Remoting

What distributed process frameworks outside .NET do you know?
Distributed Computing Environment/Remote Procedure Calls (DEC/RPC), Microsoft Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), and Java Remote Method Invocation (RMI).

What are possible implementations of distributed applications in .NET?
.NET Remoting and ASP.NET Web Services. If we talk about the Framework Class Library, noteworthy classes are in System.Runtime.Remoting and System.Web.Services.

When would you use .NET Remoting and when Web services?
Use remoting for more efficient exchange of information when you control both ends of the application. Use Web services for open-protocol-based information exchange when you are just a client or a server with the other end belonging to someone else.

What's a proxy of the server object in .NET Remoting?
It's a fake copy of the server object that resides on the client side and behaves as if it was the server. It handles the communication between real server object and the client object. This process is also known as marshaling.

What are remotable objects in .NET Remoting?
Remotable objects are the objects that can be marshaled across the application domains. You can marshal by value, where a deep copy of the object is created and then passed to the receiver. You can also marshal by reference, where just a reference to an existing object is passed.

What are channels in .NET Remoting?
Channels represent the objects that transfer the other serialized objects from one application domain to another and from one computer to another, as well as one process to another on the same box. A channel must exist before an object can be transferred.

What security measures exist for .NET Remoting in System.Runtime.Remoting?
None. Security should be taken care of at the application level. Cryptography and other security techniques can be applied at application or server level.

What is a formatter?
A formatter is an object that is responsible for encoding and serializing data into messages on one end, and deserializing and decoding messages into data on the other end.

Choosing between HTTP and TCP for protocols and Binary and SOAP for formatters, what are the trade-offs?
Binary over TCP is the most effiecient, SOAP over HTTP is the most interoperable.

What's SingleCall activation mode used for?
If the server object is instantiated for responding to just one single request, the request should be made in SingleCall mode.

What's Singleton activation mode?
A single object is instantiated regardless of the number of clients accessing it. Lifetime of this object is determined by lifetime lease.

How do you define the lease of the object?
By implementing ILease interface when writing the class code.

Can you configure a .NET Remoting object via XML file?
Yes, via machine.config and application level .config file (or web.config in ASP.NET). Application-level XML settings take precedence over machine.config.

How can you automatically generate interface for the remotable object in .NET with Microsoft tools?
Use the Soapsuds tool.

What are CAO's i.e. Client Activated Objects ?
Client-activated objects are objects whose lifetimes are controlled by the calling application domain, just as they would be if the object were local to the client. With client activation, a round trip to the server occurs when the client tries to create an instance of the server object, and the client proxy is created using an object reference (ObjRef) obtained on return from the

Nagaraj .NET

creation of the remote object on the server. Each time a client creates an instance of a client-activated type, that instance will service only that particular reference in that particular client until its lease expires and its memory is recycled. If a calling application domain creates two new instances of the remote type, each of the client references will invoke only the particular instance in the server application domain from which the reference was returned.
In COM, clients hold an object in memory by holding a reference to it. When the last client releases its last reference, the object can delete itself. Client activation provides the same client control over the server object's lifetime, but without the complexity of maintaining references or the constant pinging to confirm the continued existence of the server or client. Instead, client-activated objects use lifetime leases to determine how long they should continue to exist. When a client creates a remote object, it can specify a default length of time that the object should exist. If the remote object reaches its default lifetime limit, it contacts the client to ask whether it should continue to exist, and if so, for how much longer. If the client is not currently available, a default time is also specified for how long the server object should wait while trying to contact the client before marking itself for garbage collection. The client might even request an indefinite default lifetime, effectively preventing the remote object from ever being recycled until the server application domain is torn down. The difference between this and a server-activated indefinite lifetime is that an indefinite server-activated object will serve all client requests for that type, whereas the client-activated instances serve only the client and the reference that was responsible for their creation. For more information, see Lifetime Leases.
To create an instance of a client-activated type, clients either configure their application programmatically (or using a configuration file) and call new (New in Visual Basic), or they pass the remote object's configuration in a call to Activator.CreateInstance. The following code example shows such a call, assuming a TcpChannel has been registered to listen on port 8080.

How many processes can listen on a single TCP/IP port?
One.

What technology enables out-of-proc communication in .NET?
Most usually Remoting;.NET remoting enables client applications to use objects in other processes on the same computer or on any other computer available on its network.While you could implement an out-of-proc component in any number of other ways, someone using the term almost always means Remoting.

How can objects in two diff. App Doimains communicate with each other?
.Net framework provides various ways to communicate with objects in different app domains.
First is XML Web Service on internet, its good method because it is built using HTTP protocol and SOAP formatting.
If the performance is the main concern then go for second option which is .Net remoting because it gives you the option of using binary encoding and the default TcpChannel, which offers the best interprocess communication performance
What is the difference between .Net Remoting and Web Services?
Although we can develop an application using both technologies, each of them has its distinct advantages. Yes you can look at them in terms of performance but you need to consider your need first. There are many other factors such authentications, authorizing in process that need to be considered.

Point	Remoting	Webservices
If your application needs interoperability with other platforms or operating systems	No	Yes, Choose Web Services because it is more flexible in that they are support SOAP.
If performance is the main requirement with security	You should use the TCP channel and the binary formatter	No
Complex Programming	Yes	No
State Management	Supports a range of state management, depending on what object lifetime scheme you choose (single call or singleton call).	Its stateless service management (does not inherently correlate multiple calls from the same user)
Transport Protocol	It can access through TCP or HTTP channel.	It can be access only through HTTP channel.

Nagaraj .NET