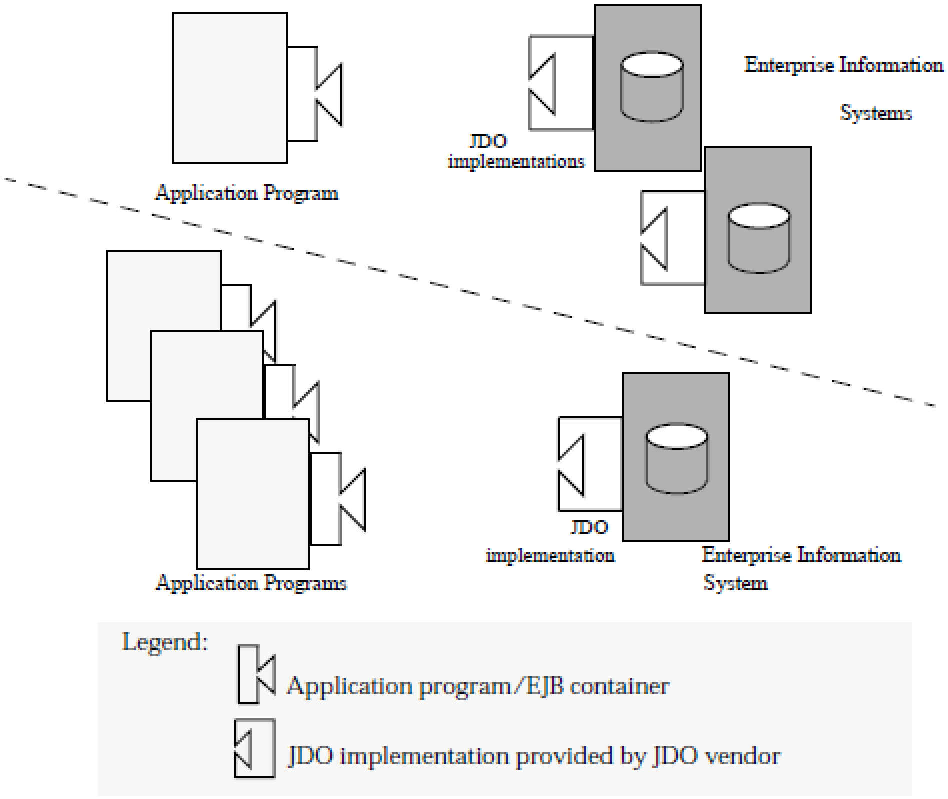
LIYANA ARACHCHIGE RANIL

SHORT NOTES / JAVA DATA OBJECTS (JDO) / BUSINESS TIER

1. JDO is a specification introduced by SUN to cater for Object Relational Persistence
2. JDO conveniently pass the development of PERSISTENCE logic to a VENDOR specified interface
3. JDO was not embraced by the Community
4. JDO involved java byte code ENHANCING to add persistence related Meta-Data
5. The MAPPING to relational data base from Java world is done after the Vendor Interface
6. JDO supports EIS , RELATIONAL DB , OBJECT ORIENTED DB etc

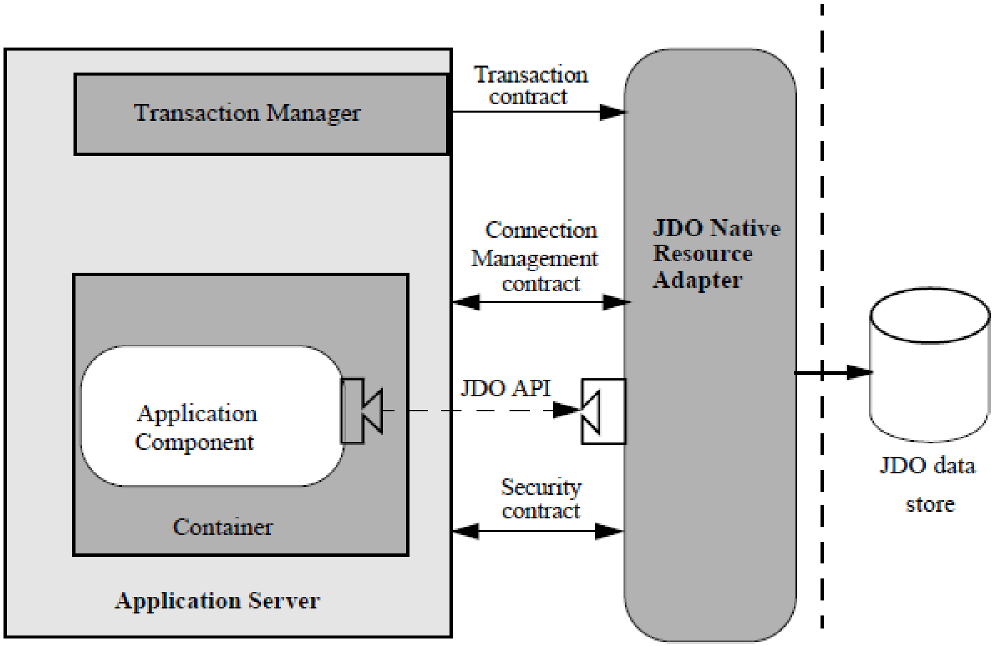
Figure 1.0 Standard plug-and-play between application programs and EISes using JDO



7.

8.

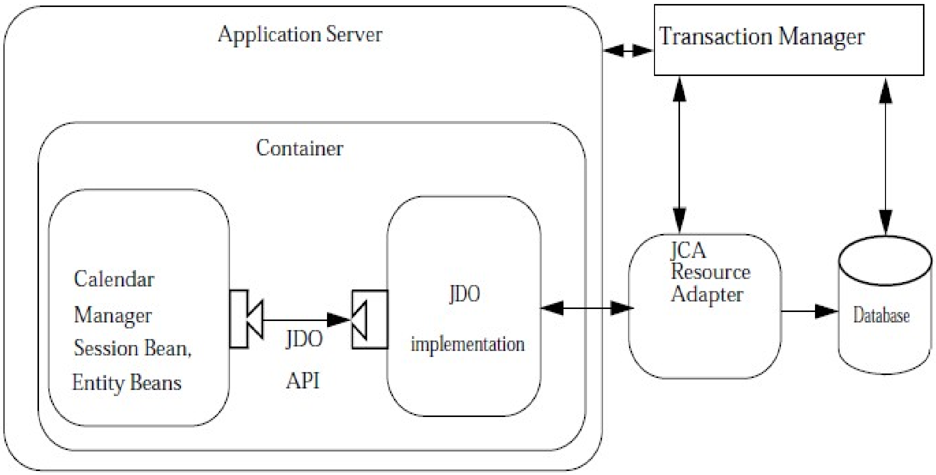
Figure 3.0 Contracts between application server and native JDO resource adapter



9. JDO in ENTERPRISE environment

10.

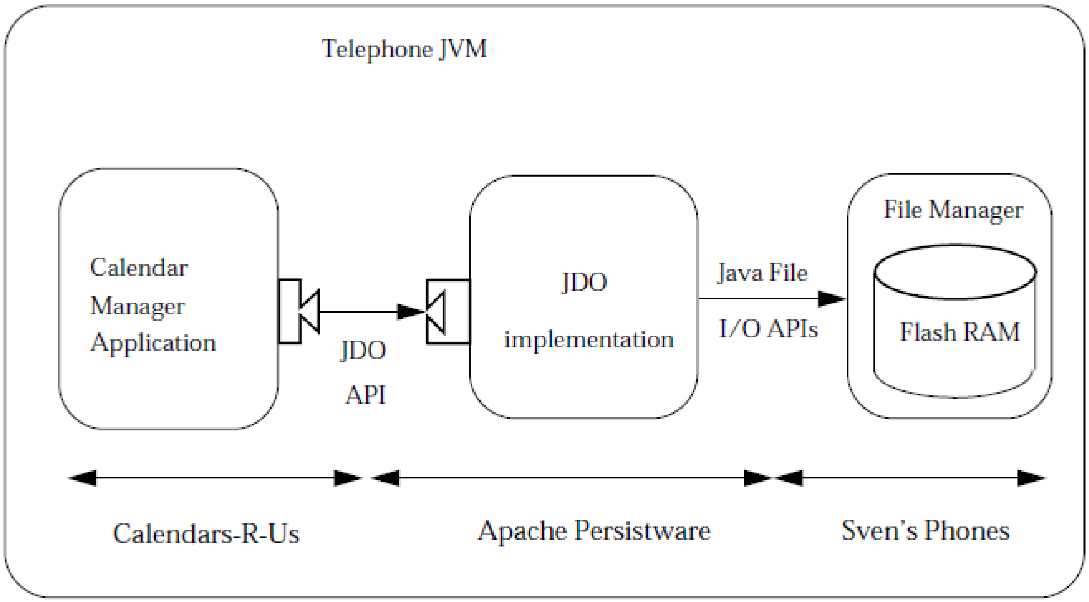
Figure 6.0 Scenario: Enterprise Calendar Manager



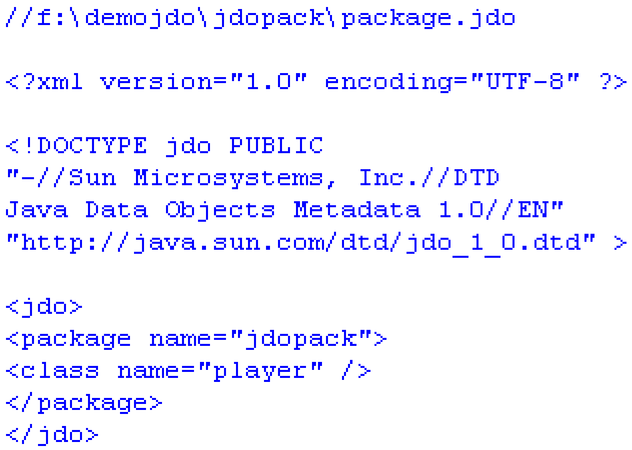
11. JDO in EMBEDED environment

12.

Figure 5.0 Scenario: Embedded calendar manager



13. JDO also needs XML file to map PERSISTABLE objects to DATABASE (XML meta data file for JDO)
14. XML meta data file is usually named as "**package.jdo**"



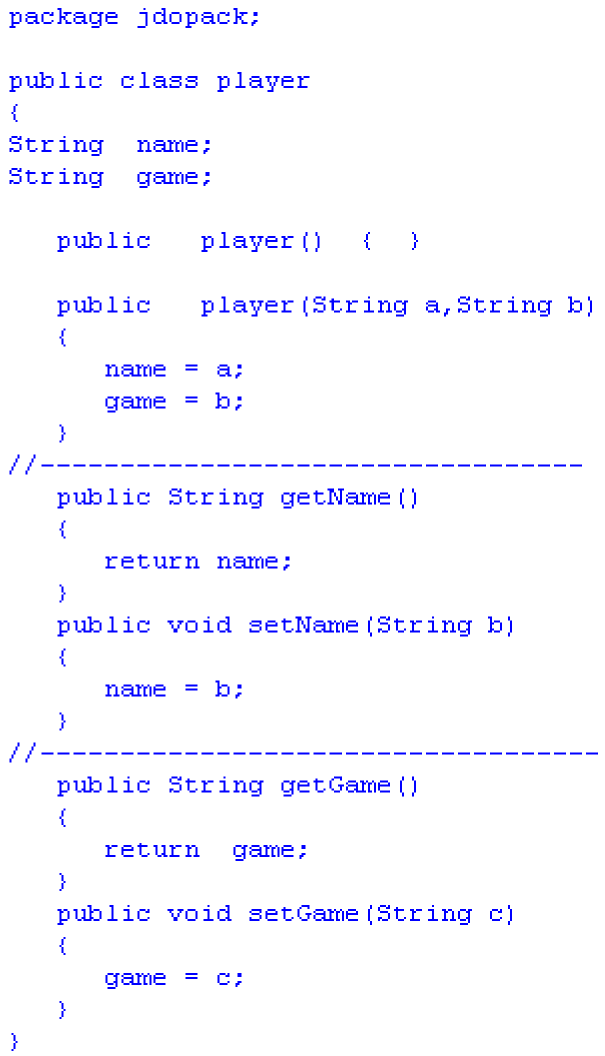
```
//f:\demojdo\jdopack\package.jdo

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE jdo PUBLIC
"-//Sun Microsystems, Inc.//DTD
Java Data Objects Metadata 1.0//EN"
"http://java.sun.com/dtd/jdo_1_0.dtd" >

<jdo>
<package name="jdopack">
<class name="player" />
</package>
</jdo>
```

a.

15. Usually all the JDO persistable classes must implement javax.jdo.spi.PersistanceCapable interface. Since this is practically impossible to do for all the classes , an ENHANCER can be used

16. ENHANCER would enhance the CLASS FILES to be PERSISTANCE CAPABLE

17. A typical JDO persistable object



```
package jdopack;

public class player
{
String  name;
String  game;

   public   player()  {  }

   public   player(String a,String b)
   {
      name = a;
      game = b;
   }
//------------------------------------
   public String getName()
   {
      return name;
   }
   public void setName(String b)
   {
      name = b;
   }
//-------------------------------------
   public String getGame()
   {
      return  game;
   }
   public void setGame(String c)
   {
      game = c;
   }
}
```

a.

18. Once JDO persistable classes are compiled , you need to run the ENHANCER to enhance the classes

19. ENHANCER would in tern use Compiled class files and the JDO XML meta data file to ENHANCE the persistable classes

20. JDO supports both DATA STORE IDENTITY and APPLICATION IDENTITY , JPA only supports APPLICATION IDENTITY for persistable objects