**REACT NATIVE**

React native là một công cụ giúp chúng ta lập trình đa nền tảng để tạo ra các ứng dụng trên môi trường native. Nó là một framework mã nguồn mở được phát triển bởi Facebook, cho phép sử dụng Java script để phát triển phần mềm trên điện thoại di động Android và IOS. React native cũng giống như React chúng sử dụng các native components thay vì các web components.

Ưu điểm:

* Khả năng tái sử dụng code và các components đã được phát triển sẵn.
* Có một cộng đồng developers hùng hậu.
* Sự tuyệt vời của Live and Hot reloading (Tiết kiệm được nhiều thời gian).
* Tiết kiệm công sức khi có thể code 1 mà có thể chạy cho cả ios và android.

Bên cạnh đó cũng có một vài những nhược điểm cần phải kể đến:

* Vẫn đòi hỏi native code.
* Hiệu năng sẽ thấp hơn với app thuần native code.
* Bảo mật không cao do dựa trên JS.
* Quản lý bộ nhớ.
* Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

**Những khái niệm React Native**

**Thành phần(Components):**

* Components là một khái niệm cơ bản của cả React và React native. Chính việc chia nhỏ ứng dụng thành các components nhỏ tạo nên tính tái sử dụng cao và khả năng mở rộng của chúng.

**Props(Properties):**

* Đây là một dữ liệu bất biến, khi sử dụng không nên thay đổi giá trị. Các component nhận props từ component cha. Không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.

**State**

* State thì hoạt động khác với Props. State là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu mutable. Vì đặc điểm này nên chúng ta hay sử dụng State để thay đổi dữ liệu của view, binding data lại view khi có thay đổi. Nhưng chúng ta không dùng this.state để gán lại giá trị thay đổi cho nó, mà chúng ta sẽ dùng this.setState.

**FIREBASE**

**Firebase** là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server*.*

**Các cách thức hoạt động của Firebase**

### Firebase Realtime Database:

Khi đăng ký một tài khoản trên **Firebase** để tạo ứng dụng, bạn đã có một cơ sở dữ liệu thời gian thực. Dữ liệu bạn nhận được dưới dạng JSON. Đồng thời nó cũng luôn được đồng bộ thời gian thực đến mọi kết nối client.

Dữ liệu được tự động cập nhật dữ liệu mới nhất và được truyền qua kết nối an toàn SSL có bảo mật với chứng nhận 2048 bit.

Trong trường hợp bị mất mạng, dữ liệu được lưu lại ở local. Vì thế khi có mọi sự thay đổi nào đều được tự động cập nhật lên Server của **Firebase**. Bên cạnh đó, đối với các dữ liệu ở local cũ hơn với Server thì cũng tự động cập nhật để được dữ liệu mới nhất.

### Freebase Authentication:

Hoạt động nổi bật của Firebase là xây dựng các bước xác thực người dùng bằng Email, Facebook, Twitter, GitHub, Google. Đồng thời cũng xác thực nặc danh cho các ứng dụng. Hoạt động xác thực có thể giúp thông tin cá nhân của người sử dụng được an toàn và đảm bảo không bị đánh cắp tài khoản.

### Firebase Hosting:

**Hosting** được phân phối qua tiêu chuẩn công nghệ **bảo mật SSL** từ mạng CDN. Hoạt động này giúp lập trình viên tiết kiệm thời gian thiết kế, xây dựng và phát triển ứng dụng.

## **Ưu điểm của Firebase**

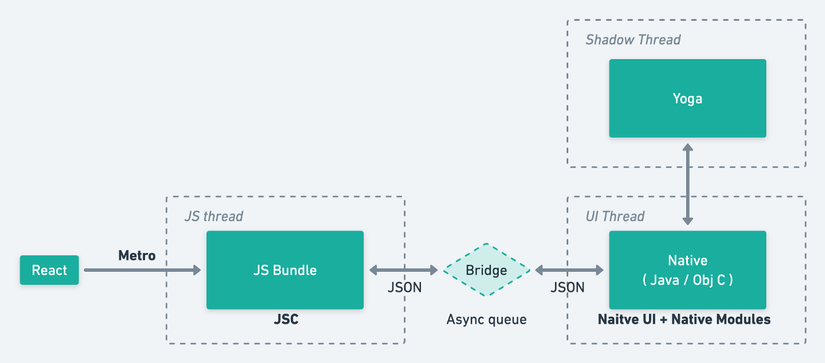
* Tạo tài khoản và sử dụng dễ dàng
* Tốc độ phát triển nhanh
* Nhiều dịch vụ trong một nền tảng
* Được cung cấp bởi Google
* Tập trung vào phát triển giao diện người dùng
* Firebase không có máy chủ
* Học máy (Machine Learning)
* Tạo lưu lượng truy cập
* Theo dõi lỗi
* Sao lưu

## **Hạn chế của Firebase:**

* Không phải là mã nguồn mở
* Người dùng không có quyền truy cập mã nguồn
* Firebase không hoạt động ở nhiều quốc gia
* Chỉ hoạt động với Cơ sở dữ liệu NoSQL
* Truy vấn chậm
* Không phải tất cả các dịch vụ Firebase đều miễn phí
* Firebase khá đắt và giá không ổn định
* Chỉ chạy trên Google Cloud
* Thiếu Dedicated Servers và hợp đồng doanh nghiệp
* Không cung cấp các API GraphQL

**KIẾN TRÚC REACT NATIVE**

1. **Các luồng trong React Native:**



Có 3 luồng chính trong môi trường thực thi react native là JS Thread, Native Thread, Shadow Thread. Trong kiến trúc hiện tại, giao tiếp giữa các luồng này xảy ra qua thư viện được gọi là "Bridge".

**Javascript Thread:** là nơi tất cả code Javascript được đọc và biên dịch, và là nơi xử lí hầu hết các logic nghiệp vụ của ứng dụng. Metro sẽ dịch các cú pháp Javascript thay thế (Typescript hoặc JSX), và kết hợp tất cả code Javascript thành một tệp duy nhất. Phần code này sẽ được chuyển tới công cụ JavascriptCore (JSC) để có thể chạy được.

Native Thread: là nơi các đoạn code native được khởi chạy. Nó giao tiếp với JS Thread bất cứ khi nào có nhu cầu thay đổi UI hoặc truy cập các hàm Native. Chúng ta có thể chia Native Thread thành Native UI và Native Modules. Tất cả các Native Modules đều được khởi động khi chúng ta sử dụng ứng dụng. Điều đó có nghĩa là module Bluetooth sẽ luôn luôn ở trạng thái kích hoạt bởi React Native kể cả khi không có nhu cầu sử dụng.

Shadow Thread: là nơi mà React Native sử dụng để tính toán bố cục của bạn được tạo bằng thư viện React. Nó sử dụng Layout Engine riêng của Facebook có tên là Yoga nhằm tính toán flexbox layout, sau đó gửi kết quả về phía Native UI.

Bridge: cho phép native code và Javascript giao tiếp với nhau. Bridge là hàng đợi để gửi tin nhắn được mã hóa dưới dạng chuỗi JSON. Bridge gắn liền với vòng đời của React Native, khởi chạy hoặc dừng React Native thường có nghĩa là khởi tạo hoặc phá bỏ Bridge.