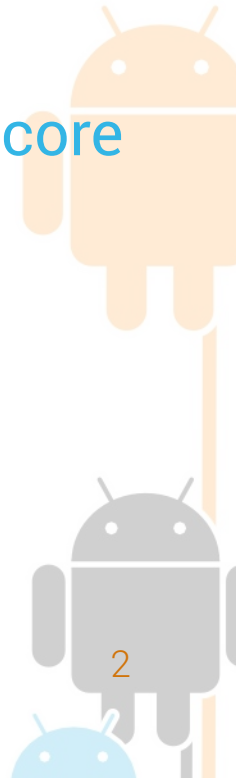
IN01
Programmation Android

01 – Introduction

Yann Caron
Avec l'aide de Jean-Marc Farinone

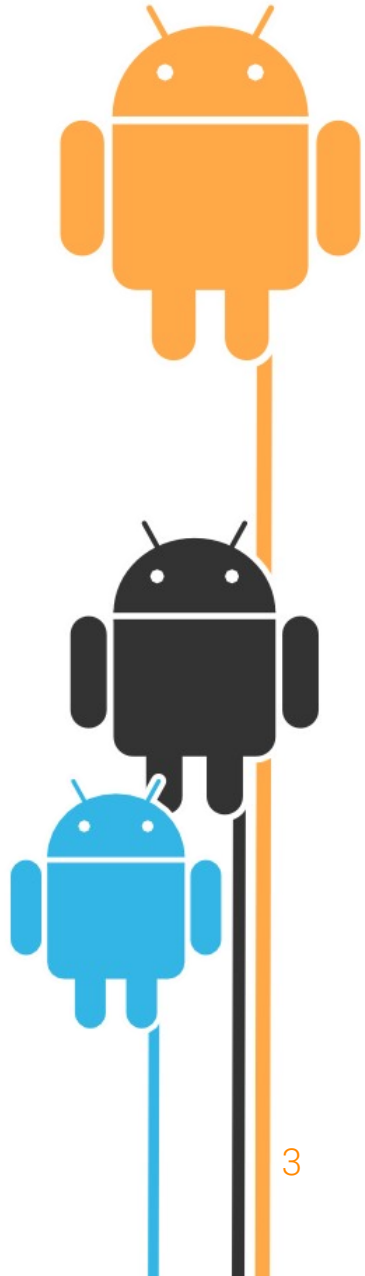
Présentations

- Yann Caron
- Skyguide (SCADA C/C++ et Java WinCC OA, SI C#.net / SqlServer)
- EICnam – Algoid
- CyaNn74@gmail.com
- Sur Developpez – http://caron-yann.developpez.com (pas encore au niveau de JMDoudou)
- Google+ (Yann Caron ou Algoid)
- Et vous ?



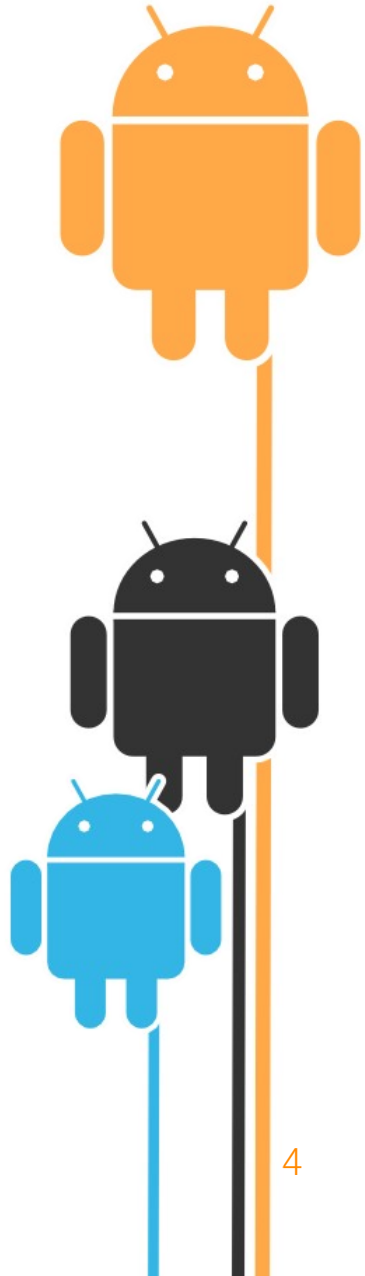
Sommaire du cours

- 01 – Présentation d'Android
- 02 – La plateforme Android
- 03 – IHM Bases
- 04 – Databases
- 05 – Google Map
- 06 – Publication
- 07 – Techniques avancées



Sommaire - Séance 01

- La plateforme
- Historique et versions
- Architecture, JAR vs DEX
- Outils et IDE
  - Eclipse – ADT
  - Netbeans – NAndroid
  - Android Virtual Device (AVD)
  - Android Debug Bridge (ADB)



IN01 – Séance 01

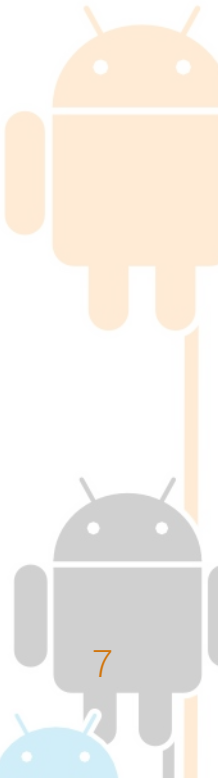
La plateforme

La plateforme Android

- Android : système embarqué open source pour smartphone, tablette, MP3
- Une startup rachetée en 2005 par Google
- Logo bugdroid (libre de droits CC BY 3.0)

La plateforme Android

- OHA (Open Handset Alliance)
  - Consortium Google, opérateurs, constructeurs et éditeurs logiciels
  - Favoriser l'innovation sur les appareils mobiles
  - Plateforme véritablement ouverte, complète
  - Et... gratuite

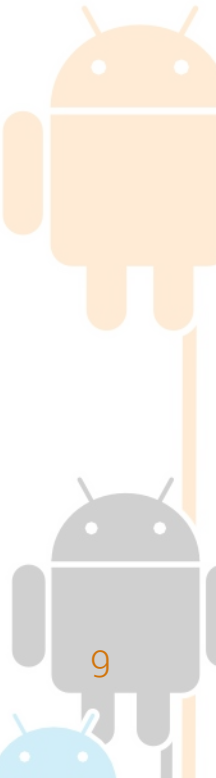


IN01 – Séance 01

Historique et versions

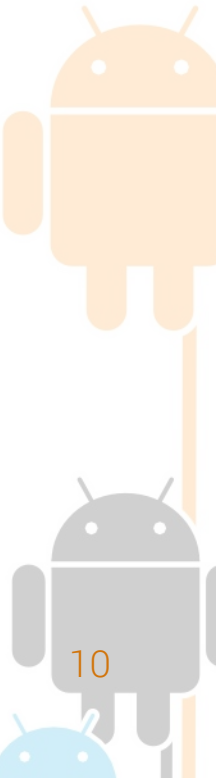
La plateforme Android

- Noyau Linux
- Open source (Open Governance Index de 23 % selon http://www.visionmobile.com)
- Site de référence : http://developer.android.com
- Source : http://fr.wikipedia.org/wiki/Android



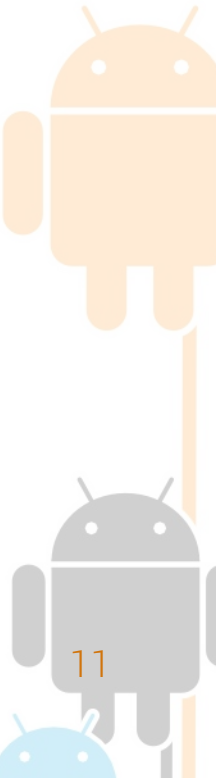
Versions

- 1.0 : Connue des développeurs : fin 2007
- 1.1 : Incluse dans le 1er téléphone, le HTC Dream
- 1.5 : Cupcake : avril 2009
- 1.6 : Donut : septembre 2009
- 2.0 et 2.0.1 : à cause de nombreux bogues
- 2.1 : Eclair : janvier 2010
- 2.2 : FroYo (Frozen Yogourt) : mai 2010



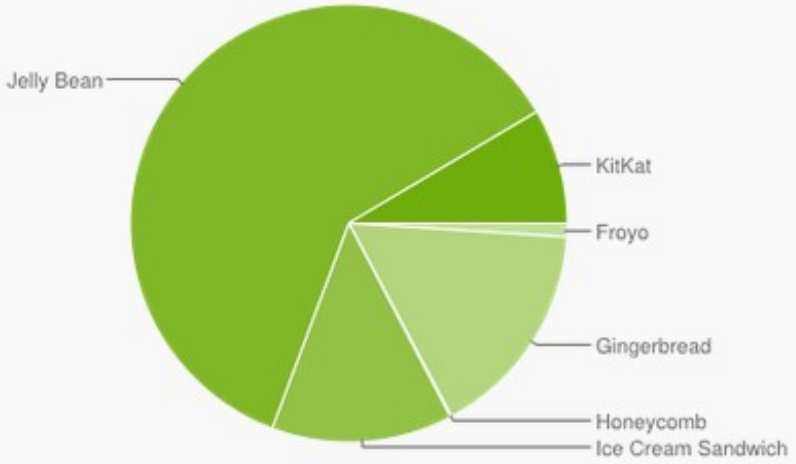
Versions

- 2.3 : Gingerbread (pain d'épice) : décembre 2010
- 3.0 : Honeycomb (rayon de miel) : janvier 2011
- 4.0 : Ice Cream Sandwich : version unifiée Smartphone, Tablette et GoogleTV : octobre 2011
- 4.1 : Jelly bean : juillet 2012
- 4.2.2 : API 17 : février 2013
- 4.3 : API 18 : juillet 2013
- 4.4 : KitKat : novembre 2013



Parts des versions – mai 2014

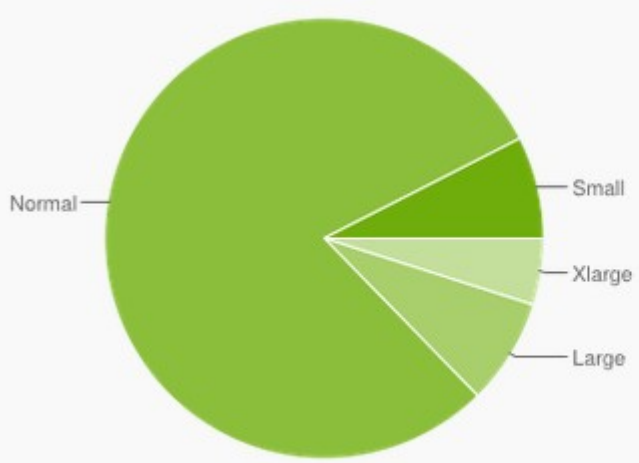
Version	Codename	API	Distribution
2.2	Froyo	8	1.0%
2.3.3 - 2.3.7	Gingerbread	10	16.2%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	13.4%
4.1.x	Jelly Bean	16	33.5%
4.2.x		17	18.8%
4.3		18	8.5%
4.4	KitKat	19	8.5%

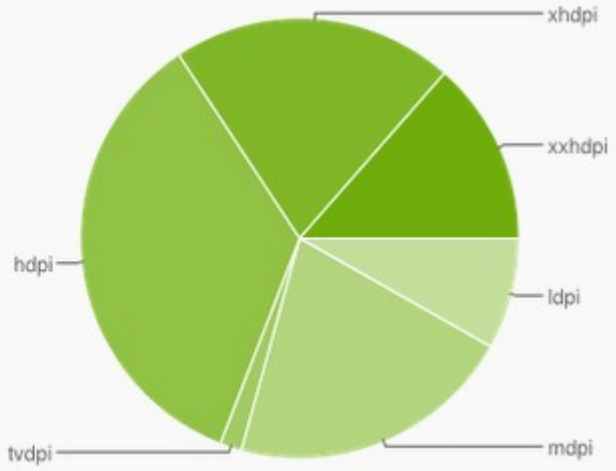


- À étudier avant chaque projet
- Source : http://developer.android.com/about/dashboards/index.html

Parts des résolutions et densités

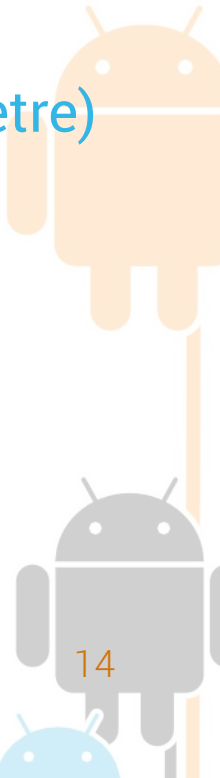
	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	7.5%						7.5%
Normal		12.5%		33.9%	19.9%	13.5%	79.8%
Large	0.6%	4.4%	1.6%	0.6%	0.6%		7.8%
Xlarge	0.1%	4.2%		0.3%	0.3%		4.9%
Total	8.2%	21.1%	1.6%	34.8%	20.8%	13.5%	





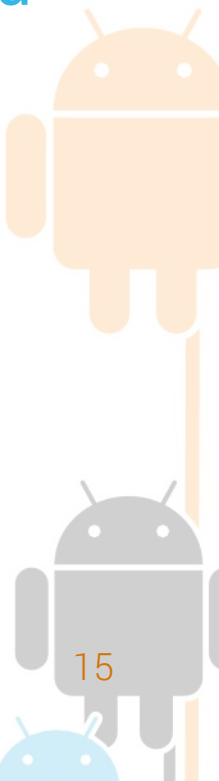
Smartphone != ordinateur

- Android tire parti des particularités des smartphones :
  - interface homme-machine adaptée (tactile, widget)
  - divers modes : vibreur, sonnerie, silencieux, alarme
  - notifications (d'applications, d'e-mails, de SMS, d'appels en instance)
    de boussole, accéléromètre, GPS
  - divers capteurs (gyroscope, gravité, accélération linéaire, baromètre)
  - NFC, RFID (technologie de cartes à puce, HF courte portée)
  - téléphonie (GSM) et réseau EDGE, 3G, 4G, etc.



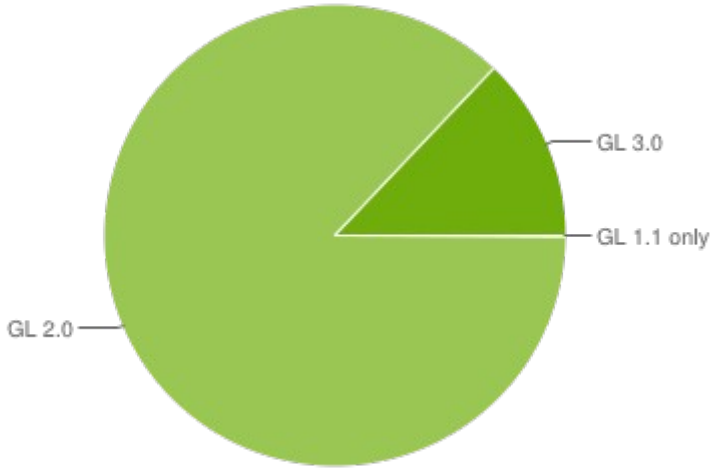
Smartphone != ordinateur

- En plus de ce qu'on peut avoir sur un ordinateur :
  - navigateur
  - bibliothèques graphiques 2D, 3D (Open GL)
  - base de données (SQLite, DB4O), applications de rendu multimédia (audio, vidéo, image) de divers formats
  - réseau Bluetooth et Wi-Fi
  - Webcam, APN
- Et des outils et bibliothèques Java (XStream…)



Parts des versions d'OpenGL

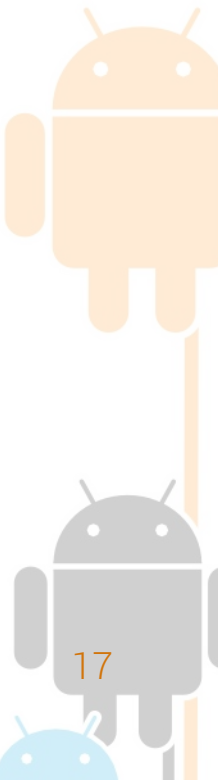
OpenGL ES Version	Distribution
1.1 only	0.1%
2.0	87.0%
3.0	12.9%



- Projets en 3d : jeux, rendu
- LibGDX, AndEngine, Unity, ShiVa, etc.

Google Play

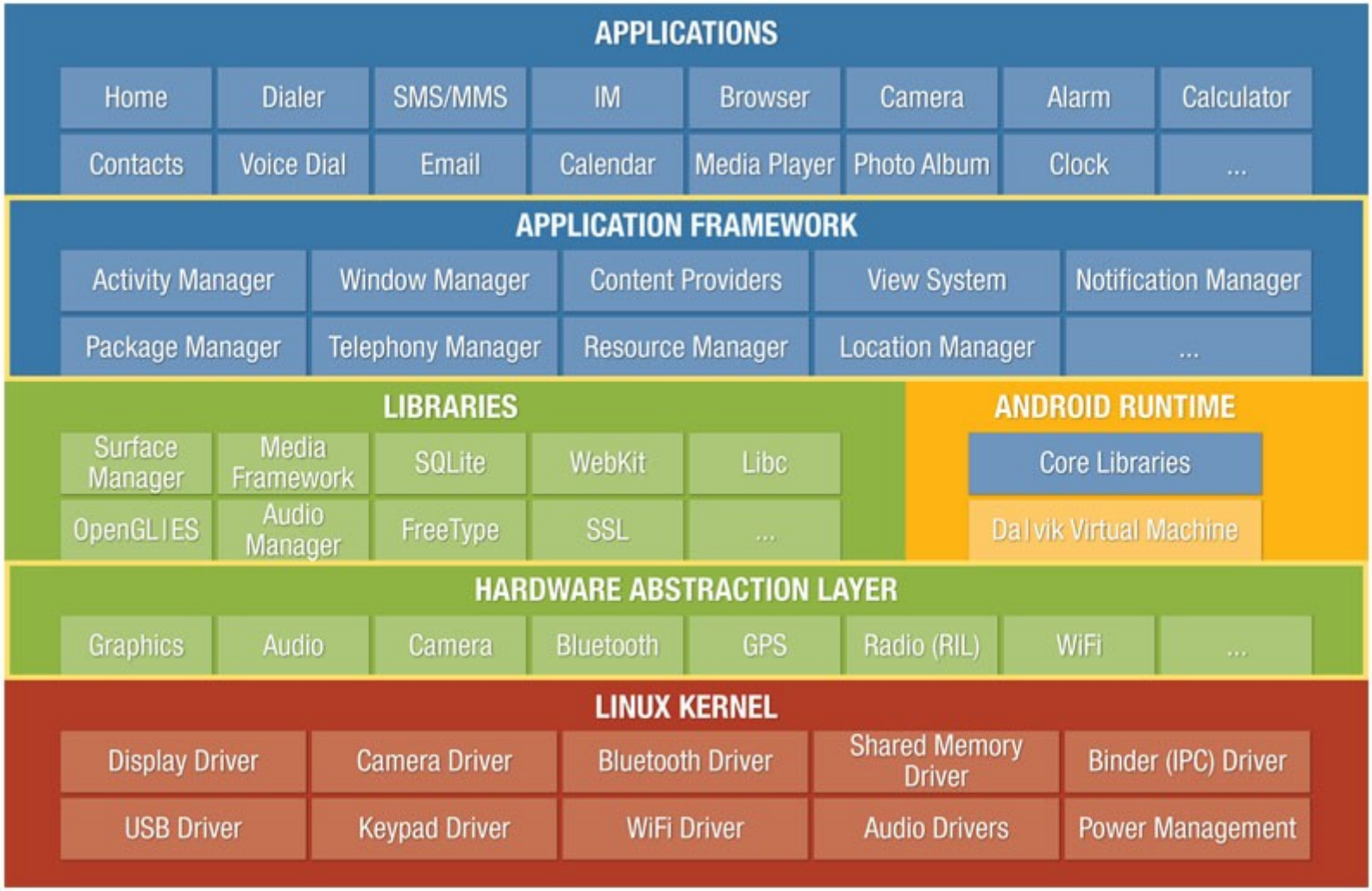
- Android Market est « né » le 22 octobre 2008
- Google Play viendra le remplacer le 6 mars 2012. Il fusionne les autres services Google (VOD, musique, livres, bd, etc.)
- 30 octobre 2013, Google Play compte 700 000 applications et égalise avec l'app store d'Apple
- Les développeurs sont rémunérés 70 % contre 30 % qui rétribuent Google
- Chaque nouveau développeur paie 25 $ comme frais de dossier (une seule fois)



IN01 – Séance 01

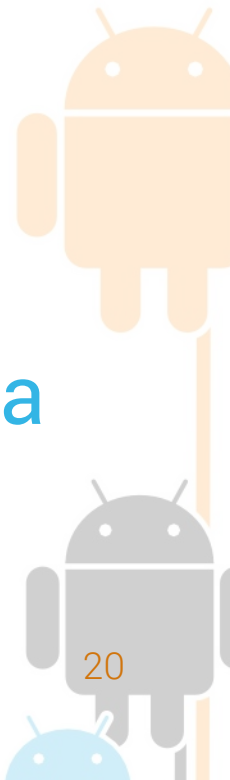
Architecture, JAR vs DEX

Vue d'ensemble



Architecture

- Multiplateforme mais performant
  - « Write once, execute everywhere » la philosophie du langage Java
- Partie haut niveau (en bleu) du Java
- Partie bas niveau (en vert et rouge) du code c/c++ compilé
- La Dalvik VM permet d'interpréter le code Java



Architecture partie Java

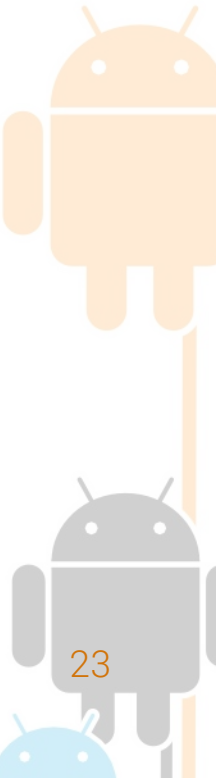
- La couche « Applications » : Android est utilisé dans un ensemble contenant déjà des applications natives comme un client de mail, des programmes pour envoyer des SMS, d'agenda, de navigateur web, de contacts personnels
- La couche « Application Framework » : cette couche permet au programmeur de construire de nouvelles applications. Cette couche fournit la gestion :
  - des Views (= IHM)
  - des ContentProviders = l'accessibillité aux données des autres applications (ex. : les contacts) et donc les partages de données
  - des ressources = les fichiers non code comme les images, les écrans (Resource Manager)
  - des Notifications (affichage d'alerte dans la barre de titre)
  - des Activitys = l'enchaînement des écrans

Architecture partie compilée

- La couche "Libraries" (bibliothèques) = couche logicielle basse pour utiliser :
  - les formats multimédia : images, audio et vidéo
  - les dessins 2D et 3D, bitmap et vectoriels
  - une base de données SQL (SQLite)
  - l'environnement d'exécution (Android Runtime). Toute application est exécutée dans son propre processus, dans sa propre Dalvik virtual machine
  - le noyau Linux sur lequel la Dalvik virtual machine s'appuie pour gérer le multithreading, la mémoire. Le noyau Linux apporte les services de sécurité, la gestion des processus, etc.

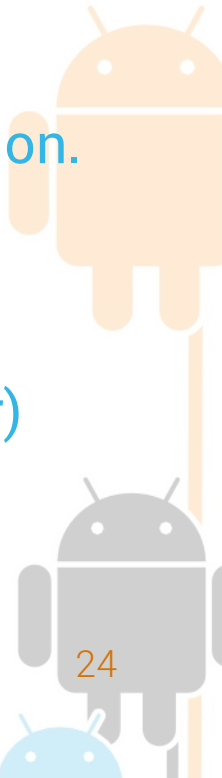
Dalvik Virtual Machine (DVM)

- Est la machine virtuelle Java pour les applications Android
- Conçue pour exécuter du code Java pour des systèmes ayant des contraintes de place mémoire et rapidité d'exécution
- Exécute du code .dex (Dalvik executable) = des .class adaptées à l'environnement Android
- Écrit par Dan Bornstein d'où le nom (= village islandais dont sont originaires certains de ses ancêtres)
- A été choisi par Google, car une machine Android peut lancer plusieurs instances de la DVM efficacement (comme en Java)
- Référence : http://en.wikipedia.org/wiki/Dalvik_virtual_machine

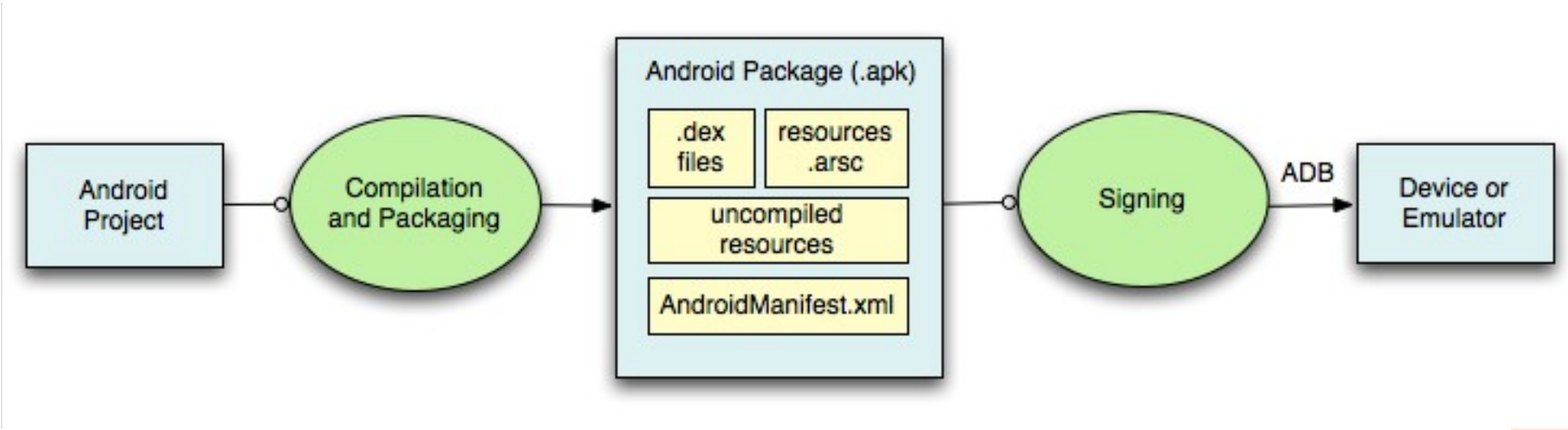


Dalvik Virtual Machine (DVM)

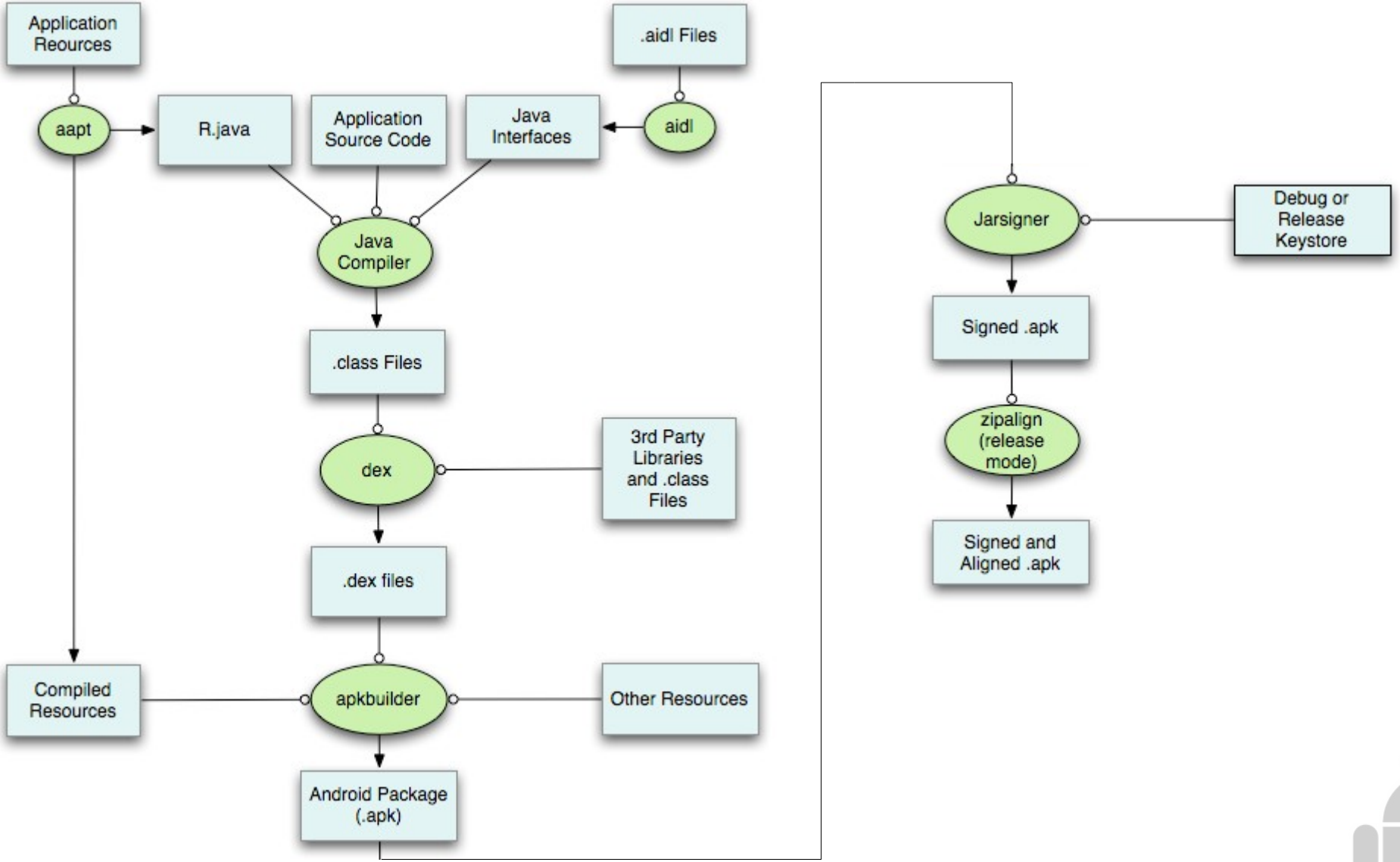
- Le code de la DVM est open source (Apache License 2.0) : http://code.google.com/p/dalvik/
- Machine à registre (register based) vs Java, machine à pile (stack based)
- JIT (Just in time compiler) introduit avec Android 2.2
- Performances controversées (selon Oracle, 3x moins par rapport à HotSpot). Facteur 100 avec une application native
- Un remplacement prévu. ART qui précompile le byte-code à l'installation.
  - Inconvénient, des app plus volumineuses
  - Un gain de performance non encore prouvé
- Android Asset Packaging Tool (AAPT) convertit les JAR en dex (dexer)



En détail


Android Project
Compilation and Packaging
Android Package (.apk)
.dex files
resources .arsc
uncompiled resources
AndroidManifest.xml
Signing
ADB
Device or Emulator

Encore plus en détail



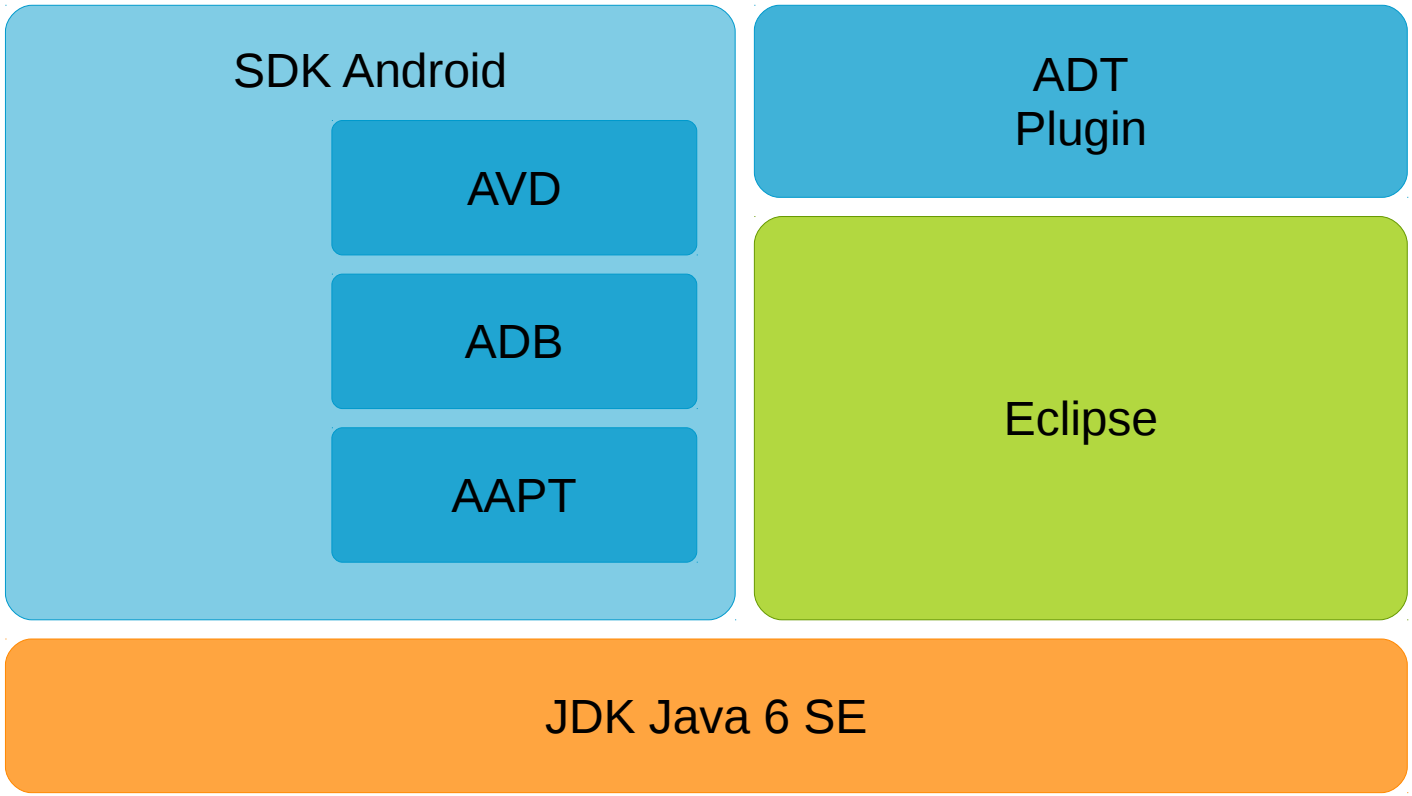
IN01 – Séance 01

Outils et IDE

Installation

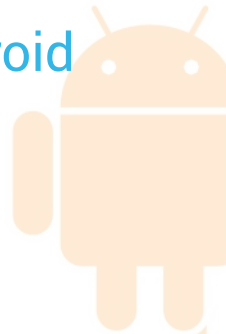
- Facilitée depuis fin 2012
- Installer JDK Java6 SE
- Télécharger l'ADT Bundle depuis
- http://developer.android.com/sdk
- Il inclut : Eclipse, le SDK Android, le plugin Eclipse

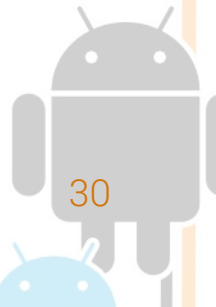
La pile des outils



Le SDK Android

- l'Android SDK (Software Development Kit) amène des outils :
  - un environnement de développement
  - une machine virtuelle Java adaptée : la Dalvik virtual machine
  - un environnement debugueur DDMS (Dalvik Debug Monitor Service) utilisant adb (Android Debug Bridge)
  - un environnement de construction d'applications Android AAPT (Android Asset Packaging Tool)
  - des émulateurs de téléphones ou de tablettes AVD (Android Virtual Device)
  - et une énorme API (voir http://developer.android.com/reference/packages.html)





Eclipse

- Développement en Java
- Divers outils
  - WYSIWYG
  - File explorer
  - Screen capture
  - Threads, Heap
  - Allocation tracker
  - Profiler, etc.

IDE Autres

- Android Studio
  - Solution de Google annoncée au Google IO 2013
  - Basé sur Intellij IDEA
- Intellij IDEA (standalone)
- AIDE – Android IDE – Java, C++
  - Solution tablette et smartphone
  - Compatible avec les projets Android Studio
- Netbeans et NBAndroid
  - Pas de WYSIWYG, certains aspects mal intégrés, en partie payant.

Outils indispensables

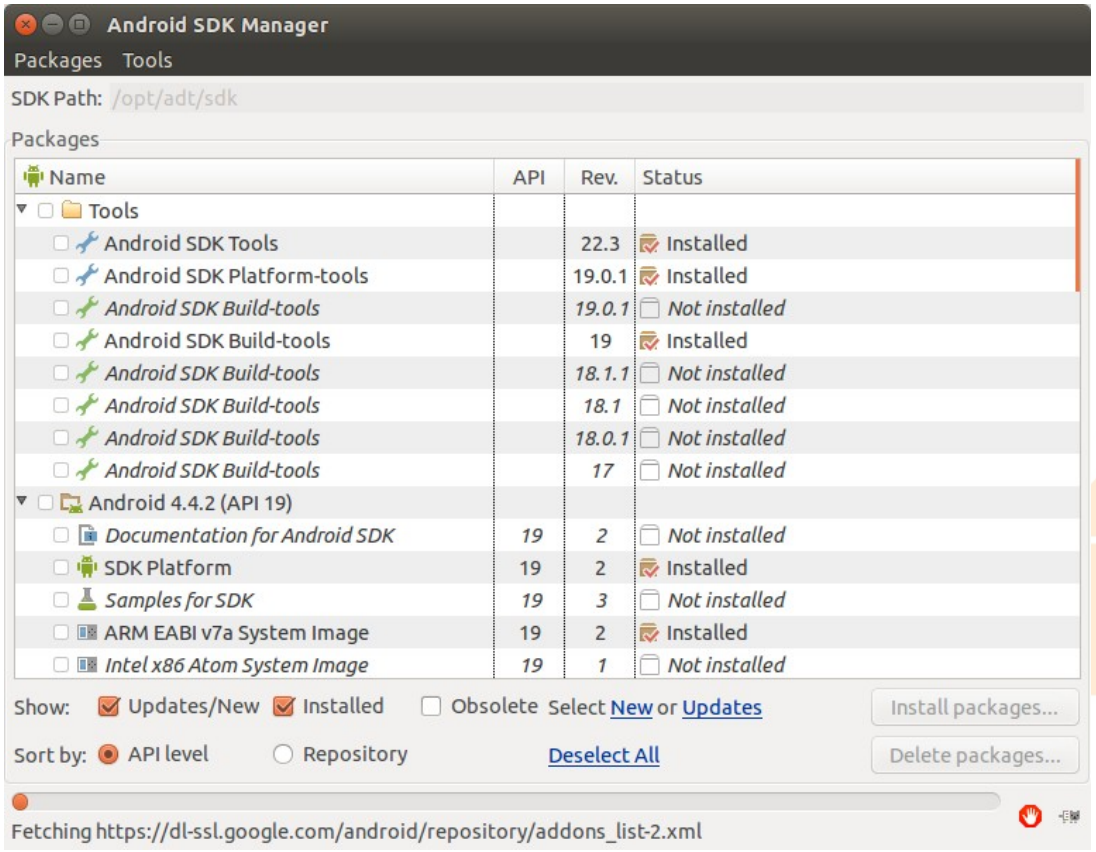
- aLogCat
- Un explorateur de fichiers
- APK Manager, Advanced Task Killer
- Un admin de bases de données SQLite
- Optionnel : Google Analytics





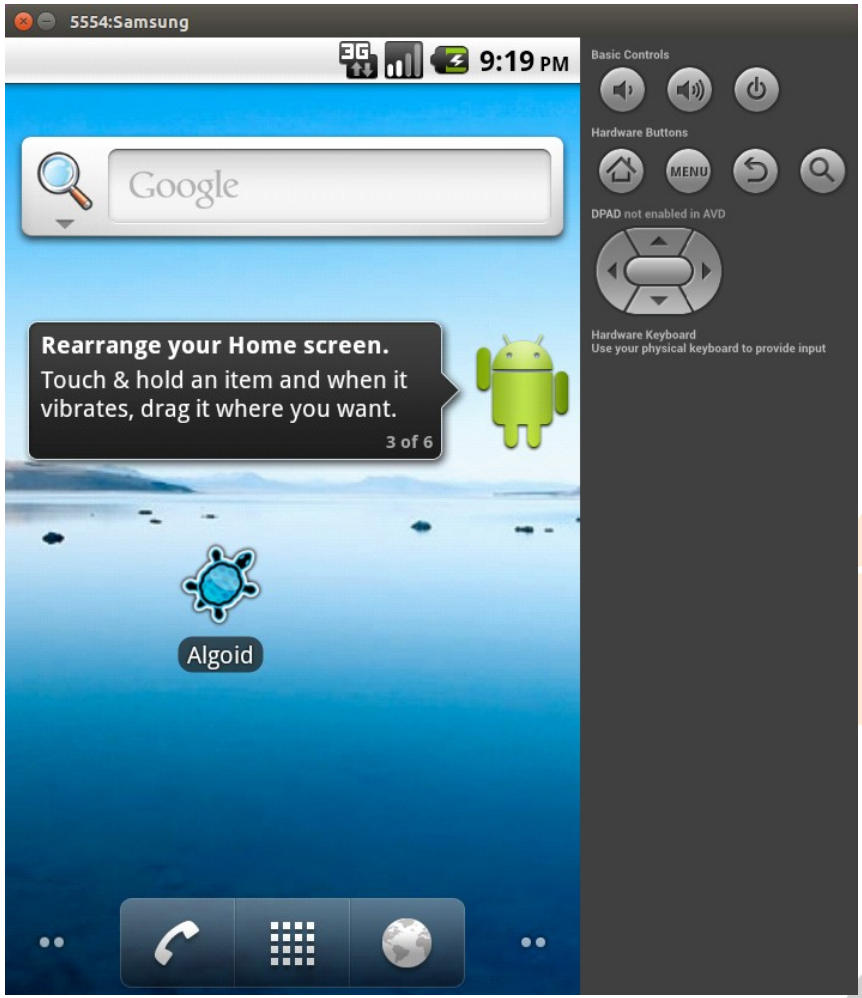
Android SDK Manager

- Gestionnaire de versions de SDK centralisé
- Google fournit les données au format XML



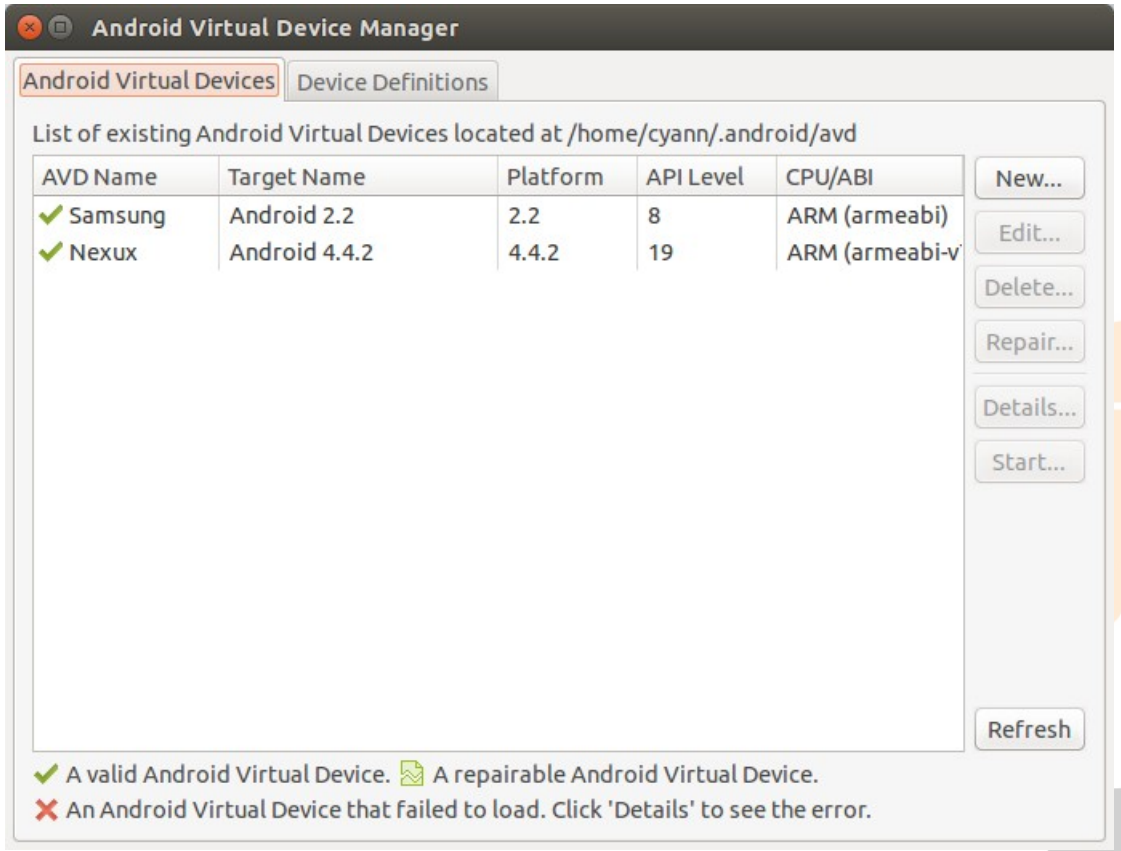
Android Virtual Device (AVD)

- Multiplatforme : win, Linux, macos
- Multirésolution
- Multi-os
- Mais plus lent qu'un vrai device
- Senseurs émulés



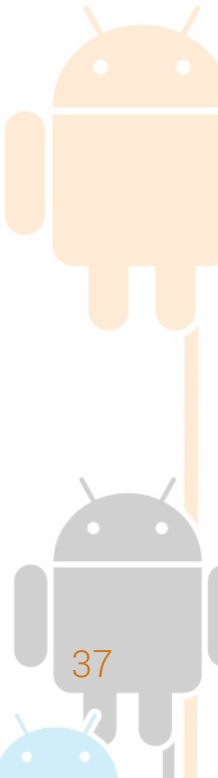
AVD Manager

- Un gestionnaire des émulateurs centralisé
- Configurateur
- Accessible depuis Eclipse

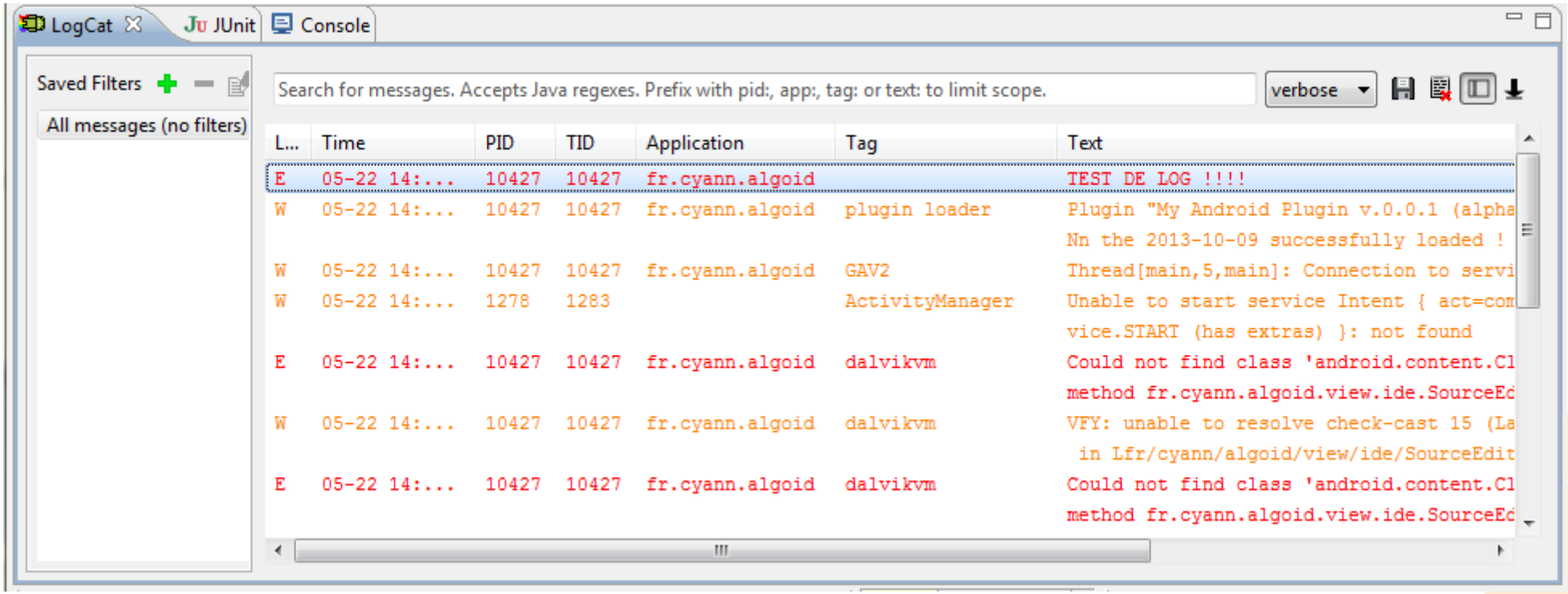


Android Debug Bridge (ADB)

- Outil en ligne de commande
- Un client/serveur (et service) dédié au développement sur Android
- Server : start-server, kill-server, devices
- Shell : shell, logcat
- Data : install, pull, push
- Et bien d'autres : http://developer.android.com/tools/help/adb.html

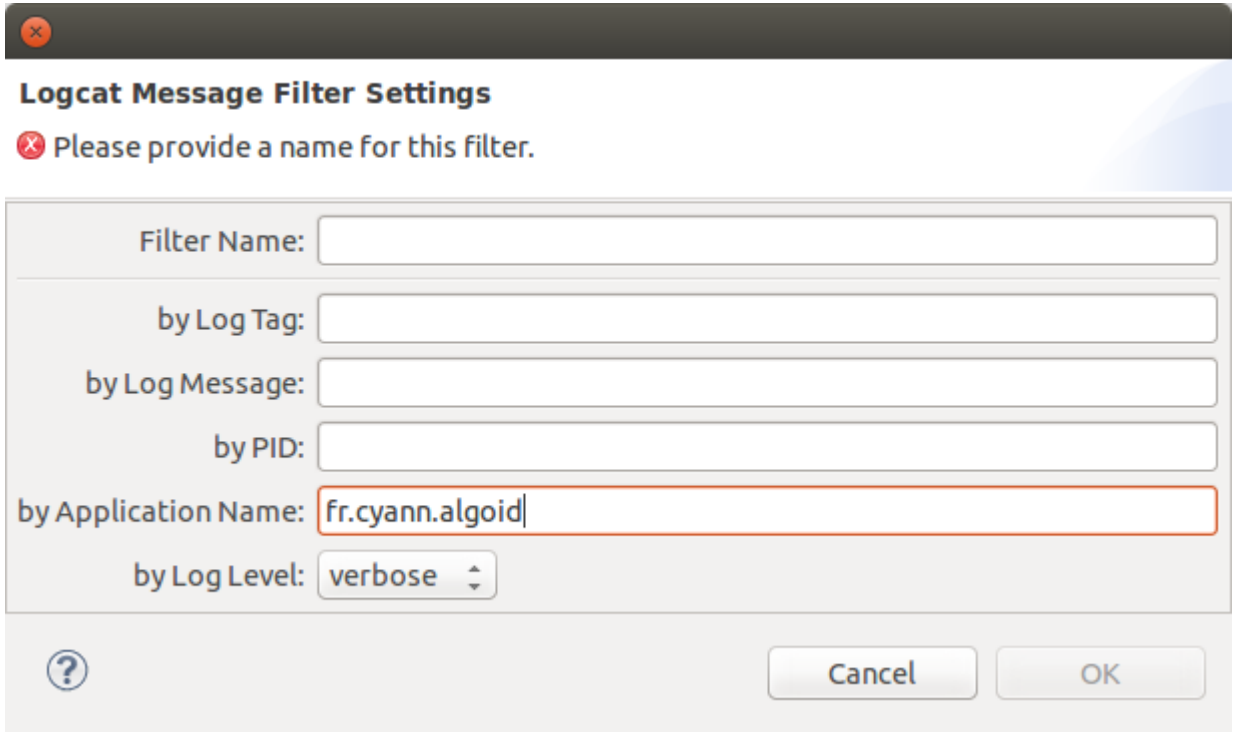


Logcat



- Un logger sur chaque device
- Accessible depuis ADT ou une app (aLogCat)

Logcat



- Filtre depuis Eclipse
- Recommandé

Fin

- Merci de votre attention
- Des questions ?