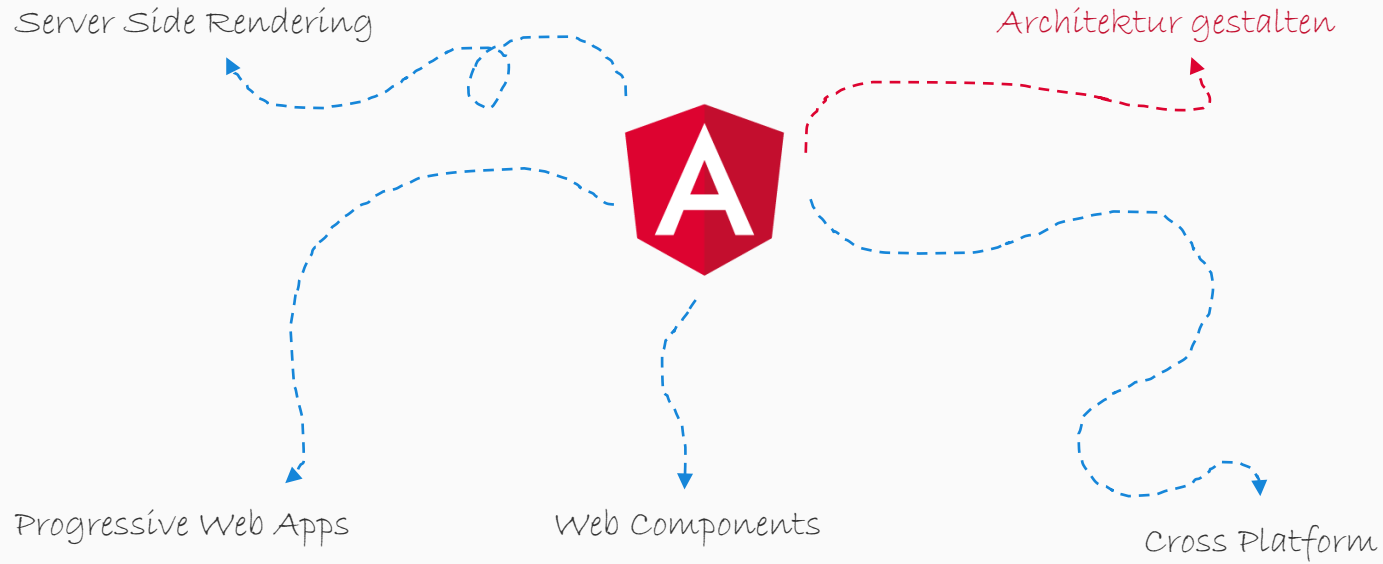


Enter the Angular Universe

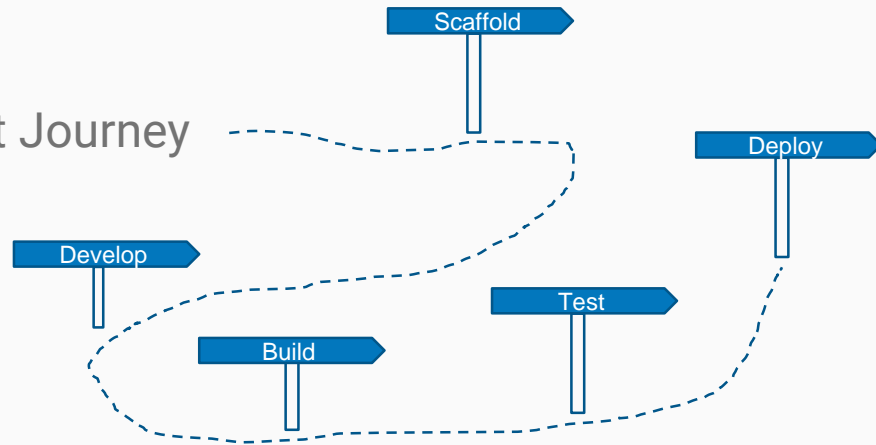


One Framework. Mobile & Web



Es geht um Effektivität

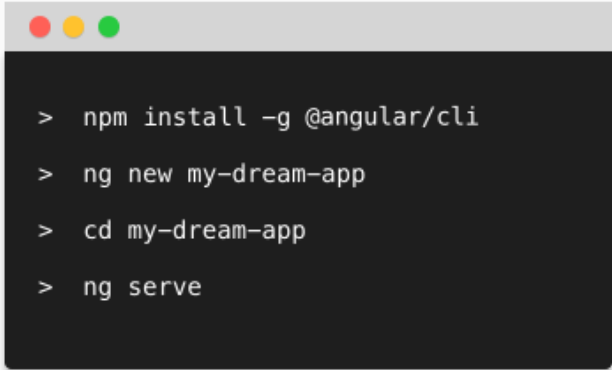
- Konsistenz der Plattform
- Gleich zu bedienende APIs
- Automatisierte Development Journey





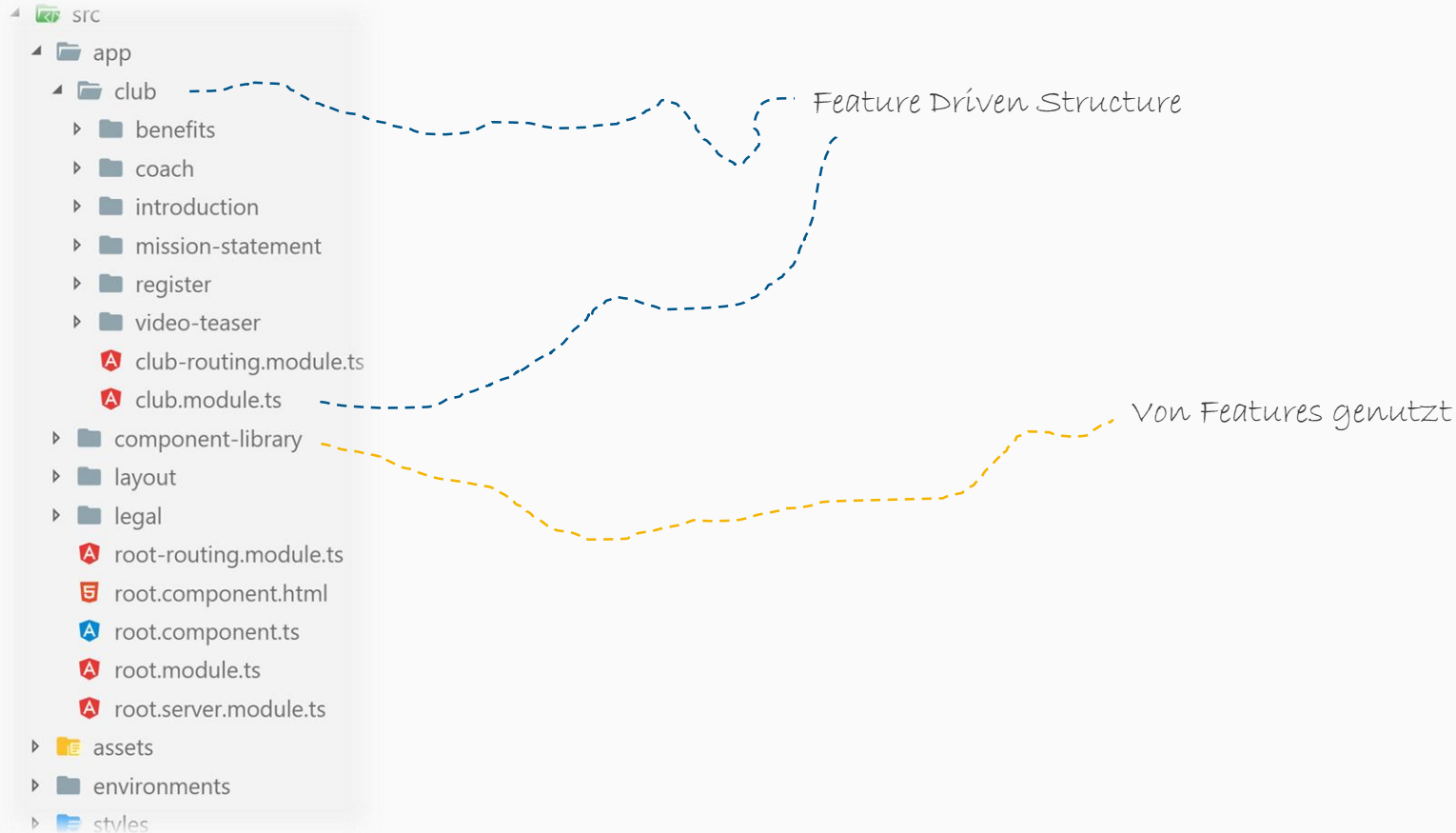
Kompakter Überblick

- Erzeugt Projektstruktur
- Bietet Development-Workflow an
 - Development Server
 - Build-Process
- Stellt Test-Setup zur Verfügung
- Erzeugt Code-Coverage-Report
- Code-Generation
- <https://cli.angular.io/>



```
> npm install -g @angular/cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```



Der herkömmliche Projektaufbau





NRWL EXTENSIONS

Nx by Nrwl

- Ergänzt , durch eigene *Schematics*
- Steigert Produktivität
 - *Unterstützt mehre Apps & Libraries*
- Fügt Skripts zur Analyse hinzu
-  github.com/nrwl/nx

Nx installieren

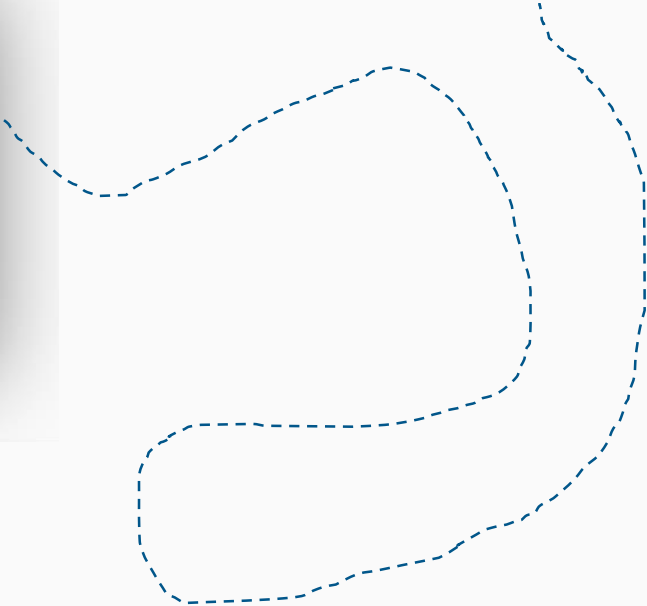


```
1 npm install -g @nrwl/schematics  
2  
3 create-nx-workspace myworkspace
```

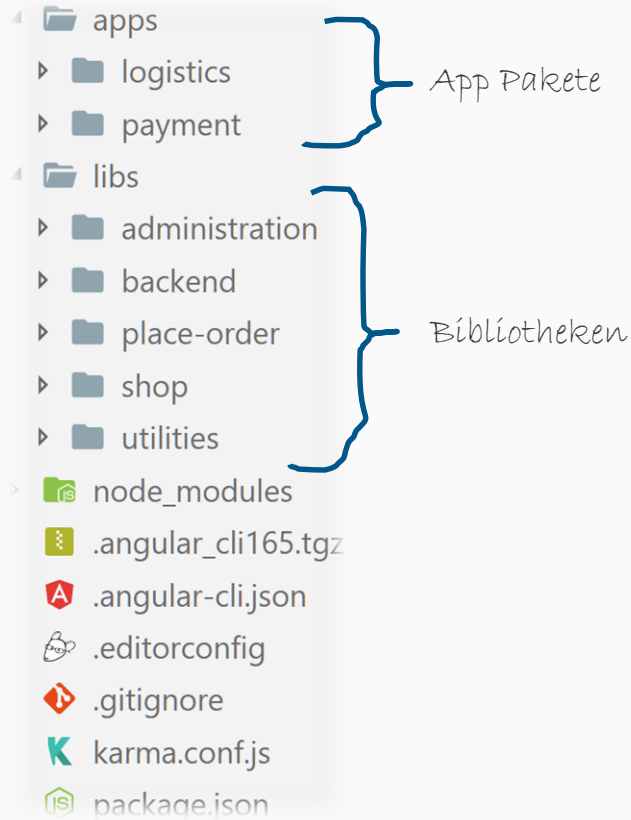
Enterprisify CLI-Projects

```
1 npm install @nrwl/nx
2
3 # @nrwl/schematics konfigurieren
4 # .angular-cli.json -> schematics
5 ng g workspace <name-of-current-app>
```

Bestehende Projekte können zu einem Workspace migriert werden.

A dashed blue line originates from the text 'Bestehende Projekte können zu einem Workspace migriert werden.' and points downwards and to the left, ending near the terminal output of the command 'ng g workspace <name-of-current-app>'. This indicates that the command is used to migrate existing projects into a workspace.

Für Projekte mit großen Ambitionen



```
1 ng generate app <App>
```

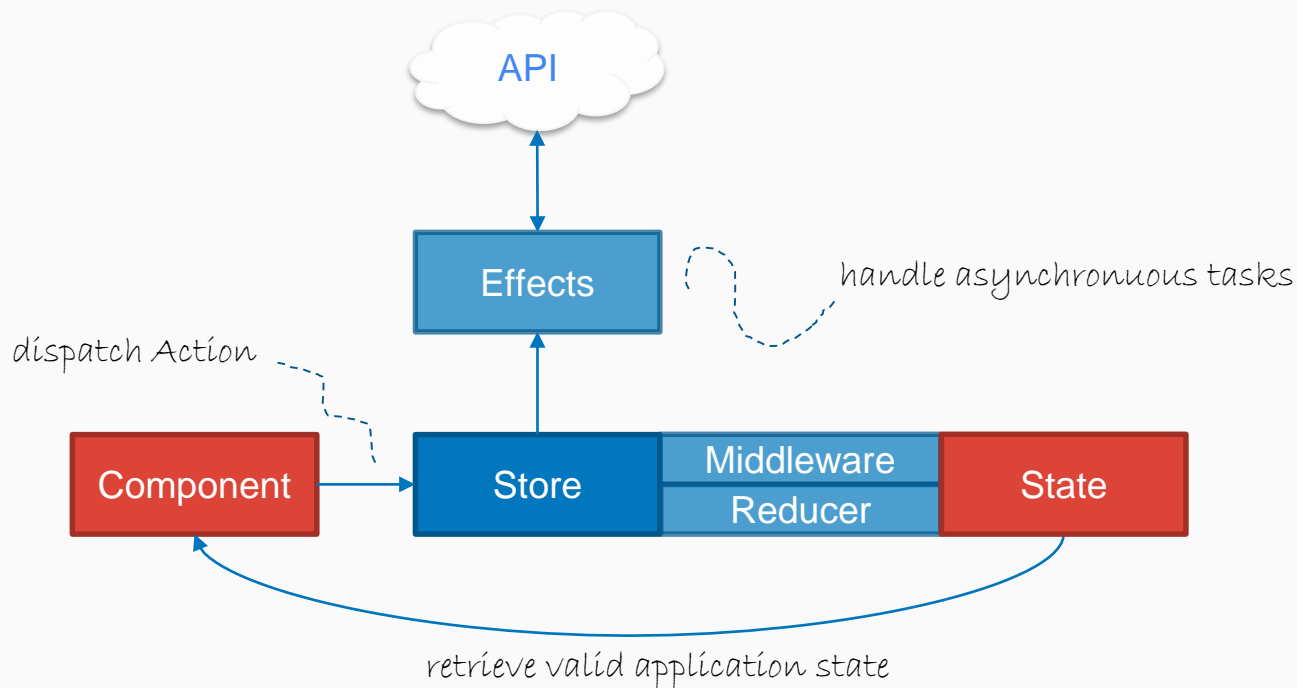
```
1 ng generate lib <Bibliothek>
```

Angular Playbook



<https://angularplaybook.com/>

Redux



Teile und herrsche

- Einfacher Tipp, der das Programmieren vereinfacht.
- Komponenten lassen sich klassifizieren.
- Von diesem Ansatz profitieren **Projektstruktur** und **Architektur**.

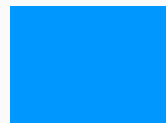
Presentational Components

- Daten visualisieren
- Oberflächenlogik umsetzen
 - Animations
 - Events
- Konsumiert keine Datenservices
- Weisen eine hohe Wiederverwendbarkeit auf

Text



Image



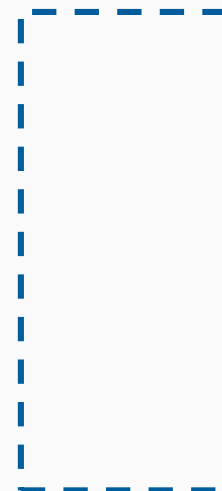
Multimedia



@Input



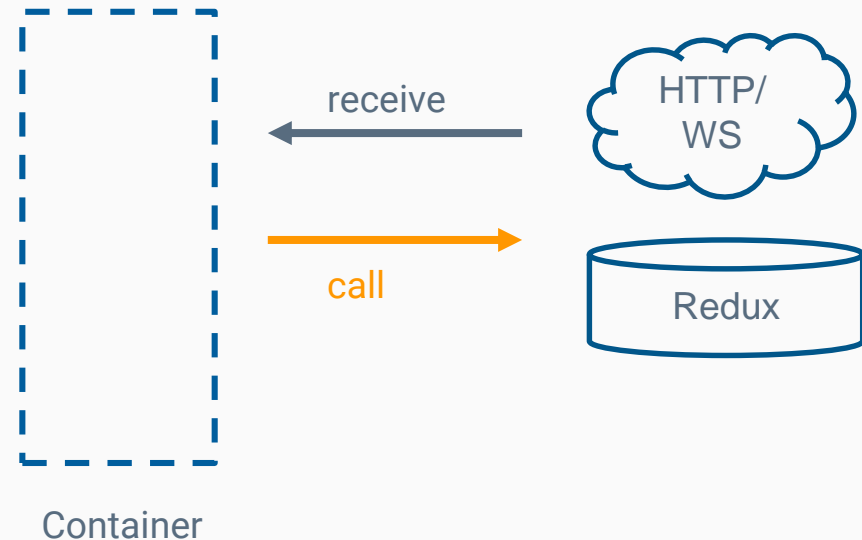
@Output



Container

Container Components

- Komponent orchestrieren
 - Host Components
 - Kommunikation
- Nutzt Datenservices
- Sind für einen bestimmten Use Case vorgesehen



Im Vergleich

	Presentational Components	Container Components
Wiederverwendbarkeitsgrad	Hoch	Niedrig
Austauschbarkeitsgrad	Hoch	Niedrig
Abhängigkeiten zu Services*	Keine	1-n
Abhängigkeiten zu Redux	Ja	Nein
Code-Komplexität	Niedrig	Mäßig

PWA

Motivation

- Aufwand des Deployments reduzieren

- Ladezeiten verbessern

caching

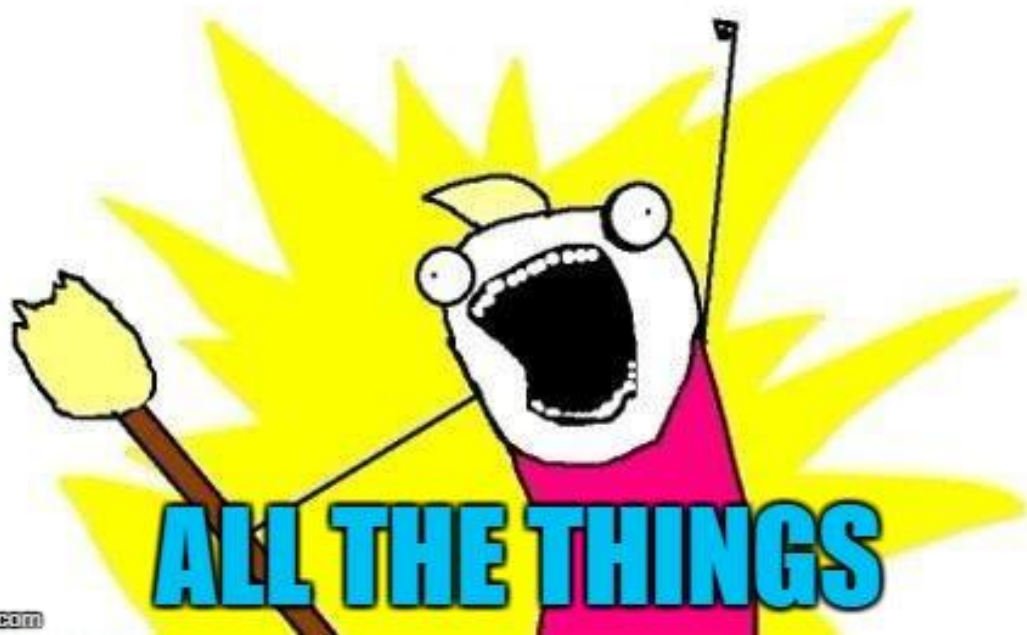
- Offline arbeiten

AppShell

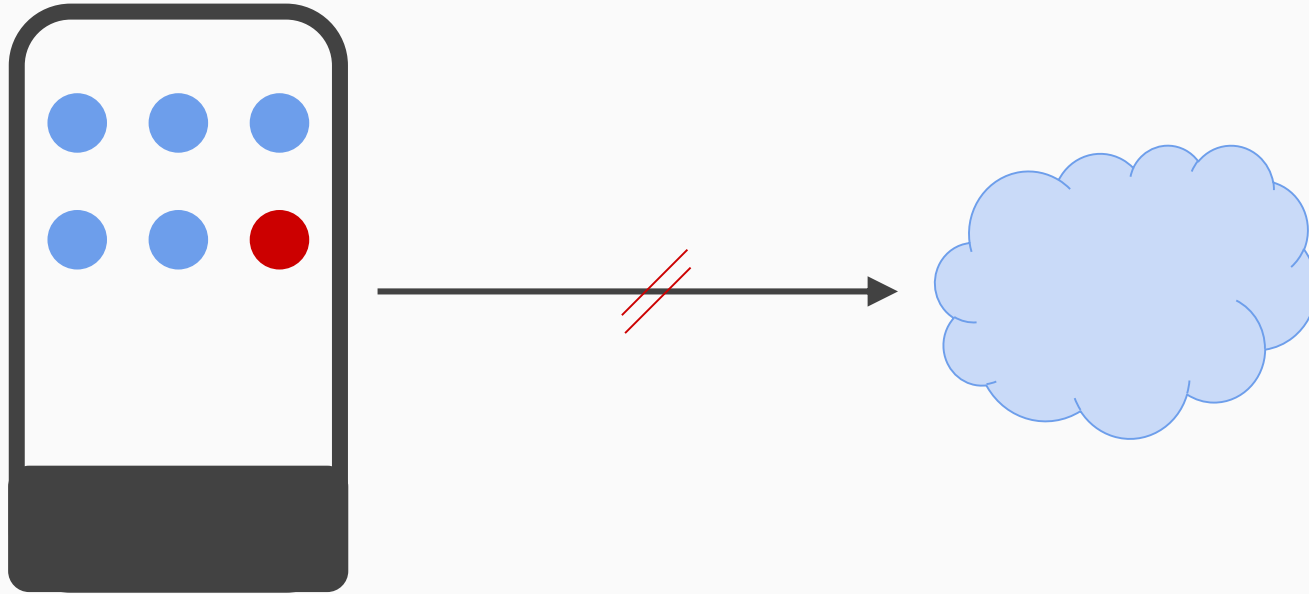
- Nutzererfahrung intensivieren

Push-Notifications

PWA



Problem von HTML 5 Webapps



● Native App

● HTML5 App

Metadaten für die WebApp



```
1 {  
2   "manifest_version": 2,  
3   "name": "My Extension",  
4   "version": "versionString",  
5  
6   "default_locale": "en",  
7   "description": "A plain text description",  
8   "icons": {}  
9 }  
10
```

manifest.json



```
1 npm i @angular/service-worker
```

Service Worker konfigurieren



APIs



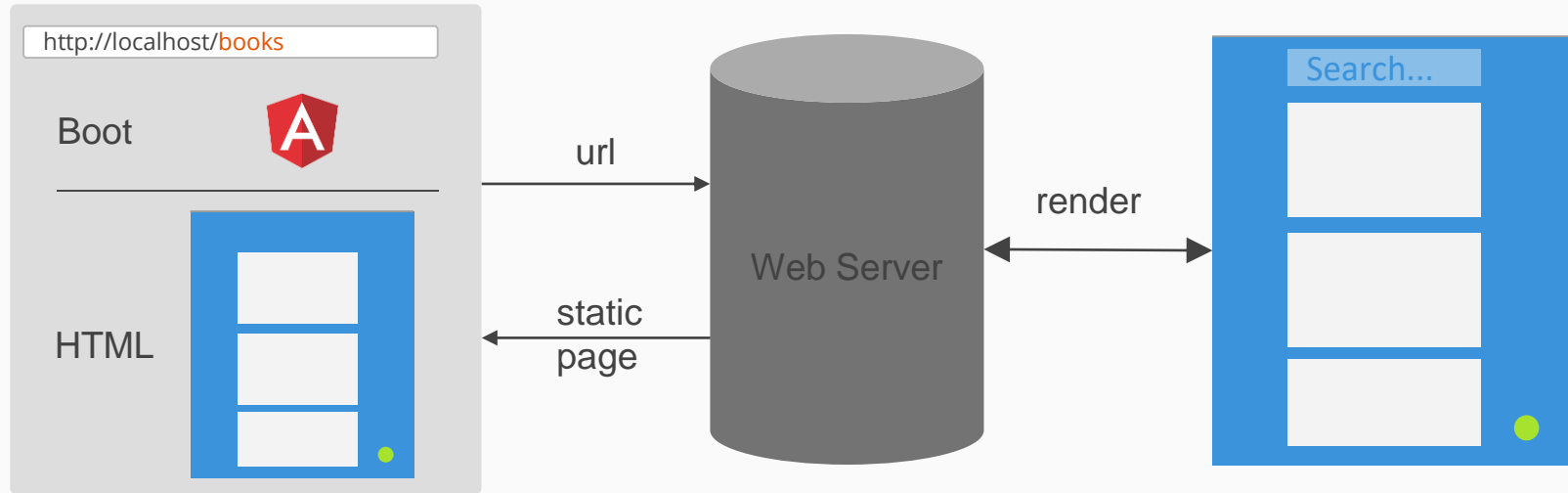
Assets

```
1 {  
2   "dynamic": {  
3     "group": [  
4       "urls": {  
5         "https://www.googleapis.com": {  
6           "match": "prefix"  
7         }  
8       }  
9     ]  
10  },  
11  "static": {  
12    "urls": {  
13      "/index.html": "...f03aea9",  
14      // ...  
15    }  
16  }  
17 }
```

ngsw-manifest.json



Prozess



Zwei Wölfe

Universal Rendering

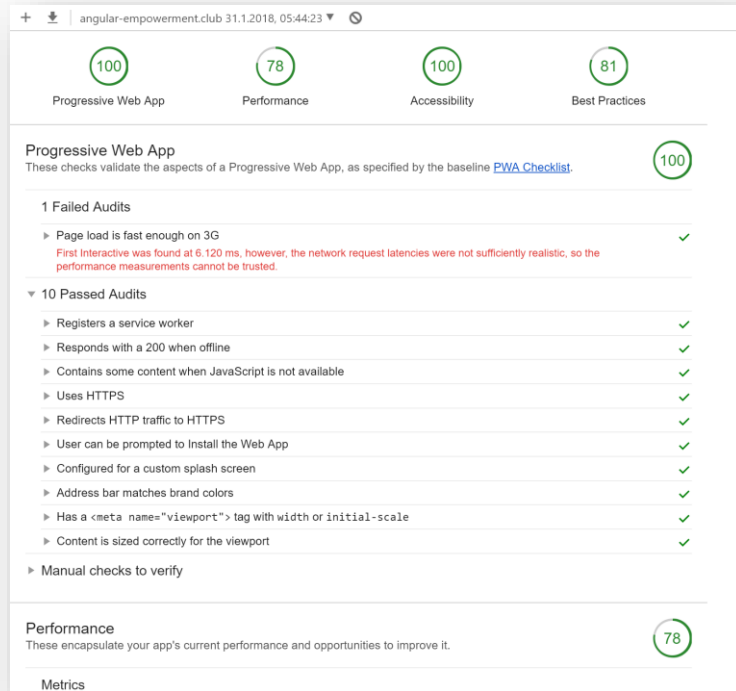
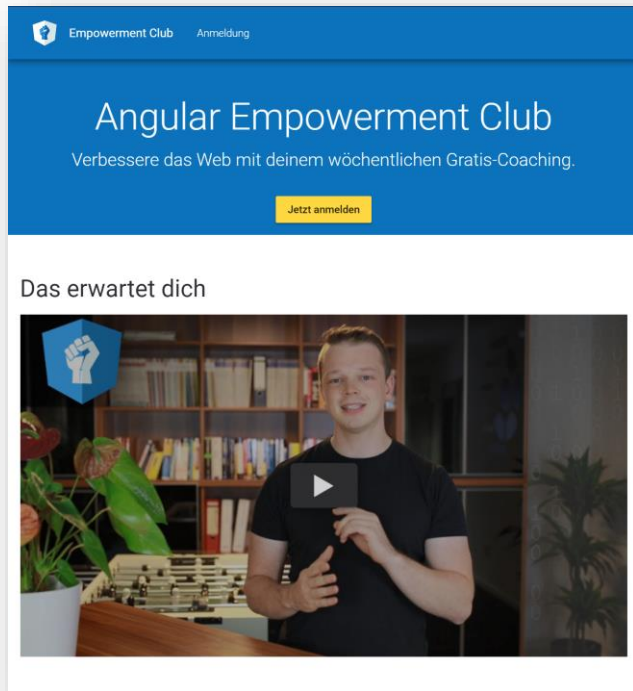
- Rendering per Request
- HTTP-State-Transfer cached API-Requests
- Unterstützt Lazy-Loading

Prerendering

- Rendering zur Compilezeit
- Generierung Statische Websites
- Nutzt Universal Rendering



Audit für Ihre App



In Chrome Developer Tools integriert

Ich freue mich auf
Ihre Fragen.



Gregor Woiwode

<https://angular-empowerment.club>

 @GregOnNet

Ressourcen

 github.com/GregOnNet/enter-the-angular-universe

