# Introduction

Un smartphone est un téléphone mobile évolué disposant des fonctions d'un assistant numérique personnel, d'un appareil photo numérique et d'un ordinateur portable. La saisie des données se fait le plus souvent par le biais d'un écran tactile ou, plus rarement d'un clavier ou d'un stylet. Selon le principe d'un ordinateur, il peut exécuter divers logiciels/applications grâce à un système d'exploitation spécialement conçu pour mobiles, et donc en particulier fournir des fonctionnalités en plus de celles des téléphones mobiles classiques comme : l'agenda, la télévision, le calendrier, la navigation sur le Web, la consultation et l'envoi de courrier électronique, la géolocalisation, le dictaphone/magnétophone, la calculatrice, la boussole, l'accéléromètre, le gyroscope, la messagerie vocale visuelle, la cartographie numérique etc. Les appareils les plus sophistiqués bénéficièrent rapidement de la reconnaissance vocale et de la synthèse vocale.

Il est possible de personnaliser son smartphone en y installant des applications additionnelles telles que des jeux ou des utilitaires via un magasin d'applications en ligne différent pour chaque système d'exploitation. Il est nécessaire d'avoir une connexion internet haute débit par l'intermédiaire d'un réseau de téléphonie mobile pour pouvoir utiliser le maximum de leur potentiel. À partir de fin 2007 et du lancement de l'iPhone, ce marché s'étend considérablement jusqu'à dépasser en quelques années celui des téléphones mobiles classiques. En 2014, les ventes mondiales annuelles de smartphones dépassent le milliard.

Il existe plusieurs systèmes d’exploitation dédiés aux smartphones.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ventes mondiales de smartphones par systèmes d’exploitation | | | | |
| **Système d’exploitation** | **2011 (1 trimestre**[**29**](https://fr.wikipedia.org/wiki/Smartphone#cite_note-29) **)** | | **2014 (année**[**3**](https://fr.wikipedia.org/wiki/Smartphone#cite_note-zdn-3)**)** | |
| **Unités** | **Part de marché** | **Unités** | **Part de marché** |
| Android Google | 36 267 800 | 36 % | 1 059 000 000 | 81,5 % |
| iOS Apple | 16 883 200 | 16,8 % | 193 000 000 | 14,8 % |
| Windows Mobile | 3 658 700 | 3,6 % | 34 000 000 | 2,6 % |
| RIM BlackBerry | 13 004 000 | 12,9 % | 6 000 000 | 0,5 % |
| Symbian Nokia | 27 598 500 | 27,4 % | 0 | N/A |
| Linux | 0 | N/A | 0 | N/A |
| Autres | 3 357 200 | 3,3 % | 7 000 000 | 0,6 % |
| **Total** | **100 769 300** | **100 %** | **1 297 000 000** | **100 %** |

## Historique

Android est à la fois une entreprise, un OS, un framework, un écosystème et un logiciel libre. Pour démêler tout cela, commençons par le début.

### La parenthèse General Magic

En 1989, pas encore de smartphone non plus mais les grandes entreprises du secteur informatique ont déjà quelques idées de R&D sur le sujet. Parmi elles, IBM – qui accouchera du IBM Simon en 1992, le tout premier smartphone, mais aussi Apple chez qui Andy Rubin se fait embaucher. Rapidement, Andy Rubin montre un profil d’entrepreneur et participe à la création d’une entreprise au sein d’Apple : General Magic.

Bien trop en avance sur son temps, la vision de General Magic était assez proche de ce qui a pu se faire avec J2ME ou avec Android : Un langage de programmation converti en bytecode, exécuté par une machine virtuelle, de façon distante sur internet ou sur le cloud.

Le support d’Apple étant insuffisant, un tour de table fut fait par General Magic qui obtient l’investissement de grandes entreprises comme Sony et Motorola en 1992 sans agacer Apple. Alors que les tout premiers smartphones voient le jour et qu’Apple se prépare à sortir le Newton, la firme à la pomme attaque General Magic en justice. Un certain nombre de facteurs mèneront à la faillite de la compagnie.

### Le début : Dangers inc.

Plus tard, en 1999, Andy Rubin fonde Dangers inc. dans un contexte où les smartphones peinent à se vendre aux USA tandis qu’en Europe, Nokia commence à accumuler les millions.

L’entreprise Danger accouche rapidement -en 2002- d’un terminal, le Hiptop, ou SideKick. Ce terminal fonctionnait sous DangerOS, un système principalement basé sur Java. Développé en étroite collaboration avec T-Mobile, ce téléphone remporta un très grand succès aux USA notamment auprès des adolescents friands de messagerie instantanée et de SMS.

Le SideKick fut vendu dans beaucoup de pays anglo-saxons mais aussi au Québec par Fido. Mais l’opérateur québécois se fit racheter par le canadien Rogers qui décida de discontinuer le SideKick en raison de son positionnement concurrentiel par rapport à son offre BlackBerry, laquelle bâtait à l’époque son plein.

Dangers attire l’attention et se fait finalement acheter par Microsoft qui en fera rien. L’essentiel des employés de Dangers, dont Andy Rubin, quittèrent Microsoft et notre cher Andy créa une entreprise dont le nom ne vous est pas inconnu.

### Android inc., la consécration

C’est donc en 2003 que la société Android nait. L’entreprise n’est pas très claire quant à ce qu’elle développe mais l’on sait que cela touche au développement mobile.

Déjà très proche de T-Mobile, la compagnie se rapproche progressivement de Google qui l’achètera en 2005. Alors qu’Apple développe l’iPhone dans son lab de Paris – sous la houlette de Jean-Marie Hullot – l’équipe Android travaille activement à un grand projet : faire revivre le SideKick. Mais il ne s’agit pas encore d’une guerre larvée. Ce n’est pas le nouveau SideKick contre l’iPhone.

Au contraire, Google est très enthousiaste à l’idée de l’iPhone. Son objectif est en effet de multiplier les pages vues (et donc les affichages de publicité) sur le mobile et, pour se faire, de bousculer les différents acteurs du marché : Microsoft avec Windows Mobile, Nokia avec Symbian et RIM avec Blackberry. Ces derniers ont en effet beaucoup de mal à offrir une expérience de navigation web convaincante et Google souhaite pousser l’iPhone et sa propre solution, Android, pour forcer les autres acteurs à réagir et à améliorer leurs OS.

C’est dans ce contexte qu’est annoncé le T-Mobile G1, le premier terminal Android.



Figure 1 Commercialisé sous le nom HTC Dream en France

# Architecture d'Android

## Pile logicielle

Le système d'exploitation est constitué d’une pile de composants. Le sens de lecture se fait de bas en haut, puisque le composant de plus bas niveau (le plus éloigné des utilisateurs) est le noyau Linux et celui de plus haut niveau (le plus proche des utilisateurs) est constitué par les applications.

Le système d'exploitation d'Android est basé sur Linux. Si on veut être plus précis, c'est le noyau (« kernel » en anglais) de Linux qui est utilisé. Le noyau est l'élément du système d'exploitation qui permet de faire le pont entre le matériel et le logiciel. Par exemple, les pilotes WiFi permettent de contrôler la puce WiFi. Quand Android veut activer la puce WiFi, on peut imaginer qu'il utilise la fonction « allumerWifi() », et c'est au constructeur de spécifier le comportement de « allumerWifi() » pour sa puce. On aura donc une fonction unique pour toutes les puces, mais le contenu de la fonction sera unique pour chaque matériel.

La version du noyau utilisée avec Android est une version conçue spécialement pour l'environnement mobile, avec une gestion avancée de la batterie et une gestion particulière de la mémoire. C'est cette couche qui fait en sorte qu'Android soit compatible avec tant de supports différents.

Si vous regardez attentivement le schéma, vous remarquerez que cette couche est la seule qui gère le matériel. Android en soi ne s'occupe pas de ce genre de détails. Je ne veux pas dire par là qu'il n'y a pas d'interactions entre Android et le matériel, juste que, quand un constructeur veut ajouter un matériel qui n'est pas pris en compte par défaut par Android, il doit travailler sur le kernel et non sur les couches au-dessus, qui sont des couches spécifiques à Android.

## Android et Java

C'est cette couche qui fait qu'Android n'est pas qu'une simple « implémentation de Linux pour portables ». Elle contient certaines bibliothèques de base du Java accompagnées de bibliothèques spécifiques à Android et la machine virtuelle « Dalvik ».

Votre code est une suite d'instructions que l'on trouve dans un fichier .java qui sera traduit en une autre suite d'instructions dans un autre langage que l'on appelle le « bytecode ». Ce code est contenu dans un fichier .class. Le bytecode est un langage spécial qu'une machine virtuelle Java peut comprendre et interpréter. Les différents fichiers .class sont ensuite regroupés dans un .jar, et c'est ce fichier qui est exécutable. En ce qui concerne Android, la procédure est différente. En fait, ce que vous appelez Java est certainement une variante particulière de Java qui s'appelle « Java SE ». Or, pour développer des applications pour Android, on n'utilise pas vraiment Java SE. Pour ceux qui savent ce qu'est « Java ME », ce n'est pas non plus ce framework que l'on utilise (Java ME est une version spéciale de Java destinée au développement mobile, mais pas pour Android donc).

À noter que sur le schéma le JDK et le JRE sont réunis, mais il est possible de télécharger le JRE sans télécharger le JDK.

La version de Java qui permet le développement Android est une version réduite amputée de certaines fonctionnalités qui n'ont rien à faire dans un environnement mobile. Par exemple, la bibliothèque graphique Swing n'est pas supportée, on trouve à la place un système beaucoup plus adapté. Mais Android n'utilise pas une machine virtuelle Java ; une machine virtuelle tout étudiée pour les systèmes embarqués a été développée, et elle s'appelle « Dalvik ». Cette machine virtuelle est optimisée pour mieux gérer les ressources physiques du système. Elle permet par exemple de laisser moins d'empreinte mémoire (la quantité de mémoire allouée à une application pendant son exécution) ou d'utiliser moins de batterie qu'une machine virtuelle Java.

La plus grosse caractéristique de Dalvik est qu'elle permet d'instancier (terme technique qui signifie « créer une occurrence de ». Par exemple, quand vous créez un objet en java, on instancie une classe puisqu'on crée une occurrence de cette classe) un nombre très important d'occurrences de lui-même : chaque programme a sa propre occurrence de Dalvik et elles peuvent vivre sans se perturber les unes les autres. La figure suivante est un schéma qui indique les étapes nécessaires à la compilation et à l’exécution d'un programme Android standard.

On voit bien que le code Java est ensuite converti en bytecode Java comme auparavant. Mais souvenez-vous, je vous ai dit que le bytecode Java ne pouvait être lu que par une machine virtuelle Java, mais que Dalvik n'était pas une machine virtuelle Java. Il faut donc procéder à une autre conversion à l'aide d'un programme qui s'appelle « dx » qui s'occupe de traduire les applications de bytecode Java en bytecode Dalvik, qui, lui, est compréhensible par la machine virtuelle.

## Application

Structure d’une application  
• Activité (android.app.Activity)  
Programme qui gère une interface graphique  
• Service (android.app.Service)  
Programme qui fonctionne en tâche de fond sans interface

• Fournisseur de contenu (android.content.ContentProvider)

Partage d’informations entre applications

• Ecouteur d’intention diffusées (android.content.BroadcastReceiver) :

Permet à une application de récupérer des informations générales (réception d’un SMS, batterie faible, …)

### Intents

Les Intents permettent de gérer l'envoi et la réception de messages afin de faire coopérer les applications. Le but des Intents est de déléguer une action à un autre composant, une autre application ou une autre activité de l'application courante.

Un objet Intent contient les information suivantes:

• le nom du composant ciblé (facultatif)

• l'action à réaliser, sous forme de chaine de caractères

• les données: contenu MIME et URI

• des données supplémentaires sous forme de paires clef/valeur

• une catégorie pour cibler un type d'application

• des drapeaux (informations supplémentaires)

On peut envoyer des Intents informatifs pour faire passer des messages. Mais on peut aussi envoyer des Intents servant à lancer une nouvelle activité.

## La sécurité avec Android

Le système Android installe toutes les applications Android avec un identifiant unique d'utilisateur et de groupe. Chaque fichier de l'application est privé à cet utilisateur généré, les autres applications ne peuvent pas accéder à ces fichiers. En outre, chaque application Android est lancée dans son propre processus.

Par conséquent, par l'intermédiaire du noyau Linux sous-jacent, chaque application Android est isolée des autres applications en cours d'exécution.

Si des données doivent être partagées, l'application doit le faire explicitement via un composant Android qui gère le partage des données, par exemple, via un service ou un fournisseur de contenu.

## Les permissions avec Android

Android contient un système d'autorisation et prédéfinit des autorisations pour certaines tâches. Chaque application peut demander les autorisations nécessaires et définir de nouvelles autorisations. Par exemple, une application peut déclarer qu'elle nécessite l'accès à Internet.

Les permissions ont différents niveaux. Certaines autorisations sont automatiquement accordées par le système Android, certaines sont automatiquement rejetées. Dans la plupart des cas, les permissions demandées sont présentées à l'utilisateur avant l'installation de l'application. L'utilisateur doit décider si ces permissions seront données à l'application.

Si l'utilisateur refuse une permission demandée, l'application concernée ne peut être installée. La vérification de l'autorisation est effectuée uniquement lors de l'installation, les permissions ne peuvent être refusées ou accordées après l'installation.

Une application Android déclare les permissions requises dans son fichier de configuration AndroidManifest.xml. Elle peut également définir des permissions supplémentaires qu'elle peut utiliser pour restreindre l'accès à certains composants.