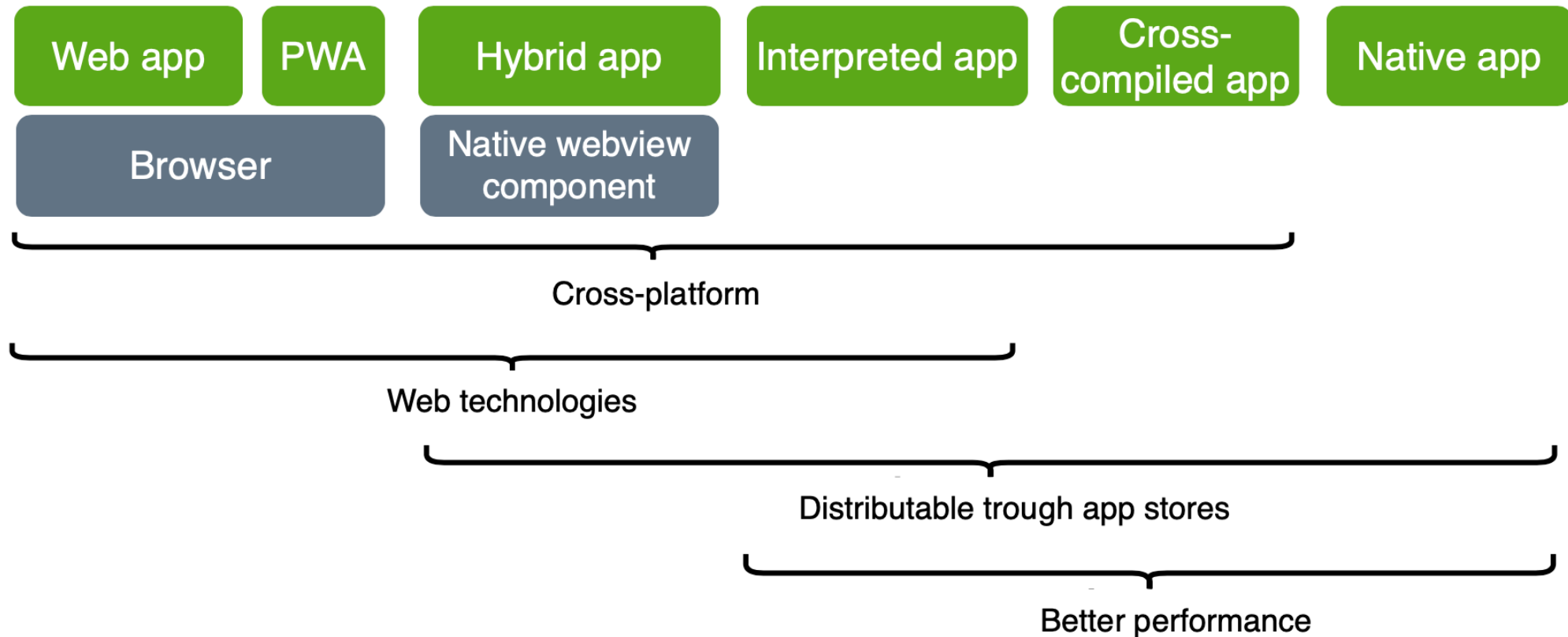


# Cross-Platform

## Roadmap

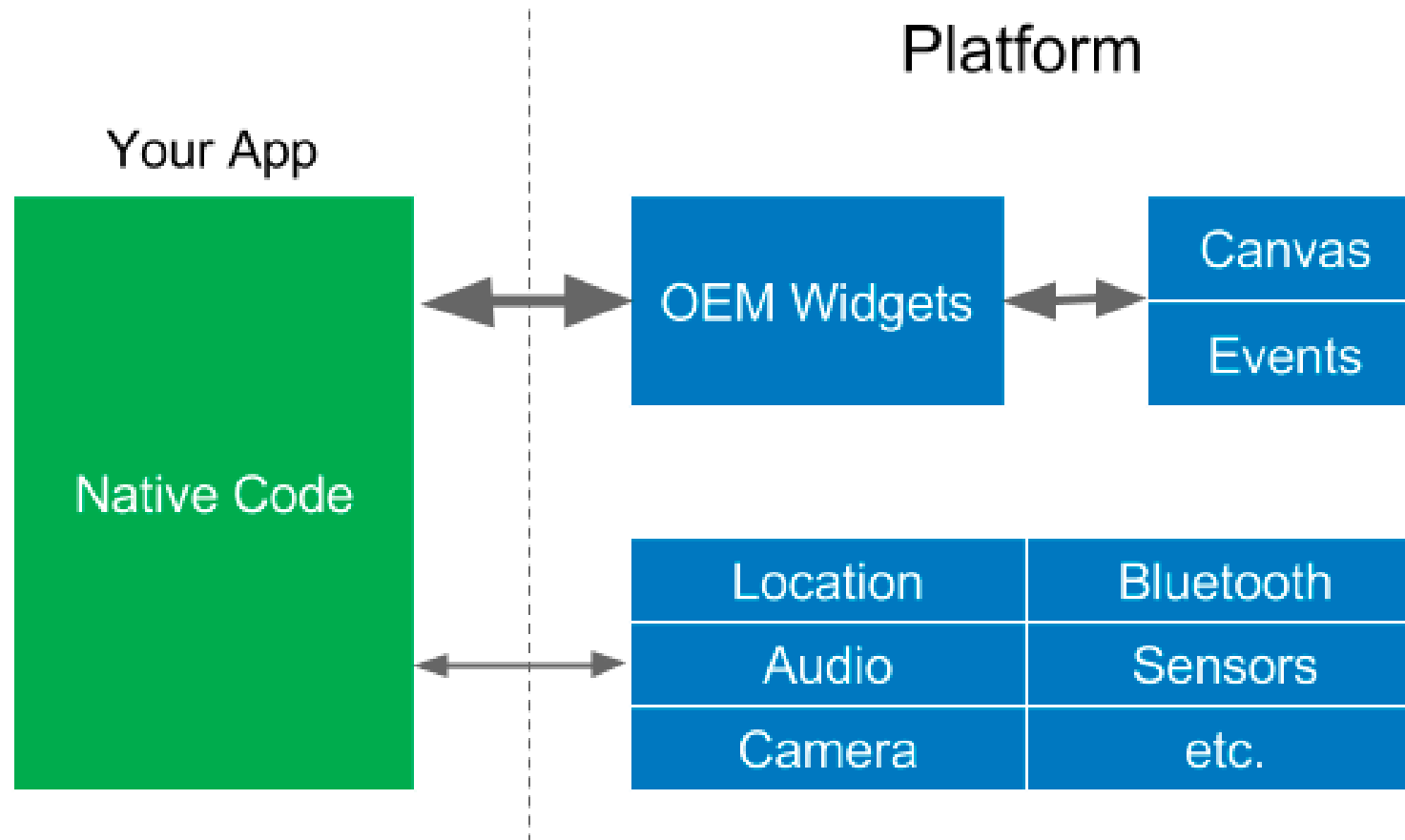
# Các công nghệ phát triển ứng dụng di động

- Ứng dụng di động có thể được phát triển theo nhiều cách tiếp cận:



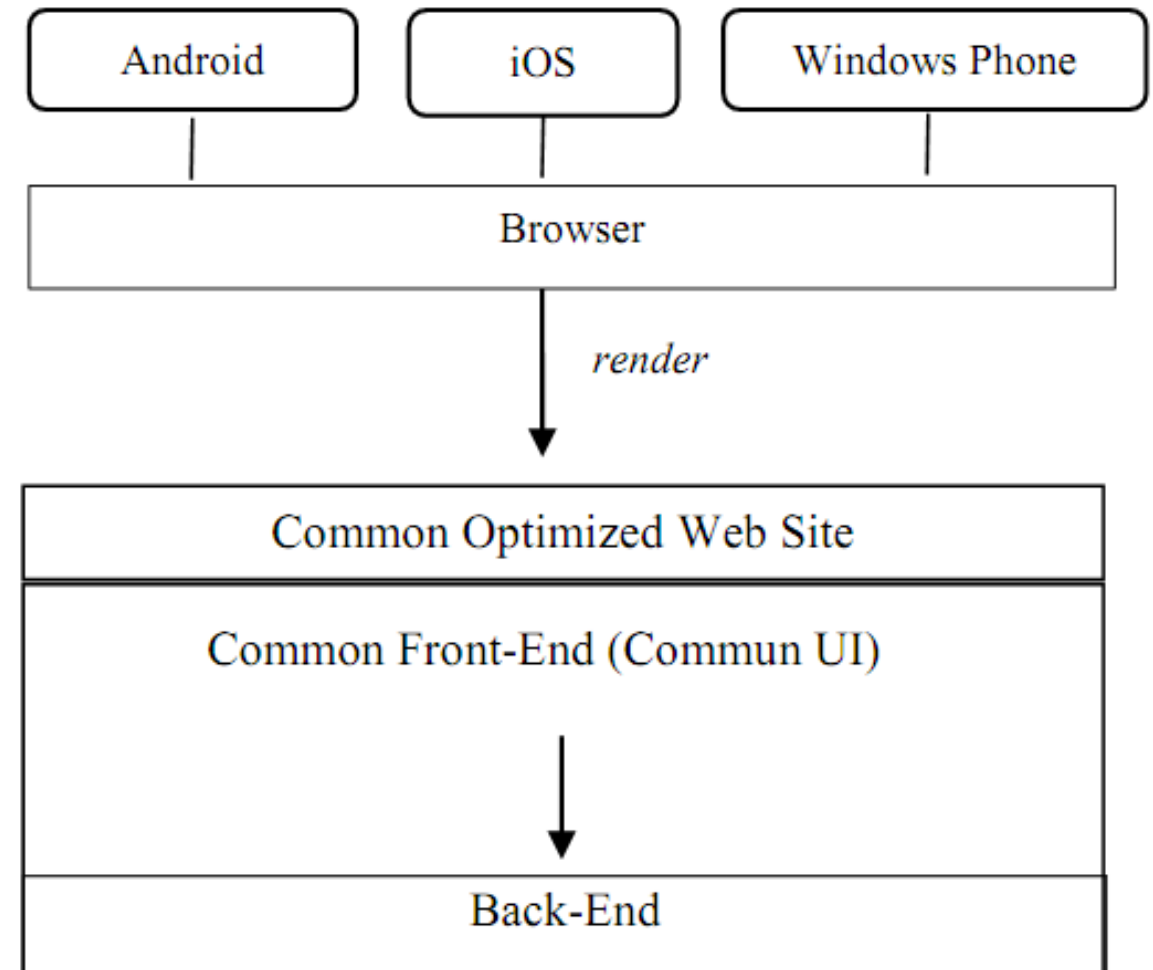
# Các công nghệ phát triển ứng dụng di động

- Native applications:



# Các công nghệ phát triển ứng dụng di động

- Web applications (Web app / PWA):
  - Cách tiếp cận web dựa trên trình duyệt web dành cho thiết bị di động.
  - Sử dụng HTML, CSS và JavaScript; và dựa vào trình duyệt làm môi trường runtime và hưởng lợi từ việc hỗ trợ trình duyệt của các nền tảng di động.
  - Trong cách tiếp cận này, ứng dụng được triển khai như một trang web được tối ưu hóa duy nhất cho thiết bị di động.



# Các công nghệ phát triển ứng dụng di động

- Web applications (Web app / PWA):

## Ưu điểm



- Các trình duyệt web đã có sẵn trên tất cả các thiết bị có liên quan
- Hỗ trợ CSS cho sự khác biệt về độ phân giải màn hình
- Tận dụng kỹ năng thiết kế web hiện có. Số lượng lớn các framework CSS & JS có sẵn với tài liệu
- Không cần cập nhật ứng dụng di động

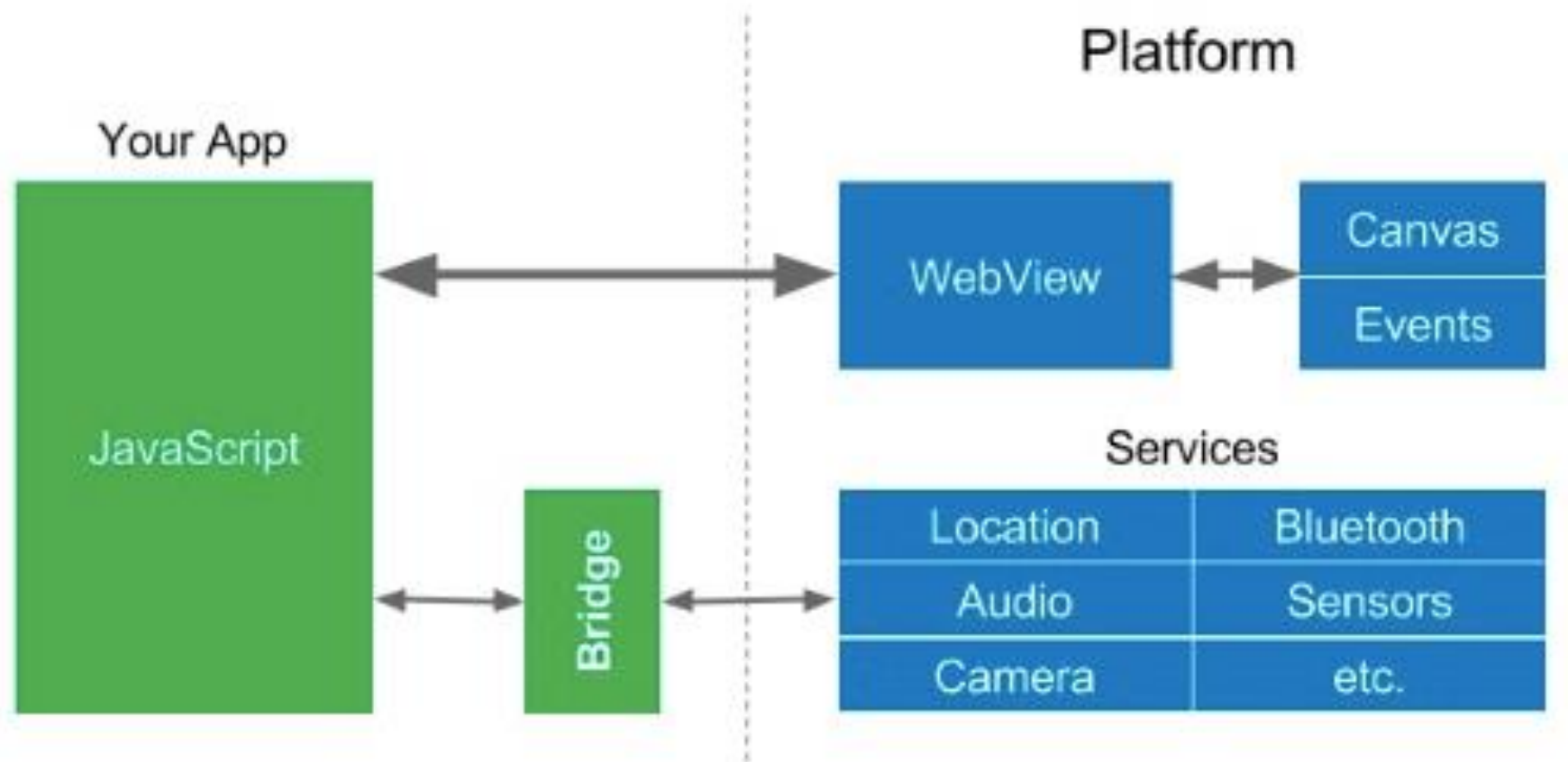


## Nhược điểm

- Khó khăn khi truy cập các API gốc (mặc dù một số ví dụ như dịch vụ vị trí hiện được hỗ trợ)
- Chậm! Phụ thuộc vào kết nối mạng khả dụng
- Khó phù hợp với giao diện native
- Không thể phân phối qua các cửa hàng ứng dụng

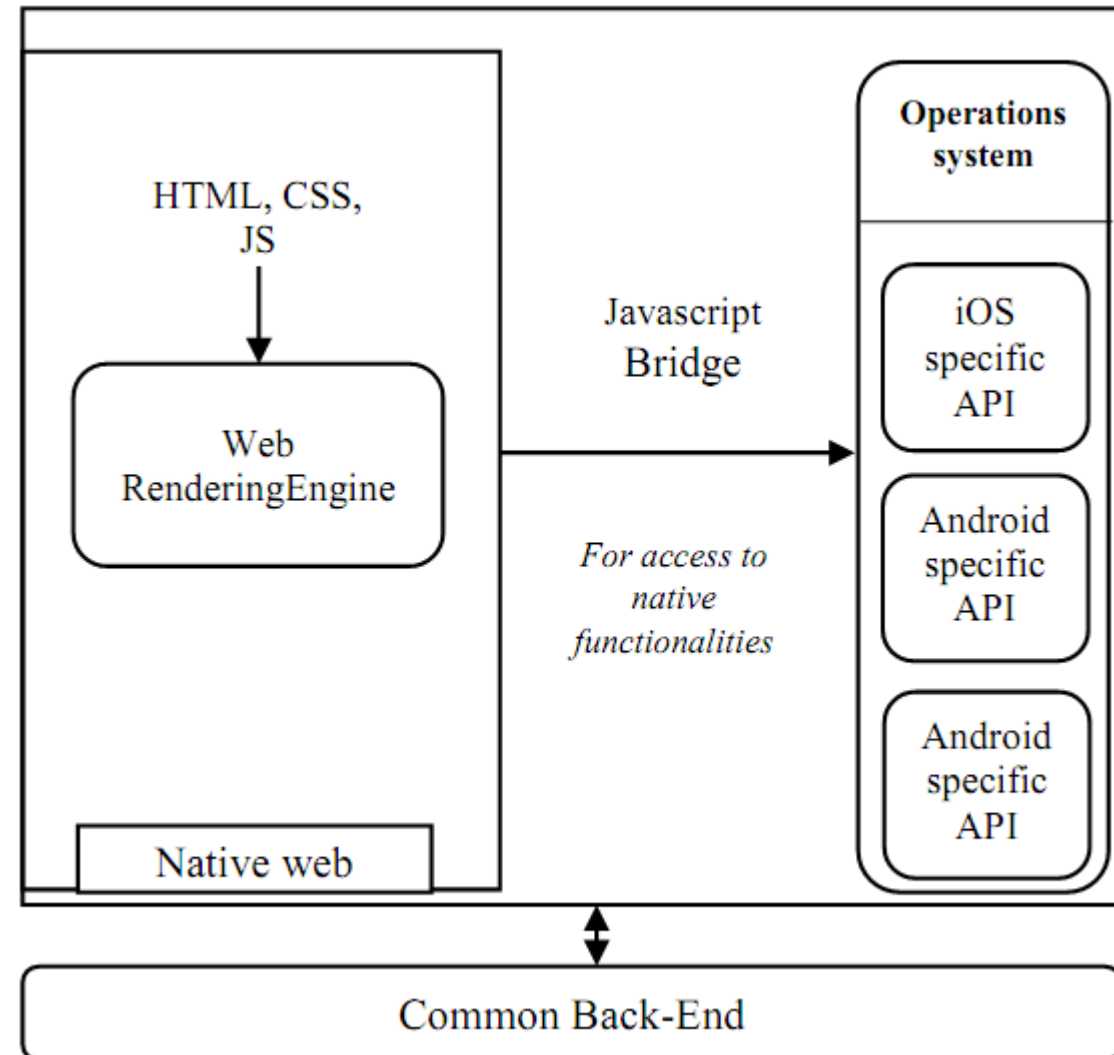
# Các công nghệ phát triển ứng dụng di động

- Hybrid applications:



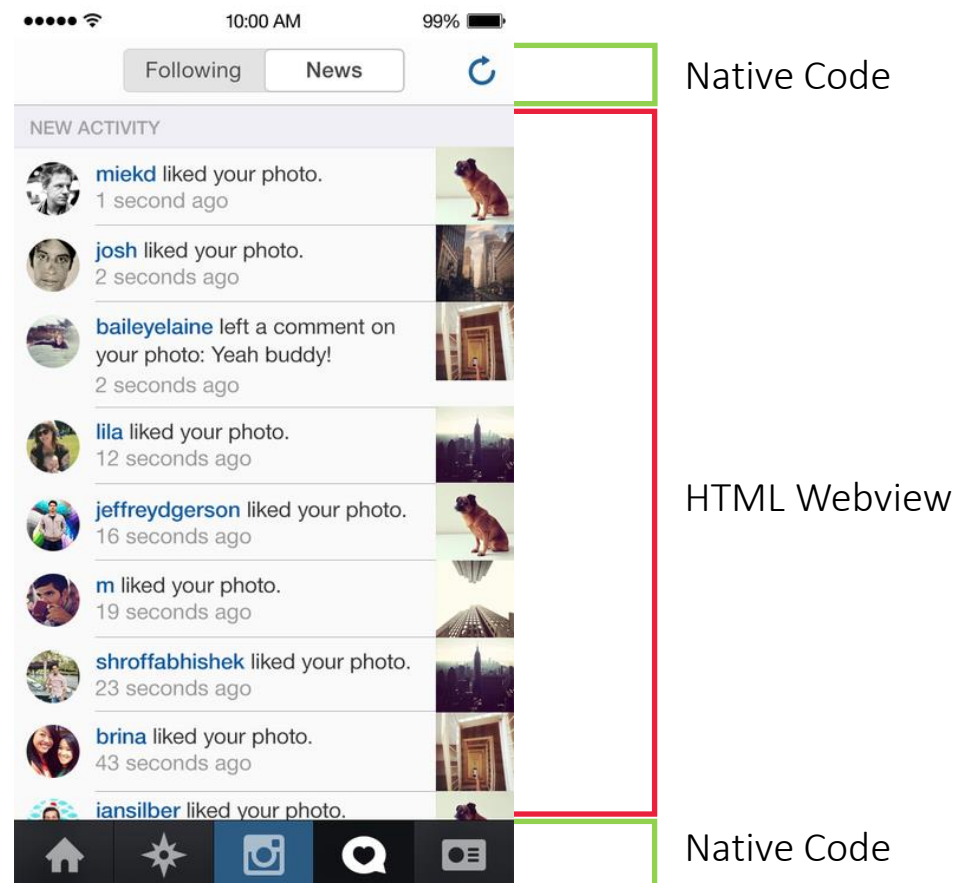
# Các công nghệ phát triển ứng dụng di động

- Hybrid applications:
  - Hybrid: một thuật ngữ chỉ nhiều loại ứng dụng cố gắng thu hẹp khoảng cách giữa phát triển web và phát triển gốc (native)
  - Đơn giản nhất, hybrid có thể chỉ liên quan đến việc đóng gói một ứng dụng web để nó có thể được triển khai cho các cửa hàng ứng dụng khác nhau
  - Hybrid cũng có thể phức tạp hơn qua việc sử dụng native UI và native API.



# Các công nghệ phát triển ứng dụng di động

- Hybrid applications: ví dụ sử dụng Webview





# Các công nghệ phát triển ứng dụng di động

- Hybrid applications:

## Ưu điểm



- Nhiều ưu điểm của các ứng dụng web tiêu chuẩn (tận dụng các bộ kỹ năng & framework hiện có)
- Không phụ thuộc vào mạng
- Hỗ trợ plugin JavaScript cho một số API gốc – cho phép chức năng không thể thực hiện được với các ứng dụng web tiêu chuẩn
- Phân phối qua các cửa hàng ứng dụng

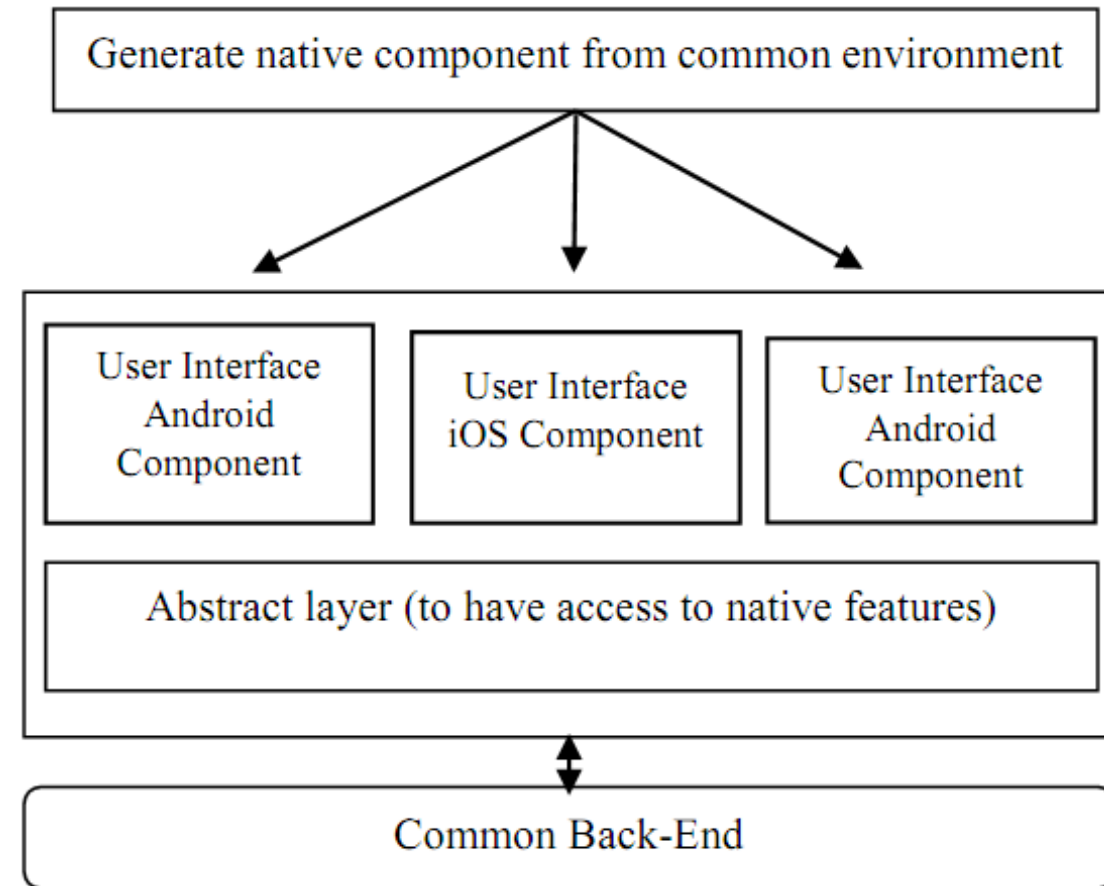


## Nhược điểm

- Vẫn còn chậm! (mặc dù bây giờ tắc nghẽn là CPU / bộ nhớ hơn là mạng)
- Vẫn khó để đạt được giao diện và cảm giác nguyên bản
- Phân phối qua các cửa hàng ứng dụng - hiện phải tuân theo các nguyên tắc thiết kế (có thể khắc nghiệt, đặc biệt là đối với iOS)

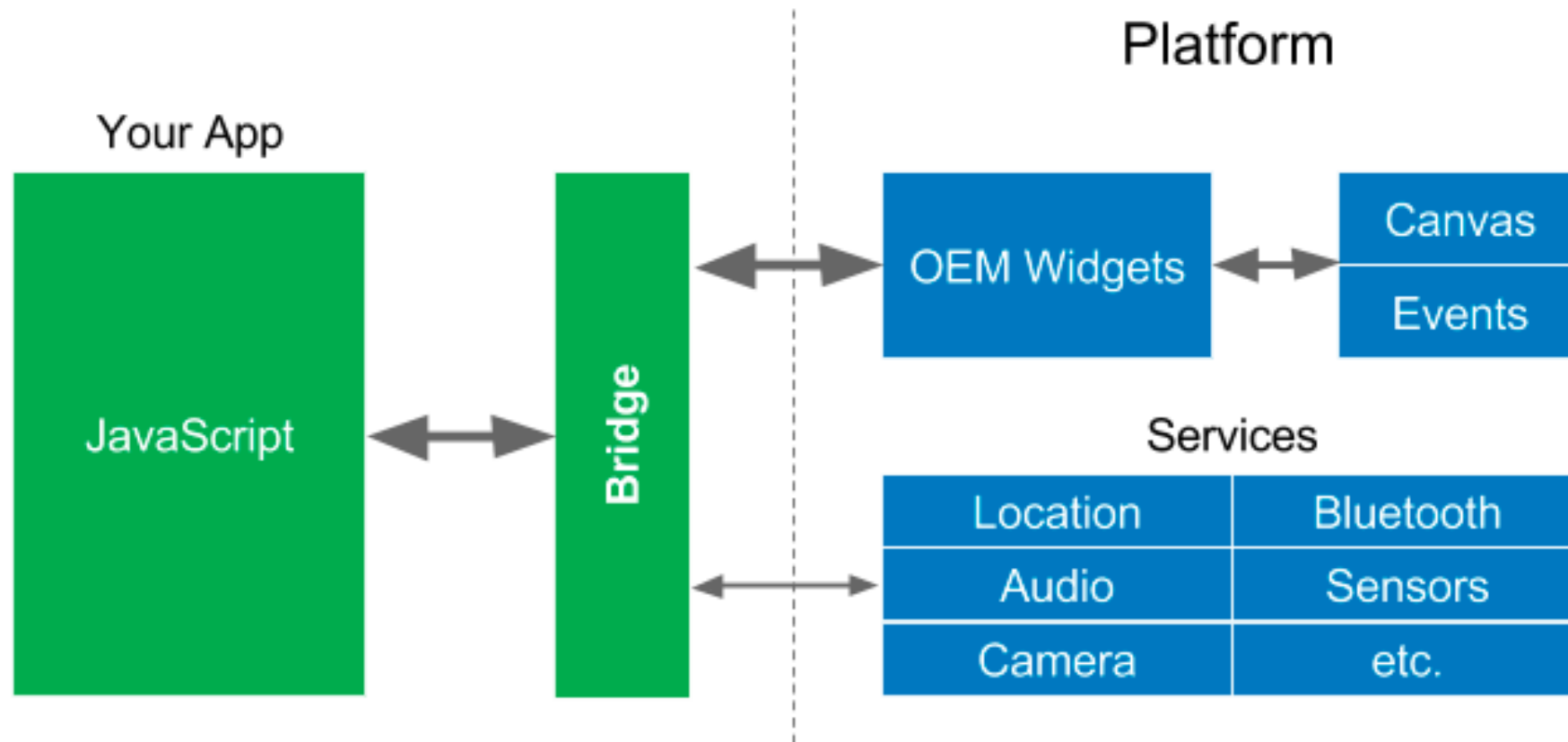
# Các công nghệ phát triển ứng dụng di động

- Interpreted App:
  - Phương pháp thông dịch sử dụng ngôn ngữ chung (như JavaScript hoặc các ngôn ngữ khác) để viết mã giao diện người dùng và tạo thành phần tương đương cho thành phần gốc cho mỗi nền tảng.
  - Các tính năng native được cung cấp bởi một lớp trừu tượng giúp diễn giải mã trong thời gian chạy trên các nền tảng khác nhau để truy cập các API gốc.



# Các công nghệ phát triển ứng dụng di động

- Interpreted App: với công nghệ JavaScript



# Các công nghệ phát triển ứng dụng di động

- Interpreted App:

## Ưu điểm



- Cho phép các giao diện người dùng gốc



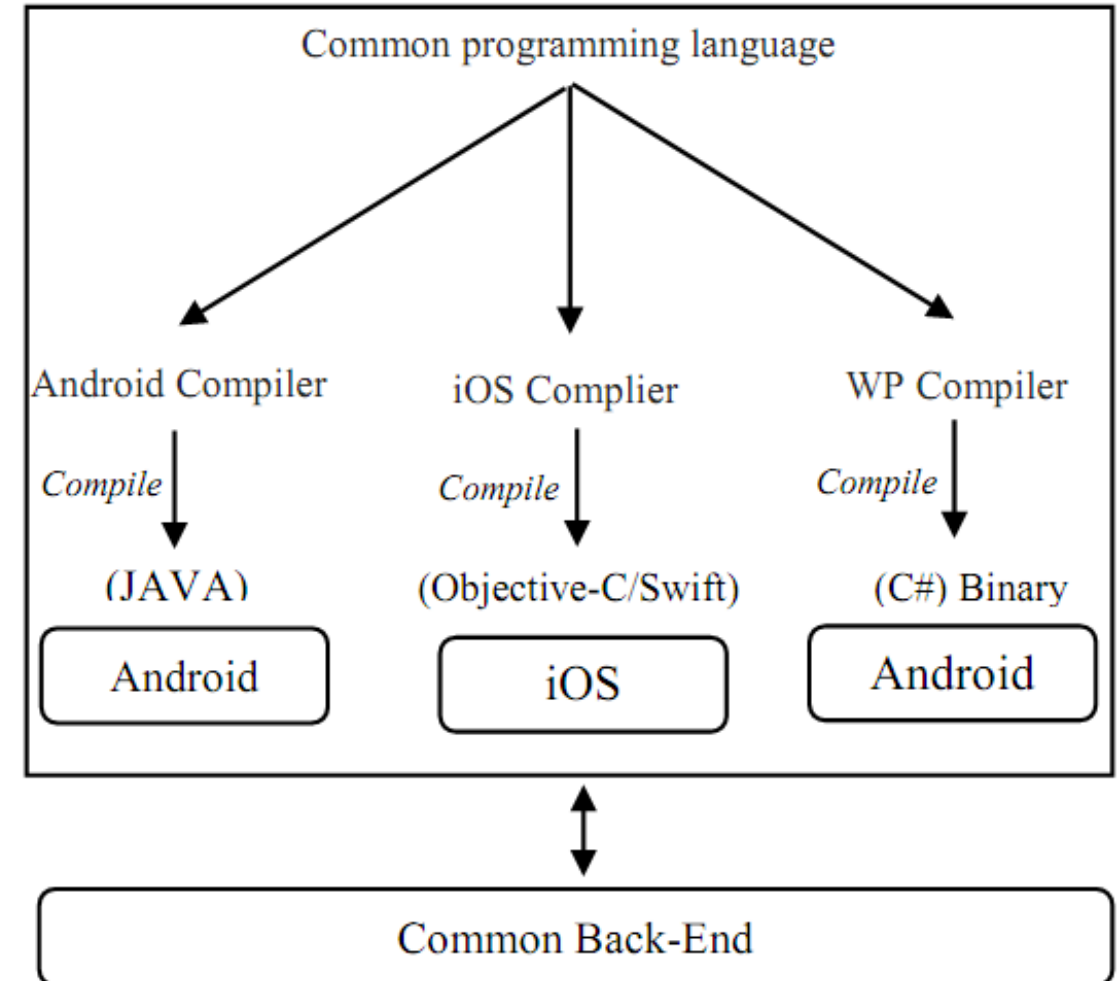
## Nhược điểm

- Phụ thuộc vào môi trường phát triển.
  - Nói chính xác hơn, các tính năng mới dành riêng cho nền tảng như các tính năng giao diện người dùng mới sẽ không được cung cấp cho các ứng dụng trừ khi chúng được hỗ trợ bởi môi trường phát triển.
- Sự suy giảm hiệu suất ứng dụng do gọi lớp trừu tượng trong thời gian chạy.

# Các công nghệ phát triển ứng dụng di động

- Cross-Compiled App:

- Trong phương pháp biên dịch chéo, các nhà phát triển viết mã với việc sử dụng bất kỳ ngôn ngữ lập trình thông thường nào.
- Các mã này được chuyển đổi bởi các trình biên dịch chéo thành một mã gốc cụ thể.



# Các công nghệ phát triển ứng dụng di động

- Cross-Compiled App:

## Ưu điểm



- Có thể còn nhanh hơn các ứng dụng dựa trên JavaScript - có thể đạt đến tốc độ của các ứng dụng gốc hoàn toàn
- Giao diện tự nhiên
- Có thể sử dụng ngôn ngữ lập trình "thực"

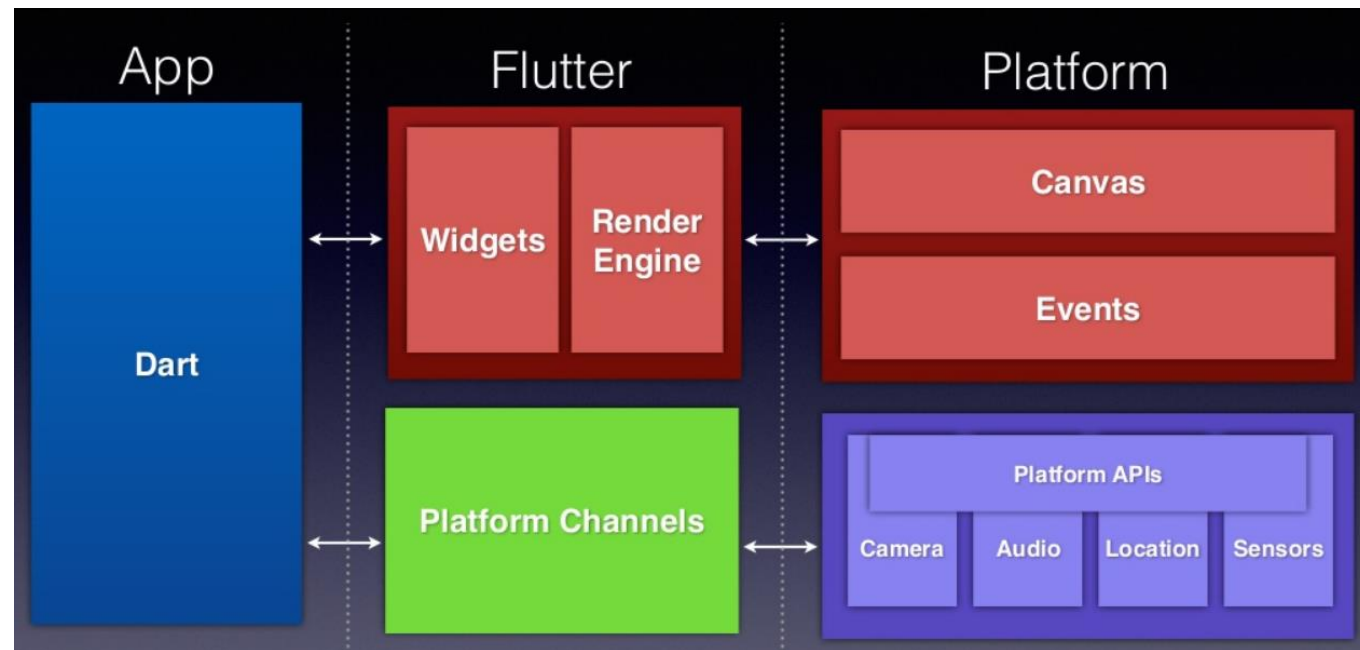


## Nhược điểm

- Không thể tận dụng kỹ năng phát triển web hiện có
- Có thể cần học nền tảng hoàn toàn mới

# Các công nghệ phát triển ứng dụng di động

- Flutter:
  - Flutter không phụ thuộc vào widget của nhà sản xuất thiết bị gốc, nó tự hiển thị các giao diện đồ họa dựa trên engine hiệu năng cao của riêng nó.
  - Các widget được hiển thị trên Skia canvas và gửi lên nền tảng thiết bị. Nền tảng thiết bị sẽ hiển thị các canvas và gửi lại các sự kiện khi người dùng tương tác



# Các nội dung chính về Cross-platform framework

## Tổng quan về Framework

## Xây dựng ứng dụng





# Flutter roadmap

