

# Développement Mobile

Solutions de développement  
mobile Cross Platforms



UNIVERSITÉ  
DE LORRAINE



2022 - Alexandre Leroux ([alex@sherpa.one](mailto:alex@sherpa.one))

# SOLUTIONS CROSS PLATFORMS

Le développement d'applications mobiles correspond au paradigme du "*logiciel distribué*" en opposition au logiciel partagé (web) :

- **Technologies natives** (SDK),
- **Compilation** de fichier binaire exécutable,
- **Versionning**,
- **Intégration au système d'exploitation** (gestion de la mémoire, accès aux fonctionnalités hardware et au système de fichiers...).

# SOLUTIONS CROSS PLATFORMS

Par défaut, les applications mobiles sont développées avec les **langages spécifiques à chaque plateforme permettant de compiler un binaire optimisé** :

- **Objective-C / Swift** : production d'un binaire **IPA** pour **iOS**,
- **Java / Kotlin** : production d'un binaire **APK** pour **Android**.



iOS



Android



# Programming for Android: What Language to Use?

<https://code.tutsplus.com/tutorials/programming-for-android-what-language-to-use--cms-37535>

	 Flutter	 Kotlin
 <b>Performance</b>	App speed is fast because of the Hot Reload functionality.	Apps build with Kotlin are efficient, secure, optimal and perform better.
 <b>Popularity</b>	Flutter is gaining almost near popularity as Kotlin, despite being very new	Kotlin is trending as compared to Flutter on Google trends.
 <b>Community</b>	Flutter has more substantial community support on GitHub.	Kotlin way lacks beat the Flutter community on GitHub.
 <b>UI Experience</b>	Users can build a custom user interface with Flutter	With Kotlin, users can have their native app experience
 <b>Testing Support</b>	It works as a B-a-a-S and provides some in-built widget testing and app testing features	It provides a rich set of unit testing functionality for Android apps.
 <b>Integration</b>	Flutter integrates well with the Android and iOS platforms.	Kotlin provides multiplatform availability for every native environment.
 <b>Minimum Supported Platform</b>	Android v4.1x+, v10(jelly bean), v16, and iOS 8+	Kotlin supports even the older versions of Android and iOS 8+.

## SOLUTIONS CROSS PLATFORMS

Pour distribuer une application sur l'intégralité du parc mobile (iOS + Android), il est donc nécessaire de développer **une version spécifique de l'application destinée à chaque plateforme, compilée avec le langage natif associé.**

# SOLUTIONS CROSS PLATFORMS

Cette particularité du développement mobile **multiplie les frais** :

- Pour les clients / **MOA** : frais de développement (App mobile + APIs), délais de réalisation, frais de maintenance, licence de distribution (iOS / Android),
- Pour les entreprises de développement / **MOE** : frais de recrutement, équipement, formation, distribution et maintenance, gestion des erreurs et retards de réalisation...

# Les différentes approches de développement mobile cross platforms

# SOLUTIONS CROSS PLATFORMS

Face à la problématique de développement mobile natif,  
**différentes solutions cross platforms ont émergé.**

L'objectif étant de réaliser des **applications mobiles à destination d'iOS et Android, sur la base d'un code source commun.**

# SOLUTIONS CROSS PLATFORMS

Pour la plupart des langages de programmation, un framework a proposé ou décliné une approche de développement mobile cross platforms.

Langage	Framework(s)
JS	React Native, Cordova, Ionic, NativeScript
Java	Codename One
Dart	Flutter
C++	QT
C#	MAUI (ex Xamarin), Unity, Uno, Godot
Ruby	Ruby Motion
Python	Kivy
Kotlin	Kotlin Mobile SDK
Go	Go Mobile

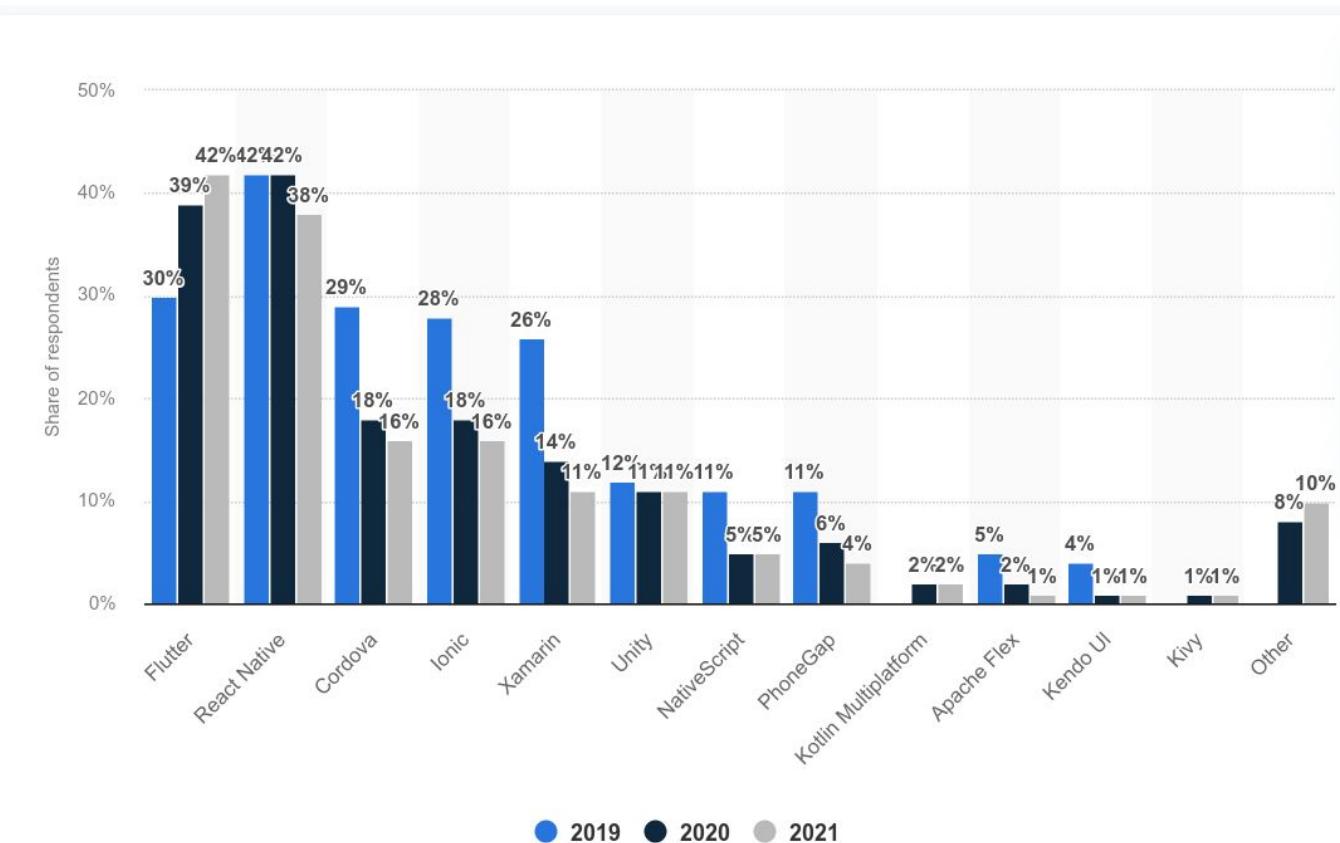
# SOLUTIONS

## Les solutions cross platforms les plus courantes

Les frameworks mobiles cross platforms les plus réputés sont :

- **Flutter,**
- **React Native,**
- **MAUI (ex Xamarin),**
- **Ionic Framework / Cordova.**

# Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021



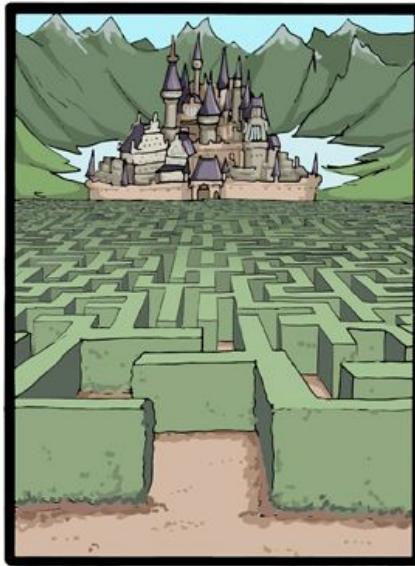
<https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>

# SOLUTIONS

## Etat de l'art en 2021

- **Flutter et React Native sont les frameworks leaders,**
- Flutter est considérée comme étant la solution la plus prometteuse pour l'avenir,
- Ionic Framework et Xamarin sont distanciés,
- Les PWA ne sont pas encore assez matures pour être envisagées en alternative aux applications natives,
- NativeScript, Unity, Kivy, Windev Mobile... sont considérées comme des solutions de "*niche*".

Développer une app native  
par plate-forme et maintenir  
3 codes-sources

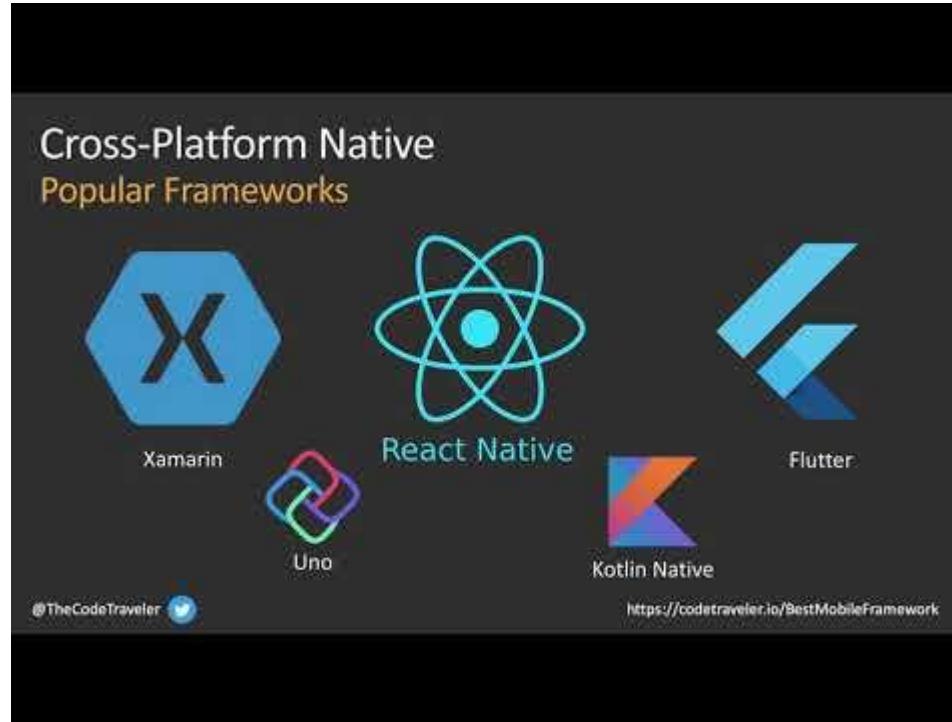


Utiliser une techno unique  
(Phonegap, Adobe Air,  
Appcelerator) et maintenir  
1 seul code-source



CommitStrip.com

<https://www.commitstrip.com/fr/2014/08/18/the-dilemma-of-mobile-apps-development/>



## Choosing the best mobile app framework



## Retour de Veille : Développement mobile avec Kotlin / Flutter / React Native

# XAMARIN vs REACT NATIVE vs IONIC vs NATIVESCRIPT COMPARISON

		Xamarin		React Native	NativeScript	Ionic
<b>Code</b>		C# + Java, Kotlin/Swift, Objective-C		JavaScript+Java, Kotlin/Swift, Objective-C	JavaScript/TypeScript+Java, Kotlin/Swift, Objective-C	HTML, CSS, TypeScript, JavaScript
<b>Compilation</b>	iOS	AOT		Interpreter	Interpreter	JIT+WKWebView
	Android	JIT/AOT		JIT	JIT	JIT
<b>Portability</b>		iOS, Android, Windows, Mac OS		iOS, Android	iOS, Android	iOS, Android
<b>Code reuse</b>		Xamarin iOS/Android	Xamarin.Forms	Up to 70 percent of code	Up to 90 percent of code	Up to 98 percent of code
		Business logic, Data access, Network communication	Up to 96 percent of code			
<b>UI engineering</b>		Native	Code sharing for the cost of native experience	Customization with built-in UI components	Code sharing for the cost of native experience	Code sharing for the cost of native experience
<b>Performance</b>		Close to native	Moderate-low	Close to native	Close to native	Moderate-low
<b>UI rendering</b>		Native UI controllers		Native UI controllers	Native UI controllers	HTML, CSS
<b>GitHub Stars</b>		5k		69.3k	15k	35.3k
<b>Price</b>		Open Source/Visual Studio for commercial use \$599-2,999		Open Source	Open Source/Sidekick cloud services for \$19-249	Open Source/Ionic Pro \$29-199
<b>Community</b>		Large		Growing	Growing	Large

Xamarin vs React Native vs Ionic vs NativeScript: Cross-platform Mobile Frameworks Comparison

<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>

# SOLUTIONS

## 5 approches de développement mobile cross platforms

1. Application web progressive (PWA),
2. Application hybride (web dans natif) : Ionic / Cordova,
3. Application mobile interprétée à l'exécution : React Native,
4. Application mobile interprétée à la compilation : Flutter.
5. Application mobile low code / no code

# SOLUTIONS

## Critères de comparaison entre les différentes approches de développement mobile cross platforms

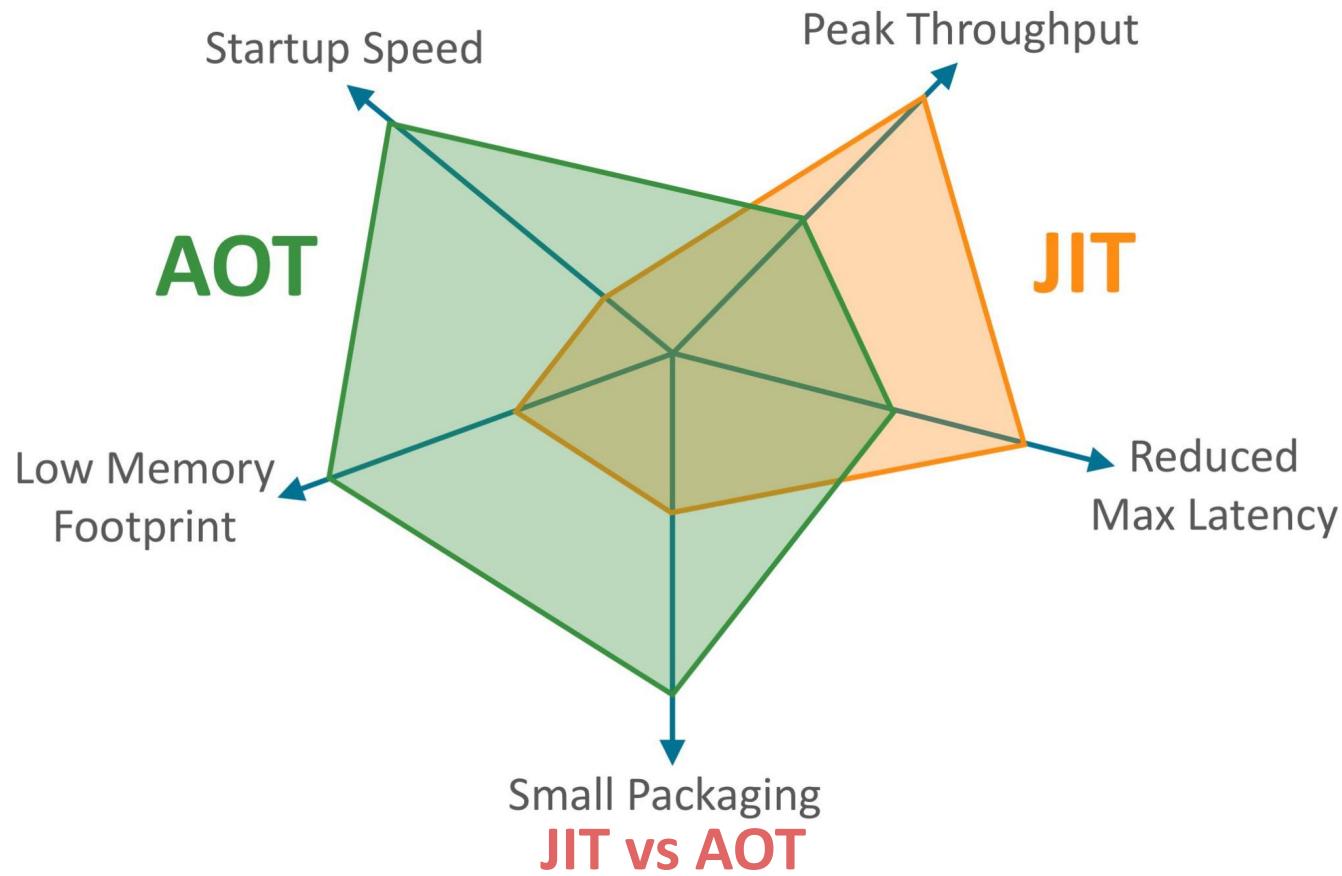
- Look and feel (UI / UX) :
  - **UI fidèle à la plateforme** : style Cupertino pour iOS et Material Design pour Android,
  - **UI agnostique** (identique quelle que soit la plateforme),

# SOLUTIONS CROSS PLATFORMS

- **Capacité à interagir avec la couche native** : interactions avec les capteurs (GPS, accéléromètre, lecteur d'empreinte...) et les fonctionnalités natives (systèmes de fichiers, camera, ...),
- **Plébiscite des entreprises** (demande, offres d'emploi...),
- **Réactivité** / mise à jour afin de garantir une compatibilité avec les nouvelles versions d'iOS et Android.

# SOLUTIONS CROSS PLATFORMS

- **Performance / fluidité / poids du binaire compilé :**
  - Langage tiers interprété au runtime (JIT),
  - Langage tiers interprété à la compilation (AOT),
- **Communauté de développeurs** (forums, tutoriels, articles, ouvrages, plugins...),
- **Quantité et qualité des composants et plugins,**



<https://learnlanga.medium.com/how-does-jit-and-aot-work-in-dart-cab2f31d9cb5>

# SOLUTIONS

## JIT (*Just in time*) = Run Time

A l'exécution (Runtime), le code tiers (Java, JS, Dart, C#...) est compilé par une machine virtuelle en langage natif (Java pour Android / Objective-C pour Swift).

En phase de développement la méthode JIT rend possible le "*Hot Reloading*" de l'application.

# SOLUTIONS

## AOT (*Ahead of time*) = Build Time

A la compilation, le code tiers (Java, JS, Dart, C#...) est converti en langage natif (Java pour Android / Objective-C pour Swift).

Le code est plus performant, puisqu'il est déjà compilé lorsqu'il est exécuté par le runtime, mais le fichier binaire généré est plus lourd.

# SOLUTIONS CROSS PLATFORMS

React Native, Xamarin et Flutter emploient à la fois les méthodes de compilation AOT et JIT.

# SOLUTIONS

## Solutions de développement de Jeux Vidéo Mobiles

A noter : des solutions de développement de jeux vidéo telles que **Unity 3D**, Godot, Cocos 2D peuvent être utilisées pour réaliser une application mobile cross platforms.

## Bien plus qu'un simple moteur

Unity dispose d'outils garantissant votre réussite à chaque étape de votre développement.



### Pipeline de rendu universel

Le **pipeline de rendu universel** est optimisé de façon à offrir des graphismes de qualité optimale, des performances et une évolutivité permettant de toucher un plus grand nombre d'appareils et de publics.

Il s'agit d'une solution de rendu puissante, prête à l'emploi avec une suite complète d'outils d'artiste pour la création de contenu, que vous réalisez des projets 3D, 2D, RV ou RA.

[En savoir plus](#)



### Outils 2D natifs

Unity a amélioré ses outils 2D pour vous aider à créer de superbes expériences 2D. Grâce à des flux de production efficaces, tous les membres de votre équipe, qu'ils soient spécialisés dans la création, la conception, l'animation ou l'ingénierie, peuvent réaliser davantage de projets.

[En savoir plus](#)



### Unity comme bibliothèque

Intégrez des fonctionnalités alimentées par Unity, telles que la RA, le rendu 3D/2D en temps réel, des mini-jeux 2D et plus encore, directement dans vos applications mobiles natives.

[En savoir plus](#)



### Système de ressources adressables

Ce système avancé rationalise le développement, la compilation et le chargement du contenu du jeu. En chargeant les ressources par « adresse », il maîtrise facilement la gestion des ressources, et simplifie la création et le déploiement des packs de contenu. Il accélère également les itérations, ce qui permet aux studios d'affiner et de tester



### Programmation de scripts visuels

Bolt dans Unity permettra aux membres de votre équipe de créer une logique de programmation de scripts à l'aide de graphes visuels déposés et glissés plutôt que du code saisi manuellement. Les artistes et les concepteurs avancent beaucoup plus vite dans leur travail dans l'Éditeur sans demander de l'aide aux programmeurs, ce qui permet à ces



### AR Foundation

Framework conçu pour le développement en RA, qui vous permet de concevoir des expériences immersives une seule fois, puis de les déployer sur divers appareils mobiles et de RA portables. Il comprend des caractéristiques particulières pour chaque plateforme, ainsi que des fonctionnalités Unity uniques, telles qu'un rendu photoréaliste, de la physique, des optimisations d'appareil et bien plus

# Développement mobile avec Unity 3D

<https://unity.com/fr/features/mobile>

# SOLUTIONS CROSS PLATFORMS

Ces solutions sont envisagées lorsque :

- l'interface emploie des **composants UI basiques** (boutons, menu déroulants, inputs...) ou à l'**apparence graphique personnalisée**,
- et/ou que l'animation multimédias et la performance ont une part importante dans le périmètre du projet.

## SOLUTIONS CROSS PLATFORMS

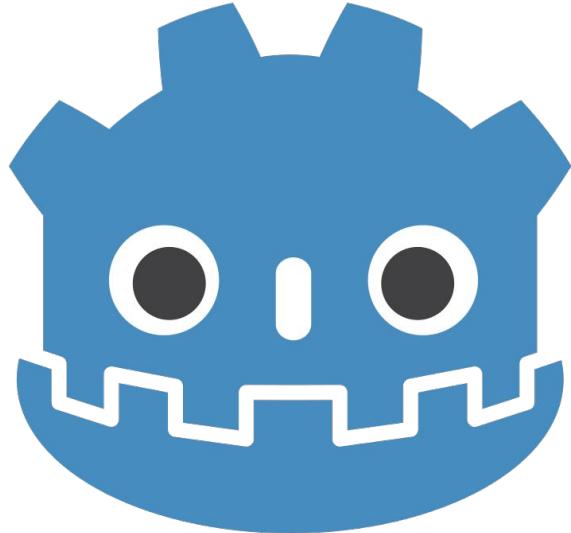
Les applications mobiles ludiques / pédagogiques à destination des enfants sont souvent réalisées avec ces solutions afin de réaliser des interfaces graphiques sur mesure.



unity



Développement d'application mobile  
cross platforms avec Unity 3D



**GODOT**  
Game engine

Godot

<https://godotengine.org/>

**Projet web intégré**  
dans la webview  
d'une application  
native (Cordova,  
Ionic...)

**Langage tiers (C++,  
Python, Dart..)  
converti en  
langage natif puis  
compilé en binaire**  
(MAUI, Kivy, QT,  
Flutter...)

**Langage tiers  
interprété au  
Runtime**  
(ReactNative,  
NativeScript)

**Langage natif  
compilé en  
binaire**  
(Swift, Objective-C,  
Java, Kotlin)

cross platforms

mono platform

## 4 approches de développement mobile

# SOLUTIONS

## Approche JIT (*Just In Time*)

PWA,  
Application hybride (Ionic),  
Application mobile  
interprétée à l'exécution  
(React Native).

## Approche AOT (*Ahead of Time*)

Application mobile  
interprétée et convertie à la  
compilation (Flutter).

# SOLUTIONS

## 1 - PWA (Progressive Web Application)

Application réalisée avec une **stack technologique web** (HTML / CSS / JS + assets) couplée à un fichier **manifest.json** et des **Service Workers JS**, capable de fonctionner en mode **off line** et interagir partiellement avec la couche native du terminal.

Interface graphique **responsive** (adaptation au format et à l'orientation de l'écran).

# SOLUTIONS

## 2 - Application hybride (web dans natif)

Intégration d'une application web (HTML / CSS / JS / assets) affichée au sein d'un conteneur de type webview d'une application mobile communiquant avec la couche native via l'API Cordova intermédiaire.

**Solution employée par :**

- **Cordova**
- **Ionic Framework**

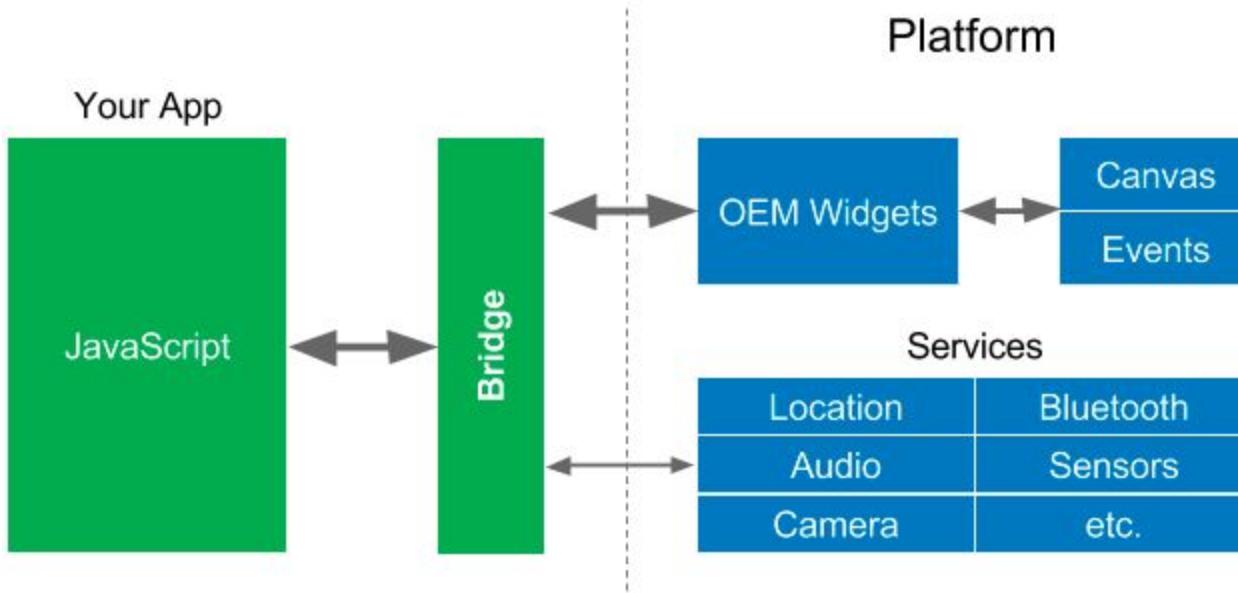
# SOLUTIONS

## 3 - Application interprétée à l'exécution

L'application est codée avec un langage de script non natif (ex : JS, Ruby, Python, C#...). L'interface est décrite à l'aide d'un langage de balisage neutre.

A l'exécution, les scripts sont interprétés par une machine virtuelle afin de générer et manipuler des composants mobiles natifs.

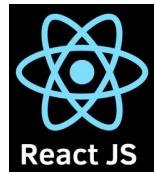
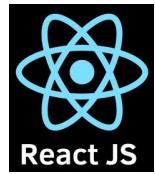
**Solution employée par React Native, NativeScript, Kivy, Xamarin...**



React Native emploie une couche intermédiaire nommée "**Bridge**"  
(JS / Java pour Android et C/C++ pour iOS)  
afin de communiquer avec la couche mobile native.

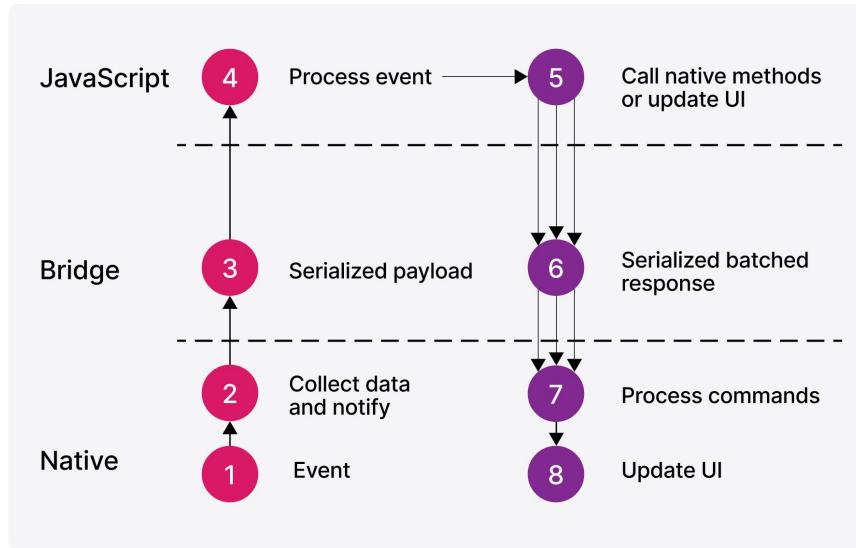
Contrairement à Cordova, React Native emploie des composants UI mobiles natifs.

## Fonctionnement de React Native



## React Native Bridge

<https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>

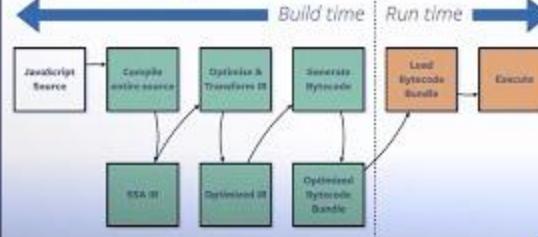


Communication entre la couche mobile native  
et la couche JS de React Native, via le Bridge



POWERED BY  
**{callstack}**

### Hermes Pipeline



JIT JS Engine Pipeline vs Hermes Pipeline

# SOLUTIONS

## 4 - Application interprétée à la compilation

L'interface graphique est intégralement générée par un langage tiers interprété en langage natif et compilé.

Plutôt que d'utiliser les composants UI des SDK iOS et Android, l'interface est dessinée sur-mesure.

**Solution employée par Flutter et Unity.**

## SOLUTIONS CROSS PLATFORMS

Une application Flutter initialement écrite en Dart, est compilée en code natif avec la bibliothèque ARM C/C++, ce qui lui confère un avantage certain en termes de performance.

	Native	Hybrid-Native	Hybrid-Web
<b>Examples</b>	iOS and Android SDKs	React Native, Xamarin, NativeScript, Flutter	Ionic
<b>Languages</b>	Obj-C, Swift, Java	JS + Custom UI Language / Interpreter	HTML + CSS + JS
<b>Code Reuse</b>	Totally Separate Code Bases per Platform	Shared Business Logic with Different UI Codebases	One codebase, UI codebase stays the same
<b>Target Platforms</b>	iOS & Android Native Mobile Apps	iOS & Android Native Mobile Apps	iOS, Android, Electron, Mobile and Desktop Browsers as a Progressive Web App, and anywhere else the web runs
<b>Investment</b>	Largest investment in staff and time	Medium investment in staff and time	Lowest investment in staff and time
<b>UI Elements</b>	Native UI independent to each platform	A selection of Native UI elements for iOS and Android UI elements are specific to the target platform and not shared	Web UI elements that are shared across any platform, conforming to the native look & feel of wherever they are deployed
		Custom UI elements begin to require split UI code bases	Easily use completely custom UI elements
<b>API Access / Native Features</b>	Separate Native API & Codebases for each App	Abstracted Single-Codebase Native Access through Plugins (with ability to write custom Plugins)	Abstracted Single-Codebase Native Access through Plugins (with ability to write custom Plugins)
<b>Offline Access</b>	Available	Available	Available
<b>Performance</b>	"Native Performance" with well written code.	Indistinguishable difference on modern devices with well written code. See below.	Indistinguishable difference on modern devices with well written code. See below.

## Native vs Hybrid Native vs Hybrid Web

<https://ionic.io/resources/articles/ionic-vs-flutter-comparison-guide>

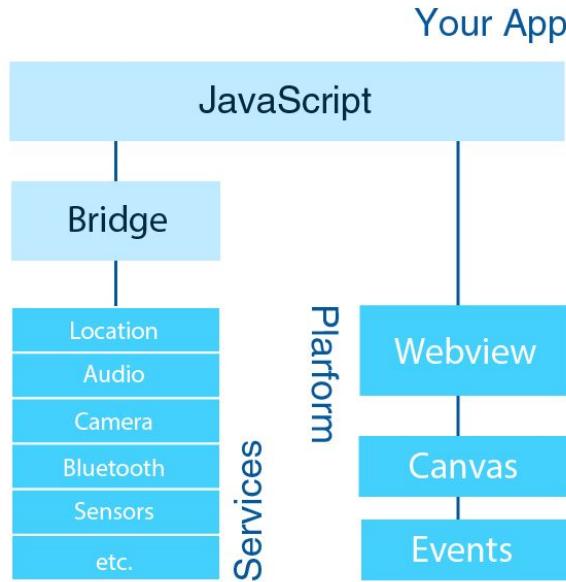
# Key Features: Native, Web, & Hybrid

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Full (with plugins)
Performance	High	Medium to High	Medium to High
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Medium to High
Code Reuse	No	Yes	Yes

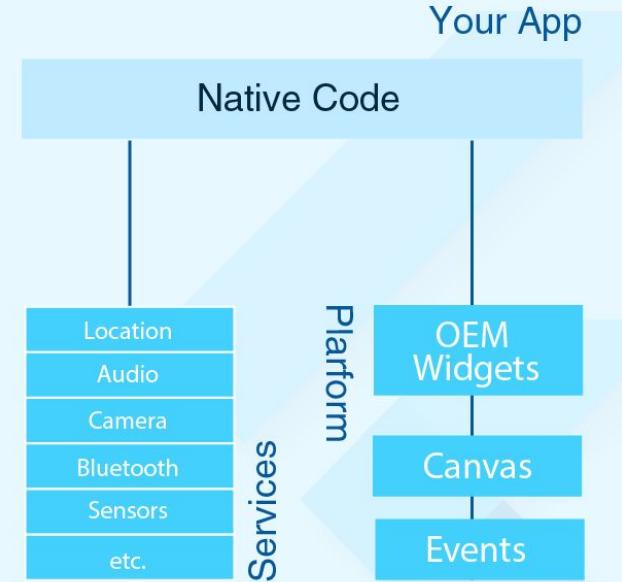
## Native vs Web vs Hybrid

<https://ionic.io/resources/articles/what-is-hybrid-app-development>

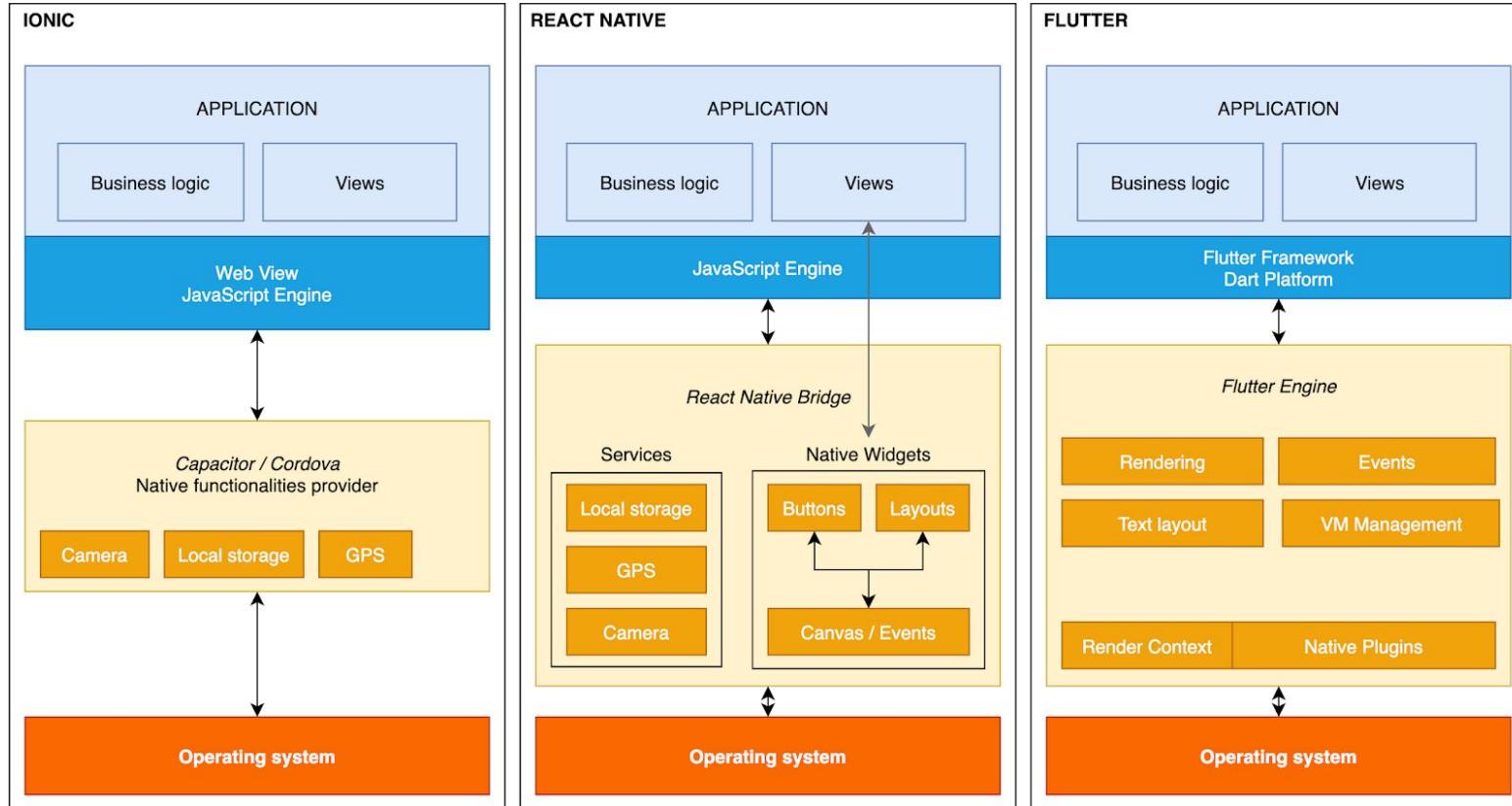
## React Native:



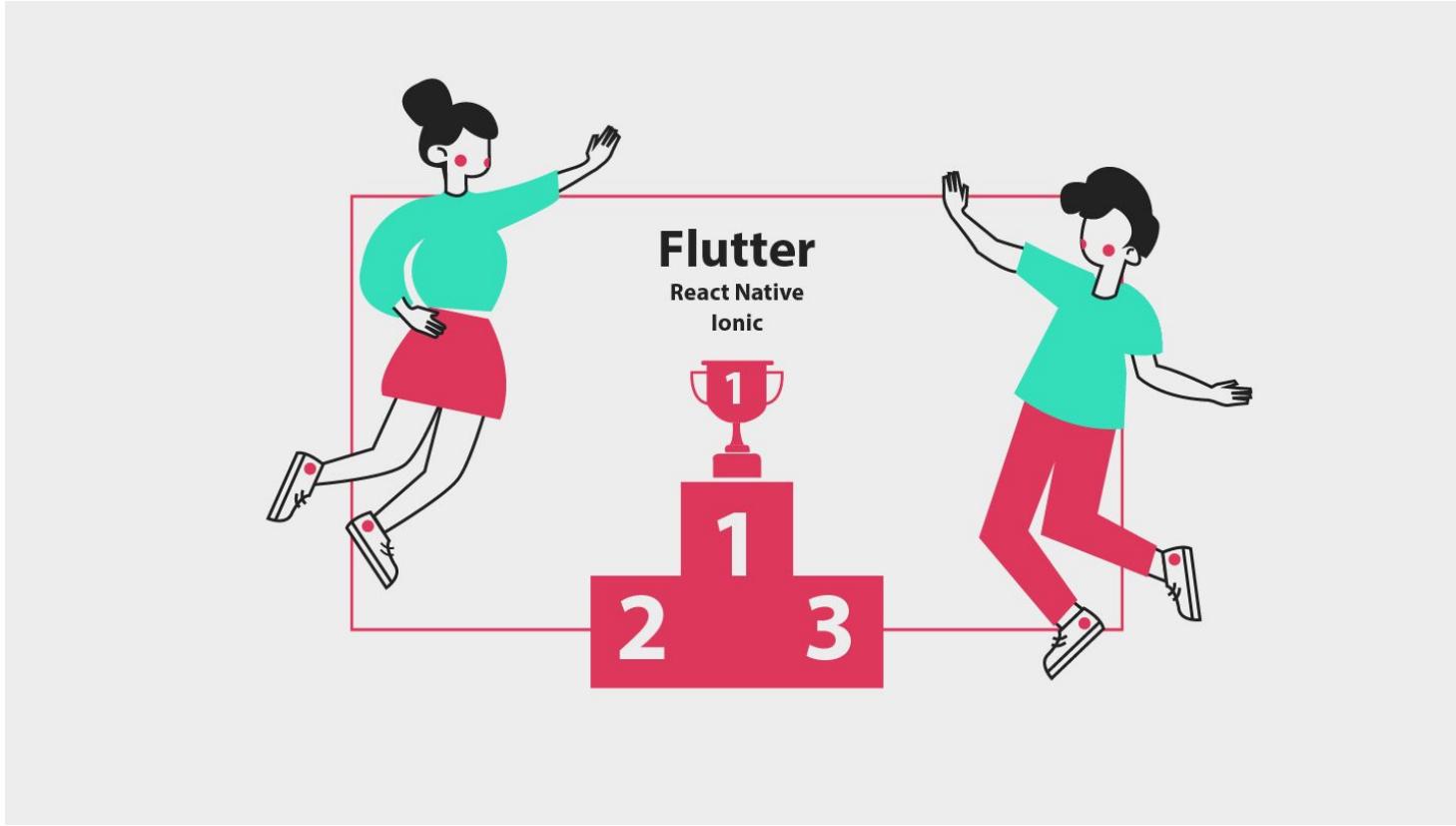
## Layout Flutter:



Différence des approches React Native / Flutter



## Comparaison des 3 approches techniques de développement mobile cross platforms



## Ionic vs React Native vs Flutter

<https://allbright.io/blog/ionic-reactnative-flutter-comparison>

# SOLUTIONS CROSS PLATFORMS

## Application mobile hybride



APACHE  
**CORDOVA**<sup>TM</sup>

Mobile apps with **HTML, CSS & JS**

Target multiple platforms with **one code base**

Free and **open source**

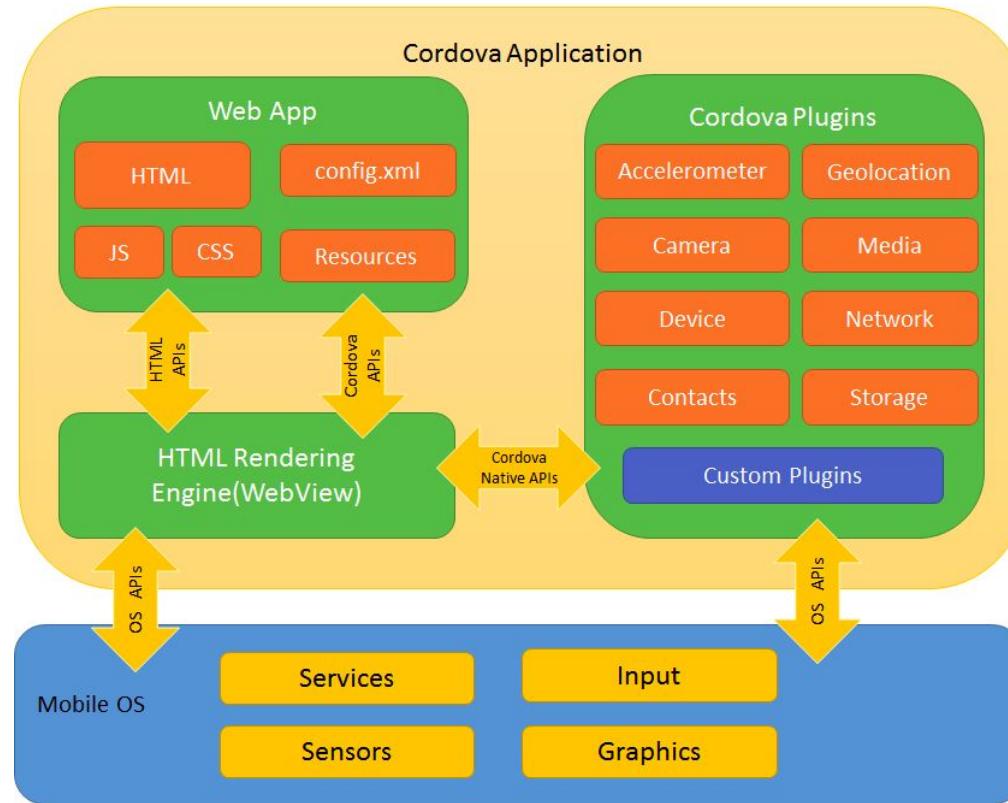
Android icon • Apple icon • Windows icon

[GET STARTED](#)   [DOCUMENTATION](#)

<https://cordova.apache.org/>

# SOLUTIONS CROSS PLATFORMS

Cette solution a été initialement proposée par **Apache Cordova**, puis reprise et adaptée par de nombreux autres frameworks (Adobe PhoneGap, Ionic Framework, Onsen UI...).



## Architecture de Apache Cordova

# SOLUTIONS CROSS PLATFORMS

Le concept de Cordova repose sur l'utilisation d'**une stack technologique Web (HTML, CSS, JS)** afin de générer une interface responsive et tactile, **affichée au sein d'un conteneur mobile natif de type *webview*.**

# SOLUTIONS CROSS PLATFORMS

**La webview est un composant mobile commun à Android et iOS** permettant d'afficher une page web dans un conteneur plein écran (full screen) et sans surcouche graphique (chrome less).

On peut comparer une webview (mobile) à une iframe (html).

## SOLUTIONS CROSS PLATFORMS

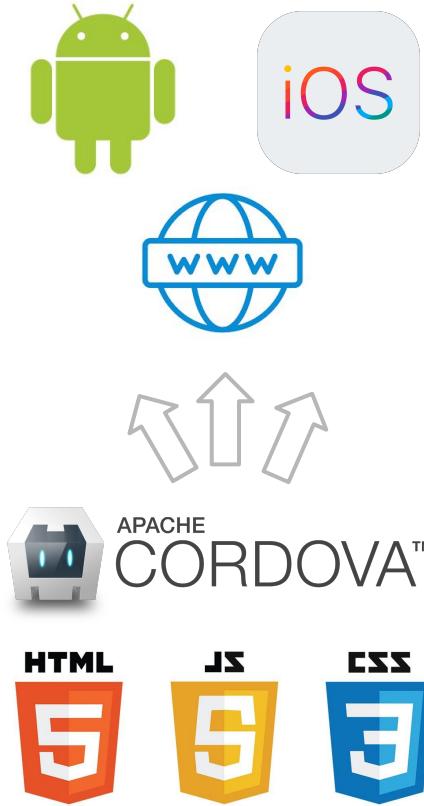
Les sources de l'application web (fichiers HTML, CSS, JS + assets) sont chargées dans la webview depuis un répertoire local de l'application mobile native (ce répertoire est intégré au binaire de l'application mobile).

# SOLUTIONS CROSS PLATFORMS

L'application web peut communiquer avec le réseau afin de :

- charger des données provenant d'APIs externes,
- charger des assets externes (JS, images, PDF, audio, vidéo...),





**Cordova emploie une stack technologique web commune  
pour proposer une solution mobile cross platforms**

# SOLUTIONS CROSS PLATFORMS

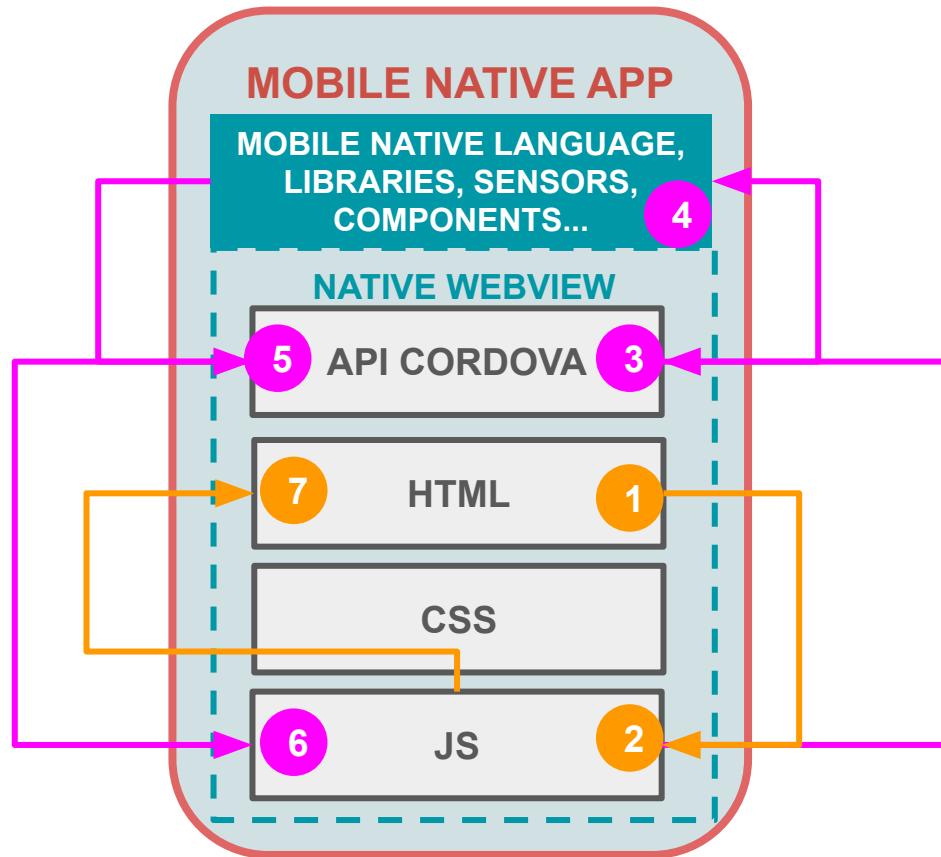
Lorsqu'une application Cordova est compilée en fichier binaire, elle prend “*l'apparence*” et le “*comportement*” d'une application native traditionnelle :

- distribution depuis le store d'applications de la plateforme,
- affichage d'une icône sur l'écran “*home*” du terminal mobile.

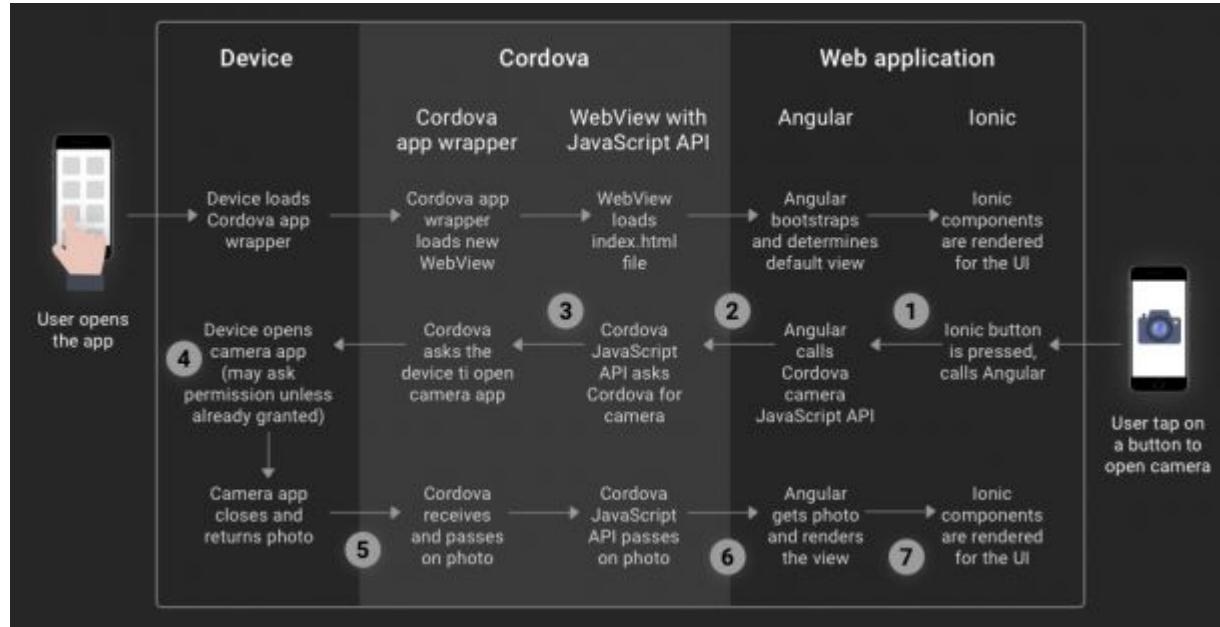
# SOLUTIONS CROSS PLATFORMS

Le langage **JS** communique avec la couche native à travers une API mise en place par Cordova.

L'API permet d'interpréter les signaux émis par JS et d'exécuter les commandes équivalentes en langage natif associé aux bibliothèques dédiées.



**Processus de communication  
entre la couche Web et la couche mobile native, via l'API Cordova**



## Ionic Framework stack internal working architecture

# SOLUTIONS CROSS PLATFORMS

Il est ainsi possible de faire appel à des fonctionnalités mobiles natives spécifiques (caméra, accéléromètre, batterie, système de fichiers...) depuis l'application web embarquée dans l'application native.

Platform:	Android	iOS	OS X	Windows 8.1, Phone 8.1, 10	Electron
CLI shorthand:	android	ios	osx	windows	electron
Cordova CLI Development Platform					
Mac	✓	✓	✓	✗	✓
Windows	✓	✗	✗	✓	✓
Linux	✓	✗	✗	✗	✓
Core Plugin APIs					
BatteryStatus	✓	✓	✗	✓ Windows Phone 8.1 only	Tests Pending
Camera	✓	✓	✗	✓	✓
Capture	✓	✓	✗	✓	Tests Pending
Connection	✓	✓	✗	✓	Tests Pending
Device	✓	✓	✓	✓	Tests Pending
Events	✓	✓	✗	✓	Tests Pending
File	✓	✓	✓	✓	Tests Pending
Geolocation	✓	✓	✗	✓	Tests Pending
Globalization	✓	✓	✗	✓	Tests Pending
InAppBrowser	✓	✓	✗	uses iframe	Tests Pending
Media	✓	✓		✓	Tests Pending
Notification	✓	✓	✗	✓	Tests Pending
Splashscreen	✓	✓	✗	✓	Tests Pending
Status Bar	✓	✓	✗	✓ WP 8.1 only	Tests Pending
Storage	✓	✓	✗	✓ localStorage & indexedDB	Tests Pending
Vibration	✓	✓	✗	✓ WP 8.1 only	✗

# SOLUTIONS

## Plugins Cordova

Cordova propose des plugins permettant à l'application de communiquer avec la couche native du terminal.

Ces plugins sont fournis sur le catalogue **NPM** par Cordova et des développeurs tiers.

# SOLUTIONS

## Capacitor

Capacitor est présenté par Ionic Framework comme le successeur naturel de Cordova en tant que moteur d'exécution web au sein d'une webview.

# SOLUTIONS CROSS PLATFORMS

Capacitor propose **une liste de plugins codés en langage natif (Objective C ou Swift / Java ou Kotlin)** par Ionic ou par des tiers, permettant d'exécuter des fonctionnalités natives du terminal mobile.

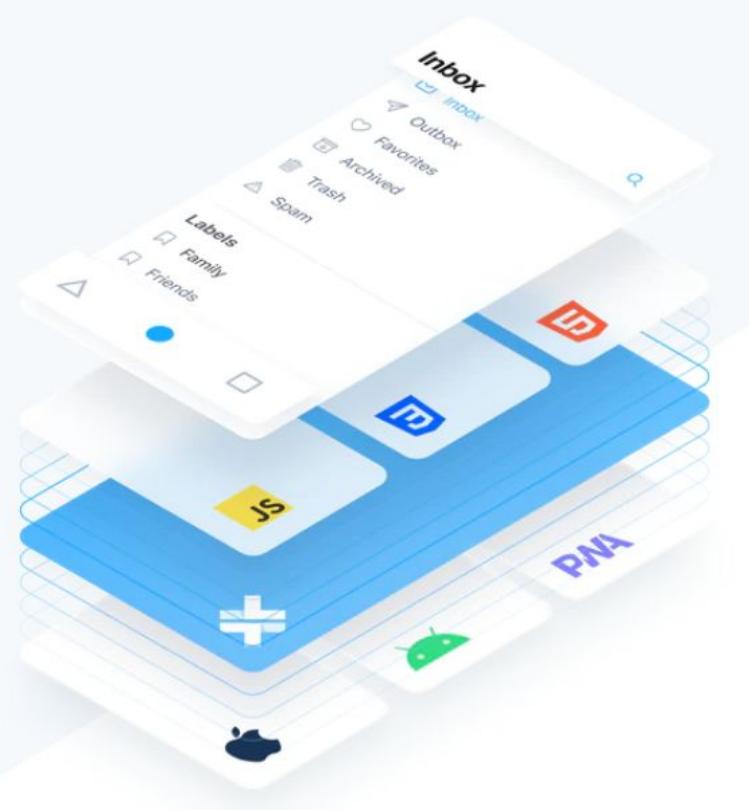
# A cross-platform native runtime for web apps.

Capacitor is an open source native runtime for building Web Native apps. Create cross-platform iOS, Android, and Progressive Web Apps with JavaScript, HTML, and CSS.

[Install Capacitor →](#)

[Explore Plugins](#)

[Migrate from Cordova →](#)



<https://capacitorjs.com/>



<https://capacitorjs.com/>

## Capacitor Plugins

This repository contains the official Capacitor plugins maintained by the Capacitor team. You can find more plugins in the [Capacitor Community](#).

⚠ These plugins are for Capacitor 3 ⚠

Capacitor 2 core plugins are bundled with Capacitor itself.

@capacitor/action-sheet	./action-sheet	npm v1.0.6
@capacitor/app-launcher	./app-launcher	npm v1.0.7
@capacitor/app	./app	npm v1.0.6
@capacitor/browser	./browser	npm v1.0.6
@capacitor/camera	./camera	npm v1.0.6
@capacitor/clipboard	./clipboard	npm v1.0.6
@capacitor/device	./device	npm v1.1.0
@capacitor/dialog	./dialog	npm v1.0.6
@capacitor/filesystem	./filesystem	npm v1.0.6
@capacitor/geolocation	./geolocation	npm v1.2.0
@capacitor/haptics	./haptics	npm v1.1.3
@capacitor/keyboard	./keyboard	npm v1.1.3
@capacitor/local-notifications	./local-notifications	npm v1.0.9
@capacitor/motion	./motion	npm v1.0.4
@capacitor/network	./network	npm v1.0.6
@capacitor/push-notifications	./push-notifications	npm v1.0.7
@capacitor/screen-reader	./screen-reader	npm v1.0.6
@capacitor/share	./share	npm v1.0.7
@capacitor/splash-screen	./splash-screen	npm v1.1.6
@capacitor/status-bar	./status-bar	npm v1.0.6
@capacitor/storage	./storage	npm v1.2.3
@capacitor/text-zoom	./text-zoom	npm v1.0.6
@capacitor/toast	./toast	npm v1.0.6

## Capacitor Official Plugins

<https://github.com/ionic-team/capacitor-plugins>



## Capacitor Community

The Capacitor Community org contains plugins, tools, and other projects that are driven by the community of Capacitor developers and maintainers.

[🔗 https://capacitorjs.com](https://capacitorjs.com)

[🐦 @capacitorjs](#)

## Capacitor Community

<https://github.com/ionic-team/capacitor-plugins>

## Xamarin/React Native



Native UI Controls

Native Runtime

Native APIs

Device OS

## Capacitor/Cordova

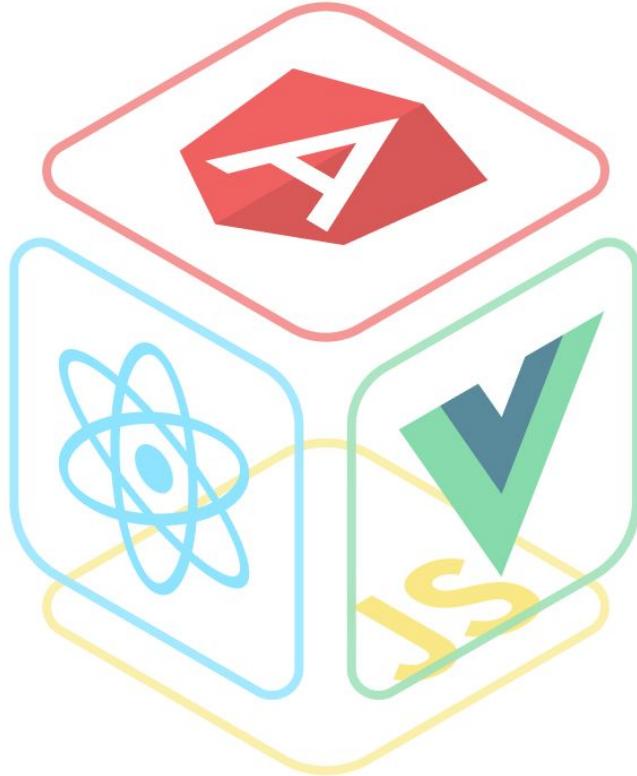


Web UI Controls

Native Runtime

Native APIs

Device OS



FRAMEWORK AGNOSTIC

## Build with the tech you prefer

We don't make assumptions about the tech stacks you or your team prefers to build with. That's why Ionic is engineered to integrate seamlessly with all best frontend frameworks, including Angular, React, Vue, or even no framework at all with vanilla JavaScript.

START WITH:    

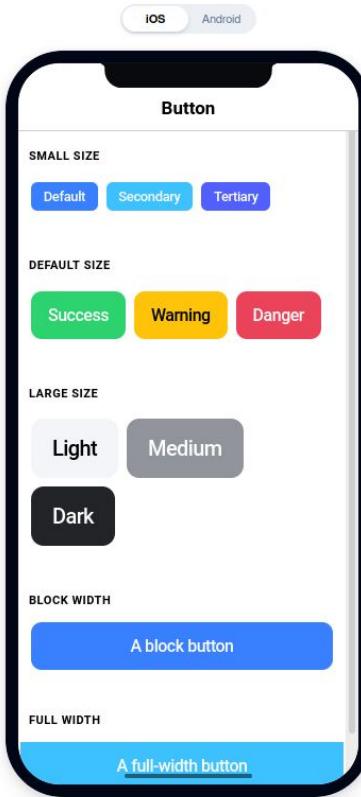
# Ionic Framework

<https://ionicframework.com/>

# SOLUTIONS CROSS PLATFORMS

**Ionic Framework** reprend la solution libre Apache Cordova et intègre un thème UI dont l'apparence dynamiquement s'adapte selon la plateforme d'exécution :

- style **Material Design** sur Android,
- style **Cupertino** sur iOS,



iOS



Android

Rendu dynamique d'un template Ionic Framework via CSS

**Action Sheet**  
Action Sheets display a set of options with the ability to confirm or cancel an action.

**Alert**  
Alerts are a great way to offer the user the ability to choose a specific action or list of actions.

**Badge**  
Badges are a small component that typically communicate a numerical value to the user.

**Input**  
Inputs provides a way for users to enter data in your app.

**Item**  
Items are an all-purpose UI container that can be used as part of a list.

**List**  
Lists can display rows of information, such as a contact list, playlist, or menu.

**Button**  
Buttons let your users take action. They're an essential way to interact with and navigate through an app.

**Card**  
Cards are a great way to display an important piece of content, and can contain images, buttons, text, and more.

**Menu**  
Menus are a common navigation pattern. They can be permanently on-screen, or revealed when needed.

**Modal**  
Modals slide in and off screen to display a temporary UI and are often used for login or sign-up pages.

**Checkbox**  
Checkboxes can be used to let the user know they need to make a binary decision.

**Chip**  
Chips are a compact way to display data or actions.

**Content**  
Content is the quintessential way to interact with and navigate through an app.

**Navigation**  
Navigation is how users move between different pages in your app.

**Popover**  
Popover provides an easy way to present information or options without changing contexts.

**Progress Indicators**  
Progress indicators visualize the progression of an operation or activity.

**Date & Time Pickers**  
Date & time pickers are used to present an interface that makes it easy for users to select dates and times.

**Floating Action Button**  
Floating action buttons are circular buttons that perform a primary action on a screen.

**Radio**  
Radio inputs allow you to present a set of exclusive options.

**Refresher**  
Refresher provides pull-to-refresh functionality on a content component.

## Ionic UI Components

<https://ionicframework.com/docs/components>

# SOLUTIONS CROSS PLATFORMS

**Cordova (ou Ionic) convient bien à des projets mobiles “basiques”.**

Les **limites de Cordova (ou Ionic)** sont cependant atteintes lorsqu'il s'agit de parfaitement s'intégrer au système d'exploitation (iOS / Android), d'interagir avec celui-ci ou encore de mettre en oeuvre des fonctionnalités spécifiques et/ou nécessitant une optimisation de la gestion de la RAM et du CPU.

# SOLUTIONS

## Avantages de l'intégration web dans une couche native

- **HTML + CSS + JS est une stack technologique libre, largement implantée :**
  - recrutement ais  de d veloppeurs Front End, g galement capables de travailler sur des projets web standards,
  - frais de formation et de documentation moindres,
  - relative facilit  de mise en oeuvre.

# SOLUTIONS CROSS PLATFORMS

- l'application web développée pour être intégrée dans une application mobile peut ensuite être **réemployée dans différents contextes, pour un "*usage universel*" :**
  - Web,
  - Desktop (via Electron),
  - PWA.

# SOLUTIONS CROSS PLATFORMS

- Possibilité d'utiliser ou de développer des plugins tiers codés en langage natif, associés à l'API Cordova.

<https://cordova.apache.org/docs/en/10.x/guide/hybrid/plugins/index.html>

# SOLUTIONS

## Inconvénients de l'intégration web dans une couche native

- L'interface UI (réalisée à base d'images et de CSS) **ne reprend pas intégralement les codes graphiques des composants mobiles natifs** pouvant causer une incohérence UI / UX),
- **Les performances et la fluidité de l'interface sont inférieures à celles d'une application native,**

PERFORMANCE OBSESSED

## Fast apps. Out-of-the-box.

Ionic is built to perform and run fast on all of the latest mobile devices. Build blazing fast apps with a small footprint and built-in best practices like hardware accelerated transitions, touch-optimized gestures, pre-rendering, and AOT compiling.

[Start building →](#)

Up to

**60** FPS

Scrolling on mobile  
and desktop

Consistent

**100%**

Lighthouse PWA  
performance score

Down to

**1.8** Sec

Time to  
Interactive

Up to

**2** x

Faster than previous  
versions

**Ionic Framework tente de proposer les meilleures performances  
avec sa stack technologique web**

# SOLUTIONS CROSS PLATFORMS

- Bien qu'il soit possible d'utiliser ou de développer des plugins ayant accès aux fonctionnalités natives mobiles, **les capacités fonctionnelles d'une application mobile hybride sont inférieures,**
- **Nécessité d'adapter le code natif (Java, Kotlin, Swift ou Objective-C) d'un plugin lorsque le cahier des charges du client est très spécifique.**

# SOLUTIONS CROSS PLATFORMS

- **Les interactions de l'utilisateur sont interprétées à l'exécution par l'API Cordova**, laquelle convertit les messages JS en instructions natives.  
Lorsque l'API reçoit une réponse de la couche native, elle transcrit à JS qui adapte le DOM HTML en retour.  
Ce processus asynchrone peut générer une latence désagréable en terme UX.

# SOLUTIONS CROSS PLATFORMS

- **Pour des raisons de performances en deçà des standards, mais aussi pour des raisons stratégiques (non emploi des langages natifs propriétaires), l'usage d'une webview en tant que moyen d'affichage intégral de l'interface est vu d'un mauvais œil par Apple et Google, ce qui, à terme, pourrait provoquer l'incompatibilité et la non distribution de ce type d'application mobile hybride depuis les stores.**

# SOLUTIONS CROSS PLATFORMS

In just over one week's time, Apple will begin rejecting new apps using UIWebView. We wanted to reach out with details on how to update your Ionic apps.

If you receive an email from Apple after uploading an app binary to the App Store, you'll need to update the app. The email reads something like this:

*We identified one or more issues with a recent delivery for your app, [App Name & version number]. Your delivery was successful, but you may wish to correct the following issues in your next delivery:*

*ITMS-90809: Deprecated API Usage - Apple will stop accepting submissions of apps that use UIWebView APIs. See <https://developer.apple.com/documentation/uikit/uiwebview> for more information.*

*After you've corrected the issues, you can use Xcode or Application Loader to upload a new binary to App Store Connect.*

You can find complete details including upgrade instructions on the [Ionic blog](#), but here's the summary:

- Using Cordova? Update to cordova-ios 5.1.0 and above, install the WKWebView plugin, and update each Cordova plugin to the latest version.
- Using Capacitor? Update to the [latest version](#) then sync the project.
- Submit a new version of your app before the deadline to ensure that it has been updated correctly.

If you're an Ionic Enterprise customer, your Customer Success Manager should already be in touch. If you're not a customer and would like to learn more about Enterprise Support options, please [reach out here](#) for extra assistance. Our Enterprise Support team is available and able to help. We've already helped a number of our customers navigate this change quickly & easily.

Cheers,

The Ionic Team

**En mars 2019, Apple annonce qu'elle refusera les applications mobiles qui utilisent l'API du composant "UIWebView", exploitée par les solutions basées sur Cordova.**

# SOLUTIONS CROSS PLATFORMS

## Note

In apps that run in iOS 8 and later, use the `WKWebView` class instead of using `UIWebView`. Additionally, consider setting the `WKPreferences` property `javaScriptEnabled` to `false` if you render files that are not supposed to run JavaScript.

**Depuis mars 2019,  
il est nécessaire d'utiliser l'API du composant WKWebView.  
<https://developer.apple.com/documentation/uikit/uiwebview>**

# SOLUTIONS CROSS PLATFORMS

## Medium



Florent Morin

[Follow](#)

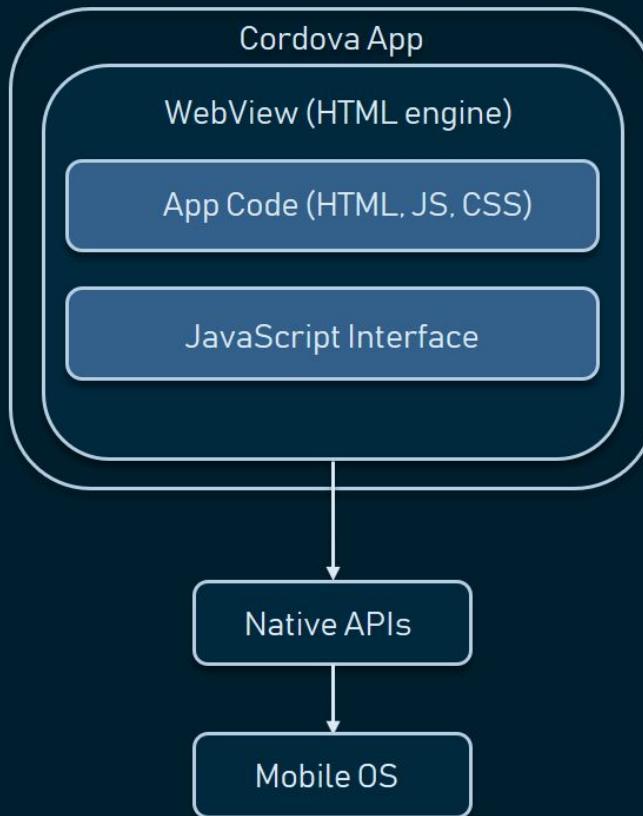
Creative technologist - Freelance  
Mar 4 · 7 min read

### Comment et pourquoi les apps hybrides web vont mourir sur iOS

Cordova et Ionic sont les principaux acteurs à vendre du rêve depuis des années. Un rêve qui pourrait se transformer en cauchemar.

<https://medium.com/@florentmorin/comment-et-pourquoi-les-apps-hybrides-web-vont-mourir-sur-ios-3eecc8dc8656>

# WEBVIEW RENDERING IN IONIC



Xamarin vs  
React Native vs  
Ionic vs  
NativeScript:  
Cross-platform  
Mobile  
Frameworks  
Comparison

<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>

## Application mobile interprétée à l'exécution

# SOLUTIONS CROSS PLATFORMS

En alternative à la solution Cordova, les autres frameworks misent sur l'**emploi d'un langage intermédiaire de plus haut niveau, interprété à l'exécution (Runtime)**.

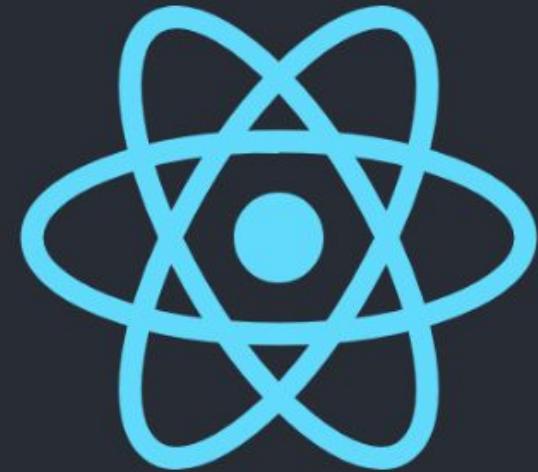
Il s'agit de l'approche **JIT ("Just In Time")**, qui s'oppose à l'approche **AOT ("Ahead Of Time")**.

# React Native

Learn once, write anywhere.

Get started

Learn basics >



React Native  
<https://reactnative.dev/>

# SOLUTIONS CROSS PLATFORMS



## SOLUTIONS CROSS PLATFORMS

En 2012, Mark Zuckerberg déclare : *"La plus grande erreur que nous ayons commise en tant que société a été de parier trop sur HTML par opposition au natif"*.

Lors d'un *hackathon* en 2015, les ingénieurs de Facebook esquissent le projet React Native.

# SOLUTIONS CROSS PLATFORMS

React Native reprend, côté mobile, une partie des concepts clés  
React.js, côté web :

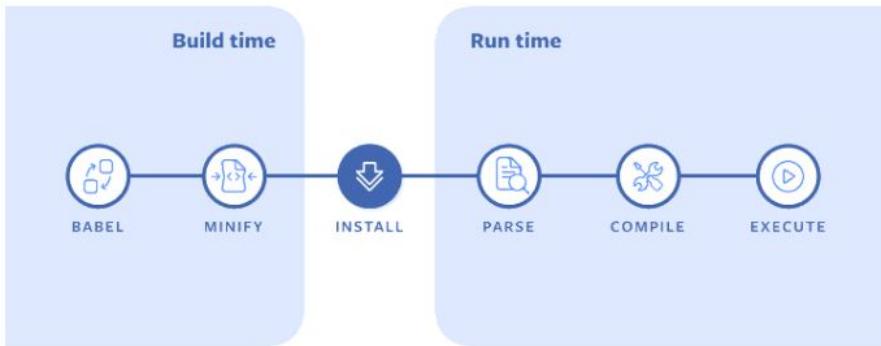
- JS E6+,
- Templates JSX,
- Paradigmes Déclaratif et Réactif,
- Programmation fonctionnelle,
- Gestion d'état (State)

# SOLUTIONS CROSS PLATFORMS

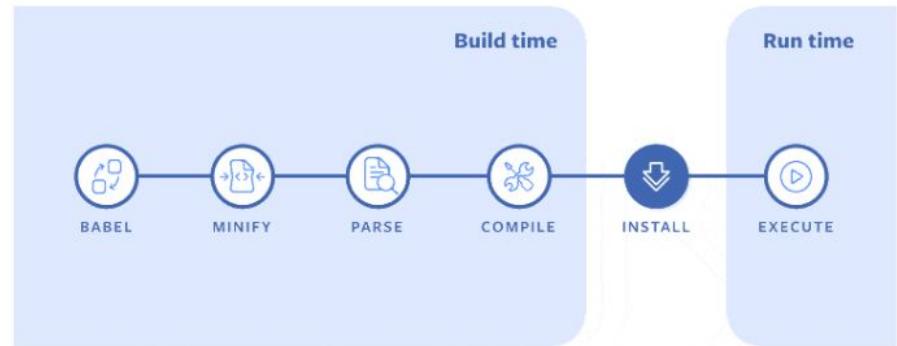
Pour exécuter JavaScript, React Native utilise **JavaScriptCore** (runtime JS de Safari), à la fois pour Android et pour iOS.

React Native va prochainement utiliser un nouveau moteur JavaScript exclusivement optimisé pour React Native : **Hermes**.

## Interpretation with conventional engine



## Bytecode precompilation with Hermes



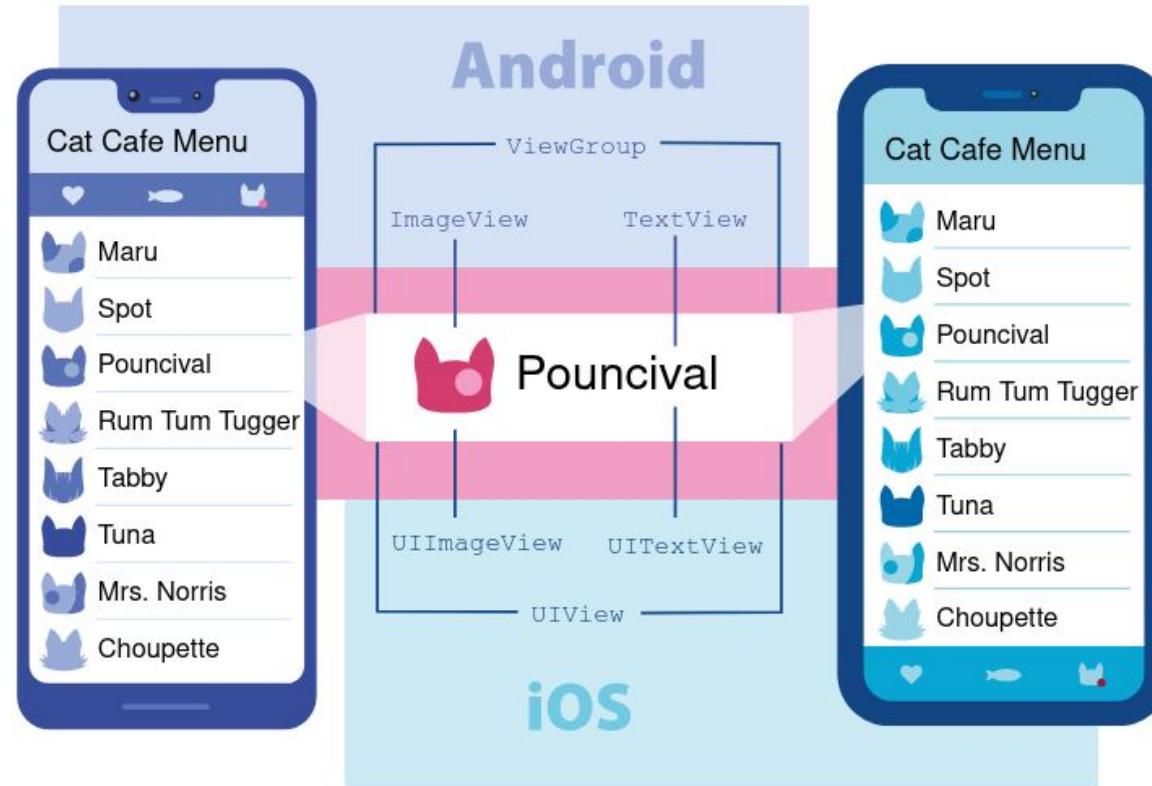
Byte Precompilation with Hermes

Avec son nouveau moteur JS, Hermes,  
React Native devrait améliorer ses performances

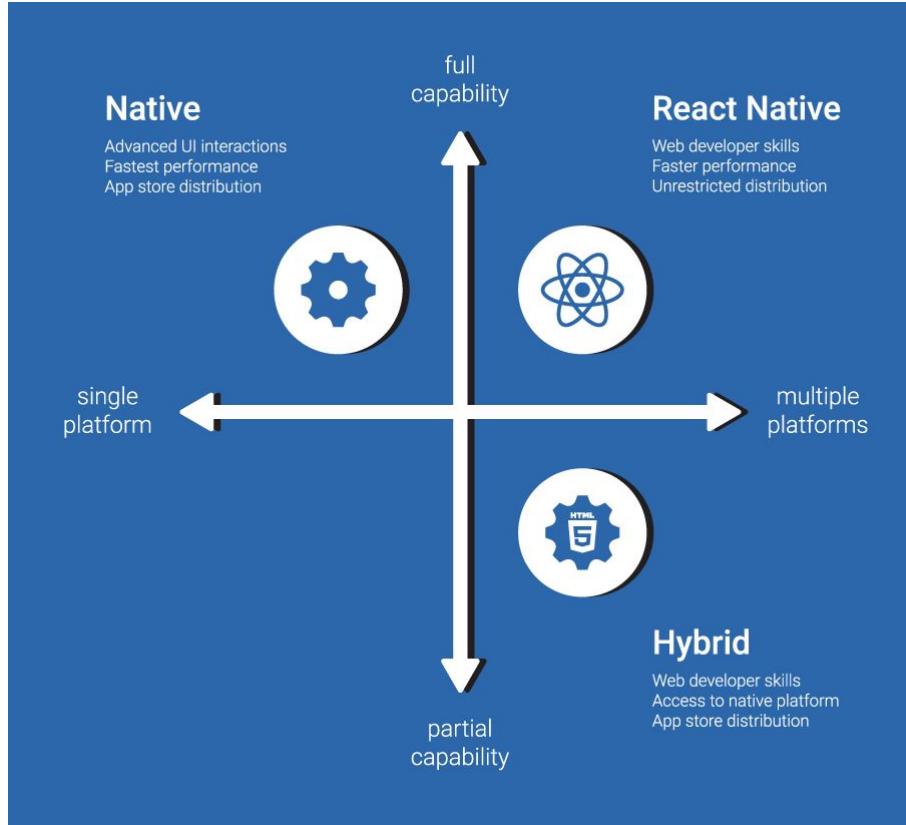
<https://medium.com/habilelabs/hermes-a-new-javascript-engine-for-react-native-4369ad17881f>

# SOLUTIONS CROSS PLATFORMS

- **JS** est utilisé en tant que **langage de script**,
- Le code **JS** est converti à l'exécution par un "*Bridge*" en **instructions en langage natif** (Java ou Objective-C),
- Les balises **JSX** sont converties à l'exécution en **composants mobiles natifs** selon la plateforme (iOS / Android).
- **Réactivité** : à mesure que l'état (State) de l'application évolue, seules les parties de l'UI concernées sont mises à jour



Adaptation dynamique de l'UI  
selon la plateforme d'exécution



## Native vs React Native vs Hybrid



## Retour d'expérience sur React Native

<https://medium.com/@Nextoo/react-native-a6ea0138f1b7>

## SOLUTIONS CROSS PLATFORMS

React Native connaît rapidement **un fort succès autant auprès des développeurs** (notamment, les utilisateurs de React.js) **que des entreprises.**

Pendant plusieurs années, React Native est la solution cross platforms de référence.



Quelques célèbres applications mobiles  
réalisées avec React Native

# Why Airbnb is Moving Off Of React Native



By Adam Conrad

Article | Monday, September 24 2018

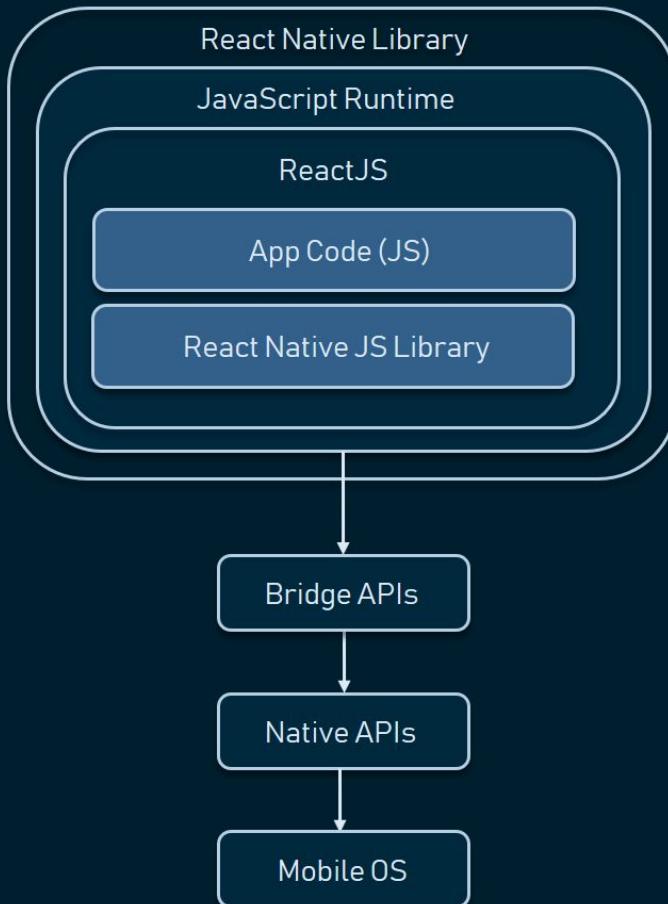
The image shows a red background with the white Airbnb logo on the left and a white React Native logo (an atom symbol) next to it. Below them, the words "React Native" are written in large white font, and "Airbnb" is written in a smaller white font below it.

The screenshot displays the "Explore Airbnb" screen at 12:30 AM. It features a search bar with the placeholder "Try 'Cape Town'", a "Carrier" signal icon, and a battery level icon. Below the search bar, there are three main categories: "Homes", "Experiences", and "Restaurants". Under "Experiences", there are two visible items: "BOAT RIDE" (Sail past Japan's largest port with...) and "CHEESE TASTING" (Funny cheesemonger takes you on a Tour...). To the right of these, there is a "See all >" button. The overall design is clean and modern, typical of a native mobile application.

En 2018, Airbnb décide d'abandonner React Native pour la réalisation de son application mobile, au profit du développement natif, ce qui porte un coup à l'image de marque de React Native.

<https://softwareengineeringdaily.com/2018/09/24/show-summary-react-native-at-airbnb/>

# REACT NATIVE RENDERING



Xamarin vs  
React Native vs  
Ionic vs  
NativeScript:  
Cross-platform  
Mobile  
Frameworks  
Comparison

<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>



NativeScript is pretty cool!

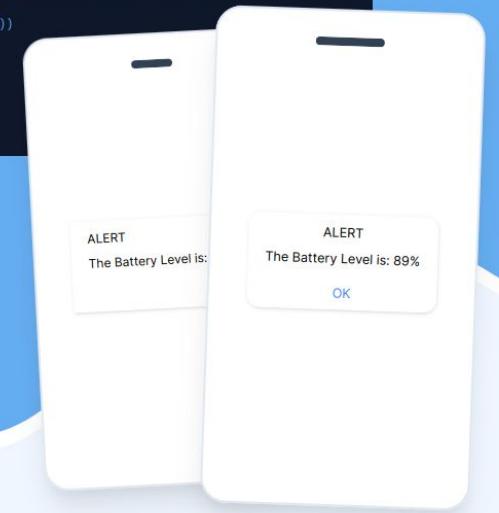
# Empower JavaScript with native APIs

Liberate your development by using platform APIs directly without leaving your ❤️ of JavaScript.

```
myApp/src/app.ts
const formatMessage = level => `The Battery Level is: ${level}%`  
// iOS direct native access!  
let value = UIDevice.currentDevice.batteryLevel * 100  
  
// Android  
let value = bm.getIntProperty(BatteryManager.BATTERY_PROPERTY_CAPACITY)  
  
alert(formatMessage(value))
```

Playground

Btw, you can totally try it!



# NativeScript

<https://nativescript.org/>

# SOLUTIONS CROSS PLATFORMS

**NativeScript**, le framework JS concurrent de React Native, reprend en partie son mode de fonctionnement.

En revanche, NativeScript est "*agnostique*" il peut aussi bien intégrer JS (Vanilla), Typescript, Angular, Vue ou React.



# SOLUTIONS CROSS PLATFORMS

Pour exécuter JavaScript, NativeScript utilise

- **V8 (runtime JS de Chrome)** pour Android,
- **JavaScriptCore (runtime JS de Safari)** pour iOS.

Contrairement à React Native qui utilise JavaScriptCore à la fois pour iOS et Android.

# SOLUTIONS CROSS PLATFORMS

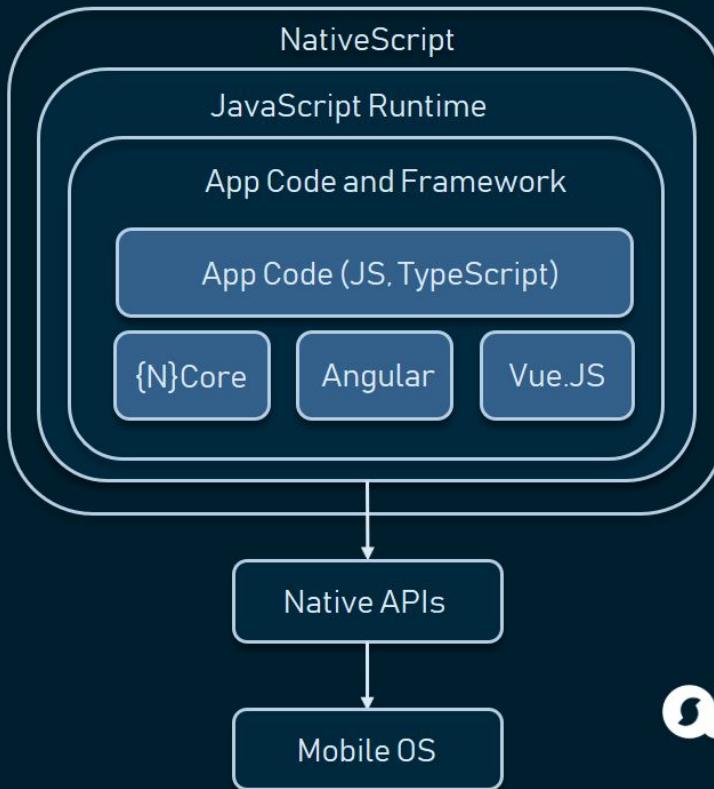
Malgré de bonnes idées, **NativeScript dispose d'une communauté de développeurs beaucoup plus faible que celle de React Native**, ce qui freine sa progression et son adoption :

- moins de tutoriels et articles en ligne / livres,
- documentation moins aboutie et moins mise à jour,
- moins de demande de la part des entreprises.

# SOLUTIONS CROSS PLATFORMS



# NATIVESCRIPT RENDERING



Xamarin vs  
React Native vs  
Ionic vs  
NativeScript:  
Cross-platform  
Mobile  
Frameworks  
Comparison

<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>

# Xamarin

Free. Cross-platform. Open source.

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

Get Started

Develop on Windows and macOS



## Mobile Apps

Build native Android, iOS, tvOS, watchOS, macOS, and Windows apps with .NET with native API access



## Cross-platform

Develop native, cross-platform mobile apps and shared native user interface



## Tooling

Develop across Windows and Mac with world class developer tools for the entire app life cycle



## Ecosystem

Integrate with modern backend services, native APIs, components, and more

# Xamarin

<https://dotnet.microsoft.com/apps/xamarin>

# SOLUTIONS

## Xamarin

Xamarin est la solution de développement mobile cross platforms proposée par **Microsoft**.

Xamarin se décline en 2 versions :

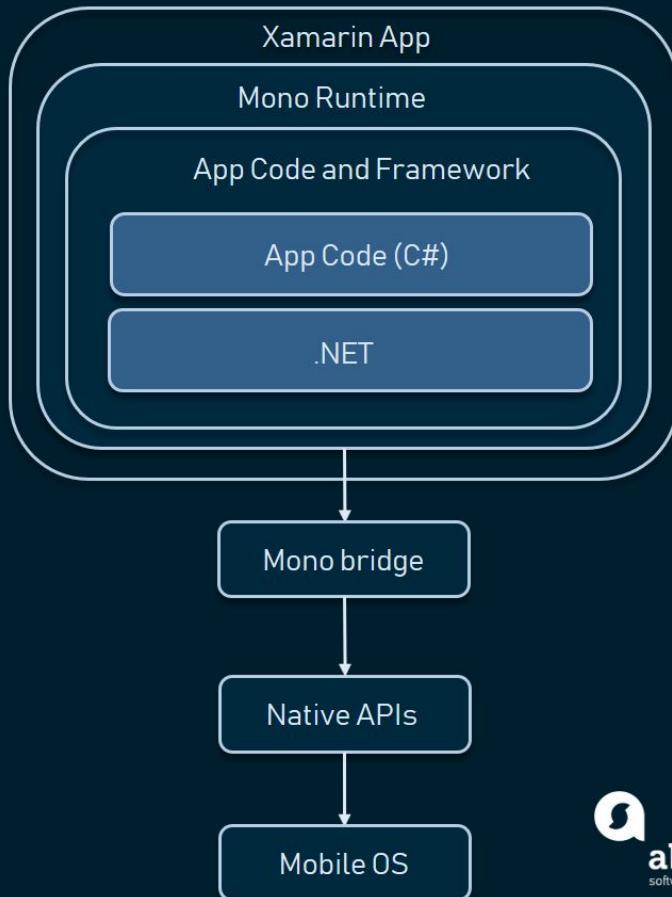
- **Xamarin classique** : 1 interface iOS / 1 interface Android,
- **Xamarin Forms** : interface unique (concept proche de celui de Flutter).

# SOLUTIONS CROSS PLATFORMS

Xamarin s'intègre à l'écosystème .Net de Microsoft :

- langage de programmation **C#**,
- **Mono** pour compiler le code C# en langage natif en mode "*Just In Time*" à destination d'Android et iOS,
- **XAML** en tant que langage de description de template, réalisés avec **Forms**.

# XAMARIN RENDERING

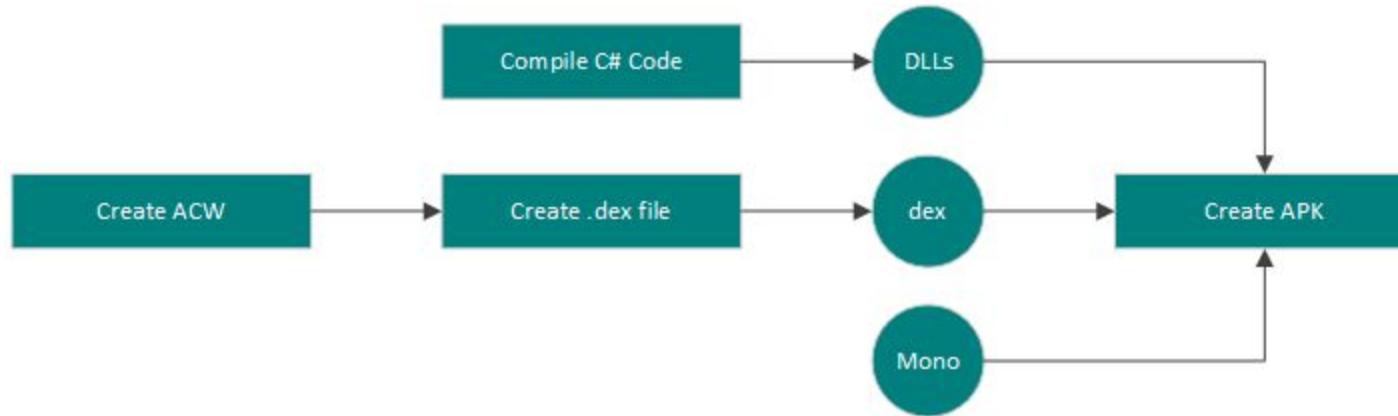


Xamarin vs  
React Native vs  
Ionic vs  
NativeScript:  
Cross-platform  
Mobile  
Frameworks  
Comparison

<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>

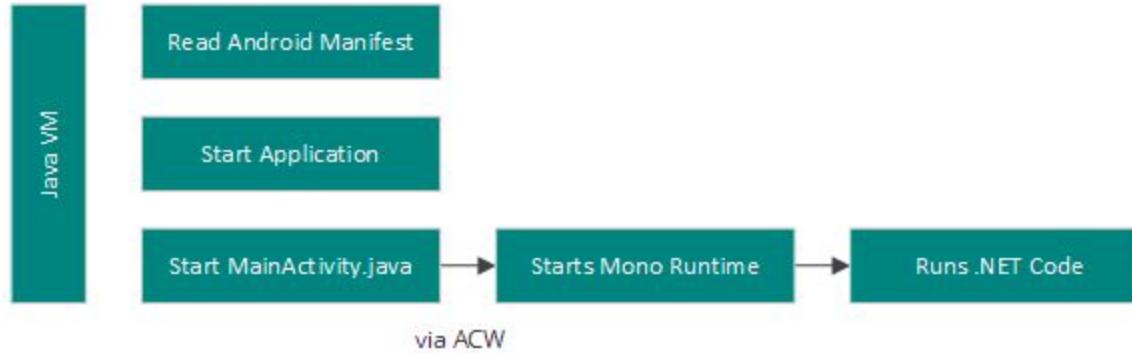


**altexsoft**  
software r&d engineering



## Processus de compilation d'une application mobile réalisée avec Xamarin

<https://www.xamarinhelp.com/xamarin-android-aot-works/>



## Processus de chargement d'une application mobile réalisée avec Xamarin

<https://www.xamarinhelp.com/xamarin-android-aot-works/>

# SOLUTIONS

## Xamarin remplacé par MAUI

Microsoft a récemment annoncé le remplacement de Xamarin par **MAUI** (Multi-Platform App UI).

L'objectif est d'uniformiser la plateforme .Net (Xamarin étant à l'origine un rachat de Microsoft).

**MAUI sera disponible avec la version 6 de .Net.**

## AUGUST MOBILE FEST #1

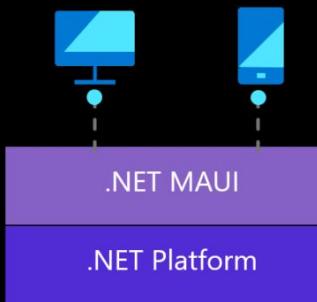
Flutter, Kotlin, MAUI, ...  
Xamarin est-il mort ?



### Xamarin est-il mort ?

<https://blog.cellenza.com/xamarin/flutter-kotlin-MAUI-xamarin-est-il-mort/>

# Introducing .NET Multi-platform App UI



Cross-platform, native UI

Single project, single codebase

Deploy to multiple devices, mobile & desktop

Evolution of Xamarin.Forms

Targeting .NET 6, previews end of year

**Build beautiful, native UI for any device**

[github.com/dotnet/maui](https://github.com/dotnet/maui)

## Développement mobile : Microsoft met à la retraite Xamarin

<https://www.lemondeinformatique.fr/actualites/lire-developpement-mobile-microsoft-met-a-la-retraite-xamarin-83681.html>



Kotlin



Multiplatform

Mobile



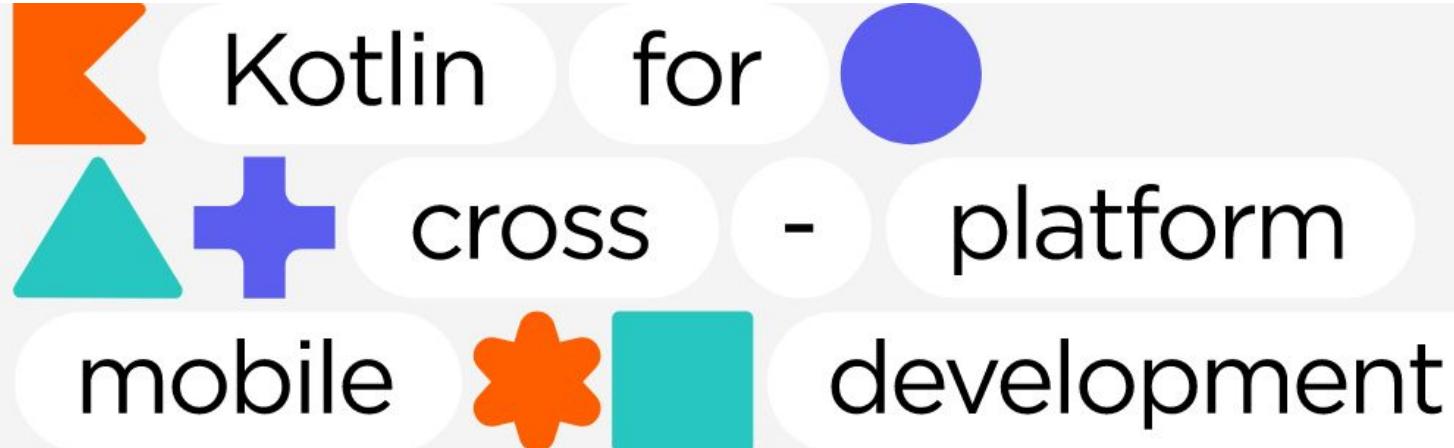
Kotlin Multiplatform Mobile

# SOLUTIONS

## Kotlin Mobile

Le langage de programmation **Kotlin** est surtout connu pour son intégration à Android.

Jetbrains, propose une solution cross platforms indépendante dont l'approche est comparable à React Native, Xamarin ou Flutter.



Write the business logic for your iOS and Android apps with less time and effort - using nothing but Kotlin

Get started

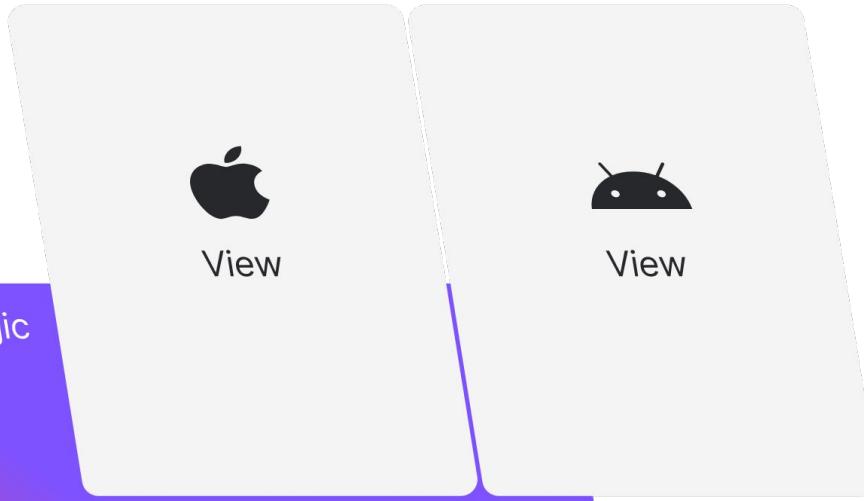
<https://www.jetbrains.com/lp/mobilecrossplatform/>

## SOLUTIONS CROSS PLATFORMS

Kotlin Mobile permet de produire des applications natives compilées en langage natif (Java / Swift) après compilation du code Kotlin par **LLVM** (*Low Level Virtual Machine*).

Tout comme pour les langages de bas niveau C et C++, la gestion de la mémoire doit être effectuée par le développeur.

## Native Code



## Shared Code

Business logic  
and core

iOS  
specific APIs

Android  
specific APIs

## Business logic and core

expect fun platformStuff()



### iOS specific APIs

```
actual fun platformStuff() {  
    AVFoundation  
    Core ML  
    LocalAuthentification  
    Accounts  
    CloudKit  
    and lots of community  
    libraries  
}
```



### Android specific APIs

```
actual fun platformStuff() {  
    CameraX  
    MLKit  
    Biometric  
    Room  
    Play Services  
    and Lots of community  
    libraries  
}
```

# Kotlin Multiplatform Mobile

# Application mobile interprétée à la compilation

# SOLUTIONS CROSS PLATFORMS

Flutter est **la plus récente des solutions mobiles cross platforms.**

La solution **open source** de **Google** a bénéficié d'une rapide adoption par les développeurs et les entreprises, qui lui a permis de se hisser **à la hauteur de React Native**.

# SOLUTIONS CROSS PLATFORMS

**Flutter s'est inspiré des concepts forts de React Native :**

- paradigme déclaratif,
- gestion d'état (widgets stateful / stateless),

# SOLUTIONS CROSS PLATFORMS

Flutter emploie **Dart**, un langage de programmation typé statiquement, comparable à JS ou Java.

**Flutter propose une expérience de développement particulièrement appréciée** : intégration à Android Studio et Visual Studio, Hot Reloading efficace, langage relativement intuitif, concept des widgets...

## Flutter



Google Ads



Alibaba

## React Native



Instagram



Netflix



Yandex Go



eBay



Amazon



Uber



Capital One



Groupon



Viber



Pinterest

# SOLUTIONS CROSS PLATFORMS

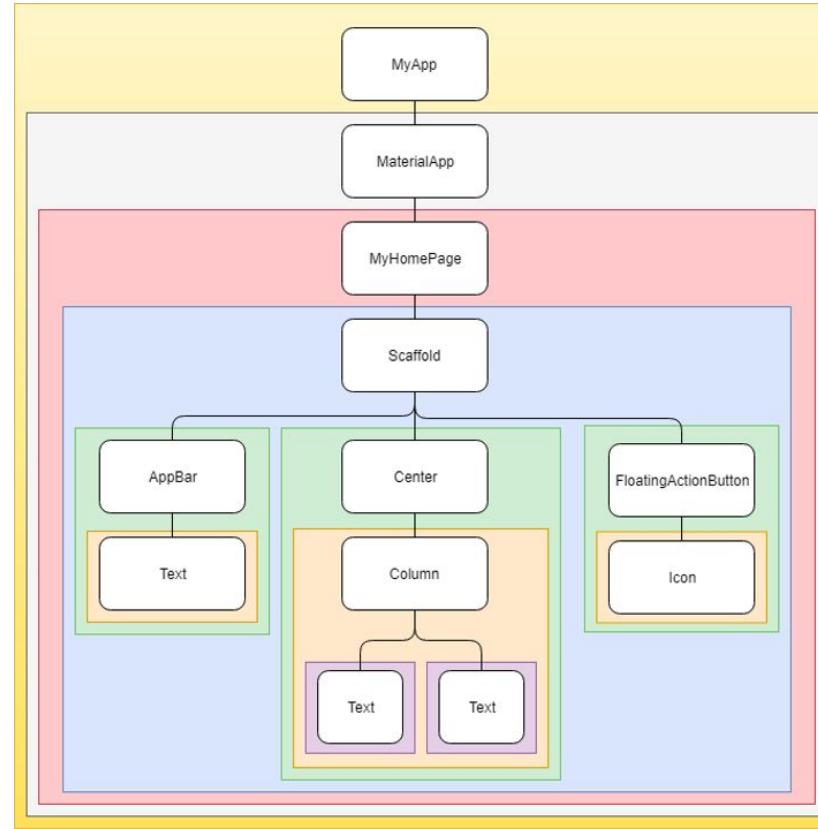
**Flutter se différencie des autres solutions mobiles cross platforms en prenant le parti de ne pas employer les composants UI natifs iOS et Android proposés par les deux SDK.**

En effet, une application Flutter *"dessine au pixel"* l'intégralité de l'interface.

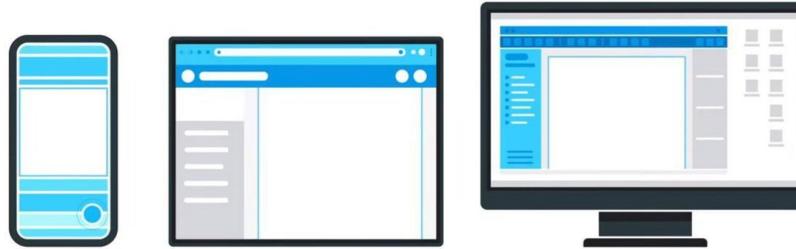
# SOLUTIONS CROSS PLATFORMS

Flutter propose une **riche collection de composants mobiles UI au style Material Design (Android) ou Cupertino (iOS)**.

Une interface réalisée avec Flutter s'affiche à l'**identique sur iOS et Android**.



**Dans une application Flutter tous les composants d'interface sont des widgets (éléments interactifs et de mise en page)**



**Flutter vise la réalisation d'application universelle (mobile, web et desktop) basée sur le même code source.**

# SOLUTIONS CROSS PLATFORMS

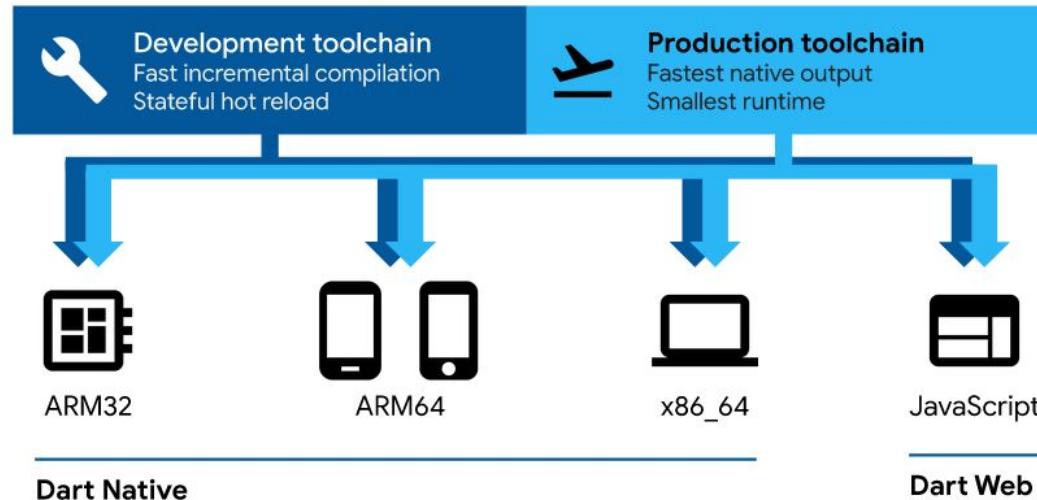
Flutter emploie à la fois **les méthodes de compilation JIT et AOT**:

- **JIT (Dart VM)** en phase de développement, permettant de réaliser du Hot Reloading,
- **AOT (dart2native)** en phase de compilation (build),

## Dart: The platforms

Dart's compiler technology lets you run code in different ways:

- **Native platform:** For apps targeting mobile and desktop devices, Dart includes both a Dart VM with just-in-time (JIT) compilation and an ahead-of-time (AOT) compiler for producing machine code.
- **Web platform:** For apps targeting the web, Dart includes both a development time compiler (dartdevc) and a production time compiler (dart2js). Both compilers translate Dart into JavaScript.



**Compilations JIT et AOT avec Dart**  
<https://dart.dev/overview#platform>

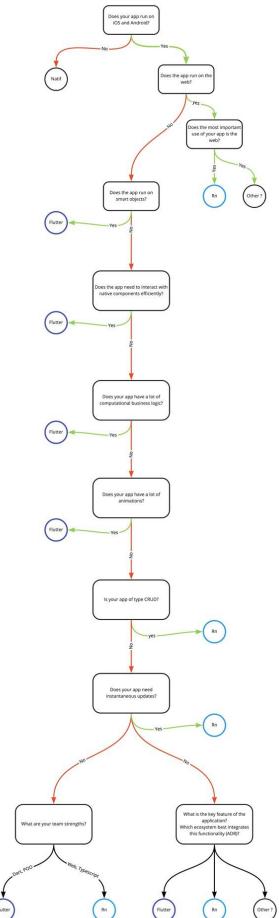
# Flutter vs Native vs React-Native: Examining Performance

Ihor Demedyuk & Nazar Tsybulskyi

## Flutter vs Native vs React-Native: Examining Performance

<https://inveritasoft.com/blog/flutter-vs-native-vs-react-native-examining-performance>

You want to build a mobile app

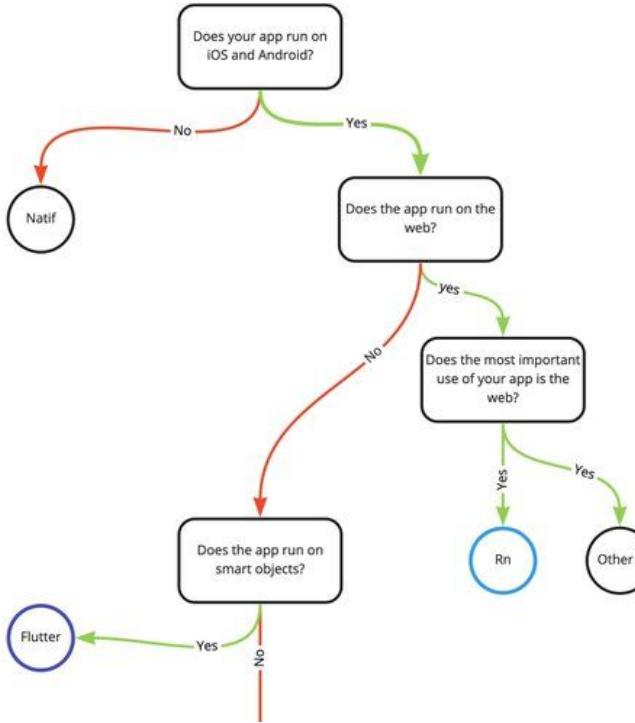


# React Native or Flutter: the decision tree

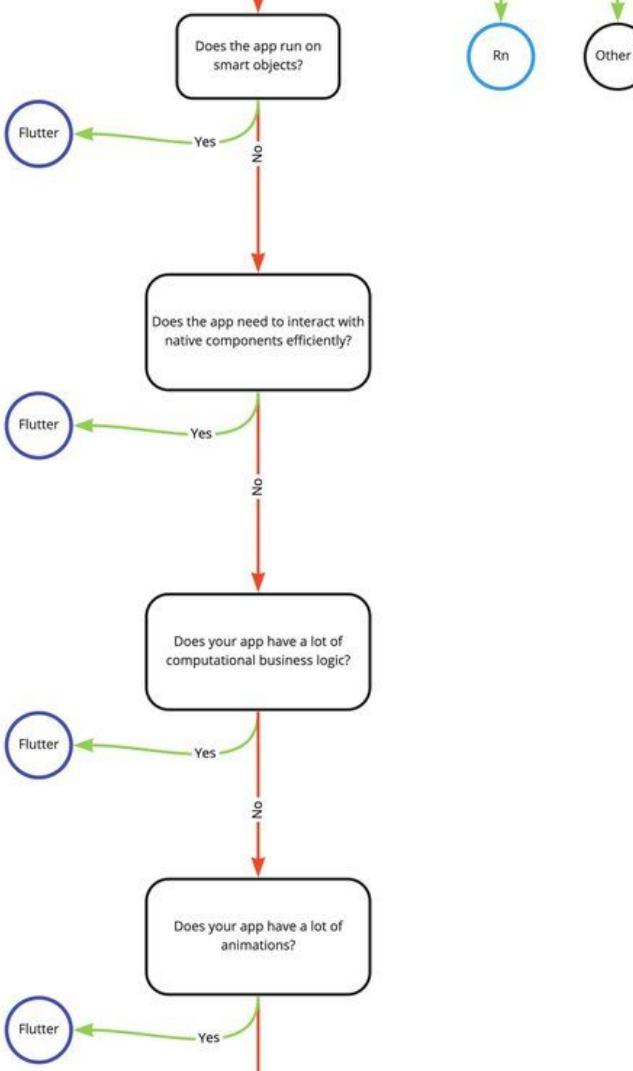
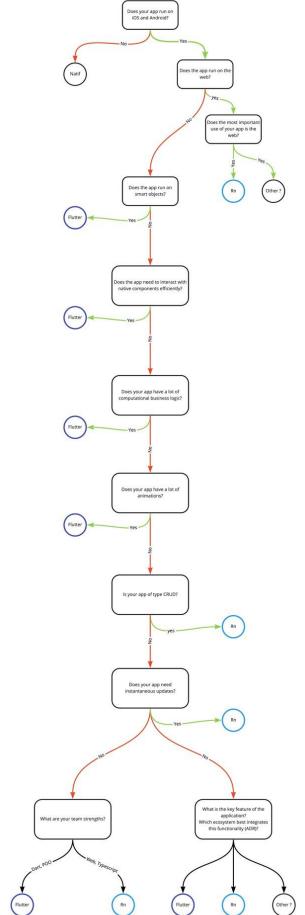
<https://blog.bam.tech/developer-news/react-native-or-flutter-the-decision-tree>

1/4

You want to build a mobile app

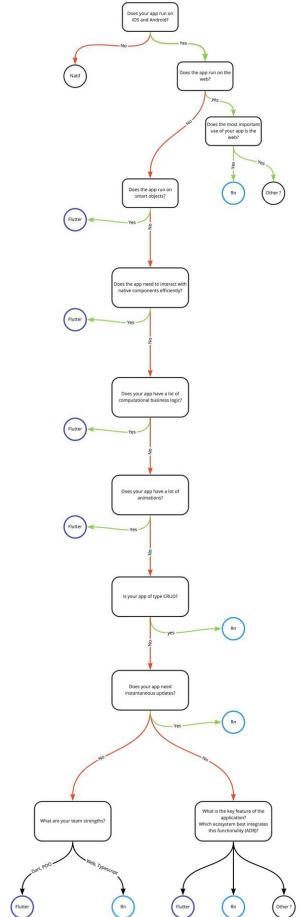


You want to build a mobile app



2/4

You want to build a mobile app



Does your app have a lot of computational business logic?

Flutter

Does your app have a lot of animations?

Flutter

Is your app of type CRUD?

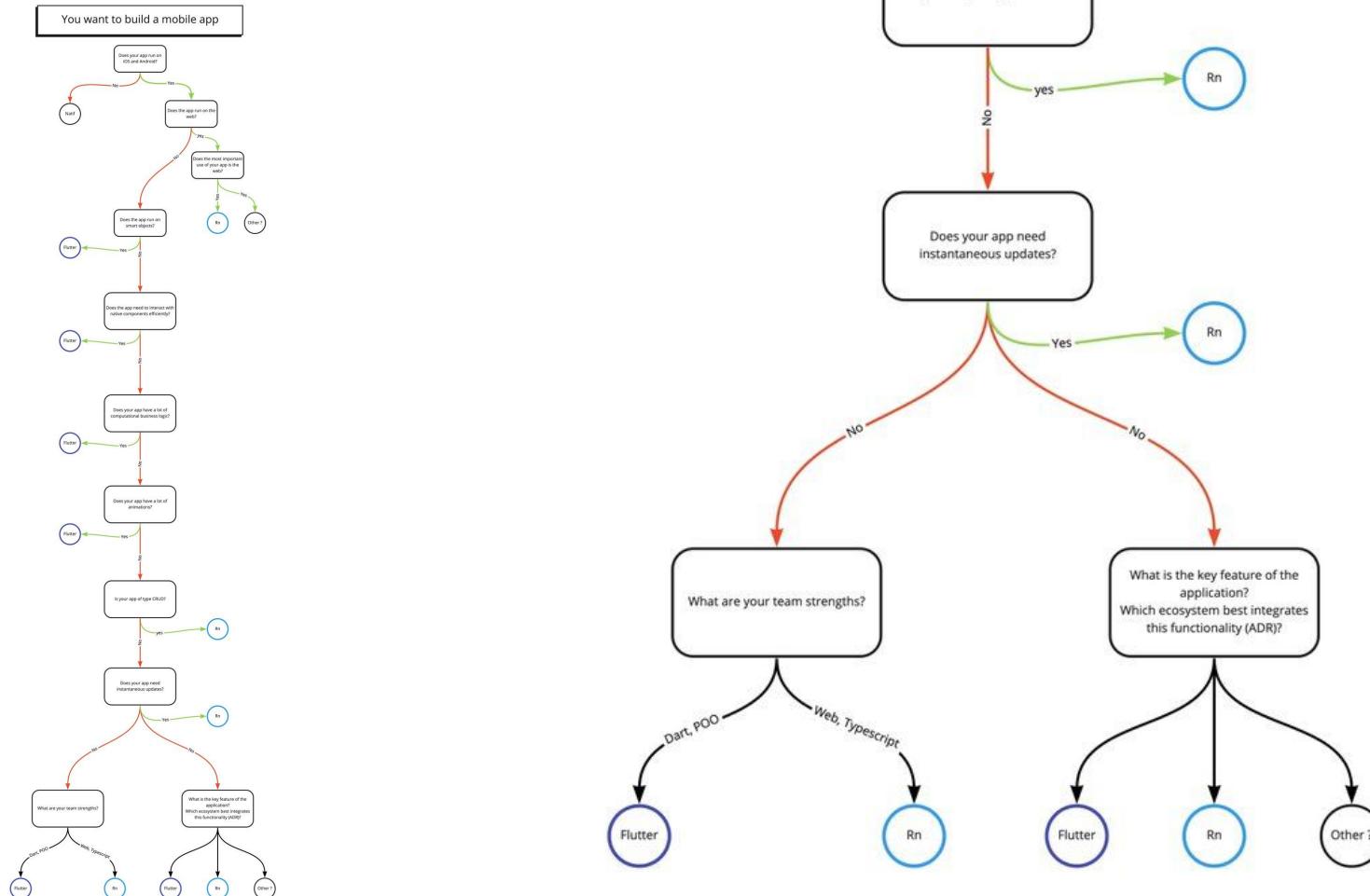
Rn

Does your app need instantaneous updates?

Rn

3/4

140



# SOLUTIONS CROSS PLATFORMS

**Application mobile**

**No-Code / Low Code**

# SOLUTIONS CROSS PLATFORMS

De solutions mobiles cross-platforms dites "***Low Code***" ou "***No Code***" sont apparues :

- Kony (<https://www.kony.com/quantum/>),
- PowerApps (<https://powerapps.microsoft.com/fr-fr/>),
- Zoho (<https://www.zoho.com/fr/creator/>),
- Xojo (<https://www.xojo.com/>),
- Build Box (<https://www.buildbox.com/>)...



## Cross-platform

Code once then deploy on macOS, Windows, Linux, the web, iOS and Raspberry Pi.



## Rich Framework

With modern support for graphics, database servers and internet protocols, build anything you can imagine.



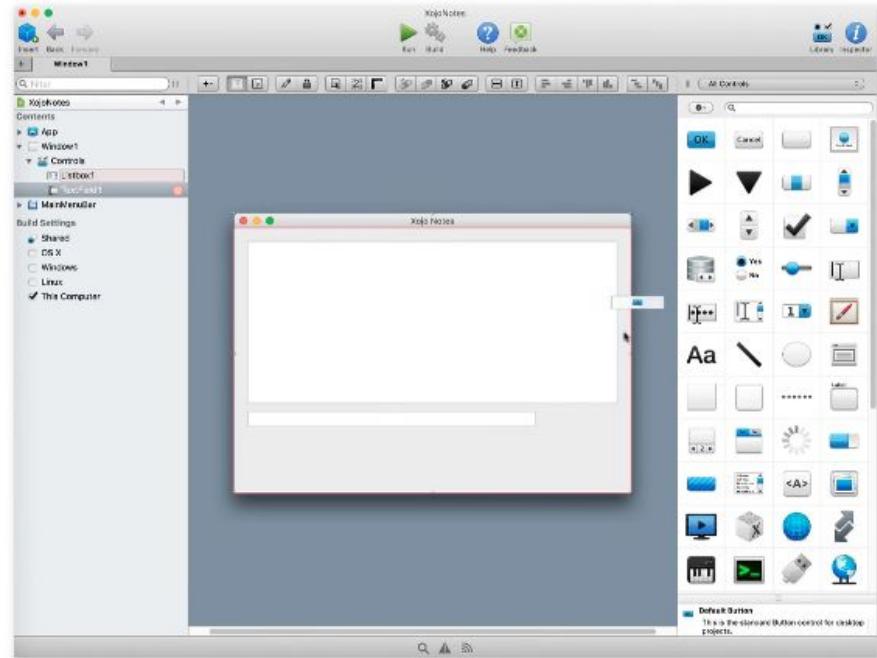
## Drag & drop UI

Create your native user interface quickly and intuitively with drag and drop.



## Abstraction

Xojo abstracts you from the platform details allowing you to focus your time and energy on what makes your app unique.



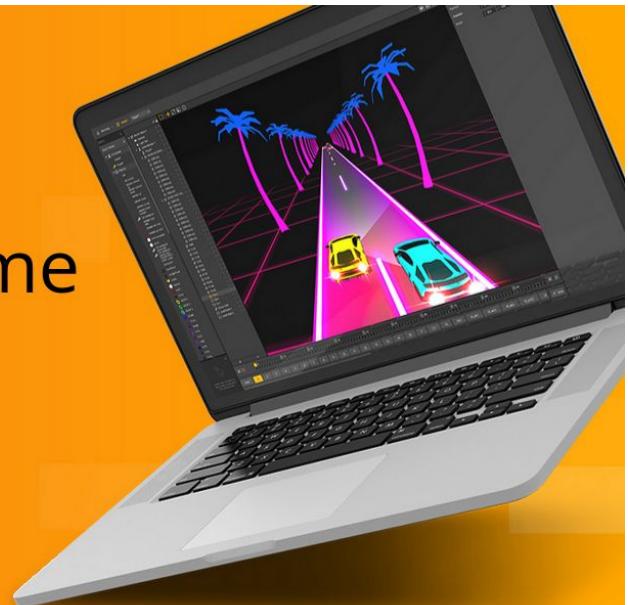
**Xojo, solution mobile cross-platforms "*low code*"**  
<https://www.xojo.com/>



# No-Code mobile game development.

The power to design, build, and publish 3D & 2D mobile games without coding.

DOWNLOAD FREE



Buildbox solution "*no code*"  
pour le développement de jeux mobiles cross platforms

<https://www.buildbox.com/>