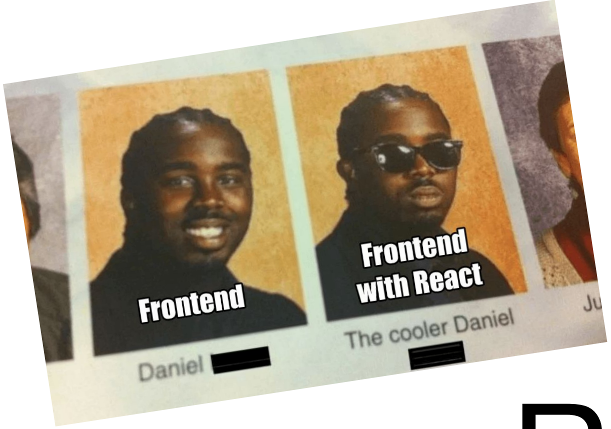


OH, YOU REALLY LOVE FLUTTER
I'VE NEVER HEARD THAT BEFORE, GO ON...

React Native

Pros

I DON'T ALWAYS BUILD MOBILE APPS
BUT WHEN I DO, I USE REACT NATIVE
imgflip.com

Use Javascript / Typescript

Use popular javascript language thus it has an extensive base package, if you’re react developer you can easily learn to react native, here developers have freedom of choice because it facilitates code reuse and cost-saving

Example :

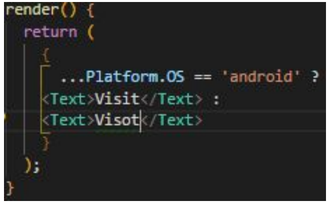
- Many prodiginow project use javascript based frameworks for backend and frontend we can code reuse and implement that in react native
- There’s many community packages that javascript based like popular utils package lodash, date-fns etc or even something like pusher for websocket in laravel, or native web api websocket that we can use in react native

Single Codebase

Like another hybrid-framework it has one code base which allow developers to develop apps for multiple devices at same time

Example :

- Not only Android and IOS react native now can deployed to web using framework like Expo
- If there’s something want to distinguish between android, ios or web can use Platform.OS



```
render() {
  return (
    {
    [
      ...Platform.OS == 'android' ?
    <Text>Visit</Text> :
    <Text>Visot</Text>
    }
  );
}
```

Easy Integration

Some people say that react native have difficult integration but it’s depend on how community create their package for developer, now integration in react native is much easier

Example :

- React native use auto-linking so we just need to install npm package and all the native code for the integration will linked automatically for both ios and android
- Even if there’s some configuration in native code, it’s just essential things like android push notification key, or add some permission in info.plist or android manifest
- Widely supported package in expo SDK that work out of the box like API Reference - Expo Documentation
- Many wide community package you can use https://reactnative.directory or customize
- Now in 2022 you don’t even need to touch yourself to configure native code there’s tools like Expo CLI, Expo Prebuild that you can use for automate that task

Community Support & Dev Support

Until now there’s so huge community support for react native and it will expand more in the future

Example :

- Every day there’s new update from dev support itself like right now in 2022 react native use new react version that can use Suspense, useTransition and many more that can improve performance-wise
- In 2022 there’s new rendering system Fabric · React Native that boost performance of react native other than Hermes, JIT, Turbo Module etc
- Recent update from react native community like Shopify engineering team created FlatList that use recycling method which is FlashList which boost performance of FlatList (FlashList – fast and performant React Native list (shopify.github.io)) or Expo Engineering Team created dynamic routing but in mobile device like NextJS (Evan Bacon 🥓 on Twitter: "Today, Expo is reimagining native routing! ✨ Automatically create (dynamic) routes with files 🚀 Zero boilerplate 🔗 Automatic deep linking 🧩 Nested layouts 🔥 Built on React Navigation for easy adoption. Try the @expo router beta now! https://t.co/JeNzDIC8YD https://t.co/7QEZhZ2Cwc" / Twitter)
- React native now even can use flutter rendering engine which is React Native Skia (Shopify/react-native-skia: High-performance React Native Graphics using Skia (github.com))

OTA (Over The Air) Updates

Because of UI layer that react native has, there’s feature that can update application without resubmit to Appstore or Playstore which is OTA (Over The Air Updates)

Example :

- There’s small typo’s mistake from QA Team and accidentally release the product to production, with OTA we don’t need to resubmit the app and can fix it from cloud and it will automatically updated to the new version

Easy Deployment & CI CD

Because of the community support, deployment now easy with EAS (Expo Application Service) & EAS Submit, we even don’t need mac machine to deploy our application to AppStore, there’s many choice to build react native with cloud service or with internal machine (Android Studio / XCode), we can use CI CD for deployment and submit with workflow too (Expo React Native Complete CI / CD Workflow Using Github Actions | by Shafran Naizer | The Startup | Medium)

React Native

Const

If We Compare It To Flutter It Has Poor Performance

It’s correct but not definitely correct because it’s depend on developer itself for example:

- Many people often encountering issues of performance because **they don’t know in advance what react as UI layer do in general in react native**, for example there’s FlatList with 10000 items in there and there’s button where there’s state changes, if we don’t use best practice it will make UI block because react need to render again 10000 items or item that visible in the area of flatlist, for that reason there’s Pure Component / React.memo to prevent that excessive rendering
- Some people say that react native has poor performance regarding extensive calculation it’s not fully true because like I said before **they don’t know in advance what react as UI layer do in general in react native,** for extensive calculation don’t use basic function, there’s API like **useCallback, useMemo** that can prevent that memory lacking
- There’s some people that encountering issues where to many extensive stack screen slow down the app let’s say 3 stack screen with 10 list 1080p image, there’s approach to handle that case, you can use react-freeze where whe can freeze unfocused stack and freeing memory lacking from that stack, for default react-freeze is installed in many react native routing package (react navigation), another method is use package like ImageManipulator (ImageManipulator - Expo Documentation) and do thumbnail resize in client side
- And there’s so many case that people often say that RN has poor performance than flutter but it’s back to the developer itself, **we can see best practice of RN developed smoothly in enterprise app like Facebook, Instagram, Slack even Discord**

Custom UI

Even there's many community packages like (React Native Paper Home · React Native Paper (callstack.github.io), React Native Material UI Backdrop | React Native Material (react-native-material.com)), but in general react native come as framework that serve native component (Button, TextInput, ScrollView, FlatList etc) and we need to custom that component to behave or look like what we want, so we need some extra effort in customizing the UI

Example :

- There's case in my previous work in Mediatechindo, there's demand to developing app like Canva or Figma, so because there's no community packages that work like Canva or Figma do, so I need to develop it from scratch, preview : **file:///Users/yudhacode/Downloads/WhatsApp%20Video%202022-10-11%20at%206.43.01%20PM.mp4**

Conclusion

So, the choice between Framework F and framework R depends mostly on the developer’s experience.