

The background is a solid teal color with a complex, layered pattern. It features various architectural elements like brick walls, a spiral staircase, and a large circular structure resembling a tunnel or a well. There are also some abstract shapes and a small crescent moon in the upper right corner.

# Reactive Angular met RxJS

## Change Detection

Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

# What is Change Detection

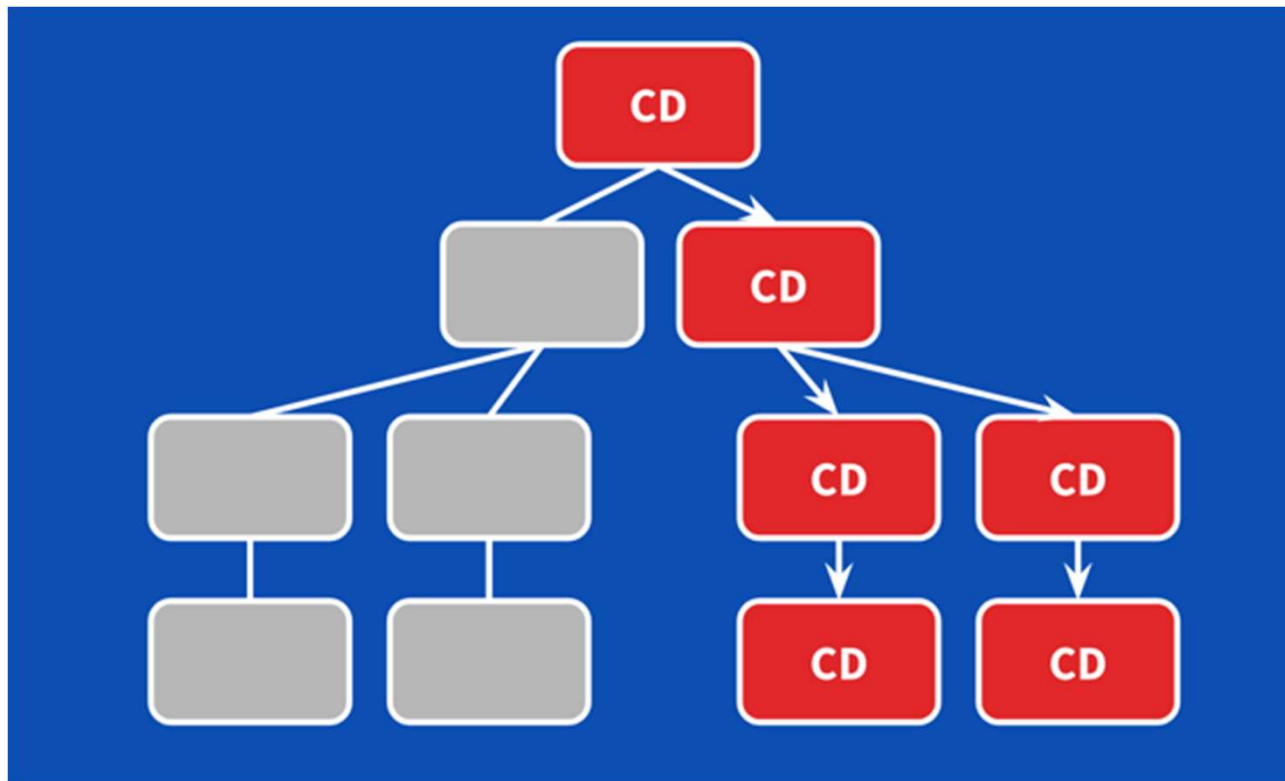
- Angular performs *Change Detection* every time something changes:
  - In the View (mousedown, mousemove, etc)
  - In the controller (data change, network request, subscription, etc)
- The view gets updated with changes in the model,
- The model gets updated with changes in the view
  - Changes are propagated to child components!

## **Where's the problem?**

*CD can be costly in bigger apps  
with lots of nested components!*

## Extra – about Change detection

- Angular uses zone.js to perform change Detection
- Angular has some strategies to optimize CD

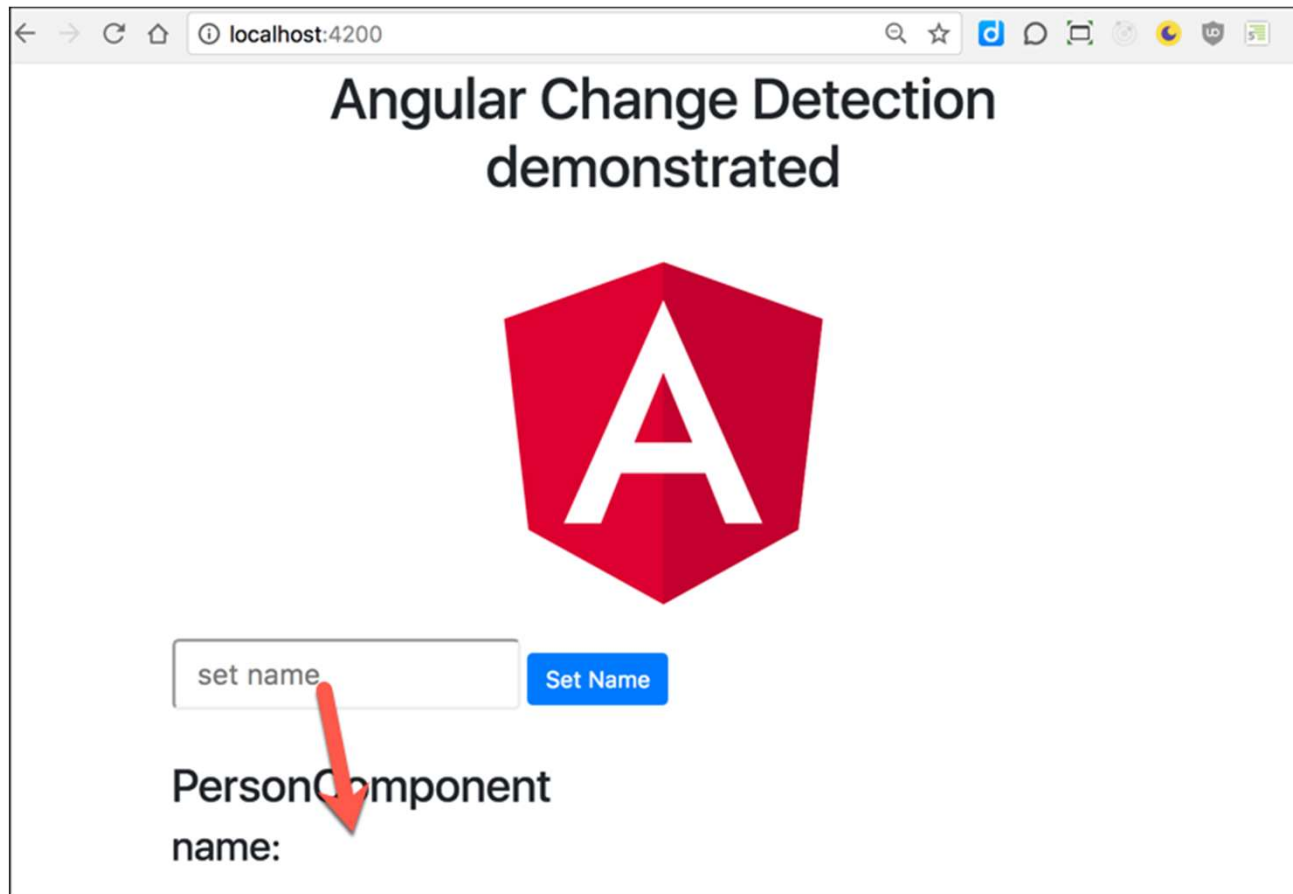


# Using CD

- `changeDetection: ChangeDetectionStrategy.OnPush` in component decorator

# Example

- Example ../800-change-detection



# Demo

- General rule: nested components are inside *one* view.
  - If the parent runs a change detection cycle, changes are propagated to the children.
  - Unless they have `.onPush()` activated
- .../800-change-detection
- (un) comment various lines

# Alligator.io



## *Understanding Change Detection Strategy in Angular*

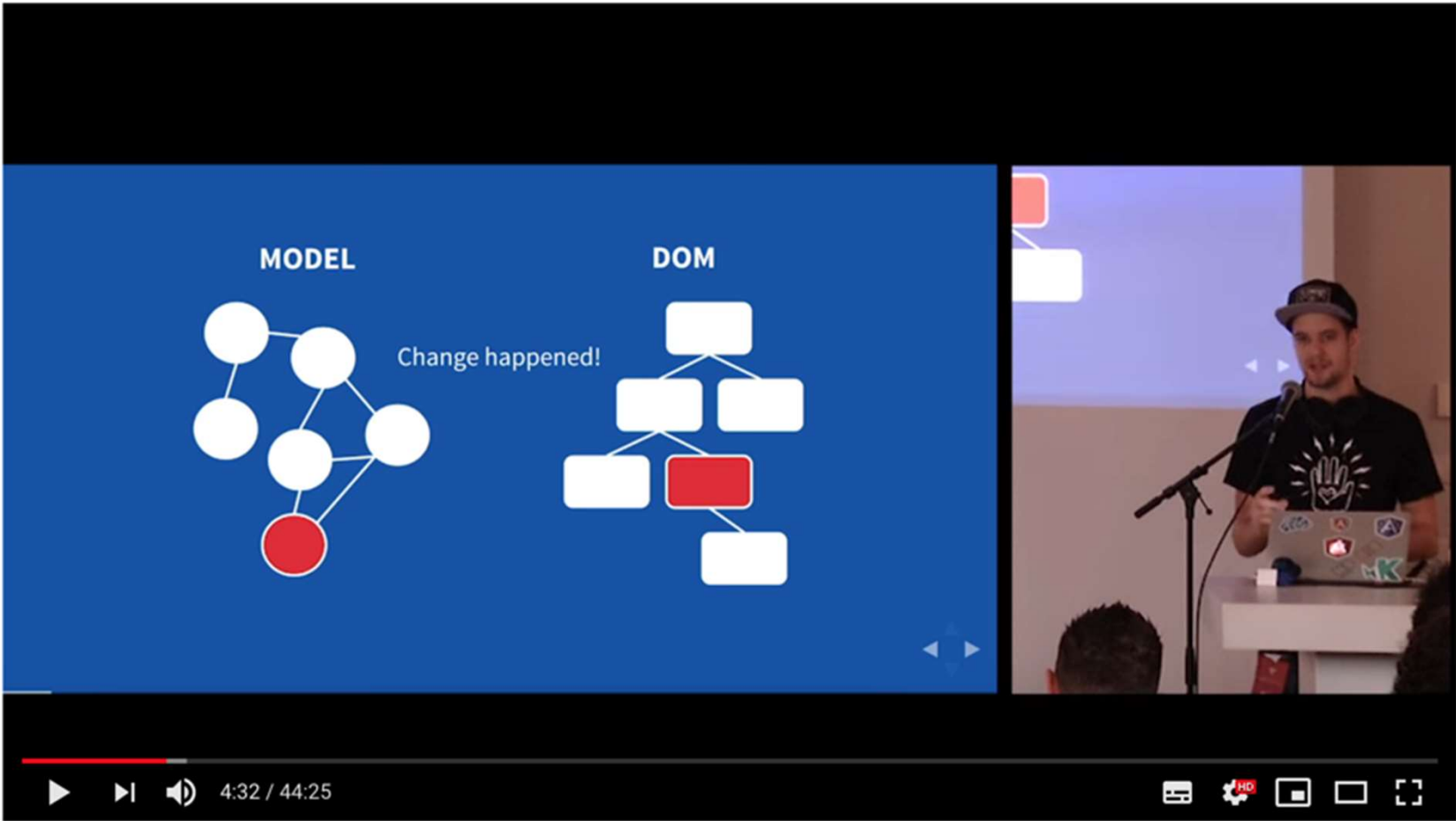


Angular performs change detection on all components (from top to bottom) every time something changes in your app from something like a user event or data received from a network request. Change detection is very performant, but as an app gets more complex and the amount of components grows, change detection will have to perform more and more work. There's a way to circumvent that however and set the change detection strategy to `OnPush` on specific components. Doing this will instruct Angular to run change

<https://alligator.io/angular/change-detection-strategy/>



# More Info



The video player displays a presentation slide with a blue background. On the left, under the heading "MODEL", is a graph of five white circles connected by lines, with one circle at the bottom highlighted in red. On the right, under the heading "DOM", is a tree structure of white rectangles, with one rectangle in the middle highlighted in red. The text "Change happened!" is positioned between the two diagrams. To the right of the slide is a small inset video of the presenter, Pascal Precht, wearing a black t-shirt and a cap, standing at a podium with a microphone and a laptop. The video player's control bar at the bottom shows a progress bar at 4:32 / 44:25, along with play, pause, volume, and other standard controls.

NG-NL 2016: Pascal Precht - Angular 2 Change Detection Explained

<https://www.youtube.com/watch?v=CUxD91DWkGM>

# More on Change Detection



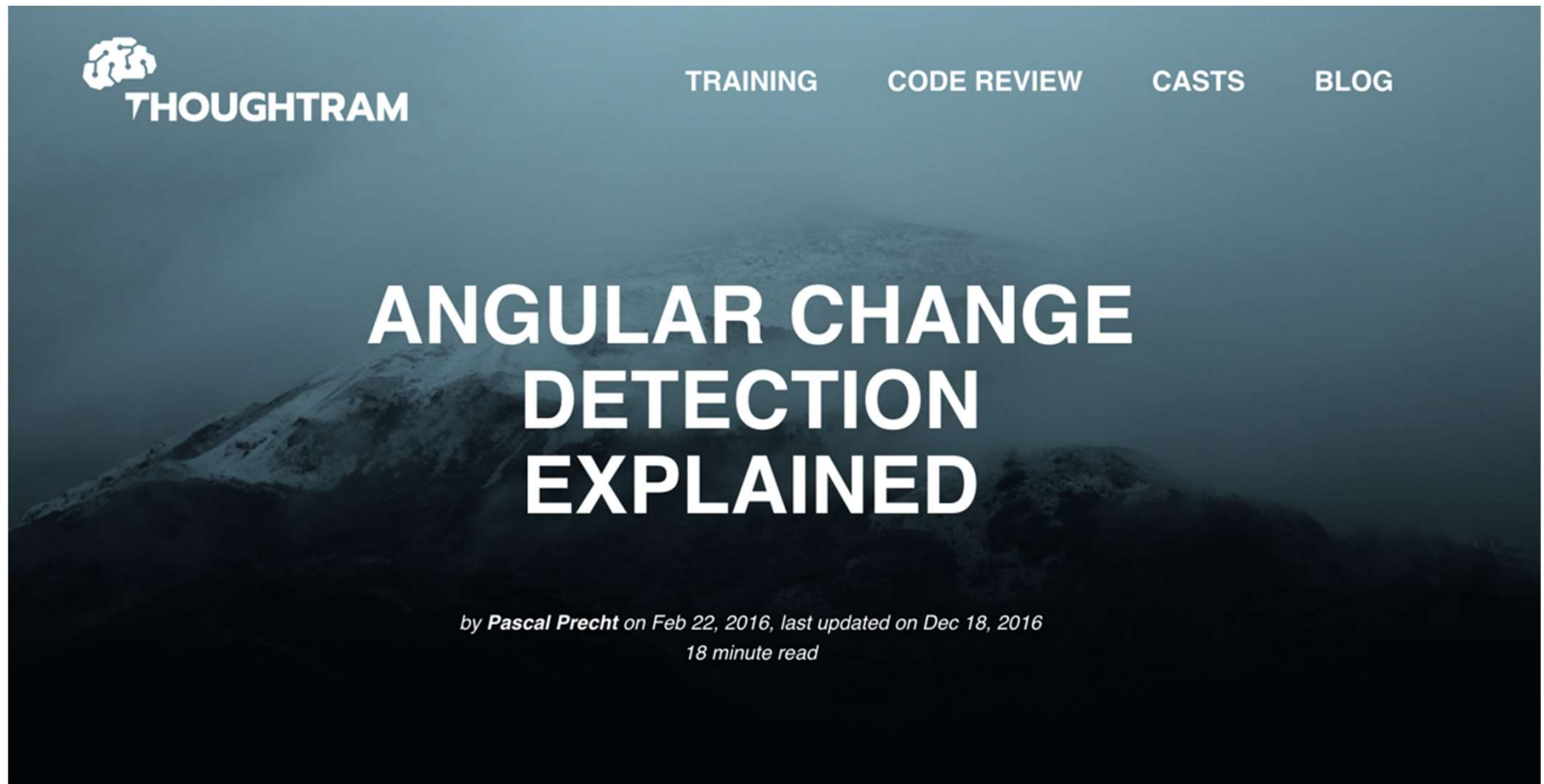
<https://blog.angularindepth.com/a-gentle-introduction-into-change-detection-in-angular-33f9ffff6f10>

# Change Detection – deep dive



<https://blog.angularindepth.com/everything-you-need-to-know-about-change-detection-in-angular-8006c51d206f>

# Thoughtram



<https://blog.thoughtram.io/angular/2016/02/22/angular-2-change-detection-explained.html>