



## Architectures Orientées Services

### Service web SOAP

Enseignante  
Linda ZITOUNI

## Définition Web Services (W3C)

Un service web est un système logiciel **identifié par un URI** (*Unified Resource Identifier*), dont les **interfaces publiques et les liaisons** sont définis et décrits **en utilisant XML**. Sa définition peut être **découverte par d'autres systèmes logiciels**. Ces systèmes peuvent interagir avec le service web d'une manière prédéfinie par sa définition, en utilisant des **messages basés sur XML**, acheminés par des **protocoles internet**.

Linda ZITOUNI

## Apports des Services Web

- Peuvent correspondre à des applications nouvelles ou déjà existantes
- S'appuient sur des standards répandus
  - Meilleure interopérabilité
- **Ne sont pas centrés sur un langage ou technologie**
  - On peut faire communiquer du .Net avec du Java ou du Python

## Standards pour les Web Services

- Format de données
  - XML, XML Schéma
- Transport
  - HTTP, HTTPS, SMTP, FTP
  - SOAP, XML-RPC
- Description de service
  - WSDL, WSCL (*Web Services Conversation Language* : définit les interfaces abstraites du service web)
- Composition
  - MS XLANG, BPXL
- Découverte, Publication
  - UDDI (Universal Description Discovery and Integration)
- ⇒ **L'utilisation des standards rend l'accès aux services indépendant de la plateforme d'implémentation**

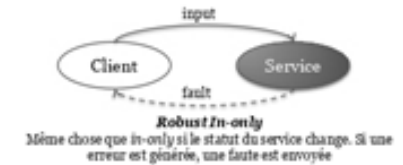
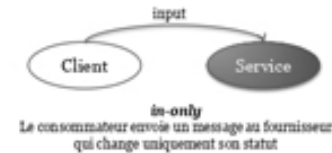
Linda ZITOUNI

## SOAP

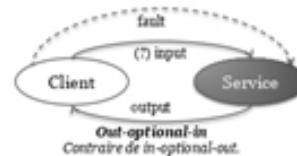
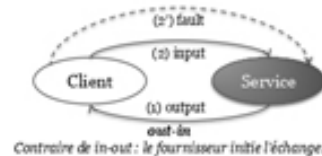
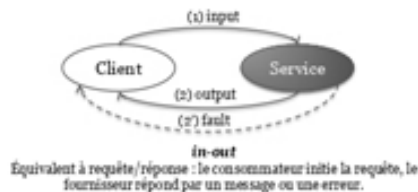
Service Oriented Architecture Protocol

- Acronyme initialement pour *Simple Object Access Protocol*, changé dans sa version 1.2 vers *Service Oriented Architecture Protocol*
- Un message SOAP
  - Permet la transmission d'un message au format XML
  - Va d'un nœud émetteur vers un nœud receveur
  - Passe éventuellement par un certain nombre de « nœuds » intermédiaires
    - Maintenir des logs, faire des vérifications ou modifier ou rediriger les messages
- SOAP s'appuie sur :
  - XML et XML Schema pour la représentation des messages
  - Les protocoles internet classiques pour la transmission des messages
    - HTTP, SMTP, JMS, FTP...

## Modèles d'Échange des Messages (1/2)



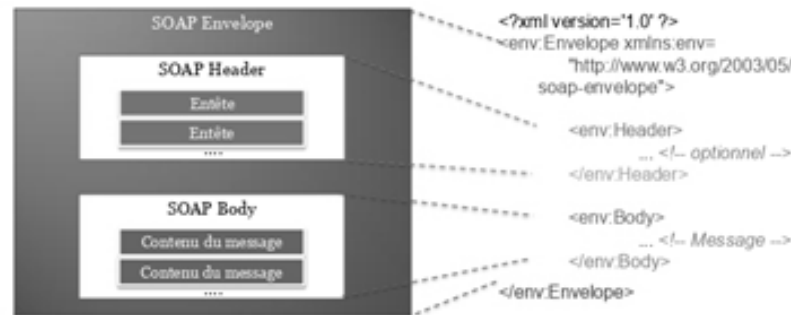
## Modèles d'Échange des Messages (2/2)



## Structure d'un Document SOAP

- Dans SOAP, les messages sont vus comme des **enveloppes** où l'application renferme les données à envoyer
- Une enveloppe SOAP contient une entête (*Header*) optionnelle, et un corps (*Body*) obligatoire
  - Leur contenu dépend de l'application, pas de la spécification SOAP

## Structure d'un Document SOAP

Entête SOAP (*Header*)

- Paramètres annexes : Propriétés du message
  - Corrélation de messages
  - Informations d'authentification
  - Informations liées aux transactions
- Chaque application peut décider de la signification des headers
- Les nœuds intermédiaires peuvent ajouter/supprimer/traiter/transformer des headers
- Un nœud intermédiaire peut traiter un ou plusieurs éléments du header
  - Sauf spécification contraire (utilisation de *env:relay*), cet élément est retiré du message
  - *Env:role* permet de spécifier les nœuds destinés à traiter l'élément

Corps SOAP (*Body*)

- Endroit où les données XML spécifiques à l'application, et qui doivent être échangées, sont placées
- L'élément *Body* est obligatoire, mais peut être vide.
- Il peut contenir plusieurs fils (appelés *des entrées*), qui peuvent être:
  - Données spécifiques à l'application: information échangée dans le WS. Représentée par une méthode et ses arguments (requête) ou une ou plusieurs valeurs (réponses)
  - Message de faute : utilisé en cas d'erreur.
- Un message SOAP contient des données ou un message de faute, mais pas les deux à la fois.

Exemple de Message SOAP (*Request*)

```

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header/>
  <env:Body>
    <commande xmlns="http://www.glares.com">
      <parfum>Fraise</parfum>
      <quantite>16</quantite>
      <livraison>
        <adresse>8 bvd N.Bohr Villeurbanne</adresse>
        <destinataire>M2TI</destinataire>
      </livraison>
    </commande>
  </env:Body>
</env:Envelope>

```

Exemple de Message SOAP (*Response*)

```
<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header/>
  <env:Body>
    <confirmation xmlns="http://www.glaces.com">
      <heure>10h</heure>
    </confirmation>
  </env:Body>
</env:Envelope>
```

## Fautes SOAP

- Utilisé lorsqu'une erreur survient
  - En général dans une réponse à un message reçu auparavant
- Se place dans la partie *Body*
- Le contenu est une explication de l'erreur:
  - Une valeur
  - Une explication textuelle
  - Un morceau de document spécifique à l'application
  - Le rôle du nœud à l'origine de l'erreur
    - Utile en présence de nœuds intermédiaires

## Exemple de Faute SOAP

```
<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header/>
  <env:Body>
    <env:Code>
      <env:Value>env:Receiver</env:Value>
    </env:Code>
    <env:Reason>
      <env:Text xml:lang="fr-FR">Plus de glaces à la fraise</env:Text>
      <env:Text xml:lang="en-GB">Nomore strawberry ice-creams</env:Text>
    </env:Reason>
  </env:Body>
</env:Envelope>
```

## Modèles de Communication SOAP

- SOAP supporte deux styles de communication:
  - RPC (Remote Procedure Call)
  - Document (ou message)

## Services SOAP *RPC-Style*

- Un service web RPC-style apparaît à l'application cliente comme un objet distant
- L'interaction entre un client et un service Web RPC est centrée autour d'une interface de service.
- Les clients expriment leurs requêtes sous forme d'appel de méthode avec un ensemble d'arguments, et le service retourne une réponse contenant une valeur de retour



## Services SOAP *RPC-Style* : Exemple

### Requête

```
<env:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.plastics_supply.com/product-prices">
  <env:Header>
    <tx:Transaction-id
      xmlns:tx="http://www.transaction.com/transactions"
      env:mustUnderstand="1">
      512
    </tx:Transaction-id>
  </env:Header>
  <env:Body>
    <m:GetProductPrice>
      <id> 450R60P </id>
    </m:GetProductPrice>
  </env:Body>
</env:Envelope>
```

### Réponse

```
<env:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.plastics_supply.com/product-prices">
  <env:Header> </env:Header>
  <env:Body>
    <m:GetProductPriceResponse>
      <price> 134.32 </price>
    </m:GetProductPriceResponse>
  </env:Body>
</env:Envelope>
```

## Services SOAP *Document-Style*

- Le corps SOAP contient un fragment de document
  - Ce n'est pas une structure XML
- L'environnement d'exécution SOAP accepte ce corps SOAP tel quel, et le transmet à l'application réceptrice sans le changer.



## Services SOAP *Document-Style* : Exemple

```
<env:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <tx:Transaction-id xmlns:tx="http://www.transaction.com/transactions" env:mustUnderstand="1">
      512
    </tx:Transaction-id>
  </env:Header>
  <env:Body>
    <po:PurchaseOrder orderDate="2004-12-02" xmlns:m="http://www.plastics_supply.com/POs">
      <po:from>
        <po:accountName> RightPlastics </po:accountName>
        <po:accountNumber> PSC-0343-02 </po:accountNumber>
      </po:from>
      <po:to>
        <po:supplierName> Plastic Supplies Inc. </po:supplierName>
        <po:supplierAddress> Yara Valley Melbourne </po:supplierAddress>
      </po:to>
      <po:product>
        <po:product-name> Injection molder </po:product-name>
        <po:product-model> G-100T </po:product-model>
        <po:quantity> 2 </po:quantity>
      </po:product>
    </po:PurchaseOrder>
  </env:Body>
</env:Envelope>
```

# Linda ZITOUNI

## WSDL

### Web Services Description Language

- Permet d'automatiser les détails de la communication entre les partenaires
- Peut être découvert à travers les registres de services
- Il est possible de vérifier si la communication entre les services est conforme au WSDL
- Éléments WSDL

- Types
- Messages
- Operation
- Port Type
- Binding
- Port
- Service

```
<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl" ... >
  <!-- types -->
  <!-- messages -->
  <!-- port types -->
  <!-- bindings -->
  <!-- service -->
</definitions>
```

## WSDL : Types

- Définitions des types de données
- Utilisés pour décrire les messages échangés
- Utilisent les schémas XML comme type canonique

```
<xsd:targetNamespace="http://example.com/stockquote.xsd"
  xmlns="http://www.w3.org/2000/10/XMLSchema">
  <element name="TradePriceRequest">
    <complexType>
      <all>
        <element name="TickerSymbol" type="string"/>
      </all>
    </complexType>
  </element>
  <element name="TradePrice">
    <complexType>
      <all>
        <element name="price" type="float"/>
      </all>
    </complexType>
  </element>
</xsd:schema>
```

## WSDL : Messages, Operations & Port Type

- Messages
  - Définitions abstraites et typées des données échangées
- Operations
  - Description abstraite d'une action
  - Fait référence à un message d'entrée et/ou de sortie
- Port Type
  - Collection d'opérations
  - Définition abstraite d'un service

```
<message name="Get Last TradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<message name="Get Last TradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
  <operation name="Get Last TradePrice">
    <input message="tns:Get Last TradePriceInput"/>
    <output message="tns:Get Last TradePriceOutput"/>
  </operation>
  <!-- More operations -->
</portType>
```

...

## WSDL : Binding, Port & Service

- Binding
  - Protocole et format de données concrets pour un port type (ex: SOAP 1.1 over HTTP, SOAP encoding...)
- Port
  - Définit une extrémité de communication unique
  - Adresse de l'extrémité pour le binding
  - URL (pour HTTP), email (pour SMTP)
- Service : ensemble de ports connexes

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="Get Last TradePrice">
    <soap:operation
      soapAction="http://example.com/Get Last TradePrice">
      <input> <soap:body use="literal"/>
      <input>
      <output> <soap:body use="literal"/>
      <output>
    </soap:operation>
  </binding>

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