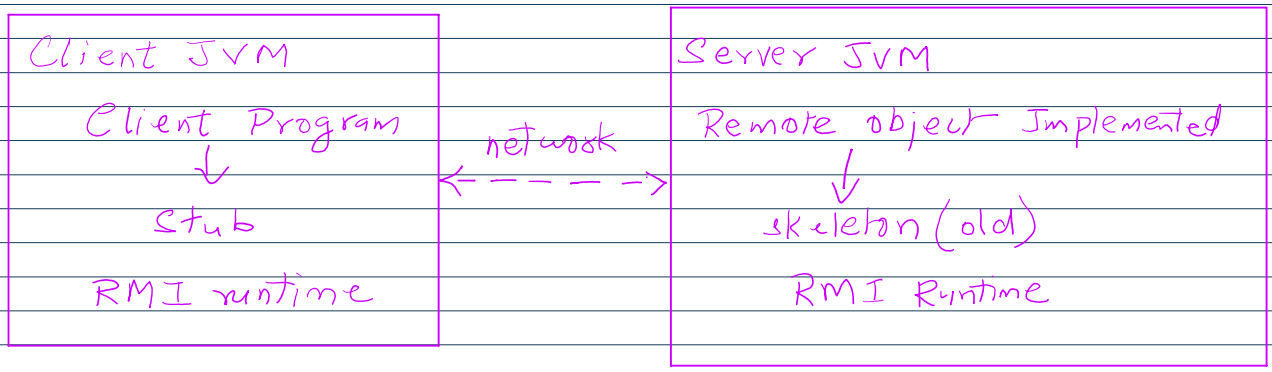
# RMI

16 April 2025 20:52

What is RMI (Remote Method Invocation)?
Java Remote Method Invocation(RMI) is an Java API that performs a mechanism to allow the invocation of methods on remote objects.

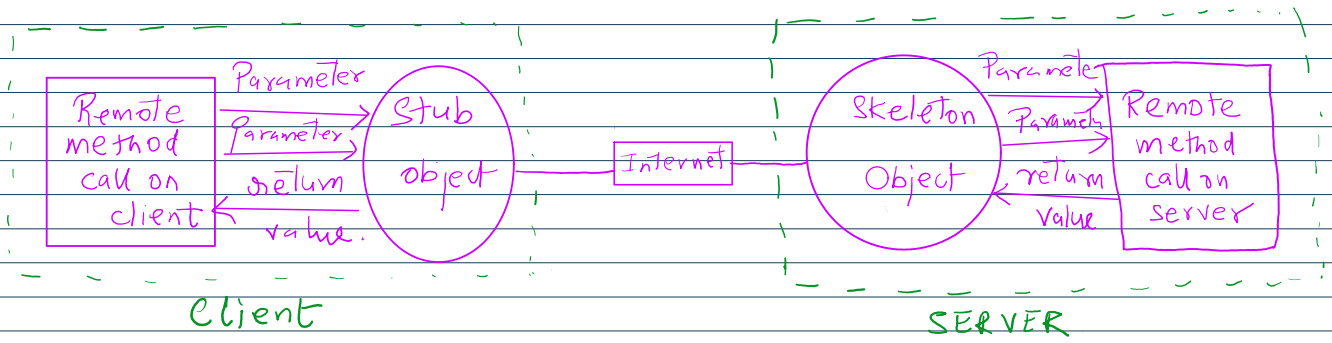
Definition:
Java RMI allows an object running in one Java Virtual Machine (JVM) to invoke methods on an object running in another JVM. This is Java's native way to implement it, distributed object oriented computing. It abstracts complex network communication and serialization, making remote communication as Seamless as local method calls.




Stub: Client-side proxy for the remote object, Handles marshaling (packing data), sending request, and unmarshalling (interpreting responses).
In other words, a stub is a local object that acts as a proxy for a remote object when a client wants to invoke a method on a remote object. It does so through the stub.

Skeleton: Server side component (prior to Java, 1.2.) today. the server side object handles request directly.
In other words, skeleton is used to receive method invocation from clients and forward them to the remote object.

RMI Registry: Acts like a phone book maps. name to remote object.




Stub: Information block → Server
- ID of remote object
- Method name to invoke
- Parameter to pass.

Skeleton.
- calls desired method
- forwards the parameter
- return value to Stub object.

Goals of RMI
(i) Minimize complexity of the Application.

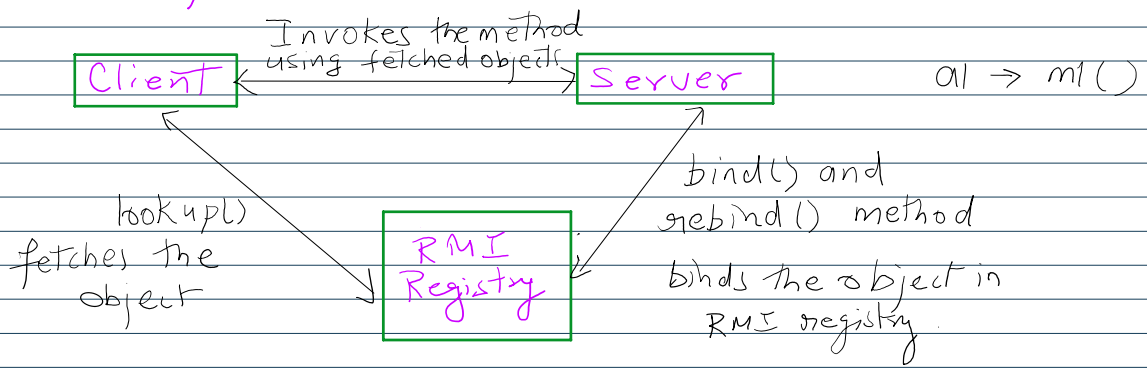
Goals of RMI

(i) Minimize complexity of the Application.

(ii) Distributed Garbage collection.

(iii) Minimizes difference between working with local & remote objects.

RMI Registry : It is a namespace on which all server objects are placed.




**How RMI works?**

1. **Client makes a method call on a remote interface.**
2. **The stub object:**
   a. **Marshall's method parameters into bytes.**
   b. **Sends data to the server using the network.**
3. **On the server, the RMI runtime:**
   a. **Receives the call and unmarshal the parameters.**
   b. **Invokes the actual method on the server object.**
4. **The method executes on the server and returns a result.**
5. **The stub receives the return value and gives it to the client program.**

```
package RMI;
import java.rmi.Remote;
import java.rmi.RemoteException;
// remote interface
public interface Calculator extends Remote {
    int add(int a, int b) throws RemoteException;
}
```

```
package RMI;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
// implementing Remote Object
public class CalculatorImp extends UnicastRemoteObject implements Calculator{
    protected CalculatorImp() throws RemoteException{
        super();
    }
    public int add(int a, int b){
        return a+b;
    }
}
```

```
package RMI;
import java.rmi.registry.LocateRegistry;
```

```java
import java.rmi.registry.Registry;
public class CalculatorServer {
    public static void main(String[] args) {
        try {
            Calculator skeleton = new CalculatorImp();
            Registry registry = LocateRegistry.createRegistry(1099); //Start RMI registry in code
            registry.rebind("CalcService", skleton);
            System.out.println("Server is running...");
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
}
```

```java
package RMI;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class CalculatorClient {
    public static void main(String[] args) {
        try {
            Registry registry = LocateRegistry.getRegistry("localhost", 1099);
            Calculator stub = (Calculator)registry.lookup("CalcService");
            System.out.println(" 15 + 5 = "+stub.add(15, 5));
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
}
```

Remote: Marker interface tells. Java that method can be called remotely.
UnicastRemoteObject: Helps export the object to receive incoming calls.
Stub: Auto generated or internal proxy to contact remote object.
Skeleton: Legacy server side proxy now internal.
Registry: Allows client to look up remote objects by name.