

# Giới thiệu học phần

Bùi Võ Quốc Bảo | Khoa CNTT

Học kì 2, 2022 – 2023

# CT484 – Phát triển ứng dụng di động

Các bạn nên hoàn thành các học phần sau để đạt kết quả tốt nhất:

- Cấu trúc dữ liệu
- Lập trình hướng đối tượng
- Cơ sở dữ liệu
- Nhập môn lập trình Web

# CT484 – Phát triển ứng dụng di động

Khi cần trợ giúp, đặt câu hỏi?

- Classroom (Các bạn đã/sẽ nhận được lời mời tham gia)
- Email: [bvqbao@cit.ctu.edu.vn](mailto:bvqbao@cit.ctu.edu.vn)

# Nội dung

Chương 1: Ngôn ngữ lập trình Dart

Chương 2: Giới thiệu về Flutter

Chương 3: Các widget cơ bản

Chương 4: Điều hướng

Chương 5: Quản lý trạng thái

Chương 6: Lưu trữ dữ liệu

Chương 7: Thiết kế giao diện đáp ứng

Chương 8: Tùy biến và phát hành

# Nội dung thực hành/bài tập

**Bài tập 1:** Ứng dụng MyShop – Thiết kế giao diện

**Bài tập 2:** Ứng dụng MyShop – Thiết kế giao diện + Điều hướng

**Bài tập 3:** Ứng dụng MyShop – Quản lý trạng thái

**Bài tập 4:** Ứng dụng MyShop – Lưu trữ dữ liệu (Firebase)

# Thời biểu (dự kiến)

- Tuần 01: Giới thiệu, Dart
- Tuần 02: Dart + Giới thiệu Flutter
- Tuần 03: Các widget cơ bản
- Tuần 04-05: Nghỉ Tết Nguyên Đán
- Tuần 06: Các widget cơ bản
- **Tuần 07: TH1**
- Tuần 08: Điều hướng + Quản lý trạng thái
- **Tuần 09: TH2**
- Tuần 10: Quản lý trạng thái (tt)
- Tuần 11: Quản lý trạng thái (tt), Lưu trữ dữ liệu
- **Tuần 12-13: TH3+TH4**
- Tuần 14: Giao diện đáp ứng, tùy biến và phát hành
- **Tuần 15: TH5 (Project)**
- Tuần 16: Thực hiện Project
- **Tuần 17: TH6 (Báo cáo Project)**
- **Tuần 18: Kiểm tra trắc nghiệm**

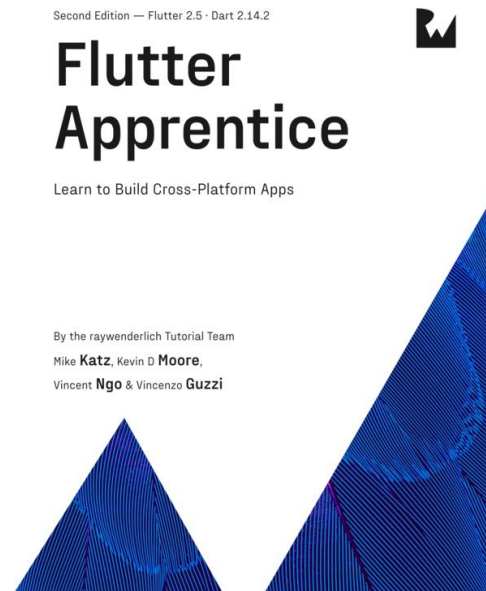
# Tài liệu tham khảo

- Ngôn ngữ lập trình Dart
  - **Dart Apprentice** by Jonathan Sande & Matt Galloway



# Tài liệu tham khảo

- Nền tảng Flutter
  - **Flutter Apprentice** by Vincenzo Guzzi, Kevin D Moore, Vincent Ngo and Michael Katz





# Phương pháp đánh giá

- Hình thức đánh giá
  - Trắc nghiệm, thực hành, làm dự án
- Nội dung đánh giá
  - Thực hành/Bài tập có hướng dẫn (4 bài): 30%
    - ✓ Không thực hành = Không đánh giá cuối kỳ = F
    - ✓ Nộp trễ bài thực hành: -50% điểm bài thực hành bị trễ/tuần
  - Đánh giá cuối kỳ: 70%, gồm 2 phần:
    - ✓ Lý thuyết: trắc nghiệm 30%
    - ✓ Báo cáo project: 40%

# Project

- Xây dựng một ứng dụng di động với Flutter
  - Thời gian xây dựng: 3~4 tuần chính để hoàn thiện (sinh viên bắt đầu phát triển ý tưởng từ lúc bắt đầu học kỳ)
  - Sinh viên tự đề xuất dự án, một vài gợi ý:
    - Ứng dụng review sách, quán cà phê, phim, ...
    - Ứng dụng mua sắm/bán hàng, ...

# Project

- Xây dựng một ứng dụng di động với Flutter
  - Yêu cầu về chức năng:  $\geq 3$  trang giao diện
    - Dùng các widget cơ bản của Flutter để tạo bố cục, xây dựng các trang giao diện, hỗ trợ tương tác người dùng
    - Có hiển thị dữ liệu trong ListView và/hoặc GridView
    - Có điều hướng giữa các trang
    - Có quản lý trạng thái chia sẻ
    - Có lưu trữ dữ liệu (có thể là cục bộ hay từ xa), có cập nhật dữ liệu (thêm/sửa/ xóa – tùy vào ứng dụng/chức năng cài đặt, không cần phải đầy đủ hết các tác vụ)

# Project

- Xây dựng một ứng dụng di động với Flutter
  - Một số lưu ý khác:
    - **KHÔNG** sử dụng dự án niên luận đã làm trước đó hoặc trong học kỳ
    - Sinh viên được phép tham khảo code tuy nhiên **KHÔNG** được lấy y nguyên dự án từ bài thực hành, hướng dẫn trên youtube, trên blog, ... Giảng viên đánh giá cách sinh viên sử dụng code tham khảo như thế nào trong dự án

# Project

- Xây dựng một ứng dụng di động với Flutter
  - 1-2 sinh viên/nhóm
  - Cách thực hiện:
    - Sinh viên tự phát triển dự án, có thể trao đổi với giảng viên
    - **Mã nguồn được quản lý với git và GitHub, phải có lịch sử commit**
    - Nộp slide báo cáo theo mẫu trên classroom
    - Sinh viên chạy demo sản phẩm đạt được và trả lời câu hỏi

# Project

- Xây dựng một ứng dụng di động với Flutter
  - Các tiêu chí đánh giá:
    - Đảm bảo các yêu cầu chức năng/cài đặt
    - Mức độ hoàn chỉnh của các chức năng được cài đặt
    - Sự hiểu biết của sinh viên về mã nguồn dự án
      - ✓ Thông qua file báo cáo
      - ✓ Thông qua vấn đáp
    - Ý tưởng, giao diện đẹp, có tính thẩm mỹ

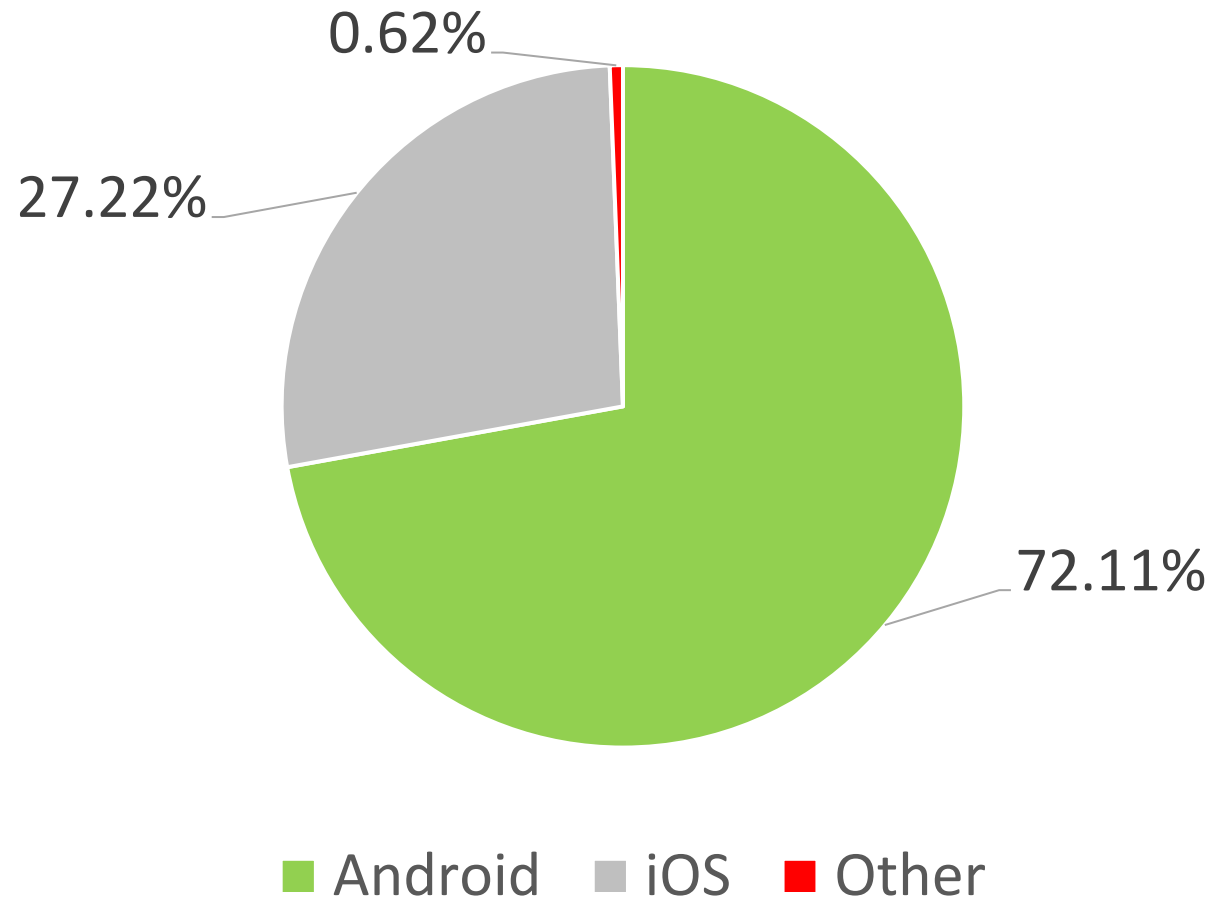
# Git và GitHub

- Tất cả mã nguồn của bài thực hành và dự án đều được quản lý bởi git và đưa lên GitHub
  - Sinh viên cần có / tạo tài khoản trên [GitHub](#)
- Giảng viên cung cấp tài liệu hướng dẫn sử dụng Git và GitHub



# Thị phần hệ điều hành di động

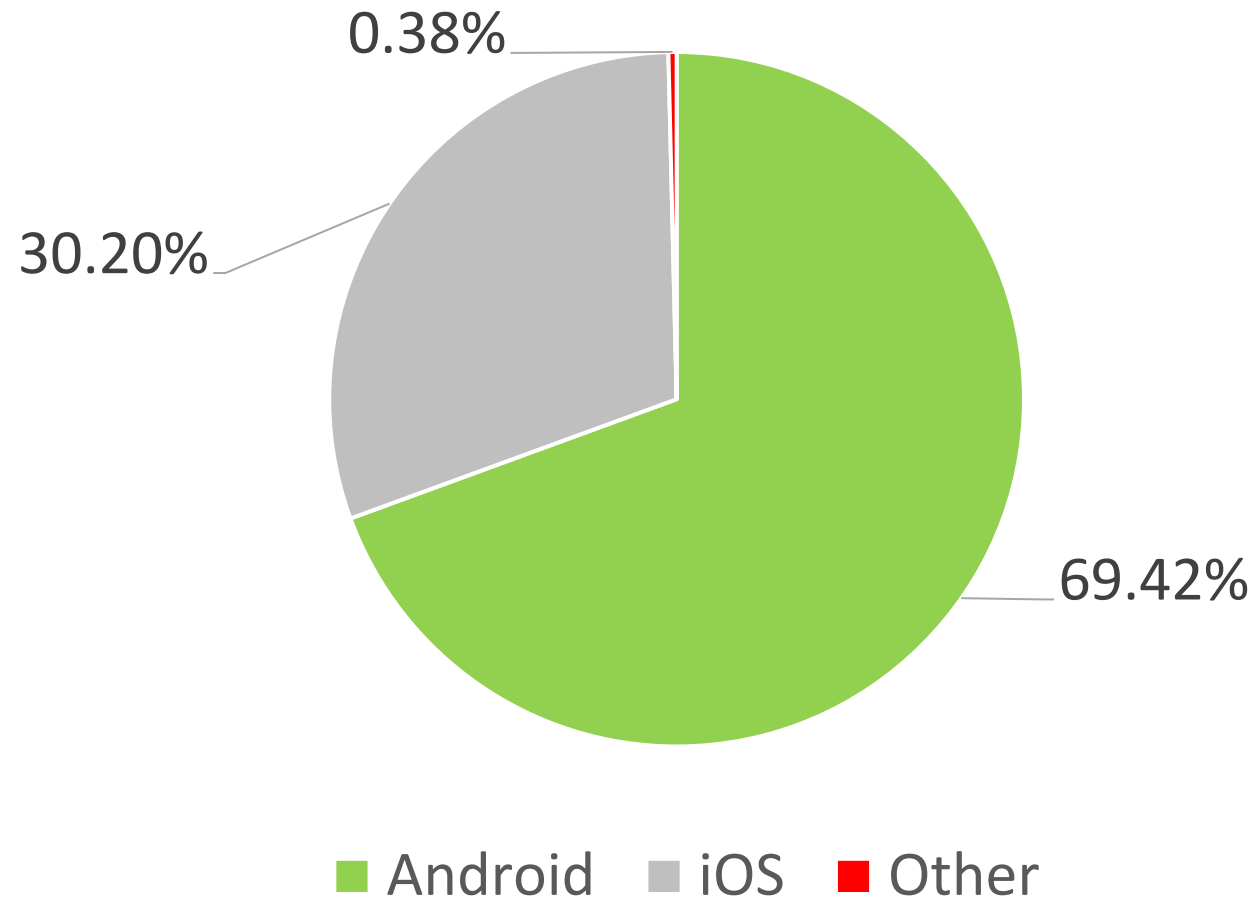
Mobile Operating System Market Share Worldwide - June 2022





# Thị phần hệ điều hành di động

Mobile Operating System Market Share in Vietnam - June 2022



# Phát triển ứng dụng di động

- Hai tùy chọn phát triển ứng dụng di động
  - Xây dựng ứng dụng gốc / nguyên bản hoàn toàn (fully native) / chính thức (official) của một nền tảng di động hoặc
  - Xây dựng ứng dụng đa nền tảng (cross-platform)
    - *Ít nhất* phần mã nguồn xử lý logic nghiệp vụ (không gồm phần GUI) có thể được sử dụng chung cho các nền tảng khác nhau

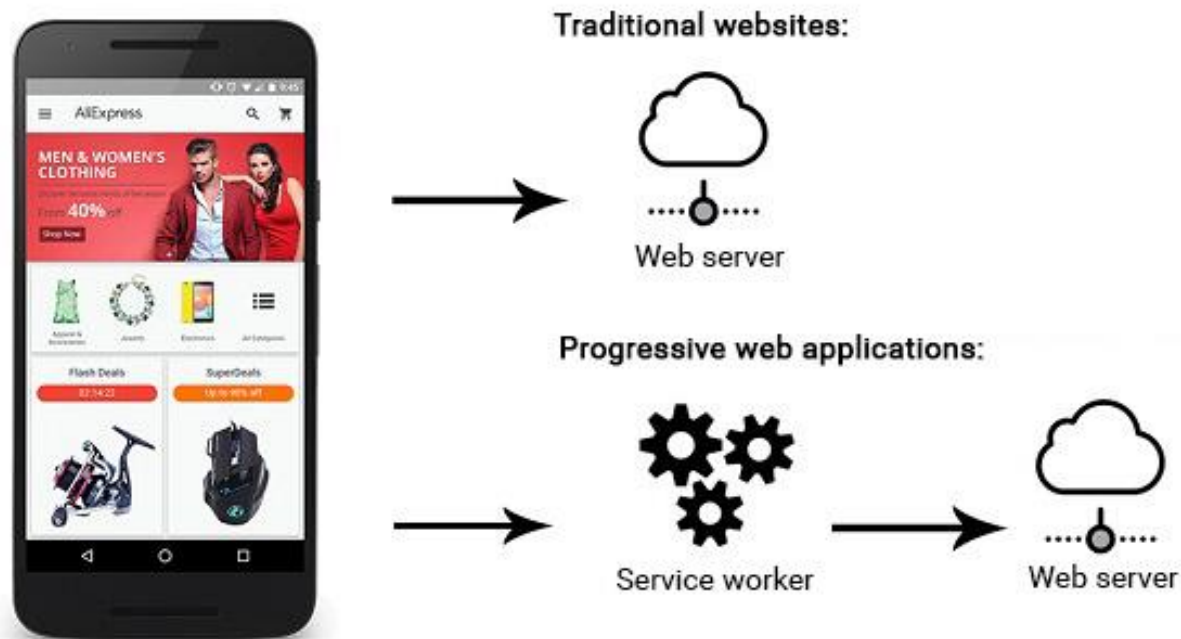
# Phát triển ứng dụng di động

- Phát triển ứng dụng di động gốc (native) cho một nền tảng
  - **Android NDK** (Kotlin/Java) cho Android, **Xcode** (Objective-C/Swift) cho iOS
  - Nhà phát triển điều khiển tối đa và thường cho hiệu năng tốt nhất



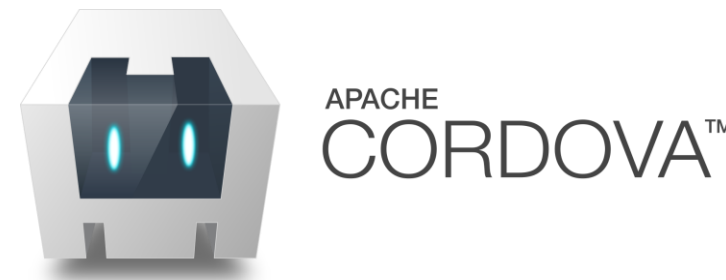
# Phát triển ứng dụng di động

- Phát triển ứng dụng di động đa nền tảng (cross-platform)
  - Ứng dụng web (HTML, CSS, JavaScript) triển khai trên thiết bị di động dạng *Progressive Web App (PWA)* trong môi trường trình duyệt



# Phát triển ứng dụng di động

- Phát triển ứng dụng di động đa nền tảng (cross-platform)
  - Ứng dụng web (HTML, CSS, JavaScript) triển khai trên thiết bị di động dùng *WebView* trong các môi trường thực thi như **Capacitor, Cordova**



# Phát triển ứng dụng di động

- Phát triển ứng dụng di động đa nền tảng (cross-platform)
  - Ứng dụng di động phát triển từ các framework hỗ trợ biên dịch ra mã lệnh gốc (native code) và sử dụng hệ thống *các widget riêng* như **Flutter (Dart)**



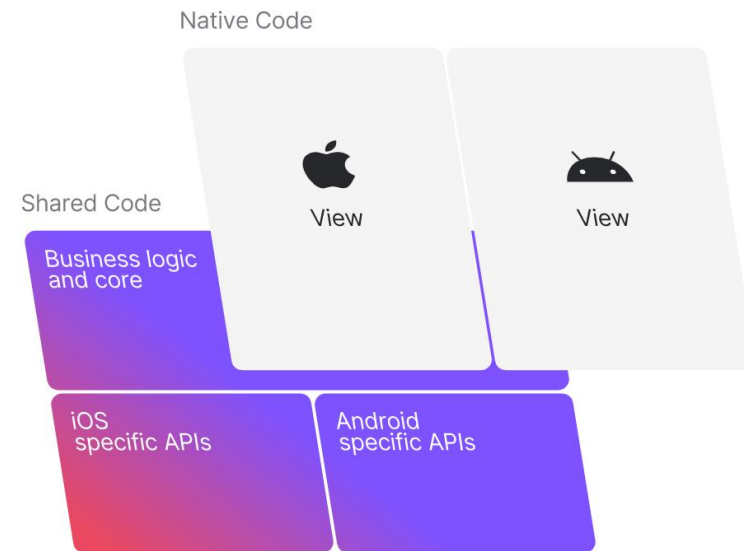
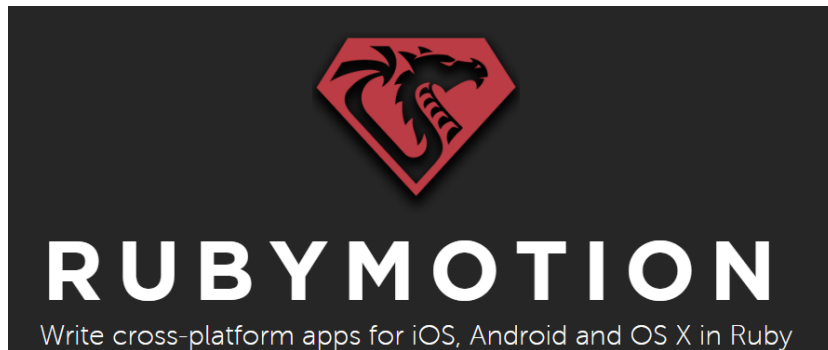
# Phát triển ứng dụng di động

- Phát triển ứng dụng di động đa nền tảng (cross-platform)
  - Ứng dụng di động phát triển từ các framework sử dụng một dạng mã lệnh trung gian gọi đến *các widget nguyên bản* (thông qua một bridge) như React Native (JavaScript), NativeScript (JavaScript), Xamarin/.NET MAUI (C#)



# Phát triển ứng dụng di động

- Phát triển ứng dụng di động đa nền tảng (cross-platform)
  - Ứng dụng di động phát triển từ các framework hỗ trợ biên dịch ra mã lệnh gốc và sử dụng các widget nguyên bản như **RubyMotion (Ruby)**, **Kotlin Multiplatform Mobile (Kotlin)**

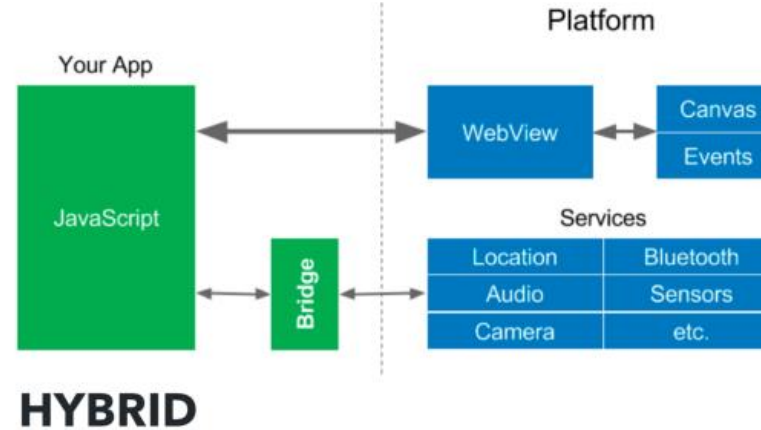
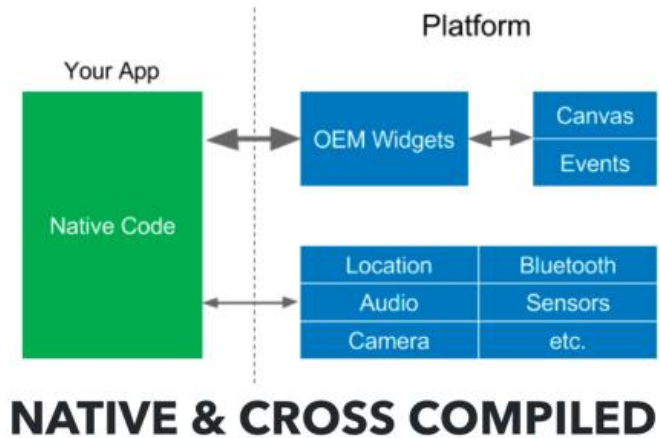


Kotlin Multiplatform Mobile (KMM)



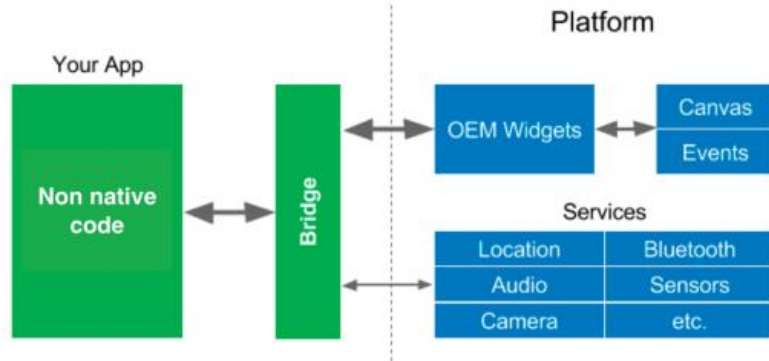
# Phát triển ứng dụng di động

E.g., Native,  
RubyMotion,  
Kotlin  
Multiplatform  
Mobile



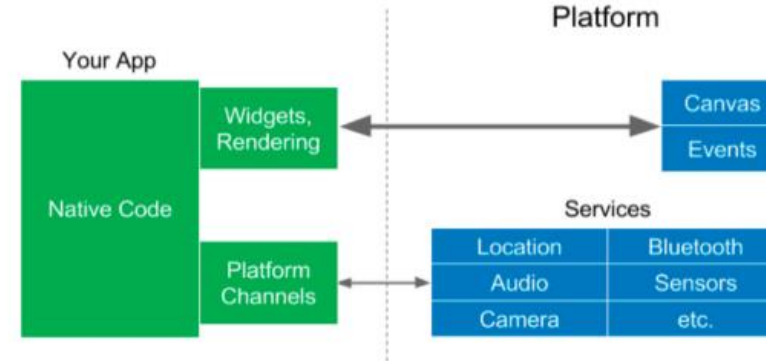
E.g., Ionic,  
Quasar,  
Framework7

## WEB NATIVE & CROSS COMPILED



E.g., React Native, NativeScript,  
Xamarin/ .NET MAUI

## FLUTTER



Câu hỏi?