



**INGENIEUR EN SCIENCES INFORMATIQUES**

**RAPPORT D'ETUDE TECHNOLOGIQUE SUR LES SOLUTIONS WEB**

**CROSS PLATFORM MOBILE DEVELOPMENT**

**(Phonegap, RhoMobile)**

**Student : Jiachen NIE**

**Parcours: IHM**

**Subject : Adaptation des Interfaces à l'Environnement**

**Date : 11/11/2014**

## I. Présentation Des Technologies

### Phonegap

Phonegap est un framework open-source développé par Adobe Systems et basé sur Apache Cordova qui est le moteur qui le fait tourner. Il fournit des APIs JavaScript aux navigateurs Web standards permettant d'appeler des fonctionnalités natives non disponibles autrement. Phonegap permet aux développeurs de créer des applications mobiles pour différentes plateformes avec les technologies web comme HTML, CSS et JavaScript. Grâce au composant permettant d'inclure une vue Web dans une application, et la technologie Web design est plate-forme indépendante, Phonegap arrive à rendre les applications disponibles dans chaque SDK. Son principe général «Un seul code, plusieurs plateformes cibles» est le meilleur résumé de ses fonctionnalités. Concrètement, Phonegap est un outil de création d'un projet type pour chaque OS. Mais en pratique, il crée une WebView qui exécute du code Web.

Depuis la version 3.0, les développeurs peuvent utiliser deux workflows de base pour créer une application mobile : Cross-plateform workflow et Plateform-centered workflow. Le premier doit être utilisé quand le but est faire l'application fonctionner sur autant de systèmes d'exploitations mobiles différents que possible avec peu de besoin de développement spécifique à une certaine plateforme. Au contraire, les développeurs peuvent se servir de ce dernier dans le but de se concentrer sur le développement sur une seule plateforme.

### RhoMobile

Similaire à PhoneGap, RhoMobile Suite propose un framework open-source Rhodes développé par Motorola permettant aux développeurs de créer des applications pour les différents systèmes d'exploitation avec les technologies web. Rhodes rend possible de développer des applications mobiles sous l'architecture MVC (Modèle-Vue-Contrôleur) dans le but de séparer le design visuel avec HTML et le design des business avec Ruby. Le code est compilé en code Ruby et puis interprété selon la plateforme spécifique par la machine virtuelle Ruby. Ses composants sont listés au-dessous :

- RhoMobile : Il est le conteneur des applications mobiles qui fonctionne comme une véritable application mais permet aux développeurs d'écrire facilement le code cross-plateforme en utilisant les technologies web. En plus de supporter les langages web standards comme JavaScript, HTML, CSS et Ruby, RhoMobile fournit des accès au dispositif, au système et au framework à travers ses bibliothèques APIs nommées Rhodes et RhoElements.

- RhoStudio : C'est un plugin Eclipse très puissant permettant de développer, tester et déboguer des applications RhoMobile. Il comprend RhoSimulator, qui est un simulateur du dispositif et de la plateforme.

- RhoConnect : C'est une application serveur qui joue le rôle d'un pont transparent pour que les données d'entreprise existent à la fois sur l'arrière-plan et le dispositif mobile. Avec RhoConnect, les utilisateurs peuvent connecter simultanément leur application à de nombreux sources de données tout aussi facilement que se connecter à une seule source de données, y compris le CRM système et le ERP système, les web services , les bases de données , etc.

- Hosted Services : Y compris la construction à distance (RhoHub) et le management des applications (RhoGallery).

## II. Contexte d'Usage

### PhoneGap

• **Plateforme** : Apple iOS, Google Android, BlackBerry, LG webOS, Microsoft Windows Mobile Phone, Nokia Symbian OS, Tizen, Bada, Firefox OS et Ubuntu Touch, Palm (couvre le plus de plateformes parmi toutes les technologies similaires).

• **Environnement** : Les développeurs veulent créer des applications mobiles avec les technologies existantes qu'ils connaissent bien sans le besoin d'apprendre à utiliser une certaine technologie d'une plateforme.

• **Utilisateur** : Les développeurs qui maîtrisent bien HTML, JavaScript, CSS.

### RhoMobile

• **Plateforme** : Apple iOS, Google Android, Microsoft Windows Mobile, RIM, BlackBerry, Window Phone 7

• **Environnement** : Pareil que précédent.

• **Utilisateur** : Les développeurs qui maîtrisent bien Ruby et HTML. Globalement, les utilisateurs ciblés sont les entreprises et les développeurs B2B.

## III. Moment d'utilisation

Ces deux technologies peuvent être utilisées quand un développeur a envie d'immigrer une application vers les autres plateformes, mais sans intention de la réimplémenter avec chaque langage et toolset spécifique pour des plateformes différents. Globalement, ces deux technologies sont tous pour faciliter la vie du développeur puisque ils permettent de diminuer le coût et le temps du développement et de réutiliser un maximum de code pour chaque cible. Donc c'est **lors de la conception**.

Pourtant, si un développeur veut essayer d'intégrer les composants natifs d'une application dans un WebView qui permet d'accéder les APIs des dispositifs, ou le développeur veut développer une interface du plugin entre les composants natifs et ceux du WebView, PhoneGap est un bon choix.

Par contre, si un des critères est la connexion avec la base de données ou la création des applications internes dans l'entreprise, il vaut mieux choisir RhoMobile grâce à son composant RhoConnect qui permet de bien faire la synchronisation des données.

## IV. Solution et modèle sous-jacent

### Phonegap

Des applications Phonegap comptent sur un fichier en commun config.xml qui définit des composants disponibles (caméra, accéléromètre, géolocalisation, ...), le nom de l'application, le splashscreen, et les paramètres globaux. Ce fichier est différent suivant les cibles. **Une application est implémentée comme une page web** (nommée en index.html par défaut) qui fait les références sur les fichiers CSS, des fichiers JavaScript, des images, des médias ou des autres ressources. A part

des third-party plugins offerts, Phonegap fournit aussi les possibilités d'inclure des bibliothèques JavaScript spécifiques et d'écrire des plugins natifs.

Le modèle sous-jacent de Phonegap, comme c'est obligatoire d'installer les SDKs ciblés avant de lancer Phonegap, cela signifie que Phonegap doit jouer un rôle de générer le code natif d'une plateforme. Donc je pense que le programme généré pour une certaine plateforme est MO (données utilisateurs), **Phonegap est en effet un modèle (M1)** parce que HTML et JS fournissent une description des applications, y compris le nom de l'application, les fonctionnalités natives que l'on veut appeler, les plugins que l'on veut utiliser, etc.

## RhoMobile

De ma part, la raison pour laquelle RhoMobile permet de faire le développement cross-plateforme est similaire à Java -- la machine virtuelle. **Grâce à la machine virtuelle Ruby, le code est compilé en code Ruby et puis interprété selon la plateforme spécifique.** Par conséquent, RhoMobile n'exige pas l'installation des SDKs ciblés.

Pareil que Phonegap, à mon avis, **RhoMobile peut aussi être considéré comme un modèle.** Il fournit le code Ruby permettant de décrire les applications. Et le code interprété par la machine virtuelle est MO (données utilisateurs).

## V. Illustration sur un exemple

Pour faire un premier essai du Phonegap, j'ai essayé avec un exemple sur le site Onsen UI. Après avoir associé la plateforme Android à Phonegap, je me suis arrivée à l'ouvrir dans AndroidStudio. Cela a montré que Phonegap permet de transformer le code HTML avec JavaScript en le code Android.

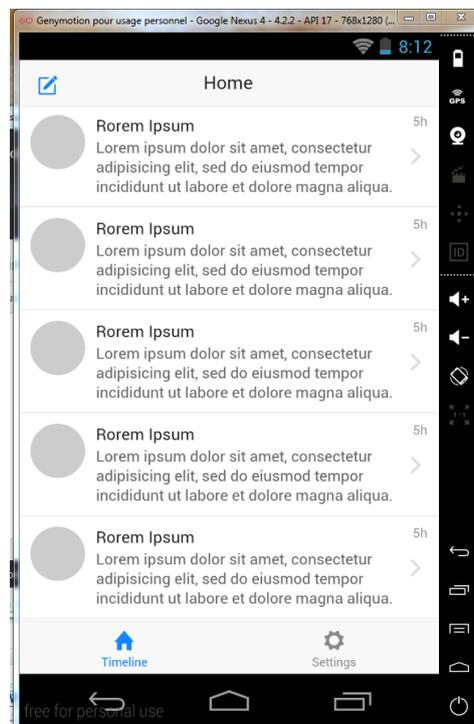


Figure 1. Résultat d'exécution d'un exemple OsenUI sur le simulateur GenyMotion

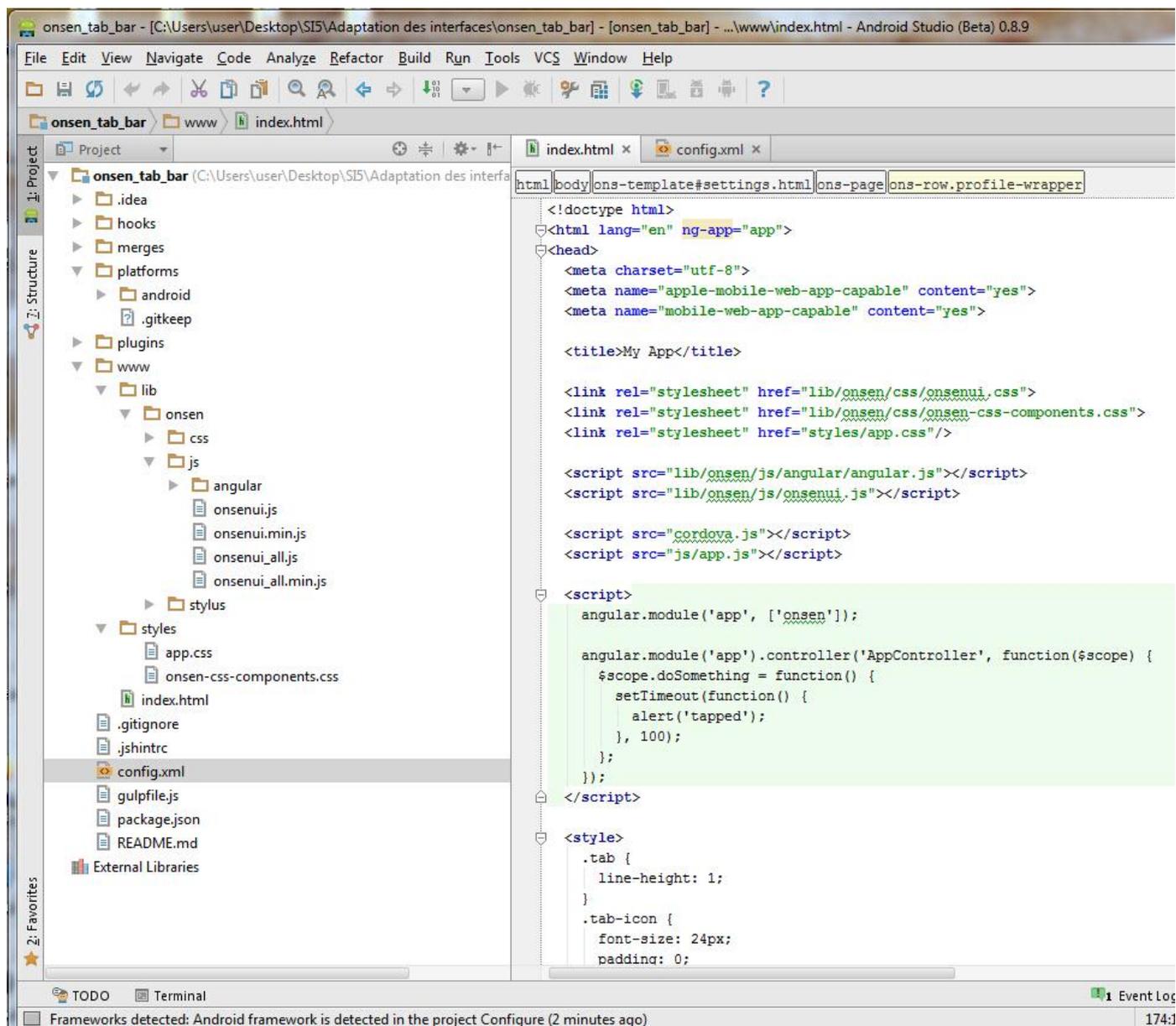


Figure 2. Répertoire du code de PhoneGap présenté dans AndroidStudio

## VI. Mes opinions

Globalement, les grandes **qualités** que les technologies cross-plateforme disposent sont la réduction du temps et du coût de développement, la maintenabilité sur différentes plateformes et la simplification d'un seul langage pour différents plateformes. Par contre, les testes sur les différentes plateformes posent toujours un **problème**. Comme c'est du JavaScript, qui est un langage explicatif, il est difficile de déboguer le code.

Concrètement, comme ces technologies sont les technologies Web, cela a réduit l'exigence sur l'expertise des développeurs. En plus, la vitesse d'exécution des programmes avec ces technologies doit être théoriquement plus lente que les programmes natifs d'une plateforme parce qu'il faut du temps pour la génération ou l'interprétation du code (même si je n'arrive pas à l'observer avec mon petit exemple de code).