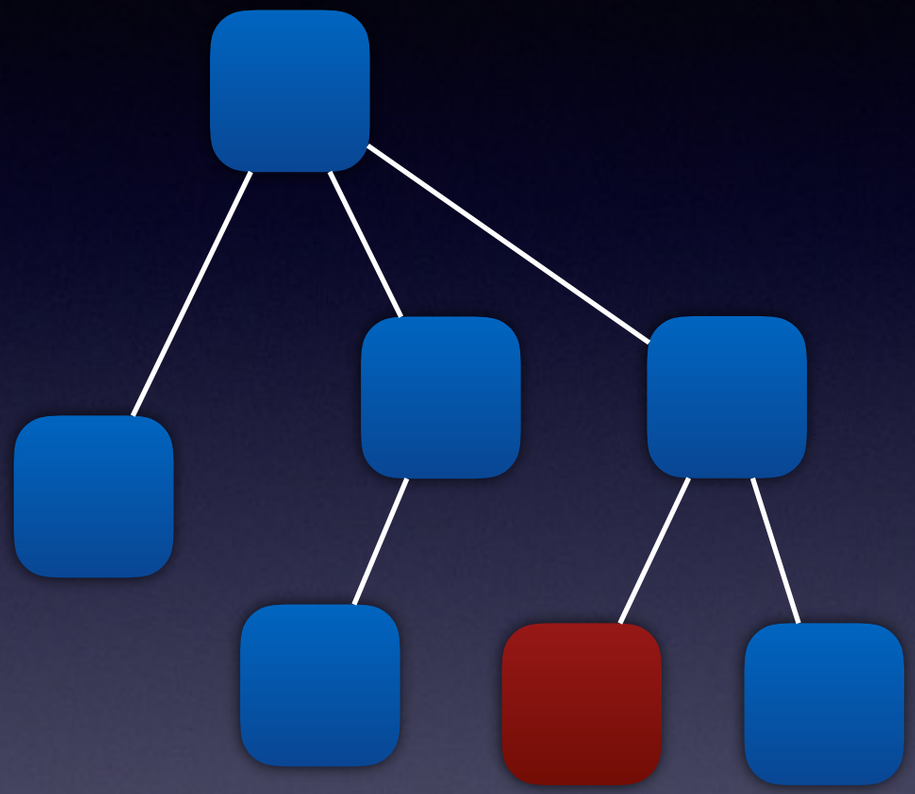
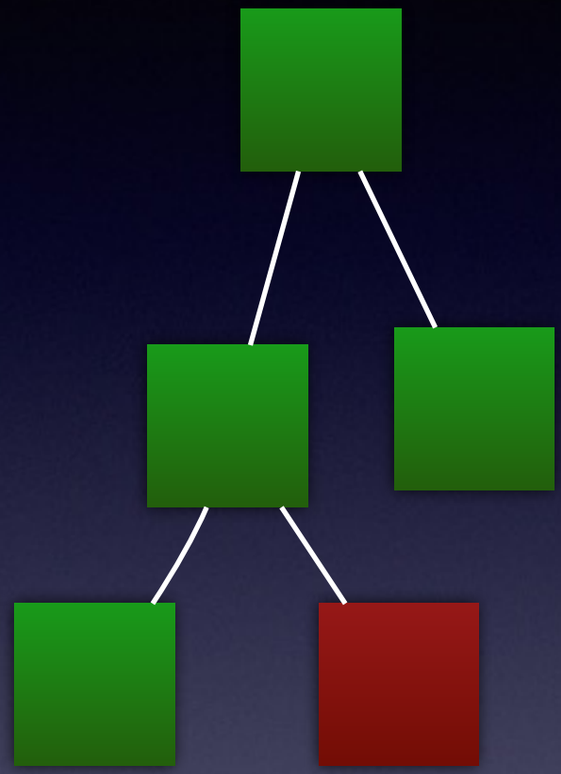
Change Detection in Angular 2

JS Model



DOM



Plan:

1. CD in Ng1.
2. Differences for Ng2.
3. Experiments with CD in a demo application.

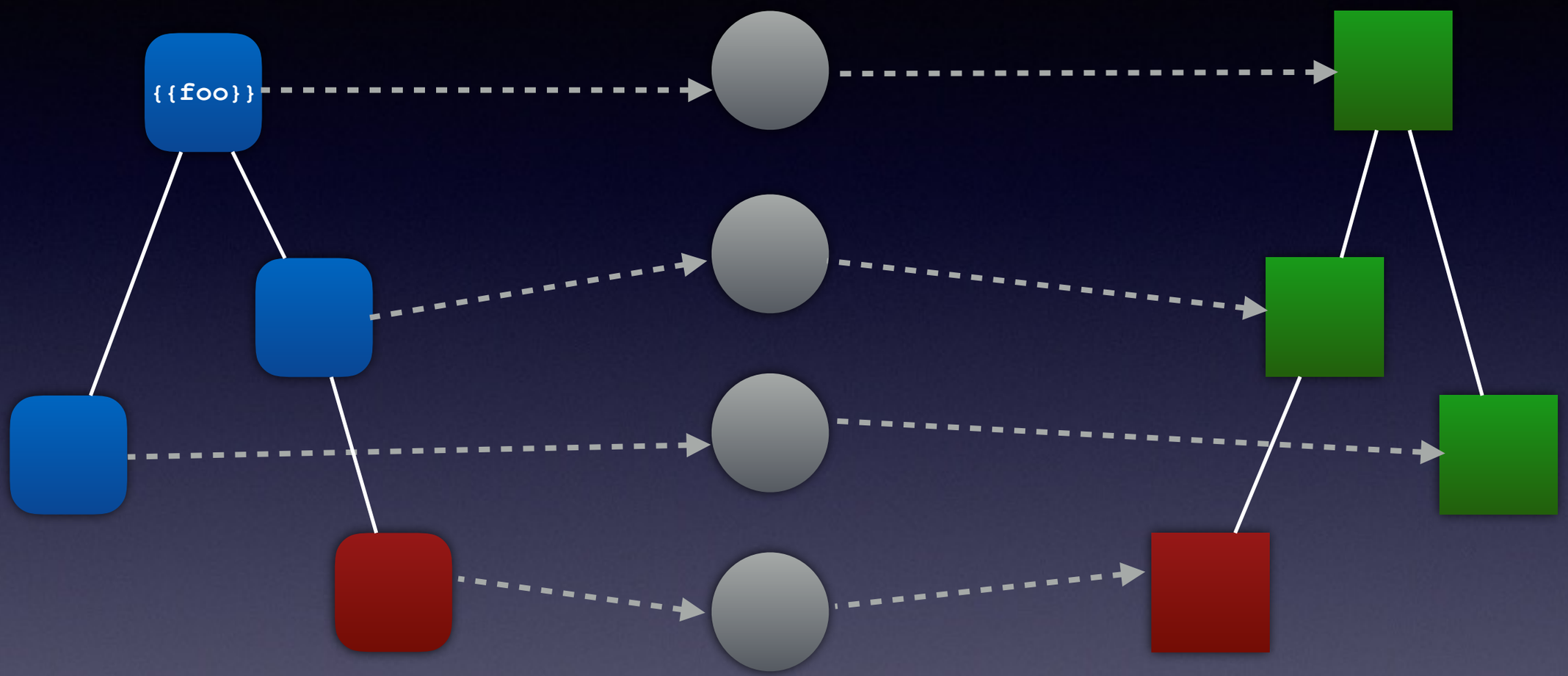
Ng: Angular
CD: Change Detection

JS Model

Watchers

DOM

$$wathers



- application phase: change model

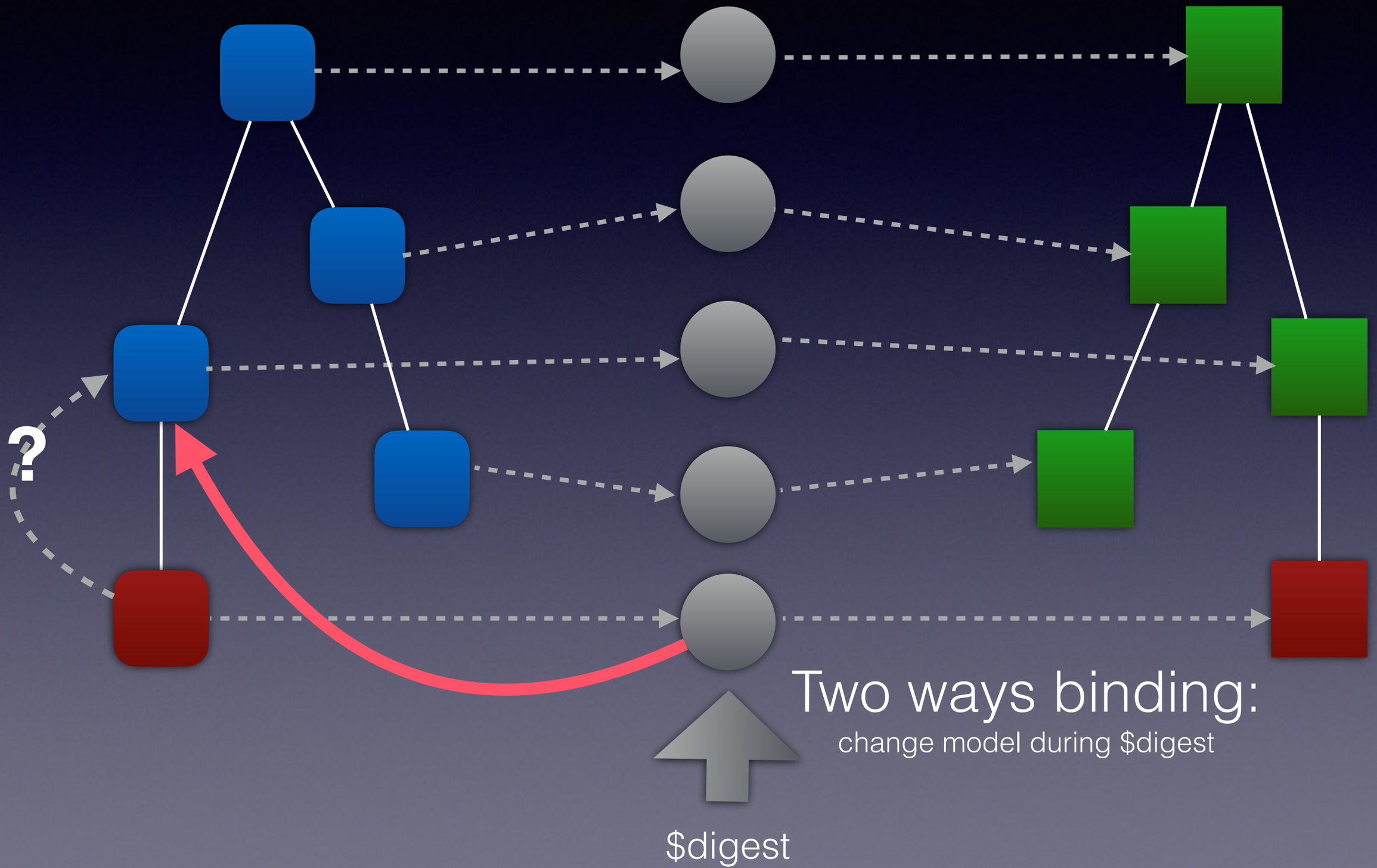
- digest phase: change view

$digest

JS Model

Watchers

DOM



Two ways binding:

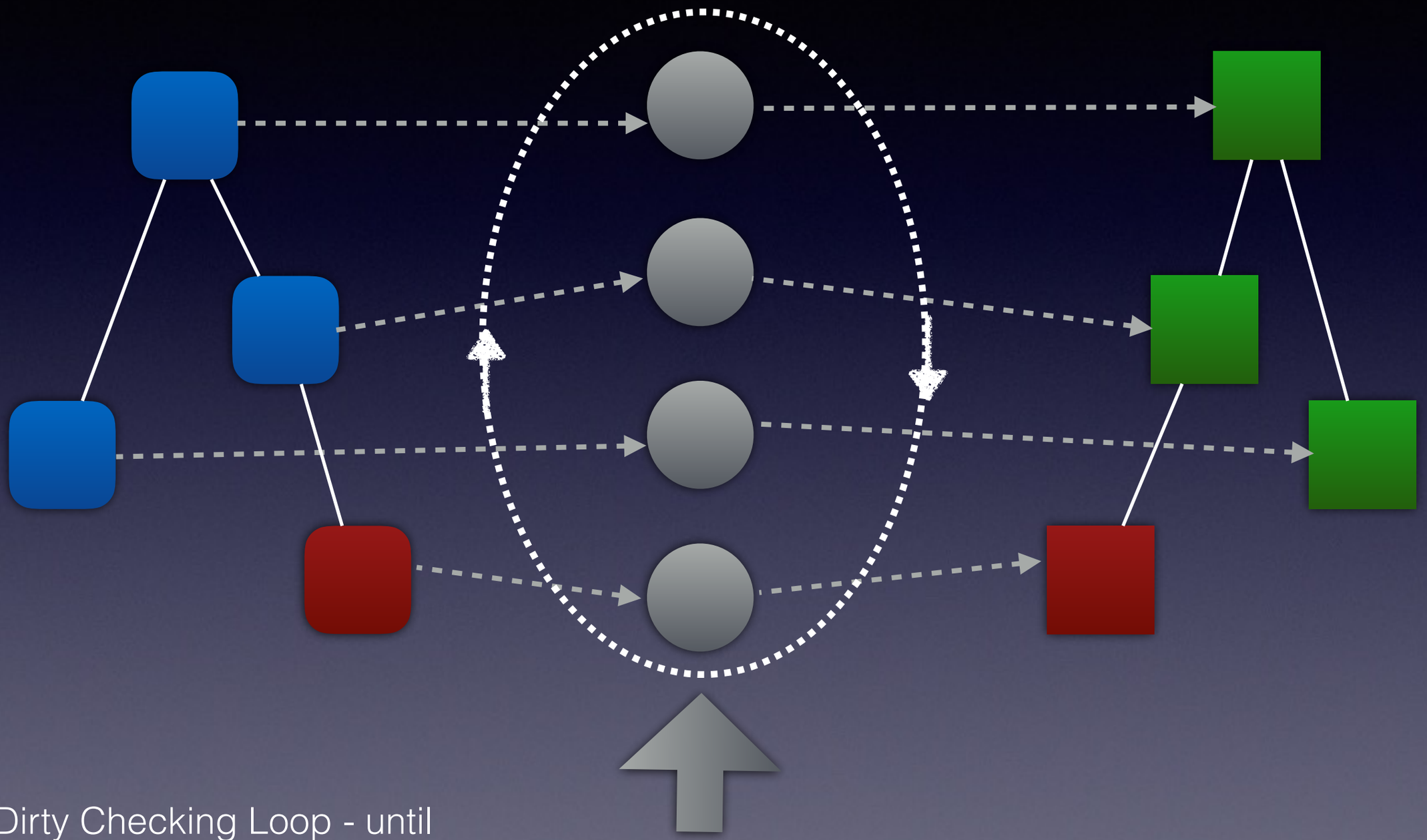
change model during $digest

$digest

JS Model

Watchers

DOM



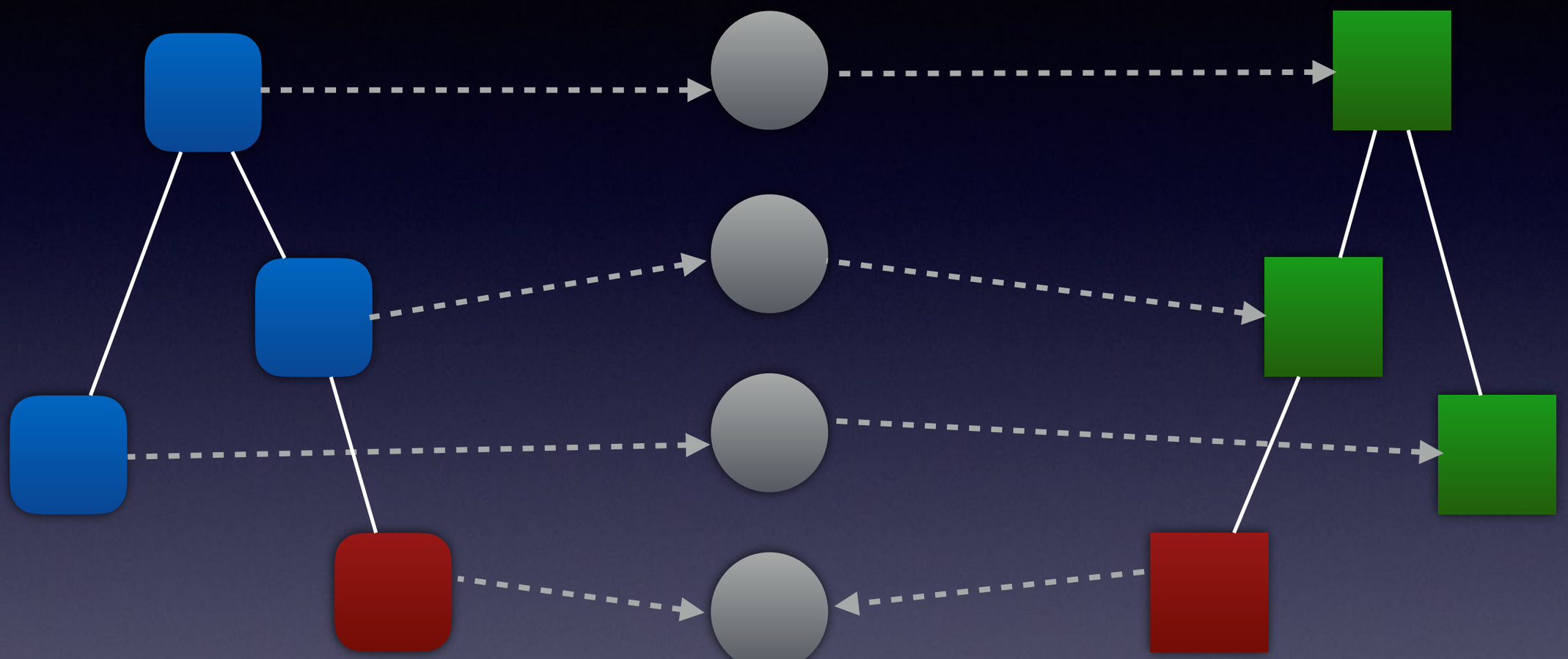
1. Dirty Checking Loop - until model stabilises

$digest

JS Model

Watchers

DOM

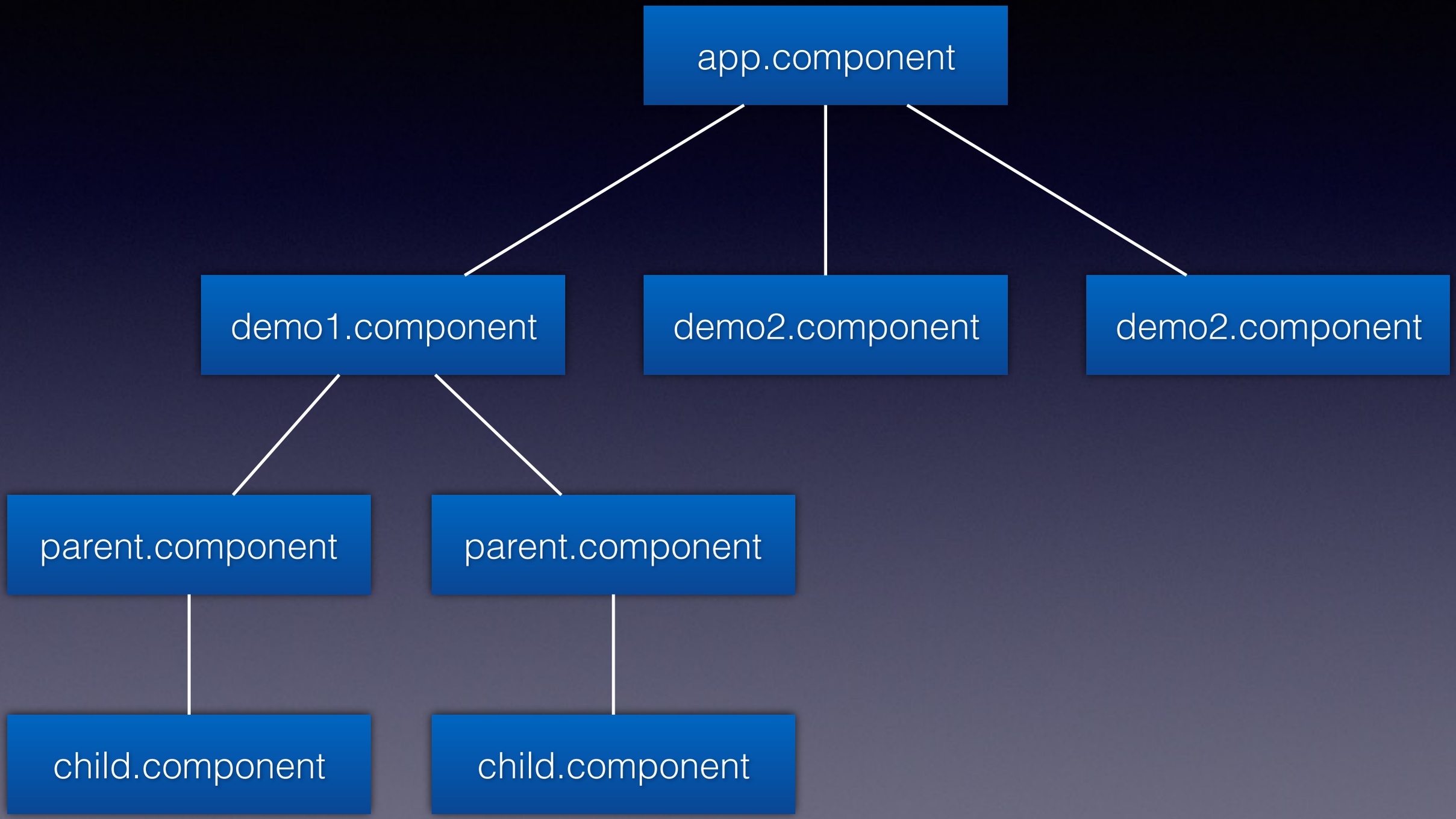


2. You should be responsible for kicking off $digest:
- $apply
- special ng directives (ng-click)
- special ng services (Timer, Http)

$digest

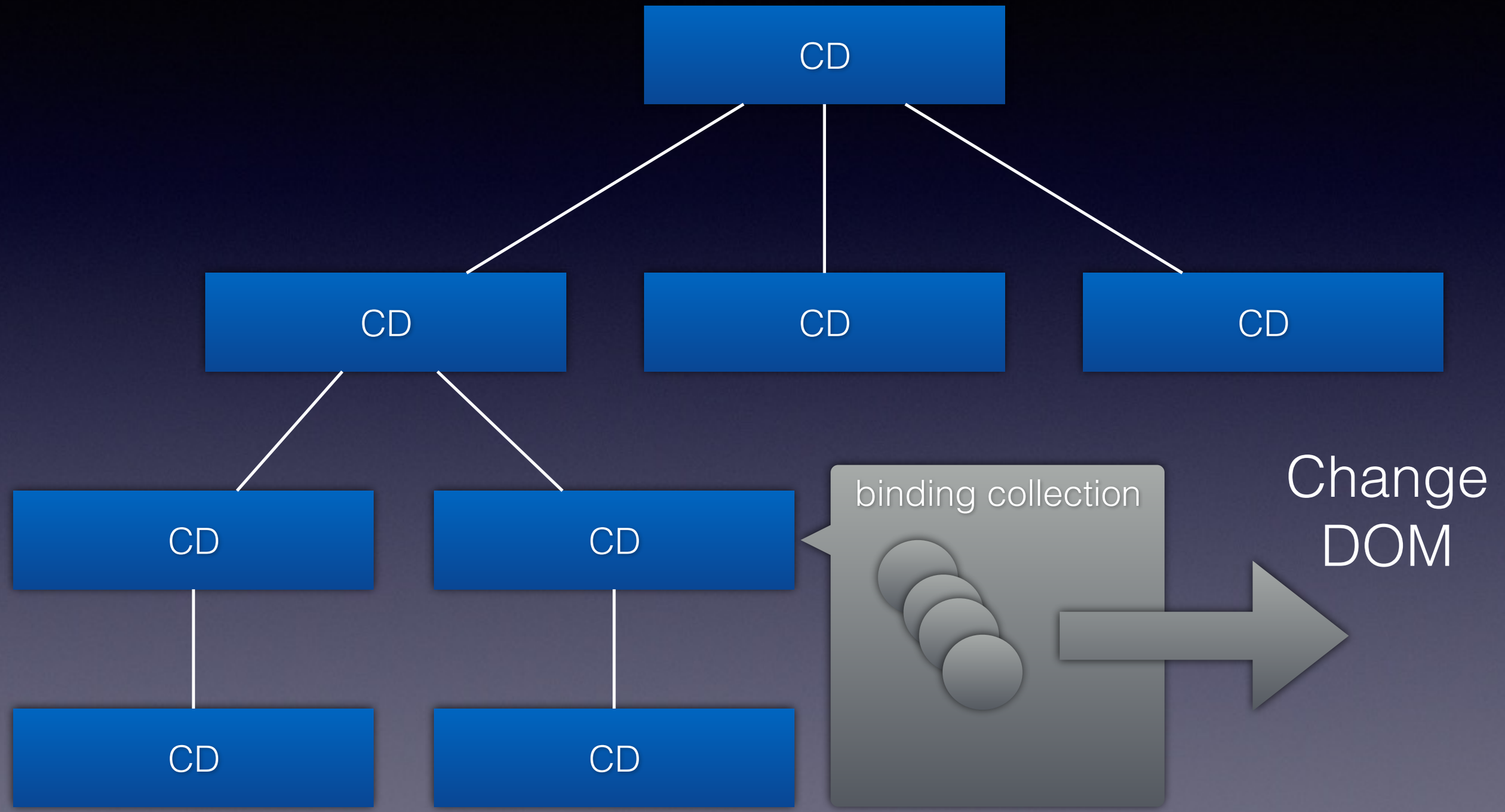
3. Poor CD customisation (only dirty checks limit: 10 as default)

Ng2: Only one-way binding:
One pass through the component tree when detecting changes



Ng2: Only one-way binding:

One pass through the component tree when detecting changes



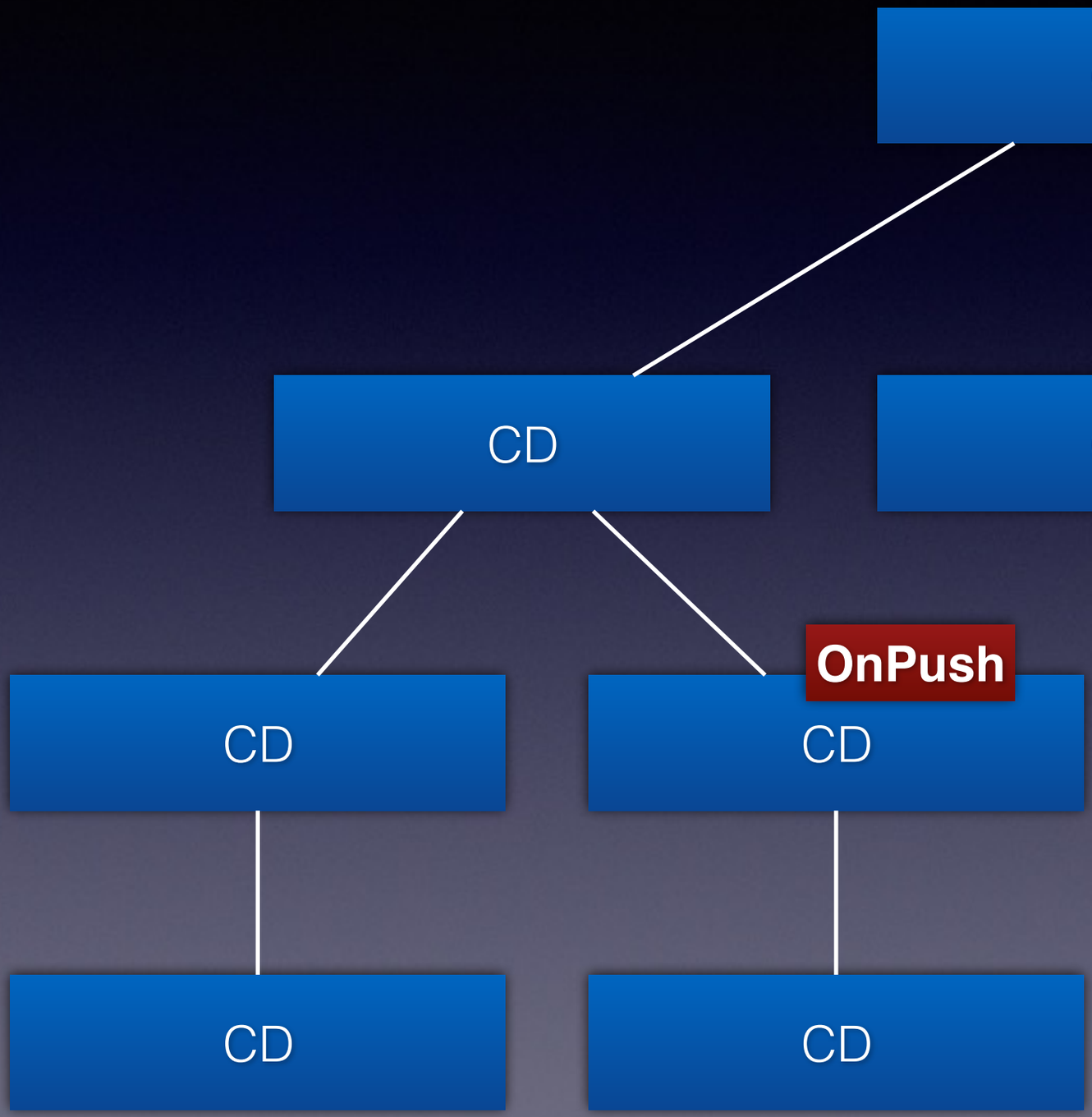
Ng2: You are responsible for kicking off CD **no more**.

NgZones: monkey patch any global asynchronous operations by the browser

CD is run by Angular 2 automatically

- no $scope.$apply();
- no ng-click;
- no $timeout

Ng2: CD customisation



CD skips a component's subtree when:

- it's set to **OnPush**
- and **non of its input property change**
- and **no event is fired inside that component**.

Summary

- CD performs automatically
- Only one pass through the component tree
- CD allows customisation
- CD can be easily traced. Represented in application life cycle.
- When being optimised (OnPush): take care of immutability, promises (observables).
- Do not neglect development mode!