

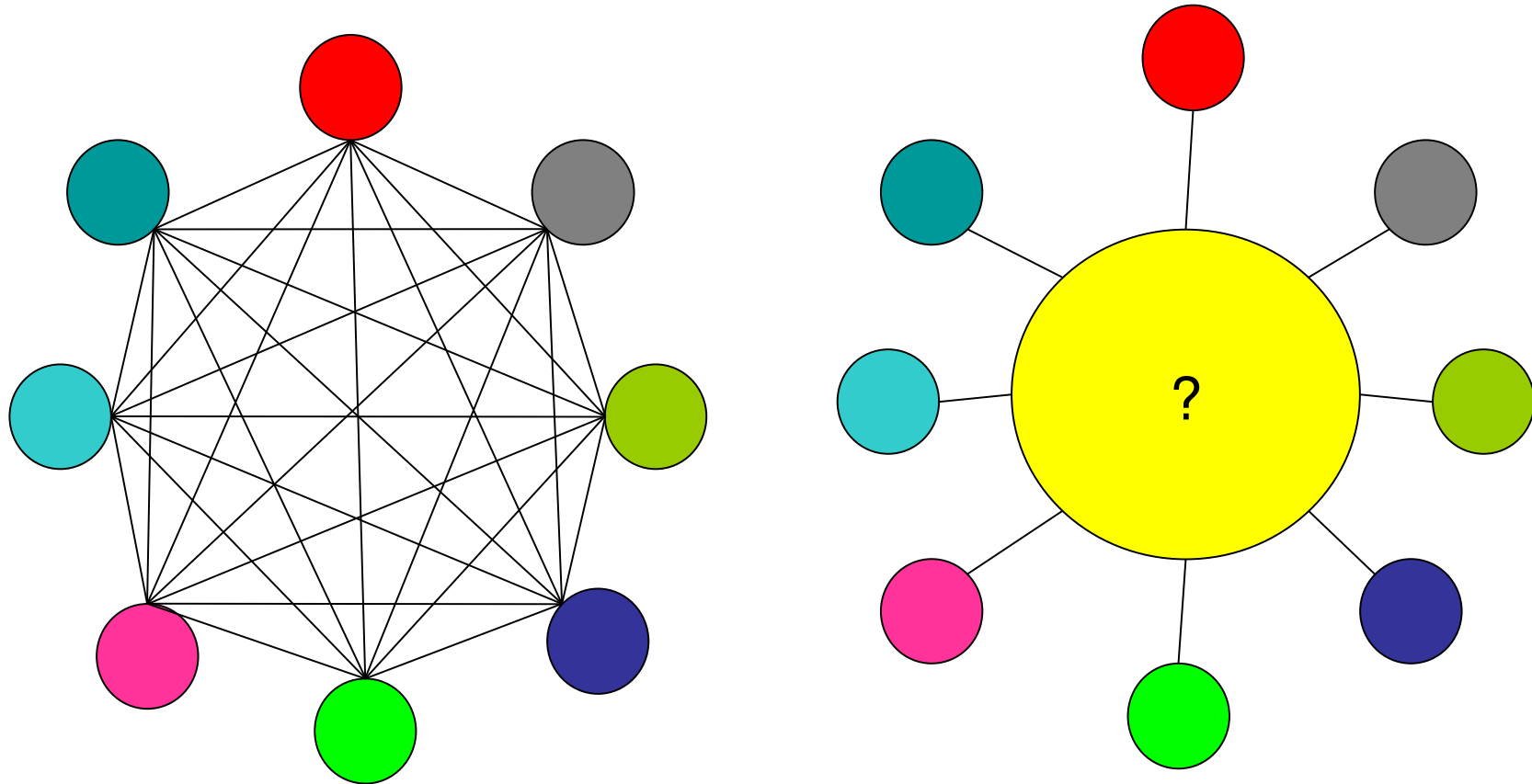
# Urbanisation des SI

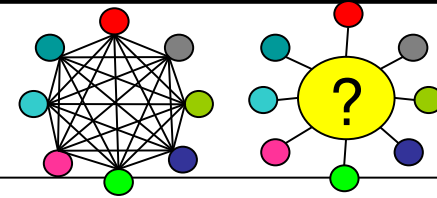
Des composants technologiques disponibles

# Plan de l'exposé

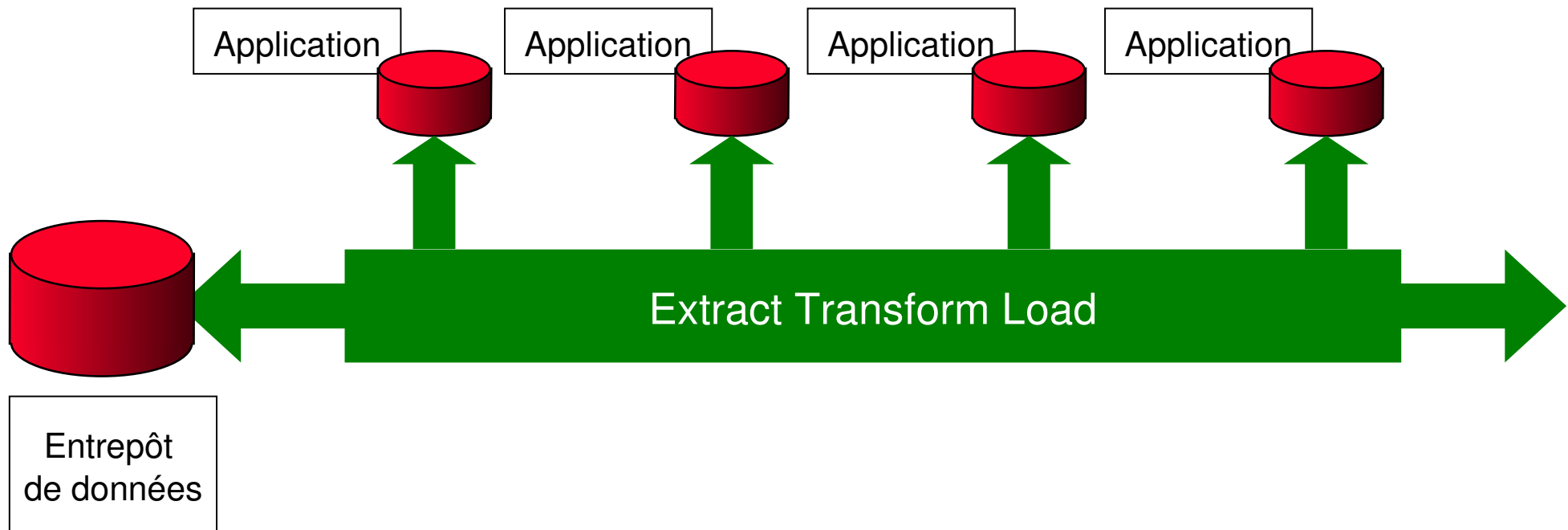
- Technologies à la mode disponibles.
  - Bus de données, ETL et EAI
  - GED et SGDT
  - Portails Web
  - Progiciels de Gestion Intégrés, (PGI, ERP),
  - Web services et SOA
- Les démarches de standardisation
  - Différence entre norme et standard Intérêts et inconvénients
  - Rôle des États et rôle du marché.
  - Disparité, redondances et lacunes des démarches
    - UN/CEFACT
    - ebXML, RosettaNet, STEP, PLIB, CIM, etc.
- Les méthodes & technologies émergentes
  - Model Driven Architecture
  - Les technologies du Web sémantique, Ontologies d'accès et d'intégrité.

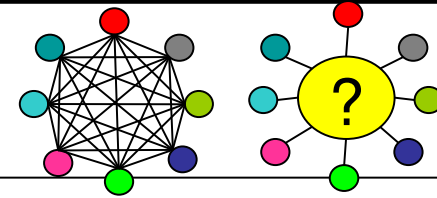
# Retour sur une problématique : les méthodes et outils d'échange de données





- Technologies d'interopérabilité
  - 1) Les ETL : Extract Transform Load
    - Outils Intergiciels (« Middleware ») de transfert de données entre bases de données
    - Connecteurs SQL,
    - agrégation, filtrage, conversion de données
    - Extracteurs/Chargeurs en masse



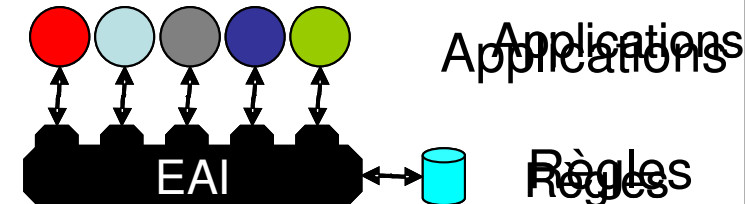


- Liaison entre applications via un Intergiciel
  - Typiquement le traitement des couches 5, 6 et 7 du modèle OSI « session », « présentation », « application »
- Concept d'EAI (Enterprise Application Integrator)

- Deux architectures

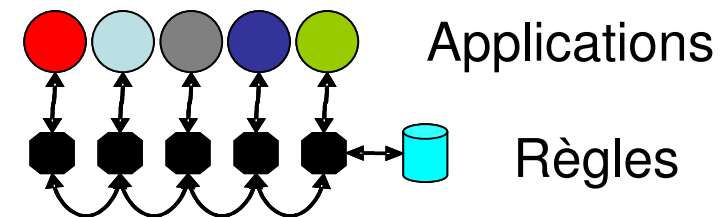
- L'architecture « Hub and spoke »

- » Seebeyond, Sun (iPlanet) et Tibco



- L'architecture « Network Centric »

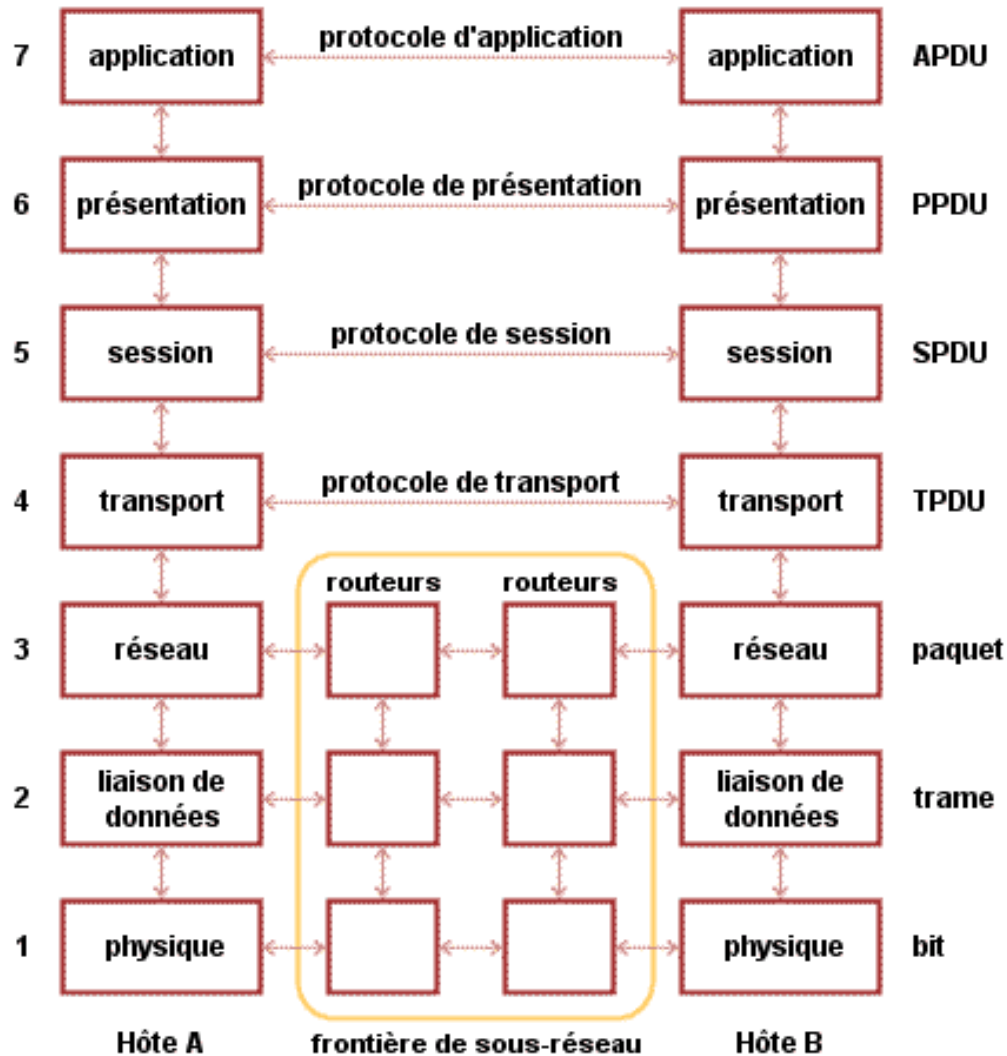
- » Web-Methods, Sybase-NEON, BEA, Vitra et Mercator



- 4 fonctions

- routage (d'une application à l'autre),
  - transformation (des données),
  - connecteurs (aux applications),
  - transport physique (des informations).

# Le modèle OSI (source wikipedia)



La **couche « application »** est le point d'accès aux services réseaux, elle n'a pas de service propre spécifique et entrant dans la portée de la norme.

La **couche « présentation »** effectue le codage des données applicatives, précisément de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises.

La **couche « session »** gère la synchronisation des échanges et les «transactions», permet l'ouverture et la fermeture de session.

La **couche « transport »** gère les communications de bout en bout entre processus (programmes en cours d'exécution).

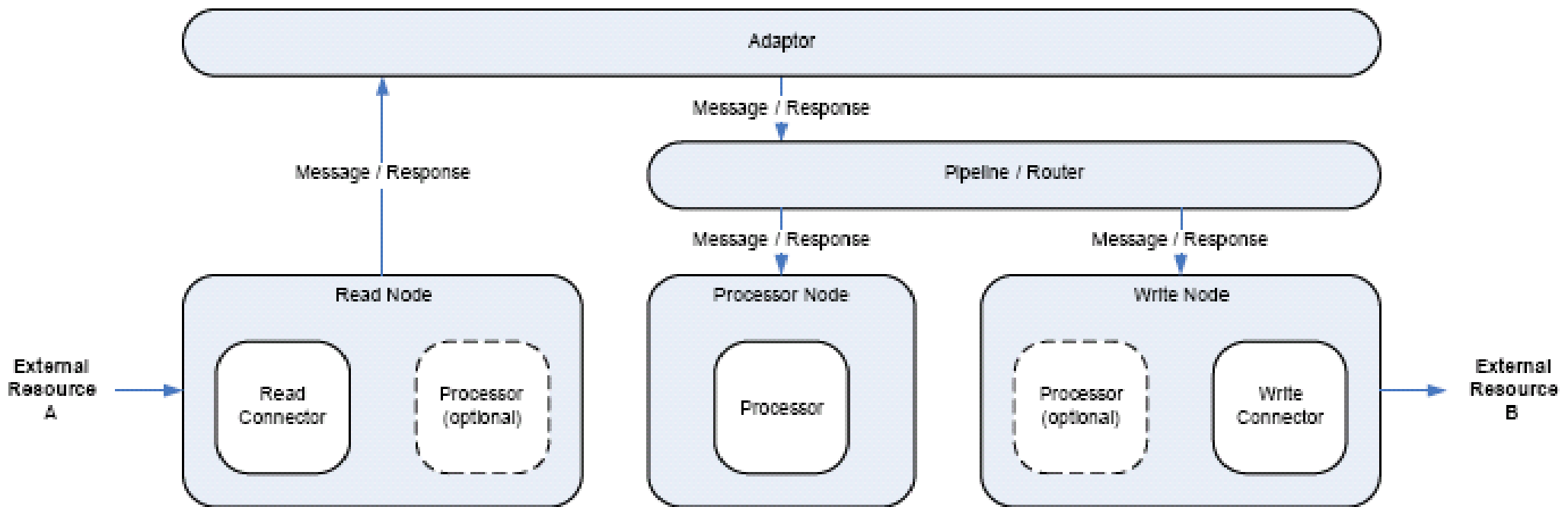
La **couche « réseau »** gère les communications de bout en bout, généralement entre machines : routage et adressage des paquets.

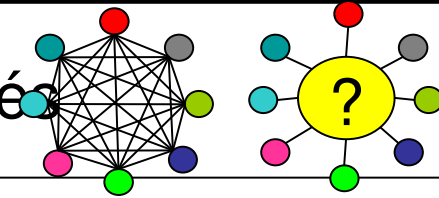
La **couche « liaison de données »** gère les communications entre machines adjacentes, directement reliées entre elles par un support physique.

La **couche « physique »** traite transmission effective des signaux entre les machines via un support physique. émission et réception d'un train de bits continu

# Technologies d'interopérabilité : les EAI

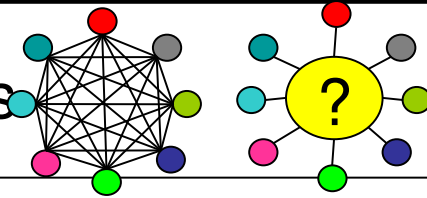
- Source : openAdaptator (un EAI libre)



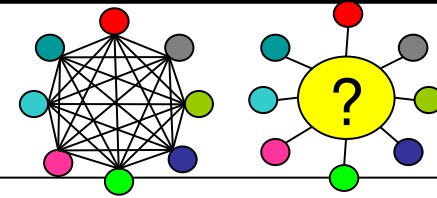


- Solution de Couplage fort, par « Object Request Broker », des objets distribués
  - L'appel de procédures et l'invocation d'objets à distance exécute une fonction d'un autre exécutable pouvant être sur une machine distante.
  - Il permet à des objets distribués de communiquer par appel de méthodes.
  - Chaque procédure peut s'exécuter sur machine distante comme si elle était locale.
    - « Dans l'offre Microsoft .NET, Net Remoting est un mécanisme de communication et de transfert de données pour des applications distribuées .Net ouvert à des applications autres via des « ponts », vers des objets écrits en Java par exemple »
    - « Dans l'offre libre Java de Sun, RMI (Remote Method Invocation) est une technique permettant de manipuler des objets distants (instanciés sur une autre machine virtuelle, éventuellement sur une autre machine du réseau) de manière transparente pour l'utilisateur, c'est-à-dire de la même façon que si l'objet était sur la machine virtuelle (JVM) de la machine locale ».
- Comme pour les solutions de partage de bases de données, le couplage fort nécessite des descriptions partagées a priori (modèles) des objets invoqués
- Les ORB donnent une approche pratique pour des « quartiers » d'un domaine de SI urbanisé, lorsque ses applications partagent a priori ses contraintes de conception architecturale





- Une technologie Web
  - autorisant
    - le découpage des applications et leur répartition sur le système d'information et ses infrastructures
    - L'hétérogénéité des langages de développement et des architectures
    - L'appel (d'offres?) de « subroutines » disponibles sur tout réseau accessible
    - Une édition de lien effectivement distendue par un couplage faible
    - La fiabilisation par la concurrence des ressources de services disponibles
    - Le déplacement des limites contractuelles de fournitures de services
    - La facturation à l'usage, en remplacement des licences complètes
  - imposant
    - La désynchronisation du fonctionnement des applications
    - Des performances réduites par la charge de transformations et de communication via le Web
  - 4 standards XML
    - SOAP, WSDL, UDDI, BPEL



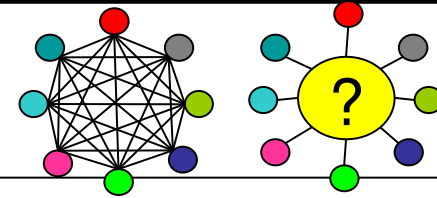
- SOA ou « Service Oriented Architecture est une façon d'organiser un système informatique comme fournisseur de (Web) services au Système d'Information
  - Adaptation d'applications anciennes ou nouvelles sous la forme de services rendus par des applications source à des applications cibles
  - Inclusion de façon non exclusive les solutions de Web services
  - Intégration d'autres solutions, via notamment l'initiative SCA « Service Component architecture ».
- Principe : chaque application « expose » les services qu'elle peut fournir et en définit les conditions d'appel et de retour des informations
- L'expression des entrées et des résultats s'effectue dans un dialecte XML (SOAP)

# Point d'orgue : Orientation Objet ou Orientation Services

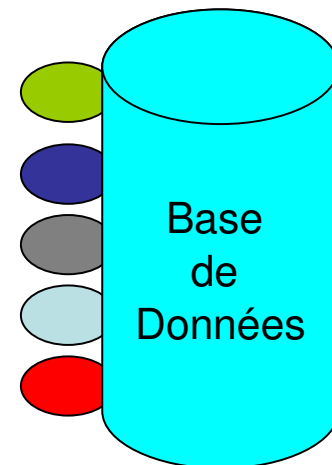
- L'orientation Objet :
  - Programmation *déclarative* fondée sur une description d'entités d'un monde réel, et de leur comportement mutuel attendu
    - Les objets sont distribués via des architectures de courtage
- L'orientation Service :
  - Programmation *procédurale* fondée sur des appels paramétrés de services programmes externes
    - dont seul le résultat écrit dans un message importe,
    - dont l'architecture peut être orientée objet, sans bénéfice d'utilisation
    - l'expérience montre qu'il est préférable d'y séparer données et traitements, quitte à découper les objets.
- L'orientation service :
  - Correspond aux préoccupations du commerce électronique
  - Simplifie l'urbanisation de systèmes d'information
    - Autorise la conservation d'applications anciennes
    - Autorise l'hétérogénéité de toutes les composantes d'un SI

# Technologie progicielle : La GED et les SGDT

- GED : Gestion Électronique de Documents
- SGDT : Système de Gestion de Données Techniques
  - Ce qu'ils ont en commun :
    - Le stockage de documents
    - La gestion de version, et de variantes successives
    - La gestion de configurations et d'applicabilités
    - La gestion de cycles de vie des éléments représentés
    - La gestion de processus (éditoriaux et de publication pour la GED)
  - Ce qui leur est spécifique
    - L'intégration d'applications techniques pour les SGDT (CAO, Calcul...)
    - L'intégration d'accès aux contenus documentaires pour la GED (indexation, moteurs de recherche etc.)
- GED et SGDT concernent ordinairement des « quartiers » urbanisés (hélas) de façon indépendante.
  - Réaliser une coopération entre workflow indépendants est « difficile ».



- L'application Unique, composite, propriétaire
  - fondée sur le partage d'une base de données
    - Modèles de données et modalités d'expression d'information communs par construction
    - Concurrence d'accès gérée par le système de gestion de la base
      - API ODBC (JDBC pour Sun, ADO.NET pour Microsoft)
    - L'architecture tient lieu d'urbanisme :  
l'application se greffe de multiples modules, avec leur cycle de vie propre
    - Dilemme:
      - Le courage (ou l'inconscience ?) du développement propriétaire homogène de l'ensemble de ses processus métiers
      - L'introuvable progiciel s'adaptant à toutes les préoccupations des métiers,
      - La réduction des préoccupations à ce que peut un progiciel choisi
      - Le risque du développement d'extensions au progiciel choisi
- Cf. les progiciels de gestion intégrée (PGI ERP)



# Métaphore urbaine de l'extensions propriétaire d'un progiciel

Jakarta, ville de Java...



# Technologie progicielle les « PGI » (ERP)

- Progiciel de Gestion Intégrée  
(Enterprise Resource Planning ou ERP)
  - Résolution de l'urbanisation par l'architecture d'une macro-application
    - Externalisation de problématique d'urbanisme auprès d'un fournisseur de solutions
  - Intégration des applications communes aux entreprises ;
    - Gestion des ressources humaines
    - Gestion comptable et financière
    - Aide à la décision
    - Gestion des approvisionnements
    - Gestion de production, logistique
    - ...
  - Méta Modèle imposé par le fournisseur, solution « prête à porter »
  - Animé par un moteur de « Workflow »
- Exemples :
  - Commerciaux :SAP, BAAN
  - Libres : OFBiz, Dolibarr

# Technologies : Portails et Gestion de contenus,

- Les solutions « à client léger » (i.e. navigateur Web) sont privilégiées pour limiter les coûts de déploiement)
    - (retour logique du « terminal »)
    - Frontaux d'applications lourdes (GED)
    - Exploitation des technologies « Web 2.0 » (XMLHttpRequest, Ajax)
    - Interface utilisateur pour l'exploitation
      - Exemples de solutions de portails
        - IBM WebSphere Portal
  - Les outils de gestion de contenus (Content Management System)
    - Organisent les portails web pour disposer de contenus (multimédia)
    - Séparent la spécification des contenants
  - Les applications composites (mash-up) sont de plus en plus nombreuses et de plus en plus différentes,
    - Associations d'information, superpositions de résultats,
    - Agrégation d'informations prélevées sur d'autres sites,
  - Problématique : dans la pratique, la multiplication des sites complique les processus réels des entreprises
- Pour les CMS, les pratiques ont dévoyé les usages : par exemple des utilisateurs incompetents exploitent un CMS en lieu et place et sans qualité d'une GED...
- Attention ! la fiabilité d'un Mash-up est le produit de la fiabilité de ses sources...  
 $0.8 \times 0.5 \times 0,25 = 0,1 !$



Fin du module