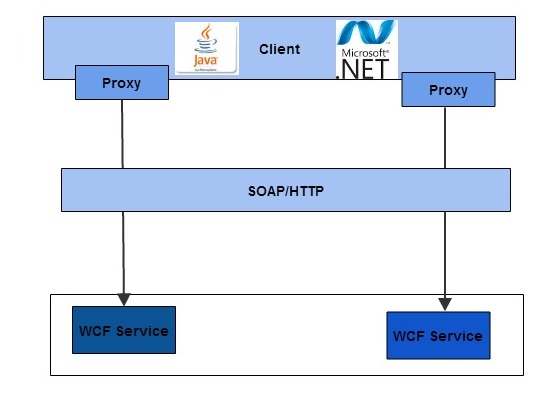
**WCF’s SOA Implementation**

****

**Rameshkartik.RS**

**Table of Contents**

[Introduction 2](#_Toc386275661)

[Why WCF has SOA implementation? 3](#_Toc386275662)

[How WCF implements SOA Patterns? 4](#_Toc386275663)

[Summary 8](#_Toc386275664)

**WHAT’s IN THIS ARTicle?**

* Why WCF has SOA’s Implementation?
* How WCF implements SOA’s patterns?
* Summary

# 

# Introduction

There is always a strong problem statement stays behind the screen whatever the invention happens in this world. For the WCF as well you can find some problem statement behind the invention of WCF. Let's discuss it. Also, you can find some SOA patterns, how it is used in the WCF Services.

# Why WCF has SOA implementation?

Actually the WCF team at Microsoft was trying to deliver by considering the following main points

* Interoperability across platforms
* Unification of existing technologies
* Service oriented programming

Interoperability across platforms – Microsoft wants the application running on windows should be communicated with legacy applications, MAC OS X Machines, Linux machines, windows clients, solaris machines and anyone else who abides by the WS-I (Web services interoperability Organization Specification).

Unification of existing technologies - WCF takes all the capabilities of the

Distributed technologies and overlay a simplified API called System. Service Model,

In which you can find all things of ASMX,WSE,System.Messaging,. Net Remoting

In one roof of WCF.

Service Oriented Programming – To increase the flexibility in the programming

Ease the business orientation of modern software projects, WCF takes us

Service oriented programming from the object oriented programming

# How WCF implements SOA Patterns?

**SOA Design Patterns**

Establishing a robust service oriented architecture requires the business to take

In the account of Design Patterns, Security and API management. SOA design

Patterns allow organizations to solve their problem by using the proven solutions.

SOA patterns have the solutions for the commonly recurring problems within

the enterprise. The implementation environment for SOA design patterns must

allow loose coupling and reuse of integration solutions. We will see in WCF how

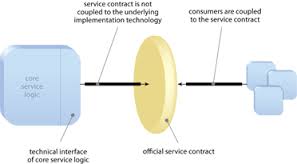
these patterns are used and why?

**Decoupled Contract**

The reason why this ‘Decoupled Contract’ has been invented is

“How can a service, express its capabilities independently of its technical implementation?”

In other words, To provide an effective enterprise service, the technical contract should be completely independent from its logical implementation which is yet to align with the other services. So only the interface details will be visible to the client not its implementation.



Diagram

By Decoupling the service contract ,the service implementation can be evolved without directly impacting service consumers, This can increase the amount of refactoring opportunities and reusability

**Proxy Pattern**

Proxy pattern is used to provide a surrogate object, which references to the other object. Also called, it is a proxy class, it represents the functionality of another class.

public interface iAction

{

void PerformAction();

}

public class ServiceClass : iAction

{

public void PerformAction()

{

Console.WriteLine("ServiceClass action performed.");

}

}

public class Proxy : iAction

{

private ServiceClass \_servClass;

public void PerformAction()

{

if (\_servClass == null)

\_servClass = new ServiceClass();

\_servClass.PerformAction();

}

}

Let's discuss how the proxy pattern is applied in the WCF. In WCF as soon we create a service reference for a wcf service, a proxy object will be created.

WCF provides the ClientBase<T> for the proxy classes to inherit, The base class has the capability to set up the communication and other operation calls. The consumers send a call against the proxy for which the SOAP messages will be send them to the service using defined binding. AS I mentioned in the above source code example, the proxy object provides a surrogate object servClass which represents the functionality of another class called ServiceClass. The same concept being used in the WCF Service and the proxy.

**Operation Context Pattern**

**It decouples the functional input parameters** from the technical information the method needs to execute. WCF provides a method called System.ServiceModel.OperationContext , this class will take over the responsibility of providing information about the call,session id,incoming message header in the SOAP envelope,and information about the identity of the caller. The context contains lots of information about the non functional and therefore it should not be the part of the DataContract. When working with the duplex mode, operation context provides the channel to be used to call back the client during the method execution. Also the operation context provides an additional context called WebOperationcontext that provides the method a more information about the request in terms of HTTP properties.

**Concurrent contracts**

WCF supports concurrent contracts by implementing multiple interfaces in a

system by allowing the service to have a multiple end points configured. We

will see these topics in much detail on my upcoming articles.

**Data Confidentiality**

WCF supports cross platform security by using the WS-\* protocols stack for

message level security. This level of security we can define in selected binding.

We will see these topics in much detail on my upcoming articles.

**Exception Shielding**

WCF automatically shields the exceptions occurred in and it does not show

those details to the client. When an exception occurs in the service, the client

Is only informed about something, went wrong not the exception details. This

is not only done for a security perspective ,the potential hackers will take advantage

If the service shows the exception details like inner exception and stack trace.

Also, it doesn’t make sense if the service provides the exception details for the

client because the client may not be a .Net Client,it would be a Java client.

WCF supports carrying the information about the exceptions in a SOAP fault. Transporting the SOAP fault and deserializing/serializing the content is

defined by the SOAP standard and is cross platform and more SOA oriented.

We will see the related topics like FAULT Contract in detail on my upcoming articles.

# Summary

The main aim of WCF is to create interoperability across platforms,

unification among the legacy systems. Also the WCF supports the SOA patterns

Decoupled, Façade, proxy, operational context patterns to define the system

as SOA oriented.