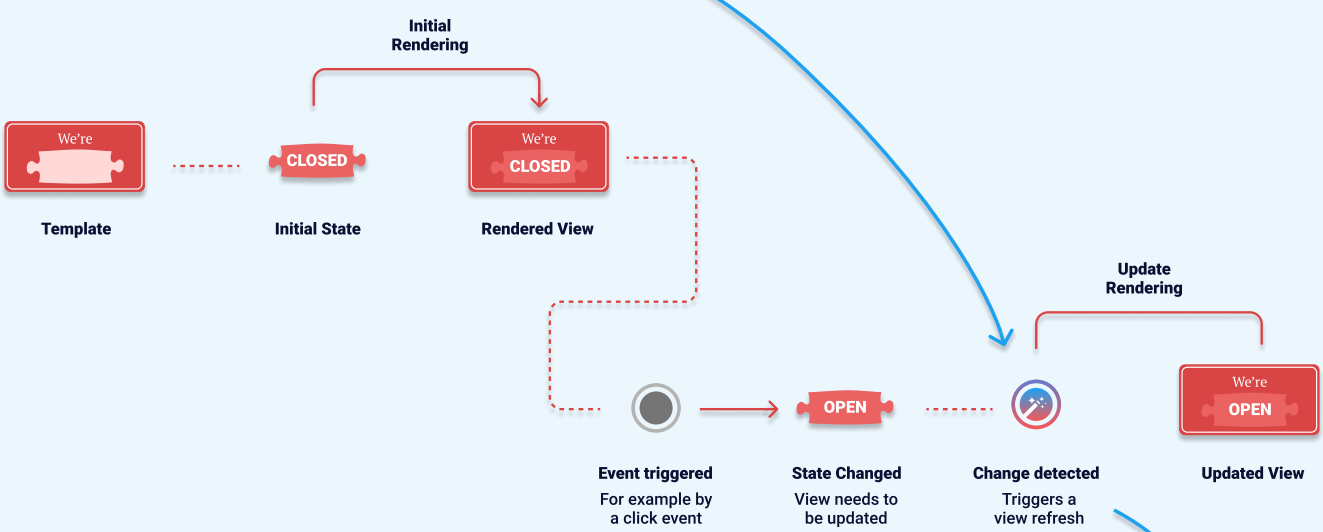
ANGULAR IVY EDITION

CHANGE DETECTION

What is Change Detection?

It detects changes in the state and triggers a refresh

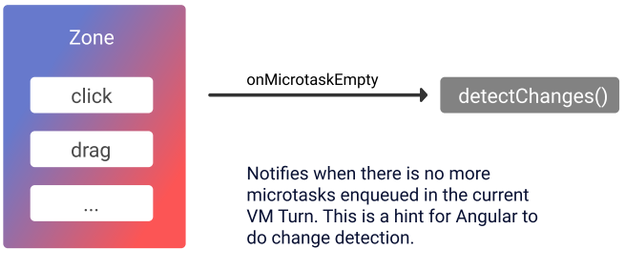


How is Angular detecting change?

What is the magic behing Angular change detection?

Patching native events

Zone.js patches most standard web APIs (such as DOM events, XMLHttpRequest, ...)



Trigger change detection

Every time one of those patched event is triggered, zone.js triggers change detection

How does Angular refresh the views?

What happens after Angular detects a change? Angular needs to refresh all the affected views. There are two strategies we can choose from. Default and OnPush.

Default (aka CheckAlways) vs. OnPush

```
@Component({
  ...
  changeDetection: ChangeDetectionStrategy.Default
})
export class ItemComponent {}
```

```
@Component({
  ...
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class ItemComponent {}
```

All Views are refreshed on every tick. That way we are sure that no view is “out of sync” with the state.

Views are refreshed when the view is dirty.

because

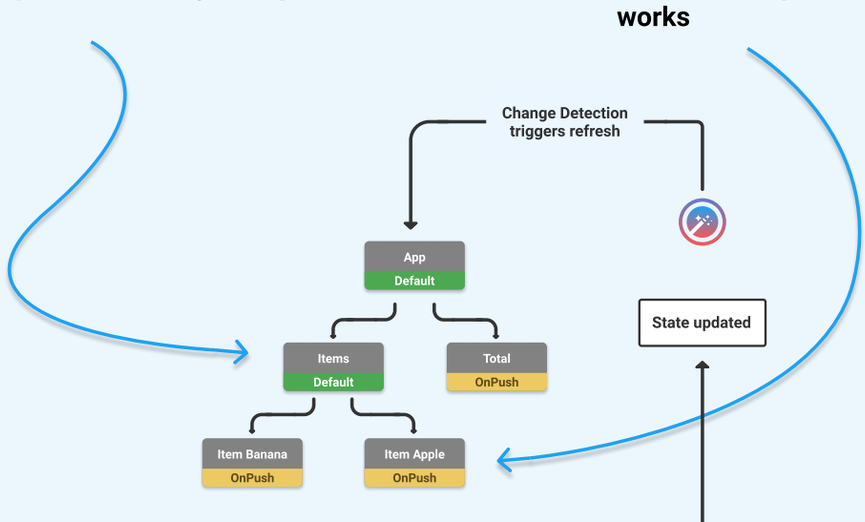
- ✅ Inputs changed
- ✅ Event was emitted from within the view
- ✅ Manually set to markFor Check

🚀 **Simple**

👉 **Can be slow since the full app is updated on every change**

🚀 **Can improve performance**

👉 **Requires a good understanding of how it works**



Shallow comparison

Inputs are compared by reference or primitive type. This means if an input is bound to an object and only a property of the object changes, no input change is detected.

```
@Component({
  selector: "person",
  template: "{{ person.age }}",
  changeDetection: ChangeDetectionStrategy.OnPush
})
export class PersonComponent {
  @Input() person;
}
```

```
<person [person]="person"></person>
```

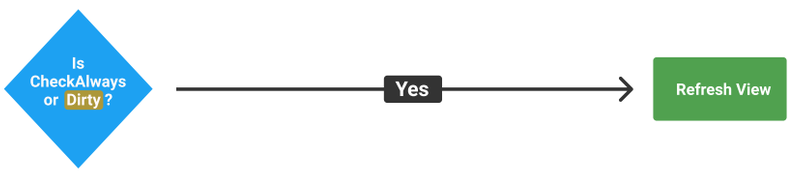
```
// Not working
// Reference unchanged
person.age = 50
```

```
// Working
// Reference changed
person = { age: 50 }
```

Behind the scenes of change detection

What happens when the user clicks a button?

The importance of Set Dirty



A view is either refreshed when the component is dirty or set to CheckAlways

A view is set to dirty if an input changed, an event was emited from a view or its ancestors or it was set manually to dirty

The two main phases of change detection

“Set Dirty” Phase

Bubble up from the view where the event was triggered and set the view and all its ancestors to dirty

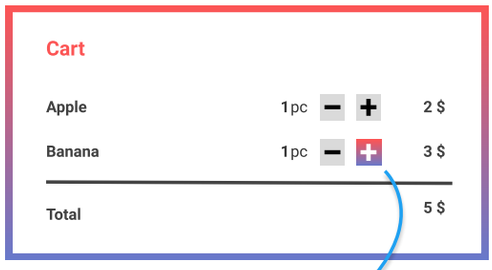
Change detection Phase

Traverse down from the root view and run change detection on the views.

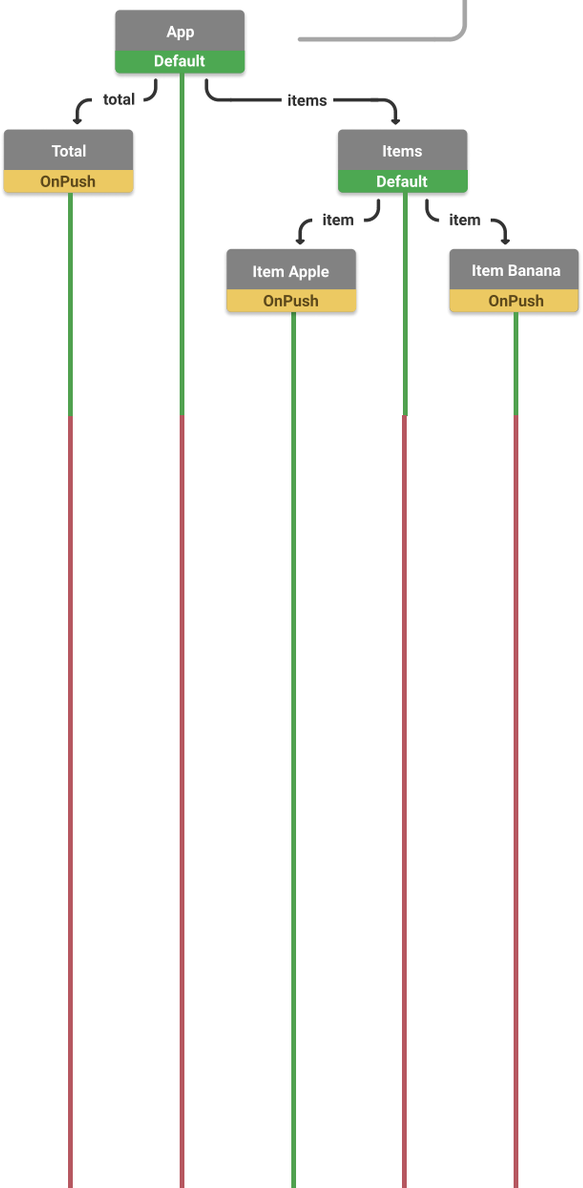
Example App

We use the example of a shopping cart to visualize how Angular applies changes to its views. The example shows what happens when the button is clicked.

```
const state = {
  items: [
    { id: 0, title: “Apple”,  qty: 1, cost: 2 },
    { id: 1, title: “Banana”, qty: 1, cost: 3 },
  ],
  get total() {
    // implementation of calcTotal is not relevant
    return calcTotal(this.items);
  }
}
```



The state is part of the app component and then forwarded via @Inputs to its child components.



Change

An application change usually happens within a view.

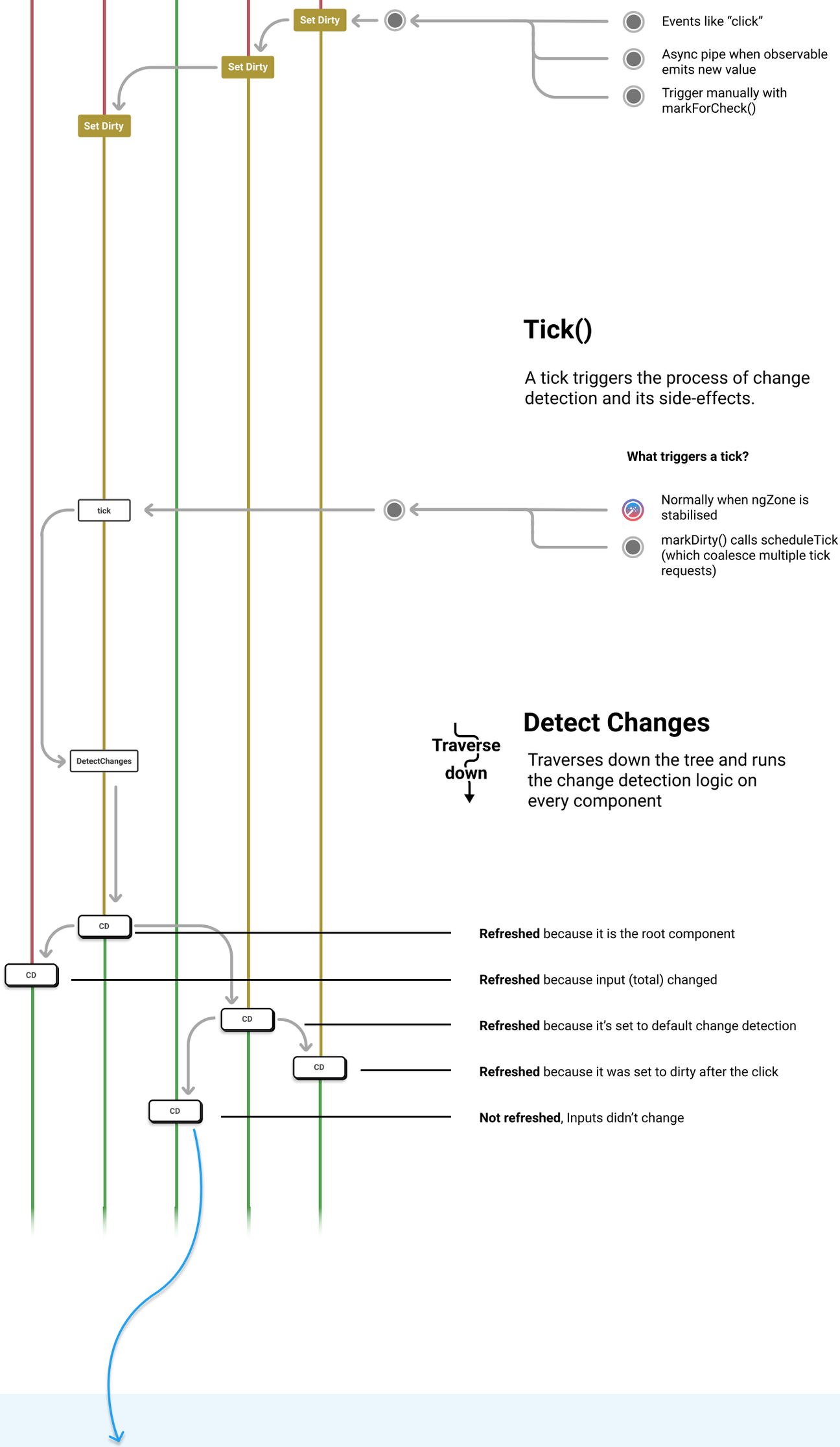
The button within the “Item Banana” view was pressed. This changes the qty of bananas in the state.

“Set Dirty” Phase

To make sure the affected components are updated on a refresh, the component and all its ancestors are marked dirty.

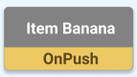
Without this process it could happen that the view is not updated if it is set to “OnPush” and the inputs didn’t change.

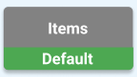
What triggers “SetDirty”?


Set Dirty
Set Dirty
Set Dirty
Events like “click”
Async pipe when observable emits new value
Trigger manually with markForCheck()
Tick()
A tick triggers the process of change detection and its side-effects.
What triggers a tick?
tick
Normally when ngZone is stabilised
markDirty() calls scheduleTick (which coalesce multiple tick requests)
Detect Changes
Traverse down
Traverses down the tree and runs the change detection logic on every component
DetectChanges
CD
CD
CD
CD
CD
Refreshed because it is the root component
Refreshed because input (total) changed
Refreshed because it’s set to default change detection
Refreshed because it was set to dirty after the click
Not refreshed, Inputs didn’t change

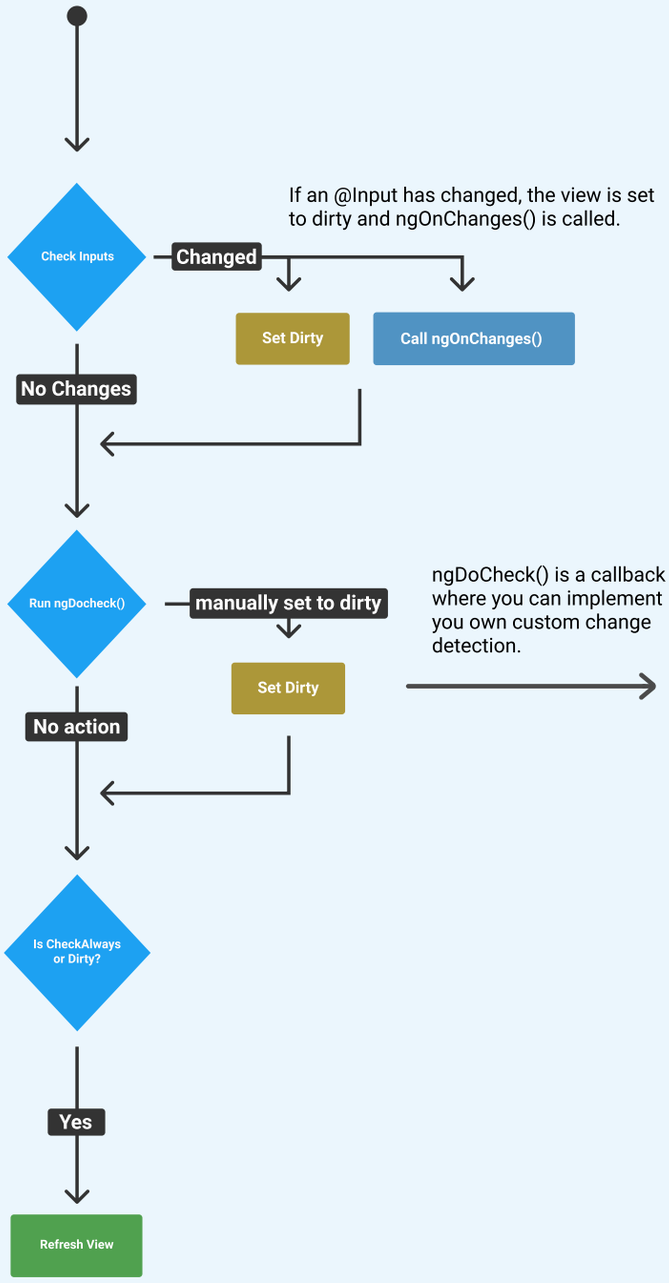
Change detection logic on every component

After the previous steps, some views are already set to dirty during the “Set dirty” phase.

For example 

or its parent 

If the view isn’t dirty or CheckAlways, it can still be set to dirty if the inputs changed or they are manually set to dirty in ngDoCheck()



Warning: ngOnChanges != “View gets refreshed”

Don’t mistake ngOnChanges for a callback which is called when a view is refreshed.
ngOnChanges is only called when an @Input changed.

It is not called when the view is refreshed for other reasons. So it could be a false positive.

```
export class AppComponent implements DoCheck {
  constructor(private cdRef: ChangeDetectorRef) {}

   ngDoCheck() {
    if (myCustomCheck) {
      this.cdRef.markForCheck();
    }
  }
}
```

A View gets refreshed when..

To sum it up: Default change detection always triggers a refresh, OnPush only when an event was triggered within the component or its children, or an input has changed.

Default (aka CheckAlways)

✅ Always

OnPush

✅ Inputs changed

✅ Was manually set to dirty in ngDoCheck

✅ Was set to dirty in the “Set dirty” phase

⚠️ Be aware that inputs are compared shallow

Chris Kohler
christiankohler.net

@kohlerchristian

Version 1.0 © 2021 Chris Kohler