

Ali Alabbas

Working on service workers, storage and networking

TWITTER

@MSEdgeDev

GITHUB

aliams



#msedgesummit



Service Workers

Going beyond the page

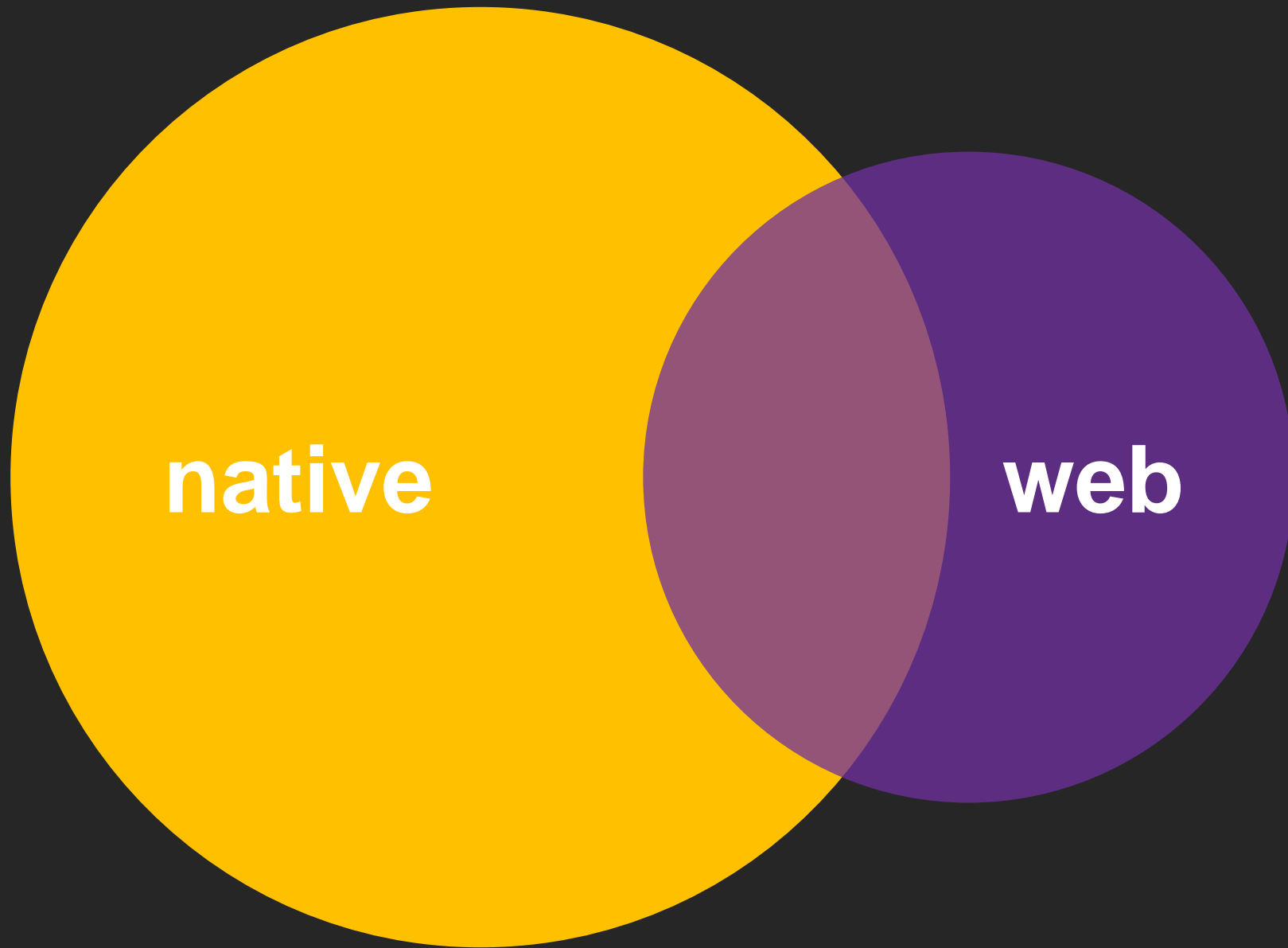


#msedgesummit

A quick history



#msedgesummit



#msedgesummit

W3C®



#msedgesummit

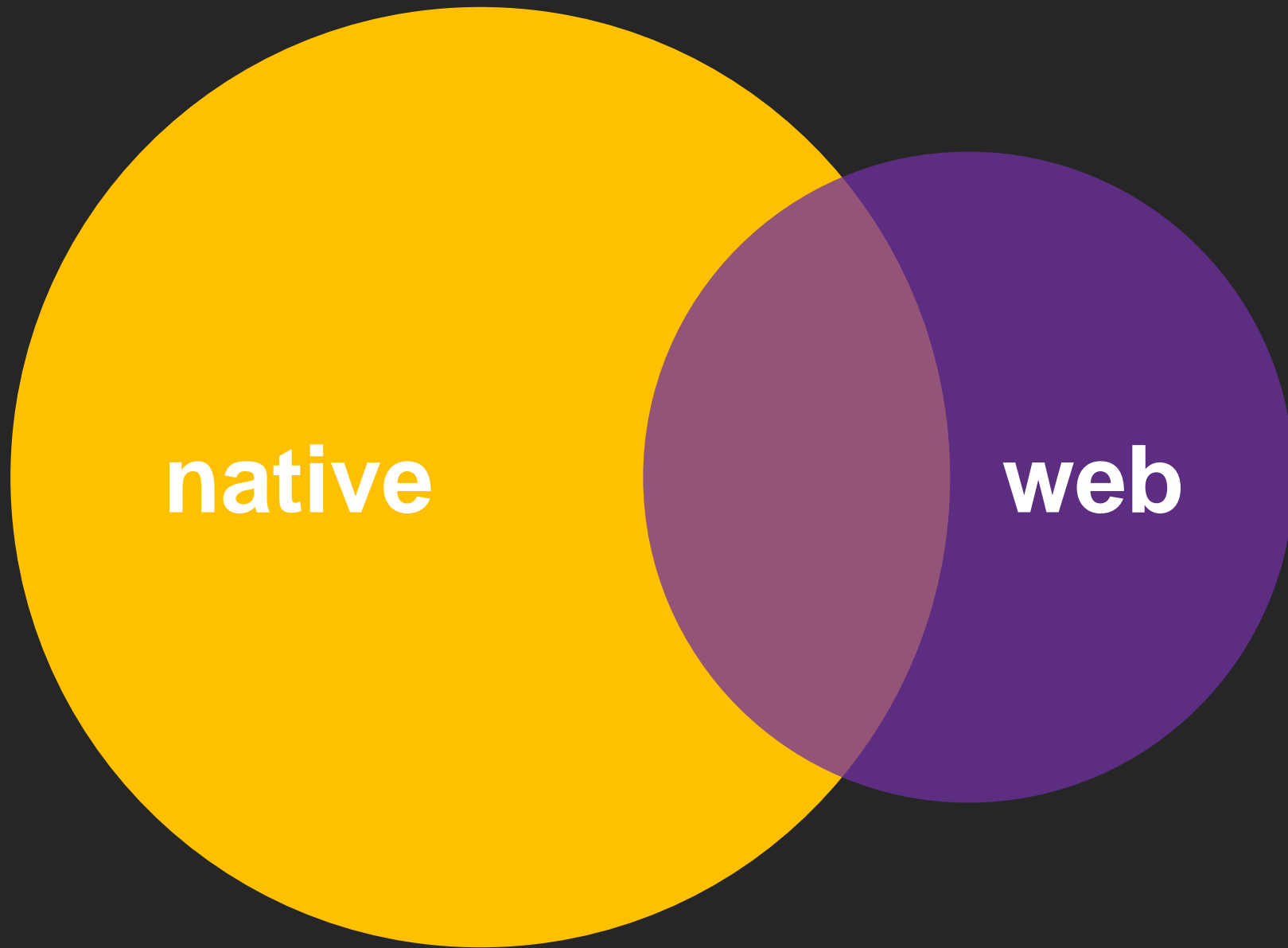
Web Graphics
Web Crypto
Indexed Database
Content Editable
Screen Orientation
Full screen API
Peer-to-peer
Web Workers
Web Payments
Web Manifest

Web Video
Web Audio
Touch Events
Gamepad API
History API
Web Sockets
Vibration API
Web Storage
Canvas
File API

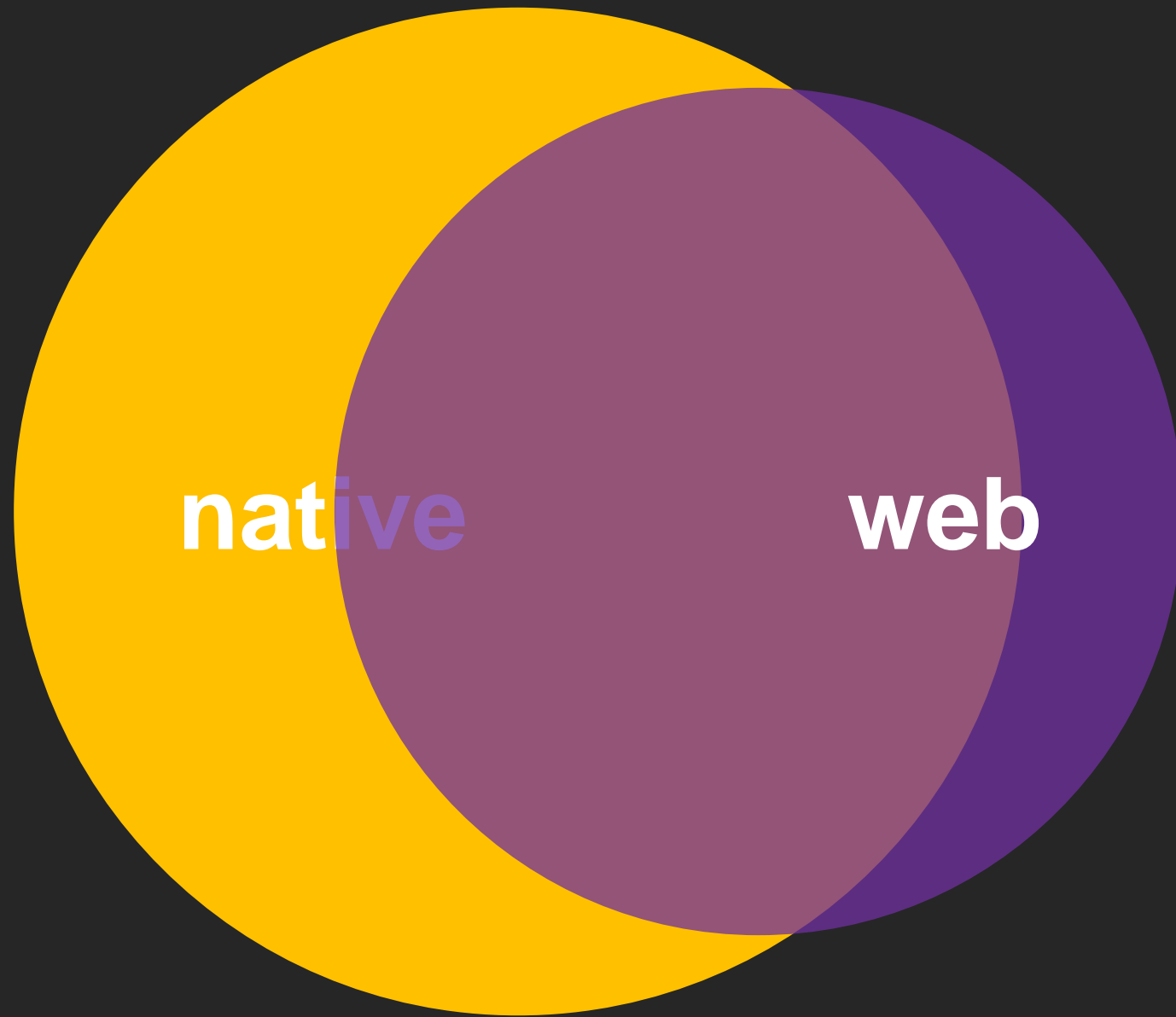
App Cache
Geolocation
Ambient Light API
Battery Status API
Web Authentication
Web Animations
Web Notifications
Speech Recognition
Speech Synthesis
Web Bluetooth



#msedgesummit



#msedgesummit



#msedgesummit

- ⚡ Start and run really **REALLY** fast!
- 📦 Work across a variety of devices, screen sizes, etc.
- 👤 Run over HTTPs
- ☎ Work offline / reliably under poor network conditions
- 💬 Can send push notifications
- 🎮 Provide better device integration



⚡ Start and run really **REALLY** fast!

📁 Work across a variety of devices, screen sizes, etc.

👉 Run over HTTPs

☎ Work offline / reliably under poor network conditions

💬 Can send push notifications

🎮 Provide better device integration



#msedgesummit

⚡ Start and run really **REALLY** fast!

📁 Work across a variety of devices, screen sizes, etc.

👉 Run over HTTPs

☎ Work offline / reliably under poor network conditions

💬 Can send push notifications

🎮 Provide better device integration



#msedgesummit


←Weather

Forecast

↺☆📌🌙⋮🔍

🏠🌀🌊🌟📅😊👤⚙️

Honolulu, HI

87°

F
C

Partly Sunny

Updated as of 8:40 PM

Feels Like 88°

Wind ↺ 17 mph

Visibility 9 mi


Barometer 29.95 in

Humidity 46%

Dew Point 64°

Daily


Thu 7



88° 77°

Mostly Sunny


Fri 8



87° 76°

Mostly Sunny


Sat 9



86° 75°

Partly Sunny


Sun 10



87° 76°

Partly Sunny

Mon 11

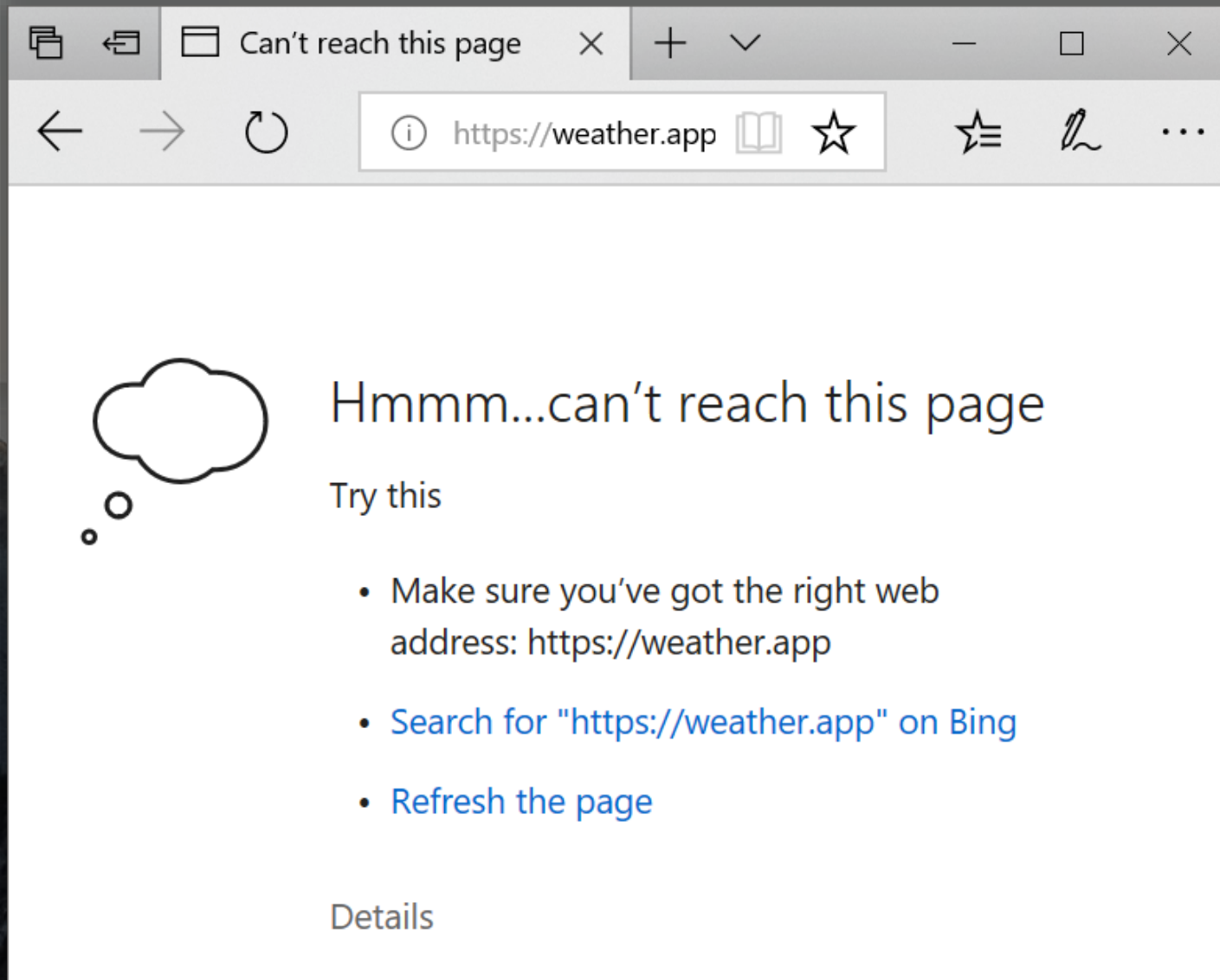


87° 76°

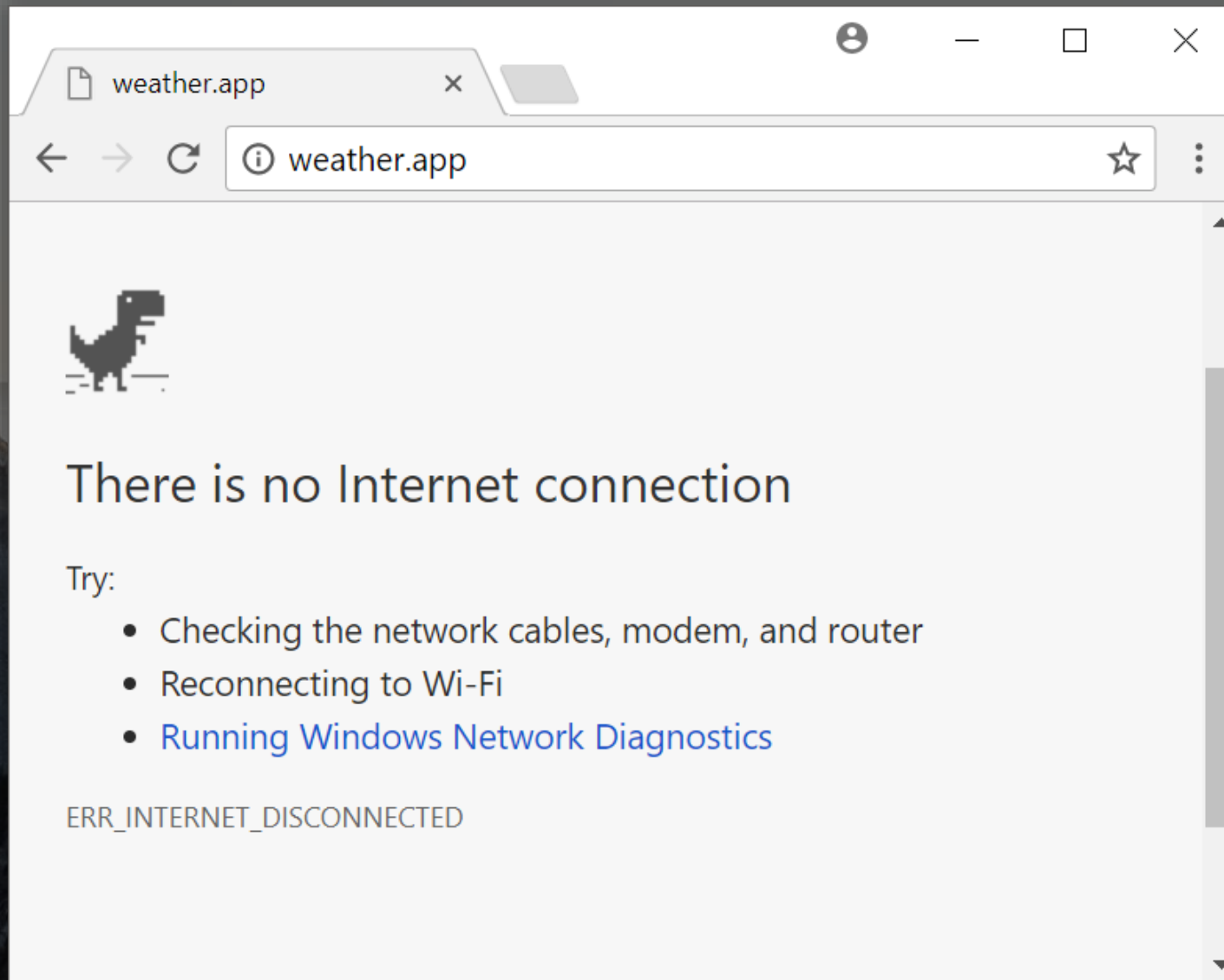
Mostly Sunny



#msedgesummit



#msedgesummit



#msedgesummit



Start and run really **REALLY** fast!



Work across a variety of devices, screen sizes, etc.



Run over HTTPs



Work offline / reliably under poor network conditions



Can send push notifications



Provide better device integration



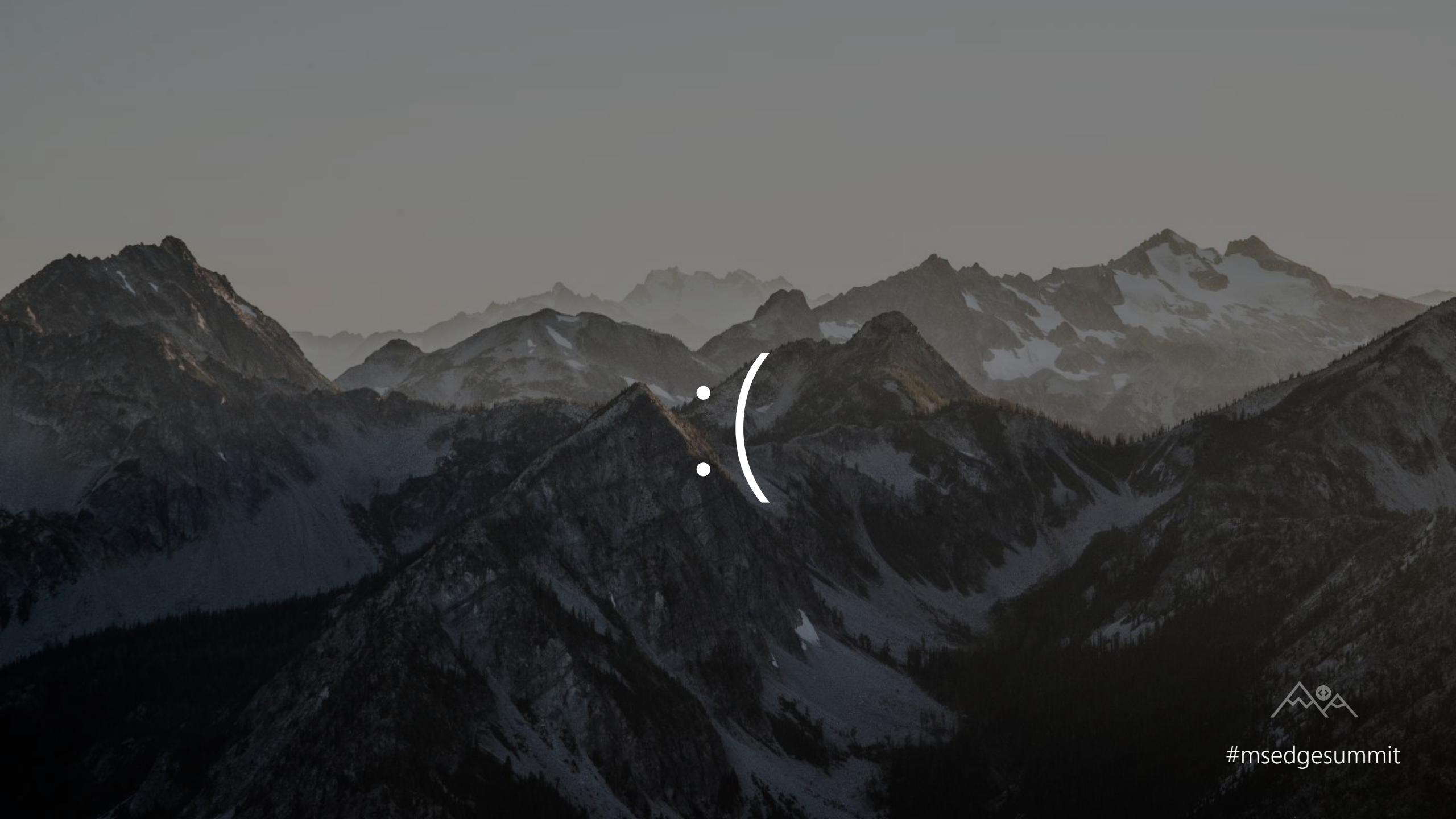
#msedgesummit



WEATHER ADVISORY
EXTREME HEAT UNTIL TOMORROW



#msedgesummit



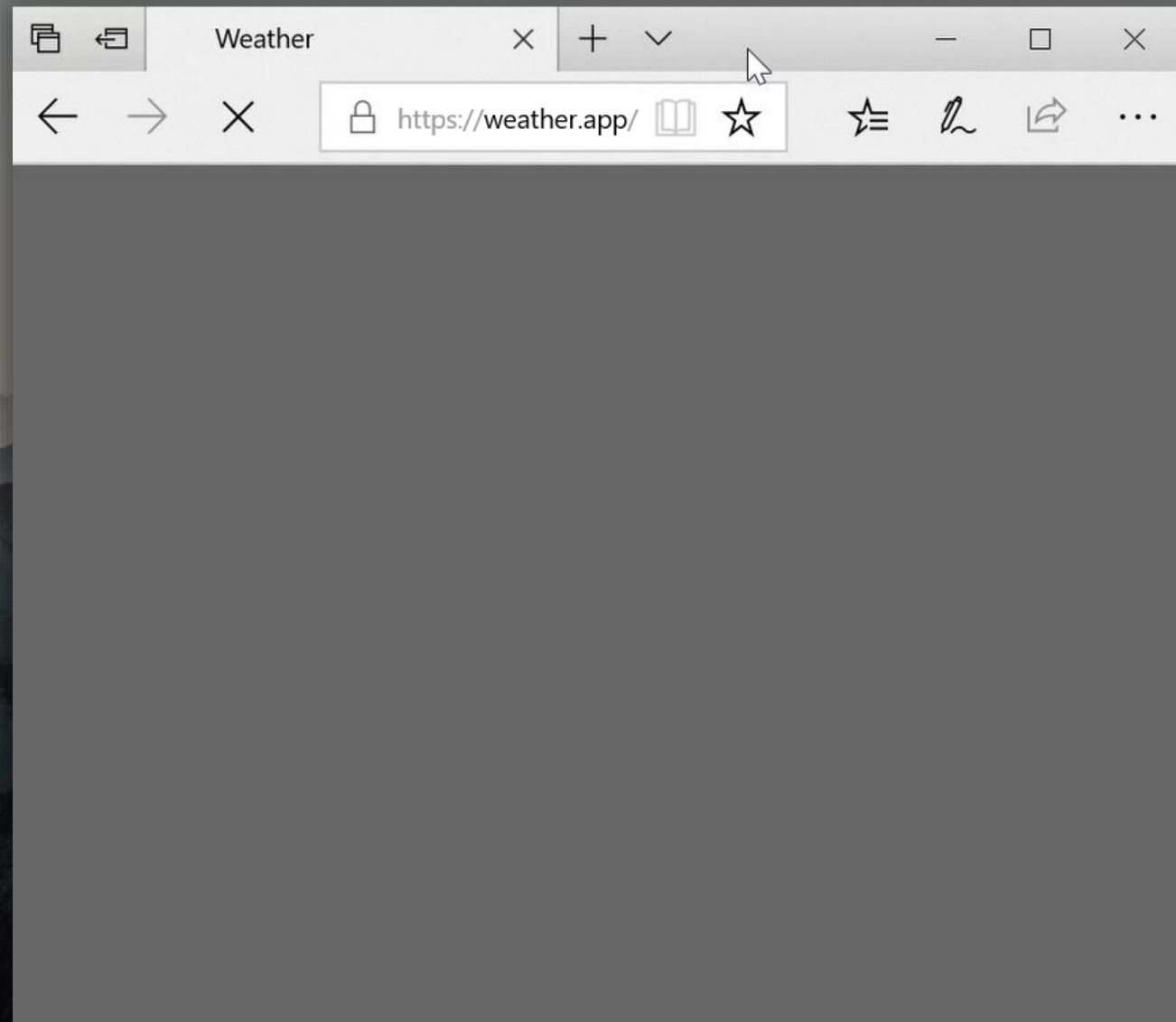
#msedgesummit

By 2012

we were able to do offline on the web



#msedgesummit



#msedgesummit

 index.html


```
1. <html manifest="appcache.manifest">
2.     ...
3. </html>
```



 index.html


```
1. <html manifest="appcache.manifest">
2.     ...
3. </html>
```



 appcache.manifest


```
1.  CACHE MANIFEST
2.  # 2017-09-01 v2
3.
4.  CACHE:
5.  /styles.css
6.  /main.js
7.  /sunny-bg.png
8.
9.  NETWORK:
10. *
11.
12. FALLBACK:
13. /index.html /fallback.html
```



 appcache.manifest


```
1.  CACHE MANIFEST
2.  # 2017-09-01 v2
3.
4.  CACHE:
5.  /styles.css
6.  /main.js
7.  /sunny-bg.png
8.
9.  NETWORK:
10. *
11.
12. FALLBACK:
13. /index.html /fallback.html
```



 appcache.manifest


```
1.  CACHE MANIFEST
2.  # 2017-09-01 v2
3.
4.  CACHE:
5.  /styles.css
6.  /main.js
7.  /sunny-bg.png
8.
9.  NETWORK:
10. *
11.
12. FALLBACK:
13. /index.html /fallback.html
```



 appcache.manifest

```
1.  CACHE MANIFEST
2.  # 2017-09-01 v2
3.
4.  CACHE:
5.  /styles.css
6.  /main.js
7.  /sunny-bg.png
8.
9.  NETWORK:
10. *
11.
12. FALLBACK:
13. /index.html /fallback.html
```



 appcache.manifest

```
1.  CACHE MANIFEST
2.  # 2017-09-01 v2
3.
4.  CACHE:
5.  /styles.css
6.  /main.js
7.  /sunny-bg.png
8.
9.  NETWORK:
10. *
11.
12. FALLBACK:
13. /index.html /fallback.html
```



“ Application Cache is a **REDACTED** .”

— JAKE ARCHIBALD

tinyurl.com/ydaabrow



#msedgesummit

APP CACHE GOTCHAS!

must change manifest to update

caching manifest file will make site stale

files come from AppCache even if online

statically defined: no dynamic caching

“magical” and difficult to debug



#msedgesummit

APP CACHE
GOTCHAS!

No control!



#msedgesummit

App Cache is useful
but we can do better!



#msedgesummit

What do we have to do?

Give the control back to you!



#msedgesummit

First,
we have to start with network requests

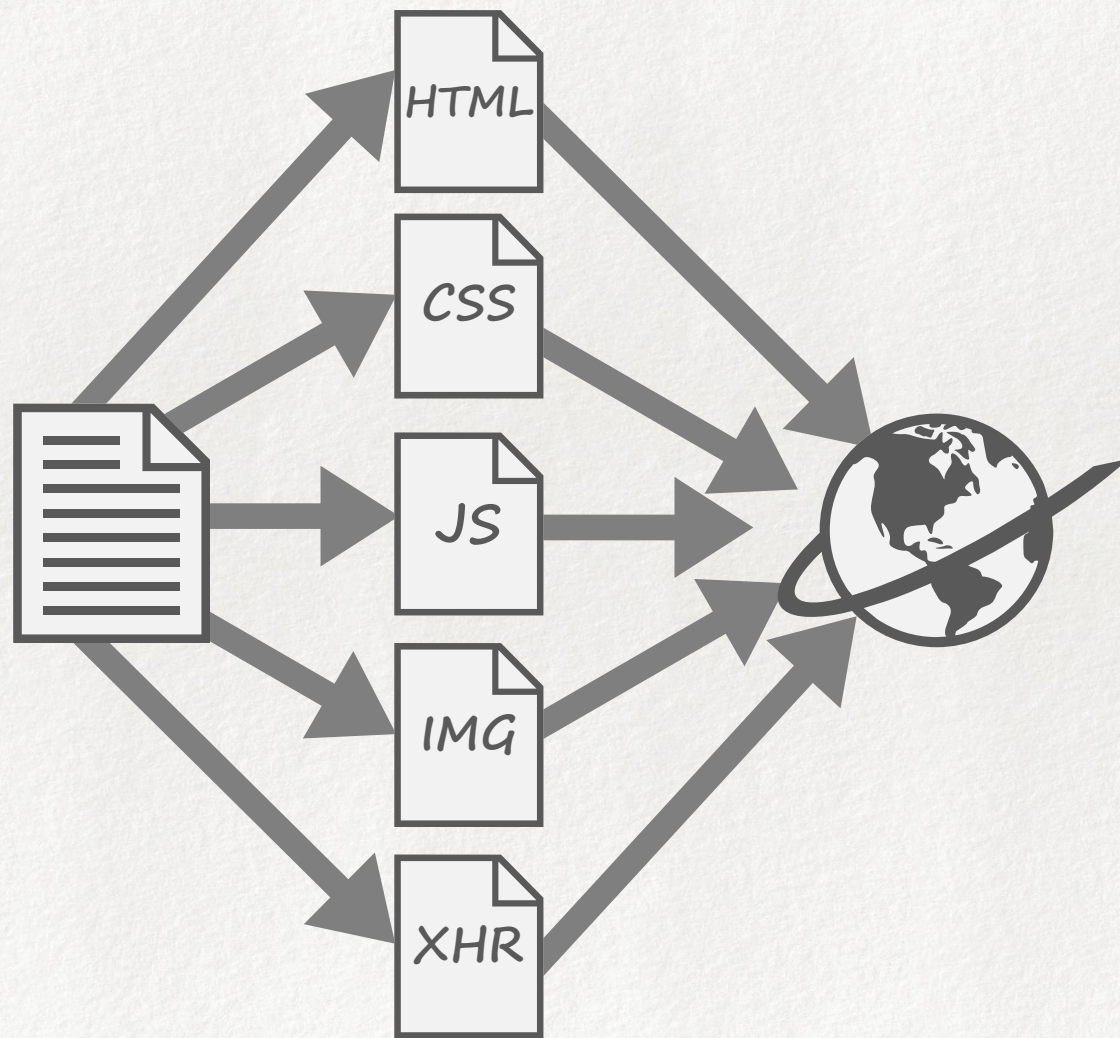


#msedgesummit

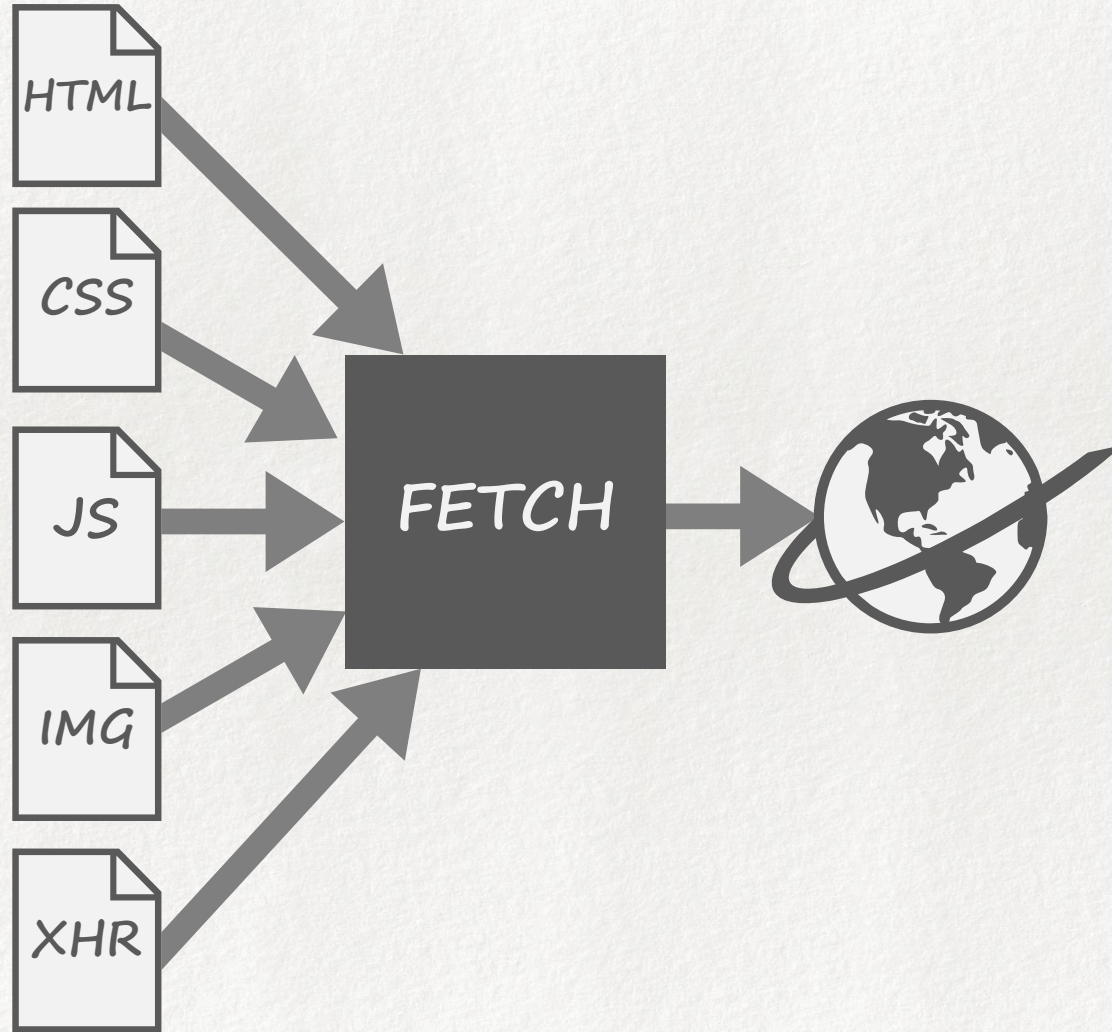
Fetch!



#msedgesummit



#msedgesummit



#msedgesummit

Exposed to JavaScript

fetch(), Request and Response



#msedgesummit

 index.html

```
/* Fetch example */
```

```
1.  fetch('weather.json')
2.    .then(function(response) {
3.      if (response.headers.get('content-type') == 'application/json') {
4.        return response.json();
5.      } else {
6.        throw new TypeError();
7.      }
8.    })
9.    .then(processJSON);
```



 index.html

```
/* Fetch example */
```

```
1.  fetch('weather.json')
2.    .then(function(response) {
3.      if (response.headers.get('content-type') == 'application/json') {
4.        return response.json();
5.      } else {
6.        throw new TypeError();
7.      }
8.    })
9.    .then(processJSON);
```



 index.html

```
/* Fetch example */
```

```
1.  fetch('weather.json')
2.    .then(function(response) {
3.        if (response.headers.get('content-type') == 'application/json') {
4.            return response.json();
5.        } else {
6.            throw new TypeError();
7.        }
8.    })
9.    .then(processJSON);
```



 index.html

```
/* Fetch example */
```

```
1.  fetch('weather.json')
2.    .then(function(response) {
3.      if (response.headers.get('content-type') == 'application/json') {
4.        return response.json();
5.      } else {
6.        throw new TypeError();
7.      }
8.    })
9.    .then(processJSON);
```



 index.html

```
/* Fetch example */
```

```
1.  fetch('weather.json')
2.    .then(function(response) {
3.      if (response.headers.get('content-type') == 'application/json') {
4.        return response.json();
5.      } else {
6.        throw new TypeError();
7.      }
8.    })
9.    .then(processJSON);
```

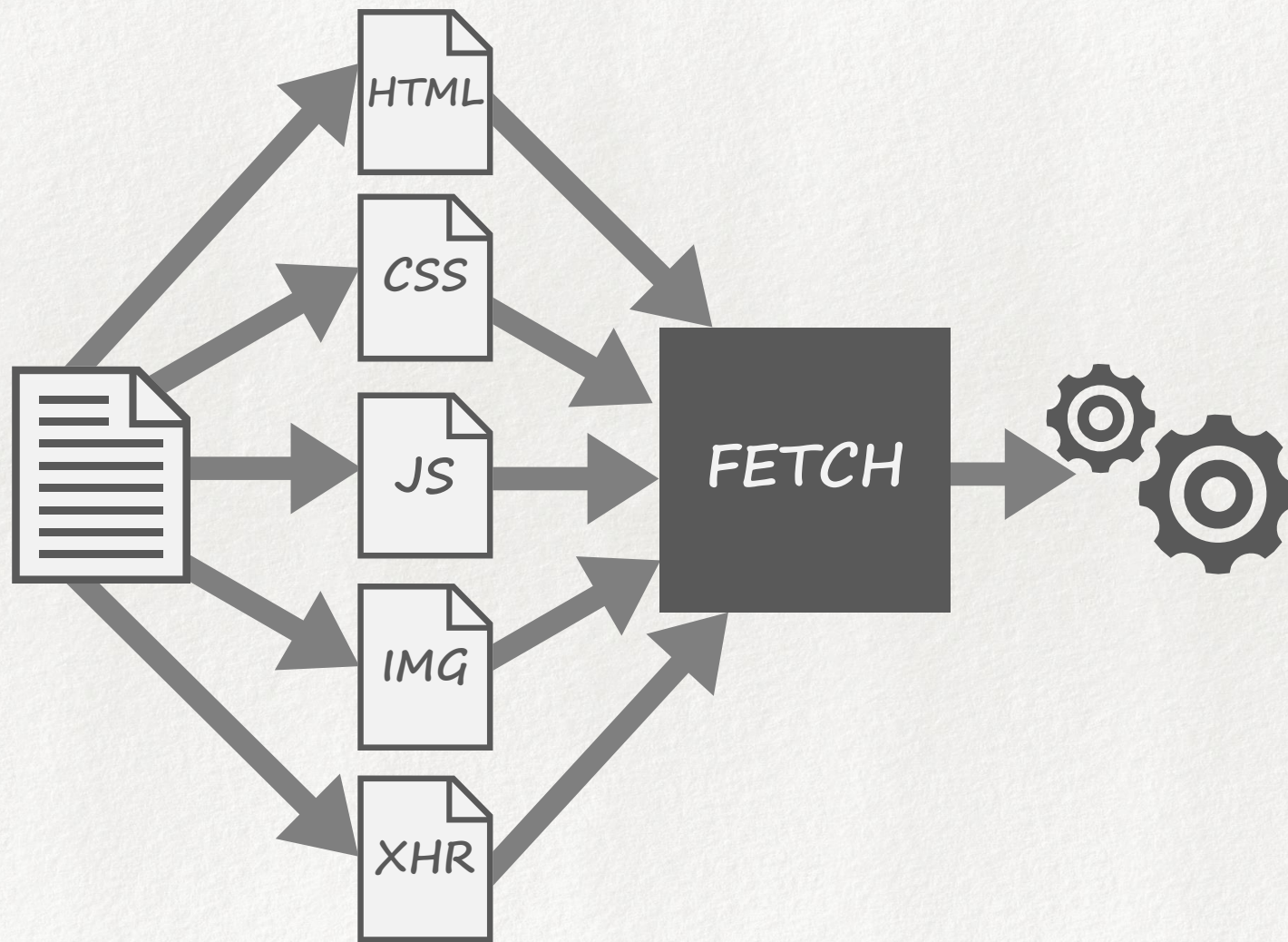


 index.html

```
/* XHR example */
```

```
1. var xhr = new XMLHttpRequest();
2. xhr.open('GET', 'weather.json');
3. xhr.responseType = 'json';
4. xhr.onload = function() {
5.     if (xhr.status == 200
6.         && xhr.getResponseHeader('Content-Type') == 'application/json') {
7.         processJSON(xhr.response);
8.     } else {
9.         throw new TypeError();
10.    }
11. }
12. xhr.send();
```





#msedgesummit

Web Workers



#msedgesummit

~~Web~~ Workers



#msedgesummit

Service Workers



#msedgesummit



#msedgesummit



```
/* intercept fetches */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         fetch(event.request);  
4.     );  
5. }
```



#msedgesummit



```
/* intercept fetches */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         fetch(event.request);  
4.     );  
5. }
```



#msedgesummit



```
/* intercept fetches */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         fetch(event.request);  
4.     );  
5. }
```



#msedgesummit

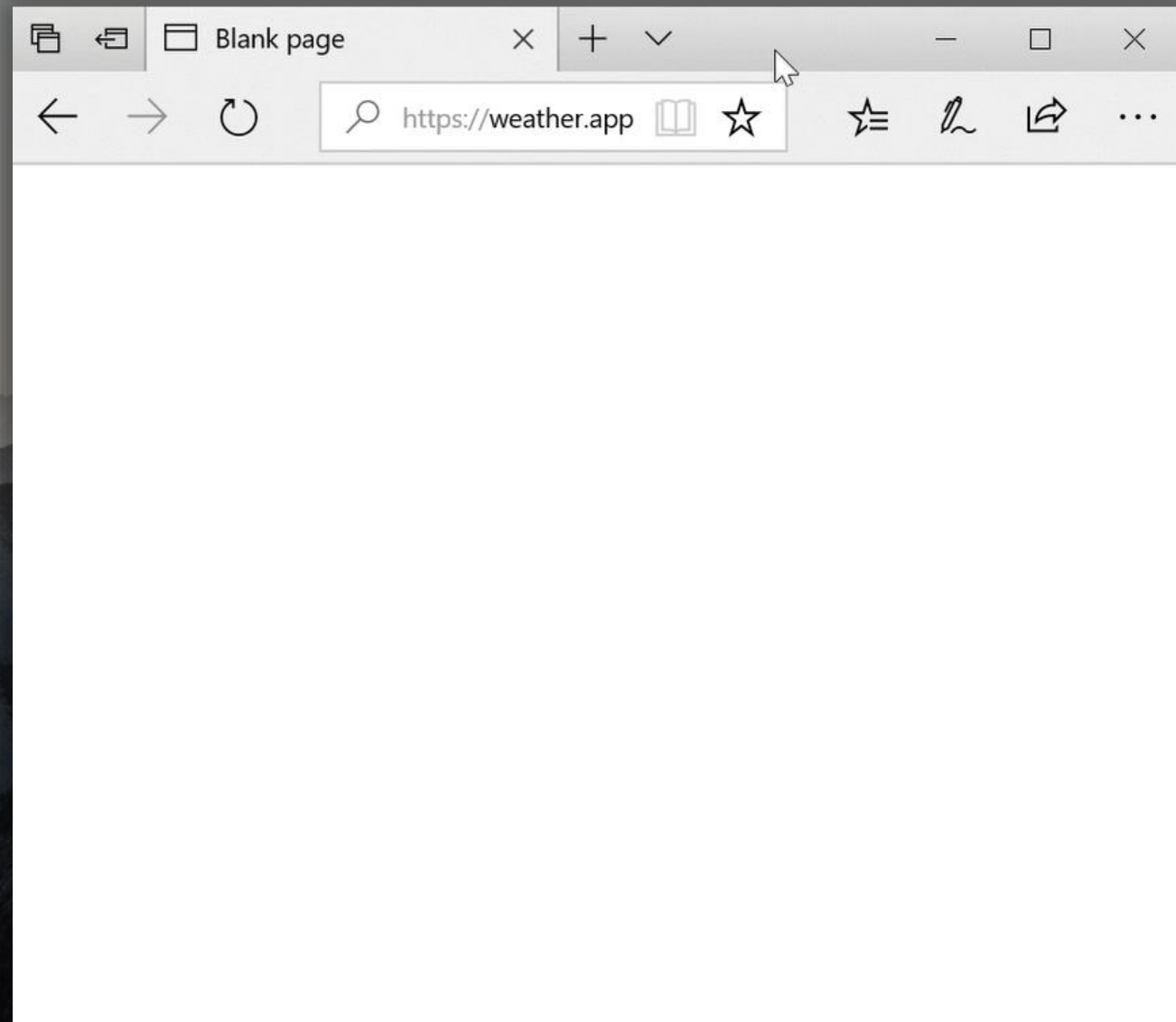


```
/* intercept fetches */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         fetch(event.request);  
4.     );  
5. }
```



#msedgesummit



#msedgesummit

We need to save Responses!



#msedgesummit

Cache APIs



#msedgesummit



#msedgesummit

CACHES API ANATOMY

key / value storage

REQUEST (URL)	RESPONSE
<code>https://weather.app/index.html</code>	<code><!doctype html></code>
<code>https://weather.app/script.js</code>	<code>console.log('test')</code>

...



#msedgesummit



Cache API

URL-addressable resources

— or —



Indexed DB

Other data



#msedgesummit

OK! Now we're in business!



#msedgesummit

 index.html

```
/* register the service worker */

1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



 index.html

```
/* register the service worker */
```

```
1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



 index.html

```
/* register the service worker */
```

```
1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



 index.html

```
/* register the service worker */
```

```
1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



 index.html

```
/* register the service worker */
```

```
1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



 index.html

```
/* register the service worker */
```

```
1. navigator.serviceWorker.register('sw.js', {scope: '/'})
2.   .then(
3.     function (registration) {
4.       console.log('Service worker registered!');
5.     },
6.     function (err) {
7.       console.error('Installation failed!', err);
8.     }
9.   );
```



Service Worker Lifecycle

Installing

Beginning of registration – set everything up



#msedgesummit



```
/* install and prime the caches */

1. self.oninstall = function(event) {
2.     event.waitUntil(
3.         caches.open('static-v1').then(function(cache) {
4.             return cache.addAll([
5.                 '/',
6.                 '/index.html',
7.                 '/styles.css',
8.                 '/main.js',
9.                 '/fallback.html'
10.            ]);
11.        })
12.    );
13. }
```





```
/* install and prime the caches */
```

```
1. self.oninstall = function(event) {  
2.     event.waitUntil(  
3.         caches.open('static-v1').then(function(cache) {  
4.             return cache.addAll([  
5.                 '/',  
6.                 '/index.html',  
7.                 '/styles.css',  
8.                 '/main.js',  
9.                 '/fallback.html'  
10.            ]);  
11.         })  
12.     );  
13. }
```





```
/* install and prime the caches */
```

```
1. self.oninstall = function(event) {  
2.     event.waitUntil(  
3.         caches.open('static-v1').then(function(cache) {  
4.             return cache.addAll([  
5.                 '/',  
6.                 '/index.html',  
7.                 '/styles.css',  
8.                 '/main.js',  
9.                 '/fallback.html'  
10.            ]);  
11.         })  
12.     );  
13. }
```





```
/* install and prime the caches */
```

```
1. self.oninstall = function(event) {  
2.     event.waitUntil(  
3.         caches.open('static-v1').then(function(cache) {  
4.             return cache.addAll([  
5.                 '/',  
6.                 '/index.html',  
7.                 '/styles.css',  
8.                 '/main.js',  
9.                 '/fallback.html'  
10.            ]);  
11.         })  
12.     );  
13. }
```



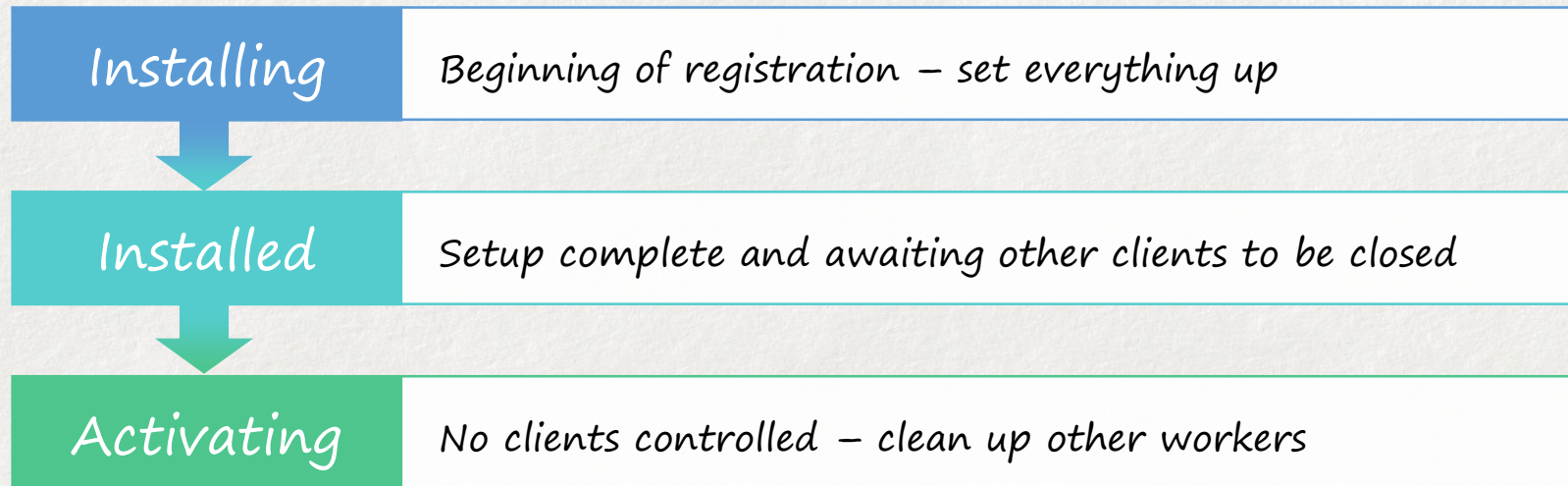


```
/* install and prime the caches */
```

```
1. self.oninstall = function(event) {  
2.     event.waitUntil(  
3.         caches.open('static-v1').then(function(cache) {  
4.             return cache.addAll([  
5.                 '/',  
6.                 '/index.html',  
7.                 '/styles.css',  
8.                 '/main.js',  
9.                 '/fallback.html'  
10.            ]);  
11.         })  
12.     );  
13. }
```



Service Worker Lifecycle





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            }));  
11.        })  
12.    );  
13. }
```





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            }));  
11.        })  
12.    );  
13. }
```





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            })));  
11.        })  
12.    );  
13. }
```





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            })));  
11.        })  
12.    );  
13. }
```





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            })));  
11.        })  
12.    );  
13. }
```





```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            }));  
11.        })  
12.    );  
13. }
```



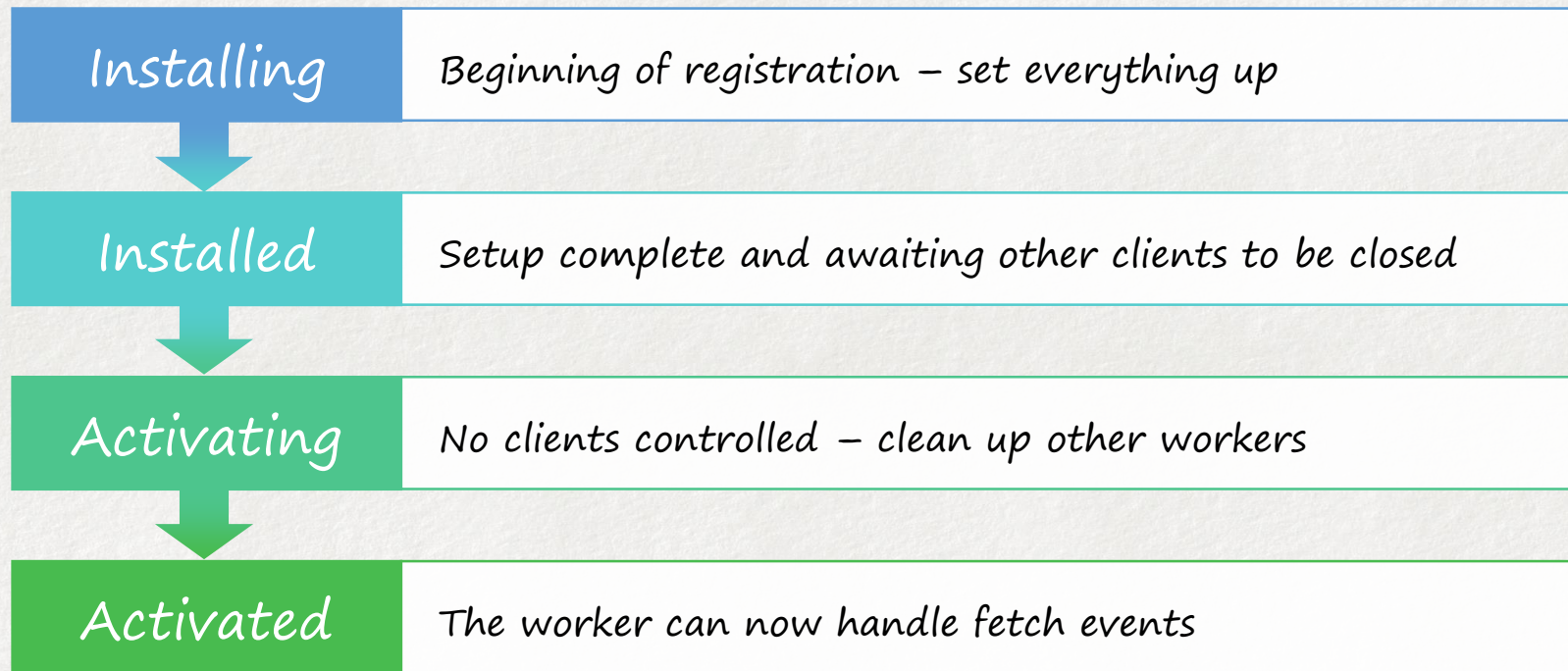


```
/* activating a service worker */
```

```
1. self.onactivate = function(event) {  
2.     var keepList = ['assets-v1'];  
3.  
4.     event.waitUntil(  
5.         caches.keys().then(function(cacheNameList) {  
6.             return Promise.all(cacheNameList.map(function(cacheName) {  
7.                 if (keepList.indexOf(cacheName) === -1) {  
8.                     return caches.delete(cacheName);  
9.                 }  
10.            }));  
11.        })  
12.    );  
13. }
```



Service Worker Lifecycle





```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





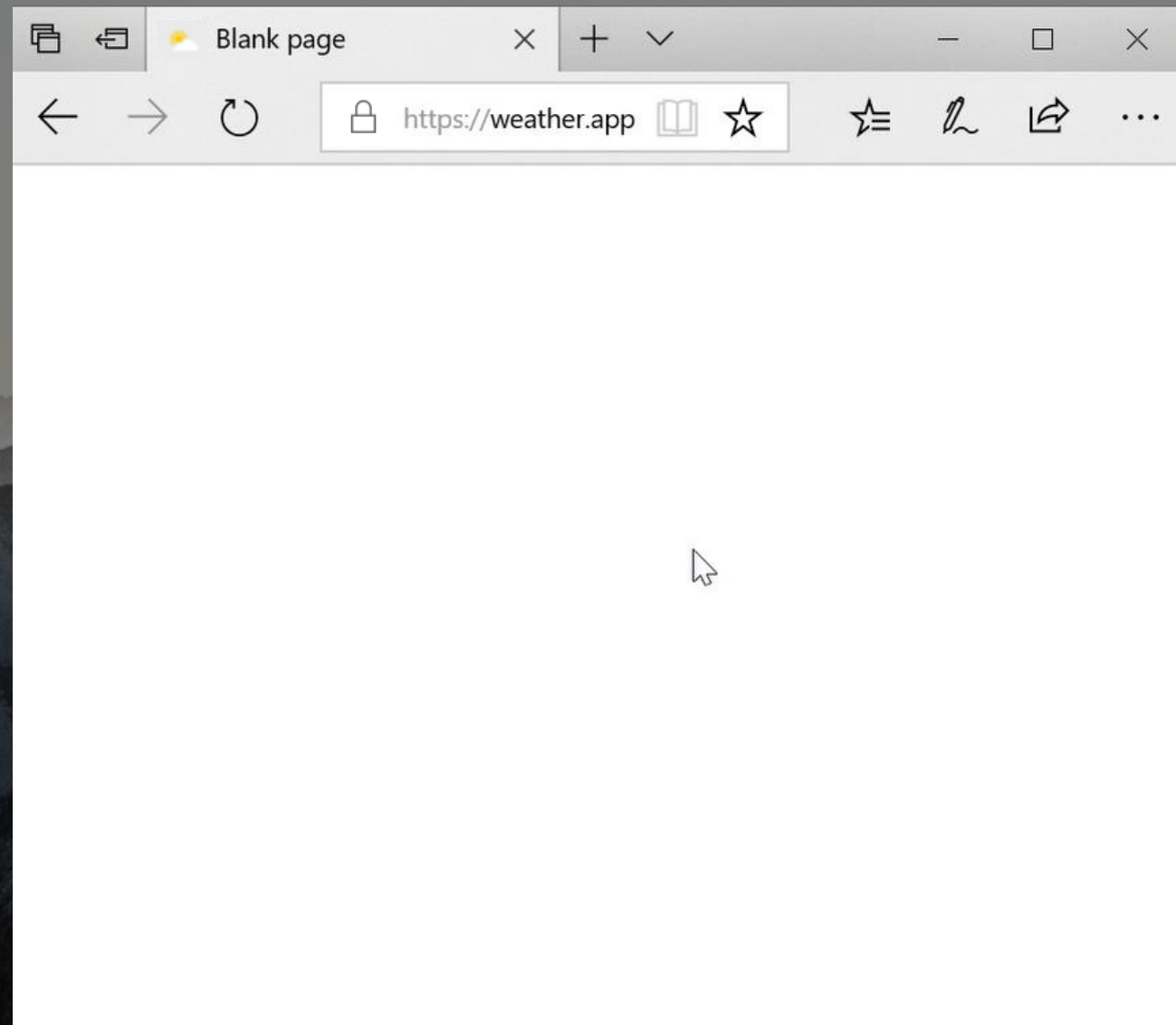
```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```



Service Worker Lifecycle



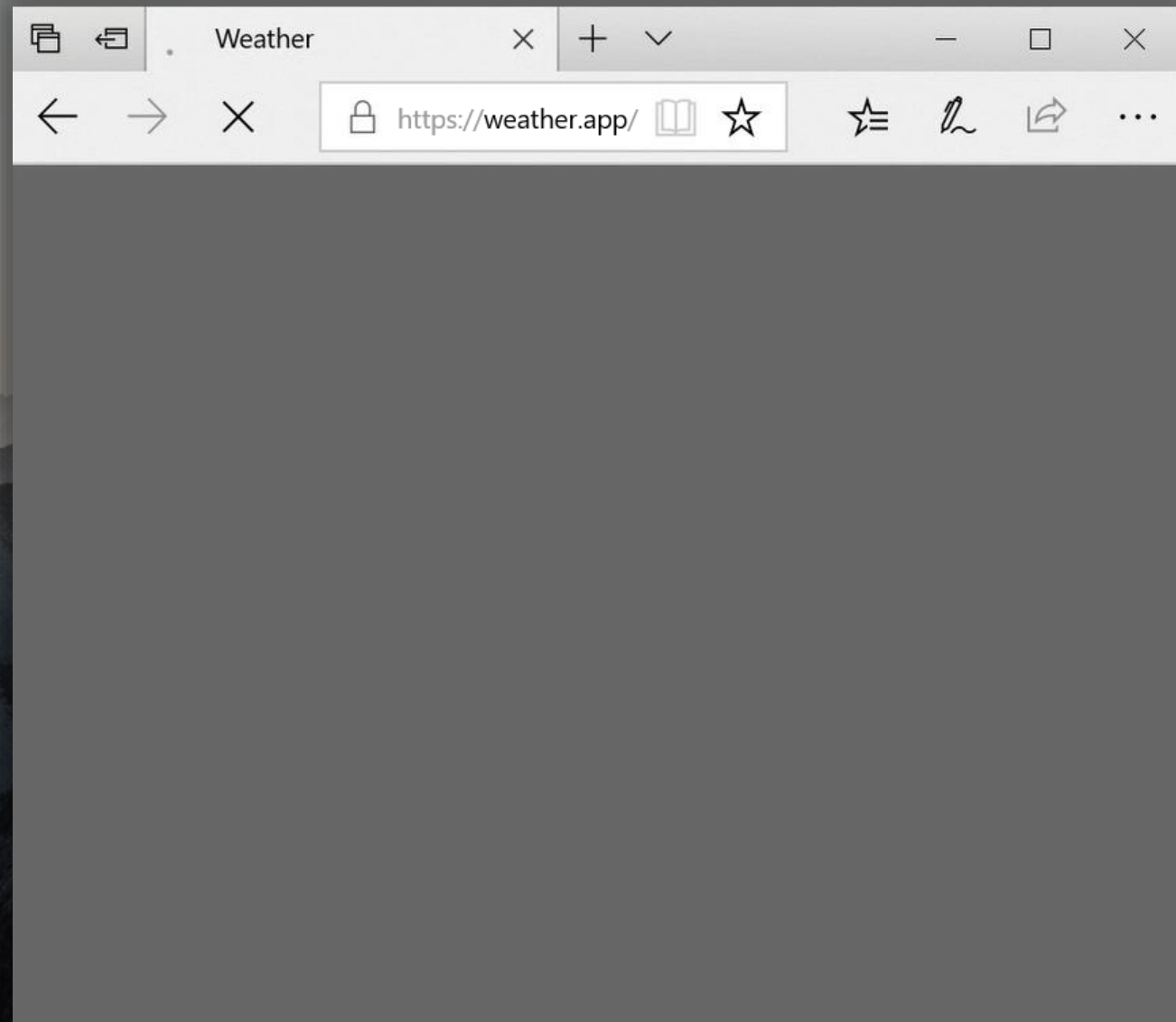


#msedgesummit

Let's cache dynamically!



#msedgesummit



#msedgesummit



```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





```
/* intercept fetches and serve from primed cache */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).catch(function() {  
5.                 return caches.match('/fallback.htm1');  
6.             });  
7.         })  
8.     );  
9. }
```





```
/* dynamic caching */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).then(function(response) {  
5.                 caches.open('dynamic').then(function(cache) {  
6.                     cache.put(event.request, response);  
7.                 });  
8.                 return response.clone();  
9.             });  
10.        })  
11.    );  
12. }
```





```
/* dynamic caching */
```

```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).then(function(response) {  
5.                 caches.open('dynamic').then(function(cache) {  
6.                     cache.put(event.request, response);  
7.                 });  
8.             return response.clone();  
9.         });  
10.    })  
11. );  
12. }
```





```
/* dynamic caching */
```

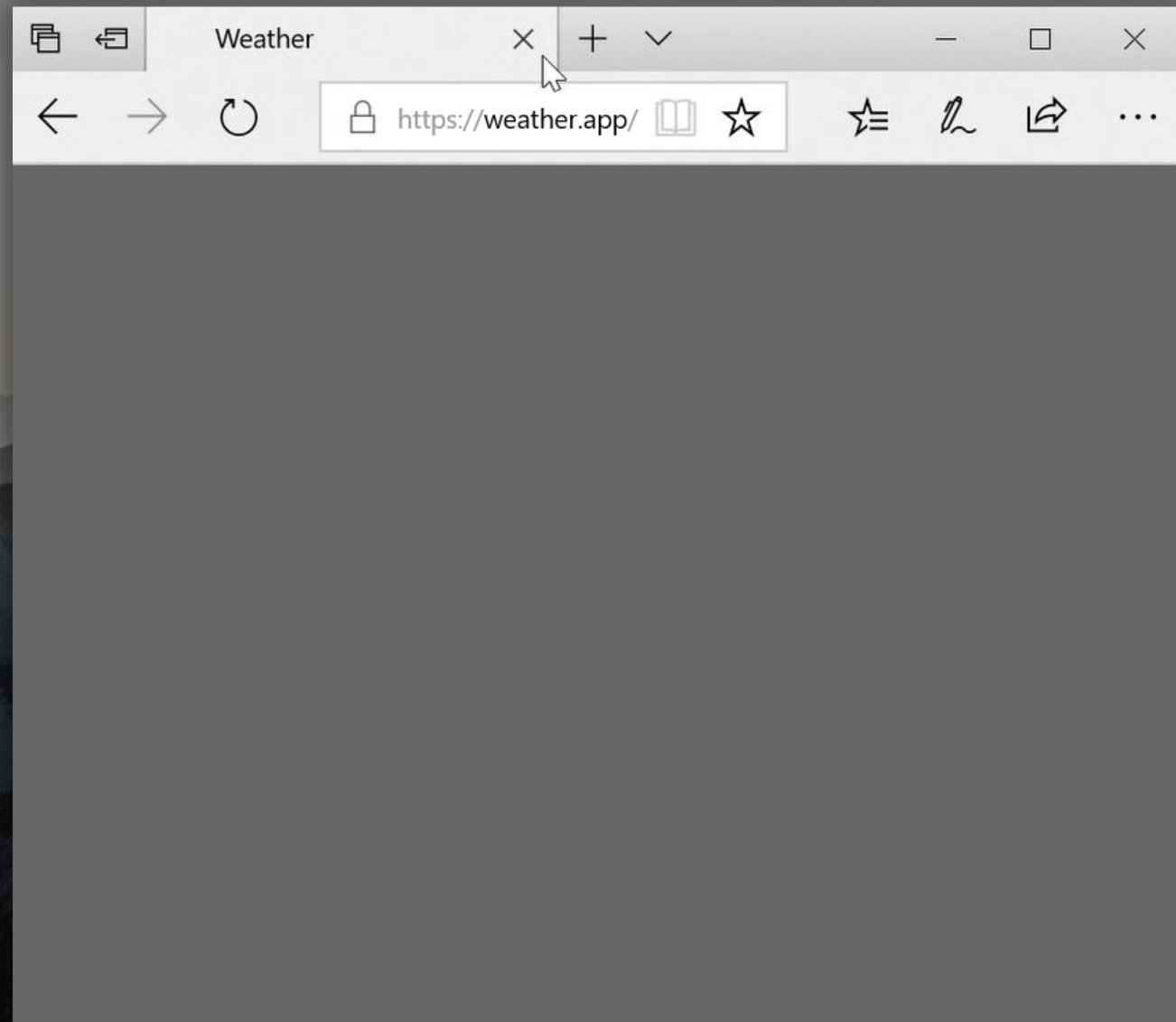
```
1. self.onfetch = function(event) {  
2.     event.respondWith(  
3.         caches.match(event.request).then(function(response) {  
4.             return response || fetch(event.request).then(function(response) {  
5.                 caches.open('dynamic').then(function(cache) {  
6.                     cache.put(event.request, response);  
7.                 });  
8.                 return response.clone();  
9.             });  
10.        })  
11.    );  
12. }
```



But... we can do better!



#msedgesummit



#msedgesummit



```
/* two paths depending on the request */
```

```
1. self.onfetch = function(event) {  
2.     if (event.request.url.indexOf('weather.json') !== -1) {  
3.         event.respondWith(fetchAndCache(event.request));  
4.     } else {  
5.         event.respondWith(  
6.             caches.match(event.request).then(function(response) {  
7.                 return response || fetchAndCache(event.request);  
8.             })  
9.         );  
10.    }  
11. }
```





```
/* two paths depending on the request */
```

```
1. self.onfetch = function(event) {  
2.     if (event.request.url.indexOf('weather.json') !== -1) {  
3.         event.respondWith(fetchAndCache(event.request));  
4.     } else {  
5.         event.respondWith(  
6.             caches.match(event.request).then(function(response) {  
7.                 return response || fetchAndCache(event.request);  
8.             })  
9.         );  
10.    }  
11. }
```





```
/* two paths depending on the request */
```

```
1. self.onfetch = function(event) {  
2.     if (event.request.url.indexOf('weather.json') !== -1) {  
3.         event.respondWith(fetchAndCache(event.request));  
4.     } else {  
5.         event.respondWith(  
6.             caches.match(event.request).then(function(response) {  
7.                 return response || fetchAndCache(event.request);  
8.             })  
9.         );  
10.    }  
11. }
```





```
/* two paths depending on the request */
```

```
1. self.onfetch = function(event) {  
2.     if (event.request.url.indexOf('weather.json') !== -1) {  
3.         event.respondWith(fetchAndCache(event.request));  
4.     } else {  
5.         event.respondWith(  
6.             caches.match(event.request).then(function(response) {  
7.                 return response || fetchAndCache(event.request);  
8.             })  
9.         );  
10.    }  
11. }
```





```
/* dynamically cache responses */  
  
1. function fetchAndCache(request) {  
2.     return caches.open('dynamic').then(function(cache) {  
3.         return fetch(request).then(function(response) {  
4.             cache.put(request, response.clone());  
5.  
6.             return response;  
7.         });  
8.     });  
9. }
```





```
/* dynamically cache responses */
```

```
1. function fetchAndCache(request) {  
2.     return caches.open('dynamic').then(function(cache) {  
3.         return fetch(request).then(function(response) {  
4.             cache.put(request, response.clone());  
5.  
6.             return response;  
7.         });  
8.     });  
9. }
```





```
/* dynamically cache responses */
```

```
1. function fetchAndCache(request) {  
2.     return caches.open('dynamic').then(function(cache) {  
3.         return fetch(request).then(function(response) {  
4.             cache.put(request, response.clone());  
5.  
6.             return response;  
7.         });  
8.     });  
9. }
```





```
/* dynamically cache responses */
```

```
1. function fetchAndCache(request) {  
2.     return caches.open('dynamic').then(function(cache) {  
3.         return fetch(request).then(function(response) {  
4.             cache.put(request, response.clone());  
5.  
6.             return response;  
7.         });  
8.     });  
9. }
```





```
/* dynamically cache responses */
```

```
1. function fetchAndCache(request) {  
2.     return caches.open('dynamic').then(function(cache) {  
3.         return fetch(request).then(function(response) {  
4.             cache.put(request, response.clone());  
5.  
6.             return response;  
7.         });  
8.     });  
9. }
```



 index.html

```
/* get fresh data from the network */  
  
1.  showLoading();  
2.  
3.  var networkDone = false;  
4.  
5.  var networkRequest = fetch('weather.json')  
6.    .then(function(response) {  
7.      return response.json();  
8.    })  
9.    .then(function(json) {  
10.     networkDone = true;  
11.     updatePage(json);  
12.   });
```



 index.html

```
/* get fresh data from the network */

1.  showLoading();
2.
3.  var networkDone = false;
4.
5.  var networkRequest = fetch('weather.json')
6.    .then(function(response) {
7.      return response.json();
8.    })
9.    .then(function(json) {
10.      networkDone = true;
11.      updatePage(json);
12.    });
```



 index.html

```
/* get fresh data from the network */  
  
1.  showLoading();  
2.  
3.  var networkDone = false;  
4.  
5.  var networkRequest = fetch('weather.json')  
6.    .then(function(response) {  
7.      return response.json();  
8.    })  
9.    .then(function(json) {  
10.     networkDone = true;  
11.     updatePage(json);  
12.   });
```



 index.html

```
/* get fresh data from the network */  
  
1.  showLoading();  
2.  
3.  var networkDone = false;  
4.  
5.  var networkRequest = fetch('weather.json')  
6.    .then(function(response) {  
7.      return response.json();  
8.    })  
9.    .then(function(json) {  
10.     networkDone = true;  
11.     updatePage(json);  
12.   });
```



 index.html

```
/* get fresh data from the network */  
  
1.  showLoading();  
2.  
3.  var networkDone = false;  
4.  
5.  var networkRequest = fetch('weather.json')  
6.    .then(function(response) {  
7.      return response.json();  
8.    })  
9.    .then(function(json) {  
10.     networkDone = true;  
11.     updatePage(json);  
12.   });
```



 index.html

```
/* get fresh data from the network */

1.  showLoading();
2.
3.  var networkDone = false;
4.
5.  var networkRequest = fetch('weather.json')
6.    .then(function(response) {
7.      return response.json();
8.    })
9.    .then(function(json) {
10.      networkDone = true;
11.      updatePage(json);
12.    });
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */

1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



 index.html

```
/* load from cache while we wait for network */
```

```
1. caches.match('weather.json')
2.   .then(function(response) {
3.     if (!response) throw Error('No data');
4.
5.     return response.json();
6.   })
7.   .then(function(json) {
8.     if (!networkDone) updatePage(json);
9.   })
10.  .catch(function() {return networkRequest;})
11.  .catch(function() {console.log('We have nothing.');})
12.  .then(hideLoading);
```



Service Worker Events

Lifetime

install
activate

Functional

fetch



#msedgesummit

Service Worker Events

Lifetime

install
activate

Functional

fetch



#msedgesummit

⚡ Start and run really **REALLY** fast!

📁 Work across a variety of devices, screen sizes, etc.

👉 Run over HTTPs

☎ Work offline / reliably under poor network conditions

💬 Can send push notifications

🎮 Provide better device integration



#msedgesummit

Service workers

A web **app model** that enables **event-driven JavaScript** to run independent of web pages



#msedgesummit

⚡ Start and run really **REALLY** fast!

📁 Work across a variety of devices, screen sizes, etc.

👉 Run over HTTPs

☎ Work offline / reliably under poor network conditions

💬 Can send push notifications

📍 Provide better device integration



#msedgesummit



Kristi Owens says



Thank you for sending that over. I'll let you know how it goes!

Monday



#msedgesummit

Weather

https://weather.app

Honolulu, HI

88°F

Partly Sunny

Updated as of 8:16 PM

Feels Like 89°

Wind E 19 mph

Barometer 29.94 in

Visibility 10 mi

Let weather.app show notifications?

Yes

No

×



#msedgesummit

Weather

← → ↻ <https://weather.app> ☆

Honolulu, HI

88°F

Partly Sunny

🔄 Updated as of 8:16 PM

Feels Like 89°
Wind E 19 mph
Barometer 29.94 in
Visibility 10 mi
Humidity 43%
Dew Point 63°



#msedgesummit



WEATHER ADVISORY
EXTREME HEAT UNTIL TOMORROW
Microsoft Edge • [weather.app](#)



#msedgesummit

Weather

← → ↻ <https://weather.app> ☆

Honolulu, HI

88°F

Partly Sunny

🔄 Updated as of 8:16 PM

Feels Like 89°
Wind E 19 mph
Barometer 29.94 in
Visibility 10 mi
Humidity 43%
Dew Point 63°



#msedgesummit

 index.html

```
/* register sw and push */
```

```
1. navigator.serviceWorker.register('sw.js')
2.   .then(function(reg) {
3.     return reg.pushManager.getSubscription().then(function(subscription) {
4.       if (subscription) return subscription;
5.       return registration.pushManager.subscribe({
6.         userVisibleOnly: true, applicationServerKey: appPubkey
7.       });
8.     });
9.   })
```



 index.html

```
/* register sw and push */
```

```
1. navigator.serviceWorker.register('sw.js')
2.   .then(function(reg) {
3.     return reg.pushManager.getSubscription().then(function(subscription) {
4.       if (subscription) return subscription;
5.       return registration.pushManager.subscribe({
6.         userVisibleOnly: true, applicationServerKey: appPubkey
7.       });
8.     });
9.   })
```



 index.html

```
/* register sw and push */
```

```
1. navigator.serviceWorker.register('sw.js')
2.   .then(function(reg) {
3.     return reg.pushManager.getSubscription().then(function(subscription) {
4.       if (subscription) return subscription;
5.       return registration.pushManager.subscribe({
6.         userVisibleOnly: true, applicationServerKey: appPubkey
7.       });
8.     });
9.   })
```



 index.html

```
/* register sw and push */
```

```
1. navigator.serviceWorker.register('sw.js')
2.   .then(function(reg) {
3.     return reg.pushManager.getSubscription().then(function(subscription) {
4.       if (subscription) return subscription;
5.       return registration.pushManager.subscribe({
6.         userVisibleOnly: true, applicationServerKey: appPubkey
7.       });
8.     });
9.   })
```



 index.html

```
/* register sw and push */
```

```
1. navigator.serviceWorker.register('sw.js')
2.   .then(function(reg) {
3.     return reg.pushManager.getSubscription().then(function(subscription) {
4.       if (subscription) return subscription;
5.       return registration.pushManager.subscribe({
6.         userVisibleOnly: true, applicationServerKey: appPubkey
7.       });
8.     });
9.   })
```





```
/* handle push event */
```

```
1. self.onpush = function(event) {  
2.     var payload = event.data ? event.data.text() : 'no payload';  
3.     event.waitUntil(  
4.         registration.showNotification('WEATHER ADVISORY', {  
5.             body: payload,  
6.             icon: 'icon.png'  
7.         })  
8.     );  
9. }
```





```
/* handle push event */
```

```
1. self.onpush = function(event) {  
2.     var payload = event.data ? event.data.text() : 'no payload';  
3.     event.waitUntil(  
4.         registration.showNotification('WEATHER ADVISORY', {  
5.             body: payload,  
6.             icon: 'icon.png'  
7.         })  
8.     );  
9. }
```





```
/* handle push event */
```

```
1. self.onpush = function(event) {  
2.     var payload = event.data ? event.data.text() : 'no payload';  
3.     event.waitUntil(  
4.         registration.showNotification('WEATHER ADVISORY', {  
5.             body: payload,  
6.             icon: 'icon.png'  
7.         })  
8.     );  
9. }
```





```
/* handle push event */
```

```
1. self.onpush = function(event) {  
2.     var payload = event.data ? event.data.text() : 'no payload';  
3.     event.waitUntil(  
4.         registration.showNotification('WEATHER ADVISORY', {  
5.             body: payload,  
6.             icon: 'icon.png'  
7.         })  
8.     );  
9. }
```





```
/* handle push event */
```

```
1. self.onpush = function(event) {  
2.     var payload = event.data ? event.data.text() : 'no payload';  
3.     event.waitUntil(  
4.         registration.showNotification('WEATHER ADVISORY', {  
5.             body: payload,  
6.             icon: 'icon.png'  
7.         })  
8.     );  
9. }
```





```
/* handle notification click */
```

```
1. self.onnotificationclick = function(event) {  
2.     event.notification.close();  
3.     event.waitUntil(clients.openWindow('https://weather.app/advisory'));  
4. }
```





```
/* handle notification click */
```

```
1. self.onnotificationclick = function(event) {  
2.     event.notification.close();  
3.     event.waitUntil(clients.openWindow('https://weather.app/advisory'));  
4. }
```





```
/* handle notification click */
```

```
1. self.onnotificationclick = function(event) {  
2.     event.notification.close();  
3.     event.waitUntil(clients.openWindow('https://weather.app/advisory'));  
4. }
```





```
/* handle notification click */
```

```
1. self.onnotificationclick = function(event) {  
2.     event.notification.close();  
3.     event.waitUntil(clients.openWindow('https://weather.app/advisory'));  
4. }
```



Service Worker Events

Lifetime

install
activate

Functional

fetch
push
notificationclick



#msedgesummit

Service Worker Events

Lifetime

install
activate

Functional

fetch
push
notificationclick
sync
...



#msedgesummit

SERVICE WORKERS PROPERTIES

HTTPS-only

no DOM access

trumps App Cache

script can't be on CDN

can be terminated any time

only async requests allowed

executes separate from page

service worker to scope is 1:1

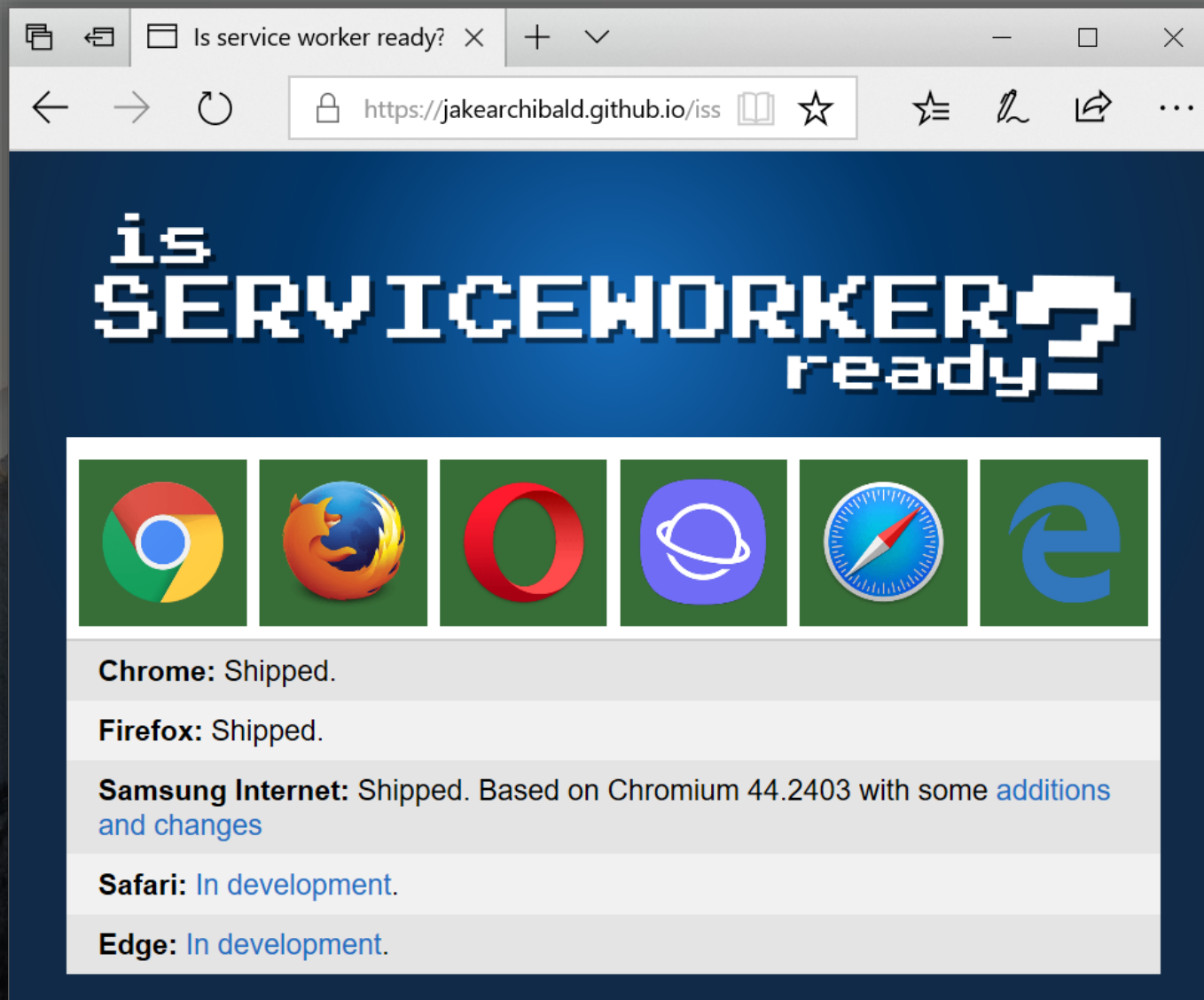


#msedgesummit

But... can I use it?



#msedgesummit



#msedgesummit



Microsoft



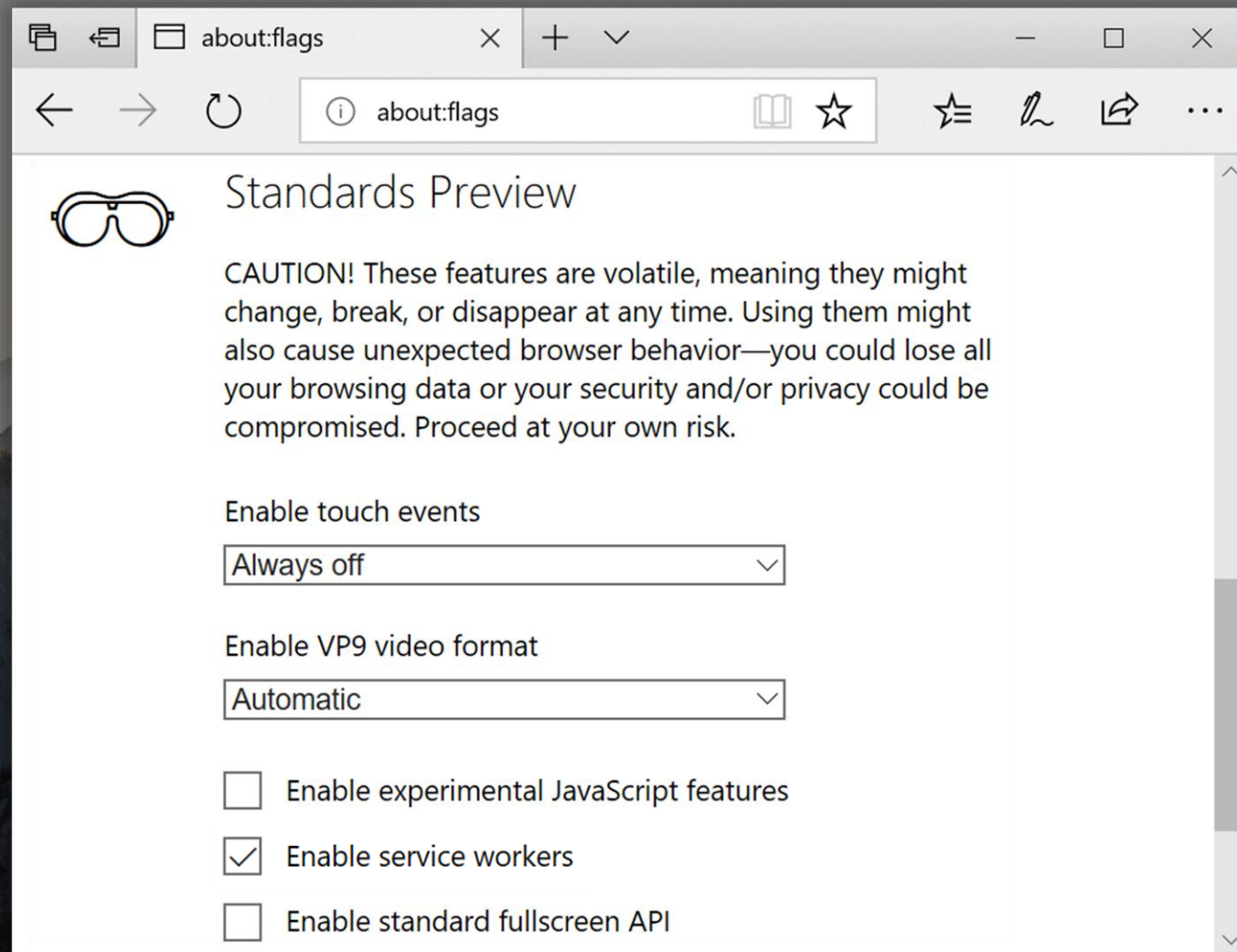
#msedgesummit

Fall Creators Update

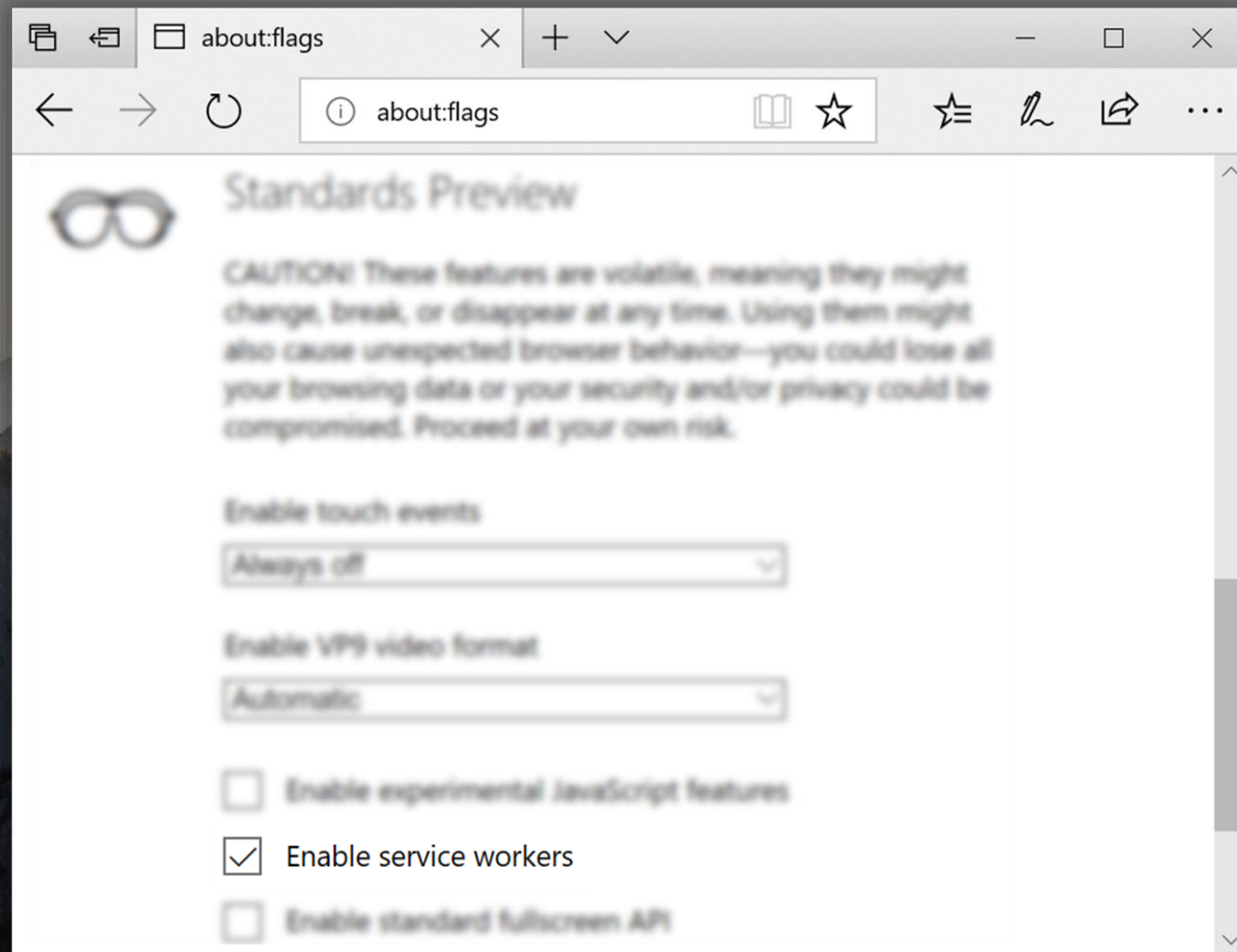
Enable in Edge using about:flags



#msedgesummit



#msedgesummit



#msedgesummit

F12

Developer Tools



#msedgesummit

localhost - F12 Developer Tools - Microsoft Edge

F12 Elements Console Debugger Network Performance Memory Emulation

localhost Service Workers

Type to filter

- Local Storage
- Session Storage
- Cookies
- Service Workers
 - localhost**
 - Cache
 - localhost
 - 10_EyeoGmbHAdblockPlus

Service Worker Overview

Scope: http://localhost/

Registration date	9/8/2017 1:59:06 PM	Push	Update	Unregister
Last Update date	9/8/2017 2:15:03 PM			
Clients	http://localhost/			

sw.js

Status	Service worker is activated and running	Stop	Inspect
Version	4CD94E13-E61A-4C0A-9D69-54B4A94CB54F		

Find (Ctrl+F)

Watches

Add...

Call... Break... DOM bre...



#msedgesummit

localhost - F12 Developer Tools - Microsoft Edge

F12 Console Debugger Network Performance Memory

Find (Ctrl+F)

sw.js

Type to filter

- Local Storage
- Session Storage
- Cookies
- Service Workers
- Cache
- (no domain)
- localhost
- sw.js**

```
1 self.oninstall = function(event) {
2   event.waitUntil(
3     caches.open('assets-v1').then(function(cache) {
4       cache.addAll([
5         '/',
6         'index.html',
7         'styles.css',
8         'main.js'
9       ])
10    })
11  );
12 }
13
14 self.onactivate = function(event) {
15   var keepList = ['assets-v1'];
16
17   event.waitUntil(
```

Watches

Add wa...

Call... Break... DOM brea...

[Worker#3 sw.j...



#msedgesummit

localhost - F12 Developer Tools - Microsoft Edge

F12 Elements Console Debugger Network Performance Memory Emulation

Find (Ctrl+F)

localhost Service Workers Cache

Type to filter

- Local Storage
- Session Storage
- Cookies
- Service Workers
 - localhost
- Cache
 - assets-v1(localhost)**
 - dynamic(localhost)
- localhost
- 10_EyeoGmbHAdblockPlus

Request url	Response body
http://localhost/	<!DOCTYPE html> <html> <head> <title>Weath...
http://localhost/index.html	<!DOCTYPE html> <html> <head> <title>Weath...
http://localhost/main.js	document.addEventListener('DOMContentLoaded'...
http://localhost/styles.css	html { font-size: 65%; overflow-y: hidden; } body {...

Watches

Add wa...

Call... Break... DOM brea...



#msedgesummit


localhost - F12 Developer Tools - Microsoft Edge

F12 Elements Console Debugger **Network** Performance Memory Emulation

Content type Find (Ctrl+F)

Name	Protocol	Method	Result	Content type	Received	Time	Initiator
http://localhost/	HTTP	GET	200 OK	text/html	(from cache)	0 s	
styles.css http://localhost/	HTTP	GET	200 OK	text/css	(from cache)	0 s	

0 errors | 2 requests

 We have detected a Service Worker on your page. Please be aware that Service Worker traffic only will show up in the Service Worker F12 instance you start from the Debugger-tab



#msedgesummit

Notifications

Action Center and Permissions



#msedgesummit



WEATHER ADVISORY
EXTREME HEAT UNTIL TOMORROW
Microsoft Edge • [weather.app](#)



#msedgesummit



WEATHER ADVISORY

EXTREME HEAT UNTIL TOMORROW

1:56p • weather.app • via Microsoft Edge



#msedgesummit

Weather

← → ↻ <https://weather.app/> ☆ ☆≡ ✍️ ↗️ ...

Honolulu, HI

88°F

Partly Sunny

🔄 Updated as of 8:02 PM

Feels Like 89°
Wind E 19 mph
Barometer 29.94 in
Visibility 10 mi
Humidity 43%
Dew Point 63°



#msedgesummit

Storage Quotas

IndexedDB, Cache APIs, and Service Workers



#msedgesummit

Quotas

IndexedDB

VOLUME SIZE	DOMAIN LIMIT	OVERALL LIMIT
≤ 8 GB	10 MB	50 MB
$> 8 - 32$ GB	50 MB	500 MB
$> 32 - 128$ GB	250 MB	4% of volume size
> 128 GB	500 MB	4% or 20 GB (whichever is smaller)



#msedgesummit

Quotas

IndexedDB & Caches API

VOLUME SIZE	DOMAIN LIMIT	OVERALL LIMIT
≤ 8 GB	20% of overall	50 MB
$> 8 - 32$ GB	"	500 MB
$> 32 - 128$ GB	"	4% of volume size
> 128 GB	"	4% or 20 GB (whichever is smaller)



#msedgesummit

Try it today!

Windows Insider Preview: enable flag in Edge



#msedgesummit

Tell us what you think!

We want to hear from you



#msedgesummit

Thank you!

DECK

aka.ms/SWSlides2017

TWITTER

@MSEdgeDev

GITHUB

aliams



#msedgesummit