The Architect's Guide to Progressive Web Apps

How PWAs can boost your app strategy by harnessing the power of the web — and what you need to know to get started today.

ionicframework.com/resources

Progressive Web Apps (PWAs) take advantage of major advances in modern web browsers, web APIs, and front-end frameworks to deliver great app experiences to mobile and desktop users.

From small mom-and-pop shops to big brands like Twitter and Pinterest, companies across all industries are demonstrating how PWAs can boost user acquisition, engagement, and ultimately, revenue.

But PWAs are not a silver bullet. They work best as part of a broader app strategy that focuses on creating a consistent, high-quality user experience across all the channels that your users want to engage with you.

This brief guide provides enterprise architects and app dev leaders with essential insights on which projects are right for PWAs, and how to build a production-ready PWA today.

CONTENTS

Why PWAs Matter

The world of applications is highly competitive and increasingly crowded. In just the last few years, the number of app store apps more than doubled, peaking at more than 2 million.

The impact of this is twofold: app fatigue for users, and immense pressure for application developers and enterprise architects to make sure their apps stand out from the crowd and provide the absolute best user experience. An app that takes more than a few seconds to load, or more than a few clicks to find and download from an app store, can cost you that user.

On the positive side, an app that delivers great content seamlessly and quickly, no matter the device, can set your company apart. This is why Progressive Web Apps (PWAs) are quickly becoming the standard way to ship these great user experiences to the web. In fact, Gartner predicts that in 2020, PWAs will replace 50% of consumer-facing apps. While there are various advantages to using PWAs, one that stands out is its ability to work offline, send push notifications, and eliminate the need to download and install from the app stores.

But what are PWAs, exactly, and how do you make them work for you? In this guide, you'll get to know PWAs, complete with statistics and real-life examples meant to answer the questions of why you need one, how they work, and how you can get started.

Gartner predicts that in 2020, PWAs will replace **50%** of consumer-facing apps.

What are PWAs?

PWAs take advantage of the latest web standards and APIs to provide the reliable, fast, and engaging experience that people expect from modern apps.

Because they're web-based, they inherently deliver all the best parts of the web, including an extremely wide reach, instant access, instant updates, and easy shareability.

Compared to traditional mobile web apps, PWAs look and feel much more like a native app — in fact, they're often indistinguishable from their native counterparts. They include offline storage and access to native features like push notifications, geolocation, and the camera — all delivered instantly through the web browser. Unlike traditional native apps though, users don't need to be routed through the app store. PWAs are also usually much smaller in size than native apps, so they load faster, work better over low-bandwidth networks, and take up less space on a user's device.

Progressive Web Apps are a new way to offer incredible mobile app experiences that are highly optimized, reliable, and accessible completely on the web. Here are some of the critical capabilities that set PWAs apart from traditional web apps.

✓ **RELIABLE**
Works offline and performs well on low-quality networks.

✓ **FAST**
Loads in seconds, with smooth interactions inside the app.

✓ **ENGAGING**
Immersive app experience with full access to native features.

✓ **DISCOVERABLE**
Easily discovered through a simple web search.

✓ **SMALL**
A fraction of the size of a traditional app store app.

✓ **FRESH**
Always up-to-date with the latest content served instantly.

✓ **INSTALLABLE**
Allows users to "keep" apps they find useful on their home screen.

✓ **ACCESSIBLE**
Runs on desktop and mobile, or anywhere you find a browser.

The facts

Don't take our word for it. According to research from Google:

3s

→ 53% of users will abandon a site if it does not load within three seconds on a mobile device

5s

→ Sites that loaded within 5 seconds earned twice as much ad revenue as slower sites

15s

→ The average mobile site takes 14 seconds to load on a 4g connection and about 19 seconds on a 3G network
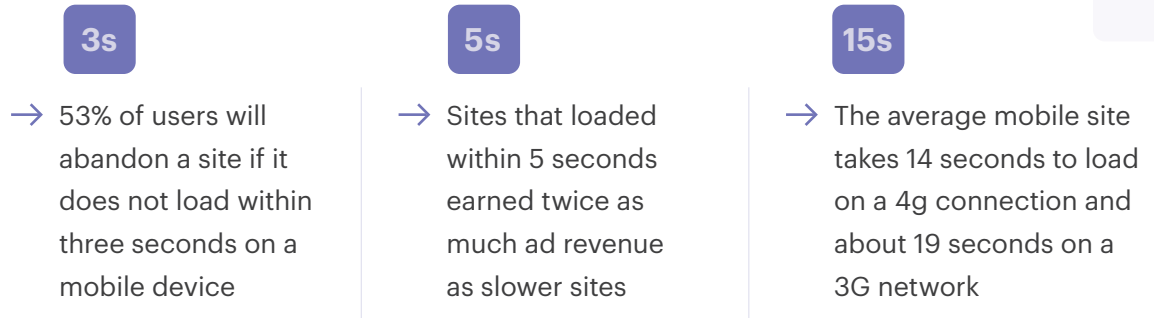
The takeaway: Those that can provide a fast, user-friendly experience don't just set themselves apart with happier users - they actually see increased revenues. This is where PWAs separate your company from the rest.

Perhaps you prefer real-life examples over research? Two companies you might have heard of implemented PWAs and saw impressive results.

Pinterest saw user engagement increase by 60%, helping to drive a 44% increase in user-generated ad revenue. These increases also represented a 2-3% bump over Pinterest's native app.

Twitter's PWA provides the exact same user experience as the native twitter app, while using less than 3% of the device storage. This combination of a better, faster, more efficient product has been a hit with users, to the tune of a 20% decrease in bounce rate and a 75% increase in tweets sent.

Although PWAs were largely pioneered by engineering-heavy companies like Twitter and Pinterest, businesses of all stripes are now achieving success after introducing PWAs. The Best Western River North Hotel saw a 300% revenue increase and 500% increase in the number of room nights booked after upgrading to a PWA. Luxury cosmetic brand Lancôme rebuilt their mobile website as a PWA, increasing conversions by 17%, and West Elm's PWA implementation earned a 15% increase in time spent and a 9% lift in revenue.

The takeaway: Whatever your industry or brand, PWAs can provide higher user engagement, lower bounce rates, and ultimately more revenue by providing a better experience for your users.

How PWAs fit into your app strategy

Let's get one thing out of the way: PWAs vs. native apps isn't a winner takes all battle. PWAs should be part of a broader app strategy focused on creating a consistent, seamless, high-quality experience across all channels, whether it's a mobile browser, a native mobile app, a desktop browser, or a native desktop app.

In general, PWAs are great for attracting new users, while natively installed apps are preferable for delivering the richest experience to loyal, repeat users. Each has its advantages and disadvantages though, so it's important to ask some questions about your brand.

Where does your user base come from?

If most of your user base already comes from the web, then a PWA is a natural fit for your product. Likewise, if you want to increase the amount of users that come from the web, then providing an upgraded web experience with a PWA can be quite persuasive.

Do you have a high bounce rate?

If you are looking to reduce bounce rates, then the instant experience that a PWA provides can help. By reducing the amount of steps it takes to get into your experience — moving from going to the app or play store, searching for your app, and installing your app, to just clicking a link — you can drastically reduce churn and bounce rate.

Do you have technical requirements only available on native platforms?

While the web continues to become more powerful every day, there are still a few features that are not yet available or available only on certain browsers. Security is the most limited today. Features that unlock access to the device, such as fingerprint authorization (like Touch ID), or biometric authentication methods, such as Android Face Unlock and Iris Unlock and iOS' Face ID are unavailable. Additionally, encrypting authentication tokens at-rest is not possible, making unauthorized access much more likely.

Perhaps the most requested feature by developers, push notifications, are not available on iOS. However, SMS can be leveraged instead, by linking directly back into the PWA experience.

Therefore, a PWA is typically not the best fit for products that have a technical requirement available only on native platforms. Even in those cases, however, it's worth considering adding a PWA to improve the user experience for new visitors, and then routing them to the app stores once they want to access more advanced functionality.

Are you building a B2E app for company employees?

For teams building business-to-employee (B2E) apps, distribution and data privacy can be a challenge, and often require a mobile device management (MDM) solution. Since PWAs are browser-based, robust application management activities like allowlists, app wrapping, and data loss prevention (DLP) that are required with a native app, don't come into play. On the distribution front, there's no need to deploy PWAs to an app store, so employees can simply download them directly from your website (or private network) when they need them. And, since information is stored away from the device at the browser level, data privacy is less of a concern (Forrester Research).

Desktop support for PWAs

PWAs have broad support across mobile and they're increasingly on desktop platforms like Chrome OS, Windows, Linux, and MacOS.

Across any of those platforms, PWAs are easily installable using your web browser of choice. In fact, it takes just two clicks: once on the plus icon in the address bar, then again on the "Install" button.

Microsoft takes it a step further, embracing PWAs as a first-class citizen on Windows 10. They can be installed directly from the Microsoft Store, show up in the start menu and apps list alongside native apps, and open in a separate window just like a normal Windows app.

Google recently started improving the PWA experience in Chrome OS. When a user visits a PWA on their Chromebook, they will be prompted to add that PWA to their device, just like on Android. Once added, that PWA will show up in the app drawer and open in a separate window.

What does all this mean for you? You can now ship one PWA and have it work just as well for users on a desktop as it does on mobile and tablets.

PWAs have broad support across mobile and they're also supported on desktop platforms like **Chrome OS, Windows, Linux, and MacOS.**

How PWAs work

PWAs utilize new, modern web APIs, most crucially Service Workers and a Web App Manifest. These two APIs have full support in Chrome (and Chrome-based browsers such as Samsung Internet), Firefox, Safari, and Microsoft Edge. These APIs make it possible to ship a full-featured app experience that hasn't been possible on the web until now.
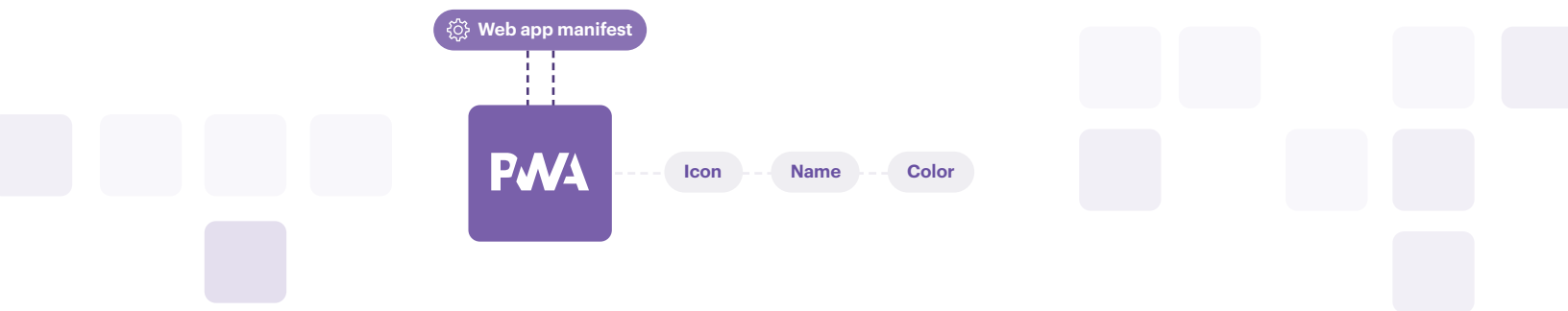
Service Workers

Service Workers are the biggest player, helping your PWA reach those reliable, fast, and engaging benchmarks. Service Workers allow your PWA to work offline — not traditionally possible on the web — and load reliably no matter the state of the user's network connection, enabling features like push notifications and background data syncing.
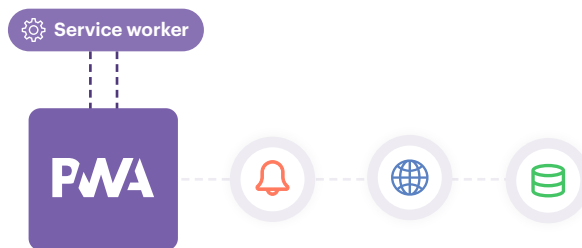
Service Workers also give you fine-grained control over caching through JavaScript APIs, letting you create a custom offline experience tailored for your app. For example, you can program your service worker to always serve cached content first, keeping your app nice and fast, but then to update the cache in the background so the next time the user opens your PWA, the content will be up to date.

Web App Manifests

Web App Manifests allow your PWA to deliver the look and feel people expect. They let you specify an icon, app name, and splash screen color. They enable users to install your PWA to their device and have it appear right alongside their native apps.

Check out these articles to get more info on Service Workers and Web App Manifests:

→ Service Workers: an Introduction

→ ServiceWorker Cookbook

→ The Web App Manifest

Try a real PWA on your device

You've read about the benefits of progressive web apps. Still not convinced? As they say, seeing is believing. Take out your mobile device, open the Camera app, then scan this QR code:

Tap to open the site link that appears. This is Ionifits - a human resources demo app that features a secure login experience, interactive employee directory, complete expense management, and more. When you're done trying the app, send the code to your team to review - it's open source, so you can check out all the implementation details.

How to get started

There are a lot of helpful resources to help you build a PWA. Here are a couple to get you started:

→   Google offers an excellent code lab that uses vanilla JS to review the key concepts of PWAs by building a simple weather PWA.

→   If you're more of a visual learner, the 2017 Chrome Dev Summit had a great talk about the basic principles to keep in mind when building your first PWA.

→   When you're ready to build a real, production-ready PWA, consider building your PWA with Ionic.

Build your PWA with Ionic

Listen, despite all their advantages, building a fast, engaging, great PWA is very challenging when you're trying to use traditional web tooling. Fortunately, you can use Ionic and our step-by-step app wizard to build your first PWA in minutes.

The wizard will ask for a few basic inputs, such as your app's name, the starter template you want to use, and which JavaScript framework you'd like to work with (you can choose between Angular, React, or Vue). Once you're done following a few simple steps, your new PWA is ready to go — all best practices are included by default.

Ionic's tools provide our recommended setup for building production ready PWAs.

So, all those features that are essential to delivering a great experience - optimizing bundle size, lazy loading, code splitting, etc. - are there to begin with and are not add-ons or improvements you have to code in later. This means that your app will only load exactly the JavaScript an individual page needs, ensuring that your PWA can achieve that all-important three-to-five second load time on the average network.

In addition to our recommended setup for production ready PWAs, you'll have access to a library of over 100 mobile-ready UI building blocks for constructing your frontend user experience.

Using Ionic for your UI components ensures that your PWA has the modern look and feel that customers expect. Our components utilize an Adaptive Styling feature that automatically transforms your UI to look and feel native to whatever platform it's running on - i.e. Material Design for Android and Cupertino for iOS - guaranteeing the user will be familiar with your UI. We also utilize the latest best practices to ensure that all your animations will be smooth and fast, even on lower end devices.

That's not all. Wouldn't it be great to deploy your PWA to the app stores? With Capacitor, our cross-platform native runtime, you can. Capacitor turns any web app into a native app, so you can run one app across iOS, Android, and the Web with the same code. Giving teams the ability to build and ship on three platforms from day one? Yes, please.

What are you waiting for?

Create a production-ready PWA with Ionic now.

About Ionic

Ionic makes it easy for professional developers and teams to build beautiful, high-performance apps for any platform or device. Ionic's open source and premium tools power more than 15% of all apps in the app stores — representing early-stage startups all the way up to major brands like GE, Target, Panera, Airbus, Mastercard, and more. At the same time, our vibrant open source community now encompasses more than 5 million developers in 200 countries.

For more information, visit ionicframework.com and follow the company on LinkedIn and Twitter.