

Assignment - 1A

my companion 43304

- * Title → Distributed applications using Java sockets and RMI.
- * Problem statement → To develop any distributed application through implementing client-server communication programs based on Java sockets.
- * Theory →

- Socket:

In distributed computing networks communication is one of the essential parts of any system, and the socket is end point of every instance of network communication.

Server port \leftarrow connection \rightarrow Client port

- RMI:

The Remote Method Invocation is an API that provides a mechanism to create distributed applications in Java.

The RMI allows an object to invoke methods on an object running in another JVM.

- Java API:

Java provides reliable stream based communication for TCP as well as the unreliable datagram communication for UDP.

Java input/output streams allow application programs to input and output various data, such as bytes, string, file, etc. The Java socket API provides the basis of TCP/UDP connection.

- TCP:

It stands for Transmission Control Protocol, which allows for reliable communication between two applications.

TCP is typically used over internet protocol, which is referred to as TCP/IP.



UDP:

It stands for User Datagram Protocol, a connection-less protocol that allows for packets of data to be transmitted between applications.

Types of sockets:

1. Stream Sockets - Allows processes to communicate using TCP. A stream socket provides bidirectional, reliable, sequenced and unduplicated flow of data with no record boundaries. After the connection has been established, data can be read from ~~the~~ and written to these sockets as a byte-stream. The socket type is SOCK-STREAM.

2. Datagram Sockets - Allows processes to use UDP to communicate. A datagram socket supports bi-directional flow of message. A process on datagram socket can receive messages in different order from sending sequence and can receive duplicate message. Record boundaries in data are preserved. The socket type is SOCK-DGRAM.

3. Raw Sockets - Provides access to ICMP. These sockets are normally datagram oriented, although their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not for most applications. Only super user processes can use raw sockets. The socket type is SOCK-RAW.

How RMI works:

1. When the client makes a call to remote object, it is received by the stub which eventually passes request to RRL.

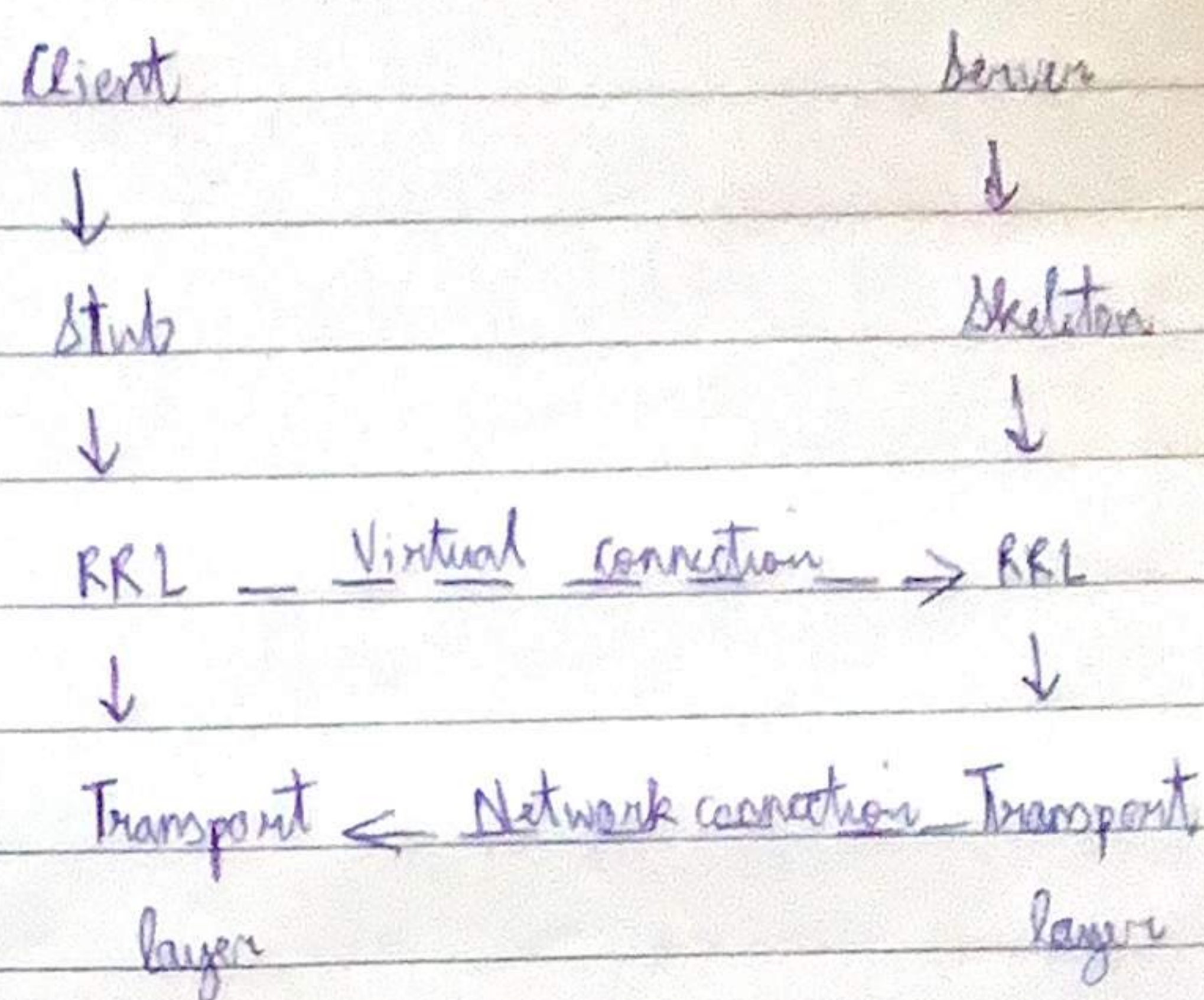
2. When the client side RRL receives the request, it invokes a method called `invoke()` of the object remote reference. It passes the request to RRL on the server side.

3. The RRL on the server side passes the request to the skeleton (proxy) on the server, which finally invokes the required object.

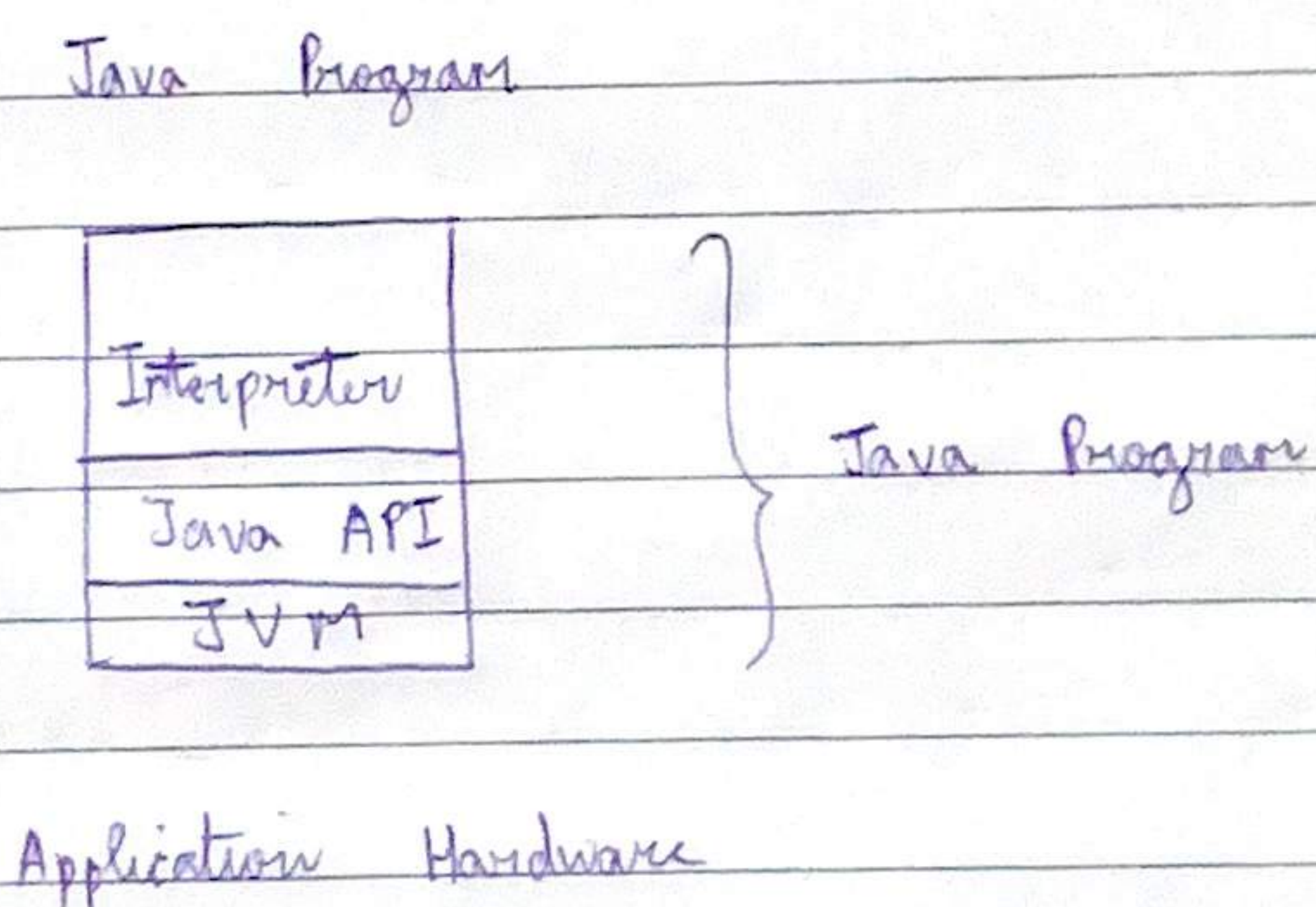


4. The result is passed all the way back to the client.

• Architecture of RMI :



• Java API :



* Conclusion → Thus in this assignment we have learnt about distributed applications, and developed a client-server program based on Java sockets.