

Web Services

Architecture et Protocoles

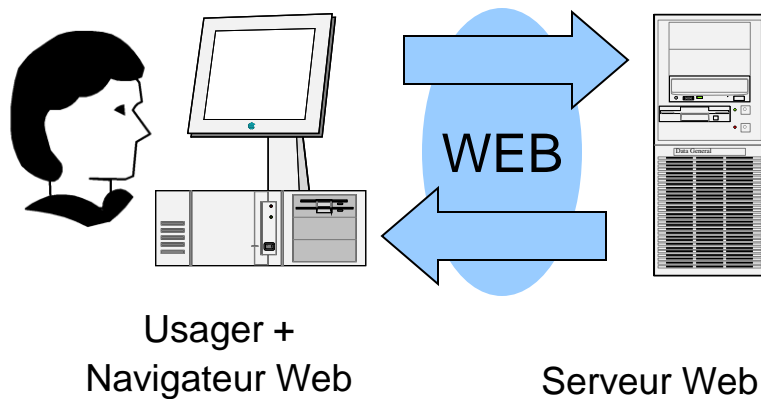
Rémi Badonnel
LORIA – INRIA Lorraine
badonnel@loria.fr

Web Services

- Contexte & définition
- Architecture orientée services
- Protocoles XML sous-jacents
 - Invocation (SOAP)
 - Description (WSDL)
 - Découverte (UDDI)
- Composition de services
- Plate-formes opérationnelles
- Bilan

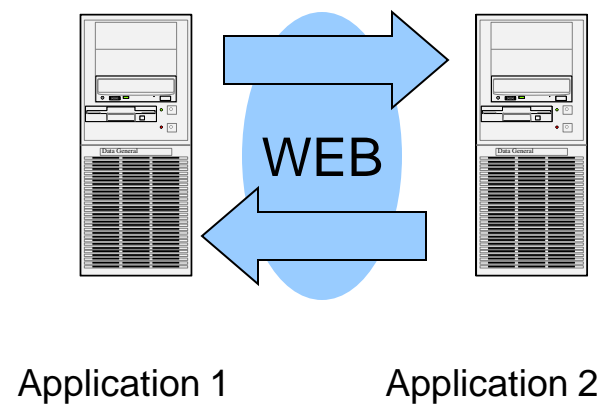
Contexte

- Evolution du Web des usagers au Web des communications entre applications



Communications entre un usager et une application

L'entité principale est l'usager.
Ex : Achat d'un livre via les pages Web d'une compagnie



Communications entre applications

L'entité principale est l'application.
Ex : Achat automatisé de fournitures (via un Web Service) ³

Contexte

- Limites des middlewares "traditionnels"
 - Mono-langage : Java RMI,
 - Mono-plateforme : DCOM sous Windows,
 - Complexe à mettre en œuvre : CORBA.
- Adaptation des architectures réparties au contexte de l'Internet où **le Web est considéré comme un nouveau middleware.**
 - Multi-langage, Multi-plateforme,
 - Spécification garantie par un organisme indépendant,
 - Simple à mettre en œuvre.

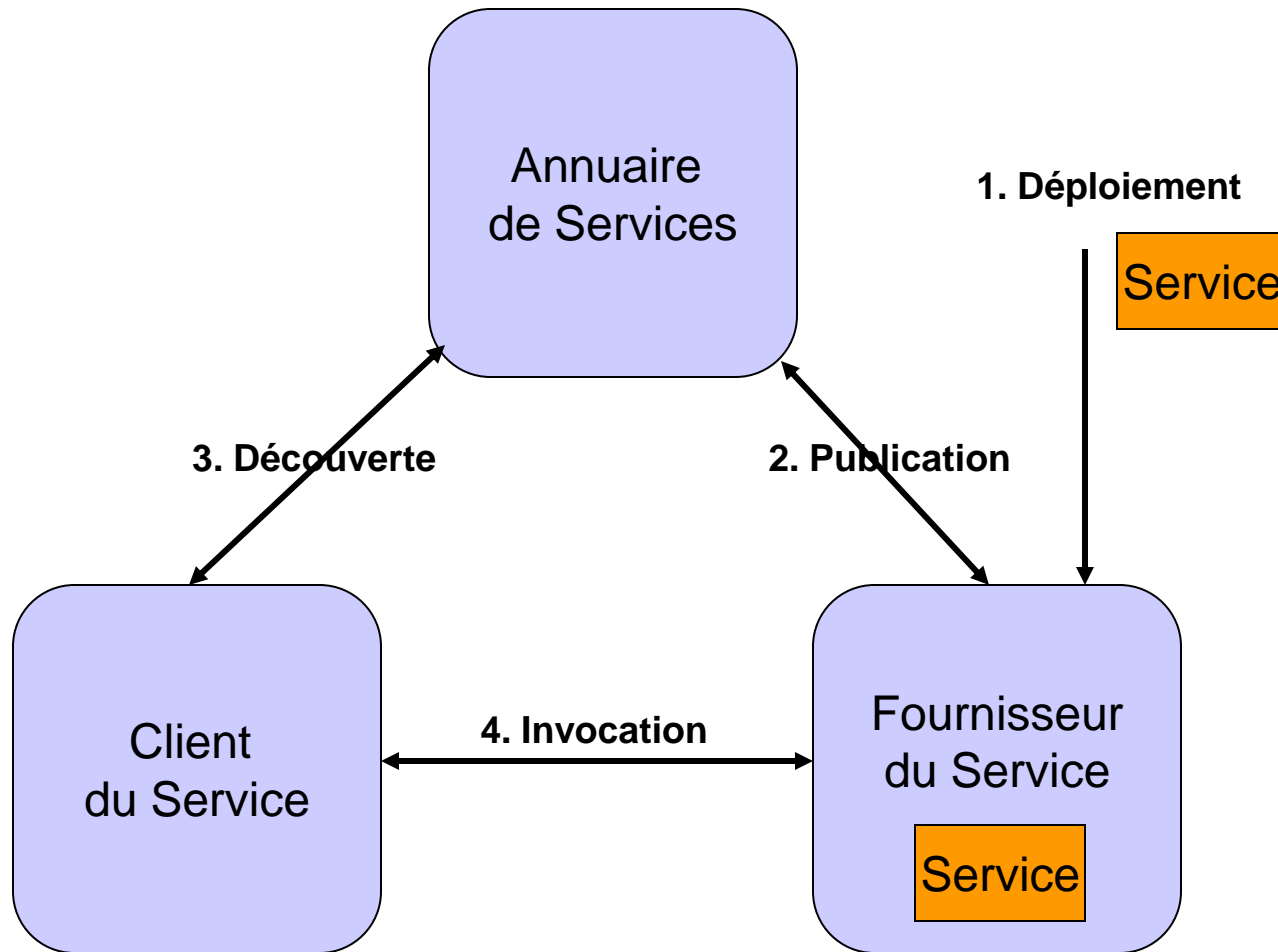
Définition

- Un Web Service est une application modulaire :
 - Mise à disposition sur l'Internet ou sur un réseau privé (Intranet),
 - Auto-descriptive, publiable et accessible en utilisant le langage XML et les protocoles standards du Web,
 - Indépendante du système d'exploitation et du langage de programmation,
 - Visant à exposer une ou plusieurs fonctionnalités (souvent commerciale(s)) d'une entreprise.

Définition

- Le modèle des Web Services est défini par une architecture et un ensemble de protocoles standardisés : SOAP, WSDL, UDDI.
 - Spécifications garanties par W3C et OASIS,
 - Interopérabilité entre implantations gérée par la WS-I (Web Services Interoperability) Organization.
- Objectifs du modèle
 - Modularité,
 - Réutilisation et composition de services,
 - Interopérabilité,
 - Dialogue entre environnements et plate-formes hétérogènes,
 - Couplage faible (communications synchrones/asynchrones),
 - Intégration,
 - Intégration du système d'information au sein et en dehors de l'entreprise,
 - Masquage de la complexité.

Architecture Orientée Services



SOA pour les Web Services

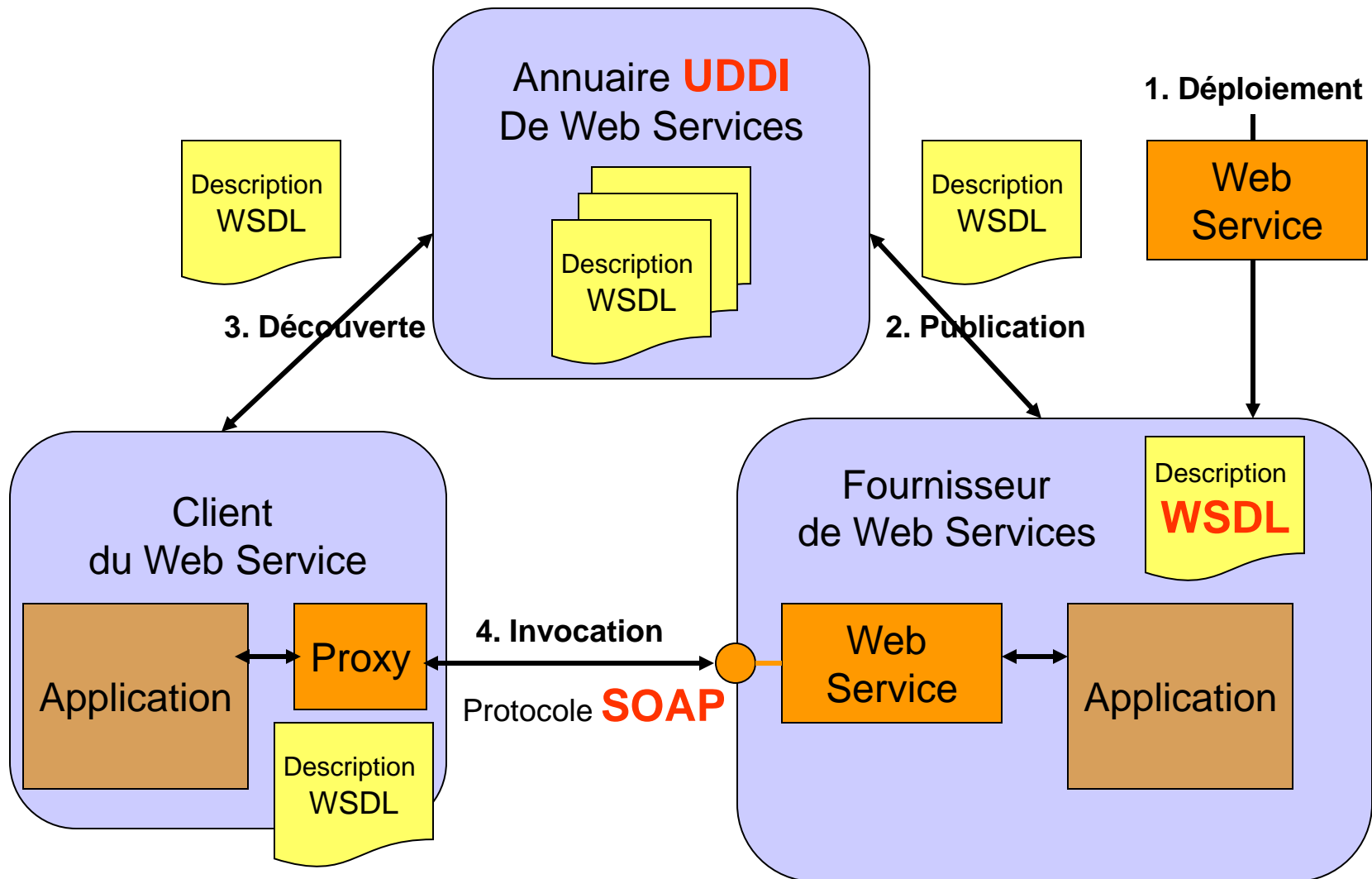
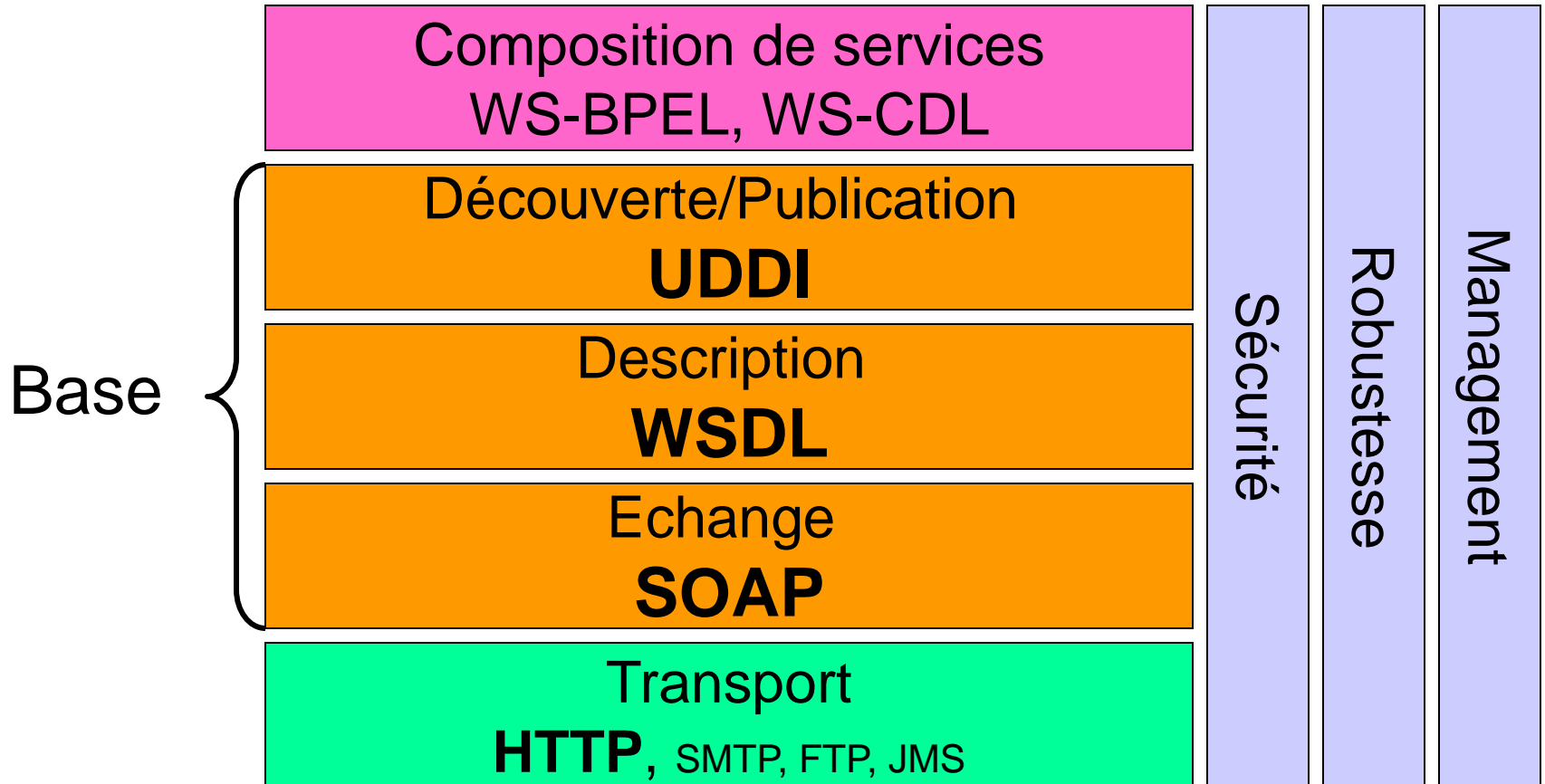
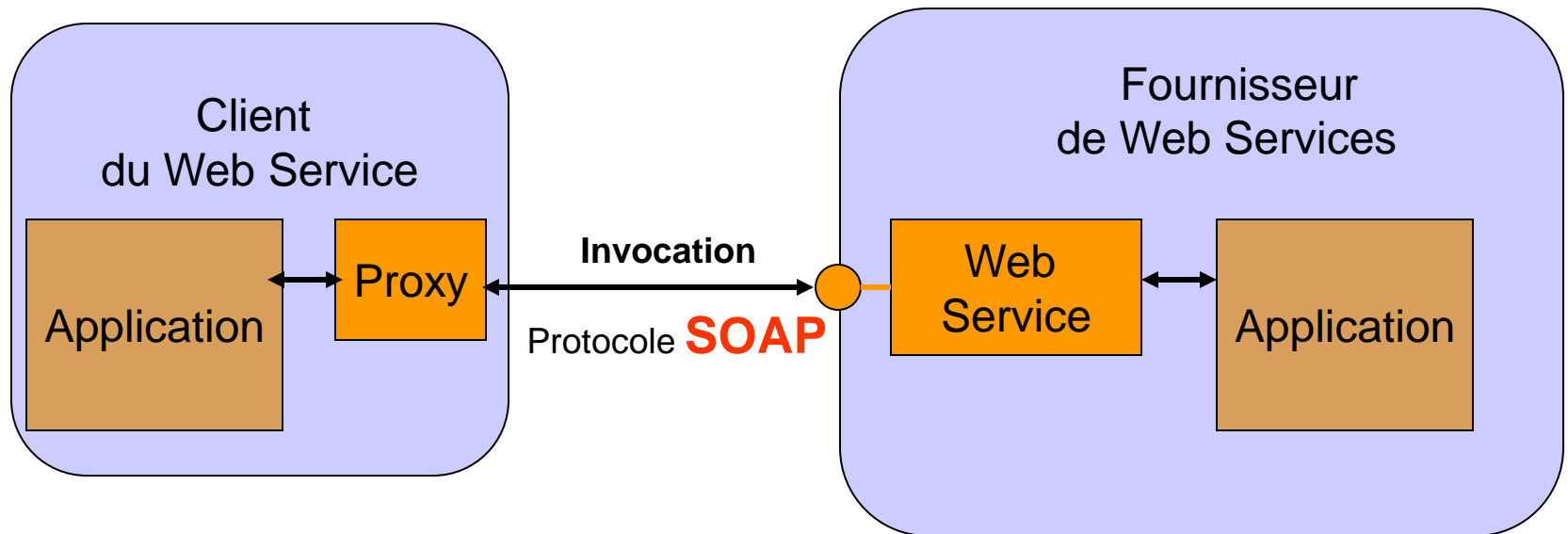


Diagramme des protocoles



SOAP

Invocation d'un Web Service



Qu'est-ce que SOAP ?

- Définition

- Protocole léger d'échange d'information structurée dans un environnement distribué,
- Format des messages en XML,
- Transport sur le protocole HTTP (usuellement),
 - Ouvert à d'autres protocoles tels que SMTP, FTP, JMS
- Large inspiration de XML-RPC

- Terminologie

- Initialement : Simple Object Access Protocol
- Temporairement : Service Oriented Architecture Protocol ?
- Aujourd'hui : SOAP n'est plus un acronyme.

Exemple : Message SOAP

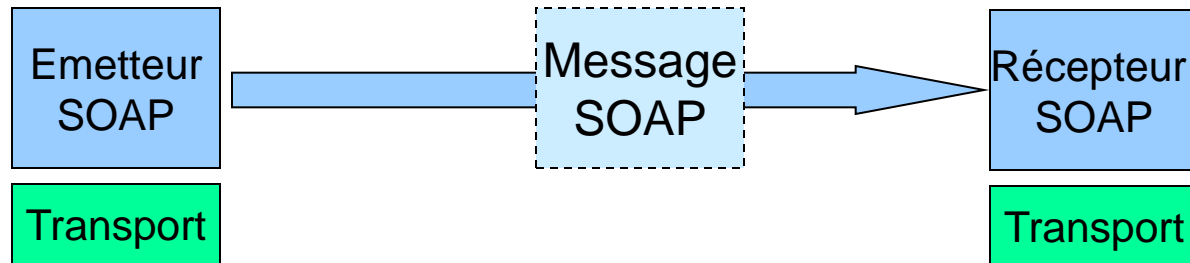
```
<env:Envelope
  xmlns:env="http://www.w3.org/2002/06/soap-envelope/"
  env:encodingStyle=" http://www.w3.org/2002/06/soap-encoding"/>
  <env:Header>
    <t:Transaction
      xmlns:t="some-URI"
      env:mustUnderstand="1">
      5
    </t:Transaction>
  </env:Header>
  <env:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>
        GEM
      </symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
```

SOAP - Spécification

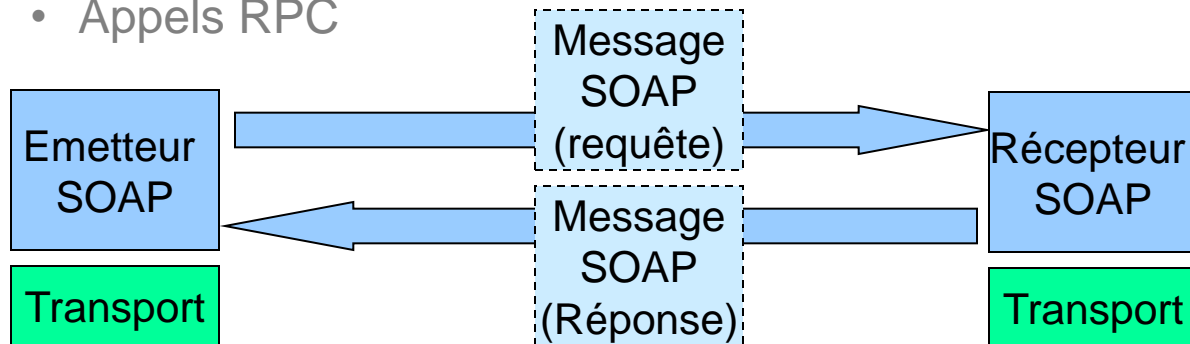
- Initiative conjointe de Microsoft et IBM,
- Spécification standard gérée par le W3C
 - SOAP 1.2, W3C recommandation W3C, juin 2003,
 - SOAP 1.1, W3C note soumise en mai 2000,
- Parties de la spécification
 - SOAP message exchange model
 - Définition du modèle de communication
 - SOAP envelope
 - Définition de la structure d'un message
 - SOAP encoding
 - Règles d'encodage des types de données
 - Using SOAP in HTTP
 - Transport des messages avec HTTP
 - SOAP for RPC
 - Conventions de représentation des appels RPC

Echange de messages SOAP

- Modèle d'échange de base
 - Message à sens unique (one-way)

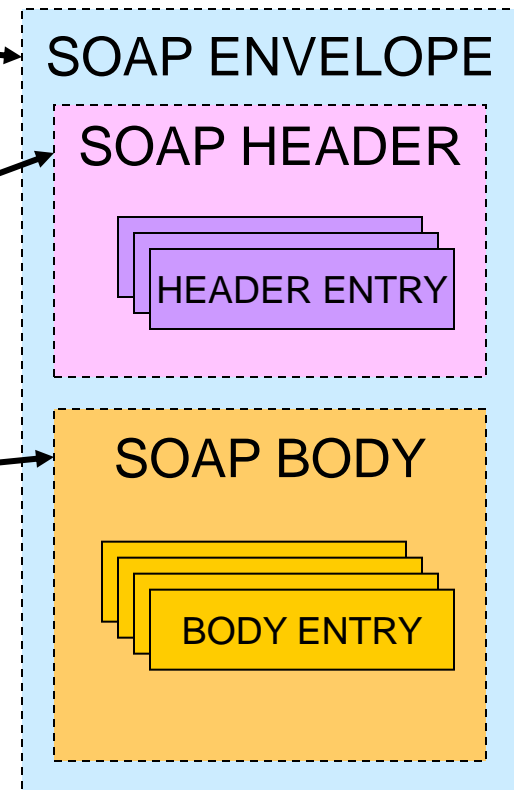


- Modèle d'échange implicite
 - Requête/Réponse
 - Appels RPC



Structure d'un message

- Envelope
 - Contenant d'un message,
 - Élément racine XML,
 - Schéma XML
 - <http://www.w3.org/2002/06/soap-envelope/>
- Header (optionnel)
 - Entrées non applicatives,
 - Ex : Numéros de session.
- Body (obligatoire)
 - Entrées applicatives,
 - Ex : nom des procédures, nom des paramètres, valeurs de paramètres,
 - Retour d'erreurs.



Entête d'un message

- Informations de l'entête <HEADER>
 - Entrées non applicatives, ne dépendant pas nécessairement de l'application réceptrice,
 - Identifiants, transactions, certificats de sécurité.
- Attributs des entrées d'entête
 - Attribut "role" vaut {none}, {next}, {ultimateReceiver} définit le rôle des nœuds SOAP dans le traitement du message,
 - Attribut booléen "mustUnderstand" vaut {0} si l'entrée est optionnelle ou {1} si elle doit être comprise par l'application réceptrice pour que le message soit traité.
- Exemple

```
<env:Header>
  <t:Transaction xmlns:t="some-URI"
    SOAP-ENV:mustUnderstand="1">
    5
  </t:Transaction>
</env:Header>
```


Corps d'un message

- Informations du corps <BODY>
 - Entrées applicatives,
 - Représentation d'un appel de procédures par un ensemble d'entrées.
- Règles d'encodage des données
 - Type simple, type composite (structure, vecteur) ,
 - Schéma XML.
 - <http://www.w3.org/2002/06/soap-encoding>

- Exemple

```
<env:Body>
  <m:GetLastTradePrice xmlns:m="Some-URI">
    <symbol>
      GEM
    </symbol>
  </m:GetLastTradePrice>
</env:Body>
```

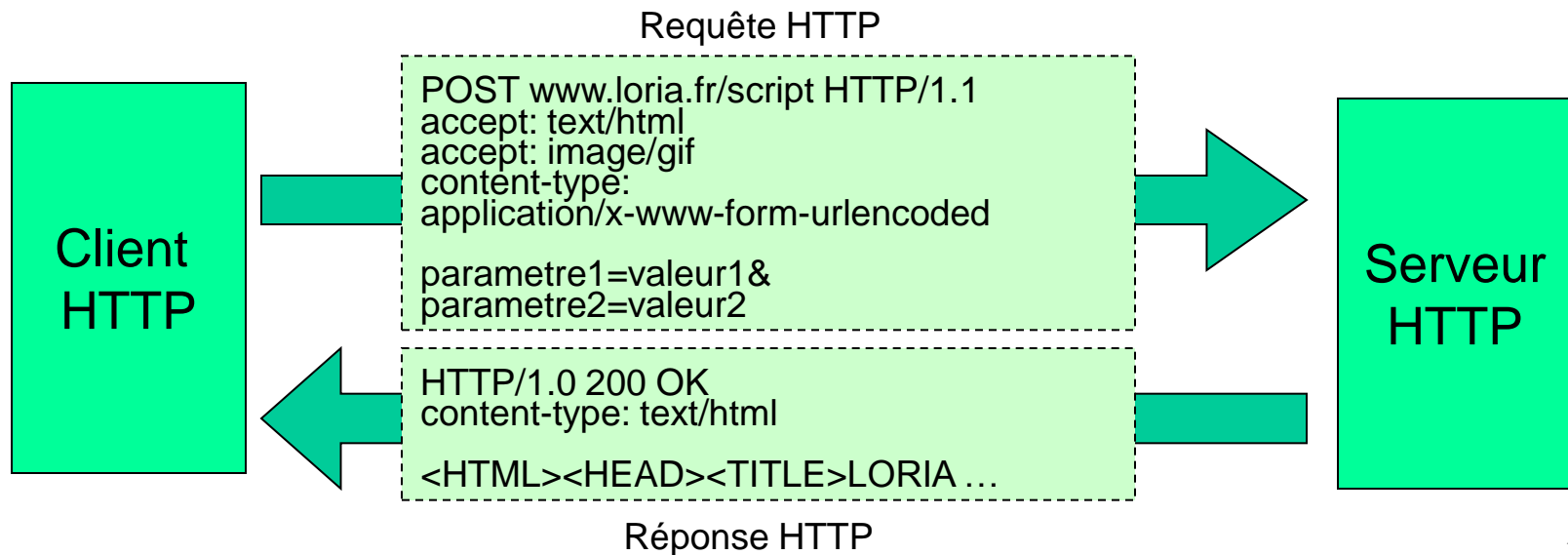
Retour d'erreurs

- Entrée <fault> dans le corps du message
- Quatre sous-éléments
 - **faultcode** (obligatoire)
 - Code d'erreur exploité par le logiciel,
 - Client, Server, MustUnderstand, VersionMismatch,
 - **faultstring** (obligatoire)
 - Explication littérale de l'erreur,
 - **faultactor**
 - Erreur lors de cheminement du message,
 - **detail**
 - Détail de l'erreur si elle porte sur le corps du message.
- Exemple

```
<env:Body>  
  <env:Fault>  
    <faultcode>SOAP-ENV:MustUnderstand</faultcode>  
    <faultstring>Must Understand Error</faultstring>  
  </env:Fault>  
</env:Body>
```

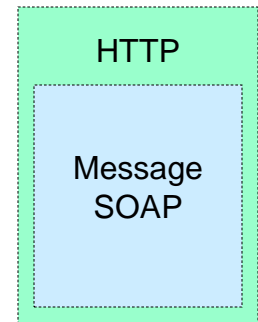
Transport de messages SOAP

- HTTP (HyperText Transfer Protocol)
 - Protocole bidirectionnel synchrone : requête/réponse
 - Transfert de fichiers (habituellement HTML)
 - Requête méthode GET/méthode POST
 - Simple et disponible sur toute plate-forme



Liaison SOAP/HTTP

- SOAP utilise le mode POST de HTTP
- Requête SOAP/HTTP
 - Ajout du champs "action"
 - Acheminement et Filtrage par les serveurs HTTP
 - Content-Type: application/soap+xml
 - Message SOAP
- Réponse SOAP/HTTP
 - Content-Type: application/soap+xml
 - Etat 2xx
 - Requête SOAP reçue et traitée avec succès,
 - Message SOAP envoyée contient la réponse ou rien (one-way model).
 - Etat 500
 - Requête SOAP échouée,
 - Envoi d'un message SOAP contenant le retour d'erreur.



Appel RPC en SOAP

- Modèle d'échange Requête/Réponse
- Sérialisation de l'appel et du retour d'appel
 - Contenu dans le corps du message SOAP
 - Conventions de nommage
 - nomDeProcédure (param1, param2, param3) : param4
 - Requête :

```
<ns:nomDeProcédure>  
    <ns:param1> valeur1 </ns:param1>  
    <ns:param2> valeur2 </ns:param2>  
    <ns:param3> valeur3 </ns:param3>  
</ns:nomDeProcédure>
```
 - Réponse :

```
<ns:nomDeProcédureResponse>  
    <ns:param4> valeur4 </ns:param4>  
</ns:nomDeProcédureResponse>
```

Requête SOAP/HTTP

POST /StockQuote HTTP/1.1

Host: www.stockquoteserver.com

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

Action: "Some-URI"

```
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-envelope/"  
  env:encodingStyle=" http://www.w3.org/2002/06/soap-encoding/">
```

```
<env:Header>
```

```
<t:Transaction xmlns:t="some-URI"> 5 </t:Transaction>
```

```
</env:Header>
```

```
<env:Body>
```

```
<m:GetLastTradePrice xmlns:m="Some-URI">
```

```
<symbol> GEM</symbol>
```

```
</m:GetLastTradePrice>
```

```
</env:Body>
```

```
</env:Envelope>
```

Réponse SOAP/HTTP

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: nnnn

```
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-envelope/"  
  env:encodingStyle=" http://www.w3.org/2002/06/soap-encoding/">
```

```
  <env:Body>  
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">  
      <Price>34.5</Price>  
    </m:GetLastTradePriceResponse>  
  </env:Body>
```

```
</env:Envelope>
```

Réponse SOAP/HTTP (erreur)

HTTP/1.1 500 Internal Server Error

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: nnnn

```
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-envelope/"  
  env:encodingStyle=" http://www.w3.org/2002/06/soap-encoding/">
```

```
<env:Body>
```

```
<env:Fault>
```

```
<faultcode>SOAP-ENV:MustUnderstand</faultcode>
```

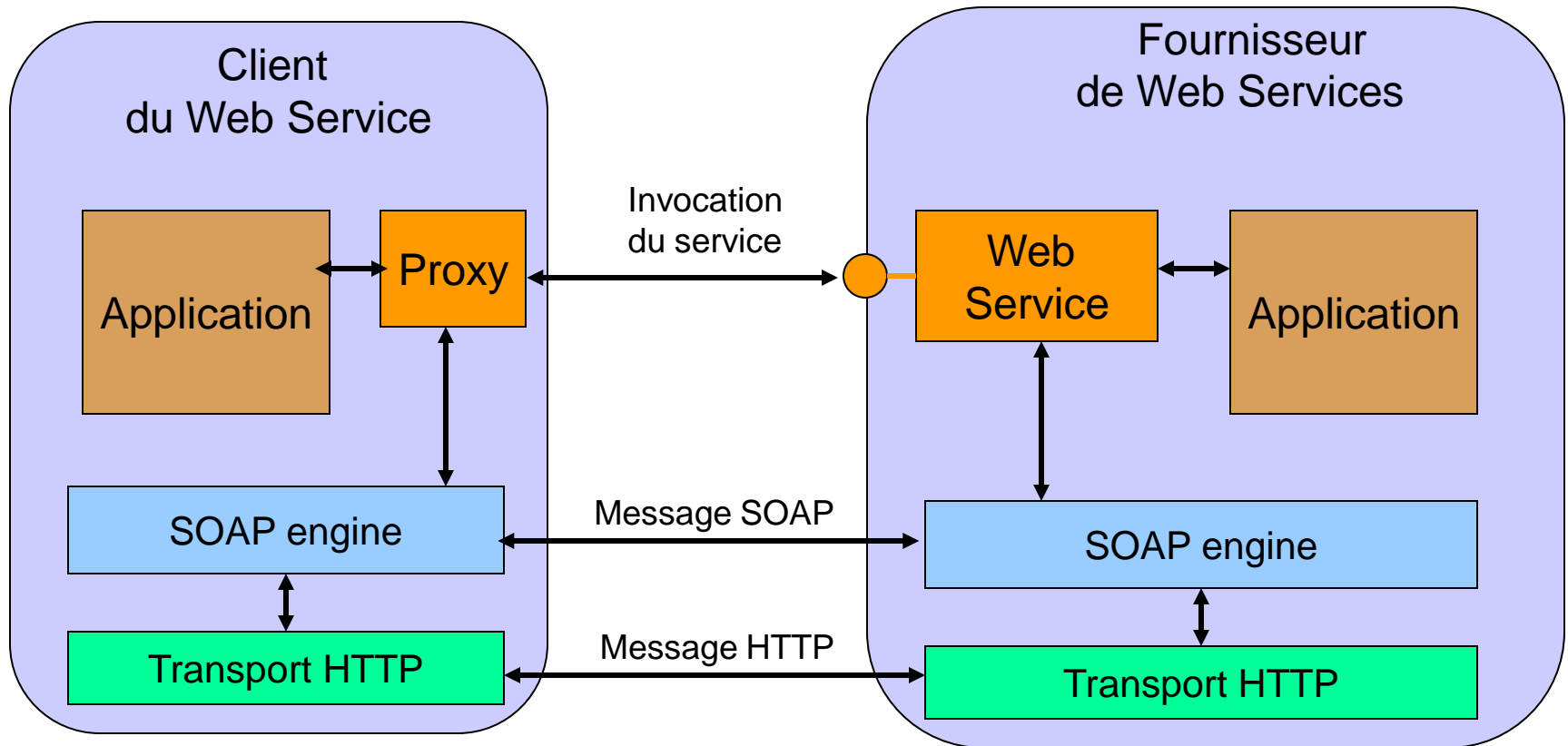
```
<faultstring>SOAP Must Understand Error</faultstring>
```

```
</env:Fault>
```

```
</env:Body>
```

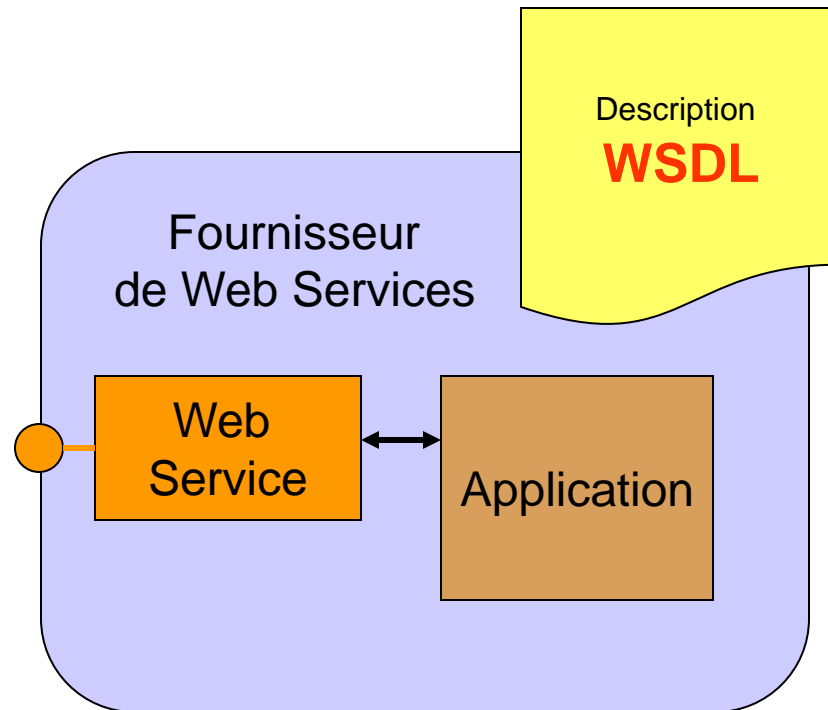
```
</env:Envelope>
```


Point de vue architectural



WSDL

Description d'un Web Service

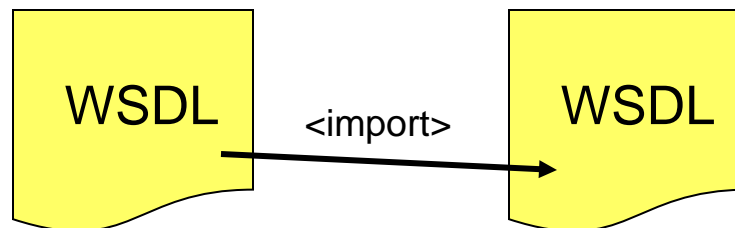
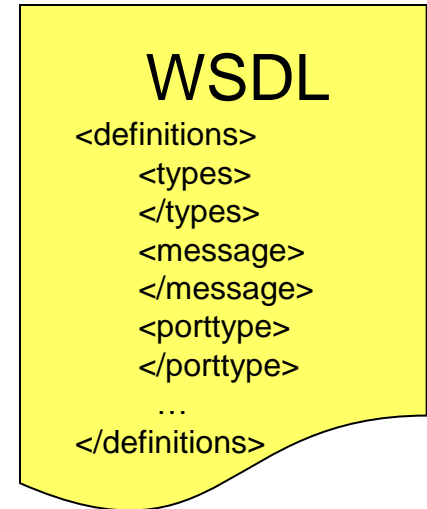


Qu'est-ce que WSDL ?

- Web Service Definition Language
- Langage de description des Web Services
 - Description abstraite de l'interface du service,
 - Ensemble d'opérations, de messages et de types de données
 - Description concrète de l'implantation du service,
 - Liaison à des formats de message concrets
 - Liaison à des protocoles de transport et des serveurs réseaux
- Document XML (Schéma XML)
- Initiative de Ariba, IBM et Microsoft
- Spécification standard géré par le W3C
 - WSDL 1.1, W3C Note, soumise en mars 2001

Document WSDL

- Défini en XML
- Structuré comme un ensemble de définitions
 - élément racine `<definitions>`
- Modulaire
 - Fragmentation des définitions en plusieurs fichiers
 - Séparation des descriptions abstraites et concrètes
 - Réutilisation de définitions de services
 - import d'autres documents WSDL et XSD

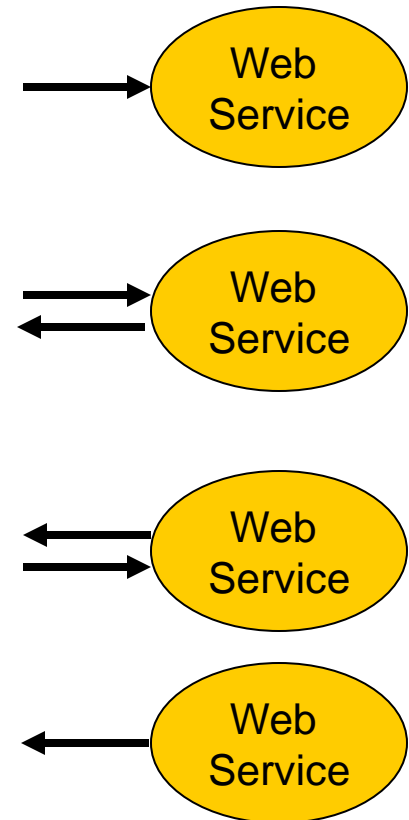


Description abstraite

- Définition de l'interface du service
- <types>
 - Définition des types de données utilisés dans les messages
 - Référence à un système de typage tel que Schéma XML
- <message>
 - Définition typée abstraite des données échangées
 - Ex : appel de méthode, retour de l'appel
- <operation>
 - Définition abstraite d'un ensemble cohérent de messages (entrées/sorties) correspondant à l'interaction avec le Web Service
 - Ex: appel de méthode + retour de l'appel
- <portType> (*type de port*)
 - Définition abstraite d'un ensemble d'opérations

Opérations

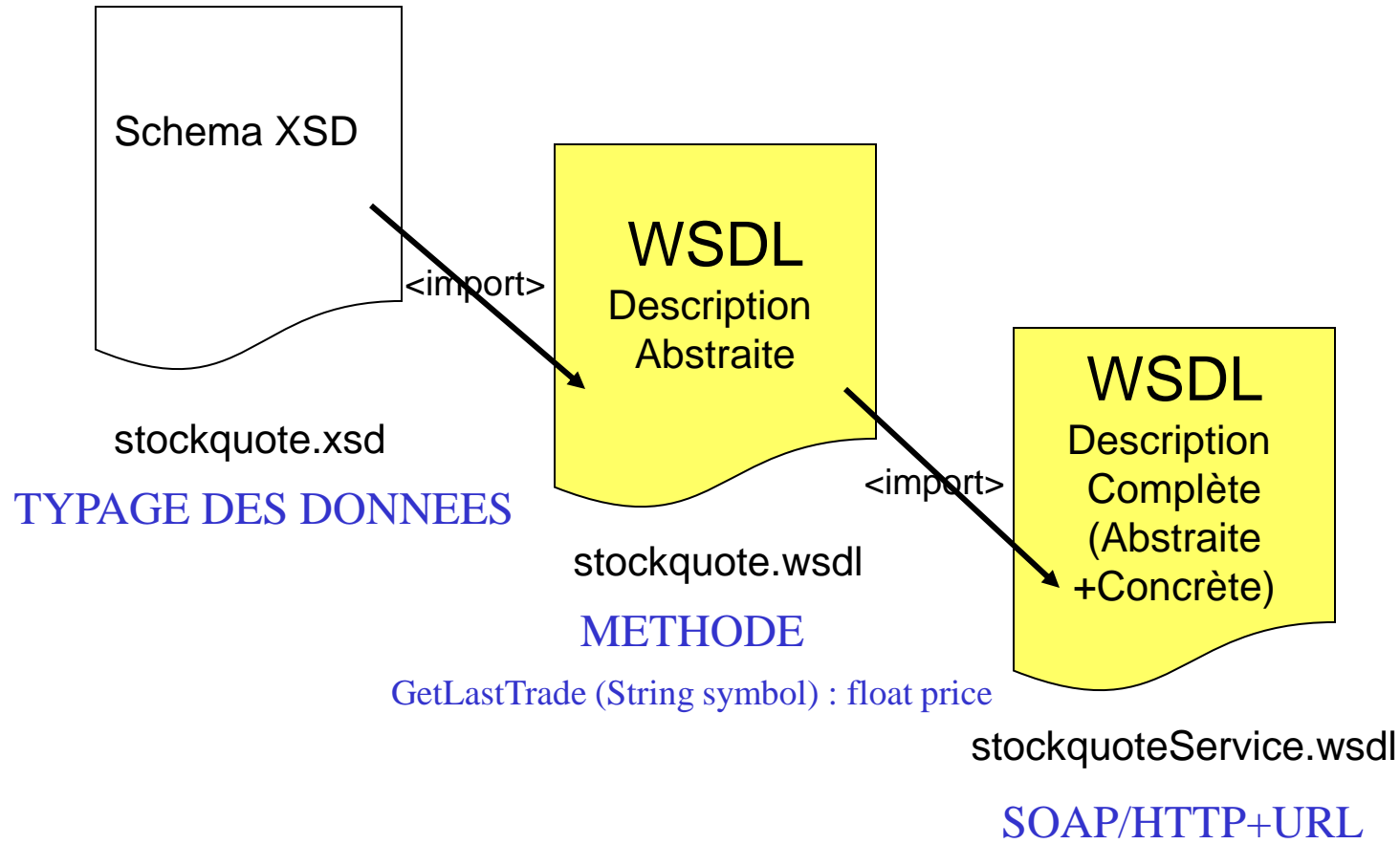
- Quatre types d'opérations possibles
 - Combinaison de messages
 - Utilisation des balises <input>, <output>
- Opération one-way
 - Le service reçoit un message <input>.
- Opération request-response
 - Le service reçoit un message requête <input> puis renvoie au client un message réponse <output> ou un message erreur <fault>.
- Opération solicit-response
 - Le service envoie un message sollicitation <output> puis reçoit du client un message réponse <output> ou un message d'erreur <fault>.
- Opération notification
 - Le service envoie un message notification <output>.



Description concrète

- Définition de l'implantation d'un service donné
- <binding> (*liaison*)
 - Description des protocoles (SOAP, HTTP...) et des formats de messages concrets pour chaque <porttype>
- <port>
 - Définition d'un point d'entrée en associant un <binding> à une adresse réseau d'un serveur (URL).
- <service>
 - Définition d'un Web Service comme un ensemble de <port>.

Exemple : StockQuote



Exemple : Schema XSD

- <http://example.com/stockquote/stockquote.xsd>

```
<?xml version="1.0"?>
<schema targetNamespace="http://example.com/stockquote/schemas"
        xmlns="http://www.w3.org/2000/10/XMLSchema">

    <element name="TradePriceRequest">
        <complexType>
            <all> <element name="symbol" type="string"/> </all>
        </complexType>
    </element>
    <element name="TradePrice">
        <complexType>
            <all> <element name="price" type="float"/></all>
        </complexType>
    </element>

</schema>
```

Exemple : Description Abstraite

- <http://example.com/stockquote/stockquote.wsdl>

- **<definitions>**

- Element racine

```
<?xml version="1.0"?>  
<definitions name="StockQuote ... >
```

- **<types>**

- Définition des types de données
- Importation du schéma XML

Schema
XML

```
<types>  
  <import namespace="http://example.com/stockquote/schemas"  
    location="http://example.com/stockquote/stockquote.xsd"/>  
</types>
```

Exemple : Description Abstraite

- <http://example.com/stockquote/stockquote.wsdl>

- **<message>**

```
<message name="GetLastTradePriceInput">  
    <part name="body" element="xsd1:TradePriceRequest"/>  
</message>
```

```
<message name="GetLastTradePriceOutput">  
    <part name="body" element="xsd1:TradePrice"/>  
</message>
```

- **<portType> & <operation>**

```
<portType name="StockQuotePortType">  
    <operation name="GetLastTradePrice">  
        <input message="tns:GetLastTradePriceInput"/>  
        <output message="tns:GetLastTradePriceOutput"/>  
    </operation>  
</portType>
```

```
</definitions></xml>
```

Exemple : Description Complète

- <http://example.com/stockquote/stockquoteService.wsdl>
 - Importation de la description abstraite

WSDL
Description
Abstraite

```
<?xml version="1.0"?>
<definitions name="StockQuote">
  <import namespace="http://example.com/stockquote/definitions"
    location="http://example.com/stockquote/stockquote.wsdl"/>
```

- **<binding>**

- Liaison SOAP/HTTP

```
<binding name="StockQuoteSoapBinding" type="defs:StockQuotePortType">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input><soap:body use="literal"/></input>
    <output><soap:body use="literal"/></output>
  </operation>
</binding>
```

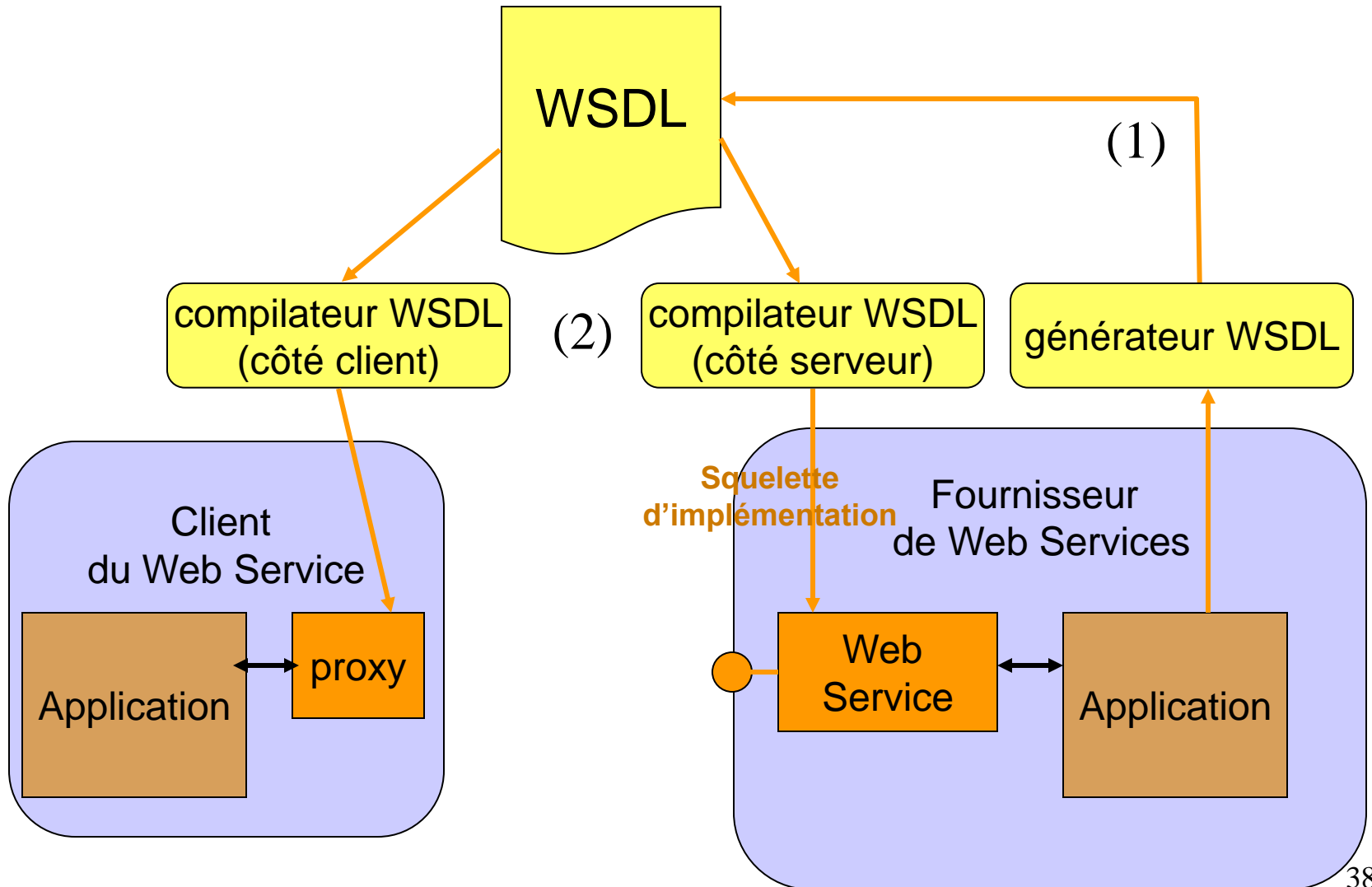
Exemple : Description Complète

- <http://example.com/stockquote/stockquoteService.wsdl>
 - `<service>` & `<port>`
 - Ensemble de points d'entrée
 - Indication de l'adresse réseau (URL)

```
<service name="StockQuoteService">  
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">  
    <soap:address location="http://example.com/stockquote"/>  
  </port>  
</service>
```

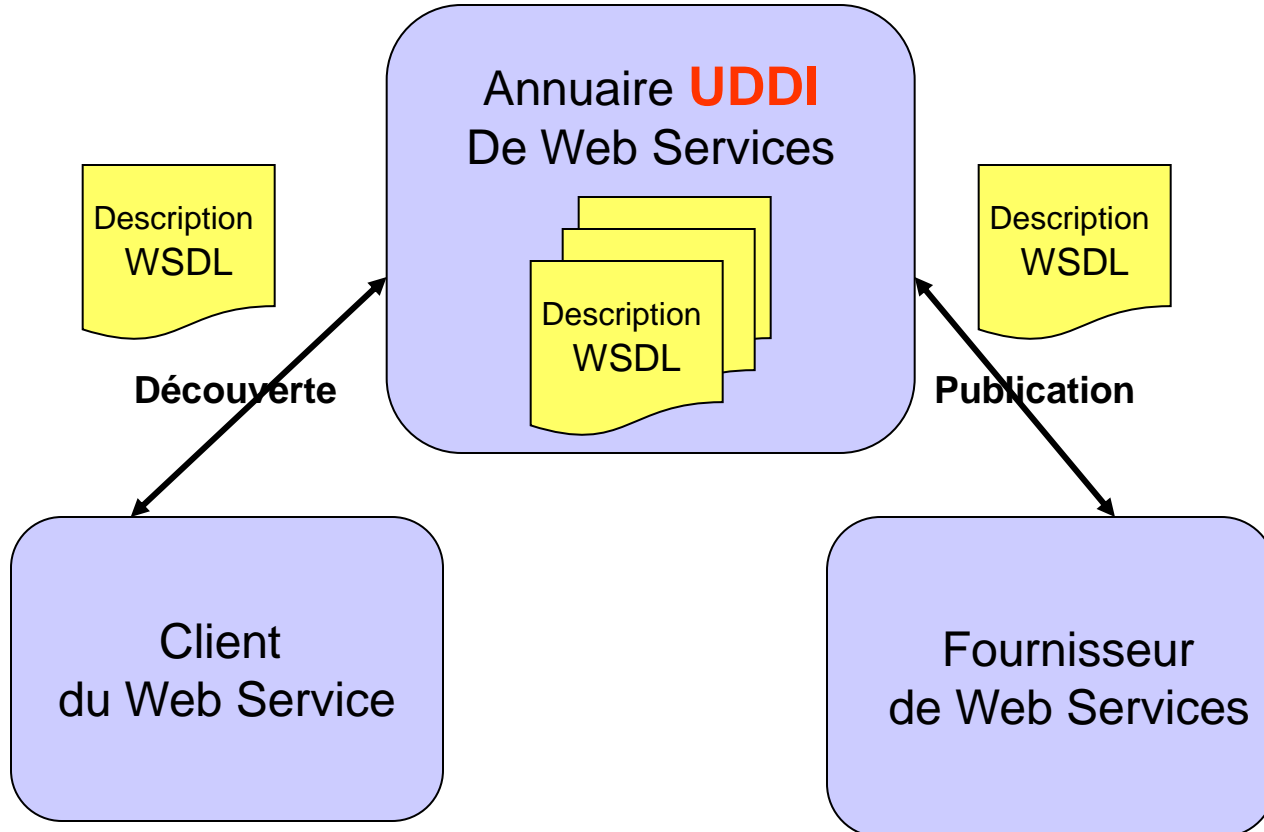
```
</definitions></xml>
```

Point de vue architectural



UDDI

Découverte et publication d'un Web Service



Qu'est ce que UDDI ?

- Universal Description, Discovery and Integration
- Service d'annuaires (Business Registry) recensant les entreprises et leurs services à l'échelle mondiale
 - Recherche et publication de services (dont Web Services)
 - Automatisation des échanges entre partenaires commerciaux
- Structure de données définie par une grammaire XML
- Accès par une API vue comme un Web Service
- Initiative de Ariba, IBM et Microsoft
- Spécification définie par un consortium industriel
 - UDDI 3.0, UDDI Working Group, OASIS

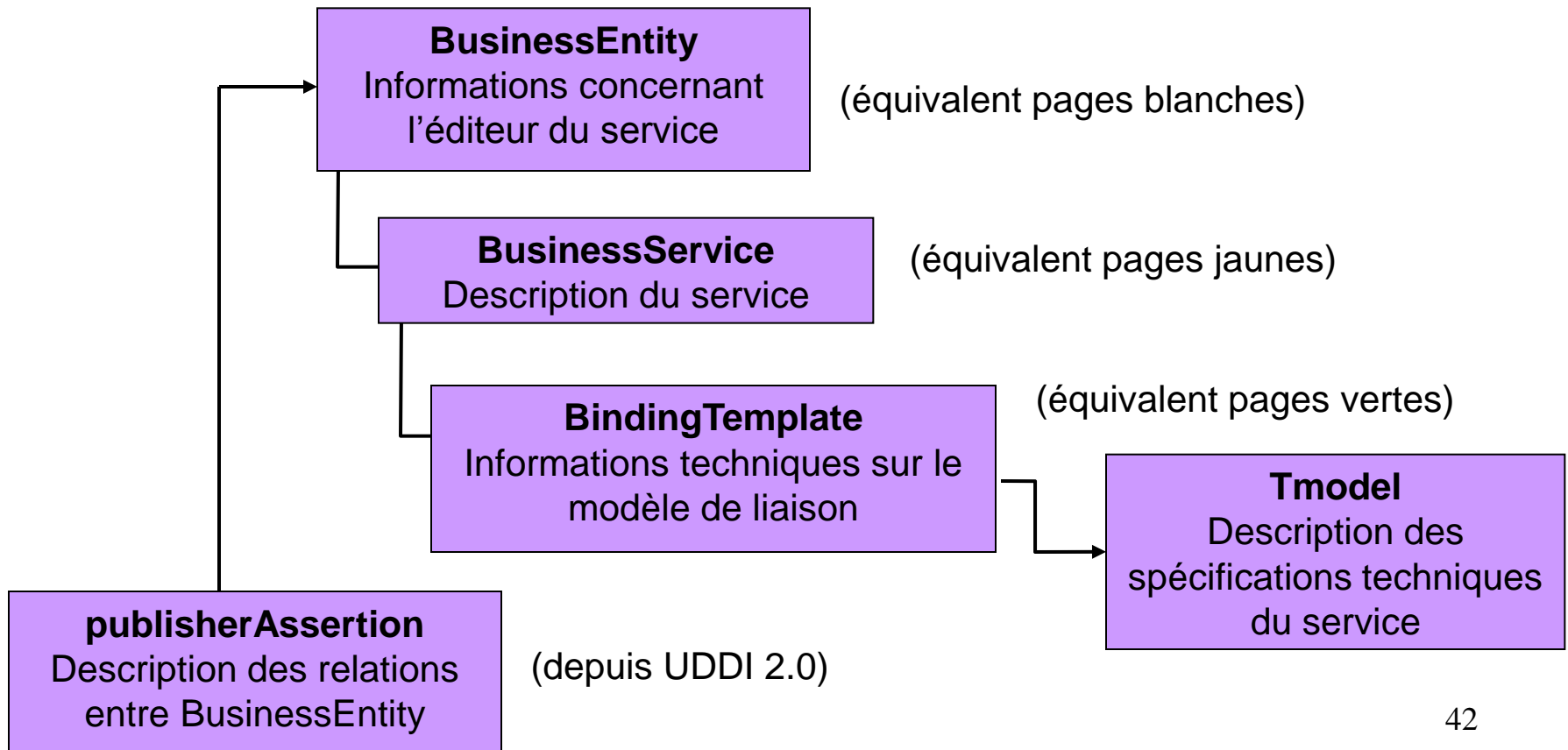
Annuaire(s) UDDI

- Pages blanches
 - Description des entreprises,
 - Nom, adresses, contacts, identifiants
- Pages jaunes
 - Catégories industrielles fondées sur des taxonomies standards (NAICS, UN/SPSC)
 - Types d'industries,
 - Types de produits de services,
 - Information sur la localité géographique.
- Pages vertes
 - Références techniques sur les services fournis,
 - Références à la spécification d'un Web Service

Structure de données

- Format défini par XML Schéma

- http://uddi.org/schema/uddi_v3.xsd

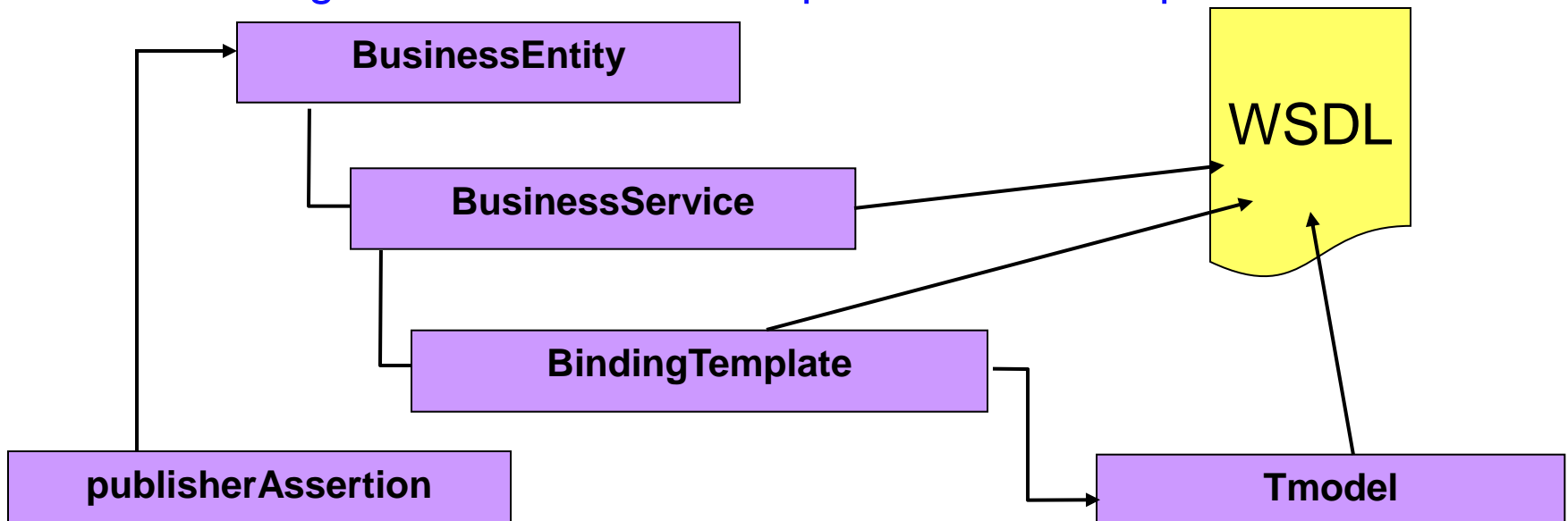


Accès à l'annuaire

- Accès direct par un navigateur Web
- Accès par un programme en utilisant une API
 - API vue comme un Web Service
 - Décrite en WSDL et Appelée avec SOAP
 - Fonctions de recherche (Inquiry API),
 - Recherche et consultation de données
 - Ex: find_business, get_serviceDetail
 - Fonctions de publications (Publishing API)
 - Création, Modification, Suppression de données
 - Ex: save_binding, delete_tmodel
 - Fonctions annexes
 - Réplication entre annuaires, Sécurisation des données

WSDL et UDDI

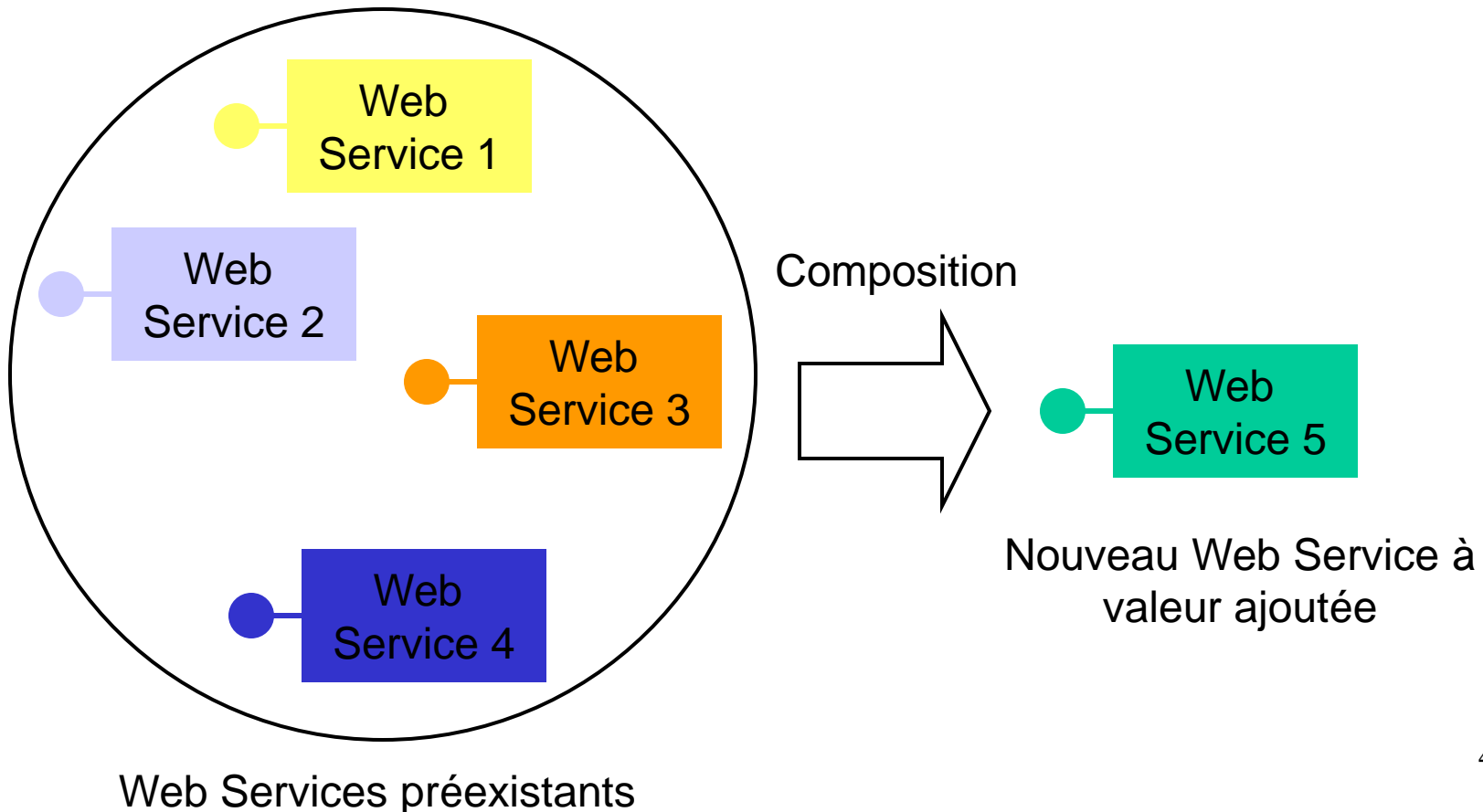
- Portée de UDDI non limitée aux Web Services
 - Accès à un service par téléphone, fax, email
- Mapping WSDL avec UDDI
 - Fragmentation de la description WSDL complet



- OASIS Technical Note, Using WSDL in a UDDI Registry

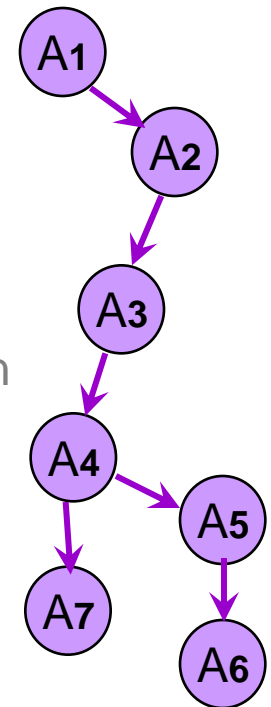
Composition de services

- Aggréger des Web Services élémentaires pour aboutir à un nouveau Web Service à valeur ajoutée



Processus métier

- Un processus métier est un enchaînement d'activités entre acteurs d'une entreprise pour aboutir à un service donné.
 - Acteurs : humains, applications/services, processus métier
- Business Process Management (BPM)
 - Cycle de vie des processus métier
 - Modélisation, Déploiement, Exploitation, Administration
 - *Business Process Management Initiative* (BPMI)
- BPM & Web Services
 - W3C Web Service Choreography Working Group
 - BEA, Intalio, SAP, Sun
 - OASIS Web Services Business Process Execution Language Technical Committee
 - IBM, Microsoft

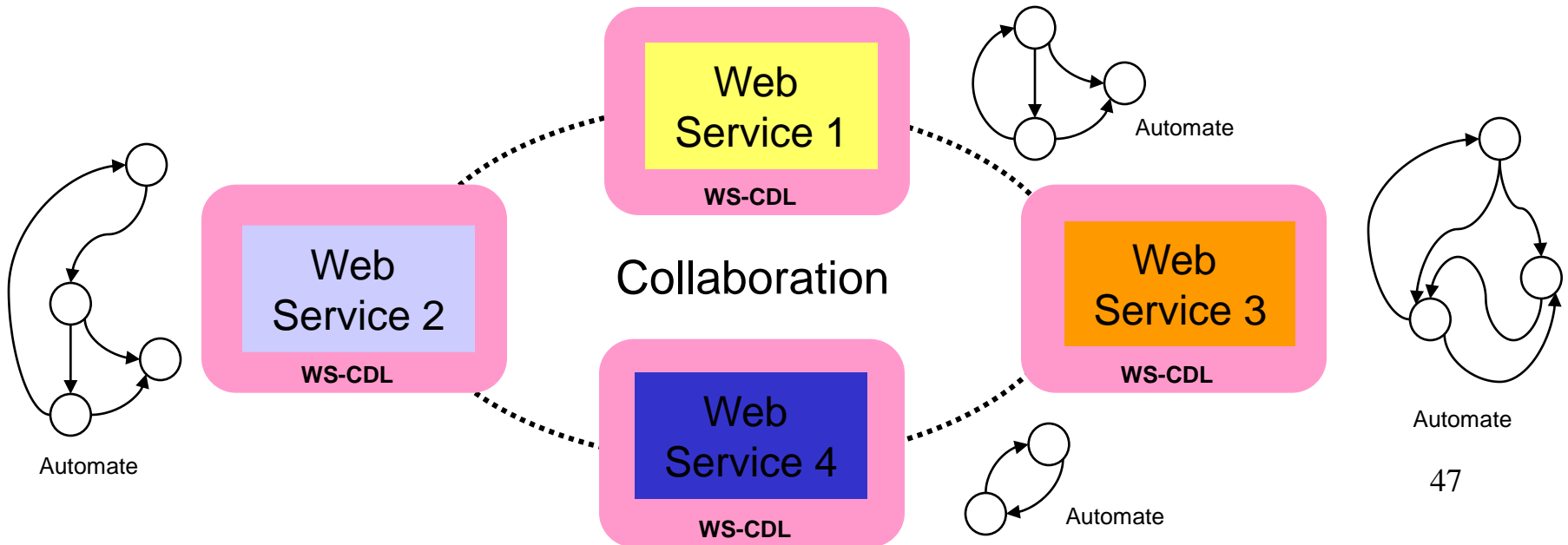


Processus Métier

Modélisation d'un processus métier (1/2)

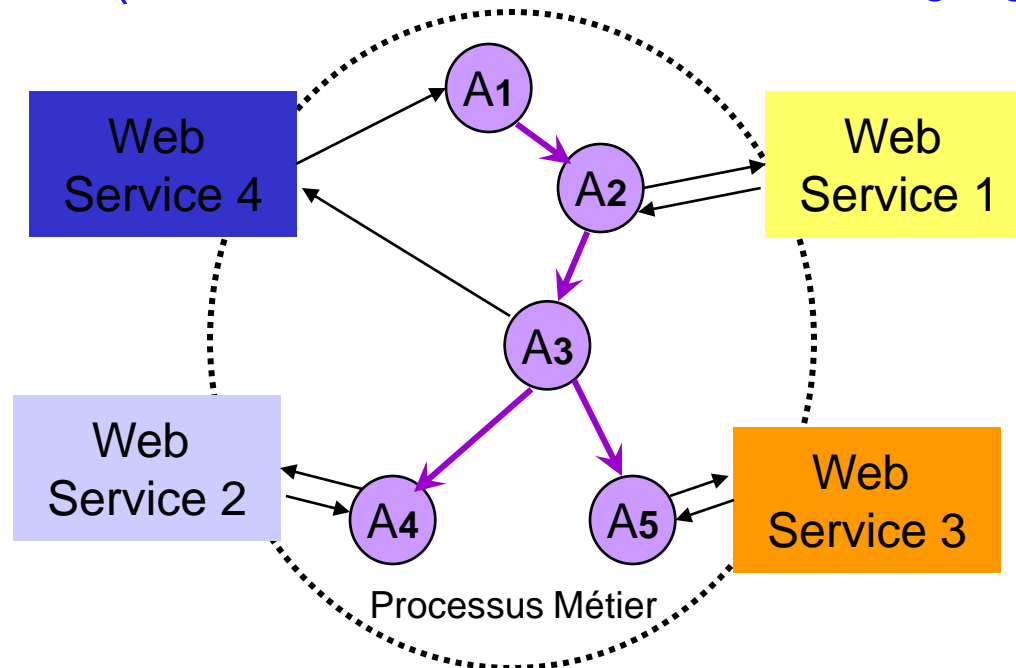
Chorégraphie

- Vision distribuée
- Collaboration publique entre participants
 - Définition de contraintes sur les interfaces
- Non exécutable tel quel (transformation en processus BPML)
- WS-CDL (*WS-Choreography Description Language*)



Modélisation d'un processus métier (2/2)

- Orchestration
 - Vision centralisée
 - Implantation privée des participants
 - Exécutable
 - WS-BPEL (*WS-Business Process Execution Language*)



Orchestration avec WS-BPEL

- Description XML d'un processus métier
- Référence les descriptions WSDL
- Activité de base
 - Invoquer un service Web <invoke>
 - Recevoir une invocation <receive>
 - Répondre à une invocation <reply>
- Activité structurée
 - Séquence ordonnée <sequence>
 - Branchement conditionnel <switch>, <if>
 - Boucle <while>
 - Exécution en parallèle <flow>
- Spécification du consortium industriel OASIS

Exemple : WS-BPEL

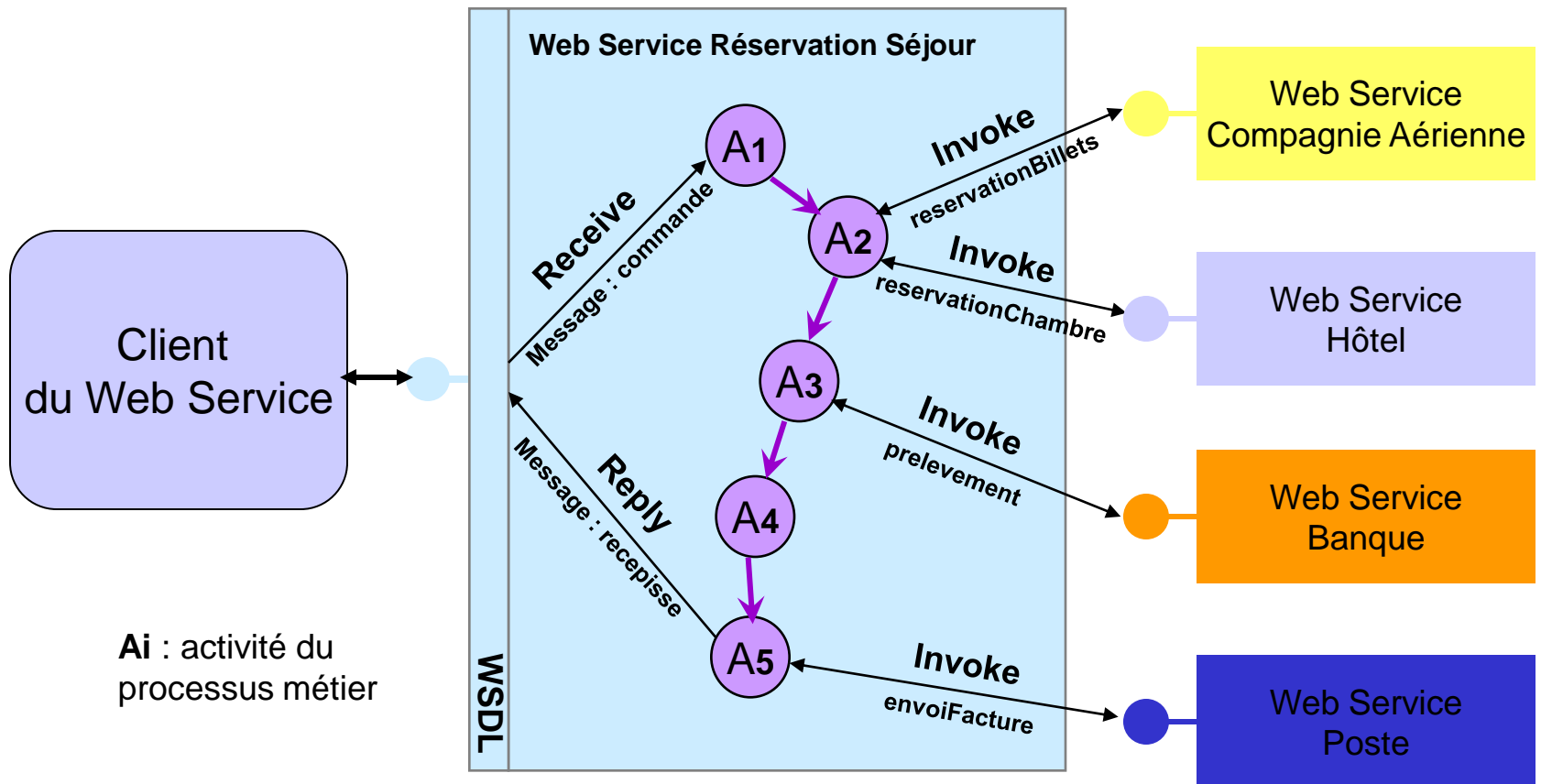


Plate-formes opérationnelles

- Evolution des plate-formes de développement d'applications d'entreprises basées sur des composants pour offrir un support pour les Web Services
- Plate-forme Java/J2EE
 - Enrichissement de la spécification par un ensemble d'interfaces (API) pour le support aux Web Services
 - Implantation Java libre d'où plusieurs éditeurs
 - Sun (Sun One),
 - IBM (WebSphere),
 - BEA (Weblogic) ...
 - Open Source : Apache (Implémentation SOAP avec Axis)
- Plate-forme .NET
 - Plate-forme "produit" contrôlé par Microsoft

Plate-formes J2EE

- Support des Web Services à travers un ensemble d'interfaces (API) telles que :
 - Java API for XML Processing (JAXP)
 - Manipulation des documents XML
 - Java API for XML Messaging (JAXM)
 - Envoi et réception de messages SOAP
 - Java API for XML-based RPC (JAX-RPC)
 - Envoi et réception de messages SOAP en mode RPC
 - Java Architecture for XML Data Binding (JAXB)
 - Sérialisation d'objets java en documents XML
 - Java API for XML Registries (JAXR)
 - Accès à des annuaires pour Web Services
- Implantation de Web Services typiquement à partir d'un Enterprise JavaBean (EJB), voire directement à partir d'une application Java.

Architecture des Web Services sous J2EE

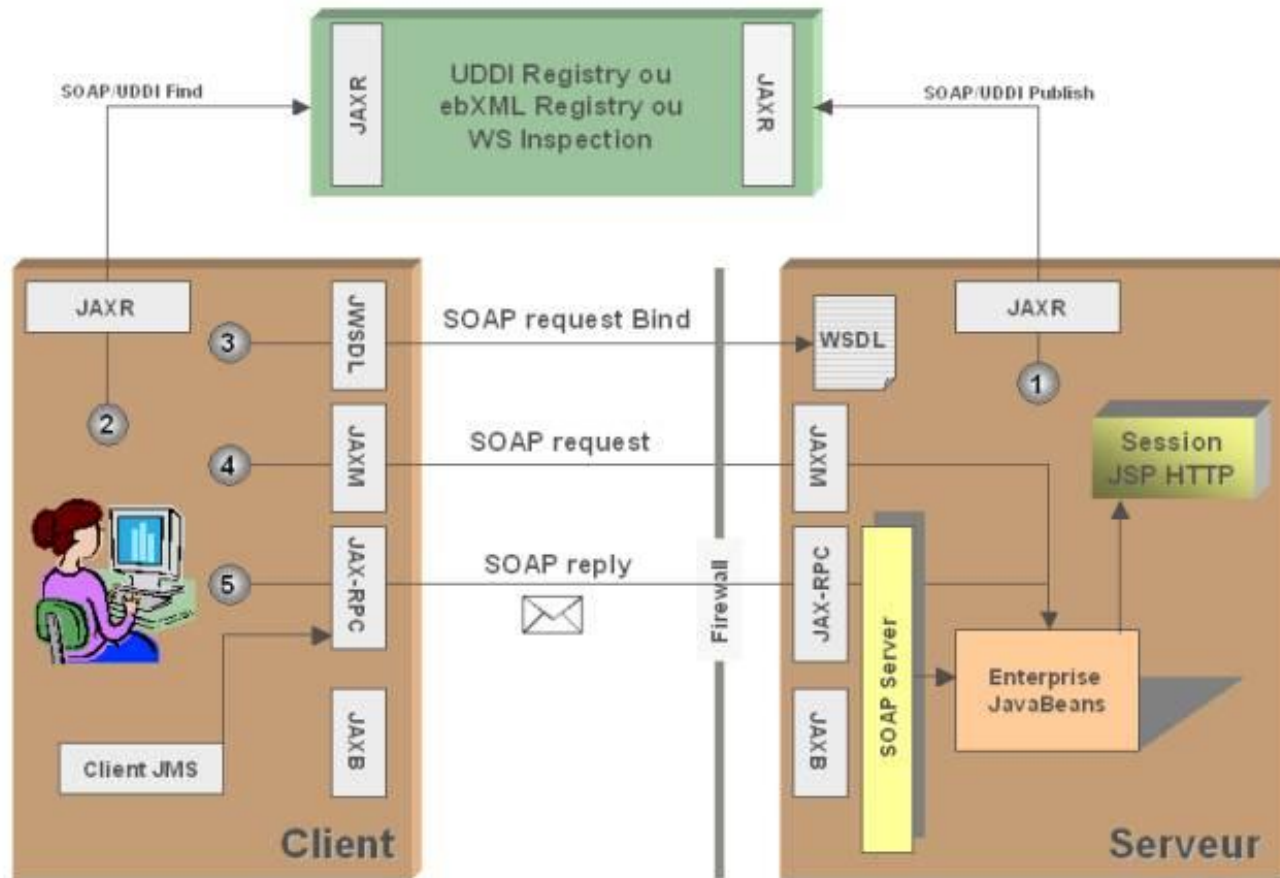
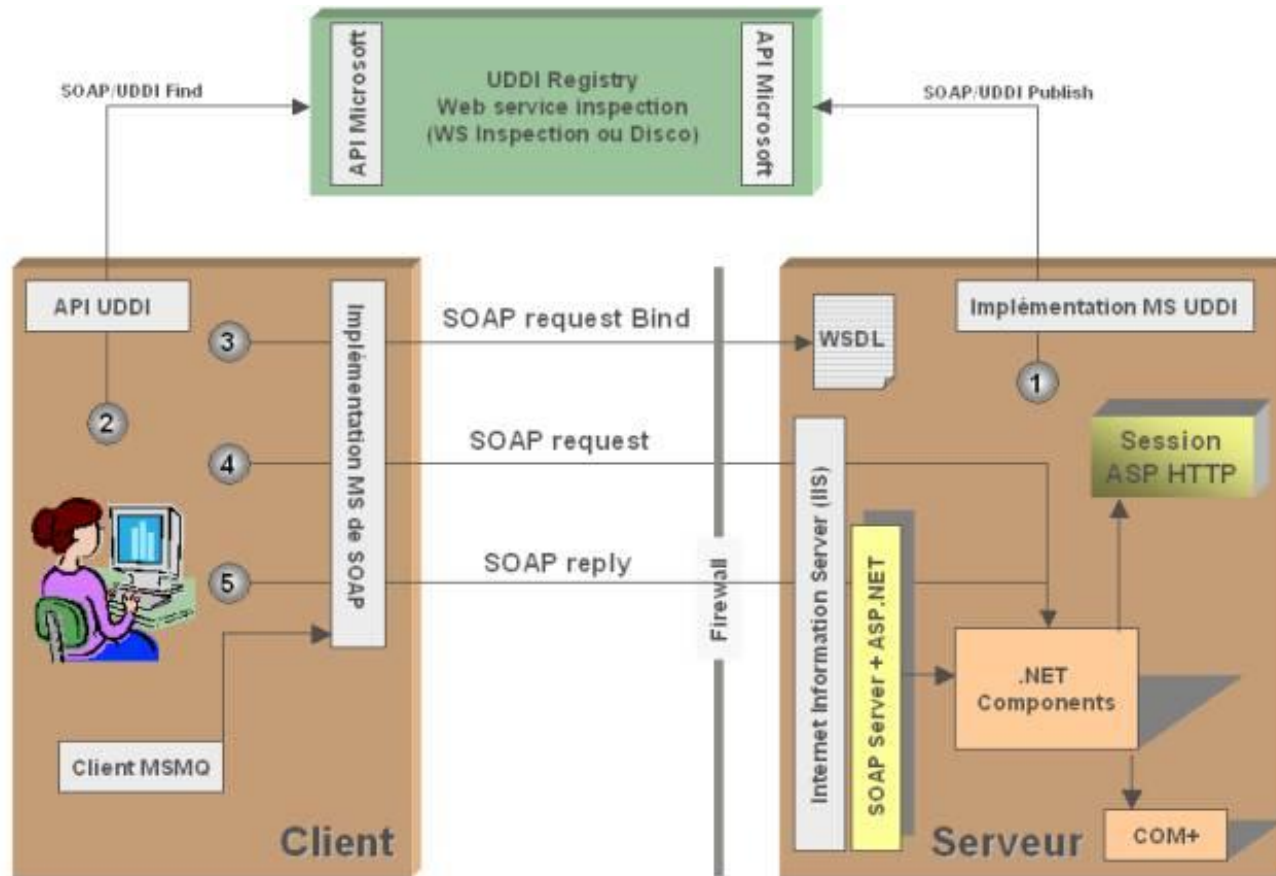


Plate-forme .NET

- Le support aux Web Services est directement intégré à la plate-forme de Microsoft.
 - Evolution de l'environnement d'exécution d'application Web (ASP) aux Web Services (ASP.NET)
- Implantation des Web Services à travers des composants .NET (incluant les composants COM/COM+)
- Implantation possible sous plusieurs langages de programmation
 - C#, C++, Visual Basic.NET, Jscript.NET, J# ...

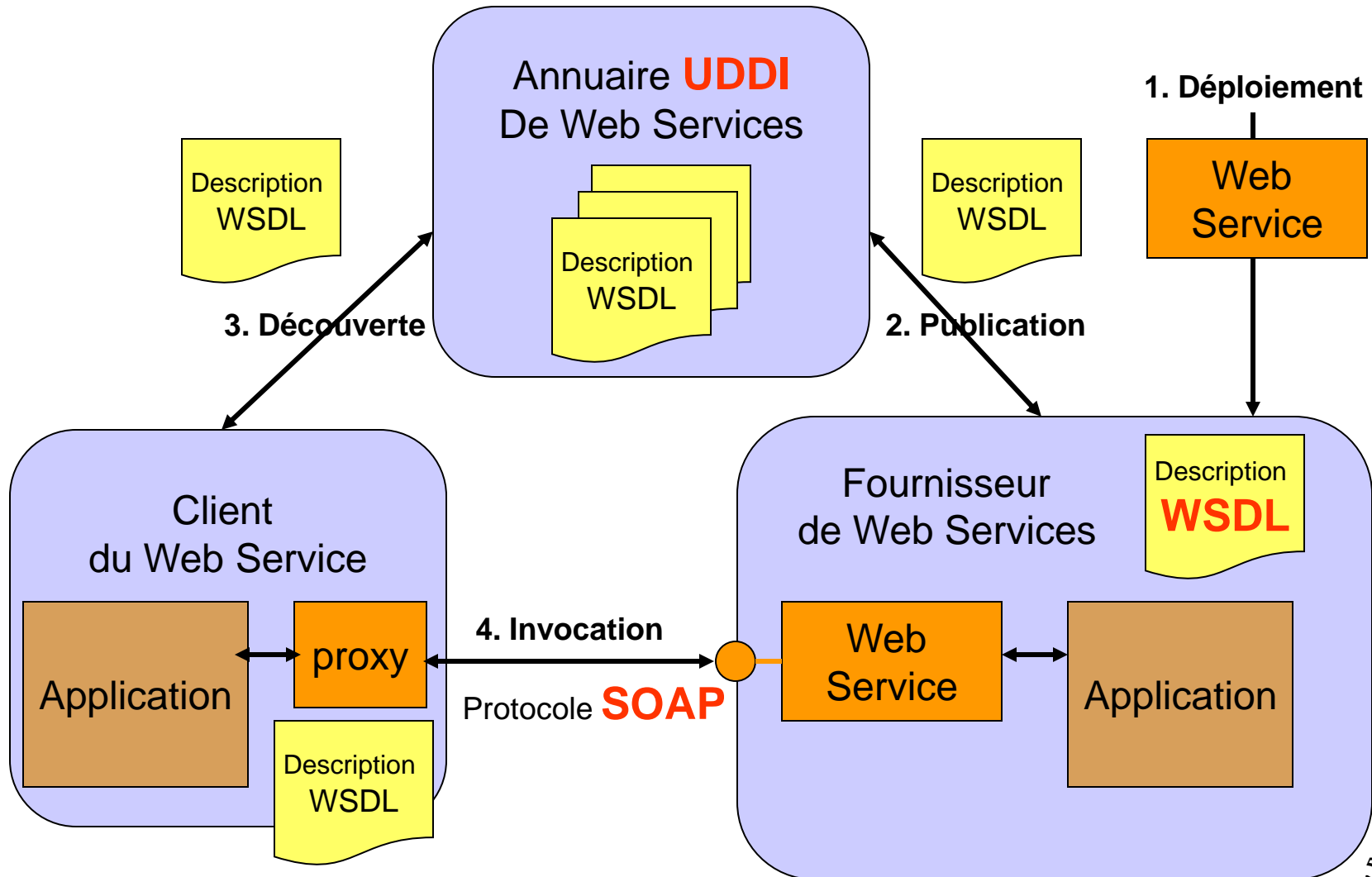
Architecture des Web Services sous Microsoft .NET



Bilan (1/2)

- Un Web Service est une application modulaire :
 - Mise à disposition sur l'Internet ou sur un réseau privé (Intranet),
 - Auto-descriptive (**WSDL**), publiable (**UDDI**) et accessible (**SOAP**) en utilisant le langage XML et les protocoles standards du Web,
 - Indépendante du système d'exploitation et du langage de programmation,
 - Visant à exposer une ou plusieurs fonctionnalités (souvent commerciale) d'une entreprise.
- Un ensemble de Web Services élémentaires peut être combiné (**WS-BPEL**) pour aboutir à un Web Service à valeur ajoutée.

Bilan (2/2)



Bibliographie

- Web Services Essentials, Ethan Cerami, Editions O'Reilly
- Services Web avec J2EE et .NET : Conception et Implémentations, Libero Maesano, Christian Bernard, Xavier Le Galles, Editions Eyrolles
- Les Web Services : Techniques, Démarches et Outils, XML, WSDL, SOAP, UDDI, Rosetta, UML, Hubert Kadima, Valérie Monfort, Editions Dunod
- Cours de Didier Donsez et Robert Voyer