Université Nice SOPHIA ANTIPOLIS

UDDI-REST : Applications réparties

Nicolas Ferry
Stéphane Lavirotte
Jean Yves Tigli

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

D'après le livre :
Services Web avec J2EE et .NET – Libero Maesano, Christian Bernard, Xavier Le Galles

REST

Reposant, Paisible

REST

- ✓ REpresentational State Transfer
  - Terme introduit par Roy Fielding dans sa thèse en 2000
  - REST n'est pas un protocole ou un format
  - C'est une manière de construire une application pour les systèmes distribués (style architectural)

- ✓ Architecture Orientée Ressource
  - Toute information qui peut être nommée est une ressource
  - Une ressource est identifiée par un identificateur (URI)

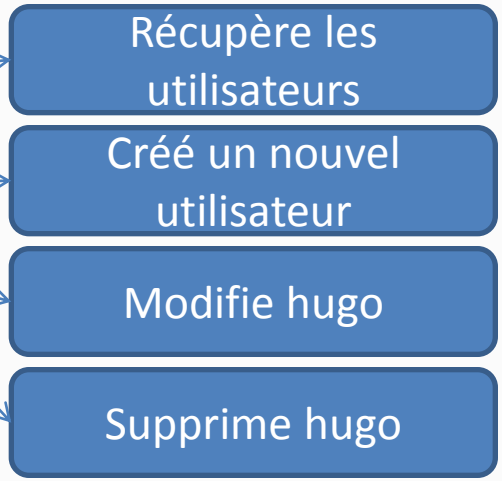
REST n'est pas un Standard

- ✓ REST n'est pas un standard
  - Pas de recommandation du W3C
  - Pas de toolkit éditée par Microsoft, IBM ou Sun
  - Mais utilise des standards: HTTP, URL, XML, HTML, MIME, ...
- ✓ REST est un style d'architecture
  - C'est un design pattern pour l'implémentation d'un système
  - Principes:
    - Les requêtes sont client-serveur
    - Les requêtes sont stateless (sans état)
    - Les clients accèdent à des ressources nommées: une ressources est représentée par une URL.
    - Les clients et serveurs adhèrent à une interface uniforme: toutes les ressources sont accédées via une interface générique: les méthodes HTTP

URI et Méthodes HTTP

- ✓ Chaque URI identifie une ressource
  - http://api.domain.com/users/hugo
- ✓ Une URI ne contient pas de verbe !
- ✓ Une même URI peut être appelée avec différentes méthodes HTTP

- ✓ GET http://api.domain.com/users
- ✓ POST http://api.domain.com/users
- ✓ PUT http://api.domain.com/users/hugo
- ✓ DELETE http://api.domain.com/users/hugo



Tiré de : http://2010.rmll.info/IMG/pdf/1-rmll-2010-07-06.pdf

Avantages de REST

- ✓ Avantages:
  - Application est plus simple à entretenir: les liens sont mieux structurés (les liens sont le moteur de l'état de l'application)
  - Absence de gestion d'état du client sur le serveur:
    - moindre consommation mémoire
    - plus de simplicité (capacité à répondre à plus de requêtes)
    - possibilité de répartir les requêtes (*load balancing*)
      - Meilleure évolutivité et tolérance aux pannes
  - Utilisation des caractéristiques d'HTTP
    - SOAP ne pré-suppose pas de protocole même s'il utilise la plupart du temps HTTP (redites entre SOAP et HTTP)
  - Utilisation des URI pour représenter (nommer) des ressources
    - Possibilité de mise en cache très facilement

Inconvénients de REST

- ✓ Inconvénients:
  - Le client doit conserver toutes les données nécessaires pour le bon déroulement d'une requête
    - Consommation de la bande passante plus importante
  - L'utilisation d'un formulaire HTML envoyant ses données avec une méthode comme DELETE en général pas compris
    - on émule ce comportement avec un champ caché qui transmettra le type de méthode d'envoi des données à l'application

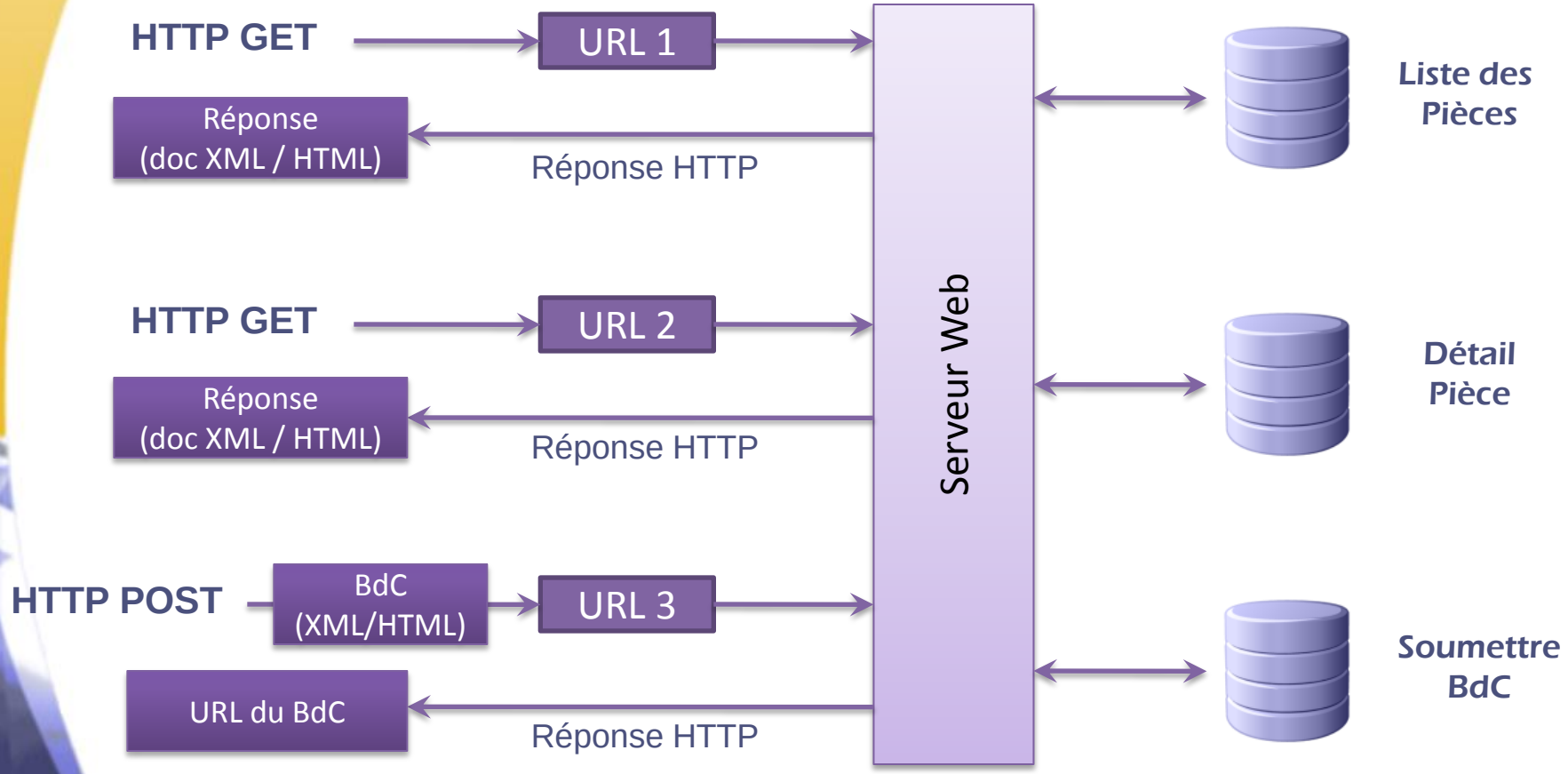
Comparaison de REST vs SOAP

Par l'exemple

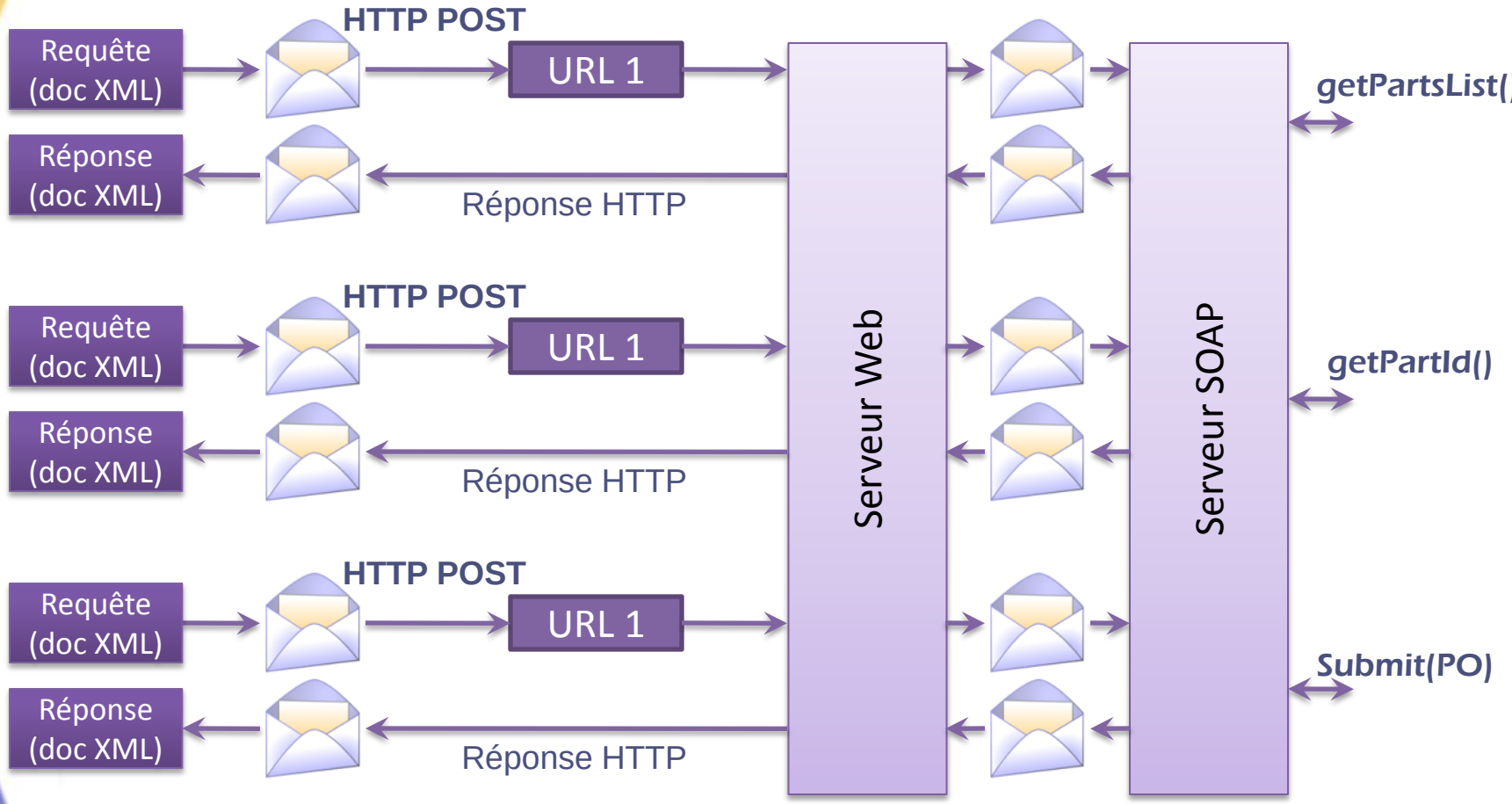
Comprendre REST par l'Exemple

- ✓ Ce type d'architecture se comprend mieux par l'exemple
- ✓ Exemple d'un Entrepôt de Pièces Détachées
  - Services Web accessibles pour les clients
    - Obtenir la liste des pièces détachées
    - Obtenir des informations détaillées sur une pièce donnée
    - Soumettre un bon de commande
- ✓ Nous allons étudier la mise en œuvre de cet exemple
  - Via REST
  - Via SOAP

Représentation REST

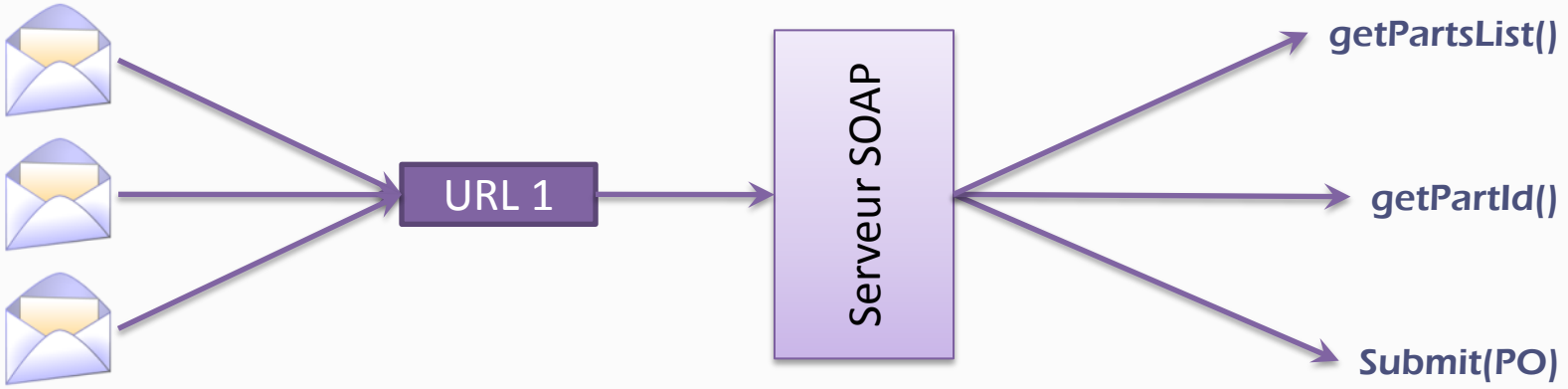


Représentation SOAP



Implémentation du WS avec SOAP

- Même si ce n'est pas une obligation, en SOAP, forme de tunneling sur la même URL



- Exemple de message SOAP:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <p:getPartsList xmlns:p="http://www.parts-depot.com"/>
  </soap:Body>
</soap:Envelope>
```

Données Retournées en REST

- ✓ **Exemple de réponse :**

```
<?xml version="1.0"?>
<p:Parts xmlns:p="http://www.parts-depot.com"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.parts-depot.comhttp://www.parts-
   depot.com/parts.xsd">
  <Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
  <Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
  <Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/>
  <Part id="00348" xlink:href="http://www.parts-depot.com/parts/00348"/>
</p:Parts>
```

- ✓ **On notera que chaque pièce de la liste a une URL qui permet d'y accéder. C'est un élément clé de REST**

Données Retournées en SOAP

- ✓ **Exemple de réponse:**

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <p:getPartsListResponse xmlns:p="http://www.parts-depot.com">
      <Parts>
        <Part-ID>00345</Part-ID>
        <Part-ID>00346</Part-ID>
        <Part-ID>00347</Part-ID>
        <Part-ID>00348</Part-ID>
      </Parts>
    <p:getPartsListResponse>
  </soap:Body>
</soap:Envelope>
```

- ✓ **Absence de lien vers les pièces retournées**
  - *A noter que les infos retournées pourraient pointer vers un service REST-ful*

REST vs SOAP ou XML-RPC

- ✓ SOAP et XML-RPC
  - Se basent sur des méthodes (appels de méthodes à distance)
- ✓ REST
  - Se base sur les ressources existantes et l'échange de leur valeur via une URL qui la nomme
- ✓ De nombreux autres points de comparaison:
  - Serveurs Proxy (Web intermédiaires)
  - Etat de transition dans une application cliente
  - Cache (i.e., performance)
  - Evolution du Web (Web sémantique)
  - Interface générique (versus interface custom)
  - Interopérabilité
  - ...

UDDI

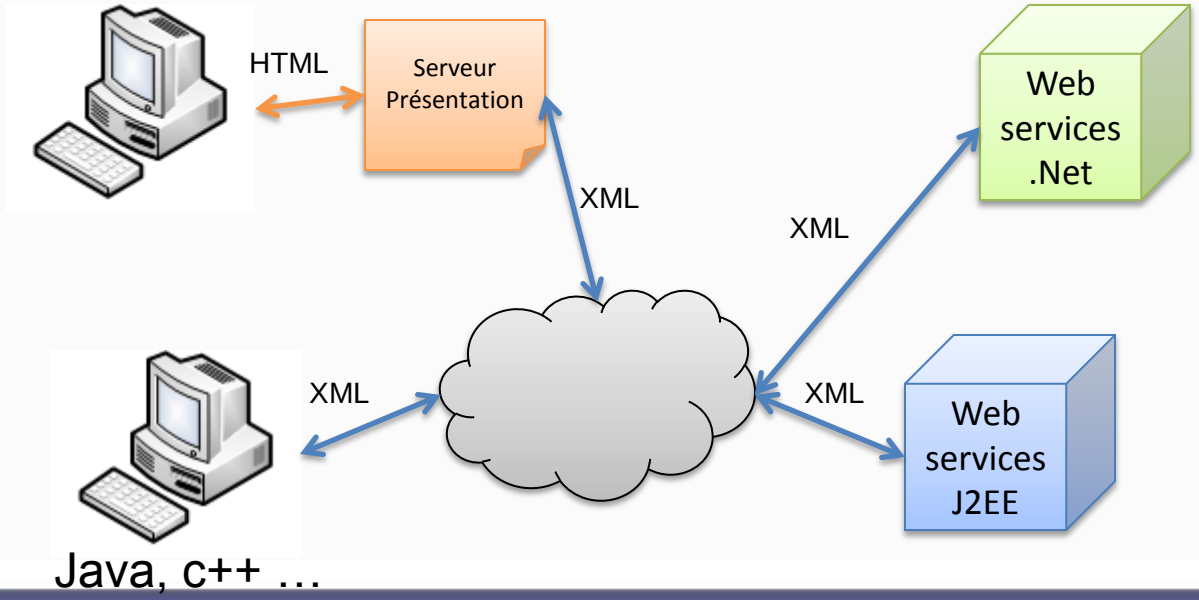
(Universal Description Discovery Integration)

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

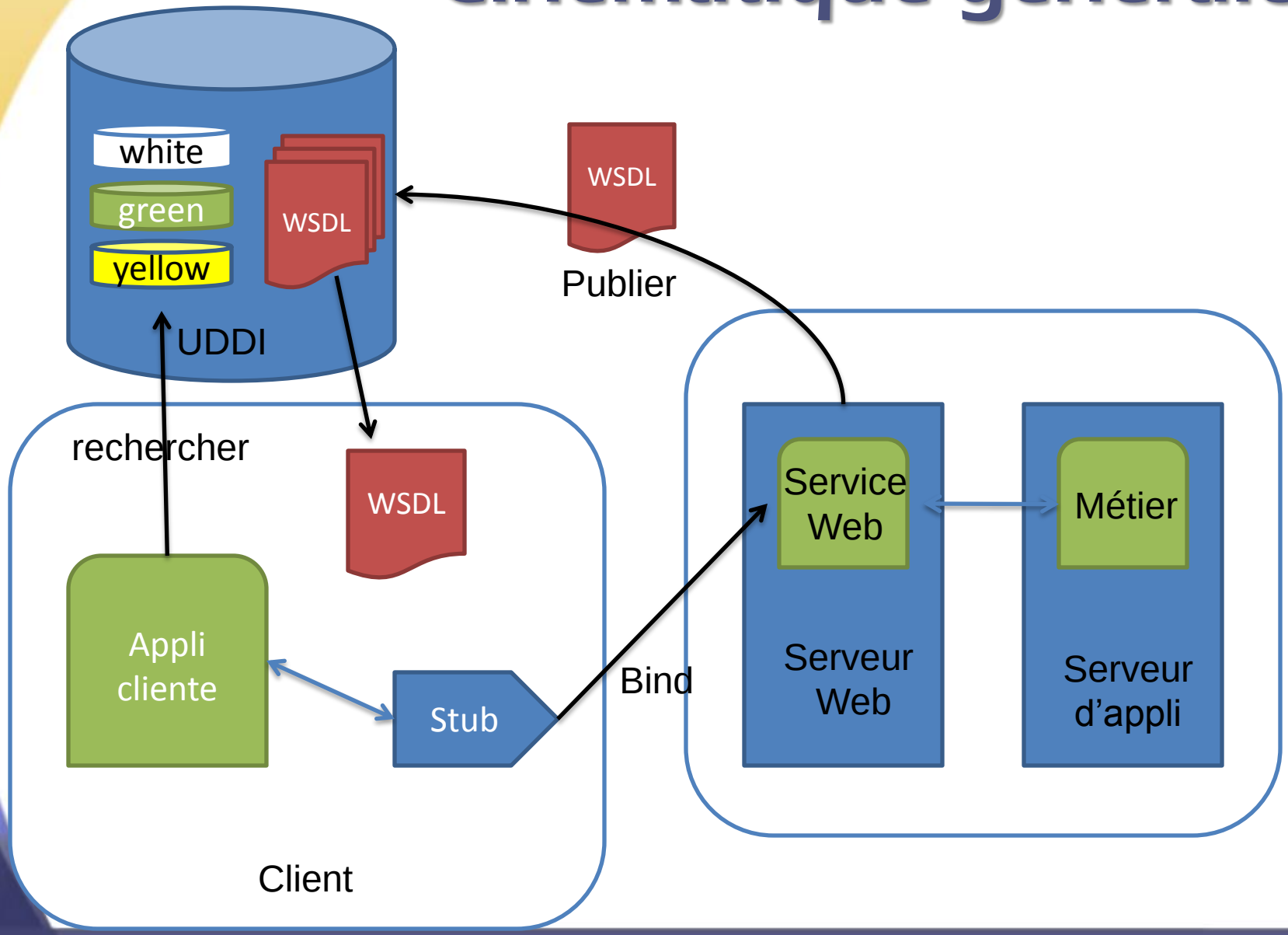
*Image : http://www.webmeimfamous.com/wp-content/uploads/2010/09/annuaire.jpg

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

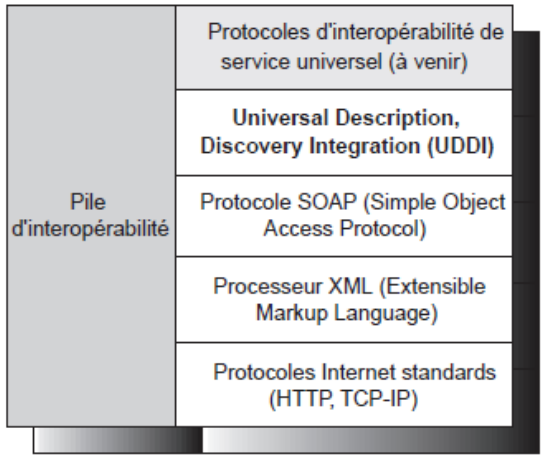


Pourquoi UDDI ?

- ✓ SOAP/WSDL permettent d'invoquer des services Web mais pas de les rechercher
- ✓ UDDI ne répond pas aux mêmes besoins que les LDAP qui références 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

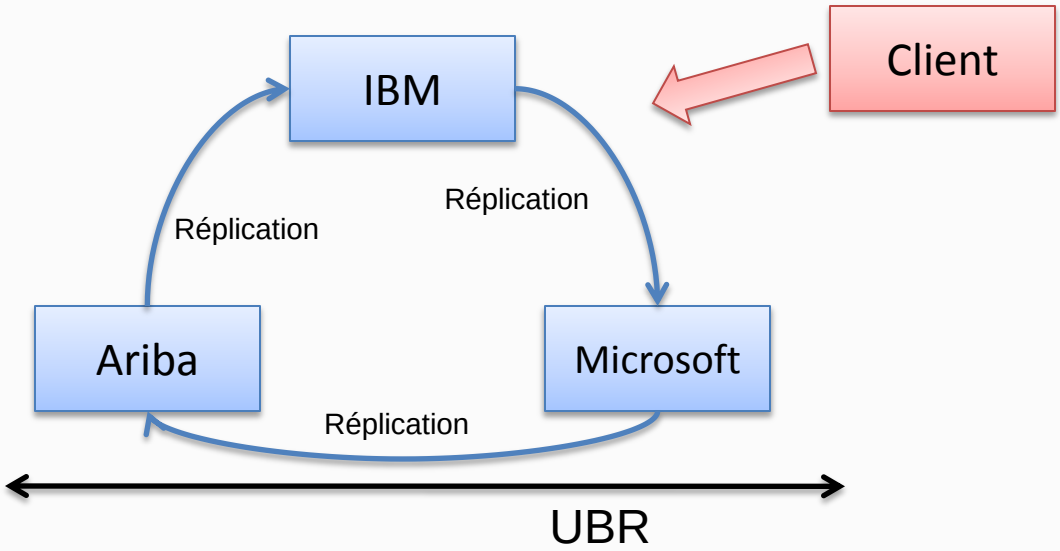
Introduction

- ✓ 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



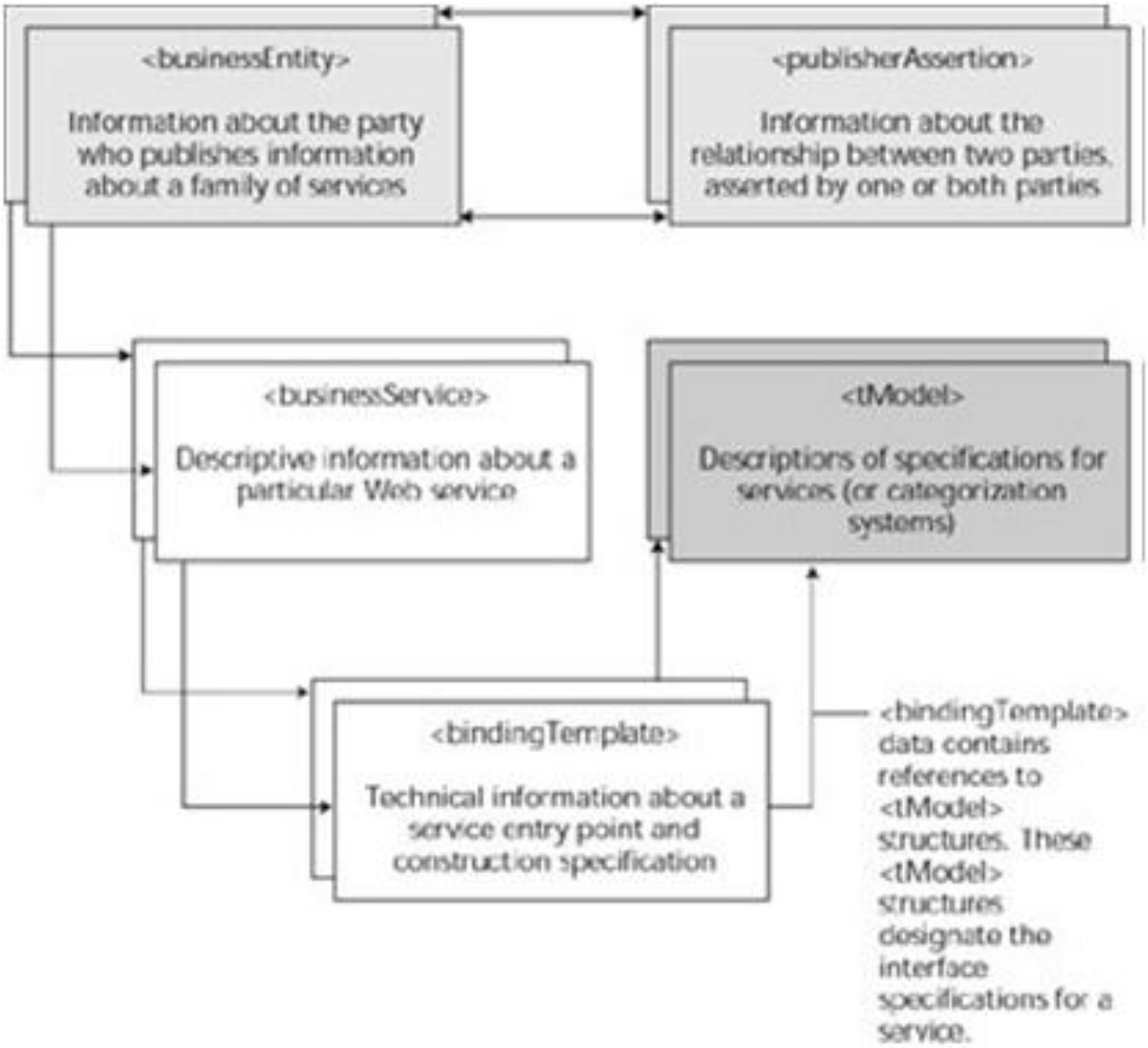
UDDI architecture

- ✓ 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


<businessEntity>
Information about the party who publishes information about a family of services
<publisherAssertion>
Information about the relationship between two parties, asserted by one or both parties
<businessService>
Descriptive information about a particular Web service
<tModel>
Descriptions of specifications for services (or categorization systems)
<bindingTemplate>
Technical information about a service entry point and construction specification
<bindingTemplate> data contains references to <tModel> structures. These <tModel> structures designate the interface specifications for a service.

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="busine                                    sEntity" />
 <complexType name="b
  <sequence>
   <element ref="uddi:d                                    " />
   <element ref="uddi:n                                    ed" />
   <element ref="uddi:d                                    axOccurs="unbounded" />
   <element ref="uddi:c
   <element ref="uddi:b                                    "0" />
   <element ref="uddi:i                                    />
   <element ref="uddi:c                                    />
  </sequence>
 <attribute name="busin                                    Key" use="required" />
  <attribute name="oper                                    onal" />
 <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="busin                                    isinessService" />
 <complexType name=
  <sequence>
   <element ref="uddi                                   axOccurs="unbounded" />
   <element ref="uddi                                   "0" maxOccurs="unbounded"
   />
   <element ref="uddi                                   ccurs="0" />
   <element ref="uddi                                   ="0" />
  </sequence>
 <attribute name="ser                                   ceKey" use="required" />
 <attribute name="bus                                   inessKey" 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                                  s="0" maxOccurs="unbounded"
   />
   <choice>
    <element re
    <element re
   </choice>
   <element ref                                  " />
  </sequence>
 <attribute nam                                 rviceKey" use="optional" />
 <attribute name="bindingKey" type="uddi:bindingKey" use="required" />
</complexType>
```

BindingTemplate

- uddi:accessPoint
- uddi:hostingRedirector
- uddi:tModelInstanceDetails → tModel

- serviceKey
- businessKey

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

- ✓ Traditionnellement pas besoin d'identification, accès anonyme à la recherche
- ✓ 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 ...)
- ✓ Les structures de données UDDI sont identifiées par des clés uniques, générées automatiquement par l'annuaire, lors de la première sauvegarde de ces structures.
- ✓ Ces clés sont constituées par des « identifiants universels uniques » (UUID ou Universally Unique Identifier)

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

UDDI : 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);}
```

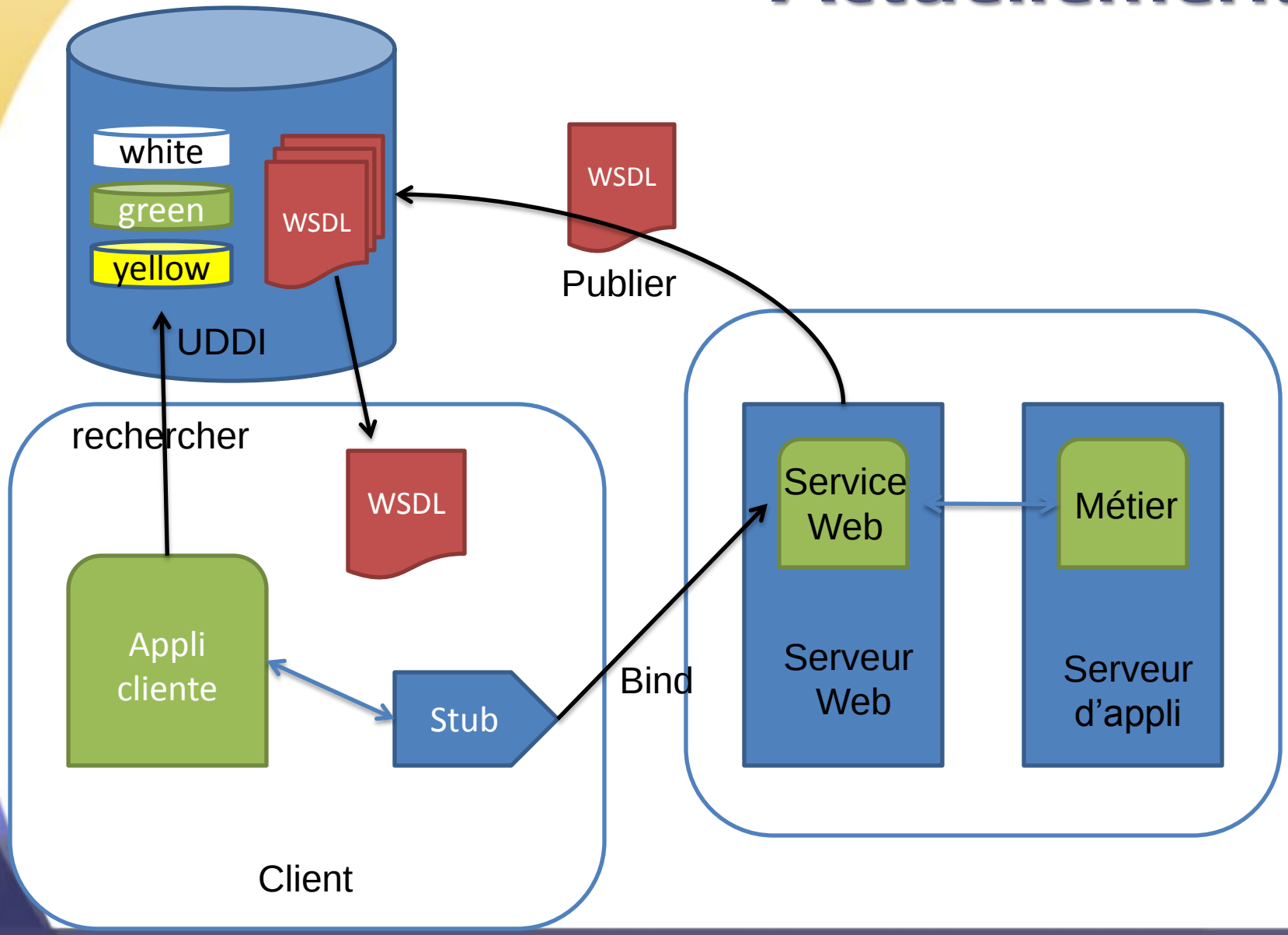
Deleting a business

```
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);}
```

Conclusion

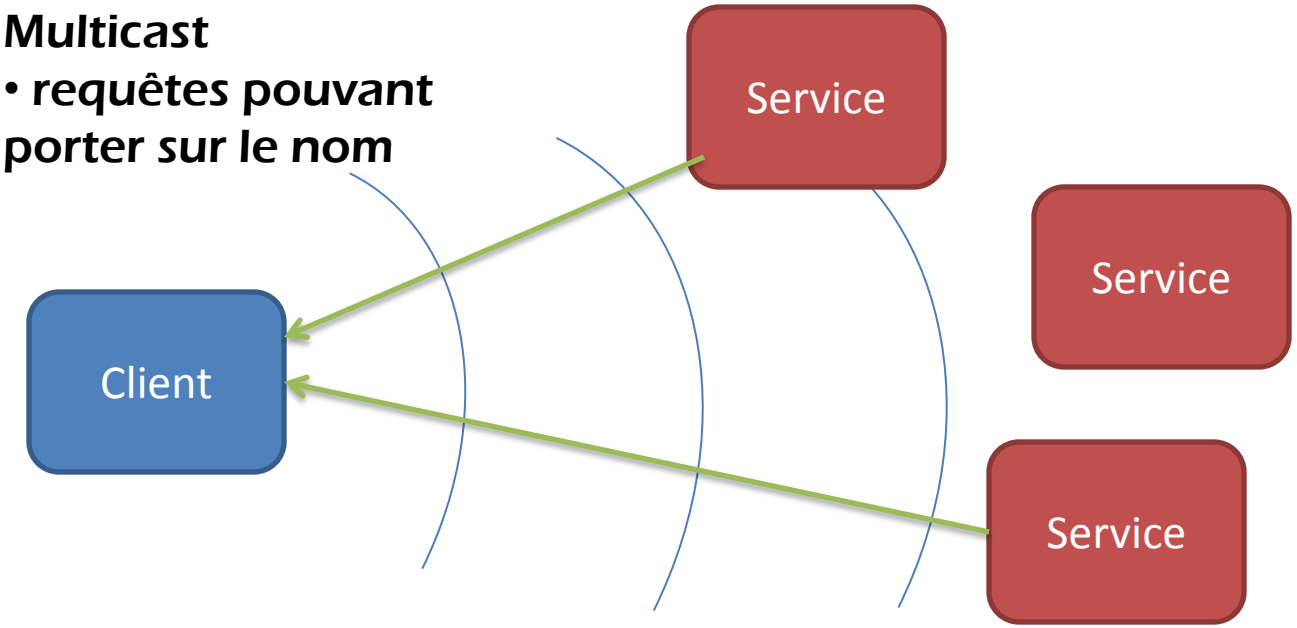
UDDI

Actuellement



On s'oriente de plus en plus vers

Multicast
- **requêtes pouvant porter sur le nom**



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

Conclusion Générale

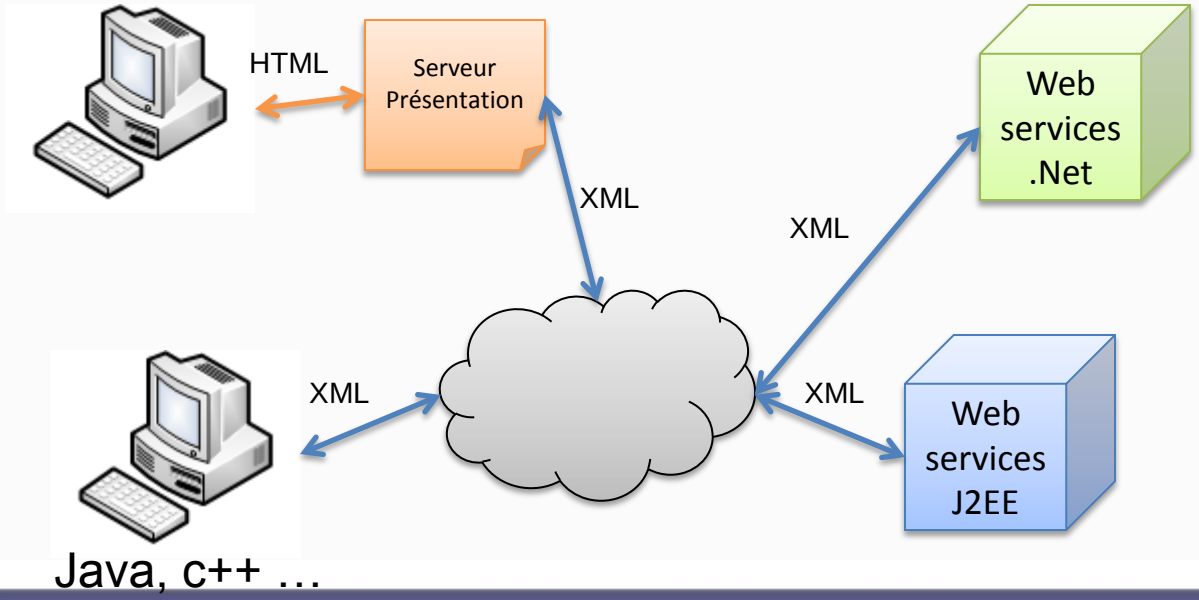
Web services

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