



Java RMI !!!

Distributed object programming example

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

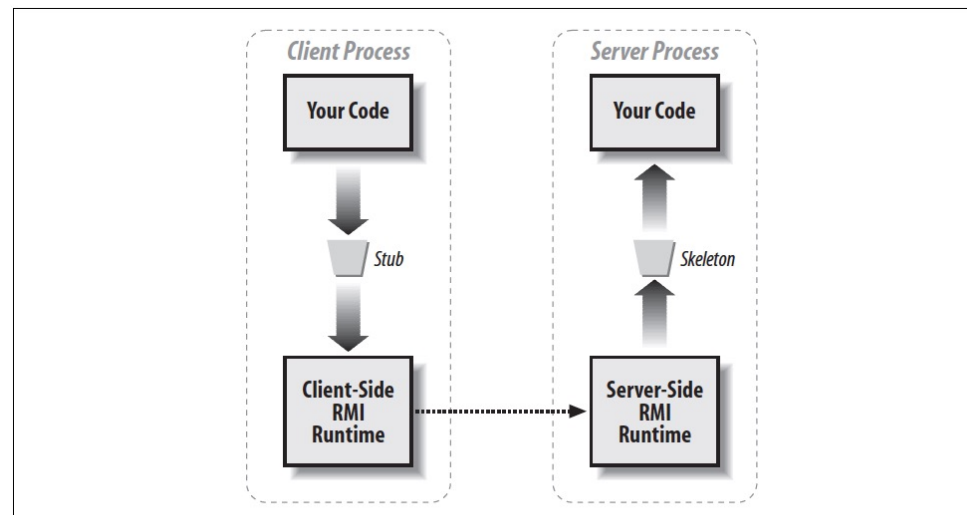
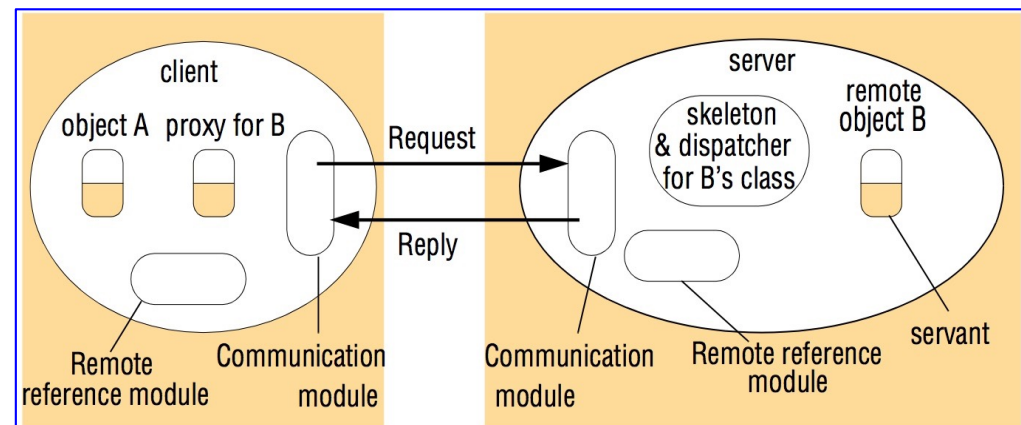


Figure 16-1. The layers in a remote call © O'Reilly Pub. Co. Book on Java RMI

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)



© 2012 Prentice Hall Kindberg Blair Dollimore Distributed Systems

Process to write a simple Java RMI example

1. Write the Java Remote Object implementation (The servant)

- Remote methods and constructor:

- `public`
- `throws RemoteException`
- Example: A remote object that listens on a specific TCP port

```
public class SDRemoteObjectImpl extends UnicastRemoteObject
implements SDRemoteObject {
```

```
    /* This constructor along with the throws clause is required so that
    * objects are appropriately created as remote objects.
    */
```

```
    */
```

```
    public SDRemoteObjectImpl(int port) throws RemoteException {
        super(port);
```

```
    } ...
```

```
    ...
```

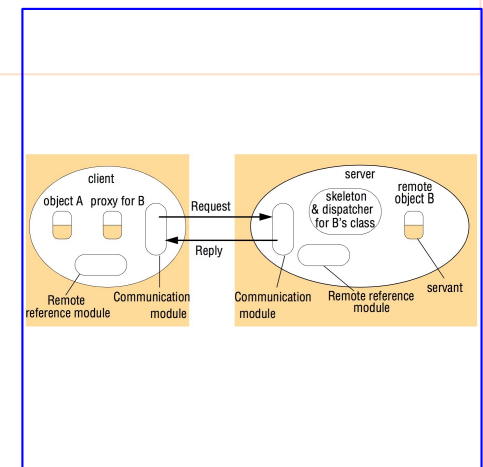
- Can have non-public methods

Process to write a simple Java RMI example

2. Write the Java Interface corresponding to the servant (The remote object itself)

- 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

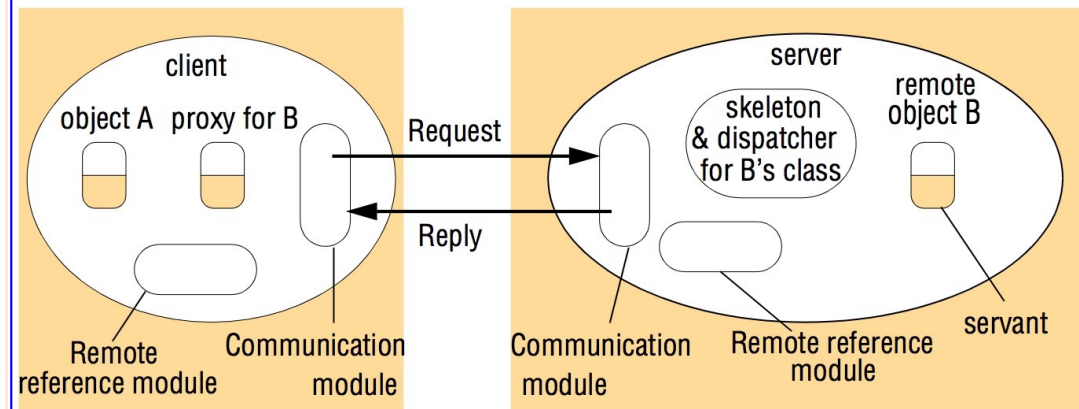
V.1.1 (c) 2013-2021 José María Foces Morán & José María Foces Vivancos



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] &`



© 2012 Prentice Hall Kindberg Blair Dollimore Distributed Systems

Process to write a simple Java RMI example

4. Write the Java RMI Server that will instantiate and start the remote object (servant)

```
SDRemoteObjectImpl cc = new SDRemoteObjectImpl(60002);
```

```
Naming.rebind("leon", cc);
```

...

- The server may need to set some properties, depending on the deployment context
- For full remote access over the Internet:

```
$ java  
-Djava.rmi.server.hostname=193.146.101.46  
-Djava.rmi.server.codebase=http://193.146.101.46/dsrmipract  
dsrmipract.Server  
leon
```

Process to write a simple Java RMI example

5. Write the RMI Client

- Must have the remote interface accessible
- Must be given a java security policy file that allows it to do dynamic class loading
- Locate rmiregistry:

```
Registry r = LocateRegistry.getRegistry("paloalto.unileon.es", 60001);
```

- Lookup servant named "leon":

```
ServantRemoteInterface cc = (ServantRemoteInterface) r.lookup("leon");
```

- Invoke the remote methods:

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

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