



# Développement d'applications réparties : Introduction

Aïcha Ben Jrad

3ème Année Licence en Sciences de l'Informatique

# Historique – émergence du réparti

- ❑ 1945 – 1980 : mainframes (Ordinateurs – gros et chers)
  - Les entreprises possédaient des ordinateurs qui travaillaient localement
- ❑ 1980 - ... : Deux évolutions technologiques majeures
  1. Développement de microprocesseurs puissants
    - Machines 8-bits ----> Machines 16-bits, 32-bits et 64-bits
    - Leurs puissance de calcul comparable à celle d'un gros ordinateur
    - Prix nettement inférieur
  2. Invention des réseaux à haut débit (*LAN - (Local-area network ) / WAN - (Wide-area network)*)
    - Possibilité d'interconnecter des centaines/milliers d'ordinateurs pour échanger de petites quantités d'informations en une milliseconde environ.

**Conséquence:** Des volumes de données plus importants peuvent être échangés à un débit de 10 Mbits/s ou plus

# Historique – émergence du réparti

## ❑ Evolution des besoins

- personnels :
  - Génération Y
  - Besoin d'être connecté
  - Gourmand en ressource
- Entreprises :
  - Accès à distance
  - Collaboration
  - Plus de ressources

**Conséquence:** convergence entre l'évolution de la technologies & des besoins des utilisateurs ---> du système centralisé vers un système composé de plusieurs machines connectées par un réseau

# Classification des systèmes informatiques – Flynn (1992)

- ❑ **SISD (Single Instruction on Single Data) :**
  - Systèmes mono-processeur + Aucun degré de parallélisme
- ❑ **SIMD (Single Instruction on Multiple Data) :**
  - Processeur vectoriel : effectuer des opérations mathématiques sur des éléments de données multiples
- ❑ **MISD (Multiple Instructions on Single Data) :**
  - La même donnée est traitée par plusieurs processeur en parallèle.
- ❑ **MIMD (Multiple Instructions on Multiple Data) :**
  - Les machines multi-processeurs où chaque processeur exécute son code de manière asynchrone et indépendante.
  - L'architecture parallèle la plus utilisée

# Classification des systèmes informatiques – Flynn (1992)

## ❑ MIMD à mémoire partagée (fortement couplées) :

- Les processeurs accèdent à une mémoire commune
- Machines parallèles

## ❑ MIMD à mémoire distribué (faiblement couplées) :

- Chaque processeur dispose de sa propre mémoire, et n'a pas accès à celle d'autres processeurs
- **Systèmes répartis/distribués**

# Qu'est-ce qu'un système réparti ?

## Définitions :

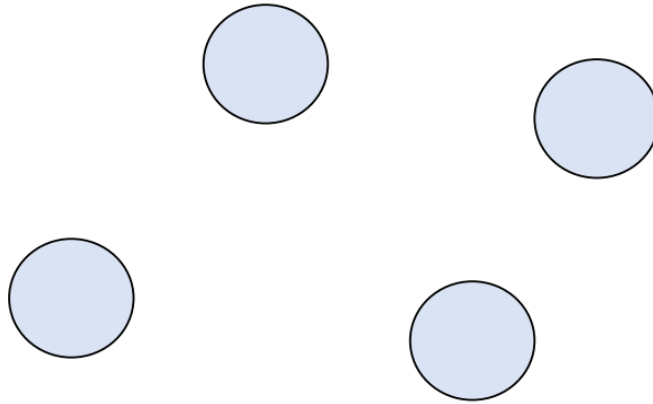
*"A distributed system is a collection of **autonomous computing elements**, connected by a **network**, which **appears** to its users as **a single coherent system**", Distributed Systems, Maarten van Steen & Andrew S. Tanenbaum*

*"... a system in which a failure of a computer you didn't even know existed can render your own computer unusable", Leslie D. Lamport*

*"A system in which **components** located on **networked** computers communicate and **coordinate** their actions by **passing messages**. The components interact with each other in order to achieve a **common goal**.", Wikipedia*

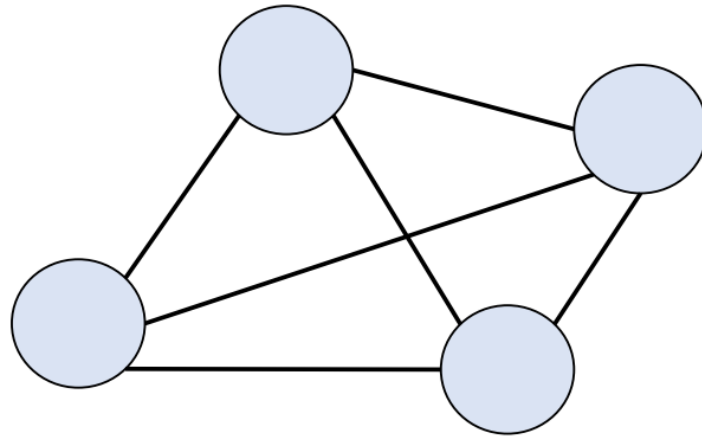
# Qu'est-ce qu'un système réparti ?

---



Composants (éléments) indépendants  
(processus logiciel ou pièces matérielles utilisé(e)s pour exécuter un programme, sauvegarder des données, etc.)

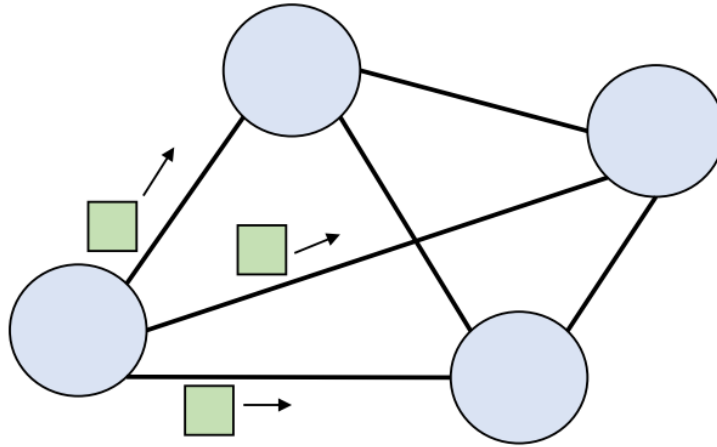
# Qu'est-ce qu'un système réparti ?



**Composants (éléments) indépendants qui sont connectés par un réseau.**

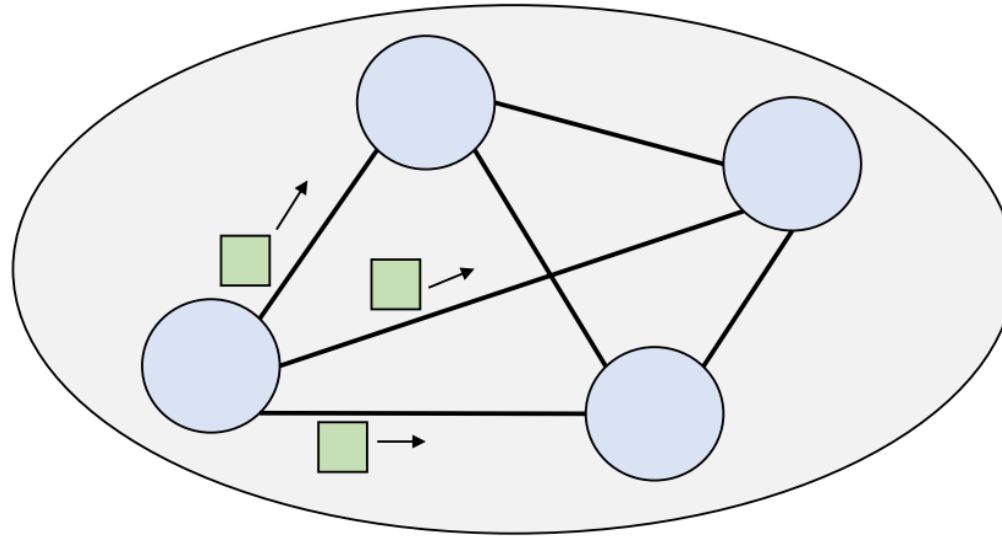


# Qu'est-ce qu'un système réparti ?



**Composants (éléments) indépendants qui sont connectés par un réseau et communiquent en passant des messages.**

# Qu'est-ce qu'un système réparti ?



**Des Composants (éléments) indépendants** qui sont **connectés par un réseau** et communiquent **en passant des messages** pour atteindre un objectif commun, et présentés à l'utilisateur **comme un système unique cohérent.**

# Systèmes répartis

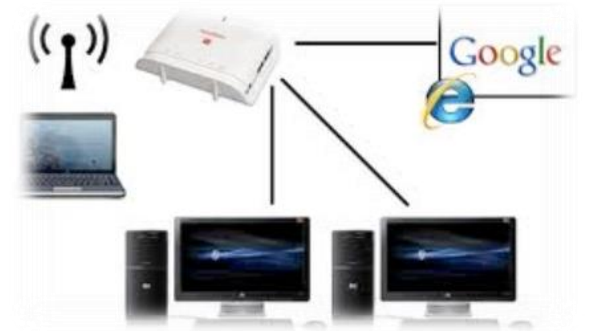
## ❑ *Vision matérielle d'un système réparti*

- Cluster d'ordinateurs : ordinateurs puissants et indépendants
- Ordinateurs standards connectés en réseau



## ❑ *Vision logicielle d'un système réparti (système)*

- ❑ Plusieurs entités logicielles s'exécutant indépendamment et en parallèle



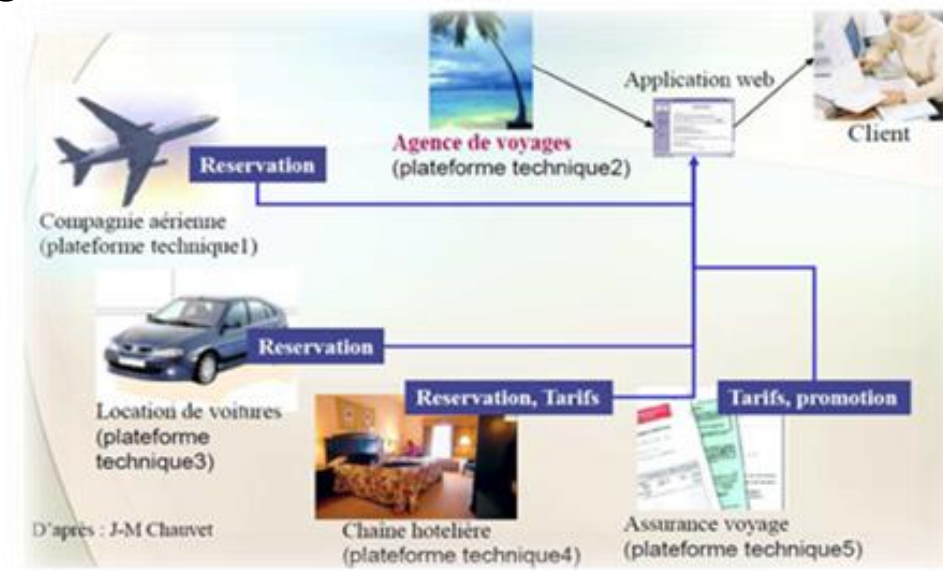
# Exemples

## ☐ *Serveurs de BD ou de fichiers*

- ☐ Physiquement : fichiers stockés sur un serveur
- ☐ Virtuellement : accessible de n'importe quelle machine, en faisant croire que ces fichiers sont stockés localement

## ☐ *Web*

- ☐ Accès distant à l'information
- ☐ Transparence de localisation



# Propriétés attendues d'un système réparti

## ❑ Disponibilité

- Prêt à l'utilisation et toujours accessible

## ❑ Hétérogénéité

- Utiliser des composants de différentes natures
  - Machines
  - Systèmes d'exploitation
  - Environnements de programmation
  - Protocoles de communications
  - ...

# Propriétés attendues d'un système réparti

## ❑ Tolérance aux pannes

- Tolère les défaillances de ses composants

## ❑ Sécurité

- Possibilités de combiner plusieurs techniques de sécurité

## ❑ Transparence

- Propriété fondamentale: tout cacher à l'utilisateur
  - La répartition doit être non perceptible : une ressource distante est accédée comme une ressource locale
  - Cacher l'architecture et le fonctionnement du système réparti

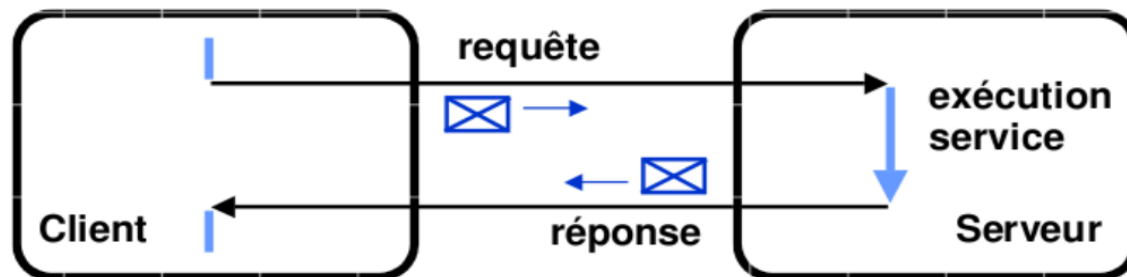
## ❑ ...

# Systèmes répartis vs Applications réparties

- ❑ **Système réparti** : des procédures réparties pour la gestion des ressources communes et de l'infrastructure, liées au matériel
  - Système d'exploitation : gestion de chaque élément
  - Système de Gestion de Bases de Données -- SGBD
  
- ❑ **Application répartie** : des **procédures réparties sur un réseau** et interagissant sur des **données réparties**, afin d'effectuer une tâche globale mais spécifique; donc il y a **dépendance entre les procédures**
  - Elle utilise les services généraux du système réparti

# Modèles de répartition

- ❑ Les composants d'une application répartie interagissent entre eux selon plusieurs modèles possibles :
  - **Modèle Client/Serveur : Requête/Réponse**
    - Le client envoie une requête et reste bloqué en attente de la réponse (communication synchrone)
    - Le serveur passif attend une requête cliente, exécute le service demandé et renvoie une réponse au client





# Modèles de répartition

- **Modèle de communication par messages :**
  - Une application produit des messages (**producteur**) et une application les consomme (**consommateur**)
  - Le producteur (**émetteur**) et le consommateur (**récepteur**) ne communiquent pas directement entre eux mais utilisent un objet de communication intermédiaire (boîte aux lettres ou file d'attente)
  - Communication **asynchrone** :
    - ✓ Les deux composants n'ont pas besoin d'être connectés en même temps grâce au système de file d'attente
    - ✓ Emission **non bloquante**: l'entité émettrice émet son message, et continue son traitement sans attendre que le récepteur confirme l'arrivée du message
    - ✓ Le récepteur récupère les messages quand il le souhaite

# Modèles de répartition

---

- **Modèle de communication par événements**
- **Modèles à base de composants**
- ...

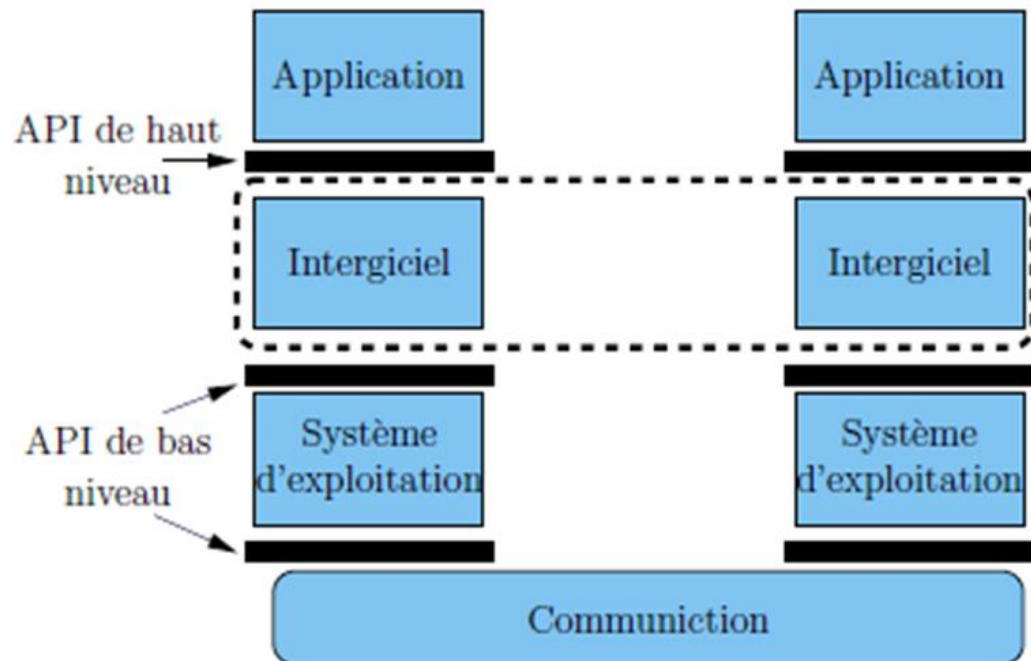
# Middleware (Intergiciel)

- ❑ Un middleware ou intergiciel est un logiciel utilisé par différentes applications pour communiquer entre elles
- ❑ **Motivations**
  - Interface OS trop complexe pour programmation d'applis distribuées
    - Hétérogénéité
    - Complexité des mécanismes (bas niveau)
    - Nécessité de gérer (voire masquer) la répartition
- ❑ **Solution : l'intergiciel**
  - Couche logicielle **intermédiaire** (répartie) entre les **niveaux bas** (systèmes et communication) et le **niveau haut** (applications) : c'est l'**intergiciel**

# Middleware (Intergiciel)

- ❑ Un middleware ou "intergiciel" permet le dialogue entre les différents composants d'une application répartie.

## Couche du milieu



## Fonctions

- ✓ Fournir une API adaptée
- ✓ Masquer l'hétérogénéité
- ✓ Masquer la répartition
- ✓ Portabilité
- ✓ Interopérabilité

# Types de middleware

---

- ❑ Middleware pour les langages procéduraux de type RPC
  - ✓ RPC : Remote Procedure Call
- ❑ Middleware pour les langages orientés objets de type RMI
  - ✓ RMI : Remote Method Invocation

RMI = RPC orienté objet

❑ ...

# Mécanismes de communication dans les SR

- ❑ Mécanismes de communication de "bas niveau"
  - ✓ **Sockets** : mécanisme universel de bas niveau, utilisable depuis tout langage (exemple, C et Java)
  
- ❑ Mécanismes de communication de "haut niveau"
  - ✓ Utiliser les services offerts par les middlewares :
    - Appel de procédure à distance (RPC)
    - Appel de méthode à distance (RMI)
    - ...

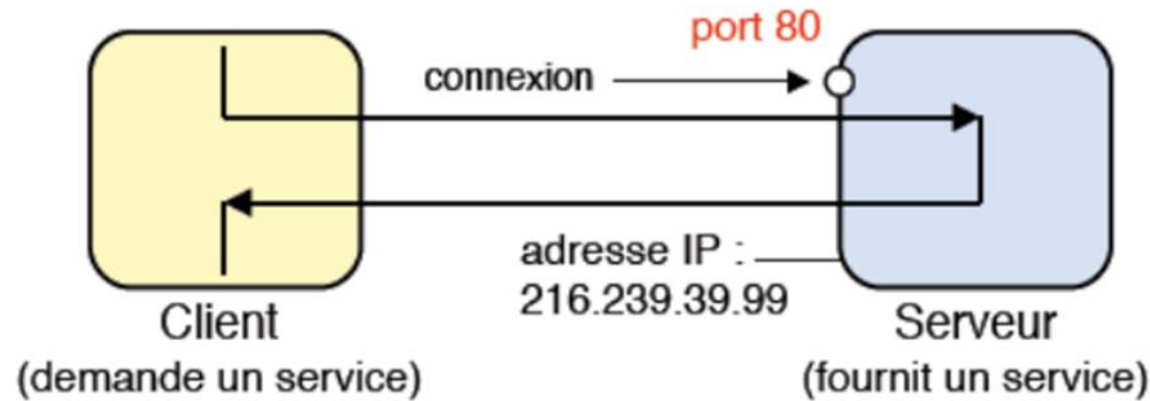
# Le réseau vu de l'utilisateur

- ❑ Un client demande un service fournit par un serveur sur une autre machine :
  - Un service est souvent désigné par un nom symbolique (email, <http://...>, etc.)
  - Ce nom doit être converti en une adresse interprétable par les protocoles du réseau
  - La conversion d'un nom symbolique (<http://www.google.com>) en une adresse IP (216.239.39.99) est à la charge du service DNS



# Le réseau vu de l'utilisateur

- ❑ Le serveur (machine physique) peut comporter différents services :
  - L'adresse IP du serveur ne suffit pas
  - Il faut préciser le service demandé au moyen d'un numéro de port, qui permet d'atteindre un processus particulier sur la machine serveur
- ❑ Un numéro de port comprend 16 bits (0 à 65535)
- ❑ Les numéros de 0 à 1023 sont réservés, par convention, à des services spécifiques. Exemples 80 – serveur web, 25 – mail, ...





# Conclusion

---

- ❑ *"Un système réparti est un système qui s'exécute sur un ensemble de machines sans mémoire partagée, mais que pourtant l'utilisateur voit comme une seule et unique machine"*
- ❑ **Objectifs** : partage de ressources, tolérance aux pannes, coûts, contraintes physiques, scalabilité
- ❑ Un mot clé important à retenir : **Transparence**