JAC444 - BTP400 Course Object-Oriented Software Development II - Java

Remote Method Invocation

Segment 1

Remote Method Invocation

In this lesson you will be learning about:

- Designing RMI application
- Developing distributed object defined by RMI interfaces
- Designing and developing RMI Server
- Designing and developing a RMI Client
- Deploying and running the RMI system

Building Calculator RMI System

1. Design and implement Java RMI Calculator interfaces
2. Develop Java code implementing classes defined by RMI Calculator interfaces
3. Develop code for Java RMI Calculator server
4. Develop code for Java RMI Calculator client program
5. Install and run RMI Calculator system

Naming Remote Objects

<u>How does a client find an RMI remote service?</u>

- RMI System includes a simple service called the RMI registry:

  rmiregistry

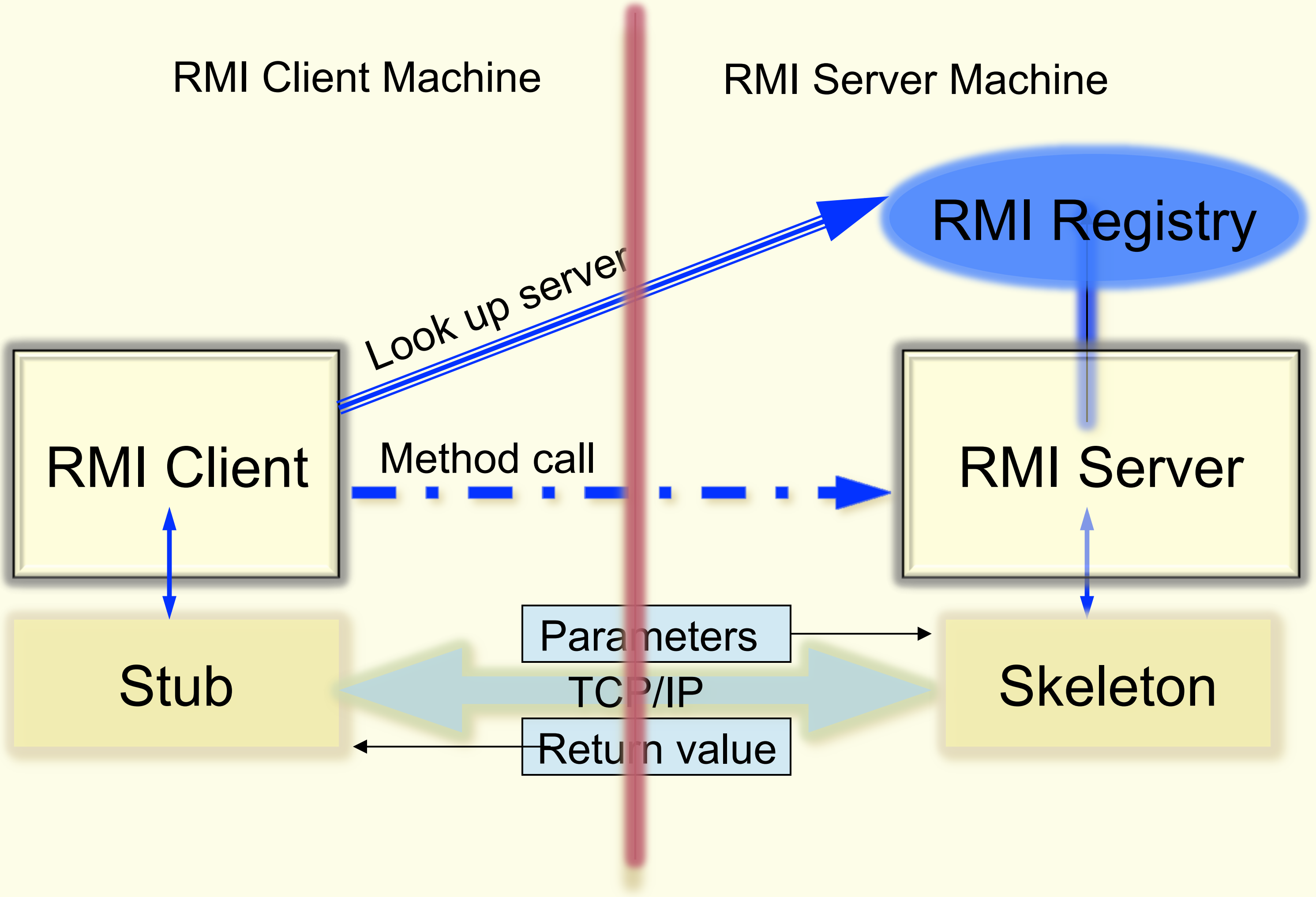
- On a server machine, a server program creates a remote service and register it in the RMI registry.

- On the client side, the RMI Registry is accessed through the class ***Naming***. The static method ***lookup(String url)*** is a method a client uses it to query a registry.

- The method returns a remote reference to the service object. The URL parameter of a lookup method takes the form:

rmi://<host_name>[:<service_port>]/<service_name>

Overview of RMI



1

Calculator Interfaces

- Interface defines all of the remote features offered by the server – Calculator.java

```
public interface Calculator extends java.rmi.Remote {

      public long add(long a, long b)
                  throws java.rmi.RemoteException;

      public long sub(long a, long b)
                  throws java.rmi.RemoteException;

      public long mul(long a, long b)
                  throws java.rmi.RemoteException;

      public long div(long a, long b)
                  throws java.rmi.RemoteException;
 }
```

2

Calculator Implementation Class

- The implementation of the interface for the remote service. ***CalculatorImpl.java***

```
public class CalculatorImpl extends java.rmi.server.UnicastRemoteObject
                                    implements Calculator {

        // Implementations must have an explicit constructor
        // in order to declare the RemoteException exception
        public CalculatorImpl() throws java.rmi.RemoteException {
              super();
        }

        public long add(long a, long b)throws java.rmi.RemoteException {
              return a + b;
        }
        …..
   }
```

3

Calculator RMI Server

- The class *CalculatorServer.java* is a very simple server that provides the bare essentials for hosting

```
import java.rmi.Naming;

 public class CalculatorServer {

   public CalculatorServer() {
     try {
        Calculator c = new CalculatorImpl();
        Naming.rebind("rmi://localhost:1099/CalculatorService", c);
     } catch (Exception e) {
        System.out.println("Trouble: " + e);
     }
   }

   public static void main(String args[]) {
         new CalculatorServer();
   }
 }
```

Calculator RMI Client

- RMI Client: CalculatorClient.java

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.net.MalformedURLException;
import java.rmi.NotBoundException;

public class CalculatorClient {

    public static void main(String[] args) {
      try {
         Calculator c =
         (Calculator)Naming.lookup("rmi://localhost/CalculatorService");

              System.out.println( c.sub(4, 3) );

         } catch (MalformedURLException murle) {
              System.out.println(murle);
         } catch (RemoteException re) {
              System.out.println(re);
         } catch (NotBoundException nbe) {
              System.out.println(nbe);
         }
    }
}
```

5 Running Calculator RMI System

- Start with the Registry. You must be in the directory that contains the classes you have written. From there, enter the following

```
rmiregistry
```

- Start the RMI calculator server hosting the Calculator service

```
java CalculatorServer
```

- Start the RMI calculator client program

```
java CalculatorClient
```

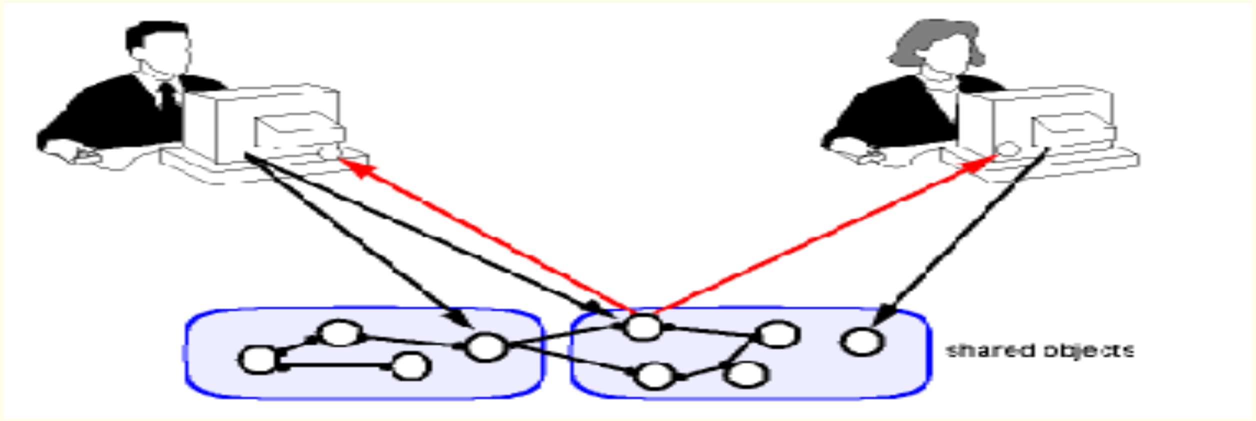
RMI over IIOP

Common Object Request Broker Architecture CORBA

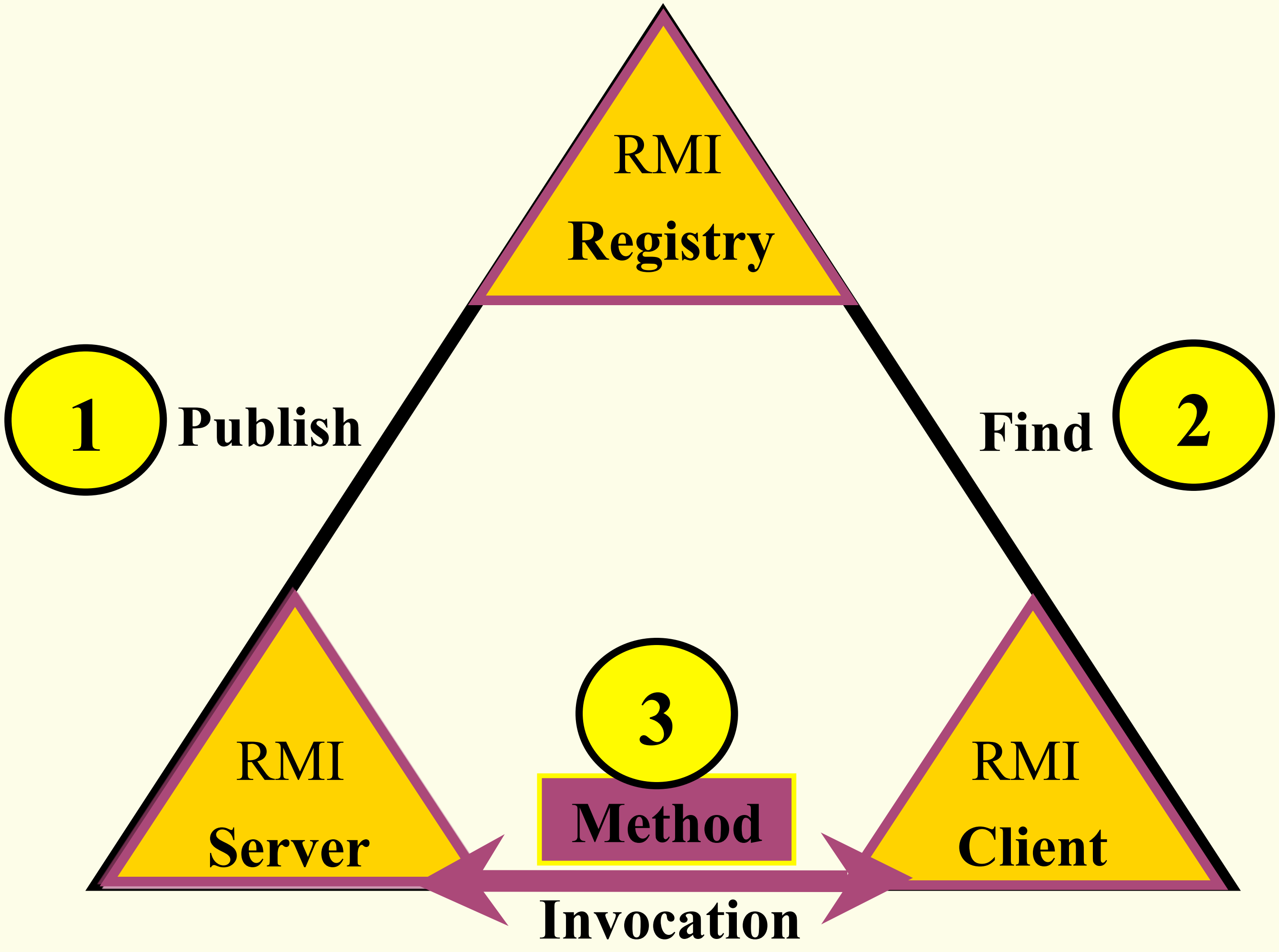
Remote Method Invocation (RMI)

over Internet Inter-Orb Protocol (IIOP)

access distributed objects on the Internet



Run RMI System



Conclusions

After completion of this lesson you should know:

- How to design distributed applications using RMI.
- How to write Java RMI programs.
- How to deploy applications using RMI tools.