

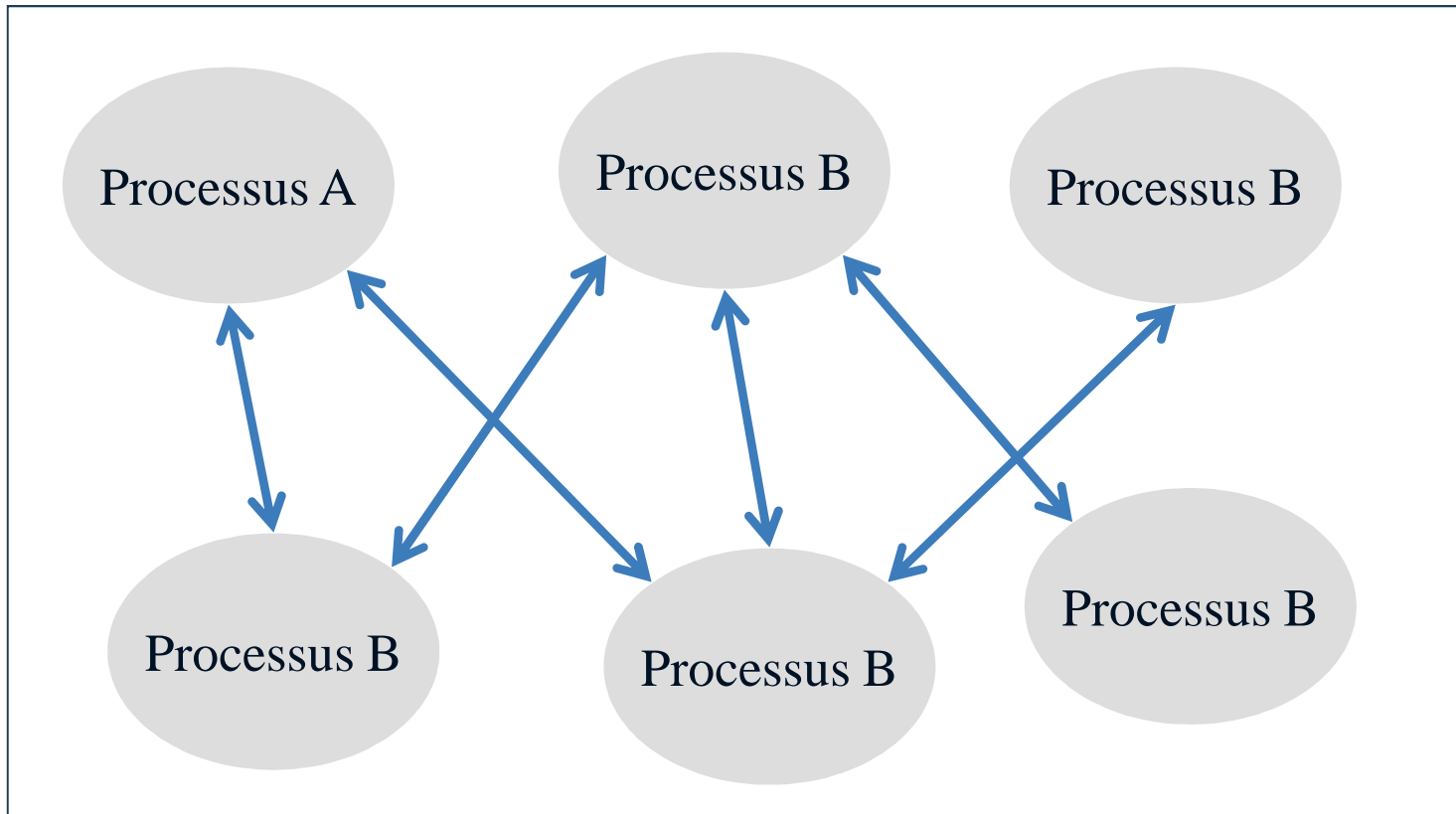
# **Architecture 2**

## **Architecture des Réseaux**

**Le point sur le cours d'archi**

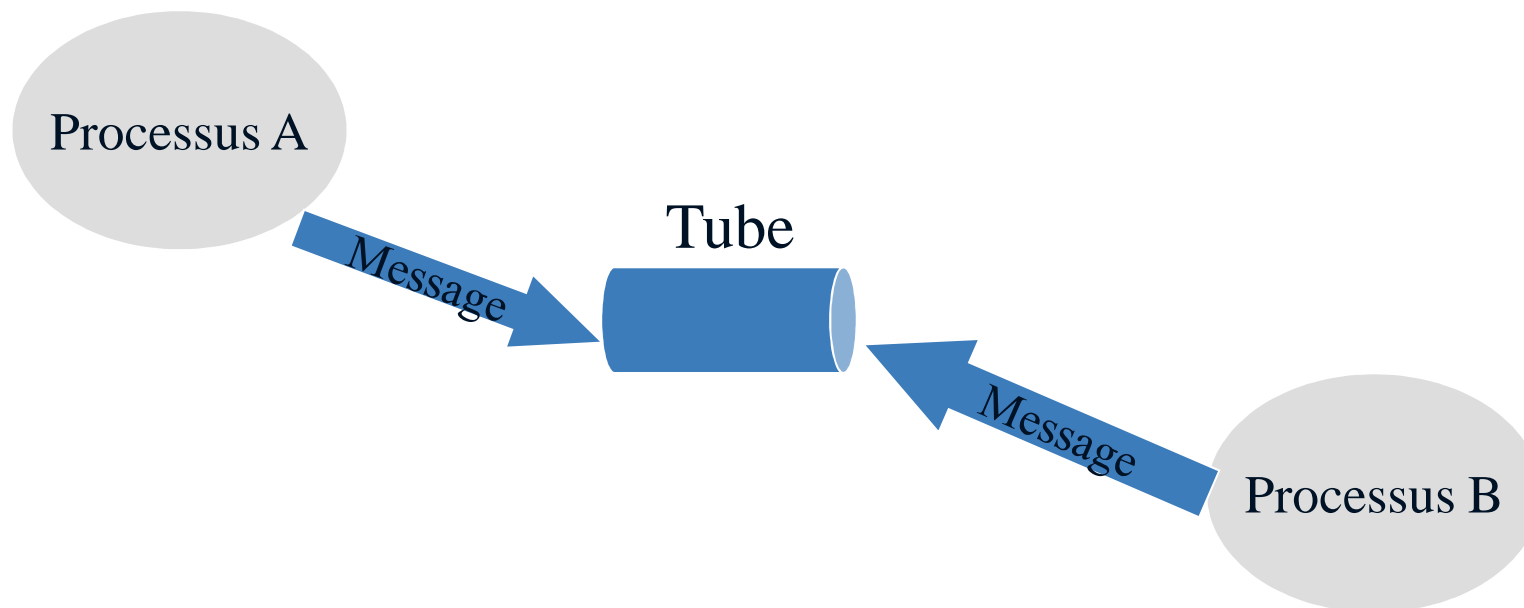
# Le point

Machine



Dans les systèmes actuels (multitâches), les processus ont vocation à communiquer entre-eux

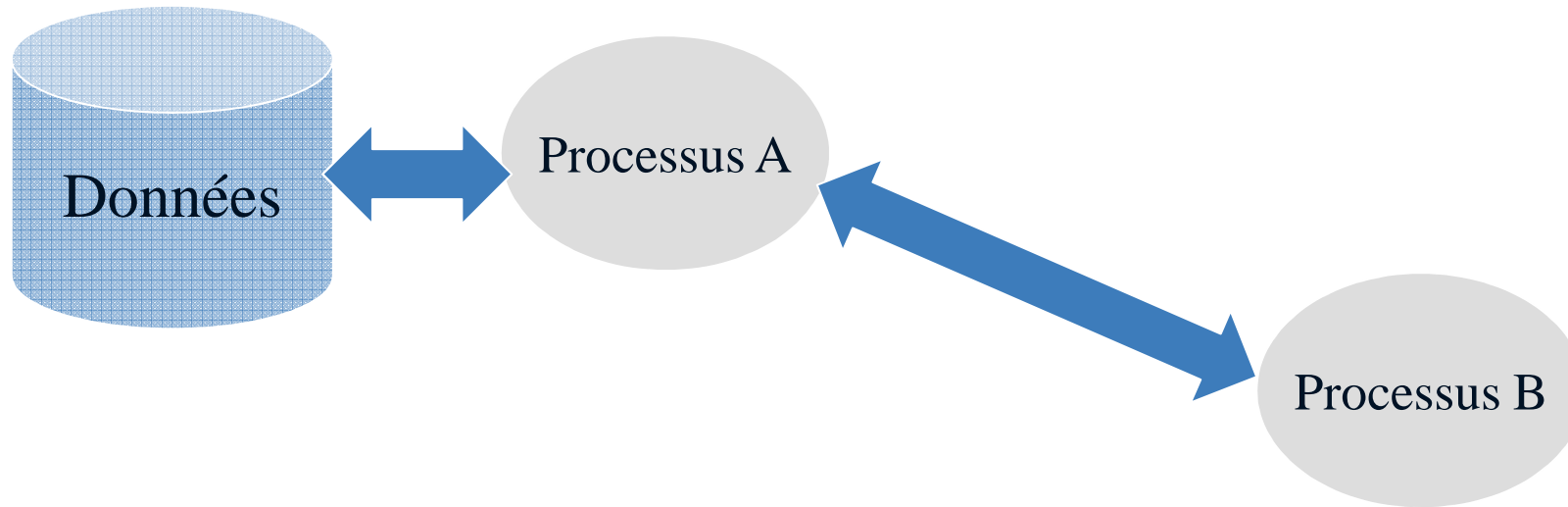
# Le point



Pour que deux processus communiquent, il existe des solutions:  
**LES TUBES.**

Ces TUBES se comportent comme des boîtes aux lettres dans lesquelles les processus déposent et récupèrent les messages.

# Le point

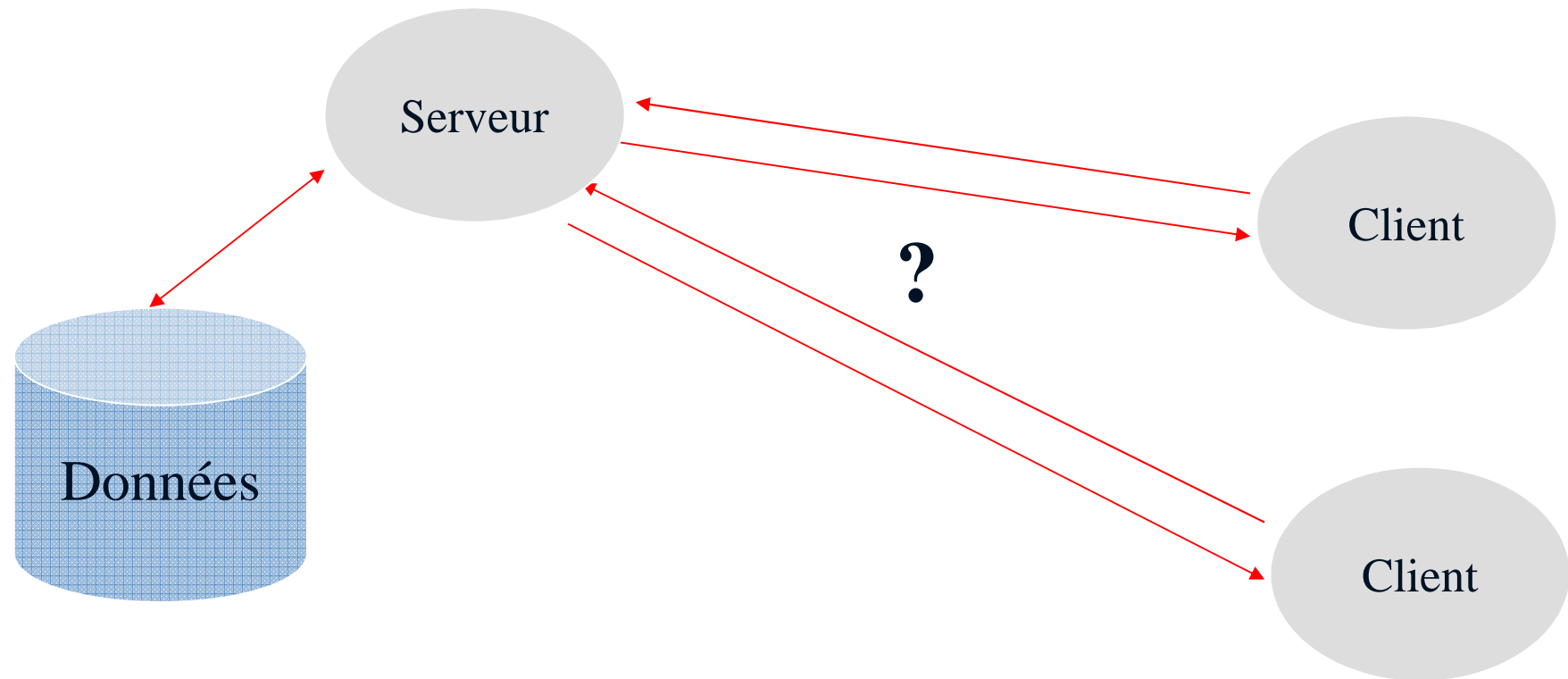


Si les processus communiquent entre-eux c'est qu'ils ont des données à partager ou à récupérer dans des **FICHIERS**.

Ainsi, un processus effectuera des requêtes et l'autre lui transmettra les réponses, puis inversement.

On parle d'applications ou d'architecture : **Client-Serveur**

# Le point



Le serveur doit être capable de communiquer avec plusieurs clients en même temps. **Mais comment ce serveur, peut-il communiquer avec plusieurs clients à la fois ???**

# Le point

Sachant que la logique d'un client-serveur est la suivante :

Mettre en service le serveur

Tant que (non fin)

Attendre (connexion client)

**// traitement requêtes client**

**Tant que (Requetes client)**

**lire(requete client)**

**traiter (requete client)**

**envoyer (réponse client)**

Arrêter le serveur

SERVEUR

Lancer le client

Se connecter au serveur;

**// requêtes client**

**Tant que (Requetes client)**

**envoyer (requete client)**

**attendre (réponse serveur)**

**traiter (réponse serveur)**

Arrêter le client

CLIENT

# Le point

On constate que , seulement une partie du code du serveur traite les demandes des clients

Mettre en service le serveur

Tant que (non fin)  
Attendre (client)

**// traitement requêtes client**  
**Tant que (Requetes client)**  
    **lire(requete client)**  
    **traiter (requete client)**  
    **envoyer (réponse client)**

Arrêter le serveur

SERVEUR

Lancer le client

Se connecter au serveur;

**// requêtes client**  
**Tant que (Requetes client)**  
    **envoyer (requete client)**  
    **attendre (réponse serveur)**  
    **traiter (réponse serveur)**

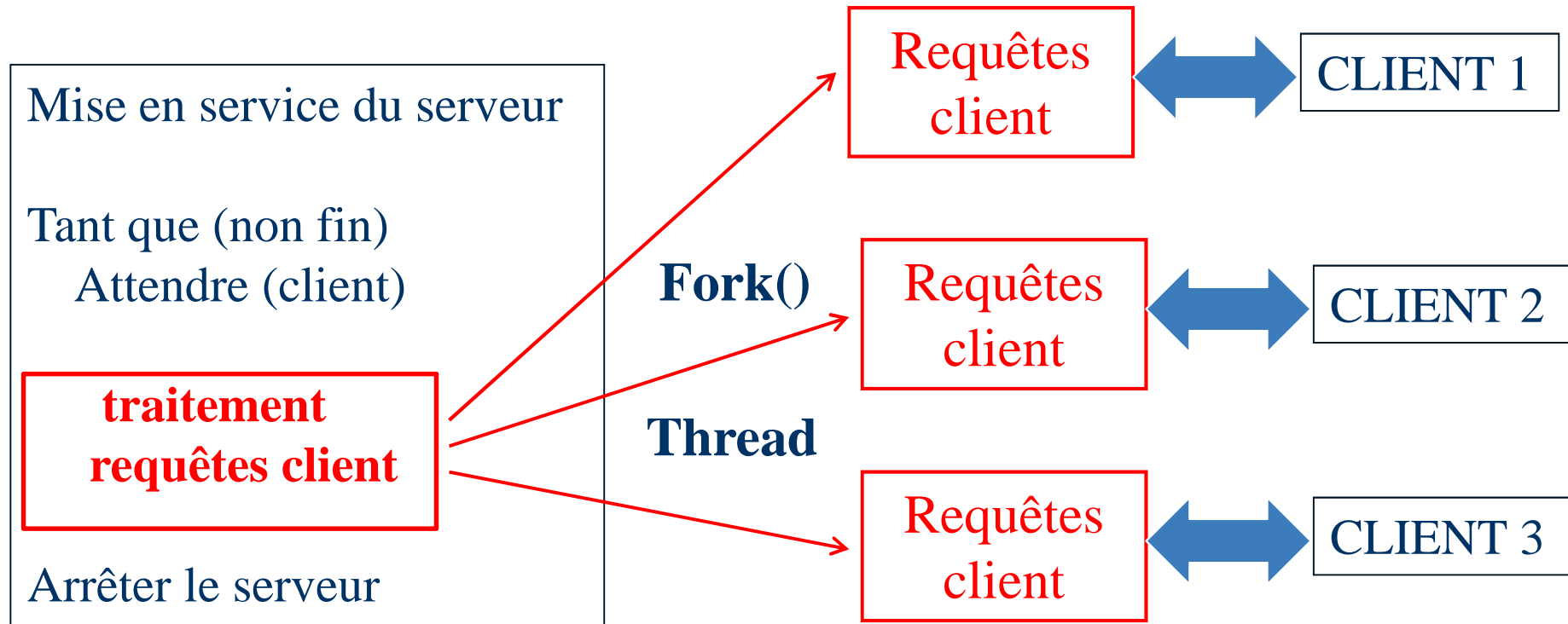
Arrêter le client

CLIENT



# Le point

Ainsi, on va dupliquer cette partie de code autant de fois que de clients

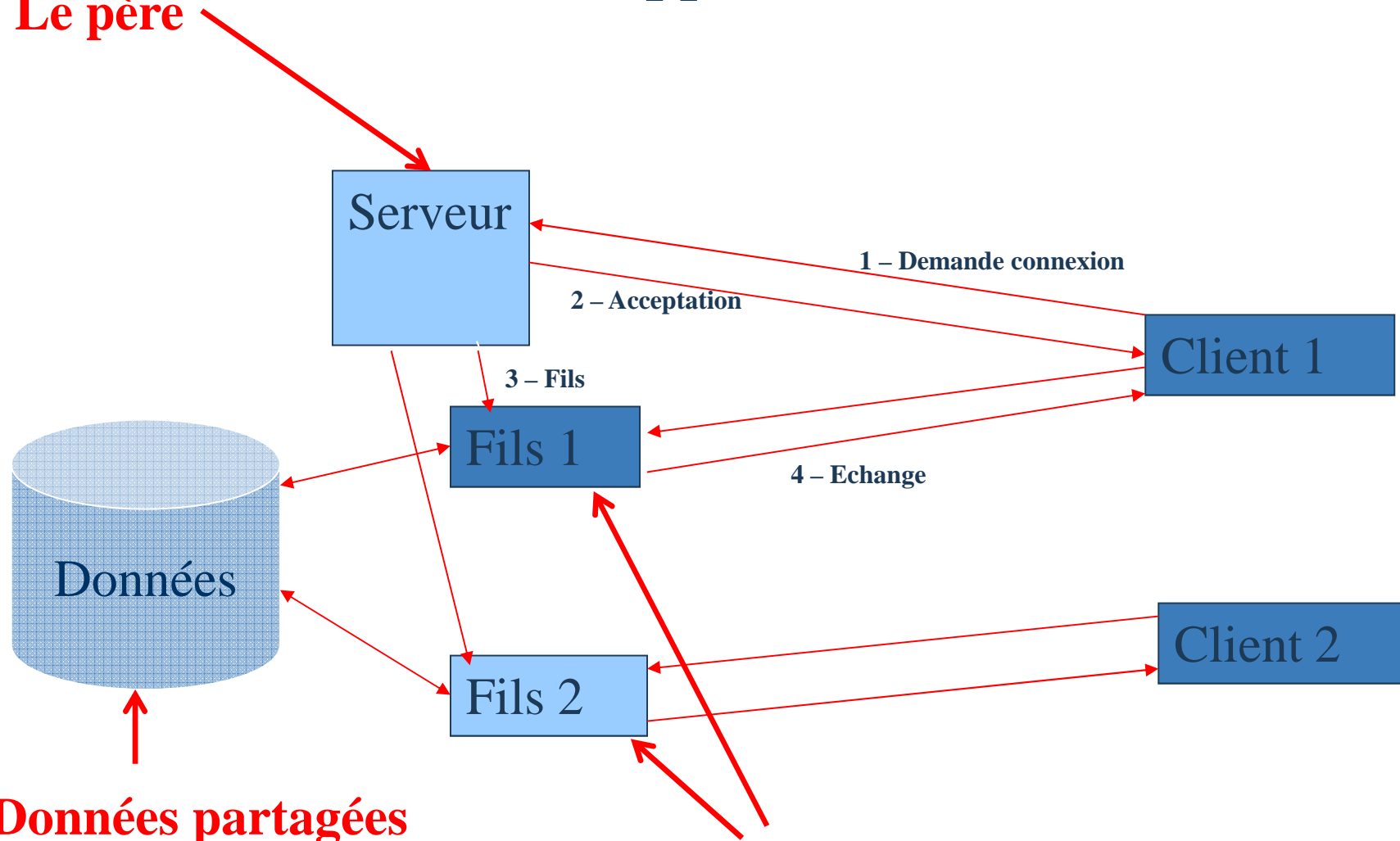


SERVEUR

# Le point

## Structure application (Client/Serveur)

Le père



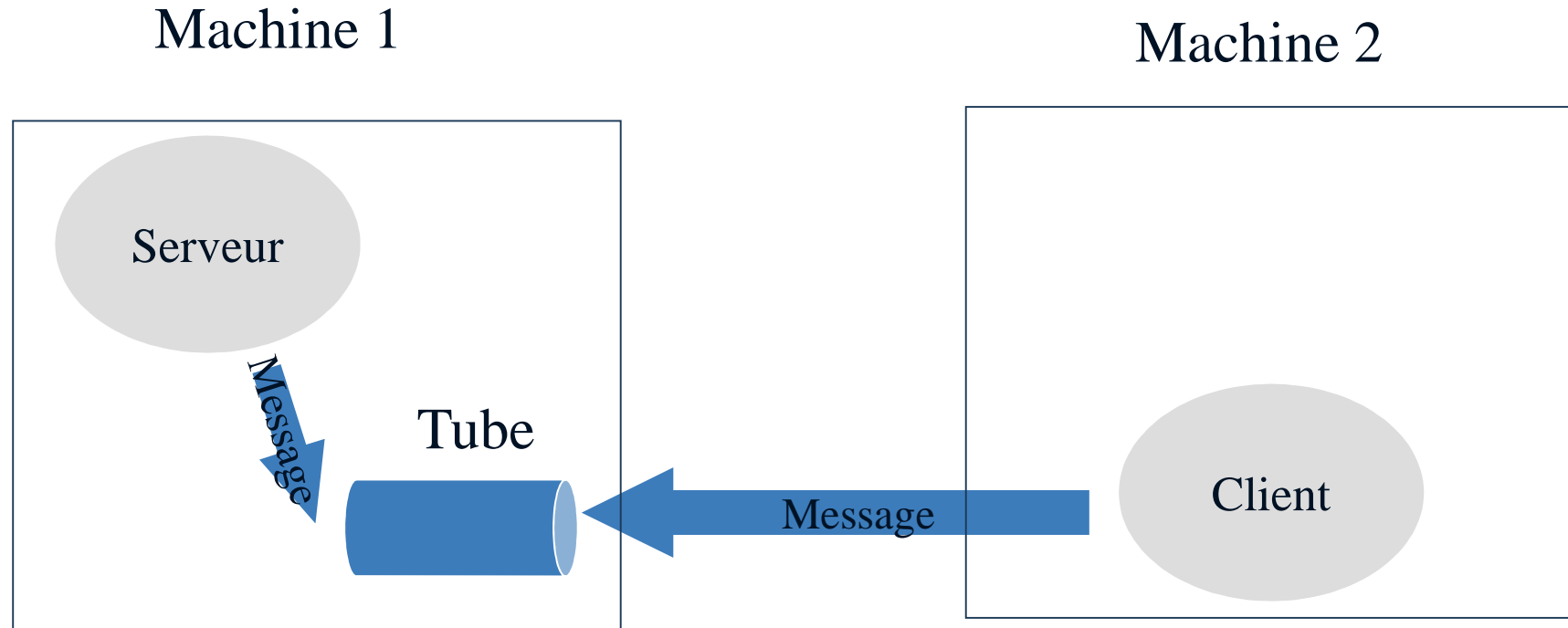
Données partagées  
Synchro via Sémaphores

Les Threads ou fork()

# **Les réseaux informatiques**

## **Introduction générale**

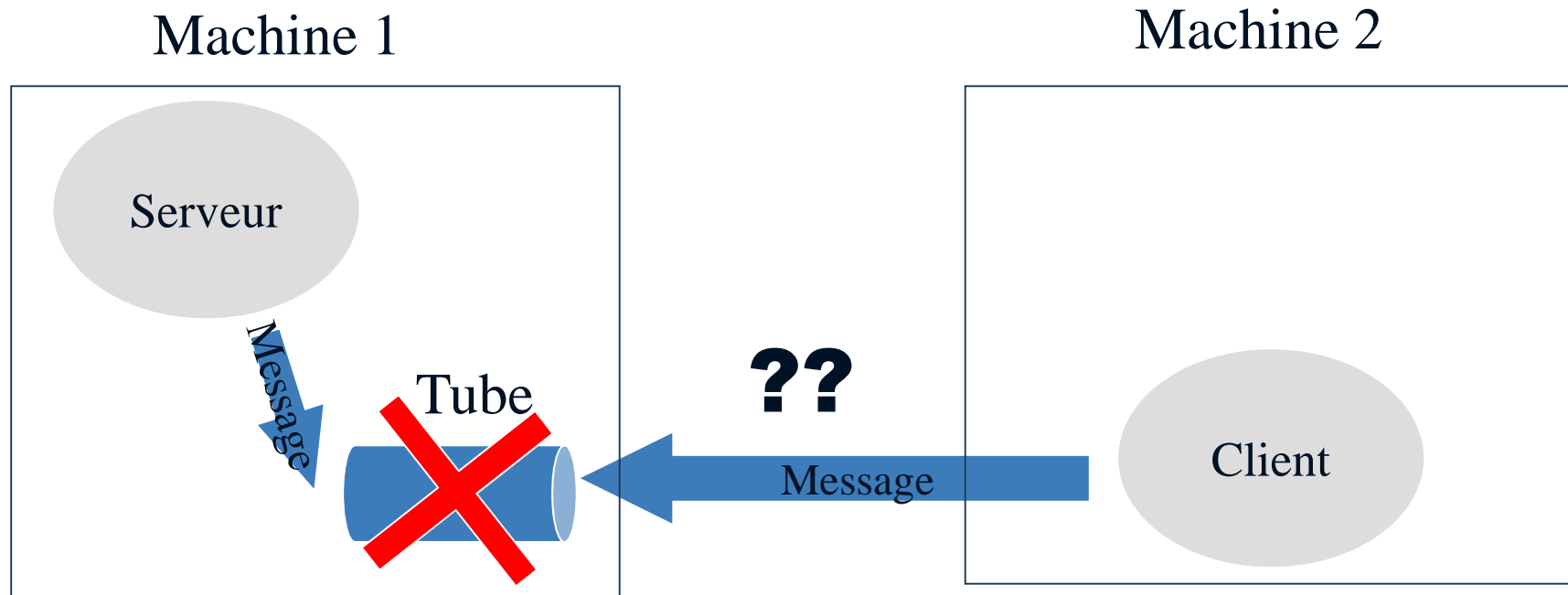
# Introduction générale



Imaginons maintenant que serveur et clients, soient situés sur des machines différentes.

**La communication avec les tubes peut-elle fonctionner ??**

# Introduction générale

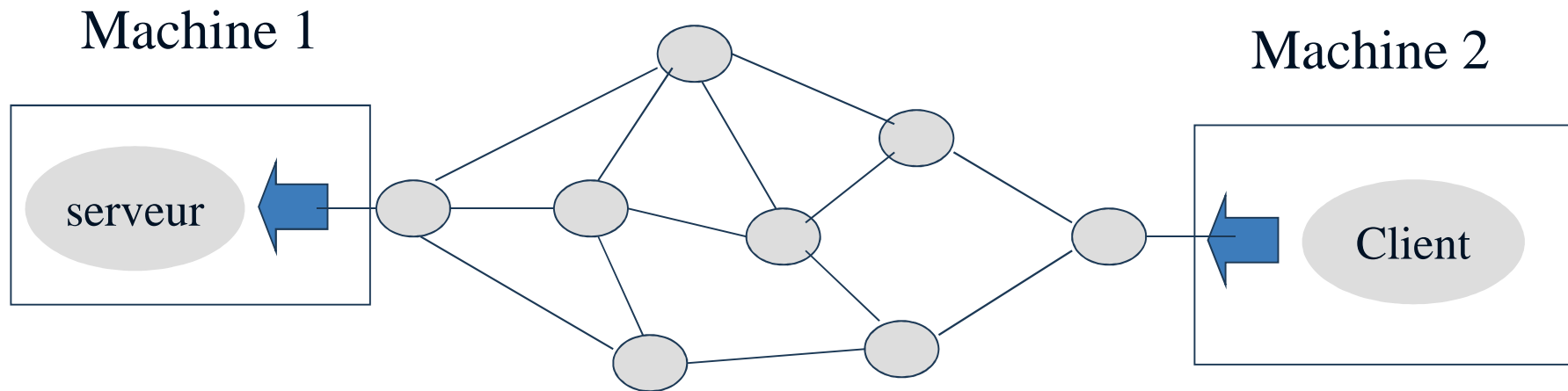


La réponse est **NON**? Il y a au moins deux raisons à cela :

- Les tubes ne sont pas accessibles entre deux systèmes
- On n'a pas précisé comment les machines sont reliées

# Introduction générale

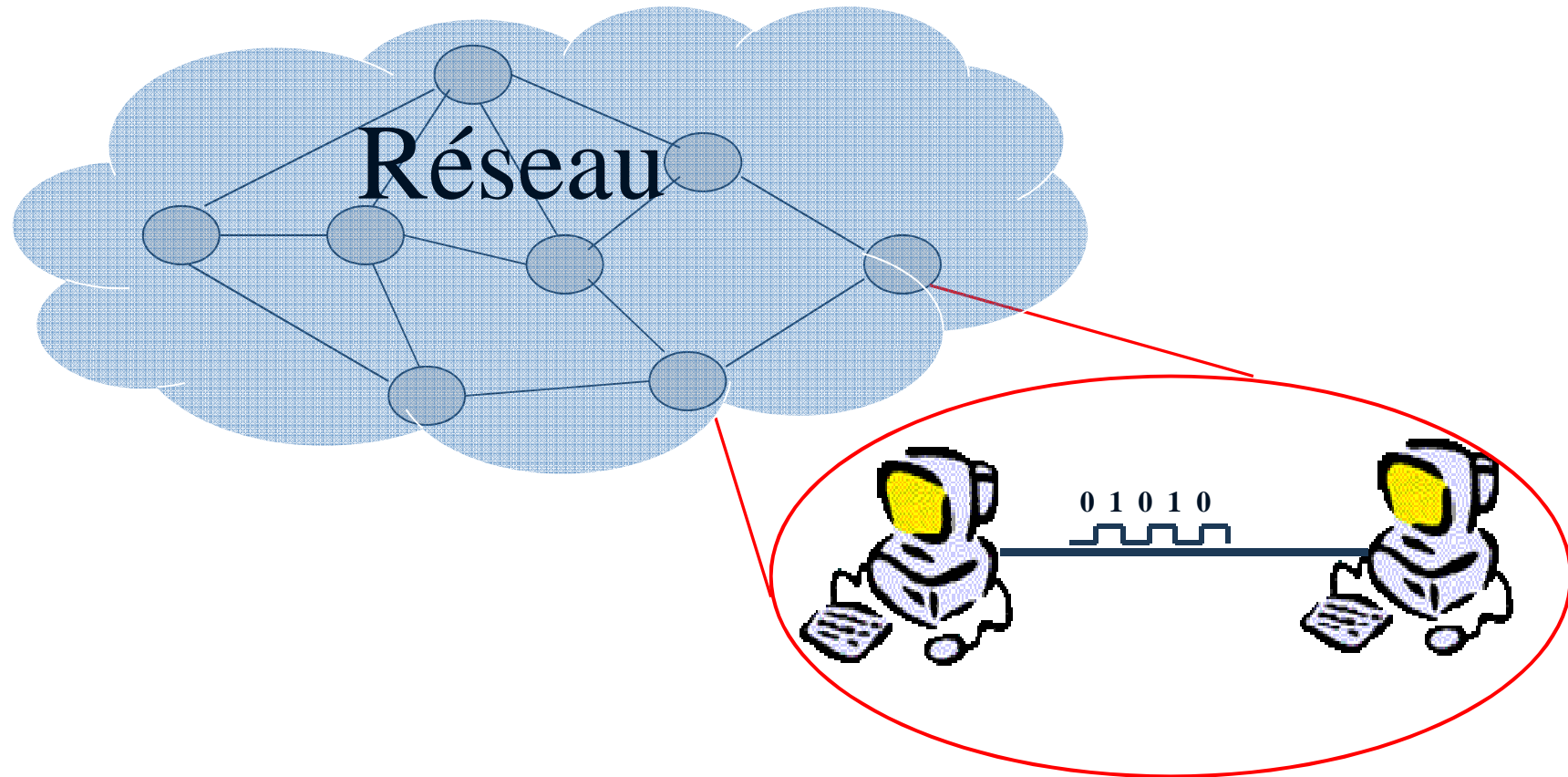
Les solutions : 1 – Créer un ensemble de voies par lesquelles nos applications , peuvent communiquer



Ces voies, sont un ensemble de routes, créées au niveau local, régional, national , international , que toutes les machines peuvent utiliser. Ces voies sont similaires à notre réseau routier, d'où l'appellation : **RESEAU INFORMATIQUE**

# Introduction générale

Les solutions : 2 – Connecter les machines au réseau



Il faut définir le matériel et les procédures, nécessaires à un échange de données binaires entre deux machines.

# Introduction générale

Les solutions : 3 – Créer une interface de communication entre les applications et le réseau



Nos applications client-serveur sont déjà assez complexes, on ne va pas leur demander, en plus, de se charger du transfert des données dans le réseau, ni de les contrôler. On va confier ce travail à des processus spécialisés dans le **TRANSPORT**.

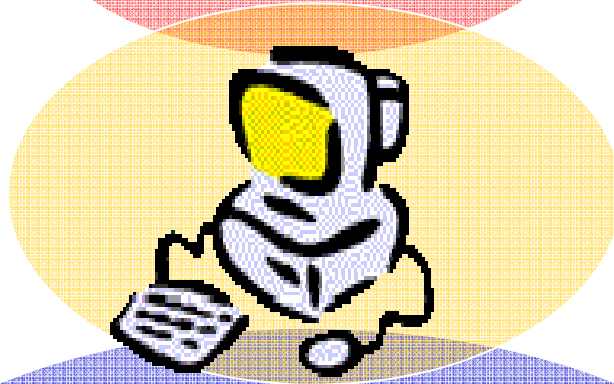


# Introduction générale

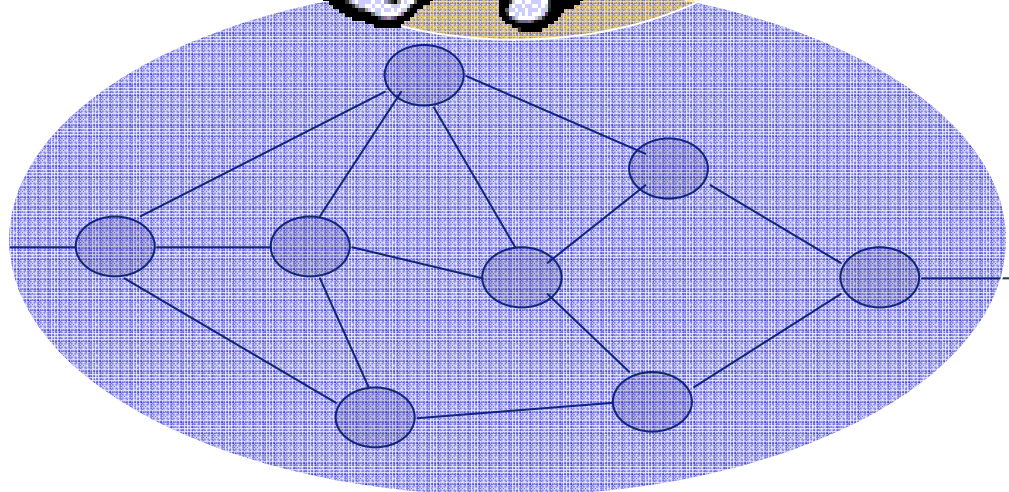
Finalelement, pour faire communiquer des machines, les fonctions à mettre en oeuvre sont nombreuses

Applications de  
communications

Les logiques des processus



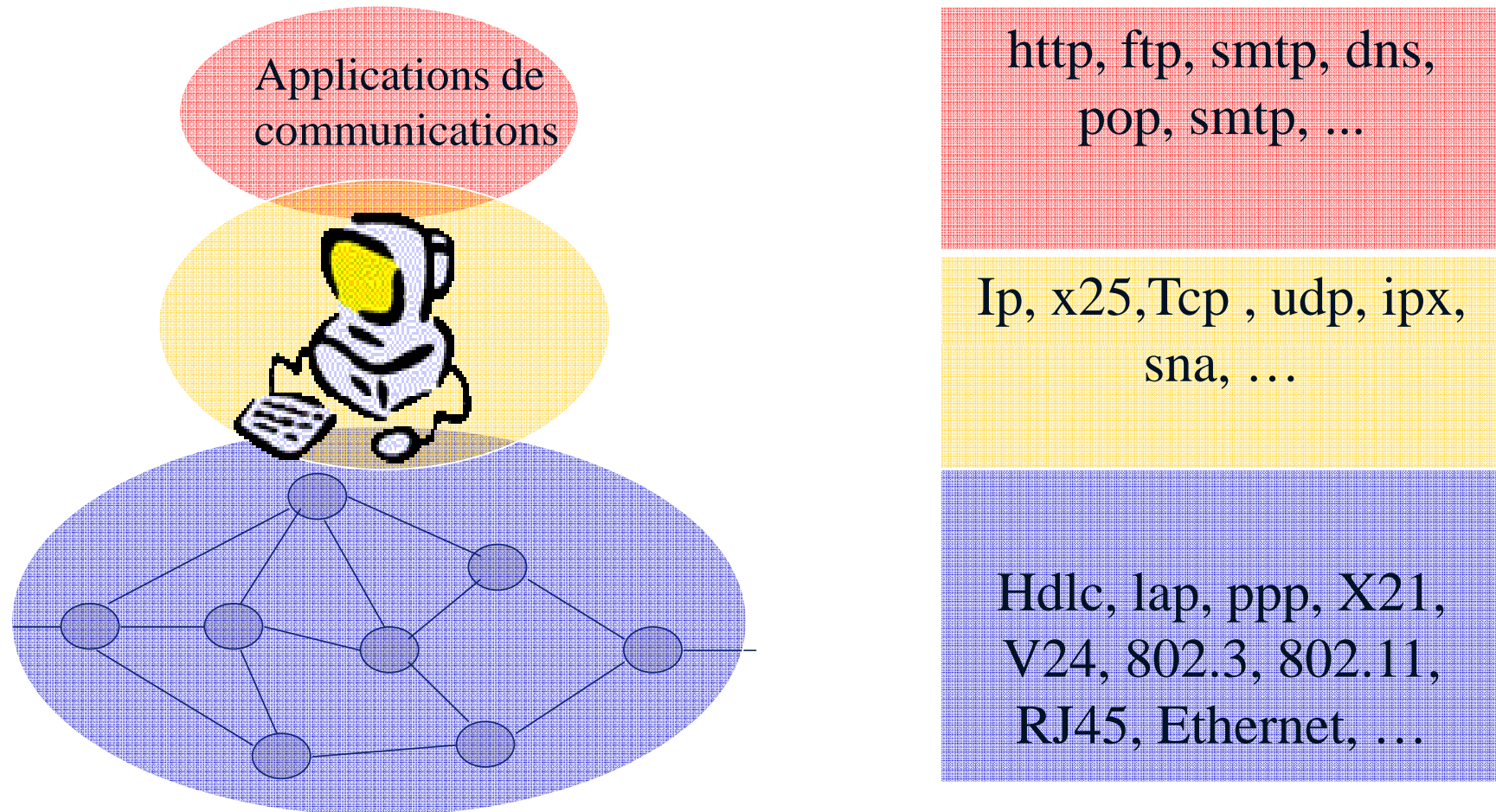
Matériel et procédures  
nécessaires compatibles avec le  
réseau



Composants , structure et  
règles de transfert dans un  
réseau

# Introduction générale

Problème : en l'absence de normes, chaque constructeur crée sa solution.



Conséquence : les solutions proposées sont souvent incompatibles entre-elles → **Besoin de normaliser**

# Introduction générale

Les principaux acteurs de la normalisation

- l'UIT (Union Internationale des Télécoms)
- l'ISO (Organisation Internationale de Standardisation)

Mais aussi

- L'IEEE (Institut des Ingénieurs Electriciens et Electroniciciens)
- France télécom (maintenant Orange)
- Microsoft, DELL, IBM ...

# Introduction générale

## La solution : Le modèle en couche OSI de l'ISO

Couche	Rôle
7 Application	Applications réseau
6 Présentation	Format des données
5 Session	Accès aux données
4 Transport	Transport et contrôle de routage
3 Réseau	Routage des paquets dans plusieurs réseaux
2 Liaison	Contrôle de l'échange entre deux machines
1 Physique	Transmission de signaux binaires

http, ftp, smtp, dns,  
pop, smtp, ...

Ip, x25, Tcp, udp,  
ipx, sna, ...

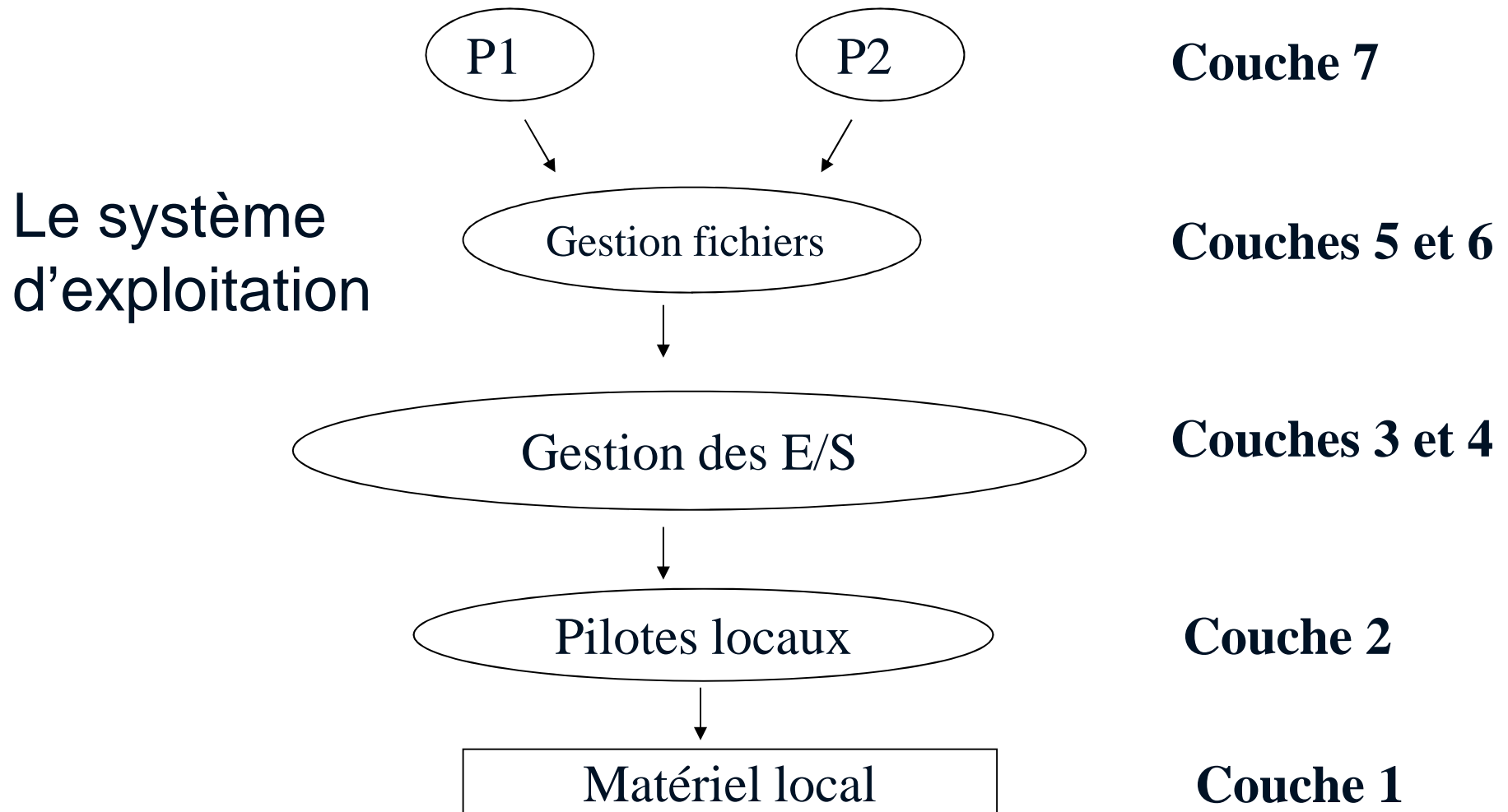
Hdlc, lap, ppp, X21,

V24, 802.3, 802.11,  
RJ45, Ethernet, ...

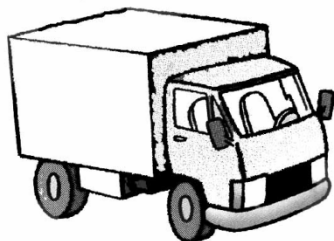
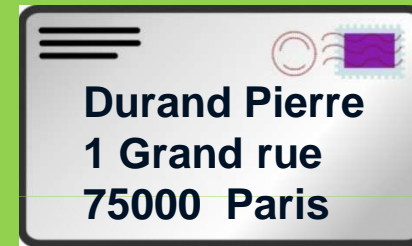
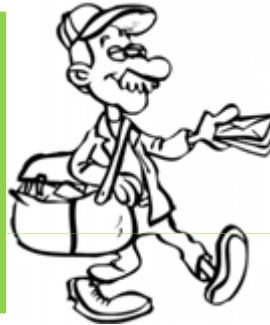
**Objectif du modèle :** découper et hiérarchiser chaque type de problème et proposer une série de solutions pour chacun.

# Introduction générale

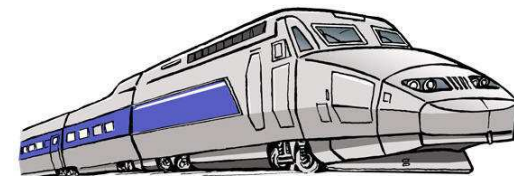
Pourquoi un modèle en couches ?



# Introduction générale

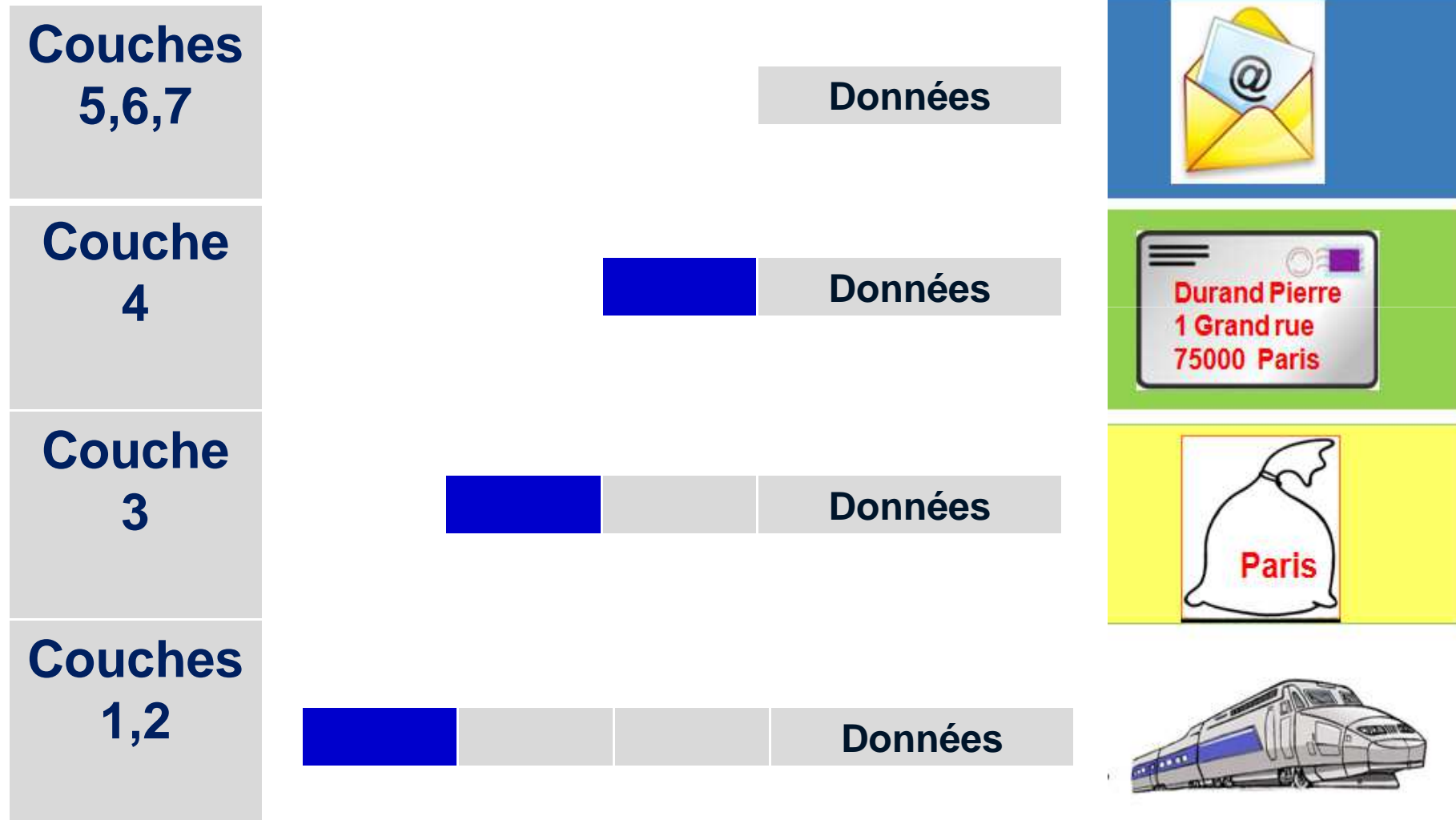


Un exemple de  
modèle en couches



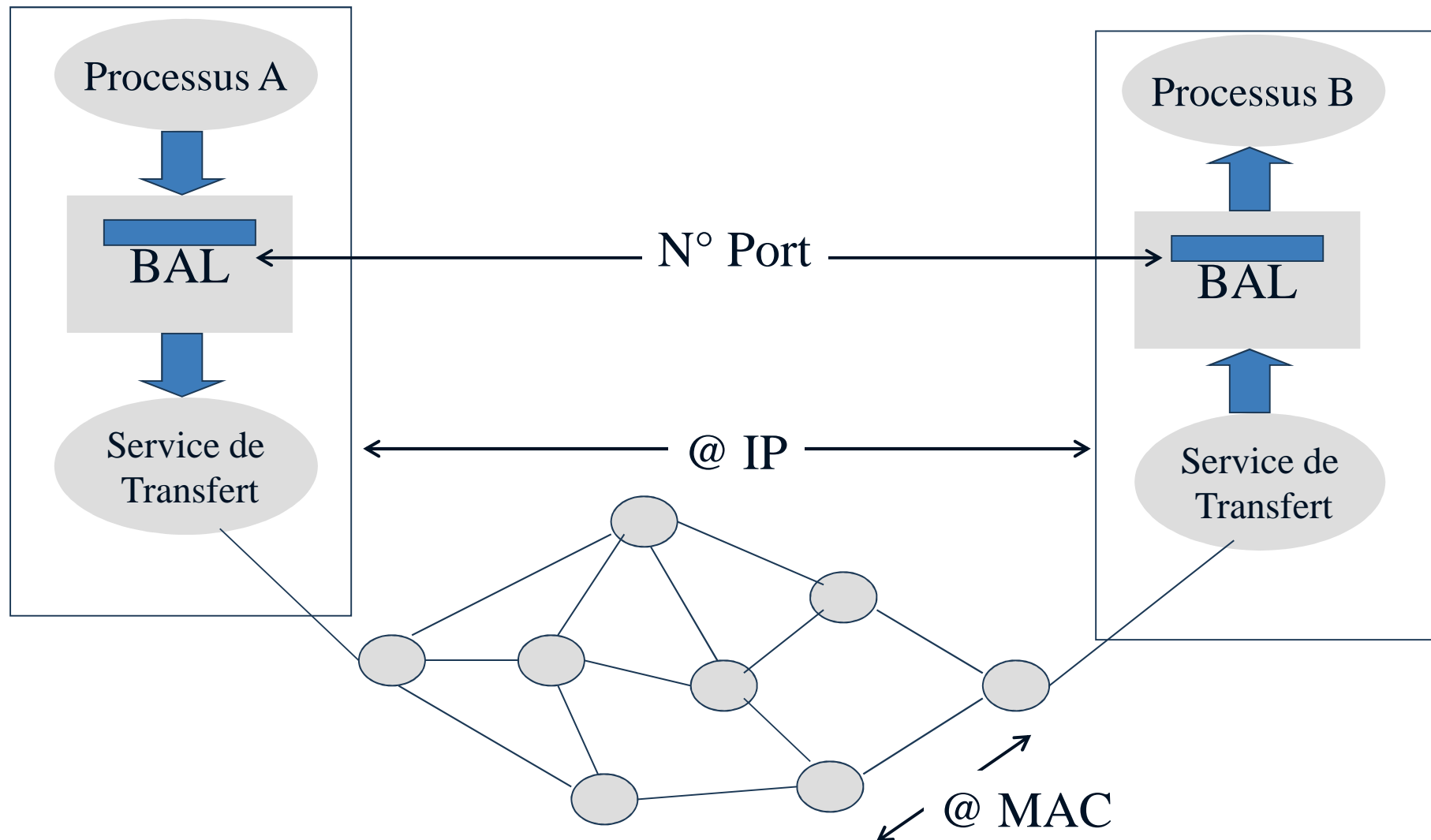
# Introduction générale

## Conséquence des couches : l'encapsulation



# Introduction générale

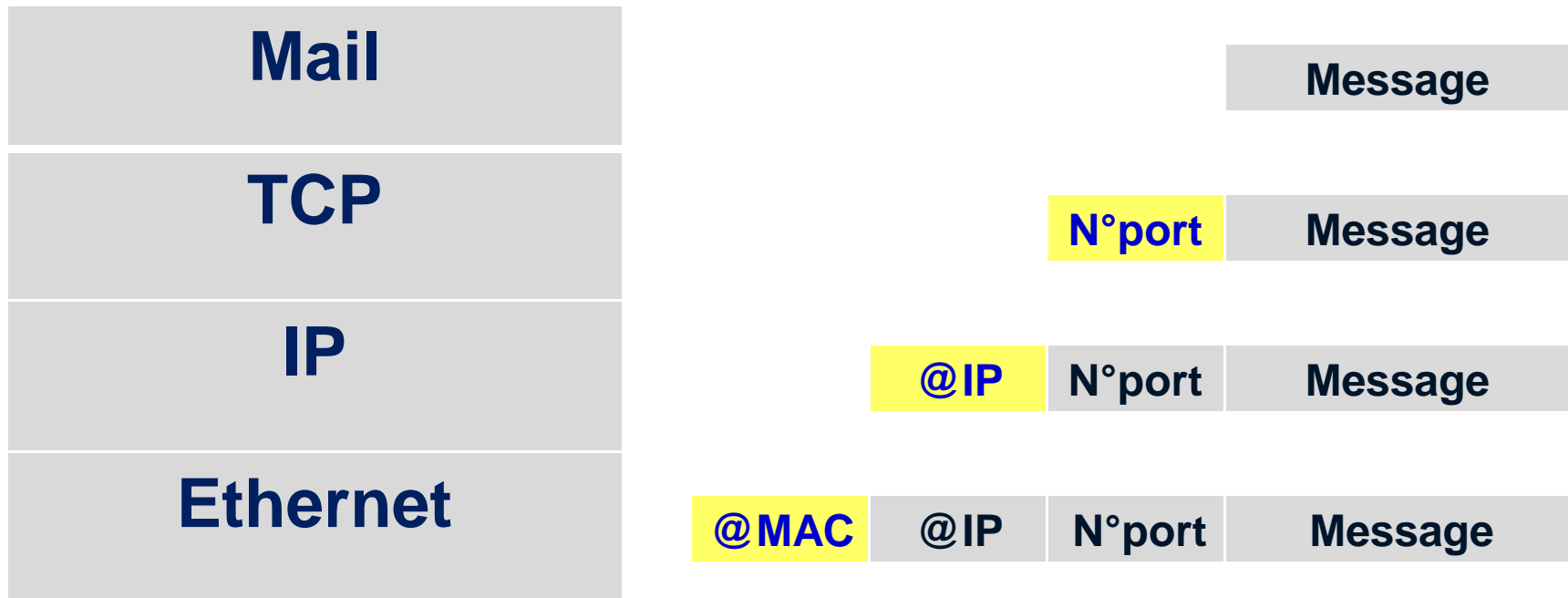
## Exemple d'encapsulation – Réseau internet





# Introduction générale

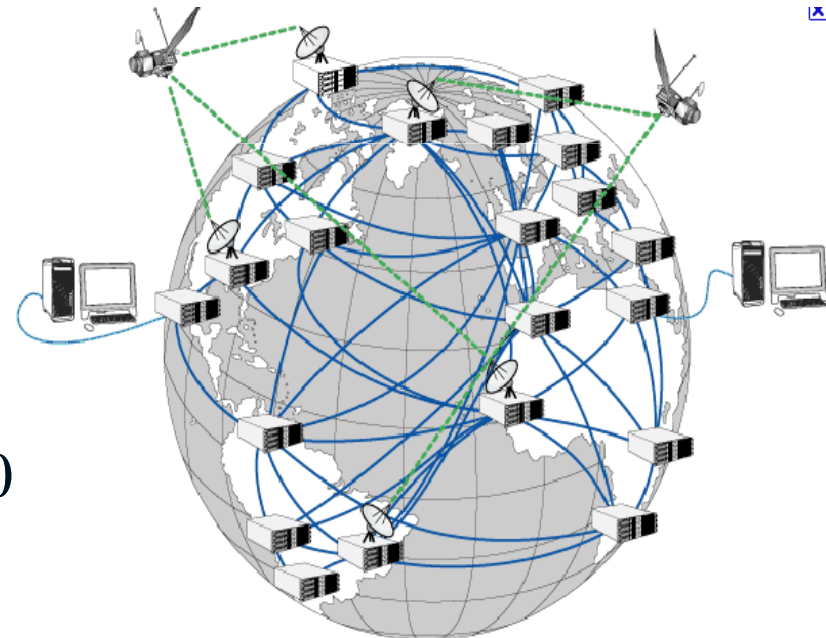
## Autre exemple d'encapsulation



# Introduction générale

Ce modèle est la base de l'internet

- 3,15 milliards d'internautes (43% population mondiale)
- 8 nouveaux utilisateurs chaque seconde
- 634 millions de sites Internet, 822000 nouveaux sites / jour
- 4 milliards de vidéos vues chaque jour sur Youtube
- 204 millions de mails /minute, >200 milliards de mails / jour
- 2,6 millions de CD créés par minute



# **Introduction générale**

**Mais l'internet c'est aussi !!!**

**Un formidable moyen de gagner de l'argent :**

- 75 000 dollars de chiffres d'affaires réalisés par Google chaque minute
- 219 000 dollars/ minute de paiements effectués au moyen de PayPal
- 142 dollars / minute = salaire Bill Gates → 75 M\$/an
- ...

**Un énorme problème de sécurité**

- 9,567 pétaoctets, de données piratées chaque mois (chiffres 2012)
- 432 millions de pirates à travers le monde.
- 232 ordinateurs infectés par un virus toutes les minutes en France
- 416 tentatives de hacking par minute en France
- ...

# Plan du cours

Dans ce cours ne  
seront présentées  
que les couches 1 à  
4 du modèle OSI.

Type de problème		Couche
Echange entre processus	7	Application
	6	Présentation
	5	Session
Fonctions de transport	4	Transport
Techniques et algorithmes de routage	3	Réseau
Echange entre 2 machines	2	Liaison
Matériel de connexion	1	Physique

# Plan du cours

## 1 Techniques de base de la transmission (couche 1)

- Elements de la transmission
- Codage du signal
- Modes de transmission
- Supports de transmission et équipements d'interconnexion
- Introduction au contrôle des erreurs

## 2 Les réseaux locaux (couche 2)

- Etude des différents réseaux
- Le réseau Ethernet et 802.3
- Initiation au câblage
- Exemples de réseaux

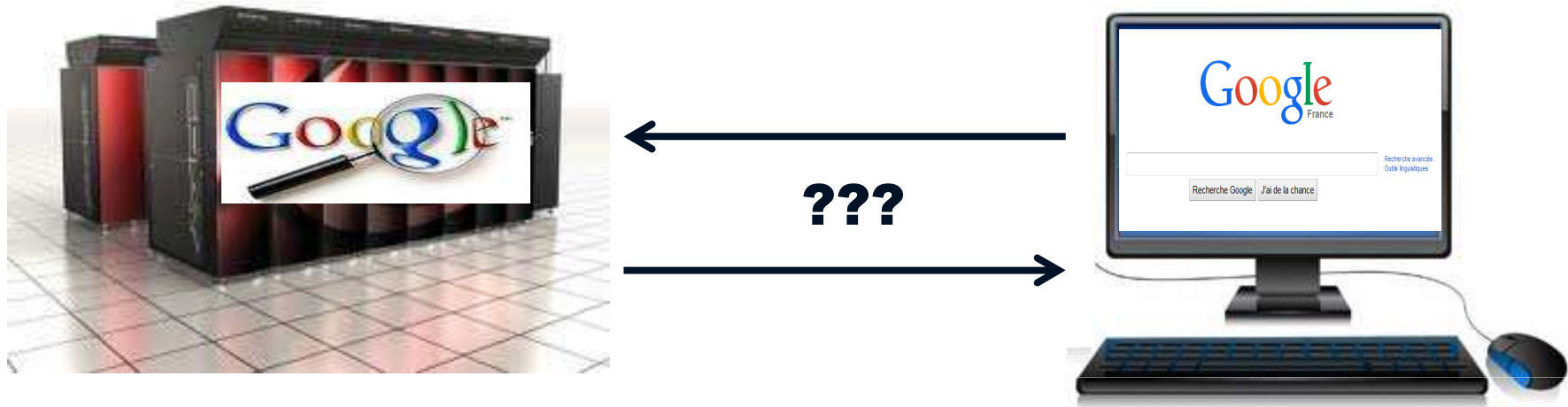
## 3 L'interconnexion de réseaux et l'architecture TCP/IP (couches 3 et 4)

- La couche internet - IP : Les mécanismes, L'adressage, Le routage
- Les protocoles annexes (ARP, DNS, DHCP, ...)
- La couche transport (TCP et UDP)

# **Les réseaux informatiques**

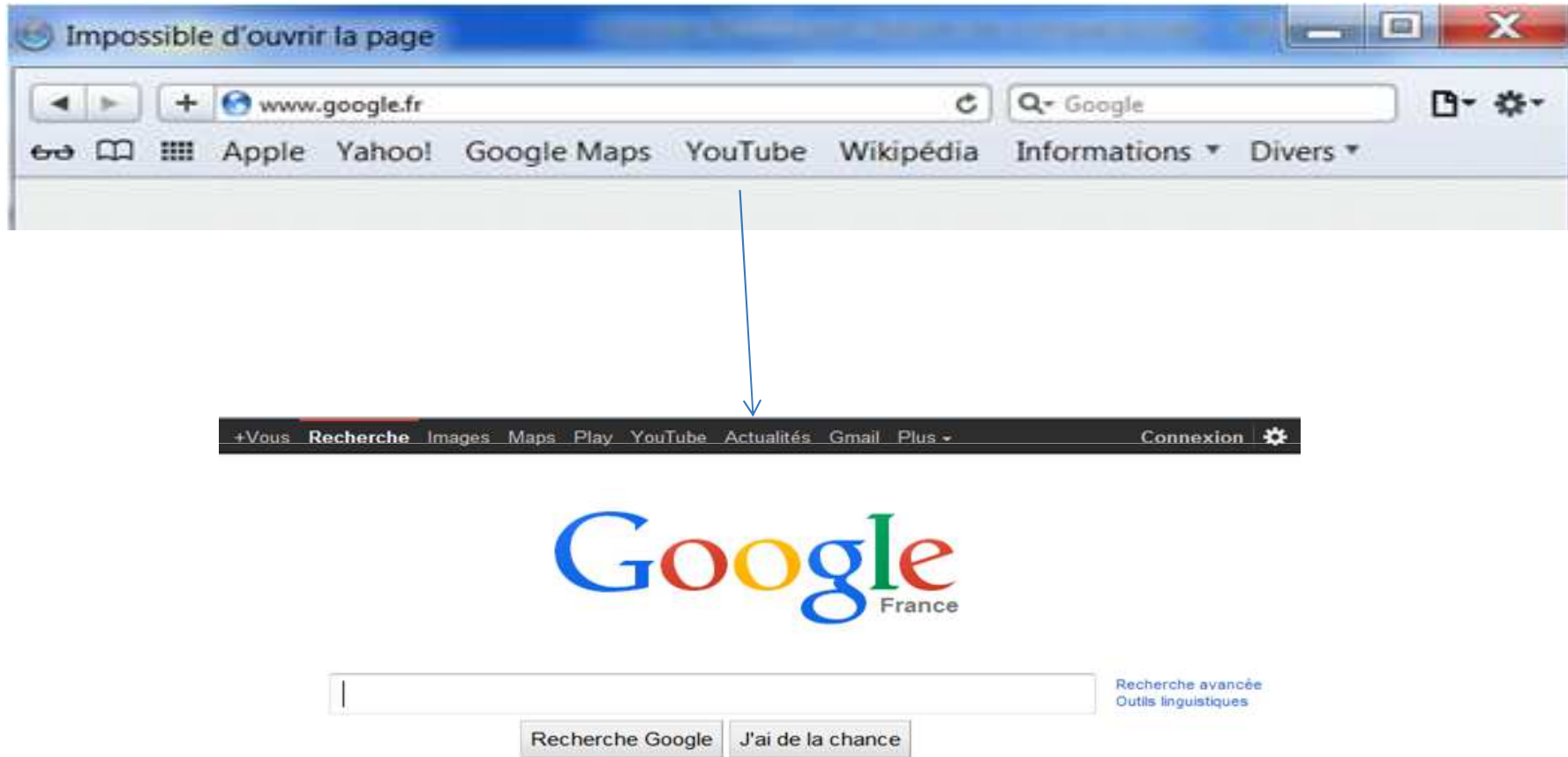
**Pour terminer !**

# Pour terminer



Ce cours devrait vous permettre de comprendre comment les machines communiquent dans un réseau.

# Pour terminer



Tout ce qui se passe entre votre navigateur et un site web,  
ne devrait plus avoir de secret pour vous !!



# Pour terminer



ARP	00-60-08-61-04-7b	Toutes machines	Qui connaît l'adresse physique de 10.10.1.1 ?
ARP	00-01-02-af-f5-e2	00-60-08-61-04-7b	L'adresse physique de 10.10.1.1 est 000102aff5e2
DNS	10.10.20.2 + 1234	10.10.1.2 + 53	Adresse IP de www.google.fr ?
DNS	10.10.1.2 + 53	10.10.20.2 + 1234	Adresse IP de www.google.fr est 173.168.12.1
TCP	10.10.20.2 + 1234	173.168.12.1 + 50	syn = demande connexion TCP au service HTTP
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack + syn : acceptation et demande connexion
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack = acceptation connexion
HTTP	10.10.20.2 + 1234	173.168.12.1 + 80	Get /http://...
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack
HTTP	173.168.12.1 + 80	10.10.20.2 + 1234	Document HTML
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack + fin
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	ack: acceptation de déconnexion
TCP	173.168.12.1 + 80	10.10.20.2 + 1234	fin → on termine la déconnexion
TCP	10.10.20.2 + 1234	173.168.12.1 + 80	ack

**Vous comprendrez les codes cachés de l'internet  
et ainsi ...**

# **Bibliographie**

Réseaux et Télécoms - Claude Servin - Dunod.

Les réseaux - Guy PUJOLLE - Eyrolles

Réseaux - A. TANENBAUM – Pearson

L'architecture des réseaux TCP/IP - Jacques Philipp -  
Ellipses