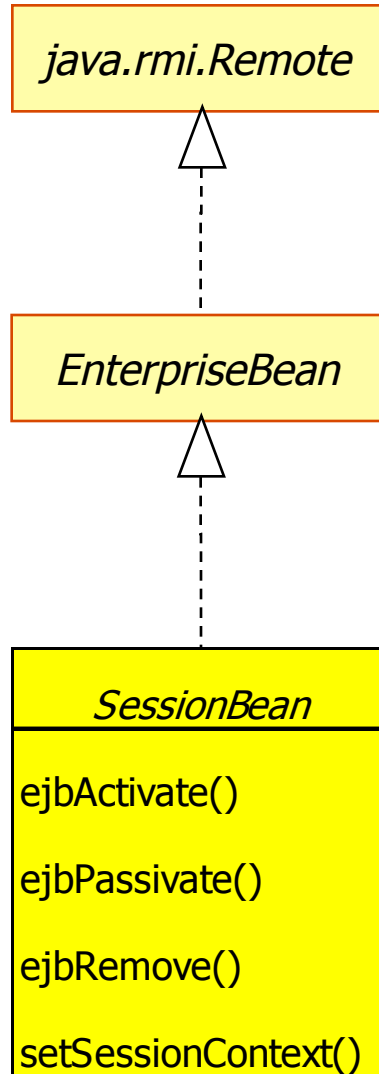
EJB EnterpriseBean (local)



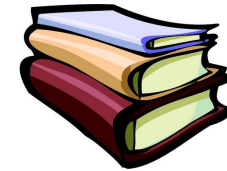
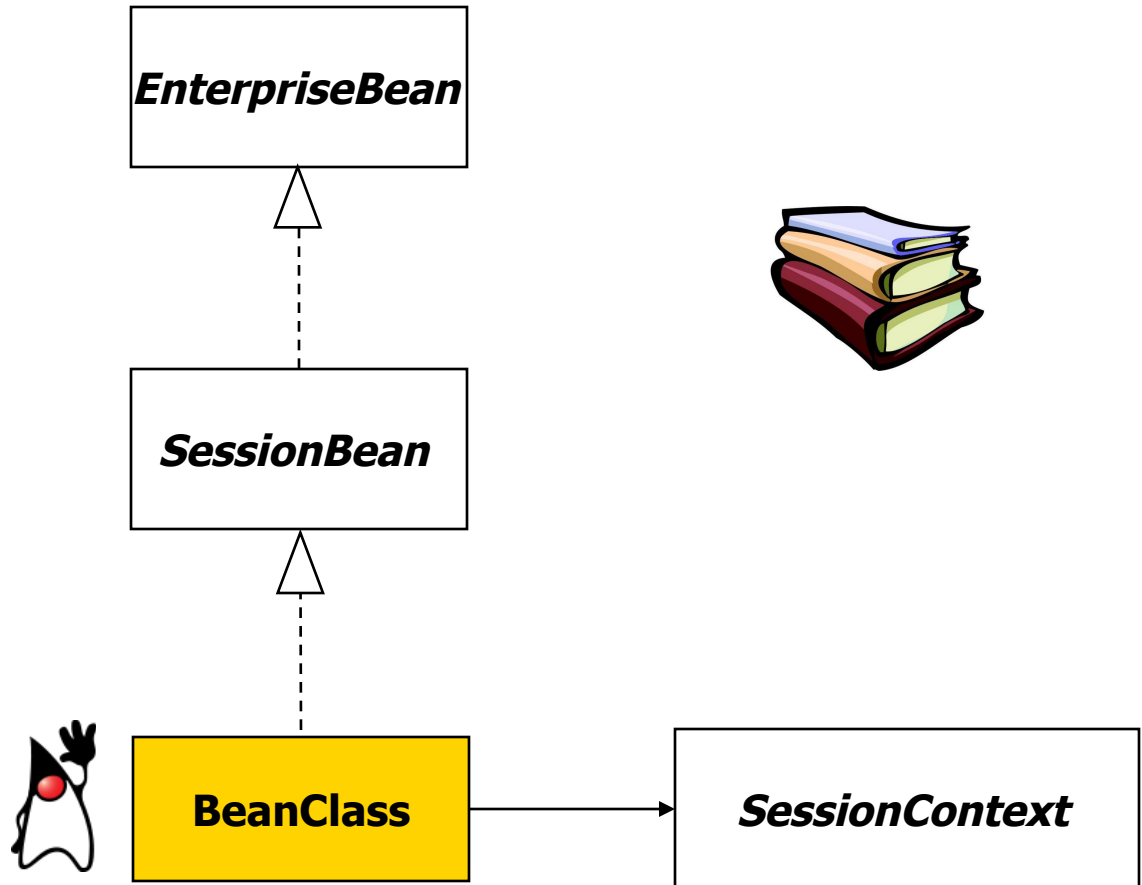
EJB Context (local)



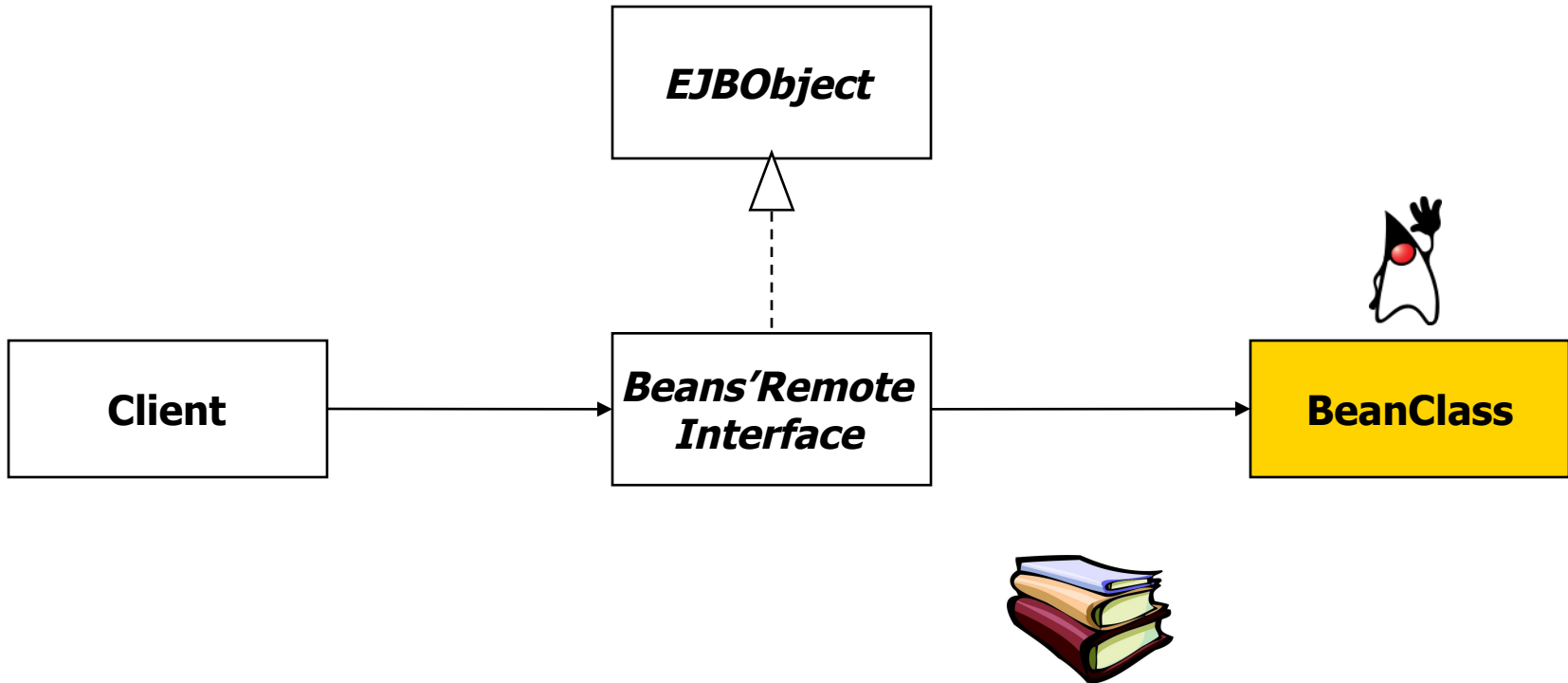
Interface SessionBean (local)



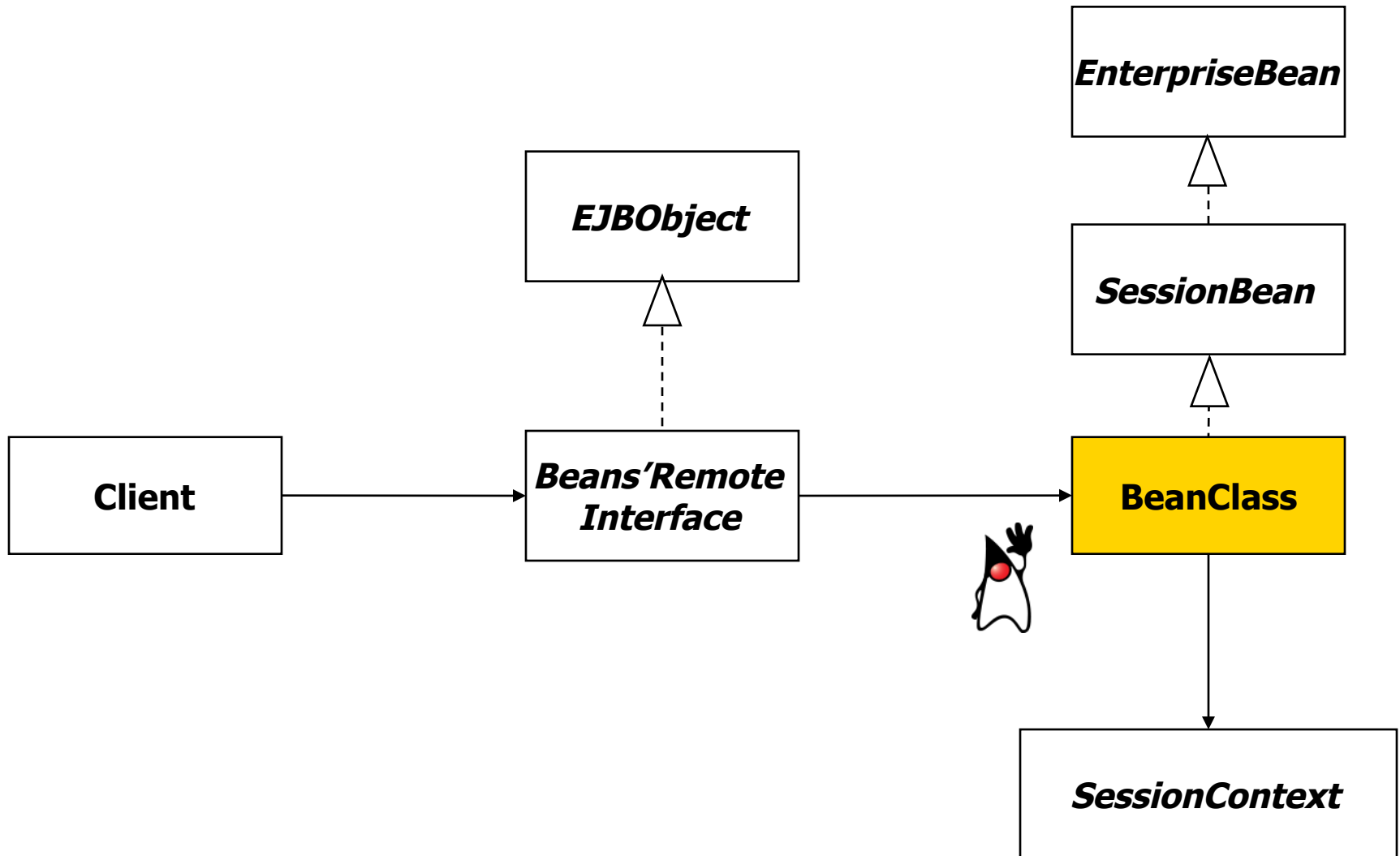
EJB Session Bean (local)



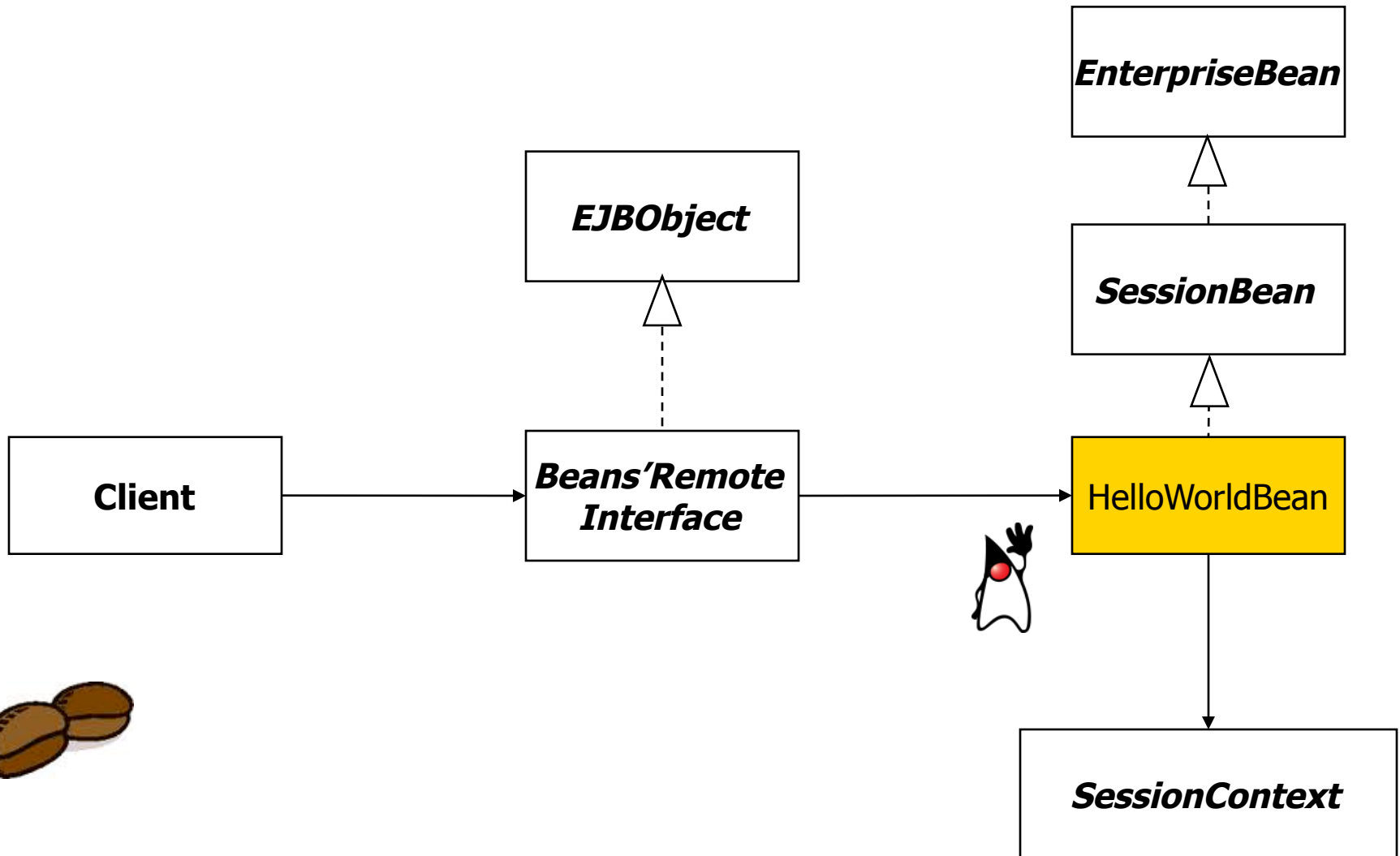
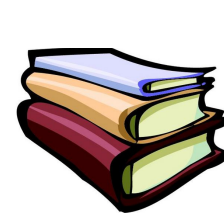
Session Beans' remote interfaces.



Session Beans' local interfaces.



Session Beans' (local) interfaces.



BeanClass



```
package helloWorld;

import java.rmi.RemoteException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;

public class HelloWorldBean implements SessionBean {

    protected SessionContext ctx;

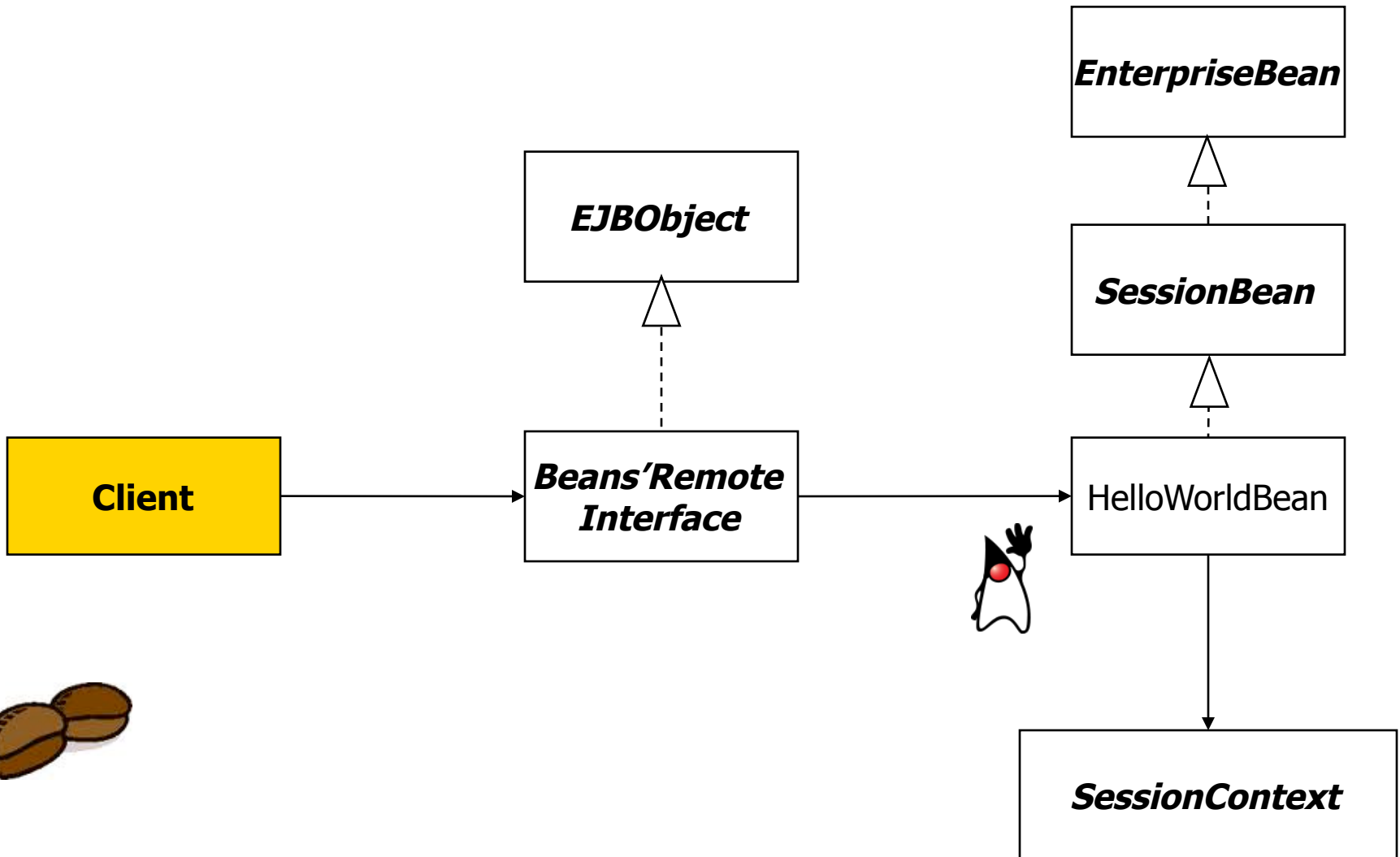
    private static final long serialVersionUID = 1L;

    public String sayHello() {
        return "Hello, world !";
    }

    public void ejbCreate() {}
    public void ejbActivate() throws EJBException, RemoteException {}
    public void ejbPassivate() throws EJBException, RemoteException {}
    public void ejbRemove() throws EJBException, RemoteException {}
    public void setSessionContext(SessionContext Context) {ctx = Context;}
}
```



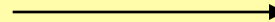
Session Beans' interfaces.



HelloClient.java



```
package client;  
import helloWorld.HelloWorld;  
import helloWorld.HelloWorldHome;  
import javax.naming.InitialContext;  
import javax.naming.Context;  
import javax.rmi.PortableRemoteObject;
```



HelloClient.java



```
public class HelloClient {  
    public static void main(String[] args) {  
        Context initialContext = null;  
        try {  
            initialContext = new InitialContext();  
            Object objRef = (HelloWorldHome) initialContext .lookup("ejb/Hello");  
            HelloWorldHome home = (HelloWorldHome) PortableRemoteObject.narrow(objRef, HelloWorldHome.class);  
            HelloWorld hello = home.create();  
            System.out.println(hello.sayHello());  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



Type Narrowing

- A client program that is intended to be interoperable with all compliant EJB container implementations must use the:
 - `javax.rmi.PortableRemoteObject.narrow` method
- to perform type-narrowing of the client-side representations of the remote home and remote interfaces.



EJB local Interface SessionBean Methods

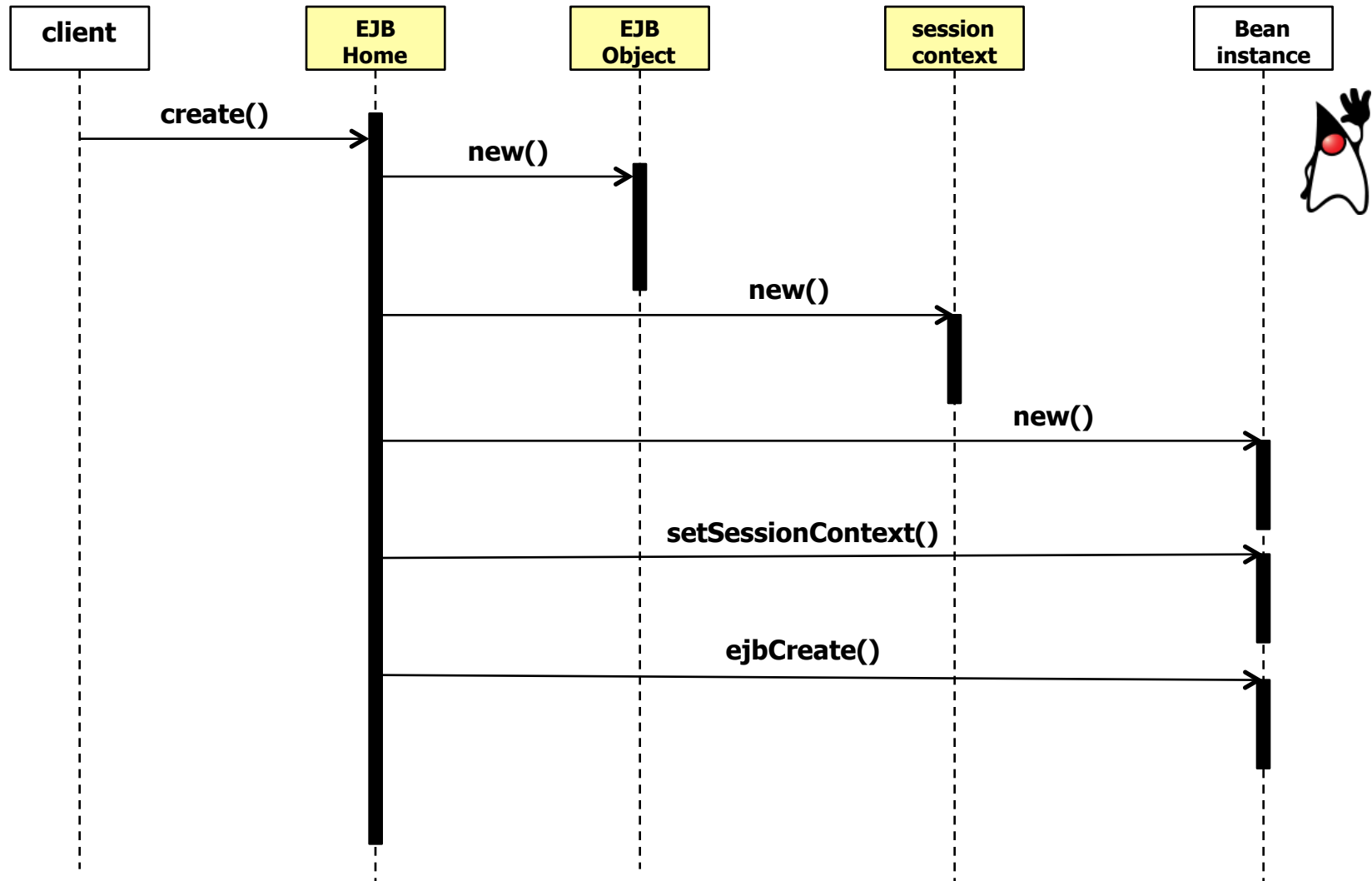


<code>ejbActivate()</code>	The activate method is called when the instance is activated from its "passive" state.
<code>ejbPassivate()</code>	The passivate method is called before the instance enters the "passive" state.
<code>ejbRemove()</code>	A container invokes this method before it ends the life of the session object.
<code>setSessionContext()</code>	Set the associated session context.





Object Interaction Diagrams for a Statefull Session Bean





■ Post development customization.

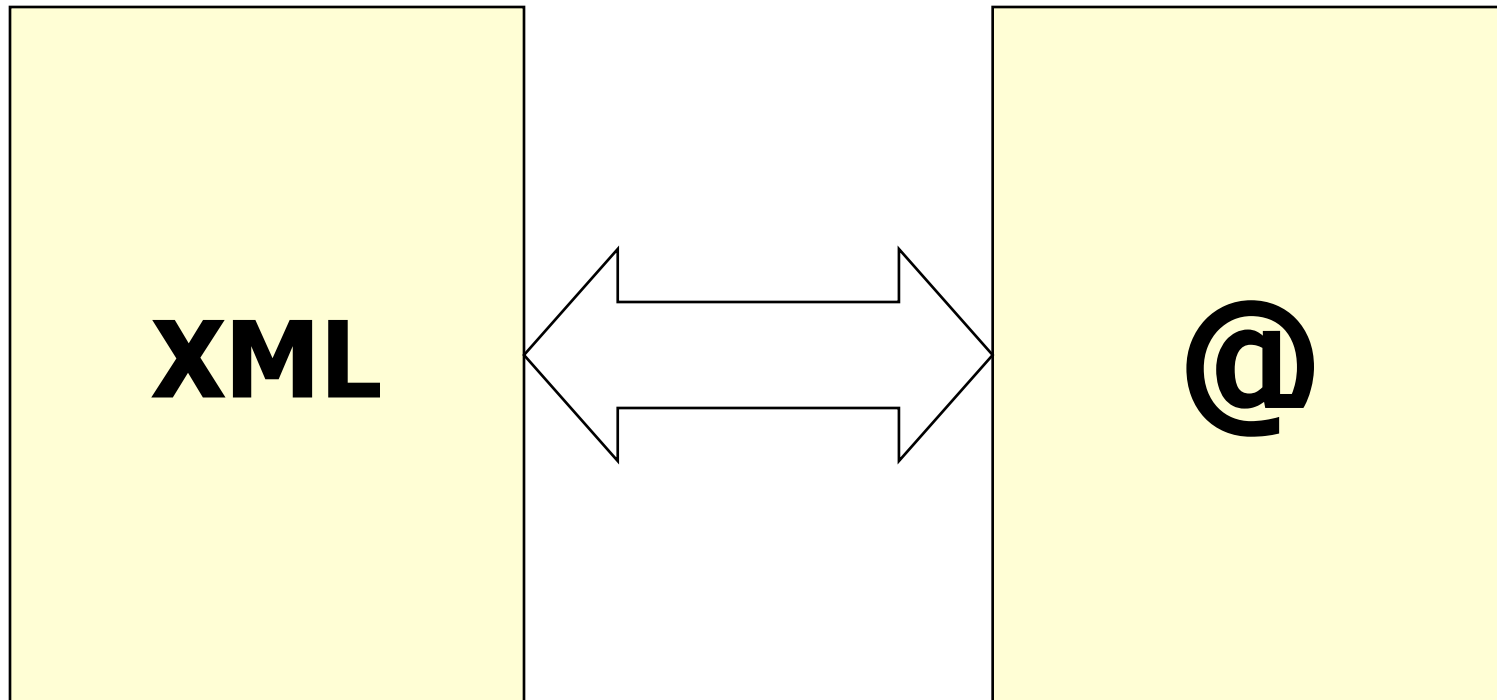
```
<ejb-jar>
  <description>HelloWorld deployment descriptor</description>
  <display-name>HelloWorld</display-name>
  <enterprise-beans>
    <session>
      <description>HelloWorld deployment descriptor</description>
      <display-name>HelloWorld</display-name>
      <ejb-name>HelloWorld</ejb-name>
      <home>helloWorld.HelloWorldHome</home>
      <remote>helloWorld.HelloWorld</remote>
      <ejb-class>helloWorld.HelloWorldBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
</ejb-jar>
```

Naming Conventions for Enterprise Beans



Item	Syntax	Example
Enterprise bean name	<code><name>Bean</code>	HelloWorldBean
EJB JAR display name	<code><name>JAR</code>	HelloWorldJAR
Enterprise bean class	<code><name>Bean</code>	HelloWorldBean
Home interface	<code><name>Home</code>	HelloWorldHome
Remote interface	<code><name></code>	HelloWorld
Local home interface	<code><name>LocalHome</code>	HelloWorldLocalHome
Local interface	<code><name>Local</code>	HelloWorldLocal
Abstract schema	<code><name></code>	HelloWorld





**Deployment
Descriptor**

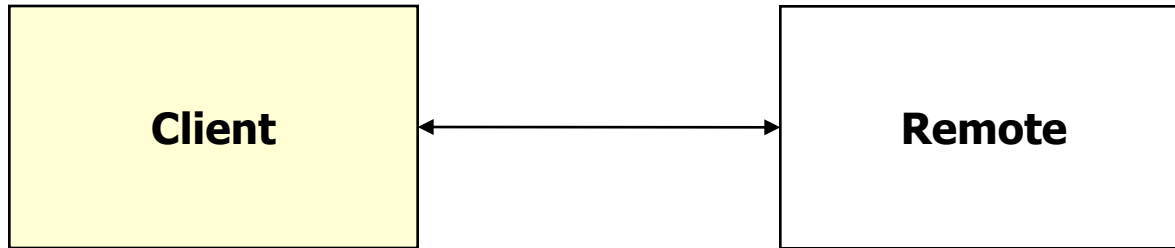
Annotations



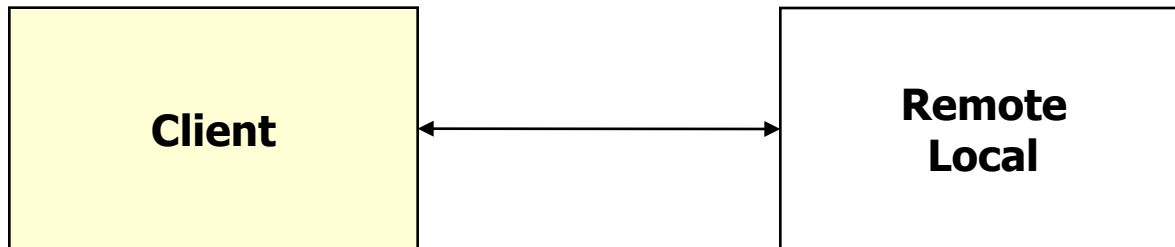
EJB versions



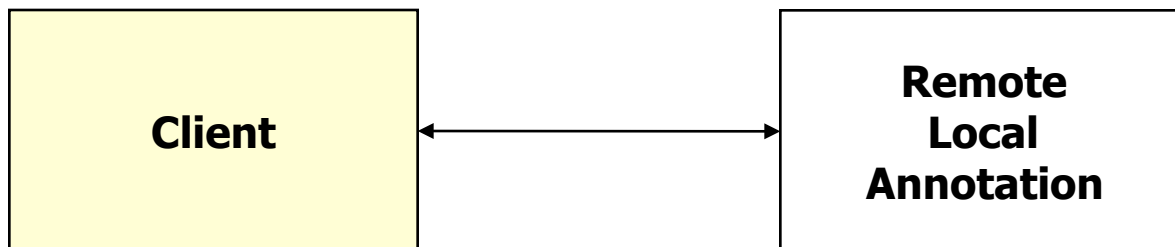
EJB 1.x



EJB 2.x



EJB 3.x



EJB 3.0 Annotation @



Local	When used on the bean class, declares the local business interface(s) for a session bean.
LocalHome	Declares the Local Home or adapted Local Home interface for a session bean.
MessageDriven	Component-defining annotation for a message driven bean.
PostActivate	Designates a method to receive a callback after a stateful session bean has been activated.
PrePassivate	Designates a method to receive a callback before a stateful session bean is passivated.
Remote	Declares the remote business interface(s) for a session bean.
RemoteHome	Declares the Remote Home or adapted Remote Home interface for a session bean.
Remove	Applied to a business method of a stateful session bean class.
Stateful	Component-defining annotation for a stateful session bean.
Stateless	Component-defining annotation for a stateless session bean.

