

Systeme d'information

Rémy Courdier



Générations et Typologie
d'architecture pour les SI

Stratégie d'évolutions des architectures applicatives pour les SI ?

✓ **Stratégie d'évolution des organisations**

✓ **Enjeux stratégiques des directions informatiques**

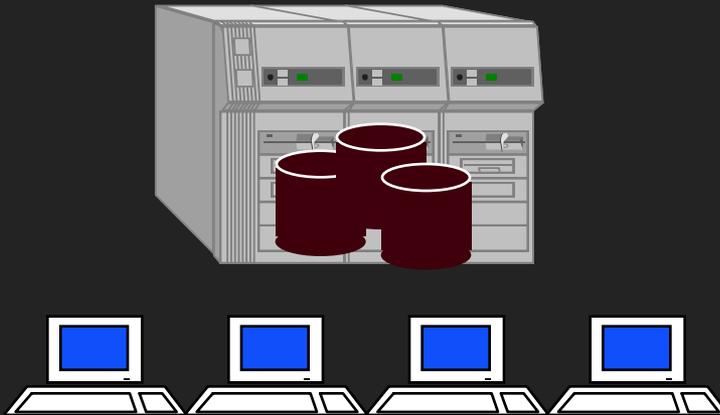
✓ **Objectifs techniques et financiers**

✓ **Caractéristiques des architectures de Systèmes d'Information actuels**

Stratégie d'évolution des organisations

◆ De Produire...

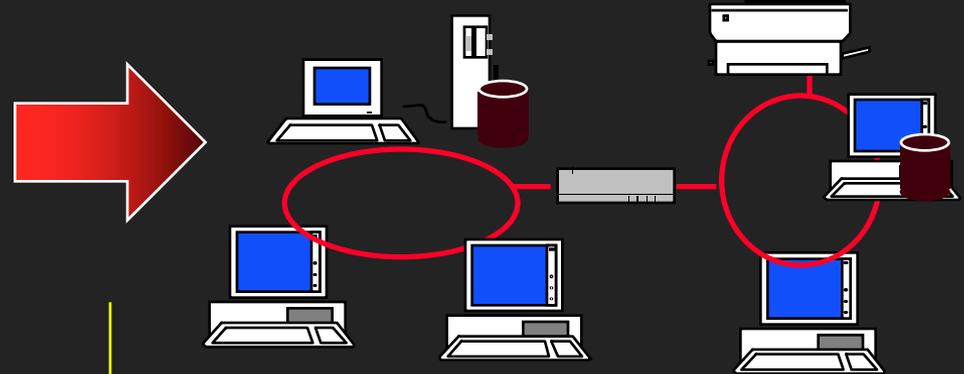
- ✓ *organisation pyramidale*
- ✓ *règne de l'ordinateur central et des directions informatiques*



centralisée, normalisée,
collective, contrôlée

■ A Vendre...

- ✓ *organisation transversale*
- ✓ *rejet du gros système informatique central*



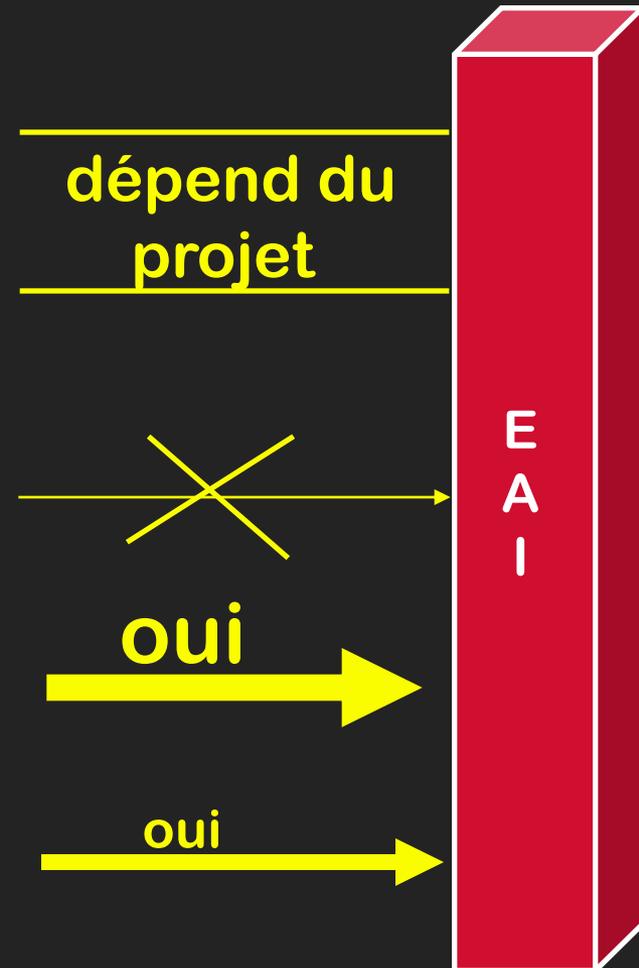
décentralisée, personnalisée
individuelle, maîtrisée

Enjeux stratégiques des directions informatiques

- ◆ Adapter les demandes nouvelles sans modifier l'architecture complète du système
- ◆ Répondre aux besoins des utilisateurs rapidement et progressivement
- ◆ Gérer la cohérence
- ◆ Réutiliser
- ◆ Utiliser des applications s'adaptant aux modifications du marché

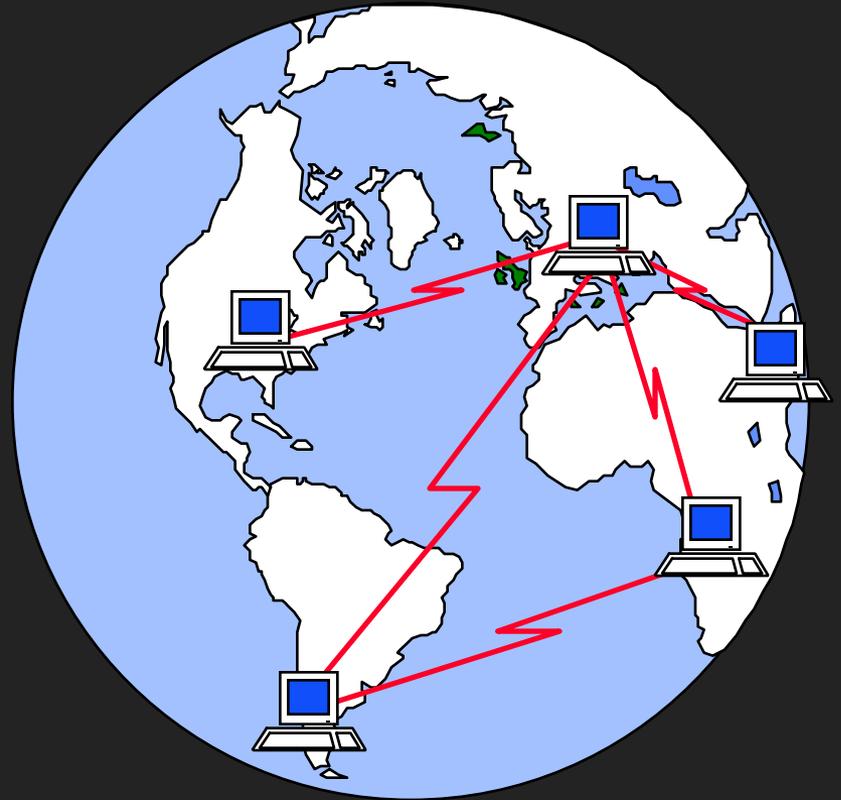
Objectifs techniques et financiers des directions informatiques

- ◆ Réduire les ressources de développement
- ◆ Réduire les dépenses Hard et Soft
- ◆ Améliorer les coûts de maintenance
- ◆ Améliorer l'évolutivité
- ◆ Mieux gérer les ressources processeur et disque



Caractéristiques des Applications « Client-Serveur »

- ◆ Données distribuées
- ◆ traitements répartis
- ◆ approche centrée sur l'utilisateur
- ◆ Environnement matériel et logiciel hétérogène



Les 3 Générations d'architecture support de Systèmes d'Informations

- ✓ **1ere Génération :**
"Le traitement frontal"
- ✓ **2ème Generation :**
"Le traitement Coopératif"
- ✓ **3ème Génération :**
"Le traitement réparti et les données Réparties"
- ✓ **Web: vers une archi. universelle**
- ✓ **Tendances**

Tendances...



- ✓ **L 'Internet MVC**
- ✓ **Les agents mobiles**
- ✓ **Le Grid Computing**

Le traitement frontal

1^{ère} Génération

◆ C/S de présentation

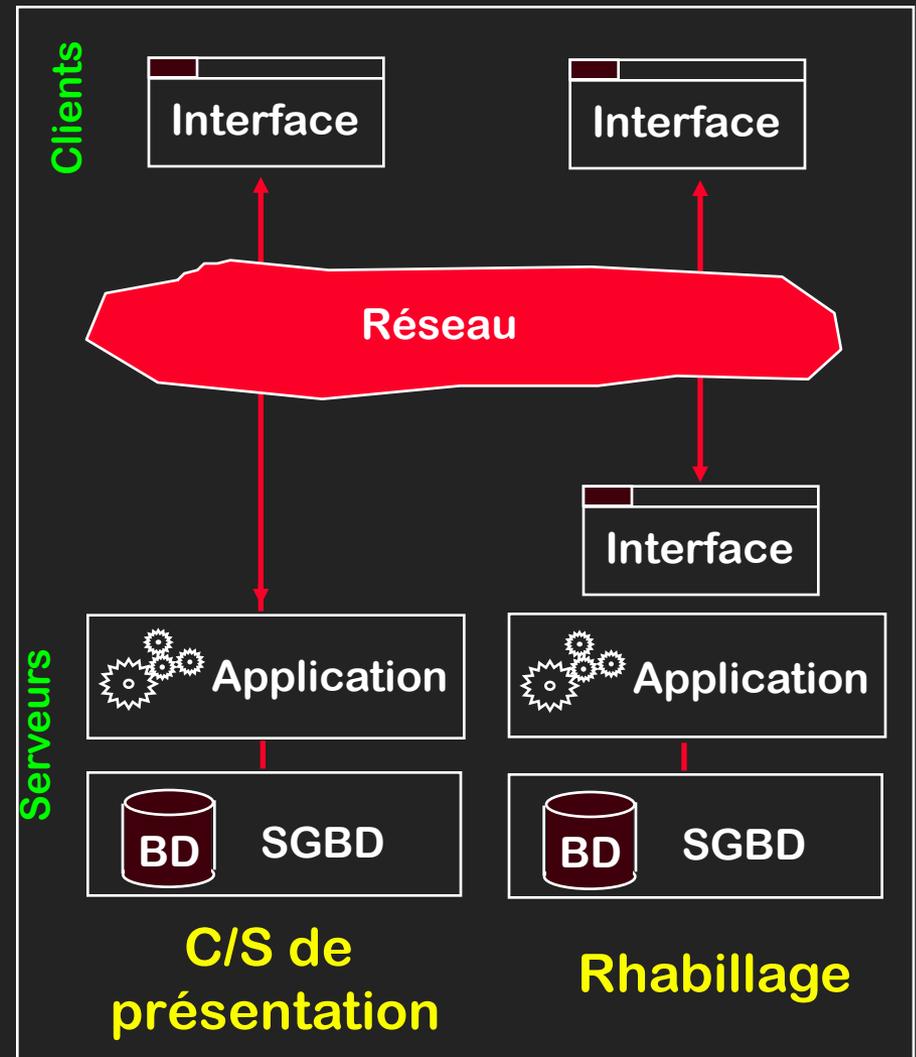
Déport du Dialogue :

- ✓ *La logique de l'interface graphique est déportée sur le poste client*
- ✓ *ex : Terminal X*

◆ Rhabillage (Revamping)

Répartition du Dialogue :

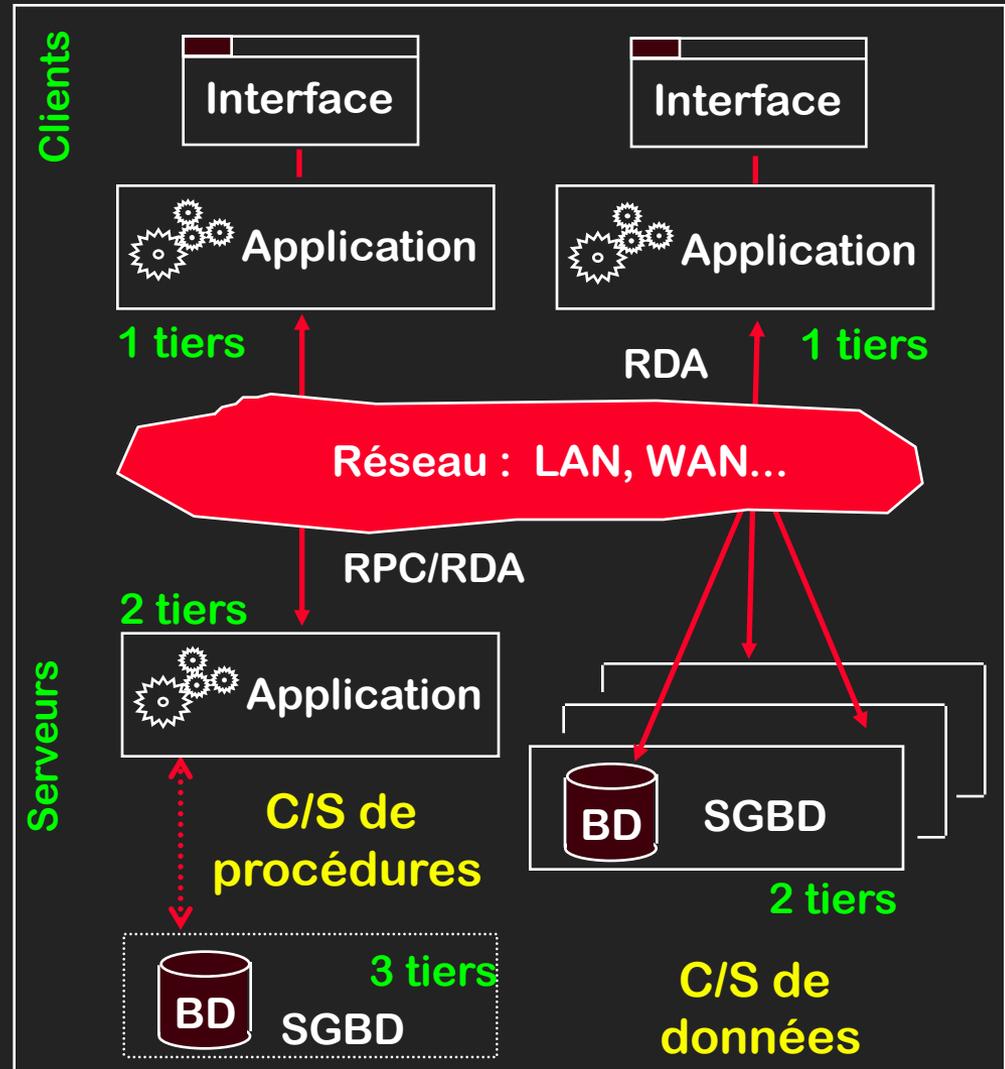
- ✓ *ré-habillage graphique d'applications existantes*
- ✓ *terminaux à "ergonomie limitée"*
- ✓ *ex : MINITEL*



Le traitement Coopératif

2^{ème} Génération

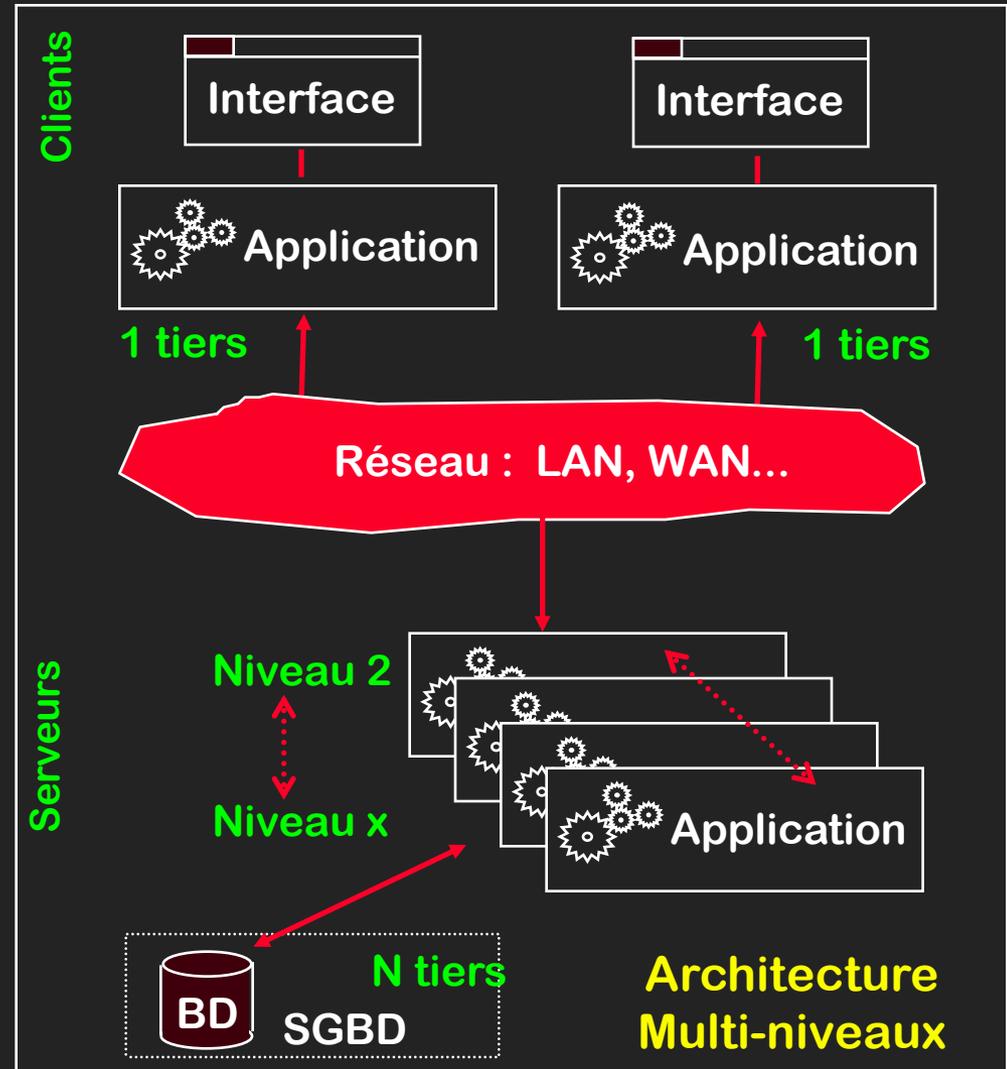
- ◆ C/S de procédures
 - ✓ *L'application Cliente sous-traite l'exécution de proc. à un serveur*
 - ✓ *Répartition partielle de l'application entre client et serveur*
- ◆ C/S de données
 - ✓ *Déport de l'application complète sur le poste client : accès à des données distantes*
 - ✓ *Possibilité d'accès explicite à plusieurs serveurs distants*



Le traitement Coopératif multi-niveaux

évolution de la 2^{ème} Génération

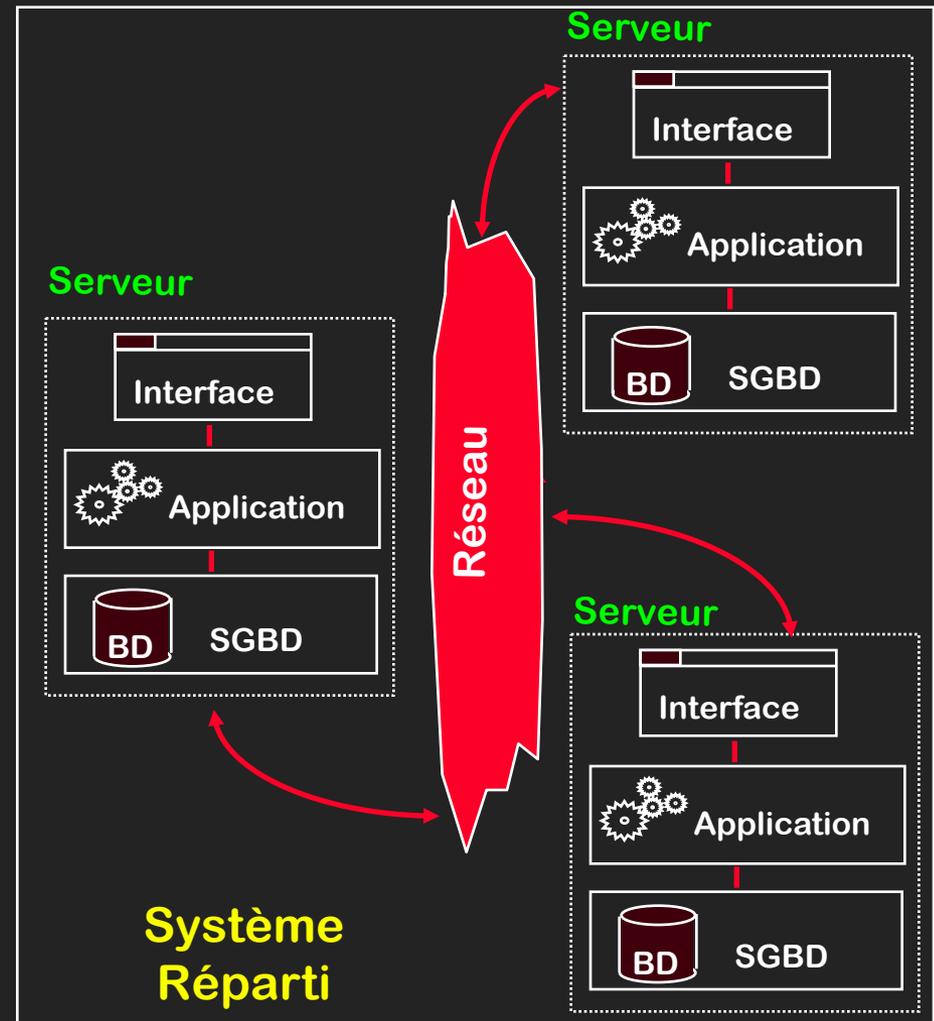
- ◆ Dans l'architecture coopérative chaque serveur effectue un service spécialisé.
- ◆ Un serveur peut donc invoquer un service d'un autre serveur
- ◆ L'architecture 3 tiers et potentiellement une architecture N-tiers à N Niveaux



Le traitement et les données répartis

3^{ème} Génération

- ◆ fin de la dissymétrie client-serveur
 - ✓ *établissement de liens serveurs-serveurs*
 - ✓ *répartition transparente des données sur plusieurs sites*
 - ✓ *Chaque site fonctionne de manière autonome*
- ◆ Limites...
 - ✓ *La mise en œuvre d'architecture performante de ce type n'est pas encore une réalité industrielle*

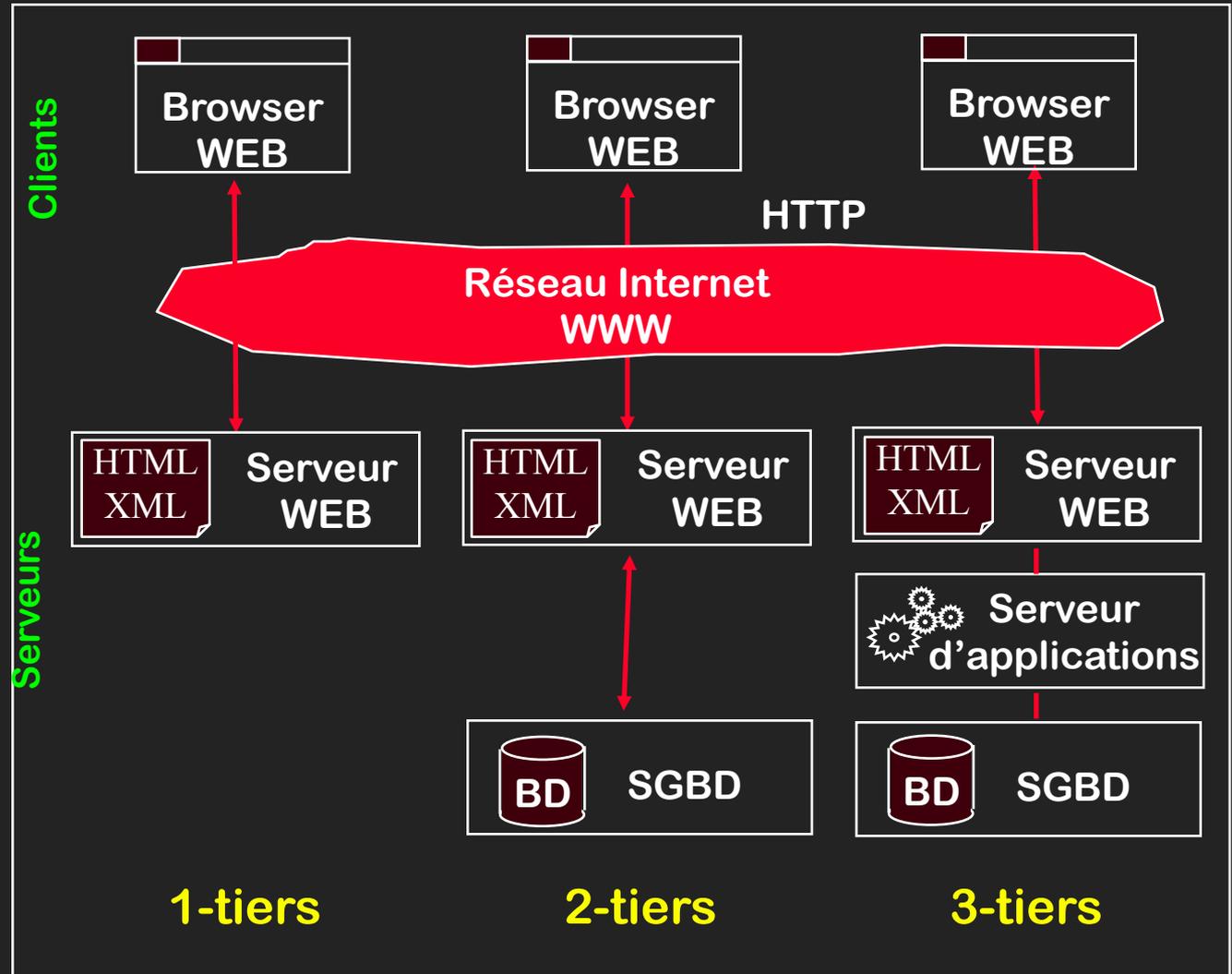


Gérer la complexité de la répartition...

- ◆ Décomposition d'une requête en sous-requêtes mono- ou multi-sites
- ◆ Principe d'optimisation de requêtes multi-sites
- ◆ Existence d'un dictionnaire global :
 - √ *localisation des données*
 - √ *transparence pour l'émetteur de requêtes*
- ◆ Utilisation automatique de traducteurs pour l'accès aux SGBD hétérogènes

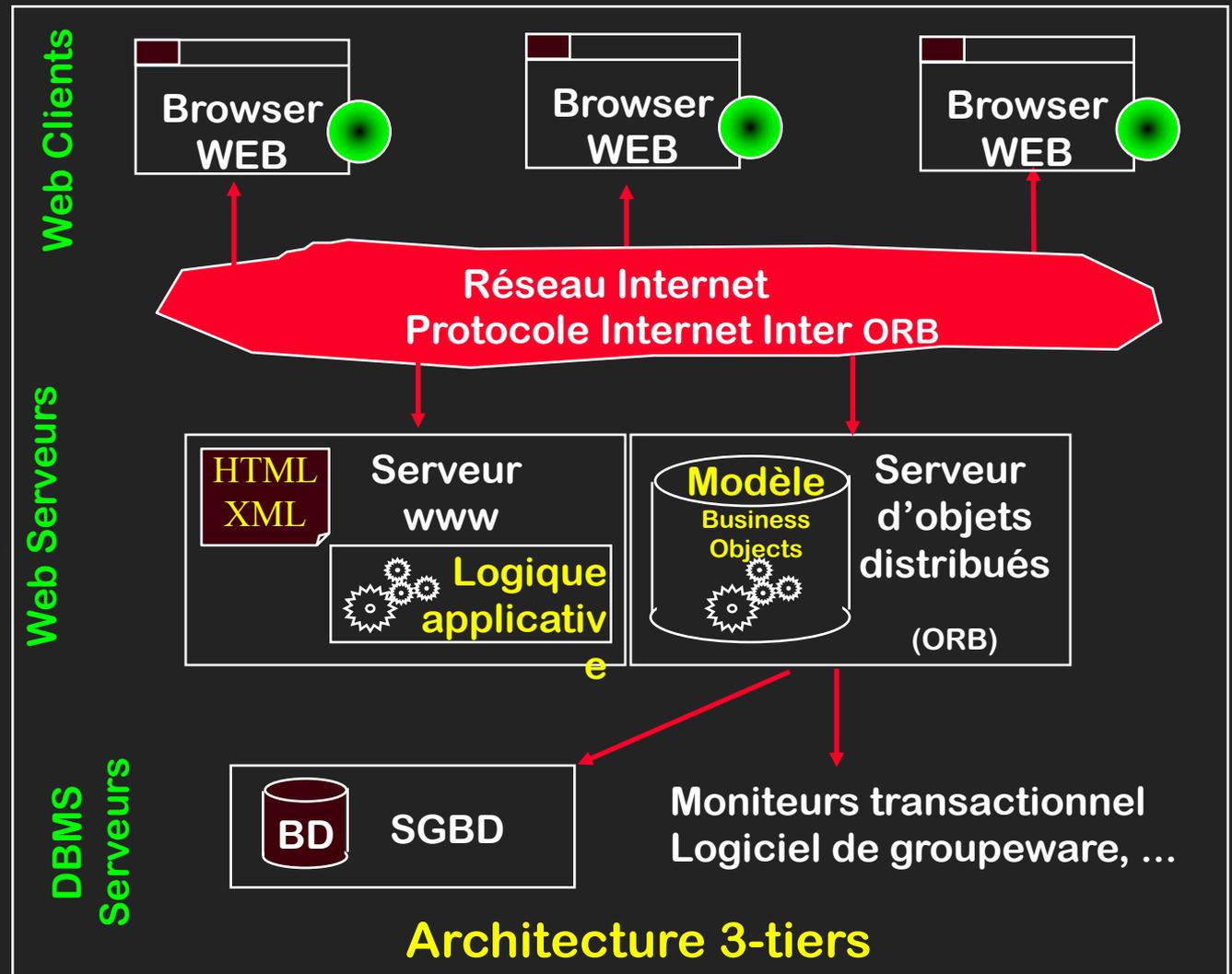
Vers une architecture universel : Intranet/Extranet/Internet

- ◆ *Le retour du C/S de présentation*
- ◆ *Utilisation de standards simples*
- ◆ *Le client fait un travail d'interprétation de fichiers et de gestion de communication avec le serveur*



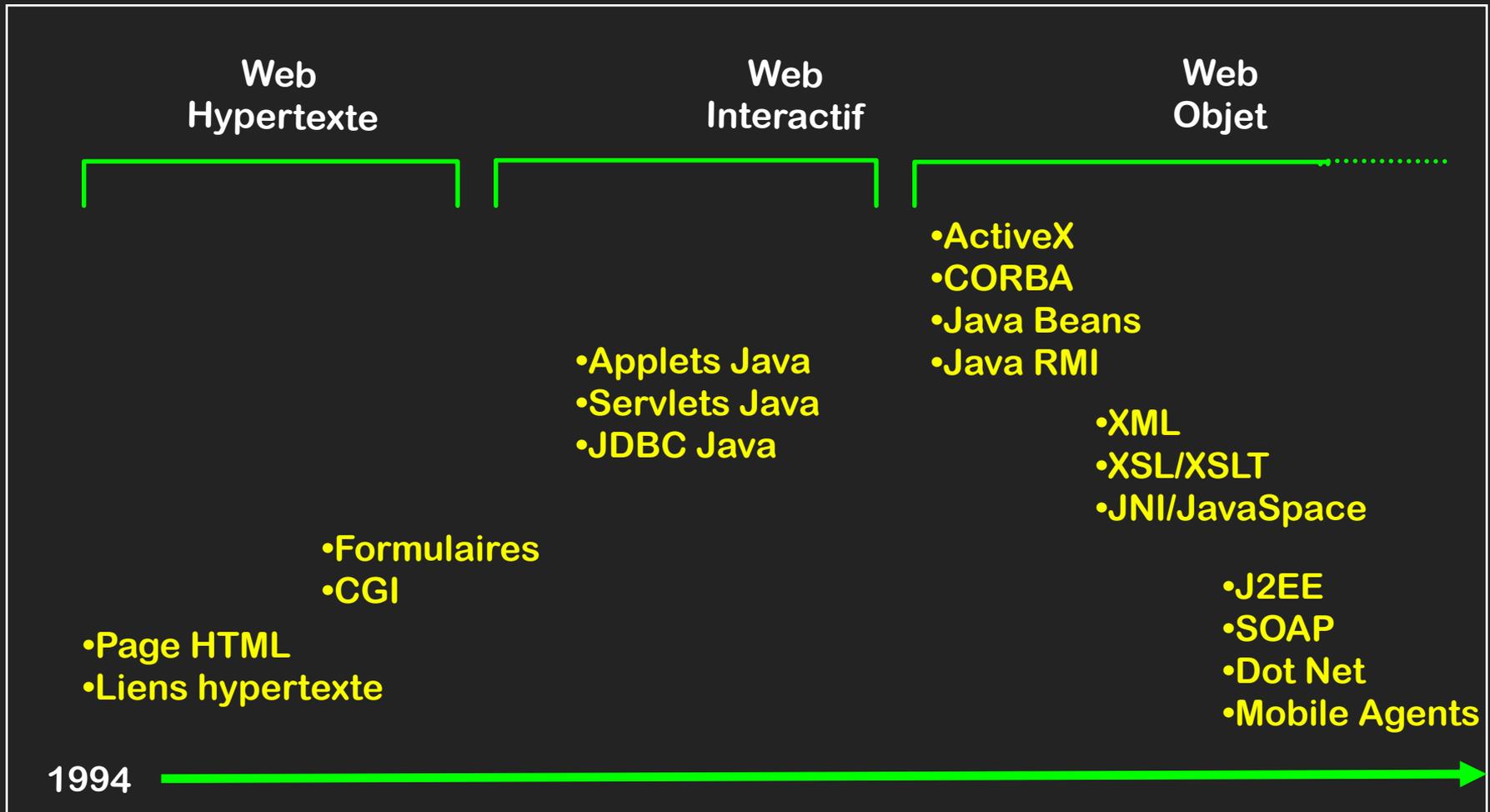
Vers une architecture répartie

- ◆ *Intégration de solutions de type Client/Serveur Universel Web avec des architectures à Objets répartis*



Systemes d'Information tournés vers l'Intranet/Internet

Evolution de la technologie Web

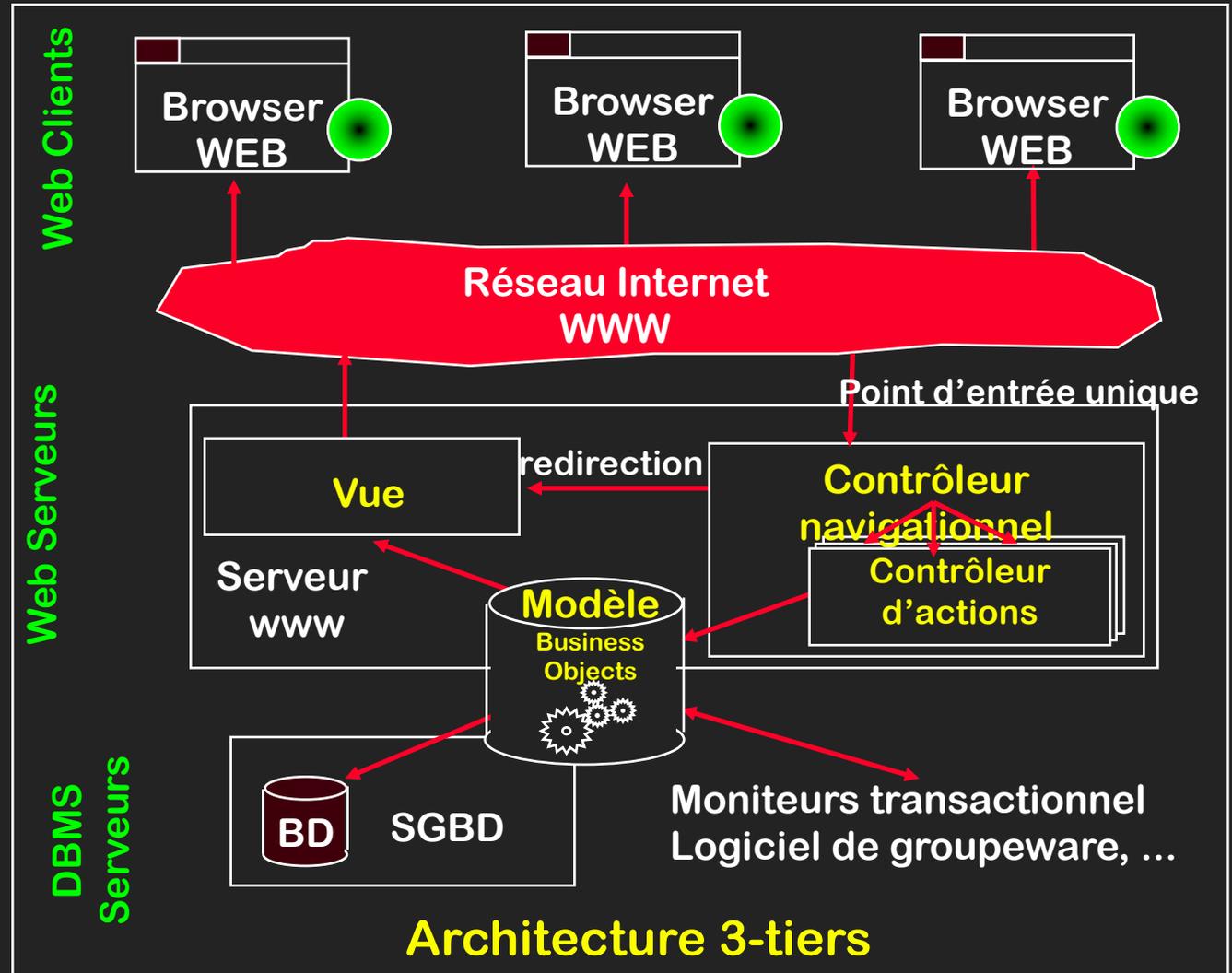


Les tendances...vers une architecture Intranet/Internet Maîtrisée

- ◆ *Mise en œuvre du Modèle MVC dans les architectures de Système d'Info. tournée vers l'Intranet/Internet*
- ◆ *Le modèle MVC : Modèle-Vue-Contrôleur*
 - √ *Modèle/Model*
 - ▶ *Représente le domaine d'application, donnée & logique opératoire*
 - √ *Vue/View*
 - ▶ *Représente l'ensemble des éléments de présentation externes*
 - √ *Contrôleur/Controller*
 - ▶ *Module responsable des règles de navigation et d'enchaînement des actions utilisateurs*
- ◆ *Design Pattern Issu du génie logiciel objet*

Architecture de SI orientée Web & MVC

- ◆ Le contrôleur navigationnel intercepte les requêtes HTTP Client
- ◆ Traduit les requêtes en actions à réaliser
- ◆ Délègue les actions à des Business opérations (web indépendants)
- ◆ Aide à la sélection de la prochaine vue



Tendances... Les agents sur le web

◆ Qu'est ce qu'un agent ?

- ✓ *Si on se réfère à la définition du dictionnaire : du latin : « agens » : celui qui agit. « Un Agent est une personne chargée des affaires et des intérêts d'un individu, d'un groupe ou d'un pays, pour le compte desquels elle agit »*
- ✓ *Un Agent est orientée « tâche », il déploie une activité (ou un service) dans le but de faire quelque chose de précis.*
- ✓ *Un agent logiciel est une entité informatique qui réalise de manière autonome des tâches pour un utilisateur.*

◆ Un agent logiciel est caractérisé par :

- ✓ *un certain degré d'autonomie, d'interactivité, de réactivité.*

◆ Agent mobile

- ✓ *Le processus Agent se déplace d'un serveur à un autre, sur le réseau. Le code de l'objet est transporté et ses données, aussi. Ensuite, il continue son exécution sur la nouvelle machine.*
- ✓ *Exemple : Un Agent de sauvegarde peut se déplacer sur plusieurs serveurs dans le but de faire des restaurations de fichiers.*

Caractéristiques avancées des agents ?

- ◆ **Capacité sociale**
 - ✓ *Les Agents interagissent avec les autres Agents grâce à des langages et protocoles de communication entre Agents (et éventuellement des humains)*
 - ✓ *Exemple : Notre Agent de sauvegarde interagit avec un autre Agent de sauvegarde sur un réseau. Ces deux Agents peuvent se mettre d'accord pour se partager le travail afin que la tâche soit achevée plus vite.*
- ◆ **Autonomie et Pro-activité**
 - ✓ *Des agents sont dits « pro-actif » si ils n'agissent pas seulement en réponse à leur environnement, mais ils sont capables d'avoir un comportement guidé par un but, en ayant la possibilité de prendre l'initiative.*
 - ✓ *Exemple : Un Agent réseau peut décider, de lui-même, pendant un temps où il est non actif, de faire des statistiques sur les routeurs pour améliorer son activité future.*
- ◆ **Capacité apprentissage**
 - ✓ *Un Agent aura la capacité d'apprendre si il sait acquérir de la connaissance, de l'information ou des habitudes.*
 - ✓ *Exemple : Un agent doit se déclencher pour un traitement de sauvegarde à une certaine heure. Mais l'utilisateur l'arrête dans sa tâche car il ralentit son travail. L'Agent va apprendre à différer son exécution pour éviter de gêner l'utilisateur à cette heure de la journée.*

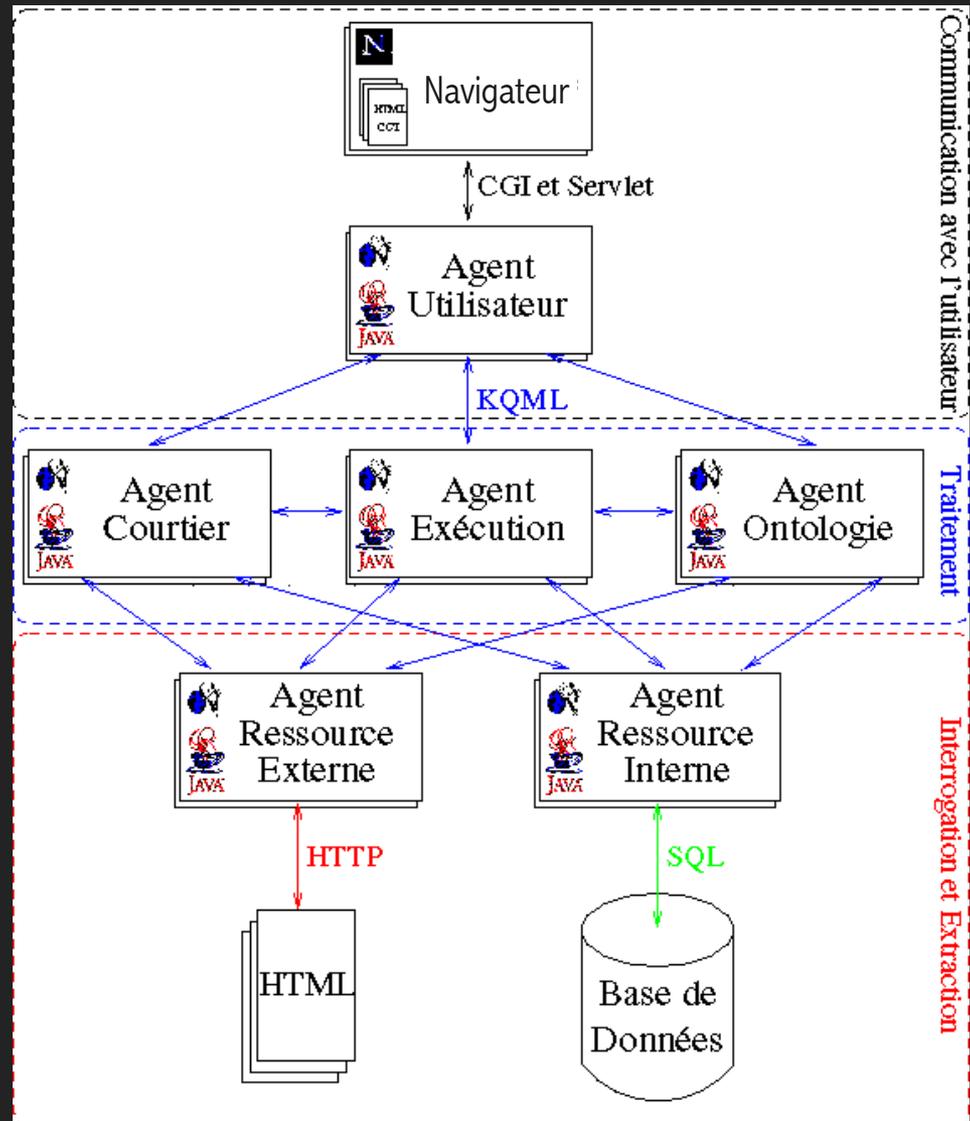
Architecture d'agents - exemple

Les agents de l'exemple :

- ✓ *Agent courtier* : c'est un Agent qui permet de faire de courtage d'information au sein d'un système. Il permet de trouver les Agents selon leurs services ou capacités.
- ✓ *Agent exécution* : c'est un Agent qui supervise l'exécution de certaines tâches complexes. Cet Agent décompose ces tâches en sous-tâches.
- ✓ *Agent ressource* : c'est un Agent qui contrôle une ou plusieurs sources de données.
- ✓ *Agent ontologie* : c'est un Agent qui s'occupe des différentes ontologies du système.
- ✓ *Agent utilisateur* : c'est un Agent qui prend en charge les tâches de l'usager.

Protocole d'échange et ontologie :

- ✓ *Les agents interagissent avec le support d'ontologies définissant l'univers de discours d'un domaine donné.*

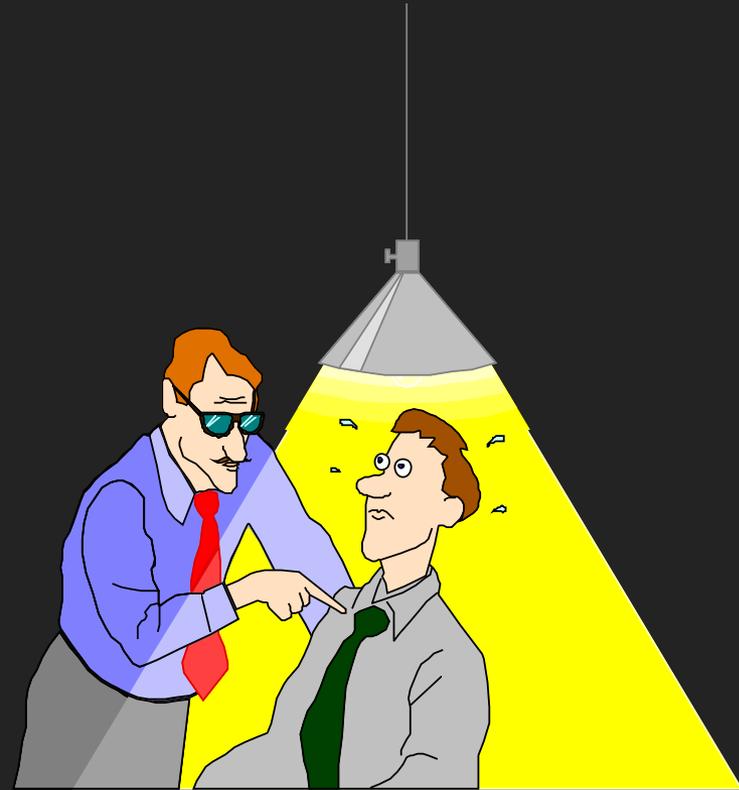


Cette figure fait référence au doc. suivant : LES AGENTS INTELLIGENTS SUR INTERNET, IRESTE, Stéphane Anglerot, Guillaume Bonnet, Guy Regnault

Architecture de SI

Génération et Typologie d'architectures

Questions ?



Règles d'or d'architecture applicative d'un SI (1)

- ◆ **Autonomie locale de traitement et accessibilité transparente maximale aux données communes**
 - √ *les données des serveurs sont partagées et contrôlées de manière transparente pour le client*
- ◆ **Assurance du fonctionnement continu des services**
 - √ *en cas de rajout de sites serveurs ou de SGBD*
 - √ *en cas de changement de version des logiciels*
- ◆ **Indépendance vis-à-vis de la localisation des infrastructures**
 - √ *utilisation de middleware assurant l'interaction avec les hôtes distants*
- ◆ **Indépendance vis-à-vis de la gestion des transactions**
 - √ *utilisation de protocoles de transaction fiables (terminaison...)*

Règles d'or d'architecture applicative d'un SI (2)

- ◆ Indépendance vis-à-vis de la gestion de l'intégrité des données
 - √ *les applicatifs clients ne doivent pas intégrer la gestion de l'intégrité qui est assurée par le serveur pour tous les clients*
- ◆ Indépendance vis à vis de l'application cliente, du réseau et du serveur de base de données
 - √ *les 3 niveaux : client (Matériel, OS, GUI) , réseau, serveur BD doivent pouvoir être modifiés sans impacter les autres*
- ◆ Architecture ouverte et coopérative
 - √ *fidélité aux standards depuis l'API jusqu'au SGBD*
- ◆ Gestion centralisée de la coordination de l'architecture applicative globale
 - √ *allègement des applicatifs clients et simplification du système*

Règles des architectures orientée web

- ✓ *Le client se contente de gérer l'aspect, le comportement de l'interface utilisateur et les contrôles de saisies (C/S de présentation)*
- ✓ *L'essentiel des traitements surtout les accès et mises à jour de données sont réalisés à partir d'un serveur*
- ✓ *Les application sont simples et fonctionnent sur l'auto-apprentissage : aide en ligne, pas de manuel*
- ✓ *Internet et Intranet sont traités sur le même plan avec un niveau de performance raisonnablement équivalent*
- ✓ *L'interface utilisateur est soignée*

Architecture applicative des SI

Règles d'or

Questions ?

