



+



BUILD FOR MOBILE

# Progressive Web Apps

Christina Kayastha  
Senior Software Engineer  
Vistaprint, Cimpres  
@christikaes

# HELLO WORLD!

My name is Christina (:

Christina Kayastha  
Senior Software Engineer  
Vistaprint, Boston, MA  
@christikaes

I'm all about:

- Icecream
- Community Events
- Bleeding Edge Technology



1

USER INTERFACE



2

EXPERIENCE DESIGN



3

VIRTUAL/AUGMENTED REALITY



4

BUILD FOR MOBILE



5

MANAGE STATE



6

AUTH & DATABASE





+



BUILD FOR MOBILE

# Progressive Web Apps

Christina Kayastha  
Senior Software Engineer  
Vistaprint, Cimpres  
@christikaes



# Progressive Web App

WHAT is it ?

HOW can I use it ?

WHY should I care ?



Why should I care  
about MOBILE ?

# 1992

# 2000

# 2007



First  
'SmartPhone'



Smartphone  
becomes  
commercial

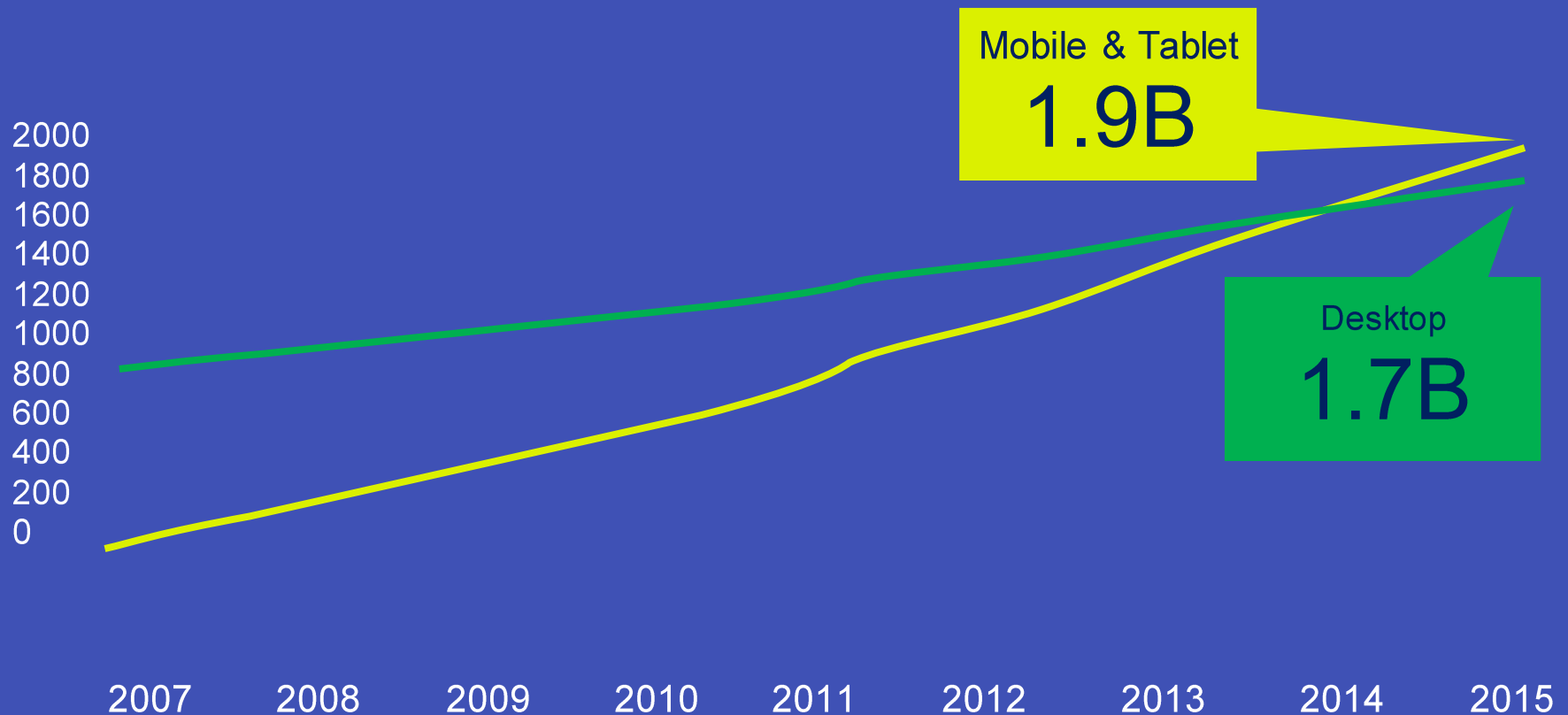


First  
Touchscreen  
'SmartPhone'

# 2014

## Number of global Mobile users surpasses Desktop

### Number of global users (Millions)

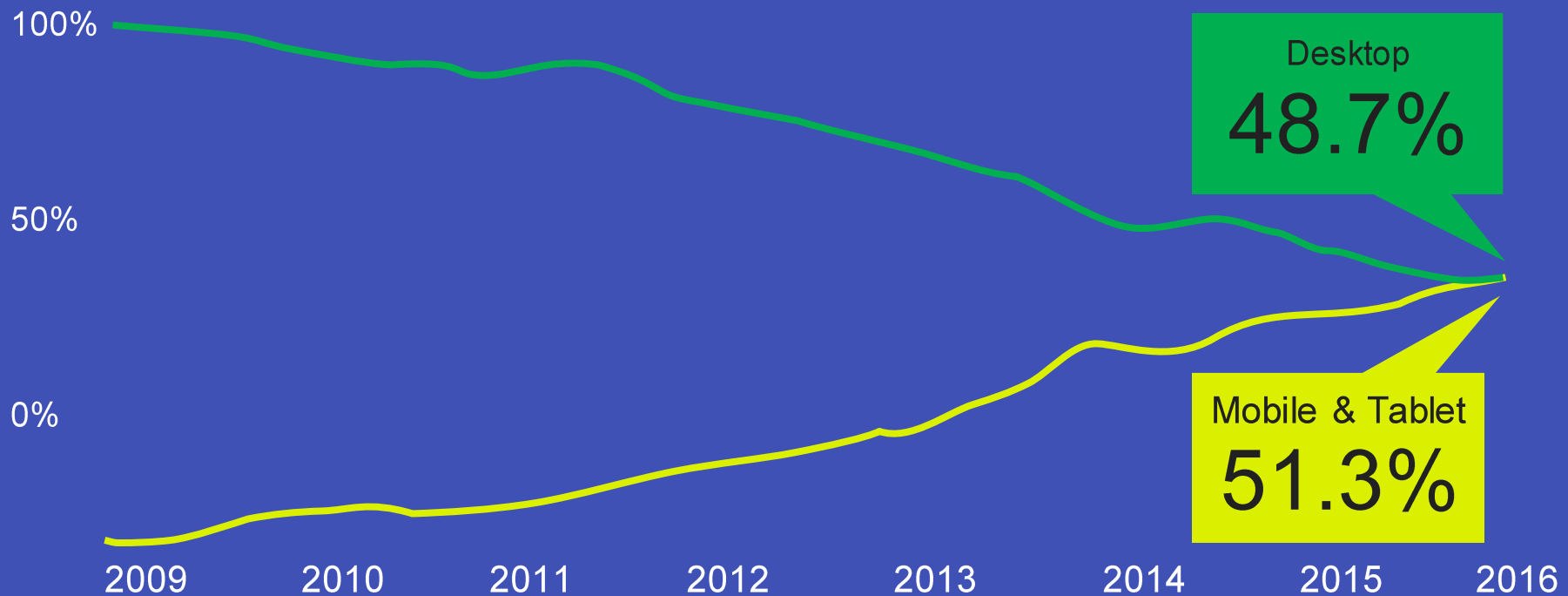




# 2016

## Internet usage on Mobile surpasses Desktop

### Internet Usage world wide



# 2017

We check our phones 74 times a day!



# 2020



Mobile traffic data will increase 8 fold between 2015-2020



5.4B

MOBILE PHONE

5.3B

ELECTRICITY

3.5B

WATER



I DO care  
about MOBILE !



Mobile internet usage  
has already outgrown  
Desktop—the mobile web  
continues to become more  
and more powerful!

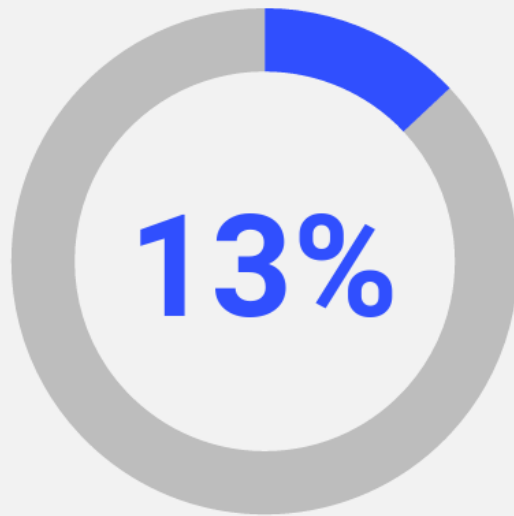


What does  
PROGRESSIVE  
WEB APP  
mean anyway?

# Progressive?

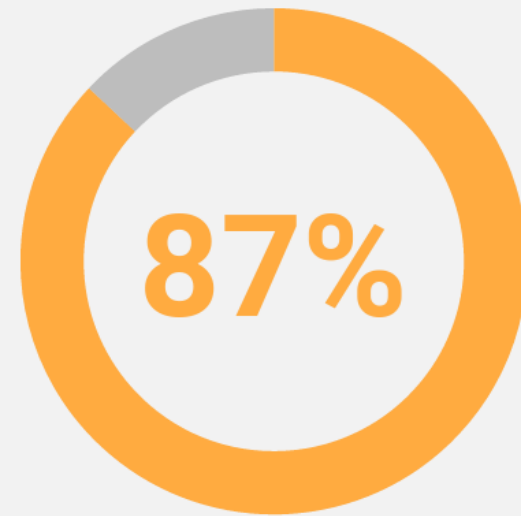
Happening or developing gradually or in stages; proceeding step by step.

(of a group, person, or idea) favoring or implementing social reform or new, liberal ideas.



Mobile web

VS.



Apps

Source: comScore Mobile Metrix, U.S., Age 18+, June 2015



# ?%

Of user's time  
spent in top 3 apps

# 80%

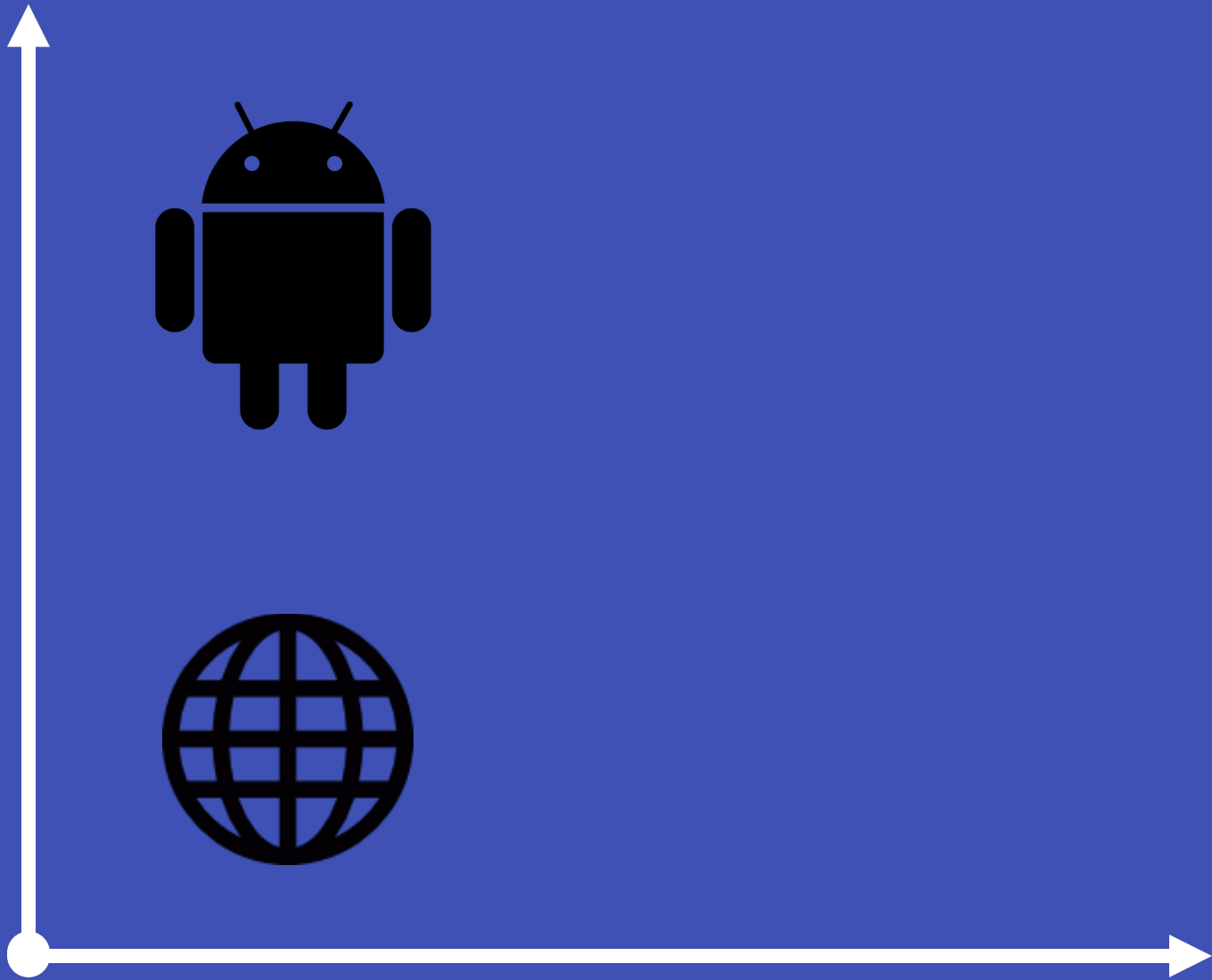
Of user's time  
spent in top 3 apps



New Applications  
Installed per Month

# 0

## New Applications Installed per Month



Reach

# Progressive Web Apps

Are a whole new way of caring  
about your User Experience

# User experiences need to be:

Reliable

Fast

Engaging

# User experiences need to be:

Reliable

Fast

Engaging





# Ajax: A New Approach to Web Applications

February 18, 2005 | By Jesse James Garrett

If anything about current interaction design can be called “glamorous,” it’s creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn’t on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.

Despite this, Web interaction designers can’t help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web’s rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.

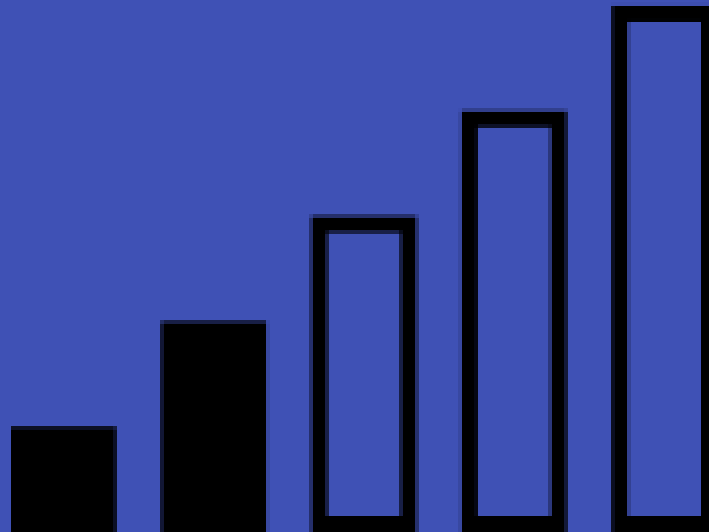
That gap is closing. Take a look at [Google Suggest](#). Watch the way the suggested terms update as you type, almost instantly. Now look at [Google Maps](#). Zoom in. Use your cursor to grab the map and scroll around a bit. Again, everything happens almost instantly, with no waiting for pages to reload.







# Lie-fi



# Reliability means never showing the Downasaur



# User experiences need to be:

Reliable

Fast

Engaging

# ?%

Users abandon sites  
that take more than  
3 seconds to load



# 53%

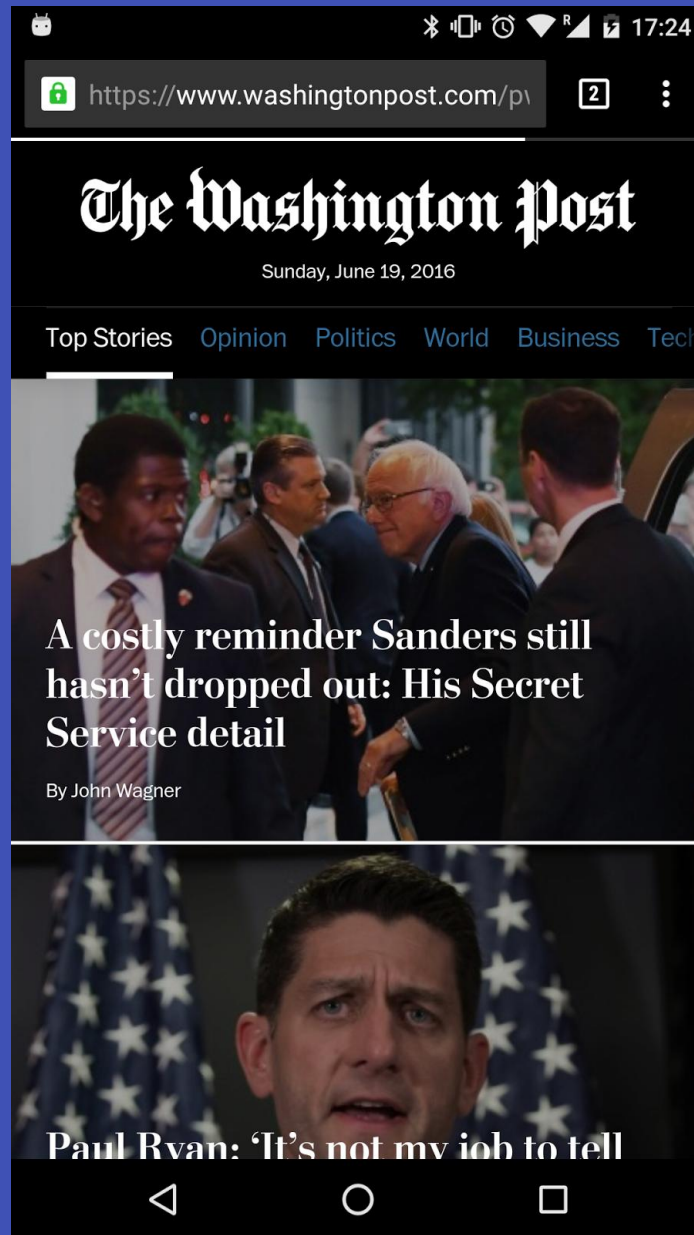
Users abandon sites  
that take more than  
3 seconds to load

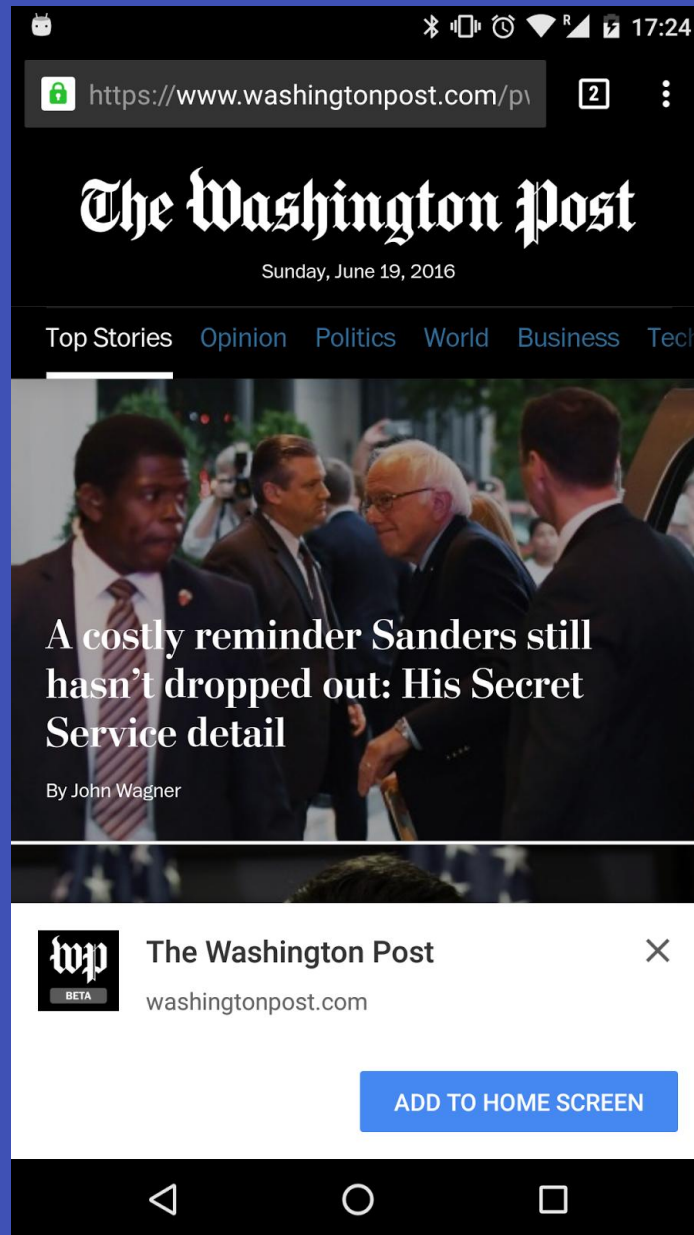
# User experiences need to be:

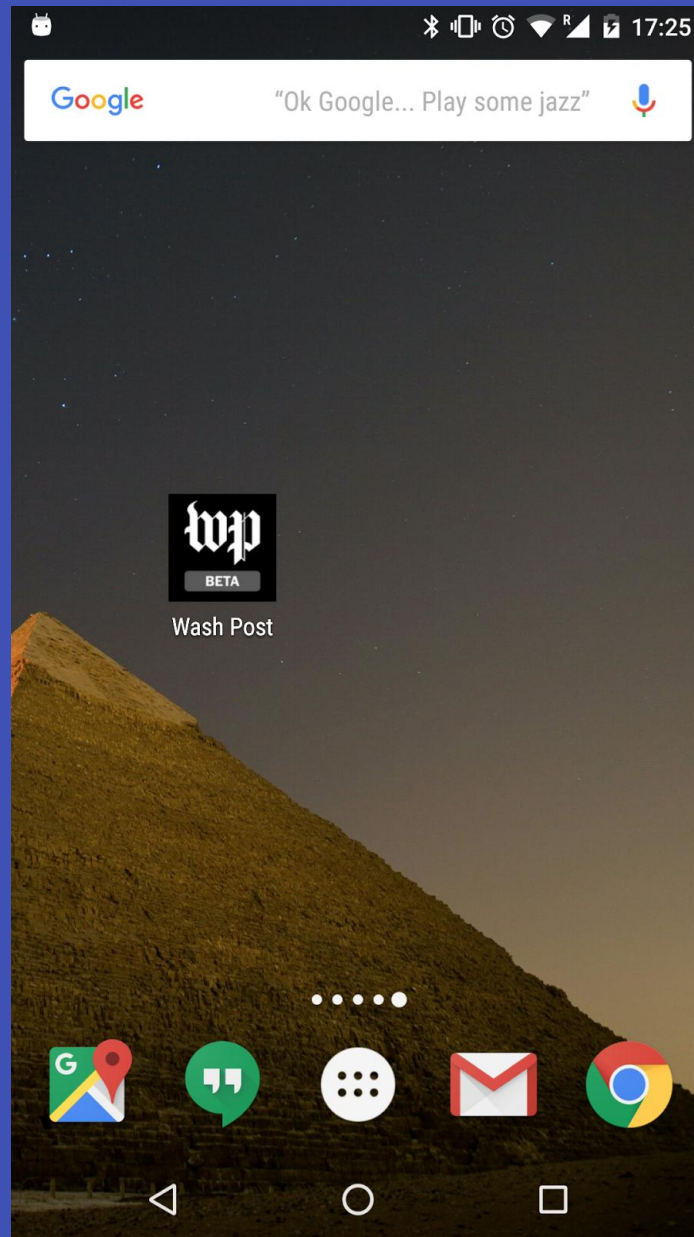
Reliable

Fast

Engaging

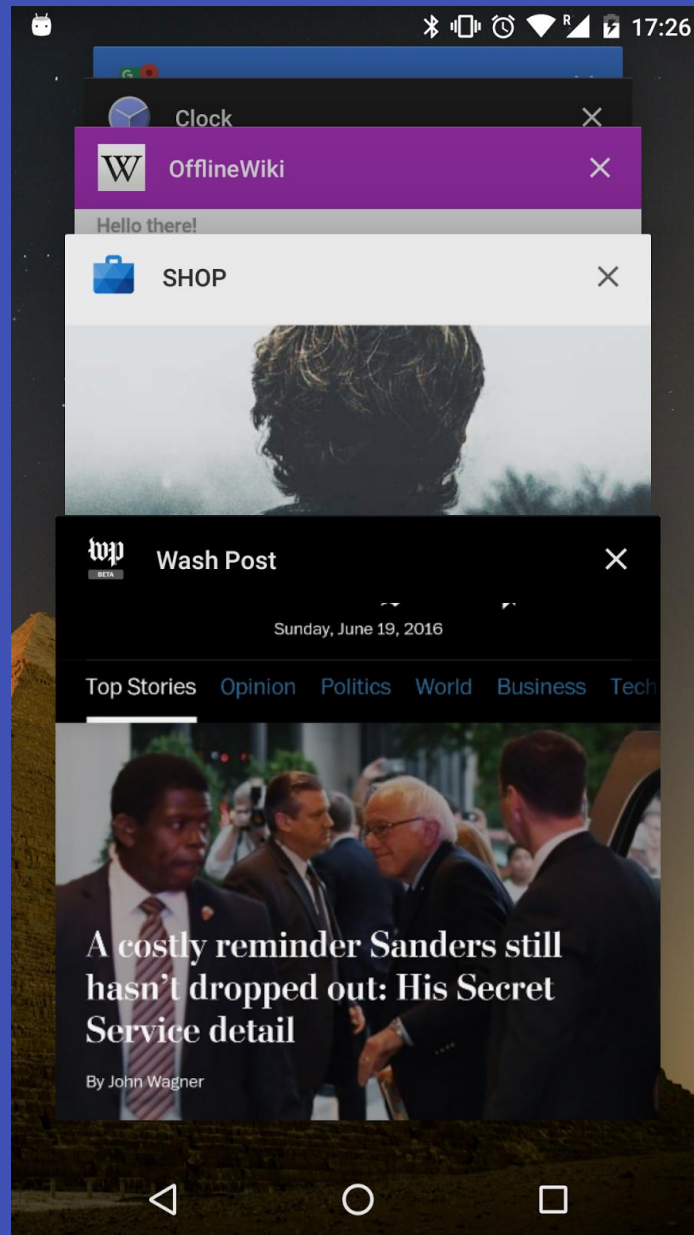












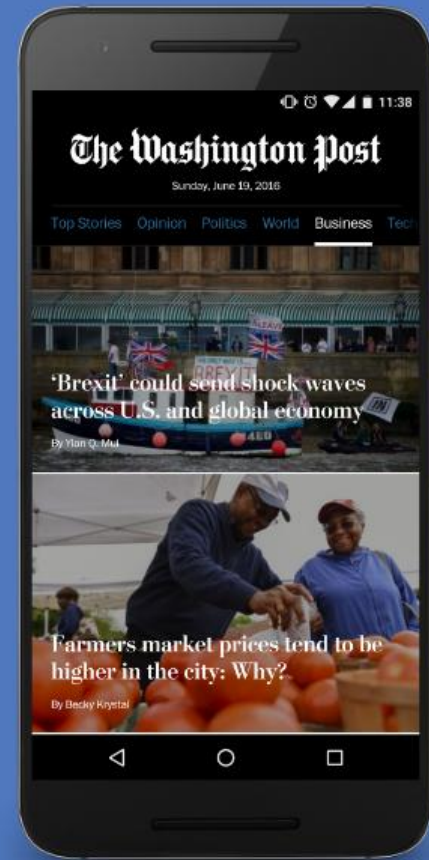


# The Washington Post

## 80ms

article page load time in  
the Progressive Web App

[Poynter article](#)



# User experiences need to be:

Reliable

Fast

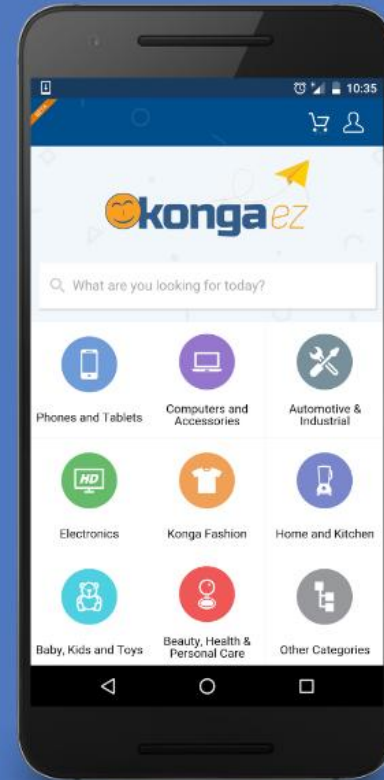
Engaging

# Progressive Web Apps

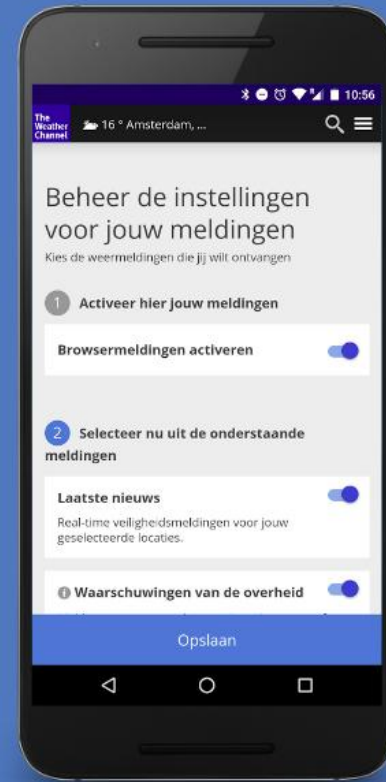
Are a whole new way of caring  
about your User Experience



Both data points are compared to native app







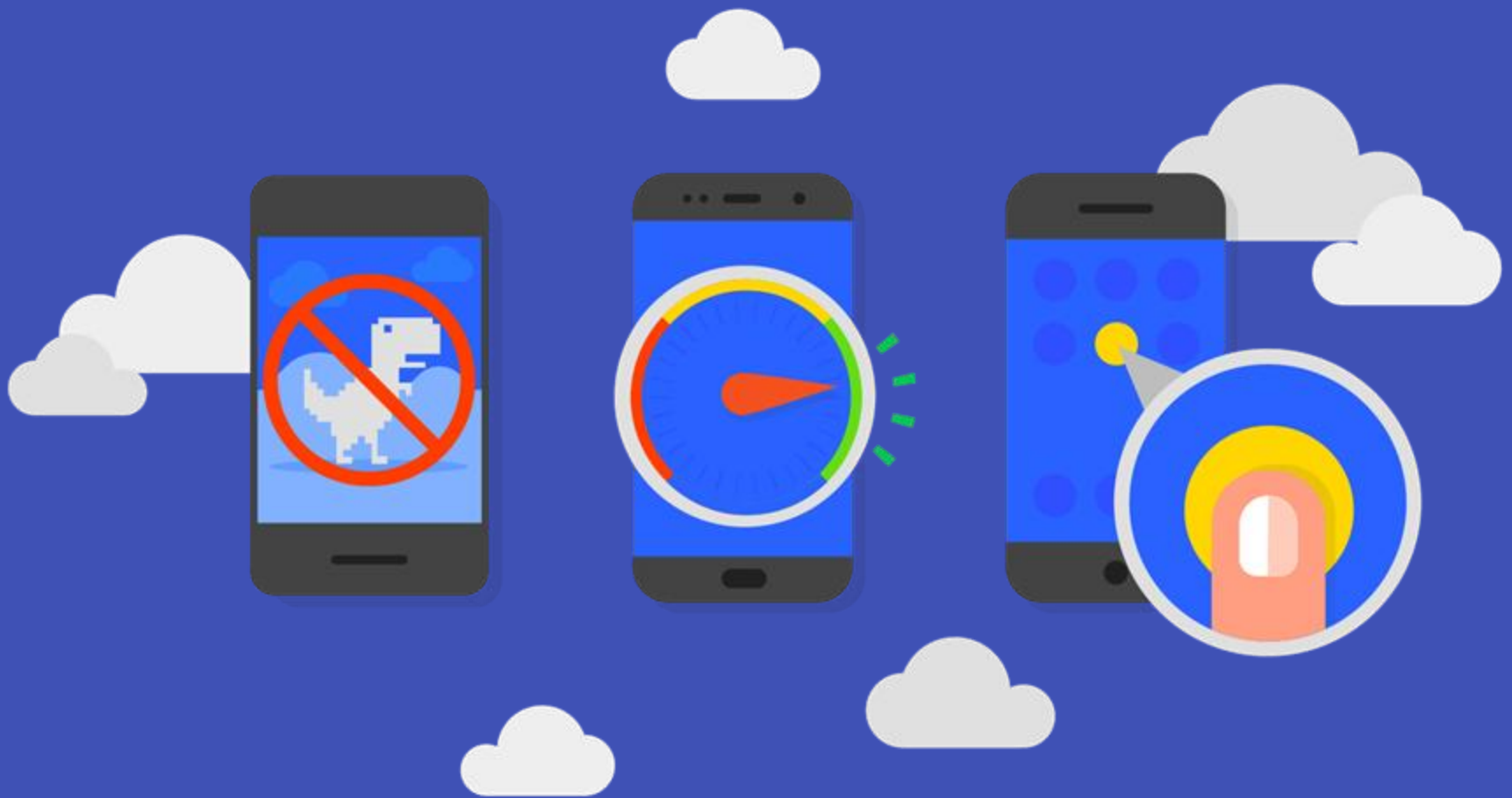


A Progressive Web App  
uses modern browser  
capabilities for native app-  
like experiences that are  
**Reliable, Fast and  
Engaging**



# HOW do I build a Progressive Web App?

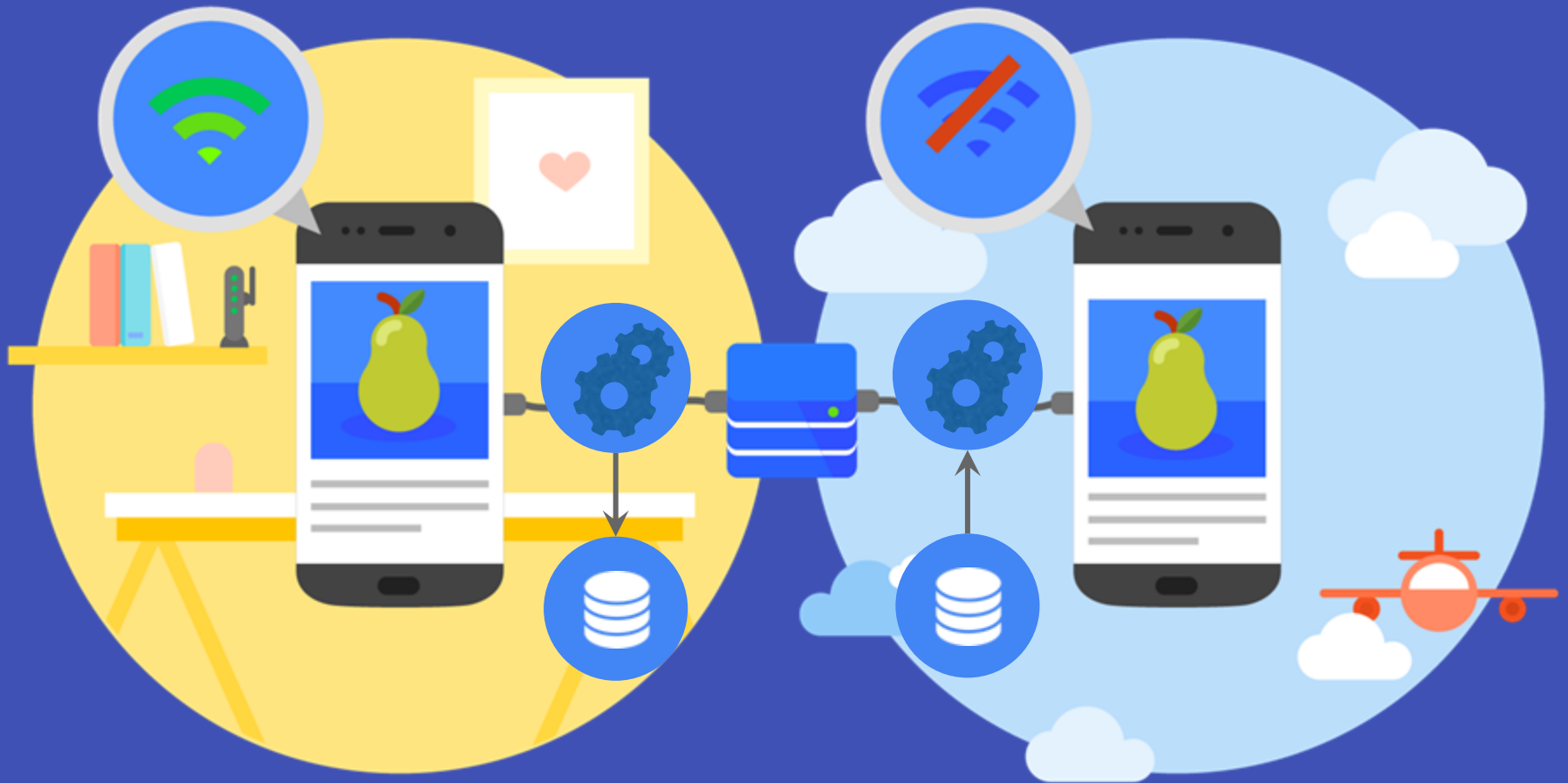




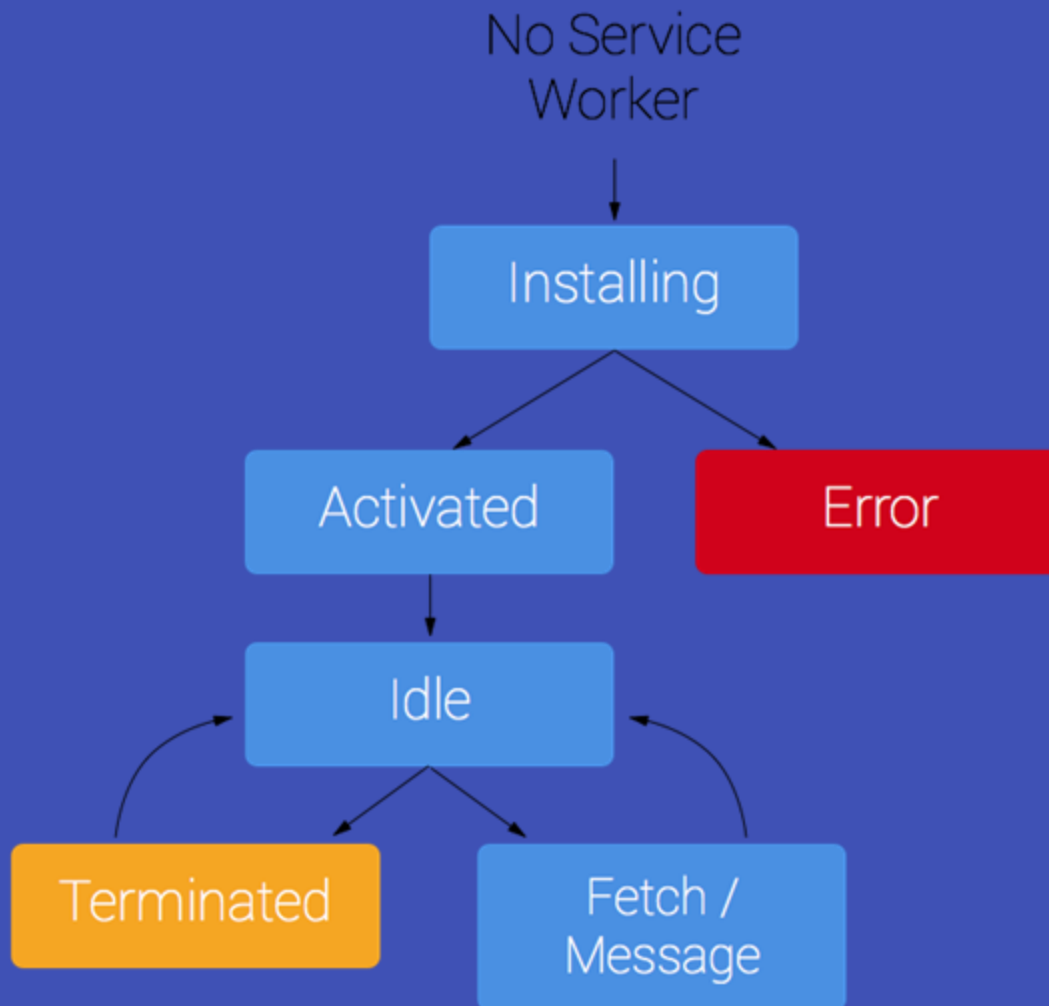
Reliable

Fast

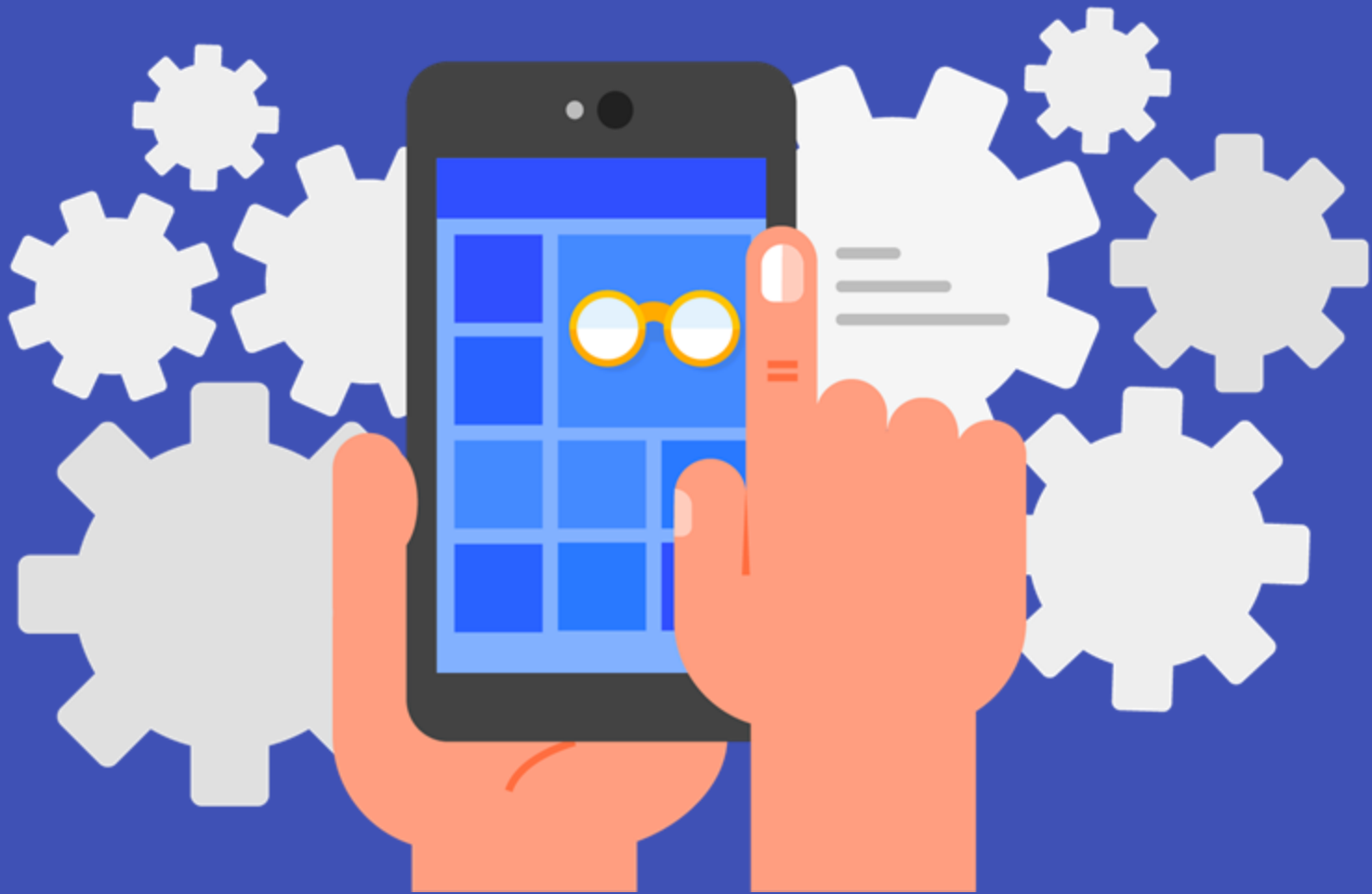
Engaging



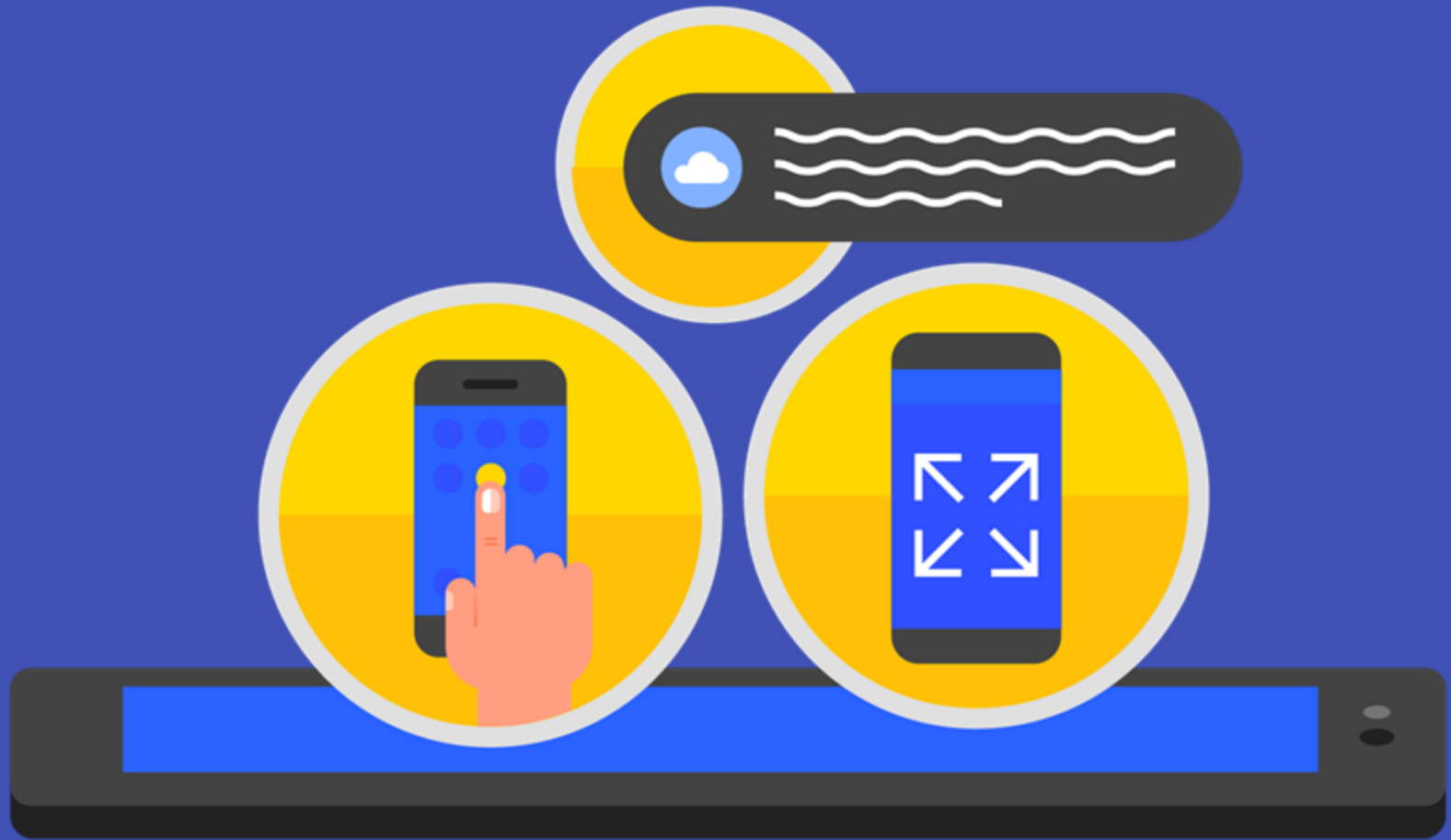
# Reliable with Service Worker



# Reliable with Service Worker



# Fast with App Shell Cache



# Engaging with Homescreen & Push



# To build a PWA:

- 1) Service Worker: Go Offline
- 2) Manifest: Engage Users



# WHAT is Angular Mobile?



Angular Mobile is a collection of plugins and libraries for the App Shell, Service Worker, and App Manifest...





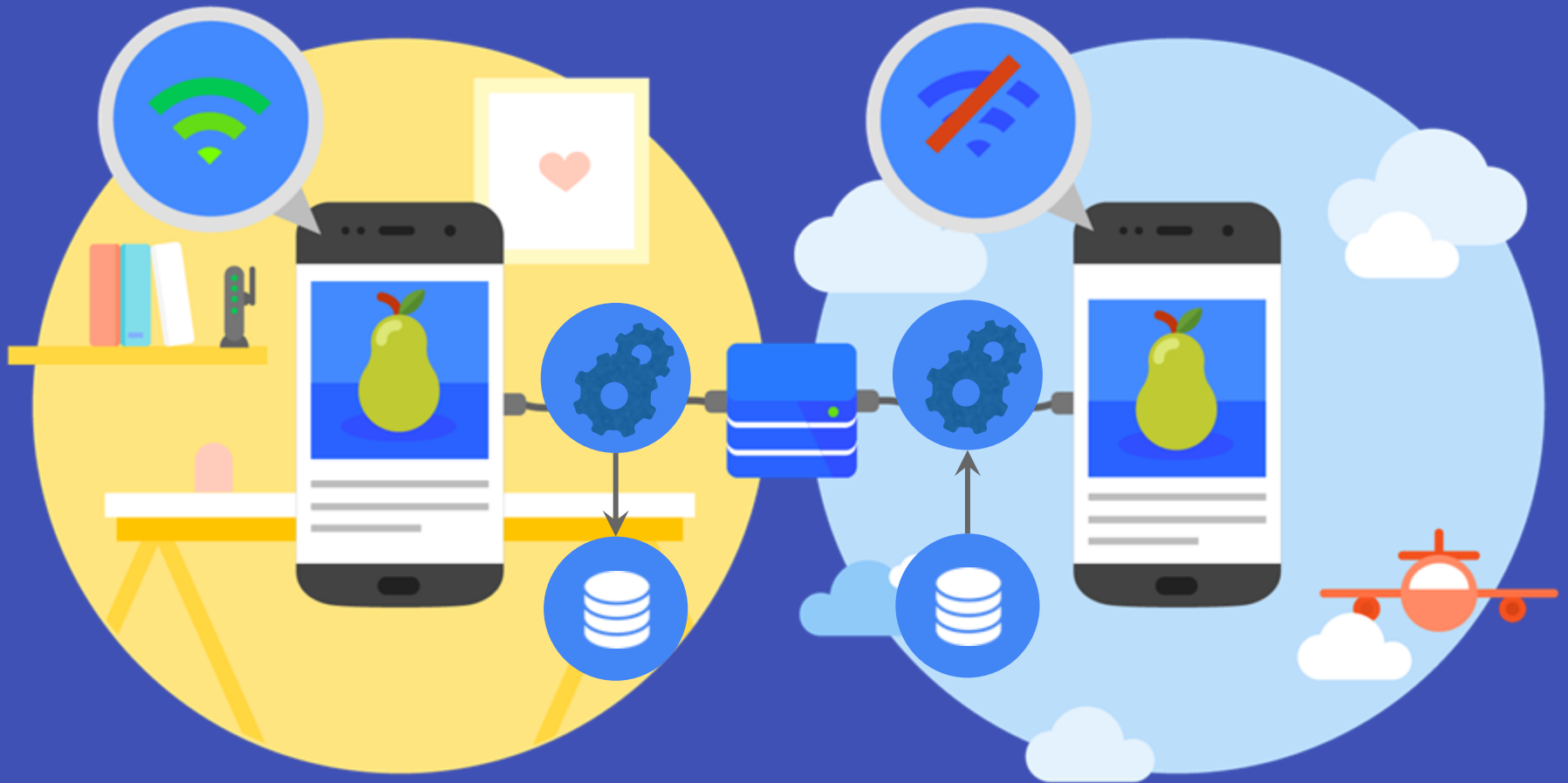
Angular Mobile  
is not ready for use...



There's PWA support  
built into the Angular CLI!



# HOW do I build an Angular Progressive Web App?



# Reliable with Service Worker



# Fast with App Shell Cache



jeffbcross commented on Oct 11, 2016

Contributor + 🗨️

Context: the mobile flag got removed during the transition from a broccoli-based build system to Webpack-based build system a few weeks back. I was on a 6-week-long leave at the time, so I'm not sure all the details of why it was dropped, but I think it was due to Universal not being updated to support Angular 2.0.0 yet. But shout out to @filipesilva and other folks who worked hard to migrate mobile features to the Webpack CLI! The team worked hard to not drop support, but they had a tough choice to make in order to not make users wait for Angular 2.0.0 final support.

When I returned from leave, I'd started adding back the `--mobile` flag in #2320. But after some time working on it and trying to get tests to pass, I realized that even if I merged that PR, the custom App Shell generation plugin was prone to cause further headaches for maintainers and users of CLI for a couple of reasons.

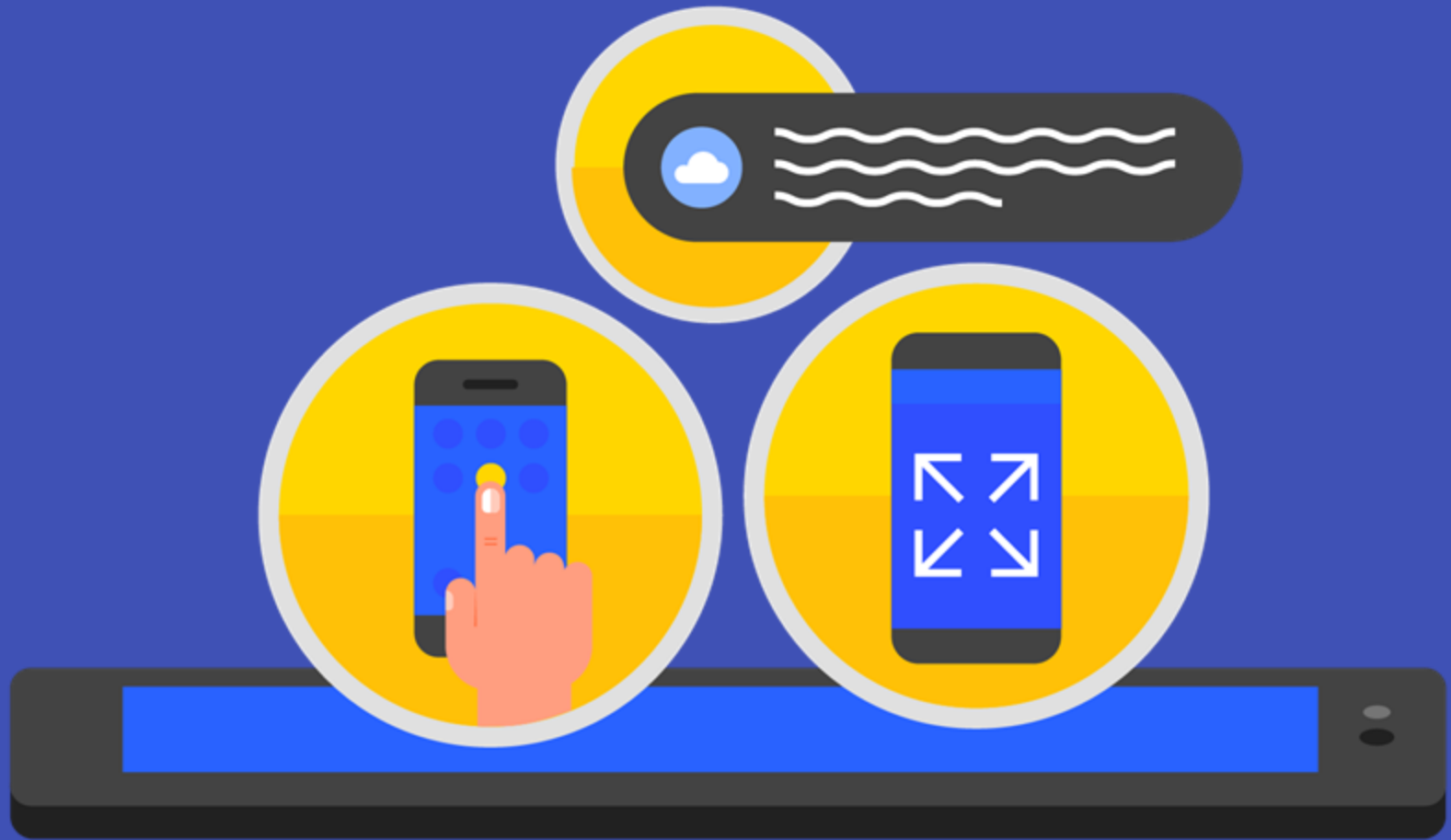
1. Rendering the app shell with Universal requires using services and components that can be rendered in Node. The app shell integration with CLI didn't provide helpful errors when compiling, which could leave users confused when things go wrong. We're exploring how we can generally make build-time pre-rendering more user-friendly outside of the CLI.
2. CLI currently doesn't have an API to add extensions to core behavior, and the current build system isn't very supportive of a feature like Universal pre-rendering. The team working on CLI are planning to support an add-on API in the future, and from what I understand, are also planning to implement a sort of meta build process around Webpack, but don't quote me on that.

To answer the original question, our plan is to re-incorporate the Angular Service Worker plugin into CLI for projects that use the `--mobile` flag. @alxhub is working hard on this. We're also going to generate the web app manifest and icons as we were doing before. For the time being, the App Shell generation won't be built into the CLI. We're working on exploring ways to make pre-rendering simpler, and will provide updated docs on mobile.angular.io on how to generate App Shell with CLI and with other build tools.



21

# Fast with AppShell Cache



# Engaging with Homescreen & Push



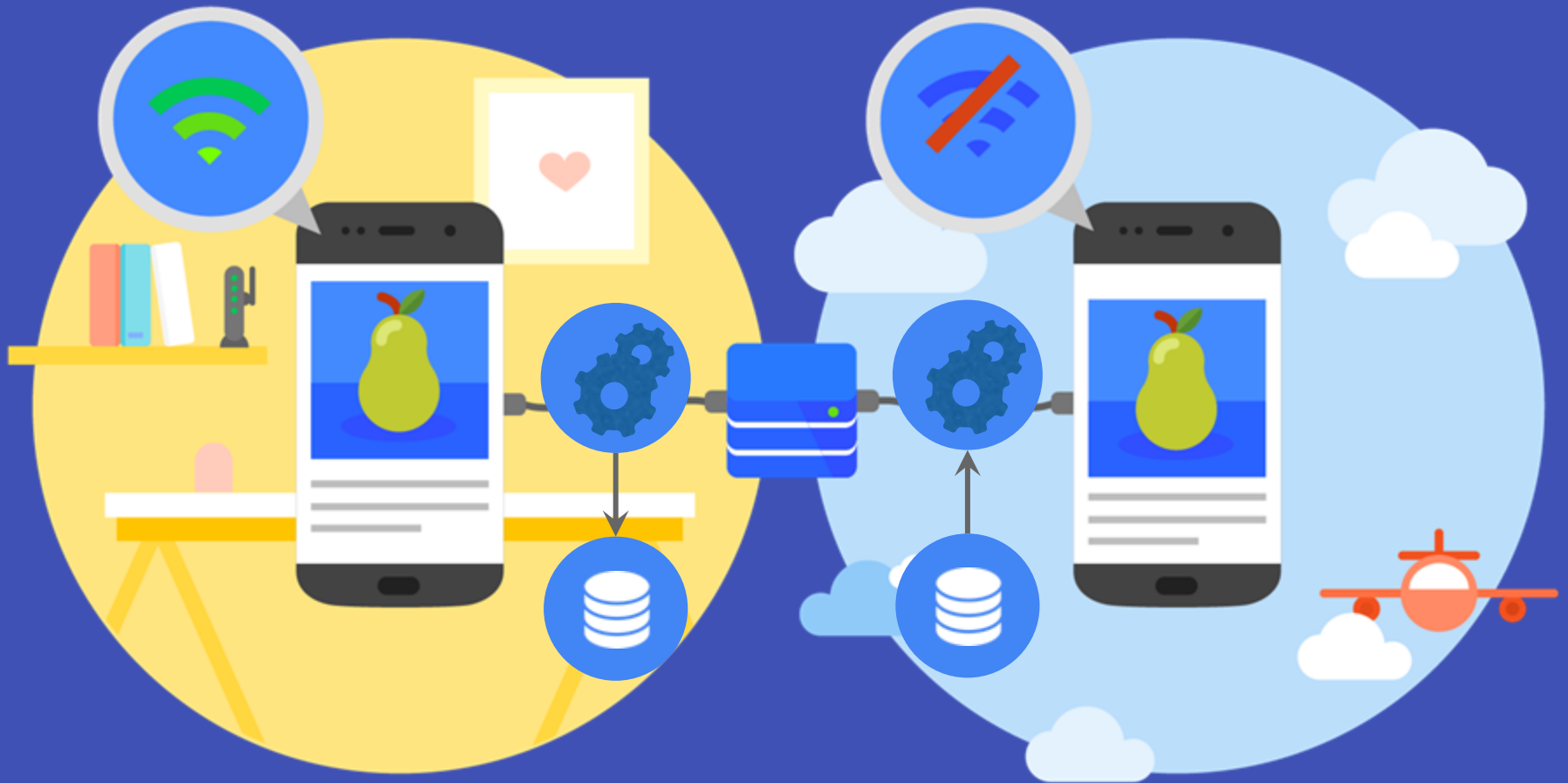
# To build an Angular PWA:

- 1) Service Worker: Go Offline
- 2) Manifest: Engage Users





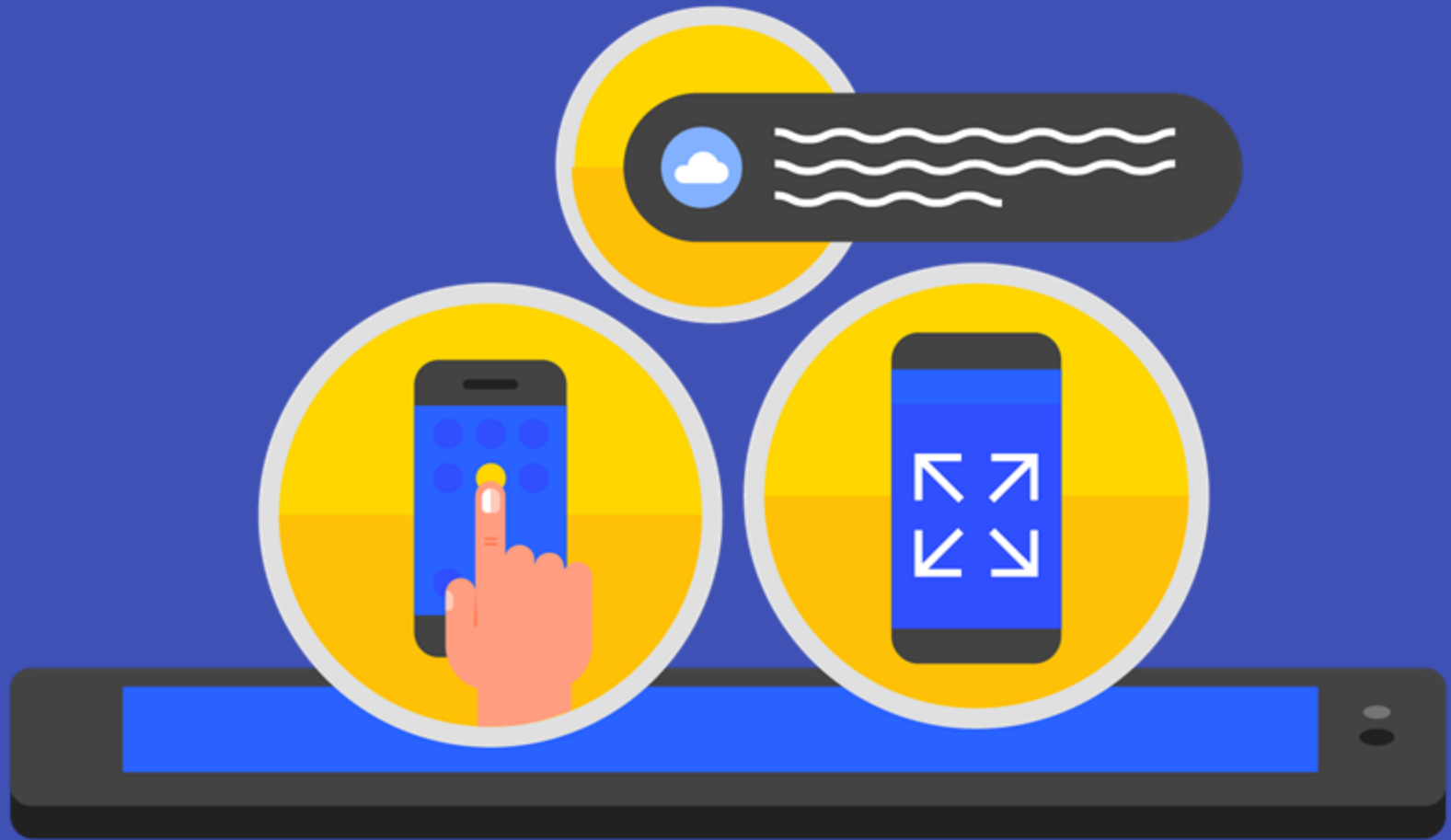
# HOW do I build an React Progressive Web App?



# Reliable with Service Worker



# Fast with App Shell Cache



# Engaging with Homescreen & Push



# To build an React PWA:

- 1) Service Worker: Go Offline
- 2) Manifest: Engage Users



WHY should I use  
PWAs?



# Progressive Web Apps are awesome because:

- Build reliable, offline-ready apps
- Load up content fast
- Engage your users
- ...



# Progressive Web App

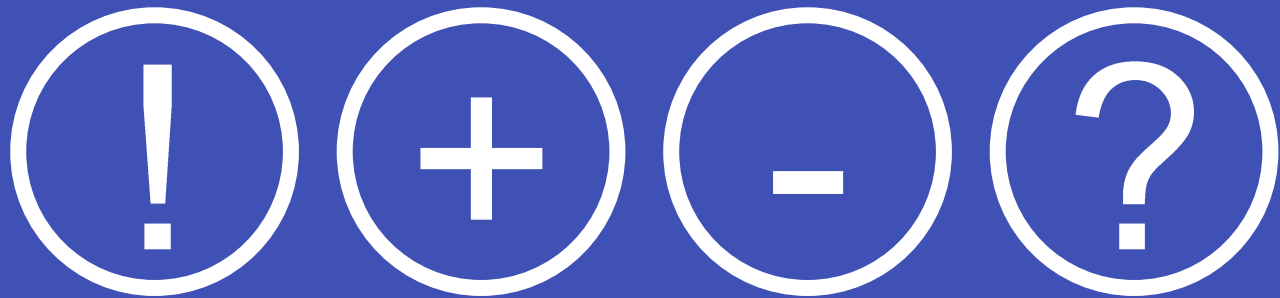
WHAT is it ?

HOW can I use it ?

WHY should I care ?



# THANK YOU!





BUILD FOR MOBILE

# Progressive Web Apps

Christina Kayastha  
Senior Software Engineer  
Vistaprint, Cimpres  
@christikaes