

R3.06

Architecture des réseaux

Les services

Introduction

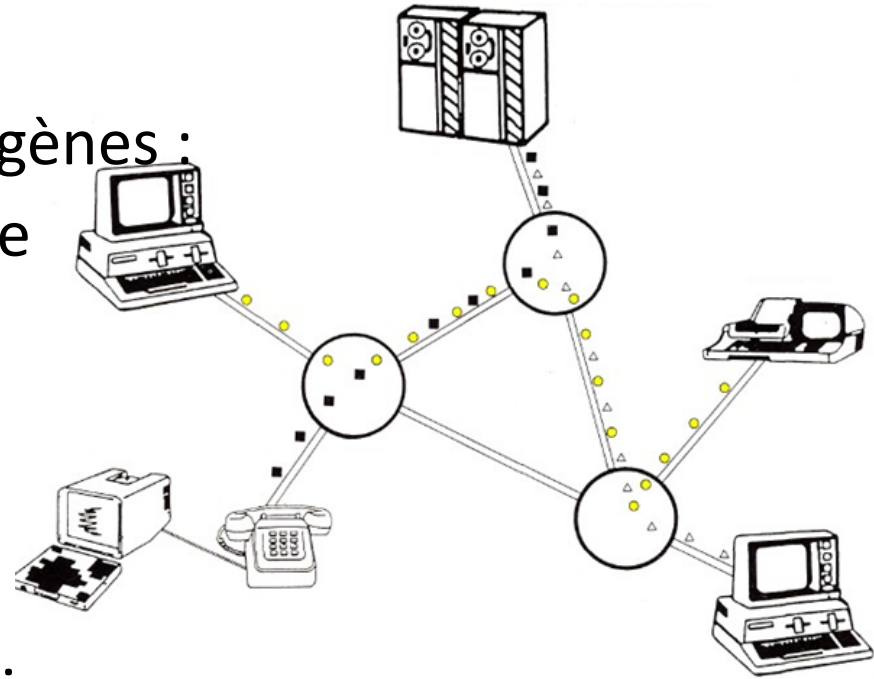
Le monde de l'internet c'est :

Des millions de liaisons hétérogènes :

- fibre, cuivre, radio, satellite
- débit, bande passante

Des millions de routeurs.

Mais il faut donner envie d'utiliser ce réseau pour cela il faut déployer des applications informatiques .



Introduction

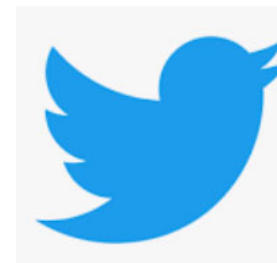
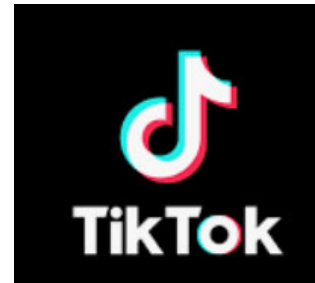
L'internet c'est aussi :

Des milliers d'applications
client-serveurs,

Des dizaines de milliards de
mails et de documents
échangés,

Des milliards de transactions
bancaires, ...

→ c'est un monde de services



Introduction

Avec tous ces services, internet devient :

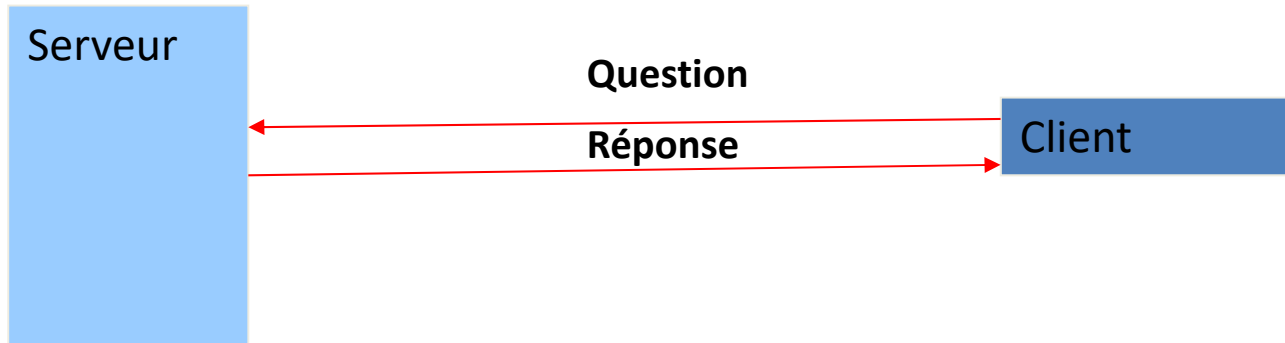
- un « accélérateur » de développement pour les PME,
- un marché de l'emploi en pleine expansion.



Grâce à des technologies du «Web » qui évoluent sans cesse pour s'adapter à une demande toujours plus forte.

Introduction

Sur Internet, nous avons des applications qui s'exécutent sur des machines différentes et communiquent entre-elles.



Pour que les applications puissent fonctionner correctement elles doivent « dialoguer » en parfaite synchronisation.

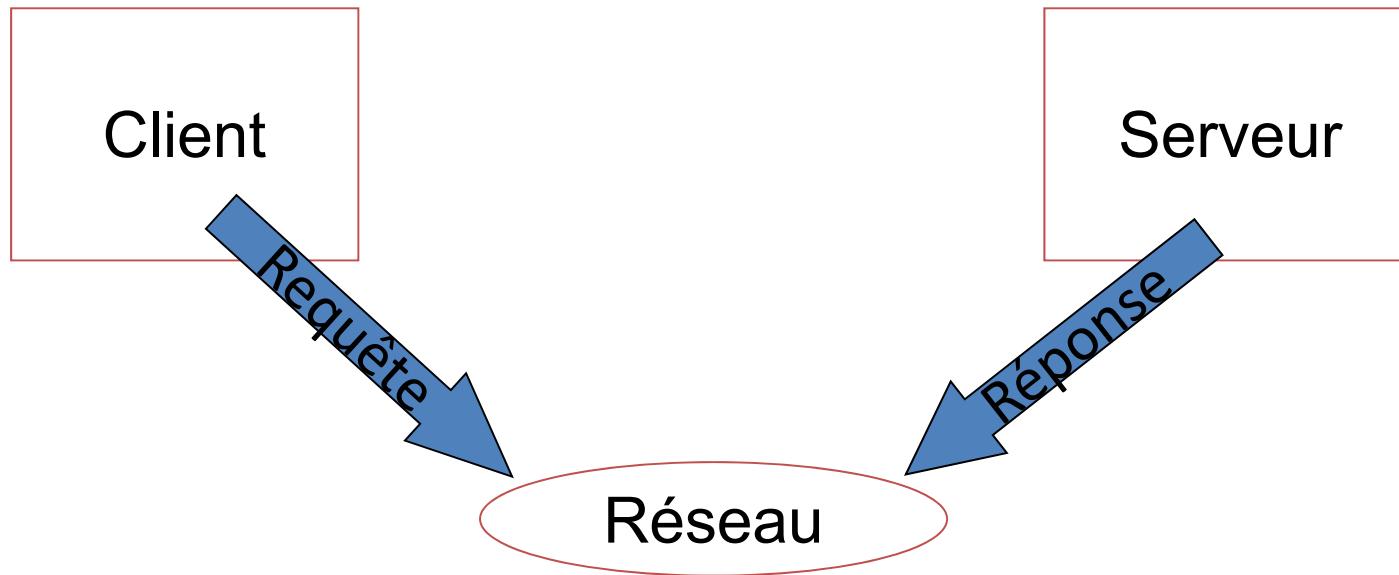
Pour cela il faudra normaliser :

- La logique des applications : l'une attendra les requêtes et l'autre les lui transmettra puis inversement
- Le format des données échangées

On parle d'applications Client/Serveur

Introduction

Approche client-serveur

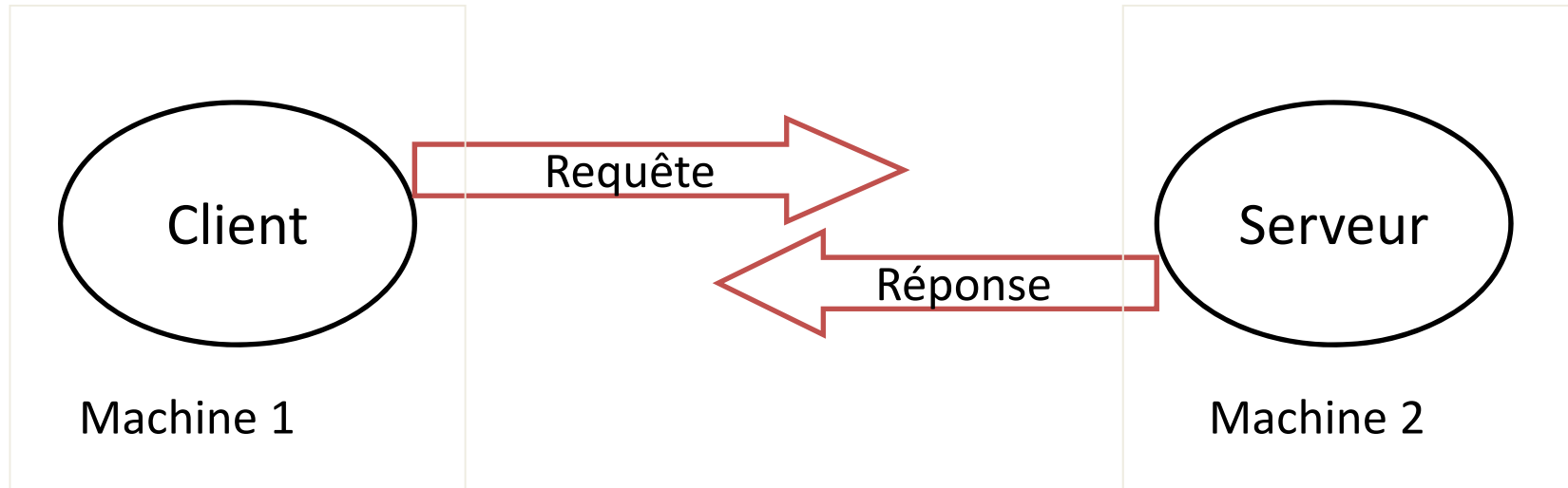


Client et serveur dialoguent par l'intermédiaire d'un réseau existant. Il est nécessaire d'interfacer les applications avec le réseau, mais en leur masquant les aspects techniques.

→ Mise en œuvre d'une API de programmation : socket (déjà vu)

Introduction

Approche client-serveur

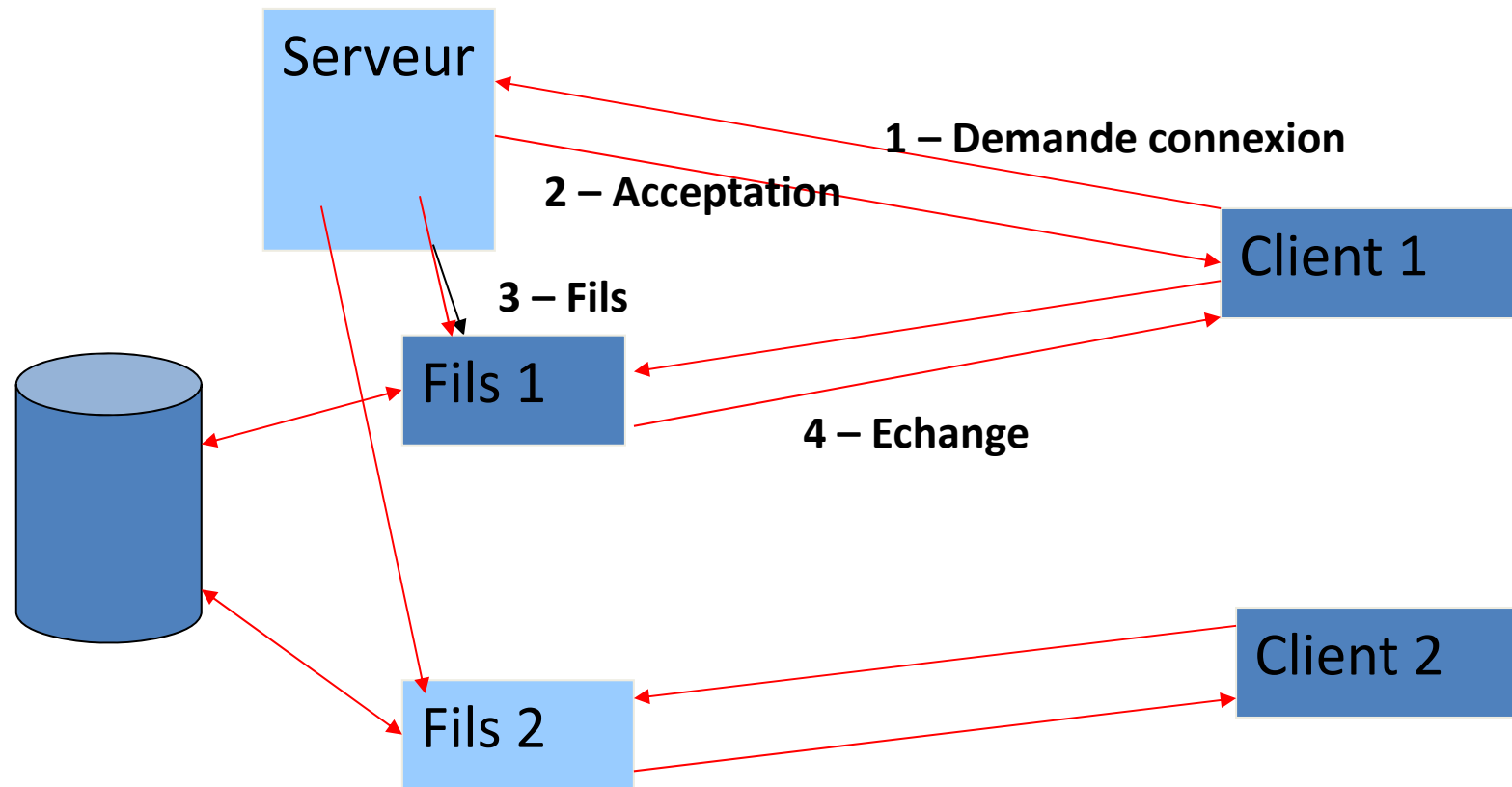


- . Le client envoie dans un **premier message** : « **une requête** » d'exécution d'un traitement à un serveur.
- . Le serveur effectue le travail et fournit dans **un second message** : « **la réponse** ».

→ Il est nécessaire de normaliser le format des données

Introduction

Approche client-serveur



La vocation d'un service est de pouvoir satisfaire plusieurs clients à la fois tout en partageant des ressources communes.
Pour gérer plusieurs clients, le serveur doit générer et gérer plusieurs fils d'exécution.

→ **Il est nécessaire de synchroniser les dialogues**

Introduction

Grâce a toutes ces recommandations il existe aujourd'hui une multitude de services normalisés :

- 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

Ce que nous allons voir par la suite :

1 – Comment on accède à un service : URL

2 - Comment on normalise un service : RFC

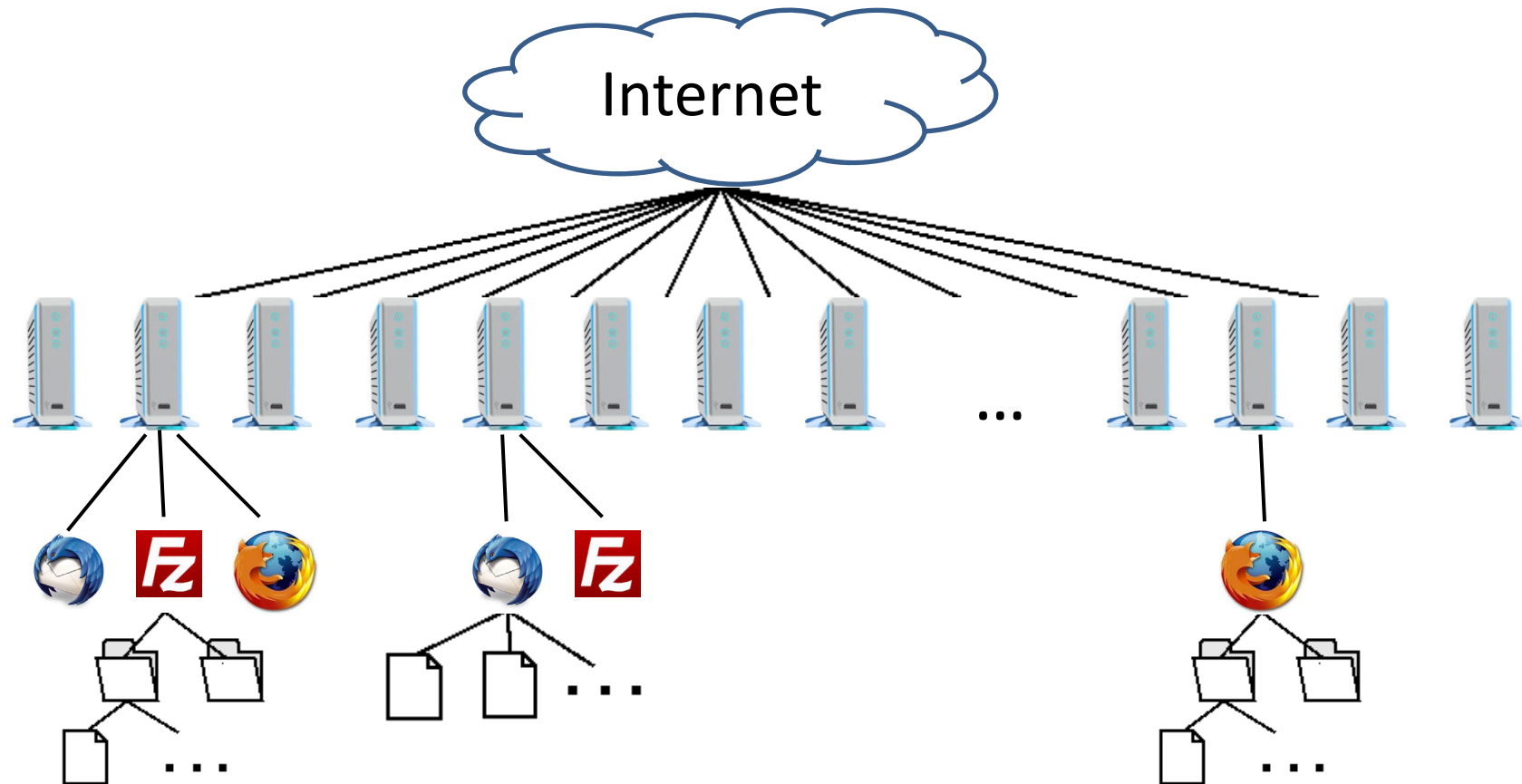
3 – Des exemples de services normalisés : HTTP – DNS

4 – Le traitement du format des données : XML ...

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

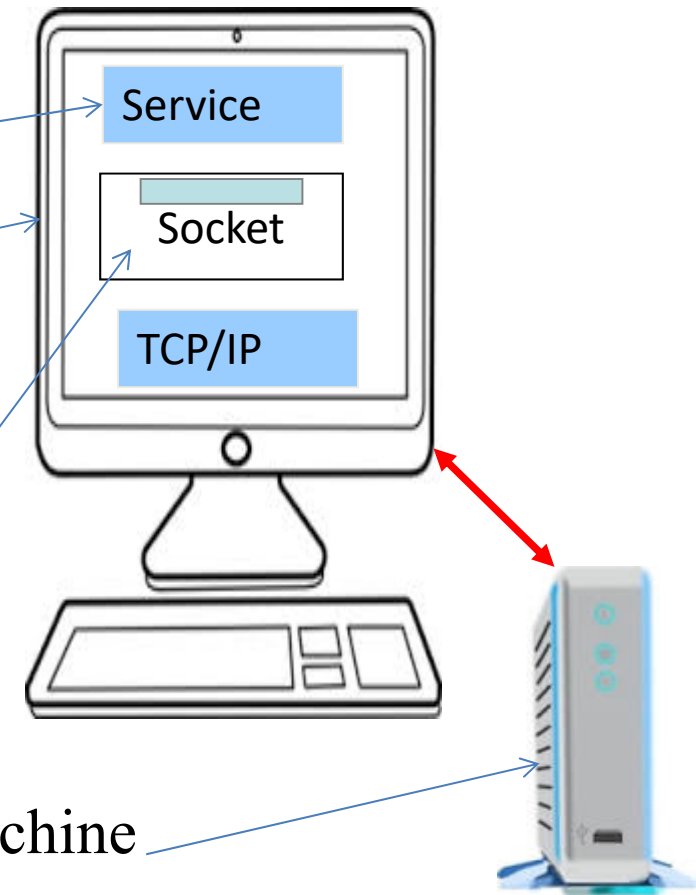
→ @IP, nom.machine,

3 - Le moyen d'accéder à la ressource

→ Numéro de ports (80, 53, 20, ...)

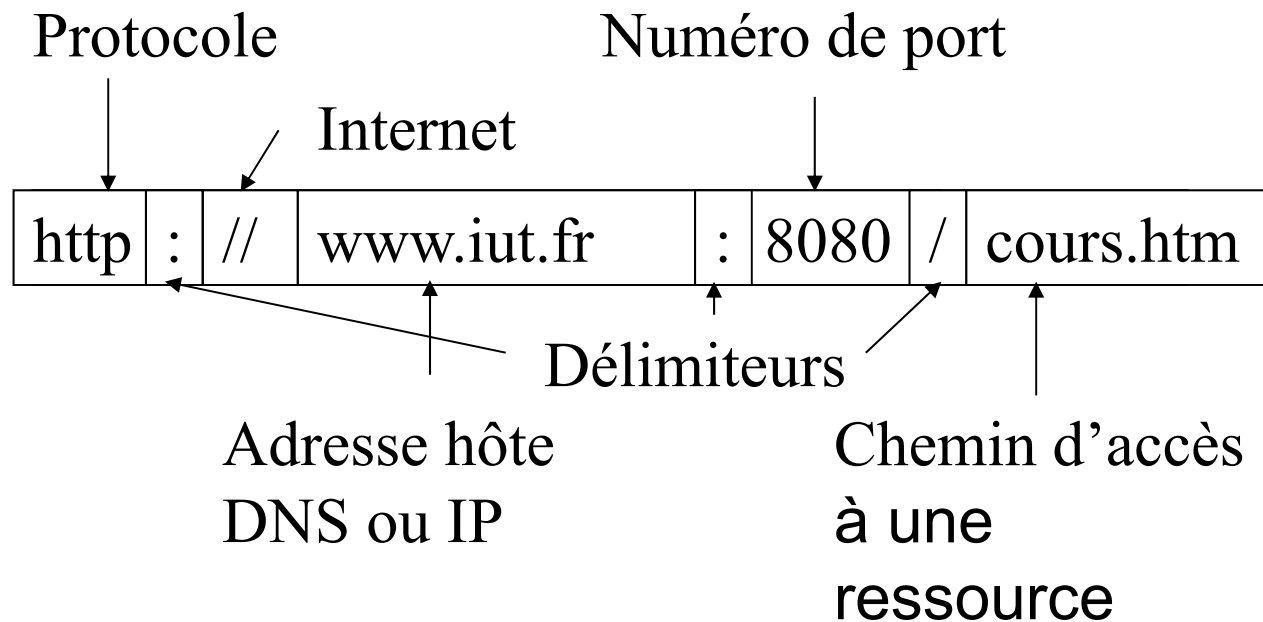
4 - La localisation de la ressource dans la machine

→ /home/name/fic, ...



Désignation

Le schéma d'adressage doit reprendre tous ces éléments.



URL : Localisateur Ressource Unifié

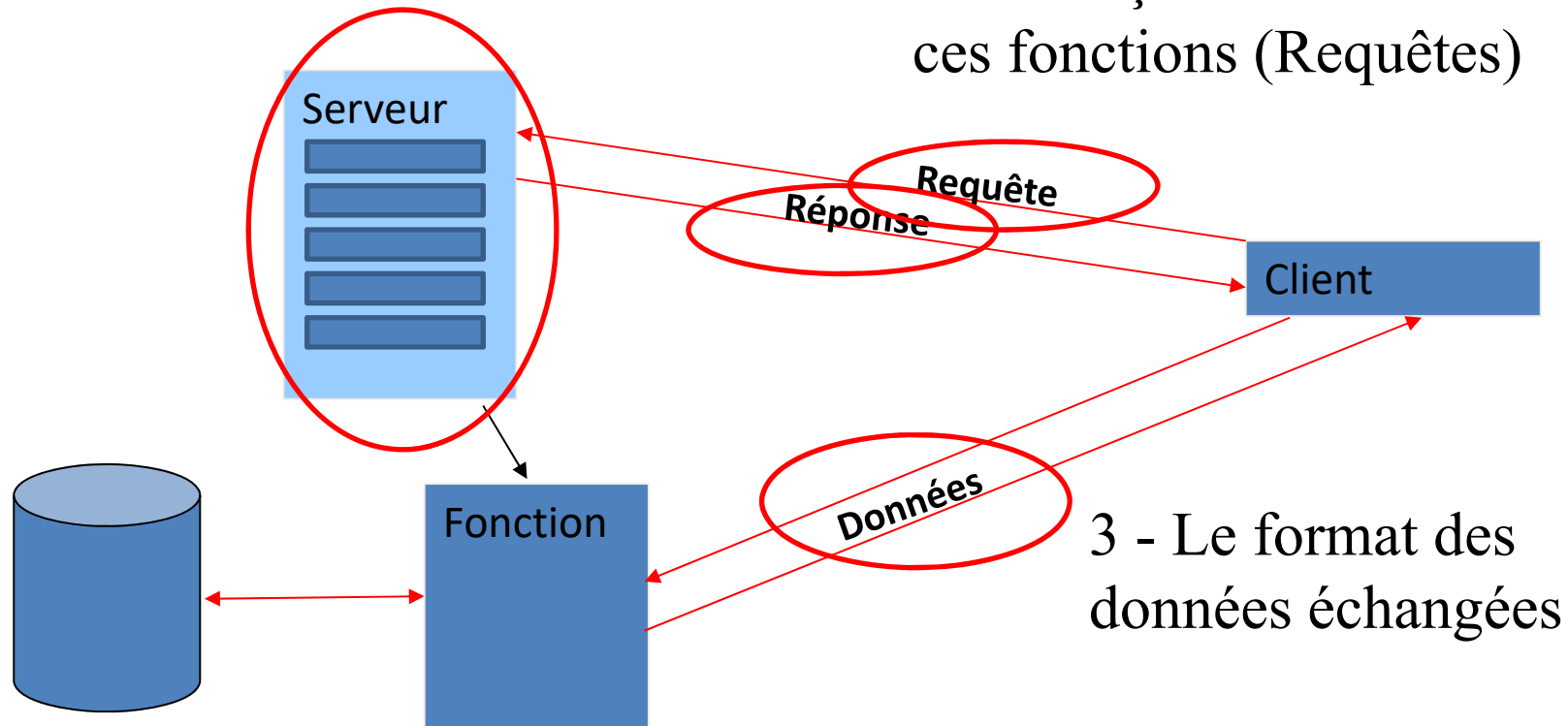
La normalisation des services

Normalisation des services

Normaliser un service, consiste à définir :

1 - Les fonctions du serveur,
accessibles aux clients

2 - La façon d'accéder à
ces fonctions (Requêtes)



3 - Le format des
données échangées

Normalisation des services

Tout comme le matériel, l'ISO recommande, dans couches hautes de l'OSI, des services normalisés.

	Applications TCP/IP directes				Applications pile SUN/OS
7. Application		EXEMPLES			NFS: "Network File System"
6. Présentation	DNS: Domain Name System	SMTP: Simple Mail Transfer Protocol	HTTP: Hyper Text Transfer Protocol	FTP: File 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				

Les descriptions de ces services, sont contenues dans des RFC (Requests For Comment)

Structure d'un service et 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.

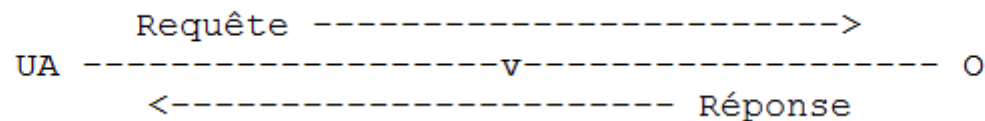
Structure d'un service et RFC

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 requêtes

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

4 – La description des requêtes

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 – 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

6 – 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

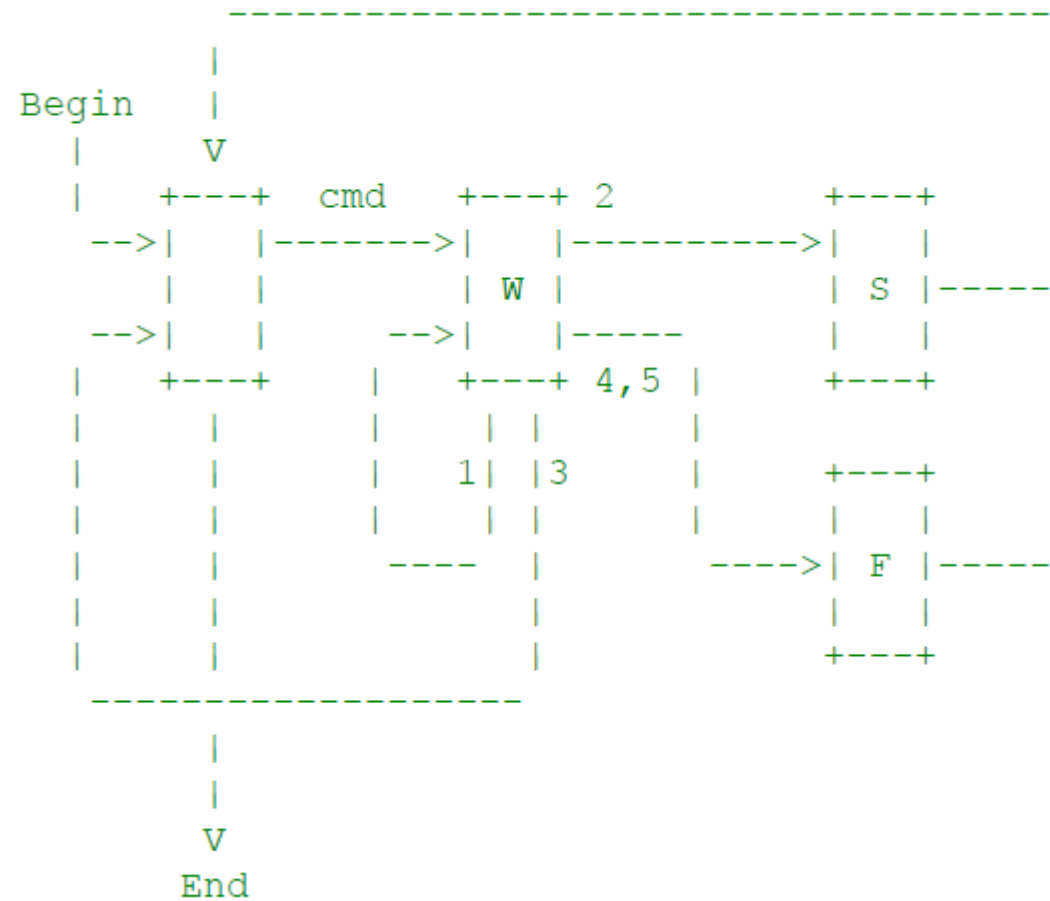
7 – Le format des données à échanger

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

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

Structure d'un service et RFC

8 – Les échanges types – Diagrammes d'états



Quelques RFC incontournables

Il existe plus de 8500 RFCs

RFC 791 → IP V4

RFC 2460 → IP V6

RFC 2616 → HTTP 1.1

RFC 1866 → HTML 2.0

RFC 1034 /1035 → DNS

RFC 2131 → DHCP

RFC 5321 → SMTP (messagerie)

RFC 1736 → URL

RFC 1 → méthode d'envoi des paquets d'un ordinateur à l'autre dans le cadre du tout premier réseau de l'époque