深入探讨PWA之Service Worker

舒晨 shuchery@gmail.com

几点思考

- Service Worker 是什么
- 它有哪些能力?
- 哪些场景可以用到?
- 它跟现有的缓存方案有什么区别?
- 它会带来什么问题?

PWA 有哪些特点?

- 添加到主屏幕，全屏体验 (App Manifest)
- 离线存储 (Service Worker)
- 消息推送 (Web Push & Notification)
- 后台数据同步 (Service Worker & Background Sync)

Web App Manifest

- 添加到主屏幕，自定义图标。
- 启动桌面图标，自定义启动图。
- 启动后，隐藏地址栏，全屏展示。

manifest.json

根据配置项安装到桌面

引入manifest.json

```
<head>
  <title>Minimal PWA</title>
  <meta name="viewport" content="width=device-width, user-scalable=no" />
  <link rel="manifest" href="manifest.json" />
  <link rel="stylesheet" type="text/css" href="main.css">
  <link rel="icon" href="/e.png" type="image/png" />
</head>
```

index.html

Chrome PC/Mobile

1. chrome://flags
2. Desktop PWAs 开启
3. 重启浏览器，设置，安装.

Service Worker

资源缓存，请求拦截，快速响应请求

Service Worker 是一种 Web Workers

Web Workers 包括 (Worker, Shared Worker, Service Worker)

单线程 计算密集 阻塞UI

Web Workers

```
// Create worker
const myWorker = new Worker('worker.js');

// Send message to worker
myWorker.postMessage('Hello!');

// Receive message from worker
myWorker.onmessage = function(e) {
  console.log(e.data);
}
```

main.js

Web Workers

```
// Receive message from main file
self.onmessage = function(e) {
  console.log(e.data);

  // Send message to main file
  self.postMessage(workerResult);
}
```

worker.js

WorkerGlobalScope:

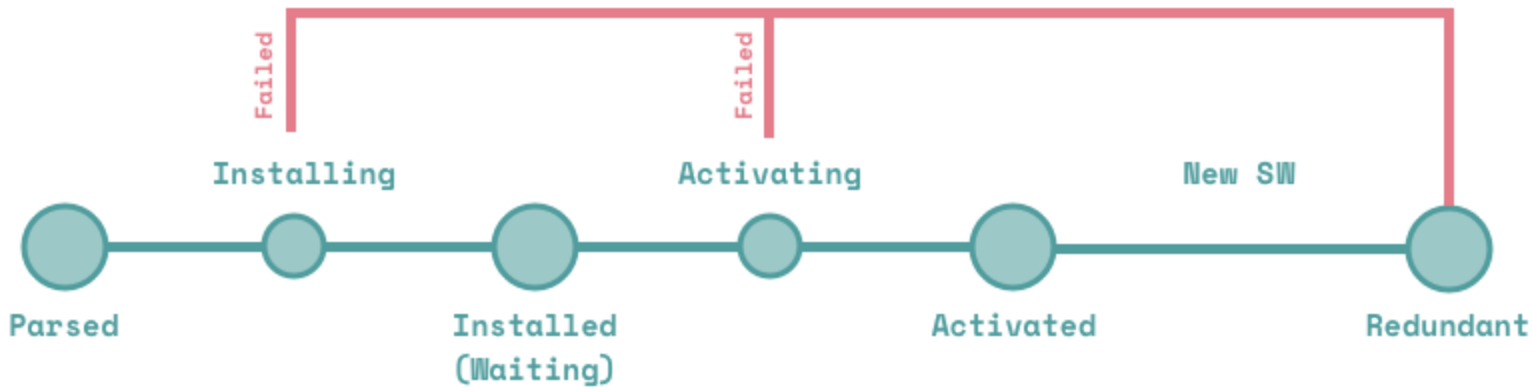
1. DedicatedWorkerGlobalScope
2. SharedWorkerGlobalScope
3. ServiceWorkerGlobalScope


Browser
Page
Service Worker
Network
Cache

Service Worker 特点

1. 线程独立，不影响当前网页渲染。
2. 无法操作DOM，通过 postMessage 跟主线程通信。
3. 离线缓存静态资源。
4. 基于异步 Promise 实现
5. 代理和拦截请求并自定义响应
6. 必须在 HTTPS / localhost 环境下工作

lifecycle


Failed
Failed
Installing
Activating
New SW
Parsed
Installed
(Waiting)
Activated
Redundant

注册

```
navigator.serviceWorker.register('./sw.js', { scope: './' });
```

service worker scope

navigator.serviceWorker.register('./a/b/sw.js'});

- /a/b/
- /a/b/c
- /a/b/c/d

- /a/
- /a/f
- /a/e

安装

```
const cacheFiles = ['./index.html', 'style.css', 'main.js'];

self.addEventListener('install', function(event) {
  // Perform install steps
  // add file to cache storage
  cache.addAll(cacheFiles)
});
```

sw.js

激活

```
self.addEventListener('activate', function(event) {
  event.waitUntil(
    // Get all the cache names
    caches.keys().then(function(cacheNames) {
      return Promise.all(
        // Get all the items that are stored under a
// different cache name than the current one
cacheNames.filter(function(cacheName) {
  return cacheName != currentCacheName;
}).map(function(cacheName) {
  // Delete the items
  return caches.delete(cacheName);
})
```

sw.js

Lifecycle Example

self.skipWaiting()

跳过 Waiting 从 Installed 直接到 Actived

强制激活

event.waitUntil()

它能确保当前事件在promise结束之后完成

fetch push sync

拦截请求

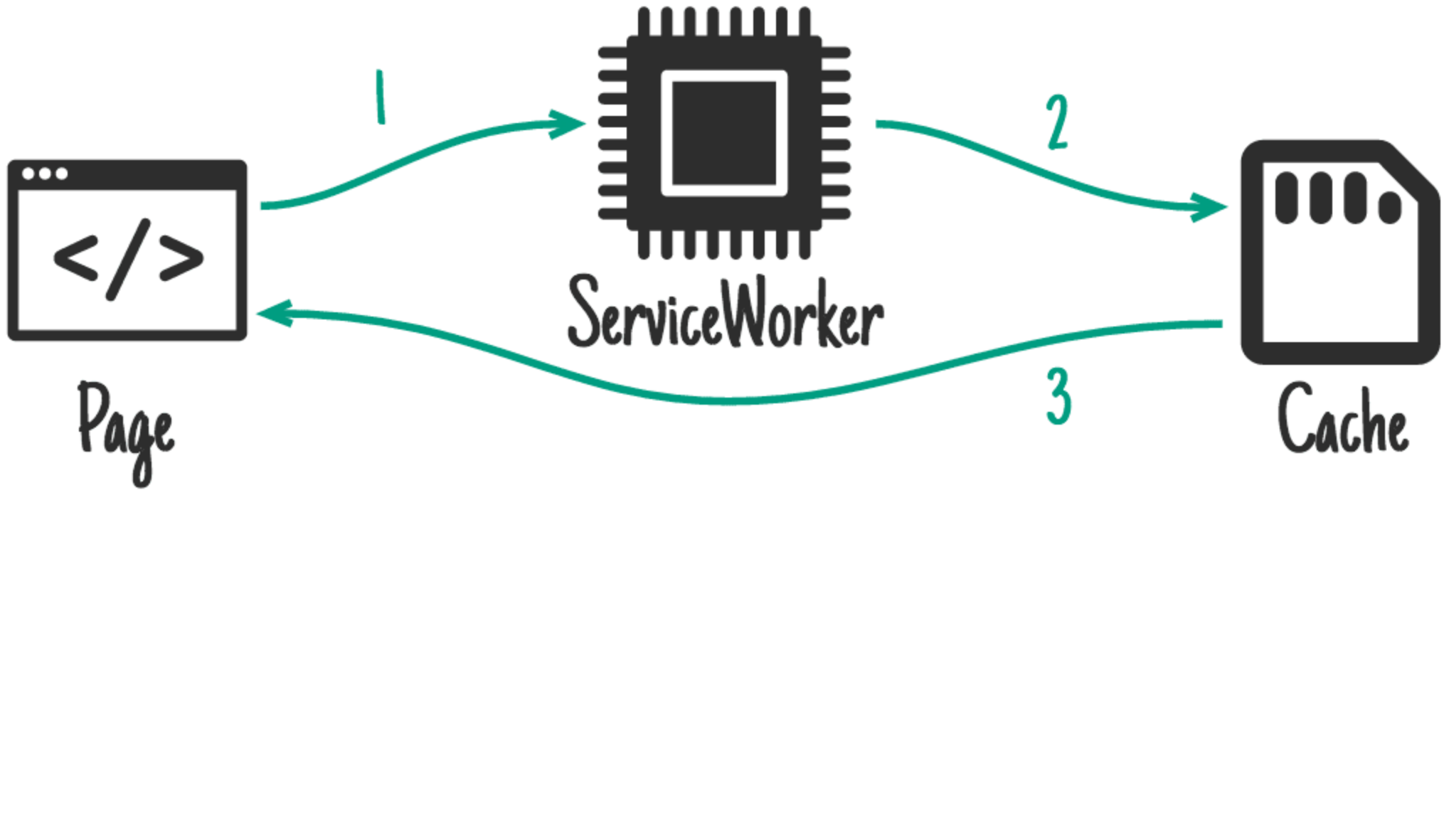
```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        // Cache hit - return response
        if (response) {
          return response;
        }
        return fetch(event.request);
      }
    )
  );
```

sw.js

缓存策略

1. 仅使用缓存(cache only)
2. 仅使用网络(network only)
3. 缓存优先(cache first)
4. 网络优先(network first)
5. 缓存验证(stale-while-revalidate)
6. 速度优先(speed first)

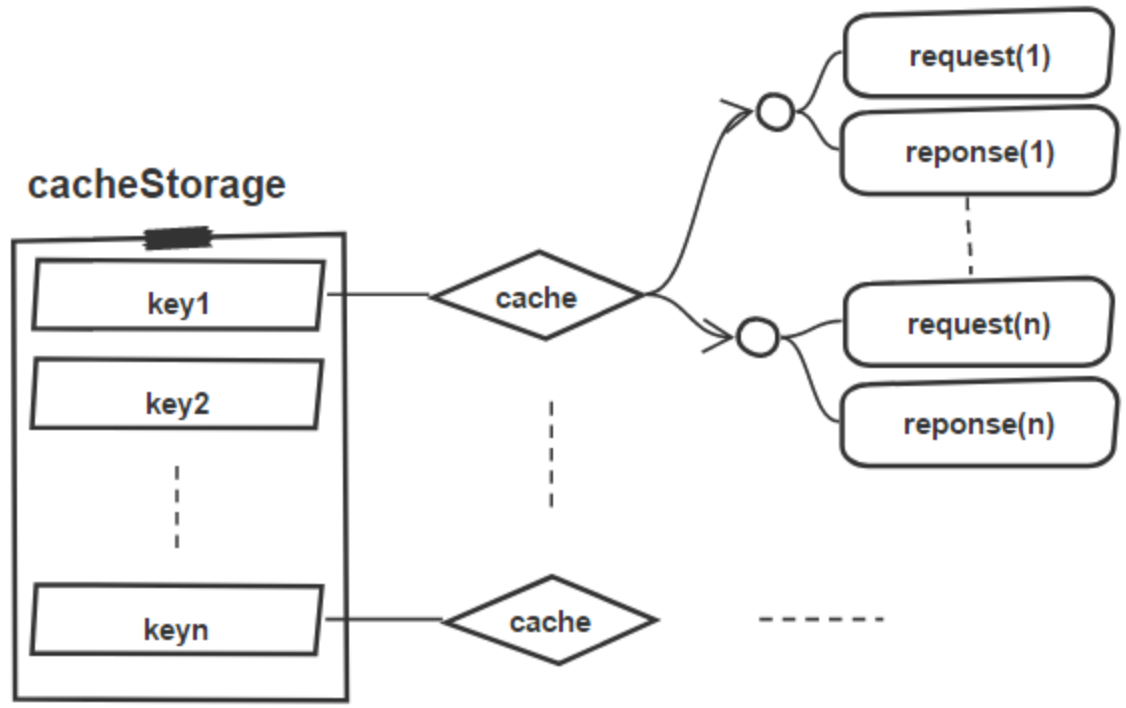
cache only



cache only

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

适用于长期不会改变的资源，如 font, icon



Console Elements Sources Network Audits Performance Application Memory Security Redux Jsbridge

Application
- Manifest
- Service Workers
- Clear storage

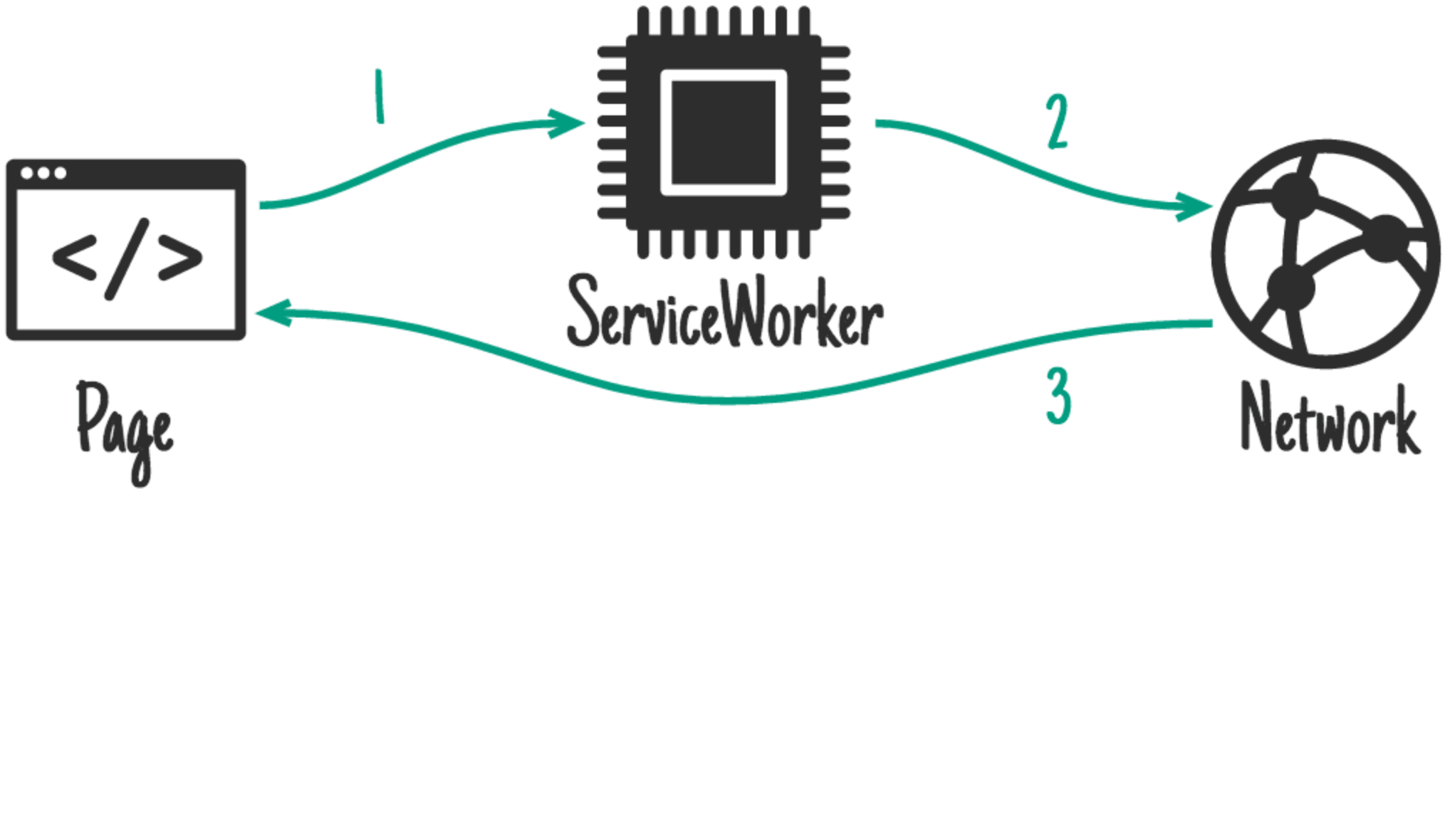
Storage
- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache
- Cache Storage
  - my-site-cache-v3 - http://localhost:3000
- Application Cache

Frames
- top

Path	Response-Type	Content-Type	Content-Length	Time Cached
/	basic	text/html; charset=UTF-8	1,550	2019/3/8 上午10:24:35
/index.html	basic	text/html; charset=UTF-8	1,550	2019/3/8 上午10:01:19
/static/cat.jpg	basic	image/jpeg	58,996	2019/3/8 上午10:24:36
/style.css	basic	text/css; charset=UTF-8	72	2019/3/8 上午10:01:19

network only

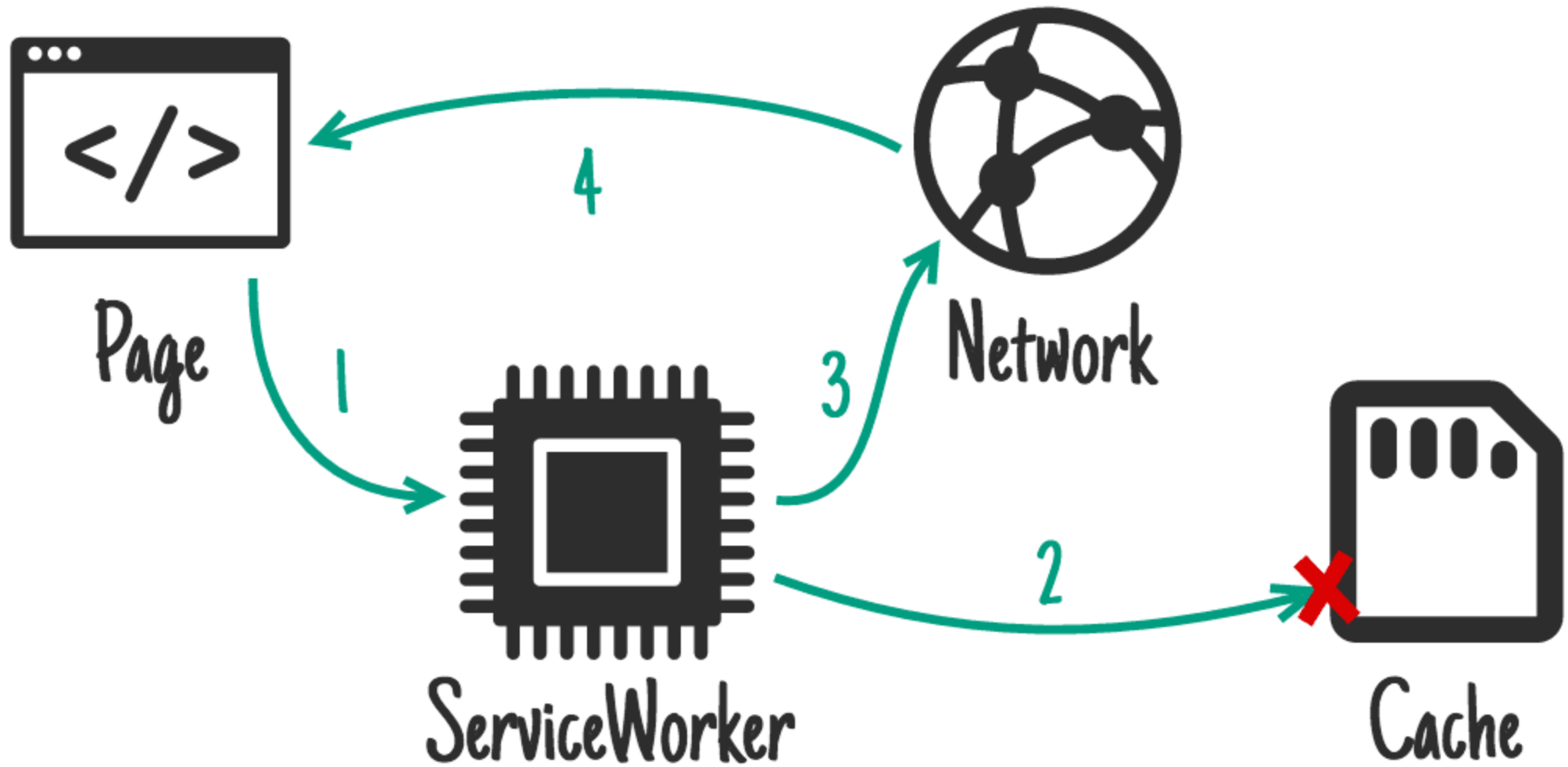


network only

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

适用于不需要离线访问的场景，像 POST request, analytics pings

cache first

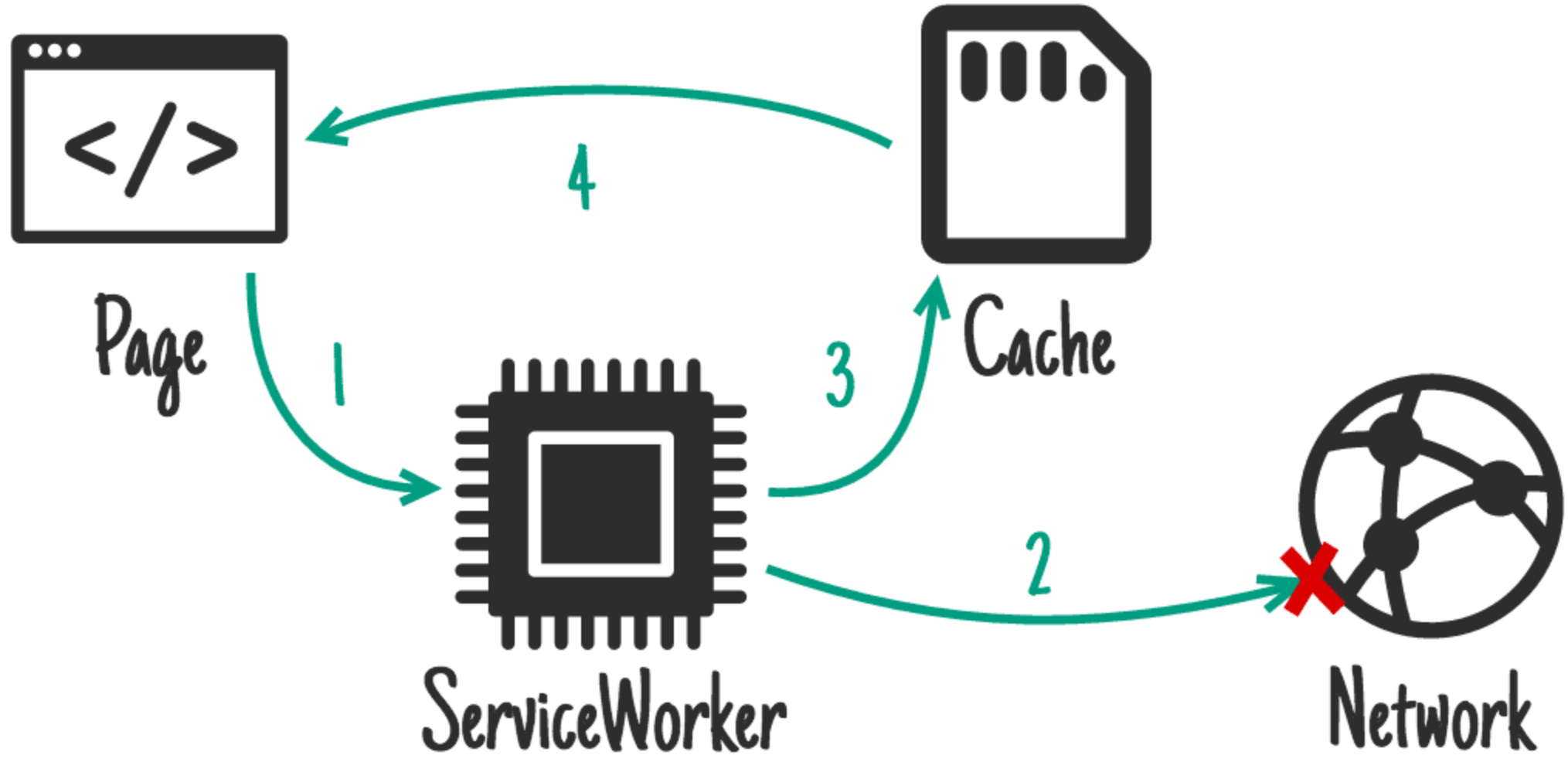


cache first

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        // Cache hit - return response
        if (response) {
          return response;
        }

        // IMPORTANT:Clone the request. A request is a stream and
        // can only be consumed once. Since we are consuming this
        // once by cache and once by the browser for fetch, we need
        // to clone the response.
        var fetchRequest = event.request.clone();

        return fetch(fetchRequest).then(
          function(response) {
            // Check if we received a valid response
            if(!response || response.status !== 200 || response.type !== 'basic') {
              return response;
            }

```

对大多数静态资源都适用的场景，优化幅度最大

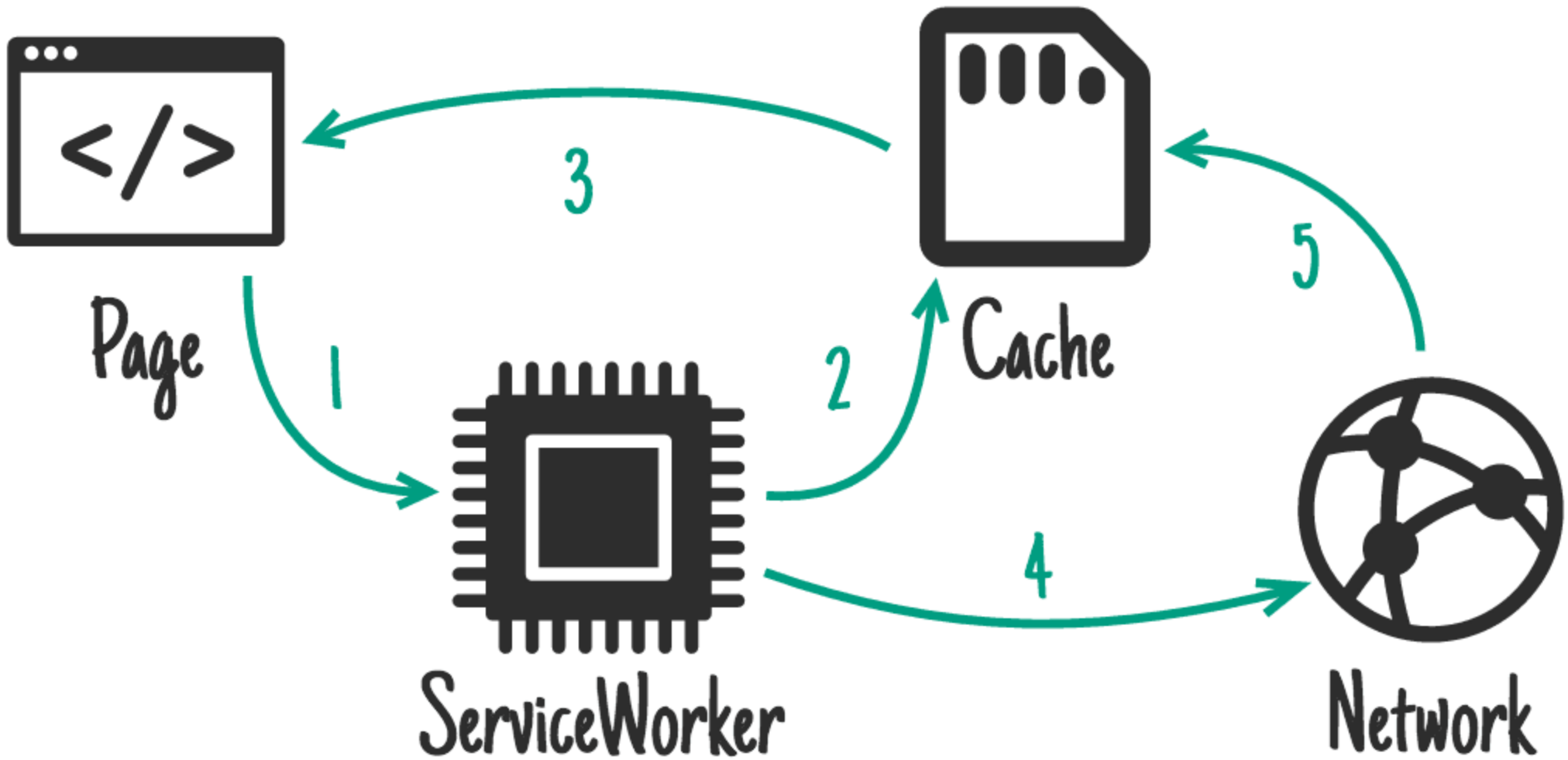
network first



network first

对资源实时性要求高的场景，弥补弱网不足

缓存验证



stale-while-revalidate

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

适用于频繁更新的资源，但对实时性要求不高的场景。比如头像等用户信息

serivce worker 更新

- sw.js 本身走http缓存，服务端不缓存。
- Cache-Control: no-cache；
- /sw.js?v=buildVersion
- 浏览器更新机制，每24h会更新一次。
- 更新后的第一次访问还是老的内容，需要第二次进入才能看到更新的内容

缓存实时生效

1. sw激活并清除老的缓存之后，通过postMessage告诉主线程
2. 主线程监听消息，获取最新内容

消息通讯

更新缓存，给主线程发送消息

消息通讯

```
navigator.serviceWorker.addEventListener('message', function (e) {
  if (e.data === 'sw.update') {
    // show toast and reload
  }
});
```

主线程接收消息，提示更新

How to Remove a Service Worker?

卸载

```
navigator.serviceWorker.getRegistrations().then(
  function(registrations) {
    for(let registration of registrations) {
      registration.unregister();
    }
});
```

main.js

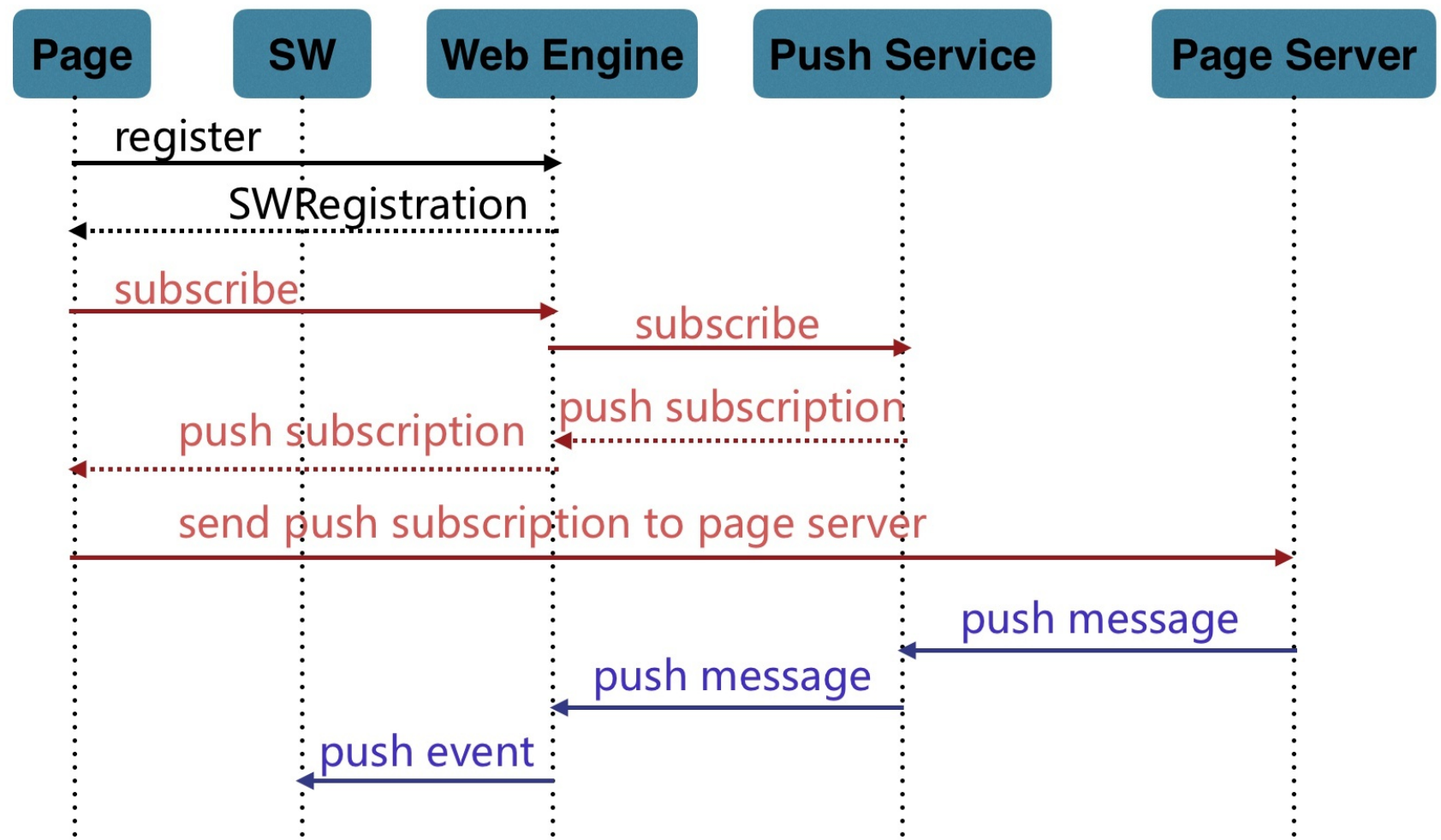
本地存储

1. cookie
2. localStorage/sessionStorage
3. Cache Storage
4. IndexDB

Web Push / Notification

实现消息推送，提高网站转化率

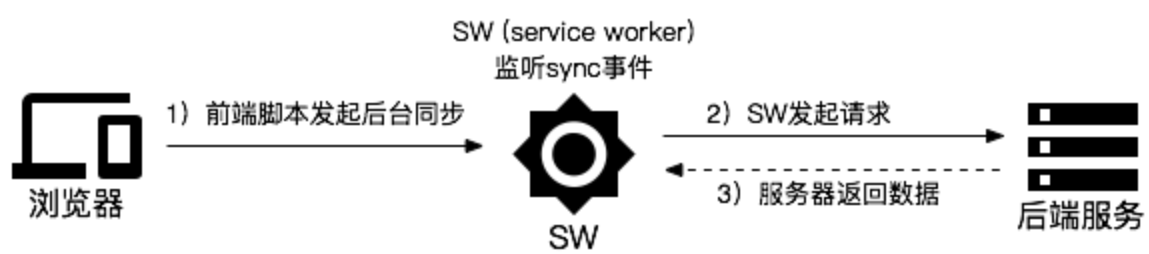
推送在Chrome内不可用?


Page
SW
Web Engine
Push Service
Page Server
register
SWRegistration
subscribe
subscribe
push subscription
push subscription
send push subscription to page server
push message
push message
push event

Push Demo

Background Sync

web邮件客户端、即时通讯工具


SW (service worker)
监听sync事件
1）前端脚本发起后台同步
2）SW发起请求
3）服务器返回数据
浏览器
SW
后端服务

后台同步

```
// Register your service worker:
navigator.serviceWorker.register('/sw.js');

// Then later, request a one-off sync:
navigator.serviceWorker.ready.then(function(swRegistration) {
  return swRegistration.sync.register('myFirstSync');
});
```

main.js

后台同步

```
self.addEventListener('sync', function(event) {
  if (event.tag == 'myFirstSync') {
    event.waitUntil(doSomeStuff());
  }
});
```

sw.js

Sync Demo

sw cache 和 memory cache 区别

Name	Status	Protocol	Type	Initiator	Size	Time	Waterfall
data:image/png;base...	200	data	png	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
20180921-fdd-px.mp4	206	h2	media	Other	32.6 MB	8.06 s	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
data:image/svg+xml;...	200	data	svg+xml	Other	(from memory cache)	0 ms	
20180921-fdd-px.mp4	206	h2	media	Other	2.5 MB	592 ms	
errors	200	h2	xhr	VM132 ...	217 B	195 ms	
errors	202	h2	xhr	Other	102 B	173 ms	
hm.gif?hca=1EA862D49BCF...	200	http/1.1	gif	hm.js?3...	299 B	95 ms	

88 requests | 46.5 MB transferred | Finish: 28.49 s | DOMContentLoaded: 1.74 s | Load: 3.32 s

请求分类

1. Service Worker
2. Memory Cache（浏览器preload/preloader/link/src）
3. Disk Cache（http headers/强制缓存和协商缓存）
4. 网络请求

工程实践

- CDN资源(Access-Control-Allow-Origin)
- 多页面的项目(多个Service Worker 业务相似度)
- 降级方案(动态开关)

App Manifest Support

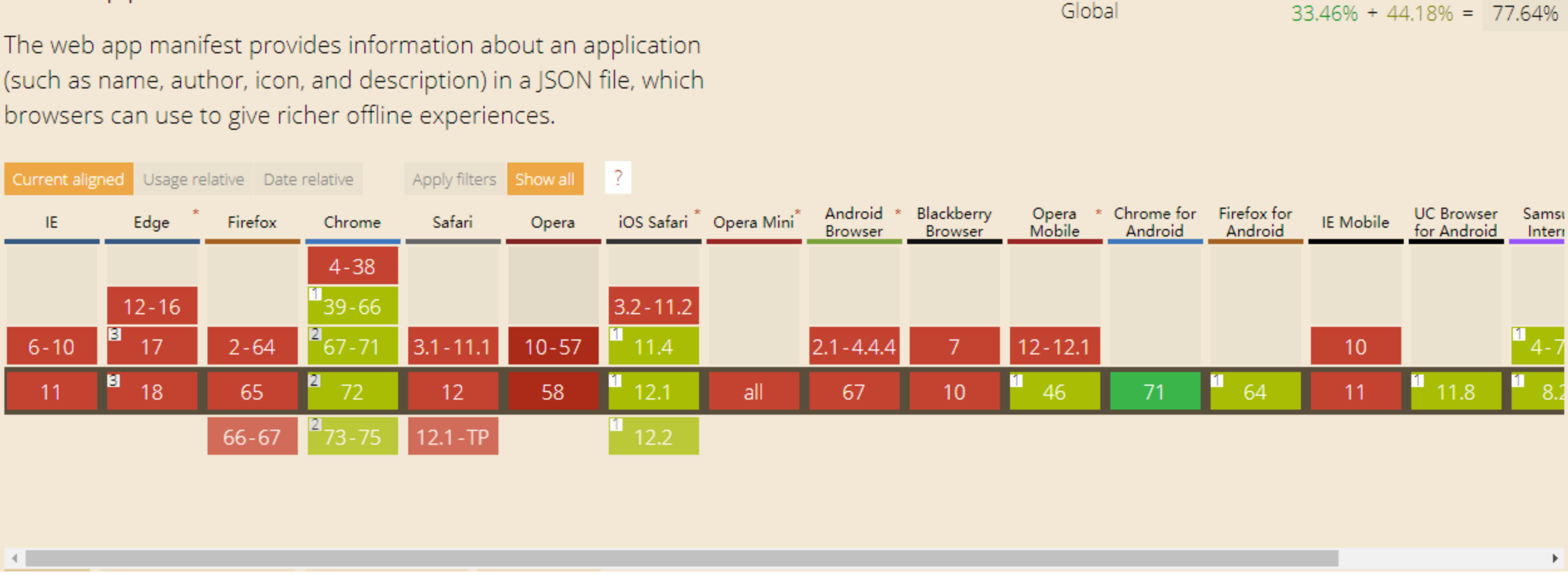
Web App Manifest - WD

The web app manifest provides information about an application (such as name, author, icon, and description) in a JSON file, which browsers can use to give richer offline experiences.

Usage
Global

% of all users ?
33.46% + 44.18% = 77.64%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?



Service Worker Support

Service Workers - WD

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

Usage
Global

% of all users
85.89% + 0.16% = 86.05%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			70						
	17	64	71		11.4				4
11	18	65	72	12	12.1	all	71	11.8	8.2
		66	73	12.1	12.2				
		67	74	TP					
			75						

Notes | Known issues (0) | Resources (8) | Feedback

Details on partial support can be found on is ServiceWorker Ready?

Push API Support

Push API - WD

API to allow messages to be pushed from a server to a browser, even when the site isn't focused or even open in the browser.

Usage: Global

% of all users: 74.13% + 0.58% = 74.71%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsu Inter
		2-43													
		2 44													
		2 4 45													
		2 46-51													
		2 4 52													
		2 53-59	4-43		10-36										
	12-16	2 4 60	1 2 44-49	3.1-9	37-41										
6-10	17	2 61-64	2 50-71	3 9.1-11.1	2 42-57	3.2-11.4		2.1-4.4.4	7	12-12.1			10		4-7
11	18	2 65	2 72	3 12	2 58	12.1	all	67	10	46	71	64	11	11.8	8.2
		2 66-67	2 73-75	3 12.1-TP		12.2									

Background Sync Support

Background Sync API - UNOFF

Usage % of all users ?

Global 68.33%

Provides one-off and periodic synchronization for Service Workers with an onsync event.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsu Inter
			4-48		10-35										4
6-10	12-17	2-64	49-71	3.1-11.1	36-57	3.2-11.4		2.1-4.4.4	7	12-12.1			10		5-7
11	18	65	72	12	58	12.1	all	67	10	46	71	64	11	11.8	8.2
		66-67	73-75	12.1-TP		12.2									

参考

Service Worker： 简介
Service Worker 生命周期
离线指南
Workbox 3： Service Worker 可以如此简单
如何优雅的为 PWA 注册 Service Worker
使用Service Worker进行后台同步 - Background Sync
一文读懂前端缓存

Thanks!