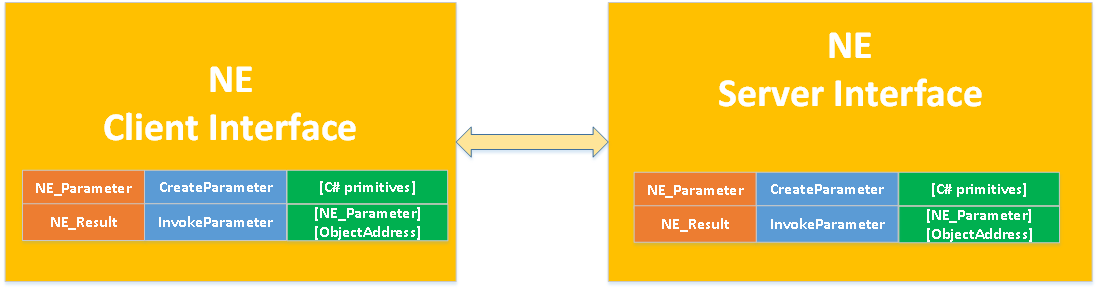
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Our idea is to create a high level language Interface between Host Application and **Interface Engine**. Language of choice is as there are many LUA interpreters written in C#. We will be picking **Nlua** from the below list because the documentation is straight forward.  We will be picking a one of our previous WinForms assignment as dummy Host Application and integrate it with our **NE\_Server\_Interface** classes to expose it’s the address of the objects. To connect to our Client Application to Host Application, we will be using **Windows Communication Foundation [WCF]** configured **to Duplex Contract**, **Net Named Pipe** binding.  **NE\_Server\_Interface** and **NE\_Client\_Interface** will be the heart of our construct.    Since **WCF** only takes in defined method signatures, object methods cannot be exposed dynamically. So a handful functions capable to dynamically invoke methods in objects will be used. For now consider the following methods used **for NE\_Interfaces [NE\_Client/Server\_Interface]**:   |  |  |  | | --- | --- | --- | | **Return Type** | **Method** | **Parameters** | | NE\_Parameter | CreateParameter | (C# primitives) | | NE\_Result | InvokeParameter | (NE\_Parameter,ObjectAddress) |   These methods suggest that it is appropriate to create Host Application objects only in Host Application space and not in Client Application space. To be precise, Client Application ideally should only be handling C# Primitives as Common Assemblies between applications. It will then be possible at our convenience to exchange small packets of data of non-deep or shallow objects through **WCF**. |



|  |  |
| --- | --- |
| **C# LUA Interpreters** | **Description** |
| MoonSharp | A Lua interpreter written entirely in C# for the .NET, Mono and Unity platforms |
| http://www.moonsharp.org/ |
| Nlua | NLua is the bind between Lua world and the .NET world |
| http://nlua.org/ |
| LuaInterface | LuaInterface is a library for integration between the Lua language and Microsoft .NET platform's Common Language Runtime (CLR). Lua scripts can use it to instantiate CLR objects, access properties, call methods, and even handle events with Lua functions. |
| http://luaforge.net/projects/luainterface/ |

