Erik Witt

Service Workers in Action

Building an Offline-Capable Webshop in under 90 minutes

Mobile

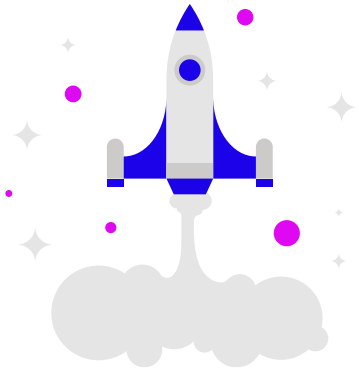
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

code.talks

Baqend

What I Have Todo with Service Workers?

+



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Baqend

Research:
- NoSQL Databases
- Caching Algorithms for Dynamic Data

Practice:
- Speed Kit Lead
- Combining Caching Algorithms with Service Workers

Theory

Practice

Agenda

Case Study

PWA Overview

Basic PWA

Service Wroker Overview

Offline + Caching Strategies

Cache Coherence

Implement Speed Kit

Implement Web Push

Demo

Page: https://bit.ly/2p7bA2X

Repository:
https://github.com/ErikWitt/code-talks19

Demo

https://bit.ly/2p7bA2X

What are Progressive Web Apps?

Why Do(n't) We Love Native Apps?

Great.

On **Homescreen**
In **App Stores**
Loading **Fast**
Work **Offline**
Use **Phone APIs**
Receive **Push Notifications**

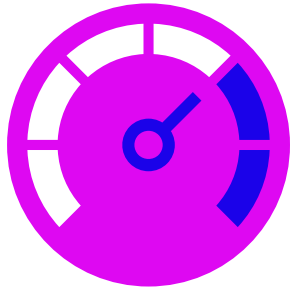
Weak.

Need **Installation**
Not **Cross Platform**
Tedious **Release and Update Processes**
Maintaining **Multiple Versions**

Progressive Web Apps

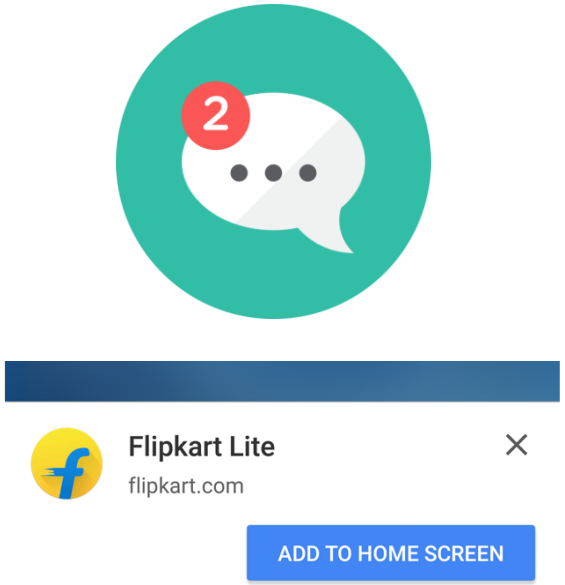
Combine the best from **native** and **web apps**.

Progressive Web Apps (PWA)



+

+



Fast **Loads** through Caching

Offline Mode (Synchronization)

Add-to-**Homescreen** and **Push Notifations**

Advantages of PWAs

Discoverable
E.g. in search engines

Progressive
Enhance on capable browsers

Installable
Easy access from home screen

Re-engageable
Engage through Web Push

Linkable
Link into apps through URLs

Responsive
Fit any form factor

Network independant
Offline mode

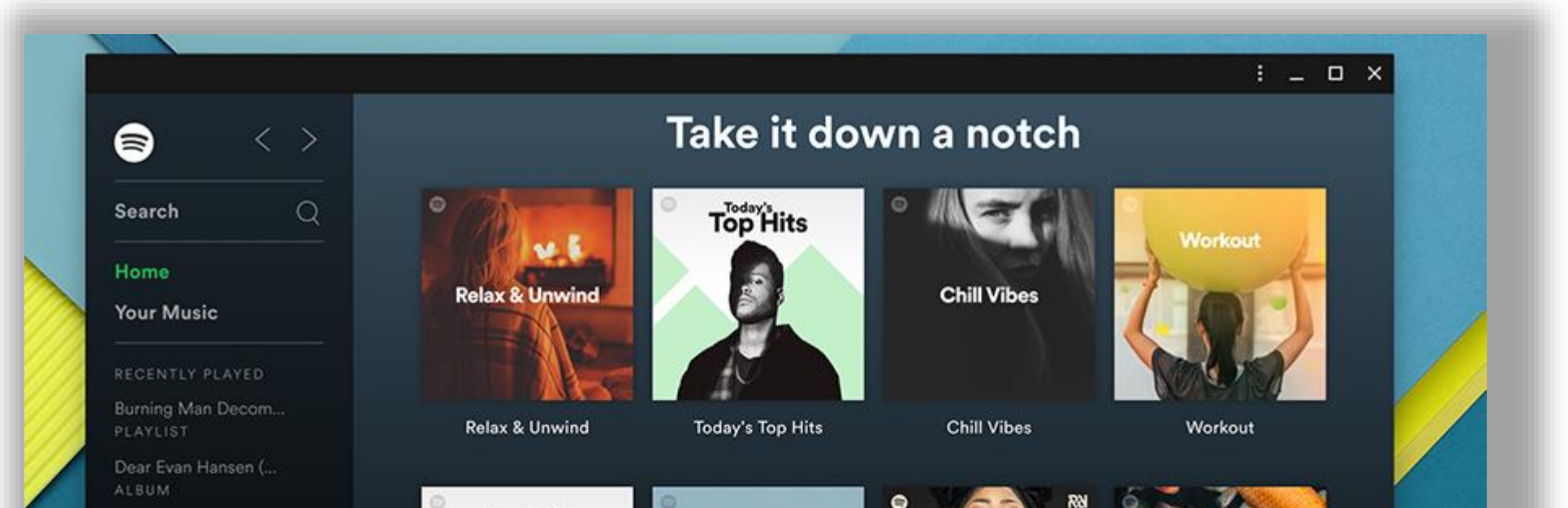
Safe
HTTPS & recognizable URLs

https://developer.mozilla.org/en-US/docs/Web/Apps/Progressive/Advantages

Just Released: Desktop PWAs

Chrome >73 now supports **Desktop PWAs** on every platform

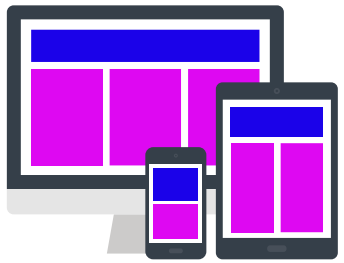
Customizable installation process/UI (with Event beforeinstallprompt)

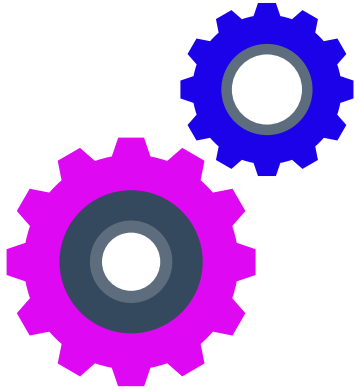


PWA Building Blocks

PWAs are **best practices** and **open web standards**

Progessively enhance when supported





1. Manifest

2. Service Worker

PWA Implementation

PWAs are **best practices** and **open web standards**

Progessively enhance when supported

1. Manifest declares Add-to-Homescreen:

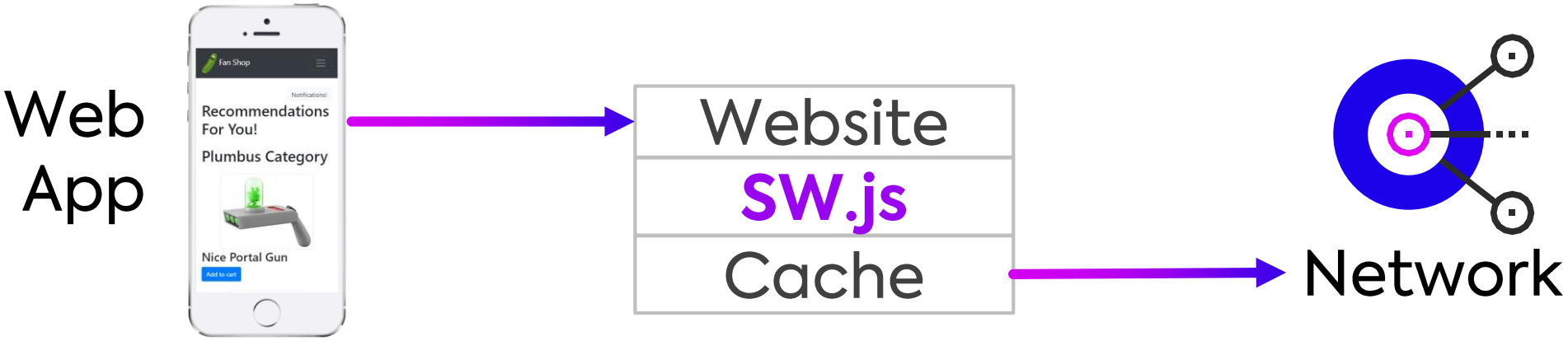
```
<link rel="manifest" href="/manifest.json">

{
  "short_name": "DAHO.AM PWA",
  "icons": [
    {"src": "icon-1x.png", "type": "image/png", "sizes": "48x48"}],
  "start_url": "index.html?launcher=true"
}
```

PWA Implementation

PWAs are **best practices** and **open web standards**

Progessively enhance when supported

2. Service Workers for caching & offline mode:

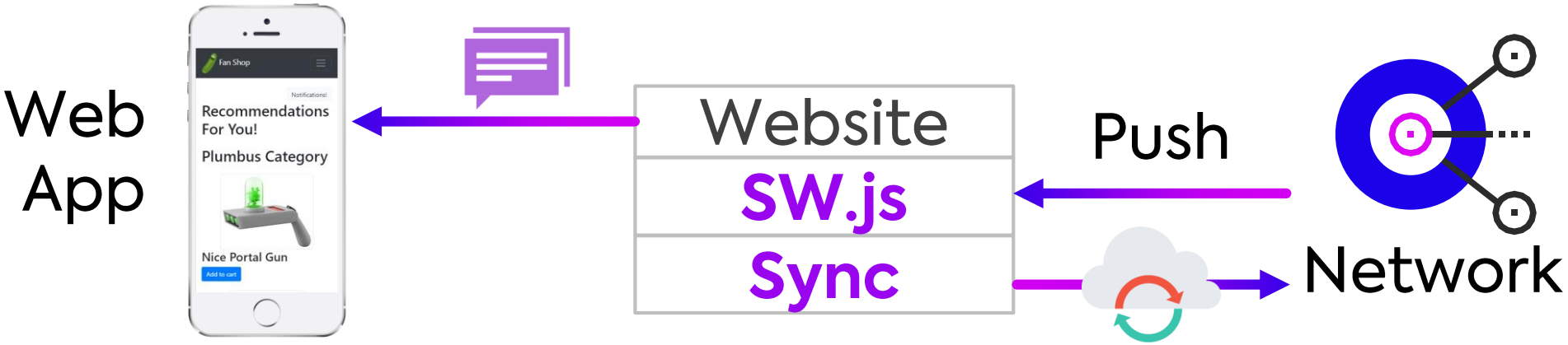


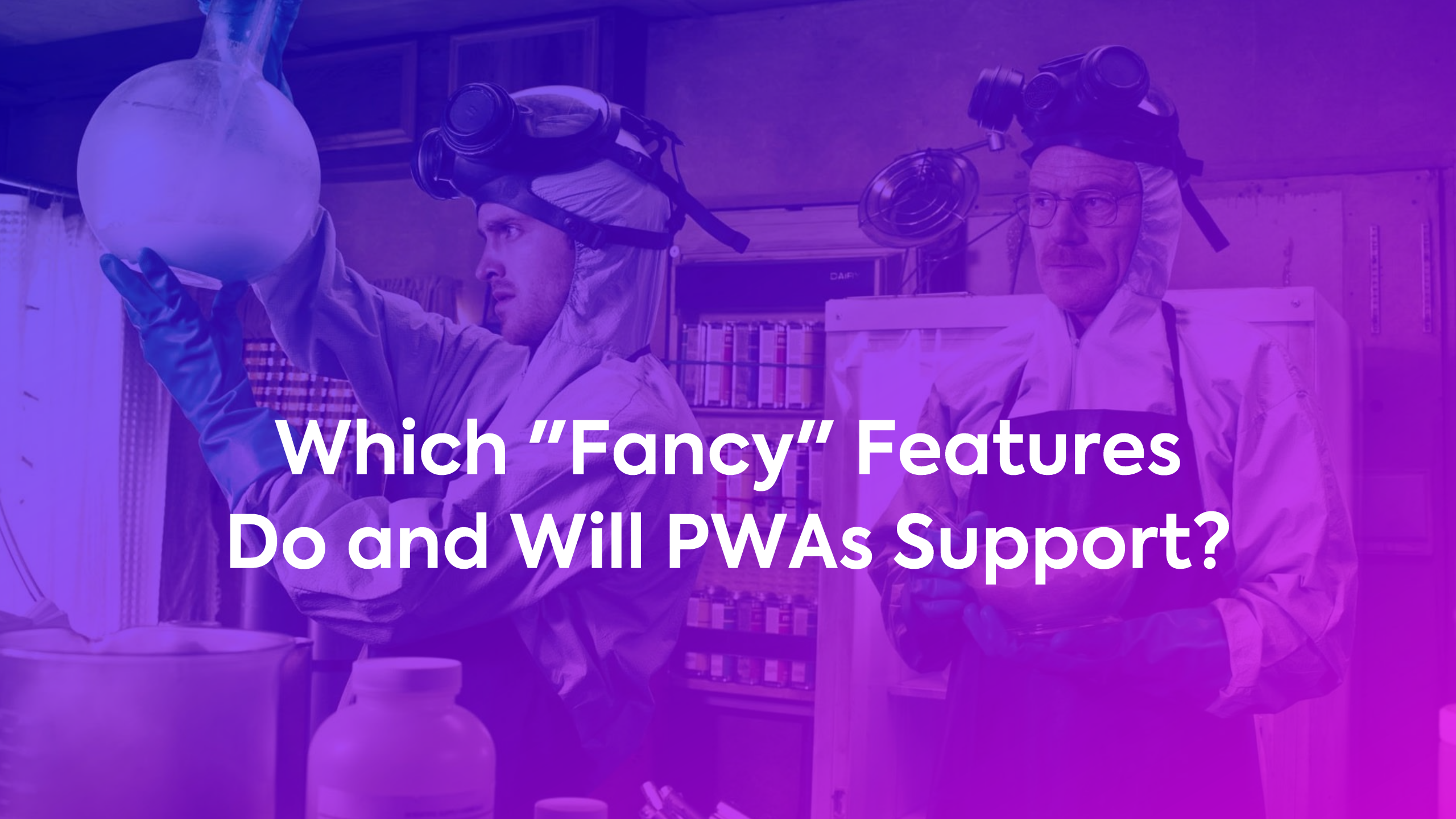
PWA Implementation

PWAs are **best practices** and **open web standards**

Progessively enhance when supported

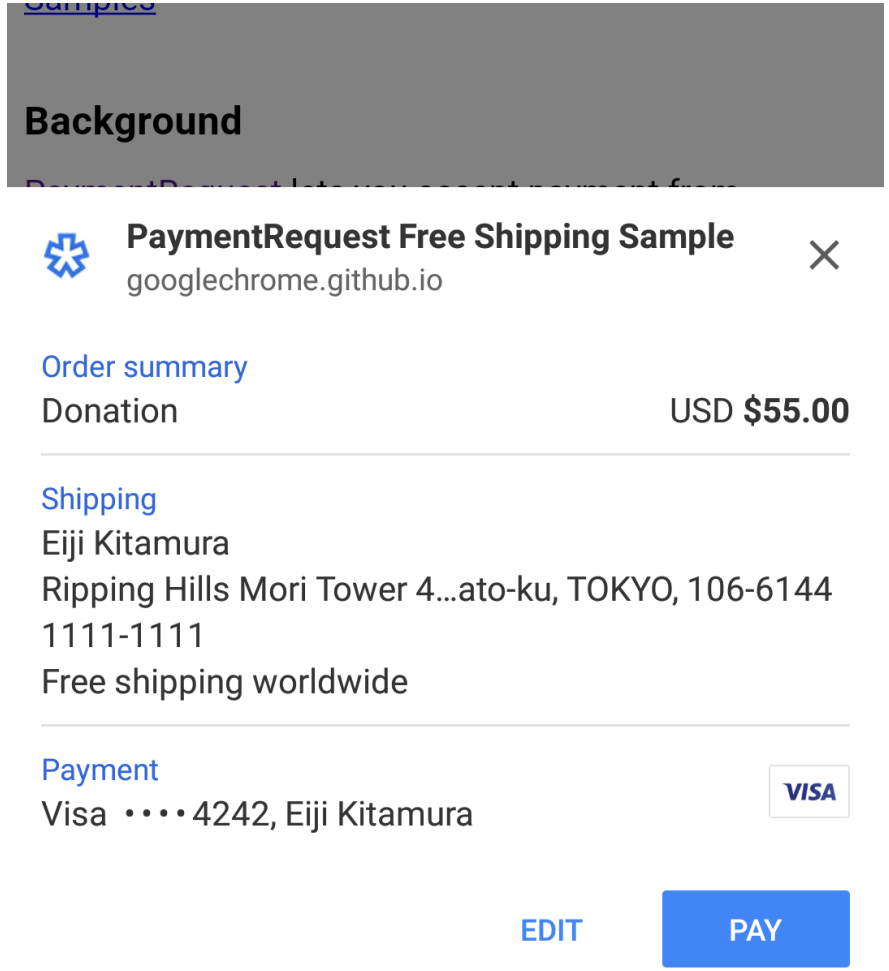
3. Add Web Push and Background Sync:





Which "Fancy" Features Do and Will PWAs Support?

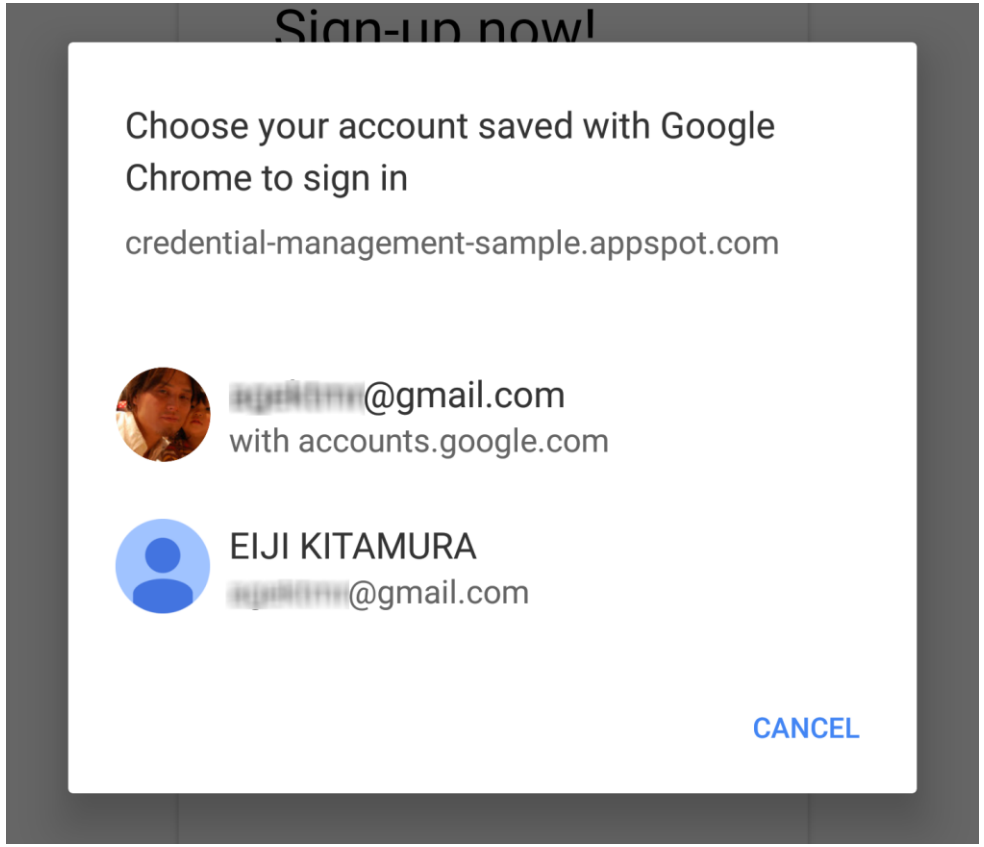
Integrated Payment



Web Payment APIs

- Goal: replace traditional **checkout** forms
- Just ~10 LOC to implement **payment**
- Vendor- & Browser-**Agnostic**

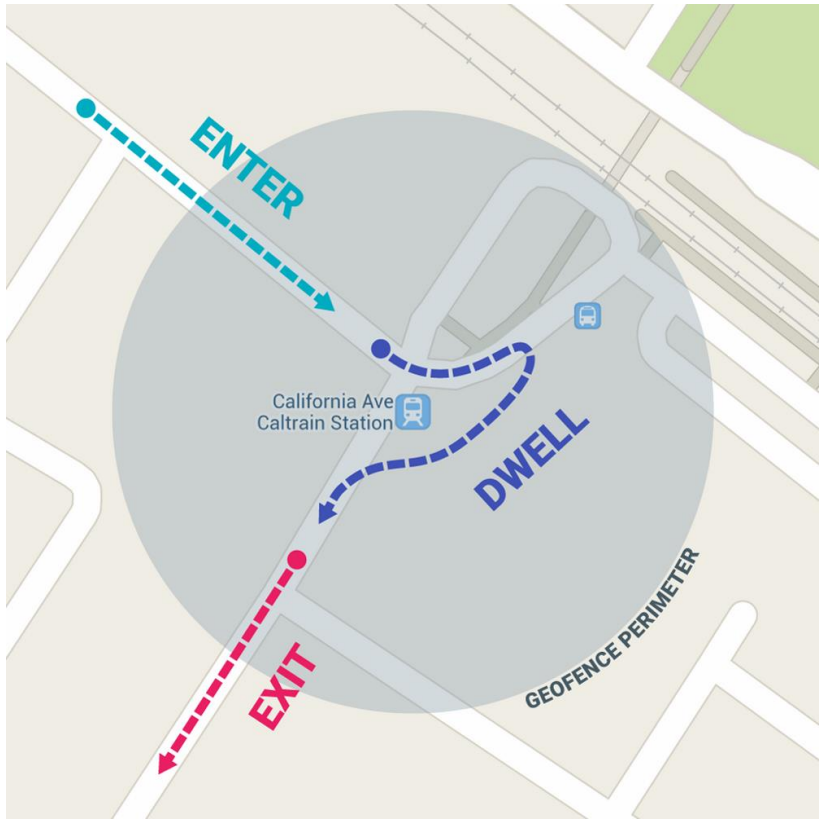
User Management



Credentials Management API

1. Click **Sign-in** → Native Account Chooser
2. Credentials API **stores** information for future use
3. **Automatic** Sign-in afterwards

Leverage Geolocation



Geofencing

- **Notify** web app when user leaves or enters a defined area
- Requires **permission**

Conversational Interfaces

Web Speech API

Native Speech Recognition in the Browser:

```
annyang.addCommands({
  'Hello Code Talks': () => {
    console.log('Hello you.');
  }
});
```

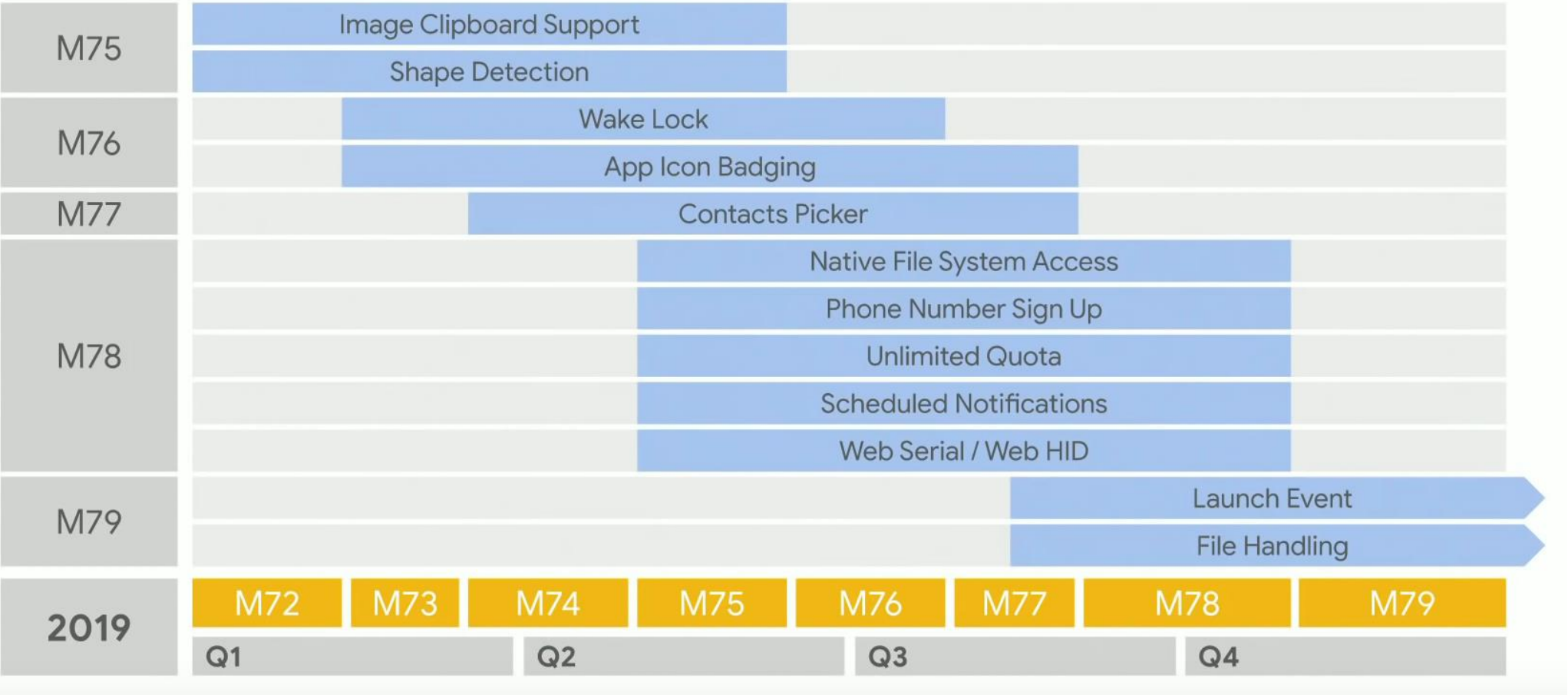
Seamless Sharing

Web Share API

- **Share** site through native share sheet UI
- Service Worker can register as a **Share Target**

Many More in the Pipeline

Planned Shipping Dates of Chrome PWA Features.

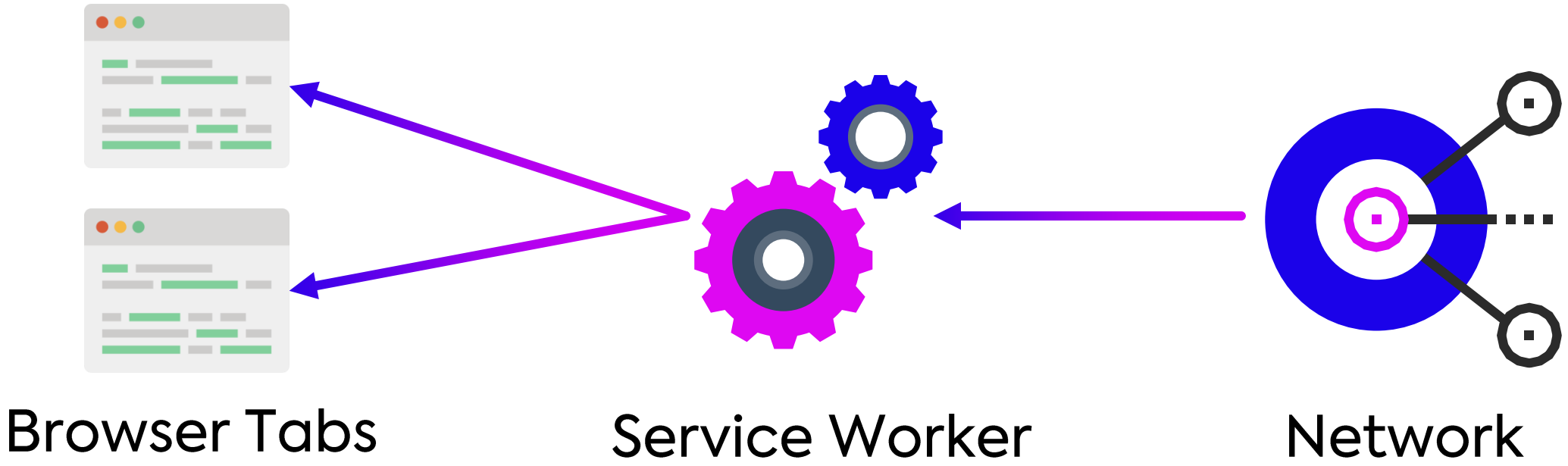


https://www.youtube.com/watch?v=2KhRmFHLuhE (Google I/O 2019)

Demo

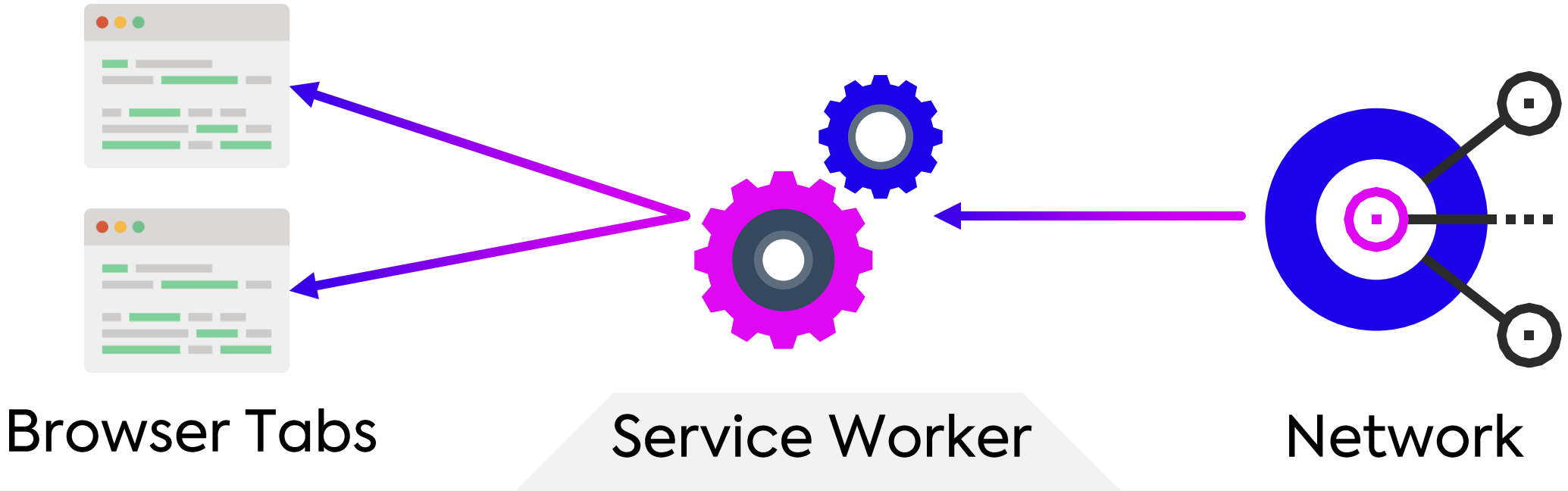
What are Service Workers?

What Are Service Workers



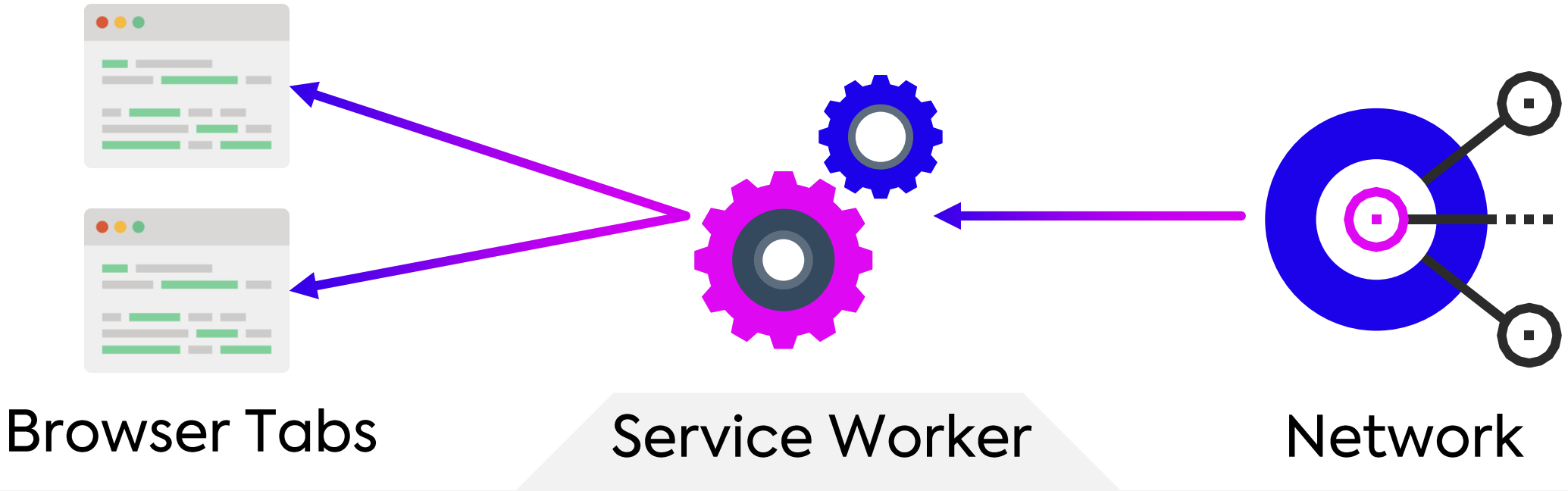
Programmable **Network Proxy**, running as a separate **Background Process**, without any **DOM Access**.

What do Service Workers Do?



- **Cache** Data (CacheStorage)
- **Store** Data (IndexedDB)
- Receive **Push**
- Respond when **Offline**

What do Service Workers Do?



- **Intercept** HTTP Requests
- **Sync** Data in Background
- Hide **Flaky Connectivity** from the User

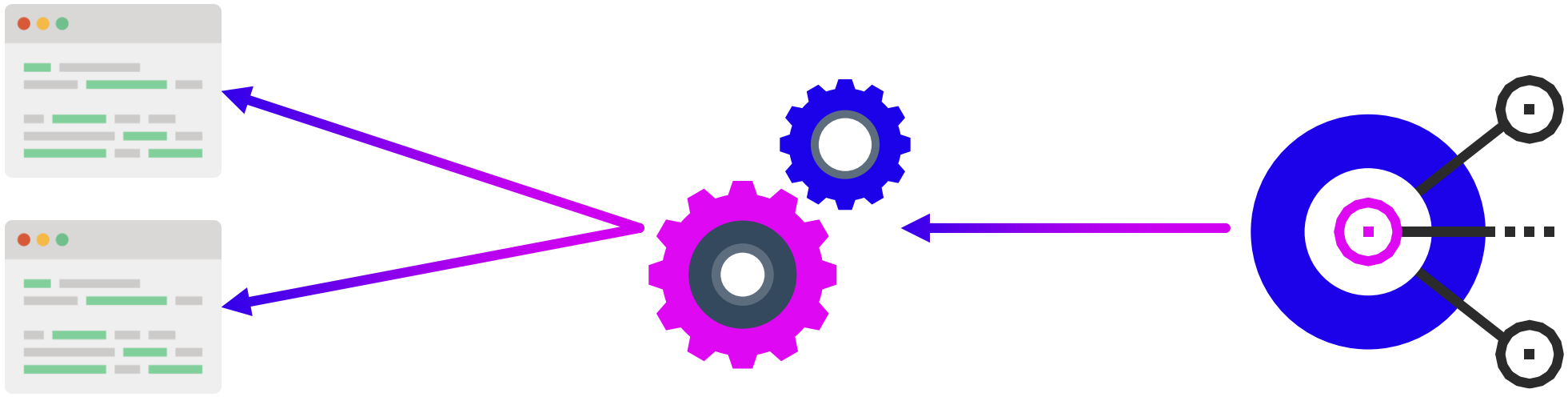
Service Workers Browser Support

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			75	5.1	12.1				
	17	68	76	12.1	12.4				9.2
11	18	69	77	13	13.1	all	76	12.12	10.1
	76	70	78	TP					
		71	79						
			80						

Supported by **>90%** of browsers.

Requires **TLS Encryption**.

How are Service Workers Registered



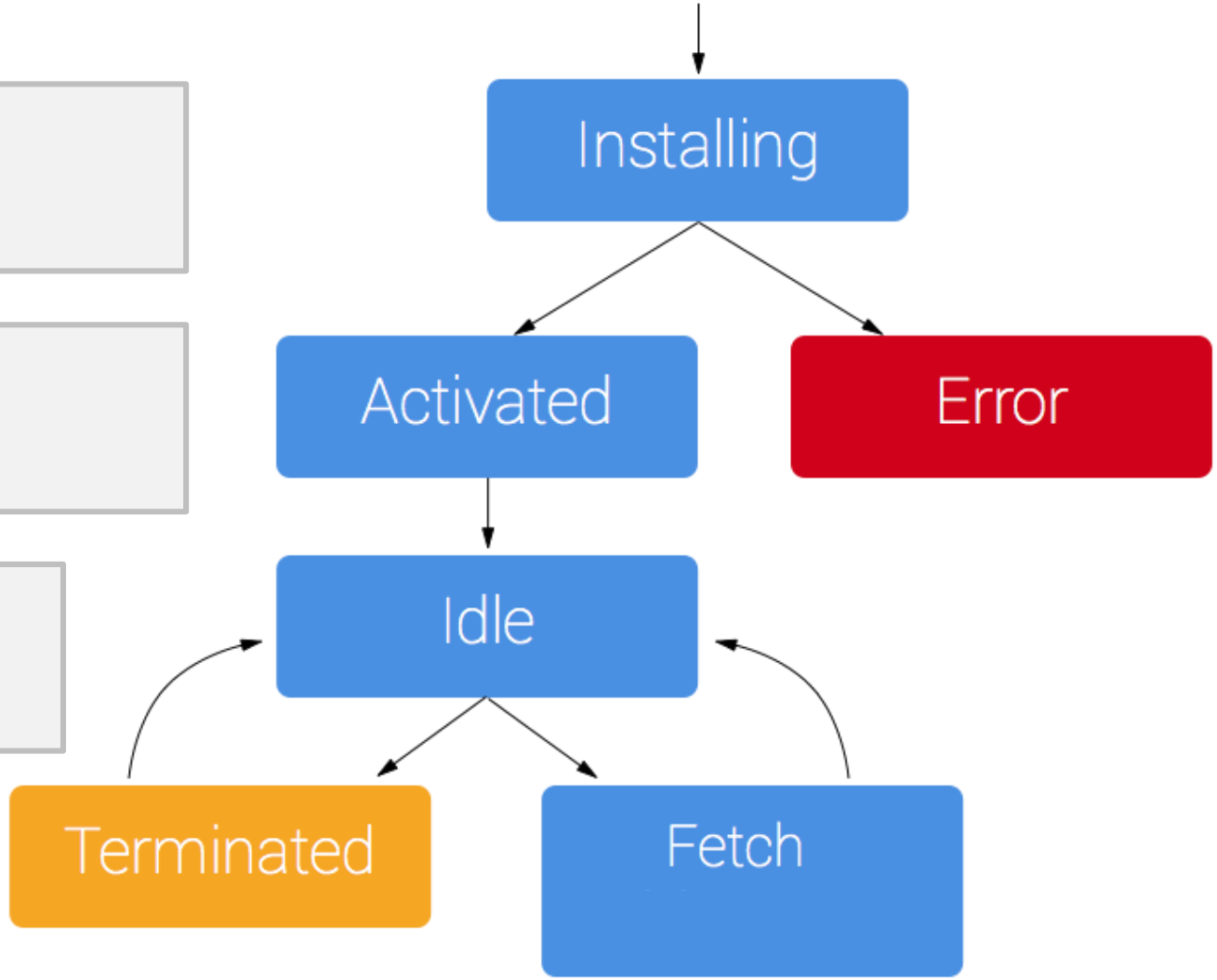

```
<script>
    navigator.serviceWorker.register('/sw.js');
</script>
```

Service Worker Lifecycle

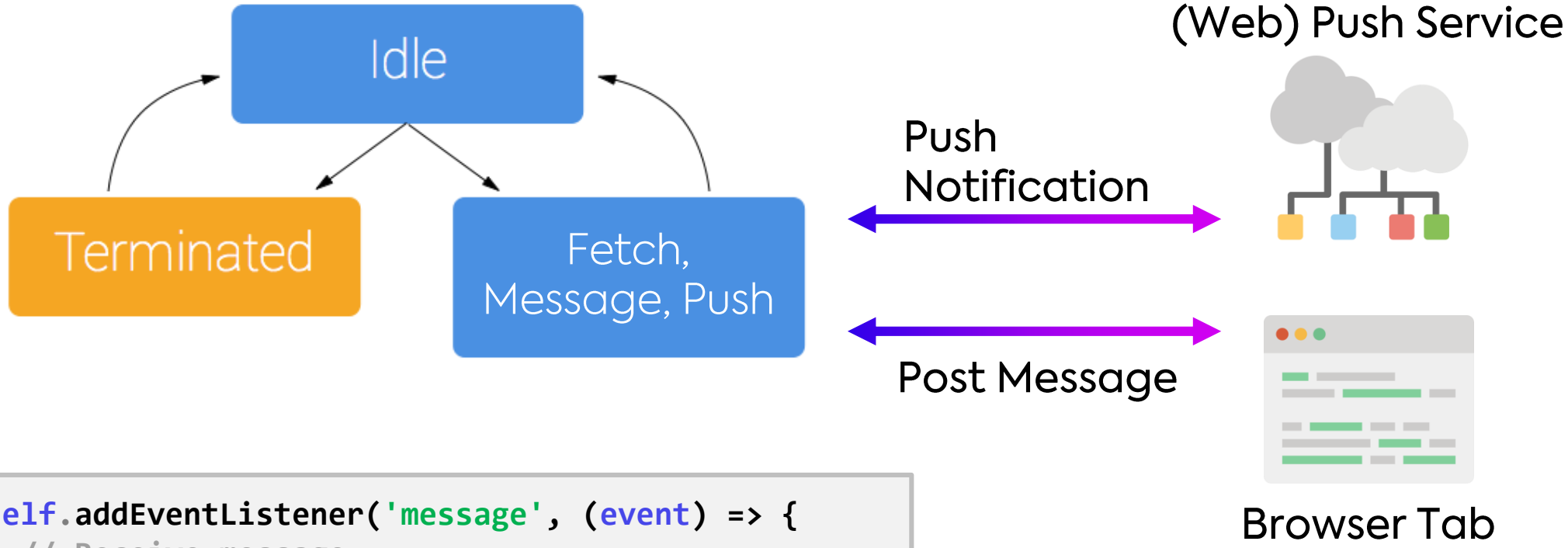
```
self.addEventListener('install', (event) => {
  // Perform install steps
});
```

```
self.addEventListener('activate', (event) => {
  // Perform activate steps
});
```

```
self.addEventListener('fetch', (event) => {
  // React to fetch event
});
```



Service Worker Communication

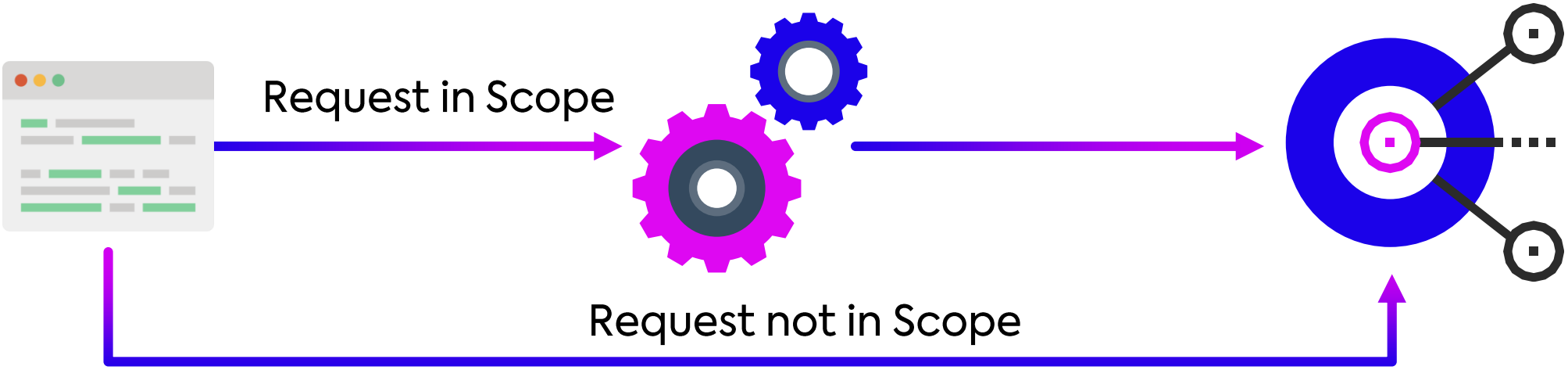



```
self.addEventListener('message', (event) => {
  // Receive message
});
```

```
// Send message to browser tab
const client = await clients.get('id');
client.postMessage(someJsonData);
```

```
self.addEventListener('push', (event) => {
  // Receive push notification
});
```

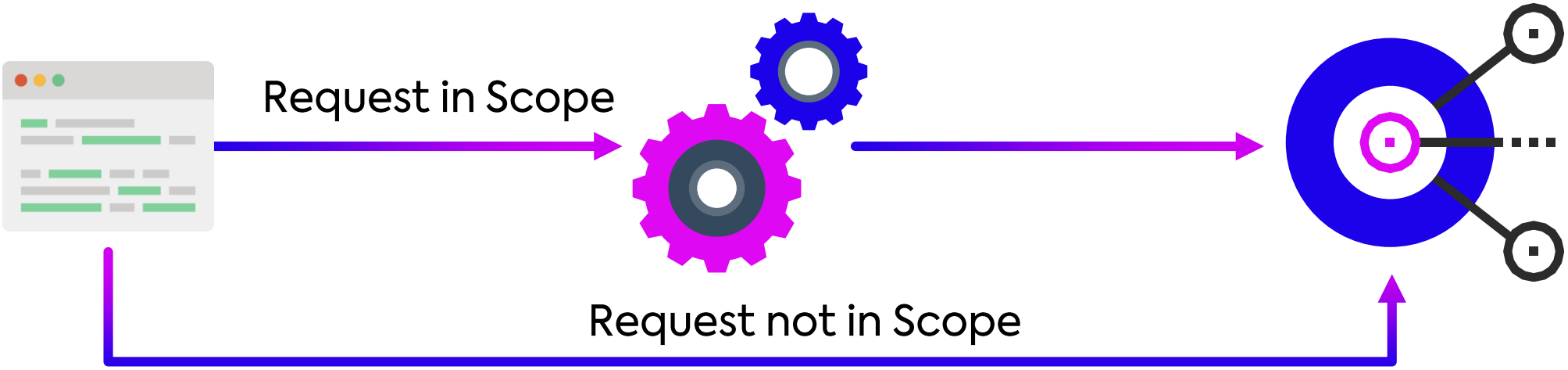
Service Worker Scope



Scope determines which requests go to the Service Worker

```
// Default (and maximum) scope is location of Service Worker
// Gets all requests starting with '/path/'
navigator.serviceWorker.register('/path/sw.js');
```

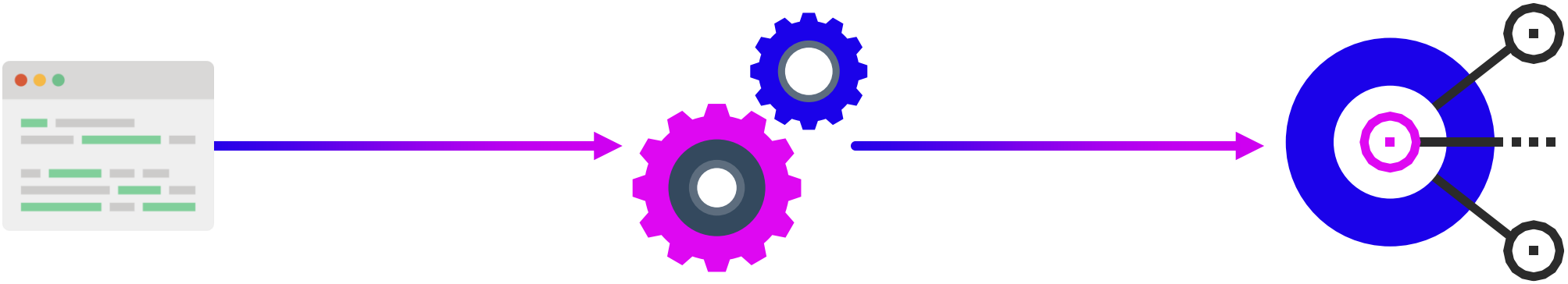
Service Worker Scope



Scope can be **restricted** but **not widened**

```
// Scope option can further limit which requests got to Service Worker
// Gets all requests starting with '/path/subpath/'
navigator.serviceWorker.register('/path/sw.js', { scope: '/path/subpath/' });
```

Service Worker Request Interception

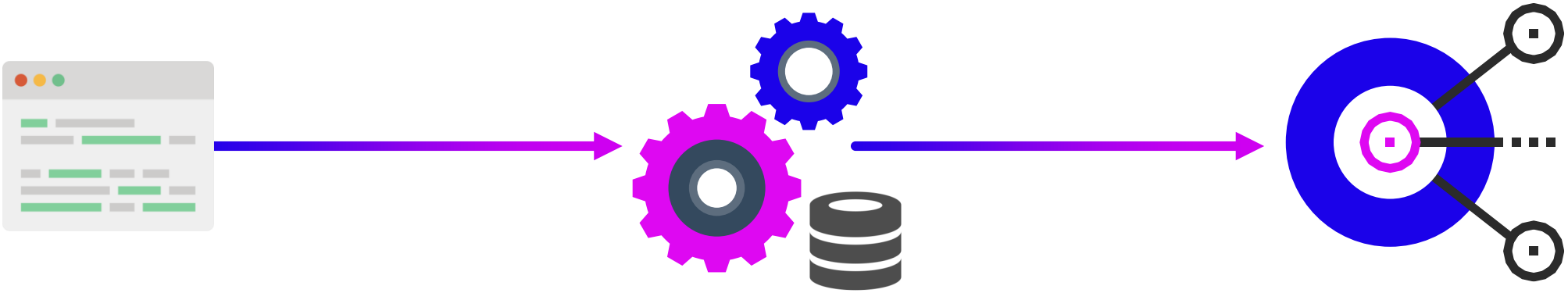


```
self.addEventListener('fetch', (event) => {
  // React to fetch event
  const { url } = event.request;
  event.respondWith((async () => {
    const request = new Request(url.replace('.com', '.de'))
    const response = await fetch(request);
    const text = await response.text();
    const newText = text.replace('Goethe', 'Schiller');
    return new Response(newText, { status: 200 });
  })());
});
```

There is so much you can do:

- **Rewrite** Requests
- **Change** Responses
- **Concat** Responses
- **Cache** Responses
- **Serve** Cached Data
- …

Service Worker Persistence



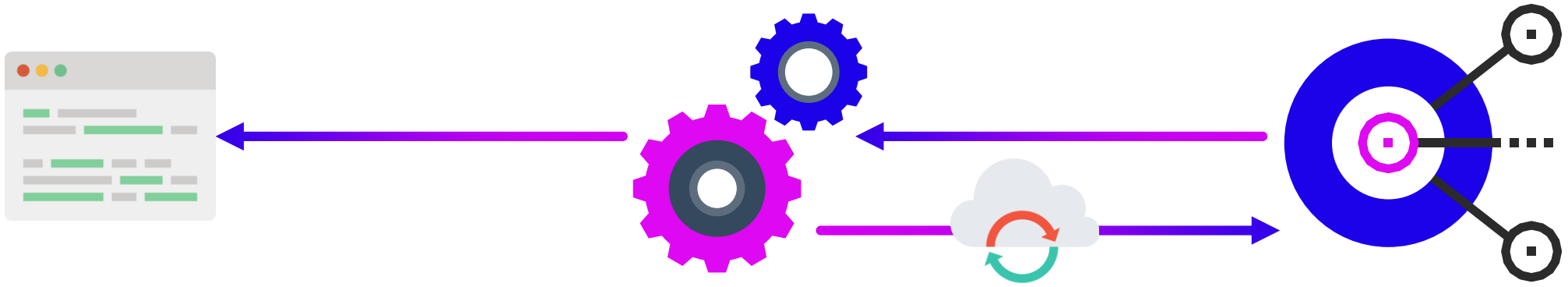
IndexedDB

an actual database in the browser

- Stores Data **Persistently**
- Stores **Structured** Data
- Supports **Range Queries**
- **Browser Support** 94%

Experimental

Service Worker Background Sync



One-off Sync

- executed when user is **online**
- **retried** when failed (exponential backoff)

Use Cases

- Save **file** when online again
- Send **email** when online again

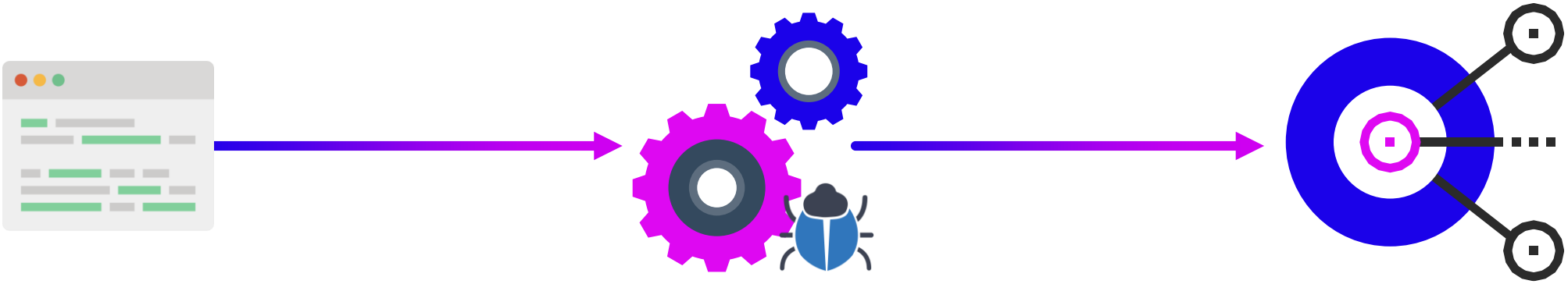
Periodic Sync

- executed when online, according to **period options**

Use Cases

- Load updates to **social media time-line** when browser closed

Service Worker Debugging



Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits

Application
- Manifest
- Service Workers
- Clear storage

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

Cache
- Cache Storage
- Application Cache

Service Workers

☐ Offline ☐ Update on reload ☐ Bypass for network ☐ Show all

www.baqend.com Update Unregister

Source sw.7dbf553e.js

Received 23.1.2018, 15:38:40

Status ● #823 activated and is running stop

Clients https://www.baqend.com/ focus

Push `Test push message from DevTools.` Push

Sync `test-tag-from-devtools` Sync

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits

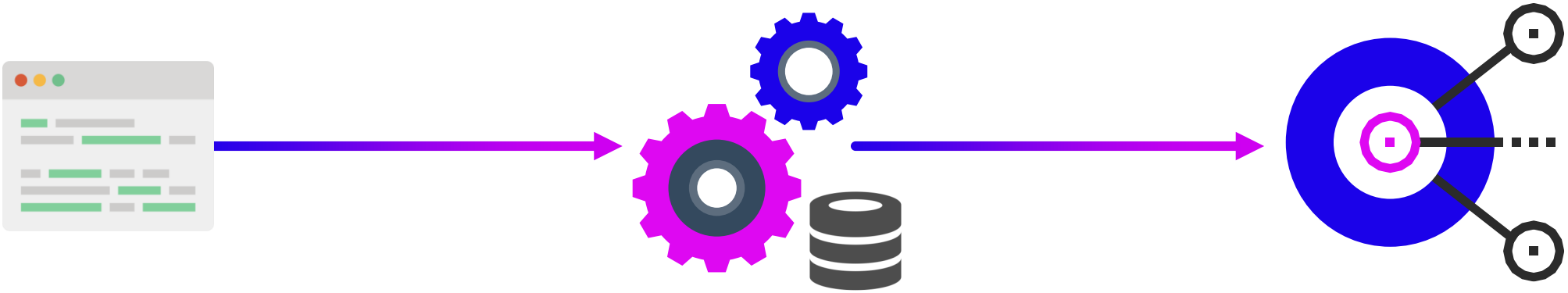
Network | Filesystem » | sw.7dbf553e.js:formatted

- top
  - www.baqend.com
    - css
    - fonts
    - img
    - js
    - (index)
    - manifest.293e2c44.json
  - Wappalyzer
  - connect.facebook.net
  - embed.tawk.to
  - fonts.googleapis.com
  - fonts.gstatic.com
  - www.google-analytics.com
  - FraD6fj-1516730773916 (abou
  - XkVhsaD-1516730773915 (ab
  - ZNEaMt3-1516730773916 (ab
  - v3
- sw.7dbf553e.js

```
14      }
15      var a = {};
16      return t.m = e,
17      t.c = a,
18      t.d = function(e, a, s) {
19          t.o(e, a) || Object.defineProperty(e, a, {
20              configurable: false,
21              enumerable: true,
22              get: s
23          })
24      }
25      ,
26      t.n = function(e) {
27          var a = e && e.__esModule ? function() {
28              return e['default']
29          }
30          : function() {
31              return e
32          }
33          ;
34          return t.d(a, 'a', a),
35          a
36      }
37      ,
38      t.o = function(e, t) {
39          return Object.prototype.hasOwnProperty.call(e, t)
40      }
41      ,
42      t.p = '',
43      t(t.s = 13)
44  }
45  )([function(e, t, a) {
46      'use strict';
47      t.b = function(e) {
48
```

Line 32, Column 9

Service Worker Caching



Cache Storage

Stores Request/Response pairs

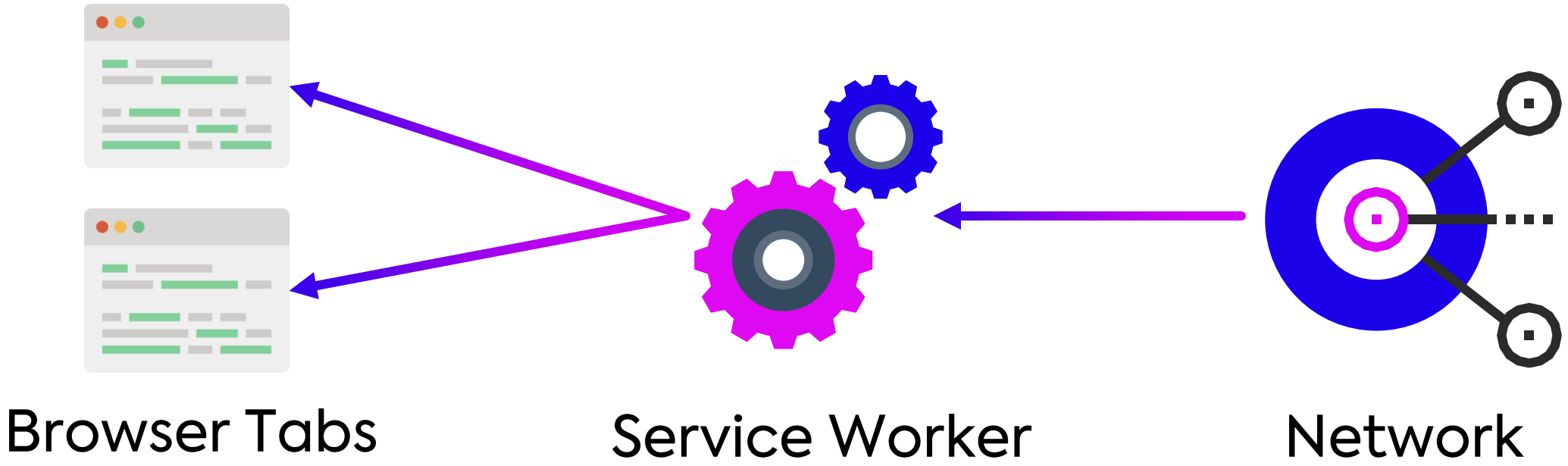
Cache Storage

- **Programmatically** managed
- **Persistent** and non-expiring

- Supports only **HTTP**
- Only caches **GET** requests (no HEAD)

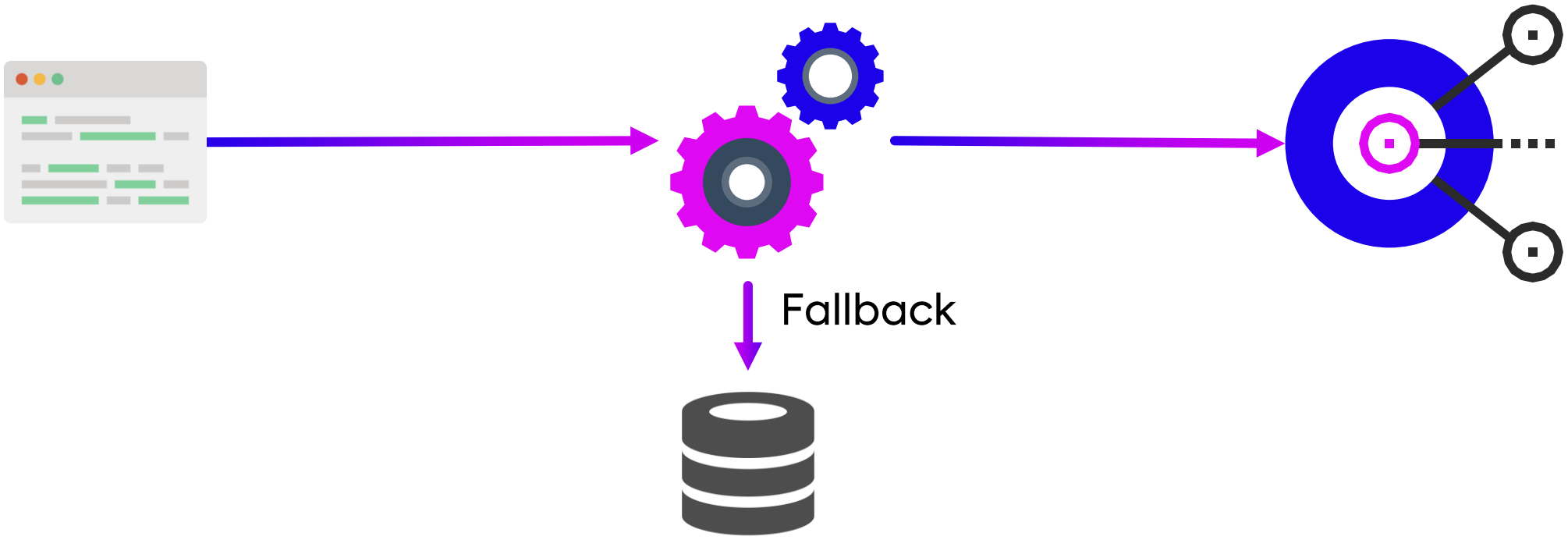
Break => Demo

What Are Service Workers



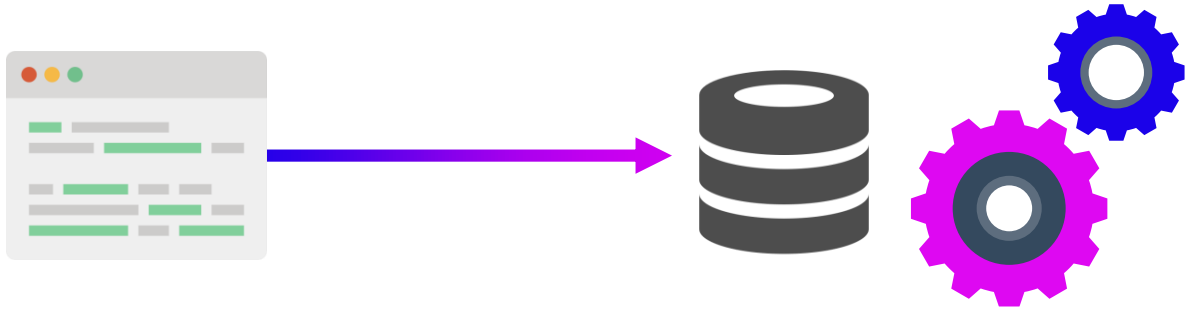
Programmable **Network Proxy**, running as a separate **Background Process**, without any **DOM Access**.

Caching Strategies Network, Cache Fallback



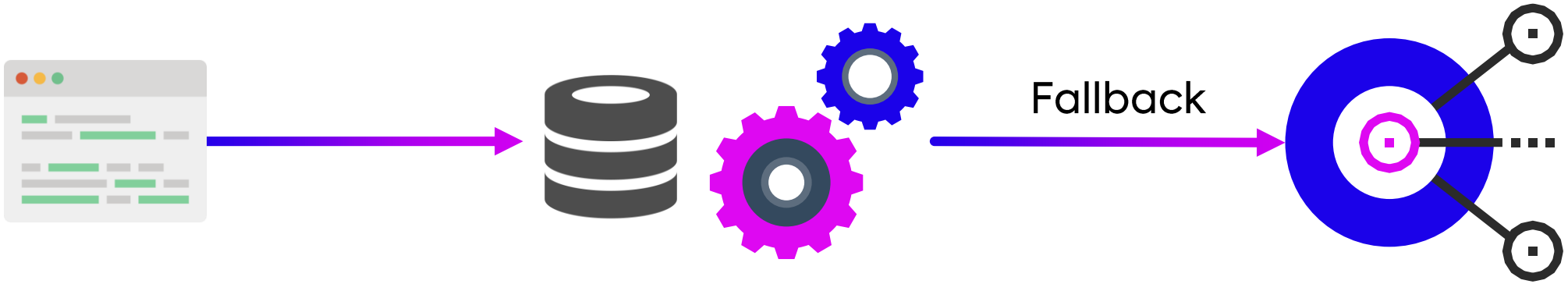
Gets requests from network, the cache acts as fallback (offline mode).

Caching Strategies Cache Only



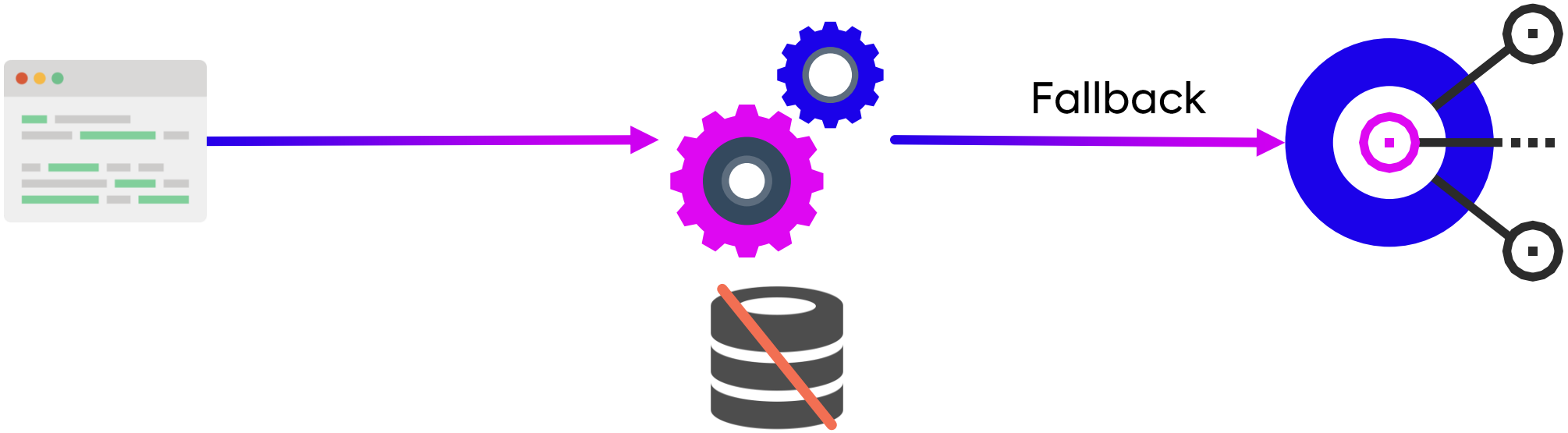
Gets all requests from cache or fails.

Caching Strategies Cache, Network Fallback



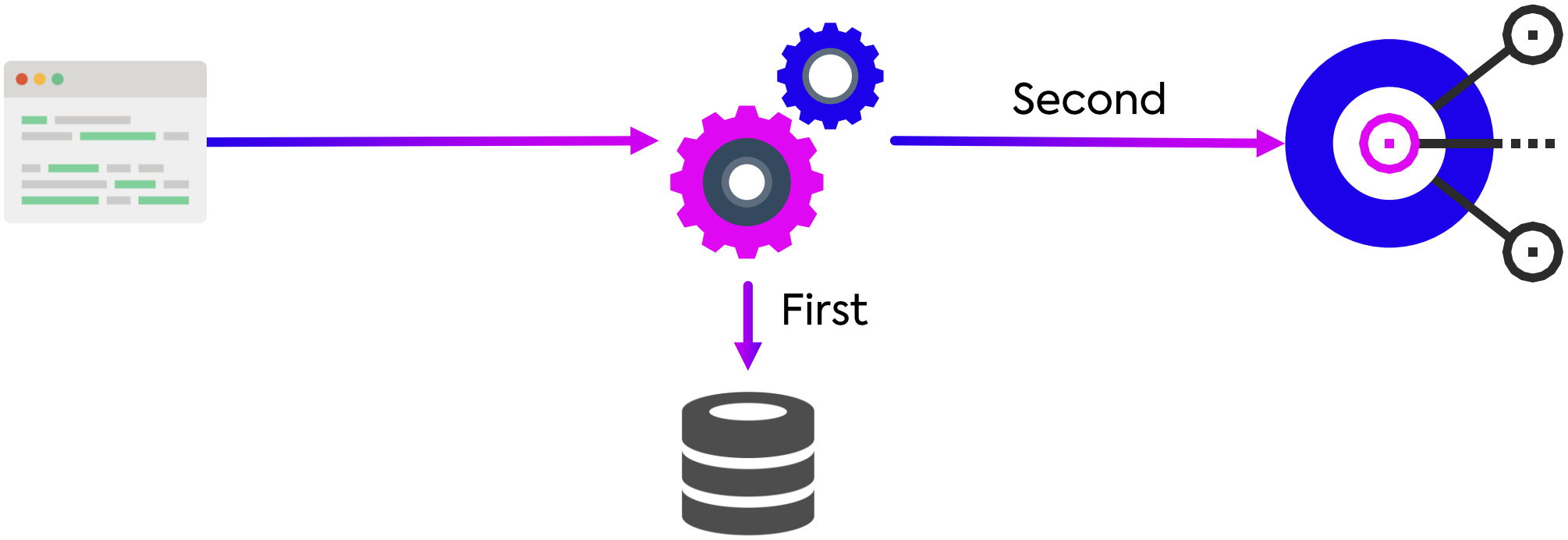
Gets requests from cache & uses network as fallback.

Caching Strategies Network Only



Gets requests from network only.

Caching Strategies Cache, Then Network

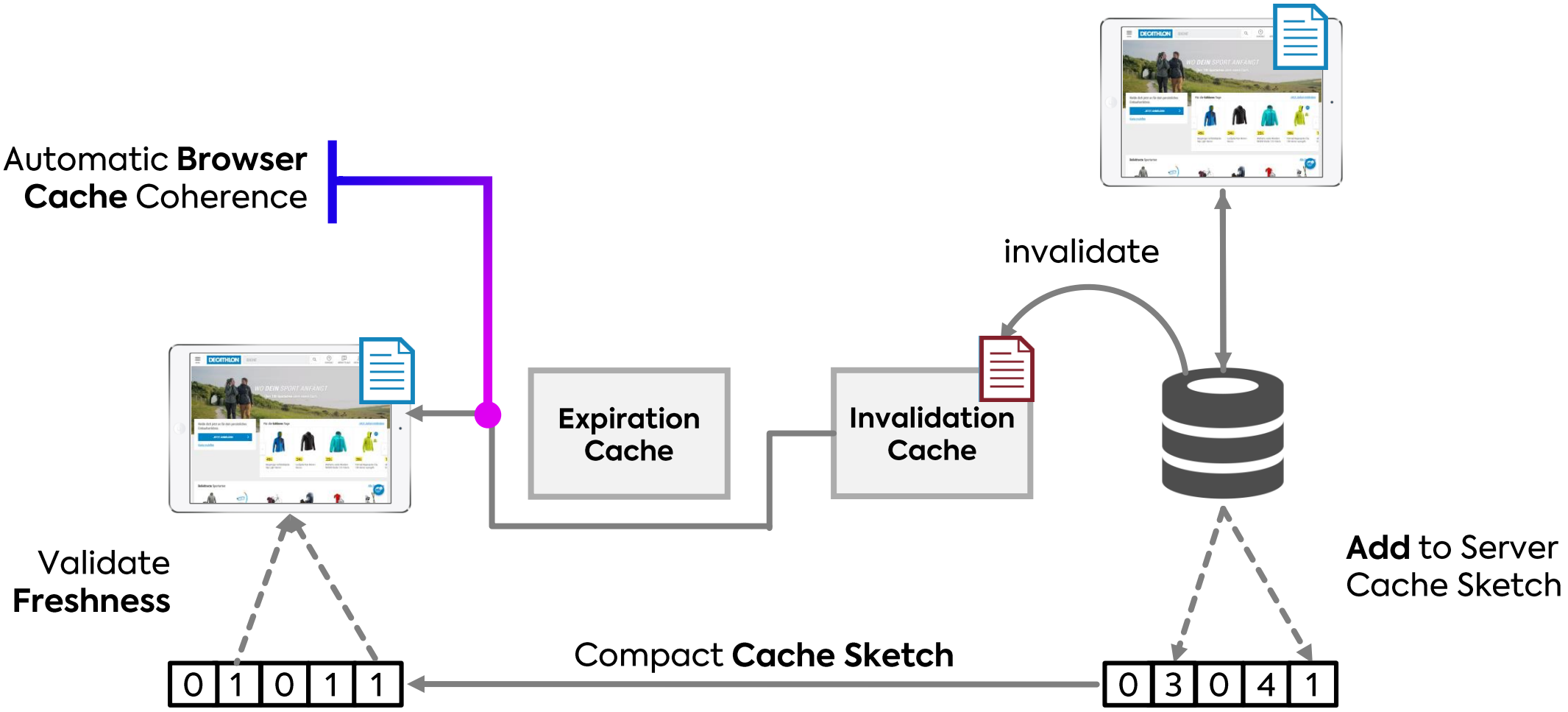


Gets requests from cache first and from network in background.

Major Challenge

Cache Coherence
+
Personalization

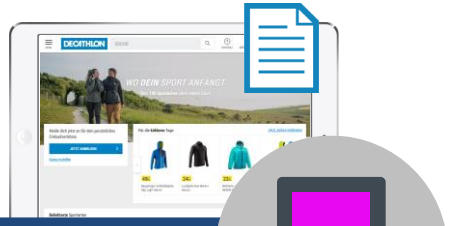
How To Solve Cache Coherence

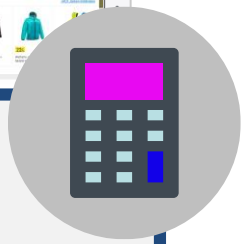


[GSW+15, GSW+17]

How To Solve Cache Coherence

Automatic **Browser Cache** Coher





Example

1500 entries & **0.0019%** false positives → **7 KB** size

↓

>25% cache hits on HTML in production

Validate **Freshness**

dd to Server Cache Sketch

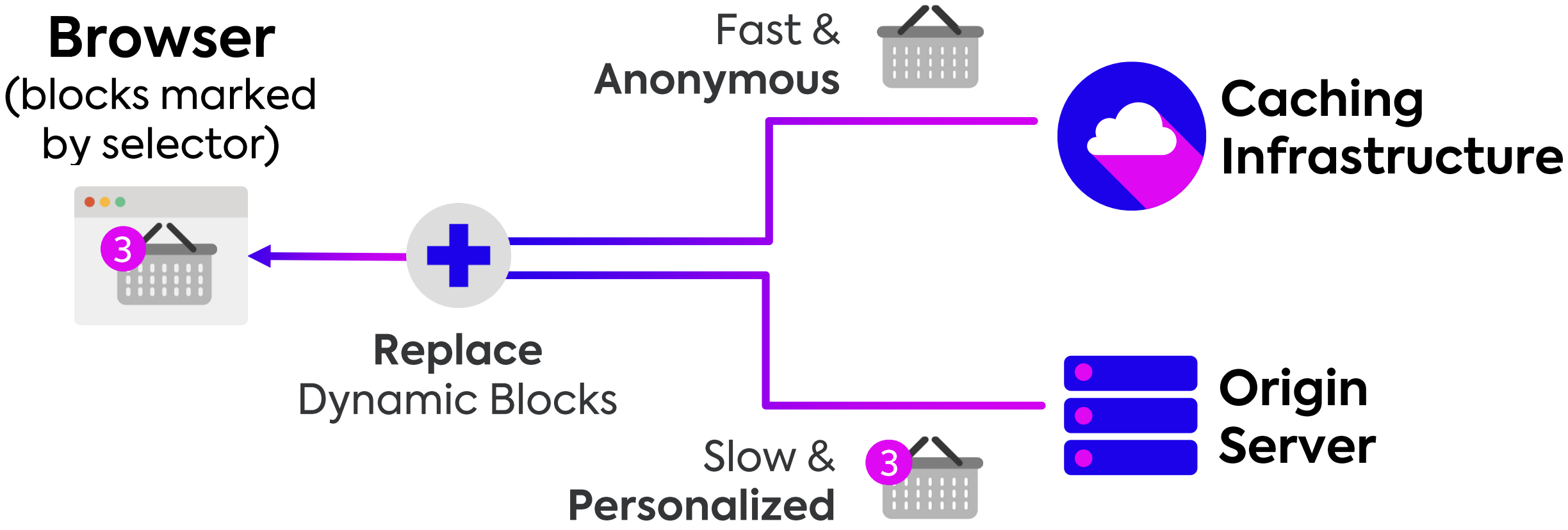
Compact **Cache Sketch**

0	1	0	1	1

0	3	0	4	1

[GSW+15, GSW+17]

How to Solve Personalized Content

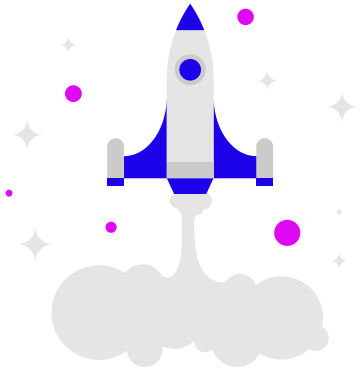


We Are Baqend

We bring performance research to practice.

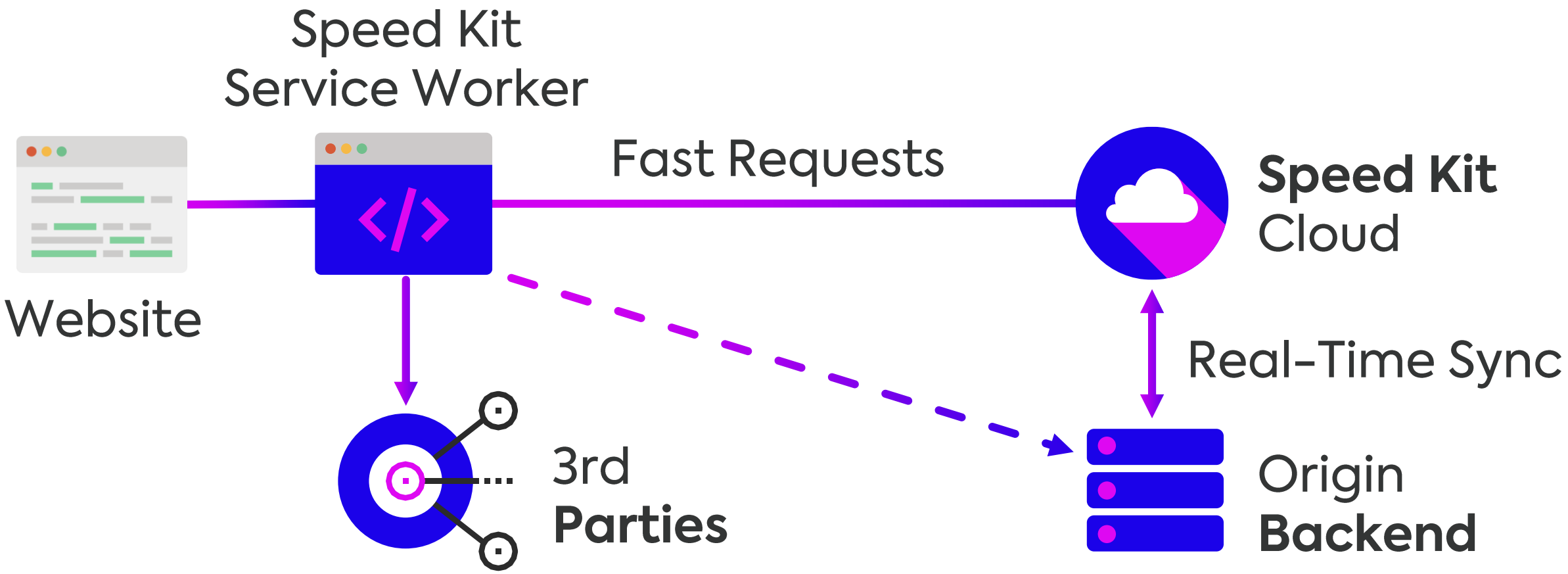
30+ man-years of **web performance research**

Novel technology for caching **dynamic data**

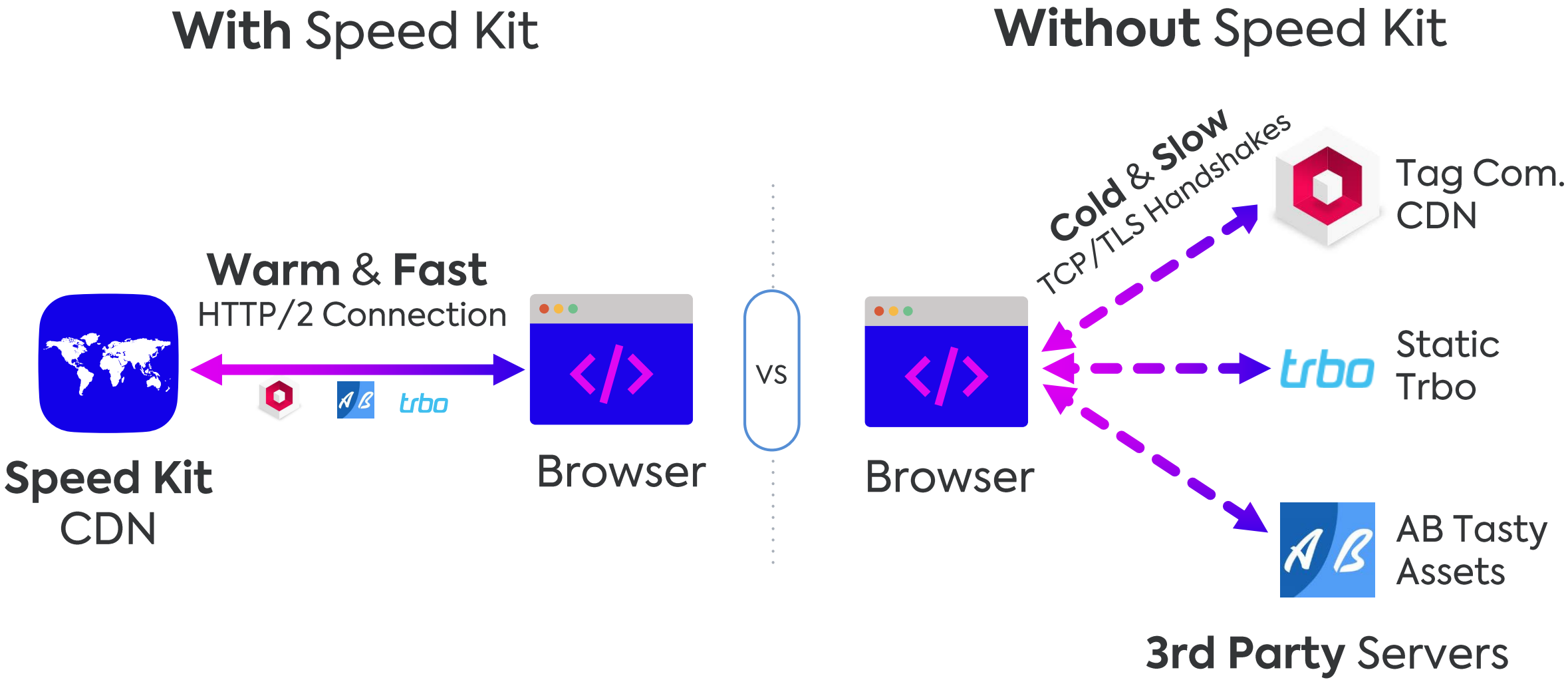


Speed Kit – SaaS for e-commerce speed

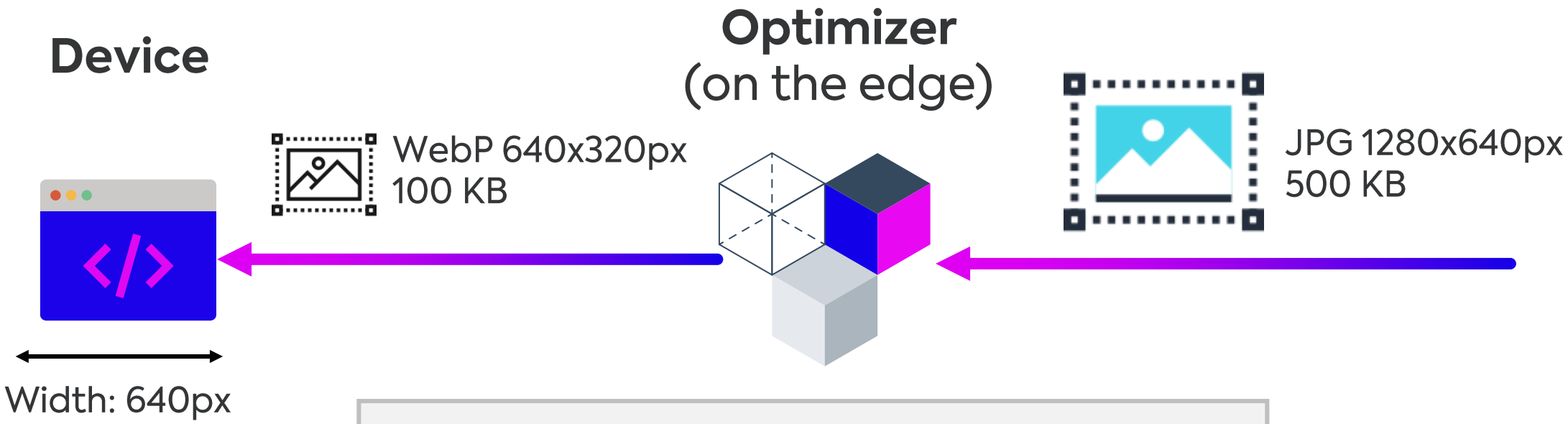
How Speed Kit Works



Accelerated 3rd Parties



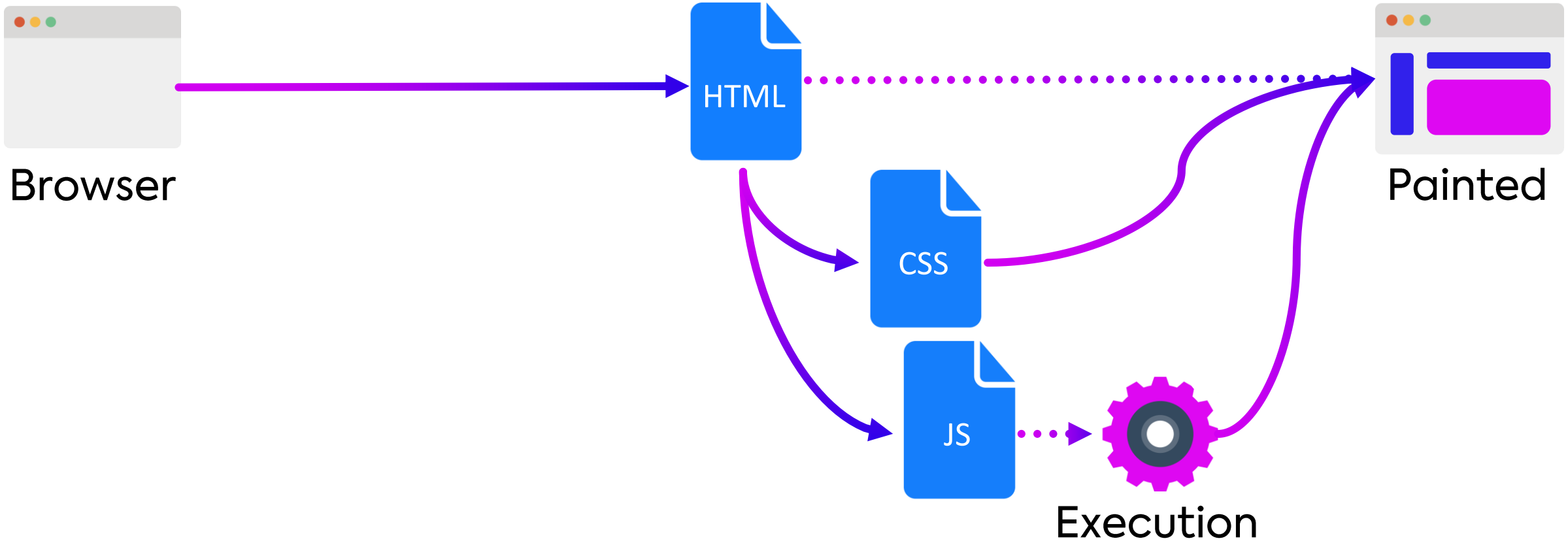
Automatic Image Optimization



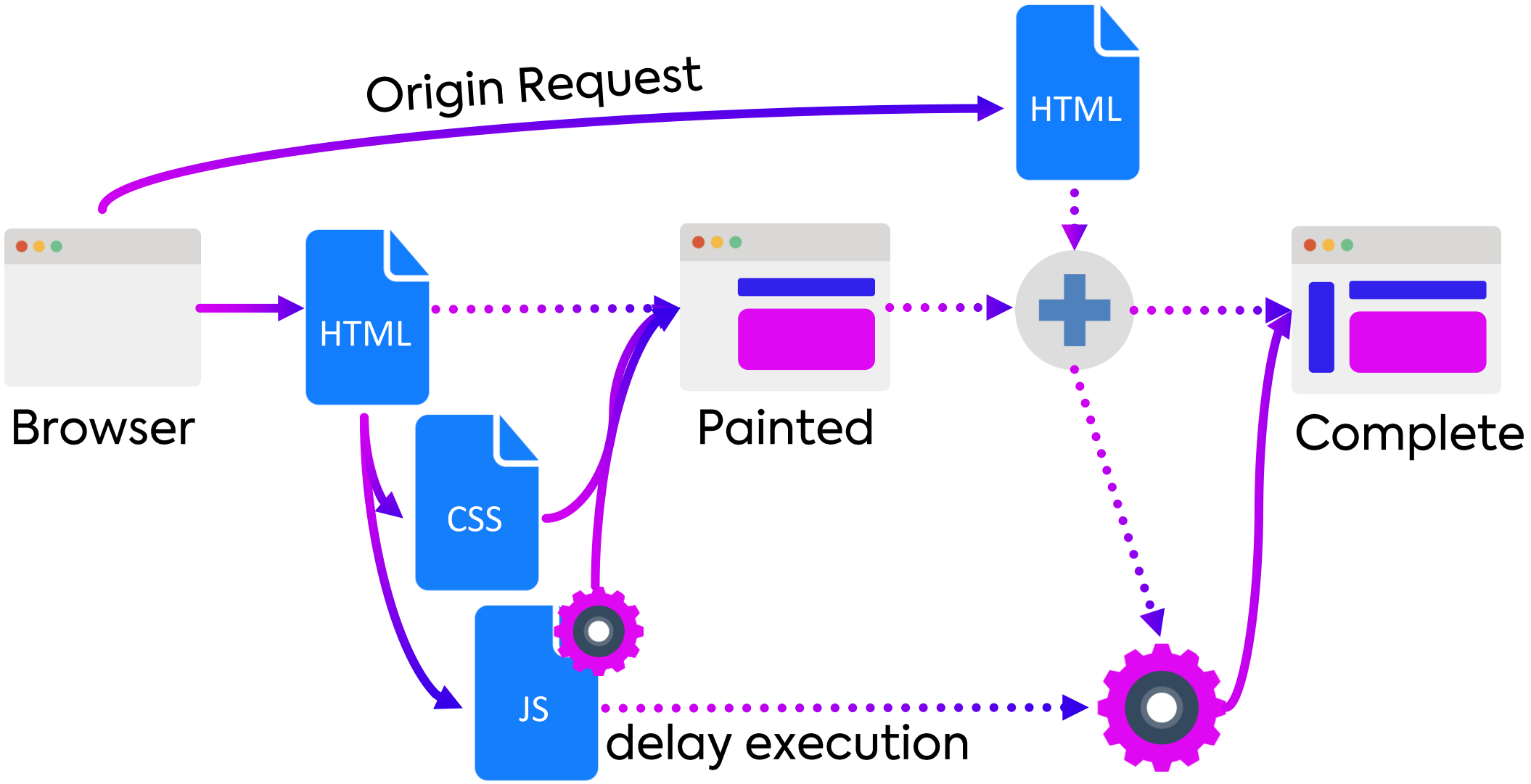
- Images transcoded to **WebP**
- Rescaled to match **Screen Size**
- JPG and PNG **Recompression**

Dynamic Blocks in Detail

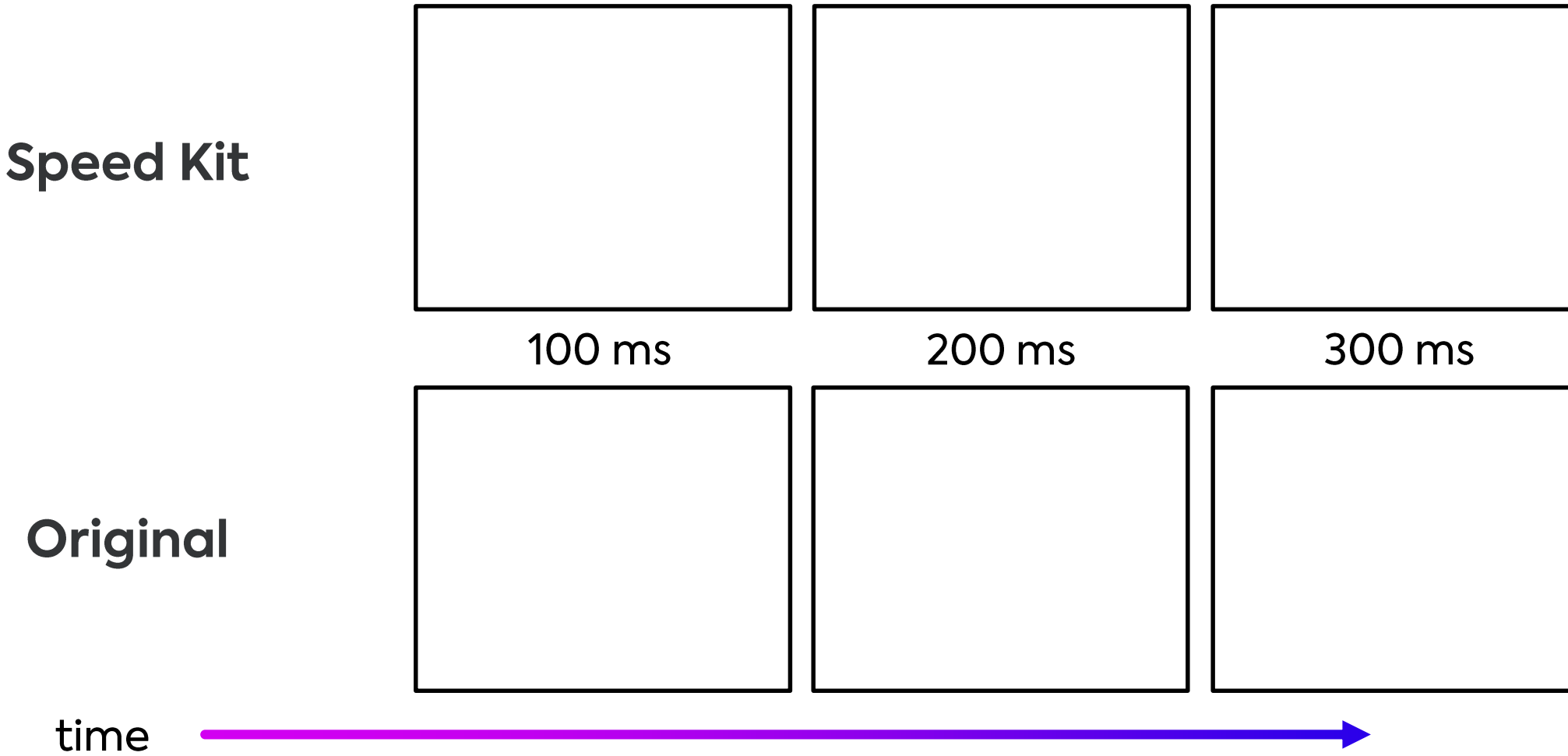
How the browser builds a page



Dynamic Blocks in Detail



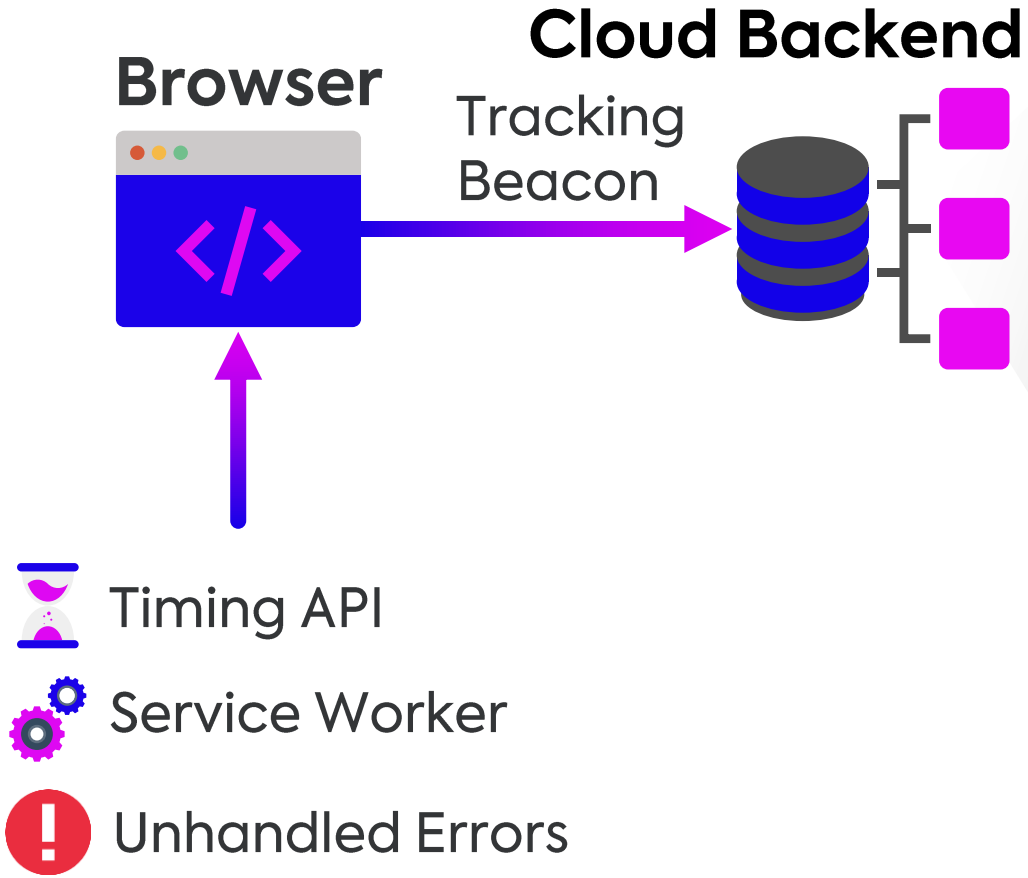
How Dynamic Blocks Help Rendering



Demo

What do we gain?

Real-User Monitoring



Performance
- Time-to-First-Byte
- First (Contentful) Paint
- DOM Timer
- First Input Delay

User Engagement
- Session Length
- Time on Site
- First User Interaction
- Bounce Rate

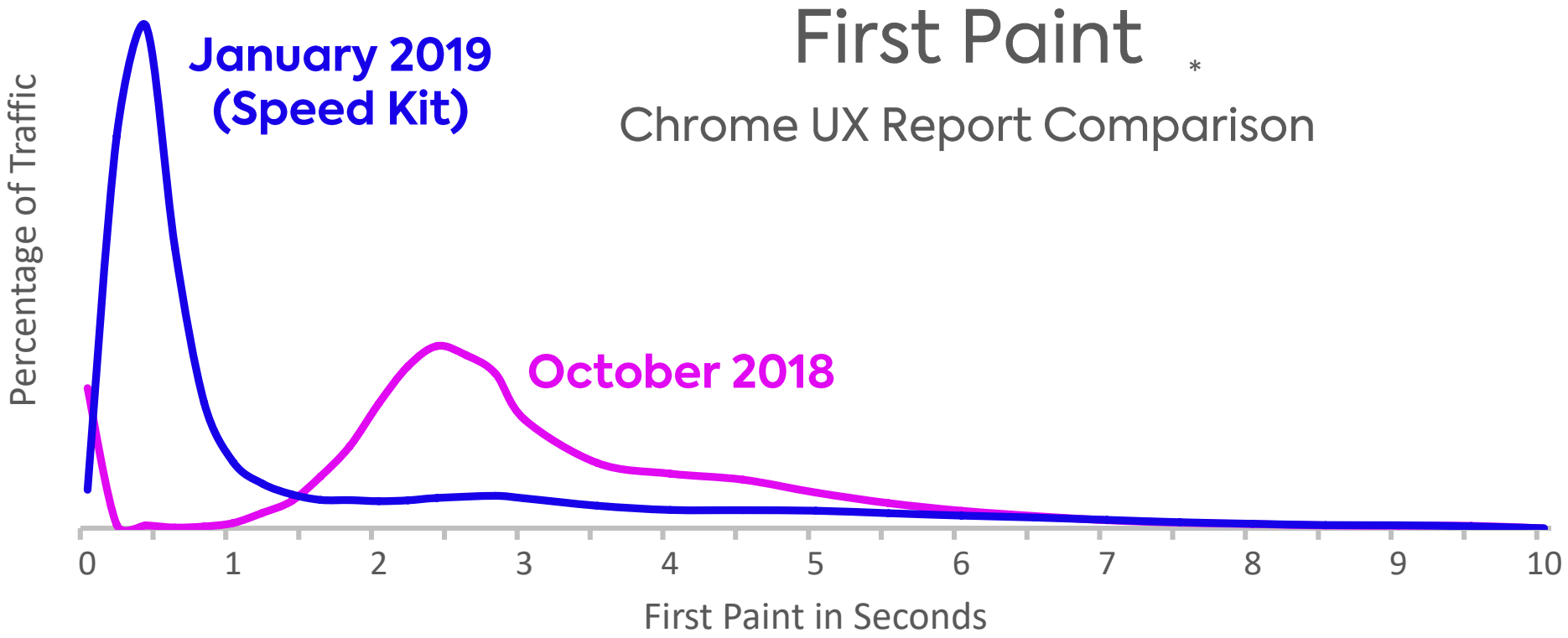
Business KPIs
- Cart Size
- Transactions
- Conversion Rate
- Revenue

Q&A Metadata
- Page Views & Sessions
- Browser Distribution
- JavaScript Errors
- Caching Insights

Performance Uplift

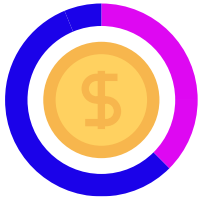
*Median performance compared between the two A/B test split groups on the Appelrath website

CrUX Data Analysis



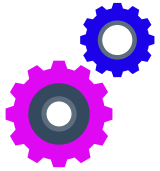
*comparing appelrath.com Chrome UX report data from October 2018 with January 2019

Appelrath: Business Impact



User-based Conversion Rate

+ >10%

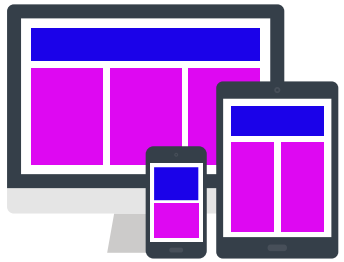


Overall Time for Speed Kit Setup

2 Days

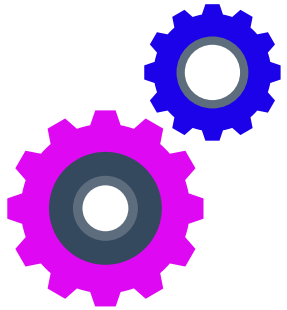
Wrap Up

PWAs



Super cool alternative to native apps

Service Workers



Powerful programmable network proxy

Use Case

Speed Kit: Smart CDN through Service Workers

Learn more about this topic:

https://blog.baqend.com/

Edit

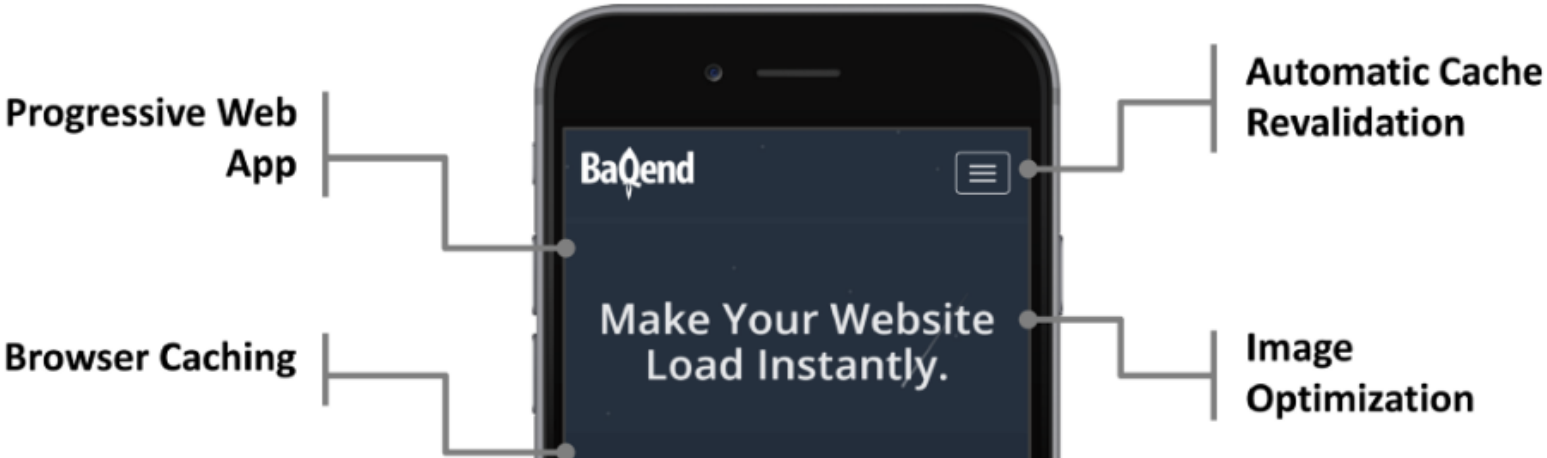
Applause from you, Konstantin Möllers, and 12 others

Wolfram Wingerath
Distributed systems engineer at Baqend, a serverless backend for faster websites. Background in database research & developing Baqend's real-time query engine.
Apr 29 · 34 min read

Rethinking Web Performance with Service Workers

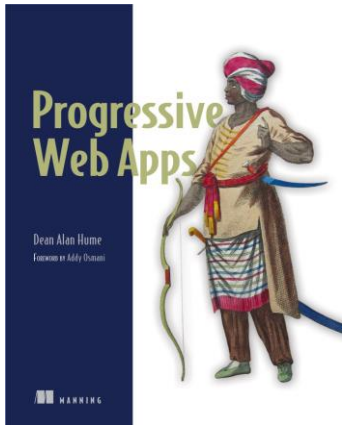
30 Man-Years of Research in a 30-Minute Read

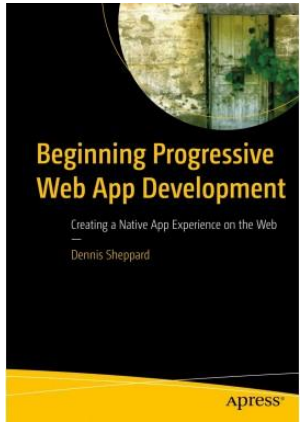
This article surveys the current state of the art in page speed optimization. It contains the gist of more than 30 man-years of research that went into Speed Kit, an easy-to-use web performance plugin to accelerate any website.



Learn More About Service Workers

Recommended Books





Blogs

Baqend Blog
On Building a Faster Web

https://blog.baqend.com/

Ilya Grigorik
Internet plumber

https://www.igvita.com/

Jake Archibald wrote...
I discovered a browser bug

https://jakearchibald.com/

Guides & Tutorials

Progressive Web Apps

A new way to deliver amazing user experiences on the web.

https://developers.google.com/web/progressive-web-apps/

Progressive web apps

Jump to: PWA advantages Core PWA guides Technology guides

https://developer.mozilla.org/en-US/docs/Web/Apps/Progressive

Thank You!

Questions?

Erik Witt, Speed Kit Lead ew@baqend.com

Baqend