Université de Toulouse

Institut de Recherche en Informatique de Toulouse

Les Services Web et SOAP

IUT A Paul Sabatier
Semestre 4

Julien Broisin
mailto: broisin@irit.fr

Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

Contexte

- Evolution technologique
  - Banalisation des réseaux de télécommunication
  - Performance des réseaux (débit et fiabilité)
  - Rapport coût/ performance des équipements
  - Convergence informatique et téléphonie
- Evolution des besoins
  - Structure des entreprises (communication et partage)
  - Accès universel à l'information (web)
  - Informatique distribuée « grand public » (TV, e-commerce)

Systèmes répartis

Caractéristiques des systèmes répartis

- Liaison
- Hétérogénéité
- Fiabilité
- Sécurité
- Mise à l'échelle
- Persistance
- Intégration
- Performance
- Administration

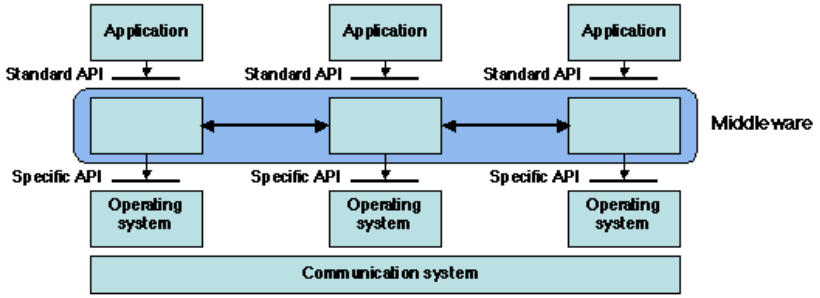
Difficile à concevoir, à programmer, à déployer et à administrer

Besoin de modèles et d'outils adaptés

Outils de programmation

- Outil de base : les sockets
- Middleware : couche logicielle répartie destinée à
  - Faciliter la programmation réseau
  - Masquer la répartition des traitements/données
  - Masquer l'hétérogénéité des machines/systèmes
  - Faciliter l'interopérabilité des applications
- Fournir des services pour solutionner les problèmes !!

Localisation du Middleware

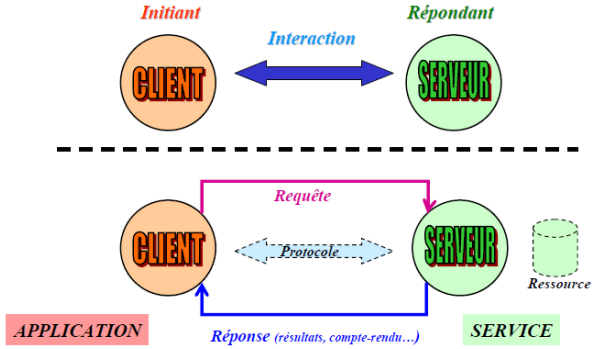


Différents types de Middleware

- Base de données
  - Hétérogénéité / répartition des données
- Object Request Broker
  - Avantages de la communication par objet
  - Cache le détail d'implémentation des objets
- Modèle à messages
  - Publish/Subscribe
  - Vers plus d'asynchronisme
- Modèle Client / Serveur

Le modèle Client/Serveur

- Modularité, transparence, répartition
- Architecture orientée services



Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

Architecture orientée service (1/2)

- Service Oriented Architecture (SOA)
- Une architecture
  - description formelle d'un système qui définit ses objectifs, fonctions, propriétés et interfaces
  - inclut aussi la description des composants internes du système et leurs interactions
- Un service
  - composant logiciel accessible à travers un réseau qui fournit une fonctionnalité à un demandeur de service

Architecture orientée service (2/2)

- Fait référence à la mise en œuvre de systèmes répartis qui délivrent des fonctionnalités à travers des services
  - Qui peuvent être sous le contrôle de différents domaines
- Focalise sur l'indépendance des services qui interagissent
- Désigne l'architecture d'un système, pas son implémentation

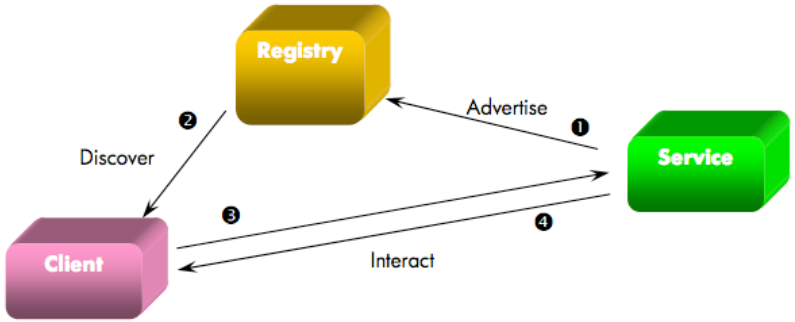
Caractéristiques des services (1/2)

- Ils peuvent être utilisés individuellement, ou intégrés (composés) afin de fournir des services de plus haut niveau (réutilisation de fonctionnalités existantes)
- Ils communiquent avec les Clients à travers l'échange de messages : ils sont définis par les messages qu'ils acceptent et qu'ils retournent

Caractéristiques des services (2/2)

- Ils peuvent participer à un *workflow* où l'ordre des messages envoyés et reçus affecte le résultat des opérations exécutées par un service (chorégraphie de services)
- Ils décrivent leurs interfaces, fonctionnalités, politiques ou protocoles de communication
- Les détails d'implémentation sont invisibles

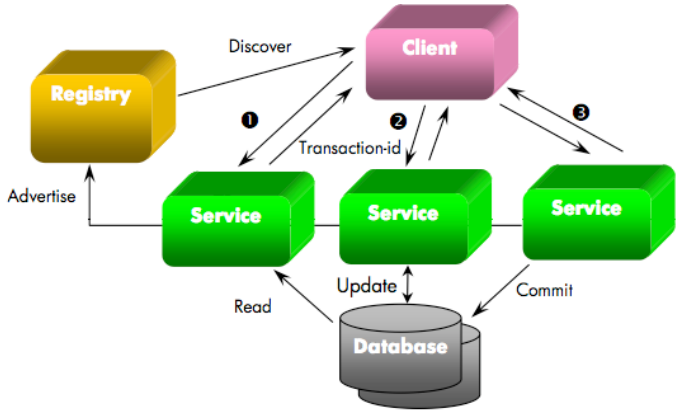
Interaction simple dans une SOA



Service avec ou sans état

- Service sans état (stateless) : l'interaction se termine lorsque le Client a reçu la réponse
- Service avec état (stateful) : le service stocke des informations relatives à une étape et nécessaires à la réalisation de l'étape suivante pour finaliser l'interaction
  - Le Client doit s'adresser au même service pour l'étape suivante
  - Implique des délais, parfois même des échecs

Services sans état : illustration



Conception d'un service

- Privilégier les services sans état
  - Après chaque étape intermédiaire, le service doit fournir au Client les informations nécessaires et suffisantes à la réalisation de l'étape suivante par un autre service qualifié pour poursuivre la transaction
  - Le Client doit fournir ces informations à n'importe quel service afin de traiter l'étape suivante
  - Le service doit accepter et traiter les informations fournies par le Client, qu'il ait exécuté ou non l'étape précédente

Interaction complexe dans une SOA

- Services beaucoup plus complexes
  - Restreints à certains utilisateurs
  - Différentes qualités de service (QoS)
  - Requiert un paiement avant l'utilisation
- Différents modes de communication
  - Interaction unidirectionnelle (pas de retour)
  - Réponse émanant d'un autre service que celui demandé
- Nécessitent une négociation avant l'exploitation

Avantages d'une SOA (1/2)

- Indépendance (*loose coupling*)
- Flexibilité : un service peut être localisé sur n'importe quel serveur, et retrouvé tant que l'annuaire est à jour
- Passage à l'échelle : les services peuvent être ajoutés ou retirés à la demande
- Transparence : une nouvelle implémentation peut être proposée sans perturbation pour les utilisateurs de services

Avantages d'une SOA (2/2)

- Tolérance aux fautes : si un service devient indisponible, un Client peut interroger l'annuaire pour retrouver un autre service offrant les mêmes fonctionnalités, et ainsi poursuivre son traitement

Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

Qu'est-ce qu'un service web

- Programme applicatif
  - Identifié par une URI (Uniform Resource Identification)
  - Dont les interfaces et liens (*bindings*) peuvent être définis, décrits, découverts par des artefacts XML
  - Qui supporte des interactions directes avec d'autres programmes applicatifs
  - Utilisant des messages XML
  - Accessible via les protocoles de l'Internet

Organismes de standardisation

- World Wide Web Consortium (W3C)
  - http://www.w3.org
  - HTML, XHTML, CSS, XML, XSLT, XML Query, ...
- Organizations for Advancement of Structured Information Standards (OASIS)
  - http://www.oasis-open.org
  - Standards e-Business pour les services web
- Web Services Interoperability Organizations (WS-I)
  - http://www.ws-i.org
  - Interopérabilité des messages échangés

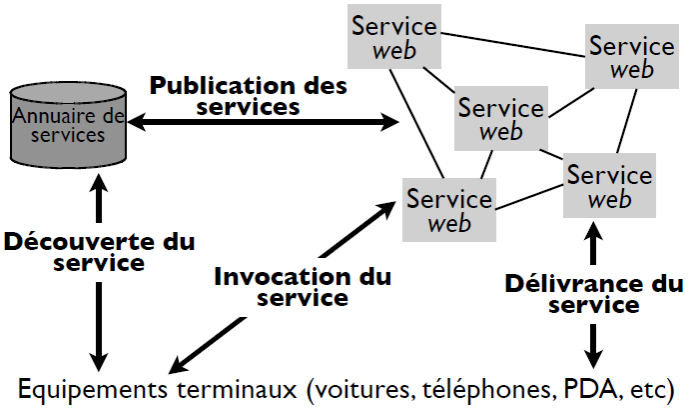
Caractéristiques des services web

- XML partout
- Orienté Messages
- Indépendant des langages de programmation
- Localisé dynamiquement
- Dynamiquement agrégé/composé
- Accessible à travers l'Internet
- Faiblement couplé
- Fondé sur des standards de l'industrie

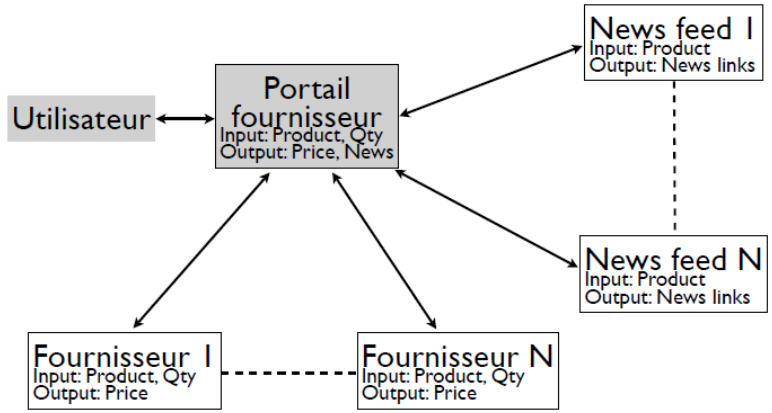
Avantages des services web

- Interopérable
  - connexion à travers des réseaux hétérogènes via les standards du web
- Economique
  - composants “recyclables”, forte intégration d’applications
- Automatique
  - pas d’intervention humaine, même pour des cas complexes
- Disponibilité
  - services sur n’importe quel équipement, n’importe où, n’importe quand
- Scalable
  - pas de limitation sur le type d’application ni sur leur nombre

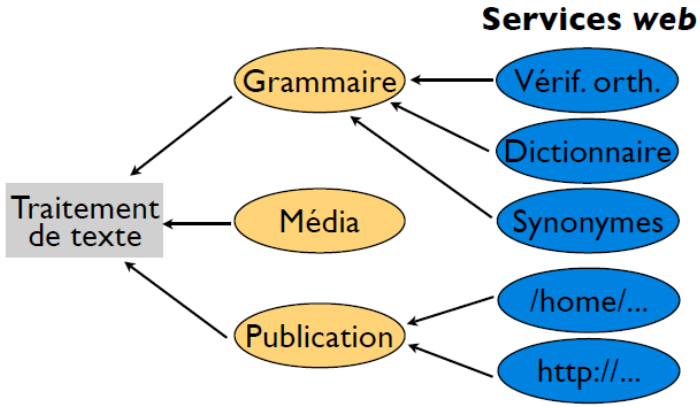
Architecture de haut niveau



Agrégation de services web



Applications virtuelles



Pile des services web

- Différents standards sont respectivement responsables de
  - La description du service
  - La publication et de la découverte du service
  - L'invocation du service

Discovery	**UDDI**
Description	**WSDL**
XML messaging	**XML-RPC, SOAP, XML**
Transport	**HTTP, SMTP, FTP, BEEP**

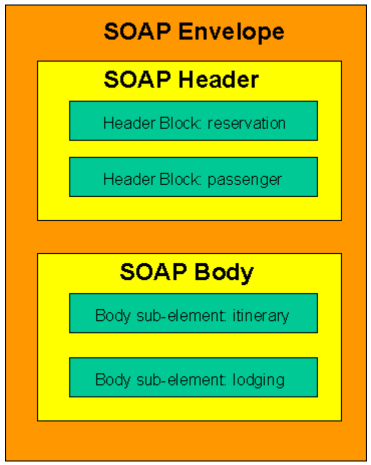
Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

SOAP

- Simple Object Access Protocol
- Définit le format des messages échangés
- Indépendant des plates-formes et langages de programmation
- XML est utilisé pour l'encodage des données
  - Protocole orienté "texte"

Message SOAP : format (1/2)



Message SOAP : format (2/2)

- Document XML contenant
  - Un élément *Envelope* obligatoire qui identifie le document XML comme un message SOAP
  - Un élément *Header* optionnel qui contient des informations sur l'entête du message
  - Un élément *Body* obligatoire qui contient les informations sur la requête/réponse
  - Un élément *Fault* optionnel qui renseigne sur les erreurs produites lors du traitement d'un message

Règles de syntaxe

- Un message SOAP doit être un document XML
- Un message SOAP doit faire référence à la spécification d'une enveloppe SOAP
- Un message SOAP doit faire référence à la spécification des règles d'encodage SOAP
- Un message SOAP ne doit pas contenir de référence à une DTD
- Un message SOAP ne doit pas contenir d'instructions de traitement XML

Squelette d'un message SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
  ...
  ...
</soap:Header>

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

</soap:Envelope>
```


L'élément *Envelope*

- Représente l'élément root d'un message SOAP
- Définit le document XML comme un message SOAP
- Utilise l'espace de nommage xmlns:soap
  - http://www.w3.org/2001/12/soap-envelope

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

Encodage (1/2)

- Définir les types de données utilisés dans le document
- Fondés sur le schéma XML du W3C
- Types simples
  - Types natifs du schéma XML (types simples, énumérations, tableaux de bits)
- Types composés
  - Structures
  - Tableaux
  - Types complexes

Encodage (2/2)

- Les types de données s'appliquent à n'importe quel élément enfant
- Attribut *encodingStyle*
- Syntaxe
  - soap:encodingStyle="URI"

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
...
Message information goes here
...
</soap:Envelope>
```


L'élément *Header*

- Permet de rajouter des informations spécifiques à l'application (authentification, sécurité, etc.)
- Si présent, premier sous-élément de *Envelope*
- Chaque sous-élément doit spécifier un espace de nommage
- Trois attributs s'appliquent à chaque sous-élément
  - *actor*, *mustUndertand*, *encodingStyle*
  - Définissent comment le sous-élément doit être traité

L'élément *Header* : exemple

- Authentification

```
<soap:Header>
    <wsse:UsernameToken>
        <wsse:Username>Scott</wsse:Username>
        <wsse:Password Type="wsse:PasswordText">
            le mot de passe de Scott
        </wsse:Password>
    </wsse:UsernameToken>
</soap:Header>
```

L'élément *Body*

- Renferme le contenu du message (ce qui doit être traité par le destinataire) : la requête ou la réponse
- Les sous-éléments de l'élément Body peuvent être qualifiés par un espace de nommage

```
<SOAP-ENV:Body>
    <ns1:doubleAnInteger xmlns:ns1="urn:MySoapServices">
        <param1 xsi:type="xsd:int">123</param1>
    </ns1:doubleAnInteger>
</SOAP-ENV:Body>
```


L'élément *Body:Fault*

- Dans le cas d'une erreur, contient le message de celle-ci
- Si présent, doit apparaître comme un sous-élément de l'élément Body
- Ne peut apparaître qu'une seule fois dans un message SOAP
- Contient 4 sous-éléments
  - *faultcode* : code d'identification de l'erreur
  - *faultstring* : une explication lisible de la faute
  - *faultactor* : nœud à l'origine de la faute
  - *detail* : information spécifique à l'application

L'élément *Body:Fault* : exemple

```
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Client</faultcode>
    <faultstring>Message does not have necessary info</faultstring>
    <faultactor>http://gizmos.com/order</faultactor>
    <detail>
      <PO:order xmlns:PO="http://gizmos.com/orders/">
        Quantity element does not have a value
      </PO:order>
      <PO:confirmation xmlns:PO="http://gizmos.com/confirm">
        Incomplete address: no zip code
      </PO:confirmation>
    </detail>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
```


L'encapsulation SOAP/HTTP

- Un message SOAP est une requête/réponse HTTP conforme aux règles d'encodage SOAP
- HTTP + XML = SOAP
- Une requête SOAP peut être une requête HTTP POST ou HTTP GET
- Une requête HTTP POST définit au moins deux informations : le type MIME (application/soap+xml) et l'encodage des caractères (par ex : utf-8) du message HTTP

Exemple de message SOAP : requête

```
POST /Employees HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
  <SOAP:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <SOAP:Body>
        <ns1:getEmployeeDetails
         xmlns:ns1="urn:MySoapServices">
            <param1>1016577</param1>
        </ns1:getEmployeeDetails>
    </SOAP:Body>
  </SOAP:Envelope>
```


Exemple de message SOAP : réponse

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
  <SOAP:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <SOAP:Body>
        <ns1:getEmployeeDetailsResponse xmlns:ns1="urn:MySoapServices">
            <return>
                <item>Bill Posters</item>
                <item>+1-212-7370194</item>
            </return>
        </ns1:getEmployeeDetailsResponse>
    </SOAP:Body>
  </SOAP:Envelope>
```


Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

Le langage WSDL (1/2)

- Web Services Description Language
- Document écrit en XML
- Produit par les fournisseurs de services
  - Description abstraite des services web
  - Opérations et messages transmis depuis et vers un service web
  - Lien concret vers un protocole de transport et un format de message

Le langage WSDL (2/2)

- Web Services Description Language
- Exploité par les demandeurs de services
  - Localisation d'un service web
  - Découverte des méthodes offertes et de leur(s) mode(s) de transaction
  - Invocation d'un service

Structure d'un document WSDL

- Cinq éléments majeurs
  - *types*
  - *message*
  - *portType*
  - *binding*
  - *service*
- Peut contenir d'autres éléments d'extension
- Possibilité de décrire plusieurs services dans un seul document WSDL

Les éléments majeurs

- *types*
  - types d'encodage pour les données échangées
- *message* : comparable aux paramètres d'une fonction
  - format des messages d'entrée et de sortie de chacune des opérations
- *portType* : comparable à une librairie de fonctions
  - opérations proposées par le service
- *binding*
  - protocoles de transport mis en œuvre et modes d'interaction de chaque opération
- *service*
  - localisation du service

Exemple de fichier WSDL (1/4)

```
<definitions targetNamespace="http://localhost/server.php">
 - <types>
   - <xsd:schema targetNamespace="http://localhost/server.php">
       <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
       <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
     </xsd:schema>
   </types>
 - <message name="sayhelloRequest">
     <part name="param1" type="xsd:string"/>
   </message>
 - <message name="sayhelloResponse">
     <part name="result" type="xsd:string"/>
   </message>
 - <portType name="helloservicePortType">
   - <operation name="sayhello">
       <documentation>une fonction qui dit bonjour</documentation>
       <input message="tns:sayhelloRequest"/>
       <output message="tns:sayhelloResponse"/>
     </operation>
   </portType>
```

Exemple de fichier WSDL (2/4)

- L'élément *types*
  - Encodage : http://schemas.xmlsoap.org/soap/encoding/
  - Format du message : http://schemas.xmlsoap.org/wsdl
- Les éléments *message*
  - type des messages *sayhellorequest* : *param1* de type *string*
  - type des messages *sayhelloresponse* : *return* de type *string*
- L'élément *portType*
  - nom du port : *helloservicePortType*
  - une opération : *sayhello*
  - format des messages d'entrée : *sayhellorequest*
  - format des messages de sortie : *sayhelloresponse*

Exemple de fichier WSDL (3/4)

```
<binding name="helloserviceBinding" type="tns:helloservicePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="sayhello">
    <soap:operation soapAction="http://localhost/server.php/sayhello" style="rpc"/>
    <input>
      <soap:body use="literal" namespace="http://localhost/server.php"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://localhost/server.php"/>
    </output>
  </operation>
</binding>
<service name="helloservice">
  <port name="helloservicePort" binding="tns:helloserviceBinding">
    <soap:address location="http://localhost/server.php"/>
  </port>
</service>
</definitions>
```


Exemple de fichier WSDL (4/4)

- L'élément *binding*
  - le mode d'interaction est RPC
  - le protocole de transport utilisé est HTTP (http://schemas.xmlsoap.org/soap/http)
  - pour l'opération *sayhello*, encodage des messages d'entrée et de sortie : *literal*
- L'élément *service*
  - le service s'appelle *helloservice*
  - le port *helloservicePortType* est accessible à l'adresse http://localhost/server.php

Récapitulatif WSDL

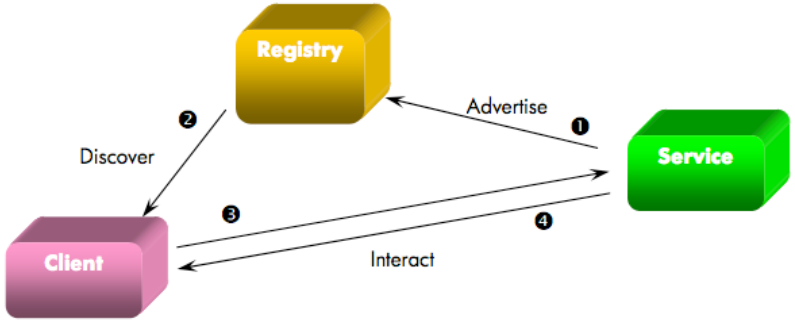
- Description des opérations assurées par un ou plusieurs services web
  - production automatique par les logiciels auteur
  - nom de chaque opération
  - messages d'entrée et de sortie
- Description des détails de communication lors d'échanges entre applications
  - Type de communication
  - Protocoles mis en œuvre
- Pas de pré-arrangement manuel entre applications

Contenu du cours

- Introduction
- Les architectures orientées services (SOA)
- Les services web
- Le protocole SOAP
- Le langage WSDL
- Les annuaires UDDI

Rappel : principe d'une SOA

- L'annuaire fournit un moyen pour publier et trouver des informations sur les services web



Qu'est-ce qu'un annuaire

- Infrastructure qui permet la publication et la découverte de services, d'ontologies ou d'autres artefacts pour le partage d'informations
- Tierce partie qui facilite les interactions dynamiques entre applications faiblement couplées
- Disponible au sein des organisations comme une ressource partagée souvent disponible à travers un service web...

L'annuaire UDDI

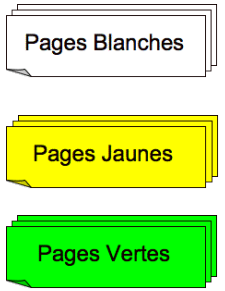
- Spécialisé dans le stockage d'informations relatives à des services web
- Annuaire d'interfaces de services web décrites en WSDL
- Communication fondée sur l'échange de messages SOAP
- Utilise les standards du W3C et de l'IETF
  - XML
  - HTTP, DNS

Problèmes visés par UDDI

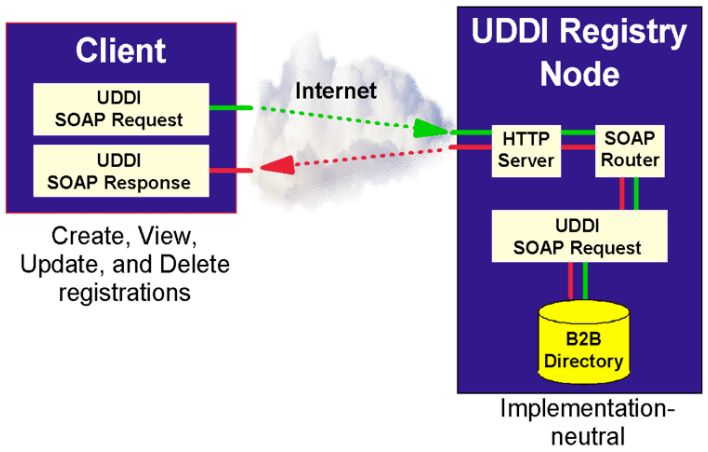
- Rendre possible la découverte du bon partenaire parmi les milliers actuellement disponibles
- Définir les règles de transaction
- Atteindre de nouveaux clients et faciliter l'accès aux clients existants
- Etendre les offres d'une entreprise en proposant de nouveaux services

Données d'un annuaire UDDI

- Informations sur les services proposés par une ou plusieurs entreprises
- Pages blanches
  - Adresse, contact, etc.
- Pages jaunes
  - Catégorisation industrielle
- Pages vertes
  - Informations techniques concernant les services



Principes UDDI



Cas d'usage

- L'industrie publie une interface 'standard' UDDI pour la réservation de vols aériens
- Les compagnies aériennes enregistrent des services retournant les horaires et prix de leurs vols
- Les agences de voyage recherchent dans l'annuaire UDDI les descriptions des services des compagnies aériennes
- Les agences de voyage communiquent directement avec divers services de différentes compagnies

Références

- W3 Schools
  - http://www.w3schools.com
  - http://www.w3.org/TR/soap12-part0/