Web services

Windows Communication Foundation (WCF)

Th.S Cao Thái Phương Thanh
caothaiphuongthanh@gmail.com

WCF

- Introduction
- WCF Elements
- WCF simple tutorial
- WCF implementation of Purchase Order

WCF – Introduction

- WCF – Windows Communication Foundation
  - "The unified programming model for building service-oriented applications on the Windows"
  - Interoperability
    - Broad support for WS – specifications
    - Compatible with existing distributed application techonogies
  - Productivity
    - Visual Studio integration
    - Unify today's distributed technologies
  - Servicer oriented development

WCF – Introduction

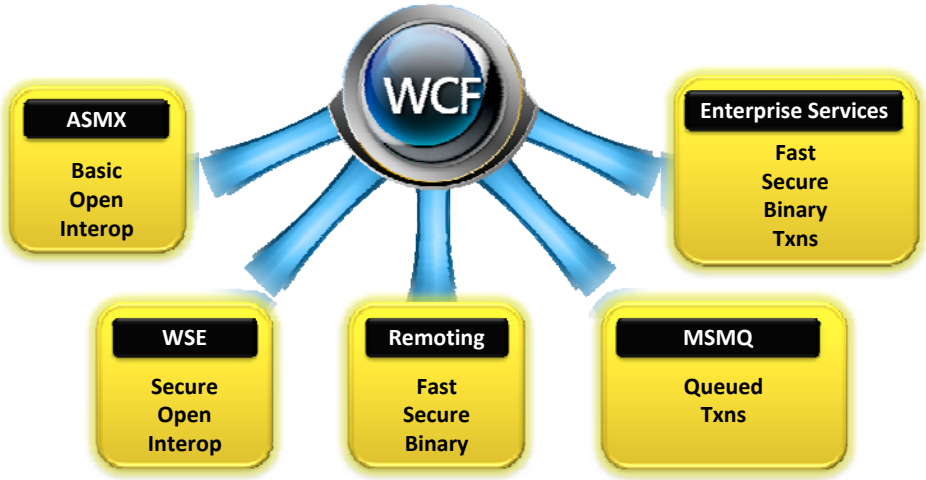
- WCF – Windows Communication Foundation
  - "The unified programming model for building service-oriented applications on the Windows"
  - Interoperability
    - Broad support for WS – specifications
    - Compatible with existing distributed application techonogies
  - Productivity
    - Visual Studio integration
    - Unify today's distributed technologies
  - Servicer oriented development

WCF – Introduction

- Before WCF, we have:
  - Remote Procedure Call
  - COM+
  - .NET Remoting
  - ASP.net web service
  - Hard code TCP Socket
  - MSMQ
  - …

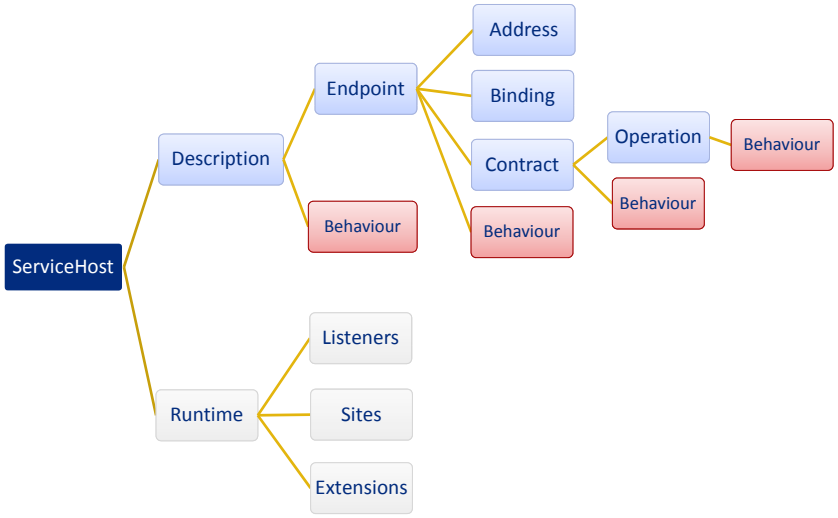
WCF – Introduction

- Now WCF unifies programming models

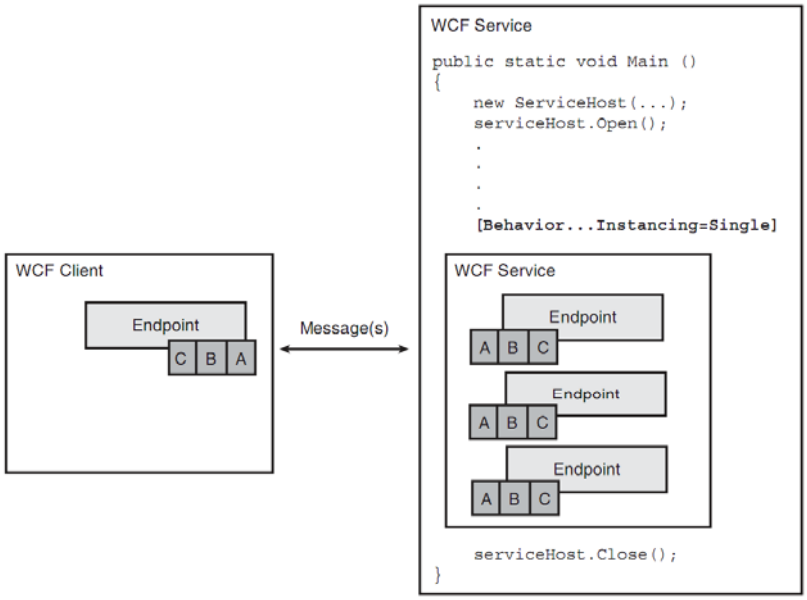


WCF – Introduction

- WCF elements

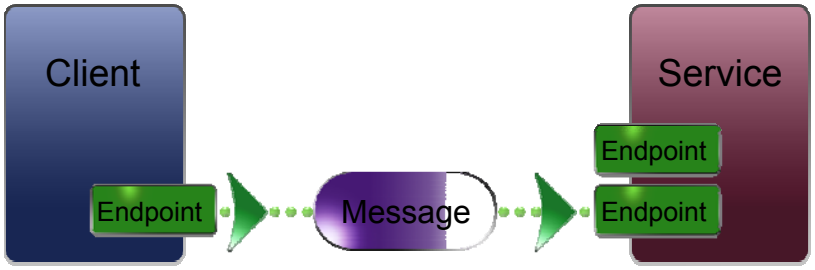


WCF – Elements

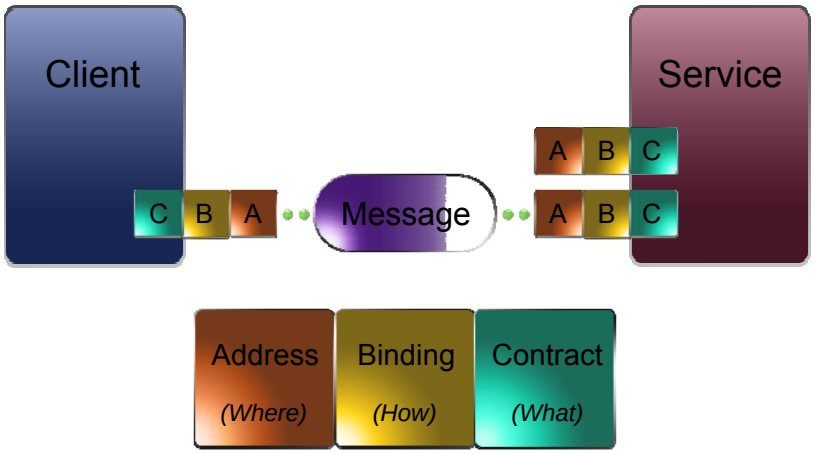


WCF – Elements

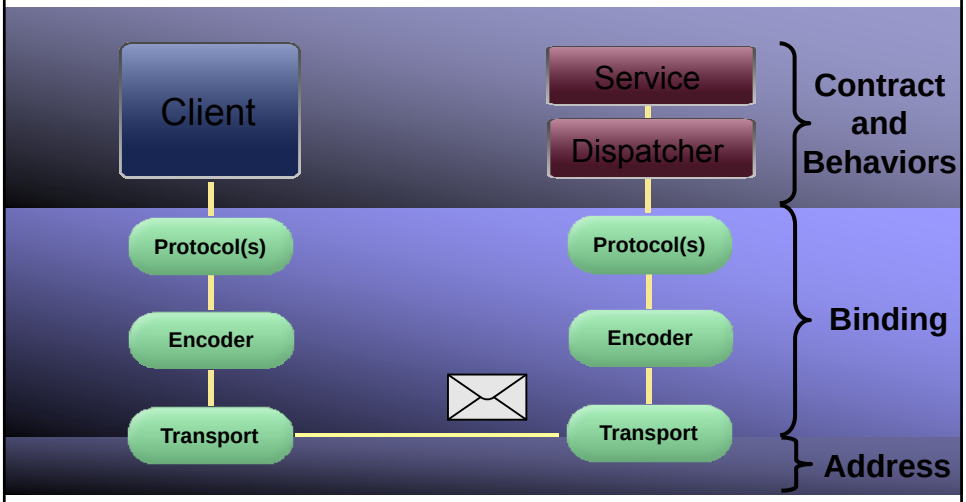
- Endpoint: the service endpoint contains the information about the Address, Binding, Contract, and Behavior required by a client to find and interact with the service at this endpoint.



WCF – Elements



WCF – Elements



WCF – Elements

- Address: where the service is
  - Combination of transport, server name, port and path.
    - http://server:345/Service
    - net.tcp://localhost:8002/MyService
    - net.pipe://localhost/MyPipe
    - net.msmq://localhost/private/MyService
  - Transport is determine by the binding

WCF – Elements

- Binding: how to talk to the service
  - Transport: HTTP, TCP, MSMQ
  - Message format and encoding
    - Plain text
    - Binary
    - Message Transmission Optimization Mechanism (MTOM)
  - Communication security
    - No security
    - Transport security
    - Message security
    - Authentication and authorizing callers

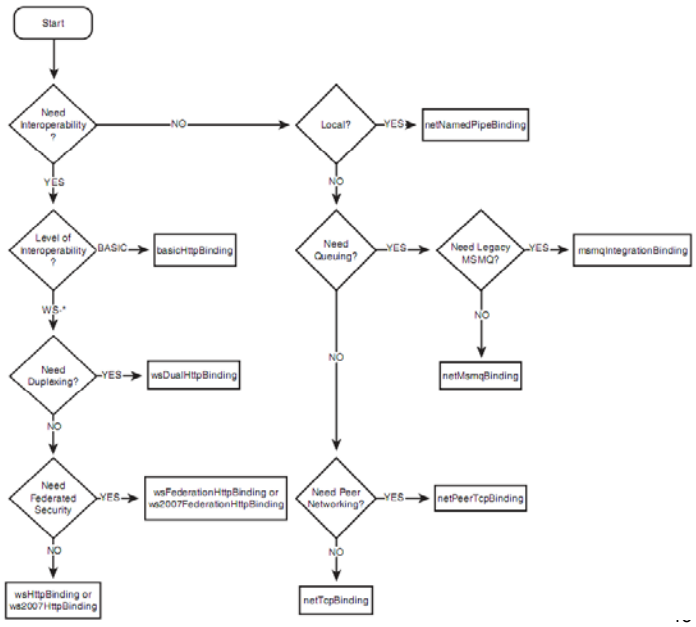
WCF – Elements

- Standard Bindings in WCF 3.0

Binding Name	Transport-Level Security	Message-Level Security	WS-* Interoperability	WS-*Transactions	Durable Reliable Messaging	Reliable Sessions	Performance	Communication		
								Request/Reply	One-way	Duplex
basicHttpBinding	X	X	X				Good	X	X	
wsHttpBinding	X	X	X	X		RS*	Good	X	X	
wsDualHttpBinding	X	X	X	X		RS*	Good	X	X	X
netTcpBinding	X	X		X		RS*	Better	X	X	X
netNamedPipeBinding	X			X			Best	X	X	X
netMsmqBinding	X	X			X		Better		X	
netPeerTcpBinding	X						Good		X	X
msmqIntegrationBinding	X				X		Better		X	
wsFederationHttpBinding	X	X	X			RS*	Good	X	X	
ws2007HttpBinding	X	X	X	X		RS*	Good	X	X	
ws2007FederationHttpBinding	X	X	X			RS*	Good	X	X	

WCF – Elements

Choose a binding that suit you need:



WCF – Elements

- Contract: what the service can do
  - Service contract: define operations, communications, behaviours
    - map to WSDL
  - Data contract: define data entities, parameter types
    - map to XSD
  - Message contract: define message format
    - map to SOAP
  - Fault contract: define error types

WCF – Elements

- ServiceContract: define a set of operations
  - OperationContract: define a single method
  - Note: the service interface & implementation

```
[ServiceContract]
public interface IService
{
    [OperationContract]
    string GetData(int value);
}

public class ConcreteService : IService
{
    public string GetData(int value)
    { ... }

    public string OtherMethod()
    { ... }
}
```


Multi service contracts

```
[ServiceContract]
public interface IGoodStockService
{
    [OperationContract]
    double GetStockPrice(string ticker);
}
[ServiceContract]
public interface IGreatStockService
{
    [OperationContract]
    double GetStockPriceFast(string ticker);
}

[ServiceContract]
public interface IAllStockServices : IGoodStockService,
IGreatStockService { };

public class AllStockServices : IAllStockServices
{
    public double GetStockPrice(string ticker)
    {
    }
    public double GetStockPriceFast(string ticker)
    {
    }
}
```

WCF – Elements

- DataContract: specify type as a data contract
  - DataMember: member that is a part of contract
  - Note: properties declaration

```
[DataContract]
public class CustomType
{
    [DataMember]
    public bool MyFlag { get; set; }

    [DataMember]
    public string MyString { get; set; }

}
```


WCF – Elements

- DataContract
  - Name: overide name of the type

```
[DataContract(Name="Contact")]
public class Person
{
   [DataMember(IsRequired=true, Name="SurName")]
   public string LastName;

   public string FirstName; //Not included in contract
}
```

WCF – Elements

- DataContract : inheritance

```
[DataContract]
public class Employee { ... }

[DataContract]
public class Manager : Employee { ... }

[ServiceContract]
public interface IEmployeeService
{
   [OperationContract]
   public void AddEmployee(Employee e);
}
```


WCF – Elements

- MessageContract : control the SOAP

```
[MessageContract]
public class ComplexProblem
{
    [MessageHeader]
    public string operation;
    [MessageBody]
    public ComplexNumber n1;
    [MessageBody]
    public ComplexNumber n2;
    [MessageBody]
    public ComplexNumber solution;

    // Constructors…
}
```

WCF – Elements

- FaultContract :
  - In WCF, exceptions are passed as FaultException
  - FaultContract specifies what kind of Exceptions an openration can throw

```
[ServiceContract]
public interface IEmployeeService
{
    [OperationContract]
    [FaultContract(typeof(ValidationException))]
    public void AddEmployee(Employee e);
}
```


WCF – Elements

- FaultContract :
  - Providers (i.e. implemented services) throw FaultException

```
public class EmployeeService
{
    public void AddEmployee(Employee e)
    {
            ...
            throw new FaultException<ValidationException>
                (new ValidationException(errorMsg));

    }
}
```

WCF – Elements

- FaultContract :
  - Consumers catch FaultExceptions

```
EmployeeServiceProxy proxy = new EmployeeServiceProxy();
try
{
     ...
    proxy.AddEmployee(emp);
}
catch(FaultException<ValidationException> e)
{
     //Do stuff with exception here
 }
 catch(FaultException e)
 {
    //Will catch all other types of Fault exceptions...
 }
```


WCF

- WCF example
  - Foreteller:
    - Foretell operation contract
    - Foretune data contract
  - Service host:
    - Self – host: WCF service library
    - IIS host (web service): WCF service application
  - WCF test tool : debug / run WCF project
  - WCF edit configuration tool : Right Click in the config file

WCF

- WCF example: WCF edit configuration tool
  - Create new service
    - Select service type (from bin after building the project)
    - Enter address
  - Add the Base Address of Host (could use the endpoint's address)
  - Create Service Behavior Configuration
    - Add serviceMetaData: allow service discoverable
      - HttpGetEnabled
  - Select Service's BehaviorConfiguration

WCF – Elements

- Three (3) ways to 'talk' between consumers – providers

  - One way: invoke function
  - Request – reply: synchronous functions
  - Duplex: asynchronous function (callback)

WCF – Elements

- One way & Request - Reponse

```
[ OperationContract(IsOneWay = true)]
void DoBigAnalysisFast(string ticker);

[OperationContract]
void DoBigAnalysisSlow(string ticker);
```

Demo OneWay

(Note: this example is the console application which hosts WCF service by code)

WCF – Elements

- In WCF, consumers might call services' functions aynchronously
  - Client Generate asynchronous operations

Demo Asynchronous

WCF – Elements

- Duplex: asynchronous communication (callback style)
  - use wsDualHttpBinding
  - Duplex asymetric

```
[ServiceContract(Session=true,
          CallbackContract=typeof(ICalculatorResults)]
public interface ICalculatorProblems
{
    [OperationContract(IsOneWay=true)]
    void SolveProblem (ComplexProblem p);
}

public interface ICalculatorResults
{
    [OperationContract(IsOneWay=true)]
    void Results(ComplexProblem p);
}
```

WCF – Elements

- Duplex: asynchronous communication (callback style)
  - use wsDualHttpBinding
  - Duplex symetric

```
[ServiceContract(Session=true,
          CallbackContract=typeof(IChat)]
public interface IChat
{
    [OperationContract(IsOneWay=true)]
    void Talk(string text);
}
```

Demo Duplex

WCF

- WCF allows SOA implemented within .NET framework unitedly
- Endpoint: a connection point in service model
  - Address
  - Binding
  - Contract: Service contract, Data contract
- Note: the usage of interface

WCF vs ASMX

http://keithelder.net/2008/10/17/WCF-vs-ASMX-WebServices/

Simple but not very powerful.

WCF vs ASMX

http://keithelder.net/2008/10/17/WCF-vs-ASMX-WebServices/

Implement SOA using interfaces.

Code one then config to expose services in many different ways

Configuration is important and require learning

WCF - Configuration

- The WCF programming model supports:
  - Declarative (via attributes)
  - Imperative (via code)
  - Configurative (via XML config files)
    - App.config (Web.config) must has at least one endpoint and one behavior to allow communication

    → http://msdn.microsoft.com/en-us/library/ms733932.aspx

→ Configuration should be (self) studied more

References

1. Paul Andrew, Real World SOA using WCF and WF, 2007
2. Mahesh Krisnan, Pratical WCF Part 1
3. Nguyễn Bá Quang, Trịnh Minh Cường; Windows Communication Foundation Part 1
4. http://www.wcftutorial.net/
5. http://msdn.microsoft.com/en-us/library/ms734712.aspx
6. http://www.deitel.com/ResourceCenters/Programming/WindowsCommunicationFoundation/Tutorials/tabid/2831/Default.aspx
7. http://www.switchonthecode.com/tutorials/wcf-tutorial-basic-interprocess-communication (Communication via code)
8. http://msdn.microsoft.com/en-us/netframework/dd939784

Issues not discussed yet

- WCF:
  - WCF configuration: http://msdn.microsoft.com/en-us/library/ms733830.aspx
  - WCF security: http://msdn.microsoft.com/en-us/library/ff648370.aspx