



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Performance Tuning

[angular-architects.io](https://angular-architects.io)

## Turbo Button



# Quick Wins

Bundling

Minification

enableProdMode()

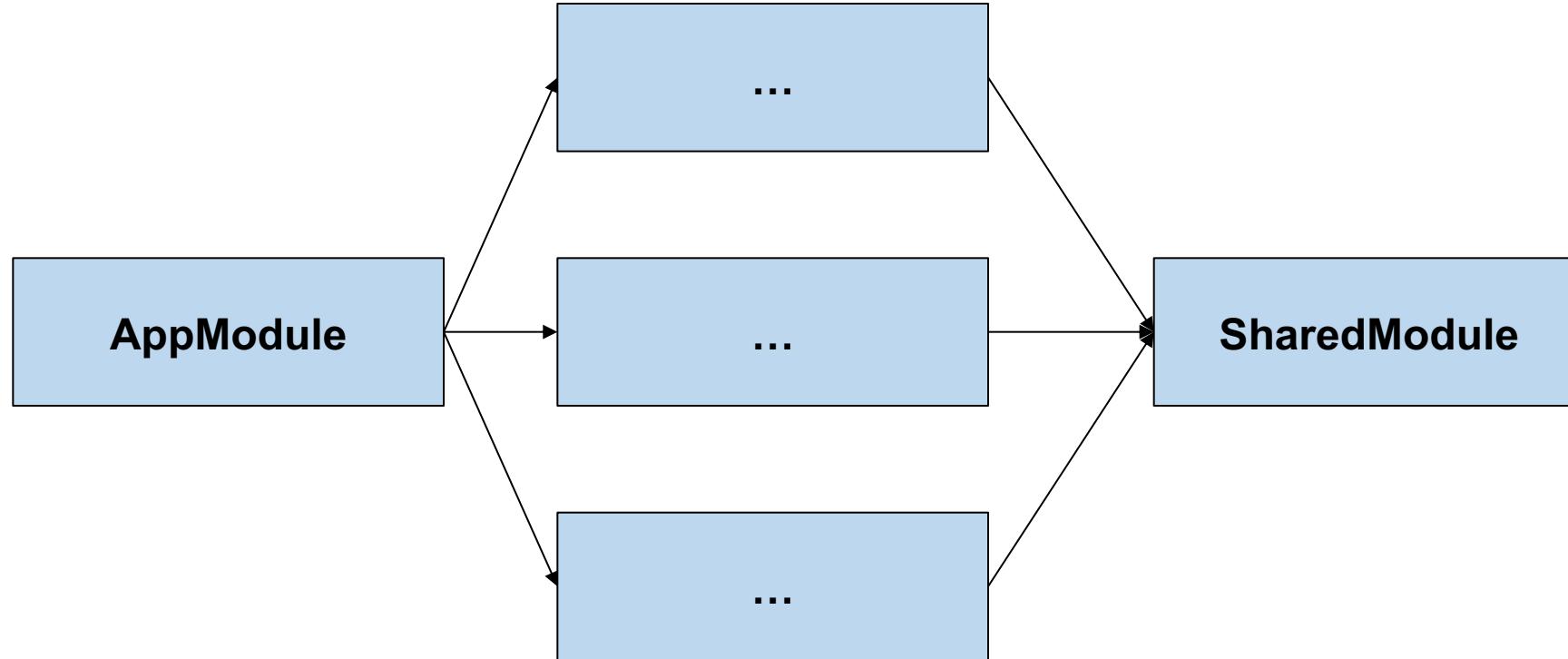
# Contents

- Lazy Loading and Preloading
- Performance for Data Binding with OnPush
- AOT and Tree Shaking

# Lazy Loading



# Module Structure

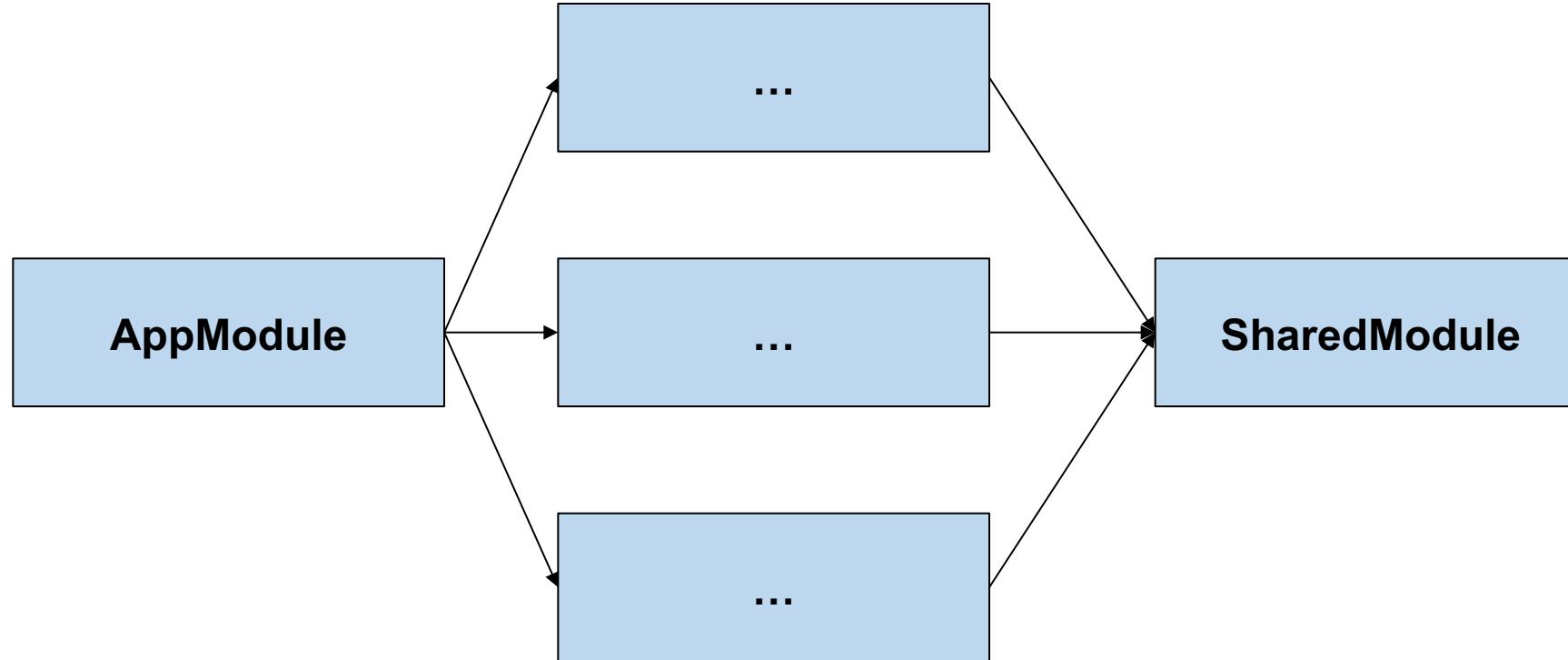


**Root Module**

**Feature Modules**

**Shared Module**

# Lazy Loading



**Root Module**

**Feature Modules**

**Shared Module**

# Root Module with Lazy Loading

```
const APP_ROUTE_CONFIG: Routes = [
  {
    path: 'home',
    component: HomeComponent
  },
  {
    path: 'flights',
    loadChildren: () => import('./flight-booking/flight-booking.module')
      .then(m => m.FlightBookingModule)
  }
];
```

# Routes for "lazy" Module

```
const FLIGHT_ROUTES = [
  {
    path: '',
    component: FlightBookingComponent,
    [...]
  },
  [...]
}
```

# Routes for "lazy" Module

```
const FLIGHT_ROUTES =      [
  {
    path: 'subroute',
    component: FlightBookingComponent,
    [...]
  },
  [...]
}
```

flight-booking/ subroute

Triggers Lazy Loading w/ loadChildren

# Lazy Loading

- Lazy Loading means: Loading it later
- Better startup performance
- Delay during execution for loading on demand

# Preloading



# Idea

- Module that might be needed later are loaded after the application started
- When module is needed it is available immediately

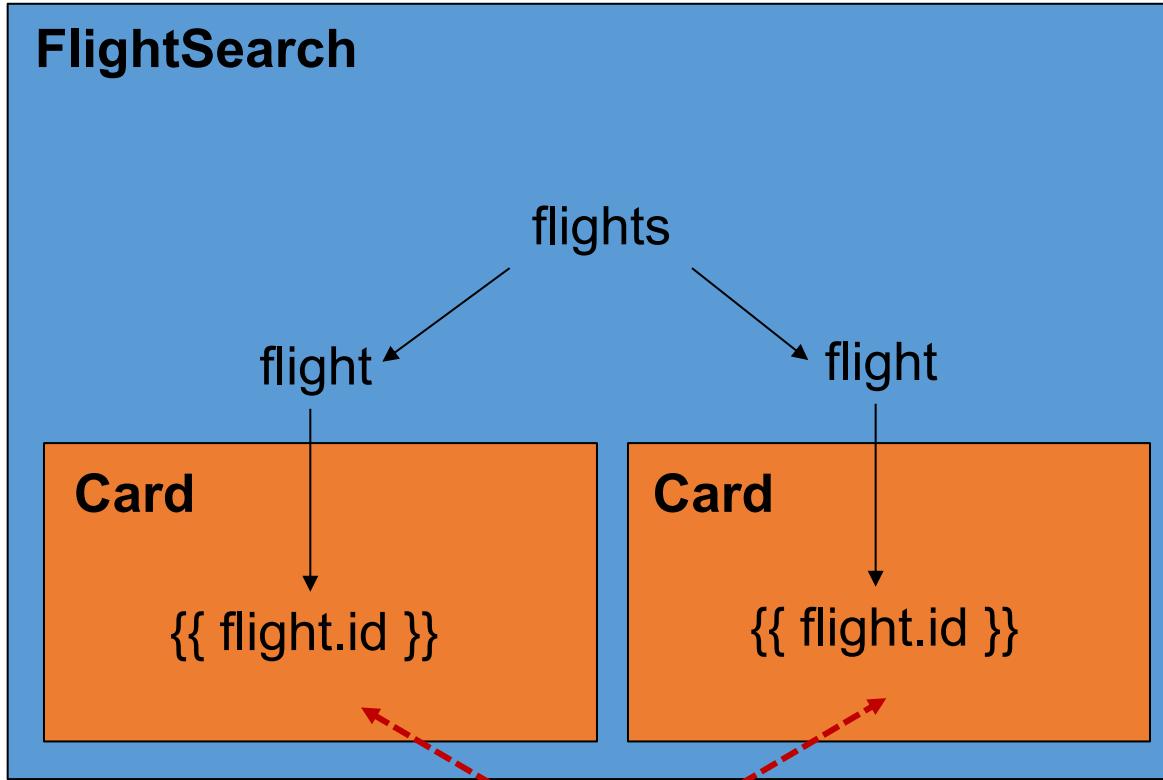
# Activate Preloading

```
...
imports: [
  [...]
  RouterModule.forRoot(
    ROUTE_CONFIG,
    { preloadingStrategy: PreloadAllModules });
]
...
```



# Performance- Tuning with OnPush

# OnPush



Angular just checks when “notified”



ANGULAR  
ARCHITECTS

INSIDE KNOWLEDGE



SOFTWARE  
ARCHITECT

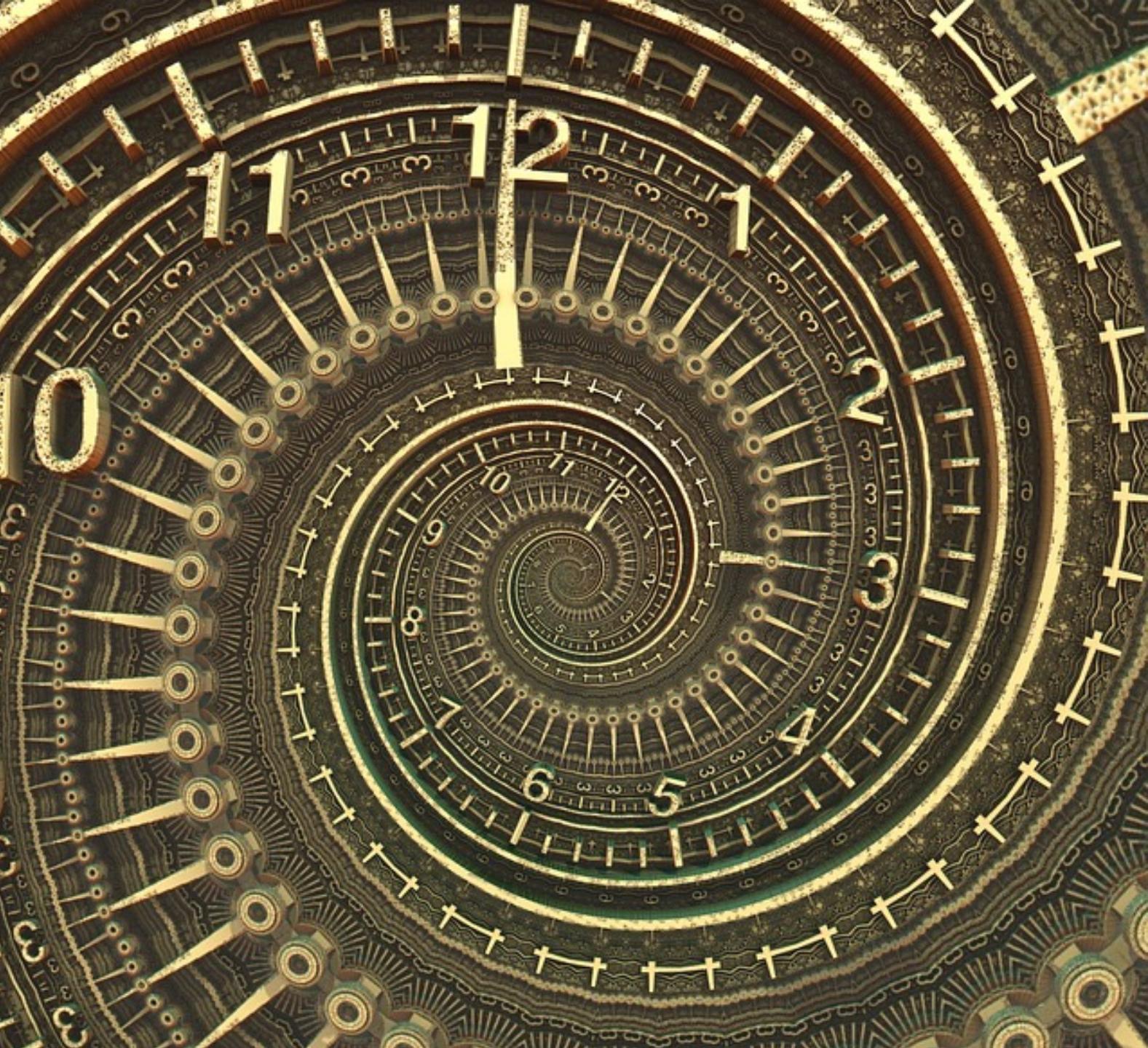
# "Notify" about change?

- Change bound data (@Input)
  - OnPush: Angular just compares the object reference!
  - e. g. oldFlight === newFlight
- Raise event within the component
- Notify a bound observable
  - {{ flights\$ | async }}
- Trigger it manually
  - Don't do this at home ;-)
  - At least: Try to avoid this

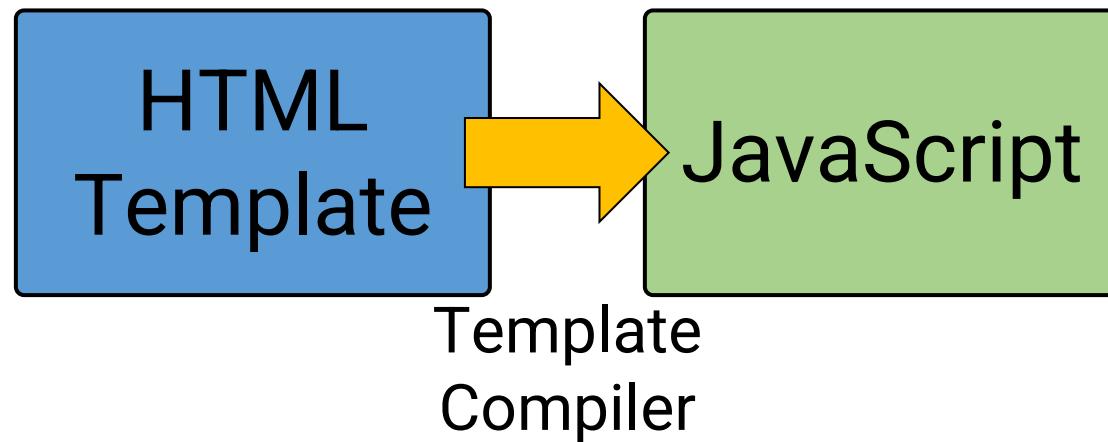
# Activate OnPush

```
@Component({
  [...]
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class FlightCard {
  [...]
  @Input() flight;
}
```

# Ahead of Time (AOT) Compilation



# Angular Compiler



# Approaches

- JIT: Just in Time, at runtime
- AOT: Ahead of Time, during build

# Advantages of AOT

- Better Startup-Performance
- Smaller Bundles: You don't need to include the compiler!
- Tools can easier analyse the code
  - Remove unneeded parts of frameworks
  - Tree Shaking

# Angular CLI

- ng build --prod
- @ngtools/webpack with AngularCompilerPlugin
- Can be used without CLI too

# Production Mode without AOT (no lazy loading)

FLIGHT42 Flight42

HOME Flight Search Passenger Search

FLIGHT BOOKING

## Flight Search

Elements Console Sources Network **Performance** Memory Application Security Audits Augury > ... X

Screenshots  Memory

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms

FPS CPU NET

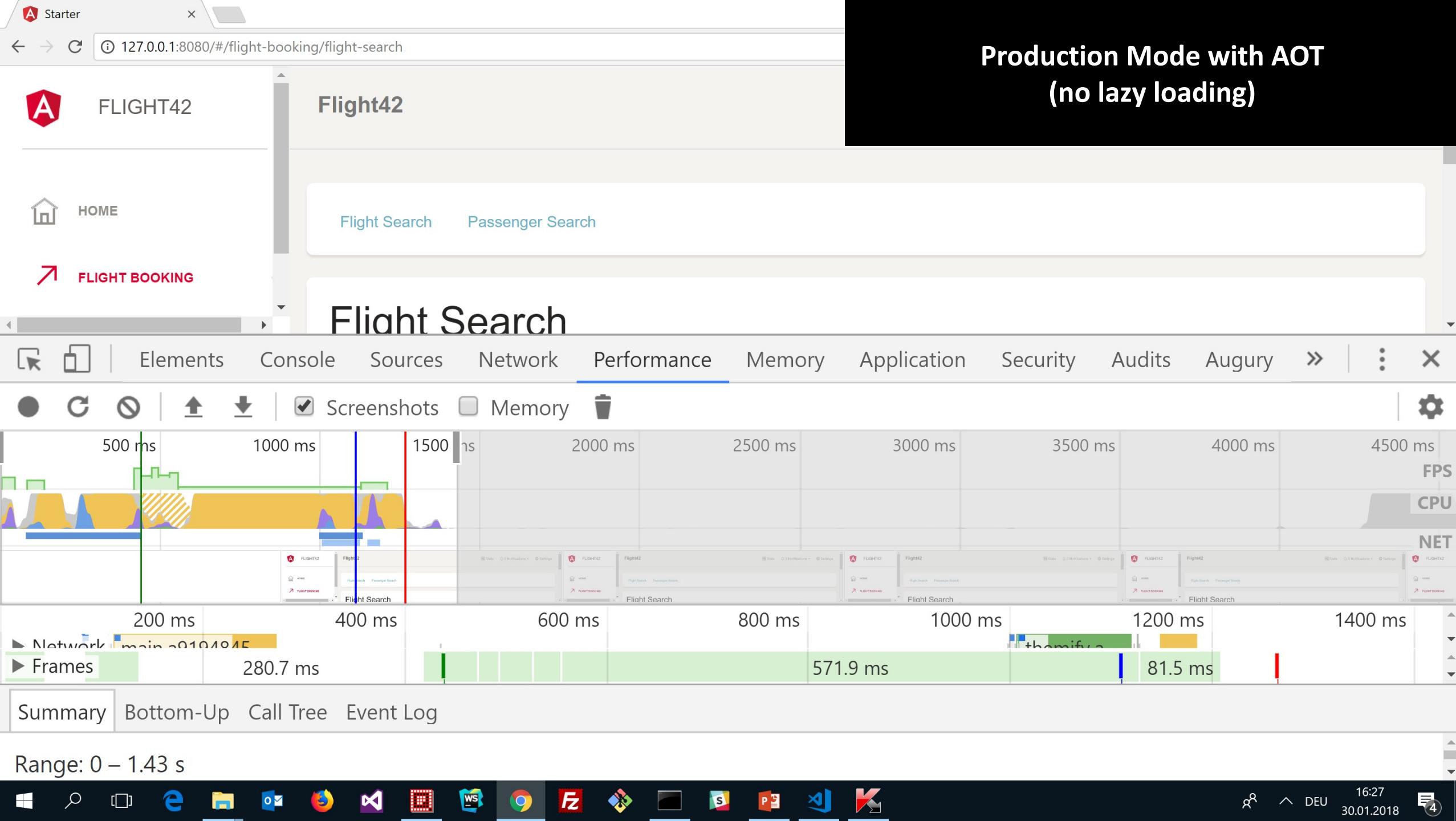
1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms

Network Frames 2234.0 ms

Summary Bottom-Up Call Tree Event Log

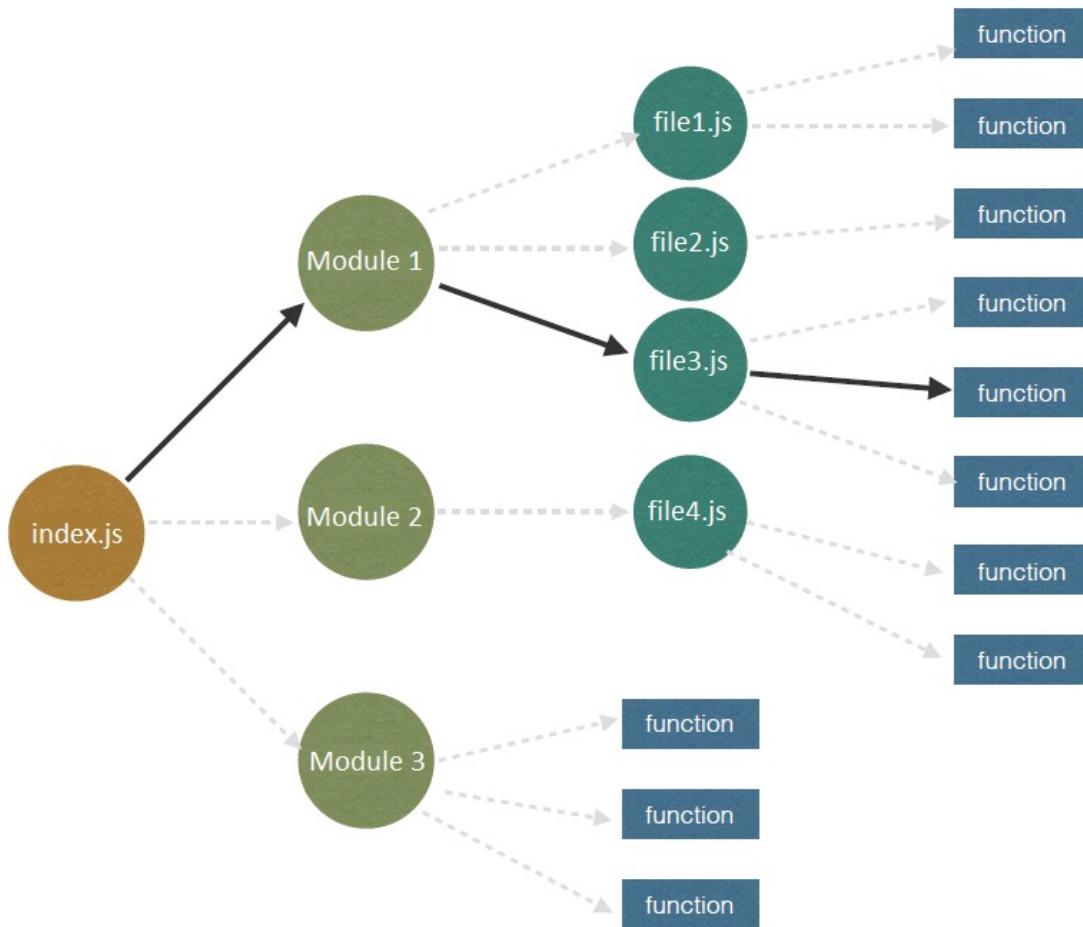
Range: 0 – 6.93 s

16:30 DEU 30.01.2018 4



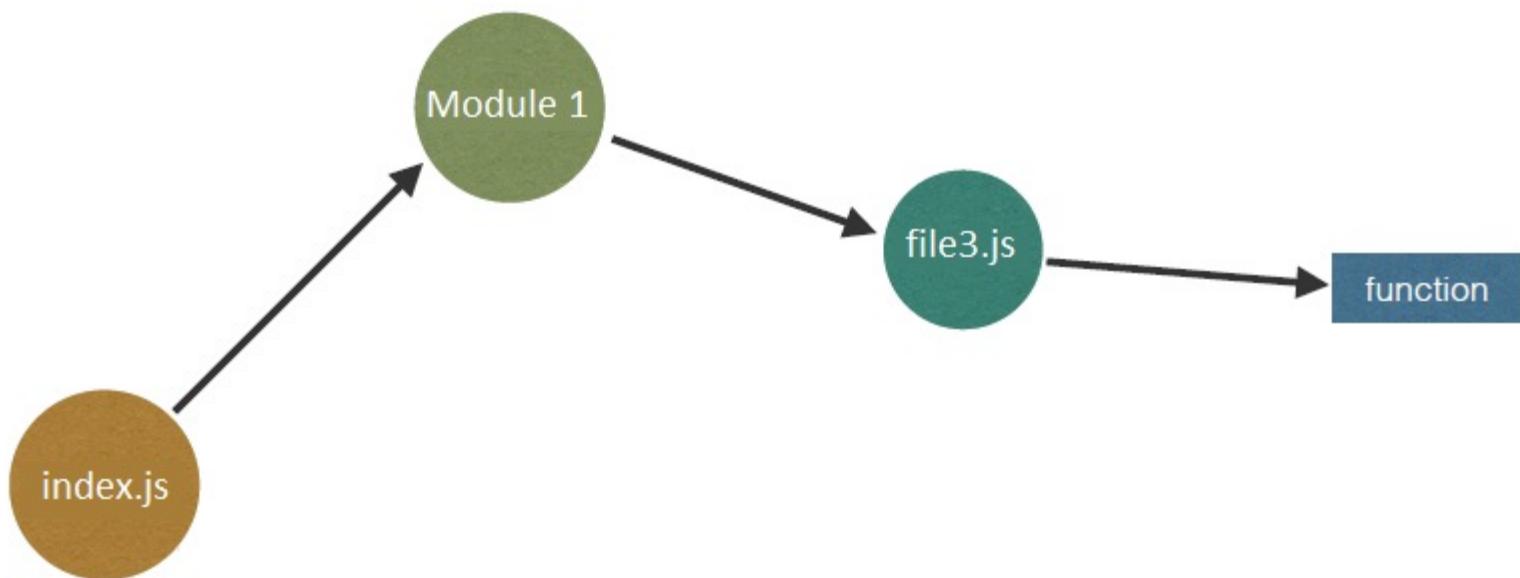
# Tree Shaking

Before Tree Shaking



# Tree Shaking

After Tree Shaking



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE



SOFTWARE  
ARCHITECT

## Bundles without AOT and Tree Shaking

vendor.978ac3ef762178ef4aa8.bundle.js

node\_modules

**JIT Compiler**

@angular

platform-browser-dynamic

esm5

platform-browser-dynamic.js  
+ 1 modulescore  
esm5

core.js

router.js +  
23 modules

common.js http.js

common  
esm5forms.js +  
2 modulesplatform-browser esm5 http  
esm5 http.jsplatform-browser-dynamic  
esm5main.ts  
+ 68  
modules

polyfills.7c4efb87d4ba5dbbc58c.bundle.js

node\_modules

zone.js

dist

zone.js

core.js  
modules

# Lab



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE



# Conclusion

Quick Wins

Lazy Loading  
and  
Preloading

OnPush w/  
Immutables and  
Observables

AOT and Tree  
Shaking



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE

