



Global Knowledge®

La place du développement dans le  
projet informatique

# PRESSENTATION

- Qui suis-je ?



# Objectifs de la formation

- Appréhender le vocabulaire, les modèles, les technologies, les outils et frameworks utiles à tout concepteur développeur
- Identifier les composants d'une application informatique, les architectures possibles d'exécution des applications les apports des technologies dans le cycle de vie d'un projet
- Positionner la conception, la réalisation et les tests dans le développement applicatif pour mieux comprendre leurs intérêts et leurs évolutions
- Se rendre compte de l'intérêt de l'industrialisation de la production de logiciels pour l'amélioration continue de la qualité des applications

# PRESSENTATION

- Qui êtes-vous?
- Quelles sont vos attentes ?



# SOMMAIRE

---

- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# INTRODUCTION AU SYSTEME D'INFORMATION

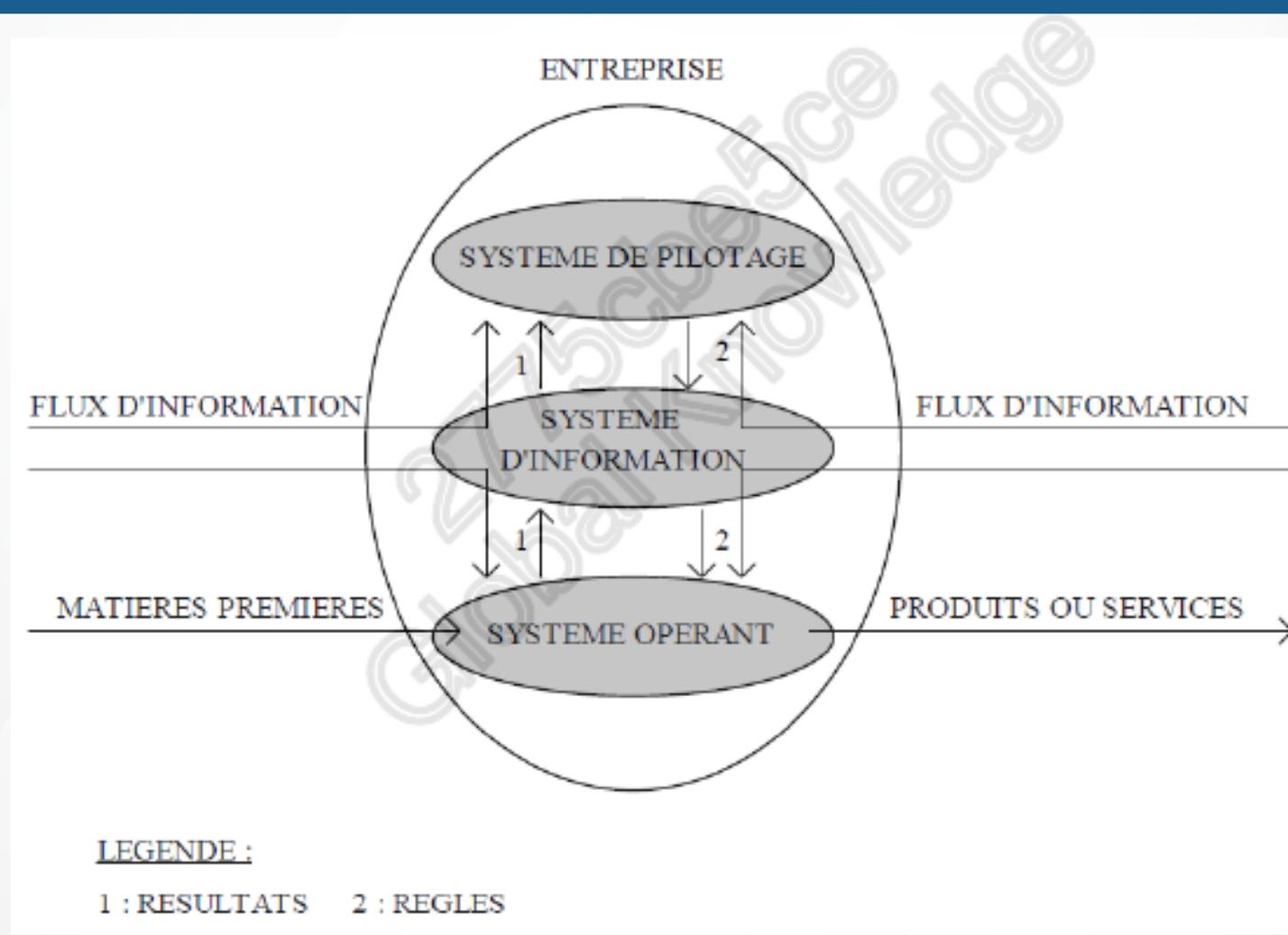
Objectifs : *Situer le cadre de la formation et définir le SI, l'informatique, la DSI*



# LE SYSTEME D'INFORMATION - SI

- L'entreprise peut être représentée par un ensemble de trois systèmes :
  - un système opérant (ou logistique),
  - un système de pilotage (ou de direction, de gestion, de décision),
  - un système d'information.
- Le SI peut être lui-même représenté comme un ensemble comprenant :
  - une base d'information,
  - un processus d'information composé de deux moniteurs,
    - l'un qui gère le stockage des données,
    - l'autre qui gère la circulation de l'information.

# LE SYSTEME D'INFORMATION



# LE SYSTEME D'INFORMATION

- Le SI est le **véhicule de la communication**. Possède un langage dont **les mots sont les données**. Peut être décrit via 5 composantes :
  - **Humaine informelle** univers du discours, communications informelles entre individus,
  - **Humaine formelle** communications formelles imposées par l'organisation et les règles qui en découlent,
  - **Informatique formelle** la traduction informatisée de la partie informatisable du SI,
  - **Informatique informelle** informatique individuelle qui n'observe pas de règles précises au sein de l'organisation,
  - **Externe** à l'entreprise qui est celle constituée par l'environnement et qui peut être formelle et informelle.

# LE SYSTEME D'INFORMATION



- Un système informatisé n'est donc qu'une composante du système d'information.
  - Socrate (SNCF),
  - les distributeurs automatiques de billets (DAB)
  - autres guichets automatisés bancaires (GAB),
  - les applications internet grand public
- 
- ne sont donc que des applications informatisées d'un système d'information d'entreprise beaucoup plus large.

# L'INFORMATIQUE

- « Science de l'information ».« Ensemble des techniques de la **collecte**, du **tri**, de la **mise en mémoire**, de la **transmission** et de l'**utilisation** des informations traitées automatiquement à l'aide de programmes mis en oeuvre sur des ordinateurs »
- L'informatique nécessite
  - des Hommes,
  - des machines (hardware),
  - des logiciels (software),
  - des méthodes.



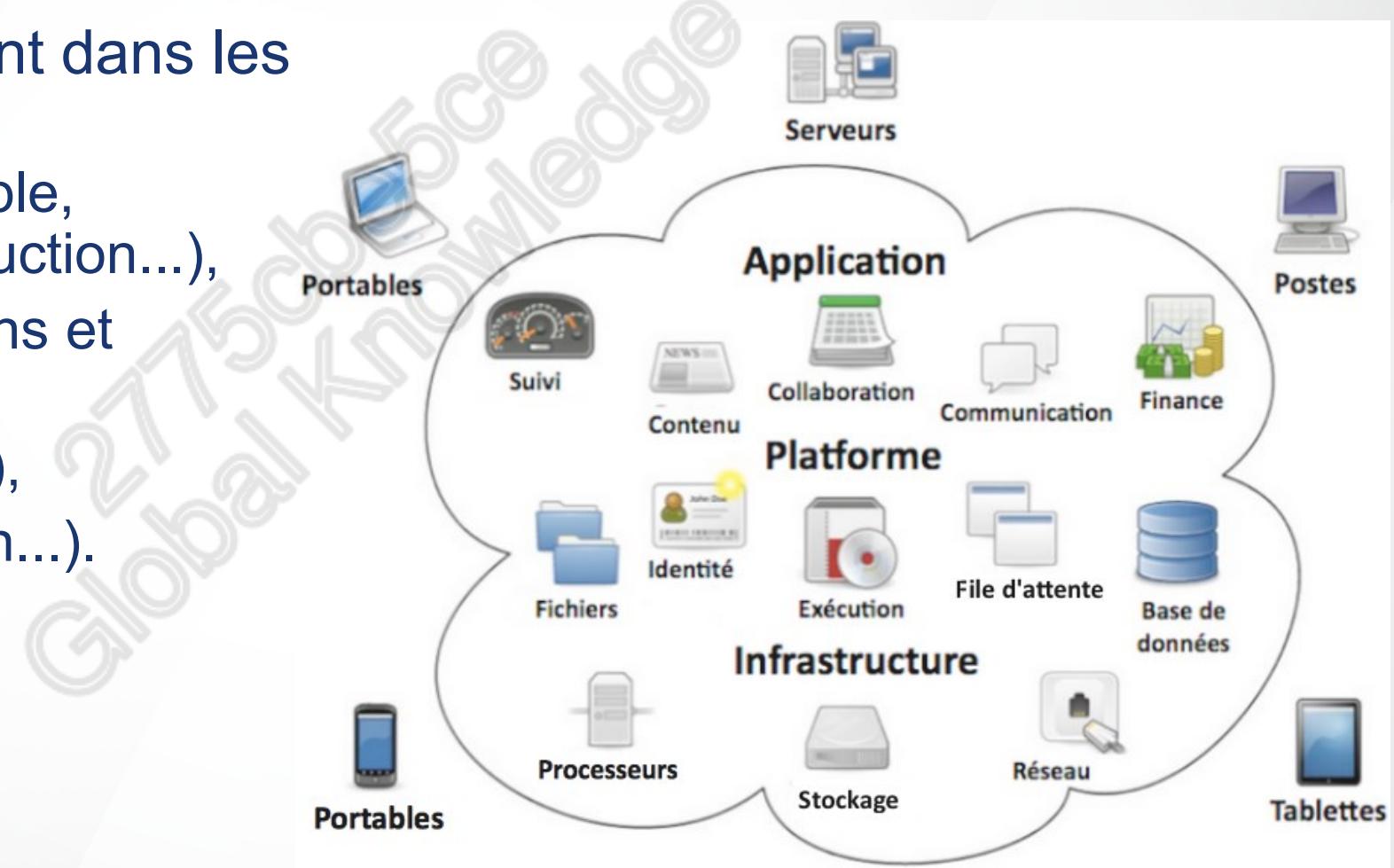
I  
N



O  
U  
T

# L'INFORMATIQUE

- L'informatique intervient dans les domaines :
  - de la gestion (comptable, commerciale, de production...),
  - des télécommunications et réseaux,
  - industriel (robotique...),
  - scientifique (simulation...).
  - ...



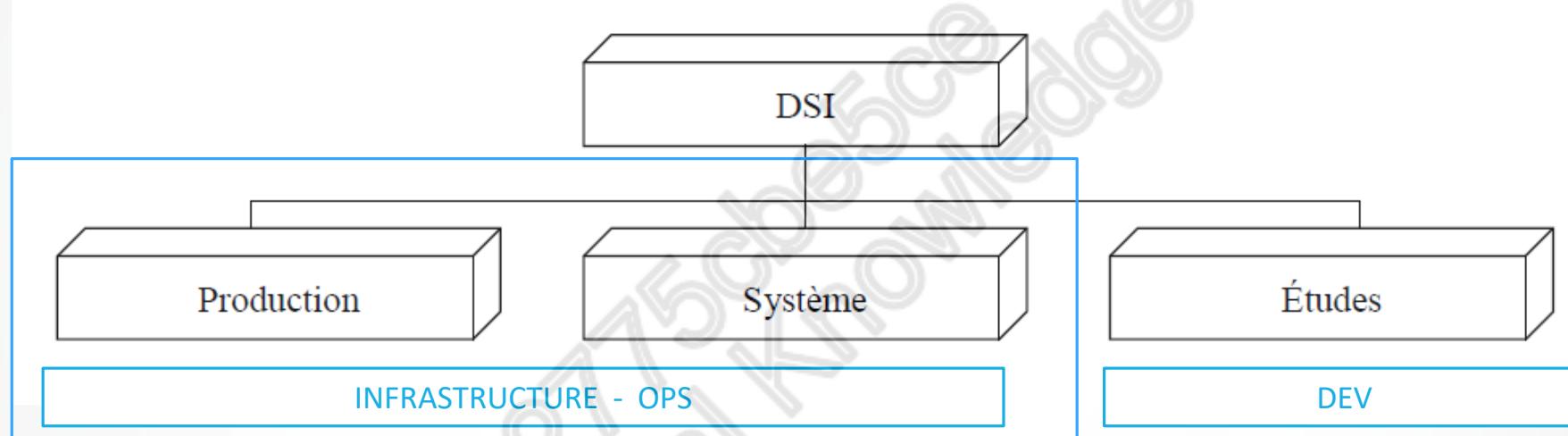
# LA DIRECTION DES SYSTEMES D'INFORMATION

- La DSI, remplit 3 grands rôles, correspondant, dans la majorité des entreprises, à 2 ou 3 services différents :
  - **la Production (ou Exploitation) :**
    - en charge du bon fonctionnement quotidien du système d'information, tant d'un point de vue matériel qu'applicatif,
  - **le Système,**
    - dont le rôle est de mettre en œuvre les évolutions nécessaires afférentes au matériel et aux logiciels systèmes, mais aussi de résoudre les pannes et les dysfonctionnements pouvant survenir, en relation avec les fournisseurs concernés,
  - **les Études ou Développement :**
    - en charge du développement des nouveaux applicatifs et de la maintenance des applicatifs existants.

O  
P  
E  
R  
A  
T  
I  
O  
N  
S

D  
e  
v

# LA DIRECTION DES SYSTEMES D'INFORMATION



- Production et Système, peuvent former un seul service, que l'on nomme l'infrastructure, les Ops (opérations), distinct de la mise en oeuvre d'applications pour les utilisateurs du SI, service nommé aussi les Devs (développeurs).

**DEVOPS**  
Philosophie  
pour  
l'alignement  
des équipes

# SOMMAIRE

---

- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# L'INFRASTRUCTURE

Objectifs : *Présenter les concepts des data centers, des systèmes d'exploitation, des réseaux et de la virtualisation pour comprendre le fonctionnement de l'infrastructure*



# DEFINITION de l'infrastructure



- L'infrastructure désigne l'ensemble des éléments de type matériel et les logiciels composant le système informatique d'une entreprise ou d'une organisation.

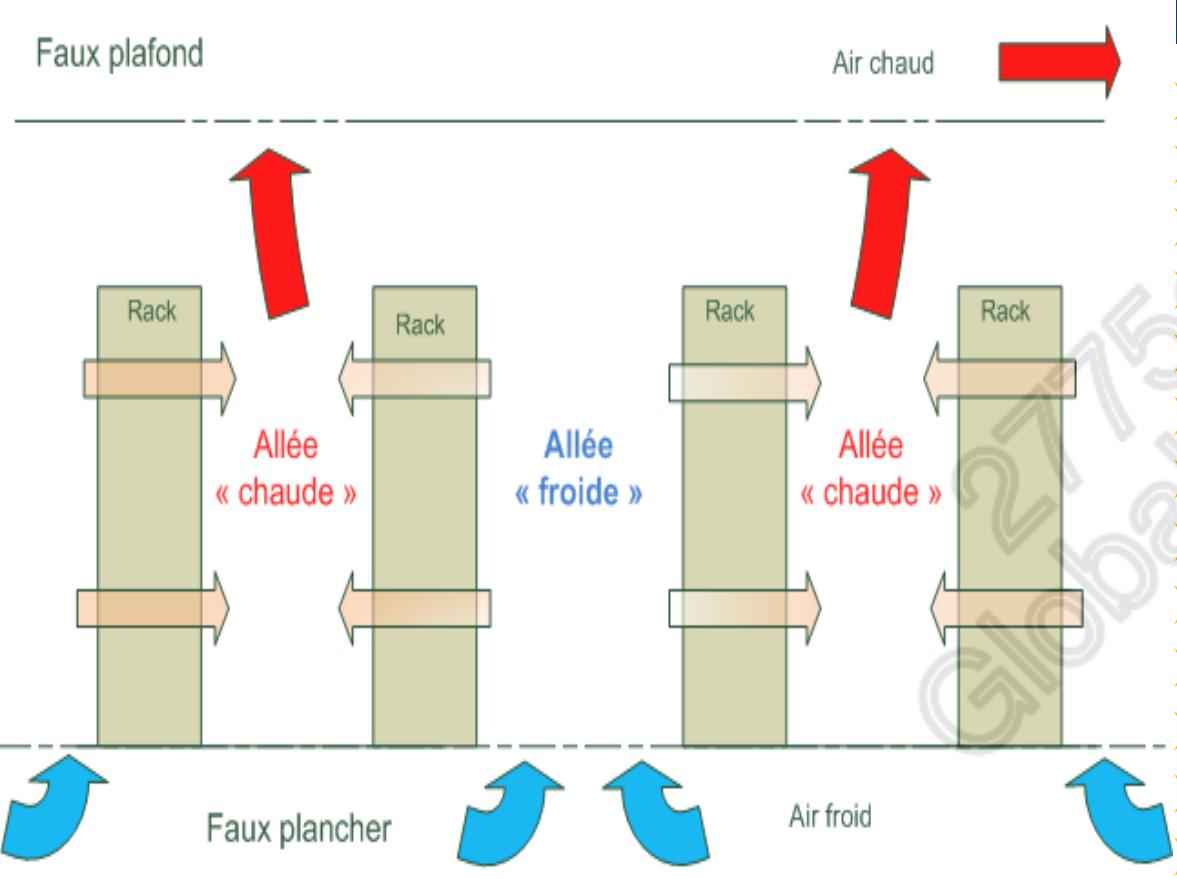
# DATA CENTER



## Centre de traitements des données

- Local, site où sont stockés les serveurs, construits en baie, en Rack, en U
- **Hardware** : UC, disques, réseau

# DATA CENTER - Equipements



## Installation des baies en face à face

- Climatisation
- Contrôle de la poussière (filtration de l'air)
- Unité de distribution de l'énergie
- Bloc d'alimentation d'urgence et une unité de secours
- Câbles de cuivre en Ethernet pour liaisons inter
- Fibres optiques pour liaisons inter-sites ou inter
- Conduites pour câbles au-dessus/dessous du plancher
- Système d'alerte d'incendie
- Extinction automatique des incendies
- Surveillance par caméras en circuit fermé
- Contrôle des accès, sécurité physique du bâtiment
- Surveillance 24h/24 et 7j/7 des serveurs
- Service de sécurité continuellement présent

# DATA CENTER – Hébergements et Tier

**Hébergeur = Gestion externalisée des serveurs**

**Choix en fonction :**

- De la capacité des serveurs Puissance Espace de stockage Bande passante
- De la sécurité → **Classification** faite par Uptime Institute : 4 Tiers

I - **basique** : Infrastructure non redondante. Maintenances et pannes génèrent l'arrêt du site.

II - **Redondance partielle** : Eléments de production de froid et d'électricité redondants, pas la distribution. Certaines maintenances et pannes génèrent un arrêt du site, sur les circuits de distribution.

III – **Maintenabilité** : Tous les composants sont maintenables sans arrêt de l'informatique. Maintenances n'arrêtent pas le site, erreurs humaines, certaines pannes peuvent entraîner l'arrêt.

IV : **Tolérance aux pannes**. Aucune panne n'arrête l'informatique. Absence de SPOF (Single Point of Failure). Tolérant aux maintenances, pannes (uniques) et incident même graves (incendie par exemple).

# LE SYSTÈME D'EXPLOITATION - OS

- « Programme, pouvant être livré par le constructeur de l'ordinateur, permettant à un utilisateur d'accéder aux ressources de la machine. »
- Le système d'exploitation assure :
  - la gestion des périphériques,
  - la gestion de la mémoire,
  - la gestion des tâches
    - (exécutions de programmes),
  - la gestion des anomalies.



# LE SYSTÈME D'EXPLOITATION - OS

- C'est un **logiciel** permettant d'utiliser (exploiter) un système informatique.
- Il a 5 rôles : ( 5A)
  - Aide Simplifier accès aux ressources
  - Abstraction des périphériques
  - Augmentation Etendre les ressources
  - Arbitrage de l'accès aux ressources
  - Autorisation limiter l'accès



# Types de système d'exploitation : Systèmes propriétaires, ouverts, libres

- OS **propriétaire** (legacy system) est un système dépendant entièrement de l'architecture matérielle de la machine pour laquelle il a été conçu.
- OS **ouvert** (open system) est un système d'exploitation conçu pour pouvoir dialoguer avec d'autres systèmes, même hétérogènes.
  - 1<sup>er</sup> système ouvert : **UNIX**  noms commerciaux différents
- OS **libre**, est un système qui n'a pas été développé dans un but commercial chacun est libre de l'utiliser et de le modifier.
  - Cas de **Linux**, qui initialement a été conçu (années 90 en utilisant une version d'Unix) et est aujourd'hui présent sur de nombreuses plateformes.
- Aujourd'hui, la plupart des constructeurs d'OS dit propriétaire (sauf Apple) ont ouvert leurs systèmes.

# Types de système d'exploitation : Systèmes propriétaires, ouverts, libres

- **BULL** : GCOS 8 sur grands systèmes DPS 8 et DPS 9000,GCOS 7 sur minisystèmes DPS 7 et DPS 7000,IX sur minisystèmes Escalla (UNIX). 
- **IBM** : z/OS (MVS) sur grands systèmes; i, (z/Series, OS/400) sur Power Systems, AIX sur System p (machines RS/6000) est un système UNIX.
- **HP** : MPE sur mini systèmes HP 3000,HP-UX sur minisystèmes HP 9000 (UNIX).
- **Sun Microsystems** : Sun-Solaris (UNIX).
- **MICROSOFT** : Commercialise Windows et Windows Server.
- **APPLE** : Commercialise les systèmes Mac OS X (Leopard, Snow Leopard, Lion, Mountain Lion).
- **Linux** : L'assemblage de Linux pour fournir un système clé en main est appelé une distribution, il existe différentes distributions : RedHat, SuSe,Debian, Ubuntu, Knoppix, Mandriva, Slackware, Gentoo...

# Types de système d'exploitation : Systèmes embarqués



- Les systèmes embarqués sont des systèmes d'exploitation prévus pour fonctionner sur des **machines de petites tailles** : des appareils électroniques autonomes (sondes spatiales, robot, ordinateur de bord de véhicule, GPS, etc.), possédant une **autonomie réduite**.
- Ainsi, une caractéristique essentielle des systèmes embarqués est leur gestion avancée de l'énergie et leur capacité à fonctionner avec des ressources limitées.
- Ces systèmes d'exploitation sont le plus souvent propriétaires.



# Types de système d'exploitation : Systèmes embarqués

- Les principaux systèmes d'exploitation des smartphones et tablettes :
  - iOS (Apple) sur iPhone et iPad
  - Android (Google) sur téléphones et tablettes Samsung (gamme Galaxy), HTC, Sony (gamme Xperia)...
  - Windows Phone (Microsoft) sur téléphones Nokia (gamme Lumia), HTC... Microsoft commercialise sa propre tablette sous le nom de Surface.
  - BBOS (Blackberry)



# Types de système d'exploitation : Systèmes embarqués

- Le marché des smartphones et des tablettes est aujourd'hui en très forte évolution. De nouveaux produits arrivent tous les 6 mois, les nouvelles versions des systèmes d'exploitation suivent quasiment le même rythme.
- Côté parts de marché, Apple et Samsung se partagent plus des trois quarts du gâteau.
- Blackberry et Microsoft tentent une conquête du marché. Windows tablant sur la création d'un écosystème où PC, tablette et téléphone sont interconnectés et partagent la même interface utilisateur...
- A suivre...

# RESEAU : Types de réseau

- La connexion de deux ordinateurs nécessite la présence d'un réseau. Il peut s'agir :
  - d'un **réseau local** (LAN, *Local Area Network*), lorsqu'il est localisé au sein d'un bâtiment,
  - d'un **réseau étendu** (WAN, *Wide Area Network*), lorsqu'il relie des ordinateurs situés dans des bâtiments distincts.
  - de **VPN** , (*Virtual Private Network*), réseau d'entreprise privé, utilisant Internet comme support de transmission, en permettant d'envoyer des données sécurisées.
- Jusqu'au 1er janvier 1998, les réseaux étendus passaient uniquement par le biais de France Télécom – Orange, aujourd'hui sur le territoire national d'autres opérateurs sont apparus.

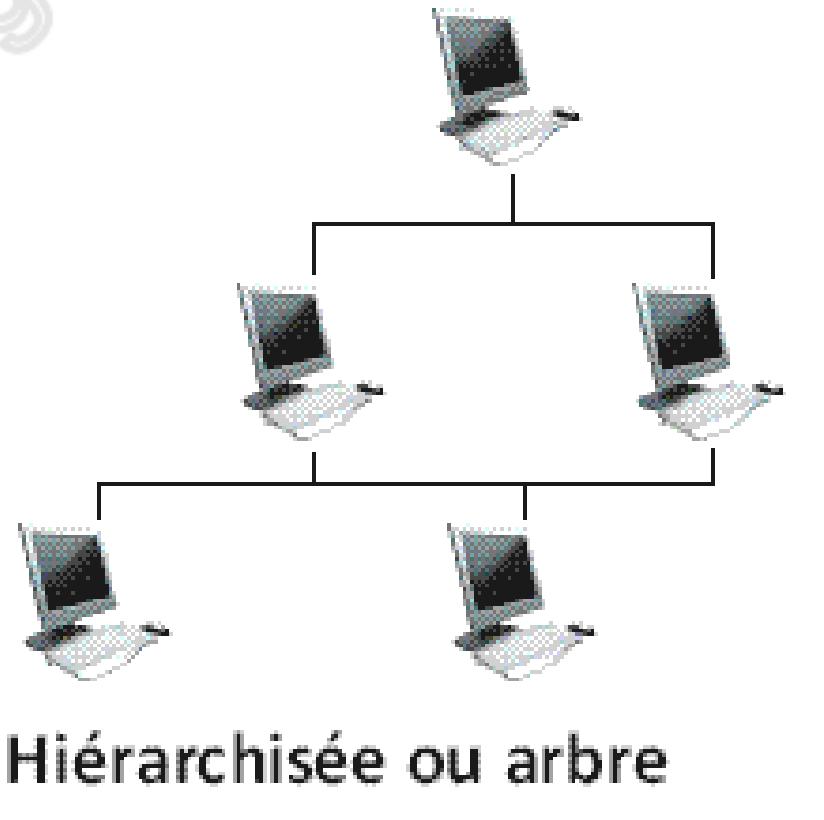
# RESEAU : Types de réseau

- La répartition des machines **dans un réseau local** se fait selon 5 configurations possibles, on parle de **topologie** de réseaux
- Il existe 2 modes de propagation classant les topologies :
  - Le mode diffusion, il n'utilise qu'un seul support de transmission. Le message est envoyé à tous, chaque destinataire l'analyse pour savoir si le message lui est adressé. (Cas des topologies bus et anneau).
  - Le mode point à point, le support de transmission est relié à deux unités, qui pour communiquer passent par un intermédiaire nommé le **noeud**. (Cas des topologies en étoile ou maillée).

# RESEAU : Types de réseau

## ➤ Topologie hiérarchique ou en arbre :

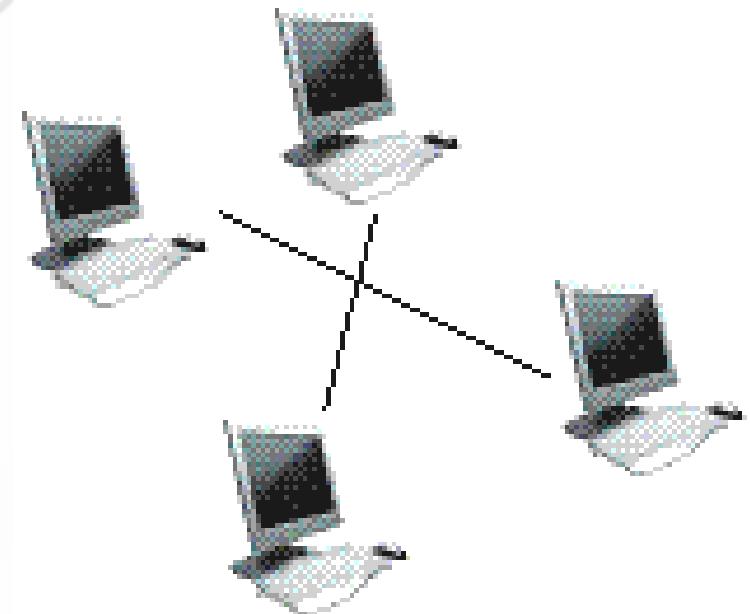
- La machine de haut niveau est connectée à des machines inférieures, nommées **branches**.
- Les machines inférieures pouvant être aussi connectées à d'autres niveaux inférieurs



# RESEAU : Types de réseau

## ➤ Topologie en étoile :

- Les machines sont connectées à un matériel central, nommé concentrateur ou **hub** en anglais.
- Il s'agit d'un boîtier, permettant de raccorder les câbles réseaux des machines, pour assurer la communication.

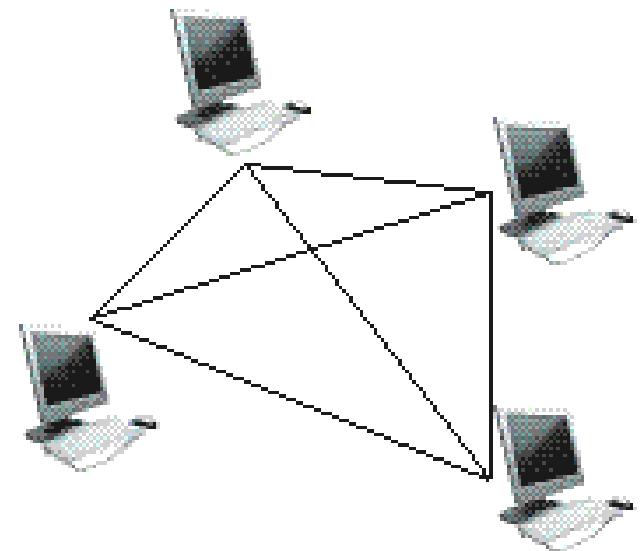


Etoile

# RESEAU : Types de réseau

## ➤ Topologie maillée :

- Il s'agit de plusieurs liaisons point à point, cas d'**Internet**.
- Chaque machine est reliée à toutes les autres.
- L'inconvénient est le nombre de liaisons nécessaire pour relier les machines entre elles.

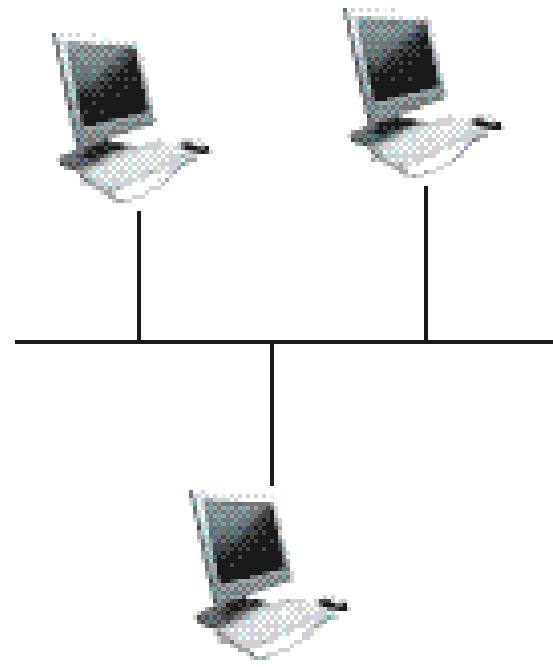


**Maillée**

# RESEAU : Types de réseau

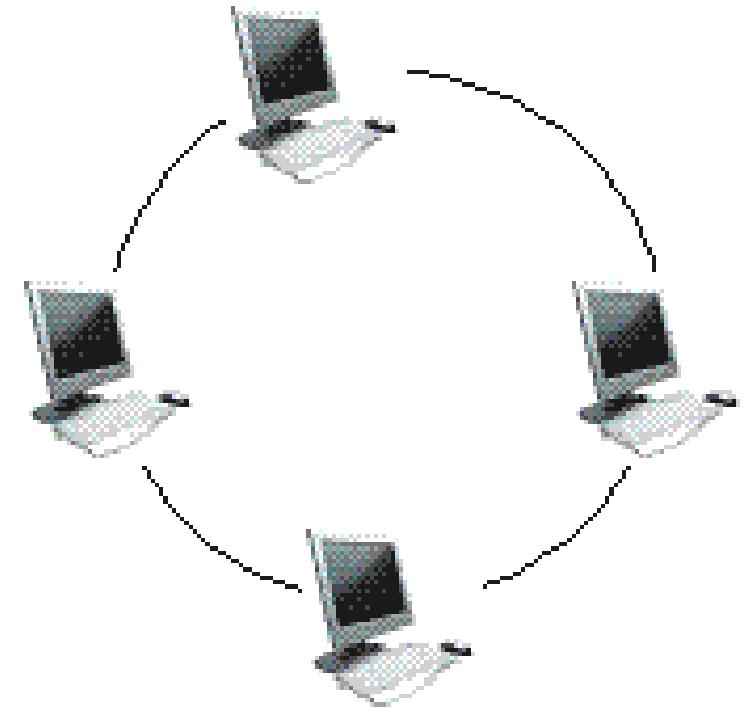
## ➤ Topologie en bus (topologie des réseaux)

- Cas la plus simple à mettre en œuvre.
- Chaque machine est reliée à la même ligne de transmission, que l'on nomme un bus
- L'inconvénient est que si une connexion est défectueuse, l'ensemble du réseau est affecté



# RESEAU : Types de réseau

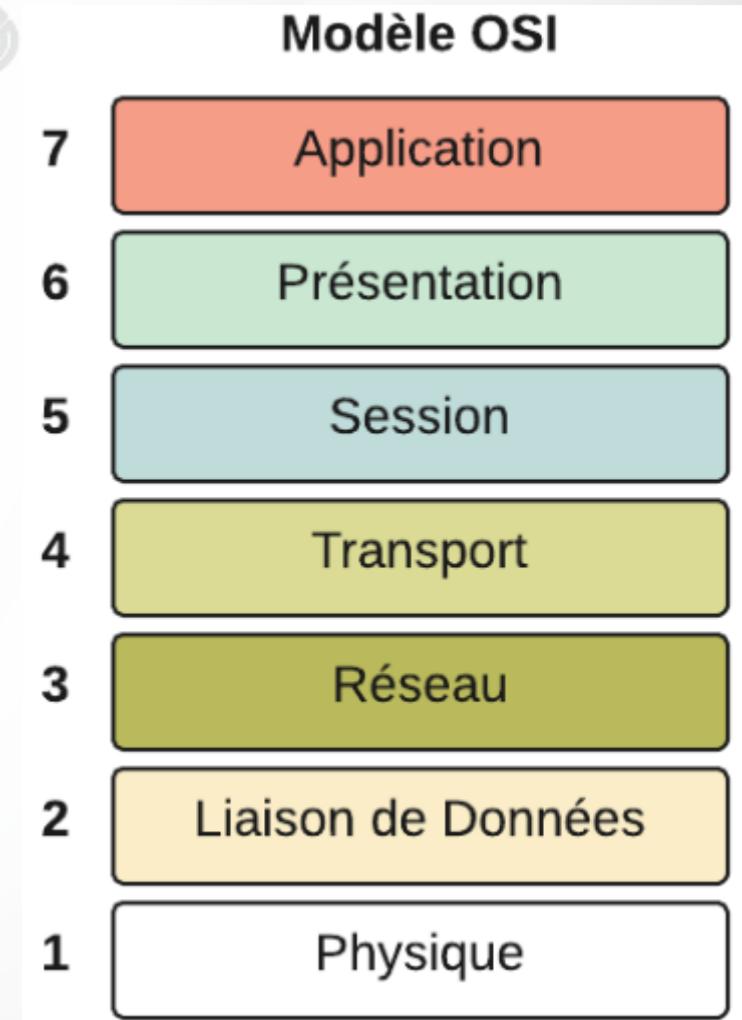
- Anneau à jeton :
- Les machines sont reliées à un **répartiteur** nommé MAU (Multistation Access Unit ), qui permet d'allouer un temps de parole à chaque machine,
- Elles communiquent donc à tour de rôle.



Anneau

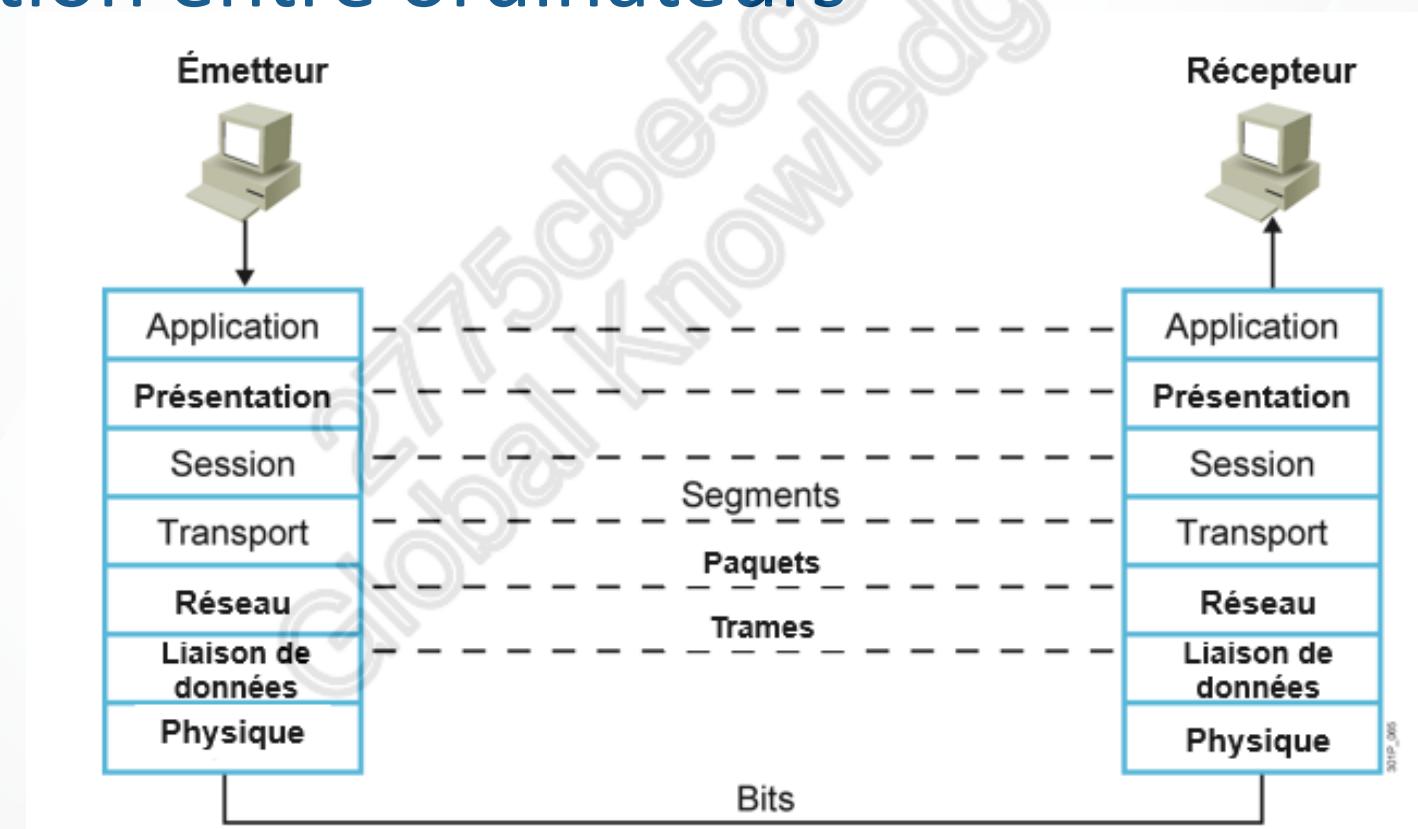
# Les protocoles réseau

- La communication entre ordinateurs hétérogènes n'est possible qu'à condition que chaque protagoniste **adopte un langage commun.**
- La traduction vers ce langage a été modélisée par l'ISO (International Standard Organization) dans le modèle OSI (Open System Interconnections) à 7 couches
- Un protocole réseau est un ensemble de règles régissant la transmission de données sur le réseau.



# Les protocoles réseau

## La communication entre ordinateurs

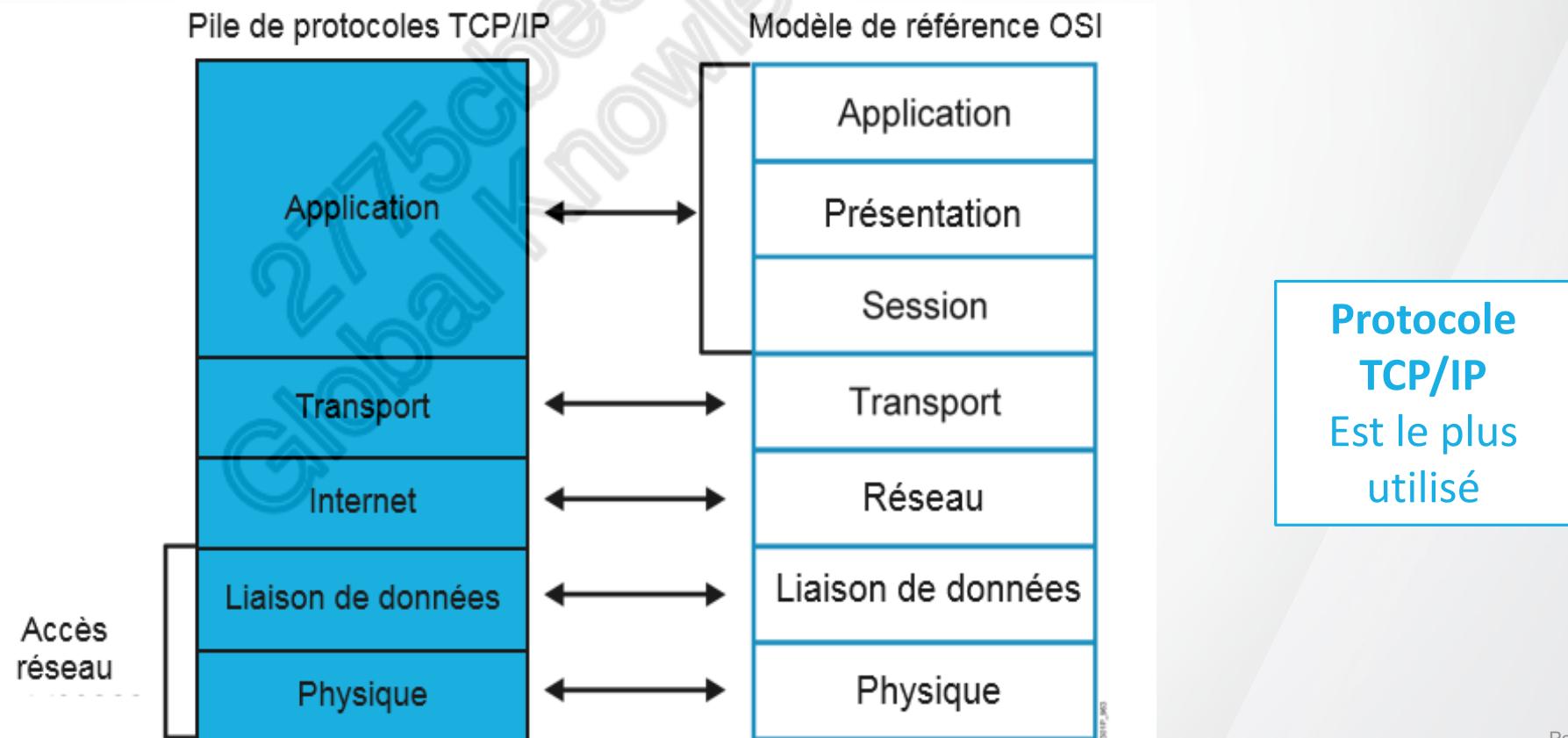


01101101 1101001

# Les protocoles réseau – TCP/IP

Association de 2 protocoles : Transmission Control Protocol et Internet Protocol.  
TCP/IP a été développée +ou- en même temps que l'OSI (organismes différents)

**TCP/IP**  
**4 couches**  
Parallèle avec  
OSI →  
Couche  
réseau  
découpé en 2  
(D'où pile à 5  
couches)



# Les équipements réseaux

- Un **hub**, ou un concentrateur, est un appareil permettant de brancher plusieurs appareils informatiques à un port d'un PC. (Equivaut à une multiprise).
- Un **switch**, ou un commutateur, effectue des opérations sur la couche 2 = couche liaison de l'OSI. C'est un boitier qui permet de relier plusieurs câbles au réseau, il dispose de plusieurs ports Ethernet. Le switch utilise les adresses MAC pour adresser plus rapidement les machines et éviter que chaque destinataire vérifie si la communication lui est adressée. Au niveau matériel le switch ressemble au hub.



# Les équipements réseaux

- Un **router**, effectue des opérations sur la **couche 3** = couche réseau de l'OSI.
  - C'est un élément intermédiaire du réseau qui permet de transmettre à un autre noeud les paquets.
  - Le router utilise les adresses IP pour diriger les données



# Les équipements réseaux

- Un **firewall**, permet de faire respecter la politique de sécurité du réseau.
  - Gère les connexions entrantes et sortantes de l'entreprise vers Internet. Cela peut-être un matériel informatique ou un logiciel.



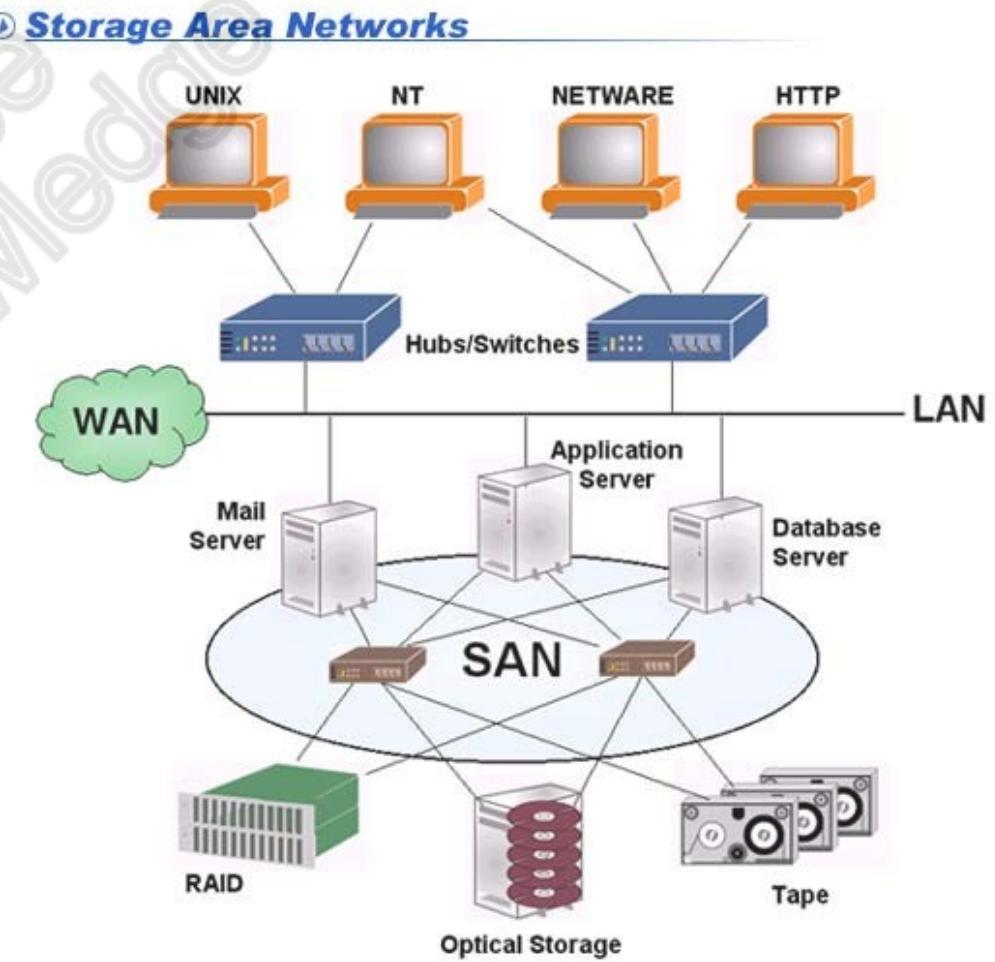
# Les équipements réseaux

- Un **NAS** (Network Attached Storage), est un équipement, relié au réseau dont la principale fonction est le stockage de données.
  - Cela permet de centraliser les sauvegardes de données venant de serveurs hétérogènes.
  - Les applications les plus souvent stockées sur des serveurs NAS sont les messageries d'entreprise et les serveurs web.



# Les équipements réseaux

- Un **SAN** (Storage Area Network) est un réseau de stockage.
  - Son but est de mutualiser les ressources de stockage.
    - (Mutualiser : c'est-à-dire utiliser un même équipement pour des applications différentes afin de partager les coûts).
    - Le NAS, peut être le point d'entrée de ce réseau, ou y être intégré.



# SYSTEME ET RESEAU

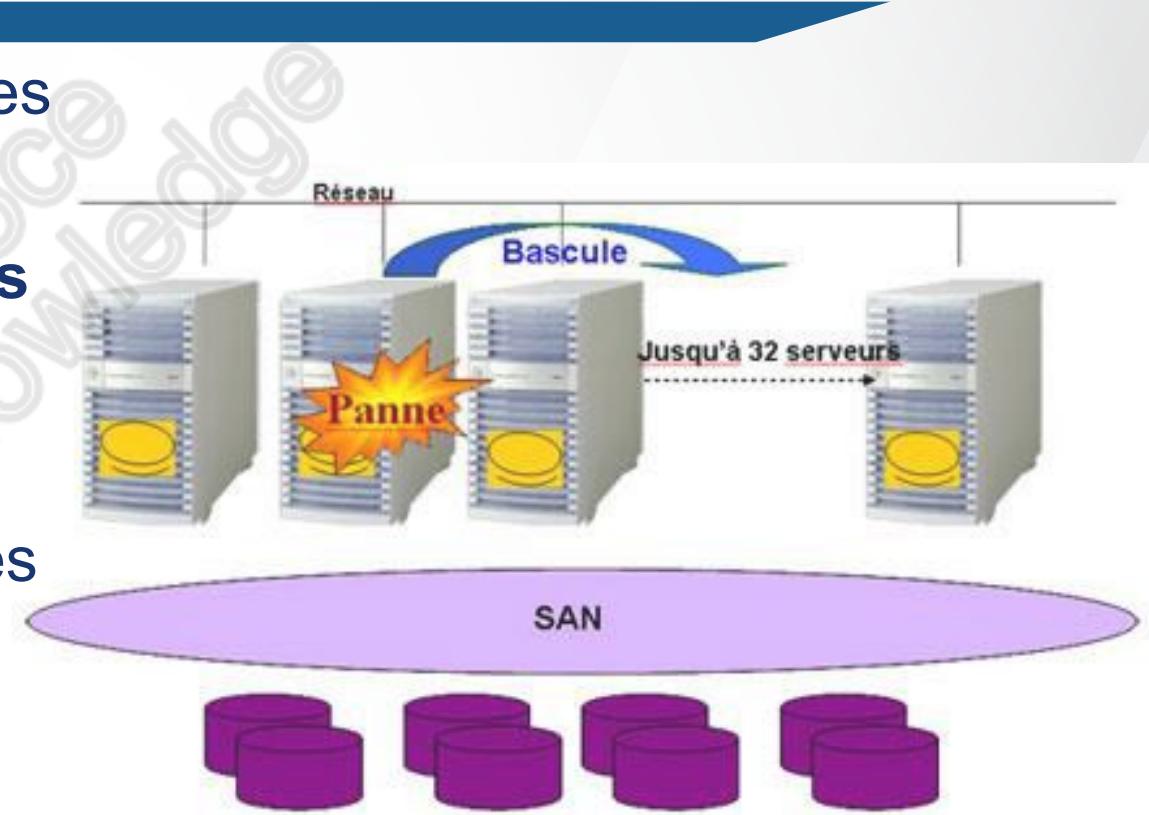
- Un **noeud** (node en anglais) est l'extrémité d'une connexion. Le noeud peut-être :
  - Un hub , ou un concentrateur,
  - Un switch, ou un commutateur
  - un routeur,
  - un serveur.

# SYSTEME ET RESEAU

- Un **cluster** est une **grappe** de serveurs (ou ferme de calcul) constitué de 2 serveurs au minimum que l'on appelle dans ce cas des noeuds.
- Le cluster partage une « baie de disques » commune aux différents serveurs du cluster, cette baie peut permettre :
  - d'assurer la disponibilité des services,
  - de répartir la charge de calcul,
  - de répartir la charge du réseau.

# SYSTEME ET RESEAU

- Le cluster permet une **gestion centralisée** des ressources des serveurs (processeurs, mémoire, disque dur, bande passante sur le réseau) et permet de **multiplier les capacités** des machines.
- En cas de panne d'un serveur, le logiciel de *clustering* est capable de transférer l'information du serveur défaillant vers d'autres serveurs de la grappe.
  - Cette mise en œuvre de disques partagés se retrouve dans le cadre d'un réseau SAN.
- Le cluster est un ensemble de machines homogènes et localisées, organisées en grappe

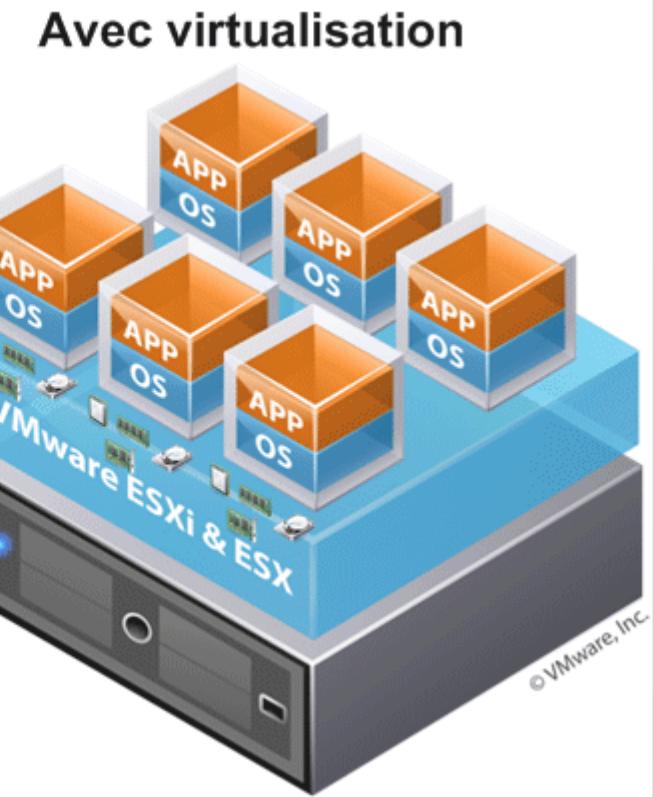
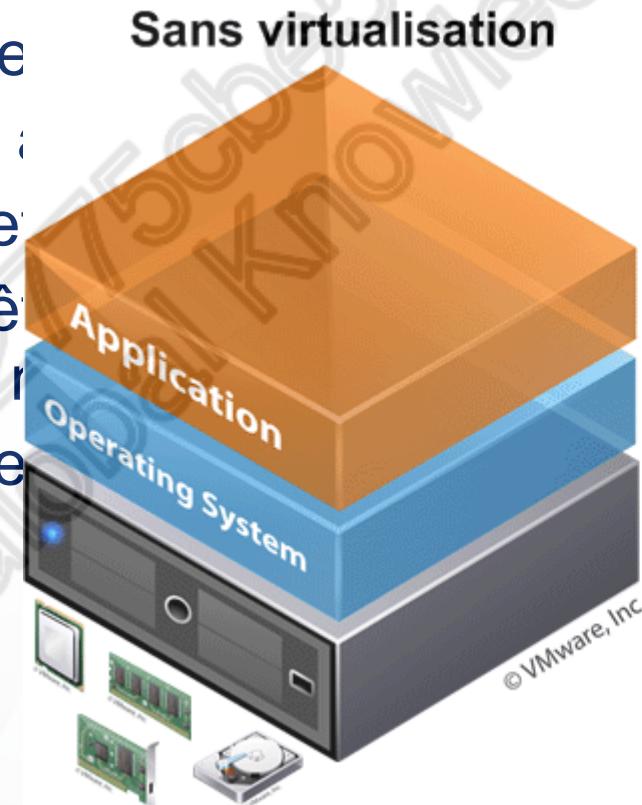


# LA VIRTUALISATION : Objectifs

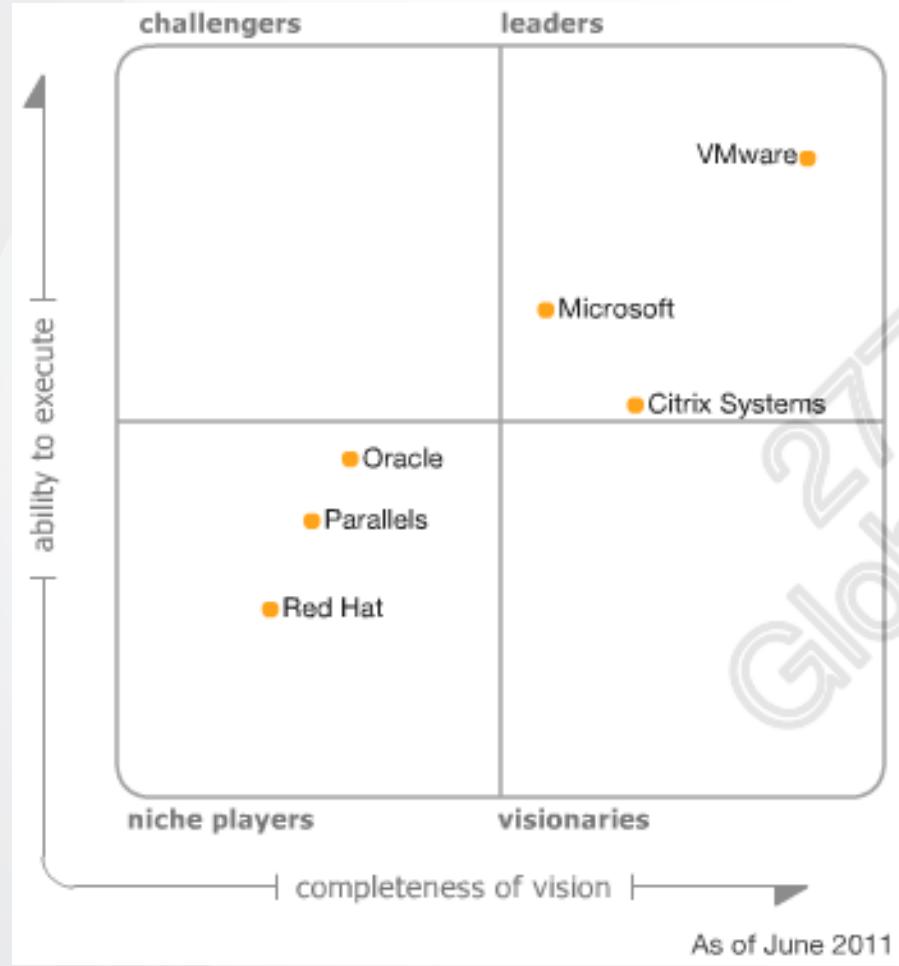
- La virtualisation s'applique principalement à deux grandes familles :
  - le stockage d'une part (disques et bandes magnétiques),
  - les serveurs d'autre part.
- Le but de la virtualisation de serveurs et de faire fonctionner sur un même ordinateur plusieurs systèmes d'exploitation, comme s'ils fonctionnaient sur des ordinateurs différents.
- Dans l'entreprise, il existe souvent plusieurs machines qui ne sont pas utilisées à 100% de leur capacité.
  - L'idée est de regrouper plusieurs serveurs sur une même machine afin d'utiliser le maximum de la capacité.
- La virtualisation simplifie les tâches d'administration, de gestion, de réPLICATION, de copie d'informations.

# LA VIRTUALISATION : Notions prises en charge

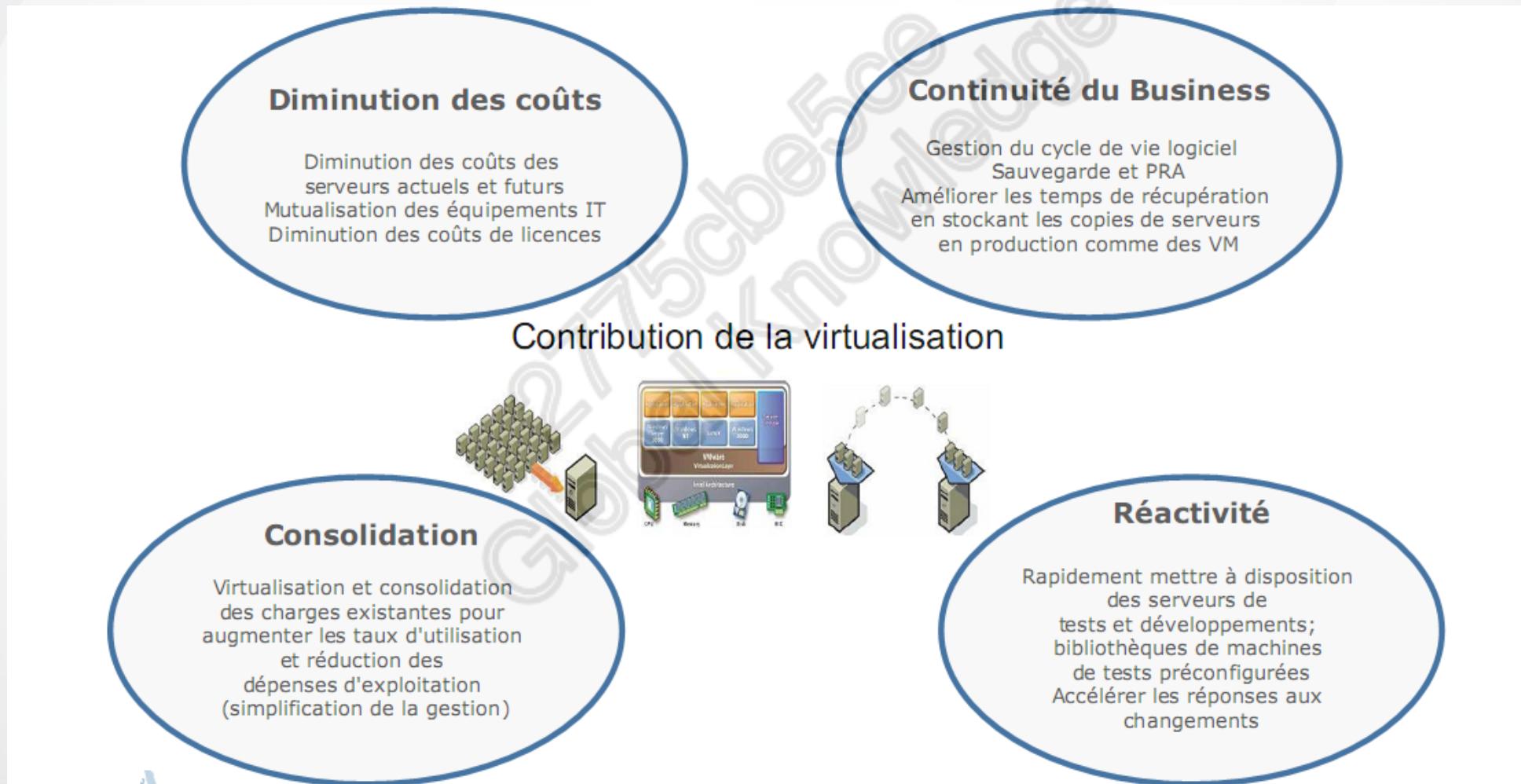
- La virtualisation prend en charge les notions suivantes :
  - Abstraction matérielle/logicielle.
  - Système d'exploitation hôte
  - Système d'exploitation (ou invité)
  - Partitionnement, isolation et sécurité
  - Images manipulables : arrêt, sauvegarde/restauration – redémarrage
  - Réseau virtuel : le réseau est partagé entre les invités.



# LA VIRTUALISATION : Offres



# LA VIRTUALISATION : Enjeux



# SOMMAIRE

---

- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# LES ARCHITECTURES DISTRIBUEES

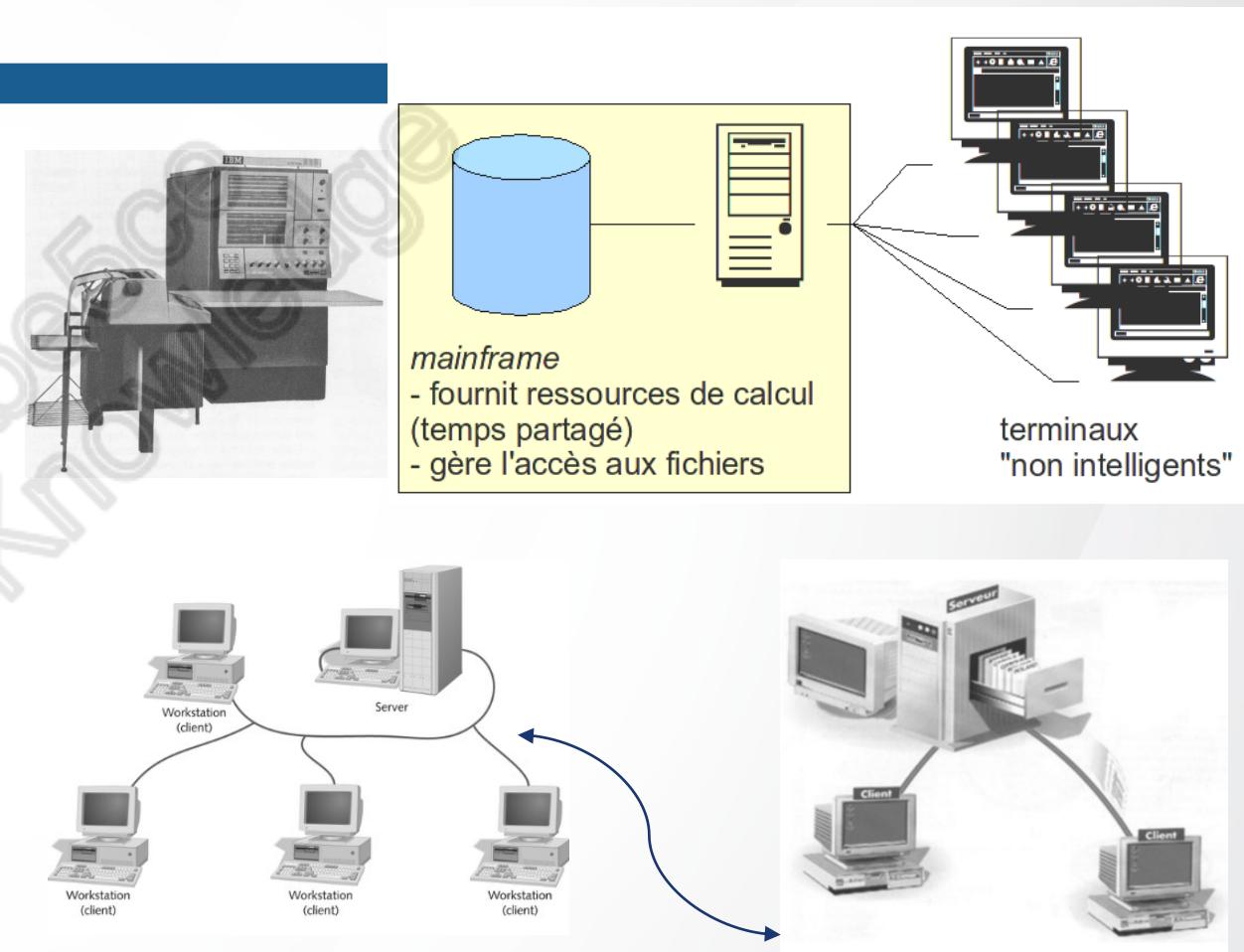
Objectifs : *Décrire les différents types d'architecture et lister les enjeux de chaque architecture.*



# EVOLUTION DES ARCHITECTURES

## Avant les années 90

- **L'informatique centralisée**
  - Terminaux passifs Ordinateur central
  - Solution « propriétaire »
- **L'informatique décentralisée**
  - l'interconnexion des différents ordinateurs
  - Maîtres/Eclaves, échanges de fichiers
- **Micro informatique isolée**
  - Arrivée des IHM graphiques - Bureautique
- **Micro informatique intégrée**
  - L'utilisateur n'a qu'1 pc :
    - Émulation vers le site central
    - Bureautique en local



# EVOLUTION DES ARCHITECTURES :

## Après 90

### ➤ L'informatique distribuée

- Aujourd'hui, le SI d'une entreprise est bien souvent composé de logiciels et de réseaux provenant de quatre ou cinq fournisseurs différents.
- Ces différents éléments sont interconnectés pour assurer la disponibilité des applications temps réel.
- Cette architecture peut supporter des applications :
  - un modèle classique, toutes les ressources qui sont localisées sur la même machine,
  - un modèle client/serveur, les ressources utilisées peuvent être n'importe quelle machine du réseau.

### ➤ On nomme client le poste qui se connecte au serveur

Une architecture client-serveur

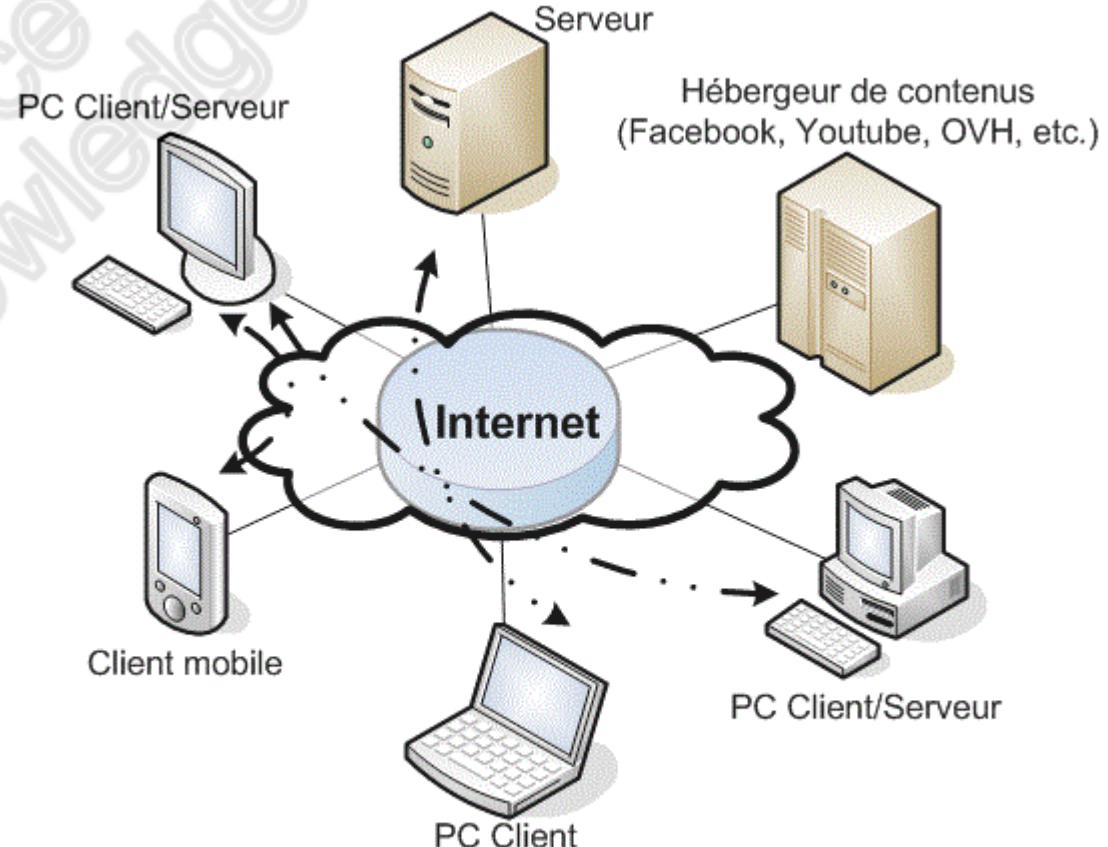


# EVOLUTION DES ARCHITECTURES :

## Après les années 2000

### ➤ L'informatique mobile et Internet

- Avec le développement des smartphones, des tablettes et autres appareils, il devient possible de se connecter en mobilité.
- Cette architecture peut supporter des applications :
  - Natives : l'application est développée pour le système d'exploitation du smartphone concerné.
  - En ligne : il s'agit alors d'application web indépendante du type de smartphone.



« Vrai Internet » : réseau maillé d'échange

# L'INTEROPERABILITE ET PORTABILITE

L'interopérabilité entre les composants d'un SI distribué peut se définir comme **la capacité de ces composants à échanger des services et des données.**

Repose sur des conventions adoptées entre les 2 parties **les protocoles d'échange**

**Impose l'indépendance vis-à-vis :**

- Des plates-formes
- Des couches réseau
- Des systèmes d'exploitation
- Du langage de programmation
- Des formats de données

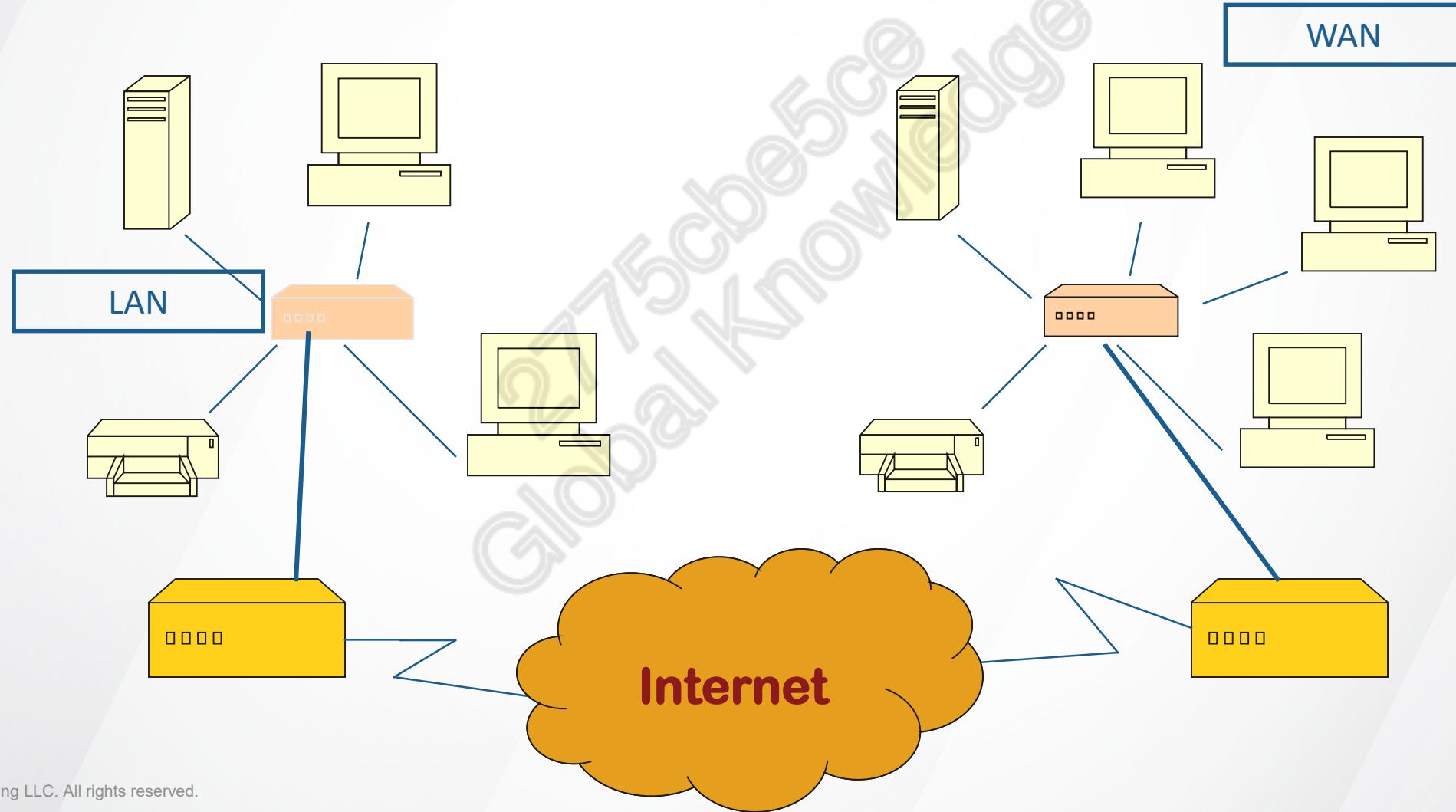
# L'INTEROPERABILITE ET PORTABILITE

## Normes et standards

- Ont émergé au cours de l'évolution du client serveur, 4 standards fonctionnels d'interopérabilité : CORBA, DCOM, Java RMI, Web Services.
- **On tend vers les services Web : SOAP, WSDL et UDDI.**
- Quelque soit le standard, on retrouve toujours 3 informations :
  1. Protocole de communication entre Client/Serveur
  2. Interface de définition d'échanges
  3. Annuaire référençant le service échangé
- Les évolutions matérielles et technologiques des 15 premières années du XXIème siècle ont permis l'émergence du **cloud computing**, l'informatique dans les nuages.  
Nous reviendrons sur ce concept, après avoir décrit les architectures

# LES ARCHITECTURES DISTRIBUEES

Si constituée de plusieurs entités (composants et clients) séparées en termes d'accès réseau.



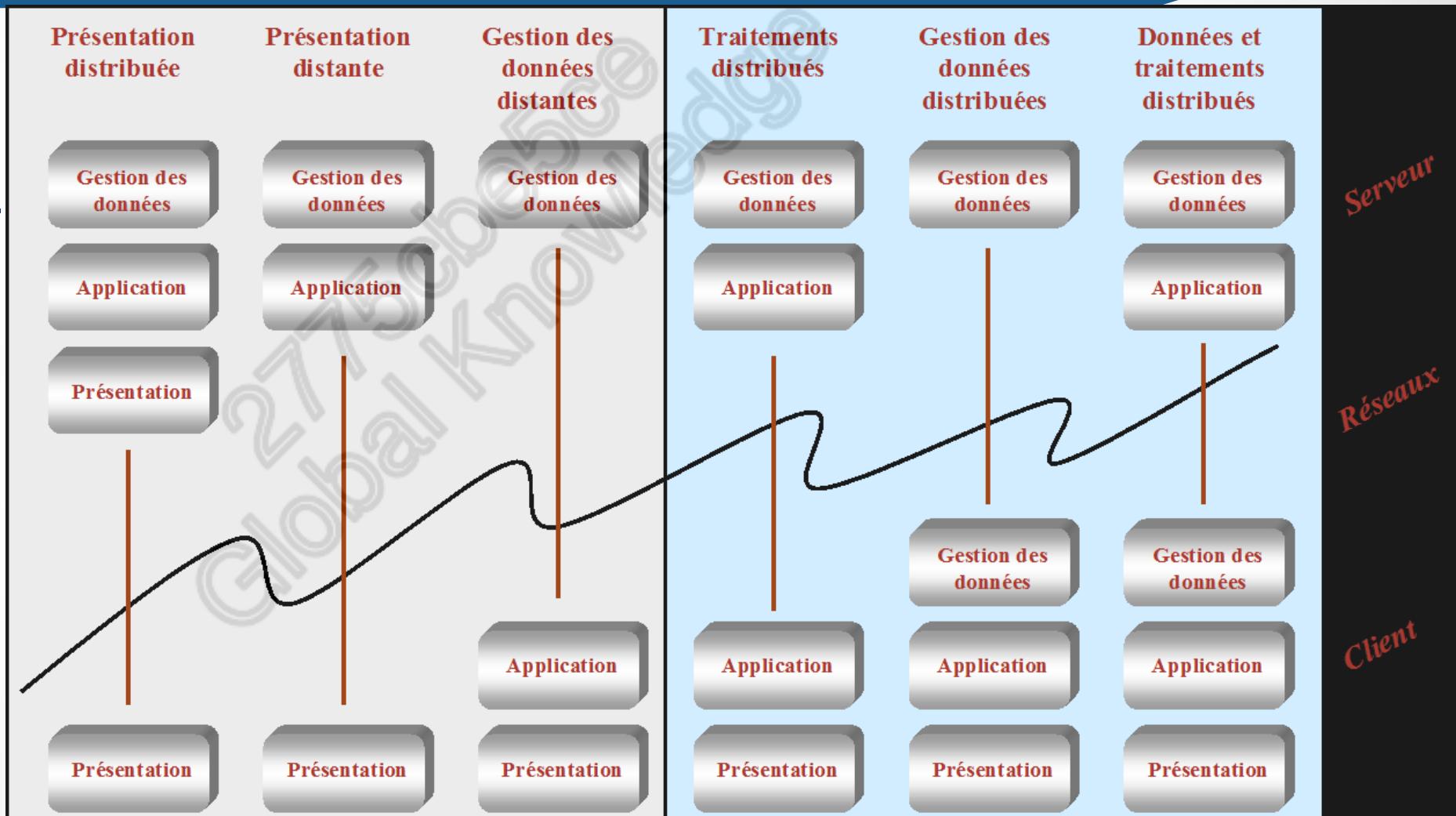
# LE MODELE DU GARTNER GROUP

- L'échange de messages C/S met en œuvre des mécanismes complexes, pris en charge par un **middleware** = une interface de communication universelle entre processus, permettant de rendre presque transparent l'échange.
- Tout programme applicatif contient trois types de traitements :
  - les traitements métiers (application)
  - liés à la **présentation** à la gestion de l'interface utilisateur
  - les traitements liés aux **données** manipulées
- Le Gartner Group, cabinet de conseil américain, a modélisé les différents types de répartition des trois composants (présentation, données, application) sur deux machines.

Modèle  
Vue  
Contrôleur

# LE MODELE DU GARTNER GROUP

- Ce graphique montre les différentes évolutions des architectures 2-tiers.
  - Nous y voyons les **principaux modèles de répartition des couches** entre le client et le serveur.
- En général, on appelle client l'émetteur d'une requête et serveur le récepteur chargé de traiter la requête émise.



# Les évolutions du Client - Serveur

## ➤ Le modèle client-serveur « classique » :

- Tout au long des années 80/90 2000, les évolutions du paysage informatique vont remettre en cause les conditions même d'organisation du travail et de production logicielle. Elles se caractérisent par :
  - une augmentation des performances matérielles de la micro-informatique (vitesse des processeurs, capacité de stockage accrue) ;
  - une fiabilité accrue du matériel ;
  - une diminution significative des coûts ;
  - une banalisation de l'utilisation de logiciels de bureautique qui ne sont plus seulement réservés à des informaticiens ;
  - l'arrivée des interfaces en mode graphique.

# Les évolutions du client - Serveur

- Les raisons principales des dysfonctionnements de ce type sont :
  - complexité de mise œuvre dans un environnement hétérogène,
  - absence de standard dans la couche de middleware,
  - performances moindres en regard d'une solution centralisée,
  - coût excessif du déploiement et de la maintenance,
  - syndrome du « client lourd».

# Les évolutions du client - Serveur

## Le client-serveur universel sur Internet :

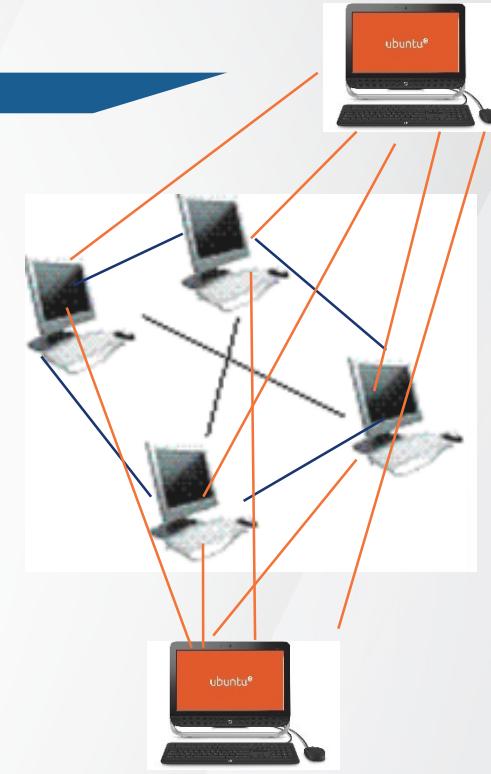
- Raz de marée Internet → bouleverse les architectures C/S et impose un modèle unique pour **Client et Middleware**.
  - Le client est redevenu « léger » : un simple navigateur est nécessaire
  - Le middleware est standardisé : protocole unique de transport d'information (TCP-IP), protocoles simples et partagés d'échange de fichiers (HTTP, FTP...).
  - L'existence d'un langage universel simple, voire simpliste, (HTML) permet de définir la mise en forme des documents complexes.
- Cette architecture simplifiée côté client peut s'enrichir :
  - de programmes téléchargés résidents sur la machine cliente (plug-in) ou d'applet dont la durée de vie est liée à la durée de l'échange avec le serveur (**session**) ;
  - de scripts inclus dans le code HTML peuvent aussi déporter certains traitements côté client.

# Evolutions C/S: Avènement des services Web

- Les grands acteurs du marché de l'informatique (Microsoft, IBM....) ont élaboré des solutions d'échanges de services applicatifs
- Les services informatiques peuvent proposer un service d'échange en ligne des fichiers basé sur des **protocoles simples et transparents** : Format standard: HTTP, Format d'échange des documents: le langage XML, Format commun d'invocation des messages : le protocole SOAP
- Le service peut être accessible sur un serveur par le biais d'une adresse standard (URL)
- On peut objecter que ce type de programmes pourrait être développé dans une architecture C/S classique mais c'est justement le mérite du service web que **d'avoir normalisé des standards en utilisant des protocoles partagés par la communauté informatique.**

# L'ARCHITECTURE WEB : Présentation d'Internet

- Issu du réseau militaire américain ARPANET créé en 1969
- S'est progressivement ouvert à la communauté scientifique et aux universités américaines, à la France en 1986 (INRIA) et enfin aux entreprises et aux particuliers depuis 1989.
- La technologie qui a permis l'engouement actuel est le protocole HTTP, qui permet d'accéder à des pages d'informations par le biais de liens hypertextes. L'ensemble des documents disponibles constitue le **World Wide Web**.
- Accessible 24 heures sur 24, Internet est aujourd'hui **un réseau mondial** de communication d'informations, reliant des centaines de millions d'ordinateurs.

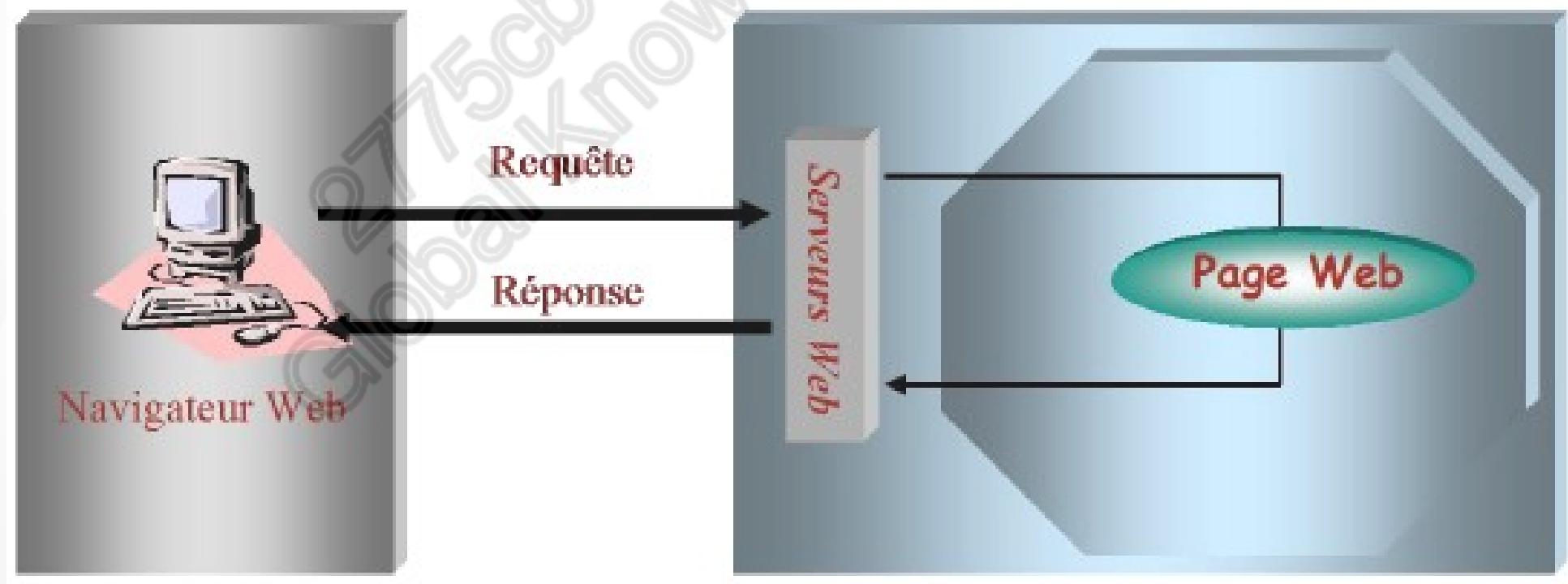


# L'ARCHITECTURE WEB : Présentation d'Internet

- Le réseau Internet offre trois fonctionnalités principales :
  - la **consultation** de pages d'informations ou la consultation et la mise à jour de bases de données, *via* le protocole applicatif HTTP, dit le **Web**
  - les échanges de **courriers** électroniques (*le mail*), via le protocole SMTP,
  - le **téléchargement** de fichiers, en utilisant le protocole FTP.

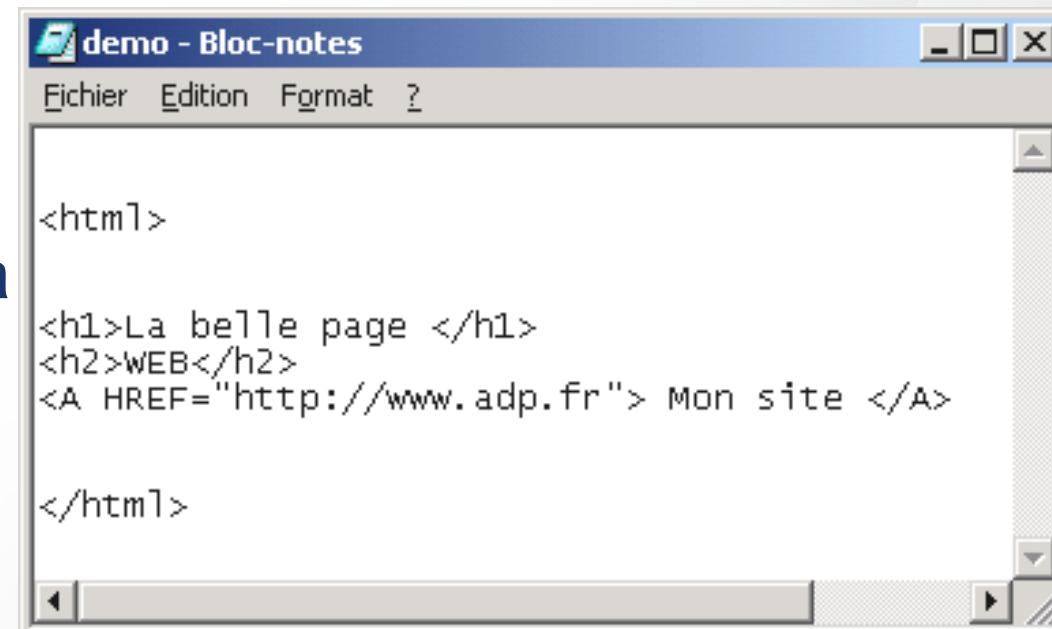
# L'ARCHITECTURE WEB : Composants

- Les trois composants essentiels d'une architecture Web sont :
  - Navigateur Web
  - Serveur Web
  - Page Web



# L'ARCHITECTURE WEB : Le navigateur Web

- Un navigateur Web repose sur la notion d'interface universelle :
  - Il envoie des requêtes à un serveur Web quel qu'il soit.
  - Il réceptionne et gère les réponses retournées par le serveur.
  - Il interprète et affiche les réponses retournées.
- Le format « universel de présentation » est HTML souvent associé à CSS (Cascading Style Sheets) pour séparer la forme du contenu d'un document. Ce ne sont pas des langages de programmation mais des langages de présentation.



```
<html>
<h1>La belle page </h1>
<h2>WEB</h2>
<A HREF="http://www.adp.fr"> Mon site </A>

</html>
```

# L'ARCHITECTURE WEB : Le navigateur Web

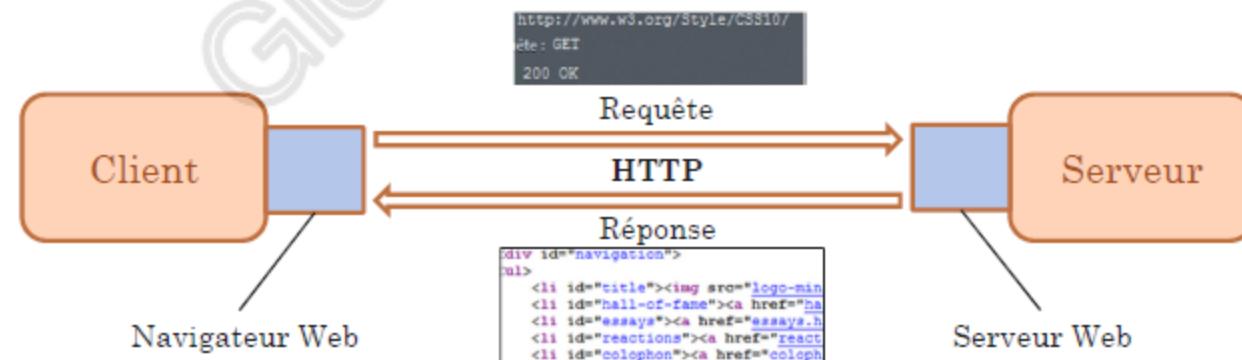
- La page **HTML** peut aussi contenir **du code** - interprété par le navigateur  
le code est souvent écrit en **JavaScript**  
(Nous reviendrons sur les différents langages du Web)
- Le navigateur peut aussi exécuter des applications dans l'environnement de la page Web (ActiveX ou Applets).
- Plusieurs navigateurs sont disponibles sur le marché et intègrent l'accès aux pages Web, le transfert de fichier, le courrier électronique :  
**Microsoft Internet Explorer, Firefox, Opéra, Safari, Google Chrome...**



**Interprétation par les navigateurs**

# L'ARCHITECTURE WEB : Le serveur Web

- Le serveur Web est un programme qui s'exécute sur un serveur.
  - Il attend des requêtes pour les interpréter.
- Suivant la nature de la requête, le serveur Web recherche une page ou exécute un programme.
- Quelque soit le traitement, le serveur Web retournera toujours une page (même si c'est une erreur).
- Le protocole de communication utilisé entre le client et le serveur est HTTP.



# L'ARCHITECTURE WEB : Le serveur Web

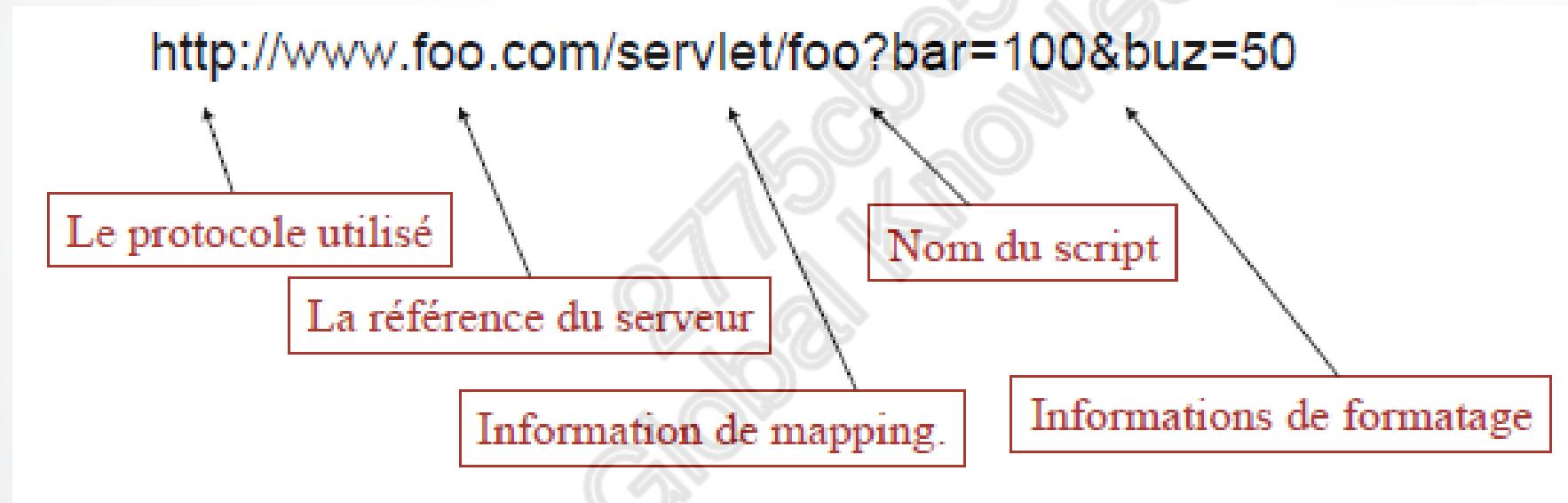
- Le serveur est dit **serveur HTTP** lorsqu'il ne gère que des documents HTML.
- On parle aussi de **Web statique**, même si la page contient en plus du HTML du code JavaScript.
- Plusieurs serveurs HTTP sont disponibles sur le marché les plus connus étant :
  - Microsoft IIS (Internet Information Server)
  - Apache (Apache Foundation).
- A partir du moment où l'on exécute une application sur le serveur pour générer la réponse faite à l'utilisateur, on parle de **Web dynamique**. Dans ce cas le serveur HTTP n'est pas suffisant, il fait appel à d'autres programmes que l'on nomme des **serveurs d'applications**.

# L'ARCHITECTURE WEB : Applications Web

- Chaque serveur est identifié par une adresse **IP** (exemple : 217.19.60.142),
- Chaque page Web est identifiée par une **URL** (Unified Resource Locator).  
exemple :  
<http://www.globalknowledge.fr/search/?SearchTerm=projet+informatique>
- Pour comprendre l'interprétation des requêtes par le serveur, il faut décrire préalablement son contenu.

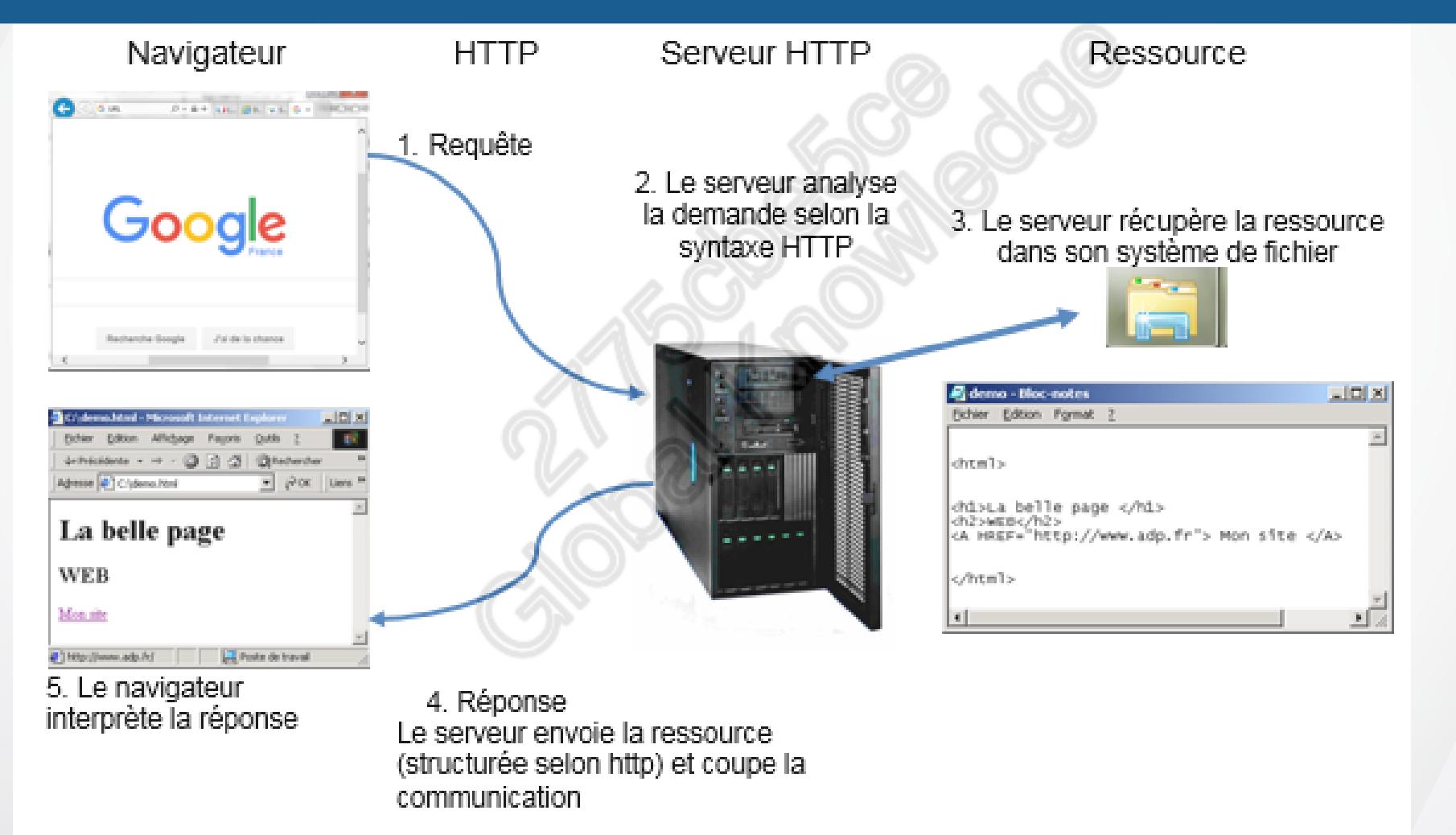
# L'ARCHITECTURE WEB : Applications Web

- Une requête HTTP est constituée des parties suivantes :



- Plus des informations d'en-tête : caractéristiques du navigateur, longueur et type de la requête, etc

# L'ARCHITECTURE WEB : Applications Web Statique



# Autour du Web : Intranet-Extranet

- L'**intranet** correspond à un ensemble de services réseau (site web, messagerie...) supportés par l'infrastructure Internet, mais dont l'accès est restreint à un groupe d'utilisateurs particulier, par exemple l'ensemble des salariés d'une entreprise.
- L'**extranet** définit une restriction moins stricte que l'intranet. C'est par exemple le cas d'un site web réservé à une entreprise et à ses clients, qui ont accès au site en saisissant le mot de passe qui leur a été communiqué.

# Autour du Web : La sécurité

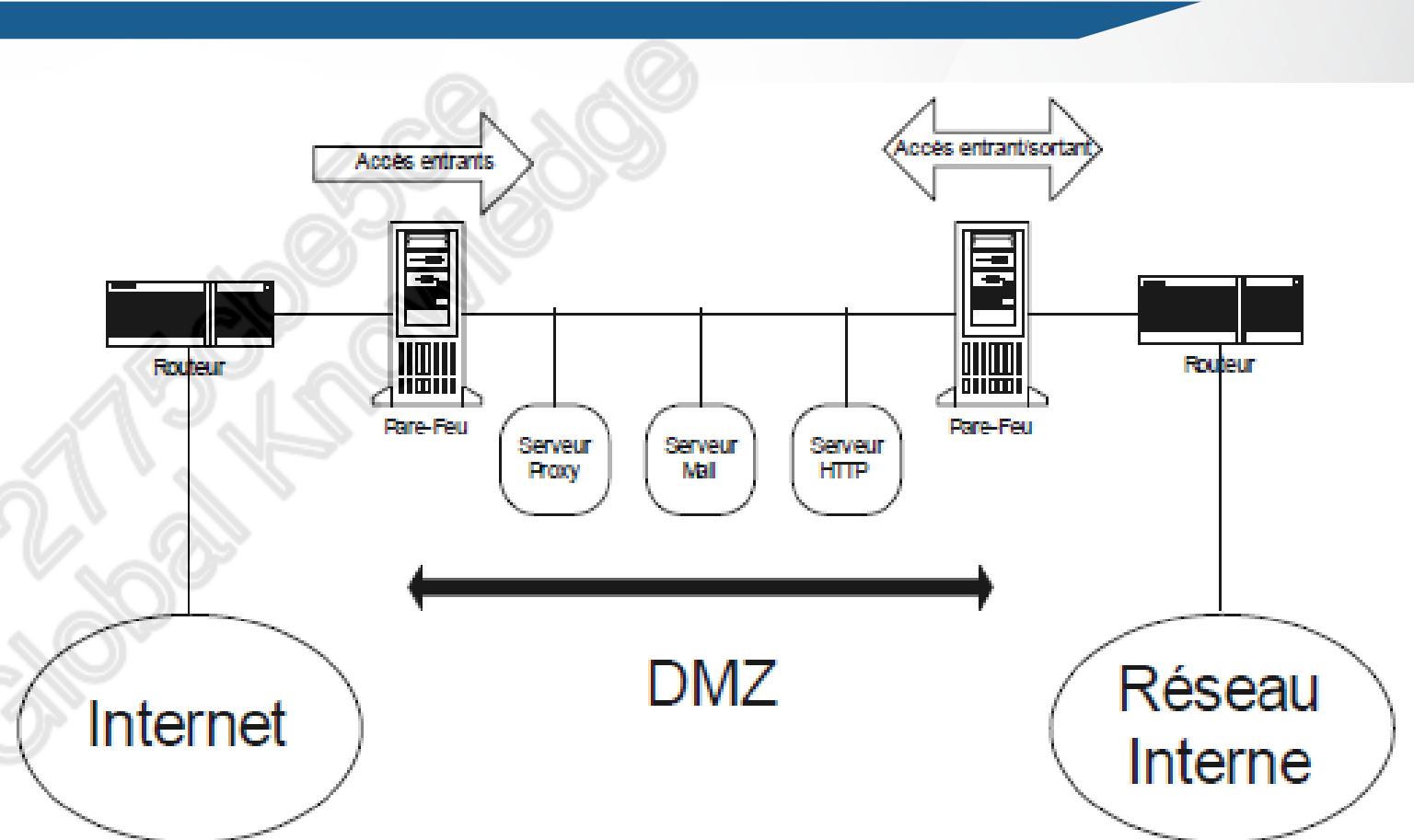
- La mise en œuvre de restrictions d'accès, ainsi que la mise en œuvre plus globalement de la sécurité du réseau Internet, passe par un certain nombre d'outils et de techniques dont les plus fréquentes sont :
  - le **firewall**, ou pare feu, composant matériel ou logiciel paramétrable qui permet de filtrer les accès à un réseau interne en fonction de l'adresse de la machine émetteur de la requête, de l'adresse de la machine destinataire, du type de service demandé...
    - Exemple : interdire à un PC externe à l'entreprise l'accès à l'intranet

# Autour du Web : La sécurité

- le **proxy-HTTP** qui, au-delà de ses fonctions de mutualisation de cache, permet d'interdire aux utilisateurs internes l'accès à certains sites et à certaines ressources Internet,
  - Exemple : interdire aux employés d'accéder à des sites à caractère non professionnel, ou susceptible de contenir des virus.
- les techniques de **cryptage** qui garantissent la confidentialité, l'intégrité et/ou l'authenticité d'un message.
  - Exemple : masquer un code carte bleu saisi sur un site d'achat en ligne.

# Autour du Web : La sécurité

- la **DMZ**, la zone démilitarisée est un sous-réseau.
- Cette DMZ est isolée du réseau local par un pare-feu qui interdit toute connexion depuis la DMZ vers le réseau local.
- Elle est ensuite isolée du réseau Internet par un autre pare-feu qui ne laisse passer que certaines connexions d'un réseau vers l'autre.



# Autour du Web : La sécurité

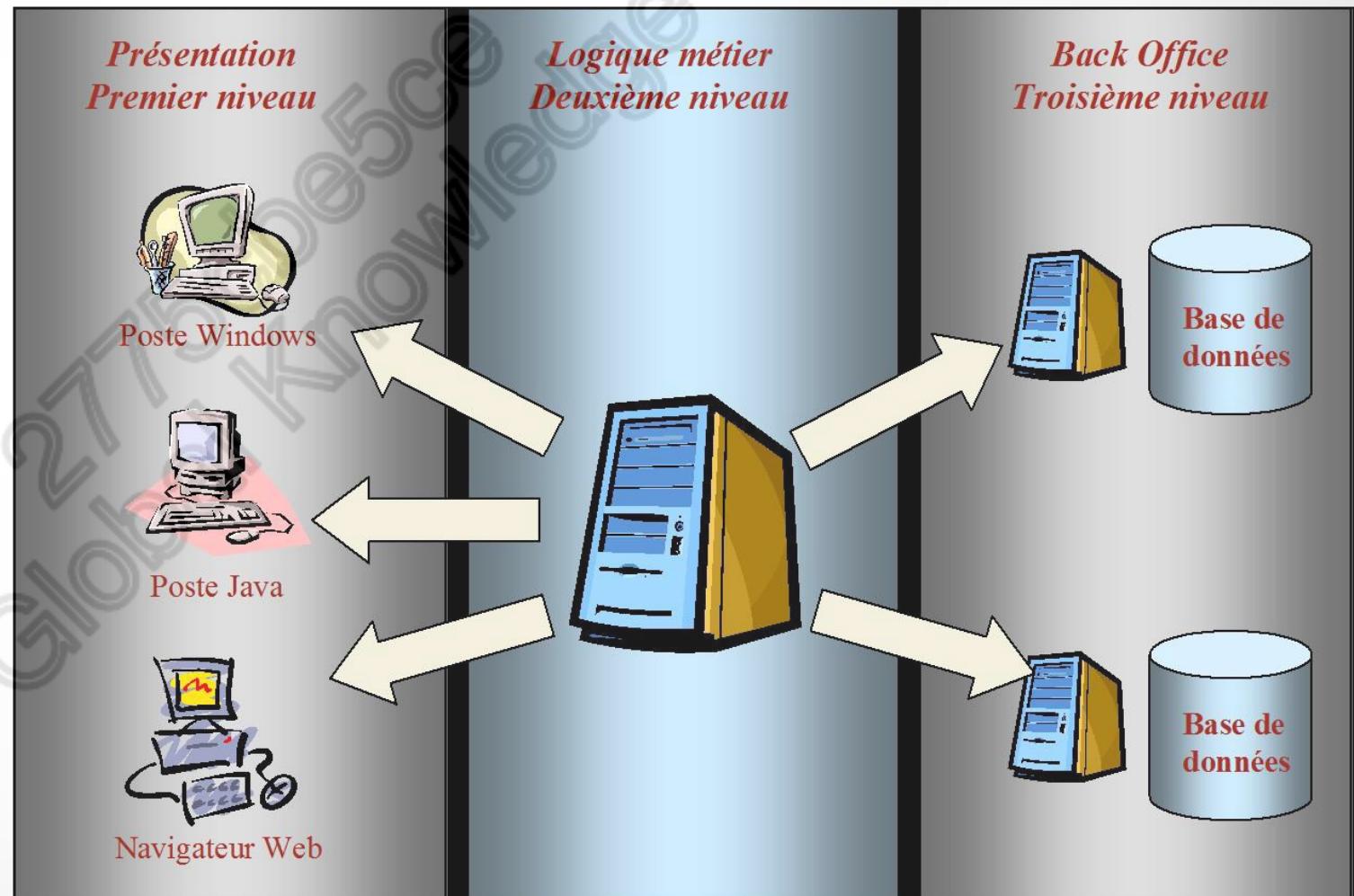
- Un **cookie** est un fichier envoyé par le serveur Web.
  - Il est stocké sur le poste utilisateur, et contient des informations permettant au serveur Web d'identifier ce même utilisateur à sa prochaine demande,
  - Le navigateur peut interdire le stockage de cookies. Cependant, le cookie n'est pas un programme enregistrant ou supprimant des informations du système de l'utilisateur. Il ne contient que ce que l'utilisateur a bien voulu donner.
- Un **logiciel malveillant** (malware en anglais) est un logiciel écrit dans le but de nuire à l'informatique.
  - Dans cette catégorie de logiciels, on trouve, les virus, les vers, les chevaux de Troie, les hoax (canulars), les spywares (logiciels espions)...
- Le pare feu a aussi pour but de lutter contre ces logiciels malveillants.
- Les antivirus sont des logiciels conçus pour neutraliser les logiciels malveillants les éditeurs les plus connus sont McAfee, Norton (Symantec), Microsoft (ForeFront).

# Autour du Web : Service Web

- Un service Web (Web Service en anglais) est un programme permettant la communication et l'échange de données via le monde du Web entre systèmes hétérogènes.
  - Les services Web reposent sur un ensemble de protocoles et de standards gérés par le W3C, (Organisme international gérant la toile).
  - Le protocole de communication est **SOAP** (Simple Object Access Protocol).
  - La description du service (son nom, le type et le nombre des données échangées, sa localisation càd son URL, son protocole de communication) est prise en compte par un langage standard **WSDL** (Web Service Description Language).
  - Le service Web peut être enregistré dans un annuaire, permettant à tous les internautes de l'utiliser à distance, cet annuaire est **UDDI** (Universal Description Discovery and Integration).

# ARCHITECTURE N-TIERS : Présentation

- Les clients accèdent au **serveur d'applications**.
  - Ce serveur fournit des services aux clients à partir de données récupérées sur un serveur de base de données.
  - Cette architecture se nomme **architecture 3 tiers**. (3 niveaux de communication)



# ARCHITECTURE N-TIERS : Présentation

- La simplicité avant tout :
  - La réduction du nombre de protocoles utilisés
  - La simplification des modèles de répartition
  - Le modèle 3-tiers séparant la présentation, la logique métier et la persistance des données.
- Le serveur d'applications étant capable de communiquer avec d'autres serveurs, il n'y a donc plus de limite d'échange et l'on arrive à des **architectures n-tiers**.

# ARCHITECTURE N-TIERS : Les enjeux

- Aujourd'hui, les applications doivent combiner :
  - L'existant sous la forme de base de données, d'applications traditionnelles; on parle du système d'information de l'entreprise (EIS)
  - De nouvelles fonctions de traitement pour offrir des services de qualité à un très large éventail d'utilisateurs
- Les nouveaux services doivent répondre à plusieurs critères de **qualité**.
  - Les services doivent être :
    - **disponibles** pour pouvoir être intégrés dans d'importants environnements de gestion
    - **sécurisés** pour sauvegarder l'intégrité des données de l'entreprise
    - **fiables et évolutifs** pour faire face aux évolutions technologiques

# ARCHITECTURE N-TIERS : Les enjeux

- Les services sont organisés dans des architectures distribuées multi-niveaux (multi-tiers) :
  - Les clients en frontal pour la présentation de l'information
  - Les données et les applications traditionnelles en back end (EIS Enterprise Information System)
  - De plus en plus, au niveau intermédiaire (middle-tiers), entre le client et l'EIS, un ensemble de modules de traitements qui réalisent la plus grande partie du travail

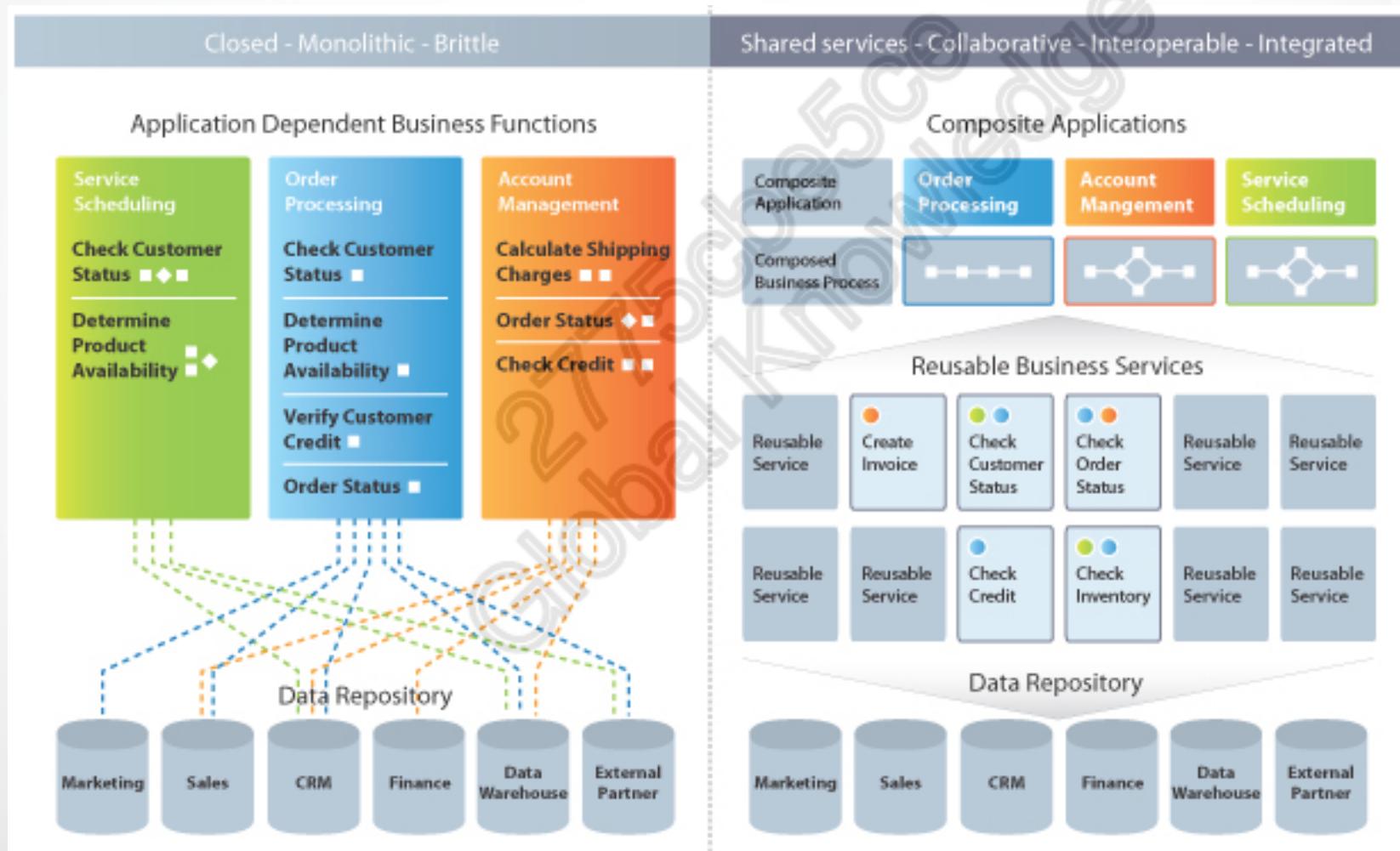
# ARCHITECTURES ORIENTEES SERVICES

## SOA : Présentation

- Une architecture orientée services (notée **SOA** pour Services Oriented Architecture) est une architecture logicielle **s'appuyant sur un ensemble de services simples**.
- L'objectif d'une architecture orientée services est donc de **décomposer** une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.
- L'idée sous-jacente **est de cesser de construire la vie de l'entreprise autour d'applications** pour faire en sorte de construire une architecture logicielle globale décomposée en services **correspondant aux processus métiers** de l'entreprise.

# ARCHITECTURES ORIENTEES SERVICES

## SOA : Présentation



# ARCHITECTURES ORIENTEES SERVICES

## SOA : Service

- Brique fonctionnelle accessible via une prise banalisée ; la prise étant matérialisée par un contrat.
- Un service est donc une façade qui se positionne devant le composant pour le standardiser. Un service est une fonction qui reçoit des messages et les reconstitue après un traitement
- Le composant sous-jacent est isolé du monde extérieur et peut ainsi être modifié à volonté tant que son contrat reste identique

# ARCHITECTURES ORIENTEES SERVICES

## SOA : Processus

- Ensemble de plusieurs activités reliées les unes aux autres pour atteindre un objectif, généralement dans un contexte organisationnel.
- **Consommateur** de service/processus  
Entité logique utilisant un service au travers de son contrat.
- **SOA**  
Principe de construction architectural du système d'informations à partir de services/processus et de consommateurs de services/processus couplés de manière lâche.
- **Les services/processus sont orientés métier : ils exposent des fonctions de haut niveau**

# ARCHITECTURES ORIENTEES SERVICES

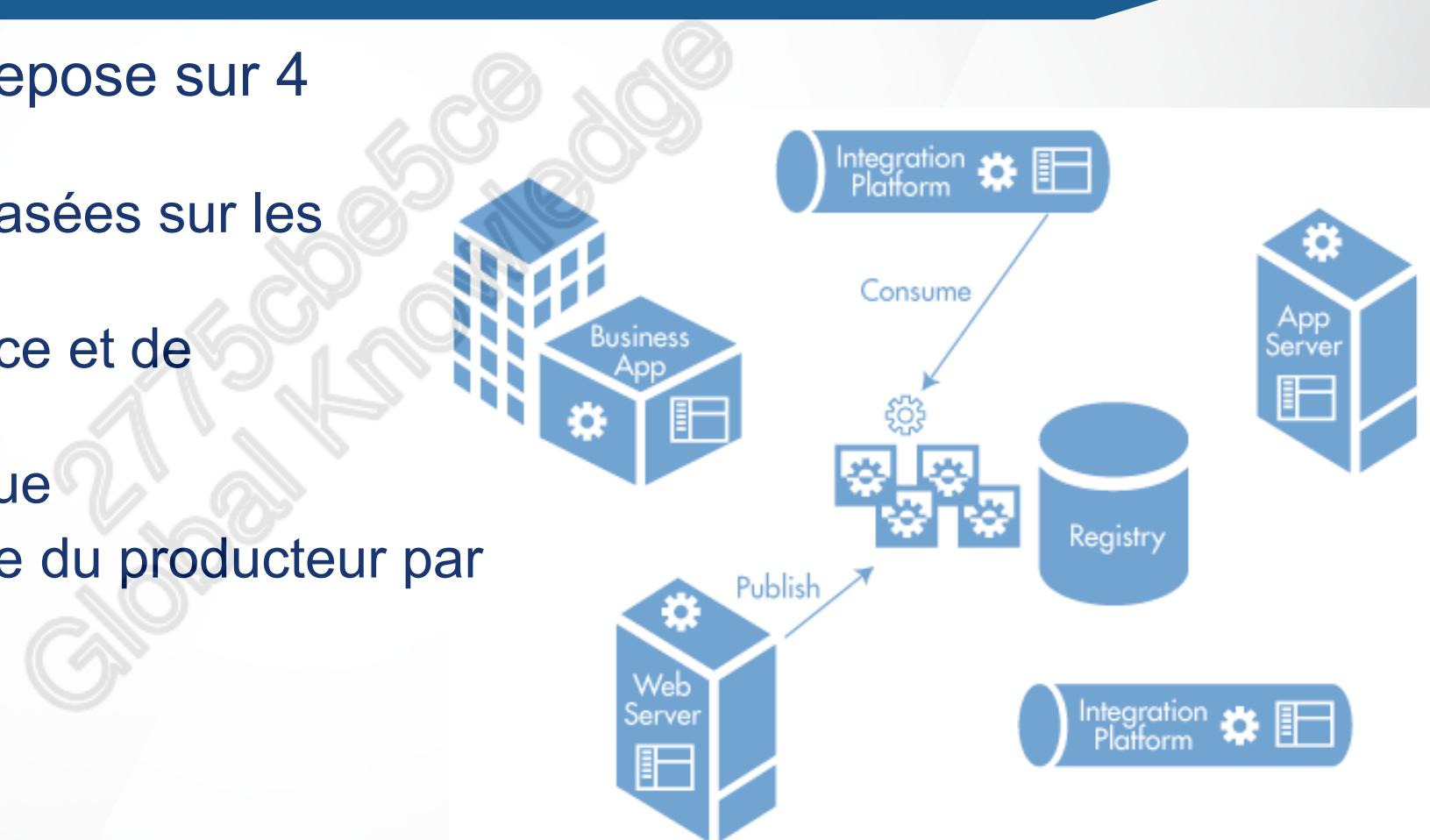
## SOA : Les enjeux

- Les axes majeurs de la SOA sont :
  - **La réutilisation et la composition**, permettant le partage de modules entre applications et les échanges inter-applicatifs ;
  - **La pérennité**, qui implique notamment le support des technologies existantes et à venir ;
  - **L'évolutivité**, car toute application est vivante, a une certaine durée de vie, peut se voir greffer de nouveaux modules et doit pouvoir répondre aux nouveaux besoins fonctionnels ;
  - **L'ouverture et l'interopérabilité**, pour partager des modules applicatifs entre plates-formes et environnements ;
  - **La distribution**, pour pouvoir utiliser ces modules à distance et les centraliser au sein de l'entreprise par exemple ;
  - **La performance**, avec en priorité l'accent mis sur la montée en charge.

# ARCHITECTURES ORIENTEES SERVICES

## SOA : Les enjeux

- La notion de Service repose sur 4 fondamentaux :
  - Support d'interfaces basées sur les standards du marché
  - Séparation de l'interface et de l'implémentation
  - Neutralité technologique
  - Découverte dynamique du producteur par le consommateur



# LE CLOUD COMPUTING : Présentation

- Le cloud computing ou l'informatique dans les nuages se définit comme :
  - Une fourniture de services et d'applications informatique en ligne accessibles partout, à tout moment et depuis n'importe quel terminal (smartphone, téléphone portable, ordinateur ou tablette).
- Le cloud permet :
  - De partager chez un fournisseur de Cloud, une infrastructure, une solution application ou une plateforme à toute personne qui en fait la demande à partir d'un site internet (portail).

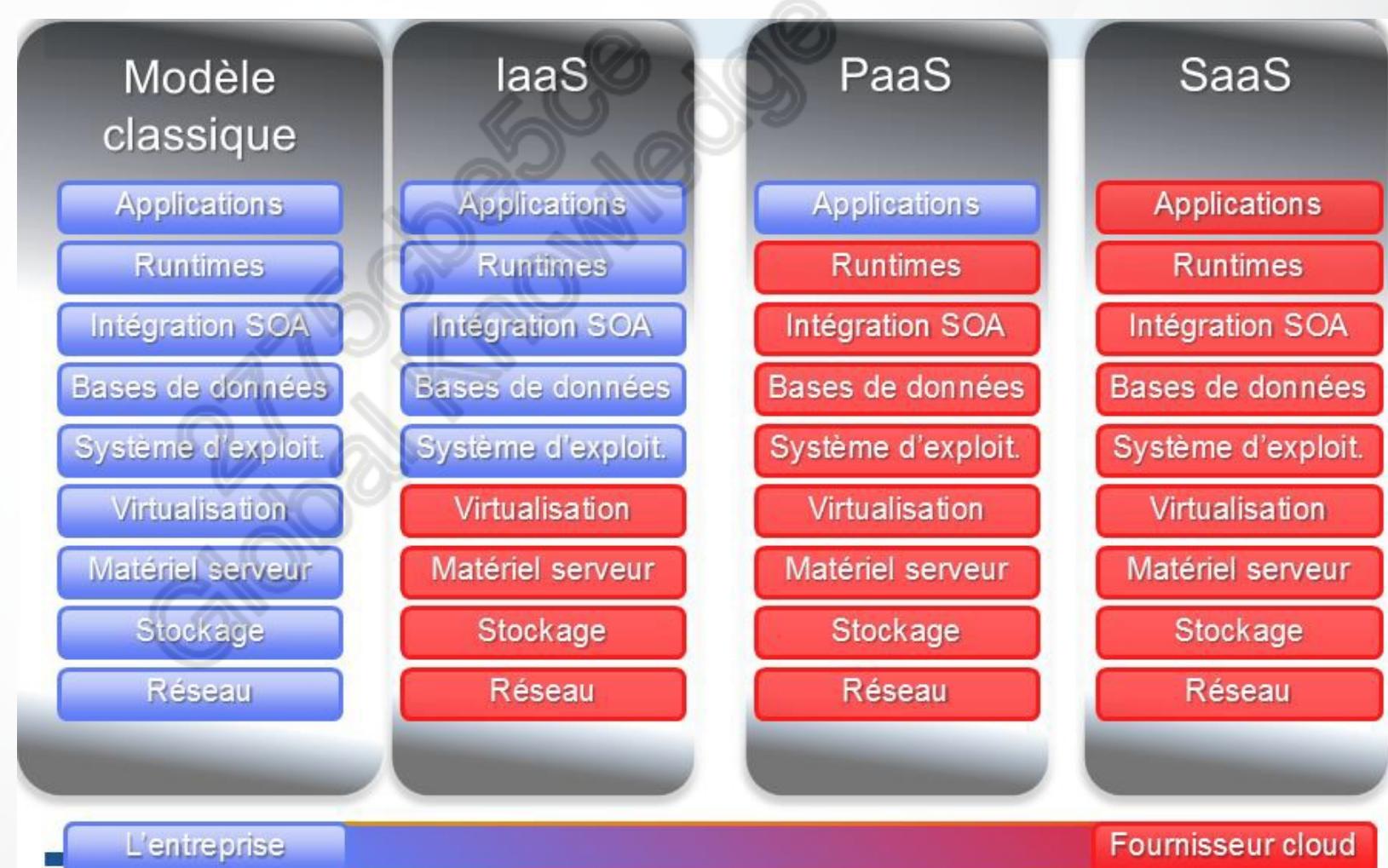


# LE CLOUD COMPUTING : Présentation

- Les caractéristiques du cloud sont :
  - Un libre service à la demande
  - Une élasticité rapide
  - Un accès réseau, des clients variés
  - Une mise en commun des ressources
  - Un service mesuré et une facturation à l'usage
- Les modèles informatiques du cloud sont :
  - IaaS (Infrastructure as a Service), juste l'infrastructure (les machines)
  - PaaS (Platform as a Service), l'infrastructure + OS + outils
  - SaaS (Software as a Service), les logiciels prêt à l'emploi

# LE CLOUD COMPUTING : Présentation

- Dans ces différents modèles, entre l'entreprise et le fournisseur qui fournit quoi :



# LE CLOUD COMPUTING : Les enjeux

- Tant en termes de technologies que d'usages, le cloud computing répond aujourd'hui aux enjeux **d'agilité, de flexibilité, de productivité et de compétitivité** des organisations.
- Les entreprises ont aujourd'hui pleinement conscience des bénéfices qu'elles peuvent tirer du cloud. La question qu'elles se posent désormais n'est plus ni pourquoi ni quand l'adopter, mais bien comment.
- Alors que les **entreprises** migrent un nombre croissant de processus et de fonctions dans le cloud, elles se retrouvent souvent confrontées à un niveau de complexité élevé, pas toujours anticipé. Elles doivent particulièrement se préparer en termes **de gestion et de sécurité** des données, **de gouvernance, d'intégration** des systèmes, **d'impact financier**, ou encore de coordination entre les différents fournisseurs de cloud.

# SOMMAIRE

---

- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# LES COMPOSANTS D'UNE APPLICATION

Objectifs : *Décrire les composants d'une application et leurs évolutions.*

*Comprendre les environnements de test, de développement, d'exécution,*

*Comprendre les outils nécessaires à la création d'une application.*

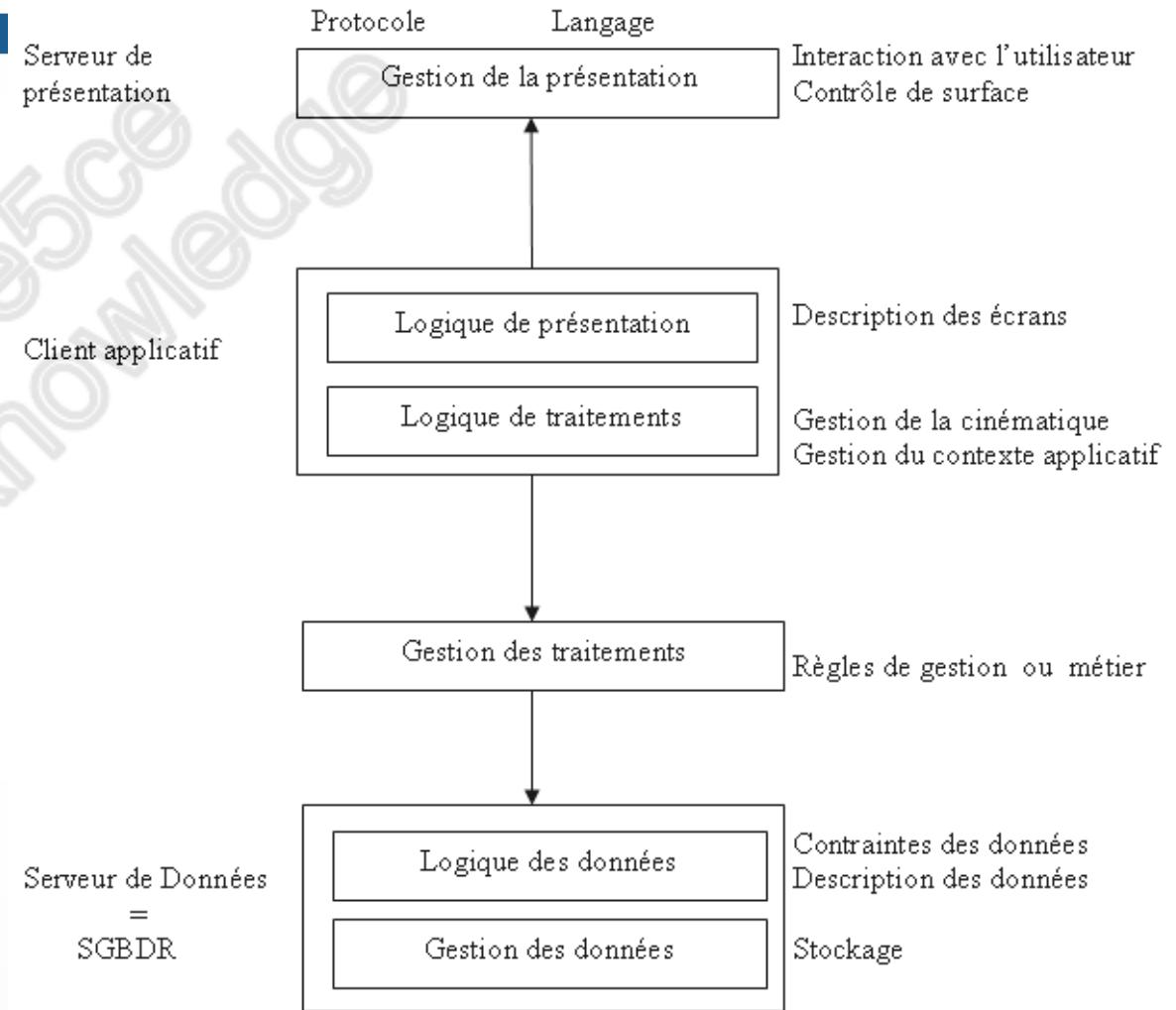


# LE MODELE EN COUCHE : Présentation

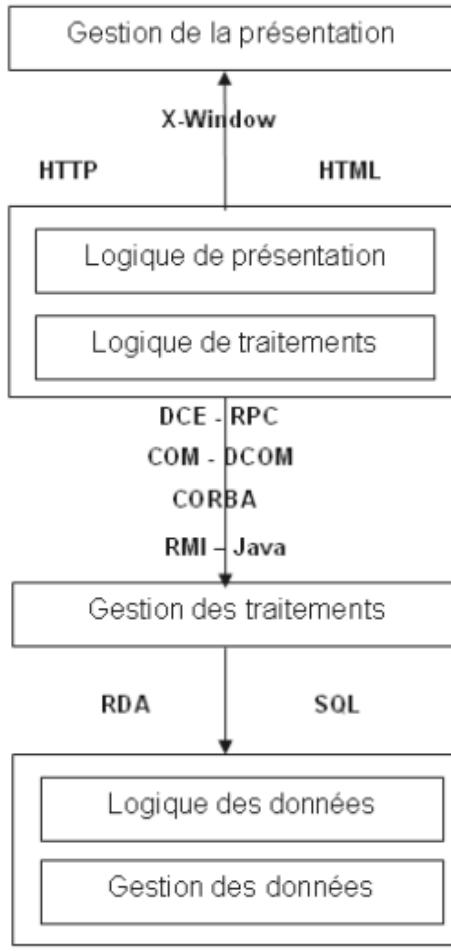
- Avec notre chapitre des architectures distribuées, nous avons noté qu'en Client/Serveur, une application était souvent présentée en 3 couches : Présentation – Traitement – Donnée.
- Nous avons pu voir aussi, que ce découpage correspondait à des architectures dites 3 tiers.  
Nous allons zoomer sur certains points pour mieux comprendre les mises en oeuvre d'aujourd'hui.
- Dans une application informatique, nous pouvons identifier **six couches logiques**, chacune ayant des rôles et des objectifs différents, l'application étant **monolithique ou découpée** pour sa diffusion sur une architecture distribuée.

# LE MODELE EN COUCHE : Présentation

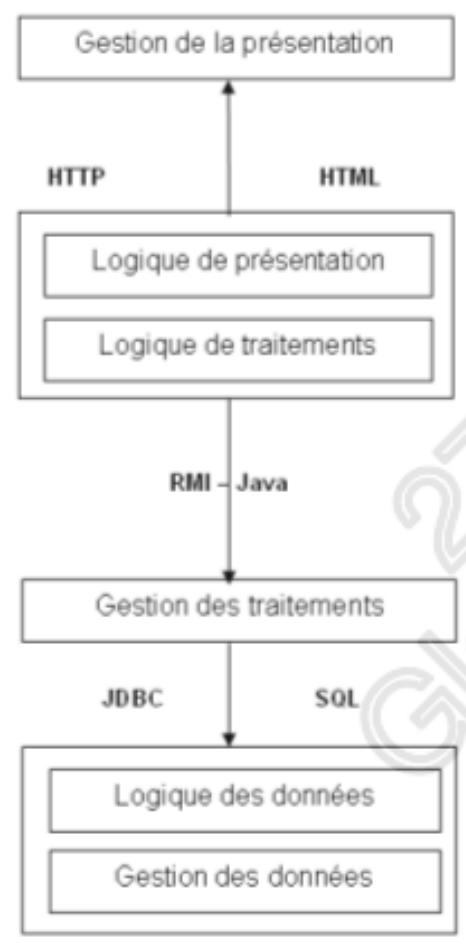
- Ce modèle en couche est la base de la compréhension des nouveaux paradigmes de distributions.
- A chaque couche, correspondra un protocole de communication et un langage d'échange entre les couches.



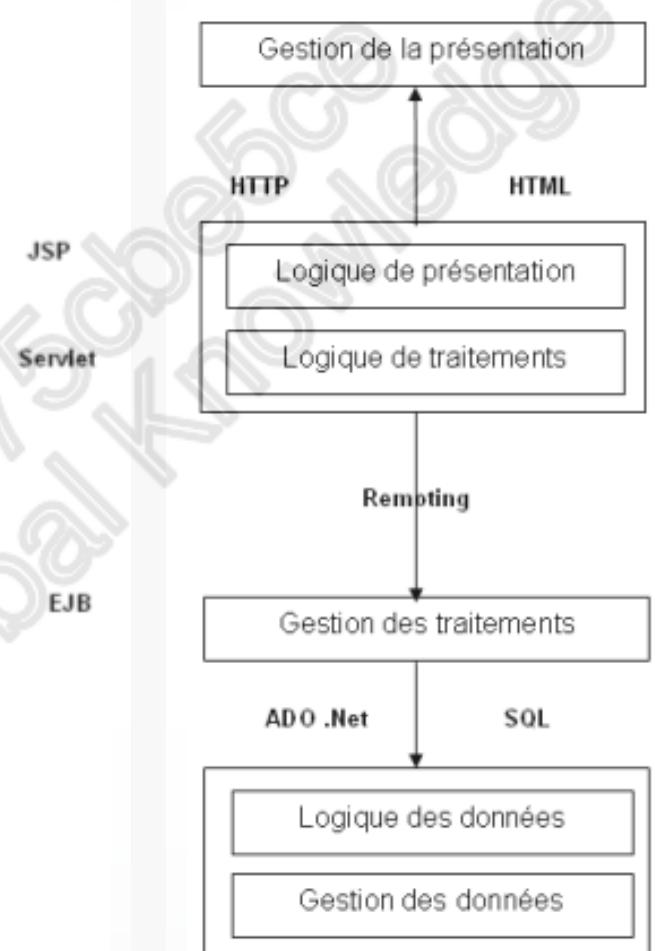
# LE MODELE EN COUCHE Les principaux modèles



Client/Serveur



Java EE



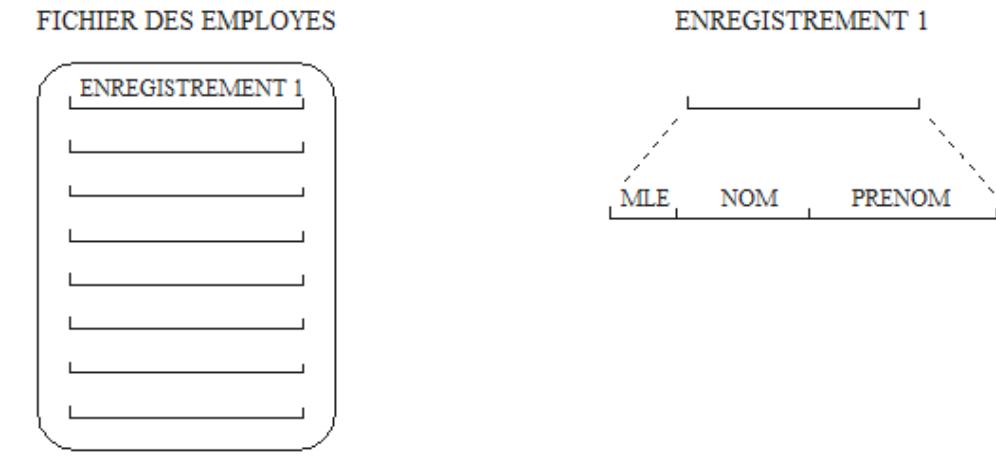
.Net



XML

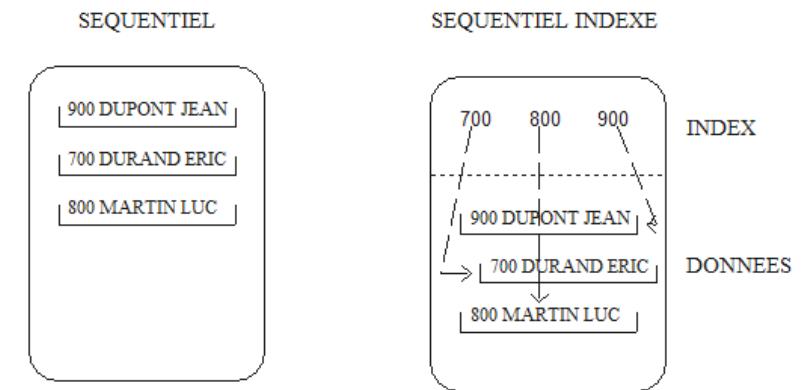
# LA COUCHE DONNEES : Concepts de base

- Outre le support physique utilisé pour le stockage des données, celles-ci sont logiquement regroupées dans des **fichiers** ou des **bases de données**.
- Un fichier ou une base de données correspond à un ensemble fonctionnellement cohérent d'informations (fichier des employés ou base des produits par exemple).
- Au sein d'un fichier par exemple, chaque occurrence d'une entité fait l'objet d'un **enregistrement**, lui-même découpé en **rubriques**.



# LA COUCHE DONNEES : Concepts de base

- **Fichiers séquentiels** = Ordre chronologique de stockage - Accès séquentiel
  - Cette organisation est à rapprocher de celle des cassettes audio
- **Fichiers séquentiels indexés** = Ordre quelconque – Accès séquentiellement ou directement par l'intermédiaire d'une rubrique définie comme la **clé d'accès**, grâce à un **index**
  - assimilable à celle d'un disque compact audio
- **Base de données** = Stockage logique indépendant de l'implantation physique - Pemet la traduction : de liens entre enregistrements, de vues logiques ensemblistes...  
Arrivée des **SGBD** système de gestion de base de données

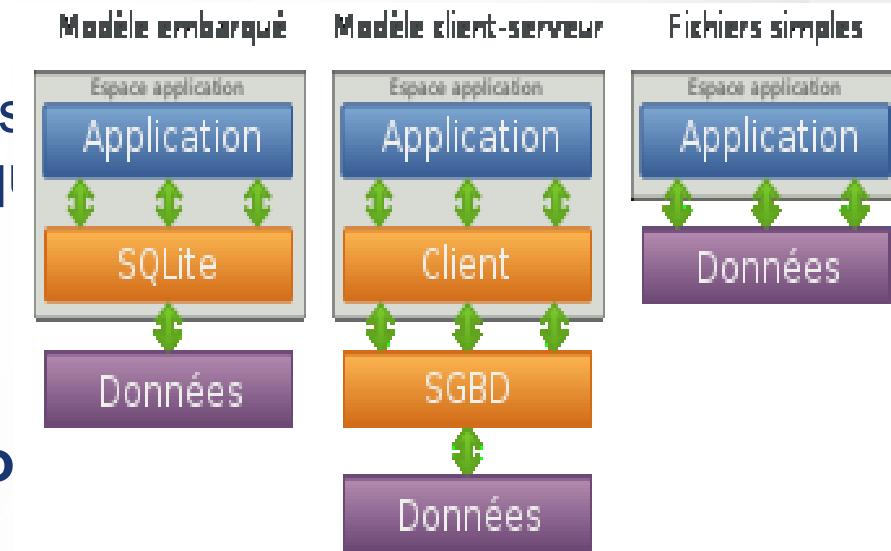


# LA COUCHE DONNEES : Les systèmes de gestion de bases de données (SGBD)

- Les bases de données ont connues quatre évolutions majeures :
  - hiérarchiques (IBM - 1960),
  - réseaux (CODASYL - 1970),
  - relationnelles (IBM - 1980).
  - Big Data, début des années 2010, même si ce terme apparaît pour la première fois en 1997
- Bien que le big data en soit à ses débuts, il bouleversera ces prochaines années votre travail.

# LA COUCHE DONNEES : Les systèmes de gestion de bases de données – SGBDr et SQL

- **Les bases de données relationnelles** - 30 ans existence « seule »
- Suite aux avantages suivants :
  - Le langage **SQL** : standard d'accès aux données, quel que soit l'éditeur
  - La théorie des ensembles : plus simple que la théorie des ensembles sur laquelle sont basés les modèles réseaux et hiérarchiques
  - Evolution du schéma des données sans modifier les programmes existants
  - Intégrées aux architectures distribuées - Nombreuses fonctionnalités liées au **développement et à l'exploitation** d'applications C/S
  - Le transactionnel et les propriétés ACID
  - Implantées sur les systèmes embarqués (SQLite)



# LA COUCHE DONNEES : Les systèmes de gestion de bases de données (SGBD)

- Principales bases de données relationnelles :
  - **ORACLE**, société Oracle,
  - La famille **DB2** ou **UDB** (Universal DataBase) d'IBM,
  - **SQL SERVER**, **ACCESS** chez **MICROSOFT**,
  - **MySQL**, base de données open source rachetée par Sun Microsystems en 2008, puis par Oracle. Son concepteur n'ayant pas voulu suivre Oracle, a reproduit sa base sous le nom de : **MariaDB**.
  - **PostgreSQL**, base de données open source, dérivée de la base **Ingres**. **Ingres**, détenue jusqu'en 2005 par Computer Associates, est passée libre en 2004

# LA COUCHE DONNEES : Les systèmes de gestion de bases de données (SGBD)

- Exemple d'accès aux données d'un SGBD/R (SGBD Relationnel)
  - Soit les tables :
    - EMPLOYEE (MLE, NOM, PRENOM, SEXE)
    - AFFECTATION (MLE, CODPRO, DATDEB)
    - DEPT (CODEPT, DESIGN, DG)

The diagram illustrates the relationships between three tables:

- DEPT** table:
  - Primary key: CODEPT [DECIMAL(10, 0)]
  - Columns: DESIGN [CHAR(20)], DG [CHAR(20)]
  - Data:

|     |              |        |
|-----|--------------|--------|
| 2   | CONSEIL      | DUPONT |
| 3   | INDUSTRIE    | PEREZ  |
| 6   | ETHNOS       | ROQUE  |
| 12  | FORMATION    | MARTIN |
| 120 | COMPTABILITE | MIRAN  |
- AFFECTATION** table:
  - Primary key: MLE [DECIMAL(10, 0)]
  - Columns: CODPRO [DECIMAL(10, 0)], DATDEB [DATE]
  - Data:

|     |     |          |
|-----|-----|----------|
| 25  | 6   | 01/04/96 |
| 53  | 120 | 02/02/97 |
| 54  | 120 | 02/02/97 |
| 55  | 120 | 02/02/97 |
| 56  | 6   | 02/02/97 |
| 100 | 12  | 01/04/96 |
- EMPLOYEE** table:
  - Primary key: MLE [DECIMAL(10, 0)]
  - Columns: NOM [CHAR(20)], PRENOM [CHAR(20)], SEXE [CHAR(1)]
  - Data:

|     |            |             |   |
|-----|------------|-------------|---|
| 25  | LORENT     | CATHERINE   | F |
| 53  | BLASQUEZ   | NICOLAS     | M |
| 54  | HERNANDEZ  | ANTOINE     | M |
| 55  | MARCIER    | JACQUES     | M |
| 56  | GALLET     | BRUNO       | M |
| 100 | DUCHE      | SYLVIE      | F |
| 103 | SOUPINACIO | JESUS       | M |
| 218 | FONTAINE   | JEAN-PIERRE | M |
| 273 | CHICHE     | GLADYS      | F |
| 286 | ROUVRAS    | PHILIPPE    | M |
| 303 | SALAMI     | JUSTIN      | M |
| 304 | ALPHANDERY | ALPHONSE    | M |
| 333 | PONCHEL    | VINCENT     | M |
| 633 | BALLARD    | GASTON      | M |
| 652 | BIBER      | ALBERT      | M |
| 672 | DUVAL      | CHRISTINE   | F |

The detailed view of the EMPLOYEE table shows the following data:

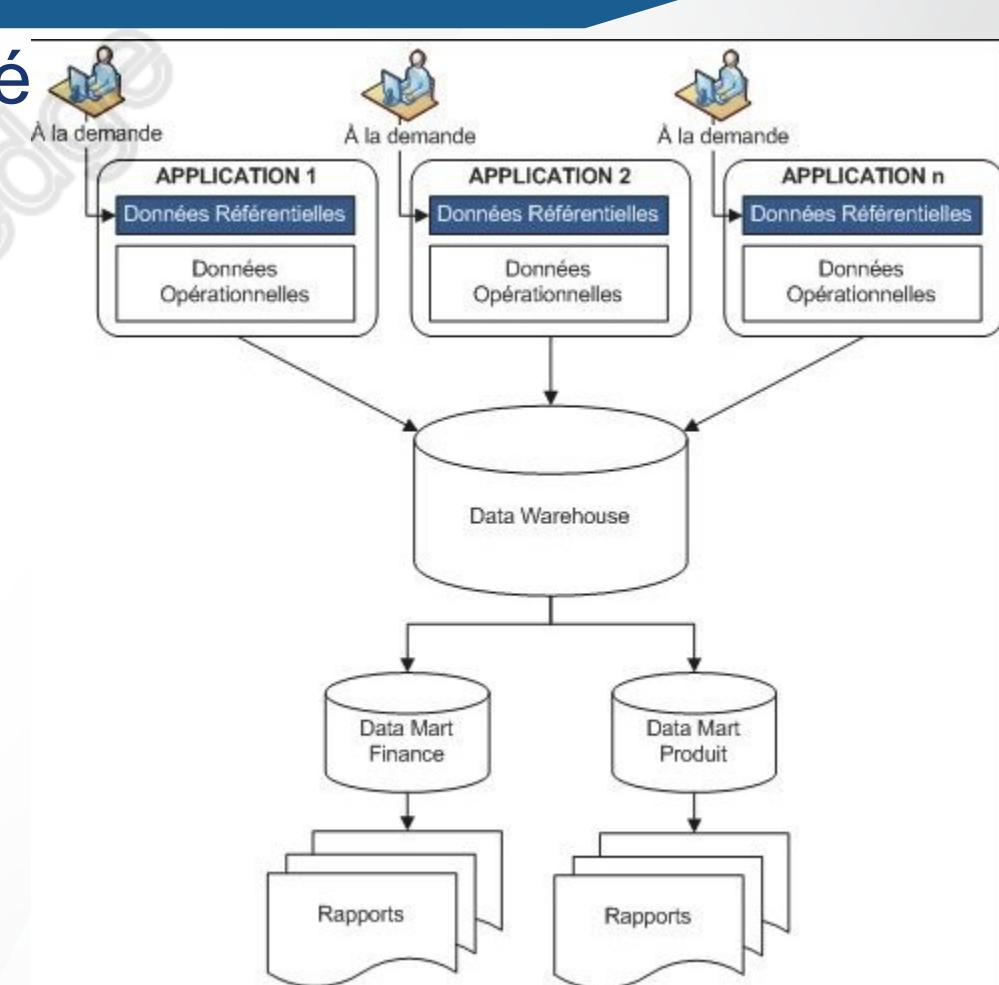
| MLE | NOM        | PRENOM      | SEX |
|-----|------------|-------------|-----|
| 25  | LORENT     | CATHERINE   | F   |
| 53  | BLASQUEZ   | NICOLAS     | M   |
| 54  | HERNANDEZ  | ANTOINE     | M   |
| 55  | MARCIER    | JACQUES     | M   |
| 56  | GALLET     | BRUNO       | M   |
| 100 | DUCHE      | SYLVIE      | F   |
| 103 | SOUPINACIO | JESUS       | M   |
| 218 | FONTAINE   | JEAN-PIERRE | M   |
| 273 | CHICHE     | GLADYS      | F   |
| 286 | ROUVRAS    | PHILIPPE    | M   |
| 303 | SALAMI     | JUSTIN      | M   |
| 304 | ALPHANDERY | ALPHONSE    | M   |
| 333 | PONCHEL    | VINCENT     | M   |
| 633 | BALLARD    | GASTON      | M   |
| 652 | BIBER      | ALBERT      | M   |
| 672 | DUVAL      | CHRISTINE   | F   |

# LA COUCHE DONNEES : Les systèmes de gestion de bases de données (SGBD)

- Liste de tous les employés de l'entreprise :  
**SELECT MLE, NOM, PRENOM FROM EMPLOYE**
- Liste des employées de sexe féminins de l'entreprise :  
**SELECT MLE, NOM, PRENOM FROM EMPLOYE WHERE SEXE = 'F'**
- Liste des employés du département de code 12 :  
**SELECT EMPLOYE.MLE, NOM, PRENOM FROM EMPLOYE JOIN AFFECTATION  
ON EMPLOYE.MLE = AFFECTATION.MLE WHERE CODPRO = 12**
- Liste des employés du département dont le responsable est "MARTIN"  
**SELECT MLE, NOM, PRENOM FROM EMPLOYE WHERE MLE IN (SELECT MLE  
FROM AFFECTATION WHERE CODPRO IN (SELECT CODEPT FROM DEPT  
WHERE DG = 'MARTIN'))**

# LA COUCHE DONNEES : Des bases relationnelles vers la Business Intelligence

- Les bases de données relationnelles ont été conçues pour **enregistrer rapidement** des données de production, en **éitant la redondance** d'information afin de **mettre à jour rapidement** les données
- Ces bases contiennent donc d'importantes données qui peuvent être analysées à des fins statistiques, marketing, voire de prises de décisions.



# LA COUCHE DONNEES : Des bases relationnelles vers la Business Intelligence

- Le traitement analytique en ligne (en anglais *online analytical procession*, OLAP) est un type d'application informatique orienté vers l'analyse sur-le-champ d'informations selon plusieurs axes, dans le but d'obtenir des rapports de synthèse tels que ceux utilisés en analyse financière.
  - Les applications de type OLAP sont couramment utilisées en BI (informatique décisionnelle), dans le but d'aider la direction à avoir une vue transversale de l'activité d'une entreprise.
- Ce type d'application s'oppose au traitement de transactions en ligne, on a donc copié les données de production du monde SQL dans des bases de données de type OLAP, ou de type SGBD.

Datawarehouse

Datamart

Datamining

DataScientist



ETL

# LA COUCHE DONNEES : De la multitude des données vers le MDM

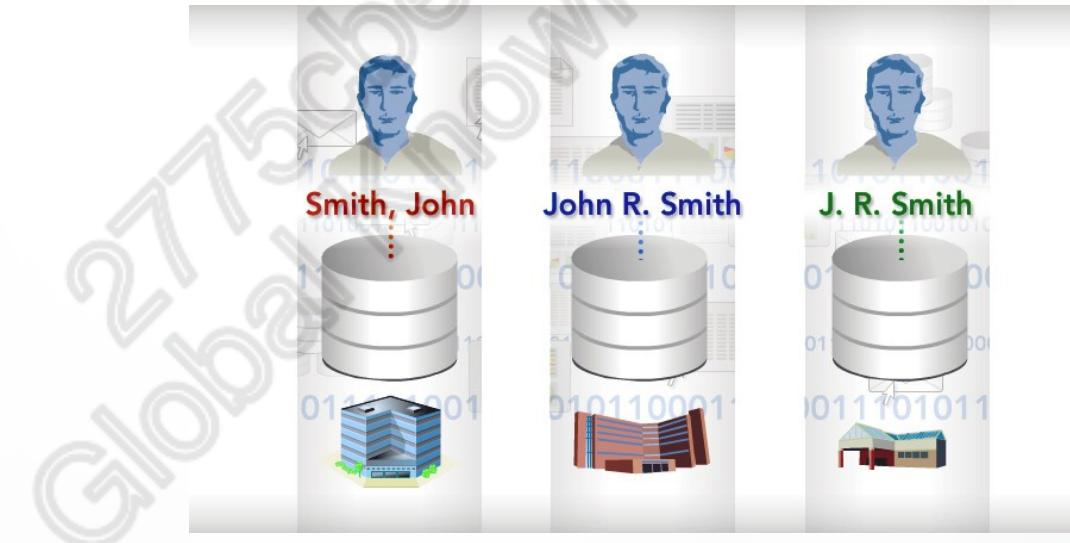
- En général, une entreprise dispose de plusieurs bases de données rangées chacune au sein d'un SI ou derrière une application métier (gestion comptable, ventes, gestion des ressources humaines, serveur de suivi de production, etc)
- C'est notamment le cas pour des structures ayant opté pour une approche best-of-breed à l'inverse d'une politique technologique articulée autour d'un ERP

Best-of-breed = littéralement, le meilleur de sa catégorie

Se dit d'une **solution logicielle** prétendant offrir des fonctions avancées sur un segment de marché bien délimité.

# LA COUCHE DONNEES : De la multitude des données vers le MDM

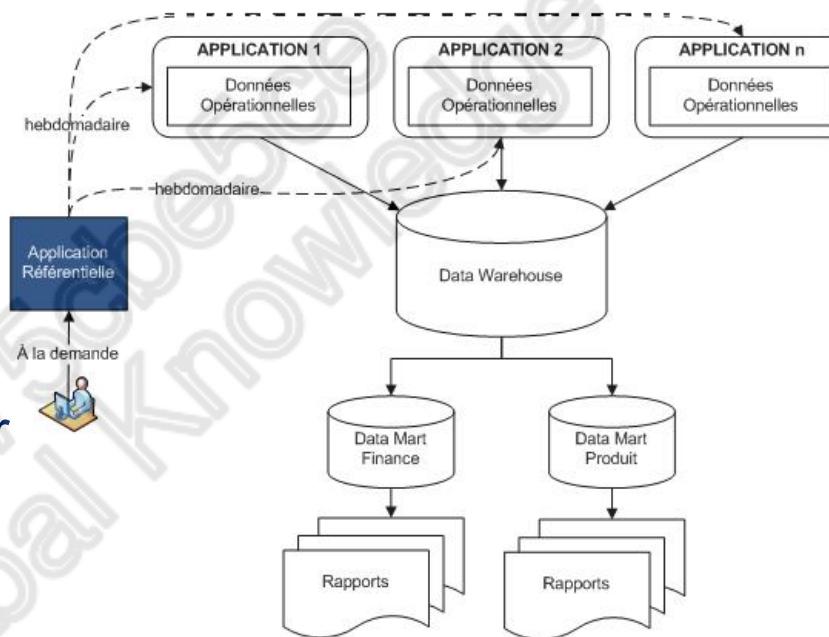
- EX: les informations du client peuvent se retrouver dupliquées dans la base de la gestion commerciale, dans la comptabilité, la BI... Qui est maître de la donnée ? Qui la met à jour ? Comment informer les autres d'un mјj



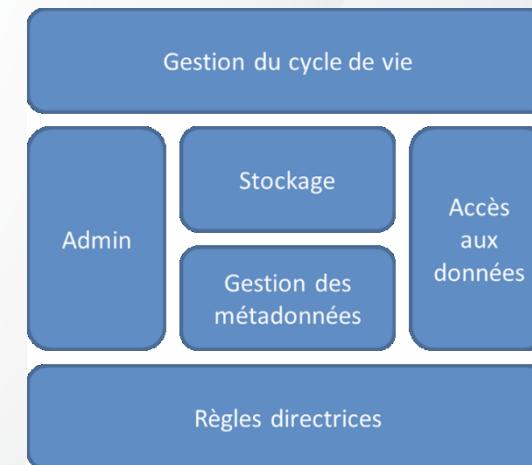
- Il est nécessaire de gérer la qualité et la cohérence des données au sein du SI, telle est la vocation, de la méthode applicative nommée MDM : Master Data Management.

# LA COUCHE DONNEES : De la multitude des données vers le MDM

- La gestion des données de référence
- Quelques acteurs :
  - Orchestra Networks : EBX
  - IBM : InfoSphere MDM, WS Product Center
  - SAP Netweaver MDM
  - Talend Enterprise MDM
  - Microsoft : SQL Server Master Data Services
  - Informatica MDM



Architecture d'un système MDM



# LA COUCHE DONNEES : Le Big Data

- Big data est le terme utilisé pour désigner des données volumineuses difficiles à manipuler par les outils traditionnels, tels que les bases de données relationnelles. Ce nouveau concept nécessite de revoir :
  - le stockage des données,
  - la gestion de recherche,
  - le partage de l'information entre de multiples utilisateurs
  - sans oublier la restitution
  - et l'analyse des données.
- Certains bouleversements sont donc à prévoir dans les prochaines années.



# LA COUCHE DONNEES : Le Big Data

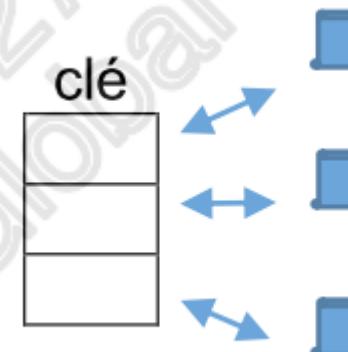
- On parle alors de **NoSQL – Not Only SQL**
- Le Big data, prend de l'ampleur dans les années 2010, avec les GAFA
- Nouveaux modèles de représentation qui permettent de **garantir les performances** sur les **volumétries** en jeu – Zetta,Yotta octect ( $2^{70}$ ,  $2^{80}$ )

Modèle clé/valeur

| Clé | Valeur |
|-----|--------|
|     |        |
|     |        |
|     |        |
|     |        |
|     |        |
|     |        |

DynamoDB, Aerospike,  
Redis et Riak

Orienté document - Format xml ou Json

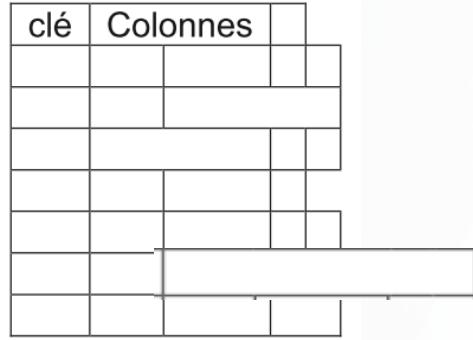


DynamoDB, MongoDB,  
CouchDB, MarkLogic

```
{"_id": ObjectId("4efa8d2b7d284dad101e4bc7"),
  "Nom" : "Dupont",
  "Prenom" : "Pierre",
  "date naissance" : { "date" : 10041995 },
  "adresse" : {
    "rue" : "50, rue de la justice",
    "code postal" : "91230",
    "ville" : "Montgeron"
    "pays" : "FR"
  },
  "telephone" : [
    { "0169001234" },
    { "0603020100" }
  ]
}
```

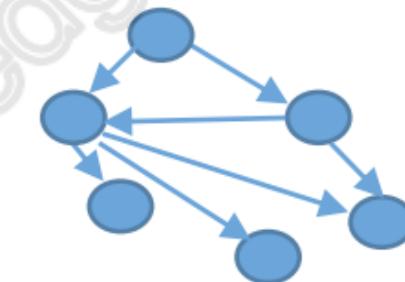
# LA COUCHE DONNEES : Le Big Data

## Modèle décolononné



Cassandra (Facebook),  
Hbase, Accumulo

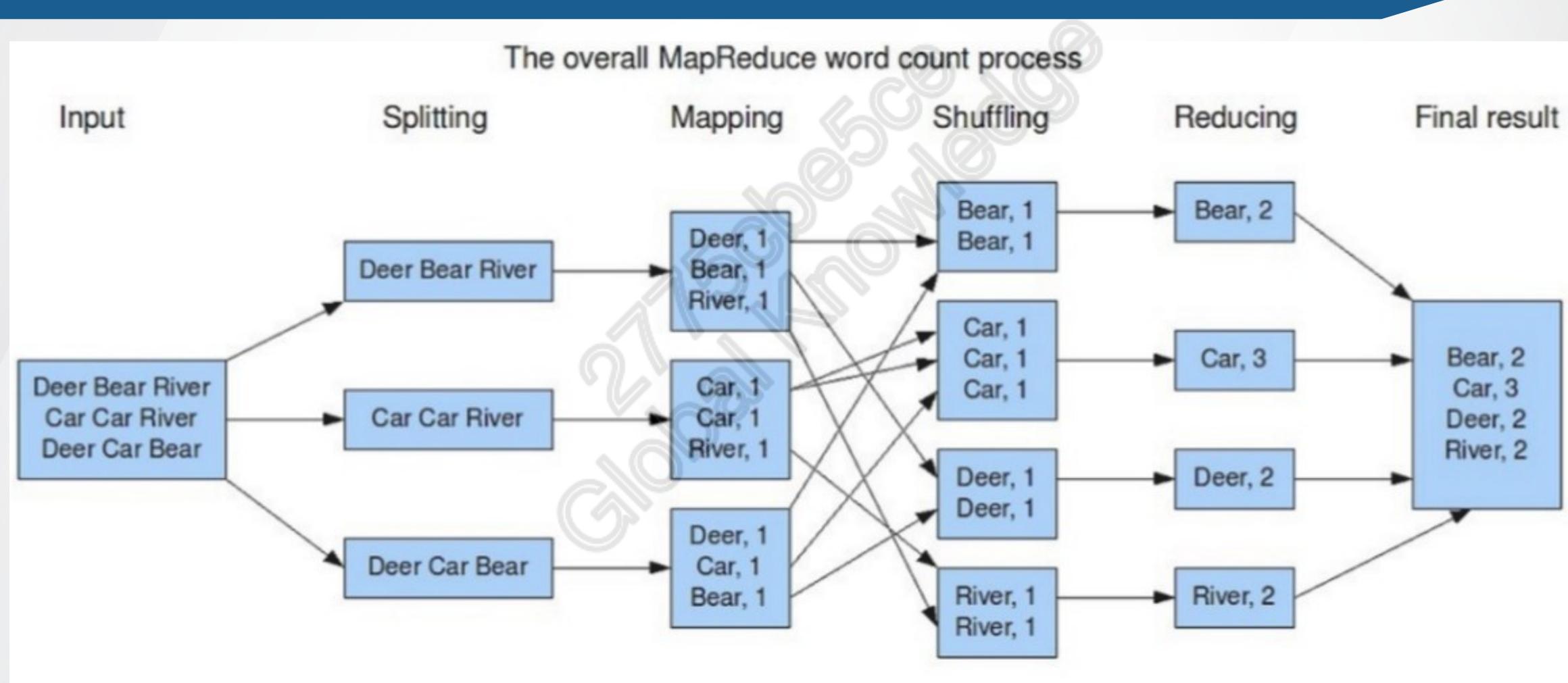
## Modèle Théorie de graphes



InfiniteGraph, Neo4j

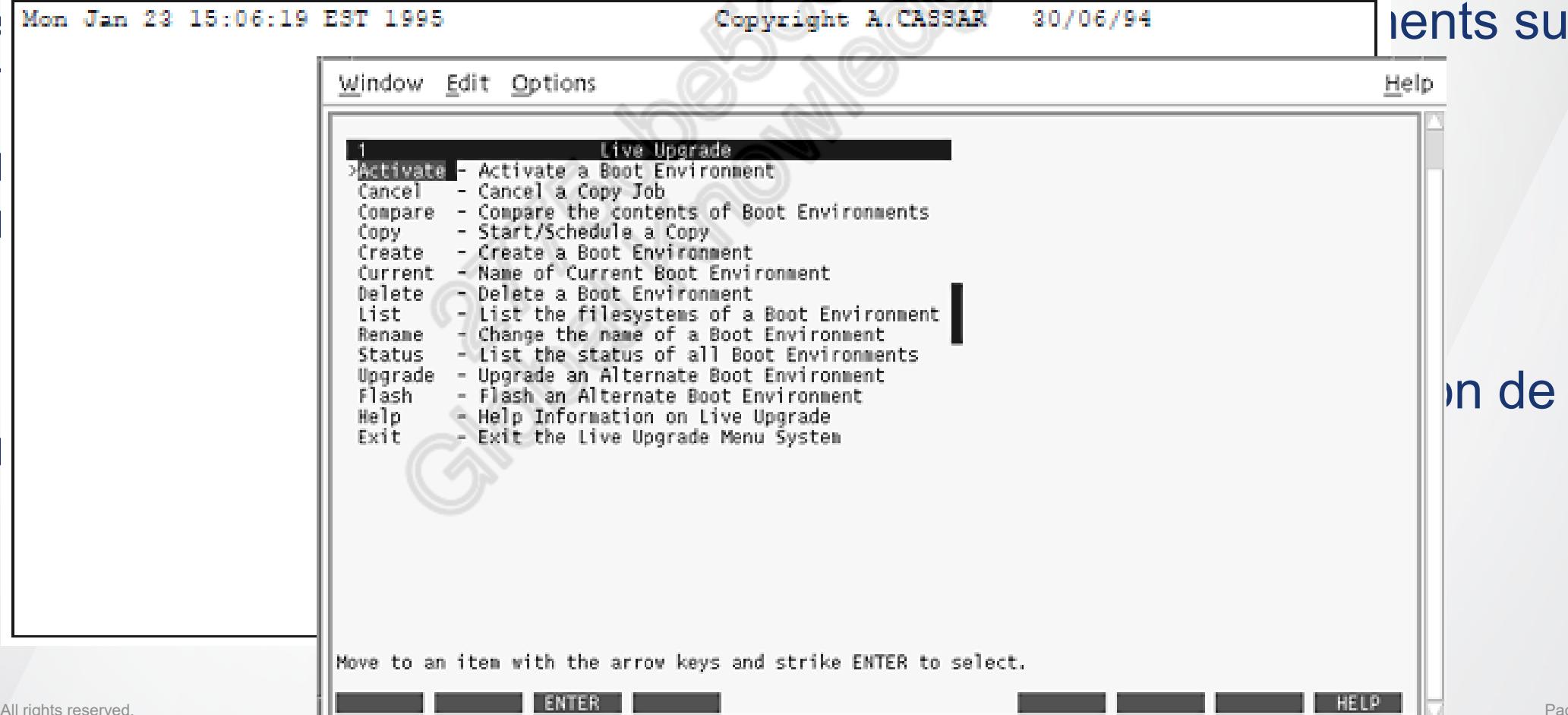
- Technologies dites de **Business Analytics & Optimization** (BAO) permettent de gérer des bases massivement parallèles
- Patrons d'architecture “Big Data Architecture Framework (BDAF) : **MapReduce** (Google) utilisé dans le framework **Hadoop**

# LA COUCHE DONNEES : Le Big Data



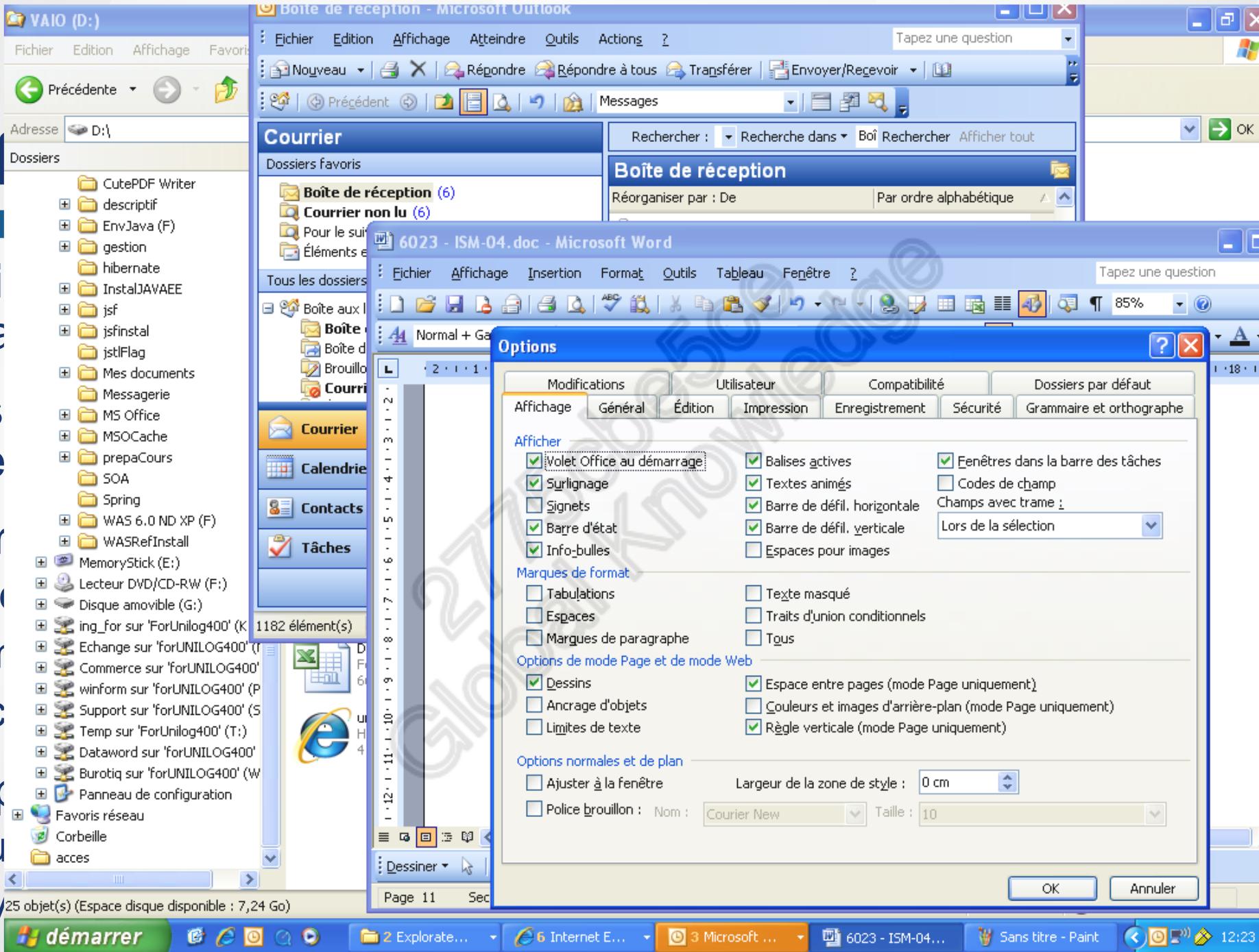
# LA COUCHE PRÉSENTATION : Les interfaces mode caractère

- « Le terme d'interface utilisateur regroupe tous les composants logiciels qui se présentent sur l'écran. »
- Les interfaces sont programmées en C.
- Seuls les programmes peuvent accéder à l'écran.
- L'évolution de la couche de présentation de l'information dépend de l'évolution de l'interface utilisateurs.



# LA C interface

- Les icônes d'images
- Elles traitent
- La première fenêtre icône ascenseur
- Elle peut couvrir ouvrir
- OUVRE



# LA COUCHE PRESENTATION : Les interfaces Web

- Les interfaces Web sont des interfaces graphiques.
- Lorsqu'il faut faire saisir des informations à l'utilisateur, on retrouve des présentations simples.



Lorsque l'application permettant cet affichage s'exécute sur le serveur et que le poste de l'utilisateur n'affiche que le résultat, on parle de **client léger**

The screenshot shows a web page titled "Training- Catalogue des Formations - Windows Internet Explorer" at the URL <http://www.formation.fr>. The page is a "Demande de devis" (Request for Quotation) form. On the left, there's a sidebar with links like "Informations pratiques", "Demande du catalogue papier", "Demande de devis", "Bulletin d'inscription papier", "Plan d'accès", and "Taille de l'écran". Below that is a "Inscriptions réalisées" section with a shopping bag icon and "0 inscriptions". At the bottom of the sidebar is a "Les promotions du jour" section with a gift icon and "Voir les promotions". The main content area has a yellow header "Demande de devis". Underneath, it says "Demande de solution de formations". There are two columns of input fields: "Vos coordonnées" and "Description de la demande". The "Vos coordonnées" column includes fields for "Civilité\*", "Nom\*", "Prénom\*", "e-mail\*", "Téléphone\*", "Société\*", "Adresse\*", "Code Postal\*", "Ville\*", and "Fax". The "Description de la demande" column includes fields for "Thèmes de la formation", "Objectifs (contexte et résultat recherché)\*", "Période envisagée", and "Contraintes particulières". At the bottom right is a large "Envoyer" (Send) button. A note at the bottom states "Les champs marqués d'un astérisque \* sont obligatoires". The status bar at the bottom shows "Terminé", "démarrer", "Explorateur Win...", "Internet Explorer", "Microsoft Office...", and "6023 - ISM-04.doc...". The date and time "12/02" are also visible.

# LA GOUACHE PRÉSENTATION

- Les intégrations de l'utilisateur
- Les intégrations prisées
- L'intégration en anglais
- A savoir...

Netvibes - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente □ Rechercher FAVORIS □

Adresse <http://www.netvibes.com/> OK

Liens Hotmail Personnaliser les liens Windows Windows Media Extranet LogicaCMG Webmail SESAME LOGIN

Ajouter du contenu | Rechercher

Entrez ici le titre de votre page

General (20) Internet Fun

Personalize this page and make it your startpage

Cambridge, OH

Mardi Mercredi Jeudi Vendredi

49°/56° 32°/51° 29°/44° 31°/43°

Prévisions météo sur 10 jours Powered by: The Weather Channel

Recherche Web

Google Yahoo! Ask Live

Veuillez saisir une recherche

Liste de tâches

pas de tâche Nouvelle tâche

Webnote

Entrez votre texte ici

Digg

All Topics All News Videos Images Friends Settings

74 SCI Myth of American Zombies Dispelled - Sleep Well. 8 Comments

120 E The Top 10 Weirdest Rock 'n' Roll Deaths 17 Comments

177 E Scientists slam publishers over Wikipedia ban 11 Comments

201 British man auctions entire life on eBay 26 Comments

271 TIME Invents Facts to Claim Americans Support Bush's Spying 81 Comments

(20) Google News - World

World U.S. Business Sci/Tech

Young Tibetans reject Dalai Lama's lead - CNN International - il y a 48 minutes

The Irrawaddy News Magazine Young Tibetans reject Dalai Lama's lead CNN...

AP: Ukrainian police officer dies in hospital after clashes, say police - CNN International - il y a 59 minutes

Christian Science Monitor AP: Ukrainian police officer dies in hospital after...

Obama to speak out on race issue - Hindu - Il y a 1 heure

NY1 Obama to speak out on race issue Hindu - 1 hour ago PHILADELPHIA...

US, Russian officials take conciliatory tone in missile talks - Monsters and Critics.com - il y a 38 minutes

Washington Post US, Russian officials take conciliatory tone in missile talks ...

Many dead in Iraq shrine city blast - Aljazeera.net - Il y a 3 heures

Turkish Press Many dead in Iraq shrine city blast Aljazeera.net -

Flickr Most Interesting Photos

Zone inconnue

12:17

http://www.flickr.com/photos/69445849@N00/2341079177

Démarrer Boîte d... RE: co... 2 Exp... 6023 - ... D:\Sup... netvibe... Netvib... FR 12:17

# LA COUCHE PRÉSENTATION : Les interfaces mobiles

- Les tablettes et les smartphones ont eux aussi apporté des nouveautés dans la présentation aux utilisateurs. De nouveaux affichages ont vu le jour, souvent « collé » au système d'exploitation sur lequel l'application est présentée.
- A l'ère de l'**Internet Mobile**, quelles que soient vos préférences en termes de marque et de modèle, en matière de smartphones, on peut considérer qu'il y a un avant et un après Iphone. Dès sa sortie, le smartphone d'Apple a fait un véritable carton... et a boosté la pratique de l'internet mobile ! Les autres fabricants lui ont tous emboîté le pas avec plus ou moins de succès !



# LA COUCHE PRÉSENTATION : Les interfaces mobiles

- Les Google Phones se sont distingués comme de véritables bijoux multimédias. Ces mobiles, équipés du système d'exploitation de Google « **Android** » se sont imposés comme de véritables smartphones tournés vers l'expérience Internet avec des browsers performants qui permettent une navigation plus rapide et plus fluide.
- L'offre de Microsoft, permet aujourd'hui d'avoir le **rendu visuel commun** aux Tablettes (surface MS &), aux téléphones, aux PC.



# LA COUCHE PRÉSENTATION : Les interfaces mobiles

- La tendance est déjà bien amorcée, depuis le lancement du premier Iphone en 2007, maintenant la quasi-totalité de l'offre mobile est équipée d'**écrans tactiles**.
- Plus rapide et plus intuitif, le tactile remporte les suffrages des utilisateurs ! Par ailleurs, les systèmes d'exploitation des mobiles, de plus en plus performants et de plus en plus poussés pourraient permettre aux smartphones de devenir de véritables mini-portables intégrant notamment des clients mails, la gestion de documents bureautiques ou l'accès à des applications en cloud-computing.

**Android OS evolution**

Oct 2008 Jan 2010 Dec 2010 Nov 2011



**iOS evolution**

2007 2008 2009 2010 2011



# LA COUCHE PRESENTATION : Le portail

- On appelle "**portail d'entreprise**" une plate-forme donnant accès à des données de l'entreprise ainsi qu'à des ressources du système d'information regroupées au sein d'une **interface**
  - Il s'agit d'une interface

- *Le portail d'entreprise* est une plate-forme d'information de l'entreprise et de ses partenaires.

- L'enjeu du portail est de faciliter l'accès à l'information

- Principaux avantages
  - Vélocité
  - Visibilité
  - Jeux de rôle
  - Collaboration
  - C



données du système  
et éventuellement les

sein du système

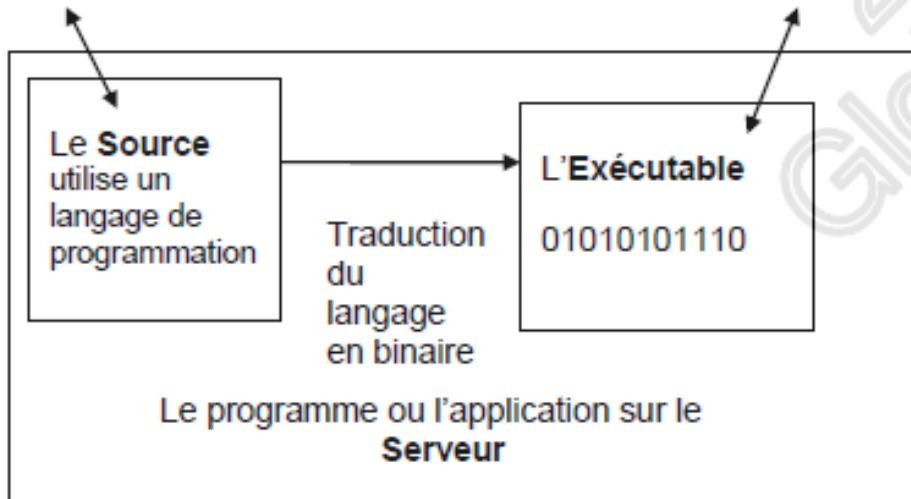
# LA COUCHE TRAITEMENT : Les langages de programmation

- L'exécution d'un traitement par un ordinateur nécessite de décrire ce traitement à l'aide d'un langage approprié.
- Tout programme existe sous deux états :
  - le **source** décrit le traitement à l'aide du langage de programmation choisi, explicite pour l'informaticien mais non pour la machine,
  - l'exécutable (module chargeable ou load-module) résulte de la traduction du source en binaire, seul format exploitable par la machine.
    - Le **binaire** est constitué d'une succession de " 0 " et de " 1 " (chiffres binaires, binary digits ou bits) qui traduisent l'absence ou la présence d'influx électriques interprétables par le processeur.
    - L'exécutable est obtenu via un mécanisme nommé **compilation**.

# LA COUCHE TRAITEMENT : Les langages de programmation



L'Informaticien



Un source utilisant un langage de Script, n'est pas traduit en exécutable au préalable.

Lorsque le script est exécuté sur la machine, celle-ci l'interprète à la volée

Source avec langage Script

Risque de plantage + élevé qu'avec 1 compilé

# LA COUCHE TRAITEMENT : Les générations de langage - 1 -

- On distingue aujourd'hui cinq générations de langages.
- **La première génération** est le **langage machine**, ou **code machine**.  
On parle aussi de **langage natif**.
  - Il est composé d'instructions et de données à traiter codées en binaire.
  - C'est le seul langage qu'un ordinateur puisse traiter directement.
    - Voici à quoi peut ressembler un programme en langage machine :

```
A1 01 10 03 06 01 12 A3 01 14
```

Il s'agit de la représentation hexadécimale d'un programme permettant d'additionner les valeurs de deux cases mémoire et de stocker le résultat dans une troisième case.  
On voit immédiatement la difficulté d'un tel langage...

# LA COUCHE TRAITEMENT : Les générations de langage - 2 -

- La deuxième génération :
  - est le langage assembleur :
    - le code devient lisible et compréhensible par un plus grand nombre d'initiés.
    - Il existe en fait un langage assembleur par type de processeur.
    - Le programme précédent écrit en assembleur donnerait ceci :

```
MOV AX, [0110]
ADD AX, [0112]
MOV [0114], AX
```

Il reste utilisé dans le cadre d'optimisations, mais a été supplanté en popularité par les langages plus accessibles de troisième génération.

# LA COUCHE TRAITEMENT : Les générations de langage - 3 -

- **La troisième génération** : utilise une syntaxe proche de l'anglais.
  - Proposés autour de 1960, ces langages ont permis un gain énorme en lisibilité et en productivité.
  - Ils ne dépendent plus du processeur, comme c'était le cas des générations précédentes, mais d'un compilateur spécifique du processeur.
  - L'idée de portabilité des programmes était lancée.
  - La plupart des langages de programmation actuels sont de 3<sup>ème</sup> génération.
    - Ada, Algol, Basic, **Cobol**, Eiffel, Fortran, **C**, **C++**, **C#**, **Java**, Perl, Pascal, Python, Ruby, ...
    - Cette génération couvre d'ailleurs tant de langages qu'elle est souvent subdivisée en catégories, selon le paradigme particulier des langages (Objet, Fonction...).

# LA COUCHE TRAITEMENT : Les générations de langage - 4 -

- **Les langages de quatrième génération**, abrégés L4G, associée à des génération automatique (accès aux bases de données) intégrant la gestion de IHM et en proposant un langage moins technique, plus proche de la syntaxe naturelle.
- Ils sont conçus pour un travail spécifique :
  - gestion de base de données (Microsoft Access, SQL, PL/SQL, TransactSQL ),
  - production graphique (Postscript),
  - création d'interface (4D).
  - Forms d'Oracle
  - VB de Microsoft, PowerBuilder, Delphi...

# LA COUCHE TRAITEMENT : Les générations de langage - 5 -

- La cinquième génération de langages sont des langages destinés à résoudre des problèmes à l'aide de contraintes, et non d'algorithmes écrits.
- Ces langages reposent beaucoup sur la logique et sont particulièrement utilisés en **intelligence artificielle**.
- Parmi les plus connus, on trouve Prolog, dont voici un exemple :

```
frère_ou_sœur(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.  
parent(X, Y) :- père(X, Y).  
parent(X, Y) :- mère(X, Y).  
mère(trude, sally).  
père(tom, sally).  
père(tom, erica).  
père(mike, tom).
```

Il en résulte que la demande suivante est évaluée comme vraie :

```
?- frère_ou_sœur(sally, erica).  
oui.
```

Ce qui signifie que Sally et Erica sont sœurs. En effet, Sally et Erica ont le même père (Tom).

# LA COUCHE TRAITEMENT : Les langages d'échanges de données - EDI

- Techniquement : des messages de toutes sortes (commerciaux, logistiques, financiers... ) transitent via un réseau entre partenaires  
Ils sont rédigés dans une **syntaxe spécifique**
- Un logiciel permet à chaque partie de traduire ces messages et d'intégrer les données à leur applicatifs évitant ainsi toute ressaisie et tout risque d'erreur
- Le langage **EANCOM** permet aux entreprises du commerce de communiquer
- Il s'appuie sur une structure et une liste de codes établie pour le langage des Nations Unies : **EDIFACT**, mais également sur les codes EAN qui doivent être parfaitement en place dans les entreprises.
- En Logistique : EANCOM est souvent associé au code à barre.



# LA COUCHE TRAITEMENT : Les langages d'échanges de données – XML

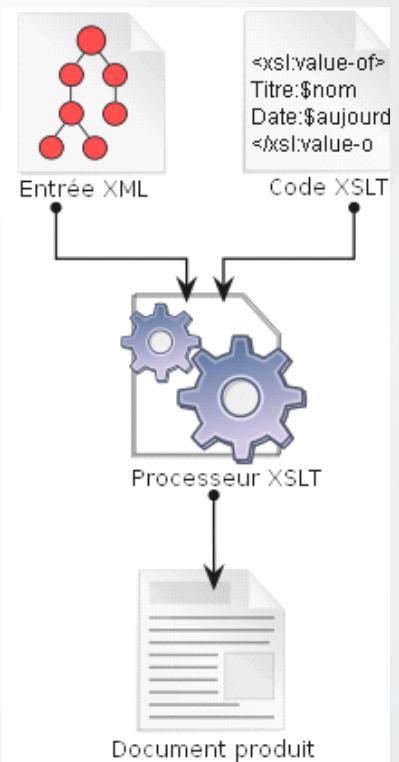
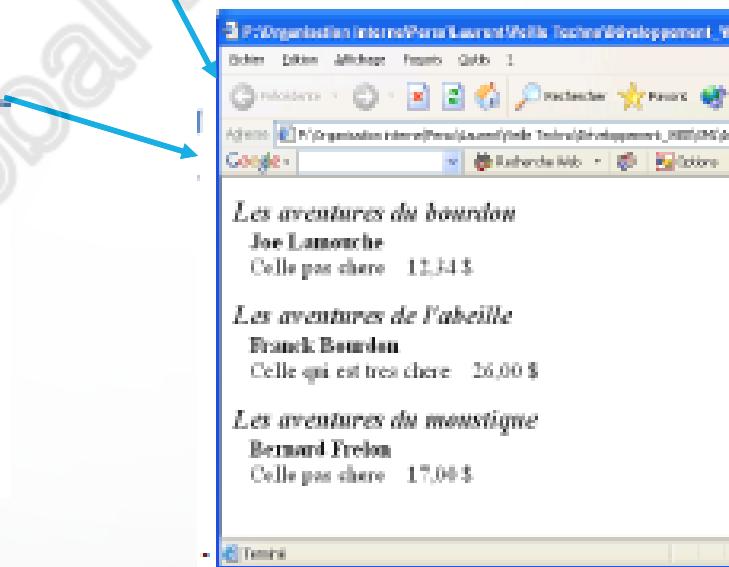
- Le standard **XML** est une recommandation de l'organisation W3C (février 1998)
- XML a été élaboré pour publier des documents sur le Web. Il tend de plus en plus à servir de **format d'échange** entre **applications** et plus généralement entre **systèmes d'information**, sur des systèmes hétérogènes
  - XML permet de définir sa propre grammaire
- Afin de manipuler les données entre partenaires, on associe à XML une définition du document une **DTD** ou un **schéma XML**.
- **XML gère les données et non leur représentation.**
  - La mise en page est prise en charge par **XSL** (eXtensible Stylesheet Language).
  - On peut aussi se servir d'XML, pour générer différents documents comme des fichiers textes ou PDF

# LA COUCHE TRAITEMENT : Les langages d'échanges de données – XML

```
<?xml version="1.0" ?>
<!-- Nom de fichier : Inventaire.xml -->
<?xmlstylesheet type="text/css" href="Inventaire02.css" ?>
<INVENTAIRE>
    <LIVRE>
        <TITRE>Les aventures du bourdon </TITRE>
        <AUTEUR> Joe Lamouche </AUTEUR>
        <RELIURE> Celle pas chere </RELIURE>
        <PAGES> 100 </PAGES>
        <PRIXT> 12,34 $ </PRIXT>
    </LIVRE>
    <LIVRE>
        <TITRE>Les aventures de l'abeille </TITRE>
        <AUTEUR> Franck Bourdon </AUTEUR>
        <RELIURE> Celle qui est tres chere </RELIURE>
        <PAGES> 50 </PAGES>
        <PRIXT> 26,00 $ </PRIXT>
    </LIVRE>
</INVENTAIRE>
```

## Feuille de style

```
/* Nom de fichier : Inventaire02.css */
LIVRE
    {display:block; margin-top:12pt; font-size:15pt}
    TITRE
        {display:block; font-
```



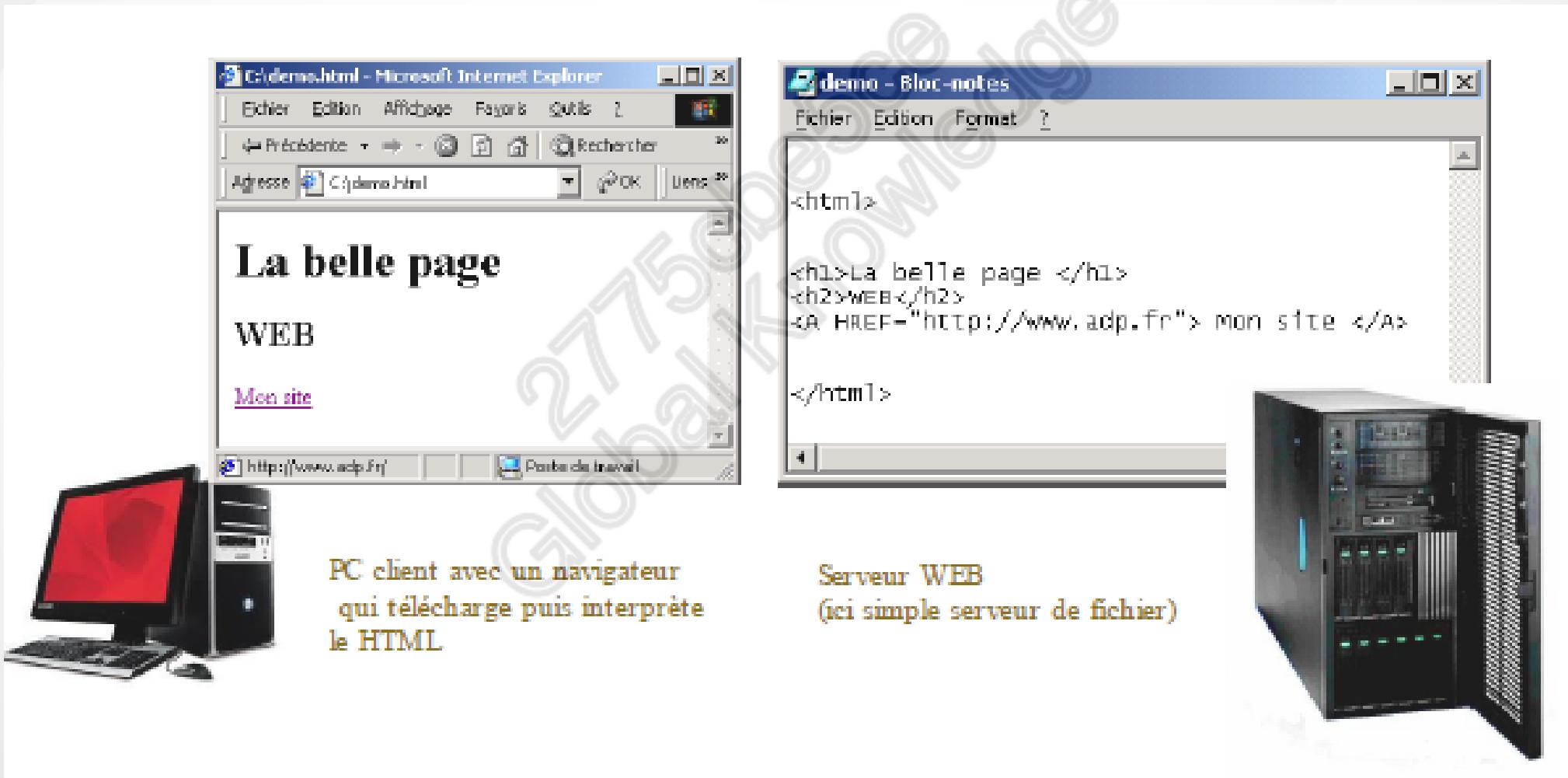
# LA COUCHE TRAITEMENT : Les langages d'échanges de données – XML

- Le langage XML dans le monde du Web facilite :
  - La recherche d'information
  - L'intégration des données de sources variées
  - La manipulation des données sur le poste client
  - La création de vues multiples de données
  - Les mises à jour incrémentales
  - La publication multi-langues
  - La navigation avancée
  - La mise en œuvre des Web Services
- **XQuery** est un langage de requête informatique permettant :
  - D'extraire des informations d'un document XML.
  - D'effectuer des calculs complexes à partir des informations extraites et de reconstruire de nouveaux documents ou fragments XML.

# LA COUCHE TRAITEMENT : Les langages du Web

- Le premier langage de présentation de page Web a été **HTML** (Hyper Text Markup Language).
- Le langage **XHTML**, pour Extensible HTML est une reformalisation de HTML.
- Le HTML est la base de tout développement Web, la dernière révision du langage est la 5, d'où l'utilisation de HTML 5.
  - Le langage HTML s'associe en parallèle à **CSS** (Cascading Style Sheets).
- **HTML** permet de coder le **contenu** (titre, paragraphe, menu...) pendant que le **CSS** le **met en forme** (couleur, choix des polices de caractère, disposition des éléments ...).

# LA COUCHE TRAITEMENT : Les langages du Web



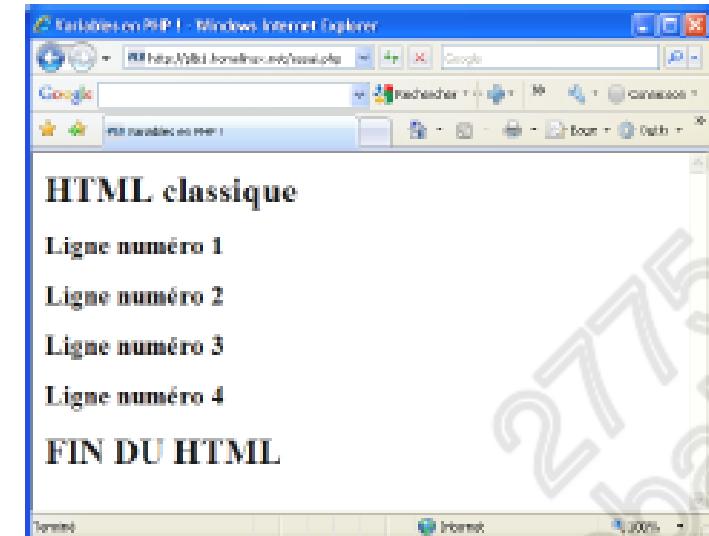
# LA COUCHE TRAITEMENT : Les langages du Web

- Pour ajouter des contrôles de saisies, des calculs, des manipulations de dates, des mises en forme dynamiques de pages, on utilise d'autres technologies : le **Dynamic HTML (DHTML)**, **JavaScript**.
- **JavaScript** est un langage interprété par le navigateur, ce langage dit langage de **script** (car il n'est pas compilé) ne pourra ni lire, ni écrire de données sur le poste local pour des raisons de sécurité.
- La page Web est stockée sur le serveur pour être envoyée aux utilisateurs, on parle d'un site Web statique (Origine du Web).

# LA COUCHE TRAITEMENT : Les langages du Web

- Dans un site **Web dynamique**, la page est construite via un **programme** stocké sur le serveur. Il existe différentes technologies pour écrire ce programme :
  - La technologie Java avec **JSP** (Java Serveur Page), .
  - La technologie Microsoft avec **ASP.Net** .
  - La technologie **PHP** (Hypertext Preprocessor).
- Au début des sites Web dynamiques : les langages PERL et C ont été les plus utilisés, mais chaque serveur pouvait exécuter n'importe quel langage qu'il supportait. Ces programmes se nommaient des CGI, (common gateway interface). Chez Microsoft existait la technologie ASP.

# LA COUCHE TRAITEMENT : Les langages du Web



```
<html>
  <head>
    <title> Variables en PHP </title>
  </head>
  <body>
    <H1> HTML classique
    <h2>
      <?php

        for($i = 1; $i < 5; $i++)
          echo "<p>Ligne numéro '$i'</p>" "\n";
      ?>
    </h2>
    FIN DU HTML
  </h1>
  </body>
</html>
```

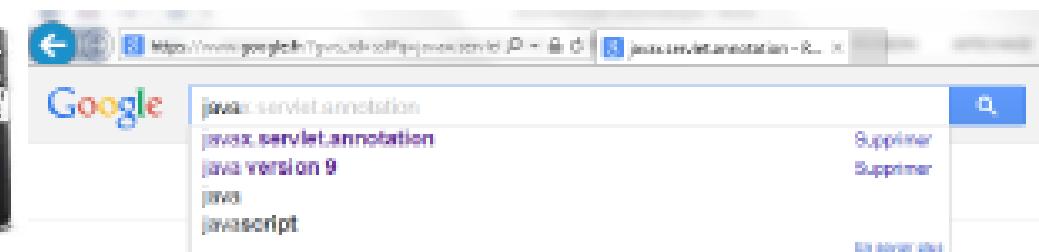


# LA COUCHE TRAITEMENT : Les langages du Web

- L'une des technologies émergentes de JavaScript est connue sous le nom d'**AJAX**.
  - AJAX est un acronyme anglais signifiant **Asynchronous Javascript And XML** (Javascript et XML asynchrones). Il s'agit d'envoyer et recevoir des requêtes HTTP, pour communiquer avec des **scripts situés sur un serveur**.
  - Cela permet d'échanger des informations sous différents formats (XML, HTML ou texte), mais son principal attrait est sa nature « asynchrone » : tout cela peut se faire sans recharger la page et sans bloquer l'utilisateur.
  - C'est ce qui permet de mettre à jour certaines parties d'une page sur la base d'évènements déclenchés par l'utilisateur (messagerie Gmail ou Yahoo, complétude à la saisie avec la barre Google, ...).

# LA COUCHE TRAITEMENT : Les langages du Web

- Les deux fonctionnalités combinées sont les possibilités de :
  - faire des requêtes vers le serveur sans avoir à recharger la page ;
  - analyser et travailler avec des documents XML.
- AJAX est une des technologies phares du mouvement **Web 2.0** qui définit les interfaces riches permettant à l'internaute une plus grande interactivité avec la page Web.
- La mouvance **Web 3.0** est le web **sémantique** (Un modèle qui permet aux données d'être partagées, réutilisées entre plusieurs applications, entreprises et groupe utilisateurs) et/ou IoT ***Internet of Things*** l'internet des Objets

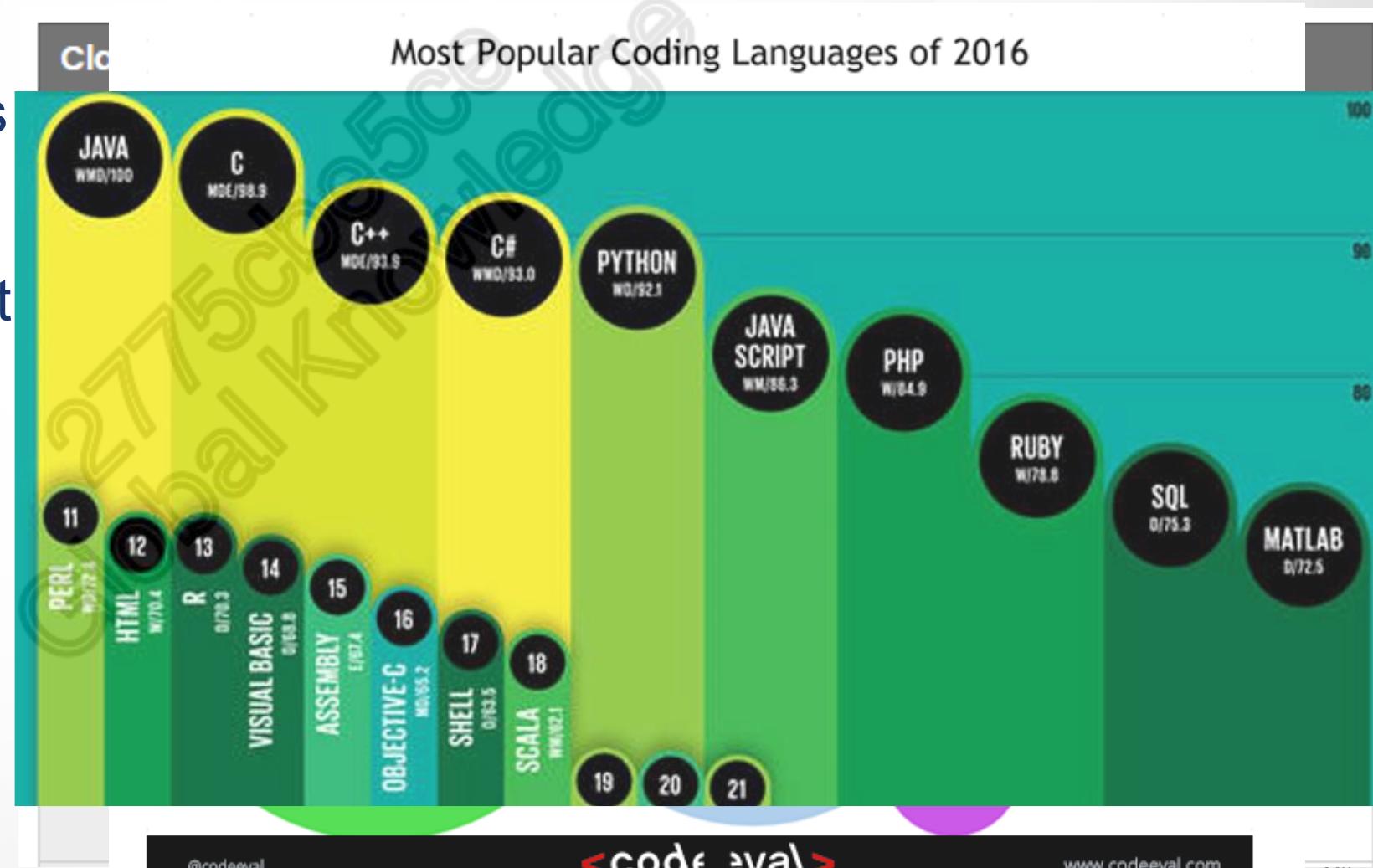


# LA COUCHE TRAITEMENT : Les langages du Web

- **jQuery** est une bibliothèque libre JavaScript qui porte sur l'interaction entre JavaScript (avec AJAX) et HTML.
  - Son but est de simplifier des commandes communes de Javascript.
  - Cet ensemble contient des fonctionnalités comme le parcours et la modification du document, la gestion des événements, des effets et animations, la manipulation de feuilles de style, des utilitaires (version du navigateur...), des plugins.
- De nombreux **langages voient le jour encore aujourd'hui**, ils sont créés pour répondre à certaines fonctionnalités. Les derniers en date : Dart, Go, Scala, Groovy, mais aussi des frameworks Angular ...

# LA COUCHE TRAITEMENT : Les langages du Web

- Chaque année, un classement des langages les plus populaires est édité par divers cabinets et instituts, qui obtiennent les informations en fonction des demandes sur les moteurs de recherche du Web,



# LA COUCHE TRAITEMENT : Les outils de développement

- Il existe deux méthodes pour développer des applications.
  - La première consiste à utiliser un simple éditeur de texte, un compilateur et l'environnement d'exécution.
  - La deuxième, de plus en plus utilisée pour une **meilleure productivité** et un confort de travail, est la manipulation de « studios de développement ».
    - Les « studios de développement » peuvent être de type **IDE**, Integrated Development Environment ou de type **RAD** Rapid Application Development.
    - Ce sont des logiciels multifonctions réunissant dans la même interface tous les outils dont le développeur a besoin. Les outils les plus manipulés sont :
      - L'éditeur de source / L'explorateur de code
      - Le compilateur, / Le débogueur
      - Le générateur d'interfaces graphiques
      - Le gestionnaire de projet

# LA COUCHE TRAITEMENT : Les outils de développement

- L'IDE est un logiciel tourné vers la productivité au niveau du code source.
  - Le programmeur peut, sans quitter son environnement, exécuter toutes les étapes de la création d'une application.
  - L'IDE intègre souvent des fonctionnalités supplémentaires telles que les explorateurs de codes ou les assistants à la programmation.
- RAD est destiné à faciliter l'interface graphique. Même si la réalisation d'une IHM devient plus facile il est nécessaire d'avoir le code source.
  - En général, les RAD offrent tous un module IDE minimal.
- Conclusion : la frontière entre les deux outils est limite. Tous les RAD intègrent des fonctions IDE et les IDE les plus poussés proposent des fonctions de RAD.
  - Lorsqu'un outil offre le meilleur des deux mondes, on parle alors de VDE, Visual Development Environment. Cependant, le terme le plus utilisé, quel que soit l'outil, reste IDE.

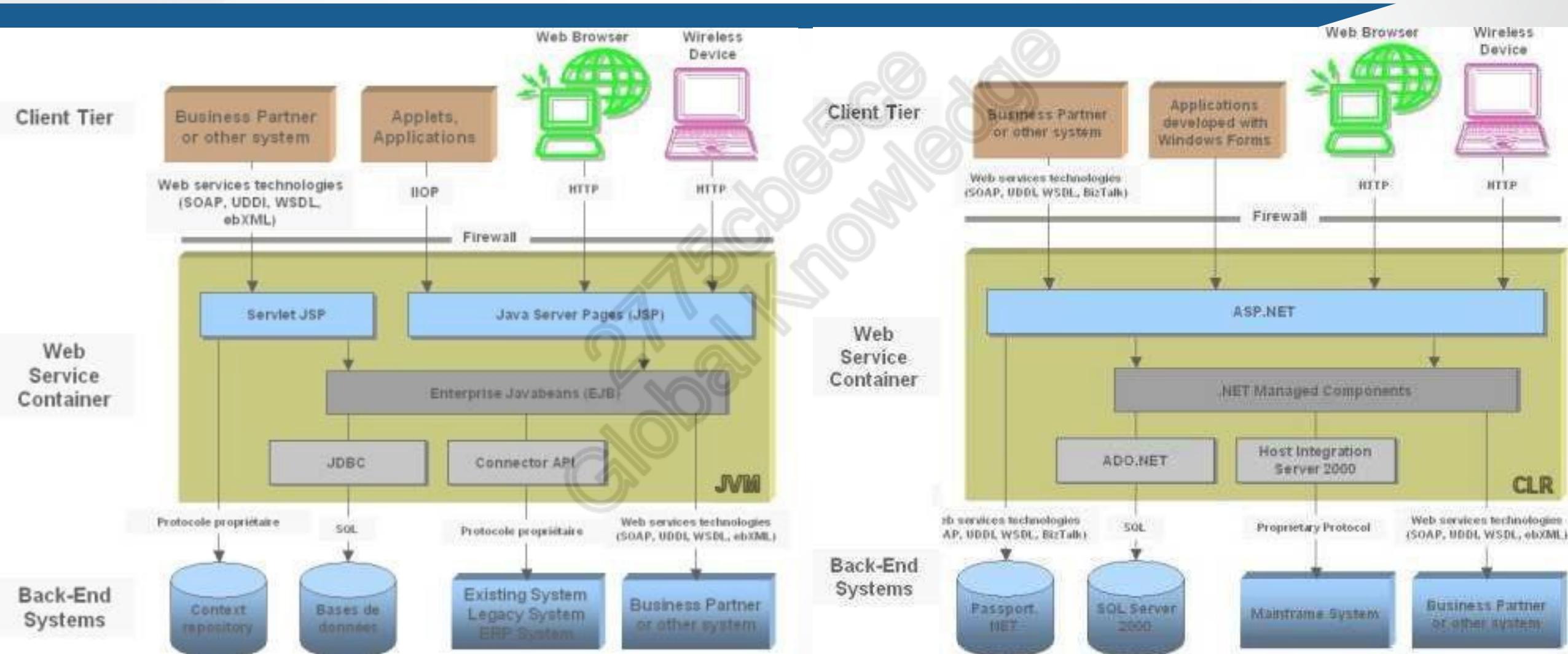
# LES PLATES-FORMES JAVA EE ET .NET : Présentation

- L'évolution actuelle tend à considérer que **le langage** de programmation, même sous sa forme la plus aboutie du L4G, **n'apporte qu'une réponse partielle aux besoins des équipes de développement** qui doivent systématiquement s'équiper de produits complémentaires pour répertorier les ressources utilisées (bases de données, annuaires), établir des connexions applicatives entre plusieurs ordinateurs (middleware)...
- Les éditeurs apportent une réponse à ce besoin plus global en **« packageant » un ensemble de produits** dont ils garantissent la compatibilité.

# LES PLATES-FORMES JAVA EE ET .NET : Présentation

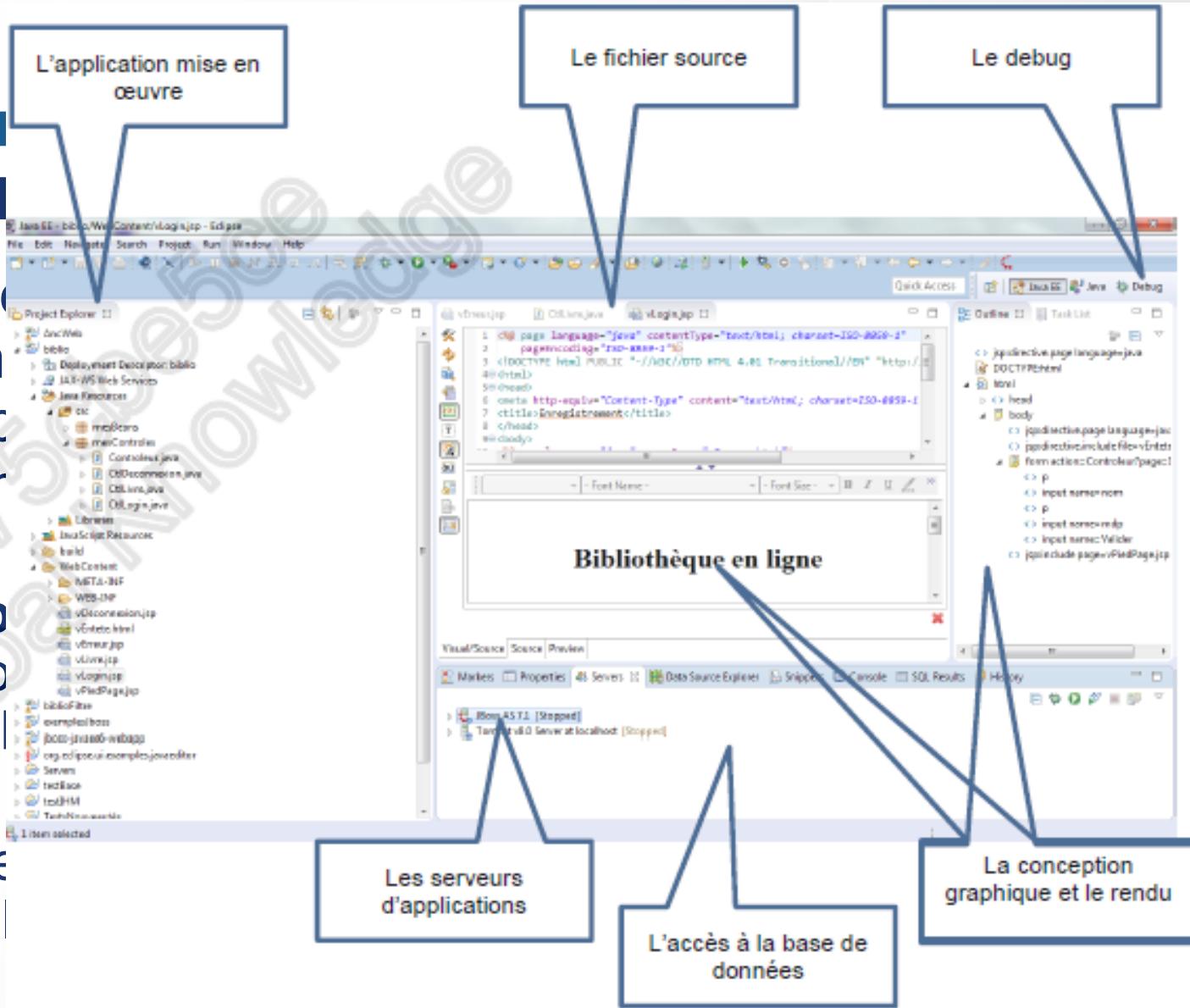
- Les offres disponibles :
  - **Java EE** (anciennement **J2EE**, Java Enterprise Edition) constituée sous l'égide d'Oracle depuis son rachat de Sun
  - **.Net**, de Microsoft.
- L'utilisation de ces plates-formes nécessite :
  - pour la conception/réalisation : des outils de développement, les fameux studios ou IDE.
  - pour les tests : des serveurs d'applications, afin d'exécuter l'application développée. Des émulateurs pour simuler un smartphone. Des liens vers des services.

# LES PLATES-FORMES JAVA EE ET .NET : Présentation



# LES PLATES-FORMES JAVA EE ET .NET : Les outils

- Microsoft propose **Visual Studio**
- Sun/Oracle a laissé l'ensemble de Java EE. Quelques-uns ont One Studio de Sun, Workshop d'IBM (anciennement WebSphere)
  - Aujourd'hui il existe un produit développeurs Java, c'est **Eclipse**, disposition du monde Open Source mis en open source en 2000 :
  - Ces IDE, supportent plusieurs quel que soit le produit, le développement pour mettre en œuvre des applicat



# LES FRAMEWORKS

- Le terme framework désigne un « cadre de travail » pour le développeur
  - Il s'agit en général d'une bibliothèque logicielle qui simplifie le développement en proposant des briques de plus haut-niveau (gain de temps)
  - L'utilisation du framework doit également permettre d'écrire un programme de qualité (lisibilité, performance, ....).
  - Un framework est souvent accompagné d'un environnement logiciel qui permet de bien l'utiliser
- Il y aura des frameworks spécifiques à Java et d'autres à .Net, d'ailleurs on parle du framework .net.

# LES FRAMEWORKS

- Pour vous donner un exemple de ce que peut être un framework, nous avons présenté les 3 couches principales d'une application (d'un logiciel) :
  - La présentation à l'utilisateur
  - Les traitements à mettre en œuvre (pour présenter les données)
  - Les données d'entreprise
- En tant que développeur, il y aura toujours à effectuer du code pour ces 3 couches mais aussi à mettre en œuvre le code nécessaire pour faire communiquer ces 3 couches.
- Soit on programme tout, soit on se fait aider d'un framework qui aura déjà codé la communication

# LES FRAMEWORKS

- Le type de framework qui peut aider à mettre en œuvre ces 3 briques, est défini comme un framework MVC : Model View Controller :
  - le Modèle étant les données d'entreprise,
  - la Vue, les présentations,
  - le Contrôleur, les traitements.
- Quand on dispose d'une solution pour résoudre un problème informatique, on parle de **Design Pattern**, de modèle de conception, c'est aussi un ensemble de bonne pratique

# LES FRAMEWORKS

- En développement logiciel, vous serez souvent confronté, aux mêmes problèmes, d'autres auront peut-être capitalisé, et auront réfléchi à une solution avant vous.
- De nombreux pattern existent, ensuite il faudra choisir le framework implémentant cette solution.
- Dans le monde Java, les frameworks MVC les plus utilisés sont Struts, JSF (Java Server Faces), Spring MVC.
- Pour définir l'accès aux données d'entreprise, il faudra utiliser le design pattern DAO (Data Access Object) et choisir un framework.
  - Hibernate a été le plus utilisé dans la technologie Java.

# SOMMAIRE

---

- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# LE PROJET INFORMATIQUE

*Objectifs : décrire les différentes démarches et méthodes de conduite d'un projet informatique.  
Présenter les acteurs et leurs rôles dans le projet.*



# DEFINITIONS ET OBJECTIF D'UN PROJET

Un **projet** est constitué par l'ensemble des actions à entreprendre pour atteindre un objectif donné.

Il est caractérisé par :

- un budget,
- un calendrier,
- une organisation (ressources humaines, logicielles, techniques),
- des produits (logiciel exécutable, manuels d'utilisation et d'exploitation, documentation).

Le projet est terminé lorsque les objectifs sont atteints.

# DEFINITIONS ET OBJECTIF D'UN PROJET

- Pour une bonne maîtrise et un bon déroulement d'un projet, il faut avoir :

Une méthodologie = ensemble des actions à entreprendre pour parvenir à un but. Elle répond aux questions : **QUE FAUT-IL FAIRE, QUI LE FAIT et QUAND ?**

Une méthode = ensemble de démarches raisonnées, suivies pour parvenir à un but. Elle répond à la question : **COMMENT FAUT-IL LE FAIRE ?**

Une méthode de projet recouvre 3 aspects : **Démarche, Approche, Conduite**

- Pour faciliter la conduite de projet, on dispose de démarches de projet, L'approche, détient des techniques de représentation, de modélisation. La conduite est aussi appelée **pilotage de projet**.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Définitions

Le **Maître d'Ouvrage** est une personne, physique ou morale qui a un besoin « métier ». Il finance et commande un ouvrage qui doit répondre à :

- Des spécifications.
- À des exigences de qualité, coûts et délais.

L'**ouvrage** est un ensemble cohérent de fournitures, livrables et de services associés. C'est le **résultat d'un projet**.

Le Maître d'Ouvrage délégué

- Veille à l'acceptation des modifications
- Mesure les conséquences



**Business Analyst**

=

Interface entre les opérationnels, les informaticiens et l'encadrement pour les projets initiés afin de répondre à des changements

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Définitions

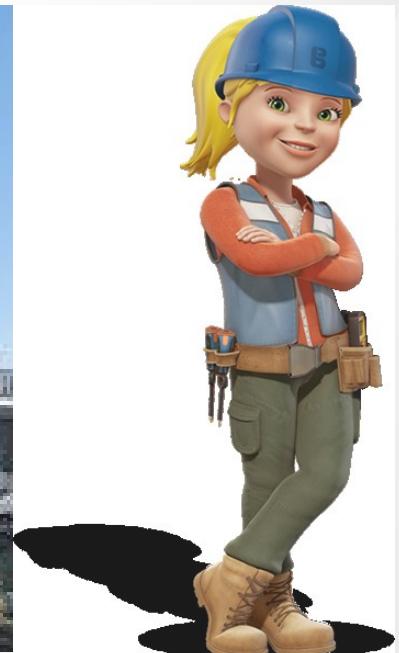
Le **Maître d'oeuvre** est une personne, physique ou morale, **responsable** de l'ensemble des **travaux de réalisation** d'un ouvrage (conception, construction et mise en oeuvre). Il assure la mise à disposition des moyens, il livre le projet.

**L'oeuvre** est un ensemble des **travaux nécessaires à la réalisation** d'un ouvrage.  
Le travail de production d'une chose.

Le Maître d'oeuvre délégué

- Délivre un produit défini
- Met à disposition des ressources

→ Sous-traitance



# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Types & responsabilités dans le projet

Le projet informatique définit 2 types de responsabilité :

- la **maîtrise d'ouvrage** a en charge la définition **fonctionnelle** de la future application
- la **maîtrise d'oeuvre** a en charge la conception **technique** et la réalisation
- la **maîtrise d'ouvrage** est garante de la **bonne mise en production** de l'application donc assume la responsabilité des tests

Le projet est donc marqué par 2 changements de responsabilité, qui nécessitent contractualisation :

- le **cahier des charges** définit les attendus de la maîtrise d'ouvrage vis-à-vis de la maîtrise d'oeuvre
- la **recette** formalise la procédure de réception et d'acceptation (ou de non acceptation) contractuelle du produit fourni par la maîtrise d'oeuvre



# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Les démarches

Les démarches de projet, découpent en cinq phases principales, tout processus de développement logiciel :

- l'**analyse** du problème,
- la **conception** de la solution,
- la **réalisation** du système,
- l'**installation** du système,
- l'**évaluation** des résultats,
- phases auxquelles il faut ajouter celle de **démarrage**, initialisation indispensable du processus.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Les démarches

Les démarches décrivent de plus les activités transverses menées tout au long du projet :

- **l'estimation** des charges et des délais phase par phase, et la ré-estimation en cours de projet le cas échéant,
- **l'évaluation** des risques inhérents au projet,
- la **planification** des différentes phases et des activités qui les constituent,
- **l'affectation** des ressources à chaque phase et à chaque activité,
- le **suivi** de l'avancement du projet,
- la communication de projet, en particulier vers les futurs utilisateurs et la maîtrise d'ouvrage.

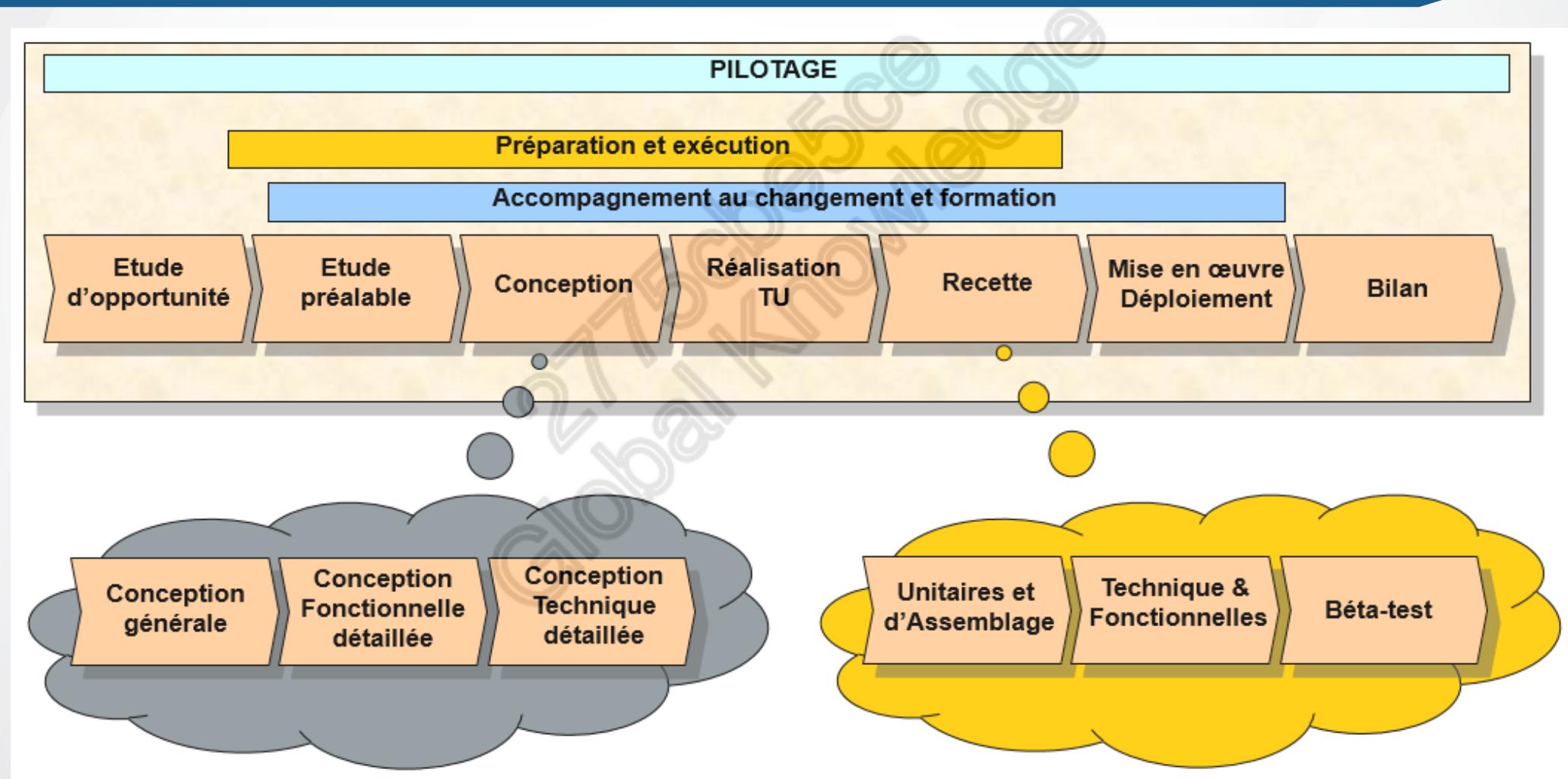
# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Les démarches



Chaque démarche a son propre vocabulaire !  
→ Avoir le même référentiel

|             | MERISE           | AXIAL                | M.C.P.                            | SDM/S                              | UP                     |
|-------------|------------------|----------------------|-----------------------------------|------------------------------------|------------------------|
| Conception  | Schéma directeur | Schéma directeur     |                                   |                                    |                        |
|             | Etude préalable  | Etude initiale       | Etude d'opportunité               | Définition des Besoins du Système  | Définition des besoins |
|             |                  |                      | Etude du système d'information    | Choix d'Architecture du Système    |                        |
|             | Etude détaillée  | Conception générale  | Elaboration du cahier des charges | Spécification Externes du Système  | Analyse                |
| Réalisation | Etude technique  | Conception détaillée | Etude du système informatique     | Spécifications Internes du Système | Conception             |
|             | Production       | Réalisation          | Programmation                     | Programmation Tests                | Implémentation         |
|             |                  |                      | Mise en œuvre                     | Conversion Installation            | Tests                  |
|             | Exploitation     |                      |                                   | Bilan                              |                        |

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Les différentes phases d'un projet



# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases

## Démarrage du projet :

- Le projet est initialisé par une demande ayant pour origine les utilisateurs, le plan informatique, le schéma directeur ou le département informatique. En début de projet, il est nécessaire de mener une brève étude qui permet :
  - d'identifier la portée du projet,
  - d'apprécier le besoin et de vérifier que les avantages attendus sont réalisables,
  - d'identifier les relations entre le domaine concerné, les autres systèmes et le plan informatique,
  - d'identifier des solutions possibles,
  - de proposer des orientations pour l'ensemble du développement, préparer le plan de travail, constituer une équipe...

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : Démarrage

Cette étape de démarrage permet de mieux cerner les attentes du demandeur, de mieux estimer les ressources nécessaires pour entreprendre le développement et de ce fait offre les informations nécessaires à une prise de décision :

- faire,
- faire partiellement,
- différer le développement,
- abandonner.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : Analyse du problème

Cette fonction recouvre une **étude** complète du système **existant** et une **analyse critique** permettant d'en détecter les dysfonctionnements pour ne pas les reproduire. Ceci se fait en étroite relation avec l'utilisateur.

Cette étape contient également **le recueil des besoins** du nouveau système et les exigences des utilisateurs dont on fera un classement.

A partir de l'étude du système existant et des nouveaux besoins, quelques solutions (les scénarios) seront présentées dans un dossier de choix soumis aux décideurs nommés en début de projet.

- On donnera une présentation fonctionnelle et technique de chaque solution ainsi que son coût de développement et une estimation du coût futur d'exploitation.

A l'issue de cette étape, un seul scénario sera retenu pour mener la suite du projet.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : La conception

**Conception de la solution :** suite à l'analyse du problème, plusieurs solutions ont été présentées et une seule a été retenue.

- Il s'agit maintenant d'en faire une description complète (étendue à tout le domaine d'étude) et détaillée.

La conception se fait à deux niveaux.

- D'une part sur un plan fonctionnel, il sera fait une description complète du nouveau système qui sera soumise à l'utilisateur pour validation,
- d'autre part sur un plan technique, cette partie correspondant à l'analyse organique.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : La conception

La **conception fonctionnelle**, ou **conception externe**, s'attache à décrire les fonctionnalités du futur système, c'est-à-dire la partie qui en sera visible par les utilisateurs :

- les fonctionnalités couvertes, *i.e.* les services rendus par l'application,
- les maquettes des écrans,
- les dessins des états.

La **conception organique**, ou **conception interne**, s'attache à décrire la structuration interne des différents composants de l'application, et en particulier :

- le découpage des fonctionnalités en programmes et sous-programmes,
- la structuration des fichiers et des bases, ainsi que leur organisation.

La conception fonctionnelle et la conception organique se mènent le plus souvent en s'appuyant sur une méthode de conception, telles que **Merise** ou **UML**

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : La réalisation – les tests

**Réalisation du système** : cette partie recouvre la construction du produit final respectant les fonctionnalités désirées par l'utilisateur et l'architecture technique mise en place.

La phase de réalisation recouvre les activités de :

- codage des programmes applicatifs,
- production de la documentation de programme, dans le but de faciliter les maintenances ultérieures,
- production des directives d'exploitation du système (politique de sauvegarde et de restauration, par exemple),
- production de guides utilisateurs et d'aides en ligne le cas échéant, voire des dispositifs de formation.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : La réalisation – les tests

La phase de réalisation du système, acceptée au sens large, recouvre aussi les tests de l'application.

- des **tests unitaires**, au cours desquels les différents programmes sont testés indépendamment les uns des autres, tests menés à partir de jeux d'essais, c'est-à-dire de cas d'utilisation conçus spécifiquement dans un but de validation,
- des **tests d'intégration**, tests de fonctionnalités qui enchaînent les différents programmes d'une même unité fonctionnelle, puis les unités fonctionnelles entre elles, menées, eux aussi, sur la base de jeux d'essai,
- des **tests systèmes**, dont l'objectif est de vérifier la capacité du système à supporter la future application,
- des **tests opérationnels**, qui sont menés à partir de données et de contextes de production.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : L'installation

**Installation du système :** lorsque l'utilisateur déclare que le logiciel est suffisamment testé et répond à ses attentes, il sera installé sur la machine réelle et l'on pourra procéder à la formation des utilisateurs finals.

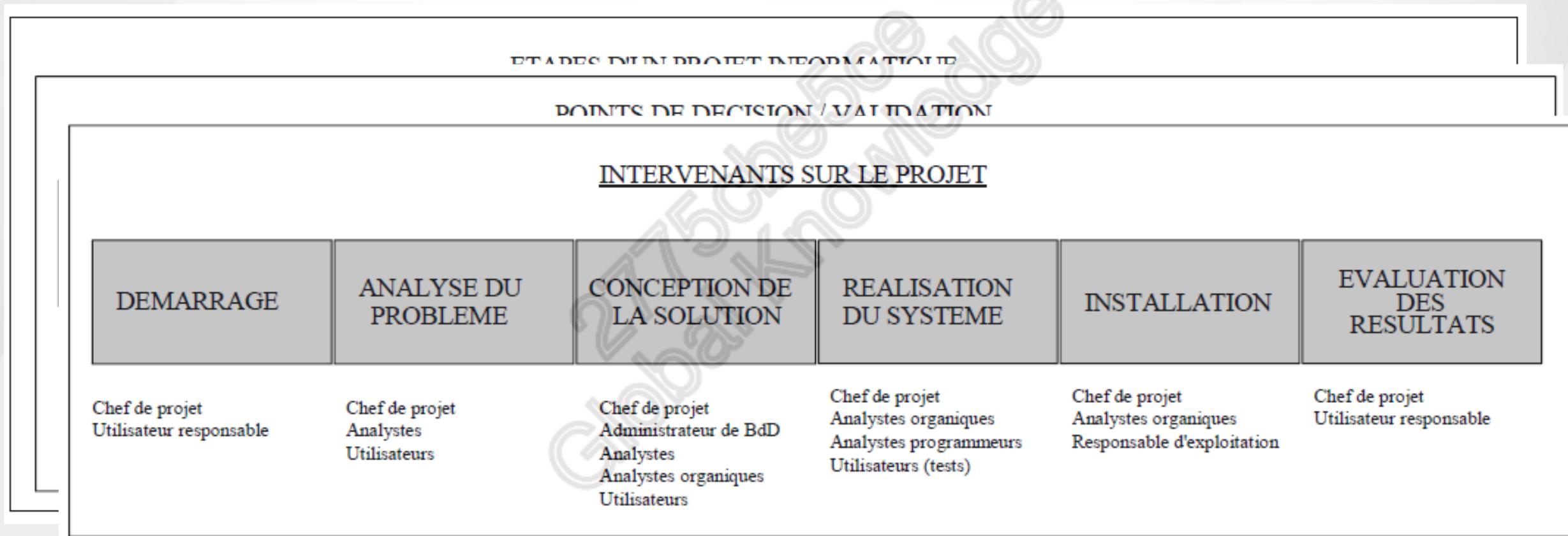
L'installation du système peut nécessiter la conception et le développement de programmes spécifiques, en particulier des programmes permettant de récupérer (ou de convertir) les données existantes le cas échéant.

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases : L'évaluation

**Évaluation des résultats** : en fin de projet, on procédera à un bilan du projet, de son déroulement, de l'atteinte des objectifs aux coûts et délais prévus, mais également un bilan du logiciel : répond-il aux attentes des utilisateurs ? Ceci afin d'envisager d'éventuelles évolutions.

Cette évaluation a aussi comme but de capitaliser l'expérience apportée par ce projet, **capitalisation** d'autant plus importante que l'évolution rapide des techniques, des outils et des méthodes informatiques rend caduques les expériences d'il y a 5 ou 10 ans...

# MAITRISE D'OUVRAGE / MAITRISE D'OEUVRE : Détail des phases



# CYCLES DE DEVELOPPEMENT

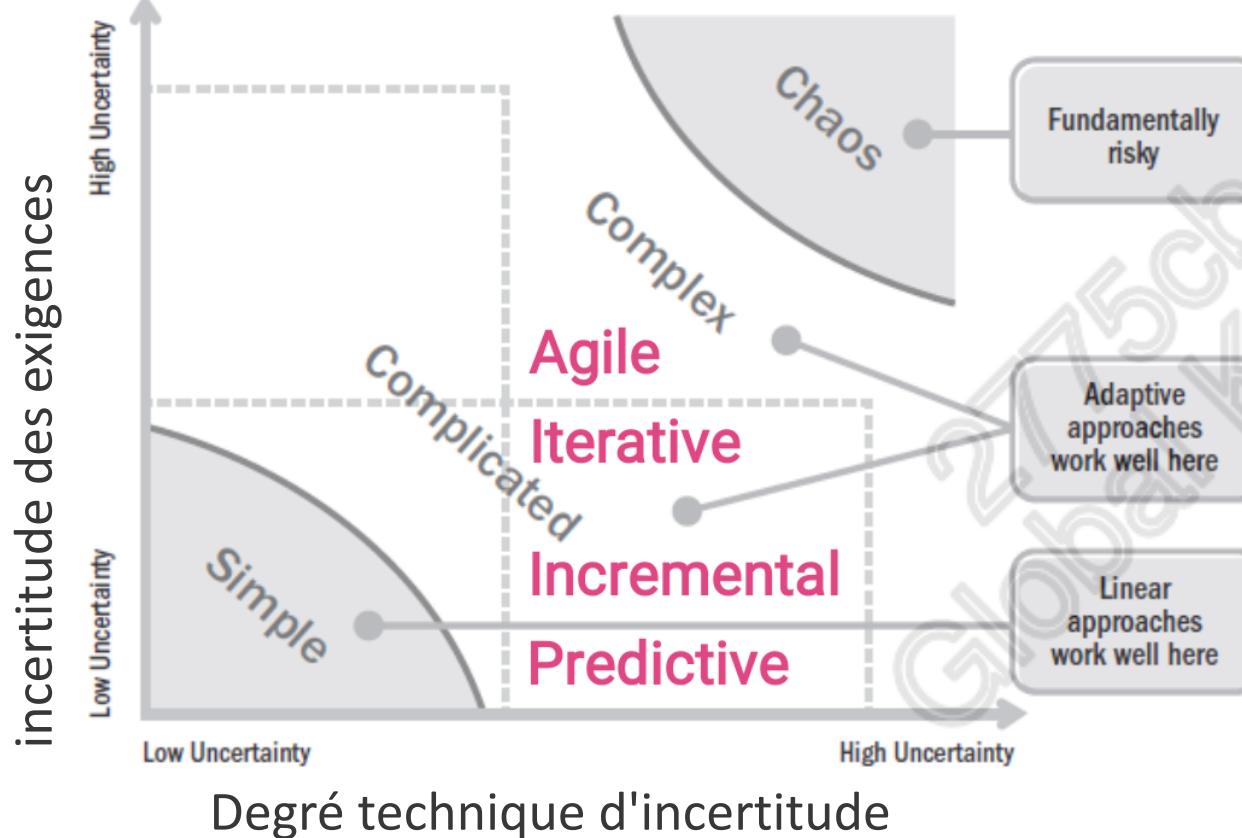
Le choix du cycle de développement du projet consiste à répartir les tâches du projet entre des tranches de temps, ou phases, séparées par des jalons.

Les démarches nécessaires pour conduire le projet peuvent aujourd'hui se classer en deux grandes familles :

- les démarches linéaires, les plus traditionnelles,
- les démarches itératives, basées sur des techniques de maquettage et/ou de prototypage.

On admet à présent que le meilleur des cycles de développement est celui qui est le mieux adapté aux caractéristiques majeures du projet

# CHOIX D'UN CYCLE DE DEVELOPPEMENT



Construction d'un pont en Agile



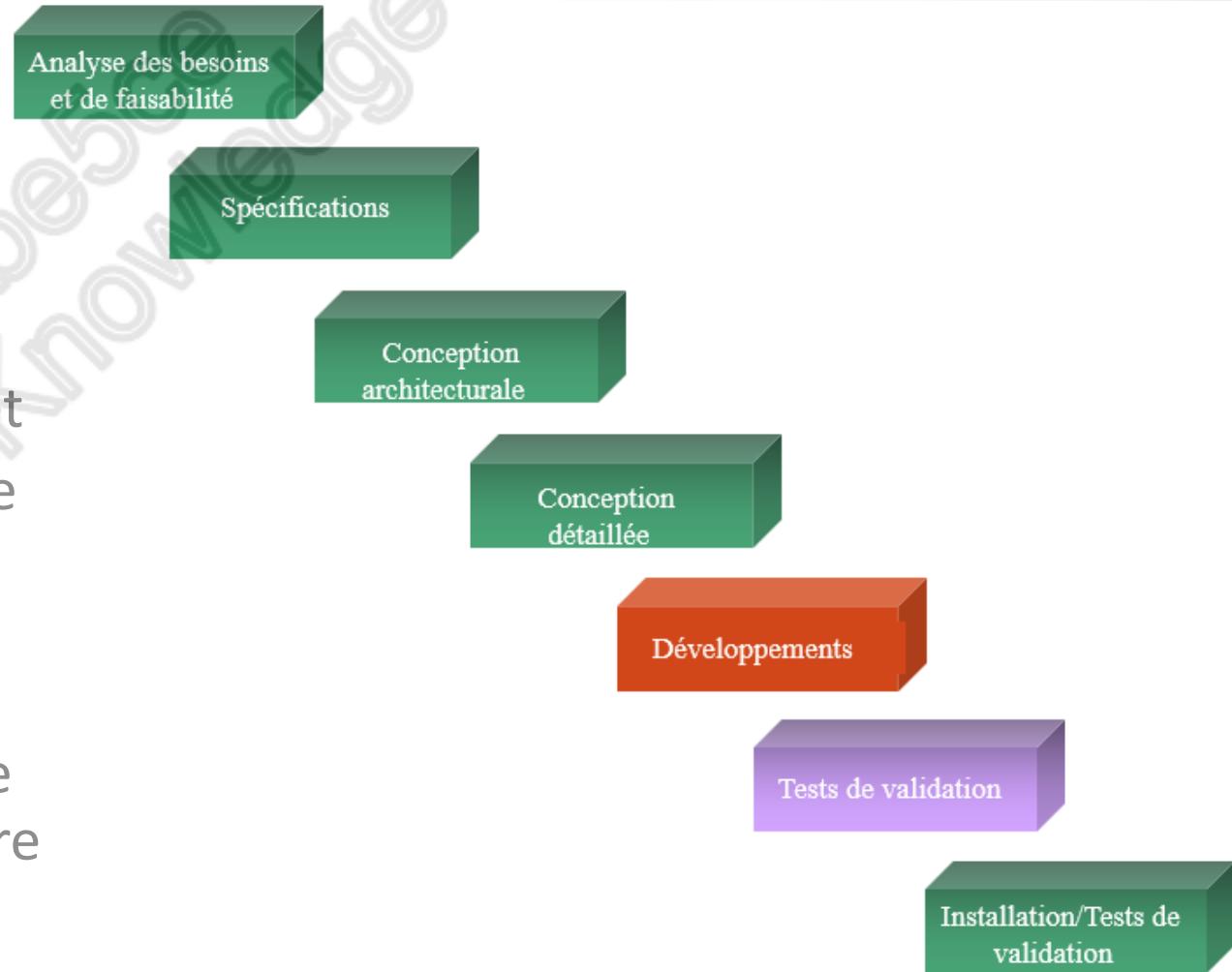
En Agile : je veux me déplacer avec 1 système de roues



# CYCLES DE DEVELOPPEMENT : Cycle linéaire

## Cycle en cascade :

- Respect des niveaux d'abstraction  
Maîtrise des processus décisionnels
- A chaque fin de phase, présentation d'un résultat, à partir duquel les décideurs peuvent choisir la poursuite ou non du projet
- Point de vue de l'organisation : paraît simple  
Mais cela se paye par une **rigidité** jugée souvent excessive et une grande **difficulté d'adaptation**
- Pour un projet important, les phases d'étude peuvent mobiliser de gros moyens, sans autre retombée que des dossiers

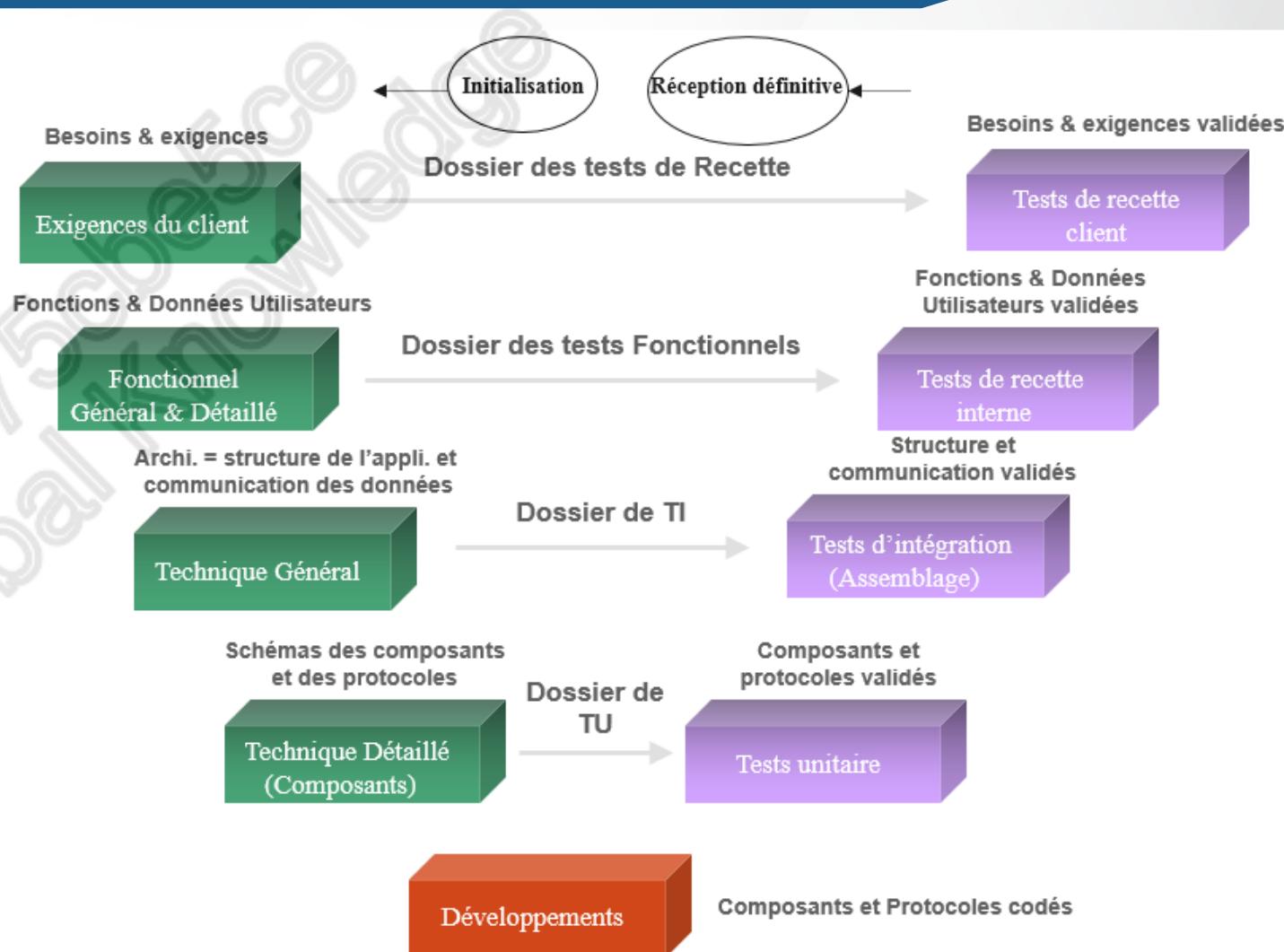


# CYCLES DE DEVELOPPEMENT : Cycle linéaire

## Cycle linéaire en V

- Une variante du cycle en cascade est le cycle en V, qui met en relation chaque phase du développement avec une des phases de test

Ces 2 cycles, définis dans des démarches linéaires, sont très découpés et rendent les retours arrière très difficiles. On peut rencontrer des difficultés d'intégrer des nouveaux besoins en conception



# CYCLES DE DEVELOPPEMENT : Cycles itératifs

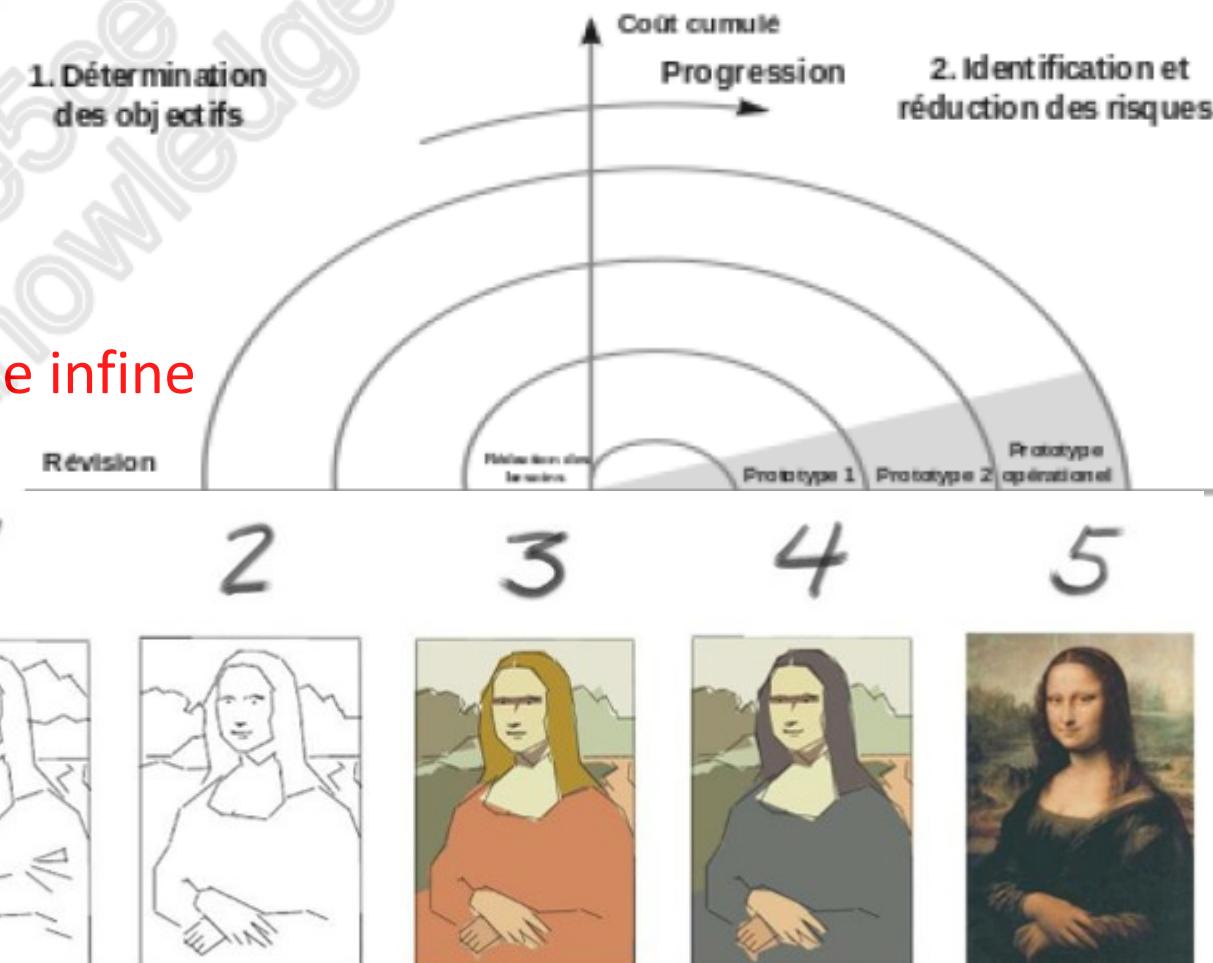
## Présentation :

- Les démarches itératives apparaissent progressivement pour pallier les déficiences des démarches linéaires.
- Le cycle de vie itératif présente trois caractéristiques principales :
  - il affiche le caractère incrémental et non linéaire du développement,
  - il met l'accent sur **l'interactivité entre émetteur du besoin et équipe de développement**, la spirale ramenant face à face les interlocuteurs à chaque passage de spire,
  - il instaure des ruptures entre les différents états de la réalisation logicielle.

# CYCLES DE DEVELOPPEMENT : Cycles itératifs : Cycle en spirale

Chaque tour de spire = l'ensemble des phases pour développer un état intermédiaire ou l'état fini du logiciel :

- analyse du besoin, en particulier détermination du niveau de finition du logiciel à obtenir,
- conception du besoin,
- réalisation du besoin,
- test du besoin,
- intégration et mep (si produit fini),
- validation du besoin,
- bilan du besoin, pour actions correctives pris compte pour le développement du besoin suivant,
- corrections éventuelles,
- planification du développement du besoin suivant

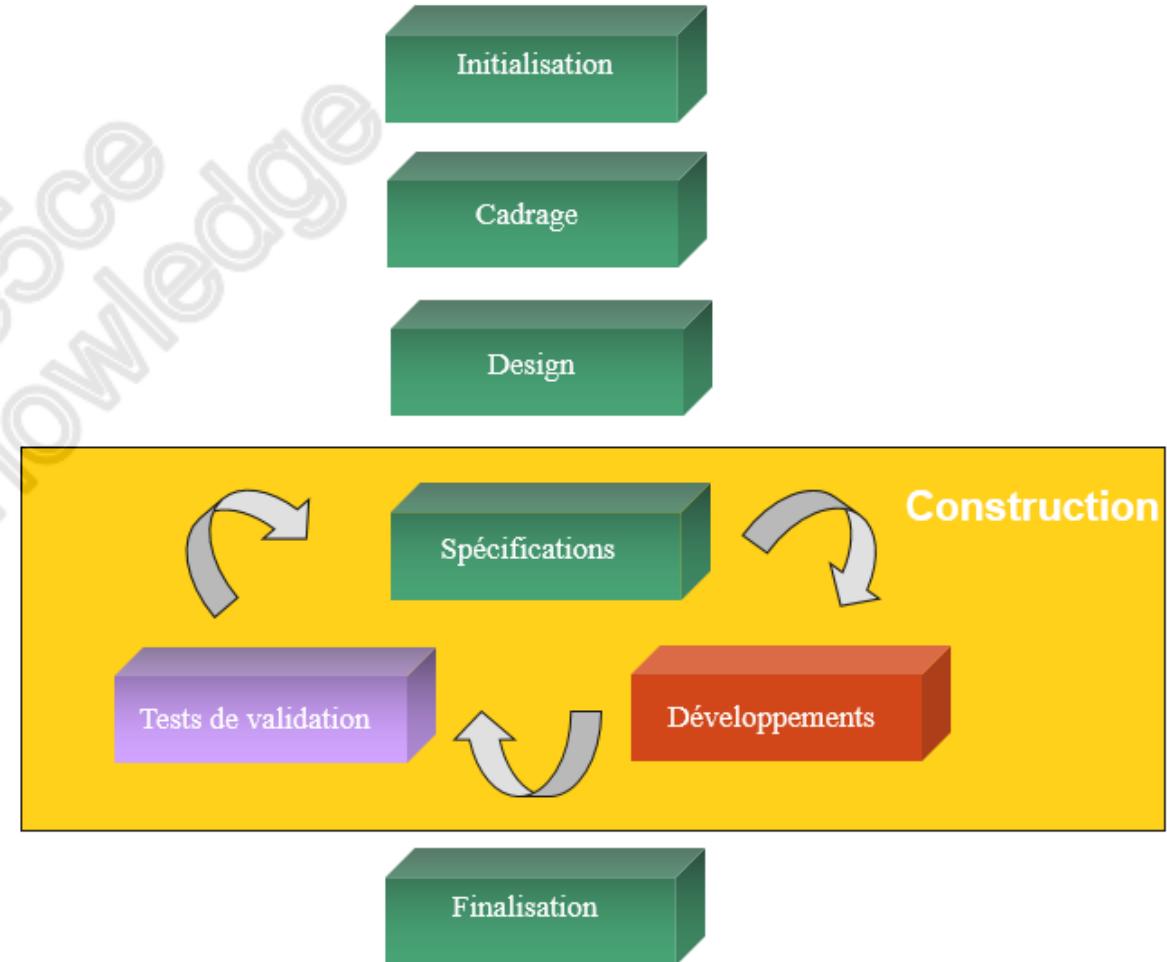


# CYCLES DE DEVELOPPEMENT : Cycle semi itératif

Le cycle semi-itératif pallie le risque de spirale infinie en intégrant une phase de définition de besoin linéaire sur l'ensemble du besoin.

Le cycle semi-itératif est découpé en trois étapes.

- Les deux premières étant classiques et consistent en l'expression des besoins et la conception de la solution.
- C'est l'étape de construction du produit qui est itératif.



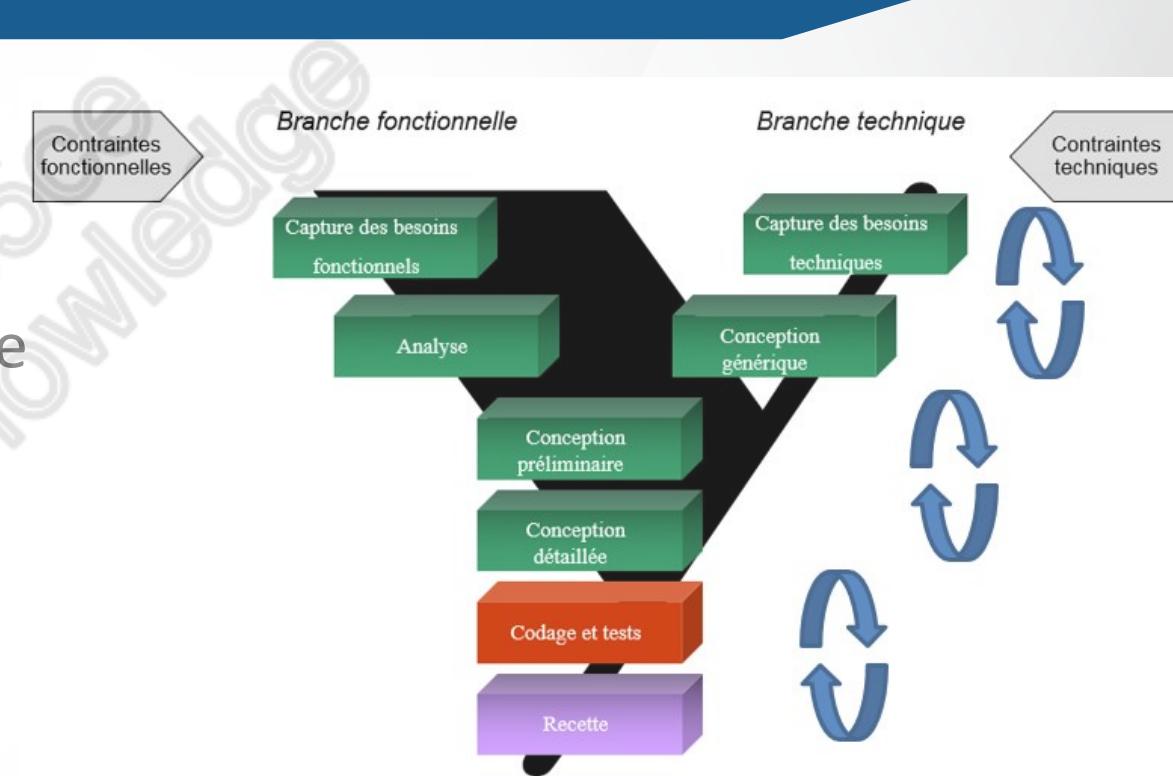
# CYCLES DE DEVELOPPEMENT : Cycle en Y

Dans le cycle en Y l'itération se fera sur l'**architecture, l'analyse, et sur la conception, réalisation.**

La branche de gauche capitalise la connaissance du métier de l'entreprise.

La branche de droite capitalise un savoir-faire technique.

On voit que les plans de tests sont parallélisés avec les phases de conception. Ils devront donc être planifiés très en amont.



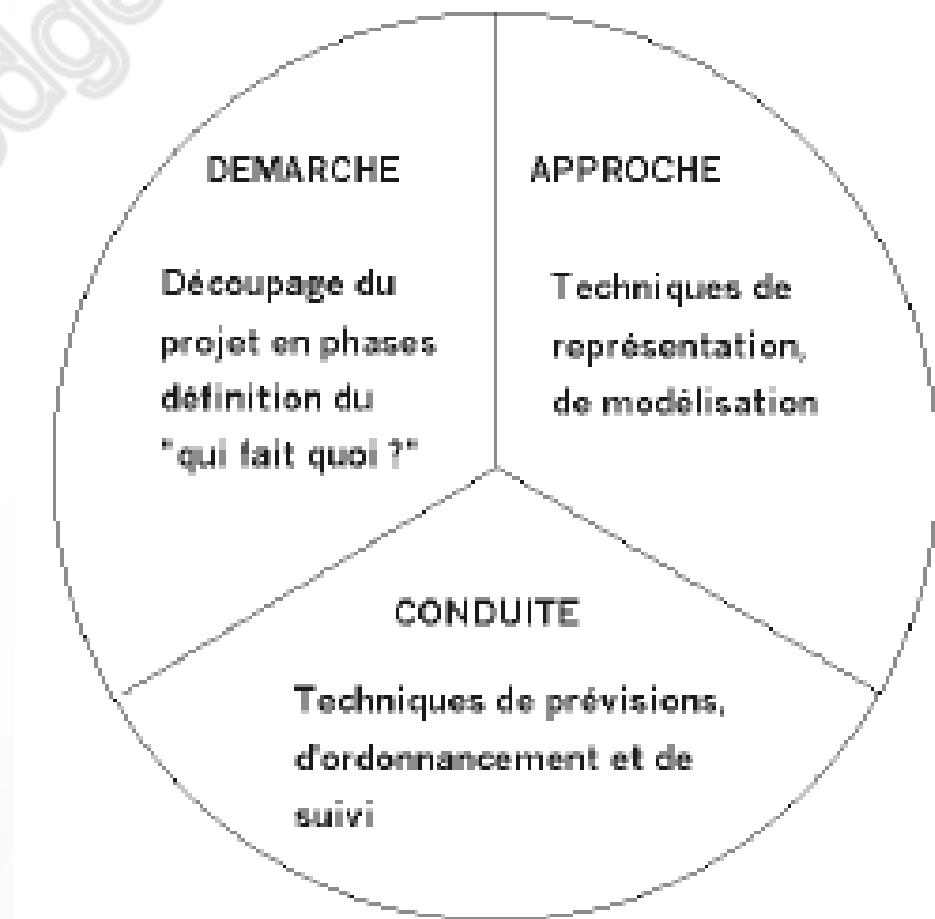
# LES MÉTHODES DE CONCEPTION : Présentation

Rappels : Une méthode de projet permet de répondre à la question :

## **COMMENT FAUT-IL FAIRE LE PROJET ?**

Elle recouvre trois aspects la démarche, l'approche, la conduite.

C'est au travers de l'approche que l'analyse du projet est réalisée.



# LES MÉTHODES DE CONCEPTION : Les différentes générations de méthodes

La première génération de méthodes de conception, les méthodes hiérarchiques, datent des années 60 et décrivent principalement le processus de conception organique. CORIG ou WARNIER sont les représentants les plus connus de cette famille



Les méthodes de conception dites « **d'analyse et de représentation systémique** » apparaissent à la fin des années 70 pour tenter de pallier les nombreuses déficiences des méthodes hiérarchiques. En France, la méthode **Merise** est utilisée de manière quasi unique au cours des années 1980/90.

L'évolution de la complexité des logiciels de gestion, ainsi que les nouvelles possibilités offertes par les outils de développement, ont donné naissance dans les années 90 aux **méthodes orientées objet** telle que la méthode **UML**.



# LES MÉTHODES DE CONCEPTION : Merise

MERISE distingue pour les données et les traitements les niveaux **conceptuel, organisationnel ou logique et physique**

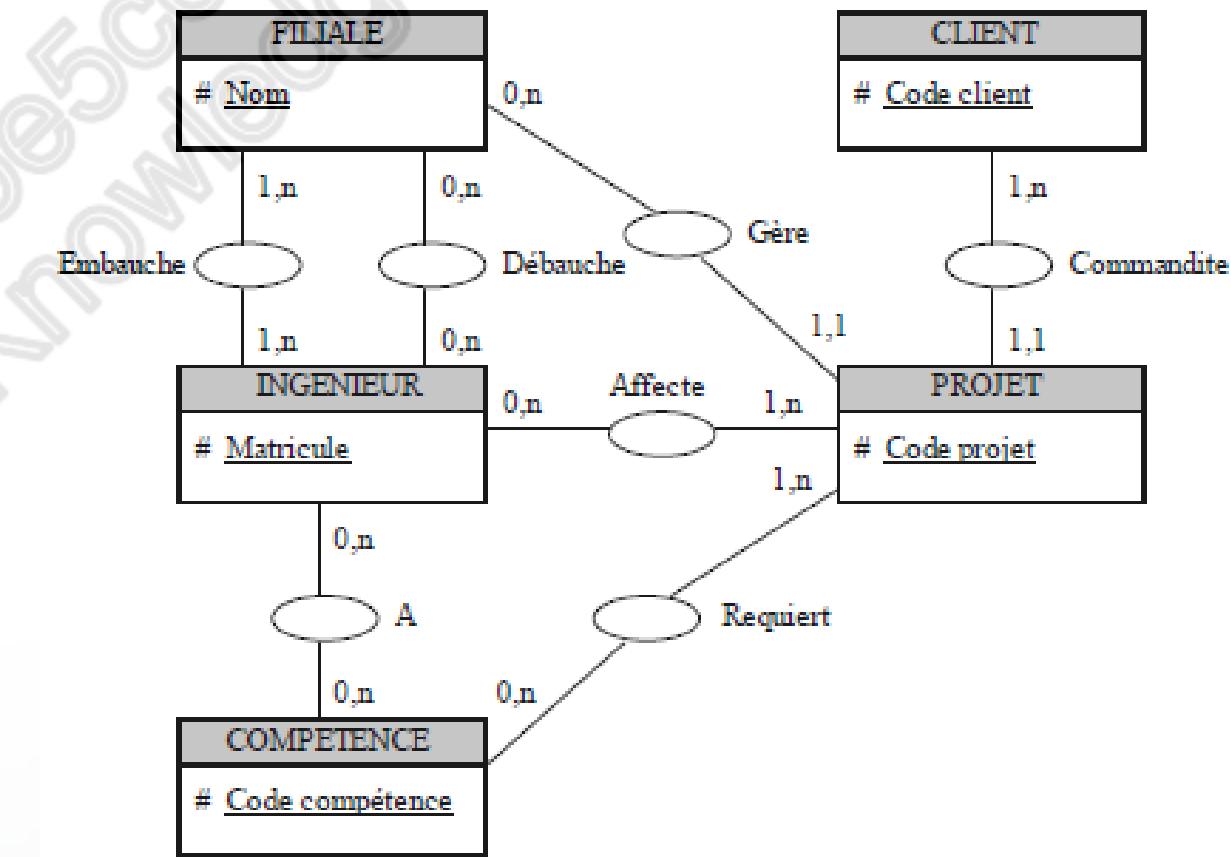
| NIVEAU D'ABSTRACTION | DONNEES                             | TRAITEMENTS                                  | CHOIX PRIS EN COMPTE                   |
|----------------------|-------------------------------------|--|--|
| CONCEPTUEL           | Modèle conceptuel des données (MCD) | Modèle conceptuel des traitements (MCT)      | Choix de gestion<br>QUOI               |
| ORGANISATIONNEL      | Modèle logique des données (MLD)    | Modèle organisationnel des traitements (MOT) | Choix d'organisation<br>QUI, OU, QUAND |
| PHYSIQUE             | Modèle physique des données (MPD)   | Modèle physique des traitements (MPT)        | Choix techniques<br>COMMENT            |

Séparation des Données et des Traitements → Non objet

# LES MÉTHODES DE CONCEPTION : Merise

Le modèle conceptuel des données est une représentation schématique des données et des rapports instaurés entre elles. Il est l'aspect statique de la représentation du système opérant dans son environnement

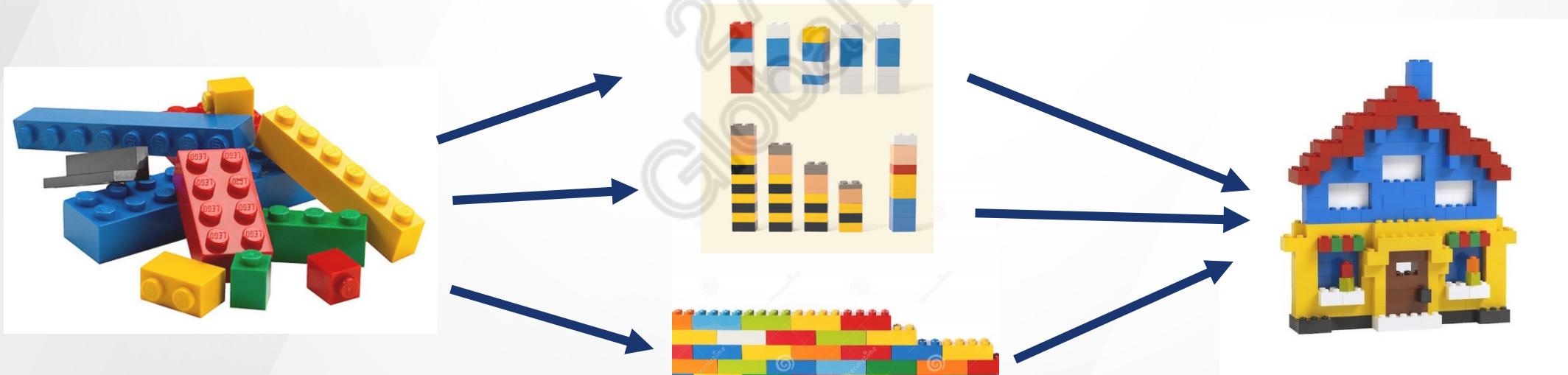
Exemple : MCD simplifié d'une SSII



# LES MÉTHODES DE CONCEPTION : UML

L'évolution des méthodes systémiques vers les méthodes objets répond à 3 buts :

- Mieux **appréhender** la conception et donc la réalisation des **logiciels complexes**, en utilisant conjointement l'abstraction et la modularisation en phase de conception
- Forcer la **réutilisation** des composants logiciels, cette réutilisation → acquisition de briques logicielles, auprès d'éditeurs spécialisés, appelées **composants logiciels réutilisables**
- Répondre aux **besoins de stockage** et d'exploitation de ressources multimédias



# LES MÉTHODES DE CONCEPTION : UML

Objet

Données = Attributs

Traitements=Méthodes

Pour atteindre ces objectifs, l'approche objet propose :

- De redéfinir la notion d'entité, rebaptisée **classe**, de manière à modéliser, conjointement aux attributs (i.e. les propriétés Merise), le comportement de l'entité, c'est-à-dire ces aspects dynamiques et réactifs, appelés méthodes,
- De mener la modélisation **des données et des traitements**, (modèle objet)
- De créer de **nouveaux types de liens** entre classes (lien d'héritage, lien de composition)
- D'encourager l'utilisation de **modèles complémentaires**, tels que des modèles dynamiques ou des modèles fonctionnels

## Exemple d'un objet Date

| Date                  |
|-----------------------|
| jour = 06             |
| mois = 02             |
| annee = 18            |
| siecle = 20           |
| recupereLibelleMois() |
| estceBissextile()     |
| ajoutJour()           |

# LES MÉTHODES DE CONCEPTION : UML

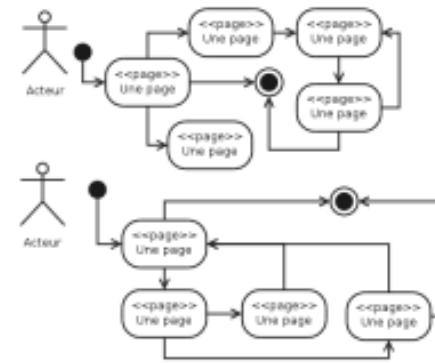
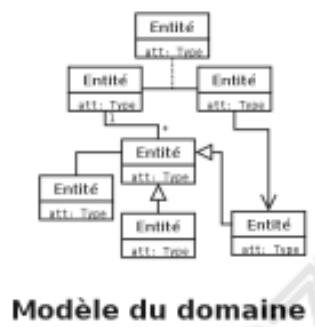
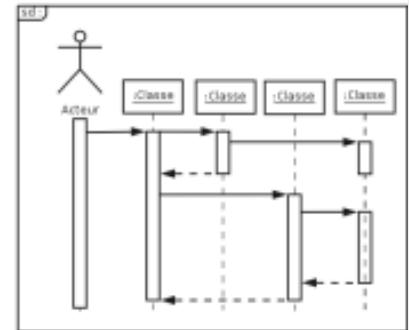
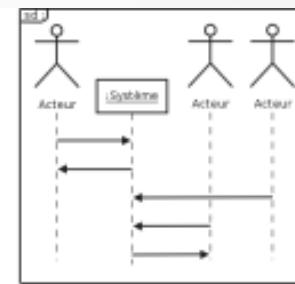
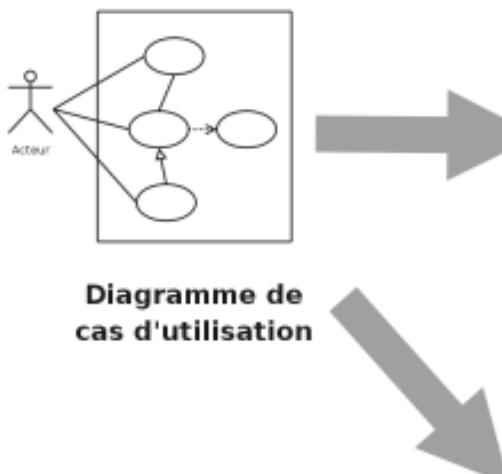
UML (Unified Modeling Language) met à disposition un certain nombre de diagrammes

| Diagramme   | But   | Concepts centraux  |
|---|---|--|
| Le diagramme des cas d'utilisation<br><i>(use-case diagram)</i> | Montrer les interactions des acteurs externes avec le système.<br>Recenser les fonctionnalités. | Cas d'utilisation, acteurs   |
| Les diagrammes d'interaction                                    | Le diagramme de séquence<br><i>(sequence diagram)</i>   | Illustrer la dynamique du système en montrant les interactions entre plusieurs objets.<br>Objets, interactions (messages), flots de contrôle |
|   | Le diagramme de collaboration<br><i>(collaboration diagram)</i>                                 | Même but que le diagramme de séquence.<br>Objets, interactions (messages)  |
| Le diagramme de classes<br><i>(class diagram)</i>               | Montrer la structure du système en termes de classes.   | Classes, relations (association, héritage)   |
| Le diagramme des objets<br><i>(object diagram)</i>              | Illustrer une portion du diagramme des classes pour clarifier un point particulier.             | Objets, relations (liens)  |
| Le diagramme d'états-transitions<br><i>(state diagram)</i>      | Exprimer le cycle de vie des instances de la classe.  | Etats, transitions   |
| Le diagramme d'activité<br><i>(activity diagram)</i>            | Décrire l'activité interne d'un objet.<br>(Technique d'algorithme).                             | Activités, flots de contrôle (décisions, conditions)   |
| Les diagrammes d'Architecture                                   | Le diagramme des composants<br><i>(component diagram)</i>                                       | Spécifier l'architecture logicielle (représenter les composants et leurs relations).<br>Composants logiciels, systèmes, sous-systèmes        |

Besoins



Maquette de l'IHM



# LE PROCESSUS UNIFIE et les méthodes AGILES

Le **cycle de vie du logiciel** n'étant pas pris en charge par UML, pour la raison suivante :  
*Comment prendre en compte la diversité des projets, des problématiques, des équipes et des cultures d'entreprise dans une seule et unique méthode ?*

Le processus uniifié PU répond à cette question (UP en anglais pour Unified Process), c'est un **processus de développement** logiciel construit sur **UML**, à base de composants.

L'objectif est de **maitriser la complexité des projets informatiques**, en **diminuant les risques**.

Il reste **générique** pour préserver une nécessaire adaptabilité au contexte d'entreprise, il donne **1 cadre au développement** logiciel

Il existe plusieurs implémentations du PU, les **méthodes Agiles** en sont une.

# LE PROCESSUS UNIFIE et les méthodes AGILES

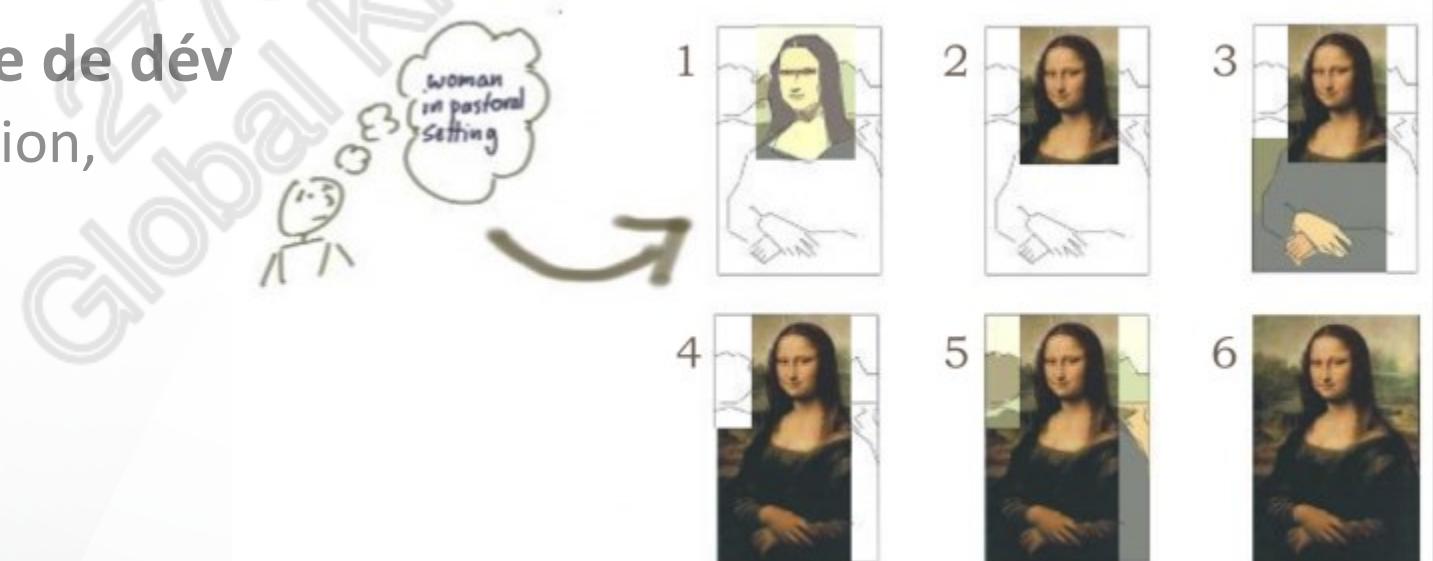
Implémentations de PU : RUP, XUP (Extreme UP), AUP (Agile UP), EUP (Enterprise UP), 2TUP (Two Tracks UP), EssUP (Essential UP)

PU = 1 méthodologie (Qui-Quoi-Quand) + 1 méthode (Comment)

PU privilégie l'interaction et les individus

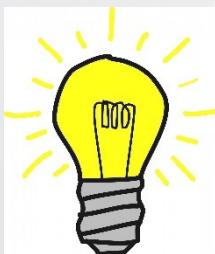
Ce processus utilise le cycle de dév

- Piloté par les cas d'utilisation,
- Centré sur l'architecture,
- Itératif et incrémental.



# LES METHODES AGILES

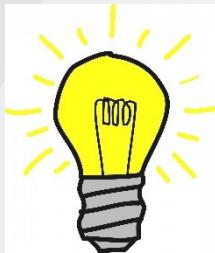
- Les méthodes dites agiles **décrivent les processus** de développement d'application. Ce sont des **bonnes pratiques de modélisation** que l'on applique à des méthodes déjà existantes.



**Accepter le changement** : Ne pas être rigide lors des dév, Rendre le logiciel malléable, Pas chercher à le concevoir dès le début mais à le faire évoluer au fur et à mesure

## AGILE

**Le logiciel doit fonctionner**, même au détriment de la documentation



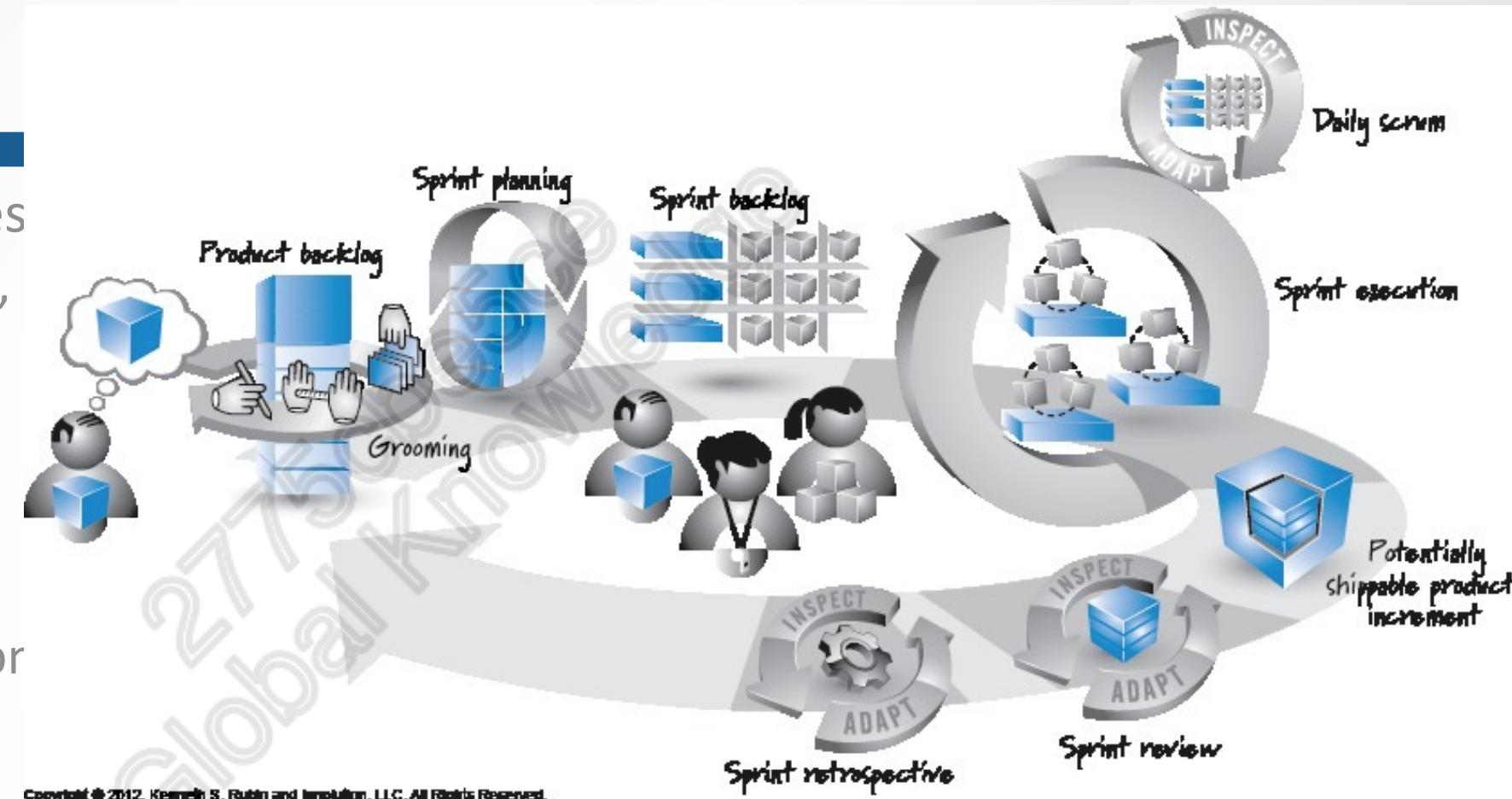
L'accent est mis sur la **communication**. Les méthodes agiles ont ainsi des avantages côté conduite de projet :

- Côté **utilisateur**, il est toujours mis au courant dans un laps de temps court ∈ équipe
- Côté **équipe de développement**, elle est considérée comme **plus importante** que les moyens matériels

# DE L'UP VERS LE RUP ET LES METHODES AGILES

## Principales méthodes agiles

- Extreme Programming XP,
- Processus Urbanisant les Méthodes Agiles PUMA,
- Adaptive Software Development ASD,
- Crystal clear,
- Dynamic Systems Development Method DSDM,
- Scrum



➤ Des outils existent pour gérer les projets, nommés AGL ils s'interfacent avec les IDE, ils ciblent soit certaines soit toutes les fonctionnalités du projet : Cycle de vie, Administration, Techniques de Modélisation, de Réalisation, Capitalisation

# QUELQUES CHIFFRES CONCERNANT LES PROJETS

En 2009, le cabinet Standish Group publie les chiffres suivants:

- Environ 30% des projets informatique n'aboutissent pas
- Plus de 30% des projets informatique dépassent le délai initial
- Environ 50% des projets informatique dépassent leur budget initial
- Et ils coûtent alors en moyenne 189% de ce budget initial !
- Moins de 40% seulement réussissent dans les délais et les budgets
- Mais ils couvrent alors seulement 42% des fonctionnalités initialement prévues !

Les causes de ces échecs sont souvent dues à 2 facteurs :

- **Le volet humain négligé,**
- **La gouvernance mal adaptée.**



# QUELQUES CHIFFRES CONCERNANT LES PROJETS

## Les sources d'erreur des logiciels

1. Mauvaise définition des exigences

2. Problèmes de communication entre clients et développeurs

3. Déviations délibérées des exigences du logiciel

4. Erreur de conception (logique)

5. Erreurs de programmation

6. Non conformité à la documentation ainsi qu'aux instructions de programmation

7. Insuffisances du processus de tests

8. Erreurs de l'interface usagers ainsi que de la procédure

9. Erreurs de documentation

# LES PROJETS

Nécessité de mettre en place l'assurance qualité pour garantir :

- Une meilleure évaluation de la performance des SI,
- Une gestion des ressources des SI plus efficace,
- Une gestion des risques plus pertinente,
- Une amélioration de la valeur des services de l'entreprise par le biais de ses SI,
- Une meilleure adéquation des SI à la stratégie de l'entreprise.

# REFERENTIELS DE QUALITES :

## Présentation

- Ces référentiels ont été mis en oeuvre pour garantir la reproductibilité d'un processus projet et l'adapter en fonction des évolutions.
- Face à une externalisation, voire délocalisation (nearshore, offshore) de plus en plus fréquente des développements informatiques, les référentiels qualités sont devenus une nécessité. Certains sortent du lot :
  - CMMI Evaluation d'entreprise sur les processus informatiques,
  - ITIL Evaluation de personne sur l'activité Informatique,
  - PMP Evaluation de personne sur le management de projet,
  - Lean Six Sigma Evaluation de personne multi-activités,
  - Prince2 Evaluation de personne sur le management de projet

# REFERENTIELS DE QUALITES :

## Présentation

| Entité évaluée       | Dispositif                                   | Activités concernées                | Secteur d'activité |
|----------------------|--|-------------------------------------|--------------------|
| Entreprise ou entité | EFQM<br>ISO 9001<br>Sarbanes-Oxley<br>SAS 70 | Toutes                              |                    |
| Processus            | ISO 15504                                    |                                     |                    |
| Service              | Certification de services                    |                                     |                    |
| Personne             | Six Sigma                                    |                                     |                    |
| Entreprise ou entité | ISO 27001                                    |                                     |                    |
| Processus            | SSE-CMM                                      |                                     |                    |
| Personne             | CobiT-CISA<br>CobiT-CISM<br>CISSP            | Sécurité des systèmes d'information |                    |
| Produit              | ISO 15408<br>Critères communs ITSEC          |                                     | Tous domaines      |
| Personne             | AFITEP-CDP<br>AFITEP-CGP<br>PMP<br>PRINCE2   | Management de projet                |                    |
| Entreprise ou entité | ISO 20000<br>TickIT<br>eSCM                  |                                     |                    |
| Processus            | CMMI<br>SW-CMM                               | Informatique                        |                    |
| Produit              | ISO 25051                                    |                                     |                    |
| Personne             | ITIL<br>PCIE<br>SWEBOK                       |                                     |                    |
| Entreprise ou entité | HAS  | Toutes                              | Santé              |
|                      | TL 9000<br>Trillium                          |                                     | Télécommunications |

# REFERENTIELS DE QUALITES : ITIL

- **Objectifs d'ITIL**
- L'objectif d'ITIL est de doter les directions des systèmes informatiques (DSI) **d'outils et de documents** leur permettant **d'améliorer la qualité** de leurs prestations.
- Quand on dit améliorer les prestations cela veut dire améliorer la satisfaction de leurs utilisateurs cités en tant que clients, tout en répondant au mieux aux objectifs stratégiques de l'entreprise.
- On doit considérer le service informatique comme un ensemble de processus étroitement liés, au service des clients.

# REFERENTIELS DE QUALITES : ITIL

- Origine d'ITIL
- ITIL (*Information Technology Infrastructure Library*) est un cadre de référence, proposé par l'OGC (Office of Government Commerce) du Royaume-Uni rassemblant, dans un ensemble de guides, **les meilleures pratiques** en matière de management des services informatiques.
- La bibliothèque ITIL a été initiée dès le début des années 80 par le gouvernement britannique afin d'améliorer le service rendu par leurs directions informatiques. ITIL s'est implanté aux USA, via des entreprises comme Accenture, Deloitte, Ernst & Young, Hewlett-Packard, ....

# REFERENTIELS DE QUALITES : ITIL

- **Couverture d'ITIL**
- ITIL couvre plusieurs domaines, permettant d'aborder l'ensemble des problématiques géré par les DSI. Le cœur de la démarche ITIL est couvert par les deux premiers domaines.
  - **Service Support**
  - **Service Delivery**
  - Infrastructure Management
  - Applications Management
  - Service Management
  - Business Perspective
  - Business Requirements
  - Technology

# REFERENTIELS DE QUALITES : ITIL

- **Bénéfices d'ITIL**
- Lorsque l'on utilise ITIL, on constate le plus généralement :
  - Satisfaction des utilisateurs (personnel et clients),
  - Clarification des rôles,
  - Amélioration de la communication entre les services,
  - Mise sous contrôle des processus avec des indicateurs pertinents et mesurables, permettant d'identifier les leviers pour réaliser des économies,
  - Meilleure compétitivité,
  - Sécurité accrue (disponibilité, fiabilité, intégrité),
  - Capitalisation des données de l'entreprise ,
  - Optimisation de l'utilisation des ressources.

# REFERENTIELS DE QUALITES : CMMI

## ➤ Objectifs de CMMI

- L'objectif de CMMI est d'encourager les entreprises qui développent des logiciels, à mettre leurs processus sous contrôle, à les améliorer de façon continue et d'évaluer leur niveau de maturité sur une échelle de cinq niveaux.

## ➤ Origine de CMMI

- Le CMMI, :Capability Maturity Model Integration (Modèle intégré du niveau de maturité), est une extension de la spécification CMM, créée pour le ministère de la Défense américain en 1989 afin de déterminer si un projet interne ou tiers serait terminé dans les temps, selon le budget et les spécifications.
- La version actuelle du modèle CMMI a été réactualisée en 2006, elle n'intègre pas la gestion des ressources humaines.

# REFERENTIELS DE QUALITES : CMMI

- **Couverture de CMMI**
- CMMI propose un référentiel des meilleures pratiques en matière de développement logiciel. Bonnes pratiques liées :
  - à la gestion,
  - au développement,
  - à la maintenance d'applications
  - à la maintenance de systèmes.
- Ces bonnes pratiques sont regroupées en 24 processus, eux-mêmes regroupés en 4 types (Process Management, Project Management, Engineering et Support) et 5 niveaux de maturité.

# REFERENTIELS DE QUALITES : CMMI

## ➤ Niveaux de maturité

- **Initial** : Les facteurs de réussite des projets ne sont pas identifiés, la réussite ne peut donc être répétée.
- **Reproductible** : Les projets sont pilotés individuellement et leurs succès sont *répétables*.
- **Défini** : Les processus de pilotage des projets sont mis en place au niveau de l'organisation par l'intermédiaire de normes, procédures, outils et méthodes.
- **Géré** : La réussite des projets est quantifiée. Les causes d'écart peuvent être analysées.
- **Optimisé** : La démarche d'optimisation est continue.

# SOMMAIRE

---

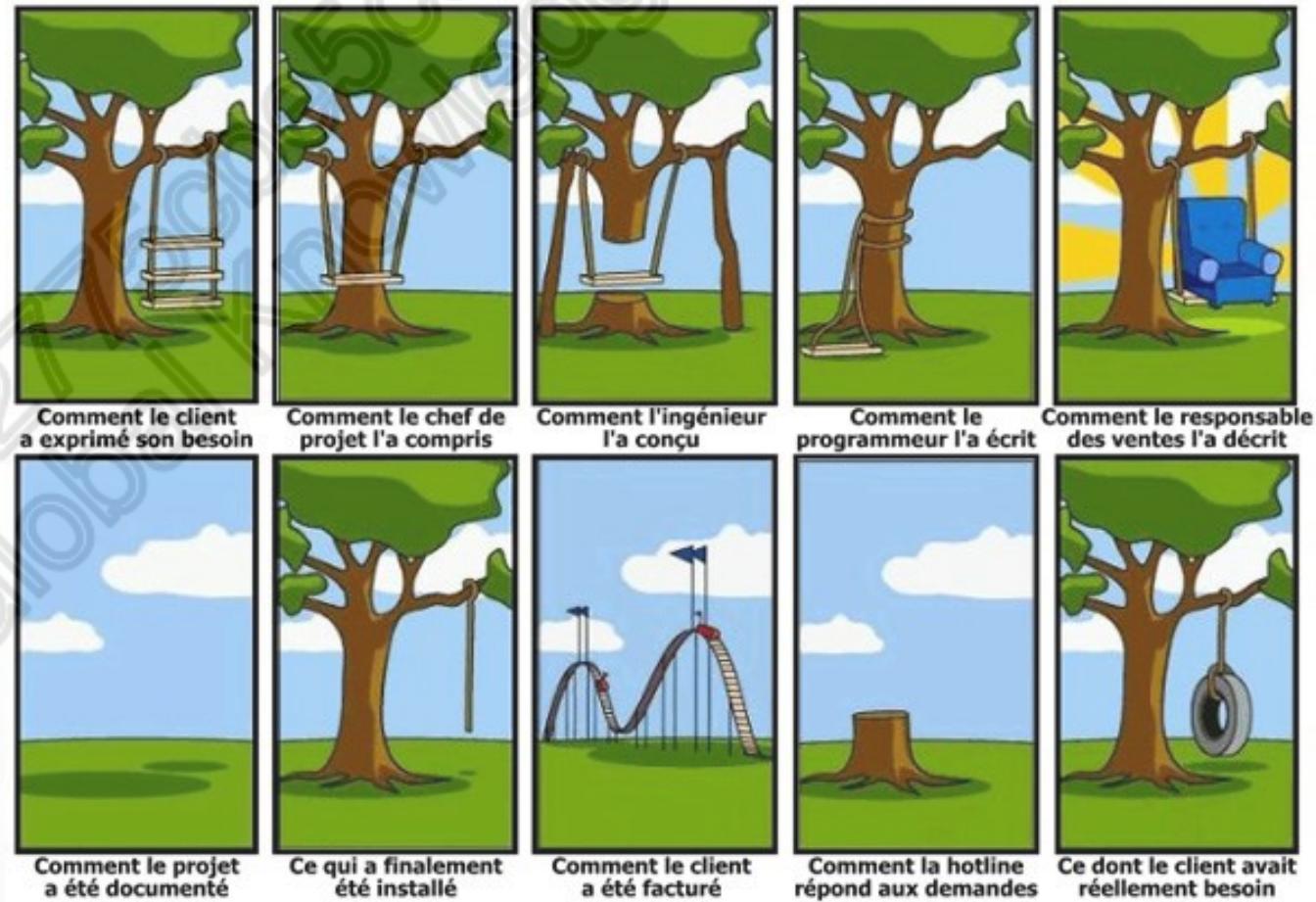
- **Chapitre 1 : INTRODUCTION AU SYSTEME D'INFORMATION**
- **Chapitre 2 : L'INFRASTRUCTURE**
- **Chapitre 3 : LES ARCHITECTURES DISTRIBUEES**
- **Chapitre 4 : LES COMPOSANTS D'UNE APPLICATION**
- **Chapitre 5 : LE PROJET INFORMATIQUE**
- **Chapitre 6 : POUR CONCLURE**

# POUR CONCLURE



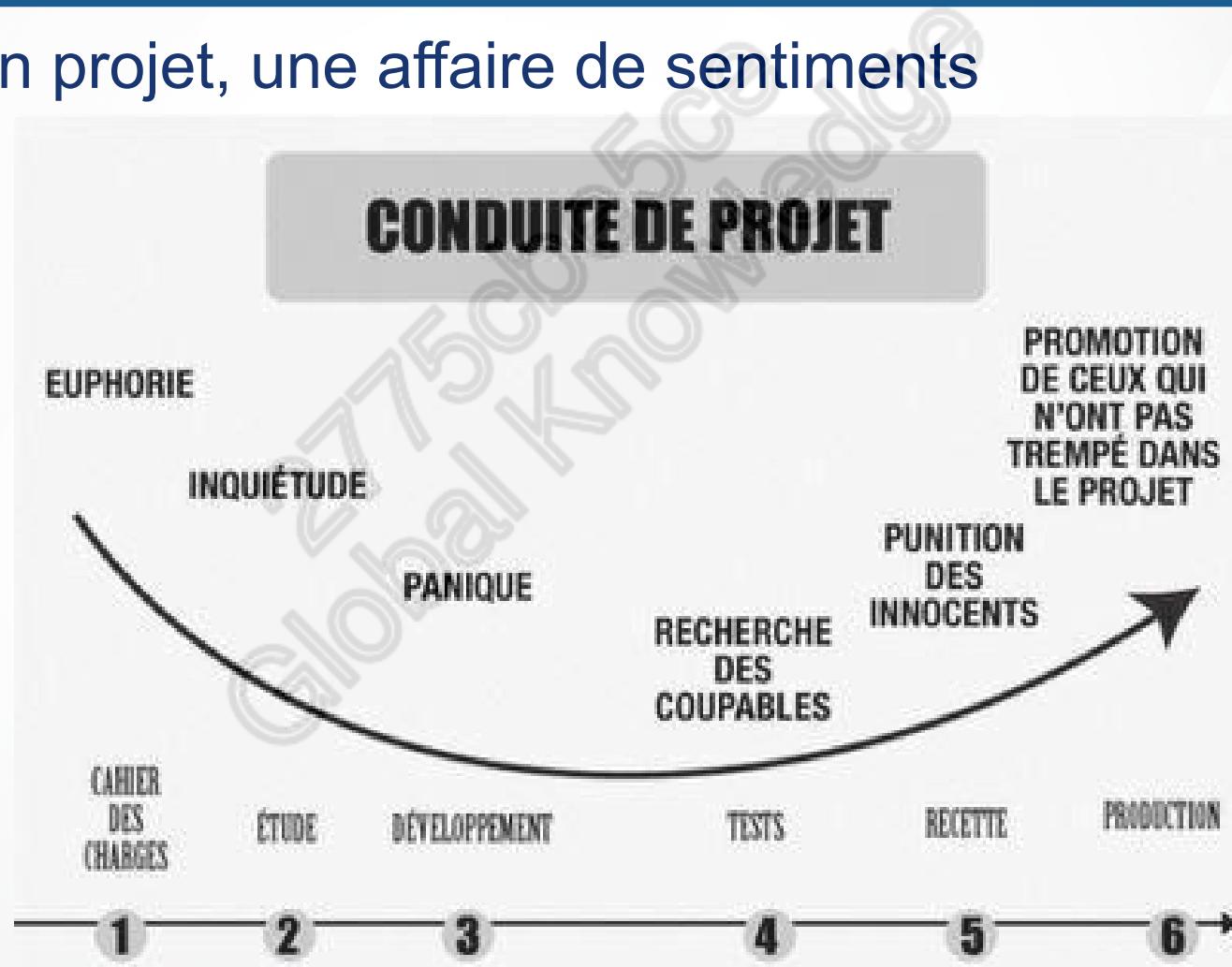
# UN PEU D'HUMOUR

## ➤ Le besoin client est la balançoire



# UN PEU D'HUMOUR

- Les phases d'un projet, une affaire de sentiments



# UN PEU D'HUMOUR

- Une bonne équipe projet, c'est important



# Evaluation de la formation



# Formulaire et billet d'humeur

- Renseigner le formulaire Global Knowledge
- Chacun choisit 3 mots qui décrivent le mieux ses sentiments sur la formation
- A l'aise, absorbé, abattu, ahuri, agacé, allégé, agité, amusé, animé, attentif, apathique, de bonne humeur, bloqué, calme, captivé, centré, charmé, concentré, concerné, confiant, confortable, content de soi, curieux, détaché, déconcerté, détendu, déçu, emballé, embrouillé, enchanté, encouragé, ennuyé, étonné, éveillé, éreinté, étourdi, fier, fatigué, gai, galvanisé, hilare, impatient, impliqué, informé, inquiet, insouciant, indifférent, intéressé, joyeux, libre, nourri, optimiste, paisible, rassuré, ravi, satisfait, sceptique, sensibilisé, soulagé, stimulé, vindicatif, valorisé, zen



Global Knowledge®

Votre partenaire formation

MERCI