Distributed Applications – session 5

Lecturer

Mouhsen Ibrahim

Contents

- Java Packages and Interfaces
- Remote Procedure Call (RPC)
- Remote Method Invocation (RMI)
- RMI Application Structure
- Working of an RMI Application
- Marshaling, unmarshaling and RMI registry
- RMI registry

Java Packages

- A package is a set of classes, interfaces and types contained in a folder.
- Packages provide access protection and namespace management.
- Classes with similar functionality are grouped in a single package.
- java.net for network operations, java.io for IO operations.
- java.lang fundamental classes.

Java Interfaces

- An interface describes the behavior of a class.
- Interfaces are implemented by classes.
- Interfaces contain method declarations, static and final attributes but not instance attributes.
- A non-abstract class which implements an interface nust implement all the methods of the interface.
- An interface cannot be instantiated.
- An interface can extend multiple interfaces.

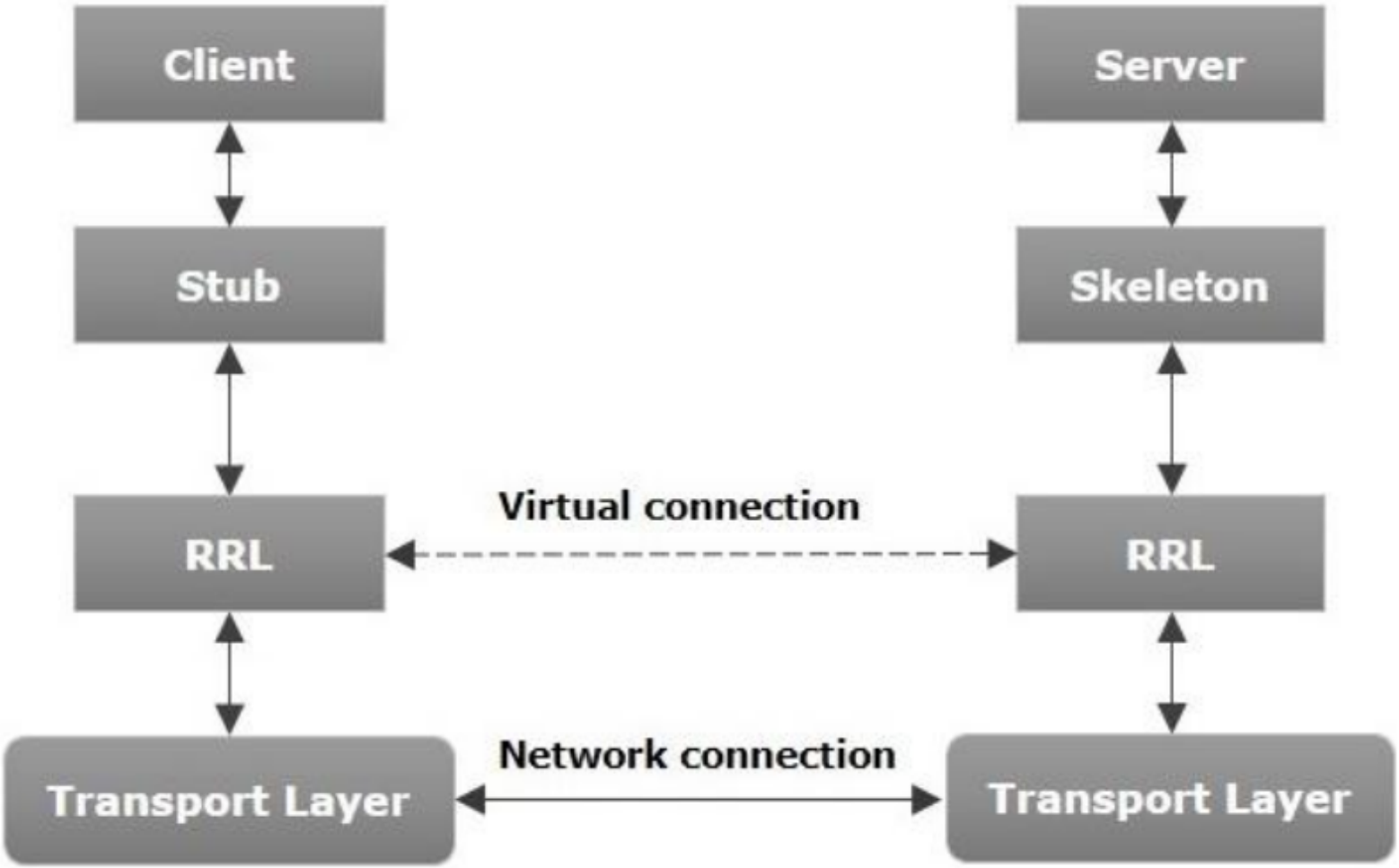
Remote Procedure Call (RPC)

- RPC is a protocol that allows one program on one computer to request a service from a program running on another computer.
- It uses the client/server model for communication.
- RFC 5531 documents the protocol, it can be found here https://tools.ietf.org/html/rfc5531
- There are many different implementations for RPC
- Open Software Foundation's Distributed Computing Environment DCE.
- Java Remote Method Invocation RMI.

Remote Method Invocation (RMI)

- It is a mechanism to allow one object in a system (JVM) to access another object in another system (JVM).
- It can be used to build distributed applications and provides remote communication between java programs.
- In RMI we create a server program which creates remote objects and make them available to the client via registry
- The client program accesses remote objects on the server and call their methods.

RMI Application Structure



RMI Application Structure

- Transport Layer: Connects client and server, manages existing connections and sets up new connections.
- Stub: It resides at the client program and acts as a proxy for accessing remote objects.
- Skeleton: It resides at the server side, the stub communicates with it to access remote objects.
- RRL (Remote Reference Layer): It is the layer that manages the references made by the client to remote objects.

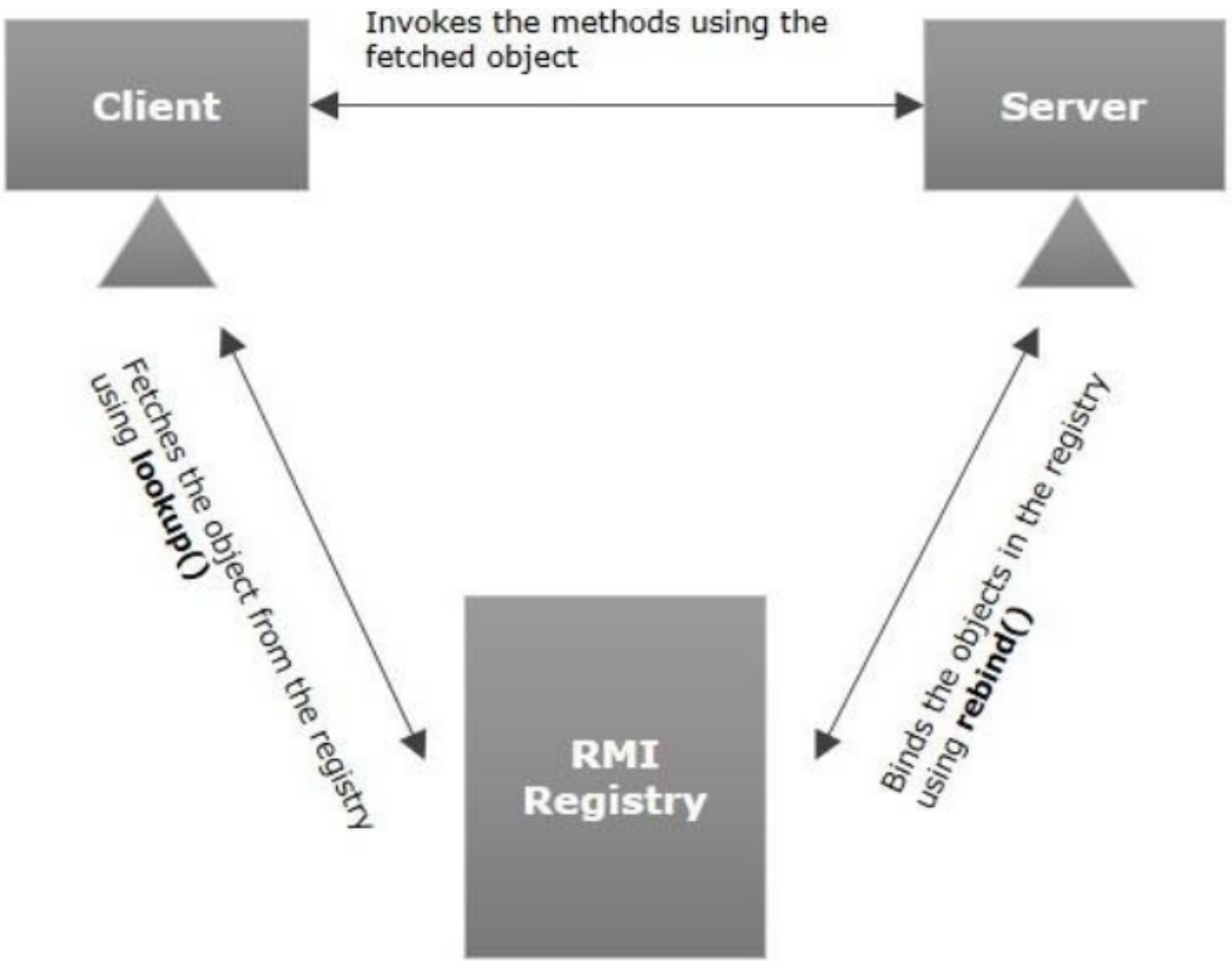
Working of an RMI Application

- When the client calls a method on a remote object, it is passed to the stub at the client which sends it to RRL at client side.
- The RRL at client side receives the request and calls invoke() method of object remoteRef. It passes the request to RRL on server side.
- The RRL on server side receives the request and passes it to the skeleton which finally calls the required method on the server.
- The result is passed back all the way to client.

Marshaling and unmarshaling

- Marshaling is the process of sending parameters to remote methods, in case of primitive data types they are included in the message and a header is attached and in case of objects they are serialized.
- Unmarshaling is the the reverse process at the server side.
- RMI Registry is a namespace on the server for all remote objects.
- The server creates a remote object using bind() method and gives it a unique name.
- The client gets a reference for the remote object using lookup() method.

RMI registry



Java Object Serialization

- An object can be serialized and stored on disk using Java's ObjectOutputStream.
- The class of the object must implement the java.io.Serializable interface to be serialized.
- Use the writeObject() method to write an object to the stream.
- An object can be deserialized from a file stored on disk using Java's ObjectInputStream.
- Use the readObject() method to read an object from the stream.

Serialization Example

- Write a program to serialize an Object of the Client class defined in the previous lecture for chat application and save it to a file.
- Write a program to deserialize the previous object from the file on disk.
- Write a program to serialize and deserialize an array of Client objects.

GOOD LUCK