



android

*An Open Handset Alliance Project*

# Introduction

O.Legrand  
G. Seront

# Open Handset Alliance Project

- Regroupement d'entreprises :
  - de télécommunications,
  - de constructeurs d'appareils mobiles,
  - de développeurs de logiciels, ...

Asus, Acer, LG Electronics, Motorola, Samsung, Sony Ericsson, Garmin, Toshiba, Google, eBay , Intel, Texas Instruments, Telecom Italia, Telefónica, China Mobile Communications Corporation, ...



# Open Handset Alliance Project

- But :
  - Offrir une plateforme pour appareils mobiles
    - Gratuite
    - Ouverte
    - Complète
- Appelée « Android »

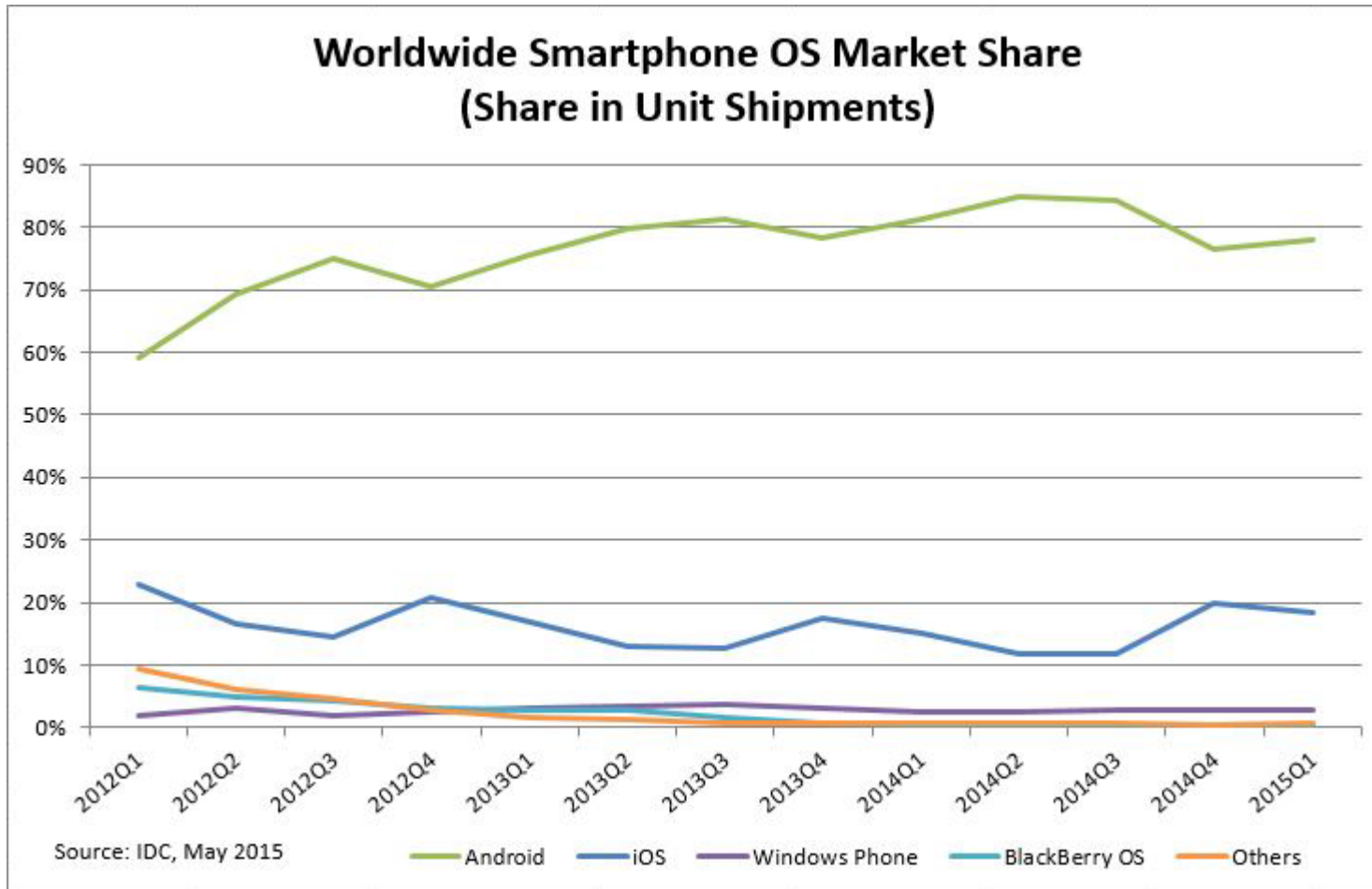


# Android

- Conçu pour les appareils mobiles
- Permet le développement d'applications en Java
- Comprend un système d'exploitation : Linux version 3.10 (pour Lollipop)
- Permet d'accéder à toutes les fonctionnalités de l'appareil

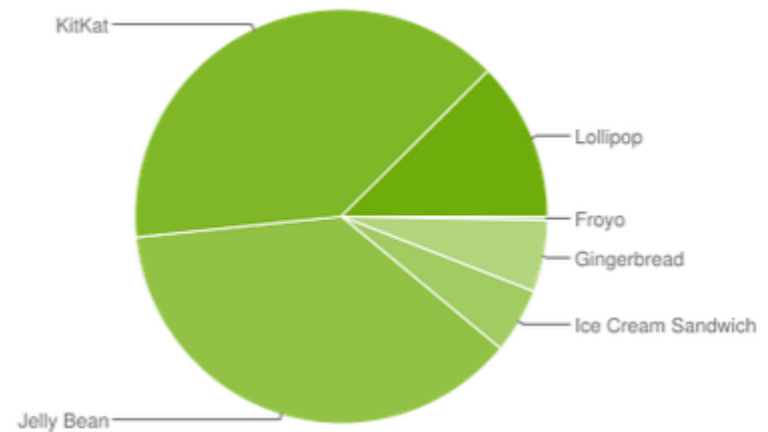


# Les concurrents



# Versions

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%



# Caractéristiques d'Android

- Environnement permettant la réutilisation et le remplacement de composants logiciels
- Machine virtuelle conçue pour les appareils mobiles
  - Dalvik Virtual Machine
- Bibliothèques graphiques
  - 2D, 3D basée sur la spécification OpenGL ES 1.0

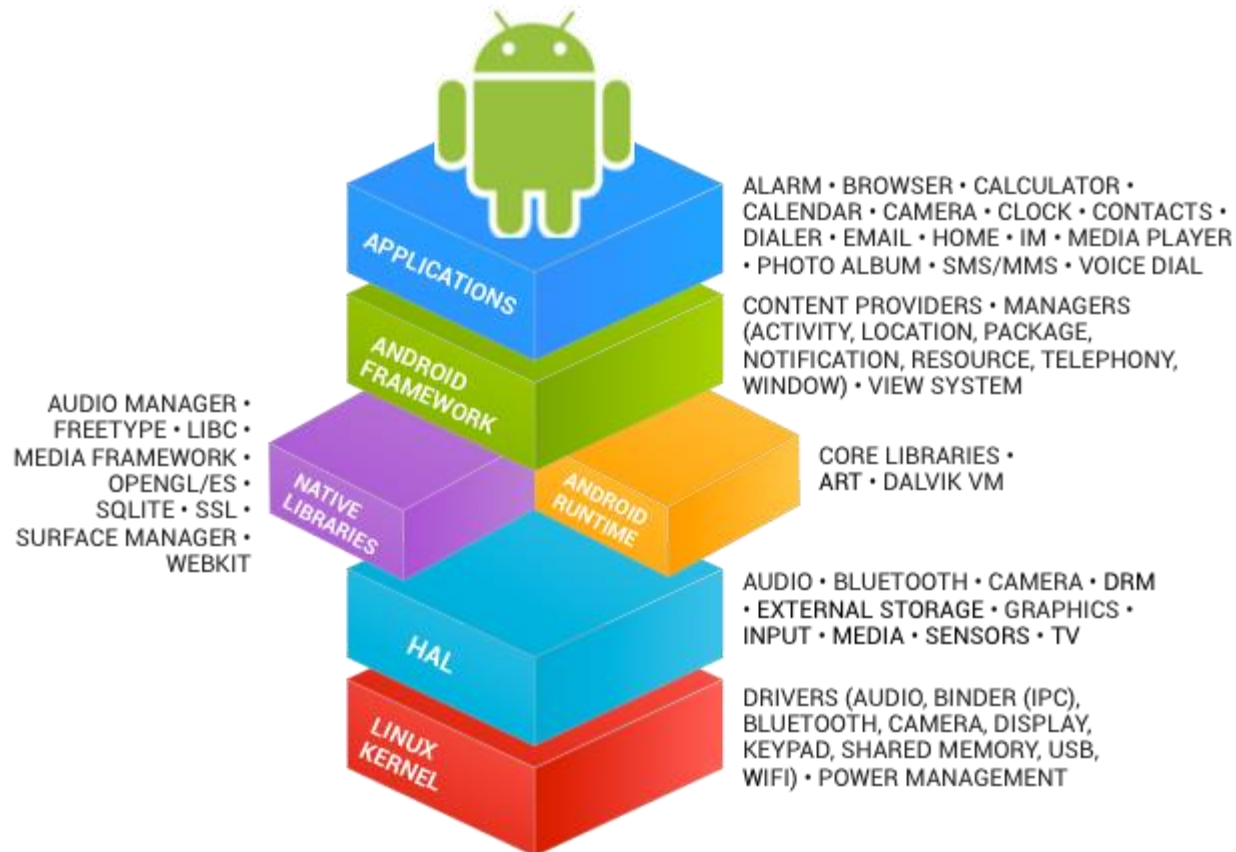


- Contient un sgbd :
  - SQLite
- Supporte les formats images, audios, vidéos :
  - MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF
- Communications :
  - Bluetooth, 3-4G, WiFi
- Utilisation possible :
  - de l'appareil photo, du GPS, de la boussole, de l'accéléromètre
- Environnement de développement :
  - émulateur
  - outils de debugging
  - Android Studio (basé sur IntelliJ)





# Architecture



# Environnement applicatif

- Android contient des applications écrites en Java :
  - gestion des SMS
  - consultation du calendrier
  - consultation de cartes
  - navigateur internet
  - gestion des contacts
- Accessibles par toute application



# Environnement applicatif

- Les développeurs ont accès à toutes les API du système (y compris les applications natives)
- L'architecture applicative facilite la réutilisation de composants logiciels :
  - toute application peut publier ses capacités, tout autre application peut alors faire appel à ces capacités
  - l'utilisateur peut remplacer tout composant par un autre présentant les même capacités



# Environnement applicatif

- Comprend les modules :
  - « Window Manager » et « View System » :
    - modules de gestion des composants graphiques
  - « Content Provider » :
    - module permettant aux applications de s'échanger des données (ex: des contacts)
  - « Ressource Manager »
    - module donnant accès à des ressources autres que du code : strings, graphiques, layouts, menus, animation.
  - « Notification Manager »
    - module permettant à une application d'afficher une alerte dans la barre d'état de l'appareil



# Environnement applicatif

- « Activity Manager »
  - gère les cycles de vie des applications. Il facilite le passage de l'une à l'autre.
- « Location Manager »
  - permet d'utiliser les données provenant du gps, de la boussole, de l'accéléromètre.
- « Telephony Manager »
  - permet d'accéder aux fonctionnalités liées au gsm.
- « XMPP Service »
  - gère les connections P2P entre appareils.
- Ces modules sont accessibles par toute application



# Native Librairies

- Android comprend des librairies c/c++ :
  - System C library
  - Media Libraries
  - Surface Manager
  - LibWebCore
  - SGL
  - 3D libraries
  - FreeType
  - SQLite
- Ces librairies sont :
  - Utilisées par les modules systèmes
  - Accessibles par toute application



# Android Runtime

- Android possède :
  - des « core libraries » présentant les même fonctionnalités que celles offertes par le langage Java
  - Une machine virtuelle propre (Dalvik Virtual Machine)



# Dalvik Virtual Machine (DVM)

- Optimisée pour les appareils mobiles
- Plusieurs DVM peuvent tourner simultanément
- Chaque application tourne dans sa propre machine virtuelle
- Un fichier source est compilé (.java->.class)
- Le byte code (.class) est transformé en (.dex) par l'outil *dx*
- La machine virtuelle exécute le fichier (.dex)





# Depuis Lollipop: Android Run Time (ART)

- Compatible avec Dalvik
  - Sauf optimisation par certains outils
- Ahead-of-Time compilation (AOT)
  - Pas « just-in-time » mais avant usage
  - dex2aot
  - Converti en code « natif »
- Meilleur Garbage Collection
- Meilleur Debugging



# Linux Kernel

- Le système d'exploitation utilisé par Android est la version 3.10 de Linux (pour Lollipop)



# Ressources

- Google maintient plusieurs sites d'aide:
  - <http://developer.android.com/index.html>
    - Pour les développeurs d'applications
  - <https://source.android.com/index.html>
    - Pour ceux qui veulent modifier ou porter Android sur d'autre plateforme
    - Les sources d'Android sont disponibles!

