



p1

## Web Development - Progressive Web App

WEI LUN HUANG

## Introduction of Progressive Web App

Because the early Web App is still not able to run offline, receive push notifications or links into hardware<sup>0</sup>, Google offered Chrome App service since 2013. However, by continuing operate with web standard organization, Google announced a new approach for developing web apps for Android in the 2015 Web Summit Conference. The innovative technology provides Service Worker and Web Push and advanced API to build the mutil-browser crossing application called “Progressive Web App (PWA)”. PWA will be able to use in-device hardware like push notifications, and offline work. In addition, there will be other features directly install into user desktop, such as adding an icon to the home screen, launching in full-screen with URL bar and displaying a splash screen effect that uses advanced web capabilities for delivering an app-like (act as Native App) user experience.

By looking back the evolution of web programs, “XMLHttpRequest” was available widely since IE 5 and in the Gecko-based browser (q web browser engine used in applications developed by the Mozilla Foundation<sup>1</sup>) from as early as 2000. Later, AJAX (the method of exchanging data with a server and updating parts of a web page without reloading the entire page<sup>2</sup>) happened in the next 5 years. Recently, PWA has taken over and seems to have become the mainstream application, these

principles of web development have certain rules to follow: Using URLs and links as the core organizing system, having accessibility in markup and styling to humans and search engines, providing additional user interface and system capabilities, and implementing without permission or payment in standards-based<sup>3</sup>. Brilliant tools that have attempted to gain access to allow developers to build with client-side technology for mobile web applications, such as Adobe AIR Applications, Windows Store Apps, Cordova, PhoneGap and other online platforms. Comparatively, PWA provided more technology possibility and expansibility. Functions like access system APIs, a set of tools for building software and applications, create blocks for programmers to make programming easier to develop<sup>4</sup>. Users used to download entirely packaged native apps, but now PWA provides a better experience to enhance the improvement of web.

### **Features of Progressive Web App**

Unlike Native App needs to be downloaded in Google Play or Apple Store, Progressive Web App presents an app-like web through browser. In addition, PWA is an existing app that could build with JavaScript Framework. As a result, the developers can easily implement with progressive web programming. In this case, “progressive”

means that the more features browser supports, the more progressively enhanced in data rendering could be. The performance depends client-side devices, so if the environment were not allowed, PWA would evolve from pages in browser tabs to have immersive, top-level apps, and maintaining the web's low friction at every moment in order to achieve app-like experience<sup>5</sup>.

Progressive Web Applications features take advantage to bring the best of mobile sites and native applications to users. These features included: Responsive Web Design (RWD), Service Workers, Web App Manifest, and Application Shell.

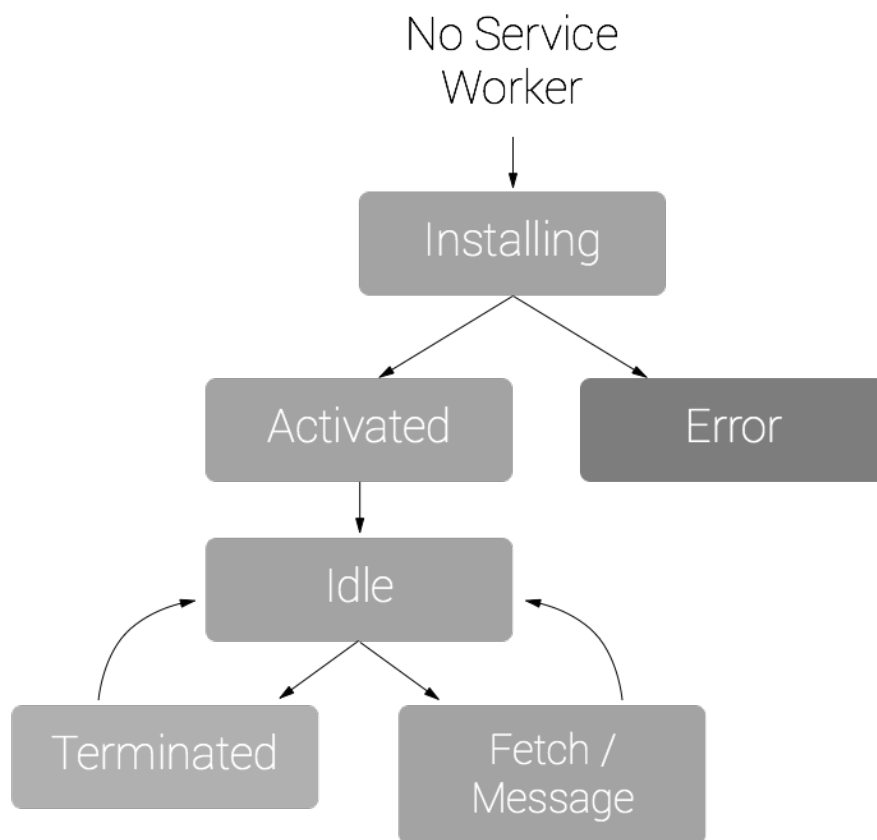
### **- Responsive Web Design (RWD)**

Progressive Web App applied the foundation of Responsive Web Design, which means the contexts are sensitive and cross-platform for building user-friendly interface<sup>6</sup>. In other words, PWA not only be an App but also be regarded as computer software. The data stores in Cloud, and the content in different devices appropriately adapt for user.

### **- Service Workers**

A service worker is a script that your browser runs in the background, separate from a

web page, opening the door to features that do not require a web page or user interaction.<sup>7</sup> Service Workers are to progressive web apps as XMLHttpRequest was to AJAX. For example, it acts as a client side proxy written in JavaScript in PWA, so you can cache (files are loaded once over the network and then saved to the local device) assets locally. No matter what rather network the user is connected to, apps load near instantly. Apps can even work offline because the service worker will respond to network requests<sup>7</sup>. Moreover, server worker can define what event has to wake it up, such as push messages can wake up a server worker.



An overly simplified version of the service worker lifecycle on its first installation.<sup>p2</sup>

## - Web App Manifest file

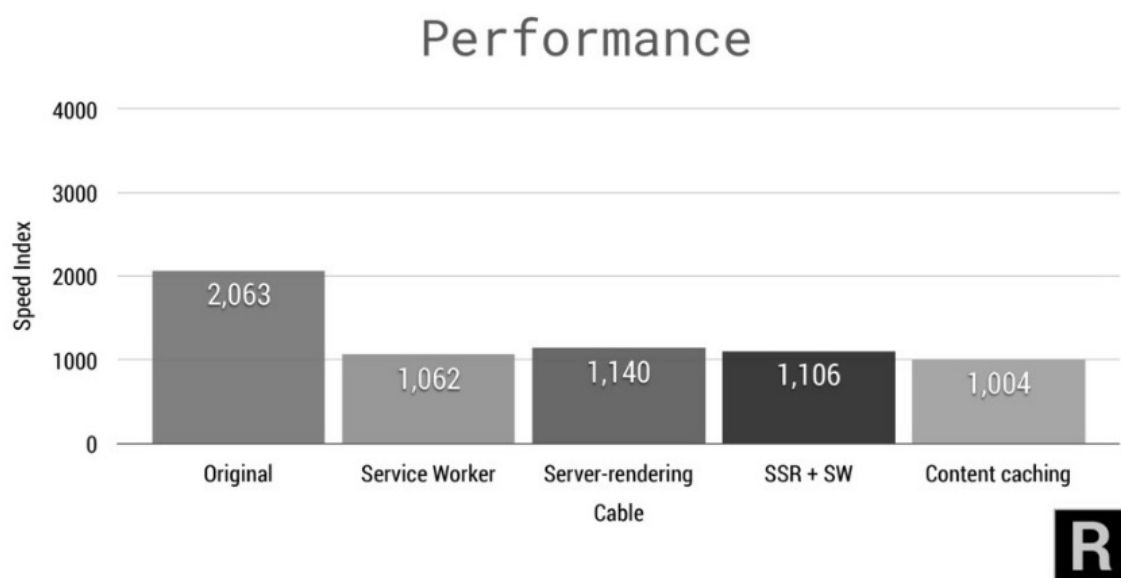
Web app manifests store metadata for a progressive web app. It provides Meta data to tell the browser how it should look when installed. Icons, description, colors, and related info convert in JSON format<sup>7</sup> that lets browsers create high-quality experiences for the launcher icon, task switcher, and splash screen. Smooth animations, scrolling and navigations keep the process silky. “Add to home screen” function in browser put the PWA on devices’ desktop help users re-engages. In order for the app install prompts to display in app must contain valid service worker registered and Web App manifest, be served over HTTPS (the secure version of HTTP, means all communications between browser and website are encrypted.<sup>8</sup>) and be visited twice, with at least 5 minutes between visits.

## - Application Shell (App Shell)

App Shell is the minimal web programming that is required to power the user interface of PWA and is one of the components that ensure reliably for good performance. In the Application Shell model, server-side rendering should be used as much as possible and client-side progressive rendering should be used as an enhancement in the same way when service worker is supported.<sup>5</sup> Its first load should

be extremely quick and immediately cached. Every consecutive time that user opens the app; the shell files are loaded from the local device's cache, which results in rapid startup times.

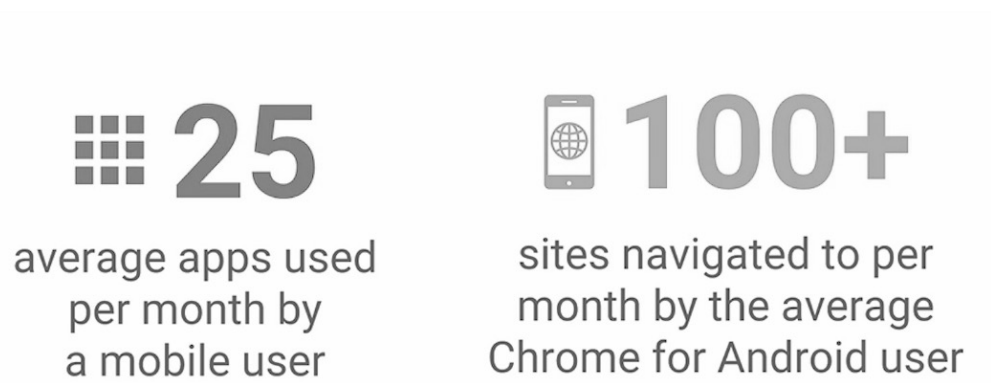
Many sites or apps assemble the page and all of its content as one before sending it down to the browser. PWA changes this model by separating the parts that change often essentially the content from the parts that do not change often. The app shell can support the content to download and display and may even store the content in a local database so that it can operate next time on open on slow network.



Speed Index in PWA Universal Render Performance.<sup>p3</sup>

## Businesses in Progressive Web App

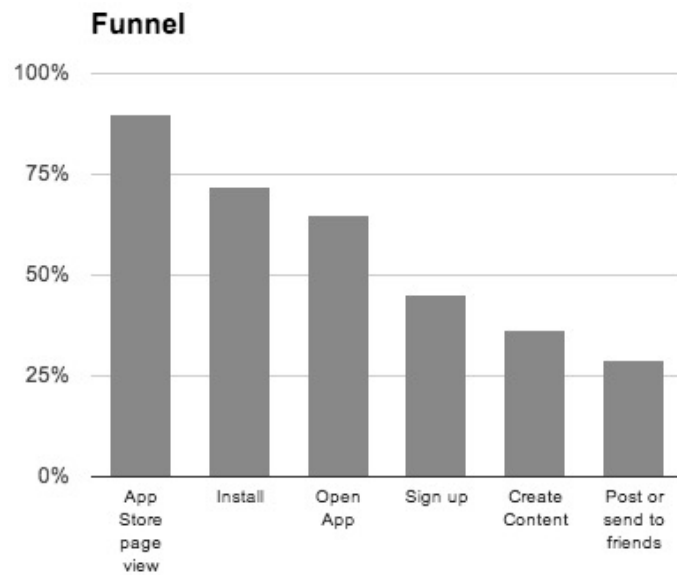
The average user visits more than 100 websites on their mobile device every month, and expectations for speed and quality higher than ever. In Contrast, the average Native Apps user only visits 25 apps per month by mobile.



Mobile App user compare with Mobile Web User<sup>P4</sup>

Progressive Web App is a web-based application, which can lightly install into user desktop, without actually searching and downloading in the app store. According the *Every Step Costs You 20% of Users* (2012), which was written by Gabor Cselle<sup>9</sup>, work as Product Manager in Google, described that every step you make a user perform before they get value out of your app will cost 20% of potential users in a consumer mobile app. As a result, letting user experienced a long process in mobile devices will be confusing and losing of customer profits.

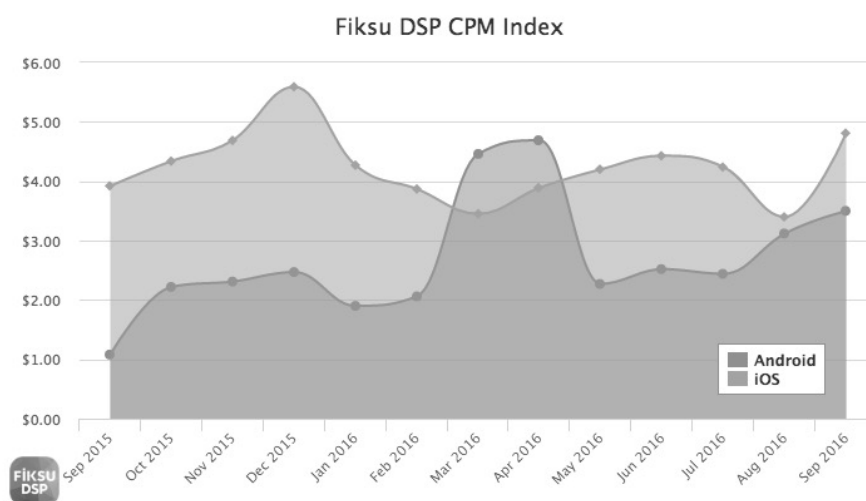




Percentage of typical consumer app funnel<sup>p5</sup>

Furthermore, the research found that it was expensive to get user to actually use their native app. Fiksu is a company that keeps tracking native application ecosystems.

Below index shows that the cost to getting a user to try out a Native App by using application ads ranges from \$1-\$2 up and down before, and that number has increased from \$2 a year ago to over \$4 today.



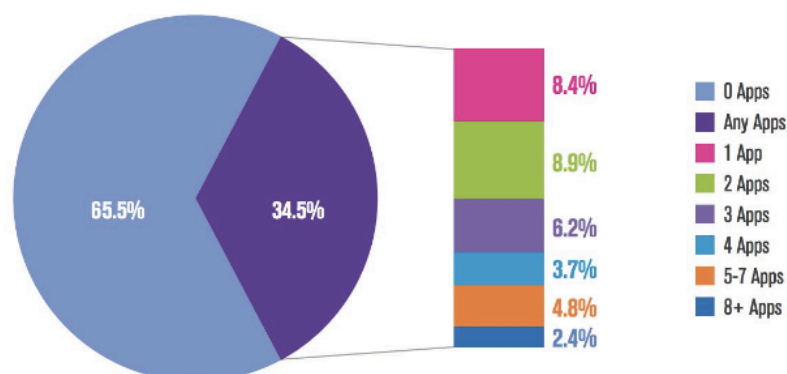
The CPM index, measuring the cost per thousand ad impressions across a wide range of exchanges and ad formats, reveals the relative competition for mobile ad impressions.<sup>p6</sup>

Flipkart, India's largest e-commerce site, decided to combine their web presence and native app into a Progressive Web Application. After that, it triples the time on its website with efficiency and has resulted in a 70% increase in conversions.

Another example is AliExpress, part of the Alibaba Group was looking for a way to provide their web users with the benefits of their app, such as the ability to work offline, performance, and re-engage users. Finally, they use Progressive Web App reach the increasing conversation by 104% than before.<sup>10</sup>

Business Insider, *Most Americans Download Zero Apps Every Month*, Steven Tweedie (2014) even wrote "65.5% of US smartphone users don't download any new apps each month." The statics showed that despite of people are using the apps they have more than ever, only approximately a third of U.S. smartphone users choose to try a new app<sup>11</sup>.

Source: comScore MobiLens, U.S., Age 18+, 3 Month Average Ending June 2014



Smartphone User's Number of App Downloads Per Month<sup>17</sup>

## Progressive Web Apps V.S Native Apps

The major advantages of Progressive Web Apps offer features that native apps lack is it can replace all the functions of native apps and websites at once. In the beginning, PWA can be reliable, load quickly and easily discover the content by search engines. On the other hand, the content-centric method native app will not show among app store search results for content cannot expose its communities widely to the app store as individual. Next, for the reason that any page or screen has a bookmark or direct link, it will be easily to save or share information in social media. Furthermore, PWA always synchronize fresh data, there is no need to go through the app stores to push updates anymore. Moreover, PWA has universal access (based on web), but app stores have some restrictions and policies in control. Finally, PWA can be extremely important in large data saving like emerging markets with slow Internet access. For example, e-commerce website Konga cut data usage by 92% for the first load by migrating to a PWA.<sup>12</sup>

However, the disadvantage of PWA to Native App is it lacks of privacy permission, such as user contacts, calendar, alarms and other features. User cannot get a phone call or use device's flashlight. It does not allow accessing system like task management, or even

modifying system settings. In the website, *Is service worker ready*<sup>13</sup>, shows that not all the browsers can support service worker running. In addition, in another website, *What Web Can Do Today*<sup>14</sup>, has several postings that list features still not available for web. Moreover, PWA is not feasible for Safari to have a pop-up “add to home screen” function. I would say that the gap between Web and App still exist.

### **Progressive Web App Final Thought**

For the new PWAs’ developing strategy, Google announced that they decided to terminate Chrome app in two years after due to Windows, Mac and Linux only 1% user frequently use it. The PWA can be forecasting in future web fiery-trend. Although PWA used to leave some problems for compatibility in iOS platform, the prompt to install the web app is currently shown under varying conditions across Opera, Chrome and the Samsung browser. Apple has indicated interest in progressive web apps for iOS, but at the time of writing, it still relies on Meta tags for web app configuration and the application cache for offline use. By paying attention at new releasing products, Apple seems to integrate app into their own system. On the contrary, Google will continue promoting Progressive Web App to develop the web more “App-like”.

## Reference

<sup>0</sup> Rahul Roy-Chowdhury, Product Mangers from Google Development team

<sup>1</sup> Wikipedia, List of web browsers, Gecko-based browser definition,

[https://en.wikipedia.org/wiki/List\\_of\\_web\\_browsers](https://en.wikipedia.org/wiki/List_of_web_browsers)

<sup>2</sup> Segue Technologies (2013), What is Ajax, <http://www.seguetech.com/ajax-technology/>

<sup>3</sup> Infrequently Note (2015), Progressive Web Apps: Escaping Tabs Without Losing Our Soul (2015),

<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>

<sup>4</sup> MuleSoft Videos (2015), What is an API?, <https://www.youtube.com/watch?v=s7wmiS2mSXY>

<sup>5</sup> Addy Osmani, Getting started with Progressive Web Apps (2015),

<https://addyosmani.com/blog/getting-started-with-progressive-web-apps/>

<sup>6</sup> Jay Bryant and Mike Jones, Pro HTML5 Performance (2012 Published), Chapter 4 - Responsive

Web Design

<sup>7</sup> W3Cschool.com, JSON – Introduction, [http://www.w3schools.com/js/js\\_json\\_intro.asp](http://www.w3schools.com/js/js_json_intro.asp)

<sup>8</sup> Instant SSL by COMODO, What is HTTPS, HTTP VS HTTPS,

<https://www.instantssl.com/ssl-certificate-products/https.html>

<sup>9</sup> Gabor Cselle (2012), Every Step Costs You 20% of Users,

<http://blog.gaborcselle.com/2012/10/every-step-costs-you-20-of-users.html>

<sup>10</sup> Google developers Website, <https://developers.google.com/web>

<sup>11</sup> Steven Tweedie (2014), Business Insider, Most Americans Download Zero Apps Every Month,

<http://www.businessinsider.com/how-many-apps-people-download-per-month-2014-8>

<sup>12</sup> Google developers, Case Study, <https://developers.google.com/web/showcase/2016/konga>

<sup>13</sup> Is Service Worker Ready?, <https://jakearchibald.github.io/isserviceworkerready/>

<sup>14</sup> What Web Can Do Today?, <https://whatwebcando.today>

<sup>p1</sup> Max (2016), IONIC, What are Progressive Web Apps?,

<http://blog.ionic.io/what-is-a-progressive-web-app/>

<sup>p2</sup> Matt Gaunt, Service Workers: an Introduction,

<https://developers.google.com/web/fundamentals/getting-started/primers/service-workers>

<sup>p3</sup> AddyOsmani, Speak Deck, Progressive Web Apps Across All Frameworks,

<https://speakerdeck.com/addyosmani/progressive-web-apps-across-all-frameworks>

<sup>p4</sup> Progressive Web Apps - Chrome Dev Summit (2015) Video,

<https://www.youtube.com/watch?v=MyQ8mtR9WxI>

<sup>p5</sup> Gabor Cselle (2012), Every Step Costs You 20% of Users,

<http://blog.gaborcselle.com/2012/10/every-step-costs-you-20-of-users.html>

<sup>p6</sup> Fiksu, CPM Index: September 2016, <https://fiksu.com/fiksu-dsp-indexes/#analysis>

<sup>p7</sup> ComScore's U.S. Mobile App Report (2014), Insights,

[https://www.comscore.com/Insights/Press-Releases/2014/8/comScore-s-US-Mobile-App-Report-Available-for-Download?\\_ga=1.27795452.1596473589.1407244908](https://www.comscore.com/Insights/Press-Releases/2014/8/comScore-s-US-Mobile-App-Report-Available-for-Download?_ga=1.27795452.1596473589.1407244908)