

Microsoft
.net

Service réseaux

COURS N°6 : WEB SERVICES

Antony BRUGERE
a.brugere@gmail.com

Frédéric CHASSAGNE
frederic.chassagne@capgemini.com

Plan du cours

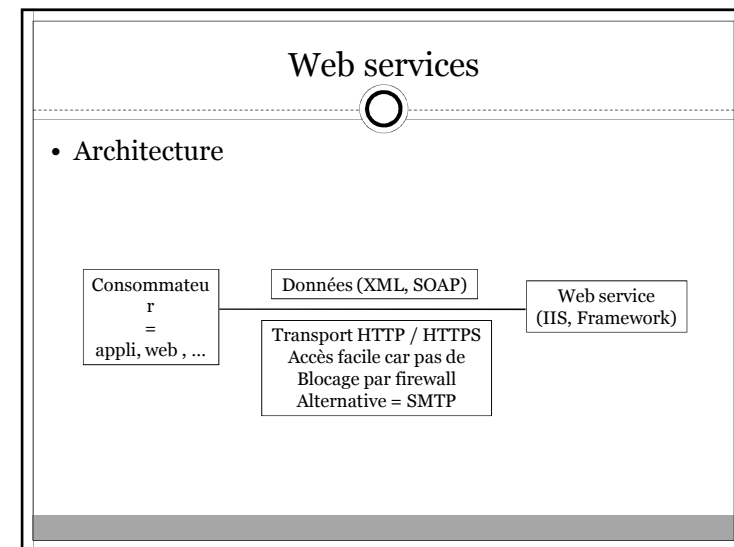
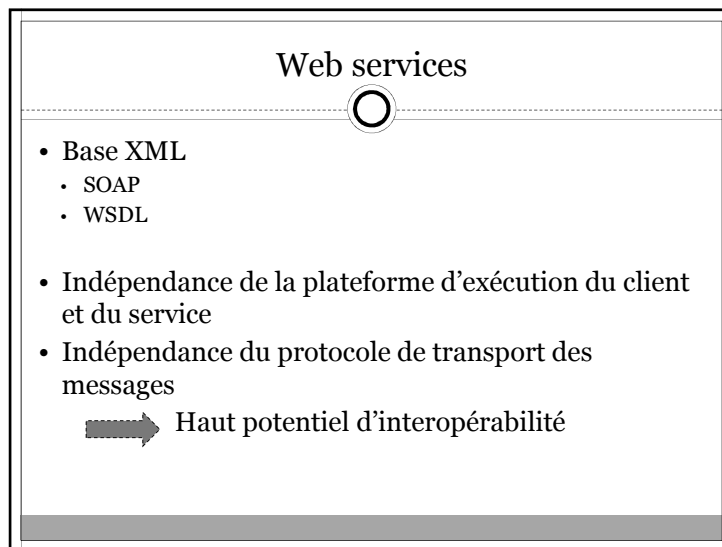
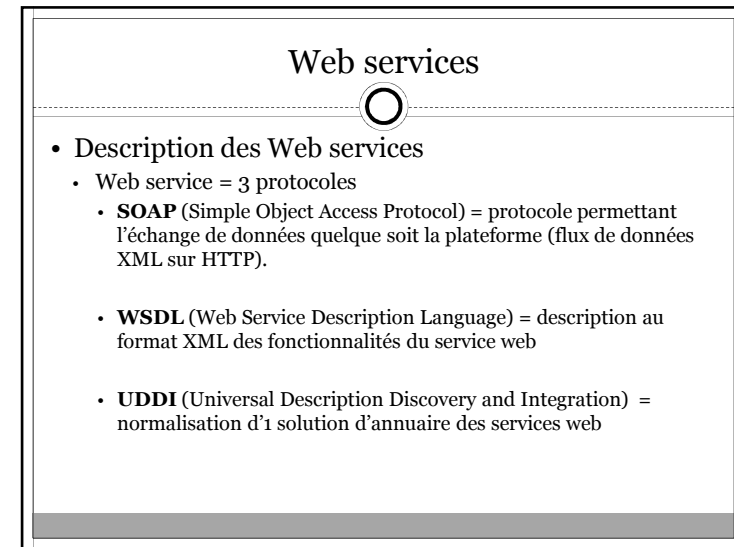
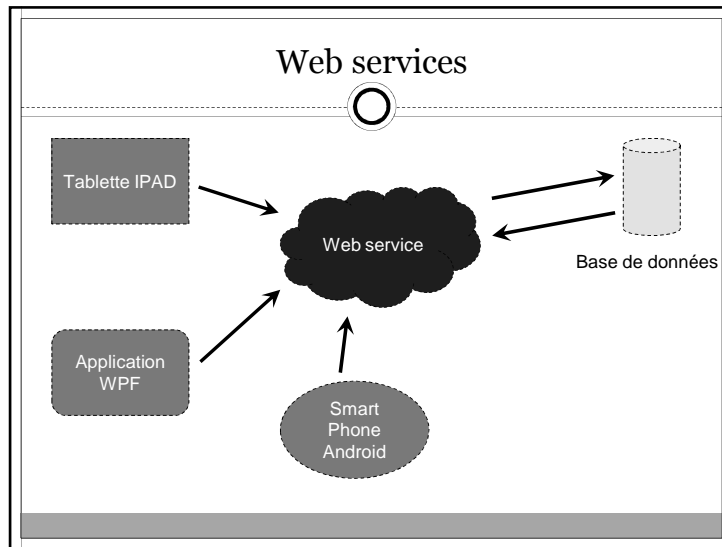
- Web services
 - Présentation
 - Architecture
 - Web service en .net

Web services

- Web service = Application avec laquelle on communique par l'échange de message
- Architecture
 - Service Oriented Architecture
 - Frontières explicites
 - Coût des transactions
 - Entité autonome
 - Déploiement et évolution d'un service indépendant des clients
 - Structure d'utilisation définie sans ambiguïté
 - Sémantique d'utilisation définie sans ambiguïté

Web services

- Service web
 - Composant développé dans n'importe quel langage
 - Composant développé sur n'importe quelle plateforme
 - Composant enveloppé dans une couche XML
- Obligation
 - Pouvoir être invoqué par n'importe quel autre service

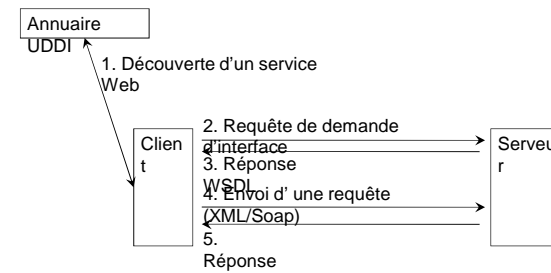


Web services

- Deux grandes façon d'aborder les Web services
 - Modèle de message requête / réponse
 - Mode de transport associé = HTTP
 - Modèle de message asynchrones, traitement par file d'attente
 - Mode de transport associé = SMTP
- Dans les 2 cas, peu de soucis de Firewall !

Web services

• Cycle de vie



Web services

- SOAP
 - Langage XML utilisé pour la rédaction des messages échangés entre services
 - Fournit un mode d'empaquetage standard des messages
 - Standard de l'industrie
 - Popularisé par des grands : Microsoft, IBM, Sun ...

Web services - SOAP

- Avantages de SOAP
 - Non étroitement lié à un langage
 - Non couplé à un protocole de transport particulier
 - Lié à aucune infrastructure d'objets distribués
 - Utilise les standards existants de l'industrie
 - Permet l'interopérabilité entre plusieurs environnements

Web services - SOAP

• Anatomie d'un message SOAP

- Message SOAP = 1 enveloppe
 - Définissant type d'encodage et namespaces
 - Contenant
 - Un entête utilisé pour décrire le message
 - Un corps contenant le message
- Exemple

```
<soap:Envelope xmlns:soap=http://www.w3.org/2001/12/soap-
envelope
soap:encodingStyle=http://www.w3.org/2001/12/soap-encoding>
</soap:Envelope>
```

```
<soap:Header>
...
</soap:Header>
<soap:Body>
...
<soap:Fault>
...
</soap:Fault>
</soap:Body>
```

Web services - SOAP

• Élément Header

- Facultatif
- Destiné à passer des données qui ne sont pas destinées à être encodées dans le corps du message
 - Exemple : Si corps du message = compressé, header contient le type d'algo de compression utilisé
- Peut servir pour
 - Authentification, relevé d'informations sur la sécurité, informations sur le routage, transactions, informations sur le paiement, ...

Web services - SOAP

• Élément Body

- Obligatoire
- Contient la charge utile du message
- Pas de restriction sur le type d'encodage
 - Chaîne de caractères, tableaux d'octets, code XML
- 1 seule contrainte
 - Caractères invalidant un document XML interdits

Web services - SOAP

• Élément Body

- 2 orientation possibles
 - Procédure
 - Communication bidirectionnelle (RPC)
 - Définit action requise et ensemble paramètres d'entrée/sortie
 - Document
 - Communication unidirectionnelle
 - Ex : Bon de commande

Web services - SOAP

• Élément Fault

- Définit un standard de communication de message d'erreur
- Contient 4 éléments
 - Faultcode (obligatoire) = Code d'erreur SOAP
 - Faultstring (obligatoire) = Explication lisible pour un humain
 - Faultactor (optionnel) = Source exacte de l'erreur
 - Détail = Détail de l'erreur non lié au Body du message

Web services - SOAP

• Élément Fault

- Principaux codes erreur

Code d'erreur	Description
VersionMismatch	Namespace invalide spécifié dans l'enveloppe
MustUnderstand	Attribut du header non compris
Client	Contenu du message = cause 1 ^{re} de l'erreur
Server	Contenu du message ≠ cause 1 ^{re} de l'erreur

Web services - SOAP

• Élément Fault

- Exemple

```

...
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Client</faultcode>
    <faultstring> The ISBN value contains invalid characters
  </faultstring>
    <faultactor>http://www.xyzcorp.com</faultactor>
    <detail> <mh:InvalidIsbnFaultDetail>
      <offending-value>19318224-D</offending-value>
      <conformance-rules>
        The first nine characters must be digits.
      </conformance-rules>
    </mh:InvalidIsbnFaultDetail>
  </detail>
</soap:Fault>
</soap:Body>

```

Web services - WSDL

• WSDL

- Langage XML de description des services web
 - Description normalisée qui permet une utilisation inter-plateforme
 - Masquage du code du web service
- Regroupe :
 - Les méthodes du service web
 - Les paramètres et les valeurs de retour
 - Le protocole de transfert (SOAP ou autre)
 - La localisation (url du service)

Web services - WSDL

- **WSDL**
 - Une description WSDL = balise définition qui contient 5 types d'éléments enfants :
 - Types = Définition de schéma des messages qui peuvent être envoyé ou reçu.
 - Message = Référence croisée associant message à sa définition dans le schéma
 - PortType = Jeu d'interfaces proposée par le service
 - Binding = Associe portType à un protocole particulier
 - Service = Ensemble des points d'entrée du service

Web services - WSDL

- **WSDL**
 - Utilisation d'un web service = Description WSDL de ensemble services mis à disposition
- **Problème** = nécessaire de connaître l'url WSDL
- **Besoin** = automatisation de la découverte
- **Solution**
 - UDDI
 - DISCO

Web services - UDDI

- **UDDI (Universal Description, Discovery and Integration)**
 - standard (microsoft, IBM, sun, Oracle, HP...)
 - Annuaire d'informations administratives et techniques sur les services
 - Enregistrement via un opérateur (microsoft ou IBM)
 - Plusieurs types de recherche (identification d'entreprises, catégories,...) et recherche de services

Web services

- **DISCO (The Web Service Discovery Tool)**
 - Création de Microsoft
 - Mécanisme décentralisé
 - Permet l'interrogation des services web proposés par un serveur
 - Implémente le paradigme de navigation

Web services -Récapitulatif



Web services



- Les raisons du succès
 - Utilisation d'HTTP
 - Protocole Internet
 - Serveur web présent dans la majorité des entreprises
 - Protocole généralement autorisé au niveau de parefeu
 - Protocole disponible sur toutes les plateformes
 - Utilisation d'XML
 - Massivement utilisé et reconnu
 - Permet de structurer l'information facilement

Web services



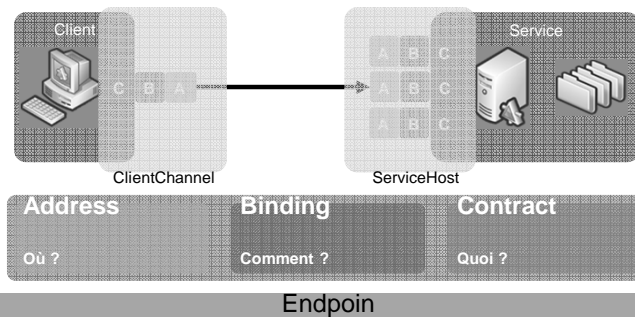
- Implémentations existantes
 - Microsoft .Net
 - Sun JavaONE : J2EE + Web services (WSDP = JAXP, JAXRPC, JAXM...)
 - Apache SOAP / Axis (JAXRPC)
 - IBM WSTK
 - Oracle, Bea, Iona, Enhydra ...

Web services en .NET – 3.5



- WCF = Unification de l'ensemble des api de communication des Framework antérieurs

Web services en .NET – 3.5

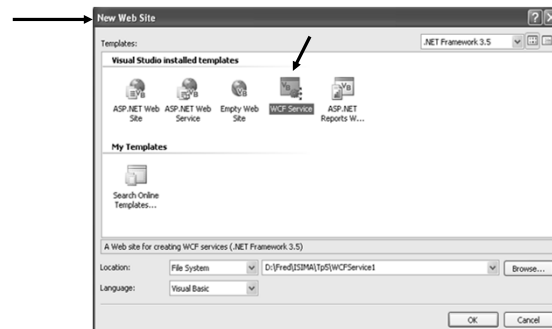


Web services en .NET – 3.5

- Modèle basique de programmation WCF
 - [ServiceContract]/[OperationContract] : définit le contrat présenté par le web service (ie ensemble des points d'entrée)
 - [DataContract] : définit les types composite utilisés dans les méthodes présentées par le web service

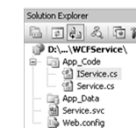
Web services en .NET – 3.5

- Dans Visual Studio



Web services en .NET – 3.5

- Structure du projet



- Par défaut, 2 méthodes exemples décrites et implémentées

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

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);
}
```


Web services en .NET – 3.5

- Les objets non standards retournés doivent être définis expressément

```
[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; }
    }
}
```

Web services en .NET – 3.5

- La classe Service implémente les points d'entrée

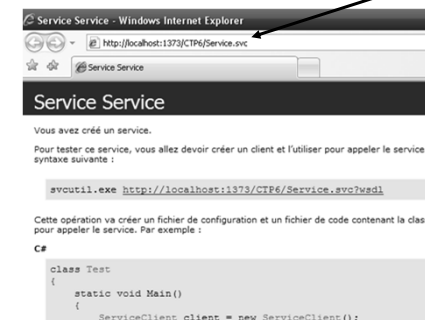
```
public class Service : IService
{
    public string GetData(int value)
    {
        return string.Format("You entered: {0}", value);
    }

    public CompositeType GetDataUsingDataContract(CompositeType composite)
    {
        if (composite.BoolValue)
        {
            composite.StringValue += "Suffix";
        }
        return composite;
    }
}
```

Web services en .NET – 3.5

- En résumé, pour développer un web service
 - Définir les objets mis à disposition par le Web Service
 - Décrire les points d'entrée dans l'interface IService
 - Implémenter les points d'entrée dans la classe Service

Web services en .NET – 3.5



```
svcuttl.exe http://localhost:1373/CTP6/Service.svc?wsdl
```

```
Cette opération va créer un fichier de configuration et un fichier de code contenant la classe de
pour appeler le service. Par exemple :
```

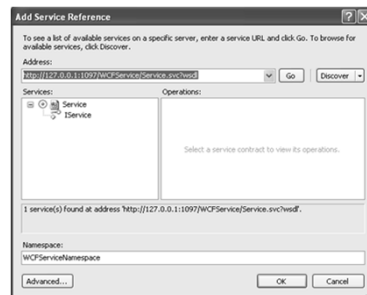
```
cs
class Test
{
    static void Main()
    {
        ServiceClient client = new ServiceClient();
    }
}
```

- Service généré
- Extension : svc
- Impossible à tester directement

Pour tester :
Ecriture de tests unitaires

Web services en .NET – 3.5

- Référencer le Web Service = « Ajouter une référence de service »



Web services en .NET – 3.5

- Appeler le Web Service depuis 1 couche qui le référence = l'Instancier

```
[TestMethod]
public void GetDataTest()
{
    WCFServiceNamespace.ServiceClient service = new WCF.Tests.WCFServiceNamespace.ServiceClient();
    string result = service.GetData(5);
    Assert.AreEqual("You entered: 5", result);
}
```

Web services en .NET – 3.5

- Exemple :
 - Encapsulation d'un service ASMX de conversion de Devises
- But :
 - Masquer l'appel à un web service ASMX externe par l'intermédiaire d'un frontal WCF
 - Découplage le service proposé du fournisseur de l'implémentation

Web services en .NET – 3.5

```
[ServiceContract] ←
public interface IService
{
    [OperationContract] ←
    double CurrencyConvert(double amount, SupportedCurrency from,
        SupportedCurrency to);
}

[DataContract] ←
public enum SupportedCurrency
{
    [EnumMember()] ←
    Euro = 0,
    [EnumMember()]
    LivreSterling = 1,
    [EnumMember()]
    Dollar = 2
}
```

Web services en .NET – 3.5

```
public class Service : IService
{
    public double CurrencyConvert(double amount, SupportedCurrency from,
        SupportedCurrency to)
    {
        net.websvcex.www.CurrencyConvertor cc = new
        net.websvcex.www.CurrencyConvertor();
        cc.UseDefaultCredentials = true;
        cc.Proxy = new System.Net.WebProxy();
        cc.Proxy.Credentials = cc.Credentials;

        net.websvcex.www.Currency wwwFrom = GetCurrency(from);
        net.websvcex.www.Currency wwwTo = GetCurrency(to);

        double conversionRate = cc.ConversionRate(wwwFrom, wwwTo);
        return amount * conversionRate;
    }
}
```

Ajout de référence Web
Vers le web service externe
[http://www.websvcex.net/
CurrencyConvertor.asmx?
WSDL](http://www.websvcex.net/CurrencyConvertor.asmx?WSDL)