

Offline First

Martin Hilscher

JUNGE HAIE

A Offline First – say what?

- ▶ UX-Design Strategy/Philosophy
- ▶ (Web-)Applications fully functional ...
- ▶ ... even without access to the network



wat

A Fallacies of distributed computing

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

A The network is reliable



A Latency is zero

L1 cache	0.5 ns	
L2 cache	7 ns	14X
Main memory	100 ns	20X
Access 1 Random Block on an SSD	150,000 ns	1500X
Ping CA → NL	150,000,000 ns	1000X

SSD access is often perceived as slow

A Bandwidth is infinite

What is the fastest way to move 2.5TB from Panama to München?



Technology

A What do we need

- ▶ Control over the Browser Cache
- ▶ Ability to intercept HTTP(S) calls

Clientside Storage

A Cookies

Pro

Worsk everywhere

Con

... the user doens't block cookies

Interface usually unconfortabel

Are transferred in full to memory

How much space am I actually allowed to use

...

⇒ We won't look at this much more

A Localstorage/Sessionstorage

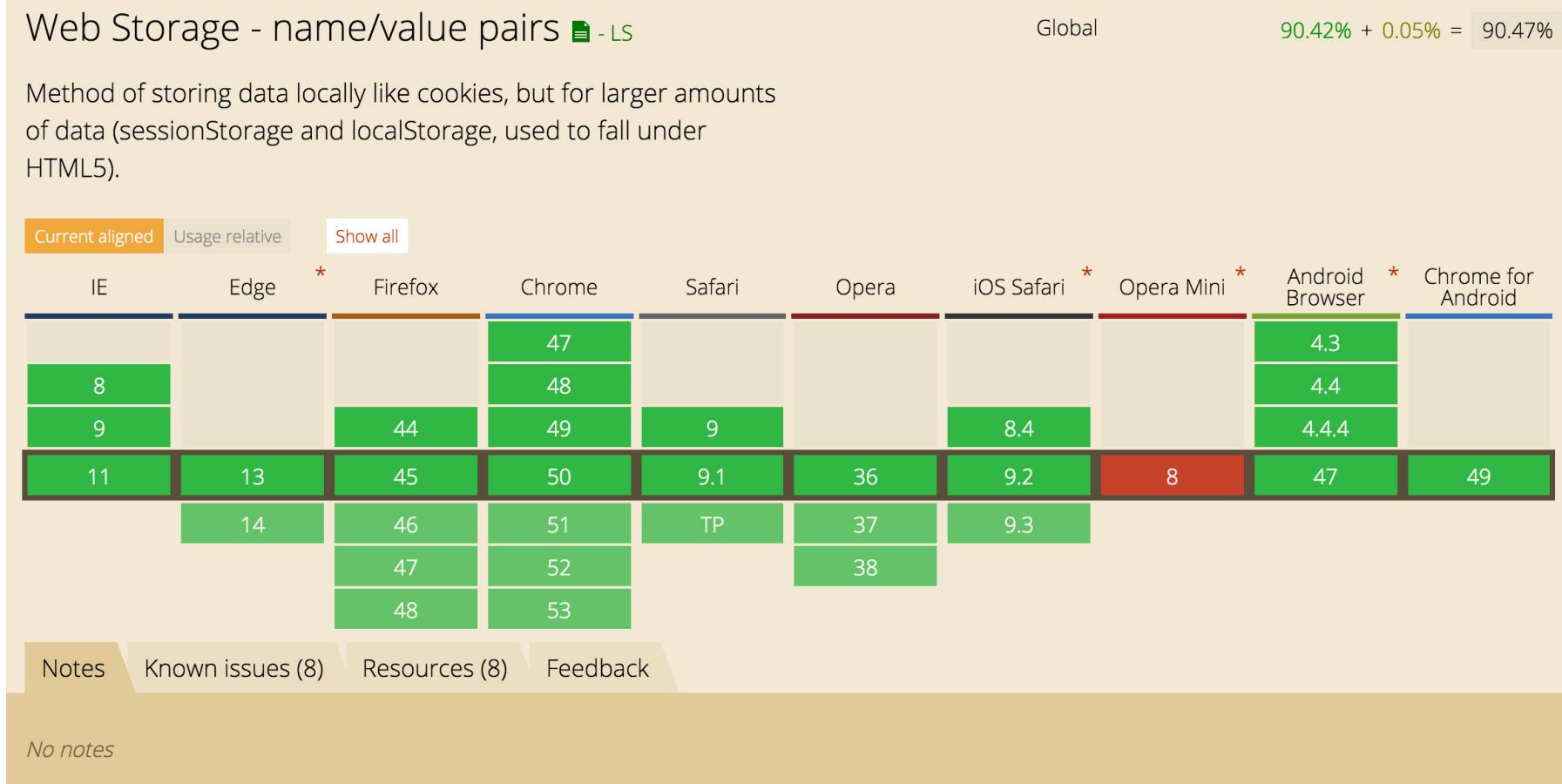
- ▶ Also called DOM-Storage or Supercookies
- ▶ Key-Value Store
- ▶ localStorage persistent
- ▶ sessionStorage only per Session

A Pros/Cons

Pros	Cons
Key-Value Store	No natives Indexing
Session Konzept already integrated	Only 5MB Storage
Broad Browser support	

Easy to use if your scenario is compatible with the limitations.

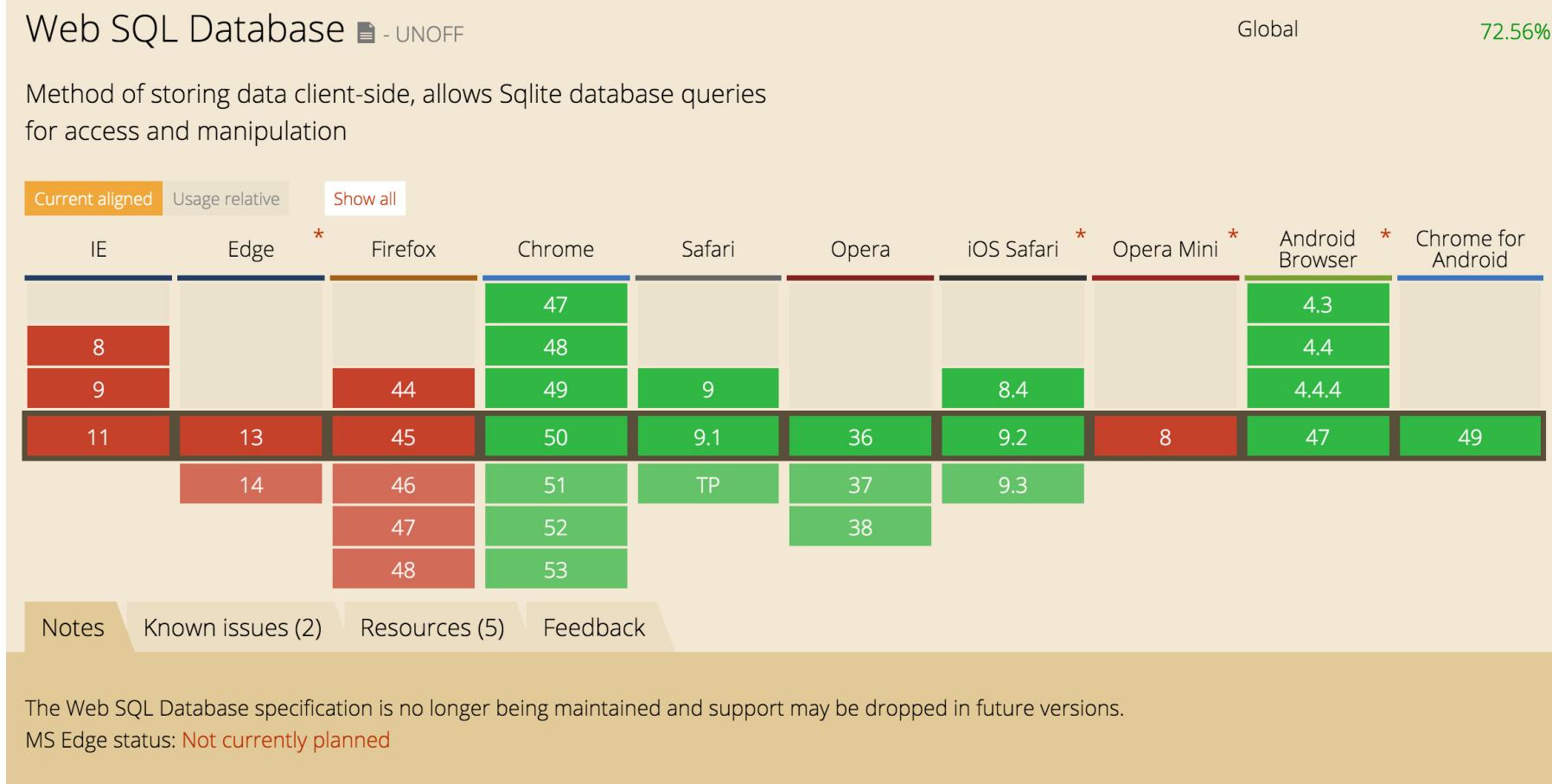
A Can I use?



A WebSQL

- ▶ Access to the SQLite in the Clients Browser
- ▶ "Most probably the worst implemented piece of SHIT ..."
- ▶ Is now deprecated
- ▶ Will be dropped in the near future

A Can I use?



What would we wish for

- ▶ Key-Value Store
- ▶ With Indexes/Indexing
- ▶ As large as possible

Say hello to INDEXDB

A Pros / Cons

Pros	Cons
-------------	-------------

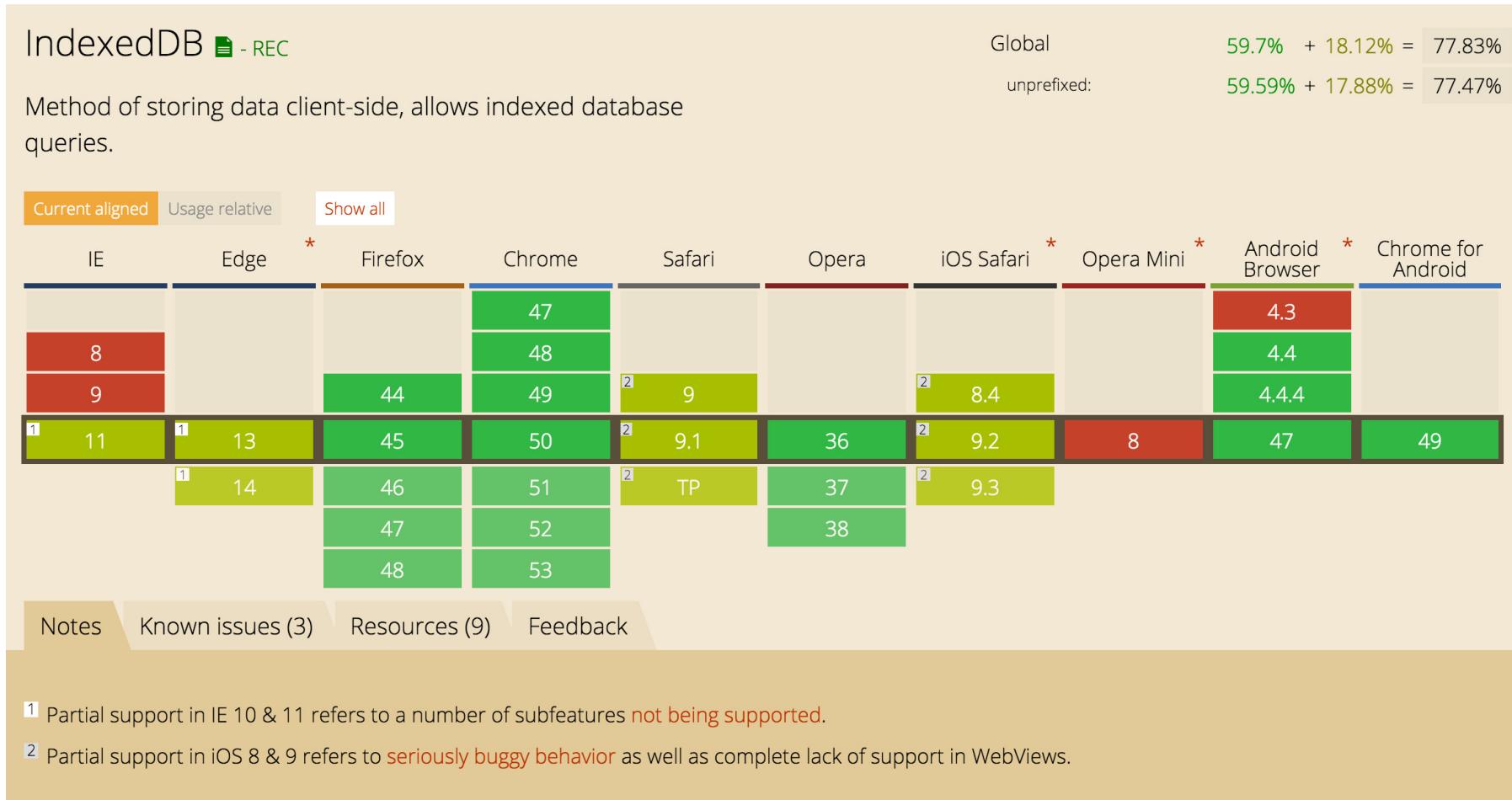
Key-Value store	API brä4\$# :-(
-----------------	-----------------

Persistent

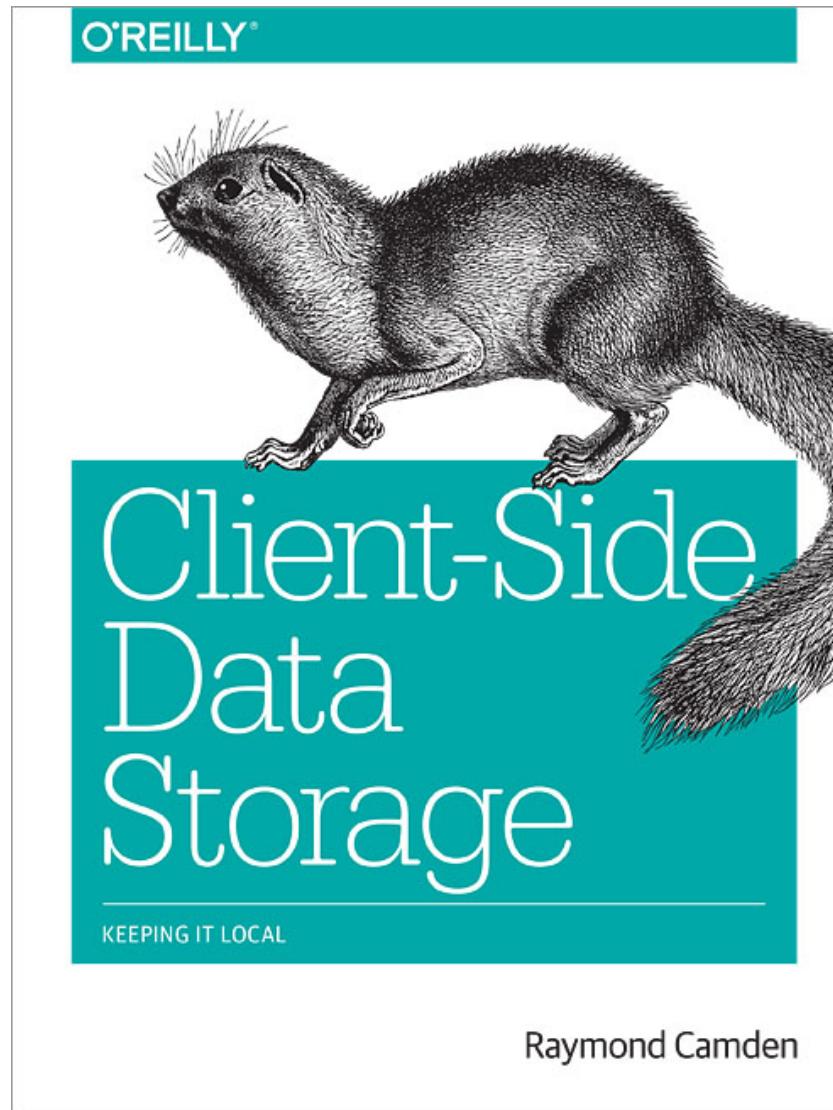
With Indexen

Except for the API exactly what we want

A Can I use?



A Further read



<http://shop.oreilly.com/product/0636920043676.do>

A PouchDB

- ▶ Key-Value Store
- ▶ Abstracts over WebSQL/IndexDB
- ▶ Implements most of the CouchDB API
- ▶ ... including the replication algorithm
- ▶ ... and the changes feed

Fine Grained HTTP Control

Appcache

A We don't talk about Appcache

"AppCache broke a lot of eggs, and failed to make an Omlet"

Jake Archibald (Coauthor der AppCache Spec)

ServiceWorker

A Serviceworker

- ▶ Developed by Google, Firefox & Samsung
- ▶ Catches all HTTP(S) calls of your Browser
- ▶ ... and gives you an Javascript API to change them

A Register

```
navigator.serviceWorker.register('/worker.js').then(function(reg) {  
    console.log(":-)", reg);  
, function (err) {  
    console.log(":-(", err)  
});
```

A Routing

```
self.addEventListener('fetch'), function(event) {
  event.respondWith(
    new Response('<h1>Sorry</h1>Nothing to see here, {
      headers: { 'Content-Type': 'text/html' }
    });
});
```

A Caching

```
self.addEventListener('install', function (event) {
  event.waitUntil(
    caches.create('cache').then(function(cache){
      return cache.addAll([
        '/staticAssets/logo.svg',
        '/staticAssets/icons.svg',
        '/js/vendor.js',
        '/js/all.js',
        '/styles/style.css'
      ]);
    })
  );
});
```

A Caching

```
self.addEventListener('fetch', function (event) {  
    event.respondWith(caches.match(event.request));  
});
```

A Caching

```
self.addEventListener('fetch', function (event) {
  event.respondWith(caches.match(event.request))
    .then (function(response) {
      return response || caches.match('fallback.html');
    });
});
```

A Caching

```
self.addEventListener('fetch', function (event) {
  event.respondWith(caches.match(event.request))
    .then (function(response) {
      return response || fetch(event.request);
    });
});
```

A Caching

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('cache').then(function(cache) {
      return cache.match(event.request).then(function (response) {
        return response || fetch(event.request).then(function(response) {
          cache.put(event.request, response.clone());
          return response;
        });
      });
    })
  );
});
```

A Caching

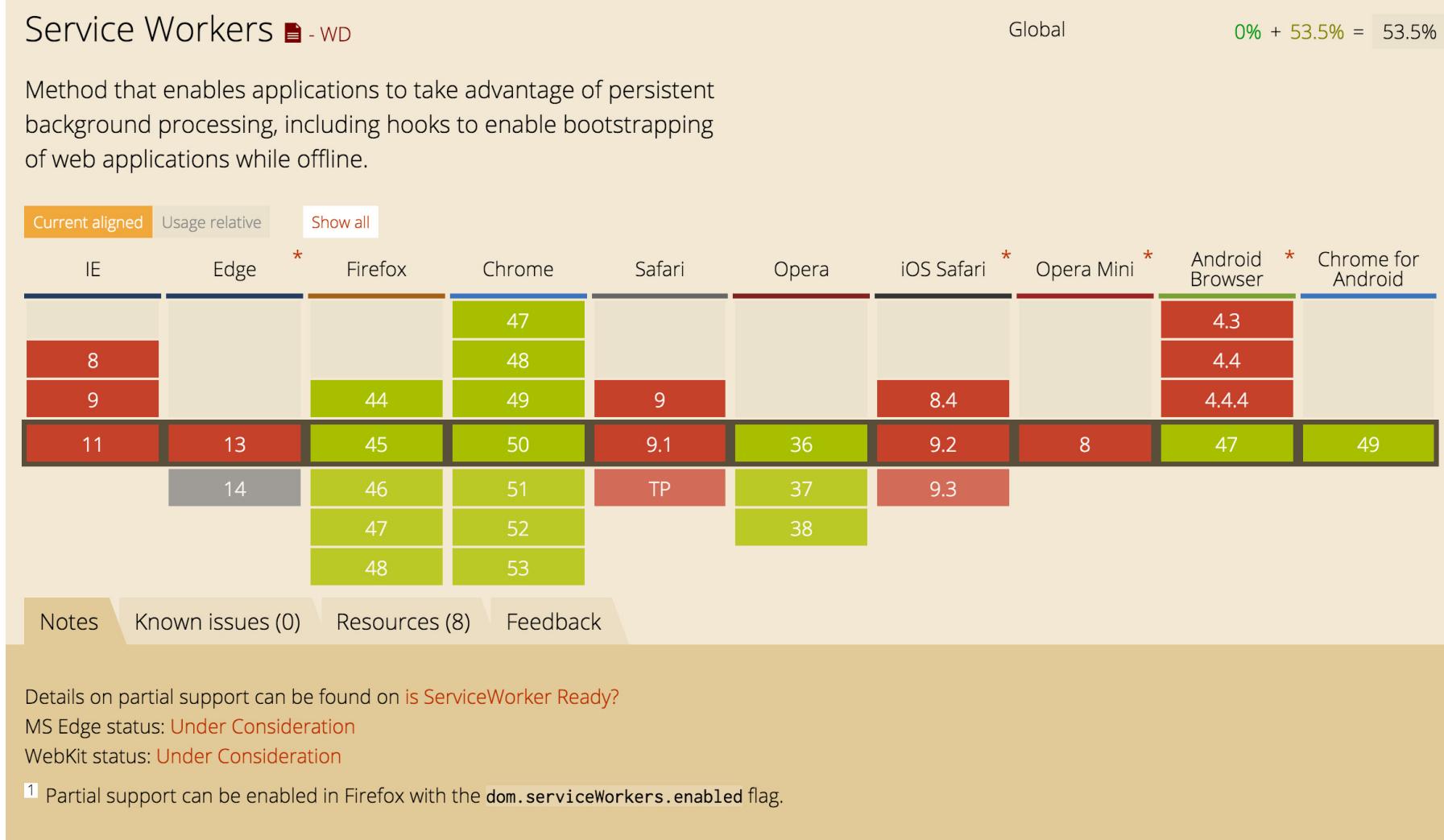
```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('cache').then(function(cache) {
      return cache.match(event.request).then(function(response) {
        var fetchPromise = fetch(event.request).then(function(networkResponse) {
          cache.put(event.request, networkResponse.clone());
          return networkResponse;
        })
        return response || fetchPromise;
      })
    })
  );
});
```

A ... and more

ServiceWorker can do even more ...

- ▶ ... react to PushNotifications
- ▶ ... update periodically in the Background

A Can I use?



UI/UX

- ▶ Sync / Save / ...
- ▶ Icons (Floppy disc for Save rly?!?)
- ▶ How do make it clear to the user that their changes are saved?
- ▶ How do we clarify that the changes have been synced to the server?
- ▶ How/Where do I add content, which has been added during an offline period?

Discuss