

## **UN SYSTÈME D'EXPLOITATION = DES SERVICES COMMUNS**

### **GESTION DES DIVERS PROCESSUS**

- **processus locaux ou distants représentés localement (proxys).**

### **GESTION DES RESSOURCES PHYSIQUES :**

- **mémoire principale pour y placer l'information propre à un utilisateur ou l'information partagée,**
- **mémoire secondaire,**
- **utilisation et partage des processeurs (unités centrales, canaux d'entrées-sorties,..)**
- **organes d'entrées-sorties et d'IHM (interfaces homme machine),**
- **canaux de communication (réseau local LAN ou étendu WAN).**

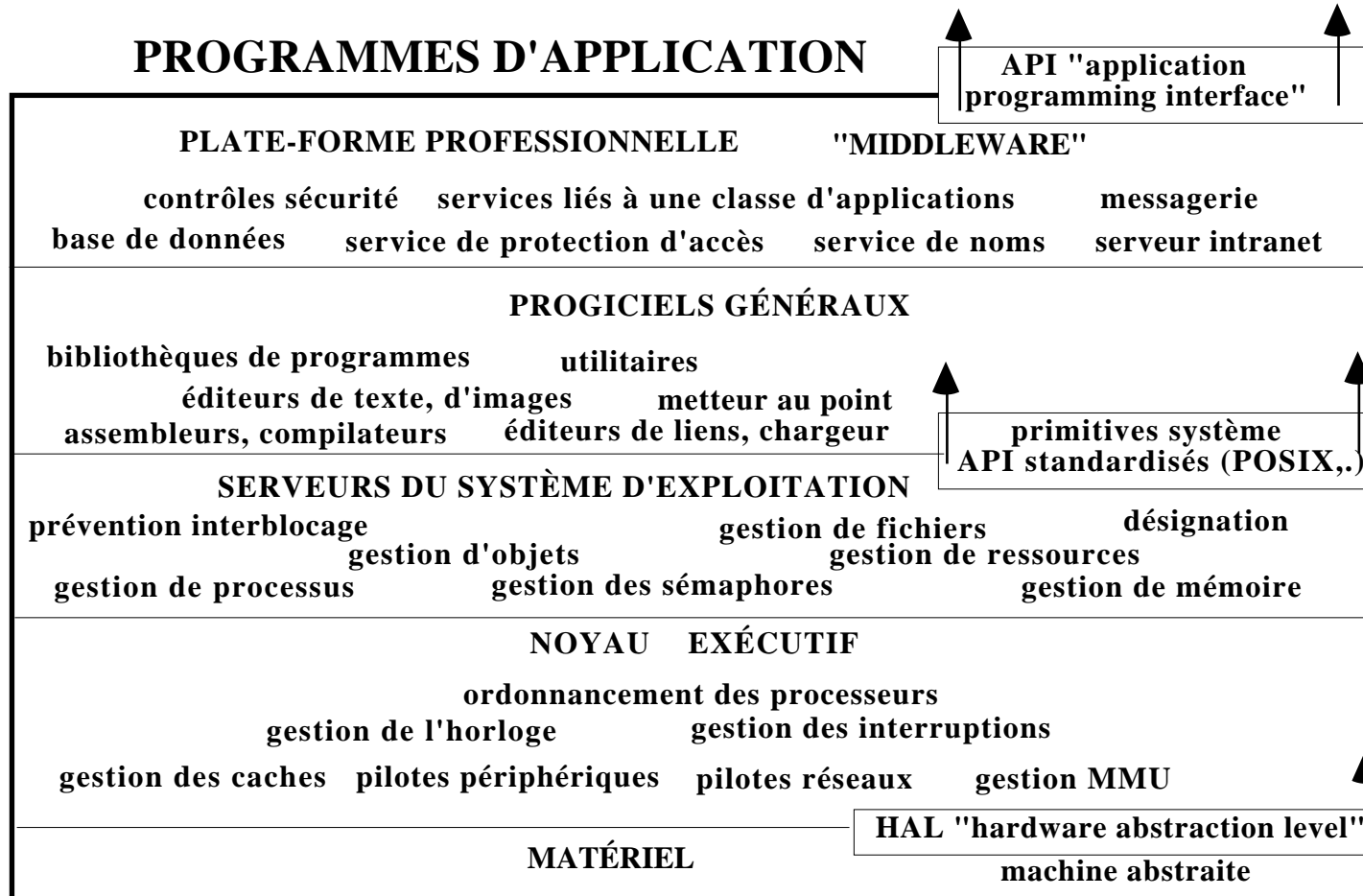
### **STOCKAGE ET ÉCHANGE D'INFORMATION ENTRE UTILISATEURS LOCAUX OU DISTANTS.**

### **PARTAGE ET PROTECTION**

- **Protection mutuelle des utilisateurs, protection contre des intrusions, sécurité d'utilisation**

### **SERVICES DIVERS**

- **facturation des ressources,**
- **mesures de fonctionnement.**
- **statistiques d'utilisation**



**PRÉSENTATION CLASSIQUE EN COUCHES DE SERVICES**

## **DIVERSITÉ DES SYSTÈMES D'EXPLOITATION**

### **DIVERSITÉ DES MODES D'UTILISATION**

- **ordinateurs personnels individuels plus ou moins puissants**
- **mode différé et le traitement par lots, optimise l'utilisation et le rendement des ressources**
- **mode temps réel avec souci de respecter des contraintes temporelles, comme des échéances**
- **mode interactif, optimise le temps de réponse**
  - **temps partagé, travail comme sur un ordinateur individuel, plus des services communs**
  - **transactionnel avec transactions de consultation et mise à jour de données communes**
- **réseaux et systèmes répartis**
- **systèmes spécialisés (embarqués, enfouis) et systèmes généraux**

### **DIVERSITÉ DES UTILISATEURS**

- **utilisateur final,**
- **programmeur ou concepteur d'applications**
- **administrateur de réseaux, de bases de données**
- **gestionnaire d'utilitaires,**
- **ingénieur système,**
- **concepteur de système**

### **DIVERSITÉ DES MATÉRIELS ET DES ARCHITECTURES ORGANIQUES**

- **monoprocasseur, multiprocesseurs, réseau local, réseau général,**
- **mémoires secondaires**
- **périphériques spécialisés ou généraux**

# COMPORTEMENT DES SYSTÈMES INFORMATIQUES

## SYSTÈMES TRANSFORMATIONNELS

- ils gèrent des programmes dont :
  - les résultats sont calculés à partir de données disponibles dès l'initialisation du programme
  - les instants de production des résultats ne sont pas contraints

## SYSTÈMES INTERACTIFS

- ils gèrent des programmes dont :
  - les résultats sont fonctions de données produites par l'environnement du programmeur
  - les instants de production des résultats ne sont pas contraints
- temps partagé : assurer un temps de réponse court pour des demandes indépendantes (édition, compilation, calcul, consultation multimédia)
- transactionnel : garantir le contrôle et la cohérence d'accès à des "fichiers" partagés (bases de données) en consultation et mise à jour

## SYSTÈMES RÉACTIFS

- ils gèrent des programmes dont :
  - les résultats sont fonctions de données produites par l'environnement du programme (le procédé à contrôler)
  - les instants de saisie des données et de production des résultats sont contraints par les dynamiques du procédé contrôlé
- temps réel : contrôle de processus industriels, de trafic, de production

## **PROPRIÉTÉS GLOBALES**

### **GARANTIE DE PERMANENCE DU SERVICE**

**sécurité, protection, cohérence, tolérance aux pannes**

### **CAPACITÉ D'ÉVOLUTION ET DE SUIVI DE LA DIFFUSION DE L'INFORMATIQUE**

**évolutivité, ouverture et intégration des systèmes**

### **INTERFACE AVEC L'UTILISATEUR**

**qualité de service, performances, ergonomie**

### **CRITÈRES DE PERFORMANCES ET INDICATEURS MACROSCOPIQUES**

**temps de traitement : durée d'exécution de la commande seule**

**temps de réponse : délai entre envoi de la commande et l'arrivée de la réponse**

**débit : nombre de commandes exécutées par unité de temps**

### **OBJECTIFS DE PERFORMANCE (contradictaires)**

**réduire le temps de réponse**

**augmenter le débit global**

**respecter les contraintes temporelles (échéance, absence de gigue)**

## MORPHOLOGIE DES SYSTÈMES INFORMATIQUES

### LA DIVERSITÉ DES UTILISATIONS, DES BESOINS

=> diversité des vues, des abstractions, des concepts

=> plusieurs niveaux de service, de fonctionnalités => concept de hiérarchisation

### A CHAQUE NIVEAU, FOURNITURE À L'UTILISATEUR :

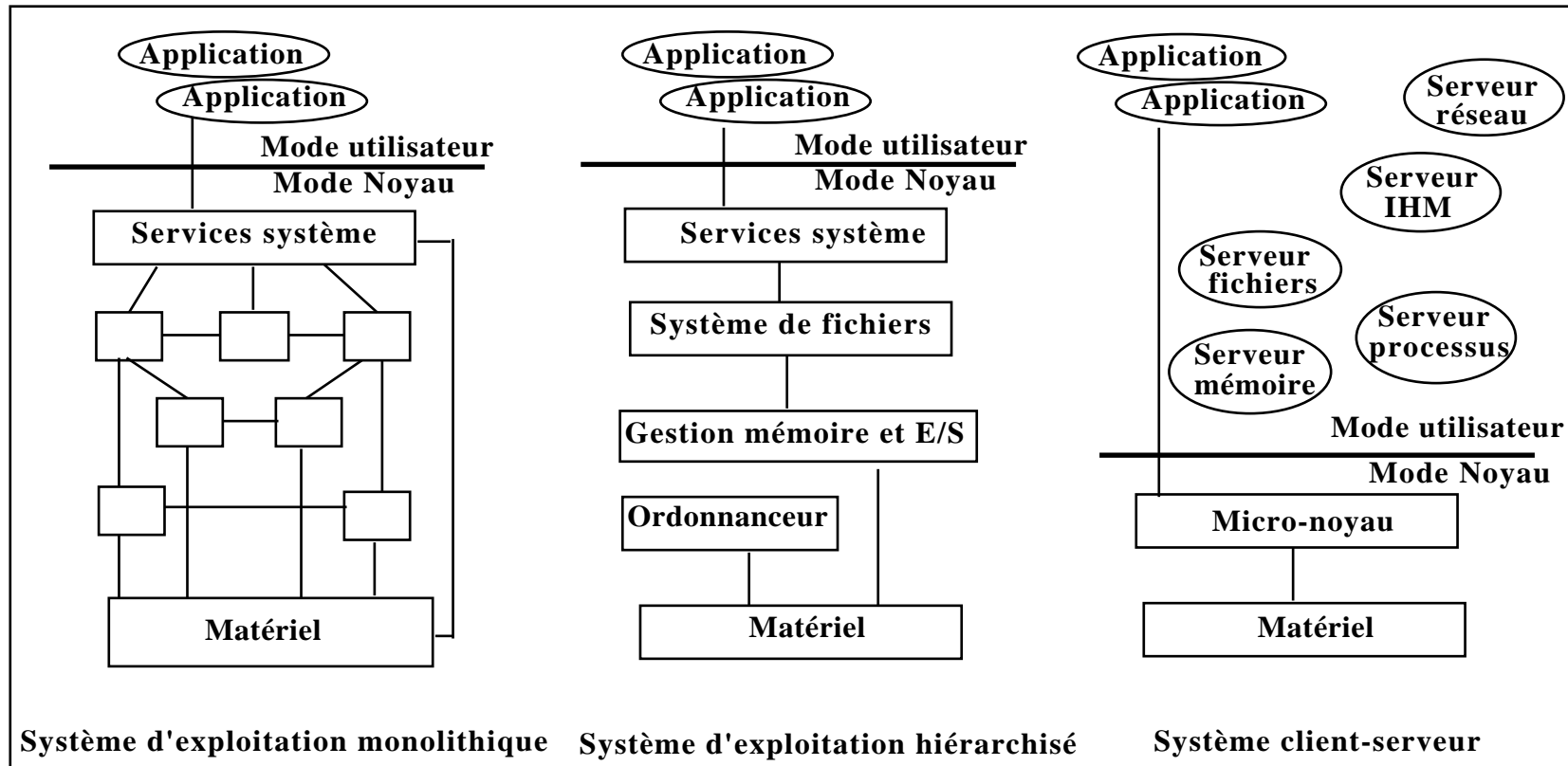
- d'une interface standard (API, fonctionnalités, primitives),
- d'une machine virtuelle ou logique adaptée à son besoin,
- de concepts et d'outils pour exprimer :
  - le parallélisme des activités,
  - la mise en commun de ressources,
  - la communication entre les activités et avec l'extérieur ,
  - le stockage d'objets informatiques,
  - la désignation des objets actifs et passifs

### MULTIPLICITÉ ET INTERPÉNÉTRATION DES ABSTRACTIONS

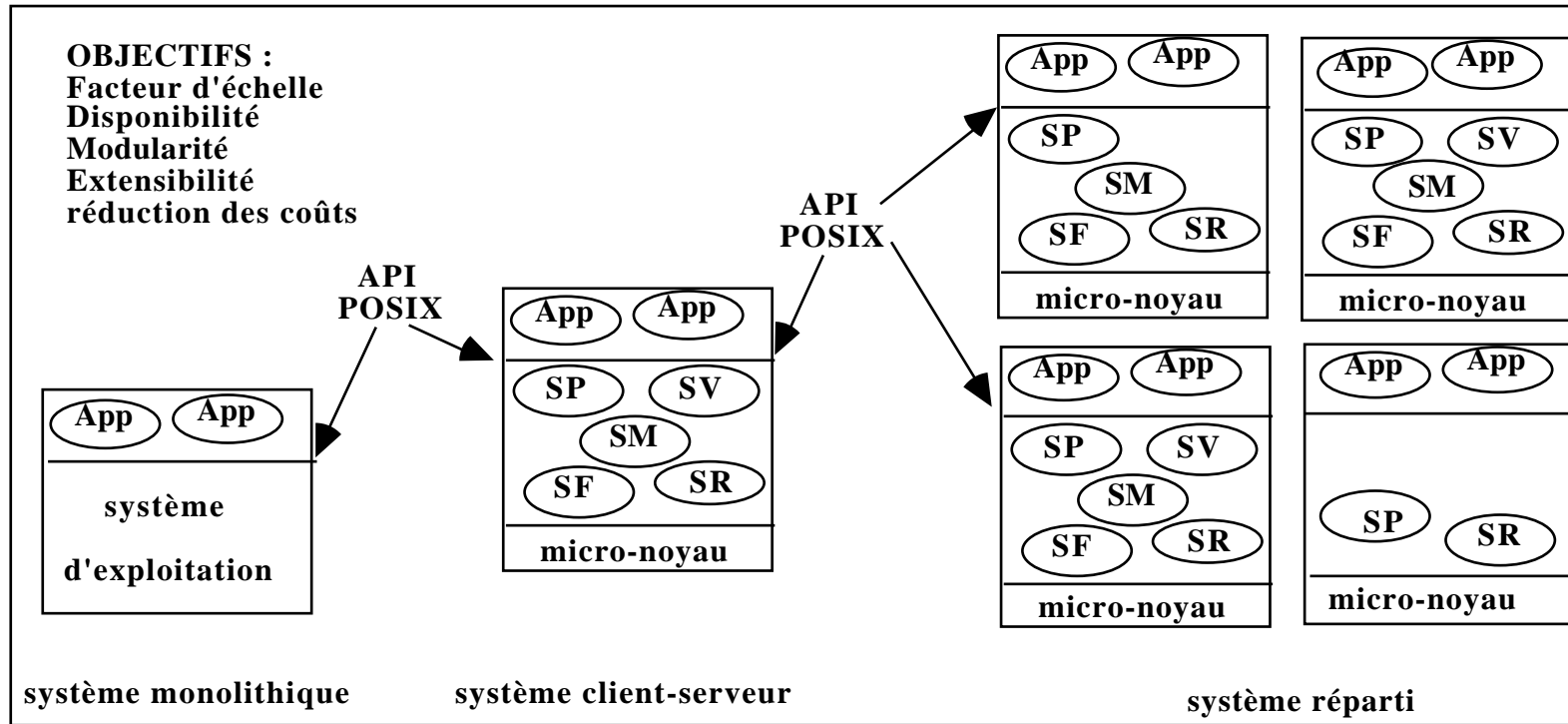
=> HIÉRARCHISATION DES CONCEPTS

=> SYSTÈMES AVEC DES ARCHITECTURES LOGIQUES ET DES MORPHOLOGIES DIVERSES

- concurrence et multiplicité des processus (Unix, Linux)
- allocation et partage des ressources (IBM/VM, VAX/VMS)
- désignation, gestion et protection de l'information (Multics, Intel iAPX 432, IBM AS400)
- temps réel (VxWorks, LynxOs)

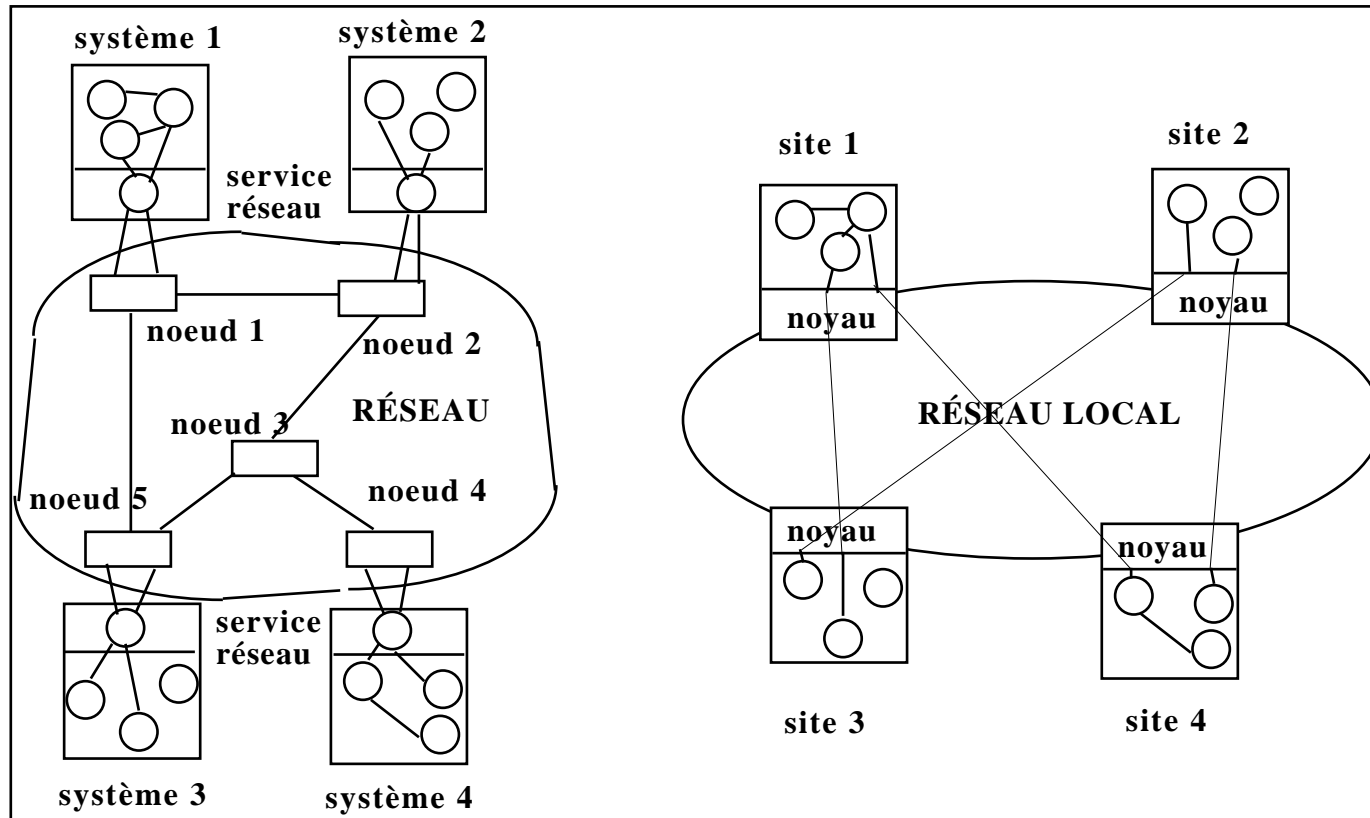


**ÉVOLUTION DES SYSTÈMES D'EXPLOITATION CENTRALISÉS**

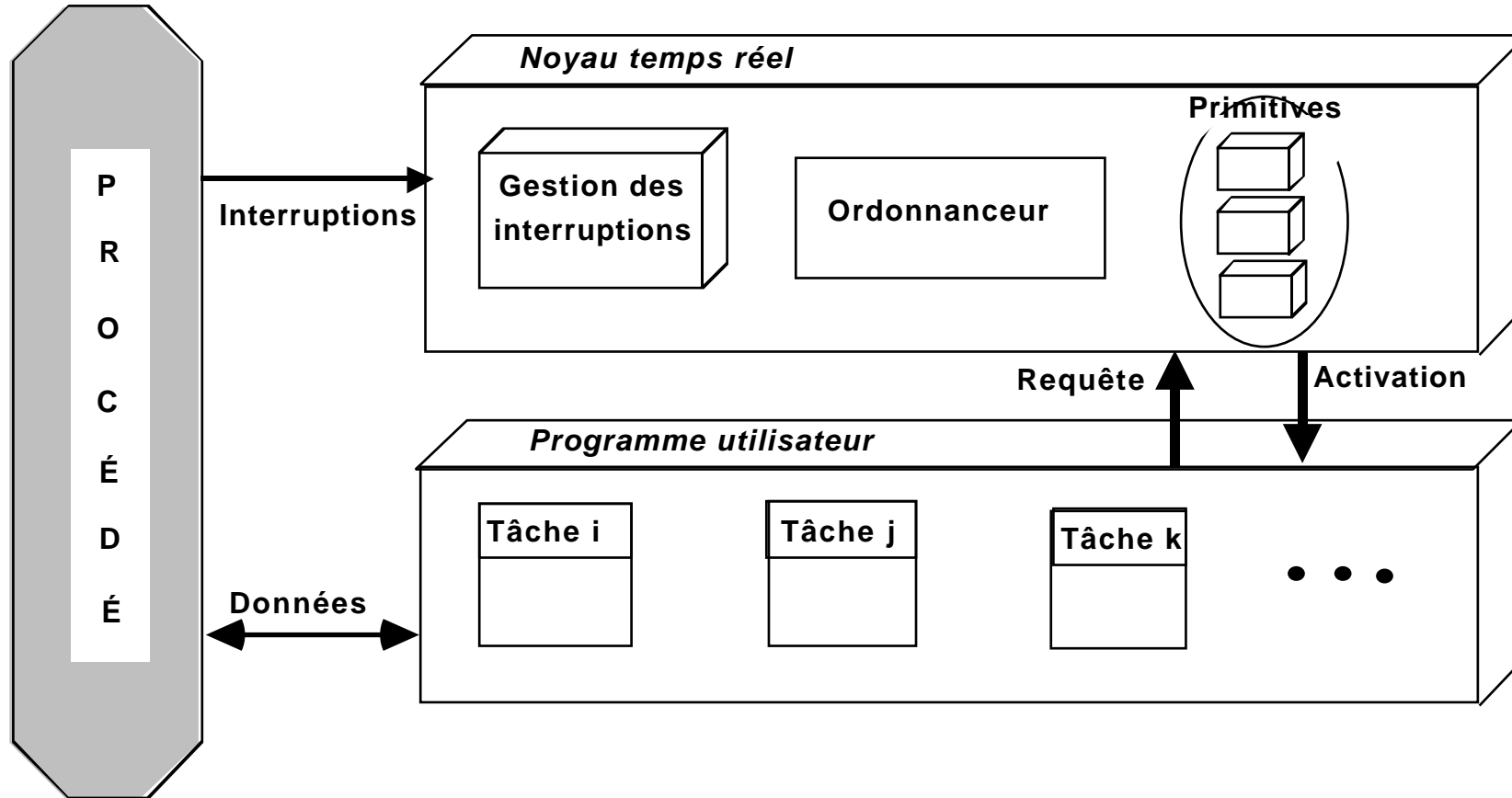


**ÉVOLUTION DES SYSTÈMES CLIENT-SERVEUR**

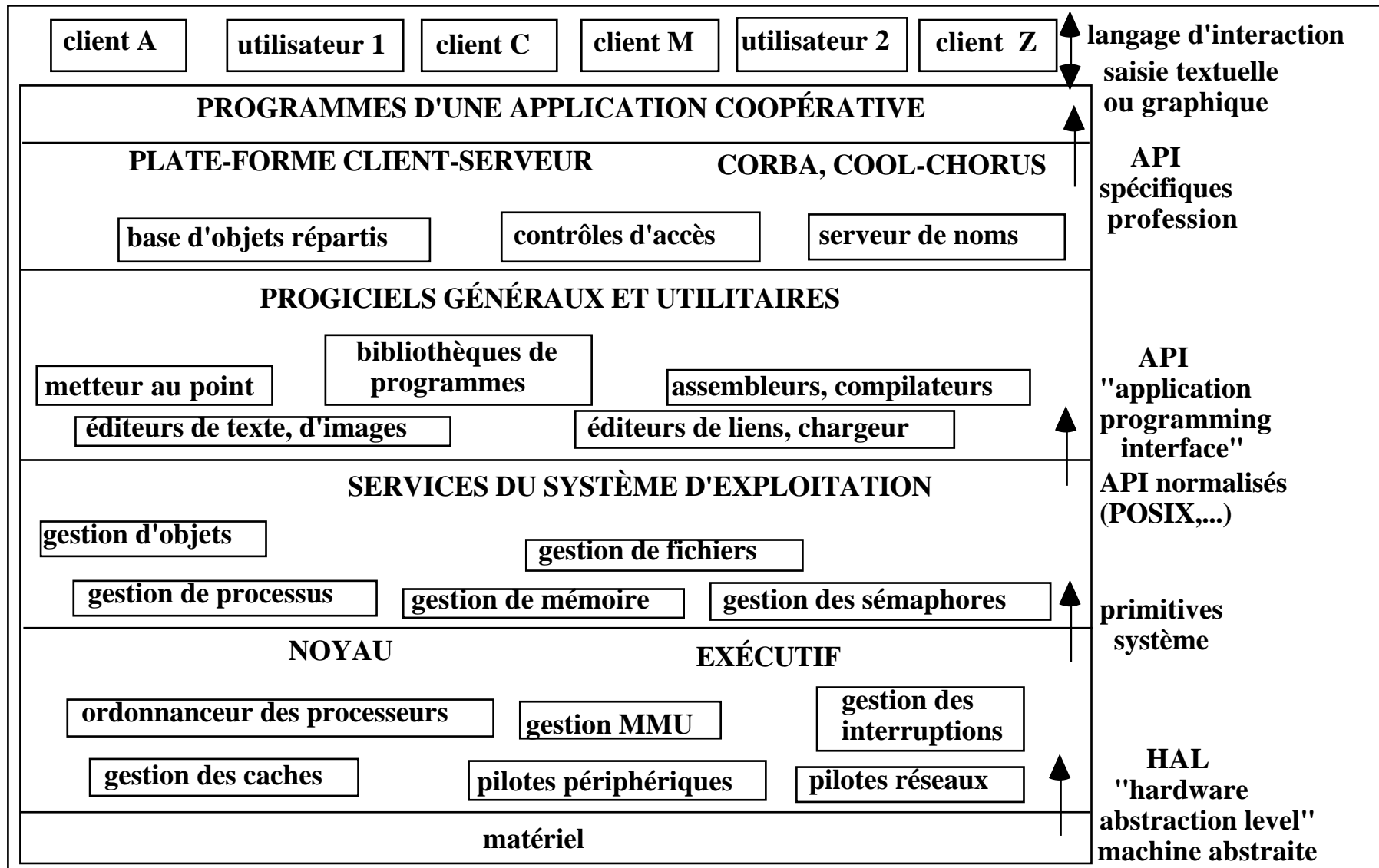


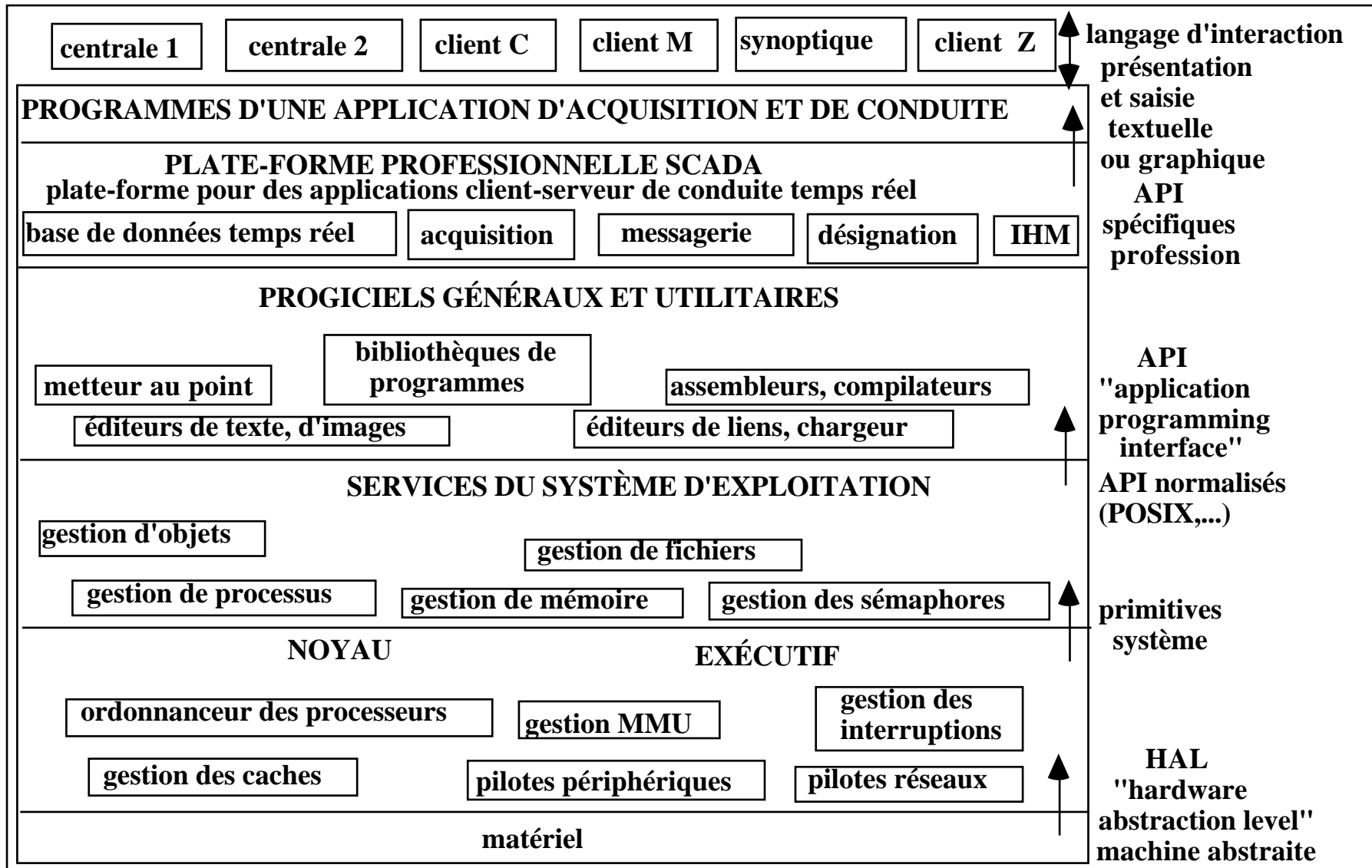


**SYSTÈMES INTERCONNECTÉS <-> SYSTÈMES RÉPARTIS**  
**"network of operating systems" <-> "single system image "**



**SCHEMA D'UN SYSTEME TEMPS RÉEL**





## ALLOCATION DES RESSOURCES AUX PROCESSUS

### RESSOURCE

entité matérielle ou logicielle nécessaire à l'exécution d'un processus

### TYPOLOGIE DES RESSOURCES

- matérielle ou logicielle • accès unique ou concurrent • unitaire ou banalisée • requérable ou non

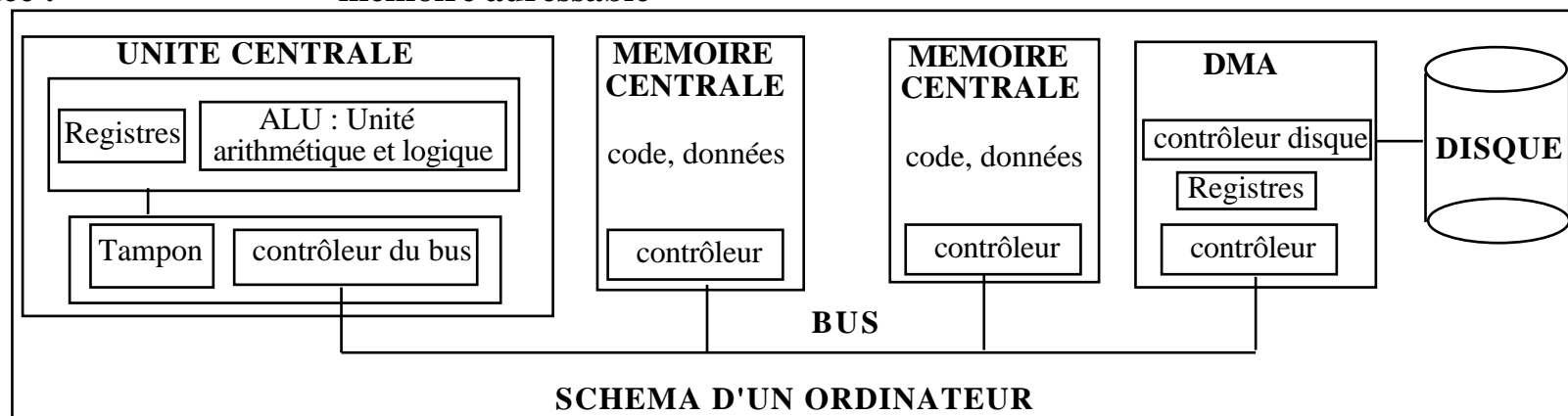
### BESOIN SPÉCIFIQUE DES PROCESSUS INFORMATIQUES (OBJETS ACTIF)

allocation conjointe de processeur et de mémoire

parce que le fonctionnement repose sur le modèle de Von Neumann

1ère idée : programme enregistré en mémoire centrale

2ème idée : mémoire adressable



### STRUCTURATION POUR FACILITER LE PARTAGE ENTRE PROCESSUS

masquer le partage par virtualisation des ressources matérielles (processeur, mémoire, entrées-sorties)

- code réentrant • données sans pointeurs • pile par processus • espace d'adressage logique (virtuel)

## **FONCTIONS DE LA GESTION DES RESSOURCES**

**allocation de ressources aux clients : processus, usagers, groupes,..**

**contrôle des accès : partage, concurrence, exclusion mutuelle, protection,..**

**fiabilité : retour des ressources, non interblocage, équité, traitement des erreurs,..**

## **PROBLÈMES SPÉCIFIQUES AUX SYSTÈMES INFORMATIQUES**

**allocation dynamique et capitalisation des ressources déjà allouées**

**le total des demandes instantanées dépasse le nombre des ressources disponibles d'où refus ou attente**

**les demandes sont imprévisibles : instants, quantité, durée d'utilisation**

## **QUESTIONS POUR UNE RESSOURCE**

**QUAND : à la demande, dans un état fiable, attente des demandeurs ou refus de service**

**DANS QUEL ORDRE : ancienneté, priorités, ordre optimal pour le service**

**COMMENT : toute la demande en une fois, ressource par ressource, ...**

**DURÉE DE L'ALLOCATION : au choix du client, limitée par allocateur,...**

## **FONCTIONS DE L'ALLOCATEUR DE RESSOURCE**

**repérage et suivi des ressources, disponibles et allouées : descripteur de ressource**

**gestion de la file des clients demandeurs : stockage des demandes, discipline de service, durée**

**régulation locale pour performances, fiabilité ou contrôle des accès**

## **STRATÉGIES GLOBALES**

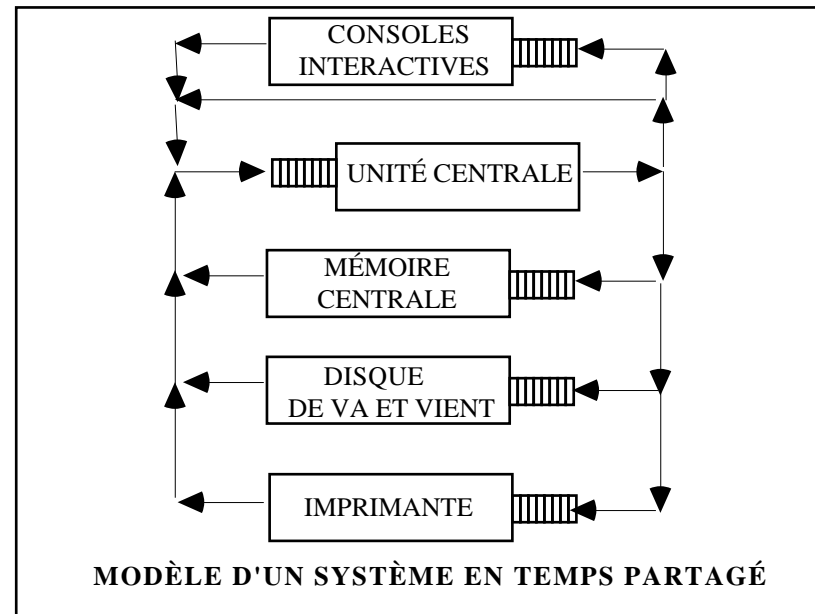
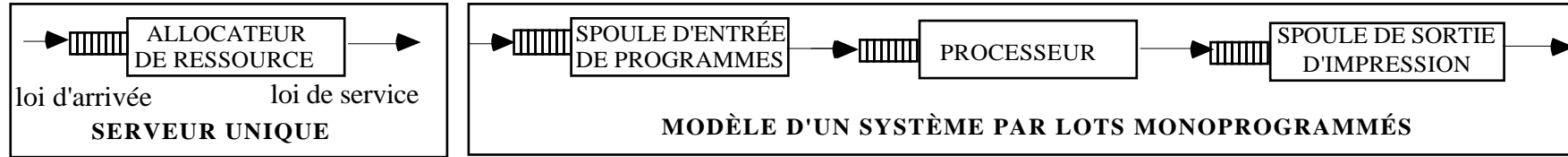
**CRITÈRES DE PERFORMANCES : • réduire le temps de réponse • augmenter le débit global •**

**ordre d'allocation entre ressources pour limiter les tailles des files d'attente**

**politique globale de contrôle de flux de clients et de régulation de la charge**

**FIABILITÉ : traitement de l'interblocage, de l'équité ou des contraintes de temps réel (échéances)**

## MODÉLISATION PAR FILES D'ATTENTE POUR ÉTUDE DES PERFORMANCES



**MODÉLISATION PAR RÉSEAUX DE PETRI POUR LES ÉTUDES DE FIABILITÉ**  
 voir cours de RO ou cours de spécialisation B ("Applications Concurrentes : Conception et Validation")

## **CHARGE D'UN SYSTÈME**

### **DIFFICULTÉ DE LA CONNAISSANCE DE LA CHARGE**

**caractère imprévisible de la demande : instant, quantité, durée**

### **MESURES DE LA CHARGE**

**propriétés statistiques (moyenne, écart type) : lois de probabilité pour le temps, les quantités**

**• charges maximales • charges type ("benchmark")**

### **CONNAISSANCE DE LA CHARGE**

**estimation explicite par annonce**

**évaluation par analyse statique ou dynamique d'un programme (temps réel)**

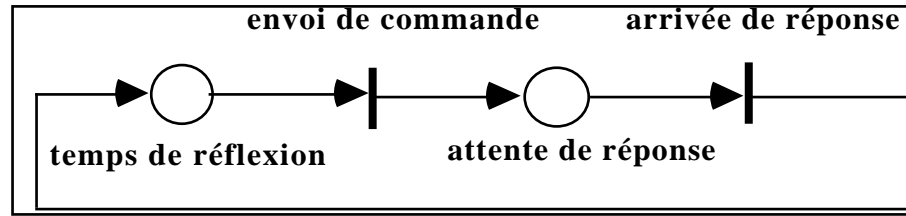
**mesure en fonctionnement**

### **EXEMPLES DE PARAMÈTRES DE CHARGE**

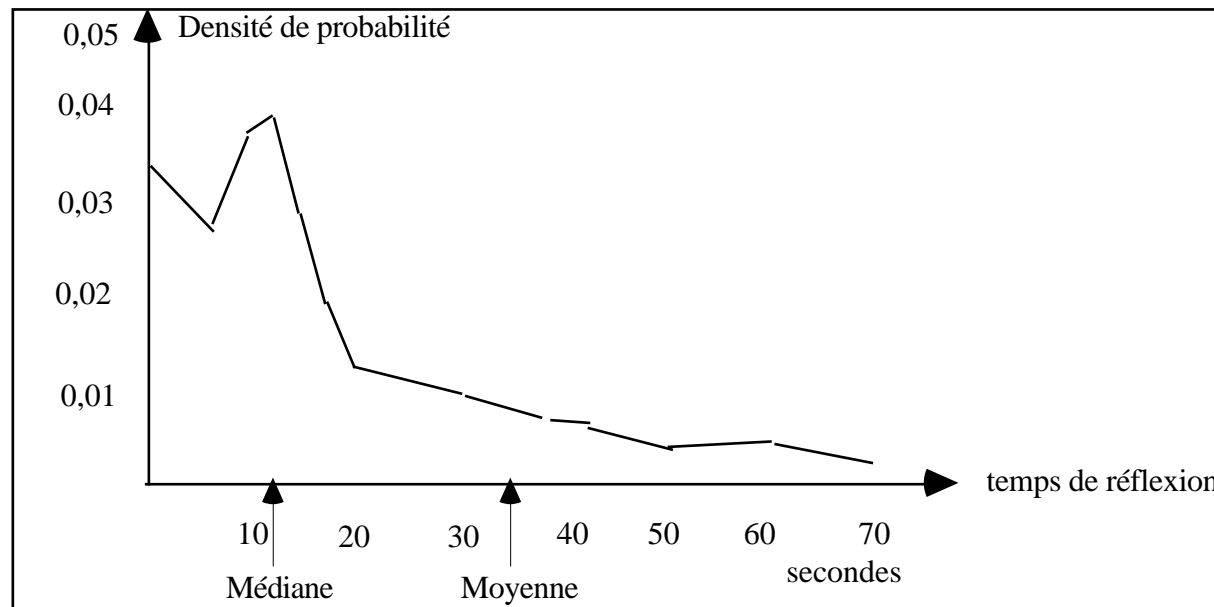
- taille mémoire utilisée • temps d'unité centrale consommé par un processus**
- comportement de l'utilisateur interactif devant sa console**
- taux d'entrées-sorties d'un programme • taux d'appels système**
- espace de travail d'un processus**
- trafic de télétraitement • trafic réseau**



**EXEMPLES DE MESURES ET DE LEUR UTILISATION**



**Comportement de l'utilisateur conversationnel**

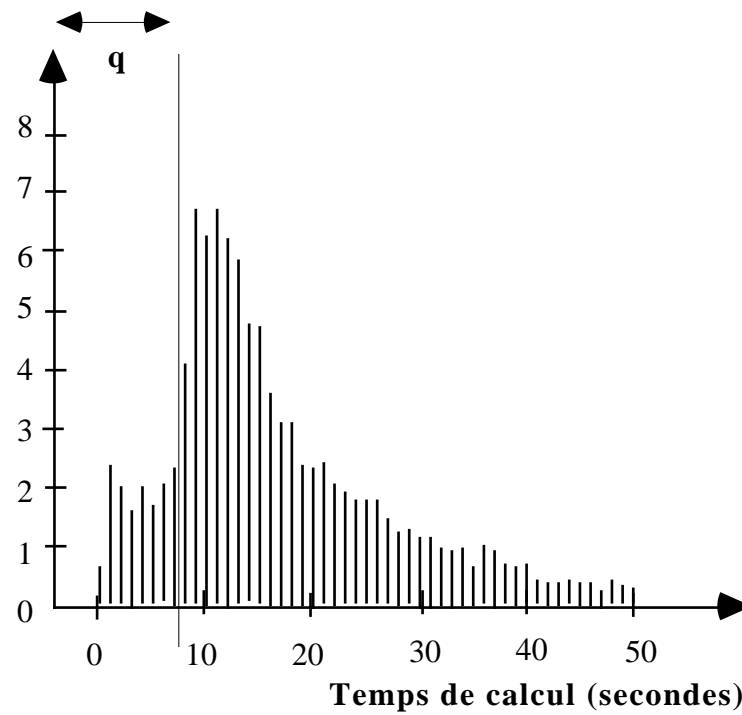
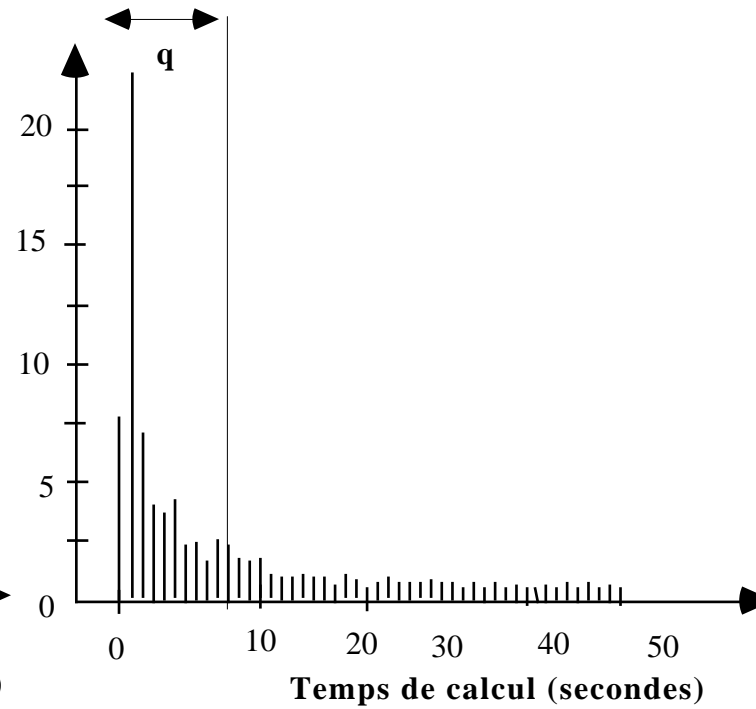


**Distribution des temps de réflexion des utilisateurs du système CTSS (A. Scherr,1965)**

**Temps de réflexion moyen élevé => libérer la mémoire occupée par l'utilisateur en cours de réflexion**

**EXEMPLES DE MESURES ET DE LEUR UTILISATION**

**Mesures des temps de calculs de travaux interactifs et de travaux par lots  
et choix d'un quantum (B. Arden, système MTS, 1969)**

**Pourcentage des travaux****Travaux non interactifs****Pourcentage des travaux****Travaux interactifs**

## EXEMPLES DE MESURES ET DE LEUR UTILISATION

### Expériences de localité pour antémémoire ("cache unité centrale")

PROGRAMMES	A	B	C	D	E
<b>Compilateur FORTRAN</b>	<b>429 000</b>	<b>34</b>	<b>10</b>	<b>29%</b>	<b>90%</b>
<b>Assembleur</b>	<b>386 000</b>	<b>17</b>	<b>5</b>	<b>29%</b>	<b>93%</b>
<b>Éditeur de liens</b>	<b>391 000</b>	<b>27</b>	<b>7</b>	<b>26%</b>	<b>91%</b>
<b>Tri</b>	<b>414 000</b>	<b>48</b>	<b>9</b>	<b>18%</b>	<b>88%</b>
<b>Tracé de courbe</b>	<b>487 000</b>	<b>33</b>	<b>11</b>	<b>33%</b>	<b>83%</b>
<b>Intégration</b>	<b>424 000</b>	<b>11</b>	<b>2</b>	<b>18%</b>	<b>99%</b>
<b>Équations différentielles</b>	<b>471 000</b>	<b>42</b>	<b>2</b>	<b>5%</b>	<b>95%</b>
<b>Compilateur COBOL</b>	<b>386 000</b>	<b>31</b>	<b>10</b>	<b>33%</b>	<b>91%</b>

**A** : nombre moyen de références à la mémoire pendant l'intervalle de 500 millisecondes

**B** : taille du programme en nombre de pages de 4 Koctets

**C** : nombre de pages regroupant 80% des références pendant l'intervalle de 500 millisecondes

**D** : pourcentage du programme sur lequel se fait 80% des références pendant l'intervalle (C/B)

**E** : coefficient de conservation

$$E = \frac{\text{nombre de pages référencées dans l'intervalle I et aussi dans l'intervalle I + 1}}{\text{nombre total de pages référencées dans l'intervalle I}}$$

### Suite des références d'un processus en mémoire virtuelle (Hatfield 1972)

