Programmation Réseau

URL, SOAP

Jean-Baptiste.Yunes@liafa.jussieu.fr

Coloriages: François Armand
armand@informatique.univ-paris-diderot.fr

UFR Informatique

2011-2012

URI - URN - URL

- Défini dans la RFC 3986 « Uniform Resource Identifier (URI): Generic Syntax », 01/2005

  - C’est en fait une nouvelle version de la RFC 2396 du 08/1998

    - Extension de la RFC 1738 « Unifrom Resource Locators(URL) » du 12/1994

- Le système de désignation des documents via le Web

URL

- URI: Uniform Resource Identifier
- URN: Uniform Resource Name
  - Équivalence : nom d’une personne
- URL: Uniform Resource Locator
  - Équivalence: adresse d’une personne
- Uniform: car des nommages permettent de s’affranchir des mécanismes d’accès (protocoles)
- Resource: ce que l’on souhaite, rien n’est imposé sur ce qu’est une ressource
- Identifier: permet la distinction entre ressources différentes

- Exemples (extraits de la RFC) :
  - ftp://ftp.is.co.za/rfc/rfc1808.txt
  - http://www.ietf.org/rfc/rfc2396.txt
  - ldap://[2001:db8::7]/c=GB?objectClass?one
  - mailto:John.Doe@example.com
  - news:comp.infosystems.www.servers.unix
  - tel:+1-816-555-1212
  - telnet://192.0.2.16:80/
  - urn:oasis:names:specification:docbook:dtd:xml:4.1.2

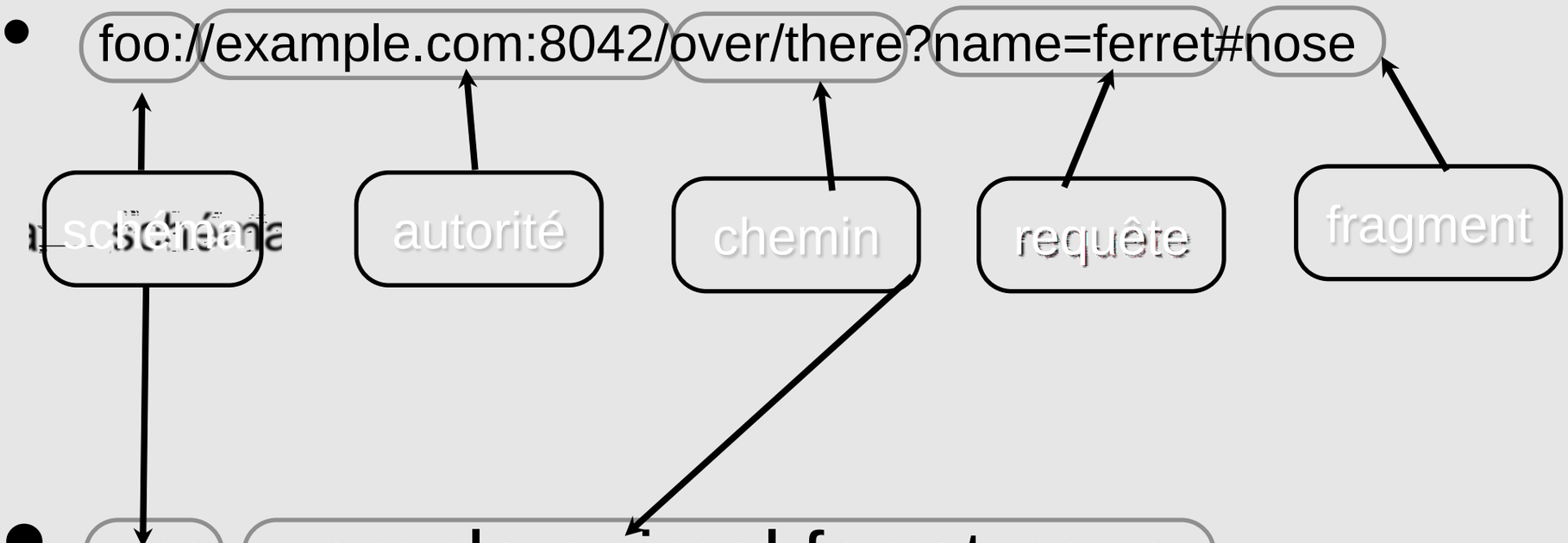
- Les URI regroupent les URN et les URL

URI

- L’URL d’une ressource change quand on déplace la ressource
  - Changement de nom réseau de machine, déplacement du fichier dans l’arborescence..
- L’URN est constant… Mais il faut (faudrait) un mécanisme pour trouver l’URL d’un URN
- En pratique quand un lien est cassé, on utilise un moteur de recherche

- Les URIs se décomposent :

- foo://example.com:8042/over/there?name=ferret#nose



- urn:example:animal:ferret:nose

java.net.URI

- outre les divers constructeurs les méthodes suivantes permettent d'extraire les composants intéressants de l'URI
  - String getScheme()
  - String getAuthority()
  - String getPath()
  - String getQuery()
  - String getFragment()
- plus diverses autres (raffinement)…

java.net.URL

- permet en particulier d'établir une liaison avec la ressource en question, via :
  - URLConnection openConnection();
    - qui permet d'envisager la manipulation de la connexion sous-jacente
  - InputStream openStream();
    - qui permet d'obtenir la ressource sous-jacente

Exemple: lire le contenu d'une ressource dont on a l'URL

```
URL url = new URL(args[0]);
```

```
url.openStream()
```

Exemple:

```
C:> java TestURL file:./TestURL.java
import java.net.*;
import java.io.*;

public class TestURL {
    public static void main(String []args) {
        try {
            URL url = new URL(args[0]);
            BufferedReader bf = new BufferedReader(new
          InputStreamReader(url.openStream())
            String s = bf.readLine();
            while (s!=null) {
                System.out.println(s);
                s = bf.readLine();
            }
            bf.close();
        } catch(Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

Exemple:

```
C:>  java TestURL http://google.fr | more
<!doctype html><html itemscope
      itemtype="http://schema.org/WebPage"><head>…..
```

- On récupère le source HTML de la page définie par l'URL
  - On ne voit rien du protocole utilisé pour accéder à cette ressource.

java.net.URLConnection

- permet de récupérer les en-têtes séparément
- du contenu
  - vers lequel on peut possiblement lire et écrire via :
    - des flots Java
- pour la lecture, on peut extraire l'Object correspondant à la ressource

- les en-têtes peuvent récupérées via :
- Map<String,List<String>> getHeaderFields();
  - attention : certains en-têtes n’ont pas de label, une clé « null » leur correspond

Exemple2:

```
Map<String,List<String>> h

URL url = new URL(args[0]);
URLConnection urlc = url.openConnection();
   urlc.getHeaderFields()
```

Exemple2:

```
C/>  java TestURL2 http://google.fr
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Date: Sat, 05 May 2012 14:27:50 GMT
Transfer-Encoding: chunked
P3P: CP="This is not a P3P policy! See
      http://www.google.com/support/accounts/bin/answer.py?hl=en&answe
      r=151657 for more info."
Expires: -1
X-XSS-Protection: 1; mode=block
Set-Cookie:
      NID=59=CZRXNM47M39wq761DwGjybeDPNculeaa4UEUQ_ffMNjjOt
      BW2a3acFO1POmJ-
      gsZ3o3DjEgZ7EMdYsbDXmeNs72GxpJvCdFyZsv9gY3GsKbDB4seX
a43U7perYmNLOUr; expires=Sun, 04-Nov-2012 14:27:50 GMT; path=/;
      domain=.google.fr; HttpOnly
PREF=ID=f6a8dd07d29c9551:FF=0:TM=1336228070:LM=1336228070:S
      =34ZX9WAPGNCIDDOo; expires=Mon, 05-May-2014 14:27:50 GMT;
      path=/; domain=
.google.fr
Content-Type: text/html; charset=ISO-8859-1
Server: gws
Cache-Control: private, max-age=0
class sun.net.www.protocol.http.HttpURLConnection$HttpInputStream
```

Exemple2:

```
C/> java TestURL2 file:./TestURL.java
```

- des informations peuvent être récupérées :
  - qui concernent la connexion elle-même
    - propriétés
  - qui concernent la ressource
    - méta-information

- propriétés de la connexion :
  - boolean getAllowUserInteraction();
  - int getConnectTimeout();
  - boolean getDoInput();
  - boolean getDoOutput();
  - int getReadTimeout();
  - boolean getUseCaches();

- méta-information :
  - String getContentEncoding();
  - int getContentLength();
  - String getContentType();
  - long getDate();
  - long getExpiration();
  - long getIfModifiedSince();
  - long getLastModified();

- pour le contenu :
  - soit au plus bas-niveau avec :
    - InputStream getInputStream();
    - OutputStream getOutputStream();
  - soit en instanciant une classe appropriée si possible :
    - Object getContent();

Exemple3:

```
URL url = new URL(args[0]);
URLConnection urlc = url.openConnection()


InputStream is = urlc.getInputStream();
```

Exemple4:

```
URL url = new URL(args[0]);
URLConnection urlc = url.openConnection();

Object o = urlc.getContent();


                urlc.getAllowUserInteraction();
            urlc.getConnectTimeout();
            urlc.getContentLength();
```

Exemple4:

```
C:/> java TestURL4 http://google.fr
```

```
C:/> java TestURL4 file:./TestURL.java
```

libcurl

- Manipuler des ressources accédées par URL en C
- libcurl : http://sourceforge.net/projects/curl/
  - DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, Telnet and TFTP
  - + SSL +…
  - « Bindings » vers ~40 langages!
- Voir la page sur les « compétiteurs »!

Exemple ftpget.c

```
/* Copyright (C) 1998 - 2011, Daniel Stenberg, <daniel@haxx.se>, et al.
 * This software is licensed as described in the file COPYING, which
 * you should have received as part of this distribution. The terms
 * are also available at http://curl.haxx.se/docs/copyright.html.
 *
 * You may opt to use, copy, modify, merge, publish, distribute and/or sell *
 * copies of the Software, and permit persons to whom the Software is
 * furnished to do so, under the terms of the COPYING file.
 *
 * This software is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY
 * KIND, either express or implied.
```

Libcurl

```
curl_global_init(CURL_GLOBAL_DEFAULT);
curl = curl_easy_init();

curl_easy_setopt(curl, CURLOPT_URL,
                 "ftp://ftp.fa.fr/pub/toto");

curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION,
                 my_fwrite);

curl_easy_setopt(curl, CURLOPT_WRITEDATA, &ftpfile);
```

```
curl_easy_setopt(curl, CURLOPT_VERBOSE, 1L);
curl_easy_perform(curl);
curl_easy_cleanup(curl);
curl_global_cleanup();
```

Libcurl

- SOAP

Problèmes des RPC

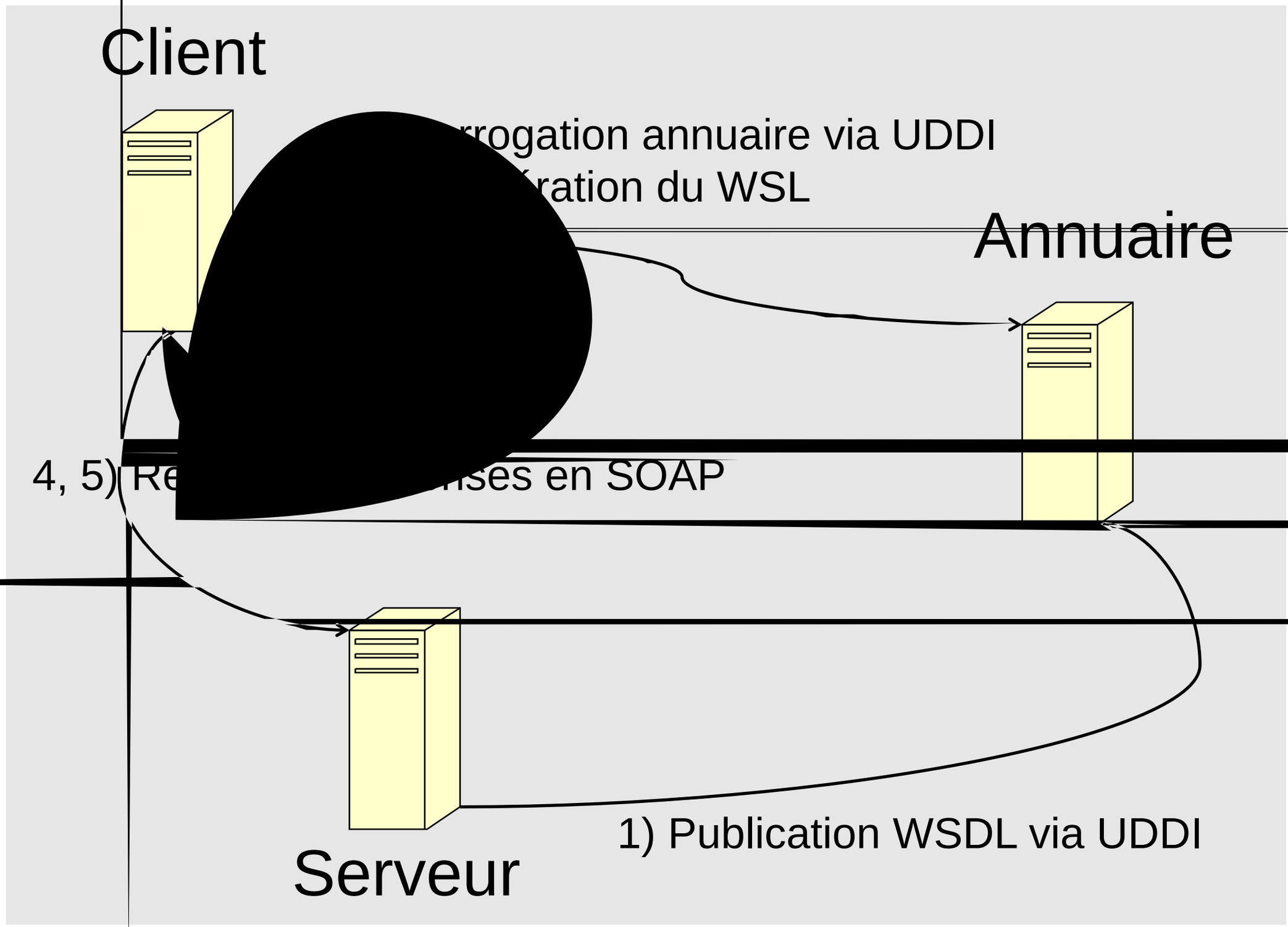
- RPC ou RMI
  - Pratique pour invoquer des méthodes, fonctions, services à distance…
  - L'OMG (Object Management Group) a défini des standards (CORBA) pour ce genre de mécanismes
    - Transports basés sur IIOP / GIOP
    - OK pour réseaux locaux (intra entreprise)
    - KO pour internet (problème de firewall)

Solution : SOAP

- Utiliser HTTP (ou HTTPS) comme transport
- Résultats sont fournis par des services Web
- => Interagir avec les services Web sous une forme classique de RPC

- SOAP: RPC vers des WEB services
  - HTTP + XML
    - HTTP comme transport
    - XML pour sérialiser / désérialiser les données

- Initialement **S**imple **O**bject **A**ccess **P**rotocol
- Complété par
  - WSDL:
    - Web Service Description Language
  - UDDI:
    - Universal Description, Discovery and Integration


Client
rrogation annuaire via UDDI
ration du WSL
Annuaire
4, 5) Re nses en SOAP
1) Publication WSDL via UDDI
Serveur

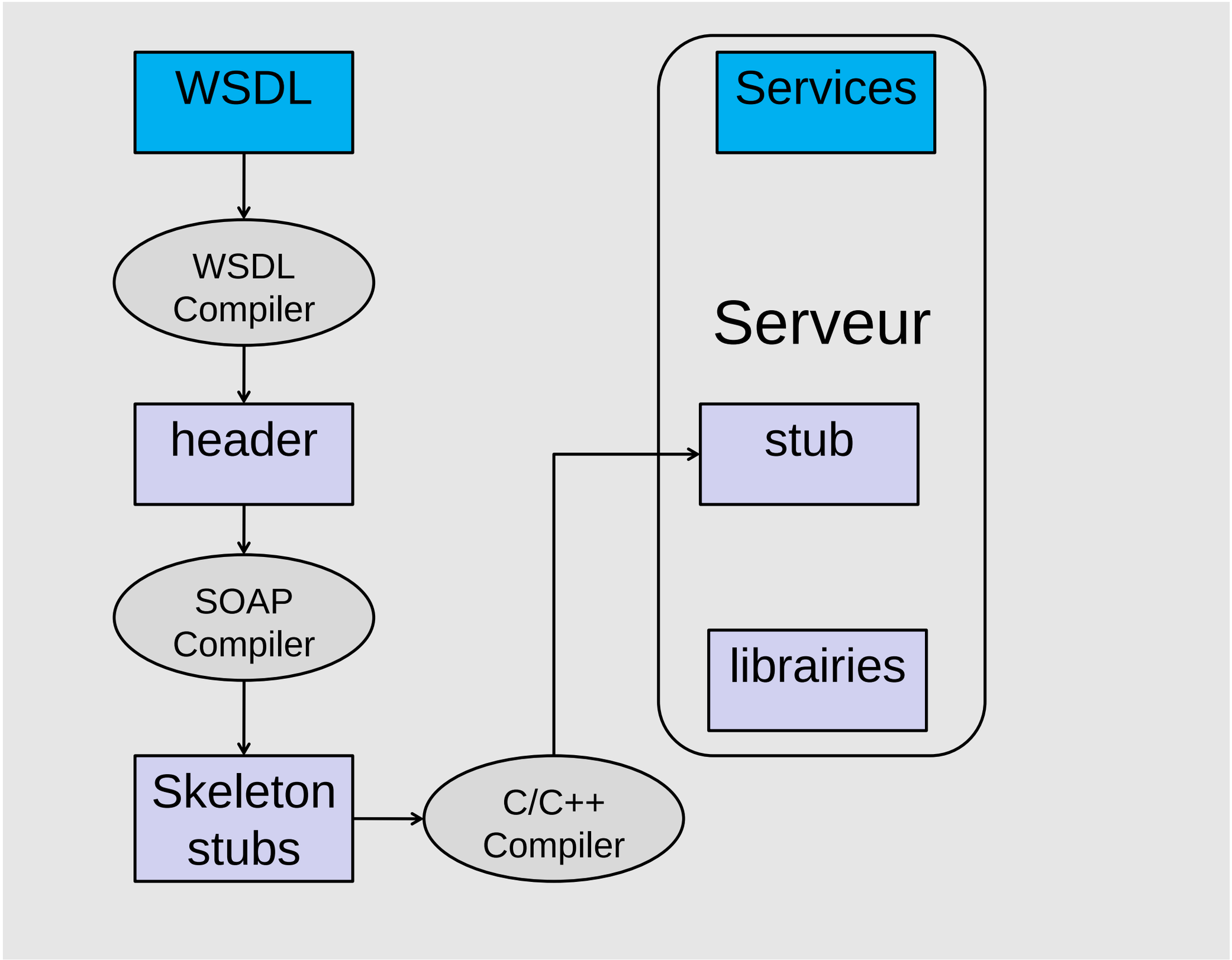
SOAP / Java: un exemple

- En général, on utilise des « framework » qui cachent la mécanique SOAP
  - De la même manière que les mécaniques RPC ou RMI sont masquées au programmeur
- Suivant les frameworks les approches sont variables
  - class Java => WSDL / Soap
    - Ex: AXIS2 java2wsdl
  - WSDL/SOAP => class Java
    - Ex: AXIS2 wsdl2java

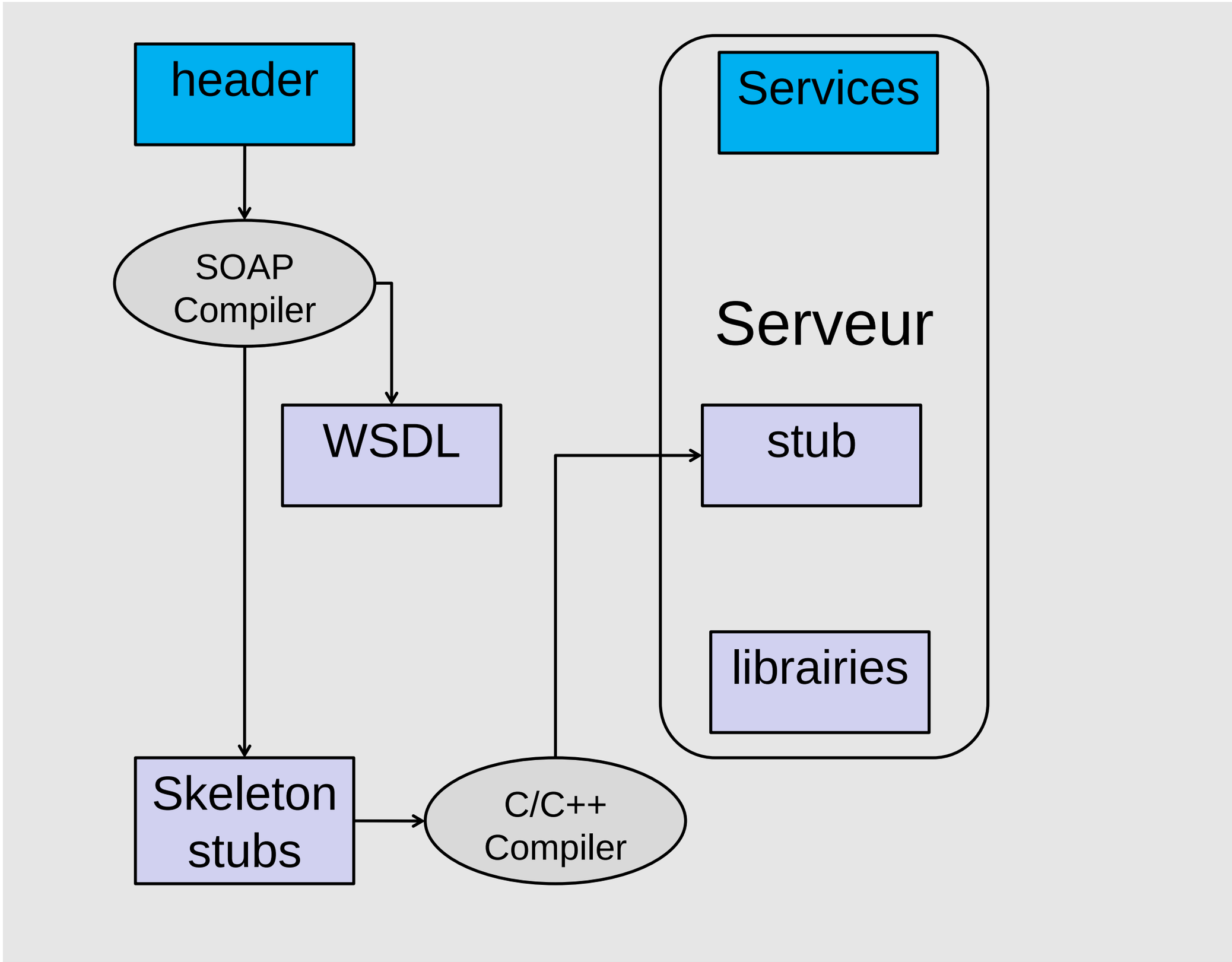
SOAP / C: un exemple

- gSOAP
- http://www.cs.fsu.edu/~engelen/soap.html
- On peut partir d'un WSDL
- On peut aussi partir de headers en C!

SOAP / C: un exemple



SOAP / C: un exemple



Calc.wsdl

Calc.wsdl

Calc.h

```
// Contents of file "calc.h":
//gsoap ns service name: calculator
//gsoap ns service style: rpc
//gsoap ns service encoding: encoded
//gsoap ns service port: http://mydomain/path/calculator.cgi

//gsoap ns service namespace: urn:calculator

int ns__add(double a, double b, double &result);
```

Calcclient.c

```
#include "soapH.h“
#include "calc.nsmap"

const char server[] = "http://websrv.cs.fsu.edu/~engelen/calcserver.cgi";

int main(int argc, char **argv)
{
        struct soap soap;
        double a, b, result;

        soap_init1(&soap, SOAP_XML_INDENT);
        a = strtod(argv[1], NULL);    b = strtod(argv[2], NULL);

        soap_call_ns__add(&soap, server, "", a, b, &result );

        if (soap.error)  {
                soap_print_fault(&soap, stderr);  exit(1);
        }  else    printf("result = %g\n", result);

        soap_destroy(&soap);
        soap_end(&soap);
        soap_done(&soap);

        return 0;
}
```

Calcserver.c

```
#include "soapH.h“
#include "calc.nsmap“
int main(int argc, char **argv)
{
        SOAP_SOCKET m, s; /* master and slave sockets */
        struct soap soap;
        soap_init(&soap);
        if (argc < 2)
                soap_serve(&soap);          /* serve as CGI application */
        else  {
                m = soap_bind(&soap, NULL, atoi(argv[1]), 100);
                if (!soap_valid_socket(m))    {
                        soap_print_fault(&soap, stderr);      exit(-1);
                }
                fprintf(stderr, "Socket connection successful: master socket = %d\n", m);
                for ( ; ; )    {
                        s = soap_accept(&soap);
                         fprintf(stderr, "Socket connection OK: slave socket = %d\n", s);
                        if (!soap_valid_socket(s))      {
                                soap_print_fault(&soap, stderr);        exit(-1);
                         }
                        soap_serve(&soap);
                        soap_end(&soap);
                 }
         }
        return 0;
}
```

Calcserver.c

```
int ns__add(struct soap *soap, double a, double b, double *result)
{
	*result = a + b;
	return SOAP_OK;
}
```

- Attention!
- Les données sont sérialisées avec XML
  - Verbeux, voire très verbeux
    - Chaque élément d’un tableau (de caractères) par exemple va être encapsulé dans <tag>élément</tag>
- On peut compresser (gSoap)… on gagne du volume à transporter, mais on perd tu temps!

- Solutions alternatives
  - JSON
  - Google Protocol Buffers
  - XML binaire
  - …