Distributed Systems (H0N08A)

Report: –

Andi Škrgat *(r0876363)*
Team Partner *(r1234567)*

October 16, 2021

What is the role of stub and skeleton in Java RMI? Do the other remote communication technologies have similar concepts?

What is the role of stub and skeleton in Java RMI? Do the other remote communication technologies have similar concepts? Stub is an object which resides on client side. It acts as gateway for all outgoing requests from client. Stub first initiates a connection with remote Virtual Machine and after that marshals the parameters to the remote Virtual Machine. Then it waits for the result, unmarshals the return value or exception and returns result to the caller. Skeleton in Java RMI is object which reads parameters for some remote method, calls method on remote object and later marshals the result to the caller. Sun RPC also uses stub functions, client stub on client side and server stub on server side. Their functionalities are same as in Java RMI(marhsaling, creating packages for transferring over the networkin and unamarshaling) with the difference that server stub executes a local procedure call to the actual server function. In CORBA a stub is a mechanism which issues requests as a client and skeleton delivers them to the CORBA object implementation. Based on these examples we can say some remote communication technology always has to have some similar mechanism for communication between client and server.

2. What is the difference between serializable and remote? In your Java RMI assignment, which classes were made serializable and which classes were made remote? Motivate your choice.

First difference is that Remote is class and Serializable is interface. Remote interface serves to identify which methods can be called from a non local machine. Only methods from object which extended Remote class are visible and can be accessed remotely. Serializable is used for all objects which are transferred as arguments from client to server or as a result from server to client. Serializable is only BookingDetail because this is the only class whose instance is transferred as argument in addBooking method. IBookingManager is Remote so it "publishes" its methods to the outside. Every method from object which extends Remote must throw RemoteException. In Java RMI all primitive types and all serializable objects are passed by value while remote objects are passed by reference.

3. What role does the RMI registry plays? Why is there no registry in the other remote communication technologies?

RMI registry is the binder for Java RMI and it is used as name service. It stores mappings between names and references to remote objects which are hosted on compute. RMI registry must be run on server. There are two main methods for registering: bind and rebind and the only difference is that rebind can be called even if there is an object mapped to concrete name. Clients then need to use lookup function from Registry so they would be able to call right method from object registered on RMI registry. Other remote techonologies don't use naming registry because they work differently. For example gRPC searches for IP address.