

# Chapitre 2 :

**Principaux problèmes soulevés et services offerts par  
un système réparti**

# Cohérence et synchronisation (1)

- **Architecture :**
  1. Centralisée = 1 horloge. Ordre total.
  2. Répartie = plusieurs horloges non synchronisées + communications **asynchrones**  $\implies$  plus d'état global facilement calculable et présence d'**indéterminisme logique**.
- **Problèmes** : partage de données réparties et coordination répartie.
- **Solutions** : algorithmes dédiés d'élection, d'exclusion mutuelle, de consensus, de terminaison, etc.

# Cohérence et synchronisation (2)

- **Exemple 1 : partage de données réparties**
  - Echanges d'email  $\implies$  consensus réparti
  - Réponses reçues avant les questions.
- **Exemple 2 : coordination répartie**
  - Mise au point d'un programme : comment arrêter plusieurs processus répartis simultanément ?  $\implies$  synchronisation.
  - Comment réexécuter une application ?  $\implies$  indéterminisme logique.

# Désigner les ressources du système

- "Désigner un objet consiste à lui affecter un nom permettant de lui faire référence" [MAR 88].
  - Nom indépendant de la localisation géographique (pannes, migrations) et du temps.
  - Nom = ressource  $\implies$  réutilisation.
  - **Adresse**  $\neq$  **nom**. Translation nom  $\implies$  adresse = service de localisation (chaîne de références, diffusion, fonctionnelle, serveur de localisation).
  - Espace de nommage/adressage, contexte (ex : désignation hiérarchique).
  - Serveur centralisé ou par type de ressource (système de fichiers) ; performance, résistance aux pannes, facilité de réalisation, répartition ou non du catalogue (cohérence).

# Support de l'hétérogénéité

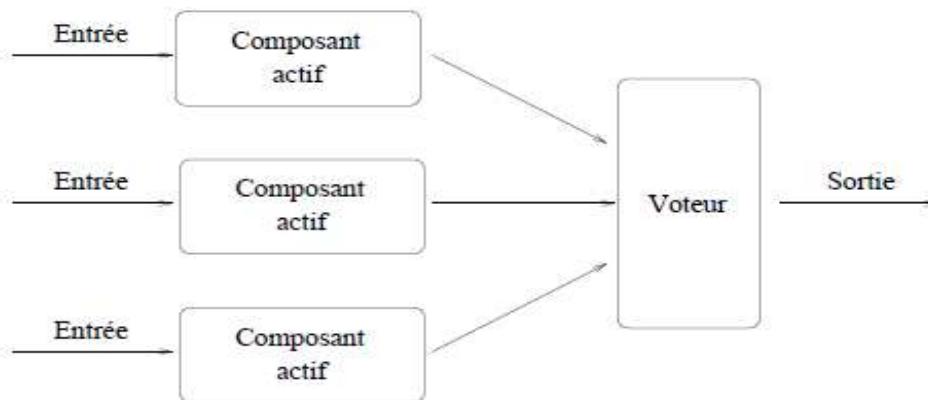
- **Services interopérables** : comment s'affranchir des différences matérielles et logicielles des machines du système ?
- **Sources d'hétérogénéité** :
  1. La représentation des données en mémoire :
    - Le problème des petits indiens (systèmes big indian et little indian).
    - Représentation des flottants (ex : PC sur 80 bits, norme IEEE en 64 et 128 bits).
    - Entiers complémentés à 1 ou 2.
    - Alignement des données en mémoire.
  2. Les langages de programmation, les exécutables, les systèmes d'exploitation.
  3. Les protocoles de communication.

# Transparence aux pannes (1)

- **Objectif** : cacher à l'utilisateur l'occurrence de pannes. Reste un problème difficile dans un système asynchrone.
- *Un système distribué est un système avec lequel je ne peux rien faire car une machine que je ne connais pas est en panne". Lamport.*
- Type de pannes : pannes temporelles, pannes dites "silencieuses" (crash), pannes byzantines.
- Solutions par points de reprise globaux  $\implies$  comment prendre un état cohérent du système (message du futur).
- Solutions à base de redondance.

# Transparence aux pannes (2)

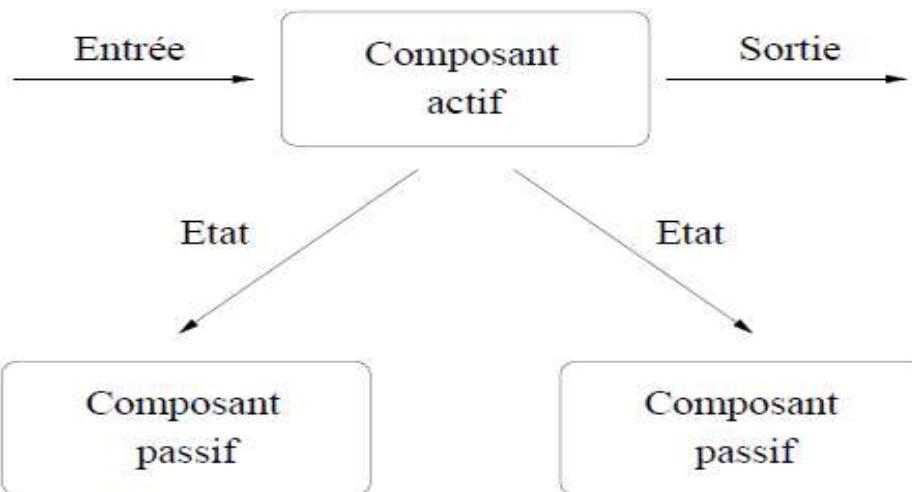
- Redondance active :



- Groupe de serveurs + vote (consensus).
- Application déterministe.
- Consensus impossible dans un système asynchrone.

# Transparence aux pannes (3)

- Redondance passive :



- Maître/esclaves (journalisation).
- Application non déterministe mais systèmes coûteux.
- Quoi et comment journaliser ?