Windows Communication Foundation (WCF)

WEEK 09

MURTAZA MUNAWAR FAZAL

Service Contracts

WHAT DOES YOUR SERVICE DO?

Service Contract

```
using System.ServiceModel;

[ServiceContract]
public interface ICalculate
{
    [OperationContract]
    double Add( double a, double b);
    [OperationContract]
    double Subtract( double a, double b);
}
```

Service Contract: OneWay

```
[ServiceContract]
public interface IOneWayCalculator
{
    [OperationContract(IsOneWay=true)]
    void StoreProblem (ComplexProblem p);
}
```

Service Contract: Duplex Asymmetric

```
[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);
}
```

Data contracts

WHAT OBJECT DATA NEEDS TO FLOW BACK AND FORTH?

Data Contract

```
[DataContract]
public class ComplexNumber
{
    [DataMember]
    public double Real = 0.0D;
    [DataMember]
    public double Imaginary = 0.0D;

    public ComplexNumber(double r, double i)
    {
        this.Real = r;
        this.Imaginary = i;
    }
}
```

Message contracts

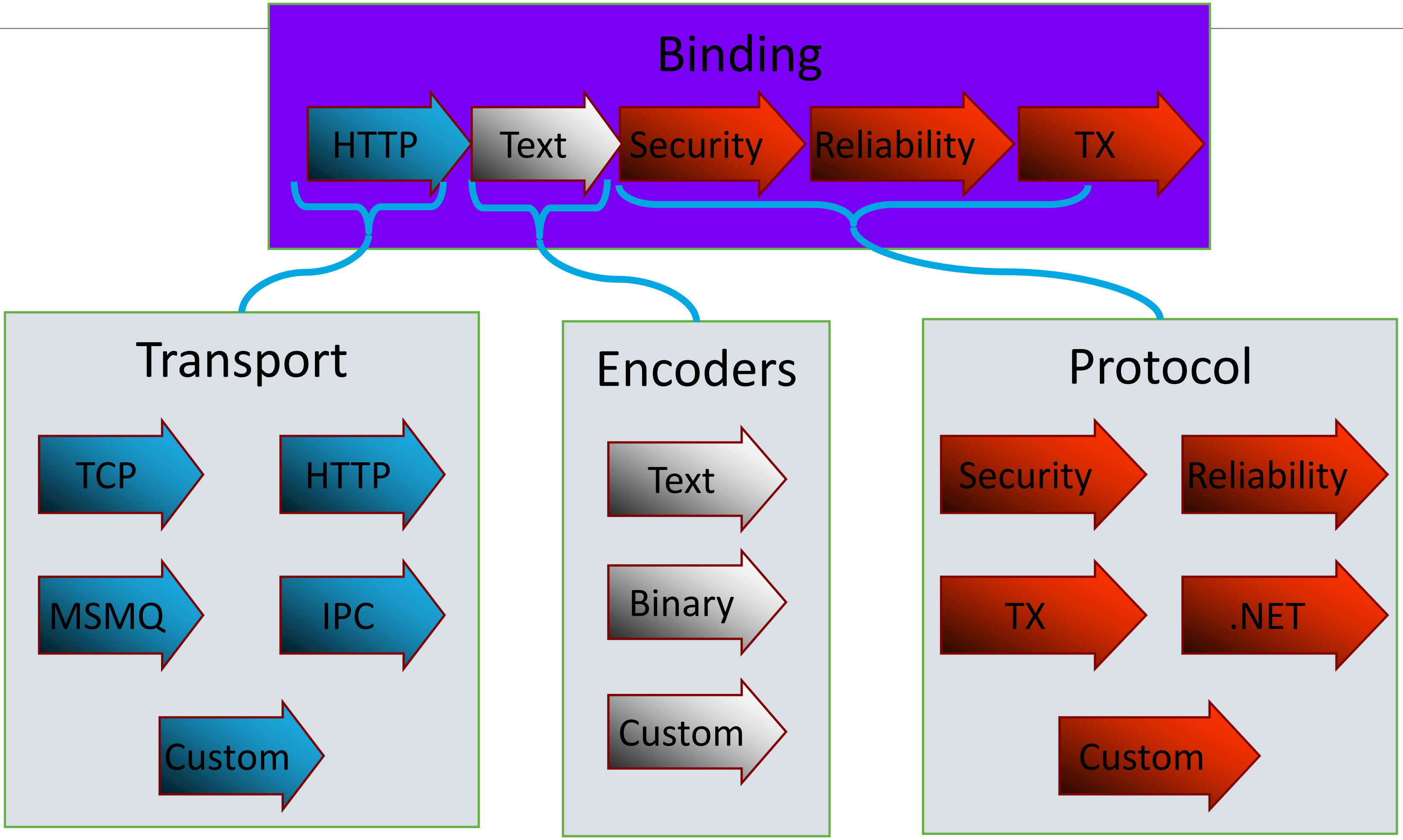
DEFINES THE MAPPING BETWEEN THE TYPE AND A SOAP ENVELOPE

Message Contract

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

    // Constructors…
}
```

bindings

Bindings & Binding Elements



Standard Bindings

Binding	Interop	Security	Session	TX	Duplex
BasicHttpBinding	**BP 1.1**	**N, T**	**N**	**N**	**n/a**
WSHttpBinding	**WS**	**M, T, X**	**N, T, RS**	**N, Yes**	**n/a**
WSDualHttpBinding	**WS**	**M**	**RS**	**N, Yes**	**Yes**
WSFederationBinding	**Federation**	**M**	**N, RS**	**N, Yes**	**No**
NetTcpBinding	**.NET**	**T, M**	**T ,RS**	**N, Yes**	**Yes**
NetNamedPipeBinding	**.NET**	**T**	**T, N**	**N, Yes**	**Yes**
NetPeerTcpBinding	**Peer**	**T**	**N**	**N**	**Yes**
NetMsmqBinding	**.NET**	**T, M, X**	**N**	**N, Yes**	**No**
MsmqIntegrationBinding	**MSMQ**	**T**	**N**	**N, Yes**	**n/a**

N = None | T = Transport | M = Message | B = Both | RS = Reliable Sessions

Code vs. Config

Defining Endpoints

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration
xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <services>
      <service serviceType="CalculatorService">
        <endpoint address="Calculator"
                  binding=“basicHttpBinding"
                  contract="ICalculator" />
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

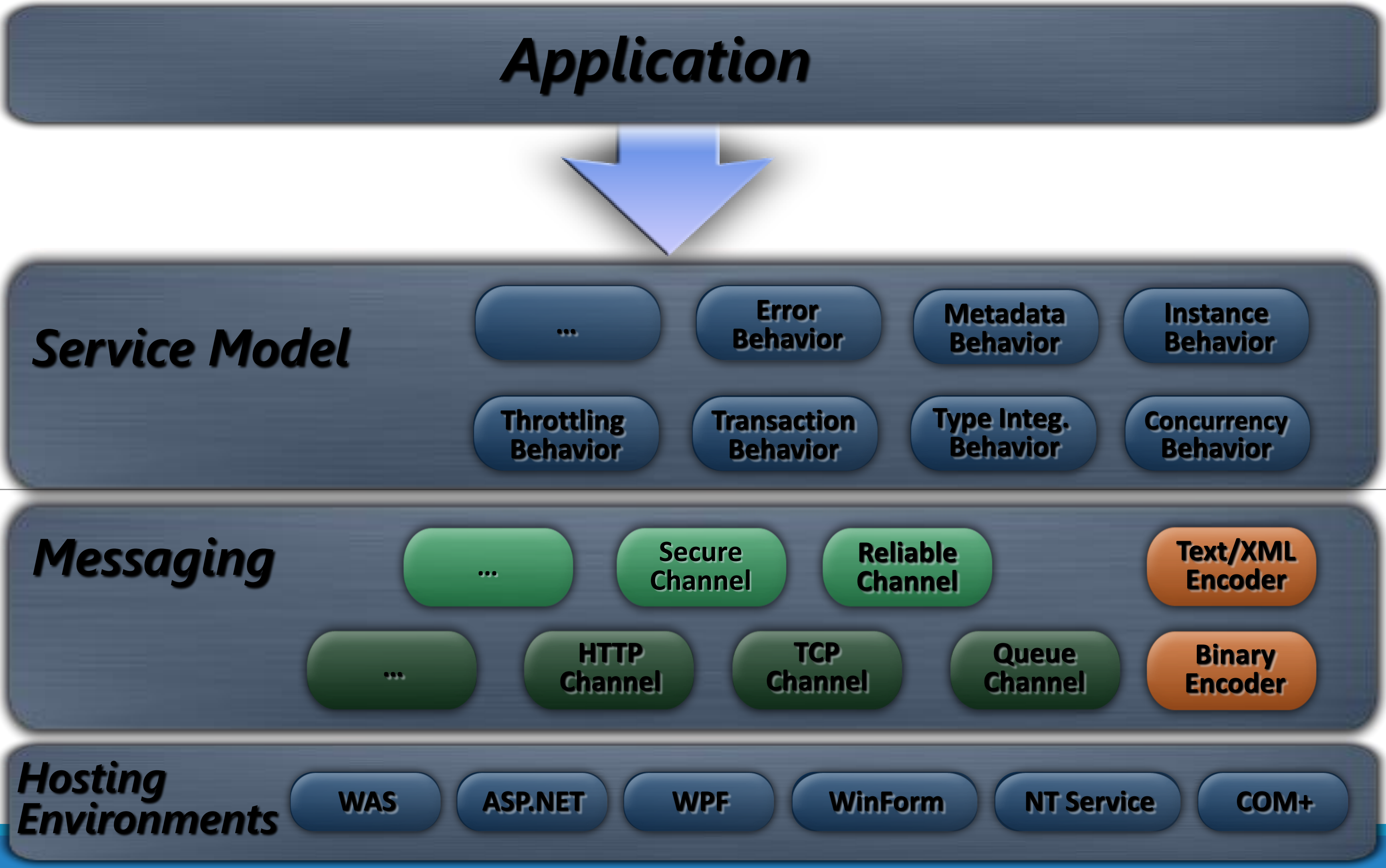
Configuring Bindings

```
<endpoint address="Calculator"

bindingSectionName="basicProfileBinding"
          bindingConfiguration="Binding1"
          contractType="ICalculator" />
<bindings>
  <basicProfileBinding>
    <binding configurationName="Binding1"

hostnameComparisonMode="StrongWildcard"
             transferTimeout="00:10:00"
             maxMessageSize="65536"
             messageEncoding="Text"
             textEncoding="utf-8"
    </binding>
  </basicProfileBinding>
</bindings>
```

Custom Bindings

```
<bindings>
    <customBinding>
        <binding configurationName="Binding1">
            <reliableSession bufferedMessagesQuota="32"
                inactivityTimeout="00:10:00"
                maxRetryCount="8"
                ordered="true" />
            <httpsTransport manualAddressing="false"
                maxMessageSize="65536"
                hostnameComparisonMode="StrongWildcard"/>
            <textMessageEncoding maxReadPoolSize="64"
                maxWritePoolSize="16"
                messageVersion="Default"
                encoding="utf-8" />
        </binding>
    </customBinding>
</bindings>
```

WCF Summary


Application
Service Model
...
Error Behavior
Metadata Behavior
Instance Behavior
Throttling Behavior
Transaction Behavior
Type Integ. Behavior
Concurrency Behavior
Messaging
...
Secure Channel
Reliable Channel
Text/XML Encoder
...
HTTP Channel
TCP Channel
Queue Channel
Binary Encoder
Hosting Environments
WAS
ASP.NET
WPF
WinForm
NT Service
COM+

WCF Summary

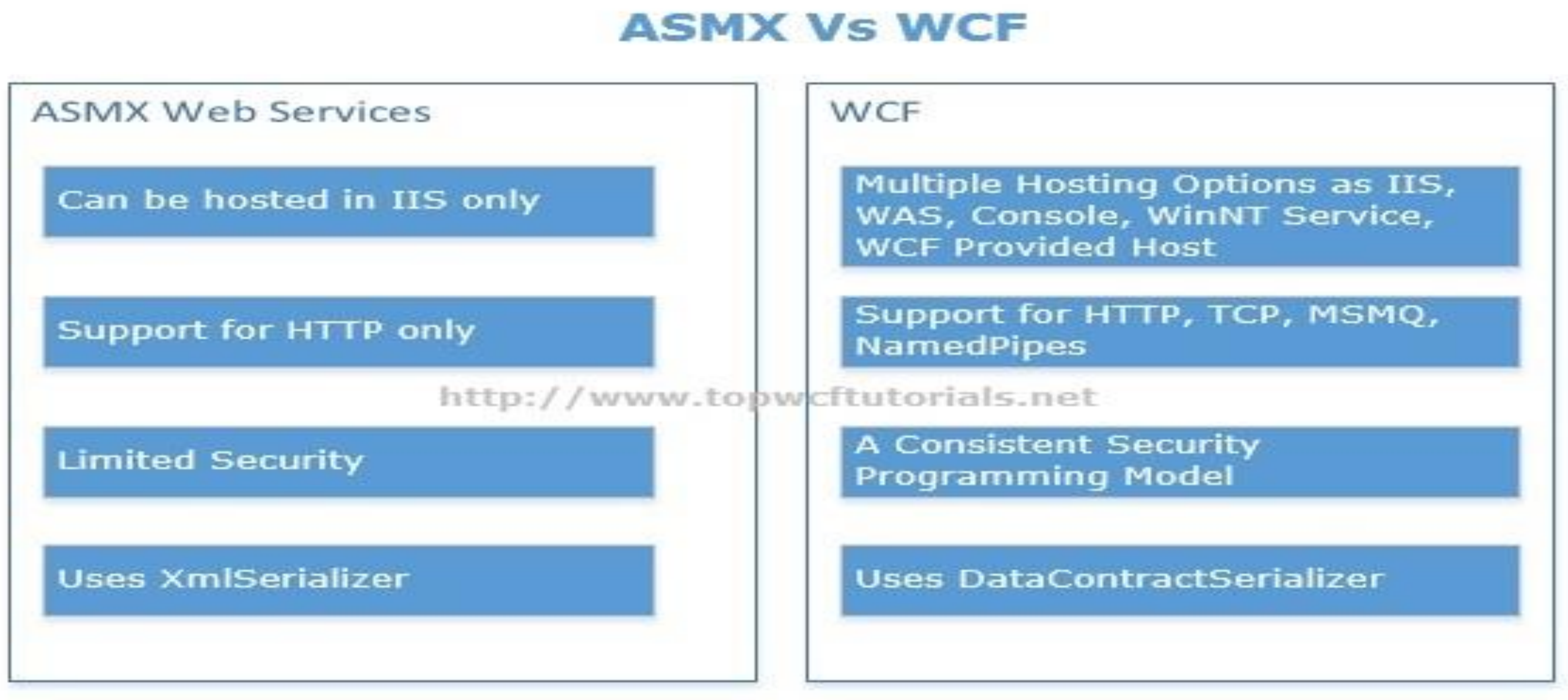
WCF is the future of distributed computing

It combines the best of all existing Microsoft distributed computing stacks

It uses WS-* standards for interoperability and .NET value-add for performance and integration with existing solutions

Generate proxy class using svcutil.exe

Web Services v/s WCF



ASMX Vs WCF

ASMX Web Services	WCF
Can be hosted in IIS only	Multiple Hosting Options as IIS, WAS, Console, WinNT Service, WCF Provided Host
Support for HTTP only	Support for HTTP, TCP, MSMQ, NamedPipes
Limited Security	A Consistent Security Programming Model
Uses XmlSerializer	Uses DataContractSerializer

Choosing Between WCF and ASMX

Characteristic	ASMX	WCF
Development effort	ASMX requires a lower level of development skills than WCF.	WCF can require a higher level of development skills because it has a larger, more flexible programming model.
Flexibility	ASMX is accessible only through the basic HTTP transport.	With configuration, WCF service implementations can support many transport options.
Security	The ASMX security layer has fewer security features than the WCF security layer.	The WCF security layer has more security features than the ASMX security layer.
Performance	ASMX has lower performance characteristics than WCF.	WCF has higher performance characteristics than ASMX.