P15/37561/2016

KYALO KELVIN MUINDI

DISTRIBUTED SYSTEMS (ASSIGNMENT 2)

a).JAVA RMI:

The Java Remote Method Invocation (Java RMI) is a [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) [API](https://en.wikipedia.org/wiki/Application_programming_interface) that performs [remote method invocation](https://en.wikipedia.org/wiki/Remote_method_invocation), the object-oriented equivalent of [remote procedure calls](https://en.wikipedia.org/wiki/Remote_procedure_call) (RPC), with support for direct transfer of [serialized](https://en.wikipedia.org/wiki/Serialization#Java) Java classes and collection. Its system allows an object running in one Java Virtual Machine (VM) to invoke methods of an object running in another Java VM. RMI provides for remote communication between programs written in the Java programming language.

RMI provides the mechanism by which the server and the client communicate and pass information back and forth. It treats a remote object differently from a non-remote object when the object is passed from one Java virtual machine to another Java virtual machine. Rather than making a copy of the implementation object in the receiving Java virtual machine, RMI passes a remote ***stub*** for a remote object.

b).DCOM

Distributed Component Object Model (DCOM) is a proprietary Microsoft technology that allows Component Object Model (COM) software to communicate across a network. DCOM is enhanced with COM applications to facilitate remote procedural calls and a Distributed Computing Environment (DCE) dedicated to Windows application and platform support.

DCOM can also work on a network within an enterprise or on other networks besides the public Internet. It uses [TCP/IP](https://searchnetworking.techtarget.com/definition/TCP-IP) and [Hypertext Transfer Protocol](https://searchwindevelopment.techtarget.com/definition/HTTP) . DCOM comes as part of the Windows operating systems. It supports remote objects by running on a protocol called **Object Remote Procedure Call** (ORPC). A DCOM client calls into the exposed methods of a DCOM server by acquiring a pointer to one of the server object's interfaces. The client object then starts calling the server object's exposed methods through the acquired interface pointer as if the server object resided in the client's address space

References:

* [*"Java RMI Release Notes"*](http://docs.oracle.com/javase/1.5.0/docs/guide/rmi/relnotes.html)*. Oracle. Retrieved 9 May 2012.*
* Starr Andersen, Technical Writer; Vincent Abella, Technical Editor “[DCOM Security Enhancements](https://technet.microsoft.com/en-us/library/bb457156.aspx#EIAA)”- August 09, 2004
* Ann Wollrath; Roger Riggs; [*Jim Waldo*](https://en.wikipedia.org/wiki/Jim_Waldo). [*"A Distributed Object Model for the Java System"*](http://pdos.csail.mit.edu/6.824/papers/waldo-rmi.pdf) *(PDF). Retrieved 2009-02-11*.