

M3102

Services réseaux

Les services

Introduction

L'Internet est essentiellement connu par le grand nombre de services qui y sont proposés:

- DNS : Fonctions de nommage
- TELNET : Accès à des machines distantes (1971)
- SMTP : Messagerie
- POP : Téléchargement des mail
- FTP : Transfert de fichiers (1971)
- WEB / HTTP : Echange d'informations via le Web

Mais aussi

- SNMP : Administration du réseau
- SLP : Protocole Localisation de Service
- IPP : Protocole d'Impression Internet
- SSH : Sécurisation couche Transport
- NTP : Synchronisation des horloges des ordinateurs dans l'Internet

...

Introduction... quelques dates

1966 - 1969 : Préparation du projet Arpanet (réflexions, appel d'offre construction des commutateurs).

1969 : Arpanet fonctionne entre 4 commutateurs.

1971 : 15 commutateurs, 24 hôtes

1972 : Spécification du premier accès distant

1973 : Spécification du premier transfert de fichiers

1974 : Première utilisation du terme **Internet**.

1977 : RFC messagerie électronique -> **smtp**.

1978 : Travaux sur TCP/IP séparation TCP et IP

1982 : Normalisation complète de TCP IP.

1983 : Séparation Milnet. et Arpanet -> Internet

1984 : Première version de **dns**.

1986 : Apparition des **news**.

1991 : Apparition du **web (html, http)** (CERN Genève)

2000 : Des centaines de milliers de réseaux et des centaines de millions d'hôtes.

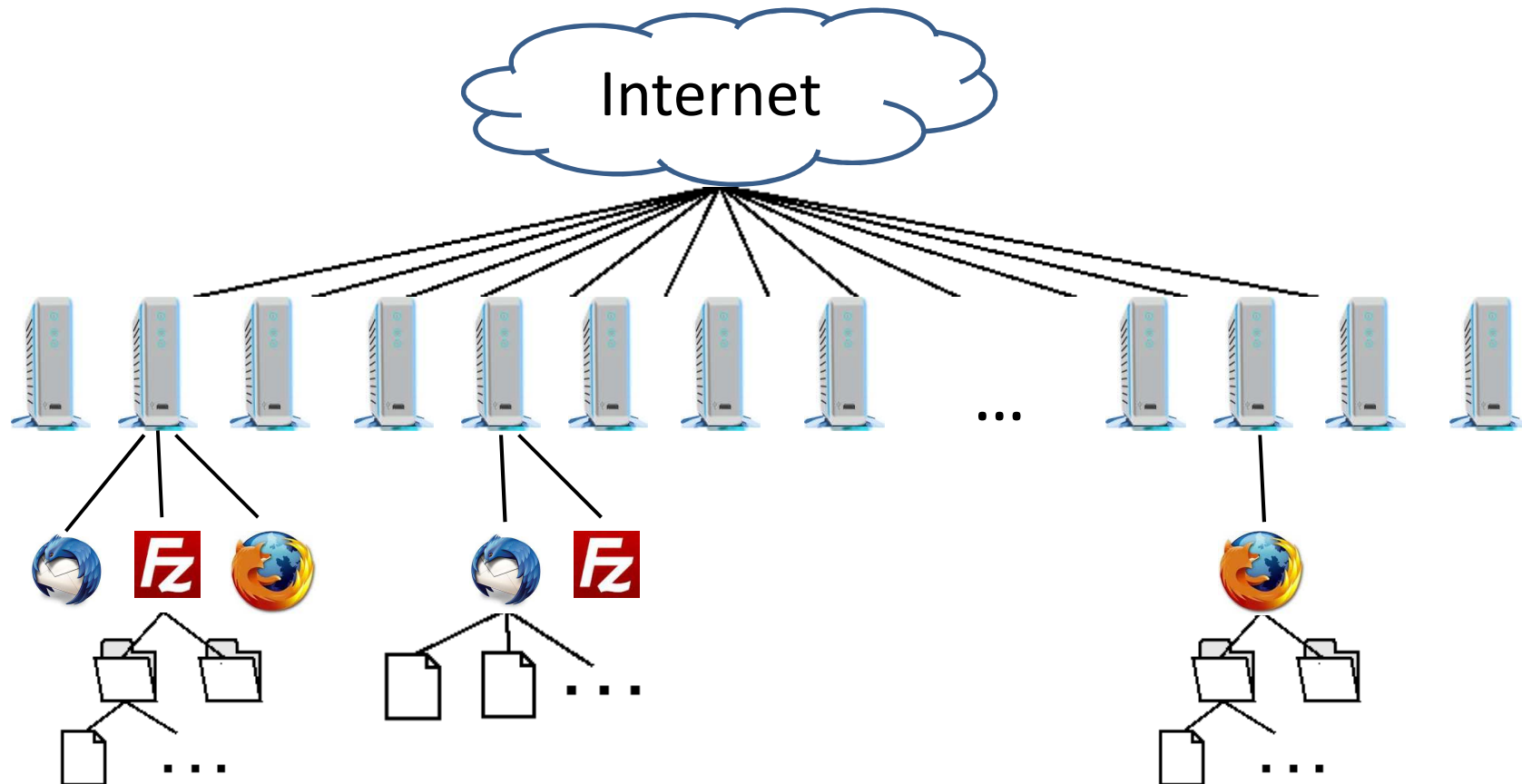
2008 : Des millions de réseaux et 1.5 milliards d'hôtes

2014 : Des dizaines de millions de réseaux et 2.4 milliards d'hôtes

La désignation des services
ou URL

Désignation

L'internet est une espace, permettent l'accès à des centaines de services et des millions de documents associés à ces services.



Pour accéder à un service et/ou à une ressource de ce service, il est nécessaire de les désigner sans ambiguïté dans cet espace.

Désignation

Pour désigner une ressource il faut préciser :

1 - Le type d'application concernée

→ Protocole (HTTP, FTP, SMTP, ...)

2 - La machine

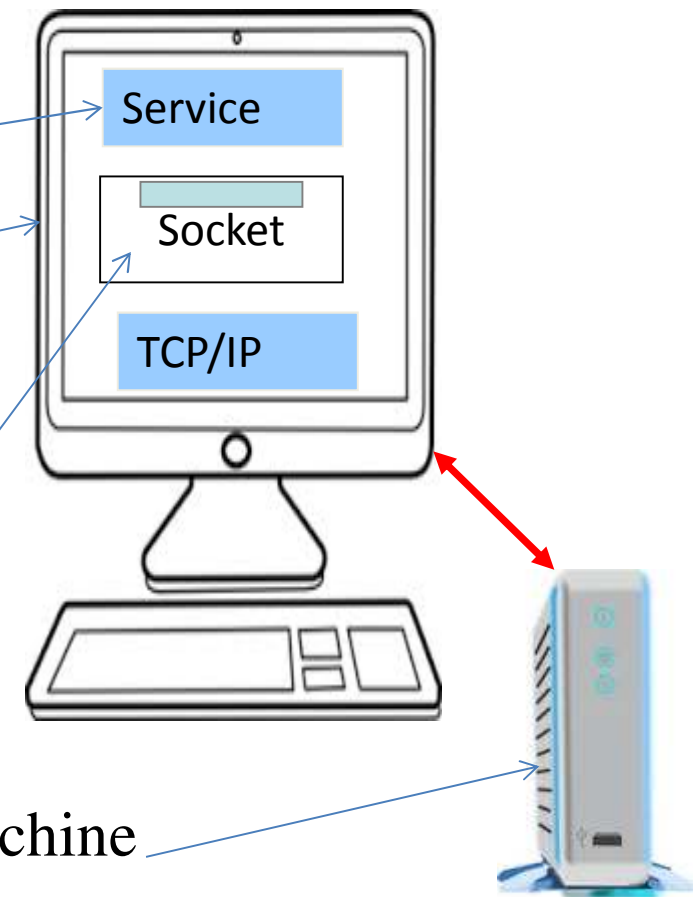
→ @IP, nom.machine,

3 - Le moyen d'accéder à la ressource

→ Numéro de ports (80, 53, 20, ...)

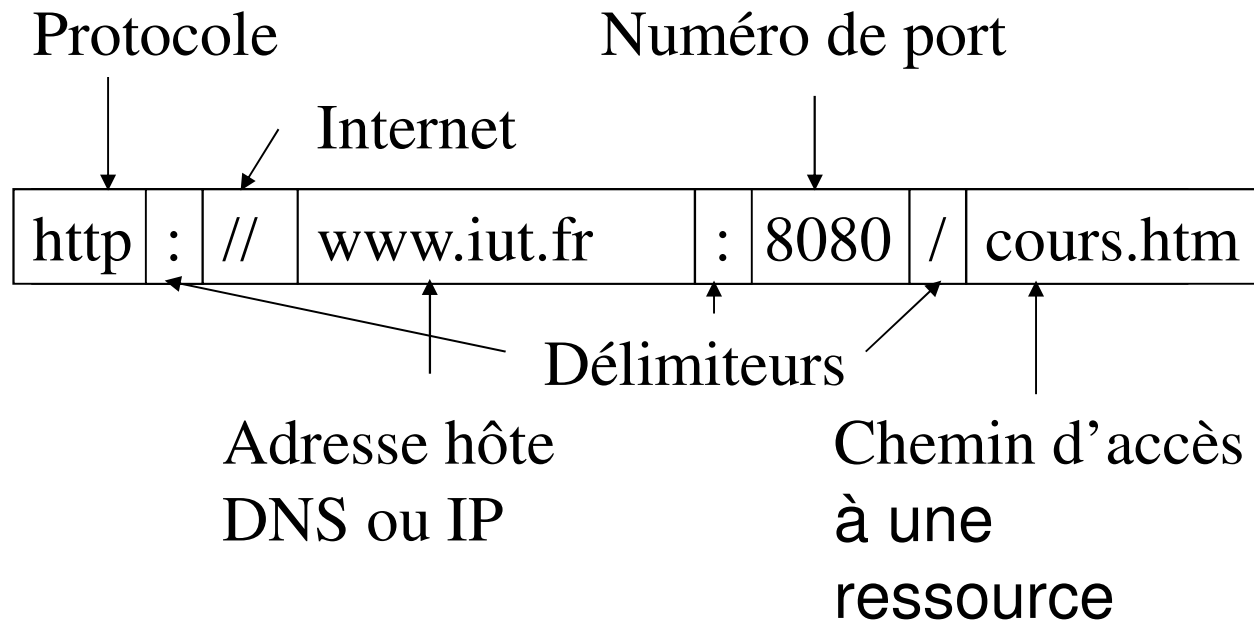
4 - La localisation de la ressource dans la machine

→ /home/name/fic, ...



Désignation

Le schéma d'adressage doit reprendre tous ces éléments.



URL : Localisateur Ressource Unifié

Les services et le besoin de normalisation

Besoin de normaliser

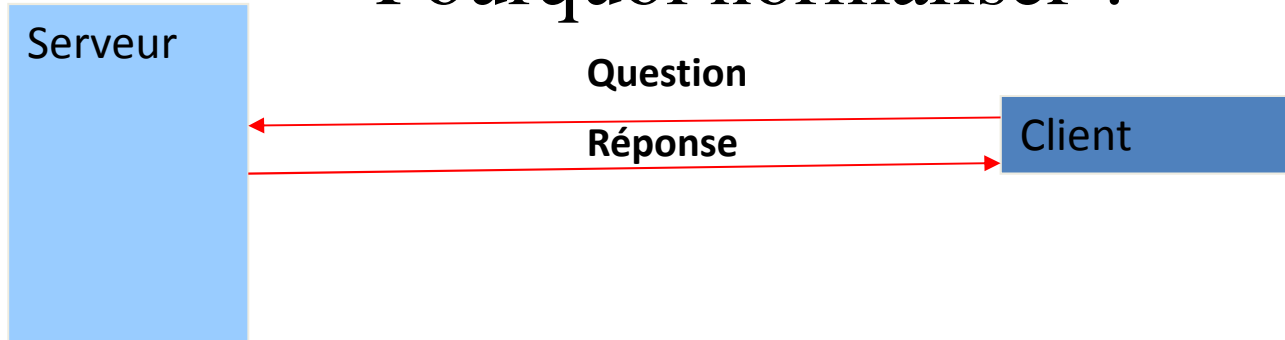
L'Internet est soumis à un contrôle international de plus en plus rigoureux.

→ Il est indispensable de normaliser ces applications que l'on nomme
SERVICES.

	Applications TCP/IP directes				Applications pile SUN/OS
7. Application	DNS: Domain Name System	EXEMPLES		FTP: File Transfer Protocol	NFS: "Network File System"
6. Présentation		SMTP: Simple Mail Transfer Protocol	HTTP: Hyper Text Transfer Protocol		XDR:"External Data Representation"
5. Session					RPC:"Remote Procedure Call"
4. Transport	TCP: Transmission Control Protocol (connecté) UDP: User Datagram Protocol (non connecté)				
3. Réseau	IP: Internet Protocol				
2. Liaison	Encapsulation IP (sur LAN ou liaisons SLIP,PPP)				
	Pratiquement tout support de transmission				
1. Physique	Réseaux Publics	Lignes spécialisées Point à Point	Réseaux Locaux	Réseau téléphonique RNIS, ATM	

Besoin de normalisation

Pourquoi normaliser ?



Sur internet , on déploie des applications client/serveur.

Ces applications doivent :

- se synchroniser parfaitement
- utiliser les mêmes formats de données

Mais ces applications sont développées:

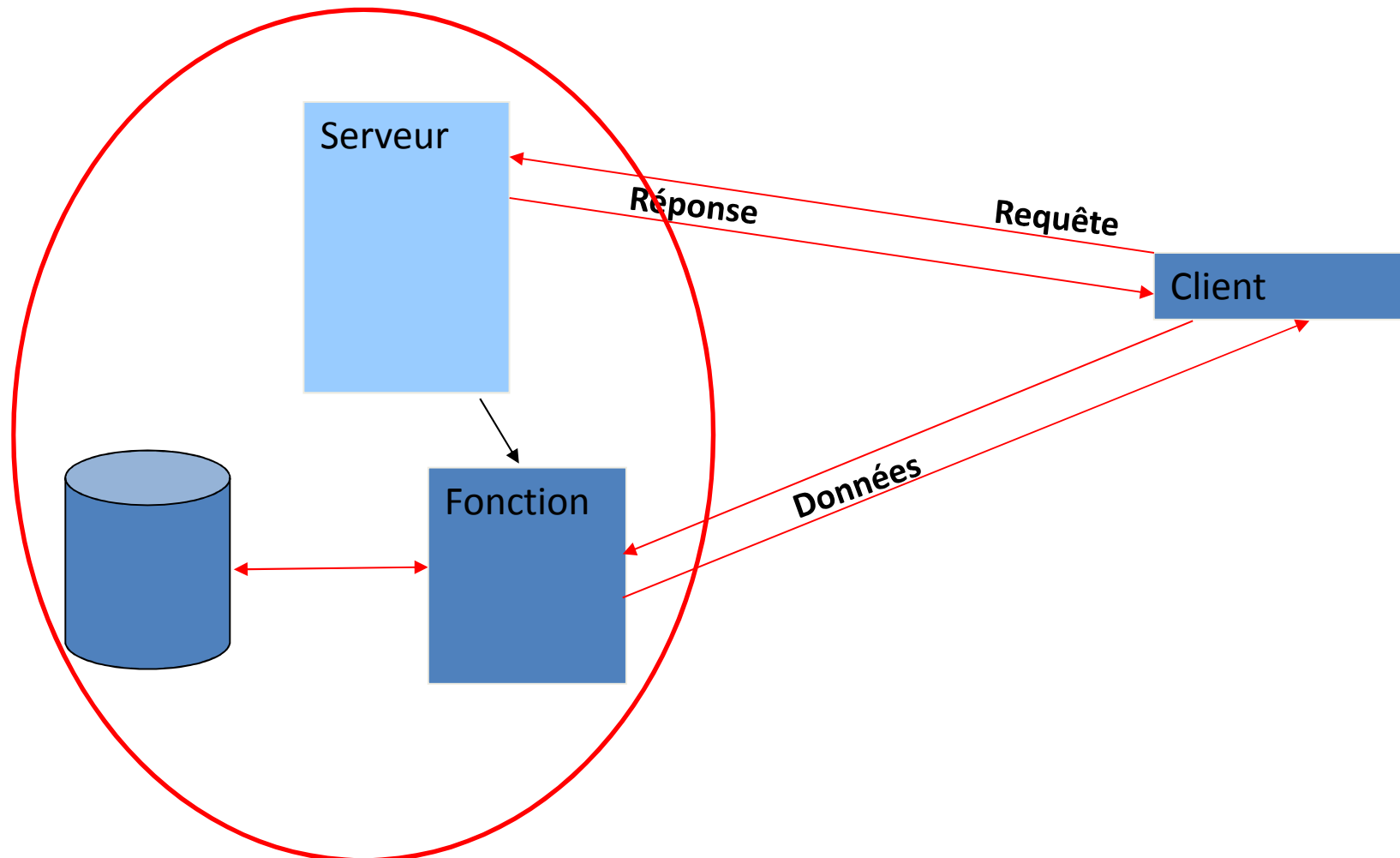
- sur les machines différentes
- par des personnes qui ne se concertent pas
- avec des langages différents

→ Nécessité de définir des fonctionnalités communes

Besoin de normalisation

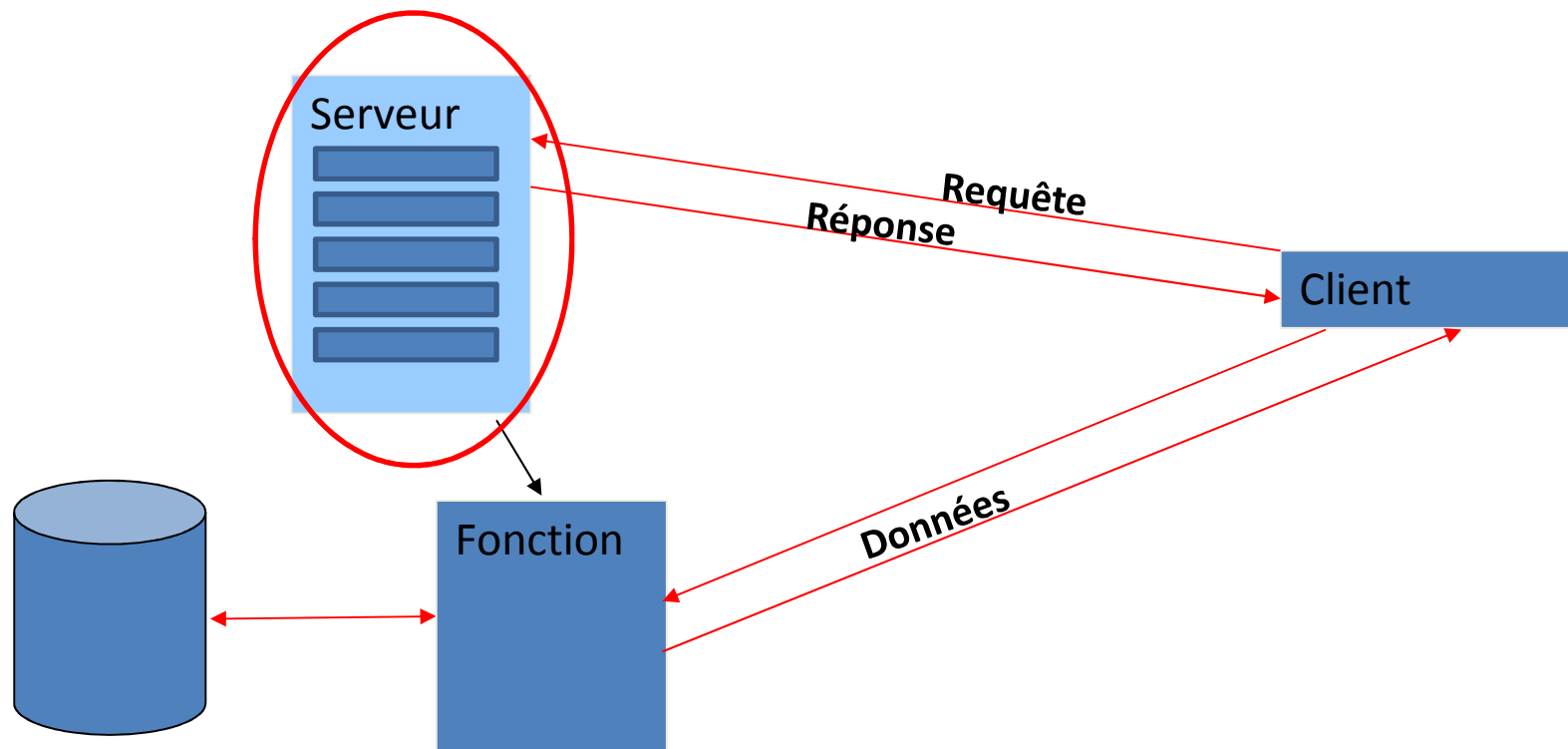
Normaliser un service, consiste :

1 - A définir ce que le serveur va faire.



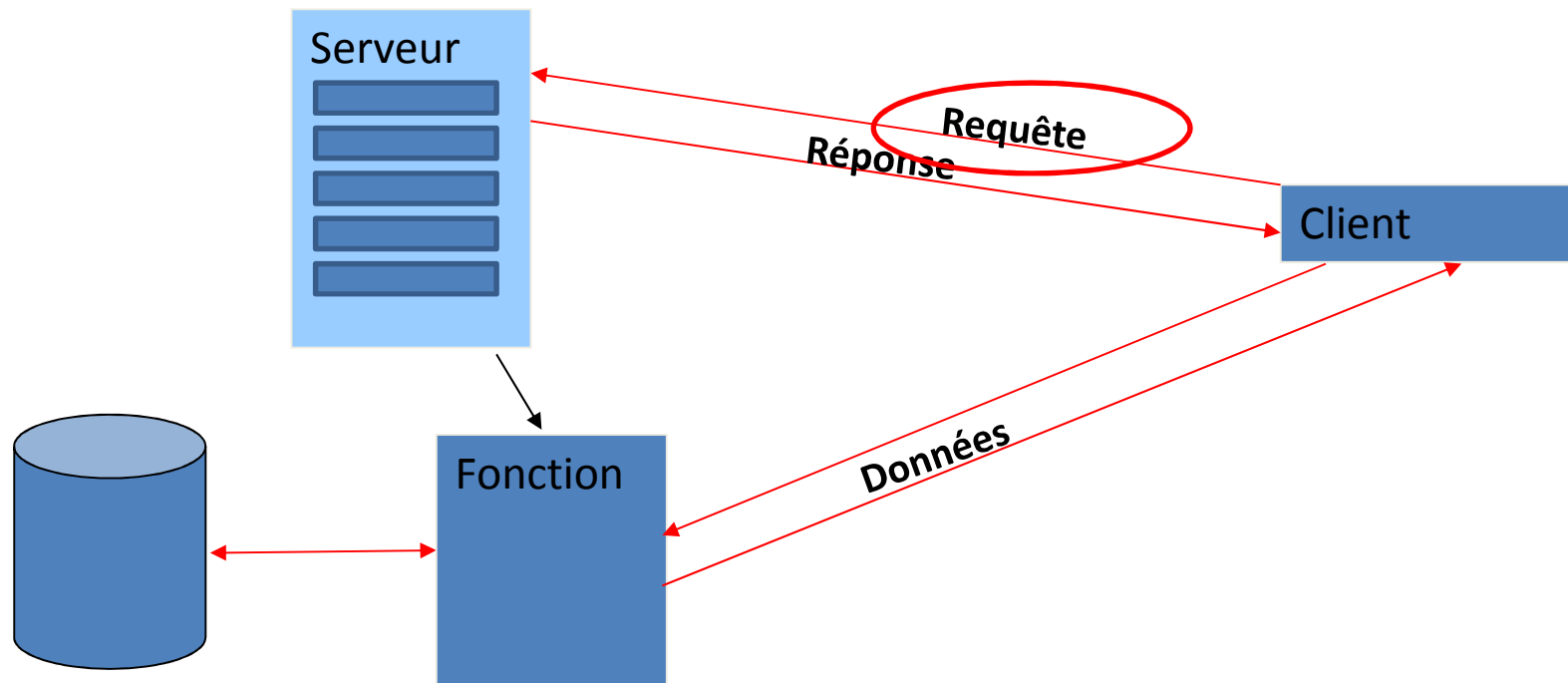
Besoin de normalisation

2 - Définir les fonctions accessibles aux clients (Méthodes ou sous programmes)



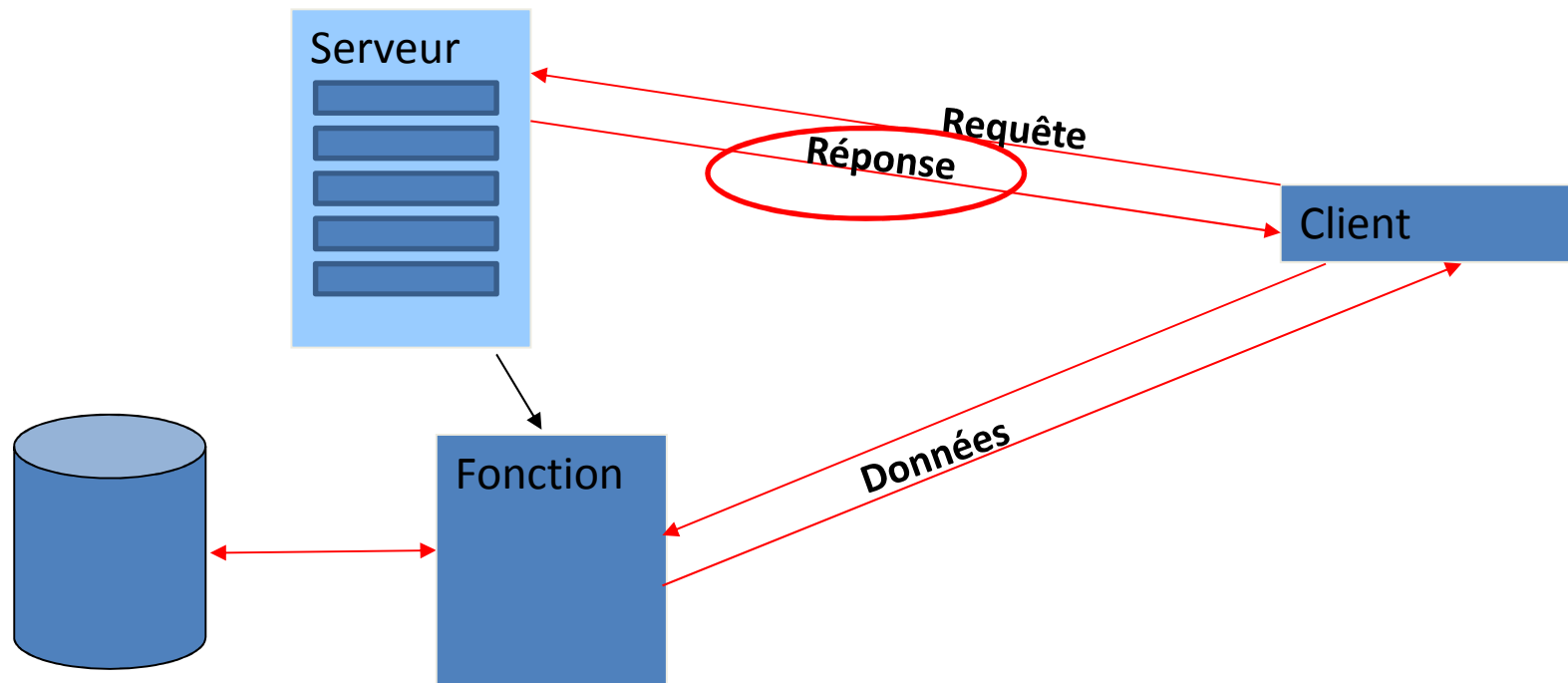
Besoin de normalisation

3 - Définir la façon d'accéder à ces méthodes (Requêtes)



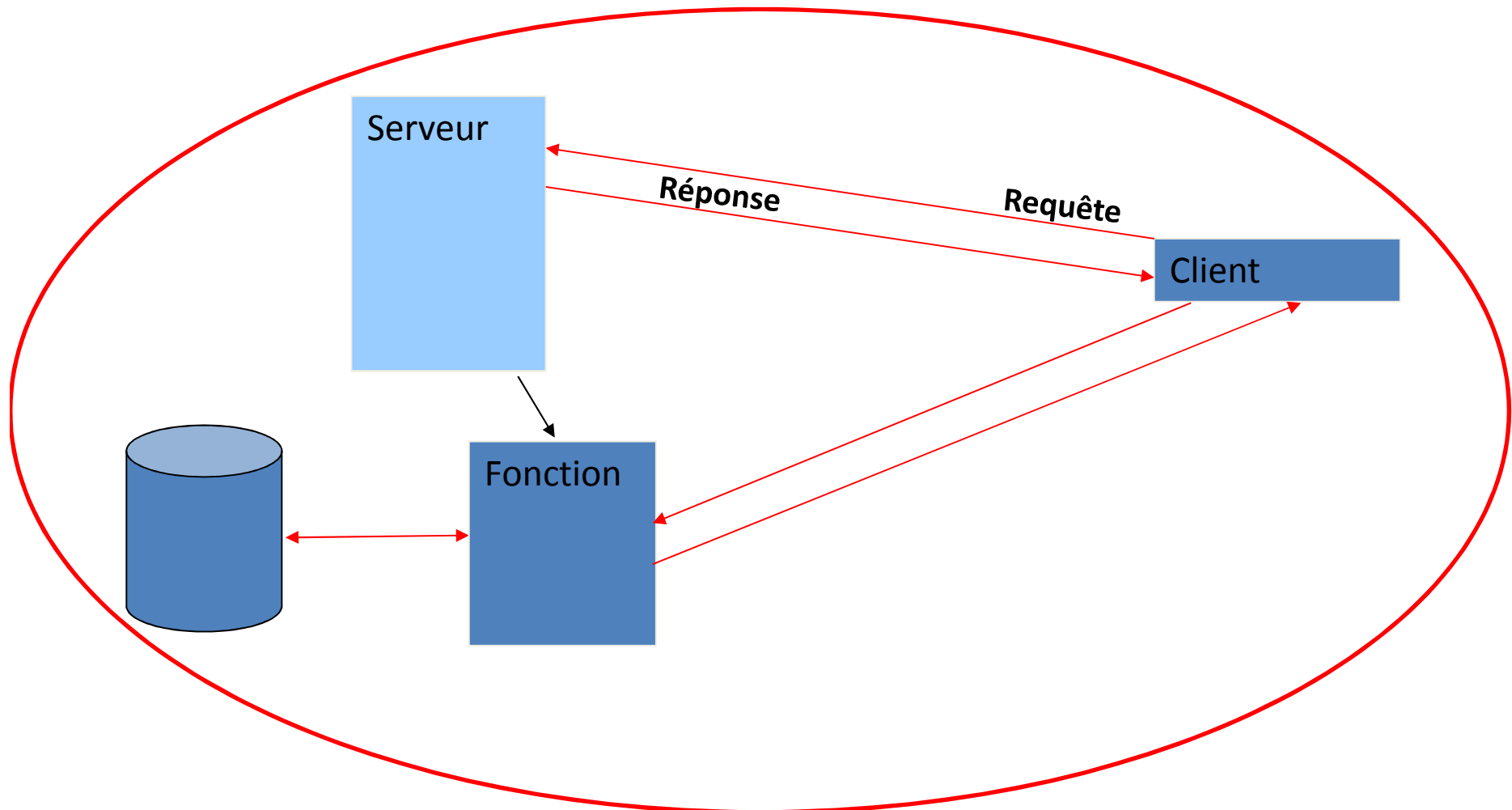
Besoin de normalisation

4 - Définir les réponses possibles aux requêtes



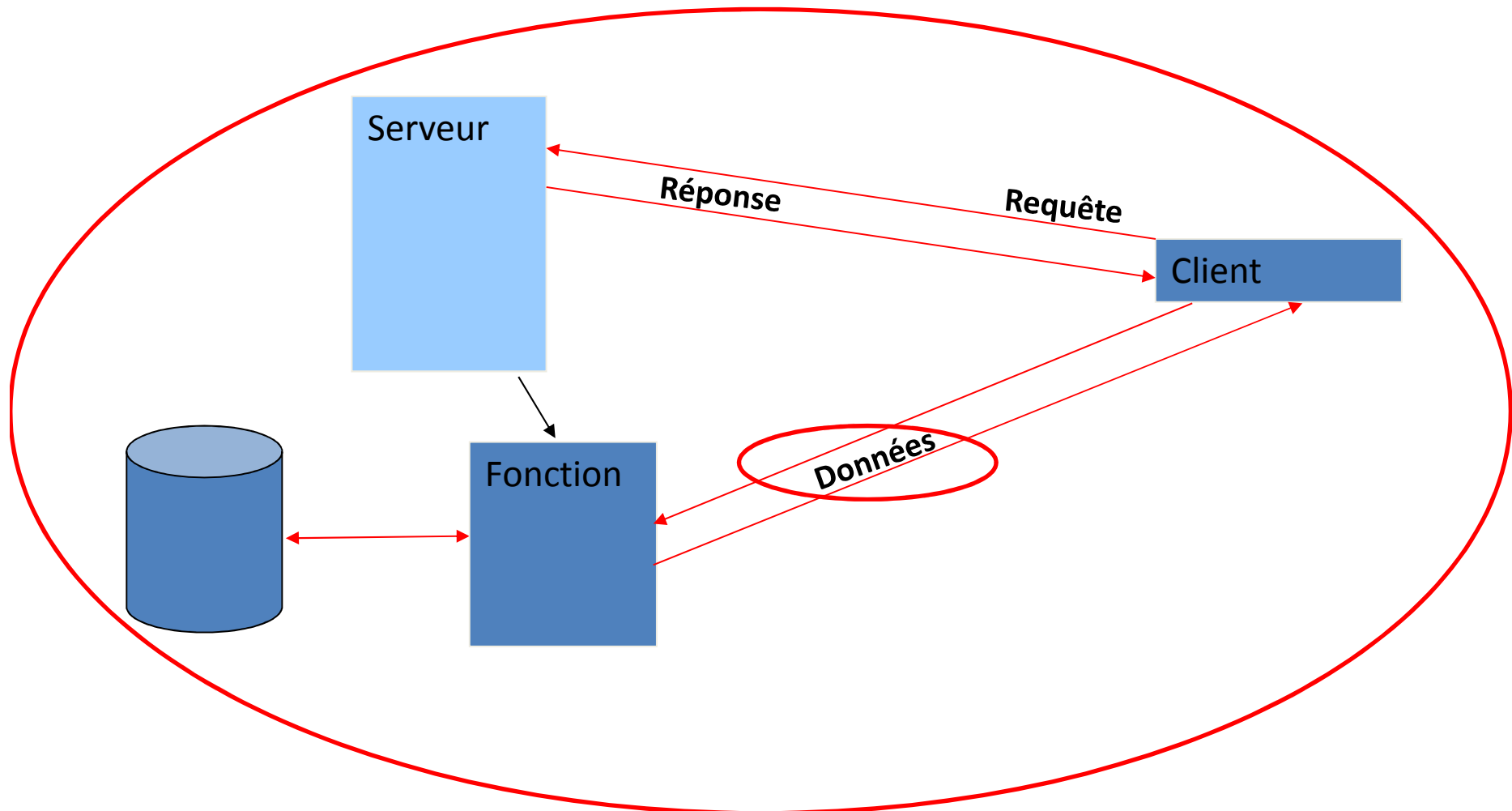
Besoin de normalisation

5 - Définir le comportement général de l'application



Besoin de normalisation

6 - Définir le format des données échangées



Description des services :
RFC (Requests For Comment)

Structure d'un service et RFC

Les fonctionnalités de chaque service sont décrites dans des RFC.

Qu'est-ce qu'un RFC ? série numérotée de documents officiels décrivant les aspects techniques d'Internet (Wikipédia)

RFC 959 STD 9	FTP - File Transfer Protocol Protocole de la couche applicative. Utilisé pour le transfert fiable de fichiers sur Internet.
RFC 854 STD 8	TELNET - Protocole TELNET Protocole de la couche applicative. Utilisé pour se connecter à un serveur distant, Terminal Virtuel Internet.
RFC 830	DNS - Système réparti pour le service des noms Internet
RFC 827	EGP - Protocole de passerelle extérieure (EGP)
RFC 826 STD 37	ETHERNET - Protocole de résolution d'adresse Ethernet Conversion des adresses de protocole réseau en adresses Ethernet à 48 bits pour la transmission sur matériel Ethernet
RFC 821 STD 10	SMTP - Simple Mail Transfer Protocol Protocole de la couche applicative. Utilisé pour envoyer des e-mails par les logiciels de messagerie électronique (KMail, Messenger, etc.)
RFC 819	DNS - Convention de désignation de domaine pour les applications d'utilisateur de l'Internet

Structure d'un service et RFC

Les RFC sont rédigées sur l'initiative d'experts techniques, puis sont revues par la communauté Internet dans son ensemble.

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June 1999

Plusieurs liens existent :

<http://www.rfc.fr/>

<http://abcdrfc.free.fr/>

<http://www.ietf.org/rfc.html>

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

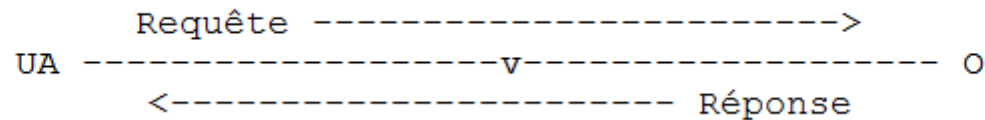
Structure d'un service et RFC

Dans les RFC on trouve :

1 – La façon d'accéder au service (URL)

```
http_URL          = "http:" "://" host [ ":" port ] [ chem_abs ]  
  
host              = <un nom Internet d'ordinateur valide ou une adresse IP  
                  (sous forme numérique), comme définie en Section 2.1 de la  
                  RFC 1123>  
  
port              = *DIGIT
```

2 – La structure générale des échanges



Structure d'un service et RFC

3 – La liste des méthodes

```
Méthode          = "GET"                ; Section 8.1  
                  | "HEAD"              ; Section 8.2  
                  | "POST"              ; Section 8.3  
                  | nom_de_méthode
```

4 – La description des méthodes

8.1 GET

La méthode GET signifie "récupérer" le contenu quel qu'il soit de la ressource (sous forme d'une entité) identifiée par l'URI-visée. Si l'URI-visée identifie un processus générant dynamiquement des données, ce sont les données produites qui sont renvoyées dans l'entité au lieu du source de l'exécutable appelé, sauf si ce texte lui-même est la sortie du processus.

Structure d'un service et RFC

5 – Les réponses

Les codes réponses informent de la suite donnée à la requête

→ Code + libelle

Le premier chiffre définit la catégorie:

1xx un message informatif.

2xx une opération réussie.

3xx une redirection vers une autre URL

4xx une erreur du client

5xx une erreur du serveur

Structure d'un service et RFC

6 – La liste des réponses

Code_état	= "200" ; OK	OK
	"201" ; Created	Créé
	"202" ; Accepted	Accepté
	"204" ; No Content	Pas de contenu
	"301" ; Moved Permanently	Changement définitif
	"302" ; Moved Temporarily	Changement temporaire
	"304" ; Not Modified	Non modifié
	"400" ; Bad Request	Requête incorrecte
	"401" ; Unauthorized	Non autorisé
	"403" ; Forbidden	Interdit
	"404" ; Not Found	Non trouvé
	"500" ; Internal Server Error	Erreur interne serveur
	"501" ; Not Implemented	Non implémenté
	"502" ; Bad Gateway	Erreur de routeur
	"503" ; Service Unavailable	Indisponible

7 – La description des réponses

9.2 Succès 2xx

Cette classe précise que la requête du client a été correctement transmise, interprétée, et exécutée.

200 OK

La requête a abouti. L'information retournée en réponse dépend de la requête émise, comme suit:

Structure d'un service et RFC

8 – Le format détaillé des requêtes et des réponses

Requête_complète	= Ligne_requête	; Section 5.1
	*(En-tête_générale	; Section 4.3
	En-tête_requête	; Section 5.2
	En-tête_entité)	; Section 7.1
	CRLF	
	[Corps_entité]	; Section 7.2
Réponse_complète	= Ligne-état	; Section 6.1
	*(En-tête_générale	; Section 4.3
	En-tête_réponse	; Section 6.2
	En-tête_entité)	; Section 7.1
	CRLF	
	[Corps entité]	; Section 7.2

Structure d'un service et RFC

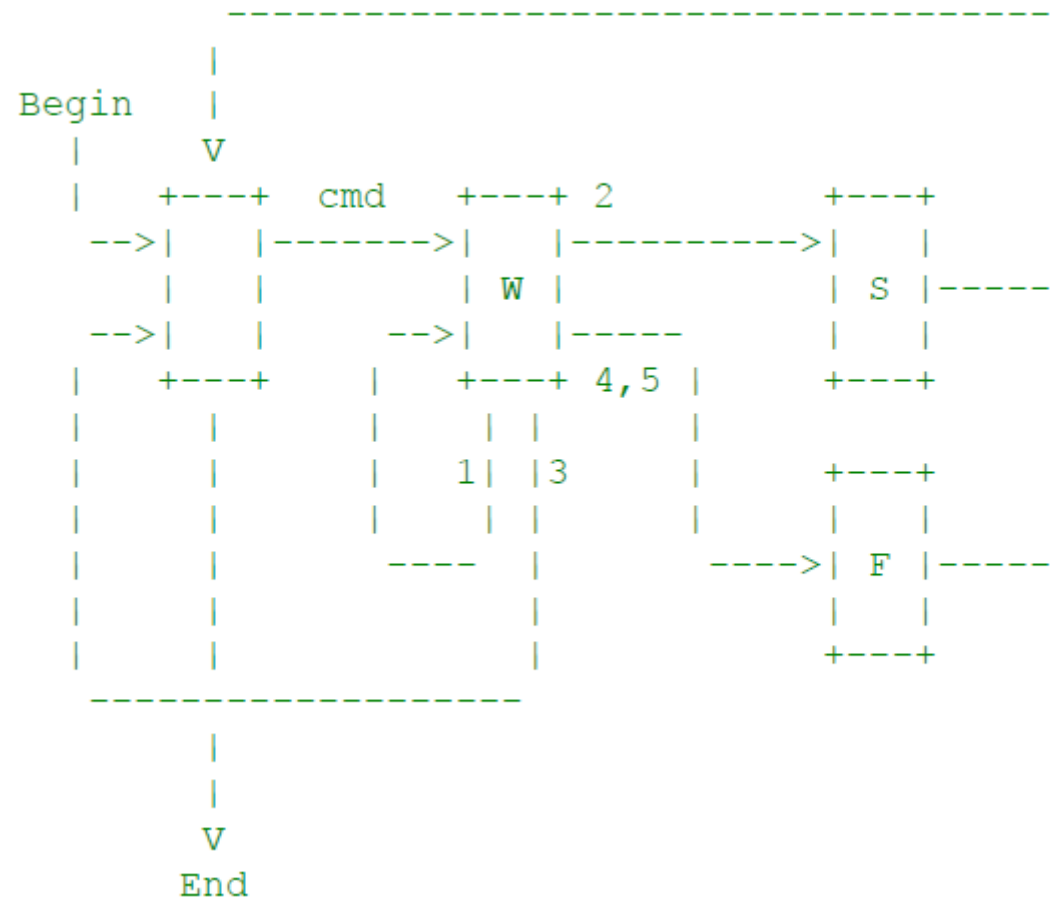
9 – Le format des données

```
+-----+-----+-----+
| Descripteur |      Compte      |
|  code=16   |      = 6      |
+-----+-----+-----+
```

```
+-----+-----+-----+
| Marqueur | Marqueur | Marqueur |
|  8 bits  |  8 bits  |  8 bits  |
+-----+-----+-----+
```

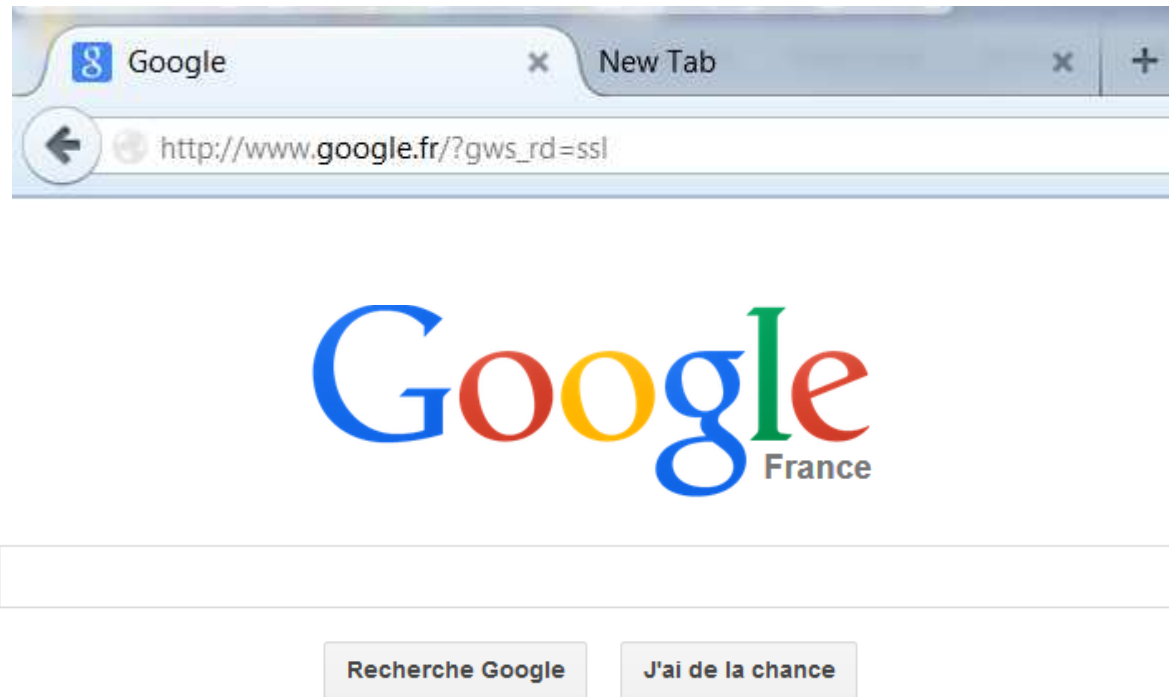
Structure d'un service et RFC

10 – Les échanges types – Diagrammes d'états



Exemple de service :
HTTP

Le protocole HTTP



Protocole de communication client-serveur défini à partir de 1989 pour supporter les échanges de données du 'World Wide Web' :
conçu à l'origine pour transporter des textes HTML.

Il est défini par les RFC 1945 (1.0), 2616 (1.1)

Le protocole HTTP

Caractéristiques de base :

- **HTTP : un protocole client-serveur basé sur le mode message**

 - Tous les échanges sont basés sur un couple de **messages requête réponse**.

 - A chaque couple requête réponse est associé une **'méthode'**.

- **Approche asynchrone** : un client n'est pas bloqué par une requête (il peut émettre plusieurs requêtes en parallèle).

Le protocole HTTP

Caractéristiques de base :

- HTTP utilise le **transport fiable TCP** , mais d'autres réseaux sont possibles).
- une ressource est un **ensemble d'informations** identifié par une URL (Le plus souvent un **fichier**)

http	:	//	www.iut.fr	:	80	/	index.html
------	---	----	------------	---	----	---	------------

Le protocole HTTP

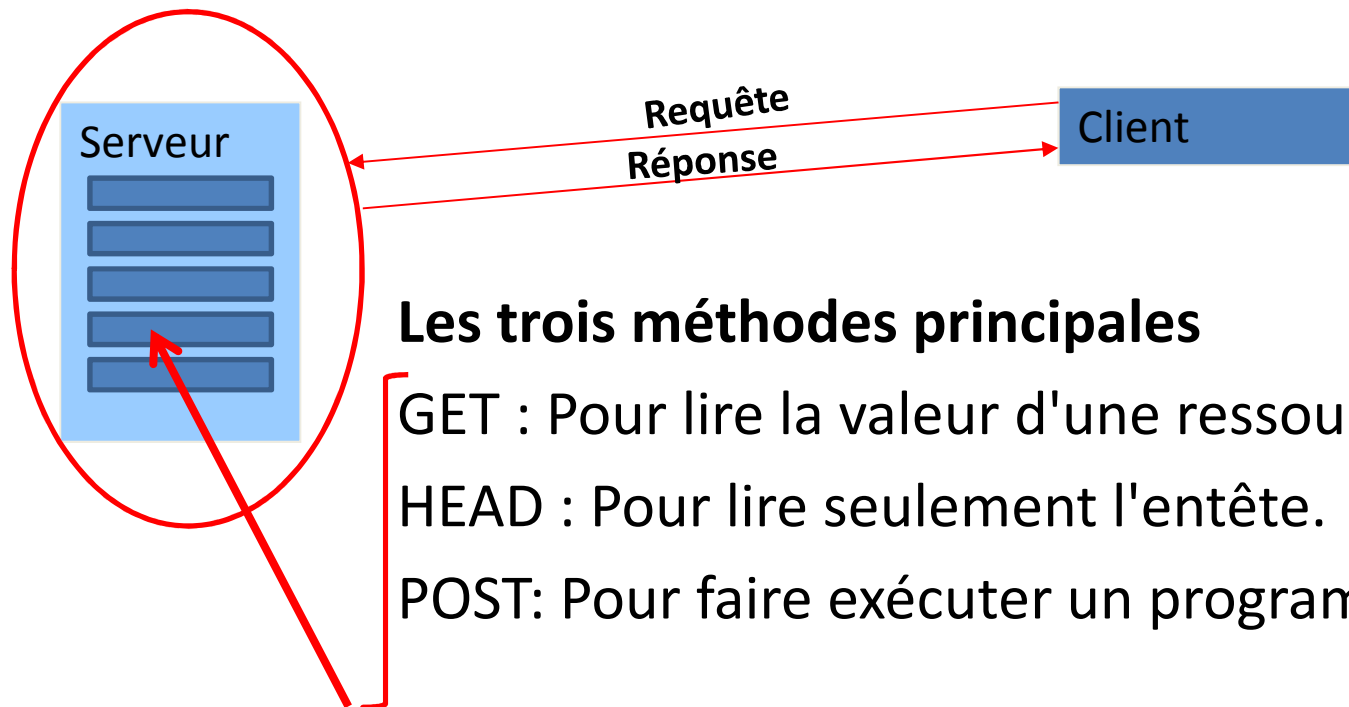
Logique des échanges HTTP

- 1 - Le client HTTP/1.0 ouvre une **connexion TCP** (le **numéro de port par défaut de http** est 80, un autre numéro est possible).
- 2 - Le client envoie **un message de requête**(exemple type : obtenir une ressource en fonction de différents paramètres).
- 3 - Le serveur répond par **un message de réponse** (la **ressource demandée**).
- 4 - Le client **ferme la connexion TCP**

Remarque : En HTTP chaque ressource nécessite sa propre connexion TCP.

Le protocole HTTP

La liste des méthodes HTTP



Les trois méthodes principales

GET : Pour lire la valeur d'une ressource.

HEAD : Pour lire seulement l'entête.

POST: Pour faire exécuter un programme distant.

Les méthodes annexes

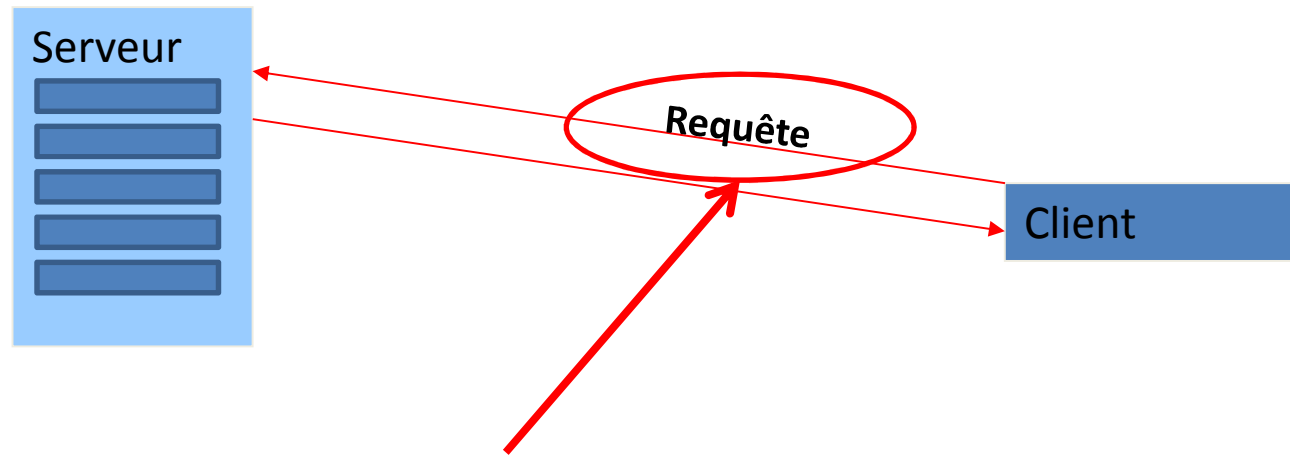
PUT : Pour envoyer une ressource sur un site distant.

DELETE : Pour détruire une ressource distante.

LINK : Pour établir un lien entre deux ressources

Le protocole HTTP

Le format de la ligne des requêtes



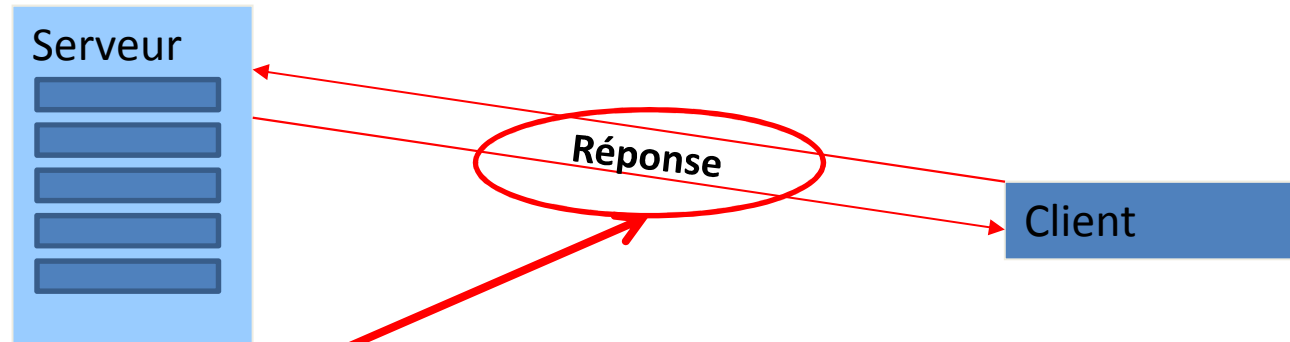
Trois parties séparées par des espaces

- Un **nom de méthode**,
- Le **chemin d'accès à la ressource**,
- La **version du protocole HTTP**.

Exemple : GET /chemin HTTP/1.0

Le protocole HTTP

Le format des réponses



La réponse se compose de trois parties :

- . Un **code retour**,
- . Un **entête**,
- . Un **texte (à balises)** représentant la page demandée.

HTTP/1.1 200 OK

Date: Wed, 25 Oct 2000 10:02:12 GMT

Server: Apache/1.3.6 (Unix) PHP/3.0.7

Last-Modified: Wed, 18 Oct 2000 11:07:00 GMT

ETag: "ac-221e-39ed8454"

Accept-Ranges: bytes

Content-Length: 8734

Connection: close

Content-Type: text/html

<html>

<head>

<link rel="stylesheet"

.....

Le protocole HTTP

Les codes réponses les plus courants

"200" ; OK

"201" ; Created

"202" ; Accepted

"204" ; No Content

"301" ; Moved Permanently

"302" ; Moved Temporarily

"304" ; Not Modified

"400" ; Bad Request

"401" ; Unauthorized

"403" ; Forbidden

"404" ; Not Found

"500" ; Internal Server Error

Le protocole HTTP

Exemple complet d'échange

Poste A
10.20.1.1

Poste B
66.249.85.99

TCP 33667 > http [SYN] Seq=0 Ack=0 Win=5840 Len=0 →
← TCP http > 33667 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0
TCP 33667 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 →
HTTP GET / HTTP/1.0 →
← TCP http > 33667 [ACK] Seq=1 Ack=426 Win=7765 Len=0
← **HTTP HTTP/1.0 200 OK (text/html)**
TCP 33667 > http [ACK] Seq=426 Ack=1431 Win=8580 Len=0 →
← **HTTP Continuation**
TCP 33668 > http [ACK] Seq=520 Ack=2062 Win=11440 Len=0 →
← **HTTP Continuation**
← **HTTP Continuation**

Le protocole HTTP

Il existe de nombreuses implémentations du protocole HTTP

- Apache (Apache Groupe)
- IIS (Microsoft)
- Nginx (Logiciel libre)
- LiteSpeed (LiteSpeed Technologies)

Remarques :

- En 2020 on référence plus de 1,78 milliards de sites webs (1 milliard en 2017 → 78% progression).
- Parts de marché : Apache 41,57 % , Nginx 26,38%, IIS 12,43%
(source hostadvice 2020)

L'accès à un service

L'accès à un service

L'accès aux services se fait généralement via des applications toutes prêtes (applications clientes).

L'accès au web :



L'accès au transfert de fichiers :



L'accès à la messagerie :

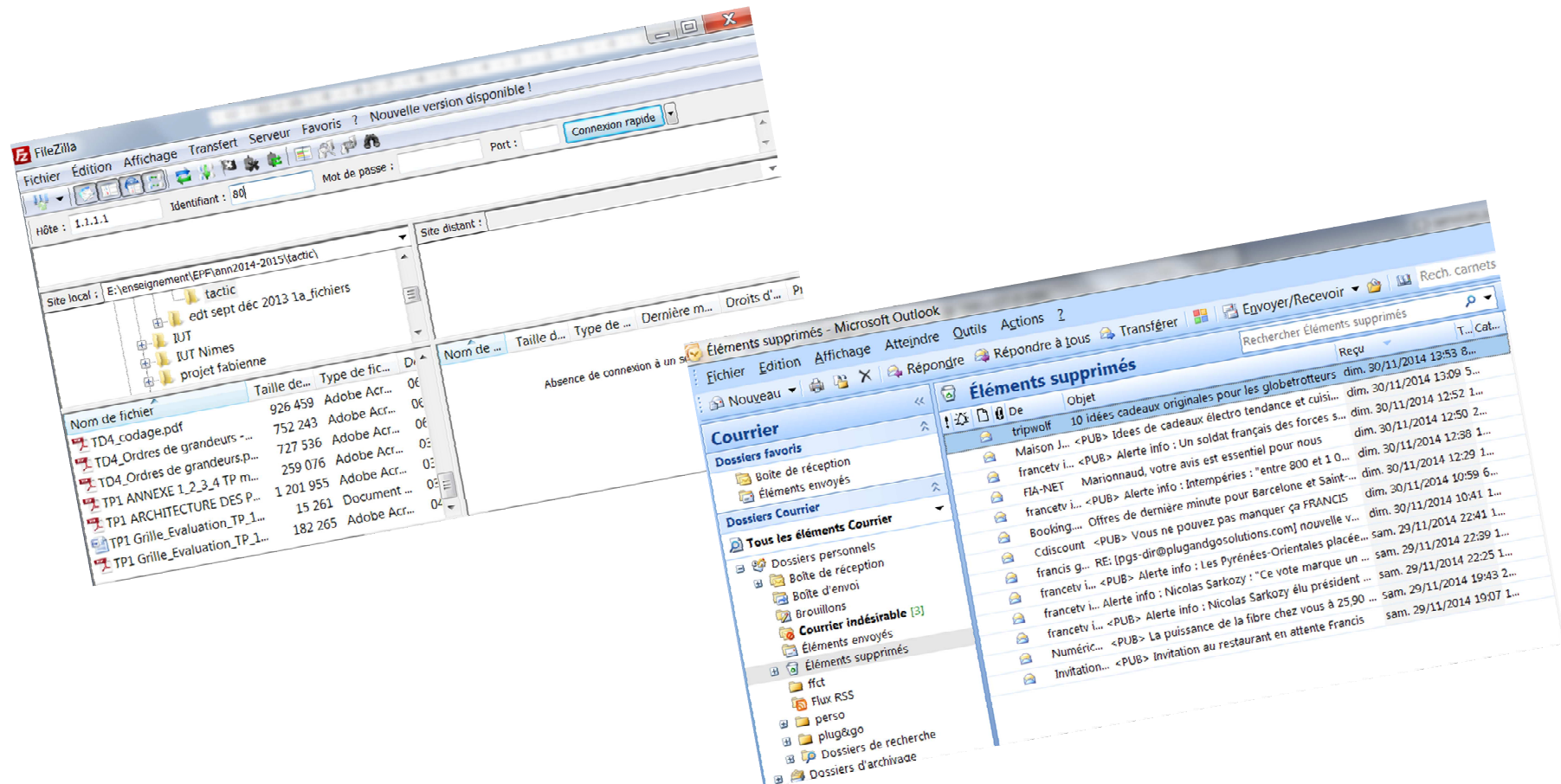


Etc..

L'accès à un service

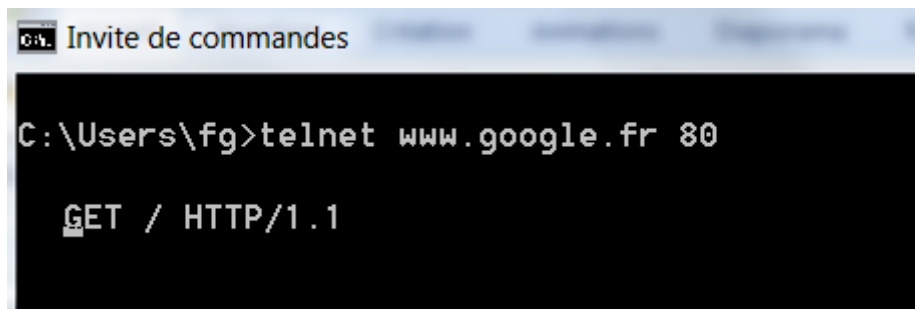
Ce type d'application propose des interfaces pour faciliter l'utilisation.

L'utilisateur ne voit pas les requêtes des protocoles utilisés.



L'accès à un service

Il est aussi possible d'utiliser des outils plus basiques comme Telnet, par lesquels l'utilisateur peut dialoguer avec le service, en lui envoyant directement des requêtes.



```
GA Invite de commandes
C:\Users\fg>telnet www.google.fr 80
GET / HTTP/1.1
```

```
HTTP/1.0 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Location: http://www.google.fr/?gfe_rd=cr&ei=7h17UPPx0IWBULr8gfgN
Content-Length: 256
Date: Sun, 30 Nov 2014 13:21:50 GMT
Server: GFE/2.0
Alternate-Protocol: 80:quic,p=0.02

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=u
ITLE>302 Moved</TITLE></HEAD><BODY>
                                <H1>302 Moved</H1>
                                The document has mo
EF="http://www.google.fr/?gfe_rd=cr&ei=7h17UPPx0IWBULr8gfgN">here</A
</BODY></HTML>
```

L'accès à un service

On peut enfin dialoguer directement avec les services en utilisant des applications en C utilisant l'interface Socket.

```
// envoi d'une requête à www.google.fr  
strcpy(mess,"GET / HTTP/1.1 \n");  
write (sock, mess, strlen(mess));
```

Réponse du serveur

```
HTTP/1.0 302 Found  
Cache-Control: private  
Content-Type: text/html; charset=UTF-8  
Location: http://www.google.fr/?gfe_rd=cr&ei=yEx4VIG4MsqA8Qep4oDADQ  
Content-Length: 258  
Date: Fri, 28 Nov 2014 10:22:00 GMT  
Server: GFE/2.0  
Alternate-Protocol: 80:quic,p=0.02
```

```
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8"><TITLE>302  
Moved</TITLE></HEAD><BODY><H1>302 Moved</H1>The document has moved<A  
HREF="http://www.google.fr/?gfe_rd=cr&ei=yEx4VIG4MsqA8Qep4oDADQ">here</A>.  
</BODY></HTML>
```

Conclusion

Le succès de l'internet est essentiellement du :

- Au grand nombre de services proposés,
- A l'amélioration des interfaces, facilitant l'accès aux services.

Tous ces services, permettent à presque 60% de la population mondiale (soit 4,6 milliards de personnes – chiffres 2020) de profiter de l'internet.

MAIS ...

L'accès aux services, en bas niveau, reste toujours possible.

Avec l'augmentation du nombre d'internautes et la multiplication des formations réseaux , il y a de plus en plus de personnes qui exploitent des failles connues dans les services.

Ceci est de nature à mettre en danger un grand nombre de machines :
on parle de sécurité des systèmes