

**EVERY PROJECT A
PROGRESSIVE WEB APP**



DE VOORHOEDE



a team of
20+ front-end devs
2 project managers

in **Amsterdam**
and (soon) **Delft**



we keep up with
the **latest web**
technology



we **share** insights
and open source
tools

OUR WORK



SITA
Create success. Together

A screenshot of the funda website's search interface. At the top, it says "Een huis dat alles heeft, zelfs terrasjes om de hoek" with a colorful illustration of houses. Below that are three buttons: "Koop", "Huur", and "Meer ▾". Underneath are two dropdown menus: "Spaarndammerstraat, Amst" and "+ 0 km ▾". Below those are two more dropdown menus: "van € 0" and "tot Geen maximum". At the bottom is a blue "Zoek" button.

Ziggo



Heb jij thuis Ziggo
Interactive Televisie? Druk
dan op de rode knop en
bekijk het resultaat!

PROGRESSIVE WEB APPS





Progressive



Responsive



Network
Independent



App-Like



Fresh



Safe



Discoverable



Re-engageable



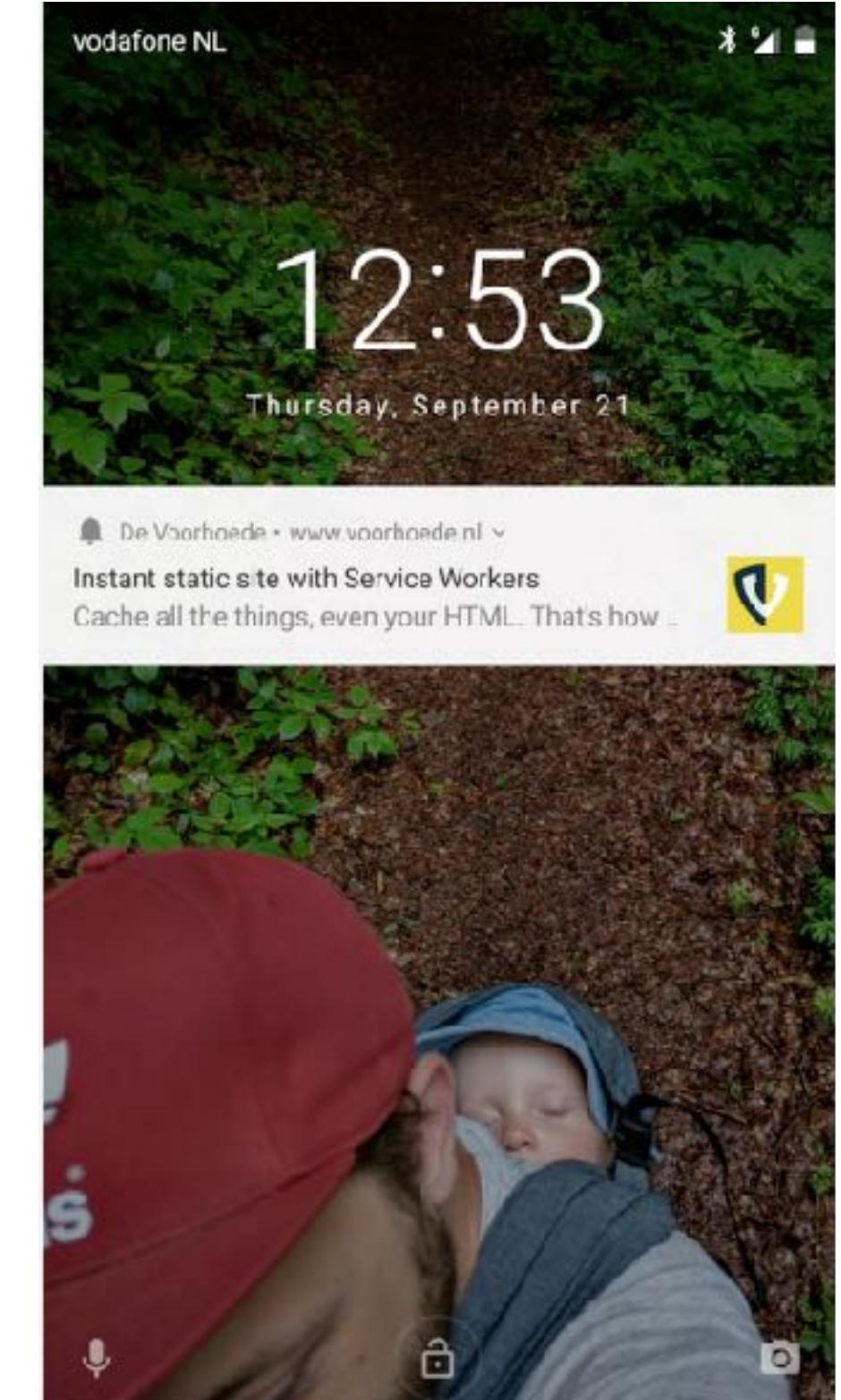
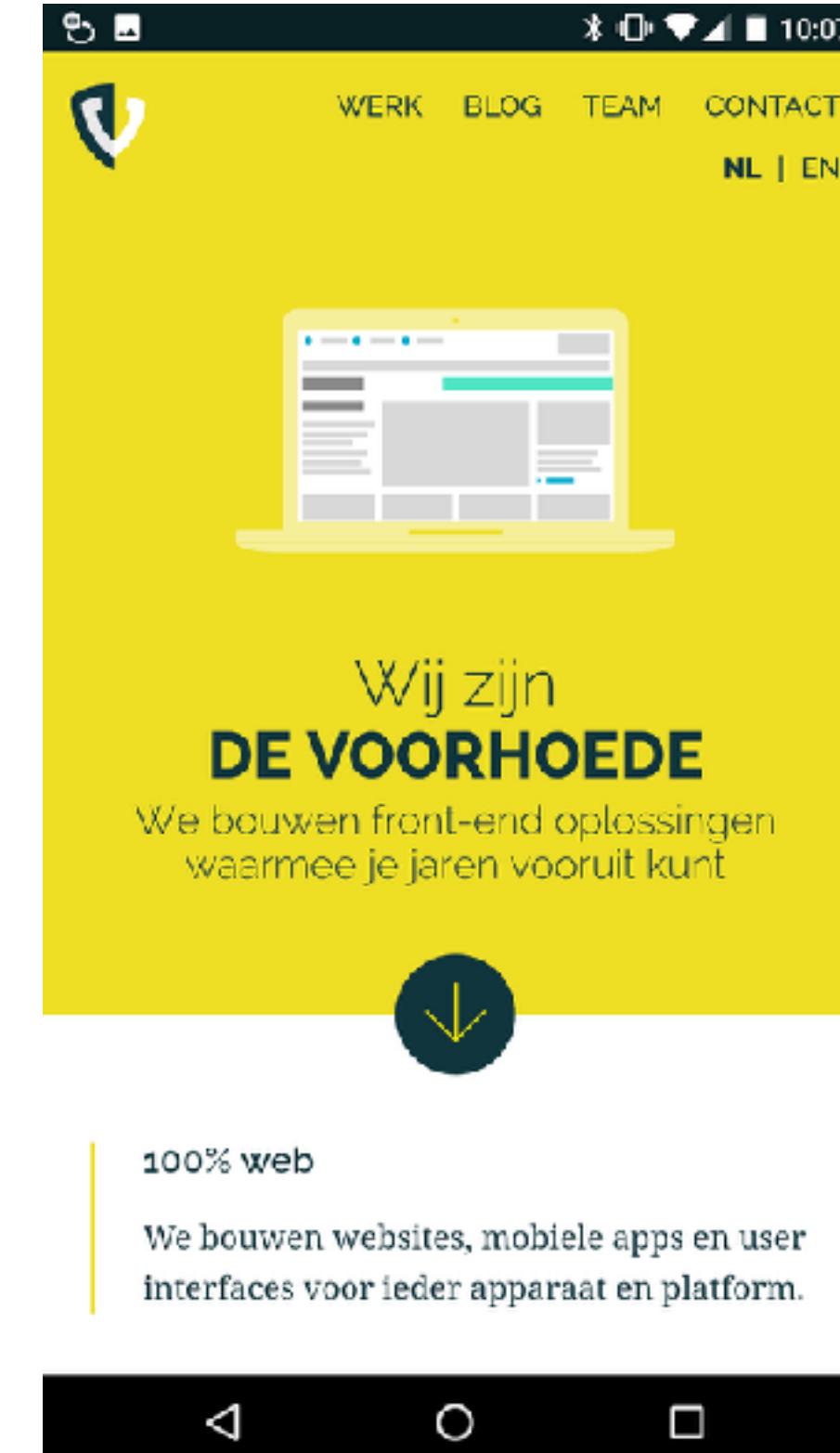
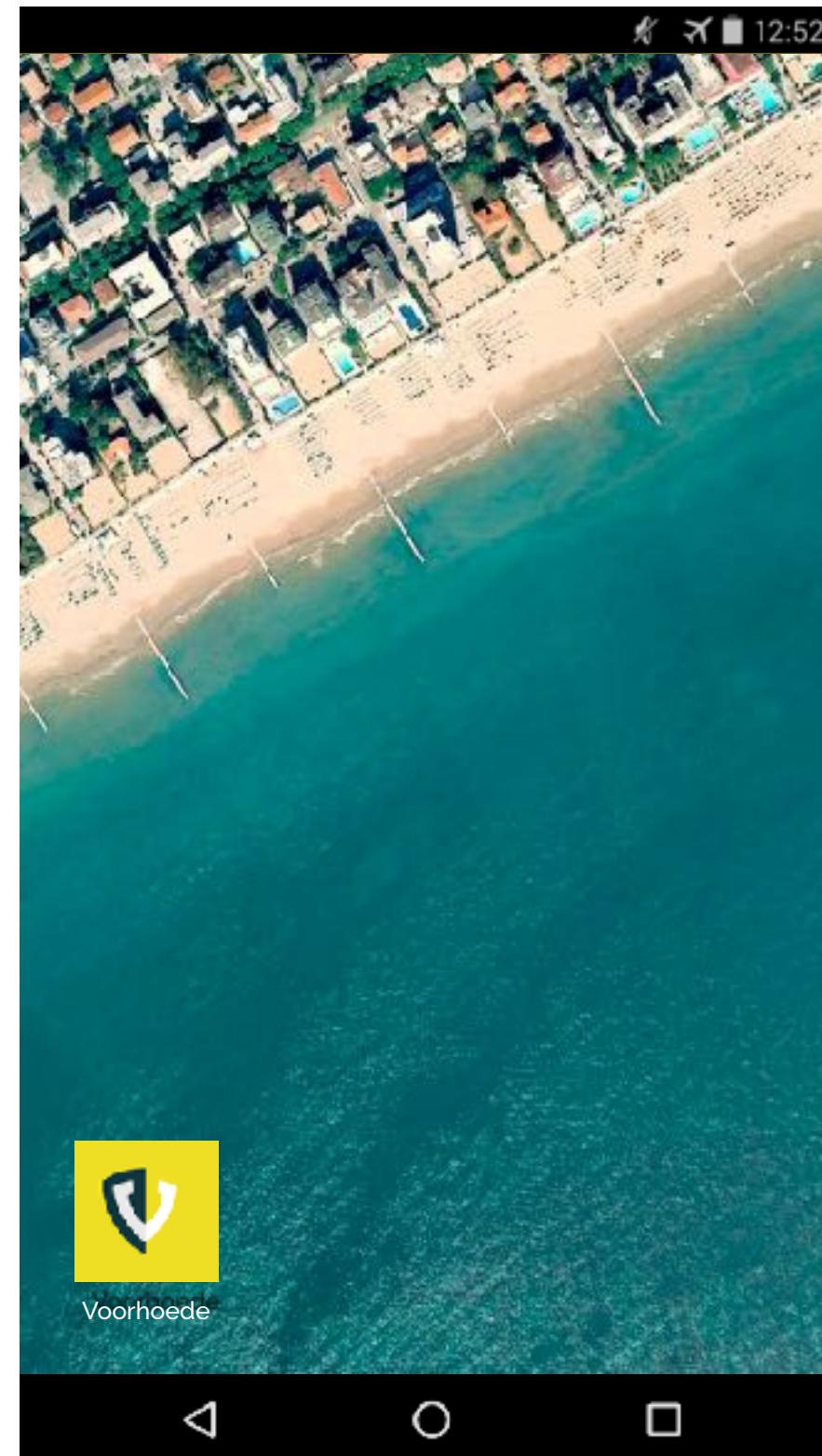
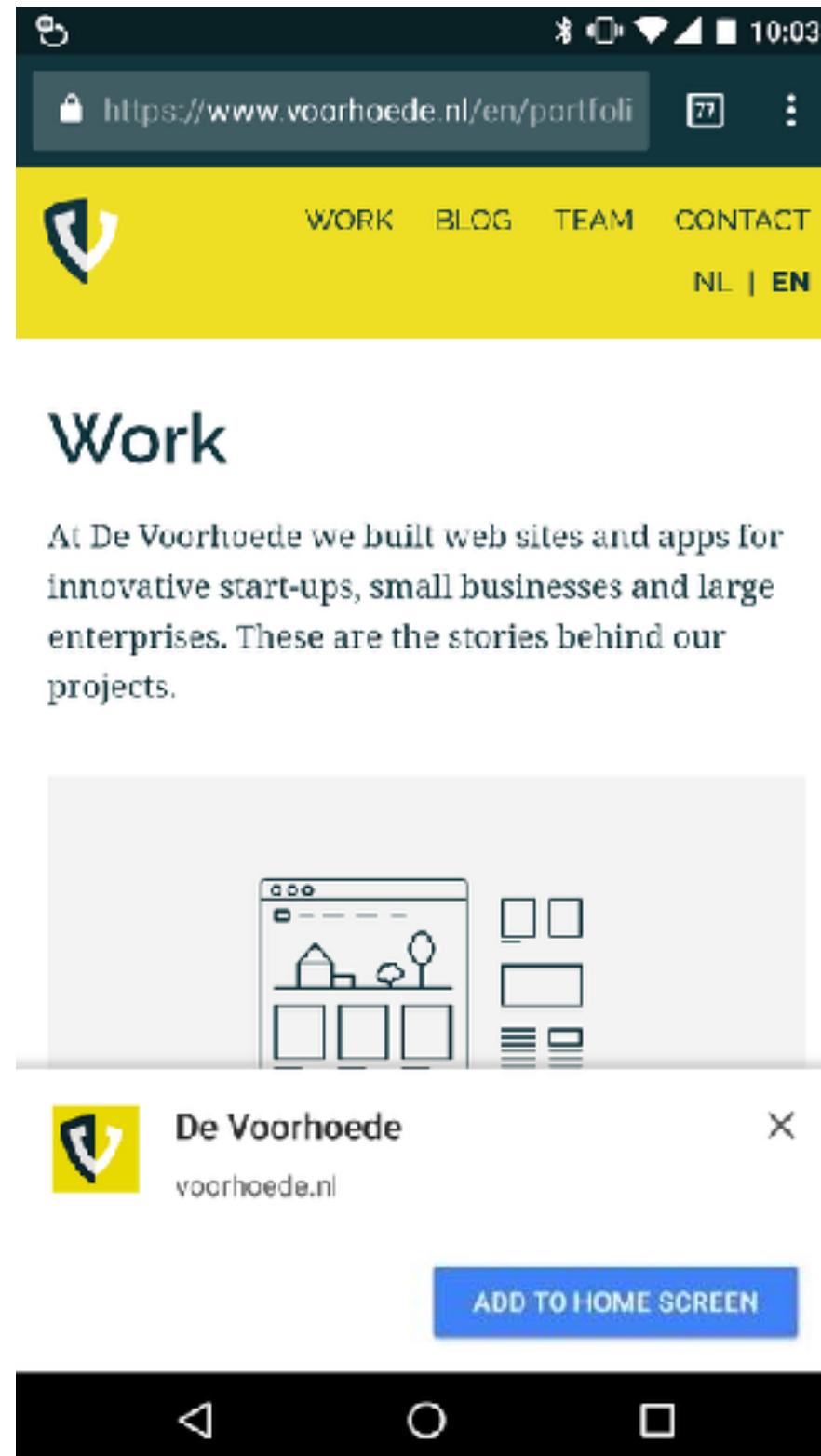
Installable



Linkable

A PWA is

FROM SITE TO PWA



voorhoede.nl

THE PWA PROMISE:

RE-ENGAGEMENT ↑

RETENTION ↑

CONVERSION ↑

use the
latest web technologies
to create a
1st class app experience

CHROME, ANDROID (2018-03)

Native Behaviors	Seamless Experience	Surroundings	Input
<ul style="list-style-type: none">LOCAL NOTIFICATIONS ✓PUSH MESSAGES ✓HOME SCREEN INSTALLATION ✓FOREGROUND DETECTION ✓PERMISSIONS ✓	<ul style="list-style-type: none">OFFLINE MODE ✓BACKGROUND SYNC ✓INTER-APP COMMUNICATION ✓PAYMENTS ✓CREDENTIALS ✓	<ul style="list-style-type: none">BLUETOOTH ✓USB ✓NFC ✘PROXIMITY SENSORS ✘AMBIENT LIGHT ✘	<ul style="list-style-type: none">TOUCH GESTURES ✓SPEECHrecognition ✓CLIPBOARD (COPY & PASTE) ✓POINTING DEVICE ADAPTATION ✓

Location & Position	Device Features	Screen & Output	Camera & Microphone
<ul style="list-style-type: none">GEOLOCATION ✓GEOFENCING ✘DEVICE ORIENTATION ✓DEVICE MOTIONS ✓PROXIMITY SENSORS ✘	<ul style="list-style-type: none">NETWORK TYPE & SPEED ✓ONLINE STATE ✓VIBRATION ✓BATTERY STATUS ✓DEVICE MEMORY ✓	<ul style="list-style-type: none">VIRTUAL & AUGMENTED REALITY ✓FULLSCREEN ✓SCREEN ORIENTATION & LOCK ✓WAKE LOCK ✘Presentation FEATURES ✓	<ul style="list-style-type: none">AUDIO & VIDEO CAPTURE ✓ADVANCED CAMERA CONTROLS ✓RECORDING MEDIA ✓REAL-TIME COMMUNICATION ✓



What Web Can Do Today

Can I rely on the Web Platform features to build my app?
An overview of the device integration HTML5 APIs

✓ Feature available in your current browser

✗ Feature not available in your current browser

Operating System
<ul style="list-style-type: none">OFFLINE STORAGE ✓FILE ACCESS ✓CONTACTS ✘SMS ✘STORAGE QUOTAS ✓TASK SCHEDULING ✘

Help,
my users have
an iPhone



image source: [bloomberg.com](#)

SAFARI, IOS (2018-03)

Native Behaviors
LOCAL NOTIFICATIONS ✘
PUSH MESSAGES ✘
HOME SCREEN INSTALLATION ✘
FOREGROUND DETECTION ✓
PERMISSIONS ✘

Seamless Experience
OFFLINE MODE ✘
BACKGROUND SYNC ✘
INTER-APP COMMUNICATION ✘
PAYMENTS ✓
CREDENTIALS ✘

Surroundings
BLUETOOTH ✓
USB ✓
NFC ✘
PROXIMITY SENSORS ✘
AMBIENT LIGHT ✘

Input
TOUCH GESTURES ✓
SPEECHrecognition ✘
CLIPBOARD (COPY & PASTE) ✓
POINTING DEVICE ADAPTATION ✓

Camera & Microphone
AUDIO & VIDEO CAPTURE ✓
ADVANCED CAMERA CONTROLS ✘
RECORDING MEDIA ✘
REAL-TIME COMMUNICATION ✓

Operating System
OFFLINE STORAGE ✓
FILE ACCESS ✓
CONTACTS ✘
SMS ✘
STORAGE QUOTAS ✘
TASK SCHEDULING ✘



What Web Can Do Today

Can I rely on the Web Platform features to build my app?
An overview of the device integration HTML5 APIs

✓ Feature available in your current browser

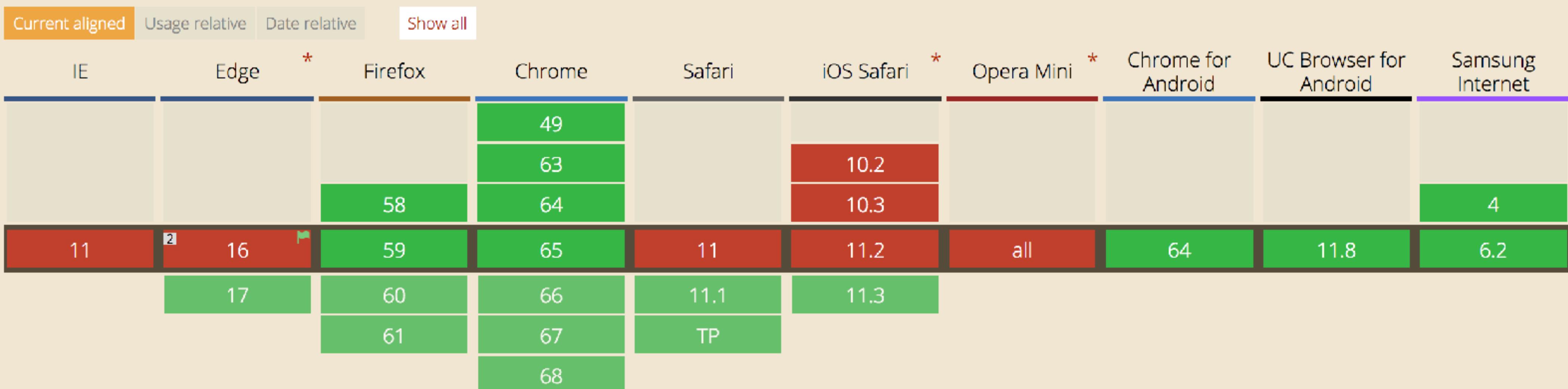
✗ Feature not available in your current browser

Service Workers 📁 - WD

Usage
Global

% of all users
75.23% + 0.18% = 75.41%

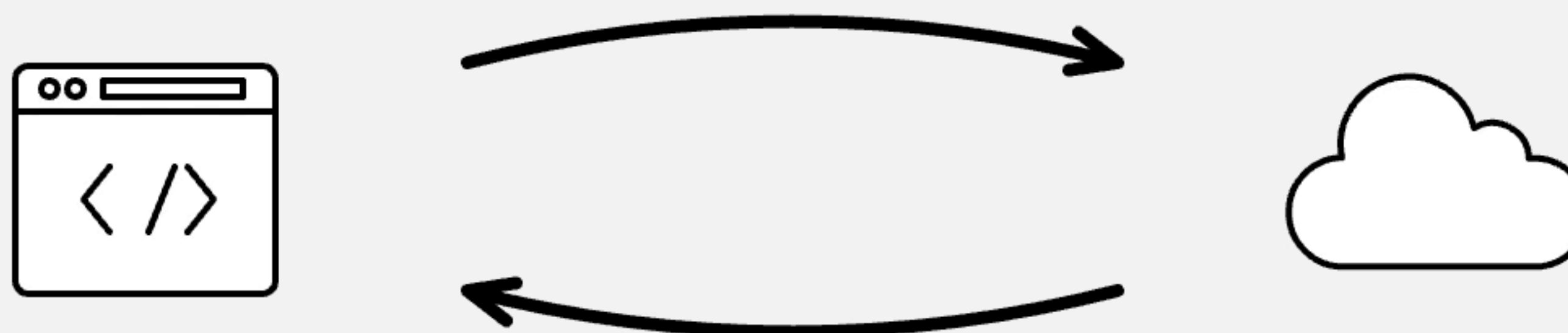
Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline.



THE SERVICE WORKER



INITIAL PAGE LOAD

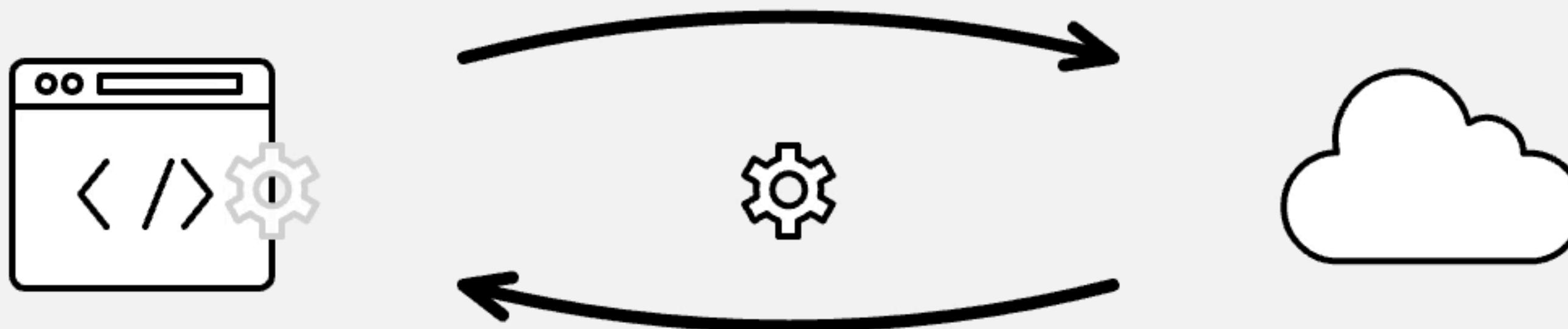


REGISTER SERVICE WORKER

```
<!-- page.html -->
<script>
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/sw.js')
}
</script>
```

```
// sw.js
console.log('Hello from the Service Worker')
```

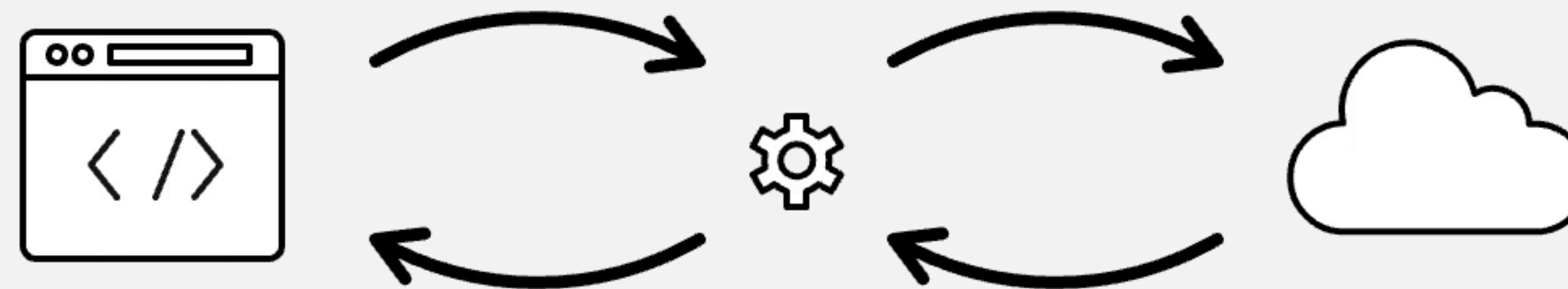
REGISTER SERVICE WORKER



SERVICE WORKER AS PROXY

```
// sw.js
self.addEventListener('fetch', async (event) => {
  console.log('request', event.request)
  const response = await fetch(event.request)
  console.log('response', response)
  event.respondWith(response)
})
```

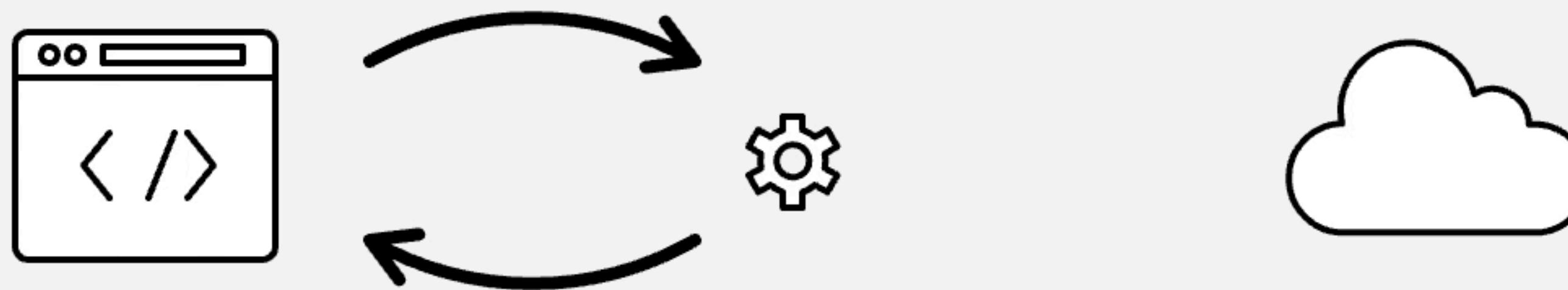
SERVICE WORKER AS PROXY



HIJACK FETCH

```
// sw.js
self.addEventListener('fetch', (event) => {
  const response = new Response('Hijacked :-)')
  event.respondWith(response)
})
```

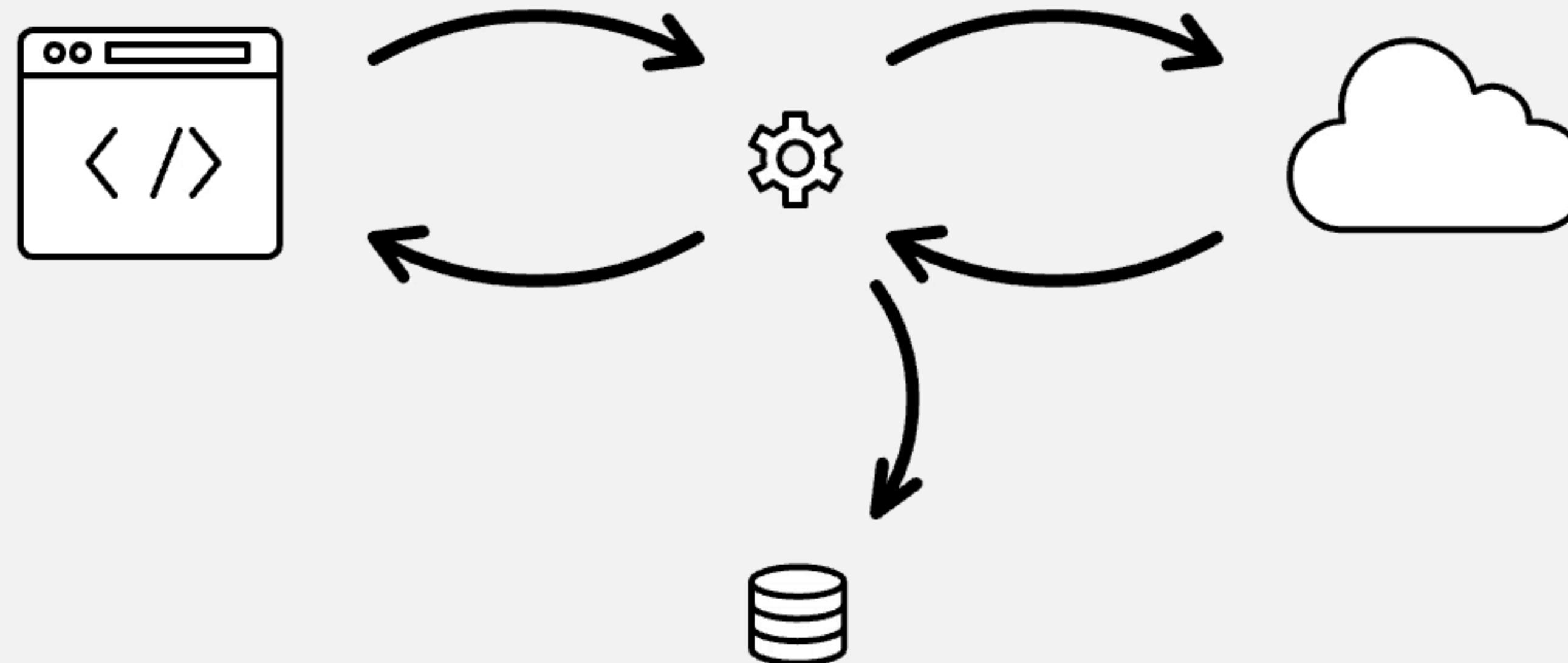
HIJACK FETCH



CACHE ON FETCH

```
// sw.js
self.addEventListener('fetch', async (event) => {
  const request = event.request
  const response = await fetch(request)
  const cache = await caches.open('all-v1')
  await cache.put(request, response.clone())
  event.respondWith(response)
})
```

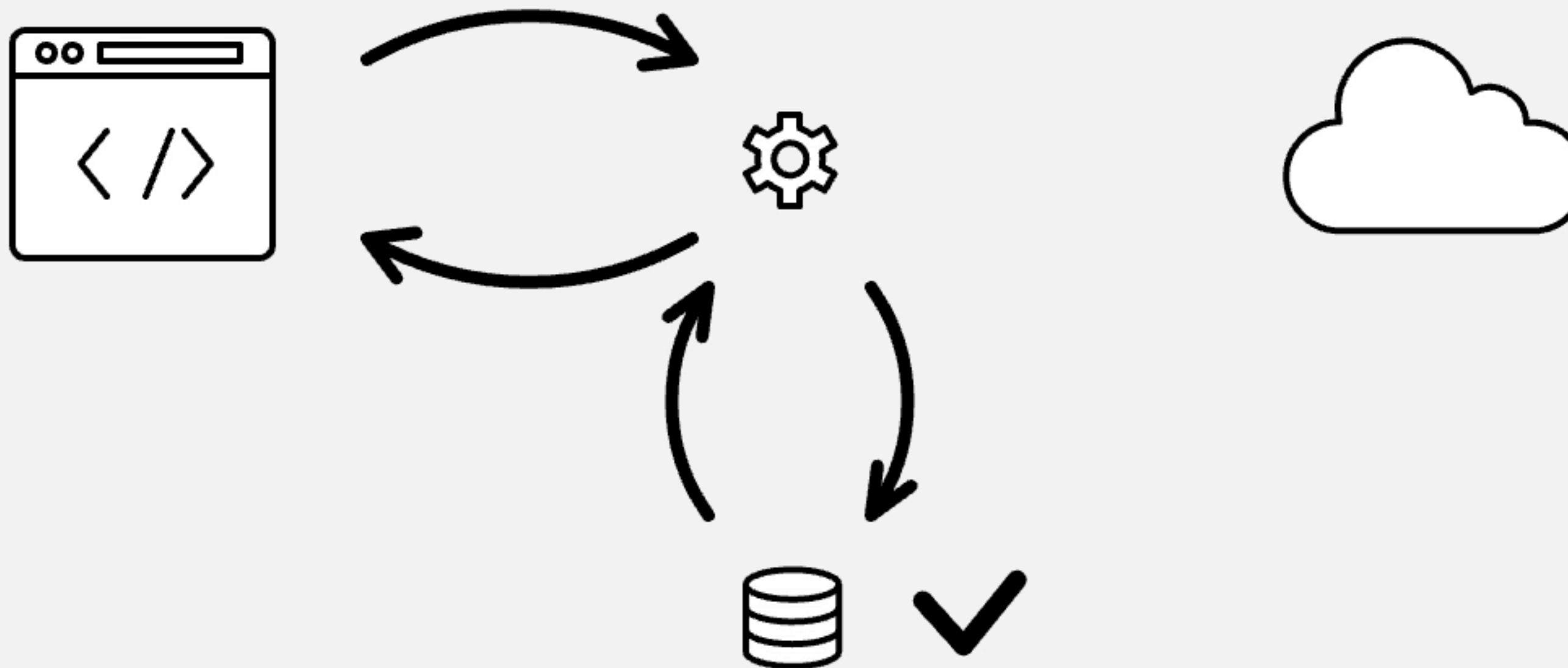
CACHE ON FETCH



SERVE FROM CACHE

```
// sw.js
self.addEventListener('fetch', async (event) => {
  const request = event.request
  const cache = await caches.open('all-v1')
  const response = await cache.match(request)
  event.respondWith(response)
})
```

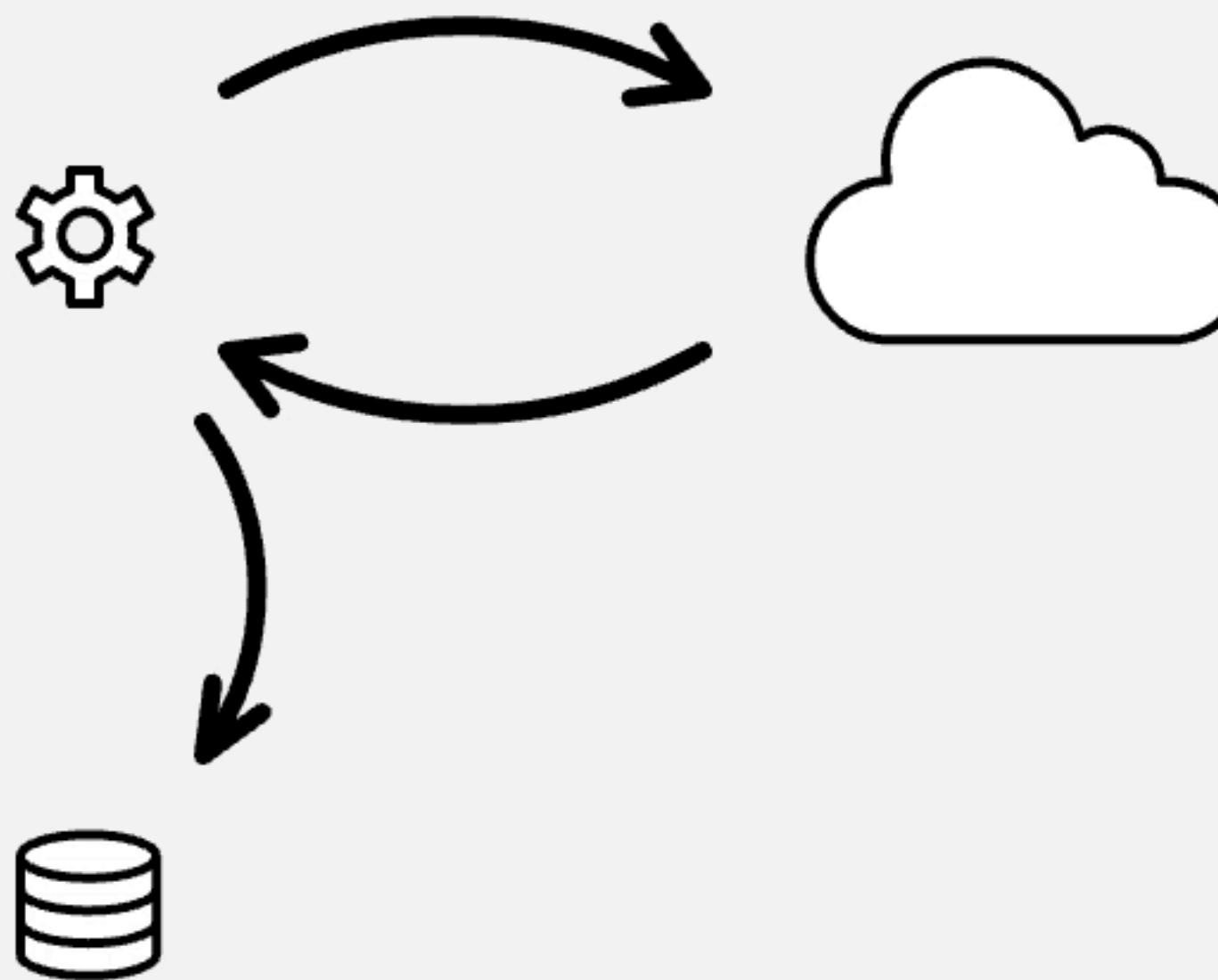
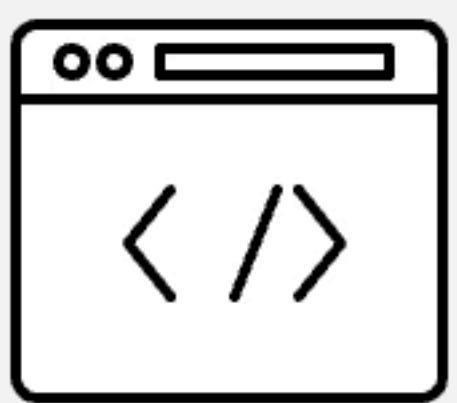
SERVE FROM CACHE



CACHE ON INSTALL

```
// sw.js
self.addEventListener('install', event =>
event.waitUntil(
  caches.open('core-v1')
    .then(cache => cache.addAll([
      '/offline/',
      '/my-app.css',
      '/my-app.js',
      '/my-font.woff2'
    ]))
    .then(self.skipWaiting())
))
```

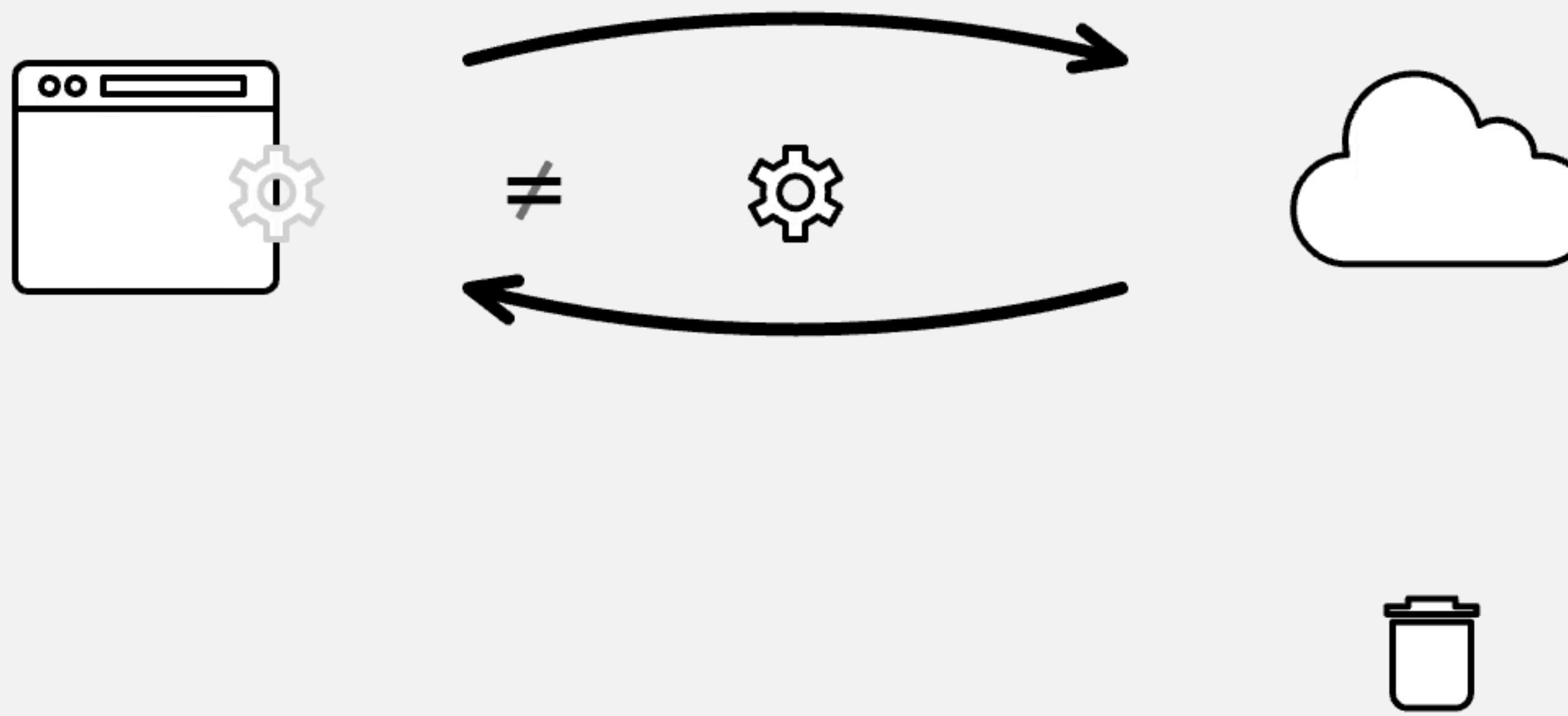
CACHE ON INSTALL



UPDATE SERVICE WORKER

```
// sw.js
self.addEventListener('activate', event =>
event.waitUntil(
  caches.keys().then(cacheNames => Promise.all(
    cacheNames.map(cacheName => {
      if (!cacheName.endsWith('-v5')) {
        return caches.delete(cacheName)
      }
    })
  )).then(() => console.log('V5 can take over'))
))
```

UPDATE SERVICE WORKER



PWAS IN PRACTICE



STARTS AS WEBSITE



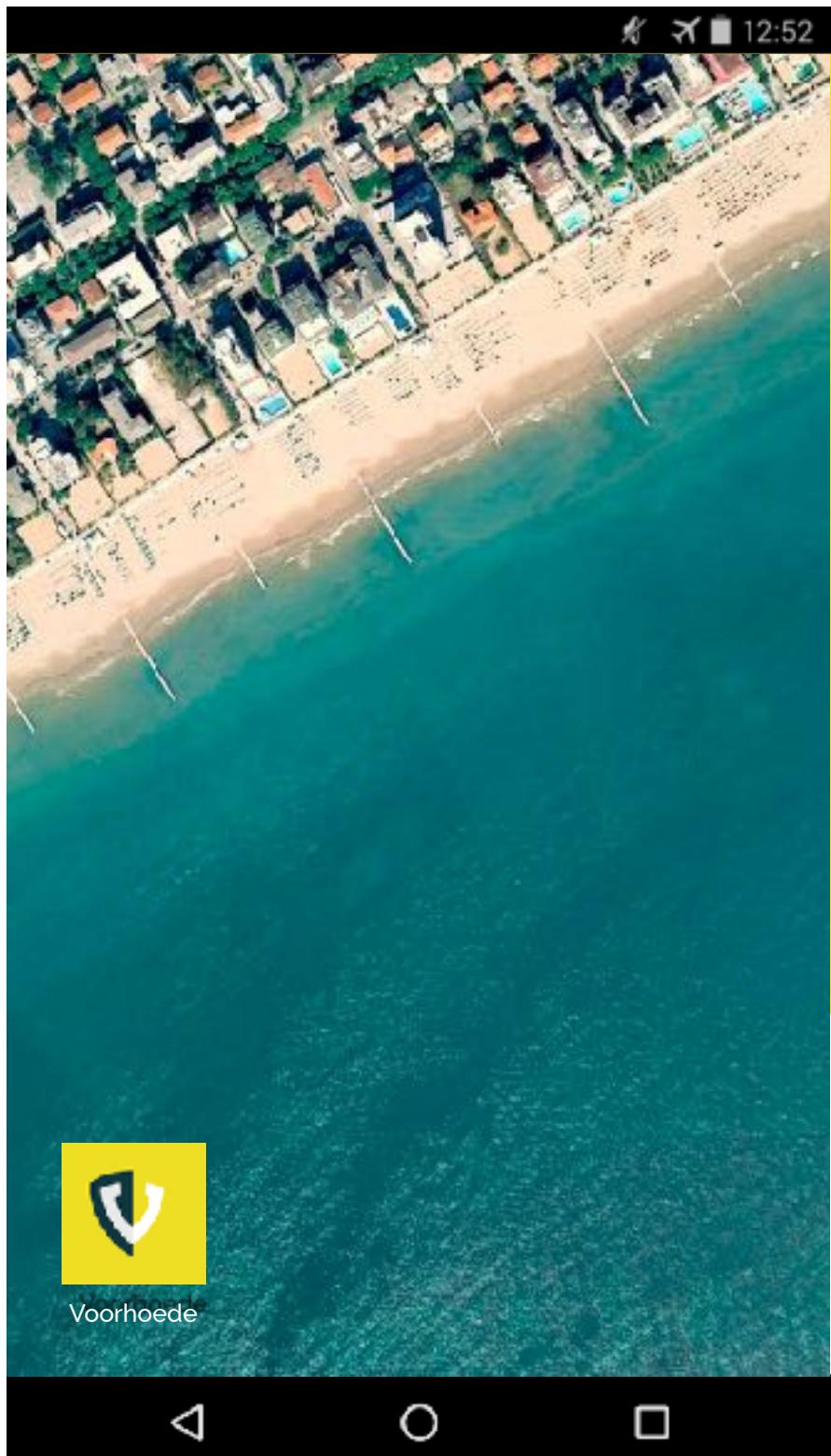
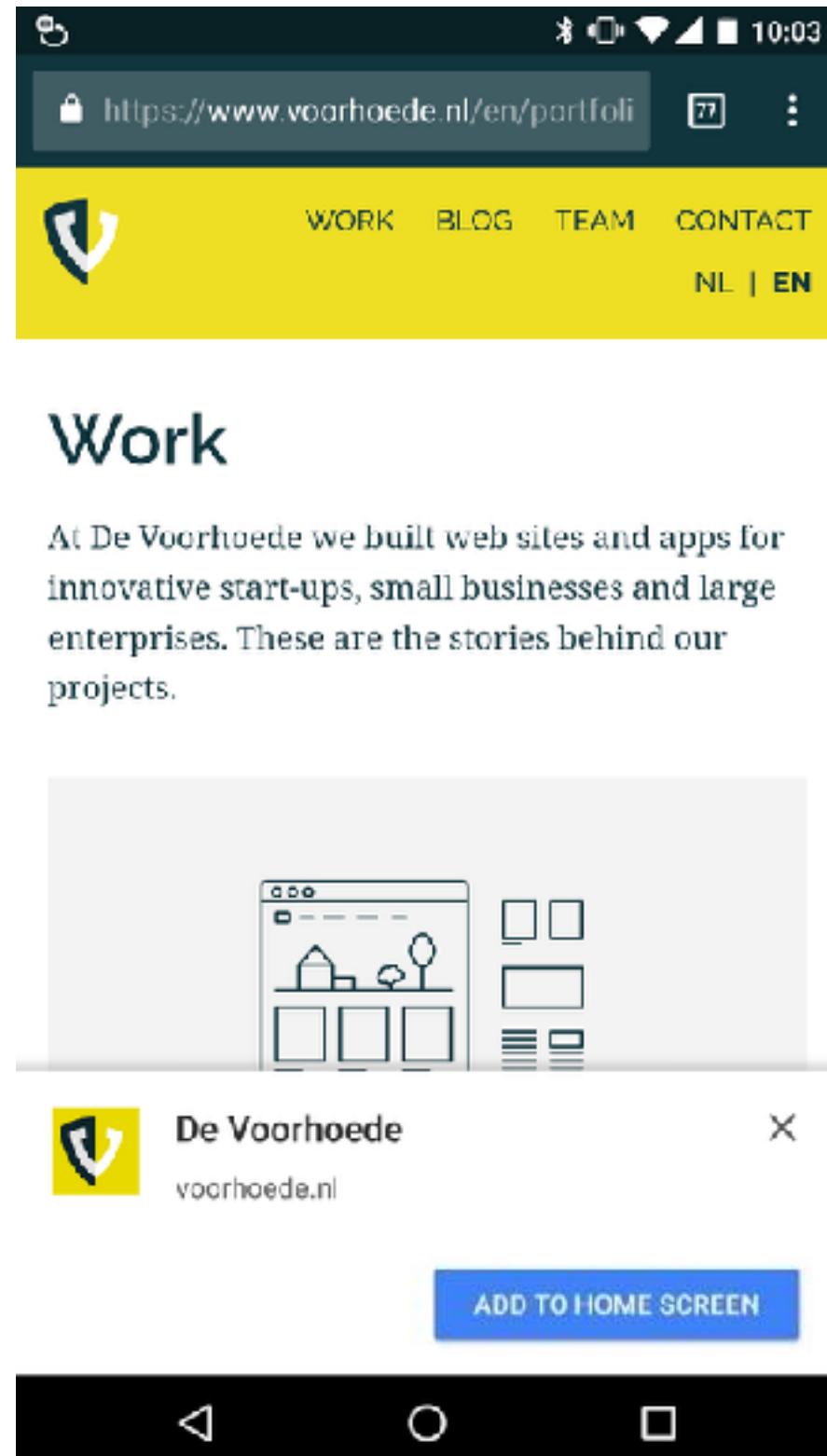
The screenshot shows a browser window with the title 'Blog | De Voorhoede'. The address bar indicates a secure connection to <https://www.voorhoede.nl>. The page has a yellow header featuring the company logo (a stylized 'V' icon) and the text 'DE VOORHOEDE front-end developers'. On the right side of the header are navigation links for 'WERK', 'BLOG', 'TEAM', 'CONTACT', and 'NL|EN'. Below the header is a large yellow background area containing the text 'Wij zijn **DE VOORHOEDE**' and 'We bouwen front-end oplossingen waarmee je jaren vooruit kunt'. A small circular arrow icon is located at the bottom center of this yellow area.

100% web

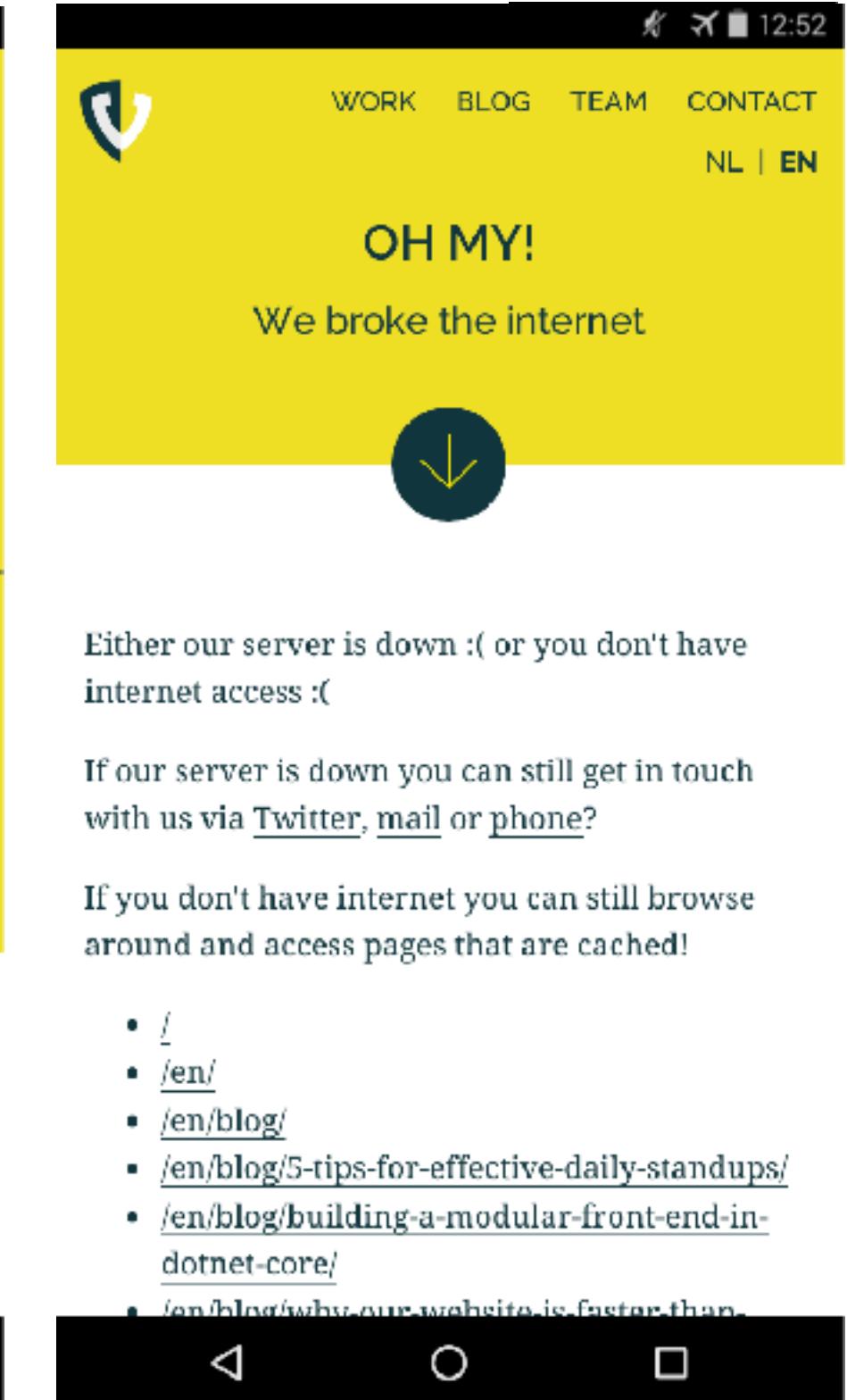
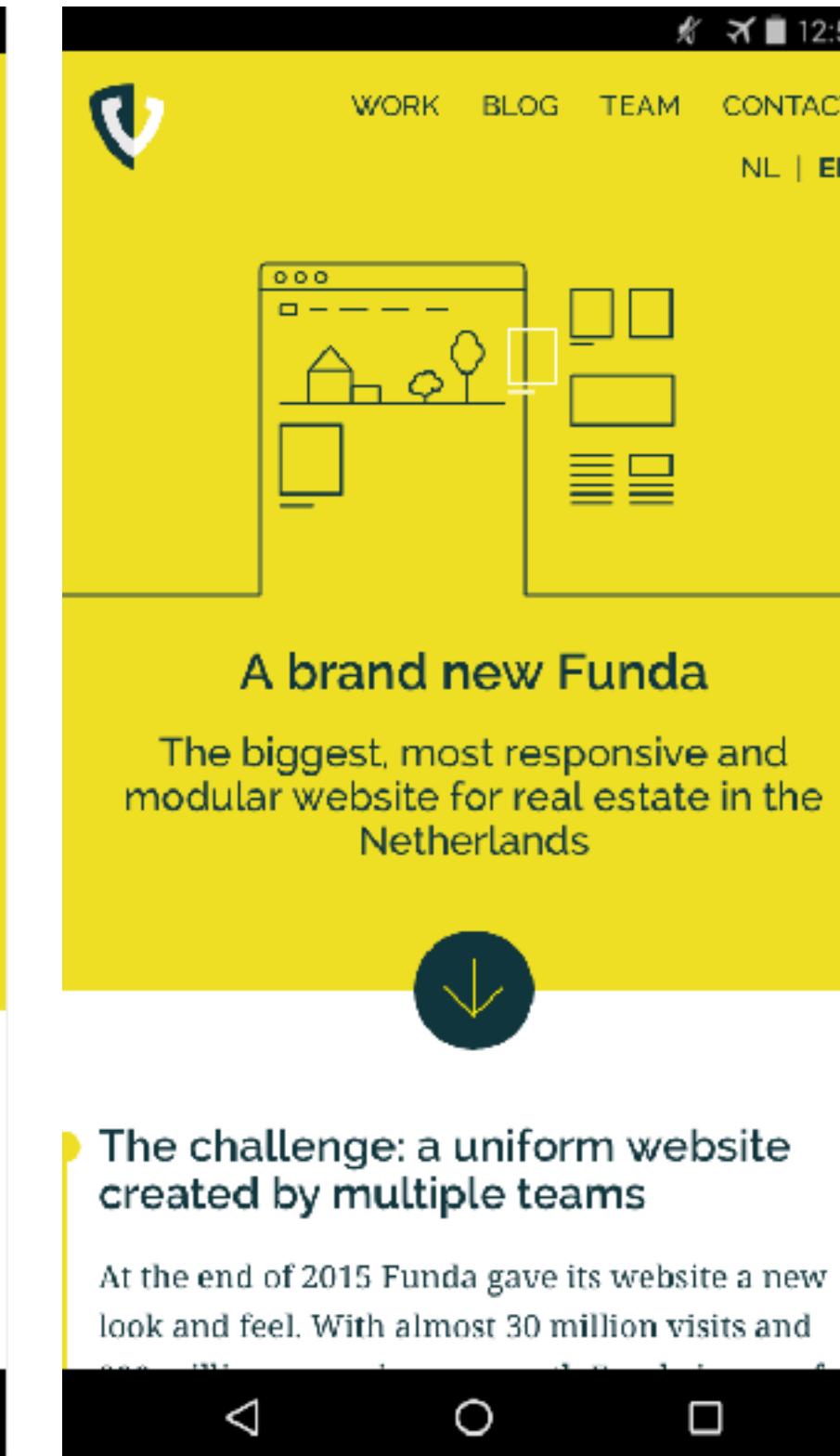
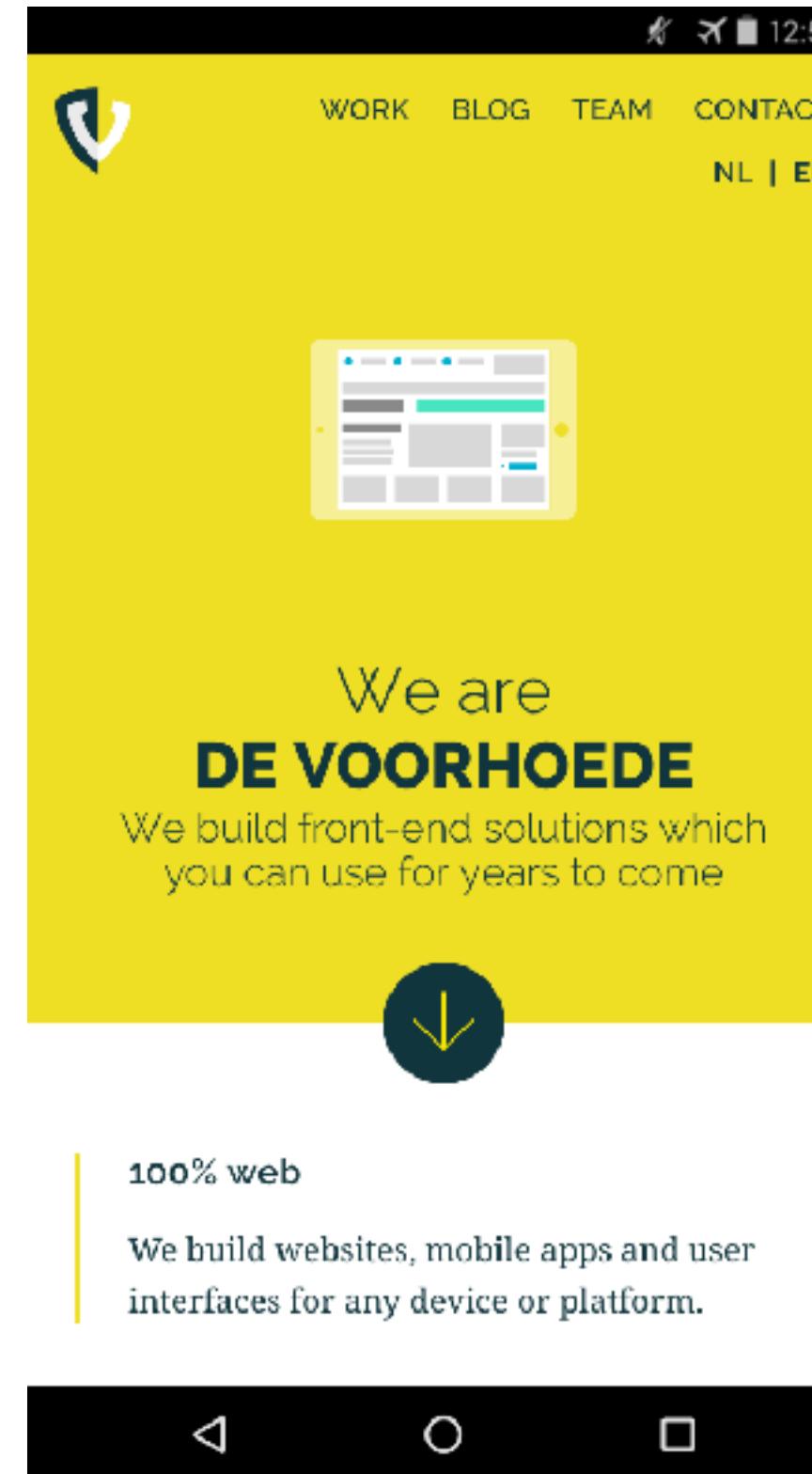
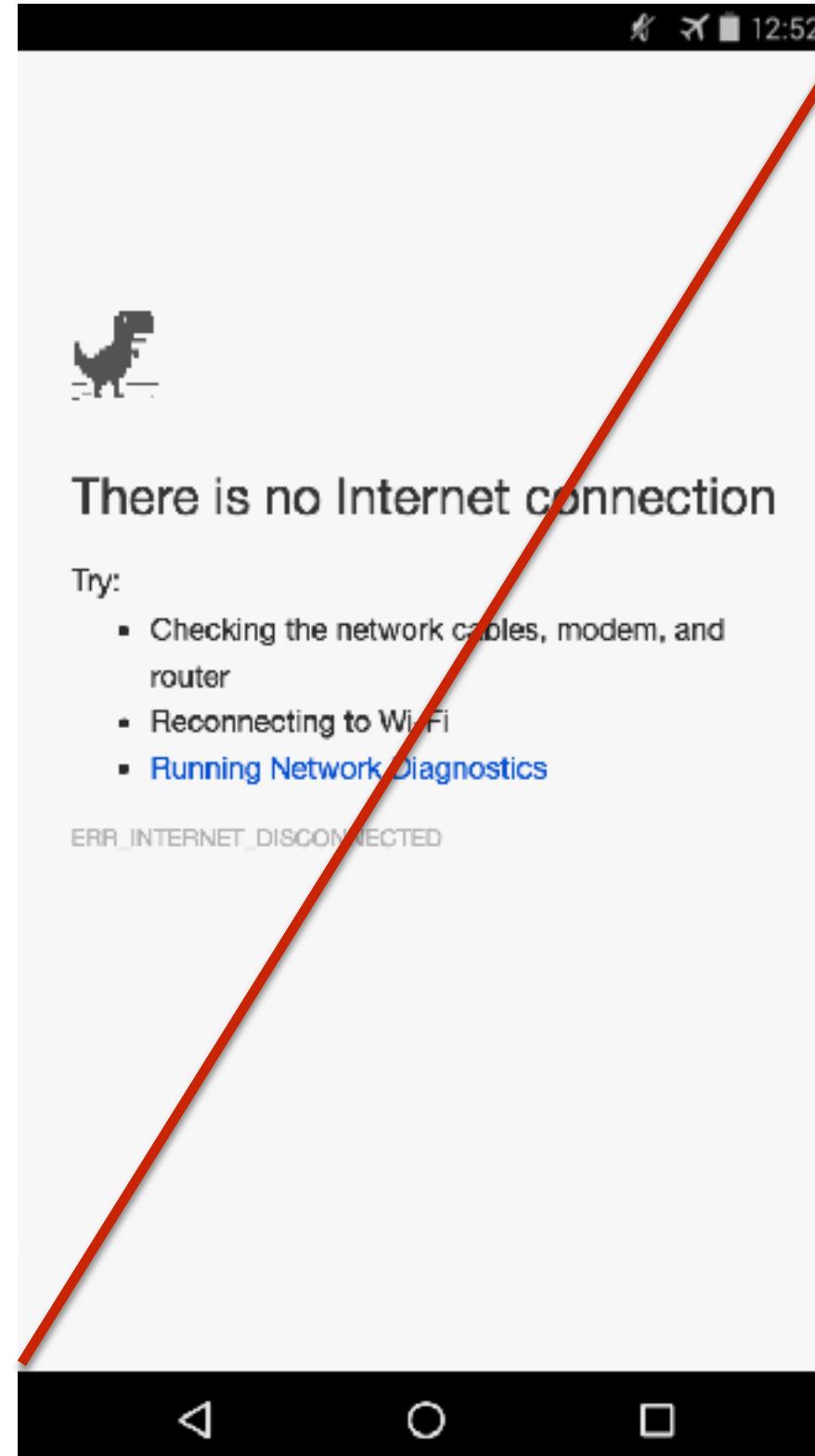
We bouwen websites, mobiele apps en user interfaces voor ieder apparaat en

voorhoede.nl

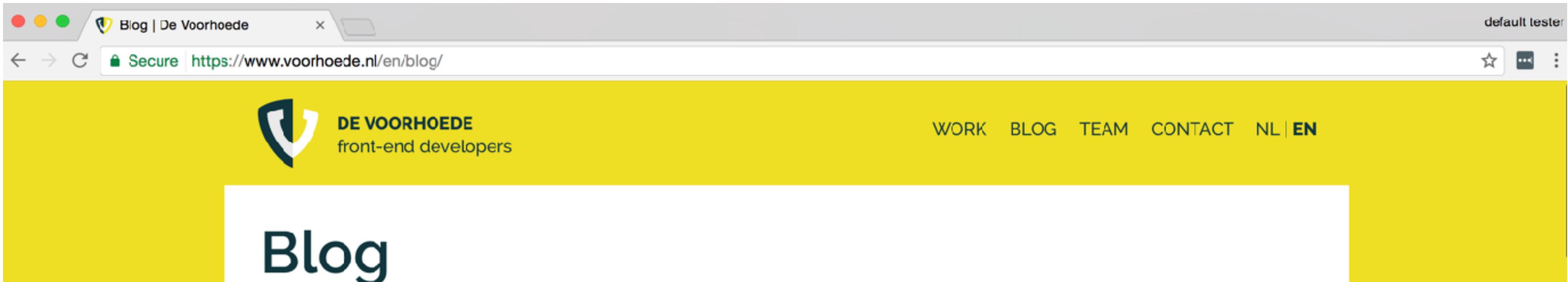
ADD TO HOMESCREEN



OFFLINE SUPPORT



OFFLINE SUPPORT



17 Mar 2017

[Fighting front-end fatigue with tooling recipes](#) ⚡



by Jasper

 available offline

16 Jan 2017

[The importance of an Interaction Engineer](#)



by Tjerk

27 Dec 2016

[Improving accessibility for the blind: 8 guidelines](#)



by Bas

21 Dec 2016

[Creating HTML email templates from scratch](#)



by Vincent

28 Nov 2016

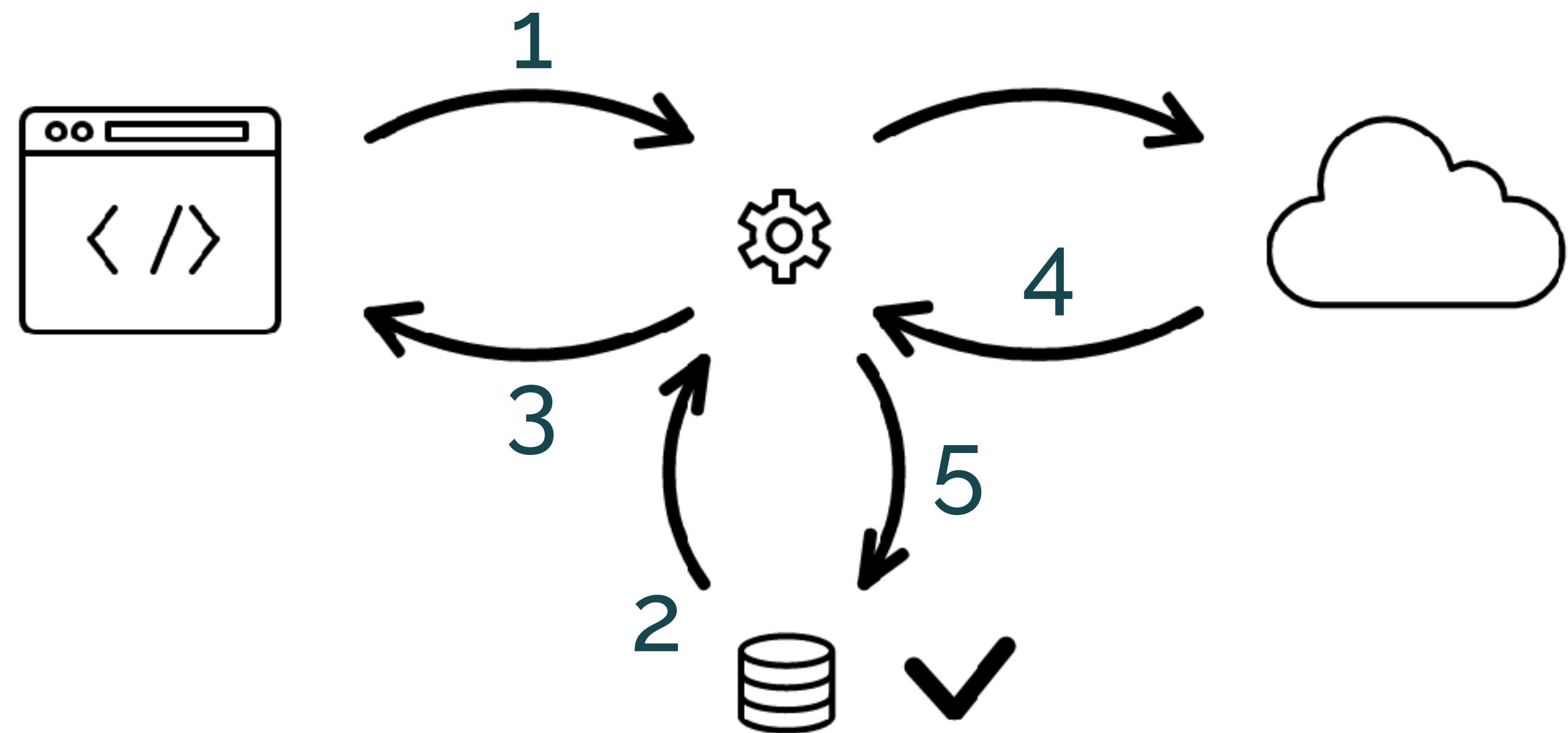
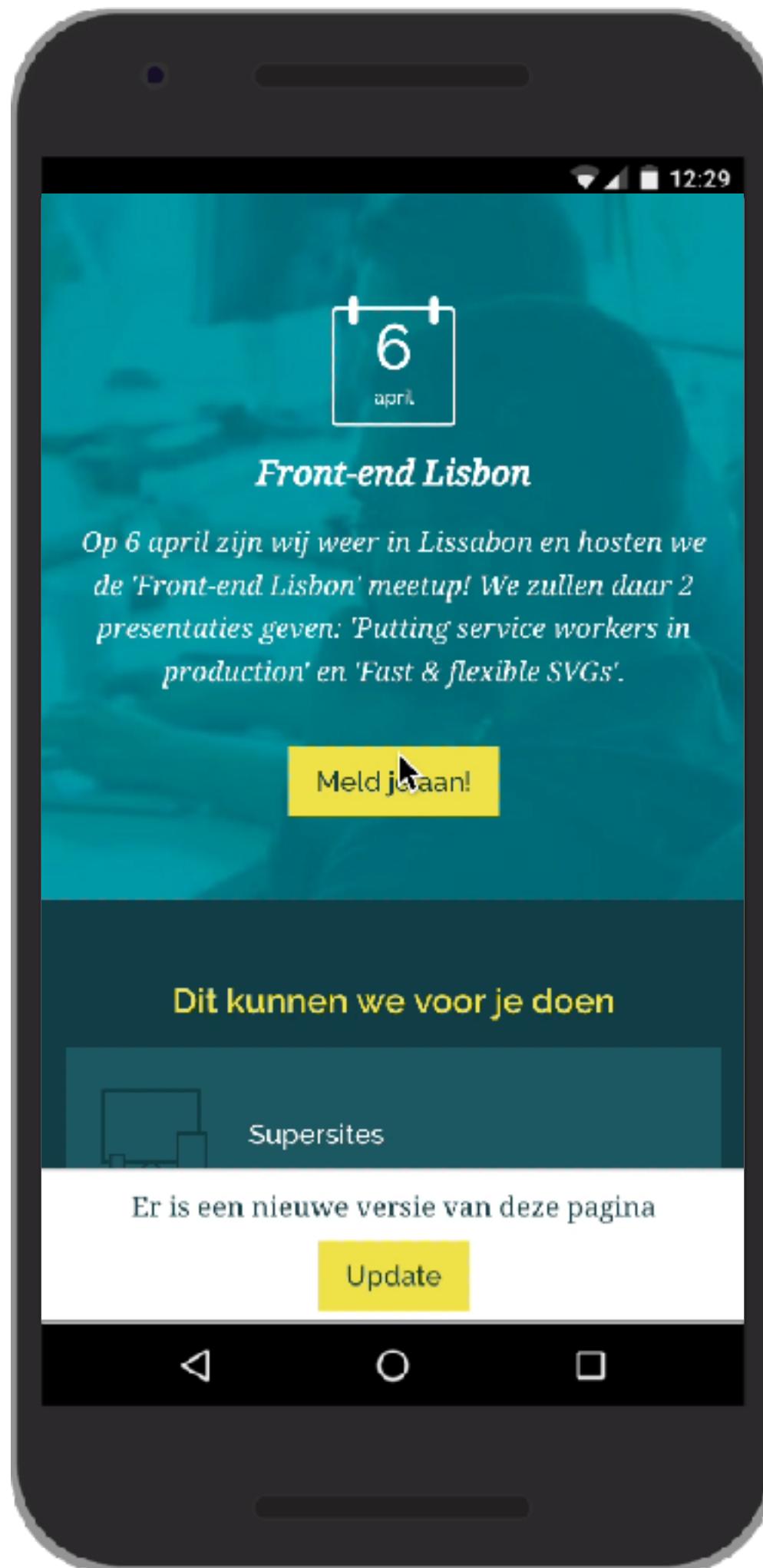
[Static site implosion with Brotli and Gzip](#)



by Thadée

[voorhoede.nl/en/blog/](https://www.voorhoede.nl/en/blog/)

STAY FRESH



STALE WHILE REVALIDATE

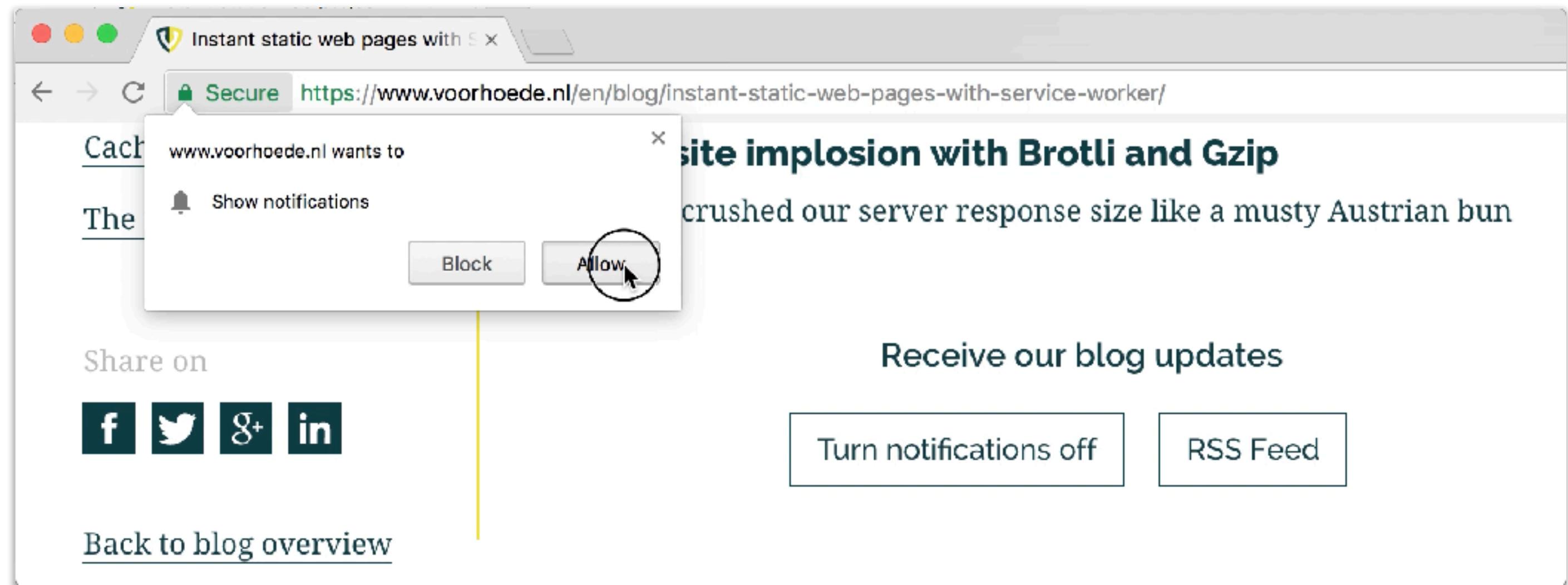
```
self.addEventListener('activate', event => {
  const outdatedCaches = [/* filters `cacheNames` */]
  broadcastOutdatedCaches(outdatedCaches)
  // ...
})

async function broadcastOutdatedCaches (cacheNames) {
  const clientList = await self.clients.matchAll()
  return Promise.all(clientList.map(
    client => client.postMessage({
      type: 'outdated',
      urls: [ /* uses `cacheNames` */ ]
    })))
}
```

STALE WHILE REVALIDATE

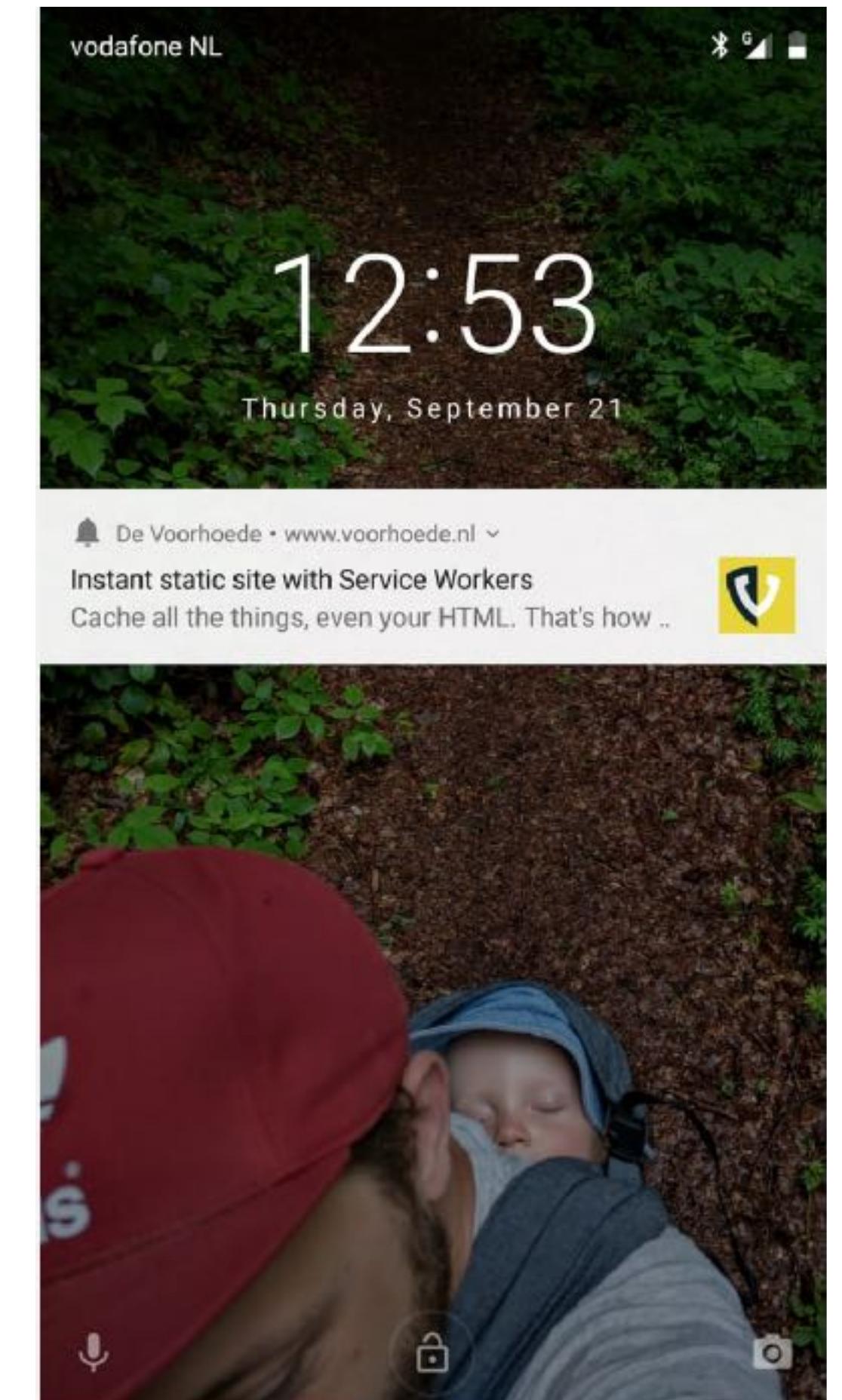
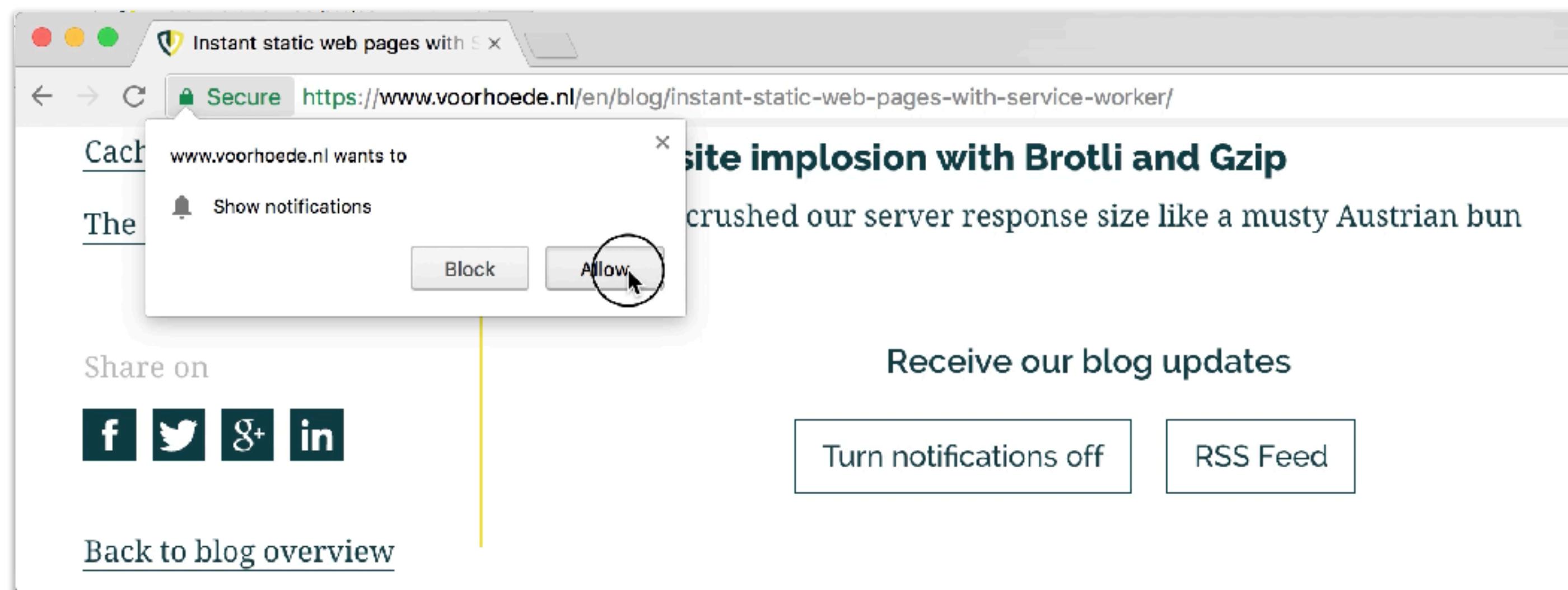
```
// page.html
navigator.serviceWorker
  .addEventListener('message', event => {
    const { type, urls } = event.data
    const currentPageIsOutdated = (
      type === 'outdated'
      && urls.includes(window.location.href)
    )
    if (currentPageIsOutdated) {
      fetchNewPage().then(showMessage)
    }
  })
})
```

PUSH NOTIFICATIONS



voorhoede.nl/en/blog/making-our-site-more-app-y/

PUSH NOTIFICATIONS



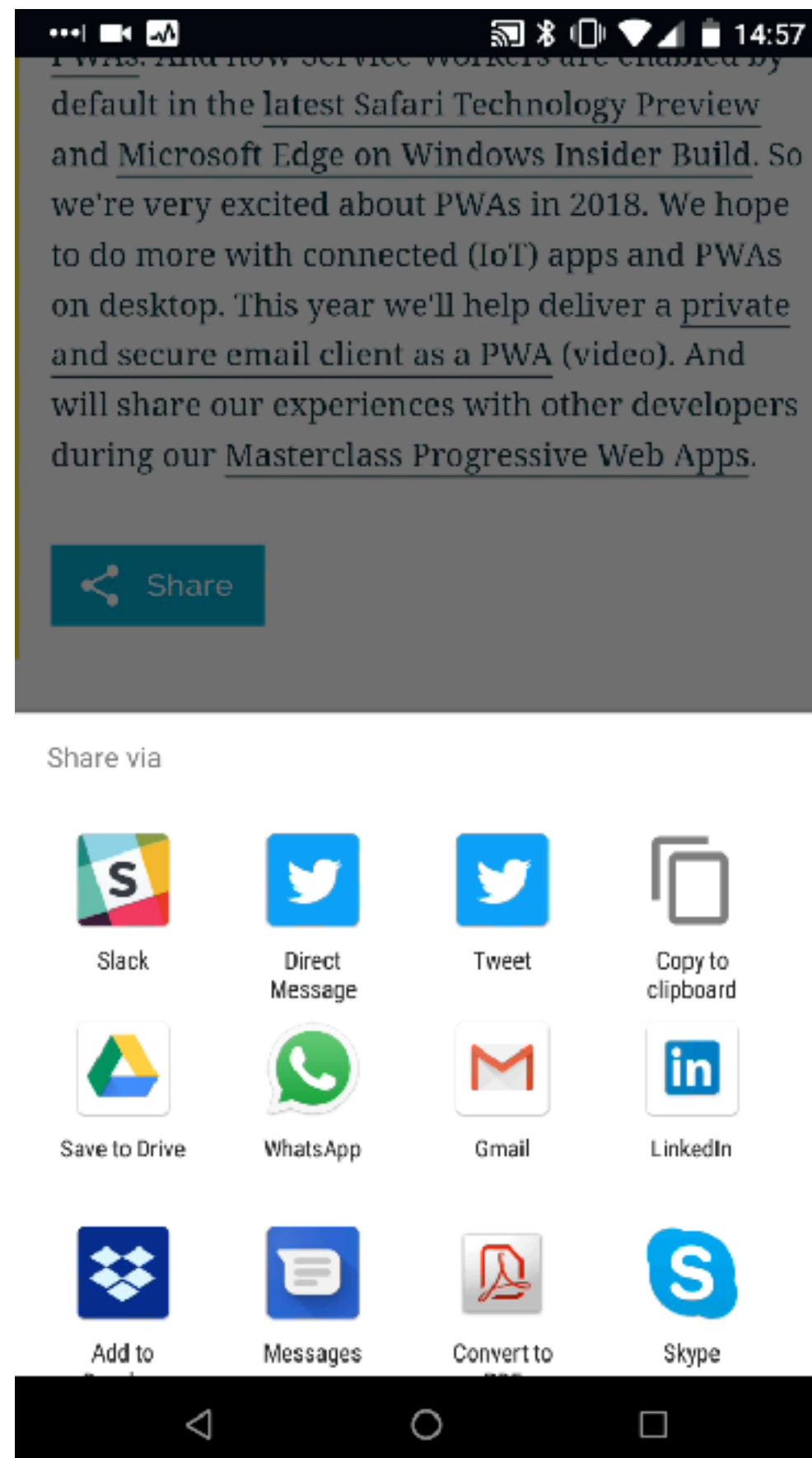
voorhoede.nl/en/blog/making-our-site-more-app-y/

PUSH NOTIFICATIONS

```
// page.html
subscribeButton.addEventListener('click', () => {
  subscribeUser()
  hideSubscribeButtons()
  showUnsubscribeButtons()
}, false)

function subscribeUser() {
  OneSignal.push(() => {
    OneSignal.registerForPushNotifications()
    OneSignal.setSubscription(true)
  })
}
```

NATIVE SHARE



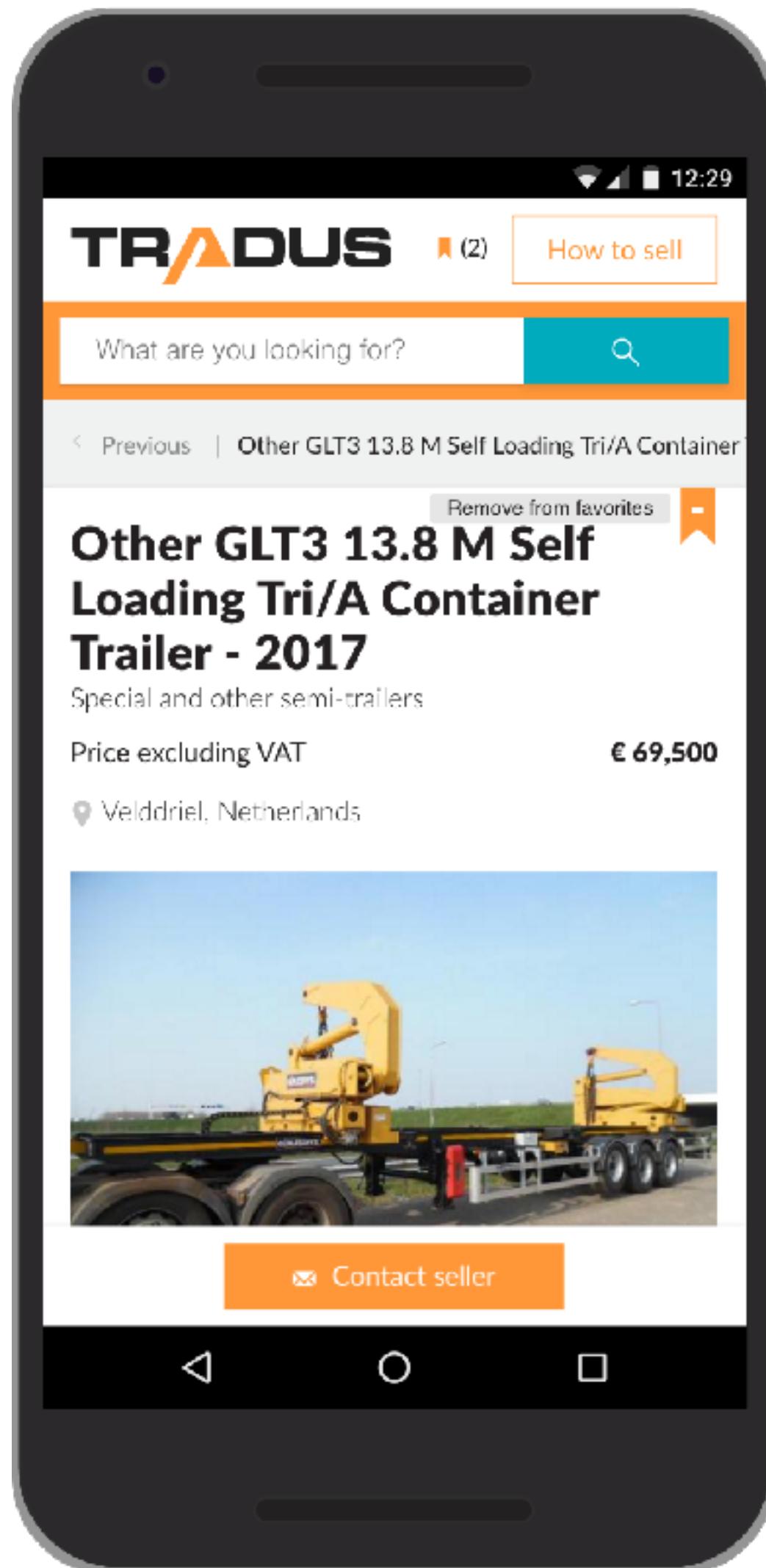
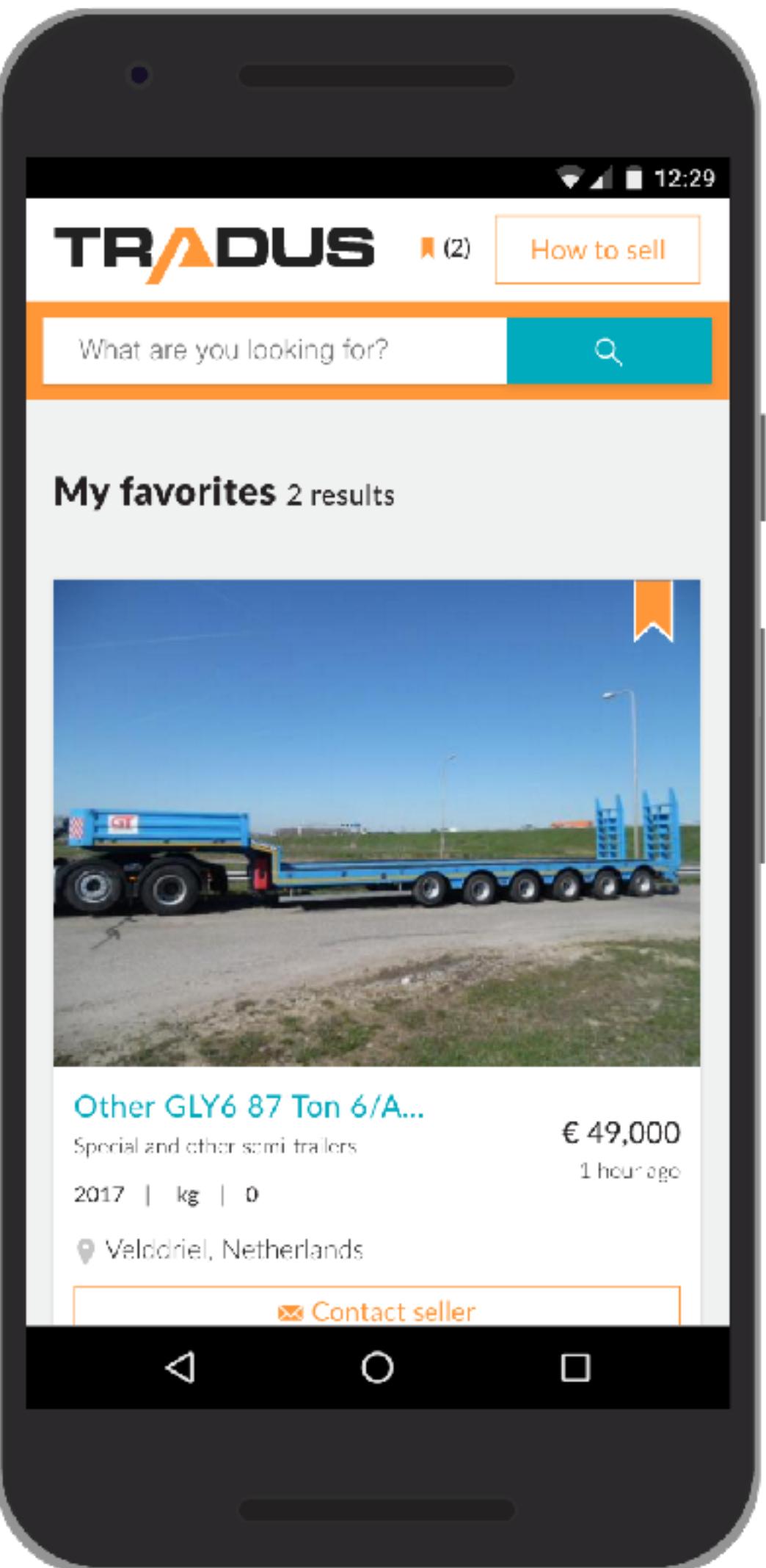
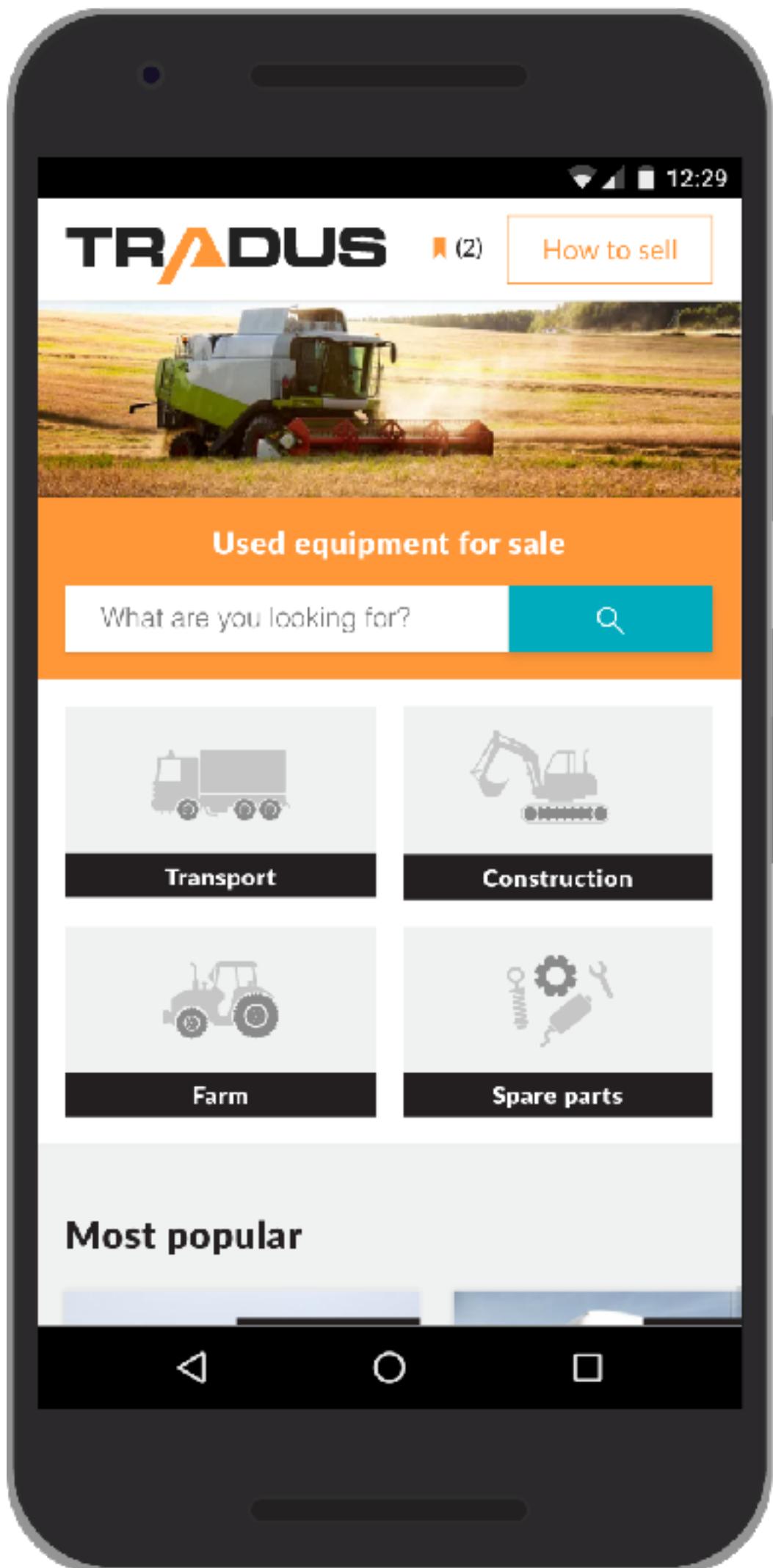
A screenshot of a desktop browser window. The title bar shows three colored dots (red, yellow, green) and the text 'Modular build tasks with Grunt'. To the right of the title bar is a button labeled 'Share a link on Twitter'. Below the title bar is a toolbar with back, forward, and refresh buttons. The main content area shows a secure connection indicator ('Secure') and the URL 'https://www.voorhoeede.nl/en/blog/modular-build'. Below the URL is a large, light gray button with the text 'Share on' in a large, sans-serif font. To the right of this button is a vertical yellow line. Below the 'Share on' button are four dark blue square icons with white social media logos: Facebook (f), Twitter, Google+ (g+), and LinkedIn (in). At the bottom of the content area is a dark blue footer bar with a white 'Back to blog overview' link.

[voorhoeede.nl/en/blog/making-our-site-more-app-y/](https://www.voorhoeede.nl/en/blog/making-our-site-more-app-y/)

NATIVE SHARE

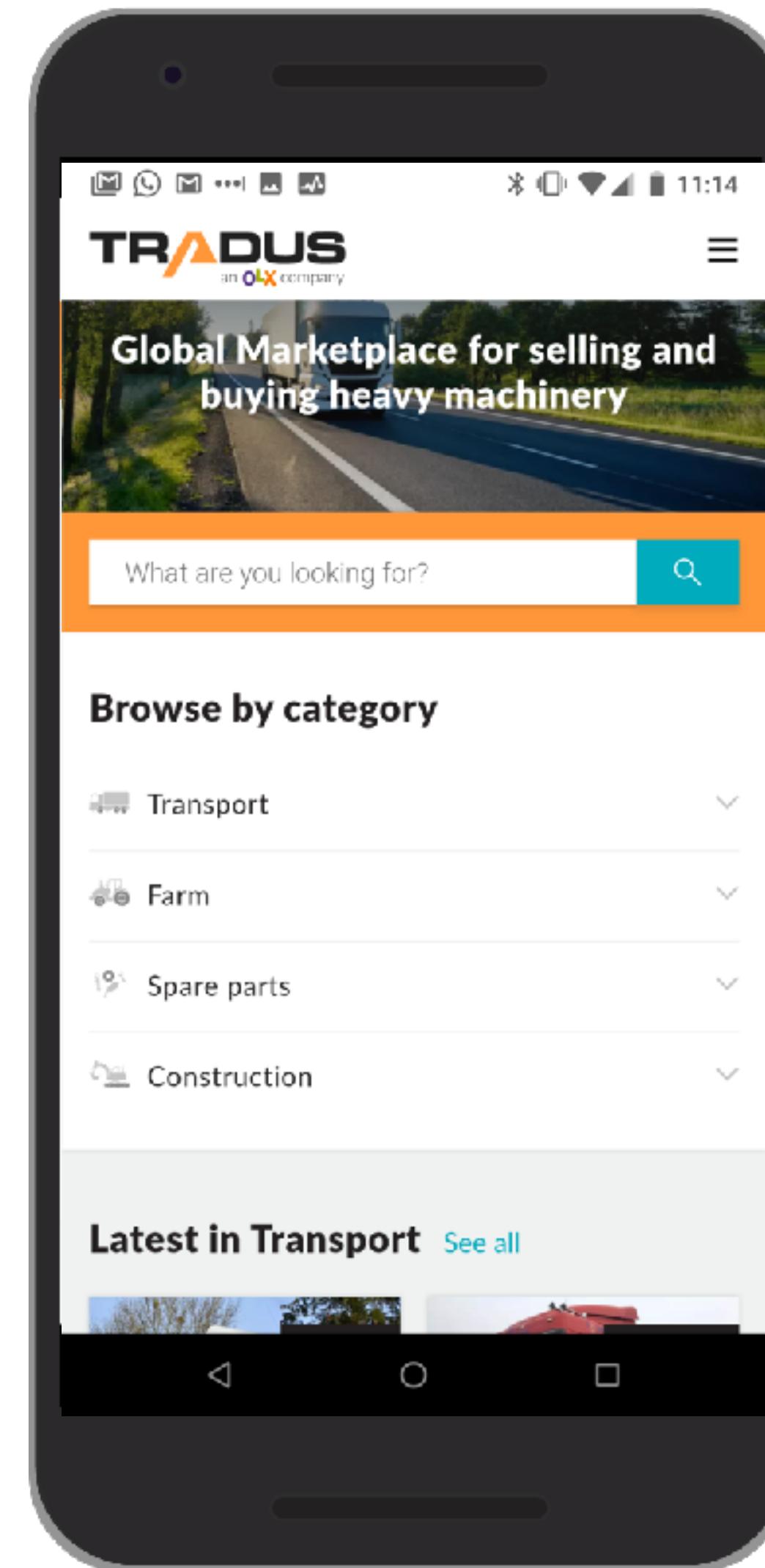
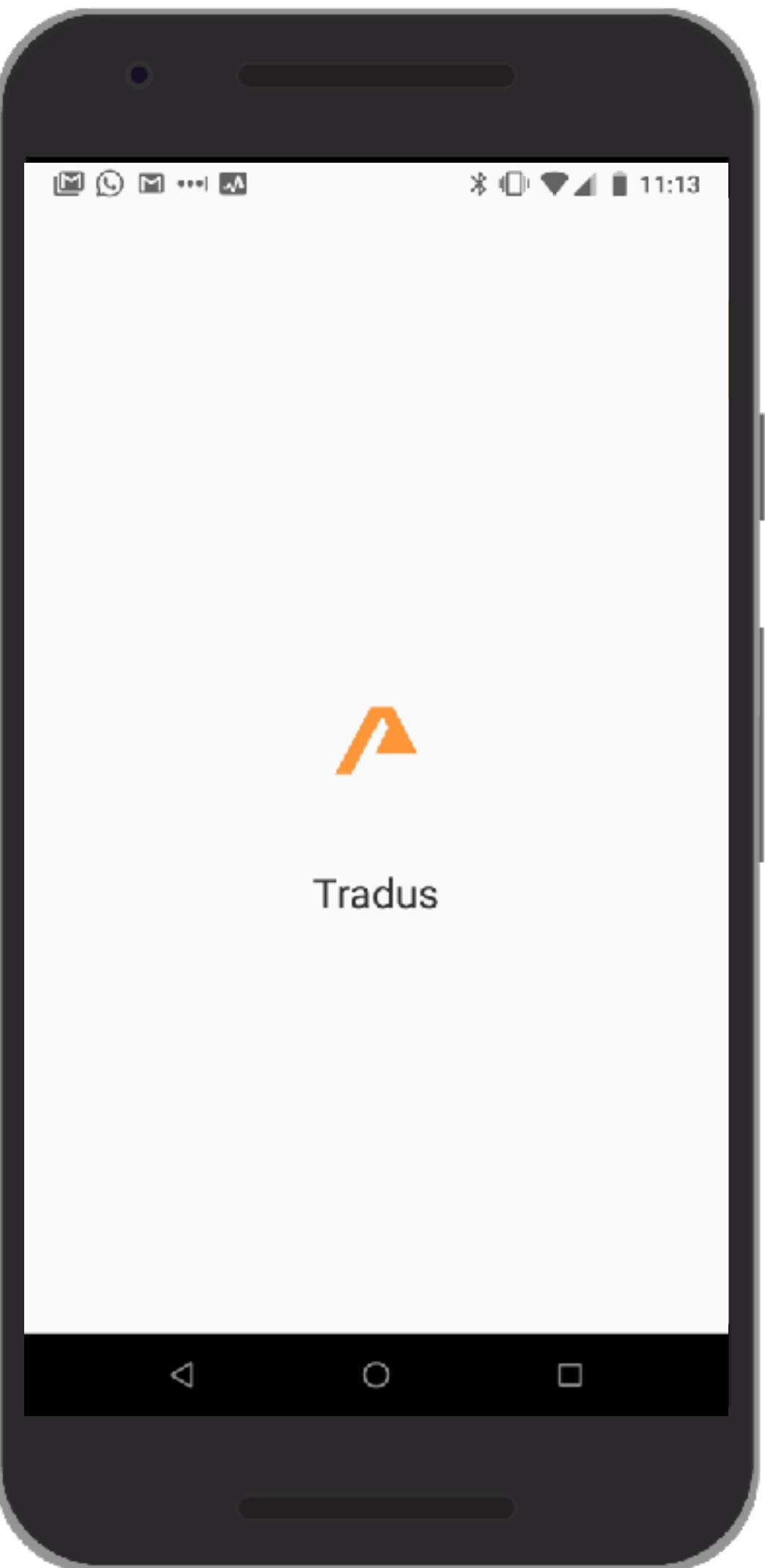
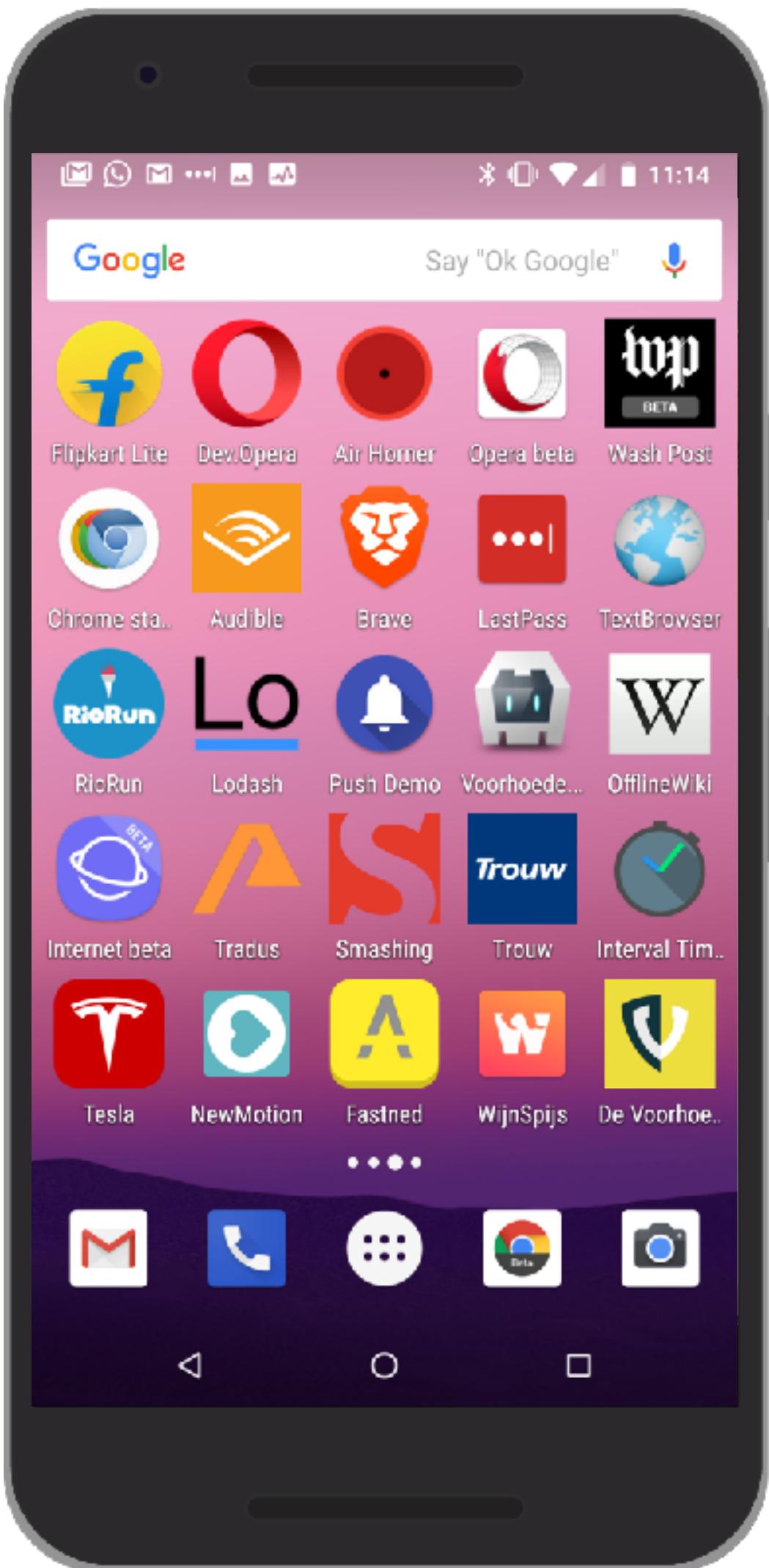
```
// page.html
if ('share' in navigator) {
  const { title, url } = getShareParams()
  const button = showShareButton()
  button.addEventListener('click', () => {
    navigator.share({ title, url })
  })
  hideShareLinks()
}
```

EXAMPLE: TRADUS.COM



tradus.com

EXAMPLE: TRADUS.COM



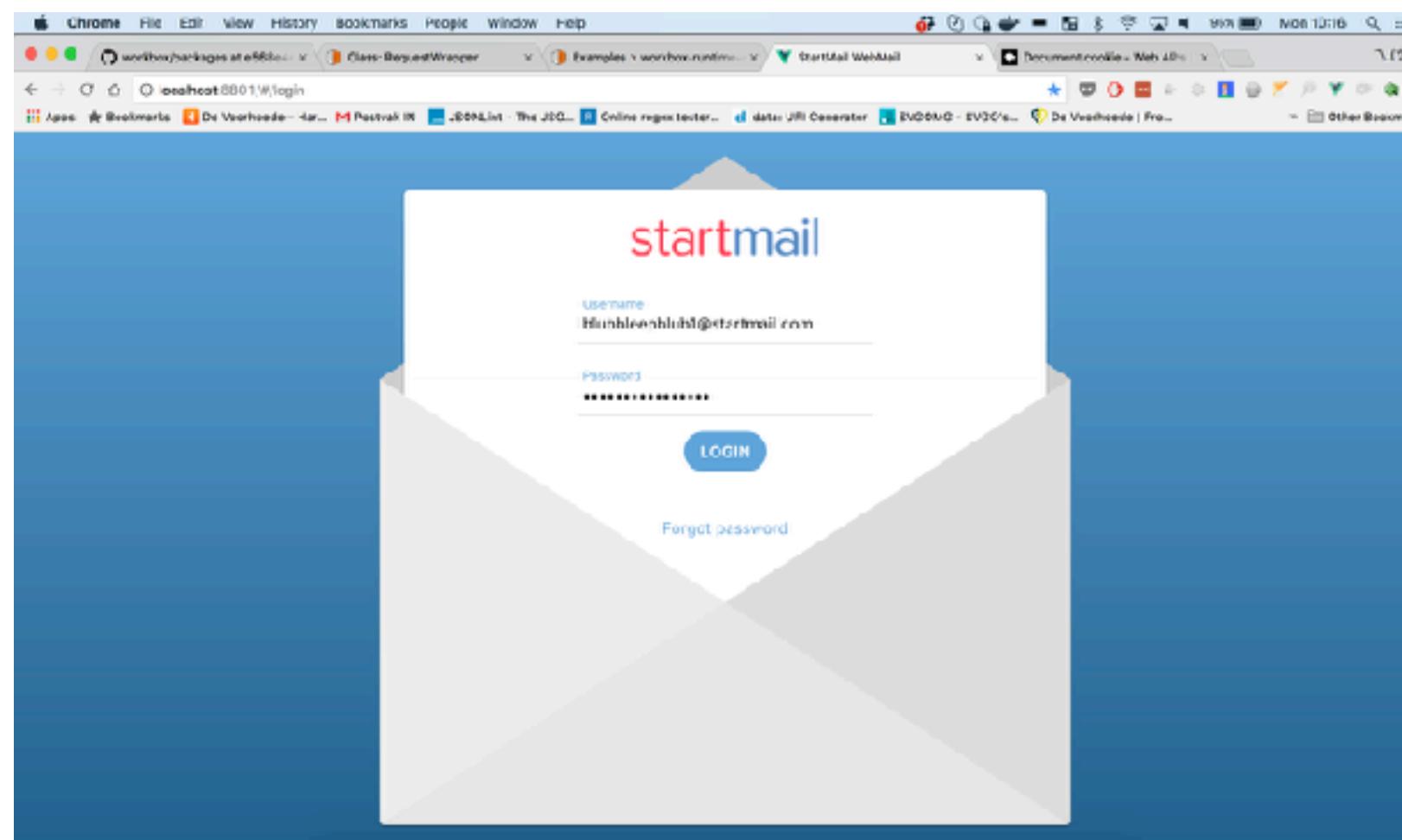
tradus.com

WEB APP MANIFEST

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

{
  "name": "Tradus\nRynek maszyn ciężkich",
  "short_name": "Tradus",
  "lang": "pl",
  "display": "standalone",
  "start_url": "/?homescreen=1",
  "background_color": "#5E4485",
  "theme_color": "#463269",
  "icons": [
    {
      "src": "/tradus-v1.png",
      "sizes": "256x256",
      "type": "image/png"
    }
  ]
}
```

EXAMPLE: STARTMAIL WEB CLIENT

A screenshot of a web browser showing the StartMail inbox. The URL in the address bar is 'localhost:8801/#/mail/czt8jkw8j0hjdsd3hxd75x2vu'. The interface includes a sidebar with links for 'Inbox', 'Archive', 'Drafts', 'Sent', 'Spam', 'Trash', 'Fortunes', and 'Notification emails'. The main area displays a list of messages under the heading 'Older'.

Message	From	Date
Demo today	Tijn Schmits	Last Thursday
Fwd: [SECURITY] [DSA 3944-1] mariadb-10.0 security update	Tijn Schmits	Last Thursday
Fwd: [USN-3392-1] Linux kernel regression	Tijn Schmits	Last Wednesday
Hey, hoe gaat het ermee?	Tijn Schmits	Aug 07, 2017
Sprint Review Today	Tijn Schmits	Jul 27, 2017
HTML in a text message	Aaryn Tonita	Jul 13, 2017
blubleepblub1@startmail.com	Aaryn Tonita	Jul 12, 2017
Re: Reminder to self: Finish NFE	Tijn Schmits	Jun 19, 2017
Test selfsending 01	StartMail Test	Jun 16, 2017
Re: Testing	StartMail Test	Jun 09, 2017

ENCRYPTION IN SW

```
async function fetchAndCache(request) {  
  const response = await fetch(request.url, {  
    headers: new Headers(),  
    credentials: 'include'  
  })  
  const cryptoKey = response.headers.get('crypto-key')  
  const encryptedResponse =  
    await encryptResponse(response.clone(), cryptoKey)  
  
  const cache = await caches.open('encrypted-emails')  
  cache.put(request, encryptedResponse.clone())  
  
  return response  
}
```

ENCRYPTION IN SW

```
async function encryptResponse(response, cryptoKey) {  
  const body = await response.text()  
  const encryptedBody = encrypt(body, cryptoKey)  
  const options = {  
    status: response.status,  
    statusText: response.statusText,  
    headers: {'content-type': 'application/json'}  
  }  
  
  return new Response(  
    JSON.stringify(encryptedBody),  
    options  
  )  
}
```

ENCRYPTION IN SW

```
async function encryptResponse(response, cryptoKey) {  
  const body = await response.text()  
  const encryptedBody = encrypt(body, cryptoKey)  
  const options = {  
    status: response.status,  
    statusText: response.statusText,  
    headers: {'content-type': 'application/json'}  
  }  
  
  return new Response(  
    JSON.stringify(encryptedBody),  
    options  
  )  
}
```

ENCRYPTION IN SW

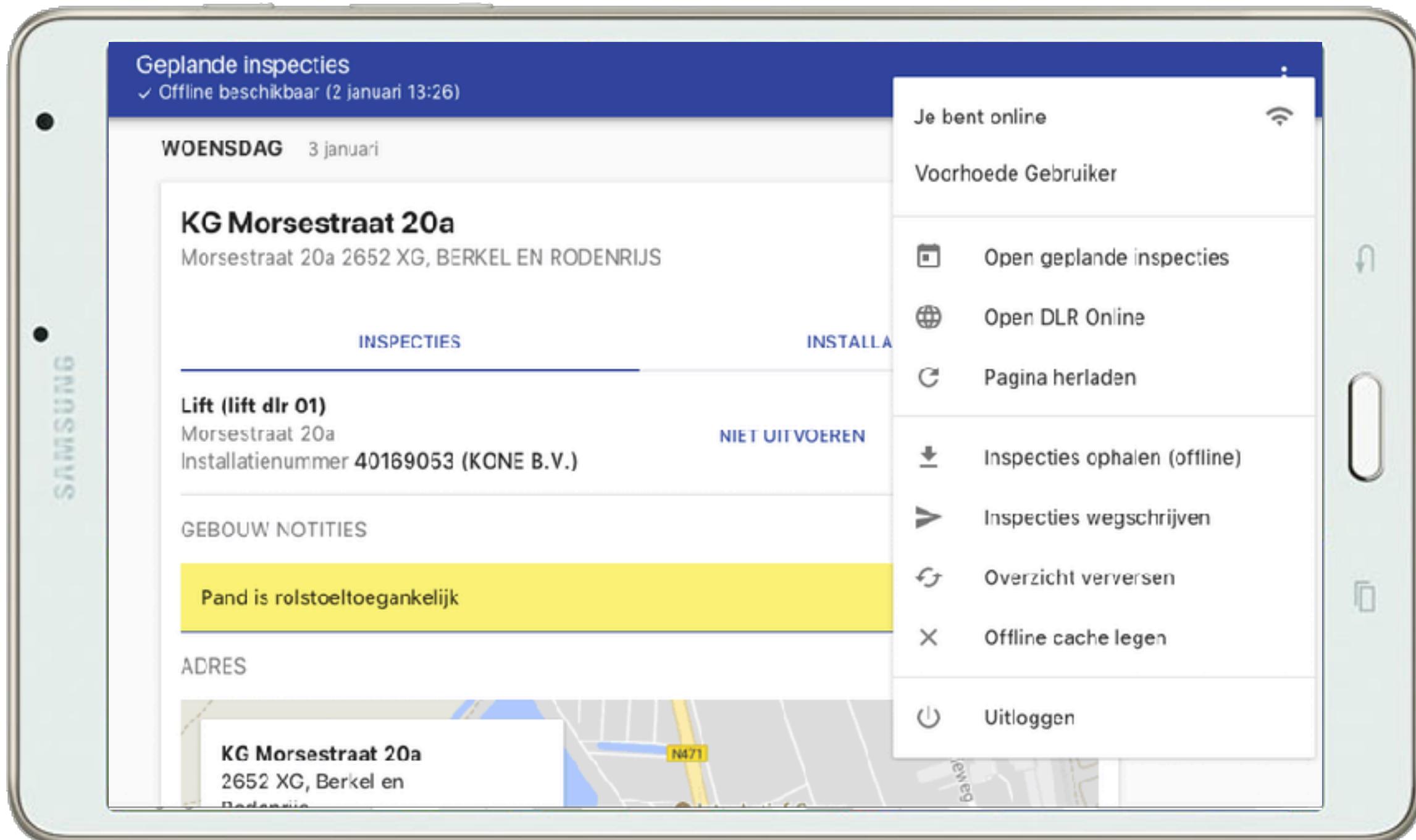
message before encryption

```
'{"account_id":"dw8ulo7jflxelumie5wt8dalt","bcc":[],"body":"<p>Hi,</p>\n<p>Good luck with the demo again today!</p>\n<p>Kind regards,<br>Tijn</p>","cc":[],"date":1502970512,"events":[],"files":[],"folder": {"display_name":"INBOX","id":"ezt8jxw8j0hjdse3hxd75x2vu","name":"inbox"}, "from": [{"email":"tijn@startmail.com","name":"Tijn Schmits"}],"id":"av3tjfm8xvmzzpaio9r2u0zzu","object":"message","reply_to":[],"snippet":"Hi, Good luck with the demo again today! Kind regards, Tijn","starred":false,"subject":"Demo today","thread_id":"2kbj6ob1pke0g8f04j62z7e9g","to":[{"email":"blubbleepblub1@startmail.com","name":""}], "unread":false}'
```

message after encryption

```
{  
  cipher_text: '2W9UmQxx2FUHG..',  
  iv: 'vomfUHS2AZrBr6XyTmM2mw..',  
  salt: 'YDX/0pAtvbXpe3WAVALX..'  
}
```

EXAMPLE: ELEVATOR INSPECTION APP



BACKGROUND SYNC

```
self.addEventListener('fetch', (event) => {
  event.respondWith(
    fetch(event.request)
      .catch(err => queueRequest(event.request))
  )
})

self.addEventListener('sync', (event) => {
  event.waitUntil(
    syncQueuedRequests().then(notifyUser)
  )
})
```

TAKEAWAYS



PWAs

growing browser support

start small

include design

RESOURCES

- **PWA Stats** success stories of PWAs with statistics.
- **Awesome PWAs** curated collection of PWA resources.
- **Awesome Service Workers** collection of SW resources.
- **Offline cookbook** by Jake Archibald.
- **Service Worker cookbook** by Mozilla.
- **PWA libraries in Production** by Addy Osmani.
- **Testing Service Workers** by Matt Gaunt.

RESOURCES

developers.google.com/web/tools/workbox/

Cache JavaScript and CSS

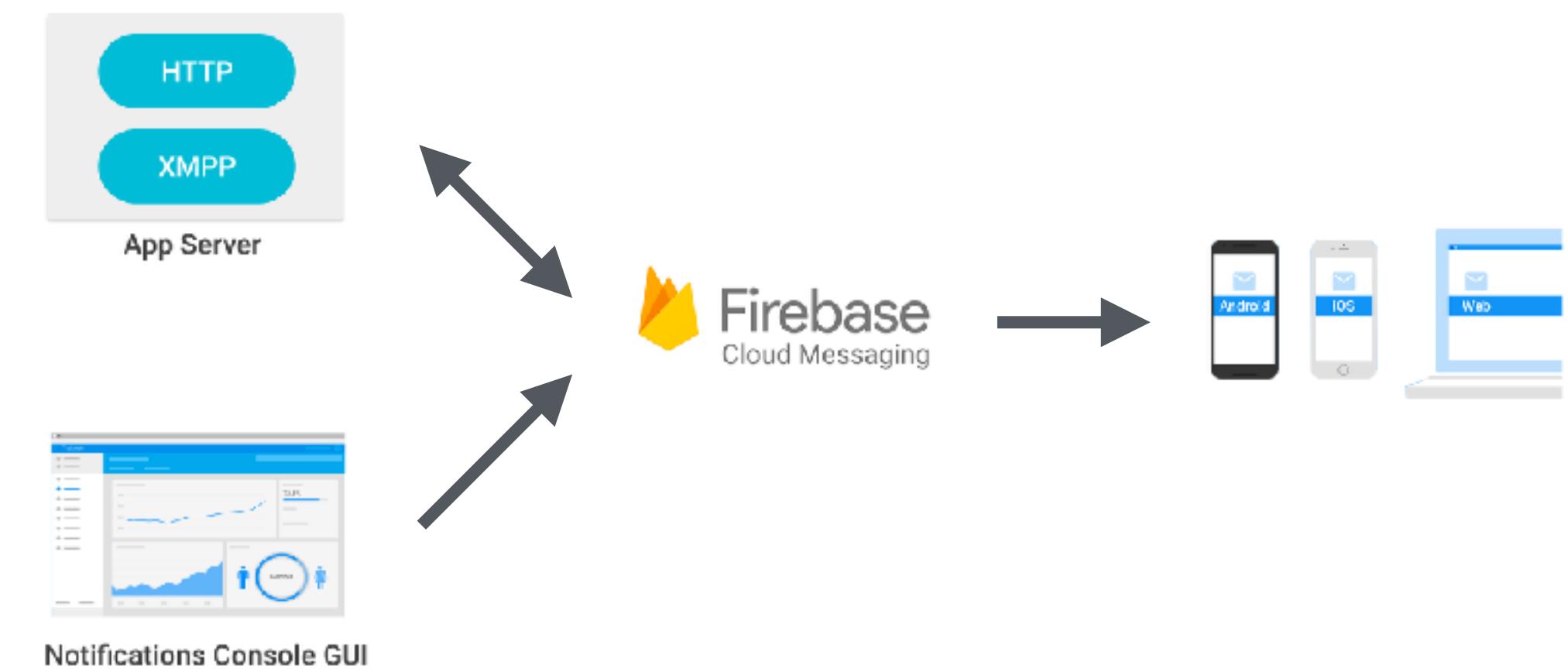
Make your JS and CSS fast by returning the assets from the cache while making sure they are updated in the background for the next use.

```
workbox.routing.registerRoute(
  '/(.?:js|css)$',
  workbox.strategies.staleWhilevalidate(),
);
```

Cache Images

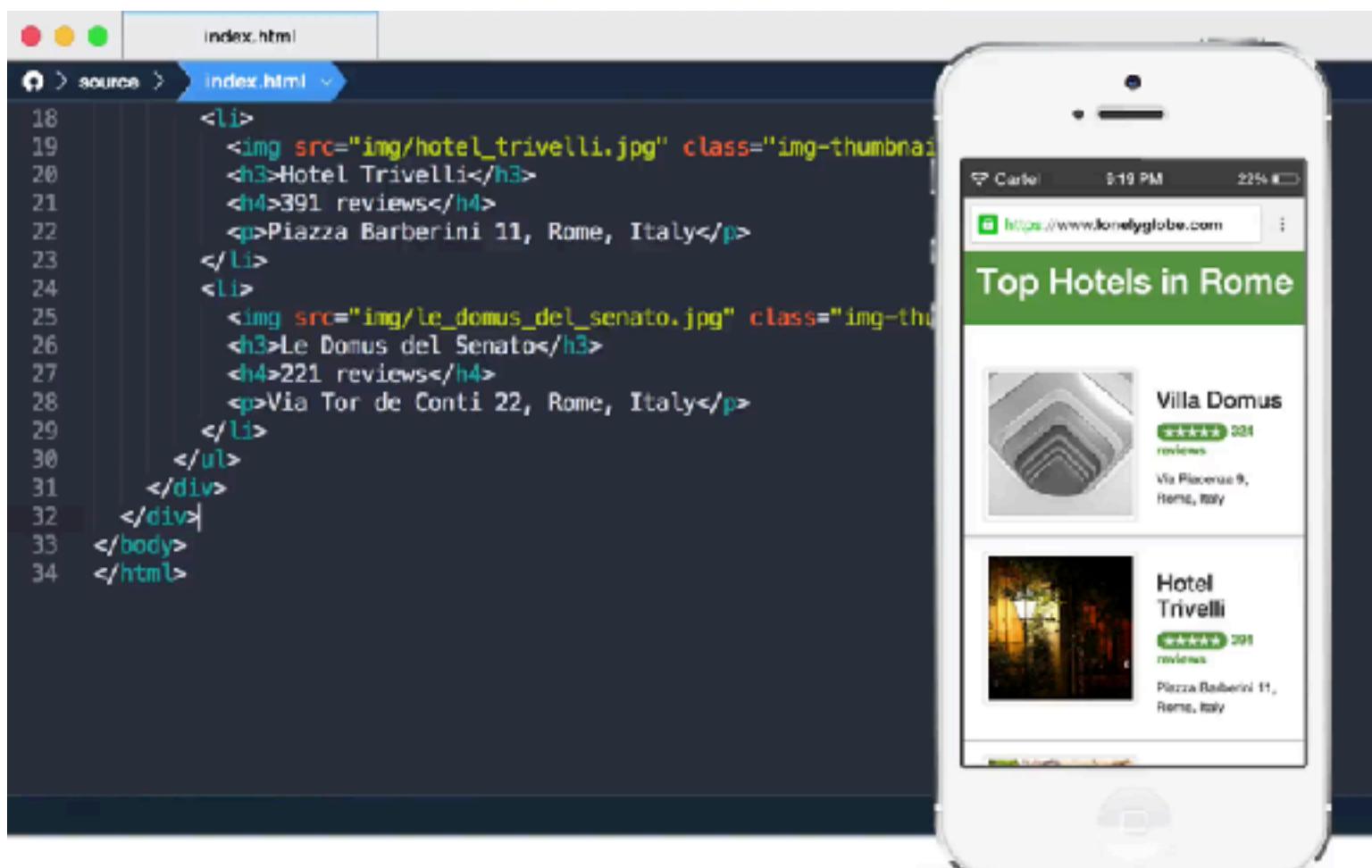
Images carry most of the weight for a web page. Use this rule to serve them quickly from the cache, while making sure you don't cache them indefinitely, consuming your users' storage.

```
workbox.routing.registerRoute(
  /\.(?:png|gif|jpg|jpeg|svg)$/,
  workbox.strategies.cacheFirst({
    cacheName: 'images',
    plugins: [
      new workbox.expiration.Plugin({
        maxEntries: 60,
        maxAgeSeconds: 30 + 24 + 60 + 60, // 30 Days
      }),
    ],
  });
);
```

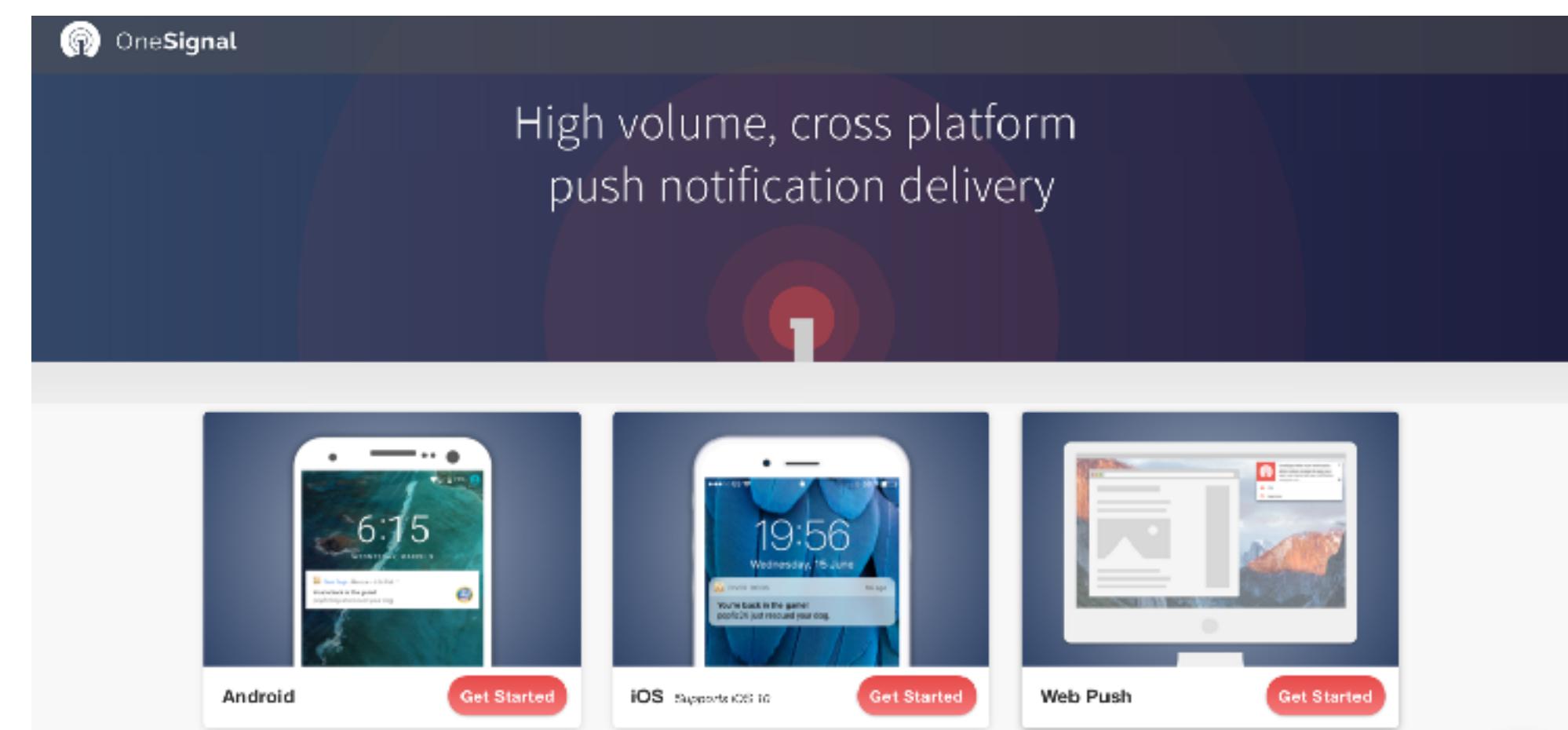


developers.google.com/web/tools/workbox/

firebase.google.com/docs/cloud-messaging/

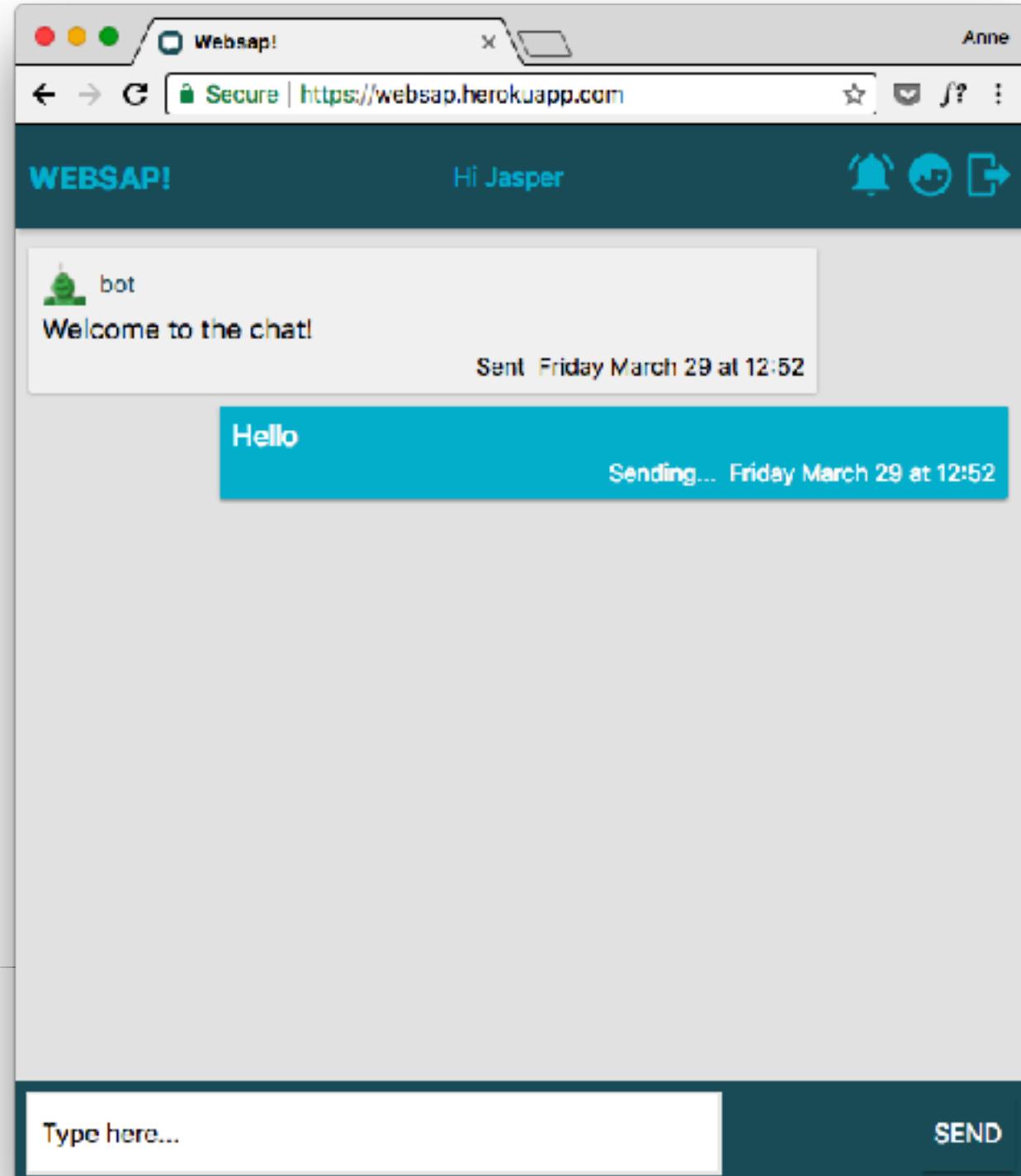


www.talater.com/upup/



onesignal.com

PWA MASTERCLASS



Friday 14 September 2018
use **CABFAB** for **20% discount**

goo.gl/kW7qLH

THANK YOU





DE VOORHOEDE

front-end developers