In WCF, all services expose contracts. The contract is a platform-neutral and standard way of describing what the service does.
WCF defines four types of contracts.

Service contracts

Describe which operations the client can perform on the service.
There are two types of Service Contracts.
ServiceContract - This attribute is used to define the Interface.
OperationContract - This attribute is used to define the method inside Interface.

```
[ServiceContract]

interface IMyContract
{

   [OperationContract]

   string MyMethod( );

}

class MyService : IMyContract
{

   public string MyMethod( )

   {

      return "Hello World";

   }

}
```

Data contracts

Define which data types are passed to and from the service. WCF defines implicit contracts for built-in types such as int and string, but we can easily define explicit opt-in data contracts for custom types.

There are two types of Data Contracts.
DataContract - attribute used to define the class
DataMember - attribute used to define the properties.

```
[DataContract]

class Contact

{

   [DataMember]

   public string FirstName;



   [DataMember]

   public string LastName;

}
```

If DataMember attributes are not specified for a properties in the class, that property can't be passed to-from web service.

Fault contracts

Define which errors are raised by the service, and how the service handles and propagates errors to its clients.

Message contracts

Allow the service to interact directly with messages. Message contracts can be typed or untyped, and are useful in interoperability cases and when there is an existing message format we have to comply with.

Types of Binding

Let us see more detailed on predefined binding

BasicHttpBinding

- It is suitable for communicating with ASP.NET Web services (ASMX)-based services that comfort with WS-Basic Profile conformant Web services.
- This binding uses HTTP as the transport and text/XML as the default message encoding.
- Security is disabled by default
- This binding does not support WS-* functionalities like WS- Addressing, WS-Security, WS-ReliableMessaging
- It is fairly weak on interoperability.

WSHttpBinding

- Defines a secure, reliable, interoperable binding suitable for non-duplex service contracts.
- It offers lot more functionality in the area of interoperability.
- It supports WS-* functionality and distributed transactions with reliable and secure sessions using SOAP security.
- It uses HTTP and HTTPS transport for communication.
- Reliable sessions are disabled by default.

WSDualHttpBinding

This binding is same as that of WSHttpBinding, except it supports duplex service. Duplex service is a service which uses duplex message pattern, which allows service to communicate with client via callback.

In WSDualHttpBinding reliable sessions are enabled by default. It also supports communication via SOAP intermediaries.

WSFederationHttpBinding

This binding support federated security. It helps implementing federation which is the ability to flow and share identities across multiple enterprises or trust domains for authentication and authorization. It supports WS-Federation protocol.

NetTcpBinding

This binding provides secure and reliable binding environment for .Net to .Net cross machine communication. By default it creates communication stack using WS-ReliableMessaging protocol for reliability, TCP for message delivery and windows security for message and authentication at run time. It uses TCP protocol and provides support for security, transaction and reliability.

NetNamedPipeBinding

This binding provides secure and reliable binding environment for on-machine cross process communication. It uses NamedPipe protocol and provides full support for SOAP security, transaction and reliability. By default it creates communication stack with WS-ReliableMessaging for reliability, transport security for transfer security, named pipes for message delivery and binary encoding.

NetMsmqBinding

- This binding provides secure and reliable queued communication for cross-machine environment.
- Queuing is provided by using MSMQ as transport.
- It enables for disconnected operations, failure isolation and load leveling

NetPeerTcpBinding

- This binding provides secure binding for peer-to-peer environment and network applications.
- It uses TCP protocol for communication
- It provides full support for SOAP security, transaction and reliability.

WCF :

Windows Communication Foundation (WCF) is an SDK for developing and deploying services on Windows. WCF provides a runtime environment for services, enabling you to expose CLR types as services, and to consume other services as CLR types.
WCF is part of .NET 3.0 and requires .NET 2.0, so it can only run on systems that support it.