

---

# Sigems échange des données de façon *structurée et interactive*

---

## *Introduction aux web-services*

**Auteurs** Sébastien Morvan – *ingénieur informatique ENSSAT*  
Nathalie Fourès – *ingénieur informatique ENSSAT*  
E-mail : smorvan@sigems.fr  
nathalie\_foures@sigems.fr

**Société** Sigems  
15 rue de la Prune  
BP 224  
47305 Villeneuve-sur-Lot  
Tél. : 05 53 49 19 38  
Fax : 05 53 70 29 50

**Version** 4.2

**Date de création** 22 mars 2004

**Date de dernière  
mise à jour** 18 mars 2005

**Sujet** Présentation du cadre développé par Sigems pour réaliser  
des échanges de données avec des logiciels externes.

Présentation de la façon d'utiliser ce cadre.

**Mots clés** échange de données, passerelle, client/serveur, web-  
service, HTTP, RPC, XML, SOAP, WSDL

# Table des matières

<b>TABLE DES MATIERES .....</b>	<b>2</b>
<b>1. CONTEXTE .....</b>	<b>3</b>
<b>2. LES DONNEES A ECHANGER.....</b>	<b>4</b>
<b>3. POURQUOI LES WEB-SERVICES ? .....</b>	<b>5</b>
<b>4. LES WEB-SERVICES SIGEMS .....</b>	<b>7</b>
1. LE SERVICE D'IDENTIFICATION PATIENT : <i>SIGEGATEPAT</i> .....	8
2. LES SERVICES RELATIFS A UN SEJOUR MCO .....	11
3. LE SERVICE DICTIONNAIRE : <i>SIGEGATEDICO</i> .....	12
4. LE SERVICE DE CONFIGURATION : <i>SIGEGATECONF</i> .....	13
<b>5. UN EXEMPLE DE CLIENT DE WEB-SERVICE .....</b>	<b>14</b>

# 1. Contexte

Dans le soucis de *mieux prendre en charge les patients*, que ce soit dans une clinique, ou dans un cabinet médical, il devient nécessaire pour ces deux partenaires de la santé d'échanger des informations sur leurs patients communs.

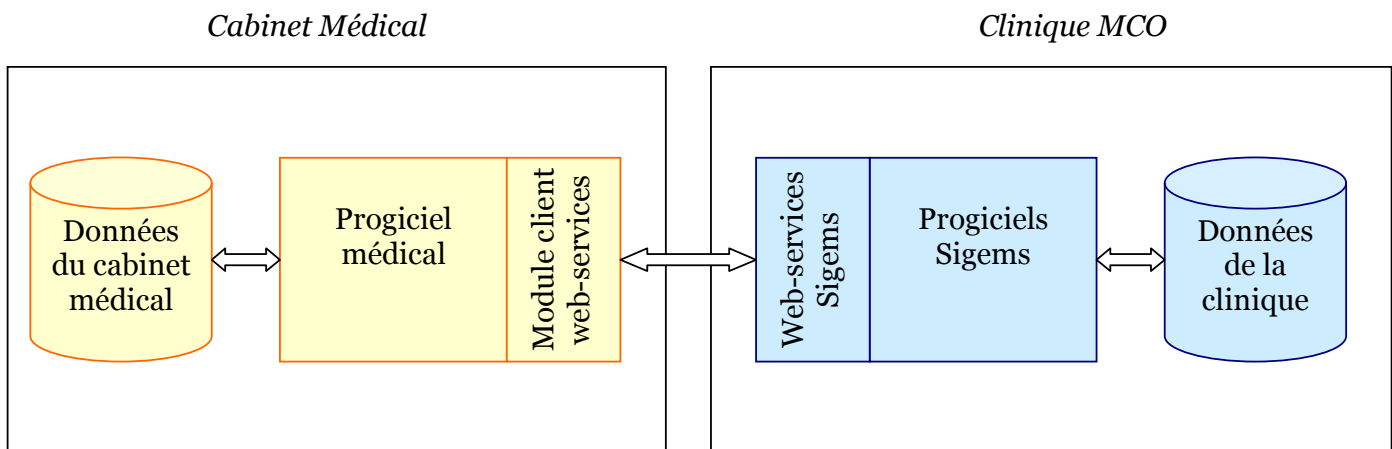
Les sociétés de service de ces deux partenaires sont responsables de la mise en place de ces échanges.

Sigems développe une passerelle pour échanger des informations administratives et médicales de patients, avec des logiciels externes. Cette passerelle est basée sur la technologie des web-services.

Ce document a pour objectif d'expliquer :

- la partie serveur de cette passerelle développée par Sigems,
- la partie cliente de cette passerelle que vous pourrez développer pour récupérer ou fournir des données hébergées par Sigems.

**Les termes client/serveur sont utilisés, dans tout ce document, au sens informatique du terme et représentent des programmes informatiques et non les utilisateurs finaux.**



## 2. Les données à échanger

La passerelle que nous pourrions mettre en place ensemble permettrait d'échanger des données concernant :

- les patients,
- les dossiers,
- les séjours,
- les actes et diagnostics faits durant chaque séjour.

Les échanges se font de façon bilatérale. Autrement dit, le progiciel externe à Sigems peut lire des données hébergées par la clinique. Il peut aussi fournir des données dont la clinique ne dispose pas.

### **Les données patient :**

La passerelle Sigems permet d'échanger les informations administratives d'un patient comme :

- son numéro d'identification unique dans la base Sigems,
- ses nom, prénom, genre,
- ses coordonnées,
- ses date et ville de naissance,
- son matricule de sécurité sociale,
- son groupe sanguin,
- les coordonnées de son médecin traitant.

### **Les données séjour :**

La passerelle Sigems permet en outre d'échanger des données concernant les séjours d'un patient c'est-à-dire :

- le numéro de séjour,
- les date et heure d'entrée,
- les date et heure de sortie,
- le mode de traitement,
- la localisation du patient durant le séjour (service, chambre, type de chambre),
- le motif d'entrée dans la clinique,
- les date et heure d'intervention.

La passerelle permet de faire une pré-admission dans la clinique via le progiciel externe, en fournissant les données ci-dessus.

### **Les données concernant les actes et les diagnostics :**

Grâce à la passerelle, vous pouvez enfin récupérer ou fournir des données comme :

- des codes d'actes ou de diagnostics,
- les date et heure auxquelles ils ont été effectués,
- le praticien qui les a effectués.

## 3. Pourquoi les web-services ?

### *Une brève définition*

Les web-services sont des services accessibles via le web, c'est-à-dire un réseau HTTP. Voici un scénario typique d'utilisation des Web services : une application cliente envoie une requête à un service, situé à une adresse donnée (URL), en utilisant le protocole SOAP. Le service reçoit cette requête, la traite et retourne une réponse.

### *Vers une solution indépendante du réseau*

La seule contrainte réseau imposée par la solution web-service consiste à ce que la machine client et la machine serveur soient connectées à un réseau TCP/IP. Ce peut être le plus vaste comme Internet, ou le plus restreint comme un réseau local.

Avec les web-services, le cabinet médical et la clinique peuvent ne pas être sur le même réseau local, ou être distants de plusieurs centaines de kilomètres, les réponses du service sont transmises en temps réel au logiciel client, de façon transparente.

### *Vers une solution indépendante des systèmes d'exploitation*

La technologie utilisée par les web-services permet d'invoquer une opération distante grâce à un message XML. Ce langage de description est interprétable par n'importe quel système d'exploitation. Ainsi, vous pouvez développer un client sur une plate-forme Windows, en C++, qui interroge un service d'une plate-forme Linux, développé en Java !

La solution web-service n'impose pas de contraintes supplémentaires quant aux choix des technologies déjà faits pour développer nos progiciels respectifs.

### *Vers une solution interactive*

Pour échanger des informations à un format XML donné, il existe d'autres solutions que celle des web-services. Par exemple, on peut imaginer que le cabinet médical dépose dans un répertoire d'une machine de la clinique, un fichier XML contenant des informations sur un patient. Le serveur de la clinique devrait alors parcourir ce fichier, traiter ses données et les insérer dans la base de la clinique.

D'une part, cette solution a beau être simple en apparence, elle rend les communications plus compliquées. En effet, le logiciel client ne sait ni quand l'information sera traitée, et ni si le traitement s'est bien passé.

D'autre part, l'échange ne se fait que de façon unilatérale. Pour obtenir un échange bilatéral, il faut que chacun des protagonistes développe :

- un programme client qui fabrique les fichiers XML de données à insérer chez le partenaire,
- et un programme serveur qui isole et traite les fichiers XML déposés par son partenaire.

Ce qui veut dire un coût de développement doublé.

Bref, les solutions cronées sont simples de prime abord, mais leur manque d'interactivité risque d'introduire des erreurs et des développements annexes lourds. La solution web-service est un peu plus compliquée à mettre en œuvre mais est interactive par définition.

### ***Vers des liens plus étroits sans dépendances techniques***

Les changements internes que peut faire Sigems dans ses bases de données ou dans la façon interne d'accéder à ses informations, sont transparents pour le client des web-services. En effet, c'est l'interface web-services qui absorbe ces changements et qui conserve les mêmes échanges avec les programmes clients.

Nous pensons ainsi à un lien ODBC, qui a la particularité de lier très fortement deux bases. Cela a l'avantage d'échanger simplement des informations, mais le gros inconvénient de lier les développements à un référentiel qui peut changer. Les développements des deux protagonistes sont alors à remettre en cause.

La solution web-service permet d'échanger des données de façon précise sans pour autant lier de façon statique les bases de données des deux protagonistes.

## 4. Les web-services Sigems

Dans cette partie, nous décrivons les principaux web-services Sigems, dont les opérations sont à la disposition des progiciels médicaux.

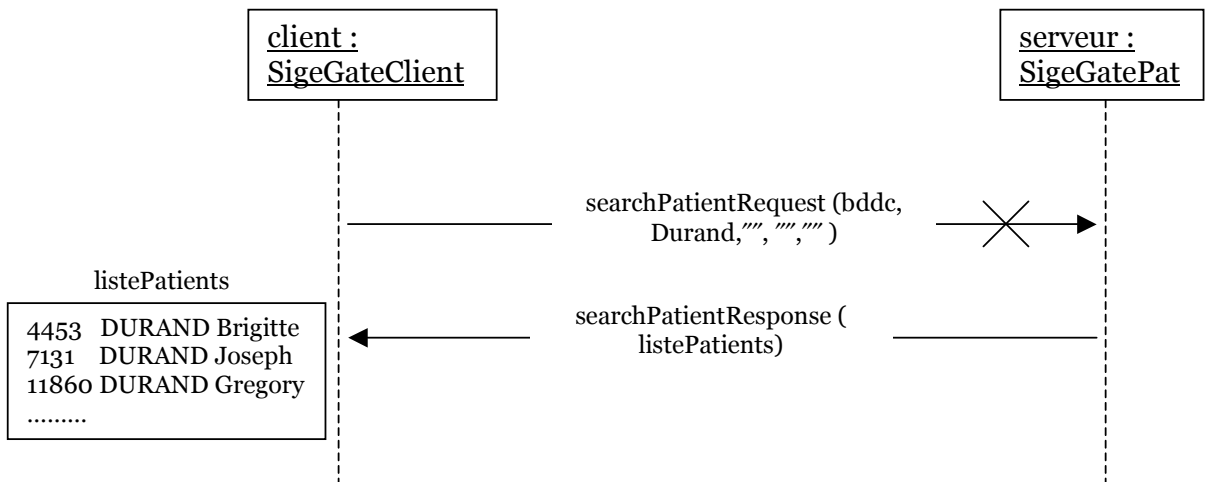
Un programme client et un programme service échangent des données sur des patients, par exemple. Par abus de langage, nous disons qu'un service et un client échangent un patient. Nous faisons le même abus de langage pour les informations médicales, mais c'est peut-être moins choquant.

*Chaque service n'est accessible que par des utilisateurs autorisés. Ainsi, nous vous fournirons, en temps voulu le mot de passe et le nom d'utilisateur nécessaires à l'authentification de vos clients web-services.*

## 1. Le service d'identification patient : *SigeGatePat*

Ce service permet de gérer les informations administratives des patients, grâce aux quatre opérations suivantes. Elles retournent toutes un status qui indique comment s'est déroulée l'opération côté serveur. Les codes correspondant à ces status sont décrits en annexes.

- **searchPatient** : recherche de patients en fonction des critères suivants : nom de naissance, nom usuel, prénom, et date de naissance. Cette opération renvoie une liste de patients dont l'identité correspond aux paramètres d'entrée fournis. Chaque élément de la liste est composé de ses nom, prénom, date de naissance et IPP dans la base Sigems de la clinique. Cette liste est en fait un fichier XML au format SOAP.

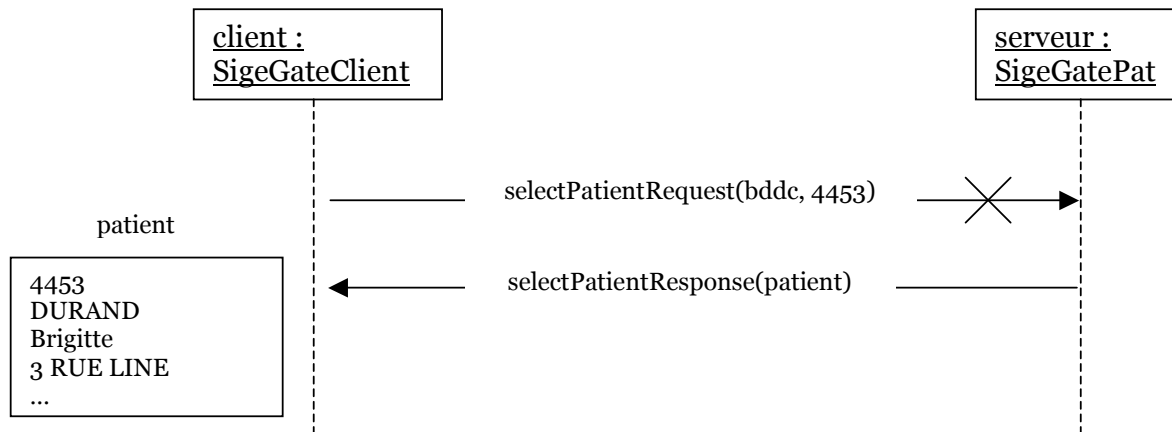


**N.B.** : vous constatez que searchPatient prend en paramètre d'entrée :

- un identifiant « bddc »,
- un nom « Durand », c'est le nom de naissance recherché dans la base Sigems,
- des chaînes vides pour le nom usuel, le prénom et la date de naissance qui signifient que ces critères ne doivent pas être pris en compte.

Le premier identifiant est le nom de la base de données dans laquelle le service doit faire sa recherche. En effet, une clinique peut avoir plusieurs bases de données. Vous devez indiquer celle que vous voulez interroger. Vous pourrez vous procurer la liste des identifiants des bases auxquelles vous avez accès grâce au web-service SigeGateConf décrit plus loin.

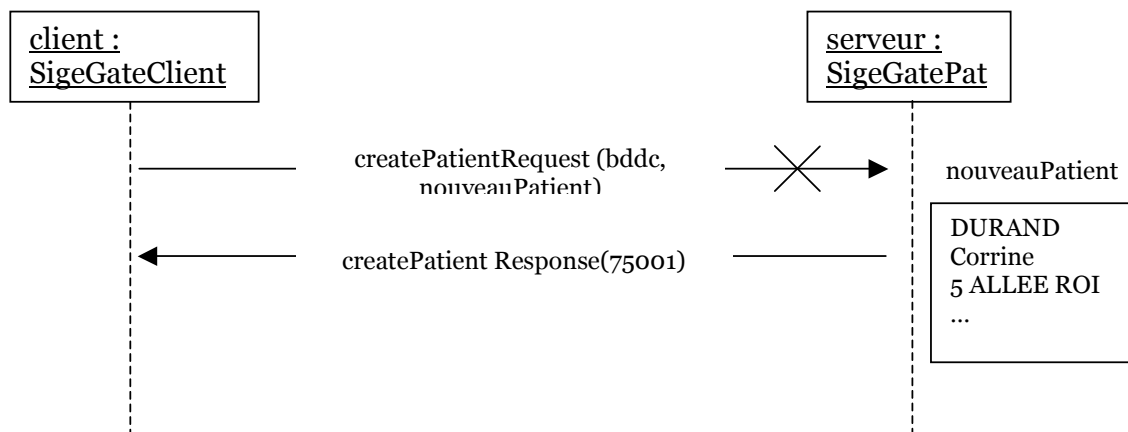
- ***selectPatient*** : sélection d'un patient dont l'IPP Sigems est connu. Le client web-services fournit un IPP au service *SigeGatePat*. Celui-ci retourne toutes les informations correspondant à l'IPP donné.



- ***createPatient*** : création d'un nouveau patient. Si le nom, prénom, date de naissance du patient, donné en paramètre d'entrée, ne sont pas déjà enregistrés dans la base Sigems, alors ce patient est inséré dans la base Sigems de la clinique. Le service retourne alors l'IPP Sigems sous lequel l'insertion vient de se faire.

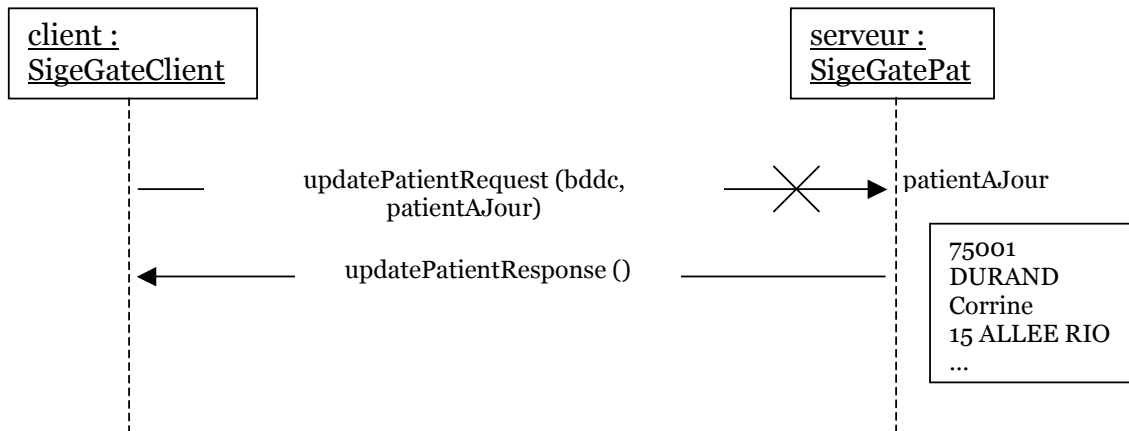
Si le patient est déjà enregistré dans la base Sigems, il n'est pas inséré de nouveau, mais l'IPP Sigems sous lequel il est déjà enregistré est retourné, sans aucune modification des données de la base Sigems. Le status mentionnera que le patient est déjà présent ou non, dans la base Sigems.

Dans l'exemple suivant, les informations du patient Durand Corrine sont enregistrées sous l'IPP 75001.



- **updatePatient** : mise à jour d'un patient dont l'IPP Sigems est connu. Le cabinet médical fournit un paramètre d'entrée *patient*, en vue de sa mise à jour dans la base Sigems. Seul un status est renvoyé pour indiquer au logiciel client si la mise à jour est effective ou non.

Cette opération est délicate. En effet, toutes les données du patient passé en paramètre d'entrée doivent être remplies, même celles qui ne doivent pas être mises à jour. Sinon, une donnée vide écrase un champ non vide de la base de données. En résumé tous les champs envoyés sont réécrits.



## 2. Les services relatifs à un séjour MCO

Les échanges concernant les informations des dossiers et des séjours sont gérés respectivement par les services *SigeGateDosBase* et *SigeGateSej*. Les échanges concernant les actes sont gérés par les services *SigeGateActeCcam* et *SigeGateActeCdam*. Voici une description introductive des opérations qu'ils supportent.

### **Echanges concernant un dossier, c'est-à-dire un séjour MCO : *SigeGateDosBase***

- *searchDossier* : recherche d'un dossier en fonction de l'IPP du patient concerné, d'une période où serait comprise la date d'entrée et d'un nom de praticien. Chacun de ces paramètres peut être vide. Cette opération retourne une liste de dossiers.

Par exemple, le progiciel médical peut récupérer tous les dossiers d'un patient, ou tous les dossiers établis entre telle et telle période.

- *selectDossier* : sélection d'un dossier en fonction d'un numéro de dossier. Cette opération retourne le dossier concerné, si le numéro fourni existe.
- *createPA* : cette opération permet de créer un dossier de pré-admission. Le numéro de dossier provisoire sous lequel il est créé est retourné au logiciel client.

Cette opération permet au praticien utilisant un progiciel externe à Sigems d'établir un dossier de pré-admission pour un de ces patients, avec toutes les réservations de lit et de bloc qui en découlent.

- *updateDossier* : cette opération permet de mettre à jour un dossier de pré-admission ou certains champs d'un dossier d'admission.

### **Echanges concernant les actes et les diagnostics : *SigeGateDiag*, *SigeGateActeCcam*, *SigeGateActeCdam***

De la même façon, les actes et les diagnostics survenus lors d'un séjour sont manipulables. Vous pouvez renseigner ou récupérer des actes ou des diagnostics pour un dossier donné, ou pour un patient donné, ou pour un praticien donné. A chaque fois, votre requête peut mentionner un intervalle de temps contenant le jour où l'acte ou le diagnostic recherché a été fait.

### 3. Le service dictionnaire : *SigeGateDico*

Nous utilisons des codes dans les messages que nous renvoyons. Par exemple, nous n'écrivons pas en toutes lettres le nom du service dans lequel séjourne un patient, mais le code associé à ce service dans notre base.

Pour que vous sachiez à quoi correspondent les codes que nous utilisons, nous avons mis à votre disposition un service *SigeGateDico* de traduction de ces codes.

Le service *SigeGateDico* permet :

- de récupérer la liste des mots-clé pour lesquels il existe des définitions, comme `modeTraitement`, `service`, `uniteMedicale`, etc.

**Opération** : c'est l'opération *getMotCleListe* qui permet de récupérer la liste des mots-clé disponibles, avec leur description, pour lesquels vous pourrez ensuite demander des définitions.

- de récupérer toutes les définitions associés à un mot-clé,

**Exemple** : en donnant à ce service un mot-clé comme `modeTraitement`, celui-ci renvoie une liste de définitions contenant chacune un couple code-libellé.

**Opération** : c'est l'opération *getDefinitions* qui renvoie la liste des définitions associées à un mot clé.

- de récupérer le libellé d'un code pour un mot-clé donné,

**Exemple** : en donnant à *SigeGateDico* un mot-clé comme `modeTraitement`, et un code comme `03`, le service renvoie une définition contenant le couple code-libellé suivant :

`03 - hospitalisation complète.`

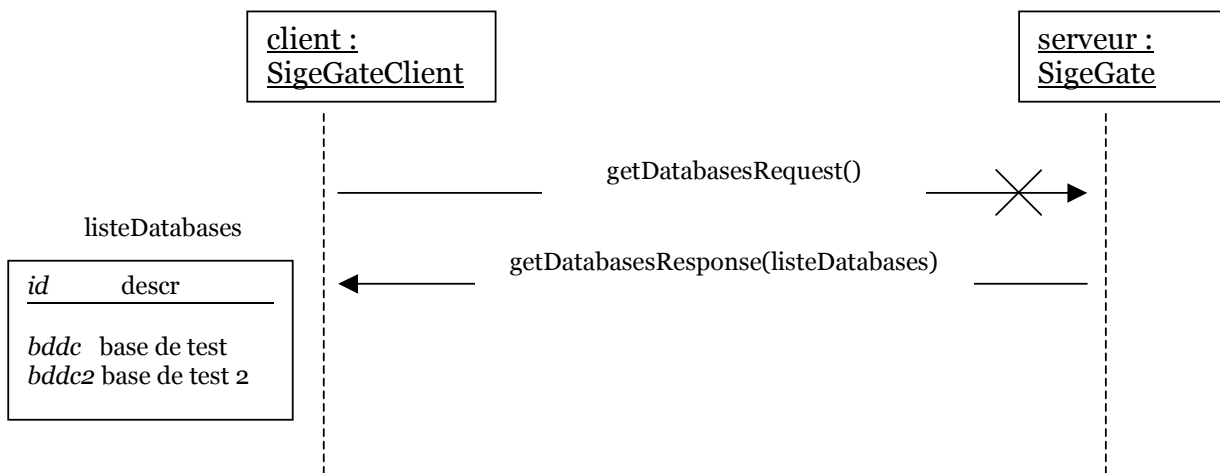
**Opération** : c'est l'opération *getDefinition* qui renvoie le libellé d'un code pour un mot-clé donné.

#### 4. Le service de configuration : *SigeGateConf*

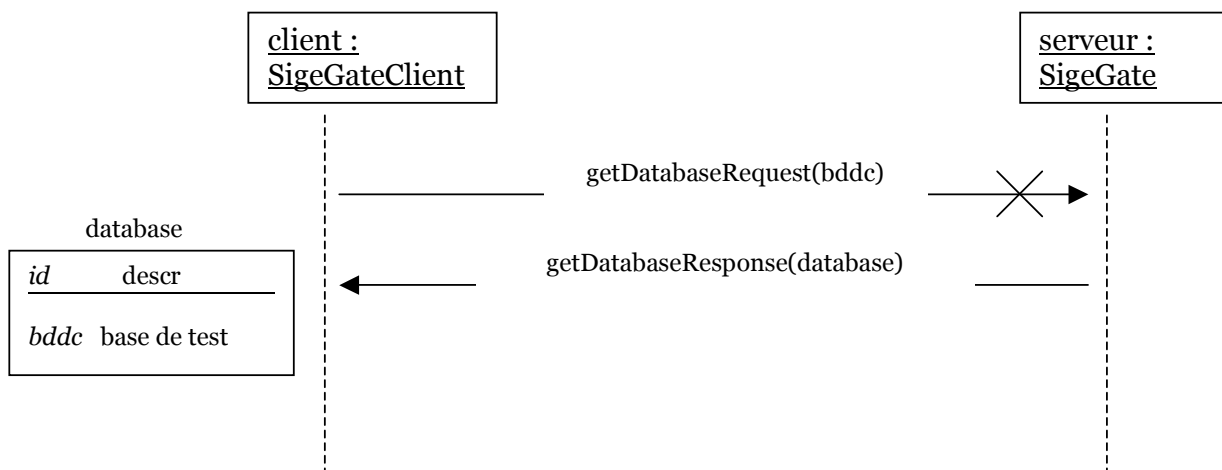
**Chaque groupe d'échanges mentionnés ci-dessus prend un identifiant de base de données, en paramètre d'entrée.** En effet, une clinique peut posséder plusieurs bases de données, typiquement lorsque des cliniques se regroupent.

Ainsi, avant tout échange de données utiles, il s'agira de faire un échange préalable de configuration. Le progiciel médical demandera au service de configuration *SigeGateConf* la liste des bases présentes sur le site de la clinique, dont l'accès lui est autorisé. Cette liste contiendra l'identifiant et le descriptif succinct de chaque base accessible.

- *getDatabases* : cette opération consiste pour le logiciel client à récupérer l'identifiant et un descriptif très succinct des bases de données qu'il peut consulter, par l'intermédiaire des web-services Sigems. Bien sûr, nous nous serons mis d'accord ensemble, au préalable sur les bases que vous voulez consulter.



- *getDatabase* : cette opération consiste à récupérer un bref descriptif d'une base dont le logiciel client ne connaît que l'identifiant. Bien sûr le logiciel client peut seulement récupérer le descriptif d'une base qu'il est autorisé à consulter.

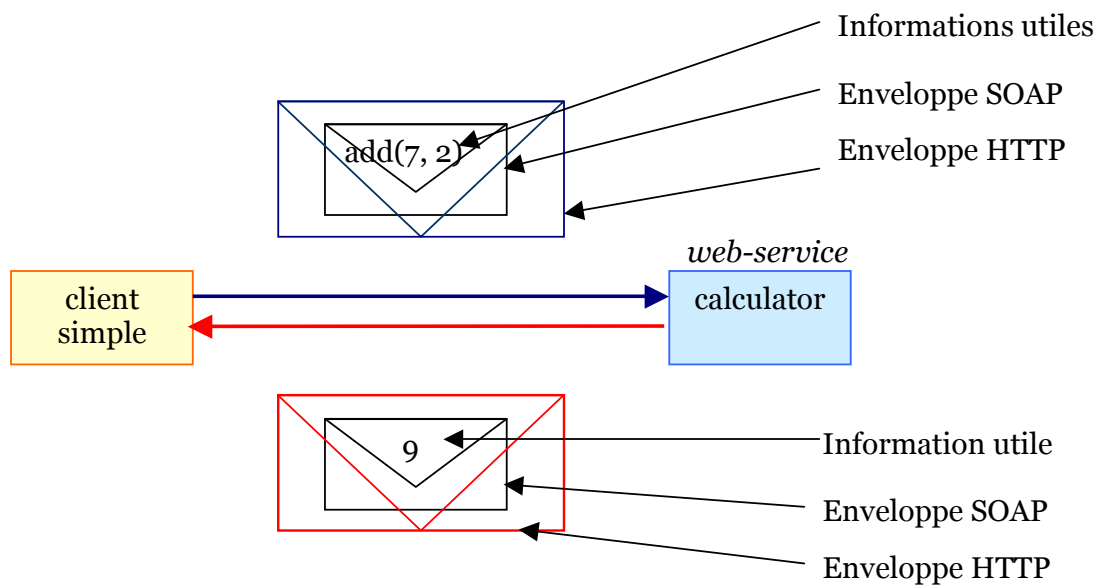


## 5. Un exemple de client de web-service

La partie serveur de la passerelle est développée par la Sigems. Pour pouvoir l'interroger, vous pouvez développer une partie cliente. Ce chapitre a pour but d'introduire en quoi consiste un client de web-service.

Si vous êtes intéressé par le développement d'un client devant interroger les web-services Sigems, nous vous ferons parvenir un document plus technique et nous vous accompagnerons dans sa mise en place.

Voici un web-service simple : calculator. Celui-ci sait additionner ou soustraire deux entiers. Le programme client devra interroger ce service en lui passant en paramètre les deux entiers à manipuler. Le client demande combien font 7 et 2, et le service calculator répond 9.

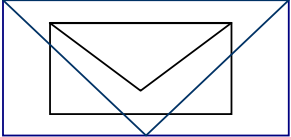
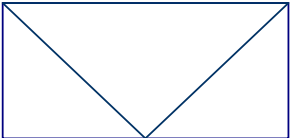
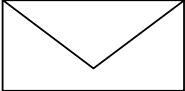


Les informations utiles échangées entre les programmes client et serveur sont empaquetées dans deux enveloppes :

- une enveloppe HTTP qui permet au message de circuler sur un réseau informatique,
- une enveloppe SOAP, qui permet de décrire les informations utiles de façon standard.

Voici les messages que le programme client est capable de créer (requête) ou de recevoir (réponse). Bien sûr, des programmes spécifiques fabriquent ou analysent ces messages, et suivant le langage de programmation ou le RAD<sup>1</sup> que vous utilisez, vos développements consisteront à :

- fournir les données utiles à des classes spécifiques (Java, C++) qui gèreront les deux enveloppes pour vous,
- ou fournir les données utiles et l'enveloppe SOAP (certains RAD) à un programme spécifique qui gèrera l'enveloppe HTTP pour vous.

<b>Gérer par le RAD ou le langage de programmation</b>	<b>A développer</b>	<b>Exemples</b>
	<p>appel à add(7, 2)</p>	<p>Java, Visual C++</p>
	<p>appel à add(7, 2) +</p> 	<p>4D 2003, Omnis 8</p>

<sup>1</sup> *Rapid Application Development* : environnement de développement haut niveau qui permet de développer rapidement des applications informatiques.

## Requête avec son *enveloppe HTTP* et son **enveloppe SOAP** :

```
POST /sigemstest/services/Calculator HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.1
Host: 212.234.37.101
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 420

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <add soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <op1 xsi:type="xsd:int">7</op1>
      <op2 xsi:type="xsd:int">2</op2>
    </add>
  </soapenv:Body>
</soapenv:Envelope>
```

L'enveloppe HTTP est en caractères italiques, l'enveloppe SOAP est en caractères gras. L'enveloppe HTTP contient :

- l'adresse IP de la machine qui héberge le service calculator, à la ligne 5 (*Host*),
- le chemin d'accès au service calculator sur la machine *Host*, à la ligne 1,
- un entête SOAPAction qui contient une chaîne vide, mais qui doit être présent pour échanger des messages SOAP,
- d'autres en-têtes relatifs au contenu du message HTTP.

L'enveloppe SOAP contient :

- un en-tête en caractères gras,
- un corps en caractères gras et italiques. C'est lui qui indique au service l'opération invoquée et les valeurs données aux opérateurs op1 et op2, ici 7 et 2.

## Réponse avec son *enveloppe HTTP* et son **enveloppe SOAP** :

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Connection: close
Date: Tue, 23 Mar 2004 16:02:28 GMT
Server: Apache Tomcat/4.1.27 (HTTP/1.1 Connector)
Set-Cookie: JSESSIONID=2DE97266A48413C8D4AC66845AAFF1E4; Path=/sigemstest

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <addResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <addReturn xsi:type="xsd:int">9</addReturn>
    </addResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

L'enveloppe HTTP nous indique que le service s'est bien déroulé. En effet, la ligne 1 dit « 200 OK ». Le corps du message SOAP indique le résultat attendu. Ici, 9.