

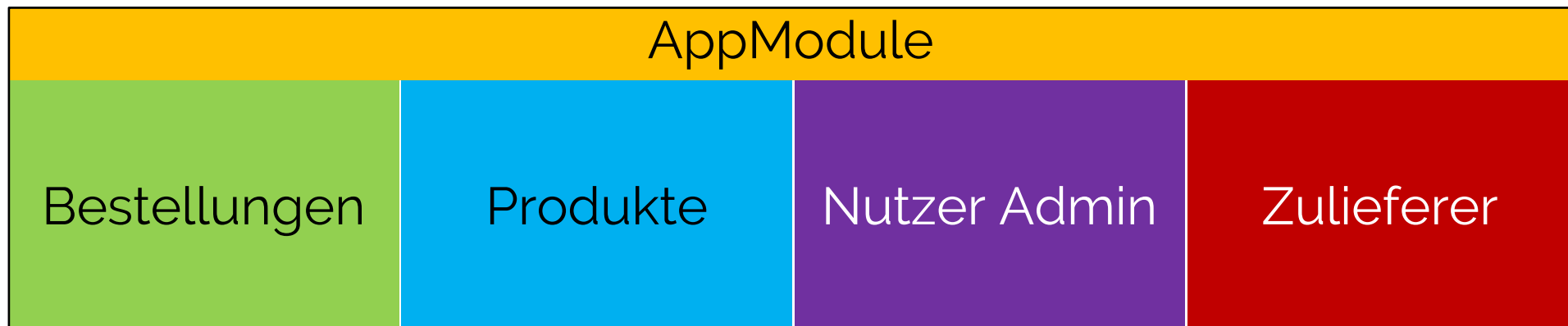
Motivation

- Reduzierung der Startup-Zeiten
- Weniger häufig benötigte Teile der Applikation nachladen
- Datenschonend für mobile Plattformen

Angular Monolith

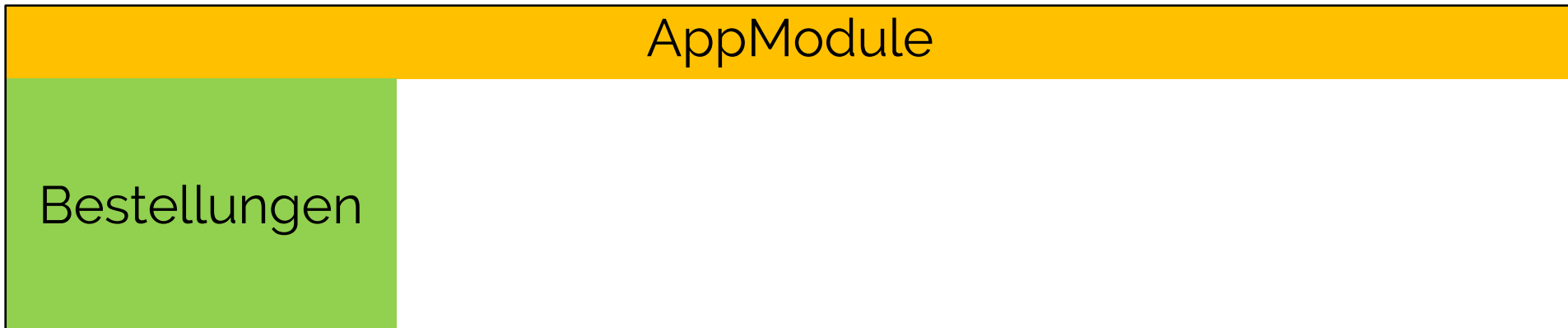
- Alle Module sind in einem JS File (Bundle)
 - Großes JS File
 - Lange Downloadzeiten
 - Manche Module werden selten benötigt

<http://mein-super-laden/admin>



Prinzip

<http://mein-super-laden/admin/produkte>



- Module werden als eigenständige Bundles erstellt und bei Bedarf nachgeladen → Reduzierung der initialen Datenmenge

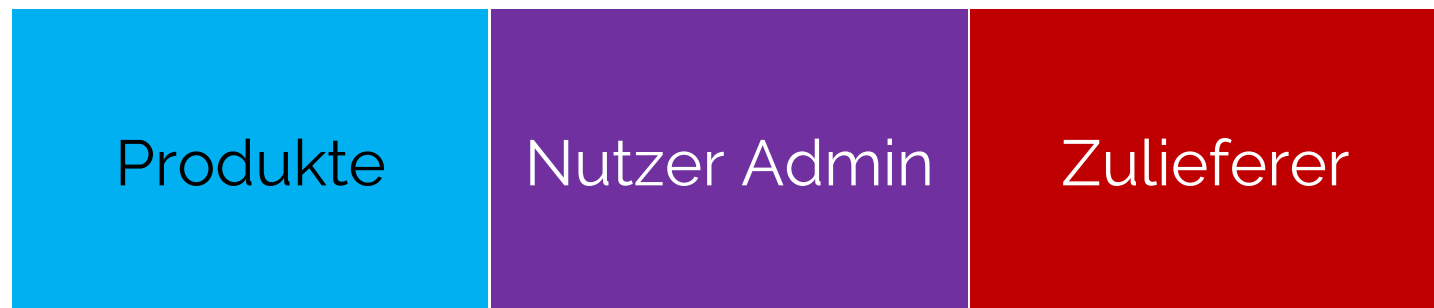
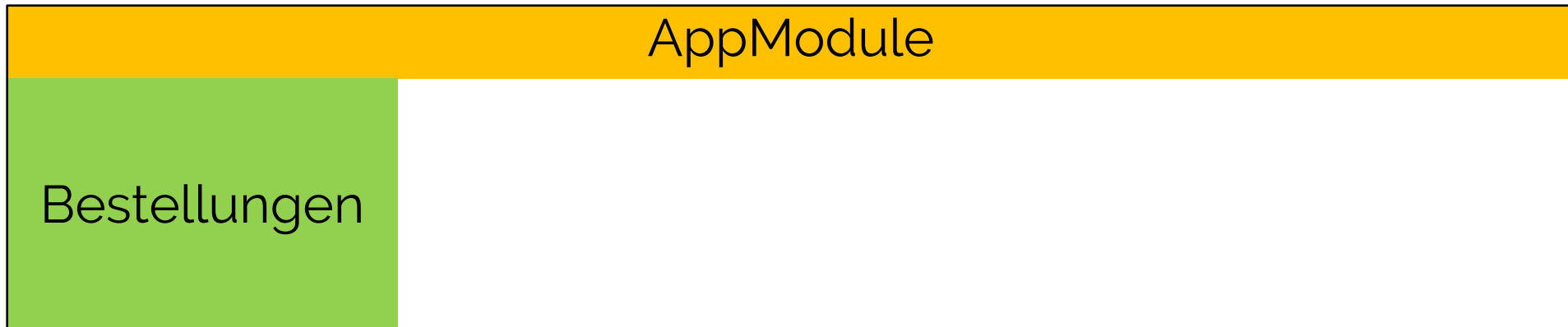
Routingänderungen

```
const appRoutes: Routes = [  
  {path: 'produkte', loadChildren: 'app/produkte/products.module#ProductModule'},  
  {path: 'bestellungen', component: OrderComponent}, ...  
];  
@NgModule({  
  imports: [..., RouterModule.forRoot(appRoutes)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule {}
```

```
const productRoutes: Routes = [  
  {path: '',  
    component: ProductComponent,  
    children: [ {path: ':id', component: ProductDetailComponent} ]  
  }];  
@NgModule({  
  imports: [RouterModule.forChild(productRoutes)],  
  exports: [RouterModule]  
})  
export class ProductRoutingModule {}
```

Preloading

<http://mein-super-laden/admin>



- Initialer Download schneller → Reduktion der Wartezeit für User
- Nach Init. Download → Restliche Module werden nachgeladen

Preloading Implementierung

```
import { PreloadAllModules } from '@angular/router';
```

```
const routeOptions = {  
  preloadingStrategy: PreloadAllModules  
}
```

```
@NgModule({  
  imports: [RouterModule.forRoot(appRoutes, routeOptions)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule {}
```

- Eigene Strategien können erstellt werden, z.B. verschiedene Module für verschiedene Nutzer laden etc.

Zusammenfassung

Um was ging es in diesem Modul?

- Aufteilung einer Applikation in einzelne Module
- Nachladen von Modulen um initiale Ladezeiten zu reduzieren
- Preloading von ausgelagerten Modulen

Wozu brauche ich das? Was werde ich damit machen?

- Verbessern der UX für Nutzer
- Datenvolumen von mobilen Nutzern schonen

Kontrollfragen

- Welche Voraussetzung muss erfüllt sein, um Lazy Loading einzusetzen?
- Welche Änderungen müssen in welchen Teilen von Angular vorgenommen werden?
- Wann ist es sinnvoll, Lazy Loading einzusetzen? Wann nicht?
- Welche Strategien gibt es beim Preloading von externen Modulen?

