



1 Hello everyone,
I'm Priyanka, started my career as backend dev at BMS, worked on various technologies like node js, c#, then moved to frontend, primarily as an iOS dev, now working on RN.

I'm here to talk about development of mobile applications



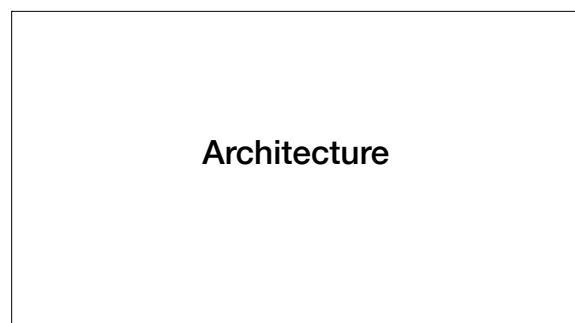
2 We want to build an app. We have majorly two platforms for which we have to build it, iOS n Android. Both of these platforms has their native language development. You can Java/Kotlin for Android and Swift/Obj C for iOS

Consider a newbie who wants to enter the mobile world & ship his first application. He'll have to learn these languages. But the learning curve here is more! Also, this can be a problem if you have timeline crunch.



3 So, I'll be mainly covering differences in flutter, RN and native.
Before starting the actual comparison, I'd want to know how many people are android devs?
iOS devs?
Cross platform?

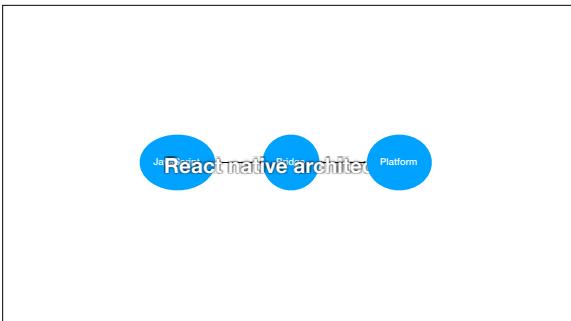
Oh well! I see a mix of an audience. That's interesting!!



4 When choosing a cross-platform mobile app development framework, it's essential to consider its technical architecture. By knowing the internals of the framework, we can make an informed decision and choose the one that is better for our project.



5

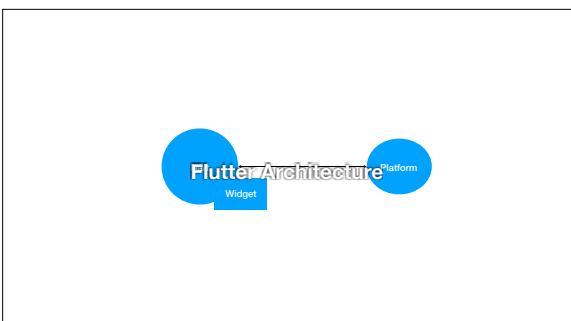


6

Technical Architecture

React Native — Flux

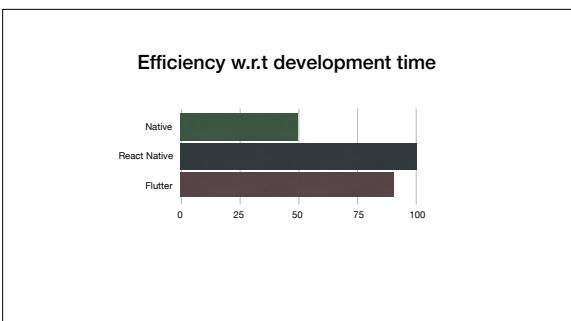
React Native architecture heavily relies on JS runtime environment architecture, also known as JavaScript bridge. The JavaScript code is compiled into native code at runtime. React Native uses the Flux architecture from Facebook. There is a detailed article on the core architecture of React Native [here](#). In short, React Native



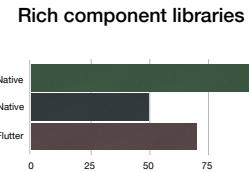
7

Flutter — Dart

Flutter uses Dart programming language which was introduced by Google in 2011 and is rarely used by developers. Dart syntax is easy to understand for JavaScript or Java developers as it supports most of the object-oriented concepts. It's easy to get started with Dart as there is great and easy-to-follow documentation available on the official Dart site

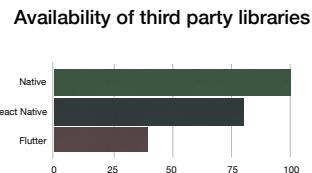


8



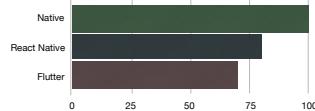
9

- **Rich component lib:** React Native — Less Components
- The core React Native framework provides just UI rendering and device access APIs. In order to access most of the native modules, React Native has to rely on third-party libraries. Even for navigation!!
- Flutter — Rich in Components
- Flutter framework is bundled with UI rendering components, device API access,



10

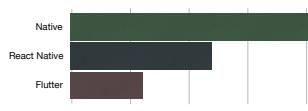
Popularity



11

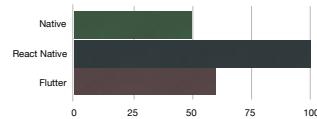
- **Popularity**
- React Native launched in 2015 and has gained in popularity ever since. There is a community of React Native developers on GitHub and lots of meetups and conferences around the world. One of the most recent conferences on React Native was React Native EU held in Poland, but there are meetups taking place in almost every major city in the world.

Real World usage



12

Deployment



13

Framework load time		
Metrics	React native	Flutter
DYLIB Loading time	209.79 ms	332.31 ms
BINDING TIME	137.12 ms	470.50 ms
Obj-c setup time	61.58 ms	50.39 ms
Total pre-main time	455.03 ms	912.9 ms

14

My experience at BookMyShow

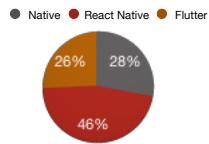
We, at BMS, evaluated RN in Aug'17(check it). Releasing the advantages of using a cross-cross-pat language, especially one codebase(which will help us avoiding the discrepancies between and and iOS apps). We decided to go with Booking Flow in RN. It worked really well at that point as we regularly conducted experiments on user interaction on that flow and monitored the success rate.

It's a hybrid app.

Github stats

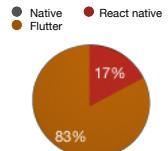
16

Number of authors



17

Open issues



18

Apps using Flutter



19

Apps using react-native



20

Take aways

- RN/Flutter:
- Both platforms & from scratch
- Hybrid: RN. New feature with no native indulgence(RN)
- Flutter: Usage of existing animations(swipe to delete), physics

21

Judge Yourself
Based on the metrics

22

"Type a quote here."

-Johnny Appleseed

23

Thank You

24