

L'avenir d'internet dans la mobilité

Pierre Le Roux

Remerciements

Je souhaite remercier la société Sogeti pour la confiance qu'elle m'a accordée durant mon stage.

Dominique Colonna pour la mission qu'il m'a confiée dans le domaine de la mobilité.

Sylvain Labasse pour l'aide qu'il m'a apportée dans l'élaboration de mon mémoire de fin d'études.

Sommaire

Introduction.....	1
I Les applications web dans la mobilité.....	8
1- Évolution des technologies web	9
2- Différents types d'application mobile	20
3- L'enjeux économique des applications web dans la mobilité	28
II Exigence des applications web dans la mobilité.....	33
1- Les applications connectées/déconnectées	33
2- La réactivité des applications internet.....	39
3- La compatibilité sur différentes plates formes	44
4- Les limites des applications web	49
III Spécificité des plates formes	54
1- Développement natif, les différents stores.....	55
2- Développement serveur.....	58
3- Les applications hybrides	62
Conclusion (7-8pages)	74

Introduction

« Internet sera à l'économie du 21^e siècle ce que l'essence fut au 20^e siècle. La puissance des ordinateurs c'est l'essence d'internet. » Citation de Craig Barret (ancien président d'Intel). L'homme est nomade par nature, l'essence l'a aidé à se déplacer plus rapidement via des voitures ou même des avions. Dans cette optique de déplacement, nous avons observé, depuis 4 ans, une montée en flèche des ventes d'appareils mobiles (1,2 milliard d'unités vendues en 2013). Cette croissance signe un tournant dans le monde numérique tout comme internet avant lui. Internet, qui a su démontrer son importance dans l'univers multimédia. Importance marquée tout particulièrement par son évolution exponentielle depuis plus de 50 ans.

Une évolution qui se dénote tant en terme technologique qu'en terme d'utilisation. D'après la CIA World Factbook, en 2008 on ne dénombra pas moins de 1,6 milliard d'abonnés à Internet contre 61 millions en 1996.

L'évolution d'internet dénote plusieurs dates importantes :

- 1962 : début des recherches par ARPA, un projet du ministère de la Défense américaine
- 1969 : connexion des premiers ordinateurs entre quatre universités américaines (UCLA, Stantford, l'Utah, UCSB)
- 1971 : 23 ordinateurs sont reliés sur ARPAnet.
- 1973 : l'Angleterre et la Norvège rejoignent le réseau Internet avec un ordinateur par pays
- 1979 : création des NewsGroups (forums de discussion) par des étudiants américains
- 1982 : définition du protocole Tcp/ip et du mot « Internet »
- 1983 : apparition du premier serveur de noms de sites
- 1990 : disparition d'ARPAnet
- 1991 : annonce publique du World Wide Web

À partir de 1992, plus d'un million d'ordinateurs étaient connectés au réseau, puis en 1996, soixante et un millions étaient connectés. Enfin en 2000 c'est l'explosion de la bulle d'internet avec plus de 368 540 000 d'ordinateurs connectés.

La croissance du nombre d'ordinateurs connectés entre 1992 et 1996 s'explique par l'apparition du premier navigateur internet, appelé NCSA Mosaic, en 1993, ouvrant internet au grand public.

Cette ouverture au marché public a permis entre autres l'envoi de mail ou même l'accès à des pages web définies comme des sites internet (site web). Ces sites internet ont permis un échange d'information, une révolution pour l'époque. On notera ainsi, l'apparition de moteurs de recherche comme Google (1998) et Yahoo (1995), ou dans l'échange de bien comme EBay (1996). On remarquera aussi le site « Comment Ça Marche », site internet français, créé en février 1999, qui illustre bien l'échange de savoir et d'information.

En 2003, Dale Dougherty invente le terme « Web 2.0 », le web 2.0 signe un tournant dans le monde d'internet. Les internautes pourront, en plus d'acquérir des informations, échanger eux-mêmes des données. C'est la naissance croissante d'applications web comme Facebook ou Wikipédia qui développe l'échange entre les internautes.

Les applications web peuvent être identifiées comme des blogs, des jeux en lignes, des sites commerçants (Amazon, EBay...). À la différence des sites web qui possèdent des pages statiques, les applications web proposent des contenus dynamiques et donc modifiables depuis un outil adéquat souvent appelé « back-office ».

Aujourd'hui, les « sites web », à proprement parlé, sont de moins en moins connus du grand public. Il s'agit de site « vitrine » qui présente des informations sur

une activité, parfois composé de cinq ou six pages statiques.

Les applications web n'étant pas toujours connectées à internet, elles peuvent être installées sur un réseau local ou chargées sur une seule machine. Certaines peuvent allier plusieurs comportements comme par exemple Deezer, service d'écoute de musique, qui propose d'écouter de la musique sur internet et depuis 2012 propose d'écouter de la musique en mode hors connexion.

Le réseau Internet a évolué en terme de technologie, mais aussi en terme de connexion. On remarquera différentes manières de communiquer avec un fournisseur d'accès internet :

- Ligne téléphonique fixe (ligne analogique, ADSL...),
- Ligne téléphonique mobile (protocole WAP, EDGE, 3G, 4G+...),
- Ligne par satellite,
- Wi-Fi,
- Fibre optique, plus récemment.

Le système de connexion qui m'a le plus intrigué est celui par ligne téléphonique mobile. D'une part pour sa croissance en volume de connexion plus qu'impressionnant depuis ces dernières années, et d'autre part pour les futures fonctionnalités qu'elle pourra offrir.

Pour la partie « nombre de connexion », le monde du mobile et d'internet a connu un vrai changement lors de la sortie des premiers « smartphones* », vers la fin des années 2010. Pourtant il était possible d'obtenir un accès à internet via un téléphone portable depuis bien plus longtemps. En France par exemple le Nokia 7110, sortie en mars 2000, est le premier téléphone portable utilisant le service WAP. Mais les inconvénients des premiers services de connexion à internet sont nombreux.

Le service WAP a rapidement été succédé par le « General Packet radio Service »

(GPRS), à cause d'un échec tant technologique que commercial.

En effet, les téléphones de l'époque ne proposaient pas, pour la plupart, de clavier « azerty » ou « qwerty », les écrans étaient minuscules, la vitesse de connexion était proche du Minitel (avec la fiabilité en moins), la facturation excessive est aussi un point non négligeable pour l'époque.

Le service WAP comptait 70.000 utilisateurs en Europe, 4 mois après sa sortie, d'après la société de conseil Ovum.

La connexion GPRS voyait donc son arrivée près d'un an après la sortie du service WAP. Cette technologie n'était qu'une amélioration du WAP. Là où le WAP offrait un débit de 9,6 kbit/s, le GPRS proposait un débit théorique de l'ordre de 171,2 kbit/s, qui en pratique était de l'ordre de 50 kbit/s. Cette amélioration permettait une utilisation plus pratique d'internet et un accès plus direct aux emails.

Même avec un débit multiplié par 18 (en théorie), le réseau n'était pas encore capable de s'adapter aux technologies du futur. Les flux vidéo demandant plus de ressource par exemple.

L'orientation vers un réseau UMTS aussi appelait 3^{ème} génération (3G), étant à l'époque bien trop coûteux, pour les opérateurs téléphoniques, mais offrait un réel débit de connexion. Dans un souci économique, les différents opérateurs choisissaient de s'orienter vers une amélioration du GPRS, c'était la naissance de le « Enhanced Data Rates for GSM Evolution » (EDGE).

L'EDGE est une vraie réponse au problème de faible débit. Ce nouveau type de connexion offrait un débit maximum théorique de 384kbit/s, en pratique ce débit sera réévalué à 200 Kbit/s. Même si l'on était déjà bien ancré dans la connexion ADSL à domicile, ce nouveau débit, toujours utilisé aujourd'hui, était un réel pas en avant pour l'époque. Aujourd'hui, l'EDGE est encore utilisé pour couvrir des zones non desservies par le réseau 3G. De plus, l'EDGE a connu une évolution en 2007, s'appuyant sur la technologie GERAN (GSM Edge Radio Access Network) qui utilise

les infrastructures GSM (exploitant le service WAP), le but était d'augmenter le débit à plus de 400 kbit/s.

La troisième génération (3G) marque un tournant décisif dans l'ère de la téléphonie mobile, le débit pouvant atteindre 1,92 Mb/s. Mais pour atteindre un tel débit il faudrait être statique devant un émetteur, rarement le cas dans le monde de la mobilité.

Pour éviter des variations trop importantes, les opérateurs français ont choisi de brider le réseau à 384Kbit/s pour obtenir un débit constant. Ce bridage permettait d'obtenir une borne passante constante indispensable pour des appels en visioconférence.

La 3G a vu son apparition sur le sol français début 2005, certes le débit était imposant, mais le bémol reste sur la couverture réseau du pays. Lorsque 85 % du territoire était potentiellement couvert par l'EDGE seulement 65 % était couvert par la 3G.

Ce débit de plus en plus constant et puissant a engendré, de la part des constructeurs de téléphone portable, la création de mobile de plus en plus accès sur le multimédia. On notera par exemple l'exploit d'Apple qui a su se faire une place dans le monde du mobile grâce à ses services multimédias proposés par l'iPhone. Par exemple : son navigateur internet, son store privé ou même ses applications embarquées dans le téléphone comme la « map » de Google. Mais Apple n'est pas le seul à proposer la nouvelle génération de téléphone portable, plus communément appelé « smartphone », son principal concurrent, Android (système d'exploitation pour téléphone portable proposé par Google), offre les mêmes services, mais sur des téléphones de différents constructeurs (Samsung, HTC, LG...).

C'est dans cet univers multimédia que les grands noms d'internet (Facebook, Twitter, EBay, Google...) ont choisi d'être présent au travers d'application téléphonique propre à chaque système d'exploitation (iOs, Android, Windows Phone...)

Ces applications, dites « native », sont développées dans le langage informatique préconisé par les OS (iOS : Objective-C, Android : Java, WP7 : C#...). Dans une telle optique les applications multiplateformes (disponible sur toutes les plateformes mobiles : iOS, Android, Windows Phone...) sont plus compliquées à mettre en place, car plus coûteuse et plus dure à maintenir. C'est pour cela que certaines applications web, autrefois utilisé principalement sur ordinateur, se sont adaptées à la mobilité (tablette, smartphone). Plusieurs applications internet ont fait le choix de ne pas mettre en place des applications natives sur téléphone portable. « Comment ça marche », « 118 218 » ou encore « Plank » (www.plankdesign.com) offrent une nouvelle ergonomie à leurs applications web qui s'adaptent à toute forme d'écran y compris les plus petits comme les smartphones.

Les WebApp sont accessibles via un navigateur web disponible sur n'importe quel smartphone. Contrairement aux applications mobiles classiques qui sont disponibles sur des stores distincts par OS, les applications web ne bénéficient pas d'une telle visibilité. Le site d'Apple propose à l'adresse « www.apple.com/webapps » un référentiel de Web App. Les applications sont triées par genre tout comme « l'App Store » (plateforme de téléchargement d'application mobile d'Apple pour iOS).

Même si l'on doit utiliser un navigateur internet pour accéder aux applications web, il est possible de les installer sur le bureau du téléphone portable.

Le cabinet « Kleiner Perkins Caufield & Byers » a rendu un rapport montrant que le monde du mobile génère 15 % du trafic internet. Alors qu'en 2009, ce chiffre était évalué à 0,9 %. On voit bien que les usages changent, surtout que ce même institut prévoit une hausse pour fin 2014, avec un trafic évalué à 30 %.

Dans cette optique, nous sommes en droit de nous poser la question suivante :

- Les applications web vont-elles supplanter les applications natives ?

En effet, avec les futurs débits délivrés par le réseau 4G+, les futurs téléphones portables toujours plus puissants et toujours plus connectés, la question reste présente.

Beaucoup de sites internet ont essayé d'y répondre, mais les explications données restent très souvent succinctes, et portent à de nombreux débats souvent creux et sans réelle finalité.

Dans l'objectif de donner une explication approfondie du sujet, j'ai choisi d'analyser dans un premier temps le développement web dans le domaine de la mobilité. De ce fait, nous analyserons l'évolution des technologies web puis nous verrons les différents types d'application mobile et enfin nous traiterons de l'aspect pécuniaire des applications web mobile.

Dans un second temps, nous verrons les différentes exigences des applications web mobiles. À savoir les aspects de connexion, la réactivité ou même la compatibilité suivant le téléphone utilisé. Dans cette même partie, il sera intéressant d'analyser les limites des applications web. Que ce soit les limites de connexion, les limites en terme de ressource ou en terme de visibilité.

Puis nous terminerons cette analyse en essayant de proposer une solution à la mise en place d'application web au travers de la mobilité.

I Les applications web dans la mobilité

Les applications web sont souvent assignées comme des applications de type **client-serveur**. Le client désigne un appareil connecté, comme un ordinateur ou un téléphone portable ou même une tablette. Alors que le serveur désigne dans la plupart des cas un ordinateur dédié au logiciel serveur. Cette architecture permet d'obtenir :

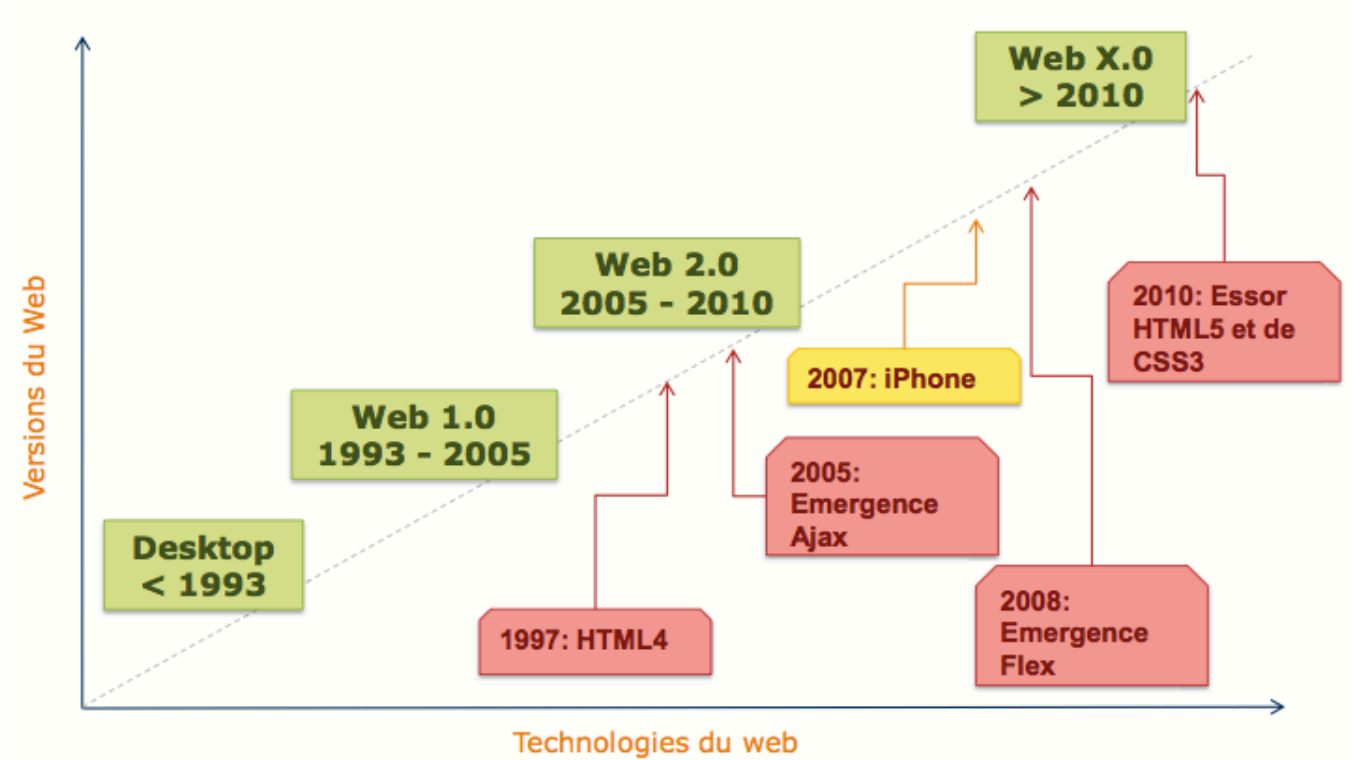
- une meilleure répartition des calculs machine,
- un stockage de données centralisé sur le serveur,
- l'identification de chaque client,
- une maintenance et un enrichissement des contenus de l'application client en dynamique,
- un espace de stockage nettement réduit pour le côté client.

Ces différents points offrent à l'utilisateur lambda la possibilité d'avoir accès à un grand nombre d'applications sur un seul terminal, car tous les stockages volumineux et les calculs gourmands sont pris en charge par le serveur (dans un grand nombre de cas).

C'est le principe de base d'internet, une application web comme "Plank" offre un contenu dynamique au client, contenu qui sera mise à jour depuis le serveur. Mais les Web App ne sont pas les seuls bénéficiaires de cette technologie, le client peut être apparenté à des applications natives d'un téléphone portable par exemple. Cette architecture regroupe un grand nombre de technologies qui ont évolué avec leur temps et leurs innovations. L'environnement client-serveur s'est bien installé dans le monde de la mobilité. Elle permet de mettre à jour certains points d'une application sans passer par un store quelconque.

1- Évolution des technologies web

i- HTML 5/CSS3



Comme le souligne ce graphique, nous sommes à l'heure actuelle dans la phase d'évolution du web 2.0. Avec l'arrivée du « Cloud computing* », des smartphones et des différentes Web App, internet commence à changer nos habitudes. Là où l'on utilisait « Microsoft Office Word », plusieurs utilisateurs choisissent maintenant « Google Docs » pour le « Cloud computing » par exemple.

Le graphique montre aussi l'émergence de Flex en 2008. Flex est un outil, mise en place par Adobe, qui permet de développer des applications/sites internet en « Flash ». Le problème de Flex est qu'il n'est pas conçu pour le mobile, mais plutôt pour les ordinateurs fixes ou portables. De plus, tous les smartphones n'ont pas de lecteur Flash intégré. Dans la même branche, « Silverlight » permet, lui aussi, de créer des applications web, mais il possède des problèmes de compatibilité avec l'OS Linux. De plus, Silverlight n'est plus maintenu depuis 2 ans.

L'HTML 5 est aujourd'hui la meilleure solution web pour constituer des applications web multiplateforme mobile.

L'HTML est constitué de multiples balises qui ont pour but de mettre en forme les pages. Par exemple, une balise <h1> permettra d'afficher un titre en grand format.

L'HTML 5 a vu le jour en 2006, mais a commencé à être démocratisé depuis environ 3ans. La mise en place de nouvelles balises comme :

- la balise « vidéo » ou « audio » pour le traitement des médias,
- la balise « canvas » qui permet d'obtenir des rendu 2D ou 3D,
- la mise en cache de contenu,
- et d'autres balises plus générales, balise propre au rendu graphique (ex : section, article, header, footer...).

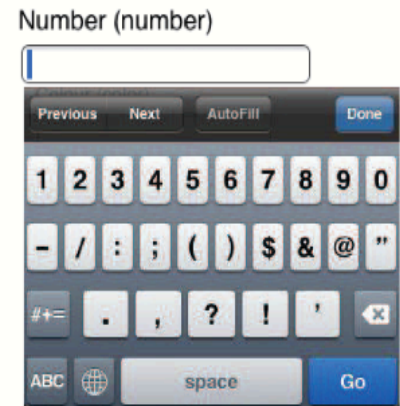
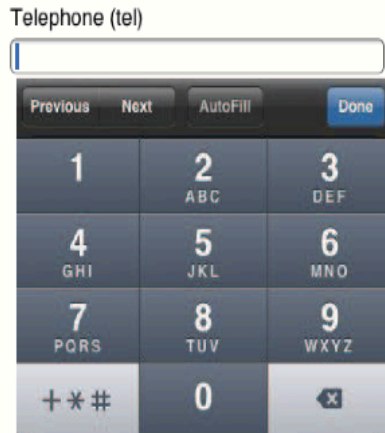
Les traitements médias sont de loin les plus importants dans la mobilité. Que ce soit pour les jeux vidéo (qui sont grandement majoritaire dans les téléchargements mobiles) ou plus généralement les traitements vidéo, audio et d'images (Instagram étant un bon exemple d'application téléphonique native utilisant des traitements média).

L'ajout de la balise vidéo permet de pallier à l'utilisation du flash, autrefois plus qu'indispensable pour la visualisation de vidéo sur internet. Les smartphones peuvent désormais lire des contenus vidéo plus facilement sur internet via cette balise.

Les « canvas » permettent, par exemple, la création de jeux vidéo. Avec des performances par fois plutôt surprenante comme le montre le jeu vidéo : SkateDuck2.

La mise en cache de contenu permettra à l'utilisateur d'avoir des contenus sauvegarder dans son téléphone. Que ce soit des pages entières ou des images ainsi que des actions utilisateurs..

Dans la saisie des formulaires web, on voit apparaître de nouveaux types de champs. Comme le type « tel », « date » ou « number » qui afficheront sur un téléphone des claviers bien distincts comme l'illustre l'image si dessous.



iPhone

D'autres fonctionnalités, dans le domaine de l'interaction entre l'utilisateur et l'application, voient le jour. Il est possible de glisser-déposer des éléments dans une page de l'application, une fonctionnalité présente dans le développement d'application mobile native.

La géolocalisation fait aussi son apparition avec l'HTML 5. Il est maintenant possible de localiser un téléphone ou ordinateur (si l'utilisateur l'accepte) via quelques lignes de code.

Ces différentes fonctionnalités permettent aux développeurs d'avoir un accès plus étendu sur le développement d'application web pour téléphone portable.

Le CSS 3 permet de gérer les aspects graphiques. Les animations, transformation 2D/3D, la gestion de la transparence, les effets de texte (comme l'ombrage) font partie des nouvelles fonctionnalités du CSS 3. Les développeurs pourront créer des transitions ou animations autrement quand JavaScript. Le CSS3 va permettre d'alléger le code JavaScript autrefois présent en abondance sur les applications web.

Contrairement à l'ancienne version, le CSS 3 offre la possibilité de gérer certains événements, autrefois géré exclusivement par le JavaScript. Il est maintenant possible de capté lorsqu'un utilisateur sélectionne un champ d'un formulaire ou s'il clique sur un lien.

Les applications web sont aussi appelées « **rich internet application** » (RIA), ce terme désigne les applications web offrant des services similaires aux logiciels installés sur téléphone ou ordinateur. Le principe d'une RIA est basé sur l'aspect graphique et la vitesse d'exécution de l'application. C'est là que l'HTML 5 et la CSS 3 offrent une vraie réponse à ce domaine d'application. Deezer ou Gmail sont des très bons exemples de RIA.

Cependant, l'HTML 5 et la CSS 3 ne permettent pas d'effectuer des traitements évolués, comme le rafraichissement d'une partie d'une page ou l'insertion d'une « scroll view¹ » ou plus généralement le contrôle de surface tactile. C'est pour cela que plusieurs framework JavaScript, spécialisés dans le domaine de la mobilité, ont vu le jour.

ii- Les nouvelles librairies

Internet a évolué avec ses technologies. L'interaction entre l'HTML et le JavaScript est un élément omniprésent dans les applications web. Dans un souci d'aide au développement, des librairies JavaScript voient le jour. On notera l'arrivée de jQuery en janvier 2006, Prototype en février 2005 ou encore accDC en novembre 2010. Ces différentes librairies sont utilisées pour simplifier l'utilisation du JavaScript. Elles permettront de mieux gérer les appels AJAX* ou des animations complexes.

Avec l'arrivée du développement web sur téléphone portable, plusieurs librairies se sont spécialisées dans ce domaine. Répondant à un besoin plus précis,

¹ Faire défiler le contenu d'une vue

contrairement à jQuery et Prototype qui sont bien plus gourmands en ressource (car plus volumineux). Ils répondent à une problématique de compatibilités sur les vieilles versions d'« Internet Explorer » ce qui n'est pas le réel problème du développement web dans la mobilité.

On notera par exemple :

- Zepto.js
- DynamicX
- Sencha
- jQuery Mobile
- jQTouch

Zepto.js

« Zepto » propose aux développeurs une syntaxe jQuery, plus intuitive et simple d'utilisation. Ce framework* a pour but d'être léger avec seulement 2Kb, ce qui rendra le téléchargement des fichiers et leurs mises en cache plus optimiser pour du développement mobile. Bien entendu zepto permettra l'accès aux fonctionnalités d'animation et de gestuel propre aux écrans tactiles. Cependant, cette librairie étant très légère elle ne permet pas de faire des traitements avancés en JavaScript.

DynamicX

« DynamicX » offre un accès à de multiples fonctionnalités mobile, comme le glisser-déposer (drag and drop). DynamicX est très accès sur l'aspect graphique des applications. Les applications qui en résultent sont très similaires à des applications mobiles. Mais, son principal défaut reste sa taille (plus de 80 Mb pour la version complète gratuite). Ce problème impacte directement les appareils mobiles d'ancienne génération (2012 et moins), comme l'iPhone 4.

Il rend les animations moins agréables, car souvent saccadées. De plus, ce framework est payant si l'on souhaite acquérir une version professionnelle permettant un accès au support et la commercialisation des applications.

Sencha

« Sencha » utilise intensivement les composants HTML 5 surtout dans le domaine des formulaires. Cela permet à cette librairie d'être plus fluide sur les terminaux mobiles. Sencha propose des fonctionnalités présentes sur les écrans tactiles (glisser déposer, diaporama tactile). De plus, les développeurs pourront avoir un accès à de nombreuses icônes et Sencha reste libre d'utilisation. Mais on retrouve un point négatif présent dans plusieurs librairies mobiles, la personnalisation des effets visuels est difficile à mettre en place. Les applications web utilisant donc cette librairie risquent de fortement se ressembler.

jQuery Mobile

« jQuery mobile » est la librairie la plus connue dans le développement web mobile. Elle a de nombreux atouts comme sa documentation exhaustive et régulièrement maintenue. La facilité de prise en main ou encore la possibilité de personnaliser les composants facilement, reste aussi un atout important. « jQuery mobile » propose bien entendu l'utilisation des fonctionnalités tactiles des téléphones portables. Le principal bémol de « jQuery mobile » est son volume trop important pour du développement mobile.

jQuery Touch

« jQuery Touch » est un plugin de jQuery, il devra donc utiliser la librairie jQuery pour faire fonctionner cet outil. C'est un point assez gênant pour du développement mobile, la plupart des fonctionnalités de jQuery ne seront pas utiles, car elles sont mises en place pour les terminaux fixes. jQuery Touch propose tout de même de nombreuses fonctionnalités mobiles. Dans la gestion des formulaires ou la gestion d'animation.

Il existe bien d'autres librairies comme Wink ToolKit, iUI, Webkit ou WebAPP.net. Toutes ces librairies ont toutes leurs avantages et leur inconvénient, mais toutes ont en point commun la gestion des aspects tactiles et ergonomiques d'une application mobile native. Libre au développeur de choisir par la suite celle qui

convient au mieux pour son projet. Que ce soit pour une application qui nécessite des configurations élevées, mais qui aura donc certaines latences ou des applications moins gourmandes, mais moins poussées fonctionnellement parlant.

La norme W3C a noté les différentes librairies par rapport aux règles du bon développement Web. Sancha obtient la meilleure note avec un total de 84 % sur le « W3C mobile checker ». Au contraire JQTouch est la librairie la plus mal notée due faite de sa lourdeur et de son abandon en terme de maintenance aux dépens de « JQuery mobile » avec un score de 55 %.

iii- Les technologies serveur

Les technologies serveur servent principalement à traiter les données dynamiques d'une application. Le monde d'internet utilise grandement le développement serveur. En effet, les rares applications web ne l'utilisant pas sont souvent des sites vitrines ou certains jeux vidéo ne nécessitant pas de sauvegarde complexe.

Le développement serveur existe depuis bien longtemps. Par exemple : le PHP, qui est le langage serveur le plus populaire dans le monde du développement web, est apparu en 1994. Mais d'autres langages plus anciens permettaient de construire des logiciels serveur comme le « Basic* » ou le « C++* ».

Les technologies serveur ont évolué avec le temps, aujourd'hui les plus importantes sont :

- .Net* (2002)
- Java (1995)
- PHP (1994)
- Ruby On Rails (2004)

PHP

Le PHP est apparu en 1994, grâce à Rasmus Lerdof. C'est en 1995 que la première version voit le jour. Mais la vraie naissance du PHP est en 1997, avec la sortie de la troisième version. Un projet d'origine personnel devient un projet de groupe. Avec la collaboration d'Andi Gutmans et Zeev Suraski qui prirent contact avec Rasmus Lerdof via internet pour discuter des structures du PHP et de la réécriture de la base du langage.

Le PHP a pour réputation d'être simple à apprendre et à mettre en place. C'est pour cela que les jeunes développeurs apprennent très souvent à utiliser ce langage pour connaître les bases du développement. De plus, le PHP est écrit en langage « C ». C'est pour cela que l'on retrouve très souvent des similitudes entre les deux langages, ce qui est utile lorsque l'on sait que le langage « C » est la base de l'apprentissage du développement informatique. Mais depuis les toutes dernières versions du PHP, son utilisation est plus étendue dans le monde de l'entreprise. Car dans un premier temps un grand nombre de développeurs ont appris les bases de développement de ce langage. Et dans un deuxième temps, la création de Framework permettant la facilité du développement voit le jour comme Zend ou Symfony.

Le PHP est le langage le plus populaire sur internet, la documentation y est très exhaustive (pour exemple, la documentation officielle sur : php.net, disponible sous 30 langues).

Le vrai problème du PHP est sa permissivité trop importante, ce qui peut être un réel problème si un projet doit durer dans le temps et donc être maintenu. Les développeurs risquent de perdre plus de temps dans la compréhension du projet sur le long terme.

Le PHP est une solution très satisfaisante pour les applications web nécessitant de faibles traitements serveur, mais n'est pas forcément le meilleur choix sur le développement de plus grosse structure.

Ruby on Rails

Le Ruby on Rails a été créé en 2004 par David Heinemeier Hansson. Cet outil combine le langage de programmation Ruby et le framework Ruby on Rails aussi appelé RoR. Il existe actuellement 3 versions majeures du RoR.

Le Ruby on Rails est, comme le PHP, un langage interprété. Mais, à la différence du PHP, il est orienté objet depuis sa création.

Le Ruby a pour avantage d'être simple à relire et à apprendre. Mais des difficultés, dans les premières versions, liées à l'installation de l'outil ont rendu le langage moins populaire que d'autres. Même si l'installation de base est assez simple il s'en suit des difficultés lorsque l'on souhaite installer des « gems* » plus spécifiques à son projet.

Le Ruby on Rails a du mal à s'installer en France, du fait de sa communauté anglophone. Il est très rare de trouver des tutoriels d'apprentissage en français, et toutes les documentations techniques sont disponibles en anglais.

Mais, au-delà des barrières de langue, le RoR est un langage facilement maintenable, car il est plus simple à lire et à comprendre qu'un site développé en PHP. Par expérience, il est plus simple de comprendre le fonctionnement d'un ancien script Ruby qu'un ancien script PHP.

.NET

Le .Net a été conçu par Microsoft. C'est un framework utilisant le langage C#. Il repose sur plusieurs technologies, comme son IDE* Visual Studio ou la multitude

de bibliothèques de haut niveau ainsi qu'un ensemble de protocoles de communication*.

Le .NET est une technologie sous licence. Le prix varie selon le revendeur, mais reste d'environ 700 €, par poste de travail, pour ce qui est de la licence de développement (Licence Visual Studio). On pourra cependant noter que la documentation est très exhaustive et qu'il existe un support d'aide pour les aspects d'installation de la solution ou l'utilisation des composants techniques du framework .NET.

Le .NET est la deuxième solution serveur choisie dans le monde du WEB d'après les chiffres de W3Techs. Cette réussite s'explique du fait que les SSII choisissent ce langage, car cette solution fait partie d'un ensemble de solutions de développement proposé par Microsoft. Et en général, les SSII ont des partenariats avec la solution Microsoft et des ressources compétentes dans ce domaine.

JAVA

Le Java est un langage de programmation développé par Sun Microsystems racheté par Oracle en 2009. Il est entièrement orienté objet. Tout comme le Ruby et le C#. Il utilise un framework pour mettre en place un développement serveur qui est le J2EE (Java 2 Enterprise Edition).

Le Java est un langage open source*, qui a la chance de posséder une grande communauté à travers internet.

Le J2EE est très riche et permet la création d'architecture serveur sophistiquée. Mais, du fait de sa couverture étendue, il s'en suit une installation de diverse librairie et d'infrastructure réseau. Ses installations sont par fois très complexes à mettre en place et peuvent engendrer des maintenances plus difficiles avec des temps de développement rallongé.

Le Java reste tout de même très prisé des sociétés de services, car il est gratuit et permet la création de projet compliqué. De plus, ce langage est très utilisé par les

jeunes développeurs, car il est très généraliste. La main d'œuvre est donc très présente sur le marché du travail.

Ces différents langages sont assez anciens et ils ont chacun leurs avantages et inconvénients. Mais les fonctionnalités sont généralement les mêmes. Les vraies différences sont leurs installations sur les serveurs ou leurs facilités d'apprentissages et de maintenances.

Dernièrement, le JavaScript qui est à l'origine un langage utiliser du côté client a vu la création d'un framework pour l'utiliser aussi du côté serveur. Ce framework s'appelle **Node.js**.

Node.js a été créé en 2009 et déjà en 2011 LinkedIn a choisi de migrer ses serveurs Ruby on Rails vers Node.js. En effet, son application mobile rencontrée des problèmes de performance. Et d'après LinkedIn, le Node.js serait 20 fois plus rapide que le Ruby on Rails.

2- Différents types d'application mobile

Dans le monde du développement mobile, le choix des technologies à adopter pour créer son projet est limité. On a le choix entre deux principes : les applications natives ou les applications web. Le choix est souvent délicat et sujet à beaucoup d'analyse sur internet.

Les grands groupes optent en général pour le développement sur les deux plateformes. Comme Facebook :



Application iPhone native



Application web sur Safari iPhone

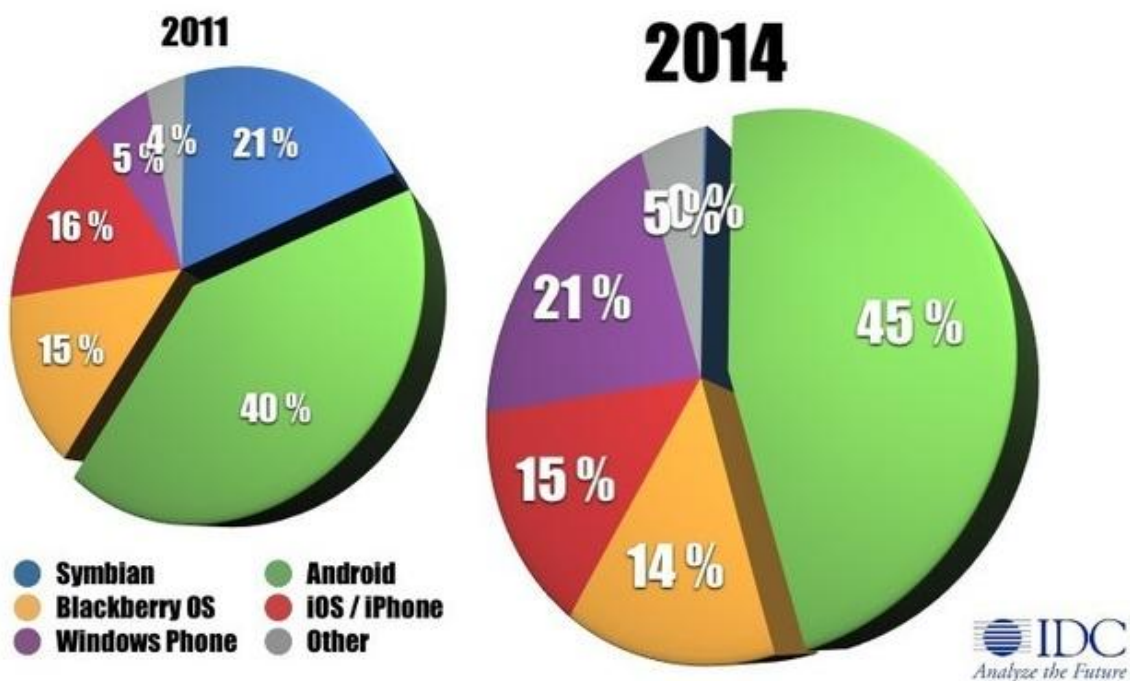
On remarque que les deux applications sont fortement ressemblantes, mais quelles sont les réelles différences ?

i- Application native

Dans le monde du développement mobile, on distingue six plateformes principales. Elles sont distinguées par six systèmes d'exploitation (OS) majeurs :

- iOS,
- Android,
- Windows Phone,
- BlackBerry,
- Symbian OS.

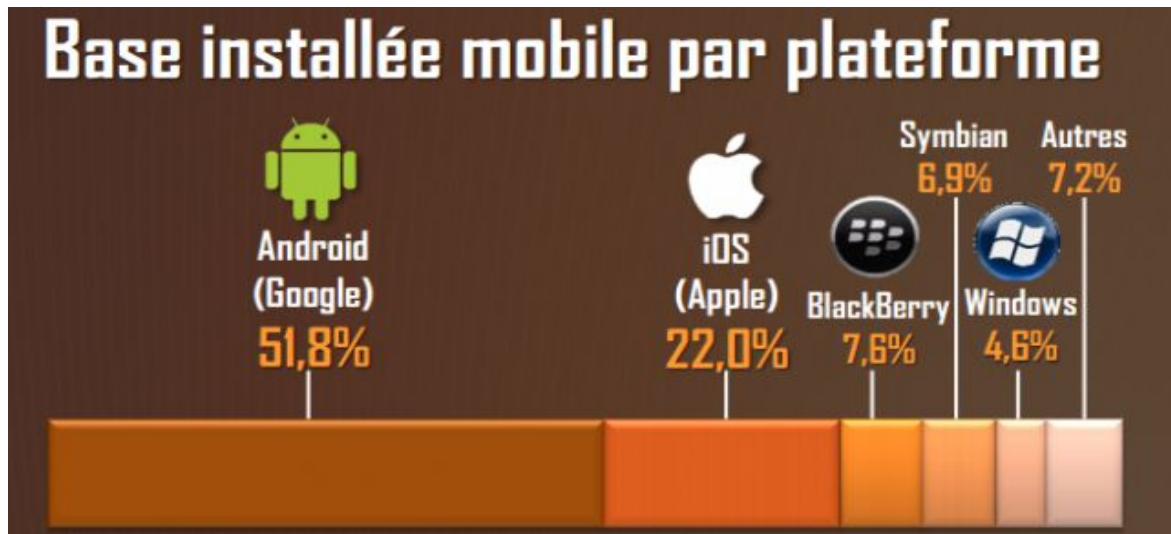
Mais, sur les six plateformes, seulement trois sont représentatives de l'avenir des OS mobiles à savoir Android, iOS et Windows Phone.



Source : <http://www.docdoku.com>

Ces graphiques de l'IDC (International Data Corporation) montrent les parts du marché mondial. On remarque donc que les trois OS regrouperont 81 % des ventes de smartphone en 2014.

Mais les chiffres sont différents sur le sol français :



Source : www.monwindowsphone.com

Ce graphique montre les résultats d'une enquête menée par l'association du marketing mobile français (MMAF) au premier trimestre 2013. On observe que les trois OS cités au-dessus regroupent 78,4 % de part de marché.

iOs

iOs est le système d'exploitation disponible sur les téléphones (iPhone), tablette (iPad et iPad mini) et lecteur MP3 (iPod touch) de la société américaine Apple.

Le développement d'application native sur cette plateforme nécessite l'utilisation du framework Cocoa* et le langage de programmation Objective-c.

L'Objective-c est une surcouche du C, créé par Brad Cox et Tom Love. Ce langage implémente les notions de programmation orientée objet.

Le développement d'application native iOs est possible avec l'utilisation de l'IDE xCode. Ce logiciel est disponible gratuitement sur le site d'Apple, mais il n'est exécutable que sur un ordinateur Mac OSX.

L'IDE ne suffit pas pour mettre en place une application iOs. Il faudra en plus installer le SDK* iOs (fourni par Apple) pour avoir accès au simulateur d'iPhone et d'iPad.

Pour ce qui concerne la mise en ligne des applications native iOS, il existe un store public, mais aussi un store privé. La licence coûte 99 \$ par an pour un particulier et 299 \$ par an pour une société. L'offre entreprise permettra d'utiliser le store privé.

Android

Android est une solution proposée par le géant du web : Google. Ce système d'exploitation est gratuit pour les fabricants qui souhaitent l'installer sur leurs appareils mobiles.

Android repose sur la fusion entre la JVM* (Java Virtual Machine) Dalvik et un noyau Linux* modifié. Le développement d'application Android est effectué en Java, mais s'exécute dans la JVM* Dalvik. De ce fait, toutes les bibliothèques disponibles en Java ne sont pas accessibles, car elles doivent être compilables sur la JVM Dalvik.

À la différence de l'iOS, il est possible d'écrire et de compiler des applications Android dans différents IDE. Mais seulement deux sont recommandés par les équipes d'Android, à savoir IntelliJ IDEA et le plugin ADT pour Eclipse.

Le développement Android est gratuit, et la mise en ligne via le « Play Store* » est elle aussi gratuite. C'est un avantage de poids lorsque l'on connaît le prix des licences du principal concurrent à savoir iOS d'Apple.

Windows Phone

Windows Phone est le dernier OS pour téléphone portable à voir le jour, il est mis en place par l'entreprise Microsoft. Son lancement a été effectué en novembre 2010 succédant à Windows Mobile qui était plus axé sur le monde de l'entreprise.

Microsoft a choisi d'entrer dans le monde du mobile grand public en utilisant une nouvelle IHM* : Modern UI (anciennement Metro). Cette IHM est la nouvelle marque de fabrique de Microsoft, car elle est reprise dans l'interface de Windows 8.

Il est possible de développer des applications Windows Phone via deux langages le C# et le VB.NET. Ces deux langages peuvent être utilisés simultanément

dans une application. Il faut savoir qu'il existe aussi deux framework pour le développement d'application Windows Phone :

- Le XNA, qui permettra la création de jeux vidéo en 2D et 3D, mais aussi tout ce qui concerne le traitement des médias (audio, vidéo).
- Et Silverlight qui sera plus orienté vers un développement d'application de gestion.

Pour l'environnement de développement, Microsoft propose une solution gratuite Visual Studio 2012 Express qui est une version dédiée à Windows Phone contrairement à Visual Studio 2012.

Pour pouvoir sortir son application sur le store Windows Phone il faudra acquérir un compte développeur évaluer à 99 \$.

Pour ce qui concerne les autres OS :

- Symbian OS, créé par Nokia, est victime du fort succès d'Apple et Google. Nokia cède son OS à Accenture (entreprise internationale de conseil) et signe un partenariat pour l'installation de Windows Phone sur leurs futurs smartphones.
- BlackBerry est l'OS de l'entreprise RIM. Ce système d'exploitation équipe tous les terminaux mobiles de cette société. BlackBerry connaît depuis quelque temps une perte de vitesse qui est due à la fois à la crise et à leur retard en terme d'innovation technologique. Le PDG du groupe, Thorsten Heins, explique que les chiffres de l'entreprise ne vont pas améliorer et explique pourquoi : « *au cours des prochains trimestres (le groupe) va augmenter ses investissements pour soutenir la sortie de nouveaux produits et services.* ». BlackBerry fait donc le pari de l'innovation technologique dans les futures années ?

Tous ces OS ont des points communs, mais ont tous la même différence : la méthode de création d'application mobile. C'est là que la « cross-compilation* » rentre en jeux.

Il existe plusieurs outils permettant la cross-compilation, WinDev Mobile ou même la technologie MONO. Cette méthode de compilation* permet de créer une seule application qui sera compilée dans tous les langages spécifiés au préalable. Le gros problème de cette solution reste la personnalisation selon l'OS ou l'accès à différent composant qui ne seront pas toujours présents sur tous les terminaux.

Les applications qui utilisent cette méthode ne sont pas toujours très poussées et elles relèvent plus de l'application dite « vitrine » montrant une activité ou une campagne publicitaire.

ii- Application web ou “Web Apps”

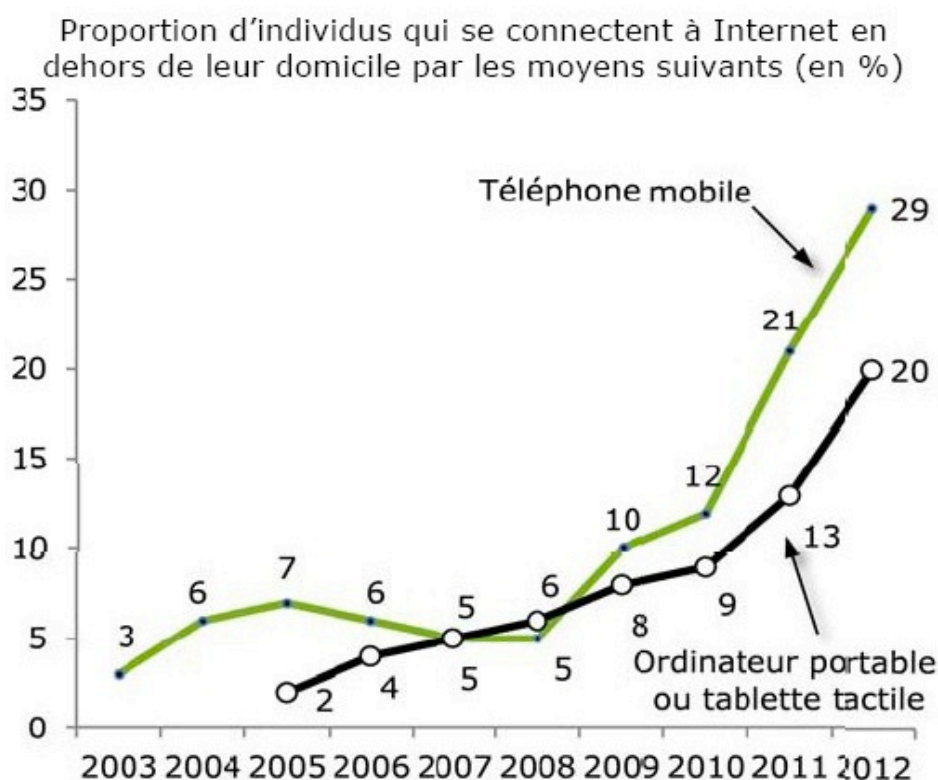
L'installation d'une application web, sur téléphone portable, est très simple. Il suffit d'utiliser le navigateur internet de son téléphone et de lancer l'application via son URL. Le seul prérequis est donc d'avoir un navigateur internet installé sur son téléphone. Cette prérogative devient obsolète lorsqu'on sait que tous les OS installés sur smartphone possèdent un navigateur internet préinstallé.

Le terme « Web Apps » désigne les sites internet optimisés pour téléphone portable. Ces « Web Apps » ont pour objectif d'être identiques visuellement à des applications dites natives (ce qui peut être contradictoire lorsque l'on sait que chaque OS mobile a sa propre identité visuelle). Même si le but premier d'une application web mobile reste sa compatibilité sur le multiplateforme il reste difficile d'être proche de chaque identité visuelle mobile.

Cette compatibilité graphique est tout de même rendue plus simple via les nouvelles librairies JavaScript citées plus haut, elle reste proche de l'aspect de base

d'une application mobile. L'HTML 5 et le CSS 3 permettent, eux aussi, d'obtenir des rendus appréciables pour les applications web. Avec des animations graphiques lançables en CSS 3 ou les différentes nouvelles fonctionnalités du HTML 5 comme le stockage de donnée en local ou le support du « multitouch* », les sites web sont de plus en plus adaptables pour le monde de la mobilité.

Les « Web Apps » sont plus qu'importants dans l'utilisation de l'internet mobile. Une enquête menée par CREDOC note que 29 % des connexions mobiles sont effectuées via un téléphone portable (voir graphique ci dessous).



Cette proportion qui est en constante évolution force les pionniers d'internet à proposer des applications internet adaptées aux téléphones portables.

Nous pouvons trouver plusieurs types d'application web, les plus courantes restent les applications de gestion comme Facebook, LinkedIn ou le Monde. On retrouvera aussi des jeux vidéo, comme SkateDuck2 ou Biolab, qui démontrent les

performances graphiques des Web Apps dans le jeu vidéo. Il faut tout de même noter que les jeux vidéo web sont généralement en 2D, les performances matérielles n'étant pas suffisamment poussées pour gérer du rendu 3D. Des applications web, dans le domaine du service, sont aussi disponibles comme la carte interactive de Google : la « Google Map » qui permet à l'utilisateur d'être localisé sur une carte et lui offre la possibilité de parcourir un itinéraire défini entre sa localisation et un point quelconque.

3- L'enjeu économique des applications web dans la mobilité

Le « Journal du Net » a recensé, en 2011, plus de 2,2 milliards d'internautes à travers le monde, et près de 5,9 milliards d'abonnés à la téléphonie mobile. Ces chiffres ne cessent d'accroître chaque année. En France, on note 50 millions d'internautes ainsi que 66 millions d'abonnés mobiles. L'enjeu économique sur une telle population est un tournant prioritaire dans le développement numérique actuel.

i- L'économie d'internet

Sur les 5 dernières années, internet a généré 40.000 milliards de dollars et représente 21 % de la croissance des pays développés. Avec plus de 340 millions de sites internet, le style de vie de la population mondiale s'est vu fortement modifier. Internet a permis de créer une nouvelle valeur économique et est devenu un secteur économique à part entière. Le « McKinsey Global Institute » s'est penché sur l'économie* des pays les plus touchés par la vague Internet.

Son étude démontre « *qu'Internet représente 3,4 % du PIB des économies des pays développés* ». Des chiffres astronomiques lorsque l'on s'aperçoit que la contribution d'internet au PIB mondial représente un montant bien supérieur à celui de l'Espagne.

Internet est le cœur de métier de nombreuse société. En 2011, les 250 firmes les plus influentes au sein des TIC* (technologies de l'information et de la communication) pesaient près de 4.600 milliards de dollars.

Au-delà des revenus générés par internet, des coûts de mise en place d'une solution web sont présents.

Une entreprise se penchera sur le choix technologique d'une solution web. À savoir le choix d'un langage serveur plutôt qu'un autre.

En effet, dans la partie « *Les technologies serveur* » différents langages et framework associé ont été mis en avant. Une solution pourra être plus rapide à mettre en place, mais l'utilisation de serveur plus puissant ouvrira des coûts plus importants. D'autres solutions vont demander l'achat de licence de développement parfois couteuse.

Le choix d'une technologie doit se faire selon l'importance et la pérennité du projet qui doit être mis en place. Dans la plupart des cas, le choix se porte sur les ressources disponibles et leurs connaissances associées. En cas de pénurie sur le marché de l'emploi le coût de la main-d'œuvre risque de s'alourdir.

Une étude sur le choix des technologies à mettre en place est obligatoire aujourd'hui pour prendre en compte les coûts de mise en place ainsi que les coûts de maintenance d'un projet.

Il existe une multitude de modèles économiques sur Internet qui sont propres à chaque application. Par exemple, Groupon offre un modèle unique en son genre en proposant des réductions sur des produits ou services locaux, par la suite l'entreprise perçoit 50 % de la vente et reverse le reste à la société qui offre le service. Mais d'autres modèles économiques sont plus répandus comme le système d'abonnement, la rémunération par la publicité (revenu premier de Google ou Facebook) ou encore l'achat de contenu directement dans l'application (majoritairement utilisé dans le monde du jeu vidéo).

Chaque modèle économique est propre à l'application qui l'utilise. C'est une réelle problématique lors de la mise en place d'un produit. Une application innovante, mais souffrant d'un modèle économique bancal risque d'être mise de côté. Car manquant de fonds d'investissement pour la maintenance de celle-ci.

ii- L'économie des applications mobiles natives

Dans le monde du développement natif, il existe 3 systèmes de monétisation très répandus :

- La vente de l'application sur un store,
- La mise en place de contenu payant au sein de l'application,
- L'affichage de publicité,
- Ou la proposition d'abonnement.

D'autres systèmes sont aussi rendus possibles comme la promotion d'autre application. Comme l'application « AppGratis » qui proposait chaque jour une application à l'origine payante et qui devenait gratuite pendant un jour. Pour apparaître dans l'application il fallait déboursier environ 10.000 €, mais en contrepartie votre application gagnée énormément en visibilité et généralement passez première en terme de nombre de téléchargement.

Il n'y a pas de méthode meilleure que d'autres. La seule mise en garde est sur l'application payante de base. En effet, l'utilisateur qui souhaite télécharger une application sur un store préférera une application gratuite à une application payante. D'autant que sur certaines plates-formes il n'est pas possible de tester le produit avant de l'acheter. Le client devra donc se reporter à une brève description de l'application ainsi que quelques copies d'écran pour faire son choix. Il est donc conseillé de proposer à la fois une application avec du contenu réduit, mais gratuit (une version d'essai) et une version payante, mais complète.

Les applications gratuites utilisent en général un système de publicité pour la rémunération. D'autres feront le choix de vente d'abonnement comme « coyote » qui propose le repérage communautaire des radars routiers ou le Monde qui a construit tout un « business model » autour de la vente d'abonnement.

Selon une étude menée par le cabinet d'étude Canalys, les principales plateformes de vente d'application mobile ont généré un revenu de plus de 2 milliards de dollars. Mais la répartition n'est pas équivalente entre chaque store. Lorsque le Google Play totalisait 51 % des téléchargements d'application, l'Apple Store, quant à lui, représenté 74 % du chiffre d'affaire sur les ventes d'application.

L'économie des applications mobiles repose sur la vente des appareils qui la composent, à savoir les smartphones. Or, les smartphones connaissent une croissance de vente exceptionnelle avec la vente de 237 millions de téléphone vendu à travers le monde durant le deuxième trimestre 2013. Il n'est pas simple d'innover en terme de modèle économique sur le lancement d'une application mobile native, car sur tous les paiements, le propriétaire du store (ou plus généralement des OS mobiles) reçoit une taxe. Les propriétaires des différents stores auraient donc plus intérêt à verrouiller les modèles économiques pour continuer à percevoir cette taxe. C'est le cas de AppGratis où Apple a choisi de la supprimer de son store, car l'application ne respectait pas des points de la nouvelle charte du store, qui excluait (par exemple) les applications promouvant d'autres applications. En quelque sorte, la publicité d'Apple pour les applications du store ne passait plus par leur soin, donc Apple ne percevait pas de taxe sur cette méthode publicitaire.

iii- L'économie du web dans la mobilité

Internet connaît bien plus de modèles économiques que le monde des applications natives mobile. D'après Fred Wilson, l'économie d'internet va se greffer à l'économie du mobile dans les futures années. En quelque sorte, les modèles économiques les plus performants dans Internet seront disponibles sur mobile. La seule différence restera sur l'approche des grands groupes, comme Apple ou Microsoft, qui ne souhaitent pas forcément ne plus avoir la main mise sur les revenus des applications installer sur leur OS mobile.

Aujourd'hui, une Web App souffre de visibilité sur le marché du mobile. Il est difficile de trouver un annuaire complet d'application web. Ce qui n'est pas le cas pour les applications natives. Une personne voulant partager son travail sur une application web ne pourra pas la proposer payante au téléchargement, car étant un contenu d'Internet son ouverture est faite via une URL publique.

C'est donc aujourd'hui la seule limite en terme de modèle économique dont souffre les Web Apps.

Mais de toute manière le modèle économique basé sur une application payante n'est pas forcément le meilleur choix. En effet, un client ayant téléchargé l'application ne formulera plus de source de revenus (sauf si l'application contient tout de même des pubs en version payante ou propose des contenus payants). Alors qu'un modèle basé sur la vente de contenu ou la mise en place de publicité rapportera plus sur le long terme (selon l'attraction de votre application).

C'est pour cela que l'avenir économique des applications web est très ouvert. Le seul problème aujourd'hui, réside sur la visibilité des Web Apps, car si une application n'est pas visible elle ne sera pas utilisée et si elle n'est pas utilisée elle ne génère pas de revenu.

II Exigence des applications web dans la mobilité

L'exigence utilisateur sur les différents types applications est un point à part entière dans le développement d'une solution informatique. L'utilisateur va se baser sur la vitesse de communication entre lui et la machine. Il risque d'avoir des faiblesses en terme de connaissance technique et donc de ne pas comprendre certains fonctionnements d'une application. Dans le domaine de la mobilité, les problèmes de connexion ou d'IHM sont par fois rédhibitoire pour un client. C'est ce qui permettra à l'utilisateur de choisir entre 2 applications qui proposent le même contenu, mais qui visuellement et fonctionnellement sont totalement différentes.

1- Les applications connectées/déconnectées

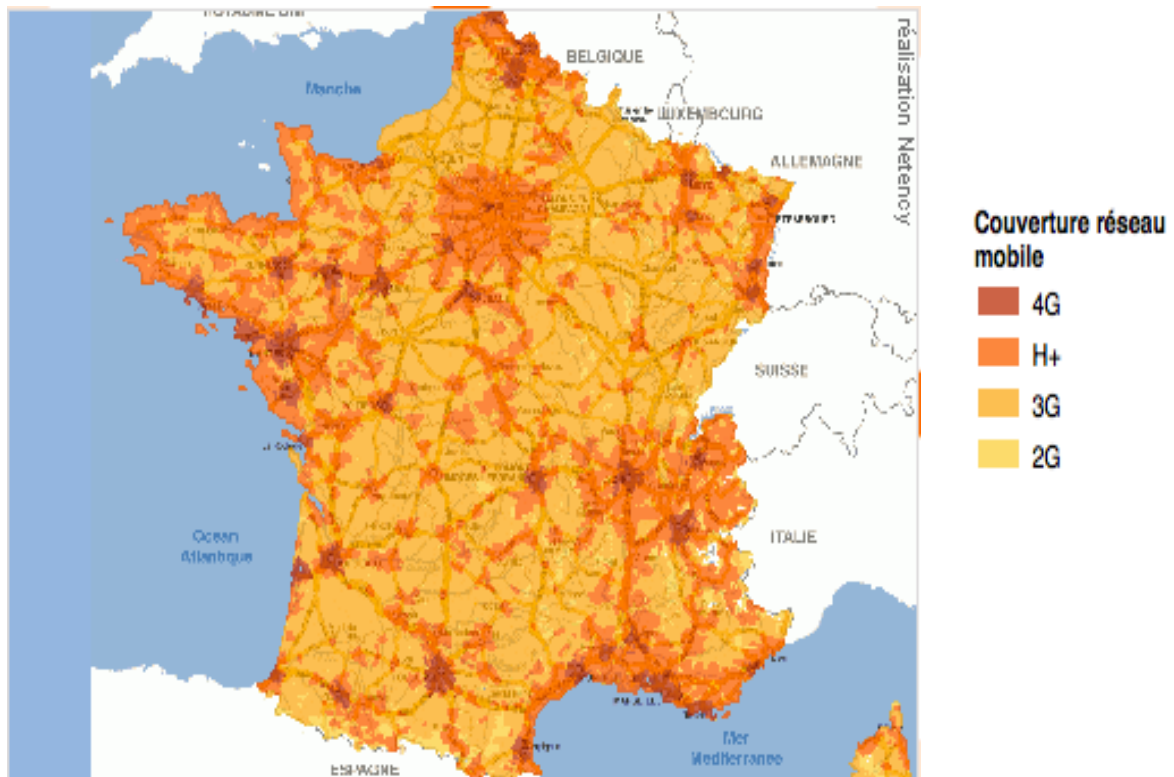
Les applications web mobiles et même les applications natives, utilisant des services internet, ont besoin d'une connexion au réseau internet pour pouvoir fonctionner. Cet accès aux réseaux est proposé par 4 opérateurs en France qui sont :

- Orange,
- SFR,
- Bouygues Telecom,
- Free.

Chacun de ces opérateurs à sa propre couverture réseau du territoire français qui est plus ou moins importante.

i- Couverture des réseaux téléphoniques mobile

Orange



Cette carte de France représente la couverture réseau du pays par l'opérateur Orange. Orange qui était anciennement appelée France Télécom. Elle est tirée du site officiel d'Orange. On peut voir que le réseau 4G* commence à prendre de l'envergure dans les grandes villes, mais que le centre de la France souffre toujours d'un réseau d'ancienne génération.

SFR

SFR obtient la meilleure couverture réseau. Lorsqu'Orange propose une couverture de 98 % de la population française avec le réseau 3G, SFR couvre, quant

à lui, 98,6 % de la population. De même avec le GSM/GPRS où SFR a une capacité de couverture de 99,6 % soit 0,6 % de plus qu'Orange.

Mais lorsque l'on regarde la couverture du réseau de demain, à savoir la 4G, SFR n'est plus en première tête. En effet, Orange couvre un plus grand espace avec environ 15 villes desservies contre 5 villes pour SFR. On peut donc en conclure que SFR a fait le choix d'améliorer ses couvertures réseau 3G plutôt que de viser vers une meilleure couverture 4G.

Bouygues Telecom

Bouygues Télécom est plus réduite en terme de couverture réseau. Avec 95 % de la population couverte pour le réseau 3G et 99 % pour le réseau GSM/GPRS. Même problème pour le réseau 4G qui pour le moment regroupe 4 villes. Mais sur le site officiel de Bouygues Télécom, l'entreprise annonce un déploiement pour le 1er octobre sur plus de 100 villes ainsi que 40 % de la population française de couverte.

Free

Free est le dernier opérateur français à faire son apparition. Il est considéré comme le réseau à bas prix. Lors de son insertion, il a eu pour effet de baisser les prix des abonnements téléphoniques avec des factures par moment diminuées de 50 % sur chaque opérateur. Mais en terme de qualité de réseau, il est situé en dernière position. On peut tout de même noter que l'offre Free Mobile n'a vu son apparition que le 10 janvier 2012. Free est pour le moment dans une meilleure progression de couverture réseau que l'était Orange ou SFR 1 an après leur lancement.

Comme on peut le voir sur cette carte, le réseau de Free occupe une grande partie du territoire. Mais Free ne propose que deux types de réseaux à l'heure actuelle :

- la 3G,
- et le GSM/GPRS

Représenté par du rouge foncé pour le premier et un rouge plus clair pour le second.



De plus, les réseaux appartenant à Free ne représentent que 37 % de la population. Ce chiffre représente à la fois le 3G et le GSM/GPRS. Mais Free peut proposer l'affichage d'une telle carte, car Free repose sur le réseau d'Orange pour les parcelles qu'il ne desservit pas.

Ces multiples opérateurs proposent donc différentes couvertures réseau. Pour choisir la meilleure couverture réseau, il ne suffit pas de regarder les chiffres, mais de vérifier que la couverture du réseau est disponible à l'endroit où vous risquez d'utiliser le plus votre téléphone.

ii- Principe des applications déconnectées

Une application est dite connectée lorsque son réseau mobile (ou réseau WiFi) lui offre la possibilité d'accéder à internet (ou juste un réseau informatique). Dans le cas où votre application utilise internet, comme les Web Apps, si votre téléphone est déconnecté d'un réseau internet, il serait judicieux de proposer à votre utilisateur un contenu, même s'il doit être réduit. En effet, dans le cas d'une application web qui ne gère pas les pertes de connexion l'affichage d'un écran blanc sera visible par l'utilisateur. En général l'écran blanc sur téléphone portable est rédhibitoire et le client risque de favoriser une autre application.

Dans le cas d'une application native, ce sera au développeur de gérer les pertes de connexion. Le développeur pourra savoir si les retours d'internet sont perçus par l'application (ou pas du tout) et pourra de ce fait notifier à l'utilisateur une perte du réseau. Par la suite, il sera en mesure de proposer un contenu statique propre à l'application.

Pour les Web Apps il est plus difficile de savoir si un téléphone est connecté à internet ou non. Mais une solution existe, elle réside dans l'anticipation. Le développeur devra anticiper une perte de connexion et utiliser la nouvelle fonctionnalité du HTML 5 : **l'enregistrement de donnée en cache**.

L'exemple type d'une application qui fonctionne en mode déconnecter est Gmail proposé par Google. Lorsque vous utilisez Gmail sur votre téléphone portable, même si vous perdez la connexion internet, vous ne serez jamais déconnecté de l'application. Car Gmail gère le mode déconnecté en HTML 5, vous pourrez toujours envoyer vos mails. Les mails seront stockés sur le téléphone et seront envoyés lorsque vous retrouverez l'accès au réseau.

Mais il existe tout de même un bémol sur l'utilisation de cette nouvelle fonctionnalité du HTML 5. La sauvegarde de fichier et de contenu sur le téléphone est bloquée à 5 Mo. En 2012, la taille moyenne d'une application native sur téléphone portable était de 23 Mo et elle ne cesse d'augmenter avec le temps.

Cette limite force le développeur à proposer des contenus plus sélectionnés. Il sera difficile de pouvoir sauvegarder une application complexe dans le cache de l'appareil. Les images représentent la partie la plus délicate à sauvegarder, car elles constituent le contenu le plus volumineux sur une page internet. Il faudra donc bien choisir l'aspect visuel de votre application et proposer une solution avec le moins d'images possible.

Pour la sauvegarde de vidéo, elle est fortement déconseillée, car le développeur ne pourra pas enregistrer beaucoup de contenu à côté de la vidéo qui devra en plus être réduite à moins de 5 Mo.

La mise en cache d'une application permet à la fois de pouvoir naviguer dans une application en mode déconnecté mais aussi d'améliorer la vitesse de chargement des pages puisque les fichiers sont enregistrés sur le téléphone en local. Le HTML 5 propose une vraie réponse au problème de connexion. Même si cette solution est moins connue du grand public, car elle est faiblement médiatisée par rapport aux balises vidéo et audio.

La mise en cache d'une ou plusieurs pages de votre application permettra une meilleure expérience utilisateur. D'une part, l'utilisateur n'aura normalement plus de page blanche assimilée à un dysfonctionnement de l'application et de plus l'application aura des performances de réactivité plus notables ce qui est important pour un client mobile.

2- La réactivité des applications internet

Dans l'univers du mobile, la concurrence est rude, la technologie étant encore jeune, le marché ne souffrent pas de demande et l'offre se veut innovante de jour en jour.

Dans une telle optique, la demande des utilisateurs sera portée sur les apports fonctionnels de celle-ci. Le client choisira entre deux solutions, fonctionnellement identique, la solution qui sera le plus agréable à utiliser que ce soit d'un point de vue réactif ou d'un point de vue esthétique.

i- La vitesse d'interprétation

La vitesse d'interprétation du langage, utilisé pour développer une application mobile, sera grandement influencée par le niveau du développeur. Mais un autre paramètre est à prendre en compte : **la compilation**. En effet, les applications natives sont toutes compiler, c'est-à-dire que les fichiers sont envoyés à un compilateur qui va traduire les fichiers dans un langage plus proche de la machine. Il faudra une version compiler par langage.

On ne peut pas compiler une application en Objective c pour des téléphones sous Android. Le problème de cette solution est la mise à jour d'une partie de l'application, car il faudra à nouveau compiler le projet et envoyer la mise à jour aux différents stores qui proposent l'application.

Les applications web sont développées avec des langages **interprétés***. Le JavaScript, l'HTML 5/CSS3 ainsi que le langage serveur (si votre application en a besoin). Les langages interprétés ont pour réputation d'être plus lents que les langages compilés, car un langage interprété aura besoin d'un interpréteur qui traduira les différents scripts pour que la machine les identifie.

Les applications web, dans leurs ensembles, utilisent énormément de JavaScript. Que ce soit pour des aspects visuels ou pour des rechargements de page via la technologie Ajax. C'est pour cela que les librairies JavaScript, pour téléphone portable, ont, dans l'ensemble, fait un travail sur la taille des fichiers. Il est beaucoup plus compliqué pour un téléphone d'interpréter 10 Ko de données, par rapport à un ordinateur qui aura une puissance de calcul bien supérieure.

De plus, l'HTML 5 ainsi que CSS 3 permettent d'alléger les scripts JavaScript. Google a porté son application Gmail en HTML 5. Les performances sont notables :

- Le temps de chargement de l'application a été réduit en moyenne de une seconde,
- L'utilisation du CSS 3 a permis une accélération du rendu graphique de 12 %,
- Environ 443.000 lignes de code en JavaScript, chiffre qui a diminué grâce au HTML 5.

Ces chiffres proviennent du site « objectdirect.com ». Ils montrent bien que des performances sont tout de même notables, même si nous ne pouvons connaître le nombre de lignes de JavaScript avant le portage HTML 5.

Les constructeurs de téléphone portable font de plus en plus attention aux performances de lecture du JavaScript. À l'image d'Apple, qui a créé un moteur JavaScript pour son navigateur mobile à savoir « Safari ». Ce moteur appelé « Nitro JavaScript » permet de compiler le JavaScript avant l'interprétation du navigateur.

Même si les navigateurs mobiles reposent sur la bibliothèque « WebKit* », bibliothèque qui utilise son propre moteur « SquirrelFish* ». SquirrelFish est un moteur JavaScript qui utilise un mode « compilé », mais il faut bien comprendre que le code JavaScript doit tout de même être compilé par SquirrelFish puis le code compilé est directement lu par la machine. Il y a quand même une étape de plus que sur des langages compilables de base.

ii- La vitesse de connexion

La vitesse de votre connexion est importante pour une application qui fonctionne dans un mode connecté. Surtout si votre application est une Web App. L'une des différences entre l'application native et l'application web est le chargement des pages qui la compose. Pour une application native, ces pages sont enregistrées sur le téléphone. Pour une application web, ces pages sont chargées depuis internet puis mises en sauvegarde sur le téléphone (pour fluidifier la navigation par la suite).

La vitesse de connexion permettra donc à une application web de charger les différentes pages qui la composent en un espace de temps plus réduit. C'est un gain de temps non négligeable vis-à-vis des scripts JavaScript qui peuvent par moment être très volumineux et demander un temps de chargement allongé.

Si l'application connectée (native ou web) offre des contenus médias, la vitesse de connexion en sera plus importante. Pour le chargement d'une vidéo ou d'une galerie d'image, la vitesse d'exécution sera bien plus agréable pour l'utilisateur.

Les applications mobiles évoluent avec le réseau téléphonique. Au début des années 2000, il n'était pas imaginable de regarder une vidéo disponible sur internet en lecture direct depuis son téléphone. Aujourd'hui, nous pouvons modifier des images, regarder des vidéos sur « You tube » ou encore écouter de la musique disponible en ligne sur internet.

Il y a différente manière pour se connecter à des services internet depuis son téléphone. Utiliser le réseau de l'opérateur téléphonique ou être connecté à un réseau WiFi. La connexion WiFi est de plus en plus utilisée en mobilité depuis que différentes mairies, en France, installent des bornes WiFi dans les endroits très fréquentés. Différentes entreprises proposent, aussi, le WiFi gratuitement dans leurs locaux, comme McDonald's ou la SNCF.

Mais les connexions mobiles sont souvent hors d'une couverture WiFi. La 3G offre un bon débit, mais cette connexion a des limites. Que ce soit sur des fichiers très volumineux ou des vidéos de très bonne qualité, sa vitesse n'est pas toujours adaptée à l'évolution de l'univers média.

Les différentes générations de téléphonie mobile			
Génération	Acronyme	Intitulé	Débit indicatif en bits/s (théorique/pratique/usuel)
1G	Radiocom 2000	Radiocom 2000 (analogique) de France Télécom, SFR 2000 (analogique) de SFR	analogique
2G	GSM (échanges de type voix uniquement)	Global System for Mobile Communication	9,05k
2.5G	GPRS (échange de données sauf voix)	Global Packet Radio Service	171,2k / 50k / 17,9k
2.75G	EDGE (basé sur réseau GPRS existant)	Enhanced Data Rate for GSM Evolution	384k / 64k / -
3G	UMTS	Universal Mobile Telecommunications System	1,9M / 144k rurale, 384k urbain, 2M point fixe / -
3.5G ou 3G+ ou Super 3G	HSDPA (dérivé de l'UMTS)	High Speed Downlink Packet Access	14,4M / 3,6M (version 5) 7,2M (version 6) / -
3.9G	LTE	Long Term Evolution	326M / 40M voir 80M après / -
4G	LTE-Advanced	Long Term Evolution Advanced	1G à l'arrêt, 100M en mouvement / - / -

Source : Wikipedia

Ce tableau illustre bien l'évolution du réseau téléphonique mobile français. Nous sommes actuellement au passage entre la 3G et la 4G. Comme on peut le voir, le réseau 4G va offrir des débits presque équivalents à des réseaux WiFi de faibles niveaux. D'après Frédéric Fleury – Devoteam — : « La 4G va faire exploser les usages autour de la télé-expertise et de l'apprentissage en ligne ». Le réseau 4G est bien plus performant que le réseau 3G. C'est le domaine multimédia qui en est le principal bénéficiaire.

Un exemple typique d'utilisation comme la possibilité de passer un entretien en déplacement et en appel vidéo depuis Skype. D'autres aspects sont améliorés comme le « cloud cumpting », l'accès à des « Google Docs » sera bien plus rapide avec l'aspect communautaire accéléré. On pourra utiliser le réseau pour faire passer de gros volume de données à analyser et donner le résultat presque en temps réel à l'utilisateur. Comme dans le domaine de la reconnaissance d'oiseau.

Ayant travaillé sur une application mobile de reconnaissance d'oiseau. J'ai eu à me poser un certain nombre de questions avec mon équipe. Nous avons choisi de prendre plusieurs photos successives des oiseaux dans le ciel pour avoir la meilleure qualité d'image. Or chaque photo représentée 10Mo, nous devions pouvoir envoyer

plus de 20 photos à la fois. Avec un téléphone sous 3G, il fallait compter plus de 5 min. Puis le serveur devait analyser les images pour ensuite retourner l'information au téléphone. Avec un réseau 4G, le résultat aurait été bien plus rapide, voire immédiat.

La vitesse de connexion est donc un point essentiel pour des applications connecter encore plus pour les Web Apps. L'utilisateur sera plus attiré par une application réactive que d'autres, plus lentes.

3- La compatibilité sur différentes plates formes

La mobilité héberge un grand nombre de marque de fabricant et de téléphone ainsi que de système d'exploitation. Chaque OS mobile propose en navigateur internet par défaut : Safari pour iOS ou encore Internet Explorer pour Windows Phone. Android n'a pas vraiment de navigateur par défaut, c'est le fabricant de téléphones qui installera le navigateur qu'il souhaite. Puis chaque OS est installé sur des téléphones qui ont des caractéristiques différentes. Plus on s'avance dans le temps, plus les téléphones offrent des vitesses de calcul intéressantes.

Ces différences sont importantes lorsque l'on parle d'application compilée ou d'application interprétée. On arrive vite à des limites matérielles pour certaines applications selon la plate forme.

L'autre différence entre chaque OS est l'aspect graphique de celle-ci. L'utilisateur d'un téléphone Android mettra plus de temps à s'habituer à un autre système d'exploitation comme iOS ou Windows Phone.

i- Pour les navigateurs

Les systèmes d'exploitation proposent généralement un navigateur internet par défaut. Or il faut savoir qu'il est possible d'installer un autre navigateur. Par exemple, sur iOS vous aurez le navigateur Safari installé, mais via l'Apple Store vous pourrez télécharger un autre navigateur comme Chrome ou Opéra mobile.

La différence entre chaque navigateur internet est l'interprétation des multiples scripts web. Sur le site de « StatCounter », on peut voir qu'entre décembre 2012 et février 2013 les deux principaux navigateurs utilisés sont : le navigateur d'Android avec 30 % des parts et le navigateur d'iPhone avec 23 % des parts. Ce sont aussi les plus avancés technologiquement pour répondre aux nouvelles fonctionnalités du

HTML 5 d'après l'entreprise « Dot Com Infoway ».

HTML 5 offre plusieurs nouvelles fonctionnalités, mais les navigateurs ne les prennent pas toujours en compte. On trouvera le plus de différence avec l'appréhension des formulaires. Un champ « date », qui offrira un calendrier pour sélectionner une date, ne sera pas compris par tous les navigateurs, seul Safari, Chrome et Opéra l'analyseront. D'autres champs ne sont compris d'aucun navigateur, on prendra l'exemple du champ « tel » qui permet de renseigner seulement un numéro de téléphone.

Les fonctionnalités ne sont pas les seuls à changer entre les différents navigateurs. L'aspect visuel change aussi, l'interprétation du CSS 3 peut être différente. Certains navigateurs vont être plus permissifs que d'autres. Des oublis dans le code, propre à des affichages de tableau ou des fermetures de balise manquante ou même des fonctions JavaScript non connus, ne seront pas identifiés de la même manière sur tous les navigateurs.

L'entreprise « Dot Com Infoway » nous présente des chiffres sur la compatibilité des applications web mobile et les fonctionnalités du HTML 5, sur téléphone portable. Le navigateur qui donne l'accès à plus de fonctionnalités est Firefox mobile avec 66 % des innovations HTML 5 prises en compte pour mobile. Puis suis Safari et Opéra. Il faut savoir que Firefox mobile n'est disponible que sur Android à l'heure actuelle.

De plus, l'entreprise nous explique que les usages ont de fortes chances de changer. En 2010, on comptait 109 millions d'applications web en HTML 5, « Dot Com Infoway » prévoit en 2016 une augmentation de 2 milliards.

Lors du développement d'une Web App, il faudra bien prendre en compte tous les navigateurs. Le mieux est de tester son application sur les navigateurs mobiles les plus utilisés :

- Safari sous iOS,

- IE sur Windows,
- Chrome/Opera sur Android.

Le test sur toutes les plates formes peut être cher si vous devez vous fournir un téléphone pour chaque OS. La solution est de faire croire au navigateur de votre ordinateur que l'utilisateur est un téléphone. Vous pouvez, pour ce faire, changer l'« User Agent ». L'« User Agent » permettra une interprétation du navigateur différente. Si vous changez votre « User Agent » en iPhone plutôt que MacBook pro, votre navigateur va interprété que vous êtes sur un iPhone. Par contre, il faudra bien entendu changer de navigateur et à nouveau changer l'« User Agent ».

ii- Pour les interfaces graphiques

En informatique, l'interface graphique est un point important, dans le développement d'une solution. Elle est aussi assimilée à une interface entre l'homme et la machine (IHM). L'IHM d'une application doit être pensé par rapport au domaine de l'application, mais aussi vis-à-vis de l'OS sur lequel l'application sera installée.

Certaines applications internet ont du faire de gros changements dans leur interface graphique pour s'adapter au format mobile. Maintenant, on peut les assimiler à des applications web.

On peut prendre l'exemple de Wikipedia ou encore « Domino's pizza » qui ont tous deux fait le choix de créer un site internet pour les versions mobiles.



Comme le montre cette image, Domino's Pizza a revu du tout au tout l'interface d'utilisation entre le site sur ordinateur et le site sur mobile. Sur le comparatif si dessus nous avons à gauche l'interface graphique de la Web App et à droite l'interface normale de l'application.

Une autre solution est de rendre votre site « responsive ». Le responsive design désigne les applications internet qui s'adaptent à la taille de l'écran. Mais l'application n'aura pas une interface optimisée pour l'appareil qui la lance. La différence entre la taille d'écran d'un smartphone et d'une télévision, dernière génération, est très importante, pourtant le principe du responsive permettra au site de s'afficher normalement sur les deux appareils. L'aspect interface homme-machine perd du sens dans ce cas, vu que la machine change du tout au tout.

Les applications natives sont sujettes à l'utilisation du modèle d'IHM de la plateforme ciblée. Elles peuvent suivre ce modèle ou créer leur propre structure. Généralement, les applications natives suivent le modèle de base de l'OS. Chaque

OS a sa manière de gêner les événements avec l'utilisateur. Comme la gestion de la suppression d'un élément, ou tout simplement la navigation dans l'application.

Les applications Web sont plus restreintes en terme de fonctionnalité graphique. Certains effets personnalisés sur une application ne peuvent pas toujours être reproduits sur une Web App, du fait de la différence entre l'application compilée et l'application interprétée. Une application web peut choisir son propre système d'interaction. Mais il faut savoir que l'utilisateur a déjà des principes de bas acquis avec l'utilisation d'application native, l'utilisateur va donc avoir des réflexes d'interaction, et s'il n'arrive pas à faire marcher votre application il risque de chercher une application concurrente.

Il faudra donc bien réfléchir à l'interface à utiliser lors de la création d'une Web App. Wikipédia est un très bon exemple de navigation simplifié sur application web. Car il propose toujours le même contenu que sur le site normal, mais l'information est aménagée pour être plus lisible par l'utilisateur.

4- Les limites des applications web

Les applications web ont des avantages non négligeables en terme de coût de développement, mais aussi de vitesse de création. La mise en ligne est simple et à la fois rapide. L'accessibilité est facilitée pour l'utilisateur. Il n'aura pas d'installation à mettre en place, juste avoir un navigateur internet. Mais le navigateur web doit être connecté à un réseau qui délivre internet. Nous atteindrons donc vite la **limite en terme de connexion** qui est de loin la plus contraignante.

Une application web peut aussi rencontrer des problèmes alors qu'elle a un accès à internet avec son téléphone portable. Des ralentissements sur internet peuvent se produire, causés par une trop grande affluence d'utilisateur sur votre application. Le serveur qui héberge votre application ne doit pas être situé à l'autre bout du globe. Pour des soucis de vitesse, car plus la distance entre le serveur et le téléphone est grande plus le temps de transfert d'information le sera aussi.

Lors du choix du serveur il ne faudra pas analyser seulement sa localisation, mais aussi si le serveur est mutualisé ou dédié. Si votre serveur est mutualisé, il sera partagé avec d'autres clients de l'hébergeur, vous serez de ce fait plus sujet à des problèmes de performance réseau et donc à des latences dans votre application web. Alors que le serveur dédié est consacré à vos applications. Vous pourrez en installer autant que vous aurez d'espace de stockage.

Si jamais l'application web, installée sur un serveur, connaît un grand succès et que l'infrastructure n'est pas suffisante pour répondre à la demande, votre application risque de ne pas pouvoir fonctionner normalement. Dans ce cas, il faudra changer de serveur avec des infrastructures capables de répondre à la demande des utilisateurs.

En général, plus votre application est utilisée, plus vos revenus seront importants. De ce fait, vous pourrez facilement adapter votre infrastructure sans problème.

Le choix des images est aussi important. Si vous faites le choix d'image de haute définition, vous risquez d'avoir des temps de chargements de page, sur votre application, qui seront allongés. Il faudra donc négliger un peu la qualité des images pour obtenir des performances plus notables. Et utiliser le plus possible les fonctionnalités du CSS 3 pour limiter le nombre d'images sur la page.

Il est possible de faire une sauvegarde de page en HTML 5, mais si votre application subit des ralentissements dus au réseau, l'appareil ne prendra pas la sauvegarde, via HTML 5. Il cherchera à atteindre la page sur le serveur, durant des phases de ralentissement votre application risque de ne pas être agréable à l'utilisation. La fonctionnalité de sauvegarde de contenu n'est utilisable que pour une perte totale du réseau.

La limite de connexion impacte le temps chargement de tous les script (CSS, JavaScript et HTML). Plus vous utiliserez différents scripts, plus votre application mettra du temps à se charger. Il faudra limiter les appels aux scripts qui sont vraiment utilisés par votre page. Car certaines librairies ne seront pas forcément utilisé dans telle ou telle page. Or, le chargement de librairie peut être contraignant. Surtout si elle n'est pas utilisée dans cette page.

Au-delà de la limite de connexion, on retrouve aussi une **limite en terme de ressource**. Les Web Apps n'auront pas accès à la totalité des fonctionnalités du téléphone. Vous n'aurez pas accès à tout le contenu du téléphone ou à différents

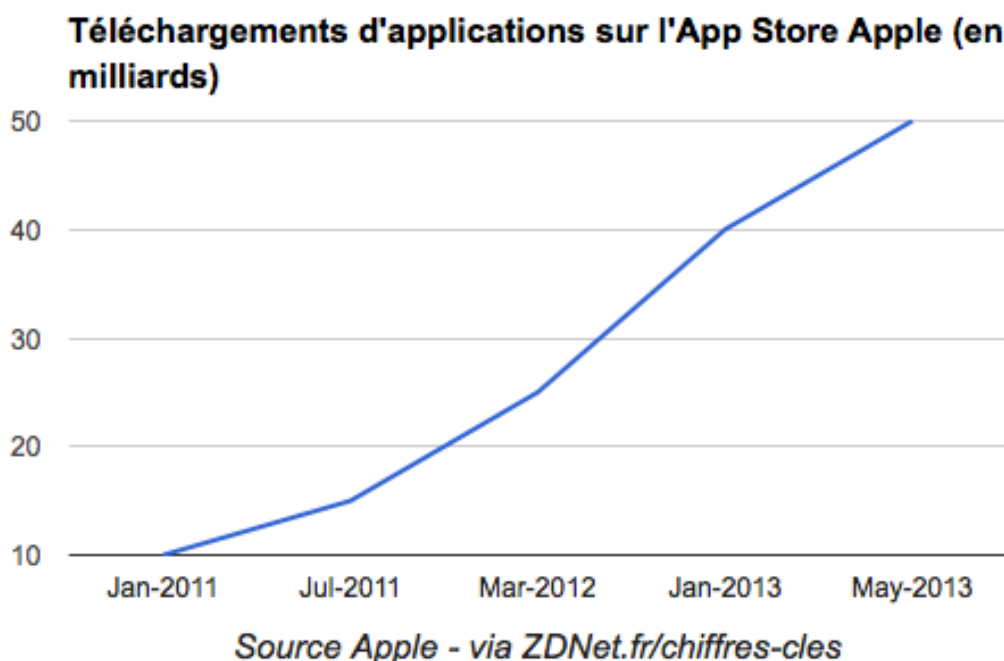
composants techniques. Comme le microphone, pour l'enregistrement de son, ou à la bibliothèque de photographie.

Dans le cadre de mon stage de fin d'études, effectué au sein de Sogeti Aquitaine, j'ai dû travailler sur une application web qui recevait des idées d'utilisateurs. Le client a demandé à ce que l'application enregistre du son pour que les utilisateurs envoient des idées par format sonore. Or, nous n'avons pas pu mettre en place cette évolution au sein de l'application web. Car nous avons atteint une limite en terme de ressource.

L'application web est une réponse au développement sur multiple plate forme et la problématique de compatibilité sur les différents appareils est difficile à prendre en compte. Certains téléphones portables peuvent afficher votre application web, mais là où un téléphone de dernière génération se comportera très bien, un téléphone moins récent risque d'avoir des latences, voire d'arrêt de calcul par rapport à sa propre puissance. Il faudra donc faire des tests. Et spécifier aux utilisateurs que certaines fonctionnalités ne sont pas disponibles sur les téléphones ne répondant pas à certaines caractéristiques techniques.

Les premières applications web utilisaient fortement l'ergonomie des applications iOS. Pour la navigation, l'utilisation des boutons typique des applications iOS (le bouton « Retou » par exemple) ou même l'utilisation de diverses icônes d'application native (comme l'icône d'accueil). Aujourd'hui, les Web Apps se veulent plus généralistes, car l'évolution des parts de marché a été nettement égalisée. Il y aura une seule ergonomie pour toutes les plates formes. Il est donc difficile d'utiliser des interfaces ou images propres à une plate forme.

Les applications natives ont accès à un store pour être téléchargées puis installées. Ce store permettra de référencer les applications natives disponibles sur l'OS. C'est un très bon moyen de mise en avant. Car les utilisateurs pourront faire une recherche précise et télécharger par la suite l'application voulue.



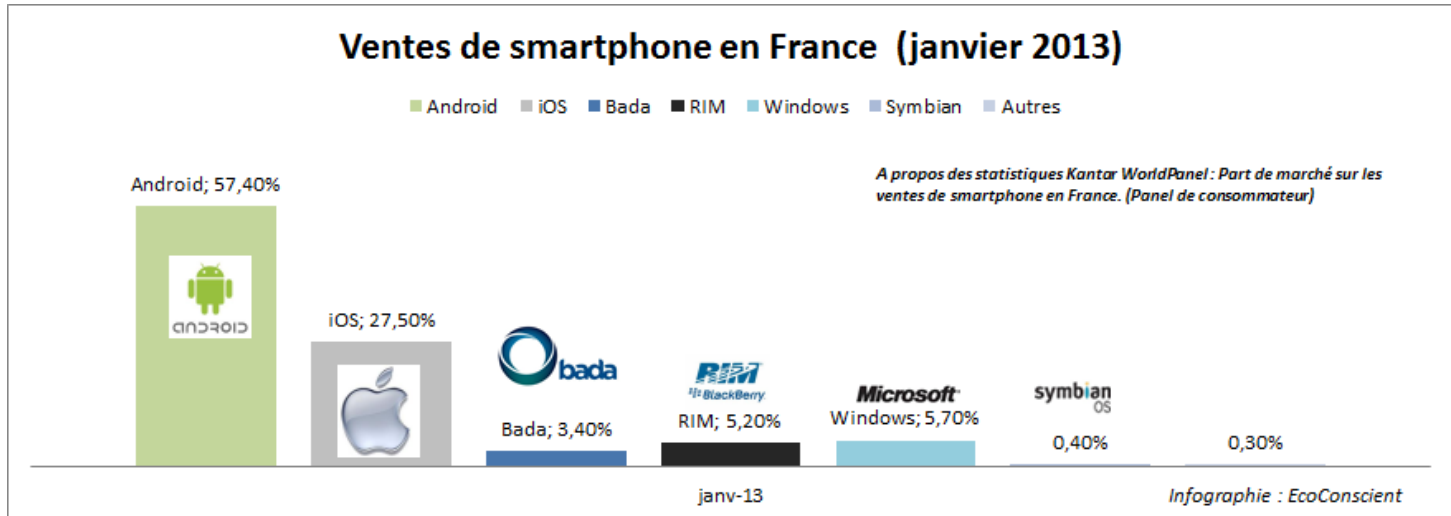
La courbe bleue représente le nombre de téléchargements d'application seulement sur l'Apple Store (les chiffres sont en milliard). À savoir que l'Apple Store regroupe, actuellement, environ 890.000 applications. Statistiquement, votre application sera téléchargée 55.000 fois. Mais les statistiques sont faussées du fait de la popularité d'un petit nombre d'applications vis-à-vis d'autres applications moins connues avec peu d'investissement sur le marketing autour de l'application.

Le problème de l'application web est son référencement. À ce jour, il n'y a pas d'annuaire d'application web performant. Certes, Google permet de retrouver des

sites ou applications via des mots clés, mais il ne fait pas encore la différenciation entre Web App et site internet pour ordinateur.

On atteint vite une **limite en terme de visibilité**, on dit qu'actuellement seules les personnes connaissant la technologie savent qu'il existe un système d'application web. Il reste à ce jour un gros travail à effectuer sur le référencement des applications web.

III Spécificité des plates formes



Comme on peut le voir sur ce graphique, les ventes de téléphone portable sont séparées entre plusieurs OS. Android et iOS représentent pas moins de 85 % des ventes de téléphone pour une courte période (janvier 2013). Or en France, 44 % de la population (11ans et plus) possède un smartphone, soit 24.1 millions.

Lorsque vous décidez de créer une application mobile seulement sur la plateforme iOS vous toucherez 28 % de la clientèle. Si vous choisissez de la développer sur Android, vous toucherez une plus grande population, mais ce chiffre peut être erroné. Car Android abrite un grand nombre de téléphones différents, dont les téléphones situés au premier prix. Ces téléphones sont en général vendus à des clients qui n'utilisent pas toutes les capacités du téléphone. En moyen le nombre de connexions via un téléphone sous iOS est plus élevé qu'Android. Il faudra donc faire aussi attention au public que l'on vise selon la plateforme.

Les clients à la recherche d'une solution d'application mobile ont tendance à demander des applications Android et iOS. Mais il ne faut pas oublier les 15 % de détenteur de smartphone qu'il reste. En effet, Black Berry et Windows Phone regroupe pas moins de 3 millions de client potentiel.

Le choix d'un développement d'application multiplateforme ou d'OS spécifiquement choisi doit être comptabilisé. La contrainte financière est très importante, car le prix peut être multiplié par 3 pour une application multiplateforme par rapport à une application dédiée à un seul système.

Un développement d'application native, mais onéreuse, ou d'application web multiplateforme, mais étant limité en terme de connexion, ressource et visibilité, doit être fait.

1- Développement natif, les différents stores

Le développement natif est certes plus coûteux qu'un développement web, mais offre à l'utilisateur un panel bien plus élargi. Il est bon de sélectionner les plateformes susceptibles de générer le plus de téléchargement pour votre application.

App Store

La plateforme iOS est apparue en juin 2007. Près d'un an après son lancement Apple communique des chiffres : 3.000 applications recensées pour 100 millions de téléchargements. Cinq ans plus tard, Apple annonce un total de 50 milliards d'applications téléchargées sur leur store pour 900.000 applications. On voit bien que l'évolution est plus que présente.

Même les revenus versés aux développeurs d'Apple sont en augmentation. En moins d'un an, ils ont plus que doublé, passant de 4 milliards à 10 milliards. Mais selon une étude menée par « Canalys », 50 % des revenus versés par Apple sont en réalité perçus par 25 sociétés qui ont le meilleur rendement.

Il faudra donc bien faire attention à ce que l'application ne soit pas perdue dans la masse. Une étude de « Adeven » révèle que 580.000 applications sur l'App Store peuvent être considérées comme « morte », car elles ne génèrent plus de téléchargement.

Windows Phone Store

Windows Phone est paru début 2010. En juin 2013, le Windows Phone Store comptait 160.000 applications, pour 2 milliards de téléchargements d'application. Les chiffres sont bien en deçà vis à vis d'Apple, mais Windows Phone est sorti 3 ans après l'apparition d'iOs. 3 ans après la sortie d'iOs l'App Store, compter 250.000 applications soit 90.000 de plus que Windows Phone Store. Mais Apple avait un avantage sur la concurrence, car à cette époque ils étaient moins nombreux à proposer des services adaptés à la nouvelle génération de mobilité.

Windows Phone connaît une évolution croissante. En 2014, l'IDC prévoit 21 % des parts de marché pour les Windows Phone dans le monde. Cette croissance est la plus importante toute plate forme confondue. Elle pousse à la non-négligence de cette plate forme pour les futurs choix stratégiques.

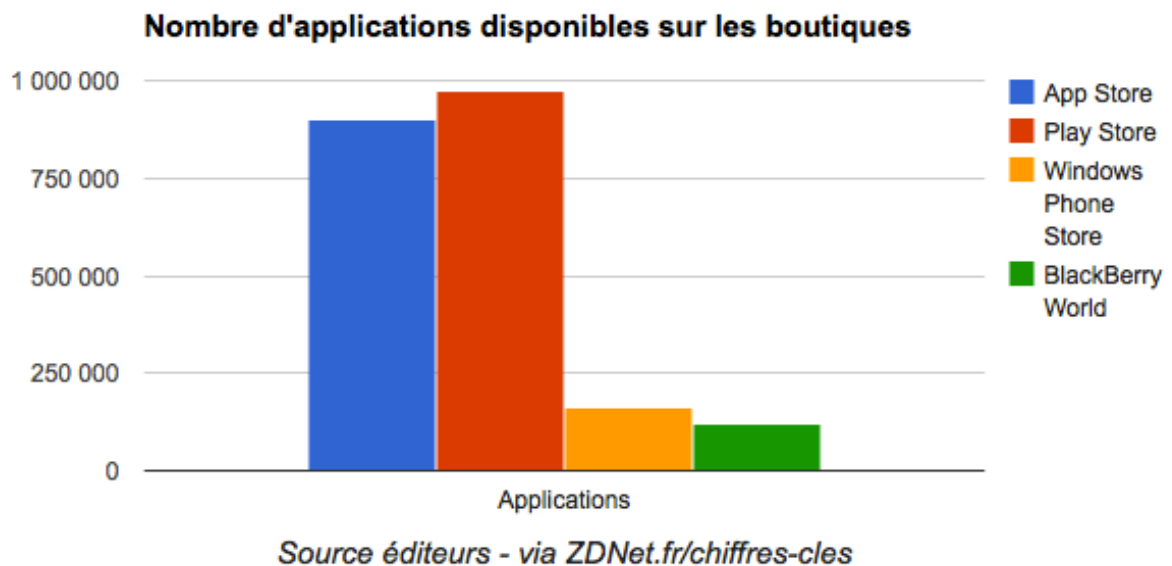
Play Store

Android est né en novembre 2007, il est aujourd'hui le système d'exploitation le plus utilisé dans le monde. Android touche une plus grande masse communautaire, car il est installé sur les téléphones de basse gamme, moyen et haut de gamme. Ce qui peut être un atout dans les chiffres mais en terme de monétisation sur le store, Play Store n'est pas le premier.

C'est l'App Store qui monétise le mieux la vente d'application. Au deuxième trimestre 2013, le Play Store gagnait un revenu évalué à 47 millions de dollars alors que l'App Store engrangeait environ 100 millions de dollars.

Le Play Store abrite un peu moins de 1 million d'applications, soit presque 100.000 de plus que l'App Store. Il est donc aussi difficile de faire connaître une application sur Android que sur iOS.

Les deux principaux stores d'application mobile sont l'App Store et le Play Store. Le Windows Phone Store est par la suite le plus prometteur. Pour les autres plates formes, seul BlackBerry World survie dans la marche du mobile.



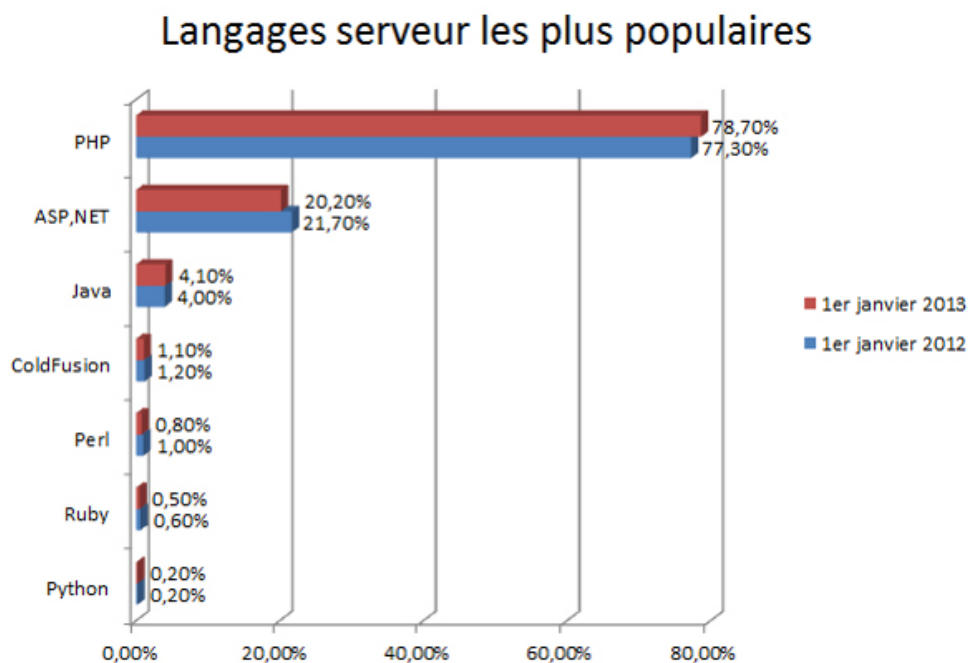
On voit que malgré son ancienneté BlackBerry World n'est pas attractif dans le monde du développement. Cela influe sur la qualité du store et donc sa fréquentation.

2- Développement serveur

La partie serveur est la partie invisible d'une application mobile. Mais les applications qui montrent des contenus ou enregistrent des contenus sur internet sont reliées à un serveur. Nous avons vu qu'il existe plusieurs types de langage serveur. Il serait favorable de savoir lequel choisir pour le développement d'une architecture favorable au bon fonctionnement de votre application.

i- Connexion entre le serveur et le client, le choix du langage

Le serveur occupe la place de traitement de donnée, il effectuera les traitements lourds de l'application, et synchronisera les données de chaque application dans une base de données.



Analyse de W3Techs

Sur ce graphique, on aperçoit que le PHP est très populaire dans le développement serveur. Mais nous avons vu qu'il n'était pas forcément la meilleure

solution pour des applications volumineuse. C'est la même problématique pour les applications mobiles. Le client (téléphone portable) communique avec le serveur via des **web services***. Il n'y a pas de langage plus adéquat pour effectuer des appels avec un système de web service. Le but est de communiquer entre le serveur et le client pour administrer une base de données.

Il faudra donc faire son choix par rapport à d'autres caractéristiques du langage. Car chaque langage est certes différent, mais effectuera au final les mêmes actions. Il faudra faire son choix par rapport à des éléments extérieurs au développement.

ii- Différentes manières pour différents langages

Le choix du langage serveur a évolué dans le temps et repose maintenant sur plusieurs critères. On retrouvera par exemple :

- Le Management
- L'environnement de l'entreprise
- La durée de vie du projet
- Les ressources

Il est rare de choisir un langage serveur par rapport à ces apports techniques, car l'ensemble des technologies serveur permet d'effectuer les mêmes traitements.

Management

L'aspect du « management » est propre à une équipe de développement. Une équipe travaillant avec la méthode Agile* choisira un langage plus optimisé dans ce

domaine comme le Ruby On Rails. Le PHP est par exemple le langage qui demandera le plus de travail pour l'adapter à une équipe Agile.

L'environnement de l'entreprise

L'environnement de l'entreprise est la différenciation entre une structure importante comme une SSII et une PME ou « start-up² ». Dans le cas d'une SSII, le choix de la technologie se fera selon les connaissances de l'entreprise. En effet, une SSII choisira un langage serveur qu'elle maîtrise plutôt que de lancer des projets dans un langage inconnu qui nécessiterait la formation d'une équipe et le potentiel recrutement de nouvelles ressources. Alors que les plus jeunes structures se pencheront vers des technologies plus récentes. La société Snapp' est un bon exemple de « start-up » bordelaise travaillant dans le monde du mobile et ayant choisi le Ruby On Rails comme langage serveur. Les points positifs, des nouvelles technologies serveur, sont leur temps de mise en marche plus rapide et leur orientation vers un développement suivant la méthode Agile.

De plus, une fois que l'entreprise a choisi une technologie (ou plusieurs), le choix est souvent préétabli. Vis-à-vis des connaissances engrangées lors des premiers développements de projets ou des interfaces réseau mises en place. La reprise de projet client est aussi un exemple de technologie serveur prédéfini, il serait dommage de recréer un projet, déjà abouti, ou d'ajouter de nouvelle fonctionnalité avec un langage différent.

La durée de vie du projet

Le choix de l'environnement serveur est parfois défini par la longévité du futur projet. Par exemple, la société Cdiscount est à l'origine d'un choix qui suscite des interrogations actuellement. L'application Cdiscount est développée avec le langage

² Entreprise possédant un fort potentiel de croissance, mais ayant un risque d'échec important.

« .NET ». Le « .NET » est une technologie propriétaire, donc couteuse. Aujourd'hui, Cdiscount réfléchit sur la possible orientation vers une solution « open source » qui serait moins couteuse sur le temps, mais nécessiterait un investissement colossal pour la recréation de l'ensemble de la structure serveur.

Les ressources

Les ressources disponibles au sein d'une entreprise ont des connaissances dans une ou plusieurs technologies. Il faudra donc orienter le choix vers la technologie que maîtrise le mieux l'équipe. Il faudra faire attention à la différence de prix, sur le marché de l'emploi. Entre un développeur de Java JEE et un développeur de PHP, la différence peut être de 4.000 euros bruts par ans

Le choix de la technologie serveur n'est pas à négliger, car elle sera l'outil qui permettra d'administrer et d'effectuer des calculs poussés pour votre application mobile native ou web.

3- Les applications hybrides

Les applications hybrides représentent des applications natives qui encapsulent des contenus internet. C'est une réelle réponse aux problématiques des Web Apps. En utilisant cette technique, il est possible de combiner à la fois une application web avec divers composant natif.

i- Principe de fonctionnement

Les applications hybrides fusionnent à la fois le côté natif et aussi le côté application web. Les applications hybrides sont développées avec les librairies JavaScript, vue au début du mémoire.

Au début, l'application est assimilé à une Web App, mais il est possible par la suite, via des composants natifs, d'encapsuler des programmes web, on parle dans ce cas d'application hybride. Le framework PhoneGap est une librairie JavaScript qui permet de faire un pont entre l'application web et l'application native. Lors de l'encapsulation de l'application web dans l'application native, PhoneGap fera le pont entre les fonctionnalités natives et web. Comme l'appareil photo, l'accéléromètre ou encore le vibreur de l'appareil.



PhoneGap permettra donc de créer des applications hybrides plus facilement et de les proposer sur plusieurs plates formes à la fois.

Il est donc possible de développer une application native en utilisant du contenu web. L'application utilisera les composants HTML 5 et CSS 3. Vous aurez donc une possibilité de manoeuvre au niveau graphique et fonctionnel assez élargi. De plus, votre application pourra être portée pour toutes les plates formes compatibles avec cette solution (Windows Phone, iOS, BlackBerry ou même Android).

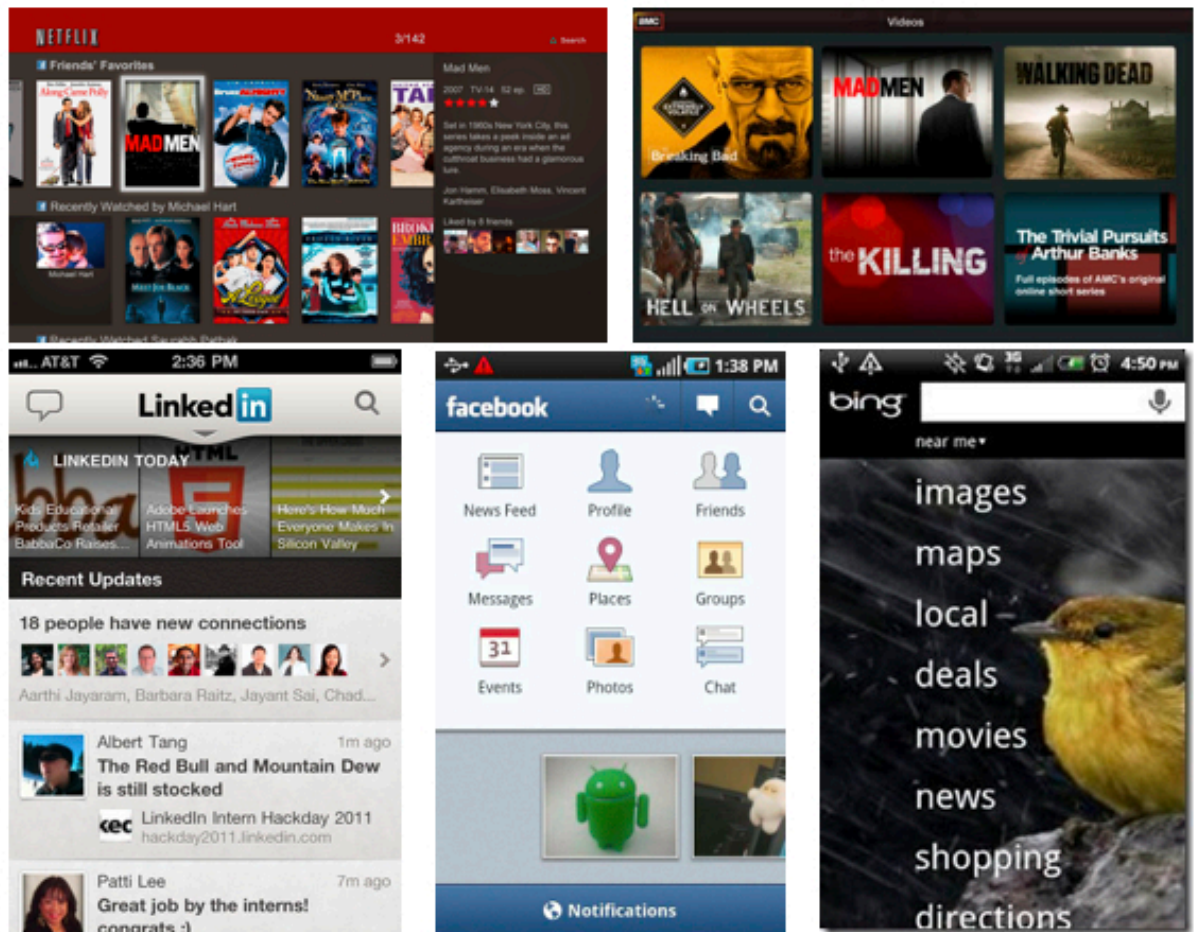
PhoneGap n'est pas le seul outil proposant cette solution. On retrouve dans le même registre Titanium Mobile. Les deux solutions sont gratuites, la différence fondamentale entre Titanium et PhoneGap est l'interprétation du JavaScript. Titanium est plus proche de l'appareil alors que PhoneGap ait plus basé sur les pages web est leur interprétation. Les performances de vitesse sont donc à l'avantage de Titanium alors que PhoneGap sera plus simple à prendre en main et à adapter.

Titanium supporte quatre plates formes différentes : iOS, Android, BlackBerry et récemment Windows Phone. PhoneGap en supporte neuf soit plus de deux fois le support de Titanium. Mais Titanium peut être utilisé sur les plates formes qui englobent le plus d'utilisateurs.

Les Web Apps peuvent donc être encapsulés dans des applications natives via un framework ou directement en utilisant quelques lignes de code natif à la portée de n'importe quel développeur.

Cette technique est présente depuis le commencement des smartphones, mais elle commence à être de plus en plus utilisée par les développeurs.

ii- Exemple l'utilisant



Voici des exemples d'application hybride. Elles sont toutes disponibles sur le store de leur plate forme, mais utilisent un contenu web.

L'exemple de Facebook est très parlant quant à cette mouvance du HTML 5. En 2007, Facebook a choisi de créer leurs applications de manière native. Mais en 2011, après avoir constaté que leur effectif était plus spécialisé dans le domaine du web par rapport au domaine du développement mobile natif (Objective-c, Java, .NET...) , Facebook a choisi de passer sur une méthode hybride pour harmoniser leur solution mobile vis-à-vis des différentes plates formes.

Ce choix est à la fois porté sur un problème de ressource, mais on peut aussi se douter qu'au niveau de la maintenance des applications natives, cela devait demander plus de temps. Car il fallait maintenir au temps d'application mobile qu'il y avait de plates formes couvertes par les différentes applications native.

Les applications web ont un autre point fort, leur portabilité au-delà des téléphones ou des tablettes. Une application web pourra être utilisée aussi bien sur une console de salon que sur une télévision connecter à internet. L'application hybride pour concrètement être adapté bien plus facilement qu'une application native, à bien plus de terminaux.

Les applications hybrides peuvent être diverses et variées. Dans le cadre de mon stage de fin d'études, j'ai eu à travailler sur une application hybride. À l'origine, nous avions une application web que nous devions faire porter en natif pour ajouter des fonctionnalités d'enregistrement audio ou l'ajout d'un système de notification. Le choix s'est donc porté sur une solution hybride pour limiter le temps de développement. Les améliorations ont été ajoutées avec succès et simplicité.

Des applications hybrides couvrent des domaines très différents, de la prise de photo (PixWithMe) à la navigation sur une carte (Google Map), on retrouvera presque tous les types d'applications. Seules les applications demandant énormément de ressource graphique, comme les jeux vidéo 3D, sont absentes à cause de souci de performance.

iii- Les enjeux

Nous avons vu qu'il existe à l'heure actuelle trois possibilités pour créer une application mobile :

- L'application native,
- L'application web,
- Et l'application hybride.

L'application native

L'application native possède plusieurs points positifs. La visibilité sur le store de la plate forme étant son principal, car le store lui offre un potentiel de clientèle très intéressant ce qui permettra de plus facilement faire connaître son application. Les applications natives ont aussi accès à toutes les fonctionnalités proposées par le système d'exploitation. Que ce soit la gestion de l'appareil photo du terminal ou l'accès au système de notification de la plate forme. Des animations pousser peuvent aussi être mises en place. La réactivité de l'application sera bien mieux prise en compte par rapport aux deux autres solutions, car l'application native est la seule qui propose un code totalement compilé.

L'application web

L'application web a la possibilité d'être utilisée sur n'importe quel téléphone portable qui possède un navigateur internet. Le coût en développement est donc réduit. La maintenance de l'application sera elle aussi bien moins couteuse, par rapport à la maintenance de plusieurs applications natives proposant le même

contenu. De plus, les applications web ne sont pas soumises aux taxes des différents stores, mais ne pourront pas y être référencées.

L'application hybride

L'application hybride est économiquement plus proche de l'application web. Le coût de développement pour une application multiplateforme sera au même niveau qu'une application web. L'accès au store est le point positif le plus important. Certes, le développeur devra payer des taxes aux différents stores, mais il aura la possibilité de gagner en visibilité.

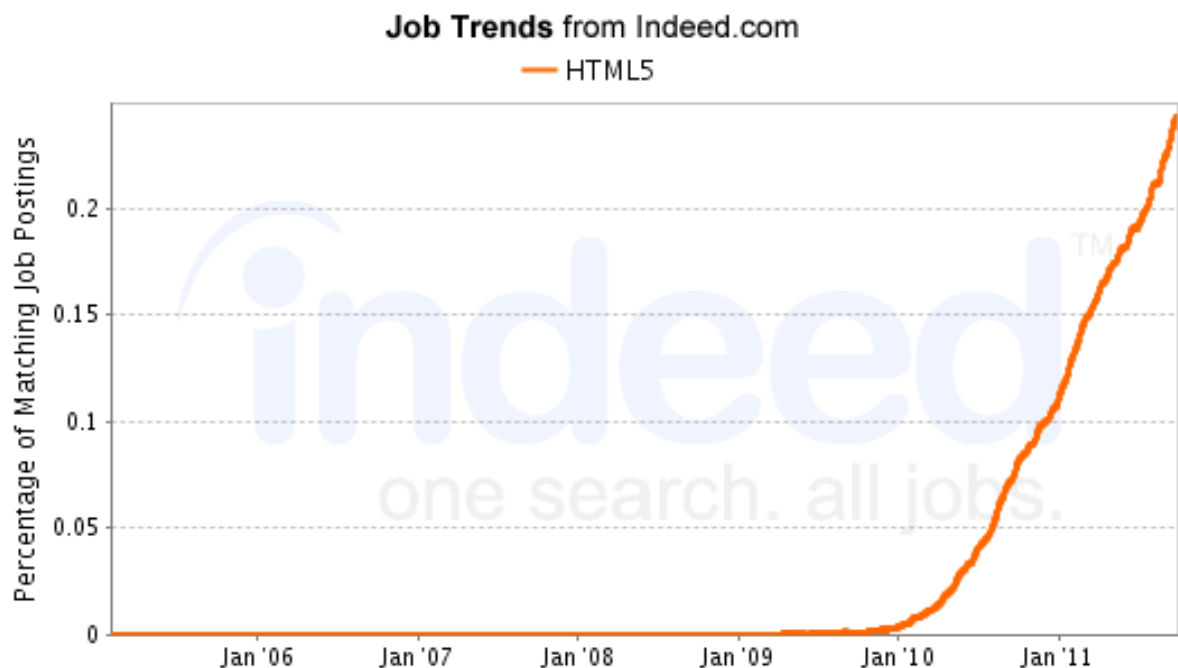
Le tableau ci-dessous illustre bien les enjeux de l'application hybride. Il montre que les applications hybrides attirent à la fois les avantages de la solution native et à la fois les avantages d'une solution web. Mais au niveau de la vitesse, il omet de préciser que les parties, de l'application hybride en HTML 5, seront forcément plus lente qu'une application native.

	Accès aux fonctionnalités de l'appareil	Vitesse	Coût de développement	App Store	Processus de validation
Native	Complet	Très rapide	Cher	Disponible	Obligatoire
Hybride	Complet	Vitesse native	Raisonnable	Disponible	Obligatoire
Web	Partiel	Rapide	Raisonnable	Indisponible	Aucun

(Source : Worklight).

Le marché de l'emploi est lui aussi en évolution. Il est aujourd'hui plus facile de trouver des développeurs web que des développeurs d'application mobile native. Pour une simple application portée sur trois plates formes, il faudra trois développeurs pour une application native contre un pour une application hybride.

Une étude sur le marché de l'emploi américain montre que la tendance HTML 5 est de plus en plus présente sur le marché de l'emploi :



Depuis le lancement du développement mobile à grande échelle et l'élaboration de diverses bibliothèques JavaScript spécialisées dans le mobile, le profil de développeur HTML 5 est de plus en plus présent.

L'évolution constante des ventes de smartphone et tablette est représentée par une clientèle potentiellement conséquente (24 millions en France). L'enjeu est

donc de pouvoir proposer une application mobile sur les différentes plates formes à l'ensemble de la clientèle.

L'application hybride permettra donc l'élaboration d'applications multiplateformes qui resteront proches fonctionnellement des applications natives. Les ressources seront plus simples à trouver et le temps de développement sera diminué. Cela induira directement sur le coût total de l'application mobile. L'application pourra toucher la majorité du parc téléphonique français, qui ne cesse d'augmenter.

iv- Les avantages et inconvénients

Le développement hybride possède des avantages et des inconvénients vis-à-vis du développement natif ou du développement web.

Les **avantages** sont multiples :

- Les diverses bibliothèques JavaScript permettant le développement hybride,
- la maintenance,
- le développement multiplateforme,
- le coût de développement réduit,
- un langage standard,
- la gestion des modes connecter et déconnecter,
- l'accès aux fonctionnalités des téléphones portables,
- l'accès au store d'application mobile,
- la gestion des versions de mises à jour,
- l'accès aux modèles économiques des applications natives,
- et l'accès à l'espace de stockage de l'appareil.

Les **inconvénients** sont moins nombreux, mais tout de même présents :

- la performance graphique,
- la performance en terme de vitesse,
- le respect des normes graphique,
- l'installation de l'application.

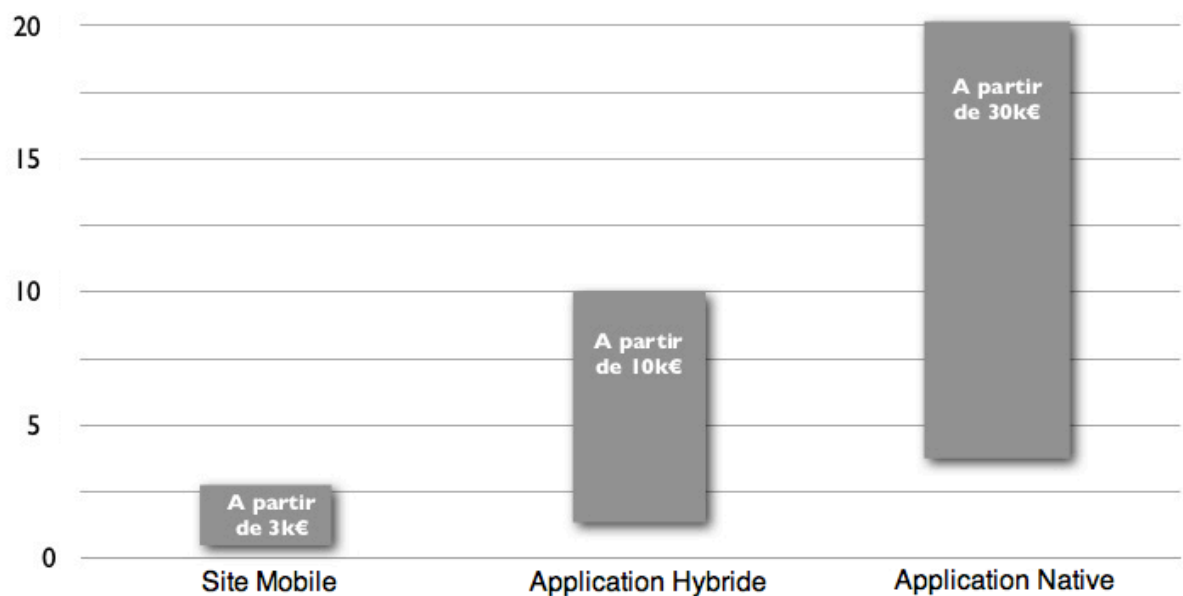
On retrouve donc plus d'avantages que d'inconvénients. Pour ce qui est des avantages, on se rend compte que le principal reste le développement multiplateforme.

Le développement multiplateforme permettra donc de couvrir une plus grande clientèle. Certaines applications, comme des applications pour des campagnes politiques, sont plus propices au développement multiplateforme. Grâce à cet avantage, il en découle d'autres. Comme l'utilisation d'un langage plus standard (HTML, CSS, JavaScript) par rapport aux multiples langages natifs. Puis on retrouve la maintenance qui est simplifiée puisqu'il n'y a qu'une seule application à maintenir.

Il est vrai que trouver une ressource pour du développement hybride est plus évident que de devoir chercher plusieurs ressources pour du développement natif multiplateforme. Surtout que le développement hybride a la chance d'être assisté par des bibliothèques JavaScript comme PhoneGap, Titanium ou encore une solution Ruby avec Rhodes.

La maintenance d'une application sera donc bien moins coûteuse dans le temps et prendra bien moins de temps et de ressource. Car à l'inverse d'un développement natif multiplateforme, vous n'aurez qu'une seule application à maintenir.

Le développement hybride d'application multiplateforme permettra d'obtenir des temps de création et des coûts bien plus intéressants que l'application native.



Source : Needeo

Ce graphique, proposé par la société Needoe, montre la différence en le prix d'application hybride par rapport à des application Native. L'application native est évalué à presque trois fois le prix d'une application hybride.

La solution d'une Web App souffre d'un lourd défaut vis-à-vis de la connexion permanente ainsi que la perte du référencement via un store. La gestion du mode déconnecté, en HTML 5, n'est pas forcément la meilleure solution. Lorsque l'on s'aperçoit que le développeur dispose d'un espace de stockage de 5 Mo avec une application web, alors qu'en utilisant un système hybride le développeur dispose d'un accès direct au stockage du téléphone.

En plus de disposer de l'espace de stockage du téléphone, le développeur pourra tout aussi accéder aux fonctionnalités telles que la caméra, le microphone ou encore le système de notification. Les applications hybrides ne souffriront pas d'un manque de fonctionnalité comme une Web App. Mais gardons leur souci de latence vis-à-vis d'une application native. Les normes graphiques, de chaque type d'application native, seront aussi difficiles à mettre en place pour une application

hybride.

Les applications hybrides bénéficieront de l'accès au store de vente d'application mobile de chaque plate forme. Contrairement à un Web App, une application devra t'être installée sur le téléphone portable. Mais l'application hybride pourra, tout comme une application native, accéder au gestionnaire de version des stores et être mises à jour à la fois par la partie internet, mais aussi via les stores des différentes plates formes.

L'accès au store permettra aux applications hybrides de s'adapter aux différents modèles économiques natifs. Les solutions hybrides obtiendront donc la possibilité d'être monétisé, ou d'ajouter les systèmes de publicité des différents OS. L'application hybride pourra aussi proposer des contenus payant, exactement comme une application native. Mais l'application hybride pourra aussi se reposer sur les modèles économiques d'internet. Le modèle hybride obtiendra un choix de modèle économique bien plus varié qu'une application native.

Conclusion

Romain Pellerin : « *Récemment nous avons vu cette application pour valider sa fiche d'imposition... Donc finalement nous n'allons l'utiliser qu'une fois par an... un site mobile aurait suffi* ». L'application web ne répond pas encore à la demande des utilisateurs mobiles. La Web App souffre d'un manque de visibilité ainsi qu'un lourd défaut lié au problème de connexion à internet. Mais comme le souligne Romain Pellerin, la Web App répond à des types d'application éphémère, qui n'ont pas pour but une utilisation courante. Faire le choix d'un développement d'application web à l'heure actuelle est un choix propre à l'application, car les Web App ne peuvent pas, à l'heure actuelle, écraser le marché des applications natives.

L'application web ne pourra pas être démocratisée tant que les problèmes de connexion et de visibilité ne seront pas résolus. Le souci lié à la connexion à internet risque d'être réglé dans les prochaines années avec l'arrivée de la 4G. Mais pour ce qui concerne le souci de référencement à ce jour Apple proposé des applications web mobiles sur son site, mais leur service a pris fin en juin. Il est très difficile de trouver d'autre référenceur d'application web mobile sur internet.

Le choix du développement web dans la mobilité est à ce jour un choix financier plus que technique. Les ressources sont plus simples à trouver et le temps de développement sera nettement réduit pour du développement multiplateforme. La mise à jour de l'application web sera aussi facilitée. Le développeur devra mettre à jour les fichiers web sur le serveur, alors que si l'application était native il faudrait passer par un processus de validation pour certains systèmes d'exploitation (iOS, Windows Phone).

L'application native est historiquement la plus utilisée, car elle était la solution proposée par les différentes plates formes au début de l'ère des smartphones. L'application web a été plus utilisée après 2009, avec l'arrivée des diverses bibliothèques JavaScript. Aujourd'hui l'application hybride nous permet de trouver une réponse par

rapport à la place des applications internet dans la mobilité.

L'application web mobile élaborée est donc présente sous forme d'application hybride. L'application hybride qui est la fusion entre application web et application native.

L'application hybride pourra à la fois gérer plus facilement la partie connexion à internet, mais permettra aussi l'accès aux stores des différentes plates formes. Les principes de monétisation deviendront donc plus en adéquation avec la partie mobile (vente de l'application, vente de contenu), mais l'application hybride gardera aussi les modèles économiques web qui sont bien plus nombreux (et ils ne sont soumis à aucune taxe vis-à-vis des propriétaires des systèmes d'exploitation).

Les librairies permettant l'élaboration d'application hybride sont créées en JavaScript et permettent l'accès aux fonctionnalités du téléphone portable. Comme le microphone, la gestion de la caméra ou le système de notification. Le développeur pourra donc se limiter à la connaissance du langage JavaScript pour mettre en place son application hybride.

Mais l'application hybride qu'avait mise en place Facebook, en 2011, n'a pas convaincu le grand public. Et un an plus tard, Facebook est revenu en arrière dans ces choix de technologie. Les utilisateurs déploraient une application trop peu fluide et par moment les déconnexions étaient mal gérées par le programme, et l'affichage de page blanche sur l'application était constaté. Le passage en langage natif de Facebook a permis de rendre l'application bien plus fluide et de gérer les problèmes de connexion. C'est donc à nouveau un changement d'objectif qu'affiche Facebook qui avait déjà transformé ses applications natives vers un modèle hybride en 2011.

L'application hybride n'est peut-être pas le modèle le plus perfectionné, mais il a le mérite de proposer un développement multiplateforme à moindre coût. Mais ce

n'est pas encore une technologie assez mature pour pouvoir supplanter les applications natives.

Actuellement, le choix entre l'application native, l'application web et l'application hybride doit être fait suivant la nature du projet (voir annexe 1). L'application hybride ne sera pas adaptée pour des applications demandant de la fluidité et de la vitesse.

Sur le store d'iOs, il est fréquent de télécharger et installer des applications natives qui auraient pu être développées avec une solution hybride comme des applications affichant des résultats sportifs (football, basket...). Ce genre d'application demande un système de notification très performant, mais ne cherche pas à obtenir l'accès aux fonctionnalités du smartphone ou d'afficher des animations très élaborées. Le choix d'une application hybride aurait été plus judicieux dans un souci de portage de l'application dans un mode multiplateforme.

L'application hybride n'est donc pas encore assez connue et pas assez utilisée par les petites entreprises. Pourtant c'est une solution utilisable dans bien des cas, et cette méthode permettra de réduire les financements d'une application mobile multiplateforme.

Aujourd'hui, les systèmes d'exploitation mobile ne sont pas adaptés à une utilisation qui consacre toutes les applications pour le web. Même s'il essaye de tirer au mieux les performances JavaScript (comme Apple avec son moteur JavaScript développé spécialement pour Safari Mobile).

Des systèmes d'exploitation basés sur les technologies du web ?

Si les OS actuels ne permettent pas une optimisation des temps de calcul et des performances graphiques adéquates au monde du web, peut-être que l'avenir du web se trouve dans les futurs OS mobiles.

Trois nouveaux systèmes d'exploitation vont voir le jour dans une optique qui est plus axée sur le web :

- **Tizen**, qui est un projet de la fondation Linux,
- **Firefox OS**, qui a été élaboré par Mozilla,
- Et **Open WebOS**, qui est la version open source de webOS développée par HP.

Firefox OS est disponible au total dans cinq pays. Trois pays européens l'Espagne, le Portugal et la Pologne et deux pays d'Amérique du Sud le Brésil et le Venezuela. C'est une plateforme très jeune, sortie en juin 2013. La société Mozilla compte s'installer dans d'autres territoires avant la fin de l'année. Ce système propose une interface utilisateur entièrement en HTML 5 et CSS 3. Firefox OS est basé sur un noyau Linux appelé Gonk, pour son fonctionnement. Contrairement à Tizen, Firefox OS utilise des composants Android pour l'accès à diverses fonctionnalités du téléphone portable. Le moteur de rendu JavaScript est Gecko développé par la fondation Mozilla. Gecko est un moteur ouvert, il peut être utilisé par d'autres plateformes comme Android, mais certaines fonctionnalités risquent d'être fermées.

Les applications mobiles, proposées par Firefox OS, sont entièrement faites en HTML 5 et JavaScript. En s'appuyant sur ces technologies, ce système d'exploitation met en première page le web dans la mobilité. Mais contrairement à une Web App, qui est disponible sur toutes les plateformes, les applications Firefox OS développées en HTML 5 seront compatibles seulement avec cet OS. Même si la technologie est la même, les sources JavaScript ne vont pas être adaptées pour

n'importe quel OS. Car les différents systèmes d'exploitation (iOS, Android, Windows Phone...) ne permettent pas de gérer les interactions entre le JavaScript et les composants du téléphone portable.

Tizen est proposé par la fondation Linux, il repose sur un noyau Linux couplé avec du HTML 5 pour l'interface graphique. Tizen utilise la librairie EFL (Enlightenment Foundation Libraries) pour ce qui est de la gestion graphique. WebKit est lui utilisé pour ce qui concerne le contenu web du système d'exploitation.

Tizen reçoit le soutien de plusieurs constructeurs de téléphone portable, comme Samsung ou Nokia. Samsung ayant abandonné le système d'exploitation mobile « Bada », va installer Tizen sur sa gamme de « Samsung Wave³ ». Cela représente un marché important, car Samsung est le constructeur de téléphone portable qui obtient depuis quelque temps les meilleurs chiffres de vente d'appareil mobile.

Le système d'exploitation « Open WebOS » est lui aussi basé sur un noyau Linux. À l'origine, Open WebOS est un OS proposé par l'entreprise HP, qui en août 2011 a décidé d'abandonner sa plateforme mobile et a rendu son système d'exploitation en open source. Depuis, le projet a été repris par la société HP qui a racheté WebOS en février 2013. C'est donc LG qui joue le rôle de commanditaire pour le développement de ce nouveau système d'exploitation mobile.

Firefox OS est le seul des trois systèmes à avoir fait son apparition dans le marché du mobile. Tizen prévoit son lancement en fin d'année 2013 sur un terminal. Open WebOS est lui toujours en cours de développement, même si des versions bêta ont été installées sur une tablette Google Nexus 7.

Le point commun des trois systèmes d'exploitation est leur base Linux/HTML 5. Cela va permettre la compatibilité des différentes applications entre ces OS. Le nombre d'applications disponible pour ces plateformes sera donc bien plus

³ Modèle de smartphone proposé par la marque Samsung

important, car il combinera les trois plates formes à la fois.

Les nouveaux systèmes d'exploitation pour téléphone portable sont entièrement adaptés à un environnement web. Il reste à voir si les interactions avec le serveur internet ne poseront pas de problème lors de l'utilisation. Mais de toute évidence, la qualité du réseau reste important dans du développement web. Avec l'arrivée du réseau 4G, la qualité et le débit du réseau risque d'être fortement amélioré.

À l'heure actuelle, les applications web n'ont pas les capacités à supplanter les applications natives. Elles ne sont qu'une solution de développement offrant des avantages et des inconvénients vis-à-vis de l'application native. Nous avons vu qu'il était tout de même possible de fusionner les deux types d'application (web et native) pour confectionner une application hybride. Cette solution permet en effet d'utiliser son application web sous forme d'application native. Mais le principe de liberté d'internet n'est plus réellement en l'état. Certains stores comme l'App Store d'Apple et le Windows Phone Store effectuent des vérifications sur les applications. Ils peuvent interdire une application, car elle ne répond pas au critère qu'ils ont eux-mêmes défini. Une application web doit être par principe libre de contenu.

Les trois nouveaux systèmes d'exploitation, cité au dessus, offrent au web une nouvelle place dans la mobilité. En effet ici on ne parle plus d'application web par rapport aux applications natives, mais d'application native qui sont développées directement avec les technologies du web.

L'avenir du web se trouve donc plus dans les futurs OS mobiles qui sont tournés vers le développement internet que dans les actuels qui restent plus tournés vers leur choix technologique.

On pourrait donc dire que l'application web ne pourra pas s'imposer par rapport à l'application native, car ce sont deux applications différentes qui ne répondent pas aux mêmes exigences. Le choix reste à faire vis-à-vis de la méthode de création de

l'application mobile. Mais les futurs OS eux proposeront des applications web en terme d'application native.

Il restera donc à voir si les futurs systèmes d'exploitation mobile permettront d'obtenir des applications aussi fluides et rapides que sur nos OS actuels.

Bibliographie :

Introduction :

- Radio : France Info
- Wikipédia,
- ZDNet : Antoine Duvauchelle le 11 Décembre 2012,
- 01Net,
- Christophe Clément medcost.fr,
- <http://philippe-b.forumpro.fr> article "GSM / GPRS / UMTS / EDGE / 3G / WIFI",
- Clubic : article "Téléphonie de troisième génération : EDGE, 3G, etc" Publié par Julien Jay & Alexandre Laurent le 18 Juillet 2006
- <http://fr.slideshare.net/needeo/les-applications-mobiles-sites-mobiles-web-apps>

Partie I chapitre 1.1

- <http://ser-info-02.ec-nantes.fr/users/info3/weblog/2ee24/>
- <http://www.alsacreations.com/tuto/lire/1372-formulaires-html5-nouveaux-types-champs-input.html>
- http://www.w3schools.com/html/html5_intro.asp
- <http://www.w3schools.com/css3/default.asp>
- <http://j-willette.developpez.com/critiques/html/html5-quels-enjeux-pour-la-mobilite-et-le-ria/>

Partie I chapitre 1.2

- <http://www.queness.com/post/6706/9-mobile-framework-to-kick-start-your-mobile-development-career>
- <http://aerilys.fr/blog/?tag=jqtouch>
- <http://validator.w3.org/mobile/>

Partie I chapitre 1.3

- <http://fr.wikipedia.org/wiki/PHP>
- <http://php.net/manual/fr/history.php.php>
- <http://www.scriptol.fr/programmation/langages-populaires.php>
- http://fr.wikipedia.org/wiki/Microsoft_.NET
- <http://www.commentcamarche.net/contents/548-j2ee-java-2-enterprise-edition>
- <http://www.microsoft.com/en-us/server-cloud/windows-server/buy.aspx>
- <http://en.wikipedia.org/wiki/Nodejs>
- <http://blog.vincentbrouillet.com/post/2010/06/27/Quelle-solution-de-technologie-Web-Php,-Java,-.Net,-Rails,-Python>
- <http://www.journaldunet.com/developpeur/outils/linkedin-de-ruby-on-rails-a-node-js-1012.shtml>

Partie 1 chapitre 2.1

- <http://olivierguillet.com/2012/02/panorama-plateformes-mobiles/>
- <https://developer.apple.com/programs/start/ios/>
- <http://www.docdoku.com/blog/2011/04/06/premier-mobile-monday-toulouse-retours-et-presentation-docdoku-en-ligne/>
- <http://fr.wikipedia.org/wiki/Objective-C>
- <http://www.journaldunet.com/developpeur/expert/51048/le-developpement-multi-plateforme---enjeux--promesses-et-realite.shtml>
- <http://lapetitedev.blogspot.fr/2013/04/jquery-mobile.html>
- http://fr.wikipedia.org/wiki/WinDev_Mobile
- <http://fr.wikipedia.org/wiki/Xamarin>

Partie 1 chapitre 2.2

- <http://mobithinking.com/native-or-web-app>
- <http://olivierguillet.com/2012/02/les-differents-types-dapplications-mobiles->

[natives-web-apps-hybrides-flash/](#)

- <http://www.apple.com/webapps/index.html>

Partie 1 chapitre 3

- <http://www.journaldunet.com/web-tech/chiffres-internet>

Partie 1 chapitre 3.1

- <http://www.challenges.fr/high-tech/20121004.CHA1644/l-ocde-quantifie-le-poids-grandissant-d-internet-dans-l-economie-mondiale.html>
- <http://www.atlantico.fr/decryptage/institut-mckinsey-evalue-8000-milliards-dollars-revenus-generes-internet-219858.html>
- <http://www.atlantico.fr/atlantico-light/internet-inquietant-mais-excellent-pour-economie-selon-francais-748230.html>
- <http://www.economie.gouv.fr/files/rapport-mckinsey-company.pdf>

Partie 1 chapitre 3.2

- <http://blog.beule.fr/analyses-dexpert/application-native-vs-web-application-le-duel-fraticide/>
- <http://www.monetilab.fr/6-moyens-rentabiliser-application-mobile-iphone-android-generer-revenus/>
- <http://www.pcinpact.com/breve/61093-applications-app-store-android-market-ovi-store-world.htm>
- <http://www.iphonegen.fr/android-market-croissance-plus-rapide-app-store-actualite-7800.html>
- <http://www.zdnet.fr/actualites/le-windows-phone-market-place-monte-en-puissance-39784035.htm>
- <http://www.canalys.com/newsroom/11-quarterly-growth-downloads-leading-app-stores>

- <http://www.eco-conscient.com/art-714-quels-sont-les-parts-de-marche-des-fabricants-de-smartphone-apple-rim-htc-samsung-nokia.html>

Partit 2 chapitre 1.1

- couverture-reseau.orange.fr
- <http://www.meilleurmobile.com/documents/mobiles/reseaux.jsp>

Partit 2 chapitre 2.1

- <http://www.journaldunet.com/developpeur/tutoriel/theo/060601-theo-difference-langages-compiles-interpretes.shtml>
- <http://arstechnica.com/apple/2011/06/ios-5-brings-nitro-speed-to-home-screen-web-apps/>
- <http://blog.loof.fr/2008/06/du-javascript-compil.html>
- <http://blog.objetdirect.com/wordpress/wp-content/uploads/2011/07/1-S%C3%A9minaire-HTML5.pdf>

Partit 2 chapitre 2.2

- <http://www.cle-3g.fr/difference-3g-4g/>
- <http://www.journaldunet.com/solutions/mobilite/applications-et-services-pros-en-4g.shtml>

Partit 2 chapitre 3.1

- <http://www.presse-citron.net/quel-navigateur-offre-la-meilleure-compatibilite-avec-les-web-apps-html5-infographie>
- <http://lewebmobile.fr/2011/01/le-monde-des-navigateurs-de-smartphones-traduction/>

Partit 2 chapitre 3.2

- <http://graphism.fr/design-mobile-diffrencier-web-application-2>
- http://fr.wikipedia.org/wiki/Interface_graphique
- <http://wdfriday.com/blog/2013/01/jquery-mobile-une-introduction-au-framework-et-principes-de-base/>

Partit 2 chapitre 4

- <http://www.graphiste-freelance-lyon.fr/webapp-application-iphone-android.html>
- <http://tekcollab.imdeo.com/mobile-webapp-vs-application-native/>
- <http://blog.beule.fr/analyses-dexpert/application-native-vs-web-application-le-duel-fraticide/>
- <http://www.directoid.com/top/top-10-perte-performance-site-web.html>
- <http://www.zdnet.fr/actualites/app-store-d-apple-5-ans-de-croissance-et-de-rivalite-avec-google-en-chiffres-39792302.htm>

Partit 3

- <http://www.zdnet.fr/actualites/france-44-de-la-population-equipee-de-smartphones-en-majorite-sous-android-39791046.htm>

Partit 3 chapitre 1

- <http://www.bbc.co.uk/news/technology-23240971>
- <http://www.zdnet.fr/actualites/app-store-d-apple-5-ans-de-croissance-et-de-rivalite-avec-google-en-chiffres-39792302.htm>
- <http://www.monwindowsphone.com/160000-applications-disponibles-sur-le-windows-phone-store-t25523.html>
- <http://www.smartphonefrance.info/actu.asp?ID=8498>

- <http://www.zdnet.fr/actualites/app-store-d-apple-5-ans-de-croissance-et-de-rivalite-avec-google-en-chiffres-39792302.htm>
- <http://www.pcinpact.com/news/81514-le-play-store-aurait-surpasse-app-store-en-nombre-telechargements.htm>

Partit 3 chapitre 3.1

- <http://www.generation-nt.com/gartner-applications-mobiles-hybride-natif-html5-entreprise-actualite-1689762.html>
- <http://www.journaldunet.com/solutions/mobilite/web-apps-html5-ou-apps-natives/web-apps-et-apps-hybrides-les-atouts.shtml>
- <http://blog.brightcove.com/fr/2011/11/la-montee-en-puissance-des-applications-hybrides>

Partit 3 chapitre 3.2

- <http://blog.brightcove.com/fr/2011/11/la-montee-en-puissance-des-applications-hybrides>
- <http://phonegap.com/app/>

Partit 3 chapitre 3.3

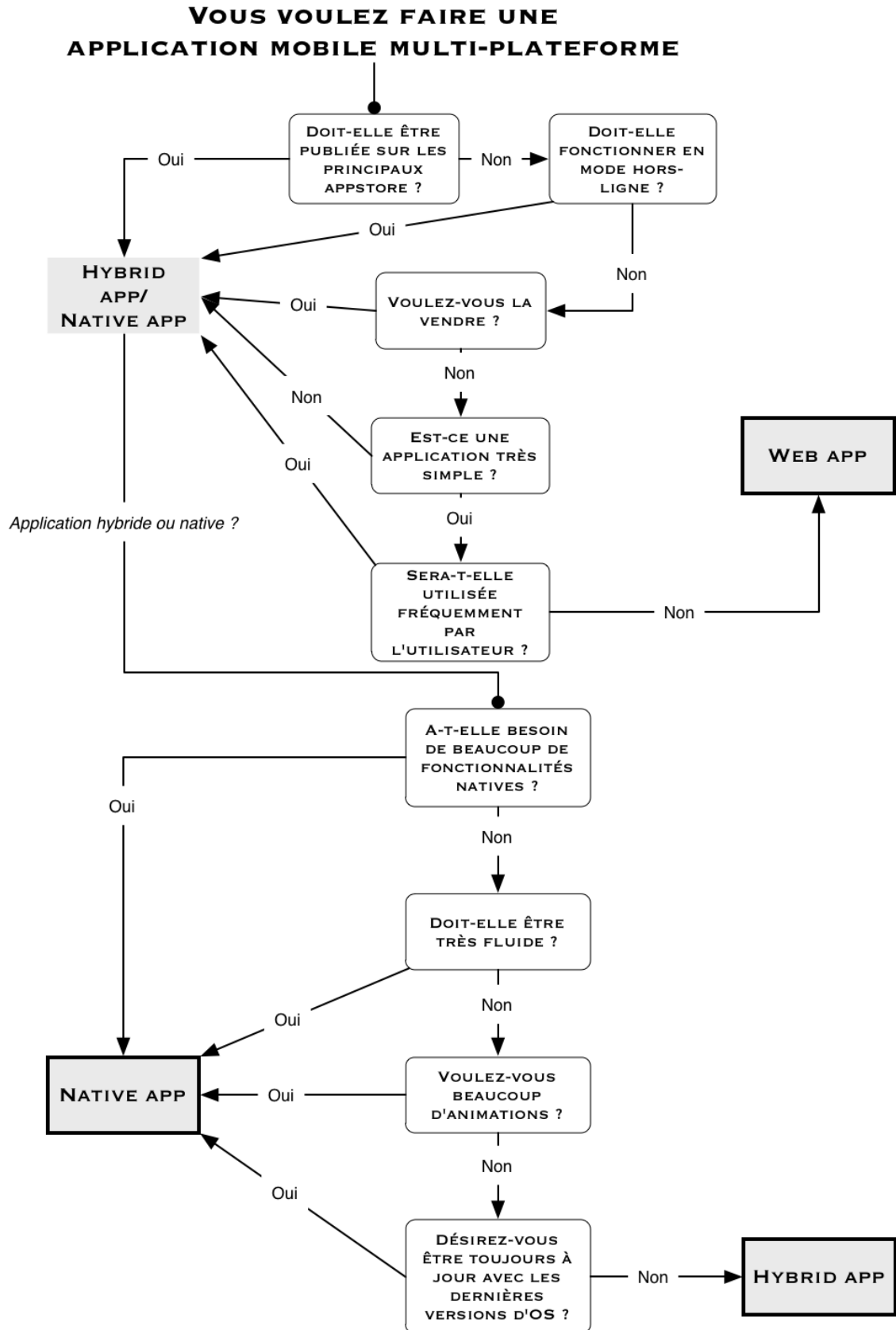
- <http://pro.clubic.com/creation-de-site-web/langage-programmation/actualite-497418-developpement-mobile-web-natif-retours-enjeux.html>
- <http://lecercle.lesechos.fr/entreprises-marches/high-tech-medias/mobilite/221167318/application-mobile-native-web-hybride-6-poin>
- <http://blog.brightcove.com/fr/2011/11/la-montee-en-puissance-des-applications-hybrides>
- <http://www.journaldunet.com/developpeur/expert/51048/le-developpement-multi-plateforme---enjeux--promesses-et-realite.shtml>

Partit 3 chapitre 3.4

- <http://butterflyeffect.fr/blog/application-mobile-native-ou-hybride-il-faut-choisir/>
- <http://www.lesnouveauxmedias.fr/developpement/web-app-mobile-vs-application-native%C2%A0-pas-facile-de-trancher%C2%A0/>
- <http://www.developpeur.orange.tn/news/developpement-mobile%C2%A0-natif-vs-html-vs-hybride>
- <http://www.journaldunet.com/developpeur/expert/51048/le-developpement-multi-plateforme---enjeux--promesses-et-realite.shtml>

Conclusion

- <http://olivierguillet.com/2012/02/panorama-plateformes-mobiles/>
- <http://fr.wikipedia.org/wiki/Tizen>
- http://fr.wikipedia.org/wiki/Firefox_OS
- <http://lecollectif.orange.fr/articles/2013-lannee-des-os-mobiles-alternatifs/>
- <http://www.01net.com/editorial/587571/lg-rachete-webos-pour-l-integrer-dans-ses-tv-connectees/>
- <http://www.itespresso.fr/video-tristan-nitot-mozilla-firefox-os-present-continent-2013-62340.html>



Glossaire

4G : La 4G est la 4^{iem} génération de réseau mobile.

.NET : « nom donné à un ensemble de produits et de technologies informatiques de l'entreprise Microsoft pour rendre des applications facilement portables sur Internet. »
Wikipedia

(méthode) Agile : nouvelle méthode de développement, qui repose sur un ensemble de pratique et de valeur l'équipe, l'application, la collaboration et l'acceptation du changement.

AJAX : architecture qui permet un dialogue, entre le serveur et le client, plus interactivement et plus dynamiquement.

API : « Automate Programmable Industriel est un dispositif électronique programmable destiné à la commande de processus industriels par un traitement séquentiel. Il envoie des ordres vers les *préactionneurs* (partie opérative ou *PO* côté actionneur) à partir de données d'entrées (capteurs) (partie commande ou *PC* côté capteur), de consignes et d'un programme informatique. » *Wikipedia*

Basic : le basic (Beginner's All-purpose Symbolic Instruction Code) est un langage de programmation de haut niveau.

C++ : Le C++ est un langage informatique proche de la machine permettant une gestion de la mémoire à de hauts niveaux.

Cloud computing : nouvelle manière d'accéder à des contenus ou des applications. Les données sont enregistrées sur internet, et les applications peuvent t'être manipulées par plusieurs personnes simultanément.

Cocoa : est une API native d'Apple pour le développement orienté objet sur son système d'exploitation Mac OS X.

Compilation : procéder de transformation d'un langage de programmation en un langage proche de la machine.

Cross-compilation : est un compilateur croisé capable de créer un code exécutable pour différente plate forme avec un seul langage informatique.

Développement orienté objet : méthode de développement qui consiste en la définition et l'interaction de briques logicielles appelées objets.

Eclipse : Logiciel permettant l'écriture de script pour différent langage. Et qui propose plusieurs outils d'aide au développement.

E-économie : économie autour d'internet.

Framework : "ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel"

Wikipedia

Gem : représente une bibliothèque ou un programme Ruby.

IDE : « (Integrated Development Environment) est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné. »

Dicodunet

IHM : Interface Homme Machine, acronyme qui désigne les interactions entre l'utilisateur et l'application graphique.

IntelliJ IDEA : est un IDE qui permet le développement d'une solution Android.

Interprété : un langage dit interprété est soumis à un interpréteur qui analyse le code et exécute l'application.

Java : langage informatique de haut niveau.

JVM : « La **machine virtuelle Java** (anglais *Java virtual machine* abr. JVM) est un appareil informatique fictif qui exécute des programmes compilés sous forme de bytecode Java. » *Wikipedia*

Linux : « est le nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau Linux. » *Wikipedia*

Multitouch : action qui consiste à utiliser plusieurs points sur un écran tactile simultanément.

MVC : « modèle-vue-contrôleur est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. » *Wikipedia*

Open source : Le code open source désigne des travaux libres d'accès. On pourra le redistribuer et le modifier.

Play Store : est une application permettant l'accès à la boutique d'application pour Android

Protocole de communication : « est une spécification de plusieurs règles pour un type de communication particulier. » *Wikipedia*

SDK : est un ensemble d'outils qui aide le développeur à créer une application.

Smartphones : représente la nouvelle génération de téléphone portable assimilé à de petits ordinateurs.

SquirrelFish : est le moteur d'interprétation du code JavaScript des navigateur qui repose sur WebKit.

TIC : la notion TIC (technologies de l'information et de la communication) regroupe « les techniques utilisées dans le traitement et la transmission des informations, principalement de l'informatique, de l'Internet et des télécommunications. » *Wikipedia*

WebKit : « est une bibliothèque logicielle libre permettant aux développeurs d'intégrer facilement un moteur de rendu de pages Web dans leurs logiciels. » *Wikipedia*

Web Service : Permet le dialogue entre un poste client (téléphone, tablette, borne, ordinateur...) et un serveur distant.