



Web 8.3

Mais où donc en est le web ???

23 octobre 2007
Paris, la Défense

Xavier PARADON
Directeur Technique

Valtech Training

xavier.paradon@valtech.fr



Thèmes abordés



Partie 1 – Problématiques actuelles du web

Partie 2 – Web 2.0 ou RIA, quelles solutions ?

Partie 3 – Web 8.3, encore des évolutions ?



© Valtech 2007



Le web et ses problèmes



Productivité des technologies serveurs : ASP, PHP, JSP

- Scripts serveurs mal écrits
- Complexes à faire évoluer
- Basés sur une syntaxe HTML trop pauvre

Mode page et convivialité

- Navigation longue et complexe
- L'utilisateur est souvent en attente
- Les pages chargées sont très semblables

Mode déconnecté

- Quasiment inexistant

Richesse de l'IHM

- Plus de fonctionnalités
- Contenus plus riches et interactifs (vidéos, audio)

Universalité

- Programmation cross-browser complexe (beaucoup de technologies)
- Soucis de portabilité



Nous sommes tous Web 2.0



Le web 2.0 est une notion très floue combinant

- Des aspects marketing ou stratégiques
- Des aspects technologiques

Pas de définition formelle

- Terme vide de sens dans la pratique
 - Chacun y voyant ce qu'il veut
- Majoritairement associé à une IHM enrichie (RIA)
 - Prédominance du côté technique
- Parfois associé à une démarche visant à placer l'utilisateur au cœur de l'application web
 - L'utilisateur devient un acteur, il publie de l'information



RIA = Rich Internet Application



Ensemble de technologies ayant en commun :

- Un environnement d'exécution intégré au navigateur
- Des composants graphiques
- Une gestion d'évènements côté client

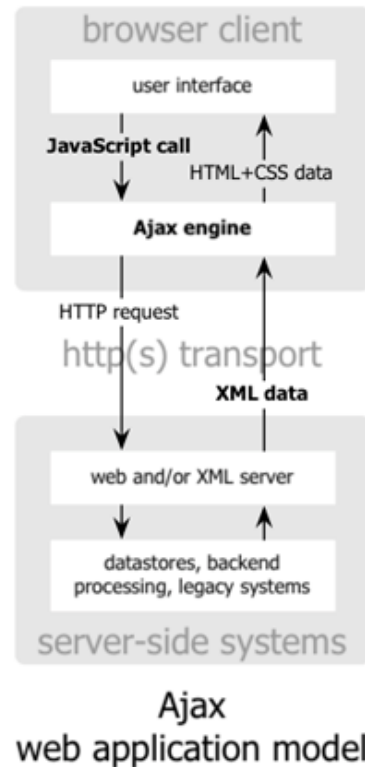
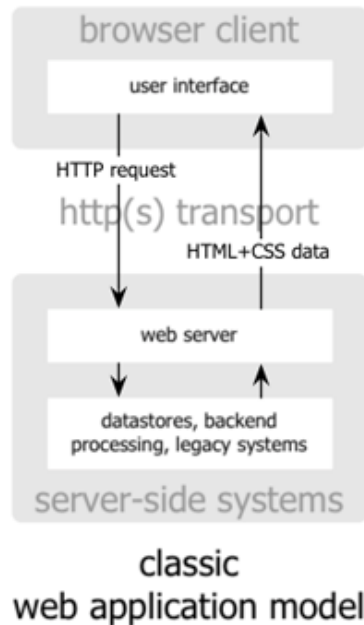
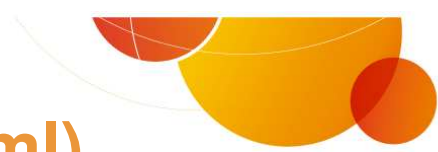
Type de développement web prônant :

- Une richesse accrue de l'IHM
- L'abandon du mode page
- Une communication avec le serveur simplifiée
 - Échange de données via Http
- Des gains de productivité associés



Technologies RIA avec Javascript

AJAX (Asynchronous Javascript And Xml)



L'interface graphique n'est pas rafraîchie à chaque action

Le mode page disparaît, seule les données transitent par HTTP(S)

La requête HTTP émise est asynchrone

Le modèle événementiel est obtenu grâce aux fonctions JavaScript au dessus de HTML.

Technologies RIA avec Javascript Frameworks Javascript/AJAX



AJAX repose essentiellement sur l'écriture de code JavaScript

- Devant régler le problème de la mise à jour de l'IHM (DHTML)
- Devant fournir des fonctionnalités graphiques (Drag and Drop ...)
- Devant fonctionner sur différents navigateurs
- Peut être facilité par l'intégration de bibliothèques Javascript
 - Dojo
 - Rico
 - Script.aculo.us
 - Et tant d'autres ...



Technologies RIA avec Javascript JSF ou ASP.Net



JSF (Sun) ou ASP.Net (Microsoft) ne sont pas des technologies RIA à proprement parler

- Elles restent basées sur une logique de scripts serveurs

Leur système de composants côté serveur gérant leur affichage peut très bien s'associer à AJAX

- Gain de productivité substantiel
- Autre moyen de ne pas écrire de Javascript

Un des moyens les plus populaires d'améliorer la convivialité des applications web

- Bibliothèques JSF spécialisées AJAX
 - IceFaces
 - RichFaces



Technologies RIA avec Javascript GWT (Google Web Toolkit)



Solution basée sur une compilation Java en Javascript

- testable hors du navigateur (hosted mode)
- optimisée (taille, rapidité, inlining ...)
- autant de versions que de navigateur * langue

Permet de coder son IHM avec une API à la Swing

- Gain de productivité très important
- Beaucoup moins de technologies à maîtriser
- Possibilité d'intégrer des composants Javascript (JSNI)
- Disponibilité de Google Gears pour gérer un mode déconnecté
 - Plug-in embarquant une base de données dans le navigateur
 - Solution très récente



Technologies RIA avec Plug-ins



FLEX (Adobe)

- Basé sur le player Flash pour l'exécution (plug-in)
- Format MXML pour décrire l'interface graphique
- Langage ActionScript pour le code client

Silverlight (Microsoft)

- Basé sur un plug-in (IE, Firefox, Safari) et prévu pour Linux
- Format XAML pour décrire l'interface graphique
- Langages .Net pour le code client (inclus une mini CLR)

Ces technologies sont concurrentes pour l'intégration de médias (audio, vidéo) dans des sites web

Le RIA aujourd'hui



AJAX ce n'est pas forcément du RIA !!!

Pousser la logique RIA c'est supprimer ses pages

- Médiation côté serveur Http pour déclencher les traitements
 - Webservices, REST, ou autre forme de RPC

Mais de nombreuses limitations apparaissent

- Gestion des boutons du navigateur
- Gestion du mode déconnecté
- Taille et mise en cache des parties clientes
- Performances du navigateur
- Limitations liées à l'intégration dans le navigateur
 - Graphiques
 - Restriction d'accès aux mécanismes de l'OS
 - Sécurité d'exécution



Supprimer le navigateur ?



Beaucoup de limitations naissent du navigateur, or

- Les contenus applicatifs de type RIA n'ont pas toujours besoin d'être référencés
- Les applications peuvent être lancées en ligne
 - JavaWebStart, ClickOnce
- Il ne garantit pas l'universalité du client
 - Javascript reste dépendant du navigateur
 - Lutte d'éditeur pour les plug-in clients :Flash vs silverlight



Alors pour demain



Des applications grand public avec beaucoup de contenus riches

- Utiliserons le navigateur et ses plug-ins

Des applications très riches fonctionnellement et graphiquement (usage spécifique)

- Devront se poser la question de l'abandon du navigateur

Avec des solutions existantes

- Client SWT, Swing communiquant avec des webservices et déployé via JavaWebStart
- Des solutions Desktop comme EclipseRCP
 - La différence avec une solution RIA tendant à disparaître

Avec des solutions nouvelles

- AIR (Adobe) : Un run-time pour faire tourner les applications Flex hors du navigateur
- JavaFX : solution encore peu claire et techniquement peu novatrice

