



The Technology Behind Progressive Web Apps Service Workers in Detail

Felix Gessert

fg@baqend.com

May 2018 – New Technology Meetup



[@baqendcom](https://twitter.com/baqendcom)



What we are going to cover.

PWA



Core Features
Building Blocks
Implementation

Service Workers



Lifecycle
Network Interception
Caching Strategies

Speed Kit



Cache Coherence
Performance-
Measures

Why do(n't) we love native apps?

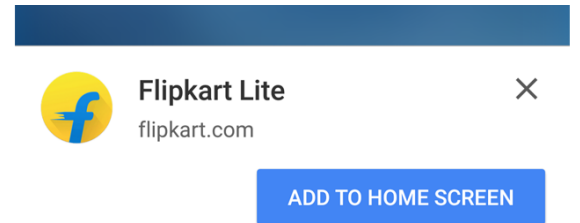
Progressive Web Apps

seek to combine the great from **native** and **web apps**

A person wearing a white long-sleeved shirt is holding a smartphone with both hands. The person's left wrist is visible, wearing a silver-toned watch with a white face and a metal link bracelet. The watch face has 'ck' and 'Calvin Klein' printed on it. The background is a dark, textured surface. The text 'What are Progressive Web Apps?' is overlaid in the center in a large, white, sans-serif font.

What are Progressive Web Apps?

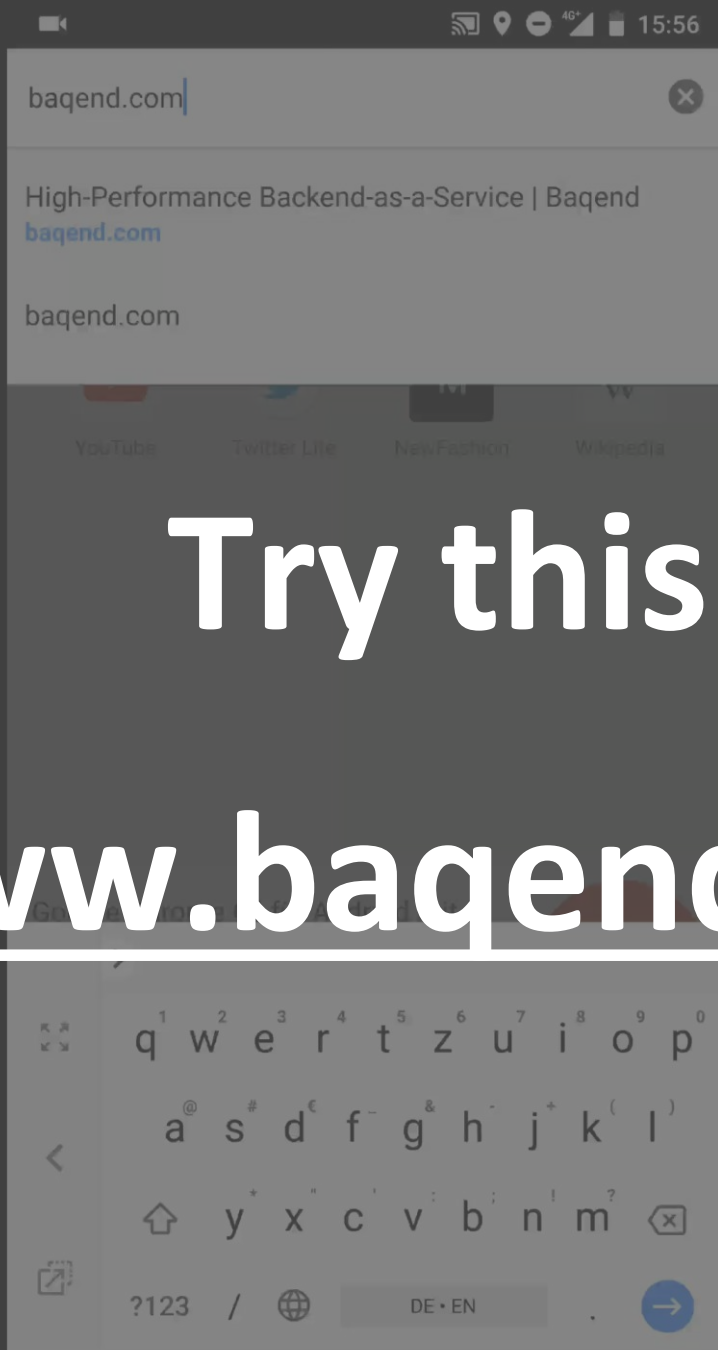
Progressive Web Apps (**PWAs**)



Fast **Loads**
through Caching

Offline Mode
(Synchronization)

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



Try this:

www.baqend.com

Building Blocks of PWAs

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



1. **Manifest**

Progressively enhance
when supported



2. **Service Worker**

Implementing PWAs

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

Progressively enhance
when supported

1. **Manifest** declares Add-to-Homescreen:

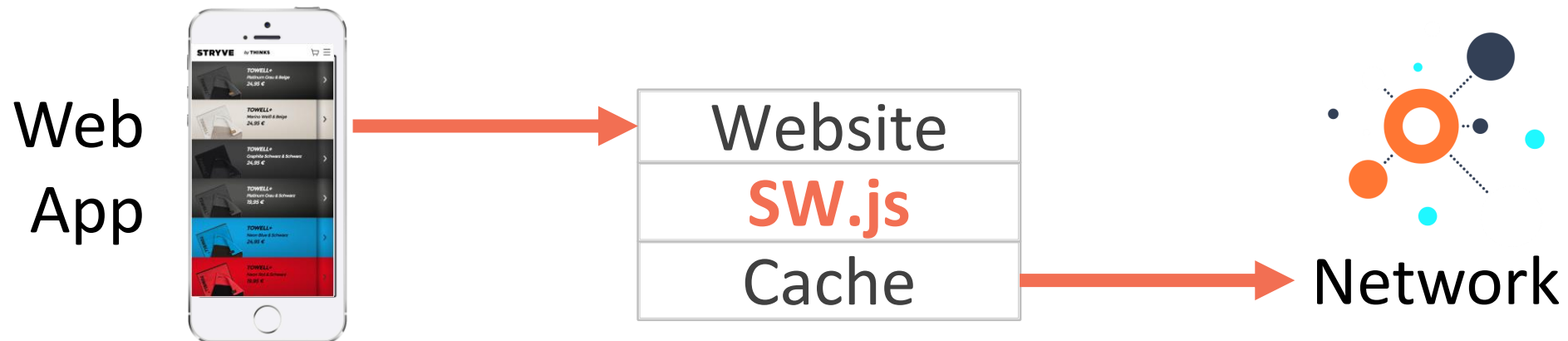
```
<link rel="manifest" href="/manifest.json">
{
  "short_name": "Codetalks PWA",
  "icons": [
    {"src": "icon-1x.png", "type": "image/png", "sizes": "48x48"}],
  "start_url": "index.html?launcher=true"
}
```


Implementing PWAs

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

Gracefully degrade when
not supported

2. **Service Workers** for caching & offline mode:

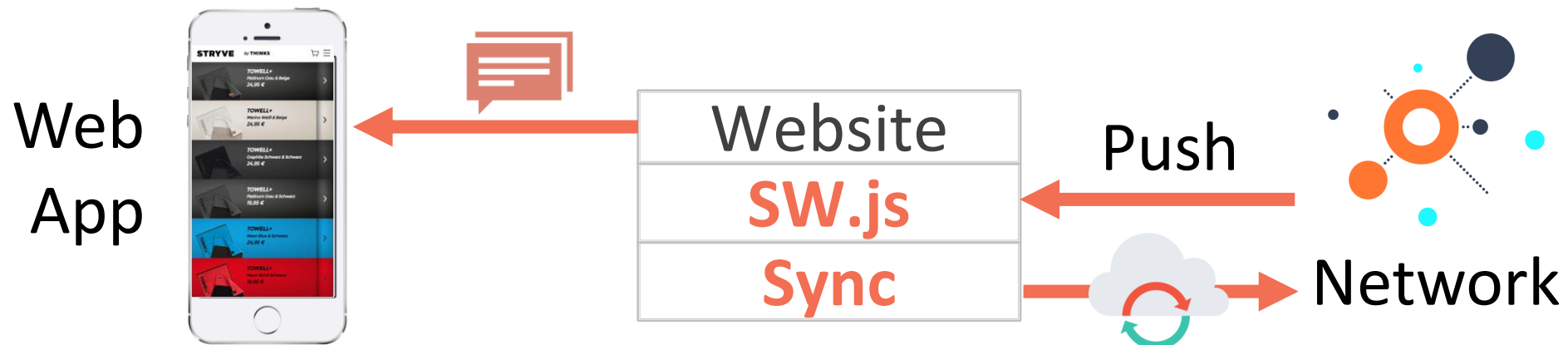


Implementing PWAs

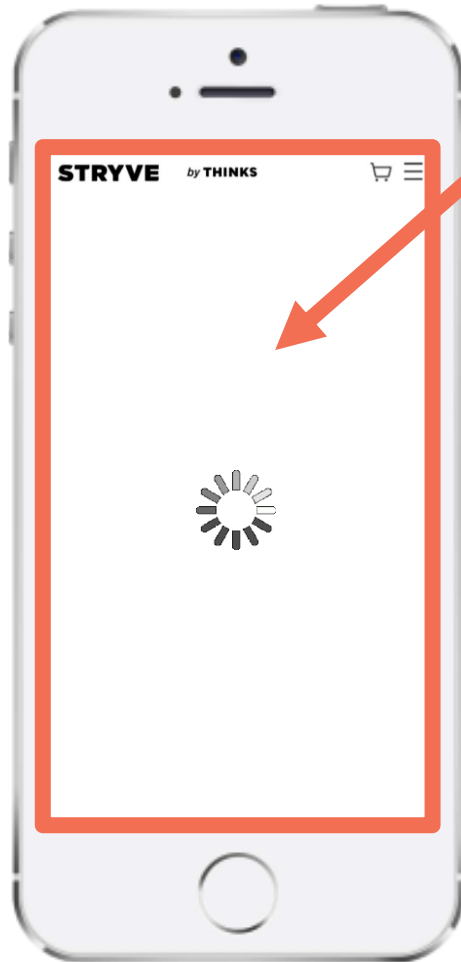
PWAs are **best practices**
and **open web standards**

Progressively enhance the
user experience

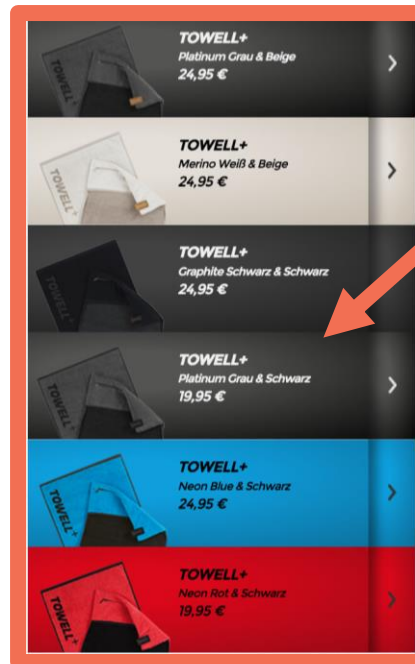
3. Add **Web Push** and **Background Sync**:



Typical Architecture: App Shell Model



App Shell: HTML, JS, CSS, images with app logic & layout

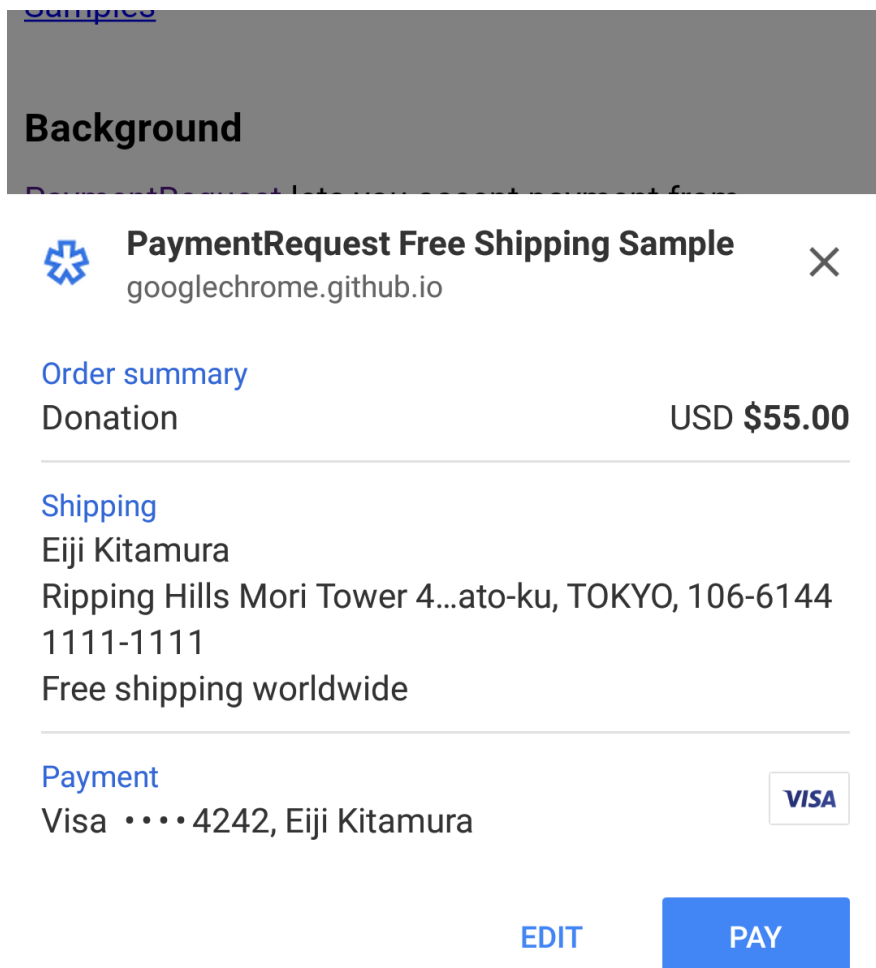


Content: Fetched on demand & may change more often

A stylized, blocky robot head is visible in the background, rendered in a dark, muted color. It has a square face with two circular eyes and a small, rectangular mouth. The robot's head is positioned on the right side of the frame, with its body extending downwards and to the left. The overall aesthetic is minimalist and modern.

What is the future of Progressive Web Apps?

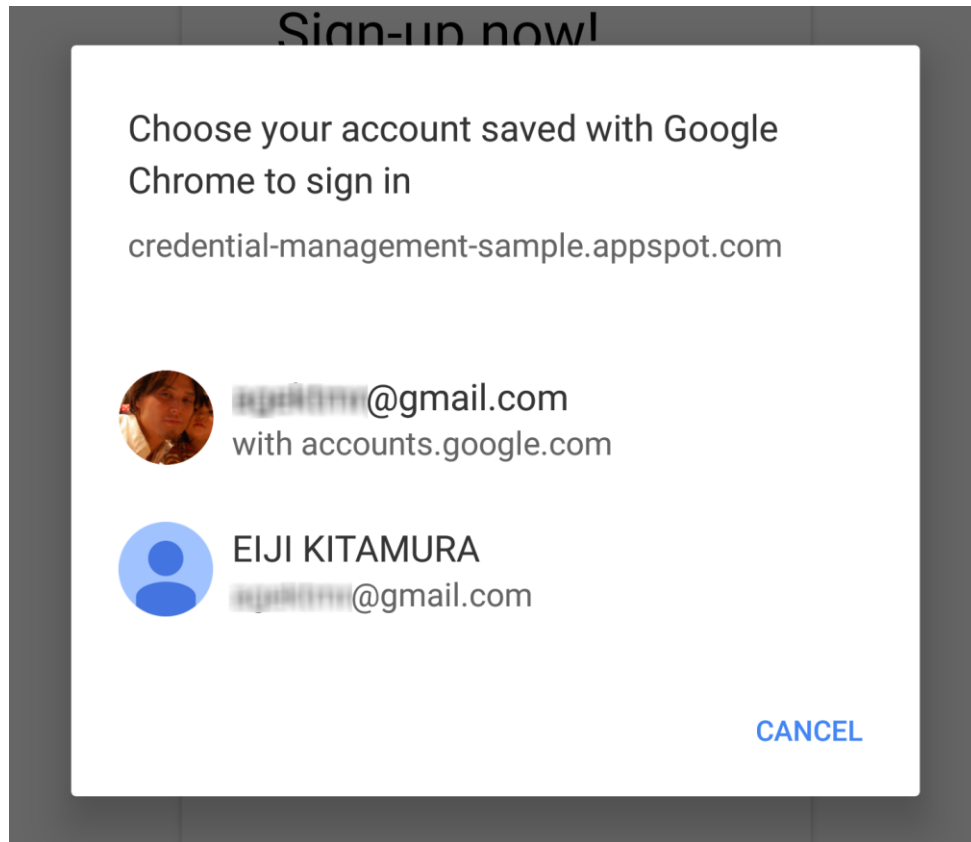
The Future of PWAs is bright.



Payment Request API

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

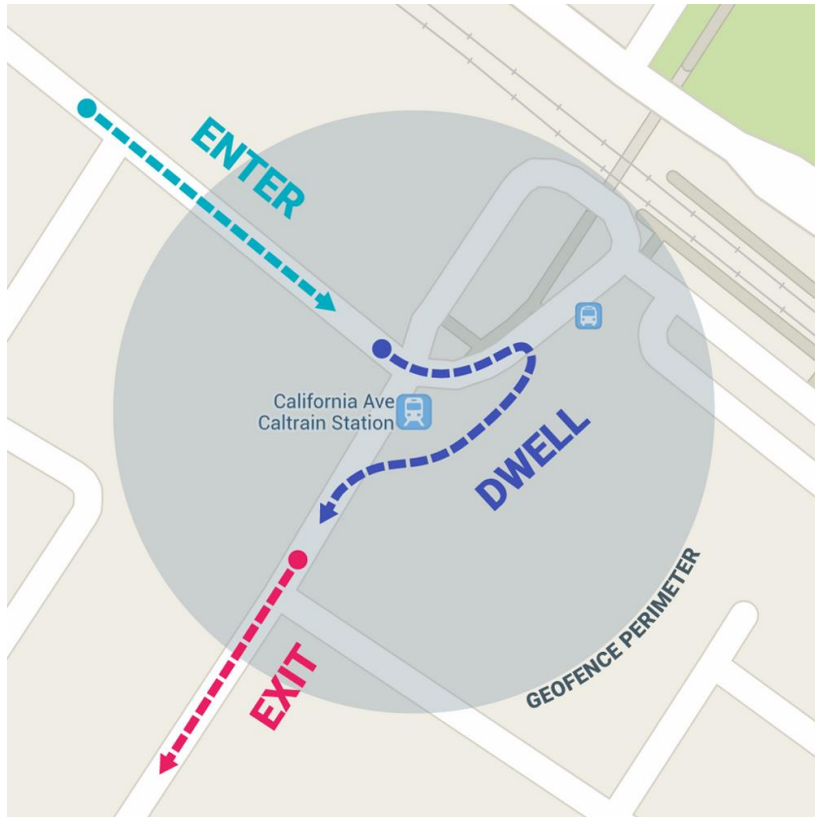
The Future of PWAs is bright.



Credentials Management API

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

The Future of PWAs is bright.



Geofencing

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

The Future of PWAs is bright.



Web Speech API

Native Speech Recognition in the Browser:

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

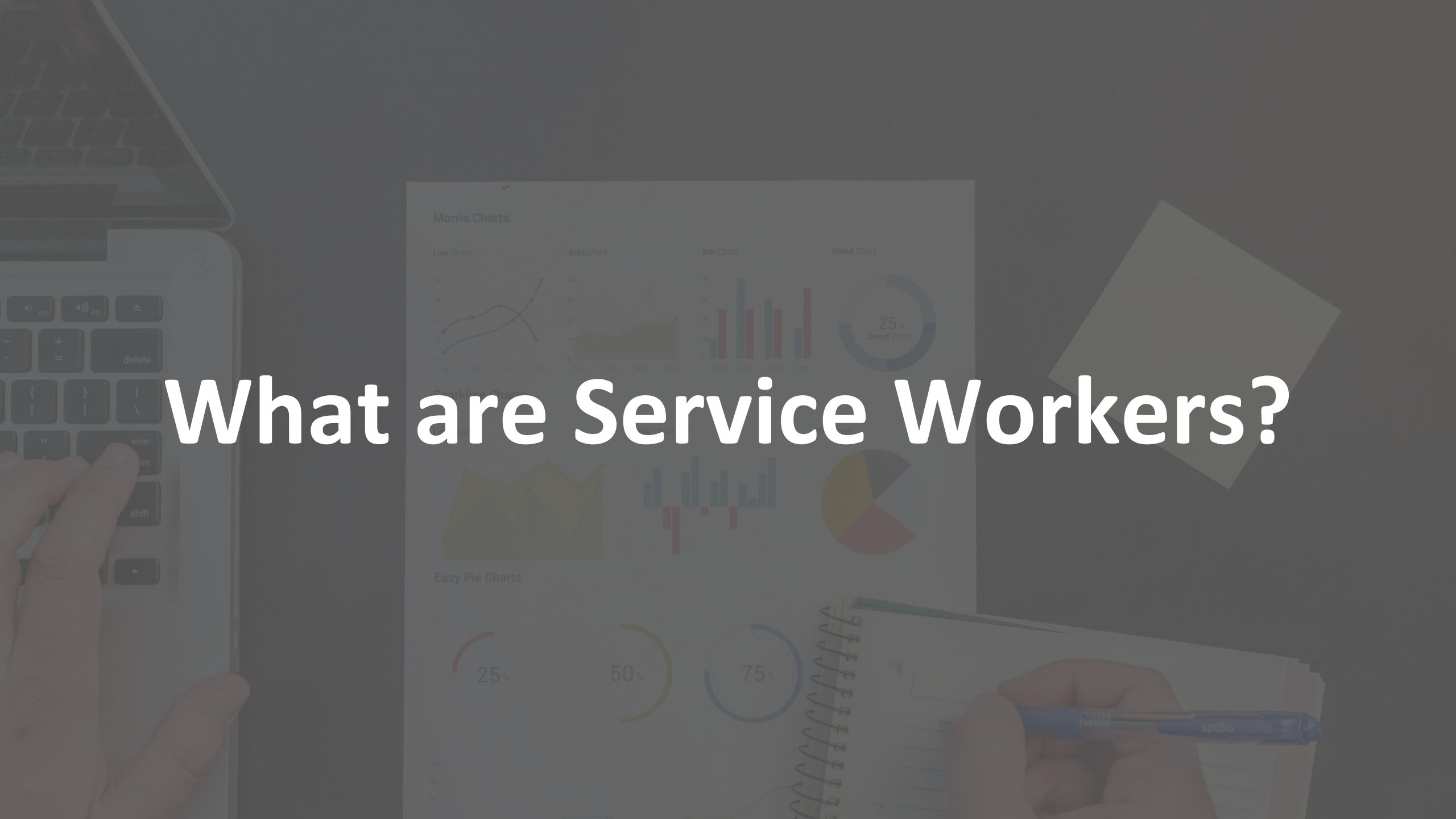

The Future of **PWAs** is bright.



Web Share API

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

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

Browser Support for Service Workers

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			64		10.3				
	16	58	65	11	11.2				4
11	17	59	66	11.1	11.3	all	66	11.8	6.2
	18	60	67	TP					
		61	68						
			69						

Supported by ~90% of browsers.

Requires TLS Encryption.

Late, but all in: Microsoft

Publish PWAs to
Microsoft Store



**PWA
BUILDER**

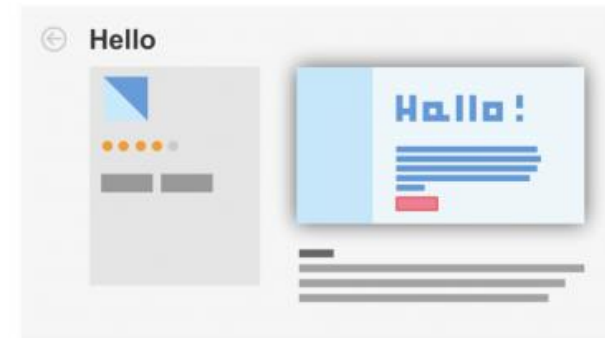
or



Bing Crawl
PWAs



Convert to
AppX



Microsoft Store



How are **Service Workers** registered?



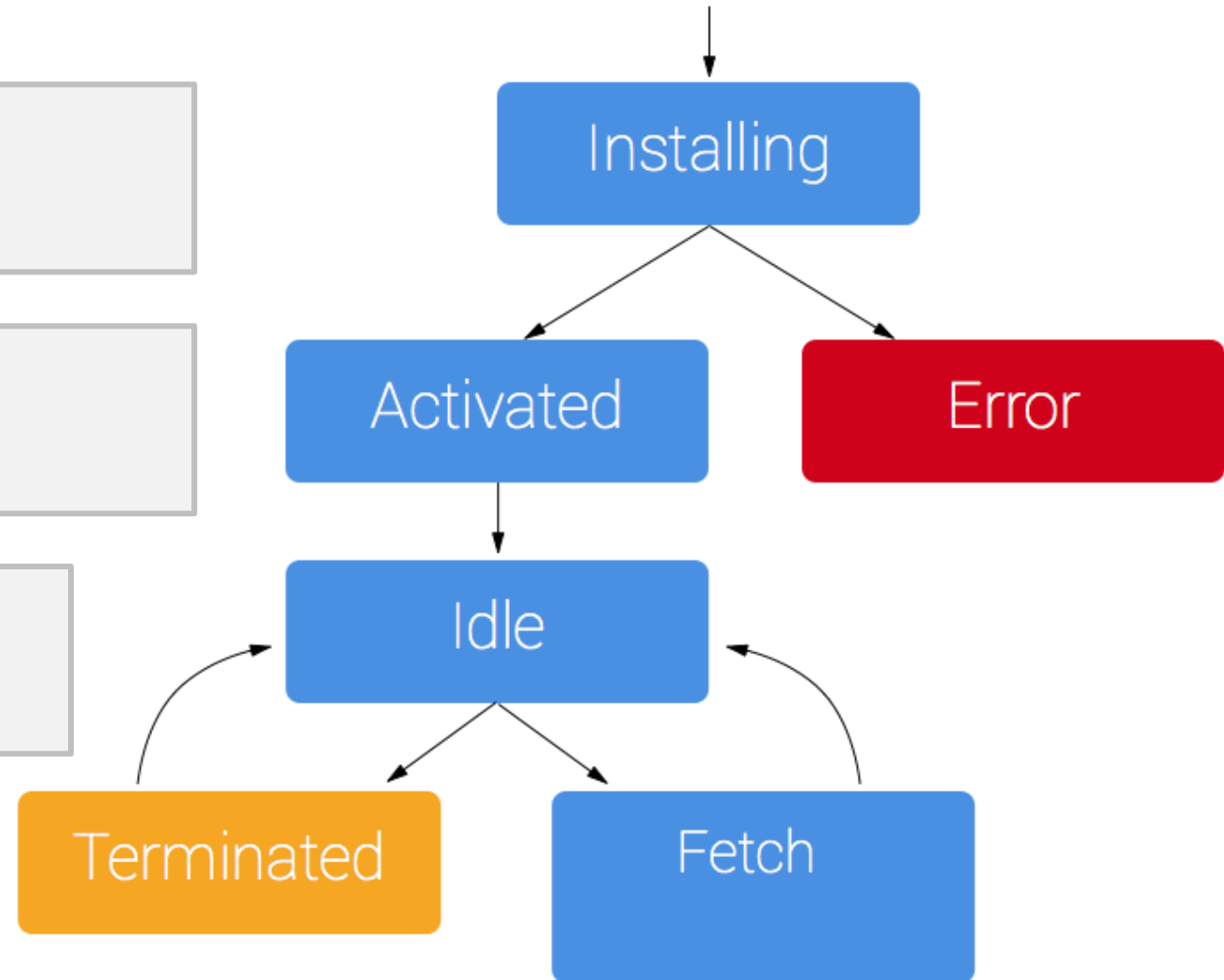
```
<script>  
  navigator.serviceWorker.register('/sw.js');  
</script>
```


How does the Lifecycle look like?

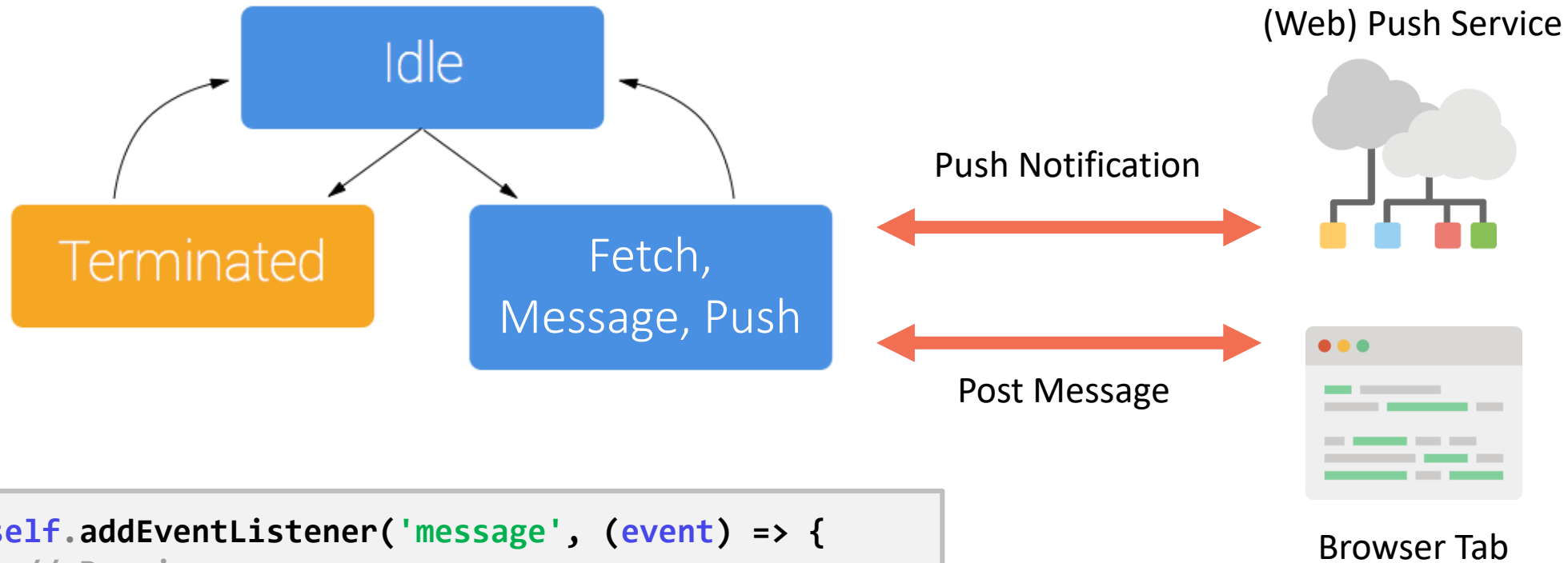
```
self.addEventListener('install', (event) => {  
  // Perform install steps  
});
```

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

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



How to Communicate with Service Workers?



```
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  
});
```

Intercepting Network Requests



```
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** Request
- **Change** Response
- **Concat** Responses
- **Cache** Responses
- **Serve** Cached Data
- ...

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 **Persistence**

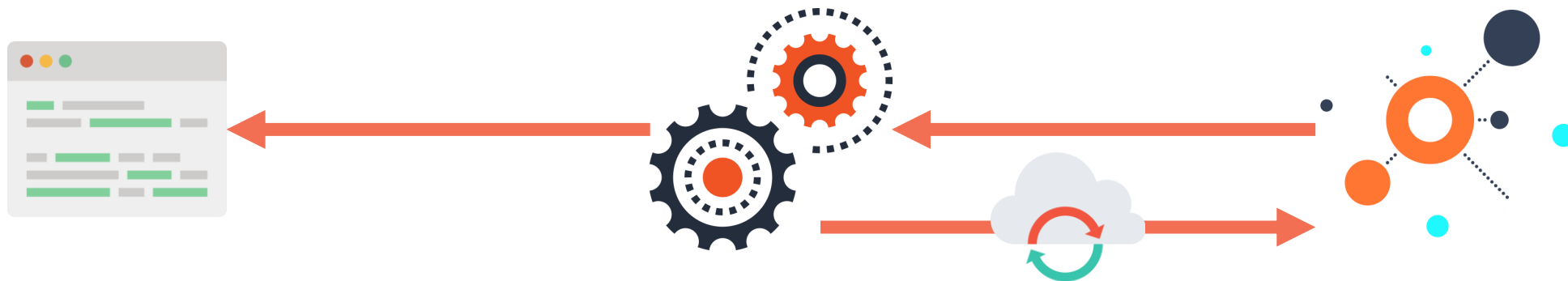


IndexedDB

an actual database in the browser

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

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 timeline** when browser closed

Service Worker Debugging



Chromium DevTools interface showing Service Worker debugging for **www.baqend.com**.

Service Workers Panel:

- Manifest: **Service Workers** (selected)
- Clear storage
- Storage: Local Storage, Session Storage, IndexedDB, Web SQL, Cookies
- Cache: Cache Storage, Application Cache
- Service Workers: **www.baqend.com** (Source: **sw.7dbf553e.js**, Received 23.1.2018, 15:38:40, Status: **#823 activated and is running**)
- Actions: **Update**, **Unregister**
- Clients: **https://www.baqend.com/** (focus)
- Push: Test push message from DevTools. (Push button)
- Sync: test-tag-from-devtools (Sync button)

Sources Panel:

- sw.7dbf553e.js:formatted (selected)
- Line 32, Column 9: `t.p = '';`

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)

Caching Strategies – Cache Only



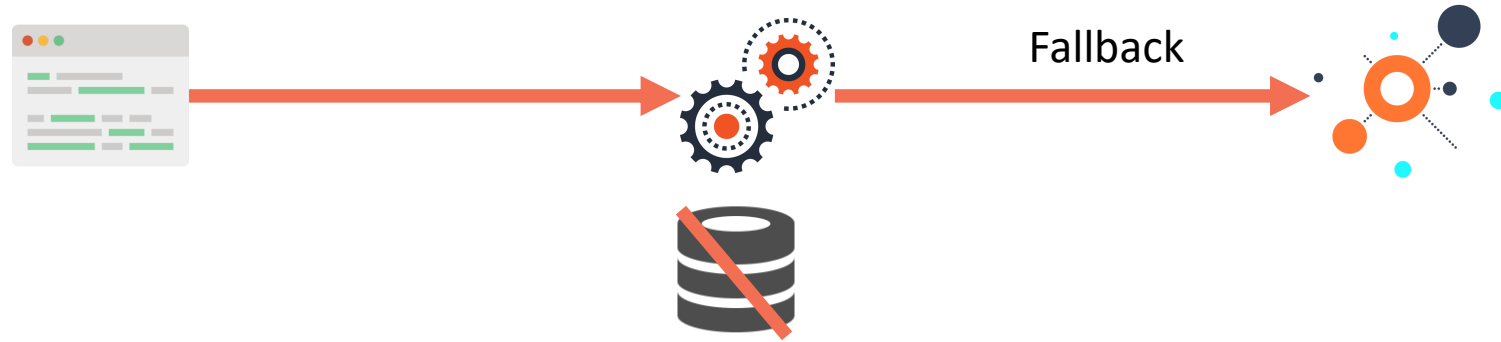
Gets all requests from cache or fails.

Caching Strategies – Cache, Network Fallback



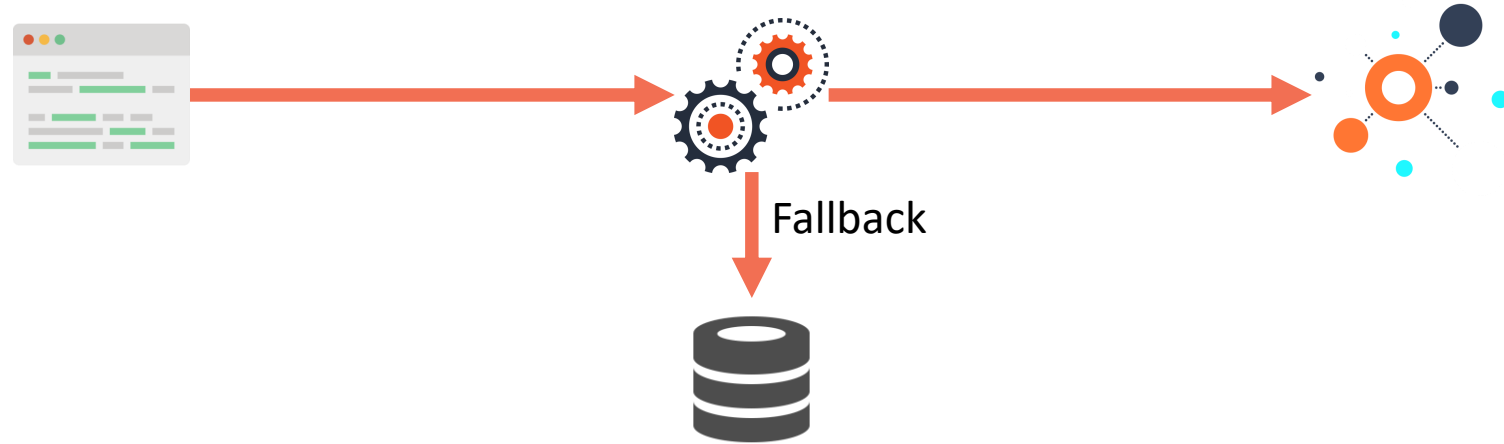
Gets requests from cache & uses network as fallback.

Caching Strategies – Network Only



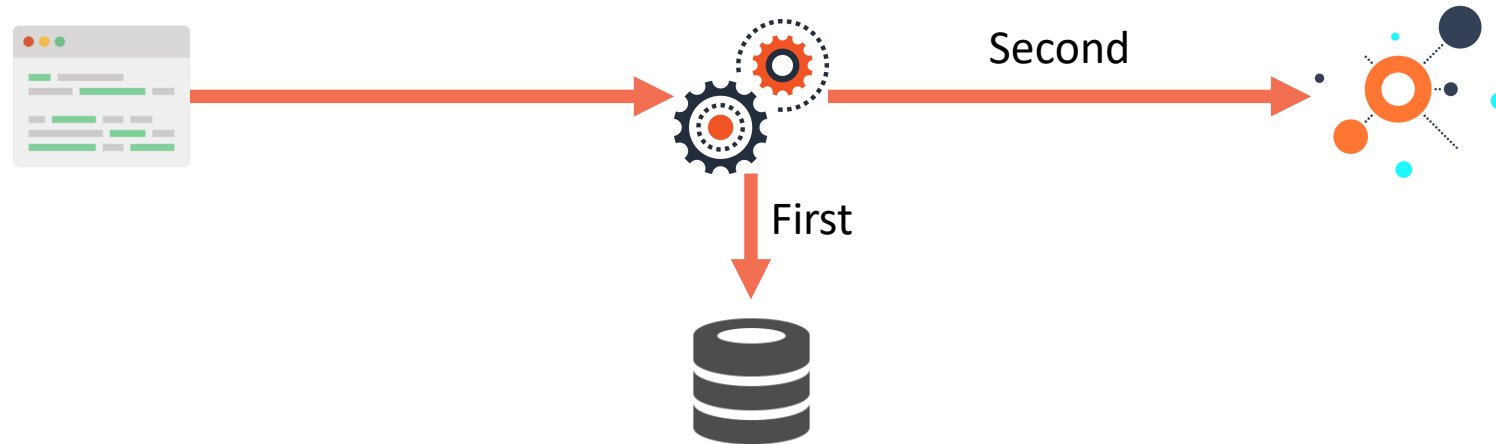
Gets requests from network only.

Caching Strategies – Network, Cache Fallback



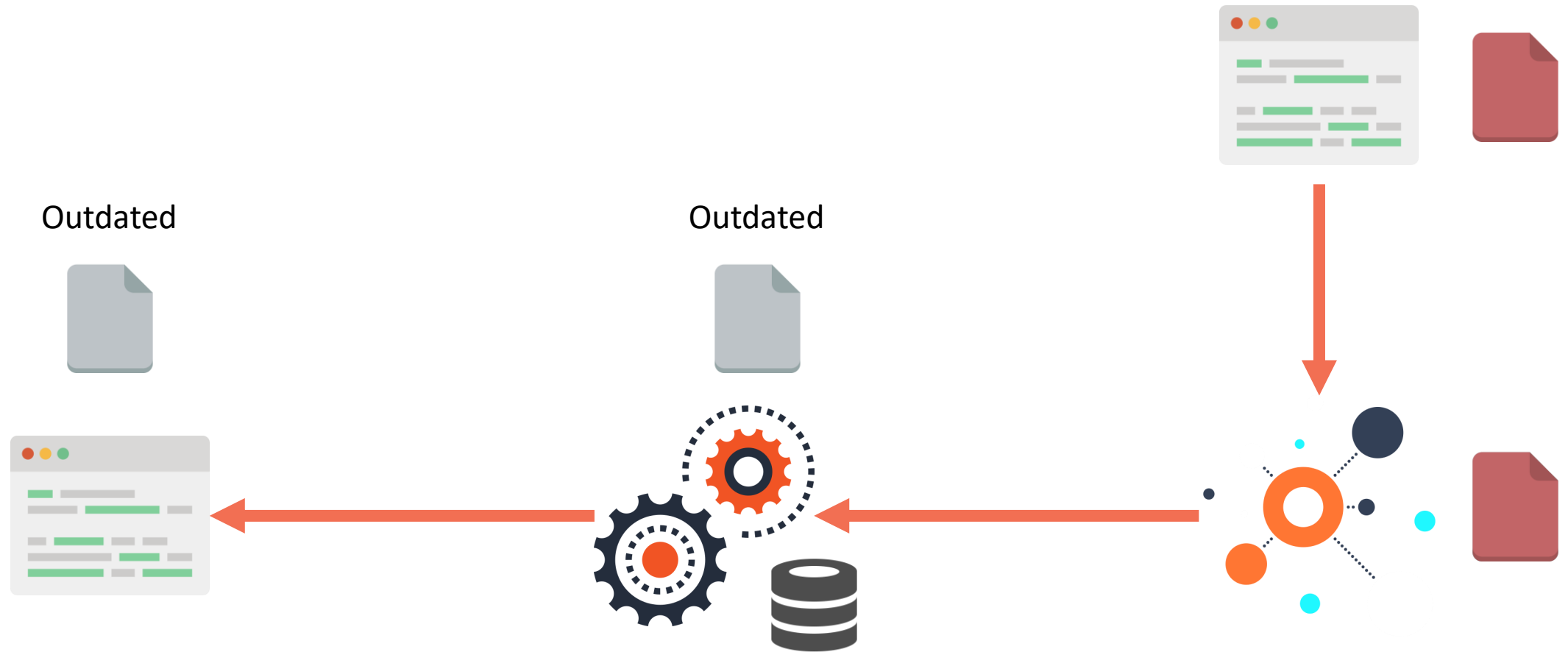
Gets requests from network, the cache acts as fallback (offline mode).

Caching Strategies – Cache, then Network



Gets requests from cache first and from network in background.

Major Challenge: Cache Coherence



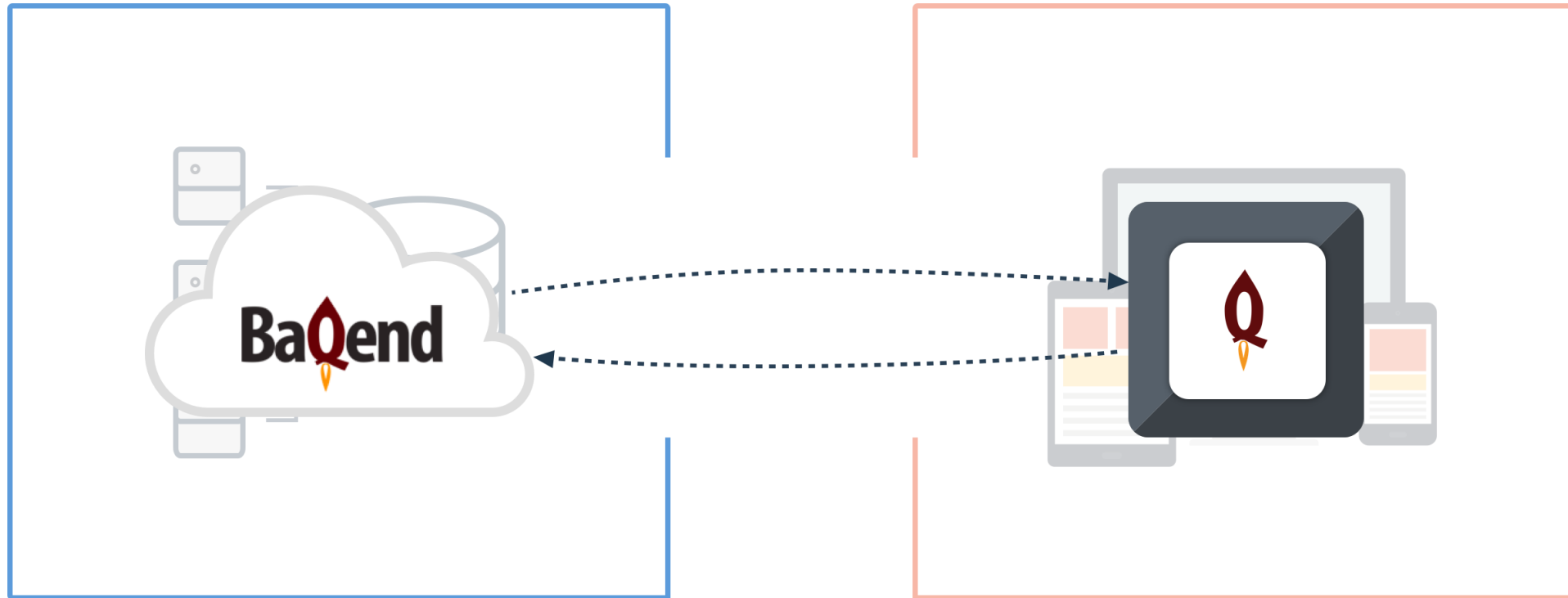
➡ All strategies either serve **outdated data** or **degrade performance**

A technician's hands are shown working on a device. In the background, a Tektronix 1155 oscilloscope displays two waveforms on its screen. To the right, a LEADER LV 5800 multimeter is connected to the device with test leads. The multimeter screen shows various settings and a list of numbers. On the left, a color calibration chart with vertical bars of various colors is visible. The entire scene is overlaid with a semi-transparent dark grey filter.

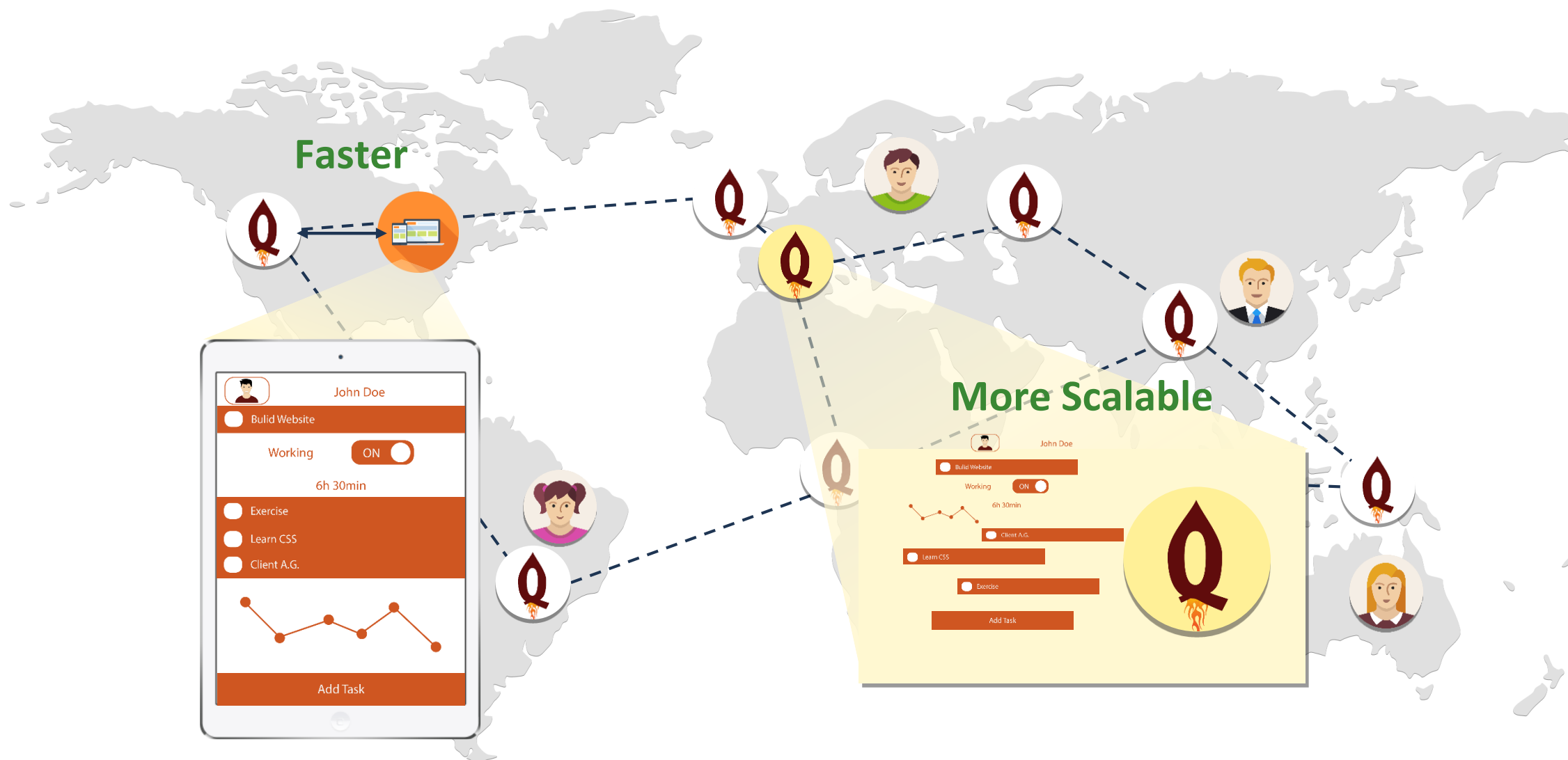
What we do with Service Workers

Speed Kit

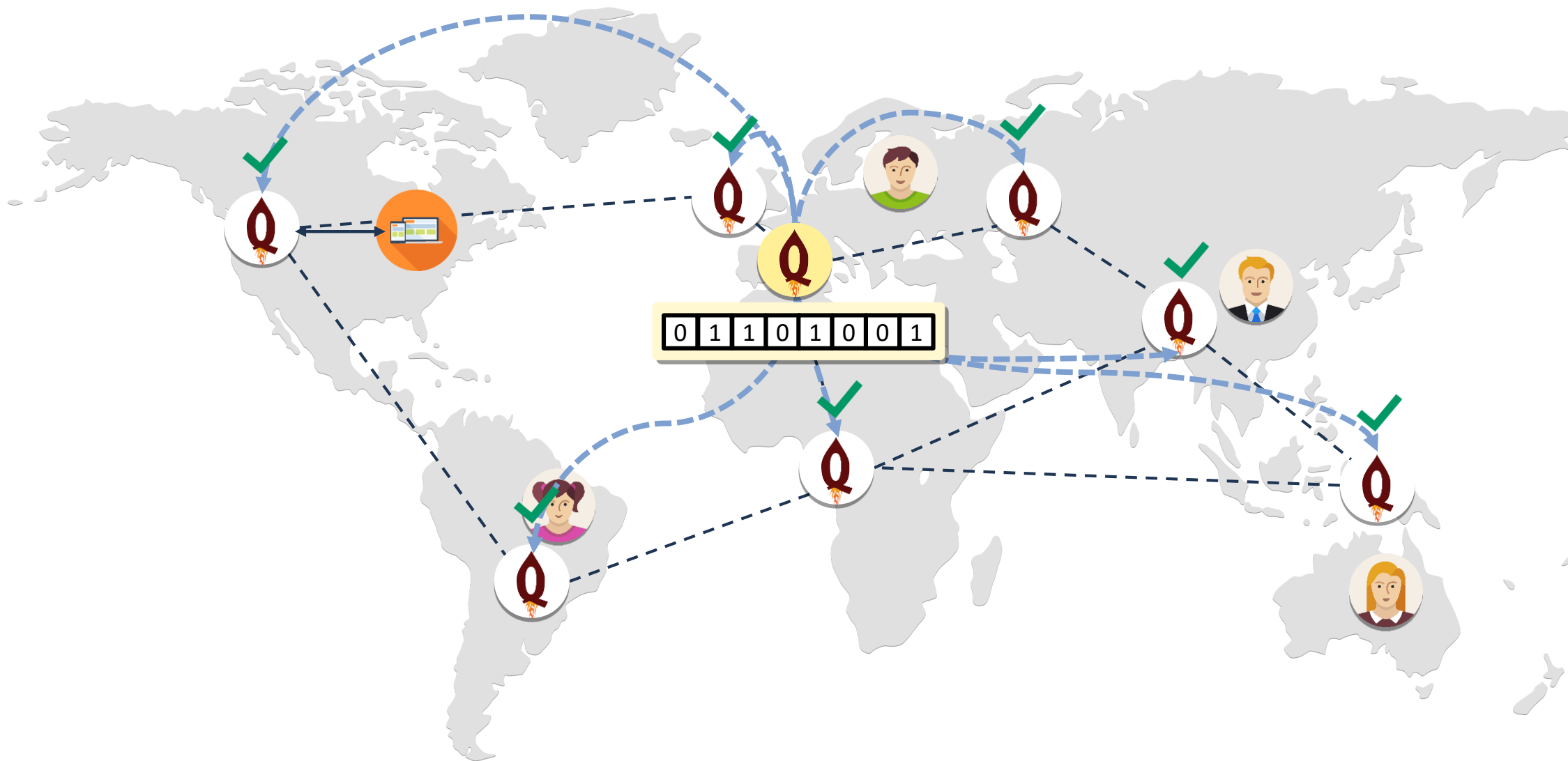
Turning Websites into Instantly-Loading
Progressive Web Apps



What **Speed Kit** does.



What **Speed Kit** does.



What Speed Kit does.

Backed by

30 Man-Years of Research

📖 F. Gessert, F. Bücklers, und N. Ritter, „ORESTES: a Scalable Database-as-a-Service Architecture for Low Latency“, in *CloudDB 2014*, 2014.

📖 F. Gessert und F. Bücklers, „ORESTES: ein System für horizontal skalierten, real-time Zugriff auf Cloud-Datenbanken“, in *Informatiktag 2013*, 2013.

📖 F. Gessert und F. Bücklers, *Performanz- und Reaktivitätssteigerung von OODBMS vermittelt durch Web-Caching-Verfahren*. Bachelorarbeit, 2010.

📖 M. Schaarschmidt, F. Gessert, und N. Ritter, „Towards Automated Polyglot Persistence“, in *BTW 2015*.

📖 S. Friedrich, W. Wingerath, F. Gessert, und N. Ritter, „NoSQL OLTP Benchmarking: A Survey“, in *44. Jahrestagung der Gesellschaft für Informatik*, 2014, Bd. 232, S. 693–704.

📖 W. Wingerath, F. Gessert, S. Friedrich, N. Ritter „Real-time stream processing for Big Data“, *Big Data Analytics it - Information Technology*, 2016

📖 F. Gessert, W. Wingerath, S. Friedrich, N. Ritter “NoSQL Database Systems: A Survey and Decision Guidance“, *Computer Science - Research and Development*, 2016

📖 F. Gessert, S. Friedrich, W. Wingerath, M. Schaarschmidt, und N. Ritter, „Towards a Scalable and Unified REST API for Cloud Data Stores“, in *44. Jahrestagung der GI*, Bd. 232, S. 723–734.

📖 F. Gessert, M. Schaarschmidt, W. Wingerath, S. Friedrich, und N. Ritter, „The Cache-keeper: Revisiting Expiration-based Caching in the Age of Cloud Data Management“, in *BTW 2015*.

📖 F. Gessert und F. Bücklers, *Kohärentes Web-Caching von Datenbankobjekten im Cloud Computing*. Masterarbeit 2012.

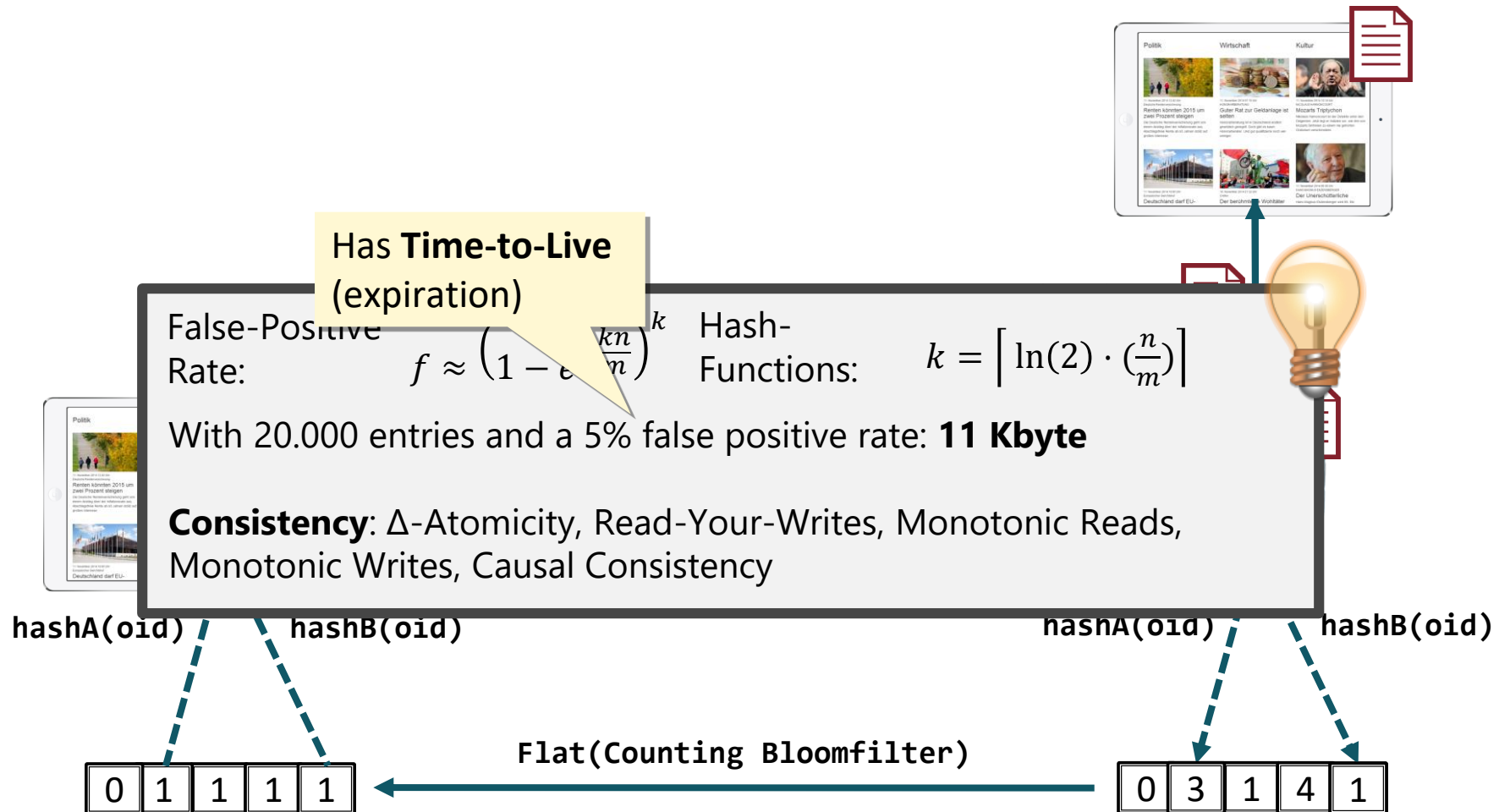
📖 W. Wingerath, S. Friedrich, und F. Gessert, „Who Watches the Watchmen? On the Lack of Validation in NoSQL Benchmarking“, in *BTW 2015*.

📖 F. Gessert, „Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis“, *BTW 2015*

📖 F. Gessert, N. Ritter „Scalable Data Management: NoSQL Data Stores in Research and Practice“, *32nd IEEE International Conference on Data Engineering, ICDE*, 2016

📖 F. Gessert, N. Ritter „Polyglot Persistence“, *Datenbank Spektrum*, 2016.

How Speed Kit solves Cache Coherence

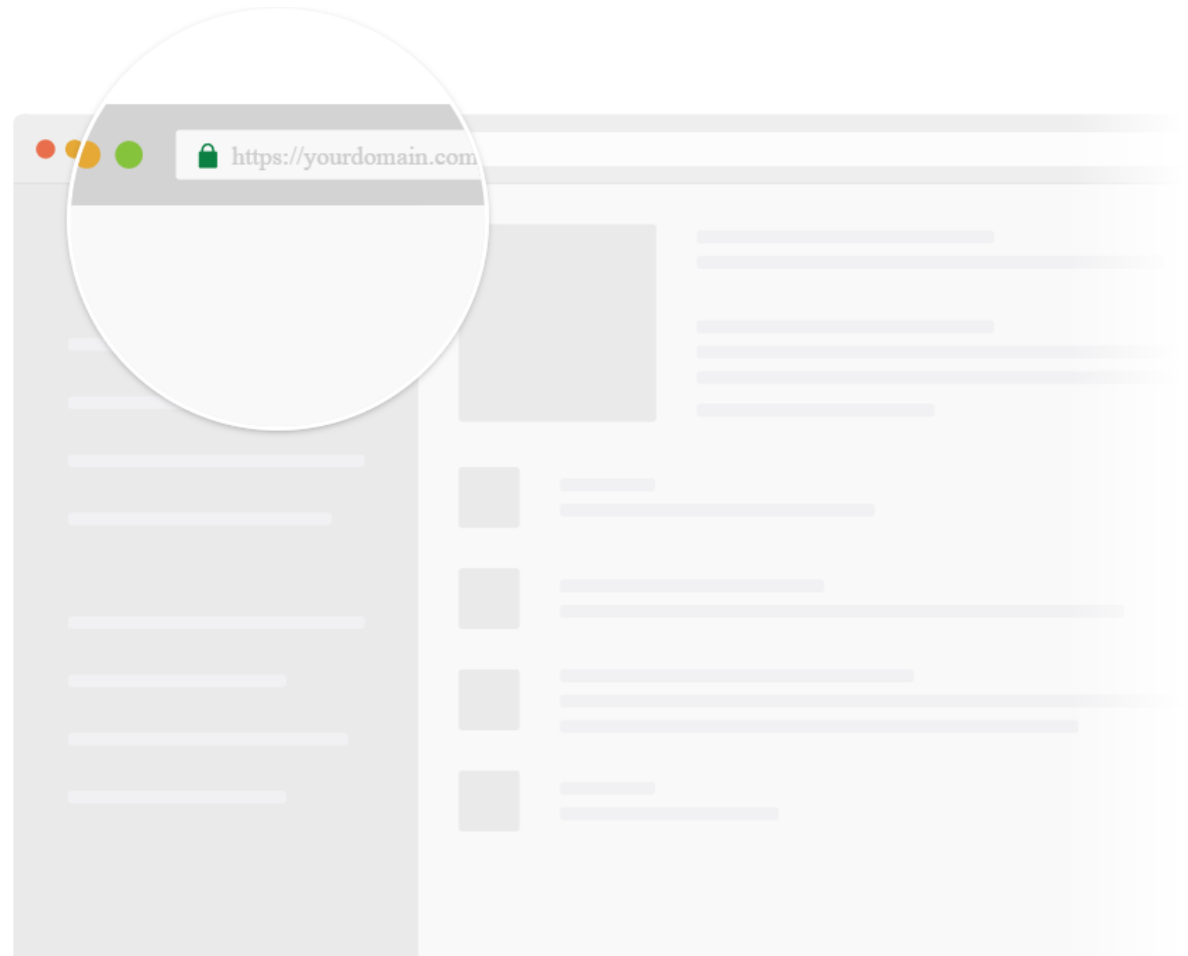




Adding **Speed Kit** to a Site

1. Configure Domain

Set which sites/URLS
Baqend should accelerate
(white- and blacklist,
dynamic blocks).



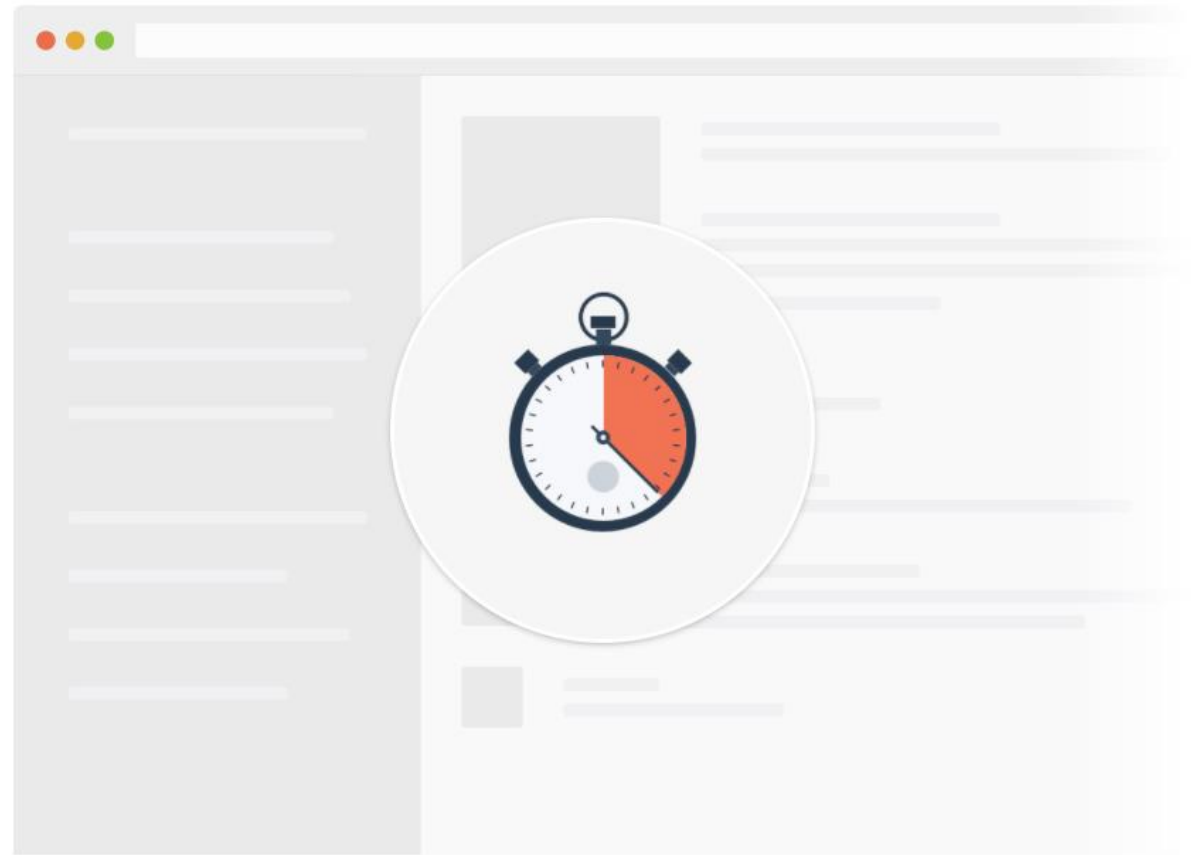
2. Include Code Snippet

Add Speed Kit to your website's HTML and upload the Service Worker.

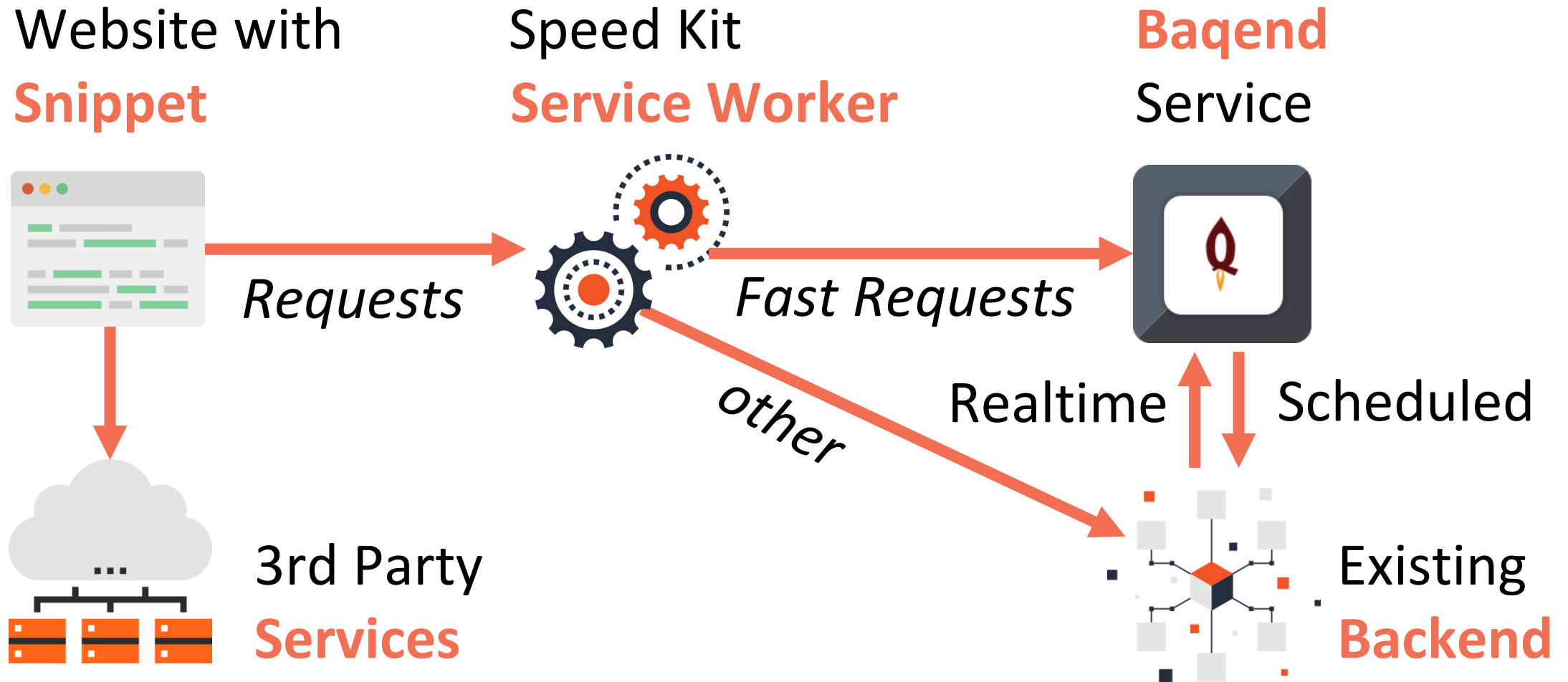


3. Speed Kit is Active

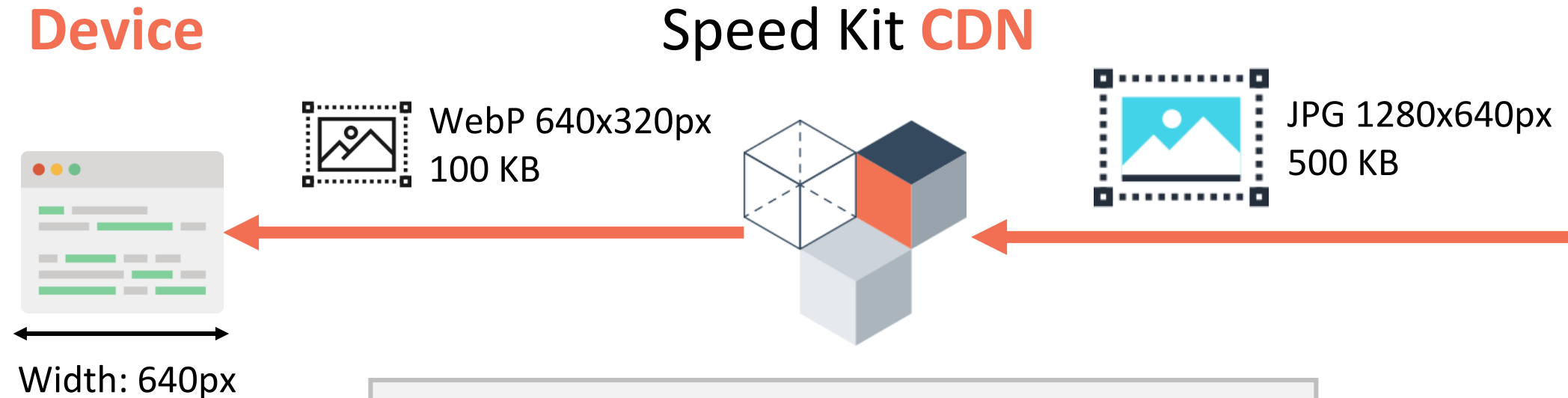
Speed Kit intercepts requests and serves them through Baqend's infrastructure.



How it **works** under the hood



Optimized & Cached Images



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

A person stands in the center of a vast, dark cave, illuminated by a bright light source from the distance. The cave walls are composed of large, textured rock formations. The word "Demo" is overlaid in the center of the image.

Demo



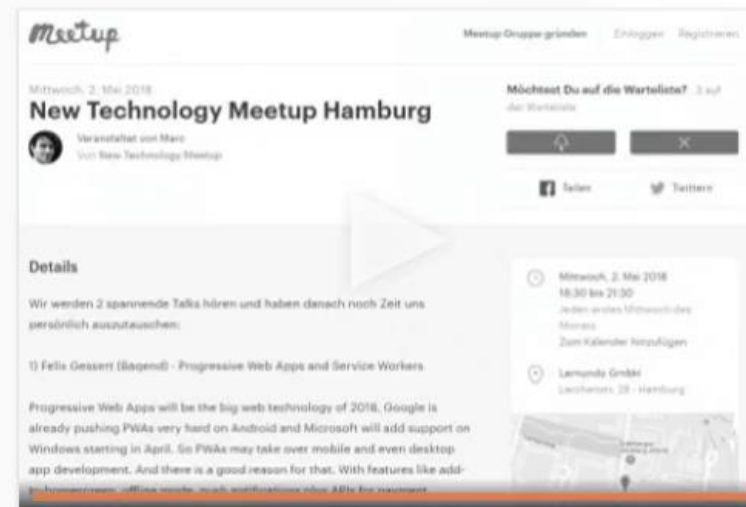
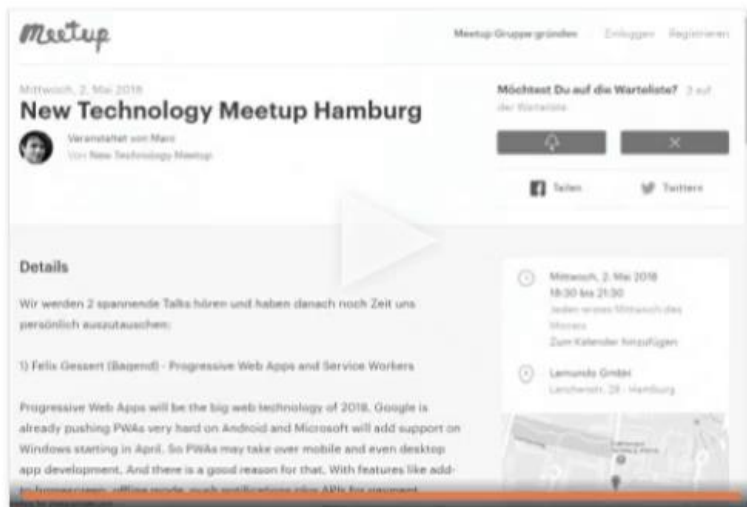
Now, we have a
Progressive Web App.
**How do we measure
web performance?**

<https://www.meetup.com/de-DE/meetup-group-bdCSNBEI/events/249149394/>

Your Website
1131ms

3.1x
Faster

With Speed Kit
366ms



With Speed Kit

0.4s

Your Website

1.1s

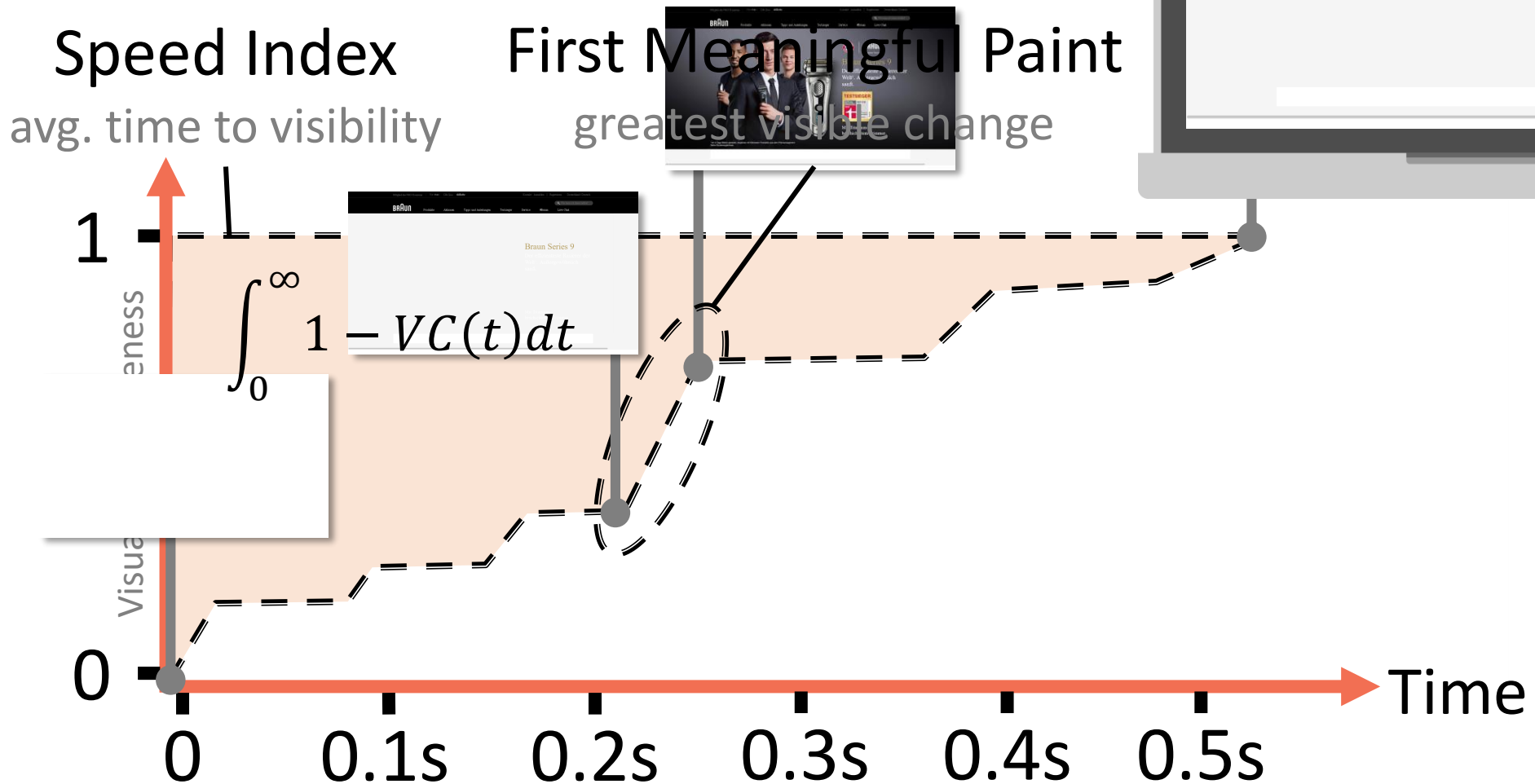
Excellent

Poor



Show Details

User-perceived performance



Does it work for **Your Site?**

www.example.com

Go

test.speed-kit.com

Wrap Up

PWA



Super cool
alternative to
native apps

Service Workers



Powerful
programmable
network proxy

Speed Kit



Combines Service
Workers & cache
coherence



Edit 🔍



Learn more about this topic:

<https://blog.baqend.com/>

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

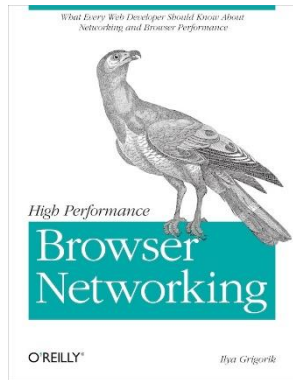
30 Man-Years of Research in a 30-Minute Read

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.

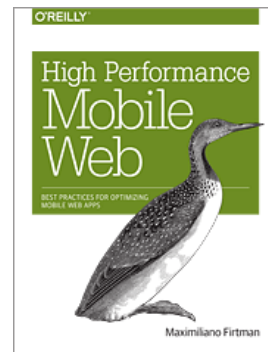
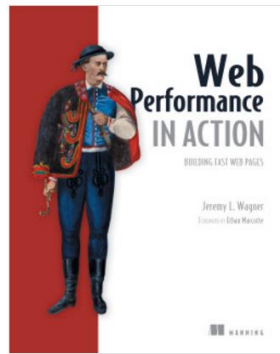


Web Performance Literature

Good Resources



<https://hpbn.co/>



Google Developers

Performance

<https://developers.google.com/web/fundamentals/performance/?hl=en>

Website Performance Optimization
The Critical Rendering Path

<https://www.udacity.com/course/website-performance-optimization--ud884>



Baqend Blog

On Building a Faster Web

<https://medium.baqend.com/>

Performance Tools

PageSpeed Insights 

☒ Mobil ☒ Desktop

<https://developers.google.com/speed/pagespeed/>

Page Speed Analyzer

Test the performance of your site!

Choose how to test:

Region of client	<input type="button" value="USA"/>	<input type="button" value="EU"/>
Cold cache	<input type="button" value="NO"/>	<input type="button" value="YES"/>

<https://test.speed-kit.com>



<https://www.baqend.com/>



<http://www.webpagetest.org/>



We are hiring.

Frontend Developers
Mobile Developers
Java Developers
Web Performance Engineers

Contact us.



Felix Gessert · fg@baqend.com · www.baqend.com/jobs.html