Les Web services

Introduction

Définition du W3C

« Un Web Service est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet »

Introduction

Un web service est une technologie permettant à des applications de communiquer entre
elles :

- en s’appuyant sur les standards du web (HTTP, XML)
- indépendamment de l’architecture sur lesquelles elles sont implémentées
- en échangeant des documents sous le format XML

But des services web

Fournir une architecture générale pour les applications réparties sur internet :

inter-opérables :

- basé sur des standards ouverts
- sans composant spécifique à un langage ou un système d'exploitation

faiblement couplées :

- limiter au maximum les contraintes imposées sur le modèle de programmation des différents éléments de l'application
- par exemple ne pas imposer un modèle objet

supportant la montée en charge :

- par exemple en n'imposant pas un modèle de type RPC

etc.

Exemples de services existants

Google (http://www.google.com/apis/) :
accès gratuit mais limité (1000 requêtes par jour après enregistrement)
propose trois opérations :
recherche
obtention d'une page depuis le cache
correction orthographique
Amazon
(http://associates.amazon.com/exec/panama/associates/join/developer/resources.html)
accès gratuit mais limité (1 requête par seconde après enregistrement)
propose recherche et gestion d'un panier d'achats
bien d'autres ! (cf http://www.xmethods.com/ par exemple)

Architecture de base

Trois acteurs :

le fournisseur de service (service provider) :

définit le service
publie sa description dans l'annuaire
réalise les opérations

l'annuaire (discovery agency) :

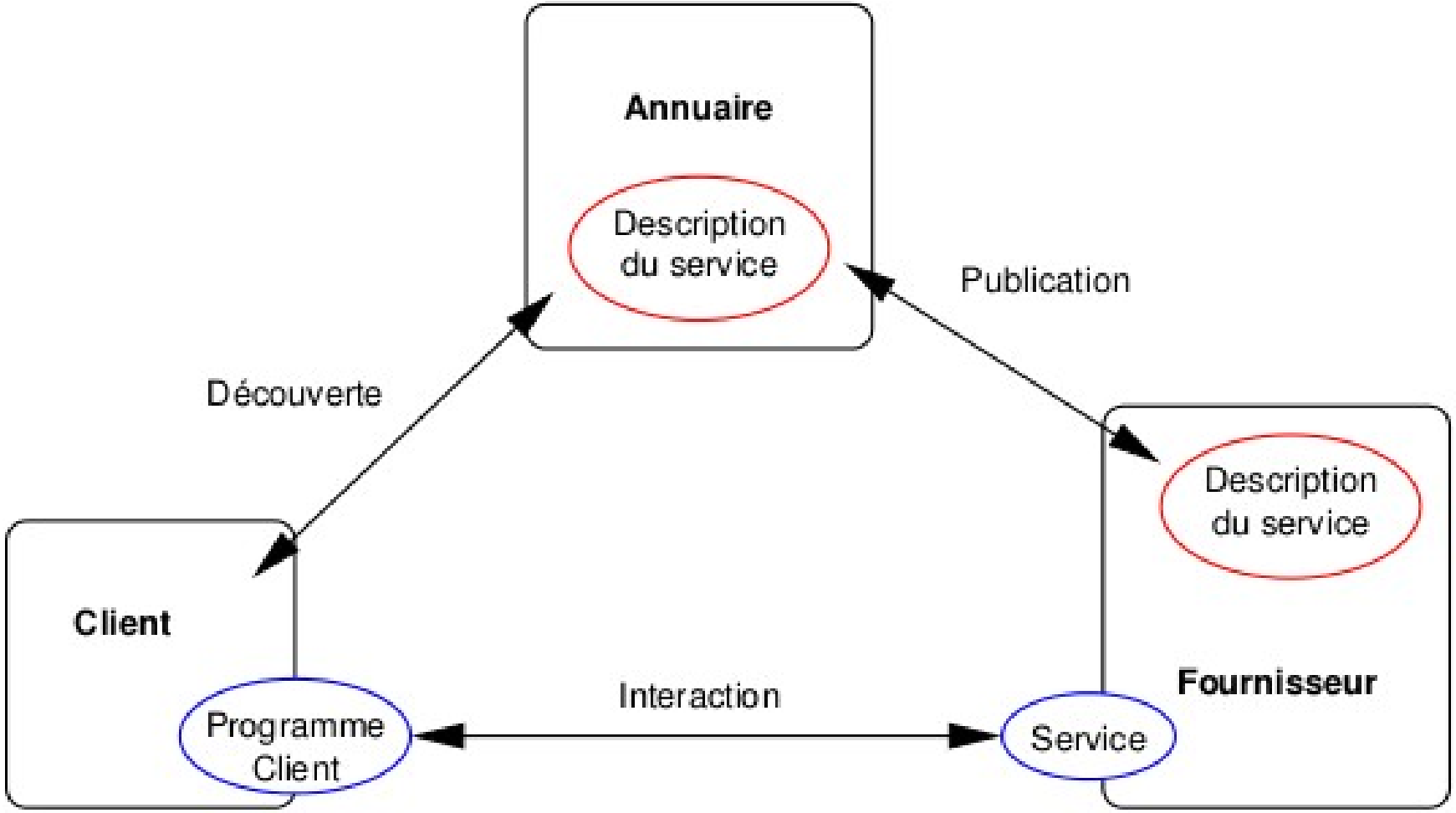
reçoit et enregistre les descriptions de services publiées par les fournisseurs

reçoit et répond aux recherches de services lancées par les clients

le client (service requestor) :

obtient la description du service grâce à l'annuaire
utilise le service

Architecture de base



Briques de l'architecture de base

SOAP (Simple Object Access Protocol) :

cadre général permettant l'échange de données
structurées au format XML
protocole de transport de ces données basé sur HTTP
Le protocole SOAP définit:
-Une enveloppe;
- une mise en œuvre sur HTTP (HTTP Extension Framework);
-Un ensemble de règles de codages;
-Un fonctionnement en modèle client / serveur (RPC)

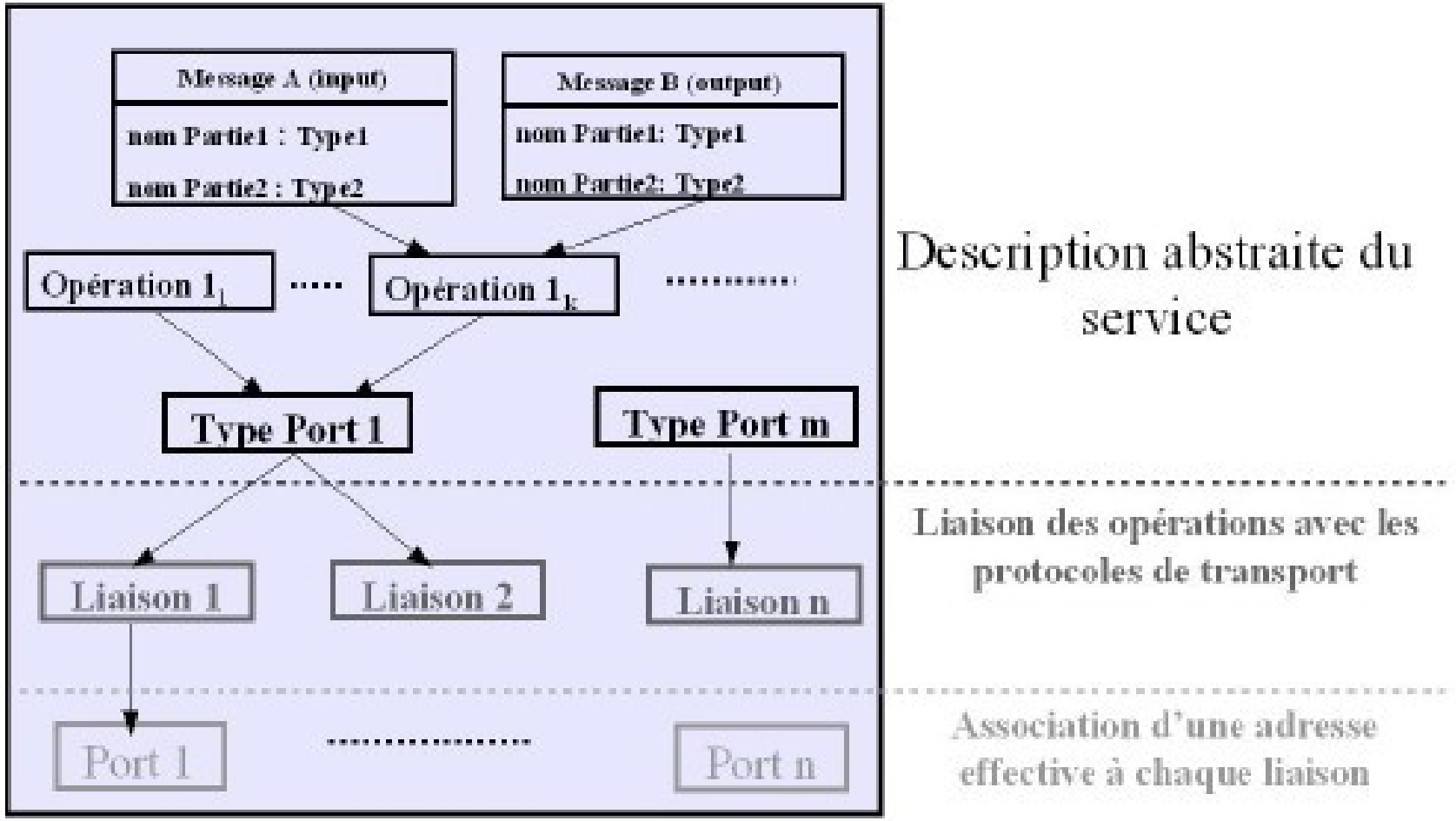
WSDL (Web Services Description Language) :

dialecte XML permettant de décrire un service web

UDDI (Universal Data Description Interface) :

annuaire permettant d'enregistrer et de rechercher des
descriptions de services web

Structure d'un fichier WSDL



Le langage de description d'interface(WSDL)

C'est un langage de définition des interfaces des services (le contrat)

- **Donc d'une grande importance**

◆ Il représente la définition d'un services Web vue par le fournisseur

◆ Il doit contenir toutes les information nécessaire au client pour consommer le service (auto-suffisant)

◆ Il joue exactement le même rôle que IDL sauf qu'il n'exprime pas des objet distant mais un service

Décrire les services
comme un ensemble d'opérations et de messages abstraits relié (bind) à des protocoles et des serveurs réseaux

Éléments d'une définition WSDL

<types>

• Contient les définition de types utilisant un système de typage (comme XSD).

<message>

• Décrit les noms et types d'un ensemble de champs à transmettre

• Paramêtres d'une invocation, valeur du retour, ...

<porttype>

• Décrit un ensemble d'opérations. Chaque opération a zero ou un message en entrée, zero ou plusieurs message de sortie ou de fautes

<binding>

• Spécifie une liaison d'un <porttype> à un protocole concret (SOAP1.1, HTTP1.1, MIME, ...). Un porttype peut avoir plusieurs liaisons !

<port>

• Spécifie un point d'entrée (endpoint) comme la combinaison d'un <binding> et d'une adresse réseau.

<service>

• Une collection de points d'entrée (endpoint) relatifs.

Élément <types>

• **Contient les définition de types utilisant un système de typage (comme XSD).**

n Exemple

```
<!-- type defs -->
<types>
<xsd:schema targetNamespace="urn:xml-soap-address-demo"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<xsd:complexType name="phone">
<xsd:element name="areaCode" type="xsd:int"/>
<xsd:element name="exchange" type="xsd:string"/>
<xsd:element name="number" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="address">
<xsd:element name="streetNum" type="xsd:int"/>
<xsd:element name="streetName" type="xsd:string"/>
<xsd:element name="city" type="xsd:string"/>
<xsd:element name="state" type="xsd:string"/>
<xsd:element name="zip" type="xsd:int"/>
<xsd:element name="phoneNumber" type="typens:phone"/>
</xsd:complexType>
</xsd:schema>
</types>
```

Élément <message>

- **Décrit les noms et types d'un ensemble de champs à transmettre**
- **Paramêtres d'une invocation, valeur du retour, ...**

Exemple

```
<!-- message declns -->
<message name="AddEntryRequest">
<part name="name" type="xsd:string"/>
<part name="address" type="typens:address"/>
</message>
<message name="GetAddressFromNameRequest">
<part name="name" type="xsd:string"/>
</message>
<message name="GetAddressFromNameResponse">
<part name="address" type="typens:address"/>
</message>
```

Element <porttype>

Décrit un ensemble d'opérations.

Plusieurs types d'opérations

- **One-way**
  - **Le point d'entrée reçoit un message (<input>).**
- **Request-response**
  - **Le point d'entrée reçoit un message (<input>) et retourne un message corrélé (<output>) ou un ou plusieurs messages de faute (<fault>).**
- **Solicit-response**

• Le point d'entrée envoie un message (<ouput>) et recoit un message corrélé (<input>) ou un ou plusieurs messages de faute (<fault>).

-Binding HTTP : 2 requêtes HTTP par exemple

• Notification

- Le point d'entrée envoie un message de notification (<ouput>)

Paramêtres

• Les champs des messages constituent les paramêtres (in,out, inout) des opérations

Element <porttype>

Exemple

```
<!-- port type declns -->
<portType name="AddressBook">
<!– One way operation -->
<operation name="addEntry">
<input message="AddEntryRequest"/>
</operation>
<!– Request-Response operation -->
<operation name="getAddressFromName">
<input message="GetAddressFromNameRequest"/>
<output message="GetAddressFromNameResponse"/>
</operation>
</portType>
```

Élément <binding>

- Spécifie une liaison d'un <porttype> à un protocole concret (SOAP1.1, HTTP GET/POST, MIME, ...).
- Un porttype peut avoir plusieurs liaisons !

Exemple de binding sur SOAP et HTTP

```
<!-- binding declns -->
<binding name="AddressBookSOAPBinding" type="AddressBook">
<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="addEntry">
<soap:operation soapAction=""/>
<input><soap:body use="encoded"
namespace="urn:AddressFetcher2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </input>
<output><soap:body use="encoded"
namespace="urn:AddressFetcher2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
</operation>
```

Élément <binding>

```
<operation name="getAddressFromName">
<soap:operation soapAction=""/>
<input><soap:body use="encoded"
namespace="urn:AddressFetcher2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
<output><soap:body use="encoded"
namespace="urn:AddressFetcher2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></
output>
</operation>
</binding>
```

Élément <service>

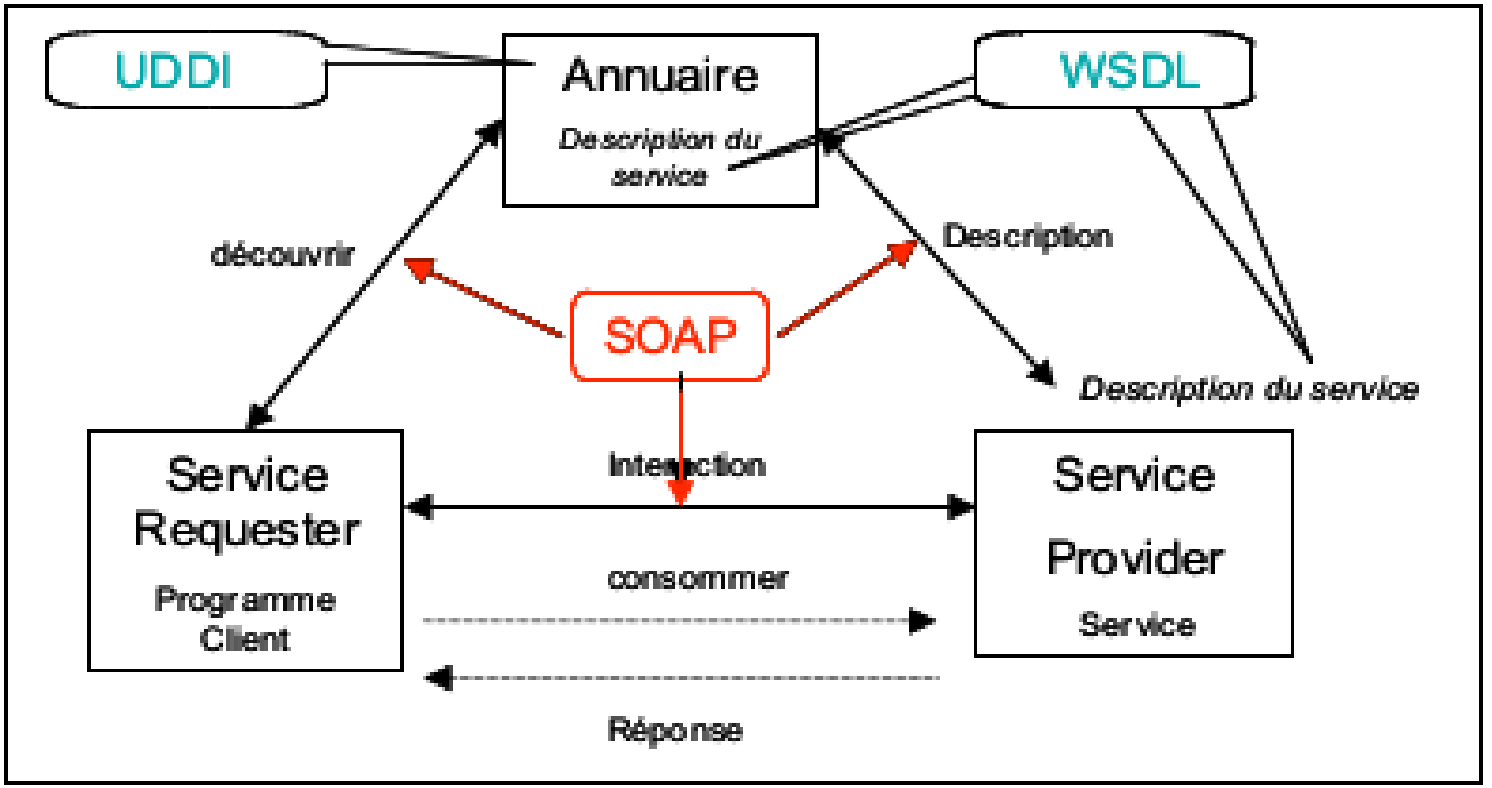
- **Une collection de points d'entrée (endpoint) relatifs**

Exemple

```
<?xml version="1.0" ?>
<definitions name="urn:AddressFetcher"
targetNamespace="urn:AddressFetcher2"
xmlns:typens="urn:xml-soap-address-demo"
xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
...
<!-- service decln -->
<service name="AddressBookService">
<port name="AddressBook" binding="AddressBookSOAPBinding">
<soap:address
location="http://www.mycomp.com/soap/servlet/rpcrouter"/>
</port>
</service>
</definitions>
```

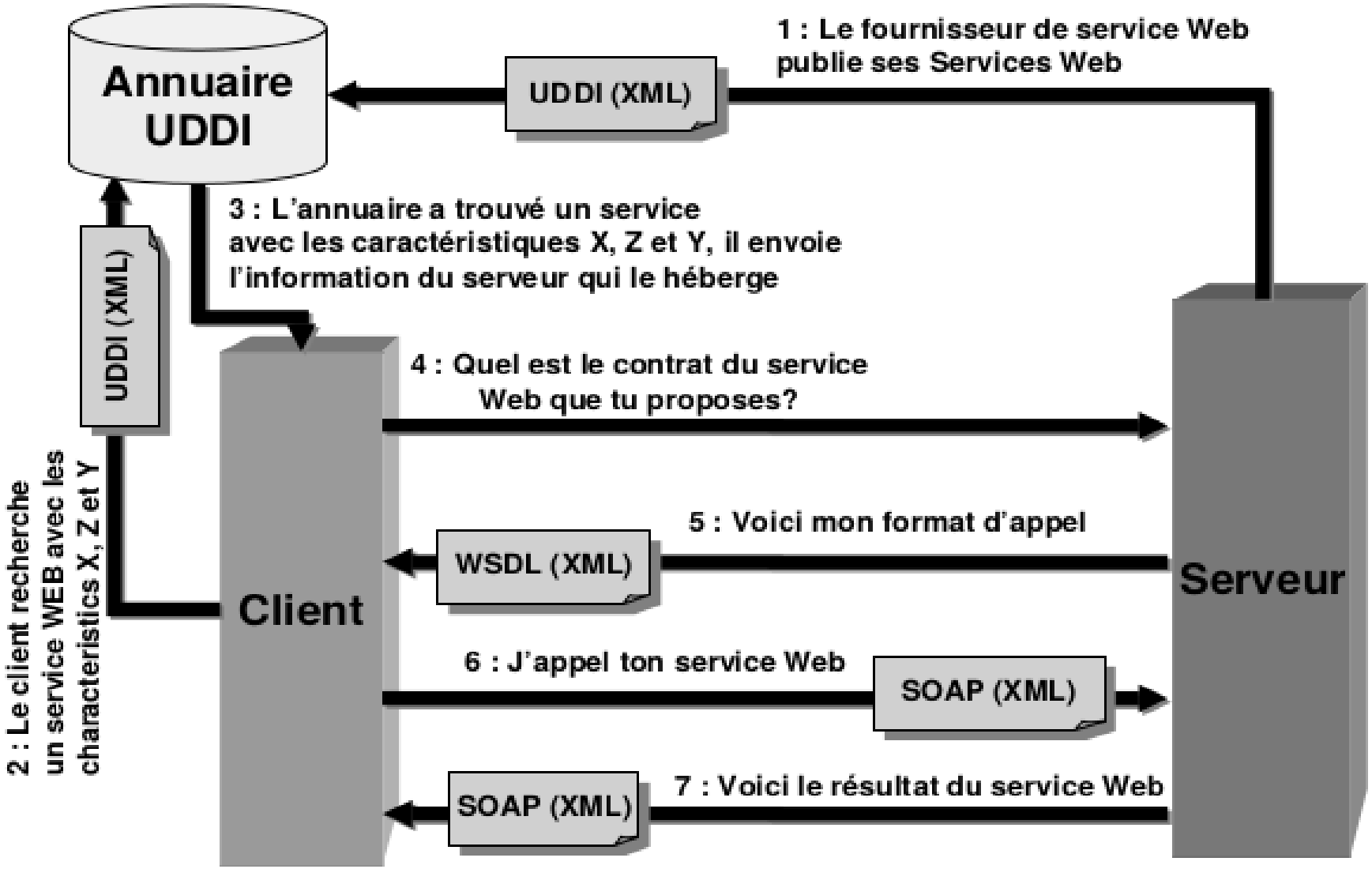
Architecture orientée service

Une architecture orientée services (notée SOA pour Services Oriented Architecture) est une architecture logicielle s'appuyant sur un ensemble de services simples.
Lorsque l'architecture SOA s'appuie sur des web services, on parle alors de WSOA, pour Web
Servic



Architecture WSOA

Résumé d'un Services Web



Introduction

Introduction

Introduction

Introduction