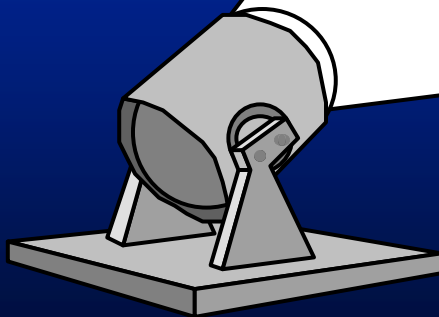


Client / Serveur

Rémy Courdier



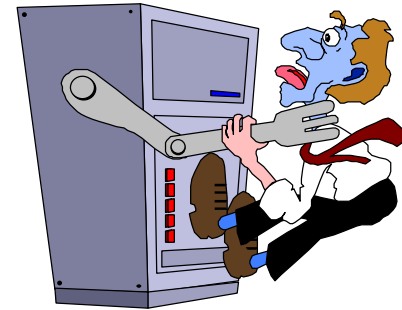
Génération et Typologie
d'architecture C/S

Troisième partie :

Les 3 Générations du Client-Serveur

- ✓ **1ere Génération :**
"Le traitement frontal"
- ✓ **2ème Generation :**
"Le traitement Coopératif"
- ✓ **3ème Génération :**
"Le traitement réparti et les données Réparties"
- ✓ **L'internet : vers le Client/Serveur universel**
- ✓ **Tendances**

Tendances...



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

Le traitement frontal

1^{ère} Génération

◆ C/S de présentation

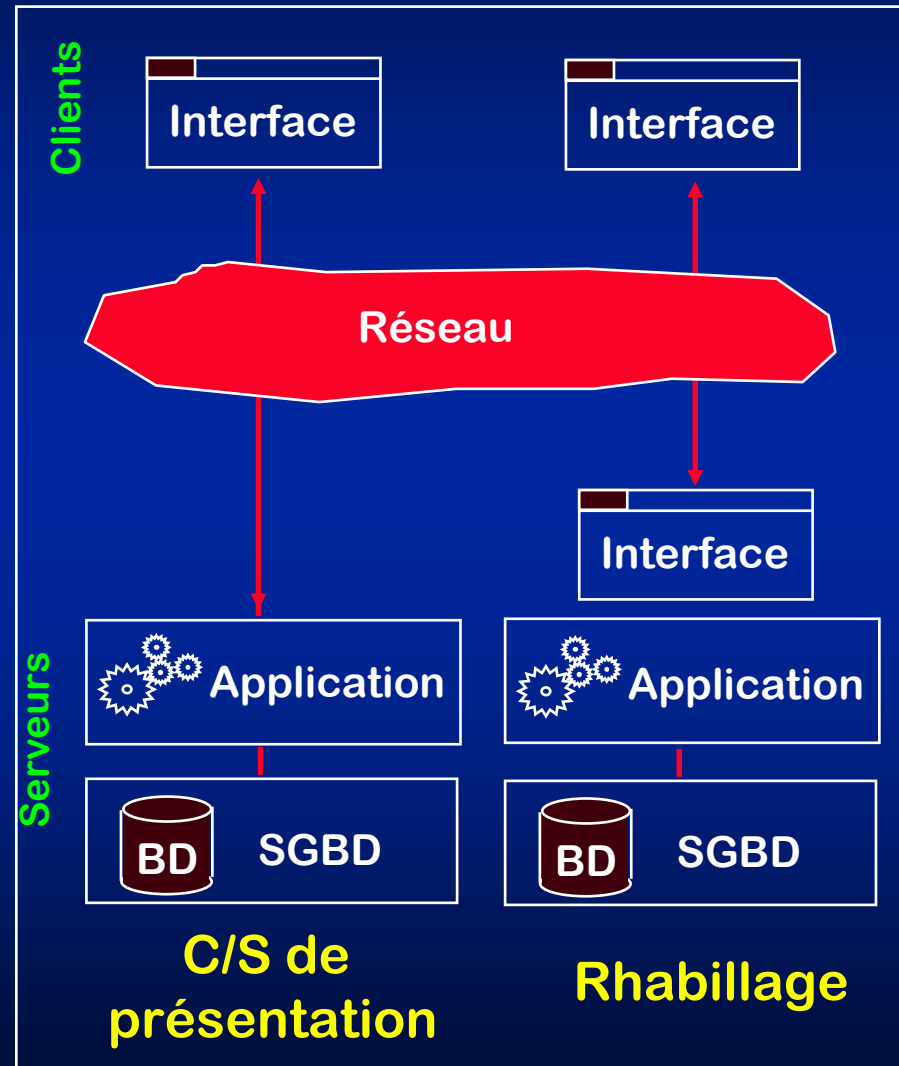
Déport du Dialogue :

- ✓ *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*



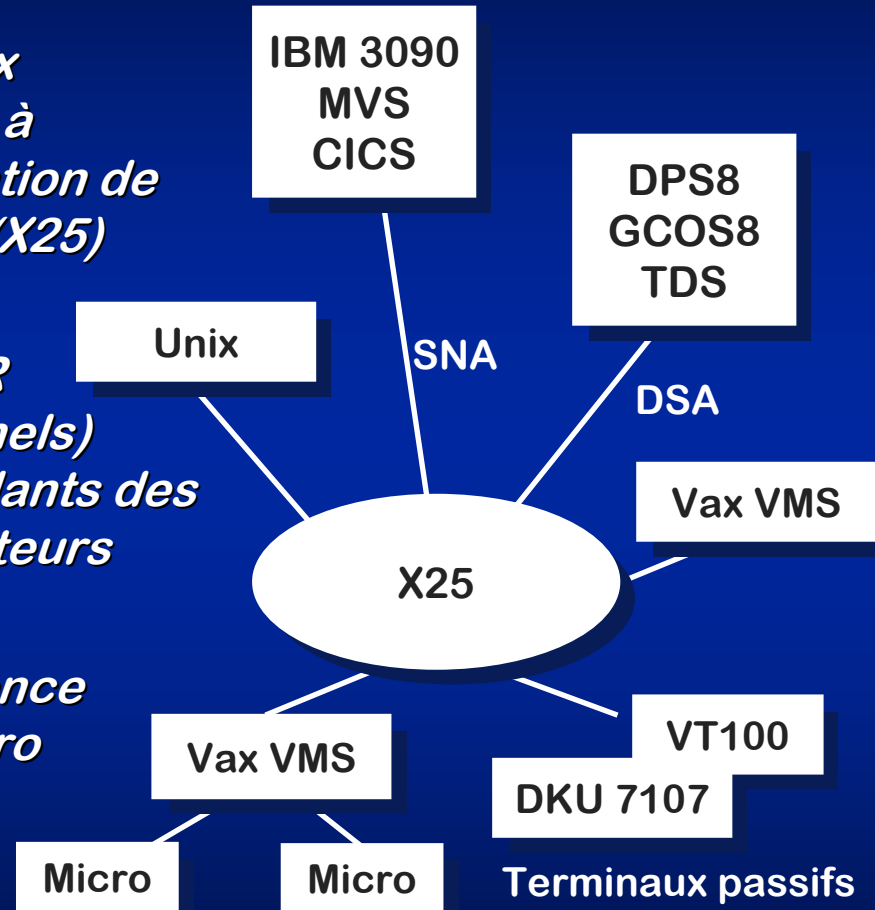
Evolution des archi. informationnelles



▣ Réseaux partagés à commutation de paquets (X25)

▣ SGBD-R (relationnels) indépendants des constructeurs

▣ Emergence de la micro

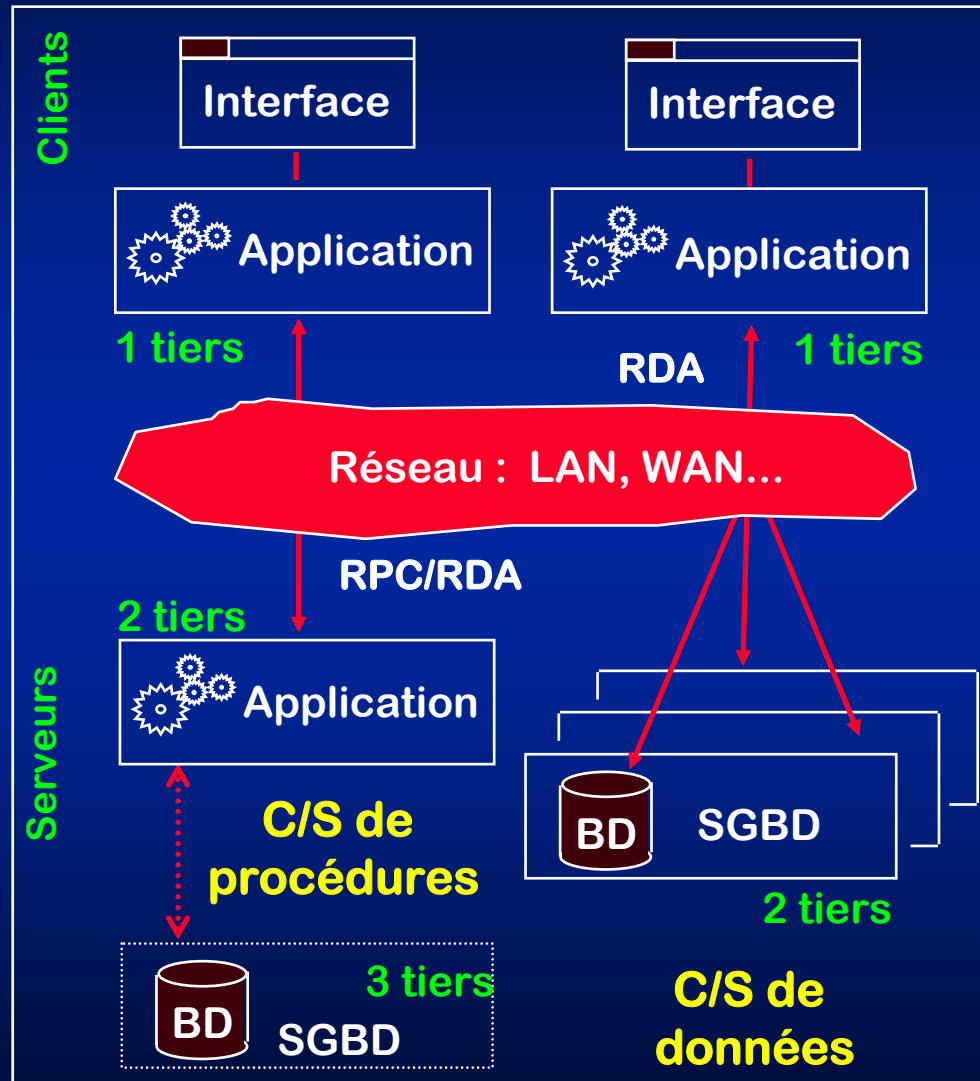


vers le traitement coopératif...

Le traitement Coopératif

2^{ème} Génération

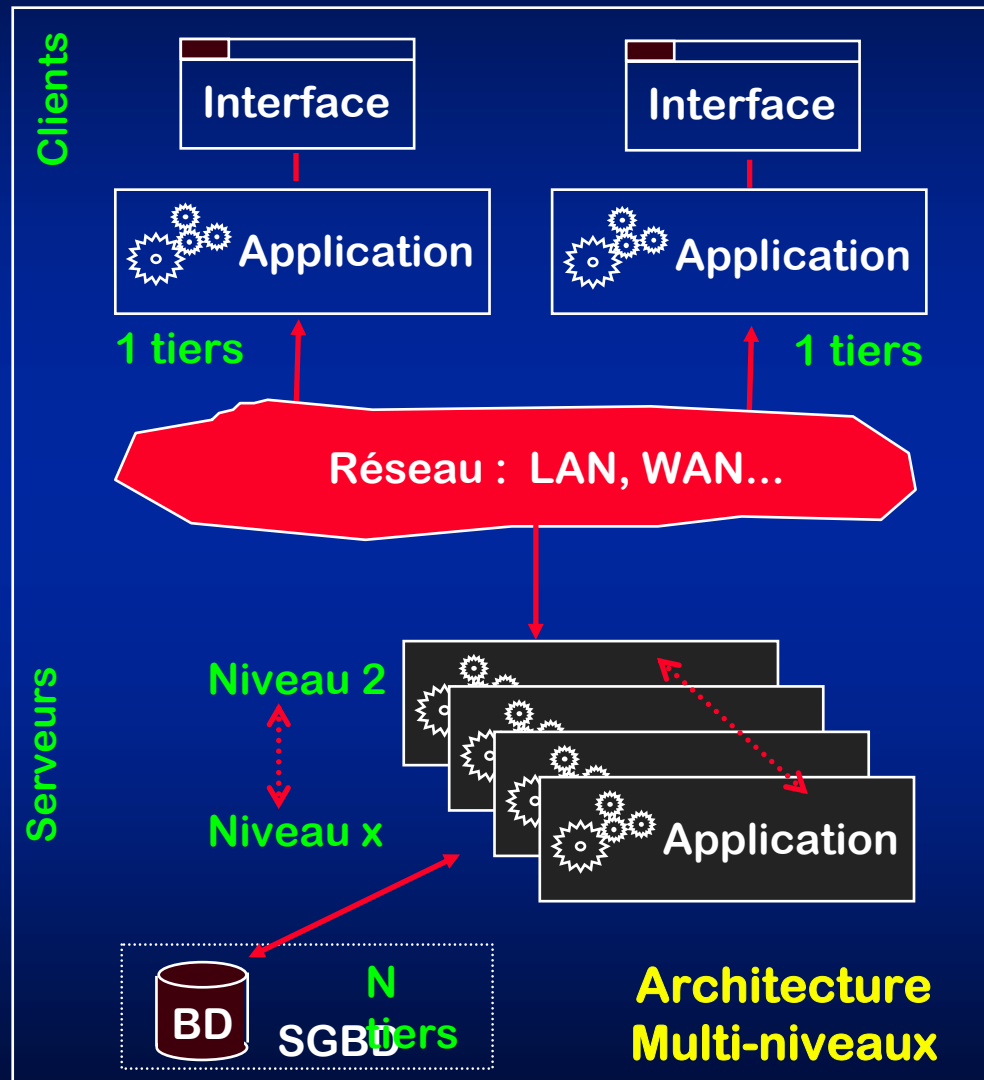
- ◆ C/S de procédures
 - ✓ *L'application Cliente sous-traite l'exécution de proc. à un serveur*
 - ✓ *Répartition partielle de l'application entre client et serveur*
- ◆ C/S de données
 - ✓ *Déport de l'application complète sur le poste client : accès à des données distantes*
 - ✓ *Possibilité d'accès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



Le traitement Coopératif multi-niveaux

évolution de la 2^{ème} Génération

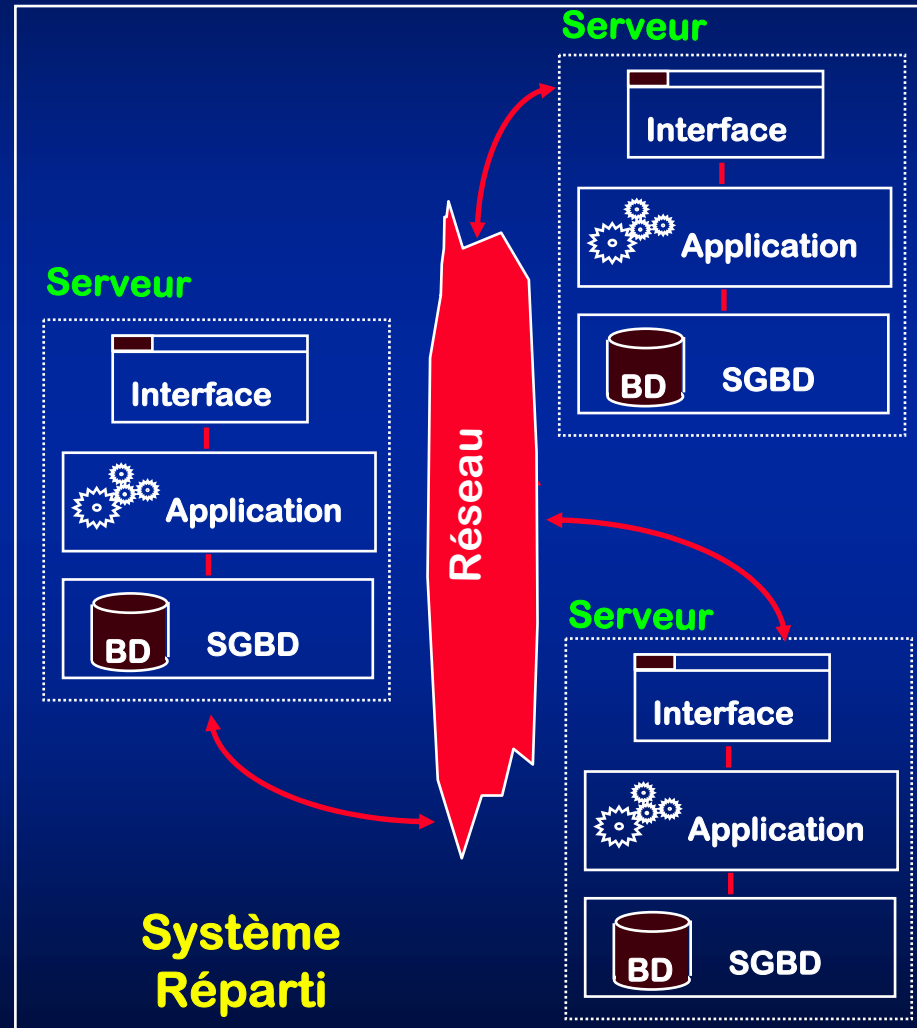
- ◆ Dans l'architecture coopérative chaque serveur effectue un service spécialisé.
- ◆ Un serveur peut donc invoquer un service d'un autre serveur
- ◆ L'architecture 3 tiers et potentiellement une architecture N-tiers à N Niveaux



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*



Gérer la complexité de la répartition...

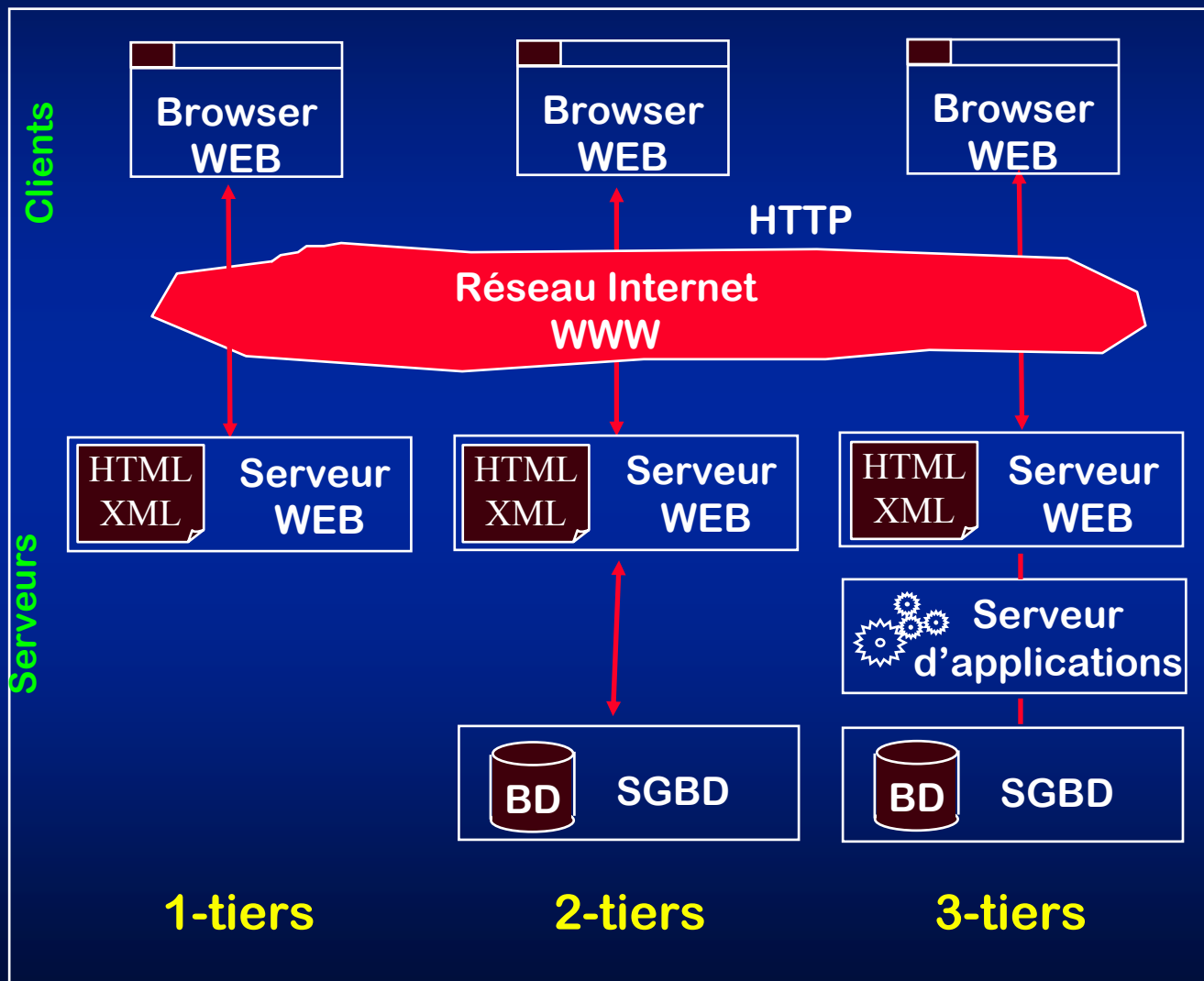
- ◆ Décomposition d'une requête en sous-requêtes mono- ou multi-sites
- ◆ Principe d'optimisation de requêtes multi-sites
- ◆ Existence d'un dictionnaire global :
 - √ *localisation des données*
 - √ *transparence pour l'émetteur de requêtes*
- ◆ Utilisation automatique de traducteurs pour l'accès aux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

L'internet : vers le C/S universel

“Le deuxième âge du C/S”

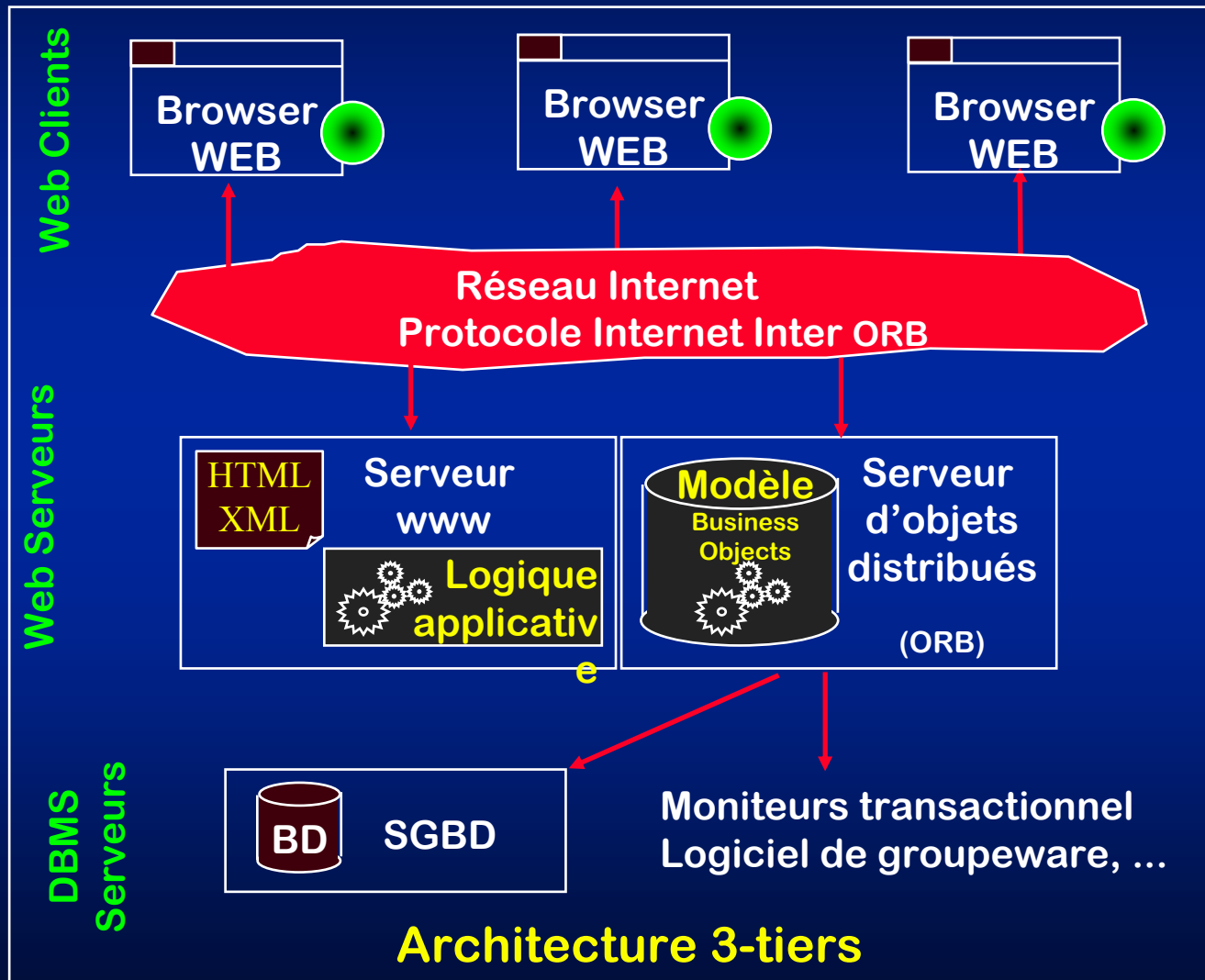
*Le retour du C/S
de présentation
Utilisation de
standards
simples*

*Le client fait un
travail
d'interprétation
de fichiers et de
gestion de
communication
avec le serveur
1 connexion par
objet contenu
dans la page*



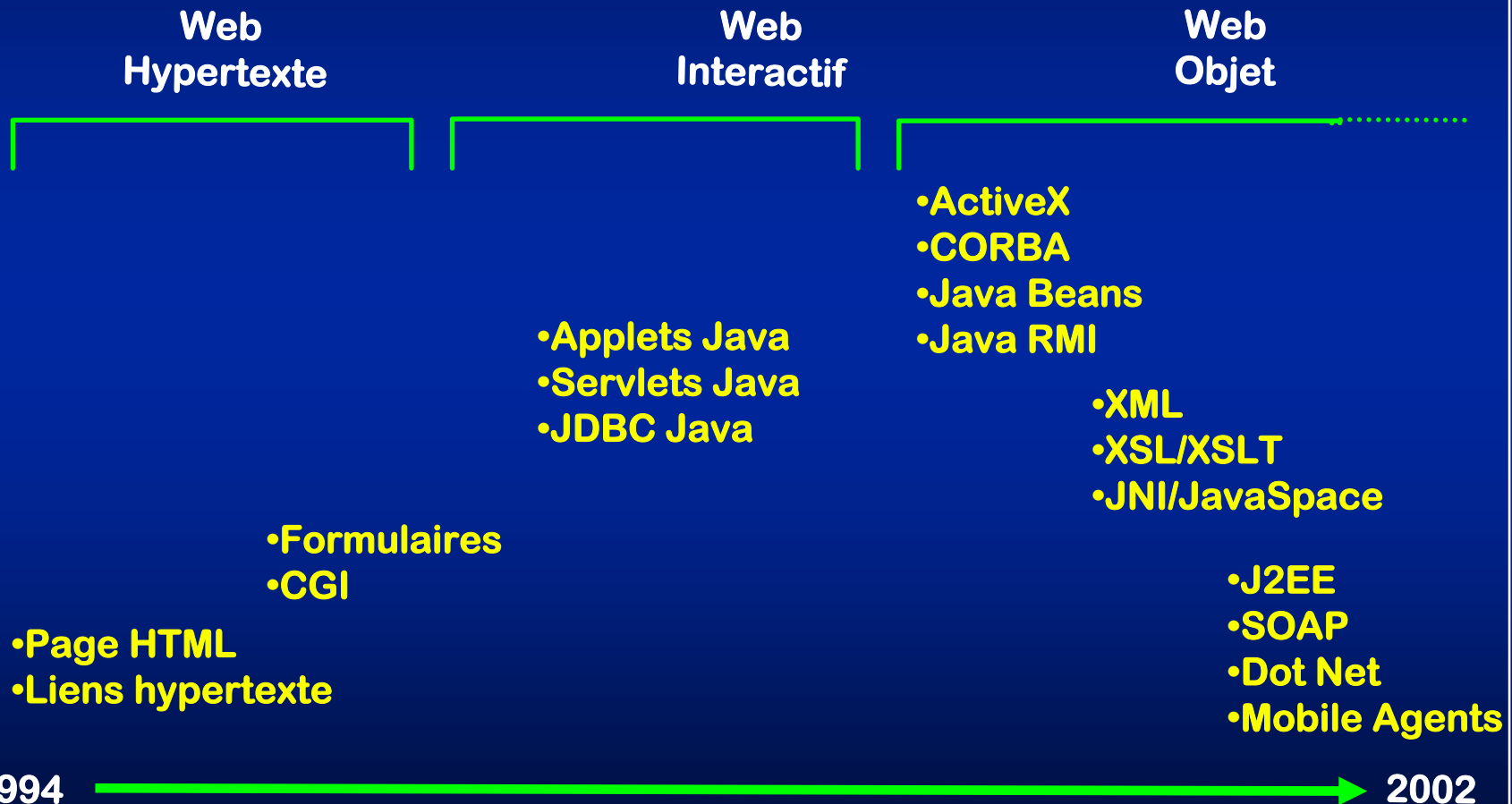
L'Internet : vers le C/S réparti

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



Systemes C/S répartis et le Web

Evolution de la technologie Web



Les tendances...vers le C/S Internet Maîtrisé

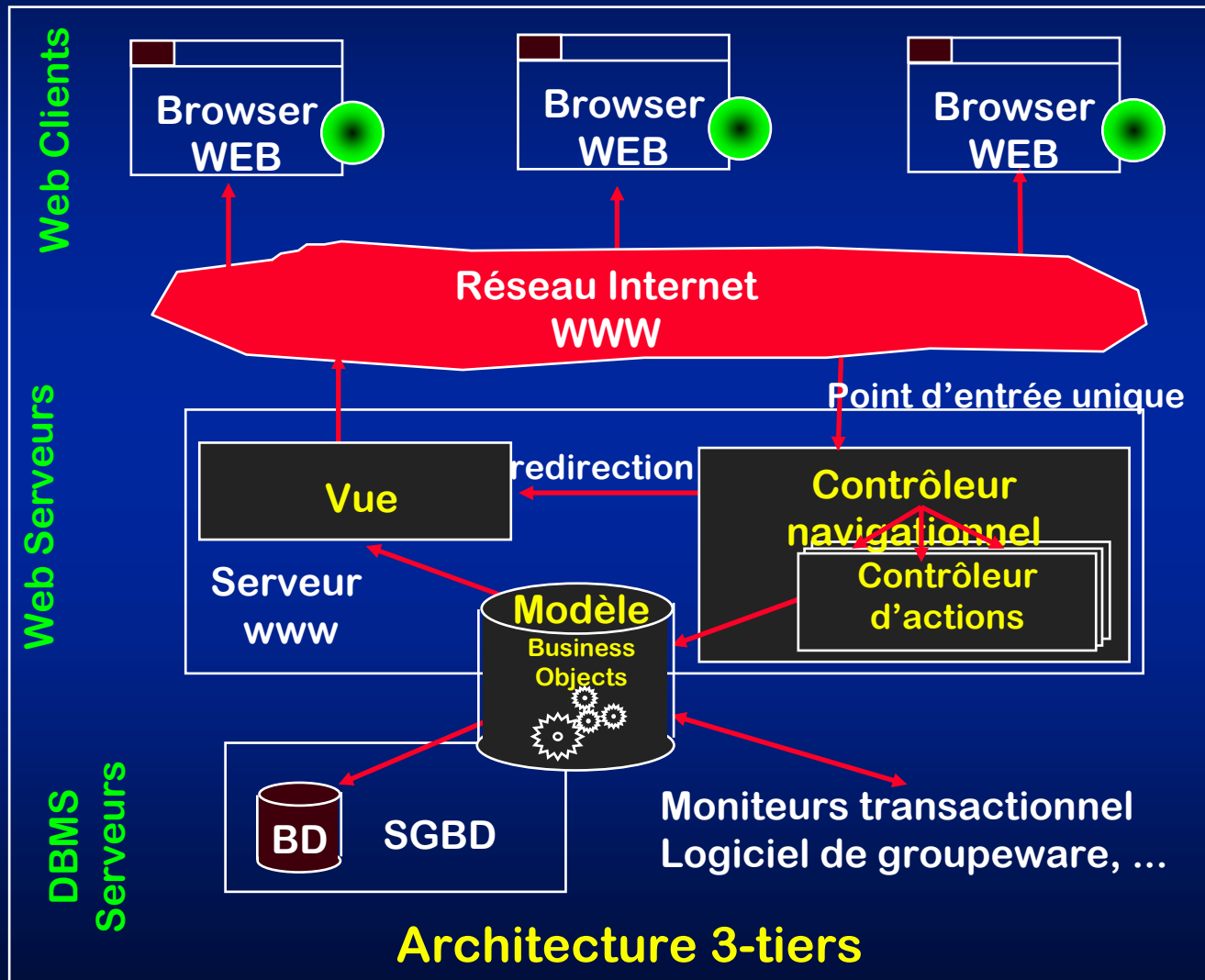
- ◆ *Mise en oeuvre du Modèle MVC dans les architectures Client/serveur de l'Internet*
- ◆ *Le modèle MVC : Modèle-Vue-Contrôleur*
 - √ *Modèle/Model*
 - ⊙ *Représente le domaine d'application, donnée & logique opératoire*
 - √ *Vue/View*
 - ⊙ *Représente l'ensemble des éléments de présentation externes*
 - √ *Contrôleur/Controller*
 - ⊙ *Module responsable des règles de navigation et d'enchaînement des actions utilisateurs*

Design Pattern Issu du génie logiciel objet, notamment de Smalltalk

L'intérêt : vers une méthodologie de conception permettant la

Architecture C/S de l'Internet & 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



Tendances... Les agents sur Internet

- ◆ Qu'est ce qu'un agent ?
 - √ *Domaine très récent. -> pas de définition universelle.*
 - √ *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 »*
- ◆ la définition donnée par Caglayan et Harrison [CAG] : *Agent logiciel : entité informatique qui réalise de manière autonome des tâches pour un utilisateur.*
- ◆ Un Agent est orientée « tâche », il déploie une activité (suite de fonctionnalités offertes par son environnement) dans le but de faire quelque chose et sera caractérisée 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 avancée des agents



◆ La capacité d'apprendre

- ✓ *Un Agent aura la capacité d'apprendre si il sait acquérir de la connaissance, de l'information ou des habitudes.*
- ✓ *Exemple : Un Agent grâce à sa capacité de réactivité, doit se déclencher à une certaine heure. Mais l'utilisateur l'arrête dans sa tâche (qui pourrait être une sauvegarde) car il ralentit le travail de l'utilisateur. L'Agent va apprendre à différer son exécution pour éviter de gêner l'utilisateur.*

◆ La capacité sociale

- ✓ *Les Agents interagissent avec les autre Agents (et éventuellement des êtres humains) grâce à des langages de communication entre Agents. Cette capacité est à la base pour la coopération entre les Agents.*
- ✓ *Exemple : Notre Agent de sauvegarde rencontre 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.*

◆ Haut degré d'autonomie et Pro-activité

- ✓ *L'Agent fonctionne sans intervention directe humaine ou autre et a une forme de contrôle sur ses actions et sur leur état interne.*
- ✓ *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.*

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

Architecture d'agents - exemple 1

◆ Ontologie :

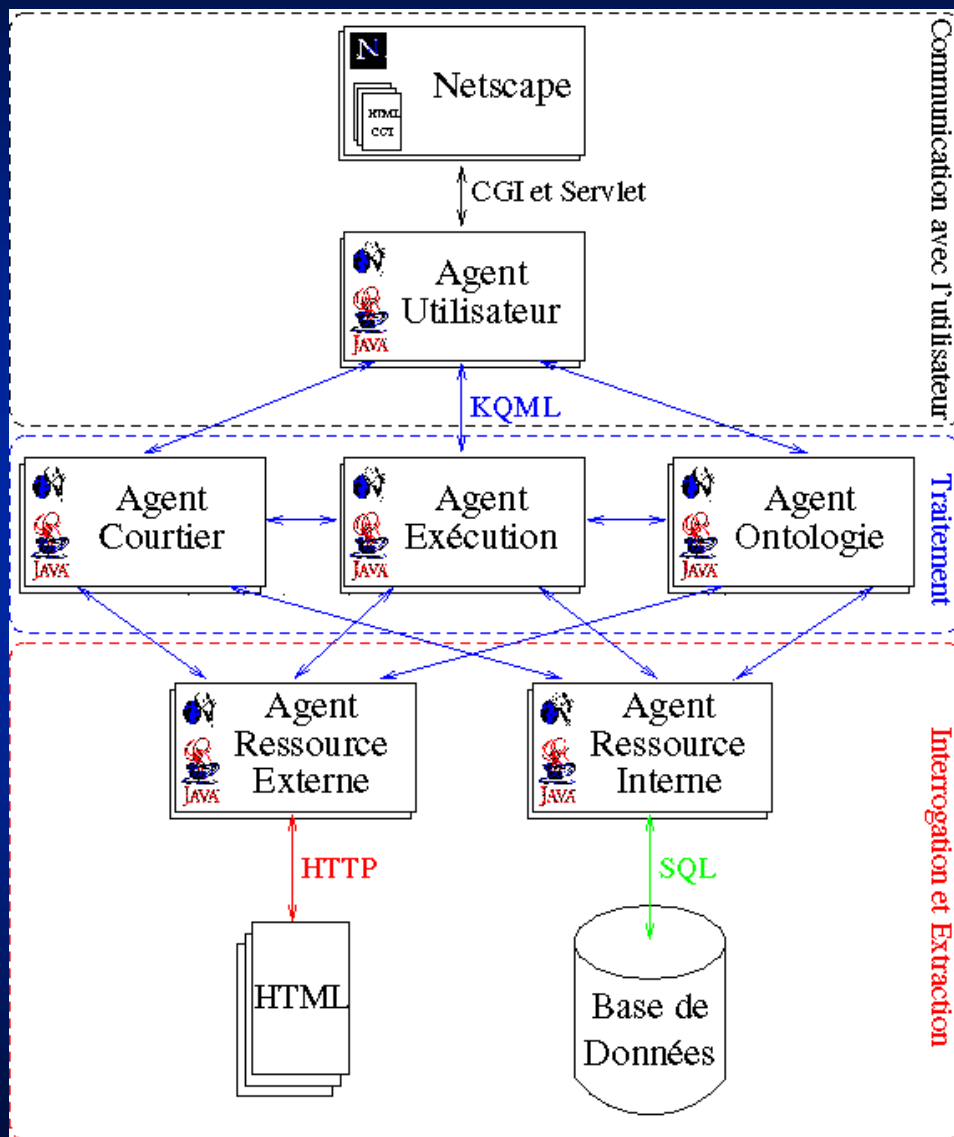
- ✓ Une ontologie peut être considérée comme l'univers de discours d'un domaine donné.

◆ KQML :

- ✓ Knowledge Query and Manipulation Langage
- ✓ Définit un ensemble de performatifs, verbes dérivés des actes du langage naturel
- ✓ www.csee.umbc.edu/kqml/

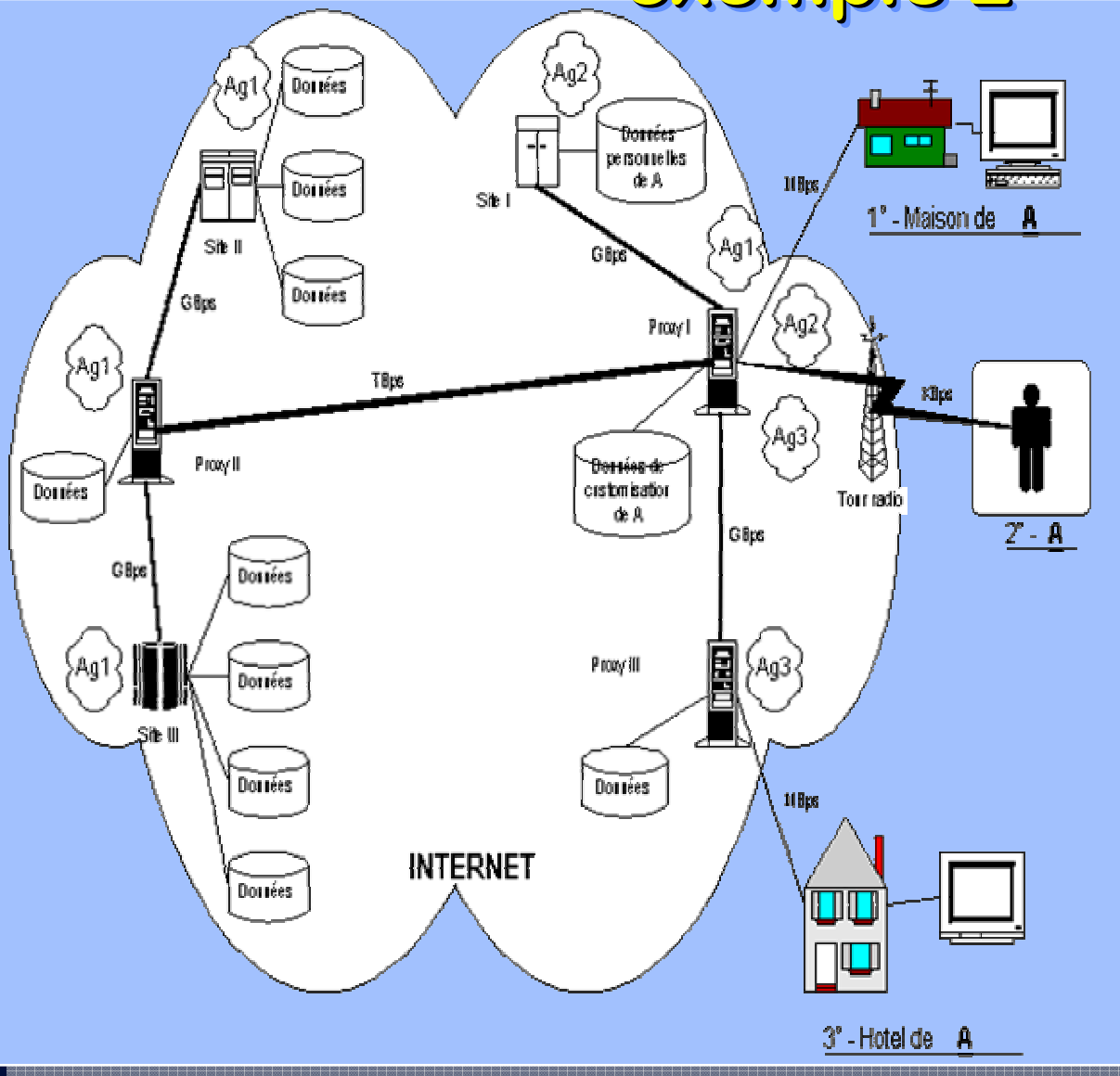
◆ Les agents de l'exemple :

- ✓ **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. Ces sources peuvent être internes ou externes.
- ✓ **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'utilisateur.



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

Architecture d'agents mobiles - exemple 2



1. **A se connecte à partir de l'ordinateur de sa maison. Il accède à ses données personnelles (sur le site I) et reçoit les informations personnalisées du Proxy I. Il démarre une recherche à travers le net, sur un ensemble de sites. Un Agent se déplacera sur tous les sites choisis, l'un après l'autre, et fera la recherche sur les bases de données hébergées sur le site.**
1. **A se connecte à partir de son téléphone portable. Il veut consulter son courrier électronique qui est sur le site I. La communication, de mauvaise qualité, est interrompue plusieurs fois pendant le transfert.**
1. **A se connecte à partir de l'ordinateur de sa chambre d'hôtel. Il veut de nouveau consulter sa messagerie et retrouver un fichier dans ses données personnelles (sur le site I). Il en profite pour consulter les informations, personnalisées, envoyées par Proxy I. Ensuite, il récupère le résultat de la recherche effectuée par ses Agents mobiles.**

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

Le Grid Computing : Définition

◆ But et Principe

- √ *Utiliser les ressources inexploitées des systèmes informatiques.*
- √ *Une grille est un ensemble d'ordinateurs dépourvu de serveur central capable de regrouper ses ressources inemployées - puissance de calcul et stockage - pour les redistribuer à un ou plusieurs bénéficiaires.*
- √ *Tout ressemble beaucoup à un système d'exploitation : chaque machine reliée à la grille prend en charge la tâche dont l'ensemble du système a besoin".*

◆ Avantage et Limites :

- √ *Puissance de calcul utilisable & Espace de stockage considérables libérés.*
- √ *Limites : les problèmes de parallélisation des traitements ainsi que les taux de transferts : cette techno n'est mal adaptée aux applications qui ont besoin d'échanges de données très rapides entre processeurs.*

◆ Utilisation :

- √ *Les grilles sont donc surtout utilisées par des centres de recherche privés ou publics qui ont besoin d'une puissance de calcul immense.*
 - Exemple programme seti@home aide à la recherche d'éventuels d'extraterrestre
 - Le Decrypton en France qui a permis de décrypter le Protéome
- √ *Utilisation professionnelle :*
- √ *"Google utilise une ferme de serveurs reliés entre eux pour gérer le volume des connexions qu'il reçoit. La grille - ou plutôt l'ancêtre de grille - ainsi constitué forme un seul cluster géant".*

◆ Niveau de maturité

- √ *la technologie n'est pas encore tout à fait mûre*
- √ *la référence technologique en la matière pour la partie technique est l'infrastructure de la grille : Globus - qui est en open source". www.globus.com*

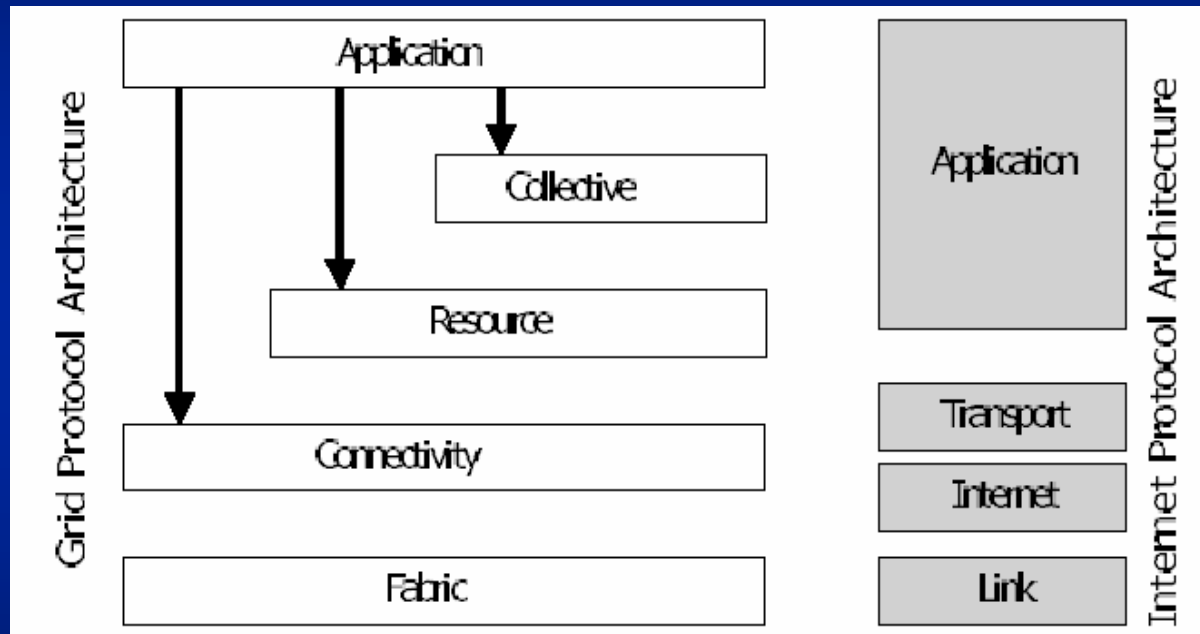
Le Grid Computing : Normalisation

- ◆ Organisme de normalisation :
 - √ *le Global Grid Forum - NDLR: organisme qui est au Grid Computing ce que le W3C est au Web)*
 - √ *P2P Working Group*
 - √ *Ces deux organisme ont fusionnés en 2002*
- ◆ Types :
 - √ *Grilles d'information : partage d'informations*
 - √ *Grilles de stockage : multiplication des possibilités de stockage*
 - √ *Grilles de calcul : multiplication des « flops »*
- ◆ Positionnement :
 - √ *Cluster Computing : 10-100 systèmes interconnectés*
 - √ *Grid Computing : Dans les milliers de systèmes interconnectés*
 - √ *P2P : Des millions de systèmes*

Les contraintes les services et les niveaux de sécurités ne sont pas les mêmes

Le Grid Computing : Protocol

Les couches de l'architecture de grille et ces relations par rapport aux architecture de protocole Internet. Il y a un mapping entre ces deux architecture puisque chacune d'entre elle part du réseau physique pour arriver aux applications.

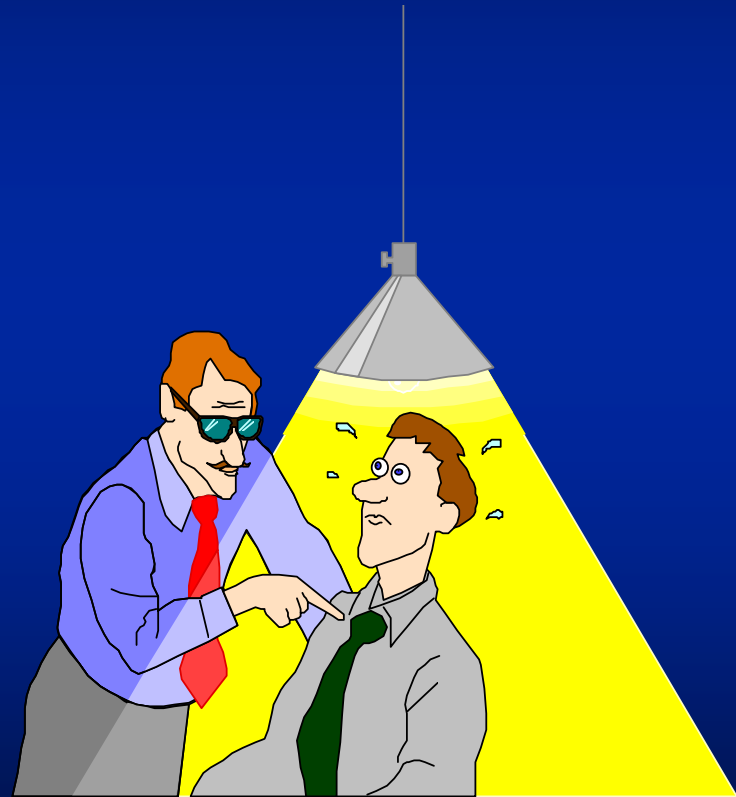


Architecture définie dans Globus, et utilisée dans le NASA's Information Power Grid, l'European Data Grid, le Particle Data Grid, ...

Client / Serveur

Génération et Typologie de C/S

Questions ?



Règles du Client/ Serveur

✓ ***Les 8 règles
d'or du C/S
premières
générations***

✓ ***Les 7 règles
d'or du C/S
internet***



Et de 4...

- ✓ *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 du client et du serveur*
 - ⊙ 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 serveurs*
 - ⊙ utilisation de middleware assurant l'interaction avec les serveurs distants
- ✓ *Indépendance vis-à-vis de la gestion des transactions*

...pour arriver à 8

- ✓ *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 ts 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 SQL 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 globale*
 - ⊙ allègement des applicatifs clients et simplification du système

Règles du C/S de l'Internet

- ✓ *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 applications 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*

Client / Serveur

Règles d'or

Questions ?

