NYIT

**Fall 2021**

Homework No: 14

Title: Remote Procedure Call and Distributed Computing Environment DCE

**Name:** Patade, Yash Arun

**Class ID#:**  11

**School ID#:** 1284979

**Course:** Java Networking

**Course ID:** CSCI 725

**Date:** 12/12/2021

Assignment Contents

[Question: RPC 3](#_Toc90764607)

[1) What software engineering principle does RPC use demonstrates? 3](#_Toc90764608)

[2) Identify reused software on the distributed application architecture model shown below. 3](#_Toc90764609)

[3) Remote Method Invocation. 4](#_Toc90764610)

[Definition 4](#_Toc90764611)

[4) Why not just use sockets? 4](#_Toc90764612)

[5) Advantages of RMI. 5](#_Toc90764613)

[6) Java-RMI vs. CORBA and DCOM. 5](#_Toc90764614)

[7) What does RPC help distribute and what does RMI help distribute? 6](#_Toc90764615)

[8) Object Serialization Example 1. 7](#_Toc90764616)

[9) Object Serialization Example 2. 8](#_Toc90764617)

[References 11](#_Toc90764618)

# Question: RPC

# 1) What software engineering principle does RPC use demonstrates?

Answer) Development based on reuse.

Graphical user interface, application

Description automatically generated

# 2) Identify reused software on the distributed application architecture model shown below.

Graphical user interface, application

Description automatically generated

Answer)

• RPC1 and RPC2

• API1 and API2 ,

• OS Transport Services A and B, (OS-A and OS-B)

# 3) Remote Method Invocation.

Definition: Remote Method Invocation or RMI is a true distributed object oriented computing application interface for Java, written to provide easy access to objects existing on remote hosts (In remote memory spaces), so easy as accessing local objects.

– RMI provides virtual locality, virtual local presence of remote objects with remote methods.

• Objects truly exist in the remote memory space, but appear as existing in the local memory space.

– RMI allow passing objects as parameters to remote methods

# 4) Why not just use sockets?

• Designing distributed application by separately designing one module (Client) than the other (Server) using Sockets OOP API (e.g., java.net.\* package) is extremely tedious (Involved, demanding, network presence/technology aware.

• Programmer must know:

– When to use Sockets API,

– Where to use server socket and where just socket,

– Plan, expect and be ready to handle all possible exceptions, etc.

• Networking aware distributed application development is error prone for implementing complex application-level protocols and hard to introduce even small modification.

• General SE idea is to standardize most of the involved code that demands costly expert work and so make it reusable. – Standardize for reusability!

# 5) Advantages of RMI.

• Object-Oriented

• Safe and Secure

• Easy to Write/Easy to Use

• Connects to Existing/Legacy Systems (JNI)

• Write Once, Run Anywhere

• Distributed Garbage Collection

• Parallel Computing

# 6) Java-RMI vs. CORBA and DCOM.

• Java-RMI is a Java-specific middleware spec that allows client Java programs to invoke server Java objects as if they were local.

• Java-RMI is tightly coupled with the Java language.

• Since Java-RMI is tightly coupled with The Java Language, Java-RMI can work with true sub-classes.

• Because of this, parameters passed during method calls between machines can be true Java Objects. This is impossible in DCOM or CORBA at present.

• If a process in an RMI system receives an object of a class that it has never seen before, it can request that its class information/specs/description be sent over the network.

• Over and above all this, Java-RMI supports Distributed Garbage Collection that ties into the local Garbage Collectors in each JVM.

– Efficient heap memory space cleaning and free space maximizing.

# 7) What does RPC help distribute and what does RMI help distribute?

Answer)

• RPC helps distribute C-language functions.

• RMI helps distribute Java objects.

Diagram

Description automatically generated

# 8) Object Serialization Example 1.

Text

Description automatically generated

Output:

Graphical user interface, text, application

Description automatically generated

# 9) Object Serialization Example 2.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Output:

Text

Description automatically generated

# References

[1] https://www.geeksforgeeks.org/serialization-in-java/