

UDDI

(Universal Description Discovery and Integration):

Applications réparties

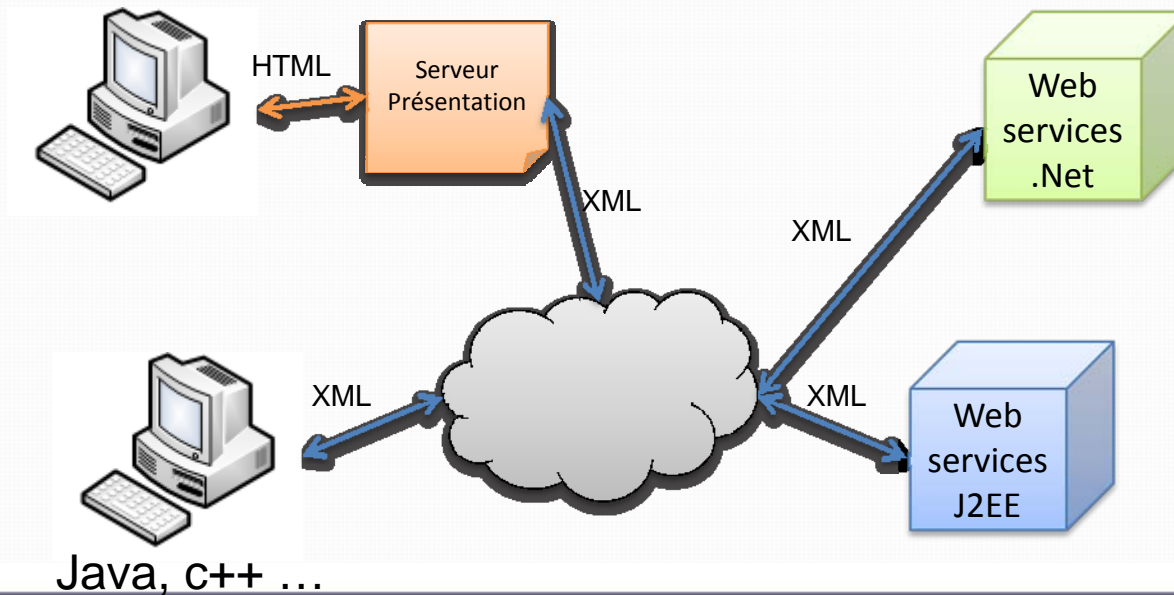
Nicolas Ferry
Stéphane Lavirotte
Dino Lopez-Pacheco

D'après les cours de :
Jenny Benois-Pineau, LABRI
Didier Donsez, LIG

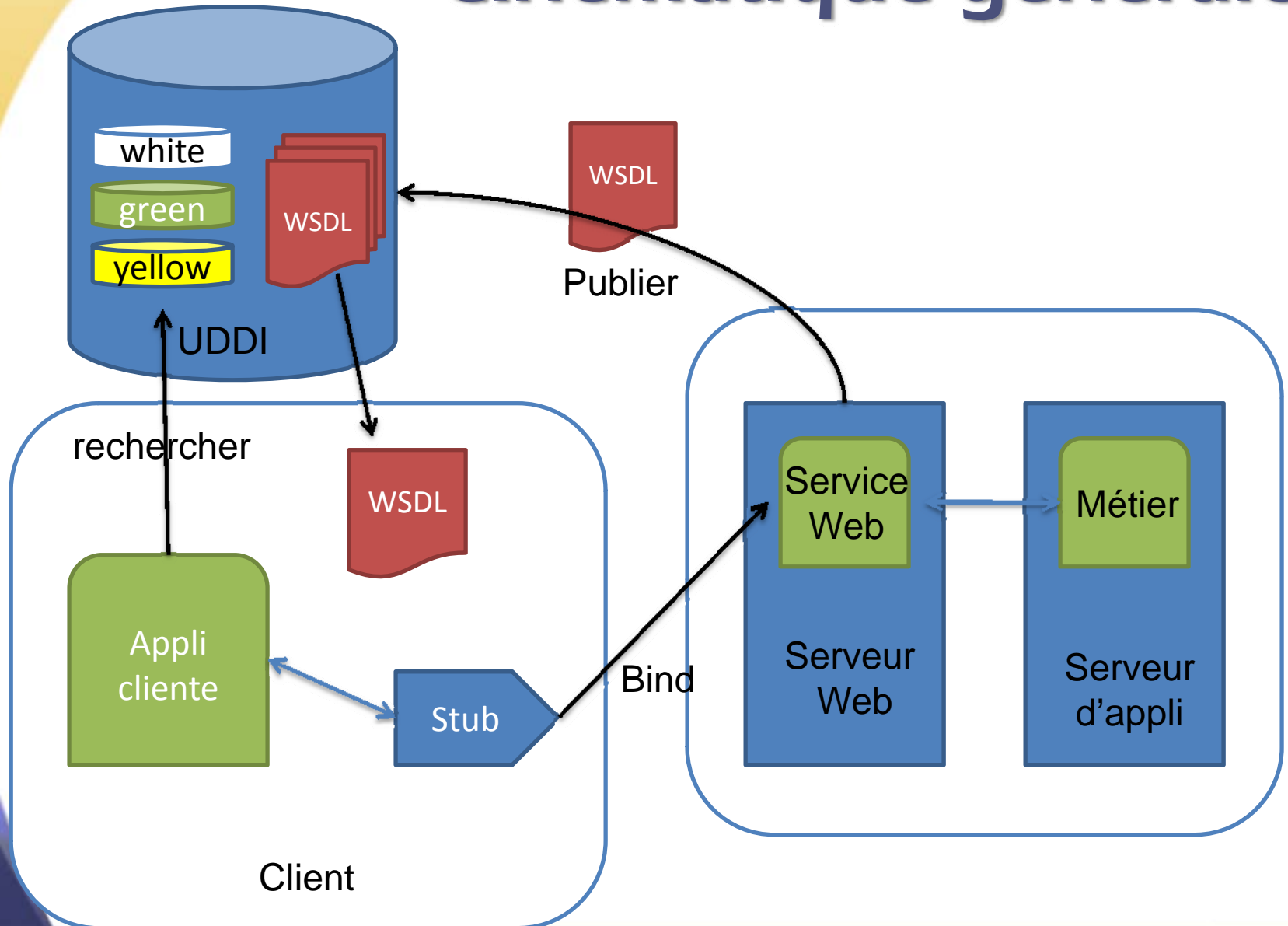
Un Service Web, c'est quoi ?

✓ Caractéristiques

- Réutilisable
- Indépendamment de
 - la plate-forme (UNIX, Windows, ...)
 - du langage pour l'implémentation (VB, C#, Java ...)
 - la plate-forme de développement sous-jacente (.NET, J2EE, Axis...)



Cinématique générale



Pourquoi UDDI ?

- ✓ SOAP/WSDL permettent d'invoquer des services Web mais pas de les rechercher
- ✓ UDDI ne répond pas aux mêmes besoins que les LDAP qui référencent aussi bien personnes, objets que dispositifs matériels
- ✓ Les annuaires UDDI sont orientés B2B
 - On y trouve des informations techniques (WSDL)
 - Des informations sur l'entreprise

Introduction

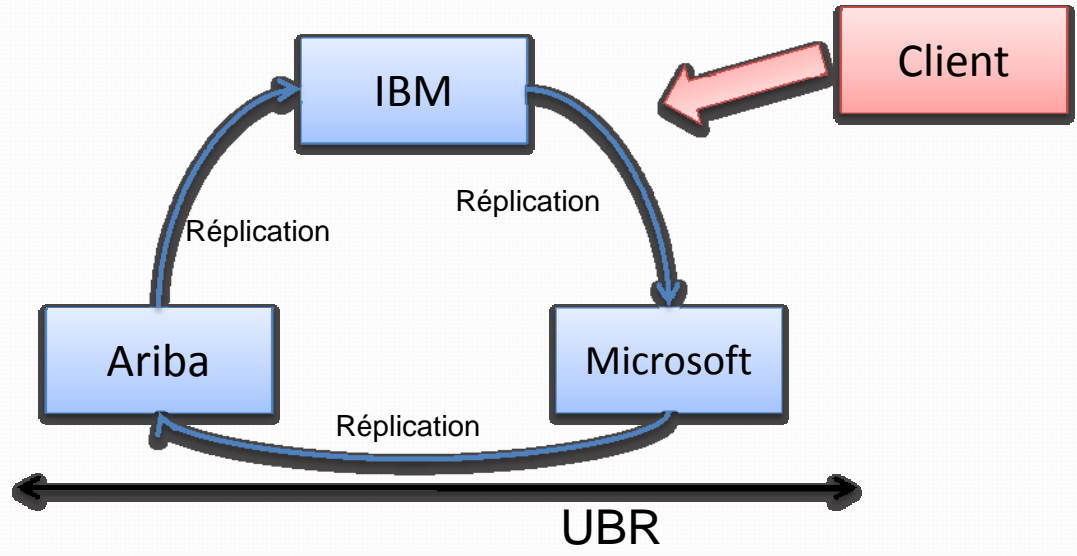
- ✓ **Le standard UDDI a été proposé par Microsoft, IBM et Ariba en septembre 2000.**
 - **Trois annuaires majeurs:**
 - **Le premier est hébergé par Microsoft.**
 - **Le second par IBM.**
 - **Le troisième (plus récent) par HP.**

- ✓ **UDDI est une spécification qui décrit comment publier et découvrir des services Web sur un réseau**
 - **Objectif : Faire un annuaire mondial d'entreprise et de leurs services.**
 - **Deux types d'actions possibles sur un annuaire UDDI**
 - **Publier son service**
 - **Rechercher un service**

- ✓ **Grammaire XML (schéma XML) selon la spécification**
 - **Soumission/interrogation basées sur SOAP**
 - **Mais aussi possible en XML-RPC ou Corba**

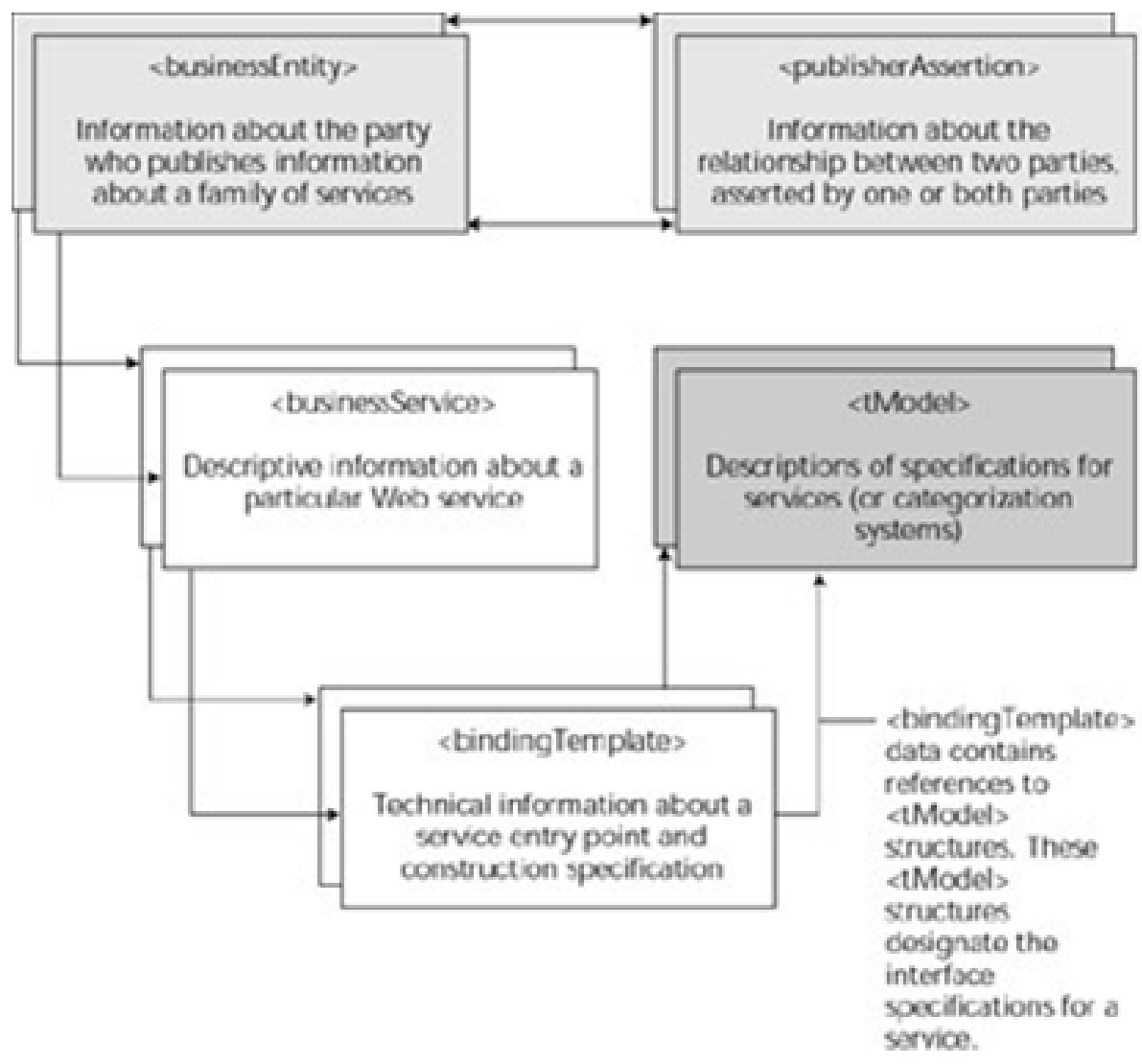
UDDI architecture

- ✓ De nombreux nœuds dans un UBR (UDDI Business Registry)
 - Donne l'impression d'accéder à un système unique mais en réalité composé d'un ensemble de nœuds
 - Les données sont synchronisées par réplication toutes les 24h



- ✓ On peut mettre en place son propre annuaire privé.

Modèle UDDI



Le modèle UDDI

- ✓ Le modèle UDDI comporte 5 structures de données
 - **BusinessEntity** : Ensemble d'informations sur l'entreprise qui expose le service
 - **BusinessService** : Ensemble d'informations sur les services exposés par l'entreprise
 - **BindingTemplate** : Ensemble d'informations sur le lieu d'hébergement du service
 - **tModel** : Ensemble d'informations sur le mode d'accès au service (WSDL !)
 - **publisherAssertion** : Ensemble d'informations contractuelles pour accéder au service

Exemple Schéma BusinessEntity

```
<element name="businessEntity" type="BusinessEntity" />  
<complexType name="BusinessEntity">  
  <sequence>  
    <element ref="uddi:name" />  
    <element ref="uddi:description" />  
    <element ref="uddi:contacts" maxOccurs="unbounded" />  
    <element ref="uddi:businessServices" minOccurs="0" />  
    <element ref="uddi:identifierBag" />  
    <element ref="uddi:categoryBag" />  
  </sequence>  
  <attribute name="businessKey" type="string" use="required" />  
  <attribute name="operational" type="boolean" use="optional" />  
  <attribute name="authorizedName" type="string" use="optional" /> *  
</complexType>
```

BusinessEntity

- uddi:name
- uddi:description
- uddi:contacts
- uddi:businessServices
- uddi:identifierBag
- uddi:categoryBag

- businessKey
- authorizedName



Exemple Schéma BusinessService

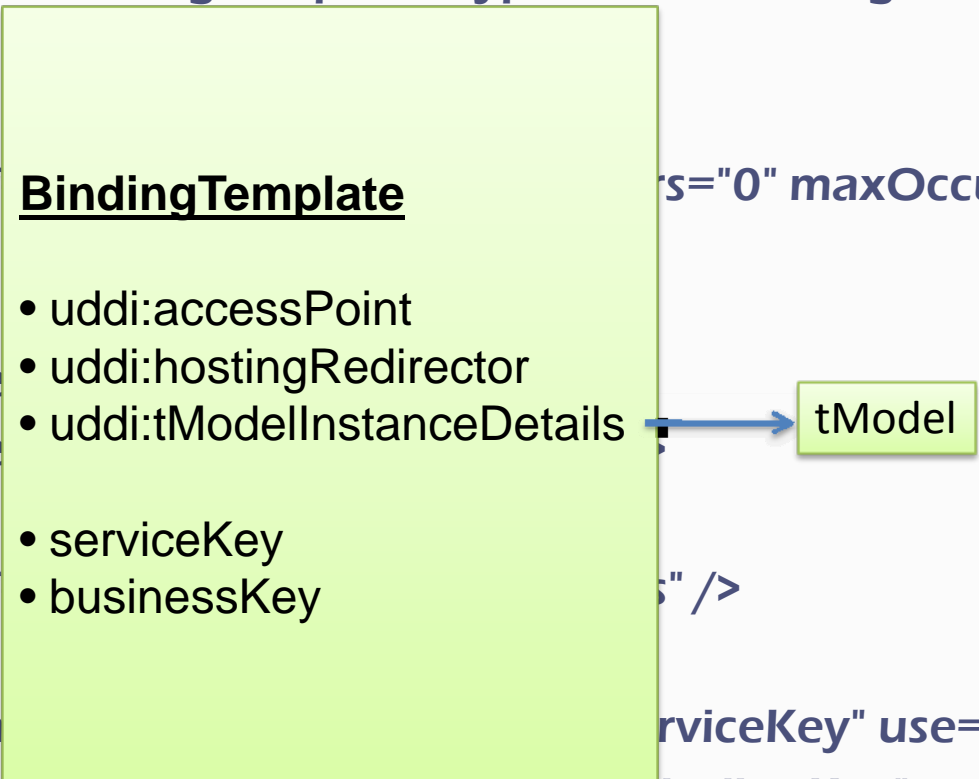
```
<element name="businessService" />
<complexType name="BusinessService">
  <sequence>
    <element ref="uddi:name" minOccurs="0" maxOccurs="unbounded" />
    <element ref="uddi:description" minOccurs="0" maxOccurs="unbounded" />
    <element ref="uddi:bindingTemplate" minOccurs="0" />
    <element ref="uddi:categoryBag" minOccurs="0" />
  </sequence>
  <attribute name="serviceKey" use="required" />
  <attribute name="businessKey" use="optional" />
</complexType>
```

BusinessService

- uddi:name
- uddi:description
- uddi:bindingTemplate
- uddi:categoryBag
- serviceKey
- businessKey

Exemple Schéma BindingTemplate

```
<element name="bindingTemplate" type="uddi:bindingTemplate" />  
<complexType  
  <sequence>  
    <element ref="uddi:accessPoint" minOccurs="0" maxOccurs="unbounded" />  
    <choice>  
      <element ref="uddi:hostingRedirector" />  
      <element ref="uddi:tModelInstanceDetails" />  
    </choice>  
    <element ref="uddi:businessKey" />  
  </sequence>  
  <attribute name="serviceKey" use="optional" />  
  <attribute name="bindingKey" type="uddi:bindingKey" use="required" />  
</complexType>
```



WSDL et tModel

- ✓ **tModel de Hertz** (http://www.tutorialspoint.com/uddi/uddi_with_wsdl.htm)
- ✓ **<tModel authorizedName="..." operator="..." tModelKey="...">**
 - <name>HertzReserveService</name>**
 - <description xml:lang="en">** WSDL description of the Hertz reservation service interface **</description>**
 - <overviewDoc>**
 - <description xml:lang="en">** WSDL source document. **</description>**
 - <overviewURL>**
http://mach3.ebphost.net/wsdl/hertz_reserve.wsdl
</overviewURL>
 - </overviewDoc>**
 - <categoryBag>**
 - <keyedReference tModelKey="uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4" keyName="uddi-org:types" keyValue="wsdlSpec"/>**
 - </categoryBag>**
- </tModel>**

Recherche de Web services

- ✓ Les annuaires UDDI ont pour but de localiser des services Web hébergés dans le monde entier
- ✓ Différents types de recherches:
 - Pages blanches (*BusinessEntity*)
 - On cherche un service par contact, nom et adresse
 - Pages jaunes (*BusinessService*)
 - On recherche un service par sujet, par domaine
 - Pages vertes (*BindingTemplate*)
 - On recherche un service en fonction de ces caractéristiques techniques
- ✓ Les opérateurs UDDI vous garantissent la sécurité et l'intégrité des services Web contenus dans un annuaire

Primitives de recherche

- ✓ **find_binding** : pour obtenir la liaison du service considéré
 - ✓ **find_business** : pour obtenir l'identité de l'entreprise qui expose le service
 - ✓ **find_relatedbusiness** : pour obtenir les entreprises en relations avec celle qui publie le service
 - ✓ **find_service** : pour obtenir l'interface du service
 - ✓ **find_tModel** : récupère le modèle de données associées
-
- ✓ **Pour plus de détails**
 - **get_businessDetail**
 - **get_serviceDetail**
 - **get_bindingDetail**
 - **get_tModelDetail**

Publication de service Web

- ✓ La publication d'un service Web requiert que l'entreprise s'authentifie auprès de l'opérateur UDDI
- ✓ Il faut fournir les données nécessaire à l'exploitation du service Web (IP, Nom de domaine, modalités d'utilisation ...)

Primitives de publication

✓ Enregistrer

- save_business
- save_service
- save_binding
- save_tModel

✓ Détruire

- delete_business
- delete_service
- delete_binding
- delete_tModel

✓ Sécurité

- get_authToken
- discard_authToken

caractéristiques importantes

- ✓ Neutre en terme de protocole – comme tout registre, il peut y avoir des pointeurs sur n'importe quoi (email, web page ...)
- ✓ Possibilité de faire des recherches par domaine d'activité, service, Web Service, binding
- ✓ Pas de QoS
- ✓ Nécessite un modérateur
 - Risques d'entrées erronées, de doublons, de fraude
- ✓ Nœuds privés et publics autorisés

Finding a Business

```
using Microsoft.Uddi;  
using Microsoft.Uddi.Api;  
  
try {  
    // Configure the connection for the UDDI node that is to be accessed  
    Inquire.Url = "http://uddi.rte.microsoft.com/inquire";  
  
    // Create an object to find a business  
    FindBusiness fb = new FindBusiness();  
    fb.Names.Add("Microsoft");  
  
    // Send the prepared find business request  
    BusinessList bizList = fb.Send();  
    Console.WriteLine("Businesses found=" +  
        bizList.BusinessInfos.Count.ToString());  
} catch (UddiException e) {  
    Console.WriteLine("UDDI exception: " + e.Number + " - " + e.Message);  
} catch (Exception e) {  
    Console.WriteLine("General exception: " + e.Message);  
}
```

Saving a business

```

try {
    // Configure for the site that are going to access
    Publish.Url = "https://uddi.rte.microsoft.com/publish";
    Publish.User = " *** insert your user name *** ";
    Publish.Password = " *** insert your password *** ";

    // Create an object to save a business
    SaveBusiness sb = new SaveBusiness();

    // Add a business entity and allocate a name
    sb.BusinessEntities.Add();
    sb.BusinessEntities[0].Names.Add(" *** insert your business name *** ");

    // Send the prepared save business request
    BusinessDetail savedB = sb.Send();

    // Interpret the returned business detail to examine the allocated business key
    Console.WriteLine("Business: " + savedB.BusinessEntities[0].Names[0].Text);
    Console.WriteLine(" Allocated key: " + savedB.BusinessEntities[0].BusinessKey);

} catch (UddiException e) {
    Console.WriteLine("UDDI exception: " + e.Number + " - " + e.Message);}
catch (Exception e) {
    Console.WriteLine("General exception: " + e.Message);}
  
```

Deleting a business

```
try {  
    // Configure for the site that are going to access  
    Publish.Url = "https://uddi.rte.microsoft.com/publish";  
    Publish.User = " *** insert your user name *** ";  
    Publish.Password = " *** insert your password *** ";  
  
    // Provide the key to the business you want to delete  
    // e.g. string bizkey = "86673abe-3ca1-4147-893c-83ac0d941c76";  
    string bizkey = " *** insert the business key *** ";  
  
    DeleteBusiness db = new DeleteBusiness();  
    db.BusinessKeys.Add(bizkey);  
  
    DispositionReport drep = db.Send();  
    Console.WriteLine("Disposition report: " +  
        drep.Results[0].ErrInfo.ErrCode.ToString());  
} catch (UddiException e) {  
    Console.WriteLine("UDDI exception: " + e.Number + " - " + e.Message);}  
catch (Exception e) {  
    Console.WriteLine("General exception: " + e.Message);}
```

WS Inspection

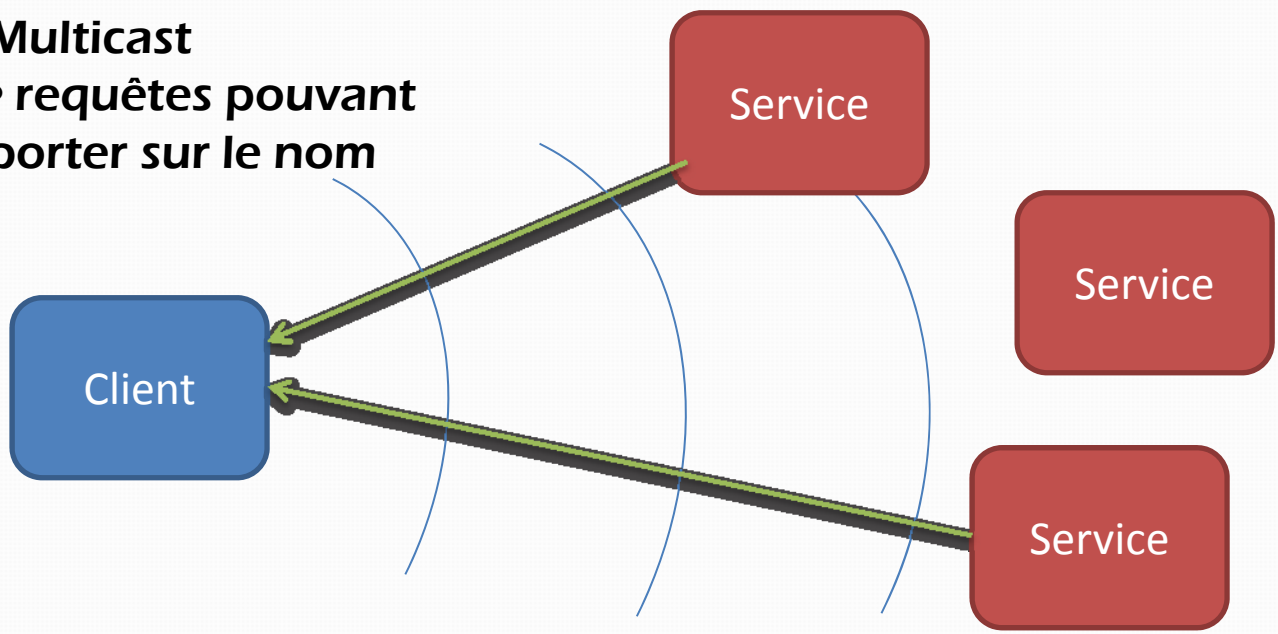
- ✓ IBM et microsoft ont proposé une autre norme plus simple:
 - WS inspection
- ✓ Mêmes objectifs que UDDI, les services sont aussi décrits en WSDL
- ✓ WSIL représente une entité spécifique, ses services, ses contacts, est fourni par celui qui le représente
 - On considère que l'on connaît déjà le fournisseur donc pas d'équivalent à businessEntity

WS-Discovery

✓ Sans annuaire !

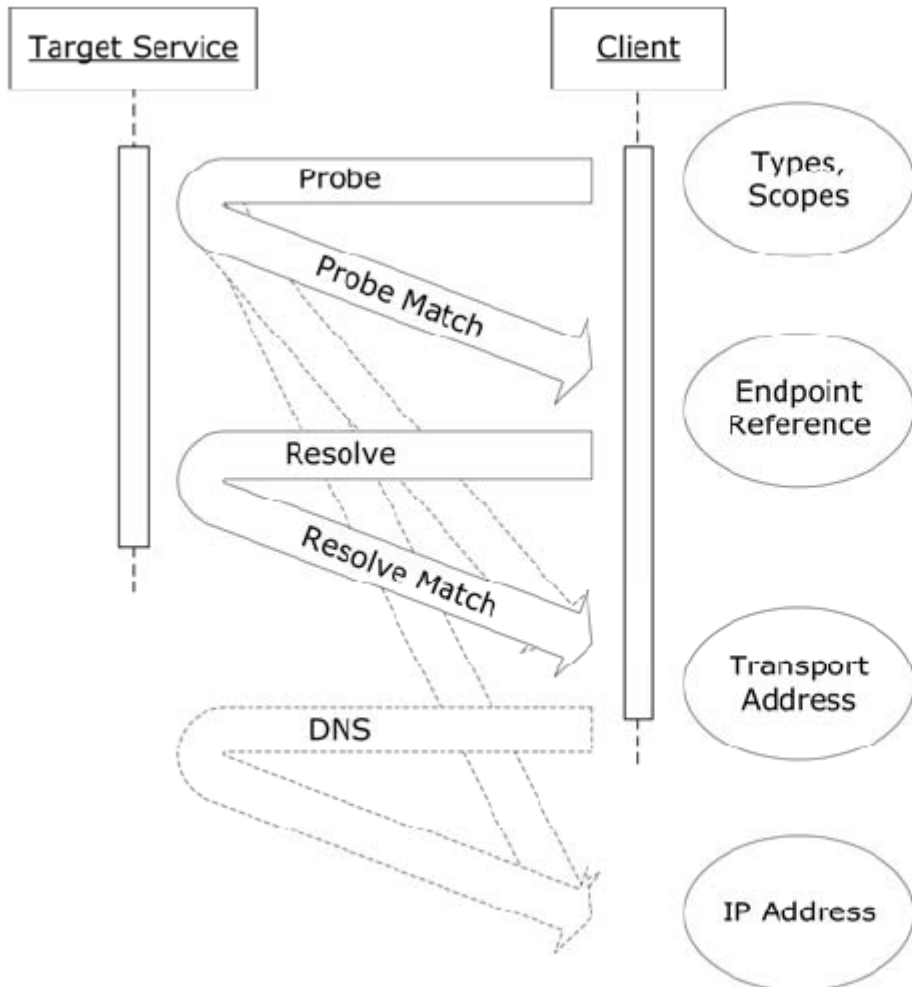
Multicast

- requêtes pouvant porter sur le nom



Quand un service rejoint le réseau il envoie également un message l'annonçant (moins de polling)

WS-Discovery



Examples

d:PrintBasic

```
<a:EndpointReference>  
  <a:Address>  
    uuid:57205e8e-0ddf-463f-9f36-8ff944adb4  
  </a:Address>  
  <a:ReferenceProperties>  
    <acme:Svc>power</acme:Svc>  
  </a:ReferenceProperties>  
</a:EndpointReference>
```

http://printer2.acme.example.com/power_ctr

216.16.43.13

Web services pour dispositifs

Une approche sans annuaire !

Motivations

- ✓ **Faire travailler des dispositifs les uns avec les autres**
- ✓ **Limitation à de l'envoi de données sur le réseau**
- ✓ **Conserver le code de l'application «privé»**
 - Etre agnostique par rapport au réseau, à l'OS, au langage de programmation
 - Mise à jour de l'implémentation sans affecter l'interopérabilité
 - **Augmenter les performances**
 - **Augmenter les possibilités**
- ✓ **Etre d'accord sur le format et le sens des données**
 - Choix de protocoles éprouvés (W3C, IETF)
 - Définition des Dispositifs et Services par un Forum

Introduction

- ✓ **Le forum UPnP™ –une initiative industrielle**
 - Conçue pour permettre la connectivité simple et robuste entre dispositifs de différents fournisseurs

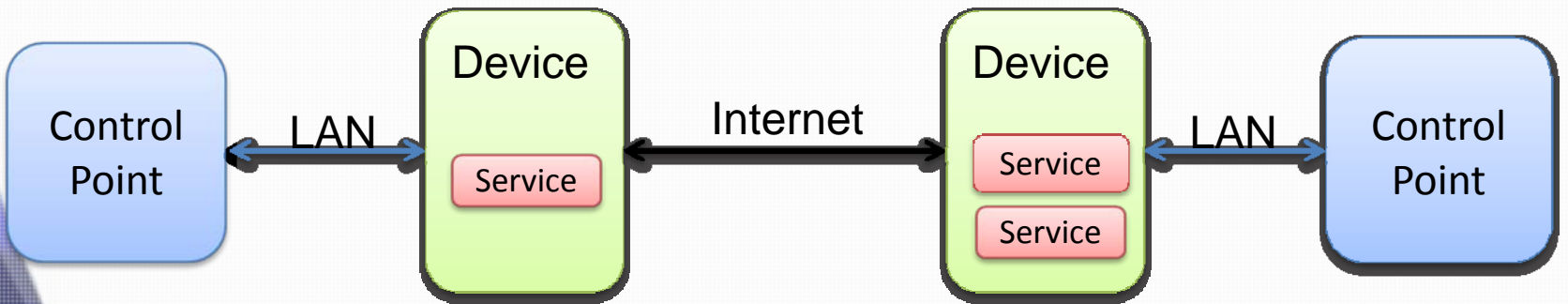
- ✓ **L'architecture d'UPnP permet la découverte puis le contrôle de dispositifs sur un réseau, indépendant – Des OS**
 - Des langages de programmation,
 - Des réseaux physiques et locaux.

- ✓ **La technologie d'UPnP s'appuie pour cela sur des normes existantes IP, UDP et TCP, HTTP, XML et SOAP**

- ✓ **<http://www.upnp.org/>**

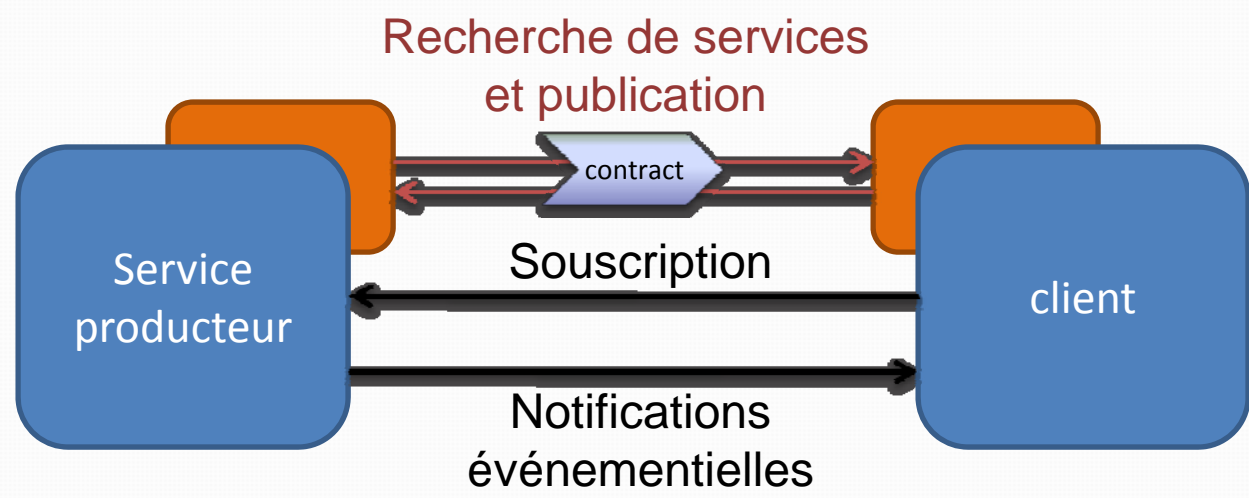
Architecture

- ✓ **Services** : plus petite entité, propose des actions
- ✓ **Device** : entité se composant de services et d'autres device (un root device inclus des embedded device) (Serveur)
- ✓ **Control point** : permet de découvrir et de contrôler des devices. (Client)
- ✓ **Un dispositif peut être control point et device**



Spécificités

- ✓ Recherche et Découverte distribuée
 - Les entités logicielles se découvrent dans leurs environnement immédiat.
- ✓ Communication événementielle



- ✓ Aujourd'hui UPnP
- ✓ Demain ... DPWS (Device Profile For Web Services)

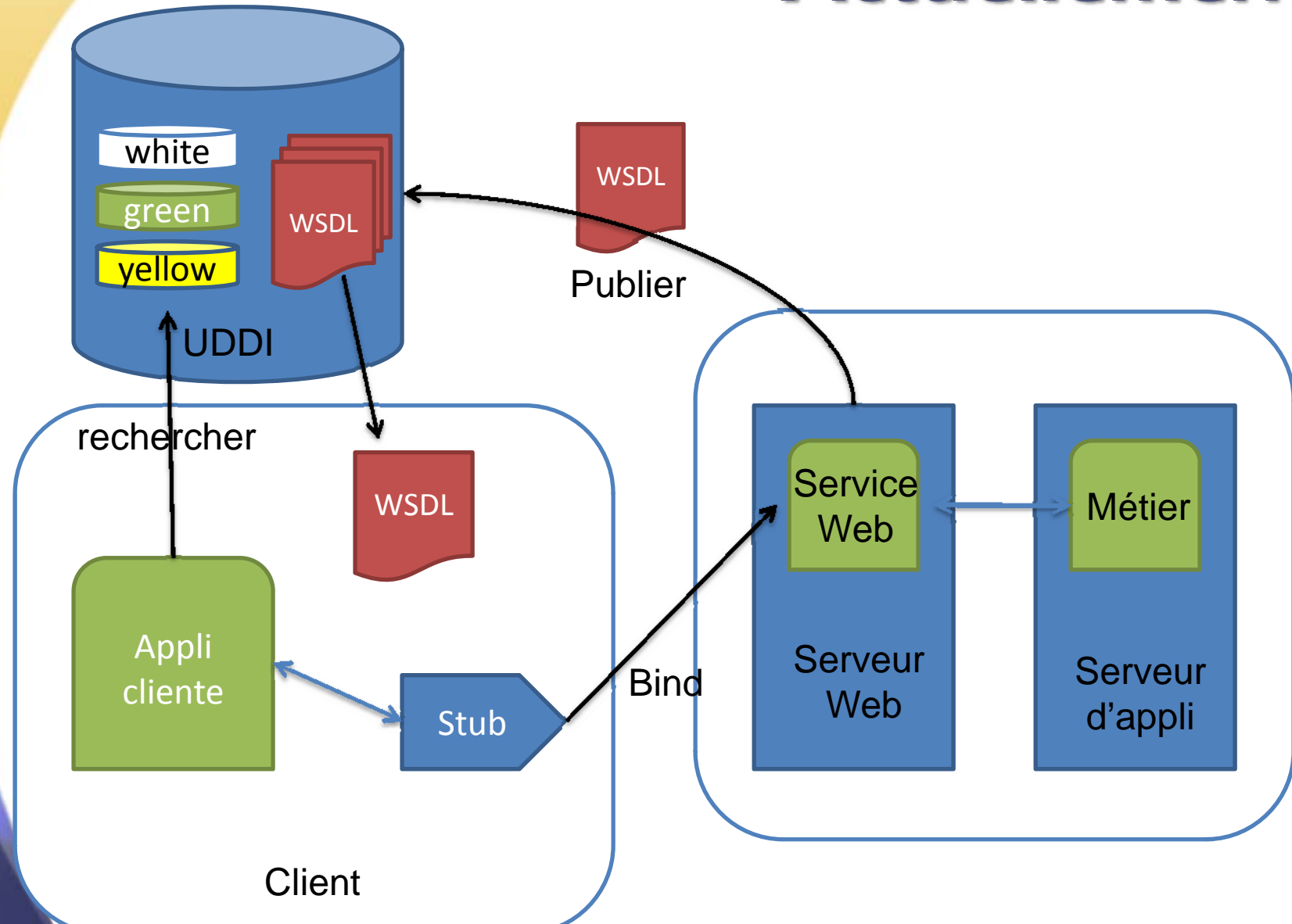
La pile UPnP

1. **Adressing** : acquisition d'une adresse ip dans le réseau (Auto-IP si pas de DHCP)
2. **Discovery** : mise en relation des entités du réseau avec SSDP. Cette phase est utilisée par les control points ainsi que les device pour s'identifier sur un réseau puis pour confirmer leurs présences.
3. **Description** : phase visant à fournir aux points de contrôles des informations décrivant les devices, services. Cette description est écrite en XML.
4. **Control** : permet au point de contrôle d'envoyer des commandes aux devices
5. **Eventing** : informer les points de contrôle des changements d'état.
6. **Presentation** : un device peut fournir une URL de présentation des devices.

Conclusion

UDDI et UPnP

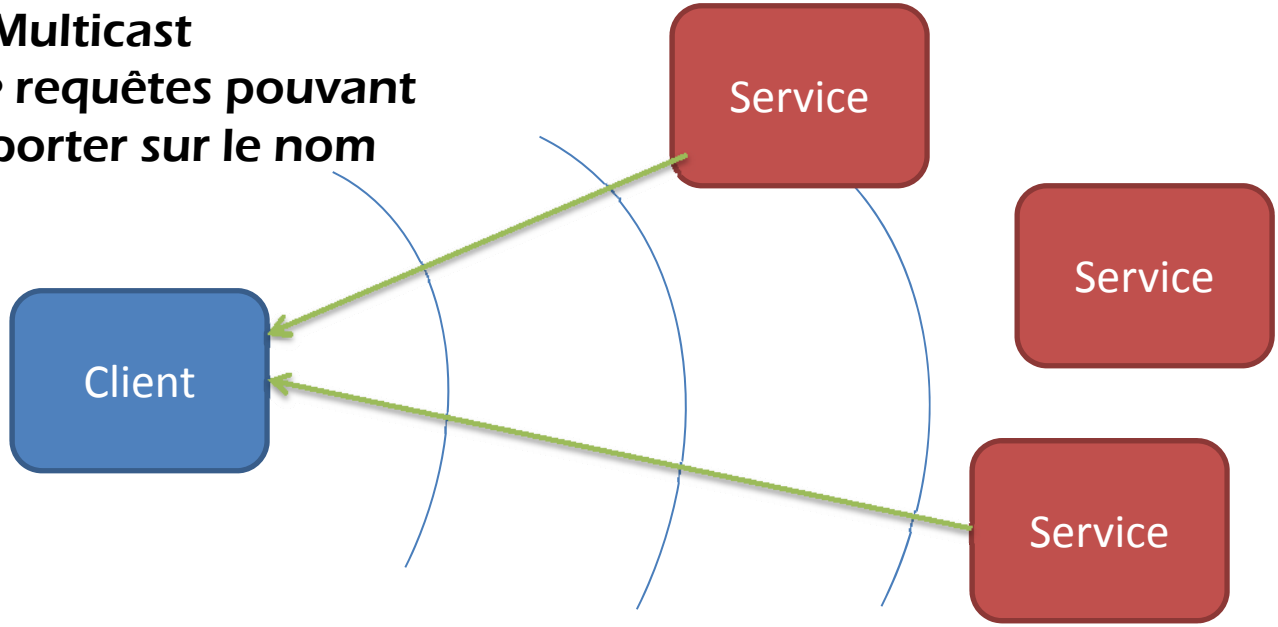
Actuellement



On s'oriente de plus en plus vers

Multicast

- requêtes pouvant porter sur le nom



Quand un service rejoint le réseau il envoie également un message l'annonçant (moins de polling)

Conclusion Générale

Web services

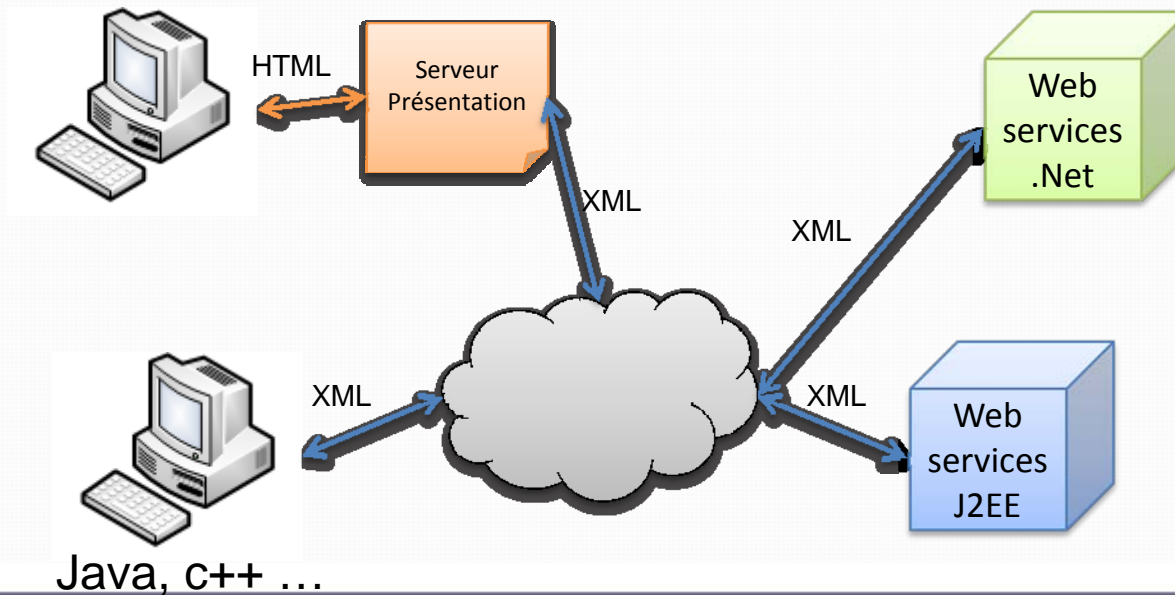
Les Services Web

- ✓ La 3ème génération du Web
- ✓ Technologies Standards du Web
 - SOAP
 - WSDL
- ✓ Technologies non standardisées
 - UDDI
 - WSDD, WSFL, ASMX, ...

Un Service Web, c'est quoi ?

✓ Caractéristiques

- Réutilisable
- Indépendamment de
 - la plate-forme (UNIX, Windows, ...)
 - du langage pour l'implémentation (VB, C#, Java ...)
 - la plate-forme de développement sous-jacente (.NET, J2EE, Axis...)



Ils permettent

- ✓ **Les Services Web permettent d'interconnecter :**
 - Différentes entreprises
 - Différents matériels
 - Différentes applications
 - Différents clients

- ✓ **Remplacer les outils actuels (RPC, DCOM, CORBA, RMI) par une approche entièrement ouverte et interopérable, basée sur la généralisation des serveurs Web avec scripts CGI.**

- ✓ **Faire interagir des composants hétérogènes, distants, et indépendants avec un protocole standard (SOAP).**

- ✓ **Attention on arrive vite au plat de spaghettis avec des services interconnectés partout !!**
 - Réponses : SOA et orchestrations

Références

- ✓ Cours UDDI Didier Donsez
<http://membres-liglab.imag.fr/donsez/cours/wsdluddi.pdf>
- ✓ Spécification WS-Discovery
<http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>
- ✓ Code
<http://www.informit.com/articles/article.aspx?p=101150&seqNum=4>
- ✓ Cours UPnP Didier Donsez
<http://membres-liglab.imag.fr/donsez/cours/upnposgi/tutorial.htm>
- ✓ Cours UPnP Stéphane Lavirotte/Gaetan Rey/Jean Yves Tigli
<http://kistren.polytech.unice.fr/cours/iam01/Cours%20UPnP.pdf>
- ✓ http://www.tutorialspoint.com/uddi/uddi_with_wsdl.htm