

ASD. Java RMI “more general example”

- Distributed Object programming example
- Outline of step-by-step instructions for building a simple, yet general distributed application in Java RMI
- Script of practical:
<http://paloalto.unileon.es/ds/Lab/RMI/DS-Lab-RMI-English-2017.pdf>
- Source code:
<http://paloalto.unileon.es/ds/Lab/RMI/sdrmiexample.zip>

What is Java RMI?

- Java Remote Method Invocation
 - Distributed Objects in Java
 - *Some* of the methods of a distributed object can be invoked from a remote client

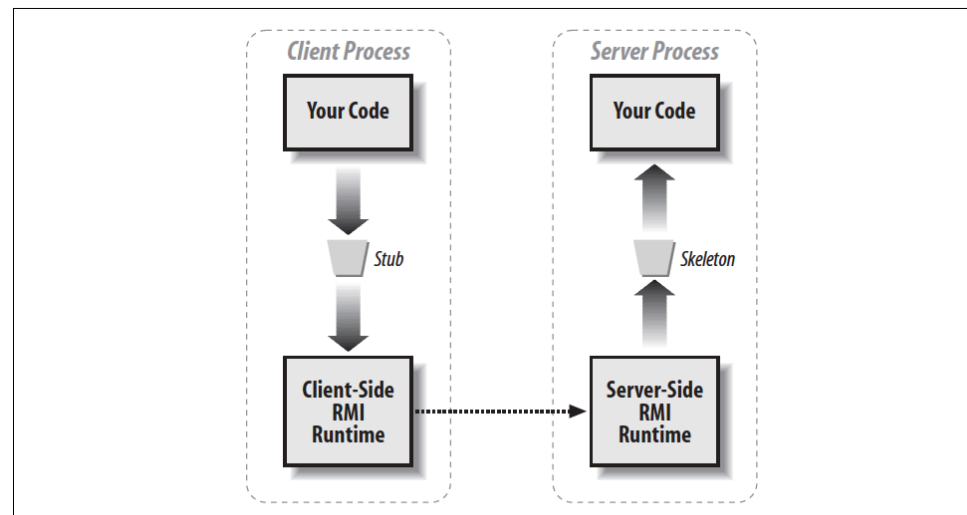
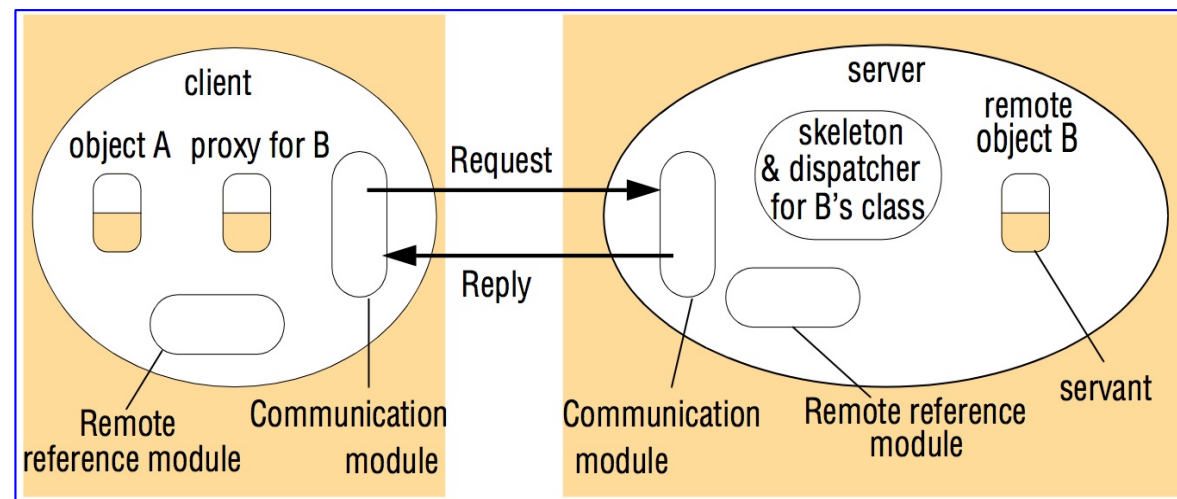


Figure 16-1. The layers in a remote call

Context

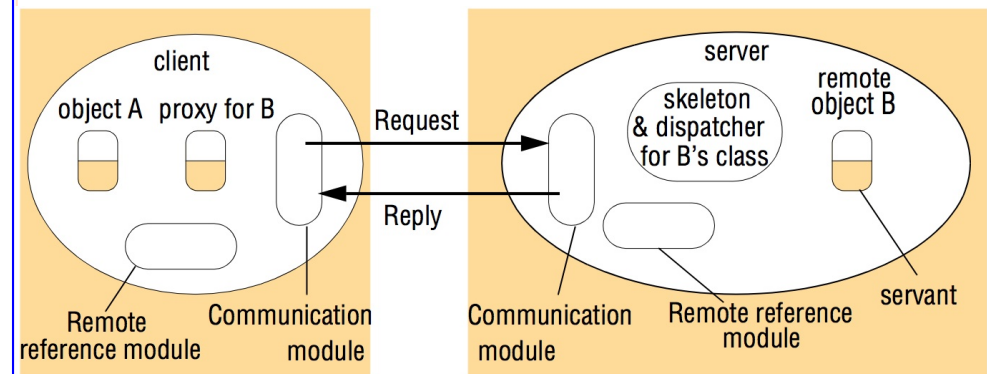
- We aim to compose an **RMI Client/Server** application
- **Server** exports a remote object (Also known as **Servant**) to the network
- **Client** accesses the Servant via RMI and invokes its remote methods
 - Which Servant's methods can be remotely invoked? Those included in the **Remote Interface**
 - The **client** only knows the **Remote Interface** (the implementation is away, on the remote RMI server)



Process to write a simple Java RMI example

1. Write the Java **Remote Object** implementation (**Servant**)

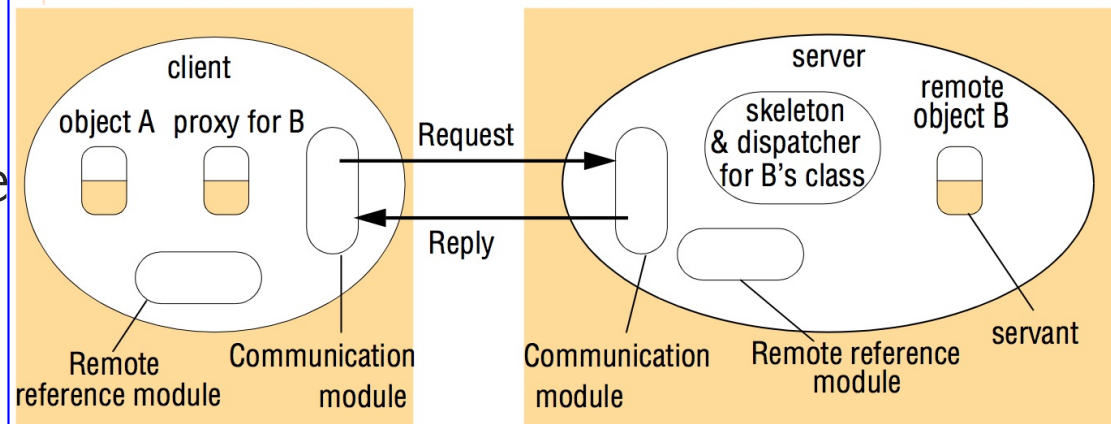
- extends `UnicastRemoteObject`
- Remote methods and constructor:
 - `public`
 - throws `RemoteException`
- (Can have non-public methods)



Process to write a simple Java RMI example

2. Write the **Java Interface** corresponding to the **Servant**

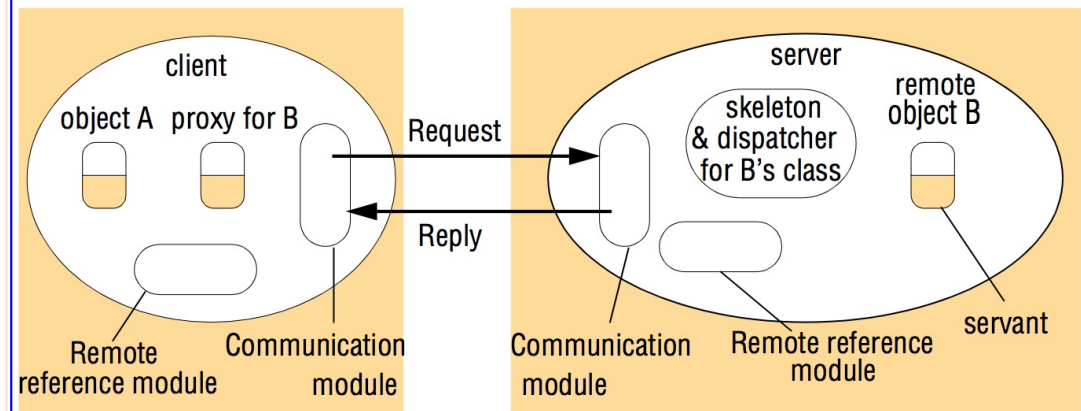
- Which are the Servant's remote methods?
- The **Remote Interface** specifies the signature of those methods
- **Remote Interface** will allow us to compile the client without the Servant's implementation source code or class



Process to write a simple Java RMI example

3. Start Java `rmiregistry`

- Servants are registered into the `rmiregistry`
 - Contains entries [**Name, Servant**]
 - It must be running before any servant can be registered
 - Unix: `$ rmiregistry [options] &`



Process to write a simple Java RMI example

4. Write the **Java RMI Server** that will instantiate and start each **Servant**

//Instantiate Servant:

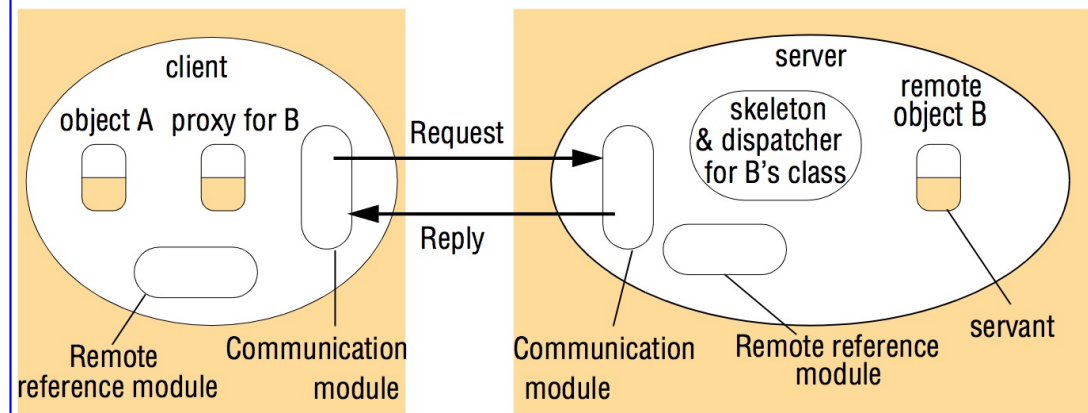
```
SDRemoteObject ro = new SDRemoteObjectImpl();
```

//Register it with the rmiregistry:

```
Naming.rebind("//localhost:1099/Aslan", ro);
```

5. Run the **Server**

- Make sure to set the Java property `"java.rmi.server.codebase"` properly so that the Servant's class is accessible to the client (annotated serialization) when the rmiregistry uploads the proxy object



Process to write a simple Java RMI example

6. Write the RMI Client and run it

(Must have the *remote interface* accessible)

- Locate rmiregistry:

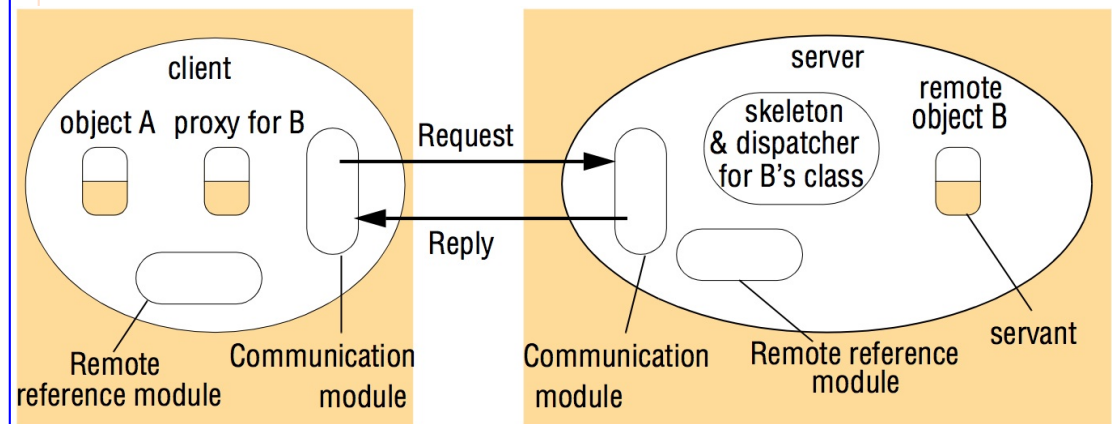
```
Registry r = LocateRegistry.getRegistry("192.168.2.131");
```

- Lookup servant named "Aslan":

```
SDRemoteObject cc = (SDRemoteObject) r.lookup("Aslan");
```

- Invoke the remote methods:

```
cc.longStringHash(ts);    cc.factorial(7);
```



ASD. Java RMI “more general example”

The End