

# Les Clients Riches

Voici mon essai de synthèse sur les technologies regroupées sous le terme générique de "Client Riche". Points communs et différences RCP? RIA? Ajax? Xul? Flex? Laszlo? Svg? Xaml?

1 - Finalement, y'a quoi derrière ce terme?

Un client riche, c'est :

- Une des technologies permettant d'améliorer la couche présentation d'une application n-tiers (web ou autre) - du moins orientée réseau
- sensée combler le gap entre application native (installée sur le bureau) et application Web au niveau de l'ergonomie tout en restant portable

- Une technologie qui intègre la problématique de déploiement et de mise à jour de version (qui garde les qualités de déploiement des applications web actuelles)

Ces technologies ont maintenant une maturité suffisante pour fournir des bibliothèques de composants graphiques (boîte, liste, grilles...) offrant richesse et ergonomie se rapprochant des applications de type client lourd.

on peut classer les clients riches en deux grandes familles :

- " Rich Internet Applications " (RIA)

- " Rich Desktop Applications " (RDA)

RIA : Composant essentiel du "Web 2.0".

Ce qui les distingue?

la nécessité d'installer (ou pas) un socle d'exécution sur le poste client. Pour les RIA ce socle d'exécution est tout simplement un navigateur Web standard.

2 - Principe de fonctionnement

une description XML de l'interface permet de générer à la volée à partir d'un moteur et d'une librairie de "Widgets" (composants graphiques) l'interface graphique utilisateur.

La réaction aux actions utilisateurs est assurée par une intégration avec un langage de script ou interprété/chargé dynamiquement.

La suite présente " l'essentiel ", et n'a pas vocation à être exhaustive.

3 - Panorama

Les clients riches « RIA »

- XUL - Extensible User Interface Language (Mozilla)
- Ajax - Asynchronous Javascript and Xml
- Flash, Adobe Flex, OpenLaszlo
- Applets Java (mis de côté)
- SVG &ndash; Scalable Vector Graphics (mis de côté)

Les clients riches « RDA »

- Eclipse RCP
- Flex desktop
- XAML &ndash; Extensible Application Markup Language (Microsoft) - windows « vista »
- Java Web start

## 4 - XUL

Précurseur, Xul est un langage XML de description d'interface graphique riches (Prononcé "zool").

Il fait partie du framework Mozilla et donc disponible sur les dérivés comme Firefox.

Il offre un ensemble de composants graphiques (les widgets XUL) du niveau de ceux disponibles en natif sous Windows ou Mac OS, et qui dépassent ce que l'on peut faire avec HTML. L'ensemble reste portable (mais lié à Mozilla/Firefox).

Il utilise un modèle de programmation événementiel pour gérer les actions de l'utilisateur : JavaScript

même si XUL a été initialement conçu pour fonctionner dans un navigateur web (mode RIA), des extensions le rapprochant du principe RDA sont apparues (comme XulRunner qui permet de lancer des applications XUL sans avoir besoin d'installer Mozilla ou Firefox).

XUL est désigné pour permettre le portage d'applications sur d'autres périphériques comme les PDA ou les mobiles.

## 5 - Ajax

Ajax est le nom donné à une technique de conception de pages HTML qui utilise des scripts Javascript coté client pour échanger de la donnée avec le serveur Web sans nécessairement rafraichir la page Web.

Comme exemple Ajax le plus typique, essayer Google suggest, qui complète la saisie en allant chercher des mots clés sur le serveur.

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 est l'utilisation d'un ensemble de technologies existantes :

- Présentation basée sur les standards HTML et CSS
- Javascript pour intercepter les événements et envoyer la requête au serveur web
- Les échanges de données sont supposés se faire en XML (mais pas forcément)
- Recupération asynchrone de donnée via un objet XMLHttpRequest et Javascript
- Javascript pour modifier l'arbre DOM qui est associé à la page courante

### Des détails

Les technologies HTTP et HTML ne permettent en standard d'associer qu'une seule action à un événement utilisateur : l'envoi d'une requête HTTP vers le serveur web.

Inconvénients:

- Une « latence » dans la réaction aux actions utilisateurs (qui dépend du réseau)
- Le transit de la totalité des informations de présentation même si seulement une partie doit changer
- L'utilisateur est forcé d'attendre que l'ensemble de la réponse du serveur Web soit rapatriée

Ajax remplace le comportement naturel du navigateur:

- Envoie d'une requête en arrière plan
- Régénération partielle d'une partie de la page HTML
- Avec les informations qu'il recevra en retour (dépourvues de toute information de présentation) au moment où elle seront arrivées (requête asynchrone)
- le tout sans forcément bloquer l'utilisateur

Un nouvel intermédiaire est né: le moteur Ajax.

Il existe déjà une offre de solutions prenant en charge cette nouvelle complexité liée à Ajax.

Appelons-le moteur Ajax. On peut le considérer comme un nouveau tier.

Ce moteur est une couche Javascript embarquée dans le navigateur qui va prendre en charge cette gestion asynchrone de la communication avec le serveur et la mise à jour partielle de la page Web.

## 6 - Flash

Flash (Adobe) ce n'est pas 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, &hellip;)
- et d'un look très séduisant
- Flash 6.0+ minimum &ndash; dernière version 8.0+ (avril 2006)

Le plugin flash est installé sur 95% des postes connectés à Internet. On considère qu'une application Flash bénéficie du mécanisme de déploiement simple du Web : téléchargé par un navigateur mais exécuté par le plugin flash. Comme socle d'exécution, il est possible de dire que c'est aussi bien qu'une JVM Java ou le navigateur même.

Technologies permettant de développer une application Flash:

- Flex (Adobe). Payant. Pourvu d'un IDE wysiwyg. Partir de Flex 2.0 pas encore sorti
- Laszlo. Open Source. Un outil de développement est fourni par IBM sous forme d'un plugin d'Eclipse.
- Languages: XML et JavaScript (ActionScript 3.0) A Noter:
- Le runtime de Laszlo pourra être dans le futur : DHTML, Java, SVG
- (Flex s'interface avec Java coté serveur)

## 7 - Les clients riches « RDA »

Les clients riches « RDA » décrètent la fin de l'utilisation du navigateur web comme « conteneur d'application ».

- utilisé pour télécharger et exécuter/afficher la couche de présentation
- il est nécessaire que ce nouveau conteneur soit installé sur le poste qui exécutera l'application, rendant non accessible l'application depuis un poste banalisé
- ne pas assimiler le conteneur à l'application !
- Le conteneur fournit un socle d'installation et déploiement, de mise à jour et d'exécution aux applications

## 8 - Les conteneurs « RDA » Il existe principalement:

- Eclipse « RCP » (« Rich Client Platform »)
- S'appuie sur SWT et (Librairie graphique très rapide car elle est native à l'OS au contraire de SWING) JFace (Surcouche de SWT des composants graphiques)
- Adobe Flex version « desktop »
- le futur Windows « Vista » qui intègre directement le moteur de rendu dans le système d'exploitation (basé XAML)
- XAML est un langage de description d'IHM riches (et dynamiques)
- Inclu dans Avalon, nouvelle génération de l'UI (sera supporté par IE 7)
- Avantages: performances, robustesse, très configurable.
- inconvénients: technologie Microsoft-only &ndash; dispo quand l'OS sera sortie
- Java Web start
- Finalement rien de plus qu'un conteneur pour des applets
- Remplace le navigateur qui force à les inclure dans une page Web

- Mêmes Avantages et inconvénients que les applets: 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 ;-)

#### 9 - Conclusions RCP: Type d'application

- Application métier type ERP (Financier, scientifique, médical, etc.)
- Points forts
- Gère les matériels: lecteurs de code à barre, les caisses, imprimantes, etc.
- mode « hors connexion » possible
- Réactivité indépendante de la bande passante RIA: ça va marcher!
- Pourquoi: parce que tout utilisateur aime immédiatement !
- Améliore l'ergonomie
- Réactivité moins dépendante de la bande passante
- Va remplacer les applications web actuelles
- Un nouveau genre d'application est apparue (Web 2.0)
- RIA: Flex vs Ajax, mais Ajax seul ne suffit pas
- RIA = AJAX + Tools + Framework + Controls

#### Points communs

- Une installation et une mise à jour automatisées
- Une capacité à gérer des préférences de l'utilisateur
- Une réactivité équivalente aux applications « natives » de l'OS tout en étant indépendant de l'OS malgré tout!
- Moins de BP utilisée car seule la donnée passe sur HTTP

#### 10 - Références

- XUL: <http://xulfr.org/xulplanet/xultu/> (tutorial FR)
- Ajax en 10mn: [http://developer.mozilla.org/fr/docs/AJAX:Premiers\\_pas](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](http://www.macromedia.com/devnet/flex/articles/first_flexapp.html) (tutorial ENG)
- Eclipse RCP:
- <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/adessaig/rcp.html> (tutorial FR)
- <http://www.eclipse.org/rcp/> (home)
- Java Web start: <http://java.sun.com/products/javawebstart/>