

Point sur les solutions de développement d'apps pour les périphériques mobiles

Par Hugues MEUNIER

1. INTRODUCTION

a. Une notion importante : le responsive web design

Nous sommes en train de vivre une nouvelle ère avec l'explosion des périphériques mobiles (smartphones, phablettes, tablettes, montres et objets connectés) et la baisse brutale des ventes de PC depuis 2011. Les entreprises qui souhaitent toucher un maximum de personnes à travers leurs applications se doivent de prendre en compte ces éléments. Aussi, le développement d'applications se complexifie car les développeurs doivent concevoir des applications polymorphes qui s'adaptent quel que soit le format du périphérique utilisé.

Depuis quelques années est apparu le concept de responsive web design (conception de sites web adaptatifs) qui permet de définir les règles de développement d'applications Web qui peuvent être affichées sur différents périphériques (Notebook, TV, Tablette ...). Les règles sont au nombre de trois :

- L'utilisation du « media queries » CSS3
- L'utilisation de grille fluide qui s'auto-dimensionne en fonction de la taille de l'écran
- L'utilisation d'images dimensionnée en unités relatives

Le responsive design a été étendu à tout type d'objets échangés (App html5, email, formulaire d'application ...). Différents Framework très utiles pour développer des applications modernes implémentent le responsive web design.

Il faut bien prendre en compte le fait qu'une application moderne doit s'adapter aux comportements de l'utilisateur qui sont multiples et qui sont opérés avec des périphériques complètement différents. Sinon, le risque est de se couper d'une partie non négligeable de la population visée. Certains parlent d'une révolution similaire à celle de l'Internet à la fin des années 90.

b. Les différents choix pour le développement d'applications mobiles

Il y a foncièrement deux choix structuraux qui s'offrent aux candidats au développement d'applications mobiles :

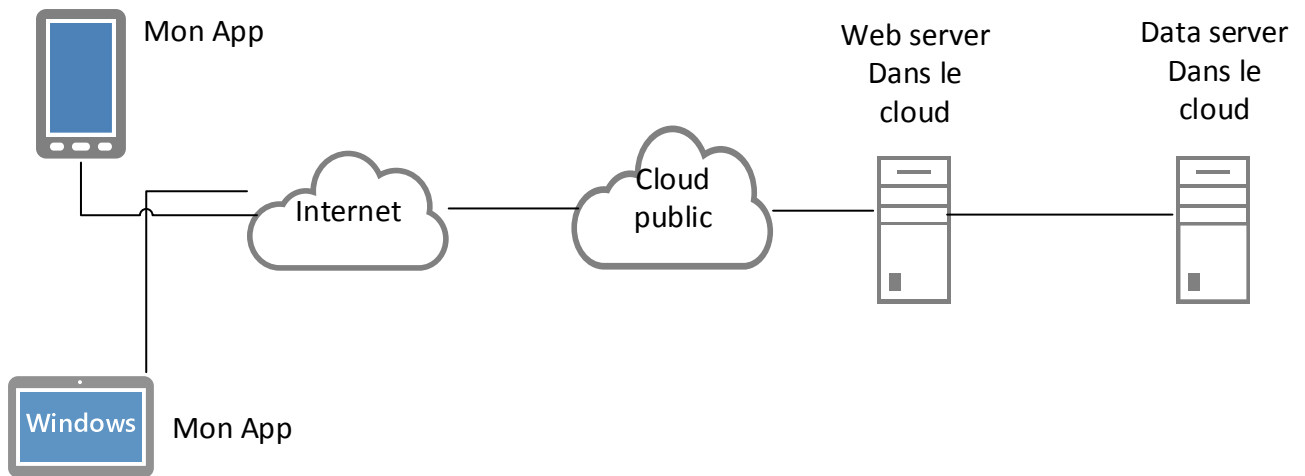
- Développer des applications natives,
- Développer des applications universelles,
- Développer des applications hybrides

2. Quels choix d'architecture pour les applications mobiles

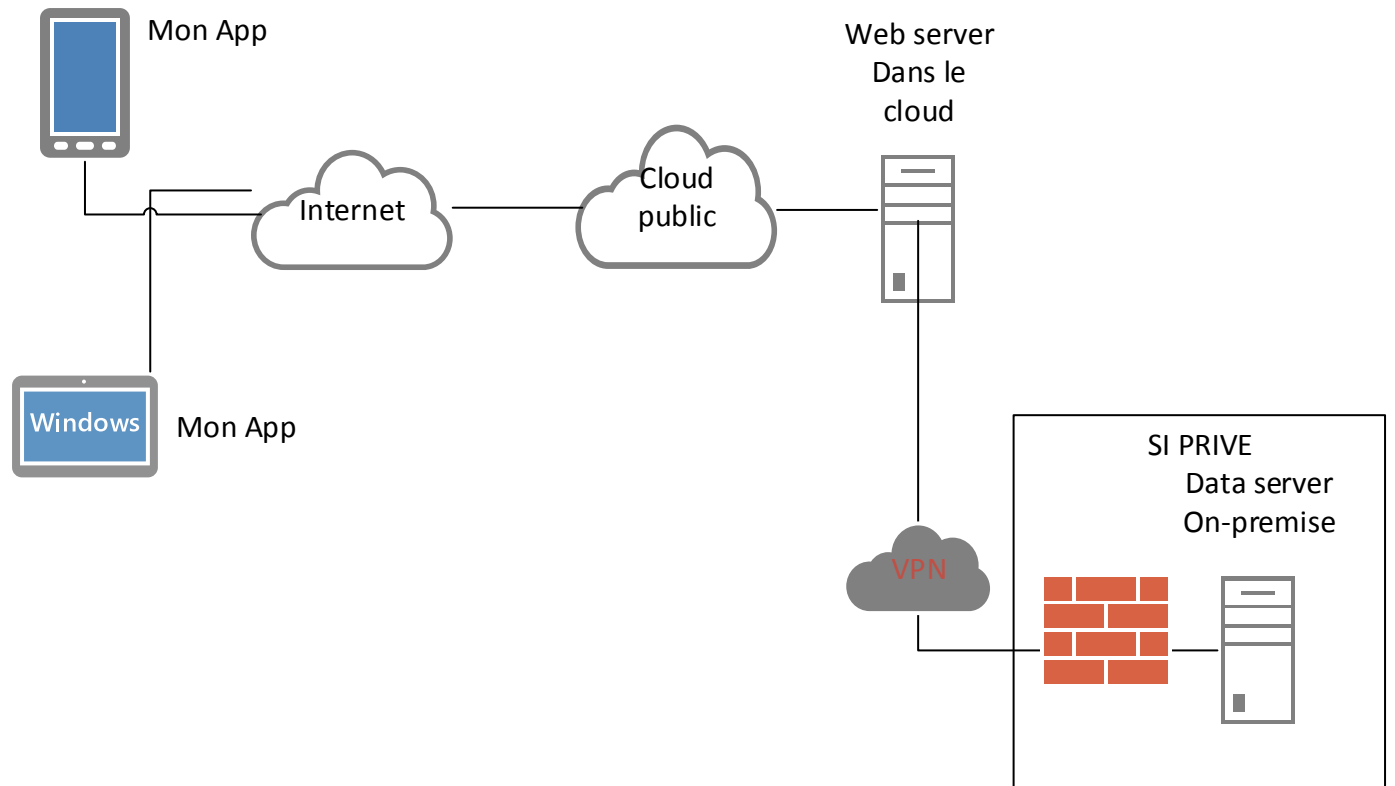
On ne peut pas se lancer directement dans les développements sans un minimum de conception au niveau de l'architecture. Les choix sont nombreux lorsque l'on souhaite connecter ses applications mobiles à des systèmes d'information privés. Même si html5 met à disposition une base de données locale aux périphériques, celle-ci est fortement limitée et ne peut convenir lorsque le développeur souhaite stocker de manière centralisée des informations recueillies avec l'application.

Les offres Cloud offrent de nouvelles solutions en termes de stockage des données applicatives. Ces solutions sont présentées ci-dessous sous forme de schéma.

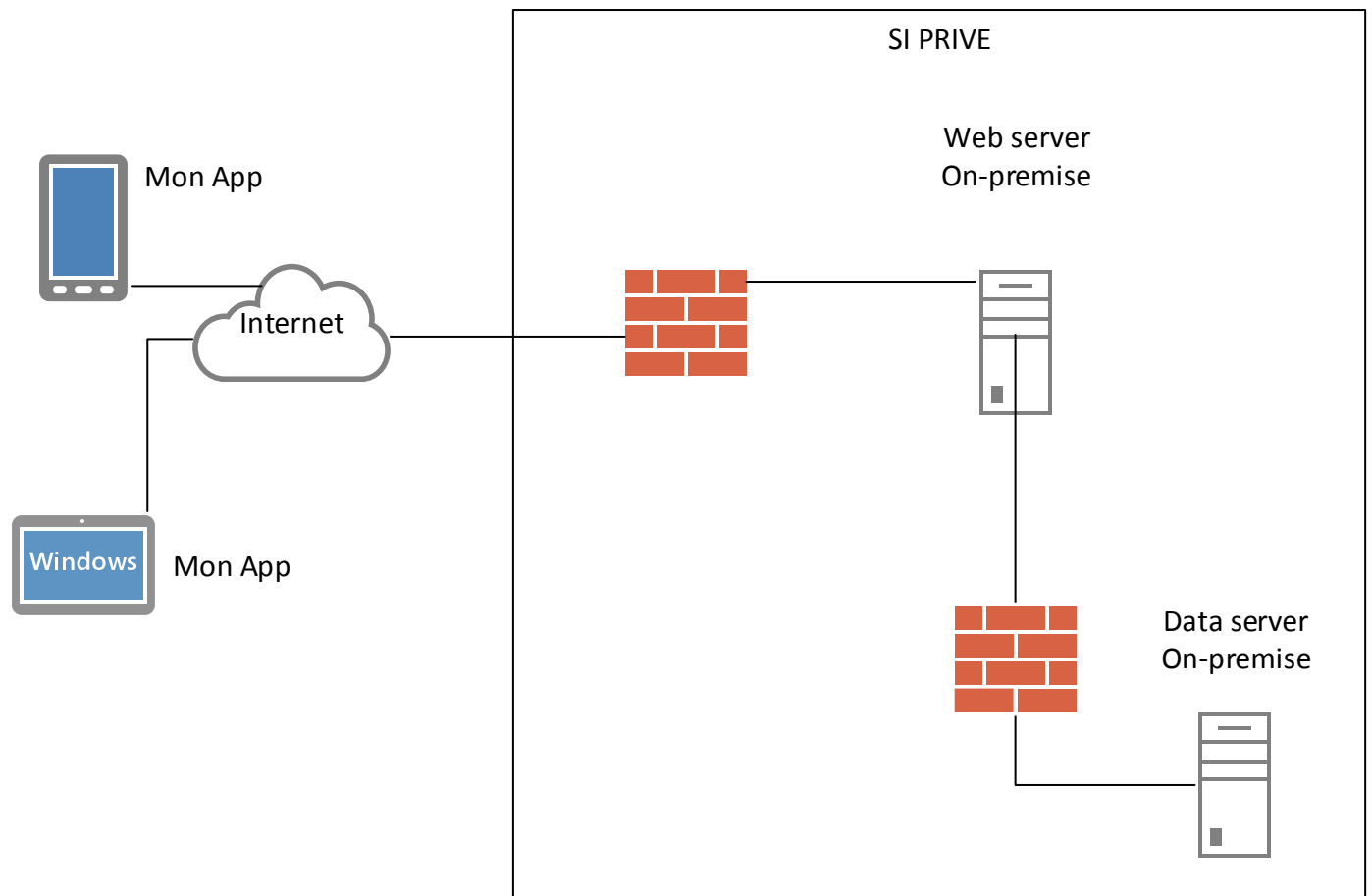
a. Les App in The Cloud : tous les éléments sont dans des Clouds publics



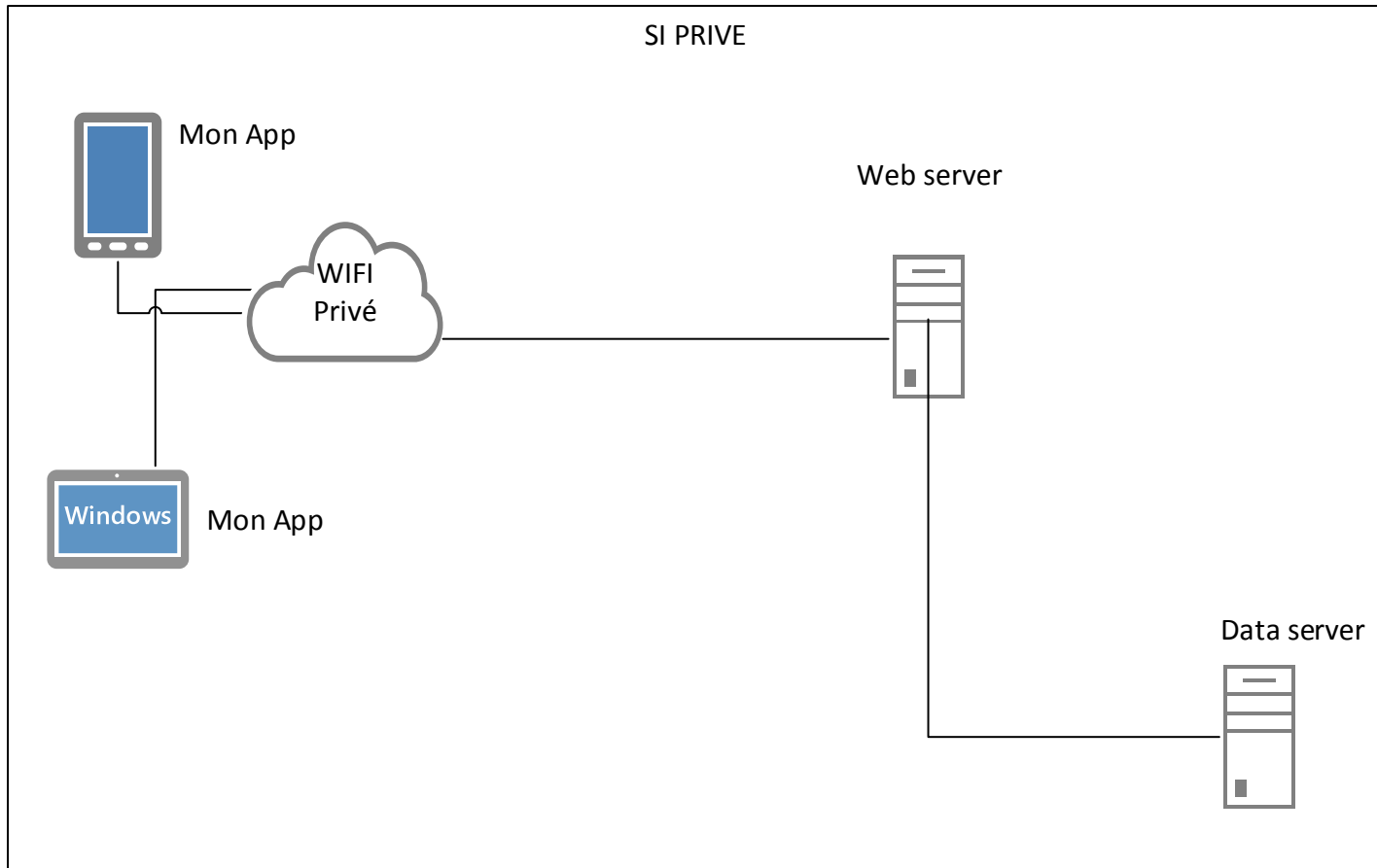
b. Les App hybrides : certains éléments sont conservés on-premise



- c. Les App on-premise : tous les éléments exceptés les périphériques sont conservés dans le SI de l'entreprise



d. Les App privées : tous les éléments sont conservés dans le SI de l'entreprise



3. Les architectures applicatives

Les applications modernes (depuis plusieurs années déjà) implémentent des couches (layered applications) :

- La couche d'accès aux données
- La couche métier
- La couche de service
- La couche de présentation

La couche d'accès aux données est gérée complètement dans les outils de développement que ce soit dans les Framework de développement rapide de type Rail On Ruby, dans le JDK (les EJB) ou dans le monde .NET (Microsoft Entity Framework). Nous sommes dans une démarche code First où la conception de l'application et le code sont les éléments générateurs de la partie données. Les applications mobiles s'adaptent facilement aux approches de type code First.

La couche métier comporte l'ensemble des règles propres à l'entreprise et au métier. Elle est constituée de librairies et de POCO (.NET) ou POJO (Jee).

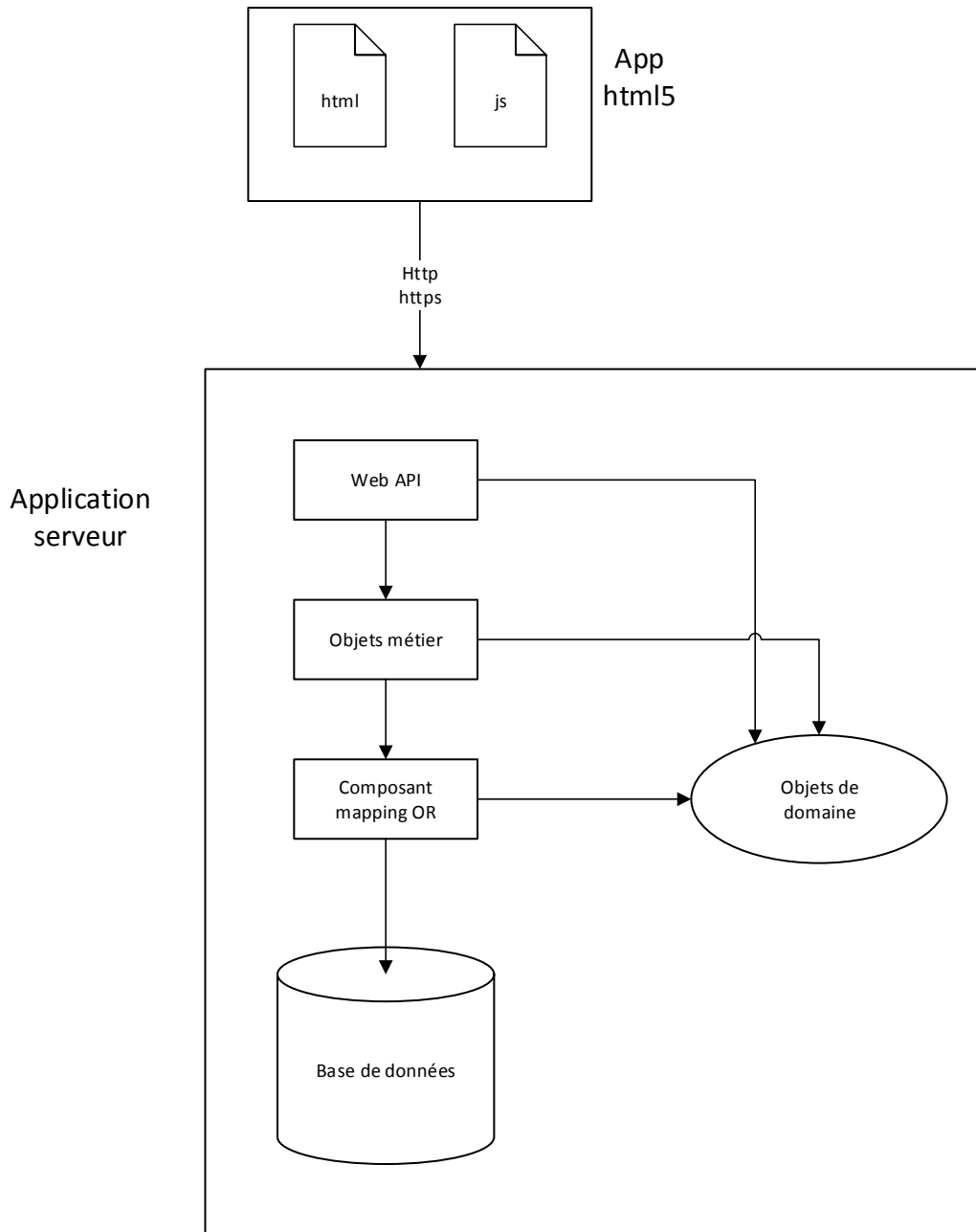
La couche de service comporte les services métier qui sont exposés pour les applications internes ou externes voir les clients de l'entreprise. Le débat entre SOAP et REST reste ouvert même si maintenant la majorité des développements concerne des services RESTFULL. On parle alors d'une Web API qui expose des services Web REST.

La couche de présentation peut être développée suivant différents pattern d'architecture mvc, mvvm. Elle peut être également développée en html5 Javascript en implémentant (ou pas) un pattern d'architecture. Néanmoins, il faudra encore faire un choix structurant pour la couche présentation :

- Le code html et javascript est produit par un serveur WEB et envoyé au client (périphérique mobile) partiellement ou totalement (struts, asp.net, asp.net mvc, node.js, php ...)
- Le code html est directement présent sur le client

L'architecture que je défends est une architecture du deuxième genre :

- La couche présentation est composée d'un ensemble de page html5 et de code javascript qui appelle la couche de service et qui se met à jour partiellement suivant le contexte d'une single page (SPA)
- La couche service est une WEB API



Cette architecture est universelle sur tous les OS mobiles ou non à partir du moment où le périphérique supporte le standard html5. Donc je peux toucher l'ensemble des smartphones, tablettes, notebooks et pc du marché. Néanmoins elle comporte quelques limitations qui peuvent être gérées par l'addition d'un composant dans l'architecture.

4. Le développement d'applications mobiles natives

a. Cas du développement natif Android

Pour développer des App Android, il faut :

- Télécharger la dernière version du jdk disponible sur le site d'Oracle
- Installer Eclipse + Adt packagé ensemble et disponible chez Google
- Télécharge les sdk Android via package-manager
- Configurer les émulateurs

Le langage de développement est java. Le principe du développement Adt est une parfaite séparation des couches notamment entre l'interface graphique (fichiers xml) et le code (des classes java).

- ▶ Vous pouvez consulter l'article <http://www.hmeunier.com/article-initiation-au-developpement-mobile-sur-android-122282322.html> qui vous permettra de débiter dans le développement pour Android.

Pour publier des applications Android dans le Google Play Store, il suffit de :

- Compiler votre projet et générer le package (extension apk)
- Signer le fichier apk avec un certificat généré en local
- Obtenir un compte développeur Google
- Payer une fois 25 \$

Il n'y a pas de procédure de certification pour soumettre une application dans le Google Play. Une vérification peut être faire a posteriori en cas de problème...

b. Cas du développement natif IOS

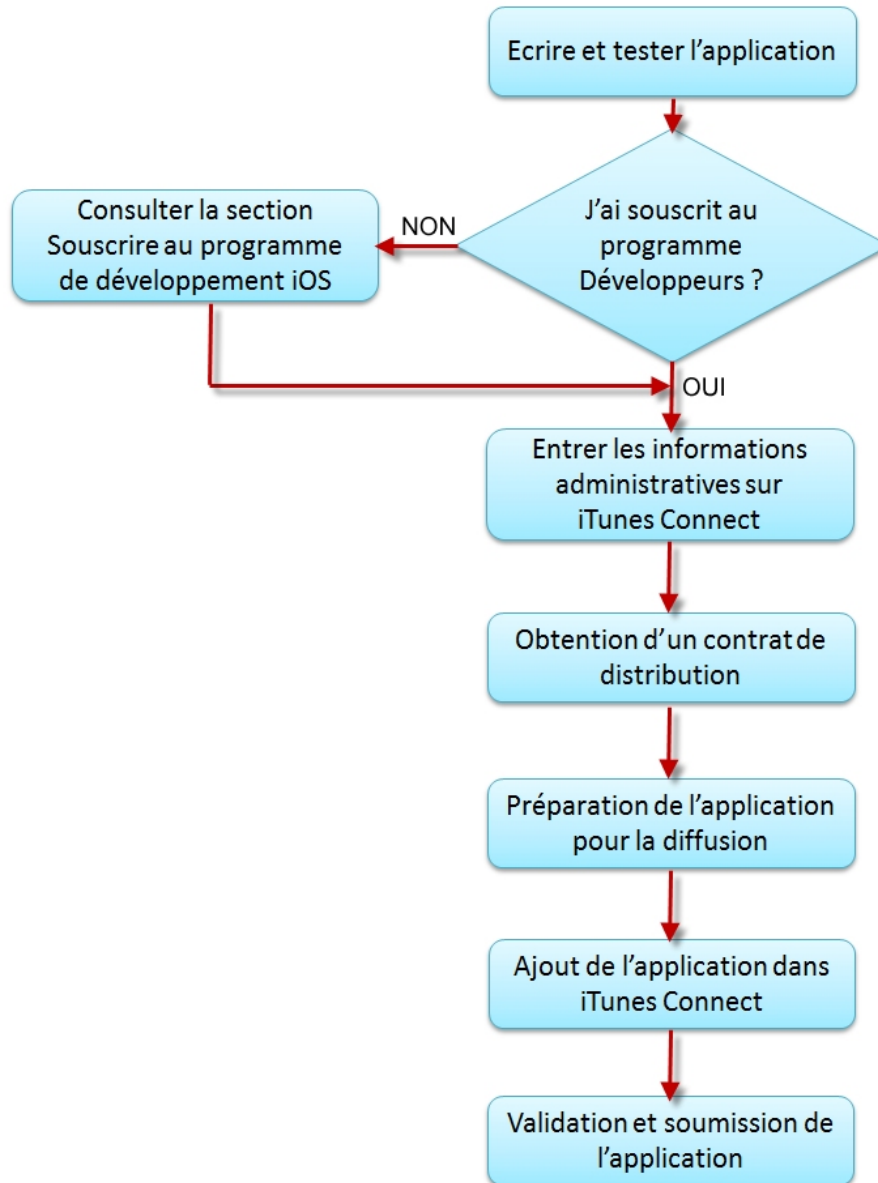
Pour développer des App IOS, il faut :

- Un MAC avec au minimum OS X 10.8 (Mountain Lion)
- Télécharger Xcode depuis le MACOS Store et l'installer
- Installer les IOS SDK (en fonction des versions que l'on souhaite viser)

Le langage de développement est le langage C et plus précisément Objective C qui nécessite de maîtriser la programmation événementielle.

Vous pouvez consulter le site Apple pour les développeurs :
<https://developer.apple.com/>

Le logigramme pour publier une application dans l'Appstore est :



Vous pouvez également consulter le lien sur le site du Zéro ici :
<http://fr.openclassrooms.com/informatique/cours/creez-des-applications-pour-iphone-ipad-et-ipod-touch/proposer-une-application-sur-l-app-store>

c. Cas du développement natif Windows Phone

Pour développer des applications natives Windows Phone, il faut :

- Télécharger et installer Visual Studio 2013 Express (ou bien acheter une version payante ou bien s'abonner à MSDN)
- Télécharger et installer le SDK Windows Phone 8.1
- Choisir le type de développement (html5, c# xaml, asp ,net MVC)
- Compiler et générer un package xap
- Vous avez à disposition avec le SDK Windows Phone, un ensemble d'émulateurs s'exécutant sous hyper-V

- Vous pouvez consulter cet ensemble d'articles qui donnent un aperçu du développement Windows Phone :
- <http://fr.openclassrooms.com/informatique/cours/apprenez-a-developper-pour-windows-phone-en-c/windows-phone-8-2>

Pour publier une application dans le store, il faut :

- Créer un compte live ID
- S'inscrire au programme développeur et payer la somme de 99\$
- Qui permet de développer pour Windows et Windows Phone
- Générer l'app au format xap
- Passer sur le poste l'App certification Kit
- Préparer une icône et des copies d'écran
- Soumettre l'app sur le store
- La certification prend quelques jours et est très stricte

Le site Microsoft consacré au développement Windows Phone contient beaucoup d'informations indispensables : <http://dev.windowsphone.com/>

Depuis les dernières mises à jour des produits Microsoft (Visual Studio 2013 update 2, Windows Phone 8.1, Windows 8.1 update 1), il est possible de développer des applications universelles qui s'exécutent sur des périphériques mobiles (Windows Phone, Windows RT) et sur des PC ou notebooks (Windows 8.1).

5. Le développement d'applications mobiles universelles

Le développement d'applications mobiles universelles est possible en s'appuyant sur le standard html5, le langage javascript et un framework additionnel qui permet de simplifier le développement. A la base, le développement javascript est verbeux et peu maintenable. L'offre de Framework est large mais ma préférence va largement à JQuery Mobile (qui s'appuie sur JQuery).

a. Rappels sur html5

Html5 est le nouveau standard du Web qui permet de développer des applications riches directement exécutables dans le navigateur sans addon ou plugin. Il intègre un espace de stockage local, une base de données locale, la possibilité de lire des vidéos en streaming directement dans les pages et obtenir des infos de géolocalisation et les informations GPS du périphérique sous-jacent.

La plupart des navigateurs récents du marché sont compatibles html5 (IE10 minimum).

Néanmoins, les applications html5 étant faites pour s'exécuter sur tous les périphériques, il n'est pas possible d'accéder à certaines fonctions très spécifiques des smartphones ou tablettes (l'accéléromètre par exemple). Pour cela, il faut utiliser un framework additionnel comme PhoneGAP.

b. Les offres de framework html5

Vous pouvez consulter le lien ci-dessous qui donne un comparatif tout à fait pertinent de ces offres : <http://www.markus-falk.com/mobile-frameworks-comparison-chart/> . On peut citer, parmi ce comparatif :

- ▶ JQuery mobile
- ▶ Kony
- ▶ PhoneJS
- ▶ AppAccelerator
- ▶ Antenna
- ▶ ...

c. Présentation de JQuery Mobile

Jquery mobile est un framework Javascript hyper puissant :

- ▶ Do more, write less
- ▶ Html + javascript + css3 : très verbeux et fastidieux

Il est compatible avec tous les OS mobiles : IOS, Android, Windows Phone.

Il est compatible avec tous les formats de périphériques : Smartphone, tablette ou phablette.

Beaucoup de ressources sont disponibles sur Internet.

Avec JQuery Mobile, on peut développer un site Web mobile mais aussi des Apps.

Consulter le site officiel pour plus de renseignement : <http://www.jquermobile.com>

JQuery Mobile est dans les leaders du quadrant magique du Gartner sur le développement html5.

6. Le développement d'applications mobiles hybrides

Une application mobile hybride est une application développée sur les technologies html5 css javascript avec en plus un framework ou composant spécifique aux périphériques que l'on souhaite viser. Le framework le plus célèbre est Cordova (PhoneGAP). Le principe reste de développer des applications html5 javascript mais celles-ci seront de toute façon packagées dans un fichier spécifique à chaque plateforme (apk, xap ou ipa).

Apache Cordova apporte donc des bibliothèques Javascript qui permettent d'accéder à pratiquement toutes les fonctionnalités des périphériques alors que Html5 n'accède qu'à certaines fonctions seulement.

Pour toutes les informations sur Apache Cordova, c'est ici <http://cordova.apache.org/>

Une petite remarque concerne le fait qu'il faut du coup installer les trois AGL de développement, installer et configurer le framework Cordova sur chacun des AGLs.

Le principal avantage est de ne développer qu'une seule fois le code donc c'est un gain de temps et de ressources spécialisées assez rares sur le marché.

7. Conclusions

Pour les applications qui ne nécessitent pas d'accès aux fonctions évoluées des différents périphériques (accéléromètre, fonctions spécifique des écrans ...), il convient de privilégier le développement universel html5 css javascript en utilisant un framework spécifique comme JQuery Mobile qui permet de s'assurer que les applications s'afficheront correctement sur la plupart des périphériques. Dans ce cas, l'AGL a peu d'importance et vous pouvez choisir votre éditeur préféré. S'il fait de l'auto completion et du debug Javascript ce sera un plus. Si vous optez pour une solution gratuite, Eclipse et Visual Studio 2013 Express peuvent convenir. C'est la méthode de développement à privilégier.

Pour les autres applications, il faut privilégier les solutions hybrides (basées sur Cordova ou un autre framework gratuit ou payant (KONI)).

Le développement d'applications natives doit rester marginal.