Panorama des technologies et architectures

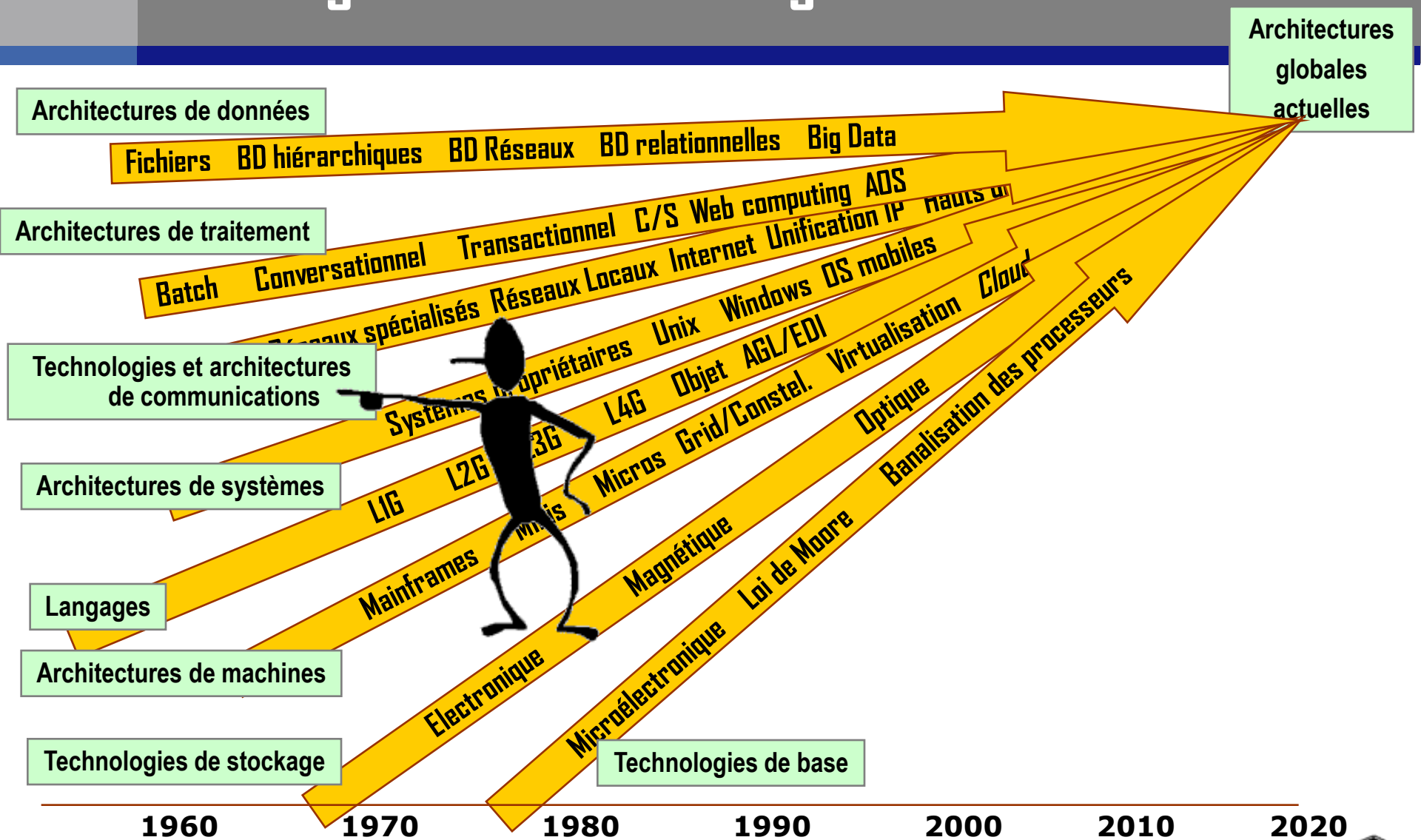
Extrait

Plan

- C01 Présentation de la formation : Architectures et technologies
- C02 Technologies de base
- C03 Capacité de programmation
- C04 Architectures de machines et de stockage
- C05 Architectures de systèmes
- C06 Architectures de réseaux
- C07 Architectures de données
- C08 Architectures de traitement
- C09 Architectures applicatives
- C10 Intégration dans une architecture globale
- C11 Architectures globales
- C12 Plates-formes de développement
- C13 Problématiques transversales : Sécurité, fiabilité, performance, évolutivité.
- C14 Bilan. Synthèse.

Architectures de réseaux

Convergence des technologies et des architectures

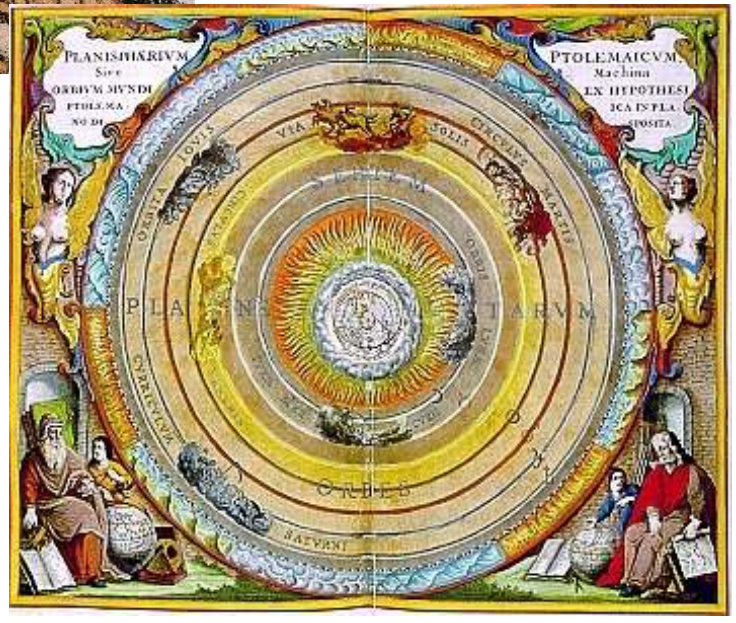


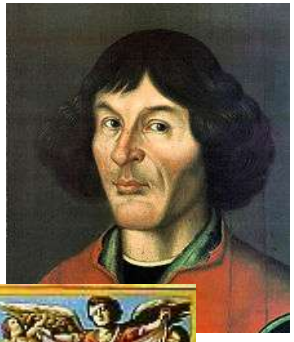
Quelques questions

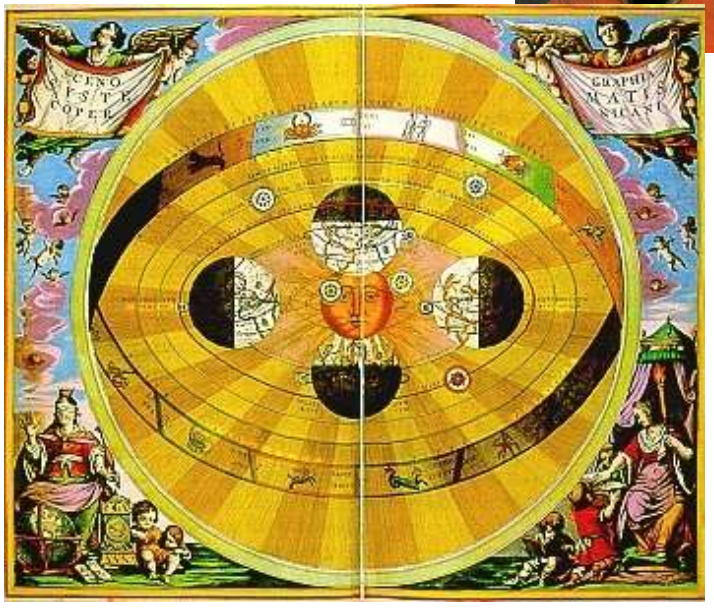
Une révolution copernicienne

- **Accéder aux ressources de l'ordinateur central** => Notion de télétraitement, de modem
- **Développer des applications transactionnelles** => Systèmes de gestion de transaction type CICS
- **Banaliser les ressources et offrir à l'utilisateur un choix de services distants** => Système de gestion de réseau type SNA
- **Administrer les réseaux** => Systèmes d'administration de réseaux type Netview
- **Partager les ressources PC** => Réseau local
- **Bâtir un ensemble cohérent** => Fédérer réseaux étendus et réseaux locaux au sein d'une architecture cohérente
- **Tirer parti des avantages des solutions Internet** => Les technlogies Internet fournissent cette cohérence autour d'IP

Une révolution copernicienne







Une révolution copernicienne

Avant-hier

Le client est lié à une machine

Une révolution copernicienne

Une révolution copernicienne

Une révolution copernicienne

Une révolution copernicienne

Une révolution copernicienne

Normalisation des échanges client-serveur (HTTP)

Normalisation des échanges serveur-serveur (AOS)

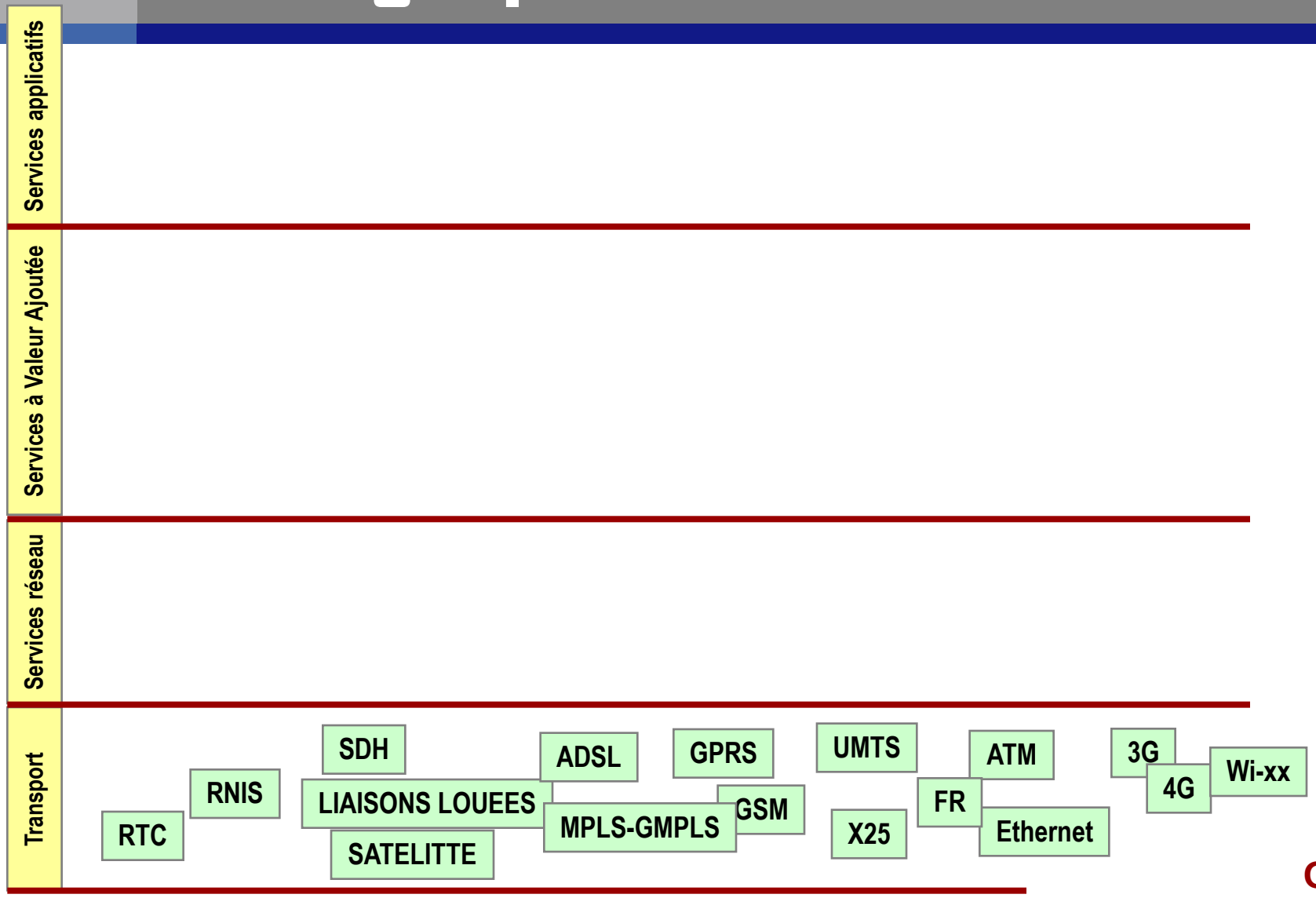
Capacité à construire de nouvelles applications parfaitement adaptées aux exigences spécifiques d'une organisation en assemblant des briques standard de services web (XML, SOAP, WSDL, UDDI)

Réseau
avec intelligence et QOS

Multiplicité des clients

Normalisation des échanges client-client (POP/SMTP/IMAP)

Cartographie des services IP



Cartographie des services IP

Services applicatifs

Services à Valeur Ajoutée

Services réseau

ACCES INTERNET

GESTION ROUTEURS

RESEAUX PRIVES VIRTUELS

CONNECTIVITE IP
(INTRANET / EXTRANET)

VOIP

INTERCONNEXION RESEAUX LOCAUX

GATEWAYS

IAP

Transport

SDH

ADSL

GPRS

UMTS

ATM

3G

RNIS

4G

Wi-xx

LIAISONS LOUEES

GSM

FR

RTC

MPLS-GMPLS

X25

Ethernet

SATELITTE

Opérateur

Cartographie des services IP

Services applicatifs

Services à Valeur Ajoutée

GROUPWARE

MESSAGERIE

PACKAGES SECURITE

VISIO ET AUDIO CONFERENCE

CALL CENTER

EDI

SERVICES INTERNET A VALEUR AJOUTEE

PAIEMENT SECURISE

CTI

HEBERGEMENT

ADMINISTRATION DE RESEAU A DISTANCE

ISP

Services réseau

ACCES INTERNET

GESTION ROUTEURS

RESEAUX PRIVES VIRTUELS

CONNECTIVITE IP
(INTRANET / EXTRANET)

VOIP

INTERCONNEXION RESEAUX LOCAUX

GATEWAYS

IAP

Transport

SDH

ADSL

GPRS

UMTS

ATM

3G

RNIS

LIAISONS LOUEES

GSM

FR

4G

Wi-xx

RTC

MPLS-GMPLS

X25

Ethernet

SATELITTE

Opérateur

Cartographie des services IP

Services applicatifs

ERP

SERVICES FINANCIERS

E-LEARNING ET TRAVAIL COLLABORATIF

GRC/CRM

SOLUTIONS METIERS

BUREAUTIQUE EN LIGNE

GESTION PLACES DE MARCHE

GESTION DES WORKFLOW

GESTION SUPPLY CHAIN

COMMERCE ELECTRONIQUE

ASP

Services à Valeur Ajoutée

GROUPWARE

PACKAGES SECURITE

MESSAGERIE

VISIO ET AUDIO CONFERENCE

CALL CENTER

EDI

SERVICES INTERNET A VALEUR AJOUTEE

PAIEMENT SECURISE

CTI

HEBERGEMENT

ADMINISTRATION DE RESEAU A DISTANCE

ISP

Services réseau

ACCES INTERNET

GESTION ROUTEURS

RESEAUX PRIVES VIRTUELS

CONNECTIVITE IP
(INTRANET / EXTRANET)

VOIP

INTERCONNEXION RESEAUX LOCAUX

GATEWAYS

IAP

Transport

SDH

ADSL

GPRS

UMTS

ATM

3G

RNIS

LIAISONS LOUEES

GSM

FR

4G

Wi-xx

RTC

MPLS-GMPLS

X25

Ethernet

SATELITTE

Opérateur

Cartographie des services IP

Services applicatifs

ERP

SERVICES FINANC

E-LEARNING ET TRAVAIL COLLABORATIF

GRC/CRM

SOLUTIONS METIERS

GESTION PLACES DE MARCHE

GESTION DES WORKFLOW

COMMERCE ELECTRONIQUE

Fournisseur d'applications hébergées (*ASP*)

ASP

Services à Valeur Ajoutée

GROUPWARE

MESSAGERIE

VISIO ET AU

ENTER

EDI

SERVICES INTERNET A VALEUR AJOUTE

PAI

TI

HEBERGEMENT

ADMINISTRA

NCE

Fournisseur de services Internet (*ISP*)

ISP

Services réseau

ACCES INTERNET

GESTION ROUTEURS

RES

VOIP

CONNECTIVITE IP (INTRANET / EXTRANET)

INTERCONNEXION

GATEWAYS

Fournisseur d'accès Internet (*IAP*)

IAP

Transport

SDH

ADSL

GPRS

UMTS

ATM

3G

RNIS

LIAISONS LOUEES

GSM

FR

4G

Wi-xx

RTC

MPLS-GMPLS

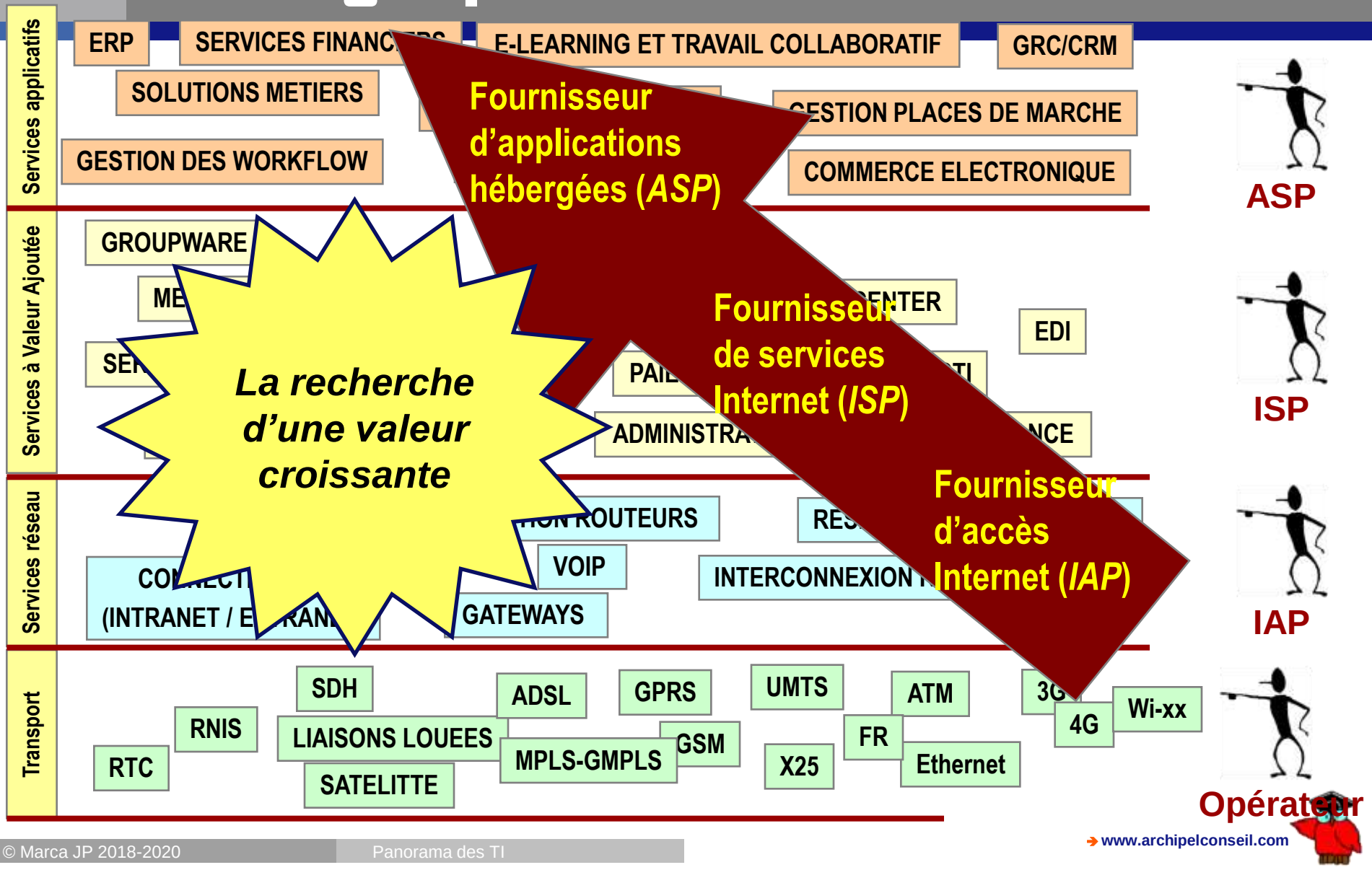
X25

Ethernet

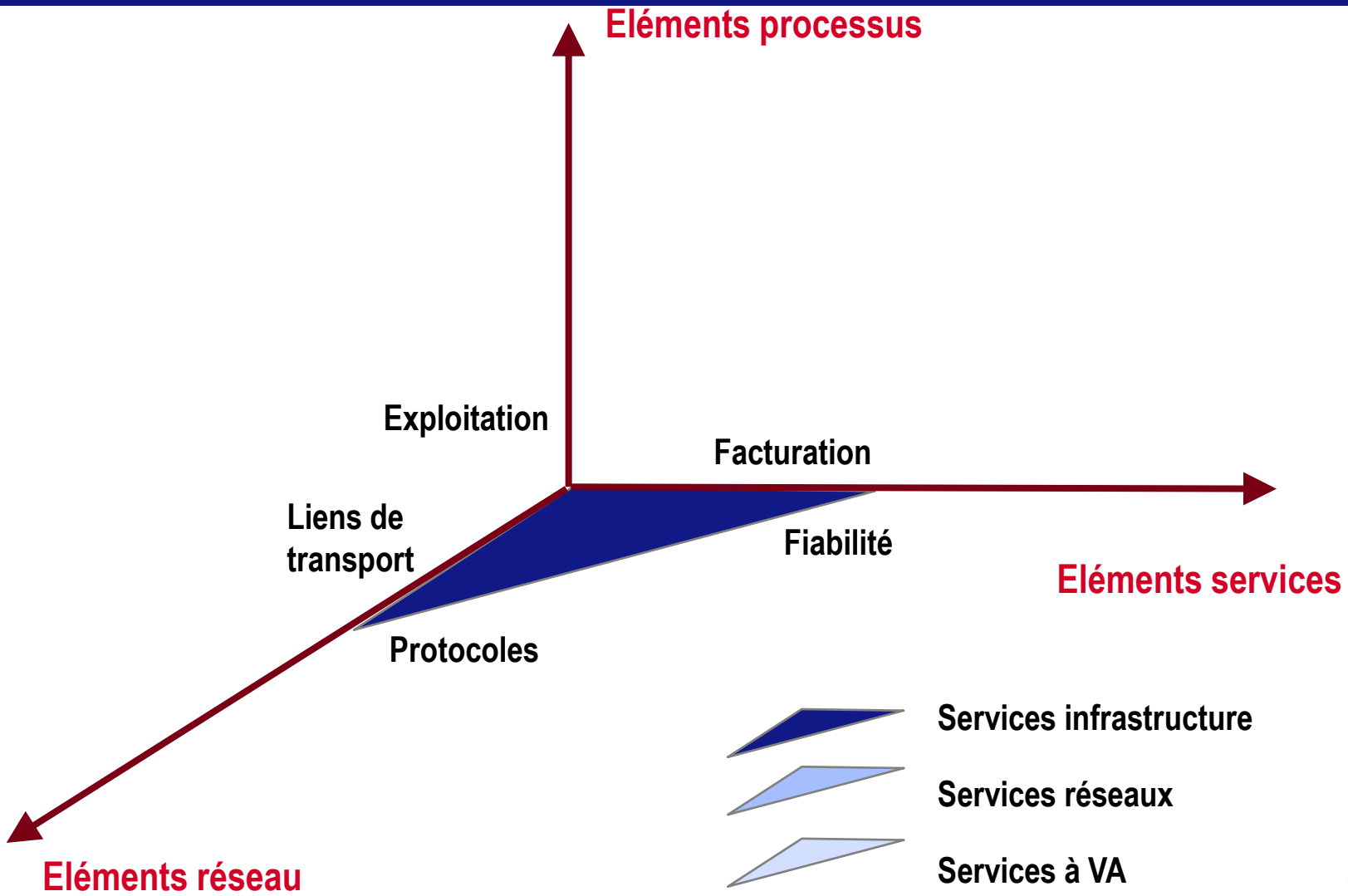
SATELITTE

Opérateur

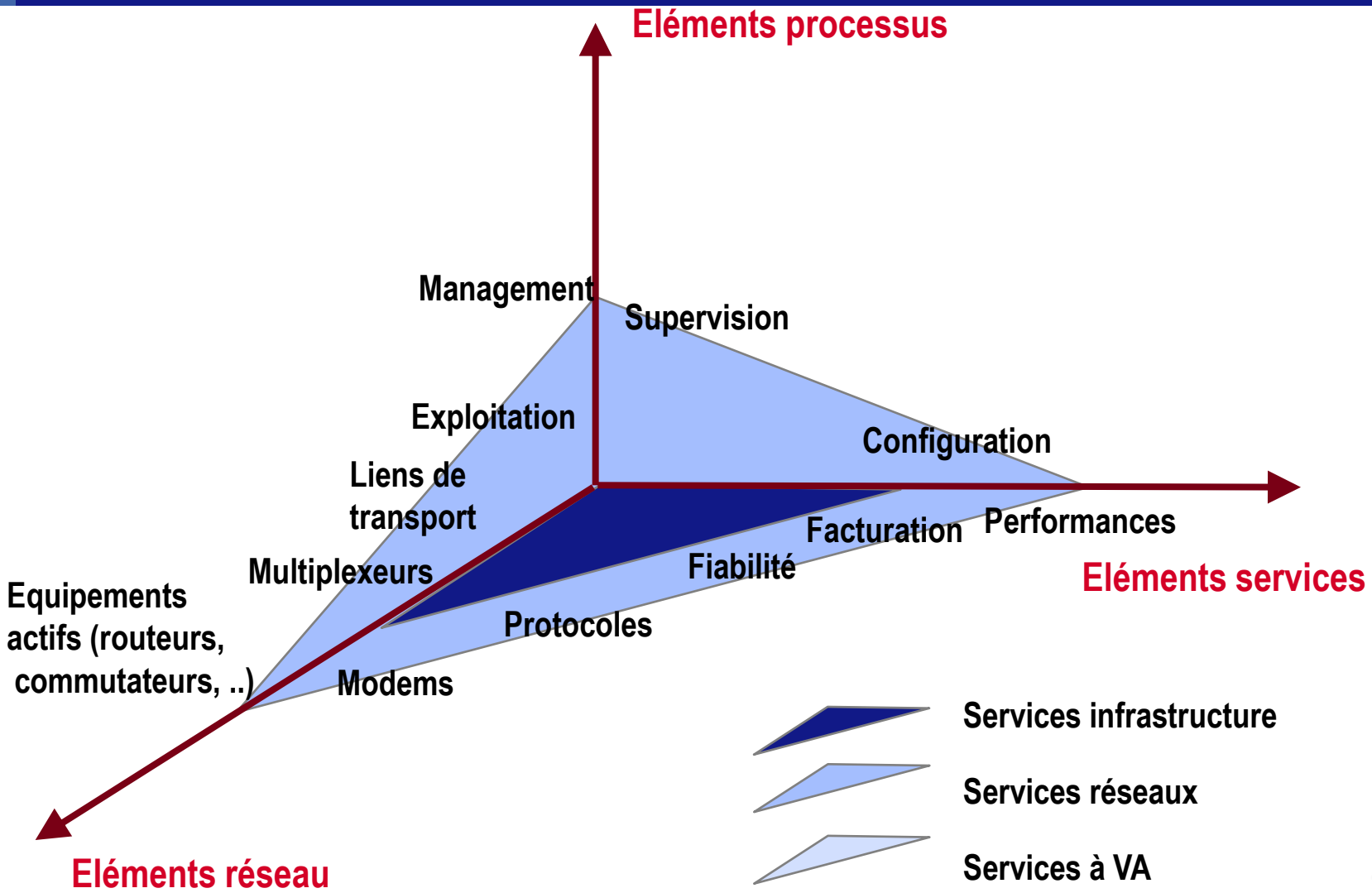
Cartographie des services IP



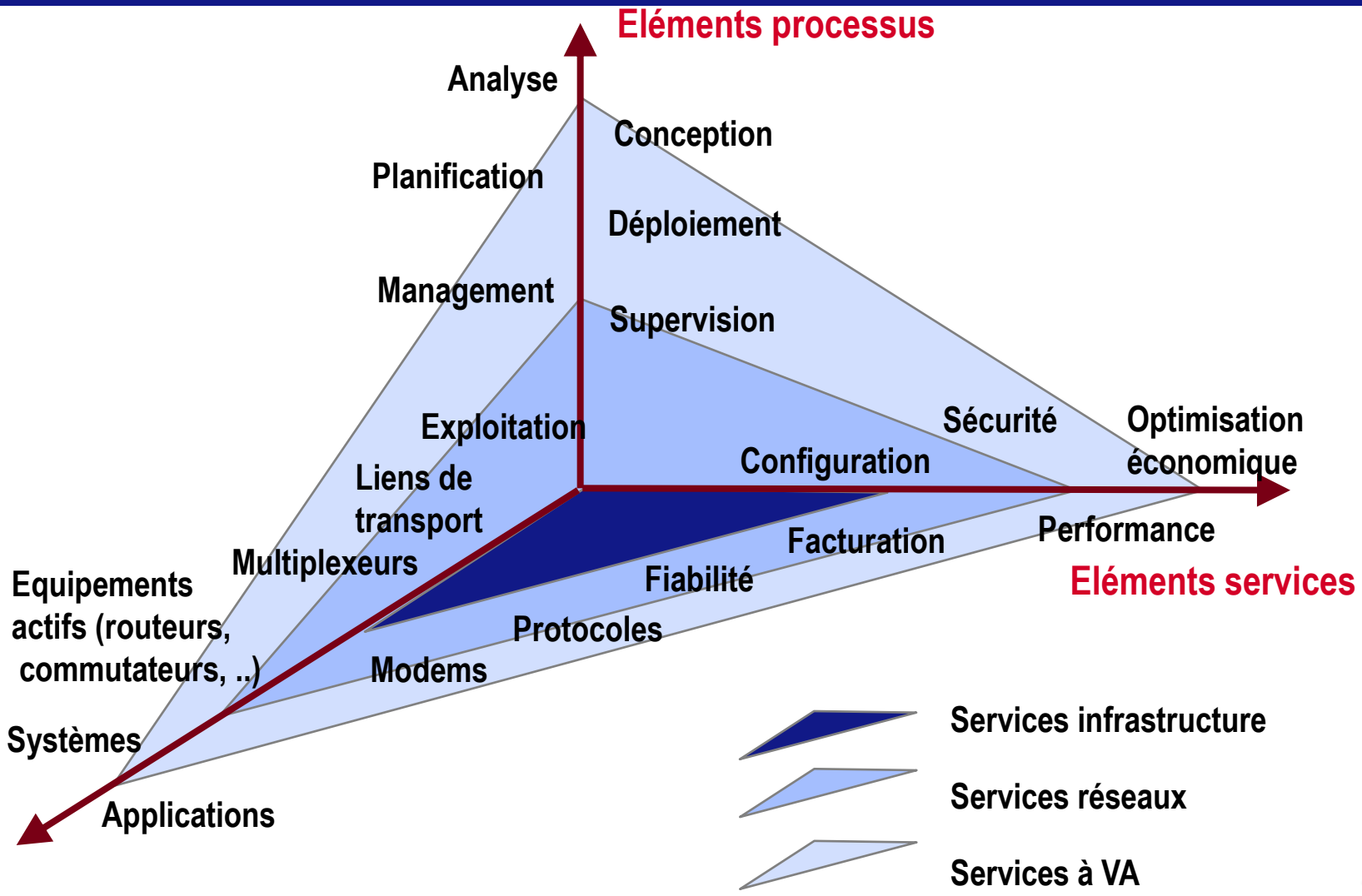
Une offre multidimensionnelle



Une offre multidimensionnelle

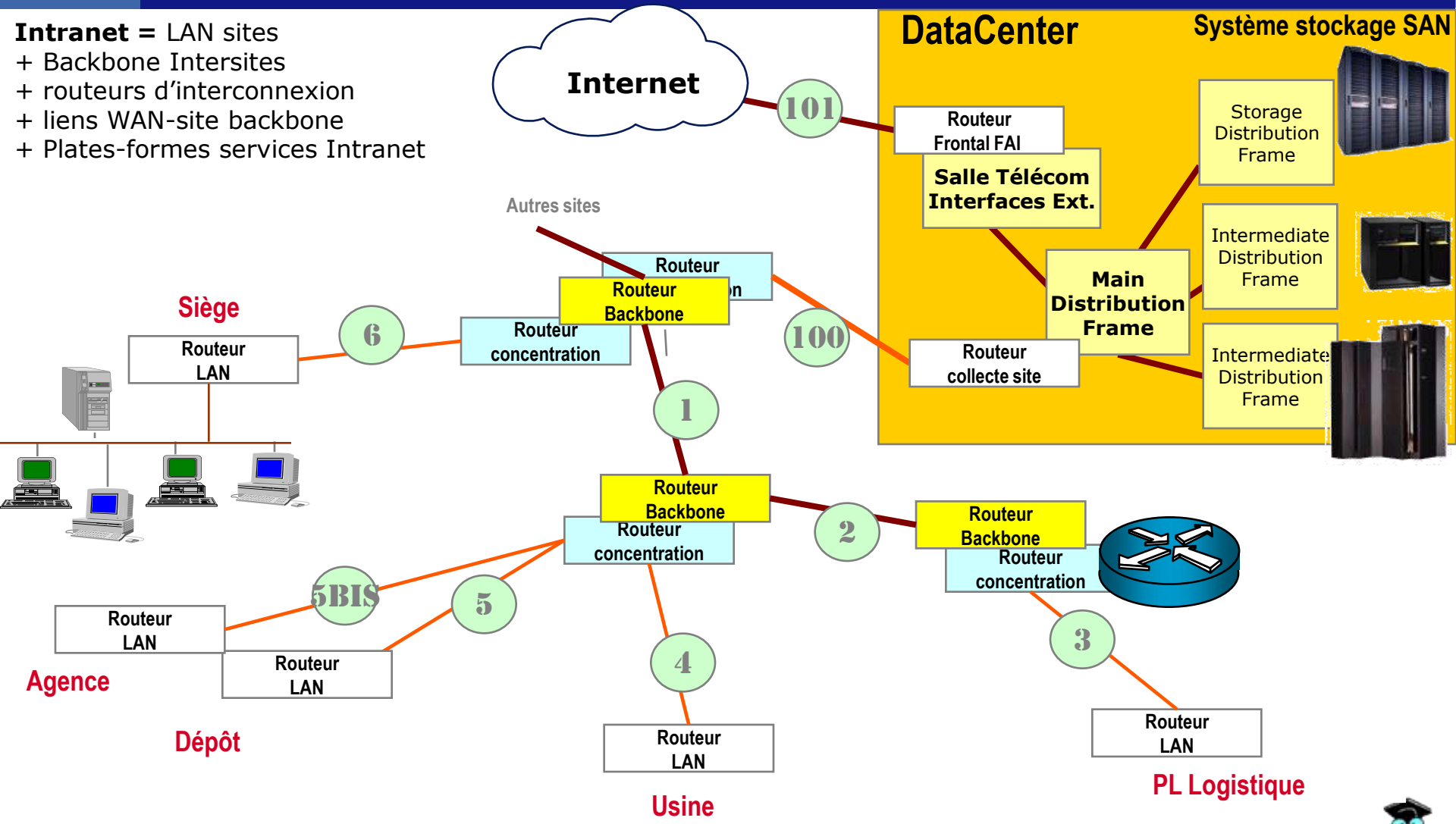


Une offre multidimensionnelle

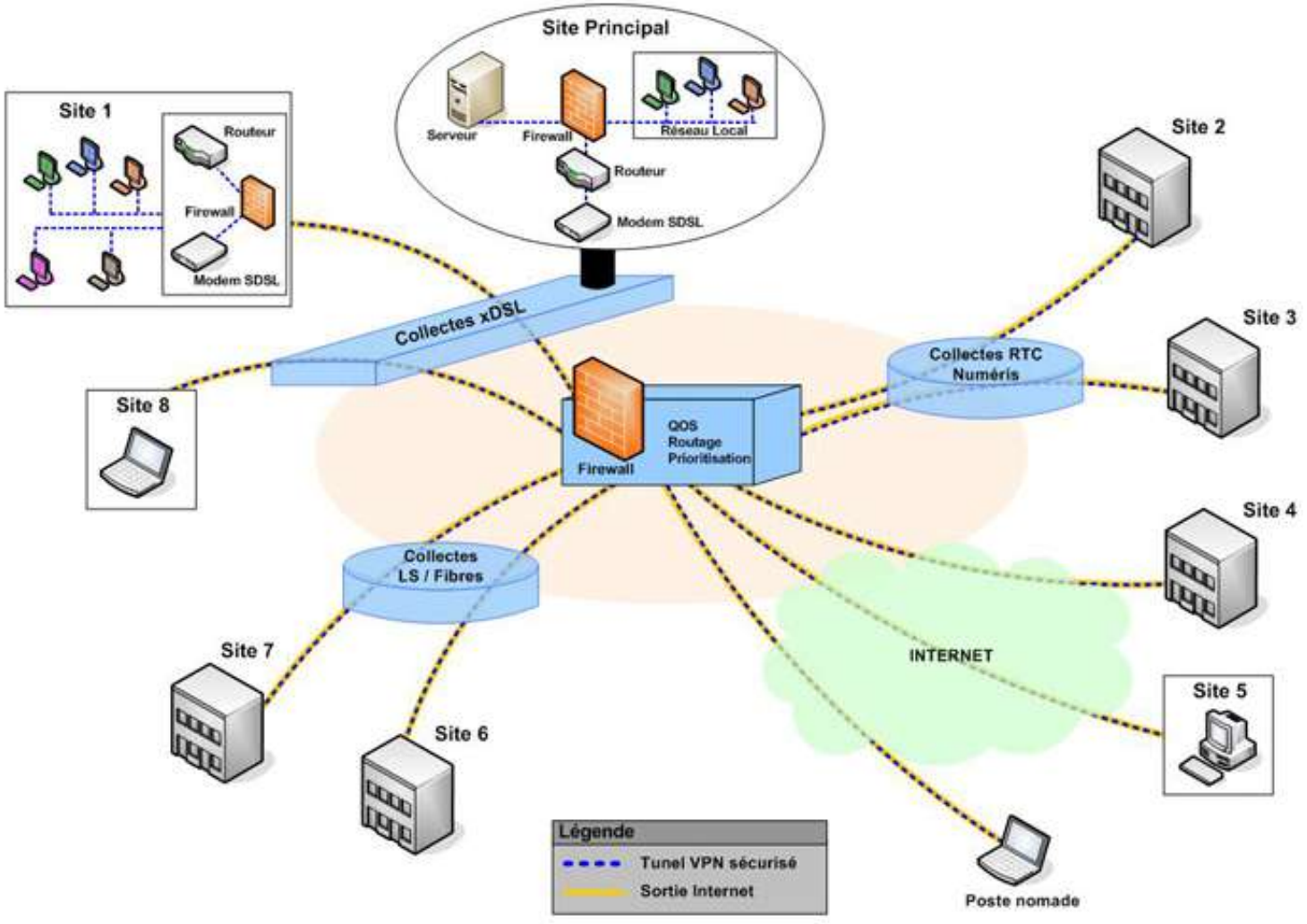


Exemple de réseau

Intranet = LAN sites
+ Backbone Intersites
+ routeurs d'interconnexion
+ liens WAN-site backbone
+ Plates-formes services Intranet



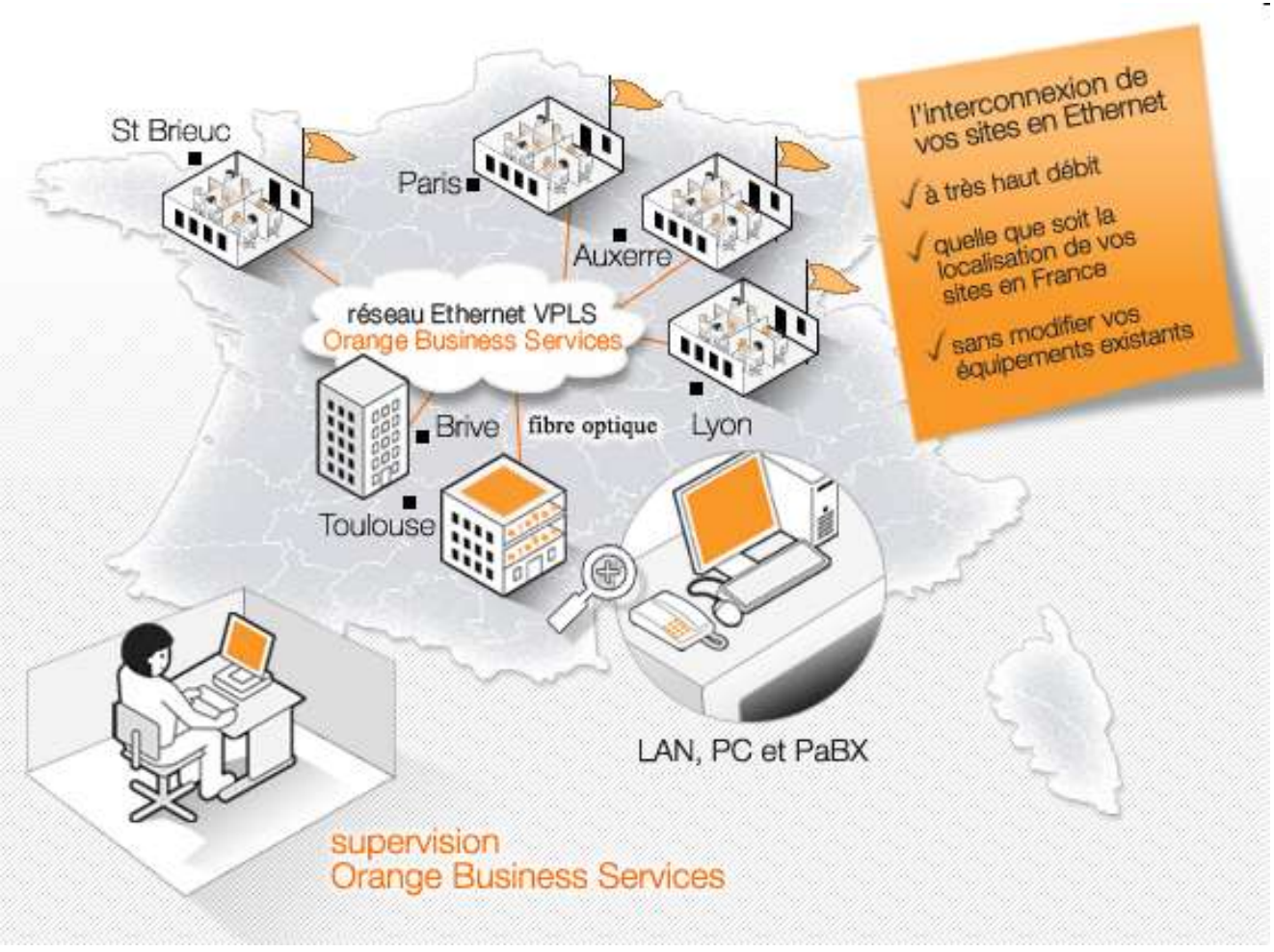
Concept VPN


Site Principal
Serveur
Firewall
Réseau Local
Routeur
Modem SDSL
Site 1
Routeur
Firewall
Modem SDSL
Collectes xDSL
Site 2
Site 3
Collectes RTC
Numéris
Site 8
Firewall
QOS
Routage
Prioritisation
Site 4
Collectes
LS / Fibres
INTERNET
Site 7
Site 6
Site 5
Légende
Tunel VPN sécurisé
Sortie Internet
Poste nomade

Concept VPN

- Le **réseau privé virtuel** (RPV ou VPN = *Virtual Private Network*) correspond à une interconnexion de réseaux locaux via une infrastructure partagée avec d'autres organismes.
- Cette infrastructure peut être Internet. La connexion s'opère au moyen de tunnels de type IPSec.
- Elle peut être un réseau privé mis en place par un opérateur pour offrir des services de VPN aux entreprises, en s'engageant non seulement sur les connexions, mais aussi sur les niveaux de performance et de sécurité.
- Le VPN doit fonctionner comme un réseau privé, tout en bénéficiant des avantages de la mutualisation des ressources.
- VPN Frame Relay (niv. 2), VPN IP (niv. 3), MPLS (niv. 2.5), VPN Ethernet (niv. 2), VPLS Ethernet multipoint-à-multipoint pour l'interconnexion LAN.

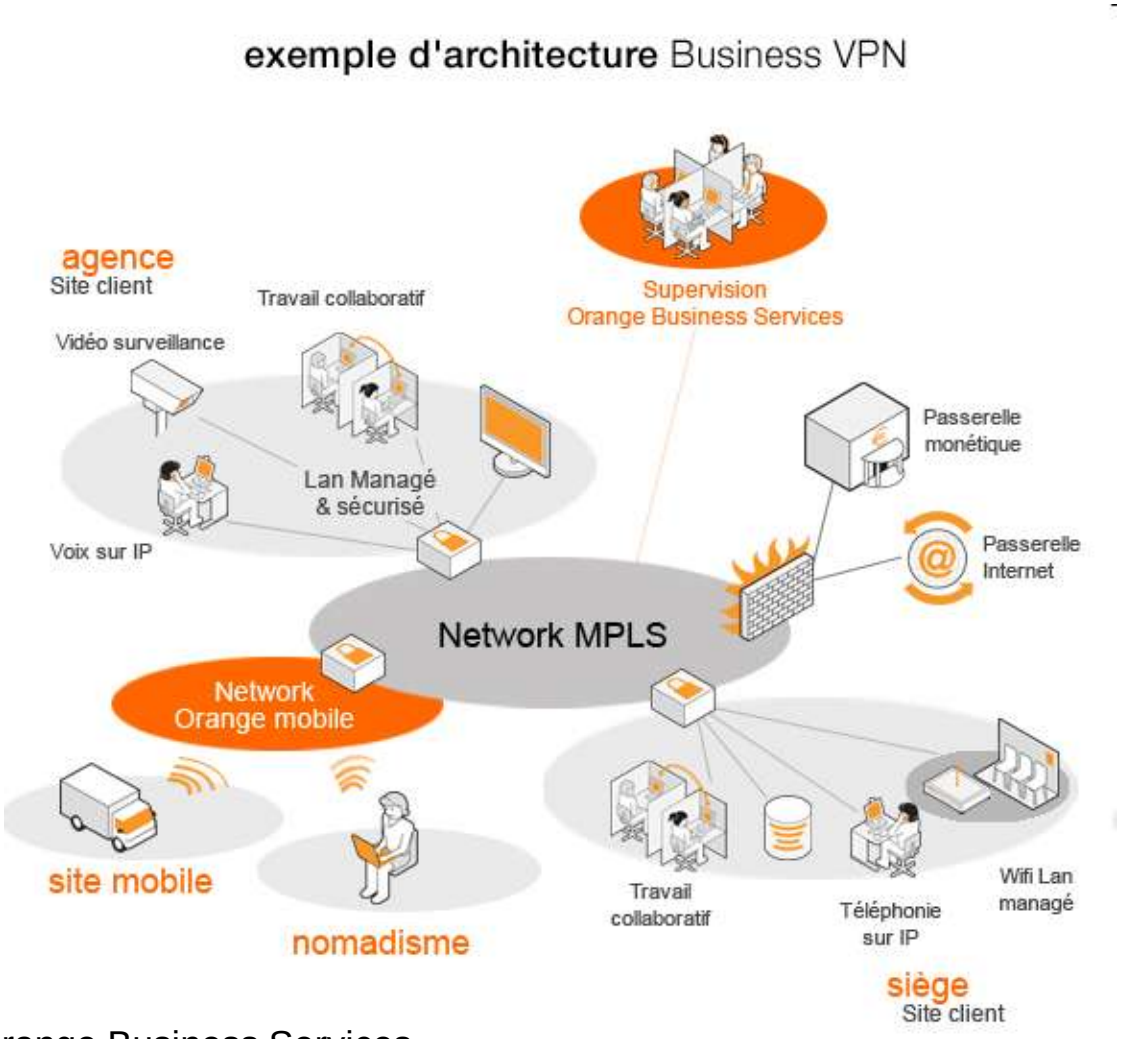
Les gagnants : IP et Ethernet



Service **VPN Ethernet** avec un coeur de réseau **VPLS** (Virtual private LAN service), permettant des fonctionnalités avancées (any-to-any, Classes de Service, transparence aux VLAN) comparables à celles proposées sur les VPN IP, et des débits allant jusqu'au Gigabit

Source : Orange Business Services

Les gagnants : IP et Ethernet



Disponibles en France métropolitaine et dans les DOM, les services Business VPN reposent sur un réseau **IP VPN MPLS** (Multi-Protocol Label Switching) dédié aux entreprises et hautement sécurisé.

Source : Orange Business Services

Le cloud vu par ...

- *par Amazon*
- par *Google*
- *par Salesforce.com*
- *par Microsoft*
- *par IBM*

Une prédiction enfin réalisée ?

« I think there is a world market for maybe five computers. »

NY 1943

Thomas Watson, Fondateur d'IBM

Cloud privé vs cloud partagé

- Cloud privé :
  - Modernisation de l'infrastructure (fluides, énergie, procédures, plan de reprise d'activité, …);
  - Optimisation du taux d'utilisation des machines;
  - ROI en conséquence.
- Appel à prestations externes (IaaS) sur cloud partagé :
  - Réduction des coûts d'infrastructure de 20 à 40% du fait de la mutualisation;
  - Niveau de SLA, assistance à la migration, persistance des machines virtuelles, etc;
  - Coûts liés au changement;
  - ROI en conséquence;
- Les gains ne sont pas systématiques;
- *Cloudbursting* : Faire temporairement appel au cloud en cas de besoin ponctuel (Surcharge, opération marketing, ..) .

Référentiel cloud

- Livre blanc Syntec :
- http://www.syntec-informatique.fr/actualites/liste-actualites/publication-du-livre-blanc-cloud-computing-de-syntec-informatique
- Position du Cigref :
- http://www.cigref.fr/cigref_publications/RapportsContainer/Parus2010/Position_CIGREF_sur_le_Cloud_computing_Septembre_2010_CIGREF.pdf
- Le rapport de la Commission Européenne
- http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf
- Le blog du cloud
- http://www.cloud-buzz.com/

Réseaux du futur

- Les concepts clés des réseaux du futur sont :
  - l'**ubiquité** de services supportés par une grande diversité de technologies d'accès (fixes et mobiles avec une capacité toujours croissante);
  - la **convergence** des réseaux et des services, permettant la conception de services évolués et la diminution du coût total;
  - l'**interopérabilité** de systèmes hétérogènes permettant cette convergence ;
  - la **mobilité** généralisée des usagers et des services et l'urbanisation des réseaux à trois grandes échelles:
    - les grands réseaux d'infrastructure,
    - les réseaux d'extrémité plus dynamiques et auto-configurés (réseaux mesh, réseaux ad-hoc, etc),
    - les réseaux de capteurs, contrôleurs et RFID de nouvelle génération, et autres objets miniaturisés et intelligents.

Objets en réseau

- **Internet des objets**, Internet du monde physique, M2M (Machine-to-Machine), « Intelligence ambiante »;
- Etiquettes RFID communiquant à distance des informations sur les objets qui en sont dotés sans besoin de les manipuler;
- Les étiquettes recèlent un identifiant, supposé unique, du produit qu'elles marquent (exemple EPC diapo suivante);
- Pour éviter les collisions : 2 types de protocoles, le parcours d'arborescence et l'accès aléatoire;
- Etiquettes passives (activées par un champ électromagnétique extérieur engendré par le lecteur) et étiquettes actives disposant d'une source d'énergie propre.
- Problématiques d'usage : implants humains, accumulation des émissions électromagnétiques.
- En 2020, le nombre d'objets connectés pourrait atteindre 80 milliards dans le monde (source Idate).
- Selon Cisco, l'enjeu économique s'élèverait à pas moins de 14.400 milliards de dollars au cours de la prochaine décennie !

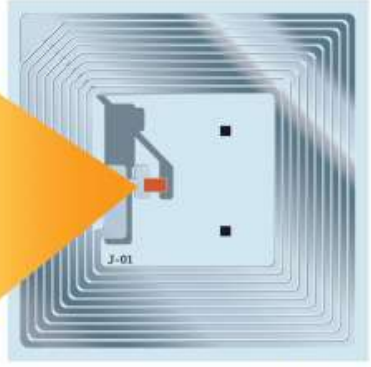
Objets en réseau

Un numéro unique dans un tag RFID pour identifier chaque objet.

The EPC Number disected (96 bits version)

21.203D2A9.16E8B8.719BAE03C

- Header 8 bits
- EPC Manager 28 bits (> 268 Million)
- Object Class 24 bits (> 16 Million)
- Serial Number 36 bits (< 68 Billion)



Spécifications EPC : 96 bits

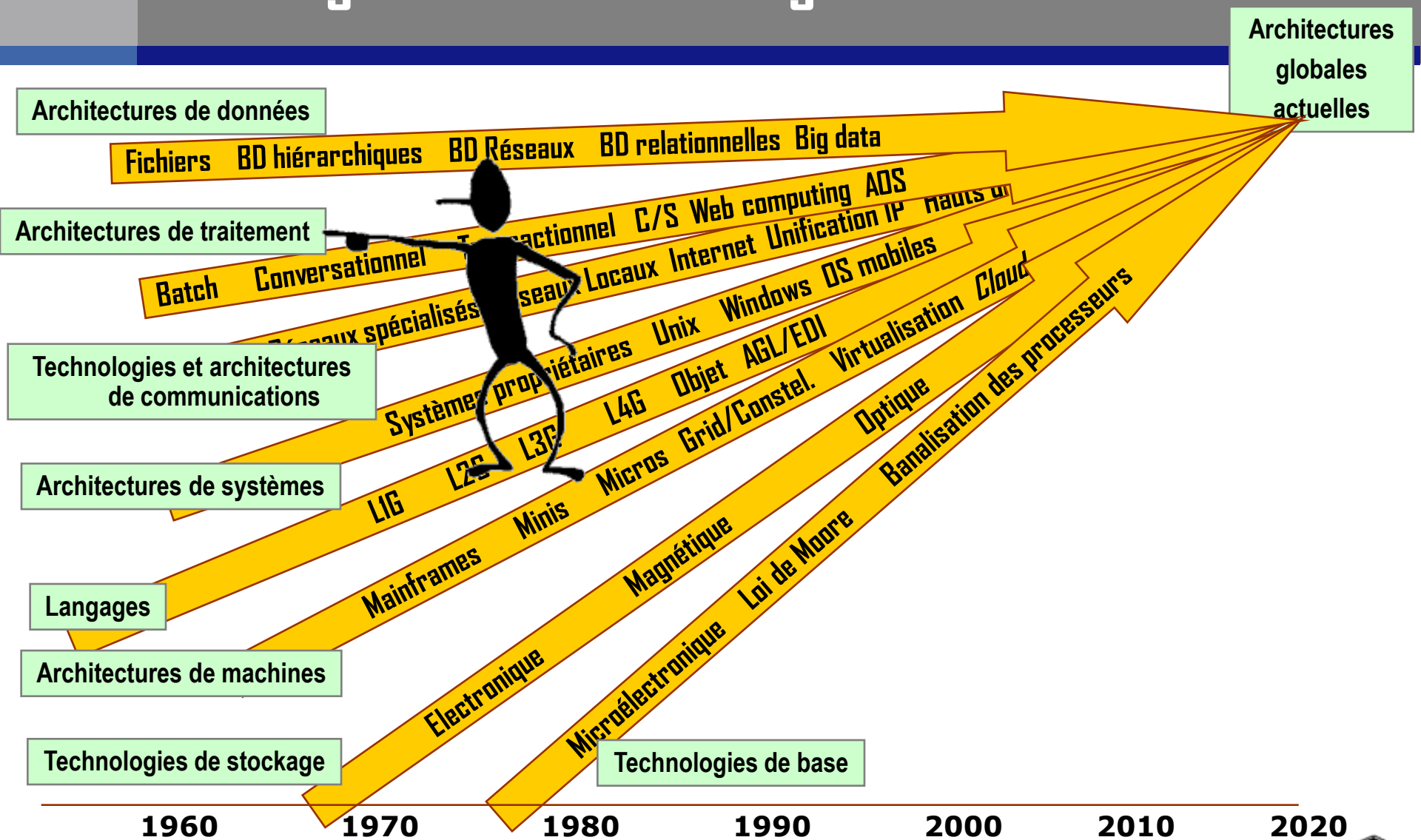
Spécifications EPC : le code EPC peut être créé, soit à partir d'un GTIN + un numéro de série, soit à partir d'un SSCC.

Plan

- C01 Présentation de la formation : Architectures et technologies
- C02 Technologies de base
- C03 Capacité de programmation
- C04 Architectures de machines et de stockage
- C05 Architectures de systèmes
- C06 Architectures de réseaux
- C07 Architectures de données
- C08 Architectures de traitement
- C09 Architectures applicatives
- C10 Intégration dans une architecture globale
- C11 Architectures globales
- C12 Plates-formes de développement
- C13 Problématiques transversales : Sécurité, fiabilité, performance, évolutivité.
- C14 Bilan. Synthèse.

Architectures de traitement

Convergence des technologies et des architectures

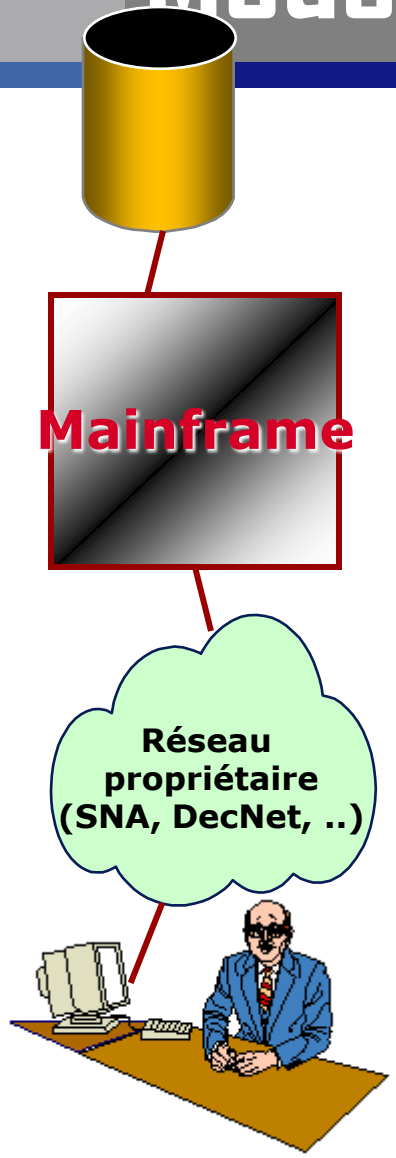


Quelques questions

Concept de transaction

- Une **transaction** est un événement au cours duquel une base de données passe d'un état 1 à un état 2, à la suite d'une opération initiée par l'utilisateur sur son poste de travail.
- Ce peut être par exemple une réservation de place, un mouvement de stock, un achat, un paiement, l'affectation d'un collaborateur à un autre poste, la mise à jour d'un planning de fabrication, etc.
- La transaction est la base des opérations de gestion.
- Elle doit être menée à son terme indépendamment de l'architecture informatique mise en œuvre.
- Mode Transactionnel vs Mode par lots (***batch***)
- Part importante du *batch* dans les travaux quotidiens

Mode transactionnel de base



Mode transactionnel classique : L'utilisateur utilise son terminal pour collecter les données et recevoir immédiatement les données du traitement;

- Poste de travail passif;
- Protocoles propriétaires;
- Données et traitement au niveau du serveur.
- Respect des propriétés ACID : Atomicity, Consistency, Isolation, Durability)

Une exigence à préserver

- Une exigence à préserver indépendamment de l'architecture choisie :
- **AC**ID
- **Atomique** : la suite d'opérations est indivisible. En cas d'échec en cours d'une des opérations élémentaires, la suite d'opérations déjà réussies doit être complétement annulée (*Rollback*) quel que soit son nombre.
- **Cohérente** : Le contenu de la base de données à la fin de la transaction doit être cohérent, sans pour autant que chaque opération élémentaire durant la transaction donne un contenu cohérent. Un contenu final incohérent doit entraîner l'échec et l'annulation de toutes les opérations de la transaction.

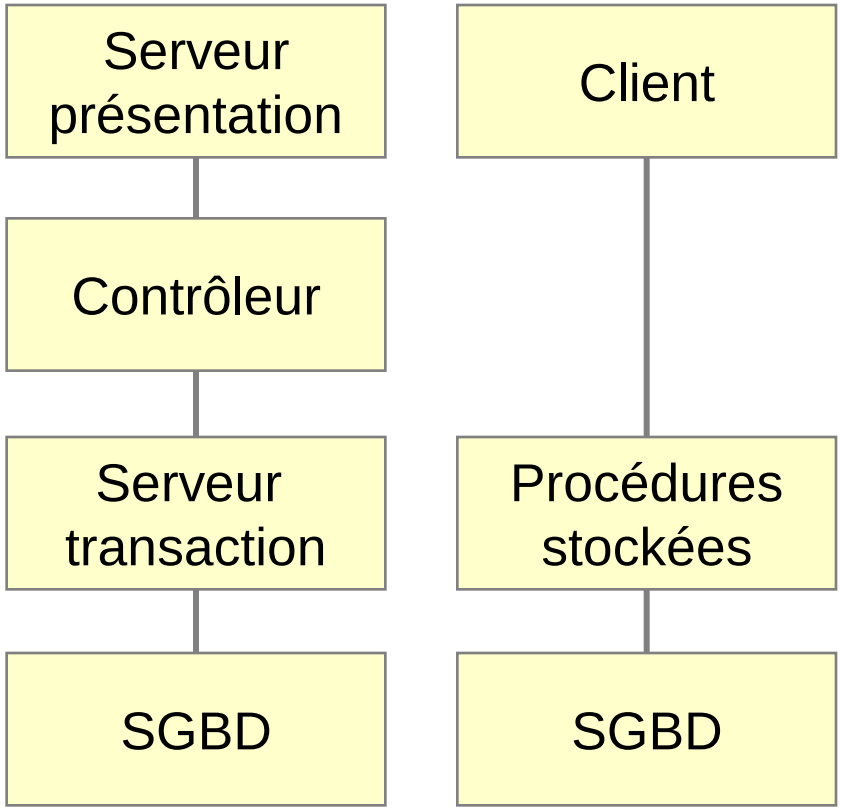
Une exigence à préserver

- AC**ID**
- **Isolée** : Lorsque deux transactions A et B sont exécutées en même temps, les modifications effectuées par A ne sont ni visibles par B, ni modifiables par B tant que la transaction A n'est pas terminée et validée (*Commit*).
- **Durable** : D'un point de vue technique, une transaction terminée ne peut pas être remise en cause, annulée ou recouverte. Lorsque deux transactions sont exécutées en même temps, le résultat de la première transaction ne pourra pas être recouvert par la deuxième. Toute tentative de recouvrement entraînera l'annulation des opérations de la transaction fautive.

Moniteur transactionnel

- Le **moniteur transactionnel** est un middleware qui surveille toutes les requêtes des clients et appelle les fonctions correspondantes sur le serveur.
- Ses principales opérations sont les suivantes :
  - Gérer le "*polling*" sur les lignes (Y-a-t-il une requête?),
  - Recevoir et analyser une requête du client par l'interface réseau;
  - Codifier et recodifier les requêtes;
  - Assurer la sécurité en authentifiant le client et la requête;
  - Lancer un processus pour satisfaire la requête demandée par le client;
  - Assurer la prise en compte simultanée de plusieurs transactions dans la même partition de mémoire;
  - Assurer les propriétés ACID, synchroniser, gérer les verrous, gérer les priorités, s'assurer du bon achèvement des tâches;
  - Recevoir la réponse du serveur et renvoyer la réponse au client;
- *CICS* (*IBM*), *Encina* (ex *Transarc*, repris par *IBM*) et *Tuxedo* (*BEA Systems*, repris par *Oracle*) sont des exemples de moniteurs transactionnels.

SGBD vs Moniteur transactionnel



- Avantages moniteur
  - Routage dans grandes applications
  - Langages et environnement plus riches
  - Serveurs bases de données répartis hétérogènes
  - Communication client-serveur
  - Environnement de gestion système
  - ➔ Transactionnel lourd
- Avantages SGBD
  - Simplicité et prix
  - Performances pour petites et moyennes applications
  - ➔ Transactionnel léger

L'héritage « Legacy »

Base de données (*Oracle*, *DB/2*, ..)
Mais aussi *DBMS*, *IMS*, *DL/1,VSAM*

Programme COBOL
(200 milliards de lignes de code en COBOL)

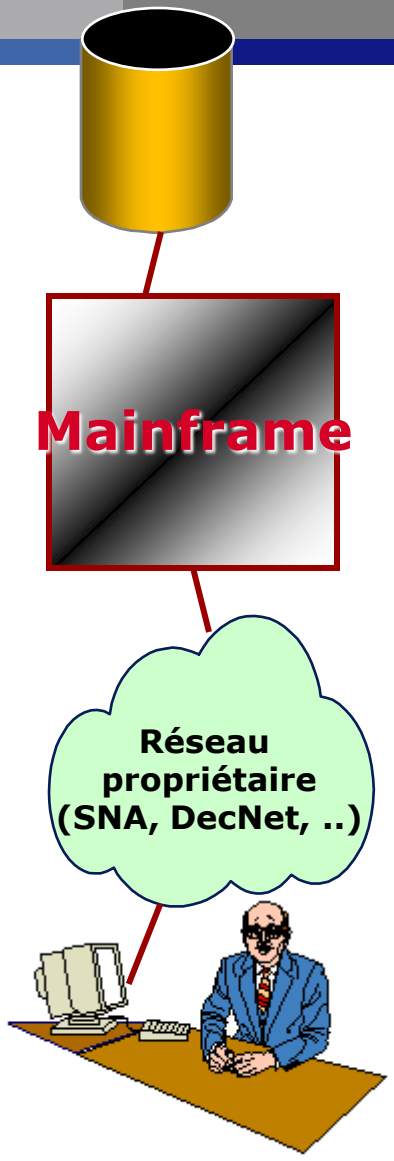
Générateurs de code type *Pacbase*

Moniteur transactionnel *CICS*

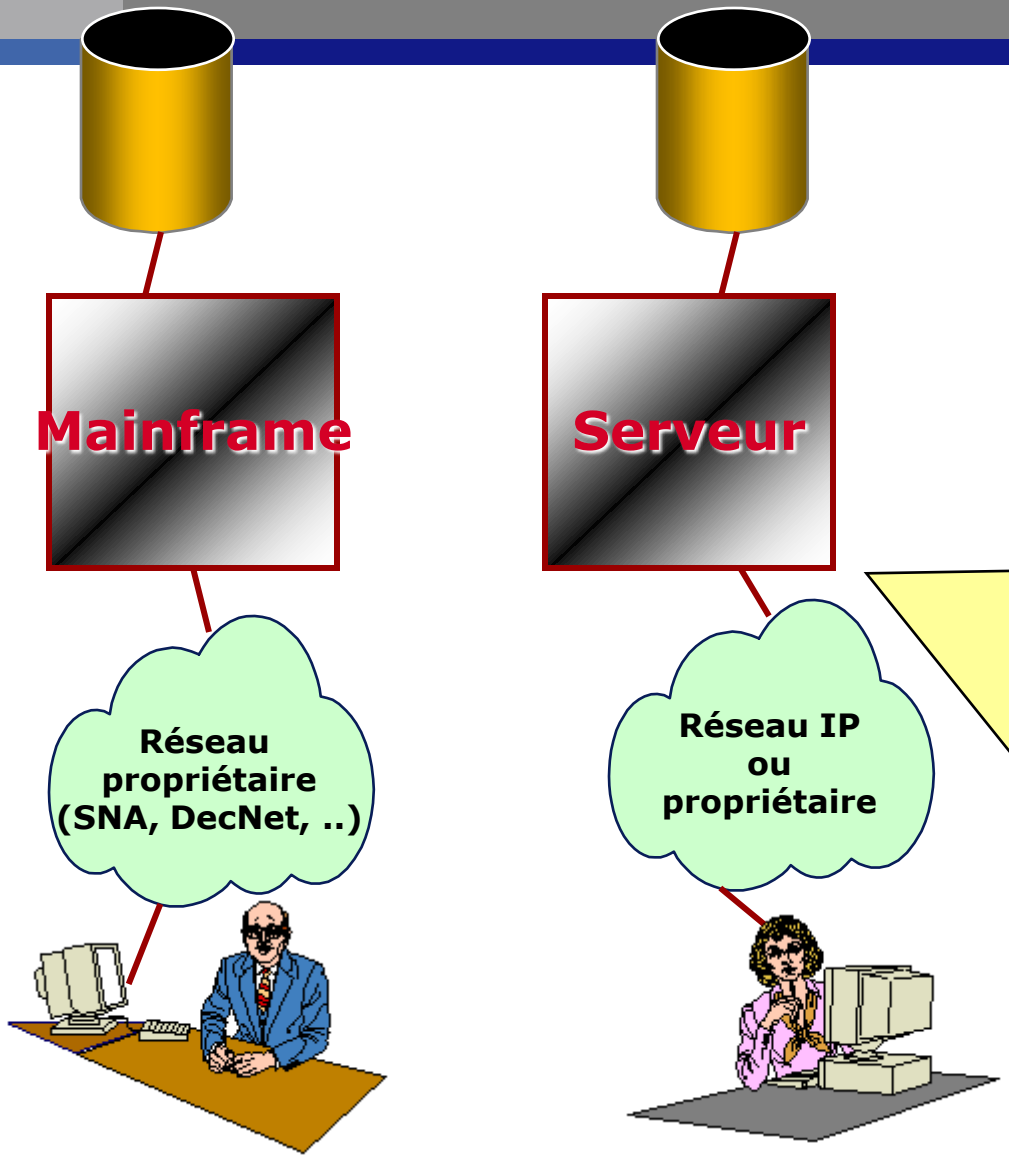
Interface utilisateur
GUI, mais aussi 3270

Le premier *Middleware*

Mode « Client/Serveur »



Mode « Client/Serveur »



Mode client/serveur

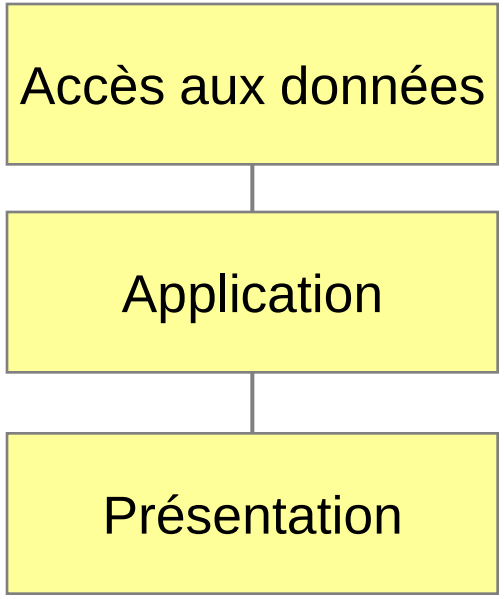
- Poste de travail « intelligent » type PC
- Réseau propriétaire ou IP
- Données et traitement partagés entre poste de travail et serveur
- Nécessité d'une couche logicielle « client » sur chaque poste de travail

Avantages et inconvénients du mode « C/S »

- Avantages :
  - Possibilité de développer des IHM (Interfaces Hommes Machine) de grande qualité.
- Inconvénients :
  - Une couche client « lourde » (*fat client*) sur le poste de travail;
  - Lourdeur du déploiement;
  - Difficulté pour tenir à jour les clients sur un grand nombre de postes.

Architecture « Three-Tier »

- Couche « Présentation » correspond à la partie de l'application visible et interactive avec les utilisateurs. C'est l'Interface Homme Machine.
- Couche » Application » gestion/métier. Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la logique et qui décrit les opérations que l'application opère sur les données, en fonction des requêtes par les utilisateurs effectuées au travers de la couche « Présentation ».
- Couche « Accès aux données » consiste en la partie gérant l'accès aux données.



Architecture « Three-Tier »

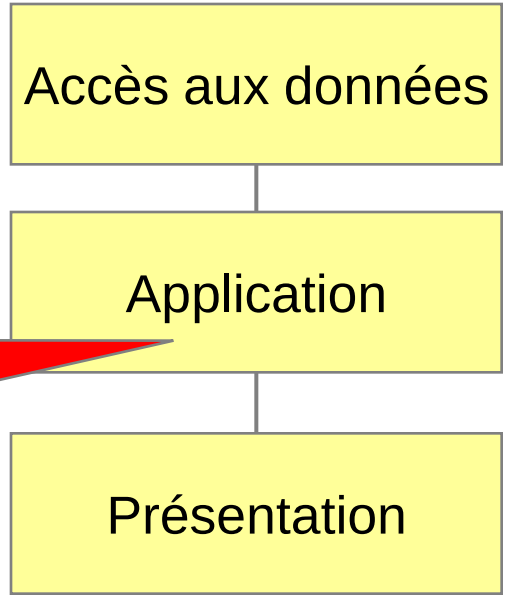
correspond
visible et
teurs.
Machine.

spond à la

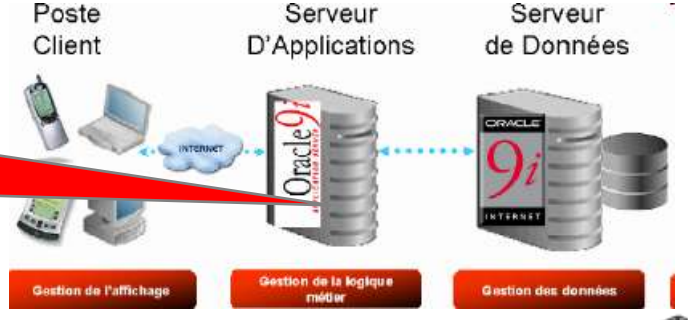
données,
par les
utilisateurs effectuées au travers de
la couche « Présentation ».

- C »

L'idée qu'une application n'est plus un bloc monolithique, mais un ensemble de composants



L'émergence du concept de serveur d'applications



Intérêt de l'éclatement

- Division de l'application en modules indépendants
  - Plus grande disponibilité
  - Evolutivité plus facile
  - Maintenance plus aisée

« 3-Tier »

RPC (Remote Procedure Call)
Protocole permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications.
Ce protocole est utilisé dans le modèle client-serveur et permet de gérer les différents messages entre ces entités.

RPC / MOM

MONIT… TRANSA…NEL

Application

Application

…on

SGBD

MOM (**Message Oriented Middleware**
Appelés aussi parfois *Brokers* de messages, le MOM est un système qui permet de faire communiquer deux applicatifs et ce de manière asynchrone.
Le MOM stocke le message (*store*) et le route (*forward*) par la suite à son destinataire lorsque celui-ci le demande.

1er niveau IHM

2e Logiqu…

Le moniteur transactionnel multiplexe les utilisateurs sur des processus serveur

« 3-Tier »

Architecture CORBA (95-2000)

Le cœur de Corba est le bus d'objets ORB (*Object Request Broker*) qui gère les transferts entre les clients et les serveurs.

ORB

IIOP de Corba ou COM+

MONI… TRANS… …NNEL

ORB

Corba

COM

Moniteur transactionnel

Moniteur

SGBD

1er niveau IHM

2e ni… Serveurs

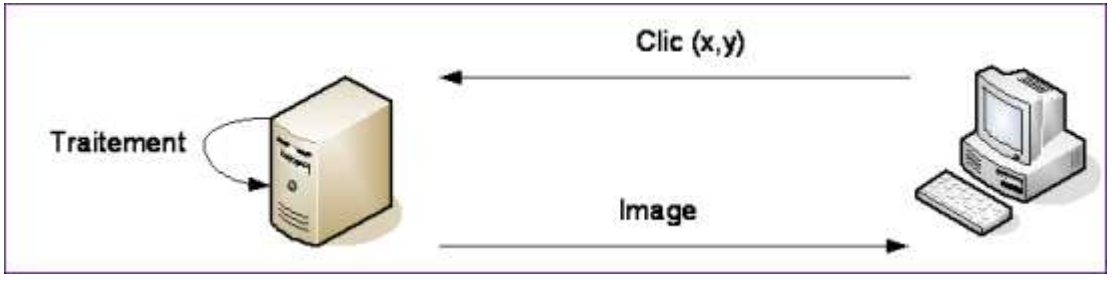
COM+ de Microsoft

Dans le cadre de DNA (*Distributed Network Architecture*) utilisation d'un modèle à objet pour offrir une vue unique d'un système distribué :

- COM+ pour le middleware
- Active X pour les applications Internet

Remédier aux défauts du client lourd

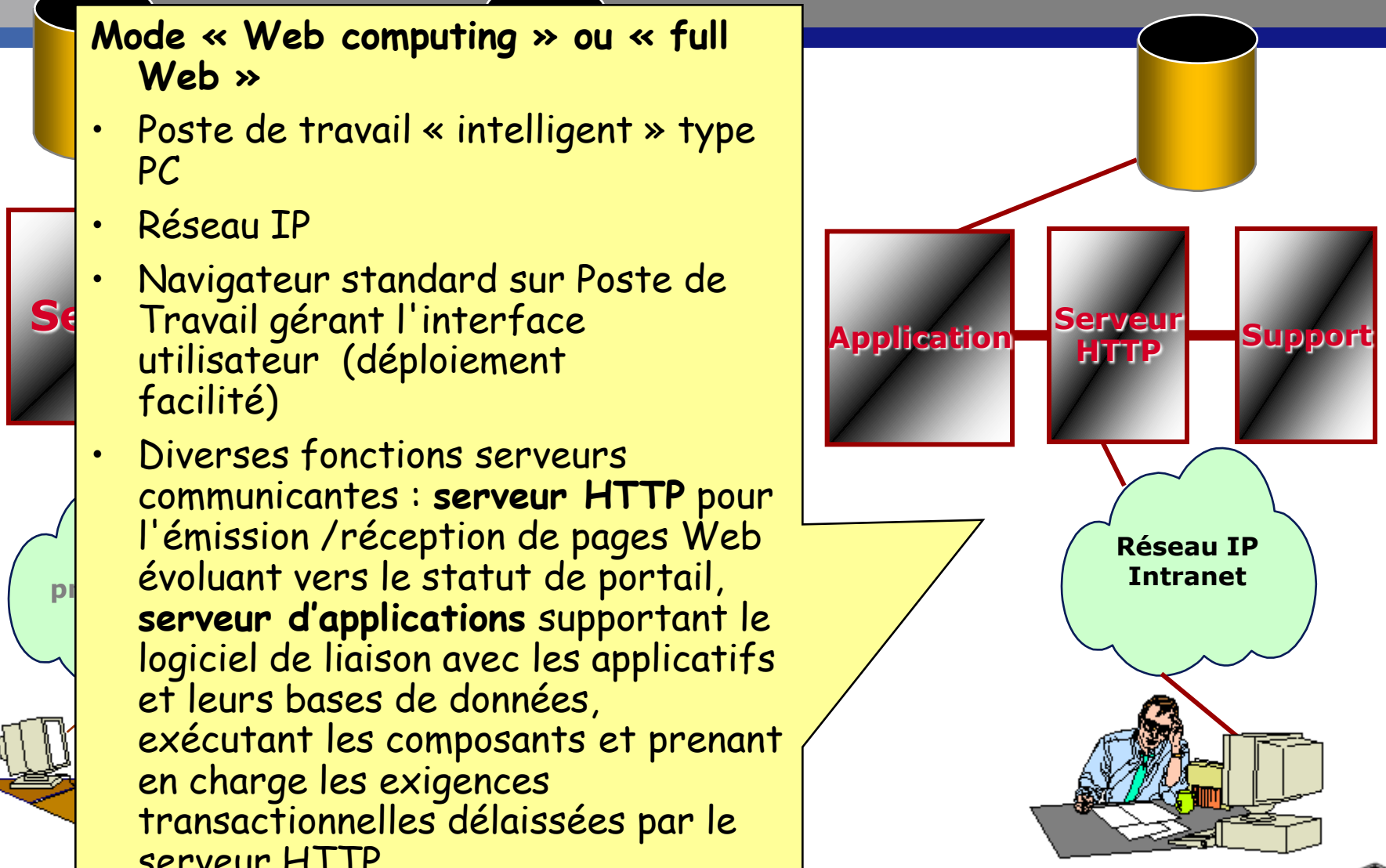
- **Client léger**
  - Tentative de résolution du problème du client lourd avec le concept de client léger (type *Citrix Metaframe* (protocole *ICA*) – *Microsoft terminal Server* (protocole *RDP*)).
  - Le client léger intercepte toutes les interactions avec l'utilisateur (clavier et souris), et les transmet au serveur.
  - Le serveur effectue son traitement en considérant que le clic et/ou la frappe de clavier a été faite en local.
  - Puis, il renvoie au client léger, une image ou des commandes de dessin d'images suivant le procédé.
  - Toute la logique applicative C/S (la partie serveur et la partie ex-client) reste donc à la charge du serveur.



Mode « full web »

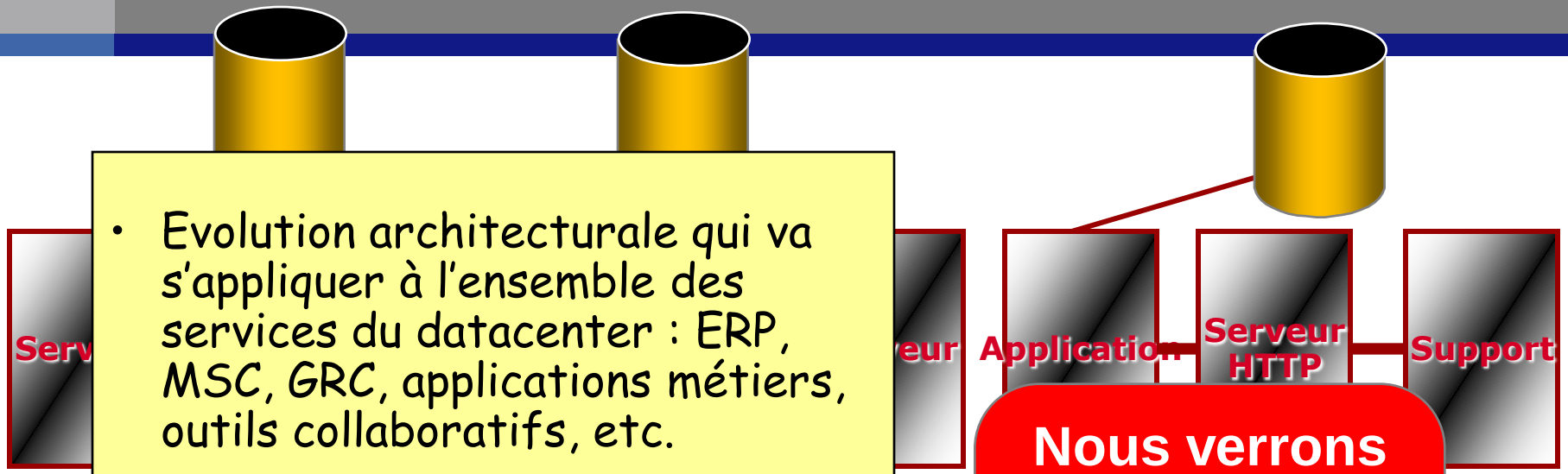
Mode « Web computing » ou « full Web »

- Poste de travail « intelligent » type PC
- Réseau IP
- Navigateur standard sur Poste de Travail gérant l'interface utilisateur (déploiement facilité)
- Diverses fonctions serveurs communicantes : **serveur HTTP** pour l'émission /réception de pages Web évoluant vers le statut de portail, **serveur d'applications** supportant le logiciel de liaison avec les applicatifs et leurs bases de données, exécutant les composants et prenant en charge les exigences transactionnelles délaissées par le serveur HTTP.



Mode « full web »

- Evolution architecturale qui va s'appliquer à l'ensemble des services du datacenter : ERP, MSC, GRC, applications métiers, outils collaboratifs, etc.
- De leur côté, les serveurs Web classiques évoluent vers le statut de portail avec leurs outils de personnalisation et de gestion de contenu, les annuaires et les outils d'administration.
- **Impératif** : Nécessité de rationaliser en mutualisant diverses fonctions.



Nous verrons comment lors de l'étude des architectures globales

La nouvelle distribution des rôles

- Le **client web** :
  - interprète les balises des pages HTML ou XML, interprète le code des scripts (Javascript, VBscript), exécute les applets Java.
  - Interagit avec un serveur web –ou directement avec un serveur d'applications en HTTP/S
  - Possède différents niveaux de sécurité configurables
- Le **serveur Web** :
  - fournit du contenu Web (HTML, …), communique via HTTP, traite des requêtes CGI et peut être le frontal d'un serveur d 'applications
- Le **serveur d'applications** :
  - Permet d'exécuter des composants (type EJB pour JEE) conformes aux standards en vigueur (JEE, .Net), indépendants du visuel et de l'accès aux données, aptes à être déployés dans un environnement donné, permettant une large possibilité d'extension de puissance et s'affranchissant du lieu;
  - Réalise des services d'administration et de sécurité (niveau composants, niveau méthode)

Le serveur d'application

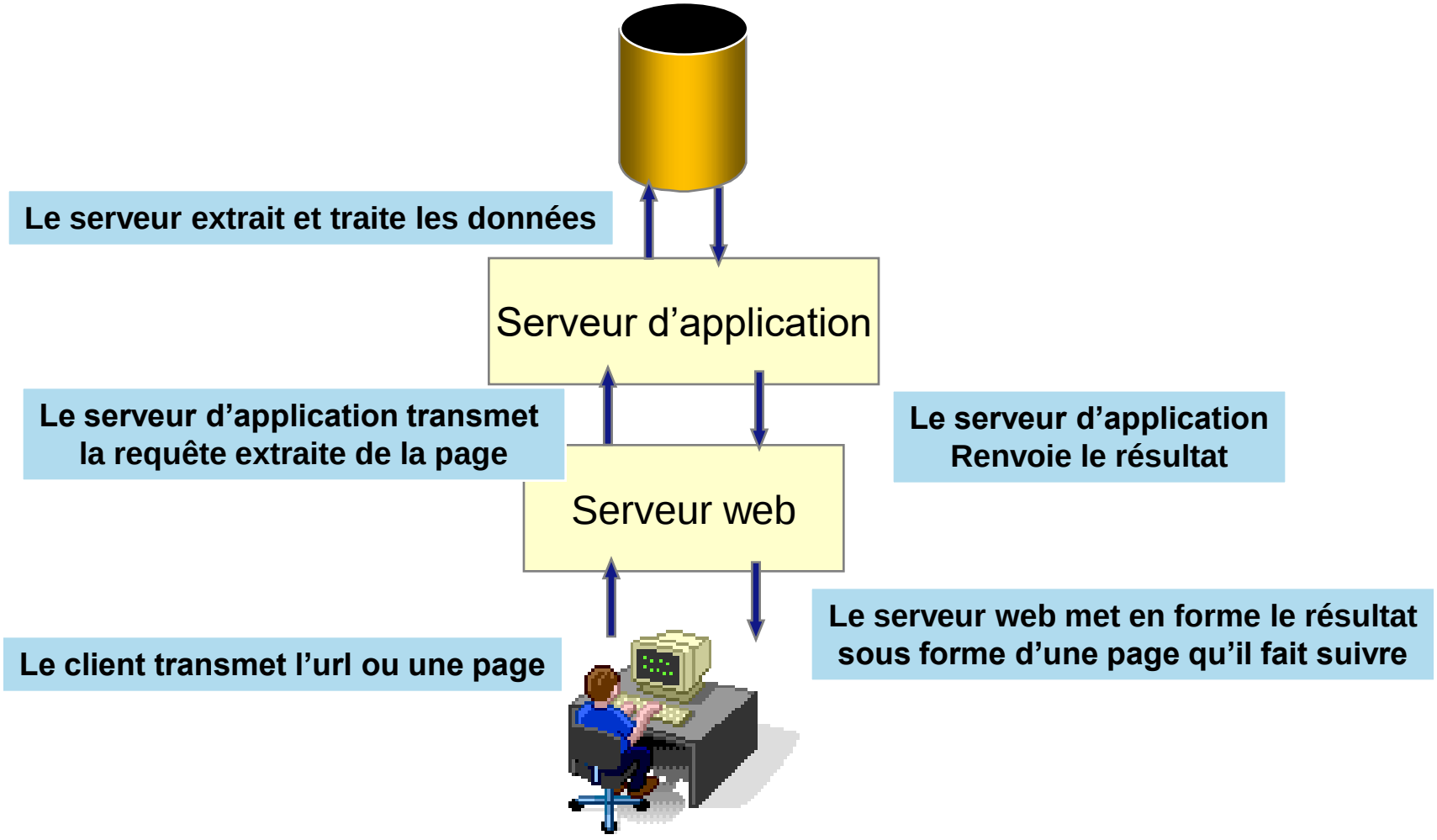
- Ce n'est pas, contrairement à ce que son nom pourrait laisser supposer, le serveur hébergeant les logiciels applicatifs (ERP, MSC, CRM, métiers, décisionnel, collaboratif, etc.).
- Au sein des architectures actuelles, le serveur d'application joue un rôle de connecteur entre les systèmes existants et les services internet.
- Il y a autant de définitions d'un serveur d'application que de fournisseurs.
- Les caractéristiques habituellement reconnues :
  - S'interfacer avec un serveur HTTP et fournir un moteur d'exécution des traitements. Le serveur d'application constitue donc l'environnement d'exécution des applications côté serveur
  - S'ouvrir à l'existant de l'entreprise par le biais d'interfaces avec les applications et leurs bases de données.
  - Répondre aux contraintes transactionnelles des architectures centralisées : gestion des contextes web, différenciation des utilisateurs, répartition des charges, sécurité, distribution, accès concurrents, persistance des objets… Il prend donc en charge l'ensemble des fonctionnalités qui permettent à N clients d'utiliser une même application :
    - Gestion de la session utilisateur
    - Reprise sur incident
    - Ouverture sur de multiples sources de données

Le serveur d'application

- Lorsque l'utilisateur envoie une requête sous forme d'une URL ou d'une page contenant un questionnaire, celle-ci, après éventuelle résolution d'un nom symbolique en une adresse IP, aboutit à un serveur HTTP (Web).
- Celui-ci est indispensable mais ses capacités sur le plan applicatif sont relativement limitées.
- Il sait retransmettre au client les fichiers statiques présents sur le serveur (pages HTML , images gif ou jpeg, fichiers CSS,...).
- Dès que l'URL porte sur une page dynamique c'est à dire autorisant un traitement, le serveur Web aiguille cette demande vers une autre brique, capable de gérer ce traitement applicatif.
- Cette brique assure la fonction de base « serveur d'application » .
- Une fois, le traitement effectué, le serveur d'application renvoie la page HTML (ou autre format) au serveur Web qui se charge de la router vers le bon destinataire.

Architectures de traitement

Serveur d'application et serveur web



Serveurs d'application propriétaires

- Les moteurs *Microsoft ASP* et *Macromedia Cold Fusion* sont à ce titre des serveurs d'application (même si il sont intégrés au Serveur Web comme pour ASP).
- Aujourd'hui, les progiciels dits « serveurs d'application » vont largement au-delà de la fonction de base que nous venons de décrire :
  - *2X Application Server*
  - *Oracle* ex *BEA Weblogic* et *Oracle Application Server*
  - *Borland Application Server*
  - *IBM Websphere Application Server*
  - *Novell exteNd Application Server*
  - *Orion*
  - *Sun Java System Application Server*
  - *Nirva Application Platform*
  - *AS de l'architecture Netweaver* de *SAP*
  - *Microsoft AppFabric ex Dublin*

Serveurs d'application open-source

- Les moteurs *JSP/Servlets et PHP* sont à ce titre des serveurs d'application (même si il sont intégrés au Serveur Web comme pour *PHP*).
- Aujourd'hui, les progiciels dits « serveurs d'application » vont largement au-delà de la fonction de base que nous venons de décrire :
  - *Apache Tomcat* (moteur de *JSP/Servlets*).
  - *JBoss* ;
  - *JOnAS* (*Java Open Application Server*);
  - *GlassFish* serveur d'application *open-source* de *Sun* ;
  - *Geronimo* (le projet de *l'Apache fondation*);
  - *Mille-Xterm* - Une infrastructure libre pour le déploiement massif et centralisé de terminaux ;
  - *RIP* ;
  - *Zope* (connu pour être associé au *CMS Plone* et au langage *Python*) .

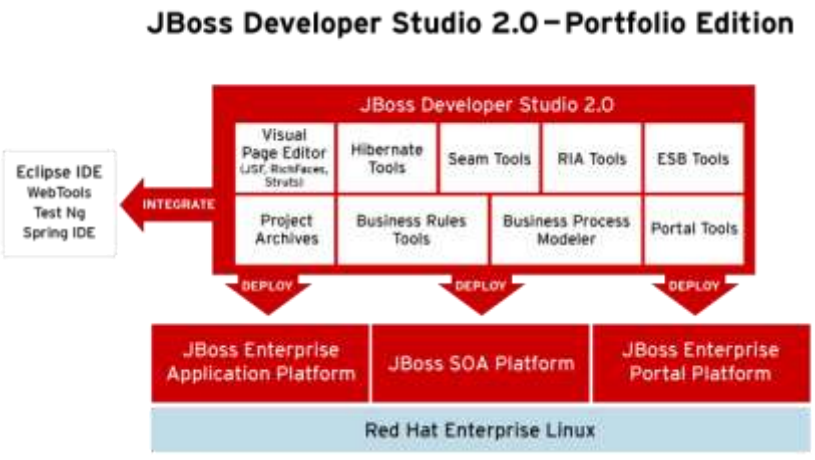
JBoss

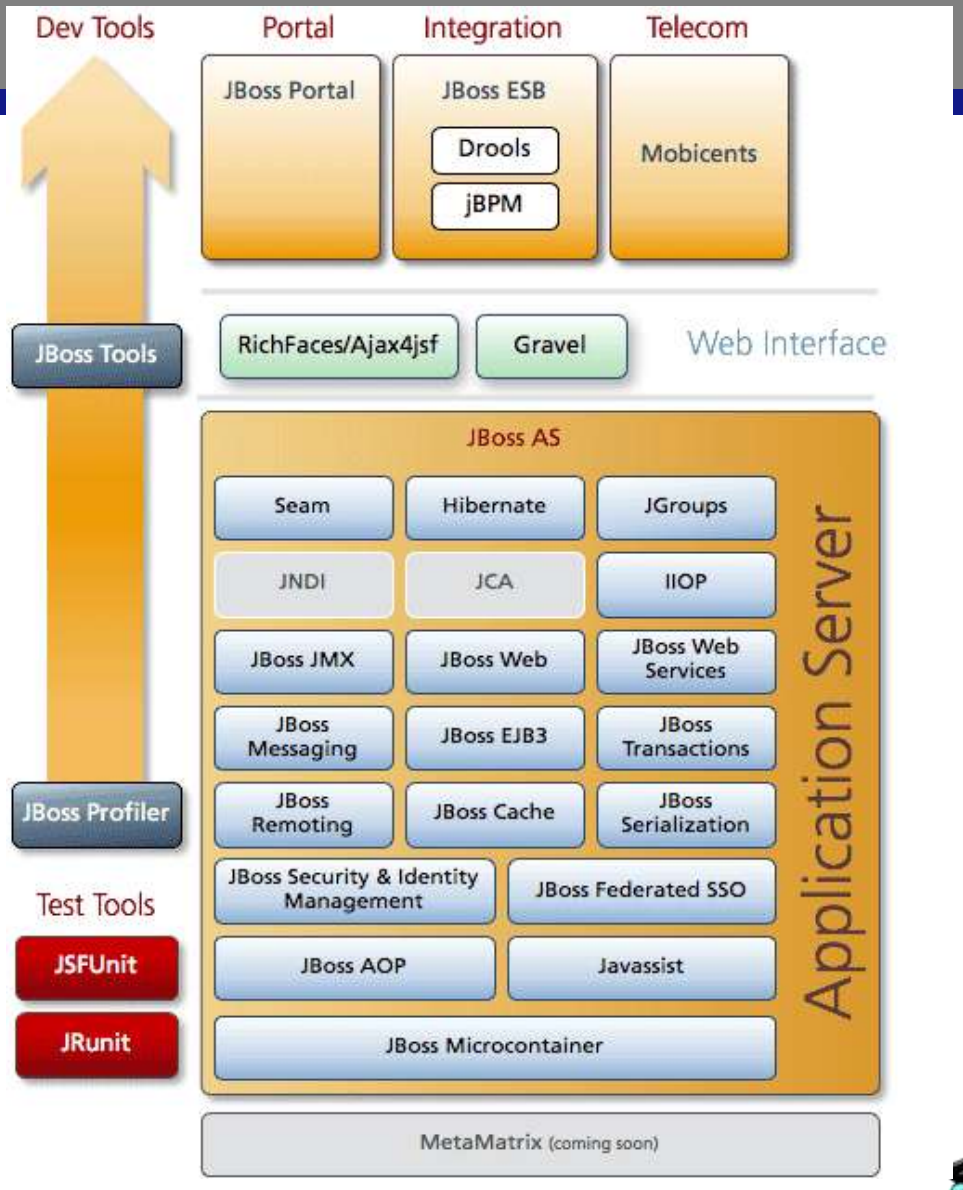
- En 1999, première version de *JBoss*, un serveur d'applications open -source proposé avec une licence *LGPL*.
- *JBoss Inc* (créée par Marc Fleury) regroupe le projet *JBoss* et d'autres projets tels que *Tomcat, Hibernate, jGroups, jBPM, etc.*
- Au printemps 2004, JBoss crée une solution middleware open -source : JEMS (JBoss Enterprise Middleware System).
- Les projets JEMS incluent *JBoss Portal*, Hibernate (*framework open -source* gérant la persistance et le *mapping* des objets d'une base de données relationnelle), *JBoss Cache* (technologie de cache fin), *JGroups* (boîte à outils pour une communication multidiffusion fiable, *JBoss Eclipse IDE*, un EDI basé sur Eclipse, et *jBPM* (moteur de *workflow open-source*).

JBoss

- En juillet 2004, *JBoss* devient le premier serveur d'application open -source compatible *JEE* (à l'époque *J2EE*), en passant avec succès plus de 50 000 tests de la suite de tests de compatibilité de Sun.
- En avril 2006, *Red Hat* rechète *JBoss Inc*.
- *JBoss Application Server* implémente entièrement l'ensemble des services *JEE* : Cela inclut *JBoss Portal, JBoss Seam, Tomcat et les frameworks Hibernate, jBPM, et Rules.*

JBoss


JBoss Developer Studio 2.0 – Portfolio Edition
JBoss Developer Studio 2.0
Visual Page Editor
Hibernate Tools
Seam Tools
RIA Tools
ESB Tools
Project Archives
Business Rules Tools
Business Process Modeler
Portal Tools
Eclipse IDE
WebTools
Test Ng
Spring IDE
INTEGRATE
DEPLOY
DEPLOY
DEPLOY
JBoss Enterprise Application Platform
JBoss SOA Platform
JBoss Enterprise Portal Platform
Red Hat Enterprise Linux

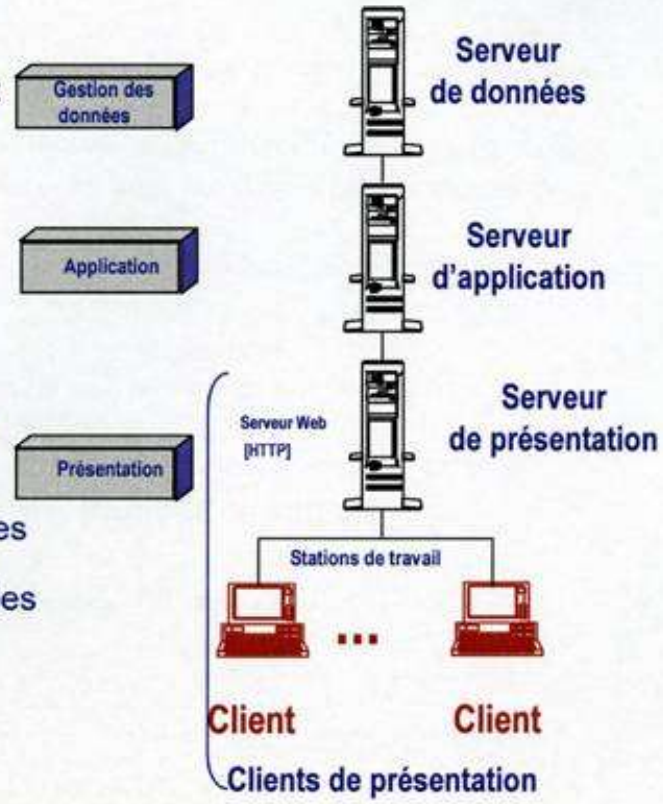

Dev Tools
Portal
Integration
Telecom
JBoss Portal
JBoss ESB
Drools
jBPM
Mobicents
JBoss Tools
RichFaces/Ajax4jsf
Gravel
Web Interface
JBoss AS
Seam
Hibernate
JGroups
JNDI
JCA
IIOP
JBoss JMX
JBoss Web
JBoss Web Services
JBoss Messaging
JBoss EJB3
JBoss Transactions
JBoss Profiler
JBoss Remoting
JBoss Cache
JBoss Serialization
JBoss Security & Identity Management
JBoss Federated SSO
Test Tools
JSFUnit
JBoss AOP
Javassist
JRunit
JBoss Microcontainer
Application Server
MetaMatrix (coming soon)

Exemple chez un grand compte

LA BANQUE POSTALE

Infrastructures informatiques de La Banque Postale
Architecture de production

- **Architecture n-tiers en mode Web**
  - De nouvelles technologies sont venues compléter les technologies traditionnelles :
    - Technologies issues d'Internet, et principalement du Web.
    - Technologies objet
    - Technologies distribuées
  - Le client est léger [« thin client »] et est instancié par un navigateur web
  - Le serveur de présentation est un serveur web [TTP]
  - Le serveur d'application traite les demandes
  - Le serveur de données héberge les données



Avantages et inconvénients du « full web »

- Avantages
  - Facilité d'installation et de déploiement (pas de couche client à déployer, le navigateur est déjà en place) => principe d'un « **client léger** »
  - Meilleure maîtrise des coûts (Bien adapté au modèle « *on demand* »).
  - Besoins en infrastructures informatiques plus légers.
  - Evolutivité et montée en version.
  - Adaptation aux solutions de mobilité.
- Inconvénients
  - Un serveur conçu pour délivrer des pages, non pour mener à leur terme des transactions.
  - Des IHM jugées parfois un peu rustiques.

Avantages et inconvénients du « full web »

- Avantages
  - allation et de déploiement (pas de à déployer, le navigateur est déjà en ncipe d'un « **client léger** »
  - îtrise des coûts (Bien adapté au *lemand* »).
  - infrastructures
  - Évo[illegible] et montée en versio[illegible]
  - Adapta[illegible]n aux solutions de mo[illegible]
- Inconvénients
  - Un serveur conçu pour déliv[illegible] mener à leur terme des transactions.
  - Des IHM jugées parfois un peu rustiques.

Une architecture globale qui doit compenser cette faiblesse

(prochains chapitres)

Pouvoir développer des interfaces plus sophistiquées

(suite immédiate)

Applications internet riches (RIA)

- « *Full Web* » vécu comme régression par rapport au C/S en ce qui concerne l'IHM.
- Combiner performance des applications PC, légèreté et simplicité des interfaces Web.
- *RIA* : Application Web lancée depuis le navigateur, exécutée dans le navigateur, offrant des caractéristiques similaires aux logiciels PC traditionnels (Concept *Macromedia* 2002)
- DHTML/AJAX http://www.openajax.org
- Quelques solutions *RIA* :
  - Communauté Java avec JavaFX
  - *Adobe (ex Macromedia) Flash / Flex (Composants MXML et langage ActionScript)*
  - *GWT (Google Web Toolkit)*
  - *Microsoft Silverlight (V4)*

Applications internet riches (RIA)

- Exemple de *Adobe Flash Builder for Force.com.*
- Objectif : insuffler des fonctionnalités *RIA Flash* dans le large éventail d'applications *cloud* proposées par *Salesforce.com*.
- Intégration des IDE *Force.com* et *Flex / Flash*
- Basé sur *WSDL* (*Web Services Description Language*).
- Possibilité d'exploiter directement (ou de faire évoluer) son application ou interface riche depuis n'importe quel poste connecté au Web en accédant simplement sur un compte *Salesforce.com.*
- Au-delà du fait de ne plus avoir à changer d'environnement de développement pour développer et déployer des applications *Flash* sur *Salesforce.com*, les développeurs pourront également synchroniser leurs données sur l'environnement *cloud* depuis leur poste de travail.

Rich Desktop Applications (RDA)

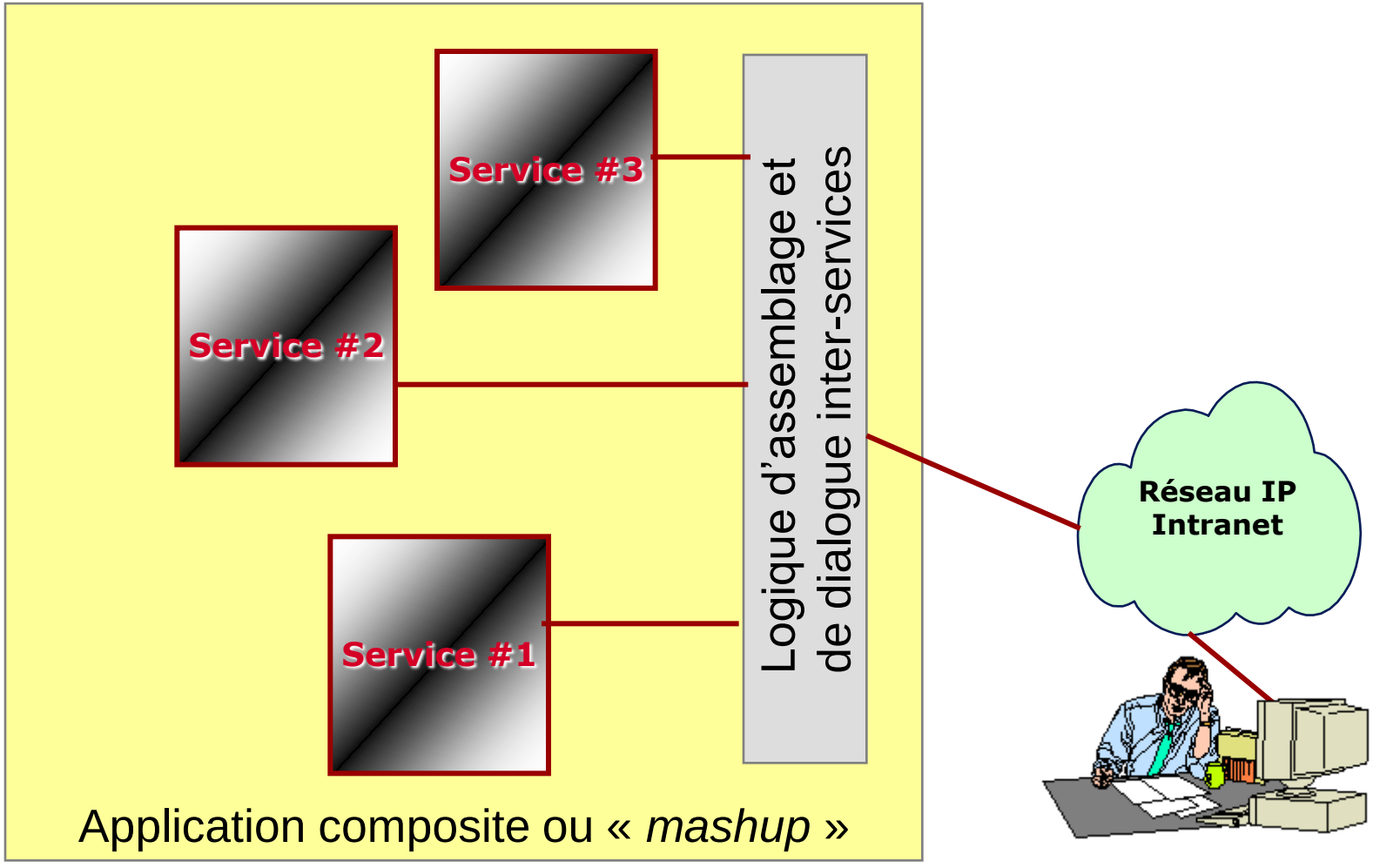
- *RDA* est une application en ligne (idem *RIA*) mais elle s'exécute en dehors du navigateur.
- Une application *RDA* accessible via *Firefox* n'est pas exécuté par *Firefox* lui même mais par *XulRunner*, une machine virtuelle.
- Intérêt des *RDA* : pouvoir allier la praticité des RIA et la robustesse des applications traditionnelles.
- Quelques technologies *RDA* :
  - en Java à l'aide de *Java Web Start*, *Eclipse RCP, Swing Application Framework* ;
  - Adobe *AIR (Adobe Integrated Runtime)*
  - *Microsoft* avec *.Net Framework WPF* et *XAML*
  - en XUL (le langage de description d'interfaces de *Mozilla*) avec *XulRunner* ;

Nouvelle importance de cette problématique

- Applications des 200s', accès via un PC windows
- Application des 201s', accès via de multiples terminaux (Desktop, laptop, tablette, smartphone, etc), compliqué par l'existence de plusieurs OS mobiles.
- Choix :
  - Framework UI du langage natif du terminal (Silverlight pour le PC windows, Objective C pour l'iPhone et l'iPad, Java pour Androïd)
  - Interface web commun, mais difficultés pour tirer parti de certaines fonctionnalités ou accèder à certains services (usage du framework Phonegap pour mettre en oeuvre les API de la caméra ou de l'accéléromètre du smartphone)

Architectures de traitement

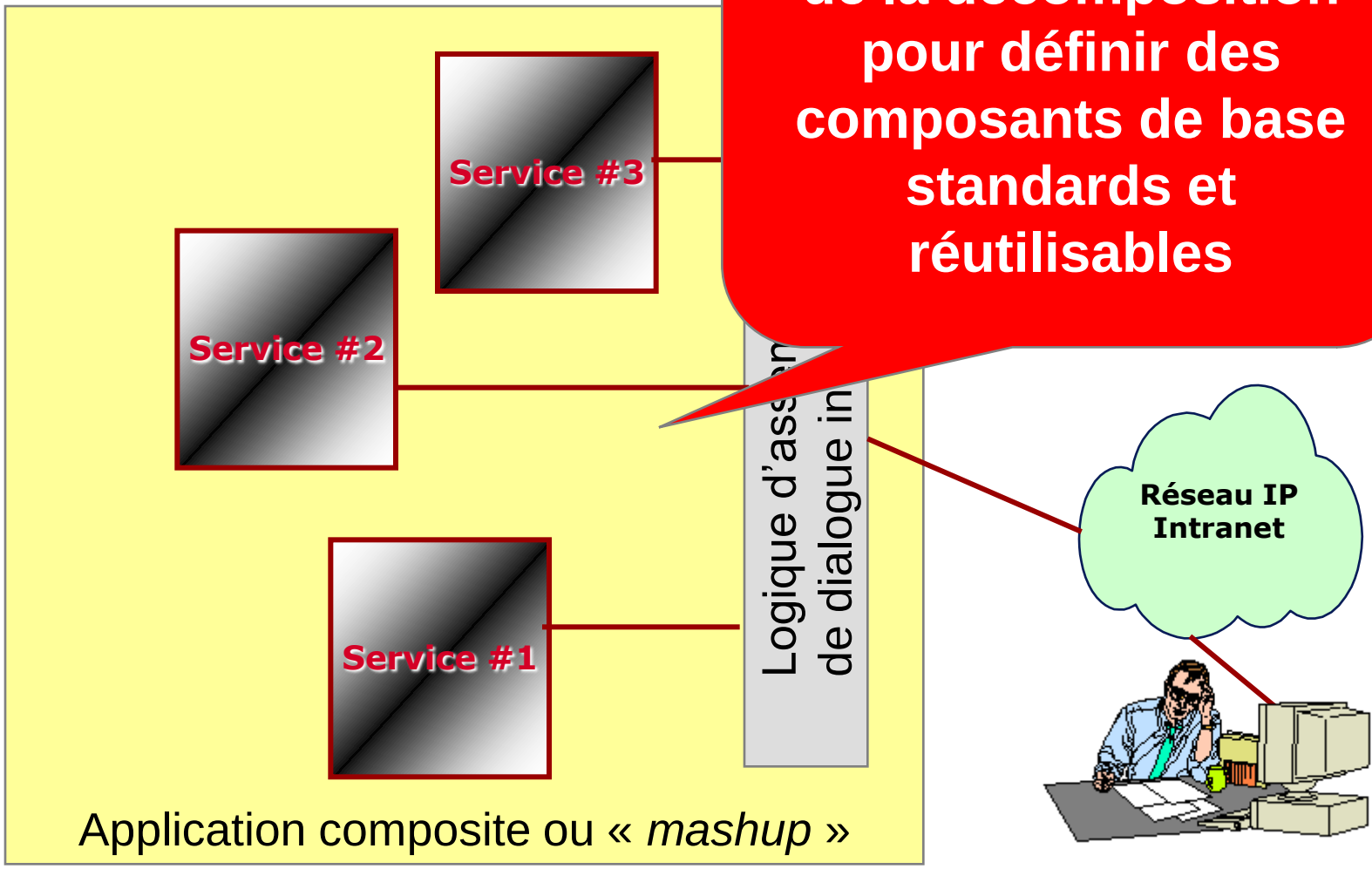
De l'application aux composants, des composants aux services



Architectures de traitement

De l'application aux composants, des composants aux services

Généraliser le principe de la décomposition pour définir des composants de base standards et réutilisables



Un exemple : les Web Services

- Les applications distribuées peuvent être bâties par assemblage de multiples composants fonctionnels qui ignorent tout de leurs multiples implémentations techniques respectives.
- Chaque composant est une boîte noire dotée d'un ensemble de fonctions dont certaines sont accessibles aux autres composants.
- Ces points d'entrée publics, appelés **services web**, sont décrits dans des documents mis à la disposition des autres services.
- Une telle isolation entre l'interface et l'implémentation des composants permet aux applications distribuées de faire appel à des services hétérogènes, déployés sur divers systèmes.

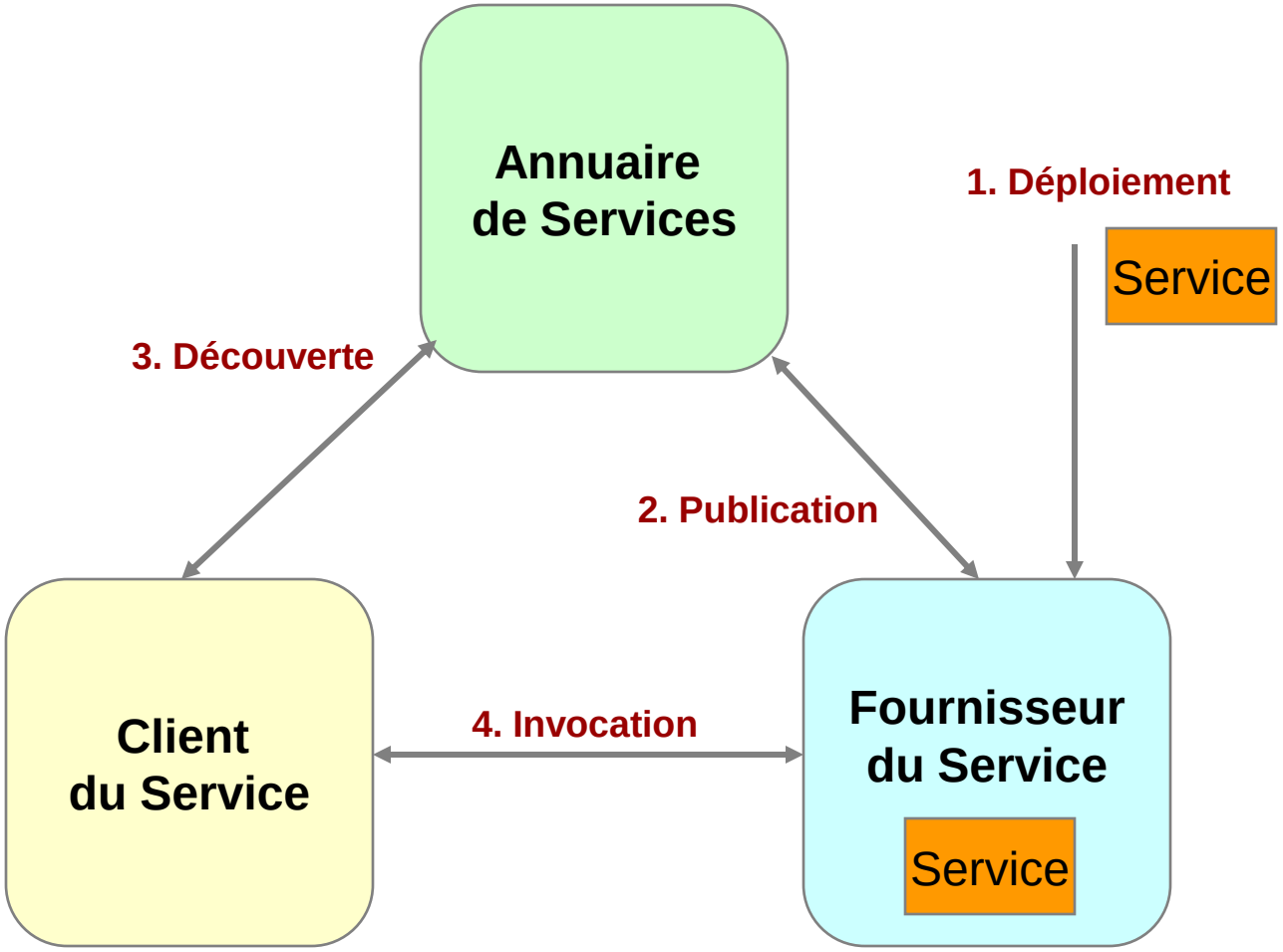
Un exemple : les Web Services

- Un ***Web Service*** est donc une application modulaire :
  - Mise à disposition sur l'Internet ou sur un réseau privé (Intranet),
  - Auto-descriptive (*WSDL*), publiable (*UDDI*) et accessible (*SOAP*) en utilisant le langage *XML* et les protocoles standards du Web,
  - Indépendante du système d'exploitation et du langage de programmation,
  - Visant à exposer une ou plusieurs fonctionnalités métier ou de gestion.
- Un ensemble de *Web Services* élémentaires peut être combiné (*WS-BPEL*) pour aboutir à un *Web Service* à valeur ajoutée.

Protocoles des Web Services

- Le modèle des *Web Services* est défini par une architecture et un ensemble de protocoles standardisés : *SOAP, WSDL, UDDI*.
  - Spécifications garanties par W3C et OASIS,
  - Interopérabilité entre implantations gérée par la WS-I (*Web Services Interoperability*) *Organization*.
- Objectifs du modèle
  - **Modularité**,
    - Réutilisation et composition de services,
  - **Interopérabilité**,
    - Dialogue entre environnements et plate-formes hétérogènes,
    - Couplage faible (communications synchrones/asynchrones),
  - **Intégration**,
    - Intégration du système d'information au sein et en dehors de l'entreprise,
    - Masquage de la complexité.

Principe des Web Services



Principe des Web Services

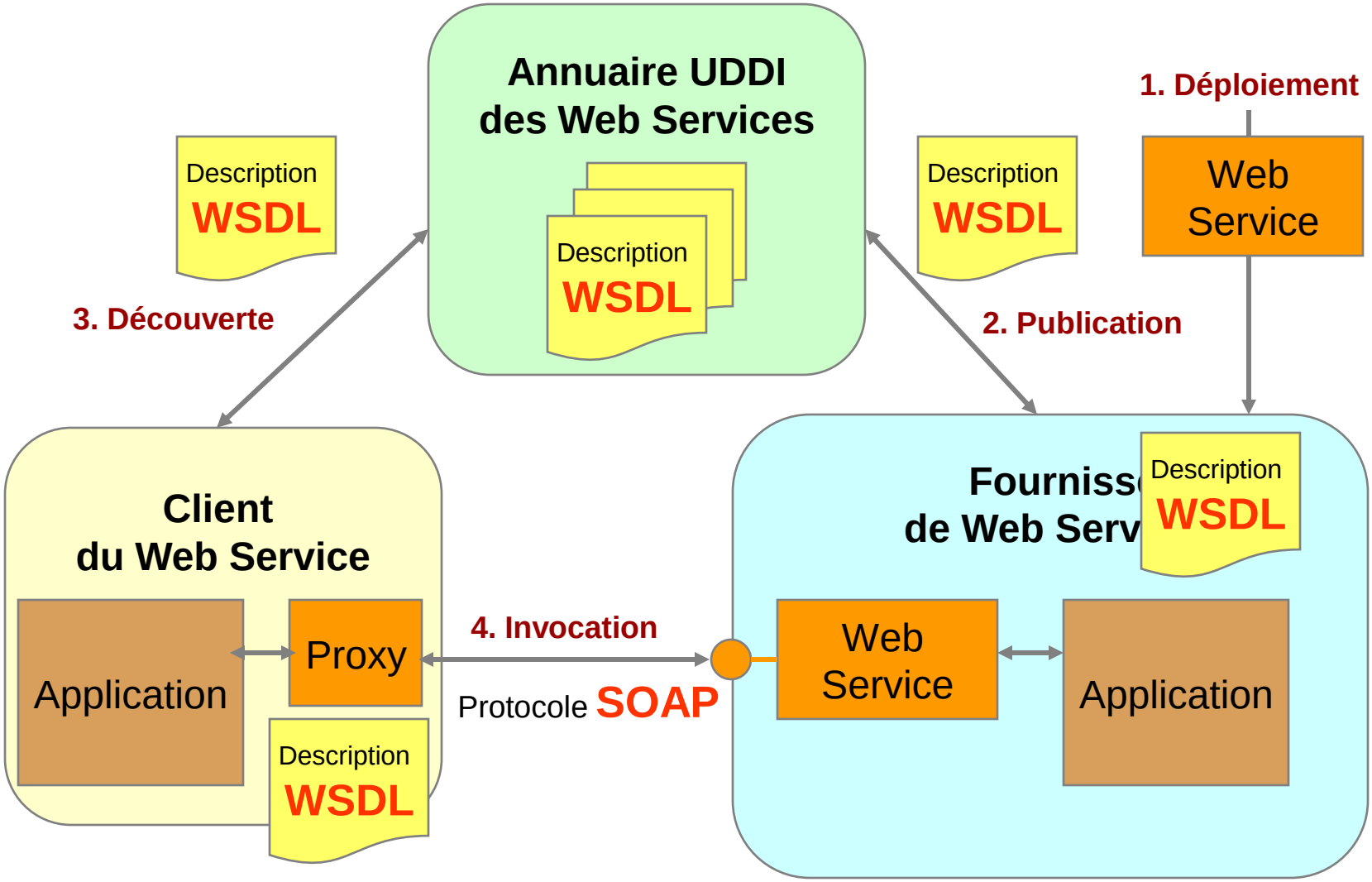
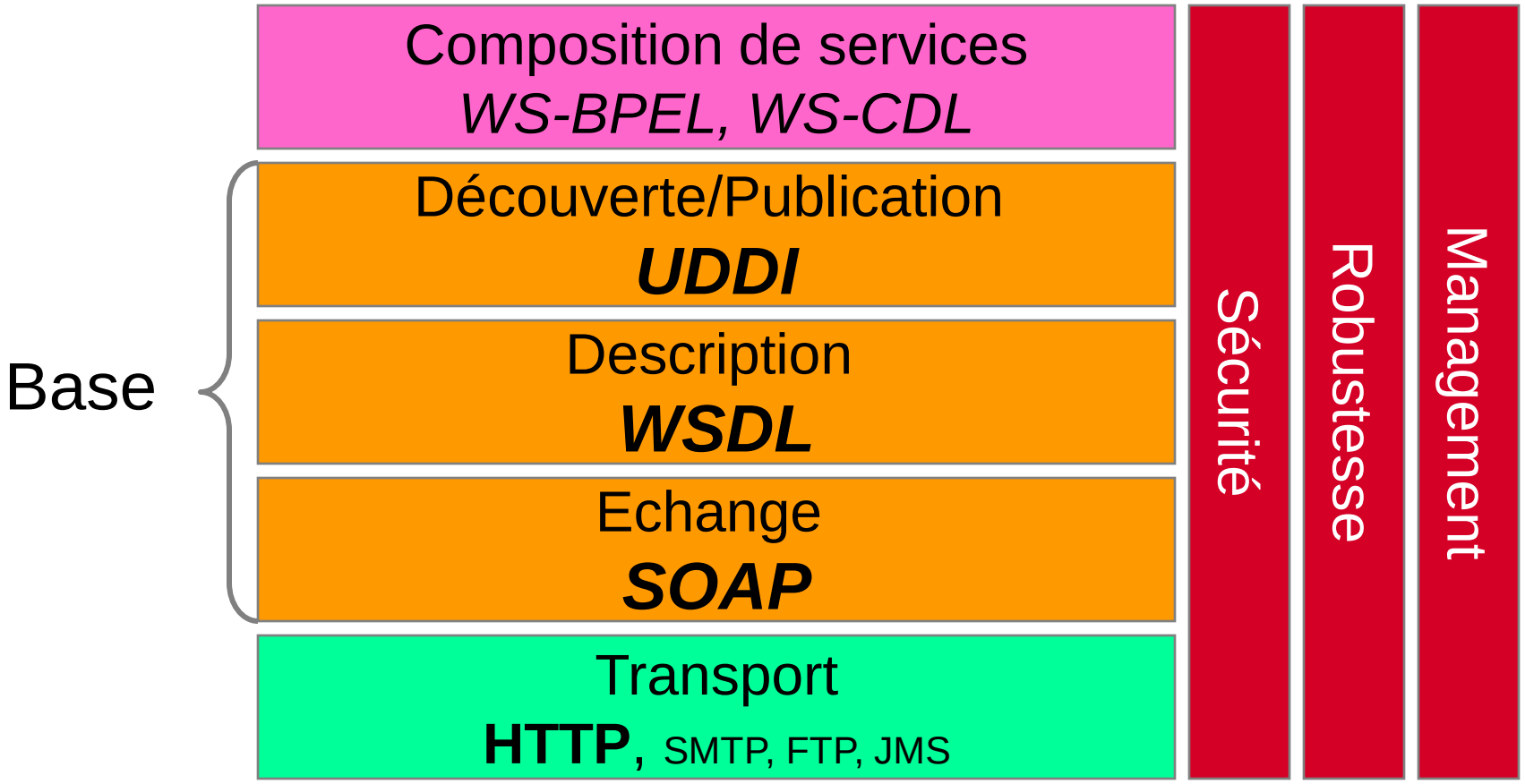
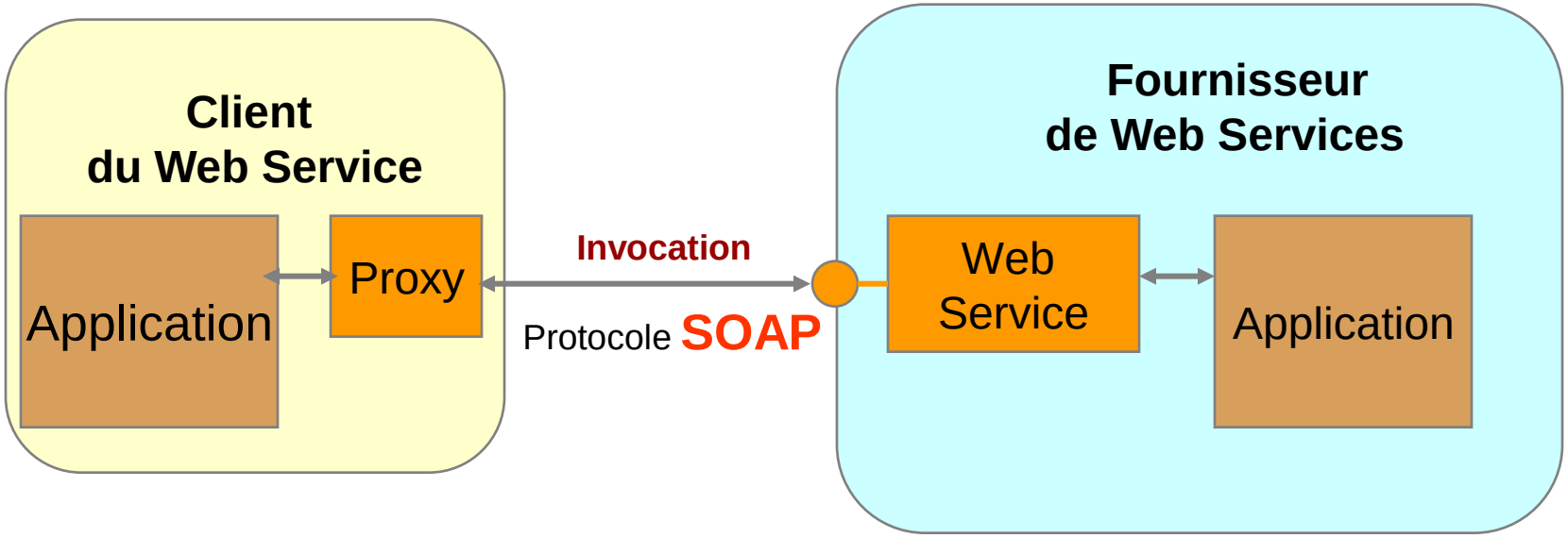


Diagramme des protocoles



Invocation d'un Web Service



Terminologie

Initialement : *Simple Object Access Protocol*
Temporairement : *Service Oriented Architecture Protocol*
Aujourd'hui : *SOAP* n'est plus un acronyme.

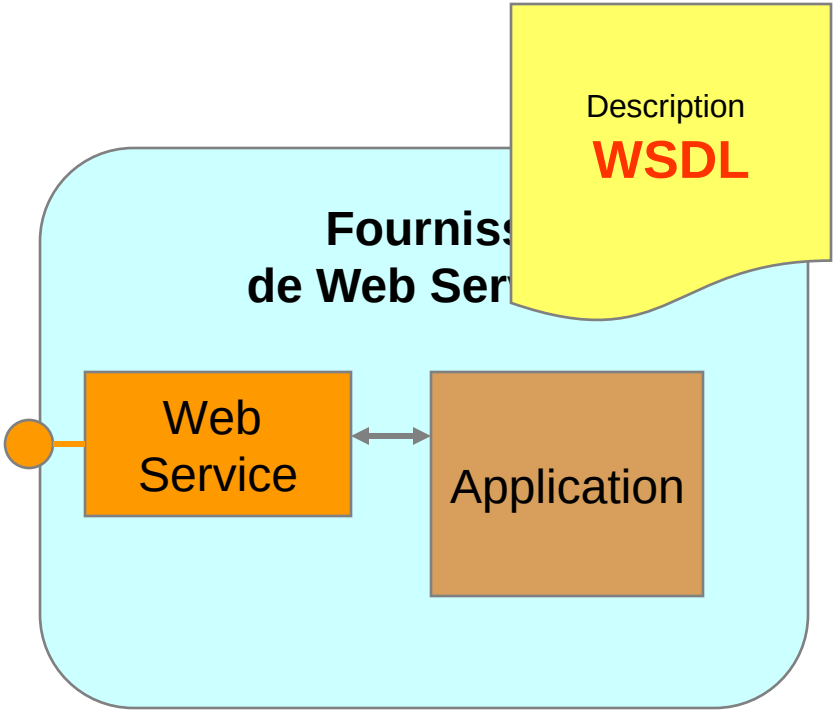
Qu'est-ce-que Soap ?

- Définition
  - Spécifié à l'origine par *Microsoft* pour transmettre des requêtes et des réponses en *XML* sur *HTTP*, ***Soap*** est un protocole d'échanges de données entre composants quelconques sur un réseau IP.
  - Utilisé pour véhiculer des appels de procédures distantes et donc pour invoquer des services web, Soap spécifie en *XML* des en-têtes de messages qui contiennent l'adresse du destinataire, le nom de la procédure à exécuter, ainsi que les paramètres attendus et retournés.
  - Si ce standard transporte des messages *XML*, il ne s'intéresse pas à leur contenu (responsabilité assurée par *WSDL*)
  - Transport sur le protocole HTTP (usuellement),
  - Ouvert à d'autres protocoles tels que *SMTP, POP3, FTP, JMS*
  - Large inspiration de *XML-RPC (Remote Procedure Call* du *C/S)*

Qu'est-ce-que XML ?

- ***XML*** se veut comme le successeur de *HTML* en tant que norme de description de documents sur l'Internet (ou sur un réseau privé au standard de l'Internet).
- Il reprend l'héritage du HTML qu'il combine avec les atouts du *SGML*, métalangage servant à décrire les langages de balises (comme *HTML*) et destiné à rendre le stockage de données indépendant de tout fournisseur de logiciel (comme *SQL* a rendu indépendant son mode d'interrogation).
- XML se révèle plus souple que les ancienne méthodes de description des composants mis en oeuvre au sein d'architecture comme Corba (Sun, IBM, Oracle, ...) ou DCOM (Microsoft).

WSDL : Description d'un web service



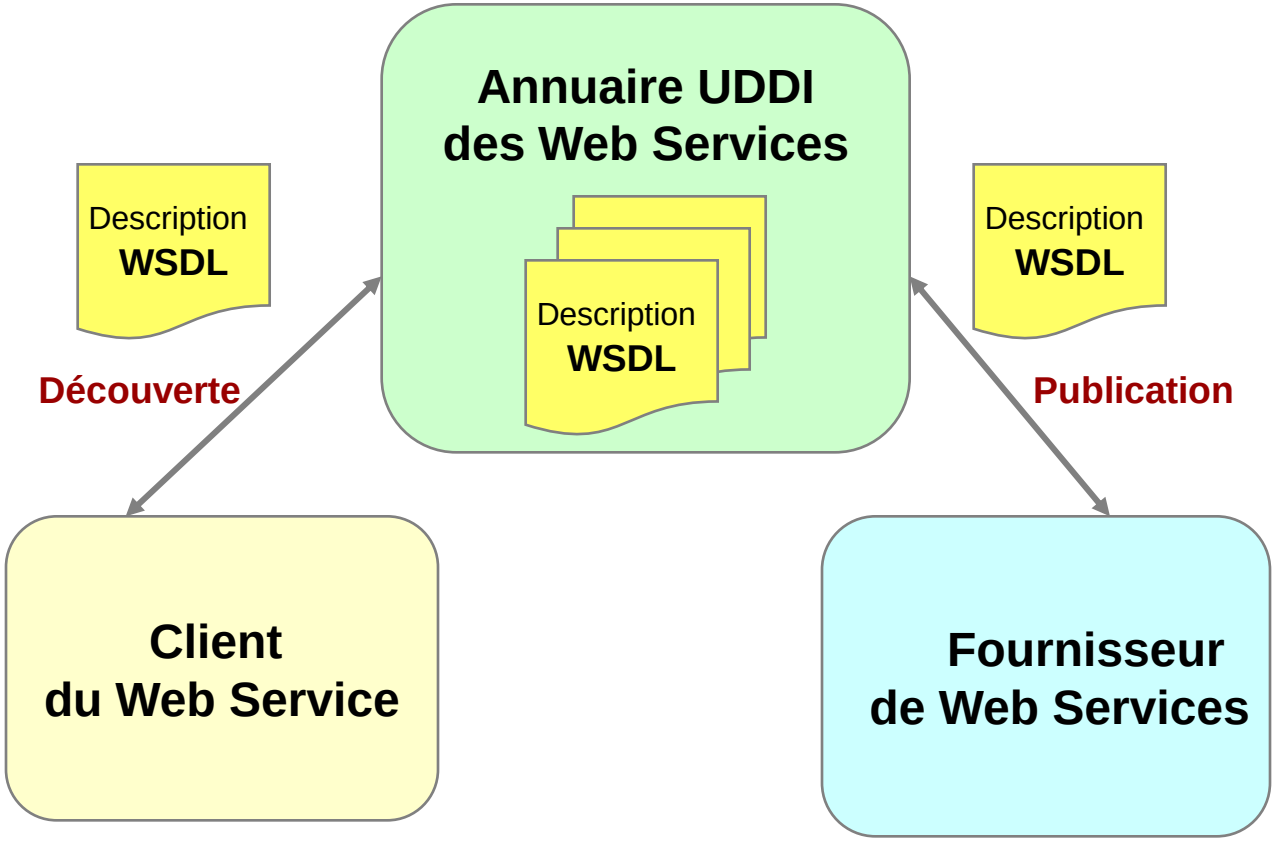
Qu'est-ce-que WSDL ?

- ***Web Service Definition Language***
- Langage de description des *Web Services*
  - Description abstraite de l'interface du service,
    - Ensemble d'opérations, de messages et de types de données
  - Description concrète de l'implantation du service,
    - Liaison à des formats de message concrets
    - Liaison à des protocoles de transport et des serveurs réseaux
- Document *XML* (Schéma *XML*)
- Initiative de *Ariba, IBM et Microsoft*
- Spécification standard géré par le W3C
  - WSDL 1.1, W3C Note, soumise en mars 2001

Qu'est-ce-que WSDL ?

- *WSDL* ne s'intéresse qu'à l'interface d'accès aux services web et n'impose aucune implémentation de leur contenu (*EJB*, servlet, composant *DCOM*, ..)
- Un service web est décrit comme un ensemble de ports, auxquels s'ajoutent une adresse Internet d'accès et au moins un structure de liaison (*binding*) qui définit un protocole d'accès (qui peut être *Soap*, mais aussi autre : *MIME*,..).
- Un port regroupe plusieurs opérations.
- Une opération correspond à une action élémentaire implémentée par un service web. Elle se compose d'un message d'entrée (requête), d'un message de sortie (réponse) ou d'un message signalant l'échec du traitement.

Découverte et publication d'un web service



Qu'est-ce-que UDDI ?

- ***Universal Description, Discovery and Integration***
- Service d'annuaires (Business Registry) recensant les entreprises et leurs services à l'échelle mondiale
  - Recherche et publication de services (dont Web Services)
  - Automatisation des échanges entre partenaires commerciaux
- Structure de données définie par une grammaire XML
- Accès par une API vue comme un Web Service
- Initiative de *Ariba, IBM et Microsoft*
- Spécification définie par un consortium industriel
  - *UDDI 3.0, UDDI Working Group, OASIS*

Qu'est-ce-que UDDI ?

- *UDDI* gère le répertoire des services accessibles sur le réseau.
- *UDDI* vise à éviter que ne se propagent sur l'Internet des annuaires hétérogènes pour la publication et la localisation de services web.
- Physiquement réparti sur les sites des fournisseurs, un annuaire logique universel doit fournir une vue sur l'ensemble des services web accessibles.
- *UDDI* doit aussi permettre la définition d'annuaires privés dans le cadre d'un intranet ou d'un extranet.
- Il s'appuie sur *XML* pour spécifier les informations permettant de publier ou de localiser un service : identité du fournisseur du composant (*URL*, description et secteur d'activité) et services proposés (nom, description textuelle, index de classification spécifique à *UDDI*)

Conception d'un web service

Voir Annexe 2

- Créer un web service à partir du modèle Services Web ASP.Net
- Création du code du WS
- Compilation via Debugger

Architecture Orientée Services

- SOA = *Services Oriented Architecture*
- AOS = Architecture Orientée Services
- SOA = une approche d'architecture permettant d'atteindre les objectifs de standardisation, de réutilisation, de flexibilité et, à terme, de réduction des coûts d'opération et des délais de déploiement de nouvelles fonctionnalités
- SOA définit un cadre de référence permettant de faire cohabiter au mieux les nouveaux développements, les progiciels et les applications existantes (*Legacy*)
- **Orientée Services** : chaque besoin métier identifié est couvert par un service métier, lui-même s'appuyant sur d'autres services et des composants.
- Construire une application revient alors à « assembler un bouquet de services »

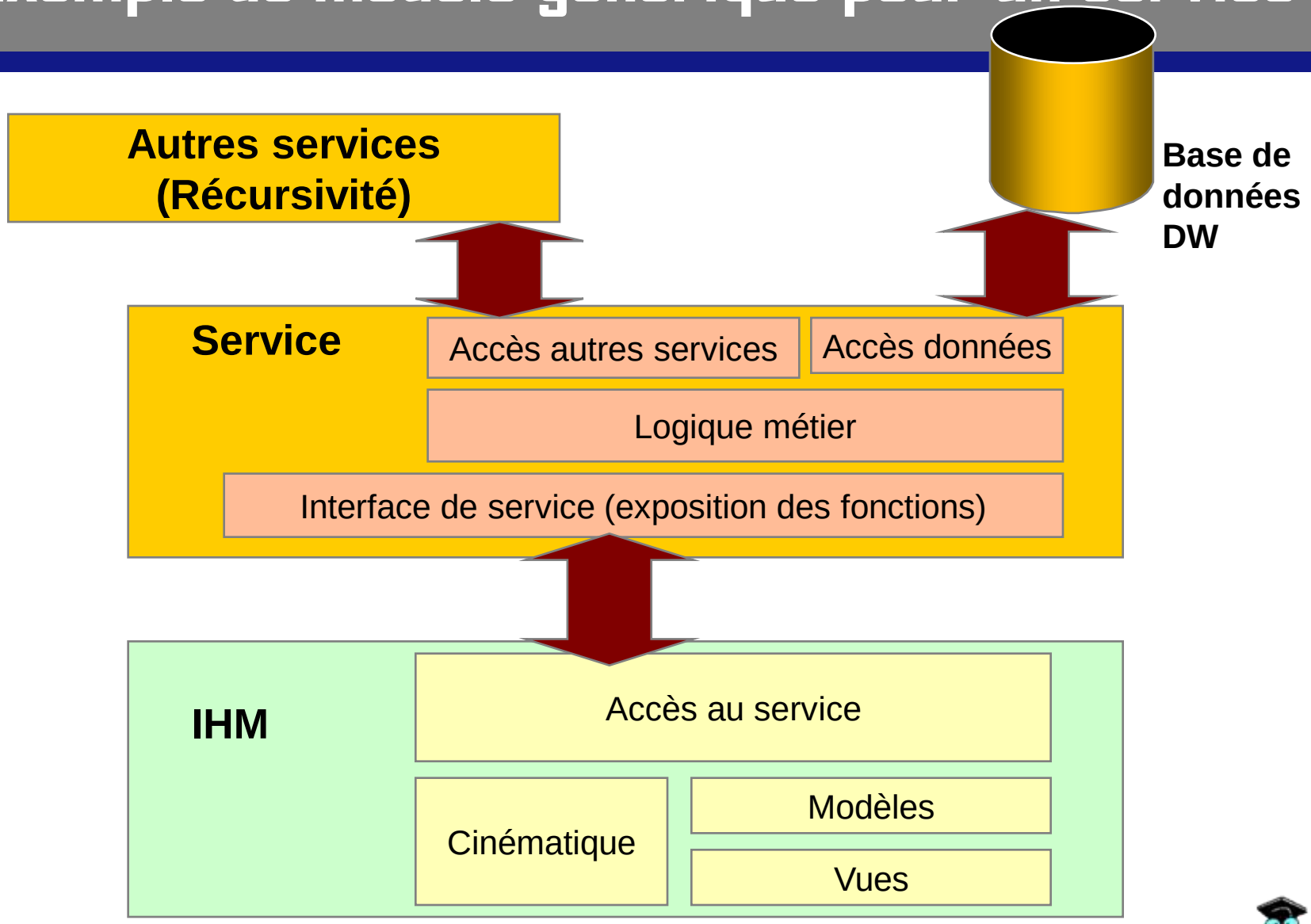
Architecture Orientée Services

- SOA = *Services Oriented Archi*
- AOS = Architecture Orientée S
- SOA = une approche d'archite d'atteindre les objectifs de sta réutilisation, de flexibilité et, à coûts d'opération et des délai nouvelles fonctionnalités
- SOA définit un cadre de référe cohabiter au mieux les nou progiciels et les applicati
- **Orientée Services** : couvert par un service métier, d'autres services et des compo
- Construire une application revien bouquet de services »

- Il ne faut pas confondre SOA et Web service.
- SOA est un principe architectural.
- Un Web service basé sur SOAP est une technologie, une façon parmi d'autres d'exposer du code applicatif en tant que service.
- On peut faire une architecture orientée services avec REST ("Representational State Transfer«)
- Bien qu'à la base orienté Ressources (concept d'URI), REST peut être utilisé pour développer des services.

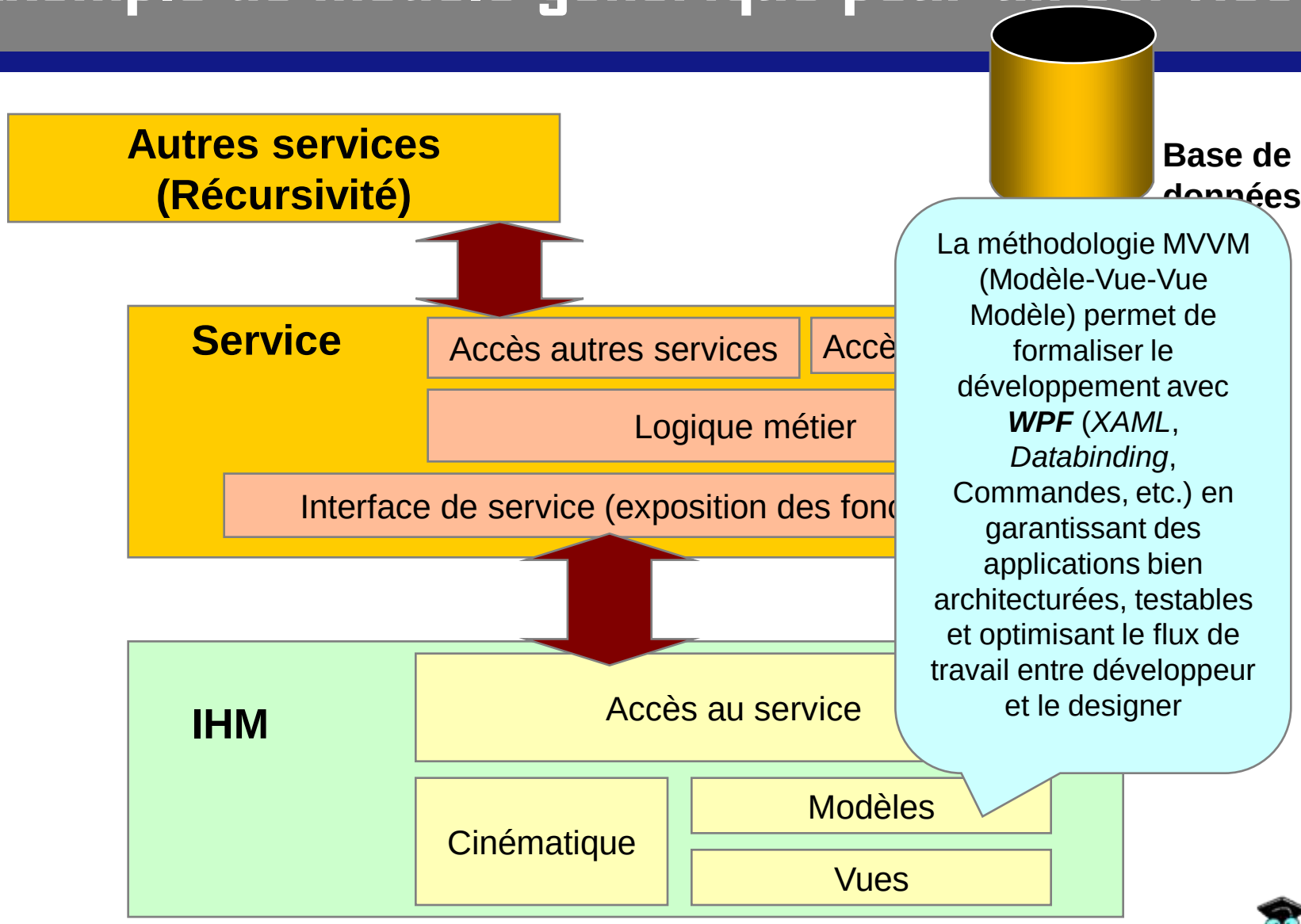
Architectures de traitement

Exemple de modèle générique pour un service

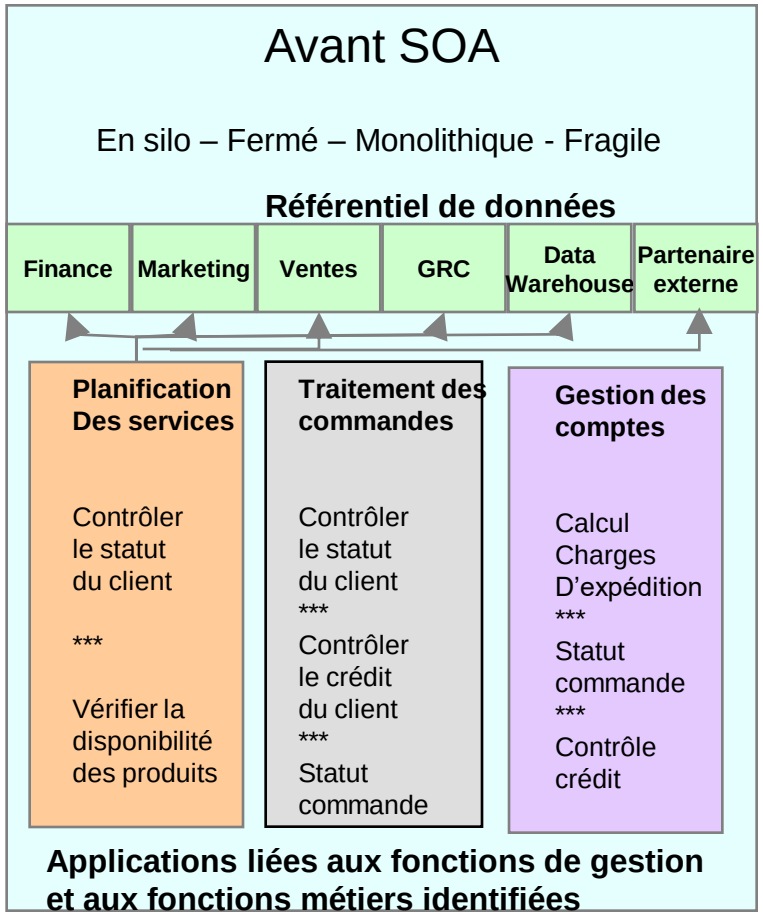


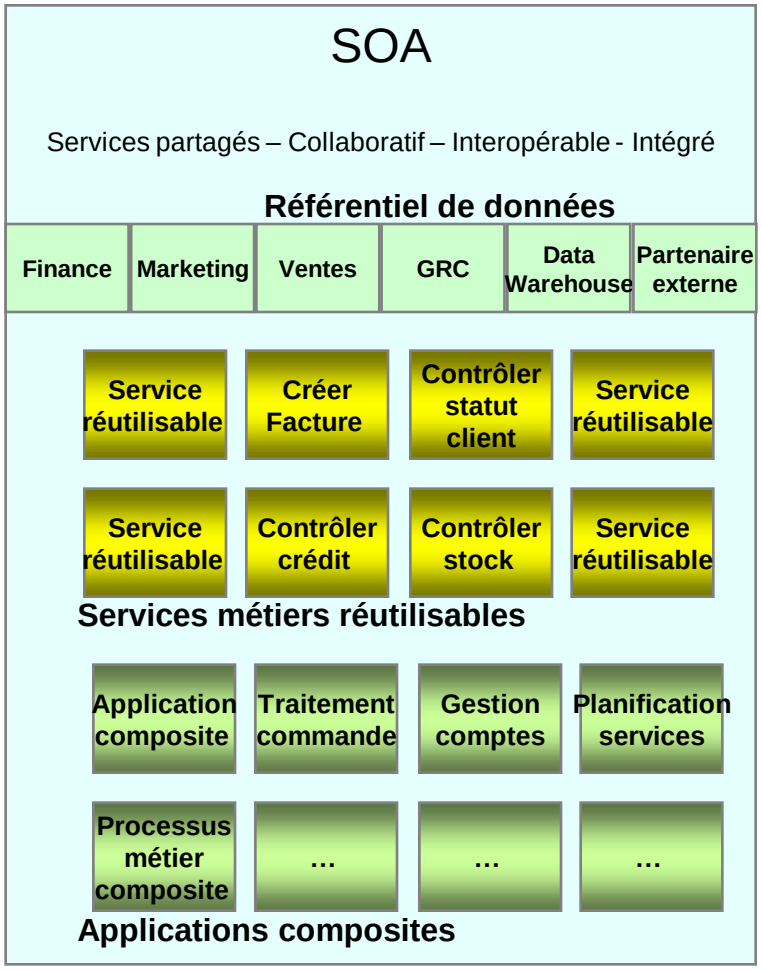
Architectures de traitement

Exemple de modèle générique pour un service

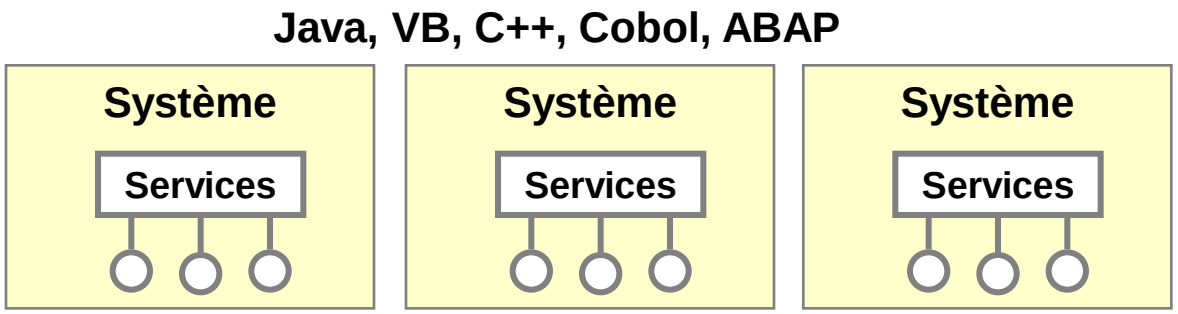


Architecture Orientée Services





SOA : Principes et composants



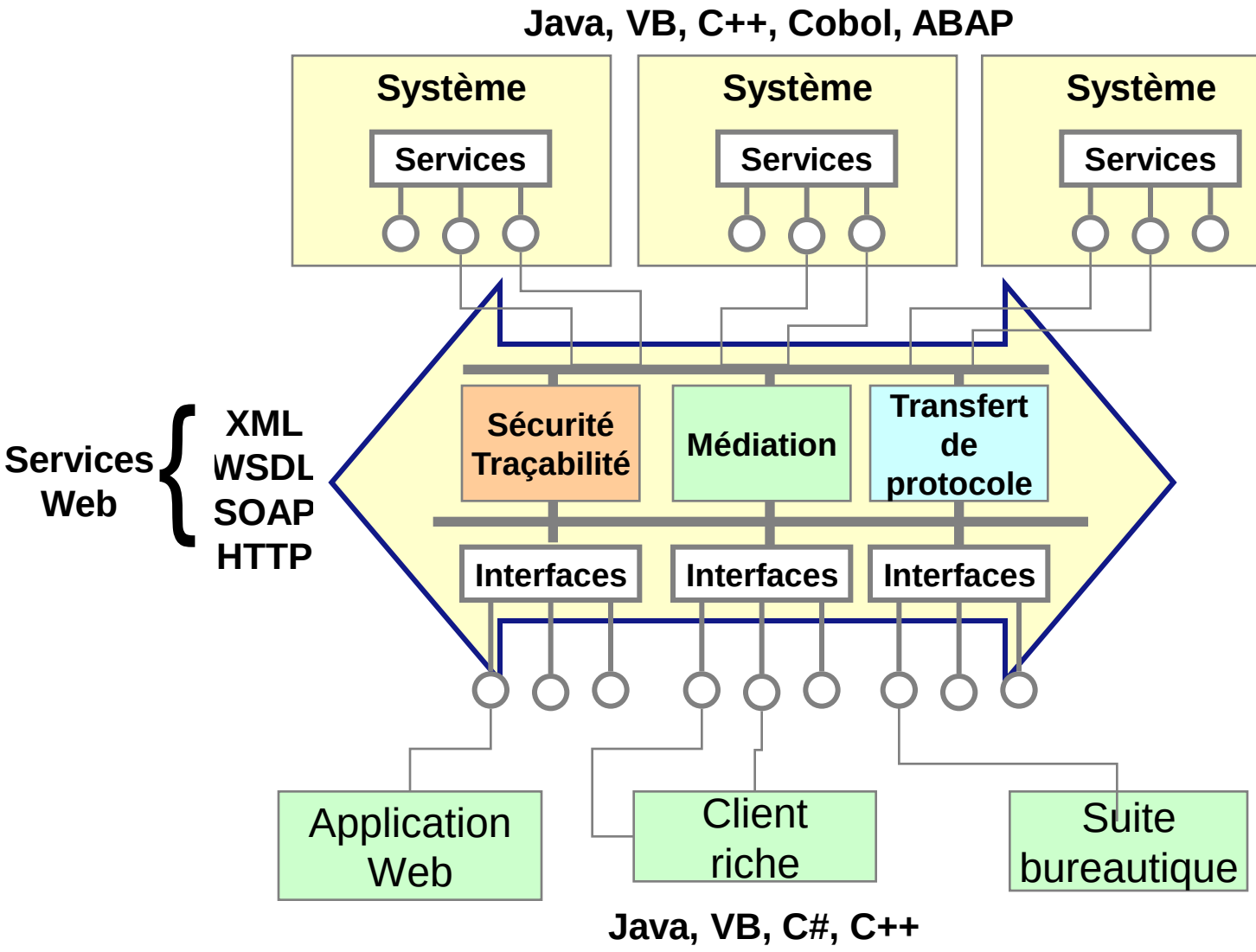
Application Web

Client riche

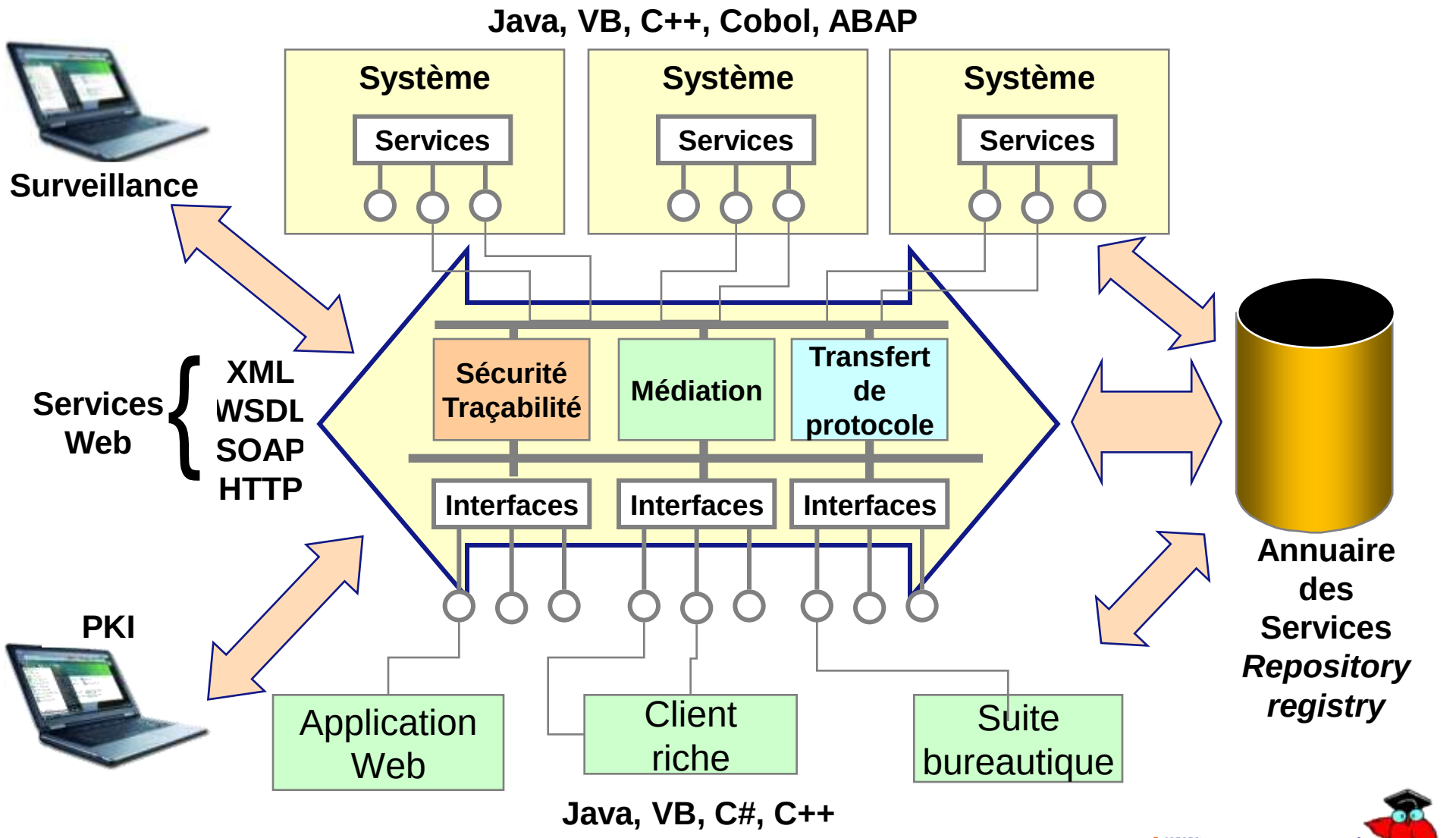
Suite bureautique

Java, VB, C#, C++

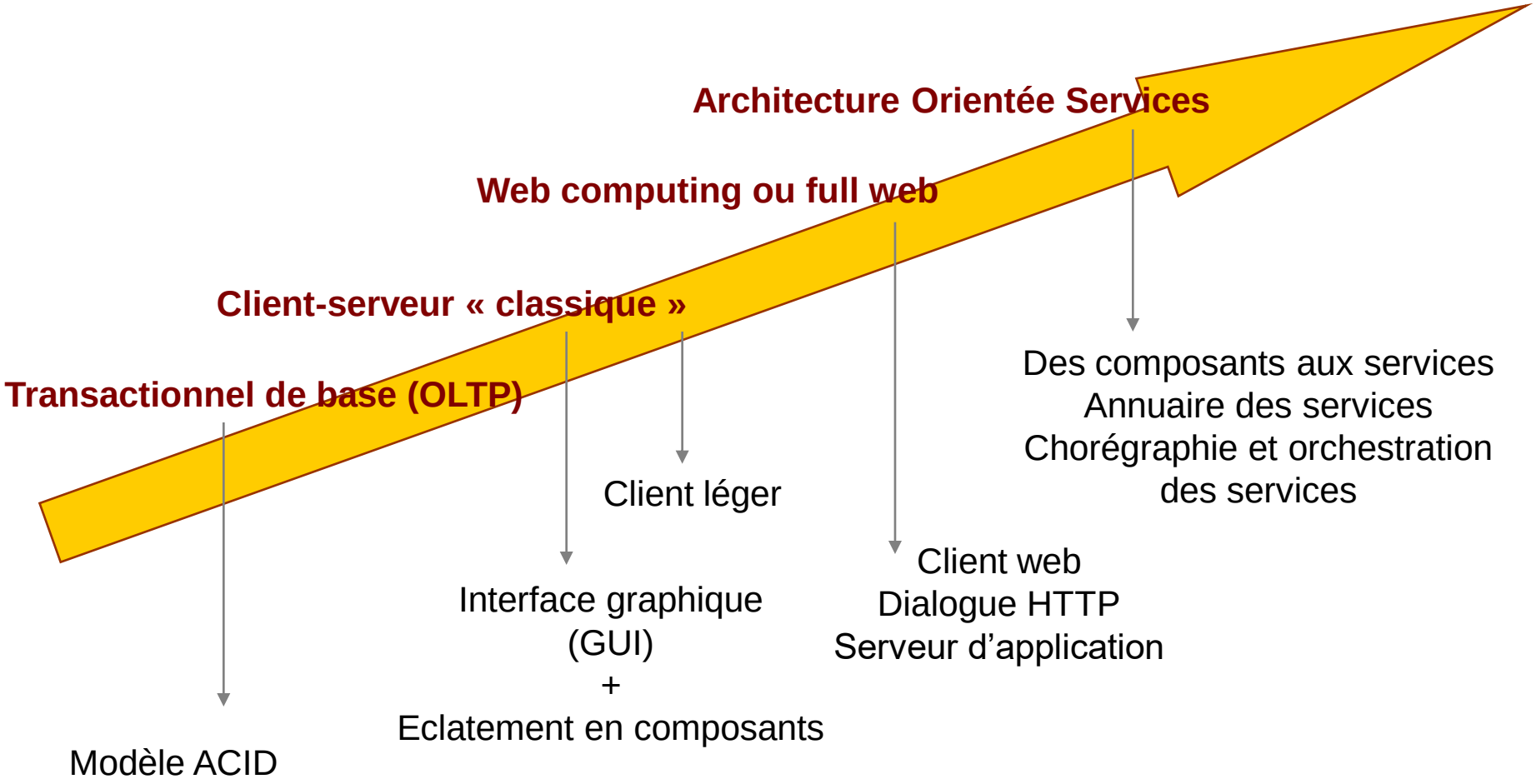
SOA : Principes et composants



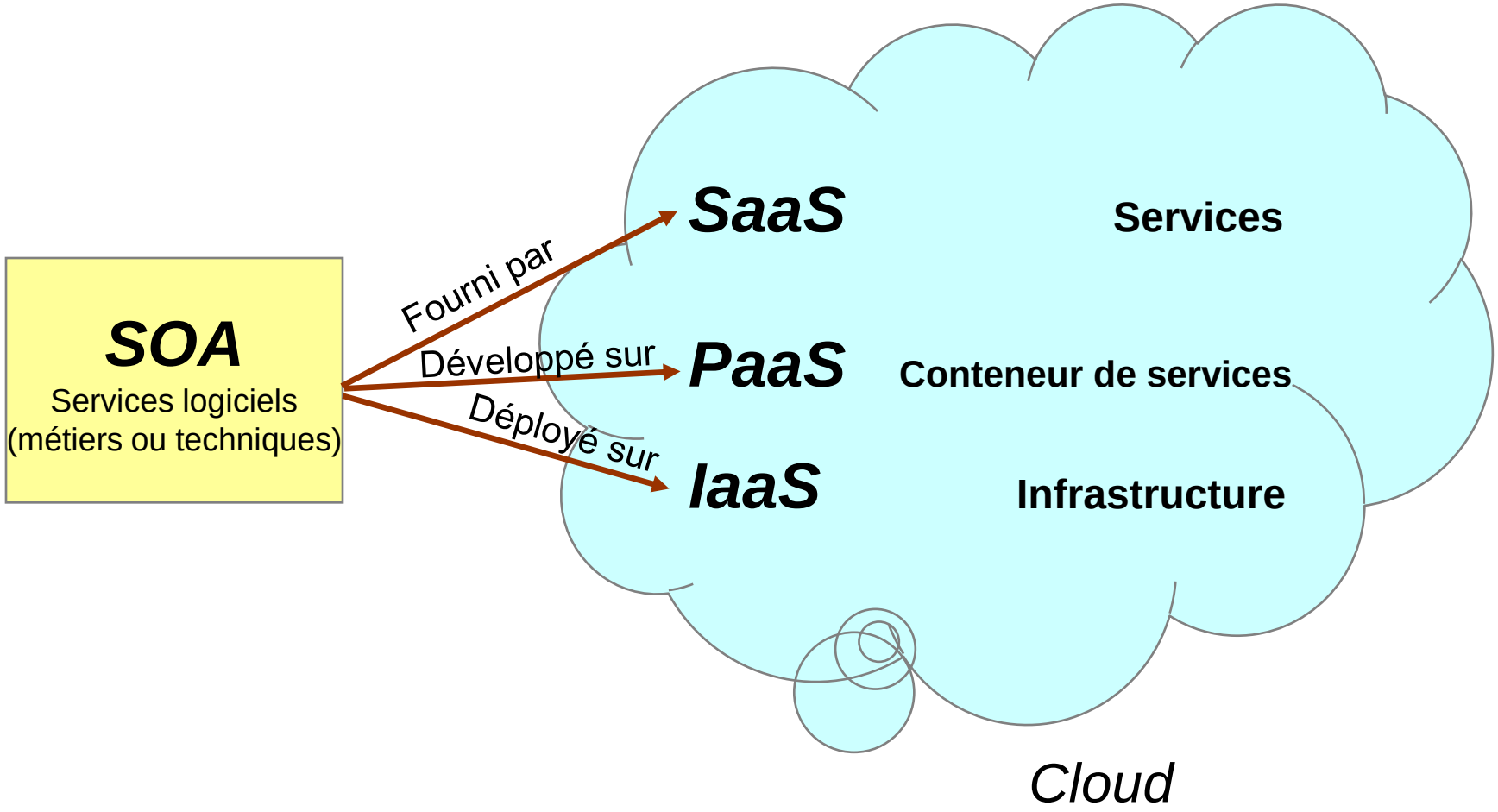
SOA : Principes et composants



Récapitulatif



SOA + Cloud


SOA
Services logiciels
(métiers ou techniques)
Fourni par
Développé sur
Déployé sur
SaaS
PaaS
IaaS
Services
Conteneur de services
Infrastructure
Cloud

SOA + Cloud

- *SOA* pour assembler de manière flexible les services internes à demeure et les services externes du *cloud;*
- *Cloud* en tant que fournisseur de ressources de type “services”, à la demande, de façon souple, grâce à l’usage des technologies Internet;
- *SOA* + *Cloud* comme support de l’industrialisation.