



# Atelier clients riches web

---

*GWT, Silverlight et Flex*

Xavier Paradon  
Directeur Technique  
Valtech Training

[xavier.paradon\[at\]valtech-training.fr](mailto:xavier.paradon@valtech-training.fr)

# Rich Internet Application

- Rich Internet Application
- GWT
- Silverlight
- Flex

## ■ Manque d'ergonomie

- Sites Complexes
  - Navigation trop profonde au fur et à mesure de l'ajout de fonctionnalités
  - Difficulté de combiner des contenus de nature différente
- Utilisation peu conviviale
  - Chargement bloquant des pages
  - Nombreux cycles requête réponse avec le serveur

## ■ Le web est entré dans les mœurs mais ...

- Les utilisateurs se lassent vite des frontaux web
  - Il faut toujours se renouveler et donc forcément innover
- Les utilisateurs acceptent de moins en moins les contraintes liées au navigateur

## ■ Un RIA ou Rich Internet Application

- Application graphique déployée dans un navigateur
- Exécutant le code de l'IHM côté client
  - Plutôt qu'une interprétation de balises HTML
  - Reposant donc sur un ensemble de composants graphiques
    - *Et un environnement d'exécution*
- Gérant les évènements utilisateurs côté client
  - Plutôt qu'en déclenchant automatiquement une requête HTTP
- Communiquant avec le serveur via des appels de service
  - Type service web (SOAP) ou orientés ressources (REST)
    - *Favorisant un appel asynchrone pour l'utilisateur*
  - Plutôt qu'en appelant du code de présentation (Servlet, page ASP)

# Ben j'ai compris c'est du javascript

## ■ Oui

- Pour la présence de l'interpréteur javascript
- Pour la gestion des évènements
  - Mais elle reste dépendante du navigateur

## ■ Non

- La majorité de l'IHM reste en HTML
- Pas d'API graphique en javascript
  - Même s'il existe des bibliothèques de composants open source
  - Pas d'utilisation courante de widgets graphiques javascript
- Pas réellement d'appel de services
  - Appel synchrone de la couche de présentation
  - Utilisation en mode "page"

# Ah bon, alors c'est de l'AJAX

## ■ AJAX = Asynchronous Javascript And XML

- Utilisation d'une requête asynchrone pour récupérer des données côté serveur

## ■ Oui (c'est possible)

- Programmation côté client
- Utilisation de composants graphiques de rigueur
  - C'est la grosse différence entre le javascript d'hier et d'aujourd'hui
- Appels orientés service
  - Remise en question du mode "page"

## ■ Non

- En réalité la plupart des applications web utilisent AJAX pour résoudre localement un problème d'ergonomie

# Ils sont forts ces consultants

## ■ A.K.A c'est encore une invention marketing

- Le client riche répond rarement à un besoin fonctionnel

## ■ Avantages possibles du RIA

- Richesse de l'interface (of course), meilleure ergonomie
- Diminution du nombre de technologies
  - Et environnement de développement cohérent
- Serveurs en mode stateless (et clients statefull)
- Souplesse architecturale : découplage technologique avec le serveur
  - À la mode avec SOA
- Cohérence accrue entre les couches (présentation et service)
- Gestion du mode déconnecté

## ■ C'est une évolution assez naturelle des technologies web

## ■ RDA = Rich Desktop Application

- Application graphique déployée en dehors du navigateur
- Avec un système de gestion de versions simplifié
  - Mise en cache des applications
  - Téléchargement et installation des évolutions
- Reprenant les grandes lignes techniques du RIA

## ■ Exemples :

- EclipseRCP
- Java Web Start
- ClickOnce Deployment
- AIR (Adobe Integrated Runtime)

## ■ Les limites RIA et RDA deviennent floues

- Le RDA communique souvent par HTTP avec le serveur
- Le RIA dispose d'un runtime (CLR, JVM, FlashPlayer)

## ■ Pourtant le RIA permet

- Le référencement des contenus
- La syndication de contenus
- Une installation transparente

## ■ Dimension de partage vs dimension applicative

# Solutions : AJAX avec bibliothèques

## ■ Qui veut écrire un RIA en javascript ?

- Peu d'aide à l'écriture
- Dur à débbugger
- Problèmes Cross-browser
- Manque de compétences

## ■ Succès des librairies : Dojo, Ext, JQuery

- Mais aussi Scriptaculous, Rico, Ajax spry, backbase, Yahoo UI, Rialto ...
- Leur pérennité est toujours sujette à caution
- L'environnement de développement s'améliore doucement

# Autres solutions

## ■ OpenLaszlo

- Solution Open-source concurrente de Flex
- Format lzx : description XML + ECMAScript API's

## ■ JavaFX

- Solution Sun basée sur JavaFXScript (mais pas Java !!!)
- Devrait disposer d'un plug-in pour les navigateurs
- La stratégie n'est pas évidente même si Sun clame que c'est du RIA

## ■ Echo2

- Framework Java générant du javascript à la volée
- Solution proche de GWT

## ■ Et d'autres

- MS Volta
  - Solution Microsoft concurrente de GWT
- XUL
  - Solution technique de Firefox

# Google Web Toolkit

- Rich Internet Application

- **GWT**

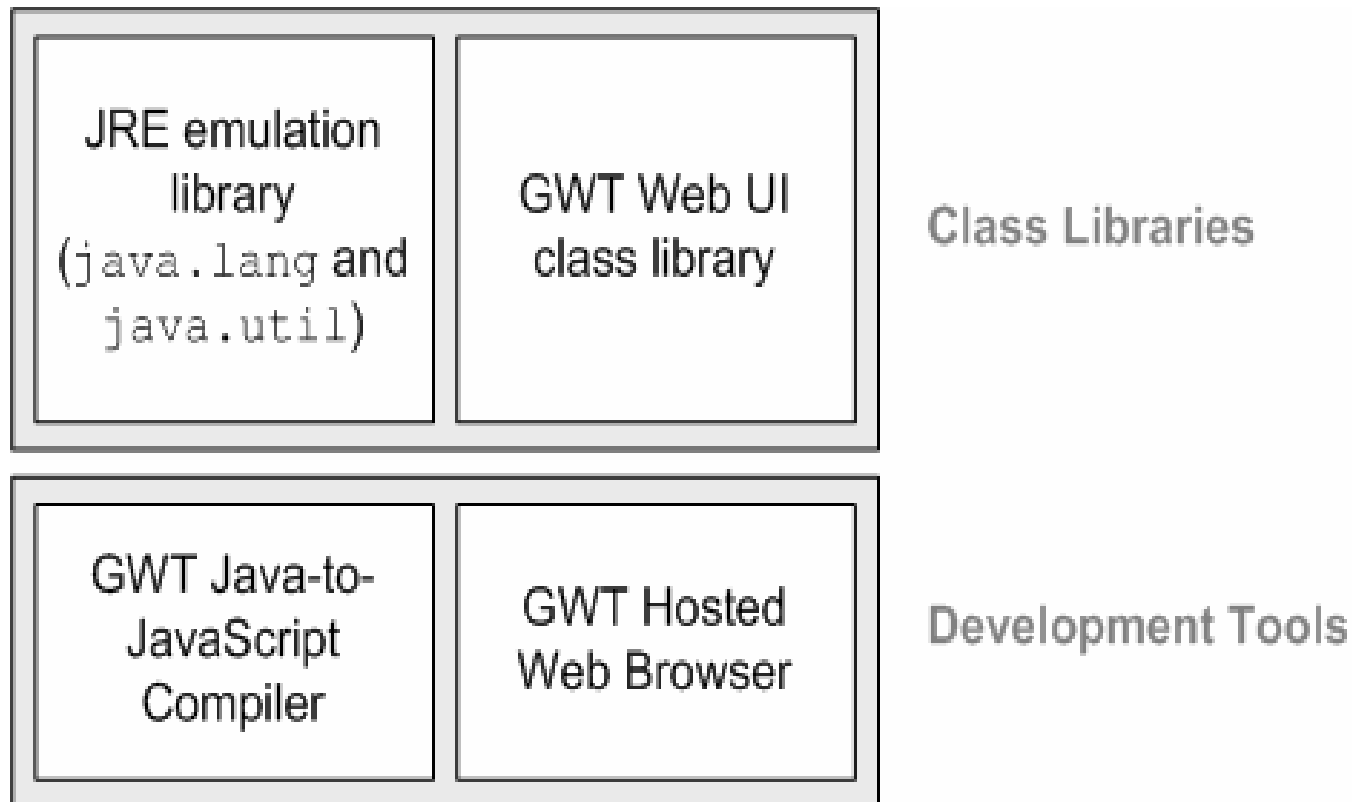
- Silverlight

- Flex

### ■ GWT est un framework de développement web open source

- Dernière version : 1.5RC1
- Basé sur un pari audacieux :
  - Ne pas écrire de code javascript
  - Ecrire son application en Java
  - La compiler en javascript
- Puisque les compétences Java sont plus nombreuses
- Puisque Java offre des garanties à l'écriture du code
  - Typage fort, compilation, gestion des exceptions ...
  - Utilisation d'IDE (et de composition visuelle)

# GWT en morceaux



# Avantages de GWT

- **Une API à la Swing**
  - Donc des composants graphiques écrits en Java
- **Un environnement de développement Java (Eclipse ...)**
- **Un environnement de test simple**
- **Un code Debuggable**
- **Un code javascript cross-browser et internationalisé**
  - IE, Firefox, Opera, Safari
- **L'intégration de bibliothèques javascript avec JSNI**
- **Une gestion du Look&Feel en CSS**
- **L'optimisation du code javascript**
- **Un modèle de communication avec le serveur (RPC)**
  - Plusieurs solutions (serveur pas forcément en java)
- **La Gestion de l'historique**
- **L'intégration avec Gears pour le mode déconnecté**

# Silverlight

- Rich Internet Application
- GWT
- Silverlight
- Flex

## ■ Solution RIA Microsoft basée sur

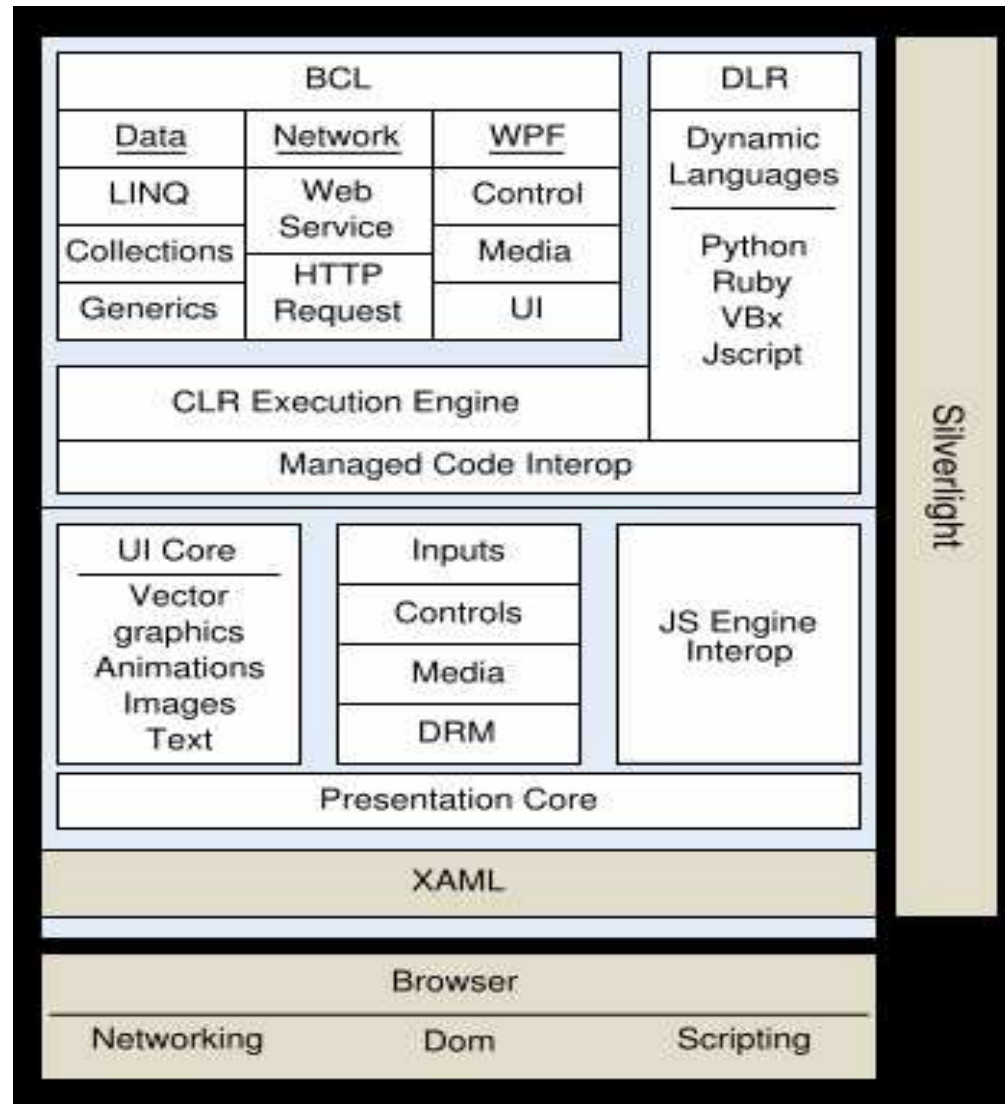
- Le plug-in Silverlight 2
  - Actuellement Silverlight 2 Bêta 2 (4,5 Mo)
  - Disponible pour Windows et Mac (IE, Firefox, Safari)
  - Support Linux prévu avec le projet Moonlight et pour Opera
    - <http://www.mono-project.com/Moonlight>
  - Support prévu pour environnement mobile (Windows Mobile, Symbian)
- Une foule d'outils et de produits
  - Gamme Expression, Visual Studio 2008
- Techniquement
  - XAML pour la description des interfaces graphiques
  - C# (ou autres langages .Net) pour coder les interactions
    - *Le plug-in embarque une CLR identique à celle de .Net 3*
    - *Les bibliothèques du framework sont évidemment dégraissées*
  - Le support graphique et multimédia pour concurrencer Flash

# Silverlight ou Silverlight 2

## ■ La première version de Silverlight

- Contient beaucoup moins de contrôles graphiques
- Format de données limités
  - Pur XML et JSON
- Utilise uniquement javascript pour coder côté client
  - Silverlight 1 = 1 fichier XAML + 1 fichier Javascript
  - Silverlight 2 = 1 fichier XAP (compilation code+XAML)

# La pile Silverlight



# Détails techniques supplémentaires

## ■ Dynamic Language Runtime

- Ruby, Python, javascript

## ■ Communication avec le serveur

- REST, Web services, WCF ...

## ■ Contrôle d'accès aux ressources et sécurité d'exécution

- Proche d'un sand box et permet
  - Cross domain sockets
  - Background threads et server push

## ■ Un cache client (isolated storage)

## ■ Côté graphisme et vidéo

- Streaming video, DeepZoom, gestion des DRM

# Flex

- Rich Internet Application
- GWT
- Silverlight
- Flex

## ■ Framework de développement open source

- Flex3SDK + BlazeDS (communication serveur Java)
  - Un client Flex peut communiquer avec différentes technologies serveur
- Mais avec un EDI payant : FlexBuilder
  - Version standalone ou plug-in eclipse
- Avec des composants supplémentaires sous licence
  - Serveur : FlexDataServices (a.k.a LiveCycle DataServices ES)
  - Client : Charting Components

## ■ Flex 3 est basé sur

- MXML pour la description de l'IHM
- Action Script 3.0 pour le code client
  - Fortement (et faiblement typé) et orienté objet
- Feuilles de styles à la mode CSS
- Le compilateur MXMLC
  - Compile MXML+CSS+ActionScript
  - Intègre les ressources et le code compilé dans un format .swf
  - Ligne de commande, script Ant, FlexBuilder, WebCompiler (JSP like)

## ■ Communication Serveur

- Appels HTTP, Web services
- AMF (Action Message Format)
  - Sérialisation binaire d'objets ActionScript
  - DataServices nécessaires côté serveur

# Flex 3 côté open source

- **Le sdk est gratuit et open source**
- **Rien d'imposé côté serveur**
  - Plus facile en Java (comme pour GWT) avec BlazeDS
- **Des librairies open source de composants additionnels**
  - ASTRA (ActionScript Toolkit for Rich Applications) de Yahoo
    - <http://developer.yahoo.com/flash/>
  - Flexlib
    - <http://code.google.com/p/flexlib/>
  - Ventanas
    - <http://code.google.com/p/ventanas/>
- **Cairngorn : framework MVC**
  - Ecrit en ActionScript 3.0
  - Mais pas encore disponible pour Flex3
  - <http://labs.adobe.com/wiki/index.php/Cairngorm>
- **En pratique FlexBuilder est très utilisé**

# Comparatif et conclusion

## ■ A chacun de se faire son opinion

- Les technologies sont assez récentes
- Les retours d'expérience réels sont peu nombreux
- Les possibilités d'amélioration sont légion

## ■ Les ateliers sont là pour ça

- Facilité de prise en main et outillage
- Compétences requises
- Accessibilité et portabilité des solutions

## ■ Très objectivement

- Techniquement GWT diffère de Flex/Silverlight
  - Pas de rendu vectoriel et autres sophistications graphiques
- Le choix entre Flex et Silverlight est une question d'éditeur
  - Et dépendra beaucoup de la pénétration de Silverlight sur le marché
  - Flash équipe 98% des navigateurs selon Adobe