

Développement natif sous iOS

Cours n°1 - Introduction

iOS

Répartition du module

CM	6h
TD	6h
TP	22h30
DS	1h30

Contact

robin.dupret@uphf.fr

Marché mobile

- Deux acteurs principaux :
 - Android (Google)
 - iOS (Apple)



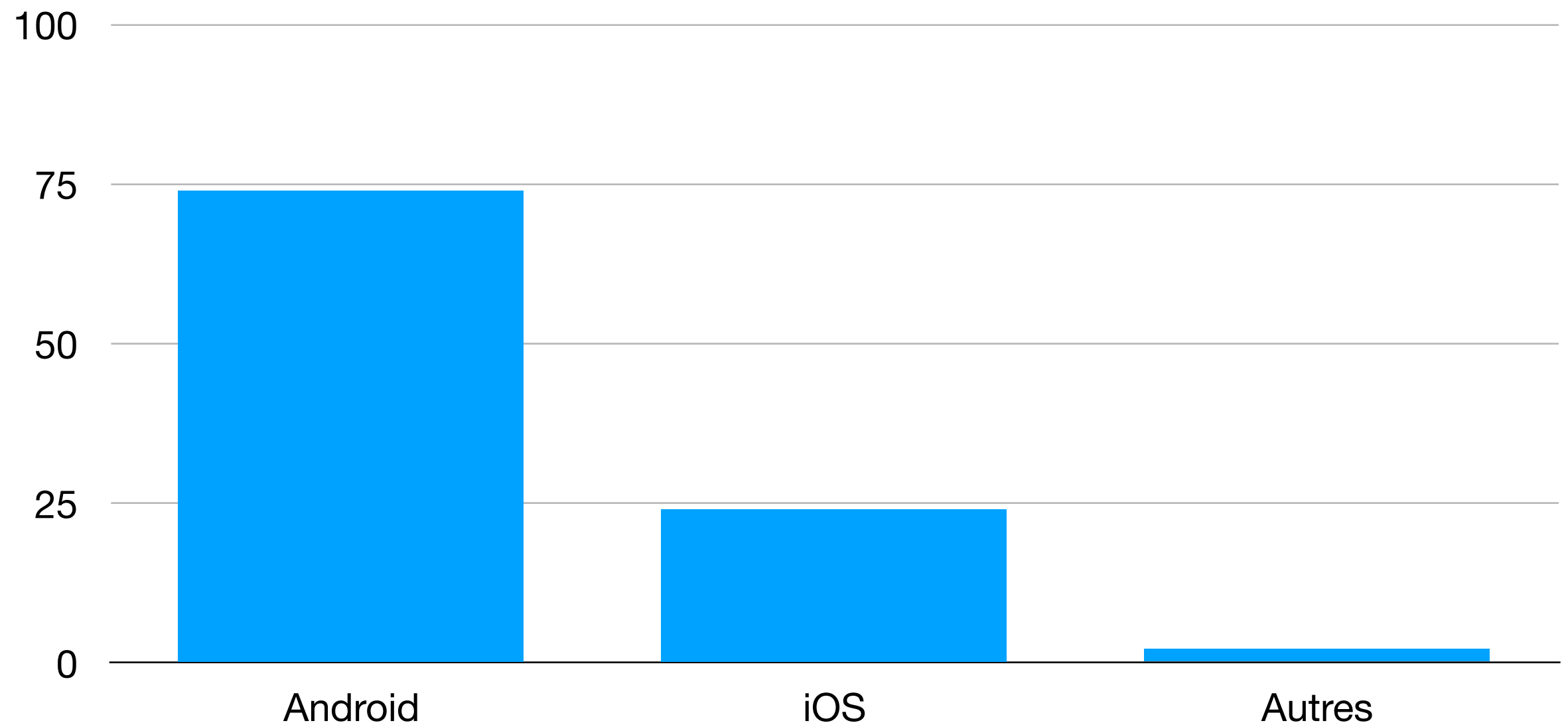
ios

Marché mobile

- Au départ que les téléphones (*smartphones*)
- Diversification
 - Tablettes
 - Montres
 - Boîtiers télé

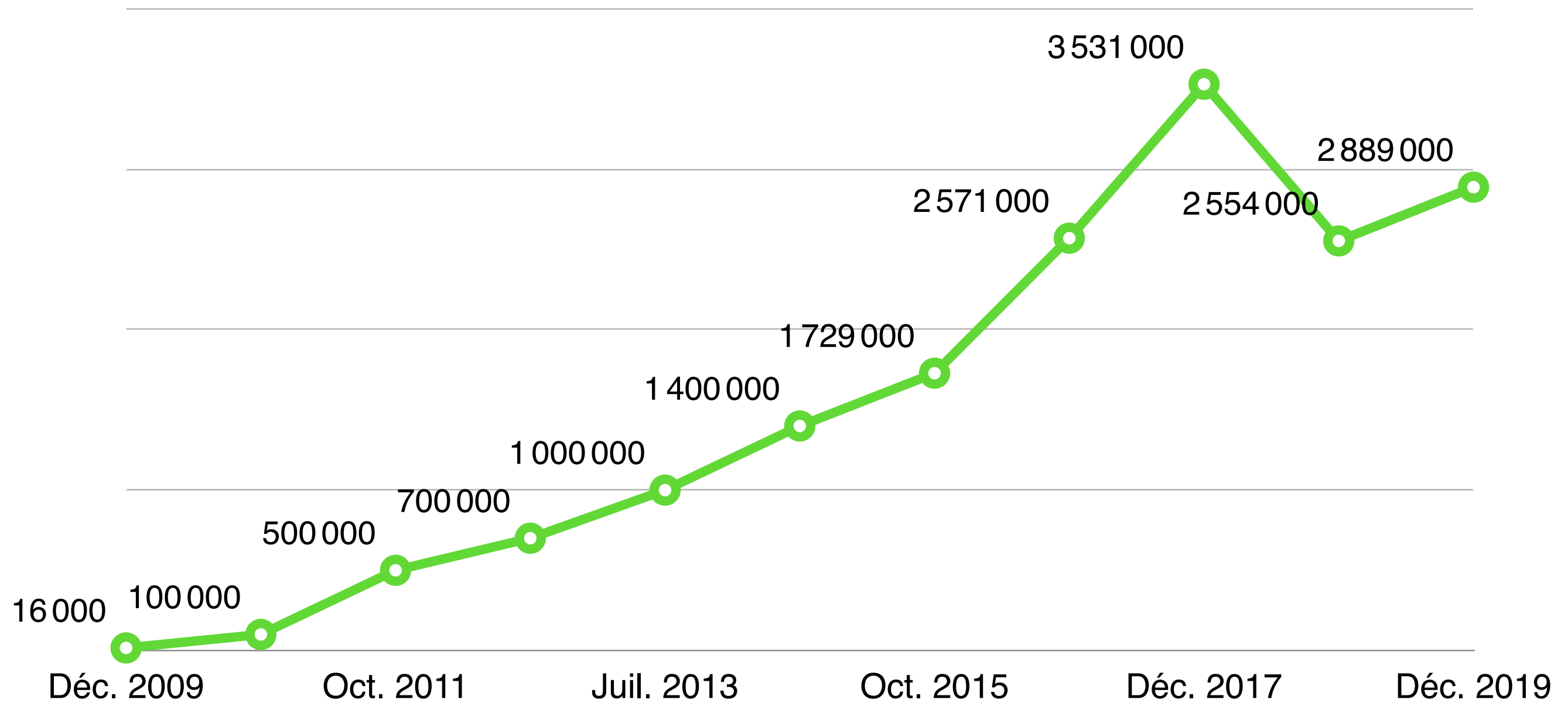


Parts de marché (Janvier 2020)



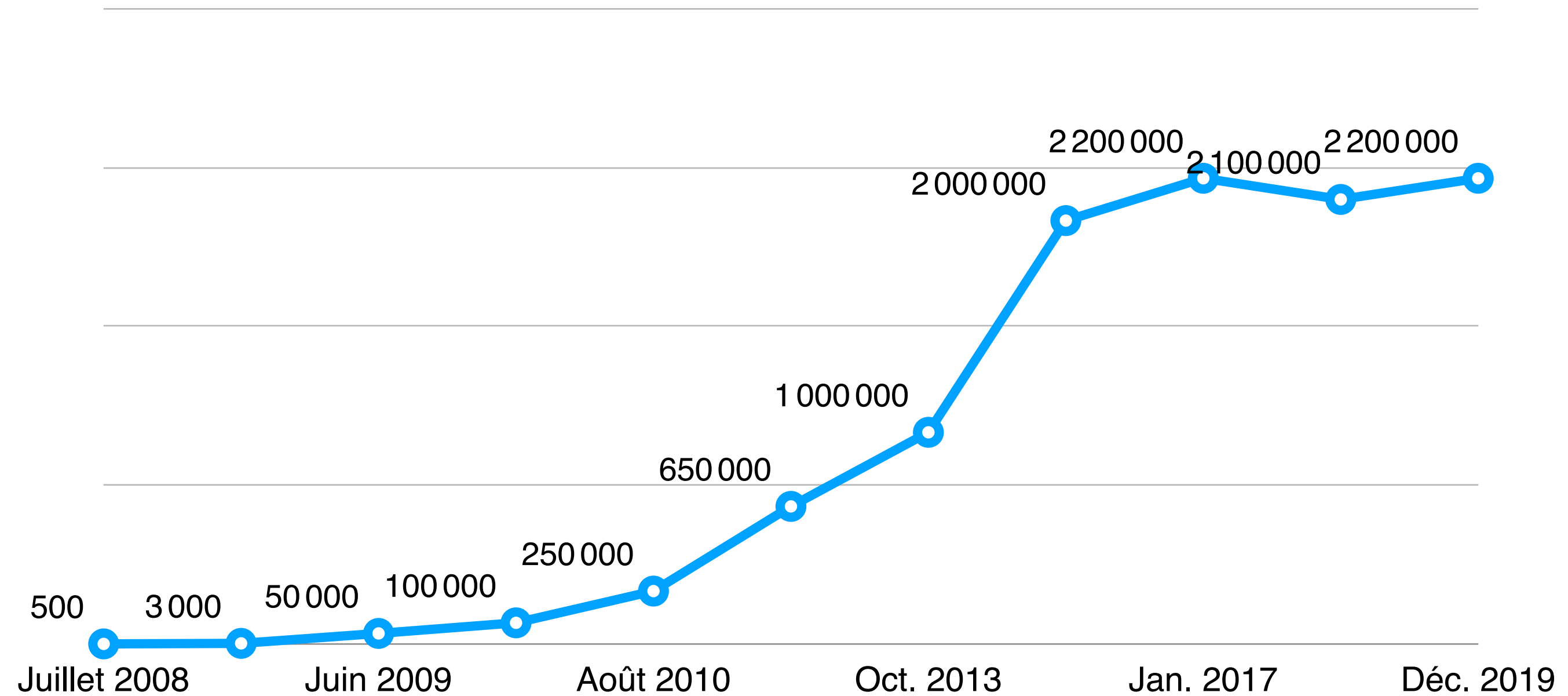
Parts de marché

Nombre d'applications sur le Play Store (Android)



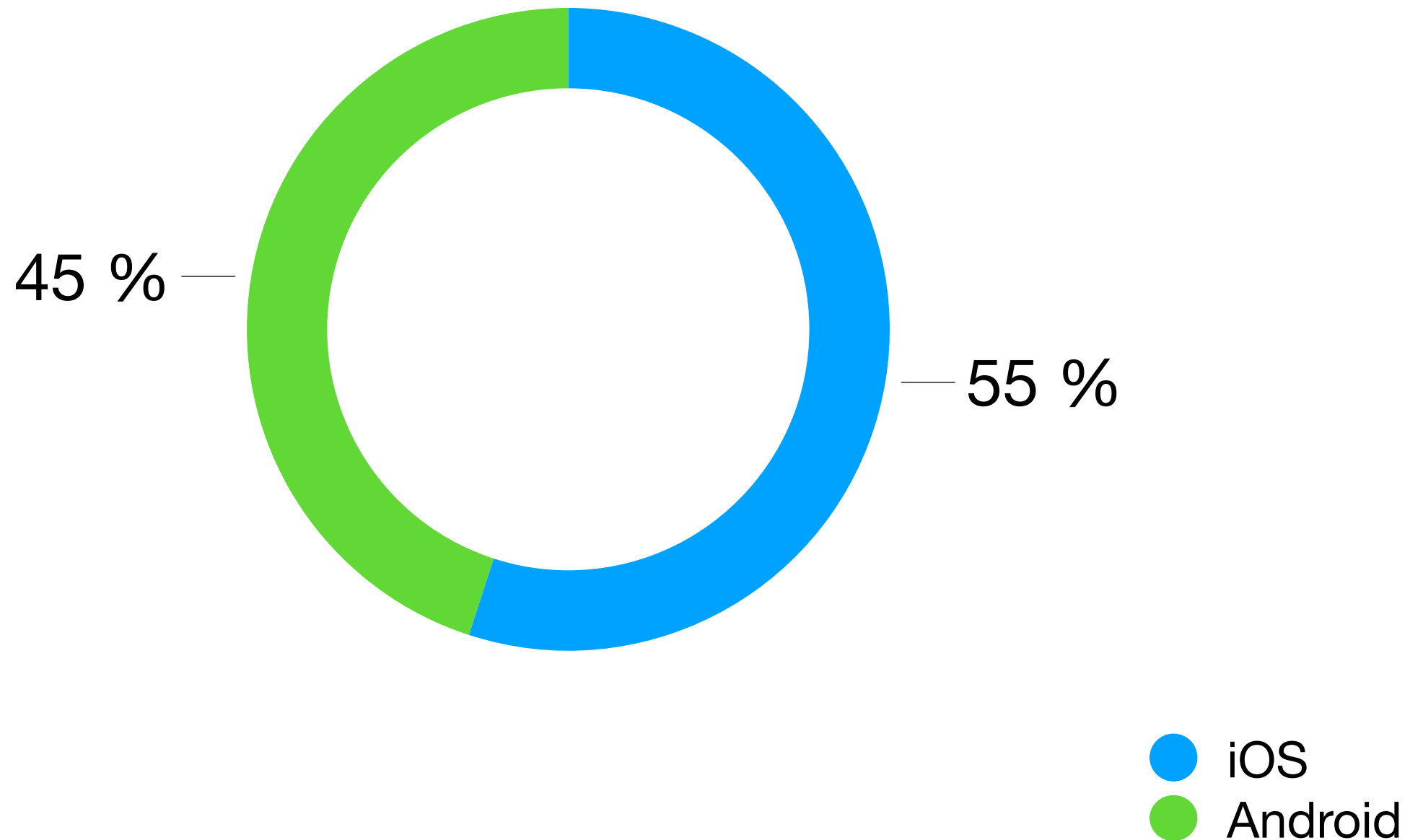
Parts de marché

Nombre d'applications sur l'App Store (iOS)



Parts de marché

Revenus générés par
les applications



Historique - iOS

- Fin des années 80 : Fondation de « NeXTStep » par Steve Jobs
 - Création du langage Objective-C
 - Création d'un OS basé sur Unix et Objective-C
- 1997 : Rachat de NeXTStep par Apple
 - Création de Mac OS X basé sur le système de NeXTStep
- 2007 : Premier iPhone doté de « iPhone OS »

Historique - iOS

- 2008 : Ouverture de l'App Store pour les iPhones
- 2010 : Sortie de l'iPad
 - « iPhone OS » devient « iOS »
- 2014 : Création du langage Swift
- 2019 : Naissance de « iPad OS »
 - Basé sur iOS
 - Possédant d'autres fonctionnalités facilitant le multi-tâches, etc.

Historique - Android

- 2003 : Formation de la « startup » Android visant à proposer des OS pour appareils potos puis téléphones mobiles
- 2005 : Rachat par Google
- 2008 : Premier smartphone commercialisé avec Android (HTC Dream)
- Mai 2014 : Android représente plus de 50% du marché mobile

Historique - Android

- 2016 : Les premiers téléphones « Pixel », entièrement fabriqués par Google, sont commercialisés
- 2019 : Dû au contexte politique entre les Etats-Unis et la Chine, certains fabricants, notamment Huawei se voient contraints de fournir leur propre système d'exploitation à la place d'Android : Harmony OS

Développement

- Deux façons de développer :
 - Applications cross-plateformes
 - Applications natives

Cross-plateformes

- Un code partagé pour toutes les plateformes
- Avantages
 - Plus simple et plus rapide à développer
 - (Souvent) Du code peut être partagé avec une application web
- Inconvénients
 - Exécution plus lente
 - Application plus lourde
 - Limitations pour des aspects bas niveau

Cross-platformes

- Cordova
- Ionic
- React Native
- Vue Native
- NativeScript
- Flutter

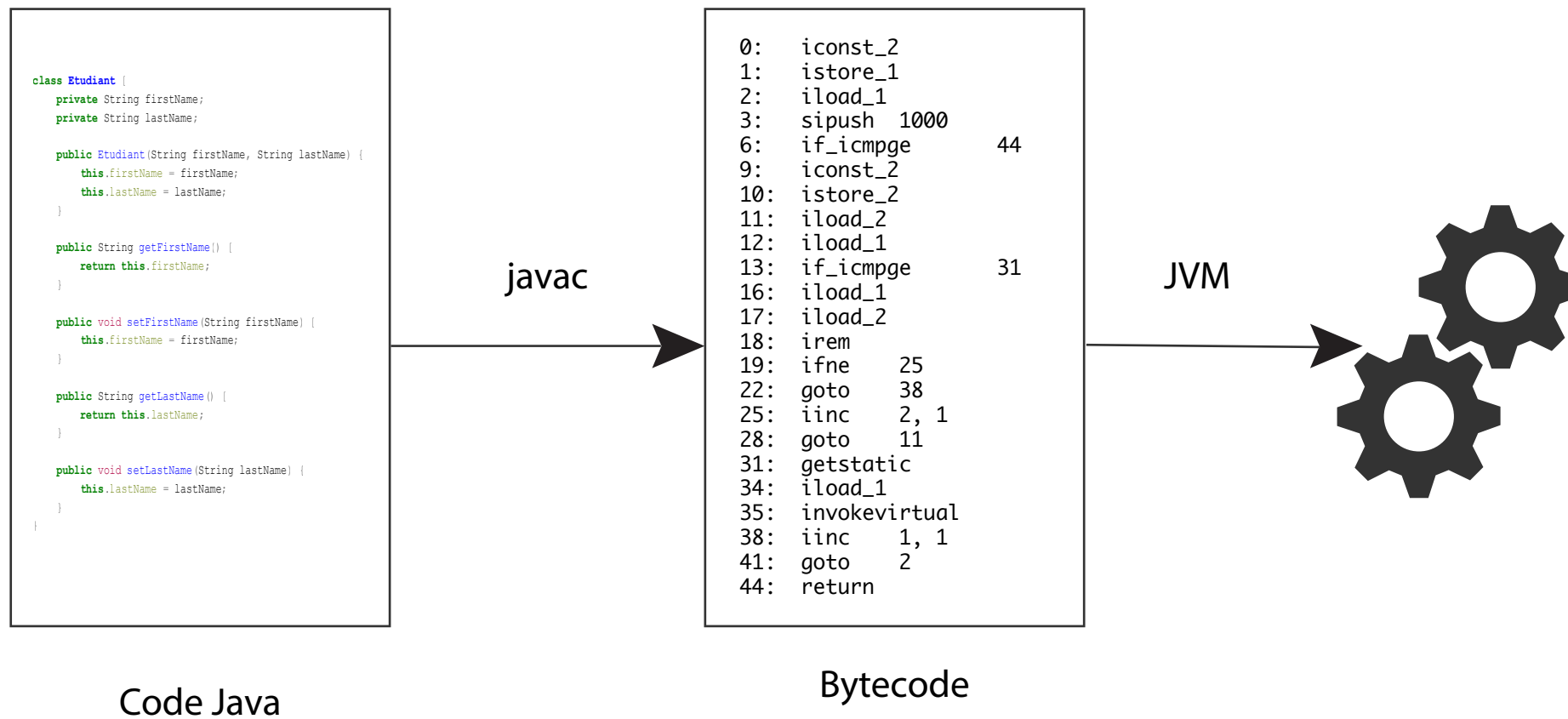
Natif

- Un code et un langage spécifique à la plateforme
- Avantages
 - Meilleures performances
 - Permet d'utiliser des bibliothèques disponibles dans d'autres langages
 - Facilités pour créer des applications cohérentes avec la plateforme ou utilisant une fonctionnalité en particulier (e.g. TouchID)
 - Accès à des fonctionnalités bas niveau
 - On n'a parfois pas le choix
- Inconvénients
 - Développement moins rapide et plus complexe

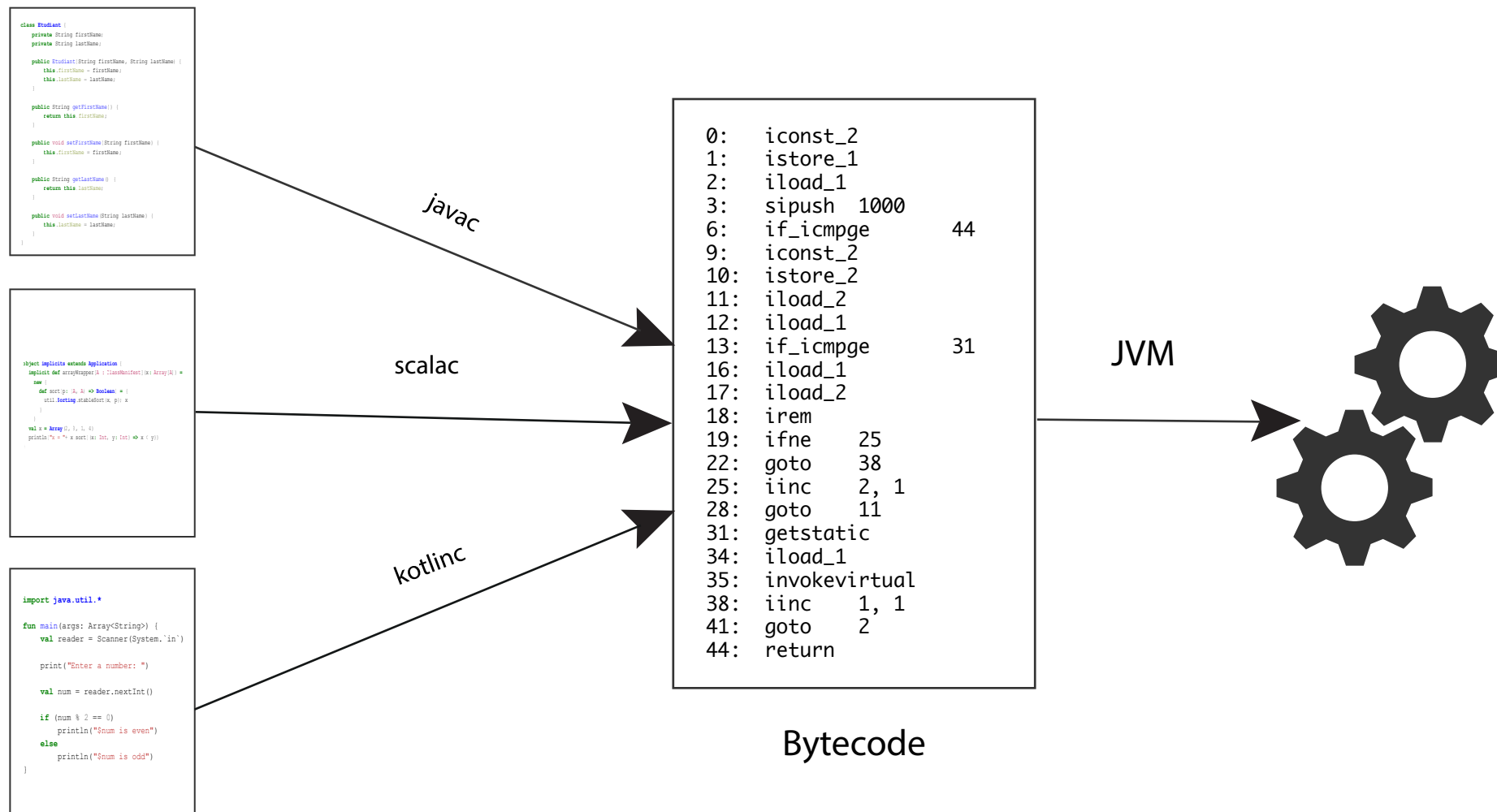
Natif

- Sous Android
 - Java
 - Kotlin
- Pour terminaux Apple
 - Objective-C
 - Swift

Natif - JVM



Natif - JVM



Natif - LLVM

```
protocol Vehicule {  
    func avancer()  
    func entretenir()  
}  
  
class Voiture : Vehicule {  
    func avancer() {  
        // ...  
    }  
  
    func entretenir() {  
        // ...  
    }  
}
```

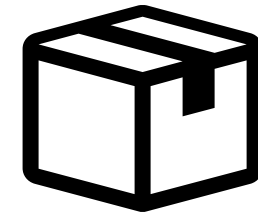
Code Swift

swiftc

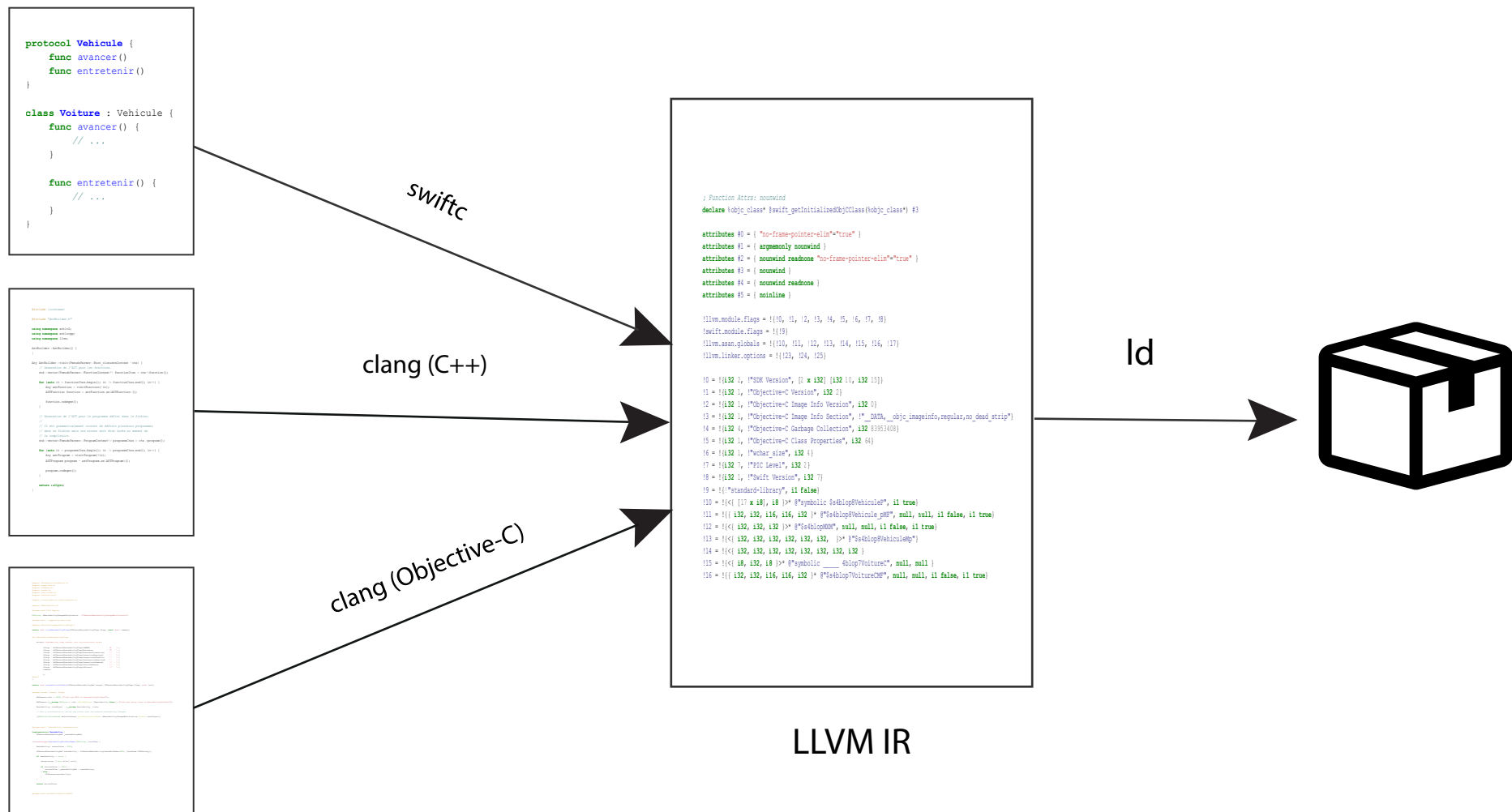
```
; Function Attrs: nounwind  
declare @objc_class* @swift_getInitializedObjCClass(@objc_class*) #3  
  
attributes #0 = { "no-frame-pointer-elim"="true" }  
attributes #1 = { argmemonly nounwind }  
attributes #2 = { nounwind readonly "no-frame-pointer-elim"="true" }  
attributes #3 = { nounwind }  
attributes #4 = { nounwind readonly }  
attributes #5 = { nolinear }  
  
!llvm.module.flags = !{!0, !1, !2, !3, !4, !5, !6, !7, !8}  
!swift.module.flags = !{!9}  
!llvm.ident.global = !{!10, !11, !12, !13, !14, !15, !16, !17}  
!llvm.linker.options = !{!18, !19, !20}  
  
!0 = !{i32 0, "OSX Version", [! x i32] [i32 0, i32 1]}  
!1 = !{i32 1, "Objective-C Version", i32 2}  
!2 = !{i32 1, "Objective-C Image Info Section", i32 3}  
!3 = !{i32 1, "Objective-C Image Info Section", !"DATA__objc_imageinfo,regular,no_dead_strip"}  
!4 = !{i32 4, "Objective-C Garbage Collection", i32 8193408}  
!5 = !{i32 1, "Objective-C Class Properties", i32 64}  
!6 = !{i32 1, "Vchar size", i32 1}  
!7 = !{i32 7, "PIC Level", i32 1}  
!8 = !{i32 1, "Swift Version", i32 7}  
!9 = !{"standard-library", i1 false}  
!10 = !{!{ [! x i8], !8 }> "Symbolic $@objc@Vehicule", i1 true}  
!11 = !{!{ i32, i32, i32, i32 }> "Symbolic $@objc@Vehicule_ptr", null, null, i1 false, i1 true}  
!12 = !{!{ i32, i32, i32 }> "Symbolic", null, null, i1 false, i1 true}  
!13 = !{!{ i32, i32, i32, i32, i32, i32 }> "Symbolic $@objc@Vehicule_ptr"}  
!14 = !{!{ i32, i32, i32, i32, i32, i32, i32, i32 }> "Symbolic $@objc@Vehicule_ptr"}  
!15 = !{!{ i32, i32, i32, i32 }> "Symbolic $@objc@Vehicule_ptr", null, null }  
!16 = !{!{ i32, i32, i32, i32, i32 }> "Symbolic $@objc@Vehicule_ptr", null, null, i1 false, i1 true}
```

LLVM IR

ld



Natif - LLVM



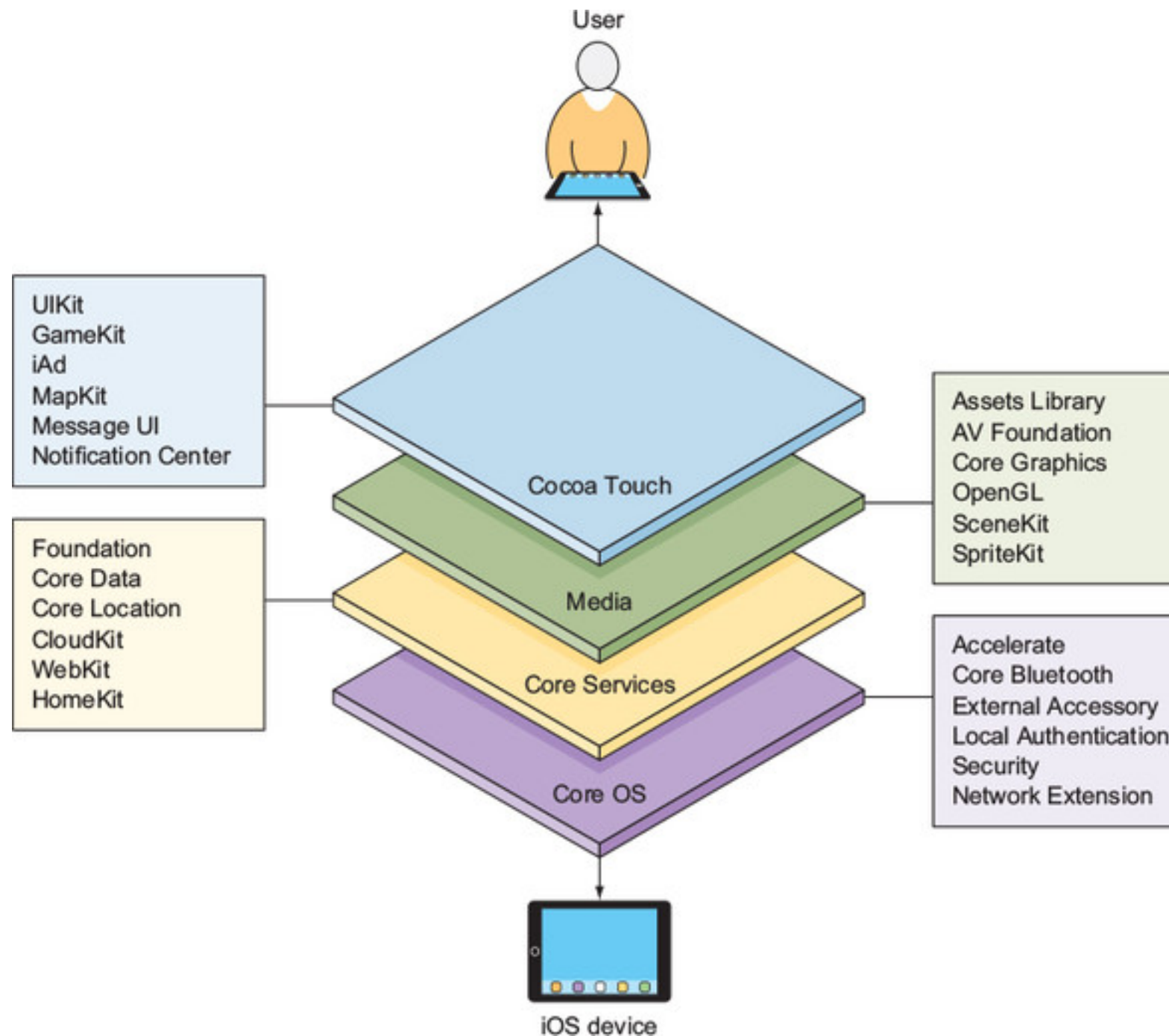
Spécificités iOS

- Et autres wearOS, iPadOS, macOS ...
- Les applications ne peuvent être développées que sur un ordinateur Apple
- Le matériel supportant ces OS est exclusivement Apple
- macOS Catalyst : Développement d'applications pour iPad et macOS

Développement sous iOS

- Avec Swift
 - Objective-C peut encore être utilisé mais l'est de moins en moins
- SDK (*Software Development Kit*) fournit
 - UIKit : Facilités pour réaliser l'interface
 - ARKit : Réalité augmentée
 - CoreML : Fonctions permettant de faire du *machine learning*
 - etc.

Développement sous iOS



Sources

- Parts de marché iOS / Android : <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- Statistiques Play Store : <https://www.appbrain.com/>
- Statistiques iOS : <https://www.appannie.com/en/insights/>
- Schéma SDK iOS : <https://livebook.manning.com/book/ios-development-with-swift/>