

# Développement d'applications mobiles - théorie

## Chapitre 1 : Introduction

---

Daniel Schreurs

10 août 2022

Haute École de Province de Liège

## 1. Introduction

Comment communiquer ?

## 2. Contexte historique

Retour en 2007

Shopping time !

La situation

L'essor des solutions web

Des solutions hybrides

## 3. Flutter

Flutter, c'est quoi ?

Quelques forces

Quelques faiblesses

## Quelques Exemples

### 4. Architecture

#### Flutter

Framework Flutter

Plug-ins

Engine

Engine

Découpe de la couche Framework

# Introduction

---

# Introduction : Comment communiquer ?

- Toutes les ressources : sur Moodle ;
- Une question relative au cours : Forum du cours (sur Moodle) ;
- Une question personnelle : [daniel.schreurs@hepl.be](mailto:daniel.schreurs@hepl.be).

## **Contexte historique**

---

# Contexte historique : Retour en 2007



Tout commence en juin 2007 avec l'iPhone.

## Contexte historique : Shopping time!



Android Market

L'App Store a ouvert 07/2008 - Android Market 10/2008.



## Contexte historique : La situation

- Développer pour les 2 plateformes ;
- Si vous avez les moyens de développer une application, vous êtes l'enfant cool du quartier ;
- Risques et des coûts élevés ;
- Les coûts explosent car il faut maintenir plusieurs bases de code ;
- Deux grandes tendances :
  - Natif ;
  - Sites Web responsives.

## Contexte historique : L'essor des solutions web

- Moins cher ;
- Technologies plus matures ;
- Merci HTML5 et aux WebViews ;
- Par exemples : [Cordova](#) ;
- Ces applications avaient souvent du mal à reproduire l'aspect et la convivialité des applications natives.

## Contexte historique : Des solutions hybrides

- En 2015 Facebook dévoile React Native ;
- Une solution hybride :
  - Même logique métier que celle du web ( JavaScript) ;
  - Utiliser un WebView, mais un système de rendu natif.
- Le succès est énorme ;
- Mais des améliorations sont possibles (voir nécessaire).

# Flutter

---

# Flutter : Flutter, c'est quoi ?

- A software development toolkit, de Google ;
- Permet de construire des applications multiplateformes ;
- Des paquets, plug-ins et beaucoup de widgets ;
- Flutter n'est pas un langage ;
- Flutter utilise [Dart](#) comme langage de programmation.

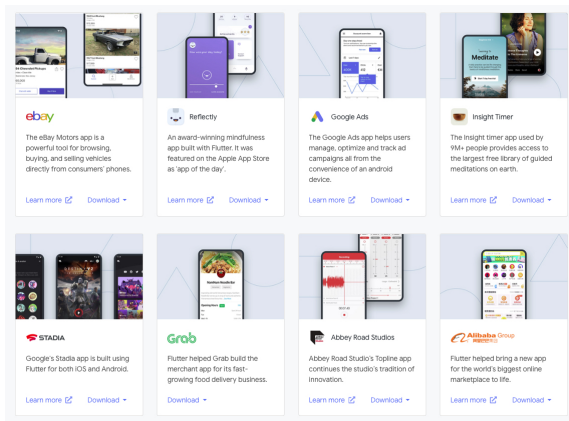
## Flutter : Quelques forces

- Flutter permet de compiler pour le Web, Android et IOS ;
- Flutter est open source ;
- Flutter utilise un langage récent [Dart](#) ;
- Flutter permet le rechargement à chaud ([hot reload](#)) ;
- Flutter [supporte nativement](#) le [Material Design](#) de Google ;
- Flutter permet de programmer des animations et transitions personnelles ou déjà existantes ;
- Flutter permet le databinding ;
- Les concepts de flutter sont proches de [SwiftUI](#) et [Jetpack Compose](#).
- Flutter permet de faire des applications accessibles.

## Flutter : Quelques faiblesses

- Ce n'est pas du développement natif ;
- N'est pas adapté pour des jeux et/ou applications audios ;
- N'est pas adapté pour des besoins très spécifiques de l'environnement Apple ;
- Ne supporte pas watchOS, tvOS etc. ;
- C'est un pari sur l'avenir, la technologie est très récente.

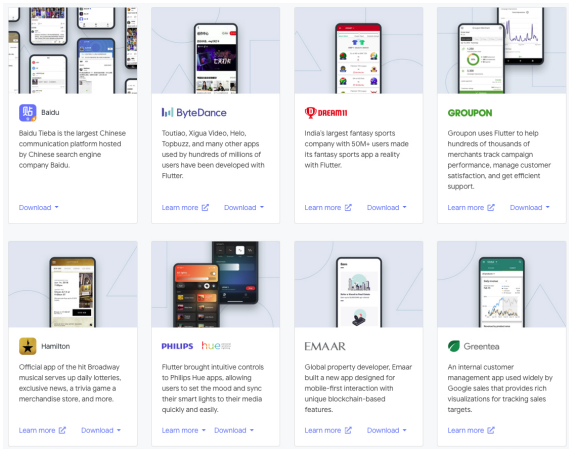
# Flutter : Quelques Exemples



Quelques exemples parmi d'autres : Apps take flight with Flutter



# Flutter : Quelques Exemples



Quelques exemples parmi d'autres : Apps take flight with Flutter

# Architecture

---

# Architecture : Flutter

## Framework Flutter

*Dart, layout, librairies,  
widgets, packages, etc.*

## Plugins

*Haut niveau*

## Engine

*Librairies C, C++ pour supporter les fonctions  
de bas niveau*

## Embedder

*Spécifique à la plateforme - compilation*

## Architecture Flutter

- Le **Framework Flutter** est écrit en Dart ;
- Il contient des bibliothèques de haut niveau avec notamment :
  - Les thèmes ;
  - Les widgets ;
  - La mise en page ;
  - Les animations.

- Les **Plugins** isolent des fonctionnalités de haut niveau :
  - La sérialisation JSON ;
  - La géolocalisation ;
  - L'accès aux caméras ;
  - Les paiements in-app, etc.

- La couche **Engine** contient les bibliothèques C++. Elle gère :
  - Les graphiques;
  - La mise en page du texte;
  - L'accessibilité;
  - L'architecture des plugins et le moteur d'exécution Dart.

- L'**Embedder** est différent pour chaque plateforme cible. Il gère :
  - L'*empaquetage* du code comme une application autonome.

## Remarque

Chacune des couches est encore une fois décomposée en plusieurs sous-couches.



# Architecture : Découpe de la couche Framework



Architecture du framework Flutter