Web 2.0 et clients riches

Panorama et outils

SOMMAIRE

1. Introduction: L'evolution du Web
2. Le Socle technique du Web
   - Les protocoles du Web
3. Client lourd, client Léger
   - Architectures client/serveur multi-tiers
4. Client Riche
5. Les clients riches RIA
6. Les clients riches RDA

Introduction

L'evolution du Web, le Web 2.0

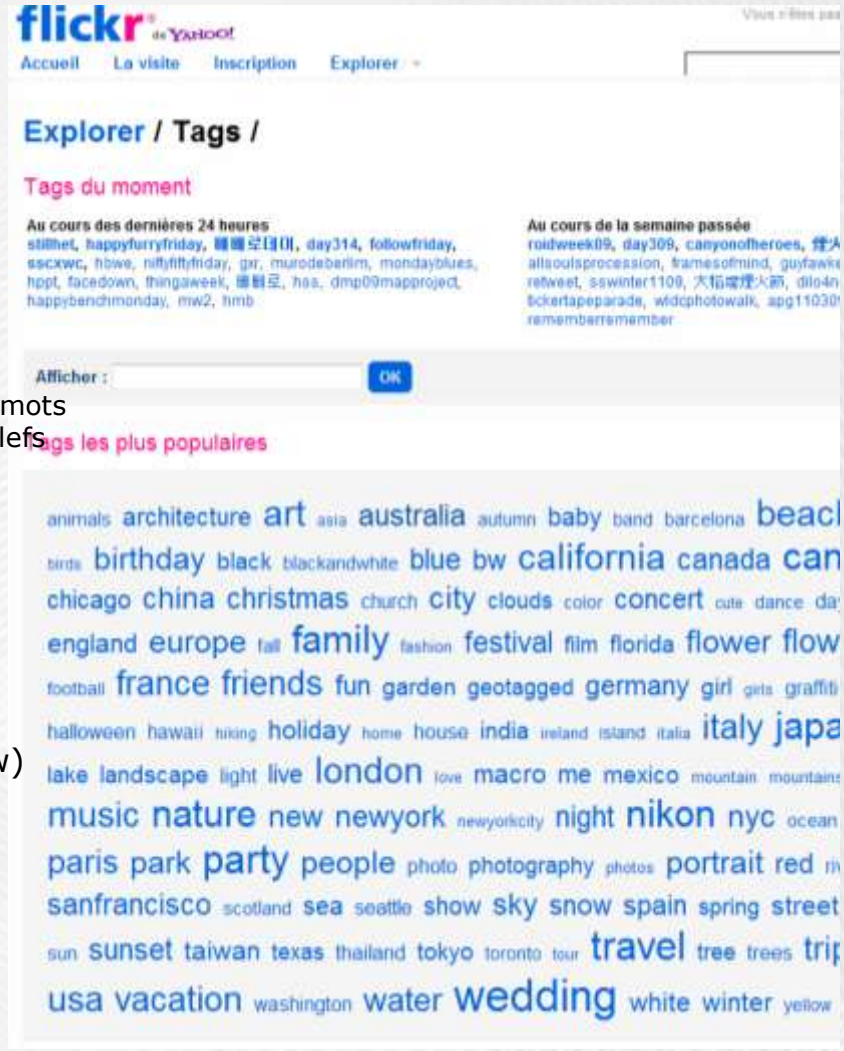
Le web 2.0 c'est quoi?

- Une évolution majeure du Web et de ses pratiques
  - Le Web passe d'une collection de sites web à une plate-forme informatique à part entière qui fournit des applications aux utilisateurs
- Une expression « marketing » à la mode
  - lancée par Dale Dougherty de O'Reilly dès 2003
  - qui a fait du buzz à partir de 2005 avec l'article fondateur « What Is Web 2.0 » lors d'une conférence
- **L'idée:** les pages web permettent aux internautes d'agir sur le contenu et la structure des pages
  → mais aussi d'interagir entre eux
  - Sans connaissances techniques
  - La Simplicité doit être la règle (pages dépouillées et ergonimie)

Ref: http://oreilly.com/web2/archive/what-is-web-20.html
Version francaise: http://web2rules.blogspot.com/2006/01/what-is-web-20-par-tim-oreilly-version.html

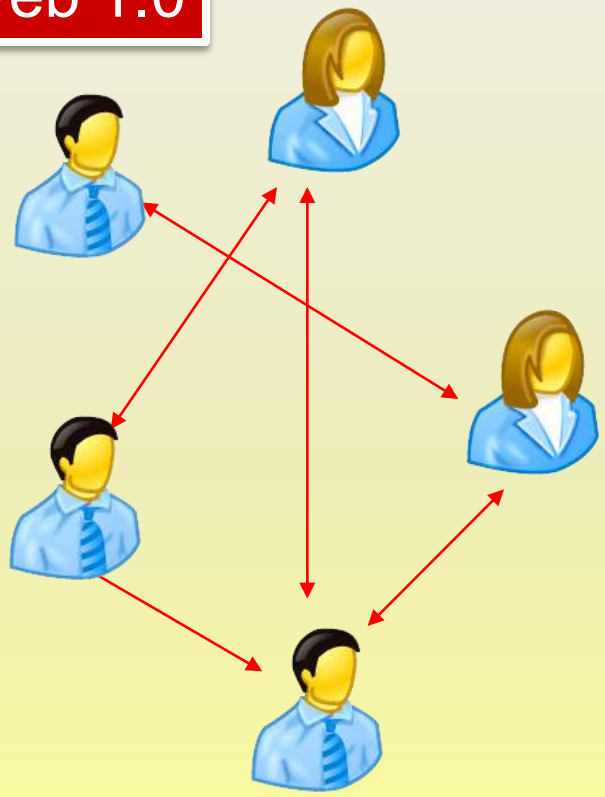
Le web 2.0 c'est quoi?

- RSS (Rich Site Summary)
- Blogs & wiki
  - Wikipedia!
- start pages (RSS, calendrier,notes..)
  - Netvibes
  - My Yahoo!, iGoogle, Microsoft Live
- del.icio.us: bookmarks (signets)
  - Partage de bookmarks
  - classer selon le principe de **folksonomie** par des mots clés (ou tags) et ainsi créer des nuages de mots clefs
  - http://delicious.com/tag/
- Flickr, stockage massif de photos
  - Le classement se fait, non en rangeant la photo dans une arborescence, mais en lui associant un ou plusieurs mots-clés
  - http://www.flickr.com/photos/tags/
- social networking
  - MySpace, Facebook
  - Twitter(3 ans), WAYN (where are you now)
  - hi5, last.fm
  - LinkedIn, Viadeo, Trombi.com
- Video
  - YouTube, dailymotion
- Applications
  - Google Agenda, docs, RTM...



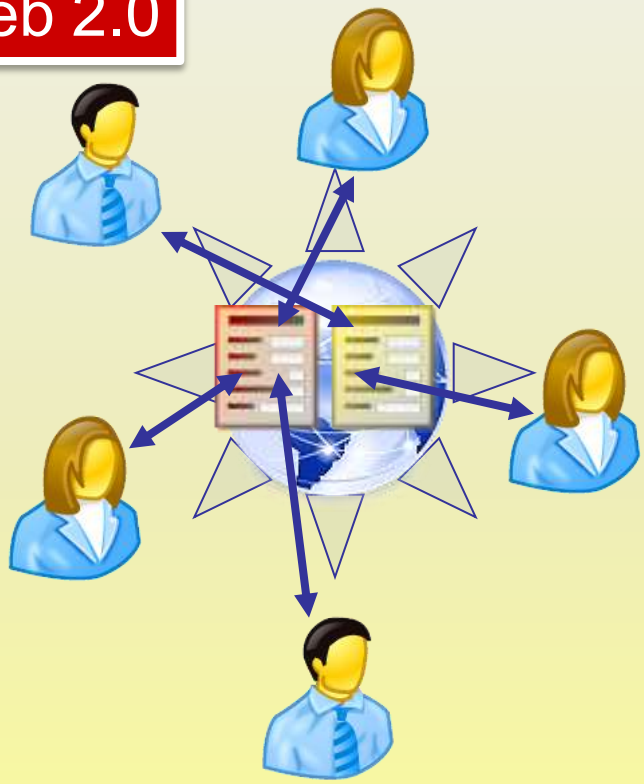
Le mode de communication a changé..

Web 1.0



→1 vers 1, ou en cercle fermé “privé”
Voix, IM, e-mail, listes de mail, doc word par mail…

Web 2.0



→N vers N, tout est Public!
Facebook, twitter, Blogs, photos en lignes, google docs...
(On s’adresse à des applications)

Comparaison: avant/après

Web 1.0 → Web 2.0

DoubleClick → Google AdSense
Ofoto → Flickr
Akamai → BitTorrent
Britannica Online (universalis)→ Wikipedia

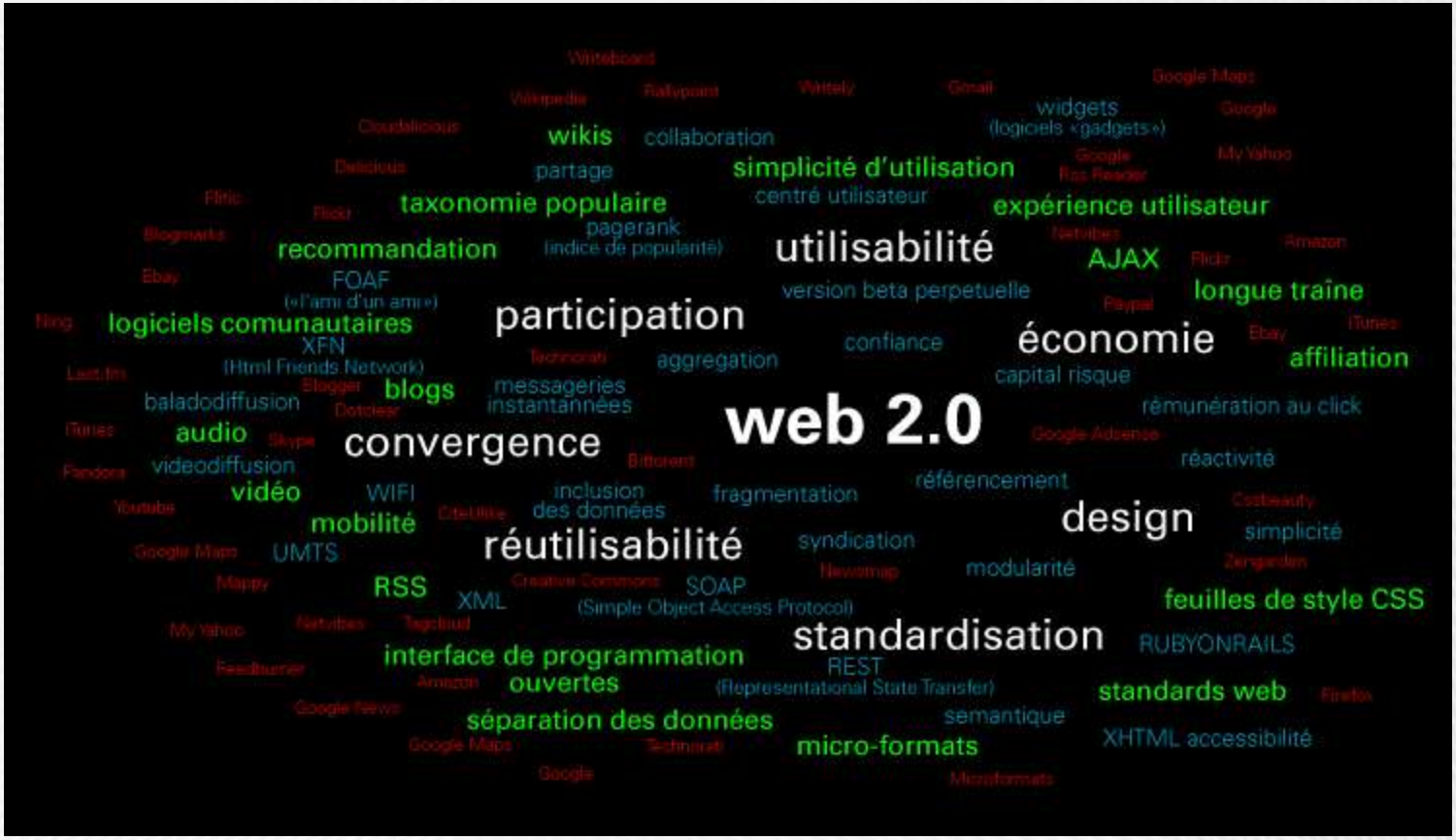
Site web personnel → blogs
pages vues → coût au clic
publication → participation

Système de gestion des contenus (CMS) → Wikis
Répertoires/ arborescence (taxonomie) → tags (folksonomie)
rigidité du contenu → syndication de contenu(souscrire)

Changements en profondeur
Information classique et campagne de presse → buzz
(exemple twitter et JSark)

Le web 2.0 un concept fumeux?

- Le nuage du web 2.0

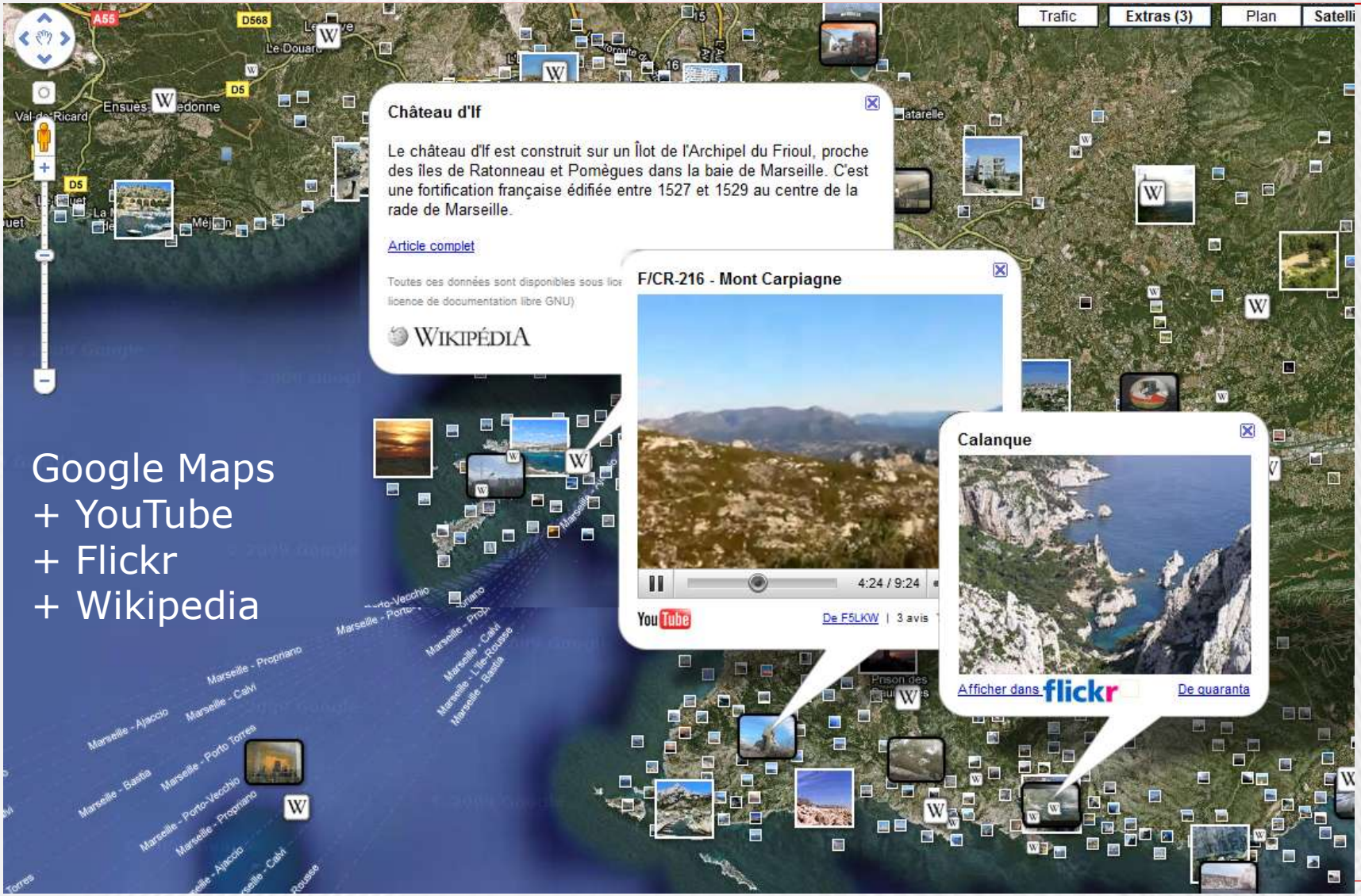


(Origine: Dale Dougherty /Tim O'Reilly)

Le web 2.0 c'est quoi?

- Les 7 principes du Web 2.0 :
  - Le web devient une plateforme
    - Il fourni des applications
    - Il fournit des Services web (des fonctions accessibles à des programmes)
      - Exemple: Utiliser une carte Google Map ou Mappy dans les pages jaunes
  - Tirer partie de l'intelligence collective
    - Chacun apporte un contenu (blog, wiki, réseaux sociaux, rss intégré)
      - Exemple: Le web collaboratif (Wikipedia)
    - "utilisation" des interactions des usagers pour améliorer l'offre
      - Exemple: Google et son utilisation des liens, Folksonomie elle meme
    - et profiter de la sagesse des foules
  - La puissance est dans les données
    - Le rôle central des données (Bases de données spécialisés, contenus & flux RSS)
    - Les Services web permettent de les combiner pour créer de nouvelles applications (mash-up)
    - Du coup la propriété et le contrôle des données devient critique (Google!)
  - Services « prêt à consommer »
    - La fin du logiciel en tant qu'objet
    - Toujours accessible: personne ne se demande quelle version de Google il utilise
  - Des modèles de programmation légers /simplifiés
    - Mode « Beta » perpétuel
    - Les usagers sont participants, Ils sont observés et ce qu'ils n'utilisent pas disparaît dès le prochain cycle de modification du service
  - Le logiciel se libère de l'ordinateur
    - Gmail, premier Logiciel Web. Mais aussi Google Docs, Zimbra, Jamendo
  - Enrichir les interfaces utilisateurs

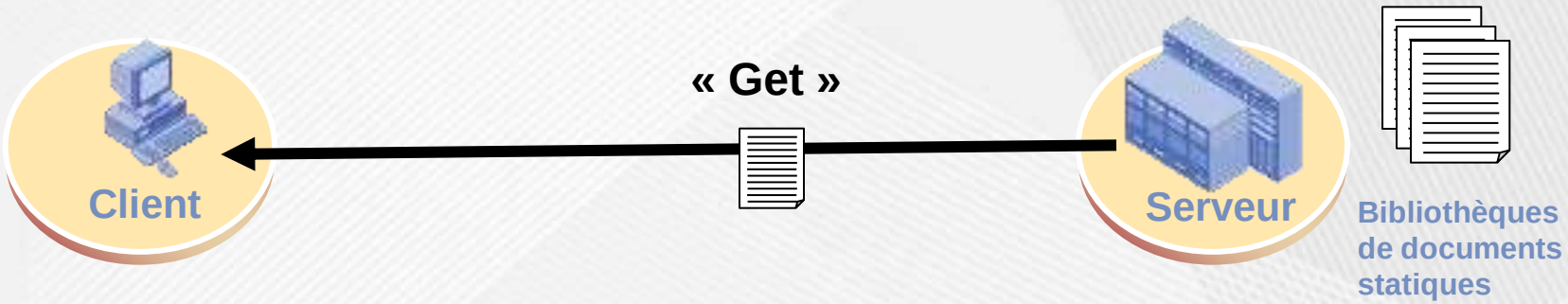
Exemple de « mash-up »



Du coup, le web 1.0 c'est quoi?

- Initialement (jusqu'à 1995 environ), Le web ne se composait que de pages web statiques
  - rarement mises à jour
    (car connaissances techniques requises)
  - Utilisant le concept nouveau d'hypertexte
  - Contenu principalement textuel, enrichi avec des images

Echanges HTTP



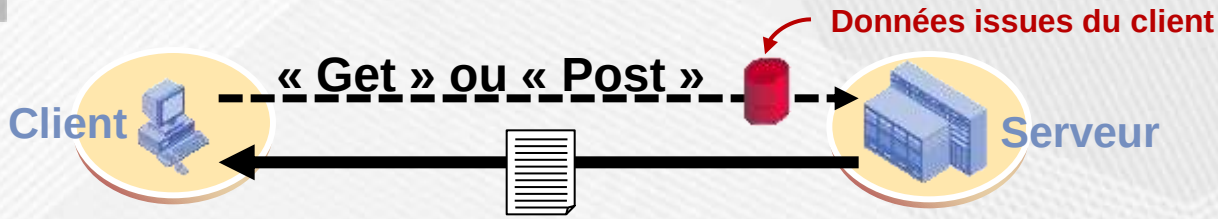
Entre les 2? le web « 1.5 »

- En 1993, le World Wide Web (WWW) etait encore tout petit, mais en pleine explosion
- Le besoin s'est fait sentir de permettre d'appeler des executables pour amener un peu d'interactivité
- 1993, Création de la première spec CGI
  - Common Gateway Interface
  - littéralement « Interface passerelle commune »
- CGI permet
  - D'executer un programme sur le serveur
  - de passer des paramètres au programme
  - Dont il peut tenir compte pour générer des pages web
- CGI a directement amené l'invention du moteur de recherche
  - la chaîne de caractères contenant les termes recherchés est un exemple classique de paramètre CGI

Entre les 2? le web « 1.5 » (2)

- La première évolution d'importance du Web?
  - solutions se basant sur un web dynamique
  - Un contenu dynamique est nécessairement connecté à une base de données
  - Cad que les contenus renvoyés au client ne sont plus identiques qlqsoit le client
  - Le serveur renvoie des pages web (dynamiques) créées à la volée à partir d'une base de données – Celle-ci évolue
- Le Web ne s'est vraiment développé qu'à partir du moment où il a proposé des contenus dynamiques
  - Exemples: publication d'un catalogue, commandes en ligne, services d'annuaires..
- A ce stage, le web est considéré principalement comme un outil de diffusion et de visualisation de données
- Le web dynamique se compose de pages
  - Contextuelle, interactive, dédiée à l'utilisateur, mises à jour à la volée
  - Connaissances techniques plus ou moins requises pour les mettre à jour (voir le concept de CMS)
  - Où l'esthétique revêt une très grande importance (HTML avancé, CSS, Flash…)

Echanges HTTP



Historique de l'internet et du Web

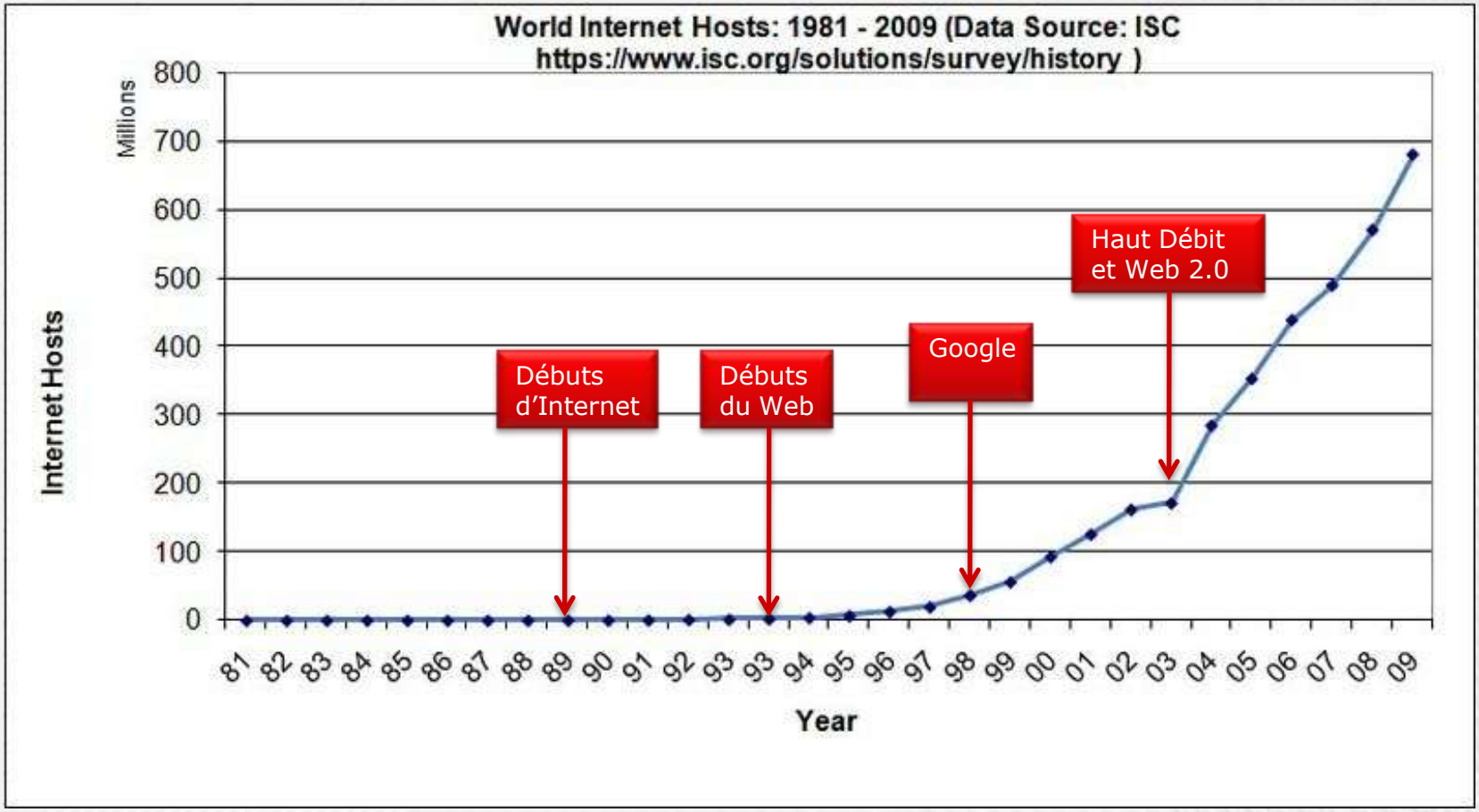
- L'histoire du Web est récente
- Elle est intimement liée à l'evolution
  - des réseaux
  - des systèmes d'exploitations
  - des langages de programmation
- Pourquoi?
  - Car les navigateurs sont maintenant vu
    - Comme un conteneur d'applications
    - Donc potentiellement comme un remplacement (ou une évolution) du bureau du système d'exploitation
      - *(D'ailleurs, Microsoft voire Apple ne s'y est pas trompé)*
      - *Voir aussi les OS en ligne (WebOS) tels que eyeos.org ou zkdesktop (http://zkdesktop.sourceforge.net/) egalement Echo (http://demo.nextapp.com/echo3csjs/) Ou Bindows (http://www.bindows.net/demos/)*
- C'est pourquoi vous les verrez dans le slide suivant

Histoire de l'internet et du Web

- 1972 Langage C devient populaire (en developpement depuis 1969) – et première démonstration d'Arpanet
- **1973 Création d'Unix, réécris en C (pour info Linux apparaît en 1992)** (--Et naissance de Nadir ^_ ^)
- **1983 TCP/IP deviennent les seuls protocoles de l'ARPANET + Création du système DNS → Début d'internet - Et Unix (version BSD) intègre TCP/IP en natif**
  - 1983 MS-DOS, première version à supporter un disque dur et des répertoires (sur IBM PC)
  - 1983 Les réseaux sont locaux, Novell Netware domine dans les entreprises avec son système d'exploitation réseau pour PC
- 1984 (jusqu'à 1988) le CERN (Centre Europeen pour la Recherche Nucléaire) déploie TCP/IP sur ses réseaux internes
  - 1984 création de X-Windows, dès le départ orienté réseaux en client/serveur
  - 1984 lancement Apple Macintosh, avec son interface graphique fenetrée
- 1985 Création de Windows 1.0, simple interface fenêtrée au dessus de DOS, initialement créé sur un Macintosh, echec
- 1986 Novell Netware gère TCP/IP en plus de ses protocoles natifs (IPX)
- 1989 Le CERN ouvre son réseau à l'extérieur en TCP/IP
- **1990 Invention du WWW au CERN par Tim Berners-Lee qui crée le premier Web Serveur et Web Browser en proto (et aussi création de URL et HTTP)**
  - 1990 sortie de Windows 3.0 la première version à connaître un large succès
- **1991 Véritable début du Web, qui devient public grâce à Robert Cailliau (CERN)** (…et 1ere spécification CORBA)
- 1993 le CERN libère les droits des technologies du web (dans le domaine public)
- 1993 création du protocole CGI
- **1994 premier moteur de recherche tel qu'on les connait: WebCrawler (il y en aura très vite une vingtaine)**
  - 1994 première version réseau de Windows (3.11), au dessus de IPX ou netbeui (Novell)-Puis de TCP/IP
- 1994 Lancement de **Netscape Navigator**, premier navigateur commercial distribué à grande échelle (Windows, Unix, Linux et Mac)
- 1995 naissance d'Apache (collection de patchs du serveur NCSA HTTPd " a patchy server")
  - 1995 Windows 95 -Première version de Windows qui n'est pas un habillage de MS-DOS- et apparition de Internet Explorer 1.0 (en pack additionel payant)
  - 1995 Première version de Java
  - 1995 Allaire Cold Fusion version 1.0
- **1996 Netscape crée JavaScript (et le DOM) dans son navigateur Web Netscape Navigator 2.0 (premières animations)**
- 1997 Netscape et Microsoft crée DHTML (Dynamic HTML), qui permet de modifier le DOM d'une page HTML avec javascript
  - 1997 PHP 3, première version répandue et utilisable de PHP
  - 1997 première version de Macromedia Flash
- 1998 ASP (en option dans Windows NT 4.0), concurrent de PHP et de Cold Fusion
- **1998 apparition de Google – Mort de Netscape Navigator (qui devient Mozilla, version opensource)**
- 1999 JSP JavaServer Pages, la réponse de Sun à PHP et ASP
- 1999 Microsoft crée the XMLHTTP (requêtes aynchrones) dans Internet Explorer 5 **(Ajax est deja là!)-** Apparition du format rss
- 2000 Google devient prédominant garce au pagerank (elimine les autres moteurs de recherches)
  - 2000 apparition des sessions dans PHP avec PHP4
- 2001 Wikipédia - le vrai début du Web participatif
- 2004 création de Gmail, première grosse utilisation d'Ajax (Gmail utilise Java Servlets sous Linux)
  - 2004 Première version de **Mozilla Firefox**
- 2005 naissance du terme Ajax (asynchronous JavaScript and XML[)
- **2005 naissance du terme Web 2.0 avec l'article fondateur « What Is Web 2.0 »**
- 2007 apparition de Google Gears, avec ses fonctionnalités "Offline" (BD et serveur Local) pour les navigateurs
  - 2007 Silverlight, destiné à concurrencer Flash
  - 2007 Adobe publie AIR (Flash & Ajax), RDA
- 2008 Google Chrome, basé sur Firefox, orienté Web 2.0, amélioration de la performance du rendu Javascript, permet l'accès à des applications web hors du navigateur

Histoire du Web

- Quand cela a t'il commencé?



Le socle technique du Web

Les protocoles du Web

Le socle technique du Web

- Les protocoles associés au Web sont **standard**
- Le Web s'appuie sur:
  - **HTTP**: le transport
  - **URL**: l'adressage des documents
  - **HTML**: la description du contenu et de sa mise en forme
  - **CGI**: la passerelle autorisant l'interactivité

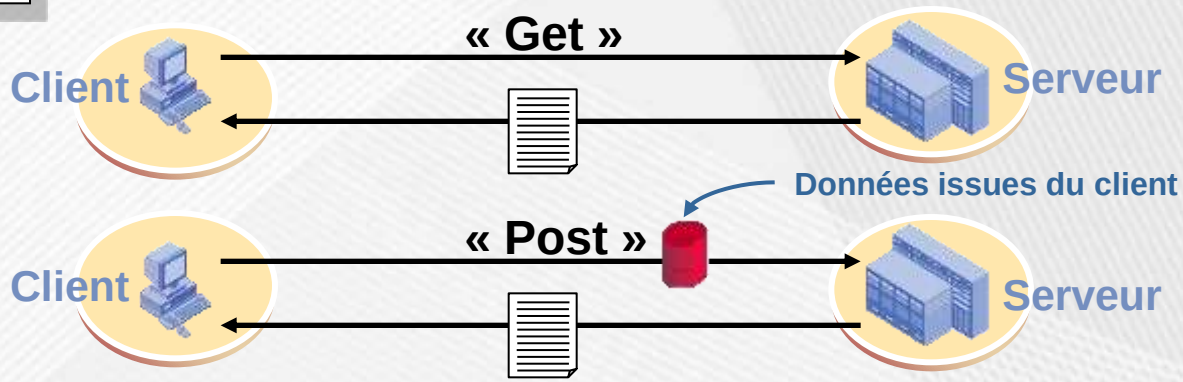
Le socle technique **URL**/HTTP/CGI

- URL (Uniform Resource Location) est un moyen de désignation universel de l'emplacement de certains fichiers.
  - Indépendant de l'OS
  - Indépendant de la machine
- Format général:
  - protocole://machine[:port]/[chemin_du_fichier]
- En particulier
  - protocole :ftp, http, autres (nntp, gopher..)
  - Machine: IP ou nom
  - Port est optionnel, 80 par defaut
  - Le chemin est optionnel, il doit normallement désigner un fichier (sinon on dit que l'URL est incomplète)

Le socle technique URL/**HTTP**/CGI

- HTTP: les principales méthodes
  - GET URL : demander le contenu de la ressource
  - POST URL: envoie de données vers une application
- HTTP: le transport
  - Architecture C/S, mode « Pull »
  - Connections courtes, « Sans état » (stateless)

Echanges HTTP



Exemple d'échange HTTP

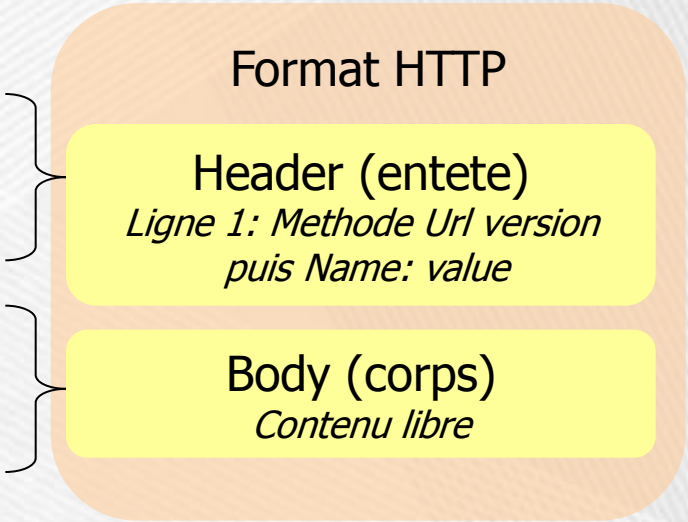
Requête:

```
GET / HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; …
Accept:
    text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/p
    lain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

Demo avec apache et un proxy http

Réponse:

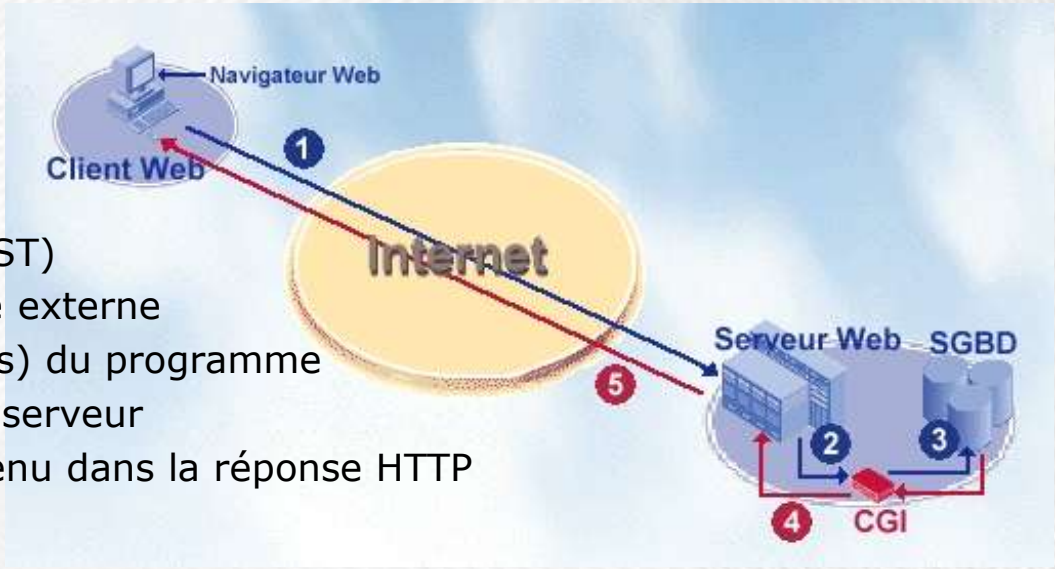
```
HTTP/1.1 200 OK
Date: Sun, 14 Aug 2005 15:10:14 GMT
Server: Apache/1.3.23 (Win32)
Content-Type: text/html

<HTML>
    <BODY><H1>Page</H1><BODY>
<HTML>
```



Le socle technique URL/HTTP/**CGI**

- Common Gateway Interface
  - littéralement « Interface passerelle commune »
  - une interface normalisée utilisée par les serveurs HTTP
  - CGI est le standard industriel
- Protocole d'échange entre serveur Web et application externe
- Mode Synchrone

- 5 étapes :
  1. Requête client HTTP (POST)
  2. Exécution du programme externe
  3. Traitements (sur données) du programme
  4. Renvoie d'un résultat au serveur
  5. Transmission de ce contenu dans la réponse HTTP



CGI: Exemple de requête POST

Requête:

```
POST /cgi-bin/message.pl HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; …) Gecko/20050511 Firefox/1.0.4
Accept: text/xml,…,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: http://localhost:8080/TP1-Messagerie%20en%20CGI/PERL/form.html
Content-Type: application/x-www-form-urlencoded
Content-Length: 117

nom=nadir&email=boussou%40gmail.com&ville=Marseille&sexe=M&age=32
[…]&message=un+petit+message.%0D%0ADeuxieme+ligne%0D%0A
```

Réponse:

```
HTTP/1.1 200 OK
Date: Sun, 14 Aug 2005 15:03:36 GMT
Server: Apache/1.3.23 (Win32)
Transfer-Encoding: chunked
Content-Type: text/html

51
<HTML>
<BODY BGCOLOR=WHITE>
<CENTER>
<H1>Message recu</H1>
</BODY>
</HTML>
```

Références

- Les slides précédents sont une vue simplifiée des protocoles décrits (HTTP, CGI, URL)

- les documents décrivant ces standards sont des « RFC »
  - RFC = « Request For Comment »
  - RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0
  - RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1
  - RFC 1738 - Uniform Resource Locators (URL)
  - RFC 1808 - Relative Uniform Resource Locators
  - RFC 3875 - The Common Gateway Interface (CGI) Version 1.1

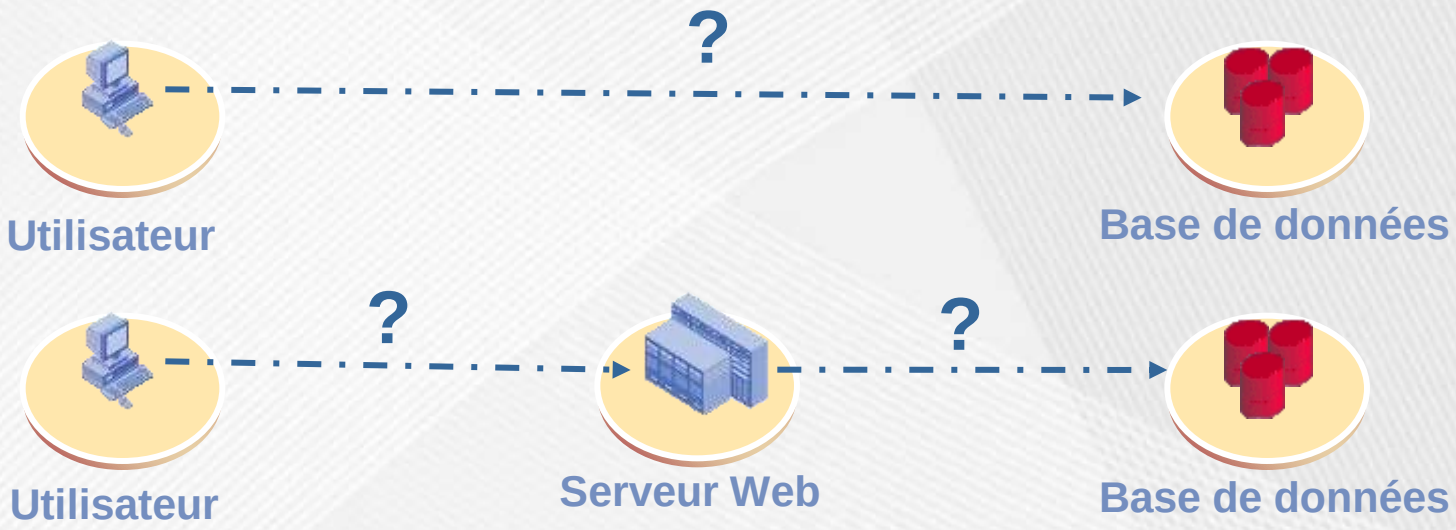
- a voir sur http://www.faqs.org/faqs/ par exemple

Client lourd, client Léger

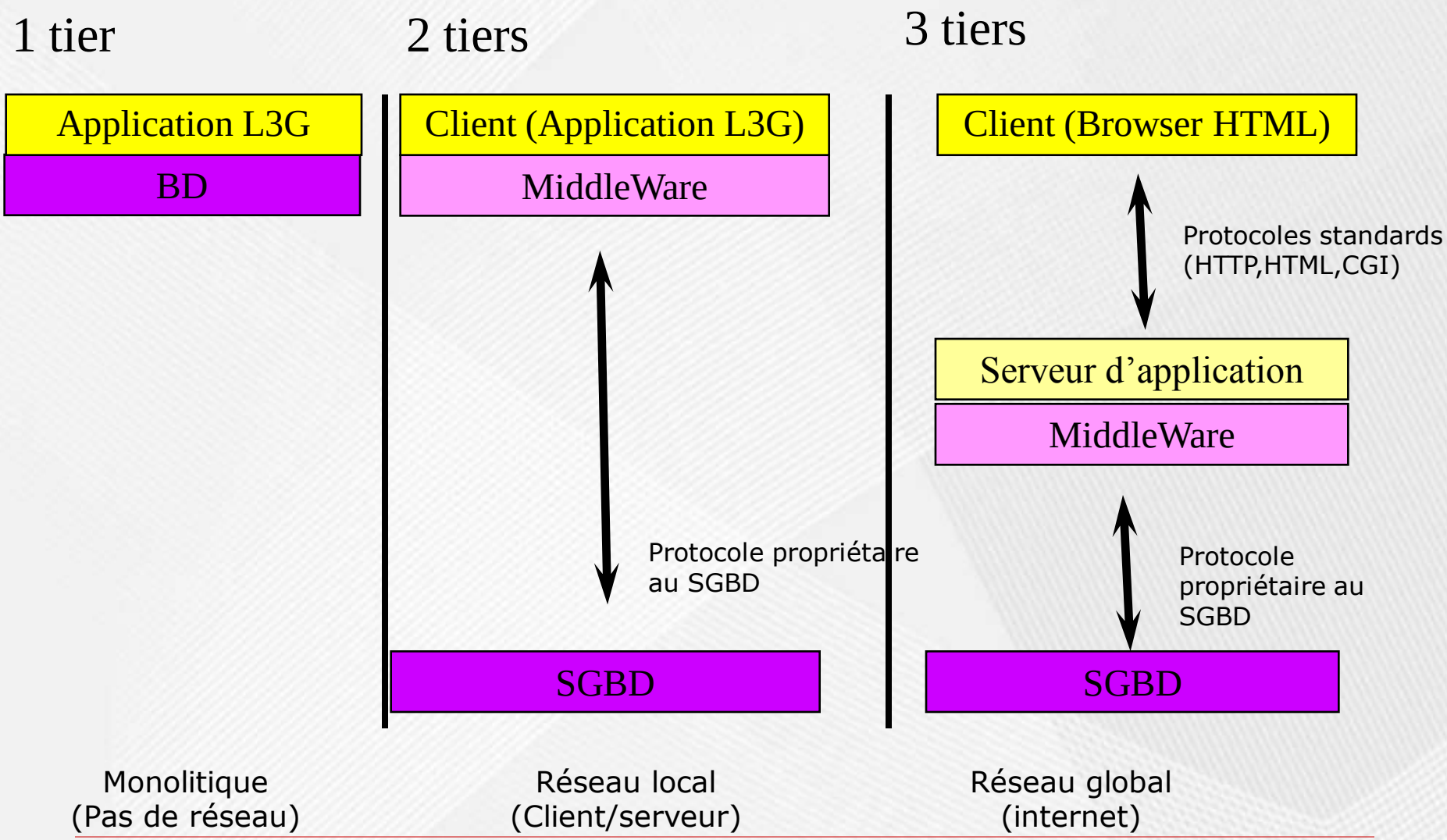
Architectures client/serveur multi-tiers

Evolution des architectures client/serveur

- Les attraits du multi tiers
  - Pourquoi l'industrie s'est orienté vers ces technos?
  - Au point de les utiliser en interne (intranets)?
  - Au point même de redévelopper des applications existantes?

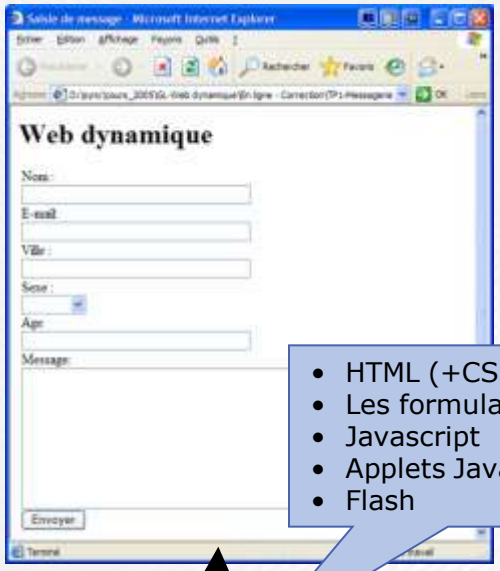


Modèles d'architectures



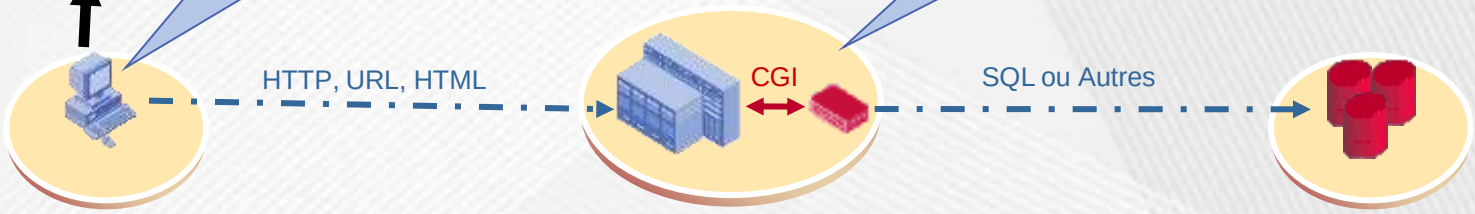
Les langages des applications Web

Seuls les scripts serveurs offrent une vue complète (code tier client/tiers serveur) dans un même fichier, c'est ce qui fait leurs succès



- HTML (+CSS)
- Les formulaires HTML
- Javascript
- Applets Java
- Flash

- Programmes CGI
  - Perl et C
- Les scripts serveur
  - ASP
  - PHP
  - ColdFusion
  - JSP



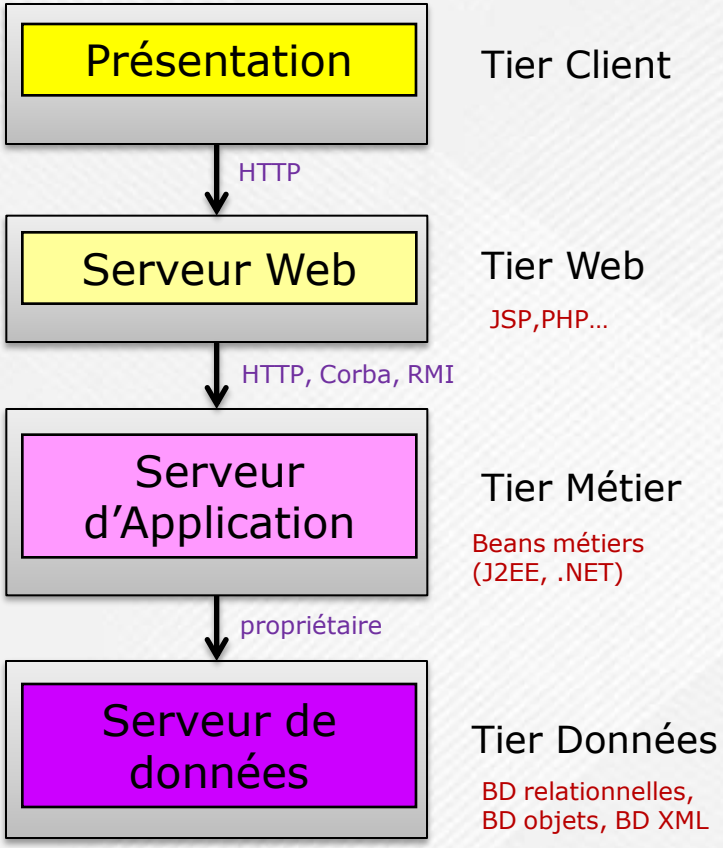
Navigateur

Serveur Web

Base de données

Architectures client/serveur

Généralisation avec les technologies de serveur d'applications



Rôles répartis entre les tiers:

- IHM de présentation des données
- Logique de présentation
- Logique métier
- Contrôle d'accès
- Gestion de la bande passante

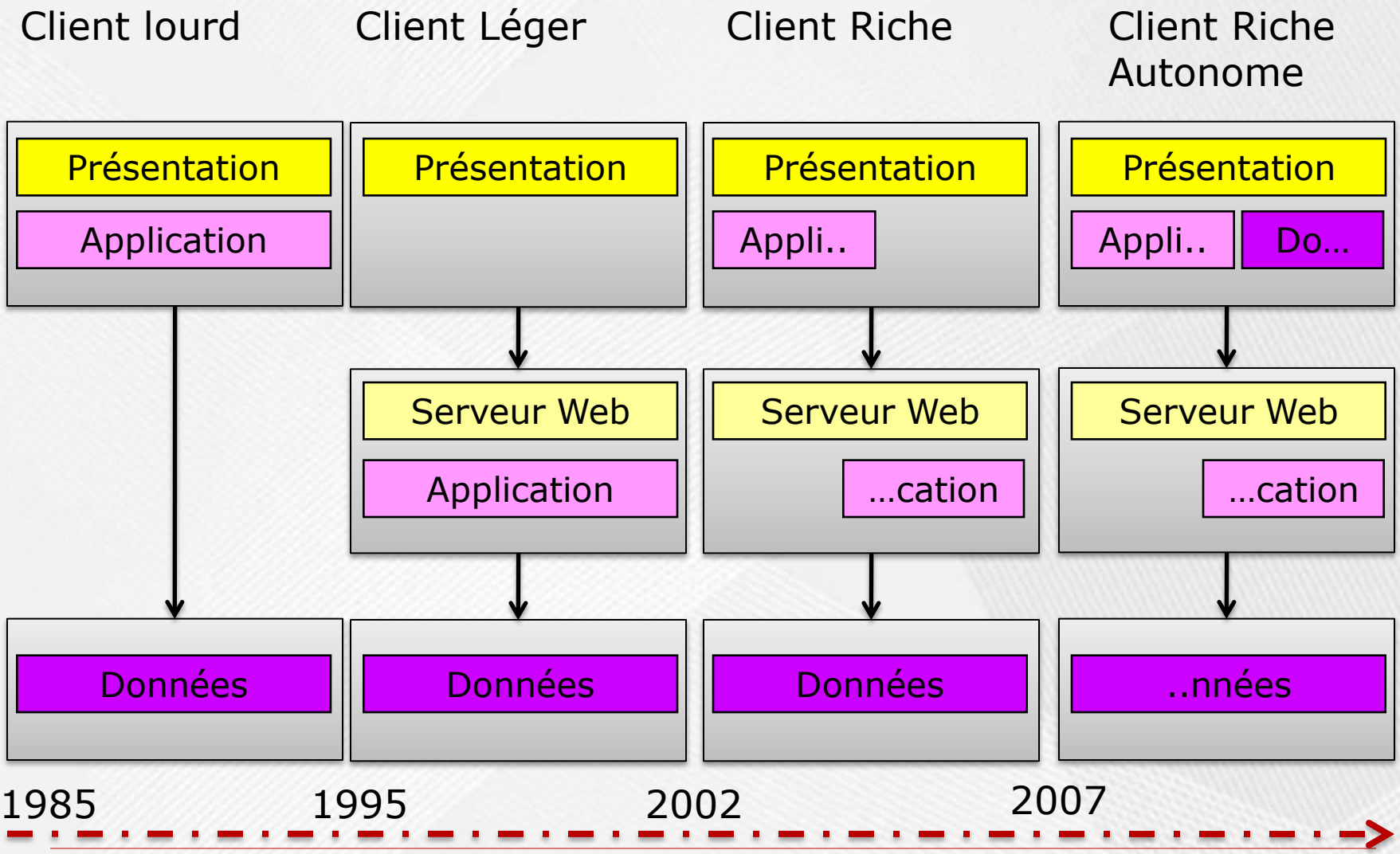
la chaine de développement Web

- Une application Web typique sera divisée
  - Une partie côté navigateur (cliente)
  - Une partie côté serveur
- Particularité:
  - La totalité des fichiers de l'application se trouvent sur le serveur
  - Mais les contenus de ces fichier ne seront pas exécutés au même moment (HTML, PHP, Javascript)
  - Ni par le même tier
- Au cours du dev, une réflexion sur la position d'une fonctionnalité doit être menée (le tier)
  - Exemples:
    - la validation d'un formulaire
    - Une calculette euro
    - afficher l'heure ?

la chaine de développement Web

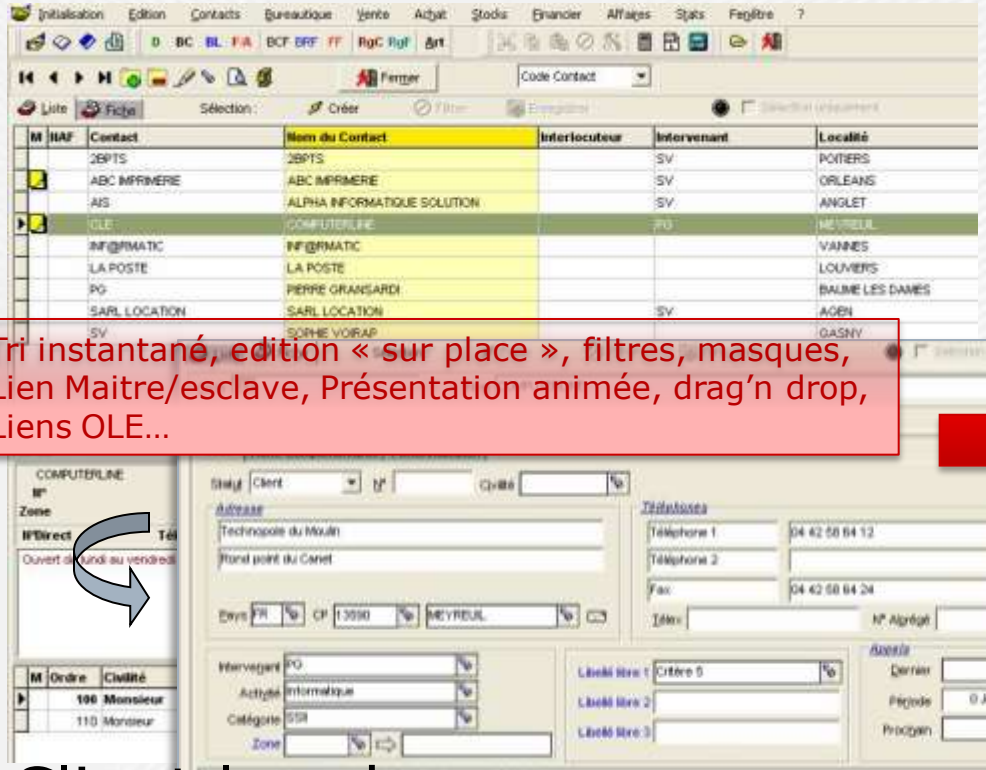
- Exemple de la validation d'un formulaire
  - par exemple vérifier qu'un numéro de tel ou un email a un format correct)
- Réflexion sur la position (le tier) à mener
  - Vérifier coté client?
    - utiliser JavaScript
    - Avantage : rapidité, pas d'accès réseau
    - Inconvénient : javascript peut être désactivé sur le navigateur
  - Vérifier coté serveur?
    - utiliser CGI/PHP/ASP/JSP/CF sur le serveur
    - Avantage : impossible de désactiver la vérification par le client
    - Inconvénient : accès serveur (aller-retour), pas de vérification pas à pas sur le client
  - Vérifier des deux cotés?
    - Avantage : rapidité, vérification pas à pas sur le client, robustesse (vérification au moins sur le serveur)
    - Inconvénient : double codage

Architectures client/serveur



Client léger : une régression!

- Le passage du Client lourd au client léger
  - une amélioration archtecturelle
  - **mais une nette régression pour l'utilisateur !**



Tri instantané, edition « sur place », filtres, masques, Lien Maitre/esclave, Présentation animée, drag'n drop, Liens OLE…

…Pas grand-chose:
Plusieurs pages, temps de réponses, contrôle simples

Nom	Groupe	Email	Portable
Nadir	Ami	Nadir@iut.fr	06 66 66 66 66
Ian	Ami	ian@iut.fr	06 77 77 77 77
2 contacts			

Client lourd

Client léger

Client lourd

- Une application client lourd est **déployée** et **exécutée** sur le système d'exploitation de chaque utilisateur
- Utilisabilité:
  - une interface graphique riche: composants evolués
  - capacités de traitements élevées
  - Interactivité forte (retour instantané)
- Langages
  - Visual C++, Visual Basic, Delphi, Java, …
- Exemple typique
  - MSN
  - Outlook/Thunderbird

Client léger

- Une application client léger
  - accessible via un navigateur, pas de déploiement
  - Executé à distance : La logique métier est présente sur le serveur
- Utilisabilité
  - interface graphique limitée, composants basiques
  - Interactivité limité (mode « page », requete/réponse, 1 réponse = 1 page)
  - L'utilisateur attend
  - Il peut être dérouté par la nouvelle page
- Langages
  - HTML, scripts serveur, javascript
- Exemple typique
  - WebMessenger
  - Hotmail

Comparaison et limites

	Client lourd	Client léger
Richesse des composants	**+** (tri, filtre, masques, arbres...)	**-**
Réactivité	**+**	**-**
Maintenance	**-** Déploiement cher	**+** Pas de déploiement requis
Exploitation des ressources du client	**+** Integration OLE, interaction fichiers et OS (drag'n drop etc)	**-**
Sécurité	**+**	**-**
Protection du logiciel	**+** Décompilation	**-** (sur le serveur et sur le client)
Utilisation sans réseau	**+**	**-**
Asynchronisme	**+** Push & pull (événements)	**-** Pull uniquement
Poids sur le client	Consommateur de Mémoire et de CPU	Poste client très léger donc peu coûteux
Portabilité	**-** (installation nécessaire)	**+** (mais différence navigateurs)

On a enrichi…Le serveur

Les premières evolutions du Client léger otn concerné le serveur

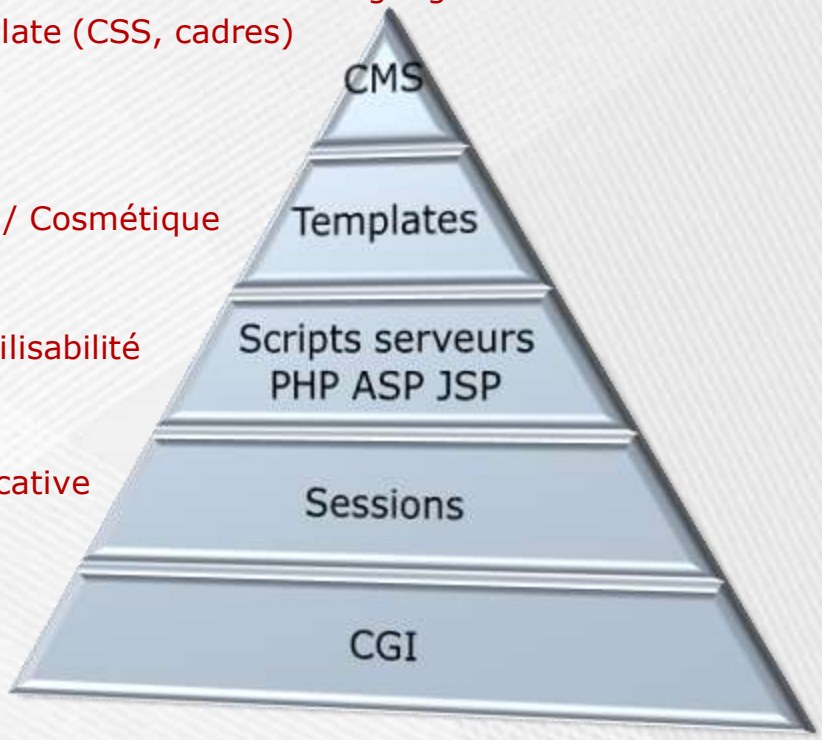
Gestionnaire de contenu: code produisant des sites sans programmation séparant
- contenu (Articles et media associés) et eventuellement multi-languages
- Mise en forme (“Skin” du site via un template (CSS, cadres)
- Structure (Navigation du site / Menu)

Séparation logique / Cosmétique

réutilisabilité

Logique & Cohérence applicative

Disponibilité serveur/OS



On est en train d’enrichir Le client

- Ce cours a pour but de présenter les différents types de clients « enrichis »
- Le but ayant été de résoudre toutes les faiblesses du client léger
- En 2008, ce processus est quasiment finalisé

Les clients riches

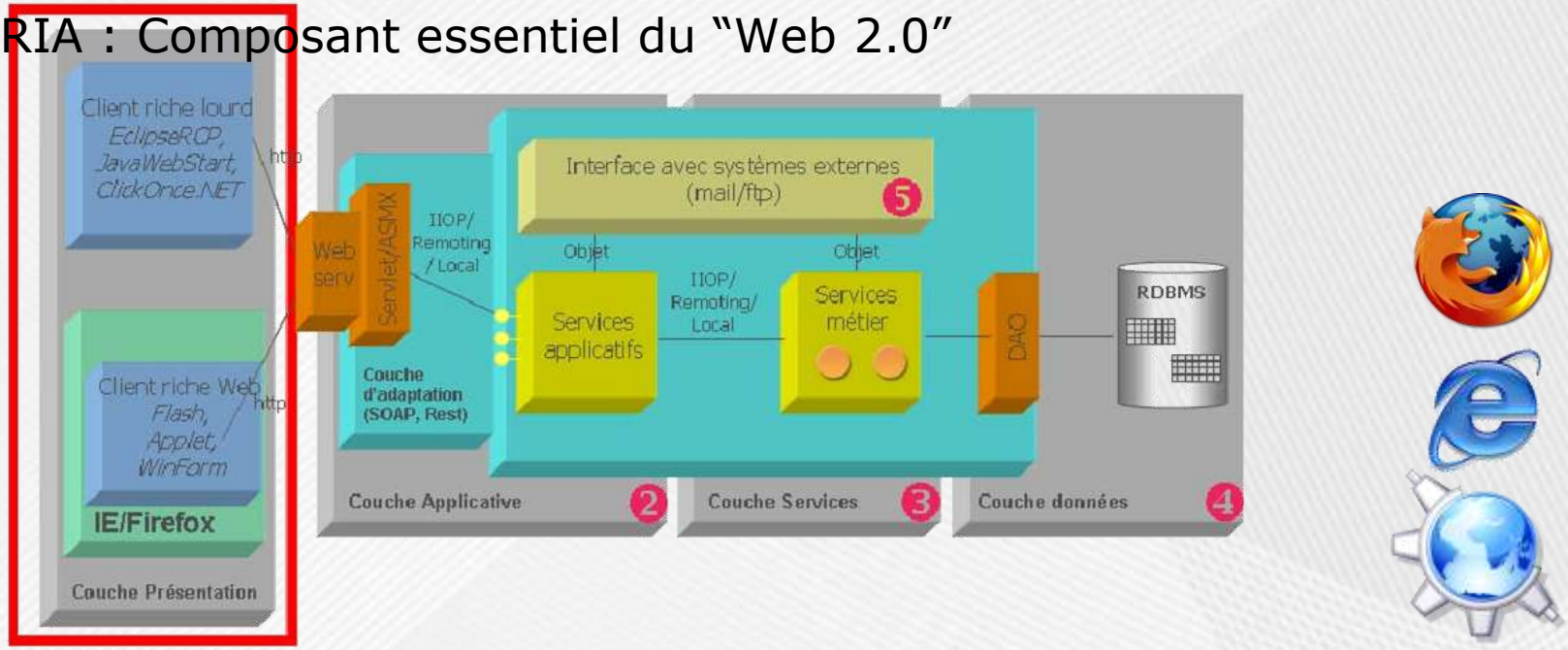
La Partie visible du Web 2.0

Client riche? Définition

- Le Web 2.0 est associé au concept de Client riche
  - le terme client riche est utilisé par de nombreux éditeurs pour pousser leur solution
  - C'est une évolution du client/serveur, pas un produit
- Un client riche, c'est :
  - Une technologie permettant d'améliorer la couche présentation d'une application C/S Web
  - Disposants de composants graphiques de haut niveau (composants liés maitre/esclave, calendrier…)
  - La disparition du développement d'application en mode « page »
  - Facilement déployable et « upgradable »
- L'objectif des clients riches est d'allier :
  - les qualités de déploiement des applications web actuelles (clients légers)
  - avec l'ergonomie des applications de type client lourd

Technologies de clients riches

- on peut classifier les clients riches en deux grandes famille :
  - les « Rich Internet Applications » : améliore le client léger
  - les « Rich Desktop Applications »: améliorer le client lourd
- Ce qui les distingue:
  - la nécessité d'installer (ou pas) un environnement d'exécution sur le poste client (qui se résume à un simple navigateur dans le cas des RIA).
- RIA : Composant essentiel du "Web 2.0"



RDA (Rich Desktop Applications)

- Désigne les clients riches qui s'exécutent directement sur le poste client des utilisateurs.
  - On quitte le navigateur, on revient sur le bureau
  - Ils décrètent la fin de l'utilisation du navigateur web comme « conteneur d'application ».
- S'appuie sur les principes de développement des clients lourds (Java Swing ou Windows Forms)
- tout en automatisant et en masquant les procédures de mise à jour des applicatifs sur les postes clients.
- Du point de vue utilisateur, le client riche ressemble exactement à un client lourd.
- Mais il est nécessaire que ce conteneur soit installé sur le poste qui exécutera l'application
  - On perd l'avantage du un poste banalisé universel
- L'administrateur, quant à lui, se retrouve déchargé des tâches de déploiement et mise à jour.
- Le conteneur fournit un socle
  - d'installation et déploiement
  - de mise à jour
  - Et d'exécution aux applications

Panorama RIA & RDA

- Les clients riches « RIA » (« Rich Internet Applications »)
  - Ajax - Asynchronous Javascript and Xml
  - Flash, Adobe Flex, OpenLaszlo
  - Applets Java, JavaFX (Java en mode script dédié au RIA)
  - (SVG – Scalable Vector Graphics-eliminé de la course)
  - (Silverlight-ou WPF-plugin pour navigateur destiné à concurrencer Flash-eliminé de la course)

- Les clients riches « RDA » (Rich Desktop Applications)
  - Mozilla XUL - Extensible User Interface Language (Livré avec Firefox)
  - Eclipse RCP - « Rich Client Platform »
  - AIR: Adobe Integrated Runtime
  - Sun JavaWebStart
  - Microsoft ClickOnce

RIA vs RDA vs Bureau

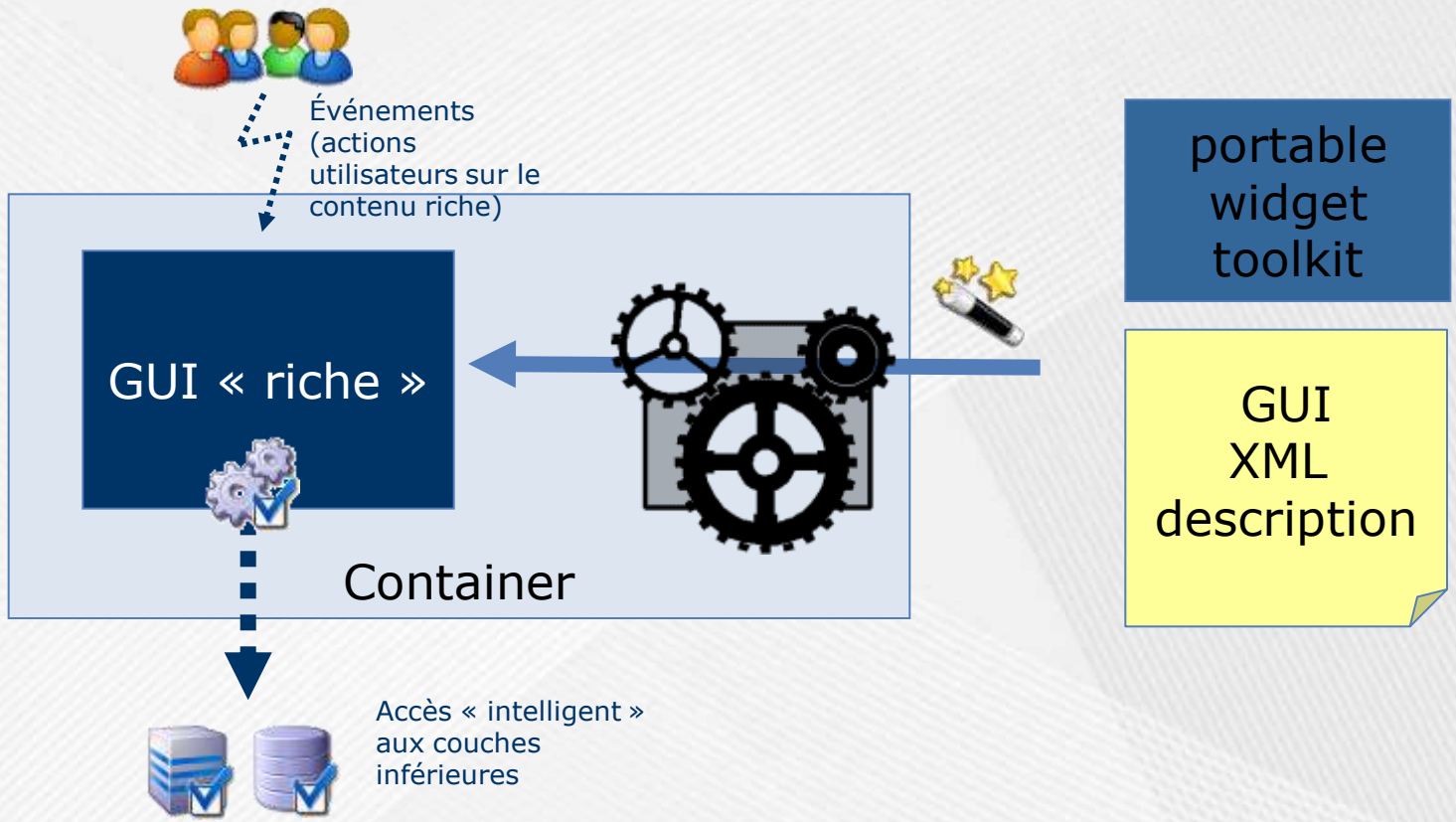
- L'un des principe essentiel du Web 2.0
  - Le web devient une plateforme: Il fourni des applications

- Avec les RIA…
  - les navigateurs sont maintenant vu comme un conteneur d'applications, a-t-on encore besoin d'un bureau?
  - Le RIA remet en cause le rôle du bureau du système d'exploitation pour l'utilisation des applications
  - (Voir les applications et même les OS en ligne tels que eyeos.org) (*egalement Echo (*http://demo.nextapp.com/echo3csjs/ ou *Ou Bindows (*http://www.bindows.net/demos/))

- Avec les RDA…
  - On veut montrer que les conteneurs d'applications peuvent être plus riche qu'un navigateur, a-t-on encore besoin d'un navigateur!?
  - Le RDA veut remettre en cause le rôle du navigateur comme conteneur d'applications universel

- Nous verrons bien
  - Le fait que le navigateur soit pré-installé sur le bureau est il un avantage suffisant?
  - Peut on imaginer un PC ou terminal sans OS (ou minimal), sans capacité de stockage, juste une connexion et un conteneurs d'applications?

Principe de fonctionnement



Les clients riches RIA

Ajax a gagné

Flash

- Flash (Adobe) ce n'est plus seulement le plugin permettant d'égayer les pages web avec des petites animations
- Flash est devenu un conteneur IHM pourvu de composants graphiques
  - de haut niveau (onglet, menu, menu déroulant,…)
  - et d'un look très séduisant
  - Flash 6.0+ minimum – dernière version 8.0+ (avril 2006)
- Il bénéficie en plus du mécanisme de déploiement simple du Web : simplement téléchargé par un navigateur, exécuté par le plugin flash
  - Aussi bien que JVM ou navigateur comme socle d'execution RDA
- Le plugin flash est installé sur 95% des postes connectés à Internet
- Les programmes Flash sont
  - assez léger, donc rapide à télécharger
  - Offre un télachargement par troncons
  - Offre une fonctionnalité de streaming Video adoptée de fait sur le Web
- Inconvénients
  - référencement difficile du fait que le format soit compilé
  - Format propriétaire

Flash c'est convaincant

- C'est du vrai multimedia
- Ca pourrait etre l'avenir du Web
  - On peut construire des sites entiers avec cette technologie (sans HTML)
  - C'est devenu indexable
  - Actuellement « réservé aux artistes »
    - Portfolio etc
  - Et bien sur les jeux Flash que vous connaissez bien
- Voir
  - http://www.bestflashanimationsite.com
    - Par exemple http://www.mazda2.com.co/
    - Par exemple http://www.synestension.org/
    - Par exemple http://www.air-atlantis.com/
  - http://www.thefwa.com/ (Favourite Website Awards)
    - Par exemple http://www.the711club.com (au hasard ;-))
    - Par exemple http://cardboard.theupsstore.com

développer une application Flash

- Adobe Flex
  - propriétaire, créée par Macromedia en 2004 puis reprise par Adobe en 2006
  - multi plates-formes grâce à la technologie Flash (présent partout)
  - Son modèle de programmation fait appel à MXML (basé sur XML) et ActionScript
  - MXML est un langage de description permet de décrire la présentation des interfaces
  - ActionScript est le langage de programmation (script) orienté objet
  - Pourvu d'un IDE wysiwyg Payant
- Voir Demo: http://www.adobe.com/devnet/flex/tourdeflex/
- OpenLaszlo
  - une version opensource de Flex, qui continue de s'appuyer sur le conteneur Flash
  - Un outil de développement est fourni par ibm sous forme d'un plugin d'Eclipse.
  - Languages: XML et JavaScript (ActionScript 3.0)
  - Le runtime de Laszlo pourra être dans le futur : Flash, DHTML, Java, SVG
  - (Flex peut s'interface avec Java coté serveur)

développer une application Flash

- Le langage
  - ActionScript est orienté objet
  - Il ressemble à java
    - dans le Langage
    - Dans le mode (compilateur vers du bytecode actionscript)
    - Machine virtuelle (le lecteur flash)

- Flash est à la fois
  - Le moteur de rendu ou lecteur
  - un IDE créé par Macromedia - Il en est a la sa version CS4
  - Mais cet IDE est remplacé par Flex
    - et l'IDE Adobe Flex Builder - dont la v2010, sera nommé **Adobe Flash Builder 4** ;-)
    - l'IDE Flash est plus à conseiller pour des graphistes
    - On y trouve des fonctions de dessin, une timeline, une gestion des calques

AJAX

- AJAX : Asynchronous JavaScript and XML (JavaScript et XML asynchrones)
- Ajax est un modèle d'architecture qui utilise des technologies existantes: HTML, CSS, DOM, XML et JavaScript
  - présentation: standards HTML et CSS
  - affichage dynamique et une interaction utilisant le Modèle Objet Document (DOM) de JavaScript
  - La récupération asynchrone de données en utilisant des requêtes XMLHttpRequest
  - Ces données peuvent etre XML, du texte formaté ou autre (JSON)
- Il permet de réaliser des mises à jour asynchrones dans une page (locales à une partie d'une page)
- L'architecture Ajax est compatible avec les applications Web développées avec tous les langages serveurs (JSP, PHP…)

Exemple Ajax

Exemple Ajax (2)

- Autre exemple du mode Ajax
  - application de cartographie Google maps
  - Avec le client RIA la navigation n'est pas intérrompue, les nouveaux morceaux de cartes sont téléchargés de manière asynchrone
  - Avec le client léger classique, chaque nouvelle demande entraine le rechargement complet de la page et donc un temps durant lequel l'utilisateur ne peut plus rien faire

Dans AJAX: Javascript

- C'est un langage de script qui est incorporé à Html et qui permet l'interactivité des pages web

- Dérivé de Java (donc de C)
  - Instructions communes avec Java/C++
  - Commentaires idem // et /* … */
  - ';' conseillé mais non obligatoire
  - Fait la différence entre les majuscules et les minuscules
- Insertion
  - dans l'entête du document HTML en général
  - N'importe où dans le corps du document HTML
- Utilisé surtout pour:
  - la vérification des données saisies dans un formulaire HTML juste avant l'envoi
  - Mettre de l'animation dans les pages web
- Modèle événementiel pour réagir aux évènements déclenchés par l'utilisateur (click de souris, etc)
  - OnSubmit
  - OnClick
  - OnMouseOver
  - …

Javascript avec HTML

- Intégré dans un page HTML : <SCRIPT>

```
<Head>
<script language="JavaScript">
<!-- //masquer pour les vieux navigateurs

function ValidationForm()
{
   if (document.form1.telephone.value.length < 10)
   {
       alert("les numéros de tel. ont 10 chiffres");
       return false;
   }
   else return true;
}
//-->
</script>
</head>
<body>
<form name="form1" action="/cgi/prog" method="POST"
     OnSubmit="return ValidationForm();">
<input type=text name="telephone">
<input type=submit VALUE="Envoyer">
</form>
</body>
```

le return peut **bloquer** ou pas l'envoi vers le serveur

Autre accès:
document.forms[0].elements[0].value

Javascript et HTML

- Evènements
  - Une liste d'évènements est disponible pour chaque balise HTML (format : « on… »)
  - On leur associe du code entre double quote
- Exemple :

```
…
<A href= "url"
onMouseOver="alert('la souris vient de passer dessus');">
un lien hypertexte</A>
…
```

- Si on ajoute « return false; » l'événement est annulé (à utiliser par exemple avec onSubmit)
- **On voit au passage la fonction alert(), utile pour débugger par exemple**

Javascript et HTML

Principaux évènements :

Evénement	description	principaux tags
onBlur	perd le focus	<INPUT> <SELECT> <TEXTAREA>
onClick	click avec la souris	<INPUT> <SELECT> <A>
onChange	la valeur a changé	<INPUT> <SELECT> <TEXTAREA>
onFocus	prend le focus	<INPUT> <SELECT> <TEXTAREA>
onLoad	le document HTML est complètement chargé	<BODY>
onMouseOut	la souris sort de la zone de l'objet	<A ... >
onMouseOver	la souris passe sur la zone de l'objet	<A ... >
onSubmit	envoie d'un formulaire vers le serveur	<FORM>

Javascript:Document Object Model

- JavaScript est associé à un jeu d'objets: le DOM
- Le DOM divise les pages Web en objets
  - C'est une hiérarchie d'objets
  - Navigateur-fenêtre-document-formulaire-champs de saisies
- JavaScript permet d'accéder à ces objets et de les manipuler.

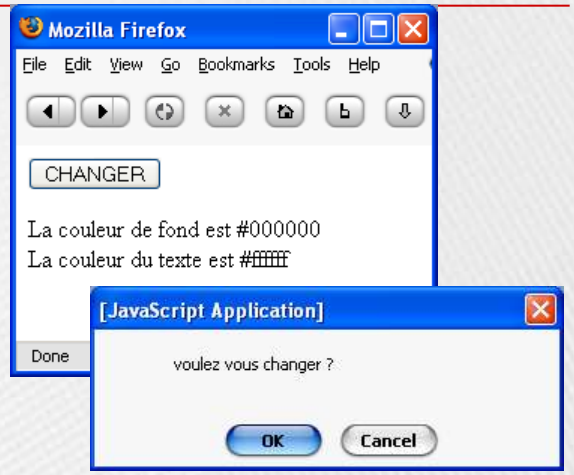
Javascript:Document Object Model

- A connaître si on veut maîtriser javascript.
- Permet de retrouver des éléments du navigateur
- Spécifications complètes à l'origine par Netscape

Javascript: Exemple pratique

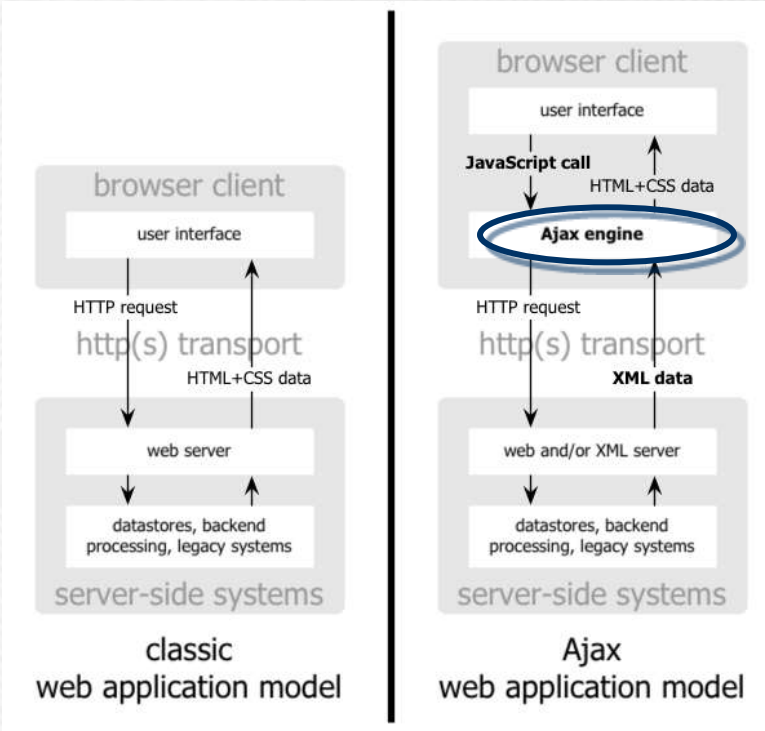
- Fonction confirm(): Permet de poser une question coté client

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!-- masquer pour les vieux navigateurs
function changer(Message) {
        if ( confirm (Message)) {
                if(document.TEST.Bete.value=="MODIFIER" )
                   document.TEST.Bete.value="CHANGER";
                else
                   document.TEST.Bete.value="MODIFIER";
                var old= document.fgColor;
                document.fgColor=document.bgColor;
                document.bgColor=old;
        }
}
// fin masquer pour les vieux navigateurs -->

</SCRIPT>
<form name="TEST">
<input type=button name="Bete" value="CHANGER" onclick="changer('voulez
        vous changer ?');" ></form>

<SCRIPT LANGUAGE="JavaScript">
document.write("La couleur de fond est ", document.bgColor,
"<BR>La couleur du texte est ",document.fgColor);

</SCRIPT>
</BODY>
</HTML>
```



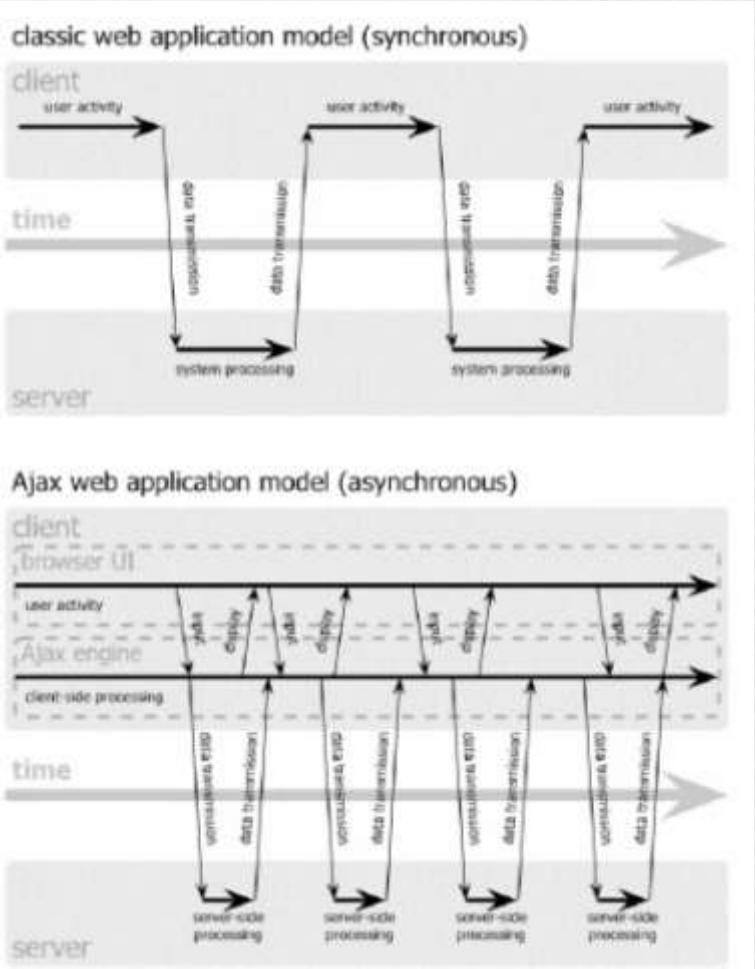
Mode de fonctionnement d'AJAX

- Ajax est le nom donné à une technique de conception de pages html qui utilise des scripts client-side pour échanger de la donnée avec le serveur Web
- Elle existe depuis des années en fait, à l'initiative de Microsoft
- Le principe de base d'Ajax se résume en un mécanisme visant à éviter de mettre à jour la page html complète à chacune des interactions utilisateur.
- Ajax n'est que l'utilisation conjointe d'un certain nombre de technologies existantes :
  - Presentation basée sur les standards XHTML et CSS
  - Javascript pour intercepter les événements et envoyer la requête au serveur web
  - Les echanges de données sont supposées se faire en XML
  - Recupération asynchrone de donnée via XMLHttpRequest
  - Javascript + DOM pour récupérer le résultat et modifier le contenu de la page courante en modifiant l'arbre DOM qui lui est associé.
- Un nouvel intermédiaire est né: un moteur Ajax



Ajax (Asynchronous Javascript And XML)

- les technologies http et html ne permettent d'associer qu'une seule action à un événement utilisateur :
  - l'envoie d'une requête http vers le serveur web
- Inconvénients
  - Une « latence » dans la réaction aux actions utilisateurs
  - Le transit de la totalité des informations de présentation même si seulement une partie doit changer
- Ajax remplace le comportement naturel du navigateur:
  - Envoie d'une requête en arrière plan
  - une régénération partielle d'une partie de la page HTML
  - avec les informations qu'il recevra en retour (dépourvues de toute information de présentation)
- Un nouvel intermédiaire, un moteur Ajax permet de:
  - Transformer tout action qui aurait généré une requête HTTP en un appel javascript vers le moteur
  - Gérer directement toute action qui ne nécessite pas un retour vers le serveur (validation, édition, navigation…)
  - Si le moteur a besoin d'information du serveur, il fait la requête en asynchrone, sans bloquer l'utilisateur



Ajax (Asynchronous Javascript And XML)

- principe d'Ajax
  - Le navigateur héberge une application et non pas un contenu.
  - Le serveur fournit des données pas du contenu.
  - L'interaction de l'utilisateur avec l'application peut être fluide et continue

- Limites d'AJAX
  - Problèmes de compatibilité avec certains navigateurs
  - Difficulté de référencement par les robots d'indexation
  - En conflit avec certaines fonctionnalités Web (« Précédent », « Favoris »)

- Demo Ajax en 10mn: http://developer.mozilla.org/fr/docs/AJAX:Premiers_pas

Frameworks Ajax (non exhaustif):

dhtmlx Dojo Echo3 YUI Ext Google Web Toolkit jQuery midori MochiKit MooTools Prototype script.aculo.us Pyjamas qooxdoo Rialto Rico SmartClient SweetDEV ZK

- **jQuery http://jquery.com/**
- **Prototype http://prototypejs.org**
  - **script.aculo.us http://script.aculo.us**
  - Rico http://openrico.org/
- MooTools http://mootools.net
- Dojo http://dojotoolkit.org
  Demo: http://demos.dojotoolkit.org/demos/
- dhtmlx http://dhtmlx.com
- YUI http://developer.yahoo.com/yui/
  demo: http://yuilibrary.com/gallery/
  - Ext http://extjs.com/
- midori http://www.midorijs.com/
- Qooxdoo http://qooxdoo.org
  demo: http://demo.qooxdoo.org
- Rialto http://rialto.application-servers.com
  demo: http://rialto.improve-technologies.com/rialto/

- Basé Java
  - Echo3 http://echo.nextapp.com/site/
    Demo: http://echo.nextapp.com/site/demo
  - Google Web Toolkit
    http://code.google.com/intl/fr-FR/webtoolkit/
  - Demo: http://gwt.google.com/samples/Showcase
  - ZK http://www.zkoss.org/ (Zero Kelvin)
    Demo: http://www.zkoss.org/zkdemo/userguide
  - SweetDEV
    http://sweetdev-ria.sourceforge.net
    Demo: http://demo.sweetdev-ria.com/
  - SmartClient http://www.smartclient.com/
    Demo: http://www.smartclient.com/featureExplorer.jsp
- Basé Python
  - MochiKit http://www.mochikit.com/
  - Pyjamas http://pyjs.org/
- Basé Microsoft
  - ASP.NET AJAX (ou Atlas) http://ajax.asp.net/
    Control Toolkit:
    http://www.codeplex.com/AjaxControlToolkit
    Démonstration Control Toolkit:
    http://www.asp.net/ajax/ajaxcontroltoolkit/samples/

A faire: Démos
script.aculo.us
qooxdoo
dhtmlx
Dojo
zk
Echo
smartclient

Java & les applets

A faire?

On a le temps?

Les clients riches RDA

Etat actuel en en 5min

Eclipse RCP: Java

- Eclipse: au départ un IDE (environnement de développement) pour Java
  - eclise.org
  - Créé par IBM (pour l'IDE WebSphere et son serveur d'application)- jeu de mots
- Eclipse possède un grand nombre d'extensions tels que diagrammes UML et le support d'autres langages, etc
- S'appuie sur
  - SWT et (Librairie graphique très rapide car elle native à l'OS au contraire de SWING)
  - JFace (Surcouche de SWT des composants graphiques)
- RCP = « Rich Client Platform »
  - Extraction de SWT et Jface, dans une executable autonome de l'IDE
- Points forts
  - Développement en Java
  - Accès natifs, gère les matériels: lecteurs de code à barre, les caisses, imprimantes, etc.
  - Le meilleur des 2 mondes
  - mode « hors connexion » possible
  - Réactivité indépendante de la bande passante et egale à un client lourd

Sun Java Web Start

- permet d'exécuter des applications riches sur tous les systèmes PC comme mobiles.
  - son système de déploiement automatique « Java Web Start » est le plus perfectionné et le plus ancien
  - Mais Java a une mauvaise réputation quant à l'ergonomie et la vélocité de ses interfaces
- On peut la voir comme une applet « hors navigateur »
  - Remplace le navigateur qui force à les inclure dans une page Web
  - Mêmes Avantages et inconvénients que les applets: assez rapide, portable, mais nécessite le plugin Sun Java, et les applis mettent du temps à télécharger
  - Mais pas de description XML du GUI (sauf dans les IDE ;-)
- Empreinte
  - JRE (100 Mo sur le disque) – la version la plus récente dispo avec J2SE 1.4.1
  - Java WS runtime (9 Mo)
- Principes
  - Délocalisation du code chez le client
  - Les Applications java sont sur un serveur Web de manière à ce qu'on puisse les télécharger et les exécuter
  - Le client met en cache l'application et la met à jour automatiquement si nécessaire à chaque lancement (Mise à jour transparente pour les clients)
  - Elle peuvent être exécutées hors connexion
  - Le moteur d'exécution est installé avec le JRE (Java runtime Environment)
- Demo http://java.sun.com/javase/technologies/desktop/javawebstart/demos.html

Microsoft ClickOnce

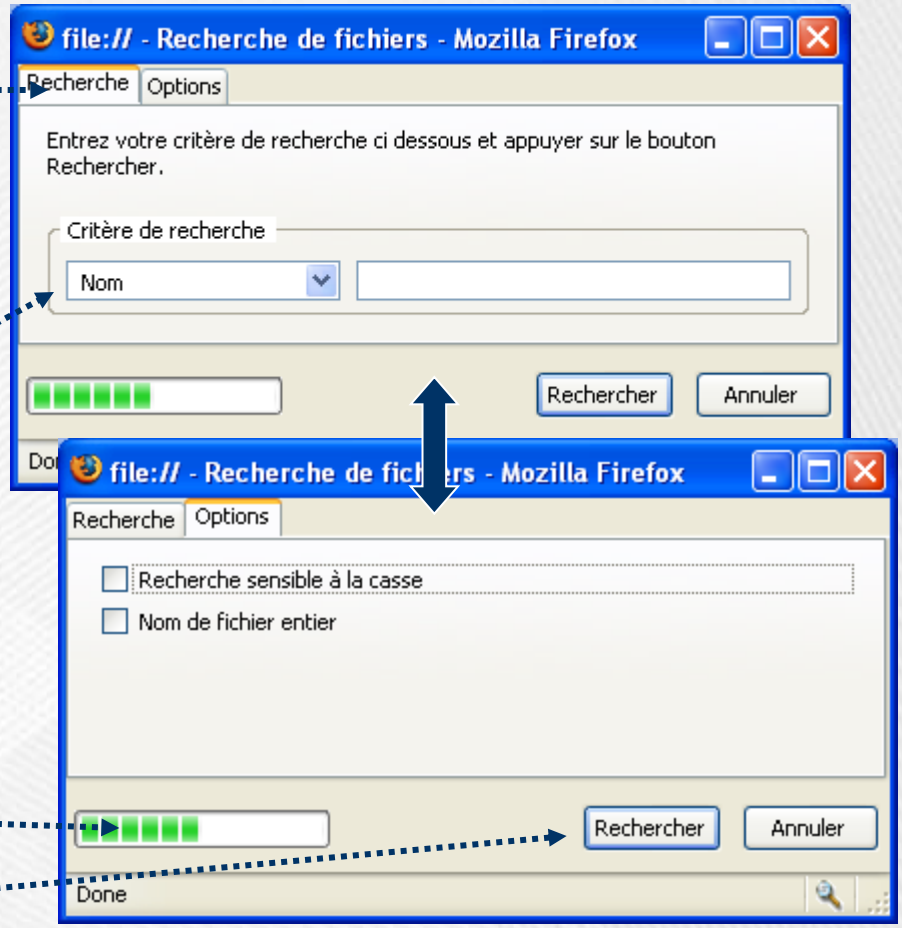
- ClickOnce est basé sur le Windows Presentation Foundation (WPF) nommé aussi « Silverlight »
  - uniquement accessible sur le framework .NET
  - unqieuemnt sous Windows
- Il représente la réponse de Microsoft à Java Web Start.
  - Microsoft utilise le même concept que Sun: solution de déploiement automatique

Xul (Mozilla Firefox)

- Précurseur, Xul est un langage xml de description d'interface graphique riches (Prononcé "zool")
- Il est nativement mieux pourvu en composants graphiques de haut niveau qu'html (ensemble de widgets XUL)
- il utilise, un modèle de programmation événementiel pour gérer les actions de l'utilisateur : JavaScript
- XUL a été initialement conçu pour fonctionner dans un navigateur web (tq Firefox), à la sauce RIA.
- Il fait partie du framework Mozilla et donc disponible sur les dérivés comme Firefox.
- Pour respecter le principe RDA :
  - XulRunner est apparu (2006)
  - Le tout n'est pour l'instant pas très stable
- XUL est prévu pour permettre le portage d'applis sur d'autres périphériques comme les PDA ou les mobiles

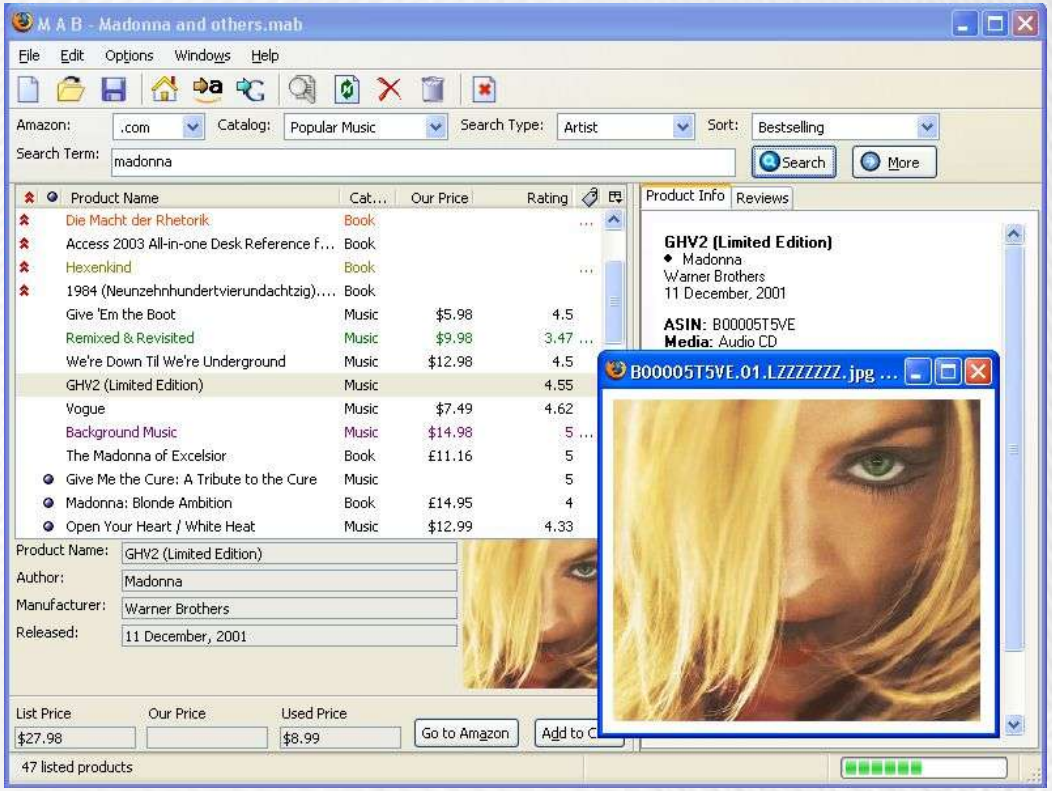
Exemple XUL

```
<tabbox>
 <tabs>
  <tab label="Recherche" selected="true"/>
  <tab label="Options"/>
 </tabs>
 <tabpanels>
  <tabpanel id="searchpanel" orient="vertical">
   <description>
    Entrez votre critère de recherche ci dessous et appuyer ....
   </description>
   <spacer style="height: 10px"/>
   <groupbox orient="horizontal">
    <caption label="Critère de recherche"/>
    <menulist id="searchtype">
     <menupopup>
      <menuitem label="Nom"/>
      <menuitem label="Taille"/>
      <menuitem label="Date de modification"/>
     </menupopup>
    </menulist>
    <spacer style="height: 10px"/>
    <textbox id="find-text" flex="1" style="min-width: 15em;"/>
   </groupbox>
  </tabpanel>
  ...
 <hbox>
  <progressmeter value="50%" style="margin: 4px;"/>
  <spacer flex="1"/>
  <button id="find-button" label="Rechercher" default="true"/>
  <button id="cancel-button" label="Annuler"/>
 </hbox>
```



Exemple XUL

- Le très connu MAB (Mozilla Amazon Browser)
  - http://www.faser.net/mab/remote.cfm
- Miro (anciennement appelé Democracy Player): télé en ligne
  - http://www.getmiro.com/
- Voir une liste ici : http://xulfr.org/wiki/RessourcesLogiciels
- OpenSI
  - logiciel de comptabilité libre
  - http://www.opensi.org/



Adobe AIR

- AIR=Adobe Integrated Runtime - anciennement nommé Apollo
  - Même principe que le Java Runtime Environment associé à JavaWebStart
- Il s'agit de Flash hors navigateur, Flash sur le bureau
  - Sortie en 2008
  - Windows, Mac et Linux
- Fonctionnalités principales
  - pour le rendu HTML/CSS et l'exécution de code JavaScript, il intègre même moteur de rendu WebKit (http://webkit.org) que les navigateurs Apple Safari et Google Chrome, Konqueror
  - intègre Flash Player pour l'exécution de fichier SWF
  - possibilité de créer, d'éditer et supprimer des fichiers
  - intégration d'une base de données locale basée sur SQLite
- Plus d'info
  - télecharger: http://get.adobe.com/fr/air/ (15Mo en 2009)
  - Voir Demo: http://www.adobe.com/devnet/flex/tourdeflex/ (37 mo)

A faire:
Démo
AIR

Conclusions

Web 2.0 et clients riches

Conclusions

- Avantages
  - (RIA) Plus grandes possibilités grace au mode déconnecté.
  - (RIA) Interfaces plus riches, les échanges entre le navigateur et le serveur sont limités aux données.
  - (RDA) Facilité de mise en oeuvre (déploiement et maintenance automatisés)

- Inconvénients
  - (RIA) Plus complexe à programmer.
  - (RIA) Abondance de technologies concurrentes.
  - (RIA) Changement dans la façon de naviguer.
  - (RDA) Nécessite un temps de télechargement et de mise à jour.

Conclusions

- RDA: niveau d'adoption resté « confidentiel »
  - Eclipse RCP: pour les Application métier type ERP
    - Financier, scientifique, médical, etc.
  - JavaWebStart
    - Technologie RDA la plus ancienne, n'a jamais vraiment décollé
  - Pour l'instant ca reste des jouets techniques
- RIA: le web 2.0 est là
  - Améliore l'ergonomie
  - Réactivité moins dépendante de la bande passante
  - RIA: tourne à un combat Flex vs Ajax, mais Ajax seul ne suffit pas
  - RIA = AJAX + Tools + Framework + Controles
  - Ajax semble avoir gagné car:
    - se base sur des standards uniquement
    - Disponible sur tout ordinateur dans le monde à coup sur
- Points communs
  - Une installation et une mise à jour automatisées
  - Une adaptation en fonction des préférences de l'utilisateur
  - Une réactivité équivalente aux applications « natives » de l'OS tout en étant indépendant de l'OS
  - Moins de BP utilisée car seule la donnée passe sur HTTP

Conclusions

- Le Web 2.0 est un concept flou
  - Mais c'est un nom donné à des pratiques de communication bien réelle-adoptées
- Web 1, 2, 3?
  - Web 1 : accès à l'information
  - Web 2 : web participatif et social, intelligence collective, …
  - Web 3 : web de données = sémantique…(?)
- Convergence du pro, perso et social
  - →identité numérique
  - Pas facile à maitriser

Conclusions :A suivre

- L’evolution des technologies de « Clients Riches Autonome »
  - BD locale
  - Moteur de synchronisation bidirectionnelles
- Acteurs
  - Google avec Gears & Chrome
  - Mozilla avec Firefox
  - Adobe avec AIR

Fin: Références

- XUL: http://xulfr.org/xulplanet/xultu/ (tutorial FR)
- Ajax en 10mn: http://developer.mozilla.org/fr/docs/AJAX:Premiers_pas
  - frameworks Ajax (non exhaustif) : AjaxAnywhere (JSP/JSF), Dojo, Bindows
- Flash, Adobe Flex, OpenLaszlo
  - http://www.openlaszlo.org/
  - http://www.macromedia.com/software/flex/ (home)
  - http://www.macromedia.com/devnet/flex/articles/first_flexapp.html (tutorial ENG)
- Adobe AIR
  - télecharger: http://get.adobe.com/fr/air/
  - Voir Demo: http://www.adobe.com/devnet/flex/tourdeflex/
- Eclipse RCP:
  - http://wiki.eclipse.org/index.php/Rich_Client_Platform (home)
  - http://eclipse.org/articles/Article-RCP-1/tutorial1.html (Tutorial en 3 parties)
- Java Web start: http://java.sun.com/products/javawebstart/
- JavaFX
  - http://javafx.com/
  - https://openjfx.dev.java.net/
- Google web toolkit
  - tutorial http://courses.coreservlets.com/Course-Materials/gwt.html

Backup slides

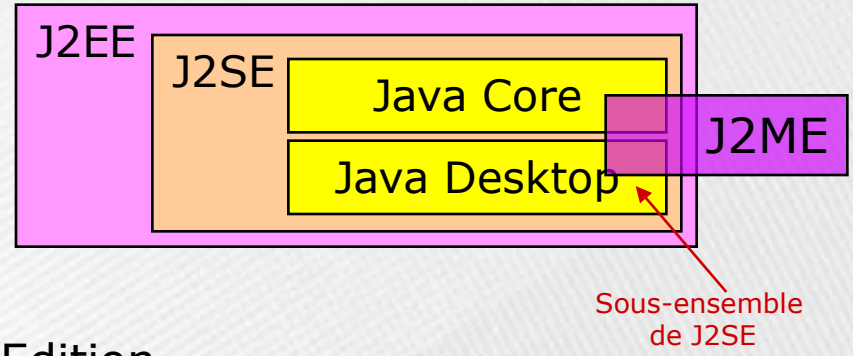
A faire si suffisamment de temps

Java

- Un langage objet
  - Dérivé de C++
  - Java est un langage compilé
- indépendance vis-à-vis de la plate-forme
  - Concept de « machine virtuelle »
  - Concept de WORE
- La « JVM » est chargée d'implémenter l'ensemble des pseudo-codes machine Java

Java: les différentes versions

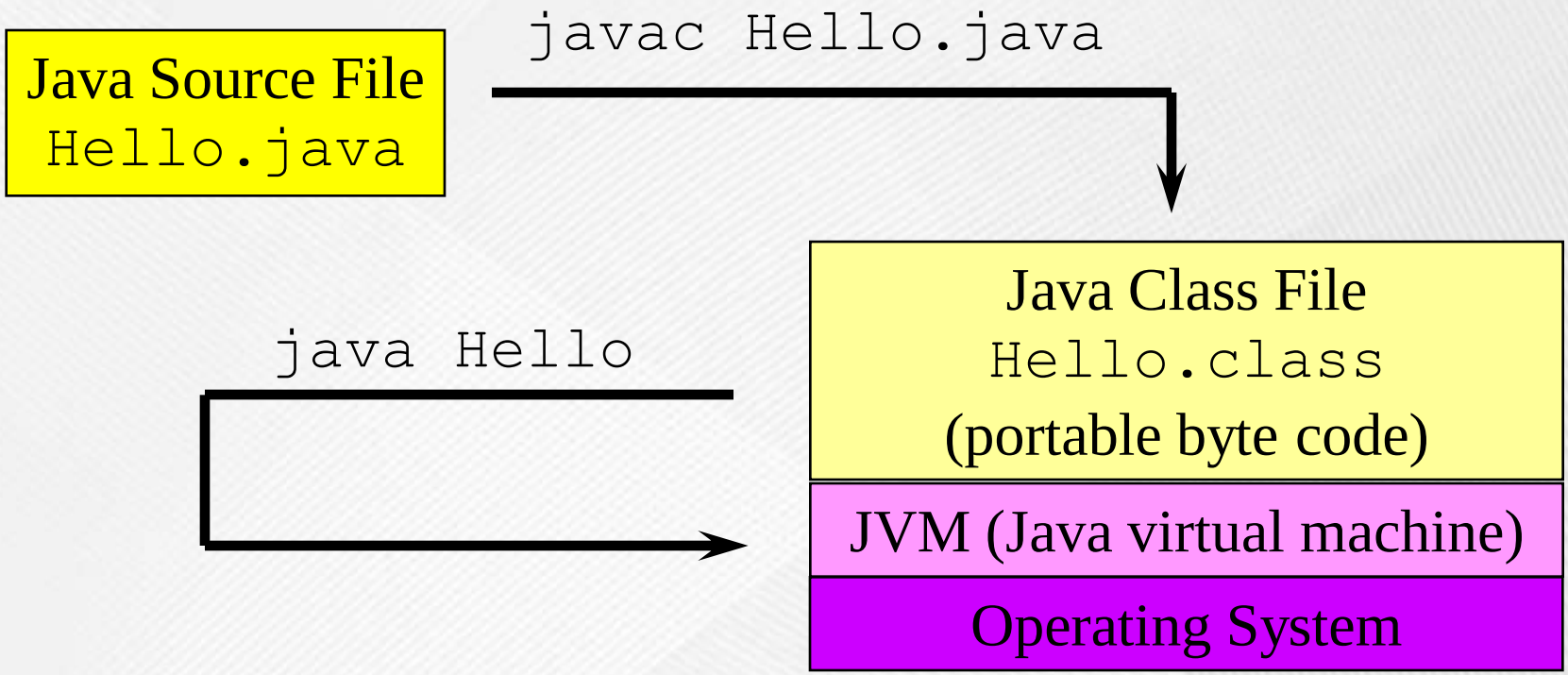
- JDK ou JRE?
  - JDK = Java Development Kit
  - JRE = Java Runtime Environment
  - JDK = JRE + compilateur
  - JRE = JVM + API Java Core
    - (JVM=Java Virtual Machine)



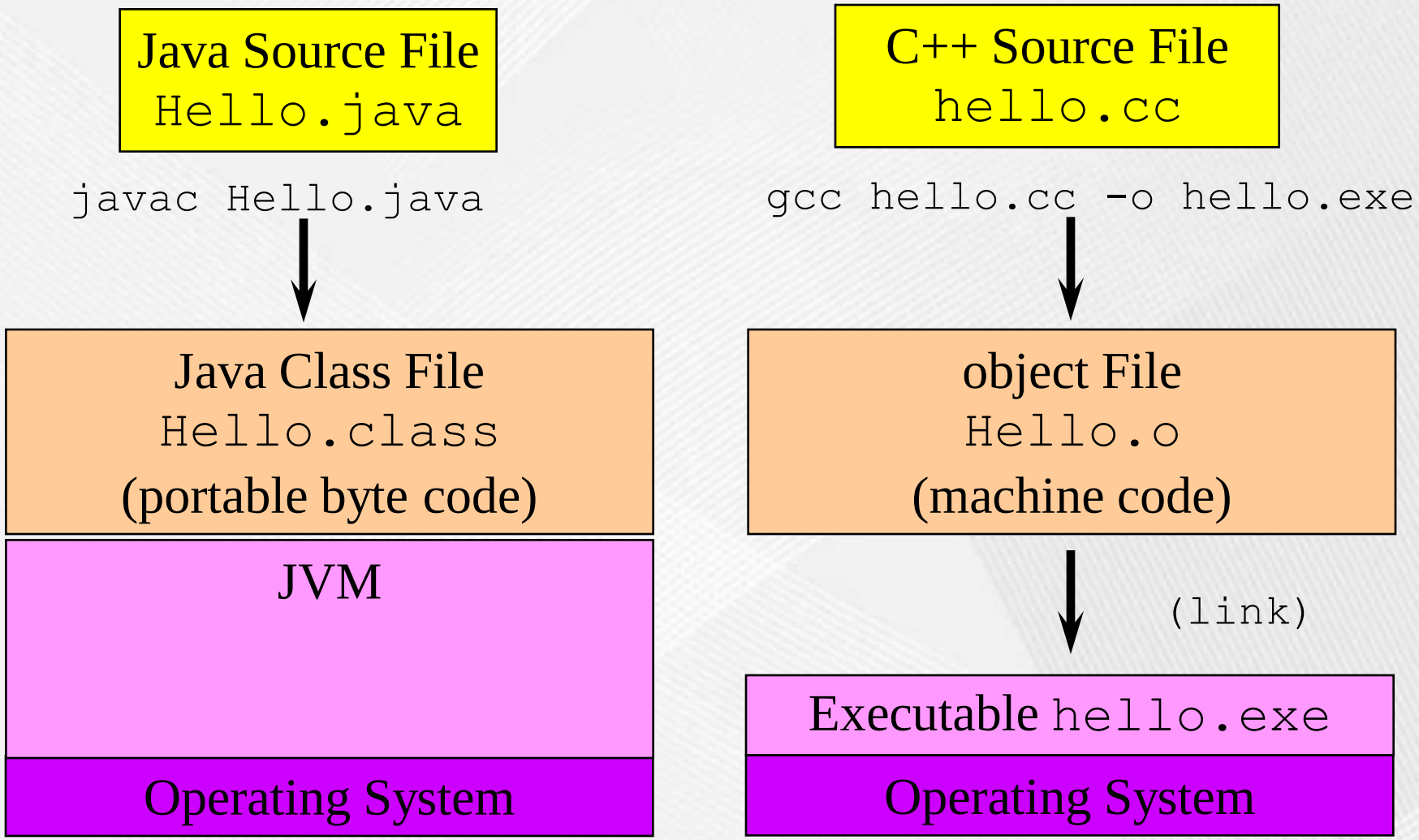
- J2SE ou J2EE ou J2ME?
  - J2SE = Java 2 Platform Standard Edition
    - Environment pour des applications classiques (de bureau) sous sur les OS classiques Windows/Linux/Mac OS
    - Sert de fondation à J2EE
  - J2EE = Java 2 Platform Enterprise Edition
    - Environment pour des applications multi-tier d'enterprise
  - J2ME = Java 2 Platform Micro Edition
    - Environment pour des applications embarquées, en particulier téléphones mobiles et PDA
    - Une version reduite et limitée de J2SE pour l'embarqué

- A noter qu'il existe une version de java pour la carte à puces (JavaCard) avec un status un peu particulier

Java est portable : Modèle VM



Byte Code vs. Machine Code



Java : les applets

- Programmes téléchargeables dans les navigateurs Web (plugin présent)
- incorporées dans une page HTML
- Le concept a été créé par Netscape
  - « détournement » de l'utilisation de Java
  - (machine virtuelle incorporée dans le navigateur)
- création d'applet :
  - Il suffit de dériver de la classe « Applet »
  - L'objet hérite alors de :
    - zone graphique rectangulaire attribuée par le navigateur
    - Un ensemble d'événements standard
    - De contraintes sécuritaires

Applets Java

HelloWorldApplet.java

```
public class HelloWorldApplet extends java.applet.Applet{
    public void paint(java.awt.Graphics g) {
        String param = getParameter("texte");
        g.drawString("Hello world! "+param, 50, 25);
    }
}
```

HelloWorldApplet.html

```
<HTML><BODY>
Ici se trouve notre applet :
<APPLET CODE="HelloWorldApplet.class" WIDTH=150 HEIGHT=50>
Votre Browser ne permet pas de visualiser les applets Java
<PARAM NAME = "texte" VALUE = "Mickey">
</APPLET>
<BODY></HTML>
```

Applets Java

Structure et événements des applets

```
public class HelloWorldApplet extends java.applet.Applet {
    public void init() {
        String param = getParameter("text");
    }

    public void destroy() {    }
    public void start() {    }
    public void stop() {    }
    public void run() {    }

    public void paint(Graphics g) {
        g.drawString(1, 1, "Hello World");
    }
    public String getAppletInfo() {
    return "Applet minimal";
  }
}
```

Applets Java comme client riche?

- L'applet est confinée dans une zone graphique rectangulaire attribuée par le navigateur
- L'applet a des contraintes sécuritaires fortes
  - Impossible d'accéder aux ressources locales
    - Fichiers
    - imprimante
  - Les connexions TCP possibles que vers le serveur d'origine
- Problèmes de déploiement
  - Communication avec le navigateur hôte (ex gestion du proxy dans un réseau privé)
  - Versions du JDK du navigateur hote
  - Variabilité également suivant l'OS
- Mais
  - Les applets peuvent communiquer avec Javascript
  - Les applets peuvent etre redimensionnées
  - Les applets peuvent ouvrir des fenetres additionnelles