

Getting started with

Progressive Web Apps



Aaron Gustafson
@AaronGustafson
noti.st/AaronGustafson

Follow along

- Slides: <https://aka.ms/pwasummit-workshop-slides>
- Work files: <https://aka.ms/pwasummit-workshop-code>
- Final Demo: <https://aka.ms/pwasummit-workshop-live>

What exactly is a
PWA?

What exactly is a
Progressive Web App?

Infrequently Noted

Alex Russell on browsers, standards, and the process of progress.

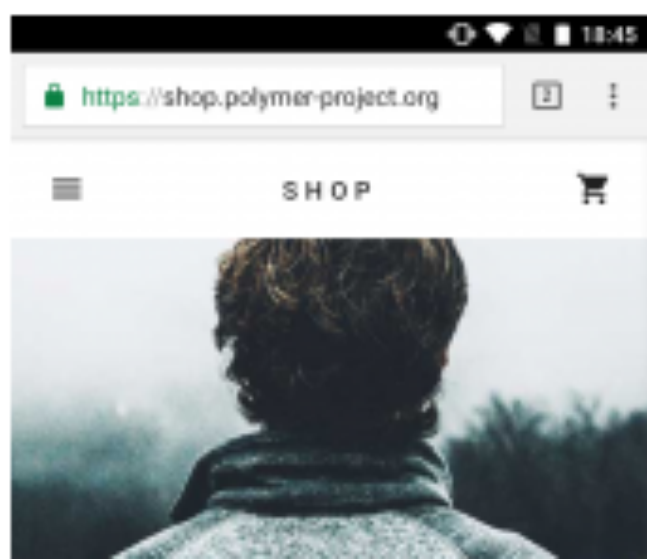
[« PWA Discovery: You Ain't Seen Nothin Yet](#)

[Web Components: The Long Game »](#)

What, Exactly, Makes Something A Progressive Web App?

Since [Frances](#) and I published a blog post last year introducing [Progressive Web Apps](#), a healthy conversation has started about what is and isn't a PWA. There are a lot of opinions and many shades of gray. What are the hard requirements? Which requirements are marginal? What's aspirational? This article outlines these requirements, attempts to classify them, and provides a baseline for "what is a Progressive Web App?"

Browsers gate Progressive Web App installation prompting and badging on criteria that they detect when users navigate to sites. These criteria have been designed to ensure that sites which invoke prompts are reliable, fast, and engaging.



What's All This Then?

I'm Alex Russell, a web developer working on Chrome, Blink, and the Web Platform at Google. I'm guilty of many JavaScript transgressions.

I help lead the team building a new application model for the web, and serve on ECMA TC39 (the standards body for JavaScript). I'm an elected member of the W3C Technical Architecture Group and am Tech Lead for Standards inside the Chrome team. I design and advocate for extensible, layered, data-driven evolution of the web platform.

My professional aim is to make the web a better platform and to the extent that I can keep politics and economics from messing in

**“Progressive Web App”
is a marketing term**

Progressive Web App

Game

Gallery

Book

Progressive Web App

Newspaper

Art Project

Tool

Progressive Web Site

Who's behind PWAs?



What's a PWA, technically?



HTTPS



Web App
Manifest



Service
Worker

Why I predict Progressive Web Apps will change the game for agencies

by PRATIK DHOLAKIYA — 28 days ago in CONTRIBUTORS

How Progressive Web Apps Could Change the Mobile Landscape

It's time retailers embrace progressive web apps

GUEST COMMENT 23 Aug 2018

Should I believe the hype?

circuit breaker
Web apps are only getting better

By Paul Miller | @futurepaul | Apr 11, 2018, 8:20am EDT

25,962 views | Mar 9, 2018, 07:30am
Why Progressive Web Apps Will Replace Native Mobile Apps

Leaki CommunityVoice

Maybe?

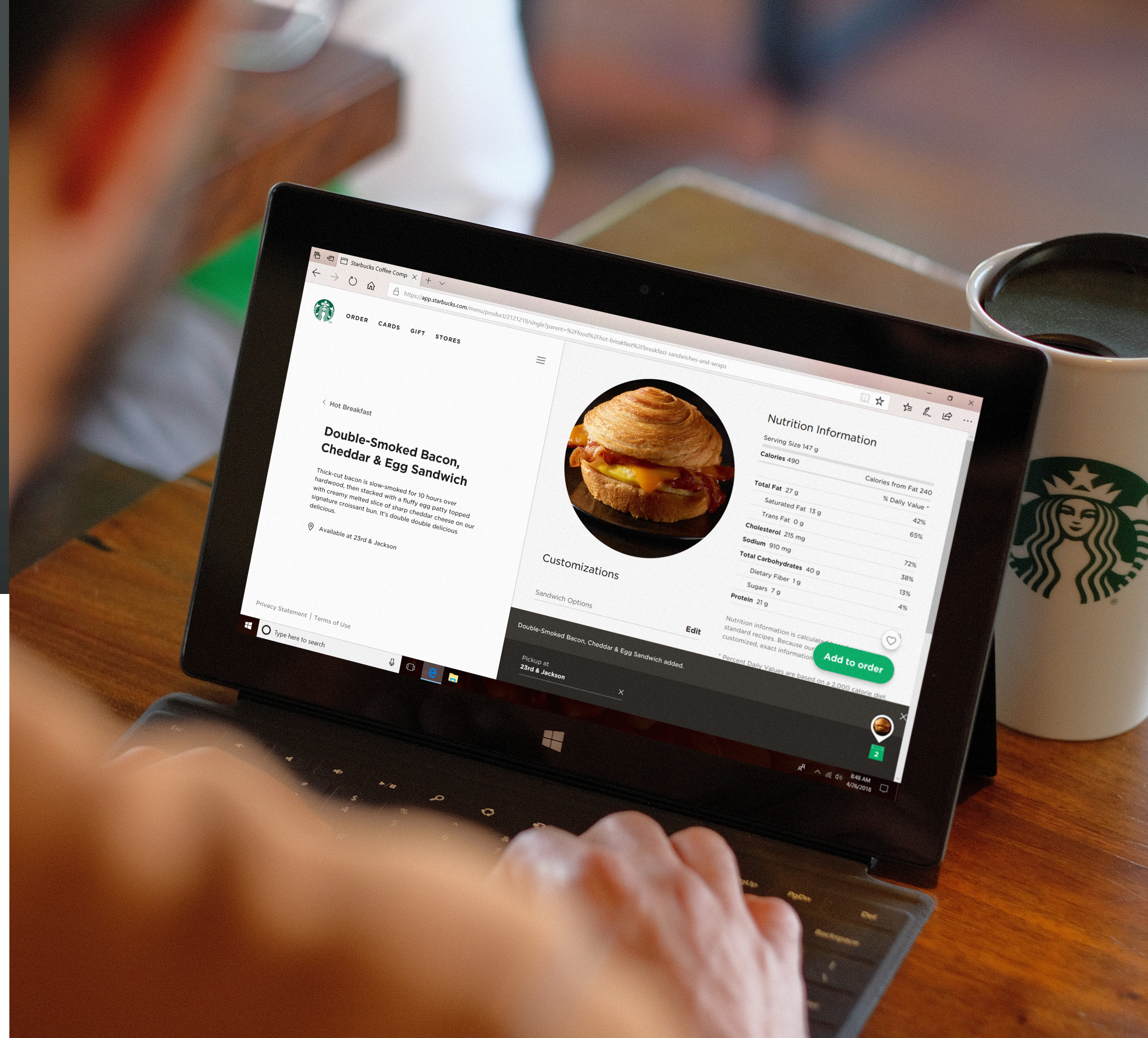
Starbucks:

2x increase in daily active users

desktop == mobile



aka.ms/google-io-2018

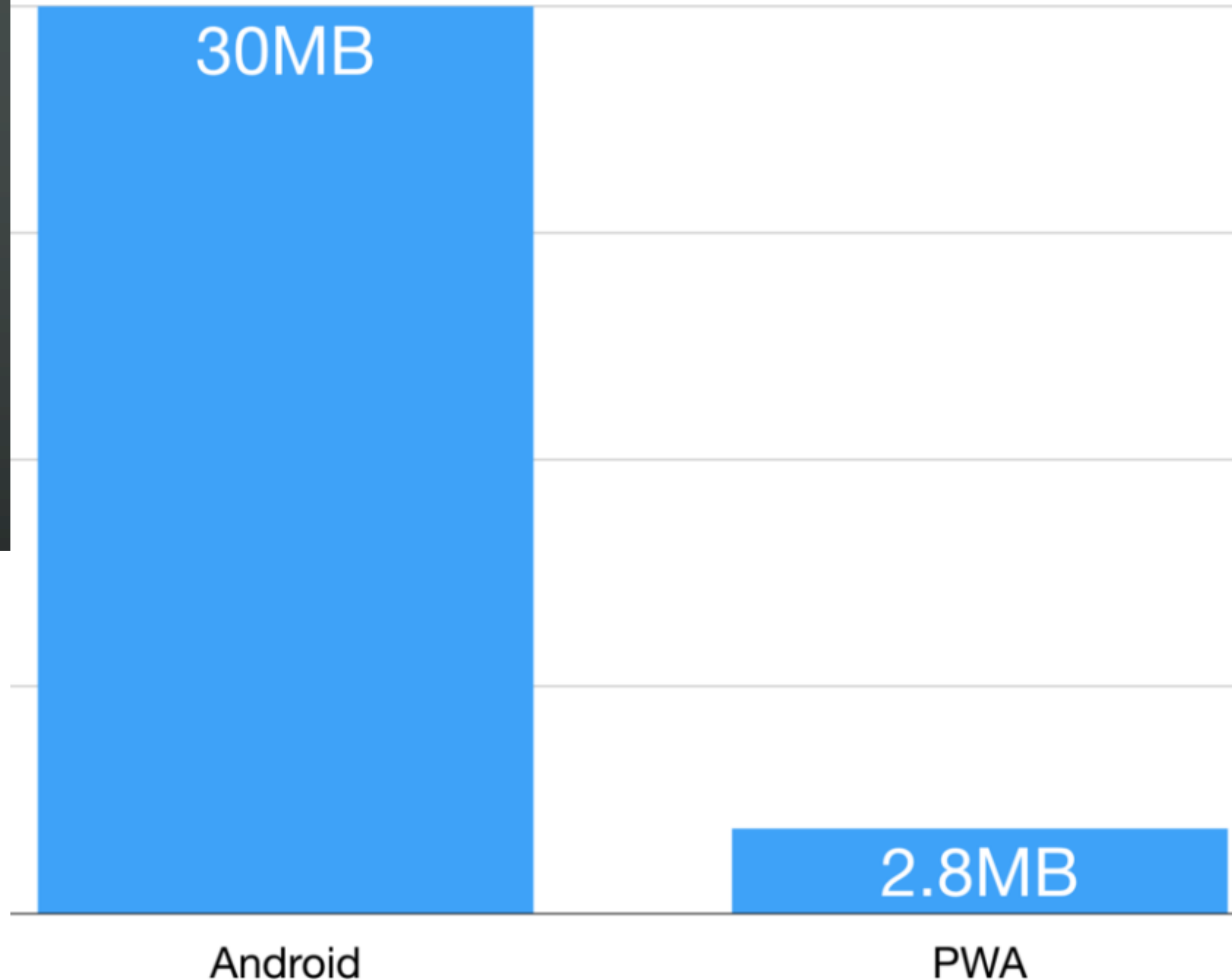


Tinder:

Core experience
with 90% less code



aka.ms/tinder-pwa-2017



Uber:

Core PWA in 50k*

**Loads in under 3s
over 2G**

* gzipped

Uber

aka.ms/uber-pwa-2021



Forbes:

61% improvement in
load time on mobile

Forbes

aka.ms/forbes-pwa-2021



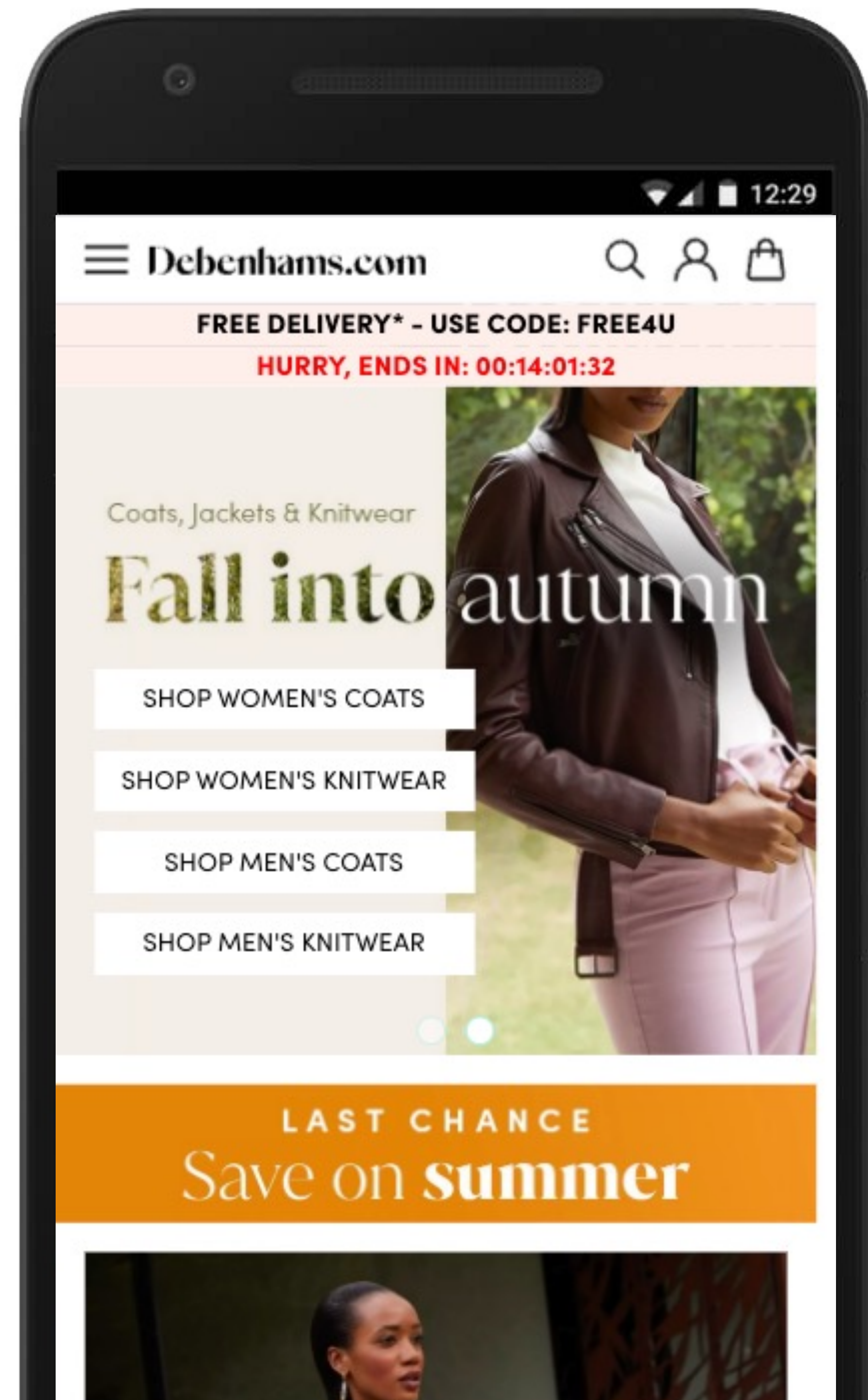
Debenhams:

40% increase in mobile revenue

20% increase in conversion

Debenhams

aka.ms/debenhams-pwa-2018



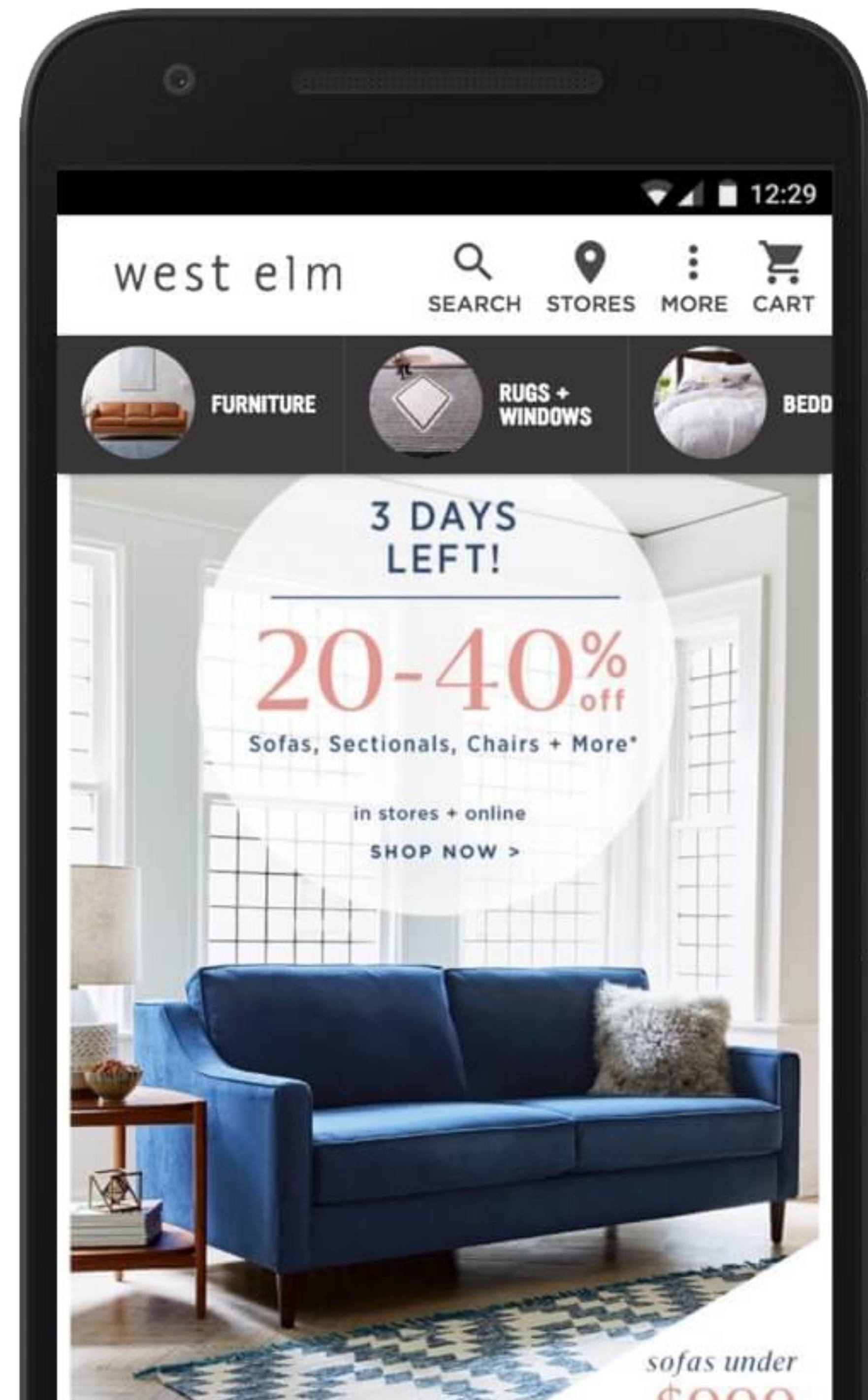
West Elm:

15% increase in
time on site

9% increase in
revenue per visit

west elm

aka.ms/west-elm-pwa-2017



Flipkart:

50% of new customers via PWA

60% of PWA users uninstalled their app

Flipkart



aka.ms/flipkart-2021

Offers



3 Colors

United Colors of Benetton Men Sneakers (Blue)

₹1,739 ~~3,499~~ 50% off  Assured

FREE Delivery available

3.8 ★ 39 ratings & 3 reviews

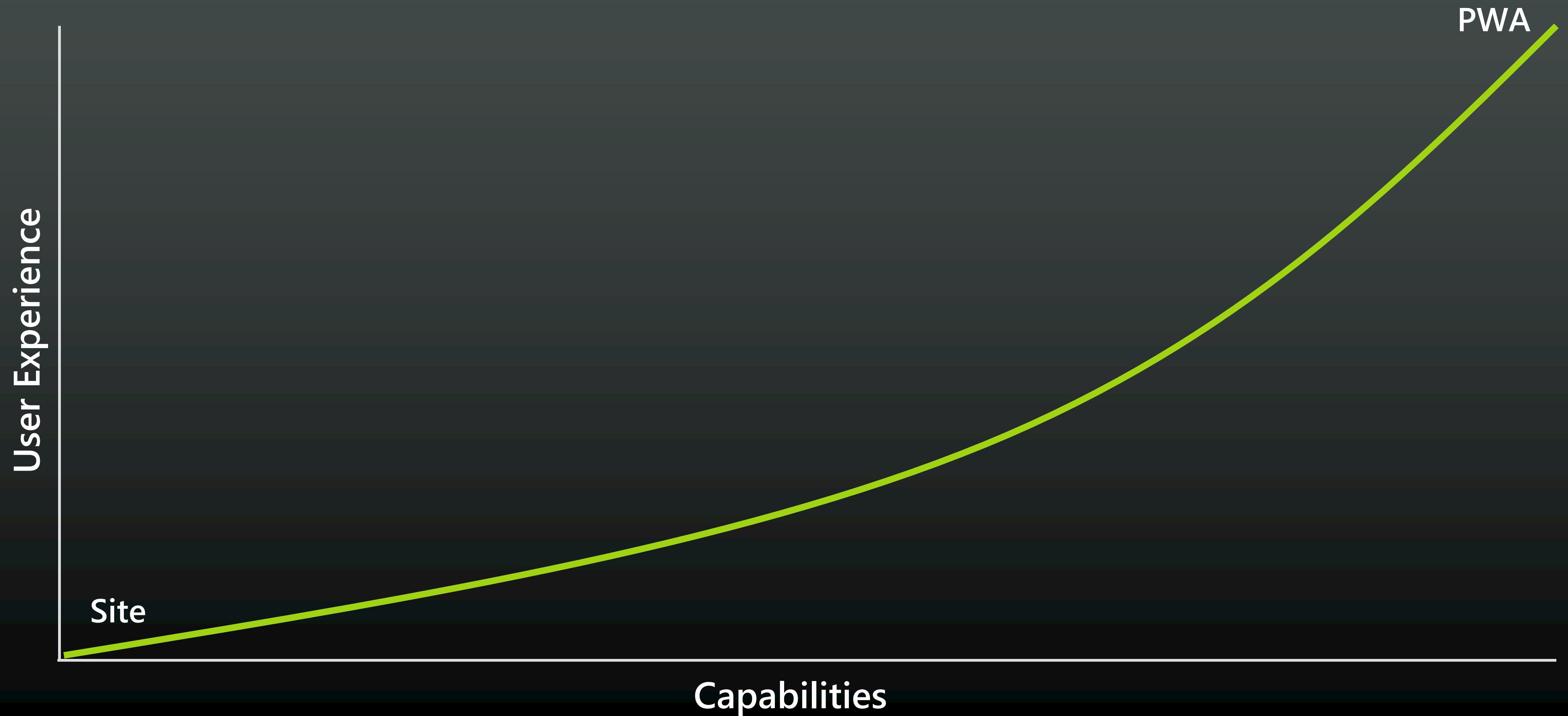
PWA Stats

PWAs start with a great web experience and then **enhance** that experience for performance, resilience, installation, and engagement

Progressive Web App

Progressive Web App Enhancement

Enhance the experience



Progressive
/prə'gresiv/

happening or developing
gradually or in stages;
proceeding step by step

Let's talk about HTTPS



HTTPS is simple (& free) now

- Many hosts include it
 - GitHub
 - Netlify
 - AWS
 - etc.
- LetsEncrypt & Certbot for everything else
<https://letsencrypt.org/>

Let's talk about the Manifest



Manifest files are JSON files

```
{  
  "property_a": "value",  
  "property_b": ["value_1", "value_2"],  
  "property_c": {  
    "nested_property": "nested_value"  
  }  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "dir": "ltr",  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "dir": "auto", // default  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "start_url": "/"  
}
```


Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "short_name": "AaronG",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "short_name": "AaronG",  
  "start_url": "/"  
}
```

Minimum Viable Manifest

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "short_name": "AaronG",  
  "start_url": "/"  
}
```

Reference in the head

```
<link rel="manifest" href="/manifest.json">
```


Reference in the head

```
<link rel="manifest" href="/manifest.json">
```

Reference in the head

```
<link rel="manifest" href="/manifest.json">
```

*app_name.webmanifest
(or similar) would also be
acceptable, but requires MIME
application/manifest+json*



Let's make a manifest!

- Open a new text document in your site and name it manifest.json
- Create a basic JSON object inside, including the following Manifest members:
 1. `lang` (and, optionally, `dir`),
 2. `name`,
 3. `short_name` (if you need it), and
 4. `start_url`.

Let's prep for install...

```
{  
  "lang": "en-US",  
  "name": "Aaron Gustafson",  
  "short_name": "AaronG",  
  "start_url": "/"  
}
```


Start at the beginning

```
{  
  ...  
  "start_url": "/"  
}
```

Where does this apply?

```
{  
  ...  
  "start_url": "/",  
  "scope": "/" // defaults to the start_url path  
}
```


What should it look like?

```
{  
  ...  
  "start_url": "/",  
  "display": "minimal-ui"  
}
```

AARON GUSTAFSON

This site is undergoing an open redesign, so if it looks strange, that's why.

[About](#) [Notebook](#) [Speaking](#) [Publications](#) [Interviews](#) [Contact](#) [Search](#)

Hi there.

My name is Aaron Gustafson and I work on the web.

I am a web standards & accessibility advocate at [Microsoft](#), where I work closely with the Edge browser team. I also help organizations elegantly tackle the challenges presented by all the crazy devices and screens you see everywhere using [progressive enhancement](#) via [Easy Designs](#) and [Rosenfeld Media](#). You'll often find me on the road, [presenting at conferences and running workshops](#) across the globe.

I serve on the [W3C's Responsive Issues Community Group](#). I founded the [Chattanooga Open Device Lab](#) and co-founded the event series [Code & Creativity](#) and [Retreats 4 Geeks](#). In previous roles, I managed the [Web Standards Project \(WaSP\)](#) and published [Web Standards Sherpa](#)

I wrote the web design book [Jeffrey Zeldman](#) calls [a "modern classic"](#) and [Jeremy Keith](#) calls ["the clearest, most beautiful explanation of progressive enhancement I've ever read"](#): [Adaptive Web Design](#). I've also contributed to

`"display": "fullscreen"`

dozens more during my tenure as both a Tech Editor and Editor in Chief for the esteemed magazine [A List Apart](#)

Need to lock orientation?

```
{  
  ...  
  "start_url": "/",  
  "display": "minimal-ui",  
  "orientation": "any"  
}
```

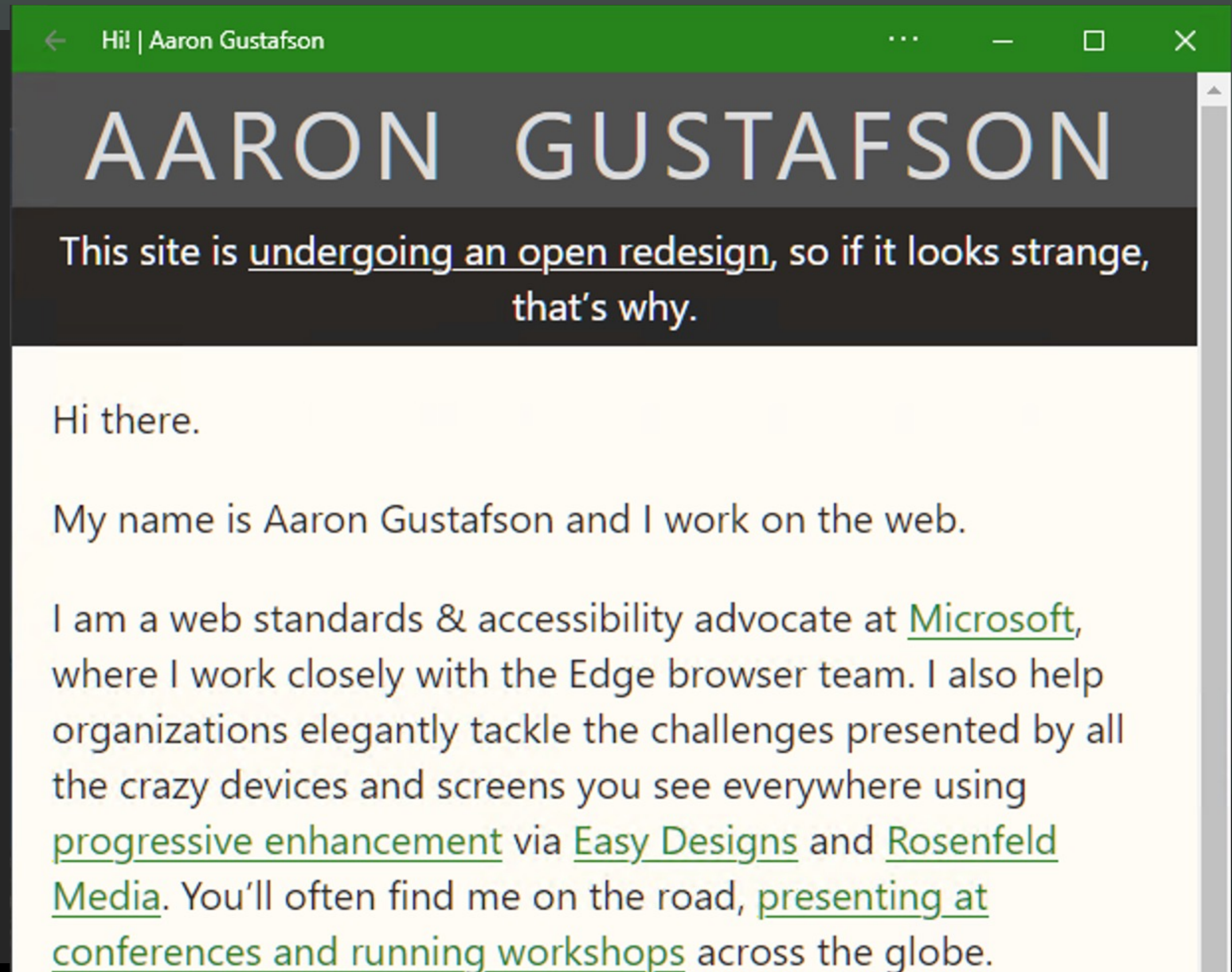
Orientation options

- "any" – no preference
- "natural" – the default orientation of the device
- "portrait"
 - "portrait-primary"
 - "portrait-secondary"
- "landscape"
 - "landscape-primary"
 - "landscape-secondary"

Details: <https://www.w3.org/TR/screen-orientation/>

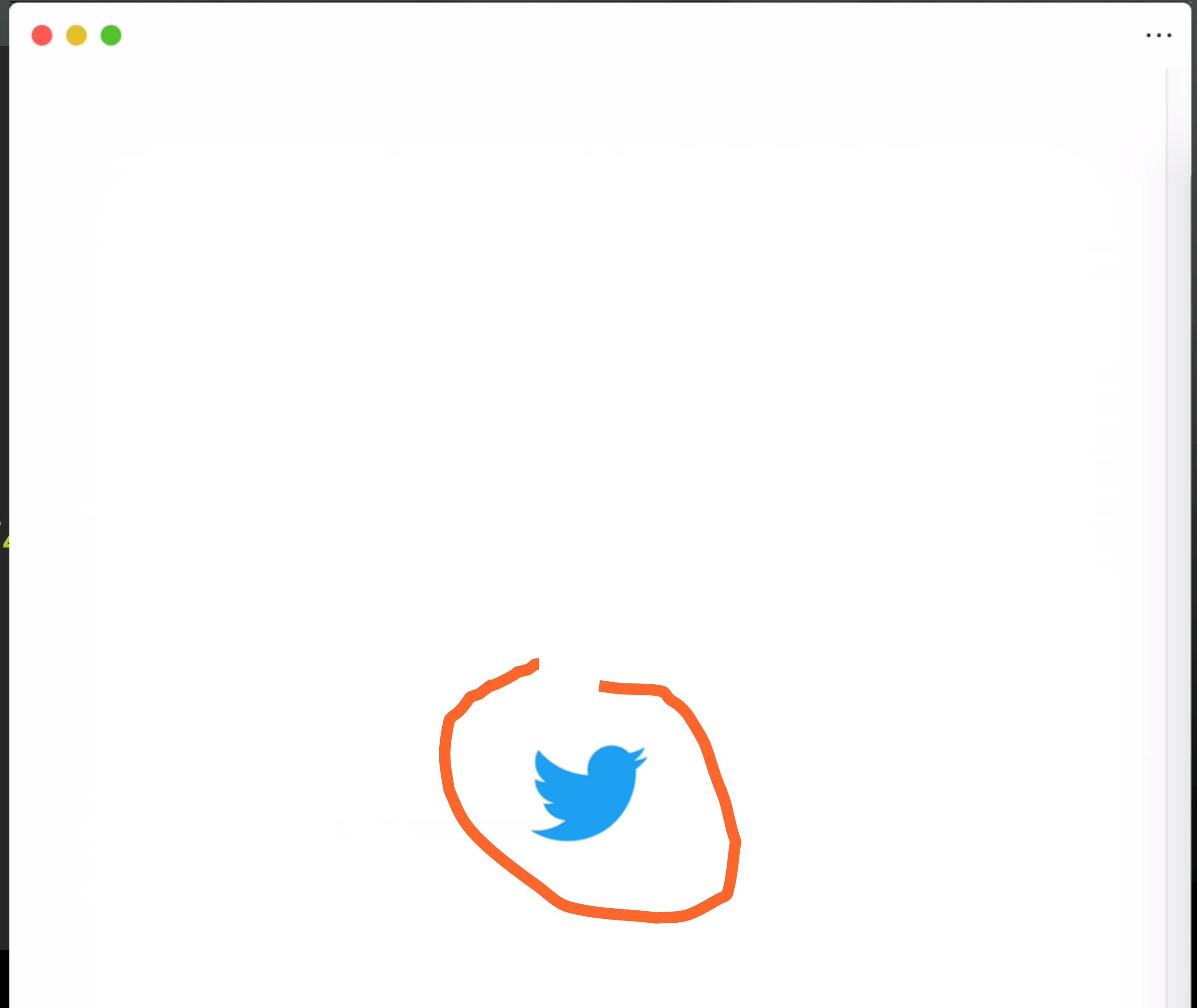
A little color...

```
{  
  ...  
  "start_url": "/",  
  "display": "minimal-ui",  
  "orientation": "any",  
  "theme_color": "#27831B"  
}
```



A little color...

```
{  
  ...  
  "start_url": "/",  
  "display": "minimal-ui",  
  "orientation": "any",  
  "theme_color": "#27831B",  
  "background_color": "#ffffcf"  
}
```



Adding icons

```
{  
  ...  
  "icons": [  
    { "src": "/i/og-logo.png",  
      "type": "image/png",  
      "sizes": "800x600" },  
    { "src": "/i/notification-icon.png",  
      "type": "image/png",  
      "sizes": "256x256" },  
    { "src": "/favicon.png",  
      "type": "image/png",  
      "sizes": "16x16" }  
  ]  
}
```

Adding icons

```
{  
  ...  
  "icons": [  
    { "src": "/i/og-logo.png",  
      "type": "image/png",  
      "sizes": "800x600" },  
    { "src": "/i/notification-icon.png",  
      "type": "image/png",  
      "sizes": "256x256" },  
    { "src": "/favicon.png",  
      "type": "image/png",  
      "sizes": "16x16" }  
  ]  
}
```

Anatomy of an ImageResource

```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600"  
}
```


Anatomy of an ImageResource

```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600"  
}
```

Anatomy of an ImageResource

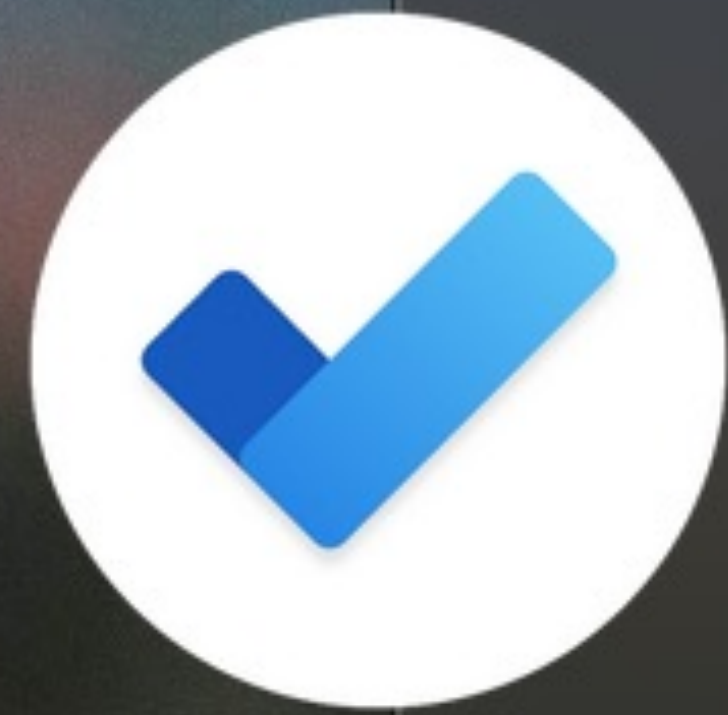
```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600"  
}
```

Anatomy of an ImageResource

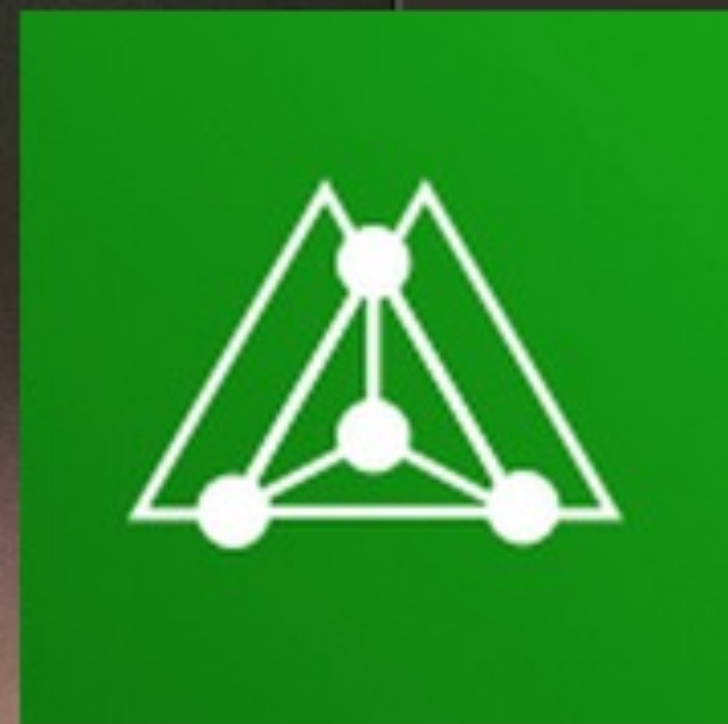
```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600"  
}
```


Anatomy of an ImageRe

```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600"  
}
```



Twitter



Anatomy of an ImageResource

```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600",  
  "purpose": "monochrome"  
}
```

If you want to enable the OS
to mask your logo over a solid fill
(such as for a badge)

Anatomy of an ImageResource

```
{  
  "src": "/i/og-logo.png",  
  "type": "image/png",  
  "sizes": "800x600",  
  "purpose": "maskable"  
}
```

image is designed
with icon masks and
safe zone in mind

Let's improve our manifest!

- Add the following Manifest members:

1. `display`,

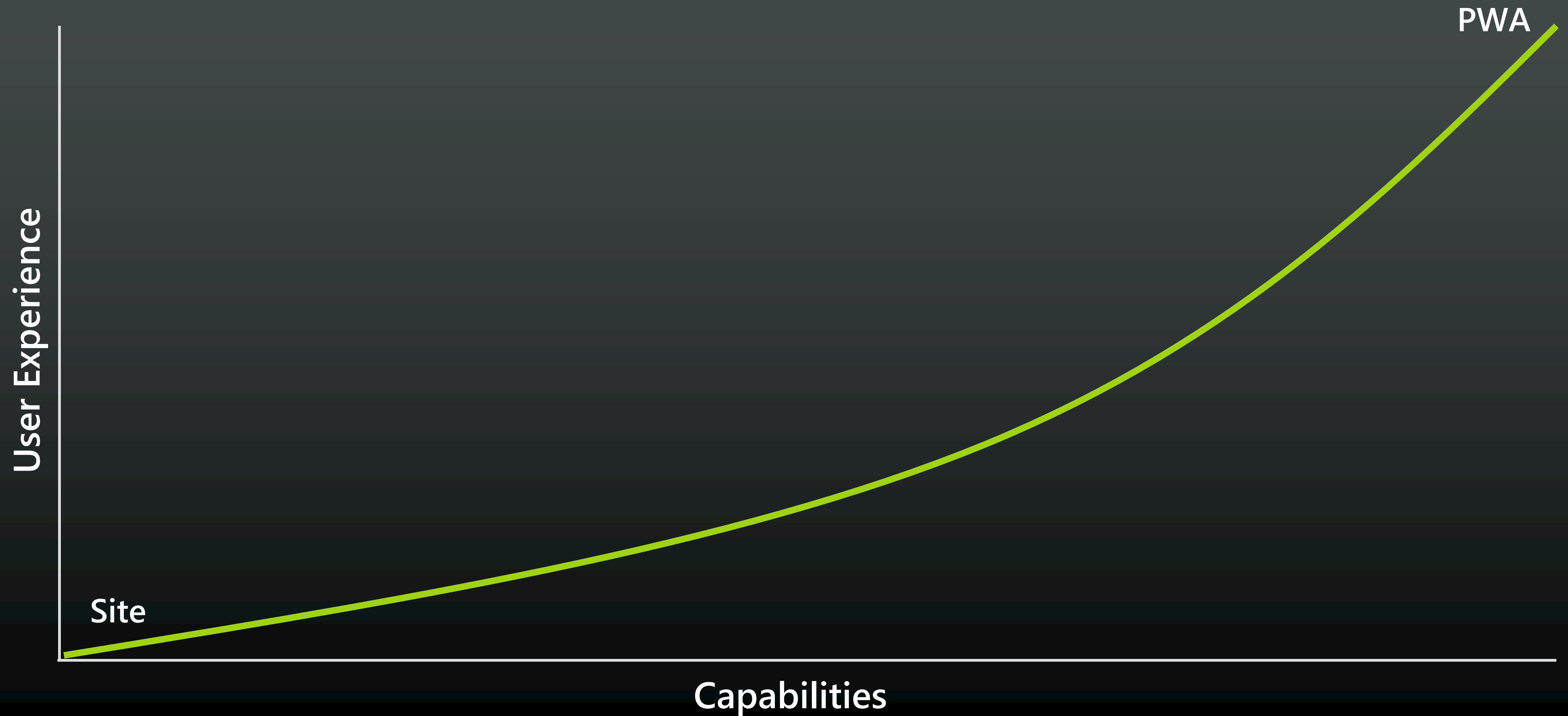
2. `theme_color`,

3. `background_color`, and

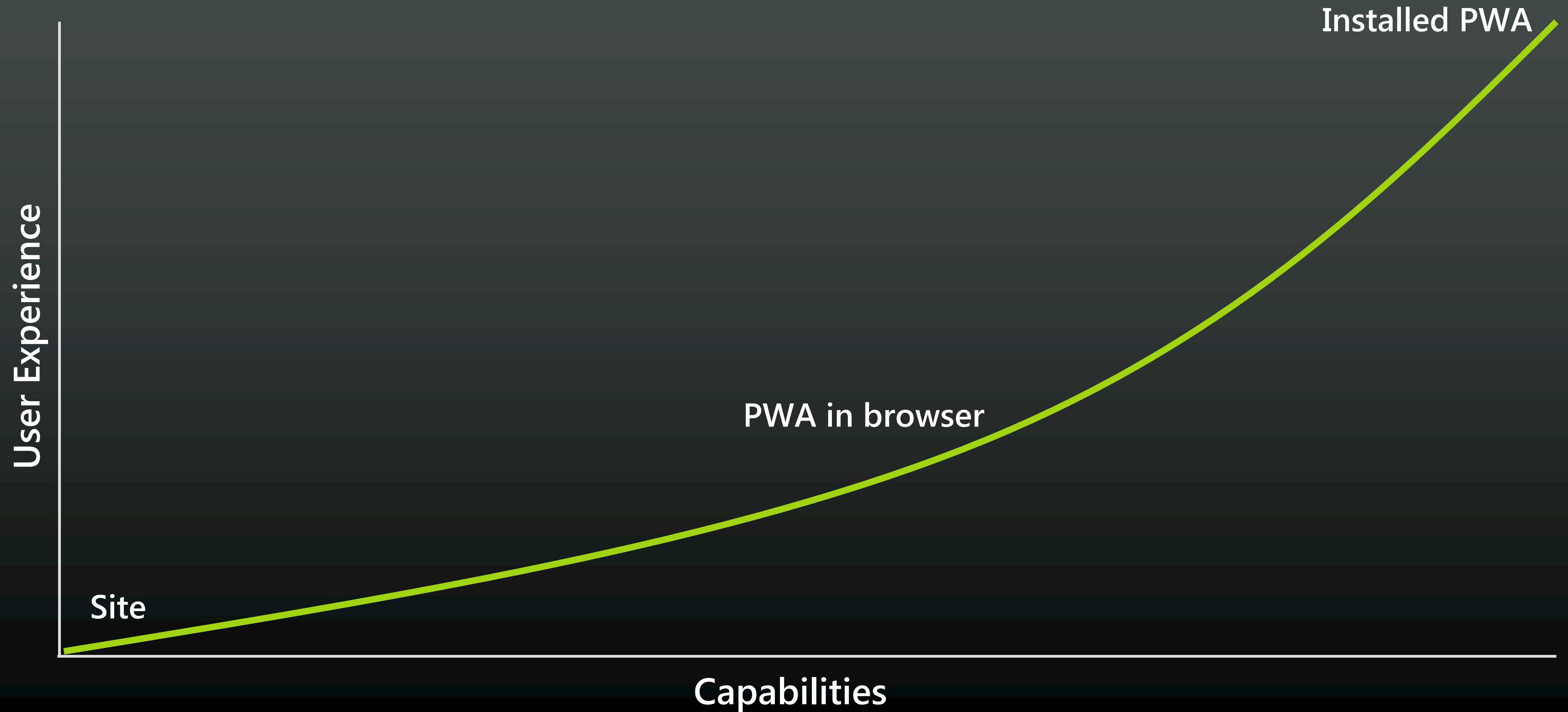
4. `icons`

Suggested sizes: 48x48, 72x72, 96x96, 144x144, 192x192, and 512x512

Enhance the experience



Enhance the experience



Let's talk about Service Worker



Registering a Service Worker

```
if ( "serviceWorker" in navigator ) {  
    navigator.serviceWorker.register( "/serviceworker.min.js" );  
}
```

Registering a Service Worker

```
if ( "serviceWorker" in navigator ) {  
  navigator.serviceWorker.register( "/serviceworker.min.js" );  
}
```


Registering a Service Worker

```
if ( "serviceWorker" in navigator ) {  
    navigator.serviceWorker.register( "/serviceworker.min.js" );  
}
```

Registering a Service Worker

```
if ( "serviceWorker" in navigator ) {  
  navigator.serviceWorker.register( "/serviceworker.min.js" );  
}
```

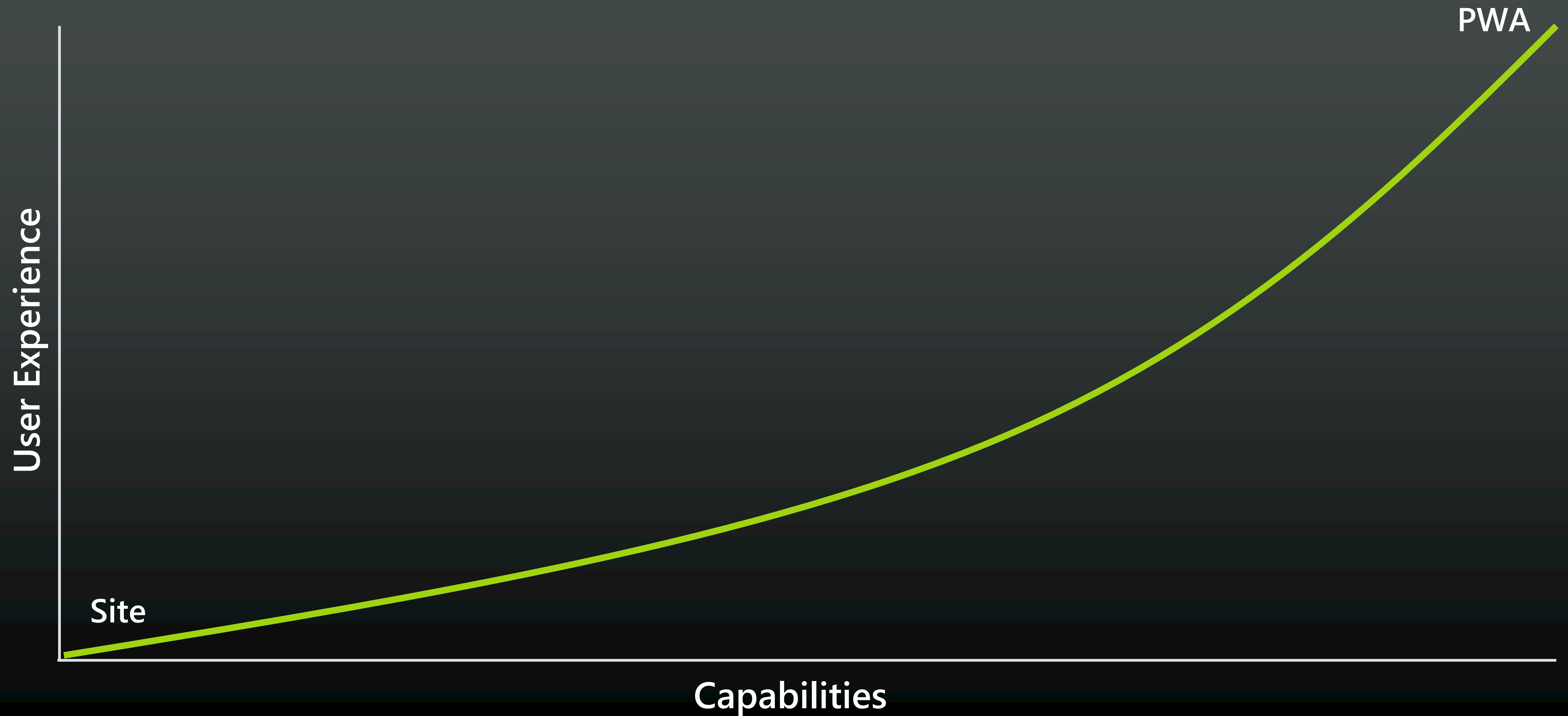
Path is important!



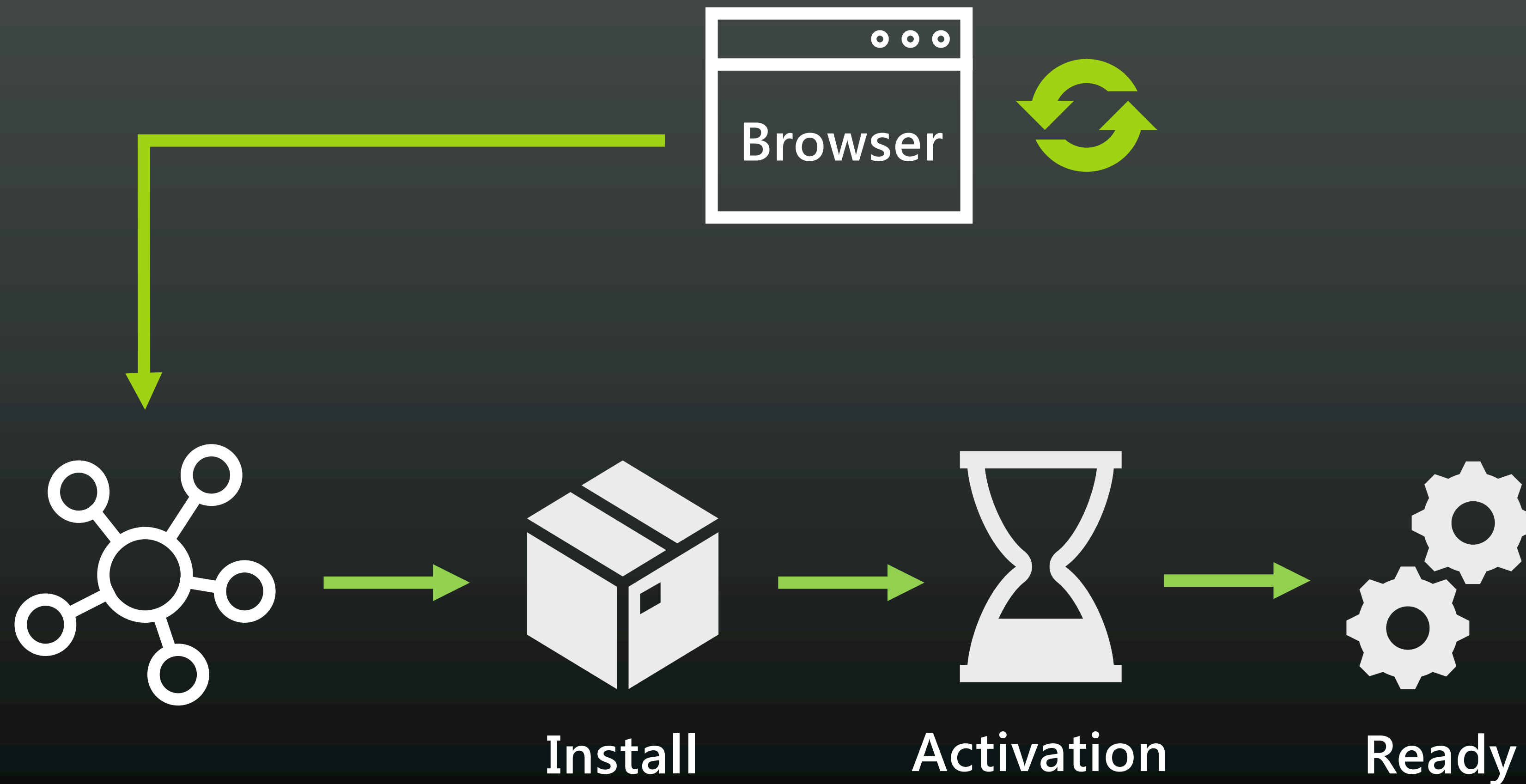
Registering a Service Worker

```
if ( "serviceWorker" in navigator ) {  
  navigator.serviceWorker.register( "/serviceworker.min.js" )  
    .then(function( registration ){  
      console.log( "Success!", registration.scope );  
    })  
    .catch(function( error ){  
      console.error( "Failure!" , error );  
    });  
}
```


Enhance the experience



The Service Worker Lifecycle



Listening for these events

```
self.addEventListener( "install", function( event ){  
  console.log( "installing" );  
});
```

Listening for these events

```
self.addEventListener( "install", function( event ){  
    console.log( "installing" );  
});
```


Listening for these events

```
self.addEventListener( "install", function( event ){  
  console.log( "installing" );  
});
```

Listening for these events

```
self.addEventListener( "install", function( event ){  
  console.log( "installing" );  
});
```

Let's make a Service Worker!

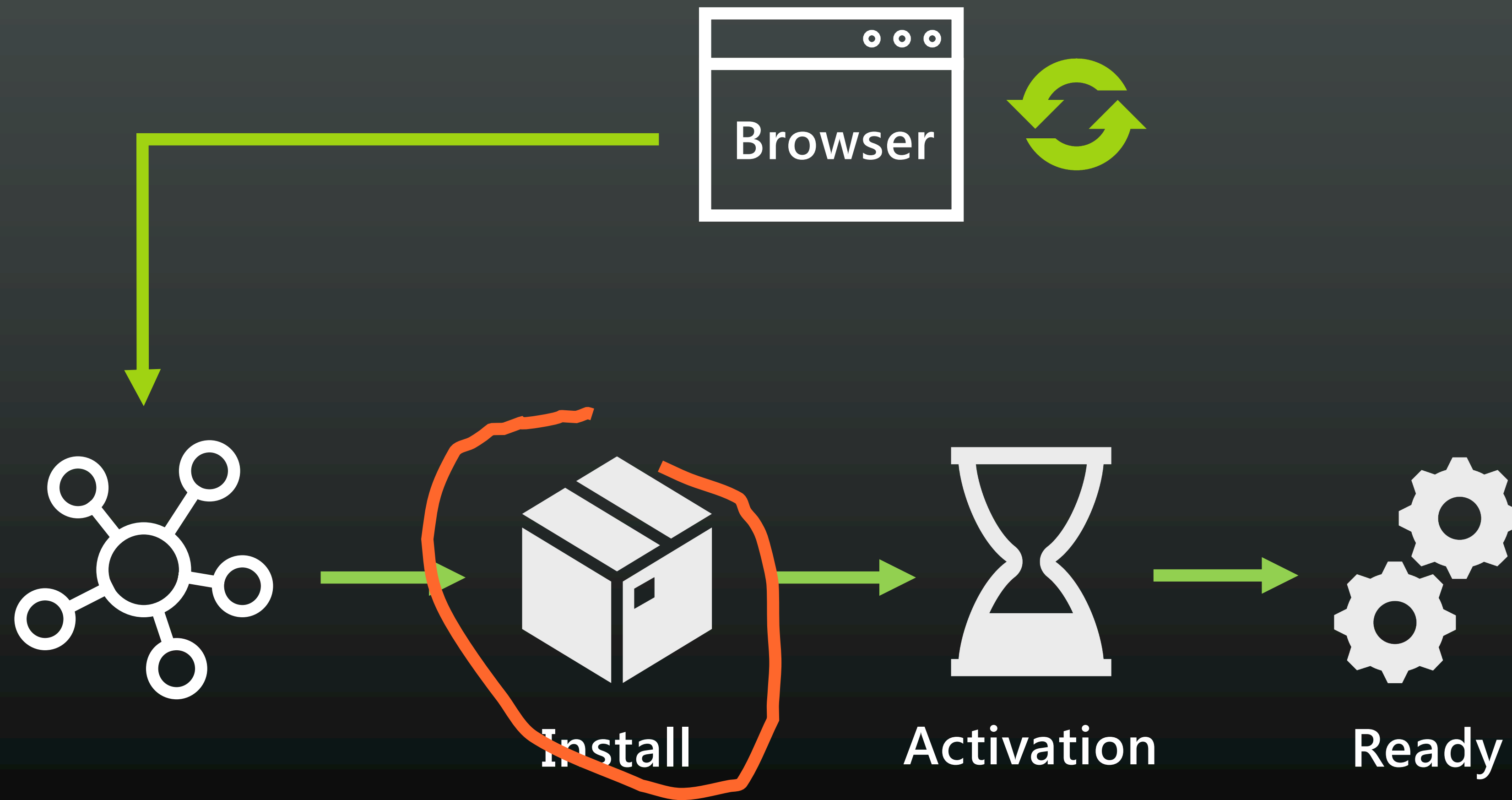
- Create a new file for your service worker
- Register the Service Worker
`navigator.serviceWorker.register(path)`
- Log to the console from the following events:
 - install
 - activate

Yours should look similar

```
self.addEventListener( "install", function( event ){  
  console.log( "installing" );  
});
```

```
self.addEventListener( "activate", function( event ){  
  console.log( "activating" );  
});
```


The Service Worker Lifecycle



Preloading assets

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open("v1").then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```

Preloading assets

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open("v1").then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```

Preloading assets

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open( "v1" ).then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```


Preloading assets

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open("v1").then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```

Preloading assets

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open( "v1" ).then( function( cache ) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```

Let's refactor

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open( "v1" ).then( function( cache ) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```

Let's refactor

```
const VERSION = "v1";

self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open( VERSION ).then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );
});
```


Let's preload assets

- Leverage the install event to pre-load some assets
 - Load your page in a browser and see that the assets are loaded
 - Bump your version number and reload the page
 - What happened?

Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```

Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```

Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```


Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```

Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```

Clean up after yourself

```
const VERSION = "v1";

// install event

self.addEventListener( "activate", event => {
  // clean up stale caches
  event.waitUntil(
    caches.keys()
      .then( keys => {
        return Promise.all(
          keys.filter( key => {
            return ! key.startsWith( VERSION );
          } )
            .map( key => {
              return caches.delete( key );
            } )
        );
      } )
  );
});
```

Let's clean up

- Leverage the activate event to clear stale caches
 - Load your page in a browser
 - What happened?

Use the latest SW immediately

```
const VERSION = "v1";

self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open( VERSION ).then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js"
      ]);
    })
  );

  self.skipWaiting();
});
```


Claim any active clients

```
self.addEventListener( "activate", event => {  
  // clean up stale caches  
  event.waitUntil(  
    caches.keys()  
      .then( keys => {  
        return Promise.all(  
          keys.filter( key => {  
            return ! key.startsWith( VERSION );  
          })  
            .map( key => {  
              return caches.delete( key );  
            })  
        );  
      })  
  );  
});  
  
clients.claim();  
});
```

Look what happens

- Add `skipWaiting()` to your install event
 - Look at the DevTools and observe how the state of the Service Worker changes with and without this line of code.
 - What happened?
- Open your site in two tabs and add `clients.claim()` to your activate event
 - Look at the DevTools in each.
 - What happened?

Talk about the network



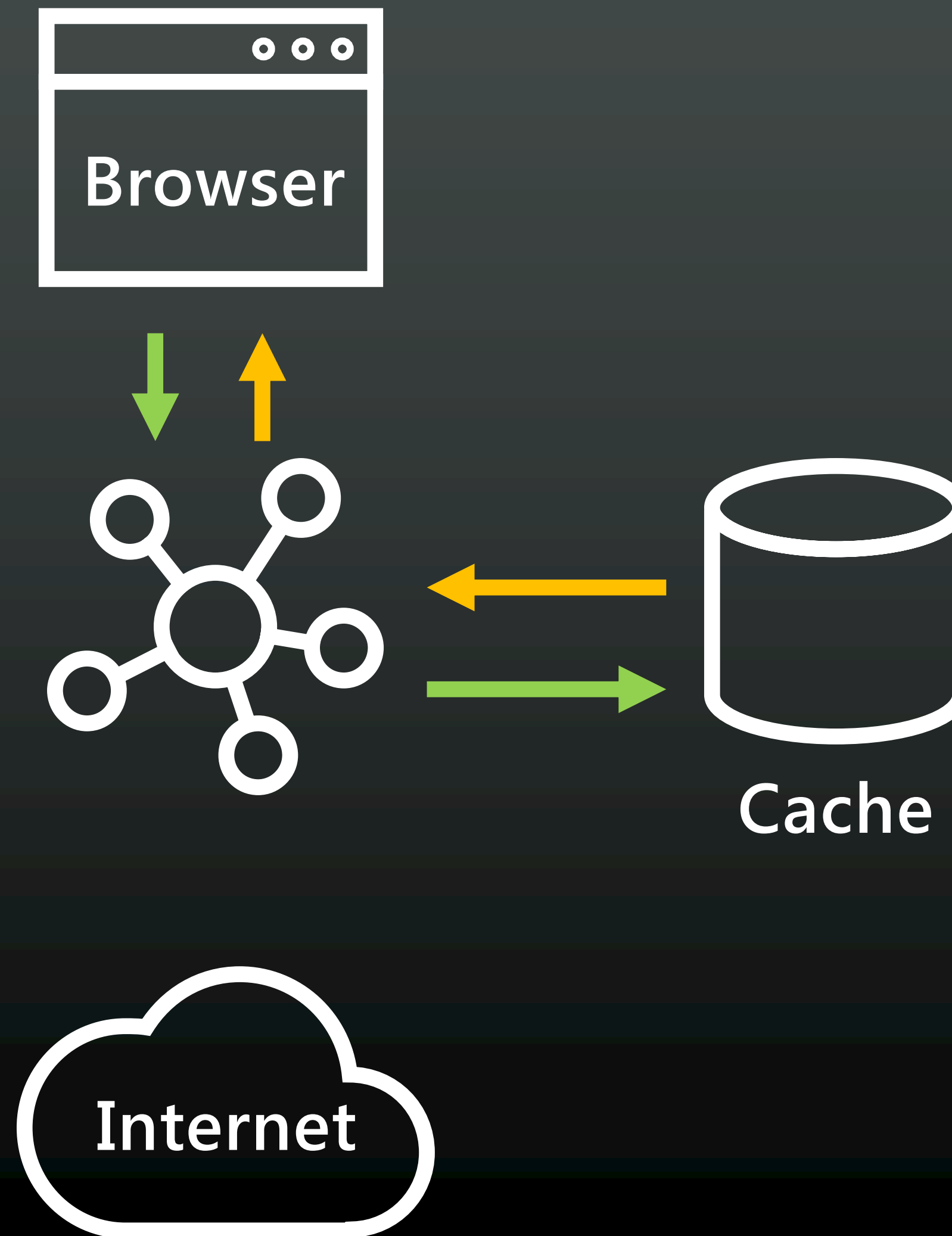
How requests are made



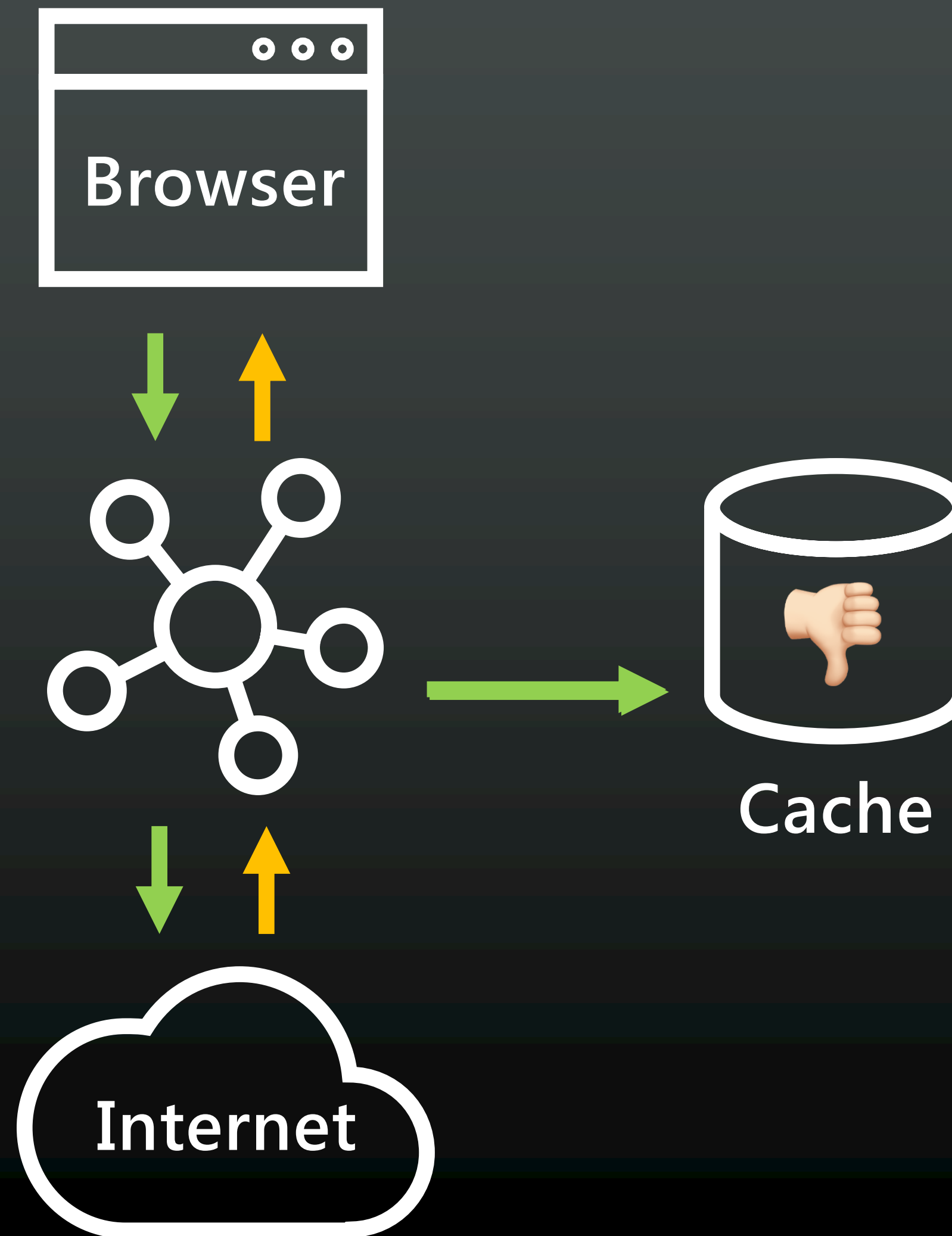
Along comes Service Worker



Along comes Service Worker



Along comes Service Worker



Intercepting requests

```
self.addEventListener( "fetch", function( event ){  
  console.log( "fetching" );  
});
```

Let's try it out

- Add a fetch event handler
 - Load your page in a browser
 - What happened?
- Instead of a string, log `event.request.url`
 - Load your page in a browser
 - What do you see?

We can issue our own fetch

```
self.addEventListener( "fetch", function( event ){  
  event.respondWith(  
    fetch( event.request )  
  );  
});
```


We can issue our own fetch

```
self.addEventListener( "fetch", function( event ){
  event.respondWith(
    fetch( event.request )
  );
});
```

What if the
request fails?

What if the request fails?

```
const VERSION      = "v1",  
      OFFLINE_PAGE = "offline.html";
```

What if the request fails?

```
self.addEventListener( "install", function( event ){
  event.waitUntil(
    caches.open("v1").then(function(cache) {
      return cache.addAll([
        "/css/main.css",
        "/js/main.js",
        OFFLINE_PAGE
      ]);
    })
  );
});
```

What if the request fails?

```
self.addEventListener( "fetch", function( event ){
  if ( event.request.mode === "navigate" ) {
    event.respondWith(
      fetch(event.request)
        .catch(error => {
          console.log( "Fetch failed; returning offline page." );
          return caches.match( OFFLINE_PAGE );
        })
    );
  }
});
```


What if the request fails?

```
self.addEventListener( "fetch", function( event ){
  if ( event.request.mode === "navigate" ) {
    event.respondWith(
      fetch(event.request)
        .catch(error => {
          console.log( "Fetch failed; returning offline page." );
          return caches.match( OFFLINE_PAGE );
        })
    );
  }
});
```

What if the request fails?

```
self.addEventListener( "fetch", function( event ){
  if ( event.request.mode === "navigate" ) {
    event.respondWith(
      fetch(event.request)
        .catch(error => {
          console.log( "Fetch failed; returning offline page." );
          return caches.match( OFFLINE_PAGE );
        })
    );
  }
});
```

What if the request fails?

```
self.addEventListener( "fetch", function( event ){
  if ( event.request.mode === "navigate" ) {
    event.respondWith(
      fetch(event.request)
        .catch(error => {
          console.log( "Fetch failed; returning offline page." );
          return caches.match( OFFLINE_PAGE );
        })
    );
  }
});
```

Let's try it out

- Add an `offline.html` page
- Pre-cache it during install
 - Remember to rev `VERSION`
- Add a fetch handler for navigations, providing the offline page as a fallback
- Turn off the network (once you know the SW is running) and see what happens

Caching strategies

- Network → cache → offline
- Cache → network → offline
- Cache vs network race → cache → offline
- etc.

Network, then cache

```
if ( event.request.mode === "navigate" ) {  
  event.respondWith(  
    fetch( event.request )  
      .catch(error => {  
        return caches.match( OFFLINE_PAGE );  
      })  
  );  
}
```

Network, then cache

```
self.addEventListener( "fetch", function( event ){  
  
    let request = event.request,  
        url = request.url;  
  
    // all the rest of the code  
  
}
```

Network, then cache

```
if ( event.request.mode === "navigate" ) {  
  event.respondWith(  
    fetch( event.request )  
      .catch(error => {  
        return caches.match( OFFLINE_PAGE );  
      })  
  );  
}
```

Network, then cache

```
if ( request.mode === "navigate" ) {  
  event.respondWith(  
    fetch( request )  
      .catch(error => {  
        return caches.match( OFFLINE_PAGE );  
      })  
  );  
}
```

Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request )
      .then( response => {
        event.waitUntil(
          caches.open( VERSION ).then( cache => {
            return cache.put( request, response );
          } )
        );
        return response.clone();
      } )
      .catch( error => {
        return caches.match( OFFLINE_PAGE );
      } )
  );
}
```


Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request )
      .then( response => {
        event.waitUntil(
          caches.open( VERSION ).then( cache => {
            return cache.put( request, response );
          } )
        );
        return response.clone();
      } )
      .catch( error => {
        return caches.match( OFFLINE_PAGE );
      } )
  );
}
```

Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request )
      .then( response => {
        event.waitUntil(
          caches.open( VERSION ).then( cache => {
            return cache.put( request, response );
          } )
        );
        return response.clone();
      } )
      .catch( error => {
        return caches.match( OFFLINE_PAGE );
      } )
  );
}
```

Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request )
      .then( response => {
        event.waitUntil(
          caches.open( VERSION ).then( cache => {
            return cache.put( request, response );
          } )
        );
        return response.clone();
      } )
      .catch( error => {
        return caches.match( OFFLINE_PAGE );
      } )
  );
}
```

Network, then cache

```
.catch(error => {  
  return caches.match( OFFLINE_PAGE );  
})
```

Network, then cache

```
.catch(error => {  
  return caches.match( request ).then( cached_result => {  
    if ( cached_result ) {  
      return cached_result;  
    }  
    return caches.match( OFFLINE_PAGE );  
  });  
})
```

Network, then cache

```
.catch(error => {  
  return caches.match( request ).then( cached_result => {  
    if ( cached_result ) {  
      return cached_result;  
    }  
    return caches.match( OFFLINE_PAGE );  
  });  
})
```


Network, then cache

```
.catch(error => {  
  return caches.match( request ).then( cached_result => {  
    if ( cached_result ) {  
      return cached_result;  
    }  
    return caches.match( OFFLINE_PAGE );  
  });  
})
```

Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request ).then( response => {
      event.waitUntil(
        caches.open( VERSION ).then( cache => {
          return cache.put( request, response );
        })
      );
      return response.clone();
    })
  ).catch(error => {
    return caches.match( request ).then( cached_result => {
      if ( cached_result ) { return cached_result; }
      return caches.match( OFFLINE_PAGE );
    });
  })
);
}
```

Caching strategies

- ✓ Network → cache → offline
- Cache → network → offline
- Cache vs network race → cache → offline
- etc.

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, response );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, response );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, response );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```


Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, reponse );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, response );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request ).then( response => {
          event.waitUntil(
            caches.open( VERSION ).then( cache => {
              return cache.put( request, reponse );
            }
          );
          return response.clone();
        })
      })
      .catch( new Response( "", {
        status: 408,
        statusText: "The server appears to be offline."
      })
    );
  });
}
```

Anyone see
an opportunity
for refactoring?

We cached a fetch twice

```
return fetch( request ).then( response => {  
  event.waitUntil(  
    caches.open( VERSION ).then( cache => {  
      return cache.put( request, response );  
    })  
  );  
  return response.clone();  
})
```

Let's make it a function

```
function cacheResponse( response ) {  
  event.waitUntil(  
    caches.open( VERSION ).then( cache => {  
      return cache.put( request, response );  
    })  
  );  
  return response.clone();  
}
```


Let's make it a function

```
function cacheResponse( response, event ) {  
  event.waitUntil(  
    caches.open( VERSION ).then( cache => {  
      return cache.put( event.request, response );  
    })  
  );  
  return response.clone();  
}
```

And we can use it like this

```
return fetch( request )  
  .then( response => cacheResponse( response, event ) )
```

Network, then cache

```
if ( request.mode === "navigate" ) {
  event.respondWith(
    fetch( request )
      .then( response => cacheResponse( response, event ) )
      .catch(error => {
        return caches.match( request ).then( cached_result => {
          if ( cached_result ) { return cached_result; }
          return caches.match( OFFLINE_PAGE );
        });
      })
  );
}
```

Cache first, then network

```
if ( /\.css$/.test(url) || /\.js$/.test(url) ) {
  event.respondWith(
    caches.match( request )
      .then( cached_result => {
        if ( cached_result ) { return cached_result; }
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          .catch( new Response( "", {
            status: 408,
            statusText: "The server appears to be offline."
          } ) )
      )
  );
}
);
}
```

Caching strategies

- ✓ Network → cache → offline
- ✓ Cache → network → offline
- Cache vs network race → cache → offline
- etc.

Who will win?

<https://git.io/v56s4>

```
document.addEventListener('DOMContentLoaded', function(event) {
  var networkDone = false;
  var networkRequest = fetch('weather.json').then(function(response) {
    return response.json();
  })
  .then(function(json) {
    networkDone = true;
    updatePage(json);
  });

  caches.match('weather.json').then(function(response) {
    if (! response) throw Error('No data');
    return response.json();
  })
  .then(function(json) {
    if (!networkDone) updatePage(json);
  })
  .catch(function() { return networkRequest; })
  .catch(function() { console.log('We have nothing. '); })
  .then(hideLoading);
});
```


Who will win?

<https://git.io/v56s4>

```
document.addEventListener('DOMContentLoaded', function(event) {
  var networkDone = false;
  var networkRequest = fetch('weather.json').then(function(response) {
    return response.json();
  })
  .then(function(json) {
    networkDone = true;
    updatePage(json);
  });

  caches.match('weather.json').then(function(response) {
    if (! response) throw Error('No data');
    return response.json();
  })
  .then(function(json) {
    if (!networkDone) updatePage(json);
  })
  .catch(function() { return networkRequest; })
  .catch(function() { console.log('We have nothing. '); })
  .then(hideLoading);
});
```

Who will win?

<https://git.io/v56s4>

```
document.addEventListener('DOMContentLoaded', function(event) {
  var networkDone = false;
  var networkRequest = fetch('weather.json').then(function(response) {
    return response.json();
  })
  .then(function(json) {
    networkDone = true;
    updatePage(json);
  });

  caches.match('weather.json').then(function(response) {
    if (!response) throw Error('No data');
    return response.json();
  })
  .then(function(json) {
    if (!networkDone) updatePage(json);
  })
  .catch(function() { return networkRequest; })
  .catch(function() { console.log('We have nothing. '); })
  .then(hideLoading);
});
```

Who will win?

<https://git.io/v56s4>

```
document.addEventListener('DOMContentLoaded', function(event) {
  var networkDone = false;
  var networkRequest = fetch('weather.json').then(function(response) {
    return response.json();
  })
  .then(function(json) {
    networkDone = true;
    updatePage(json);
  });

  caches.match('weather.json').then(function(response) {
    if (! response) throw Error('No data');
    return response.json();
  })
  .then(function(json) {
    if (!networkDone) updatePage(json);
  })
  .catch(function() { return networkRequest; })
  .catch(function() { console.log('We have nothing. '); })
  .then(hideLoading);
});
```

Let's discuss

- In what scenarios would these different caching strategies be most appropriate?
- Are there other strategies you'd like to discuss?
- Do you want to add these caching strategies to your site now?

Could we *save* data?

Look at the connection

```
let slow_connection = false,  
    save_data       = false;  
  
function testConnection() {  
  // only test every minute  
  if ( last_tested &&  
      Date.now() < last_tested + ( 60 * 1000 ) ) { return; }  
  if ( 'connection' in navigator ) {  
    slow_connection = ( navigator.connection.downlink < 0.5 );  
    save_data = navigator.connection.saveData;  
    last_tested = Date.now();  
  }  
}
```


Look at the connection

```
let slow_connection = false,  
    save_data       = false;  
  
function testConnection() {  
  // only test every minute  
  if ( last_tested &&  
      Date.now() < last_tested + ( 60 * 1000 ) ) { return; }  
  if ( 'connection' in navigator ) {  
    slow_connection = ( navigator.connection.downlink < 0.5 );  
    save_data = navigator.connection.saveData;  
    last_tested = Date.now();  
  }  
}
```

Look at the connection

```
let slow_connection = false,
    save_data       = false;

function testConnection() {
  // only test every minute
  if ( last_tested &&
        Date.now() < last_tested + ( 60 * 1000 ) ) { return; }
  if ( 'connection' in navigator ) {
    slow_connection = ( navigator.connection.downlink < 0.5 );
    save_data = navigator.connection.saveData;
    last_tested = Date.now();
  }
}
```

Look at the connection

```
let slow_connection = false,  
    save_data       = false;  
  
function testConnection() {  
  // only test every minute  
  if ( last_tested &&  
      Date.now() < last_tested + ( 60 * 1000 ) ) { return; }  
  if ( 'connection' in navigator ) {  
    slow_connection = ( navigator.connection.downlink < 0.5 );  
    save_data = navigator.connection.saveData;  
    last_tested = Date.now();  
  }  
}
```

Look at the connection

```
self.addEventListener( "fetch", function( event ){  
  
  testConnection();  
  
  let request = event.request,  
      url      = request.url;  
  
  ...  
  
});
```

Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```


Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

Use that information

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

Dynamic images? Yes please!

```
const VERSION      = "v2",  
      OFFLINE_PAGE = "offline.html",  
      SVG_OFFLINE  = '<svg ...></svg>',  
      SVG_SLOW     = '<svg ...></svg>';
```

Dynamic images? Yes please!

```
function newSVGResponse( svg ) {  
  return new Response( svg, {  
    headers: { 'Content-Type': 'image/svg+xml' }  
  });  
}
```


An SVG for your troubles

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            // Respond with an "offline" image
          );
      } else {
        // Respond with a "saving data" image
      }
    })
  );
}
```

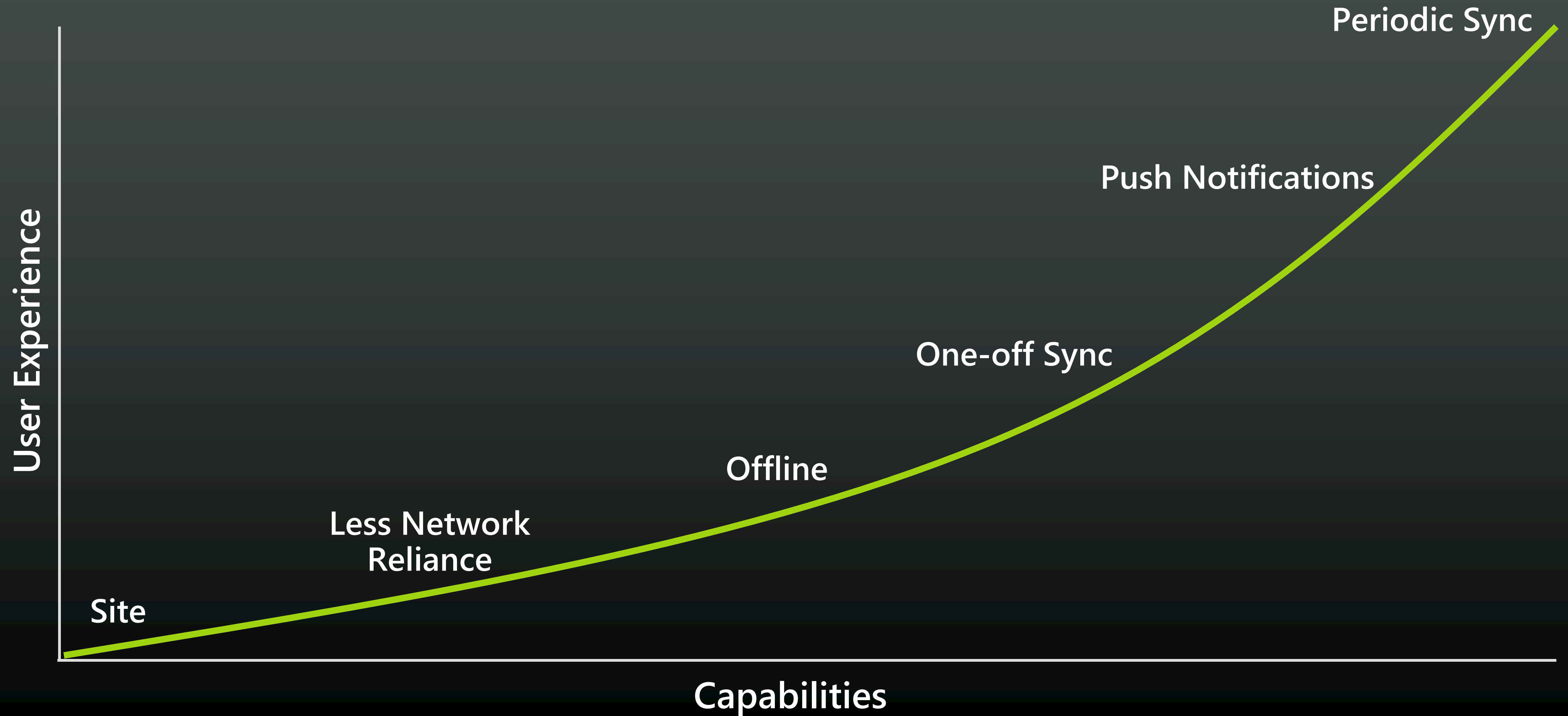
An SVG for your troubles

```
else if ( request.headers.get("Accept").includes("image") ) {
  event.respondWith(
    caches.match( request ).then( cached_result => {
      // cached first
      if ( cached_result ) {
        return cached_result;
      }
      // fallback to network
      if ( ! slow_connection && ! save_data ) {
        return fetch( request )
          .then( response => cacheResponse( response, event ) )
          // fail
          .catch(
            () => newSVGResponse( SVG_OFFLINE )
          );
      } else {
        return newSVGResponse( SVG_SLOW );
      }
    })
  );
}
```

Let's discuss

- What other ways we could use Service Workers to improve the user experience?

Moar enhancements!



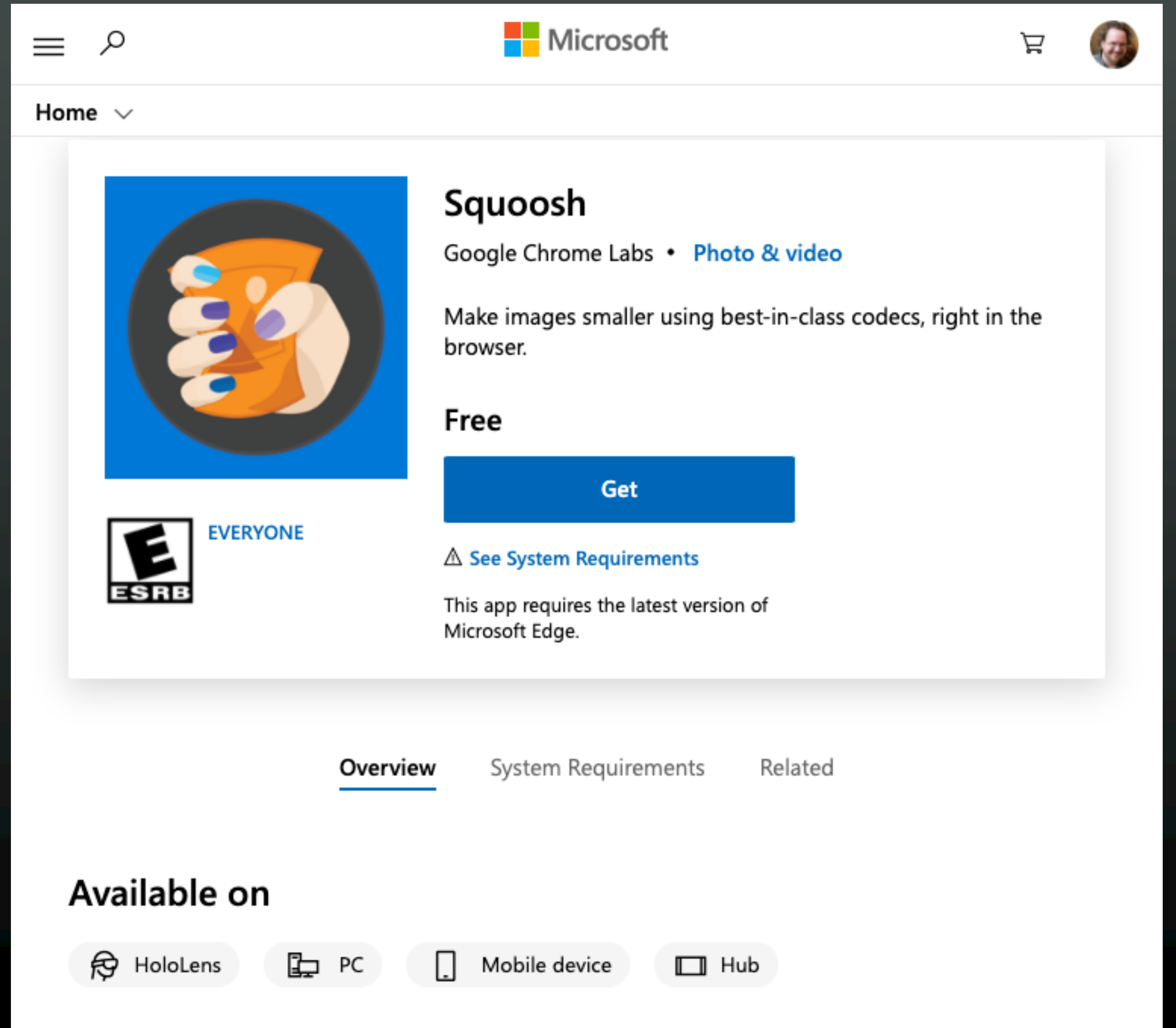
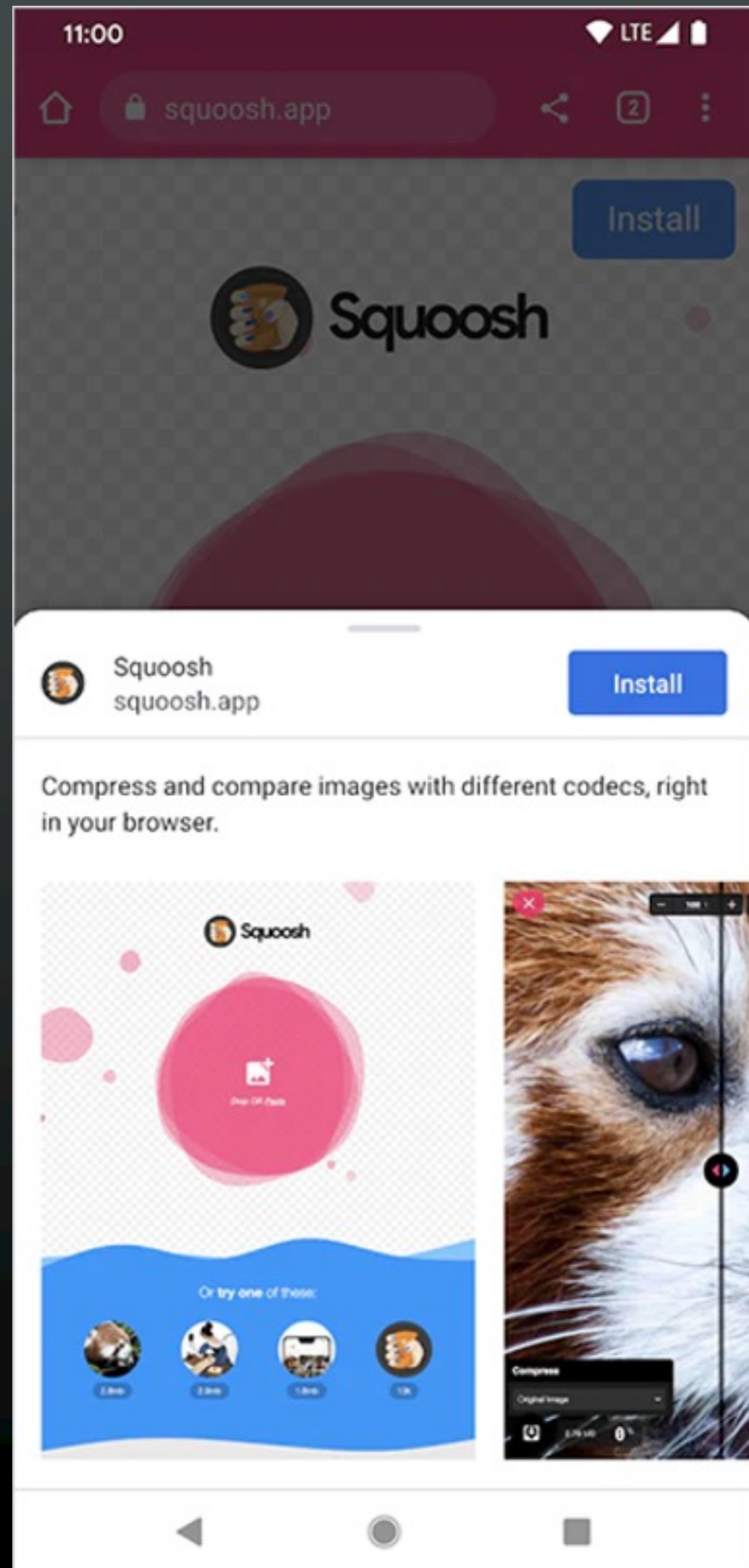
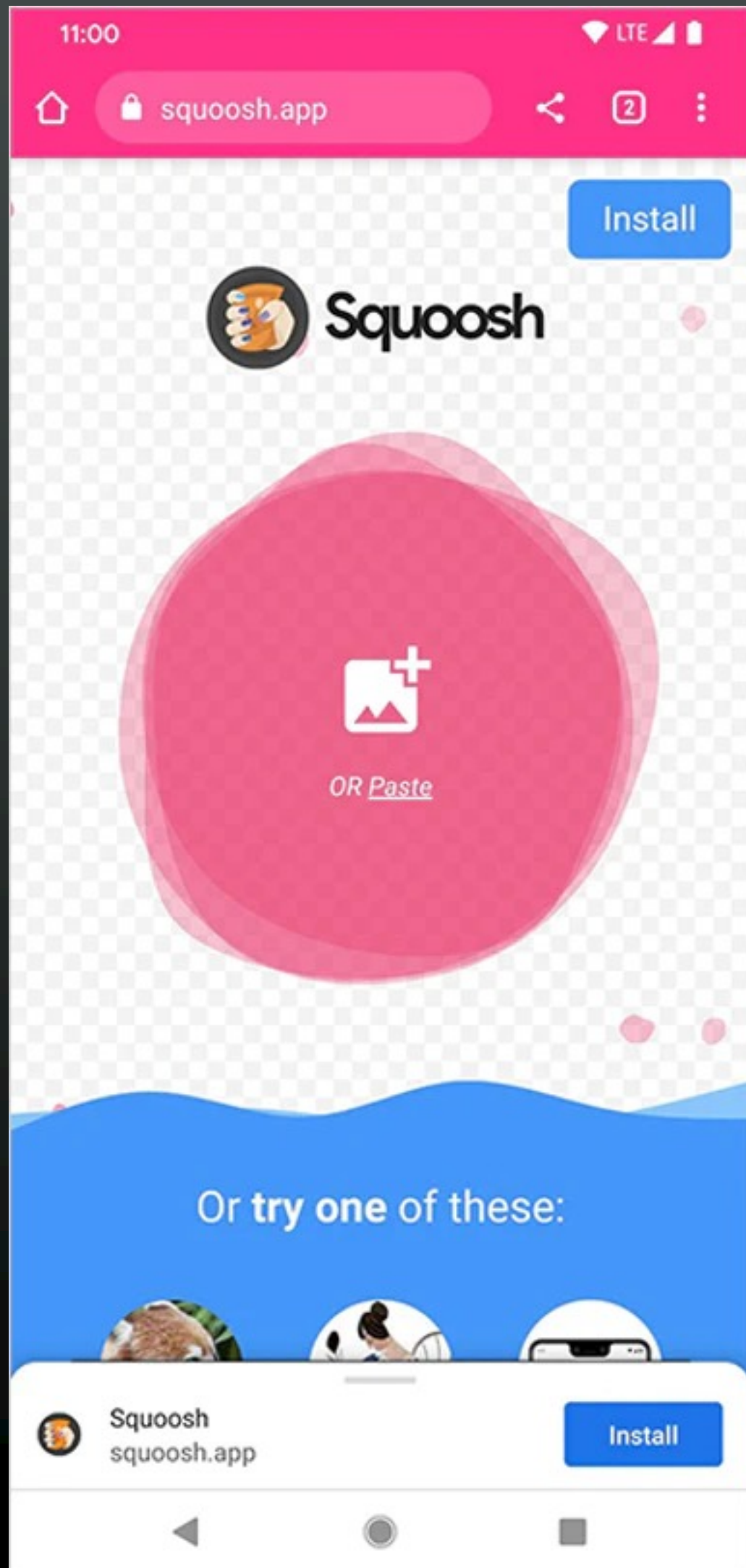
Bonus content!

- Rich Installation / Store Pages
- Shortcuts
- Share Target

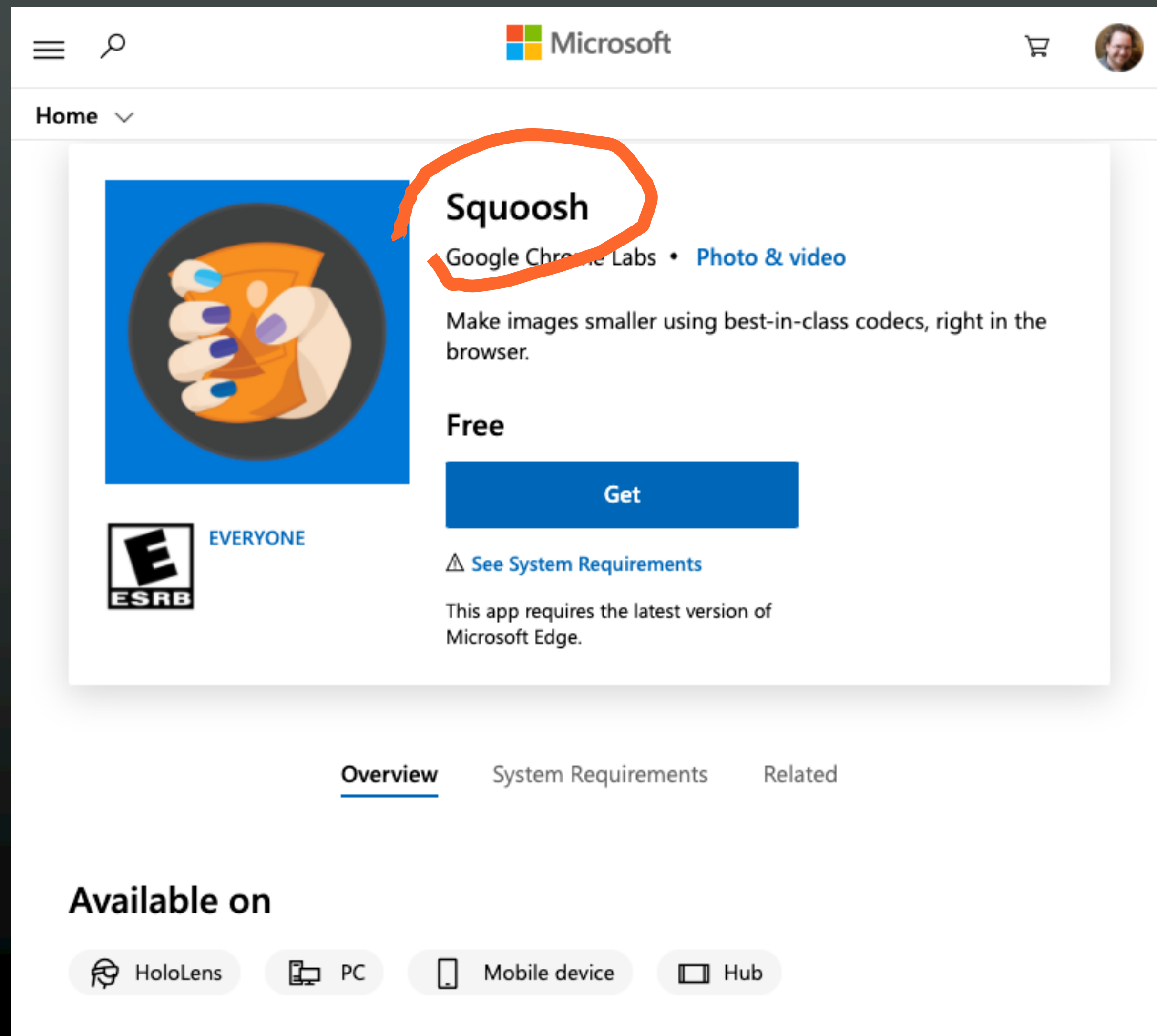
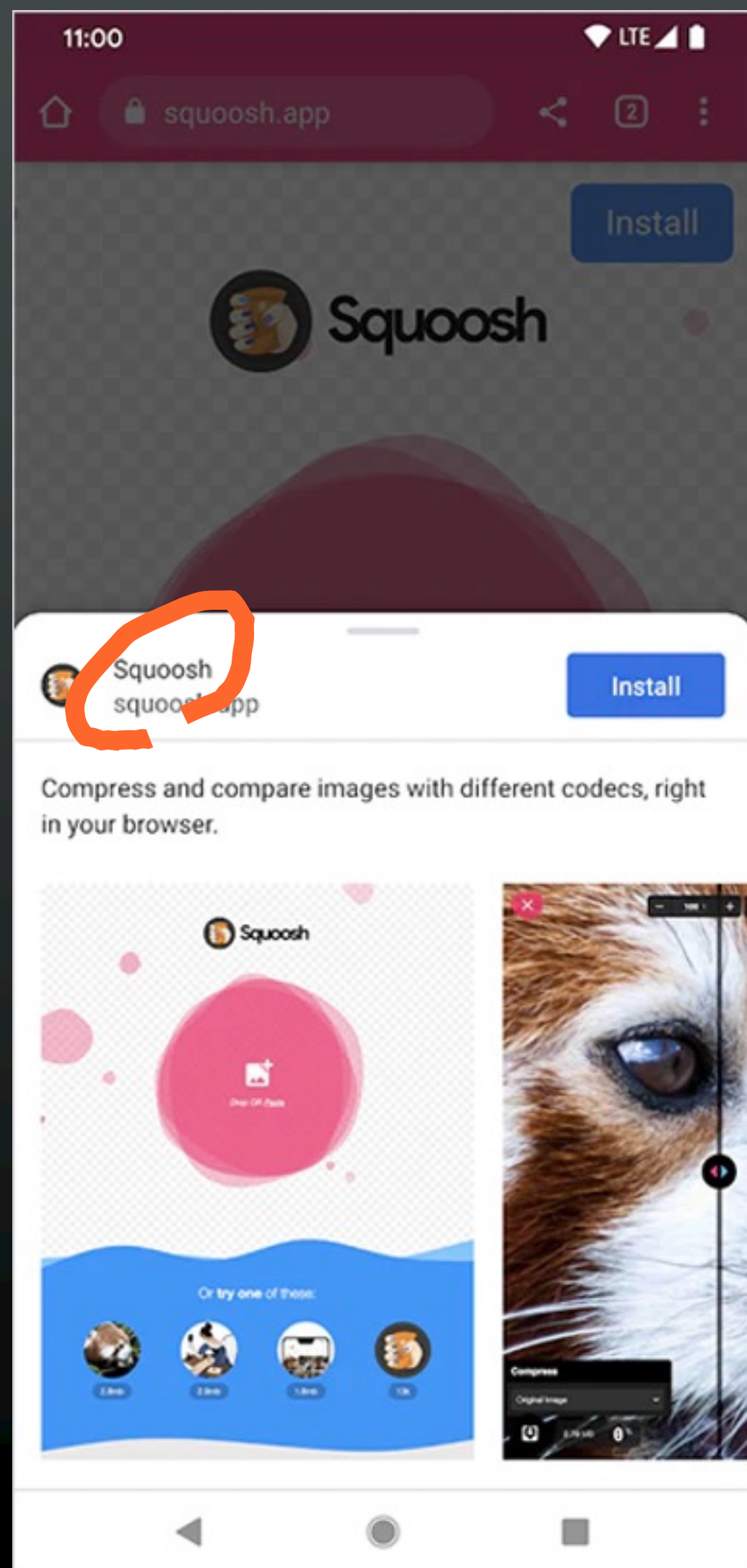
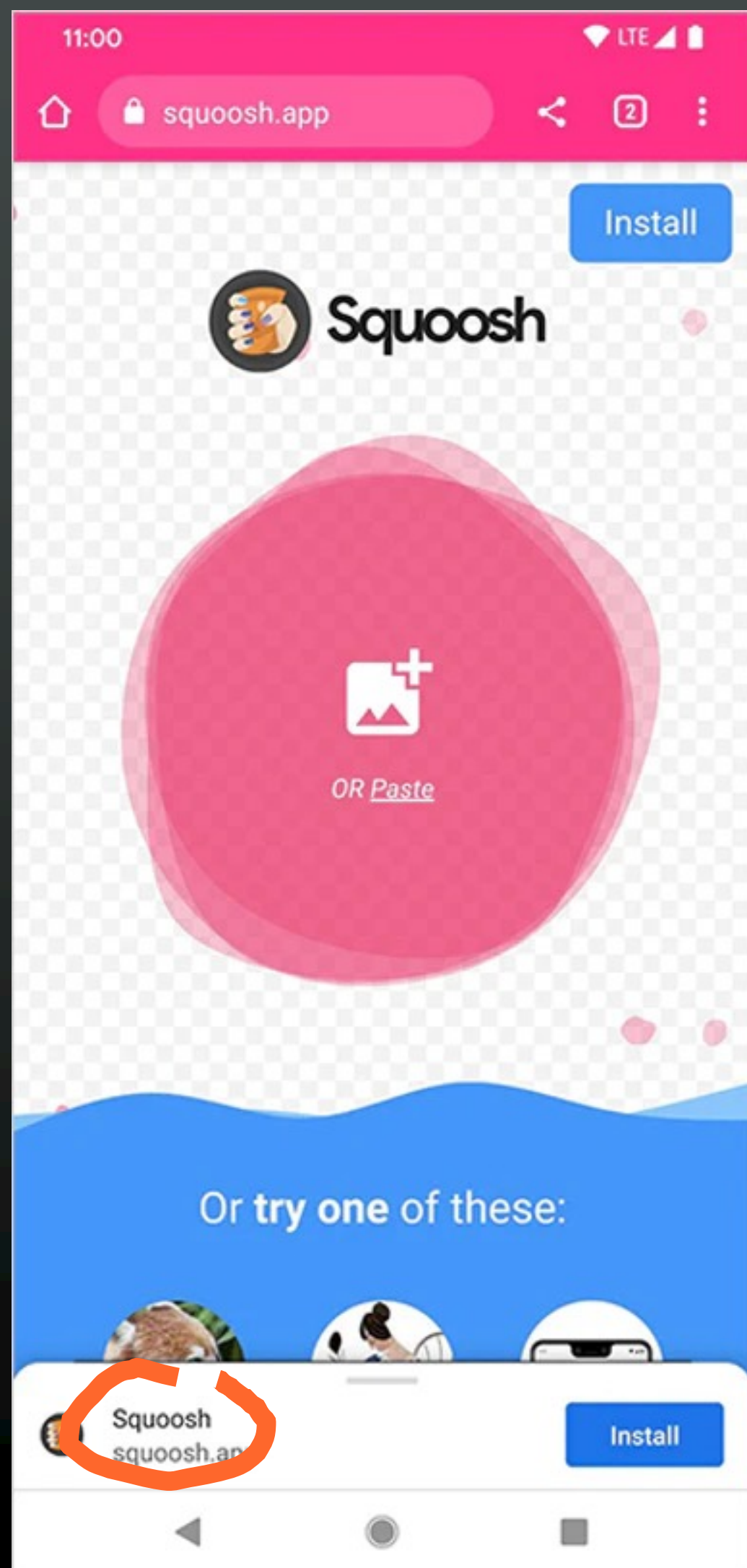
Bonus content!

- Rich Installation / Store Pages
- Shortcuts
- Share Target

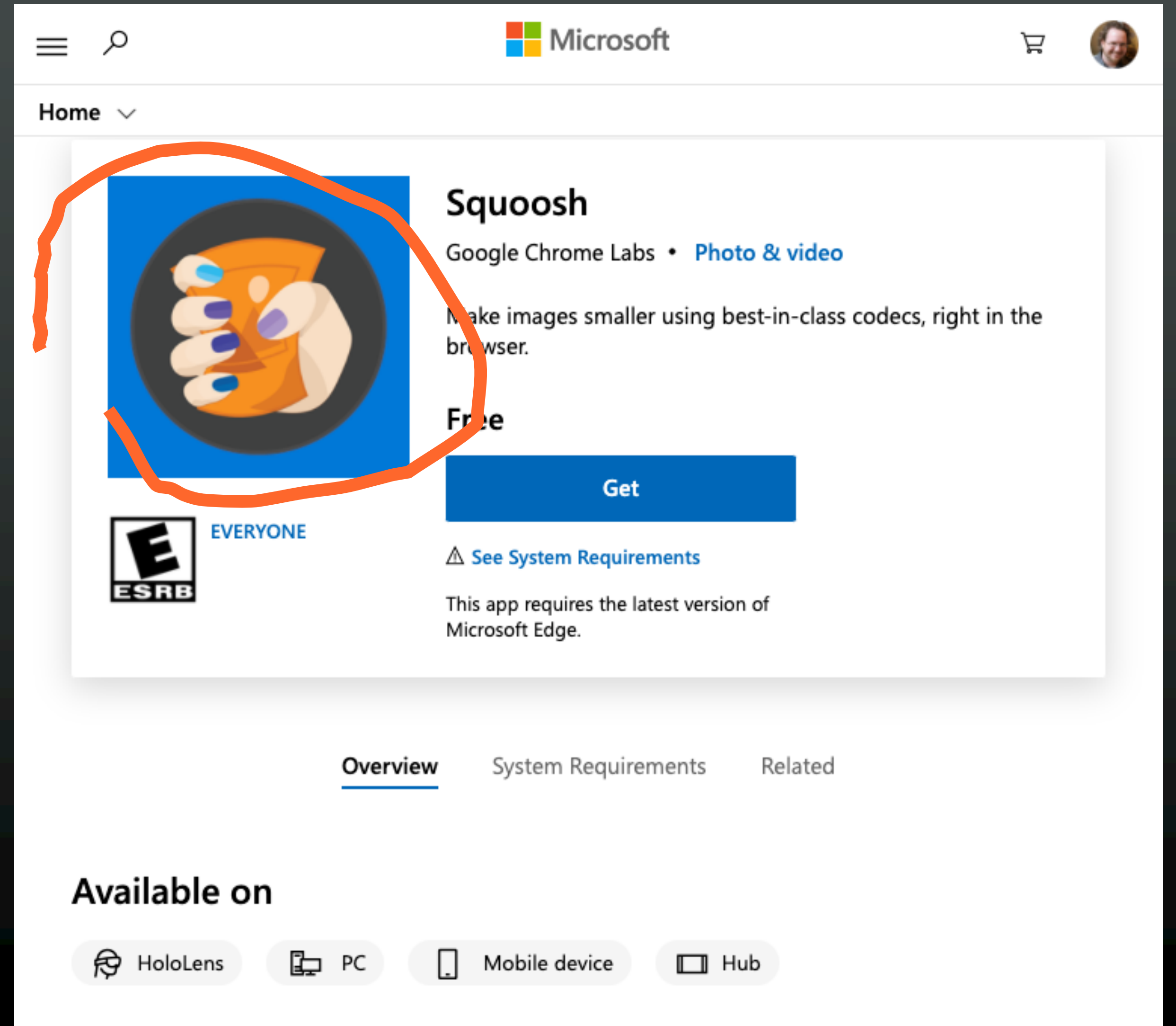
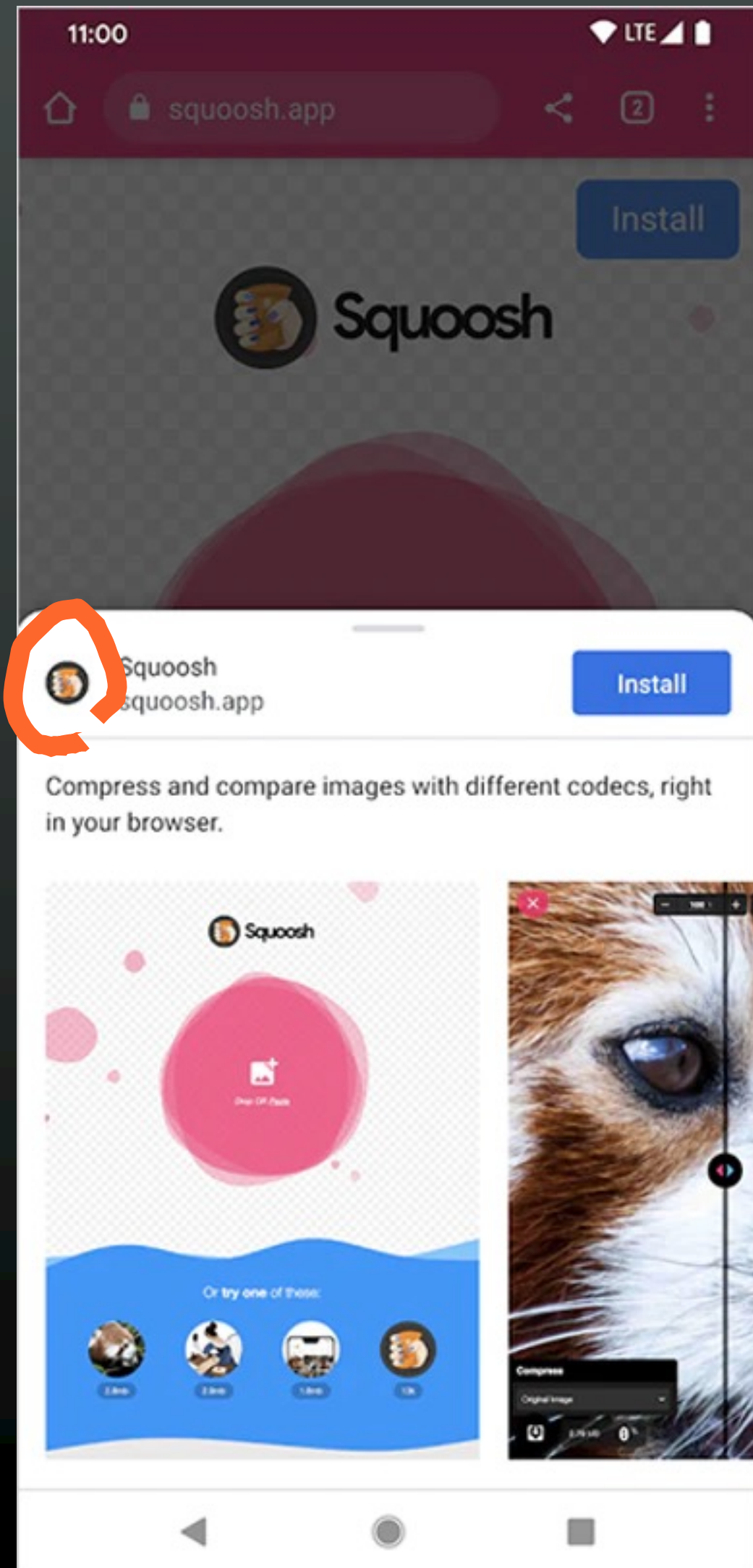
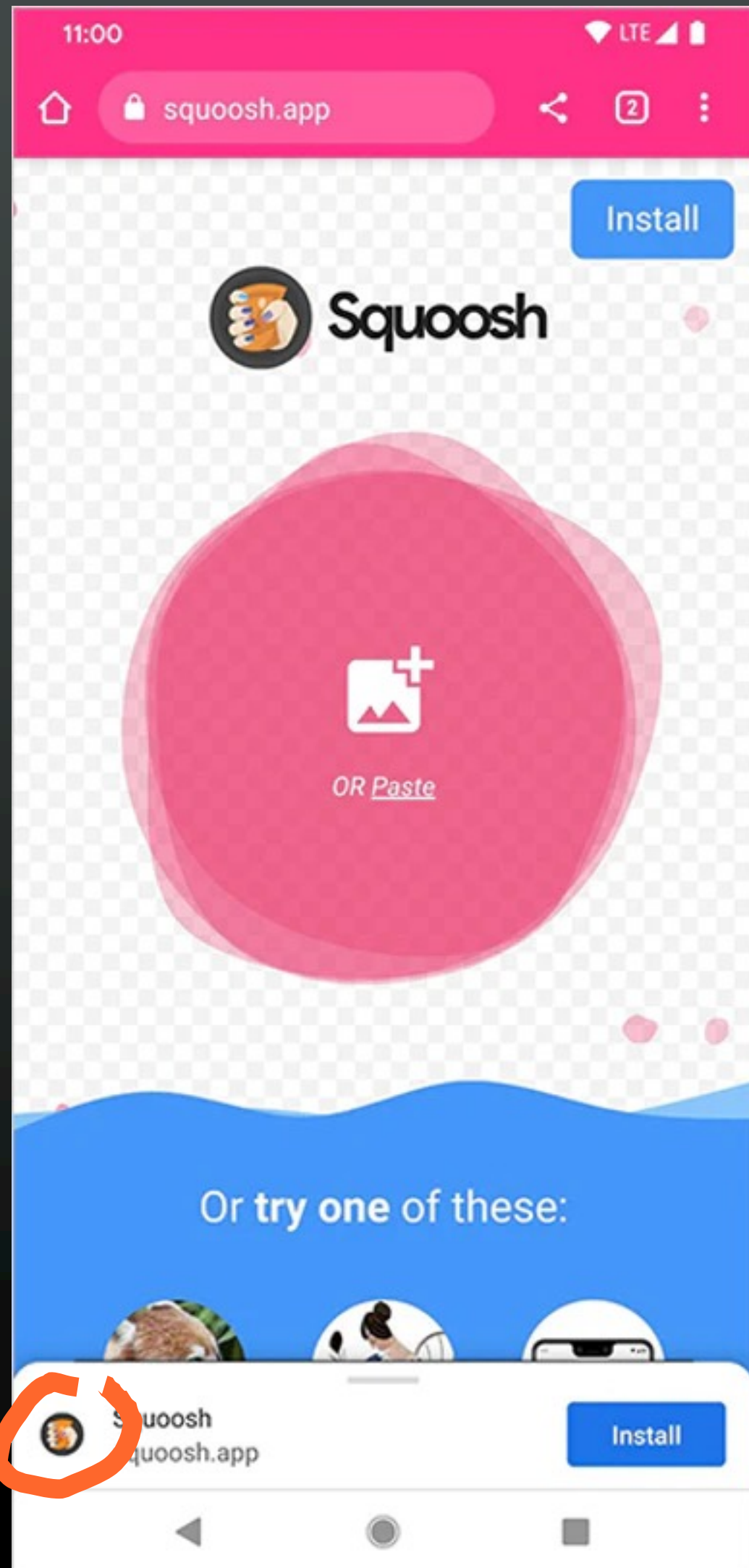
Marketing your PWA



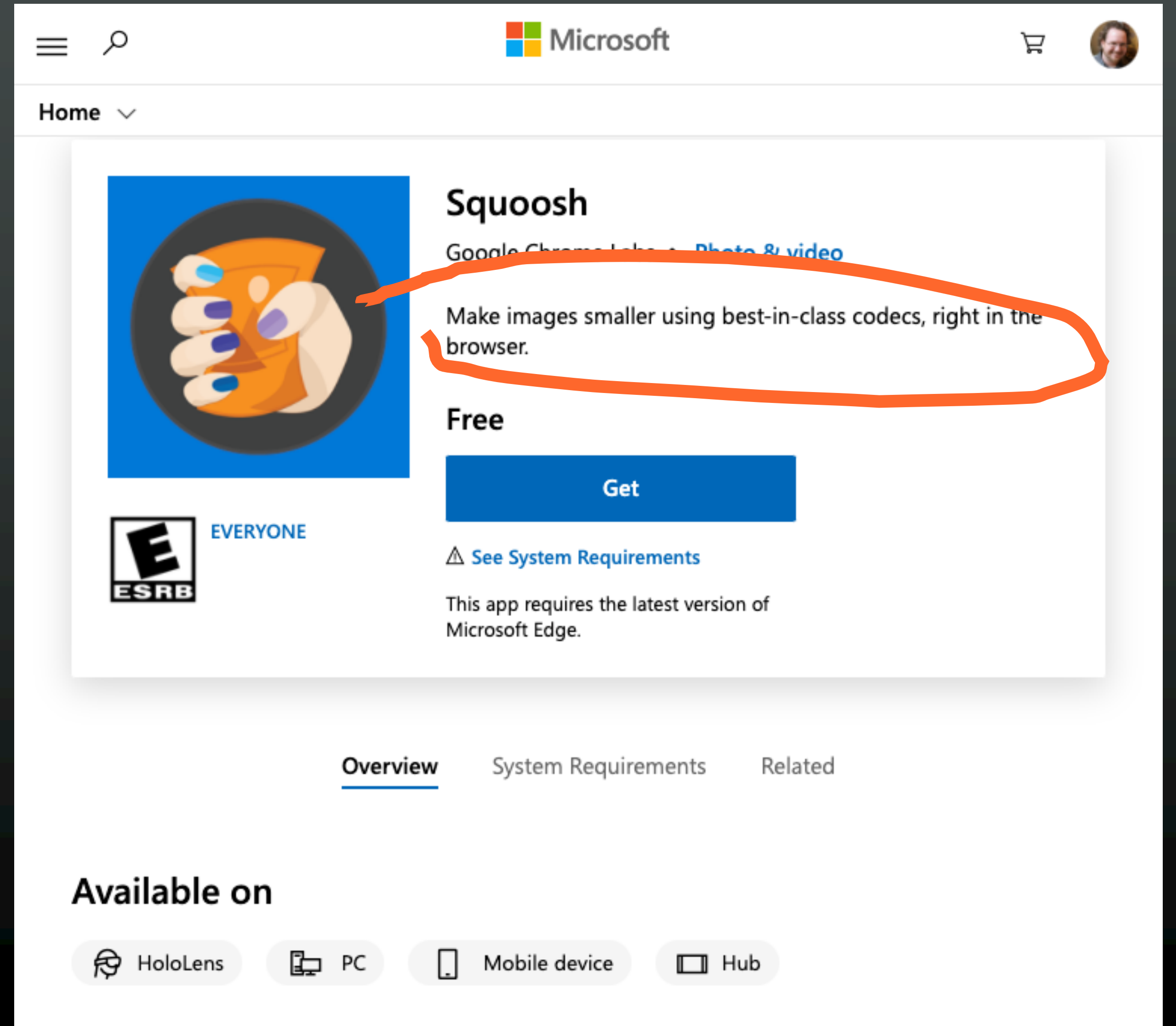
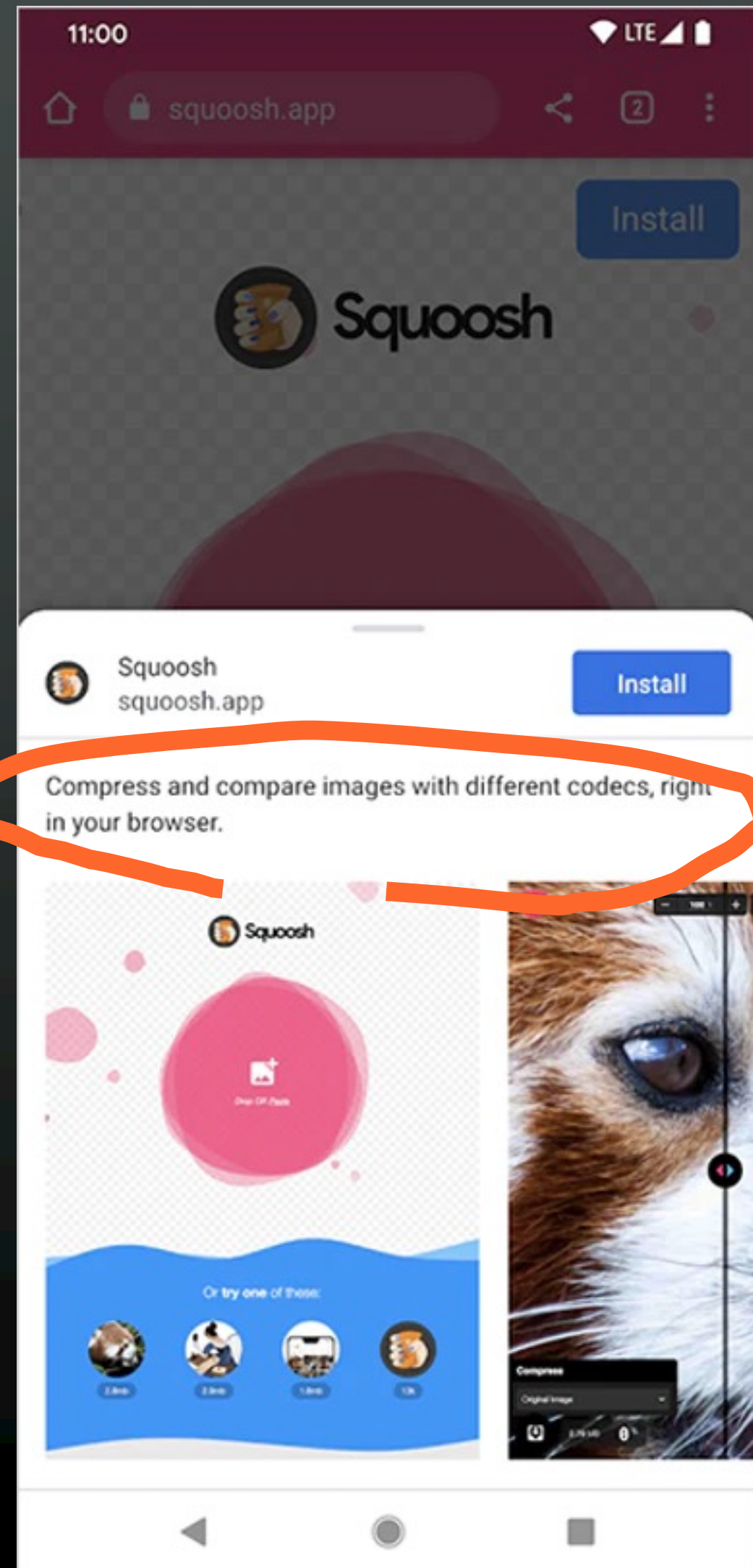
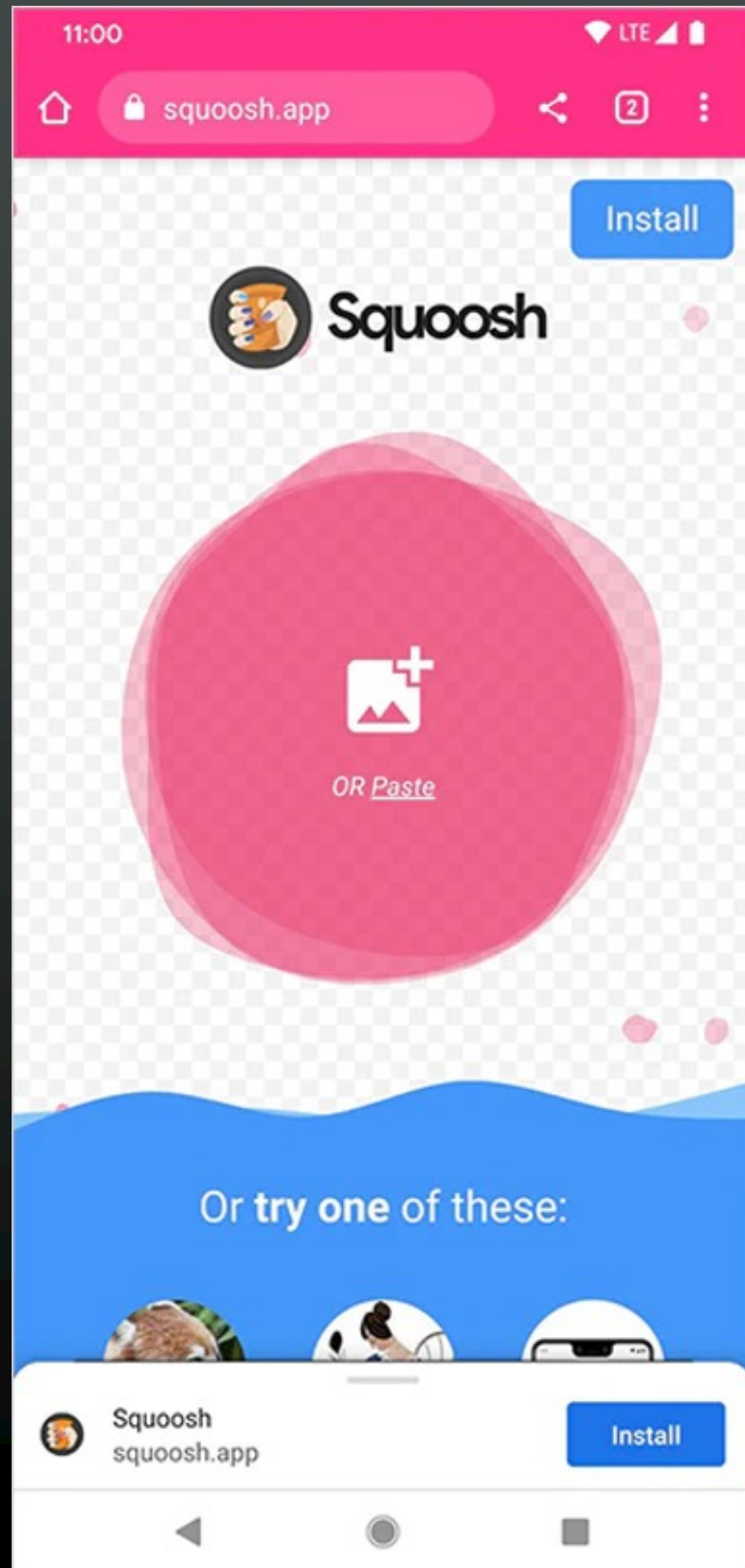
Name / Short Name



Icons



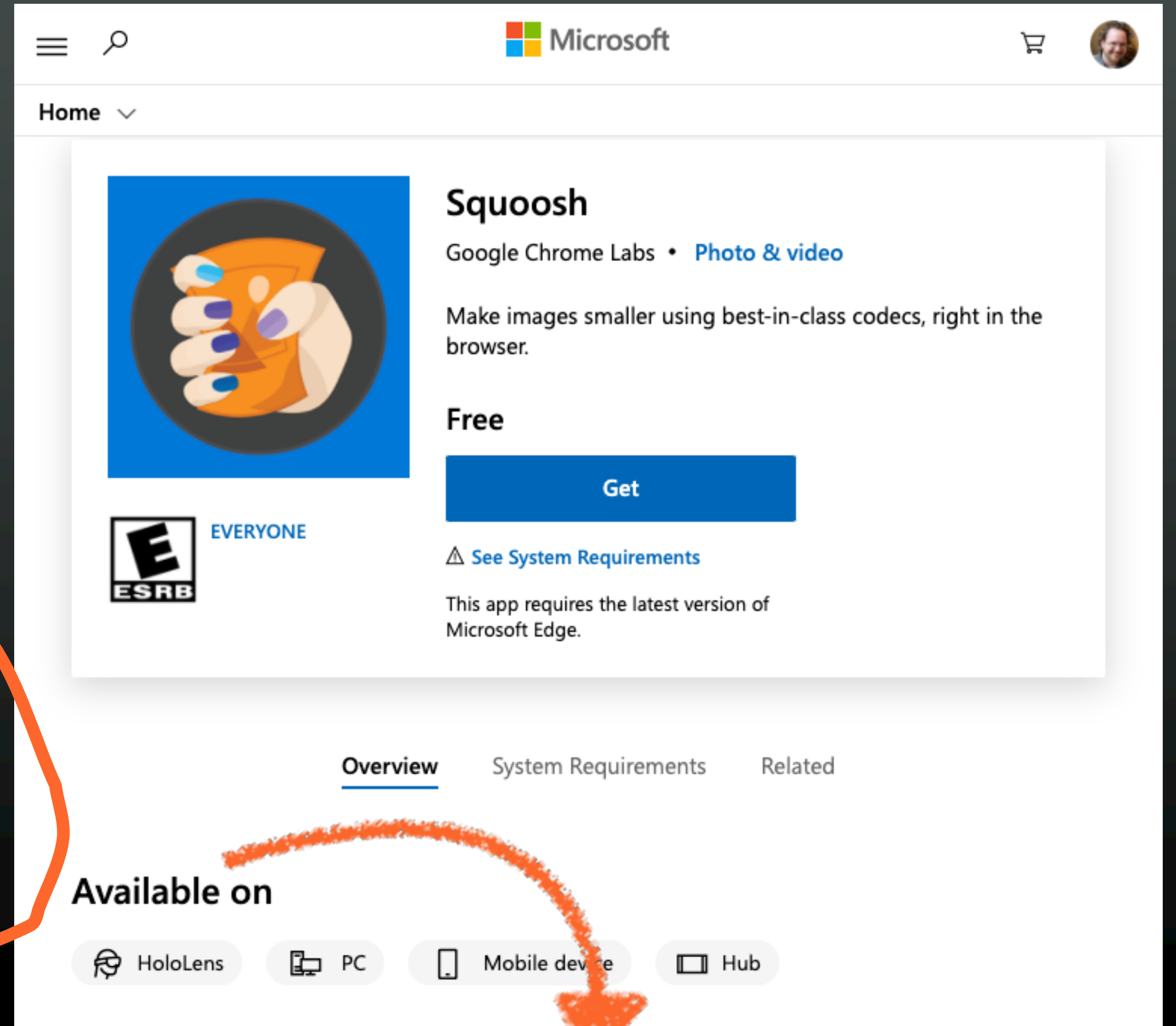
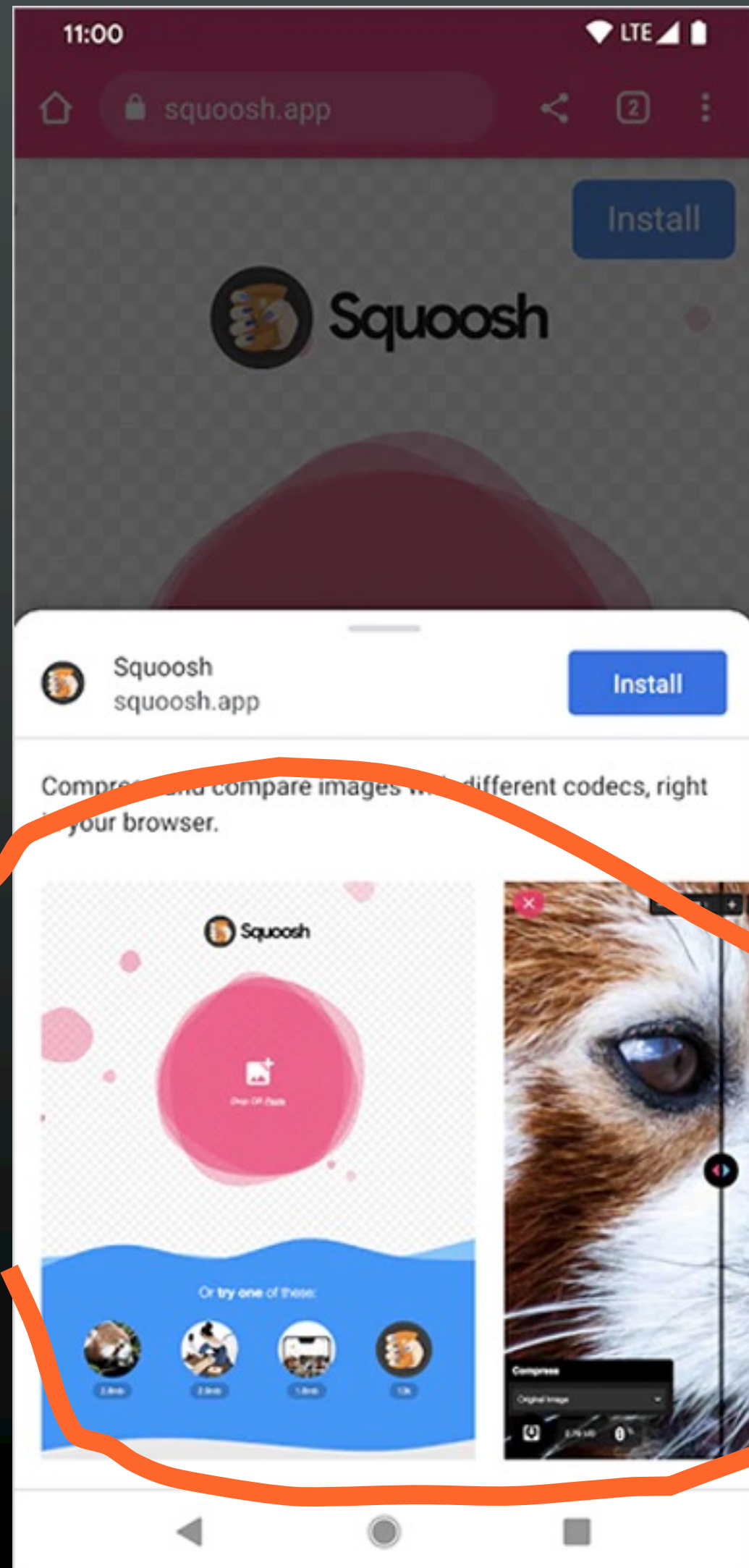
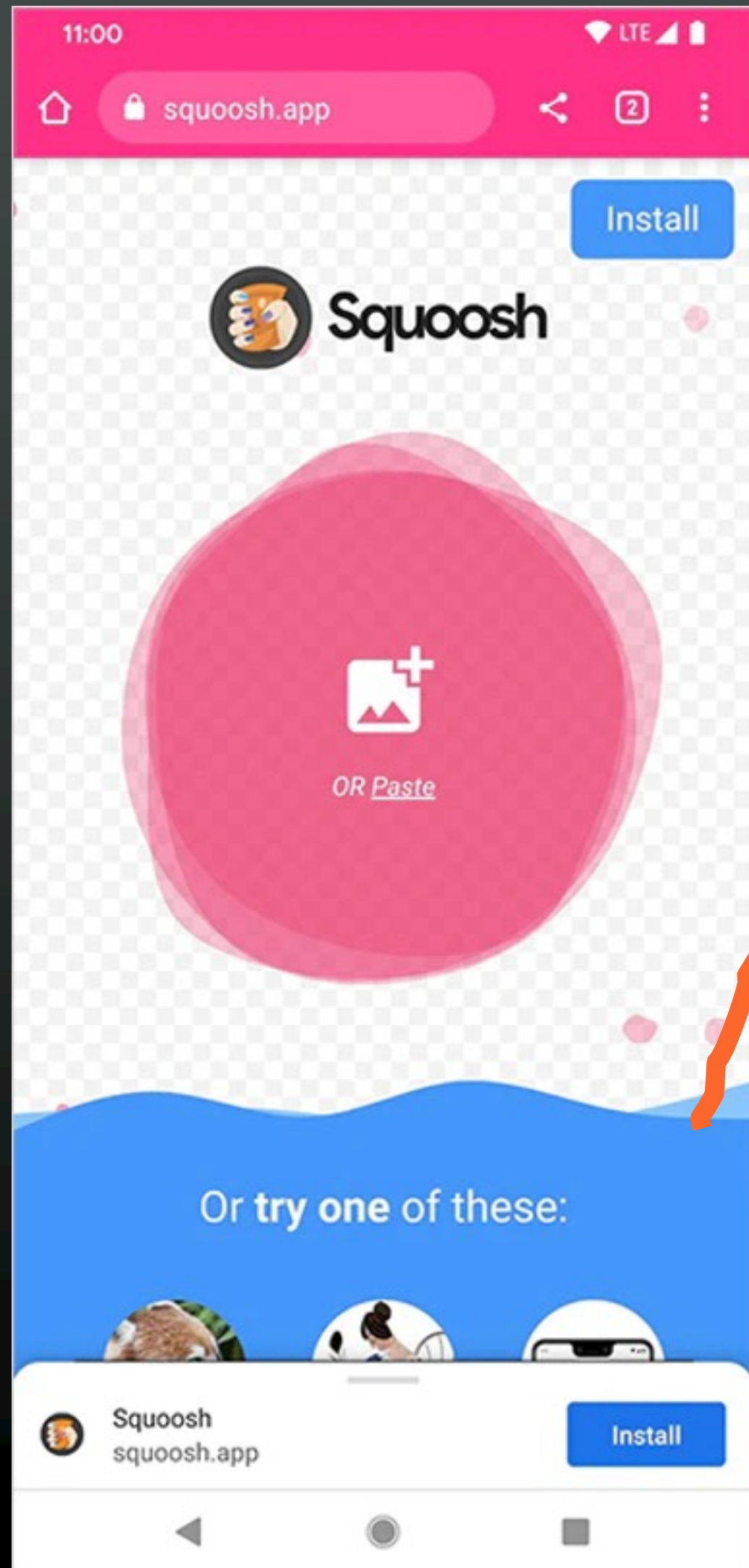
Description



Description

```
{  
  "name": "Squoosh",  
  "icons": [ ... ],  
  "description": "Make images smaller using best-in-class...",  
  "screenshots": [ ... ]  
}
```


Screenshots



Screenshots

```
{  
  "name": "Squoosh",  
  "icons": [ ... ],  
  "description": "Make images smaller using best-in-class...",  
  "screenshots": [ ... ]  
}
```

Screenshots need labels

```
{  
  "src": "/i/screenshots/main-ui.png",  
  "type": "image/png",  
  "sizes": "800x600",  
  "label": "The main Squoosh window. Pick an image to get started."  
}
```


Manifest App Information

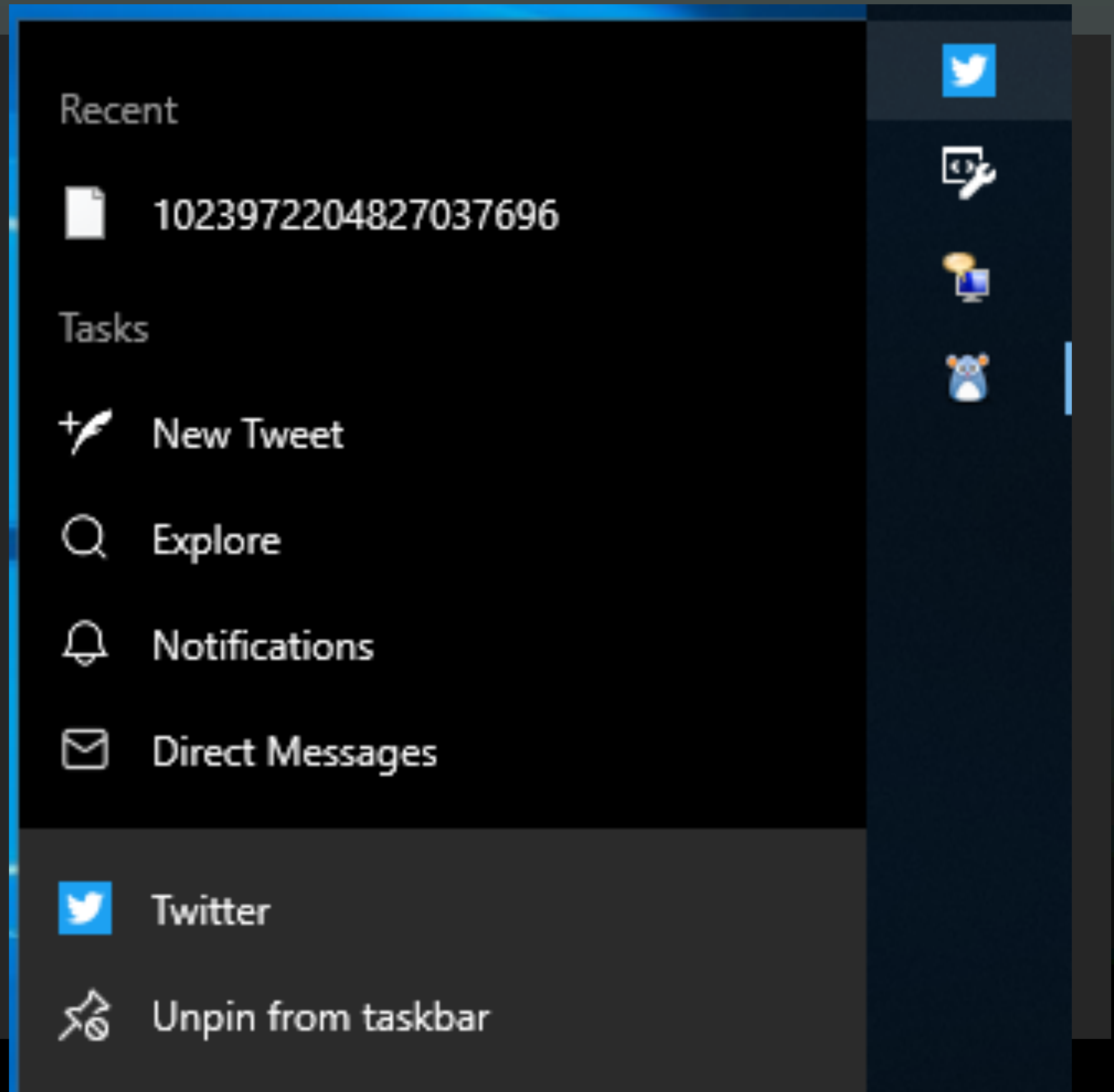
- ◉ `description`
- ◉ `screenshots`
- ◉ `categories`
- ◉ `iarc_rating_id`
- ◉ Under consideration:
 - ◉ Developer/Publisher
 - ◉ Policies
 - ◉ Banner Graphic
 - ◉ Change Log

Bonus content!

- Rich Installation / Store Pages
- Shortcuts
- Share Target

Quick access to key tasks

```
{  
  ...  
  "shortcuts": [ ... ]  
}
```



Anatomy of a Shortcut Item

```
{  
  "name": "New Tweet",  
  "url": "/compose/tweet",  
  "icons": [{  
    "src": "icon-compose.png",  
    "type": "image/png",  
    "sizes": "192x192"  
  }]  
}
```

Anatomy of a Shortcut Item

```
{  
  "name": "New Tweet",  
  "url": "/compose/tweet",  
  "icons": [{  
    "src": "icon-compose.png",  
    "type": "image/png",  
    "sizes": "192x192"  
  }]  
}
```

Anatomy of a Shortcut Item

```
{  
  "name": "New Tweet",  
  "url": "/compose/tweet",  
  "icons": [{  
    "src": "icon-compose.png",  
    "type": "image/png",  
    "sizes": "192x192"  
  }]  
}
```

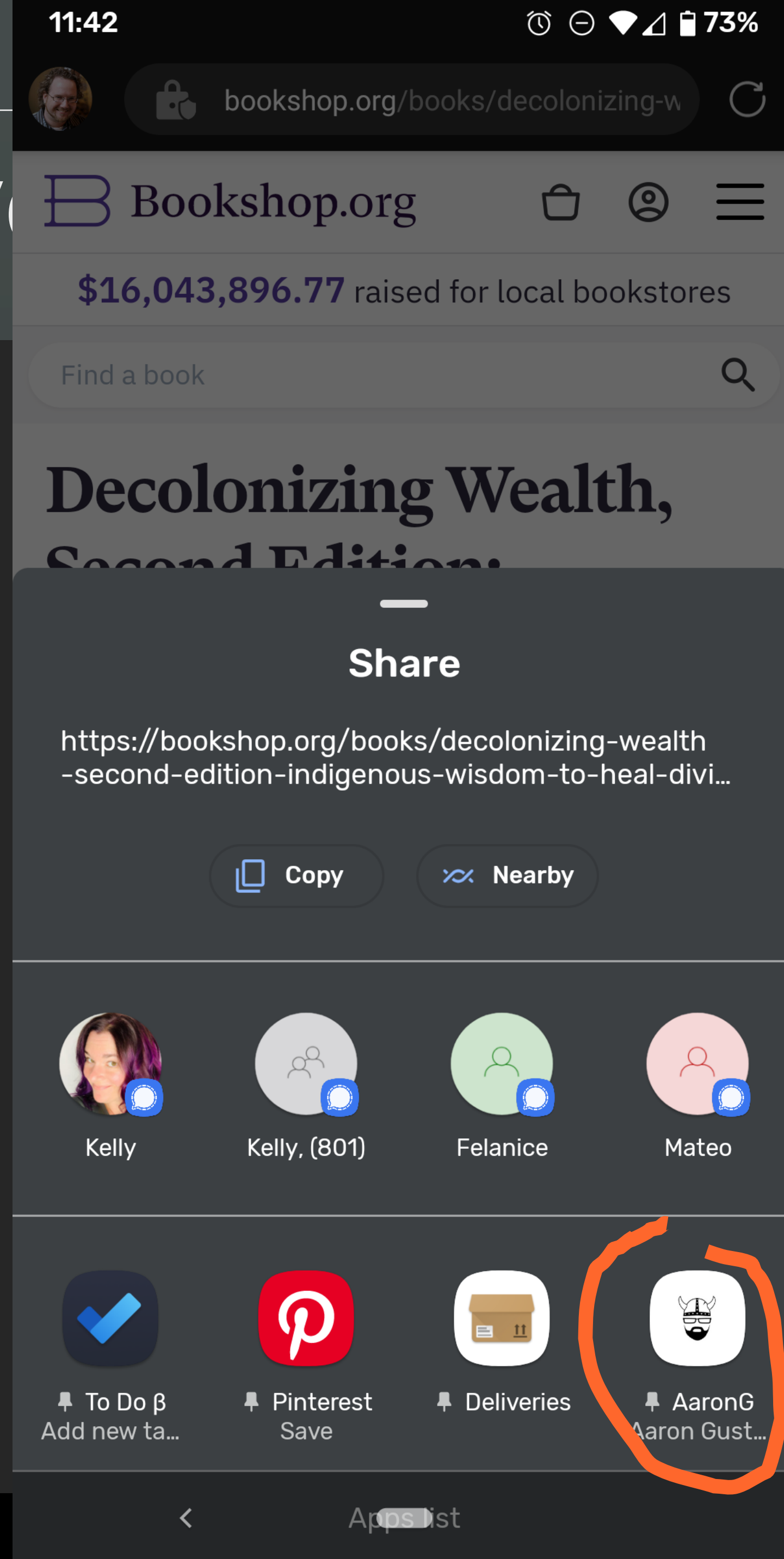

Bonus content!

- Rich Installation / Store Pages
- Shortcuts
- Share Target

@AaronGustafson

Easy sharing to y

```
{  
  ...  
  "share_target": {}  
}
```



Anatomy of a Share Target

```
"share_target": {  
  "action": "linky/poo/",  
  "method": "GET",  
  "enctype": "application/x-www-form-urlencoded",  
  "params": {  
    "title": "title",  
    "text": "body",  
    "url": "url"  
  }  
}
```

Anatomy of a Share Target

```
"share_target": {  
  "action": "linky/poo/",  
  "method": "GET",  
  "enctype": "application/x-www-form-urlencoded",  
  "params": {  
    "title": "title",  
    "text": "body",  
    "url": "url"  
  }  
}
```

Anatomy of a Share Target

```
"share_target": {  
  "action": "linky/poo/",  
  "method": "GET",  
  "enctype": "application/x-www-form-urlencoded",  
  "params": {  
    "title": "title",  
    "text": "body",  
    "url": "url"  
  }  
}
```

Anatomy of a Share Target

```
"share_target": {  
  "action": "linky/poo/",  
  "method": "GET",  
  "enctype": "application/x-www-form-urlencoded",  
  "params": {  
    "title": "title",  
    "text": "body",  
    "url": "url"  
  }  
}
```



You can also accept files

```
"params": {  
  "text": "text",  
  "url": "url",  
  "files": [{  
    "name": "externalMedia",  
    "accept": [  
      "image/jpeg",  
      "image/png",  
      "image/gif",  
      "video/quicktime",  
      "video/mp4"  
    ]  
  }]  
}]
```

PWAs **start with a great web experience** and then enhance that experience for performance, resilience, installation, and engagement

Congrats, you made a PWA!



HTTPS



Web App
Manifest



Service
Worker

Thank you!

@AaronGustafson

aaron-gustafson.com

noti.st/AaronGustafson

