

# Développement Mobile

## Les problématiques du développement mobile



UNIVERSITÉ  
DE LORRAINE



2022 - Alexandre Leroux (alex@sherpa.one)

# PROBLÉMATIQUES

Bien que les solutions cross-platforms s'améliorent, le développement mobile est actuellement l'une des disciplines les plus complexes du développement logiciel.

**Les environnements technique, logiciel, légal et matériel évoluent en permanence.**

# PROBLÉMATIQUES

Le développement  
mobile "*moderne*"  
existe seulement  
depuis **2007**,  
correspondant à  
l'année de sortie du  
premier vrai  
"*smartphone*" :  
l'**iPhone**.



# PROBLÉMATIQUES

Lorsque l'on travaille sur un projet mobile il faut **redoubler d'efforts** :

- tests de **faisabilité**,
- tests en **conditions réelles**,
- tests de **compatibilité**  
(terminal / os, plugins / os, frameworks / os...),
- obtention de **retours d'expérience des utilisateurs**,
- veille **technologique / matérielle**...

# PROBLÉMATIQUES

La **veille technologique** doit permettre d'**effectuer les meilleurs choix possibles en amont et en cours du projet** :

- application mobile ou web responsive (ou encore PWA),
- développement natif ou cross-platforms,
- distribution iOS et/ou Android,
- version minimale de l'OS visé,
- périphériques visés (formats d'écran, performances...),

# PROBLÉMATIQUES

Les changements fréquents dans l'écosystème mobile peuvent fortement **perturber un développement en cours, un déploiement ou même une application en production.**

Il est donc judicieux de **proposer aux clients une prestation de maintenance** (évolutive et / ou en conditions opérationnelles).

# PROBLÉMATIQUES

La société **Apple**, leader charismatique, **impose son "*business model*" au marché mobile** (mode de distribution via un store privé, taxes, validation avant distribution...).

Les développeurs d'applications mobiles doivent s'adapter à la stratégie d'Apple et à sa politique "*fermée*".

# Développement d'applications mobile

NOS 5  
CONVICTIONS



<https://blog.axopen.com/2018/04/nos-5-convictions-pour-le-developpement-dapplication-mobile/>



# PROBLÉMATIQUES

**La distribution d'une application mobile, sur l'App Store ou sur le Google Play Store, est soumise à une validation du propriétaire du store.**

Apple revendique un store composé uniquement d'“*applications de qualité*”, testées et approuvées au préalable (performance, sécurité, respect de la vie privée...).

# PROBLÉMATIQUES

**Apple peut refuser de distribuer l'application pour de nombreux motifs** (plus ou moins légitimes) :

- contenu inapproprié / contenu non pertinent,
- mauvaise performance (ex : fuite de mémoire...),
- interface UI / experience UX incorrecte,

# PROBLÉMATIQUES

- transmission d'informations incomplètes (auteur de l'application, numéro à contacter en cas de panne...),
- absence d'images destinées à illustrer l'application dans l'app store, format ou résolution des images inadaptés...
- application non adaptée à tel modèle (ex : "*encoche*" de l'iPhone X)...

# PROBLÉMATIQUES

Toutes ces problématiques de validation peuvent **nuire aux délais de livraison et de distribution de l'application.**

Ce délai peut provoquer des **pénalités de retard pour la MOE** ou **l'annulation du projet par la MOA**  
(ex : application liée à un événement "*daté*").

# PROBLÉMATIQUES

**Google est plus "*souple*" au niveau de la distribution.**

L'application n'est (vraisemblablement) pas soumise à un test humain.

Une application mobile Android est donc **plus rapidement disponible sur le Google Play Store** et téléchargeable par les utilisateurs.



## Développement d'applications mobiles : pourquoi iOS est plus cher qu'Android ?

<https://fr.yeeply.com/blog/developpement-dapplications-mobiles-ios-plus-cher-android/>

# Why we quit developing Android apps

Published on September 11, 2014



**Ilie Ghiciuc** | Follow  
Founder & CEO



Here it is, bluntly put: we no longer develop Android apps. Counter-productive as this decision may look for a mobile app company, we're sticking to it and here's why:

## **1. Android testing costs a lot more than iOS testing.**

With almost 19 000 distinct Android devices on the market, it is a bit unrealistic to think there is time to actually test them all or to achieve comparable performance for the apps developed.

## Why we quit developing Android Apps ?

<https://www.linkedin.com/pulse/20140911071106-15803467-why-we-quit-developing-android-apps/>

# PROBLÉMATIQUES

**Hélas, le développement d'une application cross-platforms multiplie tous les problèmes abordés précédemment...**

La réalisation d'une application cross-platforms implique l'usage d'une solution technique intermédiaire, laquelle doit également être mise à jour au gré des évolutions des systèmes d'exploitation mobiles cibles.

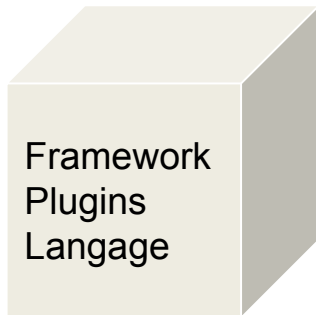


# PROBLÉMATIQUES

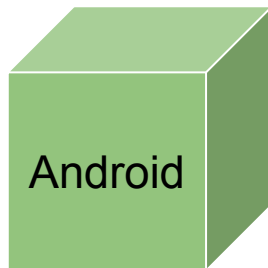
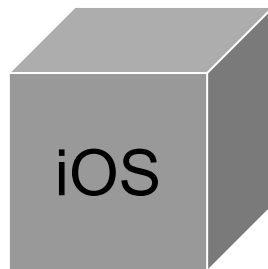
**Les différentes ressources et plugins utilisés peuvent ne plus être compatibles avec les plateformes visées, au fur et à mesure de leurs mises à jour.**



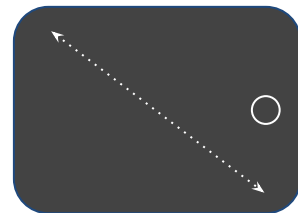
nouvelles  
fonctionnalités,  
évolution de la  
logique métier,  
correctifs,  
design...



adaptations aux évolutions  
des systèmes  
d'exploitation visés  
+  
adaptation aux évolutions  
du langage tiers utilisé



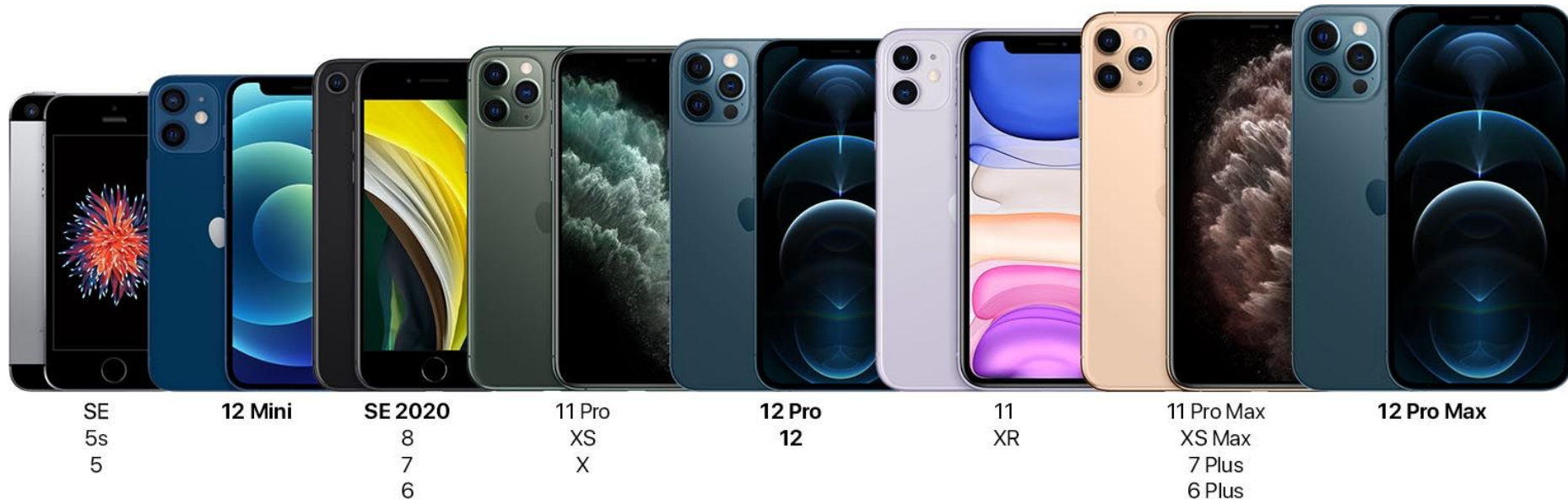
nouvelles fonctionnalités,  
normes juridiques,  
politique constructeurs,  
tendances du marché...



nouvelles caractéristiques  
techniques et physiques

**Les 4 domaines du processus de développement mobile  
où les évolutions peuvent intervenir**

# PROBLÉMATIQUES



**Multiplication des formats d'écran**

# PROBLÉMATIQUES

**Les solutions de développement cross-platforms ne tiennent pas toujours leurs promesses.**

*“Sur le papier”* le concept de *“code source unique”* est séduisant, mais dans la pratique, **il est souvent nécessaire de procéder à des adaptations pour que le projet fonctionne d’une plateforme à l’autre.**

# React Native

Learn once, write anywhere.

Gorgeous native mobile app development with Java or Kotlin for iPhone, Android etc....



## Fast development

Flutter's *hot reload* helps you quickly and easily experiment, build UI, add features, and fix bugs faster. Experience sub-second reload times without losing state on emulators, simulators, and hardware.

[Learn more](#)

### Cross platform Business Friendly GPU Accelerated

Kivy runs on Linux, Windows, OS X, Android, iOS, and Raspberry Pi. You can run the same code on all supported platforms.

It can natively use most inputs, protocols and devices including WM\_Touch, WM\_Pen, Mac OS X Trackpad and Magic Mouse, Mtddev, Linux Kernel HID, TUIO. A multi-touch mouse simulator is included.

Kivy is 100% free to use, under an MIT license (starting from 1.7.2) and LGPL 3 for the previous versions. The toolkit is professionally developed, backed and used. You can use it in a commercial product.

The framework is stable and has a well documented API, plus a programming guide to help you get started.

The graphics engine is built over OpenGL ES 2, using a modern and fast graphics pipeline.

The toolkit comes with more than 20 widgets, all highly extensible. Many parts are written in C using Cython, and tested with regression tests.

## Create Native iOS and Android Apps with JavaScript

Open source framework for building truly native mobile apps with Angular, Vue.js, TypeScript, or JavaScript.

## One codebase. Any platform.

Create purely native cross-platform user experiences with Boden

Boden empowers you to create native mobile cross-platform applications from a single codebase. It uses native widgets that match the look and feel users expect from their platform.

### Fast Development

Paint your app to life in milliseconds with Stateful Hot Reload. Use a rich set of fully-customizable widgets to build native interfaces in minutes.

### Expressive and Flexible UI

Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.

### Native Performance

Flutter's widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts, and your Flutter code is compiled to native ARM machine code using [Dart's native compilers](#).

## Xamarin

Free. Cross-platform. Open source.

An app platform for building Android and iOS apps with .NET and C#.

## RUBYMOTION

Écrivez des apps multiplateformes pour iOS, Android et OS X avec Ruby

Les promesses des frameworks mobiles cross-platforms : React Native, Codenameone, Flutter, Kivy, NativeScript, Ionic, Boden, Ruby Motion, Xamarin...

# PROBLÉMATIQUES

## UX dans un contexte d'utilisation instable

# PROBLÉMATIQUES

**Traditionnellement, un site web fonctionne avec des données distantes :**

- base de données,
- webservices / API,
- bibliothèques css / js,
- contenus (assets) : images, vidéos, fichiers pdf...

# PROBLÉMATIQUES

**Une application mobile ne peut pas se permettre de fonctionner intégralement avec des données externes accessibles par internet (comme un site).**

**Si l'application passe en mode déconnecté (offline), elle ne peut plus accéder aux données et devient inutilisable.**



# PROBLÉMATIQUES

Les utilisateurs peuvent “*comprendre*” qu’une application ne fonctionne pas “*normalement*” si la connexion internet est indisponible :

- connexion à un service externe (ex :compte bancaire),
- connexion gps,
- mise à jour de données (ex : résultats sportifs en temps réel)...

# PROBLÉMATIQUES

Ceci-étant, il faudra **toujours informer les utilisateurs sur l'état des services, les capacités fonctionnelles utilisables, et ce, en temps réel.**

Un utilisateur qui ne comprend pas l'origine d'un dysfonctionnement peut penser :

- **que l'application ne fonctionne pas => mécontentement**
- **qu'il ne sait pas utiliser l'application => frustration**

# PROBLÉMATIQUES

Dans les 2 cas, ***“l'expérience utilisateur” (UX)*** sera **négative et préjudiciable à l'application :**

- non-utilisation,
- désinstallation,
- contact sav,
- migration vers la concurrence,
- mauvaise publicité...

# PROBLÉMATIQUES

Les utilisateurs ne *“comprendront”* pas que l’application change d’apparence ou dysfonctionne si la connexion internet est **indisponible** : liaison rompue avec les visuels / css distants...

Pour cette raison, **il est important d’utiliser des ressources locales** (typographies, images...).

# PROBLÉMATIQUES

Il faut prévoir que **la connexion est intermittente, ce qui engendre de nombreuses complications :**

- interruption en cours de chargement ou transmission de données,
- interruption en cours de mise à jour de l'application...

# PROBLÉMATIQUES

**Il ne faudra pas compter sur une connexion internet de qualité :**

- se méfier des tests effectués dans des locaux disposant de la fibre et donnant l'impression d'une réactivité / fluidité importante de l'application,
- prévoir des animations de chargement / messages d'information... (feedbacks)

# PROBLÉMATIQUES

Une première solution peut consister à  
**stocker les données en local :**

- exploitation du cache (attention à la perte de données d'une session à l'autre),
- stockage des données dans une BDD locale comme SQLite,

# PROBLÉMATIQUES

- stockage des fichiers dans le dossier local de l'application

Chaque application dispose en effet d'un espace alloué sur le terminal pour enregistrer ses données dédiées, non accessibles par l'utilisateur,

Il faudra aussi déterminer les assets qui peuvent être directement inclus au build de l'application.



# PROBLÉMATIQUES

Pour toute opération nécessitant un accès distant, une solution peut consister à **tenir un registre local des opérations en cours : débutées et finalisées.**

Le chargement de fichiers ou de données distantes, la synchronisation de données locales / distantes (ex : liaison à un CRM) pourront être traités selon leur statut lorsque la connexion au réseau est disponible.

# PROBLÉMATIQUES

**Persistence  
de données  
et d'états de l'application**

# PROBLÉMATIQUES

Une application mobile fonctionne avec une surface visuelle restreinte.

Par conséquent, son “état” logiciel se matérialise par une succession d'écrans : **à chaque état correspond un écran.**

**L'état d'une application est donc rapidement identifiable.**

# PROBLÉMATIQUES

## SCHEMA EXAMPLE

### HELPFUL TIP

Start by creating simple versions of all the screens you will need for your app. Separate the app into different labeled sections. Group the screens into these sections and show the section's main screen at the top of the section, and pages that link off of that screen below. Set up a numbering system so you can reference the screens easily, or use our numbering system. We like to number each section and add decimals the deeper you go into that section.



Illustration de la multiplication des écrans dans une application mobile

# PROBLÉMATIQUES

**Or, une application mobile peut être interrompue dans son fonctionnement par différents événements :**

- interruption de la connexion internet,
- appel téléphonique, notifications, sms...
- extinction du téléphone volontaire ou involontaire (batterie),
- bug (ex : consommation de la RAM).

# PROBLÉMATIQUES

Pour se prémunir des soucis engendrés par les coupures intempestives de l'application, il peut être utile de :

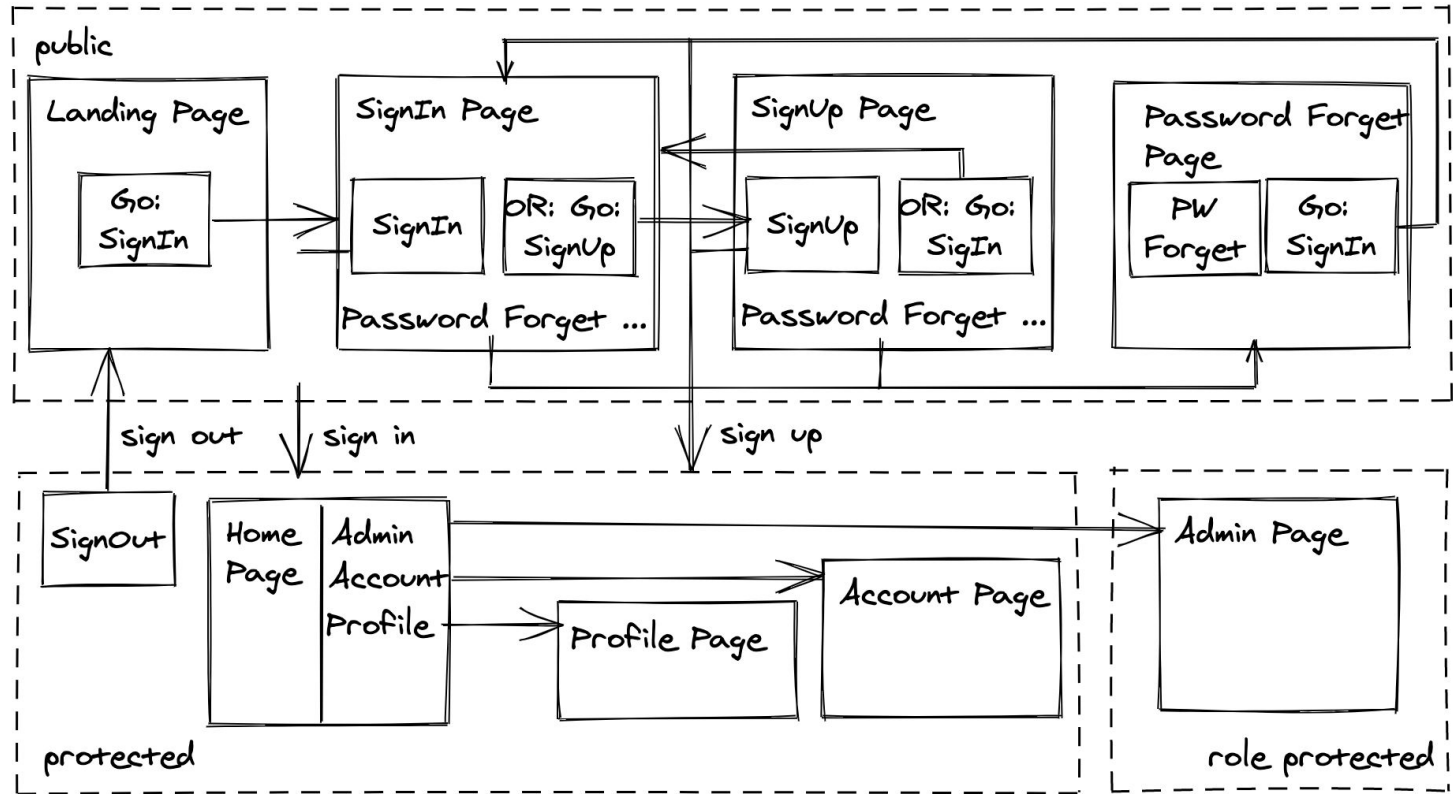
- **sauvegarder les informations sur l'état de l'application en cours d'utilisation,**
- **sauvegarder les données dans une base de données locale,**

# PROBLÉMATIQUES

Idéalement, après une interruption, il faudrait rétablir l'application dans son état précédent : écran et données.

**L'utilisateur risque en effet d'être agacé si l'application reprend "*à zéro*" après une interruption.**

## Authentication Flow



Écrans et états d'une application mobile  
(workflow sign in / sign up / sign out)



# PROBLÉMATIQUES

**Gestion des demandes d'autorisations :  
respect de la vie privée de l'utilisateur**

# PROBLÉMATIQUES

Selon leur périmètre fonctionnel, les applications mobiles vont devoir **demander l'autorisation aux utilisateurs afin d'accéder et interagir avec les données et capteurs du terminal (couche hardware).**

Sans l'autorisation de l'utilisateur, si l'application ne propose pas un scénario alternatif, elle sera inutilisable.

# PROBLÉMATIQUES

Attention, la nature des demandes de permission peut évoluer avec le temps, rendant potentiellement incompatibles des applications déjà publiées.

La mise en place d'une prestation de maintenance pourra là encore être une solution.

# A closer look at mobile permissions one year into GDPR

With **GDPR** reaching its one year anniversary May 25, Airship revealed top-level results of its global benchmark study, examining the state of mobile app user permissions across nearly 700 million people worldwide.

Avec l'arrivée du RGPD (*Règlement général sur la protection des données*), les demandes d'autorisation auprès de l'utilisateur se sont multipliées.

Region Audience & Notification Growth

	Audience Growth YoY	2019 Notification Opt-In Rate	% YoY Notification Change	Average Sends per Month	% YoY Avg Sends Change
Americas	+17.1%	64.4%	-9%	36.6	+22.4%
Europe	+21.9%	64.9%	-5%	35	+15.5%
Asia	+47.2%	69.5%	-6%	56.9	+2.5%
Oceania	+17.5%	52.5%	-5.4%	22	-3.5%
Africa	+50.3%	78.8%	+9.9%	18.6	-33.3%

iOS User Location Settings




Foreground Allowed	35.1%	-3.9%
Always Allowed	6.1%	-7.4%
Not Allowed	10.8%	-17.9%
Unprompted	45.2%	+14%
System Location Disabled	2.5%	-18.9%

Android User Location Settings




Always Allowed	45%	-11.6%
Not Allowed	49.8%	+36.2%
System Location Disabled	4.3%	-57.2%
Unknown	.9%	-63.2%


<https://www.helpnetsecurity.com/2019/05/24/mobile-permissions-gdpr/>

 Allow **App Name** to access your calendar?


2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to access your contacts?


2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to record audio?


2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to send and view SMS messages?


2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to take pictures and record video?


2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to access this device's location?

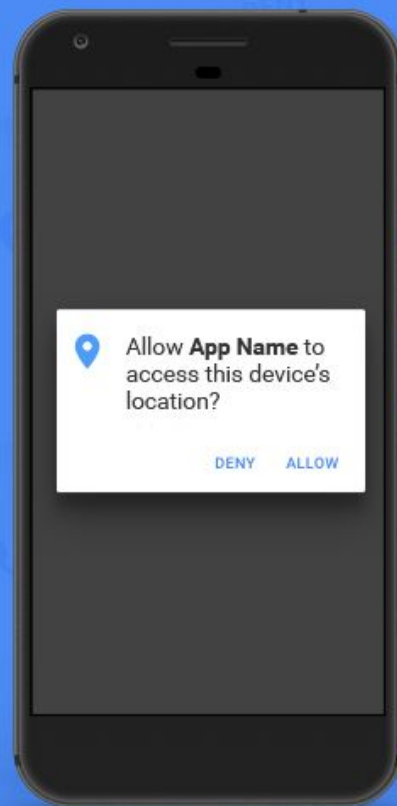
2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to make and manage phone calls?

2 of 2 [DENY](#) [ALLOW](#)

 Allow **App Name** to access photos, media, and files on your device?

2 of 2 [DENY](#) [ALLOW](#)



# PROBLÉMATIQUES

- Wifi,
- Bluetooth,
- GSM,
- NFC,
- Orientation du device,
- GPS,
- Caméra,
- Audio (Magnétophone),
- Accéléromètre,
- Gyroscope,
- Magnétomètre,



**Accelerometer**



**Gyroscope**



**Compass**



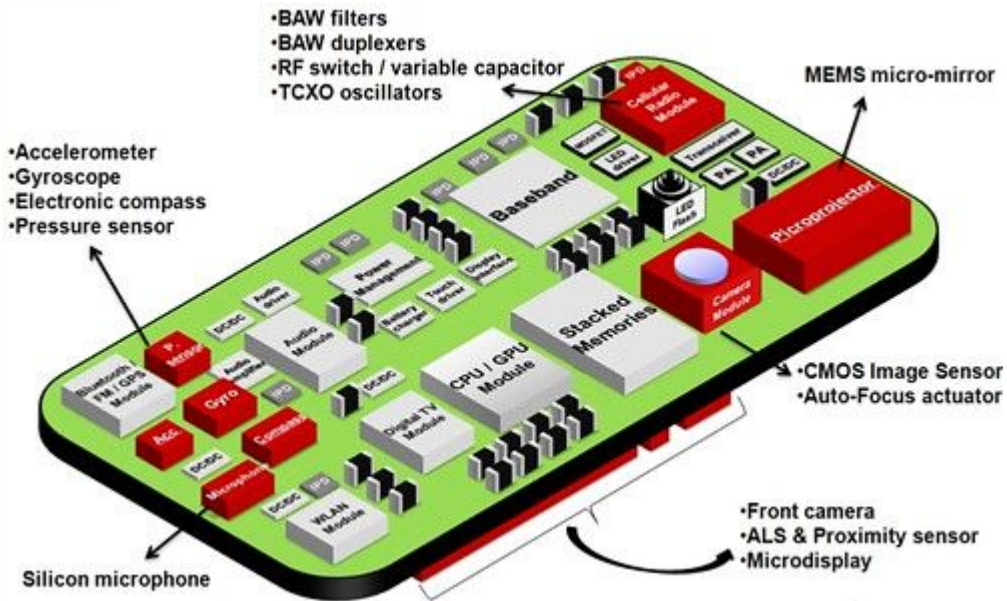
**GPS**



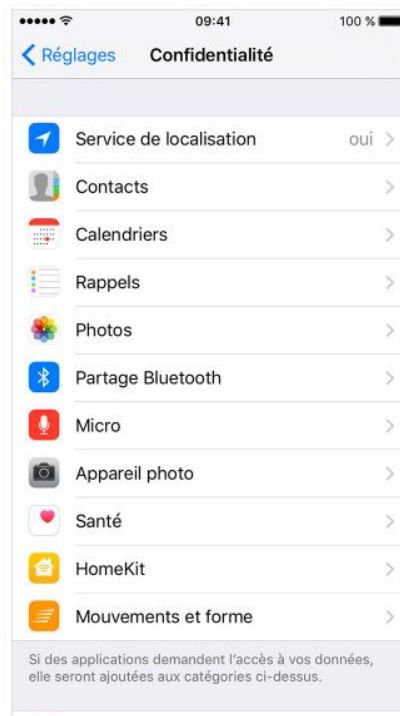
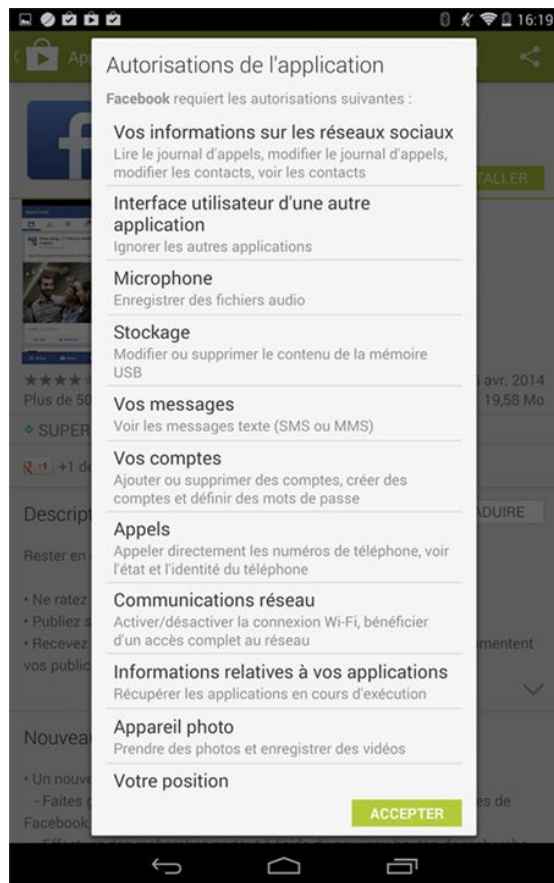
**Light sensor**



**Barometer**



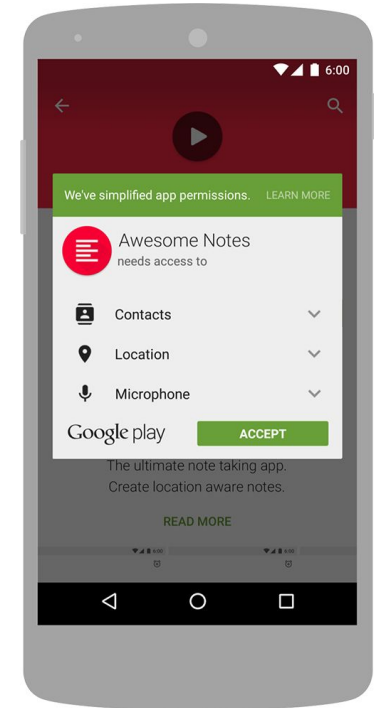
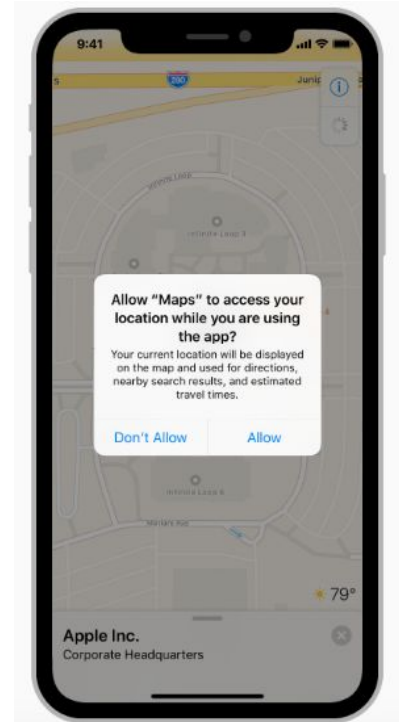
## Capteurs mobiles





# PROBLÉMATIQUES

**Les applications doivent appliquer des demandes d'autorisation pour chaque usage de données ou de capteur du terminal (appareil photo, géolocalisation, stockage interne, empreinte digitale...).**



# PROBLÉMATIQUES

## Obsolescence d'une application mobile distribuée

# PROBLÉMATIQUES



# PROBLÉMATIQUES

Les normes technologiques / politiques des distributeurs peuvent également rendre incompatible / indisponible une application préalablement validée et distribuée.

# PROBLÉMATIQUES

Si l'un des distributeurs décide de ne plus accepter les applications réalisées avec tel framework, langage, version... l'application peut devenir obsolète.



# PROBLÉMATIQUES

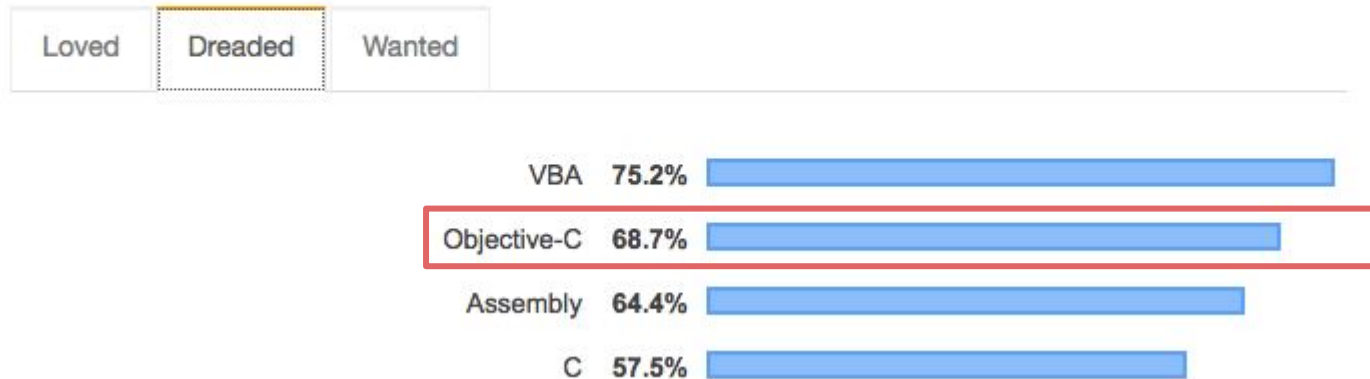
**Apple est plus sujet aux changements qu'Android.**

Apple a par exemple décidé de promouvoir son nouveau langage de développement, Swift, au détriment d'Objective-C, jugé trop complexe par les développeurs.

**Progressivement, Objective-C sera abandonné par Apple.**

# PROBLÉMATIQUES

## Most Loved, Dreaded, and Wanted Languages



**D'après un sondage de Stack Overflow de 2019,  
Objective-C arrive en 2ème position des langages de développement les plus craints**

<https://insights.stackoverflow.com/survey/2019#most-loved-dreaded-and-wanted>

```

- (id) init
{
    self = [super init];
    targetPeripheral = [NSString string];
    _length = 127;
    _data = [NSMutableData dataWithLength:_length];
    gattlookup = [[BTGattLookup alloc] init];

    targetPeripheral = nil;
    targetCharacteristic = nil;
    characteristicWriteData = nil;

    return self;
}

```

[object performAction:firstParameter];

```

graph TD
    subgraph Code
        direction TB
        C1[object]
        C2[method name being called]
        C3[parameter of method]
        C4[object.performAction(firstParameter);]
    end
    C1 --> C4
    C2 --> C4
    C3 --> C4

```

object      method name being called      parameter of method

object.performAction(firstParameter);

Class name      Parent class

```

@interface Book : NSObject {
    id data;
    int pages;
    NSString *title;
    NSString *author;
}

- (id) initWithTitle: (NSString *)aTitle;
- (NSString *) getTitle;
- (void) setAuthor:(NSString *)anAuthor;
- (NSString *) getAuthor;

+ (id) createBookWithTitle:(NSString *)aTitle;
+ (id) createBookWithTitle:(NSString *)aTitle
    andAuthor:(NSString *)anAuthor;

@end

```

Member variables

Instance Methods

Class Method



Objective - C

Objective-C



# PROBLÉMATIQUES

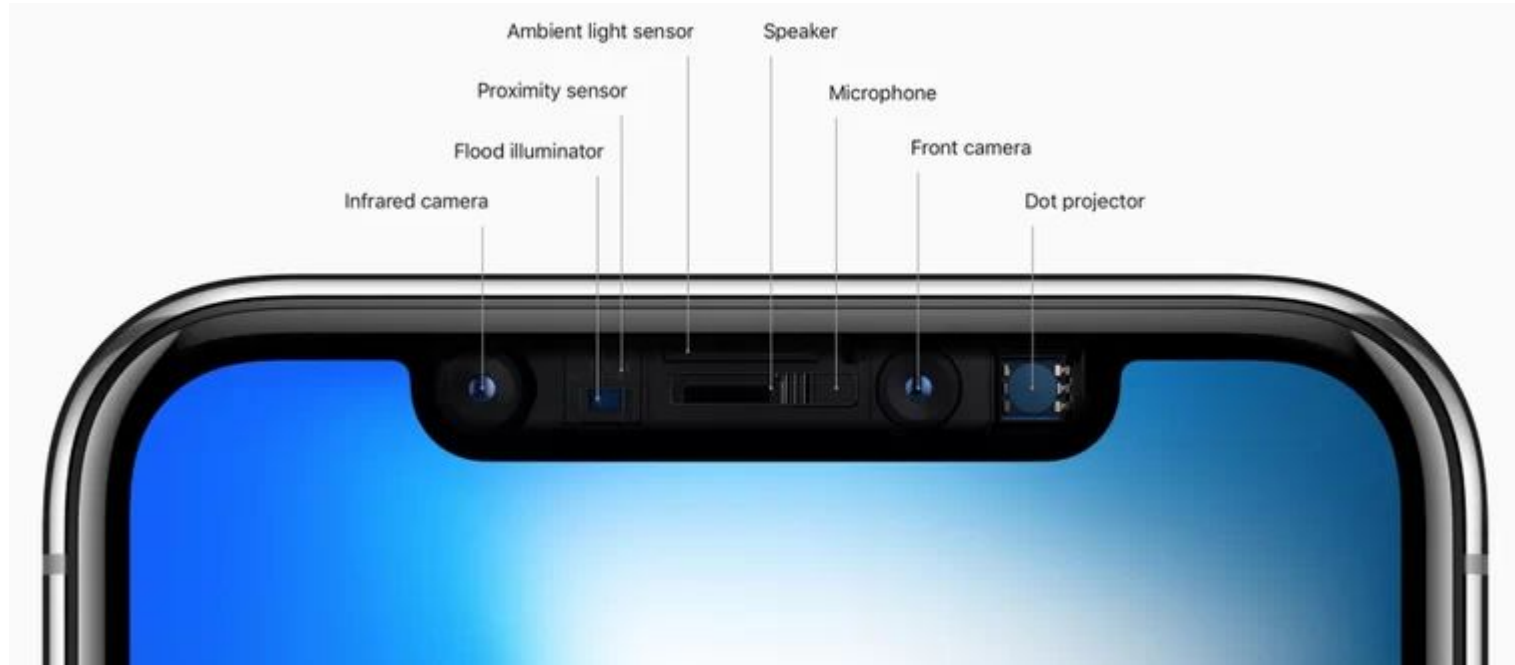
Les développeurs mobiles, spécialisés sur la distribution à destination des terminaux Apple ont donc tout intérêt à “*suivre le mouvement*” sous peine de devenir eux-mêmes obsolètes...

# PROBLÉMATIQUES

L'iPhone dispose d'un écran en partie masqué par des “*encoches*”, ce qui implique des adaptations de l'interface (UI).

**Or, Apple a décidé que toutes les nouvelles applications mobiles iOS devraient être compatibles avec son modèle iPhone X et iOS 11.**

# PROBLÉMATIQUES



Les “*encoches*” de l’iPhone X

# PROBLÉMATIQUES



## Update your apps for iPhone X.

Download Xcode 9 and start testing now with the iPhone X simulator to make sure your apps are ready to take advantage of the stunning Super Retina display. Learn about respecting safe areas, supporting adaptive layouts, and more.

<https://developer.apple.com/ios/updates-apps-for-iphone-x/>

**Mail envoyé par Apple aux développeurs iOS**

# PROBLÉMATIQUES

## Build Your Apps for iOS 11 and iPhone X

iOS 11 has brought innovative features and the redesigned App Store to hundreds of millions of customers around the world. Your apps can deliver more intelligent, unified, and immersive experiences with Core ML, ARKit, new camera APIs, new SiriKit domains, Apple Music integration, drag and drop for iPad, and more.

Starting April 2018, all new iOS apps submitted to the App Store must be built with the iOS 11 SDK, included in Xcode 9 or later. All new apps for iPhone, including universal apps, must support the Super Retina display of iPhone X.

<https://developer.apple.com/ios/update-apps-for-iphone-x/>

**Mail envoyé par Apple aux développeurs iOS**

# PROBLÉMATIQUES

## Respect Safe Areas

Your layout should fill the all-screen Super Retina display while content and controls appear correctly and are easily tappable. Use these safe areas to make sure your app works properly with the new dimensions of iPhone X.

Apps with custom layouts can also easily support iPhone X, especially if your app uses Auto Layout and adheres to these guidelines.



## Nouvelles spécifications iOS suite à la sortie de l'iPhone X

# PROBLÉMATIQUES

**Home Indicator.** Make sure your app doesn't interfere with Home Indicator, so your content and controls remain clearly viewable and tappable at all times.



**Status bar.** Allow the proper amount of space below the status bar based on the safe area. Avoid assuming a value for the status bar height, as it may cause your content to be obscured or mispositioned below the status bar.



Nouvelles spécifications iOS suite à la sortie de l'iPhone X

# PROBLÉMATIQUES

Du côté d'Android, le langage Java est également menacé par :

- le langage Kotlin : <https://kotlinlang.org/>
- le framework Flutter cross platforms : <https://flutter.dev/>,

Kotlin et Flutter sont des solutions promues par Google qui pourraient à terme remplacer Java, et même Android.



# PROBLÉMATIQUES

## Why Kotlin?



### Concise

Drastically reduce the amount of boilerplate code.

[See example](#)



### Safe

Avoid entire classes of errors such as null pointer exceptions.

[See example](#)



### Interoperable

Leverage existing libraries for the JVM, Android, and the browser.

[See example](#)



### Tool-friendly

Choose any Java IDE or build from the command line.

[See example](#)

<https://kotlinlang.org/>

# PROBLÉMATIQUES

[Docs](#)[Showcase](#)[Community](#)[Get started](#)

Try our Flutter Create contest. [Learn more.](#)

The hero section of the Flutter website. It features a large, semi-transparent blue circle in the center. Overlaid on this circle is a black smartphone frame. Inside the frame is a photo of three people (two women and one man) looking at a smartphone. The text 'Publish iOS & Android apps' is written across the center in a large, blue, sans-serif font. The word 'Publish' is in blue, 'iOS & Android' is in white, and 'apps' is in blue. In the bottom right corner, there is a small grey button with the text 'Applications'. In the bottom left corner, there is a small blue button with the text 'Customize'. In the top right corner of the hero section, there is a small blue button with the text '--release'. In the bottom left corner of the hero section, there is a small blue button with the text 'CustomizeButton/'.

Publish iOS & Android apps

Applications

# PROBLÉMATIQUES

**Certaines rumeurs prétendent que Google travaille sur un nouvel OS universel qui remplacerait Android / Chromium.**

Il est probable également que les solutions de développement soient revues à cette occasion.

**Le framework Flutter serait un prétendant sérieux au remplacement de Java pour le développement d'application vers le futur OS de Google.**



**Le SUCCESSEUR d'Android ? Google Fuchsia, on vous dit tout !**

<https://www.youtube.com/watch?v= 2pnrxMEcdk>

# PROBLÉMATIQUES

## Fuchsia : des vétérans au sein de Google veulent remplacer Android et Chrome OS

Bloomberg a pu obtenir de nouvelles informations sur le projet Fuchsia, et la confirmation qu'une équipe au sein de Google travaille bien à la fin d'Android.

[https://www.frandroid.com/marques/google/518802\\_fuchsia-des-veterans-au-sein-de-google-veut-remplacer-android-et-chrome-os](https://www.frandroid.com/marques/google/518802_fuchsia-des-veterans-au-sein-de-google-veut-remplacer-android-et-chrome-os)



# PROBLÉMATIQUES

Apple a décidé de promouvoir son dernier iPhone phare, **le modèle iPhone X**, célébrant les 10 ans de l'iPhone.

Ce modèle a la particularité de ne pas disposer de bouton mécanique, au profit d'une manipulation 100% tactile sur la quasi-totalité de la surface de l'appareil.

# PROBLÉMATIQUES



Developer update

Hello Google Play Developer,

This is a [reminder](#) that starting **November 1, 2018**, updates to apps and games on Google Play will be required to target Android Oreo (API level 26) or higher. After this date, the Play Console will prevent you from submitting new APKs with a `targetSdkVersion` less than 26.

Configuring your app to target a recent API level ensures that users benefit from significant security and performance improvements, while still allowing your app to run on older Android versions (down to the `minSdkVersion`).

## Action required

Please ensure that your apps are configured to target at least Android 8.0 (API level 26) by **November 1, 2018**. For technical advice on how to change your app's target API level to meet these requirements, refer to the [migration guide](#).

La distribution sur le Google Play pour Android nécessite également des adaptations des applications existantes pour se conformer aux nouvelles spécifications des versions supérieures



# PROBLÉMATIQUES

## Meet Google Play's target API level requirement



Google Play [will require](#) that *new apps* target at least Android 8.0 (API level 26) from August 1, 2018, and that *app updates* target Android 8.0 from November 1, 2018.

Every new Android version introduces changes that bring significant security and performance improvements – and enhance the user experience of Android overall. Some of these changes only apply to apps that explicitly declare support through their `targetSdkVersion` manifest attribute (also known as the target API level).

Configuring your app to target a recent API level ensures that users can benefit from these improvements, while still allowing it to run on older Android versions. Targeting a recent API level also allows your app to take advantage of the platform's [latest features](#) to delight your users.

This document highlights important points you should know in updating your target API level to meet the Google Play requirement. To proceed, click on on your app's current API level:

- Earlier than [Android 5.0 \(API level 21\)](#).
- Earlier than [Android 6.0 \(API level 23\)](#).

### Sommaire

Earlier than Android 5.0  
(API level 21)

Earlier than Android 6.0  
(API level 23)

Earlier than Android 7.0  
(API level 24)

Earlier than Android 8.0  
(API level 26)

Modernizing your apps

Check and update your  
SDKs and libraries

Test your app

Further information

**Spécifications pour la mise à jour des applications existantes à destination des nouvelles versions Android.**



Bonjour,

Nous vous rappelons qu'à partir du **2 novembre 2020**, les applications et les jeux sur Google Play devront cibler [Android 10 \(niveau d'API 29\)](#) ou version ultérieure. Après cette date, la Play Console vous empêchera d'envoyer de nouveaux app bundles et APK dont la propriété `targetSdkVersion` est inférieure à 29. Sachez que le niveau d'API 29 n'est pas requis pour les applications [Wear OS](#). Ceci n'affectera pas votre fiche Play Store existante.

En configurant votre application pour qu'elle cible un niveau d'API récent, vous permettez aux utilisateurs de bénéficier d'améliorations significatives en termes de sécurité et de performances, tout en leur laissant la possibilité de l'exécuter sous une version antérieure d'Android (dans la limite définie par la propriété `minSdkVersion`).

#### Action requise

Veuillez vous assurer que toutes vos applications sont configurées pour cibler Android 10 au minimum (niveau d'API 29) avant le 2 novembre 2020. Si vous souhaitez obtenir des conseils techniques sur la modification du niveau d'API cible de votre application pour répondre à ces exigences, consultez le [guide de migration](#).

# PROBLÉMATIQUES

Pour les mêmes raisons que les effets causés par les évolutions du cadre légales liées à la vie privée, **il peut être nécessaire de proposer aux clients une prestation de maintenance adaptative afin de couvrir les évolutions logicielles, matérielles et stratégiques des distributeurs.**

**A minima, il est du devoir des développeurs d'informer les clients sur l'obsolescence potentielle de leur application.**