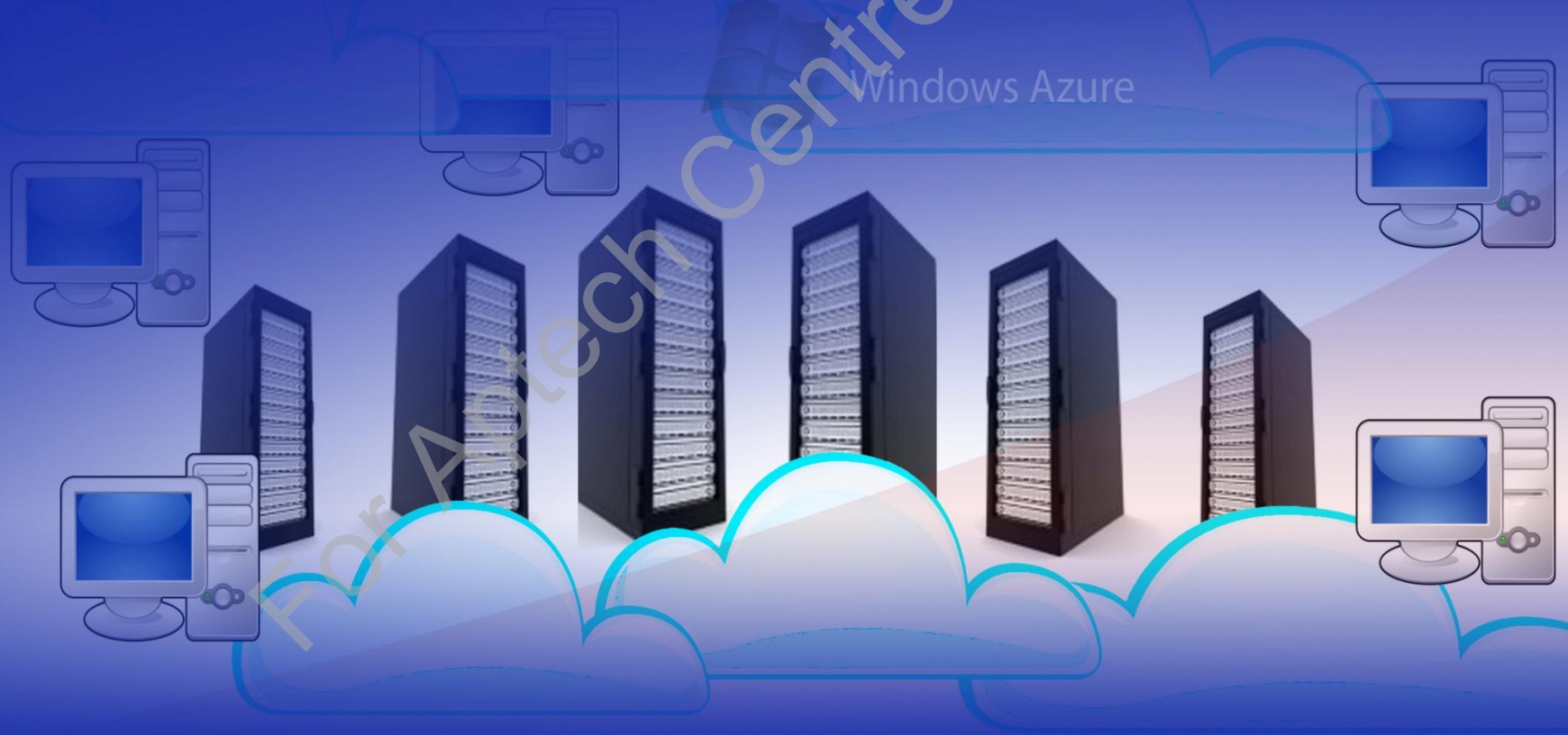
Enterprise Application Development Using Windows Azure and Web Services

Session 9

WCF Services



Learning Objectives

- Describe WCF services
- List the features of WCF that are not supported by ASP.NET Web API
- Explain the steps to create a WCF service
- Explain the process to define a service contract and implement it
- Explain how to host and configure a WCF service
- Describe the procedure to consume a WCF service from a client application
- Explain how to create and deploy a WCF cloud service to Azure

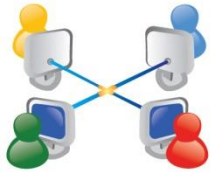
Introduction to WCF 1-2

- WCF framework is:

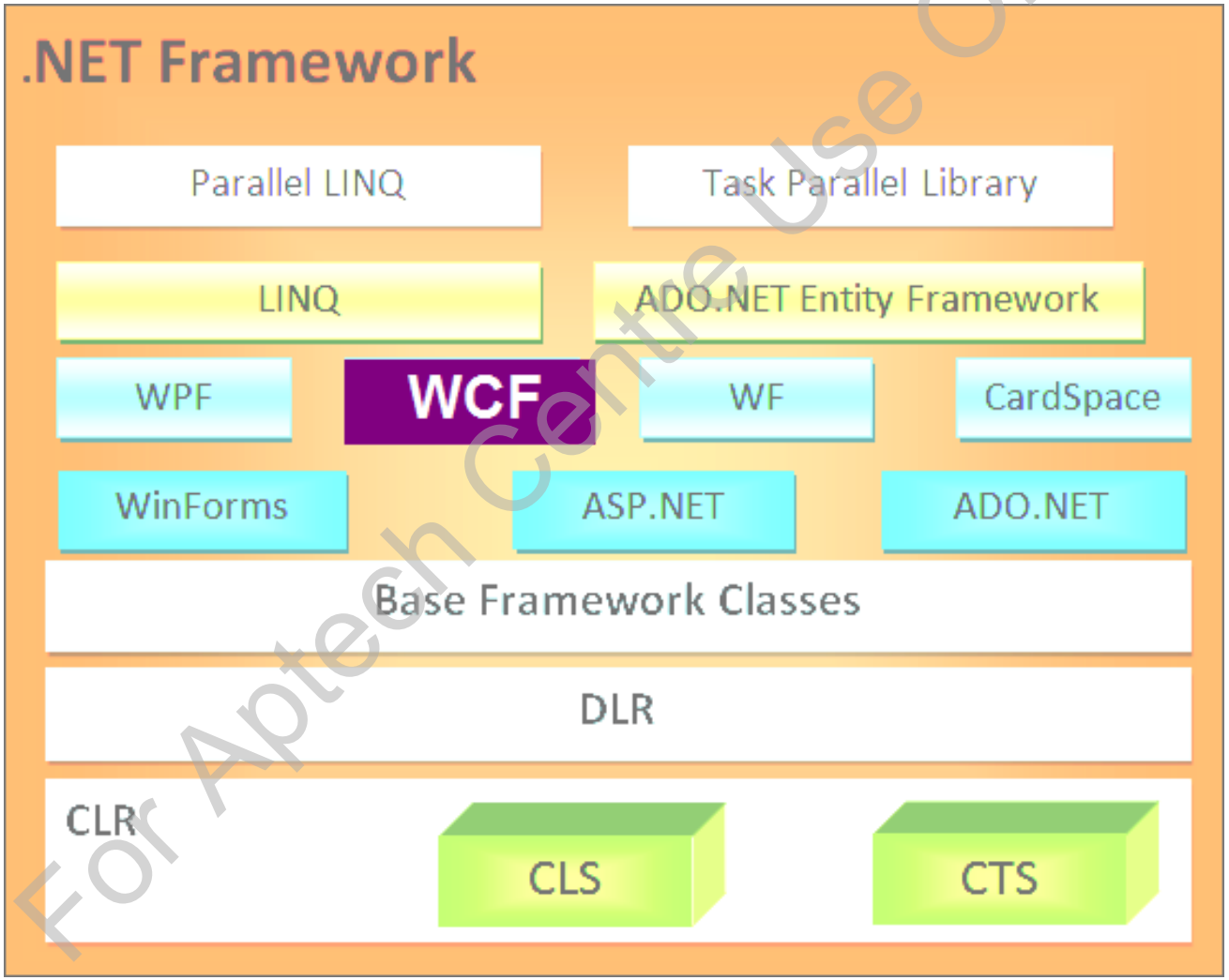
One of the best distributed computing technologies available to us.

The most up-to-date communication infrastructure made available to us by Microsoft.

One of the best solutions for building distributed applications that are based on SOA.



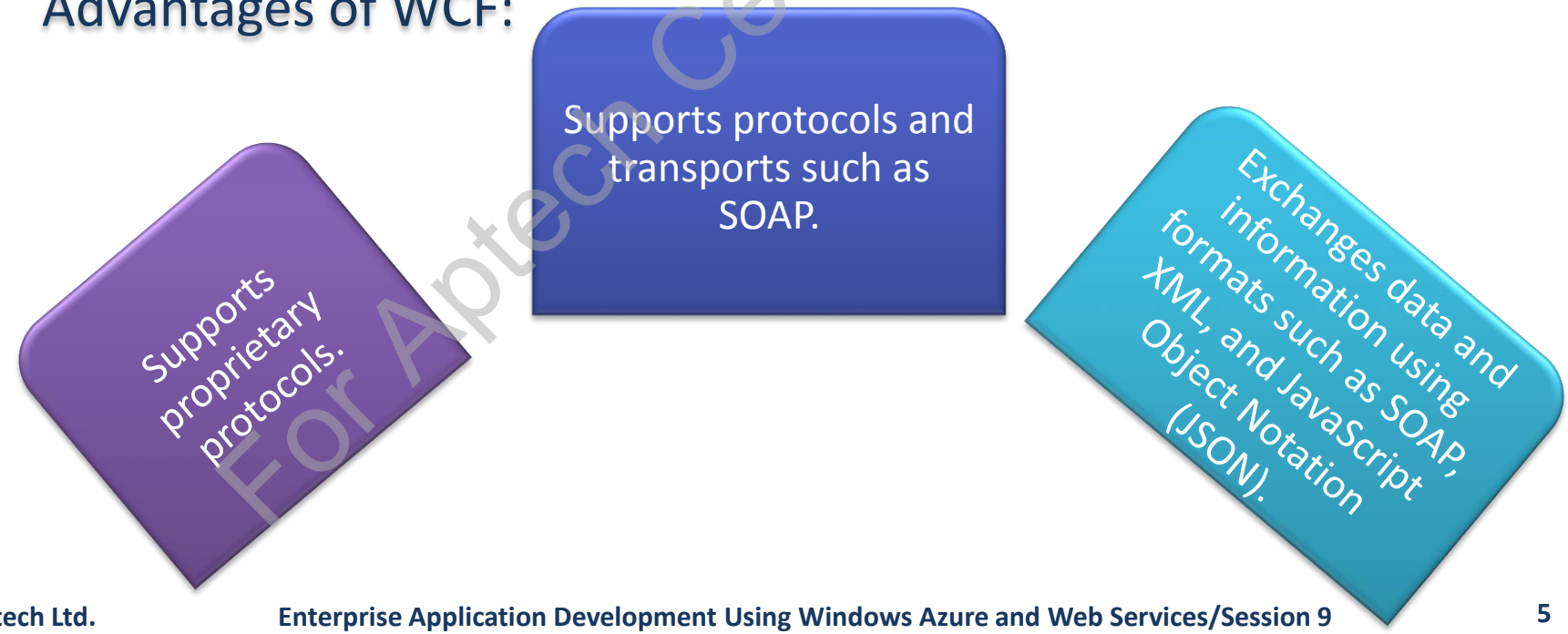
Introduction to WCF 2-2



Overview of WCF Services 1-6

Web Services	Uses HTTP for communication. Can cause restrictions in communication.
WCF	Uses any protocol for communication. Makes it easier to communicate with components of other languages.

Advantages of WCF:



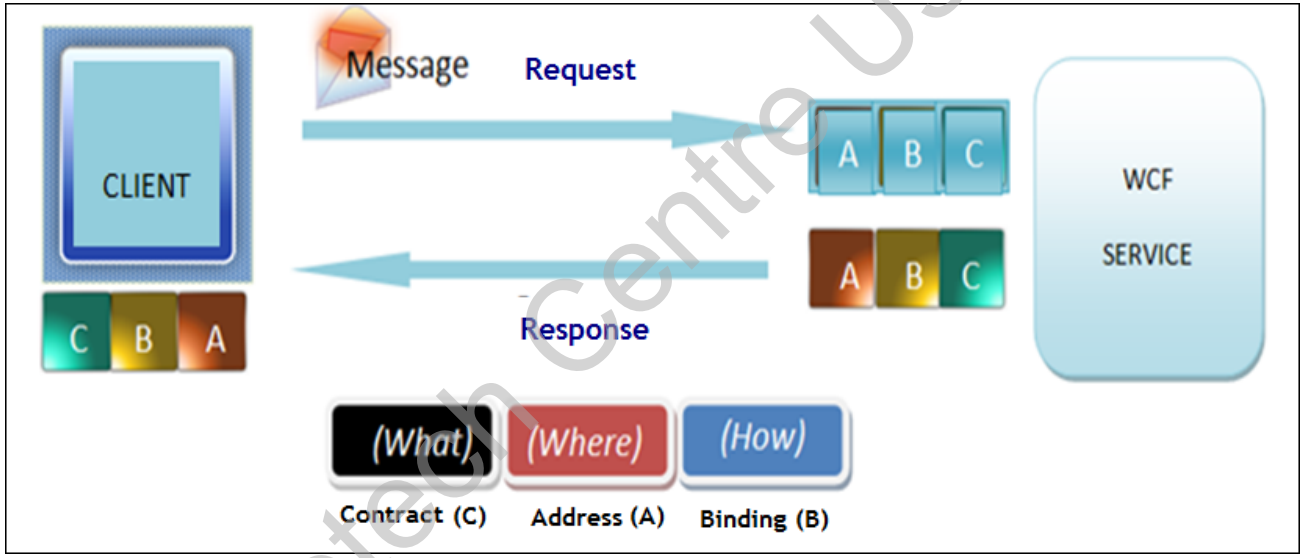
Overview of WCF Services 2-6

SOAP

- Is a lightweight protocol used to exchange data over the distributed environments.
- Uses XML for its message formatting, and usually relies on HTTP for message negotiation and transmission.
- Is also known as Remote Procedure Call (RPC) protocol.

Overview of WCF Services 3-6

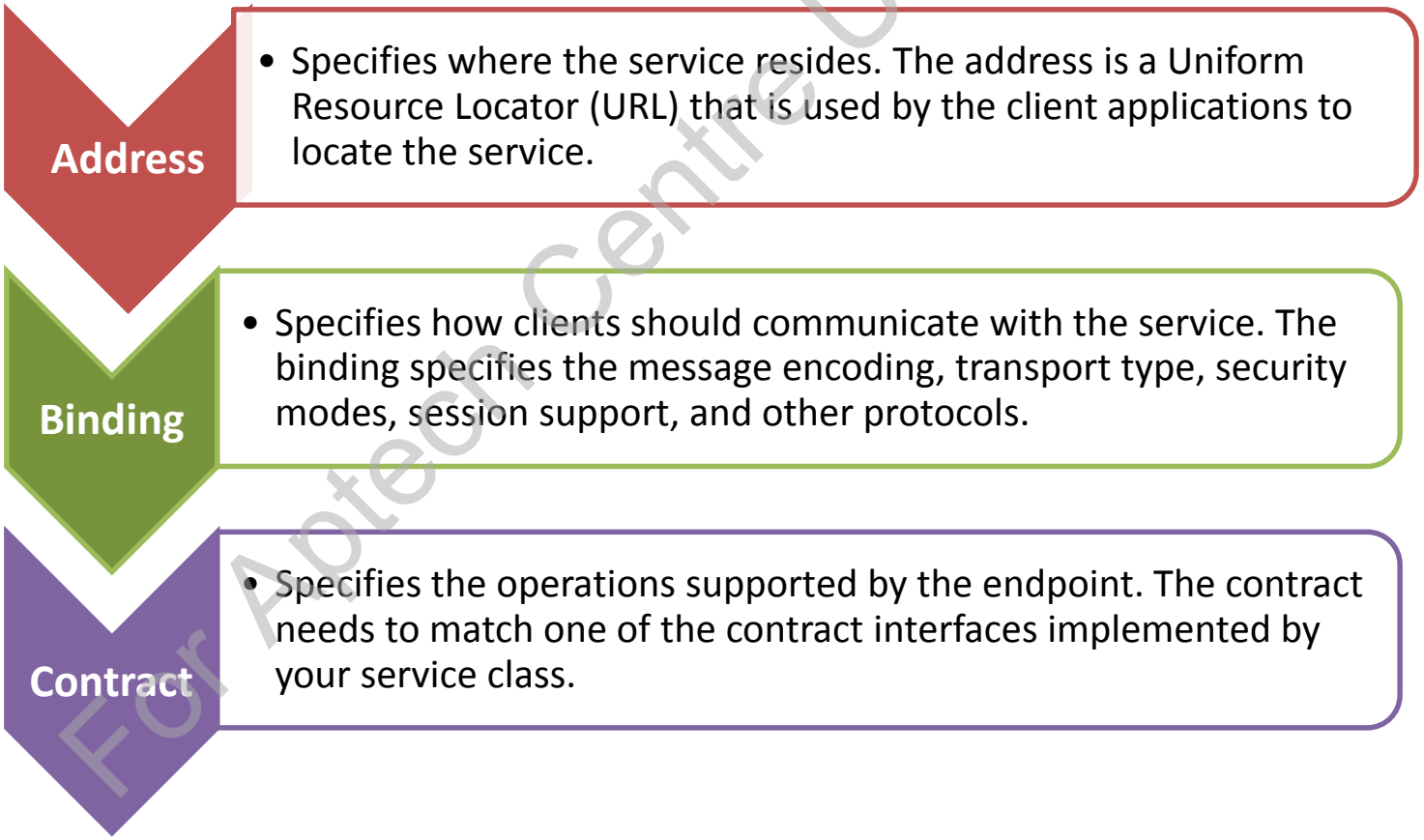
- Figure shows how a request is sent to a WCF service and how the service responds to the request:



- In the figure:
  - A client sends a request to the service.
  - The request goes as a message with one or more endpoints.
  - A service endpoint (comprising A-Address, B-Binding, and C-Contract) that defines how the service is exposed to the clients.

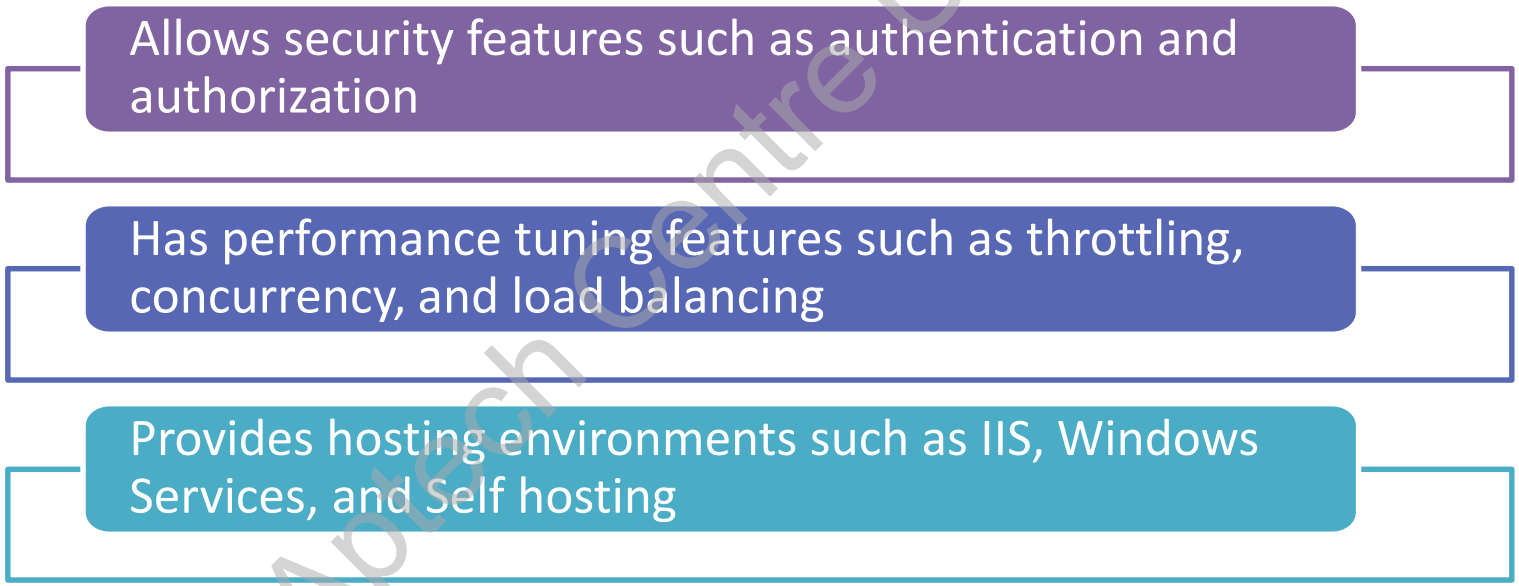
Overview of WCF Services 4-6

- The endpoint comprises A-Address, B-Binding, and C-Contract, each of which are defined as follows:

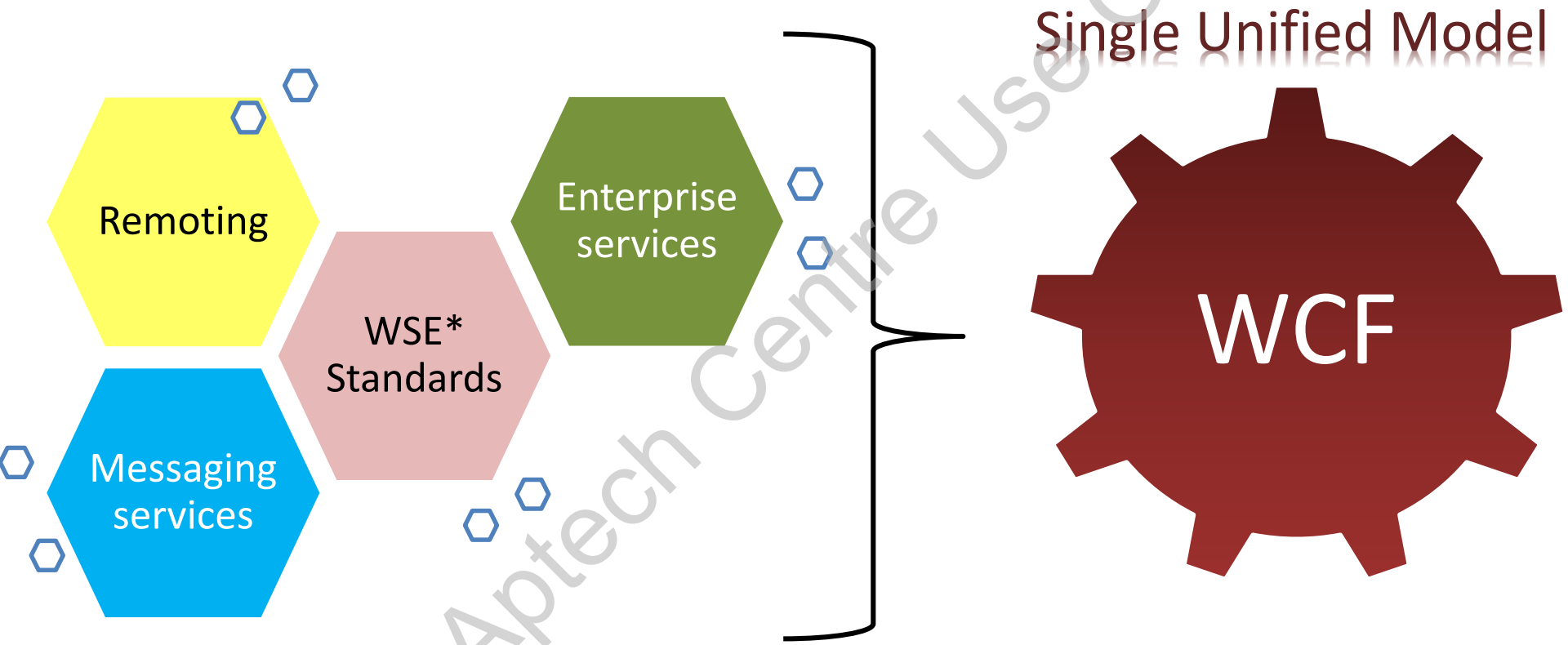


Overview of WCF Services 5-6

- Features provided by WCF are:

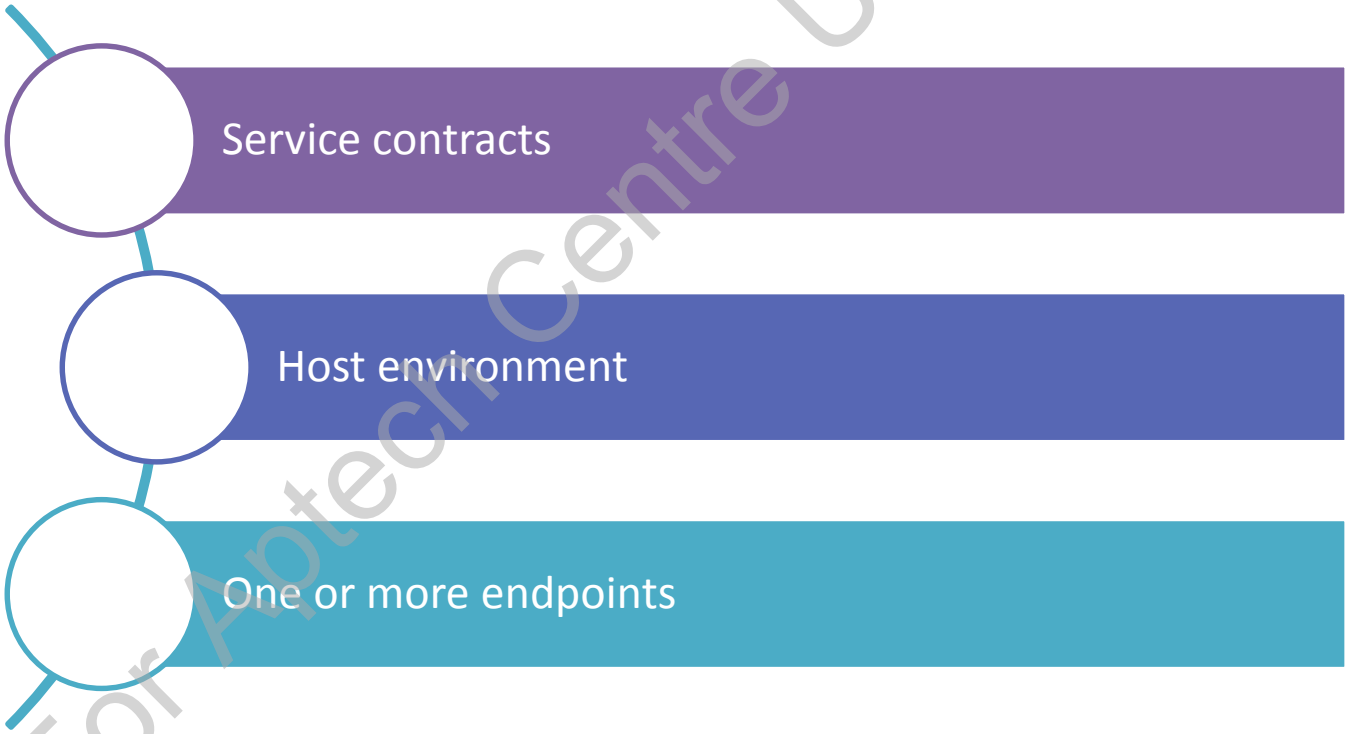


Overview of WCF Services 6-6



Components of WCF Services

- The three main components of a WCF service:



Service Contracts

- Contracts:
  - Define an important part of the architecture.
  - **Data contract**: consists of parameters that make up a message which a service can use.
  - **Message contract**: uses protocols like SOAP, which helps you acquire a finer control over the message.

- A Service contract consists of an interface with declarations of methods (in VB or C#) that will be used in them.

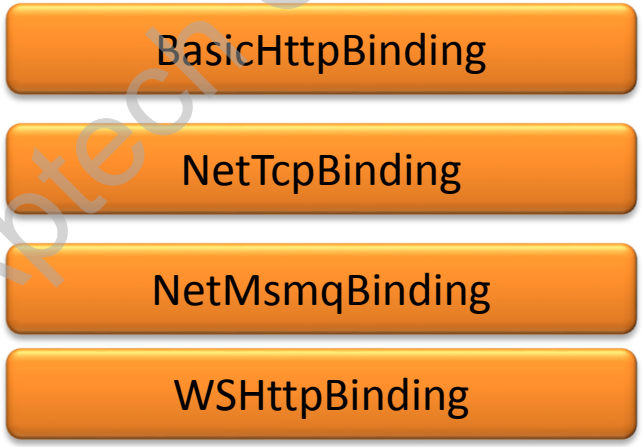
Host Environment

- Environment that hosts the developed service.
- Various ways to host a WCF service:

Bindings

Bindings

- Bindings apply the conditions required to communicate with a service.
- Various kinds of bindings that can be used based on your requirements:



Endpoints

- Endpoints are Universal Resource Identifiers (URI) which are exposed to the outside world.
- The client can connect to these WCF services from these endpoints.

Popularly Used Hosts

- Some of the popularly used hosts include:

Console Application	Windows Service	IIS	Windows Azure
• Simple Console application host	• WCF service can be controlled by the Service control manager	• WCF services can be hosted in IIS provided the service exposes at least one HTTP endpoint	• WCF services can be hosted on Windows Azure

ASP.NET Web API versus WCF Services

Following table shows the main differences between Web API and WCF:

Web API	WCF
Supports only HTTP and allows accessing from various mobile devices, browsers, and so on.	Supports transport protocols such as TCP, HTTP, UDP, and custom transports. It also allows switching between them.
Enables building Web APIs that support XML and JSON.	Supports Binary, Text, and MTOM encoding.
Uses basic protocol and formats such as XML, SSL HTTP, WebSockets, SSL, Jquery, and JSON.	Supports building services with WS-* standards such as Message Security, Transactions, and Reliable Messaging.
Allows describing a Web API such as auto-generated HTML.	Allows describing WCF SOAP services in WSDL.
Ships with .NET framework but is also available as open-source.	Ships with the .NET framework.

Working with WCF 1-10

- Endpoints:
  - Are a major infrastructure for communication in a WCF service.
  - Provide clients access to the functionality offered by a WCF service.

- Each endpoint consists of the following four properties:

Working with WCF 2-10

- The basic syntax for each of these in the configuration through `Web.config` is as follows:

Address

Syntax

```
<endpoint address="http://localhost/MyService"
binding="..." contract="..."/>
```

In this example, the `address` specifies the location of the endpoint.

Binding

Syntax

```
<endpoint address="http://localhost/MyService"
binding="wsHttpBinding" contract="..."/>
```

Here, the `wsHttpBinding` specifies the binding policy to be used.

Working with WCF 3-10

Contract

Syntax

```
<endpoint address="http://localhost/MyService"
binding="wsHttpBinding" contract="IMinfo"/>
```

Here, the contract `IMinfo` specifies contract service that needs to be accessed by the host.

Defining the Contracts

Following syntax helps to create a `ServiceContract` and `OperationContract` in code:

Syntax

```
[ServiceContract]
public interface <Name of interface>
{ }
[OperationContract]
void Operation1();
```

Working with WCF 4-10

Bindings

Following syntax helps you create a new binding instance:

Syntax

```
<specify binding type> binding = new <specify binding type>();
```

Endpoints

Following syntax helps you create endpoints in code:

Syntax

```
Uri MyUri = new Uri("<specify address>");
<specify binding type> binding = new <specify binding type>();
ServiceHost.AddEndpoint(typeof(<specify the service interface>), binding,
MyUri);
```

Working with WCF 5-10

- Steps to create and consume the service are:

Step 1

In SQL Server 2012, create a database named **Bank** with a table named `BankAccounts`. Insert sample records into the table.

```
create table BankAccounts
(
accno varchar(20),
accfname varchar(50),
acclname varchar(50),
acccontact varchar(50),
accamount float,
accactive int
);

insert into BankAccounts values('SBI1234','Charles','Rodrigues', '9876885577',69878,1);
insert into BankAccounts values('SBI1235','Maria','Lopes', '987698776',4500,1);
insert into BankAccounts values('SBI1236','Mark','DSa', '987688654',34000,1);
insert into BankAccounts values('SBI1237','Peter','Fernandes', '987764435',32378,1);
insert into BankAccounts values('SBI1238','George','Pereira', '9876885587',76878,1);
```

100 %

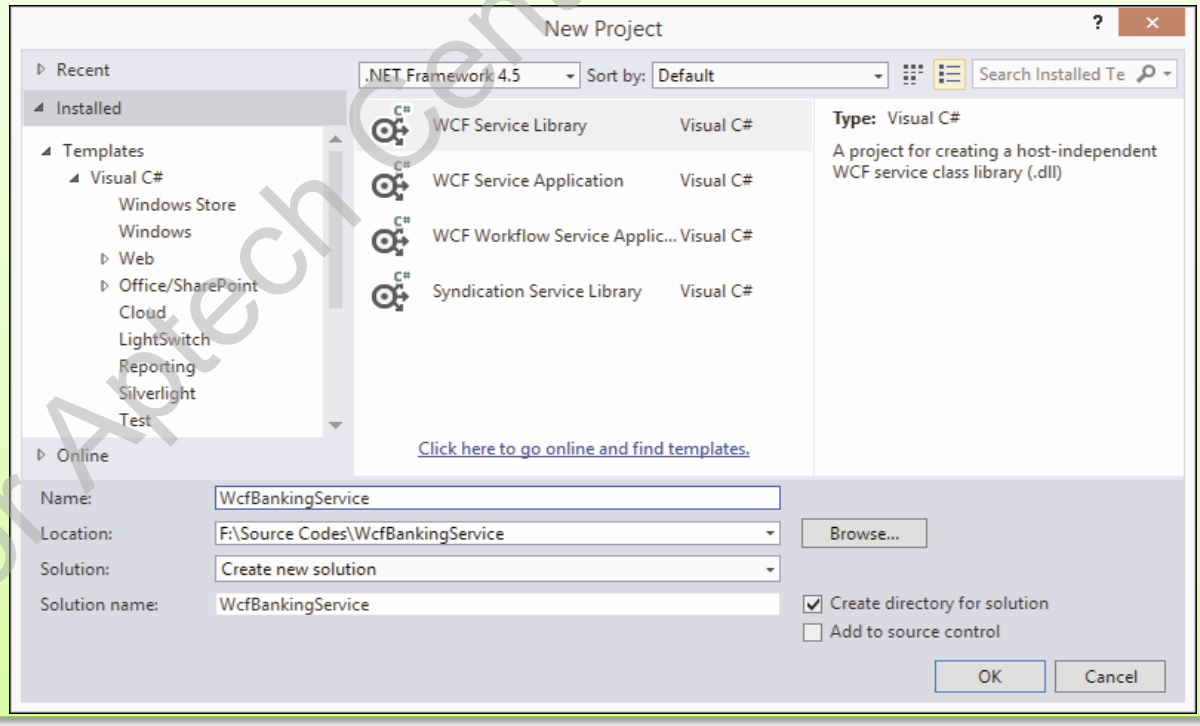
Results | Messages

	accno	accfname	acclname	acccontact	accamount	accactive
1	SBI1234	Charles	Rodrigues	9876885577	69878	1
2	SBI1235	Maria	Lopes	987698776	4500	1
3	SBI1236	Mark	DSa	987688654	34000	1
4	SBI1237	Peter	Fernandes	987764435	32378	1
5	SBI1238	George	Pereira	9876885587	76878	1

Working with WCF 6-10

Step 2 Launch Visual Studio 2013.

Step 3 Click **File → New → Project** and create a WCF Service Library `WcfBankingService` as shown in the following figure.



Working with WCF 7-10

Step 4

Rename these to `BankService.cs` and `IBankService.cs` respectively. Also, rename the class and the interface within the code. Ensure that you rename in `Web.config` as well.

Step 5

Add the following code to the `IBankService.cs` file.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
using System.Data.SqlClient
namespace WcfBankingService {
   [ServiceContract]
   public interface IBankService {
         [OperationContract]
         double WithDraw(double amount);
         [OperationContract]
         double Deposit(double amount);
         [OperationContract]
         double ShowBalance(string accno);
  }
}
```

Working with WCF 8-10

Step 6

Add the following code to `BankService.cs`. This code shows how to implement an `OperationContract`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
using System.Data.SqlClient;

namespace WcfBankingService {
  public class BankService : IBankService {
      public double WithDraw(double amount) {
         //Code to connect to the database and return balance amount
         // after deducting the withdrawal amount
         . . .
         double amt = 15000-amount;
         return amt;
      }
      public double Deposit(double amount)
      {
```

Working with WCF 9-10

```
        //Code to connect to the database and return the balance
        // after adding the deposit amount
        . . .
        double amt = 9000+amount;
        return amt;
    }
    public double ShowBalance(string accno) {
        //Code to connect to the database and return
        // the balance from the given accno
        . . .
        double balance = 2300;
        return balance;
    }
  }
}
```

Step 7 Run the service and test it by passing various amounts.

Working with WCF 10-10

- Following are the steps to create a Web application to consume the `BankingService`:

Step 1	• Create an ASP.NET Web application named **Banking**.
Step 2	• Create a Web form with components.
Step 3	• Add a service reference by using the shortcut menu options **Add** → **Service Reference**.
Step 4	• Enter the address of `WCFBankingService` and click **Go** to search for the WCF service. Specify the name as `BankServiceReference`.
Step 5	• Click **OK** to create the reference to this service.
Step 6	• Double-click **`ShowBalance`**. It will open the default code window.
Step 7	• Modify the default code.
Step 8	• Once the code is added, build and test the application.

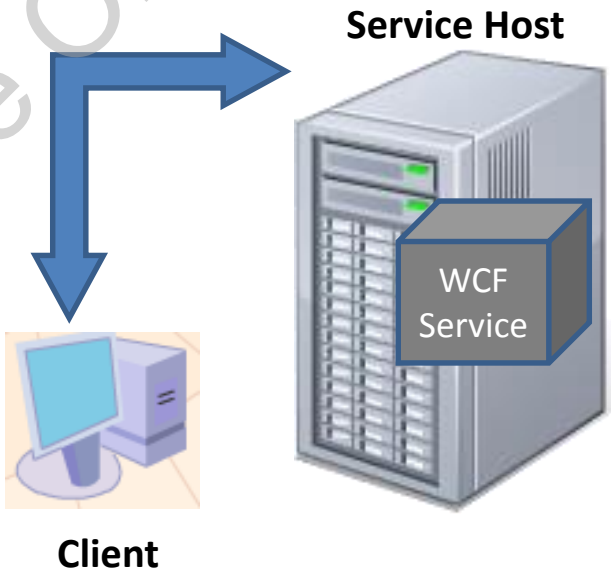
Types of Contract Attributes

- Following are the different types of contract attributes for WCF services:

Service Contract	Operation Contract	Data Contract	Fault Contract
• A WCF service contract is a standard interface. You specify an interface as a service contract by adding `[ServiceContract]` attribute above the interface definition.	• Every method that is to be exposed to the client must have the `OperationContract` attribute.	• This is declared as a class and then decorated it with the `DataContract` attribute and each of the properties to be serialized with the `DataMember` attribute.	• Use the `FaultContract` attribute in a service to define strongly typed exceptions as SOAP faults.

Configuring WCF Services 1-3

- The service host:
  - Prepares the service implementation class that will be addressed by clients.
  - Opens ports and listens to requests according to the configuration.
  - Manages the incoming requests of service and allocates resources such as memory and threads.



Configuring WCF Services 2-3

- Two ways to host a WCF service are:

Self-hosting	• Application itself hosts the service. For instance, Windows Presentation Foundation (WPF) application or a Windows Service. • The host will start when application starts, and shutdown along with application.
Web hosting	• When the user uses IIS to host WCF service, it is known as Web Hosting. • The service is invoked after IIS receives the first request to the service, and shut down when the Web application shuts down.

Configuring WCF Services 3-3

- Following code demonstrates how to host a WCF service with the `ServiceHost` class:

```
ServiceHostbankServiceHost = new ServiceHost(typeof
            (Services.WcfBankingService));
bankServiceHost.Open();

Console.WriteLine("Service has been hosted. Press Enter to stop");
Console.ReadLine();
bankServiceHost.Close();
```

- In the code:
  - Each `ServiceHost` can manage a single service type, but it opens many listeners for that service type, each with a different listening on different protocol and port.
  - The configuration can be set by using code itself, before calling the `Open` method, or it can be specified in the application configuration file (`app.config`).
  - After the service host has opened, you can close it at any time by calling the `Close` method. The `Close` method will stop the host from listening to any communication, making the service unavailable for clients.

Hosting WCF Services 1-5

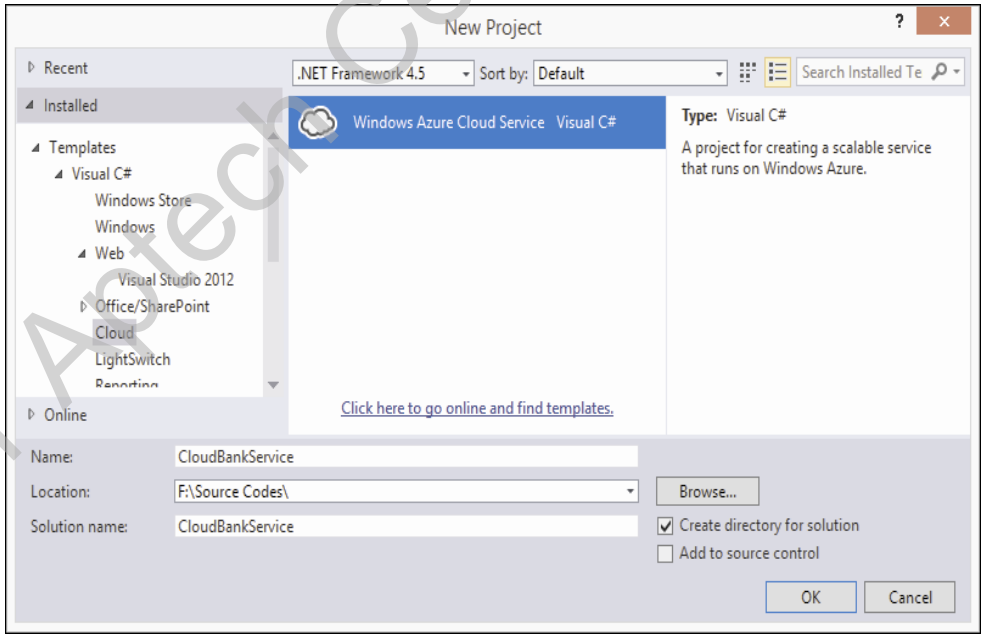
- Steps to create a new cloud service with a WCF Worker or Web Role and host it on Azure:

Step 1

Launch Visual Studio 2013.

Step 2

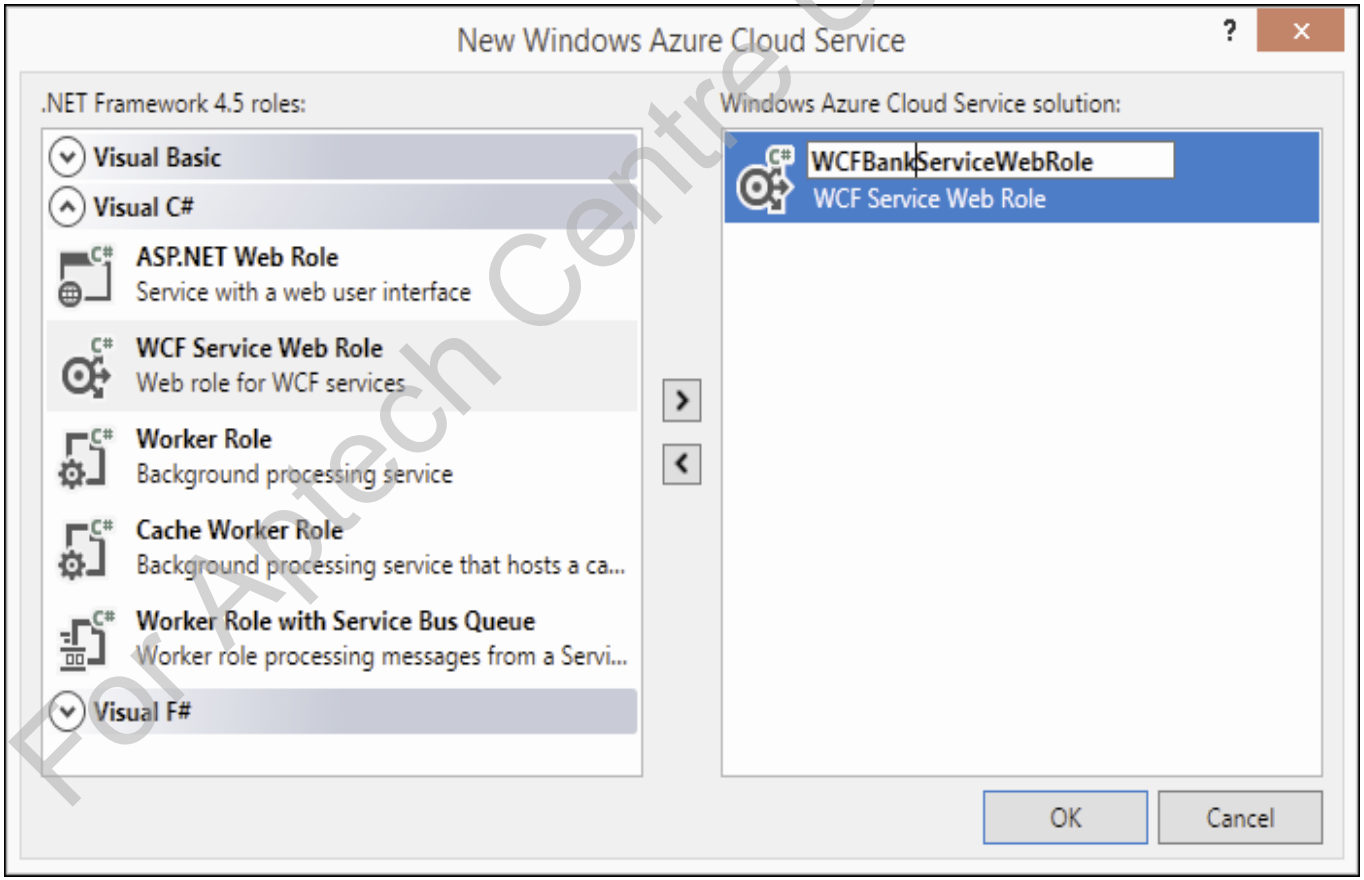
Click **File → New → Project** and create a Windows Azure Cloud service called `CloudBankService` as shown in the figure.



Hosting WCF Services 2-5

Step 3

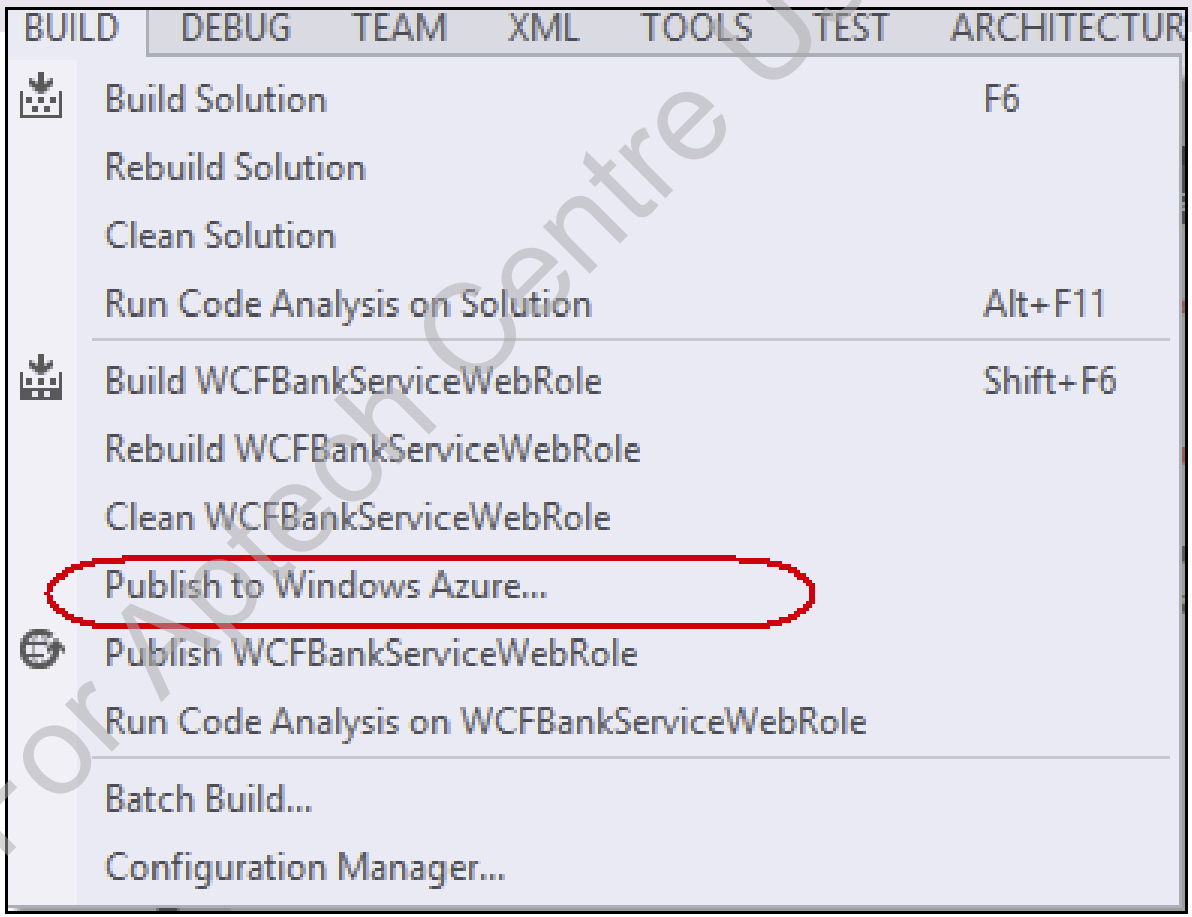
In the **New Windows Azure Cloud Service** dialog box, select **WCF Service Web Role** in the left pane and rename it to **WCFBankServiceWebRole** as shown in the figure.



Hosting WCF Services 3-5

Step 4

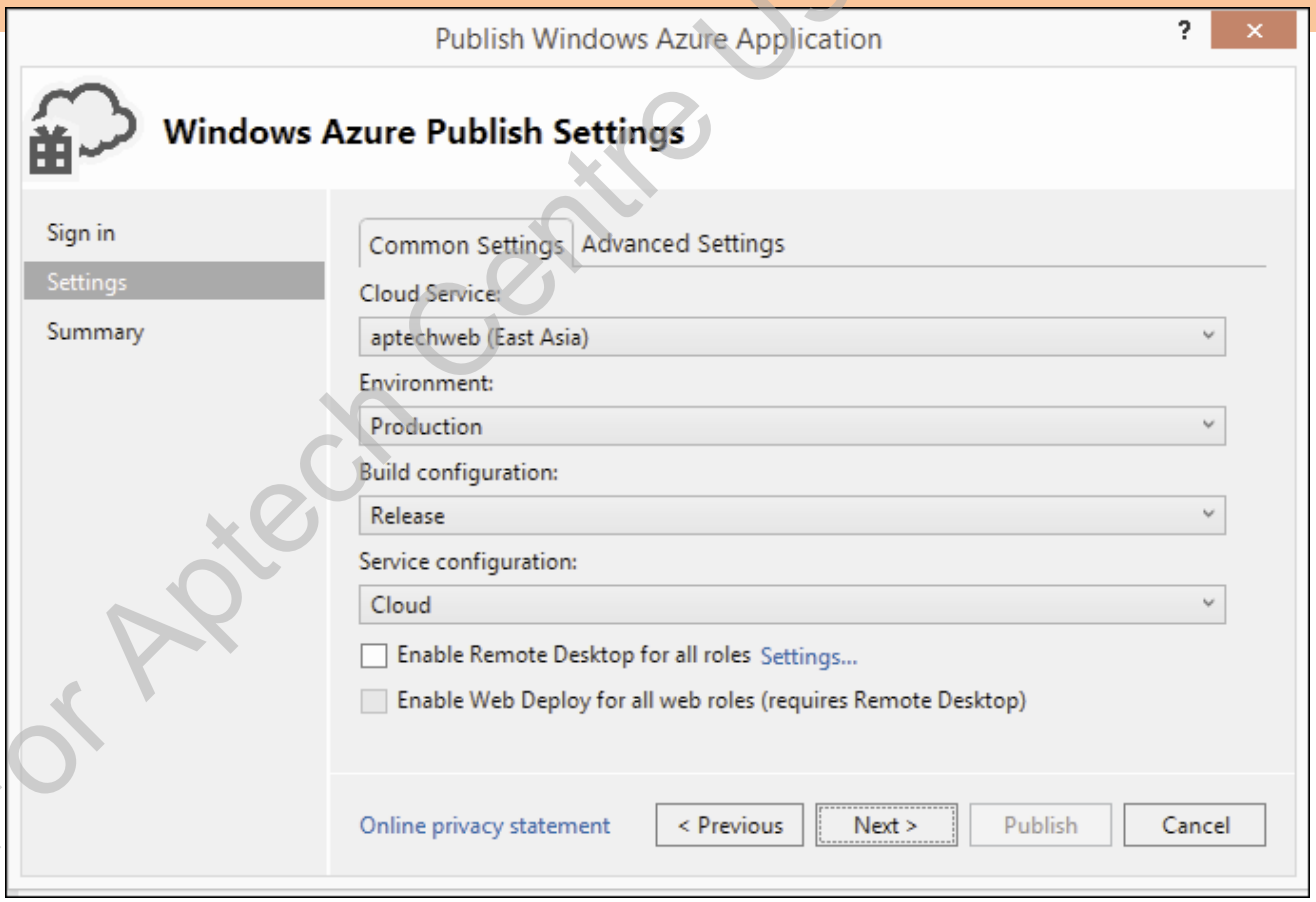
After the service is successfully created and code is added, select **Publish to Windows Azure** option on the **BUILD** menu as shown in the figure.



Hosting WCF Services 4-5

Step 5

In the **Publish Windows Azure Application** dialog box, select or specify appropriate settings as shown in the figure.



Hosting WCF Services 5-5

Step 6

- When all the settings are done, click **Publish** in the dialog box. The application will be successfully published.

Step 7

- Launch and sign in to the Windows Azure Management portal.

Step 8

- When you click the site address (for example, http://aptechweb.cloudapp.net), you will see the service listed there. This means that it has been successfully hosted in Windows Azure.

Step 9

- To use the service, you can then add it as a service reference in any Web application using the appropriate URL (for example, http://aptechweb.cloudapp.net/BankService.svc).

Summary

- WCF is a unified .NET framework for building service-oriented applications.
- WCF supports protocols and transports such as SOAP with HTTP, TCP/IP, and Named pipes.
- A WCF service is composed of three main components, namely, Service contracts, Host environment, and one or more endpoints.
- A service endpoint defines how the service is exposed to the clients.
- A service contract describes the functionality that is exposed by the service.
- A data contract is declared as a class and then decorated with the `DataContract` attribute and each of its properties are serialized with the `DataMember` attribute.
- The service host is responsible for opening ports and listening to requests according to the configuration.