



# Choosing Between a Native and a Hybrid App

In this lesson, we will cover how to choose the right type of mobile app for our use case?

We'll cover the following



- When should we pick a Native app for our use case?
- When should we pick a Hybrid app for our use case?

## When should we pick a Native app for our use case?#

*Here are the scenarios, listed below in bullet points, when we should go ahead with a Native app:*

- Use a Native app when you have heavy graphic and hardware requirements like for a mobile game or a game streaming app. In the scenario, you need top-notch performance from the app. Even a tad bit of lag is unacceptable.
- Let's say you intend to write an app that has heavy UI animations, like, a fancy social app containing a lot of animations or a finance app containing a lot of real-time charts and graphs. In this scenario, it's just not okay to have any sort of lag in the application. The application needs to be as responsive and reliable as possible.



- Use a Native app when the app is pretty complex and is reliant on hardware access, like camera, sensors, GPS, etc. to function. In the scenario, we generally have to write a lot of platform-specific code. A GPS, sensor-based health and a step-tracking app is a good example of this.
- Use a Native app when the look and feel of the app and the user experience should be just like the Native OS and when the UI needs to be, not just functional, but flawless.
- Use a Native app when you have other businesses in the same niche competing with you with native apps. It would be a blunder to offer our service via a Hybrid app. User's today aren't installing as many apps as they used to. Don't expect them to show mercy on you when you don't have a product that is better than your competition.
- Use a Native app when the app always needs to support new mobile OS features as soon as they are released.
- If you are a business that can afford dedicated teams for Android and iOS, you should go ahead with native apps. Don't even think about the Hybrid app approach.

## When should we pick a Hybrid app for our use case?

### #

- Use a Hybrid app when the app requirements are simple, and there is nothing complex. Also, use it when you don't anticipate adding any new complex features. A news app is a good example of this.  
Developing a news app as a Hybrid app will also provide the same

look and feel across all the platforms.



- Use a Hybrid app when you just cannot afford dedicated codebases for platforms but still have to hit the market. There are two approaches to this: either launch with a Native app on one platform or write a Hybrid app. This entirely depends on how you want to go ahead.

Yes, the Native apps provide top-notch performance, but you cannot entirely discard Hybrid tech on the grounds of performance and the availability of other Native features. There can be instances where you won't even need dedicated apps, and a Hybrid app could fulfil all your requirements pretty well. It all depends on your requirements.

- When you just need to test the waters with a *pre-alpha* release or a *Minimum Viable Product (MVP)*. In this scenario, it won't make sense to learn the Native tech and then write the app. You can quickly launch the MVP via a Hybrid app written with the open web technologies.
- Use a Hybrid app when you have a team that is not fluent with the Native technologies, and it would take a lot of time to learn that tech. This scenario is a trade-off between costs and performance. Also, the developer sentiment is another aspect to this.

So, these are some of the general rules that you can follow when taking our pick of the two types of apps. Another good approach is to find the businesses in the same niche, research on what technologies they have used to write their apps.

With this being said, let's move on to the next lesson where I'll discuss progressive web apps.

← Back

Next →

 Mark as

 Completed

---

 Report an Issue