



Développement Multiplateforme

DR. GHIZLANE HNINI

HNINIG@GMAIL.COM

Plan du cours

- ▶ Introduction : Développement multiplateformes
- ▶ Flutter : Architecture, fonctionnement et composants
- ▶ Configuration de l'environnement Flutter
- ▶ Le langage Dart
- ▶ Les composants d'une interface graphique de Flutter
 - ▶ Les différents widgets de Flutter
 - ▶ La notion d'état
 - ▶ Les widgets de mise en forme
- ▶ La navigation et Drawer
- ▶ Flutter et Firebase: Connexion, authentification et Opérations CRUD

Plan du cours

- ▶ **Introduction : Développement multiplateformes**
- ▶ Flutter : Architecture, fonctionnement et composants
- ▶ Configuration de l'environnement Flutter
- ▶ Le langage Dart
- ▶ Les composants d'une interface graphique de Flutter
 - ▶ Les différents widgets de Flutter
 - ▶ La notion d'état
 - ▶ Les widgets de mise en forme
- ▶ La navigation et Drawer
- ▶ Flutter et Firebase: Connexion, authentification et Opérations CRUD

Développement mobile

- Le développement d'application mobile est le processus de créer des applications spécifiquement conçus pour être utilisés sur des appareils mobiles tels que smartphones et tablettes



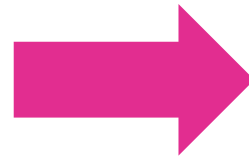
Développement mobile

- Ces applications permettent aux utilisateurs d'effectuer une variété de tâches, de divertissements, de communication, de productivité et bien plus encore, en utilisant leurs appareils mobiles.



Développement mobile

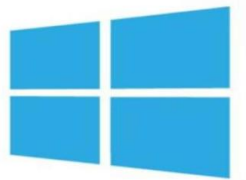
- Ces applications peuvent être développées pour différentes plateformes, notamment Android, iOS(Apple), Windows



android



iOS



Windows

Plateformes

- ▶ Se réfère généralement à un environnement ou à une infrastructure qui fournit un ensemble de ressources, de services et de fonctionnalités nécessaires pour développer, exécuter et gérer des logiciels, des applications ou des systèmes.
- ▶ Il sont conçues pour faciliter le développement et le déploiement de logiciels en fournissant un cadre cohérent et des outils appropriés pour les développeurs.

Plateformes

- **Plateforme de développement** : Un environnement de développement intégré (IDE) comme Visual Studio, Eclipse ou IntelliJ IDEA peut être considéré comme une plateforme de développement. Il fournit des outils pour la création, la modification, le débogage et le test de logiciels.



Plateformes

- **Plateforme de système d'exploitation** : Windows, macOS et Linux sont des exemples de plateformes de système d'exploitation. Ils fournissent une infrastructure de base pour exécuter des applications et des services.



Plateformes

- **Plateforme de base de données** : Les systèmes de gestion de bases de données (SGBD) tels que MySQL, PostgreSQL et Oracle Database sont des plateformes qui permettent de stocker, d'organiser et de gérer des données de manière efficace



Plateformes

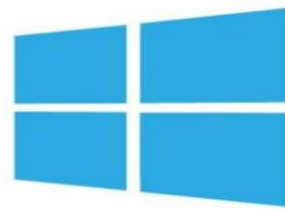
- **Plateforme de développement mobile** : iOS (pour les appareils Apple) et Android (pour les appareils Android) sont des plateformes de développement mobile qui fournissent des outils et des environnements pour la création d'applications mobiles.



android



iOS



Windows

Android

- ▶ Conçu initialement par une entreprise américaine nommée Android Inc, racheté en 2005 par Google et lancé en 2007
- ▶ Android est un système d'exploitation mobile et une plateforme open source conçu pour des applications mobiles
- ▶ Android est basé sur un Kernel Linux qui peut être installé sur n'importe quel appareil.
- ▶ Les applications Android s'exécutent sur une machine virtuelle nommée **DalvikVM**



Android

- ▶ Vous désirez réaliser une application, l'ensemble des éléments que vous devez installer.

Android

- ▶ Vous désirez réaliser une application, l'ensemble des éléments que vous devez installer.
 - ▶ **JDK:** Java Development Kit (pour la compilation Java)

Android

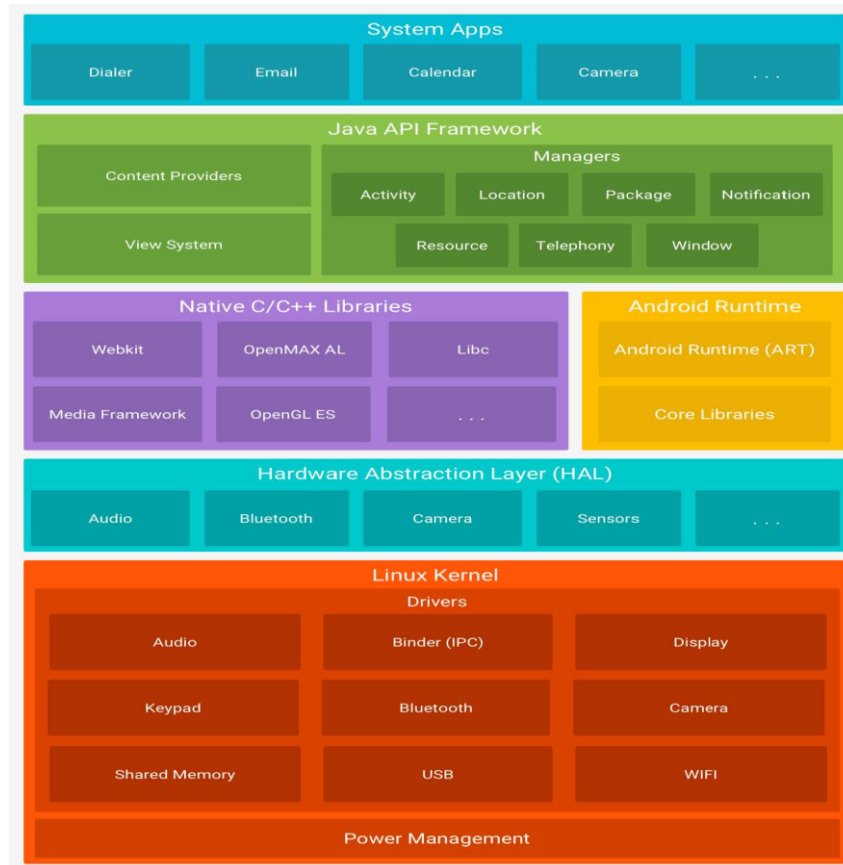
- ▶ Vous désirez réaliser une application, l'ensemble des éléments que vous devez installer.
 - ▶ **JDK:** Java Development Kit (pour la compilation Java)
 - ▶ **Android Studio:** IDE pour le développement des applications Android

Android

- ▶ Vous désirez réaliser une application, l'ensemble des éléments que vous devez installer.
 - ▶ **JDK:** Java Development Kit (pour la compilation Java)
 - ▶ **Android Studio:** IDE pour le développement des applications Android
 - ▶ **Android SDK** (Software Development Kit): Pour la compilation Android et l'émulateur.

En utilise Google Play pour distribuer nos applications Android.

Architecture d'Android



Des applications

- dont le navigateur web, une gestion de contact, calendrier ...

Le framework Android

- pour la gestion de fenêtre, téléphonie, de gestion de contenu et des fonctionnalités natives du dispositif mobile

Une machine virtuelle Java adaptée: Dalvik Virtual Machine

Des bibliothèques graphiques, multimédia

Une couche d'abstraction matériel

- couche intermédiaire entre le système d'exploitation et le matériel informatique.
- offre des fonctions standardisées de manipulation du matériel informatique, en cachant les détails complexes du matériel

(Plus simplement, elle permet aux programmes de fonctionner sans se soucier des détails compliqués du matériel)

Un noyau Linux

- Drivers pour la gestion du matériel:
(Gestion de Mémoire, Gestion de caméra, Gestion de Bluetooth)

iOS

- ▶ iOS est le système d'exploitation d'Apple, lancé en 2007
- ▶ iOS est une extension de Mac OS X.
- ▶ iOS est propriétaire, impossible de modifier son code source.
- ▶ Les applications iOS s'exécutent directement sur le système.



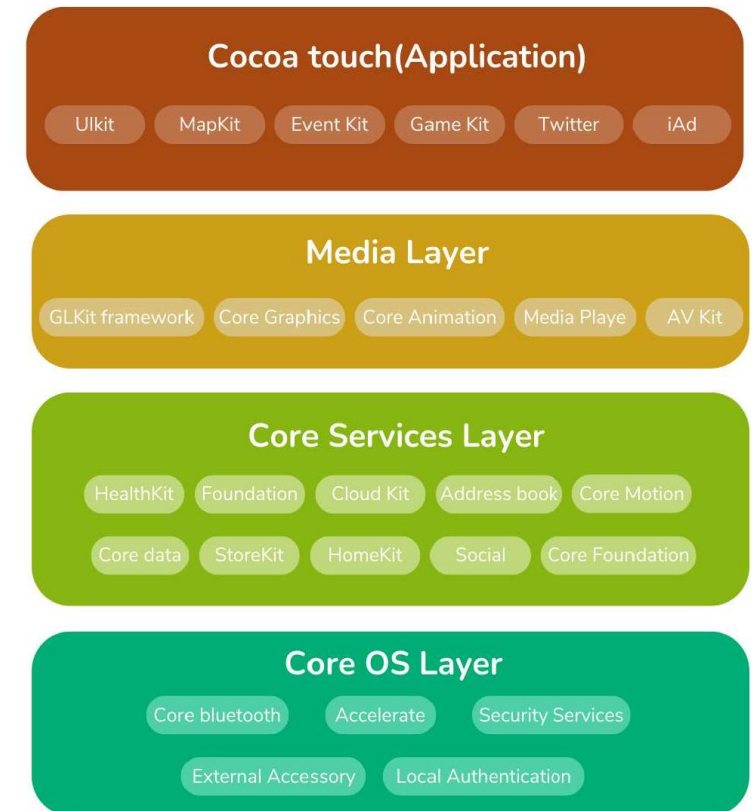
IOS

- ▶ Vous désirez réaliser une application, l'ensemble des éléments que vous devez avoir.
 - ▶ **Objective C (Swift)**: Langage de programmation
 - ▶ **XCode**: IDE pour le développement des applications iOS
 - ▶ **IOS SDK**

En utilise Apple App Store pour distribuer nos applications iOS.

Architecture iOS

- ▶ Core OS : gère par exemple le système sécurité
- ▶ Cores Services: peuvent supporter, par exemple, l'accès vers une base de données SQLite.
- ▶ Media: permet de gérer les médias audio ou vidéo.
- ▶ Cocoa touch: elle fournit tous les frameworks utiles à la création d'une application iOS.



Windows 10 mobile

- ▶ Windows 10 mobile a été conçu par Microsoft, lancé en 2015
- ▶ L'architecture de windows 10 mobile, est très proche de l'architecture de Windows 10.
- ▶ Les applications windows s'exécutent sur une machine virtuelle.



Windows 10 mobile

- ▶ Langage: C#
- ▶ Windows Phone SDK
- ▶ IDE: Visual Studio
- ▶ Microsoft MarketPlace.

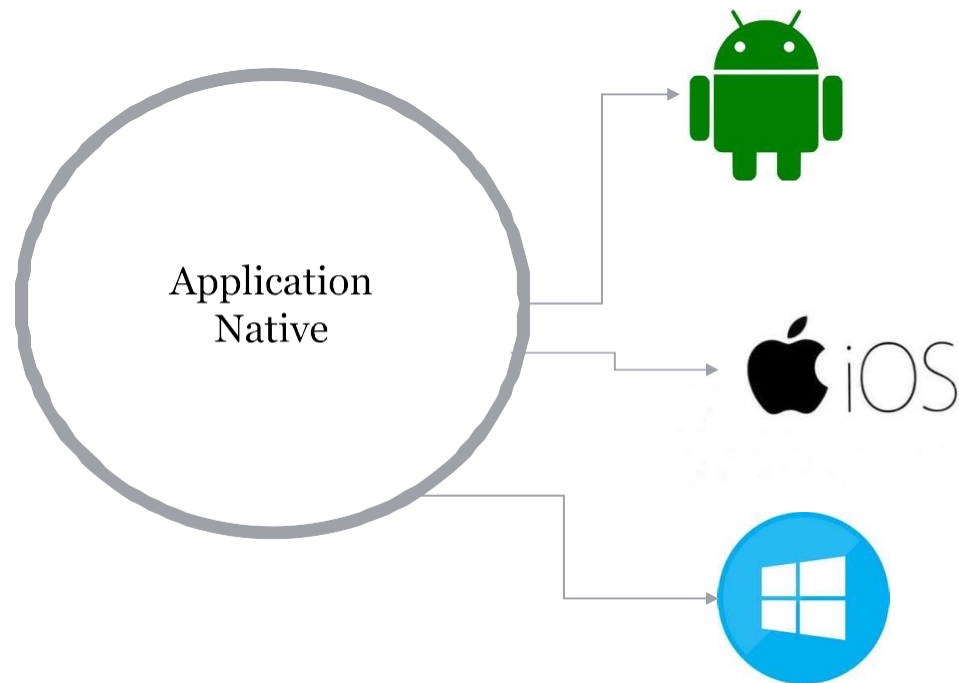


- :
- Langages : C#
- Windows Phone SDK
- IDE : Visual Studio
- Microsoft MarketPlace

Méthodes de développement mobile

Développement Mobile Native

- Les applications natives sont construites en utilisant les langages natives
- Construite pour une plateforme spécifique



Avantages et Inconvénients (Native)

Avantages

- Meilleure façon d'implémenter une application Mobile
 - Rapidité
 - Performance
 - Distribuée dans l'appStore correspondant

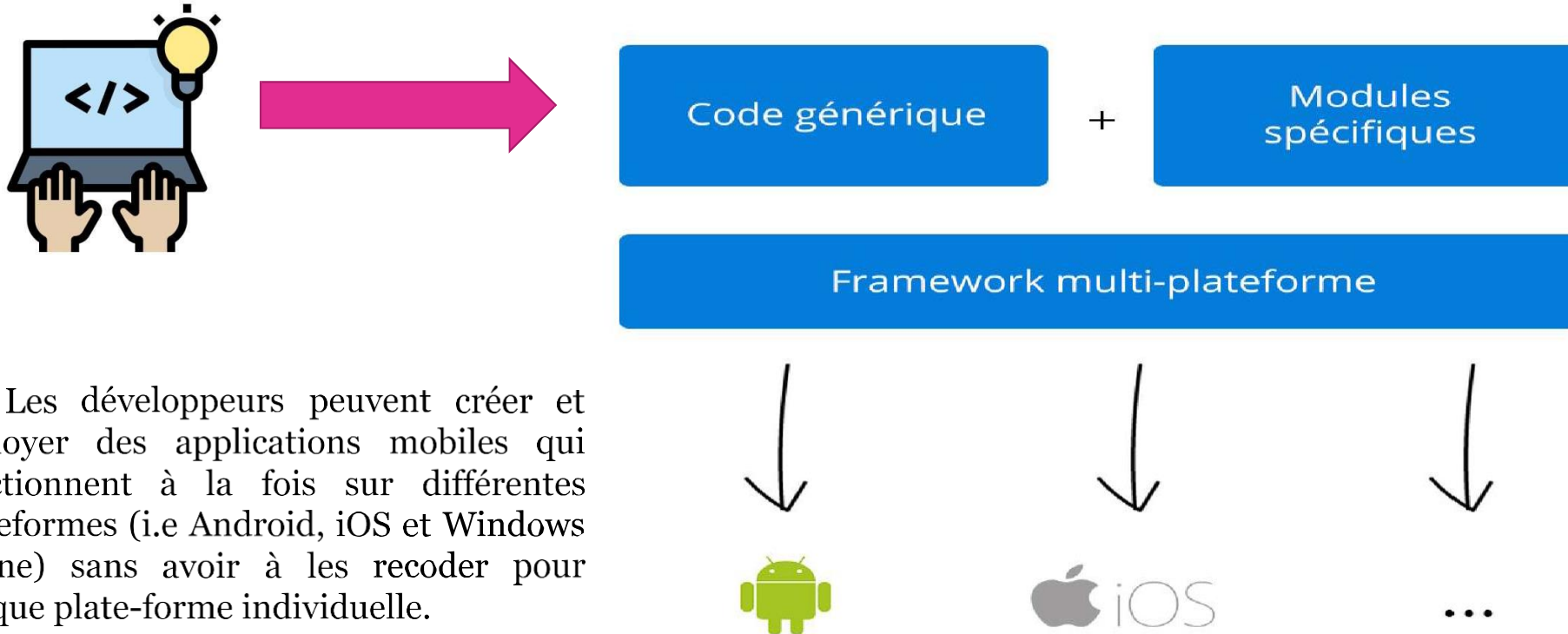
Inconvénients

Si les entreprises décident de développer des applications pour différentes plateformes, il y a beaucoup d'effort à faire

- L'application doit s'implémenter dans :
 - Java ou Kotlin pour les applications Android
 - Objective-C ou Swift pour les applications IOS
 - C# pour les applications Windows
- Coûteux de maintenir du code pour les différentes plateformes

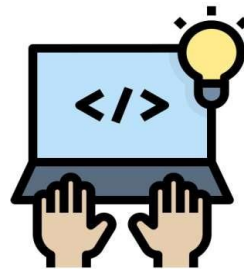
Programmation multiplateformes

La programmation mobile multiplateforme est une approche de développement d'applications qui permet de créer une seule base de code source pour déployer des applications sur plusieurs plates-formes mobiles, telles qu'Android, iOS, etc.



=> Les développeurs peuvent créer et déployer des applications mobiles qui fonctionnent à la fois sur différentes plateformes (i.e Android, iOS et Windows Phone) sans avoir à les recoder pour chaque plate-forme individuelle.

Programmation multiplateformes



Développement Mobile Multiplateformes

Développement Hybride ou basé sur le Web

Une application mobile hybride est une application dont **le code source permet de générer deux applications** : l'une pour Android et l'autre pour iOS. Elle s'appuie pour cela sur **des technologies « Web »** telles que le Javascript, HTML ou encore CSS.

Développement Native Cross-Platform

Les applications cross-platform produisent **un même code source qui, compilé, produit deux applications natives**

Programmation multiplateformes

- Quelques frameworks de développement multiplateforme

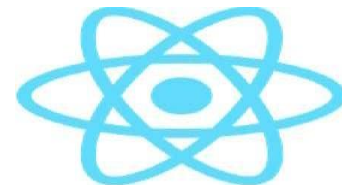
Framework pour le développement hybride ou basé sur le Web



Xamarin



Flutter



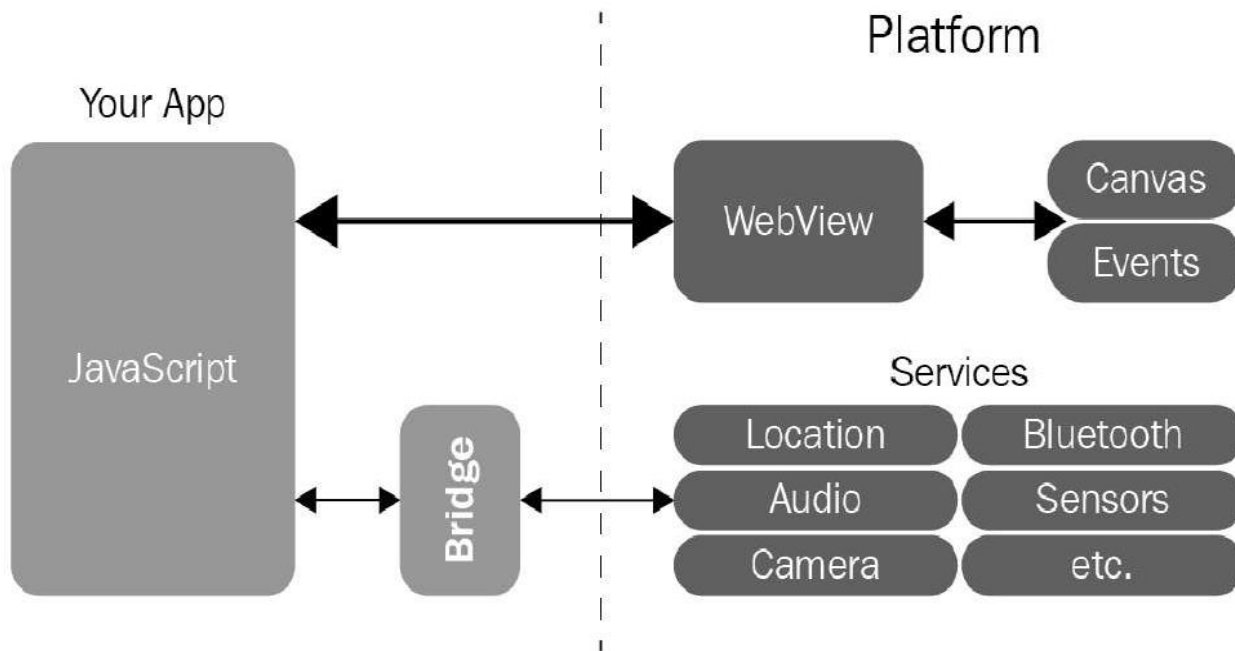
React Native

Framework pour le développement cross-platforme

Approche basée sur le web (hybride)

- ❑ Combinaison des applications web et des applications natives
- ❑ Utilisation des technologies web (HTML, CSS et Javascript) encapsulées dans un composant natif **WebView** (le seul composant natif dans l'appli).
 - L'exécution se fait au sein d'un **conteneur**/ webview, en utilisant un framework correspondant
- ❑ Cette dénomination hybride vient du fait que ces applications ne sont ni purement natives, ni purement basées sur les langages HTML, CSS et JavaScript ... c'est un subtil mélange des deux.

Architecture d'une plateforme hybride



- Le navigateur rend votre code et Ionic a besoin d'un pont pour accéder aux fonctionnalités du système d'exploitation comme une caméra
- Vous devez compter sur ce pont « Bridge » pour accéder aux fonctionnalités natives.
- Par exemple le plugin Cordova se compose de deux parties, vous permettant ainsi d'accéder à la plateforme native correspondante:
 - Une partie JS
 - Et l'autre partie native ; si c'est android c'est du Java ou Kotlin, si c'est IOS, c'est l'Objective-C par exemple

Approche hybride

Avantages :

- Facile à construire (HTML/CSS/JS)
- Un seul code pour toutes les plateformes
- Beaucoup moins coûteuses qu'une application native, en terme de temps et de budget
- Pas besoin de navigateur, ces applications s'exécutent dans le webview
- Peuvent souvent avoir accès aux périphériques de l'appareil en utilisant une API

Inconvénients :

- Plus lentes que les applications natives
- Plus coûteuses que les applications web
- Un peu moins interactive que les applications natives

Développement Cross-Platform

Framework pour le développement cross-plateforme

Un seul code



Machine Code



React Native

- React Native est un framework de développement d'applications mobiles open-source créé par Facebook en 2015
- Il permet aux développeurs de créer des applications mobiles multiplateformes en utilisant principalement React JS; une bibliothèque JavaScript également développée par facebook .
- Les développeurs Web peuvent utiliser React Native pour écrire des applications Android et iOS qui agissent et se présentent comme des applications natives.
- React est déjà l'un des frameworks d'applications multiplateformes les plus populaires

Apps populaire

- Applications créées avec le framework d'application multiplateforme React Native
 - ☐ Instagram
 - ☐ Bloomberg
 - ☐ Pinterest
 - ☐ Skype
 - ☐ Tesla

Flutter

- ❑ Flutter est un framework open source de Google permettant de créer de applications multiplateformes compilées nativement à partir d'une seule base de code.
- ❑ La première version stable de Flutter a été publiée en décembre 2018
- ❑ Flutter utilise un langage de programmation compilé, appelé dart.
- ❑ Flutter fournit des vues réactives sans avoir besoin d'un bridge JavaScript pour communiquer avec la plateforme
- ❑ Les principaux blocs de construction de flutter sont les widgets

Apps populaire

Quelques applications utilisant le framework Flutter

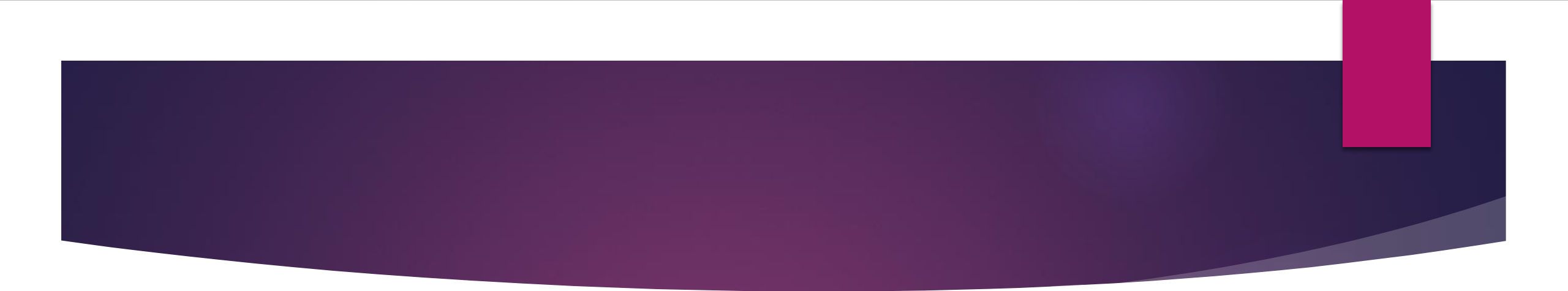
- ❑ Alibaba
- ❑ Google
- ❑ Google Ads
- ❑ Tencent

Avantage (programmation multiplateforme)

- Réutilisabilité du code source
- Gain du temps
- Gestion efficace des ressources
 - Avoir une base de code unique vous aide à gérer efficacement vos ressources.
 - Vos équipes de développement Android et iOS peuvent apprendre à écrire et à utiliser du code partagé.
- Possibilité d'atteindre un public plus large
- Délai de mise sur le marché et de personnalisation plus rapides
 - Puisque vous n'avez pas besoin de créer différentes applications pour différentes plateformes, vous pouvez développer et lancer votre produit beaucoup plus rapidement.
 - plus facile pour les programmeurs d'apporter de petites modifications à des parties spécifiques de votre base de code
 - être plus réactif aux commentaires des utilisateurs.

Challenge (programmation multiplateforme)

- Le développement multiplateforme se débat toujours avec des problèmes liés à la performance
- Les chefs de projet peuvent craindre que leur objectif d'optimiser le processus de développement ait un impact négatif sur l'expérience utilisateur d'une application
- Différentes plates-formes ont souvent des conventions d'interface utilisateur différentes, que les applications multiplateformes ne prennent pas toujours en compte. Par exemple, les applications développées pour macOS et GNOME sont censées placer le bouton le plus important sur le côté droit d'une fenêtre ou d'une boîte de dialogue, alors que Microsoft Windows et KDE ont la convention inverse.
- Le test d'applications multiplateformes peut être considérablement plus compliqué, car différentes plates-formes peuvent présenter des comportements légèrement différents ou des bogues subtils.
- Les environnements d'exécution multiplateformes peuvent souffrir de failles de sécurité multiplateformes, créant un environnement fertile pour les logiciels malveillants multiplateformes.

- 
- Quels sont les challenges de la programmation Native ?
 - Quel est la différence entre l'approche cross-plateforme et l'approche hybride basée sur le webview ?