Cours n°6 : Web services

SERVICES RÉSEAUX

Hélène CHASSAGNE
helene.chassagne@orange.fr

Frédéric CHASSAGNE
frederic.chassagne@atosorigin.com

Plan du cours

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

Web services

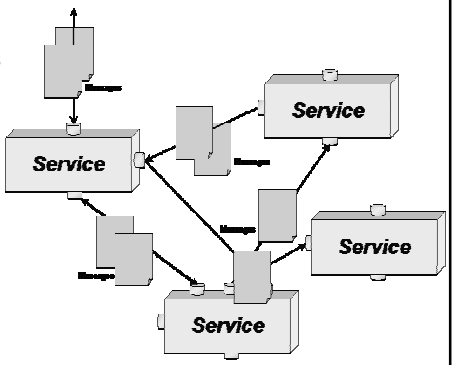
- Web service = Application avec laquelle on communique par l'échange de message
- Architecture
  - Service Oriented Architecture
    - Frontières explicites
      - Coût des transactions
    - Entité autonome
      - Déploiement et évolution d'un service indépendant des clients
    - Structure d'utilisation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

- Points forts
  - Interaction par messages
  - Découplage maximum
  - Souplesse d'évolutivité
  - Disponibilité indépendante



Web services

- Description des Web services
  - Web service = 3 protocoles
    - WSDL (Web Service Description Language) = description au format XML des fonctionnalités du service web
    - SOAP (Simple Object Access Protocol) = protocole permettant l'échange de données quelque soit la plateforme (flux de données XML sur HTTP).
    - UDDI (Universal Description Discovery and Integration) = normalisation d'1 solution d'annuaire des services web

Web services

- Base XML
  - SOAP
  - WSDL
- Indépendance de la plateforme d'exécution du client et du service
- Indépendance du protocole de transport des messages

⇨ Haut potentiel d'interopérabilité

Web services

- Architecture

Consommateur = appli, web , ...

Données (XML, SOAP)

Transport HTTP / HTTPS
Accès facile car pas de Blocage par firewall
Alternative = SMTP

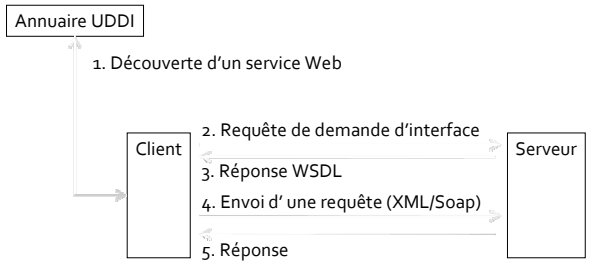
Web service (IIS, Framework)

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 - 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
  - 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
  - Permet la description des points d'entrée à un web service
  - Peut être interprété par n'importe quel langage
  - Présente comment dialoguer avec le service
- WSDL = Description de ce qui est possible
- Une fois défini, il reste à dialoguer avec le WS

Web services - SOAP

- 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

- Elément 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

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

Web services - SOAP

- Elément 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

- Elément 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

- Elément 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 1e de l'erreur
Server	Contenu du message ≠ cause 1e de l'erreur

Web services - SOAP

- Elément 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

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

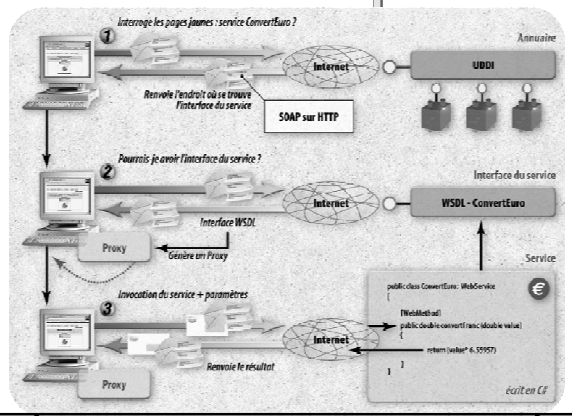
Web services

- 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
  - 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

- 2 approches des Web services en .net
  - Web services Asp.net avec le Framework 2.0
  - WCF avec le Framework 3.5

Web services en .NET – 2.0

- Présentation
  - Web services = sorte de Pages ASP.net
  - Extension .asmx
  - Hebergé par un IIS
- 3 format de liaisons des données
  - HTTP Get
  - HTTP Post
  - Format Soap
    - Avantage = Passage d'objet complexe type DataSet

Web services en .NET – 2.0

- Modèle basique de programmation d'un WS Asp.net
  - [WebService]/[WebMethod] : Définit l'ensemble des points d'entrée du web service

Web services en .NET – 2.0

- Un décorateur de méthode pour tout faire
  - [WebMethod]
- Plusieurs attributs au décorateur
  - EnableSession = true
  - BufferResponse=false
  - CacheDuration=nombre de secondes
  - Description="une description"
  - MessageName="alias nom de méthode"
  - TransactionOption=TransactionOption.[Disabled ou Required ou Supported ou NotSupported ou RequiresNew]

Web services en .NET – 2.0

- Décorateur de classe
  - Modification de l'ensemble des propriétés du WS
  - [WebService]
    - Description
    - Name
    - Namespace

Web services en .NET – 2.0

- Création d'un web service simple
- Objectif = Validation d'un numéro de carte bancaire
- Algo
  1. Multiplier chaque chiffre impair du numéro par 2.
  2. Exprimer le résultat du calcul modulo 9.
  3. Additionner tous les chiffres obtenus.
  4. Si la somme obtenue est un multiple de 10, le numéro de carte est valide.

Web services

- L'implémentation
  - Projet Web Service

```
public bool Validate(string cardNumber, DateTime
expDate)
  {
    if (expDate >= DateTime.Today)
    {
      int total = 0;int temp = 0;
      char[] ccDigits = cardNumber.ToCharArray();

      for (int i = 0; i < cardNumber.Length; i++)
      {
        if (((i + 1) % 2) == 0) total +=
int.Parse(ccDigits[i].ToString());
else
        {
          temp =
int.Parse(ccDigits[i].ToString()) * 2;
          if (temp > 9)
          {
            temp = (int)temp - 9;
          }
          total += temp;
        }
      }
      if ((total % 10) == 0) return true;
      else return false;
    }
    else
    {
      return false;
    }
  }
```

Web services en .NET – 2.0

- Transformation en Web service

```
[WebMethod]
  public bool Validate(string cardNumber, DateTime expDate)
{
        … …
}
```

Compiler et tester

Web services

Test d'un web service depuis Visual studio 2005

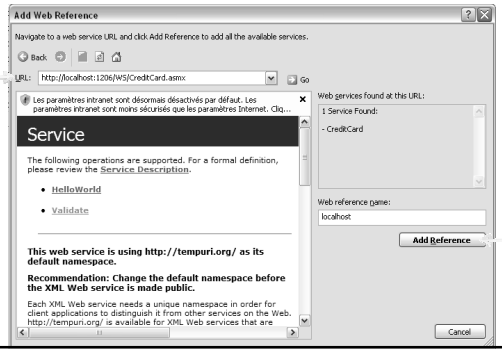
Web services

- Résultat

```
<?xml version="1.0" encoding="utf-8" ?>
<boolean
xmlns="http://tempuri.org/">false</boolean>
```


Web services en .NET – 2.0

- Ajouter une référence depuis VS 2005



Web services en .NET – 2.0

- Gestion de l'intellissense

```
protected void Button1_Click(object sender, EventArgs e)
{
    localhost.Service ws = new localhost.Service();
    if (ws.Validate(TextBox1.Text, DateTime.Now.AddDays(1)))
    else
}
protecte                                    nder, EventArgs e)
{
//////                                      AWSECommerceService();
//////                                      arch();
    MonA
    ms.S                                    Z4JPN82";
    resultSetDataGrid.DataSource = ms.SendQueryToAmazon(TextB
    resultSetDataGrid.VirtualItemCount = ms.TotalItems;
    resultSetDataGrid.DataBind();
```

SoapVersion
Timeout
ToString
UnsafeAuthenticatedConnectionSharing
Url
UseDefaultCredentials
UserAgent
Validate
ValidateAsync
ValidateCompleted

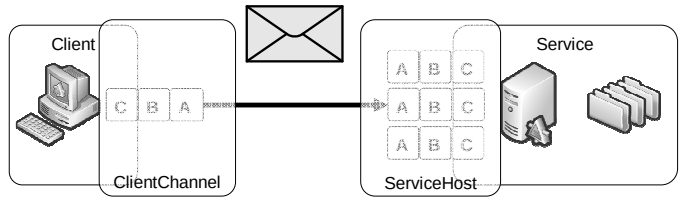
Web services en .NET – 2.0

- Génération d'une classe proxy
  - WSDL.exe
  - Utilitaire en ligne de commande
    - Syntaxe : WSDL.exe [options] urlDuWSDL
  - Paramètres
    - /u:user = nom de l'utilisateur accédant au WS
    - /p:pass = passe de l'utilisateur accédant au WS

Web services en .NET – 3.5

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

Web services en .NET – 3.5



Address	Binding	Contract
Où ?	Comment ?	Quoi ?

Endpoint

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 utilisé dans les méthodes présentées par le web service

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

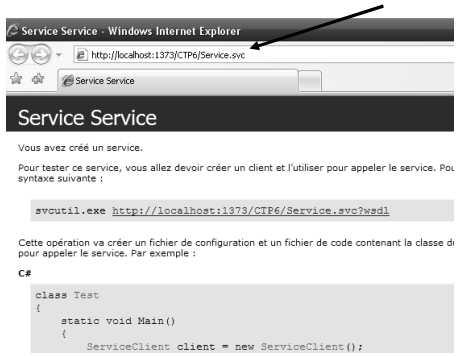
```
public class Service : IService
{
 public double CurrencyConvert(double amount, SupportedCurrency from,
    SupportedCurrency to)
  {
    net.webservicex.www.CurrencyConvertor cc = new
    net.webservicex.www.CurrencyConvertor();
    cc.UseDefaultCredentials = true;
    cc.Proxy = new System.Net.WebProxy();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

    net.webservicex.www.Currency wwwFrom = GetCurrency(from);
    net.webservicex.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.webservicex.net/CurrencyConvertor.asmx?WSDL

Web services en .NET – 3.5



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

Pour tester :
Ecriture de tests unitaires

Tests unitaires

- But
  - Permettre la validation de briques de code par l'exécution et l'analyse du résultat
- Caractéristiques
  - Automatique
  - Répétable
  - Disponible
- Microsoft Unit Testing Framework intégré à Visual Studio (projet de test)

Tests unitaires

- Modèle basique de programmation de test unitaire
  - [TestClass]/[TestMethod]
  - Une méthode de test par fonctionnalité à évaluer
- Analyse du résultat
  - Class statique Assert
  - Méthodes utiles
    - IsNotNull
    - AreEqual
    - IsTrue

Tests unitaires

- Exemple :

```
[TestMethod]
public void TestConversion()
{
    ServiceClient clt = new ServiceClient();

    double res = clt.CurrencyConvert(10, SupportedCurrency.Euro,
    SupportedCurrency.LivreSterling);

    Assert.IsTrue(res < 10);
}
```