



Progressive Web Apps

Alex Thalhammer

Inhalt

- Überblick zu PWA
- Service Worker als Herzstück von PWAs
- Daten im Browser speichern
- Installation am Home Screen
- Background Sync
- Push Notification
- Unterstützung in Angular
- Fazit



Überblick zu PWA

Web Apps

Keine
Installation

Cross-
Plattform

Einfaches
Deployment

Einfaches
Update

Durchsuchbar

Links/
Bookmarks

Native Apps

Einfach zu
starten

Sofortiger
Nutzen

Offline

Langsame
Verbindungen

Push
Notifications

Gerätezugriff

Progressive Web Apps bieten das Beste aus beiden Welten

Progressive Enhancements: Offline, Caching, Home Screen etc.

**App muss auch ohne
Erweiterungen Nutzen bieten!**



VS.



ANGULAR
ARCHITECTS

INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

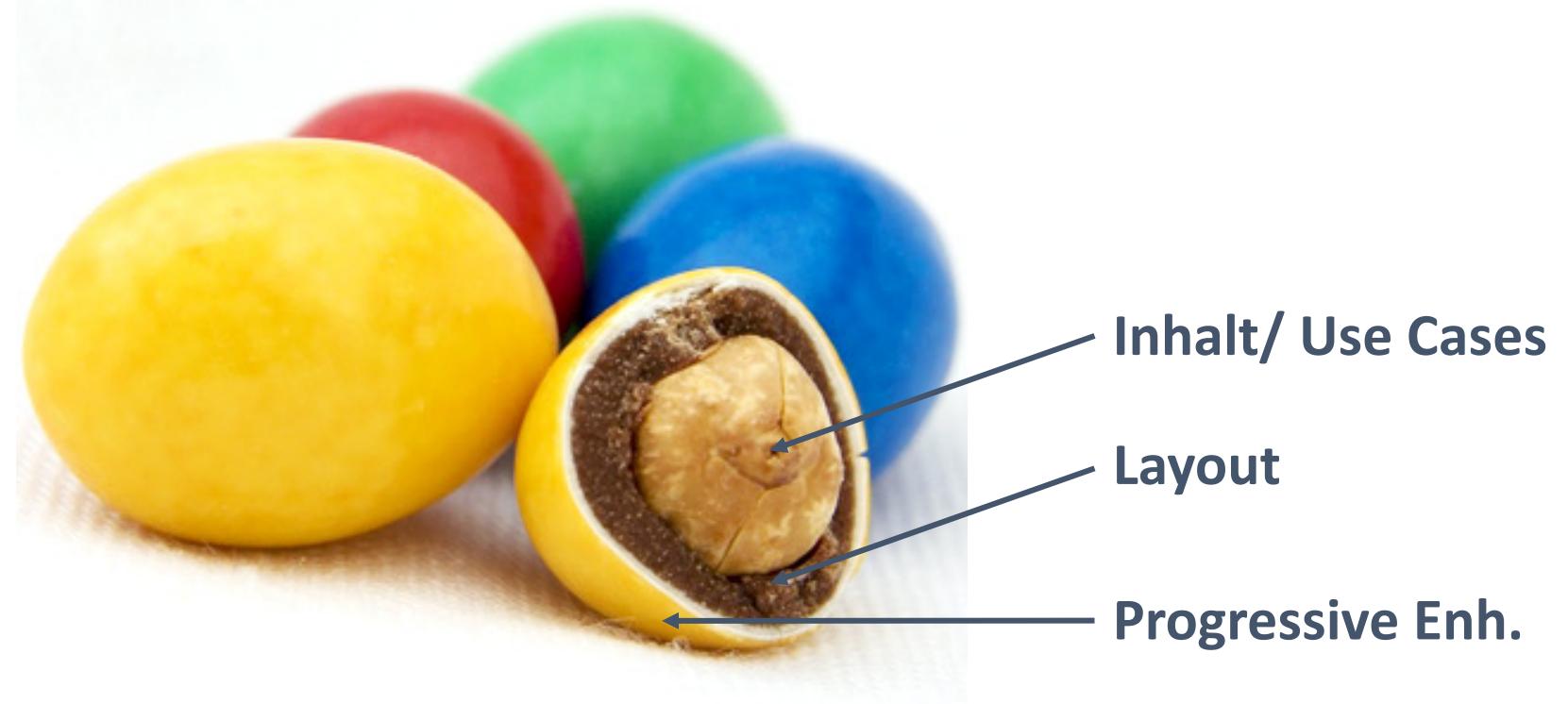


ANGULAR
ARCHITECTS

INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



[<http://alistapart.com/article/understandingprogressiveenhancement>]

Wie implementiert man Progressive Enhancements und PWA?



Service Worker

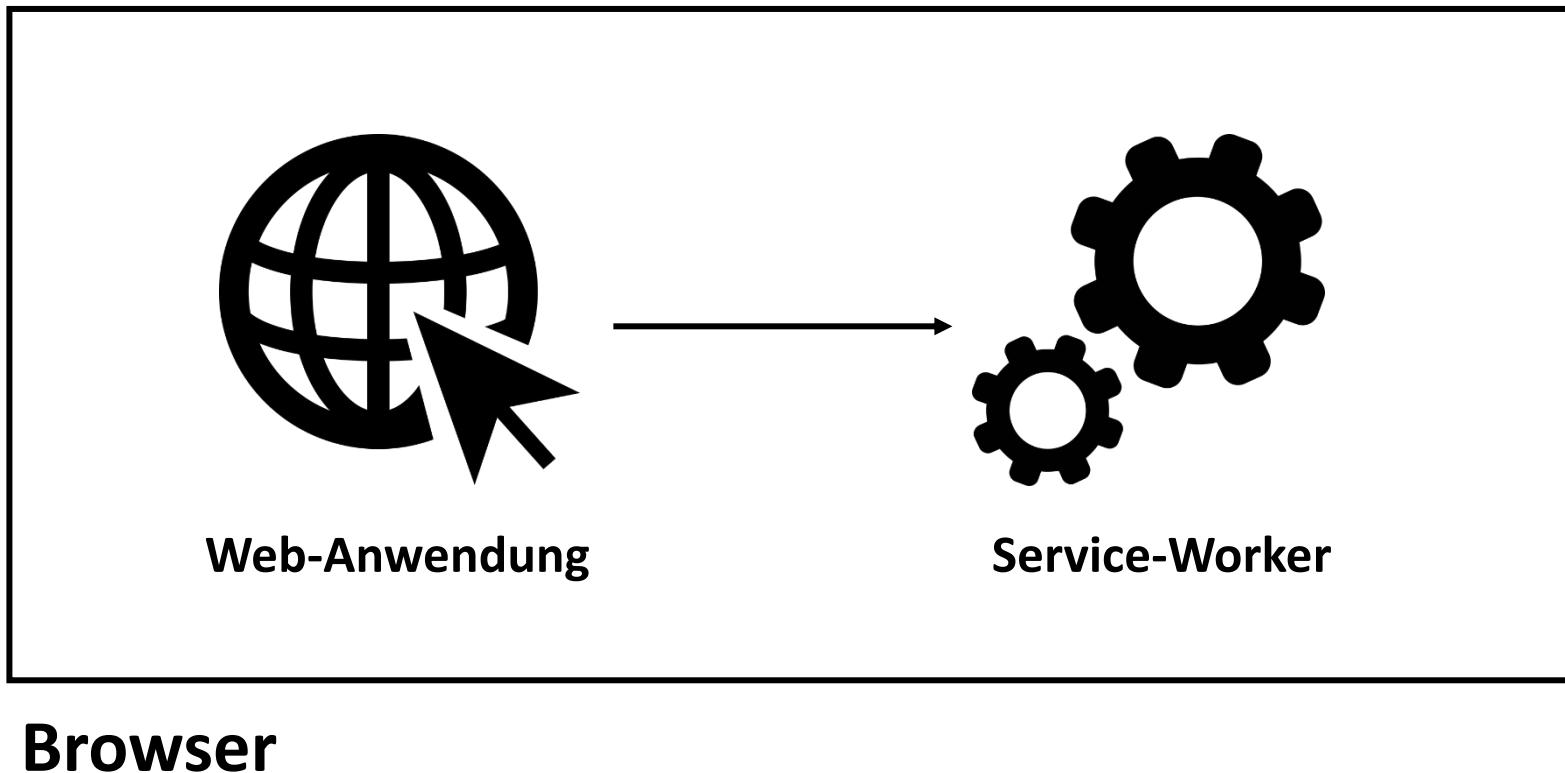
Herzstück von PWAs

Service Worker

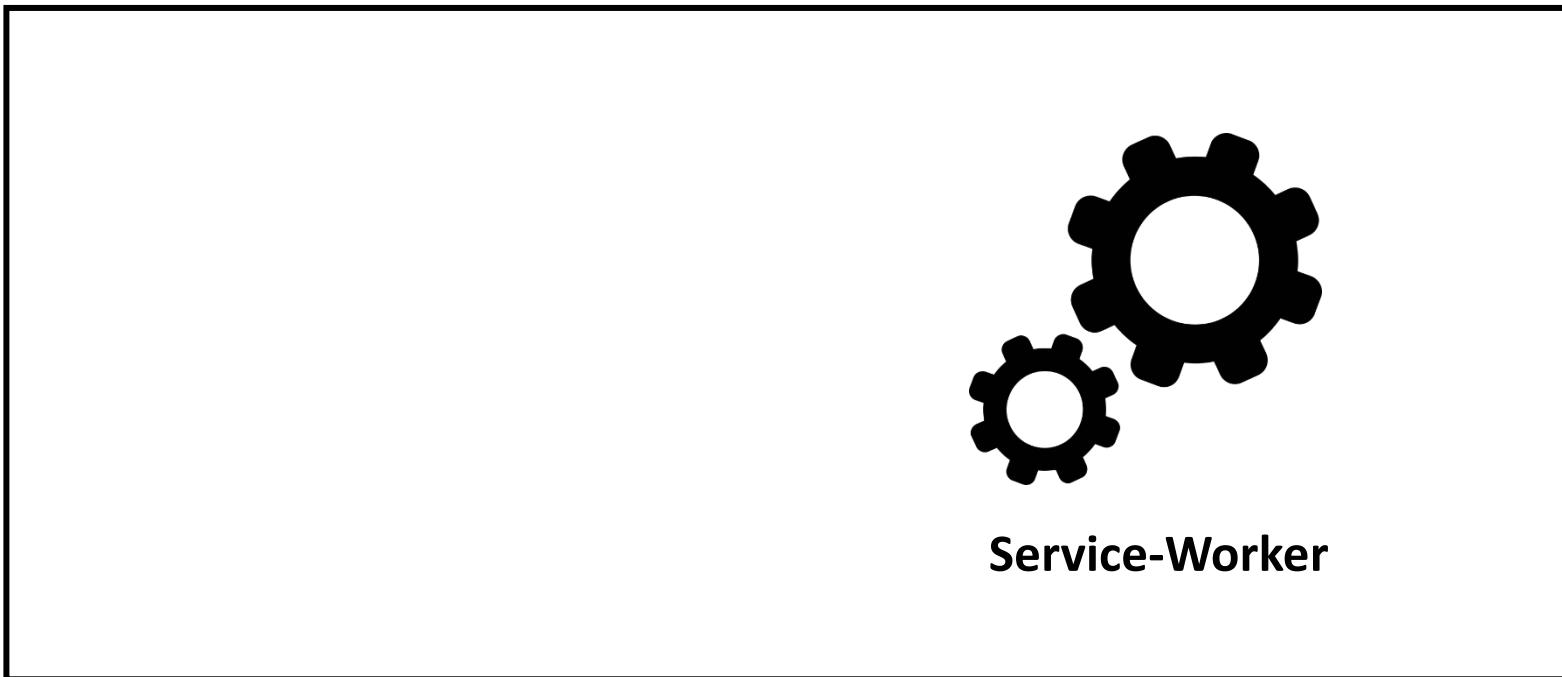


Browser

Service Worker

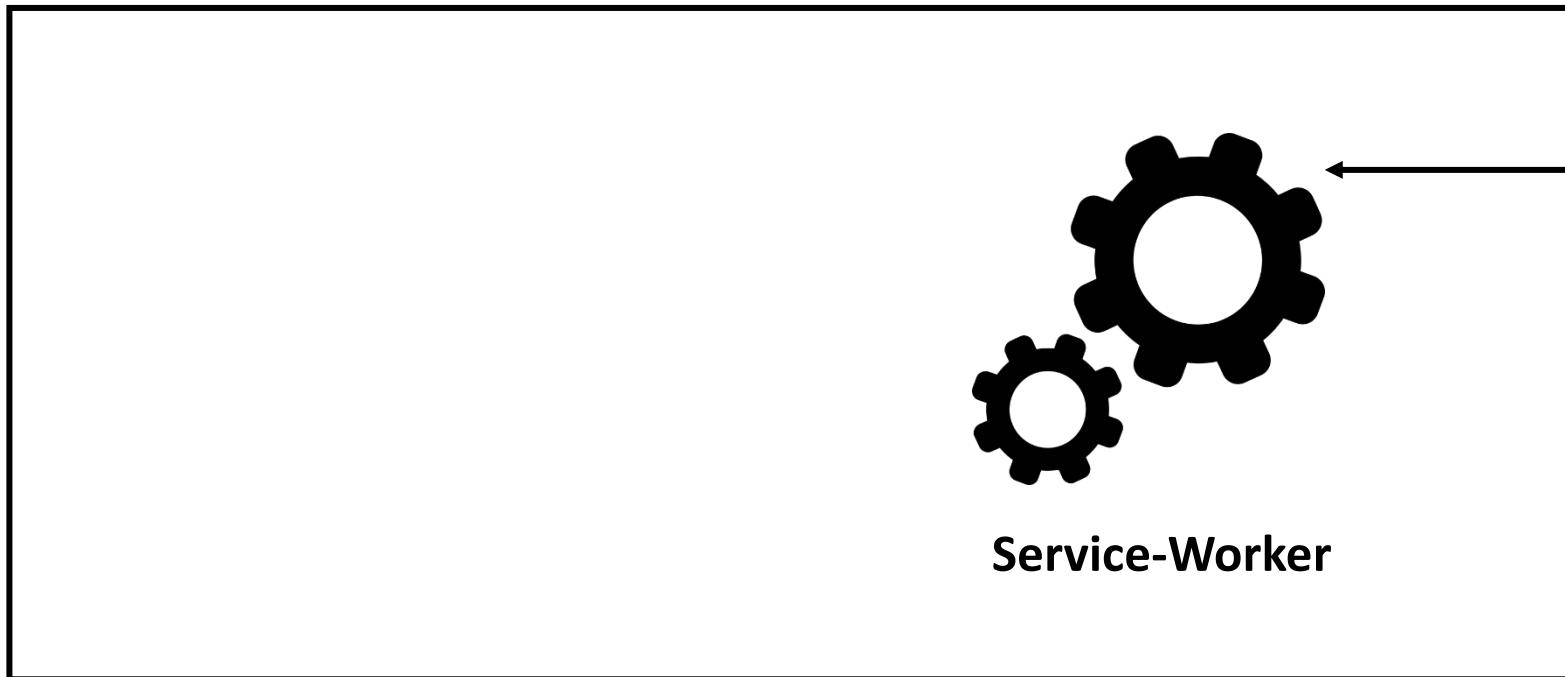


Service Worker



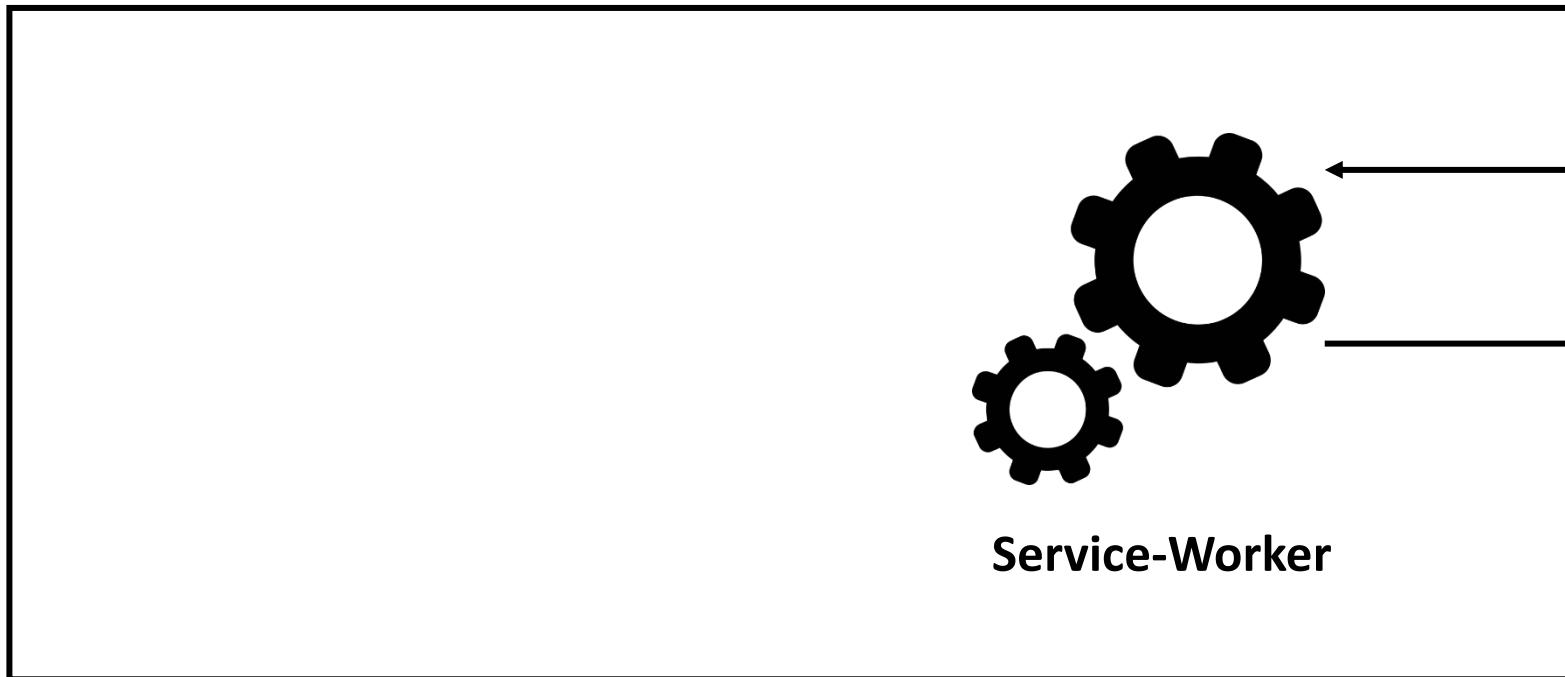
Browser

Service Worker



Browser

Service Worker



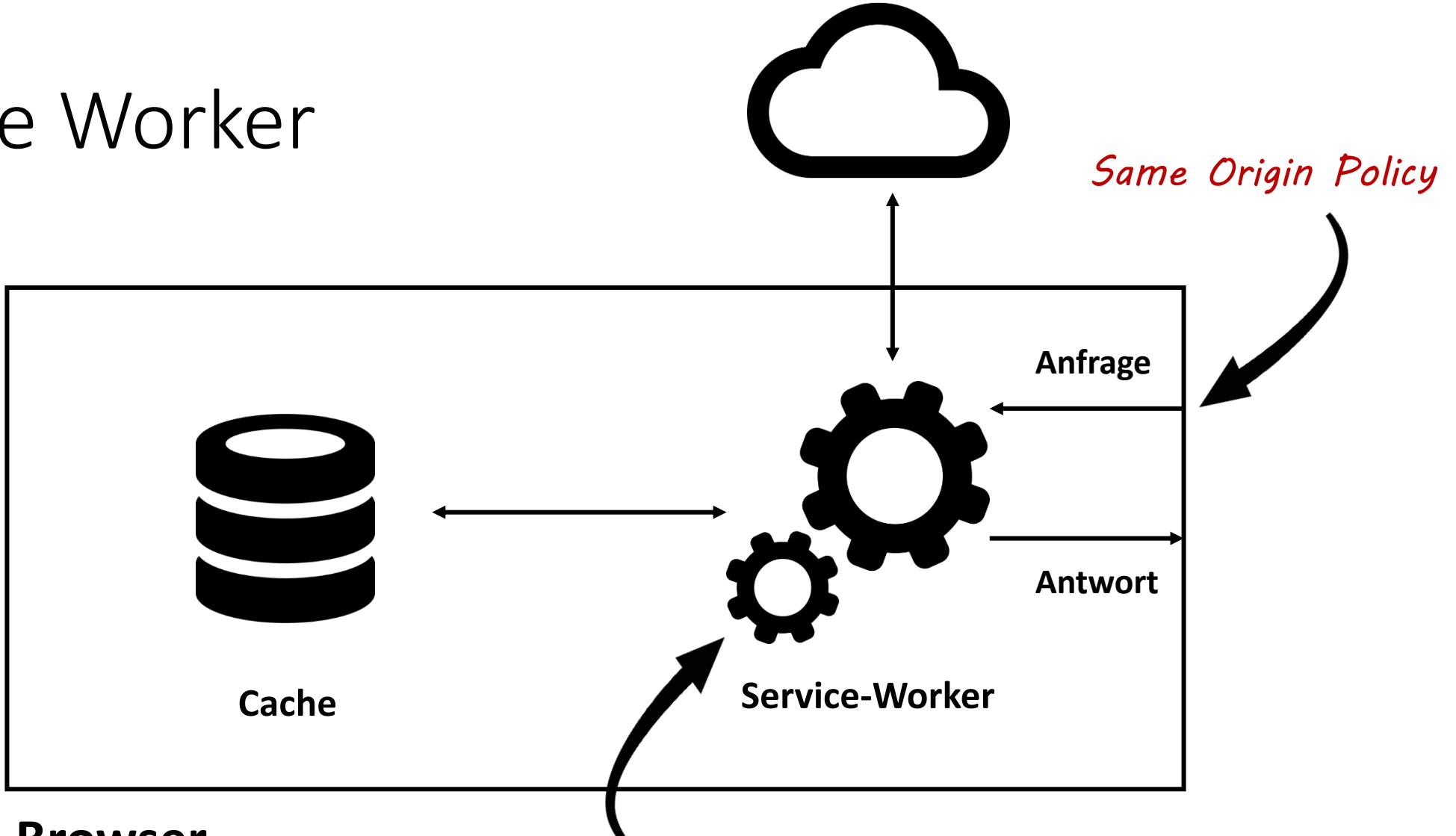
Browser

Service Worker



Browser

Service Worker



Cache Strategien

Cache only

Network only

Try Cache,
fallback to
Network

Try Network,
fallback to
Cache

...



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Service Worker registrieren

```
if('serviceWorker' in navigator) {  
  navigator  
    .serviceWorker  
    .register('/sw.js', { scope: '/' })  
    .then((r) => { console.debug('service worker started!'); });  
}
```

Service Worker Script

```
var context: any = self;

context.addEventListener('install', function(e:any) {
  e.waitUntil(fillCache()).then(() => context.skipWaiting());
});

context.addEventListener('activate', function(e) {
  e.waitUntil(clearOldCaches().then(() => context.clients.claim()));
});

context.addEventListener('fetch', function(event: any) {
  event.respondWith(cacheFirst(event.request));
});
```

FillCache and CacheFirst

```
function fillCache() {  
    return caches.open(CACHE_NAME)  
        .then(cache => cache.addAll(CACHE_FILES));  
}
```

```
function cacheFirst(request) {  
    return caches.match(request)  
        .then(response => response || fetch(request));  
}
```

Caching Strategies

<http://tinyurl.com/pwa-caching>





Installation am
Home Screen

Web App Manifest

```
{  
  "name": "Hotel PWA-Demo",  
  "short_name": "Hotel",  
  "icons": [{  
    "src": "images/touch/icon-128x128.png",  
    "sizes": "128x128",  
    "type": "image/png"  
  }, [...],  
  "start_url": "/index.html?homescreen=1",  
  "display": "standalone",  
  [...]  
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Manifest referenzieren

```
<!-- Web Application Manifest -->  
<link rel="manifest" href="manifest.webmanifest">
```

Lab



Workbox



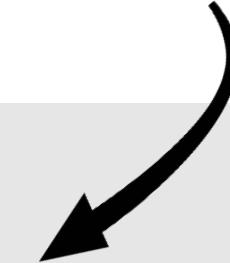
@angular/service-worker

Zwei mögliche Abstraktionen für
Service Worker

@angular/service-worker

Kommt von Angular / CLI

```
@NgModule({  
  imports: [  
    ServiceWorkerModule.register('/ngsw-worker.js',  
      {enabled: environment.production}),  
    [...]  
  ],  
  [...]  
})  
export class AppModule { }
```



- + Config-Datei mit Cache-Strategien etc.
- + ng build –prod

@angular/pwa

- Installiert @angular/service-worker
 - npm install
 - Importiert Angular-Modul
 - Generiert ngsw-config.json
 - Generiert Web App Manifest
- ng add @angular/pwa

SwUpdate Service

- checkForUpdate()
- available (Observable)
- activateUpdate()
 - Danach Reload nötig



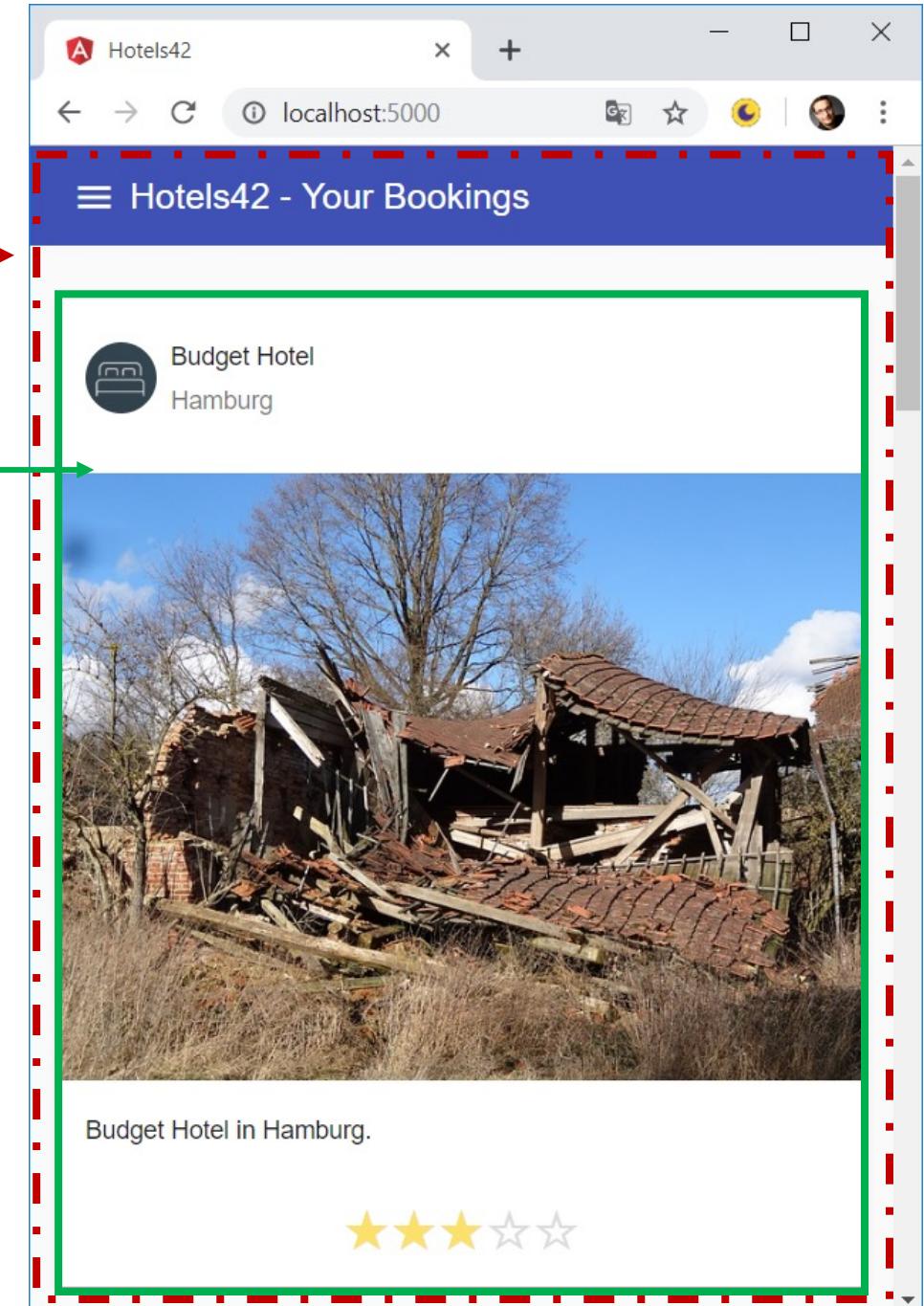
AppShell

App Shell

- App Shell: Vorrendern
 - Wie bei SSR, nur statisch!
- Content: Cachen
- Grund: Sofortiger nutzen

App Shell →

Content →



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

App Shell generieren

ng generate app-shell

Lab



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

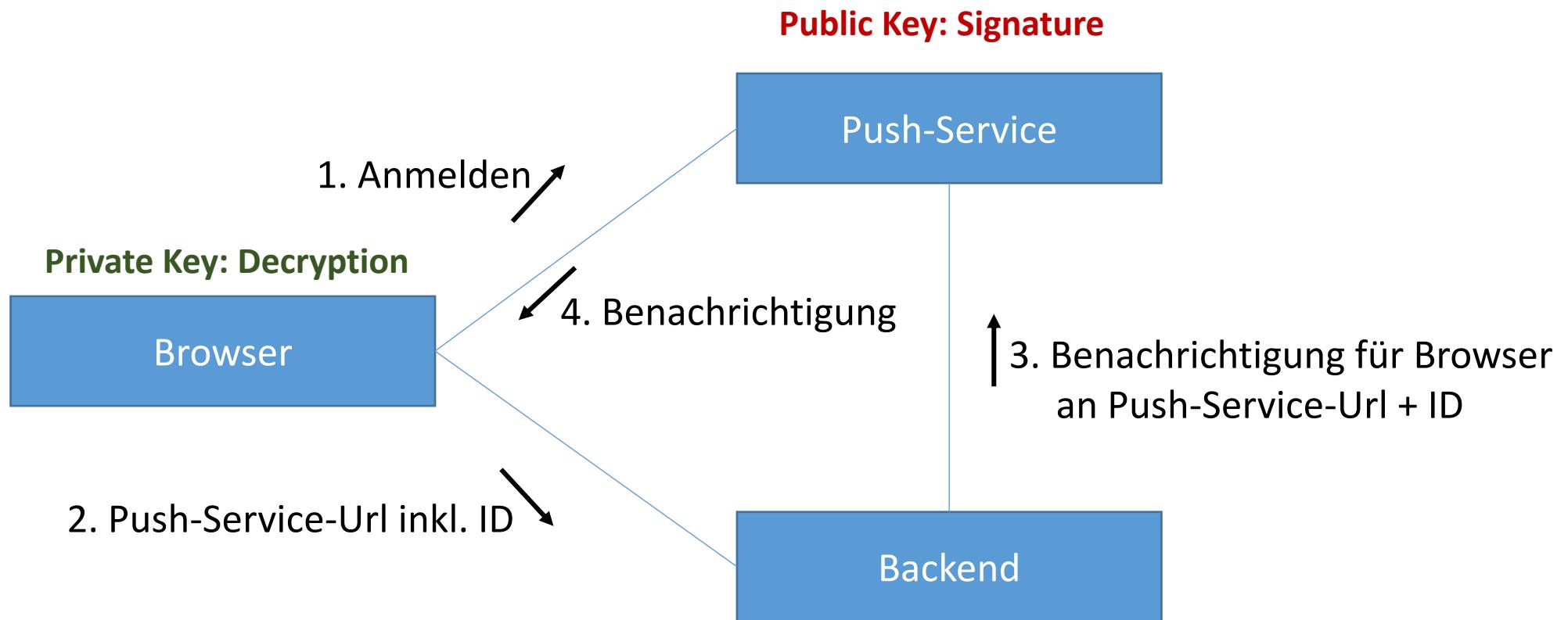


SOFTWARE
ARCHITECT



Push Notifications

Push Notifications



SwPush

- `requestSubscription(...)`
 - Rückgabewert hat Daten für zum Server
- `messages (Observable)`
- `notificationClicks (Observable)`



Daten lokal speichern

Daten offline speichern

- LocalStorage (kein Zugriff über Service Worker)
- WebDb (deprecated aber hier)
- IndexedDb

Quotas als Herausforderung

Browser	Limit
Chrome	<6% of free space
Firefox	<10% of free space
Safari	<50MB
IE10	<250MB
Edge	Dependent on volume size

[<https://developers.google.com/web/fundamentals/instant-and-offline/web-storage/offline-for-pwa>]

Fokus

- Kein langfristiger Offlinebetrieb
- Überbrücken langsamer oder fehlender Datenverbindungen

Postausgang

		BETREFF
!	AN	
◀	Datum: Heute	
	'mama@hotel-mama.com'	<i>Frische Wäsche</i>
	'bruce.wayne@wayne-enterprises.com'	<i>AW: Neuer Anzug</i>
	'office@starfleet.com'	<i>Bewerbung als Captain</i>
	'Mirko Schrempp (mschrempp@sandsme...'	<i>Keynote BASTA!</i>

Abstraktionen für Browser-Datenbanken

PouchDb

Dexie

RxDB

...



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Lab



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Background Sync

Background Sync

- App fordert **Background Sync** an
- Service Worker führt **Sync Event** aus, wenn es passend erscheint (Netzwerk, Akku, ...)
- Pläne für periodische Background Sync

Background Sync anfordern

```
let nav: any = navigator;

if ('serviceWorker' in nav && 'SyncManager' in window) {
    nav.serviceWorker.ready
        .then(reg => {
            return reg.sync.register('upload');
        })
        .catch(_ => {
            return this.buchungService.upload();
        });
} else {
    this.buchungService.upload();
}
```

The diagram illustrates the execution flow of the code. An arrow labeled "Tag" points from the red `sync.register` call to the explanatory text "Tag". Another arrow labeled "Fallback: Directly sync in Browser Thread" points from the `this.buchungService.upload()` call in the catch block to the explanatory text "Fallback: Directly sync in Browser Thread".

Sync Event in Service Worker

```
context.addEventListener('sync', function(event) {
  console.debug("Service Worker: sync, tag=" + event.tag);
  if (event.tag == 'upload') {
    event.waitUntil(bs.upload().then(_ =>
      console.debug('background-upload finished'))));
  }
});
```

Lab



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



PWAs im weiteren Sinn

PWAs im weiteren Sinn

- Moderne Browser APIs
 - <https://whatwebcando.today/>
- Welcher Browser unterstützt was?
 - <https://caniuse.com/>

Fazit

Quick Wins

Offline mit Service Worker

- Rascher Start
- Langsame Verbindungen überbrücken

Push Notifications

- Asynchrone Geschäftsprozesse
- "Benutzer zurückholen"

Web App Manifest

- Installieren auf Home Screen

Browser DBs und Background Sync

- Daten vorhalten
- Daten synchronisieren



ANGULAR
ARCHITECTS

INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Fazit

Jede Web-App sollte eine PWA sein

Sehr verlockend (nicht nur) für Business Apps

M&Ms können Ihnen den Tag retten

