



HI 00092 00042



# No Internet

Try:

- Checking the network cables, modem and router
- Reconnecting to Wi-Fi

ERR\_INTERNET\_DISCONNECTED

# Downasaur

When using Google Chrome and your internet connectivity dies, you'll see a little Tyrannosaurus.

This is called the 'downasaur'.



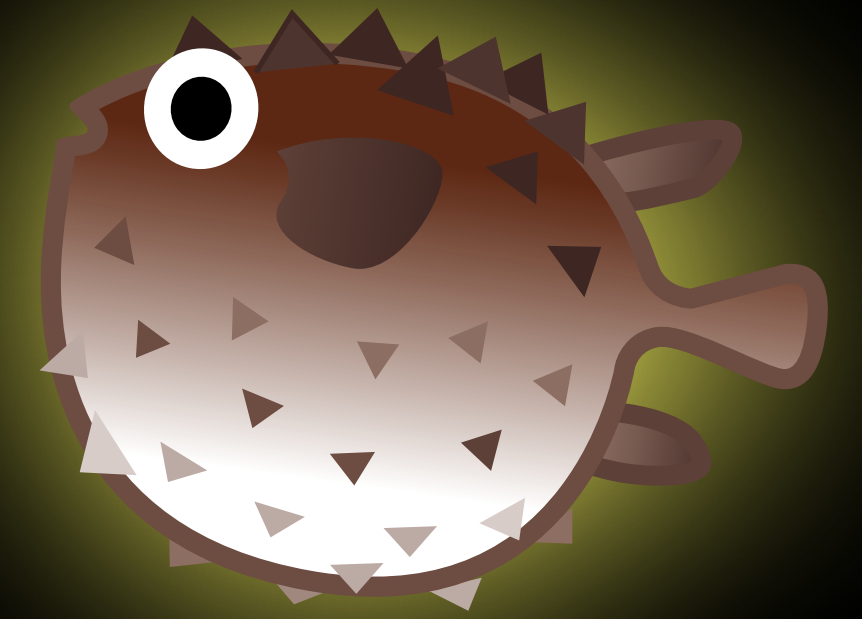


**OFFLINE FIRST!**





# PWA



- Build my first website in 1996
- Front-End Focused Senior DevOps Engineer
- Google Developer Expert in Web Technologies





# STARTER CODE

<https://github.com/rowdyrabouw/offline-first.nl/tree/01-starter-code>





```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.ico" />
  </head>
  <body>
    <main>
      <h1>Jurassic Park</h1>
      
      <h2>Welcome to... Jurassic Park!</h2>
      <p>...</p>
      <ol>
        <li><a href="/jurassic-park-1993.html">Jurassic Park</a></li>
        ...
      </ol>
    </main>
  </body>
</html>
```





# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does make it easy to lose track of exactly what happens in one or two along the way.





# Service Workers

- JavaScript file
- runs in the background in a separate thread; non-blocking
- scoped to a website
- can't modify page; no access to DOM
- asynchronous; promises or async/await
- event driven



# # Service Workers - CR

Usage % of all users ?

Global 96.15% + 0.02% = 96.17%

☆ 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 Date relative Filtered All ⚙

Chrome	Edge*	Safari	Firefox	Opera	Chrome for Android	Safari on iOS*	Samsung Internet	Opera Mini*	Opera Mobile*	UC Browser for Android	Android Browser*	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
			2-32												
			1 33-43												
			44												
			3 45												
			46-51												
			3 52												
			53-59												
			3 60												
4-39	12-14		61-67	10-26											
40-44	2 15-16	3.1-11	3 68	27-31		3.2-11.2									
45-114	17-114	11.1-16.4	69-114	32-99		11.3-16.4	4-20		12-12.1		2.1-4.4.4				2.5
115	115	16.5	115	100	114	16.5	21	all	73	15.5	114	115	13.1	13.18	3.1
116-118		16.6-TP	116-117			16.6-17									



**BROWSER**

**SERVER**

**MORE JAVASCRIPT**

**SERVICE WORKER**

```
self.addEventListener("install", (event) => {  
  console.log("%c[sw.js] Service Worker installed", "color: #FEC233");  
});  
  
self.addEventListener("activate", (event) => {  
  console.log("%c[sw.js] Service Worker activated", "color: #FEC233");  
});
```



```
if ("serviceWorker" in navigator) {  
  window.addEventListener("load", () => {  
    navigator.serviceWorker  
      .register("sw.js")  
      .then((reg) => {  
        console.log("%c[app.js] Service Worker registered", "color: #00ABD2");  
      })  
      .catch((e) => {  
        console.error(  
          "%c[app.js] Error during service worker registration:",  
            e  
        );  
      });  
  });  
}
```



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.png" />
    <script src="assets/js/app.js"></script>
  </head>
  <body>
    <main>
      <h1>Jurassic Park</h1>
      
      <h2>Welcome to... Jurassic Park!</h2>
      <p>...</p>
      <ol>
        <li><a href="/jurassic-park-1993.html">Jurassic Park</a></li>
        ...
      </ol>
    </main>
  </body>
</html>
```







WELCOME TO... JURASSIC PARK! Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application, Service Workers, Storage, Cache, Background Services

Console, What's New, Issues, Filter, Default levels, No Issues



# STATIC CACHE

<https://github.com/rowdyrabouw/offline-first.nl/tree/03-add-static-cache>



```
const STATIC_CACHE = "static-v1";
```

```
const STATIC_CACHE = "static-v1";

const STATIC_FILES = [
  "/",
  "index.html",
  "/favicon.png",
  "assets/css/style.css",
  "assets/fonts/neuland_inline.woff2",
  "assets/img/jurassic-park.jpg",
  "assets/js/app.js",
];
```



```
const STATIC_CACHE = "static-v1";

const STATIC_FILES = [
  "/",
  "index.html",
  "/favicon.png",
  "assets/css/style.css",
  "assets/fonts/neuland_inline.woff2",
  "assets/img/jurassic-park.jpg",
  "assets/js/app.js",
];

self.addEventListener("install", (event) => {
  console.log("%c[sw.js] Service Worker installed", "color: #FEC233");

});
```



```
const STATIC_CACHE = "static-v1";

const STATIC_FILES = [
  "/",
  "index.html",
  "/favicon.png",
  "assets/css/style.css",
  "assets/fonts/neuland_inline.woff2",
  "assets/img/jurassic-park.jpg",
  "assets/js/app.js",
];

self.addEventListener("install", (event) => {
  console.log("%c[sw.js] Service Worker installed", "color: #FEC233");
  event.waitUntil(
    caches.open(STATIC_CACHE).then((cache) => {
      cache.addAll(STATIC_FILES).then(() => {
        console.log("%c[sw.js] Files added to static cache", "color: #FEC233");
        self.skipWaiting();
      });
    });
  );
});
```



```
if ("serviceWorker" in navigator) {  
  window.addEventListener("load", () => {  
    navigator.serviceWorker  
      .register("sw.js")  
      .then((reg) => {  
        console.log("%c[app.js] Service Worker registered", "color: #00ABD2");  
  
      })  
      .catch((e) => {  
        console.error("%c[app.js] Error during service worker registration:", e);  
      });  
  });  
}
```



```
if ("serviceWorker" in navigator) {
  window.addEventListener("load", () => {
    navigator.serviceWorker
      .register("sw.js")
      .then((reg) => {
        console.log("%c[script.js] Service Worker registered", "color: #00ABD2");
        reg.onupdatefound = () => {
          const installingWorker = reg.installing;
          installingWorker.onstatechange = () => {
            if (installingWorker.state === "installed") {
              console.log("%c[app.js] New or updated content is available",
                "color: #00ABD2");
              if (confirm("A new version is available.\n
                Would you like to load the new version?")) {
                window.location.reload();
              }
            }
          }
        }
      })
      .catch((e) => {
        console.error("%c[app.js] Error during service worker registration:", e);
      });
  });
}
});
```





# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application | Network | Console | Application | Performance insights | Sources | Elements | Recorder

Application

- Manifest
- Service Workers**
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
- Back/forward cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler

Service Workers

Offline  Update on reload  Bypass for network

**http://localhost:1993/** [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 03/07/2022, 20:58:27

Status ● #0 activated and is running [stop](#)

Clients [http://localhost:1993/](#) [focus](#)

Push

Sync

Periodic Sync

Update Cycle

Version	Update Activity	Timeline
▶ #0	Install	
▶ #0	Wait	
▶ #0	Activate	▬

Console | What's New | Issues

Filter | Default levels | No Issues

```
[app.js] Service Worker registered app.js:6
[sw.js] Service Worker installed sw.js:2
[sw.js] Service Worker activated sw.js:6
```





# FETCH CACHE

<https://github.com/rowdyrabouw/offline-first.nl/tree/04-fetch-static-cache>



# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

10 ms	20 ms	30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	100 ms	110 ms
-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------



Recording network activity...  
Perform a request or hit ⌘ R to record the reload.  
[Learn more](#)

Empty console log area with a blue arrow pointing to the left.



```
self.addEventListener("fetch", (event) => {  
  event.respondWith(  
    // look at all caches for a match on the key (= request)  
    caches.match(event.request).then((response) => {  
      if (response) {  
        // return from cache  
        return response;  
      } else {  
        // fetch from the server  
        return fetch(event.request);  
      }  
    })  
  );  
});
```



# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Network tab active. Filter: All. 5 requests, 148 kB transferred, 147 kB resources, Finish: 28 ms, DOMContentLoaded: 24 ms, Load: 35 ms.

Name	Status	Type	Initiator	Size	Time	Waterfall
localhost	200	document	Other	2.0 kB	8 ms	[Waterfall bar]
style.css	200	stylesheet	(index)	1.7 kB	5 ms	[Waterfall bar]
app.js	200	script	(index)	1.2 kB	7 ms	[Waterfall bar]
jurassic-park.jpg	200	jpeg	(index)	114 kB	7 ms	[Waterfall bar]
neuland_inline.woff2	200	font	style.css	29.6 kB	3 ms	[Waterfall bar]

Console tab active. Filter: top. No Issues.

```
[script.js] Service Worker registered
```



Incognito

Application Performance insights Sources Elements Recorder

log  Disable cache No throttling

Invert  Hide data URLs **All** Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Blocked Requests  3rd-party requests

Status	Type	Initiator	Size	Time	Waterfall
200	document	Other	2.0 kB	8 ms	
200	stylesheet	(index)	1.7 kB	5 ms	
200	script	(index)	1.2 kB	7 ms	
200	jpeg	(index)	114 kB	7 ms	
200	font	style.css	29.6 kB	3 ms	

147 kB resources | Finish: 28 ms | DOMContentLoaded: 24 ms | **Load: 35 ms**

Issues

Default levels No Issues

registered app.js:6

Incognito

Console Application Performance insights Sources Elements Recorder

Preserve log  Disable cache No throttling

Invert  Hide data URLs **All** Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Blocked Requests  3rd-party requests

Status	Type	Initiator	Size	Time	Waterfall
200	document	(ServiceWorker)		4 ms	
200	stylesheet	(index)	(ServiceWorker)	1 ms	
200	script	(index)	(ServiceWorker)	1 ms	
200	jpeg	(index)	(ServiceWorker)	1 ms	
200	font	style.css	(ServiceWorker)	1 ms	

147 kB resources | Finish: 13 ms | DOMContentLoaded: 14 ms | **Load: 18 ms**

Issues

Filter

Default levels No Issues

Worker registered app.js:6





# OFFLINE FALLBACK

<https://github.com/rowdyrabouw/offline-first.nl/tree/05-offline-fallback>





# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

10 ms	20 ms	30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	100 ms	110
-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	-----

Recording network activity...  
Perform a request or hit % R to record the reload.

>



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park - offline</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.png" />
    <script src="assets/js/app.js"></script>
  </head>
  <body>
    <main>
      <h1>Jurassic Park is closed</h1>
      
      <h2>It seems you are currently offline ...</h2>
      <p><strong>We're sorry, this page hasn't been cached yet :-(</strong></p>
      <p>But why don't you try one of our <a href="/">other pages</a>?</p>
    </main>
  </body>
</html>
```



```
const STATIC_CACHE = "static-v1";  
const STATIC_FILES = [  
  "/",  
  "index.html",  
  "/favicon.ico",  
  "assets/css/style.css",  
  "assets/fonts/neuland_inline.woff2",  
  "assets/img/jurassic-park.jpg",  
  "assets/js/app.js",  
  
];
```



```
const STATIC_CACHE = "static-v2";  
const STATIC_FILES = [  
  "/",  
  "index.html",  
  "/favicon.ico",  
  "assets/css/style.css",  
  "assets/fonts/neuland_inline.woff2",  
  "assets/img/jurassic-park.jpg",  
  "assets/js/app.js",  
  "offline.html",  
];
```



```
self.addEventListener("fetch", (event) => {  
  event.respondWith(  
    // look at all caches for a match on the key (= request)  
    caches.match(event.request).then((response) => {  
      if (response) {  
        // return from cache  
        return response;  
      } else {  
        // fetch from the server  
        return fetch(event.request)  
      }  
    })  
  });  
});
```



```
self.addEventListener("fetch", (event) => {
  event.respondWith(
    // look at all caches for a match on the key (= request)
    caches.match(event.request).then((response) => {
      if (response) {
        // return from cache
        return response;
      } else {
        // fetch from the server
        return fetch(event.request).catch(() => {
          return caches.open(STATIC_CACHE).then((cache) => {
            return cache.match("/offline.html");
          });
        });
      }
    });
  });
});
```



```
self.addEventListener("fetch", (event) => {
  event.respondWith(
    // look at all caches for a match on the key (= request)
    caches.match(event.request).then((response) => {
      if (response) {
        // return from cache
        return response;
      } else {
        // fetch from the server
        return fetch(event.request).catch(() => {
          return caches.open(STATIC_CACHE).then((cache) => {
            if (event.request.headers.get("accept").includes("text/html")) {
              return cache.match("/offline.html");
            }
          });
        });
      }
    });
  });
});
```



# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application | Network | Console | Application | Performance insights | Sources | Elements | Recorder

### Application

- Manifest
- Service Workers
- Storage

### Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

### Cache

- Cache Storage
  - static-v1 - http://localhost:1993
- Back/forward cache

### Background Services

- Background Fetch
- Background Sync
- Notifications

### Service Workers

Offline  Update on reload  Bypass for network

http://localhost:1993/ [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)  
Received 04/07/2022, 16:09:15

Status ● #0 activated and is running [stop](#)

Push:  [Push](#)

Sync:  [Sync](#)

Periodic Sync:  [Periodic Sync](#)

### Update Cycle

Version	Update Activity	Timeline
#0	Install	<div style="width: 20px; height: 10px; background-color: #008080;"></div>
#0	Wait	<div style="width: 20px; height: 10px; background-color: #800080;"></div>
#0	Activate	<div style="width: 40px; height: 10px; background-color: #FFA500;"></div>

Console | What's New | Issues

Filter: top | Default levels | No Issues

```
[script.js] Service Worker registered app.js:6
```





# DYNAMIC CACHE

<https://github.com/rowdyrabouw/offline-first.nl/tree/06-dynamic-cache>



```
const DYNAMIC_CACHE = "dynamic-v1";
```

```
self.addEventListener("fetch", (event) => {
  event.respondWith(
    // look at all caches for a match on the key (= request)
    caches.match(event.request).then((response) => {
      if (response) {
        // return from cache
        return response;
      } else {
        // fetch it from the server and save to cache
        return fetch(event.request)

        .catch(() => {
          return caches.open(STATIC_CACHE).then((cache) => {
            if (event.request.headers.get("accept").includes("text/html")) {
              return cache.match("/offline.html");
            }
          });
        });
      }
    });
  });
});
```

```
self.addEventListener("fetch", (event) => {
  event.respondWith(
    // look at all caches for a match on the key (= request)
    caches.match(event.request).then((response) => {
      if (response) {
        // return from cache
        return response;
      } else {
        // fetch it from the server and save to cache
        return fetch(event.request)
          .then((res) => {
            return caches.open(DYNAMIC_CACHE).then((cache) => {
              cache.put(event.request.url, res.clone());
              return res;
            });
          })
          .catch(() => {
            return caches.open(STATIC_CACHE).then((cache) => {
              if (event.request.headers.get("accept").includes("text/html")) {
                return cache.match("/offline.html");
              }
            });
          });
      }
    });
});
```

# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application | Network | Console | Application | Performance insights | Sources | Elements | Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
  - static-v1 - http://localhost:1993
  - static-v2 - http://localhost:1993
- Back/forward cache

Background Services

- Background Fetch
- Background Sync

Service Workers

Offline  Update on reload  Bypass for network

http://localhost:1993/ [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 04/07/2022, 16:17:05

Status ● #1 activated and is running [stop](#)

Push

Sync

Periodic Sync

Update Cycle

Version	Update Activity	Timeline
▶ #1	Install	
▶ #1	Wait	
▶ #1	Activate	<div style="width: 100px; height: 10px; background-color: orange;"></div>



# THE LOST WORLD: JURASSIC PARK



## PLOT SYNOPSIS

The movie opens with an incident on the island of Isla Sorna (code-named "Site B" by the InGen company), located in the vicinity of Isla Nublar, where the film "Jurassic Park" took place.

Application Performance insights Sources Elements Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- static-v1 - http://localhost:1993
- static-v2 - http://localhost:1993
- Back/forward cache

Background Services

- Background Fetch


Total entries: 4

#	Name	Respons..	Content-..	Content-..	Time Ca...	Vary Hea..
0	/assets/img/jurassic-park-1993.jpg	basic	image/jpeg	137,013	04/07/2...	
1	/assets/img/jurassic-park-1997.jpg	basic	image/jpeg	74,859	04/07/2...	
2	/jurassic-park-1993.html	basic	text/html...	9,463	04/07/2...	
3	/jurassic-park-1997.html	basic	text/html...	10,744	04/07/2...	

Console What's New Issues

Filter

[script.js] Service Worker registered app.js:6



# CACHE CLEANUP

<https://github.com/rowdyrabouw/offline-first.nl/tree/07-cache-cleanup>



# JURASSIC PARK IS CLOSED



## IT SEEMS YOU ARE CURRENTLY OFFLINE ...

We're sorry, this page hasn't been cached yet :-)

But why don't you try one of our [other pages](#)?

Application Performance insights Sources Elements Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
  - static-v1 - http://localhost:1993
  - static-v2 - http://localhost:1993
  - dynamic-v1 - http://localhost:1993

Background Services

- Background Fetch

Service Workers

Offline  Update on reload  Bypass for network

http://localhost:1993/ [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 04/07/2022, 16:17:38

Status ● #2 activated and is running [stop](#)

Push

Sync

Periodic Sync

Update Cycle

Version	Update Activity	Timeline
▶ #2	Install	
▶ #2	Wait	█
▶ #2	Activate	█

Console What's New Issues

top Filter Default levels No Issues

[script.js] Service Worker registered app.js:6



```
self.addEventListener("activate", (event) => {
```

```
  console.log("%c[sw.js] Service Worker activated", "color: #FEC233");  
});
```



```
self.addEventListener("activate", (event) => {
  event.waitUntil(
    // will return an array of cache names
    caches.keys().then((keys) => {
      return Promise.all(
        // go over all available keys
        keys.map((key) => {
          if (key !== STATIC_CACHE && key !== DYNAMIC_CACHE) {
            console.log("%c[sw.js] Deleting old caches", "color: #FEC233");
            return caches.delete(key);
          }
        })
      );
    })
  );
  console.log("%c[sw.js] Service Worker activated", "color: #FEC233");
});
```



# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

DevTools Application Panel:

- Manifest
- Service Workers
- Storage
- Cache Storage
  - static-v1 - http://localhost:1993
  - static-v2 - http://localhost:1993
- Background Services

Service Workers for http://localhost:1993/:

- Source: sw.js
- Received: 04/07/2022, 16:29:59
- Status: #4 activated and is running
- Clients: http://localhost:1993/
- Push: Test push message from DevTools.
- Sync: test-tag-from-devtools
- Periodic Sync: test-tag-from-devtools
- Update Cycle:
 

Version	Update Activity	Timeline
#4	Install	
#4	Wait	█
#4	Activate	█

Console Panel:

- [script.js] Service Worker registered





# Prefetch

Prefetching is when content is downloaded in the background, this is based on the assumption that the content will likely be requested, enabling the content to load instantly if and when the user requests it.

The content is downloaded and cached for anticipated future use without the user making an explicit request for it.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.ico" />
    <script src="assets/js/app.js"></script>
    <link rel="prefetch" href="jurassic-park-1993.html" />
  </head>
  <body>

    ...

  </body>
</html>
```



```
const addPrefetchLink = (url) => {  
  link = document.createElement("link");  
  link.rel = "prefetch";  
  link.href = url;  
  document.head.appendChild(link);  
};
```



```
const addPrefetchLink = (url) => {
  link = document.createElement("link");
  link.rel = "prefetch";
  link.href = url;
  document.head.appendChild(link);
};

const linkList = document.querySelectorAll("a");
if (linkList) {
  linkList.forEach((link) => {
    addPrefetchLink(link);
  });
}
```







# WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Network Console Application Performance insights Sources Elements Recorder

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Jurassic Park</title>
    <link rel="stylesheet" href="assets/css/style.css">
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.png">
    <script src="assets/js/app.js"></script>
    <link rel="prefetch" href="http://localhost:1993/jurassic-park-1993.html">
    <link rel="prefetch" href="http://localhost:1993/jurassic-park-1997.html">
    <link rel="prefetch" href="http://localhost:1993/jurassic-park-2001.html">
    <link rel="prefetch" href="http://localhost:1993/jurassic-world-2015.html">
    <link rel="prefetch" href="http://localhost:1993/jurassic-world-2018.html">
    <link rel="prefetch" href="http://localhost:1993/jurassic-world-2022.html">
  </head>
  <body>
    <main>
      <h1>Jurassic Park</h1>
      
      <h2>Welcome to... Jurassic Park!</h2>
      <p>...</p>
      <p>...</p>
      <ol>...</ol>
    </main>
  </body>
</html> == $0
```

Styles Computed Layout

Filter :hov .cls +

```
element.style {
}

:root {
  --primary: rgb(250, 250, 250);
  --secondary: rgb(255, 27, 2);
  --tertiary: rgb(254, 246, 6);
  --dark: rgb(34, 34, 35);
  --light: rgb(220, 220, 220);
}

html {
  background-color: var(--dark);
  font-family: Arial, Helvetica, sans-serif;
  font-size: 1.5rem;
  color: var(--light);
}

html[Attributes Style] {
  -webkit-locale: "en";
}

html {
  user agent stylesheet
  display: block;
}
```

margin -  
border -  
padding -

Console What's New Issues

Filter Default levels No Issues

[script.js] Service Worker registered app.js:6

# # Resource Hints: prefetch -WD

Usage % of all users  
Global 78.25%

☆ Informs the browsers that a given resource should be prefetched so it can be loaded more quickly. This is indicated using `<link rel="prefetch" href="(url)">`

Current aligned Usage relative Date relative Filtered All ⚙

IE	Edge*	Firefox	Chrome	Safari	Opera	Safari on iOS*	Opera Mini*	Android Browser*	Opera Mobile*	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser		
								2.1-3										
			4-7	3.1-13	10-12.1	3.2-13.3		4										
6-10	12-102	2-101	8-102	13.1-15.4	15-85	13.4-15.4		4.1-4.3										
11	103	102	103	15.5	86	15.5	all	4.4-4.4.4	12-12.1	103	101	12.12	4-16.0	17.0	10.4	7.12	2.5	
		103-104	104-106	16.0-TP	87	16.0												

Notes Test on a real browser Known issues (0) Resources (3) Feedback

1 Can be enabled in Safari as `LinkPrefetch` in the developer menu



# OPTIMISED PREFETCH

<https://github.com/rowdyrabout/offline-first.nl/tree/08-optimised-prefetch>





```
// get all links on the page  
const linkList = document.querySelectorAll("a");
```



# data-\*

The data-\* global attributes form a class of attributes called custom data attributes, that allow proprietary information to be exchanged between the HTML and its DOM representation by scripts.



```
<a data-skip-prefetch href="...">... </a>
```



```
// get all links on the page without data-skip-prefetch attribute  
const linkList = document.querySelectorAll("a:not([data-skip-prefetch])");
```



# Intersection Observer API

The Intersection Observer API provides a way to asynchronously observe changes in the intersection of a target element with an ancestor element or with a top-level document's viewport.





```
const observer = new IntersectionObserver(  
  (entries) => {  
    entries.forEach((entry) => {  
      // if the link is in the viewport  
      if (entry.intersectionRatio === 1) {  
  
          fetch(entry.target.href, { cache: "no-store" });  
  
      }  
    });  
  },  
  { threshold: 1 }  
);
```



```
const observer = new IntersectionObserver(
  (entries) => {
    entries.forEach((entry) => {
      // if the link is in the viewport
      if (entry.intersectionRatio === 1) {

        fetch(entry.target.href, { cache: "no-store" });

      }
    });
  },
  { threshold: 1 }
);

// get all links on the page without data-skip-prefetch attribute
const linkList = document.querySelectorAll("a:not([data-skip-prefetch])");
if (linkList) {
  linkList.forEach((link) => {
    observer.observe(link);
  });
}
```



```
const observer = new IntersectionObserver(
  (entries) => {
    entries.forEach((entry) => {
      // if the link is in the viewport
      if (entry.intersectionRatio === 1) {
        const isFetched = localStorage.getItem(entry.target.href);
        // if the link is not fetched yet
        if (!isFetched) {
          fetch(entry.target.href, { cache: "no-store" }).then(() => {
            // save link information
            localStorage.setItem(entry.target.href, entry.target.innerText);
          });
        }
      }
    });
  },
  { threshold: 1 }
);

// get all links on the page without data-skip-prefetch attribute
const linkList = document.querySelectorAll("a:not([data-skip-prefetch])");
if (linkList) {
  linkList.forEach((link) => {
    observer.observe(link);
  });
}
```



```
const STATIC_CACHE = "static-v2";
```



```
const STATIC_CACHE = "static-v3";
```

# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application | Network | Console | Application | Performance insights | Sources | Elements | Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
  - static-v2 - http://localhost:1993
  - dynamic-v1 - http://localhost:1993
- Back/forward cache

Background Services

- Background Fetch
- Background Sync

Service Workers

Offline  Update on reload  Bypass for network

http://localhost:1993/ [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 04/07/2022, 17:26:45

Status ● #5 activated and is running [stop](#)

Push

Sync

Periodic Sync

Update Cycle

Version	Update Activity	Timeline
#5	Install	
#5	Wait	
#5	Activate	█

Console | What's New | Issues

top | Filter | Default levels | No Issues

```
[script.js] Service Worker registered app.js:6
```



# NetworkInformation.effectiveType

The `effectiveType` read-only property returns the effective type of the connection meaning one of 'slow-2g', '2g', '3g', or '4g'.



# # NetworkInformation API: effectiveType


Usage % of all users  
 Global 71.75%

☆ Current aligned Usage relative Date relative Filtered All ⚙️

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	12-18		4-60		10-47											
6-10	79-102	2-101	61-102	3.1-15.4	48-85	3.2-15.4		2.1-4.4.4	12-12.1				4-16.0			
11	103	102	103	15.5	86	15.5	all	103	64	103	101	12.12	17.0	10.4	7.12	2.5
		103-104	104-106	16.0	87	16.0										
				TP												

Notes Test on a real browser Feedback

This feature is **experimental**. Use caution before using in production.  
 See full reference on [MDN Web Docs](#).

Support data for this feature provided by:  
 MDN browser-compat-data



# NetworkInformation.saveData

The saveData read-only property returns true if the user has set a reduced data usage option on the user agent.



# # NetworkInformation API: saveData


Usage % of all users  
 Global 71.37%

☆ Current aligned Usage relative Date relative Filtered All ⚙️

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	12-18		4-64		10-51								4-8.2			
6-10	79-102	2-101	65-102	3.1-15.4	52-85	3.2-15.4		2.1-4.4.4	12-12.1				9.2-16.0			
11	103	102	103	15.5	86	15.5	all	103	64	103	101	12.12	17.0	10.4	7.12	2.5
		103-104	104-106	16.0	87	16.0										
				TP												

Notes Test on a real browser Feedback

This feature is **experimental**. Use caution before using in production.  
 See full reference on [MDN Web Docs](#).

Support data for this feature provided by:  
 MDN browser-compat-data



# OPTIMISED OFFLINE

<https://github.com/rowdyrabouw/offline-first.nl/tree/09-optimised-offline>



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park - offline</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.ico" />
    <script defer src="assets/js/app.js"></script>
  </head>
  <body>
    <main>
      <h1>Jurassic Park is closed</h1>
      
      <h2>It seems you are currently offline ...</h2>
      <p><strong>We're sorry, this page hasn't been cached yet :-(</strong></p>
      <p>But why don't you try one of our <a href="/">other pages</a>?</p>
    </main>
  </body>
</html>
```



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Jurassic Park - offline</title>
    <link rel="stylesheet" href="assets/css/style.css" />
    <link rel="icon" type="image/png" sizes="32x32" href="favicon.ico" />
    <script defer src="assets/js/offline.js"></script>
  </head>
  <body>
    <main>
      <h1>Jurassic Park is closed</h1>
      
      <h2>It seems you are currently offline ...</h2>
      <p><strong>We're sorry, this page hasn't been cached yet :-(</strong></p>
      <p>These pages are available offline:</p>
      <ul id="localStorageItems">
        <li><a href="/">Homepage</a></li>
      </ul>
    </main>
  </body>
</html>
```



```
const STATIC_CACHE = "static-v3";  
const STATIC_FILES = [  
  "/",  
  "index.html",  
  "/favicon.ico",  
  "assets/css/style.css",  
  "assets/fonts/neuland_inline.woff2",  
  "assets/img/jurassic-park.jpg",  
  "assets/js/app.js",  
  "offline.html",  
  
];
```



```
const STATIC_CACHE = "static-v4";
const STATIC_FILES = [
  "/",
  "index.html",
  "/favicon.ico",
  "assets/css/style.css",
  "assets/fonts/neuland_inline.woff2",
  "assets/img/jurassic-park.jpg",
  "assets/js/app.js",
  "offline.html",
  "assets/js/offline.js",
];
```



```
self.addEventListener("fetch", (event) => {
```

```
  event.respondWith(  
    // look at all caches for a match on the key (= request)  
    ...  
  );  
});
```





```
let setDynamicVersionInLocalStorage = true;  
self.addEventListener("fetch", (event) => {  
  
    event.respondWith(  
        // look at all caches for a match on the key (= request)  
        ...  
    );  
});
```

```
let setDynamicVersionInLocalStorage = true;

self.addEventListener("fetch", (event) => {
  if (setDynamicVersionInLocalStorage) {
    event.waitUntil(
      (async () => {
        // get the client
        const client = await clients.get(event.clientId);
        // send a message to the client
        client.postMessage({
          action: "setDynamicVersionInLocalStorage",
          dynamic: DYNAMIC_CACHE,
        });
        setDynamicVersionInLocalStorage = false;
      })()
    );
  }
  event.respondWith(
    // look at all caches for a match on the key (= request)
    ...
  );
});
```



```
// listen for messages from the Service Worker
navigator.serviceWorker.addEventListener("message", (event) => {
  if (event.data.action === "setDynamicVersionInLocalStorage") {
    console.log(
      `%c[app.js] Dynamic Cache version ${event.data.dynamic} set`,
      "color: #00ABD2"
    );
    localStorage.setItem("DYNAMIC_CACHE", event.data.dynamic);
  }
});
```



```
const retrieveCachedUrls = () => {
  const dynamicCacheVersion = localStorage.getItem("DYNAMIC_CACHE");
  const urls = [];
  caches.open(dynamicCacheVersion).then((cache) => {
    cache
      .keys()
      .then((keys) => {
        // retrieve all items from dynamic cache
        return Promise.all(
          keys.map((key) => {
            // only return pages
            if (key.url.endsWith(".html")) {
              urls.push(key.url);
              return key.url;
            }
          })
        );
      })
      .then(() => {
        cacheList(urls);
      });
  });
};
```



```
const cacheList = (urls) => {
  urls.forEach((url) => {
    // create a list item with link to cached page
    const listItem = document.createElement("li");
    const anchor = document.createElement("a");
    listItem.appendChild(anchor);
    // match the url with localStorage to retrieve text
    const text = localStorage[url] || url;
    const link = document.createTextNode(text);
    anchor.appendChild(link);
    anchor.href = url;
    // add link to offline page
    const element = document.getElementById("localStorageItems");
    element.appendChild(listItem);
  });
};

retrieveCachedUrls();
```



# JURASSIC PARK



## WELCOME TO... JURASSIC PARK!

Humans waited 65 million years for cloning technology to be advanced enough to re-create dinosaurs from DNA found in mosquitoes preserved in amber. If that's the timeline, the 29 years between Jurassic Park, the first film in the franchise, and the latest Jurassic World: Dominion doesn't seem all that long. But spreading six movies out across three decades does

Application Performance insights Sources Elements Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
  - http://localhost:1993
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups

Cache

- Cache Storage
  - dynamic-v1 - http://localhost:1993
  - static-v3 - http://localhost:1993
- Back/forward cache

Background Services

- Background Fetch

Service Workers

Offline  Update on reload  Bypass for network

http://localhost:1993/ [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 04/07/2022, 17:29:48

Status ● #6 activated and is running [stop](#)

Push

Sync

Periodic Sync

Update Cycle

Version	Update Activity	Timeline
▶ #6	Install	
▶ #6	Wait	■
▶ #6	Activate	■

Console What's New Issues

Filter

Default levels No Issues

[app.js] Observing links without data-skip-prefetch attribute app.js:60

[script.js] Service Worker registered app.js:6





# API RESPONSES

<https://github.com/rowdyrabouw/offline-first.nl/tree/10-api-responses>

# IndexedDB

NoSQL database

Tables are object stores

Store values by keys

Supports transactions for reliability

Supports indexes





# IndexedDB with usability

Promises enables async / await

Shortcuts for common transactions like getAll, put, delete

<https://github.com/jakearchibald/idb>



```
const fetchData = async (dino) => {  
  const response = await fetch(`http://localhost:3000/${dino}`);  
  const data = await response.json();  
  document.querySelector("p").innerHTML = data.description;  
}
```



```
const fetchData = async (dino) => {  
  const response = await fetch(`http://localhost:3000/${dino}`);  
  const data = await response.json();  
  document.querySelector("p").innerHTML = data.description;  
}
```

```
document.querySelector("#velo").addEventListener("click", () => {  
  fetchData("Velociraptor");  
});
```

```
document.querySelector("#trex").addEventListener("click", () => {  
  fetchData("Tyrannosaurus");  
});
```



```
import './idb.js';

let db;

const openDatabase = async () => {
  db = await idb.openDB('dino', 1, {
    upgrade(db, oldVersion, newVersion, transaction, event) {
      if (oldVersion < 1) {
        db.createObjectStore('dinosaurs', { keyPath: "name" });
      }
    }
  });
};
```

```
import './idb.js';

let db;

const openDatabase = async () => {
  db = await idb.openDB('dino', 1, {
    upgrade(db, oldVersion, newVersion, transaction, event) {
      if (oldVersion < 1) {
        db.createObjectStore('dinosaurs', { keyPath: "name" });
      }
    }
  });
};

self.addEventListener("activate", async () => {
  console.log("%c[sw.js] Service Worker activated", "color: #FEC233");
  await openDatabase();
  console.log("%c[sw.js] IndexedDB available", "color: #FEC233");
  return self.clients.claim();
});
```



```
self.addEventListener("fetch", (event) => {
```

```
});
```



```
self.addEventListener("fetch", (event) => {  
  const url = new URL(event.request.url);  
  if (url.host === "localhost:3000") {
```

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



```
self.addEventListener("fetch", (event) => {  
  const url = new URL(event.request.url);  
  if (url.host === "localhost:3000") {  
    event.respondWith(  
      (async () => {  
        const dinosaurName = url.pathname.substring(1);  
        const cachedResponse = await db.get('dinosaurs', dinosaurName);
```

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





```
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);
  if (url.host === "localhost:3000") {
    event.respondWith(
      (async () => {
        const dinosaurName = url.pathname.substring(1);
        const cachedResponse = await db.get('dinosaurs', dinosaurName);
        if (cachedResponse) {
          console.log("%c[sw.js] Returning cached response", "color: #FEC233");
          return new Response(JSON.stringify(cachedResponse));
        }
      })()
    );
  }
});
```

```
});
```



```
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);
  if (url.host === "localhost:3000") {
    event.respondWith(
      (async () => {
        const dinosaurName = url.pathname.substring(1);
        const cachedResponse = await db.get('dinosaurs', dinosaurName);
        if (cachedResponse) {
          console.log("%c[sw.js] Returning cached response", "color: #FEC233");
          return new Response(JSON.stringify(cachedResponse));
        }
        console.log("%c[sw.js] Fetching from network", "color: #FEC233");
        const response = await fetch(event.request);
        const clonedResponse = response.clone();
        const data = await clonedResponse.json();
        await db.put('dinosaurs', data);
        return response;
      })()
    );
  }
});
```



```
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);
  if (url.host === "localhost:3000") {
    event.respondWith(
      (async () => {
        const dinosaurName = url.pathname.substring(1);
        const cachedResponse = await db.get('dinosaurs', dinosaurName);
        if (cachedResponse) {
          console.log("%c[sw.js] Returning cached response", "color: #FEC233");
          return new Response(JSON.stringify(cachedResponse));
        }
        console.log("%c[sw.js] Fetching from network", "color: #FEC233");
        const response = await fetch(event.request);
        const clonedResponse = response.clone();
        const data = await clonedResponse.json();
        await db.put('dinosaurs', data);
        return response;
      })()
    );
  }
});
```





VELOCIRAPTOR

TYRANNOSAURUS

- Application
  - Manifest
  - Service workers**
  - Storage
- Storage
  - Local storage
  - Session storage
  - IndexedDB
  - Cookies
  - Private state tokens
  - Interest groups
  - Shared storage
  - Cache storage
  - Storage buckets
- Background services
  - Back/forward cache
  - Background fetch
  - Background sync

### Service workers

Offline  Update on reload  Bypass for network

**http://127.0.0.1:1993/dino/** [Network requests](#) [Update](#) [Unregister](#)

Source [sw.js](#)

Received 16/10/2024, 22:53:00

Status ● #1 activated and is running [Stop](#)

Push  [Push](#)

Sync  [Sync](#)

Periodic sync  [Periodic sync](#)

Update Cycle

Version	Update Activity	Timeline
▶ #1	Install	<div style="width: 10px; height: 10px; background-color: blue;"></div>
▶ #1	Wait	<div style="width: 10px; height: 10px; background-color: gray;"></div>
▶ #1	Activate	<div style="width: 10px; height: 10px; background-color: yellow;"></div>

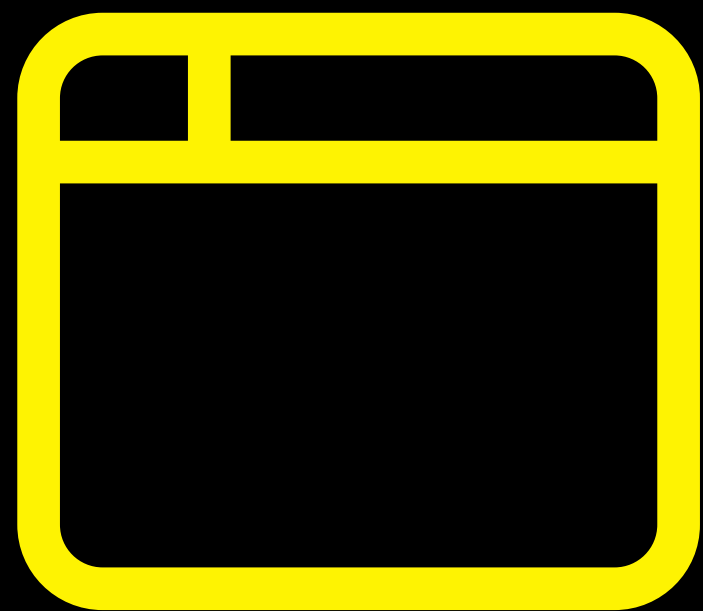
>



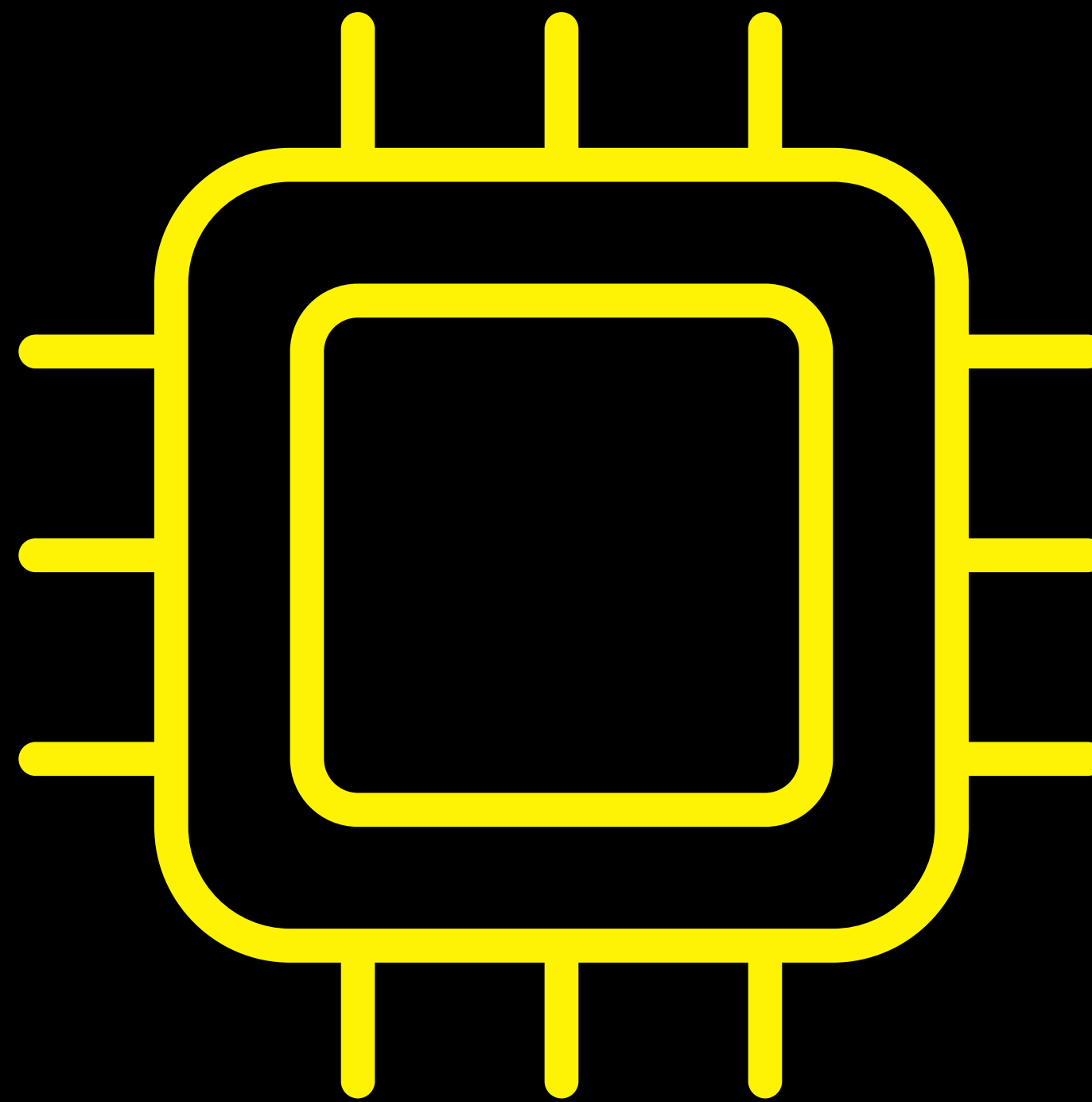


# CACHING STRATEGIES





**Browser**



**Service Worker**

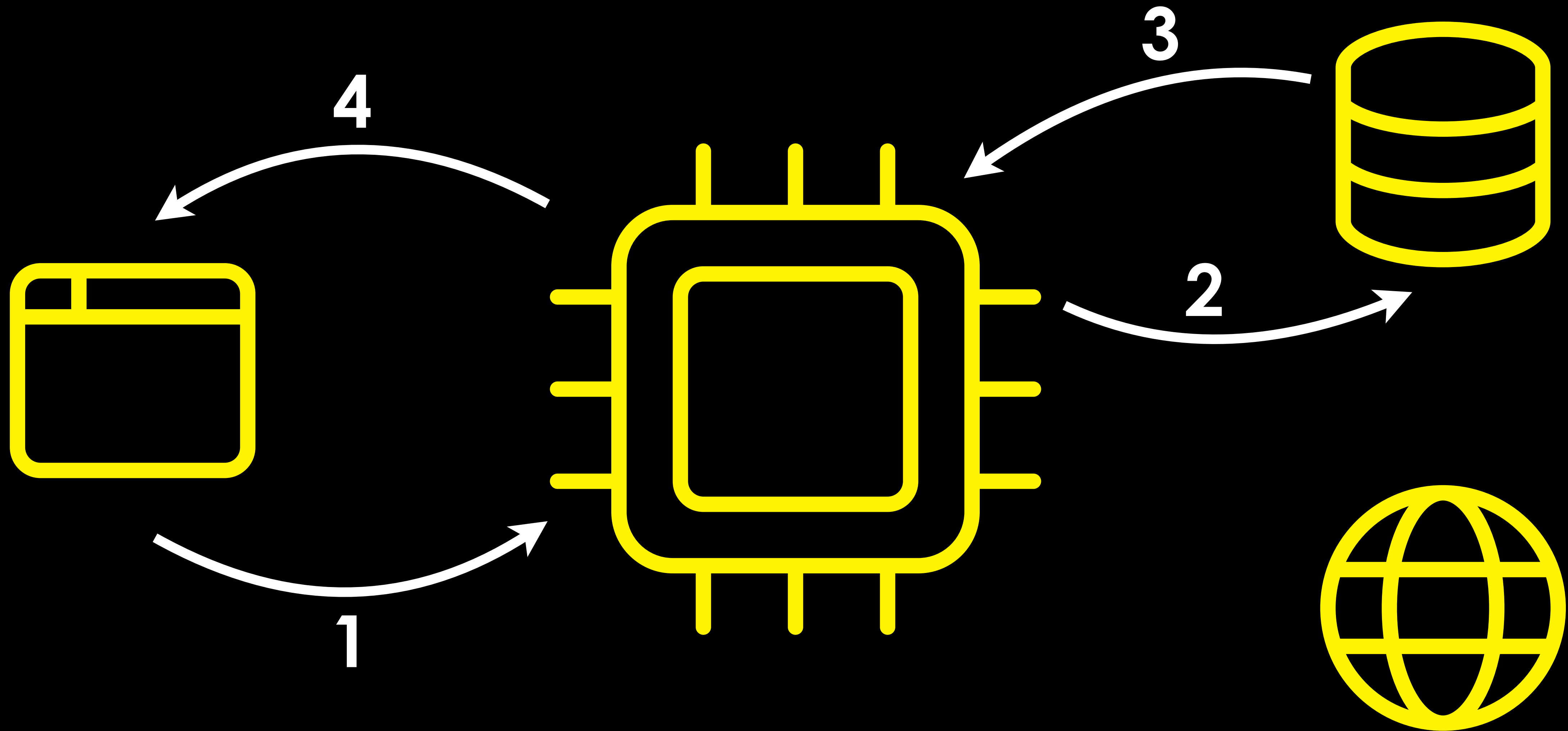


**Cache**

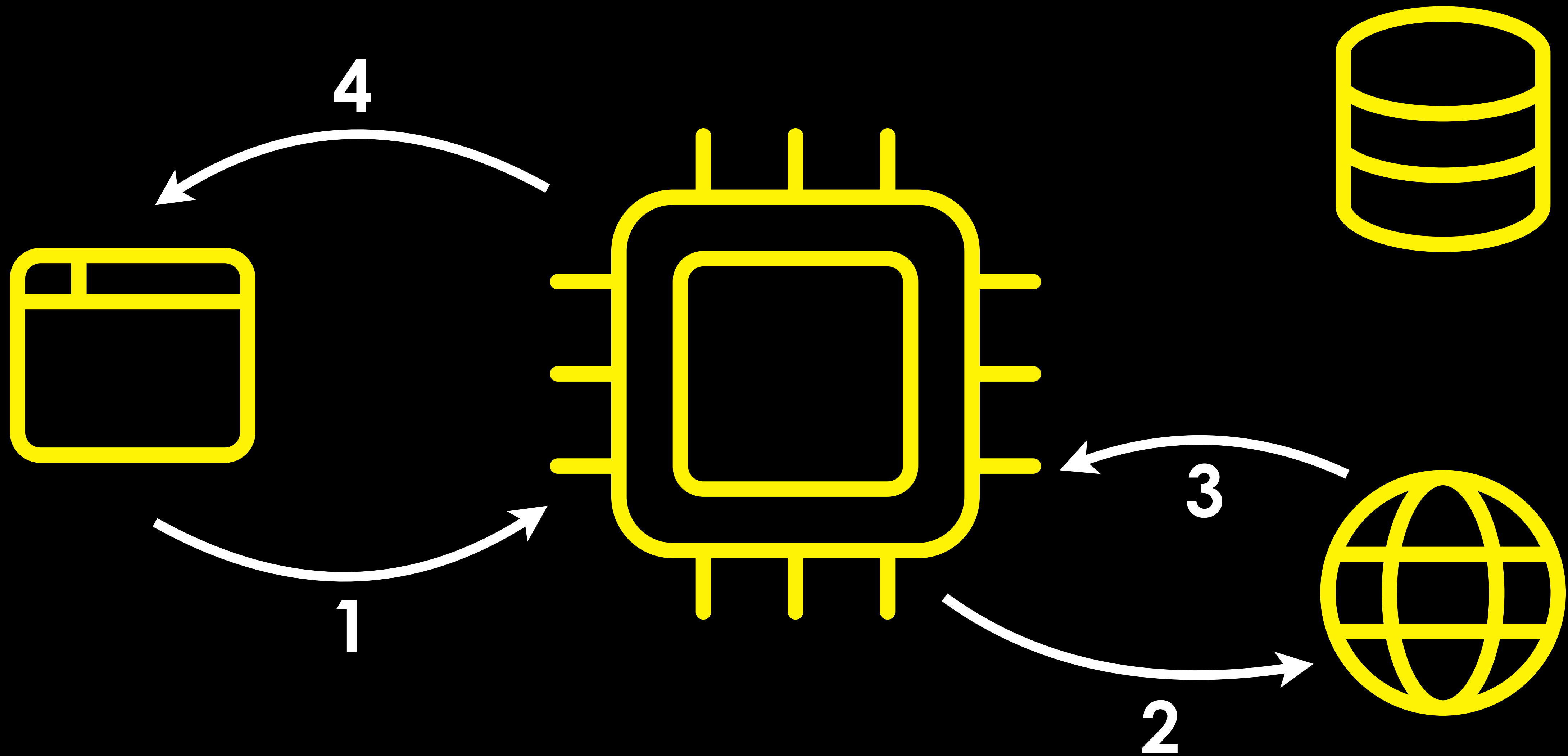
**Network**



# Cache only

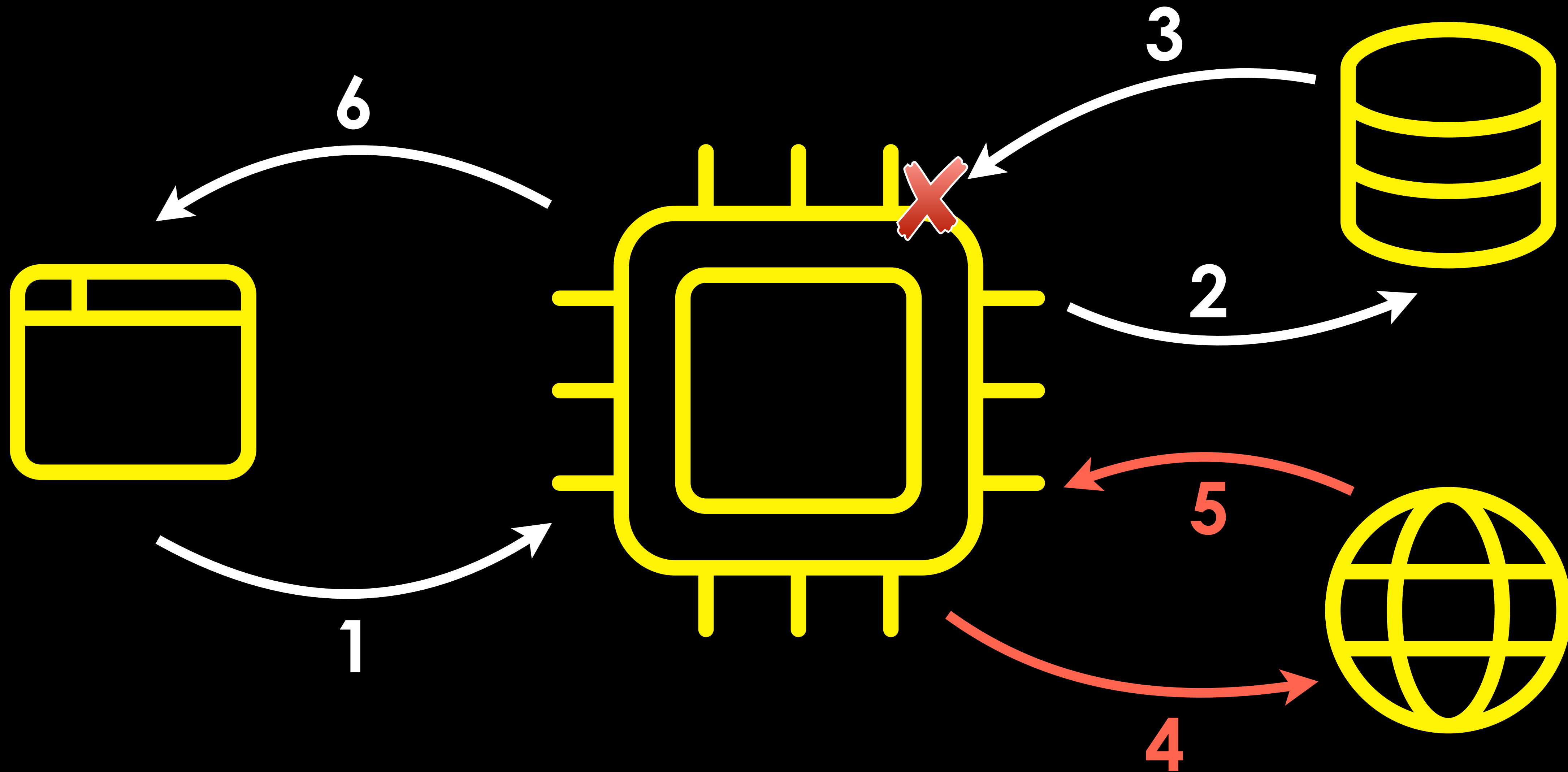


# Network only

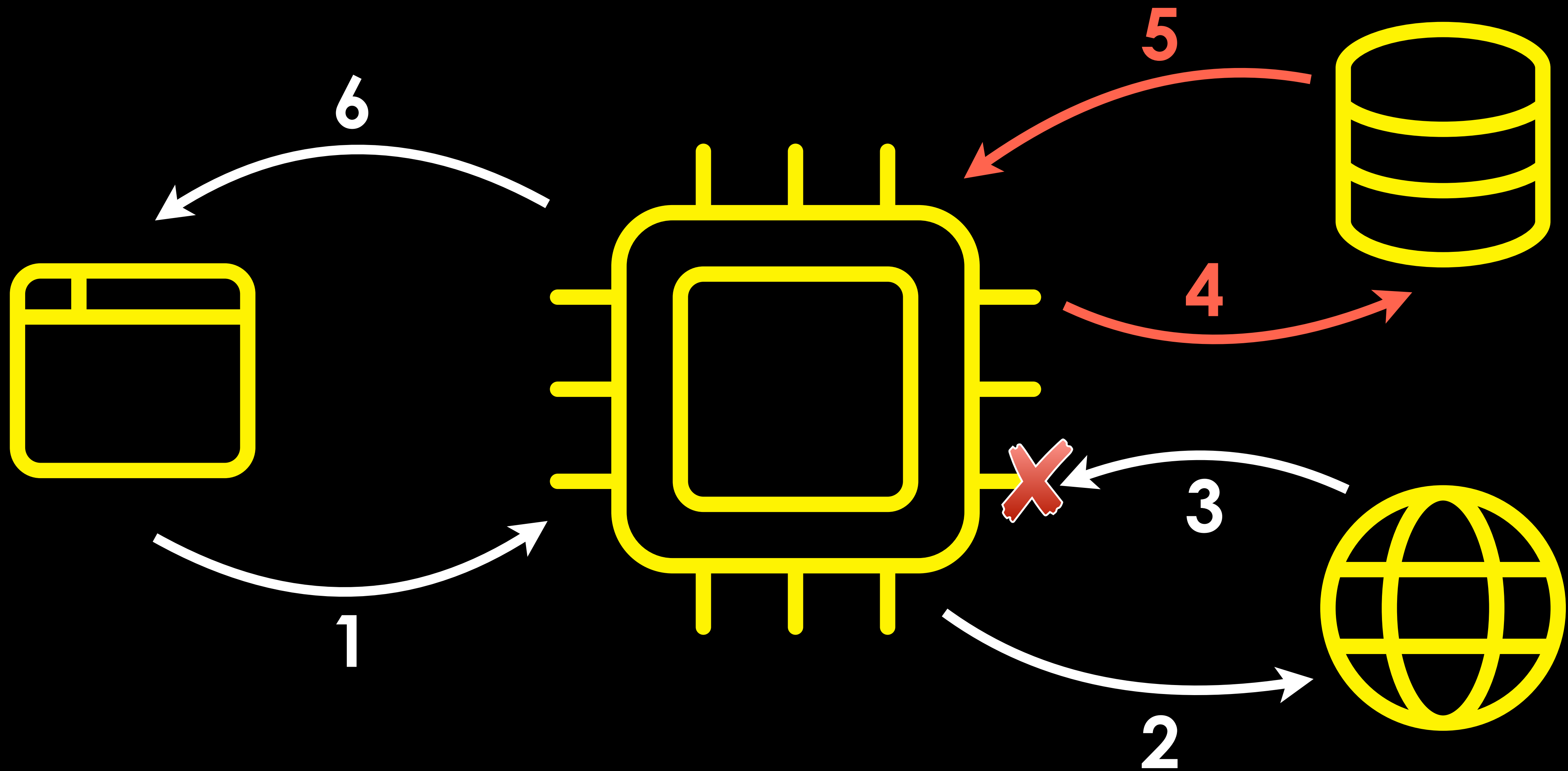




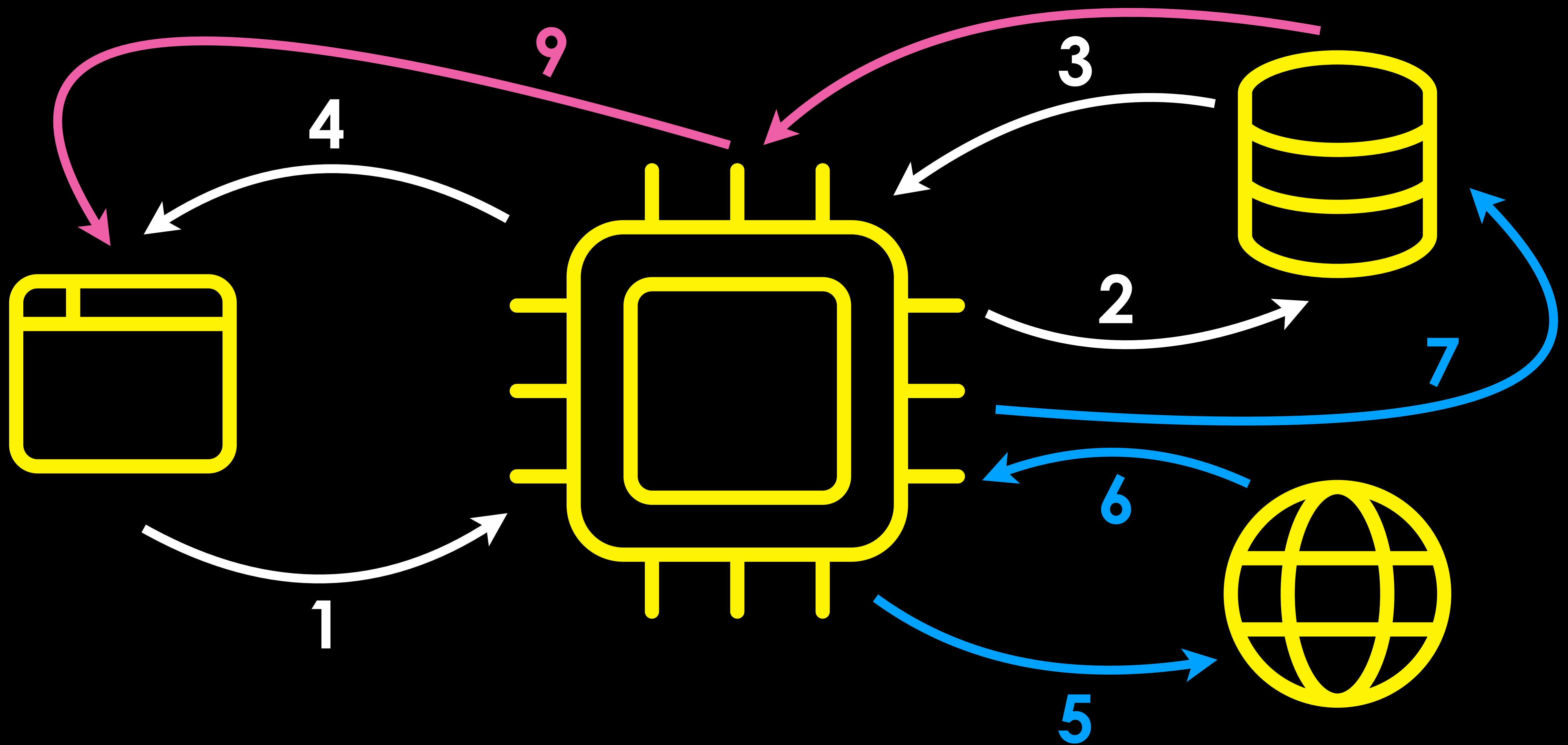
# Cache first - Network fallback



# Network first - Cache fallback



# Cache only - Cache update



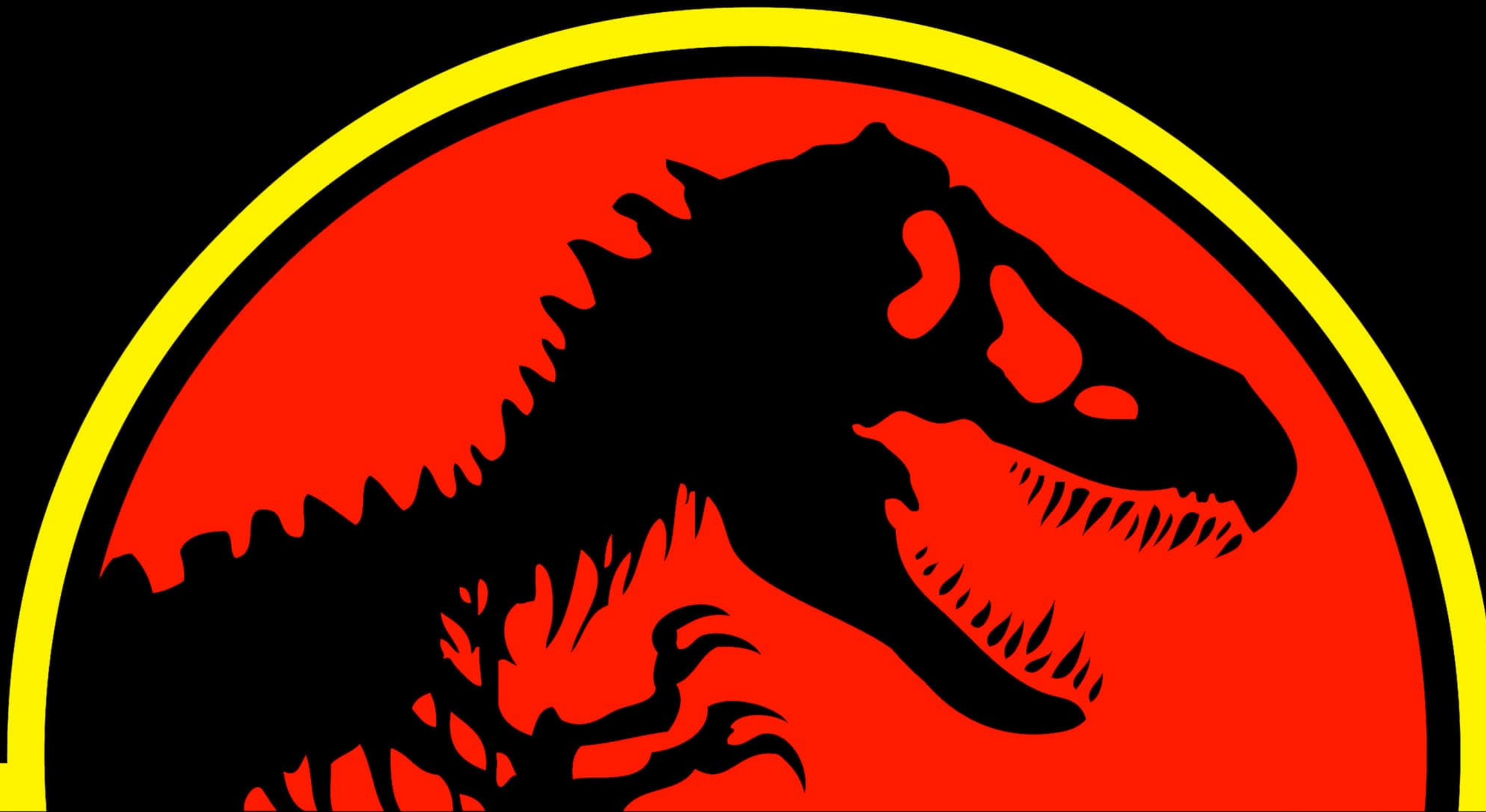


[developer.chrome.com/docs/workbox/](https://developer.chrome.com/docs/workbox/)



OFFLINE FIRST!  
@rowdyrabouw





**OBRIGADO !!!**

<https://rowdy.codes/ndc>

