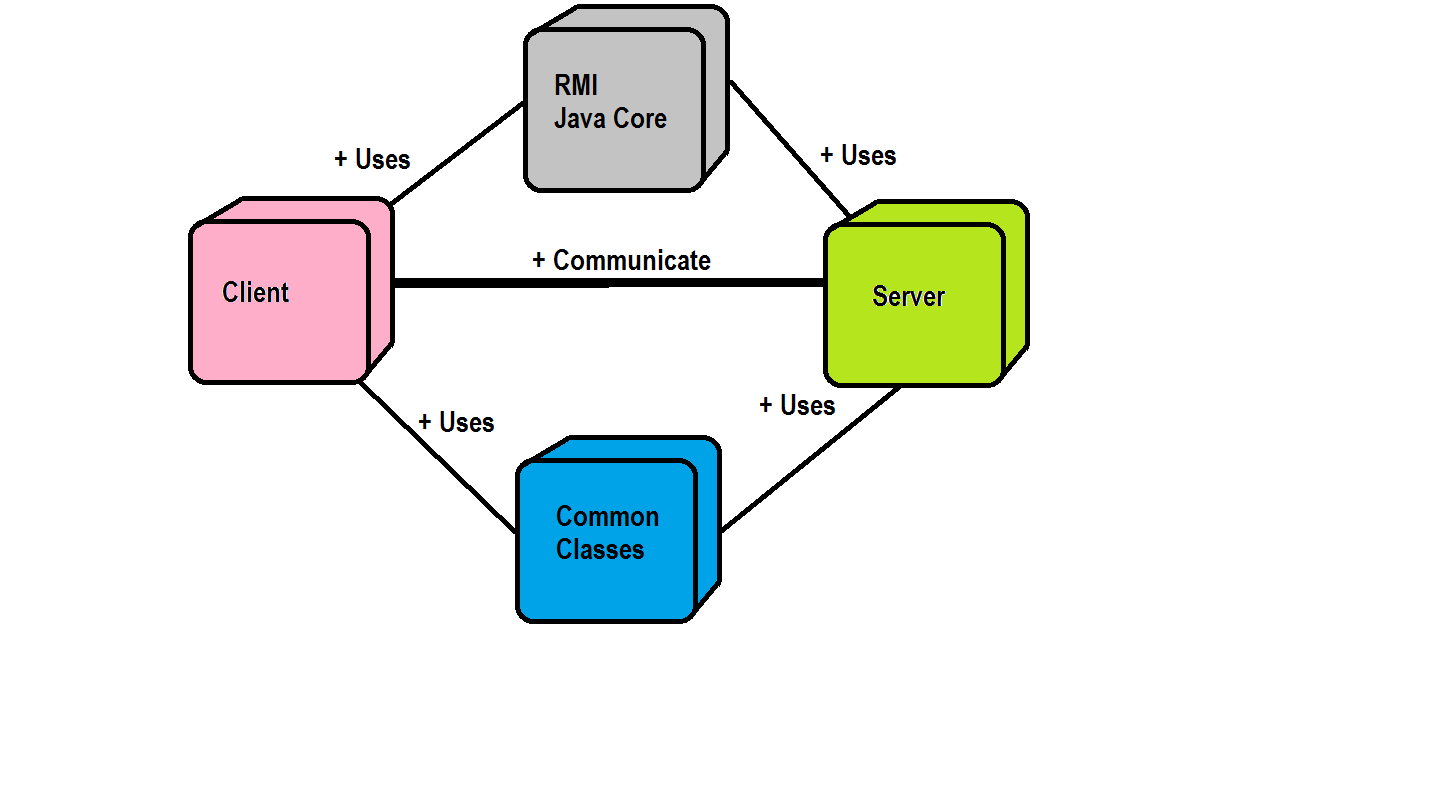
Solution description document **- Assignment 2.2**

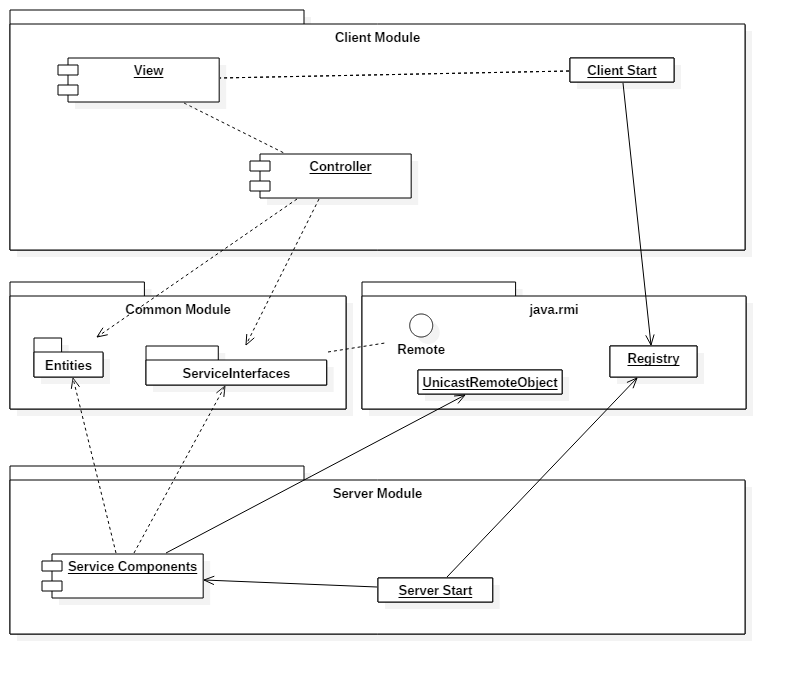
**1. Conceptual architecture of the distributed system**

The solution is implemented in 3 different modules. Besides these we are using one more module, called RMI which comes with Java Core.

****

* The **RMI** (Remote Method Invocation) is an API that provides a mechanism to create distributed application in java. The RMI allows an object to invoke methods on an object running in another JVM. The RMI provides remote communication between the applications using two objects *stub* and *skeleton*.
* **The Common Classes** – contain two packages: "entities" - contains the entitiy (Car) and "services" - contains the definition of the interface exposed by the remote object (Skeleton)
* **The Server Application** – contains two packages: "communication" - contains the server-side communication and "services" – contains the implementation of the remote object
* **The Client Application** (MVC)- contains one package "communication" with one class which contains the main method, package "view" which contains the home view of client application and package "controller" - the input mechanism of the client interface

**2. Deployment diagram**



**3. Readme**

Git repository: <https://bscridon@bitbucket.org/bscridon/ds2016_30643_scridon_beniamin.git>

Build:

* build as java application
* IDE(ex: Eclipse, Intellij etc.)
* JDK 1.8

Runtime Environment:

* JRE 1.8

Run Steps:

* run the server
* run the client

4. Bibliography

1.Java RMI: [https://docs.oracle.com/javase/tutorial/rmi/](https://docs.oracle.com/javase/tutorial/rmi/%20)