

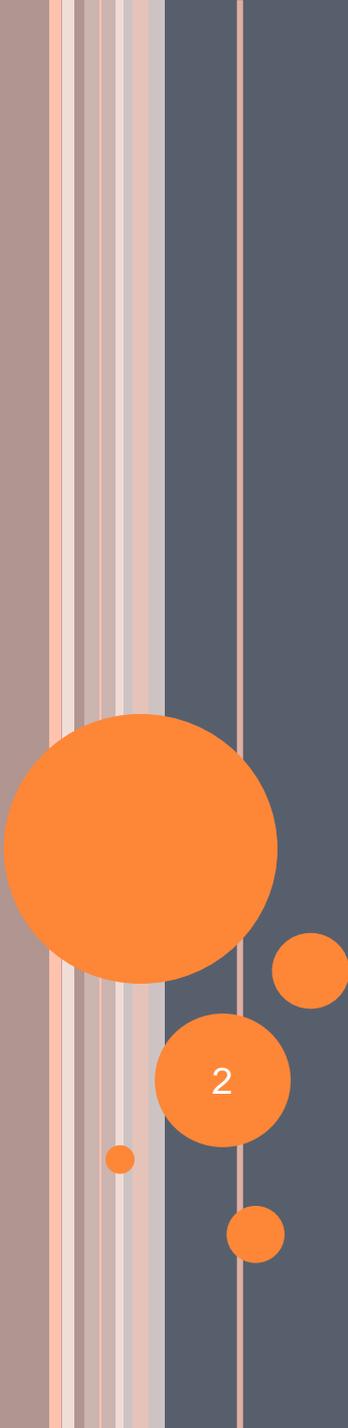
UE NSY107

INTÉGRATION DES SYSTÈMES CLIENT-SERVEUR

INTRODUCTION GÉNÉRALE

Formation au CNAM, Paris – Septembre 2018

Erik Boucher de Crèvecœur / Yves Laloum



DIPLÔME D'INGÉNIEUR EN INFORMATIQUE OPTION AISL ARCHITECTURE INGÉNIERIE DES SYSTÈMES ET DES LOGICIELS

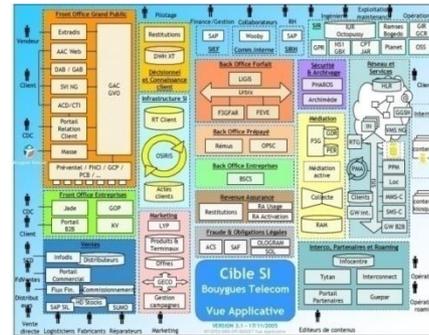
2

Yann Pollet, professeur Cnam
Erik Boucher de Crèvecœur, professeur associé

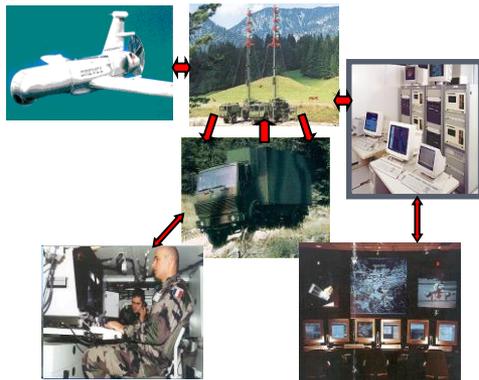
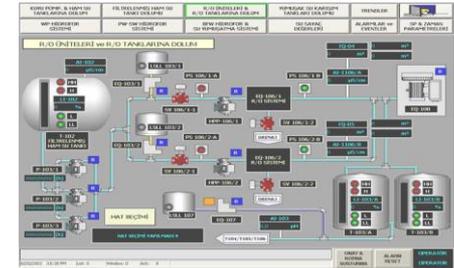
LES SYSTÈMES À LOGICIEL DOMINANT



Logiciels « personnels »



Systemes d'Information d'entreprise (CRM, ERP, SCADA, SCM, etc.)



Grands systemes (aeronautique, espace, sante, energie, industrie, environnement, ...)



Logiciels embarques (progression exponentielle)



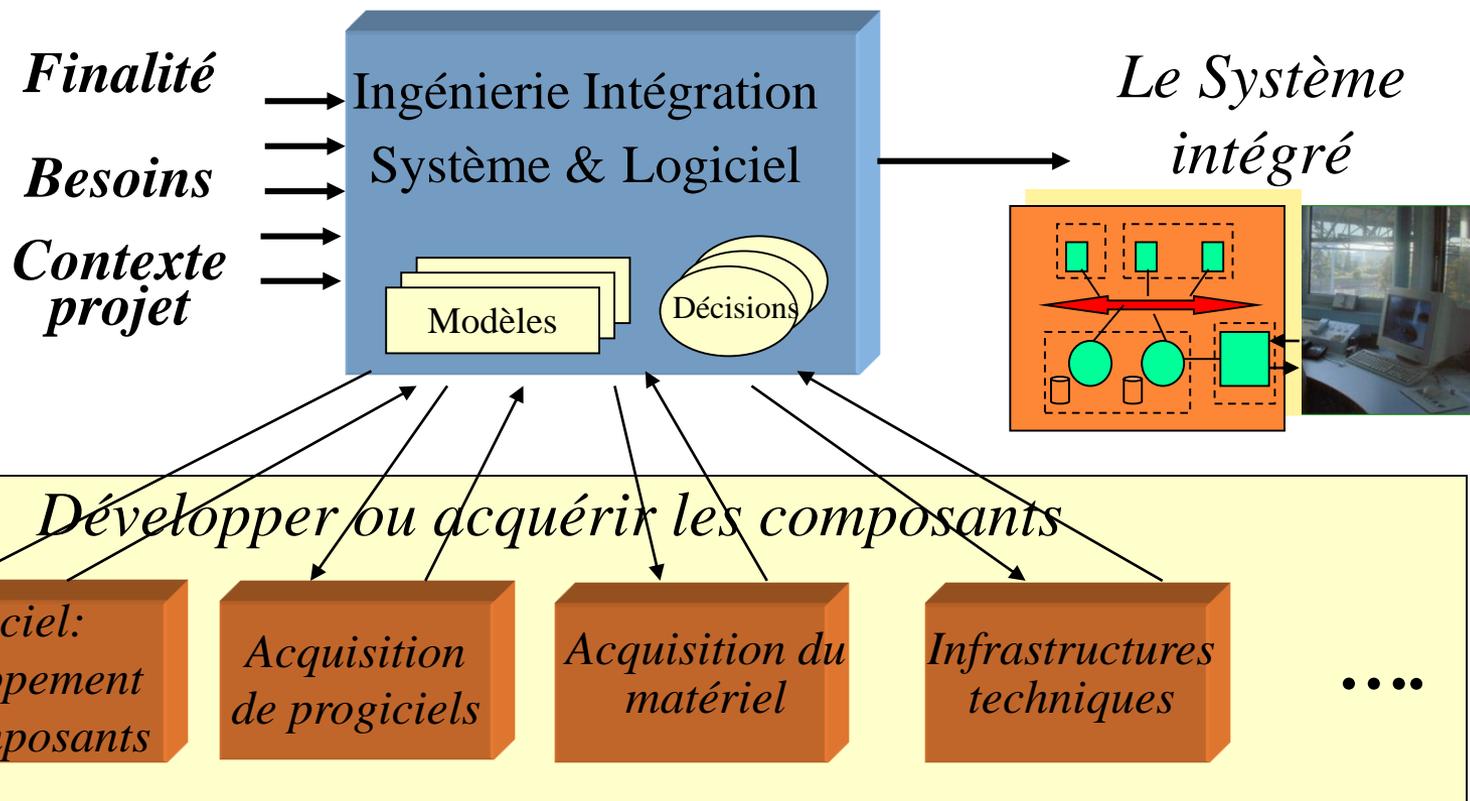
Le logiciel fait vivre plus d'**1 million de personnes** en France et représente un **poids économique global de près de 200 G€ /an**

OPTION AISL

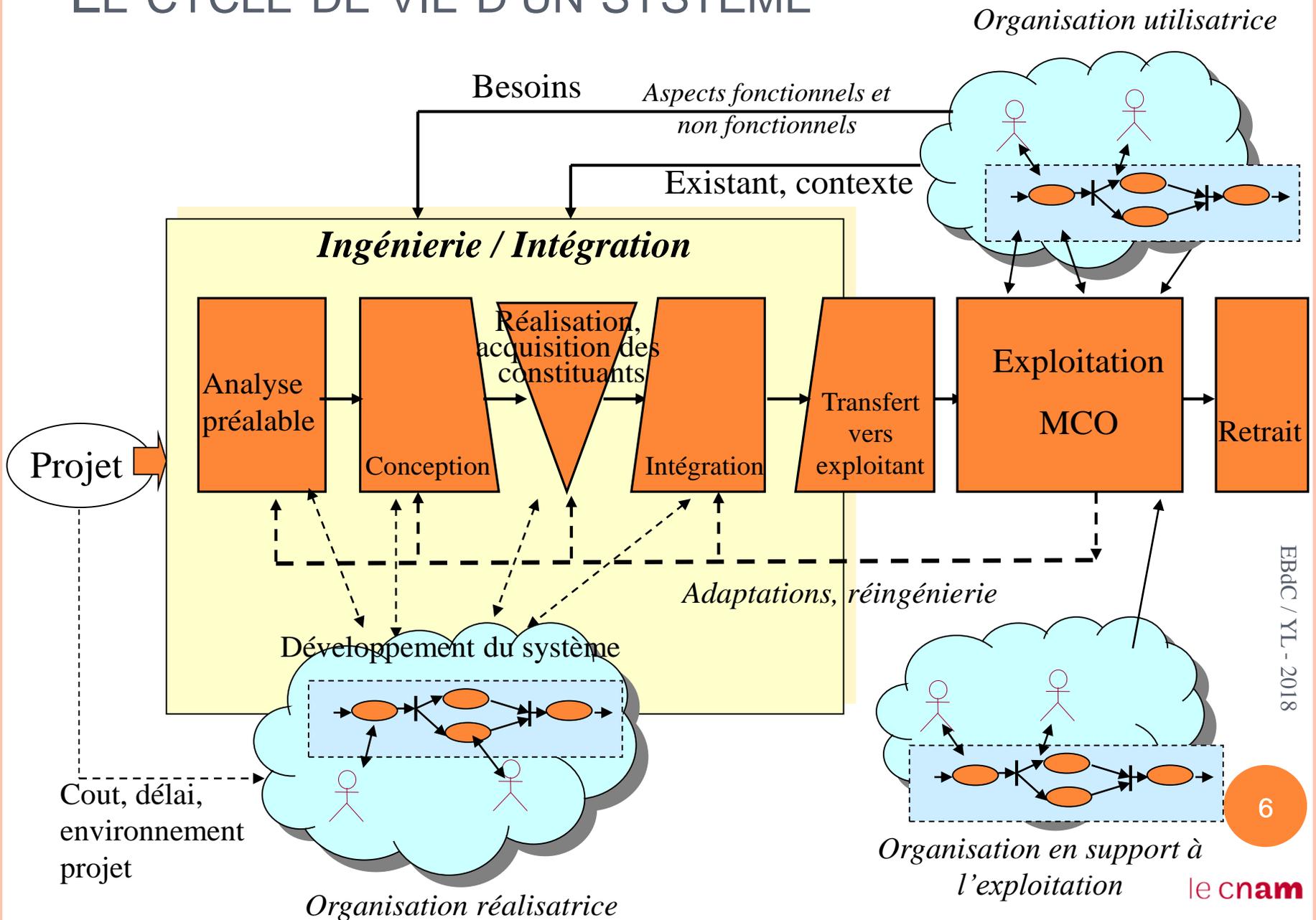
- Préparation aux **métiers de l'ingénierie et de l'intégration**
 - Importance croissante des métiers de l'ingénierie, développement de nouveaux métiers
 - Importance du management MOA et MOE
 - Prédominance croissante des activités liées à l'expression de besoins, à la spécification du système, à l'architecture, à l'intégration, aux tests
 - Développement par intégration de composants
 - Importance de l'exploitation et de la maintenance
- **Métiers visés:**
 - Architecte logiciel, architecte système
 - Chefs de projet MOE et MOA
 - Ingénieur d'intégration et tests;
 - Responsable maintenance et exploitation
 -

LE PROJET

Développement des systèmes complexes à prédominance logicielles, dans le respect de **Fonctionnalités** attendues, de **Coût**, **Délai** et **Qualité** par Intégration de composants multiples et hétérogènes, acquis ou développés spécifiquement



LE CYCLE DE VIE D'UN SYSTÈME



DIPLÔMES INFORMATIQUE AISL

Deux diplômes principaux:

- Concepteur Architecte (RNCP II), option Ingénierie de projets (bac+4)
- Ingénieur en Informatique, option AISL (bac+5)
- Licence à l'issu du cycle préparatoire I1 (bac+3)
- Recherche et continuation en thèse de doctorat

LE CURSUS INGÉNIEUR AISL

I1

Tronc commun : 7 UE couvrant les 5 domaines

60 ECTS

4 parcours

I2

<i>Intégration et architecture de systèmes</i>	<i>Systèmes embarqués et mobiles</i>	<i>Ingénierie de projets informatiques</i>	<i>Architecture logicielles</i>	2 UE de Management et organisation (12 ECTS)	
				Management de projet (4 ECTS)	Test d'anglais (6 ECTS)
				Expérience professionnelle (15 ECTS)	
2 UE de spécialisation (12 ECTS)					
1 UE complémentaire (6 ECTS)					

I3

1 couple d'UE clôturant le parcours (12 ECTS)			Information et communication (4 ECTS)	L'ingénieur au XXI ^e siècle (4 ECTS)
---	--	--	---------------------------------------	---

Mémoire d'ingénieur (préparation, rédaction et soutenance)

45 ECTS

8

LES VALEURS DE SPÉCIALISATION DE I2

- Intégration des systèmes client-serveur (NSY107)
- Conception de logiciels intranet: patrons et canevas (NSY102)
- Ingénierie de la qualité et du contrat de service: sécurité, performances (GLG102)
- Test de validation du logiciel (GLG101)
- Sûreté de la programmation orientée objets (NFP101)
- Applications concurrentes et outils (NFP103)

(6 ECTS chacune)

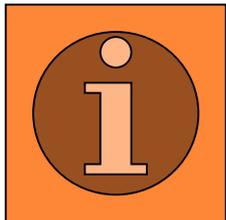
LES VALEURS DE FIN DE PARCOURS (I3)

- Architectures et technologies pour l'intégration 1 et 2 (NSY205 et NSY206)
- Ingénierie de projets informatiques 1 et 2 (GLG206 et GLG207)
- Architectures Logicielles Java 1 et 2 (GLG203 et GLG204)
- Architectures, patterns et intégration: systèmes embarqués et mobiles en Java 1 et 2 (NSY208 et NSY209)
- Construction rigoureuse des logiciels 1 et 2 (NFP209 et NFP210)

(12 ETCS au total)

ENVIRONNEMENT DE L'ENSEIGNEMENT NSY107

- **Yann POLLET** :
Professeur Titulaire de la Chaire Intégration des Systèmes
yann.pollet@cnam.fr
- **Erik BOUCHER DE CRÈVECŒUR** :
Professeur Associé au CNAM
erik.boucherdecrevecoeur@lecnam.net
- Secrétariat de la chaire Intégration des Systèmes



- **Safia SIDER**
- à mettre en copie pour toute correspondance
- **safia.sider@lecnam.net**

OBJECTIFS DE L'UE

- Objectif 1:
 - Donner une vision globale de la distribution de services et de données et des architectures modernes à travers l'ensemble des technologies existantes et coopérantes en entreprise
- Objectif 2 :
 - Introduire les outils d'ingénierie permettant de concevoir des systèmes distribués
- Objectif 3 :
 - Préparer aux valeurs de fin de parcours :
Notamment « Architectures et technologies pour l'intégration 1 et 2 (NSY205 et NSY206) »
ou NSY208 et NSY209 : Intégration des Systèmes temps réel embarqués et Java
ou GLG206 et GLG 207 : Ingénierie des projets informatiques

FORME DE L'ENSEIGNEMENT

- 1 semestre de cours

Les mardi soirs de 17H45 à 21H45 jusqu'au 22/01/2019

- Séances de cours avec un thème par séance

- 1^{ère} partie : Composants des architectures distribuées
 - Architecture serveurs. Disponibilité et performances
 - Virtualisation. Cloud computing
 - Sécurité des systèmes d'information. Principes cryptographiques
 - Introduction au client/serveur : historique, évolution et middleware objets
- 2^{ème} partie : Technologies de distribution
 - Bases de données distribuées et fédérées. Systèmes transactionnels
 - Architectures d'objets distribués
 - Middleware orientés messages. Technologies Internet
 - Web Services. Architectures REST
 - Architectures Web avec Javascript : AngularJS, NodeJS
 - Comparaison JEE et .NET
- Introduction à l'urbanisation des SI

PLANNING PRÉVISIONNEL

UE NSY107 - Intégration des systèmes client-serveur

Séance	Dates - mardi 18h00 à 21h45	Cours (Erik Boucher de Crèvecoeur, sauf indication), TD , → <i>Travaux personnels</i> Amphi Georges-Friedmann
1	25/09/2018	Introduction Générale et Présentation de l'UE
2	02/10/2018	Architecture des serveurs. Haute disponibilité et méthodologie des performances
3	09/10/2018	Virtualisation des Infrastructures. Cloud Computing
4	16/10/2018	Sécurité des systèmes d'information. Principes cryptographiques
5	23/10/2018	Introduction au client/serveur : historique, évolution et middleware objets
6	30/10/2018	Bases de données distribuées et fédérées. Systèmes transactionnels → <i>Exercices</i>
7	06/11/2018	Architectures d'objets distribuées : CORBA, RMI, COM+ → <i>Exercices</i>
	13/11/2018	PAS DE COURS → <i>Faire les exercices</i>
8	20/11/2018	Middleware orientés messages (MOM). Technologies Internet, XML → <i>Exercices</i>
9	27/11/2018	Web Services, SOAP, WSDL. Architectures REST → <i>Exercices</i>
10	04/12/2018	Architectures Web avec Javascript : AngularJS, NodeJS (Jérôme Gorin)
11	11/12/2018	TD - Correction des exercices + Présentation de l'étude de cas (à rendre début janvier)
12	18/12/2018	PAS DE COURS → <i>Lire le support « Comparaison des architectures JEE et .NET »</i>
	25/12/2018	INTERRUPTION DES COURS → <i>Faire l'étude de cas</i>
	01/01/2019	INTERRUPTION DES COURS → <i>Faire l'étude de cas</i>
13	08/01/2019	Introduction à l'urbanisation des SI, SOA et BPM → <i>Rendre l'étude de cas</i>
14	15/01/2019	TD - Correction de l'étude de cas
15	22/01/2019	Révisions et synthèse (à décaler au mercredi ???)
	29/01/2019 ou 05/02/2019	EXAMEN écrit - 3 heures

VALIDATION DE L'ENSEIGNEMENT

- 2 sessions d'examen :
 - Fin janvier/début Février 2019
 - Avril 2019 en rattrapage
 - Note requise à l'examen : 10
- Préparation à l'examen :
 - Suivi des cours et supports de cours
 - publiés sur le site du CNAM (deptinfo) au fur et à mesure.
- Un devoir de préparation à l'examen sera donné en milieu de semestre, à rendre après les vacances de Noël.
 - Devoir dans l'esprit de l'examen.

RÉFÉRENCES GÉNÉRALES UTILES

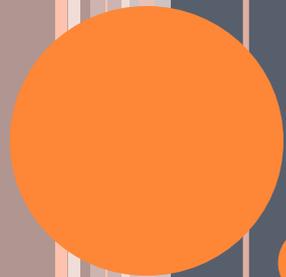
○ René J. CHEVANCE

- Serveurs et Multi processeurs Architectures parallèles Eyrolles
- Server Architectures ELSEVIER DIGITAL PRESS 2004

○ Jean Pierre MEINADIER

- Ingénierie et intégration des Systèmes Hermès sciences
- Le métier d'intégration de Systèmes Hermès sciences

○ Autres références au fur et à mesure de chaque terme abordé.



PETIT HISTORIQUE...

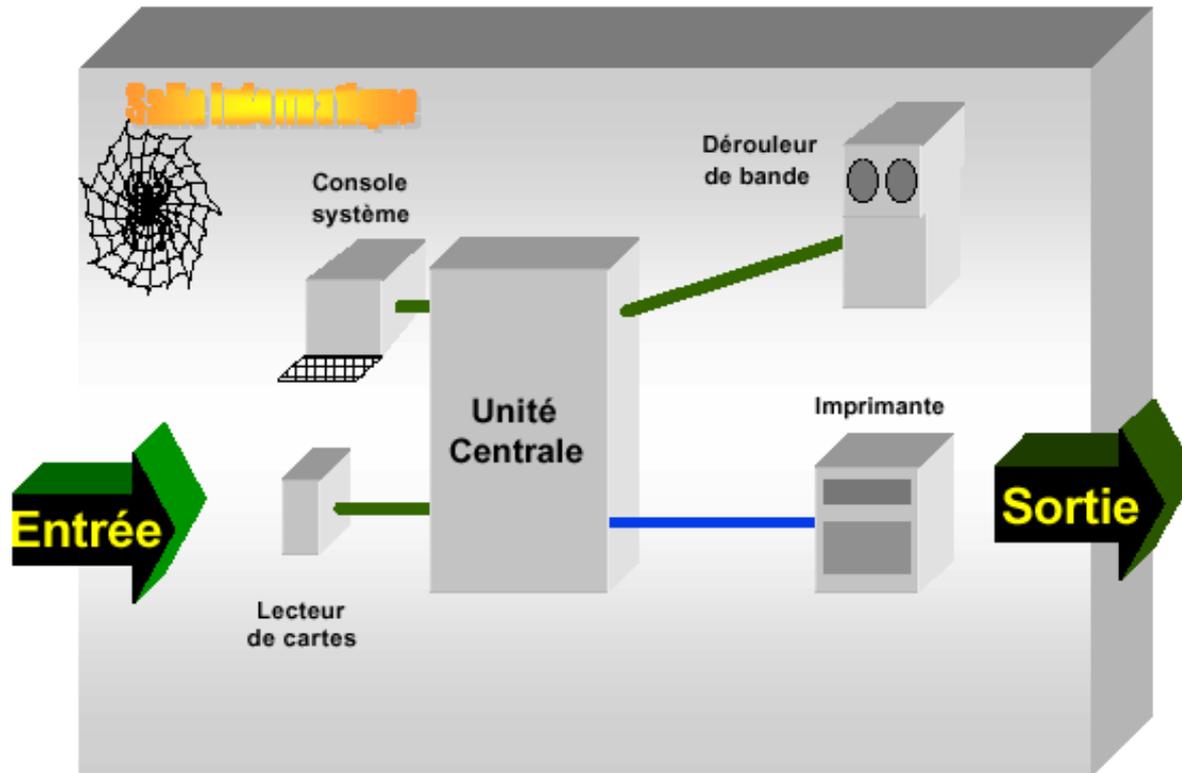


DES PREMIERS ORDINATEURS AUX SYSTÈMES URBANISÉS



AU COMMENCEMENT....

1. Début de l'informatique : années 1960



On peut difficilement parler d'interactivité: le programme ainsi que ses paramètres sont entrés dans l'unité centrale à l'aide de cartes perforées...

Une erreur sur une seule carte et tout le programme doit être relancé.

Le résultat est généralement une impression papier sur une imprimante à rouleau.

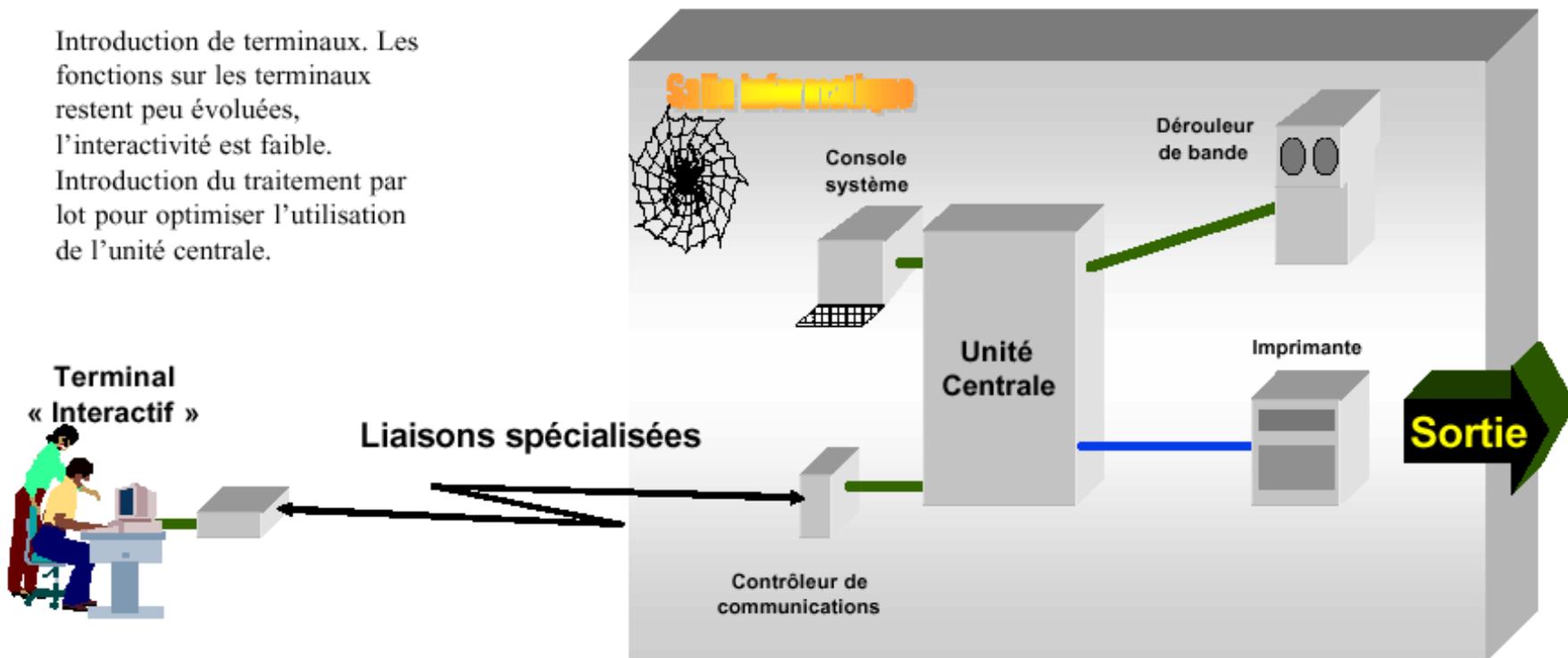
La console système sert uniquement à contrôler l'activité de l'unité centrale et à ordonnancer les tâches (on parle de lots de travaux).

La première amélioration sera l'introduction de lecteur de bande magnétique.

LES DINOSAURES...

2. années 1970 – architecture centralisée / terminaux

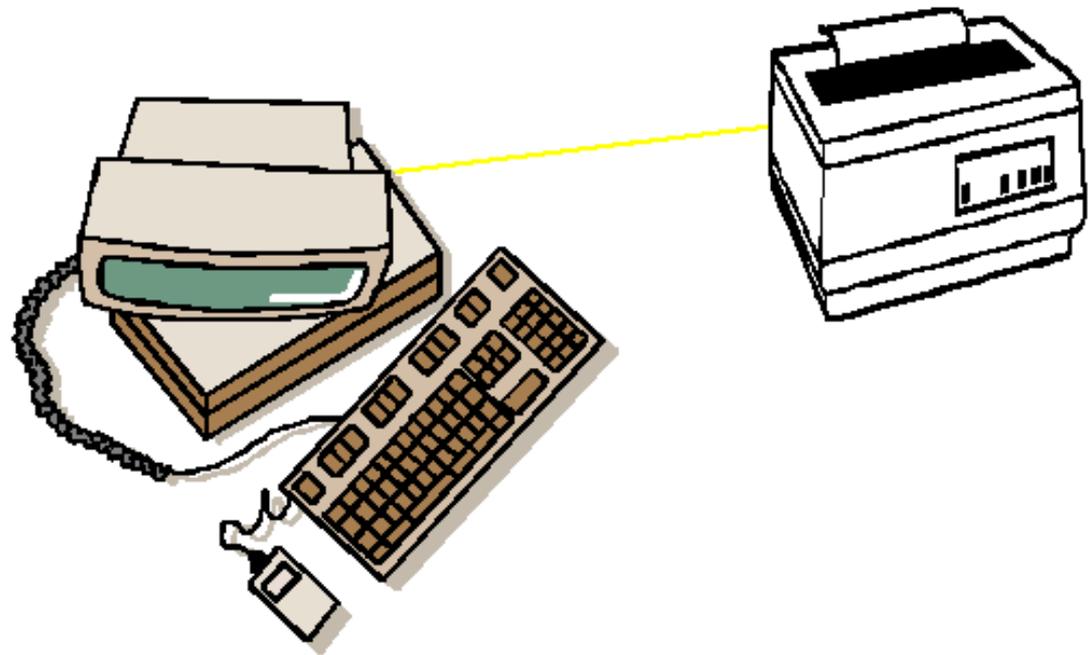
Introduction de terminaux. Les fonctions sur les terminaux restent peu évoluées, l'interactivité est faible. Introduction du traitement par lot pour optimiser l'utilisation de l'unité centrale.



LA MUTATION....

3. Début des années 1980 – introduction de la micro-informatique

En 1981, IBM sort le premier Personal Computer. C'est le début de la micro-informatique.



LES DÉBUTS DIFFICILES DE LA DISTRIBUTION

3. Début des années 1980 (suite)– échange d'information par 'vélo processing'

L'échange d'information entre ordinateurs se fait par disquette ou autre média compatible (bande, ...).

Ce moyen reste toutefois toujours parfois d'actualité lorsqu'il n'y a pas de connexion réseau, voire même pour des problèmes de sécurité (voir § sur la sécurité des réseaux).



ON COMMENCE À SE PARLER....

4. Fin des années 80 – Stations de travail & réseaux locaux

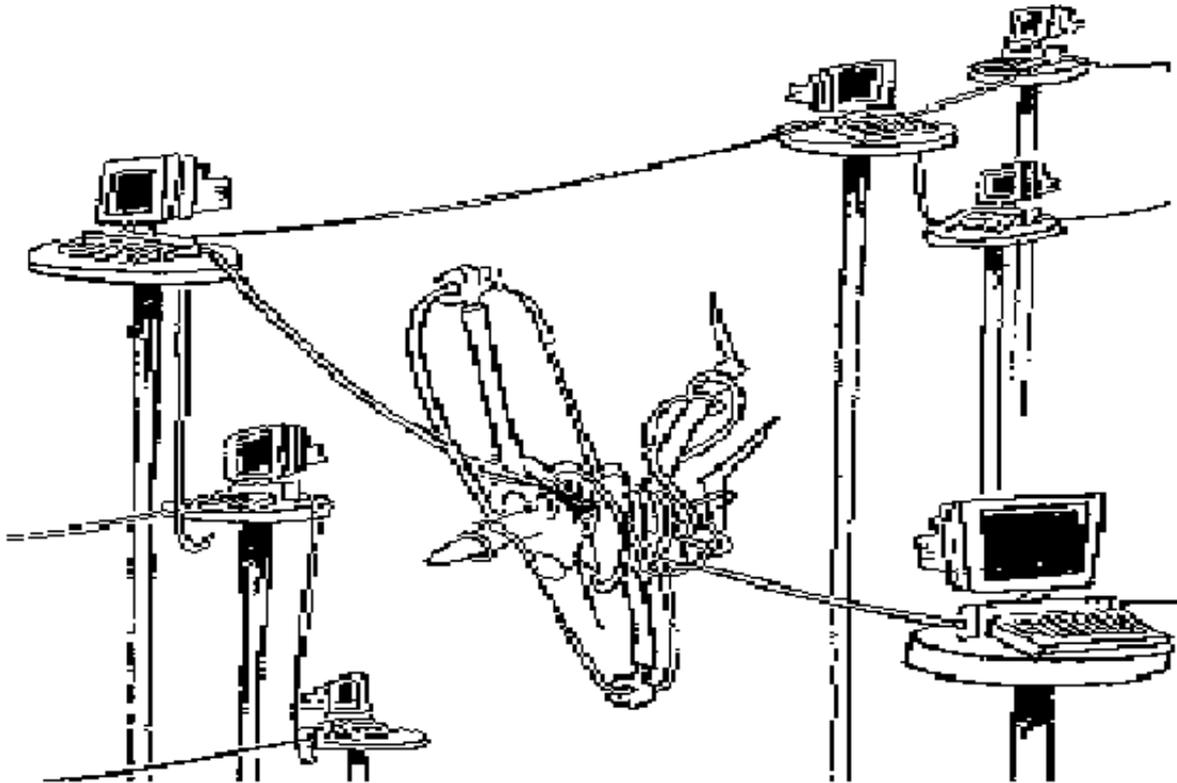
Les stations de travail prennent place pour les calculs scientifiques dans les universités et les entreprises.

Introduction des premiers réseaux locaux: Token-Ring, Ethernet.



L'INFRASTRUCTURE SE MET EN PLACE...

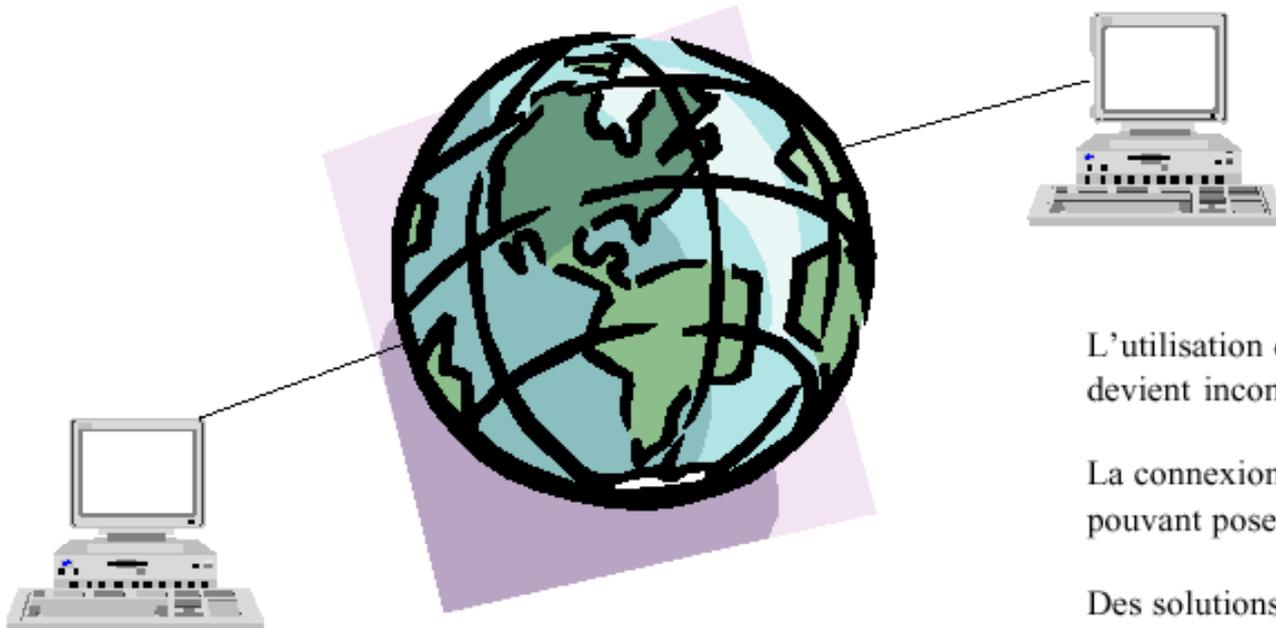
5. Milieu des années 90: Interconnexion des réseaux locaux



Au niveau des entreprises:
l'interconnexion des réseaux locaux mis en place dans les différents services devient une nécessité.
L'architecture n'est pas toujours correctement spécifiée.

LES AUTOROUTES DE L'INFORMATION....

6. Fin des années 90: Internet est incontournable.

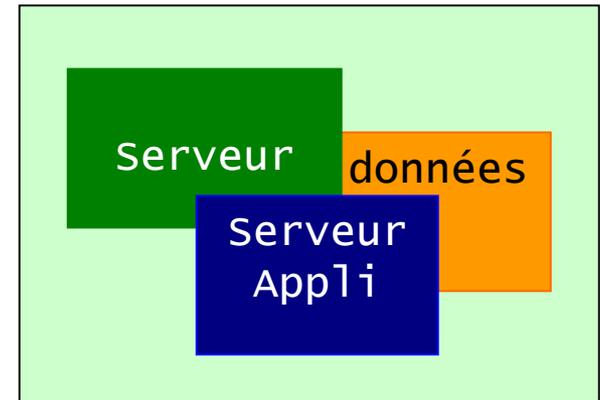
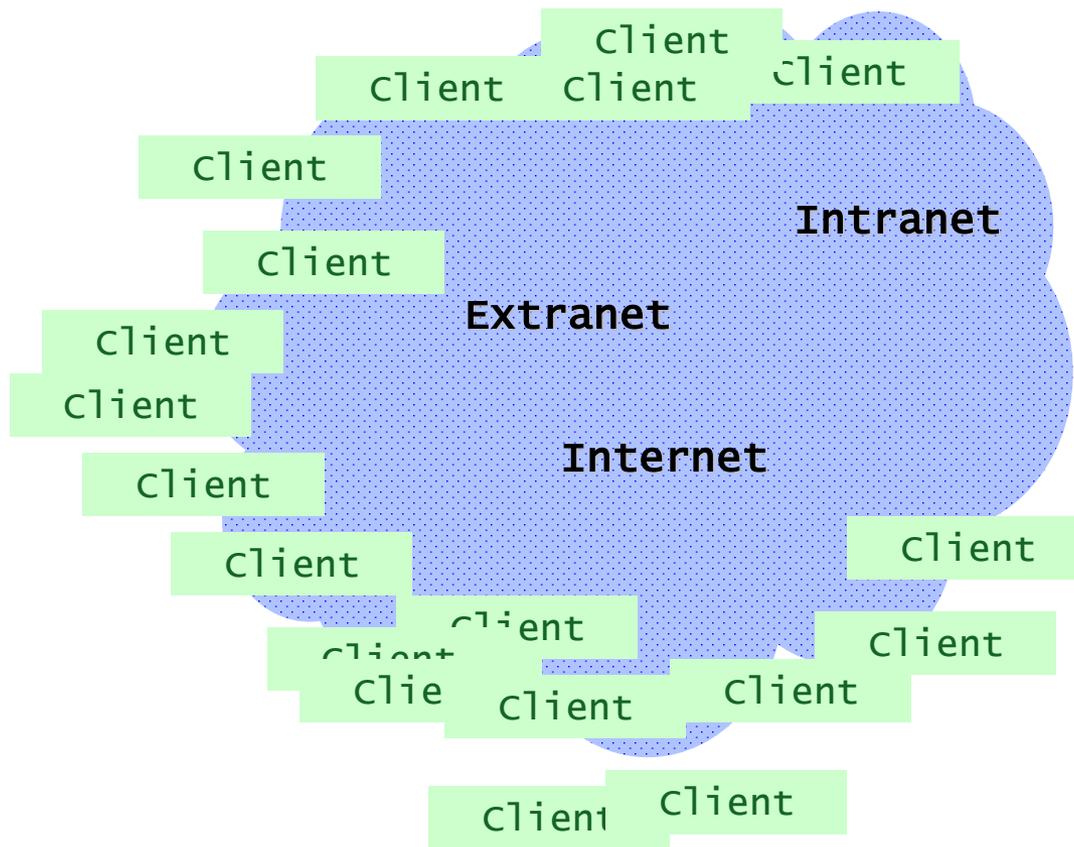


L'utilisation de technologie Internet devient incontournable.

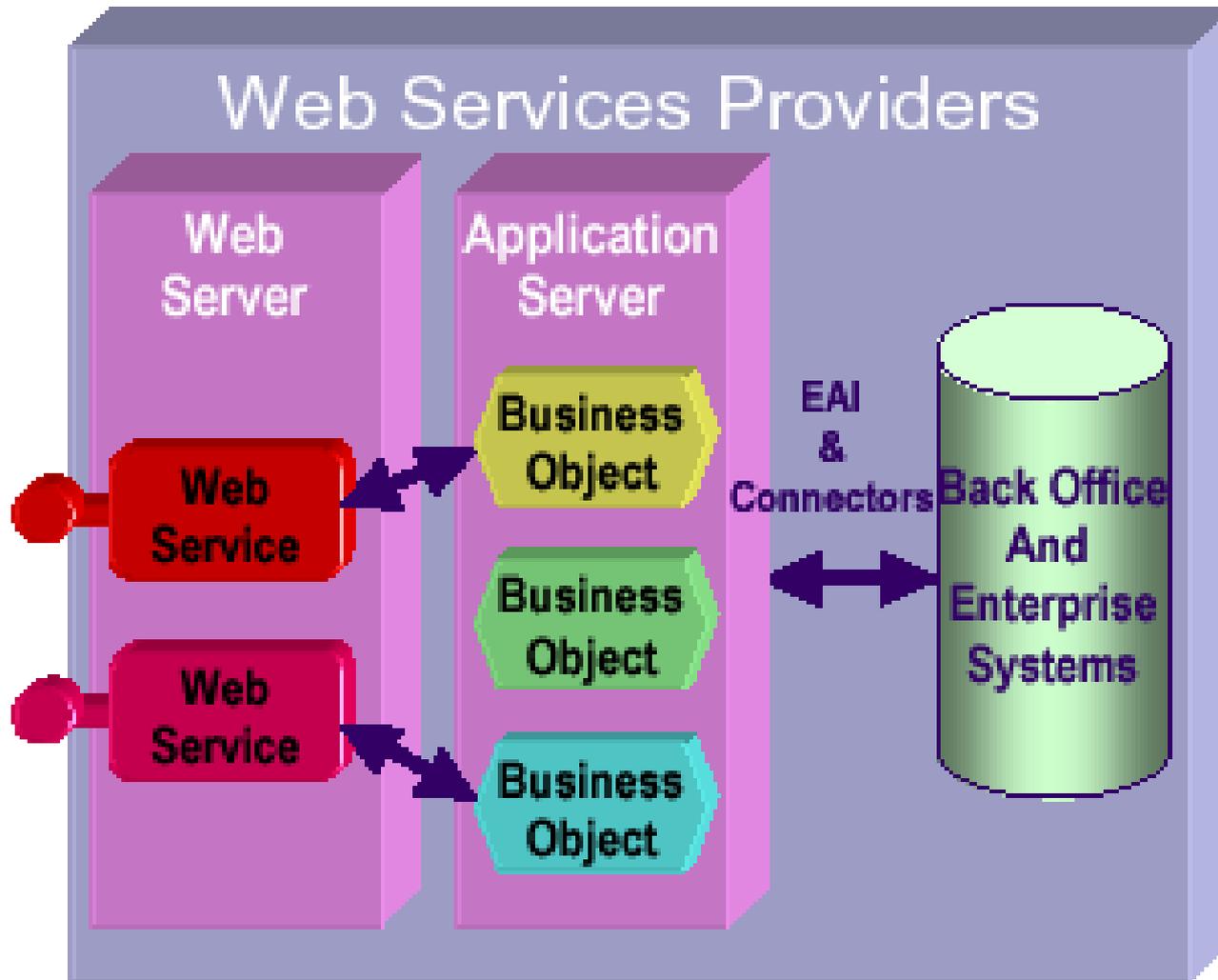
La connexion à Internet d'une entreprise pouvant poser des problèmes de sécurité.

Des solutions de développement dans un vase clos restreint à l'entreprise, porte le nom d'Intranet ou encore d'Extranet, lorsqu'il s'agit de relier plusieurs points distants par des technologies Internet.

ON COMMENCE À RATIONALISER....

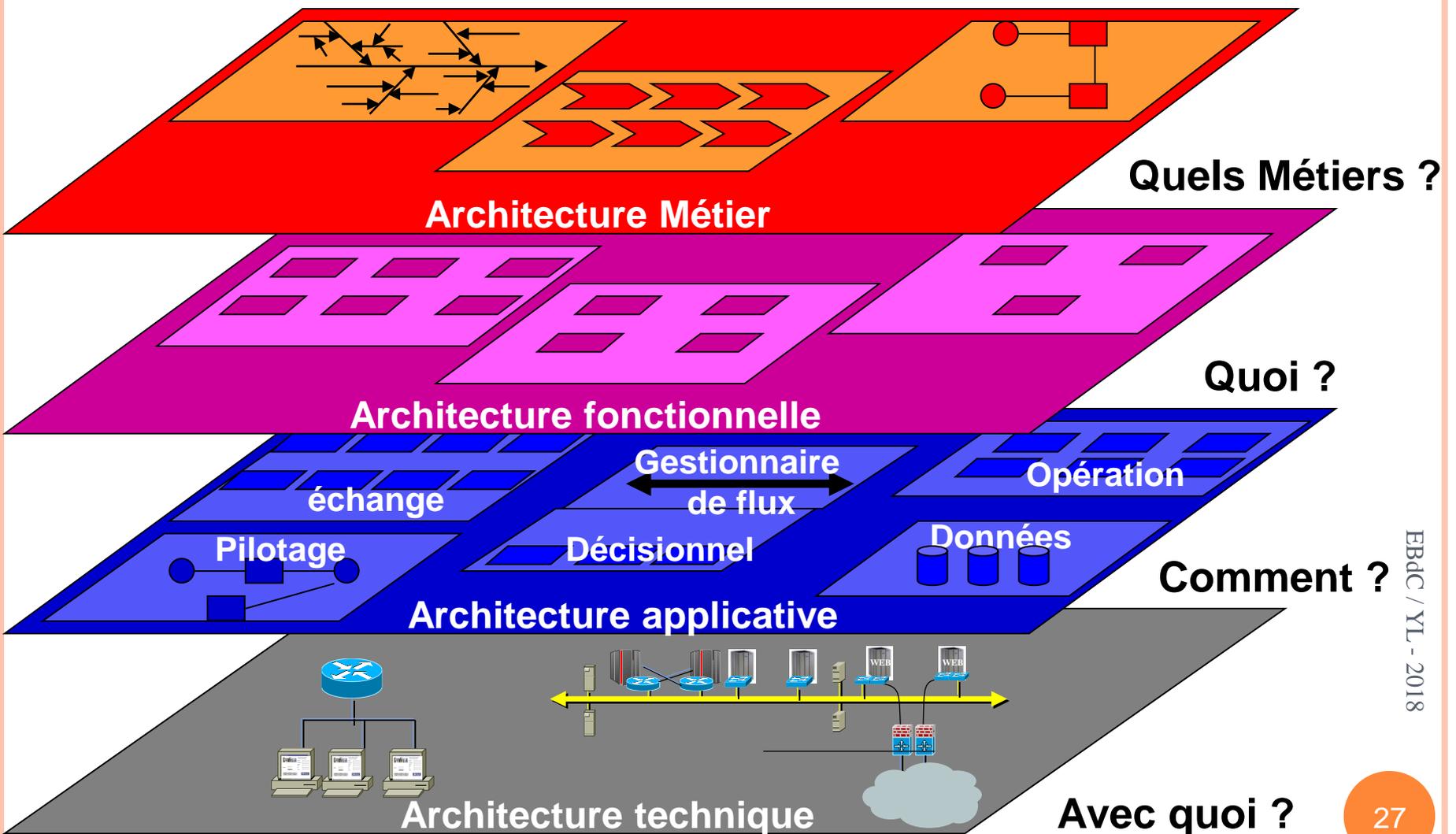


L'ACCÈS UNIVERSEL : WEB SERVICES



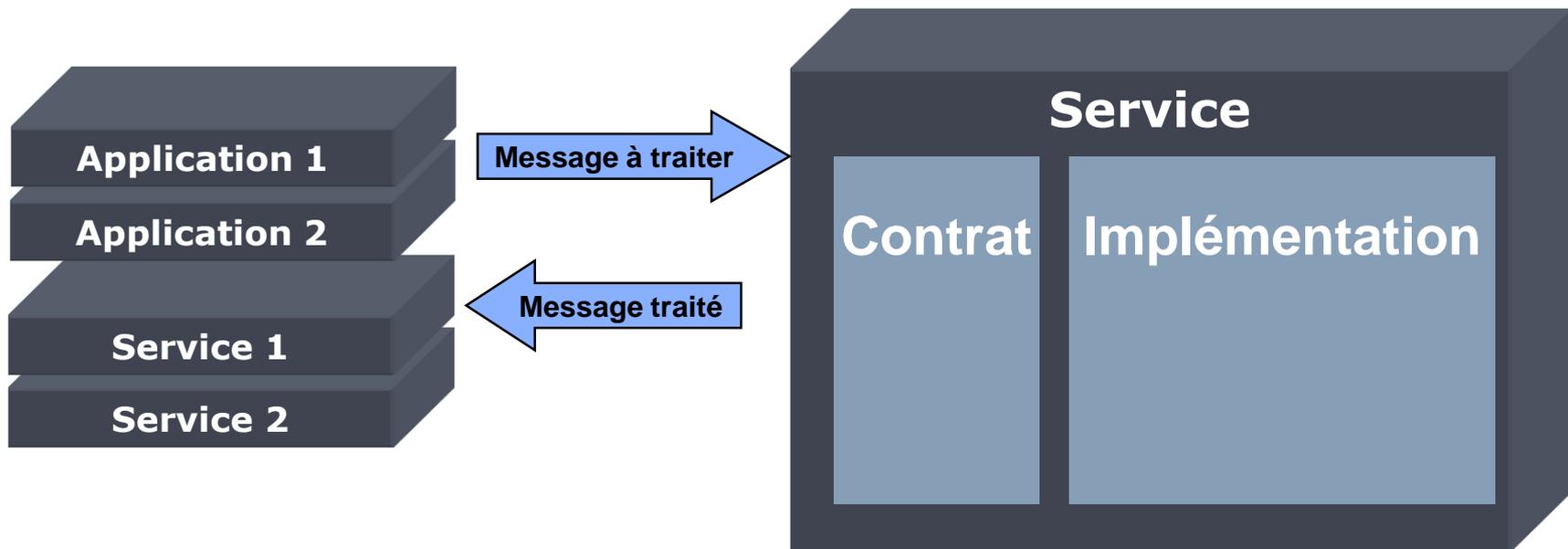
Didier Doreau, Web Services

LES DIFFÉRENTES ARCHITECTURES



ARCHITECTURE ORIENTÉE SERVICE : SOA

- Un service est une fonction qui reçoit des messages et qui les restitue après traitement.
- Une Architecture Orientée Service diffuse des services



EBdC / YL - 2018

- Un service respecte un contrat
- Un service est sans état

Un service est réutilisable

L'AVENIR : URBANISATION DU SYSTÈME D'INFORMATION

EAI à prises Web services

Bus de services distribués

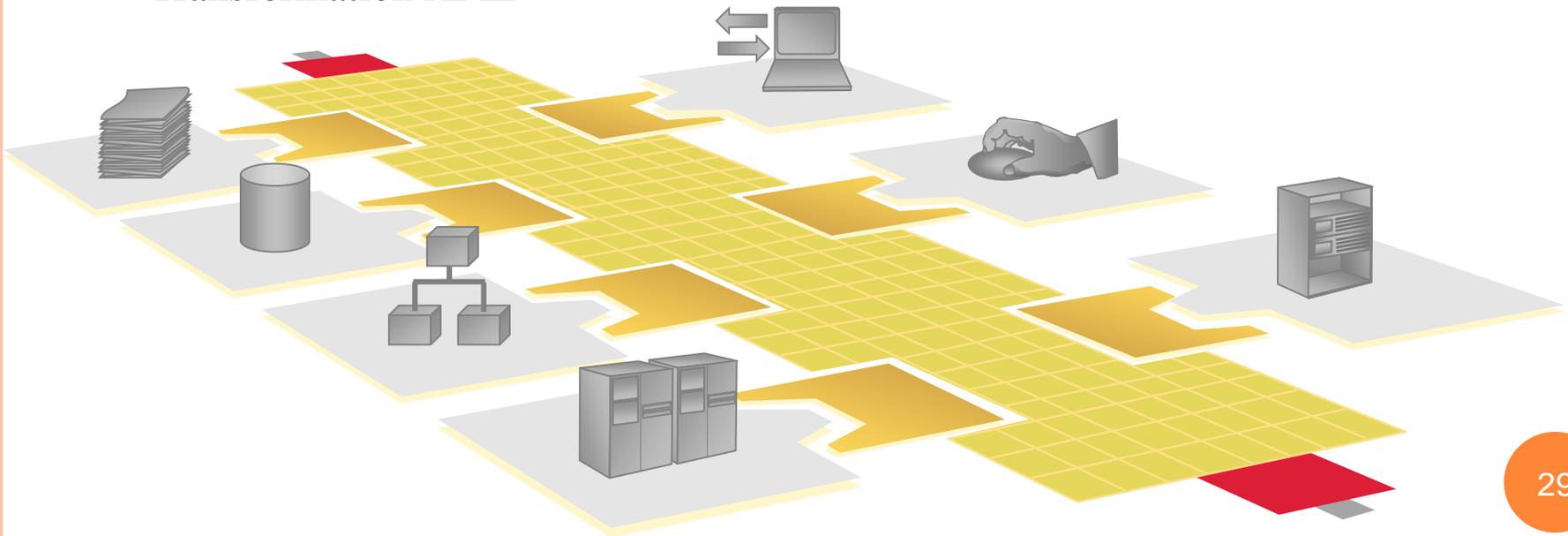
Backbone pour messagerie fiable

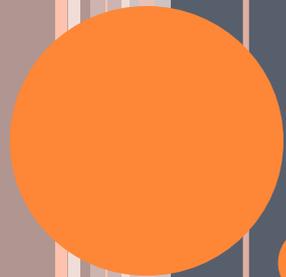
Routage intelligent basé contenu

Coordination des processus

Cadre sécurisé

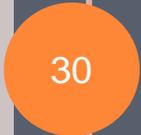
Transformation XML





ARCHITECTURES DISTRIBUÉES

BASES et CONCEPTS



30



CARACTÉRISATION DES PREMIERS SYSTÈMES CENTRALISÉS

- Tous les traitements s'effectuent sur un seul système
- Operating system généralement propriétaire
- Terminaux en mode caractère (dumb terminal)
- Exemples : IBM MVS, GCOS BULL, VMS DEC, Unix avec émulation de terminal

AVANTAGES/INCONVÉNIENTS DES SYSTÈMES CENTRALISÉS

○ Avantages :

- Sécurité plus facile à gérer
- Coûts d'administration faible par utilisateur
- Centralisation des choix:
 - Installation
 - Mise à jour
 - Utilisation des applications

○ Inconvénients :

- Interface utilisateur difficile à utiliser
- Faible autonomie de l'utilisateur
- Systèmes propriétaires
- Difficile de migrer vers un autre Système

CARACTÉRISATION DES SYSTÈMES DÉCENTRALISÉS

- Les applications sont découpées en composants s'exécutant sur des machines différentes
- En général, une partie cliente et une partie serveur. Modèle Requête/réponse.
- Downsizing : Décentralisation des traitements
- Data Distribution : Répartition des données
- Revamping : habillage d'ancienne application et relookage graphique
- Apparition du « MiddleWare »
- Exemples : RPC Microsoft, CTOS, Requêtes SQL

AVANTAGES/INCONVÉNIENTS DES SYSTÈMES DÉCENTRALISÉS

○ Avantages :

- Répartition des traitements
- Interface utilisateur plus conviviale (graphique)
- Très grande autonomie de l'utilisateur
- Meilleure performance globale.
- Données plus accessibles en cas de problèmes réseaux

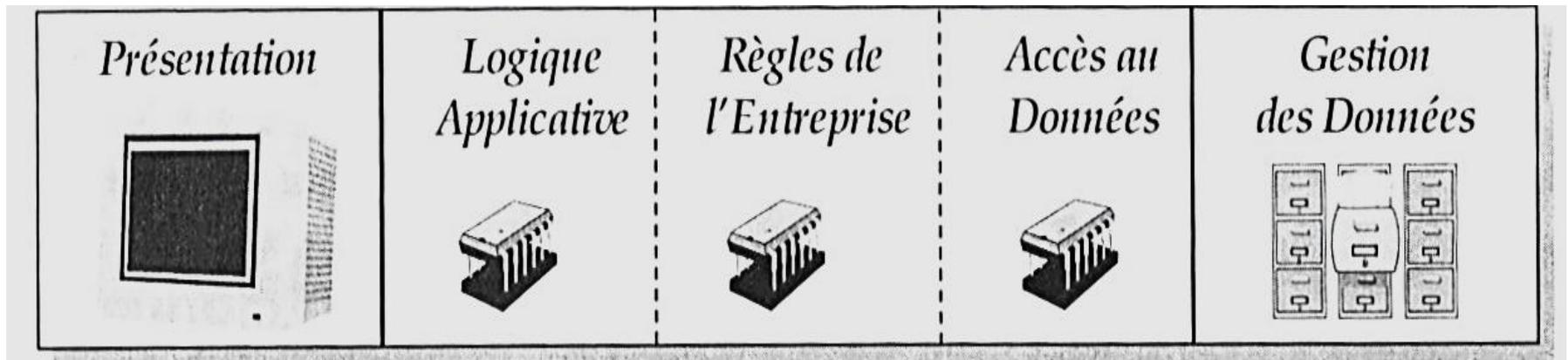
○ Inconvénients :

- Coût d'administration plus élevé par utilisateur
- Faible maîtrise des effets d'échelle
- Déploiement des applications plus compliquée
- « fat client » et gestion des configurations clients plus problématique

QU'EST CE QU'UNE ARCHITECTURE DISTRIBUÉE?

- Une plate-forme pour construire des applications
- Définit les composants d'un système et leur interaction
- Permet de construire un système complexe à partir d'éléments simples
- Permet de construire un système multi-parties qui se comporte comme un tout

LES 5 COMPOSANTS LOGIQUES D'UNE APPLICATION



La stratégie d'affectation d'un composant de l'application à un système détermine son niveau de distribution

PROBLÈMES LIÉS À LA DISTRIBUTION

○ La complexité

- Multiplicité des composants

○ L'hétérogénéité

- Multiplicité des langages
- Multiplicité des plates-formes

○ La distribution

- Multiplicité des protocoles
- Multiplicité des localisations possibles
- La gestion du parallélisme et de la concurrence

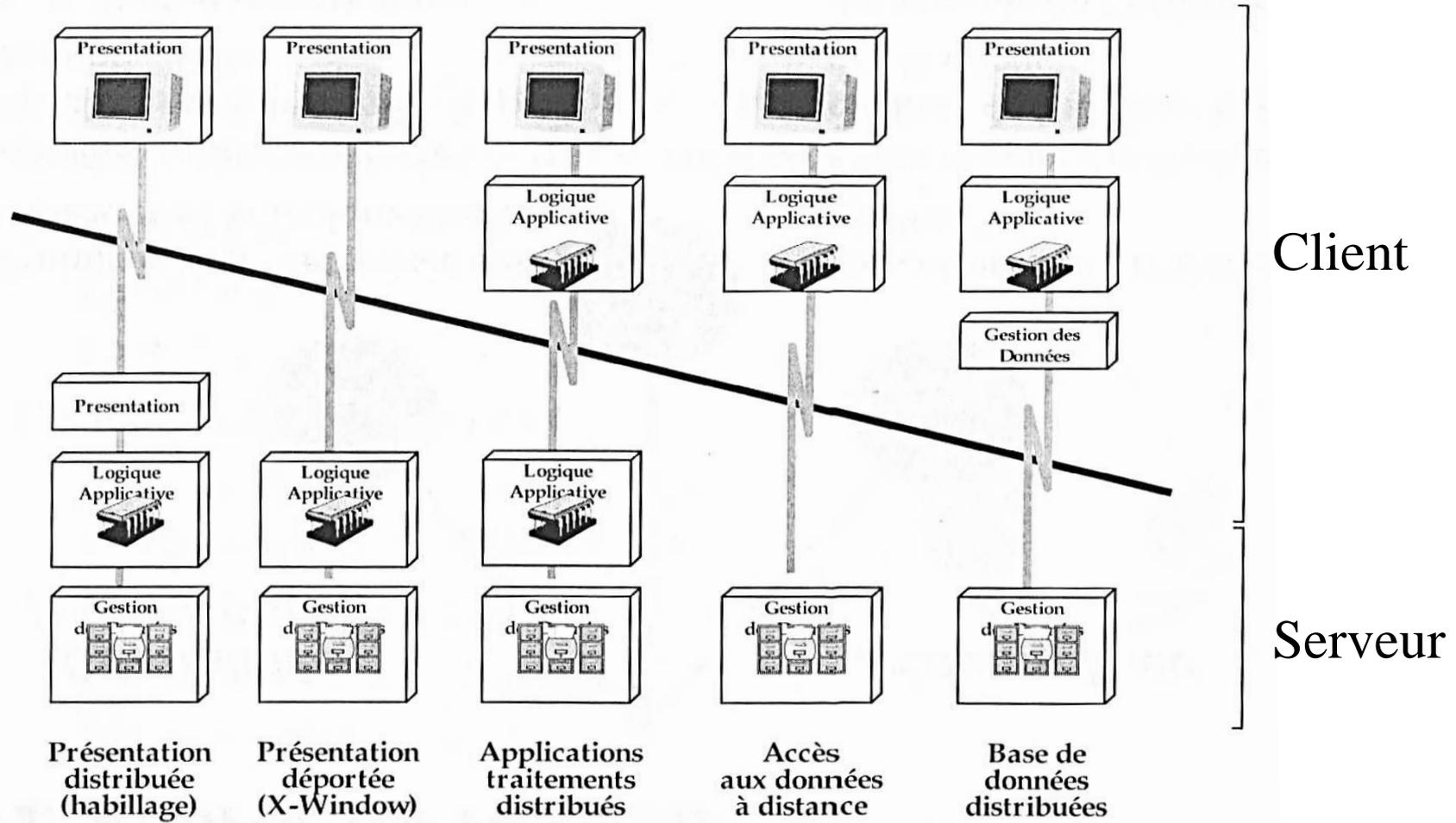
○ L'intégration

- Multiplicité des technologies
- Multiplicités des interfaces

LE CLIENT/SERVEUR

- Une architecture est dite client/serveur lorsque l'ensemble des composants d'une application est réparti sur plusieurs entités systèmes.
- Les Architectures Clients/serveurs introduisent des mécanismes permettant d'effectuer des requêtes et des réponses à travers le réseau.
- Selon la décomposition faite au niveau des composants, la répartition de charge peut être plus ou moins équilibrée (voir schéma suivant)

Les modèles de Client-Serveur



Client léger - - - - - > client lourd

LES FAMILLES DE CLIENT/SERVEUR

- Le Client/Serveur de Présentation distribuée et déportée :
 - Gestion de la présentation sur le Client parfois de manière simple (Telnet) parfois de manière évoluée exemple (X-WINDOWS)
- Le Client/Serveur de traitement
 - Gestion de la présentation et du traitement sur le Client (notion de « fat client ») | l'accès aux données s'effectuant sur le serveur

LES SERVICES RÉSEAUX ("MIDDLEWARE")

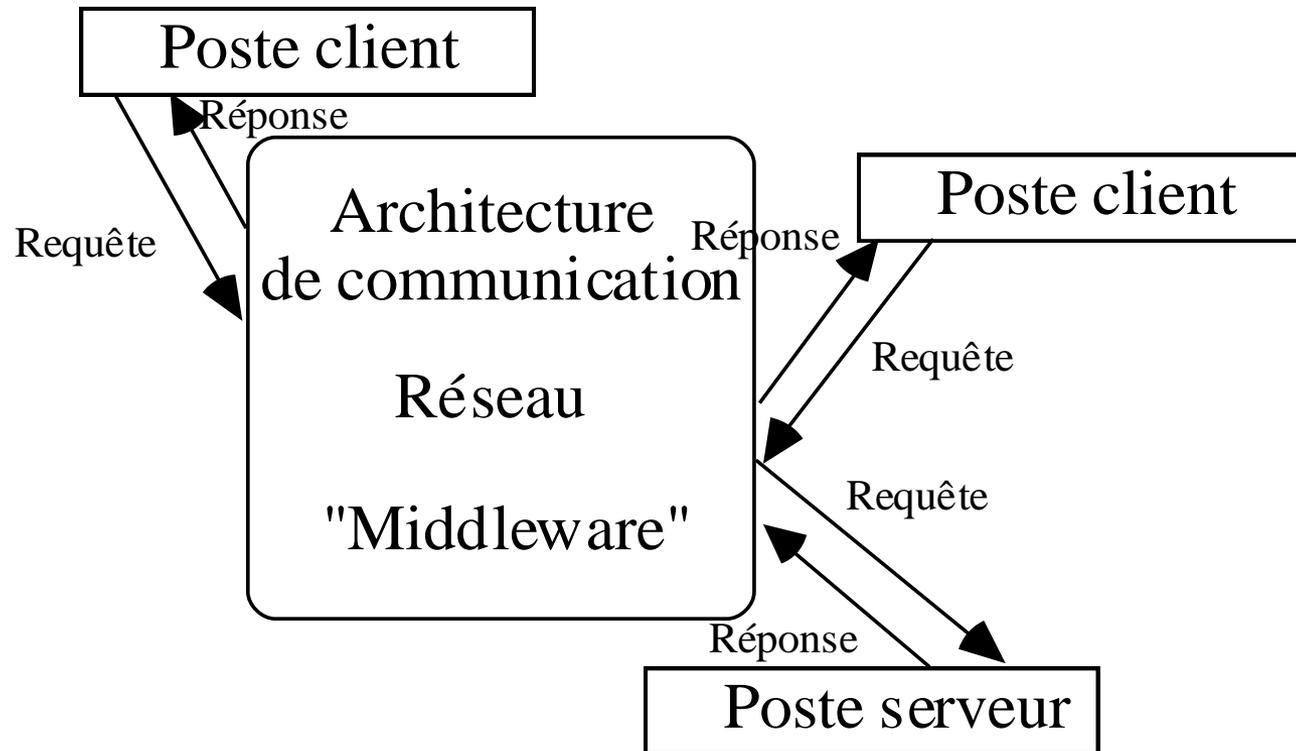
- **Le Middleware** assure les échanges de données entre un client et un serveur **en masquant les différents problèmes** potentiels liés à :
 - la répartition des données et traitements
(accès distant, baisse de performance)
 - l'hétérogénéité des matériels et des logiciels en opération.

LES SERVICES RÉSEAUX ("MIDDLEWARE")

Exemple de travail le plus fréquent en client/serveur : envoyer des requêtes d'accès à des données (type SQL) d'un client vers un serveur et recevoir les résultats :

- **Ouvrir une connexion** entre entités
- **Envoi de requêtes SQL** vers le serveur
- **Conversion des formats** de requêtes
- **Exécution** des requêtes
- **Envoi des résultats**
- **Conversion des résultats**
- (Gestion des erreurs)
- **Fermeture de connexion.**

CLIENT/SERVEUR : SCHÉMA DE PRINCIPE



CLIENT/SERVEUR: APPLICATIONS TYPES.

○ Transactionnel

- OLTP « On Line Transaction Processing »
- Les applications comportent :
 - des opérations d'accès en lecture écriture à des bases de données,
 - des traitements
 - des affichages en mode graphique sur des postes de travail.

CLIENT/SERVEUR: APPLICATIONS TYPES.

○ Aide à la décision :

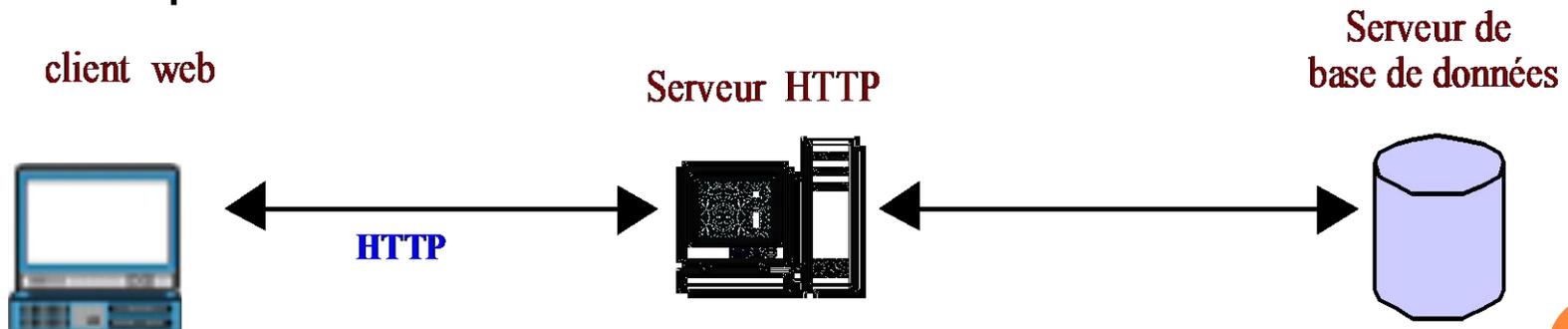
- EIS/DS « Executive Information System/Decision Support »
- Les applications comportent:
 - des accès complexes bases de données,
 - des traitements de synthèse/valorisation
 - des affichages (tableaux, histogrammes, etc)

INCONVÉNIENTS DU CLIENT/SERVEUR

- Coût d'administration élevé :
 - Les applications doivent être redéployées à chaque modification
 - Augmentation des coûts de support au niveau client à cause de la logique du « fat client »
 - Systèmes d'exploitation clients multiples et incompatibles
 - Engorgement du réseau dû aux accès bases de données
 - Les applications client-serveur montrent des faiblesses quand l'échelle monte : engorgement du serveur

LES ARCHITECTURES MULTI-TIERS

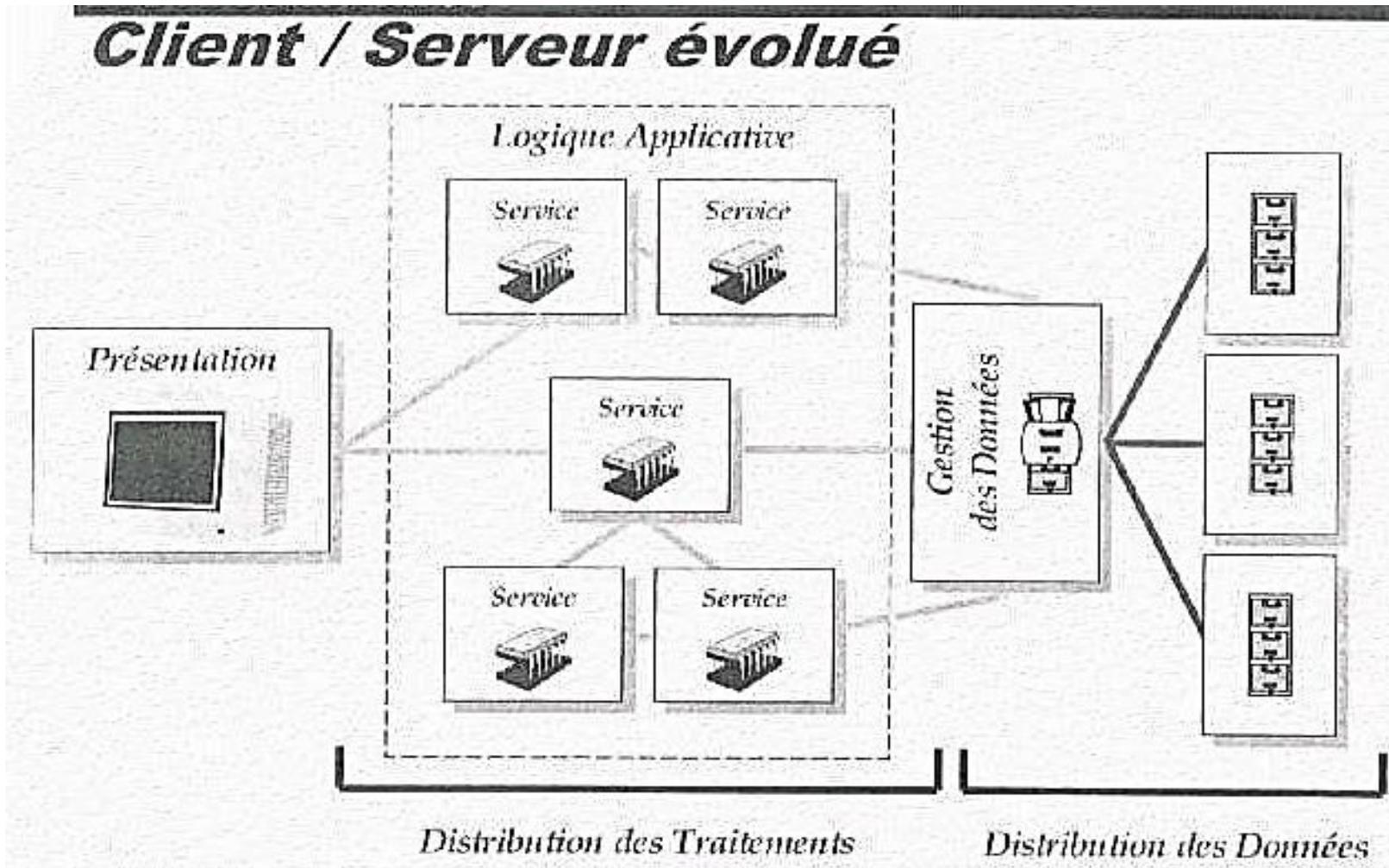
- Tier= niveau
- Le client serveur à 3 niveaux (3-tiers)
- On parle de Multi-tiers (ou « n-tiers ») lorsque les entités applicatives sont réparties sur plusieurs serveurs avec un middleware permettant la coopération des composants.
- Exemple :



ARCHITECTURE MULTI-TIERS

- Permet une distribution généralisée des composants.
- Permet une prise en compte des effets d'échelle par multiplication des plateformes de traitements
- Permet de minimiser les ressources au niveau client d'où :
 - Un coût d'administration moins élevé
 - L'utilisation de clients légers (Navigateur + applets Java)

ARCHITECTURE PAR COMPOSANTS



PROBLÈMES POSÉS PAR LE MULTI-TIERS

- Problème de la localisation :
 - Lorsqu'on répartit les traitements sur plusieurs serveurs, se pose le problème de l'adressage et de la localisation des composants
- Problème de l'hétérogénéité des plates-formes et des services
- ➔ Nécessité de définir des normes et des standards de coopération (CORBA, COM...)

LES SYSTÈMES D'OBJETS DISTRIBUÉS

- Une structuration des applications qui place la notion d'objet au centre de la conception : performante et de plus en plus utilisée.
- Objet : **associe les traitements et les données** en rendant les codes moins coûteux à développer, à maintenir, à réutiliser.
- **Abstraction** : Comportements communs => Typage.
- **Encapsulation** : Association des actions aux données.
- **Héritage** : Réutilisation du code.
- **Polymorphisme** : Réutilisation des références.

INTÉGRATION DE LA NOTION D'OBJET AU SYSTÈME RÉPARTI : LA DISTRIBUTION

- Les communications réseau offrent la possibilité d'implanter les objets de façon homogène sur les sites d'un système réparti.
- L'interface du système d'objets répartis est un ensemble de primitives ou un langage objet concurrent et réparti offrant la possibilité **d'implanter et d'exécuter à la demande** des objets (données, traitement) sur des sites quelconques.
 - **Création à distance** d'instances
 - **Exécution à distance** des méthodes.
 - **Concurrence / synchronisation.**
 - Autres fonctions : transactionnel, sécurité, optimisation du placement ...

POUR FAIRE QUOI?

○ L'objectif :

- Décomposer une application complexe en un ensemble de processus coopérant les uns avec les autres, ces processus pouvant être situés sur des machines différentes.
- Avoir une structure ouverte, où des machines peuvent se connecter pendant l'exécution d'une application.

○ Solution :

- **Face à la complexité du développement de tels systèmes et afin de les structurer, deux approches ont été généralement retenues.**

COMMENT LE FAIRE ?

○ Approche appliquée :

- Appliquer les concepts de la programmation par objets (encapsulation, héritage..)
- Décomposer le système réparti en bibliothèque de classes Ex : Services CORBA, COM+

○ Approche intégration :

- Intégrer les notions de répartition à l'objet
 - Objets à distances : Invocation d'objets à distance (bus Corba, bus Com+)
 - Objets mobiles : Copies de l'objet serveur chez le client (ActiveX, Java)
 - Objets synchronisés

UN ENJEU POUR LA NORMALISATION

- **Standardiser l'offre systèmes d'objets répartis pour intégrer des systèmes d'origine différentes dans une application :**
 - Normalisation minimum habituelle par les **protocoles** (standardiser les interactions) :
 - interopérabilité entre des implantations
 - unifier les techniques d'appel distant
 - accéder à des annuaires communs d'objets
 - Puis standardiser les **langages et techniques de développement...**
 - portabilité des applications
- **OMG** "Object Management Group"
- **CORBA** "Common Object Request Broker Architecture »
- **Microsoft OLE/DCOM** « Object Link Embedding / Distributed Component Object Model »

MERCI DE VOTRE ATTENTION

- Des questions ?

- erik.boucherdecrevecoeur@lecnam.net