.NET Remoting

What is “Remoting”

- Communicate between objects not in the same process / domain
  - E.g. different applications on same computer
  - E.g. different applications on different computers
- Similar to Java RMI
  - Uses remote objects
  - Hides implementation
  - Utilizes platform-specific implementations

What is a Remote Object?

- Any object outside the application domain.
- Facilitates communication between domains
- Objects parsed remotely has to be serialized
  - Local objects not serializable are “non remotable”

Why use remote objects?

- To communicate between different domains
- Hides the difficult parts of handling
  - Connections & Sockets
  - Marshalling (serialization of objects)
  - Reading/writing XML (or similar)

How to make objects remotable?

An object can be made remotable by either:

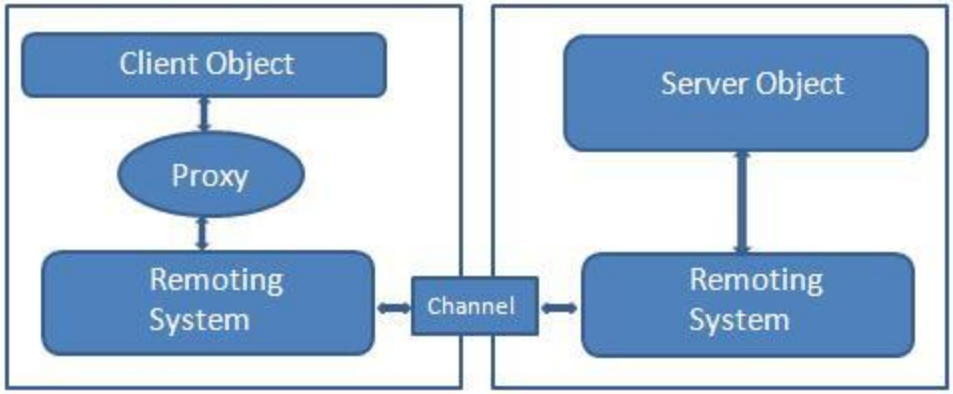
- Adding the “Serializable” tag
- Implement the ISerializable interface
- **Derive from MarshalByRefObject class**

How does it work?

Two objects:

1. Proxy object
2. Remote object

By invoking a method on the proxy, the call is forwarded to the remote object.



Source:
http://csharp.net-informations.com/remoting/csharp-remoting-architecture.htm

Overview of sample client/server

We want to create a **simple** client & server.

Consists of three parts:

- Remoteable object: “Echo”
  - Should extend “MarshalByRefObject”
- Server
  - Create a TcpChannel and register the channel
  - Register the remoteable object “Echo”
- Client
  - Create a TcpChannel and register the channel
  - Retrieve the remoteable object “Echo”

Additional Reading

1. https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-netod/bfd49902-36d7-4479-bf75-a2431bd99039
2. https://docs.microsoft.com/en-us/dotnet/framework/wcf/migrating-from-net-remoting-to-wcf