Go for Progressive Web Apps.

Get a Better, Low Cost, Mobile Presence

- A Whitepaper

Written by:
Mrityunjay Kumar
Technology Architect
Magic Software Inc.
www.magicsw.com

Typically mobile web apps really suck!! They are slow and sluggishand thesituation could get even worse with shaky internet connectivity. Hence, businesses and app developers are more inclined towards building native apps (including hybrid apps).

Let’s not forget that native applications have their own set of problems:

- Not always easy to develop and support; especially with regards to extra time to develop and support different platforms, versions and devices.
- Not a cheaper proposition, as support of multiple platforms, devices and versions often increases the cost.
- Not easy to distribute; since publishing to an app store is anongoing, tedious process.

There are some features of native or hybrid apps that if added to mobile web apps would truly be the icing on the cake creating unified apps that could be easily distributed and would work on all devices.

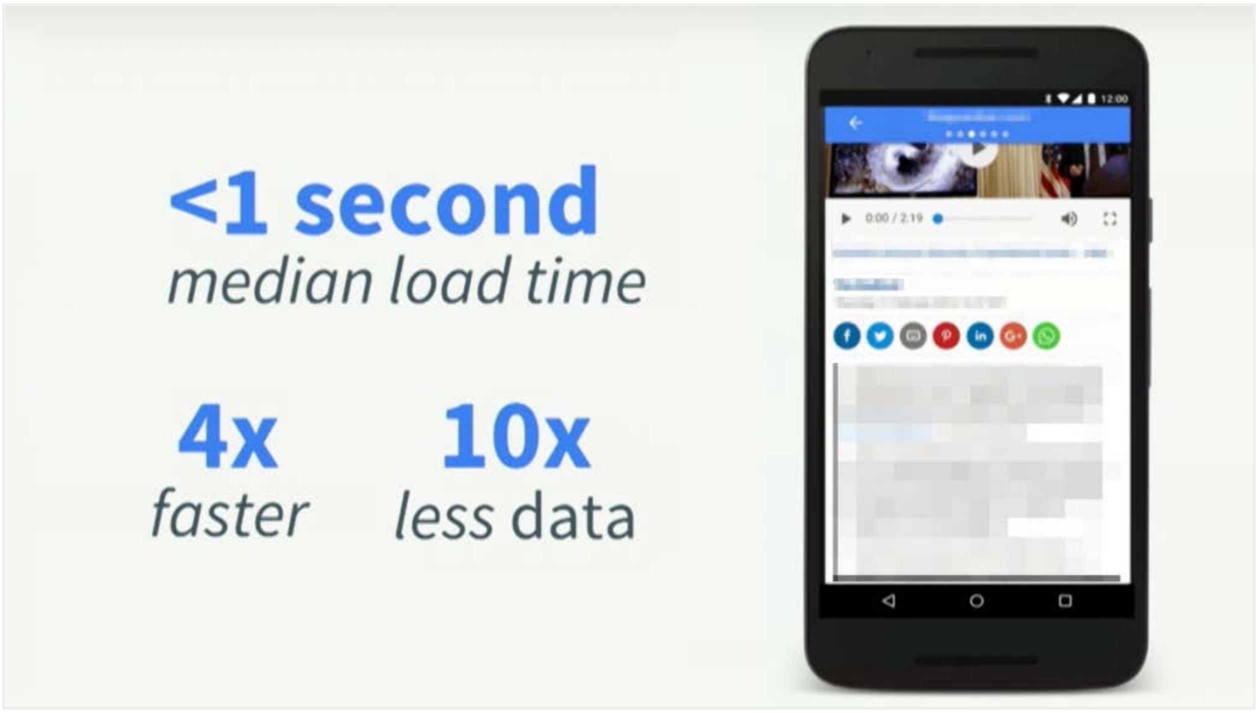
- Home screen app like icons.
- Push notification in system tray.
- Full screen immersive experience.
- One-tap sign in using device accounts.
- Secure web pay details management.
- Secure on HTTPS.
- Faster rendering.
- Work offline on flaky network.
- Background task processing.
- Access to native device capabilities like camera, microphone, accelerometer and others.

This is where**Progressive Web Apps**step in. Based on design principles and standardized browser technologies they make it possible to achieve almost all of the above mentioned features.

Progressive Web App Deep Dive

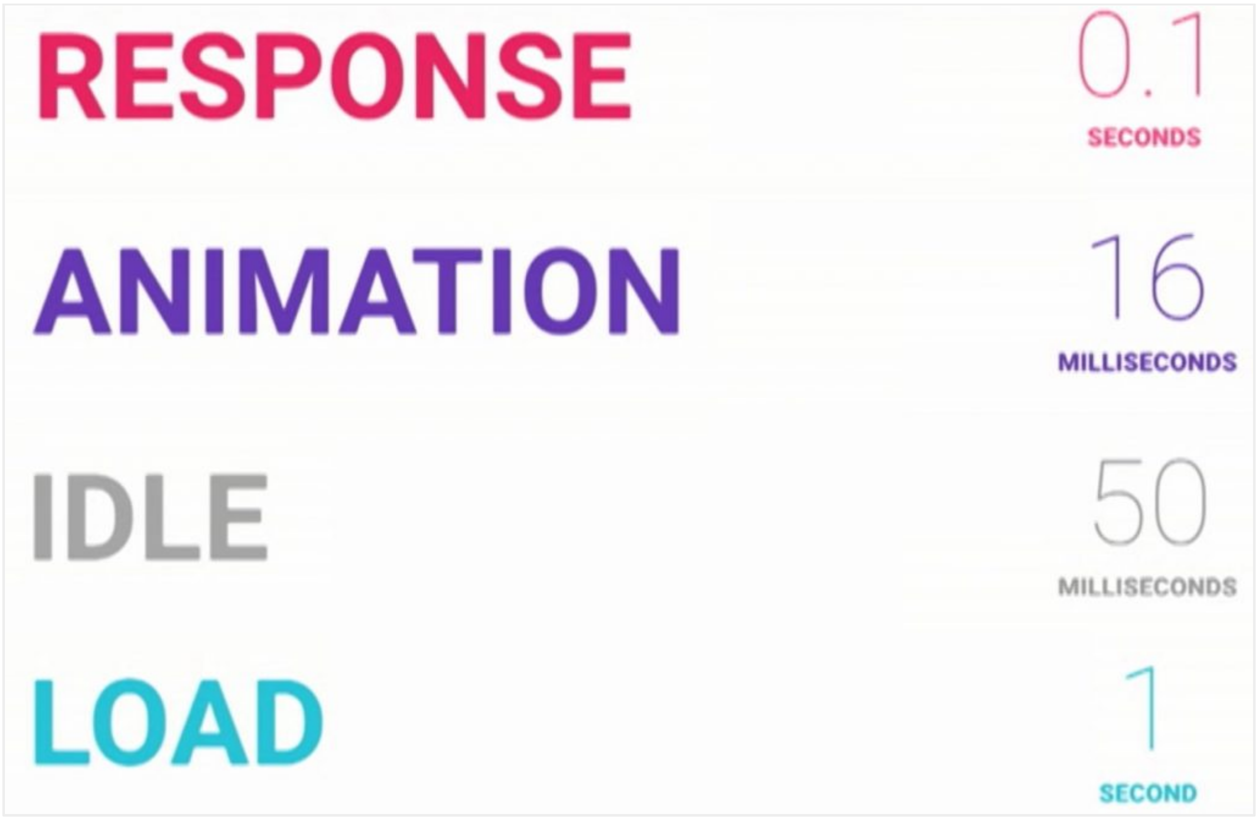
A progressive web app typically will be a single page application with responsive design. There are standards and design patterns to get a better and faster user experience on the mobile web. If built correctly,web apps can perform at par with native applications.

A progressive web app performance will always be better than that of a typical web app.



If a web app were to just look like a native one, without the responsiveness to devices the entire purpose would be defeated. Progressive web apps also provide guidelines to build a UI which animatesor respondsat 60 fps (frame per seconds).

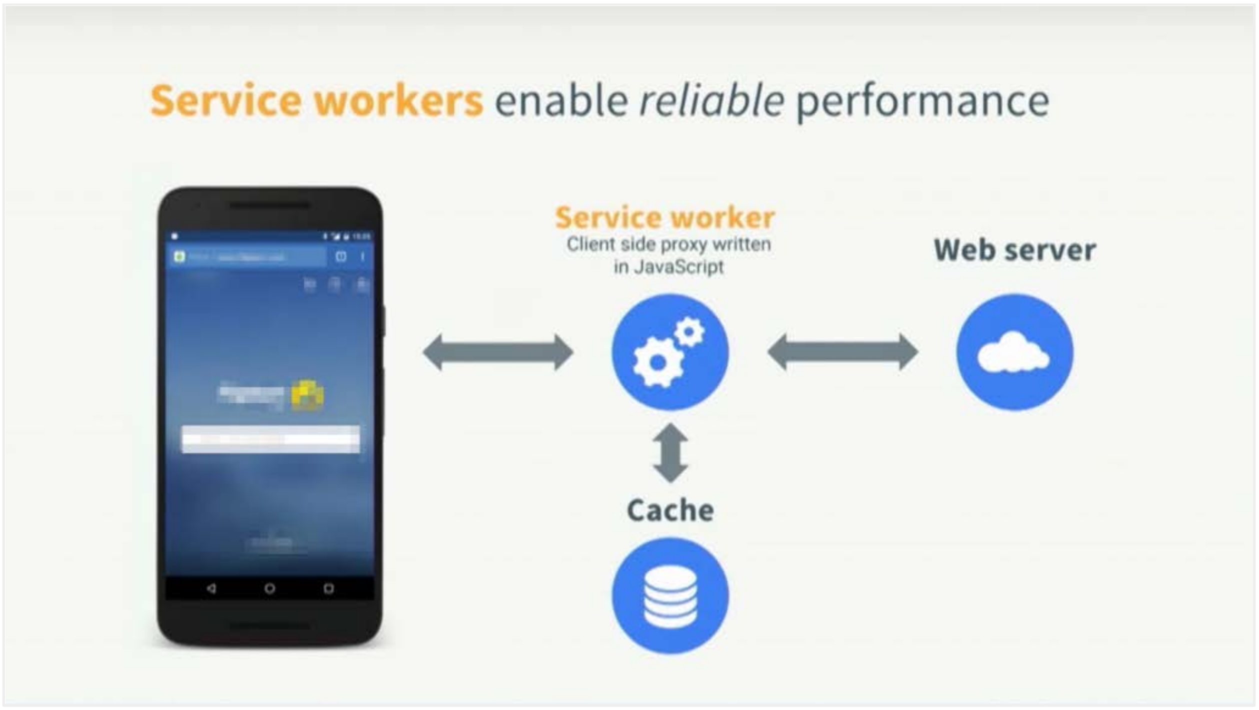
To delight users, mobile web apps need to behave,in terms of performance and interface, ideally in these time limits.



Now we have several reasons to deep dive into a few standards and design patterns, which collectively transform a normal web app into a progressive web app.

Service Worker

A service worker is a standard that comprises scriptsrun by one’s browser in the background, separate from a web page, opening the door to features, which donot need a web page or user interaction. Today, they already include features like **push notifications** and **background sync**. In the future, service worker will support things like periodic sync or geo-fencing, too.


Service workers enable reliable performance
Service worker
Client side proxy written in JavaScript
Web server
Cache

Few salient features of ServiceWorker:

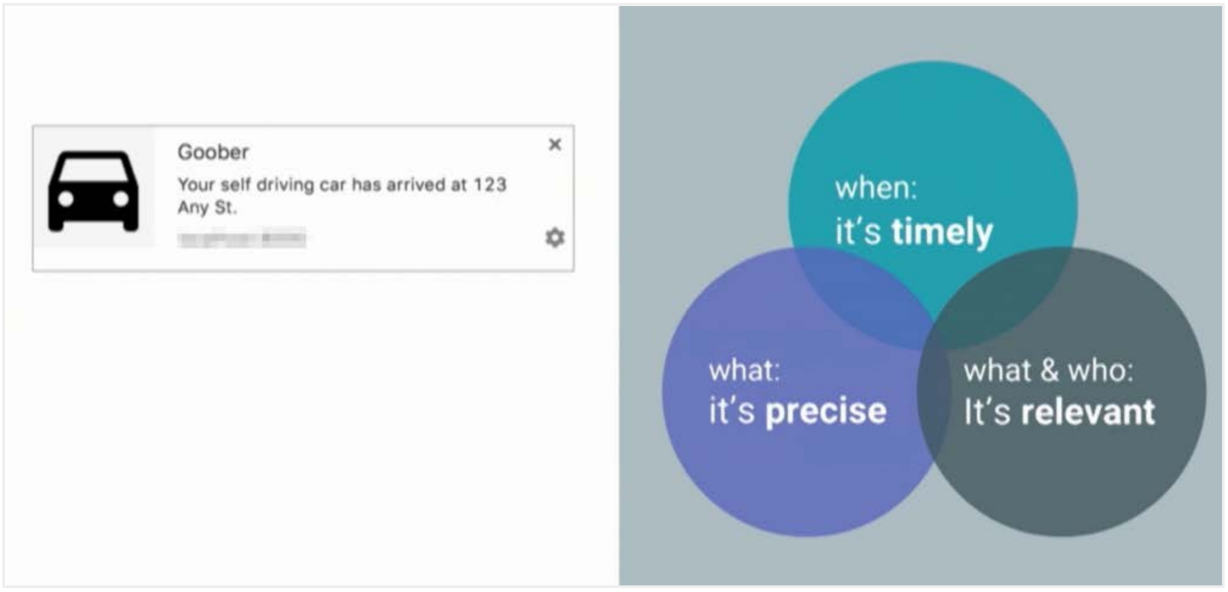
- Unlike aJavaScript Worker, which cannot access the DOM directly, a service worker can communicate with the pages it controls by responding to messages sent via the postMessage interface, and those pages can manipulate the DOM if needed.
- Service Worker is a programmable network proxy, allowing control of network requests from the page handled.
- It terminates when not in use, and restarts when next needed. Hence, we cannot rely on global state within a Service Worker's `onfetch` and `onmessage` handlers. If there is information that you need to persist and reuse across restarts, Service Worker does have access to the IndexedDB API.
- Service Worker makes extensive use of Promises.To learn more about Promises – go toJake Archibald's article.

Google Chrome, Opera, Firefox and IE Edge (In development) have support for Service Worker. Apple Safari hasa plan for the same.

Service Worker specification is managed by W3C.

Push and Notifications

Push and notifications are important ways to engage users outside the context of web page. They are some of the coolest features of a native app. When we send a timely, precise and relevant notification to users, it increases the value of the offering.



Service Worker can enable push and notification seamlessly for web. Push is information sent by the server to ServiceWorker. Notification is a message sent to the user by ServiceWorker. Just like in native apps you need GCM (Google Cloud Messaging) account for sending web push notifications.

W3C is responsible for creating specifications for bothPush API and the Notifications API.

Web App Manifest

The Web App Manifest specificationis anotherstandard managed by W3C. It is basically a JSON file which helps in providing configurations to enable several native like features in a web app.

- Add to home screen - User can now add a web app to their home screen for easy access. Home screen will display an icon of the web app. We can specify icons set (with different sizes) for add to home screen functionality in the web app manifest file. In Mobile Chrome,as with some other browsers, you get an option for it in the menu.
- Splash screen –A splash screen helps in providing a great user experience when the app is not yet ready to display the first chunk of information. In the absence of a splash screen, the user gets confused about the state of the app. In web manifest we can give the splash screen image, title and theme colorconfigurations. When user opens the web app they will see a splash screen just like in a native app.
- Theme color - We can specify a theme color to addcolor to the toolbar, address bar and other features.
- Launch style - We can specify display and orientation for the web app. Display options are browser or standalone while orientation is portrait or landscape.

To add the Web Manifest to the web page we need to use this code `<link rel="manifest" href="/manifest.json">`.

App Install Banner

Mobile Chrome provides an easy way for the web page to prompt a user to install the web app to their home page. This is in addition to the ‘add to home screen’ feature. It helps in increasing accessibility to the app.

Native Hardware Access

We can program the web to get access to native hardware capabilities. Currently available are:

1. Click to call - Easily link phone numbers for direct dial
2. User location
3. Device orientation

4. Audio/Video recording and playback (Playback is already available in several mobile browsers)

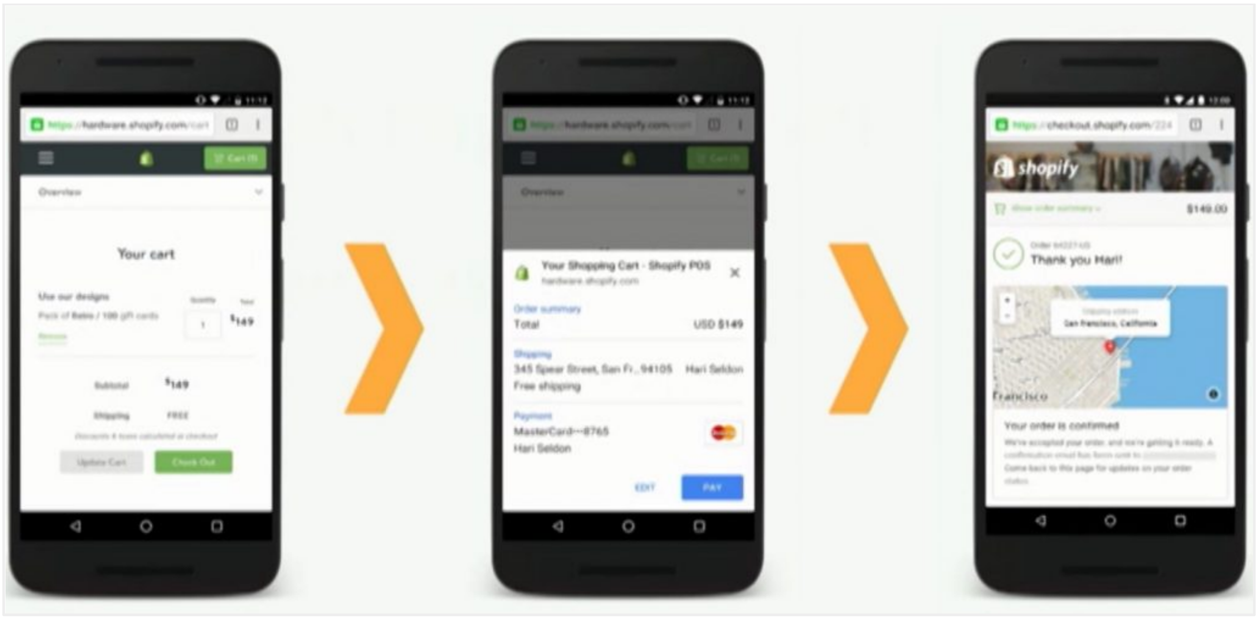
Some capabilities are work in progress:

1. Generic sensor API
2. Ambient light sensor
3. Proximity sensor
4. Accelerometer Sensor
5. Gyroscope Sensor
6. Magnetometer Sensor
7. Vibration
8. Battery status
9. Wake lock

These native hardware capability access standards are again managed by W3C.

Web Payment API

Web payment standards developed by W3C group called Web Payment Working Group (WPWG) improve the current online payment mechanism. This is important to cut average shopping cart abandonment rate, which stays at 69% across all devices. It is even higher for mobile devices.

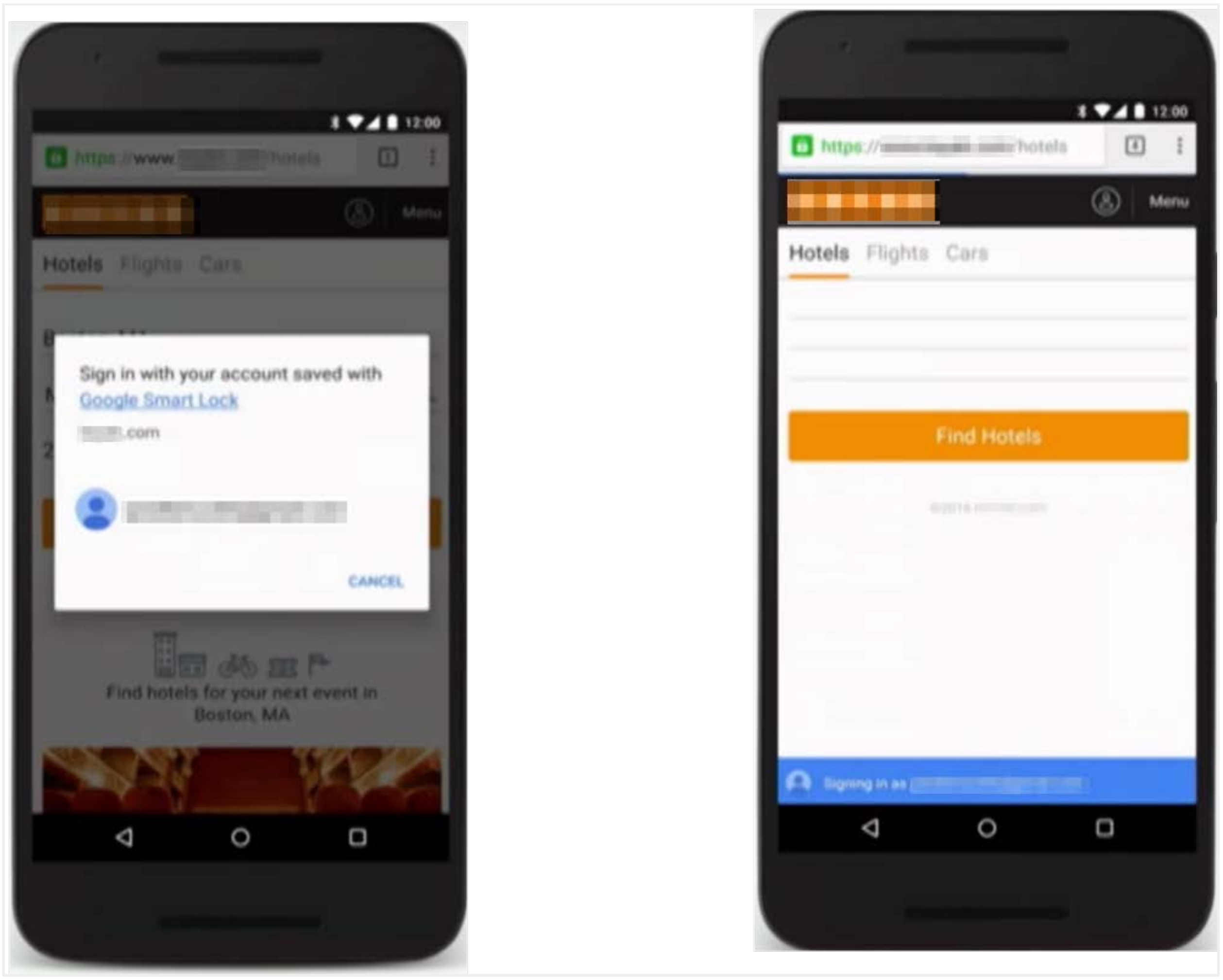


Web Payment API will cut the steps and increase the reliability for online payment. A single tap makes it the best way for a user to complete their transaction.

Another forum called Web Payments Interest Group (WPIG), which consists of finance, mobile and web industry entities, contributes actively in shaping up the Web Payment standards.

Credential API

Credential API aims to reduce the form filling for login information by a user. It enables the web to use device accounts for one tap and auto log-in and further improves the user experience.



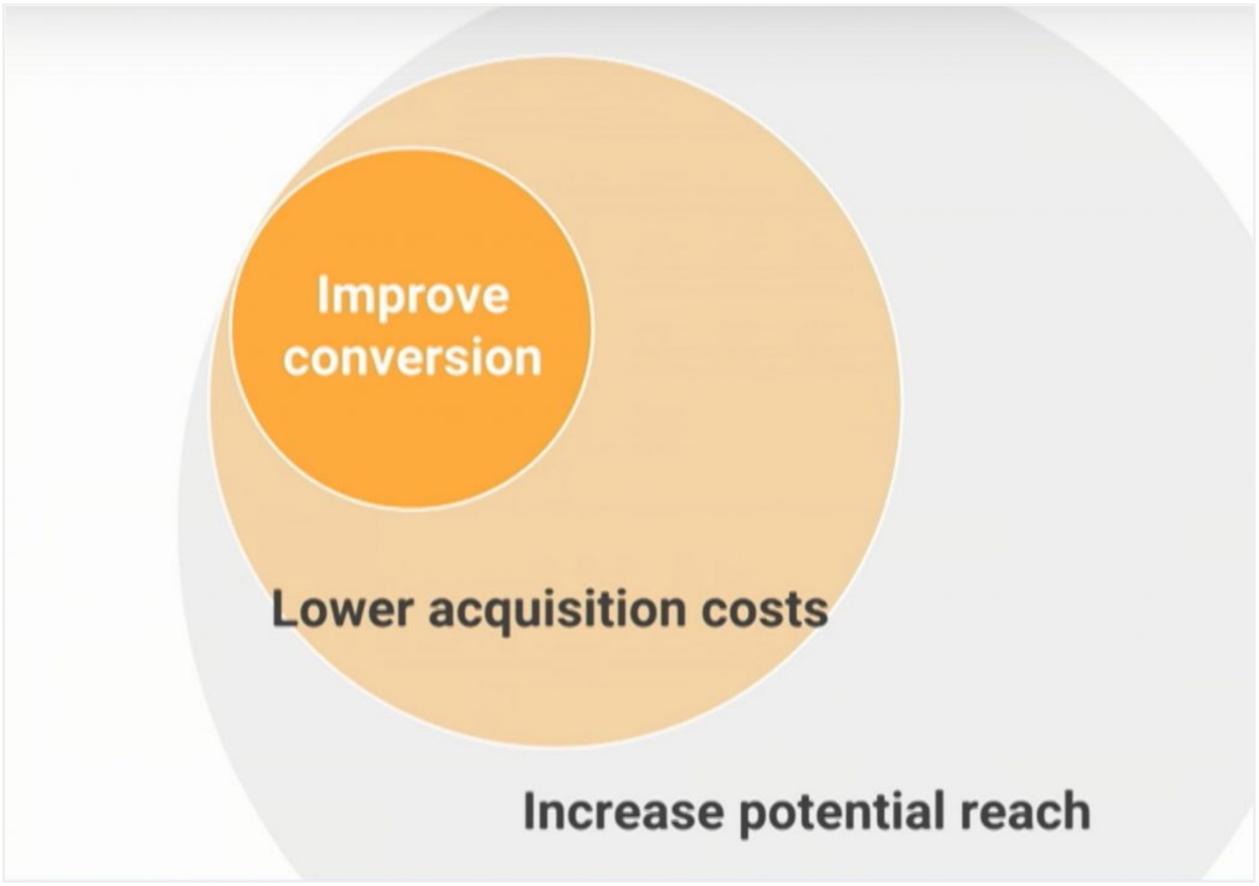
Sign in with your account saved with Google Smart Lock

Find hotels for your next event in Boston, MA

Investing in Progressive Web Apps – the Business Logic

Investment in progressive web app is critical for an organization’s mobile strategy to succeed in the long run. The Web is device independent and it gives the opportunity to explore an app without installing anything. Progressive web apps will reduce the feature gap between native apps and web appsin a short while.

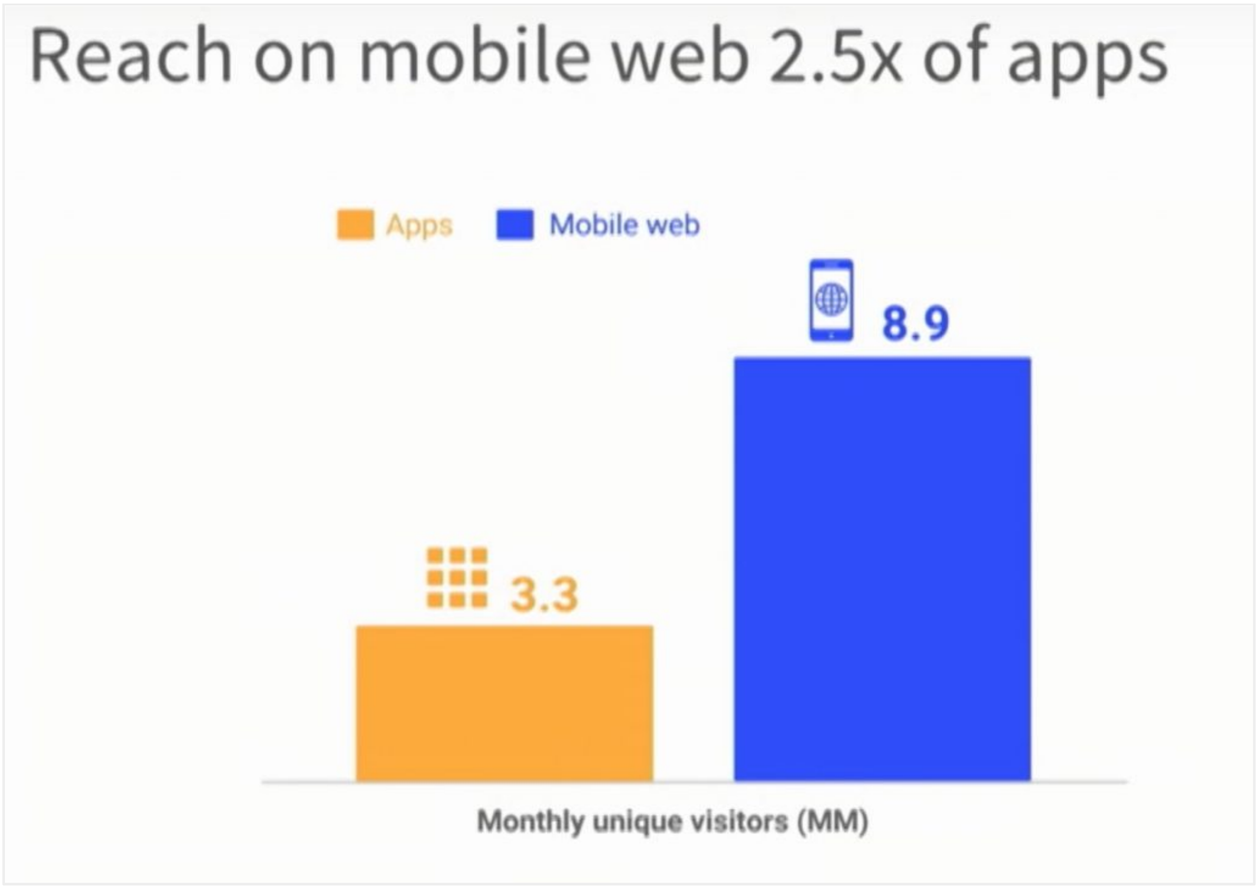
There are a few important matrices which motivate us to start investing in progressive web apps.



Increased Reach

Google Chrome has announced about 1 billion active monthly mobile users in just 4 years of its first release. Expectedly there will be even more tremendous rapid growth on the mobile web.

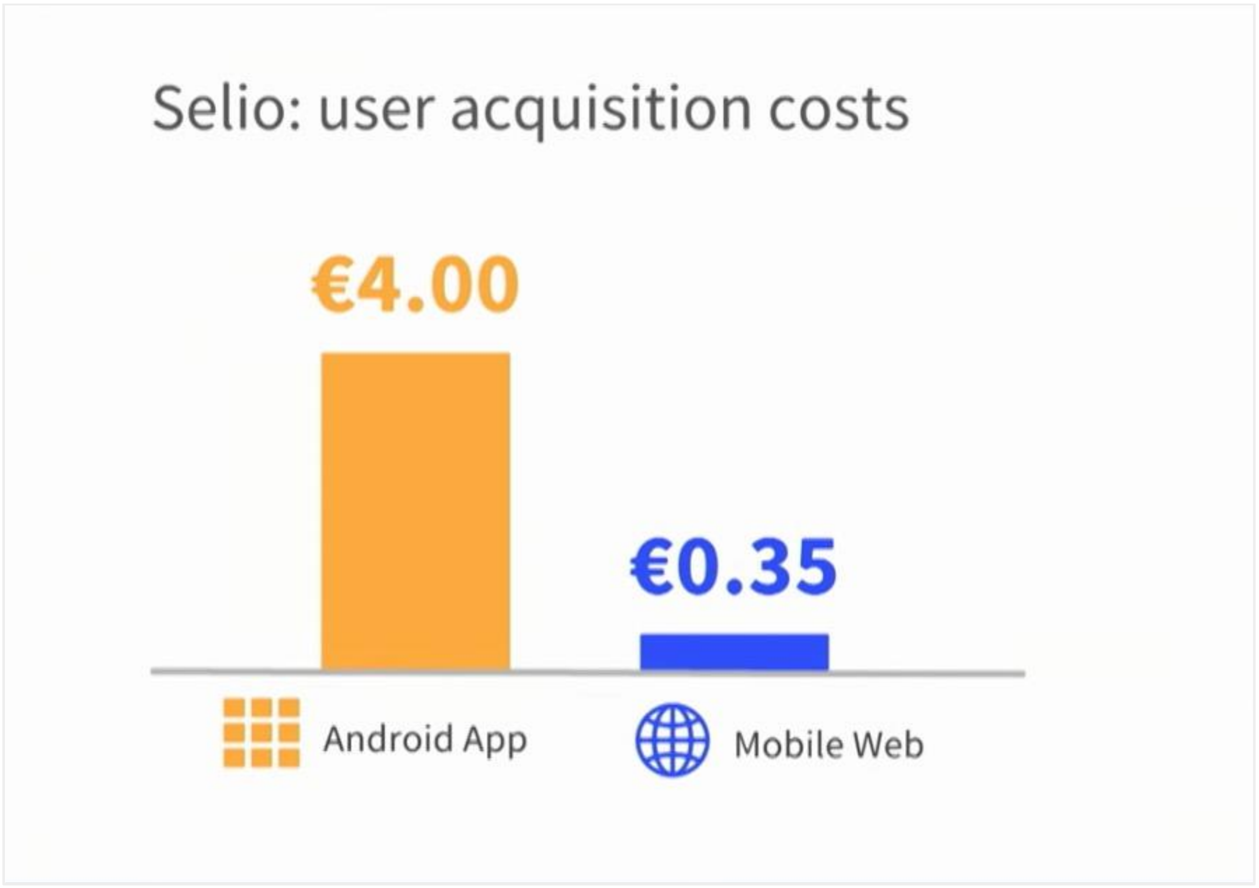
ComScore recently released data which says users are spending 2.5 x times on mobile web apps vs the top 1000 native/hybrid apps.



Lower Acquisition Costs

Google worked with several companies like Flipkart and AliExpressfor their progressive web app strategy and helped them build a better mobile web presence.

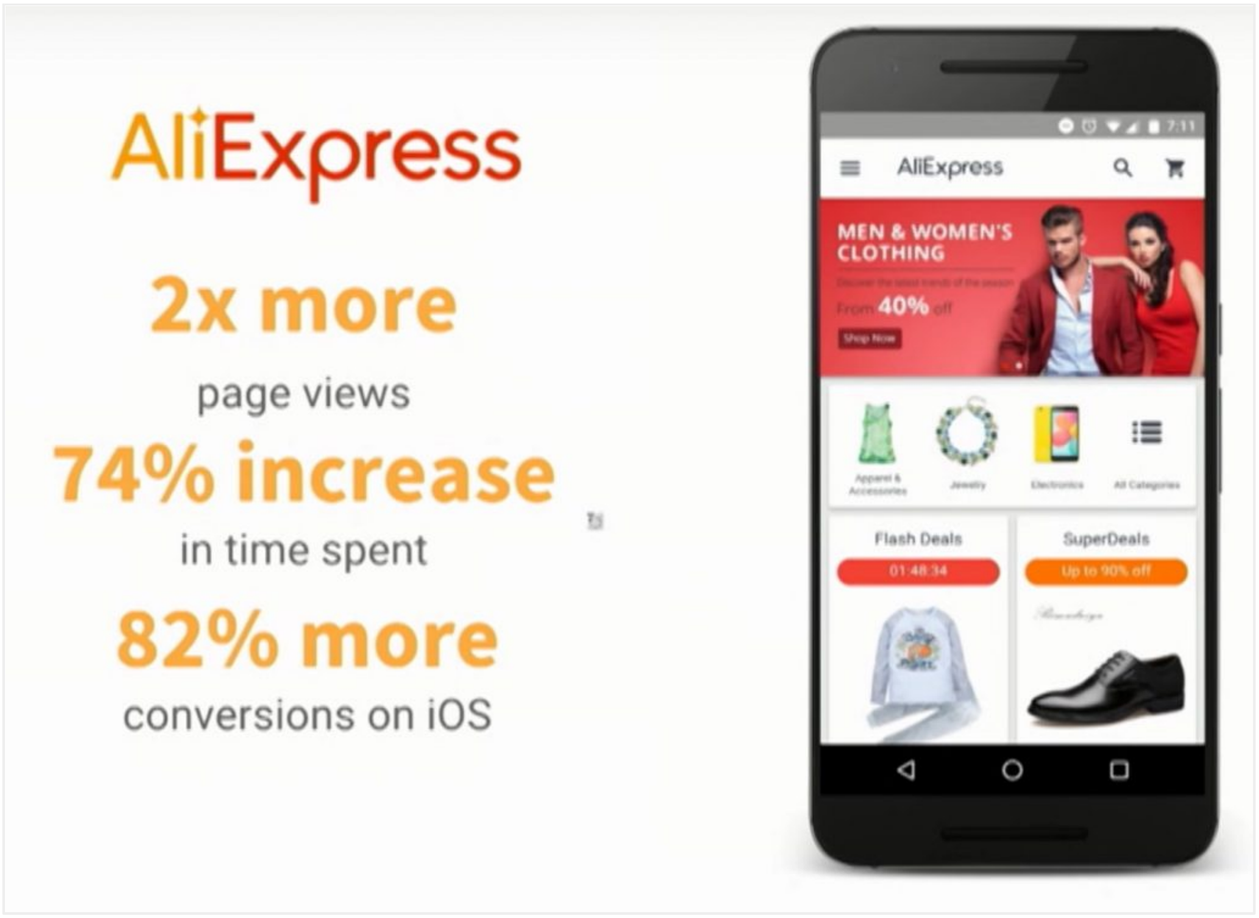
One of the key matrixes for any marketing team is the average customer acquisition cost. Seliodeclared that their average customer acquisition cost for progressive web apps is a whopping 10 times lesser than for native apps.



Improved Conversion

Conversion is another key matrix which explains the actual transactions made by acquired customers.

AliExpress revamped their mobile web presence with progressive web app. Very soon, they saw a huge 82 % more conversion on iOS devicesfollowing an improvement in their mobile web presence They have also experienced a 2X increase in pageviews and 74% more time spent on mobile web.. Hence it is critical for businesses to continue investing in progressive mobile web apps.



Conclusion

Progressive web app may look like a buzz word around improved web standards and few mobile friendly web design patterns. But we can't deny the positive impact it can bring on businesses and users.

Progressive web app bring some features and improvement for mobile web which was only available for native app in past. Lower cost of development and maintenance as well as ease of distribution will taint on the position of native app in near future.

Native app will not go away but it's significance will take a toll as progressive web app will progress.

We can start leveraging progressive web app standards and design patterns in any domain/verticals mobile web offering.Magic Software can assist organizations with their progressive mobile web app strategy.

References

1. Google progressive web app developer home page
2. Progressive Web App Dev Summit videos playlist
3. JavaScript Service Worker article
4. W3C ServiceWorker standard
5. W3C Push API specification
6. W3C Notification API specification
7. W3C web app manifest specification
8. Indexdb API specification
9. Promise design pattern
10. W3C native hardware capability access standards
11. W3C web payment API specification
12. W3C credential API specification

Copyright attribution