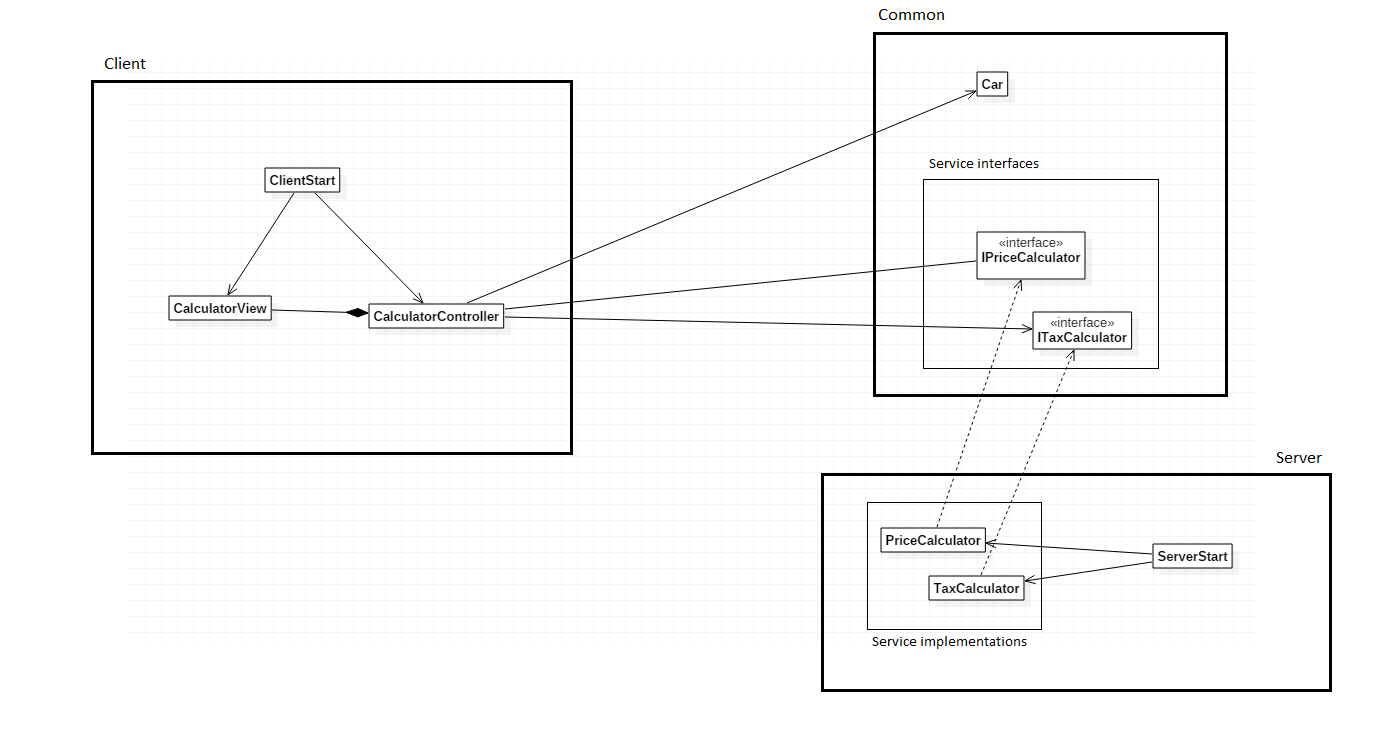
DISTRIBUTED SYSTEMS

Assignment 2

Remote Procedure Call (RPC)

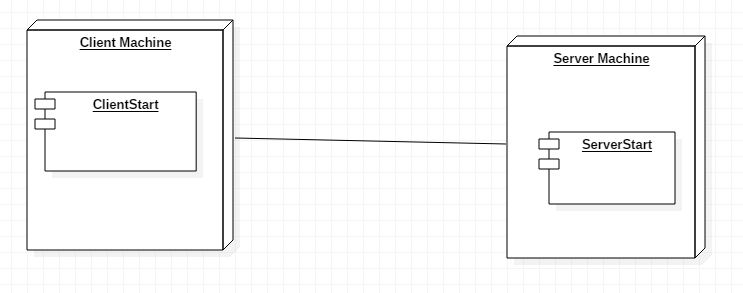
A2.2: RPC application using distributed objects (Java RMI)

* Conceptual architecture of the distributed system:

The distributed system is organized in three logical components: Server, Common and Client. The Common component contains the parts that are (and should be) identical in both the Client and the Server application, including the interfaces of the distributed objects and the entity. The Server component contains the implementations of the remote services and the code needed to start the server. The Client component contains the code needed to connect to the server, access the distributed objects and display the graphical user interface.

* UML Deployment diagram

The deployment diagram is pretty simple as we only have two machines each running a single instance: the server program, which creates the distributed objects and adds them to the registry to be used remotely; and the client program which retrieves a proxy of each distributed object from the registry and uses it to achieve the program’s goals. The user interface of the client is created using the Model-View-Controller (MVC) architectural pattern. In the current setup, all the components are expected to be on the same machine, however, with the minimal appropriate changes, the setup presented in the diagram can be achieved.

* Build and execution considerations:
  1. Open IntelliJ IDEA
  2. File | Open
  3. Browse and select the directory containing the project
  4. Click “Open”
  5. Navigate to the “ServerStart” class from the “server” module
  6. Run | Run ‘ServerStart’
  7. Navigate to the “ClientStart” class from the “client” module
  8. Run | Run “ClientStart”
  9. A window representing the user interface should appear, use it to input data for the calculations.