



Silverlight et consors

Les annonces récentes de Microsoft Silverlight, Adobe Apollo et des évolutions de Flex, tout comme celle de JavaFX de Sun donnent un peu le tournis dans le monde des “Rich Internet Applications”. Il y a de quoi y perdre son latin car ces technologies sont toutes concurrentes, mais elles présentent des différences certaines que leurs présentations marketing ne permettent malheureusement pas d’identifier clairement.

Les annonces s’appuient à la fois sur les forces de leurs auteurs, et aussi de leurs faiblesses sur le marché et de leurs ambitions pour les résorber. Tout en surfant sur la vague du Web riche, interactif et multimédia.

Pour faire simple:

- **Microsoft** cherche à séduire les concepteurs et designers de sites Web qui sont férus de Flash, alors qu’ils ont déjà plutôt la cote avec les développeurs. Ils veulent se donner une image “multi-plate-forme” pour limiter leur image “propriétaire liée à Windows”. Et se positionnent nettement sur les contenus riches, surtout vidéo.
- **Adobe** cherche à séduire les développeurs en entreprise avec Flex et Apollo, des outils plus “sérieux” que Flash. Avec une incursion discrète dans le middleware serveur et une architecture de client riche.
- **Sun** pousse l’avantage multi-plate-forme de Java avec JavaFX, ciblant à l’instar de Microsoft à la fois les postes de travail, les mobiles, mais également les set-top-boxes. Mais avec la volonté de simplifier le développement Java d’applications Internet riches.
- L’**open source** reste créatif, mais quelque peu fragmenté et sans solution d’ensemble répondant aux besoins du “rich web”. Mais comme ils dominent le reste du web avec les solutions LAMP, ils sont tranquilles pour quelques temps.

Je vais dans ce post tenter de décortiquer ces annonces en commençant par Microsoft.

Microsoft et Silverlight

Silverlight a été annoncé pendant la conférence MIX de Las Vegas de fin avril 2007. C’est une technologie de plug-in pour navigateur permettant d’exécuter des composants logiciels “riches” avec toutes sortes de contenus interactifs, notamment vidéo. Elle ne part pas de zéro puisque c’est le nouveau nom donné à ce qui était jusqu’à présent le WPF/E (Windows Presentation Foundation / Everywhere) dont une version d’évaluation était sortie en décembre 2006.

Silverlight présente plusieurs caractéristiques intéressantes:

- C'est la mise en oeuvre de l'architecture **.NET dans un plug-in** pour navigateur Web avec l'incorporation des technologies que l'on retrouve dans .NET et dans Vista: le framework Windows Presentation Framework pour la création d'interfaces graphiques riches et dynamiques, le langage XAML pour la description en XML de ces interfaces, et le support du multilingage (dans la version 1.1 de Silverlight, la 1.0 se contentant de XAML et de JavaScript). C'est donc un bon moyen de promouvoir les interfaces de programmation de Vista au delà de ce système d'exploitation. Avec une caractéristique qui semble faire la distinction avec les concurrents: une excellente rapidité d'exécution des applications.
- C'est un composant adapté au multimédia, très **puissant pour les interfaces animées et la vidéo**, supportant la HD (720p) en streaming ou download. A ce titre, Silverlight devrait être une technologie de choix pour les applications grand public de commerce électronique et de télévision numérique, sur PC comme sur set-top-box. Et aussi pour le marché publicitaire, toujours à la recherche de formats originaux et interactifs. Silverlight arrive à point nommé alors que la vidéo envahit littéralement le Web. La technologie repose sur le codec SMTPE VC1 (Windows Media Video) dont la performance en streaming est probablement la meilleure du marché (taux de compression, qualité d'image, charge de serveur).
- Il remet au goût du jour la notion de "**client riche**". La nuance entre client riche et client léger est d'ailleurs de plus en plus ténue, avec tous ces plug-ins popularisés par Flash et autres players média. C'est aussi la consolidation d'un web dont de nombreuses composantes sont et restent propriétaires. Microsoft parle de "Logiciel + service" et pas juste de "Software as a service". C'est la traduction d'une réalité simple: les solutions logicielles d'aujourd'hui qui exploitent Internet sont la combinaison de logiciels qui tournent sur votre ordinateur (fixe, portable, set-top-box, montre, whatever...) et de logiciels tournant sur des serveurs (qu'on appelle "services"). Bref, après quelques contorsions sémantiques, c'est le bon vieux concept du "client-serveur" qui survit aux générations technologiques. Microsoft se positionne d'ailleurs des deux côtés de la chaîne de valeur avec tout ce qu'il faut du côté client (Silverlight, Vista, clients riches, semi-riches, légers, etc) et du côté services (ce que propose Live: Hotmail, Photos, Contacts, Virtual Earth, Search, authentification et le nouveau **Silverlight Streaming**, un service web de streaming vidéo avec 4 Go de stockage gratuit).
- Une **offre très intégrée** d'outil de développement d'applications riches: Expression Studio qui comporte quatre composants (Expression Web pour la création d'applications Web en remplacement de FrontPage, Expression Blend pour la création d'applications desktop, Expression Design pour le dessin vectoriel et bitmap – résultat de l'acquisition de Creature House en 2003 et Expression Media pour l'encodage et la gestion des médias – issu de l'acquisition de iViews en 2006). Expression Studio peut être **évalué gratuitement** par brique. Cette intégration suit un découpage fonctionnel par métier (développeur, designer, média manager) reprenant une vision très taylorienne de la conception de site Web riche. Mais l'approche a ses limites puisqu'Expression Studio ne supporte pas encore l'intégration d'outils de gestion de version (CVS like). On se retrouve donc avec une approche du développement à trois couches: médias/contenus/design, site web, et composants logiciels, ces derniers restant développés avec Visual Studio. L'ergonomie, le look et la cohérence de la suite Expression sont vraiment attirants, par comparaison à ce que l'on trouve chez Adobe ou Sun qui reste très austère (*cf exemple avec Blend ci-dessous*).



- Une **ouverture qui s'étoffe mais qui reste limitée** : le plug-in Silverlight fonctionne sous Windows (y

compris Windows Mobile) et MacOS. Il supporte Internet Explorer, Firefox, Opera et Safari. Linux n'est pas au programme et ne le sera pas avant longtemps car Microsoft évite de toucher à ce système par peur du côté viral de la licence GPL (même s'il a des offres qui touchent bien à Linux notamment dans le domaine de la virtualisation). Microsoft s'appuiera donc sur le projet Mono mené par Miguel Iquaza chez Novell pour faire fonctionner Silverlight sous Linux. Du côté développement, Silverlight peut être programmé avec le très populaire langage objet **Ruby**, en plus des langages classiquement supportés par .NET tels que C# ou Python. L'ouverture n'est par contre pas le fort de Silverlight du côté de la vidéo : il ne supporte que le format SMTPE VC1 (de la HD au mobile). Aucune information n'a filtré sur une éventuelle architecture de codecs pour supporter d'autres formats vidéo. Donc, des conversions de formats sont prévues au programme – avec Expression Media Encoder – pour les Divx, H264, MPEG2 et autres MPEG4. En audio, le choix est légèrement plus ouvert avec le support du WMA et du MP3. Enfin, et ce n'est pas de l'ouverture mais plutôt de la capture de part de marché de concurrent, un outil de conversion est fourni pour passer de Flash à Silverlight.

D'un point de vue pratique, l'**installation de Silverlight** (1.0 beta) est rapide, avec un téléchargement de moins de 2 Mo. Tout du moins sous Windows Vista ou sous Windows XP SP2 qui disposent déjà du framework .NET. Dans le cas contraire, on est bon pour 50 Mo de téléchargement. Une fois le plug-in installé, on peut tester quelques petites applications de démonstration comme cet outil de présentation de **bandes annonces de Fox**:



En fait, il y a coexistence entre la bêta de Silverlight 1.0 (2 Mo) et une alpha de Silverlight 1.1 (4 Mo). C'est vraiment un comble! Cette dernière version qui seule supporte la CLR et les langages de programmation de .NET permet de visualiser quelques démonstrations comme ce Silverlight Airlines qui permet visuellement d'identifier le meilleur vol et parcours pour aller d'une ville à une autre des USA:



Dans son lancement de Silverlight, les premières références clients de Microsoft étaient très orientées 'médias' et réduites aux USA: CBS (pour un site communautaire en cours de création, à la MySpace), Fox que l'on a déjà vu, Major League Baseball (télévision en direct et à la demande), Skinkers (push streaming vidéo sur mobiles), Limelight Delivery Network (distribution de contenus médias) et enfin Netflix pour la consultation d'extraits vidéos.

Il y avait également quelques logiciels ou matériels en tirant parti: **Roxio Buzz**, un outil gratuit de montage vidéo relié aux plates-formes de streaming comme Silverlight Streaming (version payante pour supporter les

codecs vidéo comme le MPEG2, MPEG4 ou AAC d'Apple), Topbanana de Metaliq, un outil de montage vidéo équivalent issu d'une jeune startup, et enfin **Tarari** qui annonce un encodeur vidéo matériel (pour le VC1 de Silverlight).

Bref, avec Silverlight, on bénéficie de spécificités de .NET et de Microsoft: un choix de langages de programmation et d'outils de développement modernes, la meilleure performance d'exécution et une plateforme d'outils de création adaptée à tous les métiers. Par contre, c'est malgré un peu d'ouverture, un monde assez "Microsoft centric". Vous voulez en savoir plus? L'occasion vous en est donnée en participant à la conférence **Mix à Paris, au Cirque d'Hiver, le 21 juin**. Vous pouvez aussi lire divers compte rendu de l'annonce de MIX Vegas, notamment **chez Fred Cavazza**. Et aussi la **description assez complète** de Silverlight faite par Scott Guthrie de Microsoft Corp.

Au passage, les rumeurs sur l'acquisition de Yahoo par Microsoft me semblent peu fondées. Contrairement à l'idée répandue chez ses détracteurs, Microsoft fait très rarement l'acquisition de concurrents, à fortiori de grande taille. Les redondances entre MSN/Live et Yahoo sont trop conséquentes (mail, messagerie instantanée, portail, blog, partage photo/vidéos, etc) pour qu'une intégration puisse fonctionner sans pertes et fracas. Par contre, il n'est pas impossible que Microsoft discute avec Yahoo de partenariats technologiques divers. Pourquoi pas, relatifs à l'usage chez Yahoo de Silverlight! Mais je peux me tromper, on verra bien...

Sinon, Microsoft vient de faire, comme ça, **l'acquisition de aQuantive** hier pour la modique somme de \$6B. C'est sa plus grosse acquisition historique. Faute d'avoir pu gober DoubleClick, attrapé par Google il y a quelques semaines pour \$3B. aQuantive est l'un des principaux concurrents, certes moins connu, de DoubleClick. Il faut dire que aQuantive avait un atout de taille: ils sont basés à Seattle! Mais ceci n'a pas de rapport avec Silverlight, mais plus avec la volonté de Microsoft de monétiser ses services Internet par la publicité, face à l'omniprésent Google.

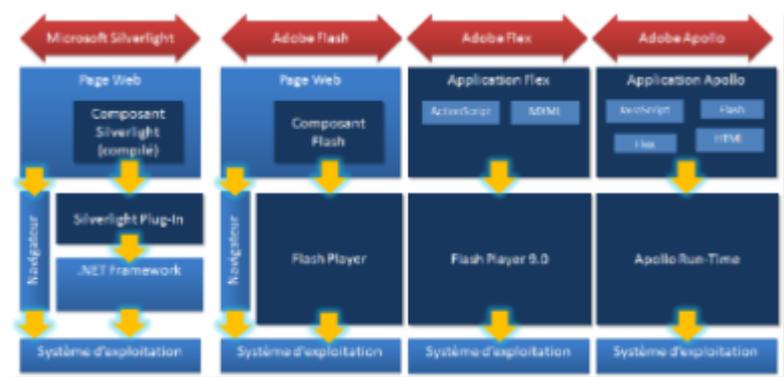
Adobe, Apollo et Flex

Apollo d'Adobe est l'équivalent le plus proche de Silverlight. A ceci près que ce run-time permet de mixer dans le même composant logiciel du code HTML, JavaScript en plus du Flash et au passage du PDF.

Un plug-in Silverlight sera écrit dans un langage supportant .NET, puis fera appel à XAML pour la description graphique de l'application. Et il sera exécutable dans le contexte d'une page HTML, elle-même rendue par un navigateur. Alors qu'un composant Appolo est en fait une application qui s'exécute directement à partir du bureau.

Du côté outil de développement, Adobe propose Flex pour créer ces applications Apollo. Flex comprend **Flex Builder**, un environnement de développement (IDE) basé sur Eclipse (*illustration ci-dessous*), Flex Framework, un modèle de programmation, et **Flex Data Services**, un service d'accès (écrit en Java) aux données gérant la synchronisation entre client et serveur. L'interface des applications Flex est codée avec MXML, une sorte d'équivalent du XAML de Microsoft.

Sur le poste de travail, l'application Apollo s'exécute avec un run-time Apollo. Une application Flex se contente du Flash Player. Dans les deux cas, on a une bonne vieille application "client riche" dotée des moyens de communication avec les serveurs sur Internet. Voici pour résumer tout cela une cette comparaison schématique d'architecture entre les applications Silverlight, Flex et Apollo (*cliquer dessus pour agrandir le schéma*):



Au delà de cette notion de client riche “standalone”, ce qui est un peu nouveau dans l’architecture d’Adobe, c’est l’intrusion dans le serveur. Jusqu’à présent, un plug-in Flash s’exécutait tout seul sur le client, sur lequel il avait été téléchargé comme simple fichier à partir d’un serveur. Une application Flex ou Apollo feront appel aux Flex Data Services tournant sur serveur. Ce n’est probablement qu’un début et il ne serait pas étonnant de voir un jour Adobe étoffer son offre de middleware serveur à partir de cette tête de pont.



Un avantage à signaler pour l’offre Adobe: elle est “plus” multiplateforme car l’éditeur supporte lui-même Linux. Même s’il ne supporte officiellement que RedHat et Novell Suse. Sa stratégie pour les mobiles semble par contre moins avancée que celle de Microsoft et de Sun, que l’on verra juste après.

Microsoft et Adobe ont deux approches propriétaires qui se battent en duel. Ils rivalisent d’ouverture dans leur monde propriétaire: Microsoft a fait standardiser le VC1, le C#, la CLR, et a mis en open source la DLR, un bout du run-time de Silverlight. Mais le langage script de Flash, Action Script, est aussi un standard ECMA, organisme convertissant régulièrement des spécifications propriétaires en standards de jure. L’annonce de la mise des outils Flex 3.0 en open source suit cette tendance. Quand on lit entre les lignes, on constate que seuls certains composants de Flex seront mis en open source. Et que cela ne concerne pas les extensions à Eclipse ni les Flex Data Services. L’approche open source est donc utilisée plus à des fins marketing qu’autre chose, tout en préservant le côté propriétaire des composants clés de la plate-forme Flex et Apollo. A l’instar de Microsoft avec Visual Studio et Expression, le Flex Builder d’Adobe reste bien payant.

Adobe a pour lui la chance qu’a Microsoft dans d’autre domaine : l’inertie de marché. Qui fait qu’il sera difficile de déloger Flash. Le web continuera donc d’être un bric à brac qui voit se côtoyer de nombreuses technologies ouvertes et propriétaires, avec une palanquée de plug-ins et autres composants logiciels incompatibles entre eux.

Avec Adobe, on est donc dans le standard du moment, un peu plus proche des standards Web, et on est plus multi-plate-forme qu’avec Microsoft.

Sun et JavaFX

Sun a annoncé JavaFX quelques jours après l’annonce de Silverlight par Microsoft. On pourrait dire: “en réaction à”. D’un point de vue du timing, c’était à l’occasion de la conférence JavaOne qui avait alors lieu, et de toutes manières, ces choses là ne se préparent pas en quelques semaines. Même s’il apparait que l’annonce est maigre et ne constitue pas une avancée autant que celles d’Adobe ou de Microsoft.

JavaFX Script est un langage de scripting qui fonctionne sur toute machine virtuelle JavaSE. Sorte d’intermédiaire entre Java et JavaScript, il s’appuie sur la bibliothèque de composants Swing servant à décrire les éléments de l’interface graphique. Mais l’ensemble ne semble pas permettre la création d’applications aussi riches visuellement que Silverlight ou Flex.

Alors, où est l'interpréteur de ce langage? L'histoire ne le dit pas et le marketing de Sun est assez obscur de ce point de vue là. L'interpréteur doit se charger avec l'application, étant lui-même écrit en Java. Ou bien JavaFX n'est qu'un nom marketing pour un sous-ensemble de Java ou de JavaScript. En tout cas, ceci n'augure rien de bon en termes de performances. Les applications JavaFX sont des fichiers .jar chargés à partir d'un navigateur et qui s'exécutent ensuite dans leur propre fenêtre.

Sun propose aussi JavaFX Mobile, un middleware pour mobile permettant d'exécuter ces nouveaux scripts. Un bon point pour Java qui est effectivement très présent sur ce marché.

Autre bon point pour Sun et Java, le côté multi-plate-forme qui est bien meilleur que pour Microsoft Silverlight et Adobe Flex/Appolo. Car Java supporte à peu près tout ce qui existe comme système d'exploitation. De plus, JavaFX en **open source**, of course.

On développe les applications JavaFX à partir de NetBeans, l'environnement de développement gratuit de Sun écrit lui-même en Java.

Donc, what's new? Pourquoi a-t-on besoin d'un langage de scripting lui-même tournant au dessus de Java? La littérature indique que cela permet d'ouvrir Java à un plus grand nombre de développeurs. Mais n'est-ce pas le rôle de JavaScript? Mystère pour moi...

Sur le site de JavaFX, on pouvait tester quelques petites applications de démonstration comme celle de Motorola qui a l'air bien animée et rapide:



Et une autre de Tesla Motors. Mais, différence notable par rapport aux démonstrations de Microsoft et Adobe: il n'y a pas de vidéos! Pourtant, cela doit être possible.

Au bilan, avec Sun et Java, on n'est pas très avancé, sauf peut-être sur les mobiles où Java est très présent.

Solutions open source

Le monde de l'open source est très divers et son approche est donc assez fragmentée. Même si elle présente l'approche de ne pas être propriétaire et proche des standards du Web.

Historiquement, les évolutions du développement Web dans ce monde passaient par Ajax et ses évolutions comme XAP (soutenu par Nexaweb) ou implémentations diverses comme chez Helmi, tout comme XUL – un vague équivalent open source du XAML de Microsoft – et SVG, un format d'image vectoriel 2D dérivé de XML. (NB: Microsoft fournit aussi une implémentation d'Ajax sur ASP.NET avec Atlas).

C'est le projet **OpenLaszlo**, d'origine italienne, qui semble le plus proche de Silverlight et de Flex. Il s'appuie sur un langage LZX de description d'interface graphique similaire à XUL (Mozilla Foundation), XAML (Microsoft) et MXML (Adobe) et sur un logiciel serveur qui compile le code pour l'envoyer aux navigateurs au

format Shockwave reconnu par le player Flash ou bien en JavaScript. Les **quelques démonstrations** d'applications réalisées avec sont assez classiques dans la forme.

Enfin, du côté du consortium ObjectWeb, il n'y a **pas de signe de vie** d'une plate-forme intégrée pour le "rich web". Ce qui s'explique par l'orientation middleware serveur et développement d'applications en entreprise du consortium et de ses différents membres.

A ce stade, aucune solution open source n'a réussi à véritablement égratigner Adobe côté part de marché, ou Microsoft du côté de la cohérence de l'architecture.

Conclusion

Je constate au vu de ces acteurs du marché que leur marketing est assez cryptique. Leurs explications contiennent peu de schémas qui expliquent qui fait quoi, quel logiciel tourne où et s'interface avec quoi. Le manque de pédagogie est flagrant. Adobe n'est pas trop mauvais, Sun est une catastrophe, et Microsoft se situe entre les deux.

Je note aussi une conséquence de l'émergence de ces applications "riches" qu'il ne faut pas négliger: elle peut fermer l'Internet et notamment empêcher l'indexation de contenus, s'ils sont tous encapsulés dans ces applications.

Bref... net net, Adobe et Microsoft restent les deux acteurs en lice dans ce web propriétaire!

Encore faut-il faire la part des choses sur le marché concerné. Il s'agit ici du marché des "applications riches". Mais l'essentiel du web, ce sont des applications "pauvres" – certes dynamiques – qui n'ont pas besoin d'être "riches". Donc, les applications Web développées avec des outils plus traditionnels comme le bon vieux PHP et le JavaScript, continueront de dominer l'Internet pendant longtemps.

Article très légèrement mis à jour le 20 mai 2007

Cet article a été publié le 19 mai 2007 et édité en PDF le 15 mars 2024.
(cc) Olivier Ezratty – "Opinions Libres" – <https://www.oezratty.net>