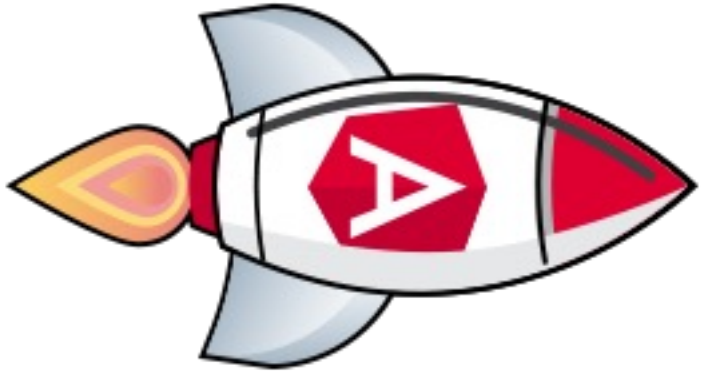
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Runtime Performance Change Detection

Alex Thalhammer

Outline 03 - Runtime Performance



- Change Detection
- Further Runtime

Outline

- Out of bound change detection
- Zone pollution by 3rd party libs
- Optimization with state or flags
- Optimization with Angular Pipes

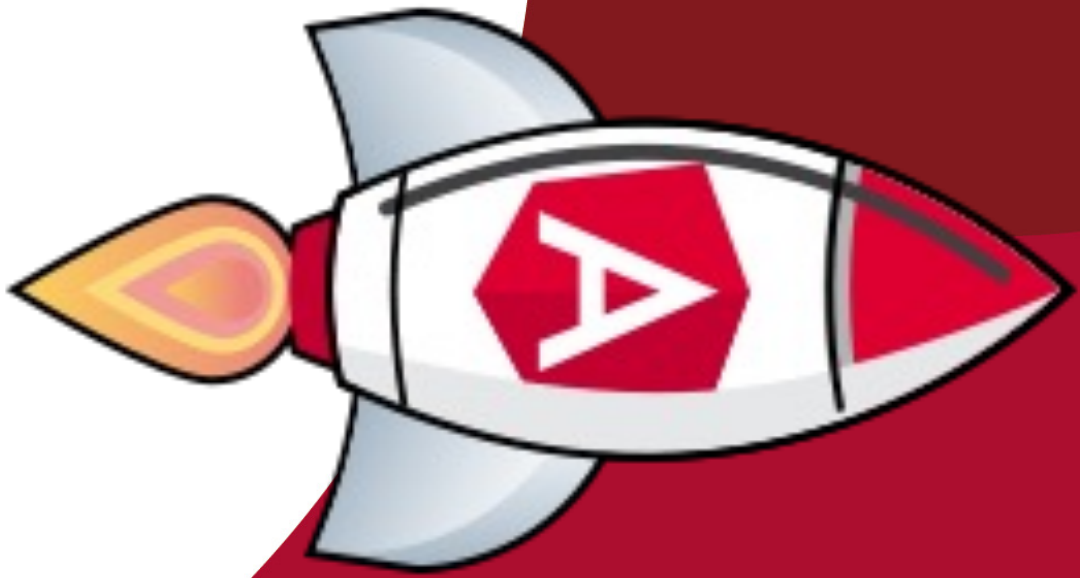
Out of bound change detection

- Problem: *Local state change triggers change detection in other comps*
  E.g. Input field keydown triggers CD in other components

- Identify:
  - Use the infamous blink() or
  - Angular DevTools Profiler
  - console.log() in lifecycle hook (e.g. ngDoCheck)

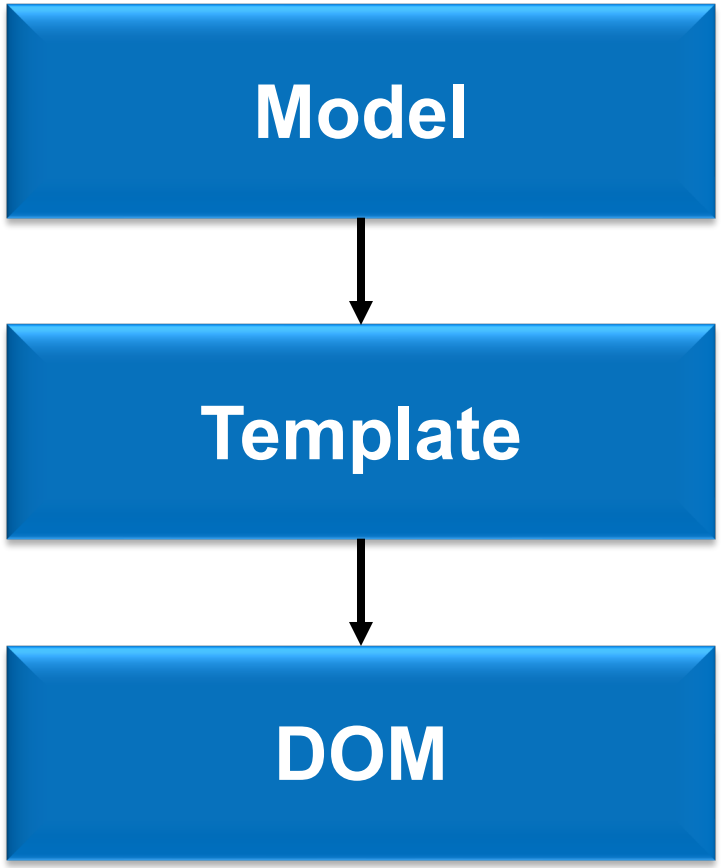
- Solution: ChangeDetectionStrategy.OnPush as default

Change Detection in Angular



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

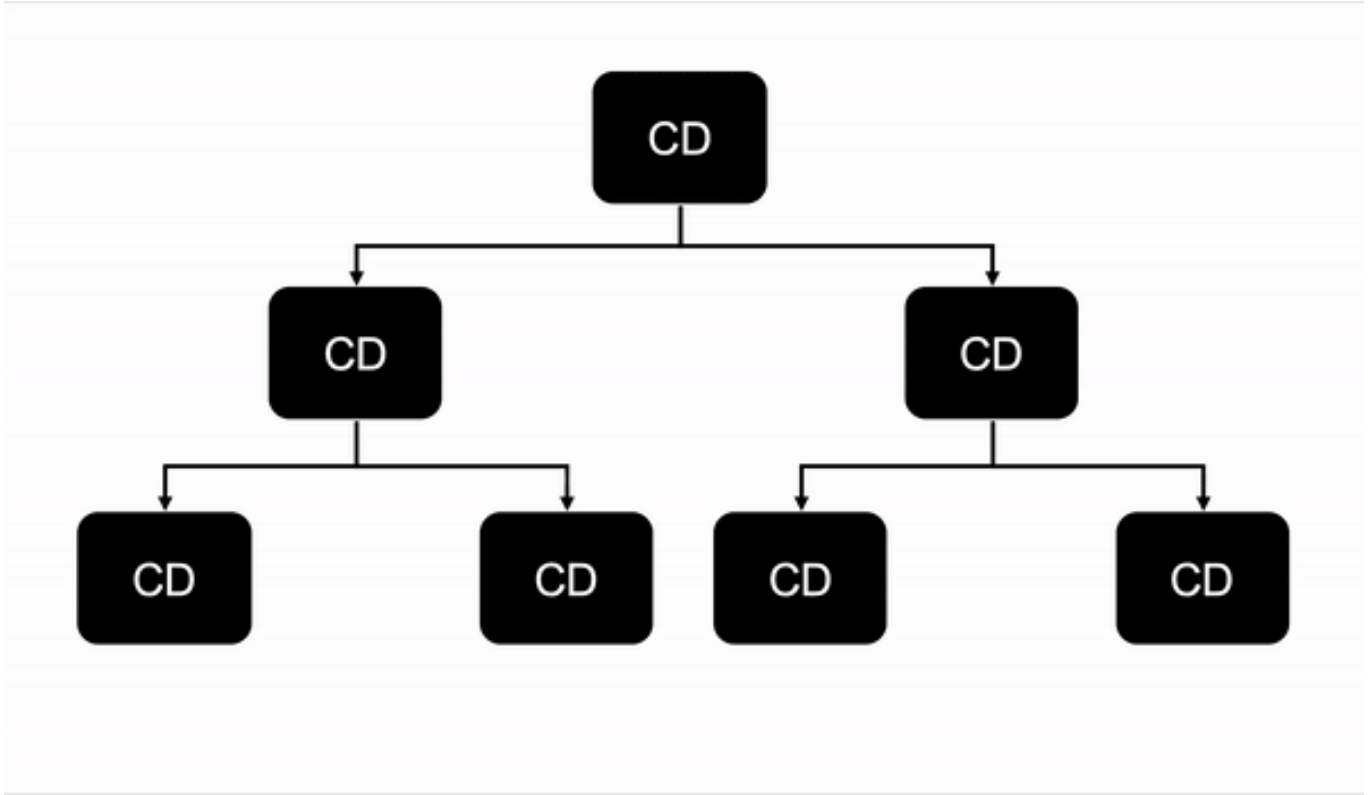
DOM Rendering



Change Detection

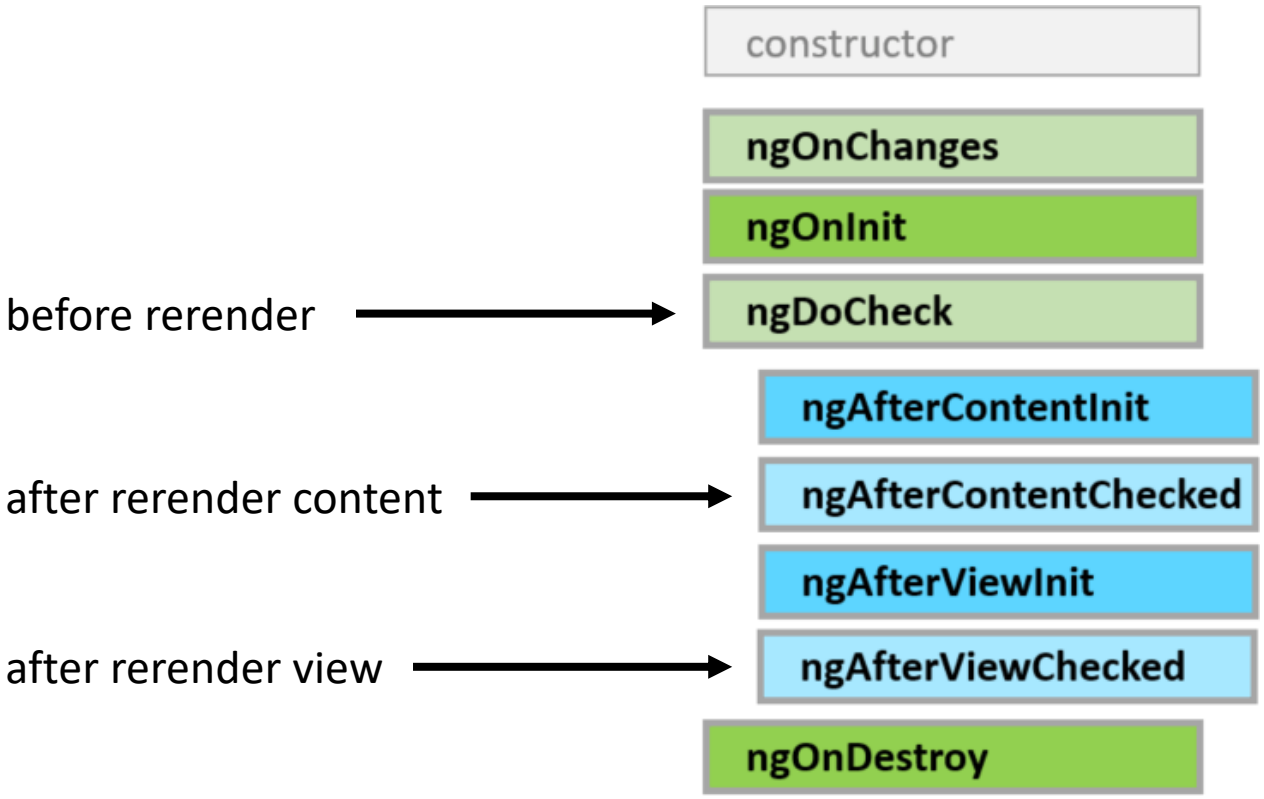
- 1.) User or App changes the model (e.g. input, blur or click)
- 2.) NG CD checks for **every component** (from root to leaves) if the corresponding component model has changes and thus its view (DOM) needs to be updated
- 3.) If yes then update / rerender the component's view (DOM)

Change Detection – From Root To Leaves



Img src: https://mokkapps.de/blog/the-last-guide-for-angular-change-detection-you-will-ever-need/

Change Detection – Rerender Components



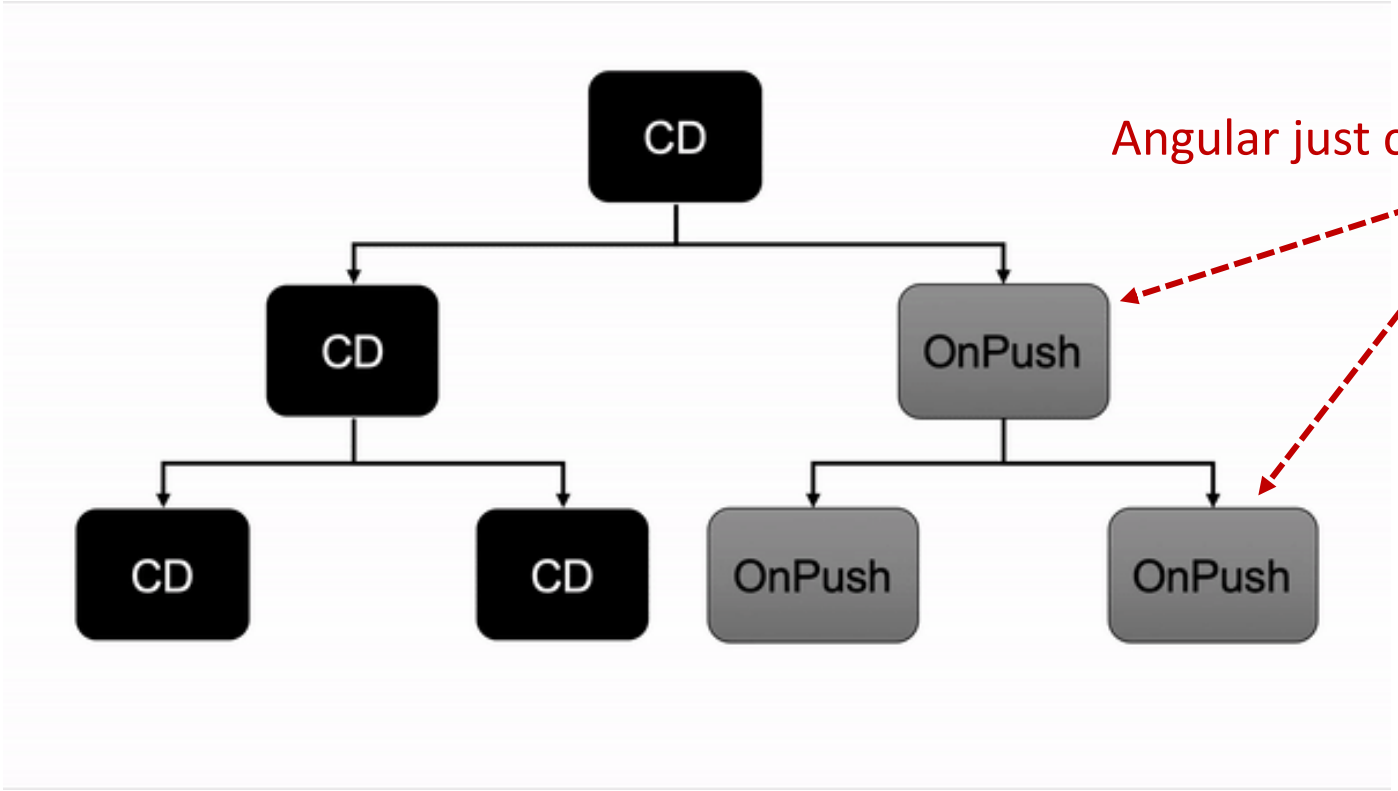
DEMO – ChangeDetection

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Performance-Tuning with OnPush

Change Detection – OnPush Strategy



Img src: https://mokkapps.de/blog/the-last-guide-for-angular-change-detection-you-will-ever-need/

Activate OnPush Strategy

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

"Notify" about change?

- 1 Change bound data (**@Input**)
  - OnPush: Angular just compares the object reference!
  - e. g. oldFlight !== newFlight (BTW: like ngOnChanges)
- 2 Raise event within the component and its children (e.g. @Output)
- 3 Emit in a bound observable into the async pipe | or update a signal
  - {{ **flights$ | async** }} | {{ **flightsSignal()** }}
- 4 Do it manually (cdr.markForCheck())
  - Don't do this at home ;-)
  - But there are reasonable cases (where we can neither use 1 nor 3)

CDR - markForCheck() vs detectChanges()

- Use CDR.**markForCheck() to notify the next CD cycle** if using **OnPush**
  - Useful when you're bypassing the ChangeDetectionStrategy.OnPush e.g. by mutating some data or you've just updated the components model

- Use CDR.**detectChanges() to trigger CD immediately** for this view and it's children respecting the its/their CD strategy
  - Useful when you've updated the model after angular has run it's change detection, or if the update hasn't been in Angular world at all
  - For the whole app (from root) you can do ApplicationRef.tick()

DEMO – OnPush

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Set OnPush as default

- Add to angular.json schematic / project.json generator
- Add a lint rule

Zone pollution by 3rd party libs (charts)

Zone pollution by 3rd party libs (charts)

- Problem: *Callbacks that trigger redundant change detection cycles*

- Identify: Use the infamous blink() or the Angular DevTools Profiler
  - E.g. MouseEvent listeners
  - requestAnimationFrame() or
  - setTimeout()

- Solution: Run outside of NG Zone
  - Inject (private ngZone: NgZone)
  - Call this.ngZone.runOutsideAngular(() => doStuff)
  - https://angular.io/guide/change-detection-zone-pollution

- Alternative: Using cdr.detach() for components

ChangeDetectorRef API, once more

detectChanges	• Runs Change Detector for the component and its children • It runs CD once also for the component which is detached from the component tree
markForCheck	• It marks component and all parents up to root as dirty • In next cycle Angular runs CD for marked components
reattach	• Re-attaches the component in the change detection tree • If parent component's CD is detached, it won't help, so make sure to run markForCheck with reattach
detach	• Detaches the component from the change detection tree • Bindings will also not work for the component with detached CD
checkNoChanges	• Changes the component and its children and throws error if change detected

Img src: https://www.telerik.com/blogs/simplifying-angular-change-detection/

DEMO – Zone Pollution

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Lab

Runtime Performance – Change Detection

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Optimization with state or flags

- Problem: *Redundant calculations for conditions*
- Identify: Methods being executed in ***ngIf** statements
- Solution: Use StateManagement like Subjects or use boolean flags or strings, that only change when they should

Optimization with Angular Pipes

- Problem: *Redundant calculations / transforming / formatting*
- Identify: Methods being executed in string interpolations in the template or similar things slowing change detection cycles
- Solution: Use (pure) Angular Pipes

Recap

- **Out of bound change detection**
- **Zone pollution by 3rd party libs**
- Optimization with state or flags
- **Optimization with Angular Pipes**

References

- Minko Gechev (@mgechev) for Angular on YouTube
  - https://www.youtube.com/watch?v=FjyX_hkscII
  - https://www.youtube.com/watch?v=f8sA-i6gkGQ
  - New in NG 17, 18+: https://www.youtube.com/watch?v=2M17gRQbgfI
- Resolving Zone Pollution
  - https://angular.io/guide/change-detection-zone-pollution
- Angular Performance Optimization using Pure Pipe
  - https://www.youtube.com/watch?v=YsOf90RZfss