

Page web du module

https://sites.google.com/esp.sn/ifall/teachingsenseignements/I2-level-niveau-I2/I2_mobile



Introduction au développement Mobile

Ibrahima FALL

Ibrahima.Fall@esp.sn

Département Génie Informatique, Ecole Supérieure Polytechnique

Université Cheikh Anta Diop de Dakar

BP 5085 Dakar-Fann, Sénégal

Avril 2019



Objectifs

■ Comprendre

- ☐ l'historique du développement mobile et les enjeux du monde du mobile
- ☐ les différentes formes de développement mobile
- ☐ le processus de création, de déploiement, et de mise à disposition et d'utilisation des applications mobiles



Plan de l'enseignement

- Introduction générale
 - Éléments de base en développement mobile
- Développement natif mobile
 - L'exemple d'Android



Chapitre 1

Introduction Générale

Terminaux mobiles.
Caractéristiques du monde
mobile. Opportunités offertes.
Types d'applications mobiles.

Vous avez dit « développement mobile »?

- Développement d'applications pour des terminaux mobiles
- Terminal mobile
 - Possibilité d'être déplacé facilement
 - Connectivité étendue à Internet
 - Utilisation sans faire recours à une prise électrique
 - [Possibilité d'être transporté dans une poche et d'être utilisé avec une seule main]*

*Propriétés bien discutables

Vous avez dit « développement mobile »?

■ Exemples de terminaux mobiles

- ☐ Téléphones de base (feature phone), téléphones intelligents (smartphones), superphone, ...
- ☐ Tablettes,
- ☐ Téléviseurs,
- ☐ Montres, baladeurs musicaux (iPod Touch, ...)
- ☐ Consoles de jeux portables (Sony PSP, Nintendo DS, ...)
- ☐ Récepteurs GPS
- ☐ ...

Ecueils spécifiques au monde mobile

■ Des contraintes matérielles

☐ Résolution et/ou taille physique petites de l'écran

- Densité de pixels (en ppp, ou ppi) plus importante nécessaire

- ☐ Du que l'appareil est utilisé plus près (environ 20 à 30 cm)

☐ Ecran d'orientation plus portrait que paysage

☐ Eclairages moins confortables

☐ Système d'entrée de texte au mieux malaisé, voire pénible

Ecueils spécifiques au monde mobile

- Des contraintes matérielles (suite)
 - Système de pointage (curseurs, mini-joystick, tactile) à faible niveau de précision
 - Durée de vie limitée de la batterie
 - Plus faible puissance de calcul (CPU)
 - Capacité de stockage limitée, surtout en RAM
 - Réseau (souvent téléphonique) à faible débit et à coût élevé
 - Est-ce toujours vrai ?

Ecueils spécifiques au monde mobile

■ Un contexte différent

□ Etat d'esprit de l'utilisateur variable

- Pressé, à la recherche d'un passe-temps, focalisé, distrait, etc.

□ Utilisateur très souvent en situation de mobilité

- Accès rapide à une information contextuelle
- Possibilité de trouver ladite information sans être fortement concentré sur l'application utilisée
- Accès à l'interface avec un minimum d'interaction manuelle, idéalement avec une seule main

Ecueils spécifiques au monde mobile

- Un contexte différent (suite)

- Relations sociales virtualisées

- Besoin accrue de fonctionnalités de partage

- Les réseaux sociaux sont le plus utilisés via des terminaux mobiles

- ...

Ecueils spécifiques au monde mobile

■ Un domaine fragmenté

□ Beaucoup de types de terminaux

- Téléphones de base (feature phones), téléphones intelligents (smartphones), superphones, ...
- Tablettes,
- Téléviseurs,
- Montres, baladeurs musicaux (iPod Touch, ...)
- Consoles de jeux portables (Sony PSP, Nintendo DS, ...)
- Récepteurs GPS

Ecueils spécifiques au monde mobile

■ Un domaine fragmenté (suite)

□ Beaucoup de navigateurs

- Plus nombreux et avec des mises à jour moins fréquentes

- Opera Mobile, Opera Mini, NetFront, FireFox Mobile, IE Mobile, Obigo, Bolt, Skyfire, UC Browser, Vision Mobile, Ovi Browser, ...

- ➔ Une liste d'émulateurs est maintenue par Mobiforge pour les tests

- <http://mobiforge.com/emulators/page/mobile-emulators>

Opportunités offertes par la plate-forme mobile

■ Disponibilité

- Plus de possesseurs de téléphones portables que d'ordinateurs
- Les utilisateurs de téléphones portables s'en séparent rarement
- Toujours allumés et prêts à l'emploi, requièrent peu de maintenance
- Peuvent être connectés partout, immédiatement, avec un minimum d'intervention de l'utilisateur

Opportunités offertes par la plate-forme mobile

■ Disponibilité (suite)

□ C'est ainsi que les utilisateurs peuvent être atteints dans presque toutes les situations

- Transports
- Lieux et situations d'attente
- Entre amis, en famille, etc.
- La chambre, la cuisine
- Un lieu touristique
- Un magasin
- etc.

Opportunités offertes par la plate-forme mobile

■ Personnalisation

□ Les téléphones sont des objets très personnels

- Possibilité pour l'utilisateur d'être identifié de manière permanente
- Stockage de grands nombres d'informations personnelles (carnets d'adresses, messages, agendas, etc.)
- Adaptation de l'expérience utilisateur au contexte d'emploi (avec l'aide des capteurs abrités)

Opportunités offertes par la plate-forme mobile

■ Innovation

□ Une montée en puissance accrue des composants matériels

■ Interactions tactiles

■ Détection de mouvements et déplacements

□ Exemple: secouer pour choisir un élément aléatoire

■ Interactions vocales

□ Reconnaissance, synthèse, intégration de la voix au Web, ...

■ Un œil sur le monde

□ Témoignages vidéo et photo, ...

■ Géolocalisation

□ Possibilité d'accès par programmes à des information de géolocalisation, ...

Types d'applications mobiles

- 3 scénarii de développement possibles
 - ☐ Web
 - ☐ Natif
 - ☐ Hybride

Applications Web

- Codées en utilisant les technologies Web
 - En adaptation continue
 - WAP, XHTML MP, XHTML Basic, ..., HTML5



Avantages du Web mobile

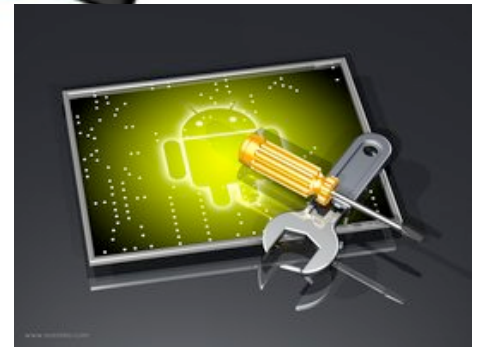
- Tendence multi-écrans et disponibilité sur toutes les plateformes
- Plusieurs standards sont en cours de réalisation au W3C pour l'accès aux composants des terminaux mobiles
- Simplicité technologique
 - On y fait du web

Inconvénients du Web mobile

- Absence de visibilité des applications codées en HTML au sein des kiosques de téléchargement
- Manque de fluidité
 - Applications moins réactives
- La compatibilité WebKit entre chacune des plateformes ne serait pas au rendez-vous

Applications natives

- Développées avec des langages spécifiques à chaque système d'exploitation
- Exemple
 - Objective-C (iOS)
 - Java (Android)
 - C# (Windows Phone)



Avantages du natif

- Utilisation d'un langage spécifique à chaque plateforme
 - Accès facile aux ressources/informations du mobile (IMEI, modèle, touches, réseaux mobiles, ...)
 - Possibilité d'une meilleure ergonomie qualitative et d'une intégration des fonctionnalités haut de gamme
 - Réalité augmentée, cartographie, jeux, ...
- Bonnes visibilité et fluidité des applications
 - Facilement repérables au sein des plateformes de téléchargement
- Plus réactives

Inconvénient du natif

- Coûts de développement cumulés souvent jugés exorbitants.
 - Une application nécessite plusieurs compétences

Applications hybrides

- Développées majoritairement en utilisant les technologies Web et où la partie native se différenciera selon les plateformes ciblées
- Remarque
 - Les outils de développement cross-platform commencent à percer

Avantage des applications hybrides

- Bénéficier des avantages des applications natives sans développer l'intégralité du produit dans le langage de chaque plateforme
- Diminuer les coûts d'équipement en mutualisant les vues web
- Etre visible sur les stores des mobiles.

Chapitre 2

Introduction à Android



Qu'est-ce que c'est? Environnements de développement. Principaux éléments d'une application Android

Un peu d'histoire ...

- À l'origine, Android Incorporated, nom d'une PME américaine
 - créée en 2003
 - Objectif
 - Développer un système d'exploitation mobile plus intelligent qui devait permettre à l'utilisateur d'interagir avec son environnement.
 - puis rachetée par Google en 2005
 - Volonté de s'introduire sur le marché des produits mobiles.
 - Concurrence: Symbian, Windows phone

... un peu d'histoire ...

- Chaque constructeur concevait un SE spécifique pour son téléphone
 - aucune base commune entre les SE de constructeurs différents
 - Conséquence
 - Développement mobile orienté matériel
 - Basé sur des langages de bas niveau comme C et C++

... un peu d'histoire ...

- Janvier 2007: la révolution iPhone
 - SE = iOS (iPhone OS)
 - Téléphone capable d'aller sur Internet, de lire des vidéos, etc.
 - Un désastre pour les concurrents d'Apple
 - Ils estiment qu'il leur aurait fallut des années de pour atteindre l'iOS

... un peu d'histoire ...

- Novembre 2007: naissance de l'Open Handset Alliance (OHA)
 - Engagement à l'ouverture, vision partagée du futur, plan de mise en oeuvre concret
 - Fournir la plateforme nécessaire à un développement novateur, plus rapide, de meilleure qualité et sans coût de licence
 - <https://www.openhandsetalliance.com/>



... un peu d'histoire ...

■ L'OHA

- Initialement 35 entreprises évoluant dans l'univers du mobile, dont Google
- Aujourd'hui plus de 80 membres

... un peu d'histoire ...

- Membres de l'OHA

- Fabricants de matériels

- Alcatel, Asus, Dell, Haier, HTC, Huawei, LG, Lenovo, Toshiba, ...

- Opérateurs mobiles

- Bouygues, China Mobile, T-Mobile, Telefonica, Vodafone, ...

- Editeurs de logiciels

- Google, Cooliris, Ebay, Motoya, NXP, Nuance, Skypop, ...

... un peu d'histoire ...

- Membres de l'OHA (suite)
 - Fabricants de semi-conducteurs
 - Intel, Nvidia, AKM, Broadcom, Marvell, Mediatek, ...
 - Sociétés commerciales
 - Aplix, Accentur, noser, TAT, Wind river, ...

... un peu d'histoire ...

- Logiciel phare de l'OHA
 - **ANDROID**
 - Pas la seule activité
- Octobre 2008: premier téléphone Android
 - **T-Mobile G1, USA**
- Déjà en 2012
 - **300 millions de terminaux compatibles Android**
 - Produits par 39 fabricants dans plus de 123 pays et reconnus par 231 opérateurs téléphoniques !!!

Les principes de base

- Ils encouragent à développer sous Android
 - Open source
 - (presque) gratuit
 - Facile à développer et à vendre
 - API à disposition, Play Store
 - Flexible
 - SE très portable et pour beaucoup de types de terminaux
 - Même des fours micro-ondes
 - ...

Android n'est pas

- Une implémentation de Java ME
 - Code écrit en Java mais non exécuté par une JVM comme pour JME
 - Non plus le bytecode Java n'est pas nativement exécutable sous Android
- Un téléphone mobile
 - Plutôt un système conçu pour supporter de nombreux matériels
- La réponse de Google à l'iPhone
 - iPhone=matériel+plateforme logicielle propriétaire
- ...

Android est

- Une combinaison de 3 composants dépendants et nécessaires
 - Un SE open-source libre pour appareils mobiles
 - Une plateforme de développement open-source pour la création d'applications pour mobiles
 - Des équipements, en particulier des téléphones portables, qui exécutent le SE ainsi que les applications développées

Android comprend

- Un modèle de conception matérielle de référence
 - Décrit les capacités nécessaires au support de la pile logicielle
- Un SE Linux
 - Taillé sur mesure pour les appareils mobiles
 - Assure l'interface de bas niveau avec le matériel, la gestion de la mémoire et le contrôle des processus
- Des bibliothèques open-sources
 - Pour le développement d'applications

Android comprend

- Un moteur d'exécution et d'hébergement des applications
 - Machine virtuelle DALVIK et bibliothèques de base
- Des applications préinstallées essentielles
 - client de courrier électronique, SMS, contacts, calendrier, navigateur, musique, photos, prise photos et vidéos, calculatrices, horloge, ...

Android comprend

- Dans la majeure partie des cas, les appareils Android sont également livrés avec des applications Google propriétaires
 - Chrome, Client Google Play Store, app. Google Maps, client Gmail, client Google Talk, lecteur YouTube, ...
- Certains constructeurs ou opérateurs rajoutent eux aussi des applications sur les nouveaux appareils

Quelques avantages qu'offrent Android

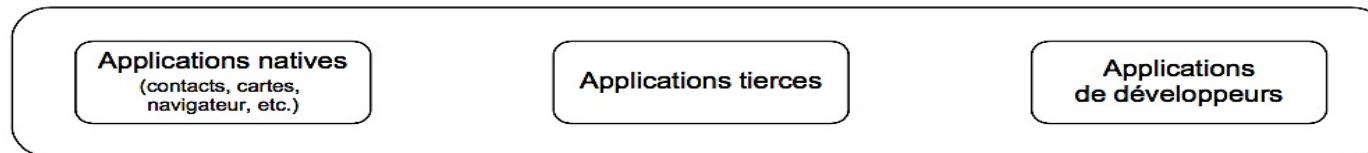
- Applications Google Maps
- Services d'arrière-plan et applications
- Données partagées et communications interprocessus
- Toutes les applications sont égales
 - Natives ou développées par des tiers

Introduction au kit de développement

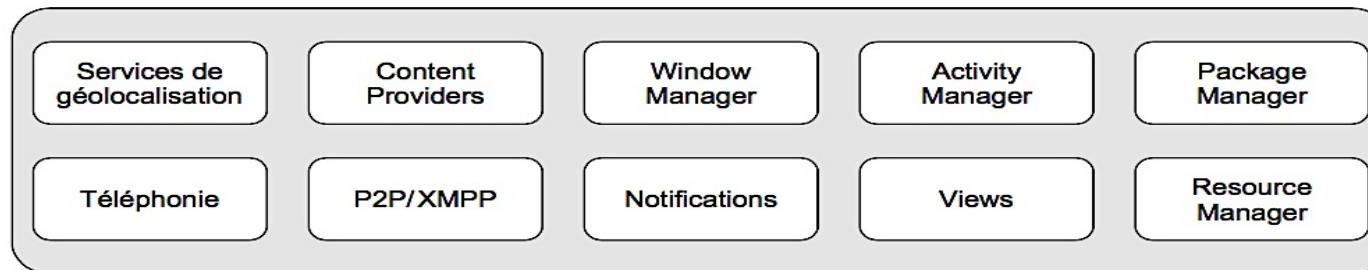
- Aperçu technique de la pile logicielle
- Les éléments constitutifs du SDK
- Les bibliothèques Android
- La machine virtuelle Dalvik

Aperçu de la pile logicielle

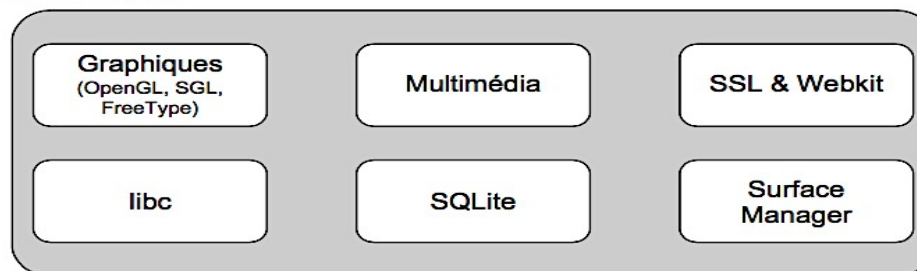
Couche applicative



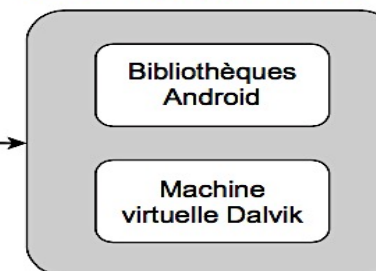
Framework applicatif



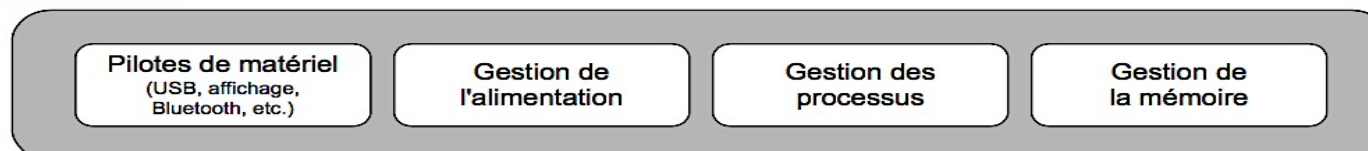
Bibliothèques



Moteur d'exécution Android

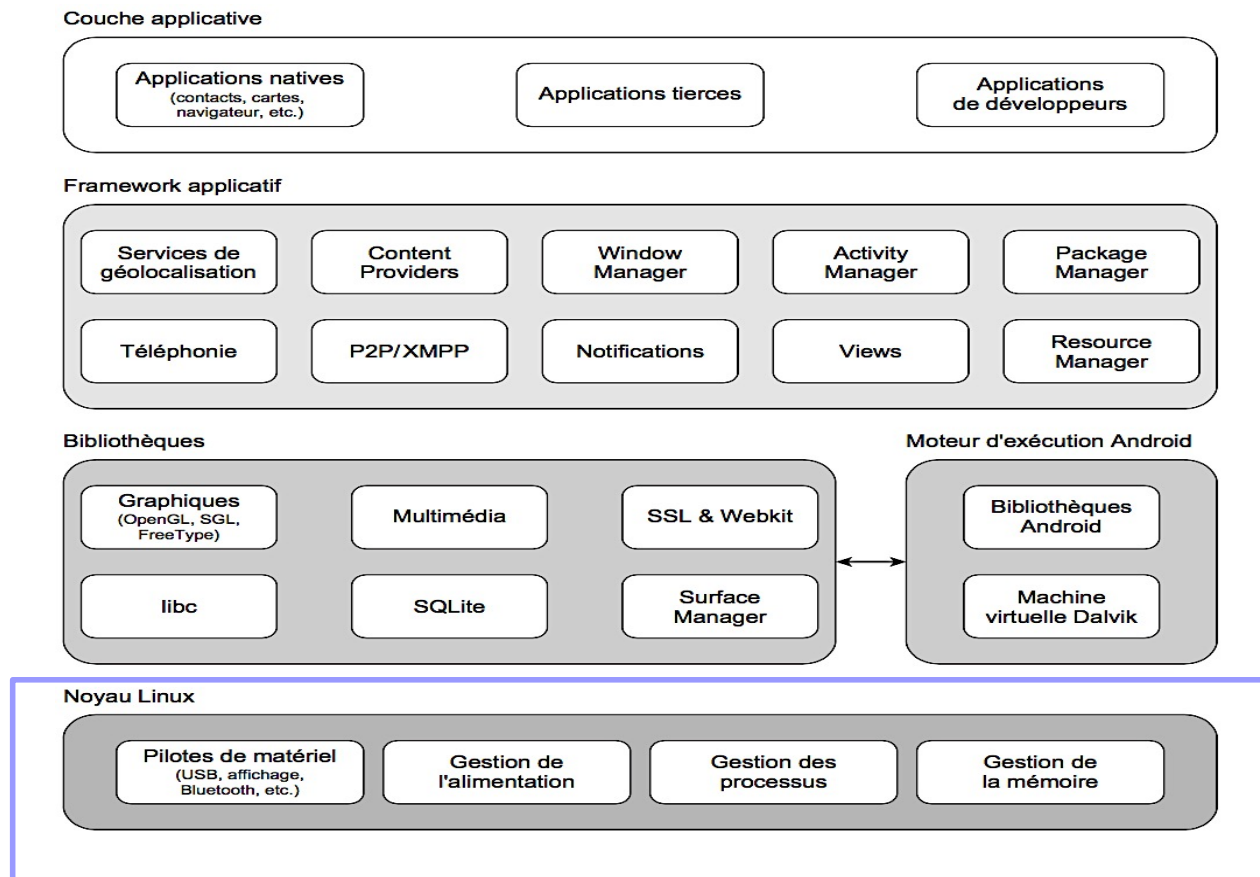


Noyau Linux



Aperçu de la pile logicielle

■ Le noyau Linux



Aperçu de la pile logicielle

- Le noyau Linux

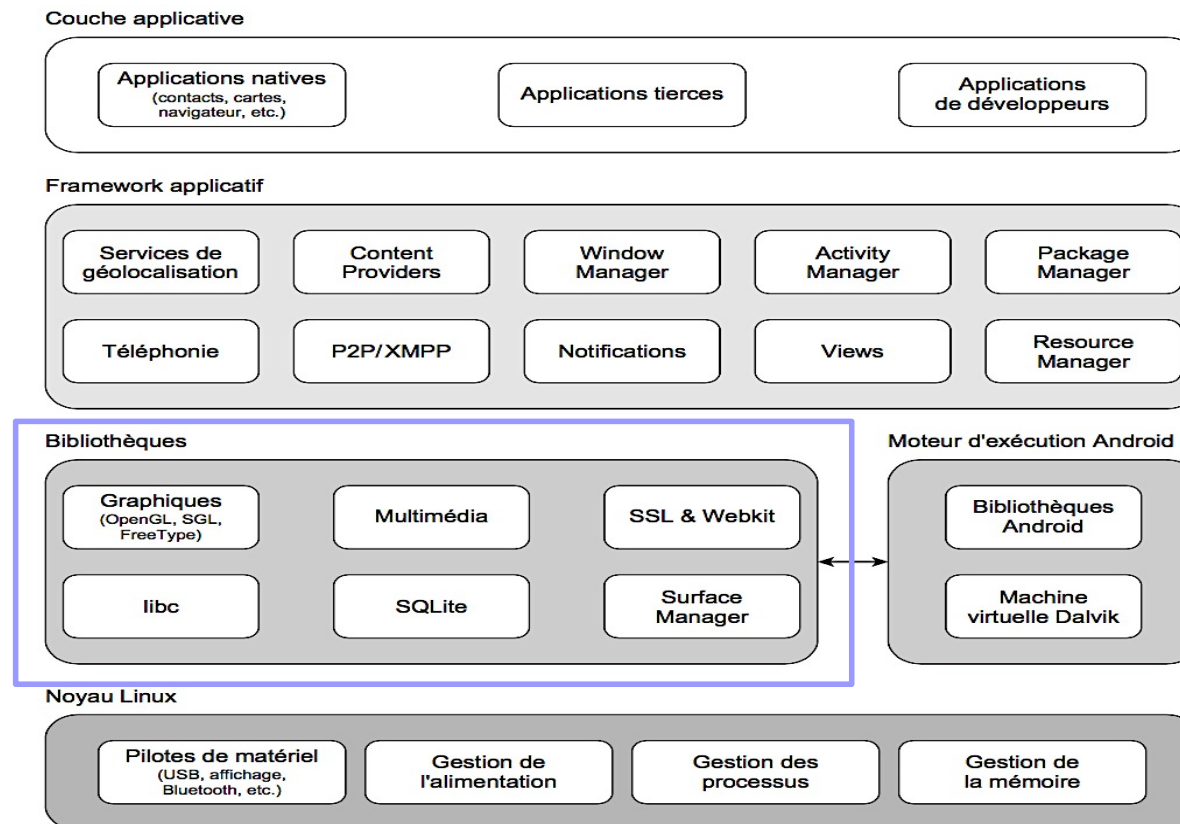
- Gere les services de base

- pilotes du matériel, gestion des processus et de la mémoire, réseau, gestion de l'alimentation

- Fournit une couche d'abstraction entre le matériel et le reste de la pile

Aperçu de la pile logicielle

■ Les bibliothèques



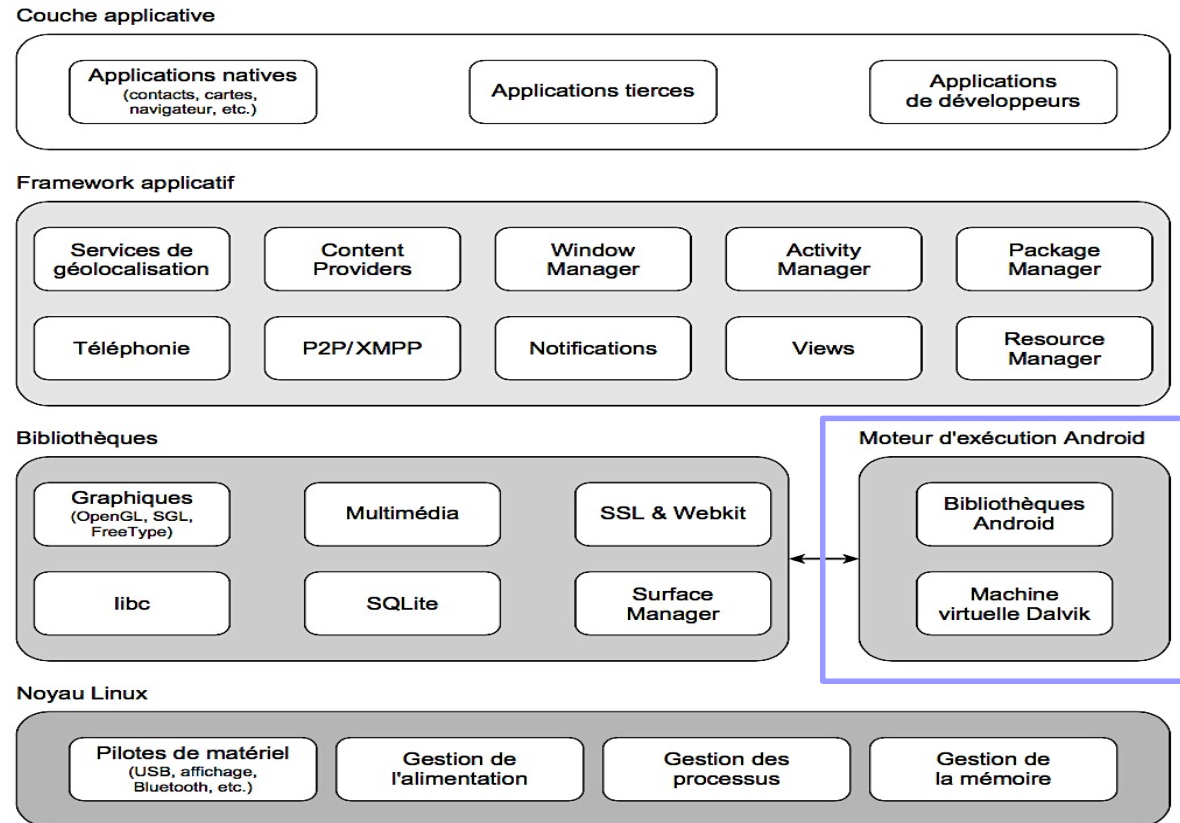
Aperçu de la pile logicielle

■ Les bibliothèques

- ☐ Bibliothèques C/C++ de base exécutées au-dessus du noyau: libc, SSL
- ☐ Multimédia
- ☐ Gestion de surface (affichage)
- ☐ Graphisme (SGL, OpenGL, etc.)
- ☐ Support natif des bases de données (SQLite) ;
- ☐ Intégration d'un navigateur web et la sécurité sur Internet: SSL, WebKit

Aperçu de la pile logicielle

■ Le moteur d'exécution



Aperçu de la pile logicielle

- Le moteur d'exécution
 - Fait d'un téléphone un téléphone Android
 - et non une implémentation mobile de Linux
 - Motorise les applications
 - Forme les bases du framework applicatif
 - avec l'aide des bibliothèques
 - Comprend essentiellement
 - les bibliothèques de base
 - la machine virtuelle Dalvik

Aperçu de la pile logicielle

- Le moteur d'exécution
 - Les applications sont écrites en Java
 - Mais sont exécutées par une machine virtuelle spécifique
 - Dalvik
 - Chaque application est exécutée dans un processus distinct dans sa propre instance Dalvik
 - Le moteur d'exécution Android qui englobe Dalvik gère la mémoire et les processus
 - Exemple: il arrête et tue les processus si nécessaire pour gérer les ressources

Aperçu de la pile logicielle

- Le moteur d'exécution
 - ...
 - La machine virtuelle Dalvik
 - Optimisée pour garantir qu'un appareil peut exécuter plusieurs instances de façon efficace
 - Les exécutables Dalvik (.dex), ont un format optimisé pour garantir une empreinte mémoire minimale
 - Dépend du noyau Linux pour les tâches de bas niveau (gestion des threads, de la sécurité, des processus, de la mémoire, etc.)
 - NB: La NDK (Native Development Kit) existe
 - Permet de créer de nouvelles bibliothèques
 - Donne accès à OpenGL
 - Code en C/C++

Aperçu de la pile logicielle

- Le moteur d'exécution

- ...

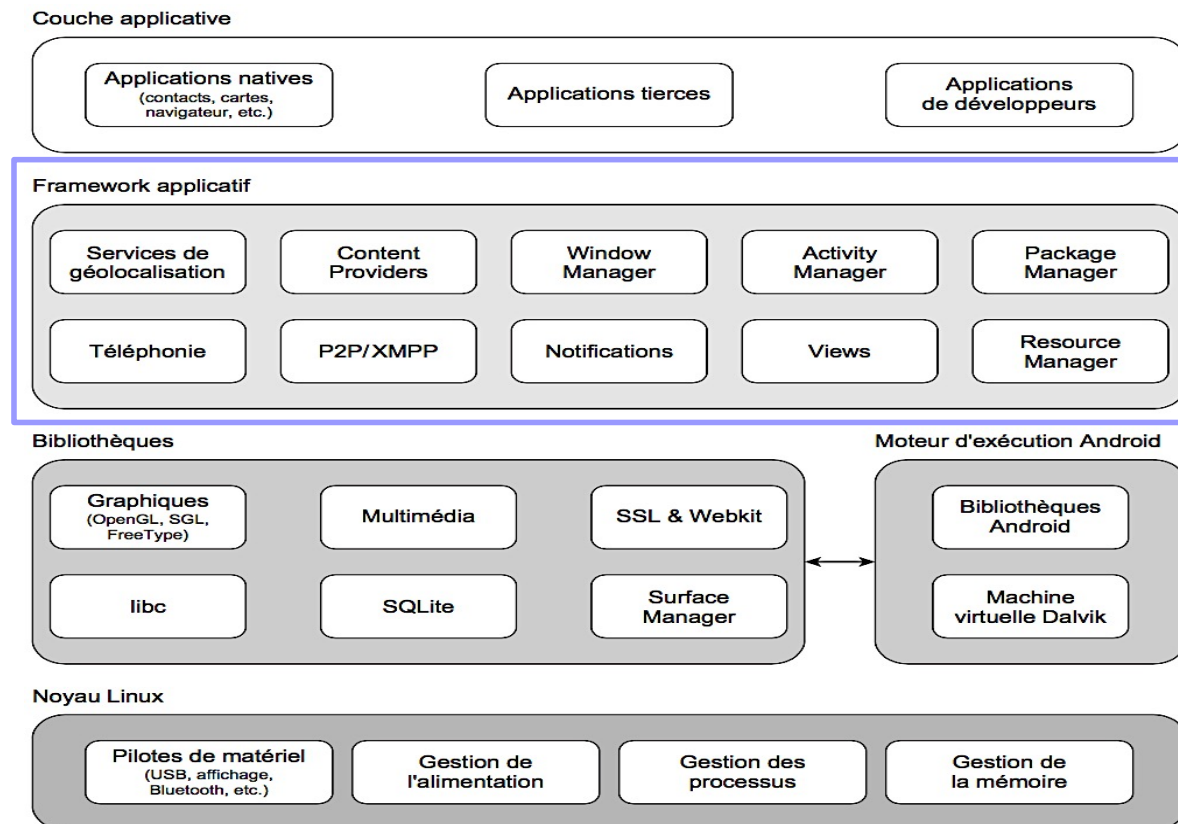
- Les bibliothèques de base

- Le code Android s'écrit en Java alors que Dalvik n'est pas une JVM

- Ce sont les bibliothèques de base qui fournissent la plupart des fonctionnalités disponibles dans les bibliothèques de base Java ainsi que dans les bibliothèques spécifiques à Android.

Aperçu de la pile logicielle

■ Le framework applicatif



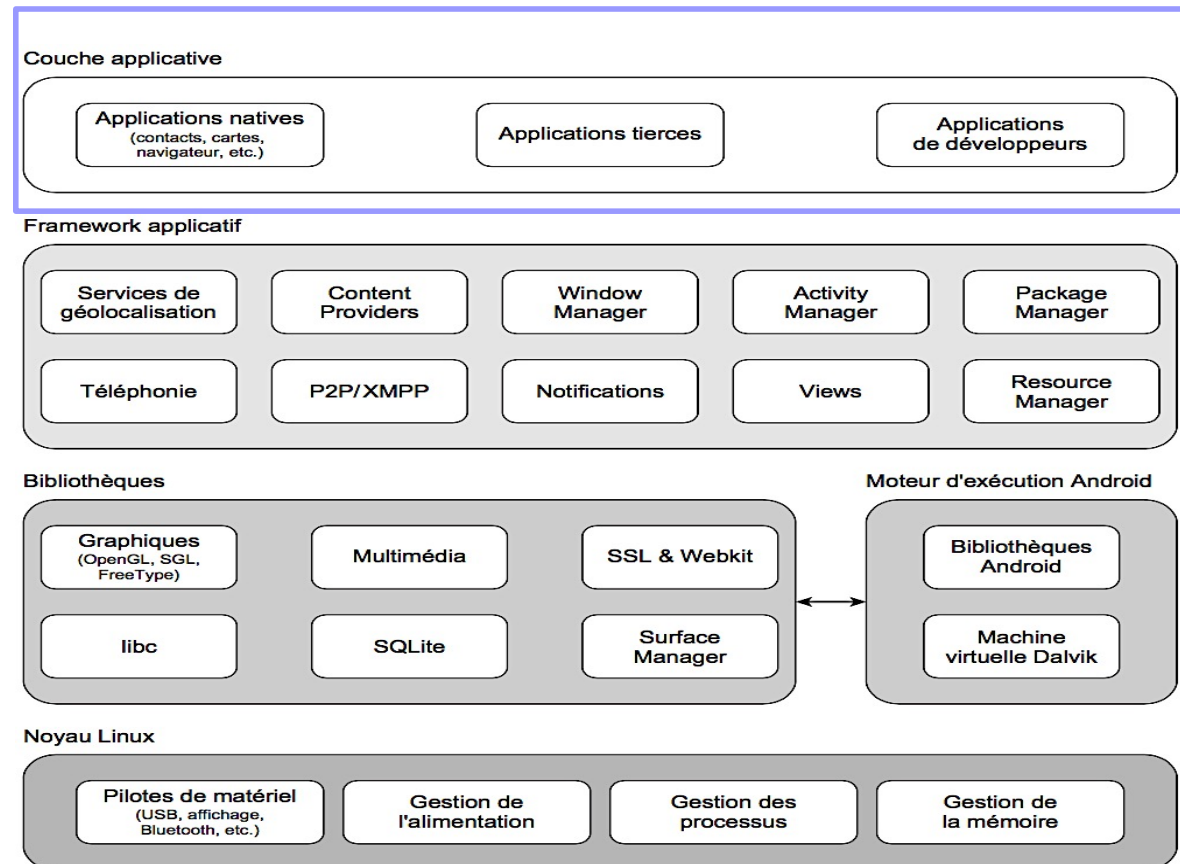
Aperçu de la pile logicielle

■ Le framework applicatif

- Fournit l'API pour le développement des applications Android
- Offre une abstraction générique pour l'accès au matériel
- Gère l'interface utilisateur ainsi que les ressources de l'application

Aperçu de la pile logicielle

■ La couche applicative



Aperçu de la pile logicielle

- La couche applicative
 - Accueille toutes les applications (natives ou tierces)
 - Est exécutée par le moteur d'exécution Android
 - Utilise les classes et services du framework applicatif

Éléments constitutifs du SDK

- Le SDK d'Android fournit ce qu'il faut pour produire des applications
 - Développer
 - Tester
 - Déboguer

Éléments constitutifs du SDK

- Les API Android
 - Donnent accès à la pile Android.
 - Utilisées par tout le monde
 - Pour développer des applications tierces comme natives
- Des outils de développement
 - Pour transformer le code source Android en applications exécutables

Éléments constitutifs du SDK

- Le Virtual Device Manager
 - Le gestionnaire d'appareils Android virtuels (AVD)
 - Simule une configuration matérielle
- L'émulateur Android
 - S'exécute au sein d'un AVD
 - Permet de voir à quoi ressemble une application et comment elle se comportera sur un vrai appareil Android.
 - Est indépendant de tout matériel
 - Fournit alors un meilleur environnement de test que n'importe quel terminal physique

Eléments constitutifs du SDK

- Une documentation complète
 - Une référence
 - Détaille ce qui est inclus dans chaque package et classe, et comment les utiliser
 - En plus de documenter le code, explique comment démarrer et donne des explications détaillées sur les fondamentaux du développement
- Des exemples de code
 - Le SDK propose une sélection d'applications exemples

Éléments constitutifs du SDK

- Un support en ligne
 - Android a rapidement généré une communauté vivante de développeurs
 - Les Google Groups
 - <http://developer.android.com/resources/community-groups.html>
 - <https://developer.android.com/support>
 - StackOverflow
 - <http://www.stackoverflow.com/questions/tagged/android>
 - etc.

Éléments constitutifs du SDK: quelques autres outils de développement

- DDMS
- AAPT
- ADB
- Logcat
- Dx
- Lint
- Hierarchy Viewer
- Etc.

Les bibliothèques Android

- On les découvre en développant
- Correspondent aux APIs qui donnent accès à la pile Android
- Permettent de gérer des
 - Activités, vues, contenus, etc.
- ...

Environnements de développements

- Environnements dédiés
 - Taillés sur mesure
 - Exemple: Android Studio
- Environnements génériques
 - Exemple: Eclipse
 - ADT (Adroid Development Tool)
 - un plugin développé et maintenu directement par des équipes de Google
 - Netbeans, etc.

TPs

- Mise en place d'un environnement de développement
- Développement de la première application

Composants essentiels d'une application Android

- Une application Android
 - formée de composants à faible couplage
- Un manifeste d'application
 - Est utilisé pour décrire chacun des composants et leurs interactions,
 - Permet de préciser
 - les métadonnées de l'application,
 - ses exigences matérielles et logicielles,
 - ses bibliothèques externes,
 - les permissions qu'elle exige,
 - etc.
 - Est défini en XML

Composants essentiels d'une application Android

- Briques de base de toute application Android
 - Activités, fragments, vues
 - Services
 - Fournisseurs de contenus
 - Intentions
 - Récepteurs de diffusion
 - Widgets
 - Notifications

Composants essentiels d'une application Android

■ Activités

☐ Couche de présentation

- Chaque écran est une activité

☐ Utilisent des vues ou des fragments pour former les UI qui afficheront l'information et répondront aux actions de l'utilisateur

☐ Equivalentes aux Form dans le développement classique

■ Vue

☐ Permettent de créer une UI

☐ Exemples: boutons, champs, etc.

Composants essentiels d'une application Android

■ Fragments

- Permettent de diviser une activité en composants réutilisables et autonomes,
- Avec pour chacun son propre cycle de vie et sa propre UI
- Facilitent la création d'interfaces dynamiques et souples
 - pouvant s'adapter à un large éventail de types de terminaux, de tailles et de densités d'écrans différents

- Chaque fragment est un module indépendant et est lié à l'activité dans laquelle il se trouve

Composants essentiels d'une application Android

■ Services

- S'exécutent sans UI
- Mettent à jour les sources de données et les activités visibles
- Déclenchent des notifications et diffusent des intentions
- Servent à exécuter des traitements continus ne nécessitent pas d'intervention de l'utilisateur
 - recherches sur le réseau
 - tâches devant continuer à s'exécuter même lorsque les activités ne sont pas actives ou visibles

Composants essentiels d'une application Android

- Fournisseurs de contenus
 - Sources de données partageables
 - Gèrent et stockent les données de l'application
 - Intéragissent généralement avec des bases de données SQL
 - Meilleur moyen de partager des données entre applications
 - Les appareils Android disposent de plusieurs fournisseurs de contenus natifs qui exposent des bases de données très utiles
 - Exemples
 - médias, contacts, etc.

Composants essentiels d'une application Android

■ Intentions

- Cadre de communication inter applications par passage de message
- Peuvent servir de lancer/arrêter des activités et services
 - ou demander l'exécution d'une action sur des données particulières
- Différentes catégories
 - Intentions explicites
 - Intentions implicites
 - Intentions à diffusion

Composants essentiels d'une application Android

- Récepteurs de diffusion
 - Consommateurs des messages diffusés par les intentions
 - S'ils répondent à certains critères spécifiques
 - Peuvent démarrer de façon automatique une application pour répondre à une intention entrante
 - Idéaux pour créer des applications basées sur des événements

Composants essentiels d'une application Android

■ Widgets

- Composants visuels qui sont généralement ajoutés à l'écran d'accueil
 - interactifs et dynamiques
- Sont une variante des récepteurs de diffusion

Composants essentiels d'une application Android

■ Notifications

- permettent d'envoyer un signal aux utilisateurs
 - sans dérober le focus ni interrompre l'activité en cours
- Moyen privilégié d'attirer l'attention de l'utilisateur à partir d'un service ou d'un récepteur de diffusion
- Exemple
 - Lorsqu'un appareil reçoit un message texte ou un appel entrant, il vous alerte en clignotant, en émettant un son, en affichant une icône ou un message
 - Ces mêmes événements peuvent donc être déclenchés depuis une application tierce

Éléments constitutifs d'une application Android

- L'architecture d'Android encourage la ré utilisation de composants
 - permettant de publier et de partager les activités, les services et les données avec d'autres applications
 - avec un accès géré par des restrictions de sécurité à mettre en place

Services applicatifs essentiels à une application Android

- Sont les pierres angulaires qui fournissent le framework utilisé pour les applications
- Gestionnaire des activités et gestionnaire des fragments
 - Contrôlent le cycle de vie des activités et fragments, notamment la gestion de la pile des activités
- Vues
 - Utilisées pour construire les UI des activités
- Gestionnaire des notifications
 - Fournit un mécanisme cohérent et non intrusif pour envoyer des signaux aux utilisateurs

Services applicatifs essentiels à une application Android

- Fournisseurs de contenu
 - Permettent aux applications de partager des données
- Gestionnaire de ressources
 - Permet à des ressources autres que du code, comme les chaînes de caractères ou les graphiques, d'être externalisées
- Intentions
 - Offrent un mécanisme de transfert de données entre les applications et leurs composants