



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Performance Tuning

angular-architects.io



Performance- Tuning with OnPush

DEMO

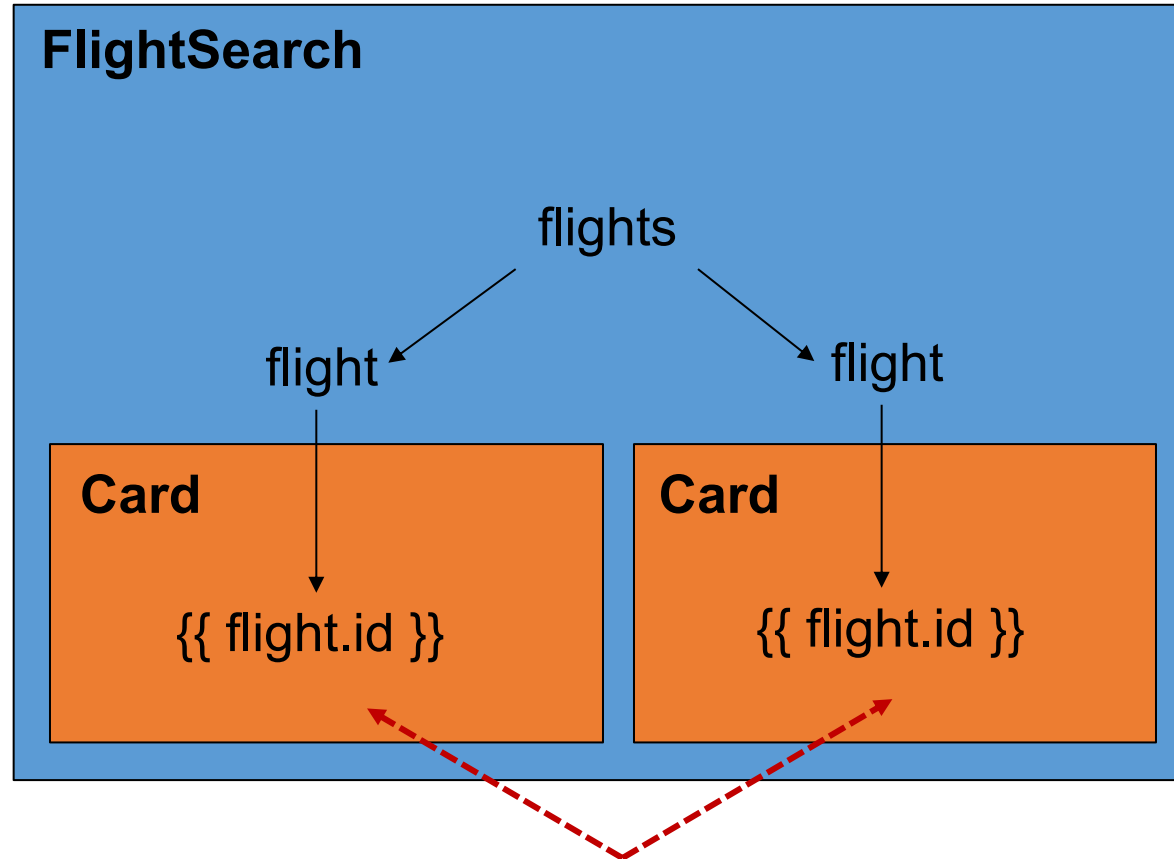


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

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;  
}
```



DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

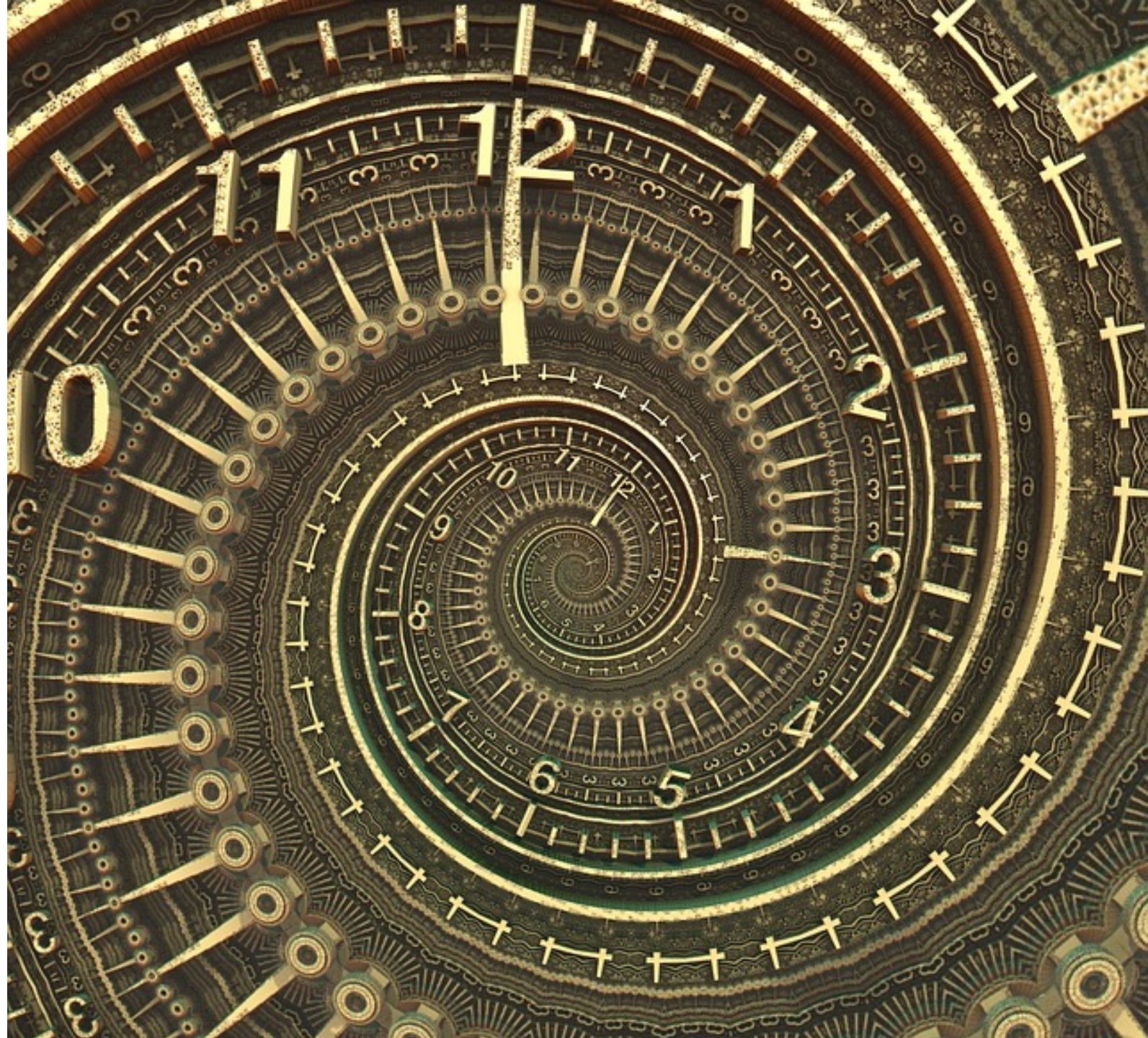


SOFTWARE
ARCHITECT

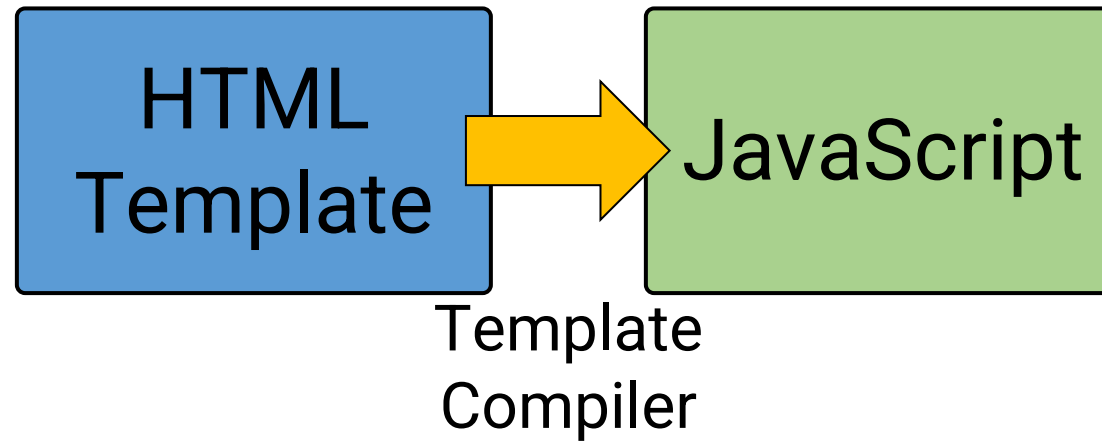
LAB



Ahead of Time (AOT) Compilation



Angular Compiler



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Approaches

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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

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

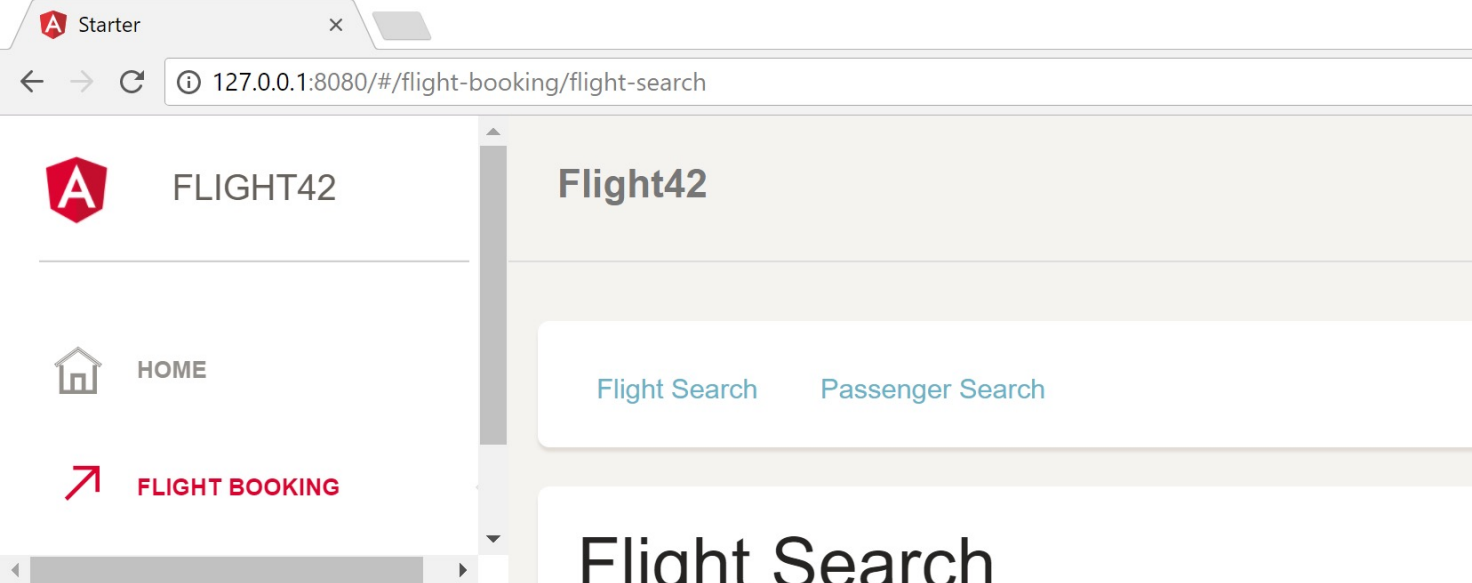
- `ng build --prod`
- `@ngtools/webpack` with `AngularCompilerPlugin`
- Can be used without CLI too



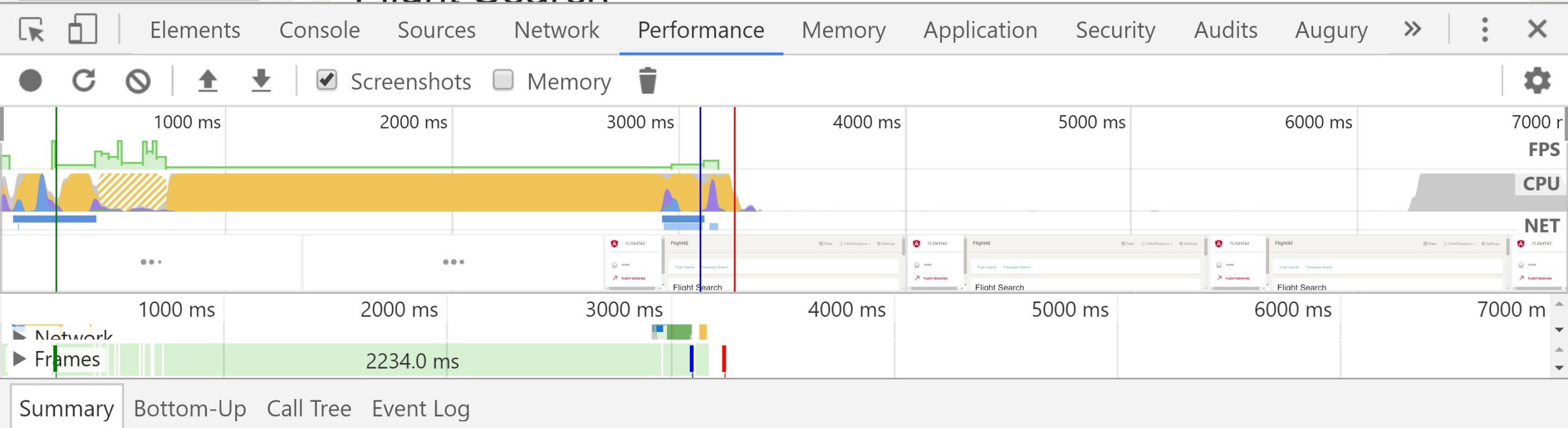
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

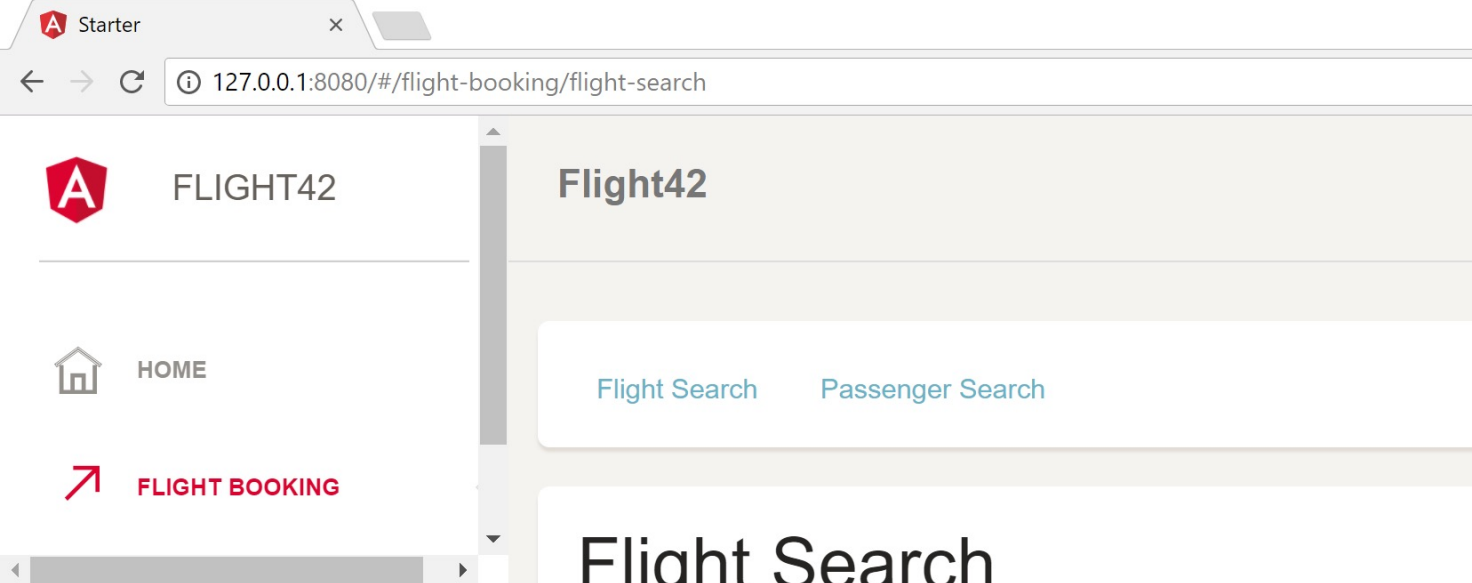


SOFTWARE
ARCHITECT

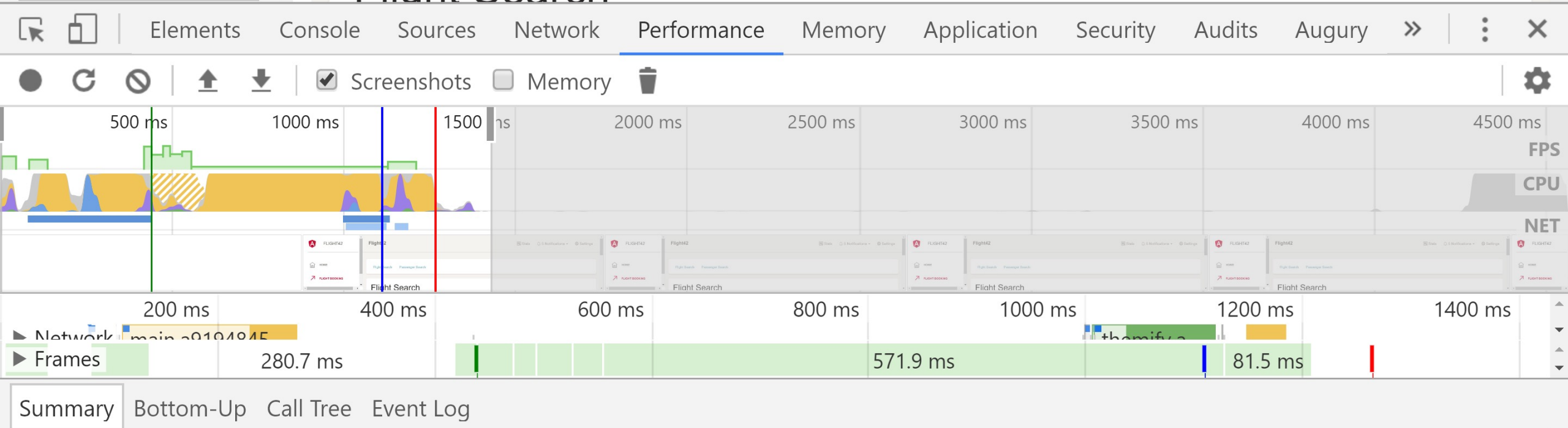


**Production Mode without AOT
(no lazy loading)**





**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



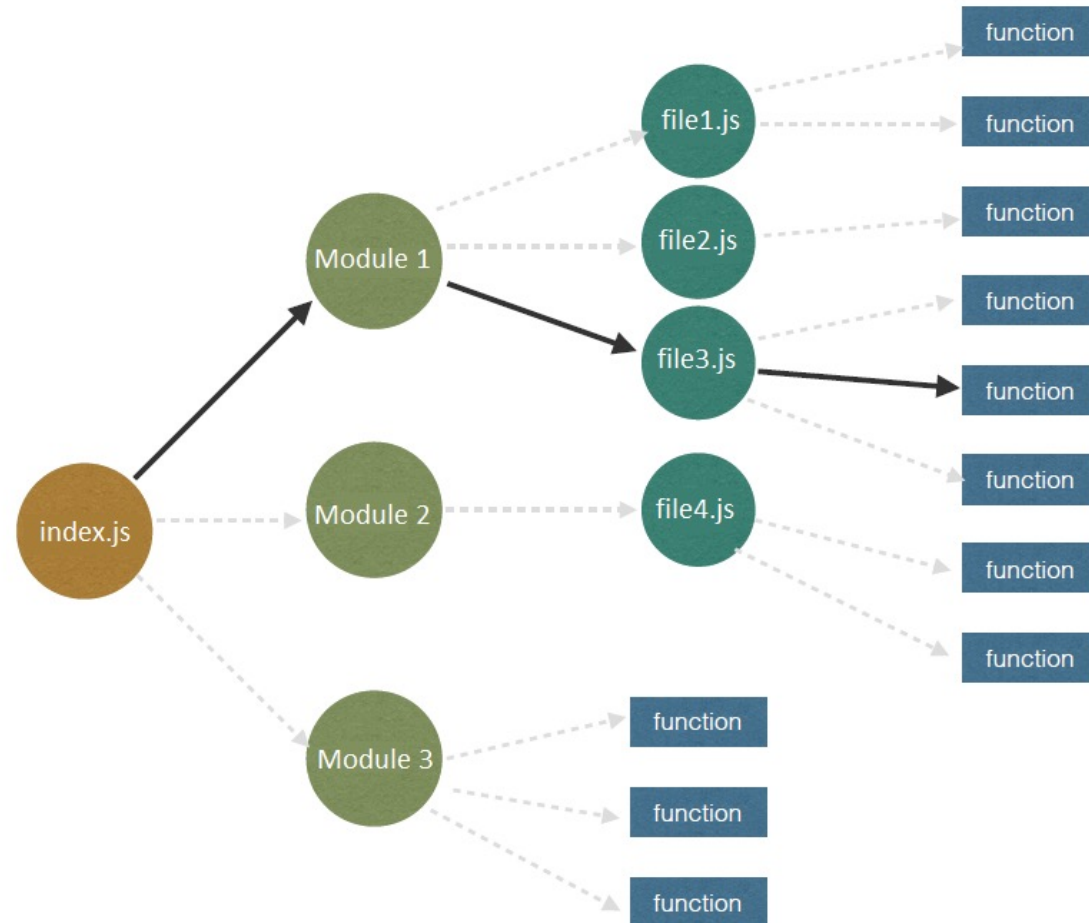
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

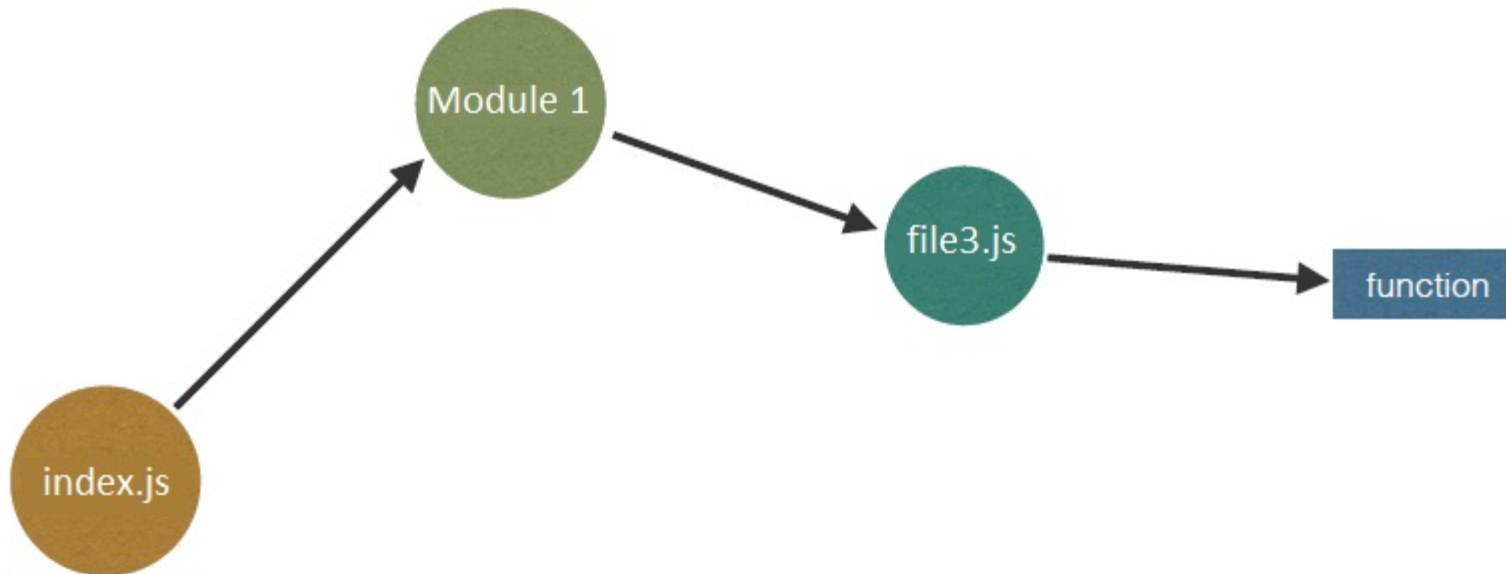
Tree Shaking

Before Tree Shaking



Tree Shaking

After Tree Shaking



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Webpack Bundle Analyzer



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
esm5router.js +
23 modulescommon
esm5

common.js http.js

forms
esm5forms.js +
2 modulesplatform-browser
esm5

platform-browser.js

http
esm5

http.js

rxjs

_esm5

add
delay.js + 2
modules
switchMap.js
+ 2 modules
fromEvent.js
+ 2 modules
merge.js
+ 2 modules
share.js
+ 4 modules
mergeMap.js
+ 2 modules
Subscriber.js

...

...

...

Subscription.js
+ 1 modulesReplaySubject.js
+ 3 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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Conclusion

Lazy Loading

Preloading

OnPush w/
Immutables and
Observables

AOT and Tree
Shaking



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

For performance deep dive

Watch this (starting at 8:30):

https://drive.google.com/file/d/15fmyedJPYSOlV_0YvFtg26XGS8tZpZ03/view

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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT