

Progressive Web Apps

by Bobby Bol

A Brief Encounter with Bobby Bol



www.alion.nl

Progressive Web Apps

Upfront Definition:

“If ever the best of the web and the best of apps had a clone child – it is PWA.”

– Deepu S Nath

Progressive Web Apps

Why do I like them?

“PWA’s are inclusive.”

The overly simplistic approach to PWAs



The overly simplistic approach to PWAs

“If it’s installable,
my work is
done here.”



Simplistic PWA: Ingredients

Simplistic PWA: Ingredients

1. Secure HTTPS connection

The Minimum Viable PWA <https://mvpwa.bobbybol.com>

index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>MVPWA</title>
6    <link rel="manifest" href="/manifest.webmanifest">
7  </head>
8  <body>
9    <h1>MVPWA</h1>
10   <script>navigator.serviceWorker.register('/sw.js')</script>
11 </body>
12 </html>
```

Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest

The Minimum Viable PWA <https://mvpwa.bobbybol.com>

```
{ } manifest.webmanifest ▶ ...  
1  {  
2    "short_name": "MVPWA",  
3    "name": "MVPWA",  
4    "start_url": ".",  
5    "display": "standalone",  
6    "icons": [  
7      {  
8        "src": "/icon-192.png",  
9        "sizes": "192x192",  
10       "type": "image/png"  
11      }  
12    ]  
13 }
```

Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest
3. Service Worker

The Minimum Viable PWA <https://mvpwa.bobbybol.com>

JS sw.js ▸ ...

```
1  self.addEventListener('install', event => {  
2    event.waitUntil(  
3      caches.open('sw-cache')  
4        .then(cache => cache.add('index.html'))  
5    )  
6  })  
7  
8  self.addEventListener('fetch', event => {  
9    event.respondWith(  
10     caches.match(event.request)  
11       .then(response => response || fetch(event.request))  
12    )  
13  })
```

Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest
3. Service Worker

} Installable Web App

Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest
3. Service Worker
4. ...

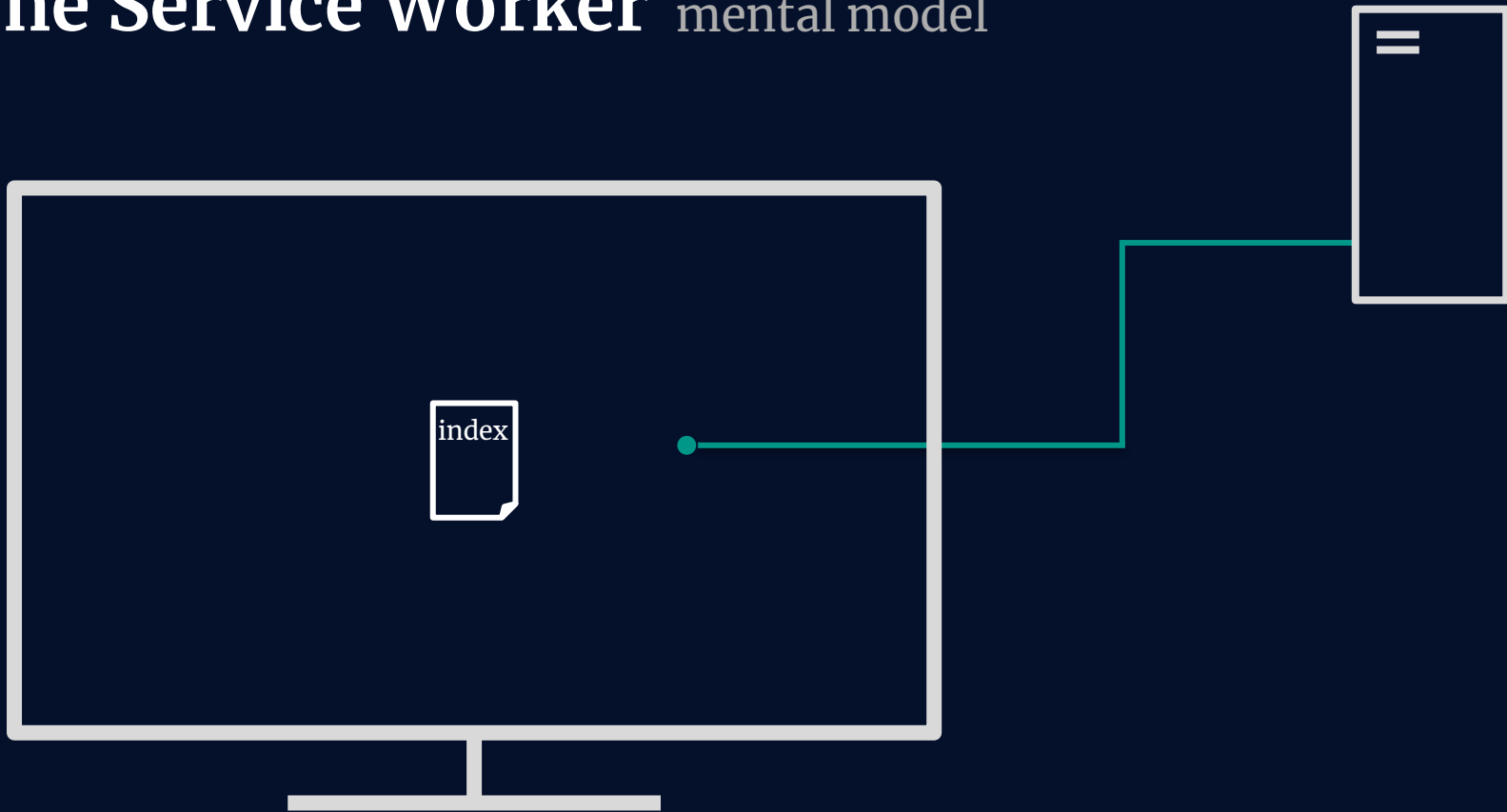
} Installable Web App

The Service Worker

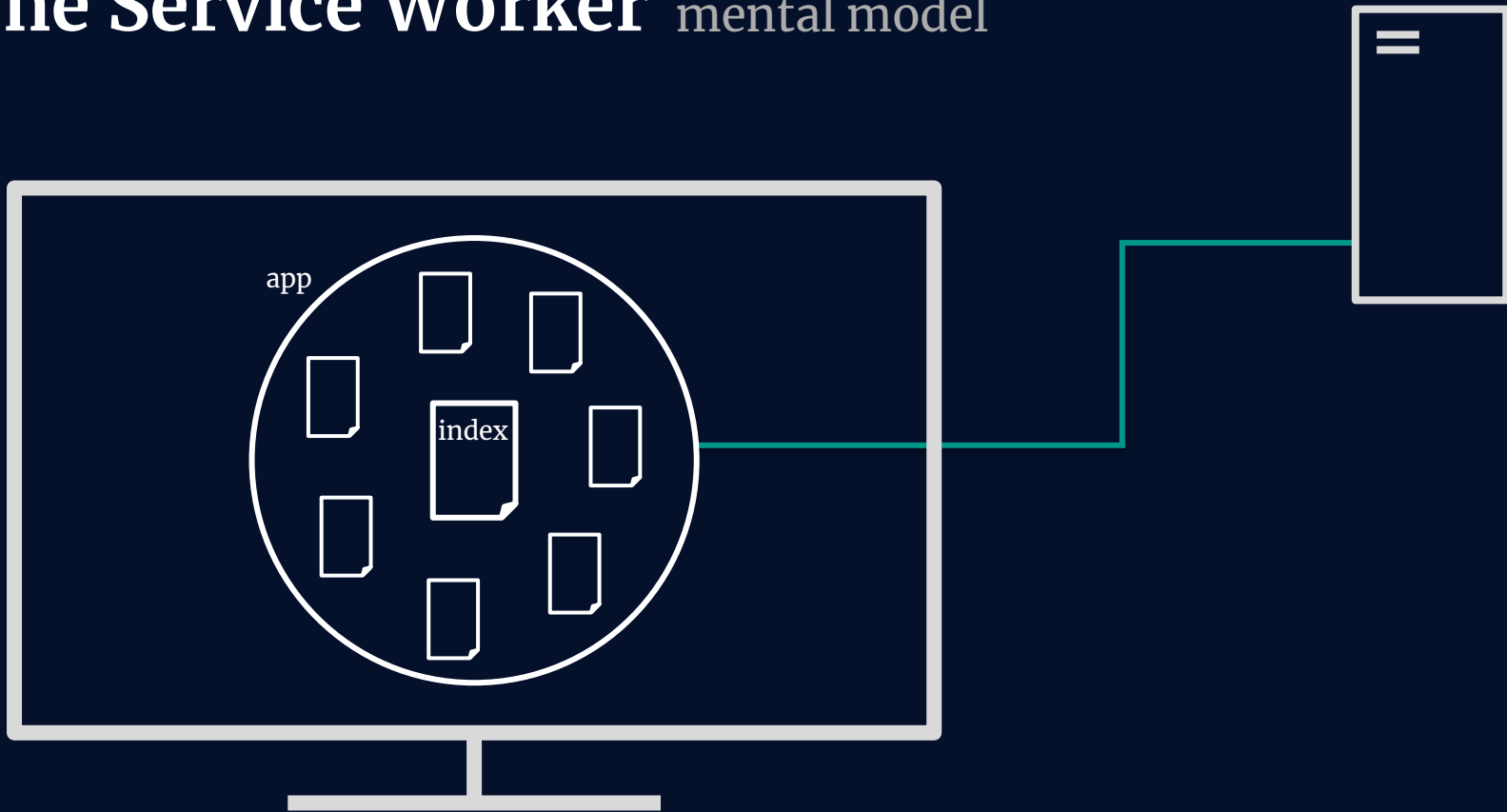
mental model



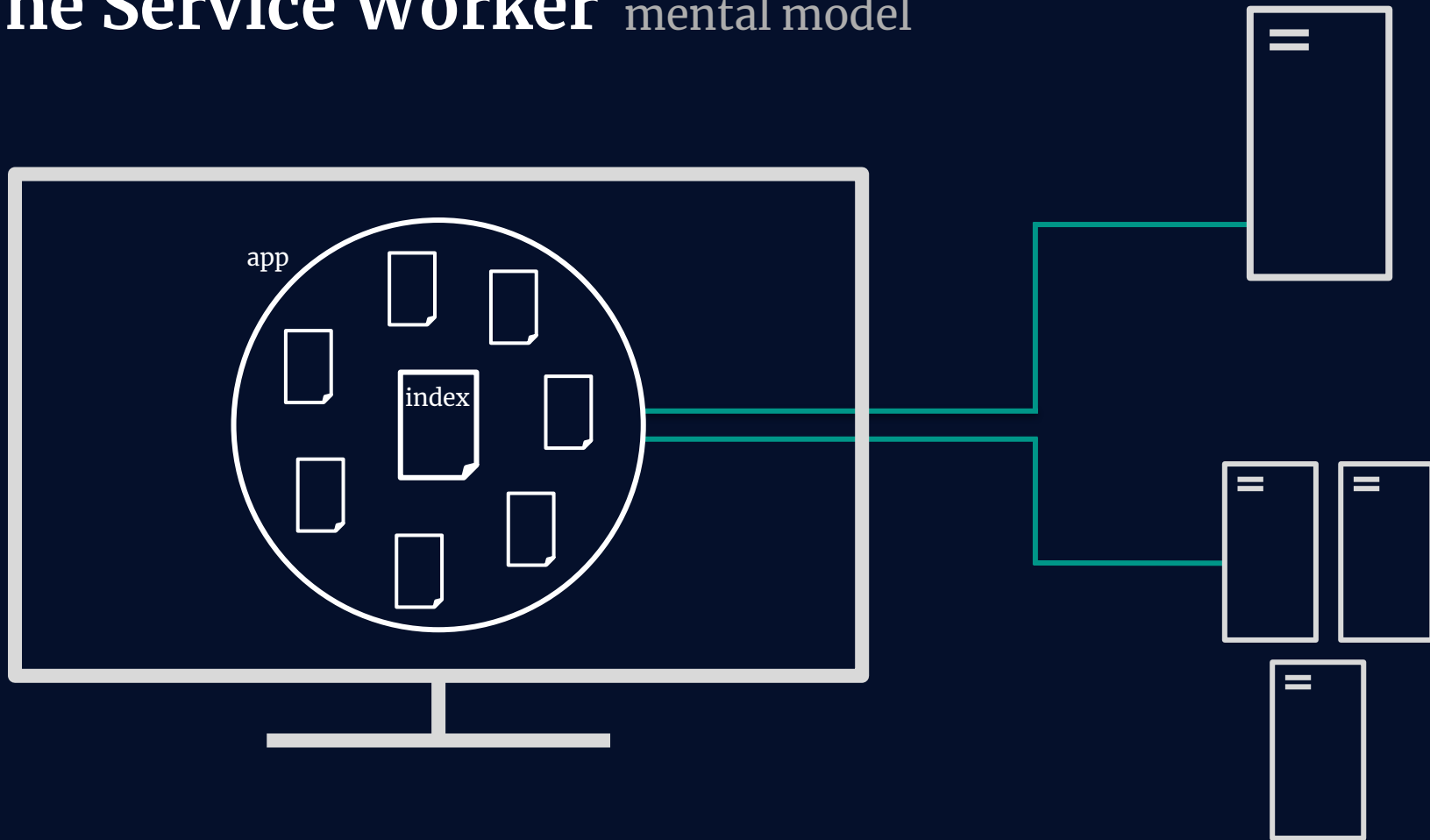
The Service Worker mental model



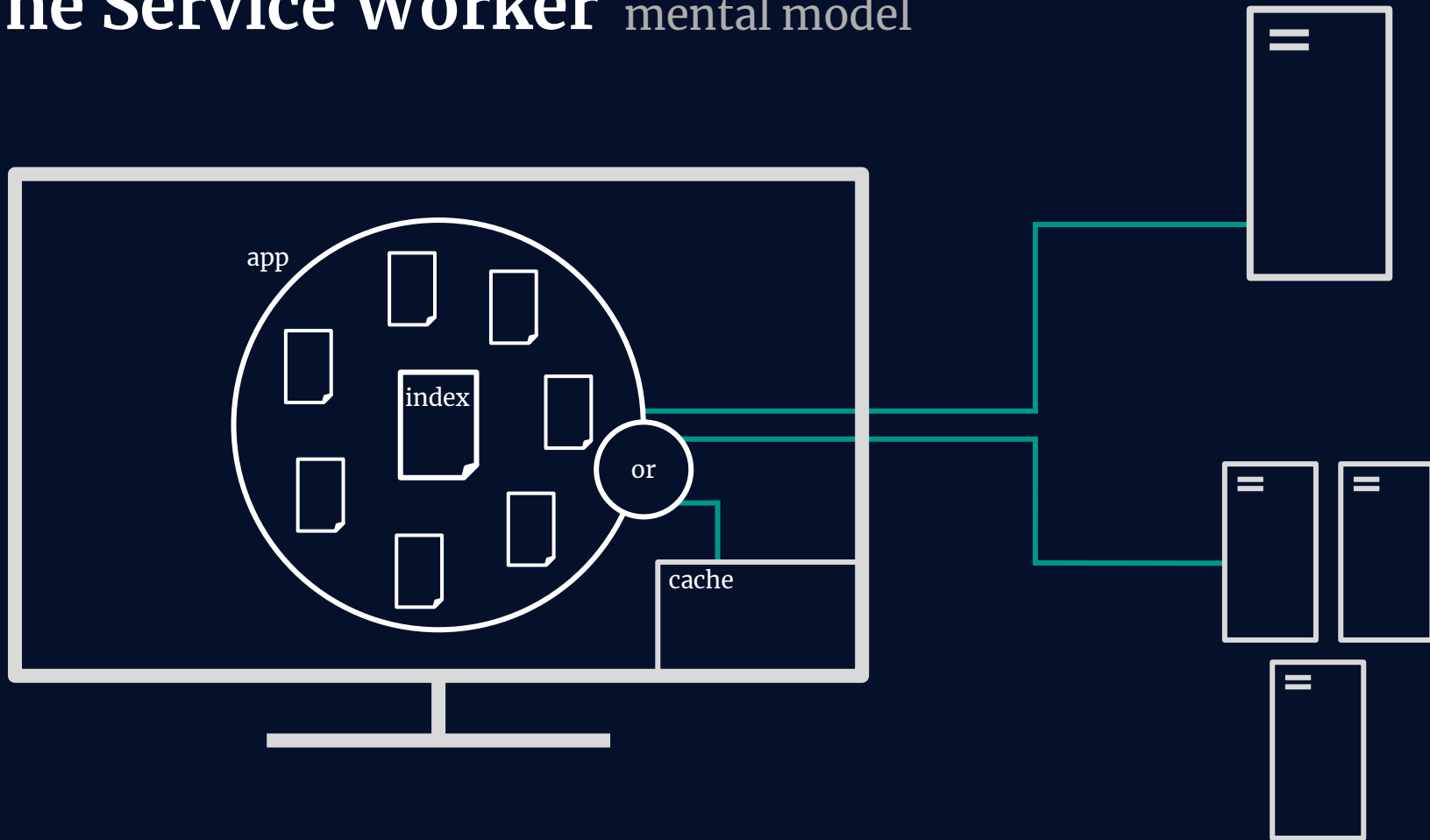
The Service Worker mental model



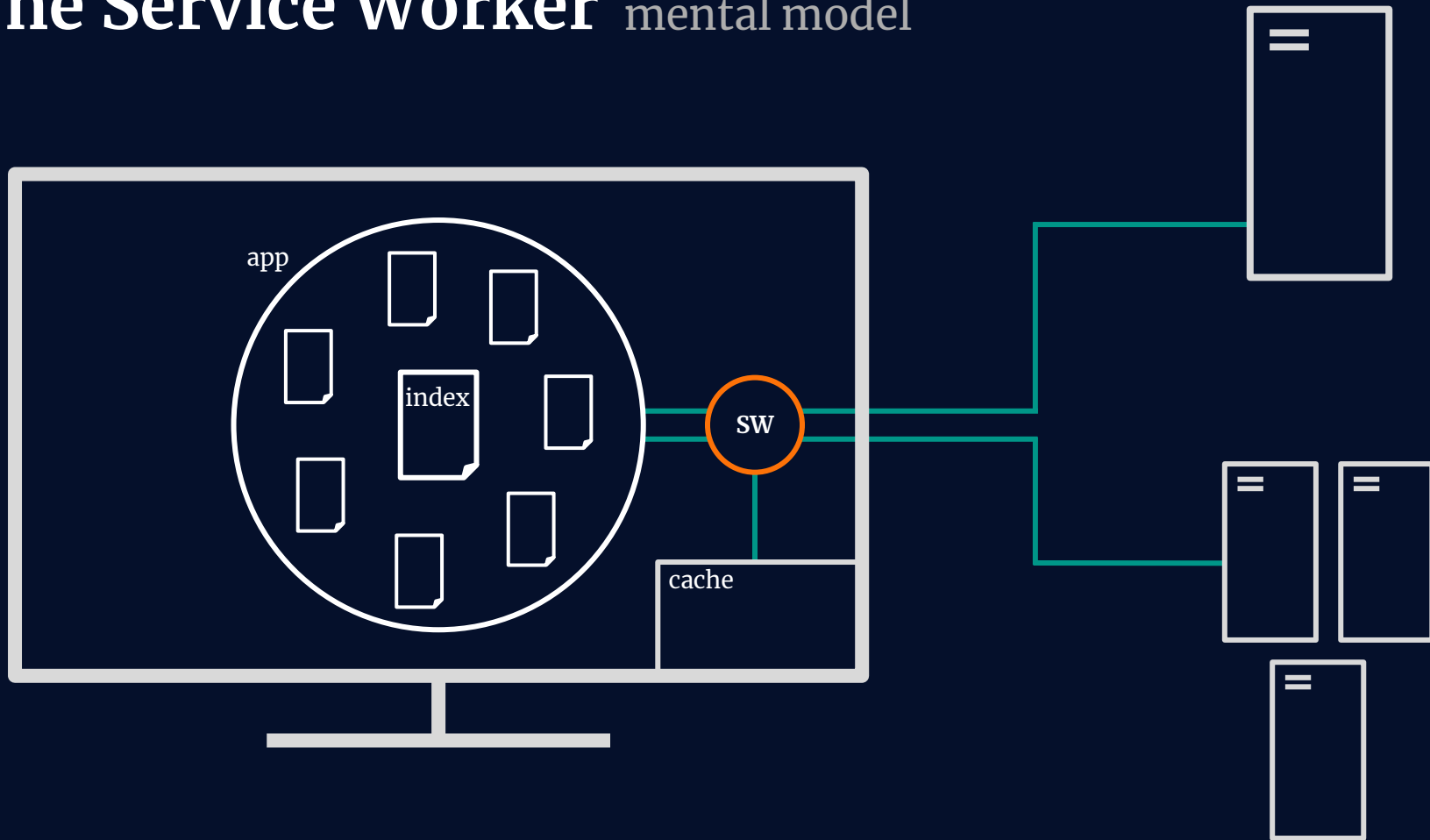
The Service Worker mental model



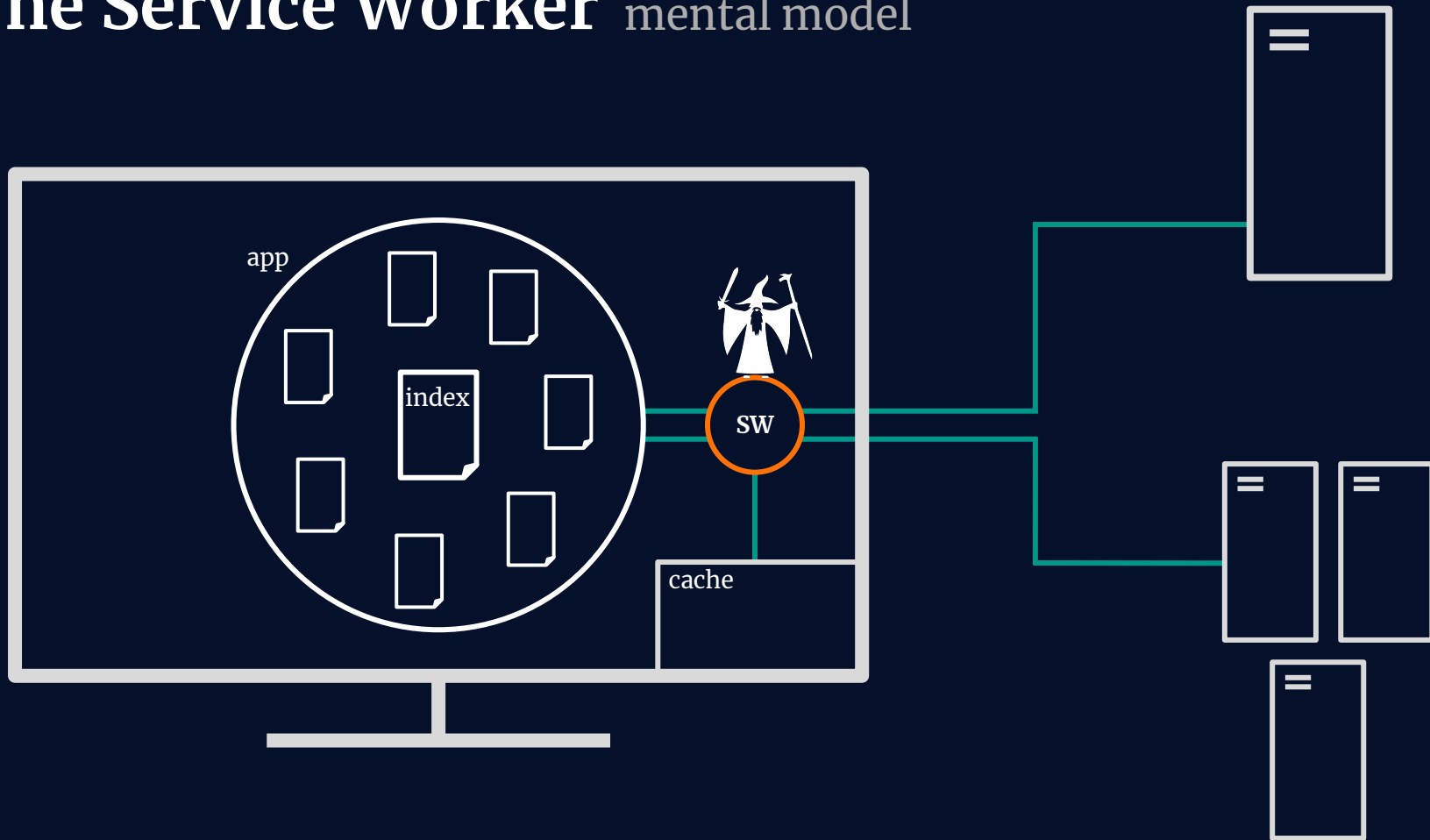
The Service Worker mental model



The Service Worker mental model



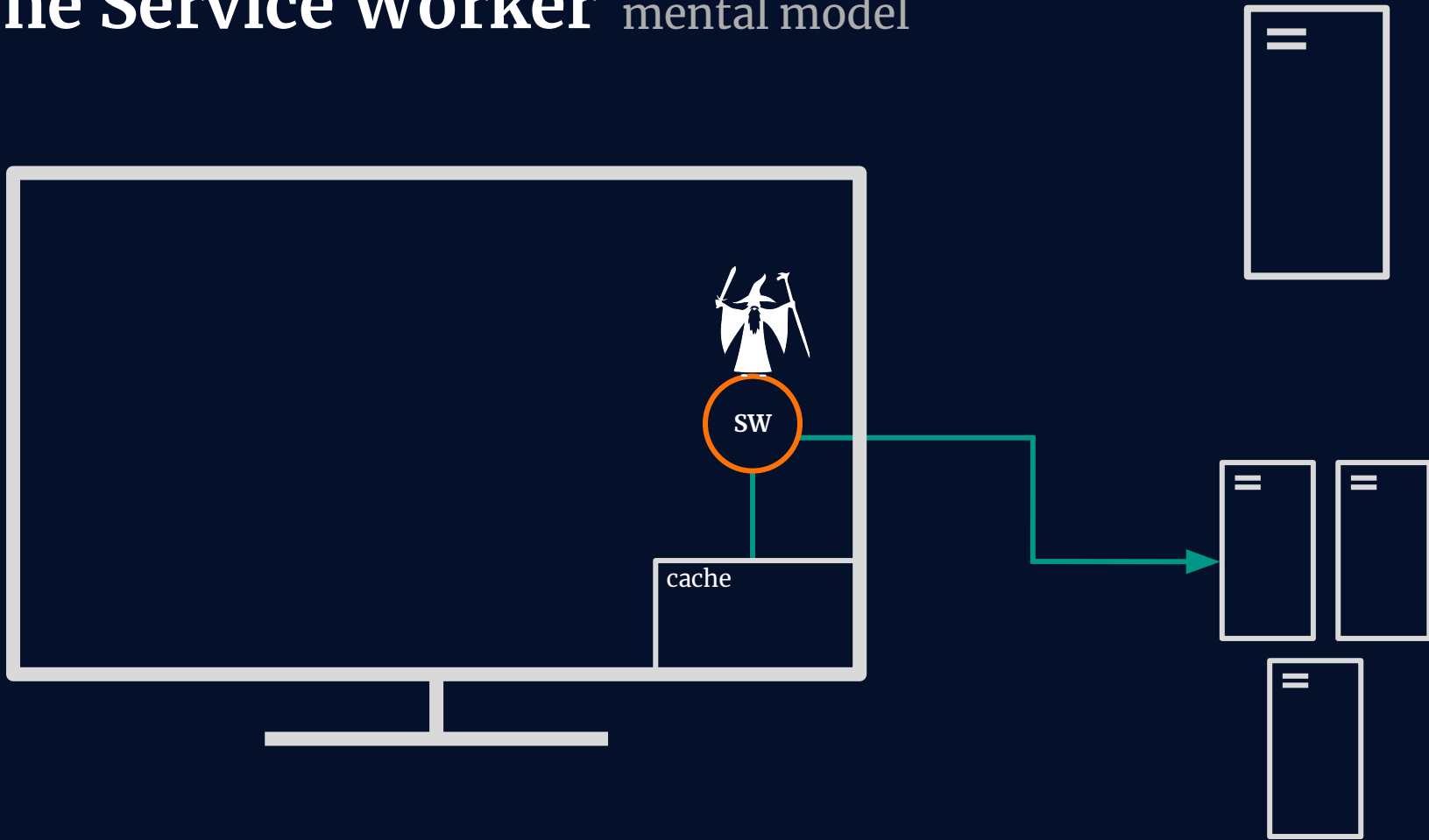
The Service Worker mental model



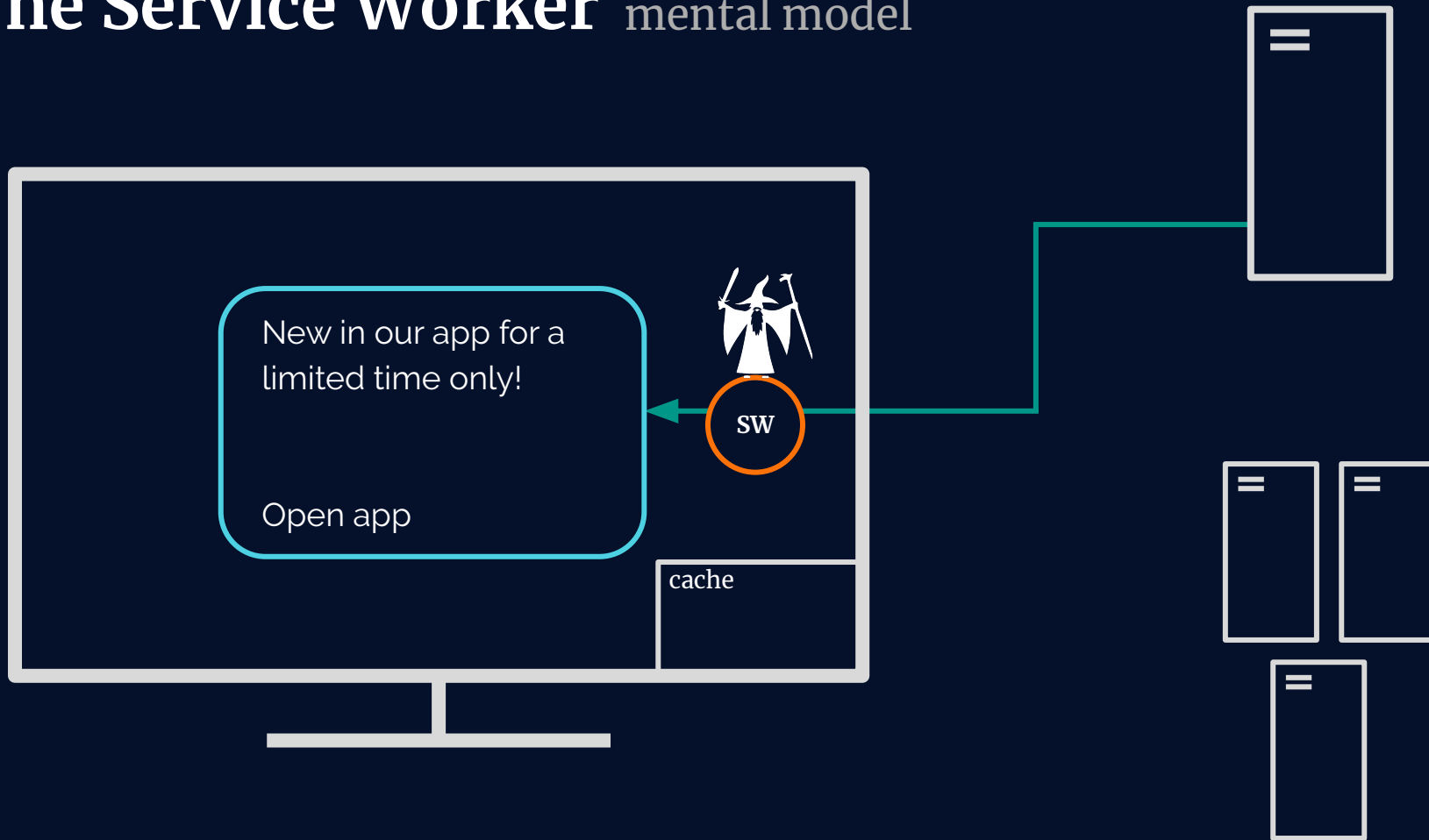
The Service Worker mental model



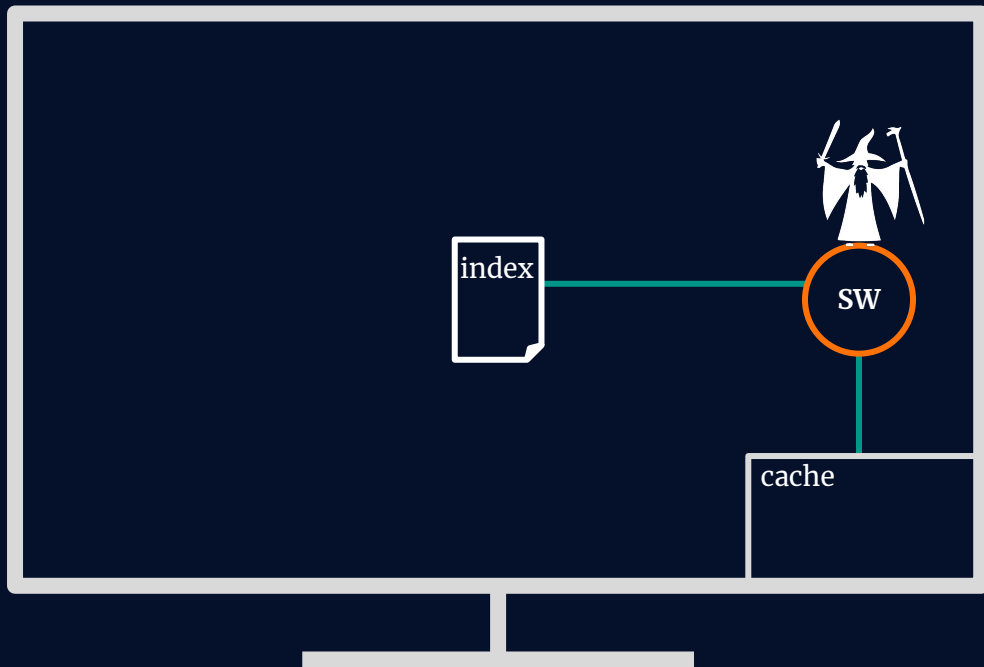
The Service Worker mental model



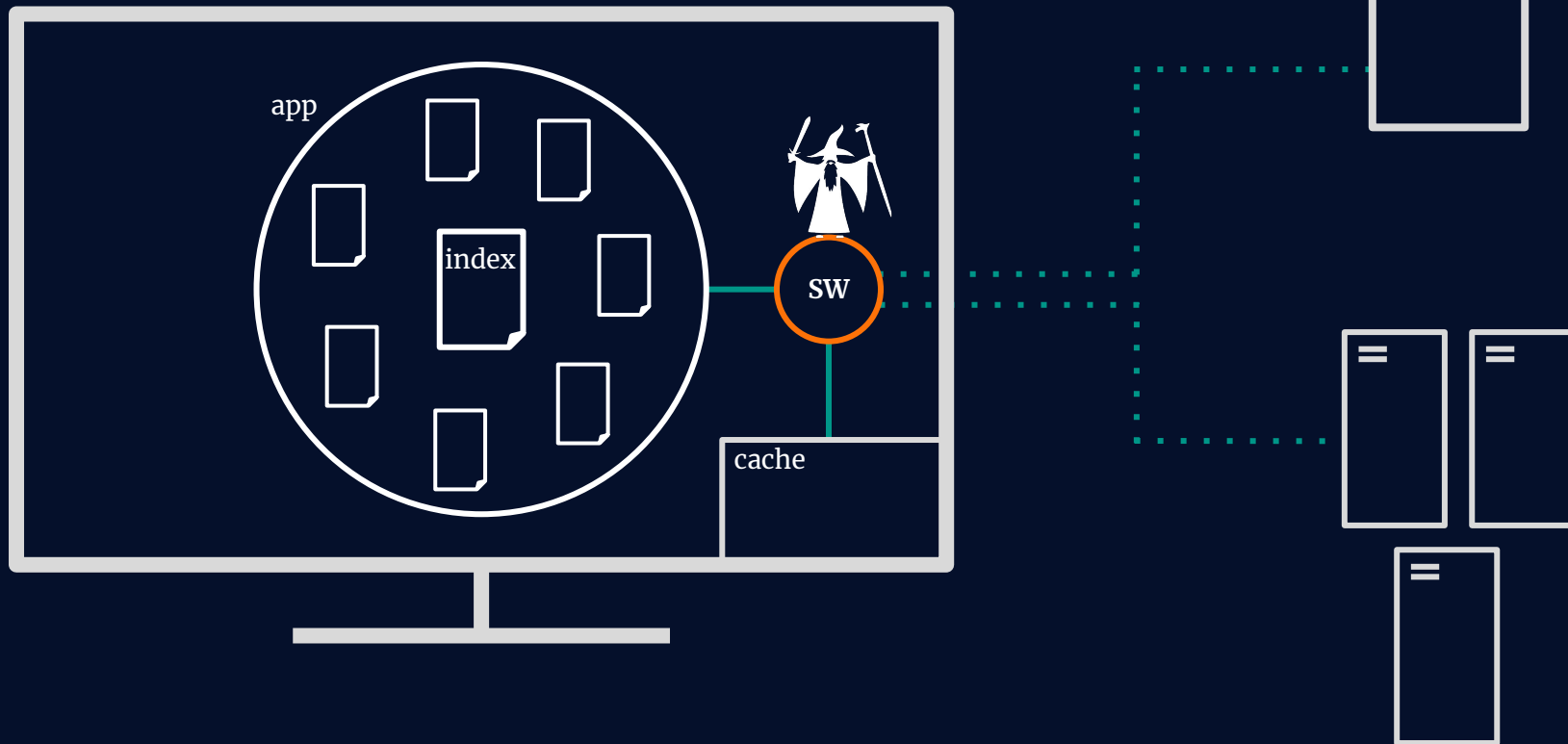
The Service Worker mental model



The Service Worker mental model



The Service Worker mental model



The Service Worker

caching strategies



// Cache-First

// Network-First

// Stale-While-Revalidate

// Network-Only

// Cache-Only

//

The Service Worker

caching strategies



- // Cache-First
- // Network-First
- // Stale-While-Revalidate
- // Network-Only
- // Cache-Only
- // Pre-caching...

Pre-caching

JS sw.js ▸ ...

```
1  self.addEventListener('install', event => {  
2    event.waitUntil(  
3      caches.open('sw-cache')  
4        .then(cache => cache.add('index.html'))  
5    )  
6  })  
7  
8  self.addEventListener('fetch', event => {  
9    event.respondWith(  
10     caches.match(event.request)  
11       .then(response => response || fetch(event.request))  
12    )  
13  })
```

Pre-caching

```
JS sw.js ▶ ...  
1  const STATIC = [  
2      '/',  
3      '/index.html',  
4      '/offline.html',  
5      'https://fonts.googleapis.com/css?family=Open+Sans:400,700',  
6      '/bundle/app.css',  
7      '/bundle/vendor.js',  
8      '/bundle/app.js',  
9      '/assets/img/logo.png'  
10 ]  
11  
12 self.addEventListener('install', event => {  
13     event.waitUntil(  
14         caches.open('sw-cache')  
15         .then(cache => cache.addAll(STATIC))  
16     )  
17 })
```

Pre-caching with



Workbox

JS workbox.-example.js ▸ ...

```
1  const gulp = require('gulp')
2  const workboxBuild = require('workbox-build')
3
4  gulp.task('build-sw', done => {
5    workboxBuild.injectManifest({
6      swSrc: 'src/sw/sw-develop.js',
7      swDest: 'sw.js',
8      globDirectory: 'src',
9      globPatterns: [
10       '**/*.html',
11       'bundle/*.{js,css}',
12       'assets/**/*.js'
13     ]
14   }).then(done)
15 })
```

JS sw.js ▸ ...

```
1  importScripts('https://storage.googleapis.com/
   workbox-cdn/releases/3.4.1/workbox-sw.js');
2
3  workbox.precaching.precacheAndRoute([]);
```


Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest
3. Service Worker
4. Pre-cache build pipeline

} Installable Web App

Simplistic PWA: Ingredients

1. Secure HTTPS connection
2. Web Manifest
3. Service Worker
4. Pre-cache build pipeline

} Installable Web App

“A decent start, but where’s the “Progressive”?”

End of Part I

Progressive PWAs

Patterns to break out of the “naive PWA” box

Welcome to Part II

Pattern 1: Wrap native prompts



Pattern 1: Wrap native prompts



```
JS deferred-prompt.js ▶ ...
1   let deferredPrompt
2
3   window.addEventListener('beforeinstallprompt', event => {
4     event.preventDefault()
5     deferredPrompt = event
6     return false
7   })
8
```

Pattern 1: Wrap native prompts

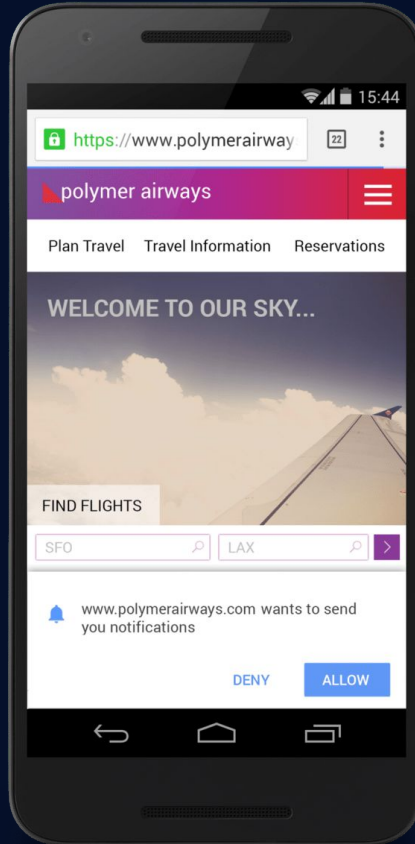
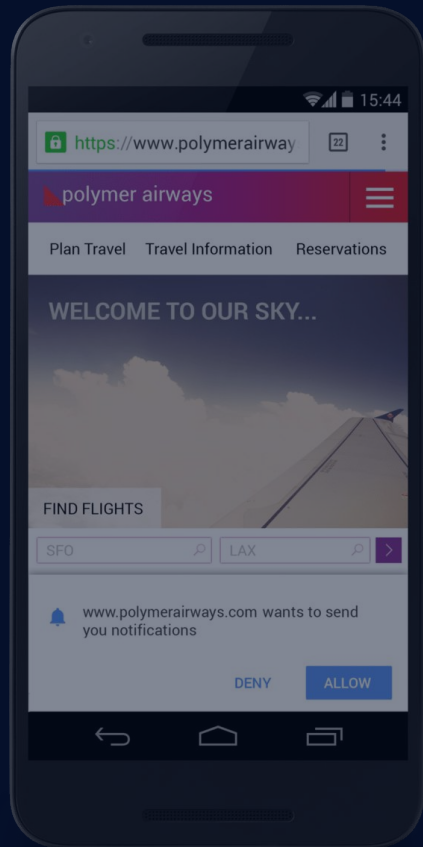



Image: <https://developers.google.com/web/fundamentals/push-notifications/permission-ux>


Pattern 1: Wrap native prompts



```
patterns ▸ JS push-notifications.js ▸ ...
1  const askForNotificationPermission = () => {
2    Notification.requestPermission(result => {
3      if(result !== 'granted') {
4        // The user doesn't want notifications
5      } else {
6        // We were granted permission to send notifications
7      }
8    });
9  }
10
11  if('Notification' in window && 'serviceWorker' in navigator) {
12    const $permissionsButton = document.querySelector('.may-we-send-notifications');
13    $permissionsButton.style.display = 'block';
14    $permissionsButton.addEventListener('click', askForNotificationPermission);
15  }
16
```

Pattern 2: Progressive Enhancement

 Alionagram



CAPTURE

Title

10 years celebration

Location

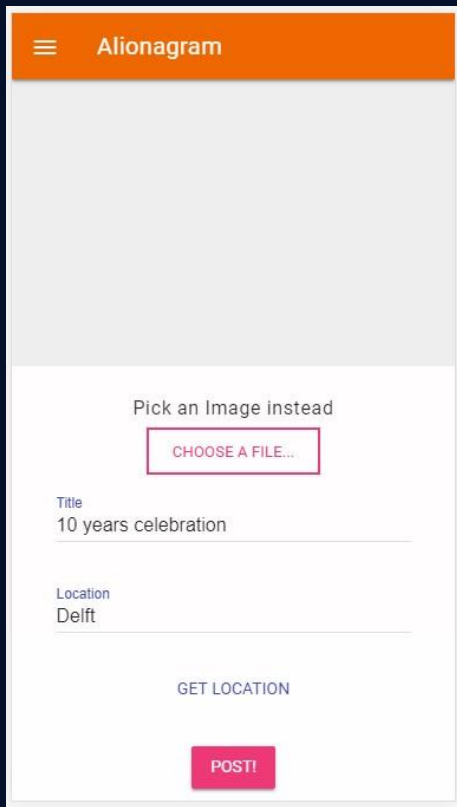
Delft

GET LOCATION

POST!

Optimal Situation

Pattern 2: Progressive Enhancement



Alionagram

Pick an Image instead

CHOOSE A FILE...

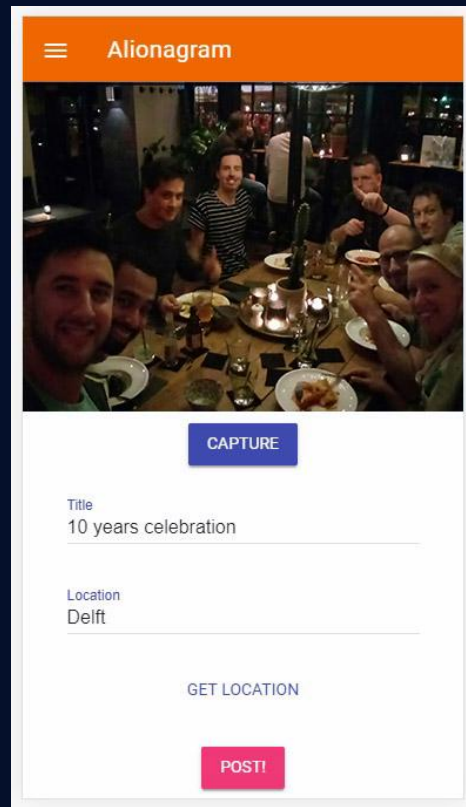
Title
10 years celebration

Location
Delft


GET LOCATION

POST!

+



Alionagram



CAPTURE

Title
10 years celebration

Location
Delft

GET LOCATION

POST!

++

Pattern 2: Progressive Enhancement

Alionagram

Pick an Image instead

CHOOSE A FILE...

Title

10 years celebration

Location

Delft

GET LOCATION

POST!

+



Alionagram

DRAG & DROP

Pick an Image instead

CHOOSE A FILE...

Title

10 years celebration

Location

Delft

GET LOCATION


POST!

×

++



Alionagram



CAPTURE

Title

10 years celebration

Location

Delft

GET LOCATION

POST!

+++

Pattern 3: Look beyond the automated solution



+

PWA Plugin

(Vue is just the example of choice)

Pattern 3: Look beyond the automated solution



+

PWA Plugin

Question: How to do local SW development??

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: How to do local SW development?

```
JS vue.config.js > [?] <unknown>
1  module.exports = {
2    pwa: {
3      workboxPluginMode: 'InjectManifest',
4      workboxOptions: {
5        swSrc: 'sw-dev.js',
6      }
7    }
8  }
```

Step 1: Configure build process

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: How to do local SW development?

JS registerServiceWorker.js x

src > JS registerServiceWorker.js > ...

```
1  /* eslint-disable no-console */
2
3  import { register } from 'register-service-worker' 2.3K (gzipped: 1K)
4
5  // if (process.env.NODE_ENV === 'production') {
6  register(`${process.env.BASE_URL}service-worker.js`, {
7    ready () {
```



Step 2: Allow SW registration in dev mode too

Pattern 3: Look beyond the automated solution

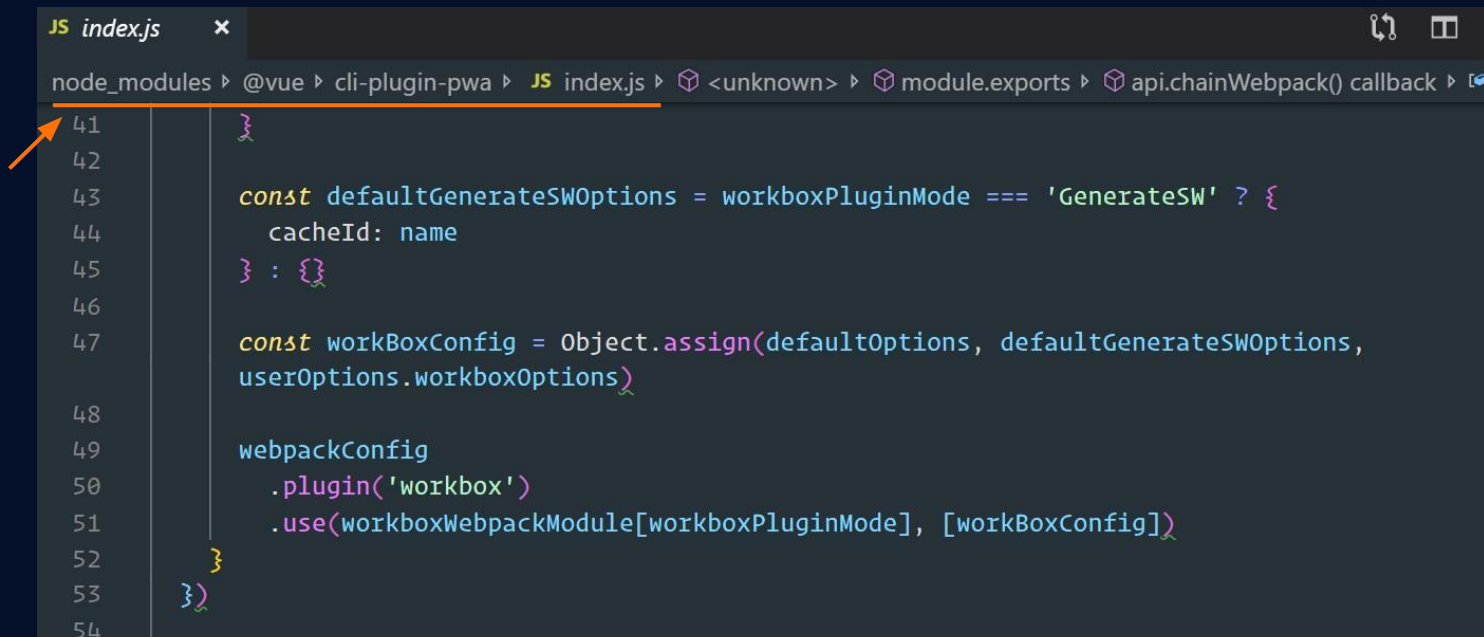
Vue CLI3 + PWA Plugin: How to do local SW development?

```
service-worker.js (index)
1  /* global self */
2
3  // This service worker file is effectively a 'no-op' that will reset any
4  // previous service worker registered for the same host:port combination.
5
6  // It is read and returned by a dev server middleware that is only loaded
7  // during development.
8
9  // In the production build, this file is replaced with an actual service worker
10 // file that will precache your site's local assets.
11
12 self.addEventListener('install', () => self.skipWaiting())
13
14 self.addEventListener('activate', () => {
15   self.clients.matchAll({ type: 'window' }).then(windowClients => {
16     for (const windowClient of windowClients) {
17       // Force open pages to refresh, so that they have a chance to load the
18       // fresh navigation response from the local dev server.
19       windowClient.navigate(windowClient.url)
20     }
21   })
22 })
23
```

Step 3: Scratch head a little 🤔

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: How to do local SW development?



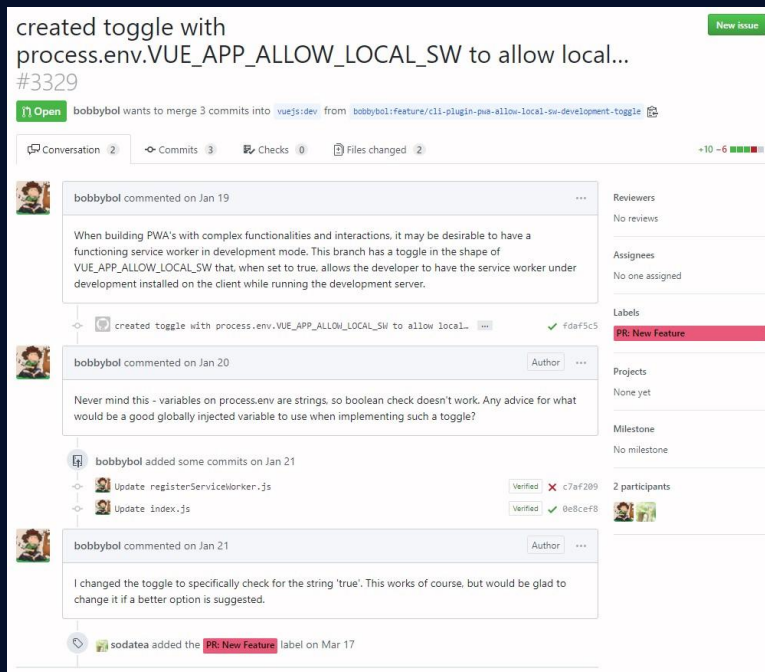
```
JS index.js x
node_modules > @vue > cli-plugin-pwa > JS index.js > <unknown> > module.exports > api.chainWebpack() callback >
41  }
42
43  const defaultGenerateSWOptions = workboxPluginMode === 'GenerateSW' ? {
44    cacheId: name
45  } : {}
46
47  const workBoxConfig = Object.assign(defaultOptions, defaultGenerateSWOptions,
48    userOptions.workboxOptions)
49
50  webpackConfig
51    .plugin('workbox')
52    .use(workboxWebpackModule[workboxPluginMode], [workBoxConfig])
53  }
54 }
```

Step 4: Start tinkering with the NPM package 🙄

```
57  if(process.env.VUE_APP_ALLOW_LOCAL_SW !== 'true') {
58    const createNoopServiceWorkerMiddleware = require('../lib/noopServiceWorkerMiddleware')
```


Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: How to do local SW development?

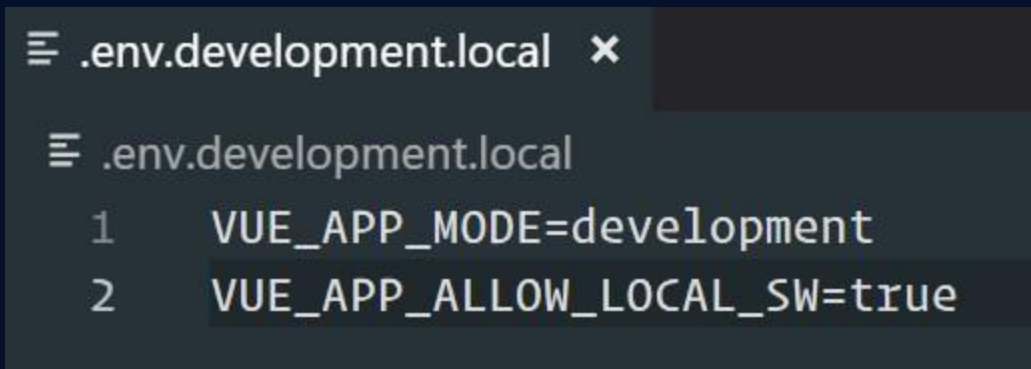


Step 5: Do it properly and make a PR 🤖

<https://github.com/vuejs/vue-cli/pull/3329>

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: How to do local SW development?

A screenshot of a code editor window. The title bar at the top reads ".env.development.local" with a close button (X) on the right. The editor content shows the same filename ".env.development.local" followed by two lines of code: "1 VUE_APP_MODE=development" and "2 VUE_APP_ALLOW_LOCAL_SW=true". The second line is highlighted with a light blue background.

```
.env.development.local X  
  
.env.development.local  
1 VUE_APP_MODE=development  
2 VUE_APP_ALLOW_LOCAL_SW=true
```

Step 6: Set your ENV variables and you can *finally* begin..

Pattern 3: Look beyond the automated solution



+

PWA Plugin

Question: What is actually being pre-cached??

Vue CLI3 + PWA Plugin: What is being pre-cached?

```

"revision": "b5b3e9a8e420b0cacf852029663d7c06",
"url": "/img/common/safari-pinned-tab.svg"

"revision": "b6216d61c03e6ce0c9aea6ca7808f7ca",
"url": "/robots.txt"

"revision": "2e834f41acb32c19e5fcd67ff68cedf3",
"url": "/web.config"

"revision": "0364284f5218bf02355ab0e435091273",
"url": "/img/common/mstile-150x150.png"

"revision": "c86bb4caaabd8157b0740f9ae4b60624",
"url": "/img/common/android-chrome-512x512.png"

"revision": "97323a4ec723ecd8a29cd7aedadd64a",
"url": "/img/common/favicon-32x32.png"

"revision": "2174cacb3f9a9fe2130989a4650d55af",
"url": "/img/common/favicon-16x16.png"

```

1b7"

```

"revision": "9649f1dfa16b0ac4906e580ae7a8cd86",
"url": "/content/images/m2/l1/video1.4.jpg"

"revision": "380a79db48d12ca360c6e739304204f3",
"url": "/content/images/m2/l1/video1.5.jpg"

"revision": "6ac8da08f8f46865286be70da1535b44",
"url": "/content/images/m1/l1/Dan@2x.jpg"

"revision": "f4a092a5871feaa38300d798c74706b7",
"url": "/content/images/m1/l6/Intro@2x.jpg"

"revision": "b20604d81aa8b848c8a3b5cd77a09bdc",
"url": "/content/images/m1/l6/Populatie2@2x.jpg"

"revision": "629c01ee537f3bf24759332065cbe0cc",
"url": "/content/images/m1/l3/Stap-1@2x.jpg"

"revision": "55c9270c031940aec3f41fce7f656adf",
"url": "/content/images/m1/l3/Start@2x.jpg"

```

All static content

```

"revision": "629097a7c9879ee13e722ad004bb92b",
"url": "/media/teaser-01.629097a7.mp4"

"revision": "bbdeb3f3f06d18244af309477dc6e588d",
"url": "/img/walkthrough-slide-03.bbdeb3f3.png"

"revision": "0151e554e358973e5b915fd905f0abe5",
"url": "/img/teaser-04.0151e554.jpg"

"revision": "fc521af2ae5ba430e65369e0aea1041e",
"url": "/img/Powered-by-blue.fc521af2.svg"

"revision": "10b2cb8b0648ea1c8094ea108a7bdc5d",
"url": "/media/teaser-07.10b2cb8b.mp4"

"revision": "9c04ddf0e0c30492d177f634b641a9d5",
"url": "/media/teaser-02.9c04ddf0.mp4"

"revision": "bdae6324a1cb0031d7418150837e4516",
"url": "/img/Logo-with-title.bdae6324.svg"

```

b"

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: What is being pre-cached?

```
src > JS router.js > ...
10 routes: [
11   {
12     path: '/',
13     name: 'home',
14     component: Home
15   },
16   {
17     path: '/about',
18     name: 'about',
19     // route level code-splitting
20     // this generates a separate chunk (about.[hash].js) for this route
21     // which is lazy-loaded when the route is visited.
22     component: () => import(/* webpackChunkName: "about" */ './views/About.vue')
23   }
24 ]
25 }
26 }
```

Pattern 3: Look beyond the automated solution

Vue CLI3 + PWA Plugin: What is being pre-cached?

```
src ▶ JS router.js ▶ ...
10 routes: [
11   {
12     path: '/',
13     name: 'home',
14     component: Home
15   },
16   {
17     path: '/about',
18     name: 'about',
19     // route level code-splitting
20     // this generates a separate chunk (about.[hash].js) for this route
21     // which is lazy-loaded when the route is visited.
22     component: () => import(/* webpackChunkName: "about" */ './views/About.vue')
23   }
24 ]
25 }
26 }
```

```
dist ▶ JS precache-manifest.4b0c7a2fcfd280b7104cc58edd8ae21.js
10 {
11   "revision": "27dff3d6cef25a1fe57f",
12   "url": "/js/app.f6f36e3d.js"
13 },
14 {
15   "revision": "be217251e140acc28c50",
16   "url": "/js/about.f84756e1.js"
17 },
18 }
```

Pattern 3: Look beyond the automated solution

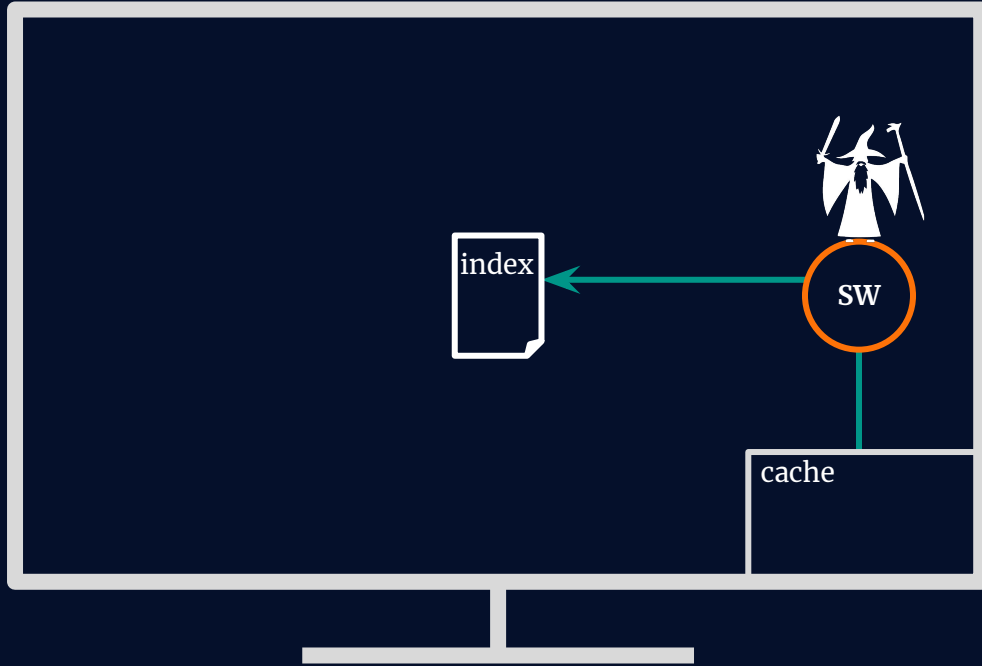
Vue CLI3 + PWA Plugin: How can we improve on this?

Go back to vue.config.js

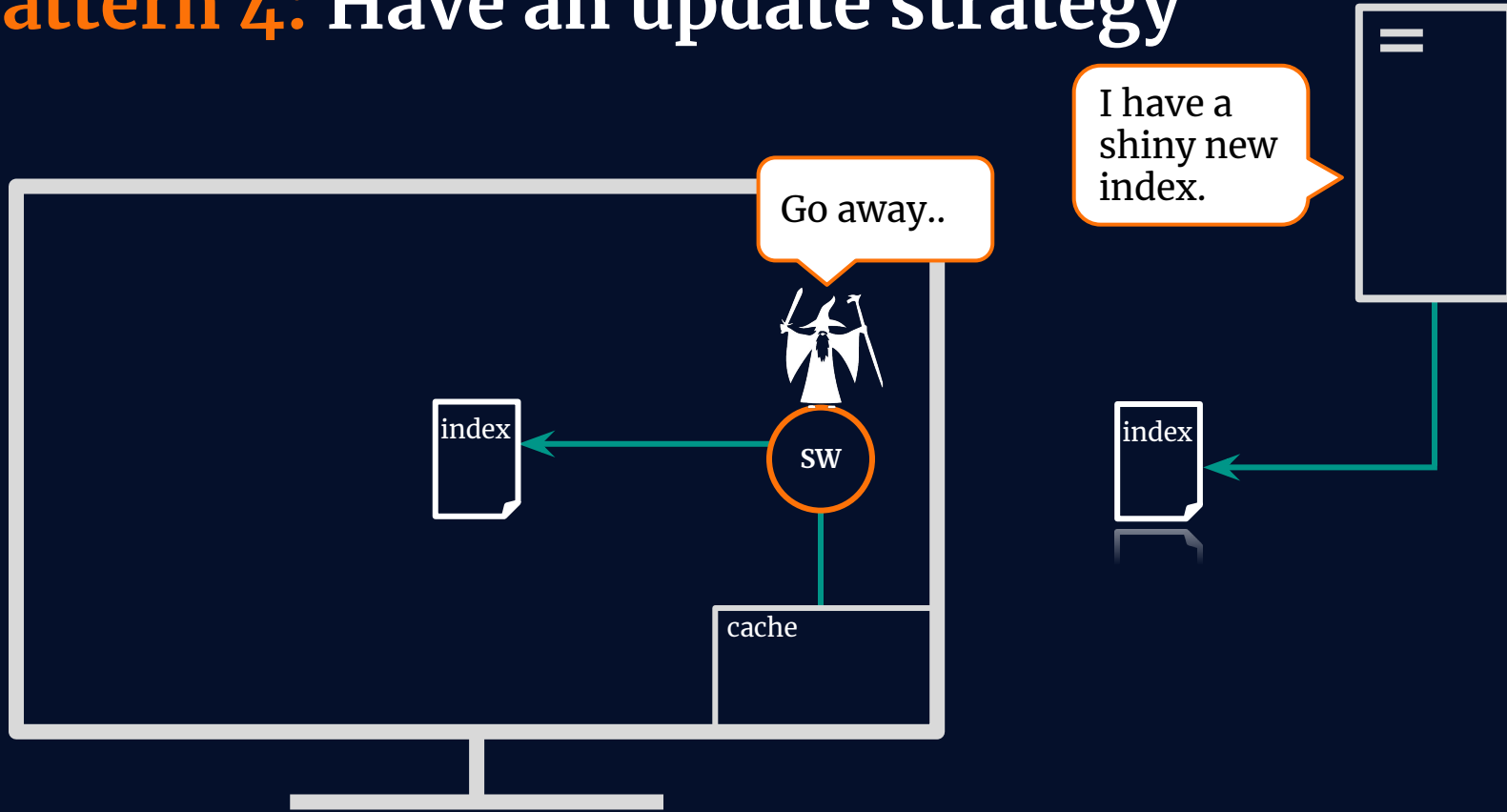
```
JS vue.config.js > [?] <unknown>
1  module.exports = {}
2    pwa: {
3      workboxPluginMode: 'InjectManifest',
4      workboxOptions: {
5        swSrc: 'sw-dev.js',
6      }
7    }
8  }
```

- a. Ditch PWA plugin and do your own Workbox configuration
- b. Tap into PWA plugin configuration and add smart glob patterns

Pattern 4: Have an update strategy



Pattern 4: Have an update strategy



Pattern 4: Have an update strategy

“Service Workers are rocket science ”

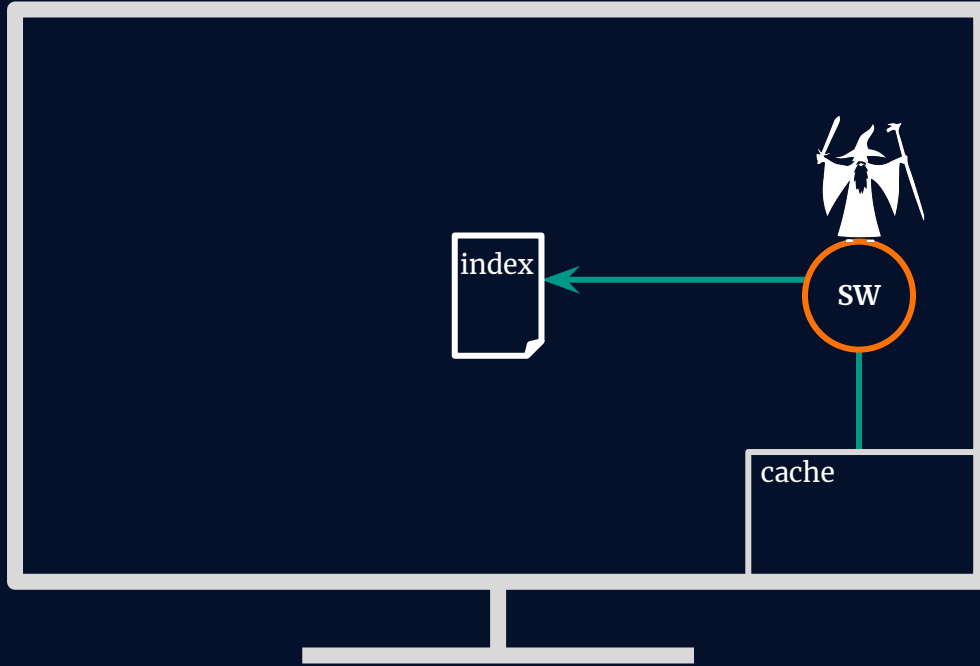
– Dan Fabulich

Pattern 4: Have an update strategy

“If a Service Worker fails, it’s possible to break an entire website in ways that we can’t fix on the server side (or at least, not right away). Imagine this: your site appears to be down, but refreshing the page won’t help, because the browser isn’t even talking to your server; it’s just talking to your broken Service Worker.”

– Dan Fabulich

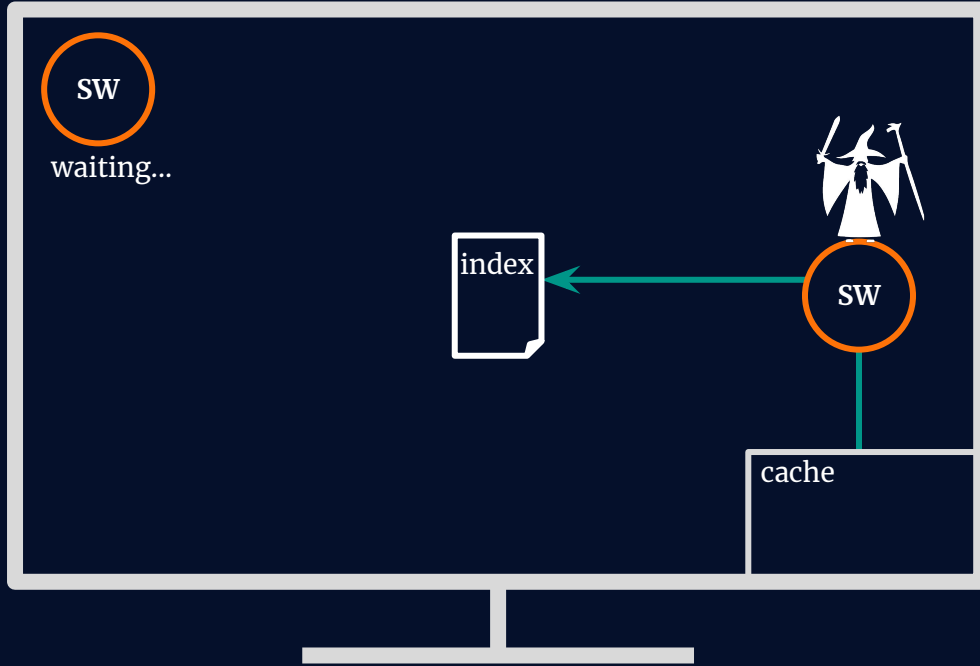
Pattern 4: Have an update strategy



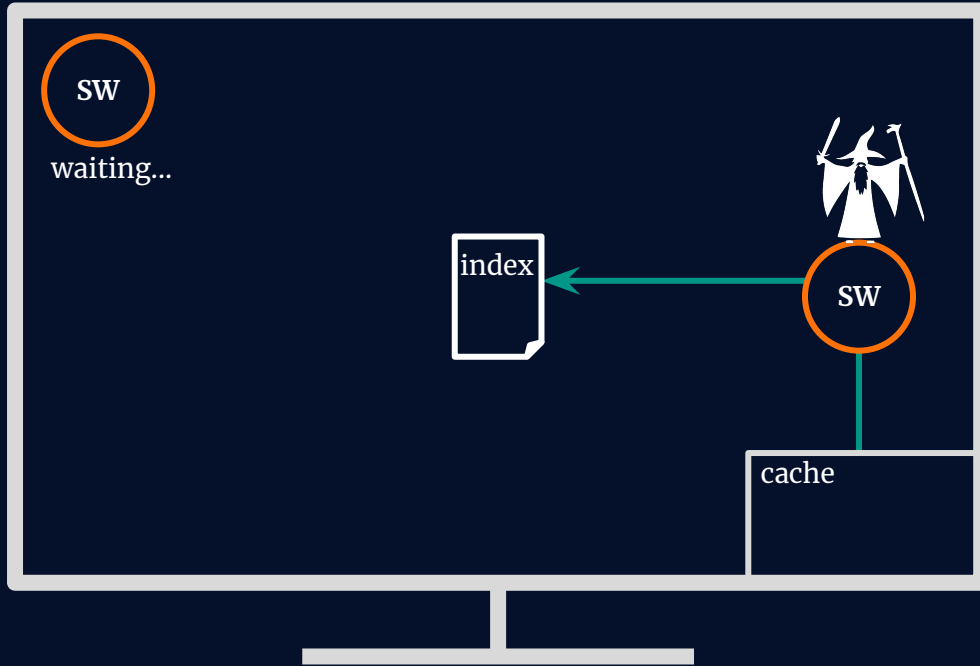
I have a shiny new SW.



Pattern 4: Have an update strategy



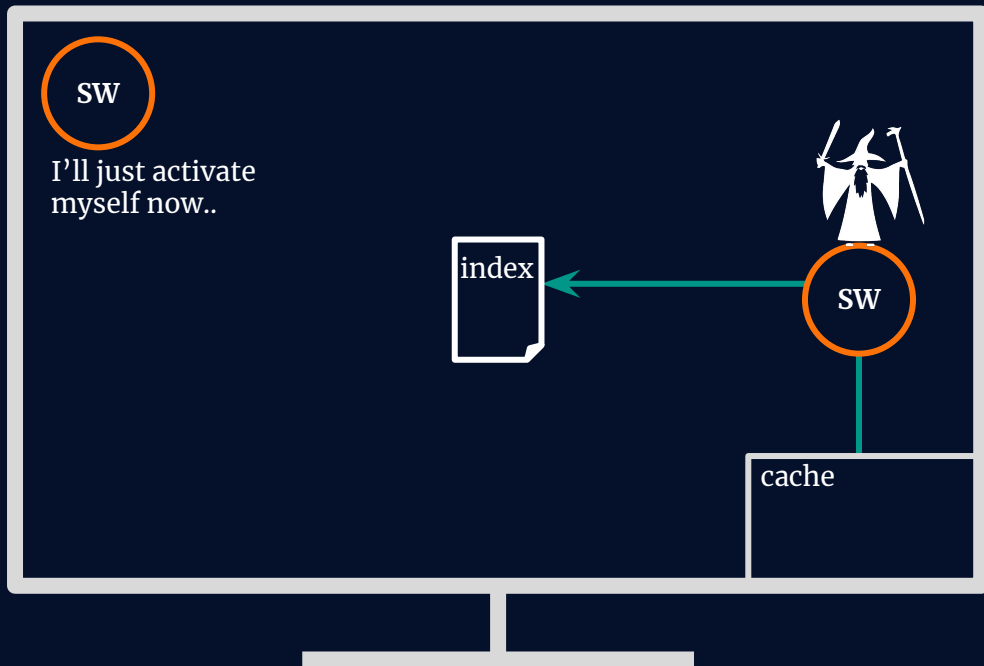
Pattern 4: Have an update strategy



A new Service Worker will wait forever, unless all “clients” are closed..

Pattern 4: Have an update strategy

Approach 1: Skip waiting



```
self.addEventListener('install', event => {  
  event.waitUntil(self.skipWaiting());  
});
```

“For every complex problem there is an answer that is clear, simple, and wrong.”

– H. L. Mencken

Pattern 4: Have an update strategy

Approach 2: Skip waiting, Claim, Refresh

SW

```
self.addEventListener('activate', event => {  
  event.waitUntil(self.clients.claim());  
});
```

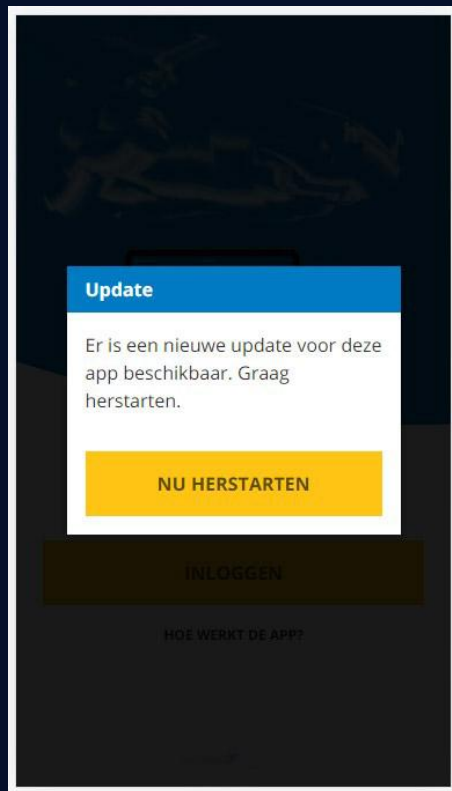
App

```
let refreshing = false;  
navigator.serviceWorker.addEventListener('controllerchange', () => {  
  if(refreshing) return;  
  refreshing = true;  
  window.location.reload();  
});
```


Pattern 4: Have an update strategy

Approach 3: Give the user some (sense of) control

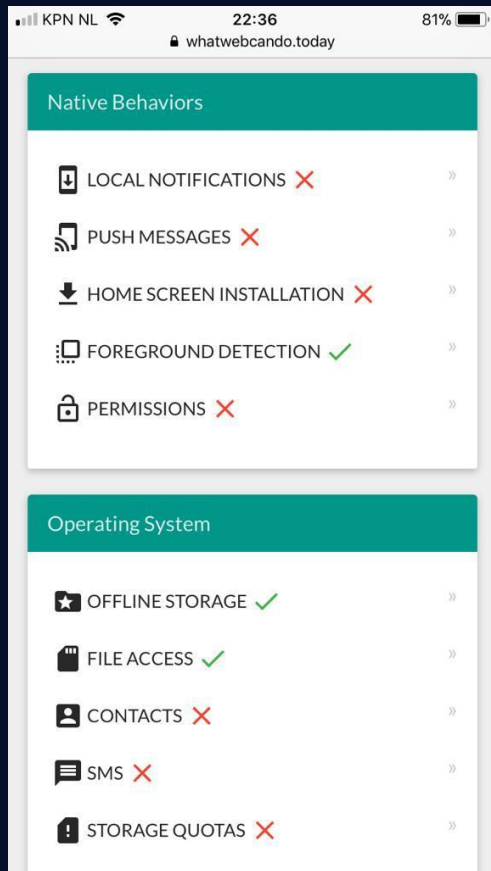
```
patterns ▸ JS update-strategy.js ▸ ...
1  import store from '@store';
2
3  if('serviceWorker' in navigator) {
4    navigator.serviceWorker.register('/sw.js').then(reg => {
5      reg.addEventListener('updatefound', () => {
6        let newWorker = reg.installing;
7        newWorker.addEventListener('statechange', () => {
8          if(newWorker.state === 'installed') {
9            if(navigator.serviceWorker.controller) {
10              store.dispatch('hasUpdate', true);
11            }
12          }
13        });
14      });
15    });
16  }
17
```



Pattern 5: Choose deliberately



caniuse.com
for single features



whatwebcando.today
mobilehtml5.com
for per-browser overview

Pattern 6: Have lots of Meetups about dev tech ;)

Questions?

Pattern 7: Bring PWAs to the Stores

So we can take over the native market
from the inside out ;)

Fin