

Mobile Web App vs. Native App? It's Complicated

Comment Now Follow Comments

I assume you have already read numerous articles about the neverending debate between mobile and native web applications. Moreover, this debate will undoubtedly grow along with Android's market share.

app-web-native.jpg

Unfortunately, the debate is more complicated than it looks. First, because the figures usually provided behind the arguments are based on new subscribers, not owners, and don't take into account all types of mobile phones. Rather, they focus only on high-end smartphones like the [July 2011 comScore U.S. Mobile Subscriber Market Share Report](#). This data gives you the false impression that the market is divided between Google and Apple, a biased vision of reality ([The proliferation of mobile platforms](#)).

The second complication: You have more than two options (web vs. native). To be more precise, we can envision four different technical configurations:

- **Native apps**, which are coded with a specific programming language (ObjectiveC for iOS, Java for Android). These mobile applications are fast, reliable, and powerful but are tied to a mobile platform. That means you must duplicate them using

the appropriate programming language in order to target another mobile platform. Nearly all games are native apps.

- **Hybrid apps**, which rely on development frameworks like [Sencha](#), [PhoneGap](#), [Titanium](#), [Rhomobile](#), [ParticleCode](#), [Corona](#), [Mosync](#), [Worklight](#), [BkRender](#)... These mobile apps offer a very interesting compromise because they ensure cross-platform compatibility and can access the phone's hardware (camera, GPS, user's contacts). IGN's mobile social network [Dominate](#) is just such a hybrid app.

- **Dedicated web app**, which is a mobile web site tailored to a specific platform or form factor, like the [LinkedIn web app](#) which was designed for Android and iOS, but not for other smartphones or feature phones.

- **Generic mobile app**, which are mobile web sites designed to match every web-enabled phone, like the [Wikipedia mobile page](#).

As you can see, you have definitely more than two options. The tricky part is that there is no best choice. It's all about context, and that context is evolving at a very fast pace. At the very least, you can count on this: If your mobile application is mainly used to display and interact with online content or services, avoid the native choice. On the other hand, if your application is mainly used offline, a native app will offer a better user experience.

eMarketer recently published a very informative chart of mobile vs. native apps' market share depending on usage:

[apps-vs-browser.jpg](#)

In any case, what you should remember is that mobile is not only about choosing between web and native apps. It requires a more sophisticated approach. Here's my advice to help you define an effective mobile strategy:

- Build an API infrastructure to allow easy and reliable access to your content and services ([APIs Drive the New Touch.Salesforce.com Platform](#))
- If you decide to use native apps, hire or train an internal team on major mobile platforms' technologies (iOS, Android) and use sub-contractors for minor mobile platforms (RIM)
- Don't try to replicate your entire web site. Rethink your offer on a local level and focus on what brings most value in a mobile context

And last but not least, bear in mind that this is a very unstable market. Everything can change within months. So don't think about delivering the most advanced mobile app. Instead, focus on acquiring understanding of your users' expectations and behavior.