

Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile. Comparatif des solutions de développement multiplateforme mobile.

LIVRE  
BLANC

## Comparatif des solutions de développement multiplateforme mobile

**Benoit Desnos**

Responsable de l'offre Mobile

[Benoit.desnoss@netapsys.fr](mailto:Benoit.desnoss@netapsys.fr)

# Sommaire

---

<b>1. Introduction</b>	<b>04</b>
Le contexte	04
Les technologies multiplateformes	06
Présentation du document	08
<b>2. Les solutions dominantes</b>	<b>09</b>
PhoneGap	09
Caractéristiques de la solution	10
Appcelerator Titanium	11
Caractéristiques de la solution	12
Rhomobile	14
Caractéristiques de la solution	15
Adobe Flex	17
Caractéristiques de la solution	18
<b>3. Les challengers</b>	<b>19</b>
Mobl	19
Caractéristiques de la solution	20
OpenPlug	21
Caractéristiques de la solution	22
Corona	23
Caractéristiques de la solution	24
MoSync	25
Caractéristiques de la solution	26
Appmobi XDK	27
Caractéristiques de la solution	28
Quickconnectfamily	29
Caractéristiques de la solution	30
WorkLight	31
Caractéristiques de la solution	32
<b>4. Tableaux comparatifs</b>	<b>33</b>
Présentation des tableaux	33
Systèmes d'exploitation supportés par les solutions	33
Fonctionnalités supportées par les solutions	34

5. Conclusion	35
6. Annexe 1 : Terminologie	37
Définition des caractéristiques étudiées	37
Description de l'environnement des systèmes d'exploitation	38
Définition des fonctionnalités	39
Définitions diverses	40
7. Annexe 2 : Méthodologie	41
8. À propos de Netapsys	42

# 1.

## Introduction

---

### Le contexte

---

La barre est passée : depuis le premier semestre 2011 il se vend plus de smartphones que de mobiles standards (IDC aout 2011). A l'instar des tablettes ([zdnet sept 2011](#)), ce marché est en pleine explosion. Mais il est aussi en pleine mutation. Un cap a été franchi en 2007 avec la sortie de l'iPhone par Apple. Le marché balbutiant des smartphones a trouvé avec l'iPhone une icône qui a déclenché un engouement immédiat pour ce nouveau type d'appareils. D'autres acteurs ont voulu marcher dans les traces d'Apple, mais l'iPhone a brusquement placé la barre nettement plus haut qu'avant. Les acteurs du marché ont été pris de court. Il faudra attendre plusieurs années et que Google et son Android atteignent une certaine maturité pour que la concurrence puisse réagir et dépasser Apple. Aujourd'hui les ventes d'appareils Android ont largement dépassé celles de l'iPhone aux Etats Unis, et légèrement dépassé en France.

Apple n'a pas fait que fabriquer un nouvel appareil. Apple a inventé un nouveau marché : celui des applications. Ce marché existait avant, mais Apple l'a tellement développé et simplifié qu'en comparaison on peut dire qu'il n'existait pratiquement pas avant. Apple a fait cavalier seul pendant quelques années sur ce marché mais est aujourd'hui talonné par Android qui propose pratiquement autant d'applications sur son Market qu'Apple dans son Store. Cependant Apple continue à représenter 75% de ce marché des applications en termes de revenu.

Le marché des téléphones et des applications est donc fragmenté. Si on souhaite diffuser une application au plus grand nombre il est devenu nécessaire de déployer celle-ci sur plusieurs systèmes d'exploitation. On parle bien de systèmes d'exploitation et pas d'appareils. C'est en effet lui qui exécute les applications plus que le matériel qui n'est que le support et fournit les interfaces et la puissance permettant aux applications de s'exécuter. On parle de plateforme quand on veut parler de l'ensemble des appareils qui partagent un même système d'exploitation et sont donc très compatibles entre eux en ce qui concerne les applications.

Un autre type d'appareils est pris en considération ici : ce sont les tablettes. Apparues en 2001 sous le nom de « Tablet PC », elles ne connaîtront de succès qu'à partir de 2009 avec la sortie de l'iPad d'Apple. Ce nouvel appareil a ouvert un nouveau marché en marge de celui des téléphones mobiles tout en utilisant des technologies pratiquement identiques. Les autres constructeurs ont cette fois réagi avec plus de vivacité mais sans grand succès pour l'instant, l'iPad devant continuer à dominer seul le marché des tablettes jusqu'en 2015 ([source : gartner](#)).

Un autre point important sur le développement d'applications mobiles concerne la technologie de développement. Chaque plateforme nécessite en effet des outils de développement différents. Si on veut déployer une application sur différentes plateformes, il semble donc nécessaire de développer autant de fois ces applications qu'on veut adresser de plateformes. Mais il existe des solutions pour permettre de ne développer qu'une fois l'application puis de la déployer sur d'autres plateformes. L'objectif de ce livre blanc est de présenter ces solutions et de les comparer entre elles, chacune ayant des avantages et des inconvénients.

## Les technologies multiplateformes \_\_\_\_\_

Pour réaliser l'ambitieux projet que l'on peut résumer par le motto de Sun Microsystems « write once, run anywhere », les solutions multiplateformes ont mis en place trois approches se mélangeant parfois les unes aux autres.

La première approche, qui semble la plus naturelle, consiste à utiliser la technologie native pour créer l'application. La solution va donc présenter à l'utilisateur les objets graphiques ou « widgets » standards de la plateforme pour chacune des plateformes supportées. Elle va par exemple utiliser une `UITableView` sur iOS pour présenter une liste et une `List View` sur Android. Il faut donc qu'elle s'adapte à chacun des widgets présents sur les plateformes supportées, ce qui représente une charge de travail importante et limite souvent la solution à un nombre limité de plateformes et de widgets supportés. Et cela conduit soit à recourir au principe du commun dénominateur qui réduit les fonctionnalités, soit à une adaptation manuelle par le programmeur qui retire de la transparence au portage multiplateforme. En contrepartie les performances sont optimales puisqu'on utilise directement les technologies natives du système d'exploitation.

La seconde approche, souvent utilisée en combinaison avec la première, consiste à dessiner les composants graphiques. Elle utilise donc des bibliothèques « bas niveau » telles qu'OpenGL. Comme les widgets sont dessinés, ils sont généralement différents des composants natifs. Les performances sont dégradées par rapport aux composants natifs, mais on gagne en flexibilité car l'outil peut plus facilement offrir un comportement cohérent sur les différentes plateformes. La liberté induite par ce mode de création permet à ces solutions de créer non seulement

des applications d'entreprise habituelles mais également, et souvent même principalement, des jeux.

La troisième approche se base sur le navigateur web. En effet celui-ci s'occupe déjà de régler la problématique du multiplateforme. Et on n'est plus ici dans les problématiques de compatibilité des navigateurs car la majorité des appareils mobiles utilise désormais le même navigateur, WebKit, et tous convergent vers le même standard, HTML 5. Plusieurs solutions proposent de fabriquer une application en « embarquant » un navigateur dans celle-ci de sorte que l'utilisateur lance une application dont l'affichage est effectivement traité par un navigateur web. Le premier inconvénient de cette solution est qu'un navigateur a un accès réduit aux ressources matérielles, et c'est pourquoi les solutions ayant adopté cette approche implémentent des accès spécifiques aux ressources natives des appareils telles que l'accéléromètre ou la géolocalisation. Nous qualifions de telles applications d'hybrides. On notera également qu'une application hybride sera généralement moins performante qu'une application native, le moteur d'affichage étant gourmand et les ressources nécessaires plus importantes.

Il arrive que des solutions proposent des approches mixtes, avec certains composants natifs et d'autres dessinés. Et toutes les solutions proposent d'intégrer aussi un navigateur web et ainsi de devenir une solution hybride ne serait-ce que partiellement.

## Présentation du document \_\_\_\_\_

La liste des solutions est assez longue avec une dizaine de solutions du marché retenues. Mais toutes ces solutions ne partagent pas une même diffusion ou une même reconnaissance du public. Les solutions les moins diffusées le sont souvent pour de bonnes raisons, comme une documentation ou une communauté inexistantes. C'est pourquoi nous avons classé les solutions en deux catégories : les dominantes et les challengers.

Pour chaque solution nous avons indiqué :

- Un descriptif de la solution,
- Ses avantages et ses inconvénients,
- Un tableau décrivant les caractéristiques de la solution.

Les différentes solutions ayant des portabilités très variables et des couvertures fonctionnelles très différentes les unes des autres, nous avons mis en place deux tableaux de comparaison :

- Un premier tableau indiquant la disponibilité de la solution pour chaque plateforme,
- Un second tableau indiquant les fonctionnalités de la solution.

Enfin nous concluons en vous donnant notre analyse sur les solutions présentées.

Vous trouverez à la fin du document deux annexes décrivant la méthodologie et la terminologie employées dans ces tableaux et dans le reste du document.



## 2.

# Les solutions dominantes



Phonegap est une solution hybride. C'est une bibliothèque permettant d'accéder depuis le JavaScript d'une page web à de nombreuses ressources matérielles de l'appareil. Le système de callback sur lequel il fonctionne pose des problèmes lors de son intégration avec des bibliothèques JavaScript tierces. Phonegap ne gère pas l'interface utilisateur qui est entièrement laissée à la charge de l'utilisateur et du navigateur embarqué. La présentation se fait donc à l'aide de CSS, de bibliothèques JavaScript tierces, ou des langages spécialisés tels que Mobl.



Une petite bibliothèque, des accès à de nombreuses ressources matérielles, une solution extensible, une communauté active, et la gratuité.

On aime



Windows phone 7 n'est pas supporté pour le moment, la conception avec callback parfois difficile à accorder avec d'autres librairies JavaScript.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Bibliothèque	
Type de déploiement	Hybride	
Gestion de L'IHM	Non	L'ergonomie est laissée à la charge de l'utilisateur. En général on choisira une bibliothèque JavaScript pour construire l'interface utilisateur
Type de mise en page	Coordonnées et pourcentage	On peut utiliser les CSS pour construire l'interface. Cependant on utilisera généralement la mise en page proposée par la solution que l'on a retenue pour créer l'interface graphique
Langage	HTML 5, CSS 3, JavaScript	
Vitesse de développement	Rapide / Moyenne	Les fonctions sont souvent basées sur un système de callback pouvant produire des problèmes de compatibilité avec certaines bibliothèques tierces. Par exemple créer un simple objet Proxy de base de données avec Sencha Touch et le faire communiquer avec l'API de Phonegap afin d'accéder à une base de données n'est pas trivial à réaliser
Courbe d'apprentissage	Rapide	Phonegap est une petite bibliothèque qui utilise toujours les mêmes mécanismes
EDI	N'importe quel EDI supportant les technologies du développement web (HTML, CSS et JavaScript)	
Outil graphique pour l'IHM	Aucun	
Débogueur	Débogueurs JavaScript (weinre, safari, chrome, bugzilla, ...)	
Documentation	<a href="#">Référence de l'API</a> , <a href="#">wiki</a>	
Outils de support	<a href="#">IRC</a> , <a href="#">Google group</a> et <a href="#">blog</a>	
Popularité	Très grande	
Outils de formation	Webinars , forum privé (avec les plans Basic et Starter) et support par email (Starter)	
Prix	La bibliothèque est gratuite (community). Les plans de support sont : Basic 249\$/y (181€), Starter 999\$/y (729€)	
Licence	<a href="#">Licence BSD modifiée</a> ou <a href="#">Licence MIT</a>	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	« Phonegap Build »	
Extensibilité	Oui à travers des <a href="#">plugins</a>	Il est possible d'étendre indéfiniment Phonegap à travers l'écriture de plugins. Cependant il faut écrire un plugin par plateforme à supporter



Titanium est une API et une plateforme JavaScript éditées par « Appcelerator ». Il permet de créer des applications natives et offre une API supportant une grande quantité de ressources matérielles. Appcelerator fournit également un IDE basé sur Eclipse, « Titanium studio », qui permet de compiler et tester son code à la volée. Le codage est rapide et il suffit de peu de code pour développer une application.

Cependant Titanium souffre de deux inconvénients majeurs. Le premier est qu'il faut une connexion Internet permanente pour pouvoir utiliser Titanium Studio, cela signifie que si on perd sa connexion ou si le site de Titanium est en maintenance, alors on ne peut plus travailler. Le second inconvénient est la documentation : il y a beaucoup d'outils mais l'information y est divisée et aucun d'eux n'est vraiment exhaustif, on peut passer de précieuses minutes (voire heures) à chercher une information. A noter également que quelques incohérences graphiques se produisent parfois, l'équipe d'Appcelerator travaille actuellement dessus. Dans l'ensemble le plus gros problème reste la documentation, car on gagne vraiment du temps à développer avec Titanium, malheureusement on perd ce temps à chercher comment résoudre tel ou tel problème.



L'application native, l'aspect natif & performances, l'accès aux ressources matérielles, la vitesse de développement, et l'extensibilité.

On aime



La mauvaise documentation, le manque de ressources d'apprentissage, l'IDE réclamant une connexion Internet permanente, pas de solution d'encapsulation du modèle de données et parfois de nombreuses fuites de mémoire apparaissent.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Runtime, Framework	
Type de déploiement	Natif	
Gestion de L'IHM	Oui	Oui
Type de mise en page	Coordonnées	
Langage	JavaScript	
Vitesse de développement	Rapide / Moyenne	
Courbe d'apprentissage	Moyenne / Longue	Malgré l'utilisation du langage JavaScript très répandu, la courbe d'apprentissage s'allonge considérablement à cause d'une mauvaise documentation. L'application de référence "Kitchen Sink" par exemple est construite sur un design non recommandé. Alors que <a href="#">Tweetanium</a> , également écrit par l'équipe de Titanium, propose de bien meilleures pratiques mais n'est pas mis en avant
EDI	Titanium Studio (basé sur Eclipse)	C'est une bonne intégration qui contient les simulateurs, la coloration des éléments de l'API et un débogueur. Cependant l'IDE a besoin d'une connexion permanente avec le serveur d'Appcelerator pour fonctionner
Outil graphique pour l'IHM	Aucun	
Débogueur	Titanium Studio	La couche d'abstraction entre l'API JavaScript et le code natif final étant très grande, il est difficile à la vue des informations délivrées par le débogueur de corriger de manière productive les exceptions générées par ce code (Objective C par exemple). Le JavaScript par contre se débogue très bien
Documentation	<a href="#">Q&amp;A</a> , <a href="#">référence de l'API</a> , wiki, <a href="#">blog</a> et <a href="#">screencasts</a>	Trop d'outils incomplets. Certaines fonctionnalités sont définies dans certains outils et pas d'autres ce qui rend difficiles les recherches : il faut chercher à travers chaque outil pour avoir une information, quand elle existe
Outils de support	<a href="#">Q&amp;A</a> , <a href="#">devlinks</a> et <a href="#">blog</a>	Le système de question/réponse basé sur un modèle similaire à « Stackoverflow » est intéressant et performant. La mise en relation avec d'autres développeurs Titanium est une excellente chose
Popularité	Très grande	
Outils de formation	Classes en anglais, par exemple : « Building Native Mobile Apps » dure deux jours et coûte £1,250.00 (1424€). En Europe il y a deux endroits où les classes sont données en anglais : Londres et Malmö (Suède)	

CRITÈRE	DESCRIPTION	COMMENTAIRE
Prix	La bibliothèque est gratuite (community). Les plans de support sont : Basic 249\$/y (181€), Starter 999\$/y (729€)	Les différences concernent la disponibilité des modules (Paypal, SMS, Gamekit, etc.) et le degré de support disponible
Licence	<a href="#">Apache public license v2</a>	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	Aucun	
Extensibilité	Oui, on peut écrire des modules pour iOS et Android pour étendre Titanium	



La société Rhomobile propose une série d'outils et de solutions de développement pour les parties clientes et serveur des applications. « Rhodes » est le Framework que propose Rhomobile pour le développement d'applications mobiles. Le développement se fait dans Eclipse à l'aide du plugin « Rho Studio » et du debugger « Rho Debugger » à installer sur l'appareil de test. Une application « Rhodes » est une application hybride de type « Ruby On Rails » pouvant accéder à certains « widgets » et aux ressources matérielles des appareils. On peut le combiner avec un Framework HTML tel que Sencha pour compléter les éléments graphiques qu'il propose.

Rhomobile propose d'autres produits en synergie avec Rhodes. Le serveur RhoConnect (ancien RhoSync) permet d'interfacer et de synchroniser très simplement les données de l'application avec celles hébergées par un site web. NB : RhoConnect n'a pas besoin que l'application cliente soit écrite avec Rhodes. RhoHub permet de développer son application en ligne, de la déployer et d'héberger un serveur de synchronisation des données. RhoConnect et Rhohub sont payants.



Quelques « widgets » natifs, la similarité au très populaire « Ruby On Rails », l'environnement de test et de développement efficace et puissant, la forte structuration du code de type MVC, et la gratuité.

On aime



La courbe d'apprentissage au début et peu d'accès aux « widgets » natifs.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Plateforme	
Type de déploiement	Hybride, Web	
Gestion de L'IHM	Oui	La solution fournit trois composants natifs (Toolbar, tabBar, splitview). Le reste de l'interface graphique est laissé à la charge du développeur. En général on choisira une bibliothèque JavaScript pour construire l'interface utilisateur
Type de mise en page	Coordonnées et pourcentage	On peut utiliser les CSS et les éléments natifs fournis. Cependant généralement on utilisera la mise en page proposée par la solution qu'on a retenue pour créer l'interface graphique
Langage	Ruby, HTML, CSS, JavaScript	
Vitesse de développement	Rapide / Moyenne	Développement similaire à une application Ruby On Rails (RoR), avec le même genre de génération par ligne de commande et l'accès aux « gems » ruby
Courbe d'apprentissage	Rapide / Moyenne	Cela demande de connaître Ruby et les conventions utilisées dans « RoR ». Pour quelqu'un ayant déjà pratiqué « Rails » ça sera très rapide
EDI	Le plugin Eclipse « RhoStudio » ou n'importe quel éditeur supportant le développement web et Ruby	RhoStudio donne accès aux lignes de commandes élémentaires et permet de construire les applications sur les plateformes supportées
Outil graphique pour l'IHM	Aucun	
Débogueur	Rhodes debugger	Pour déboguer on pourra également utiliser une « gem » telle que « Rspecs »
Documentation	<a href="#">Wiki référençant l'API</a> , <a href="#">screencasts</a>	La Documentation est bonne, elle contient des guides et une référence de l'API détaillés avec des exemples expliqués et illustrés. Les « screencasts » sont bien faits et un nouveau est créé chaque semaine. Il manque un tutoriel étape par étape permettant de créer une mini application afin de mieux comprendre le fonctionnement général d'une application Rhodes. Car ce point est très peu traité dans la documentation
Outils de support	Google group, email	Le système de question/réponse basé sur un modèle similaire à « Stackoverflow » est intéressant et performant. La mise en relation avec d'autres développeurs Titanium est une excellente chose
Popularité	Moyenne	
Outils de formation	Webinar gratuit tous les vendredis	

CRITÈRE	DESCRIPTION	COMMENTAIRE
Prix	Rhodes est gratuit, Rhohub à partir de 500MB de données (\$5K, 3671€), à partir de 5GB de données (\$10K, 7342€)	
Licence	<a href="#">Licence MIT</a>	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	RhoHub permet de créer des applications, d'héberger un serveur et de déployer ses applications (Premium \$5 K/y (3652€), Enterprise \$10 K/y (7304€))	
Extensibilité	Oui	
Produits associés	Serveur de synchronisation (RhoSync)	





Le développement mobile « Flex » profite de l'intégration des différents outils Adobe. L'environnement de développement « Adobe Flash Builder » est très complet, depuis l'intégration au workflow des outils graphiques comme Photoshop ou Fireworks jusqu'au monitoring des communications entre le serveur et l'application. « Flash Builder » permet de construire graphiquement son application à la manière d'un « Interface Builder ». Les points négatifs se retrouvent au niveau de l'ergonomie : il y a des problèmes de fluidité et de performance lors de la navigation dans de grosses applications et il n'existe pas pour le moment de « Template » complet pour imiter complètement un aspect natif.

L'environnement de développement très complet et performant, la bonne documentation et nombreux tutoriels, la communauté active, l'accès à de nombreuses ressources natives, la gestion vidéo poussée (enregistrement et lecture), et la gratuité.



On aime



Les performances, aucun aspect natif par défaut, et l'environnement de développement payant.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Framework	
Type de déploiement	Framework	
Gestion de L'IHM	Oui	Proche d'un aspect natif avec des thèmes, cependant ces derniers sont incomplets pour iOS et Android entre autres
Type de mise en page	Pourcentage	
Langage	ActionScript 3, MXML	
Vitesse de développement	Rapide / Moyenne	
Courbe d'apprentissage	Moyenne	Bonne documentation, la communauté est très active et aide beaucoup
EDI	Adobe Flash Builder	
Outil graphique pour l'IHM	Adobe Flash Builder	
Débogueur	Emulateurs et débogueur AIR d'Adobe Flash Builder	
Documentation	<a href="#">Documentation</a> , <a href="#">cookbooks</a>	
Outils de support	<a href="#">Forum</a> , <a href="#">blogs</a> , <a href="#">support</a>	
Popularité	Grande	
Outils de formation	<a href="#">Tutoriels</a> , <a href="#">vidéos</a> , <a href="#">cours privés</a> (exemple : 5 jours 1995€)	
Prix	Gratuit	
Licence	MPL 1.1 ( <a href="#">Mozilla Public License</a> )	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	Oui	
Produits associés	Le runtime AIR sur lequel tourne l'application et l'IDE Adobe Flash Builder qui permet de facilement construire, coder et tester son application	Tous les produits Adobe sont intégrables dans le workflow. Il est très facile par exemple de faire une image sous Photoshop et de la transférer dans Flash Builder

## 3.

## Les challengers



Mobl est un langage récent déclaratif et impératif, il permet de construire une application avec une simplicité déconcertante. Le langage est compilé via un plugin dans Eclipse ou en ligne de commande qui génèrent des fichiers HTML/CSS/JavaScript. Il faut noter cependant que de nombreuses ressources matérielles ne sont pas prises en charge (caméra et GPS par exemple) et que l'aspect d'une application Mobl reste éloigné d'une application native.



La rapidité du développement, la simplicité du code, « design » du langage orienté mobile, l'environnement de test et développement, et la gratuité.

On aime



La documentation pauvre, le langage jeune et pauvre, pas de changement automatique de style par plateforme, la petite communauté, et le peu d'accès aux ressources matérielles.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Langage de programmation	
Type de déploiement	Web	
Gestion de L'IHM	Oui	Plusieurs composants manquent pour obtenir un aspect vraiment natif. Cependant les animations sont particulièrement bien rendues et le projet est prometteur
Type de mise en page	Pourcentages	
Langage	Mobl, JavaScript, HTML	
Vitesse de développement	Rapide	Le langage Mobl est clair et concis
Courbe d'apprentissage	Rapide	
EDI	Plugin Eclipse	
Outil graphique pour l'IHM	Aucun	
Débogueur	Outils de débogage JavaScript	
Documentation	<a href="#">Référence API</a>	Trop succincte et incomplète
Outils de support	<a href="#">Irc</a> , <a href="#">mailing liste</a>	
Popularité	Très petite	
Outils de formation	<a href="#">Tutoriels</a>	
Prix	Gratuit	
Licence	<a href="#">Licence MIT</a>	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	Oui	
Produits associés	Bibliothèques d'accès aux ressources matérielles telles que Phonegap	



OpenPlug est un environnement de développement. Il utilise une API nommée « Elips » qui permet de coder facilement une application pour mobile en Flex et de la déployer sur les différents smartphones. Les fonctionnalités présentées ainsi que les plateformes supportées sont nombreuses. L'IDE est complet mais la documentation est inexistante et les tutoriels très peu nombreux. Cependant l'IDE intègre quelques applications exemples. Il est très difficile de trouver des informations concernant OpenPlug mais il semble plus performant que Flex seul.



De nombreuses plateformes supportées, et de nombreuses ressources matérielles accessibles.

On aime



Une petite communauté, la documentation inexistante, et l'aspect final très éloigné des applications natives.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Logiciel, bibliothèque (Elips)	
Type de déploiement	Dessiné	
Gestion de L'IHM	Oui	Loin d'une ergonomie native
Type de mise en page	Pourcentages	
Langage	XML, CSS, JavaScript, ActionScript, MSXML	
Vitesse de développement	Rapide / Moyenne	Toute l'API de Flex n'est pas présente dans « Elips », certains éléments ont été omis. Il faut parfois faire quelques recherches
Courbe d'apprentissage	Rapide / Moyenne	Rapide si on connaît déjà Flex
EDI	Openplug Studio	
Outil graphique pour l'IHM	Aucun	
Débogueur	Openplug Studio	
Documentation	API référence, tutoriels	
Outils de support	<a href="#">Forum</a> , <a href="#">blog</a> , tickets	
Popularité	Très petite	
Outils de formation	Tutoriaux, screencasts	
Prix	Gratuit, pro (79 €), premium (3990€)	
Licence	Gratuit, pro (79 €), premium (3990€)	
Open source	Non	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	Oui	
Produits associés	Produits Adobe	



Corona permet de créer des applications très performantes en un minimum de code. Il permet d'avoir accès à une bonne partie des ressources matérielles. L'interface graphique est composée de deux couches : une couche est native (alertes et champs de saisie par exemple) et une autre est dessinée en OpenGL (listes et barres de navigation par exemple). La politique actuelle de Corona est axée sur la création de jeux et par exemple l'affichage de « widgets » est encore en bêta. Le SDK est complet (simulateur, débogueur) et les tutoriels ne manquent pas. La communauté est vaste et active.



Le langage de script Lua très simple et puissant, la grosse communauté.

On aime



Orienté jeux, et peu de « widgets » natifs.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	SDK	
Type de déploiement	Natif, dessiné	
Gestion de L'IHM	Oui	
Type de mise en page	Coordonnées	
Langage	Lua	
Vitesse de développement	Rapide / Moyenne	
Courbe d'apprentissage		
EDI	Plugin Eclipse, Bundle Textmate, SDK tools	
Outil graphique pour l'IHM	Aucun	
Débogueur	Le débogueur fourni dans le SDK	
Documentation	<a href="#">Référence de l' API</a> , <a href="#">tutoriels</a> , videos	Bonne documentation
Outils de support	<a href="#">Forum</a>	
Popularité	Moyenne	
Outils de formation	Webinars, vidéos	
Prix	\$349 par an (254€)	
Licence	Payante	
Open source	Non	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	Plugins	





Mosync passe par une programmation en C/C++, donc plus bas niveau que ses concurrents. Cela permet un niveau de performances permettant de traiter des domaines comme le jeu vidéo. Cependant ses utilisateurs l'utilisent surtout pour faire des applications plus classiques mais sur une quantité de terminaux considérable. Il y a d'ailleurs nettement plus de tutoriaux et exemples pour des applications faites sur des « non Smartphones ». La communauté est dynamique, la documentation est complète. L'IDE basé sur Eclipse est très bien conçu.



Énormément de plateformes supportées en plus des smartphones (MoRE émulateur), et la bonne documentation.

On aime



La programmation bas niveau (C/C++).

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	SDK	
Type de déploiement	Natif	
Gestion de L'IHM	Oui	
Type de mise en page	Coordonnées	
Langage	C/C++	
Vitesse de développement	Lente	C/C++
Courbe d'apprentissage	Moyenne / Lente	La bonne documentation et la communauté active aident beaucoup
EDI	Mosync : IDE basé sur Eclipse	Bonne intégration
Outil graphique pour l'IHM	Aucun	
Débogueur	Dans le Mosync IDE	
Documentation	<a href="#">Documentation</a> , <a href="#">Tutoriaux</a> , <a href="#">exemples</a> , <a href="#">référence de l'API</a>	Excellente documentation, de nombreux tutoriaux et des exemples complets
Outils de support	Support commercial pour les plans basic et gold Pro ; <a href="#">Forum</a>	
Popularité	Moyenne	
Outils de formation	<a href="#">Tutoriaux</a>	
Prix	Basic Pro (199€), gold Pro (2999€)	
Licence	<a href="#">GPL2</a> , indie, basic Pro, gold Pro	
Open source	Oui	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	Oui	

## appMobi {!}

Appmobi est une petite librairie JavaScript similaire à Titanium. Son principal intérêt vient de l'environnement de simulation qui donne accès à de nombreuses ressources matérielles habituellement exclues des simulateurs comme l'accéléromètre et le GPS. Appmobi permet de simuler les principaux appareils du marché : iPhone, iPad, HTC, etc.



Un environnement complet et très intégré, une bonne documentation et nombreux tutoriels, et une communauté active.

On aime



Les performances, et l'aspect éloigné de l'aspect natif.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Plateforme (XDK) et bibliothèque JavaScript	
Type de déploiement	Hybride	
Gestion de L'IHM	Non	
Type de mise en page	CSS	
Langage	HTML 5, CSS 3, JavaScript	
Vitesse de développement	Rapide	
Courbe d'apprentissage	Rapide	
EDI	Non	
Outil graphique pour l'IHM	Appmobi XDK permet de voir comment l'application se comporte dans plusieurs terminaux du marché. On peut également tester des ressources matérielles comme l'accéléromètre ou la géolocalisation	L'outil semble buggé pour les changements d'orientation
Débogueur	Débogueurs JavaScript	
Documentation	Documentation de l'API, webservice API et guides	Excellente documentation, de nombreux tutoriaux et des exemples complets
Outils de support	<a href="#">Forum</a>	
Popularité	Petite	
Outils de formation	?	
Prix	Gratuit	
Licence	?	
Open source	?	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	?	
Produits associés	Bibliothèques JavaScript graphiques (ex : Sencha Touch)	



QCFamily est une API JavaScript permettant d'accéder à un nombre impressionnant de ressources matérielles. Cependant sa documentation est pauvre et la communauté très réduite.



De très nombreuses fonctionnalités, et un accès à de nombreuses ressources matérielles.

On aime



Une communauté inexistante, et une documentation très pauvre.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Bibliothèque	
Type de déploiement	Hybride	
Gestion de L'IHM	Non	
Type de mise en page	Pourcentage, coordonnées	CSS
Langage	JavaScript	
Vitesse de développement	?	
Courbe d'apprentissage	Rapide	Petite bibliothèque
EDI	EDI gérant les technologies web standard	
Outil graphique pour l'IHM	Non	
Débogueur	Débogueurs JavaScript	
Documentation	<a href="#">Wiki</a> , <a href="#">blog</a> , <a href="#">référence de l'API</a> . De nombreux exemples intéressants sont inclus dans la bibliothèque	Documentation pauvre mais de nombreux exemples intéressants
Outils de support	<a href="#">Google group</a> , <a href="#">twitter</a>	
Popularité	Très petite	
Outils de formation	Exemples venant avec la bibliothèque	
Prix	Gratuit	
Licence	<a href="#">Licence MIT</a>	
Open source	Oui	
Support des tablettes	?	
Outils de déploiement	Non	
Extensibilité	?	



Worklight intègre Phonegap, on retrouve donc toutes les fonctionnalités de Phonegap dans Worklight. Ce dernier ajoute des fonctionnalités pour gérer les sessions, récupérer des données depuis des webservices, internationaliser l'application, debugger, utiliser des ressources matérielles propres à certaines plateformes.

Worklight ajoute de nombreux outils dont un IDE complet basé sur Eclipse intégrant les principaux simulateurs et un serveur web. Plusieurs fonctionnalités sont disponibles comme :

- Le push.
- La connexion à un large éventail de bases de données et web services.
- Un mécanisme de mise à jour directe évitant d'avoir à faire valider chaque nouvelle mise à jour par les stores.
- Un mécanisme de gestion de version.
- Un mécanisme de « skinning » permettant d'adapter l'application à différents appareils d'une même plateforme.
- Une intégration flexible des processus d'identification et de sécurité de type « SSO ».
- Une console de gestion centralisée pour toutes les applications déployées et leurs versions, adaptateurs d'intégration et règles de push.
- Reporting et statistiques d'utilisation incluant le log d'évènements personnalisés et l'export de données non formatées à des systèmes de Business Intelligence.
- Logs d'audit personnalisables dans des buts de régulation.

Un accès aux ressources matérielles, une synchronisation client/serveur, une gestion de version, et des prix à la carte.



On aime



Cher.

On n'aime pas

## Caractéristiques de la solution

CRITÈRE	DESCRIPTION	COMMENTAIRE
Type de solution	Plateforme, EDI, serveur	
Type de déploiement	Hybride	
Gestion de L'IHM	Non	
Type de mise en page	Pourcentage, coordonnées	CSS
Langage	JavaScript, natives	
Vitesse de développement	?	
Courbe d'apprentissage	?	
EDI	Worklight Studio	
Outil graphique pour l'IHM	Non	
Débogueur	Worklight Studio	
Documentation	Guides de référence	
Outils de support	<a href="#">Forum</a>	
Popularité	Très petite	
Outils de formation	<a href="#">Tutoriels</a> , <a href="#">cours de formation</a>	
Prix	À débattre	
Licence	Adaptable, payante	
Open source	Non	
Support des tablettes	Oui	
Outils de déploiement	Non	
Extensibilité	?	
Produits associés	Frameworks graphiques	



## 4.

# Tableaux comparatifs

## Présentation des tableaux

Les tableaux de ce chapitre comparent les solutions entre elles.

Le premier tableau indique la disponibilité de ces dernières pour les principales plateformes du marché, le second indique les fonctionnalités supportées par les différentes solutions.

### Légende

Disponible	✓
Non disponible	✗
Prochainement disponible	⌚

## Systèmes d'exploitation supportés par les solutions

	ÉDITEUR	PHONEGAP	TITANIUM	RHOMOBILE	FLEX	MOBL	OPENPLUG	CORONA	MOSYNC	APPMOBI	QUICKCONNECT	WORKLIGHT
iOs	Apple	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Android	Google	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Blackberry Os	RIM	✓	✓	✓	✓	✓	✗	✗	⌚	✗	✓	✓
Symbian	Nokia	✓	✗	⌚	✗	✓	✓	✗	✓	✗	✓	✗
Windows mobile	Microsoft	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓
Windows Phone 7	Microsoft	✗	✗	✗	✗	✗	✗	✗	⌚	✗	✗	✗
Bada	Microsoft	✓	✗	⌚	✗	✓	✗	✗	✗	✗	✓	✓

## Fonctionnalités supportées par les solutions

	PHONEGAP	TITANIUM	RHOMOBILE	FLEX	MOBL	OPENPLUG	CORONA	MOSYNC	APPMOBI	QUICKCONNECT	WORKLIGHT
Application multi-threadée	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
Modèle de données	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
SQLite	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓
LocalStorage	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓
Système de fichiers	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Disponibilité réseau	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗	✗
Connexion wifi adhoc	✓	✓	✗	✗	✗	✓	⚠	✗	✗	✗	✓
Bluetooth	✗	✓	✗	✗	✗	✗	⚠	✓	✗	✗	✗
SMS	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗
Email	✗	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗
Téléphone	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗
API des contacts	✓	✓	✓	✓	✗	✓	⚠	⚠	✗	✓	✓
Cartes natives	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Géolocalisation (GPS)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Changement d'orientation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Clipboard	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗
Gesture / Multi-touch	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
Audio (Lecture)	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
Audio (Enregistrement)	✓	✓	✗	✓	✗	✓	✓	⚠	✗	✓	✓
Caméra (Prise de photo)	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Caméra (Enregistrement vidéo)	✗	✓	✗	✗	✗	✗	✗	⚠	✗	✗	✗
Vibration	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Accéléromètre	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Magnétomètre, boussole	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓
Push Notification	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✓
XMPP	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
SOAP	✗	✓	✗	✓	✗	✓	✓	✓	✗	✗	✗
YQL	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Gestion de la mémoire	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗

## 5.

# Conclusion

Les solutions multiplateformes natives ont des difficultés majeures à gérer. En premier lieu les contrôles graphiques des différentes plateformes sont en nombre important. Il faut donc décider quels « widgets » mettre en place, certains n'étant pas disponibles sur les autres plateformes, sous peine de perdre la transparence du multiplateforme. Et pour ces « widgets » communs il faut créer une abstraction apte à gommer les petites différences de comportement ou d'aspect du composant sur les différentes plateformes. Au final des petites différences par rapport à la plateforme native vont se greffer sur ces composants. Par ailleurs, les plateformes ont des [principes d'ergonomie différents](#) : on s'attend à revenir à l'accueil en cliquant sur le nom de l'application d'une application Android, mais sur iOS il n'y a pas de concept de page d'accueil et on utilise le bouton de retour pour revenir en arrière.

La puissance des terminaux et l'intégration de navigateurs performants dans les mobiles a permis d'exploiter ces derniers afin de créer des applications web, les « webapps ». Il manquait à ces sites d'être des applications avec une icône et d'avoir accès à toutes les ressources matérielles comme l'appareil photo ou le GPS. Des solutions ont été créées pour combler ces manques et permettre de fabriquer des applications complètes à partir d'une webapp : les solutions « hybrides ». Ces applications sont nettement moins performantes que les applications natives, car il faut payer le coût du navigateur web, mais elles ont l'avantage de pouvoir fonctionner facilement sur un nombre bien plus important de plateformes. Elles ont les mêmes problèmes ergonomiques

que les applications natives et ne peuvent pratiquement pas présenter d'aspect parfaitement fidèle à l'aspect natif. Mais si on décide de mettre en place un design spécifique, alors elles peuvent présenter de très belles interfaces à l'utilisateur.

Au final on peut dire que chaque catégorie a son intérêt. Les solutions natives ont pour la plupart accès à un niveau de performance permettant de créer des applications riches incluant des animations ou des ergonomies originales comme dans des jeux vidéo. Mais elles sont limitées en nombre de plateformes supportées et elles utilisent des techniques de développement généralement très spécifiques. Les solutions hybrides en revanche permettent de créer des applications qui se déploieront facilement sur une grande quantité de plateformes et elles utilisent des techniques de développement très standards. Mais elles ne peuvent atteindre ni les performances ni l'aspect natif.

Concernant les applications non multiplateformes, développées nativement pour chaque plateforme cible, elles permettent de réaliser des applications variées et de qualité profitant de toute la puissance et de l'ergonomie procurée par la plateforme cible. Mais le coût de développement est pratiquement multiplié par le nombre de plateformes cibles.

## 6.

# Annexe 1 : Terminologie

Les deux tableaux ci-dessous donnent les définitions des intitulés des tableaux présentés pour chaque solution. A leur suite figurent les définitions des termes techniques du document.

## Définition des caractéristiques étudiées

CARACTÉRISTIQUE	DÉFINITION	ENSEMBLES DE DÉFINITION
Type de solution	Nature des solutions présentées	Plateforme, SDK, Framework, Bibliothèque, logiciel
Type de déploiement	Approche prise par la solution pour réaliser des applications multiplateformes	Native, Dessinée, Hybride, Web
Gestion de L'IHM	L'interface graphique est-elle gérée par la solution ?	Oui, Non
Type de mise en page	Manière dont l'interface graphique est organisée. Les éléments sont soit positionnés à l'aide de coordonnées (x, y), soit un système de « layout » effectue le placement des éléments par pourcentage, soit les deux	Coordonnées, Pourcentage
Langage	Le langage de programmation proposé pour coder	Objective C, Java, JavaScript, HTML, XML, etc.
Vitesse de développement	Vitesse de développement en comparaison des autres solutions testées	Rapide, Moyenne, Lente
Courbe d'apprentissage	Vitesse d'apprentissage en comparaison des autres solutions	Rapide, Moyenne, Lente
EDI	EDI fourni avec la solution	Nom d'EDI
Outil graphique pour l'IHM	Existence d'un outil graphique permettant de dessiner l'interface graphique de l'application	Noms d'outils, Aucune
Débogueur	Outils de débogage habituellement utilisés avec la solution	Noms d'outils, Aucune
Documentation	La documentation fournie ou accessible pour la solution	Référence d'API, Tutoriels, etc.

CARACTÉRISTIQUE	DÉFINITION	ENSEMBLES DE DÉFINITION
Outils de support	Support disponible	Email, Forum, etc.
Popularité	Evaluation de la popularité de la solution à l'aide de Twitter, Google, et les outils de communication de chaque solution	Très grande, Grande, Moyenne, Petite, Très petite
Outils de formation	Services de formation proposés	
Prix	Prix de la solution	
Licence	Nom de la licence	MIT, GPL, etc.
Open source	La solution est elle open source ?	Oui, Non
Support des tablettes	Support des tablettes (iPad, etc.)	Oui, Non
Outils de déploiement	La solution fournit-elle un outil afin de déployer l'application dans les « stores » des différentes plateformes ?	Nom d'outil, Aucun
Extensibilité	Possibilité d'étendre la solution, en faisant des plugins par exemple	Oui, Non

Voici un tableau décrivant brièvement les différents systèmes d'exploitation du marché du Smartphone :

## Description de l'environnement des systèmes d'exploitation

SYSTÈME D'EXPLOITATION	ÉDITEUR	APPAREILS	SUPPORT DE TABLETTE	LANGAGES	COMMENTAIRES
iOS	Apple	iPhone, iPod, iPad	Oui	Objective C, Cocoa Touch	
Symbian	Nokia	Téléphones nokia.	Non	C++ et Toolkit	Une minorité des nouveaux smartphones de Nokia fonctionnent aujourd'hui avec Symbian. Windows Phone 7 est le nouveau système d'exploitation sur les smartphones Nokia.
Android	Google	Plusieurs constructeurs	Oui	Java, SDK android	
BlackBerry OS	Research In Motion (RIM)	Téléphones BlackBerry.	Oui, BlackBerry tablet os.	JavaME + extensions propriétaires	

SYSTÈME D'EXPLOITATION	ÉDITEUR	APPAREILS	SUPPORT DE TABLETTE	LANGAGES	COMMENTAIRES
Windows Mobile	Microsoft	Plusieurs constructeurs	Oui	Framework .Net	Ancien système d'exploitation mobile de Microsoft.
Windows Phone 7	Microsoft	Plusieurs constructeurs	Non, il faudra attendre WP8.	Framework .Net	Dernier système d'exploitation de Microsoft avant WP8.
Bada	Samsung	Une partie des téléphones Samsung.	Oui	C++, SDK Bada	

## Définition des fonctionnalités

FONCTIONNALITÉS	DÉFINITION
Application multi-threadée	La solution gère-t-elle le multi-thread ?
Modèle de données	La solution intègre-t-elle un système de modèle de données ?
SQLite	La solution intègre-t-elle les fonctionnalités Create, Read, Update, Delete (CRUD) de SQLite ?
LocalStorage	La solution supporte-t-elle les fonctionnalités CRUD pour le localStorage ?
Système de fichier	La solution fournit-elle un accès au système de fichier de l'appareil ?
Disponibilité réseau	La solution est-elle capable de vérifier la disponibilité du réseau ?
Connexion wifi adhoc	La solution est-elle capable de gérer des connexions adhoc en wifi ?
Bluetooth	La solution fournit-elle un accès au Bluetooth sur l'appareil ?
SMS	La solution a-t-elle une API pour envoyer des SMS depuis l'application ?
Email	La solution a-t-elle une API pour envoyer des emails depuis l'application ?
Téléphone	La solution a-t-elle une API pour passer des appels depuis l'application ?
API des contacts	La solution supporte-t-elle les fonctionnalités CRUD pour accéder à la liste de contacts ?
Cartes natives	La solution utilise-t-elle les API natives d'iOs et d'Android lors de l'utilisation de « Maps » ?
Géolocalisation (GPS)	La solution est-elle capable d'utiliser le GPS de la machine ?
Changement d'orientation (rotation)	La solution est-elle capable de détecter la rotation de l'appareil ?
Clipboard	La solution gère-t-elle les fonctions couper/copier/coller ?
Gesture / Multi-touch	La solution est-elle capable de capter les « gestes » ou le « multitouch » ?
Audio (Lecture)	La solution permet-elle la lecture audio dans l'application ?
Audio (Enregistrement)	La solution permet-elle l'enregistrement audio dans l'application ?
Caméra (Prise de photo)	La solution permet-elle la prise de photo dans l'application ?

FONCTIONNALITÉS	DÉFINITION
Caméra (Enregistrement vidéo)	La solution permet-elle l'enregistrement de vidéo dans l'application ?
Vibration	La solution permet-elle de faire vibrer l'appareil depuis l'application ?
Accéléromètre	La solution permet-elle d'accéder à l'accéléromètre ?
Magnétomètre, boussole	La solution permet-elle d'accéder au magnétomètre ou a-t-elle une API pour créer une boussole ?
Push Notification	La solution contient-elle une API pour gérer les « push notifications » ?
XMPP	La solution a-t-elle une API pour gérer les protocoles XMPP ?
SOAP	La solution a-t-elle une API pour gérer le protocole SOAP ?
YQL	La solution a-t-elle une API pour utiliser le Yahoo Query Language (YQL) ?
Gestion de la mémoire	La solution permet-elle de gérer manuellement la mémoire ?

## Définitions diverses

TERME	DÉFINITION
Runtime	Un runtime est un programme permettant l'exécution d'autres programmes et leur fournissant une couche d'abstraction du matériel et du système d'exploitation. Par exemple, un programme écrit en Java peut s'exécuter sur plusieurs systèmes d'exploitation différents et incompatibles (Windows, Linux, Mac OSX). Lors de son exécution, la machine virtuelle ou runtime fera les correspondances afin que le programme puisse tourner.
Langage de programmation de bas niveau	Un langage de bas niveau par opposition à un langage de haut niveau demande une gestion plus lourde et fine des éléments du programme. Un langage de haut niveau fournit une plus grande couche d'abstraction et manipule des éléments moins techniques. L'aspect principal généralement géré par un langage de haut niveau est la mémoire.



7.

## Annexe 2 : Méthodologie

Les informations ci-dessus ont été trouvées sur Internet, dans la documentation des solutions, via leurs outils de support ou en utilisant les solutions. Une application de test a été réalisée pour les principales solutions afin de mieux comprendre leurs environnements de travail et leurs limites. L'évaluation de la popularité a été estimée en prenant en compte la densité d'utilisation de leurs outils de communication et leur réputation sur Internet (Google et Twitter principalement). L'évaluation de la courbe d'apprentissage et de la vitesse de développement a été jugée d'après la documentation, les commentaires d'utilisateurs et la pratique de la solution (dans le cas des solutions « dominantes »).

## 8.

# À propos de Netapsys



Netapsys est une société d'ingénierie informatique spécialiste des nouvelles technologies (Java J2EE, Microsoft .NET, PHP, Javascript, ...).

Fondée en 2004 par deux experts des nouvelles technologies, Netapsys a développé et cultive un savoir-faire unique de production logicielle. S'inscrivant dans un principe d'amélioration continue, Netapsys capitalise à la fois sur les hommes et sur un outillage avancé d'intégration continue et de mesure permanente de la qualité de ses développements.

La société s'appuie sur sa capacité d'écoute et sur une veille technologique de pointe pour apporter à ses clients PME, grands comptes et services publics une prise en charge clés en main de leurs projets applicatifs, ou de l'évolution de leur système d'information.

Netapsys vous accompagne sur toutes les solutions technologiques créatrices de valeur pour les entreprises et institutions :



Présente à Paris, Nantes, Lyon, Strasbourg et Madagascar, Netapsys compte 350 collaborateurs et a réalisé en 2013 un chiffre d'affaires de 22 millions d'euros. Si vous souhaitez en savoir plus, ou tout simplement nous rencontrer, n'hésitez pas à nous contacter [contact@netapsys.fr](mailto:contact@netapsys.fr). Retrouvez également toute l'actualité de Netapsys sur notre site internet [www.netapsys.fr](http://www.netapsys.fr).



### **Netapsys Conseil**

124-126, rue de Provence

75008 Paris

Tél. : +33(0)1 70 64 27 33

### **Netapsys Atlantique**

17, rue Sanlecque

44000 Nantes

Tél. : +33(0)2 40 89 82 09

### **Netapsys Rhône-Alpes**

18, rue Bourgelat

69002 Lyon

Tél. : +33(0)4 72 16 88 88

### **Netapsys Grand Est**

11, rue de la Haye

Espace Européen de l'Entreprise

67300 Schiltigheim - Strasbourg

Tél. : +33(0)3 68 00 17 58

### **Netapsys Madagascar**

Porte B22, Immeuble ARO Ampefiloha

101 Antananarivo

Tél : +261 20 22 552 99

[contact@netapsys.fr](mailto:contact@netapsys.fr)

[www.netapsys.fr](http://www.netapsys.fr)

[blog.netapsys.fr](http://blog.netapsys.fr)