

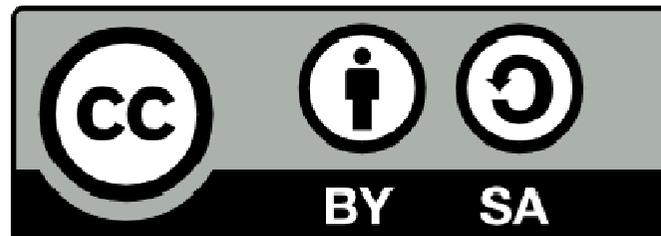
Licence

Creative Commons

Contrat Paternité

Partage des Conditions Initiales à l'Identique

2.0 France



<http://creativecommons.org/licenses/by-sa/2.0/fr>

Plan du cours

- Généralités SOAP
- SOAP par l'exemple : HelloWorld service
- Structure d'un message SOAP
- En tête d'un message SOAP (header)
- Corps d'un message SOAP (body)
- SOAP et le transport HTTP
- Outil SOAP UI

Déroulement du cours

➤ Pédagogie du cours

- Des bulles d'aide tout au long du cours
- Survol des principaux concepts en évitant une présentation exhaustive

➤ Logiciels utilisés

- Navigateur Web, SOAP UI



➤ Pré-requis

- Ingénierie des données, Schema XML, WSDL



➤ Remerciements

- TODO

Ressources

➤ Articles

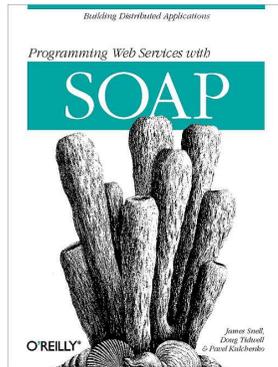
W3C www.w3.org/TR/soap/

- www.ibm.com/developerworks/xml/library/x-soapbx2/index.html
- www.soapui.org
- fr.wikipedia.org/wiki/SOAP

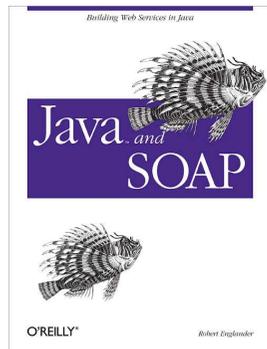
➤ Cours

- www.w3schools.com/soap
- apiacoa.org/publications/teaching/webservices/SOAP.pdf
- www.javapassion.com/webservices/SOAPBasics.pdf
- mbaron.developpez.com/divers/schemaxml/

Ressources : Bibliothèque



- Programming Web Services with SOAP
 - Auteur : James Snell, Doug Tidwell, Pavel Kulchenko
 - Éditeur : O'Reilly
 - Edition : Déc. 2001 - 264 pages - ISBN : 0596000952



- Java and SOAP
 - Auteur : Robert Englander
 - Éditeur : O'Reilly
 - Edition : Mai 2002 - 288 pages - ISBN : 0596001754



Understanding Web Services

XML, WSDL, SOAP, and UDDI

Eric Newcomer



- Understanding Web Services : ..., SOAP, ...
 - Auteur : Eric Newcomer
 - Éditeur : Addison-Wesley
 - Edition : Mai 2002 - 368 pages - ISBN : 0201750813

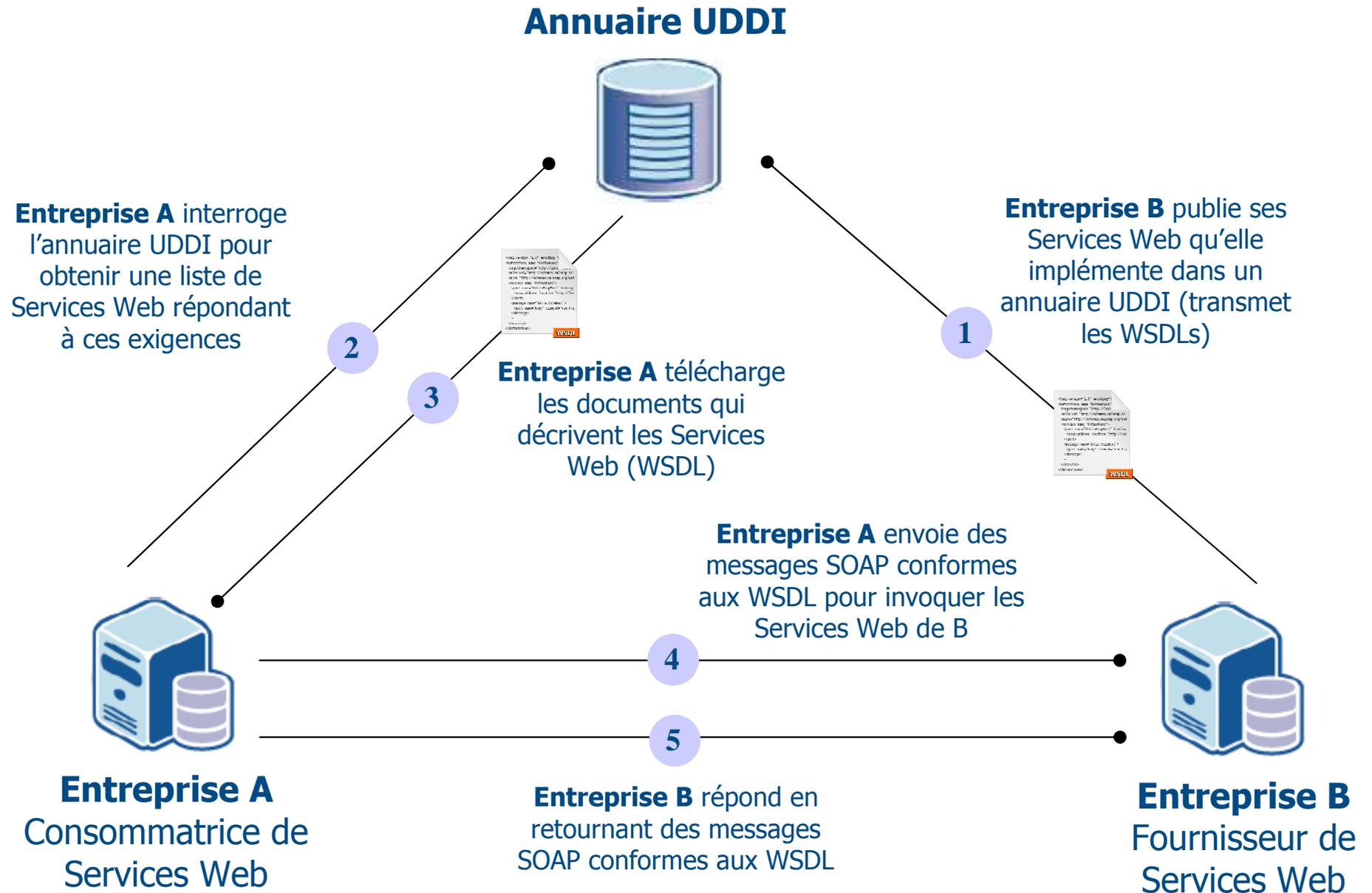
Généralités SOAP

- SOAP est un protocole de communication entre application basé sur le langage XML
- Initialement SOAP désignait l'acronyme de **S**imple **O**bject **A**ccess **P**rotocol
- Qui est derrière SOAP (Microsoft et IBM)
- Objectifs visés
 - Assurer la communication entre applications d'une même entreprise (intranet)
 - Assurer les échanges interentreprises entre applications et services Web
- Spécification du W3C
 - SOAP 1.1 : <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
 - SOAP 1.2 : <http://www.w3.org/TR/soap12/>
- Pour comparaison, SOAP est similaire aux protocoles « RPC »

Comparaison ...

	RMI	DCOM	CORBA	SOAP
Défini par	SUN	Microsoft	OMG	W3C
Plate-forme	Multi	Win32	Multi	Multi
Langage de Développement	Java	C++, VB, VJ	Multi	Multi
Langage de définition	Java	ODL	IDL	WSDL
Transport	TCP, HTTP, IIOP	IP/IPX	GIOP, IIOP	HTTP, HTTPR, SMTP
Transaction	Non	Oui	Oui	Oui
Sécurité	SSL, JAAS	?	SSL	SSL

Où est utilisé SOAP ?



Do you speak SOAP ?

- Pour lire et écrire du SOAP, les pré-requis sont
 - XML
 - XML Schema
- Le pour
 - Utile pour déboguer une application
 - Utile pour réaliser des tests via SOAP UI
 - Intercepter les messages bas niveau SOAP (via les handlers)
- Le contre
 - Les APIs fournissent une abstraction des messages SOAP

Concepts d'un message SOAP

- Les messages SOAP sont utilisés pour envoyer (requête) et recevoir (réponse) des informations d'un récepteur
- Un message SOAP peut être transmis à plusieurs récepteurs intermédiaires avant d'être reçu par le récepteur final (~ chaîne de responsabilité)
- Le format SOAP peut contenir des messages spécifiques correspondant à des erreurs identifiées par le récepteur
- Un message SOAP est véhiculé vers le récepteur en utilisant un protocole de transport (HTTP, SMTP, ...)

SOAP par l'exemple : Requête vers le service HelloWorld

➤ Exemple : Appeler les opérations du service *HelloWorld*

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:hel="http://helloworldwebservice.lisi.ensma.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <hel:makeHelloWorld>
      <value>Mickael BARON</value>
    </hel:makeHelloWorld>
  </soapenv:Body>
</soapenv:Envelope>
```

Message SOAP pour appeler l'opération *makeHelloWorld* contenant un paramètre *value*

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:hel="http://helloworldwebservice.lisi.ensma.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <hel:simpleHelloWorld/>
  </soapenv:Body>
</soapenv:Envelope>
```

Message SOAP pour appeler l'opération *simpleHelloWorld* ne contenant pas de paramètre

SOAP par l'exemple : Réponse du service HelloWorld

- Exemple (suite) : Message retour de l'appel des opérations du service *HelloWorld*

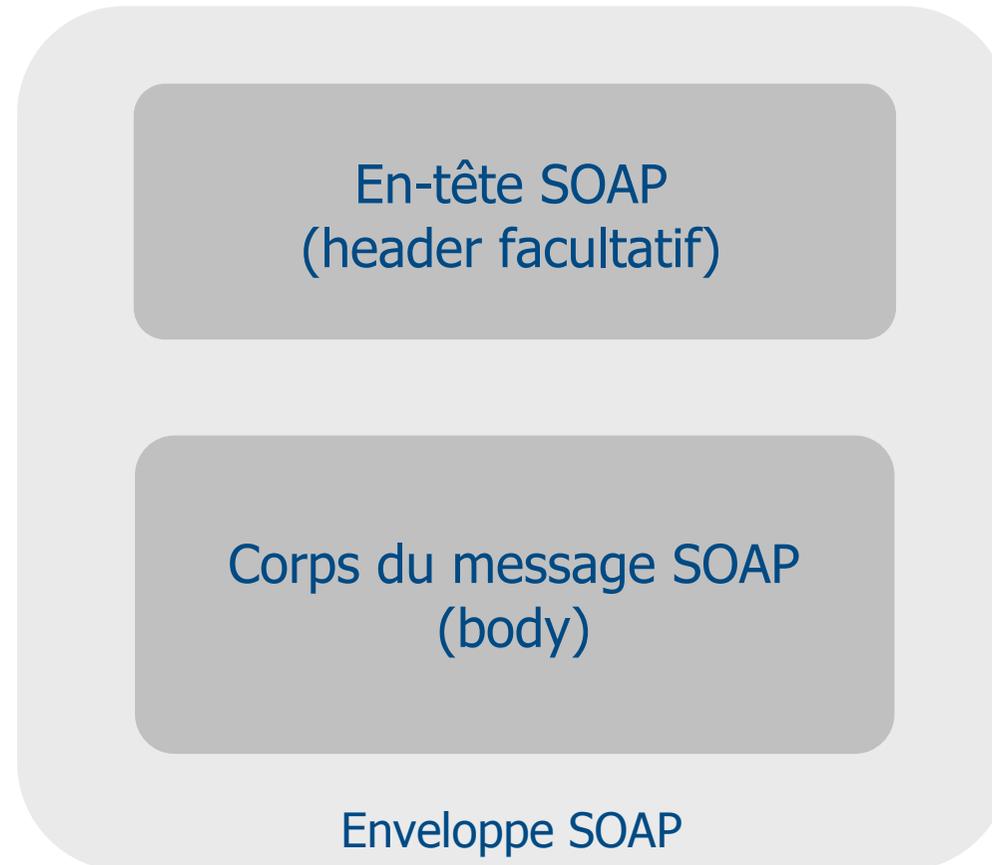
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:makeHelloWorldResponse xmlns:ns2="http://helloworldwebservice.lisi.ensma.fr/">
      <helloWorldResult>Hello World to Mickael BARON</helloWorldResult>
    </ns2:makeHelloWorldResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Les réponses sont
sensiblement identiques

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:simpleHelloWorldResponse xmlns:ns2="http://helloworldwebservice.lisi.ensma.fr/">
      <helloWorldResult>Hello World to everybody</helloWorldResult>
    </ns2:simpleHelloWorldResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Structure d'un message SOAP

- Un message SOAP est un document XML constitué d'une enveloppe composée de deux parties
 - Un en-tête (header) qui peut être facultatif
 - Un corps (body)



L'enveloppe SOAP

- L'enveloppe est la racine d'un message SOAP identifiée par la balise `<soapenv:Envelope>`
- La spécification impose que la balise et les sous balises soient explicitement associées à un *namespace*
- La spécification SOAP définit deux *namespaces*
 - *SOAP-ENV* ou *soapenv* : `http://schemas.xmlsoap.org/soap/envelope/`
 - *SOAP-ENC* : `http://schemas.xmlsoap.org/soap/encoding/`
- La requête et la réponse ont la même structure

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope">
  <SOAP-ENV:Body>
    <!-- Contenu de la Requête -->
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Message SOAP d'une
requête

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope">
  <SOAP-ENV:Body>
    <!-- Contenu de la Réponse -->
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

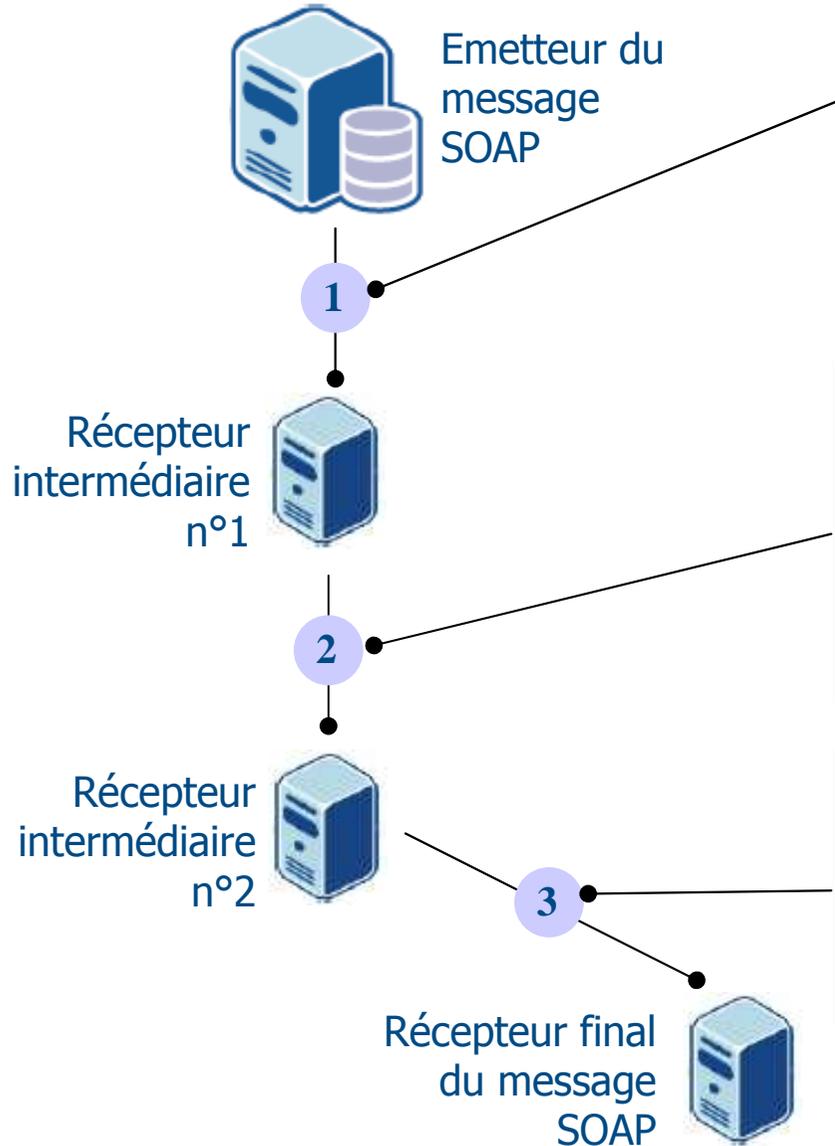
Message SOAP d'une
réponse

En-tête SOAP

- L'en-tête d'un message SOAP est utilisé pour transmettre des informations supplémentaires sur ce même message
- L'en-tête est défini par la balise *<SOAP-ENV:Header>*
 - L'élément peut être facultatif
 - Doit être placé avant le corps
- Différents usages de l'en-tête ?
 - Informations authentifiant l'émetteur
 - Contexte d'une transaction
 - Pour certains protocoles de transport (FTP par exemple), l'en-tête peut être utilisé pour identifier l'émetteur du message
- Un message SOAP peut transiter par plusieurs intermédiaires avant le traitement par le récepteur final
 - Pattern « Chaîne de responsabilité »
 - Zone lecture / écrite par les intermédiaires

En-tête SOAP

► En-tête dans une série de transferts successifs de message SOAP



```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <balise1
      SOAP-ENV:Actor="http://schemas.xmlsoap.org/soap/actor/next"
      SOAP-ENV:mustUnderstand="1" ... >
    </balise1>
    <balise2>
    </balise2>
    <balise3>
    </balise3>
  </SOAP-ENV:Header> ...
</SOAP-ENV:Envelope>
```

Valeurs conventionnelles pour la transmission par chaîne

```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <balise2
      SOAP-ENV:Actor="http://schemas.xmlsoap.org/soap/actor/next"
      SOAP-ENV:mustUnderstand="1" ...>
    </balise2>
    <balise3>
    </balise3>
  </SOAP-ENV:Header> ...
</SOAP-ENV:Envelope>
```

Le contenu de cet élément sera traité par le récepteur intermédiaire

```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <balise3
      SOAP-ENV:Actor="http://schemas.xmlsoap.org/soap/actor/next"
      SOAP-ENV:mustUnderstand="1" ...>
    </balise3>
  </SOAP-ENV:Header> ...
</SOAP-ENV:Envelope>
```

Corps SOAP

- Le corps d'un message SOAP est constitué par un élément *<SOAP-ENV:Body>*
- L'élément *<SOAP-ENV:Body>* peut contenir soit
 - Une erreur en réponse à une requête (élément *<SOAP-ENV:Fault>*)
 - Des informations adressées au destinataire du message SOAP respectant un encodage déterminé
- L'encodage des informations est précisé par les bindings du document WSDL
 - Attribut *style* (*Document* et *RPC*)
 - Attribut *use* (*encoded* et *literal*)
- Pour faire simple nous utiliserons les services Web dans le cadre de l'appel à une procédure distante

Corps SOAP

- L'objectif visé par SOAP a été de fournir un mécanisme standardisé pour l'appel de procédures distant (RPC)
- De ce fait les informations adressées au destinataire de messages SOAP doivent respecter un certain nombre de convention
- Appel d'une opération représentée par une *struct*
 - Le nom de la structure est celui de l'opération à appeler
 - Chaque paramètre de l'opération est défini comme un sous élément de la structure
 - Si un paramètre est un type complexe (*Person* par exemple) une nouvelle structure est définie contenant à son tour des sous éléments ...
- Le résultat est également représenté par une *struct*
 - Le nom de la structure est celui de l'opération suivi de **Response**
 - Les paramètres sont également structurés

Corps SOAP

➤ Exemple : corps de messages SOAP pour appeler des opérations du service Web Notebook

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:not="http://notebookwebservice.lisi.ensma.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <not:addPersonWithComplexType>
      <newPerson>
        <address>Poitiers</address>
        <birthyear>17081976</birthyear>
        <name>BARON Mickael</name>
      </newPerson>
    </not:addPersonWithComplexType>
  </soapenv:Body>
</soapenv:Envelope>
```

Message SOAP pour appeler
l'opération
addPersonWithComplexType

Paramètre de type complexe
défini dans une structure
(*newPerson*)

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:not="http://notebookwebservice.lisi.ensma.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <not:addPersonWithSimpleType>
      <name>BARON Mickael</name>
      <address>Poitiers</address>
      <birthyear>17081976</birthyear>
    </not:addPersonWithSimpleType>
  </soapenv:Body>
</soapenv:Envelope>
```

Message SOAP pour appeler
l'opération
addPersonWithSimpleType

Trois paramètres

Corps SOAP

- Exemple : corps de messages SOAP pour le résultat des opérations du service Web Notebook

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Body>  
    <ns2:addPersonWithComplexTypeResponse  
      xmlns:ns2="http://notebookwebservice.lisi.ensma.fr/">  
      <addPersonWithComplexTypeResult>true</addPersonWithComplexTypeResult>  
    </ns2:addPersonWithComplexTypeResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

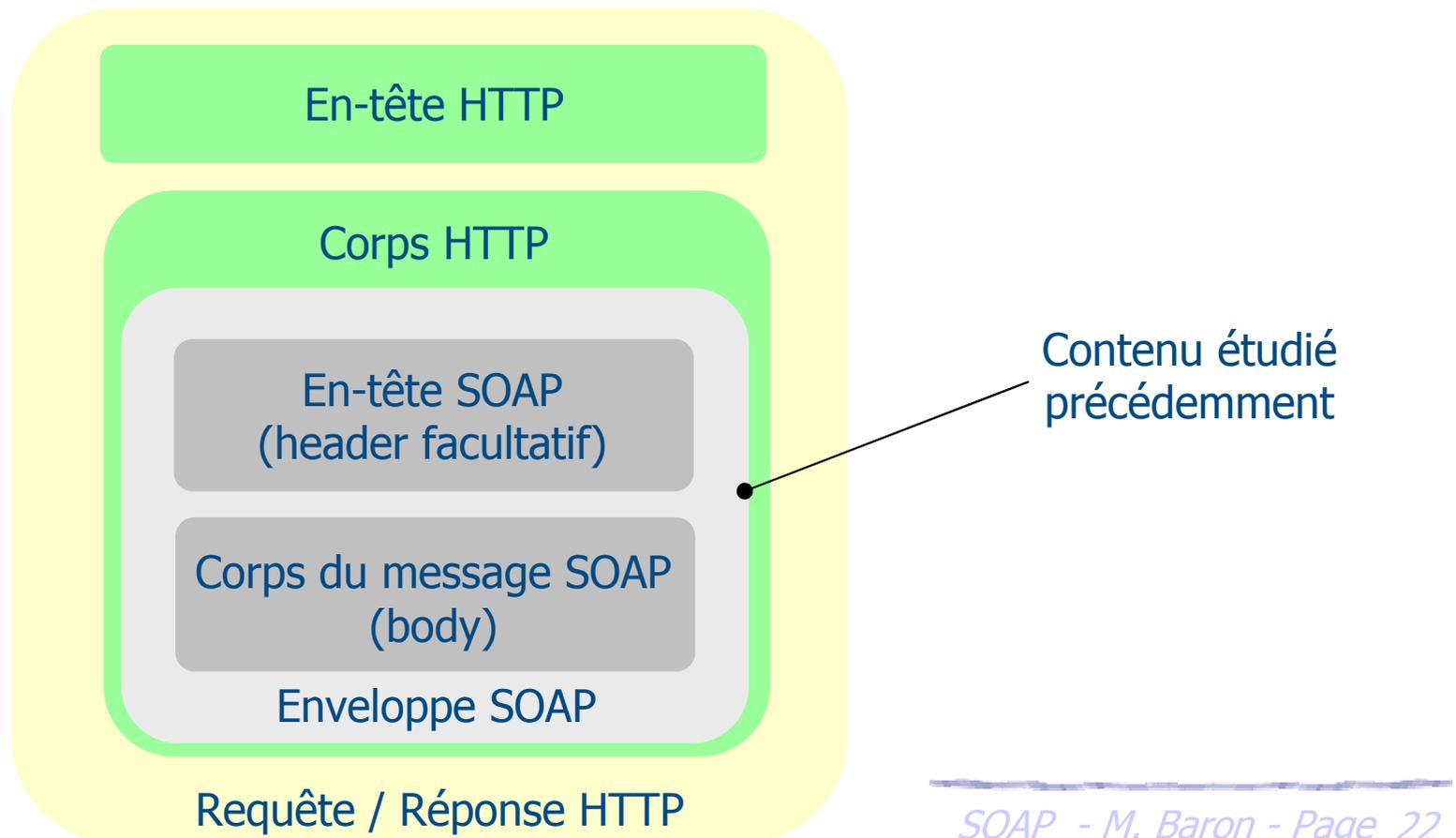
Messages SOAP pour la réponse
puisque les noms des opérations
sont suivis de *Response*

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Body>  
    <ns2:addPersonWithSimpleTypeResponse  
      xmlns:ns2="http://notebookwebservice.lisi.ensma.fr/">  
      <addPersonWithSimpleTypeResult>true</addPersonWithSimpleTypeResult>  
    </ns2:addPersonWithSimpleTypeResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Les paramètres de sorties suivent la
même convention que les
paramètres d'entrés

SOAP transporté par HTTP

- SOAP utilise un protocole de transport pour véhiculer les messages SOAP de l'émetteur au récepteur
 - HTTP, SMTP, FTP, POP3 et NNTP
- Le modèle requête/réponse de SOAP convient parfaitement au modèle requête/réponse HTTP



SOAP transporté par HTTP

- Requête SOAP HTTP
 - Méthode de type POST
 - Nécessite un attribut *SOAPAction*

```
POST http://localhost:8080/NotebookWebService/notebook HTTP/1.1
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8080
Content-Length: 459
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://notebookwebservice.lisi.ensma.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:addPersonWithComplexType>
      <newPerson>
        <address>Poitiers</address>
        <birthyear>17081976</birthyear>
        <name>BARON Mickael</name>
      </newPerson>
    </ns:addPersonWithComplexType>
  </soapenv:Body>
</soapenv:Envelope>
```

- Réponse SOAP HTTP
 - Exploite les codes retours HTTP
 - Si code de type 2xx, message SOAP reçu
 - Si code 500, message en erreur, le corps SOAP doit contenir *fault*

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Sun, 13 Dec 2009 12:00:33 GMT

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Body>
    <ns2:addPersonWithComplexTypeResponse
      xmlns:ns2="http://notebookwebservice.lisi.ensma.fr/">
      <addPersonWithComplexTypeResult>
        true
      </addPersonWithComplexTypeResult>
    </ns2:addPersonWithComplexTypeResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Traitement des messages SOAP

```

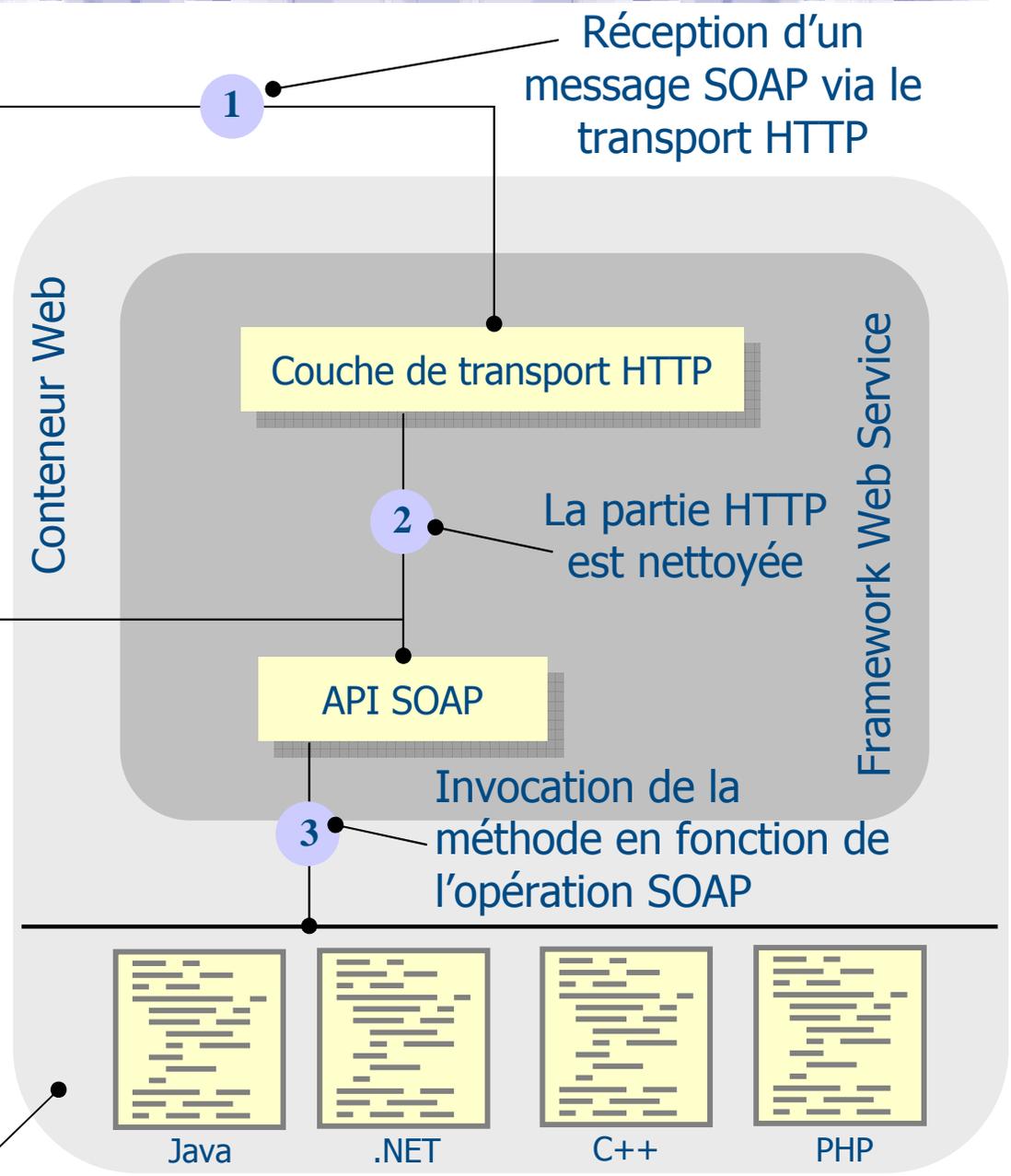
POST http://localhost:8080/NotebookWebService/notebook HTTP/1.1
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8080
Content-Length: 459
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:not="http://notebookwebservice.lisi.ensma.fr/"
  <soapenv:Header/>
  <soapenv:Body>
    <not:addPersonWithComplexType>
      <newPerson>
        <address>Poitiers</address>
        <birthyear>17081976</birthyear>
        <name>BARON Mickael</name>
      </newPerson>
    </not:addPersonWithComplexType>
  </soapenv:Body>
</soapenv:Envelope>

```

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:not="http://notebookwebservice.lisi.ensma.fr/"
  <soapenv:Header/>
  <soapenv:Body>
    <not:addPersonWithComplexType>
      <newPerson>
        <address>Poitiers</address>
        <birthyear>17081976</birthyear>
        <name>BARON Mickael</name>
      </newPerson>
    </not:addPersonWithComplexType>
  </soapenv:Body>
</soapenv:Envelope>

```



Couche d'accès aux implémentations des services Web (différents langages)

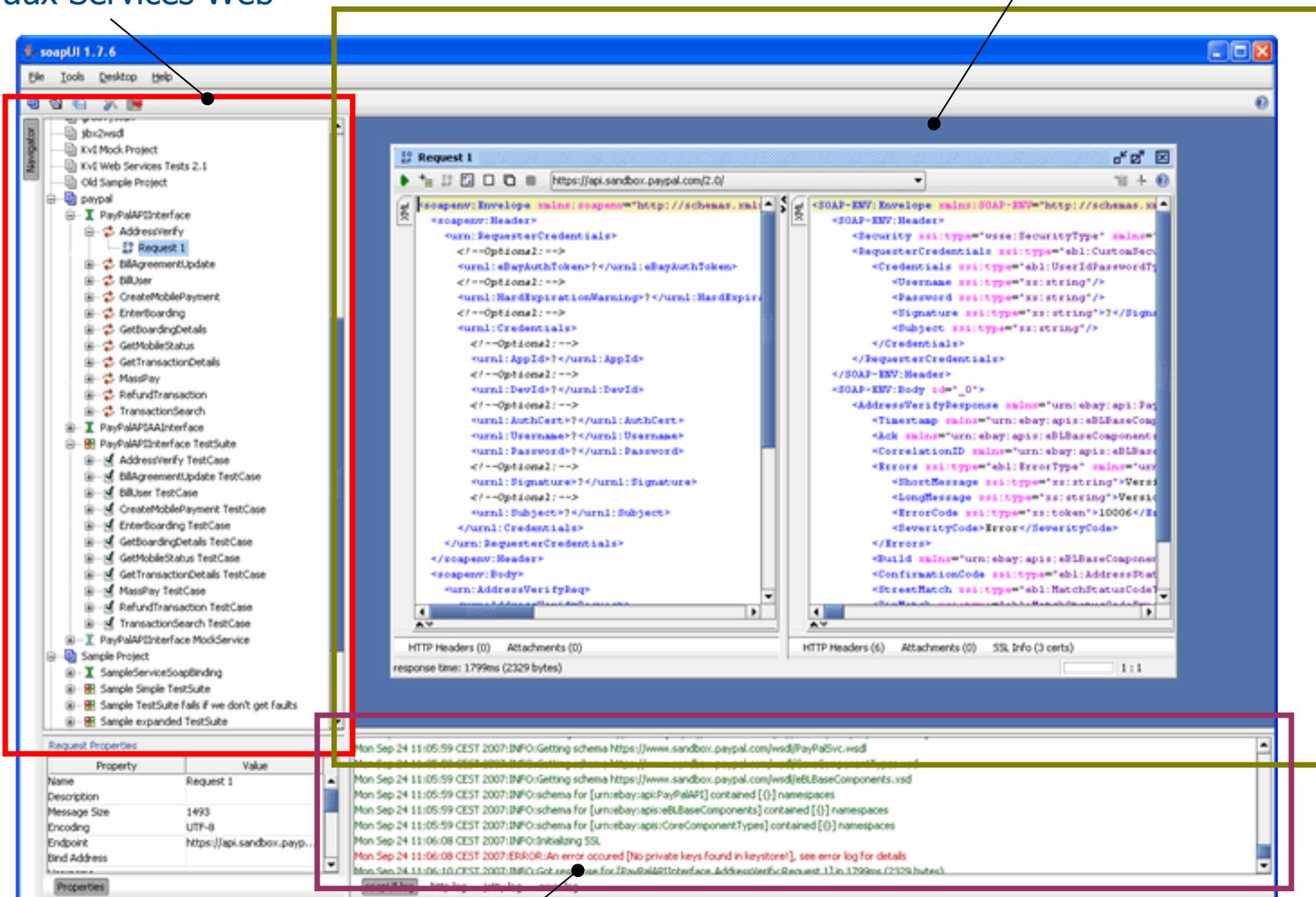
SOAP UI : outil graphique de tests de Service Web

- SOAP UI est un outil pour tester des Services Web
 - *www.soapui.org*
 - Disponible pour en *standalone* ou intégré dans les environnements de développement (Eclipse, IntelliJ, Netbeans, Maven, ...)
 - Peut s'utiliser pour n'importe quelle plateforme de développement
- Fonctionnalités de SOAP UI
 - Supporte les Services Web étendus (WSDL + SOAP + UDDI) ou REST
 - Inspecter des Services Web
 - Invoquer des Services Web
 - Développer des Services Web
 - Simuler des Services Web via des bouchons (mocks)
 - Effectuer des tests qualités (temps de réponse, ...)

SOAP UI : outil graphique de tests de Service Web

Les projets regroupant les appels aux Services Web

Les messages SOAP (requête et réponse)



Les logs de SOAPUI