

Wie JavaScript-Frameworks die Web- und Appentwicklung verändern?!

Michael Häfner

Zu meiner Person

Michael Häfner

Webentwicklung seit ca 15 Jahren

Softwareentwickler bei der RSU GmbH

RSU GmbH



Großhandelsportal für
Wiederverkäufer



Onlineshop für
Privatkunden

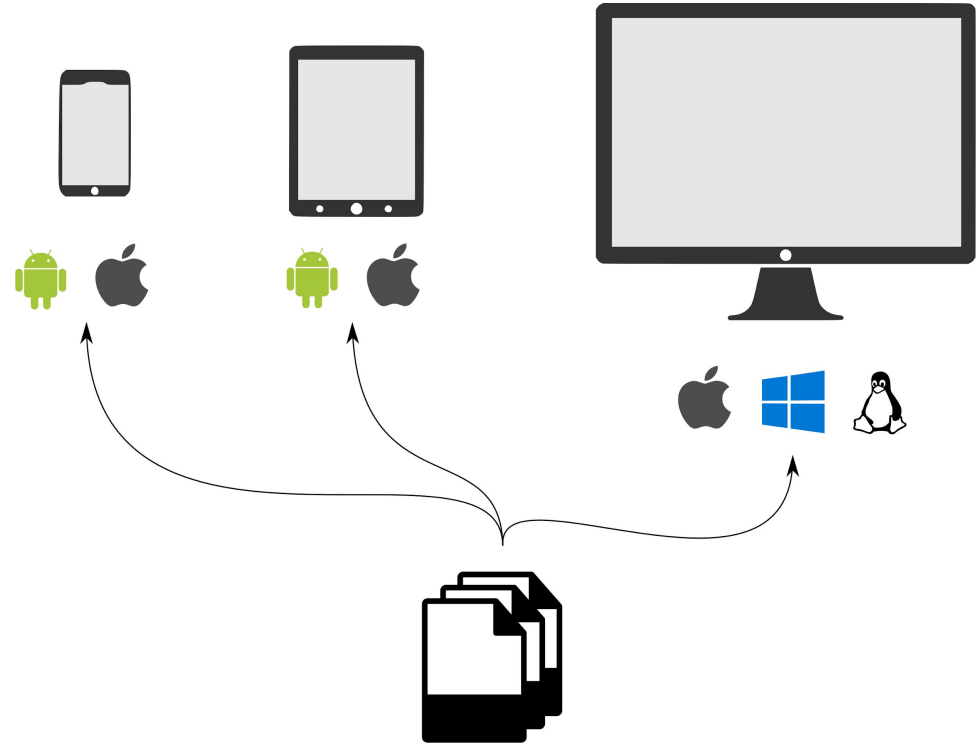
Wie JavaScript-Frameworks die Web- und Appentwicklung verändern?!



Cross Platform Apps

Theorie:

Eine Codebasis aus der für alle
Endgeräte eine App gebaut wird.



Geschichte der Webentwicklung

2007 iPhone

2008 Android

--- **mobile Websites**

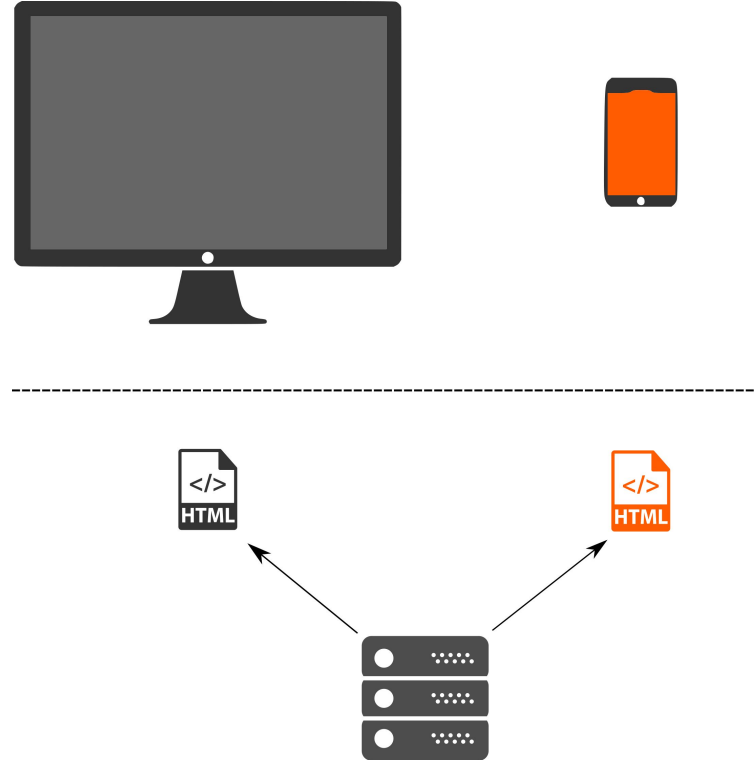
Mobile Websites

Neue Anforderungen

- Bedienung per Touch
- geringere Bildschirmgröße

Umsetzung

- Zwei Varianten der Website
(Desktop, Mobile)



Geschichte der Webentwicklung

2009 Cordova

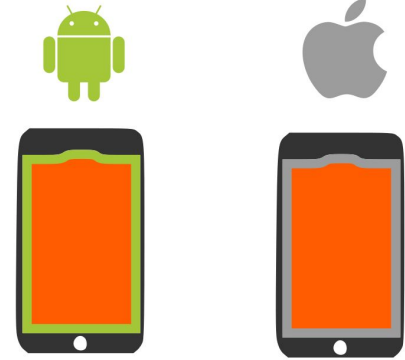
--- **hybride Apps**

Hybride Apps

Mix aus nativer App und Website

Website wird in WebView dargestellt

Zugriff auf native APIs



<https://cordova.apache.org/>

<https://capacitor.ionicframework.com/>

Geschichte der Webentwicklung

2010 iPad

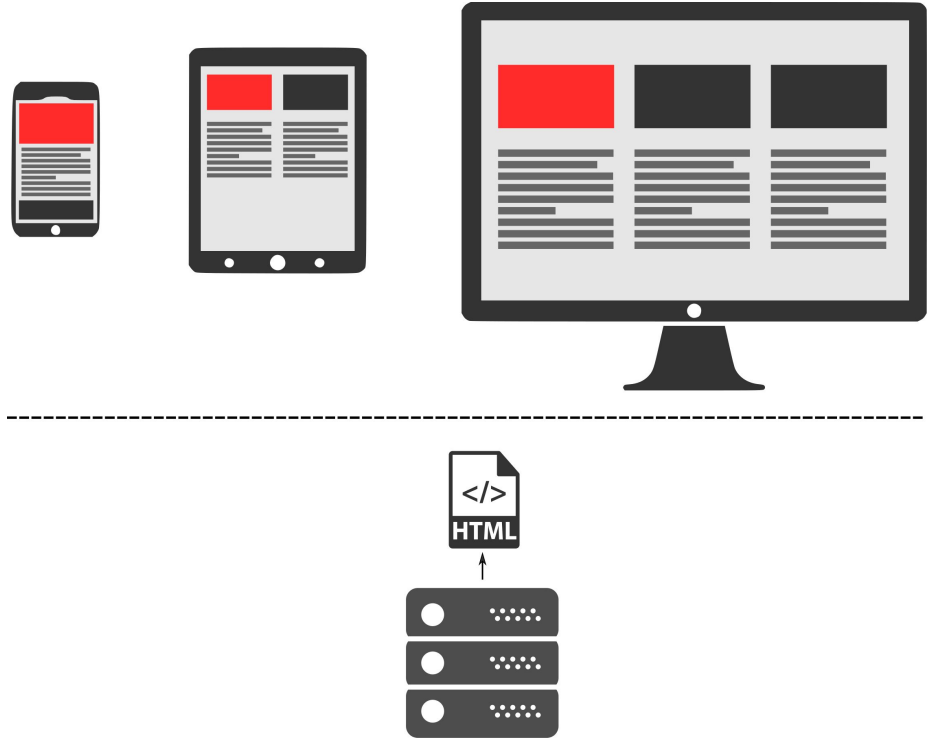
--- **Responsive Websites**

Responsive Websites

Layout passt sich automatisch der
Displaygröße an

Eine Codebasis

Umgesetzt im CSS



Geschichte der Webentwicklung

2010 AngularJS

2011 Ember.js

2013 React

--- **Single Page Applications**

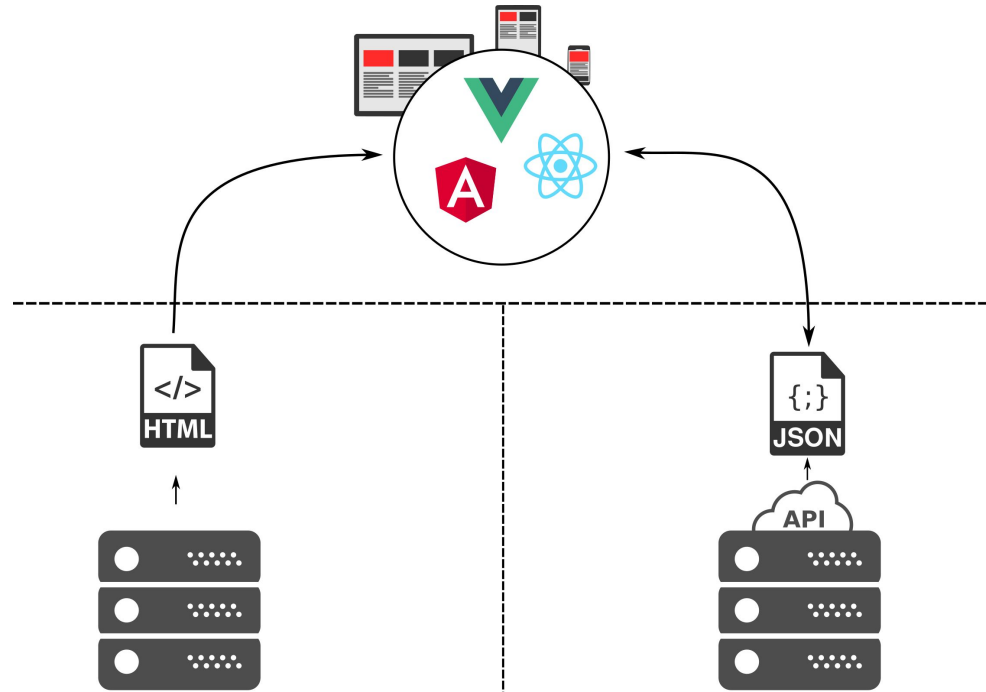
Single Page Application

Frontend ist Applikation

Backend ist “Datenlieferant”

Frontend mit JavaScript

Application-Framework geschrieben



<https://angular.io/>

<https://reactjs.org/>

<https://vuejs.org/>

Geschichte der Webentwicklung

2013 Electron

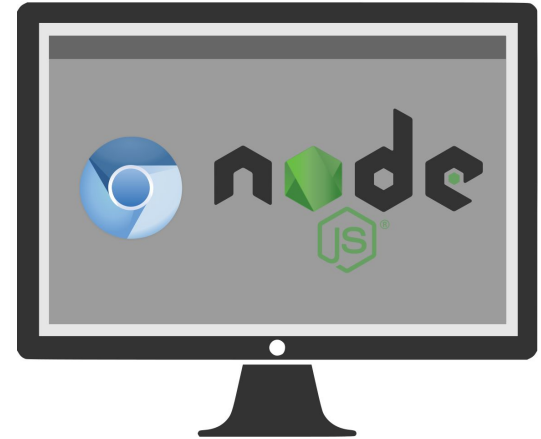
--- **Native (JavaScript) Desktop Apps**

Native (JavaScript) Desktop Apps

Mix aus Website und nativer Desktop App

Chromium zum Darstellen der Website

Nodejs für den Zugriff auf das Betriebssystem



Geschichte der Webentwicklung

2015 React Native

NativeScript

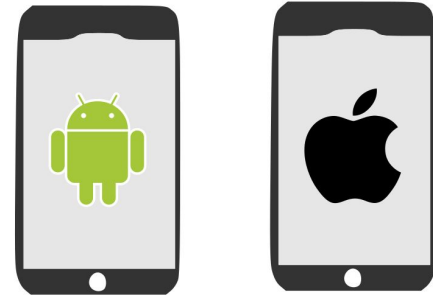
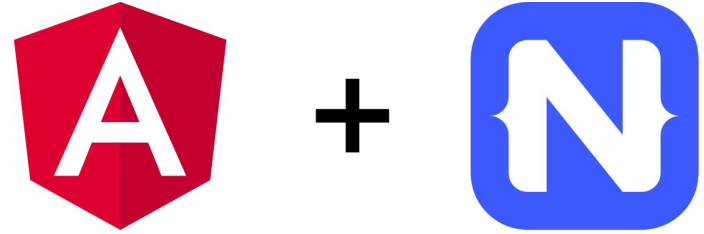
--- **Native (JavaScript) Apps**

Native (JavaScript) Mobile Apps

Native User Experience

Bessere Performance

Mächtiger als hybride Apps



Geschichte der Webentwicklung

2018 Service Worker

--- **Progressive Web Apps**

Progressive Web Apps

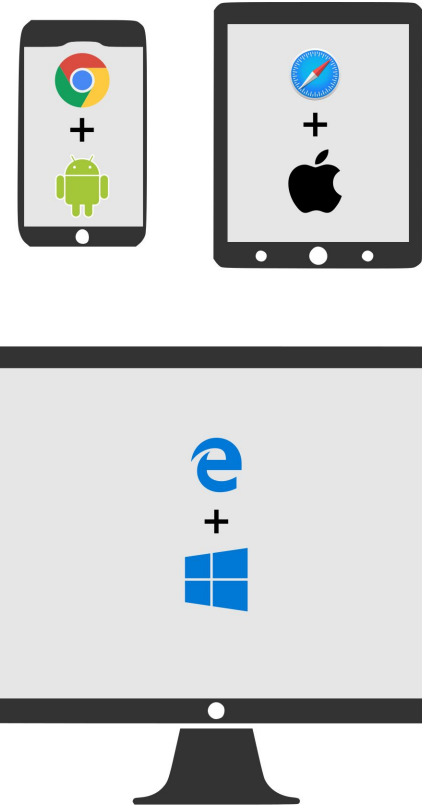
Idee: Die Lücke zu nativen Apps schließen

Funktionieren nur in modernen Browsern

Bringen neue Funktionalitäten:

- Installierbar
- Push Nachrichten
- Offlinenutzung

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



Zusammenfassung der Geschichte

Theoretisch ist es möglich Cross Platform Apps mit Hilfe von JavaScript Frameworks zu entwickeln.

Doch ist dies auch praxistauglich ?

RSU braucht einen Passwortmanager

Ende zu Ende Verschlüsselung

Möglichst eine Codebasis

Daten auf eigenen Servern

Keine neue Programmiersprache

Auf allen Endgeräten nutzbar

Offline nutzbar

Vorgehen

1. Single Page Application
2. Progressive Web App
3. Native Desktop App
4. Hybride App

Die Single Page Application

Neue Angular Applikation anlegen

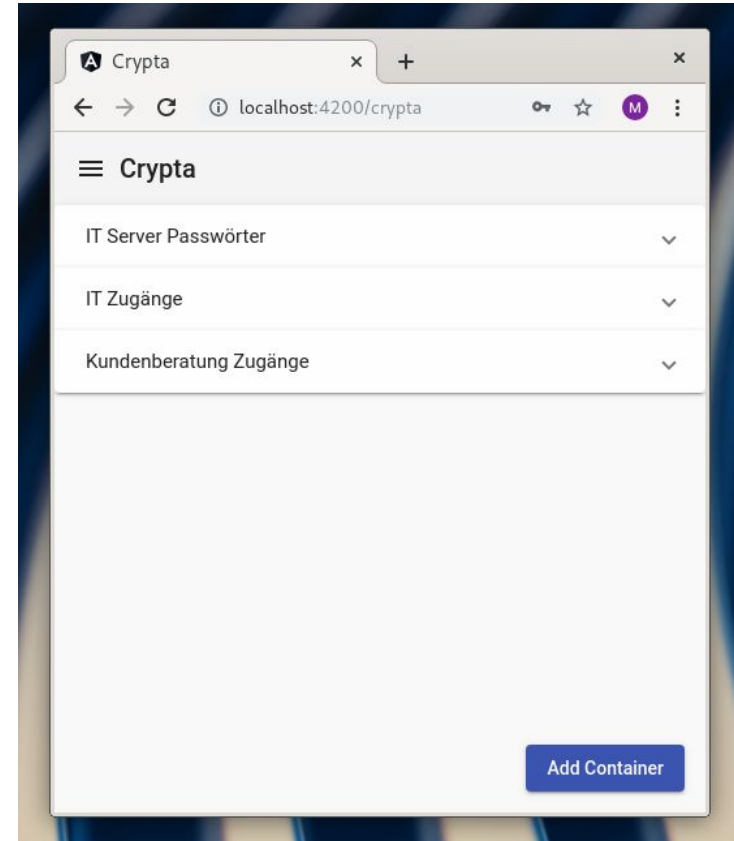
```
npm install -g @angular/cli
```

```
ng new my-app-name
```

```
cd my-app-name
```

```
ng serve
```

App entwickeln.



Die Progressive Web App

ng add @angular/pwa

Fügt Service Worker hinzu

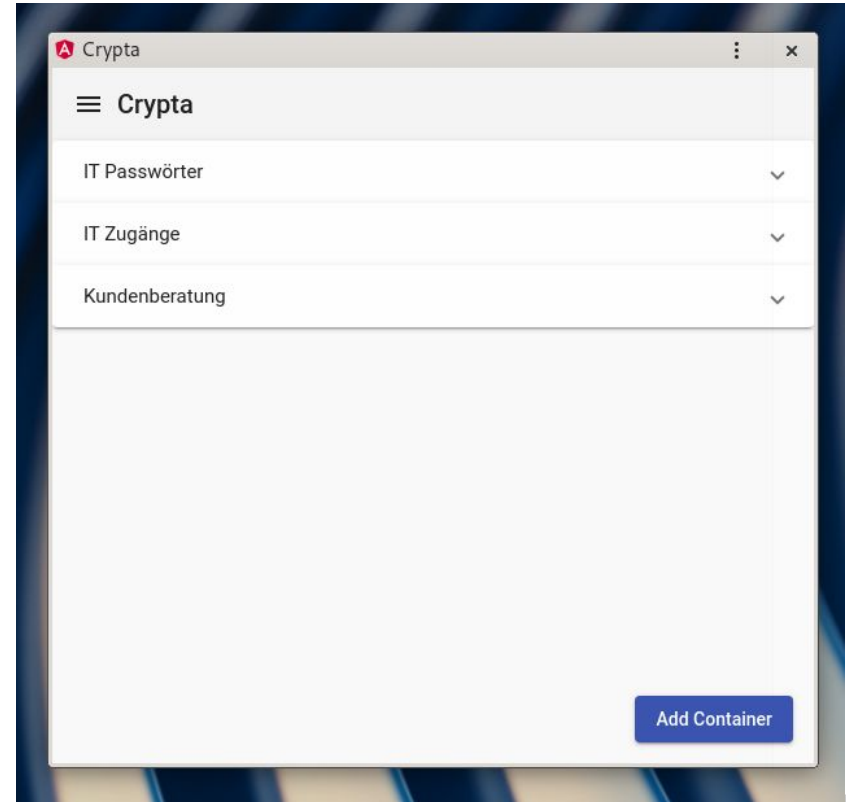
Konfiguration über ngsw-config.json

```
1  {
2    "$schema": "../node_modules/@angular/service-worker/config/schema.json",
3    "index": "/index.html",
4    "assetGroups": [
5      {
6        "name": "app",
7        "installMode": "prefetch",
8        "resources": {
9          "files": [
10             "/favicon.ico",
11             "/index.html",
12             "/manifest.webmanifest",
13             "/*.css",
14             "/*.js"
15           ]
16        }
17      },
18      {
19        "name": "assets",
20        "installMode": "lazy",
21        "updateMode": "prefetch",
22        "resources": {
23          "files": [
24             "/assets/**",
25             "/*.eot|svg|cur|jpg|png|webp|gif|otf|ttf|woff|woff2|ani"
26           ]
27        }
28      }
29    ],
30    "navigationUrls": ["/*", "!/**/*.*", "!/**/*_*", "!/**/*_/*"]
31  }
32
```

Die Progressive Web App

Unterschiede zur Webapp

- Offlinenutzung
- Installierbar
- Push Nachrichten



Die Native Desktop App

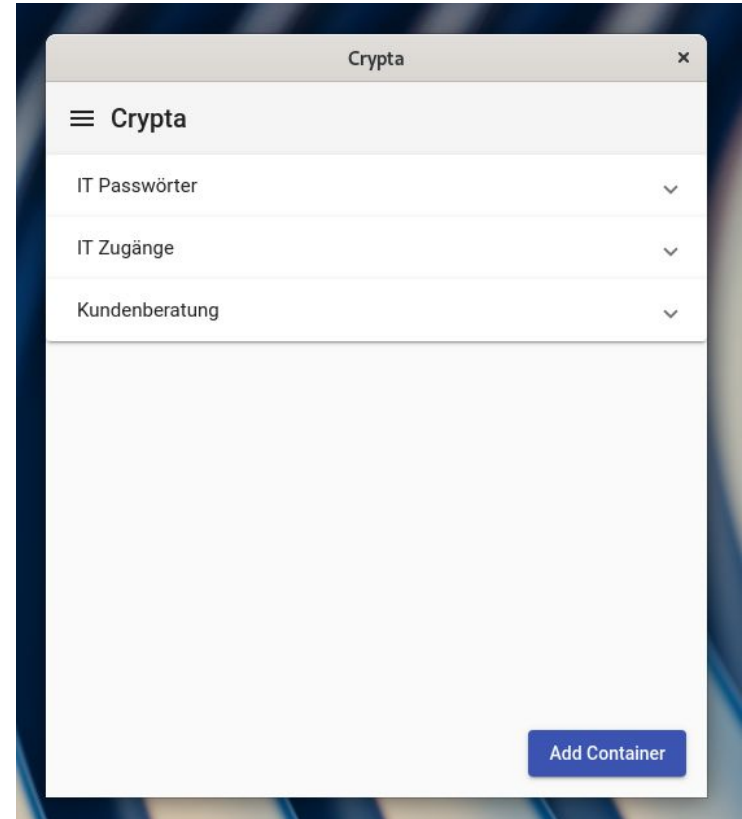
```
npm install electron
```

```
ng build --base-href ./
```

```
electron .
```

Achtung zum Builden:

- zusätzlich Libraries notwendig (Je nach Betriebssystem)



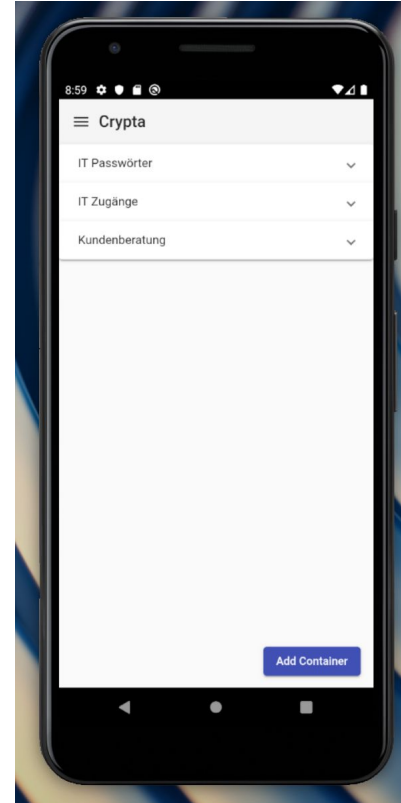
Die Hybride App

```
npm install @capacitor/core @capacitor/cli
```

```
npx cap init
```

```
npx cap add android | ios
```

```
npx cap open android | ios
```



Auswirkungen für unser Unternehmen

- kürzere Entwicklungszeit
- weniger Know-How notwendig
- Mitarbeiter können flexibel zwischen Projekten / Teams wechseln
- Recruiting kann sich auf eine Zielgruppe konzentrieren

=> Geringere Kosten

Fazit

Mit JavaScript Frameworks lassen sich mächtige Applikationen entwickeln:

- für alle Plattformen
- in kurzer Zeit
- mit geringen Kosten

Vielen Dank für die Aufmerksamkeit!

michael.haefner@rsu-reifen.de

<https://www.rsu-reifen.de/>

<https://www.rsu-reifen.de/jobs>

<https://github.com/rsu-gmbh>