University of Science and Technology of Hanoi

Distributed System

Practical Work II

# RPC

*Author:*
Nguyen Duc Dan
Nguyen Tri Huan
Nguyen Huu Chi Dat
Pham Minh Giang
Nguyen Xuan Duy Anh

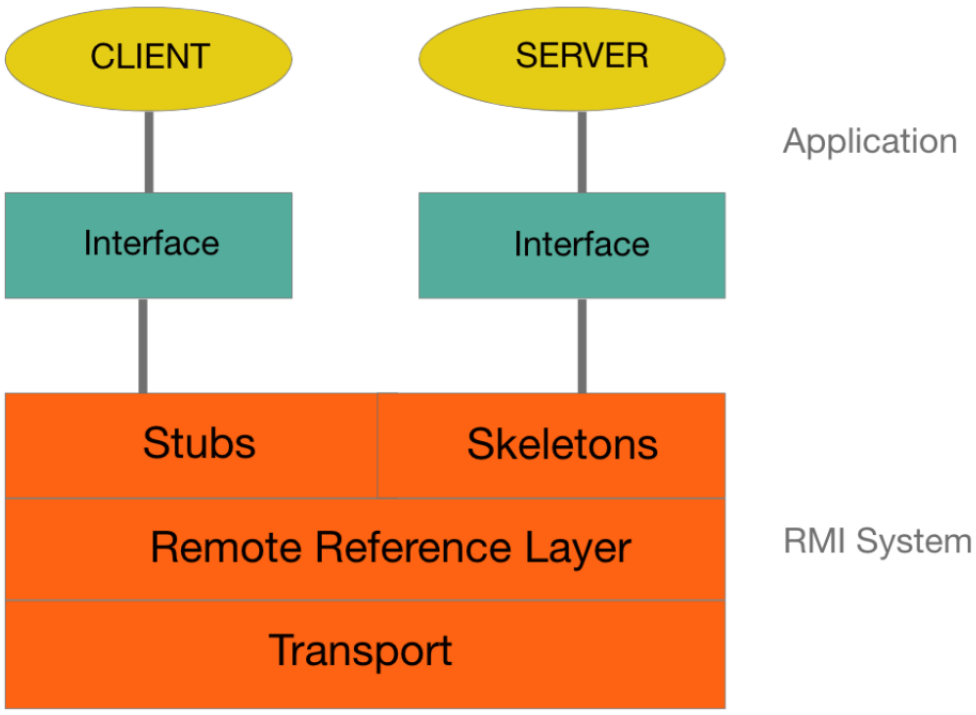
*Lecturer:*
Dr. Tran Giang Son

March 18, 2020

# Contents

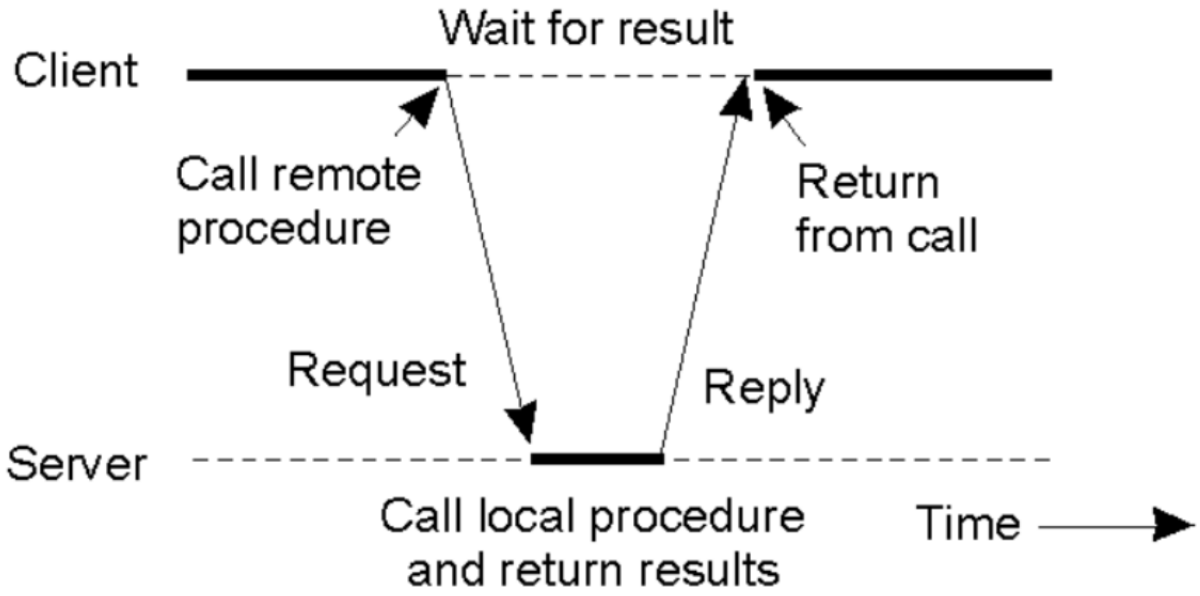
# 1 Design RPC service

**Figure** We use Java RMI:




# 2 System Organizing

**Figure**




# 3 Implement the file transfer

1. Define a remote interface
2. Implement the remote interface
3. Develop the server
4. Develop a client
5. Generate Stubs and Skeletons, start the RMI registry, server, and client

# 4 Code

## 4.1 FileClient

```
import java.io.*;
import java.rmi.*;

public class FileClient{
public static void main(String[]  argv) {
if(argv.length != 2) {
System.out.println("Usage: java FileClient
   fileName machineName");
System.exit(0);
}
try {
String name = "//" + argv[1] + "/FileServer";
FileInterface fi = (FileInterface)
   Naming.lookup(name);
byte[] filedata = fi.downloadFile(argv[0]);
File file = new File(argv[0]);
BufferedOutputStream output = new
BufferedOutputStream(new
   FileOutputStream(file.getName()));
FileOutputStream fr = new
   FileOutputStream("C:\\Users\\danng\\IdeaProjects\\ds2020\\l
fr.write(filedata, 0, filedata.length);
output.write(filedata,0,filedata.length);
output.flush();
output.close();
} catch(Exception e) {
System.err.println("FileServer exception: "+
   e.getMessage());
e.printStackTrace();
      }
   }
}
```

## 4.2 FileImpl

```
import java.io.*;
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;
```

```
public class FileImpl extends
   UnicastRemoteObject
implements FileInterface {

private String name;

public FileImpl(String s) throws
   RemoteException{
super();
name = s;
}

public byte[] downloadFile(String fileName){
try {
File file = new File(fileName);
byte buffer[] = new byte[(int)file.length()];
BufferedInputStream input = new
BufferedInputStream(new
   FileInputStream(fileName));
input.read(buffer,0,buffer.length);
input.close();
return(buffer);
} catch(Exception e){
System.out.println("FileImpl:
   "+e.getMessage());
e.printStackTrace();
return(null);
      }
   }
}
```

### 4.3 FileInterface

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface FileInterface extends Remote
```

```
   {
public byte[] downloadFile(String fileName)
   throws
RemoteException;
}
```

### 4.4 FileServer

```
import java.io.*;
import java.rmi.*;

public class FileServer {
public static void main(String argv[]) {
if(System.getSecurityManager() == null) {
System.setSecurityManager(new
   RMISecurityManager());
}
try {
FileInterface fi = new FileImpl("policy.txt");
Naming.rebind("//127.0.0.1/FileServer", fi);
} catch(Exception e) {
System.out.println("FileServer:
   "+e.getMessage());
e.printStackTrace();
      }
   }
}
```

## 5 Group participation

- Nguyen Duc Dan - BI8028 : Complete and implement the code
- Nguyen Xuan Duy Anh - BI8014 : Design RPC service
- Nguyen Tri Huan - BI8069 : System Organizing
- Pham Minh Giang - BI8054 - : Implement the file transfer
- Nguyen Huu Chi Dat - BI8040: Summarize and Write the Report