

# Offline First

Martin Hilscher

**JUNGE HAIE**

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



**wat**

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

# 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

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



**Technology**



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

# Clientside Storage

## Pro

Worsk everywhere

## Con

... the user doens't block cookies

Interface usually unconfortabel

Are transfered in full to memory

How much space am I actually allowed to use

...

⇒ We won't look at this much more

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

## **Pros**

Key-Value Store

Session Konzept already integrated

Broad Browser support

## **Cons**

No natives Indexing

Only 5MB Storage

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

## Web Storage - name/value pairs - LS

Global

90.42% + 0.05% = 90.47%

Method of storing data locally like cookies, but for larger amounts of data (sessionStorage and localStorage, used to fall under HTML5).

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			47					4.3	
8			48					4.4	
9		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

Notes Known issues (8) Resources (8) Feedback

No notes

- ▶ 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

## Web SQL Database UNOFF

Global

72.56%

Method of storing data client-side, allows Sqlite database queries for access and manipulation

Current aligned

Usage relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			47					4.3	
8			48					4.4	
9		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

Notes

Known issues (2)

Resources (5)

Feedback

The Web SQL Database specification is no longer being maintained and support may be dropped in future versions.

MS Edge status: **Not currently planned**



## What would we wish for

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

Say hello to INDEXDB

## Pros

Key-Value store  
Persistent  
With Indexen

## Cons

API brä4\$# :-)

Except for the API exactly what we want

## IndexedDB - REC

Method of storing data client-side, allows indexed database queries.

Global 59.7% + 18.12% = 77.83%  
 unprefixed: 59.59% + 17.88% = 77.47%

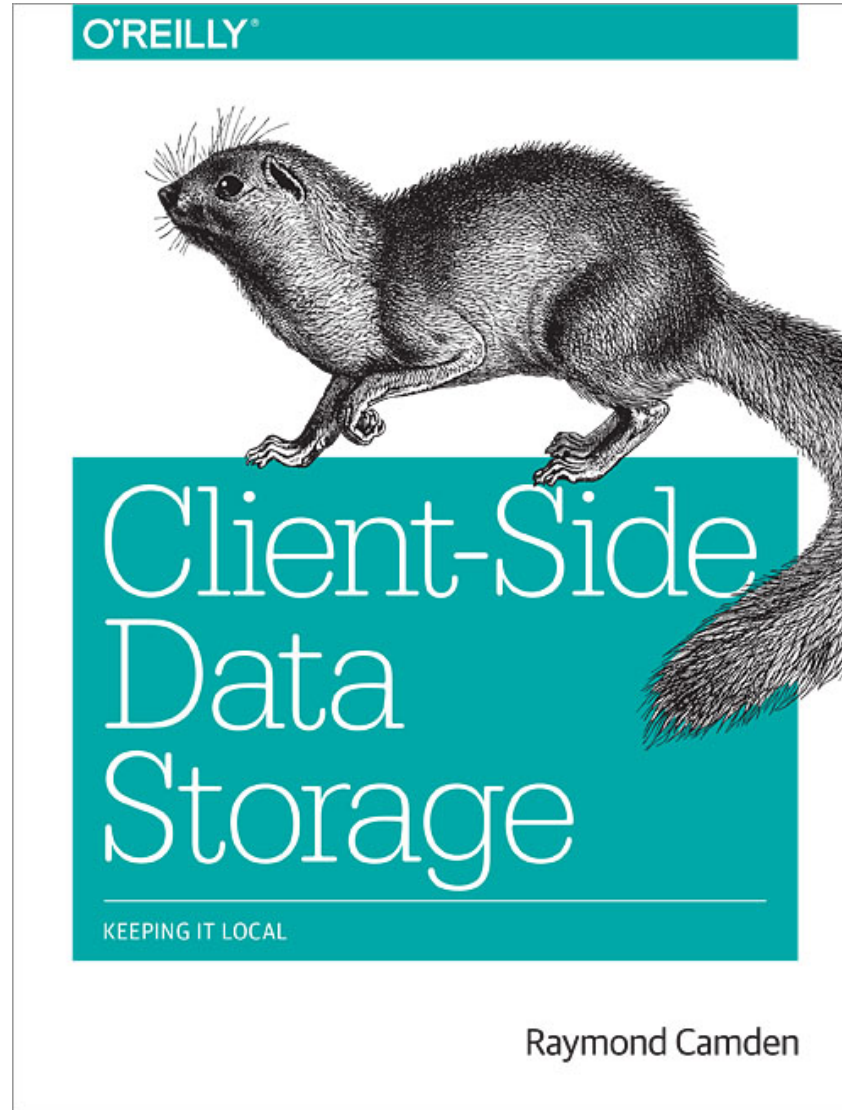
Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			47					4.3	
8			48					4.4	
9		44	49	<sup>2</sup> 9		<sup>2</sup> 8.4		4.4.4	
<sup>1</sup> 11	<sup>1</sup> 13	45	50	<sup>2</sup> 9.1	36	<sup>2</sup> 9.2	8	47	49
	<sup>1</sup> 14	46	51	<sup>2</sup> TP	37	<sup>2</sup> 9.3			
		47	52		38				
		48	53						

Notes Known issues (3) Resources (9) Feedback

<sup>1</sup> Partial support in IE 10 & 11 refers to a number of subfeatures **not being supported**.

<sup>2</sup> Partial support in iOS 8 & 9 refers to **seriously buggy behavior** as well as complete lack of support in WebViews.



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

- ▶ 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**

"AppCache broke a lot of eggs, and failed to make an Omlet"  
Jake Archibald (Coauthor der AppCache Spec)



**Servicewoker**

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

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

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

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

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

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

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



```
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;
        });
      });
    })
  );
});
```

```
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;
    })
  );
});
```

## ServiceWorker can do even more ...

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

## Service Workers - WD

Global

0% + 53.5% = 53.5%

Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline.

Current aligned

Usage relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			47					4.3	
8			48					4.4	
9		44	49	9		8.4		4.4.4	
11	13	45	50	9.1	36	9.2	8	47	49
	14	46	51	TP	37	9.3			
		47	52		38				
		48	53						

Notes

Known issues (0)

Resources (8)

Feedback

Details on partial support can be found on [is ServiceWorker Ready?](#)

MS Edge status: [Under Consideration](#)

WebKit status: [Under Consideration](#)

<sup>1</sup> Partial support can be enabled in Firefox with the `dom.serviceWorkers.enabled` flag.

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**