

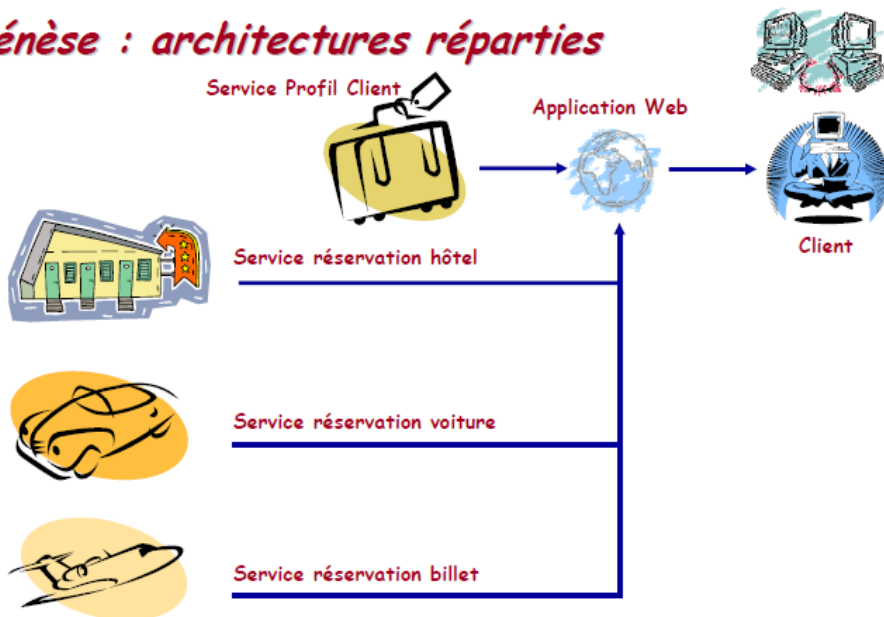


1



2

## Génèse : architectures réparties

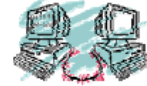


## Génèse : architectures réparties



- Sur le schéma précédent, l'application consultée par le client met en œuvre des applications réparties pour satisfaire sa demande
- Dans cette mise en œuvre applicative, on parle :
  - ▶ De transformation, lorsqu'il s'agit d'adapter le dialogue en fonction du profil utilisateur
  - ▶ D'agrégation, lorsqu'il s'agit de faire appel à des applications proposées par des partenaires ou fournisseurs
- Que ce soit de l'agrégation ou de la transformation d'informations, les applications réalisant ces tâches de façon complètement automatique et opaque s'appellent des « **services web** ».

## *Génèse : architectures réparties*



- Une application web communicante résulte alors d'un assemblage de services web.
- Certains sont internes et d'autres externes et fournis par divers partenaires et fournisseurs
- Les deux extrémités de cet assemblage sont :
  - ▶ Le tout interne : entièrement composé de services internes, on parle alors d'intégration d'applications d'entreprise ou EAI
  - ▶ Le tout externe : entièrement composé de services externes, on parle alors de portail d'entreprises.

## *Historique*



- Mais qu'est-ce qui différencie un service web d'une application distante traditionnelle ?
- L'histoire des architectures réparties est ponctuée d'évolutions liées aux :
  - ▶ protocoles d'échanges et d'accès (Corba, DCOM, RMI)
  - ▶ langages d'implémentation (C++, Java, Smalltalk, etc.)
  - ▶ interfaces (d'interaction et de présentation)
- Ces évolutions sont toujours plus riches, plus sophistiquées, et souvent plus complexes
- Elles imposent de fait, des configurations clientes suffisamment robustes aptes à recevoir de telles architectures

## Historique



- La question essentielle est de savoir si l'on peut combiner :
  - ▶ Les caractéristiques des architectures distribuées comme corba ou rmi.
    - Ce type d'architecture est devenue indispensable pour répondre aux besoins en terme complexité des applications d'entreprise
  - ▶ Les contraintes imposées par le web, à savoir :
    - client léger et souvent réduit à un simple navigateur
    - mise en œuvre du protocole HTTP



*Les services web permettent d'atteindre cet objectif*

## Principes de base



- Le modèle des services web repose sur le transport d'une demande de service entre un client et un serveur.
- Le transport est assuré par la suite de protocoles « internet/web » constituée de « TCP/IP et HTTP »



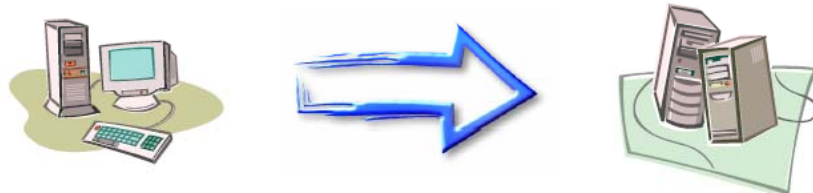
## Principes de base

- HTTP ne sait transporter que du texte (les pages HTML)
- Les échanges (requêtes et réponses) entre le client et le serveur sont donc au format texte
- Le format texte (représentation de base portable sur toute plateforme, au codage de caractères près) de représentation des informations est XML
- Les messages transportés sont donc au format XML



## Principes de base

- Les services web matérialise deux perceptions dans l'évolution des architectures.
  - *Passage du web client au web machines*
  - Adaptation des architectures distribuées au monde Web



Web Utilisateur caractérisé par :  
Navigateur HTML  
Contenu dynamique sur le serveur

Web machine caractérisé par :  
Communication directe  
entre applications

## *Principes de base*



- ▶ Les services web matérialise deux perceptions dans l'évolution des architectures.
  - ▶ Passage du web client au web machines
  - ▶ *Adaptation des architectures distribuées au monde Web*



Architectures distribuées, à savoir :  
RPC, RMI  
Corba, DCOM, JRMP

Nouveau contexte, à savoir :  
Web (HTTP)  
XML

*Les services web, c'est quoi ?*

*Les services web, pourquoi*

*Les usages*

*Les acteurs (rôles)*

*Le scénario complet*

*Les technologies*

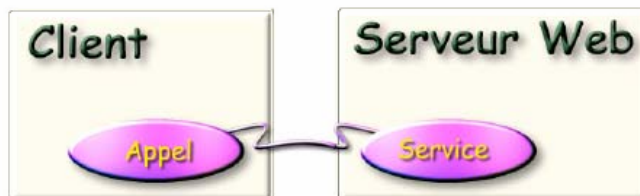
## Définition service web

Un "Service Web" est une application logicielle à laquelle on peut accéder à distance à partir de différents langages basés sur XML.

Un "Service Web" est identifié par une URL, comme n'importe quel site Web. Il s'exécute sur un "Serveur d'Applications". Peu importent l'ordinateur, le système d'exploitation ou le langage utilisés par le Client ! Une application peut ainsi utiliser plusieurs "Services Web" s'exécutant sur des serveurs distants.

## *Les Services Web, c'est quoi ?*

- Avant toute autre chose, la technologie des services web affiche les mêmes intentions que les architectures les plus anciennes en terme d'accès distant
- C'est la possibilité d'invoquer une fonction distante. En l'occurrence, sur un serveur web distant puisque le protocole de base est HTTP
- On dispose d'une infrastructure souple basée sur XML pour les systèmes distribués hétérogènes



## *Les Services Web, c'est quoi ?*

- Ils sont accessibles via le web par des protocoles bien connus
- Ils sont décrits à partir de XML
- Ils interagissent via XML
- Ils sont localisables à partir de registres
- Ils sont entièrement transversaux aux plates-formes et très faiblement couplés

15

## *Les Services Web, c'est quoi ?*

- Ils introduisent un nouveau modèle de développement basé sur ce que l'on appelle les architectures orientées services
- Une architecture orientée services se focalise sur une décomposition plus abstraite dans la résolution des problèmes. On parle de résolution dirigée par les services.
- Un service résout un problème donné
- Les services peuvent être combinés pour résoudre des problèmes de plus en plus complexes

16



## *Les Services Web, c'est quoi ?*

- L'avantage essentiel des services web concerne le fait que le client consommateur n'a pas besoin de connaître l'identité du fournisseur du service
- Le client doit simplement exprimer son besoin
- Face à un besoin, plusieurs fournisseurs de services peuvent exister
- Chacun ayant des caractéristiques de coût, de performance, de fiabilité, etc.
- Le client choisit le fournisseur (i.e. le service) correspondant le mieux à ses besoins.

## **Architecture : le modèle SOA**

L'objet des Services Web est la communication d'application à application (A2A) sur Internet.

Le but est de faciliter l'intégration des applications d'entreprise (EAI) et le e-commerce spécialement en "Business To Business" (B2B). Pour ce faire,

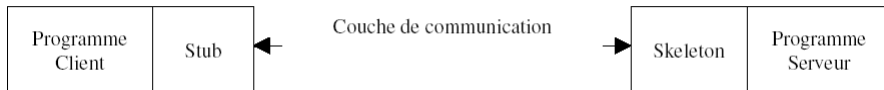
l'architecture des Services Web doit supporter les transactions asynchrones ("mode message") aussi bien que les transactions synchrones ("Remote Procedure Call" ou RPC),

- le codage des données est effectué en XML, ce qui conduit à des documents que l'homme et la machine peuvent interpréter, facilitant ainsi le débogage,
- des données arbitraires peuvent être représentées et validées au niveau système,
- les systèmes doivent être inter-opérables.

## Comparaison avec d'autres technologies

- Exécuter des procédures à distance n'est pas une révolution. Plusieurs autres technologies le permettent. Ces technologies reposent sur deux types d'architecture :

- **Architecture "Stub/ Skeleton"**



- Cette architecture a pour but de rendre "simple" la partie "Client". Le "Stub" est une portion de code installée sur la machine client, qui identifie les objets et les méthodes disponibles sur le serveur, code les appels vers celui-ci dans un format que le serveur comprend et décode les réponses du serveur dans un format accessible au client. Le "Skeleton" joue un rôle symétrique sur le serveur.

19

## Comparaison avec d'autres technologies

La connexion entre le client et le serveur est une connexion permanente pendant toute la durée de la session. Ceci limite le nombre de clients qu'un serveur peut traiter. Les appels des méthodes s'effectuent en "mode synchrone" et ignorent le "mode message" (MQSeries d'IBM ou Java Message Service).

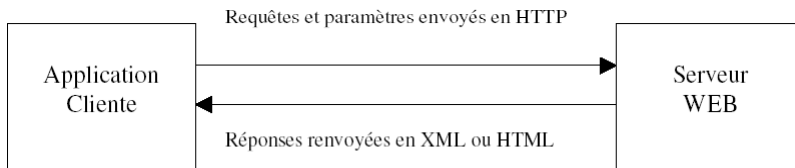
Les données sont codées en binaire, créant ainsi des problèmes de compatibilité inter plates-formes. Les différences entre les solutions basées sur cette technologie proviennent de la façon dont les données sont codées, ainsi que des transports utilisés.

Les solutions les plus connues basées sur cette technologie sont CORBA, RMI et DCOM.

20

## Comparaison avec d'autres technologies

### Architecture basée sur des transactions HTTP



Cette architecture utilise un serveur Web, tel que IIS (Microsoft) ou Apache. Les clients communiquent avec le serveur en utilisant HTTP ou HTTPS. La connexion est rompue à la fin de la transaction (et pas de la session), ce qui permet de traiter un plus grand nombre de clients au détriment des performances si un client effectue un grand nombre de transactions.

Cette architecture présente de plus l'inconvénient de ne pas avoir de répertoire référençant les services offerts. Ainsi, une collaboration étroite est nécessaire entre les clients et les fournisseurs de services.

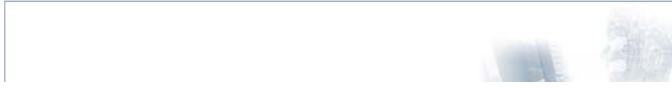
21

## Comparaison avec d'autres technologies

### Architecture basée sur des transactions HTTP

- Les solutions les plus connues basées sur cette technologie sont CGI, Servlets / JSP, ASP et PHP.
- On a de nombreuses limitations, à savoir :
  - types des données pauvres,
  - pas de format accepté pour échanger des données,
  - pas d'API bien documentée pour décrire les services,
  - pas de répertoire pour trouver les services.

22



## *Les usages*

- Les services web pour représenter des applications sophistiquées bien délimitées et sans forte interactivité.
  - ▶ Par exemple, une application qui donne les conditions du temps peut être idéalement représentée par un service.
- Les services web sont adaptés pour l'assemblage de composants faiblement couplés.
  - ▶ Ils sont définis de façon indépendante, mais peuvent interagir.
- Les services web sont adaptés à la représentation d'applications orientées messages.
  - ▶ Les mécanismes d'invocation asynchrone des applications orientées messages sont en font de bonnes candidates aux services web



## *Les acteurs (rôles)*

- Les principaux acteurs dans la technologie des services web sont :
  - ▶ Le client : celui qui utilise, invoque le service web
  - ▶ Le fournisseur : celui qui fournit le service web
  - ▶ L'annuaire : celui qui détient les informations du service web

## Les acteurs (rôles)

- Le client et le fournisseur sont les éléments principaux dans l'architecture des services web
- Un fournisseur est représenté par un serveur d'application (J2EE par exemple)
- Le fournisseur détient un ou plusieurs services qui sont représentés par des EJBs ou des servlets et qui sont enveloppés d'une couche « service »
- Le fournisseur peut être le client d'un autre fournisseur (interopérabilité)
- Une fois le service défini, il peut être déclaré dans un annuaire, on parle alors de publication du service afin de le rendre accessible aux clients

## Service web

De nombreuses normes sont utilisés dans cette architecture : "SOAP" pour l'échange de messages, "XML" langage de base pour décrire tous les documents sur lesquels les messages sont construits, "HTTP" pour transporter les messages, "WSDL" pour décrire les services et enfin "UDDI" pour les publier.

L'approche "Services Web" constitue un changement fondamental dans la manière de concevoir et réaliser les applications informatiques et de programmer les ordinateurs.

## *Le scenario complet*

- **Etape 1 : définition, description du service**
  - ▶ On doit décrire d'un point de vue informatique ce que fait le service, la solution qu'il propose, ...
  - ▶ La définition est faite au WSDL au sein du fournisseur de services (i.e. le serveur d'applications)
  
- **Etape 2 : publication du service**
  - ▶ Une fois le service défini et décrit en termes de mise en oeuvre, il peut être déclaré dans un annuaire, on parle alors de publication du service afin de le rendre accessible aux clients.
  - ▶ Comme on le verra, la publication sera effectuée au sein d'un annuaire dédié UDDI.
  
- **Etape 3 : recherche du service**
  - ▶ Le client se connecte, sur un annuaire (UDDI) pour effectuer une recherche de service.

## *Le scenario complet*

- **Etape 4 : enregistrement au service web**
  - ▶ Une fois le service trouvé par le client, ce dernier doit s'enregistrer auprès du fournisseur associé au service. Cet enregistrement indique au fournisseur l'intention du client d'utiliser le service suivant les conditions décrites dans la publication.
  
- **Etape 5 : mise en oeuvre du service**
  - ▶ Le client peut invoquer le service suivant les conditions inscrites au sein de l'annuaire lors de la publication du service web (étape 2)
  
- **Etape 6 : composition**
  - ▶ C'est la possibilité de combiner plusieurs services. En fait, un service peut devenir le client d'un autre service.

# Architecture : le modèle SOA

Le "Service Registry" est une application qui retourne au Requêteur (client) les informations permettant de trouver un service à partir de critères de recherche, et de s'y connecter.

Pour obtenir ce résultat, le fournisseur de service doit avoir publié ces informations préalablement.

29

## Le scénario complet

### Application cliente

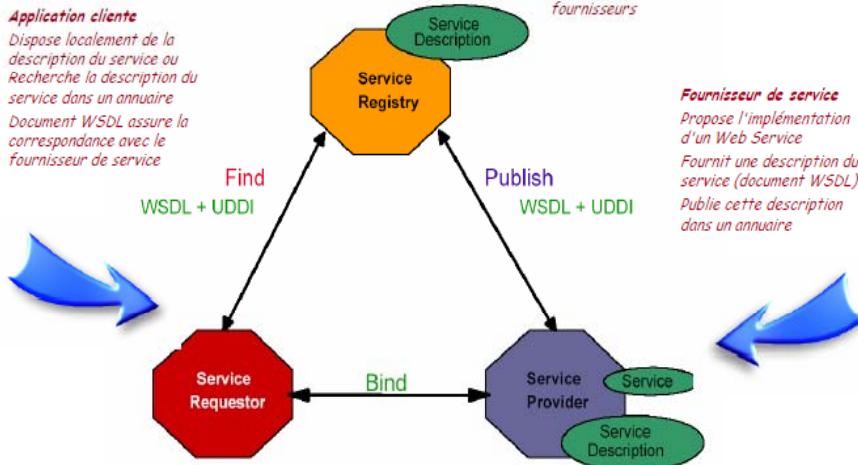
Dispose localement de la description du service ou Recherche la description du service dans un annuaire  
Document WSDL assure la correspondance avec le fournisseur de service

### Annuaire de publication

Dispose de la description des services ainsi que de leur localisation, c'est-à-dire des différents fournisseurs

### Fournisseur de service

Propose l'implémentation d'un Web Service  
Fournit une description du service (document WSDL)  
Publie cette description dans un annuaire



30

## Qui a conçu les Services Web?

Des organisations telles que OASIS et W3C dirigent les spécifications.

- **W3C**: (world wide web) responsable du développement de plusieurs standards (XML, WSDL, SOAP).
- **OASIS**: responsable du développement de standards comme UDDI, WS-Security et plus récemment WSBPEL.
- L'organisation **WS-I** est un type d'organisation différente. Elle prépare des outils et des directives pour aider les développeurs à créer les logiciels en conformité avec les standards des Services Web (WS-I Basic Profile )
- ... IBM, Microsoft, Ariba, DevelopMentor, BEA Systems, Hewlett-Packard, Sun Microsystems, SAP, Canon, Xerox, Oracle et Intel

## Les technologies

Pile de protocole

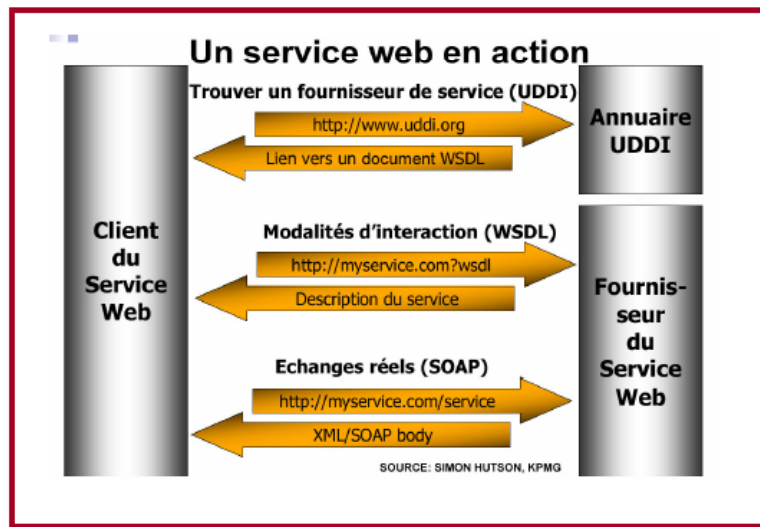
Document XML décrivant le service afin de rendre la solution des Web Services générique	<b>WSDL</b>
Protocole basé sur le standard XML pour l'échange de données structurées entre des applications réseaux	<b>SOAP</b>
Couche réseau (HyperText Transfer Protocol)	<b>HTTP</b>



## Les technologies

- Les éléments techniques utilisés sont différents puisque imposés par le Web et XML
- L'architecture des Web Services repose essentiellement sur les technologies suivantes :
  - ▶ **SOAP - Simple Object Access Protocol**  
Protocole pour la communication entre Web Services  
IIOP pour Corba ou RMI-IIOP pour les EJBs
  - ▶ **WSDL - Web Service Description Language**  
Langage de description de l'interface du Web Service  
IDL pour Corba ou Interface Java pour les EJBs
  - ▶ **UDDI - Universal Description, Discovery and Integration**  
Annuaire pour le référencement du Web Service  
CosNaming pour Corba ou JNDI pour les EJBs

## Le scenario complet





## *Le protocole SOAP*

### ➤ Rôle

- ▶ Assure les appels de procédures à distance au dessus d'un protocole de transport

### ➤ Fonctionnement côté Client

- ▶ Ouverture d'une connexion HTTP
- ▶ Requête SOAP est un document XML décrivant :
  - ▶ une méthode à invoquer sur une machine distante
  - ▶ les paramètres de la méthode

### ➤ Fonctionnement côté Serveur SOAP

- ▶ Récupère la requête
- ▶ Exécute la méthode avec les paramètres
- ▶ Renvoie une réponse SOAP (document XML) au client

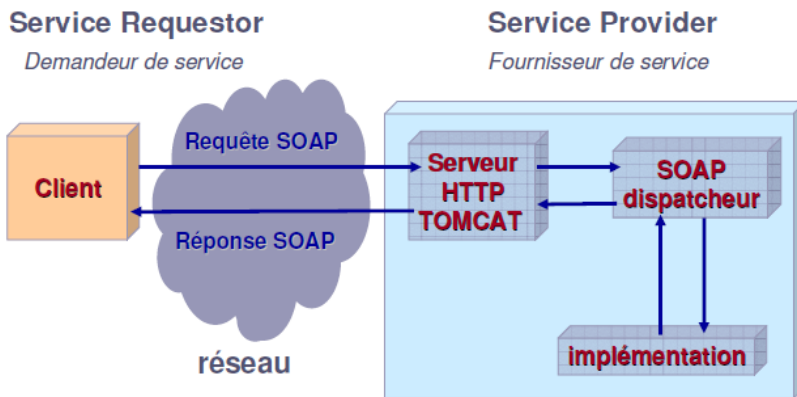


## SOAP (Simple Object Access Protocol)

SOAP est un format de messages qui permet de transmettre en XML les appels de procédures distantes. SOAP permet aussi d'envoyer en XML un document entier d'un ordinateur à un autre.

SOAP définit une enveloppe contenant un "en-tête" et un "corps". L'en-tête fournit les instructions indiquant comment traiter le message. Le corps contient l'appel de la procédure distante dans un sens et la réponse du serveur dans l'autre. L'envoi de documents attachés à un message SOAP se fait en encapsulant le message

## Le protocole SOAP



37

## Les technologies

- Afin d'être découvert, un service doit être publié. Au dessus de ces trois couches de base viennent se greffer deux couches UDDI :

Découverte de services	UDDI
Publication de services	UDDI

- On publie notre service via le document WSDL après de notre annuaire UDDI.
- Une application cliente peut découvrir et accéder à notre service lors de son exécution via un annuaire UDDI.

## Le langage WSDL

- Une interface qui cache le détail de l'implémentation du service, permettant une utilisation indépendante :
  - ▶ de la plate-forme utilisée
  - ▶ du langage utilisé
- Le fichier WSDL est au format XML et regroupe toutes les informations nécessaires pour interagir avec le Web Service :
  - ▶ les méthodes
  - ▶ les paramètres et valeurs retournées
  - ▶ le protocole de transport utilisé
  - ▶ la localisation du service

*Documents WSDL, générés par les outils de développement favorisent une intégration rapide du service*

## Le langage WSDL

- 2 types de documents WSDL :
  - ▶ le document WSDL décrivant l'interface du service
  - ▶ le document WSDL décrivant l'implémentation du service
- Documents indispensables au déploiement de Web Services
- Publication et recherche de services au sein de l'annuaire UDDI se font via ces 2 types de document WSDL

*Pour l'accès à un service particulier, un client se voit retourné l'URL du fichier WSDL décrivant l'implémentation du service. Seul l'emplacement de ce fichier WSDL est indiqué puisque ce dernier référence l'autre document WSDL décrivant l'interface de mise en œuvre du service.*

## WSDL (Web Services Description Language)

Cette spécification du W3C précise comment décrire un Service Web :

- méthodes acceptées et paramètres correspondants,
- réponses fournies en retour,
- protocoles et formats de données pouvant être traités,
- URLs du service.

Un document WSDL est un document XML contenant toutes les informations nécessaires pour contacter et utiliser un service, indépendamment de toute plate-forme ou langage de programmation.

41

## WSDL (Web Services Description Language)

L'utilisateur potentiel d'un Service Web doit d'abord obtenir le WSDL de ce service. Il créera ensuite (sur la plate-forme et dans le langage de son choix) le logiciel "Client" correspondant à son application, utilisant les "méthodes" que propose(nt) le (ou les) Service(s) Web, comme s'il s'agissait de "méthodes locales à son ordinateur". Plusieurs outils logiciels existent, permettant de traiter la communication avec le serveur, en utilisant les composants décrits précédemment.

42



## *Annuaire UDDI*

### ► Rôle

- ▶ Spécification pour la définition d'un service de registre
  
- ▶ Fournisseur
  - Déclaration du fournisseur
  - Enregistrement de ses Web Services disponibles
  
- ▶ Client
  - Requête de recherche de Web Services (SOAP)
  - Mise en relation avec le Web Service d'un fournisseur

43

## UDDI API (Universal Description, Discovery and Integration)

La spécification UDDI décrit un type de registre listant les Services Web disponibles. Différentes informations permettent de faciliter la recherche d'un service particulier. Les services enregistrés peuvent appartenir à l'une des trois catégories suivantes :

- "Public" : service ouvert à tous. Plusieurs éditeurs majeurs proposent de tels services, par exemple IBM et Microsoft.
- "Private" : service accessible seulement à l'intérieur d'une compagnie. Pour celle-ci, le catalogue UDDI permet la réutilisation de logiciels en interne.
- "Restricted" : service accessible seulement par certaines organisations autorisées (par exemple les fournisseurs et certains clients d'une entreprise).

44

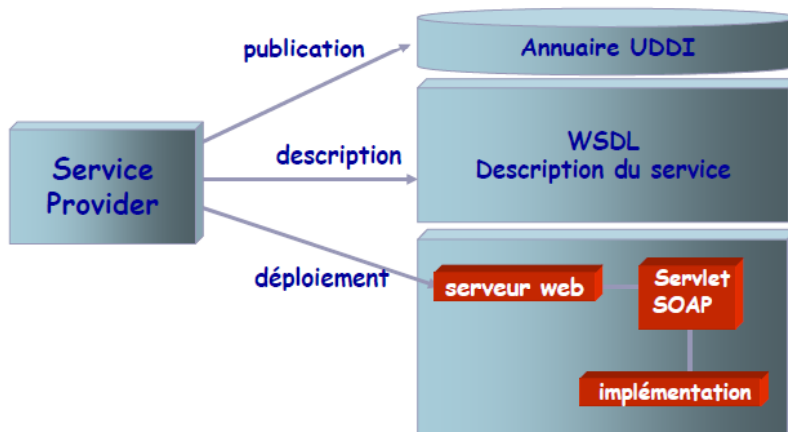
## UDDI (Universal Description, Discovery and Integration)

Un fournisseur, ayant créé un Service Web "public", doit l'enregistrer auprès du "Service Registry" de son choix, par exemple le "Microsoft public registry". Ce "Registry" réplique chaque nuit ses entrées sur les autres "public registry" existants. Ainsi, un client potentiel, interrogeant quelques jours après le "IBM public registry", trouvera ce nouveau service Web et pourra l'utiliser s'il le souhaite.

La spécification UDDI part d'une initiative lancée par ARIBA, Microsoft et IBM. Cette spécification n'est pas gérée par le W3C mais par le groupe appelé OASIS (Organization for the Advancement of Structured Information Standards).

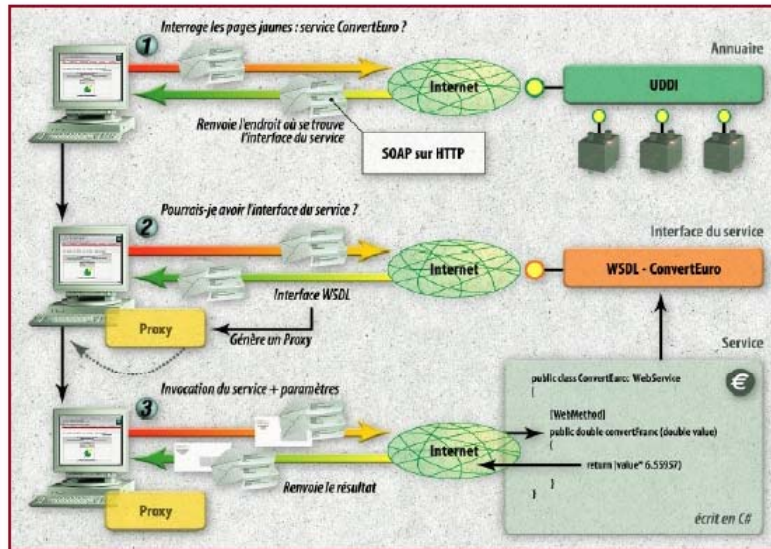
45

### *Le fournisseur de services*

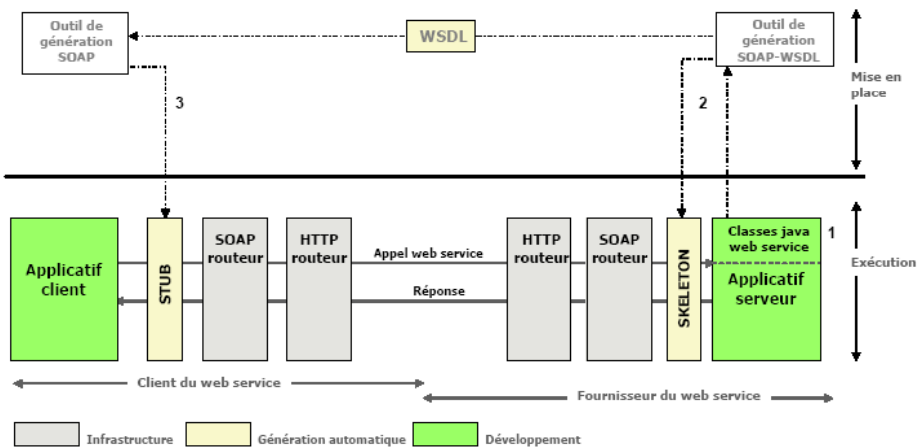


46

## Schéma fonctionnel



## Mise en place d'un service web



L'équipe cliente ne voit jamais le code source ni l'exécutable du Web service appelé. Elle connaît seulement le fichier WSDL, y compris pour la gestion des erreurs qui transitent de manière standardisée dans les messages SOAP<sup>17</sup>.



## Outils de création de services web



Différents éditeurs proposent des outils pour créer des Services Web. Citons en quelques uns :

- Apache AXIS (Apache eXtensible Interaction System) : outil "Open Source", pouvant être téléchargé librement,
- Java Web Services Developer Pack, fourni par SUN,
- Microsoft Plate-forme .NET (par exemple Visual Studio 2005 ou 2008),
- BEA Plate-forme WebLogic,
- IBM WebSphere,
- SOAP::Lite, plate-forme utilisant le langage PERL.....