

Modèle de Service ABC et WCF

Jean-Yves Tigli

Polytech of Nice - Sophia Antipolis University

Email : tigli@polytech.unice.fr

Introduction to Windows Communication Foundation (WCF)

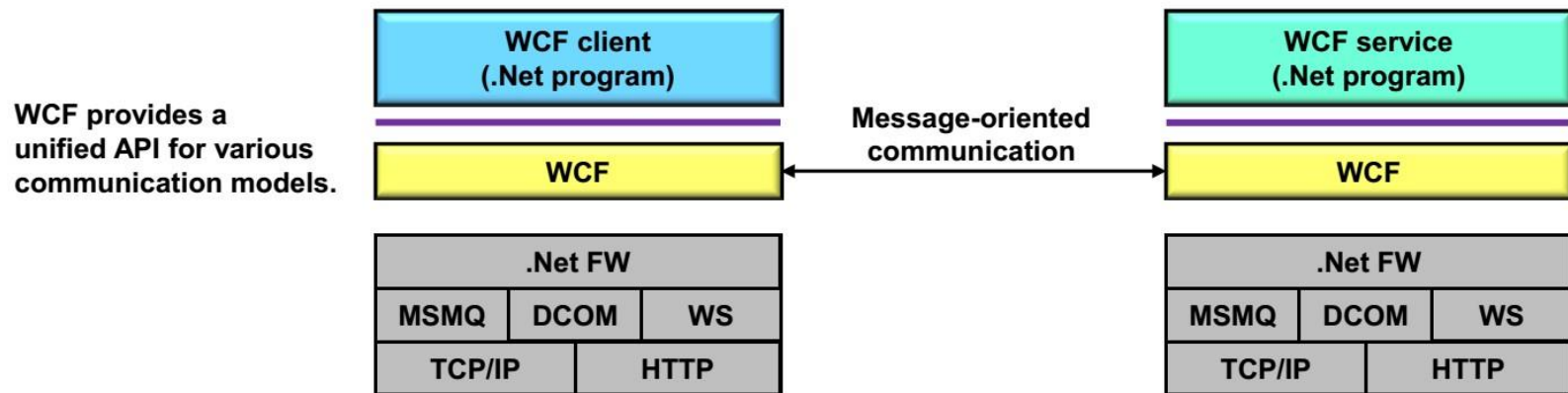
- WCF est un cadre unifié pour les applications distribuées, avec son propre modèle de programmation et son API.
- C'est un standard du Microsoft .Net Framework
- Les caractéristiques de WCF sont :
 - Un modèle de programmation orienté service
 - Une intéropérabilité avec des services non WCF grâce à l'inclusion des standards du domaine
 - Une extensibilité qui permet à un client / serveur d'être configuré pour intéropéré avec REST, SOAP, JSON, XML, et autres standards binaires.

Introduction à Windows Communication Foundation

- Modèle de programmation unifiée, orienté service
- Remplace et étend :
 - .NET Remoting
 - DCOM
 - ASP.NET Web services
 - MSMQ (Queued Messaging)
 - .NET Enterprise Services
- Neutralité dans les protocoles, flexibilité, extensibilité

Introduction à Windows Communication Foundation

- WCF utilise de nombreux types de communication
- WCF est au sommet du framework .Net

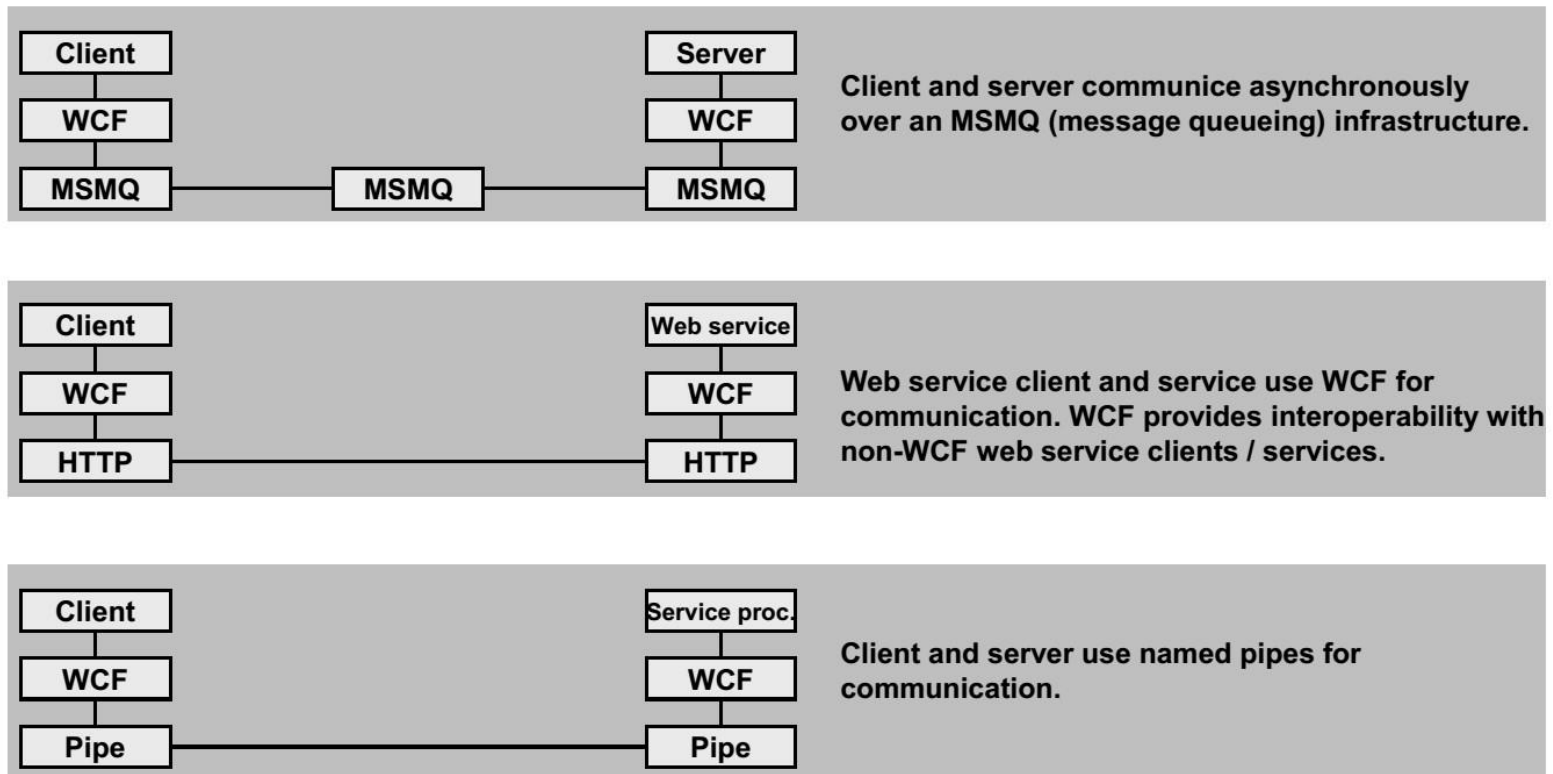


- En fait WCF est une approche orientée service qui va au-delà des Web Service
- En séparant les notions de contrat, de protocole et de format de message
- En séparant une description abstraite d'un service et son implémentation

Introduction à Windows Communication Foundation

WCF peut donc s'appuyer sur de multiples protocoles de communication

Exemples :



Modèle ABC de Service

- Le concept central de WCF est la notion de service
- Un service WCF est défini à travers la trilogie ABC :
 - L'adresse **A** où le service est disponible (l'URI du endpoint dans le cas d'un WS)
 - Le **binding B** ou comment accéder au service (quel protocole de transport ? Avec quel format ?)
 - Le **contrat C** ou que fournit l'interface du service (opérations, types de données ...)

WCF term	Question	WSDL element
A (Address)	<i>Where</i>	<service> including element <endpoint>
B (Binding)	<i>How</i>	<binding>
C (Contract)	<i>What</i>	<types> <interface>

- (A,B,C) est alors un endpoint de service, soit tous les éléments pour sa implementation et déploiement

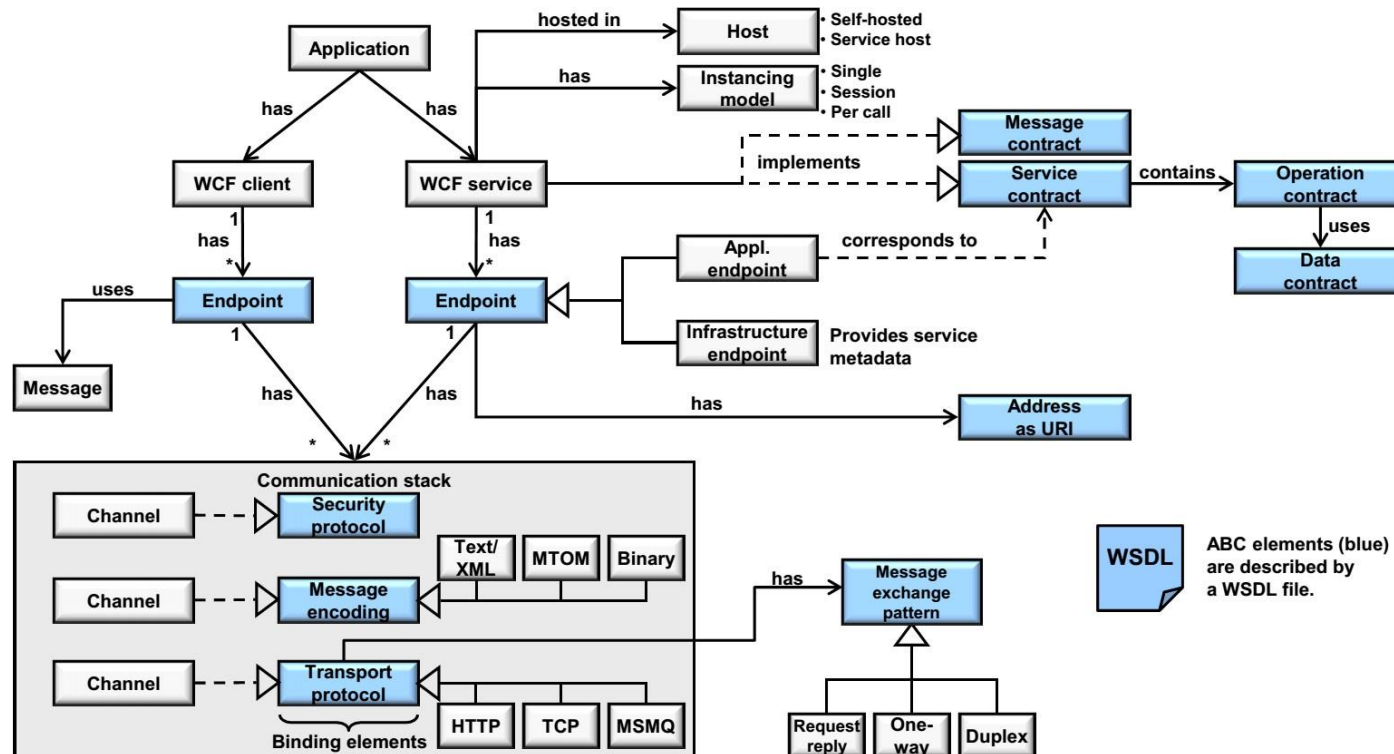
Le modèle de programmation WCF (Classes et Interfaces)

Dans le framework WCF les éléments ABC de WCF sont disponibles comme des classes spécifiques du Framework

WCF term	Corresponding class library (namespace)
A (Address)	System.Uri
B (Binding)	System.ServiceModel E.g. BasicHttpBinding (SOAP, non-secure, interoperable, non-duplex) WebHttpBinding (REST-style binding, i.e. non-SOAP)
C (Contract)	Interfaces / classes annotated with System.ServiceModel attributes: [OperationContract] [ServiceContract] [MessageContract] Data contract (definitions of types used in operation contracts): [DataContract] (System.Runtime.Serialization)
E (Endpoint)	System.ServiceModel.ServiceEndpoint

Dans le framework WCF...

Il vaut mieux procéder avec une fichier de configuration qui définiera les classes utilisées pour un binding donné ..



C'est l'idée du fichier App.config que nous verrons plus tard ...

Le « A » de ABC

L'adresse WCF définit où le service est disponible

C'est une url (uri) * ...

L'URI d'adresse de la plupart des transports se compose de quatre parties :

Par exemple, cet URI <http://www.fabrikam.com:322/mathservice.svc/secure>

L'URI possède les quatre parties suivantes :

Schéma : http:

Serveur: www.fabrikam.com

(Facultatif) Port : 322

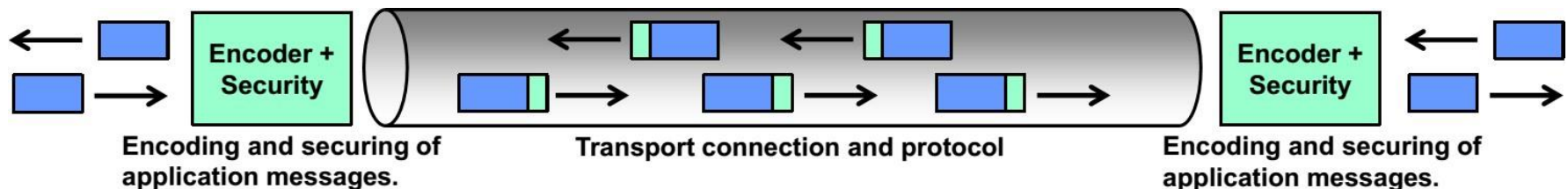
Chemin d'accès : /mathservice.svc/secure

Et potentiellement des paramètres ?param1=val1& param2=val2 & ...

(*) <https://www.w3.org/Submission/ws-addressing/>

Le « B » de ABC

- Le binding est le moyen par lequel le endpoint est accessible
- Il contient les éléments suivants :
 - Le protocole de transport
 - Exemples: TCP, HTTP, MSMQ
 - Le format des messages
 - Exemples: Text/XML (SOAP), binary, MTOM (Message Transfer Optimized Mechanism).
 - Les paramètres de configuration de la sécurité :



WCF fournit les binding suivants :

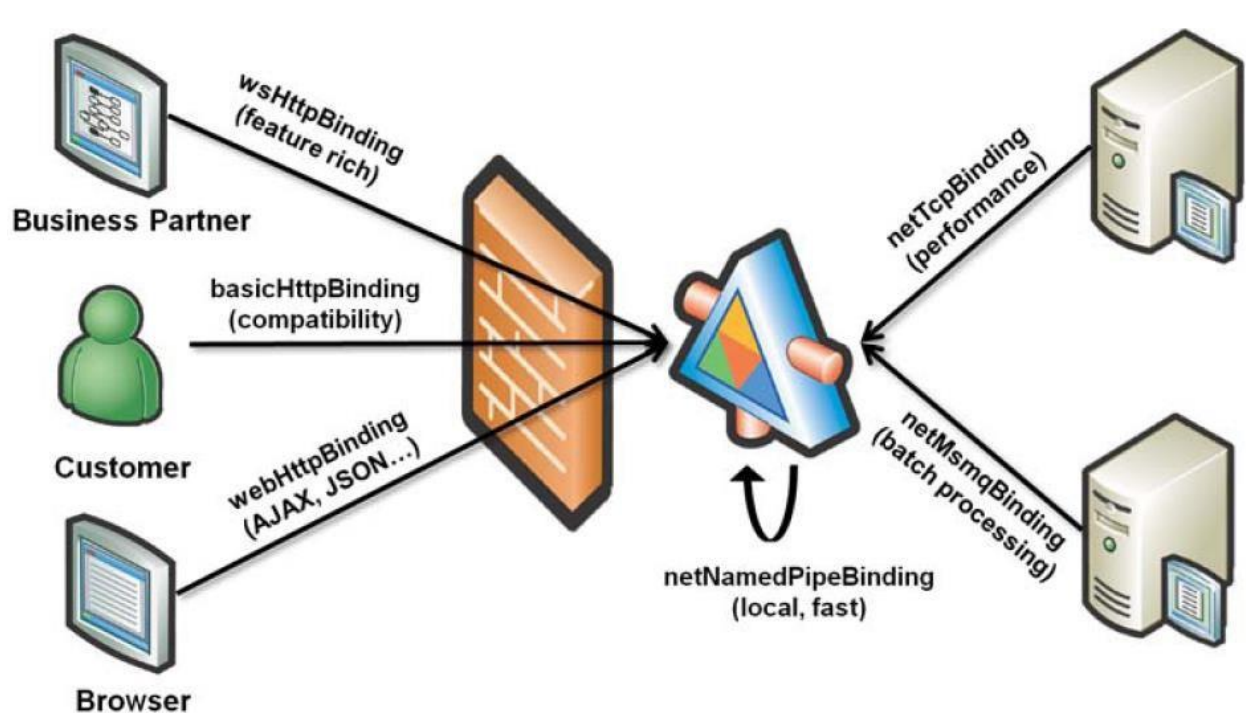
Binding	Interoperability	Security	Session	Transactions	Duplex	Encoding
BasicHttpBinding	WS-I Basic Profile	N, T, M, m	N	N	No	Text, MTOM
WSHttpBinding	WS-* standards	T, M, m	N, RS, SS	N, Yes	No	Text, MTOM
WSDualHttpBinding	WS-* standards	M, m	RS, SS	N, Yes	Yes	Text, MTOM
WSFederationHttpBinding	WS-Federation	N, M, m	RS, SS	N, Yes	No	Text, MTOM
NetTcpBinding	.NET	T, M, m, N	TS, RS, SS	N, Yes	Yes	Binary
NetNamedPipeBinding	.NET	T, N	N, TS	N, Yes	Yes	Binary
NetMsmqBinding	.NET (WCF)	M, T, N	N, TS	N, Yes	No	Binary
NetPeerTcpBinding	.NET	T	N	N	Yes	N/A
MsmqIntegrationBinding	MSMQ	T	N	N, Yes	No	MSMQ
BasicHttpContextBinding	WS-I Basic Profile	N, T, M, m	N	N	No	Text, MTOM
NetTcpContextBinding	.NET	N, T, M, m	T, RS, SS	N, Yes	Yes	Binary
WSHttpContextBinding	WS-* standards	T, M, m	N, RS, SS	N, Yes	No	Text, MTOM
WebHttpBinding	HTTP (REST)	N	N	N	No	POX

More details on WCF bindings see <http://msdn.microsoft.com/en-us/library/ms730879.aspx>.

Key:

N:	None	RS:	Reliable Session (WS-ReliableMessaging)	POX:	Plain Old XML
T:	Transport	SS:	Security Session		
M:	Message	TS:	Transport Session		
m:	mixed				

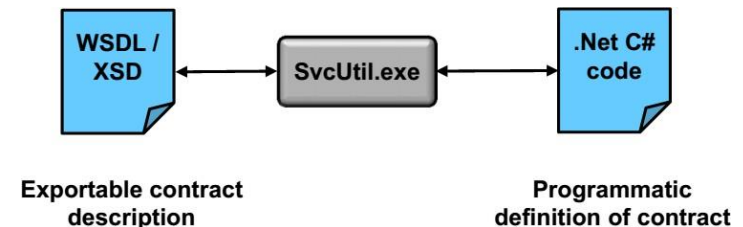
Bien choisir son ou ses Binding(s) ...



Le « C » de ABC

- Les interfaces WCF sont décrites par des contrats
- Les contrats définissent des opérations, des structures de données et des messages
- Un contrat de service est principalement défini par :
 - Un groupement d'opérations dans un service (le .Net attribute [ServiceContract])
 - Les signatures des opérations (le .Net attribute [OperationContract])
 - Les types de données des opérations (le .Net attribute [DataContract])
- A partir d'un contrat explicite on peut produire du code ou vice-versa (outil SvcUtil.exe)

SvcUtil.exe récupère la définition du service à partir d'un fichier WSDL ou directement à partir d'un service en cours d'exécution via par exemple un protocole de norme WS- MetadataExchange



Exemple de Contrat dans une classe IService1

Exposition d'opérations

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

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);

    // TODO: ajoutez vos opérations de service ici
}
```

Exemple de Contrat dans une classe

IService1

Exposition de contrats de données

// Utilisez un contrat de données comme indiqué dans l'exemple ci-après pour ajouter les types composites aux opérations de service.

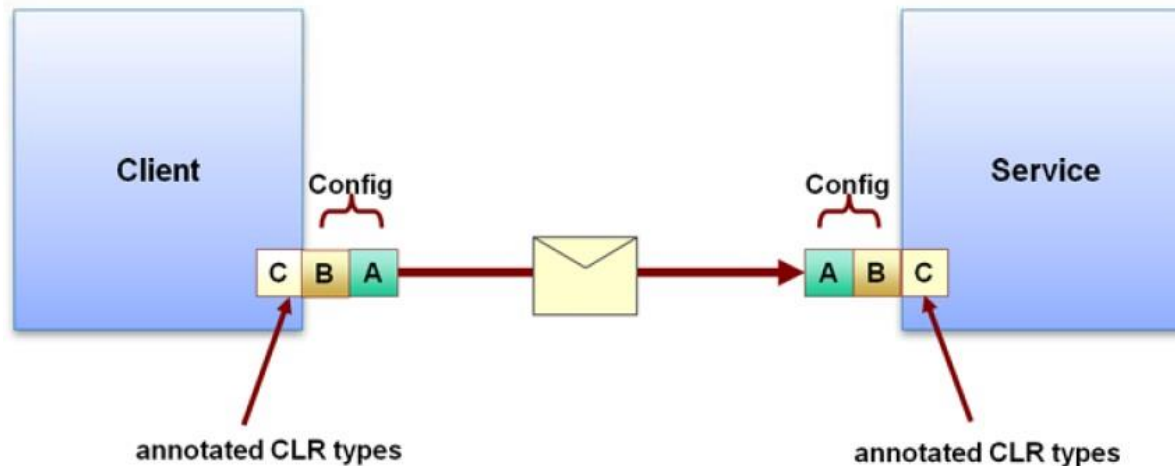
```
[DataContract]
public class CompositeType
{
    bool boolValue = true;
    string stringValue = "Hello ";

    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }

    [DataMember]
    public string StringValue
    {
        get { return stringValue; }
        set { stringValue = value; }
    }
}
```

Endpoints

- Les Applications communiquent à travers des **endpoints**
- Les Endpoints sont définis par le modèle WCF **ABC**
 - **Address** (where is the service)
 - **Binding** (which transport and encoding to use)
 - **Contract** (what operations are available)



Exemple de EndPoint et fichier de configuration App.config

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
```

```
  <system.serviceModel>
```

```
    <services>
```

```
      <service name="WcfServiceLibrary5.Service1">
```

```
        <endpoint
```

```
address="http://localhost:8733/Design_Time_Addresses/WcfServiceLibrary5/Service1"
```

```
binding="webHttpBinding"
```

```
contract="WcfServiceLibrary5.IService1"/>
```

```
      </service>
```

```
    </services>
```

```
  <behaviors>
```

```
    <endpointBehaviors>
```

```
      <behavior>
```

```
        <webHttp/>
```

```
      </behavior>
```

```
    </endpointBehaviors>
```

```
  </behaviors>
```

```
</system.serviceModel>
```

```
</configuration>
```

Namespace

Implémentation

URL de base

Binding choisi

Contrat non
exposé ici

- Deux modèles supportés
 - Self-hosting: le service a son propre processus (console, winform)
 - Windows Activation Service (WAS) est un service système d'activation des processus windows qui prend alors en charge l'hébergement du service en utilisant IIS.

Cf. : [Comment héberger un Service dans WAS](#)

Créer un **ServiceHost**, Ajouter des **Endpoints**, lancer le **ServiceHost (Open)**

```
ServiceHost host = new ServiceHost(typeof(Service));  
host.AddServiceEndpoint(typeof(IService),  
    new netTcpBinding(),  
    "net.tcp://localhost/service");  
  
host.open();  
  
console.WriteLine("Service ready ...");  
console.ReadLine();  
  
host.close();
```