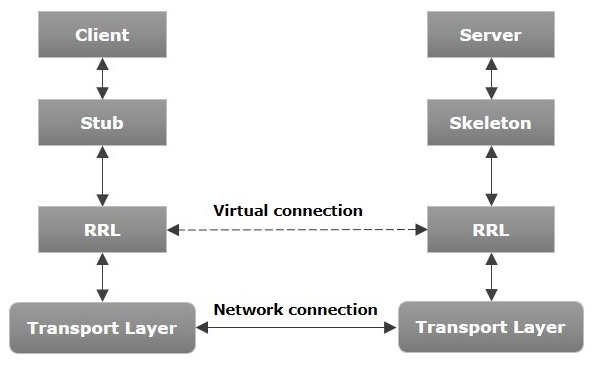
RMI = Remote Method Invocation

Distributing object-oriented programming

[](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.tutorialspoint.com%2Fjava_rmi%2Fjava_rmi_introduction.htm&psig=AOvVaw0pQxRwAHqcscKCnvCZNe4A&ust=1589351672353000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCEssLarekCFQAAAAAdAAAAABAD)

* The client calls its own proxy (stub) using a local method call
* The proxy (stub) in the client communicates with the proxy in the server, transmitting the required data with a byte encoding (parameter marshalling)
* The proxy (skeleton) in the server makes a standard method call and receives the return value
* The proxy (skeleton) in the server sends to the proxy in the client (stub) the data from the previous point
* The proxy in the client (stub) returns the result of the method invocation locally

RRL – Remote Reference Layer

RMI involves two types of data being sent:

* Parameters of the invoked method
* Returned values, thrown exceptions

Defining a type:

* Interface of the type “TRemote” – resides on both client and server
* Actual implementation “TRemoteImpl” – resides on the server