ECOLE DES MINES DE L'INDUSTRIE
ET DE LA GÉOLOGIE

Développement Mobile

2025-2026

Chapitre 2: Introduction à Flutter et Dart

2.1 Qu'est-ce que Flutter ?

Flutter est un **framework open-source** développé par Google, lancé officiellement en 2017. Il permet de créer des applications multiplateformes avec **un seul code source**, déployable sur :

- Android
- iOS
- Web
- Desktop (Windows, macOS, Linux)

Contrairement aux approches hybrides classiques, Flutter se distingue par :

- Des performances proches du **natif** grâce à la compilation directe en code machine.
- Une interface utilisateur moderne basée sur des **widgets** hautement personnalisables.
- Un cycle de développement rapide avec la fonctionnalité **Hot Reload**, qui permet de voir immédiatement les modifications du code sans redémarrer l'application.

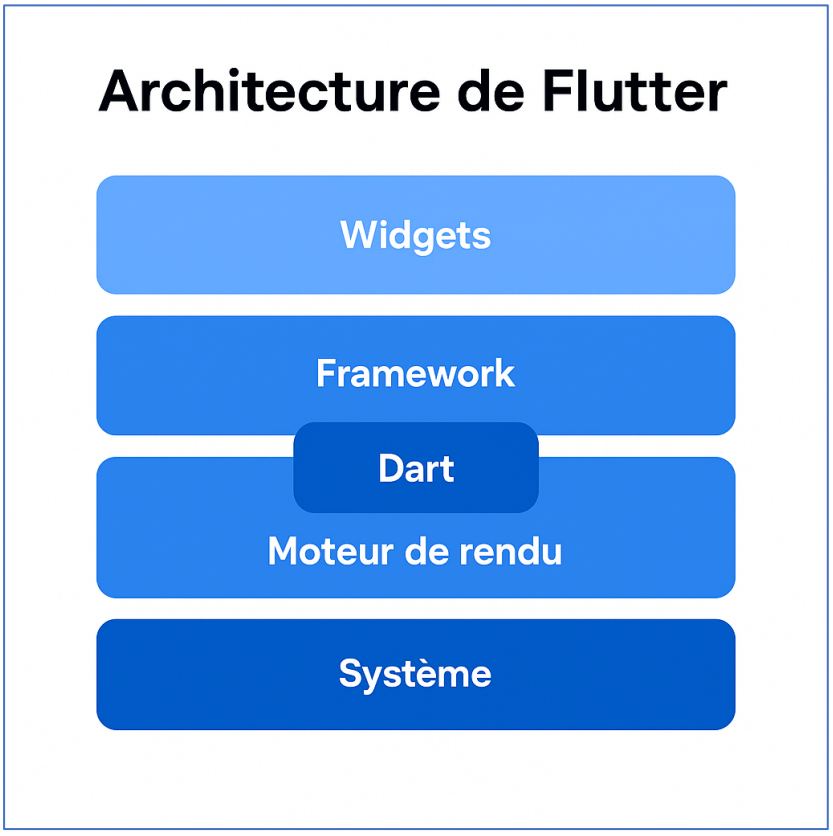
2.2 Pourquoi utiliser Flutter ?

- **Gain de temps** : un seul projet pour plusieurs plateformes.
- **Économie de coûts** : pas besoin de deux équipes (Android et iOS).
- **Richesse visuelle** : large bibliothèque de widgets respectant les standards Material Design (Google) et Cupertino (Apple).
- **Grande communauté** : support actif, documentation riche, nombreux packages disponibles sur pub.dev.

2.3 Architecture de Flutter

Flutter repose sur trois éléments essentiels :

1. **Widgets** : tout dans Flutter est un widget (textes, images, boutons, layouts).
2. **Moteur de rendu (Skia)** : assure un affichage graphique fluide et performant.
3. **Framework Dart** : contient les bibliothèques de base (Material, Cupertino, animations, etc.).



2.4 Comparaison avec d'autres frameworks

Critère	Flutter	React Native	Ionic
Langage	Dart	JavaScript	HTML/CSS/JS
Performance	Native (compilation AOT) compile le code Dart en **code machine natif**	Proche du natif (bridge JavaScript)	Moins performant
UI	Widgets personnalisables	Composants natifs	WebView
Productivité	Hot Reload	Hot Reload	Rapide mais limité
Plateformes	Android, iOS, Web, Desktop	Android, iOS, Web	Android, iOS, Web

2.5 Atelier pratique : “Hello Flutter”

Objectif : créer et exécuter la première application Flutter.

1. Créer un projet Flutter :

- flutter create hello_flutter

Cd hello_fluttert

2. Exécuter le projet :

- flutter run

3. Modifier le fichier lib/main.dart :

```
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Hello Flutter")),
        body: const Center(
          child: Text("Bienvenue dans Flutter @ Airtel Niger !"),
        ),
      ),
    );
  }
```

2.6 Qu'est-ce que Dart?

Dart est un **langage orienté objet** très **puissant** et **polyvalent**, qui a été développé chez Google au départ pour combler les défauts du **JavaScript** et avait pour objectif de le remplacer en tant que langage phare du web.

Date de première version : 10 octobre 2011



- C'est un langauge de programmation open-source, basé sur des classes et éventuellement typé.
- Aujourd'hui avec le langage Dart et Flutter, nous pouvons créer des **applications mobiles**, des **sites web** et des **logiciels de bureau**.

Caractéristiques

- Syntaxe proche de Java, JavaScript, C#.
- Compilation **AOT** (Ahead of Time) → performance native.
- Compilation **JIT** (Just in Time) → Hot Reload.
- Gestion simple de l'asynchronisme avec **async/await**.
- Typage statique, sûr et moderne.

Pourquoi Dart pour Flutter ?

- Très rapide.
- Facile à apprendre.
- Optimisé pour les widgets & animations.
- Compilation multiplateforme.

2.7 Environnement GNU/Linux, Windows et macOS

Compatibilité Flutter / Dart

Flutter fonctionne sur :

- **Windows 10/11** (64 bits)
- **macOS (Intel & Apple Silicon)**
- **Linux distributions (Ubuntu, Debian, Fedora, etc.)**

Pré-requis systèmes

- Windows: Git, PowerShell, Android Studio ou command-line tools
- macOS: Xcode (pour iOS), Homebrew (recommandé)
- Linux: curl, zip, unzip, Git, dépendances GTK

- **Installation d'un IDE Dart et écriture d'un programme Dart**
- **Installation de Visual Studio Code (VS Code)**
- **Création d'un projet Dart avec Visual Studio Code**
- **Création d'un projet Dart avec DartPad**