



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Performance Tuning

angular-architects.io

Turbo Button



Quick Wins

Bundling

Minification

`enableProdMode()`

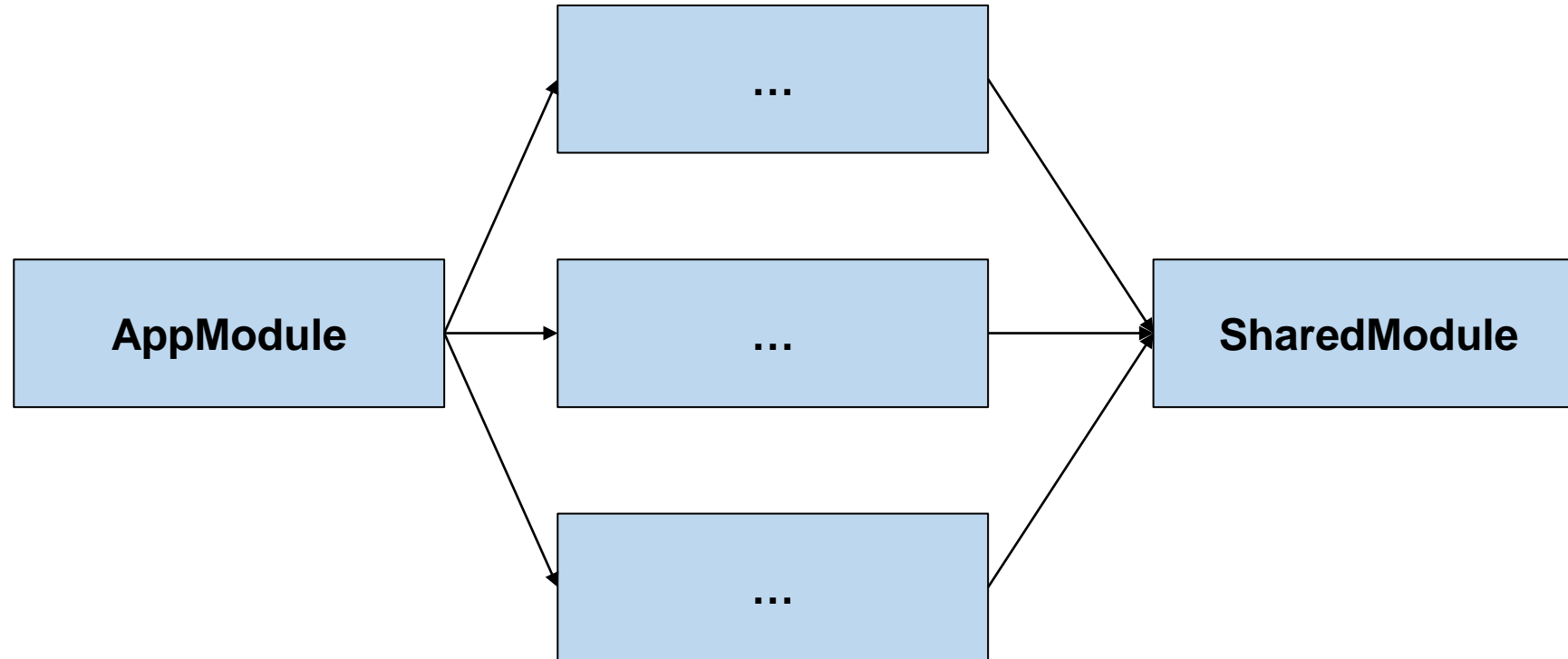
Contents

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

Lazy Loading



Module Structure

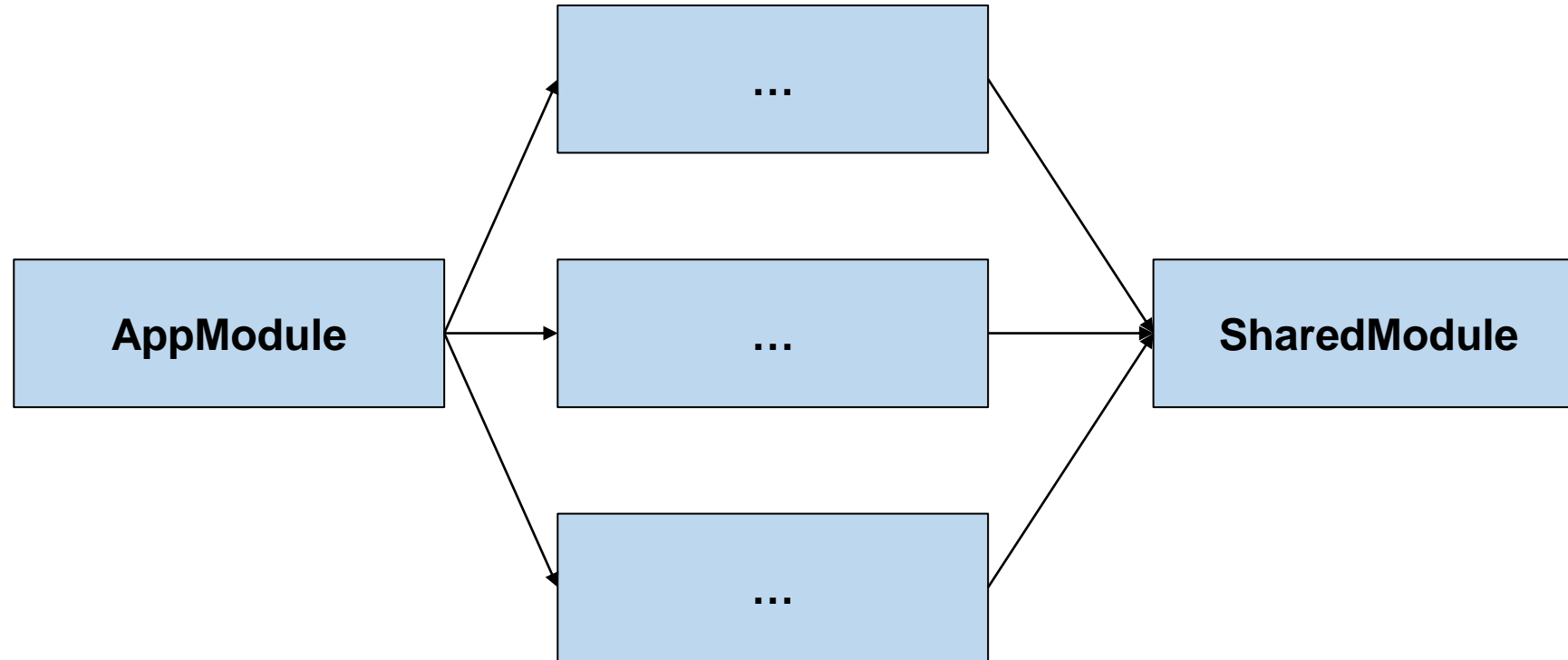


Root Module

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

DEMO

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

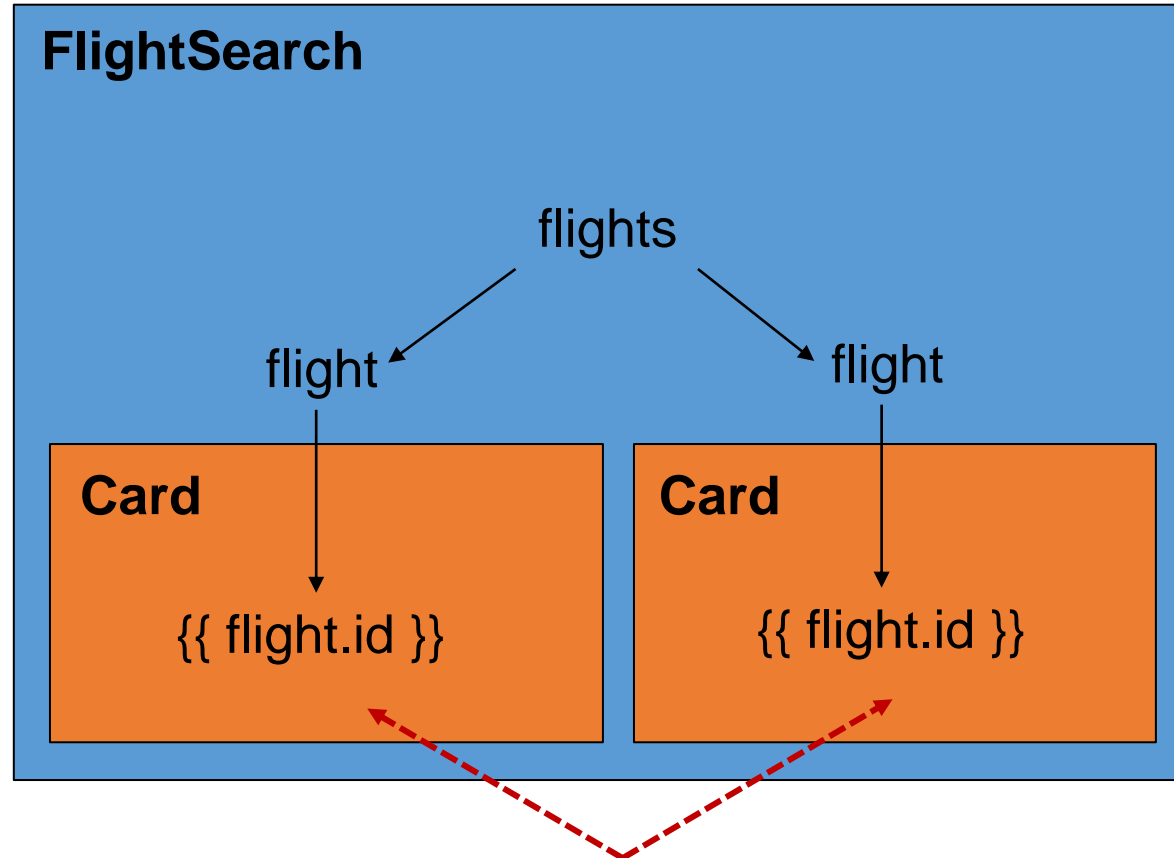
```
...
imports: [
  [...]
  RouterModule.forRoot(
    ROUTE_CONFIG,
    { preloadingStrategy: PreloadAllModules });
]
...
```



Performance- Tuning with OnPush

DEMO

OnPush



Angular just checks when “notified”

"Notify" about change?

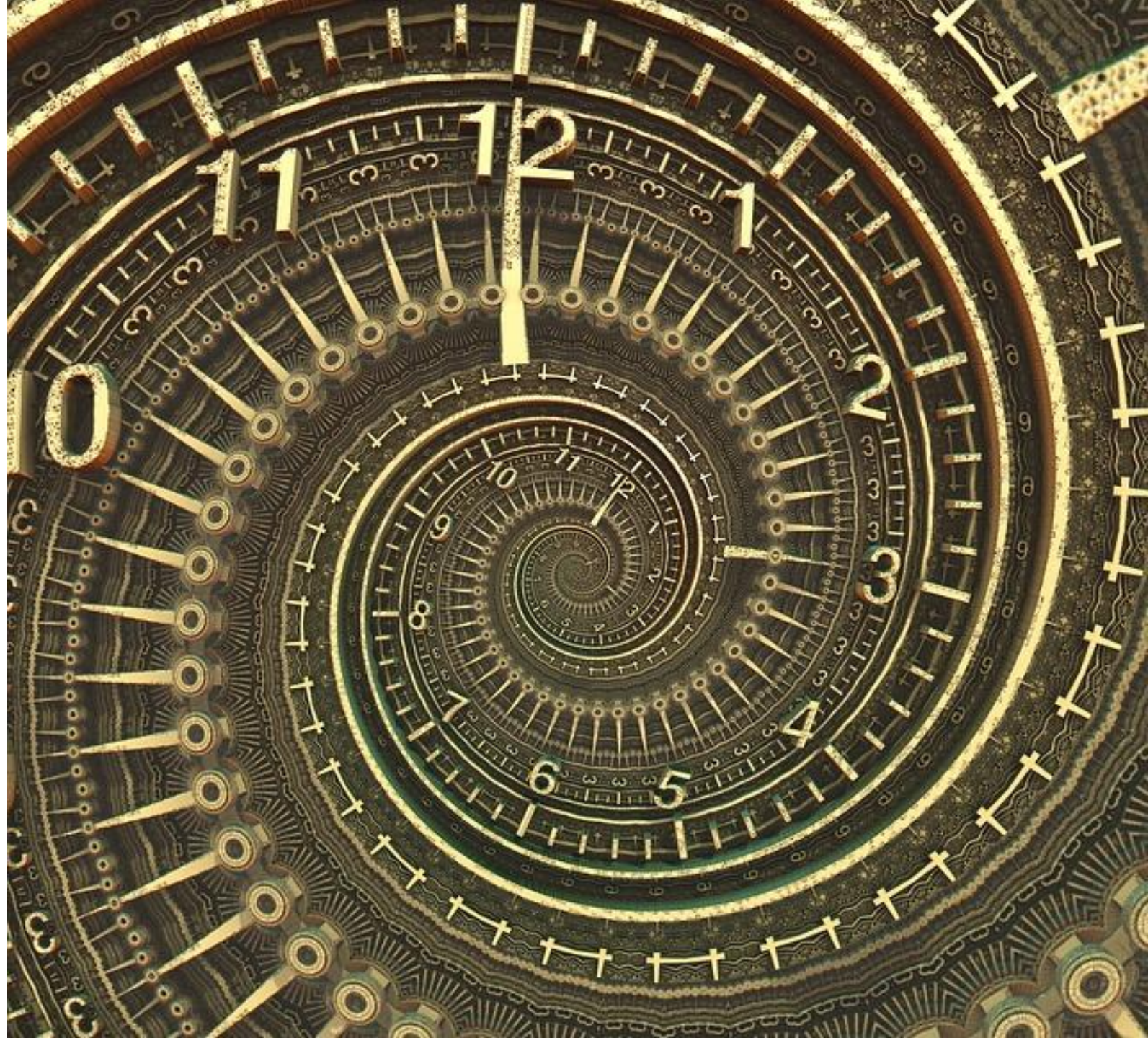
- Change bound data (@Input)
 - OnPush: Angular just compares the object reference!
 - e. g. `oldFlight === newFlight`
- Raise event within the component
- Notify a bound observable
 - `{{ flights$ | async }}`
- Trigger it manually
 - Don't do this at home ;-)
 - At least: Try to avoid this

Activate OnPush

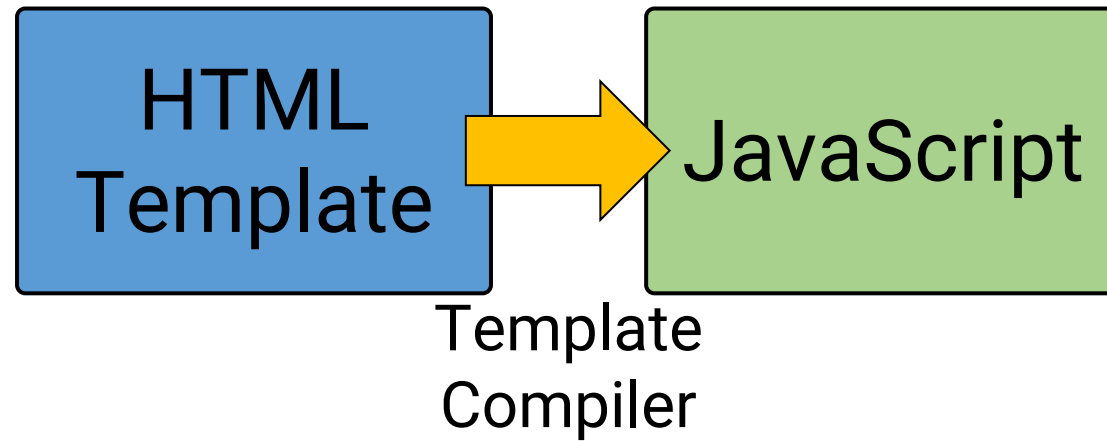
```
@Component({  
    [...]  
    changeDetection: ChangeDetectionStrategy.OnPush  
})  
export class FlightCard {  
    [...]  
    @Input() flight;  
}
```

DEMO

Ahead of Time (AOT) Compilation



Angular Compiler



Approaches

- JIT: Just in Time, at runtime
- AOT: Ahead of Time, during build
 - Since Angular 9: Default (when using Ivy)

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

DEMO

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

+ 1 modules

mergeMap.js

+ 1 modules

AsyncAction.js

+ 1 modules

ReplaySubject.js

+ 3 modules

Subscription.js

+ 1 modules

Subject.js

+ 1 modules

Observable.js

+ 1 modules

src

main.ts
+ 68
modules

polyfills.7c4efb87d4ba5dbbc58c.bundle.js

node_modules

zone.js

dist

zone.js

core-js

modules



FoamTree

LAB

Conclusion

Quick Wins

Lazy Loading
and
Preloading

OnPush w/
Immutables and
Observables

AOT and Tree
Shaking