

Réseaux et Systèmes répartis

Présentation du cours

Mme L Zenaidi- Belkhodja
Département Informatique
Université de Sidi Bel Abbès
lzenaidi7@gmail.com

01/02/2022

1

Réseaux et Systèmes répartis

Cours destiné aux : Masters 1 ISI/WIC/RSSI .

- Volume horaire : 1h 30 Cours et 1h 30 TD et 1h 30 de TP.
- Nombre de crédits :5, Coefficient: 3
- Unité d'Enseignement : UEF22
- Mode d'évaluation : 60 % Examen, 40 % Travail personnel

01/02/2022

2

Réseaux et Systèmes répartis Objectifs

Ce cours s'intéresse aux couches hautes du réseau (Application -> Transport), ses principaux objectifs :

1- S' initier aux Systèmes et Applications Répartis (distribués):

- Définir les systèmes répartis , leurs principes, leurs objectifs;
- Positionner les systèmes répartis parmi les différentes architectures d'ordinateurs existantes;
- Différencier entre les systèmes répartis et les applications réparties;
- Comprendre la mise en œuvre des applications réparties;

2-S'initier à la programmation répartie et à des environnements de développement d'applications réparties .

01/02/2022

3

Systèmes distribué = Systèmes réparti

- Certains auteurs font la distinction entre les termes :
- **Système distribué:** induirait l'idée d'une **distribution** des tâches effectuée par un coordinateur (site central). Terme utilisé par les **fabricants d'ordinateurs**.
- **Système réparti:** supposerait une **coopération** des tâches, en vue de la **répartition** du travail. Terme utilisé **en génie logiciel**.
- La littérature anglaise ne fait pas de différence, et parle de «distributed systems », où le terme distributed signifie «réparti dans l'espace».
- Dans ce cours, les deux termes seront considérés comme équivalents .

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

4

Réseaux et Systèmes répartis

Plan du cours

- 1-Introduction aux Systèmes et applications répartis.
- 2- Programmation réseau avec les sockets java.
- 3-Appel de procédures distantes (RPC : Remote Procedure Call).
- 4-Invocation de méthodes distantes (Java RMI : Remote method Invocation).
- 5-Corba: Common Object Request Broker Architecture.

01/02/2022

5



Réseaux et Systèmes répartis

Chapitre 1

Introduction aux Systèmes et Applications Répartis (distribués)

Mme L Zenaidi-Belkhodja
Département Informatique
Université de Sidi Bel Abbès
lzenaidi7@gmail.com

01/02/2022

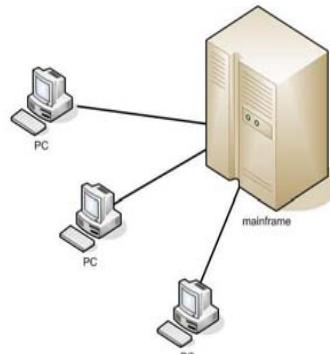
chap 1 :Introduction aux systèmes et applications répartis

6

1.1- Introduction aux systèmes répartis

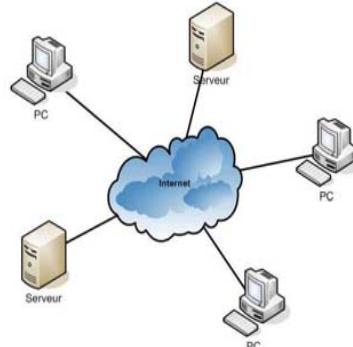
Historique des organisations informatiques

- 1- Serveur centralisé avec terminaux
- 2-Ordinateurs personnels
- 3-Systèmes répartis



Système centralisé

VS



Système réparti

01/02/2022

7

chap 1 :Introduction aux systèmes et applications répartis

1.1- Introduction aux systèmes répartis

Evolution technologique

-baisse des prix et augmentation de la performance des **microprocesseurs** et des **voies de télécommunication** (débit, fiabilité).

-banalisation des réseaux de télécommunication.

-convergence informatique et téléphonie.

Evolution des besoins

-des entreprises et des organisations : **communication et partage** (ex Intranet)

-accès universel à l'information (ex Web)

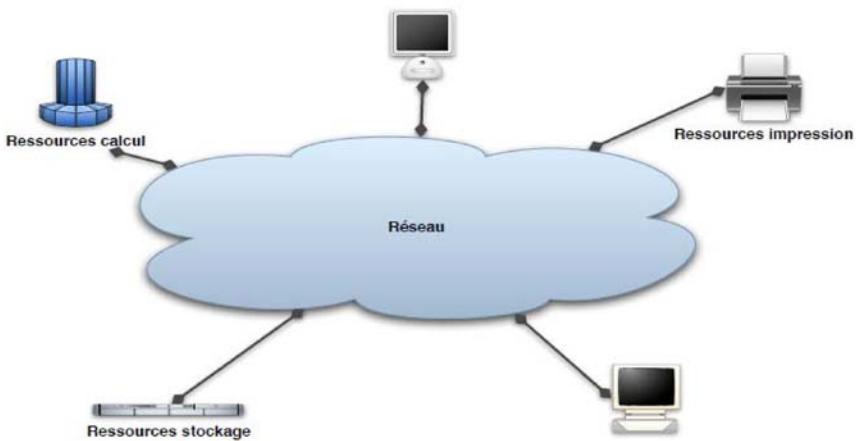
-informatique distribuée "grand public" (ex vidéo interactive)

Systèmes et Applications répartis (distribués)

01/02/2022

8

Structure d'un système réparti



Un **système réparti** est formé de composants matériels ou logiciels localisés sur des **ordinateurs en réseau** qui communiquent et coordonnent leurs actions uniquement par l'envoi de messages, en vue de répondre à un besoin donné.

L'enseignement de l'informatique
Systèmes et applications répartis

9

1.1- Introduction aux systèmes répartis

Système réparti vs Application répartie

Un **système réparti** est un ensemble de **ressources interconnectées**.

Il doit régler des problèmes :

- Dtribution et de partage de ressources.
- De communication et de synchronisation entre processus s'exécutant sur les différentes machines.
- De répartition des calculs,...

Une **application répartie** est constituée d'**entités logicielles interconnectées**, qui doivent répondre à un problème spécifique et qui peuvent être:

- Exécutées dans des **environnements d'exécution (pile+tas) séparés**
⇒ pas le même espace mémoire.
- Développées dans différents langages de programmation.
- Situées sur des plateformes distinctes **connectées via un réseau**.

Systèmes et applications répartis

11

1.1- Introduction aux systèmes répartis

Système réparti vs Application répartie

Système : gestion des ressources communes et de l'infrastructure liée de manière étroite au matériel sous-jacent.

Cache la complexité du matériel et des communications, fournit des services communs de plus haut niveau.
Ex : Système d'exploitation (centralisé/ réparti (distribué), système de communication.

Application : réponse à un problème spécifique, fourniture de services à des utilisateurs (qui peuvent être d'autres applications).

Utilise les services généraux fournis par un système (d'exploitation réseau,...).

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

10

1.1- Introduction aux systèmes répartis

Exemples d'utilisation de systèmes/applications répartis

- Internet et ses différentes applications.
- Le **World Wide Web**.
- Courrier électronique,
- Messagerie instantanée,
- Partage de fichiers en pair à pair,
- DNS: service distribué pour traduire les noms de domaine internet en adresse IP,
- Commerce par internet,
- Consultation de compte bancaire,
- Calendrier partagé,
- Radio/télévision numérique interactive (programme à la demande, télé-achat,...)
- Téléconférence,
- Moteur de recherche...

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

12

1.1- Introduction aux systèmes répartis

Définitions

Définition [Tanenbaum]: Un ensemble d'ordinateurs *indépendants* qui apparaît à un utilisateur comme un système *unique et cohérent*:

- ✓ Les machines sont **autonomes**.
- ✓ Les utilisateurs ont l'impression d'utiliser **un seul système**.

Définition [Lamport]: Un système réparti est un système qui vous empêche de travailler quand une machine dont vous n'avez jamais entendu parler tombe en panne.

Définition [Coulouris]: Un système réparti est un ensemble de machines *autonomes connectées par un réseau*, et équipées d'un logiciel dédié à la *coordination* des activités du système ainsi qu'au *partage* de ses ressources.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

13

1.1- Introduction aux systèmes répartis

Définition [Flynn]

Taxonomie de Flynn (1972)

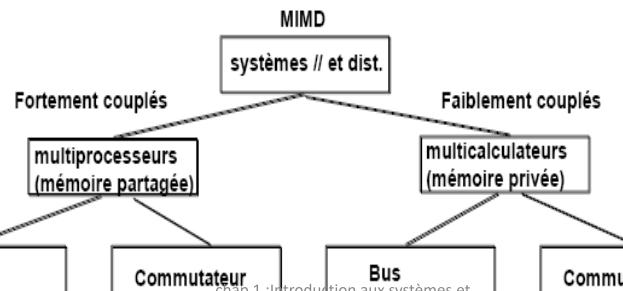
- nombre des flux d'instructions
- nombre des flux de données

SISD : un seul flux d'instruction et un seul flux de données (ordinateurs centralisés)

SIMD : un seul flux d'instruction et de multiples flux de données (machines //, vectorielles)

MISD : multiples flux d'instruction et un seul flux de données (pas de machine réelle)

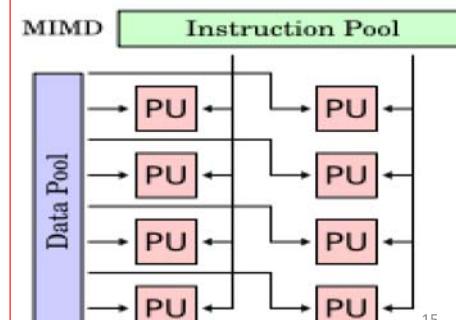
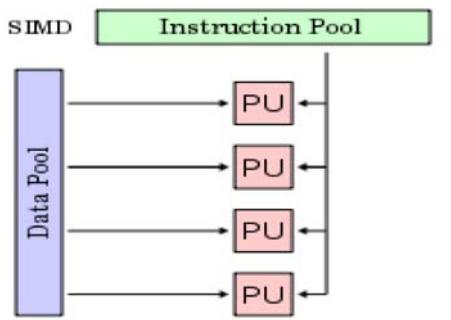
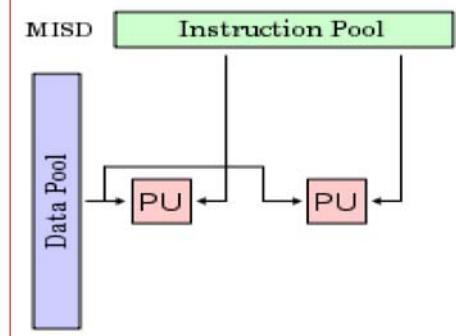
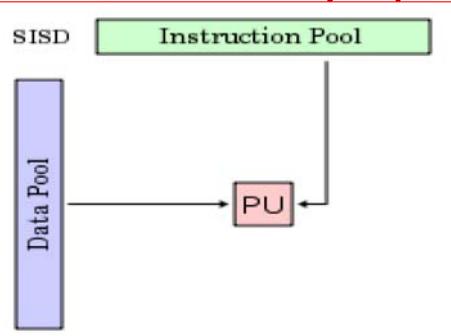
MIMD : multiples flux d'instruction et de multiples flux de données



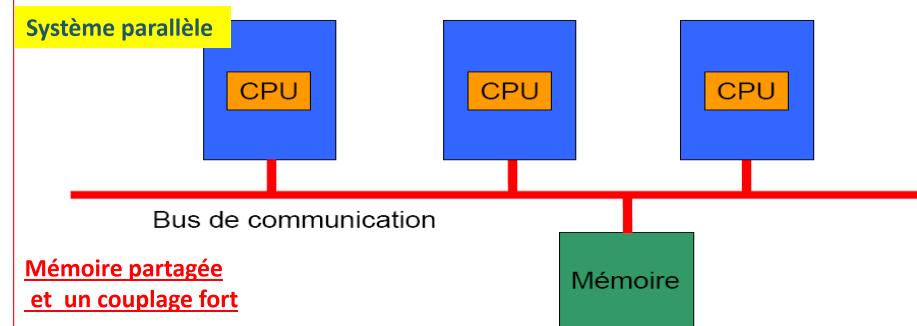
01/02/2022 chap 1 :Introduction aux systèmes et applications répartis

14

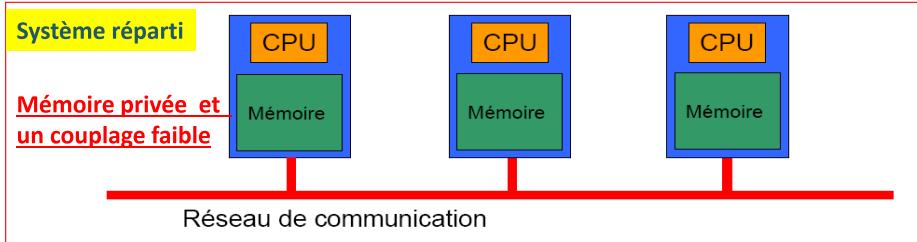
Classification de Flynn pour l'architecture des ordinateurs



Architecture MIMD à Mémoire partagée/ à Mémoire privée [Flynn]



Système parallèle
Mémoire partagée et un couplage fort



Système réparti
Mémoire privée et un couplage faible

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

16

1.1- Introduction aux systèmes répartis couplage fort/couplage faible

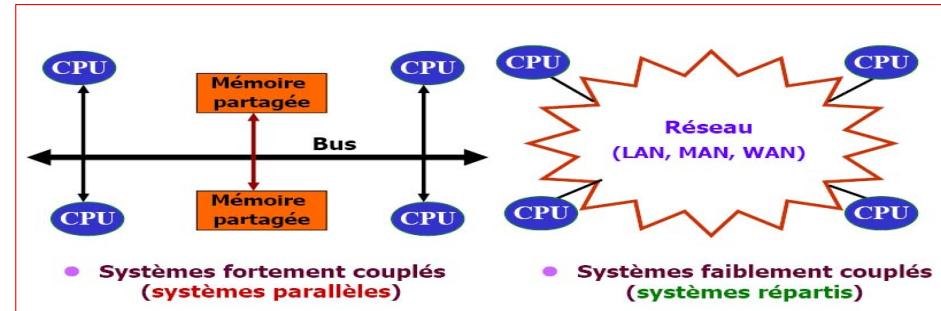
- Le **couplage** est une métrique indiquant le niveau d'interaction entre deux ou plusieurs composants logiciels (fonctions, modules, objets ou applications).
- Deux composants sont dits couplés s'ils échangent de l'information. On parle de **couplage fort** si les composants échangent beaucoup d'information. On parle de **couplage faible**, si les composants échangent peu d'information.
- Dans un système à **couplage fort**:
→ Le délai de transmission d'un message est court;
→ Le débit de transmission est élevé.
- Dans un système à **couplage faible** :
→ Le délai de transmission d'un message long;
→ Le débit de transmission est faible.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

17

Système Parallèle vs Système réparti (distribué)



- Système Parallèle:** Une machine multiprocesseurs avec un environnement du type SIMD où tous les processeurs exécutent le même programme et ont une vision uniforme de l'état global du système. C'est un réseau de machines asynchrones fortement couplées.
- Système Réparti :** processus indépendants sur des machines distinctes et communiquant par échange de messages, en général, asynchrones et faiblement couplés.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

18

1.1- Introduction aux systèmes répartis Système Parallèle vs Système réparti (distribué)

- Dans un système parallèle, Le rapport **temps de calcul** sur **temps de communication** est une caractéristique importante d'un programme parallèle, orientant la façon de le paralléliser.
- Dans un système réparti (distribué), le temps de calcul est souvent négligeable comparativement au temps de communication.
- Un système d'exploitation parallèle contrôle l'exécution de programmes sur une machine parallèle (multiprocesseurs).
- Un système d'exploitation de réseaux fournit une plateforme de machines reliées par un réseau, chacune exécutant son propre système d'exploitation.
- Un système d'exploitation réparti fournit et contrôle l'accès aux ressources du système réparti.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

19

1.1- Introduction aux systèmes répartis Système Parallèle vs Système Réparti (distribué)

Système	Non distribué	Distribué
Non Parallèle	Machine séquentielle (un seul processeur).	Réseau d'ordinateurs avec communication lente (Internet).
Parallèle	-Machines vectorielles, - Machines à mémoires partagées, - Machines SIMD.	Réseau haut débit.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

20

1.1- Introduction aux systèmes répartis

Systèmes répartis/ parallèles/ réseaux

Systèmes parallèles	Systèmes répartis	Systèmes d'exploitation de réseaux
Processeurs	Sites	Ressources
Homogènes	Hétérogènes	Hétérogènes
Partage ou non de mémoire	Mémoires individuelles	Mémoires individuelles
Couplage fort	Couplage faible	Couplage faible
Topologie du réseau d'interconnexion	Réseau LAN + WAN	Réseau LAN
Les users sont au courant de la multiplicité des Processeurs	Les users ont l'impression d'être dans un système centralisé	Les users sont au courant de la multiplicité des Machines

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

21

1.1- Introduction aux systèmes répartis

Définition (synthèse)

- Un système réparti (distribué) est donc un système généralement, faiblement couplé, composé d'un ensemble de processeurs ne partageant pas de mémoire ni d'horloge. Chaque processeur possède sa propre mémoire locale, les processeurs coordonnent leurs actions seulement par échange de messages et communiquent entre eux par divers moyens, comme des bus rapides ou des lignes téléphoniques.
- Les processeurs d'un système réparti varient en taille et en fonction. Ils peuvent inclure des petits microprocesseurs, des stations de travail, des mini-ordinateurs ou des grands systèmes informatiques. Ces processeurs sont appelés sites, nœuds, ordinateurs, machines, hôtes, selon le contexte dans lequel ils sont utilisés.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

22

Apports des systèmes répartis

- Coût: meilleur rapport Performance/Prix que les gros ordinateurs;
- Puissance de calcul et de stockage;
- Performance (accélération) grâce au calcul parallèle;
- Adaptation: à des classes d'applications réelles naturellement distribuées;
- Fiabilité: résistance aux pannes matérielles et logicielles;
- Extensibilité: croissance progressive selon les besoins.
- Partage de données et des périphériques;
- Communication;
- Souplesse: exécuter un travail sur la machine la plus disponible;

Inconvénients des systèmes répartis

- Absence de vision globale: difficulté de réalisation et d'administration;
- Problèmes relatifs aux réseaux: saturation, délais, perte de messages, sécurité.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

23

Propriétés d'un système réparti

Ce que doit offrir un système réparti

- 1- Transparence
- 2- Cohérence et synchronisation
- 3- Désignation des ressources du système
- 4- Hétérogénéité
- 5- Sécurité
- 6- Performance
- 7- Passage à l'échelle
- 8-Communication

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

24

Propriétés d'un système réparti

Ce que doit offrir un système réparti

1- Transparence

1. **Transparence à la localisation:** l'utilisateur ignore la situation géographique des ressources. Ce qui nécessite un service de nommage global (annuaire)⇒
2. **Transparence à la migration.**
3. **Transparence d'accès.** L'utilisateur accède à une ressource locale ou distante d'une façon identique.
4. **Transparence à l'hétérogénéité.** L'utilisateur n'a pas à se soucier des différences matérielles ou logicielles des ressources qu'il utilise.
5. **Transparence aux pannes** (réseaux, machines, logiciels). Les pannes et réincarnations sont cachées à l'utilisateur ⇒
6. **Transparence à la réPLICATION.**
7. **Transparence à l'extension des ressources:** extension ou réduction du système sans occasionner de gène pour l'utilisateur.
8. **Transparence à la Concurrence :** Plusieurs utilisateurs peuvent partager en même temps les mêmes ressources.
9. **Transparence au Parallélisme :** Des tâches peuvent s'exécuter en parallèle sans que les utilisateurs le sachent.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

25

Propriétés d'un système réparti

Ce que doit offrir un système réparti

- 4- **Hétérogénéité:** des données, des langages, des systèmes d'exploitation, des protocoles => **Interopérabilité**.
- 5- **Sécurité :** Accès aux données confidentielles. Authentification.
- 6- **Performance:** nécessite de faire au moins aussi bien qu'un système centralisé.
- 7- **Passage à l'échelle :** la faculté à pouvoir s'adapter au rythme de la demande en particulier et à maintenir ses fonctionnalités et ses performances en cas de forte demande.
- 8-**Communication:** par message, par RPC,..

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

27

Propriétés d'un système réparti

Ce que doit offrir un système réparti

2- Cohérence et synchronisation

- ✓ Un système répartie => plusieurs horloges non synchronisées + communications **asynchrones** ⇒ **pas d'état global**
- ✓ **Problèmes :** partage de données réparties et coordination répartie.
- ✓ **Solutions :** algorithmes dédiés d'élection, d'exclusion mutuelle, de consensus, de terminaison, etc.

3- Désignation des ressources du système

- ✓ Désigner un objet consiste à lui affecter un nom permettant de lui faire référence.
- ✓ Le nom doit être indépendant de la localisation géographique (pannes, migrations) et du temps.

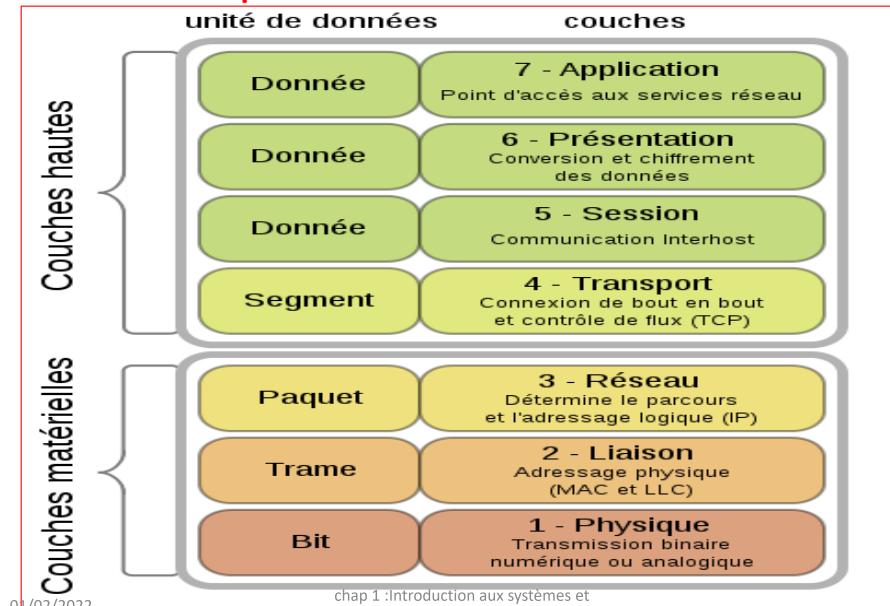
01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

26

1.2 Notion d'application répartie

Rappel sur le modèle OSI (*Open Systems Interconnection*) pour la communication en réseau



01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

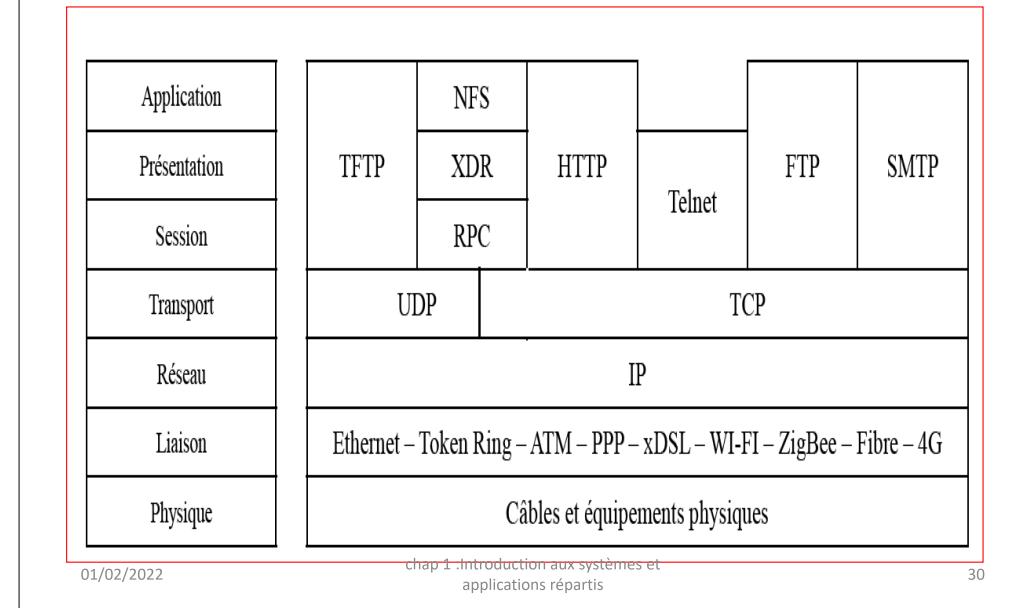
28

- La couche **application** : responsable de l'application traitée (message, transfert de fichier, ...) , utilise des protocoles applicatifs (HTTP, FTP, SMTP ...)
- La couche **présentation** : indépendance par rapport aux codages et représentations de données (conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises).
- La couche **session**: synchronisation et gestion du dialogue entre applications (gestion des sessions).
- La couche **transport**: transport d'information de bout en bout. Elle effectue un contrôle de flux. Notion de port (TCP et UDP).
- La couche **réseau**: établissement, maintien et libération des connexions, ainsi que l'adressage logique (adresse IP).
- La couche **liaison** :gestion d'accès au support physique, assure que les données envoyées sur le support physique sont bien reçues par le destinataire.
- La couche **physique**: transmission des données binaires sur un support physique.

29

1.2 Notion d'application répartie

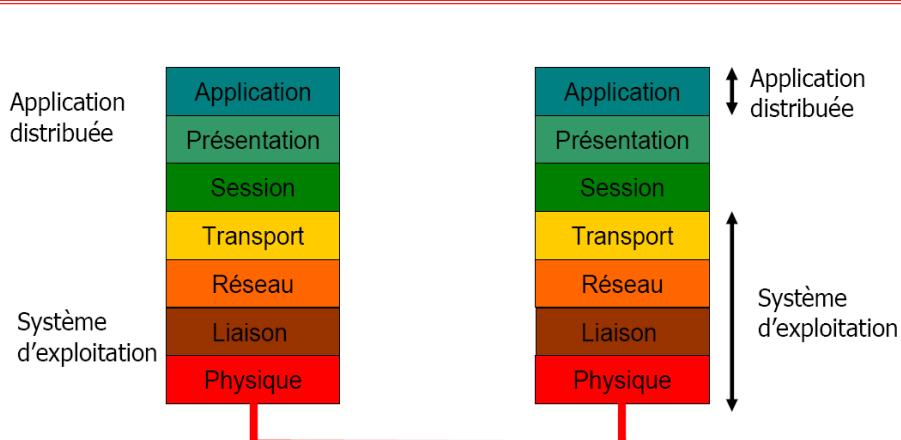
Modèle OSI et protocoles utilisés



1.2 Notion d'application répartie

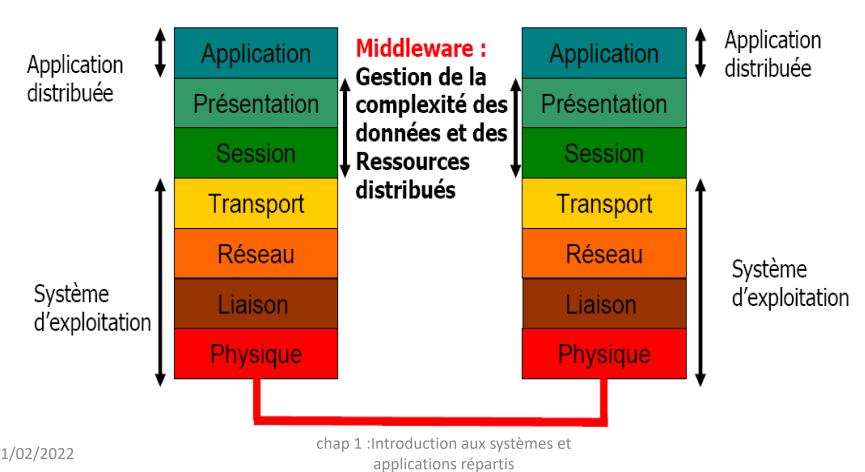
C'est un ensemble de programmes distribués sur un réseau de communication, qui collaborent pour assurer un service.

Une application informatique est dite **répartie (distribuée)** lorsqu'elle met en jeu des parties qui s'exécutent sur plusieurs machines reliées par un système de communication.



1.2 Notion d'application répartie

- Les interfaces (API) fournies par les systèmes d'exploitation réseaux sont complexes pour gérer la distribution des ressources et l'hétérogénéité.
- Solution : Introduire une couche logicielle intermédiaire entre les niveaux bas(système de communication) et le niveau haut(application).



1.3 Notion de middleware(Intergiciel)

- Le terme middleware est apparu vers 1990 tandis que le terme intergiciel est apparu autour de 1999-2000.
- Le middleware (intergiciel) est Couche logicielle** intermédiaire entre les applications et le réseau, permettant le dialogue entre des applications hétérogènes.
- Fournit un modèle de programmation simplifiant la construction des composantes des logiciels travaillant dans un système distribué.
- Offre des services de haut niveau liés aux besoins de communication des applications (temps réel, sécurisation, sérialisation, transaction informatique, etc.).

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

33

1.3 Notion de middleware(Intergiciel)

applications réparties

application

intergiciel (middleware)

système d'exploitation

machine

application

intergiciel (middleware)

système d'exploitation

machine

support physique de communication

réseaux

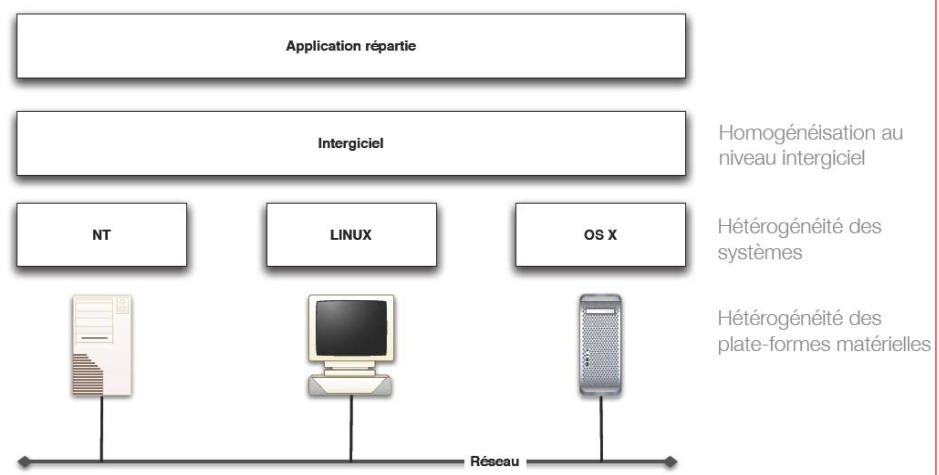
Introduction de l'intergiciel comme intermédiaire entre les applications et le réseau

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

34

1.3 Notion de middleware(Intergiciel)



Introduction de l'intergiciel pour masquer l'hétérogénéité des systèmes et des plateformes

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

35

Fonctions d'un intergiciel

- 1-Fournir une interface simple (API de haut niveau) aux applications et aux développeurs,
- 2-Masquer l'hétérogénéité des systèmes sous-jacents,
- 3-Rendre la répartition invisible (« transparente »),
- 4-Fournir des services répartis d'usage courant.

Deux types de middleware existent:

1. **Middleware explicite**: le développeur est chargé des appels transversaux (sécurité, transactions,...).
2. **Middleware implicite**: possède un contexte déclaratif. C'est le container de l'objet qui se charge de tout. **Un conteneur est un composant logiciel système qui contrôle d'autres composants, dits métier.** Tomcat est un exemple de conteneur.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

36

Séparation entre Code Métier/Code Middleware dans un middleware

- On distingue deux types de code :
 - Code Métier
 - Code Intergiciel (Middleware)
- Le code métier ou applicatif est lié à la logique de l'application développée: code de gestion d'un compte bancaire.
- Le code non applicatif ou middleware est indépendant de l'application: code de gestion de la sécurité, code de gestion de la persistance.
- Le mélange des deux rend difficile le développement et la mise au point des logiciels. Il faut donc les séparer complètement et rendre transparente la partie non applicative.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

37

Séparation entre Code Métier/Code Middleware dans un middleware

- Le code middleware devient accessible sous forme de services : Service de persistance, Service de transaction,...
- Un service est défini par une API qui fournit toutes les fonctions nécessaires aux applications: start/commit transaction, store/load data...
- La relation entre le code métier et son contexte d'exécution (configuration des services) peut se faire de manière déclarative.

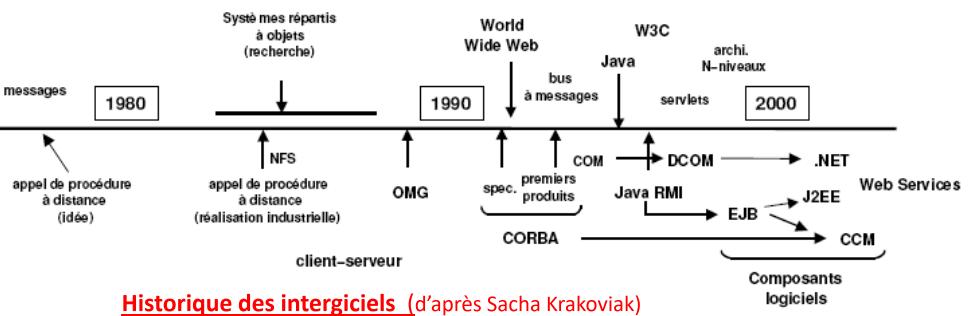
01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

38

Quelques catégories d'intergiciels

- A messages (à partir 1970): MOM: Message Oriented Middleware IBM MQSeries, Microsoft Message Queues Server, DECmessageQ, JMS...
- De bases de données (ODBC)
- A appel de procédure distante à partir 1984
- A objets répartis (CORBA 1991, JAVA RMI 1996)
- A composants (EJB 1997 , DCOM 1997(Distributed Component Object Model), COM+ 1999, CCM 2002)
- Orientés services (Services web 2004)



Avantages et Inconvénients des intergiciels

- + Cacher les détails des mécanismes de bas niveau,
- + Assurer l'indépendance vis-à-vis des langages et des plates-formes,
- + Permettre de réutiliser l'expérience et parfois le code,
- + Faciliter l'évolution des applications,
- + Réduire le coût et la durée de développement des applications.
- Perte de performances liée à la traversée de couches supplémentaires de logiciel,
- Prévoir la formation des équipes de développement.

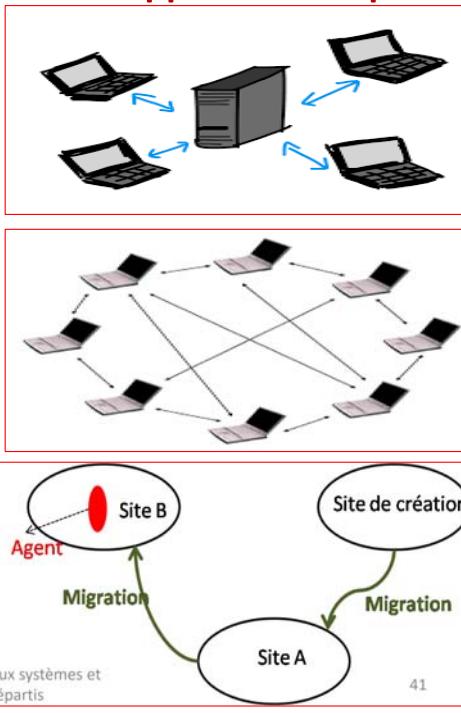
01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

40

1.4 Modèles pour la construction des applications réparties

- **Le modèle Client / Serveur.**
- **Pair à Paire:** chaque client est aussi un serveur.
- **Agent mobile ou code mobile:** une entité autonome capable de communiquer, disposant d'une connaissance, d'une capacité d'exécution propre. Pouvant se déplacer d'un site à un autre en cours d'exécution pour accéder à des données ou à des ressources. Se déplace avec son code et ses données propres, et son état d'exécution.



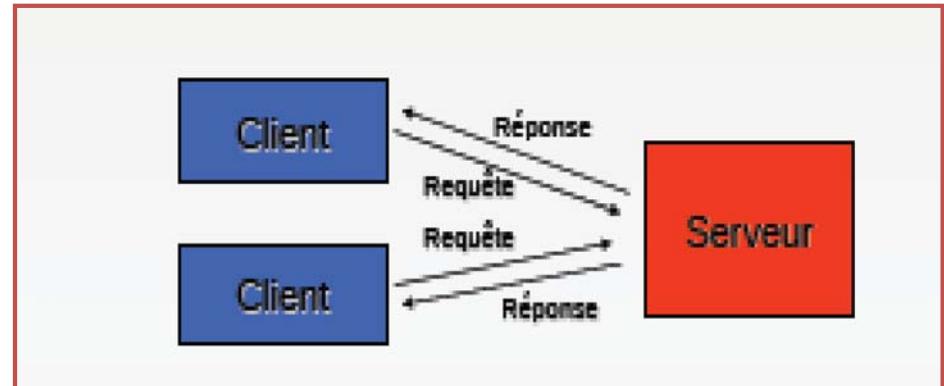
01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

41

1.4.1 Architecture Client/Serveur

Modèle **simple** et **le plus utilisé** pour le développement d'applications réparties, basé sur la **communication par messages** (plutôt que par partage de données, mémoire ou fichiers).



01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

42

1.4.1 Architecture Client/Serveur

- Effectue une distinction stricte entre les **rôles du client** d'une part et du **serveur** d'autre part.
- Peut être interprétée d'un **point de vue matériel** (des machines clientes et des machines serveurs) ou **logiciel** (entités clientes et entités serveurs d'une même application)
- Les interprétations matérielles et logicielles sont souvent liées: **une application client-serveur est communément répartie sur plusieurs machines connectées par un réseau.**
- Un client effectue une requête pour un service précis sur un serveur donné et reçoit une réponse.
- Un serveur reçoit une demande de service, la traite, et renvoie la réponse au client.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

43

1.4.1 Architecture Client/Serveur

Plusieurs **paradigmes de communication** sont possibles lors de l'implémentation de protocoles client-serveur, par exemple :

- Passage de messages synchrone.
- Passage de messages asynchrone.
- Appel de procédures à distance (RPC).
- Invocation de méthodes à distance.
- Interaction événementielle (publication d'évènements).
- Interaction par messagerie (boîte aux lettres).

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

44

Points forts de l'architecture Client/serveur

- séparation entre l'interface du service et sa réalisation,
- Couplage assez faible (le serveur n'a pas besoin des clients, **interaction faible**, peu d'informations échangées),
- Maintenance et administration du serveur assez simples,
- Souplesse : possibilité d'ajouter/supprimer des clients sans perturber le fonctionnement de l'application,
- Les ressources étant centralisées sur le serveur, d'où:
 - Sécurisation simple des données,
 - Pas de problème d'intégrité/cohérence des données.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

45

Points faibles de l'architecture Client/serveur

- Un maillon faible : le serveur
- Coût élevé : le serveur doit être très performant pour honorer tous les clients
- Apparition d'architectures de plus en plus complexes : commerce en ligne, Radio/télévision numérique interactive, moteur de recherche...

Solution ?

01/02/2022

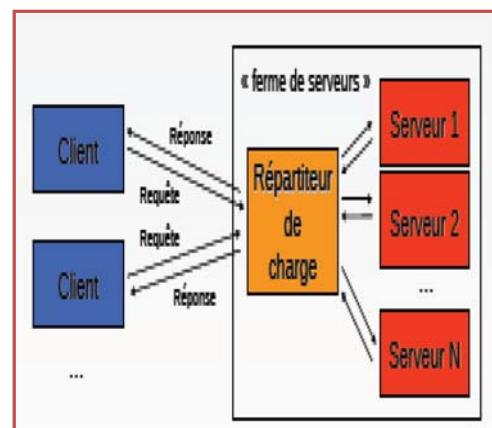
chap 1 :Introduction aux systèmes et applications répartis

46

Répartition de charge (load balancing)

Paralléliser sur plusieurs serveurs identiques s'exécutant sur des machines différentes le traitement des requêtes concurrentes. Les requêtes des clients passent par un répartiteur de charge qui les répartit sur N serveurs identiques.

Exemple: le site web de yahoo est hébergé simultanément sur plusieurs serveurs web d'adresses différentes: Le serveur de noms (DNS) joue le rôle de répartiteur de charge en traduisant «www.yahoo.com» par l'adresse de chacun des serveurs web à tour de rôle.



Points forts/ Points faibles de la répartition de charge ?

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

47

Points forts/ Points faibles de la répartition de charge

- + Transparent pour les clients,
- + Scalable : nombre de serveurs adaptable à la demande,
- + Tolérant aux défaillances : la défaillance d'un serveur n'interrompt pas le service,
- + Plus besoin de machines très chères : en mettre plusieurs,
- Les données ne sont plus centralisées mais dupliquées,
- Le répartiteur de charge devient le « maillon faible ».

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

48

1.4.2 Architecture Client/Serveur

Couches fonctionnelles

Toute application client-serveur est composée de 3 "couches" fonctionnelles:

- **Données(Persistance)**: données destinées à être conservées.
- **Traitements (Services Métiers)** : l'ensemble des règles de gestion et de la logique applicative.
- **Présentation** (interface utilisateur) : l'affichage, le dialogue avec l'utilisateur .

La répartition de ces couches entre clients et serveurs permet de distinguer entre les différents types d'architectures :

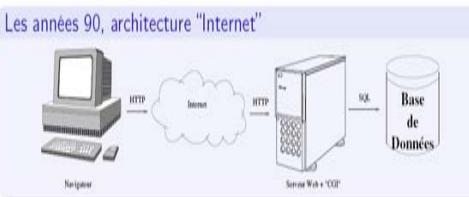
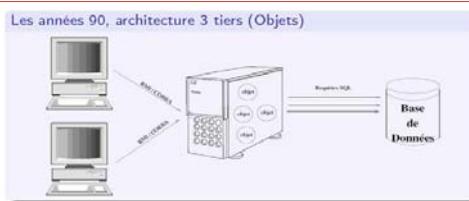
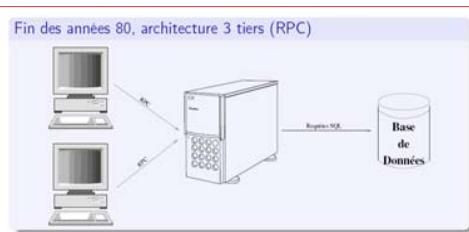
- **Architecture 2-tiers**
- **Architecture 3-tiers**
- **Architecture n-tiers**

01/02/2022

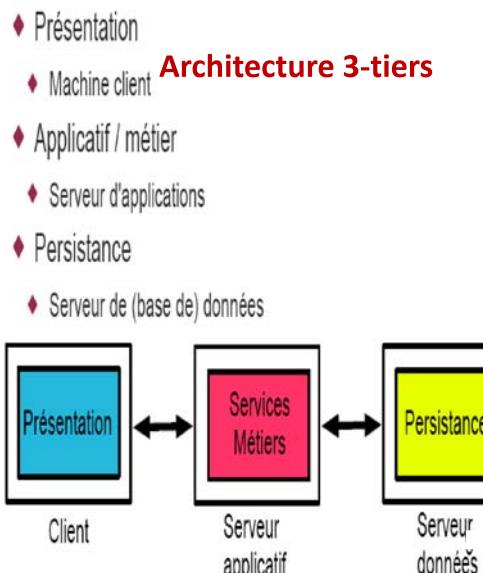
chap 1 :Introduction aux systèmes et applications répartis

49

1.4.2 Les architectures multicouches



Architecture 3-tiers



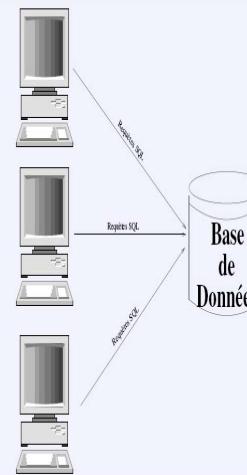
chap 1 :Introduction aux systèmes et applications répartis

51

1.4.2 Architecture Client/Serveur

Couches fonctionnelles

Les années 80, architecture 2 tiers

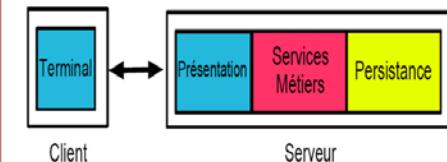


01/02/2022

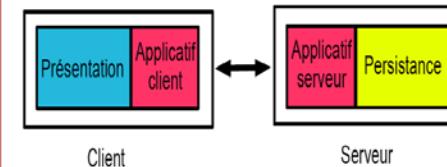
chap 1 :Introduction aux systèmes et applications répartis

Architecture 2 – tiers

- ◆ Terminal : client intègre un minimum de la partie présentation



- ◆ Applicatif : découpé entre client et serveur



50

Architecture 3-tiers : Points forts/ faibles

+ Découplage logique applicative/interface:

- Spécialisation des équipes de développement.
- Possibilité de changer ou d'avoir plusieurs interfaces utilisateur sans toucher à l'application elle-même.

+Découplage données/logique applicative:

- Possibilité d'utiliser des données existantes sans complexifier le modèle métier.
- Possibilité de changer le mode de stockage des données sans modifier la logique métier.

+Amélioration de la sécurité des données:

en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche de données.

+Réutilisation:

puisque toute la logique propre à l'application est concentrée dans le modèle métier.

- **Le client est soulagé, mais le serveur applicatif est fortement sollicité.**
L'équilibrage de la charge entre tous les niveaux va être atteint avec la génération suivante : **les architectures n-tiers**.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

52

Architecture n tiers

Une architecture n tiers se caractérise par une **distribution des services applicatifs**. Elle comprend généralement une **couche de présentation**, une **couche de services et d'objets métier** et une **couche d'accès aux données**.

- **La couche de présentation** contient les différents types de clients, léger (Web, ASP, JSP) ou lourd (Swing, WinForm)
- **La couche de service** contient les traitements représentant les règles métier (créer un compte, rechercher un client, calculer un amortissement, ...)
- **La couche d'objets métier** est représentée par les objets du domaine, c'est à dire l'ensemble des entités persistantes de l'application (Facture, Bon de Commande, Client, ...)
- **La couche d'accès aux données** contient les usines d'objets métier, c'est à dire les classes chargées de créer des objets métier de manière totalement transparente, indépendamment de leur mode de stockage (SGBDR, Objet, Fichiers, ...)

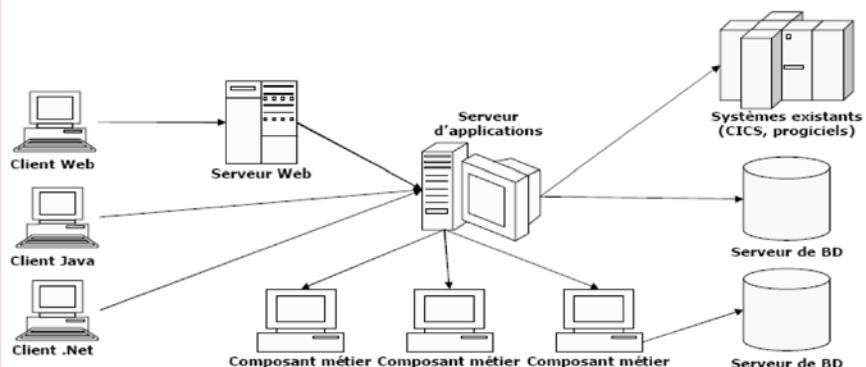
01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

53

1.4.2 Les architectures multicouches : Architecture n tiers

■ Architectures distribuée (n tiers)



Exemple d'une architecture n tiers

Exemples d'une architecture n tiers

1er tiers: Interface

- Téléphone, PDA,
- Browser, AppletViewer
- Client AWT, SWING, ou C, C++.

2 ième tiers: Gestion de la présentation, dynamité

- Serveurs web: Apache HTTP, Internet Information Services IIS ,
- Servlets, JSP,CGI, ASP.

3 ième tiers: Traitement, Logique applicative

- Serveurs d'applications, où sont installées les applications utilisées par les clients. Ils peuvent se répartir la charge de travail: (Apache tomcat, Jboss, JOnAS, GlassFish,...)
- EJB, RMI, Message Oriented Middleware (MOM): JMS

4ème tiers: Persistance, Gestion des données

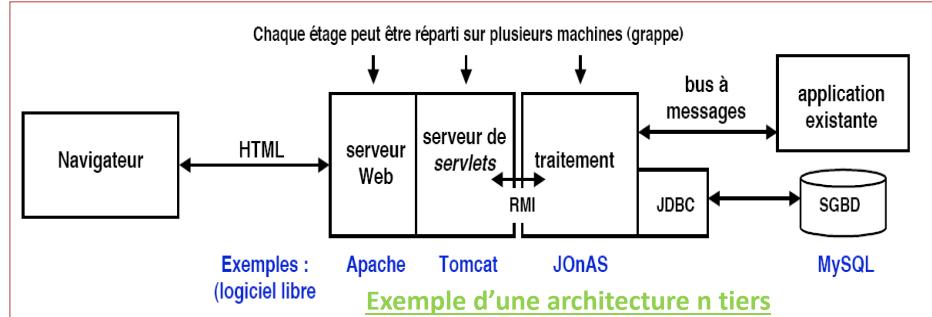
- JDBC, BD Objets, Moteur transactionnels

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

54

1.4.2 Les architectures multicouches : Architecture n tiers



Exemple d'une architecture n tiers

Remarque: L'appellation « n-tiers » pourrait faire penser que cette architecture met en œuvre un nombre indéterminé de niveaux de service, en fait ces derniers sont au maximum trois. L'architecture n-tiers qualifie la **distribution d'application entre de multiples services et non la multiplication des niveaux de service**.

Cette distribution est facilitée par l'utilisation de **middlewares ou intergiciels**.

01/02/2022

chap 1 :Introduction aux systèmes et applications répartis

55

chap 1 :Introduction aux systèmes et applications répartis

56

1.5 Conclusion

- **Un système réparti** (distribué) est un système faiblement couplé, composé d'un ensemble de processeurs ne partageant pas de mémoire ni d'horloge. Chaque processeur possède sa propre mémoire locale, les processeurs coordonnent leurs actions seulement par échange de messages.

Un système réparti doit assurer à ses utilisateurs, la transparence, la cohérence, la synchronisation, la désignation, l'Hétérogénéité, la Sécurité, la Performance, le Passage à l'échelle, la Communication.

Un système à plusieurs processeurs n'est pas forcément un système réparti .

1.5 Conclusion

- **Une application répartie** est composée de plusieurs parties s'exécutant sur plusieurs machines reliées par un système de communication.
- **La construction d'une application répartie** nécessite :
 - De définir l'architecture :
 - comment définir le modèle métier ?
 - Qu'est ce qui est distribué ? Persistant ?
 - Comment les différentes composantes communiquent elles?
 - De définir des stratégies de sécurité.
 - De choisir les technologies adéquates (Sockets, Java RMI, CORBA, JMS, Web Services)

Fin du chapitre1