



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Performance Tuning

Alex Thalhammer.



**Turbo Button**



# Contents

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**





# Lazy Loading

# Why Lazy Loading?

- Load modules when they are needed
- Improve initial load (performance → very important!)



# Root module with Lazy Loading

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

# Routes for feature module

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

```
export const FlugRouterModule =  
  RouterModule.forChild(FLUG_ROUTES);
```



# DEMO







# Preloading



# Idea

- Once the initial load (the important one) is complete load the lazy loaded modules (before they are even used)
- Once the module will come into use it's immediately accessible

# Use preloading (very easy!)

```
export const AppRoutesModule =  
  RouterModule.forRoot(  
    ROUTE_CONFIG,  
    { preloadingStrategy: PreloadAllModules }));
```





# DEMO



# LAB





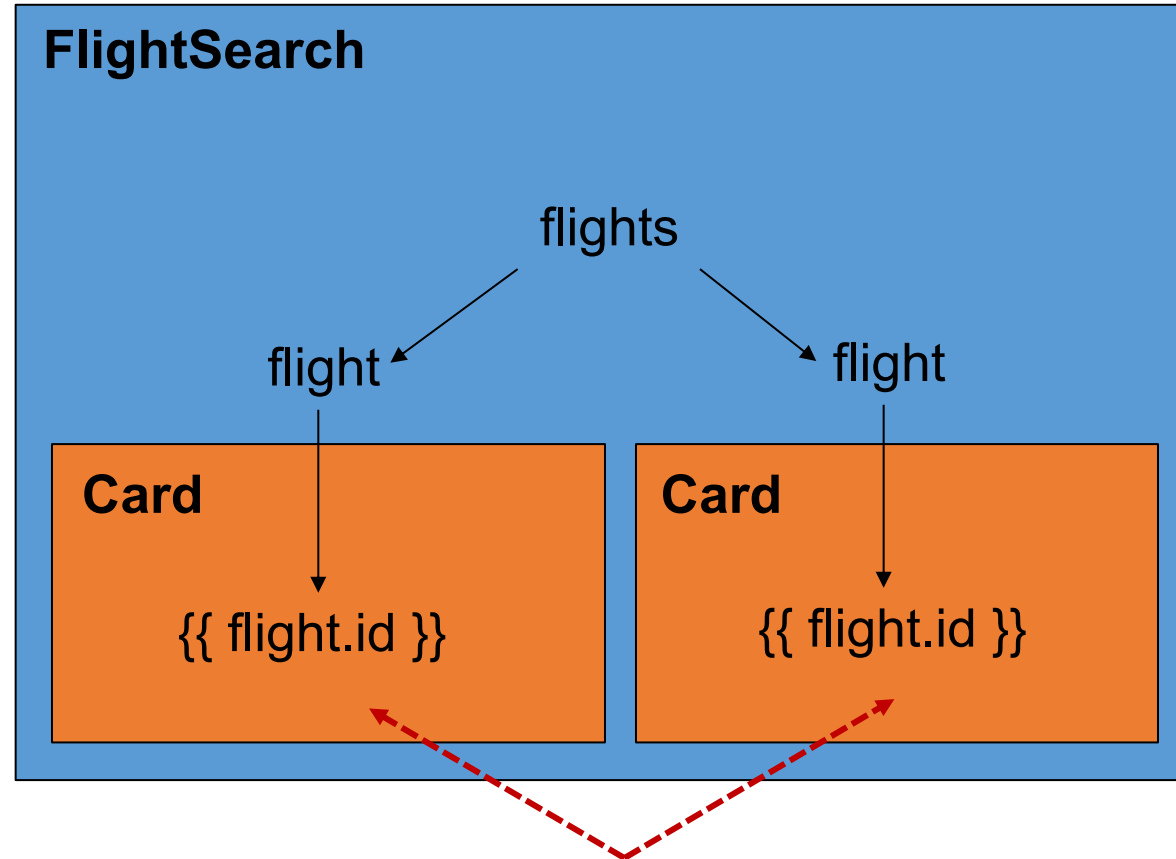
# Performance- Tuning with OnPush



# DEMO



# OnPush



Angular just checks when “notified”

# "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;  
}
```



# DEMO

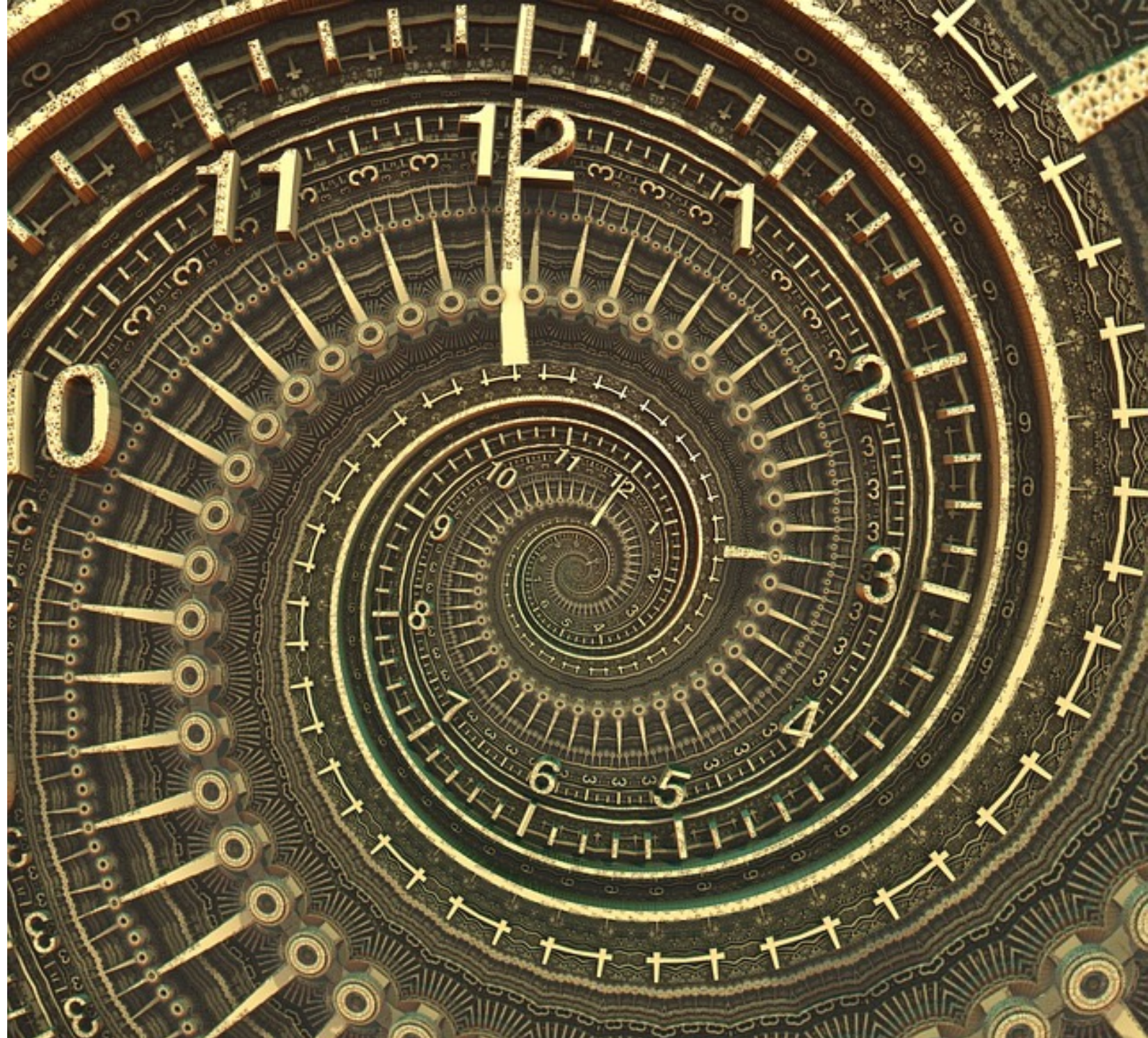


# LAB



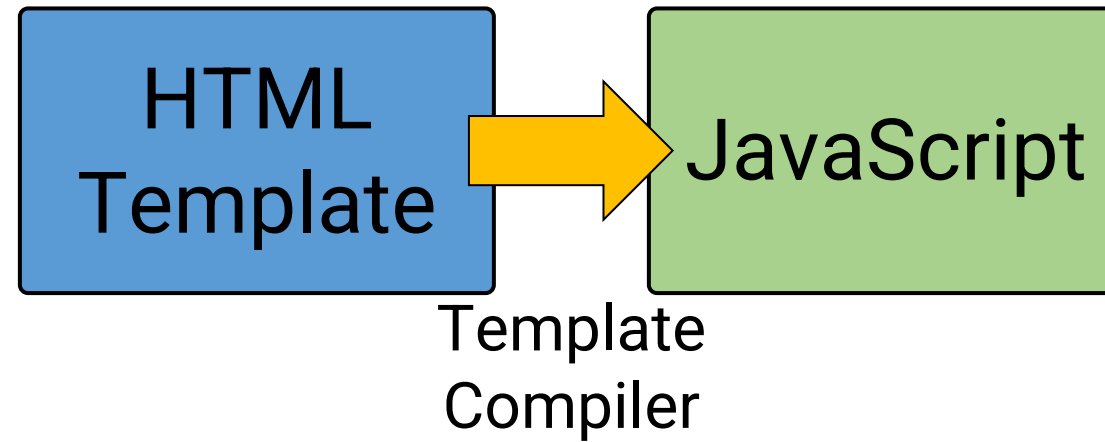


# Ahead of Time (AOT) Compilation





# 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

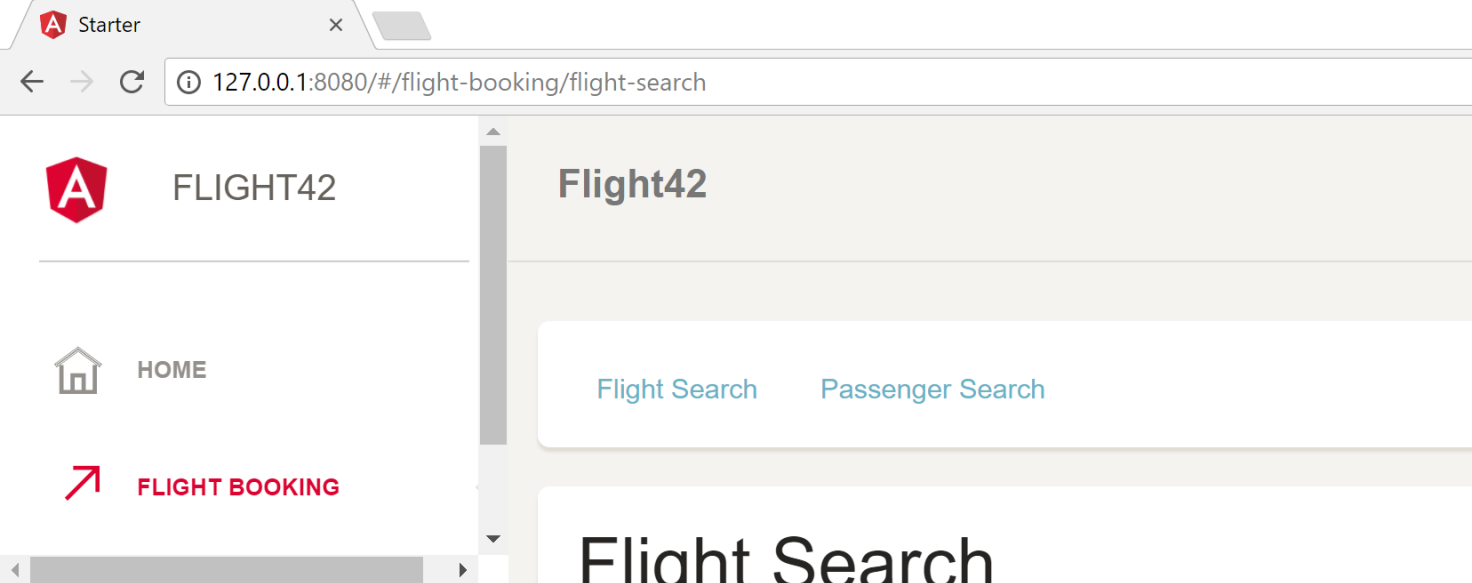
# Angular CLI

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

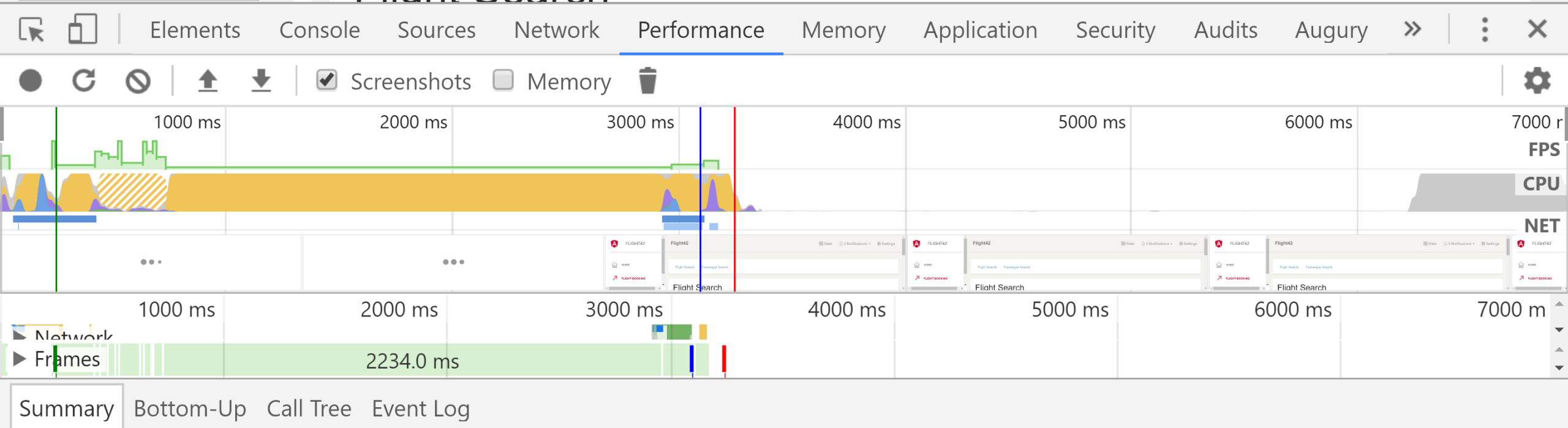


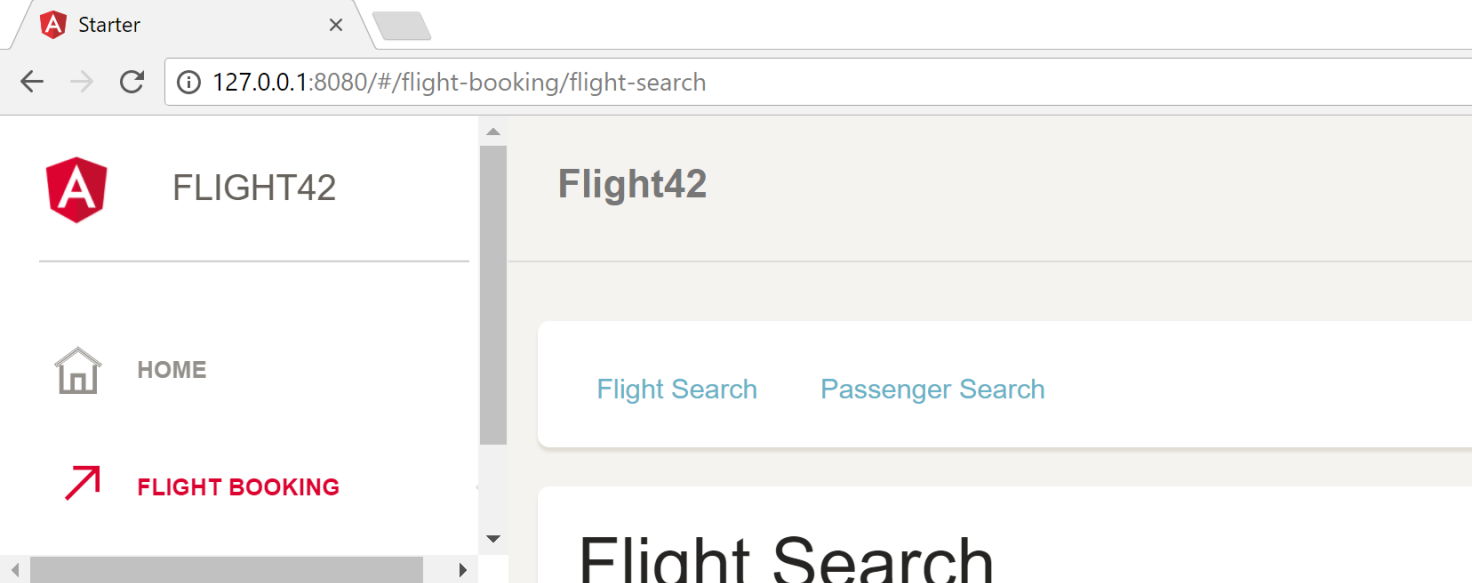
# DEMO



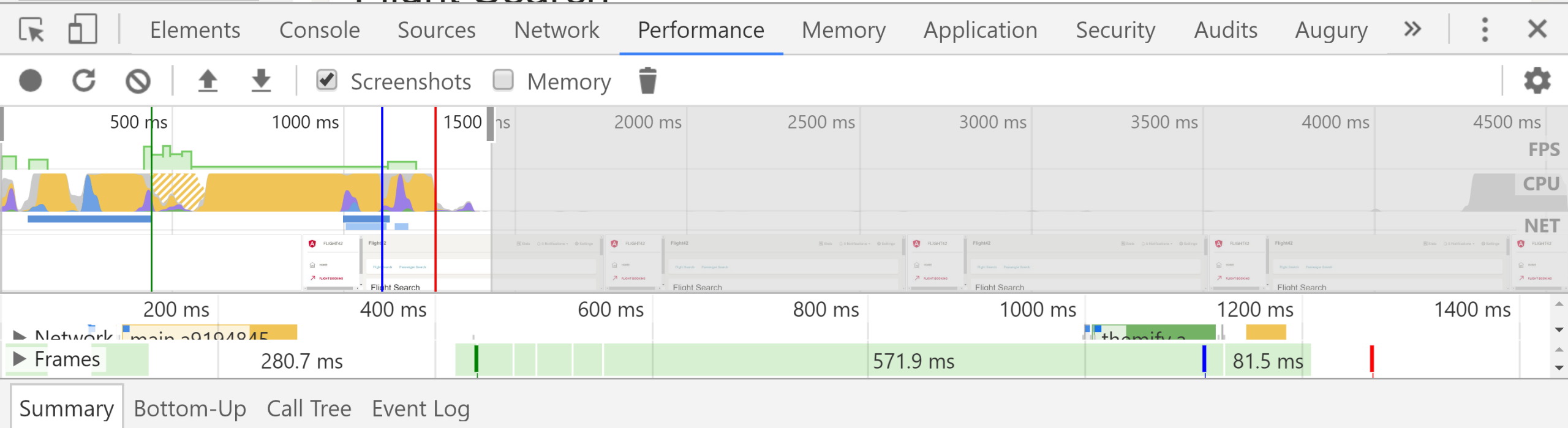


**Production Mode without AOT  
(no lazy loading)**





**Production Mode with AOT  
(no lazy loading)**



# Ivy makes AOT the default 😊

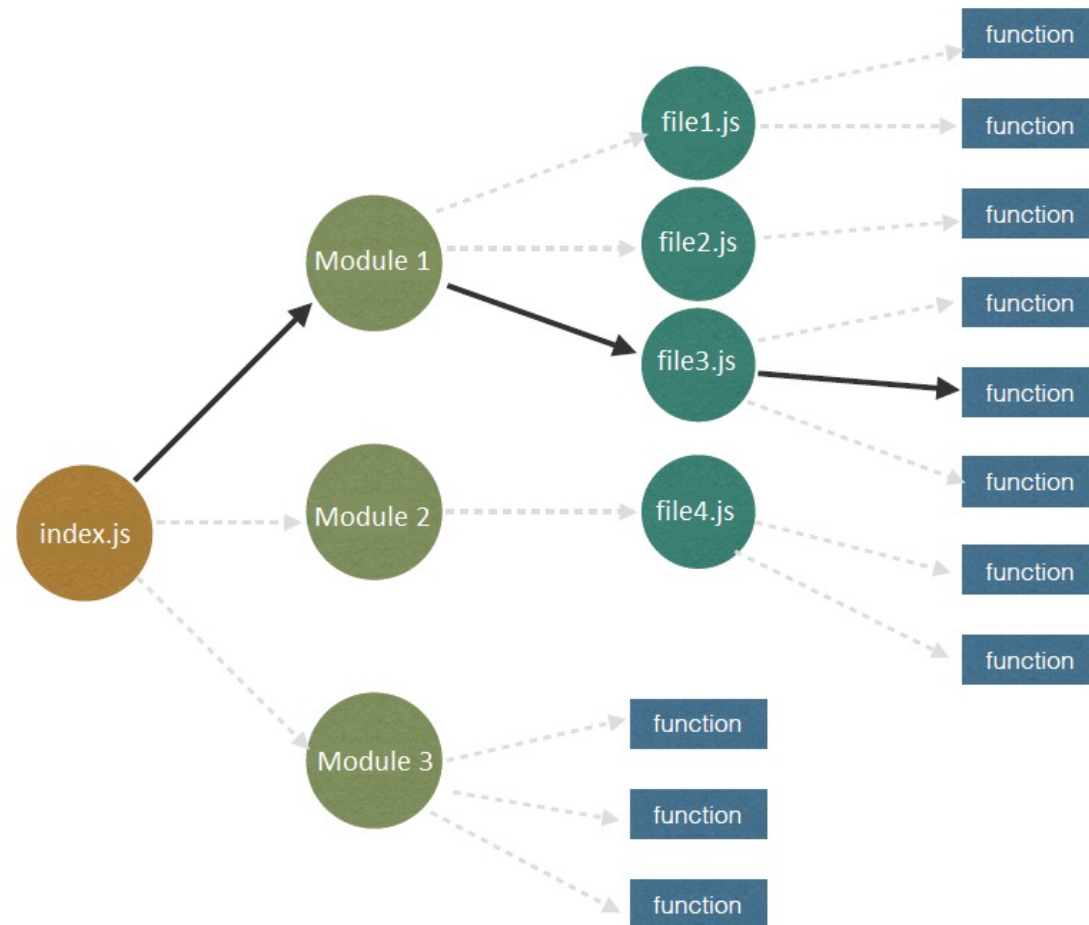
- Ivy also does a lot of under the hood optimization
- No breaking changes, nothing to do from our side 😊
- Angular ViewEngine itself was not tree-shakable
- Angular Ivy is tree-shakable 😊
- Default since NG 10, for libs default since NG 12





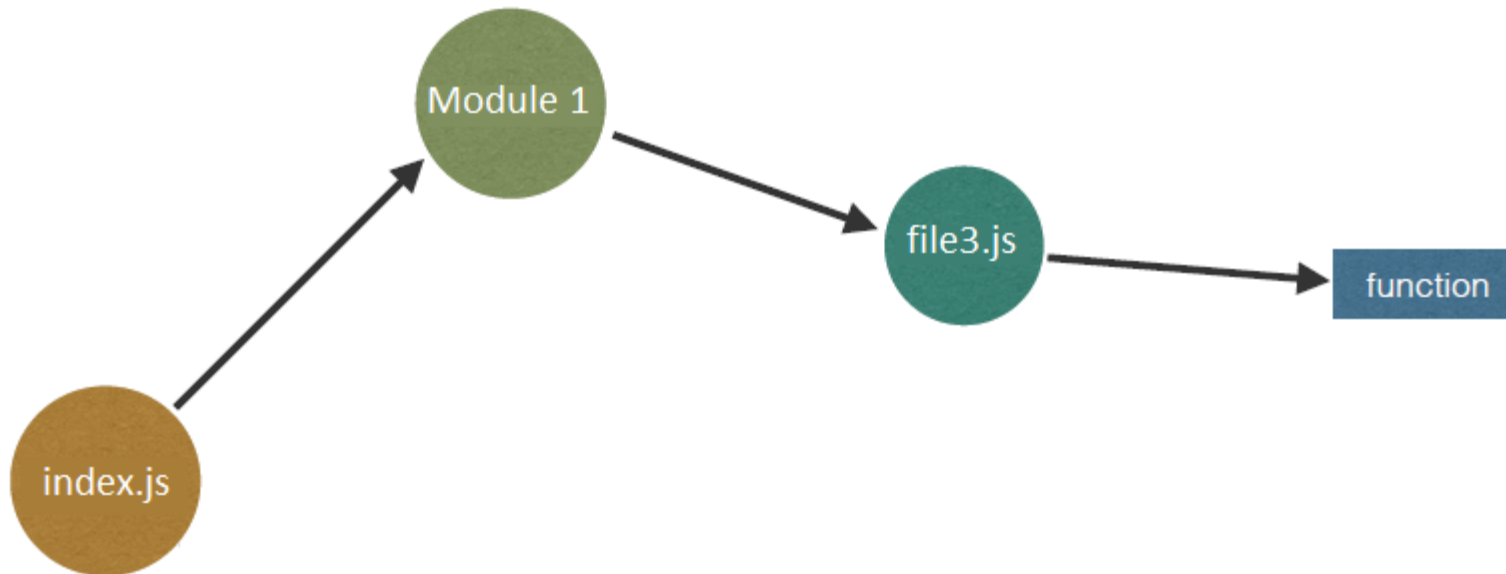
# Tree Shaking

## Before Tree Shaking



# Tree Shaking

After Tree Shaking



# Webpack Bundle Analyzer



## Bundles without AOT and Tree Shaking

vendor.978ac3ef762178ef4aa8.bundle.js

node\_modules

JIT Compiler

@angular

platform-browser-dynamic

esm5

platform-browser-dynamic.js  
+ 1 modules

core

esm5

core.js

router

esm5

router.js +  
23 modules

common

esm5

common.js

http.js

forms

esm5

forms.js +  
2 modules

platform-browser

esm5

platform-browser.js

http

esm5

http.js

rxjs

\_esm5

add

delay.js + 2

modules

switchMap.js

+ 2 modules

fromEvent.js

+ 2 modules

mergeMap.js

+ 2 modules

share.js

+ 4 modules

merge.js

+ 2 modules

...

Subscriber.js

...

mergeMap.js

...

AsyncAction.js

+ 1 modules

ReplaySubject.js

+ 3 modules

Subscription.js

+ 1 modules

Subject.js

...

Observable.js

+ 1 modules

src

main.ts  
+ 68  
modules

polyfills.7c4efb87d4ba5dbbc58c.bundle.js

node\_modules

zone.js

dist

zone.js

core-js

modules



FoamTree

# DEMO



# Conclusion

Lazy Loading

Preloading

OnPush w/  
Immutables and  
Observables

AOT and Tree  
Shaking



# For performance deep dive

Watch this (starting at 8:30):

[https://drive.google.com/file/d/15fmyedJPYSOlV\\_0YvFtg26XGS8tZpZ03/view](https://drive.google.com/file/d/15fmyedJPYSOlV_0YvFtg26XGS8tZpZ03/view)

Repo: <https://github.com/jeffbcross/victor-videos/>

