**React native:**

1. **IOS và ANDROID**: Ngày nay, chúng ta có 2 nền tảng lớn là IOS và ANDROID đang chiếm phần lớn trong thị trường mobile. Từ đó sinh ra các công cung hỗ trợ lập trình trên 2 nền tảng này như IOS là XCode và Android là Android Studio.

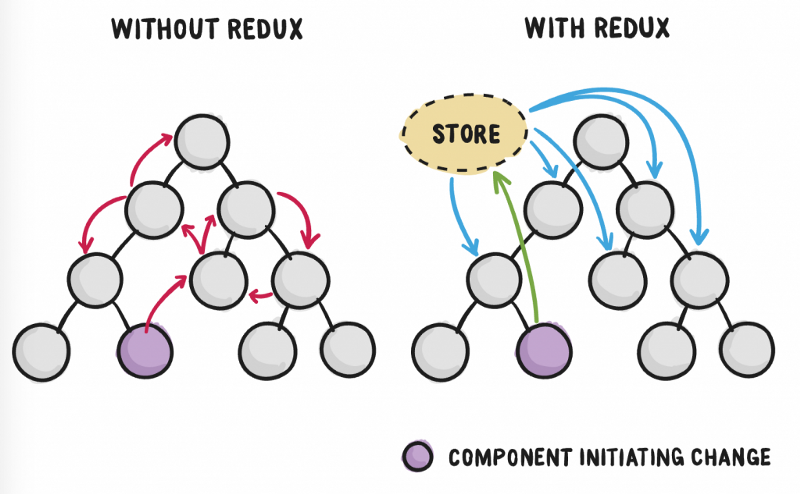
* Sinh ra Native và Hybird

1. **Native và Hybird:** 
   1. Native: Đầu tiền, chúng ta nói về Native App, Native App là các ứng dụng được được phát triển và buid trên chính các công cụ hỗ trợ nhà phát triển như Android Studio hoặc XCode. Các ứng dụng Native được build dựa trên ngôn ngữ của chính hệ điều hành và sử dụng những tính năng sẵn có của hệ điều hành mà không phải thông qua bất cứ ứng dụng bên thứ ba hay engine nào khác để vận hành nên tốc độ là nhanh nhất.
   2. Hybird: Hybrid App là sự kết hợp giữa ứng dụng Web và ứng dụng mobile. Tức là chúng vừa có thể cài đặt lên điện thoại người dùng giống như là những ứng dụng Native bình thường và bạn có thể tìm thấy chúng trên chợ ứng dụng nhưng khó có thể nhận ra đâu là ứng dụng Native và đâu là ứng dụng Hybrid thường thì người dùng sẽ không để ý quá nhiều. Đồng thời như ứng dụng Web chúng được xây dựng dựa trên HTML + CSS + JS. Nhược điểm của Hybrid App đó chính là vấn đề hiệu năng sẽ bị ảnh hưởng đáng kể cũng như không tương tác được hết những tài nguyên hệ thống phần mềm cũng như phần cứng.
2. **React Native**:

* React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động

1. **Ưu điểm:**
   1. Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
   2. Hiệu năng tương đối ổn định.
   3. Cộng đồng phát triển mạnh.
   4. Tiết kiệm chi phí
   5. Team phát triển nhỏ.
   6. Ứng dụng tin cậy và ổn định.
   7. Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
   8. Trải nghiệm người dùng tốt hơn là hybrid app.
2. **Nhược điểm**:
   1. Vẫn đòi hỏi native code.
   2. Hiệu năng sẽ thấp hơn với app thuần native code.
   3. Bảo mật không cao do dựa trên JS.
   4. Quản lý bộ nhớ khó.
   5. Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

**Redux**: **Redux** là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. **Redux** ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ **Elm** và kiến trúc **Flux**của Facebook. Do vậy Redux thường dùng kết hợp với React.



**Tai sao cần state managerment tools?**

Hầu hết các lib như React, Angular, etc được built theo một cách sao cho các components đến việc quản lý nội bộ các state của chúng mà không cần bất kỳ một thư viện or tool nào từ bên ngoài. Nó sẽ hoạt động tốt với các ứng dụng có ít components nhưng khi ứng dụng trở lên lớn hơn thì việc quản lý states được chia sẻ qua các components sẽ biến thành các công việc lặt nhặt.

trong react để share data thông qua các components anh em, một state phải live trong component cha. Một method để update chính state này sẽ được cung cấp bởi chính component cha này và pass như props đến các components con.

Điều này khiến cho bộ phận quản lý state trong app trở lên bừa bộn cũng như app trở lên vô cùng phức tạp. Đó là lý do tại sao chúng ta cần một state management tool như Redux.

**Redux-thunk**: Redux-Thunk là một middleware phổ biến nhất được dùng để xử lý các action bất đồng bộ trong Redux.

**UI Kitten**: Là một giao diện tùy biến thư viện React Native UI dựa trên Eva Design System với hơn 30 thành phần giao diện, 2 theme và các module hỗ trợ khác.

**React native navigation**: Chắc hẳn quá trình chuyển màn hình trong quá trình phát triển ứng dụng React Native đã quá quen thuộc đối với các Dev chúng ta. Và trong số các thư viện đó, React Navigation là thư viện rất được cộng đồng React Native tin tưởng và sử dụng vì sự đơn giản, dễ dùng và trải nghiệm tuyệt vời của nó.