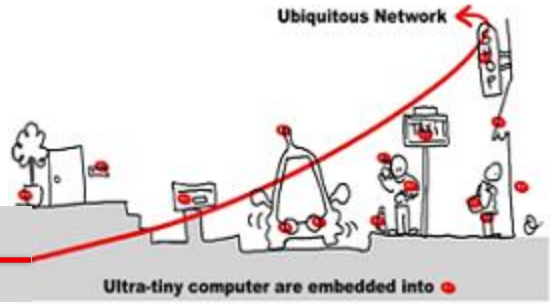
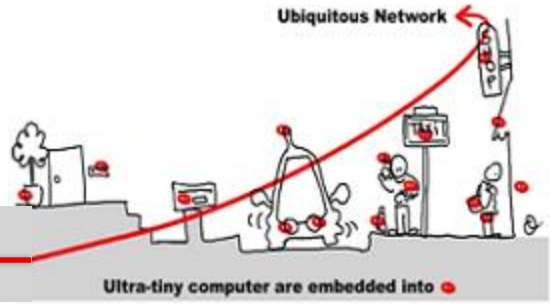


Services
Modèle ABC et WCF

Jean-Yves Tigli
http://www.tigli.fr
Polytech of Nice - Sophia Antipolis University

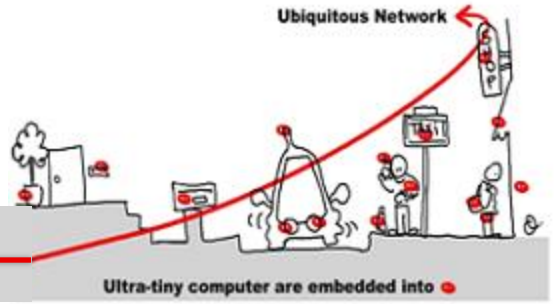
Email : tigli@polytech.unice.fr

Introduction à Windows Communication Foundation

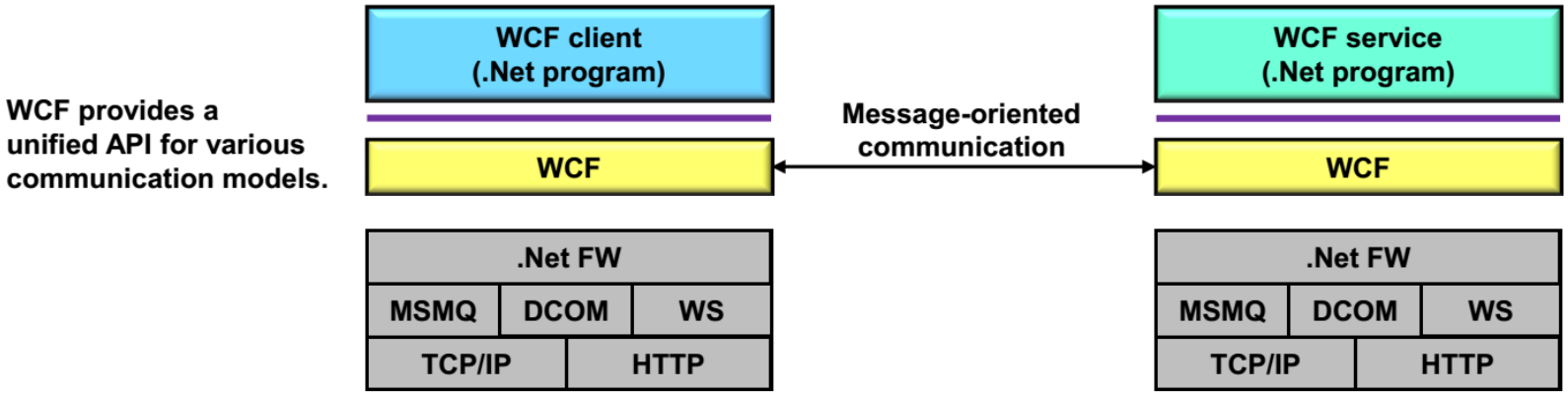


- WCF est un framework unifié pour les applications distribuées, avec son modèle de programmation et son API
- C'est le standard pour les applications distribuées Microsoft .Net
- Les caractéristiques de WCF sont :
  - Un modèle de programmation orienté service
  - Une interropérabilité avec des services non WCF grâce à l'inclusion des standards du domaine
  - Un 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

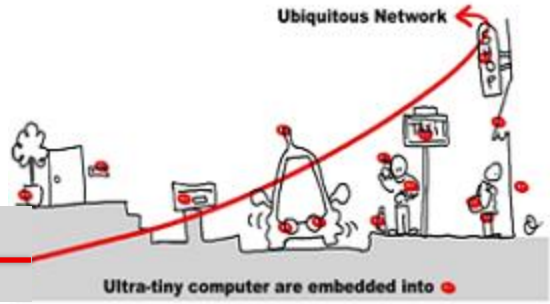


- WCF utilise de nombreux types de communication
- WCF est au sommet du framework .Net

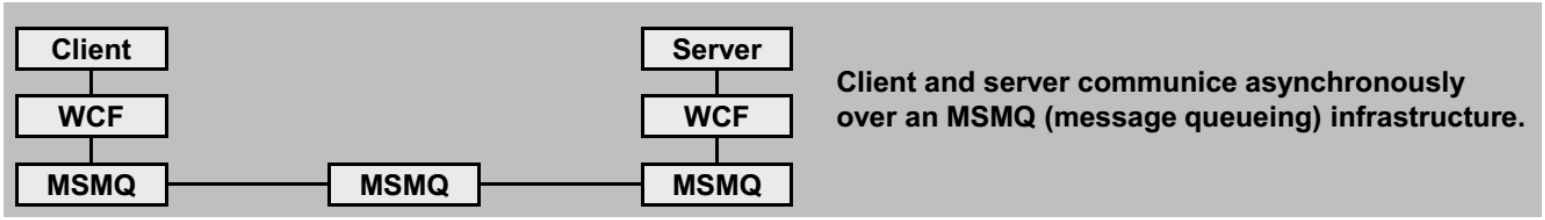


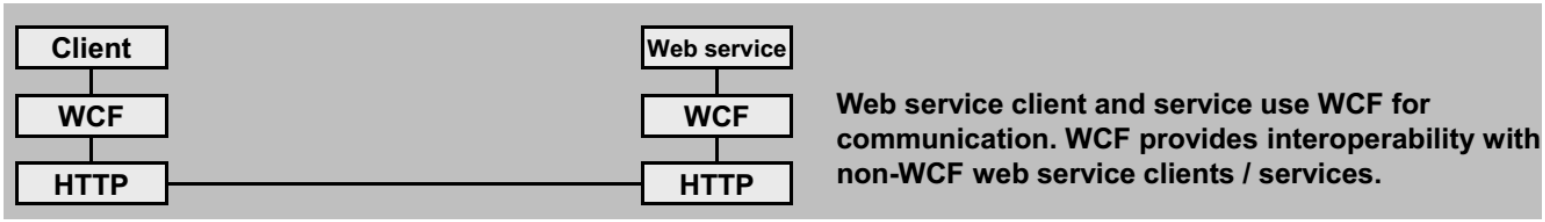
- En fait WCF est une approche orientée service qui va au-delà des Web Service
- En séparant les notions de contrat, de protocole et de format de message
- En séparant une description abstraite d'un service et son implémentation

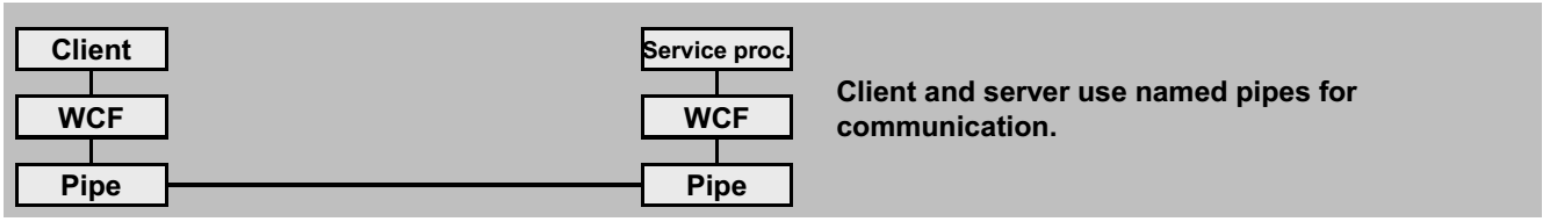
Introduction à Windows Communication Foundation



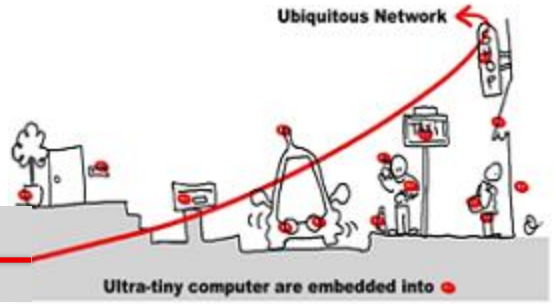
- WCF peut donc s'appuyer sur de multiples protocoles de communication







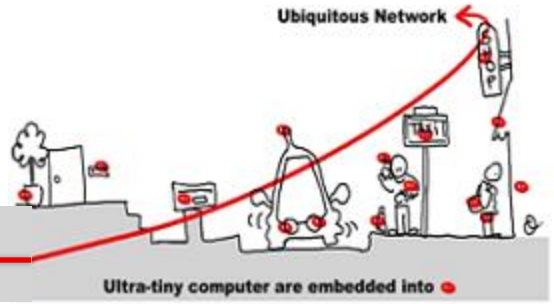
Modèle ABC de Service



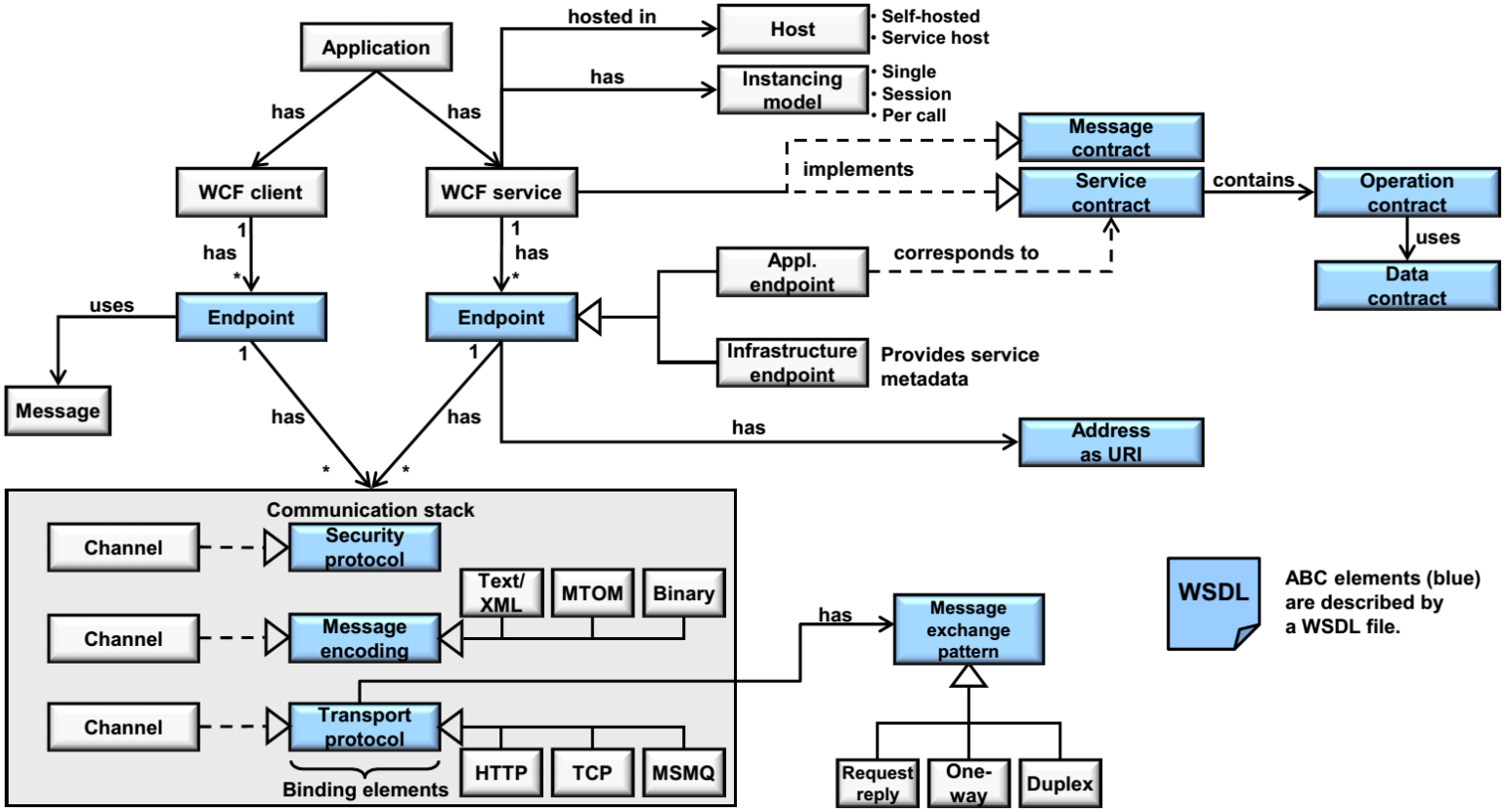
- Le concept central de WCF est la notion de service
- Qui est fourni par un « endpoint »
- Qui est accessible par une protocole de transport
- Ainsi un service WCF est défini à travers les informations ABC :
  - L'adresse A ou où le service est disponible (l'URI du enpoint dans le cas d'un WS)
  - Le binding B ou comment accéder au service (quel protocole de transport)
  - Le contrat C ou que fournit l'interface du service (opérations, types de données …)
- Exemple dans le cas de WSDL

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

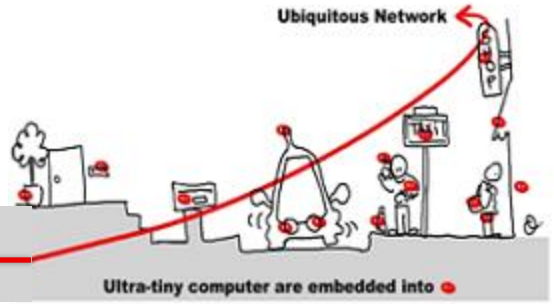
Modèle ABC de Service



- WCF fournit un modèle complet entités/relations

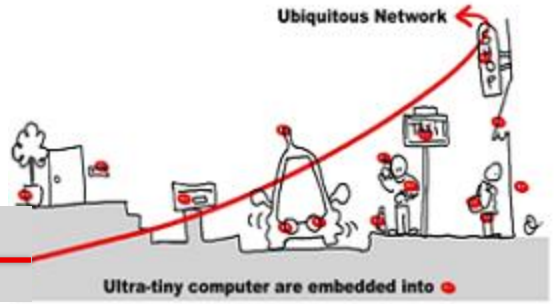


Modèle ABC de Service



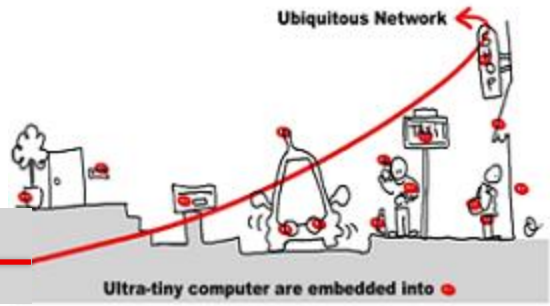
- Application: une application à un client WCF et/ou un service WCF
- Client WCF : Un client est un composant avec un endpoint pour communiquer avec un service WCF.
- Service WCF : S'exécute dans son propre process (self-hosted) ou sur un serveur spécifique (specific service hosting process, comme IIS, Windows Activation Service, Windows Service).
- Endpoint: Client et Service WCF client utilise un endpoint pour communiquer.
- Application endpoint: Le endpoint sur lequel un service applicatif est exposé/fourni.
- Infrastructure endpoint: Part of WCF system to offer metadata of an application service.
- Message: unité d'échange de données entre client et service WCF. WCF est exclusivement basé sur des messages.
- Adresse : Une adresse physique comprend un nom d'hôte, un numéro de port, et un nom de service (URI). Le service applicatif est accessible avec cette adresse
- Exemple.: http://localhost:8000/HSZ-TWSMW/DateTimeService

Modèle ABC de Service



- Le protocole de transport (transport binding): définit le protocole utilisé pour transférer les messages (ex. HTTP over TCP, ou TCP, …)
- Le format des message (message encoding binding element): définit le formatage des messages avant qu'ils soient envoyé sur le réseau grâce au protocole de transport (ex. Texte/XML, SOAP ou MTOM (Message Transmission Optimized Mechanism) pour le transfert efficace de données binaires).
- Le protocole de sécurité (security binding): définit les fonctions de sécurité qui sont appliquées aux messages (authentication, encryptage).
- Le pattern d'échange de message (vu sous le nopm de PortType) : définit comment les messages sont échangés (request-reply, one-way, duplex).
- La pile de communication : comprend différents éléments de binding, au minimum le transport binding et le message encoding binding element.
- L'hôte: l'environnement d'exécution pour le service sur un process indépendant (self-hosted) ou sur un serveur spécifique (comme IIS, Windows Activation Service, Windows Service).Either self-hosted (specific process for service)

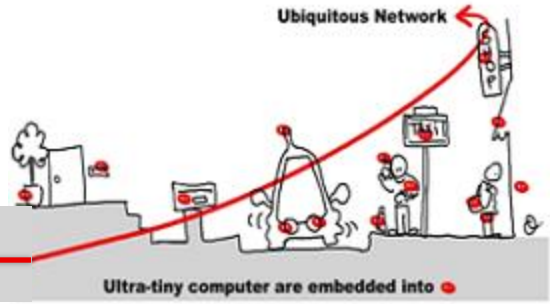
Le modèle de programmation WCF (Classes et Interfaces)



- Les éléments ABC de WCF sont disponibles comme des classes spécifiques

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

Le « A » de ABC



- L'adresse WCF définit où le service est disponible (endpoint reference, à l'instar du standard WS-Addressing)

WS-Addressing EPR schema:

```
<wsa:EndpointReference>
    <wsa:Address>xs:anyURI</wsa:Address>                                          URI
    <wsa:ReferenceProperties>... </wsa:ReferenceProperties> ?                     Properties to identify the endpoint
    <wsa:ReferenceParameters>... </wsa:ReferenceParameters> ?                     Parameters associated with an endpoint
    <wsa:PortType>xs:QName</wsa:PortType> ?                                       Endpoint type (WSDL 1.0 portType, WSDL 2.0 interface)
    <wsa:ServiceName PortName="xs:NCName"?>xs:QName</wsa:ServiceName> ?           Link to WSDL service element containing the endpoint descr.
    <wsp:Policy> ... </wsp:Policy>*                                               Security settings of endpoint
</wsa:EndpointReference>
```

WS-addressing EPR see http://www.w3.org/Submission/ws-addressing/#_Toc77464317

Example endpoint address URI:

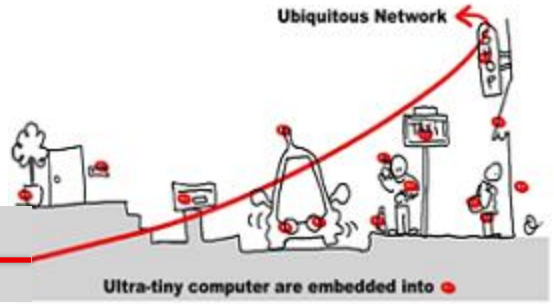
http://localhost:8000/HSZ-TWSMW/DateTimeService

- Scheme / protocol
- Machine address (IP or DNS)
- Port #
- Path under which the service is available

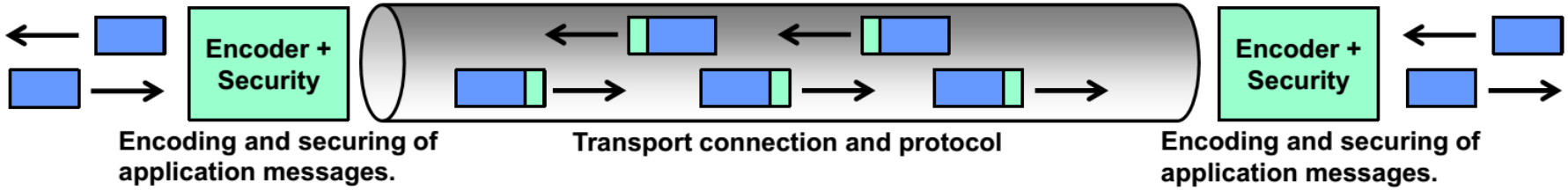
Endpoint address class `System.ServiceModel.EndpointAddress`:

- `EndpointAddress.Uri` → URI
- `EndpointAddress.Headers` → Reference properties and parameters
- `EndpointAddress.Identity` → Security settings

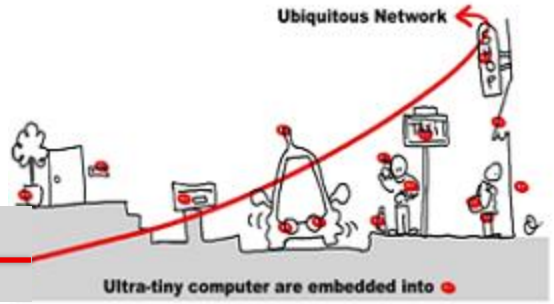
Le « B » de ABC



- 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 :



Le « B » de ABC



- WCF fournit les binding suivants :

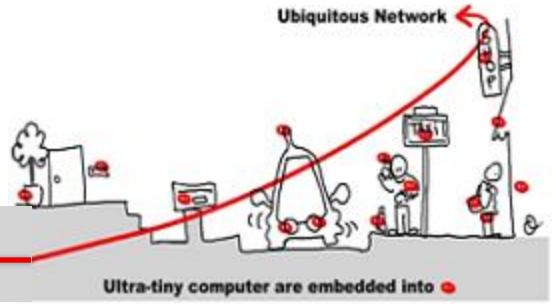
Binding	Interoperability	Security	Session	Transactions	Duplex	Encoding
BasicHttpBinding	WS-I Basic Profile	N, T, M, m	N	N	No	Text, MTOM
WSHttpBinding	WS-* standards	T, M, m	N, RS, SS	N, Yes	No	Text, MTOM
WSDualHttpBinding	WS-* standards	M, m	RS, SS	N, Yes	Yes	Text, MTOM
WSFederationHttpBinding	WS-Federation	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:

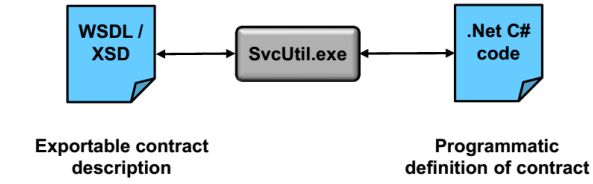
N:	None	RS:	Reliable Session (WS-ReliableMessaging)	POX: Plain Old XML
T:	Transport	SS:	Security Session	
M:	Message	TS:	Transport Session	
m:	mixed			

Le « C » de ABC

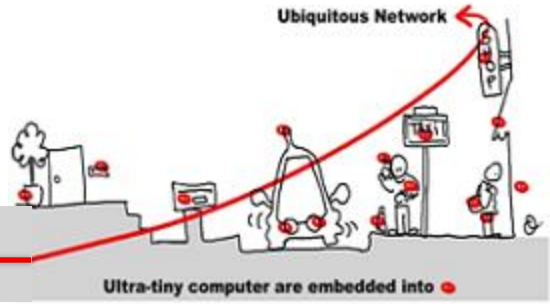


- Les interfaces WCF sont appelées des contrats
- Les contrats définissent des opérations, des structures de données et des messages
- Un contrat de service définit :
  - a. un groupement d'opérations dans un service (le .Net attribute [ServiceContract])
  - b. les signatures des operations (le .Net attribute [OperationContract])
  - c. Les types de données des opérations (le .Net attribute [DataContract])
- A partir d'un contrat vers du code ou vice-versa :

L' utilitaire SvcUtil.exe peut être utilisé pour les correspondances entre services abstraits, services interopérables (documents WSDL) et la définition des interfaces (code C #).
Soit SvcUtil.exe récupère la définition du service à partir d'un fichier WSDL sur le disque à l'aide du protocole DISCO (protocole propriétaire de Microsoft pour la découverte de service) ou directement à partir d'un service en cours d'exécution via par exemple un protocole de norme WS-MetadataExchange)

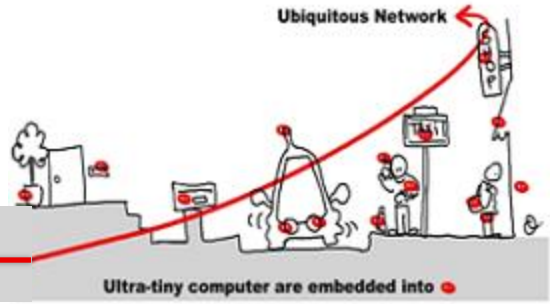


Hébergement d'un service WCF



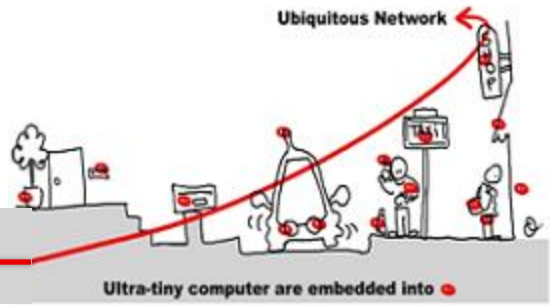
- Un service est hébergé par un processus qui exécute des services.
- WCF fournit plusieurs possibilités pour héberger des services :
  - A. Self-hosted service:
    - Le cycle de vie est géré par le service lui-même
  - B. Un processus d'hébergement standard Windows : .Net et Windows offre des processus spécialisés dans l'hébergement de services.
    - a. IIS (Internet Information Services):
      - Supporte seulement HTTP (IIS est un serveur web).
    - b. WAS (Windows Activation Services):
      - Nouveau framework sur lequel IIS 7.0 s'appuie
      - Supporte HTTP, TCP, les pipes.
    - C. Windows services:
      - Le cycle de vie est controllé par la configuration de démarrage de services.
  - Plus d'informations sont disponibles sur http://msdn.microsoft.com/enus/library/ms730158.aspx.

Les étapes pour créer un service WCF



- Selon le mode de déploiement, il faut définir des projets différents :
- Créer un projet Visual C# / WCF / WCF Service Library :
- Le fichier binaire produit sera une DLL (Dynamic Link Library), pour s'exécuter dans un WAS.
- Les fichiers de code source sont :
  - IService1.cs avec les attributs [ServiceContract] (service interface = contrat).
  - Service1.cs (implémentation du service).
  - App.config (fichier de configuration du transport binding et autres paramètres).
- Créer un projet Visual C# / WCF / WCF Service Application :
- Le fichier binaire produit sera une DLL (Dynamic Link Library), pour s'exécuter dans IIS.
- Les fichiers de code source sont :
  - IService1.cs avec les attributs [ServiceContract] (service interface = contrat).
  - Service1.cs (implémentation du service).
  - Web.config (fichier de configuration sur du HTTP binding et autres paramètres).

Les étapes pour créer un service WCF



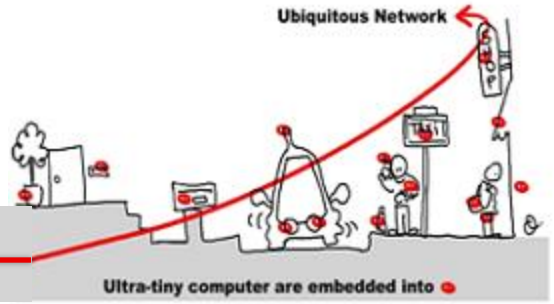
- Définition de l'interface (contrat de service comme une interface C# ou une classe C#):
  - Définit une interface d'un web service avec une interface .Net
  - Exemple un contrat défini en C# (contenant des contract sur des operations):

```
[ServiceContract(Namespace = "http://indigoo.WSMW")]
public interface IMyInterface
{
  [OperationContract]
  string MyFunction();
  [OperationContract]
  int MyOtherFunction();
}
```

- Implémentation d'un contrat de service (classe C# implémentant l'interface):
  - Créer une classe C#

```
public class MyService : IMyInterface
{
  public string MyFunction() {...}
  ...
}
```

Les étapes pour créer un service WCF



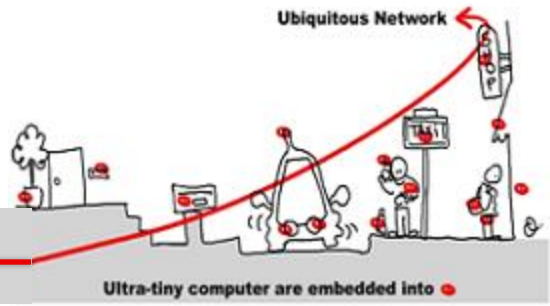
- Déploiement et exécution du service :
- Le déploiement se fait dans une DLL ou directement en lançant un nouveau processus (self-hosting service)
- Dans ce dernier cas, le code suivant doit être ajouté au service

```
selfHost.AddServiceEndpoint(typeof(MyService), new BasicHttpBinding(), "MyService");
```

- Enfin, le service peut-être lancé avec la méthode :

```
selfHost.Open();
```

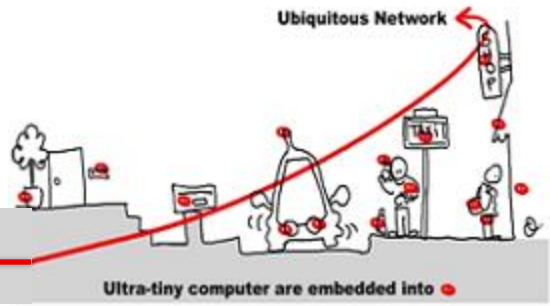
WCF, configurer plutôt que programmer



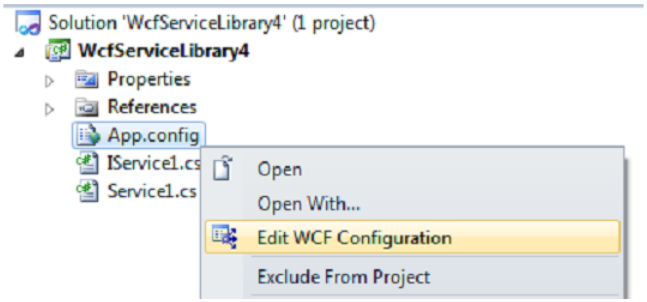
- Un Client et un service WCF peuvent être définis dans une configuration (déclarative en XML or programmatique avec des classes .Net).
- Bonnes pratiques :
  - Pour améliorer la ruitilisabiité des services web dans des environnements différents, les contrats (C), les adresses (A) et le binding ('B') doivent être séparés.
  - Définir l'interface et son implémentation par programmation sans mentionner de protocole de transport et de format de message
  - La définition du protocole de transport, de l'adresse, et du format de message sera faite dans un fichier de configuration
  - Le fichiers de configuration sont App.config ou Web.config comme :

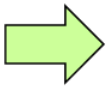
```
<system.ServiceModel>
  <services>
    <service>
      <endpoint/>
    </service>
  </services>
  <bindings>
    <!-- Specify one or more of the system-provided binding elements, for example, <basicHttpBinding> -->
    <!-- Alternatively, <customBinding> elements. -->
    <binding> <!-- For example, a <BasicHttpBinding> element. --></binding>
  </bindings>
  <behaviors> <!-- One or more of the system-provided or custom behavior elements. -->
    <behavior> <!-- For example, a <throttling> element. --> </behavior>
  </behaviors>
</system.ServiceModel>
```

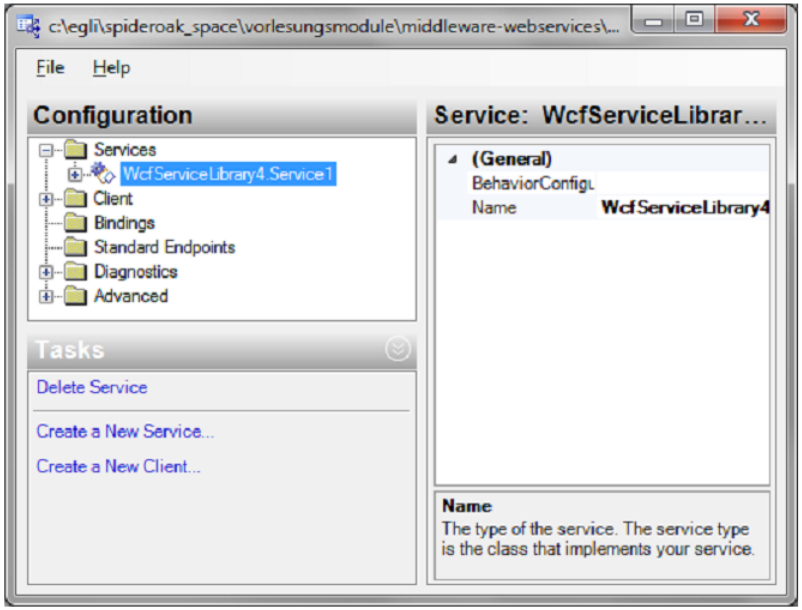
WCF, configurer plutôt que programmer

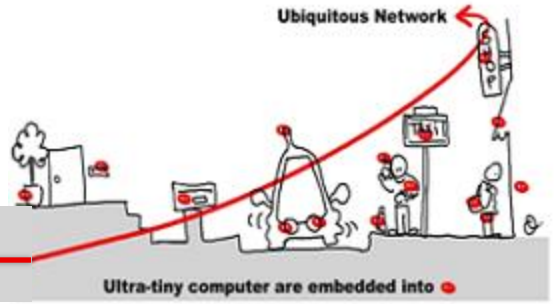


- La configuration du service peut se faire graphiquement avec SvcConfigEditor.exe: Microsoft\SDKs\Windows\v7.0A\Bin\SvcConfigEditor.exe.





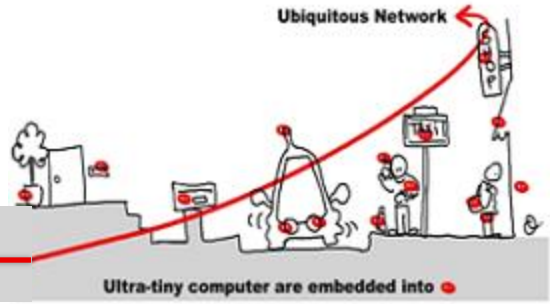




Synthèse du cours

Ce que nous n'aurons pas vu

Pile Protocolaire des Services Logiciels issus du Web

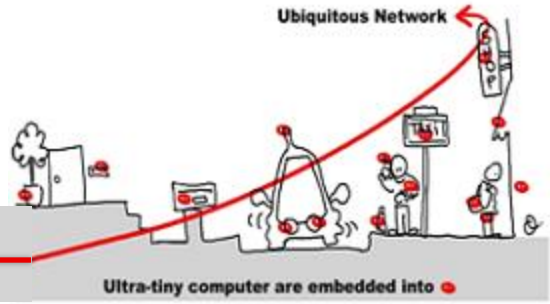


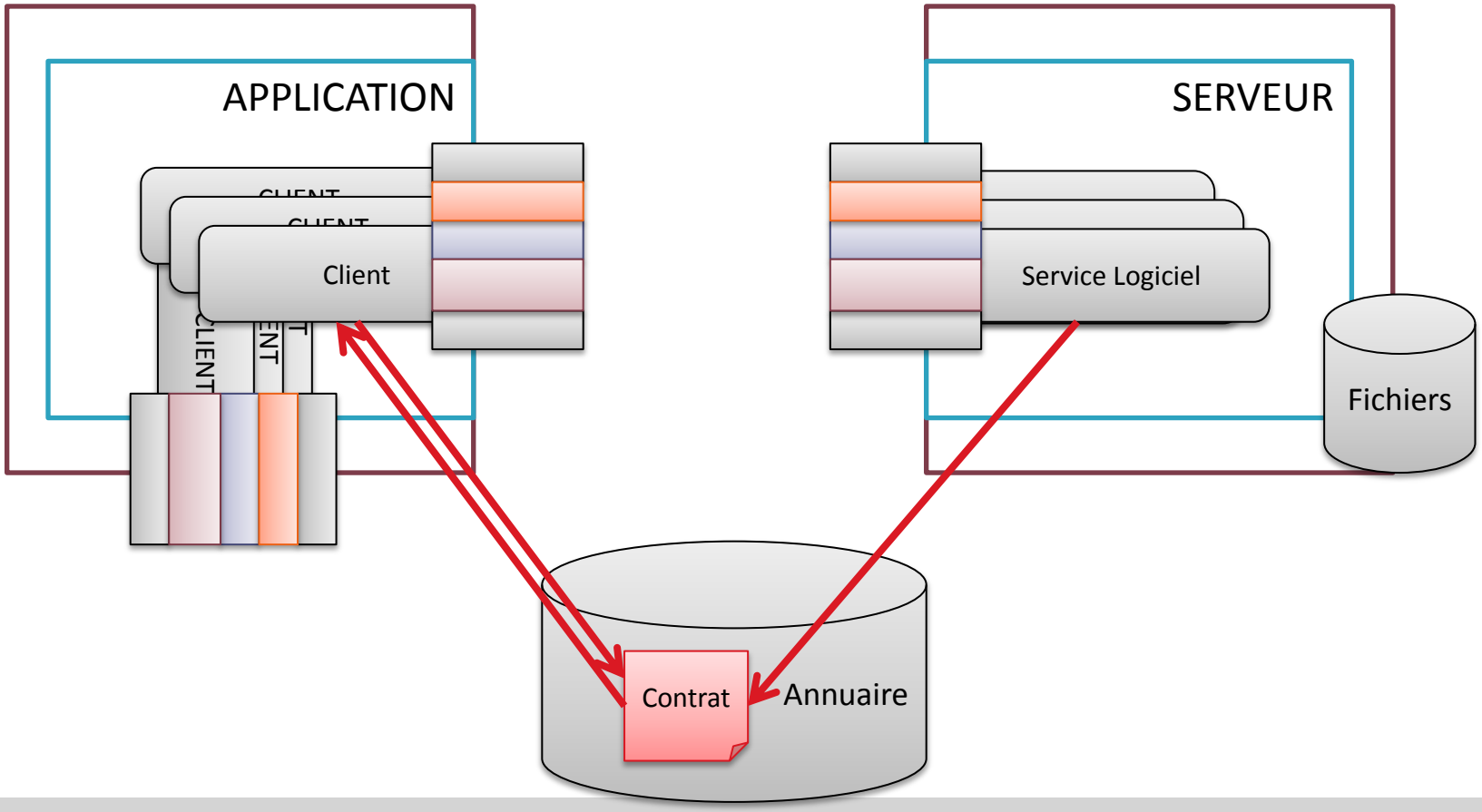
Les principaux composants ou couches d'une pile de protocoles de services Web incluent :

- **Couche Transport**— assure la transmission des messages entre les applications
- **Couche Protocole et Format D'échanges** — encode et gére la séquence des messages échangés entre le service et son consommateur
- **Couche Description de Service et Contrat** — décrit le service fourni
- **Couche Annuaire et Recherche de Services**— centralise les services au moyen d'un registre commun

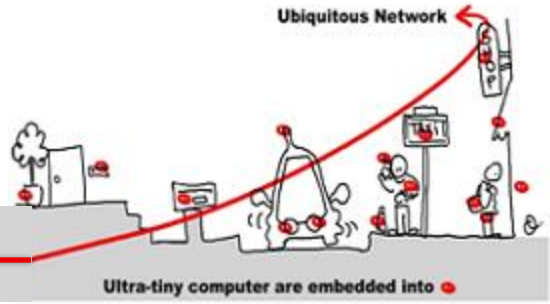
Couche 5 Applicative *ORCHESTRATION MASHUP COMPOSITION*
Couche 4 : Annuaire et Recherche de services *UDDI*
Couche 3 Description de Service et Contrat : WSDL WADL…
Couche 2 : Protocole et Format D'échanges : PORTTYPE, XML, SOAP, JSON, Binary ..
Couche 1 Transport HTTP, SMTP, FTP, TCP/IP

Architecture SOA / ROA



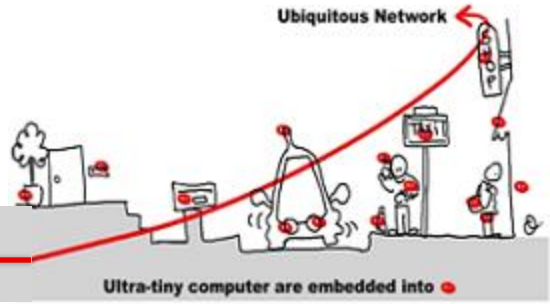


Ce que nous n'aurons pas vu



- Les annuaires (ex. UDDI)
- Les annotations sémantiques de services (ex. OWL-S: Semantic Markup for Web Services)
  - OWL-S est une ontologie pour décrire la sémantique de Services Web
  - See OWL Web Ontology Language Use Cases and Requirements
- Composition de services : orchestration et choregraphie (ex. WS-BPEL)
- Etc.

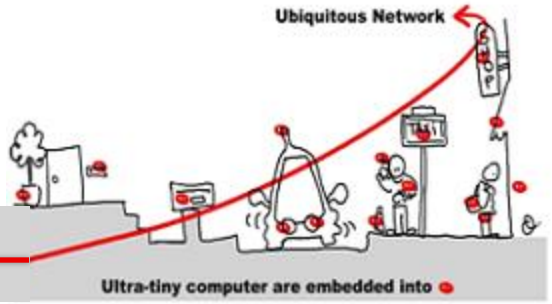
Annuaire de Services : ex. UDDI



- UDDI offre plusieurs services, en voici quelques exemples :
  - De type informationnels (nom d'entreprises, descriptions de l'entreprise, etc)
  - Enregistrement d'entreprise
  - Liste de services disponibles par entreprise
  - Informations techniques (relations bindings) sur chacun des services (ex: le WSDL)
- Voici à quoi ressemblerais une requête pour le W3Québec si celui-ci disposait de web services enregistrés sur un répertoire de services.

```
<?xml version="1.0" encoding="UTF-8"?>
        <Envelope  mlns="http://schemas.xmlsoap.org/soap/envelope/">
                <Body>
                        <find_service businessKey="*" generic="1.0" xmlns="urn:uddi-org:api" maxRows="5">
                                <name>W3Québec</name>
                        </find_service>
                </body>
        </Envelope>
```

Annuaire de Services : ex. UDDI



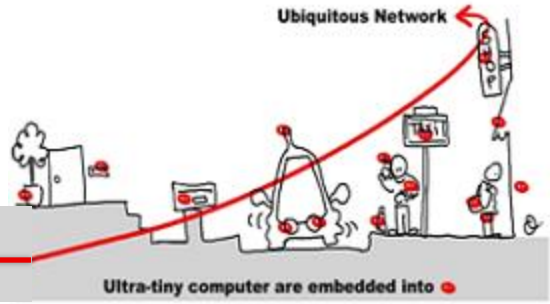
Et la réponse :

```
<businessList generic="1.0" operator="W3Québec" truncated="false" xmlns="urn:uddi-org:api">
    <businessInfos> <businessInfo
                businessKey="389457230985023948572304985 7">    <name>W3Québec</name>
                <description xml:lang="fr"> L
```

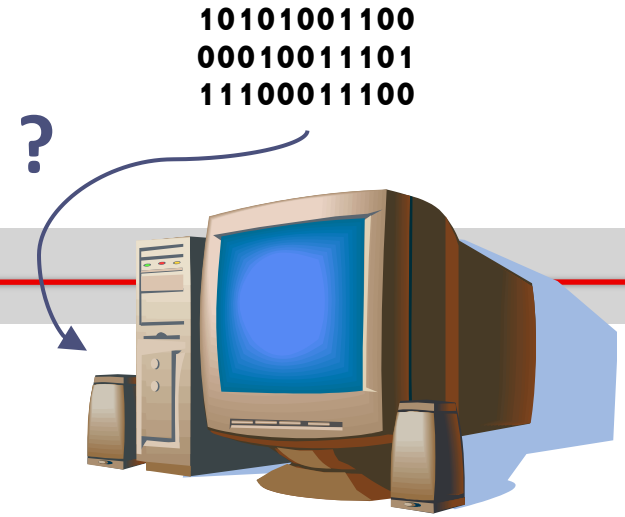
Le W3Québec est né d'une volonté de rehausser la qualité du Web et du multimédia au Québec pour qu'ils deviennent des outils de communication accessibles à tous. À cet effet, le W3Québec entend mettre en lumière les enjeux stratégiques, technologiques, économiques et sociopolitiques liés à leur utilisation et faire connaître la valeur ajoutée des normes, standards et bonnes pratiques à tous les décideurs, acteurs et professionnels du milieu.

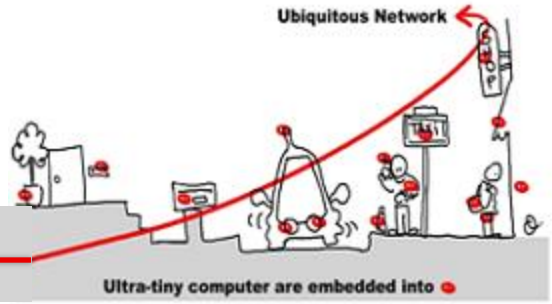
```
                </description>
                        <serviceInfos>
                                <serviceInfo
                                businessKey="847197401293874012937401293478029 8347"
                                serviceKey="457203495723049857023497580234985 7230">
                                <name>Exemple de webservice "widget service«
                                 </serviceInfo>
                        </serviceInfos>
        </businessInfo>
    </businessInfos>
```

Exemple WS-BPEL - activity




```
…
<switch>
    <case condition ="condition1">
        <invoke partnerLink ="Customer"...> </invoke>
    </case>
    <otherwise>
        <flow>
            <invoke partnerLink ="Customer"...> </invoke>
            <switch>
                <case condition="condition2">
                    <invoke partnerLink ="HotelService"...>
                    </invoke>
                </case>
                <otherwise>
                    <invoke partnerLink ="FlightService"...>
                    </invoke>
            </switch>
…
</switch>
```





LE TD D'AUJOURD'HUI SUR REST/JSON

- Un exemple sous .NET C#