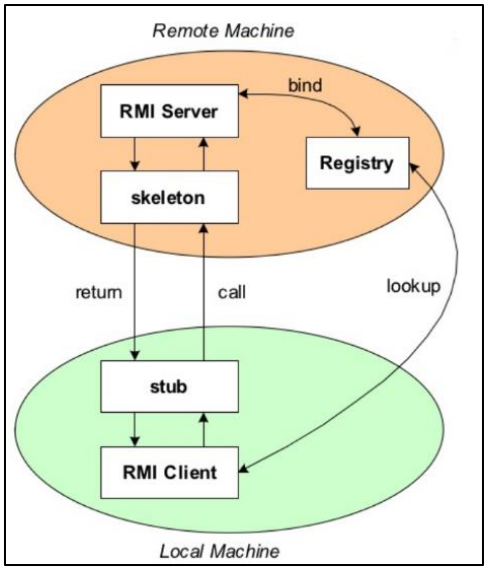
This programming assignment provides a client with useful fuel mileage information pertaining to their vehicle and a possible trip they may take. The application will ask the user about the fuel economy of their vehicle, along with the distance of the trip, and the amount of fuel in the vehicle. With this information, the server will determine how much fuel will be left inside the vehicle, and the server will also determine how far the vehicle can possibly travel to the destination. Additionally, a small diagram will be shown, demonstrating how far the vehicle will get it to the destination.



The underlying architecture of the application makes use of a remote machine which consists of a Java RMI server, RMI registry, and skeleton. The local machine contains the Java RMI client, and stub. The communication between the two machines is handled via remote method invocation – the object-oriented equivalent of remote procedure calls (RPC). The program is able to handle continuous client requests until it is manually terminated. The results of the trip can be logged and downloaded to the client in a text file for retrieval later on.

Though, the project consists of some challenges which were overcome. One of the issues was how to handle writing to the log file as well as downloading it to the client. This was solved by handling both operations in separate methods and then handling the download receipt on the client. Another small problem that occurred was how to handle a user event when they are prompted to log the results to a text file. This was completed by handling only the user's verdict (keyboard input) on the client and calling the remote methods if needed. A major roadblock was determining when to close certain input/output streams; the client would throw an exception and not allow for continuous running of the program. It was later learned that streams can only be closed once in an instance of an application – thus, the streams were closed after the client desired to exit.