

# Cours 2. Les Systèmes d'Exploitation Mobiles

---

[O.R. Merad Boudia](#)

Université d'Oran 1, Ahmed Ben Bella

M1 GBM : 2023/2024

# **Partie 1: Architecture**

## **Partie 2: Présentation des OS mobiles**

# Partie 1: Architecture

Partie 2: Présentation des OS mobiles

# Introduction

- ❑ Le monde devient de plus en plus mobile, rapide, connecté, et sécurisé.
- ❑ Les gens s'attendent à se déplacer facilement avec leurs appareils mobiles, en gardant des communications étroites avec leurs partenaires et la famille, profiter de la polyvalence des modèles d'utilisation et du contenu infini, et ceci sans se soucier de l'appareil et la gestion des données.
- ❑ Tout cela met des exigences sur les appareils mobiles dont le système d'exploitation mobile est l'âme.

# OS mobile : Spécificités

□ Un OS mobile doit assurer un fonctionnement efficace d'un appareil mobile en considérant ses ressources limitées.

➤ Physiquement limité,

- Dispositif alimenté par batterie.
- Petits écrans de différentes formes, tailles et résolutions.
- Mémoire (nettement inférieure à celle d'un laptop).
- Espace de stockage.

➤ Fonctionner dans l'incertitude,

- Un OS mobile doit fournir des méthodes robustes pour la gestion des connexions et faire face à des interruptions de service.

➤ Sécurité.

# Architecture : évolution

- ❑ L'architecture des OS mobiles a connu une évolution en trois phases:
  - ✓ OS sur PC
  - ✓ OS embarqué
  - ✓ OS orientée Smartphone
  
- ❑ Tout au long du processus, l'architecture de l'OS mobile est passée du complexe au simple à quelque chose entre les deux. Le processus d'évolution est naturellement motivé par les progrès technologiques dans le matériel, les logiciels et l'Internet.

# Evolution (1/3)

## ❑ Matériel (Hardware):

- ❑ Réduction de la taille des microprocesseurs.

## ➤ OS mobiles traditionnels (PDA par exemple):

- Impossibilité d'avoir à la fois des appareils petits et une capacité de traitement importante,

- L'OS des PDA ne supportait pas le traitement multitâche et le graphisme en 3D,

- Des fonctionnalités non disponibles comme les capteurs, écran tactile, etc.

# Evolution (2/3)

## ❑ Logiciels (Software):

- Sur un PC, le logiciel est principalement axé sur la productivité de l'utilisateur, où le clavier et la souris qui ont des entrées précises sont essentiels.
- Le logiciel sur PDA, permet à l'utilisateur de gérer les données personnelles telles que des contacts, e-mails, et ainsi de suite.
- Les OS mobiles n'étaient pas conçus pour une bonne réactivité avec une interface utilisateur riche comprenant à la fois l'écran tactile et d'autres capteurs.

# Evolution (3/3)

## □ Internet:

- Avec le développement d'Internet, en particulier après le Web 2.0, il y a une quantité considérable d'informations dans le réseau en vue d'être recherchée, organisée, exploitée, et apportée aux utilisateurs,
- Les gens sont impliqués dans le développement, y compris la contribution de l'information, le développement d'applications, et les interactions sociales,
- Les OS mobiles ne peuvent pas être auto-suffisants, ils doivent être des systèmes ouverts.

# Modèles d'utilisation (1/2)

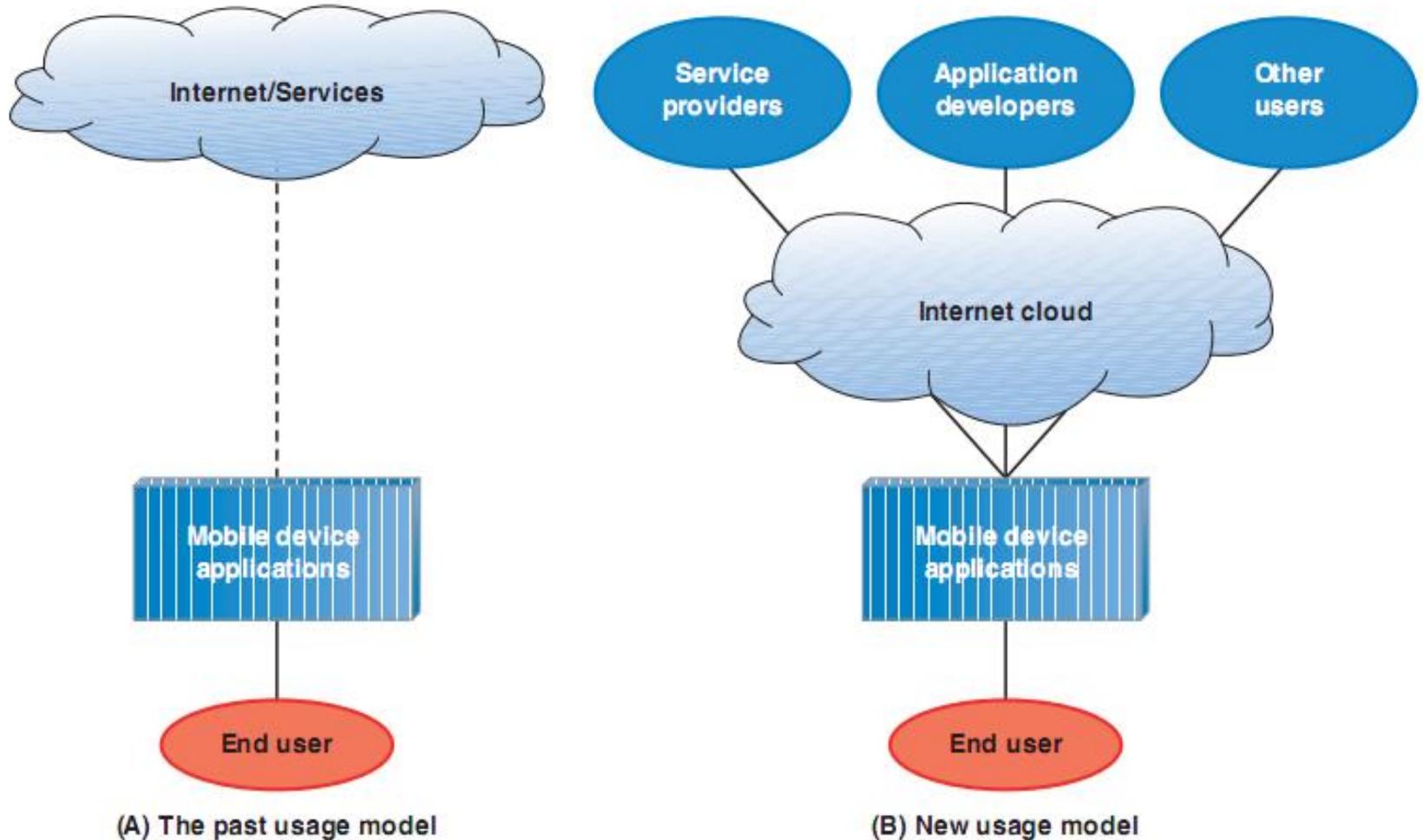
## ❑ Anciens appareils mobiles:

- ✓ Un utilisateur exécute la plupart des applications propres à l'appareil pour la gestion des données, les jeux et occasionnellement il navigue sur Internet des pages Web statiques ou accède à des services spécifiques comme l'e-mail.

## ❑ Nouveaux appareils mobiles:

- ✓ L'appareil est pratiquement un portail à divers modèles d'utilisation. Toutes les parties concernées telles que les fournisseurs de services, les développeurs d'applications, et d'autres utilisateurs contribuent en permanence et interagissent d'une manière continue à travers l'appareil, avec son propriétaire.

# Modèles d'utilisation (2/2)



(Source: Intel Corporation)

# Evolution

## ❑ Les axes majeurs:

- ✓ Hardware
- ✓ Software
- ✓ Internet

## ❑ Mais aussi (en conflit):

- ✓ L'expérience utilisateur (UX)
- ✓ La durée de vie de la batterie
- ✓ Le Cloud
- ✓ La sécurité
- ✓ L'open source

# Evolution : Axes en conflit (1/3)

## □ L'expérience utilisateur (UX) et la batterie:

- Pour obtenir une meilleure réactivité,
- ✓ Toutes les ressources matérielles doivent être disponibles pour le système afin d'exploiter leur meilleure capacité.
- En même temps, pour conserver la batterie,
- ✓ Les composants matériels doivent être inactifs le plus souvent possible.

*Il faut trouver un compromis (trade-off), un équilibre.*

# Evolution : Axes en conflit (2/3)

## □ La sécurité et l'open source :

- Un système ne voudrait pas exposer toutes ses fonctionnalités à des entités externes,
  - ✓ Problèmes de sécurité.
- D'autre part, sans exposer assez d'API,
  - ✓ Il est impossible pour les développeurs de créer des usages innovants.

# Evolution : Axes en conflit (3/3)

## ❑ Le Cloud et la sécurité:

- ❑ Comme de plus en plus de services et d'applications sont disponibles à partir du Cloud, il est naturel de considérer un modèle de dispositif qui fait confiance au Cloud.
- ✓ Mais jusqu'à aujourd'hui, un tel modèle a encore des défis techniques en matière de sécurité.

# Les processeurs « mobiles »

- ❑ L'architecture ARM (Advanced RISC Machine) est une architecture de processeur 32-bit et 64-bits RISC développée par ARM Limited.
  - ✓ Largement utilisée dans les systèmes embarqués.
- ❑ En raison de leurs caractéristiques d'économie d'énergie, les processeurs ARM dominent le marché de l'électronique mobile, où une faible consommation d'énergie représente un objectif critique de conception...
- ❑ Aujourd'hui, la famille ARM, représente environ 75% de l'ensemble de CPUs embarqués.

[Source](#)

# Les versions de ARM

Composant ARM	Génération	Exemples d'application	Introduction
ARM1	ARMv1	Premiers ordinateurs d'Acorn	1984
ARM2	ARMv2	Ordinateurs Archimedes d'Acorn	1987
ARM6	ARMv3	MessagePad d'Apple	1994
ARM7TDMI	ARMv4	GameBoy Advance, iPod	2001
ARM9E	ARMv5	Nintendo DS, Nokia N-gage	2004
ARM11	ARMv6	iPhone, iPhone3G, iPod Touch	2007
Cortex-A8	ARMv7	Palm Pre, iPhone3Gs-4, GalaxyS	2009
Cortex-A9	ARMv7	iPhone4s, GalaxyS2-S3, SonyXperiaU	2010
Cortex-A15	ARMv7	GalaxyS4-S5, Nexus	2012
Cortex-A53,A57, Cyclone	ARMv8	GalaxyS6-S7-S8, iPhone6-7-8-X	2015-2017
Cortex-X3, A715, Cortex-A710, Cortex-A510	ARMv8.5-A ARMv9-A	GalaxyS23, iPhone 14-14Pro	2023

Partenaires de ARM : (Qualcomm, Apple, Samsung, MediaTek, etc.)

[Source](#)

# La gestion de l'énergie

- ❑ La gestion de l'énergie a toujours été un défi majeur pour les concepteurs des OS mobiles et le sera encore plus à l'avenir,
  - Les demandes de l'énergie sont en augmentation rapide puisque les applications qui sont développées sont de plus en plus gourmandes.
- ❑ Cependant, la croissance de la capacité de la batterie ne pourrait jamais tenir dans l'intervalle,
  - Développement très lent dans les technologies des batteries.
- ❑ Une approche holistique (globale) doit être employée pour y remédier.

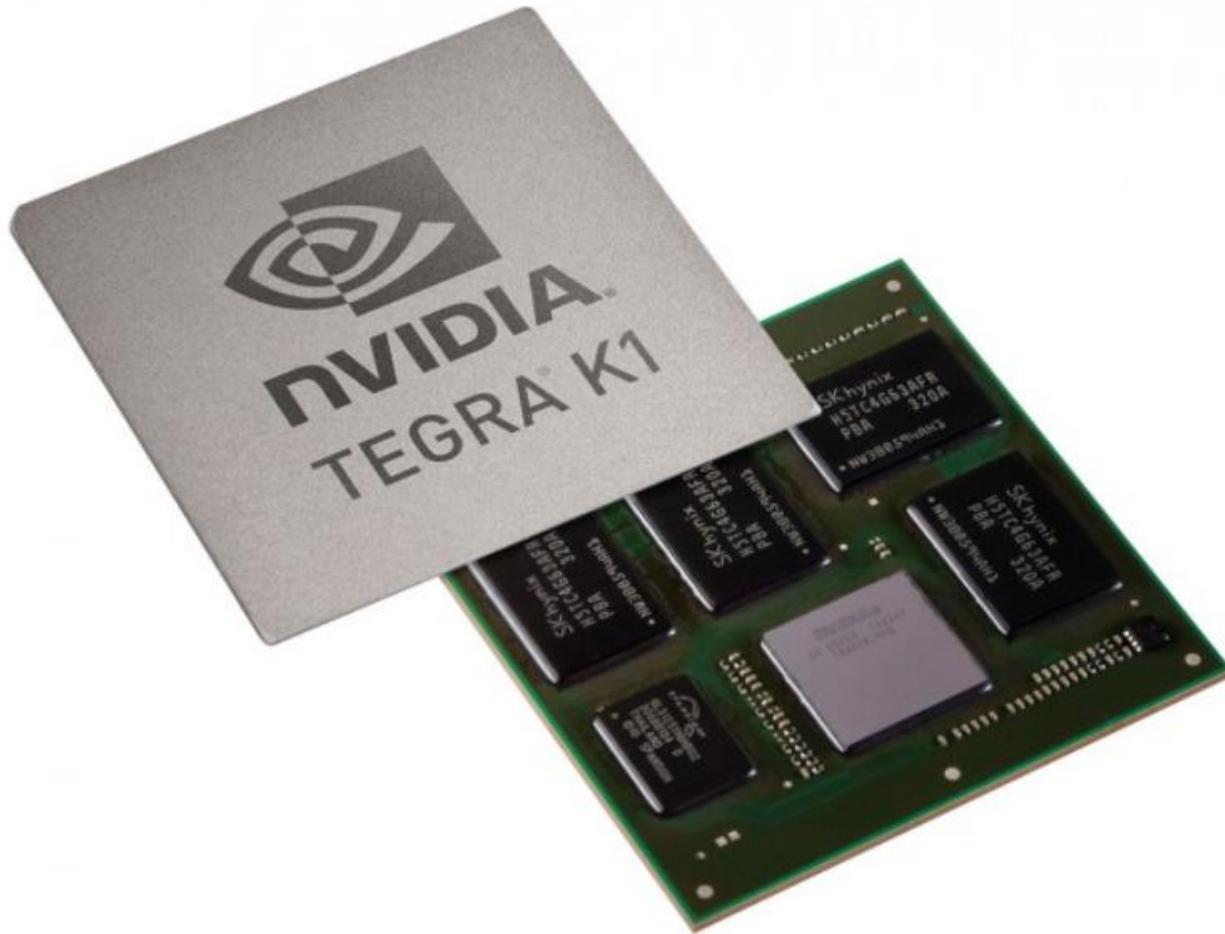
# Gestion de l'énergie du CPU

- Le processeur est considéré comme le consommateur le plus important de la puissance totale de l'appareil.
- ❖ Les processeurs modernes supportent la fonction DFVS « Dynamic Frequency and Voltage Scaling », une adaptation dynamique de tension et de fréquence du processeur.
  - ✓ Permet d'économiser une quantité importante de l'énergie du processeur alors qu'il est actif.

# Gestion de l'alimentation de l'appareil

- ❑ Un mécanisme a été mis en place pour gérer la puissance des dispositifs d'E/S lors de l'exécution.
- Les GPUs « Processeurs graphiques » modernes commencent à supporter une fonctionnalité «DFVS» similaire à celle trouvée sur les CPUs.
  - ✓ Exemple: Tegra K1, Tegra X1 de NVidia
- Des technologies comme panneau PSR (Panel Self Refresh) économise une quantité importante d'énergie pendant que l'image est statique.
  - ✓ L'idée est de désactiver le GPU quand l'image est fixe, ce qui permet de diminuer la consommation globale du Smartphone de 26 %

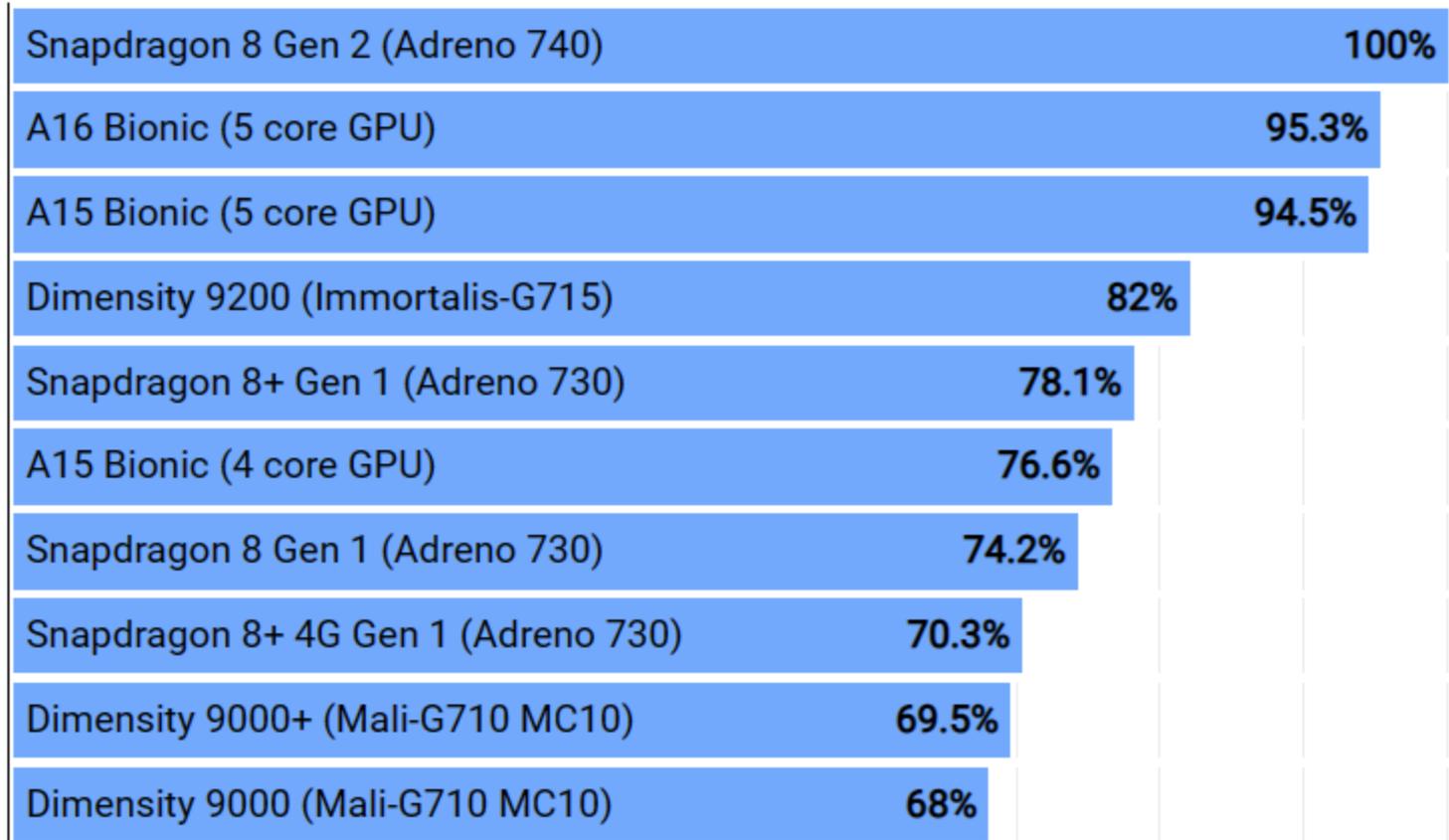
# La Tegra K1 de NVidia



*En gris (en bas) le couple CPU/GPU, en noir (grands rectangles) les puces de mémoire vive et en noir (petit rectangle) une ROM de NAND flash*

[Source](#)

# Top 10 des GPUs (03/2023)



[Source](#)

# Exemples

Smartphone	CPU	GPU
Iphone 14 Pro, Pro Max	A16 Bionic (5 Core GPU)	Intégrée
Iphone 14, 14 plus	A15 Bionic (5 Core GPU)	Intégrée
Samsung Galaxy S23	Snapdragon 8 Gen 2	Adreno 740
Samsung Galaxy S22	Snapdragon 8 Gen 1	Adreno 730
OnePlus 10 Pro	Snapdragon 8 Gen 1	Adreno 730
OnePlus 9T Pro	Snapdragon 888	Adreno 660
Huawei P50 Pro	HiSilicon Kirin 9000	ARM Mali G78 MP24
Xiaomi 12 Pro	Snapdragon 8 Gen 1	Adreno 730

# L'Openness (Ouverture) (1/2)

- ❑ L'ouverture désigne le niveau de liberté que les gens possèdent afin d'utiliser, de contribuer et d'innover pour l'OS mobile pour leurs fins.
- ✓ Pour les fabricants d'appareils mobiles, l'ouverture détermine combien ils peuvent personnaliser le système d'exploitation et différencier leurs appareils des autres.
- ✓ Pour les opérateurs, l'ouverture de l'OS mobile détermine comment leurs services peuvent être portés, migrés, déployés, et exécutés facilement sur les appareils.

# L'Openness (Ouverture) (2/2)

- ✓ Pour les développeurs, l'ouverture détermine combien il est facile de développer de nouvelles applications avec leurs idées créatives et comment leur investissement sur le développement de l'application peut être maximisé.
- ✓ Pour les consommateurs, l'ouverture signifie la manière dont ils peuvent obtenir plus d'applications.
- ✓ L'ouverture peut également donner aux gens la chance de participer à l'élaboration et à l'évolution de l'OS mobile lui-même au cours de son cycle de vie.

# Le Cloud sur mobile

- L'utilisation d'un Smartphone ou d'une tablette conduit à une plus grande adoption de ces services de Cloud :
  - ✓ 27% des utilisateurs de Smartphones et 26% des utilisateurs de tablettes tactiles utilisent le Cloud.
  
- Les services du Cloud sont présentés comme des sites Web accessible à travers des navigateurs mobiles.
  - ✓ Plusieurs services sont fournis à travers des applications web mobiles.

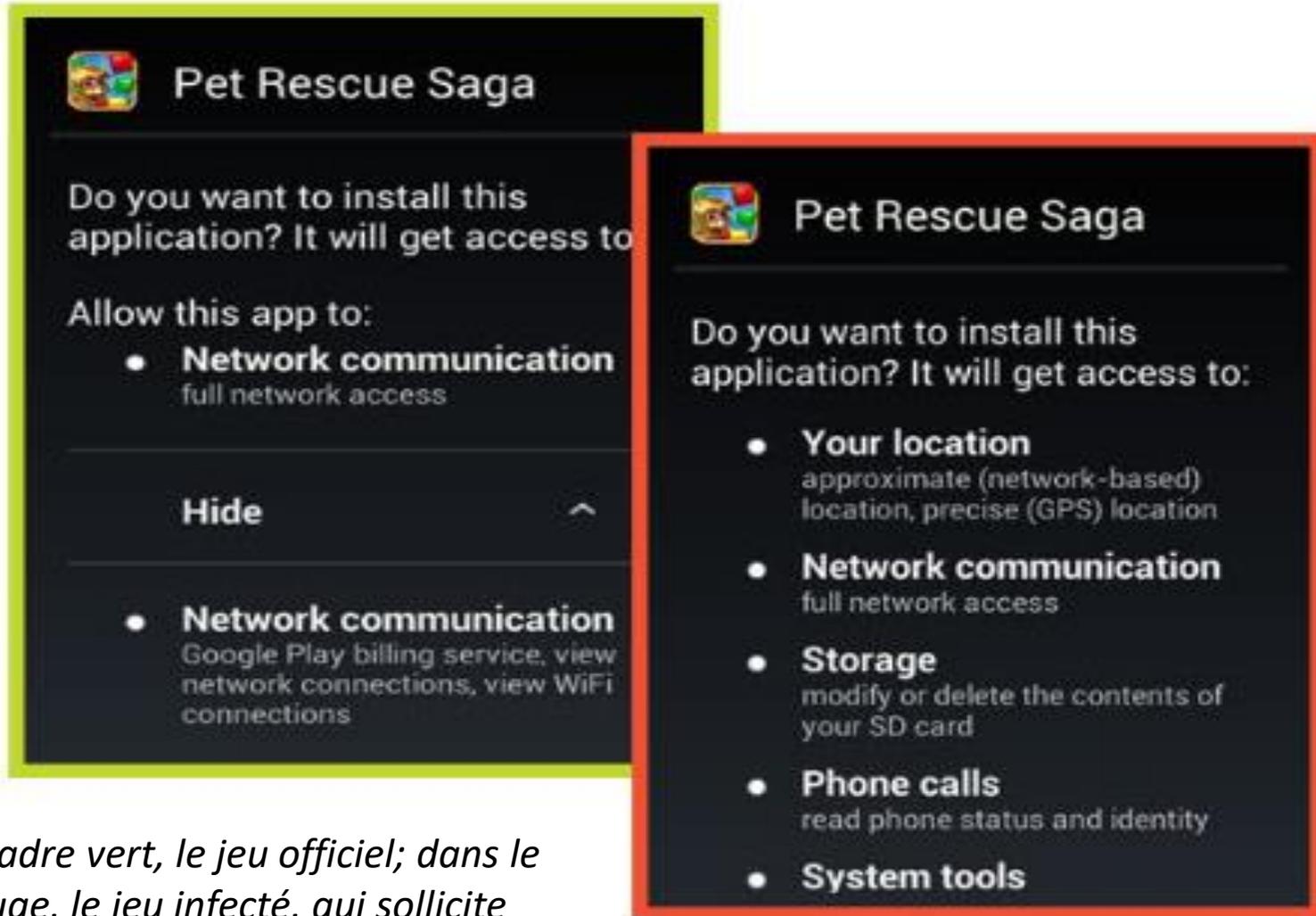
# La sécurité

- ❑ En général, les consommateurs cherchent le design ou le côté pratique de l'appareil (Batterie, taille de l'écran).
- ✓ La sécurité, invisible, impalpable, passe généralement au second plan.
  - ✓ Vol, personnes non autorisées, virus ou téléphone endommagé
    - ✓ Conduit à ... stress ou perte d'argent pour les consommateurs.
- ❑ La sécurité constitue un défi majeur pour un OS mobile. Il doit mettre l'accent sur la sécurité et sur des **mesures faciles à mettre en place** en cas d'ennui.

# Sécurité des OS mobiles actuels

- ❑ 99% des téléphones mobiles capables de se connecter à internet fonctionnent sur Android (Google), iOS (Apple).
- ✓ Virus
  - ✓ Plus de 90% des malwares sur mobiles ciblent les téléphones Android.
    - ✓ Exemple: Android Defender.
- ❑ iOS devance les autres systèmes grâce à son contrôle de la qualité et de la sécurité des applications.

# Exemple d'applis trompeuses



*Dans le cadre vert, le jeu officiel; dans le cadre rouge, le jeu infecté, qui sollicite plus d'autorisations*

(source: [F-Secure](#))

Partie 1: Architecture

**Partie 2: Présentation des OS mobiles**

# OS mobiles actuels

- 99% des téléphones mobiles capables de se connecter à internet fonctionnent sur Android (Google), iOS (Apple).
- Ces deux systèmes ne sont pas compatibles entre eux.
- Chaque système dispose de sa propre plateforme de téléchargement d'applications.

# iOS: Le système d'Apple

- ❑ iOS (anciennement nommé iPhone OS) est en développement depuis 2007 et a été créé pour l'iPhone. Depuis, il est aussi utilisé dans d'autres produits tel que l'iPad, l'iPod Touch et l'Apple TV.
- ❑ Régulièrement mis à jour par Apple, une nouvelle version majeure du système est disponible tous les ans. La version actuelle est la version 17.



# iOS: SDK

- ❑ Le kit de développement iPhone SDK a été officiellement annoncé le 6 mars 2008 par Apple.
- ❑ Contient les outils et interfaces nécessaires pour développer, installer, exécuter et tester les applications natives créées pour un appareil iOS.
- ❑ Outils de développement:
  - ✓ Xcode: écrit en Objective-C
  - ✓ Interface Builder
  - ✓ Instruments
  - ✓ iPhone Simulator
  - ✓ Swift (remplace le langage Objective-C en 2014). [Lien](#)

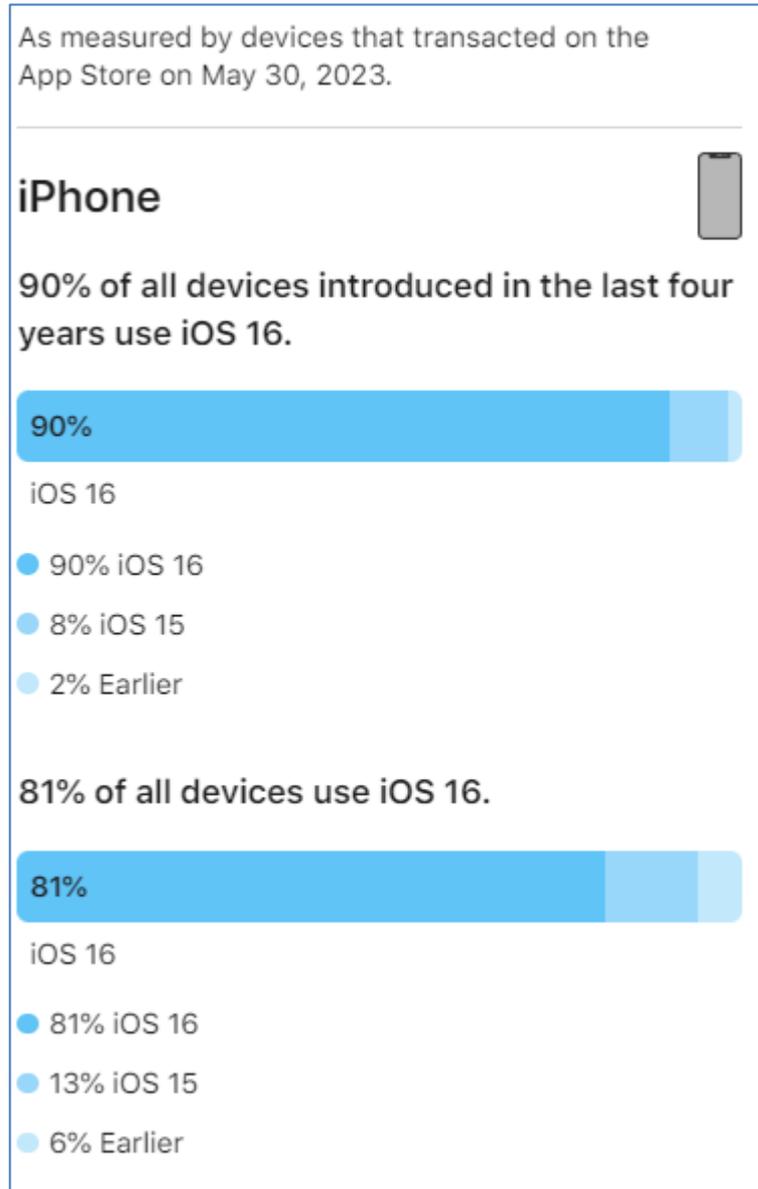
[Source](#)

# iOS: Les versions

Version	Most recent version	Recent version release date	Device end-of-life		
			iPad	iPhone	iPod Touch
iOS 5	5.1.1	May 7, 2012	1st gen	—	3rd
iOS 6	6.1.6	February 21, 2014	—	3GS	4th
iOS 7	7.1.2	June 30, 2014		4	—
iOS 8	8.4.1	August 13, 2015		—	—
iOS 9	9.3.5	August 25, 2016	2, 3rd, Mini (Wi-Fi only)	—	5th
	9.3.6	July 22, 2019	2, 3rd, Mini (Wi-Fi + Cellular)	4S	—
iOS 10	10.3.3	July 19, 2017	4th (Wi-Fi only)	5C	
	10.3.4	July 22, 2019	4th (Wi-Fi + Cellular)	5	
iOS 11	11.4.1	July 9, 2018	—		
iOS 12	12.5.7	January 23, 2023	Air (1st), Mini 2, Mini 3	5S, 6	6th
iOS 13 / iPadOS 13	13.7	September 1, 2020	—		
iOS 14 / iPadOS 14	14.8.1	October 26, 2021			
iOS 15 / iPadOS 15	15.7.3	January 23, 2023	Air 2, Mini 4	6S, SE (1st), 7	7th
iOS 16 / iPadOS 16	16.3.1	February 13, 2023	—		
	16.4 beta 2	February 28, 2023			

[Source](#)

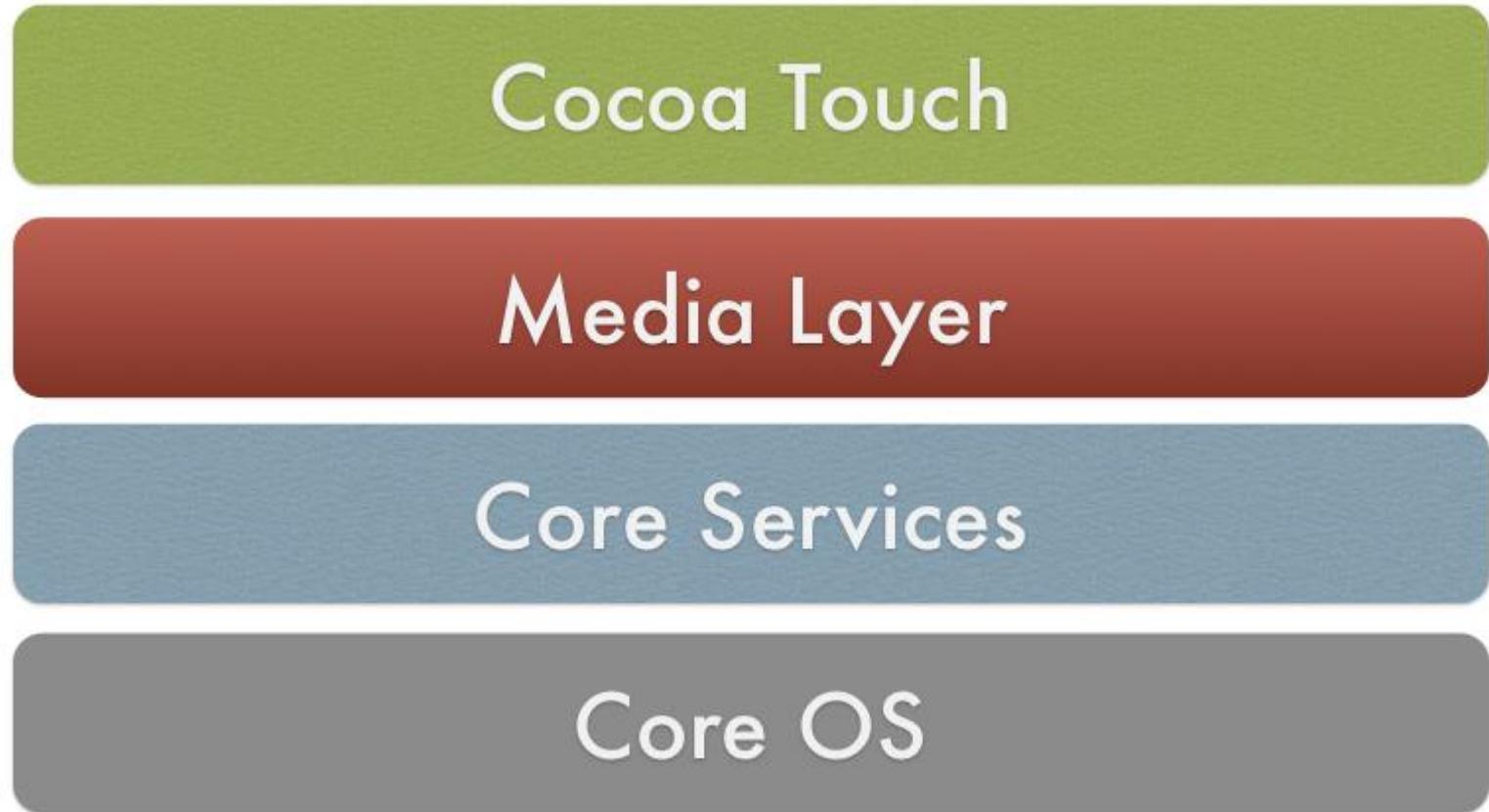
# iOS: Distribution des versions



[Source](#)

# iOS: Architecture (1/3)

- Architecture en 4 couches:



# iOS: Architecture (2/3)

## □ Core OS: Les bases de l'OS

- Le noyau d'OS X, la gestion de charge du processeur en fonction de la batterie,
- Le système de bibliothèques, le protocole TCP/IP du noyau BSD,
- La prise en charge des certificats, la prise en charge des Sockets Réseau, le système de sécurité, etc.

## □ Core Services: Gestion plus poussée du système

- La gestion des collections, la géolocalisation de l'appareil (core location), la gestion d'un carnet d'adresse, des services réseau, la gestion du réseau en lui-même,
- Le « threading », gestion des processus légers, l'accès à des fichiers, la gestion de préférences,
- SQLite, bibliothèque permettant la gestion de bases de données, des utilitaires pour la gestion d'URL, etc.

# iOS: Architecture (3/3)

- ❑ Media: gère les données multimédia
  - Core Audio, prise en charge des formats d'image JPEG, PNG, TIFF, OpenAL, prise en charge du format PDF, la gestion de l'« audio mixing », le moteur graphique Quartz,
  - La gestion de l'enregistrement audio, Core Animation, la gestion du « Video Playback », la prise en charge d'OpenGL, etc.
  
- ❑ Cocoa Touch: l'interface graphique
  - La gestion des événements Multitouch, la gestion des alertes, les contrôles Multitouch, une vue Web utilisant le moteur de rendu Webkit de Safari, la gestion de l'accéléromètre,
  - La sélection d'un contact, la gestion de contenu avec une hiérarchie, la sélection d'une image, la localisation, etc.

# iOS: Jailbreak

- ❑ **Jailbreak** est un processus permettant aux appareils tournant sous **iOS** d'obtenir un accès complet pour déverrouiller toutes les fonctionnalités du système, éliminant ainsi les restrictions et sécurités posées par Apple.
- Permet l'accès à un centre de téléchargement d'applications développées par des personnes non-affiliées à Apple (Cydia)
- L'appareil, hacké, devient vulnérable. On peut donc accéder à l'appareil à distance et obtenir toutes ses données (iCloud, Mots de passes, confidentialité). [Source](#)

# iOS: Avantages et inconvénients

## Avantages:

- ✓ Système stable,
- ✓ Expérience utilisateur simple,
- ✓ Régulièrement mis à jour et amélioré,

## Inconvénients:

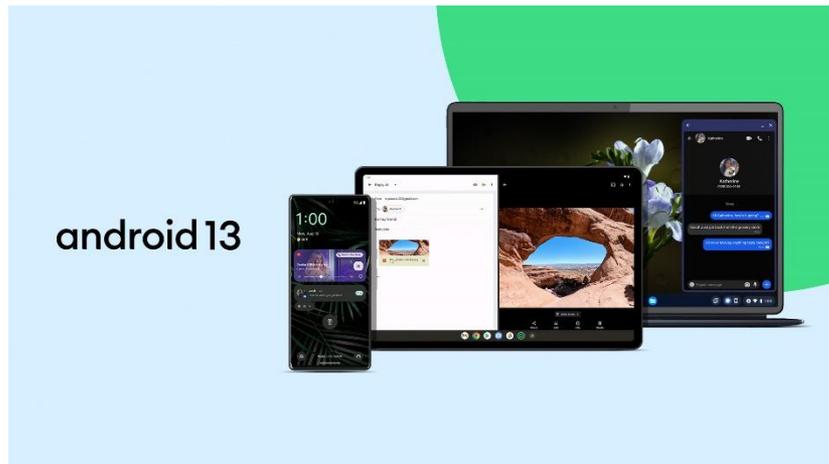
- ✓ Système verrouillé par Apple (peu de personnalisation possible, moins de souplesse),
- ✓ Prix élevé.

# Android



# Android : Le système de Google

- ❑ Android est un OS Open Source lancé en Juin 2007
  - ✓ Conçu pour Smartphone et tablette puis TV, voiture, etc.
- ❑ Google Play, qui succède à Android Market en 2012, est le magasin en ligne de Google.
  - ✓ Il permet de télécharger des logiciels, des livres, des films ou de la musique, payants ou non.
- ❑ Régulièrement mis à jour par Google, une nouvelle version majeure du système est disponible tous les ans. La version actuelle est la version 13.



# Android : SDK

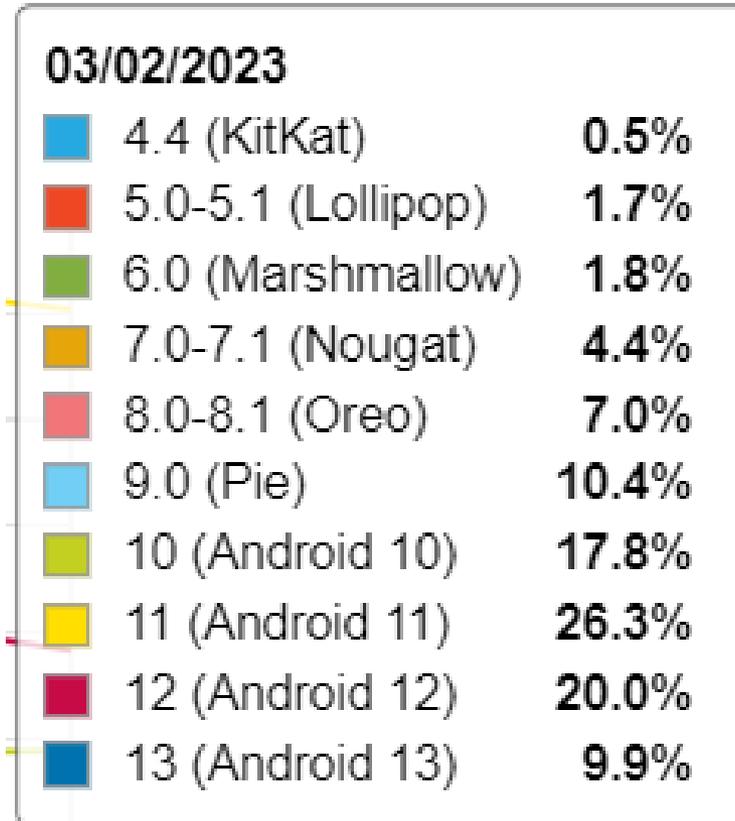
## ☐ Android SDK (Software Development Kit) définit:

- ✓ Un environnement de développement,
- ✓ Une machine virtuelle Java adaptée,
- ✓ Un environnement de construction d'application Android,
- ✓ Éditeur de mise en page visuelle,
- ✓ APK Analyzer,
- ✓ Éditeur de code intelligent,
- ✓ Profileurs en temps réel,
- ✓ Des émulateurs de téléphones ou de tablettes AVD (Android Virtual Device) et une énorme API.

[Source](#)

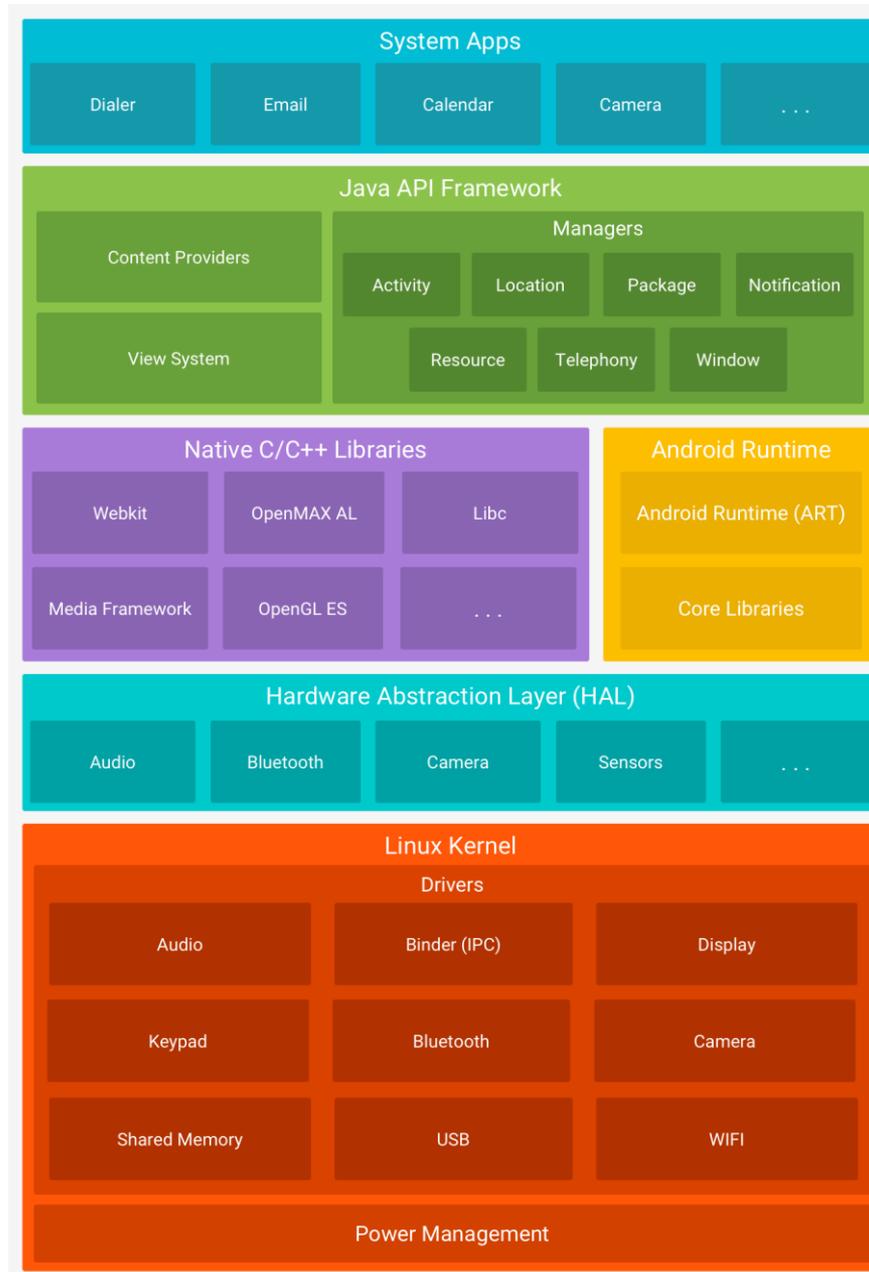
# Android : Distribution des versions

☐ (03 Mars 2023)



[Source](#)

# Android : Architecture (1/3)



# Android : Architecture (2/3)

- ❑ 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'accessibilité aux données des autres applications (ex : les contacts) et donc les partages de données
  - ✓ Des ressources = les fichiers non codes comme les images, les écrans (Resource Manager)
  - ✓ Des Notifications (affichage d'alerte dans la barre de titre)
  - ✓ Des Activités = l'enchaînement des écrans

# Android : Architecture (3/3)

- ❑ La couche "Libraries" (bibliothèques) = couche logicielle basse pour utiliser
  - ✓ Les formats multimédia : images, audio et vidéo enregistrés comme rendu
  - ✓ Les dessins 2D et 3D, bitmap et vectoriel
  - ✓ 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.
- ❑ La couche d'abstraction matérielle (HAL) fournit des interfaces standard qui exposent les capacités matérielles de l'appareil au cadre d'API Java de niveau supérieur.
- ❑ Le noyau Linux sur lequel ART s'appuie pour gérer le multithreading, la mémoire. Le noyau Linux apporte les services de sécurité, la gestion des processus, etc.

# Android 13 : Interfaces personnalisées

- ✓ Samsung : [One UI 5.1](#)
- ✓ Oppo : [ColorOS 13](#)
- ✓ OnePlus : [OxygenOS 13](#)
- ✓ Vivo : [FunTouchOS 13](#)
- ✓ Xiaomi : [MIUI 14](#)

# Android : Avantages et inconvénients

## Avantages:

- ✓ Les marques les plus connues tels que HTC, Samsung, LG, ou encore Sony Ericsson possèdent un système d'exploitation Android, ce qui laisse le choix à chaque personne de choisir sa marque de prédilection,
- ✓ Expérience utilisateur simple,
- ✓ Régulièrement mis à jour et amélioré,

## Inconvénients:

- ✓ Bug de certaines applications.
- ✓ Problème de fragmentation des versions.
- ✓ Problèmes de sécurité.

# Comparison



**VS**



# Similitudes

- ❑ Ils ont un kit de développement logiciel documenté (SDK) avec des API bien définies qui permettent aux développeurs de créer des applications pour ces systèmes,
- ❑ Ils ont des magasins d'applications en ligne pour les développeurs afin de publier et pour les utilisateurs afin de télécharger des applications,
- ❑ L'accent est mis sur la durée de vie de la batterie.
  - ✓ Les systèmes essaient de réduire au maximum la consommation de l'énergie des composants de l'appareil et les mettre en veille dès que possible.

# Comparaison : Système

	<b>iOS</b>	<b>Android</b>
Compagnie	<b>Apple Inc.</b>	<b>Open Handset Alliance</b>
Part du marché	<b>27,33%</b>	<b>71,9%</b>
Version actuelle	<b>16.3.1</b>	<b>13</b>
License	<b>Propriétaire</b>	<b>Gratuit</b>
Famille OS	<b>Darwin</b>	<b>Linux</b>
Architecture CPU	<b>ARM</b>	<b>ARM</b>
Programmé en	<b>C, C++, Objective-C, Swift</b>	<b>C, C++, Java, kotlin</b>

[Source](#)

# Comparaison : Ecosystème des apps

	iOS	Android
Magasin officiel	<b>App Store</b>	<b>Google Play</b>
Plateforme officielle de dev.	<b>MAC OS X avec iOS SDK</b>	<b>Linux, MAC OS X et Windows</b>
Cout de dev.	<b>Gratuit</b>	<b>Gratuit</b>
Cout de publication	<b>99\$/an</b>	<b>25\$</b>

[Source](#)

# Références

- ✓ Sehgal, Krishna, Alind Jain, Preeti Nagrath, and Ashish Kumar. "Recent Advances in Networks and Data Security Survey on Various Mobile Operating Systems," in ***International Conference on Innovative Computing and Communications (Springer)***, pp. 181–90. 2019.
- ✓ Ivanov, Vladimir, Alexey Reznik, and Giancarlo Succi. "Comparing the reliability of software systems: A case study on mobile operating systems." ***Information Sciences*** 423 : 398-411. 2018.
- ✓ Omelchenko, Tatiana et al. , "Protection Software for Mobile Operating Systems," in ***2018 International Conference on System Modeling & Advancement in Research Trends (SMART) (IEEE)***, pp. 54–59. 2018
- ✓ Novac, Ovidiu Constantin, et al. "Comparative study of Google Android, Apple iOS and Microsoft Windows phone mobile operating systems." ***2017 14th international conference on engineering of modern electric systems (EMES)***. IEEE. 2017.
- ✓ Li X., Wang Y., Wu J., Jiang K. and Liu B.: "Mobile OS Architecture Trends" ***Intel Technology Journal***, volume 16, issue 4, pp 178-198. 2012.