

PROGRESSIVE WEB APP

Mehmet Güngörmez
Florian Paul Bäumler



AGENDA

PROBLEMSTELLUNG

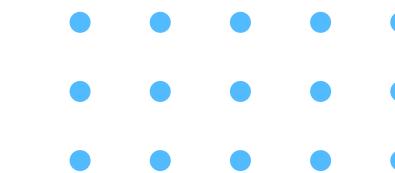
PROGRESSIVE WEB APP

Die Anatomie einer PWA

THEORIE VS. PRAXIS

IMPLEMENTIERUNG

DEMO



PROBLEMSTELLUNG

-
-
-
-
-
-
-
-
-
-

PROBLEMSTELLUNG

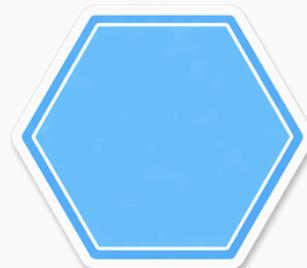
Klassische Webanwendungen vs. Native Apps



- • • • •
- • • • •
- • • • •

Klassische Webanwendungen

- Plattformunabhängig
 - Zugriff über den Browser
 - Keine Installation erforderlich
- Einschränkungen:
- Begrenzte Offline-Funktionalität
 - Eingeschränkter Zugriff auf Gerätefunktionen
 - Geringere Performance bei komplexen Anwendungen
 - Keine direkten Nutzerinteraktionen



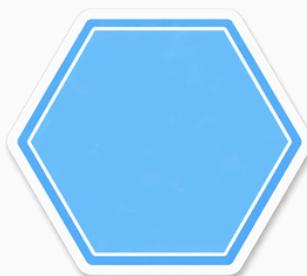
- • • • •
- • • • •
- • • • •

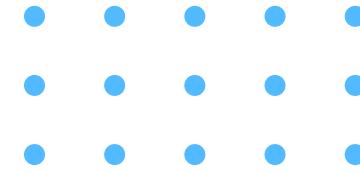
NATIVE APPS

- Hohe Performance
- Tiefe Integration ins Betriebssystem
- Vollständiger Zugriff auf Gerätefunktionen
- Gute Nutzererfahrung

-Nachteile:

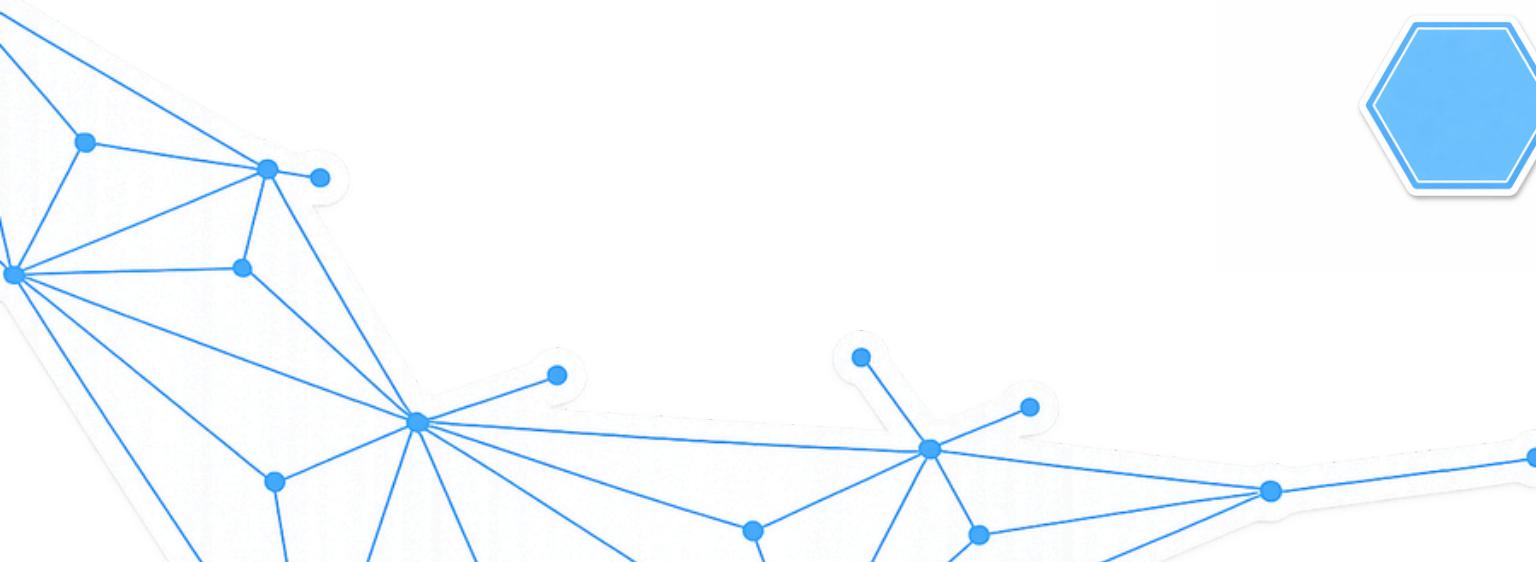
- Hoher Entwicklungs- und Wartungsaufwand
- Separate Apps für iOS und Android
- Abhängigkeit von App-Stores (Freigaben, Updates, Gebühren)

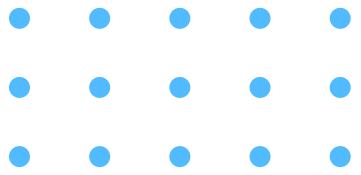




DIE LÖSUNG

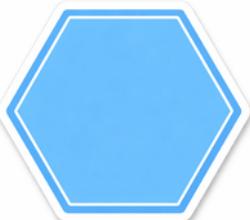
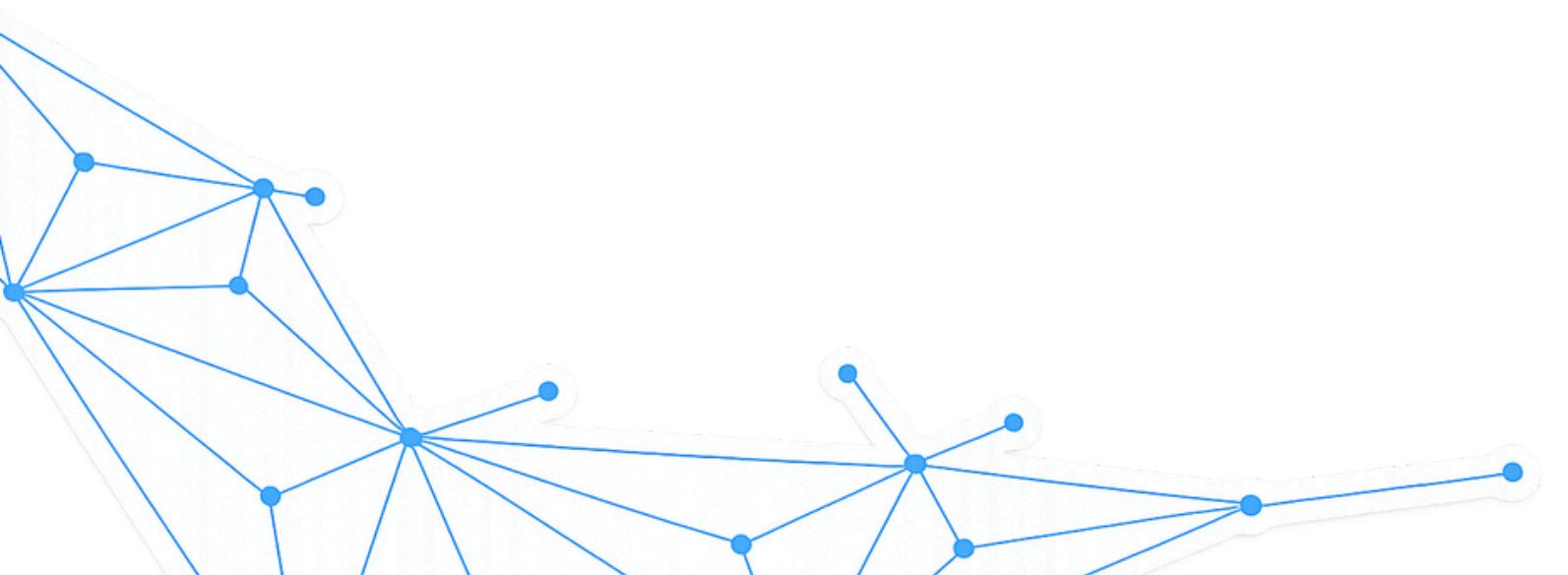
Progressive Web App





Progressive Web App

- Webanwendungen
- Moderne Webstandards
- App-ähnliche Funktionen bieten
- Nutzung über den Browser
- Optionale Installation



Historische Einordnung



Webanwendungen

1990 / 2000

...

2007

...

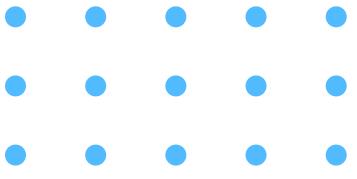
2015

2026

Progressive Web App

Native Apps





Warum Überhaupt PWA?

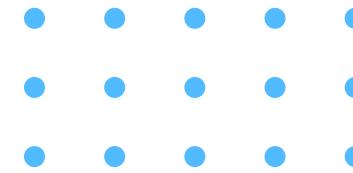


- Plattformunabhängig
- Kein APP Store
- Offline-Ready
- Speicherplatz



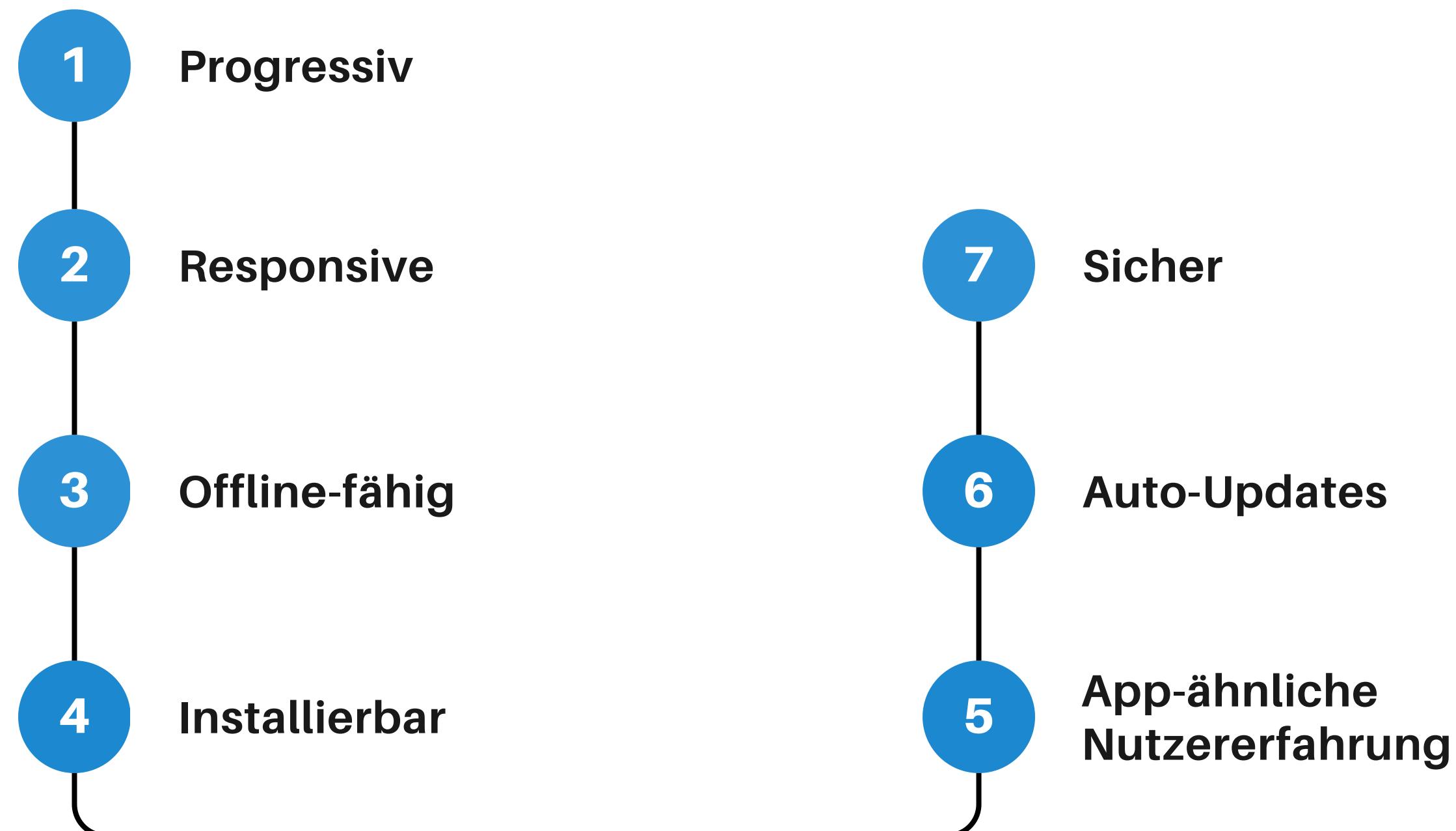
- Eingeschränkter Zugriff
- iOS-Support
- Performance

Die Anatomie einer
PWA



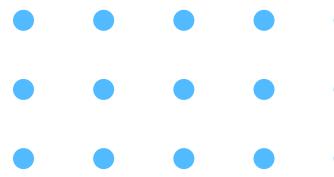
Die Anatomie einer PWA

Zentrale Eigenschaften von PWA



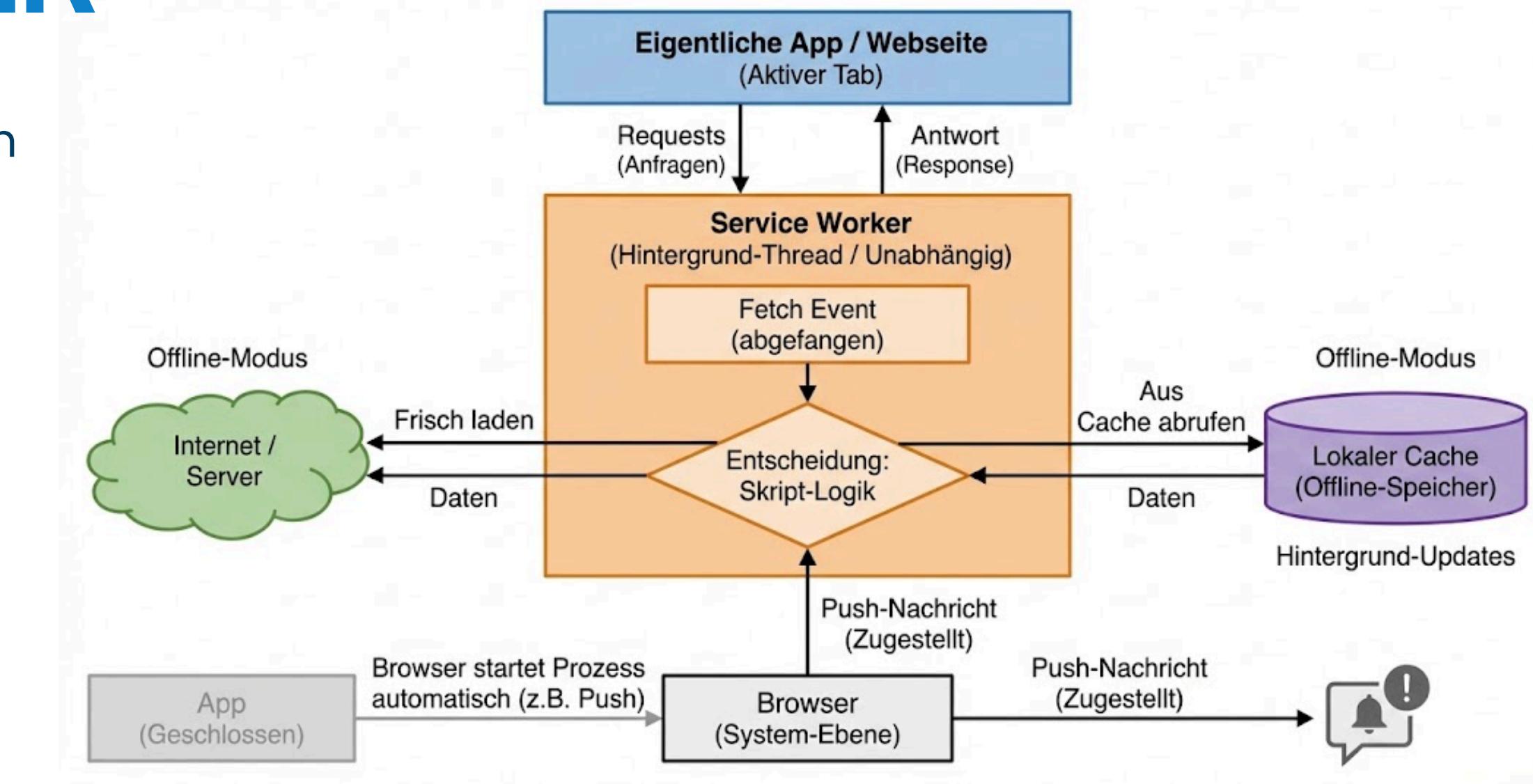
Technische Bausteine

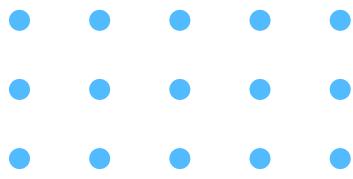




SERVICE WORKER

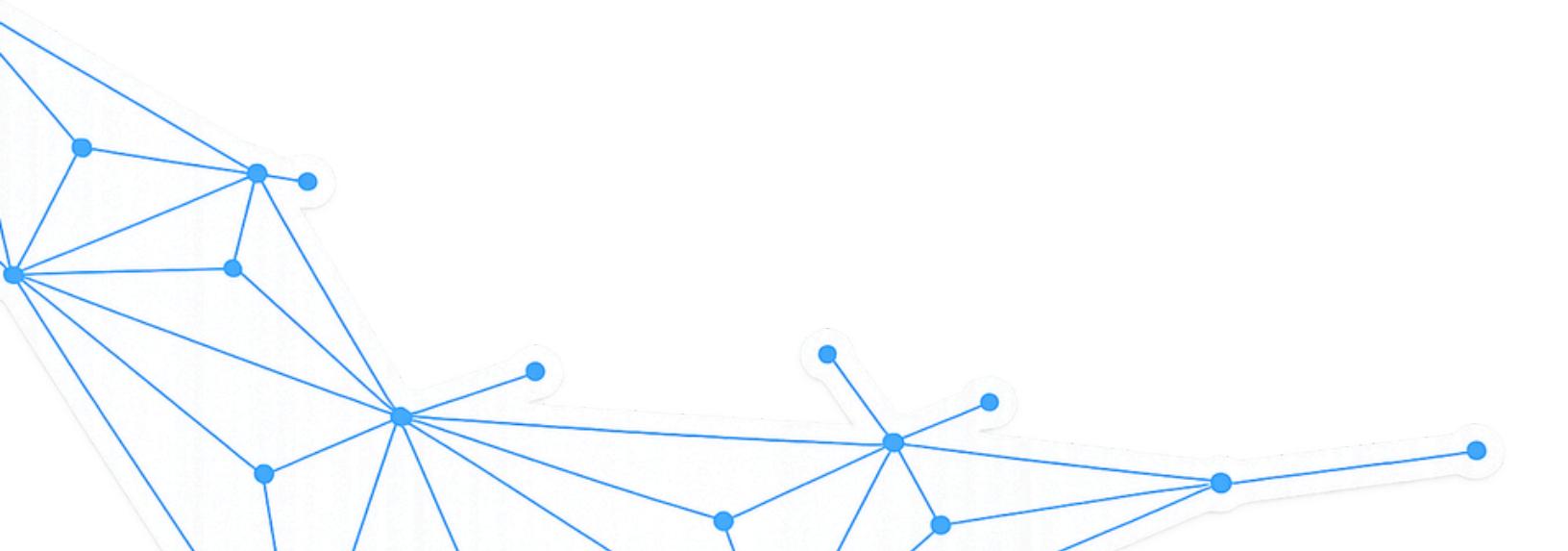
- Eigenständiges JavaScript-Programm
- Proxy zwischen App und Netzwerk
- Fängt Netzwerk-Requests ab
- Ermöglicht App-Features
- Aktiv im Hintergrund

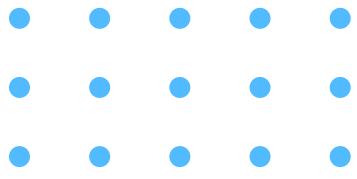




Web App Manifest

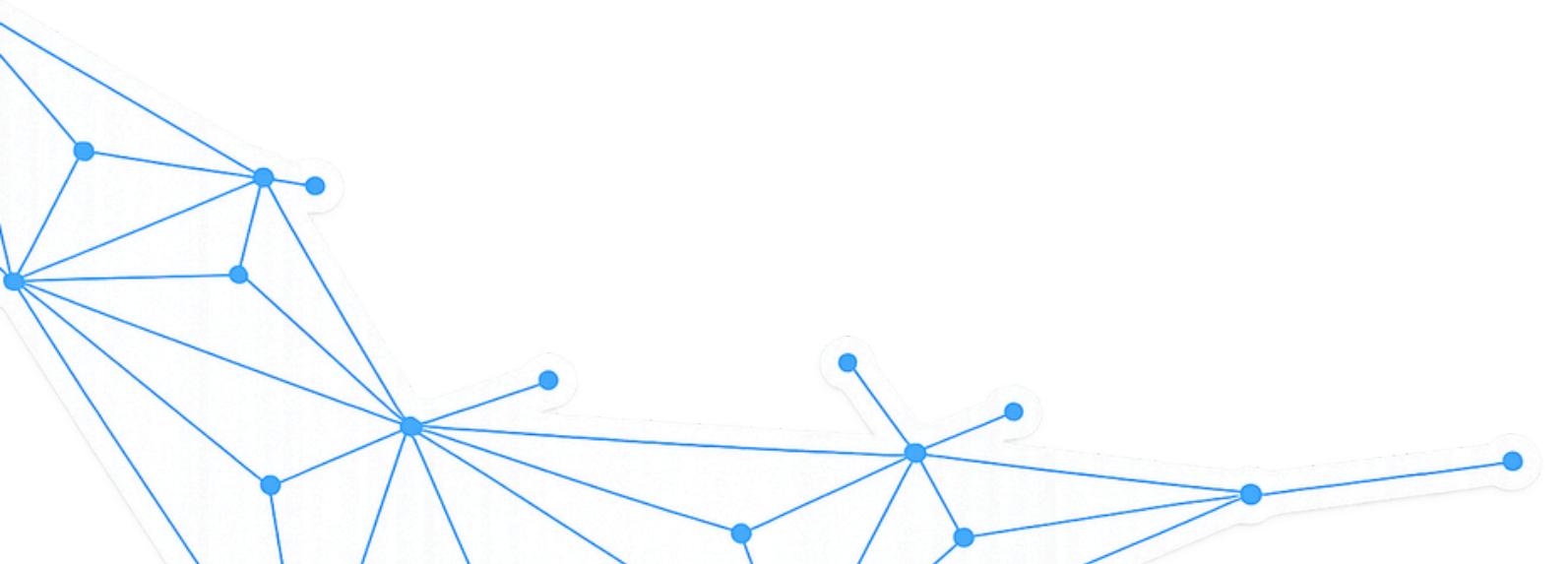
- Zentrale JSON-Konfigurationsdatei
- Ermöglicht die Installation auf dem Endgerät
- Definiert essenzielle Metadaten
- Steuert Darstellung & Erscheinungsbild der App
- Ausführung im Standalone-Modus (ohne Browser-UI)

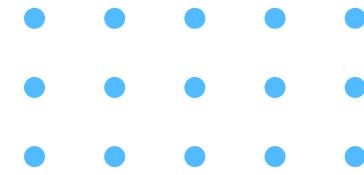




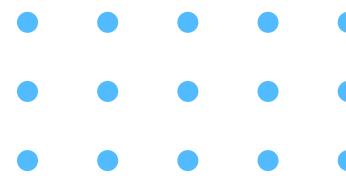
HTTPS

- Zwingende Sicherheitsbasis für PWAs
- Voraussetzung für die Ausführung von Service Workern
- Verschlüsselt die gesamte Kommunikation zwischen Client und Server
- Schützt vor Abhören und Manipulation von Daten
- Gewährleistet Integrität und Authentizität

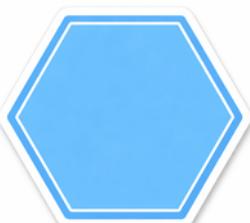


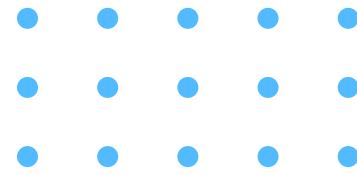


Theorie vs. Praxis



Standard Web Push Architektur in der Theorie





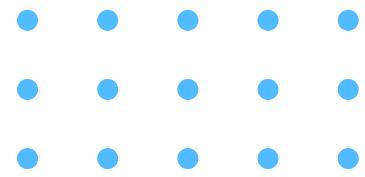
Meflo: PWA DEMO-APP

Native Simulation

- ✓ Fokus auf Look & Feel
- ✓ Integriertes Login-System
- ✓ Bewertungs-Modul
- ✓ Volle Offline-Fähigkeit

Tech Stack

- ✓ HTML5
- ✓ CSS
- ✓ Java Script
- ✓ Service Worker
- ✓ Manifest

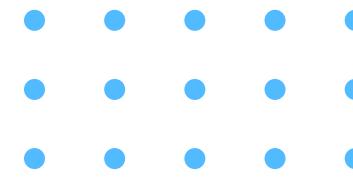


Lokale Benachrichtigung

DER INTERNE TRIGGER

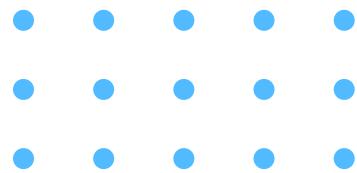
- Kein Push von außen
- Trigger: Nutzer-Event (PWA)
- Service Worker = lokaler Hub
- Notification bei Erfolg





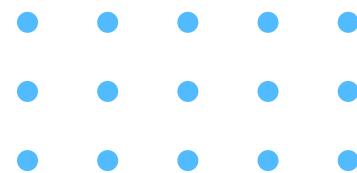
Vergleich: Theorie VS. Meflo

Feature	Standard Web Push	Meflo Implementation
Auslöser	Externes Event (z.B. neue Mail)	User Aktion (Klick)
Zustand	App kann geschlossen sein	App muss offen sein
Komplexität	Hoch	Niedrig (Reines JavaScript)
Zweck	Re-Engagement / News	Feedback zur Aktion



Implementierung





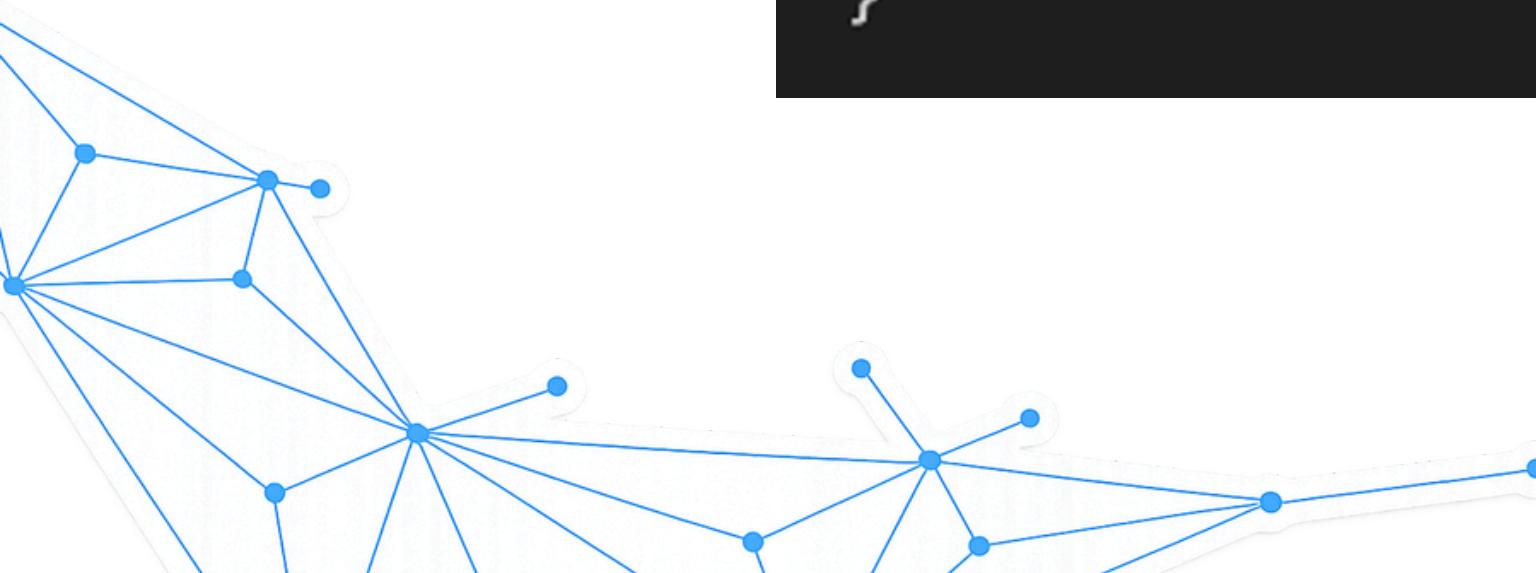
Index.html

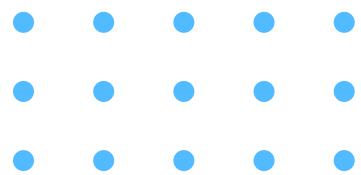
```
// Schritt 1: Login-Prüfung im Frontend (Kein Backend vorhanden)
if (user === "Admin" && password === "Affe12345") {

    // Schritt 2: Session-Merker für das Dashboard setzen
    sessionStorage.setItem('showWelcome', 'true');

    // Schritt 3: Notification-Berechtigung beim Nutzer anfragen
    Notification.requestPermission().then(function(permission) {

        // Schritt 4: Weiterleitung zum geschützten Bereich
        window.location.href = "dashboard.html";
    });
}
```





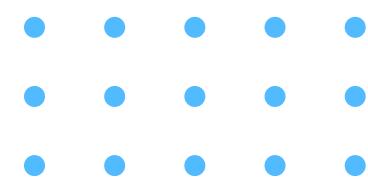
Dashboard.html

```
// Schritt 1: Kontrolle des Login-Status
if (sessionStorage.getItem('showWelcome') === 'true') {

    // Schritt 2: Registrierungs-Objekt (Schnittstelle zum System) abrufen
    navigator.serviceWorker.getRegistration().then(function(reg) {

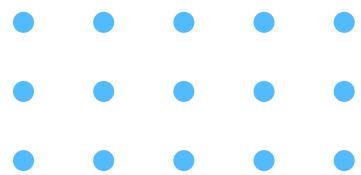
        // Schritt 3: Benachrichtigung über das Objekt auslösen
        reg.showNotification('MyFHDW', {
            body: 'Erfolgreich eingeloggt!',
            icon: 'img/homescreen192.png',
            vibrate: [100, 50, 100]
        });

        // Schritt 4: Einmal-Merker löschen
        sessionStorage.removeItem('showWelcome');
    });
}
```



Bewertung.html

```
// Schritt 1: Bereits aktiven Service Worker "greifen"  
const reg = await navigator.serviceWorker.ready;  
  
// Schritt 2: Sofortiger Befehl an das Registrierungs-Objekt  
// (Kein erneutes Abfragen der Berechtigung nötig!)  
reg.showNotification("Meflo", {  
    body: "Danke für deine Bewertung ★",  
    icon: "icons/icon-192.png",  
    vibrate: [100]  
});
```



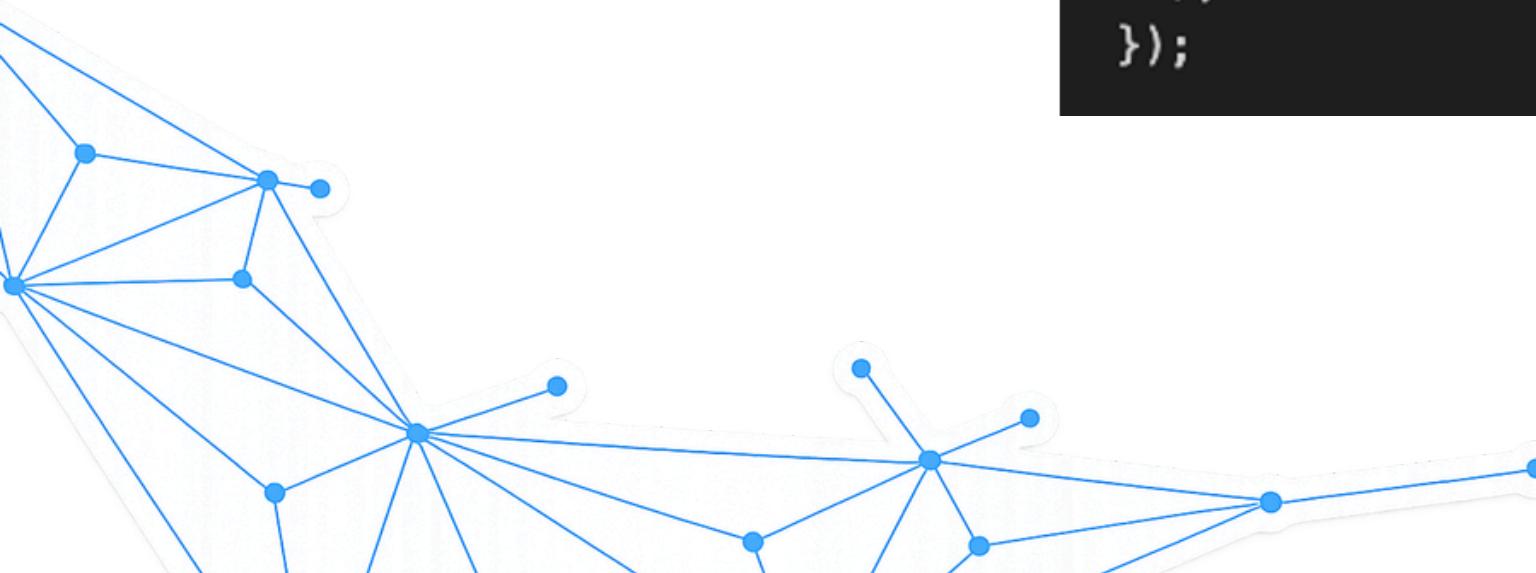
Service Worker

```
// Schritt 1: Hintergrund-Wächter für Push-Signale
self.addEventListener('push', function(event) {

    // Schritt 2: Datenpaket vom theoretischen Server entpacken
    let data = event.data ? event.data.json() : {};

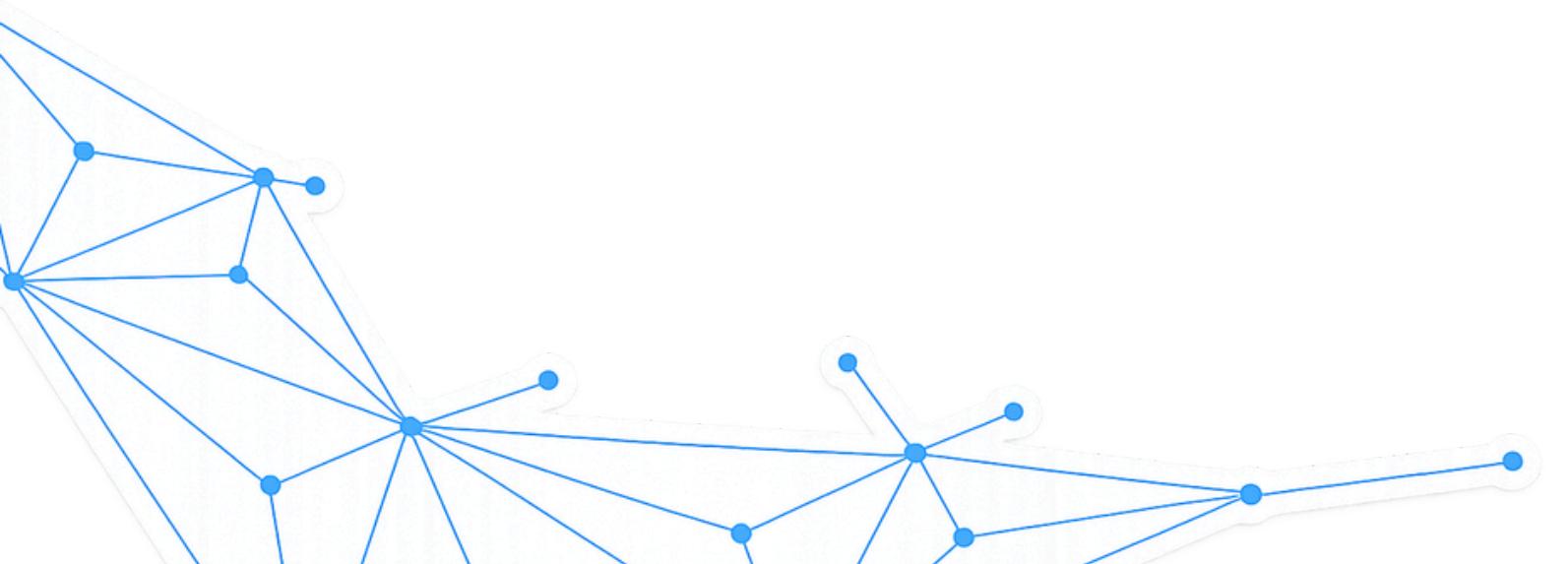
    // Schritt 3: Optische Parameter definieren
    const options = {
        body: data.body,
        icon: 'img/homescreen192.png',
        vibrate: [100, 50, 100]
    };

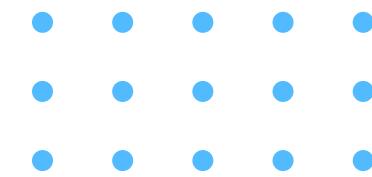
    // Schritt 4: Das System wecken und Nachricht einblenden
    event.waitUntil(
        self.registration.showNotification(data.title, options)
    );
});
```



DEMO

DEMO



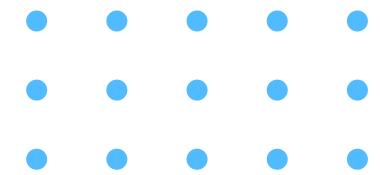


Testet MEFLO

So geht's:

- 1.QR-Code scannen
- 2.Login: Admin / Affe12345
- 3.**Wichtig:** "Mitteilungen erlauben"





Installieren von Meflo

PC / DESKTOP



Google Chrome / Edge



In der URL-Leiste rechts



Klick auf das
„Installieren“ -Icon



Bestätigen & auf
Desktop nutzen

ANDROID



Google Chrome



Menü öffnen



“App installieren“ oder...



“Zum Startbildschirm
hinzufügen“

APPLE



Safari



“Teilen“ – Button unten
mittig



Nach unten scrollen



“Zum Home-
Bildschirm“ wählen

DANKE

GIBT ES NOCH FRAGEN?



Mehmet Güngörmez
mehmet.guengoermez@edu.fhdw.de

Florian Paul Bäumler
florian_paul.baeumler@edu.fhdw.de