

Les Web Services

Issam Rabhi & Patrice laurençot

21/01/2011

Les Web Services

Plan

2

- **Concepts Généraux**
 - Définition
 - Pourquoi?
 - Technologies
 - Mécanismes
- **WSDL**
- **SOAP & REST**
- **Composition**
- **Sécurité**

Web Services – Définition

3

- Un service Web est un «objet» accessible en utilisant les protocoles standard d'Internet
- Un «composant» fournissant des données et des services à d'autres applications
- Ils s'appuient sur un ensemble de standards (WSDL, SOAP, XML-RPC et UDDI)

.....

Web Services – Définition

4

- Un service Web est une application logicielle identifiée par un URI dont les interfaces et les liaisons sont définies, décrites et découvertes en XML et qui supporte une interaction directe avec les autres applications logicielles en utilisant des messages XML via un protocole Internet

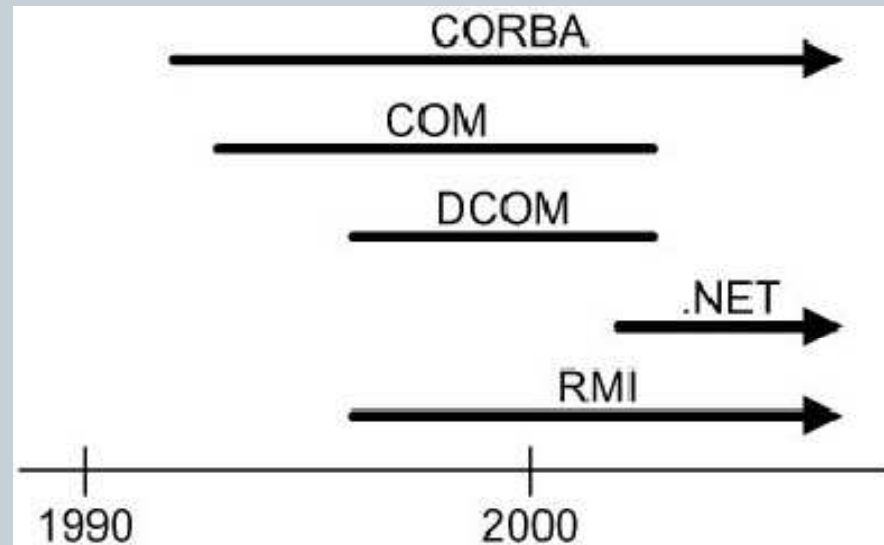
Web Services – Pourquoi?

5

- Évolution des solutions Middleware
 - RPC : Remote Procedure Call
 - MOM : Message Oriented Middleware (MQ Series, JMS)
 - Objet Distribué (COM, DCOM, CORBA, RMI, EJB)
 - Base de Données Orientées Middleware

Web Services – Pourquoi?

6



Historique des protocoles d'interopérabilité

=> Ces protocoles ne s'intègrent pas facilement avec le monde internet.

Web Services – Pourquoi?

7

- Avoir des composants utilisables dans une architecture orientée service (SOA)
- Facilité l'interopérabilité entre des applications développées pour des plates-formes et dans des langages différents
- Permettre « d'ouvrir » des applications existantes sur le monde Internet

Web Services : technologies

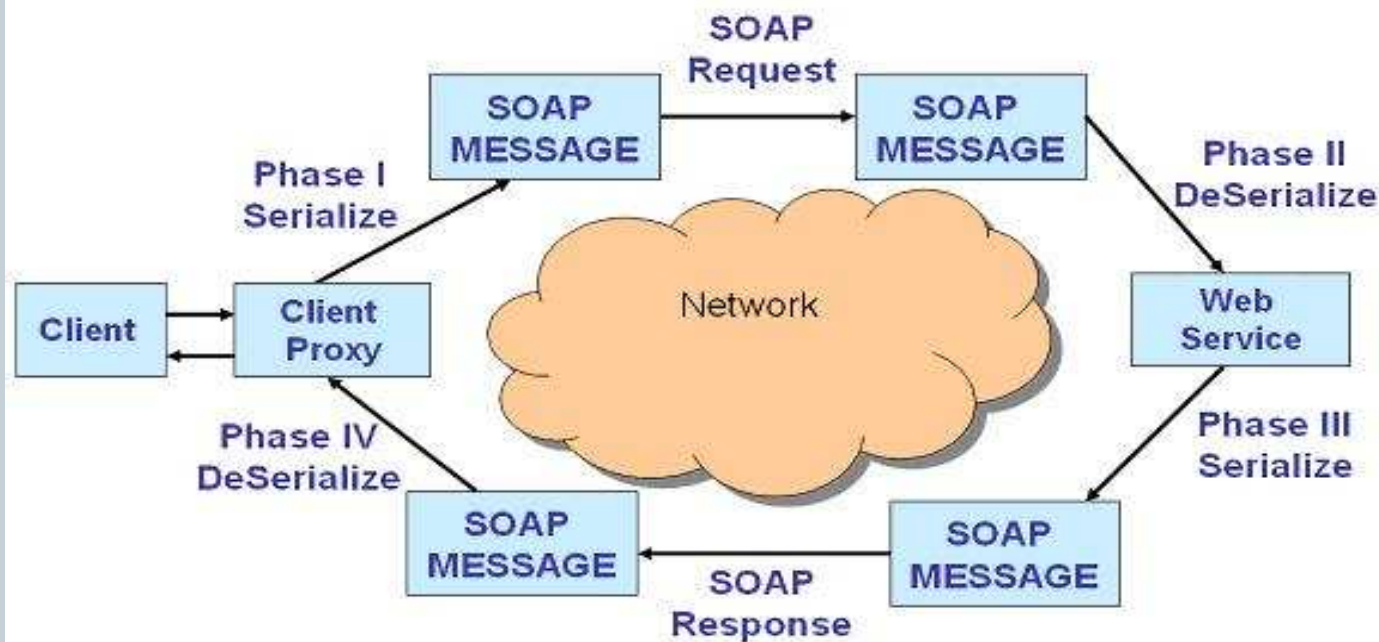
8

- **XML** (eXtensible Markup Language)
 - Échange de messages XML entre client et serveur
 - Lisible, structuré
- **HTTP, SMTP...**
 - Réutilisation des standards usuels d'Internet
- **SOAP** (Simple Object Access Protocol)
 - Protocole définissant les échanges XML entre entités
- **WSDL** (Web Services Description Language)
 - Langage de description technique des services web proposés
- **UDDI** (Universal Description Discovery and Integration)
 - Annuaire des services web disponibles et des fournisseurs de ces services

Web Services : technologies

9

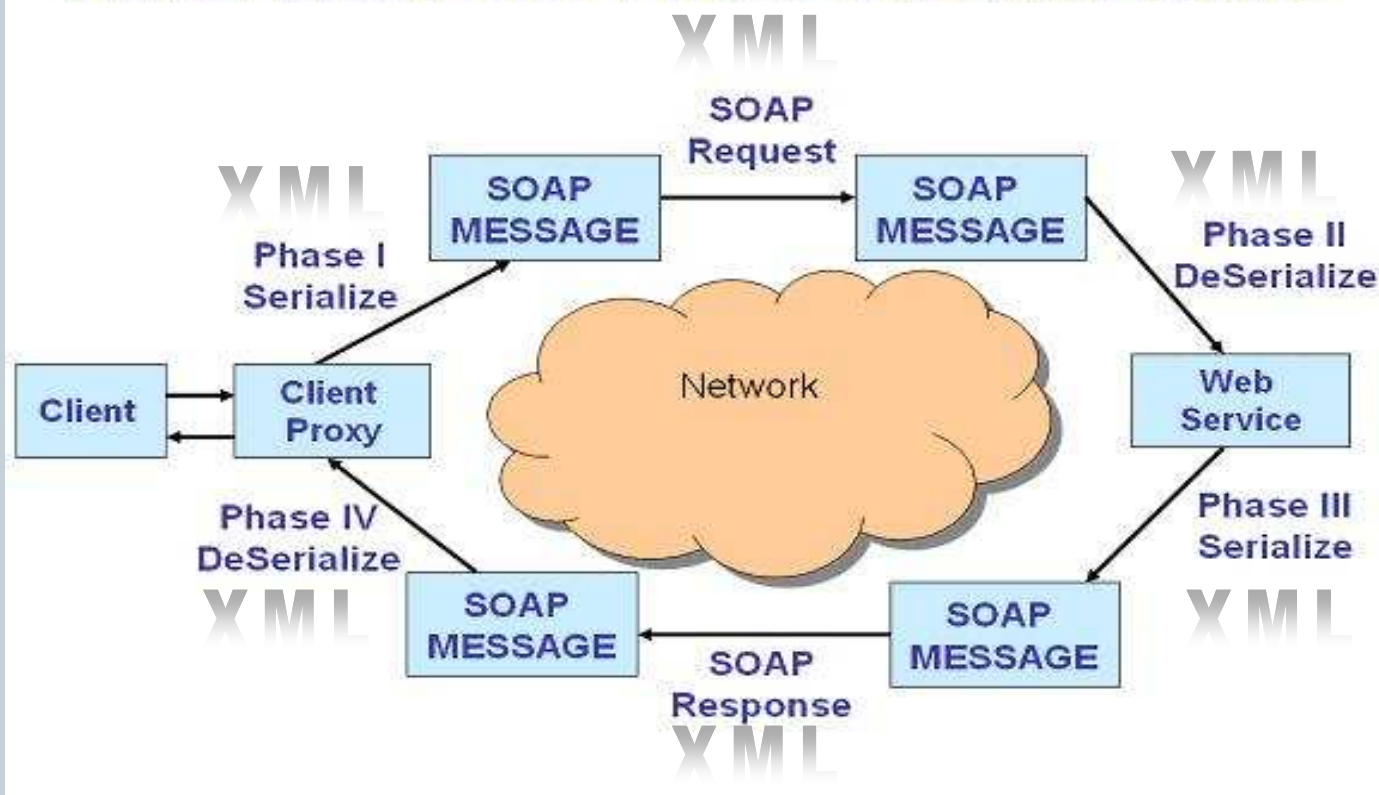
XML Web Services Architecture



Web Services : technologies

10

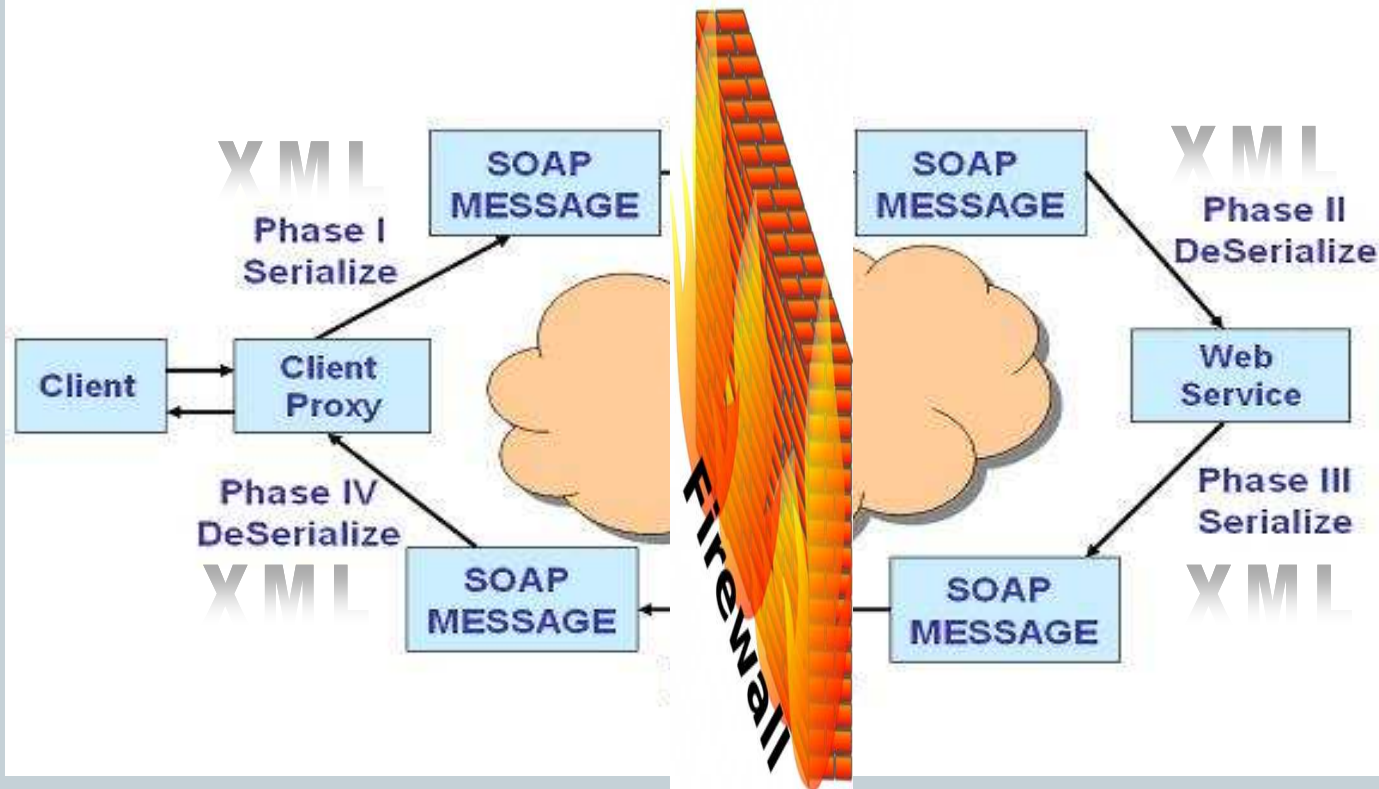
XML Web Services Architecture



Web Services : technologies

11

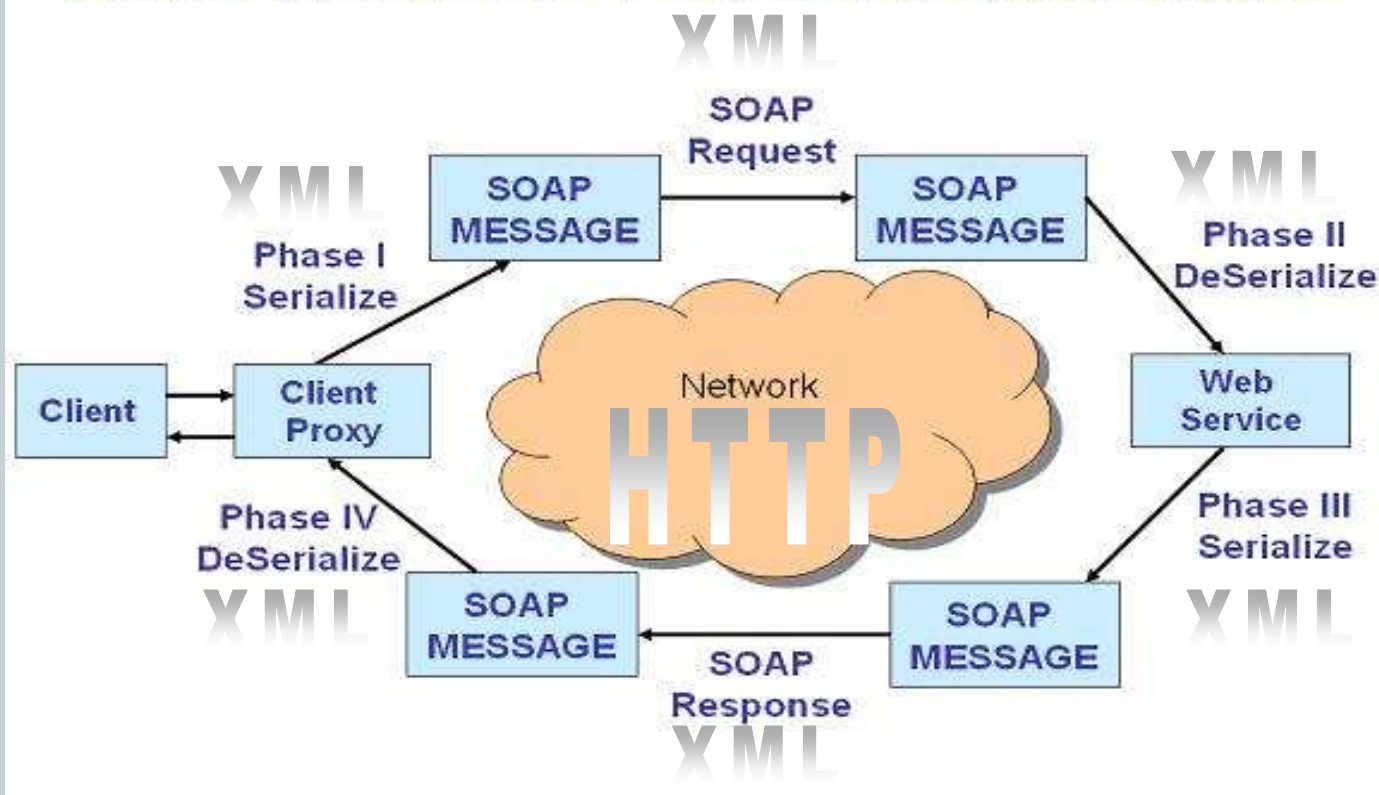
XML Web Services Architecture



Web Services : technologies

12

XML Web Services Architecture



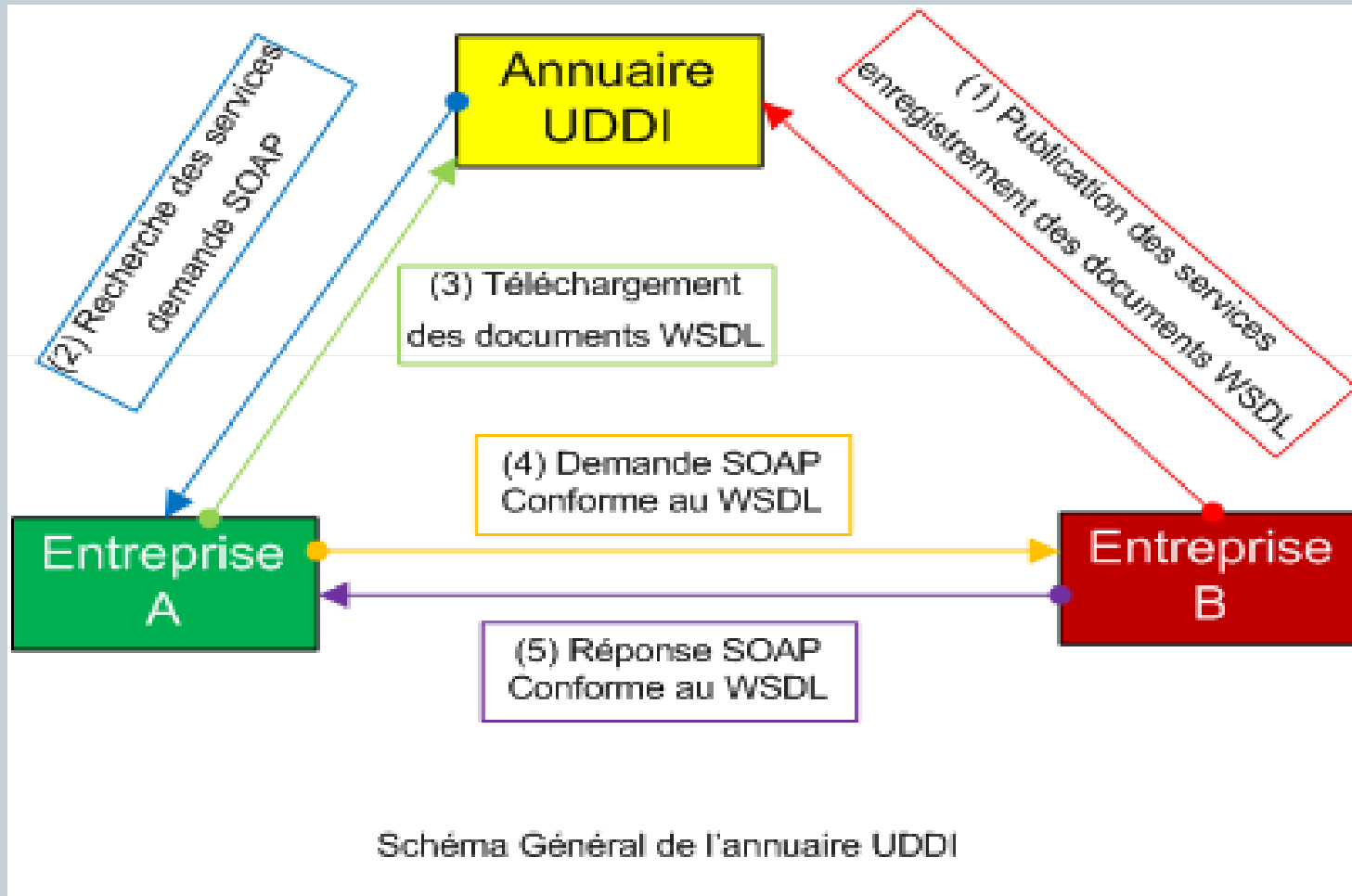
Web Services : technologies

13

- **XML** est utilisée pour l'échange des données et messages.
- Cette technologie permet de sérialiser/désérialiser les données (objet, fichier,...)
- **SOAP** permet de passer les pare-feux en utilisant le protocole HTTP.
- Les développeurs ont l'embarras du choix en matière de langage de programmation : Java, C, C++, Perl, Python, C#, et/ou Visual Basic,

Web Services – mécanismes

14



Web Services – WSDL

15

- **WSDL : Web Service Description Language**
 - Langage XML dédié à la description de tous les éléments nécessaires pour interagir avec un service réseau (ou service web)
- **WSDL permet de décrire:**
 - un service réseau (service Web),
 - et comment l'invoker (localisation, paramètre, structure de retour, ..)
- **Objectif :**
 - Décrire les services comme un ensemble d'opérations et de messages abstraits relié à des protocoles et des serveurs réseaux
 - Cet objectif se rapproche d'IDL pour CORBA.

Web Services – WSDL

16

- Un document WSDL est donc associé à un Web Service afin d'en décrire l'ensemble des éléments qu'il expose vers le monde extérieur :
 - types de données (Simple ou avec XSD)
 - types de messages
 - liaisons avec le protocole de transport et le format des messages (SOAP 1.X, HTTP Get et HTTP Post, ou encore MIME)
- Les services sont vus comme des ensembles de ports réseaux

Web Services – WSDL

17

```
<?xml version="1.0" ?>
<definitions name="CarnetAdresse" targetNamespace="http://exemple.com/carnetAdresse.wsdl"
  xmlns:per="http://exemple.com/carnetAdresse.wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xsd:schema targetNamespace="http://exemple.com/personne.xsd"
      xmlns:xsd="http://www.w3.org/1999/XMLSchema">
      <xsd:complexType name="Personne">
        <xsd:element name="Nom" type="xsd:string">
        <xsd:element name="Prenom" type="xsd:string">
        <xsd:element name="Age" type="xsd:float">
      </complexType>
    </types>
    <message name="GetPersonne">
      <part name="body" type="Personne"/>
    </message>
    <message name="AddPersonne">
      <part name="Nom" type="xsd:string"/>
      <part name="Prenom" type="xsd:string"/>
      <part name="Age" type="xsd:float"/>
    </message>
    <portType name="CarnetAdresse">
      <operation name="AddPersonne">
        <input message="AddPersonneRequest"/>
      </operation>
      <operation name="GetPersonneParNom">
        <input message="GetPersonneParNom Request"/>
        <output message="GetPersonneParNom Response"/>
      </operation>
    </portType>
```

Web Services – WSDL

18

```
<binding type="CarnetAdresseBinding">
  <soap:binding
    transport="http://schemas.xmlsoap.org/soap/http"
    style="rpc" />
  <operation name="GetPersonneParNomOpe">
    <soap:operation soapAction="http://exemple.com/ GetPersonneParNom " />
    <input>
      <soap:body use="encoded"
        encodingStyle="schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded"
        encodingStyle="schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
<service name="CarnetAdresseService">
  <port name="CarnetAdressePort" binding="per:PersonnesBinding">
    <soap:address location="http://example.com/GetPersonneParNom"/>
  </port>
</service>
</definitions>
```

Web Services - SOAP

19

- SOAP (Simple Object Access Protocol) est un protocole [RPC](#), orienté objet, bâti sur [XML](#)
- Il permet l'invocation de méthode sur des objets distants grâce à l'échange de message XML
- Bien que le protocole de transport le plus commun soit http(s), il est possible de l'utiliser sur SMTP

Web Services – SOAP - Exemple

20

■ Requête HTTP invoquant une méthode *GetOrders* :

```
POST /Orders HTTP/1.1
Host: xxx.xxx.xxx.xxx
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "http://www.someorders.com/GetOrders"
```

<?xml version="1.0"?>

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Body>

<orders:GetOrders

xmlns:orders="http://www.someorders.com/orders">

<CustomerID>ALFKI</CustomerID>

<SalesRepID>85</SalesRepID>

</orders:GetOrders>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Web Services – SOAP - Exemple

21

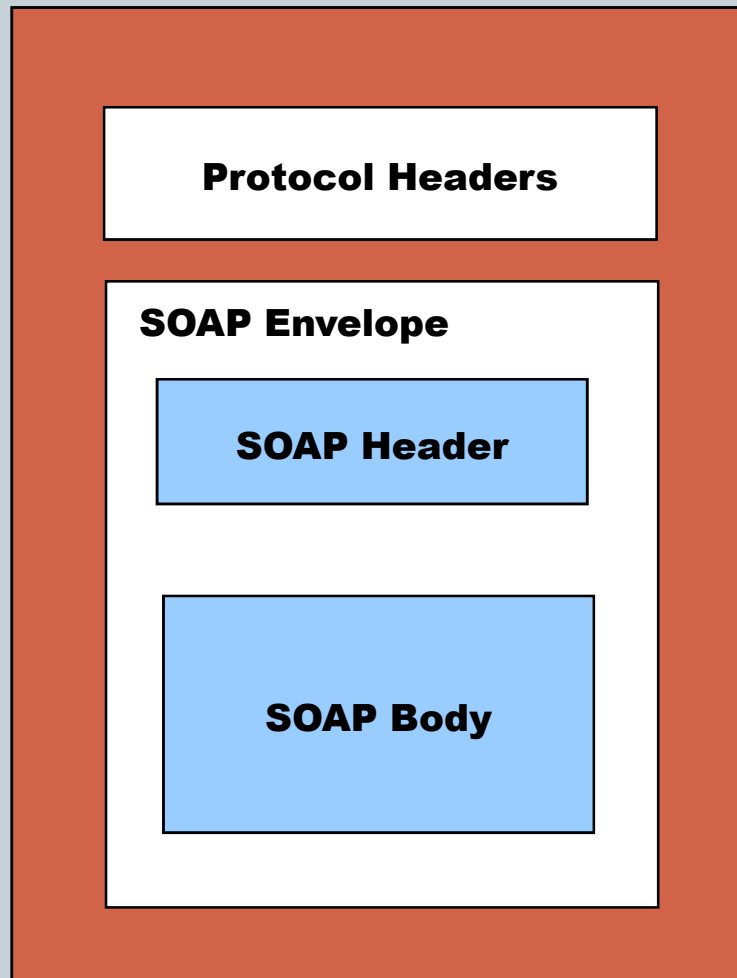
■ Réponse HTTP à la requête précédente :

```
HTTP/1.1 200 OK
MessageType: CallResponse
Content-Type: text/xml
```

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <orders:GetOrdersResponse
      xmlns:orders="http://www.someorders.com/orders">
      <.....
        données en retour sous forme XML
      .....>
    </orders:GetOrdersResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Web Services – SOAP

22



- Éléments d'un message SOAP :
 - Envelope
 - ✦ Élément pouvant contenir des déclarations d'espaces de noms ou des sous-éléments
 - Header (optionnel)
 - ✦ Permet des extensions telles que authentification, session...
 - Body (obligatoire)
 - ✦ Définit la méthode appelée, contient les paramètres
 - ✦ Peut contenir un élément Fault en cas d'erreur

Web Services – SOAP

23

- SOAP Header : Mécanisme d'extension du protocole SOAP
 - La balise Header est optionnelle
 - Si la balise Header est présente, elle doit être le premier fils de la balise Envelope
 - La balise Header contient des *entrées*
 - Une *entrée* est n'importe quelle balise incluse dans un namespace. Les *entrées* contenues dans la balise Header sont non applicatives.
 - Souvent, le Header est utilisé pour assurer l'authentification

Web Services – SOAP

24

- **SOAP Body : Le Body contient le message à échanger**
 - La balise Body est obligatoire
 - La balise Body doit être le premier fils de la balise Envelope (ou le deuxième si il existe une balise Header)
 - La balise Body contient des *entrées* qui sont des données applicatives.
 - Une *entrée* est n'importe quelle balise incluse optionnellement dans un namespace
 - Une *entrée* peut être une Fault.

Web Services – SOAP

25

- SOAP Fault : Balise permettant de signaler des cas d'erreur
- SOAP 1.1 : La balise Fault contient les balises suivantes:
 - Faultcode (Obligatoire): un code permettant d'identifier le type d'erreur.
 - Faultstring (Obligatoire): une explication en langage naturel.
 - Faultactor : une information identifiant l'initiateur de l'erreur.
 - Detail : Définition précise de l'erreur.

Web Services – SOAP

26

- Faultcode : 4 groupes de code d'erreur

- Client,
- Server,
- MustUnderstand,
- VersionMismatch

- Exemple SOAP 1.1

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  <s:Body>  
    <s:Fault>  
      <faultcode xmlns="">s:Client</faultcode>  
      <faultstring xml:lang="fr-FR" xmlns="">  
        Une opération invalide s'est produite.</faultstring>  
      </s:Fault>  
    </s:Body>  
  </s:Envelope>
```

Web Services – REST

27

- REST est l'acronyme de Representational State Transfer.
- Avec REST, l'information de base est appelée **ressource**. Toute information qui peut être nommée est une ressource : un article d'un journal, une photo, un service ou n'importe quel concept
- REST se base sur le protocole http:

Commandes

Commande	Description
GET	Requête de la ressource située à l'URL spécifiée
HEAD	Requête de l'en-tête de la ressource située à l'URL spécifiée
POST	Envoi de données au programme situé à l'URL spécifiée
PUT	Envoi de données à l'URL spécifiée
DELETE	Suppression de la ressource située à l'URL spécifiée

Web Services – REST

28

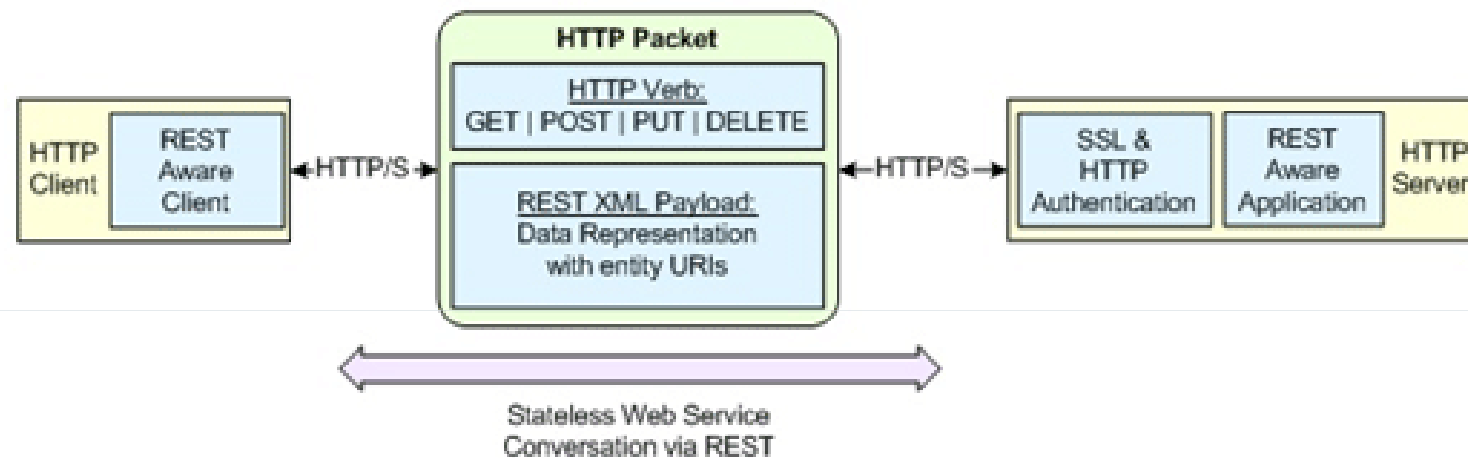


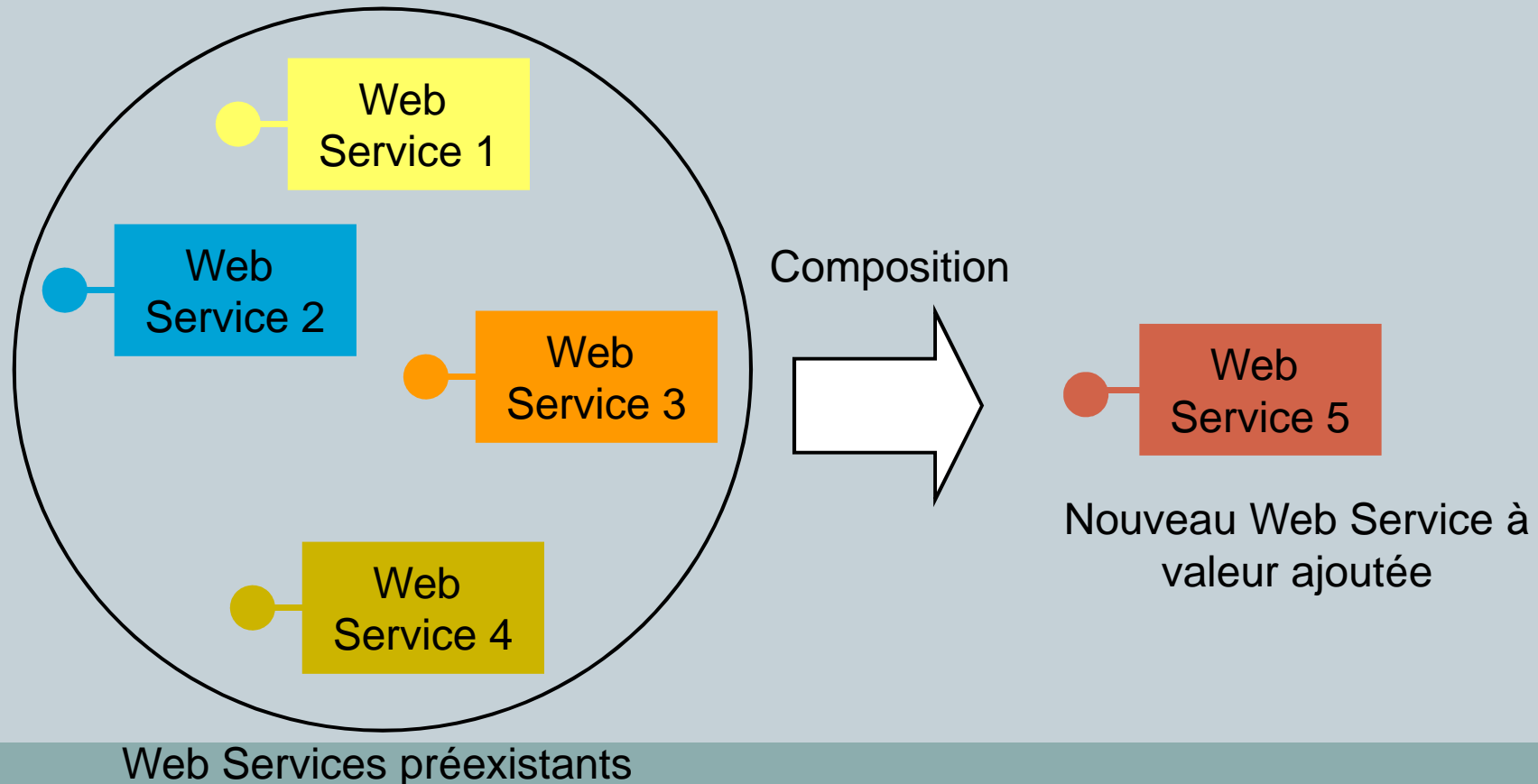
Fig 3 REST Web Service Conventions [6]

- *Plain XML is the data representation format.*
- *HTTP is the transfer protocol.*
- *HTTP's GET, POST, PUT, and DELETE are the access/manipulation verbs.*
- *URIs point to individual data records (such as customer info).*
- *HTTP authentication and SSL provide security.*

Web Services – Composition

29

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

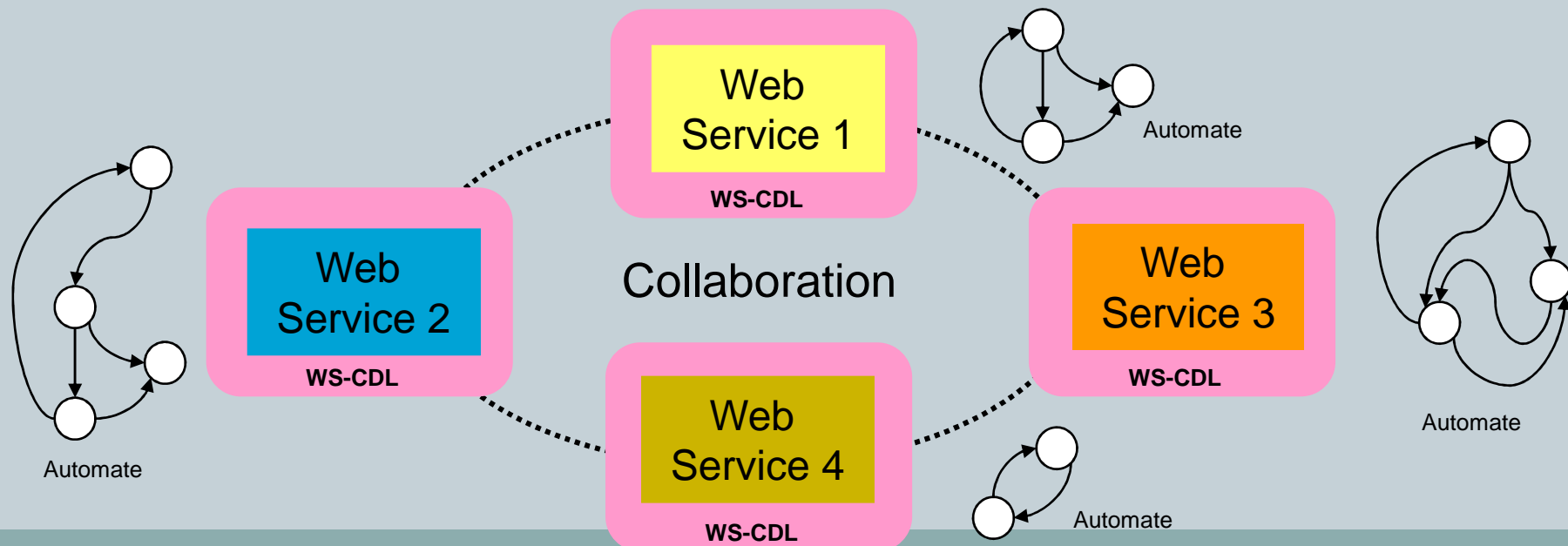


Web Services – Composition

30

- Choréographie

- Vision distribuée
- Collaboration publique entre participant
- WS-CDL (*WS-Choreography Description Language*)

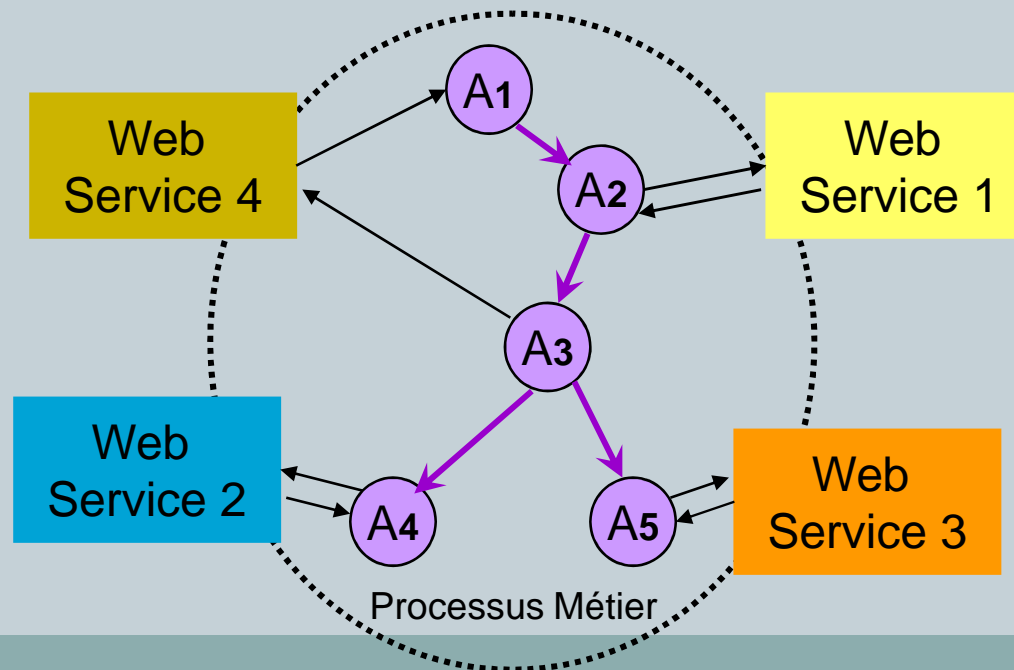


Web Services – Composition

31

- **Orchestration**

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



Web Services – Composition-Exemple

32

- **WS-BPEL:**

- Description XML d'un processus métier

- Référence les descriptions WSDL

- Activité de base

- ✦ Invoquer un service Web
 - ✦ Recevoir une invocation
 - ✦ Répondre à une invocation

<invoke>

<receive>

<reply>

- Activité structurée

- ✦ Séquence ordonnée
 - ✦ Branchement conditionnel
 - ✦ Boucle
 - ✦ Exécution en parallèle

<sequence>

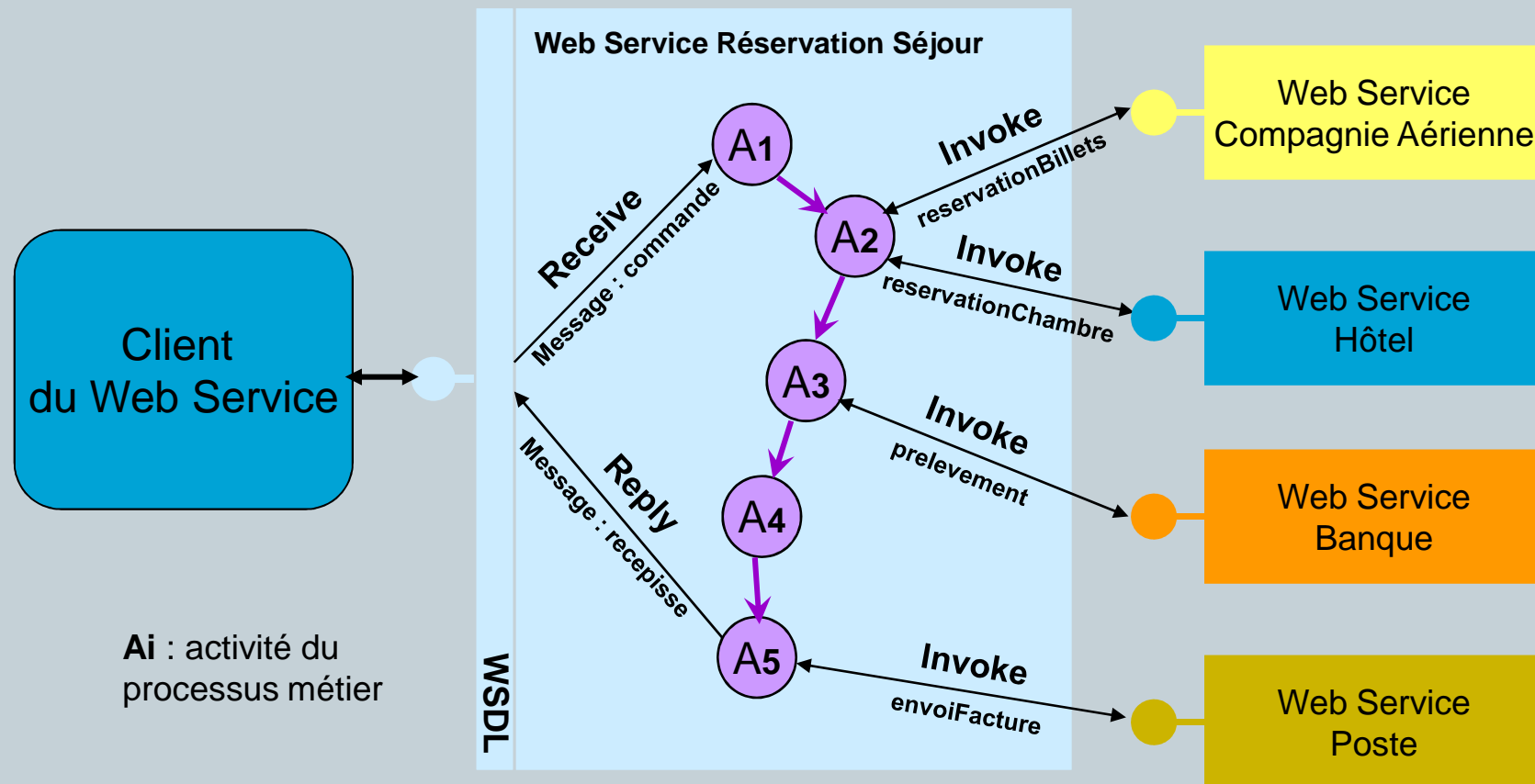
<switch>,<if>

<while>

<flow>

Web Services – Composition-Exemple

33



Web Services – Sécurité

34

- La problématique de **sécurité** des Web Services :
 - Authentification
 - Autorisation
 - Cryptage
- Les normes associées aux Web Services :
 - WS-Security;
 - SOAP Security Extensions;
 - XML Key Management;
 - ebXML Collaboration-Protocol Profile and Agreement Specification.

Web Services – Sécurité

35

- Injection SQL

Exemple:

```
<client>  
<NomduClient>ABC</NomduClient>  
<IDClient>12345</IDClient>  
</client>
```

=>

```
<Client>  
<NomduClient>ABC</NomduClient>  
<IDClient>12345; drop table clients; --</IDClient>  
</client>
```

Web Services – Sécurité

36

- Contenu malicieux
 - Message SOAP avec fichier attaché
 - ✓ Fichier dans le message XML
 - ✓ De plus en plus d'applications supportent XML nativement (Office 2003, 2007)
 - Buffer Overflow
 - ✓ Vérifier s'il y a vérification de la longueur maximale du champ
 - ✓ Exemple: service Web destiné à recevoir une liste de numéros de téléphone

Web Services – Sécurité

37

- Authentification
- Autorisation d'accès
- Encryptage / Décryptage
- Gestion des certificats
- Signature électronique
- Intégrité et confidentialité
- Non-répudiation
- Infrastructure à la clef public

Web Services – Demo

38