WCF是微软分布式应用程序开发的集大成者，它整合了.Net平台下所有的和分布式系统有关的技术，例如.Net Remoting、ASMX、WSE和MSMQ。以通信(Communiation)范围而论，它可以跨进程、跨机器、跨子网、企业网乃至于 Internet；以宿主程序而论，可以以ASP.NET，EXE，WPF，Windows Forms，NT Service，COM+作为宿主(Host)。

# Windows Communication Framework

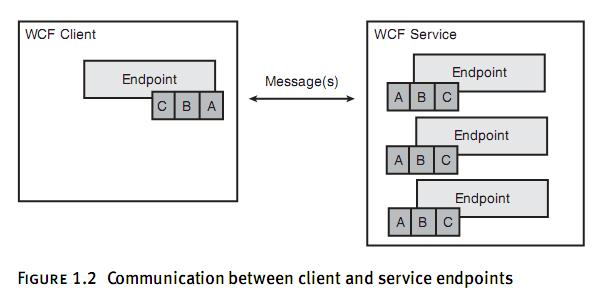
## Basic

WCF is all about services. In short, it’s all about putting distributed computing within reach of professional software developers

 a service is a set of endpoints that provide useful capabilities to clients.

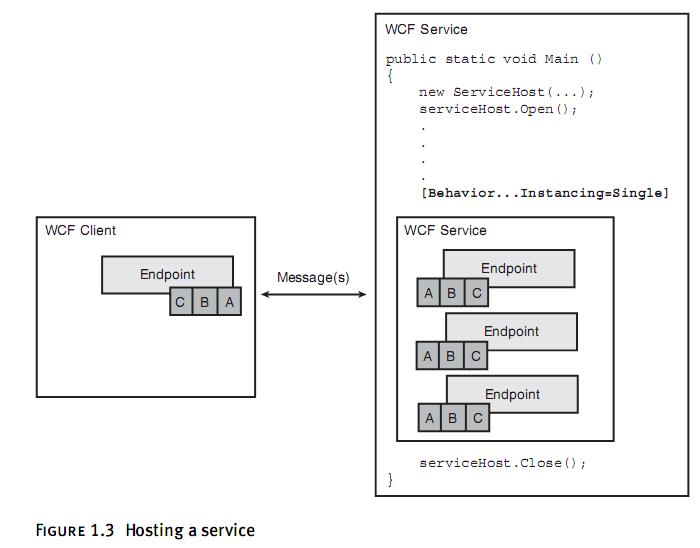
Client needs to know the ABCs about a service: the address (where), the binding (how), and the contract (what).

The binding deﬁnes the channel used to communicate with an endpoint. WCF ships with system-provided bindings that have the channels stacked and conﬁgured correctly.



For discoverability, a service may include an infrastructure endpoint called the Metadata Exchange (MEX) endpoint. This endpoint is accessible by clients to obtain the ABCs of the service and returns Web Service

Description Language (WSDL).



### Impl a  WCF service that is hosted in a console application

**1. Deﬁne the contract,**

The [ServiceContract] attribute marks a class as a contract. The [OperationContract] attribute deﬁnes methods that can be invoked on the class through the service interface

**2. Deﬁne an endpoint**

by using the AddSErviceEndPoint method on the ServiceHost class.

**3. Host the service in a process so it is listening for incoming requests**

### Writing a Service with Code and Configuration Files

You still need to code the feature or algorithm you’re exposing in the service, but endpoint addressing, bindings, and behaviors can be moved from the code into conﬁguration ﬁles.

**1. Deﬁne the contract**

**2. Host the service in an operating system process so it can be accessed by a client on the network.**

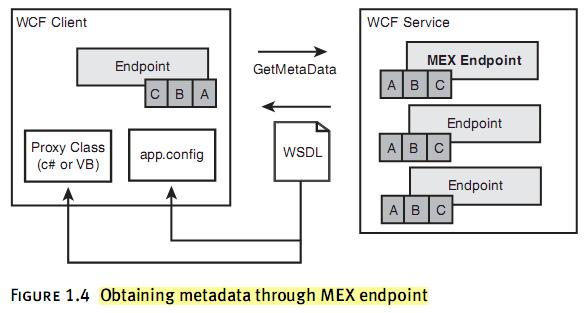
**3. Deﬁne a conﬁguration ﬁle that speciﬁes the base address for a service and the ABCs of the service endpoint**



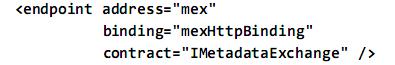
When the service is started by a host, WCF starts the listener, which listens on this address for incoming requests.

In the majority of situations, IIS is the right hosting environment to achieve great performance, manageability, and security. Another choice is Windows Process Activation Services.

MEX(MetaData Exchange) Endpoint



By default, WCF services do not expose a MEX endpoint.



When messages arrive, the ServiceHost does a few things. First, based on the channel stack deﬁned by the binding, it applies any decryption, decompression, and security rules. Second, based on the contract, it deserializes the incoming message into .NET types, creates a new object, and then invokes the proper operation on the object.

With MEX endpoint, VS can generate stub without sharing the interfaces. There are two steps to writing a client that invokes a service: ﬁrst, gen-erate a conﬁguration ﬁle and the proxy class, and second, write code that uses the proxy class to invoke the service.

ASMX is replaced by WCF in .NET 3.5 as the recommended way to publish Web services in IIS.

When hosting in IIS, the address of a service is deﬁned by the virtual directory that contains the service ﬁles. The binding will always use the HTTP/S protocol because that’s what IIS understands, so basicHttpBinding and wsHttpBinding are available.

## Contracts

**契约本身只是定义了服务，数据与消息的格式，实际的传输可以使用不同的协议！**

A contract is a description of the messages that are passed to and from service endpoints.

* **Service contracts.** Service contracts describe the functional operations implemented by the service. A service contract maps the class methods of a .NET type to WSDL services, port types, and operations. **Operation contracts** within service contracts describe the service operations, which are the methods that implement functions of the service.
* **Data contracts**. Data contracts describe data structures that are used by the service to communicate with clients. A data contract maps CLR types to XML Schema Deﬁnitions (XSD) and deﬁnes how they are serialized and deserialized. Data contracts describe all the data that is sent to or from service operations.
* **Message contracts**. Message contracts map CLR types to SOAP messages and describe the format of the SOAP messages and affect the WSDL and XSD deﬁnitions of those messages. Message contracts provide precise control over the SOAP headers and bodies. One of the advantages of message-contracts is that you can set privacy against members, but in many cases this isn't necessary.

Windows Communication Foundation 消息传递体系结构的主要目的之一是，在提供统一编程模型的同时，还允许灵活地表示数据和传递消息。这是基于将 XML 作为数据模型以及将 SOAP 和 WS-Addressing 作为消息传递框架而实现的。但是，Windows Communication Foundation 构建在这些模型基础上这一事实，并不意味着在传递消息时必须使用 XML 1.0、SOAP 或 WS-Addressing。

任何消息传递框架的另一个主要功能是，通过任意标头扩展消息负载。标头不过是随消息传递的额外信息，用于实现其他的消息处理功能（如安全性、可靠的消息传 递和事务）。对于 XML 消息，这意味着用 XML 标头扩展 XML 负载（两者都表示为在容器元素中分帧的 XML 元素）

SOAP 框架使得定义基于 XML 的协议（可通过任何传输使用，而不依赖于任何传输特定的功能）成为可能.

Message 类在本质上模拟消息正文以及消息标头和属性的集合。可用方法主要用于创建消息、读写消息正文以及操作标头和属性的集合。

消息契约和数据契约一样，都是定义在数据（而不是功能）类型上。不过数据契约旨在定义数据的结构（将数据类型与XSD进行匹配），而消息契约则更多地关注于数据的成员具体在SOAP消息中的表示

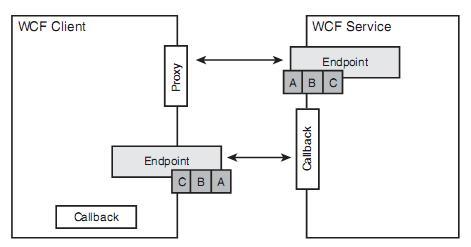
here is a preference to use data contract, followed by message contract and XML serialization. However, the ability to adapt to a preset form is ranked in the reverse order. **When you're required to conform to an existing message layout, you are more likely to be forced to use message contracts. When you are designing new message layouts, you almost always want to use data contracts.**

### **Service Contract**

三种消息交换方式：

1. **Request/Response:**
   * Synchronous/Asynchronous, 缺省的消息交换模式
2. **One-way:**
   * 只有客户端发起请求，服务端并不会对请求进行回复
3. **Duplex-way**
   * 和request/reply模式类似，也是有来有往。它可以在处理完请求之后，通过请求客户端中的回调进行响应操作
   * 消息交换过程中，服务端和客户端角色会发生调换
   * 服务端处理完请求后，返回给客户端的不是reply，而是callback请求。

A duplex service contract implements the duplex message pattern, in which unsolicited messages can be sent in either direction after the communication channel is established. Both parties need an address, binding, and contract deﬁning where, how, and what messages can be sent. If the initial channel cannot support bidirectional communication, then WCF creates a second channel, using the same protocol as was speciﬁed by the service’s endpoint.

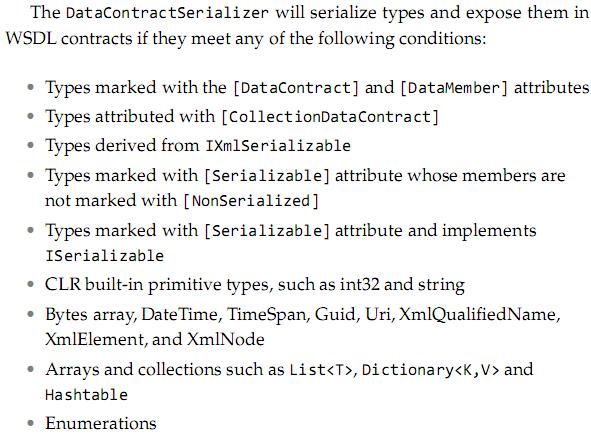


The major difference between the service-side endpoint and the client-side endpoint is that the client-side endpoint is created dynamically by WCF. There is no conﬁguration ﬁle or explicit ServiceHost call in the client code.

An endpoint can have only one contract, but a contract can be referenced by many endpoints.

### Data Contract

 Inside a WCF service, application data is represented in simple and complex types while outside the service data is represented by XMLSchema Deﬁnitions (XSD).



Define class hierarchy using XSD, with XML extension.



### Message Contract

Whereas data contracts enable interoperability through the XMLSchema Deﬁnition (XSD) standard, message contracts enable you to interoperate with any system that communicates through SOAP.

Typedmessages use MessageHeader  and MessageBodyMember  attributes to describe the structure of the SOAPheader and body. Untypedmessages do not use any attributes to describe their contents. It’s left entirely up to the runtime code to make sense of the contents.

**Typed Message**

When using message contract, both input and output parameters must be messages. More speciﬁcally, operations must contain exactly one input parameter and must return exactly one result, both of which are messages, because the request and Message Contracts response messages sent to and from the operation will map directly their SOAP representation. In addition, essage-based programming and parameter-based programming cannot be mixed,

**Untyped Message**

Untyped operation contracts enable the client and service to pass virtually any content in the SOAPbody,

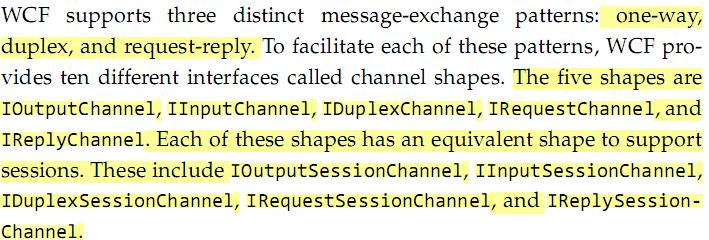
The Message class has numerous methods for creating, reading, and writing the message contents.The client is responsible for creating a message before sending it to the service and the service is responsible for creating a message to send back. Both client and server need know the content inside the message.

## Channels

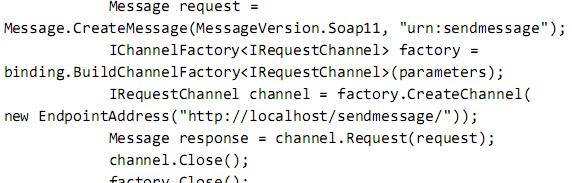
Channels are deﬁned for transports, protocols, and message interception. Channels are layered together to create a channel stack.

The goal of a channel stack is to transform a message into a wire format compatible with the sender and receiver and to transport the message. There are two types of channels that are used to do this: transport channels and protocol channels. Transport channels always sit at the bottom of the channel stack and are responsible for transporting messages using a transport protocol.  Protocol channels reside on top of other channels, they are often referred to as layered channels.

传输Channel在不同的层上，效率就不一样。注意网络7层架构。HTTP协议是在应用层上的；而TCP协议是在传输层上的，效率更高。综合来说大多数channel最后会经过解码过后通过TCP传输. netTCPBinding是 WCF新增的channel，专门用于WCF<->WCF的传输.



An example of the request-response shape in C# code?



Other formsof communication, such as one-way and duplex over HTTP, are done through shape changing. This is done by layering a protocol channel on top of the transport channel to support one-way or duplex communication.

