Windows Communication Foundation (WCF)

WEEK 09

MURTAZA MUNAWAR FAZAL

From Objects to Services

1980s

Object-Oriented

- Polymorphism
- Encapsulation
- Subclassing

1990s

Component-Based

- Interface-based
- Dynamic Loading
- Runtime Metadata

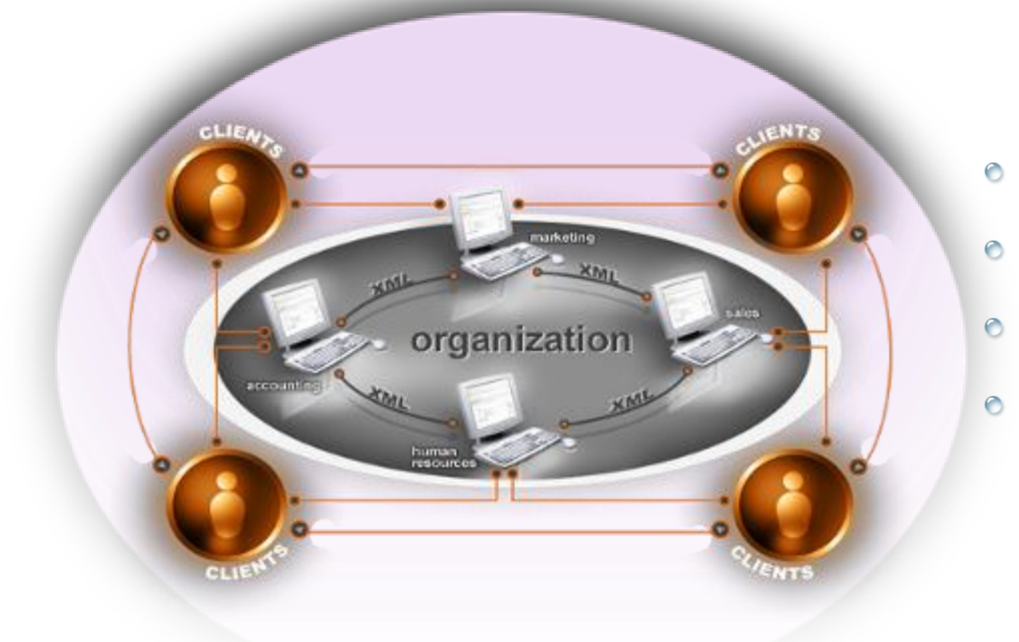
2000s

Service-Oriented

- Message-based
- Schema+Contract
- Binding via Policy

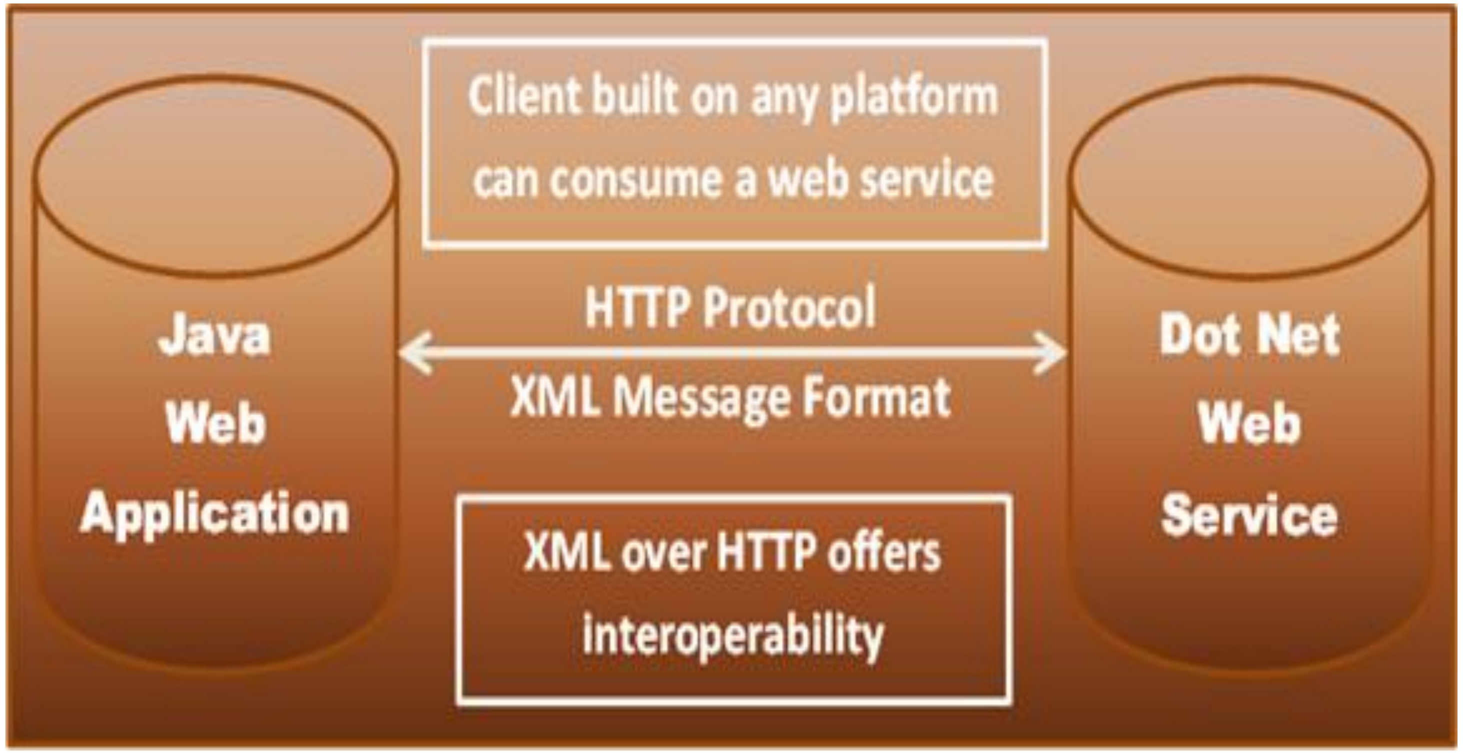
The Challenge

Radically Simplifying Distributed Application Development

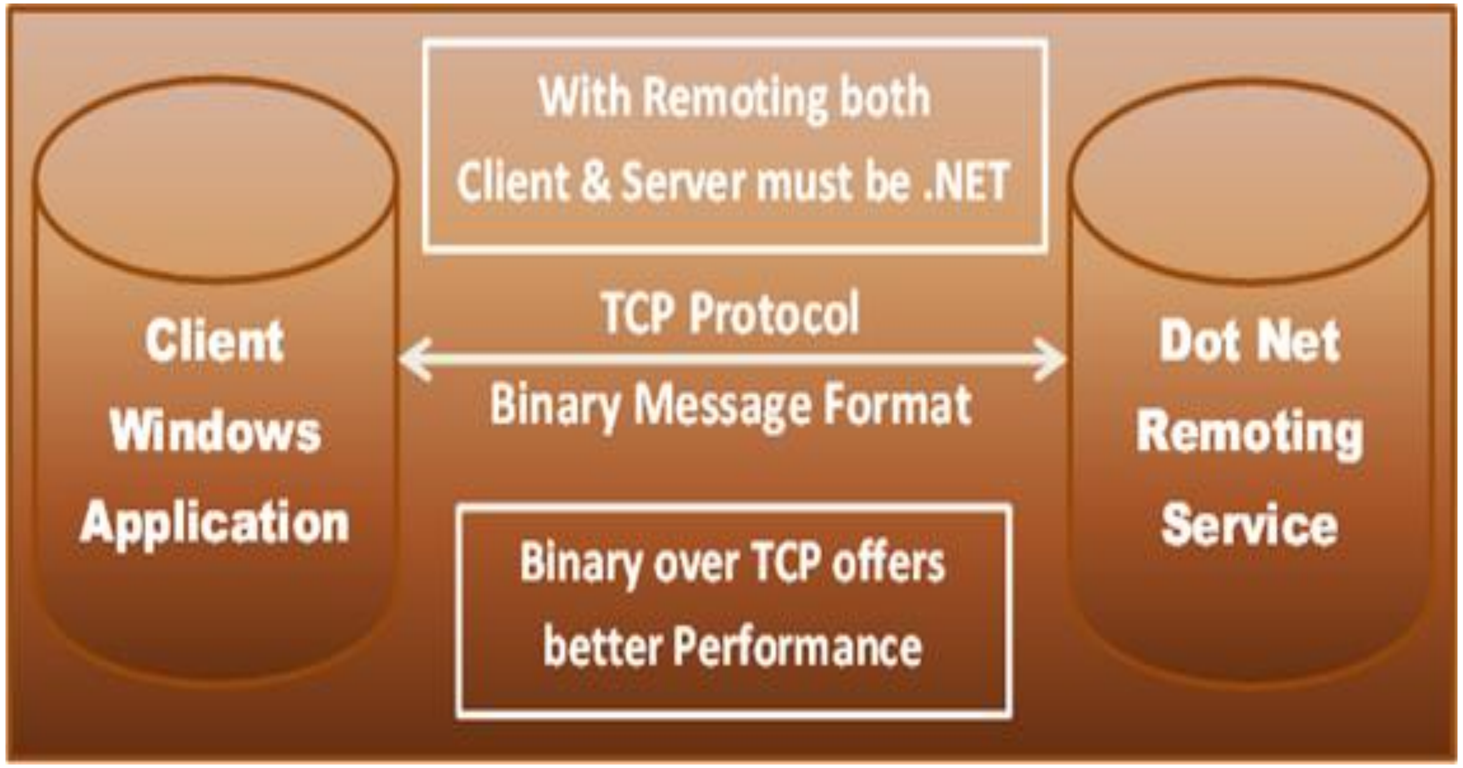


Development of connected systems remains costly and frustrating

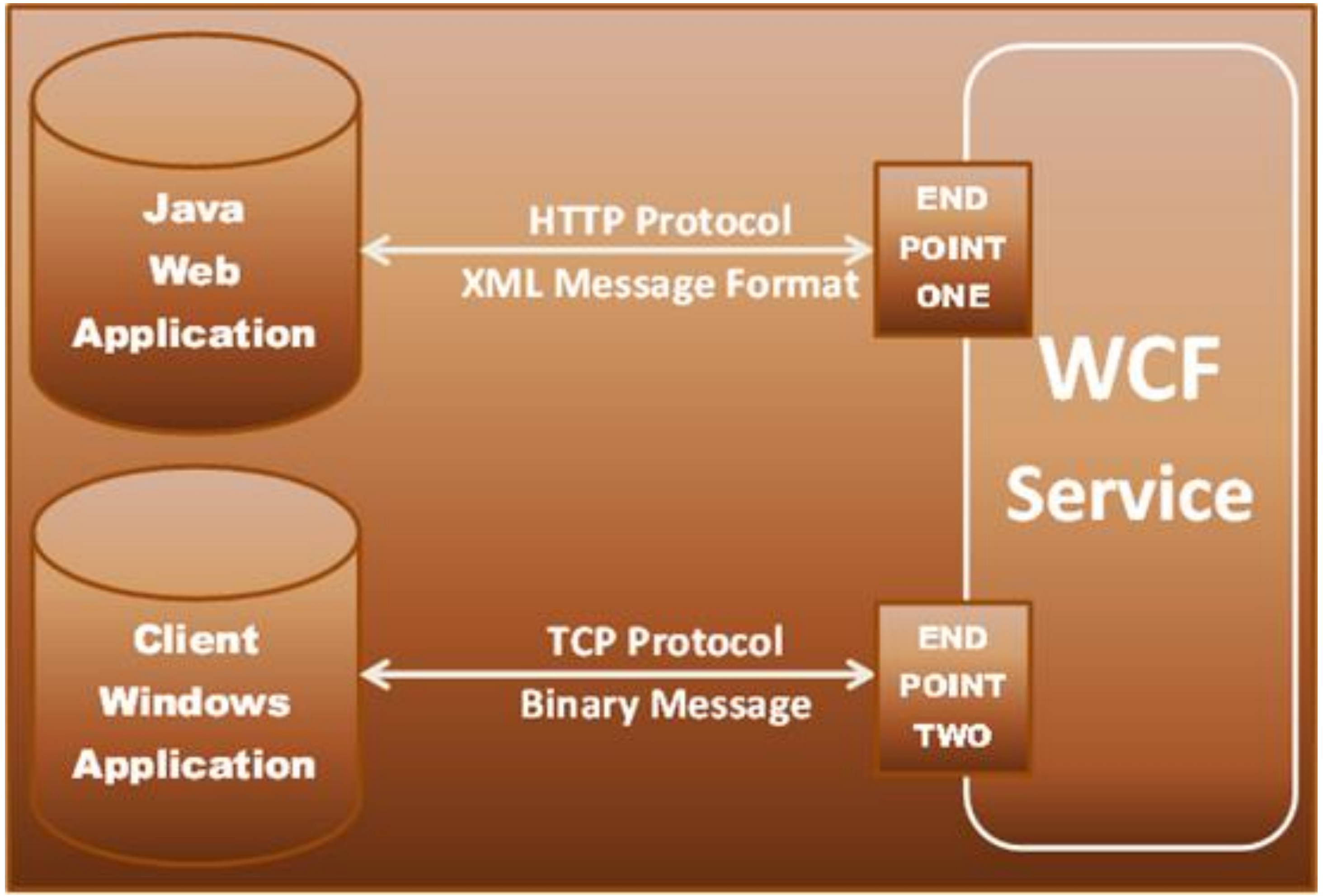
- **Different programming models for different tasks**
- **Need for security and reliable messaging**
- **Interoperability with applications on other platforms**
- **Productive service-oriented programming model needed**


Client built on any platform can consume a web service
Java Web Application
HTTP Protocol
XML Message Format
Dot Net Web Service
XML over HTTP offers interoperability

Example 1

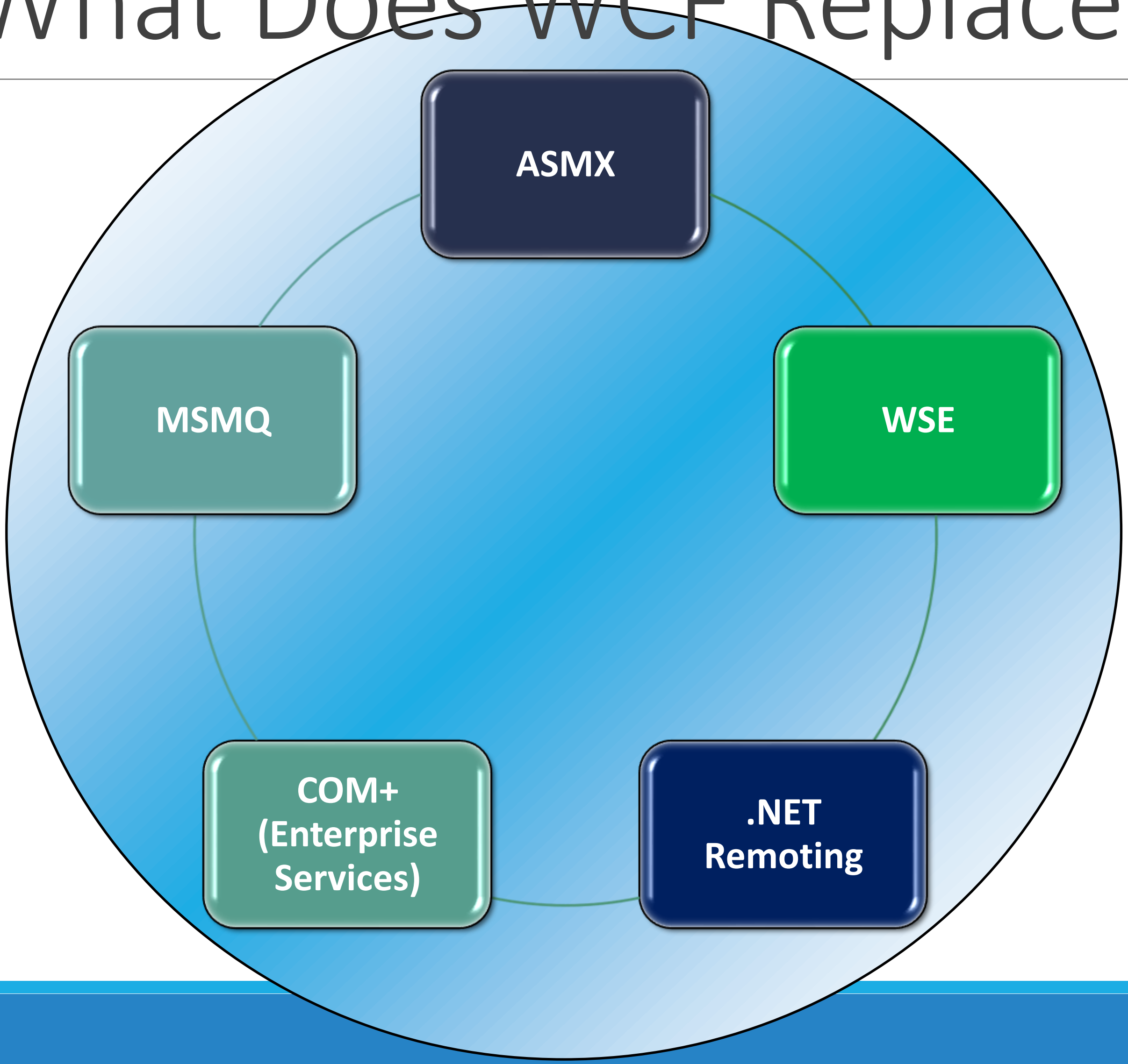

With Remoting both
Client & Server must be .NET
Client
Windows
Application
TCP Protocol
Binary Message Format
Dot Net
Remoting
Service
Binary over TCP offers
better Performance

Example 2



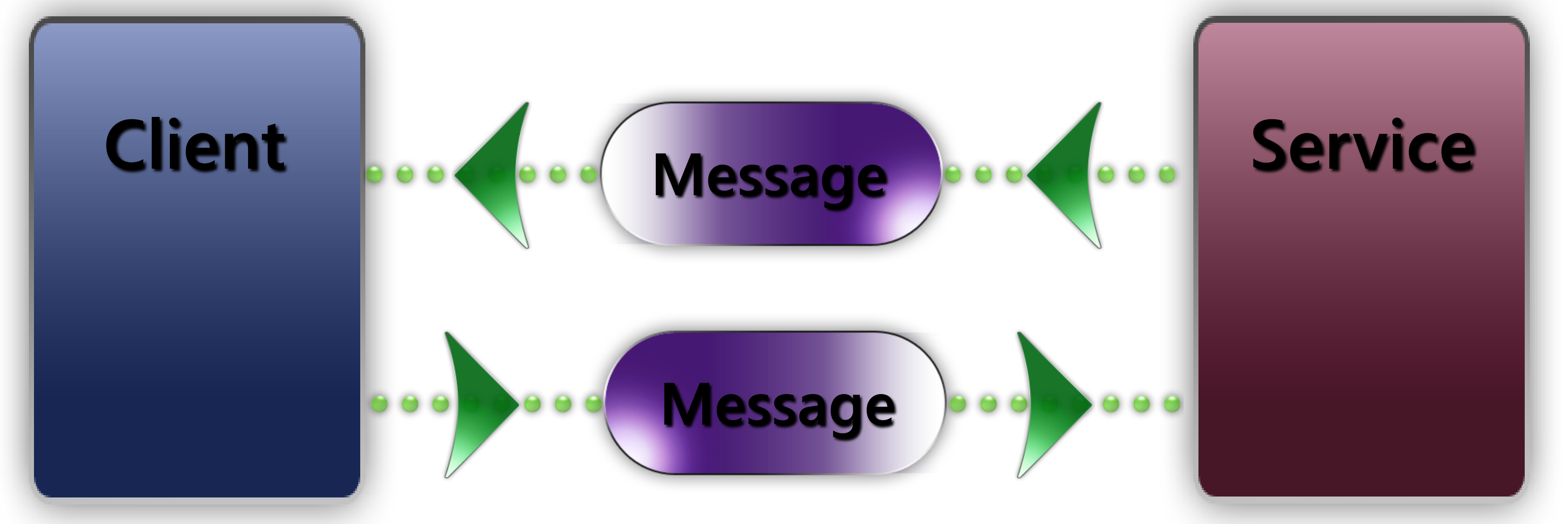
WCF Solution

What Does WCF Replace?

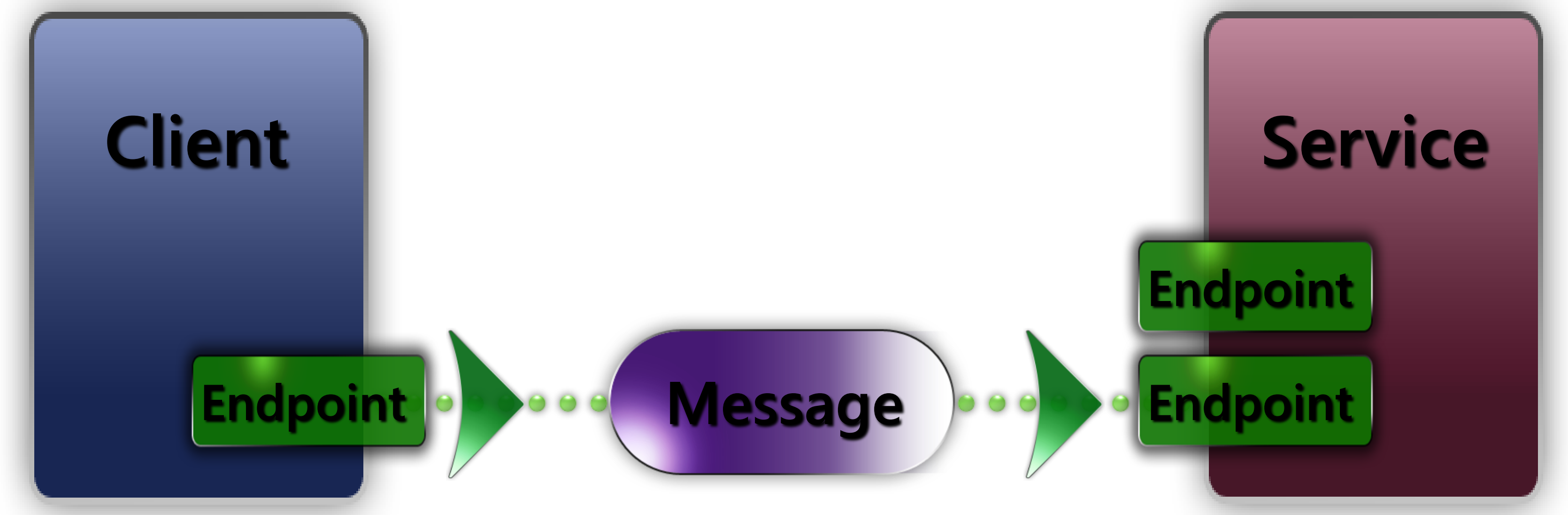


Understanding WCF principles

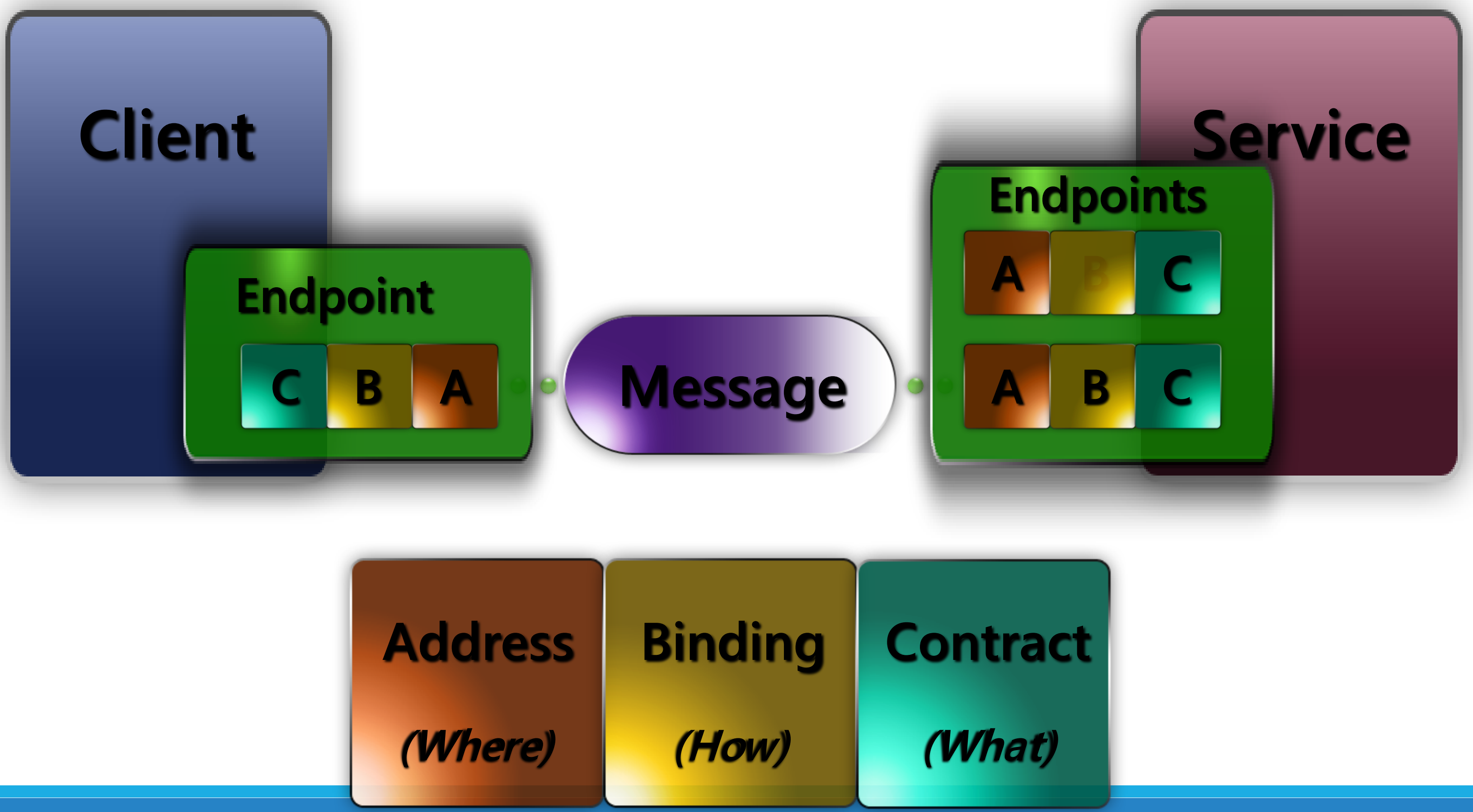
Services and Clients



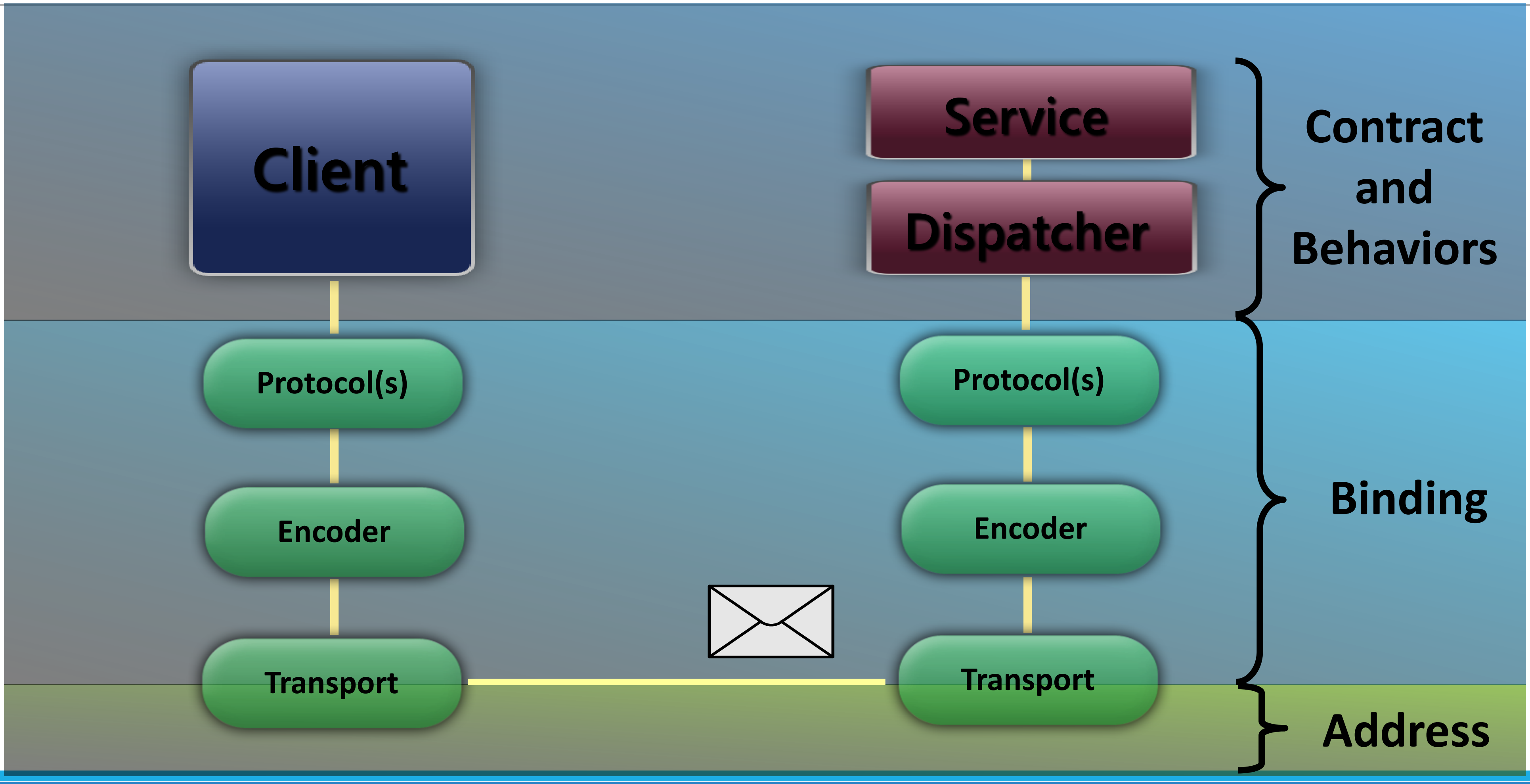
Endpoints



Address, Binding, Contract



WCF Architecture: Messaging Runtime



Address

Basically URL, specifies where this WCF service is hosted. Client will use this URL to connect to the service. e.g.

http://localhost:8090/MyService/SimpleCalculator.svc

Contracts

THE WHAT

What are Contracts?

Collection of operation that specifies what the endpoint will communicate with outside world. Usually name of the Interface will be mentioned in the Contract, so the client application will be aware of the operations which are exposed to the client. Each operation is a simple exchange pattern such as one-way, duplex and request/reply.

Three Types of Contracts

Service Contract	Data Contract	Message Contract
Defines Operations, Behaviors and Communication Shape	Defines Schema and Versioning Strategies	Allows defining application-specific headers and unwrapped body content
What does your service do	What object data is used	Allows control over the SOAP structure of messages

Service Contract

```
[ServiceContract]
public interface IService1
  {
     [OperationContract]
      string GetData(int value);

      [OperationContract]
      CompositeType GetDataUsingDataContract(CompositeType composite);

      [OperationContract]
      int Sum(int a, int b);
  }
```

Data Contracts

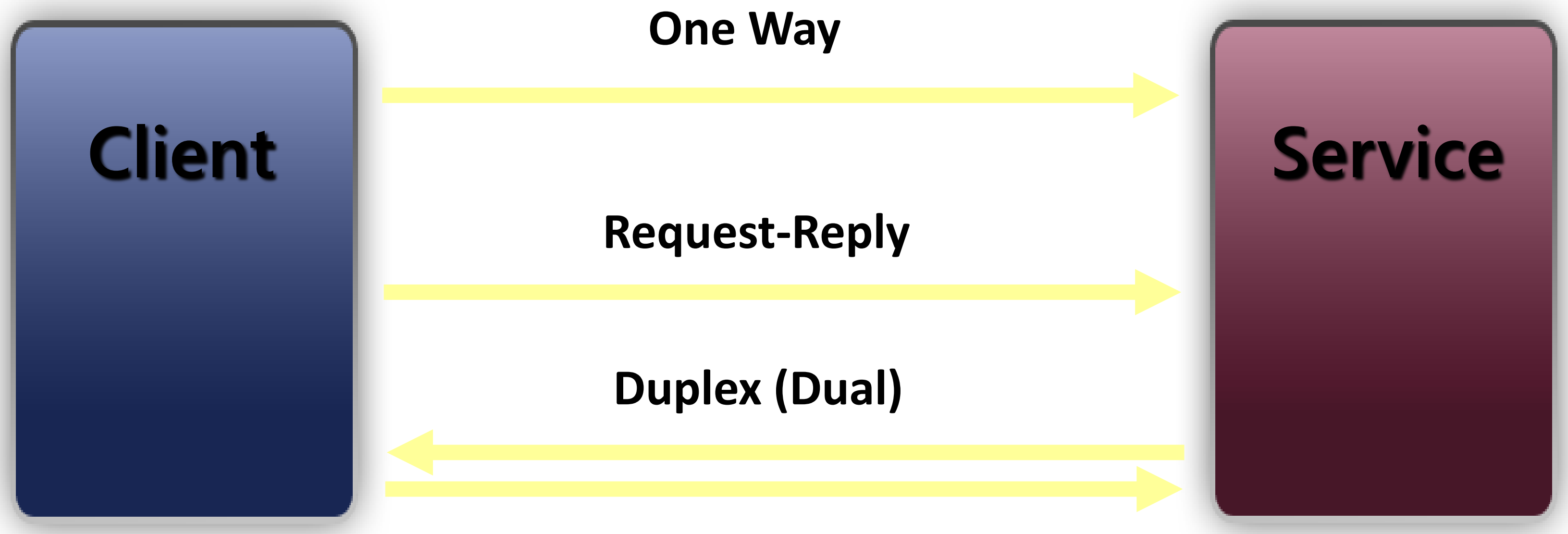
```
 [DataContract]

public class CompositeType
{

 bool boolValue = true;


[DataMember]
public bool BoolValue
{

 get { return boolValue; }
 set { boolValue = value; }
}

}
```

Message Contracts

Message is the packet of data which contains important information. WCF uses these messages to transfer information from Source to destination.

WCF uses SOAP(Simple Object Access Protocol) Message format for communication. SOAP message contain Envelope, Header and Body. SOAP envelope contains name, namespace, header and body element. SOAP Header contain important information which are not directly related to message. SOAP body contains information which is used by the target.

Ways to Talk



One Way:
- Datagram-style delivery

Request-Reply
- Immediate Reply on same logical thread

Duplex
- Reply “later” and on backchannel (callback-style)

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

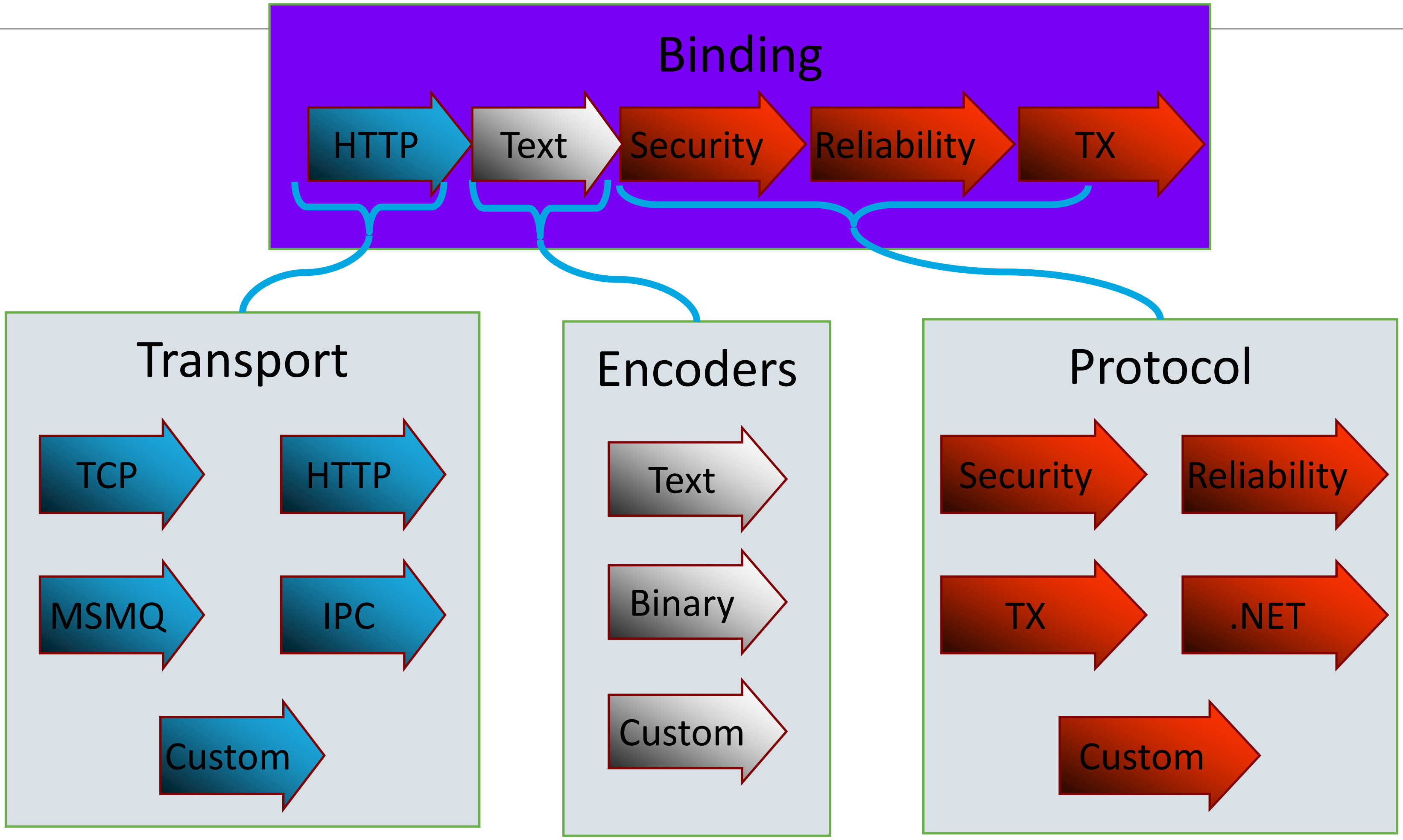
DEFINES THE MAPPING BETWEEN THE TYPE AND A SOAP ENVELOPE

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>
```

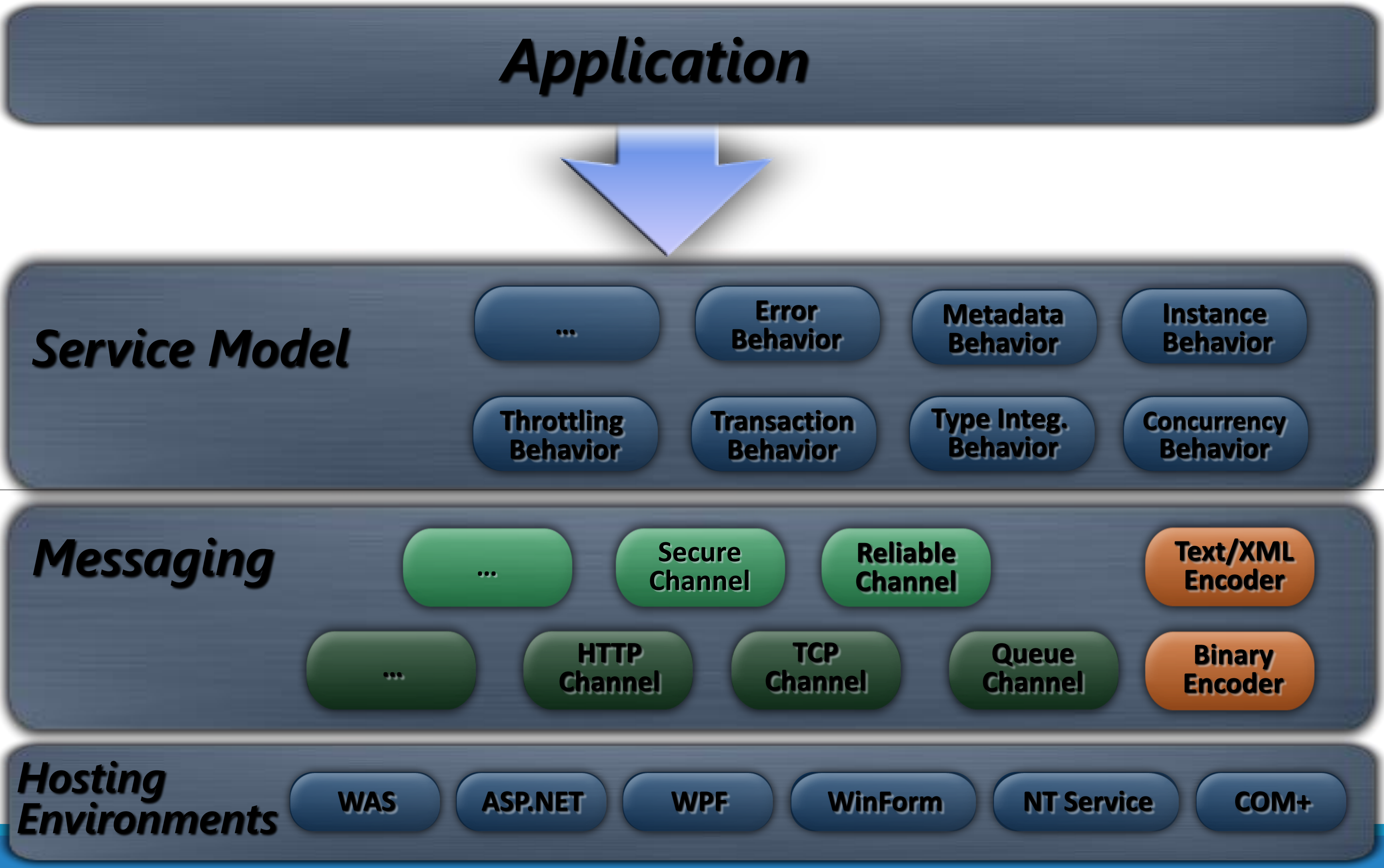
Configuring Bindings

```
<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

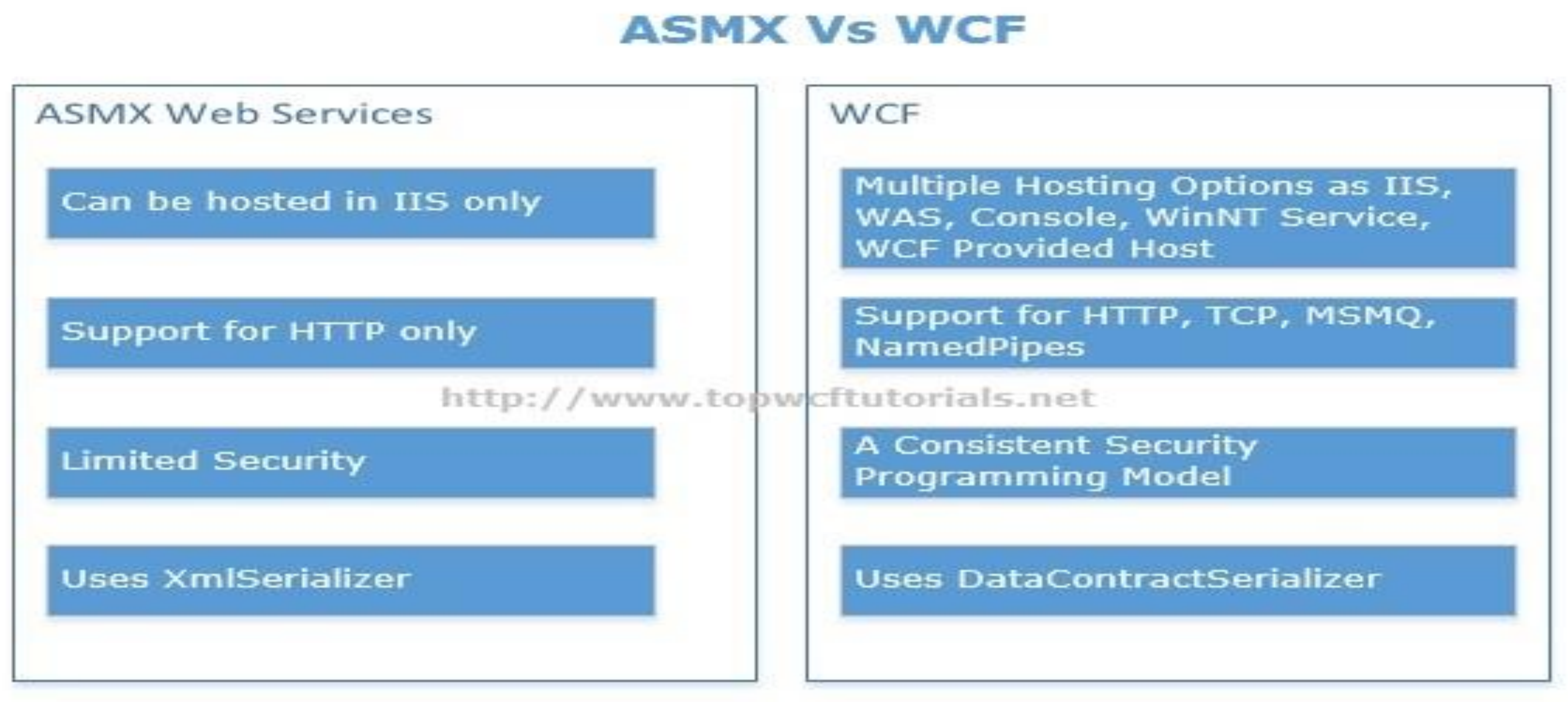
WCF is the future of distributed computing

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.