

Angular2+进阶开发实战

...

李仲辉 @广发证券

关于我

- Senior engineer @GF Securities (广发证券)
- Core co-author of "揭秘Angular 2"
- Speaker @imooc.com
- Webpack China official team member
- Former tencent&baidu

大纲

(一) 性能优化探索 (运行时)

(二) 工具与工程化 (静态)

(三) 常见问题原因及解决方法

(一) 性能优化探索

Countdown to Liftoff

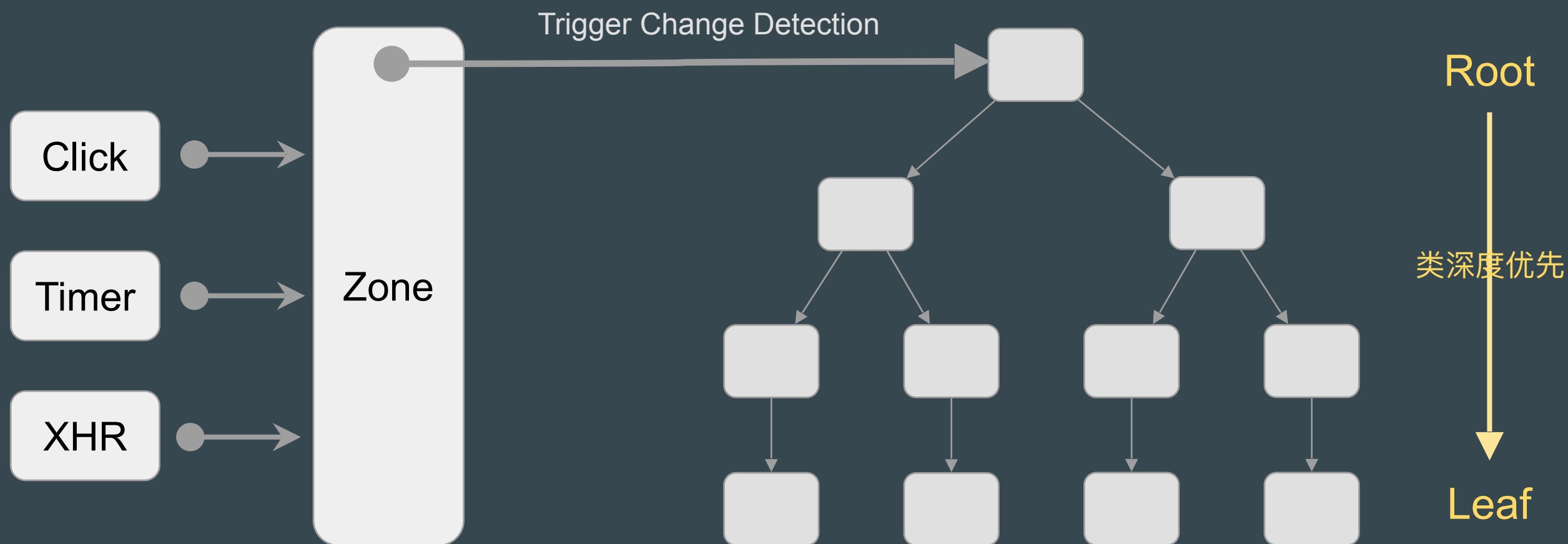
Start

Stop

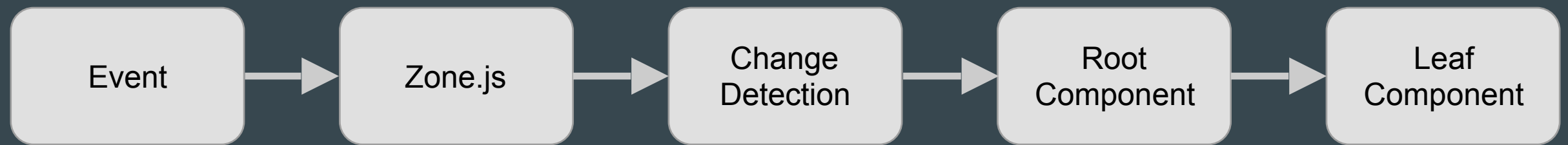
10

T-10 seconds and counting

Angular响应式处理链路



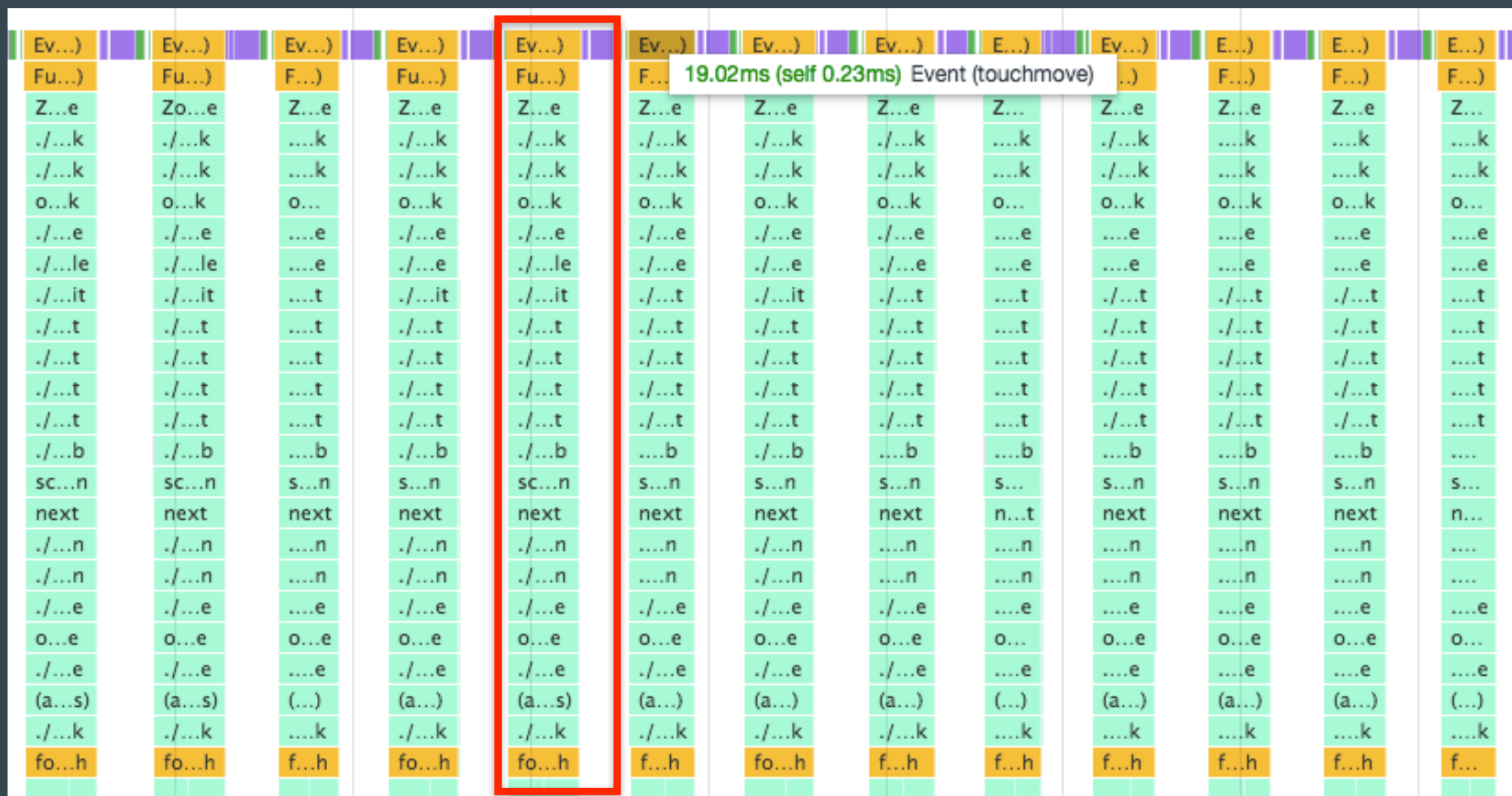
链路优化



典型例子：元素拖放


```
@Component({
  template: `
    <item
      // ...
      (touchstart)="touchstart($event)"
      (touchmove)="touchmove($event)"
      (touchend)="touchend($event)"
    ></item>
  `,
})
export class BoxComponent {
  constructor() { }
  touchstart(event: any) {
    // 记录原始位置
  }
  touchmove(event: any) {
    // 更新模型数据
  }
  touchend(event: any) {

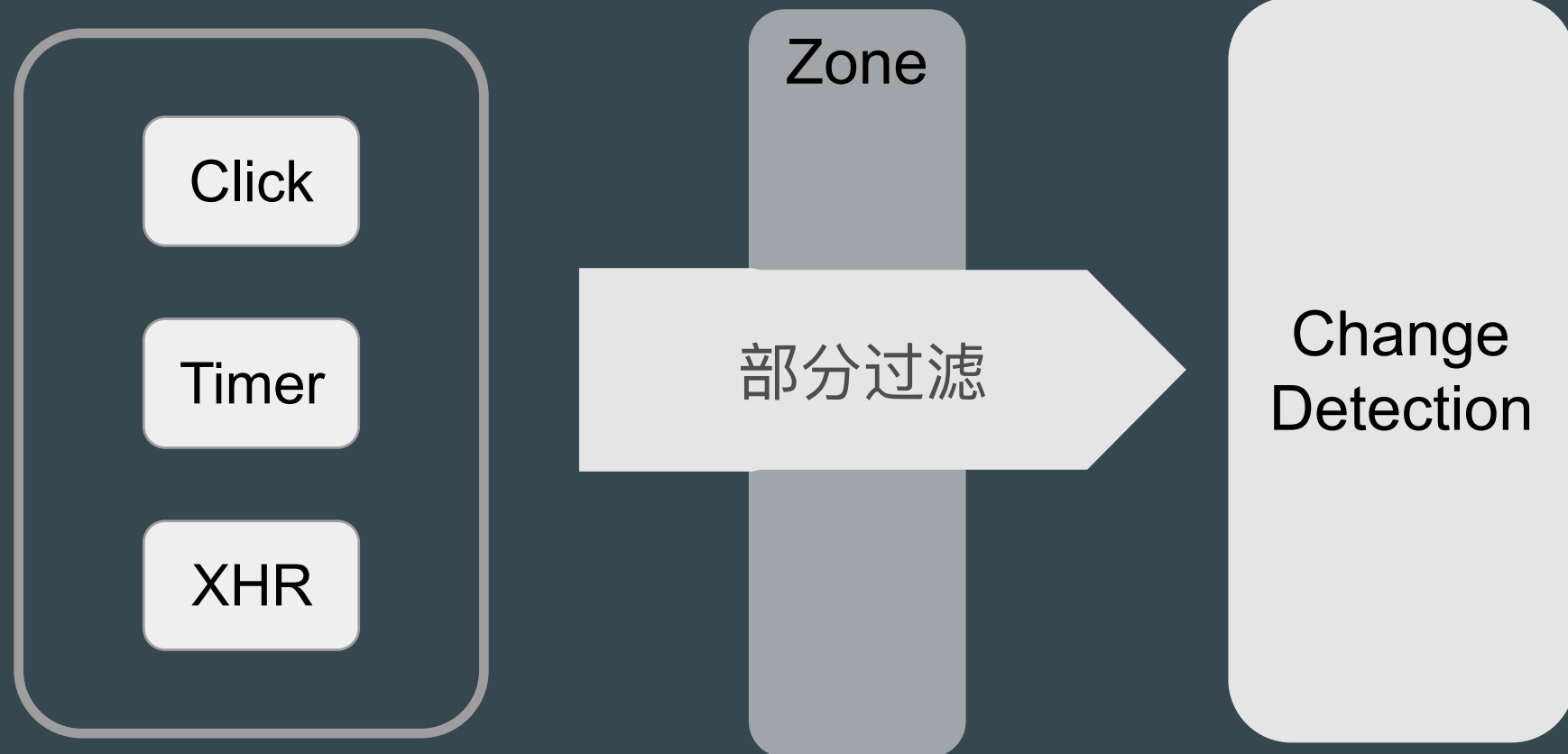
  }
}
```



红米Note 4X

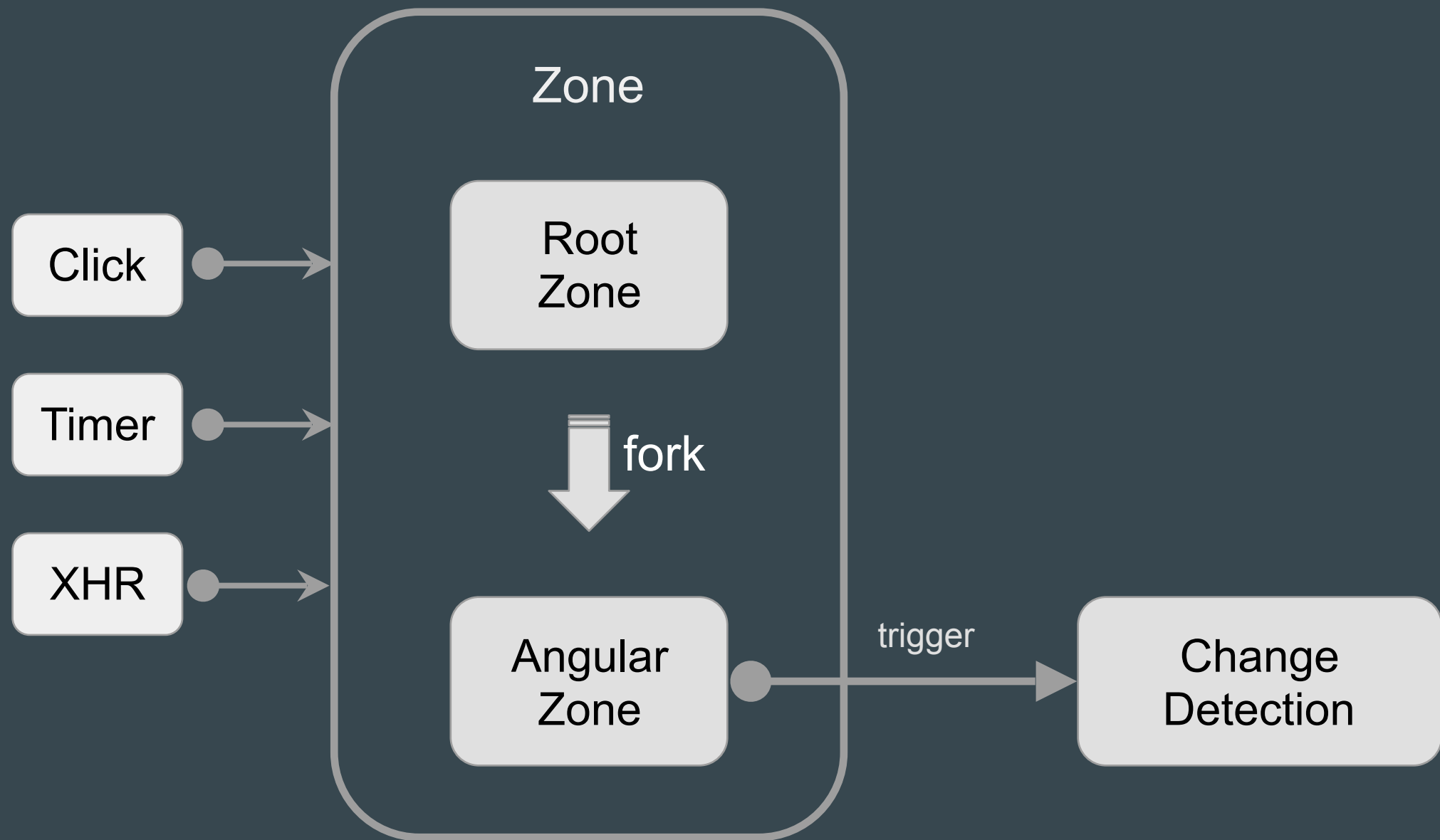
模拟500个组件 * 4个binding

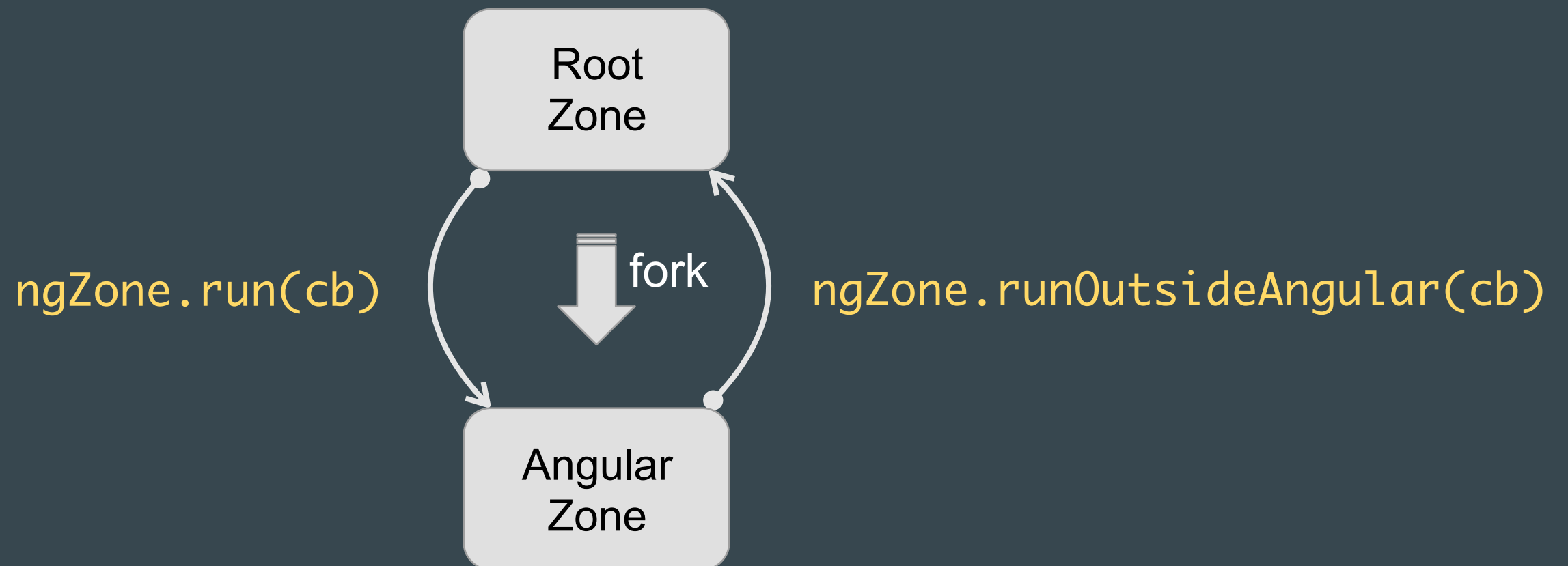
每次`touchmove`事件都触发变化检测执行，耗时约 15-20ms



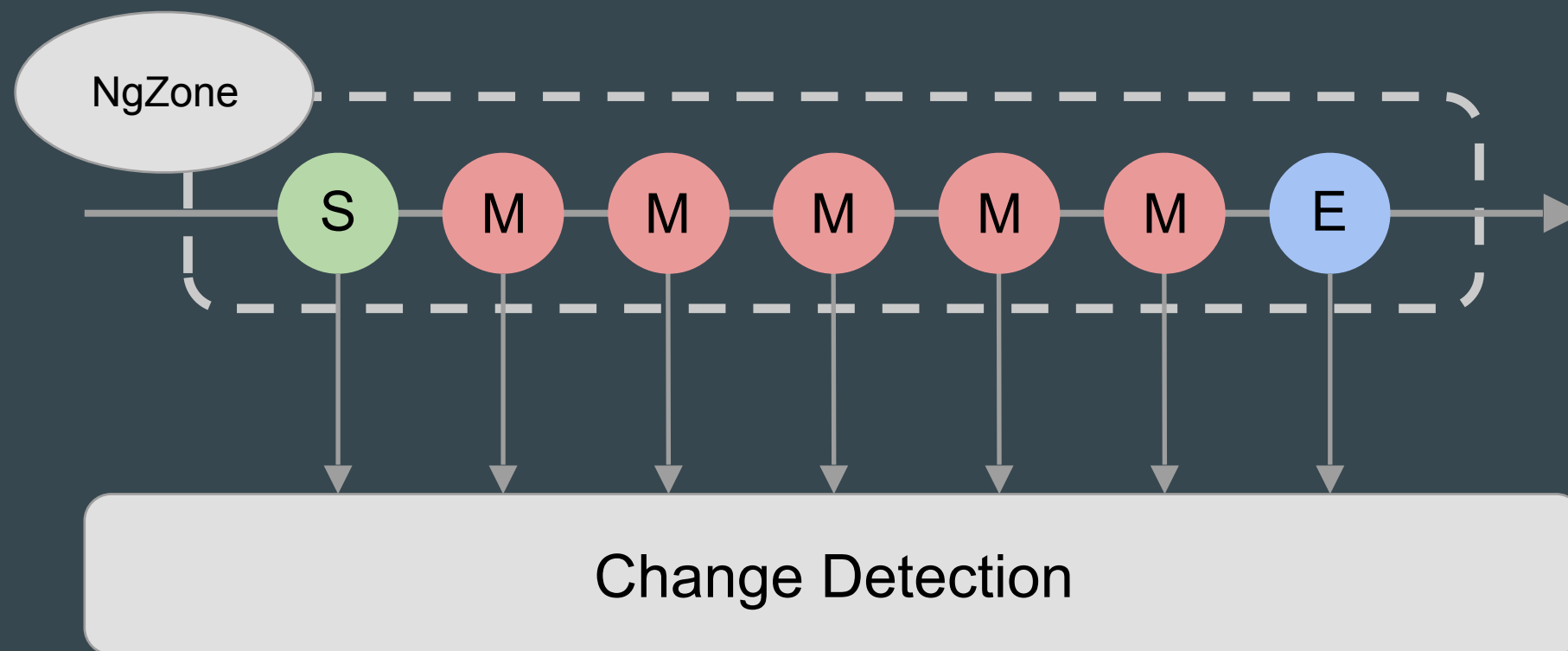
NgZone

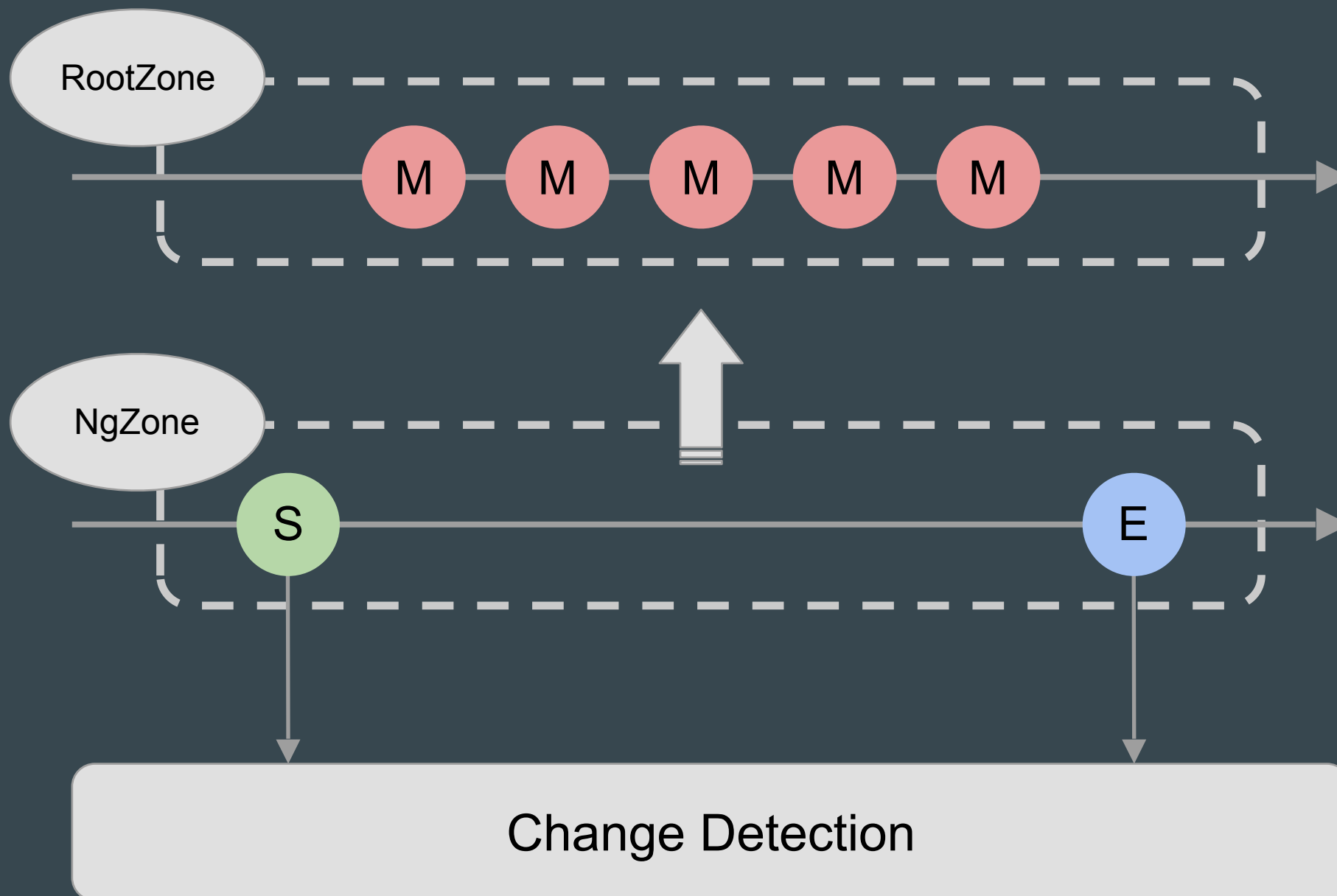
```
class NgZone {  
  run(fn: () => any) : any  
  runOutsideAngular(fn: () => any) : any  
  // ...  
}
```





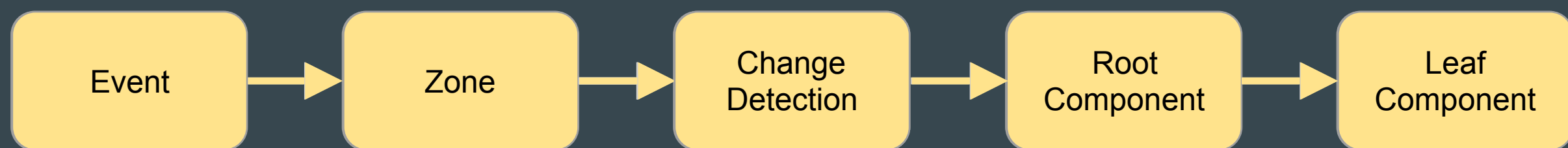
```
export class BoxComponent {  
  constructor( private ngZone: NgZone ) { }  
  touchstart(event: any) {  
    this.ngZone.runOutsideAngular(() => {  
      // 使用DOM API绑定touchmove事件  
    });  
  }  
  touchmove(event: any) {  
    // 更新元素样式  
  }  
  touchend(event: any) {  
    // 移除touchmove事件监听  
    // 更新模型数据，保持与视图一致  
    this.ngZone.run(() => {  
      // 调用run防止一些异常情况出现  
    });  
  }  
}
```

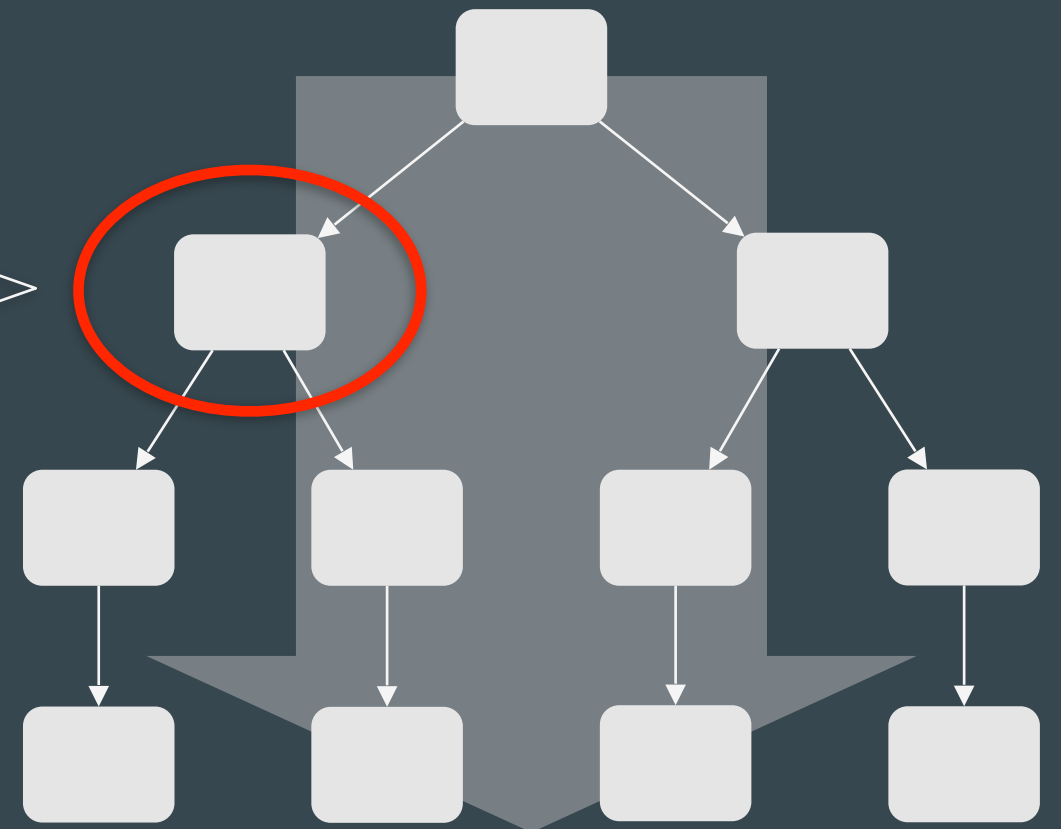
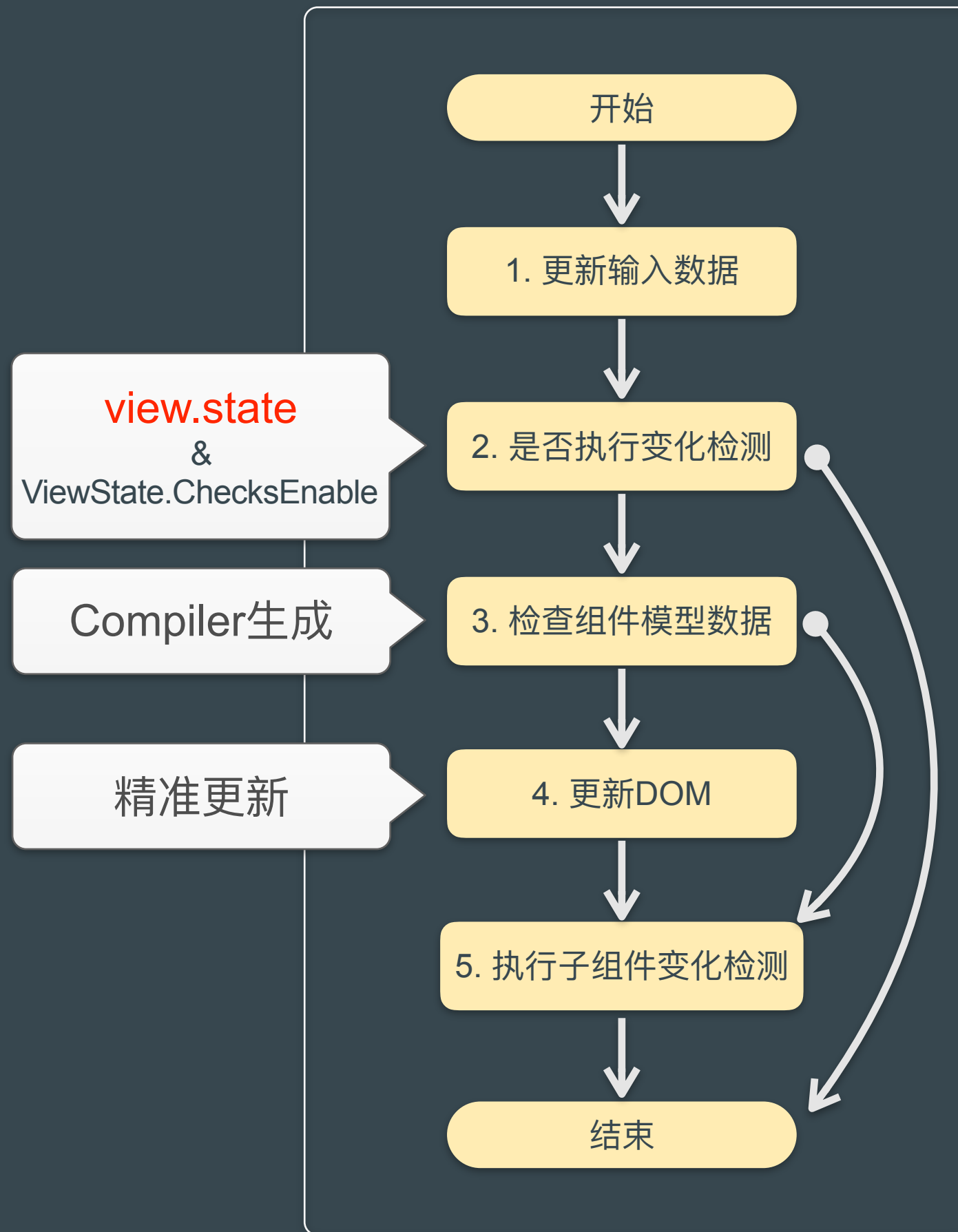






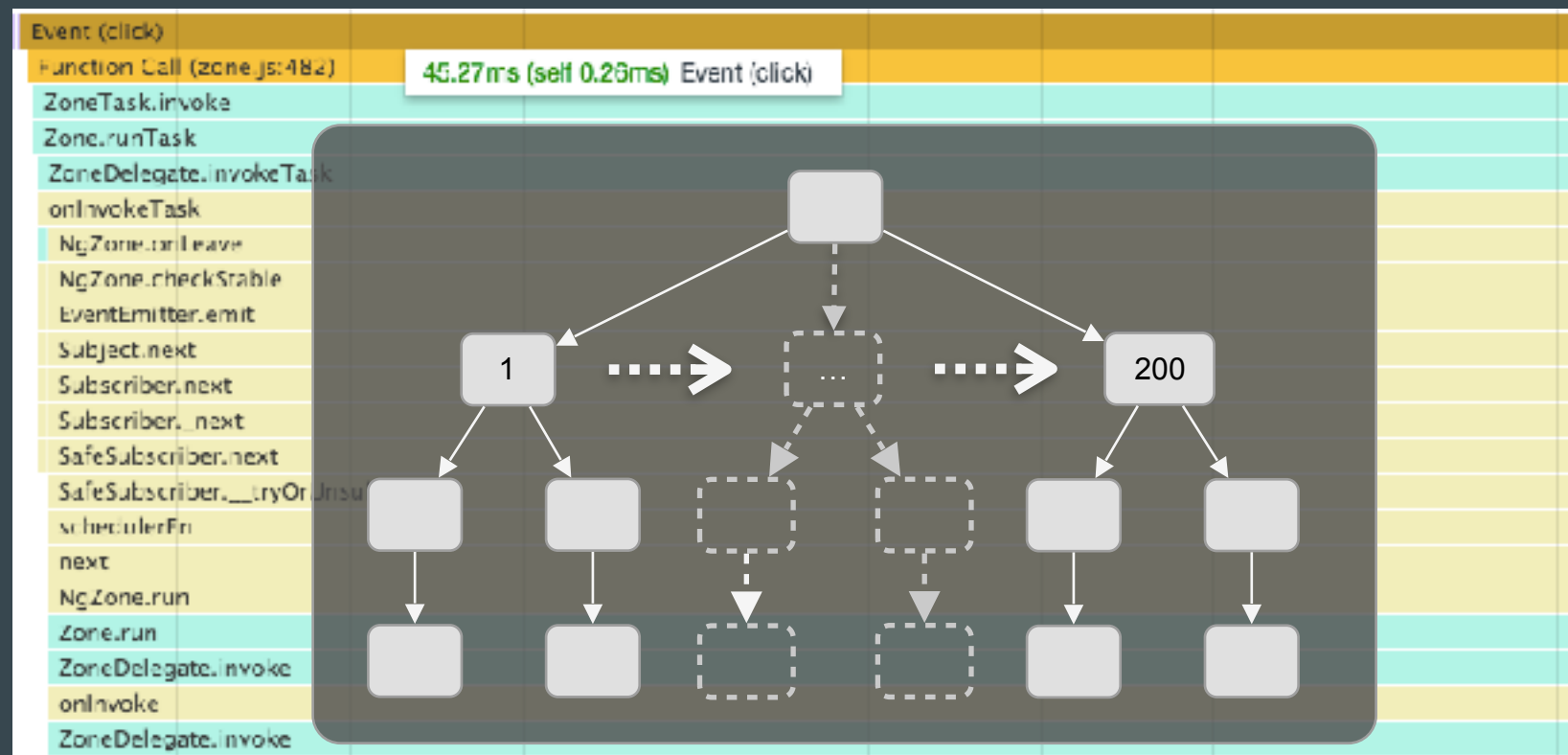
touchmove不再触发变化检测逻辑





基准性能衡量

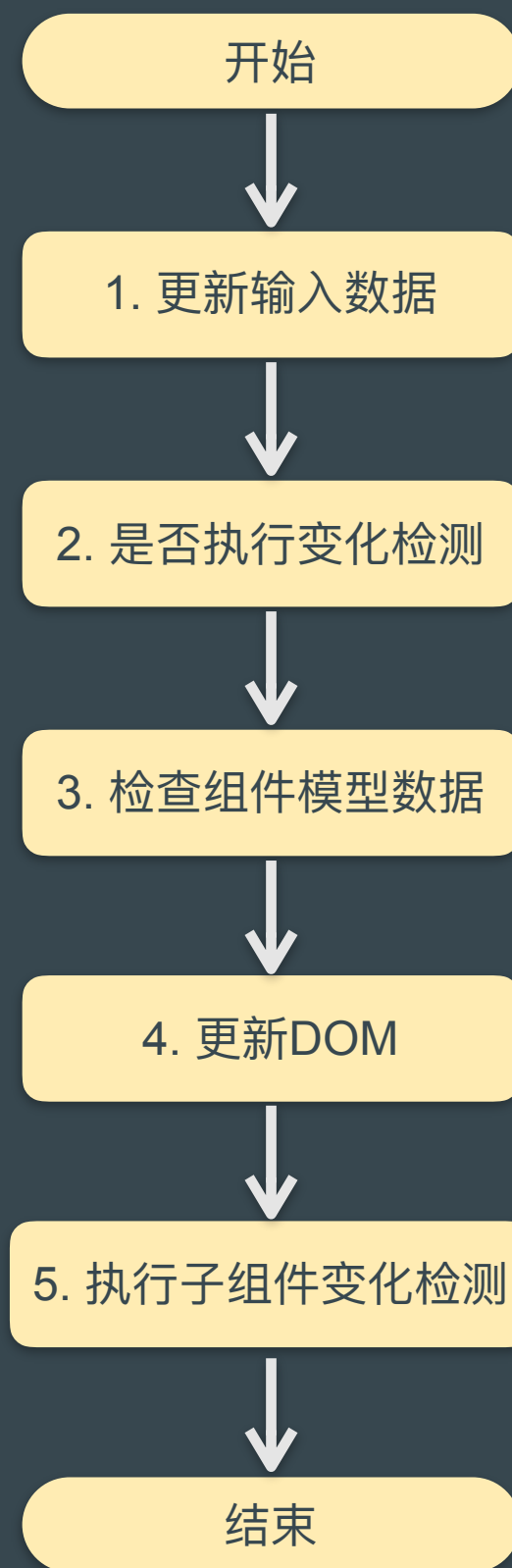
1. 单向变化检测
2. 组件数据模型优化
 - a. VM-friendliness
 - b. 只检查模板上绑定的变量（非深度遍历对象）



红米 Note 4X

1000个组件 * 3个binding

一次变化检测平均约 **45ms**



view.state **&=** ViewState.ChecksEnable



Click, Timer, XHR

强力武器

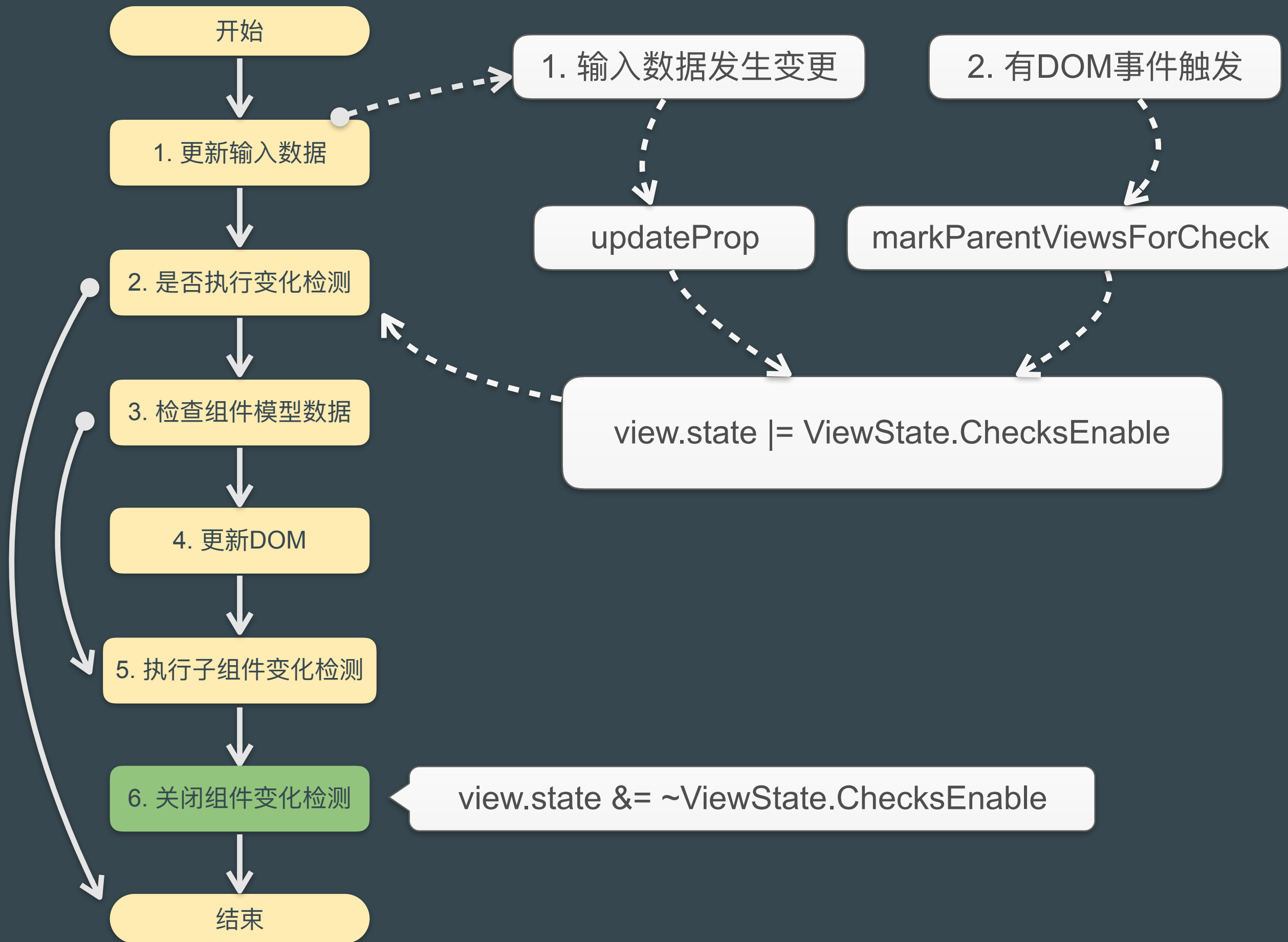
```
class ChangeDetectionStrategy {  
    Default  
    OnPush  
}
```

Default

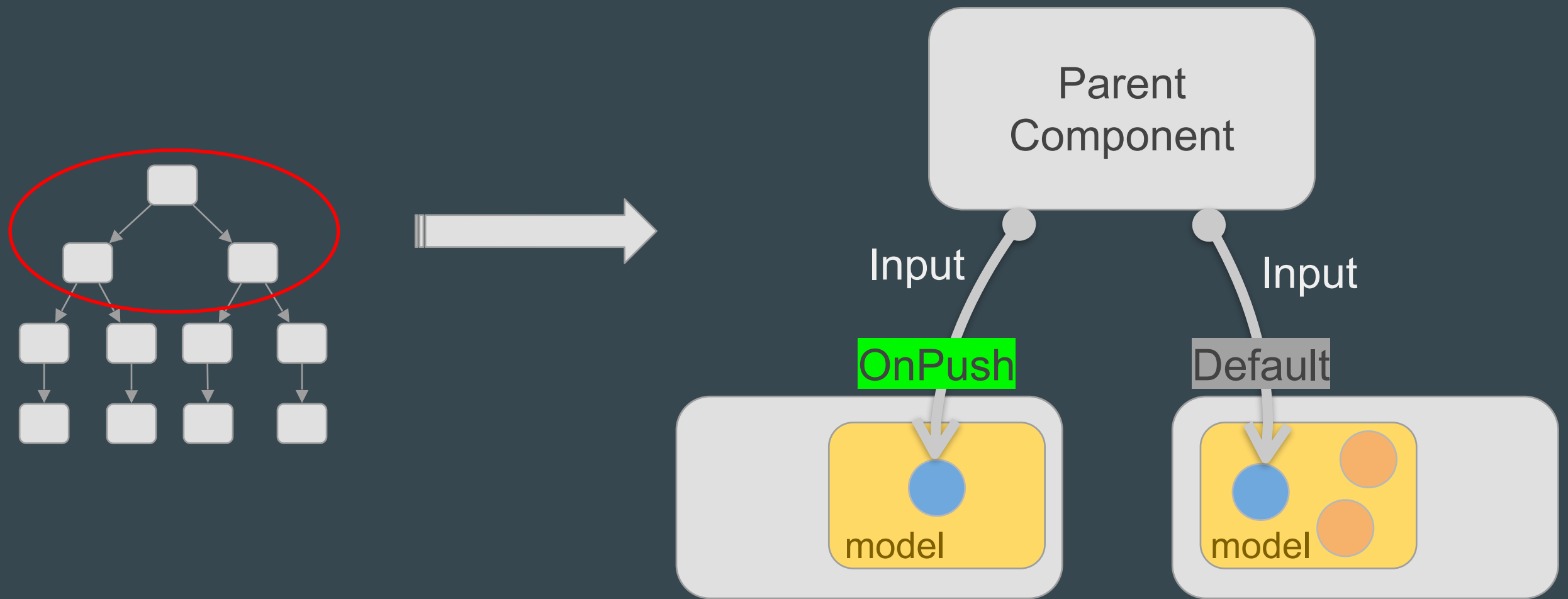
每次变化检测阶段都检测
值是否变动

OnPush

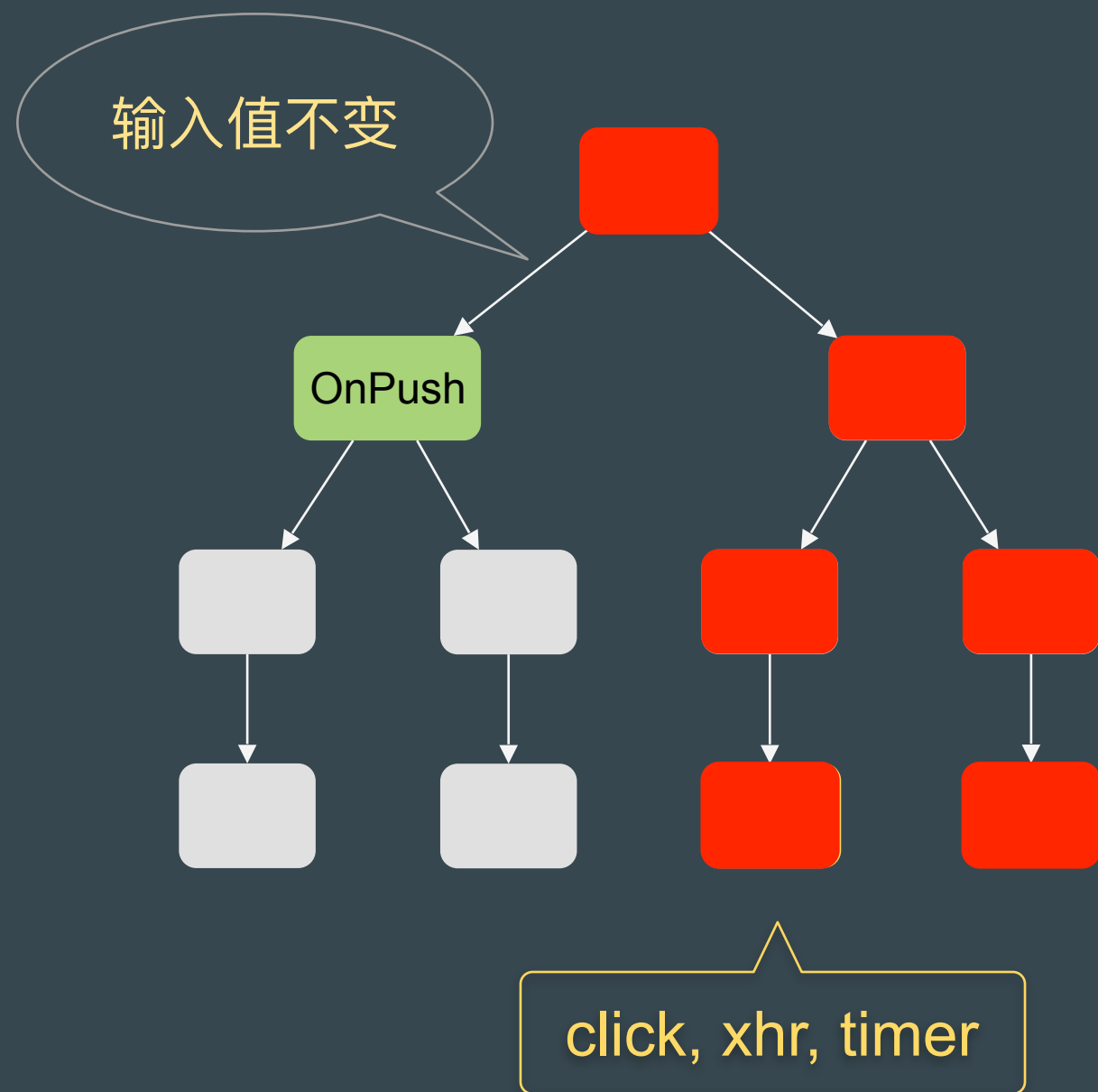
1. 当前组件的输入数据发生变更
2. 当前组件及其子组件模板有DOM事件触发

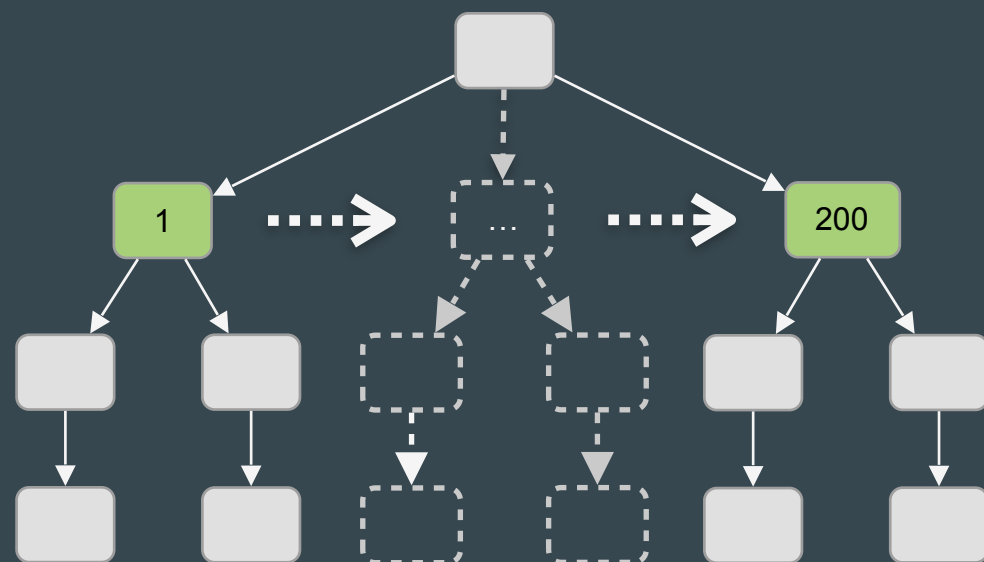
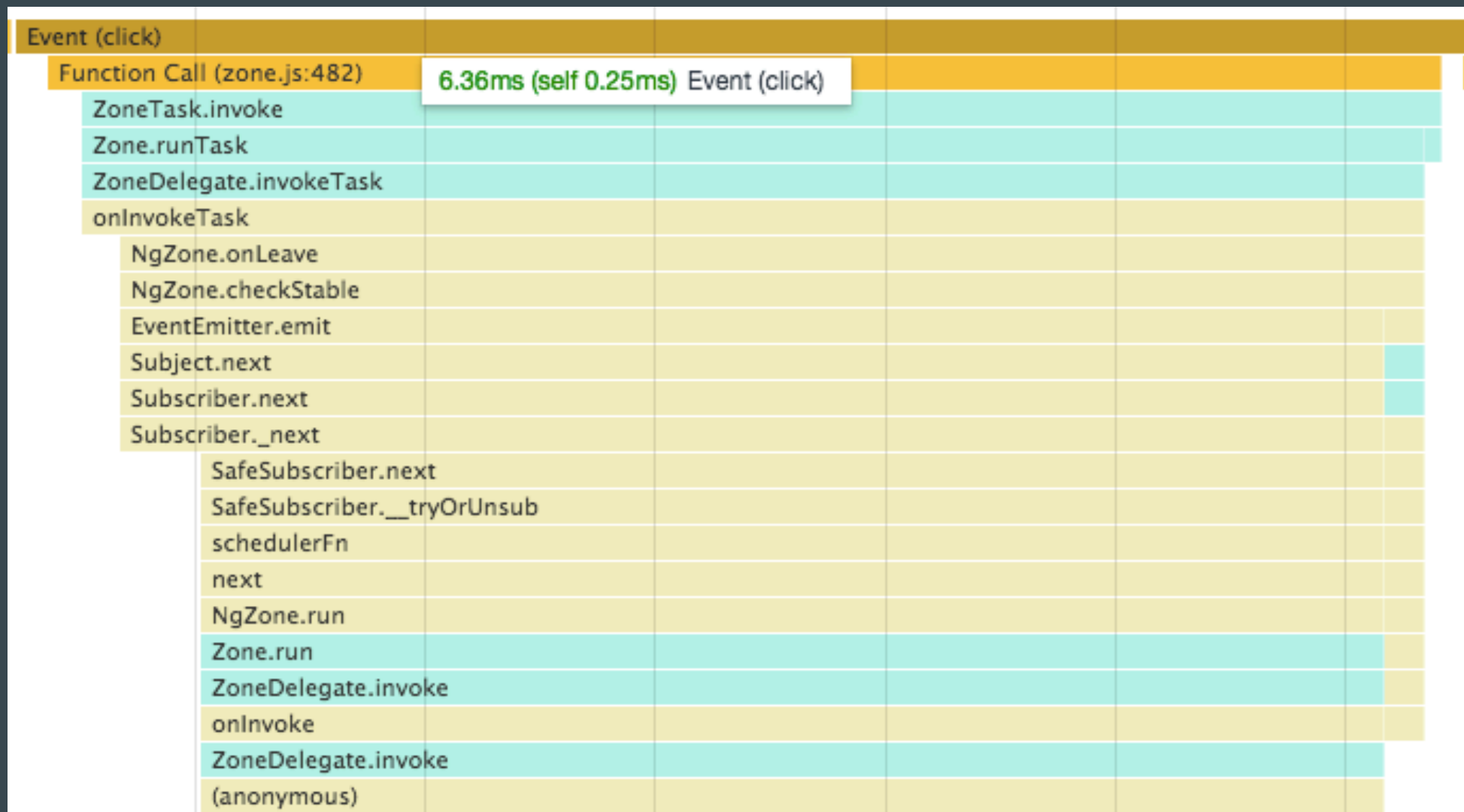


ChangeDetectionStrategy



组件数据只依赖父组件的输入数据





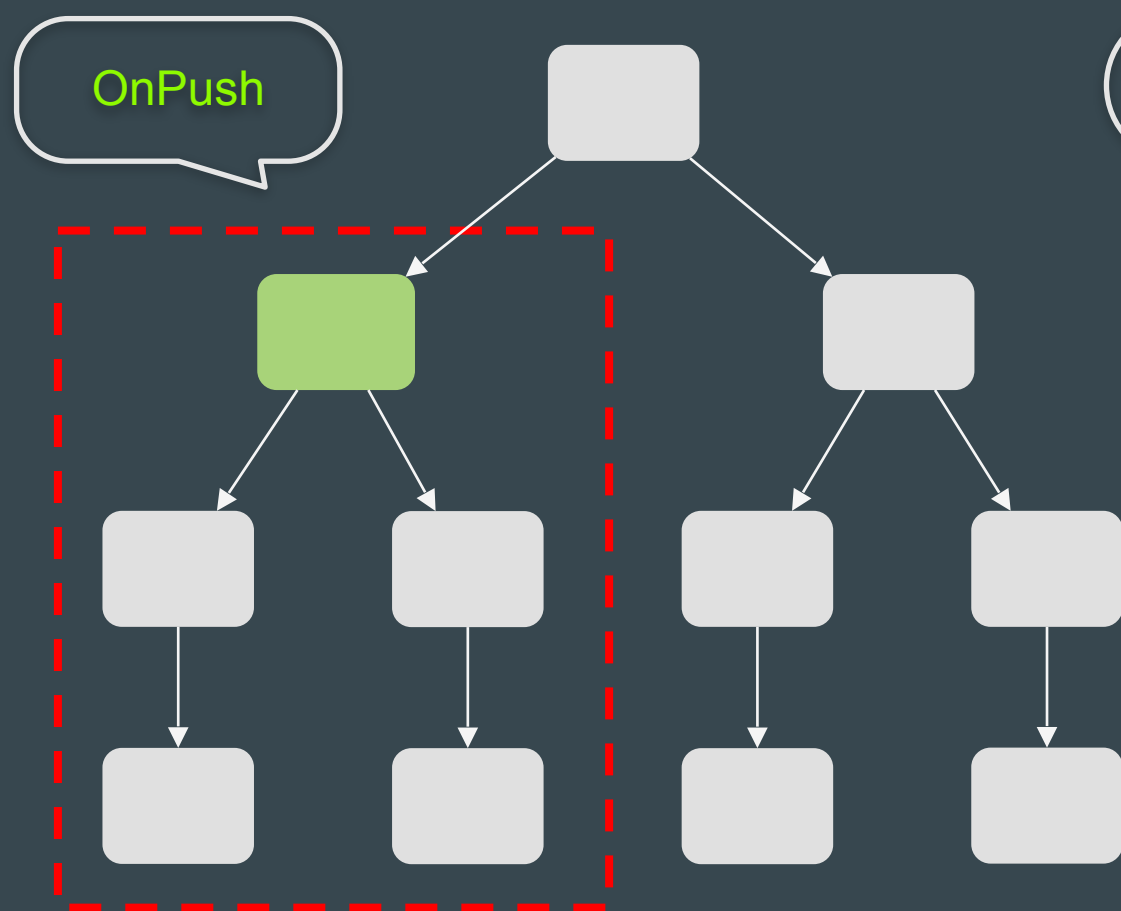
红米 Note 4X

1000个组件 * 3个binding

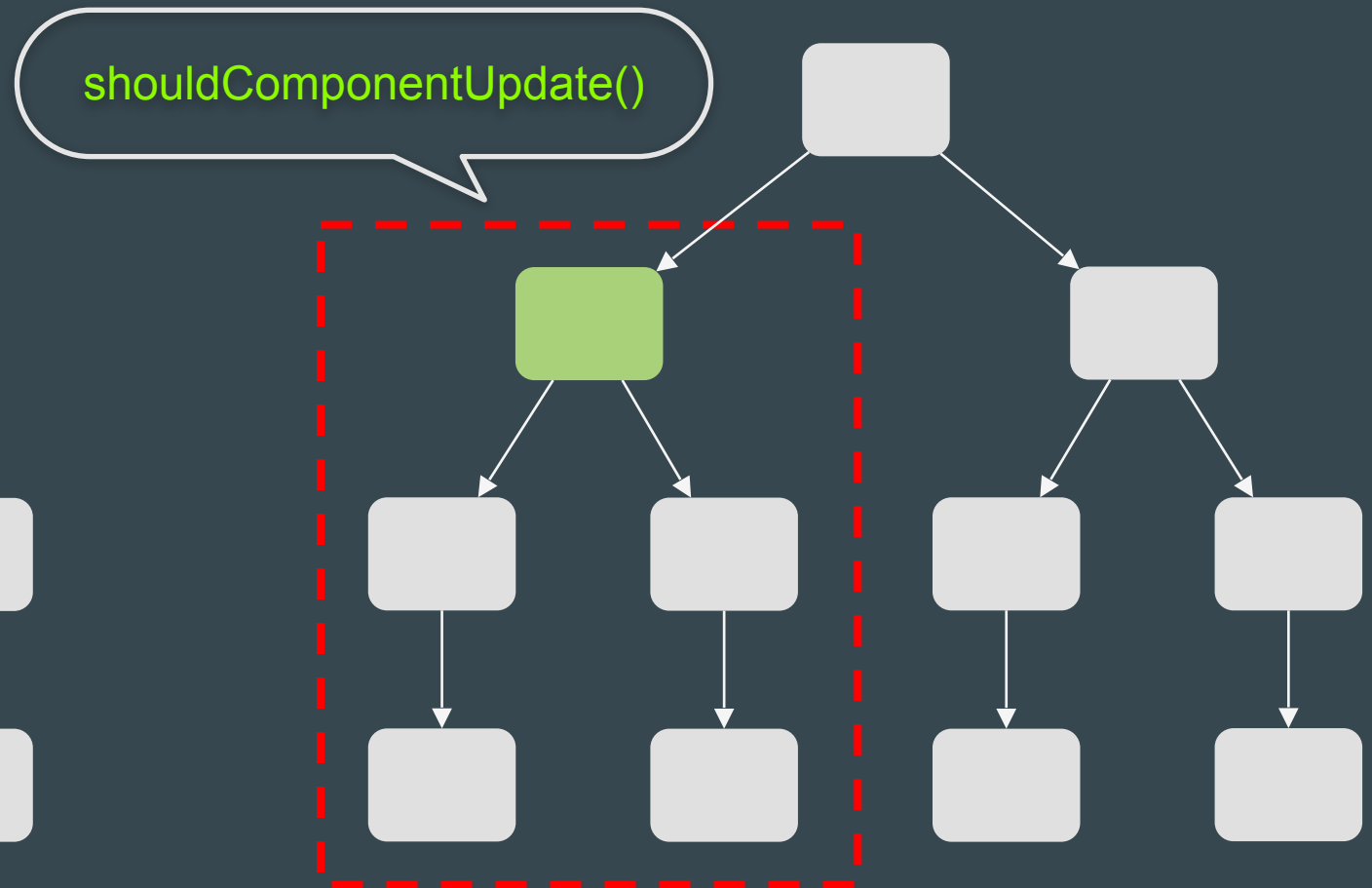
一次变化检测约花费 **7ms** 上下

性能提升约 **80%**

Angular vs React 异曲同工

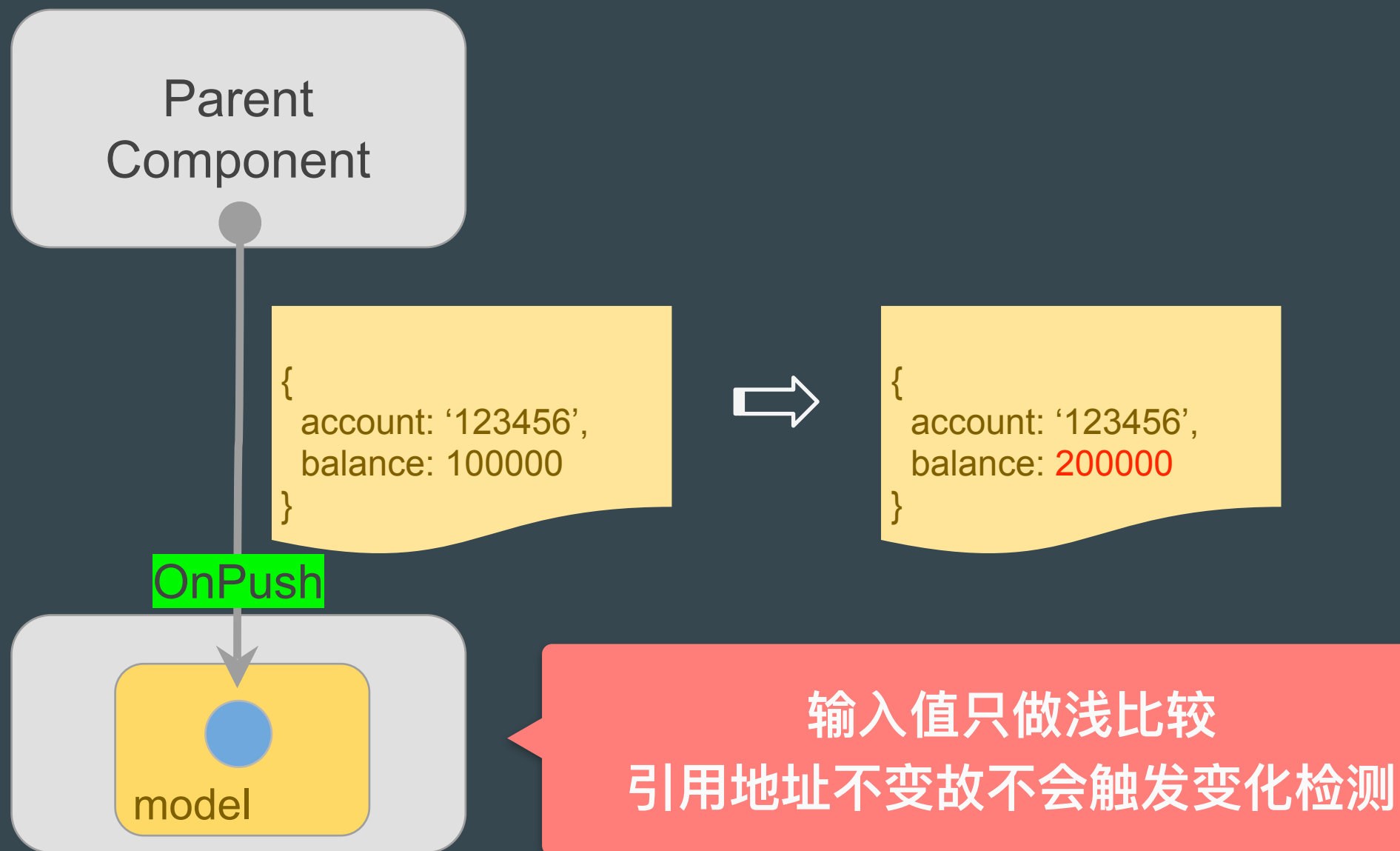


Angular



React + Redux

OnPush注意点



Immutable

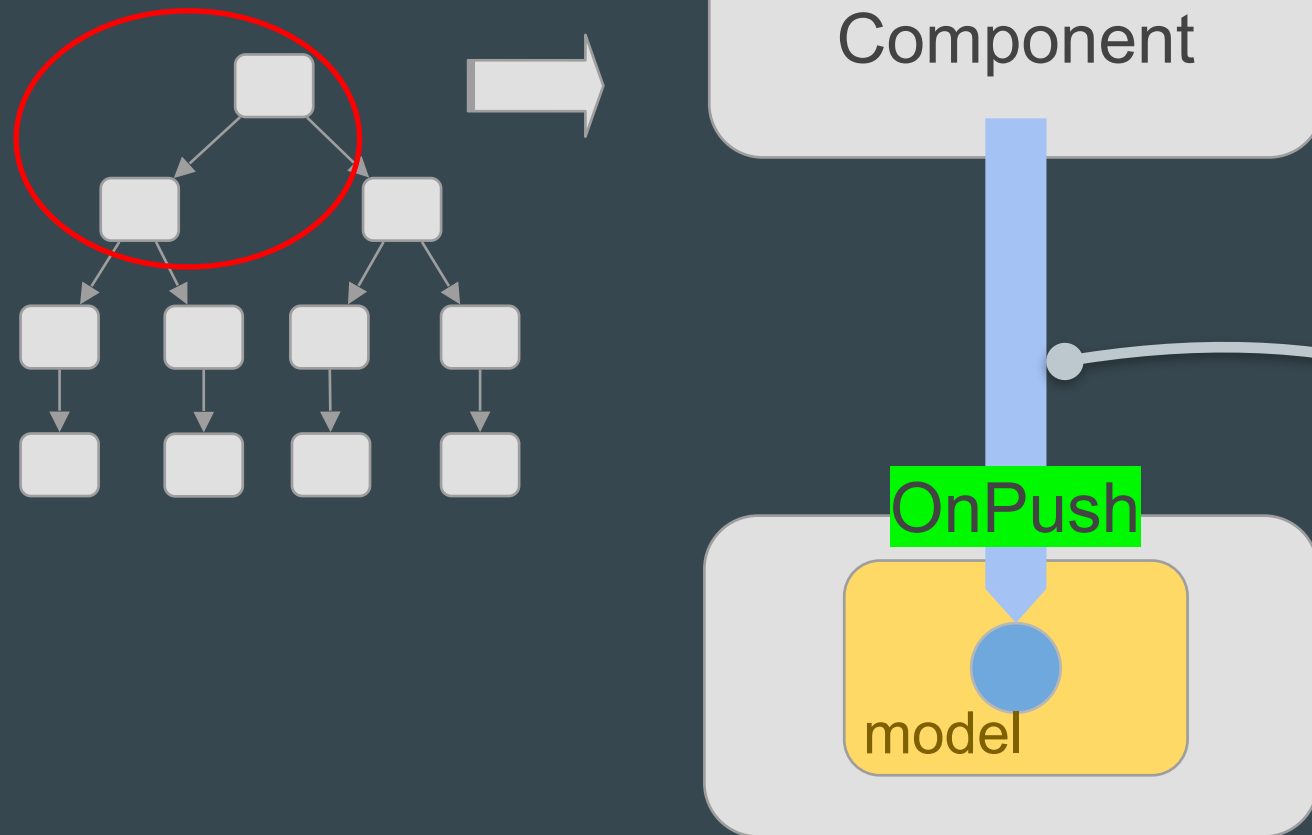
```
@Component({
  selector: 'my-app',
  template: '<my-child [data]="dataToChild"></my-child>'
})
export class MyComponent {
  private dataToChild: any;
  ngOnInit() {
    this.dataToChild = Immutable.Map({
      account: 123456,
      balance: 100000
    });
  }
  charge() {
    this.dataToChild = this.dataToChild.set('balance', 200000);
  }
}
```


Object.assign()

```
@Component({
  selector: 'my-app',
  template: '<my-child [data]="dataToChild"></my-child>'
})
export class MyComponent {
  dataToChild: any;
  ngOnInit() {
    this.dataToChild = {
      account: 123456,
      balance: 100000
    };
  }
  charge() {
    this.dataToChild = Object.assign({}, this.dataToChild, { balance: 20000 });
  }
}
```

lodash.merge()

OnPush的局限



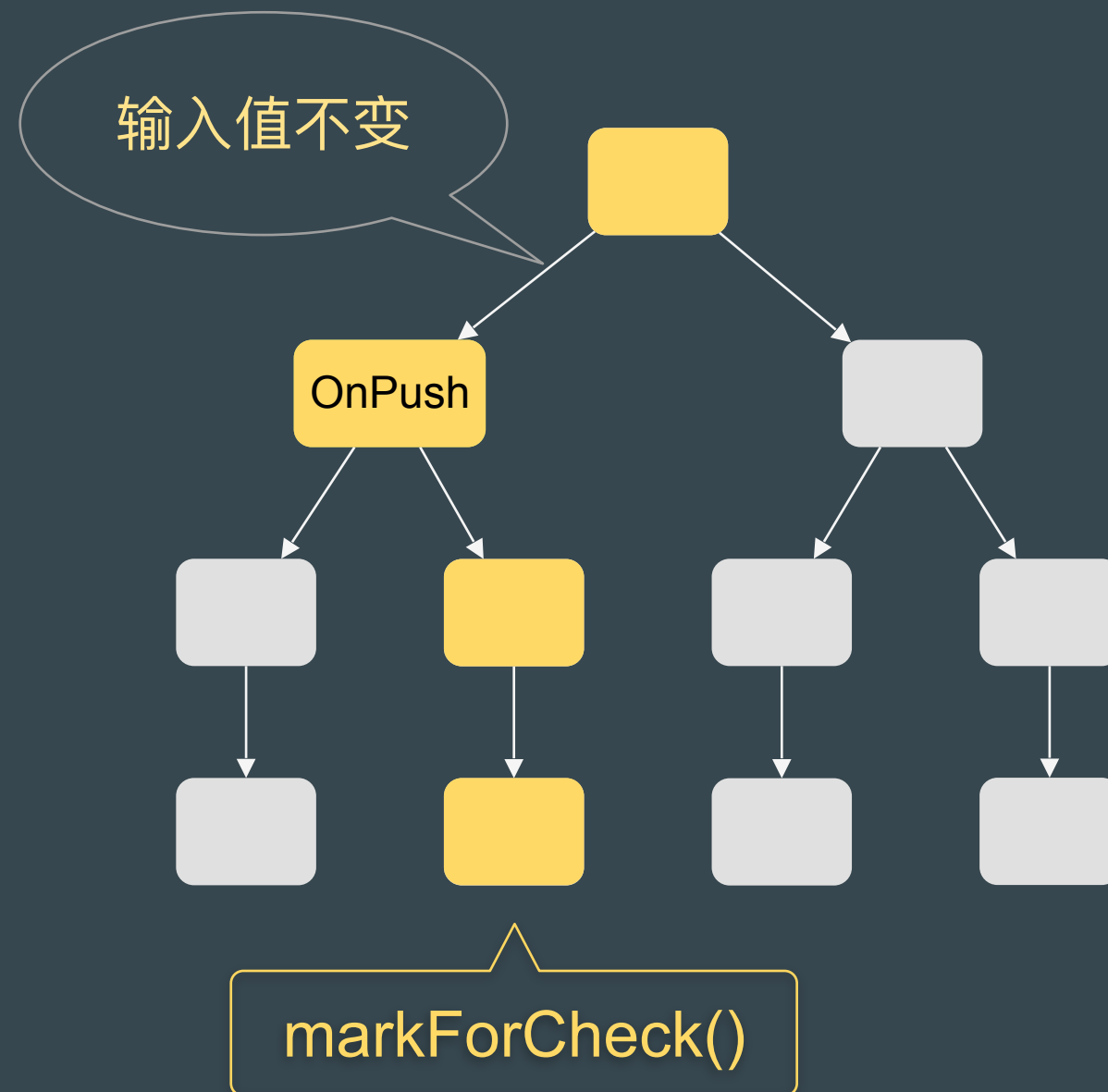
视图能更新?

```
@Component({
  template: `
    <p>Current Value: {{ curValue }}</p>
  `
})
export class BranchComponent {
  curValue: string;
  @Input() stream: Observable<string>;
  ngOnInit() {
    this.stream.subscribe(value => {
      this.curValue = value;
    });
  }
}
```

更灵活的方案

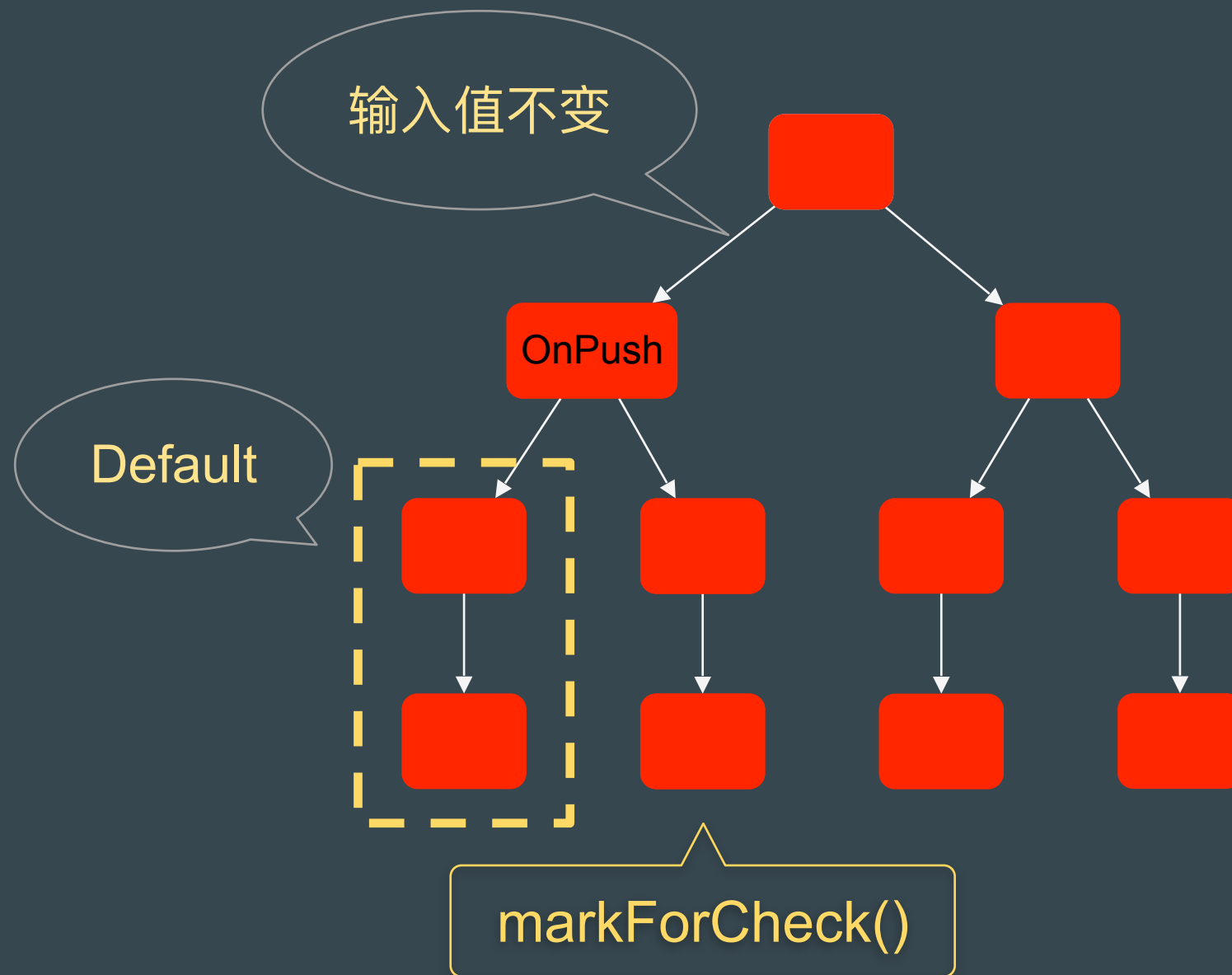
```
class ChangeDetectorRef {  
    markForCheck() : void  
    detach() : void  
    reattach() : void  
    detectChanges() : void  
    checkNoChanges() : void  
}
```

✓ markForCheck: 把当前分支的所有组件标记为可执行变化检测



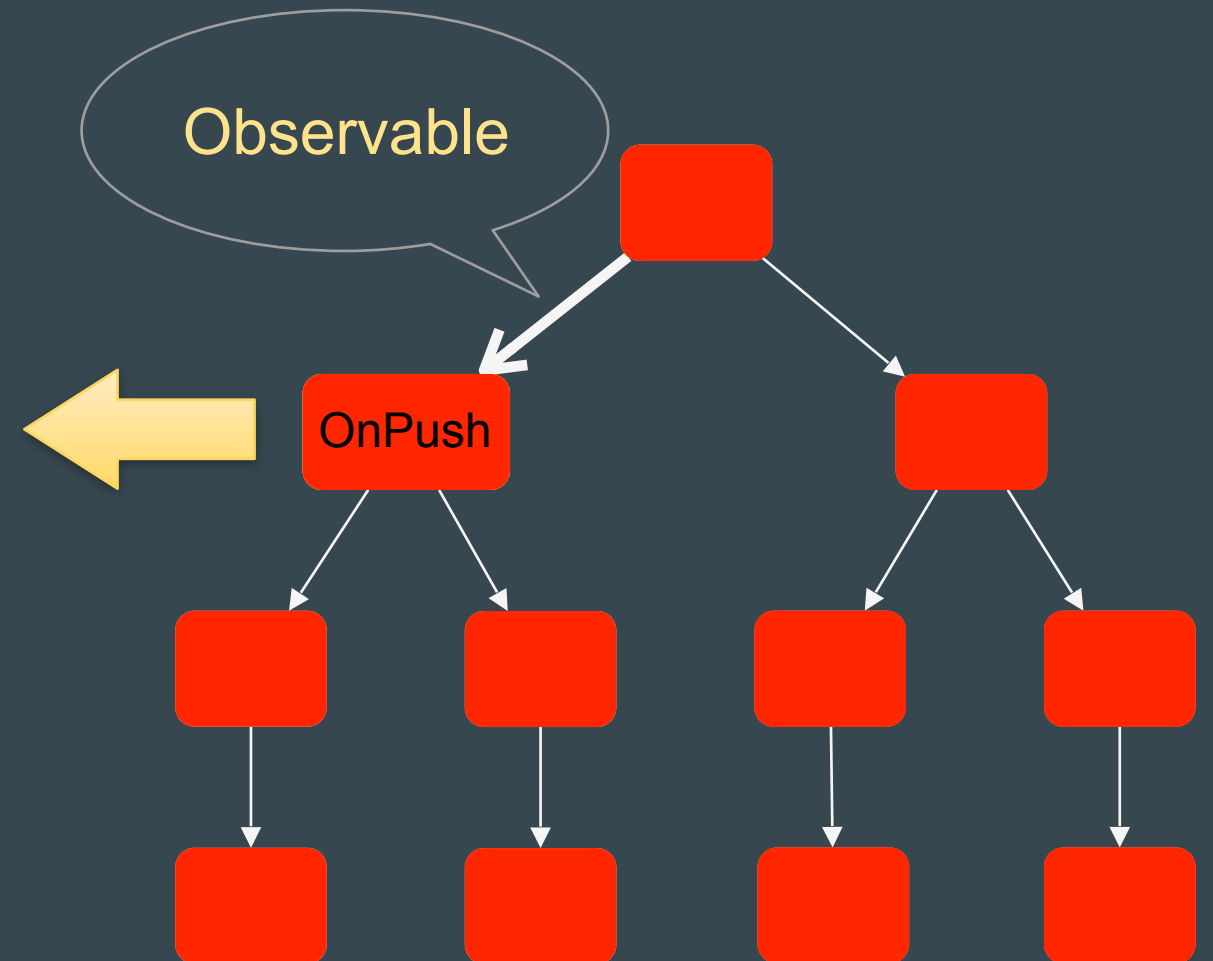
`view.state |= ViewState.ChecksEnable`

✓ markForCheck: 把当前分支的所有组件标记为可执行变化检测



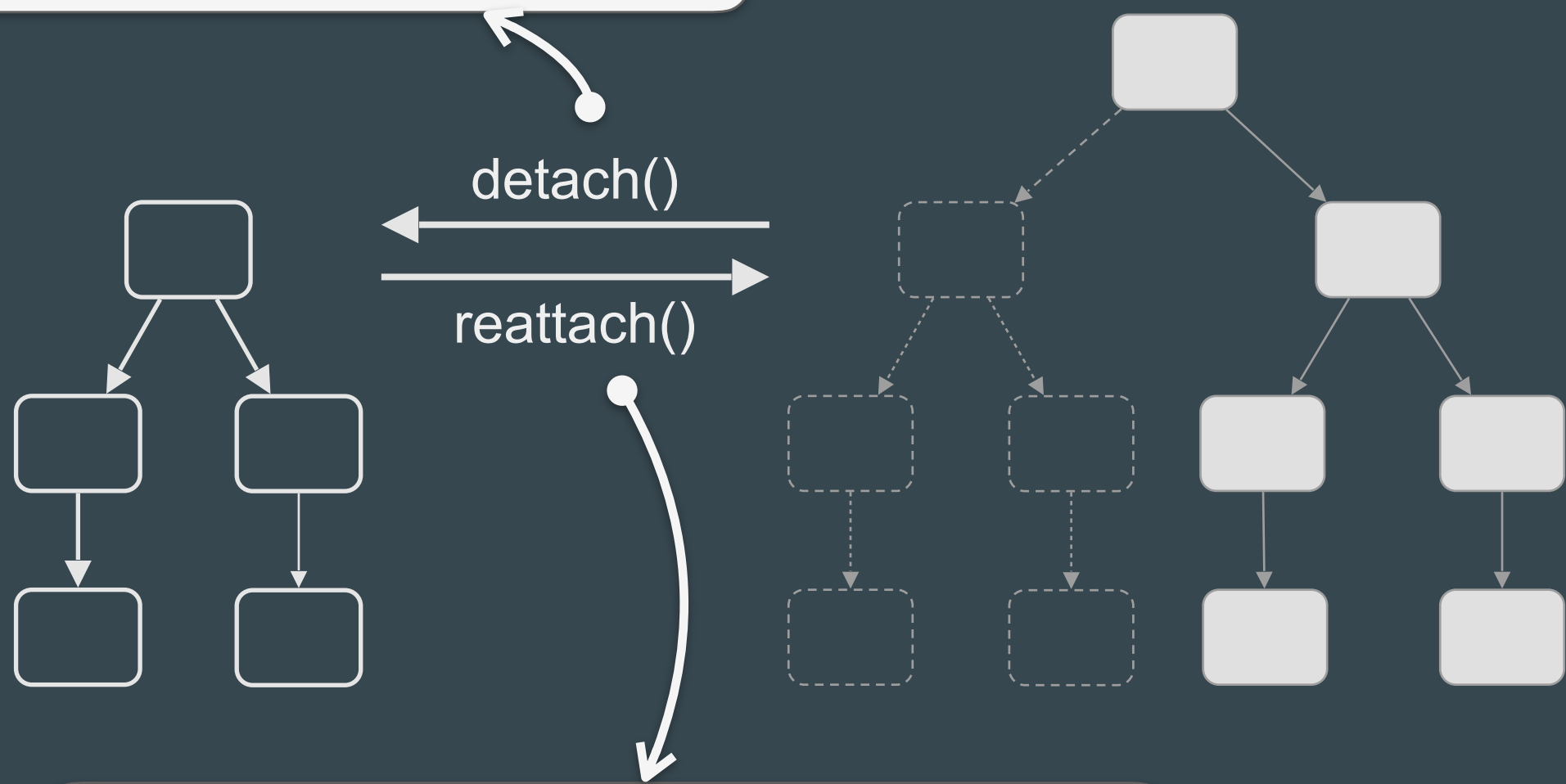
```
view.state |= ViewState.ChecksEnable
```

```
export class BranchComponent {  
  curValue: string;  
  constructor(  
    private cd: ChangeDetectorRef  
  ) { }  
  @Input() stream: Observable<string>;  
  ngOnInit() {  
    this.stream.subscribe(value => {  
      this.curValue = value;  
      this.cd.markForCheck();  
    })  
  }  
}
```



- ✓ **detach**: 把该组件及其所有子组件从组件树的变化检测体系剥离出来
- ✓ **reattach**: 把剥离出来的组件分支重新接入组件树的变化检测体系

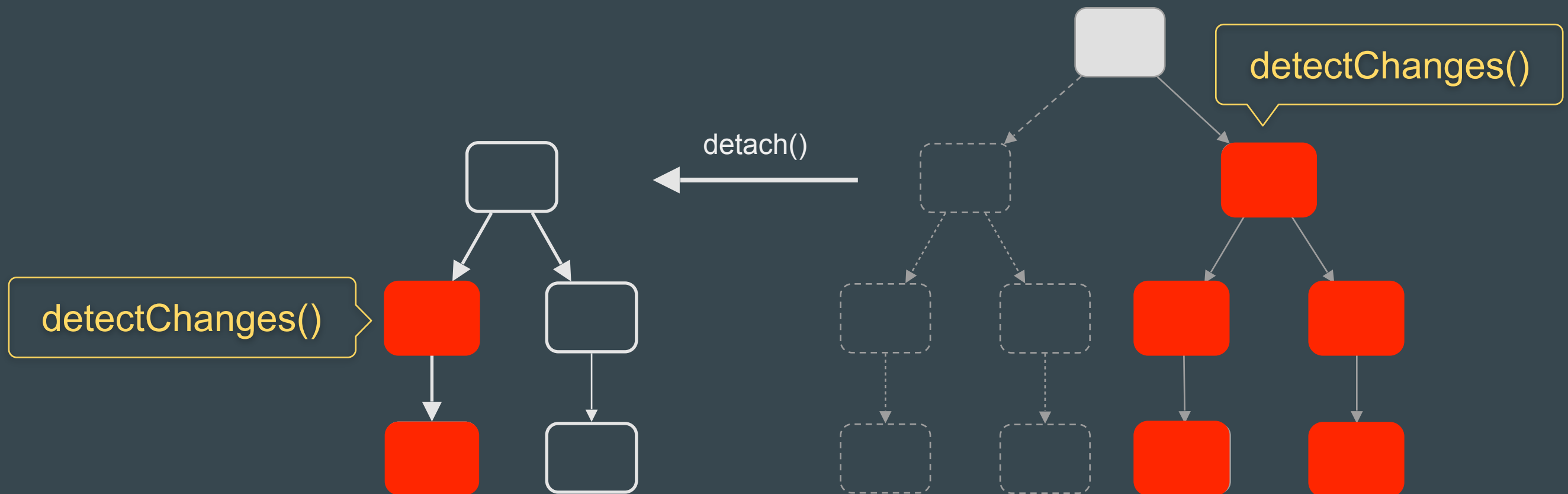
`view.state &= ~ViewState.ChecksEnable`



`view.state |= ViewState.ChecksEnable`


```
export class MenuComponent() {  
    @Input() isHide: boolean = false;  
    constructor( private cd: ChangeDetectorRef ) { }  
    ngOnChanges( changes: SimpleChanges ) {  
        if ( 'isHide' in changes ) {  
            if ( changes.isHide.currentValue ) {  
                // 隐藏时  
  
                this.cd.detach();  
            } else {  
                // 显示时  
  
                this.cd.reattach();  
            }  
        }  
    }  
}
```

✓ detectChanges: 手动触发该组件及其所有子组件的变化检测



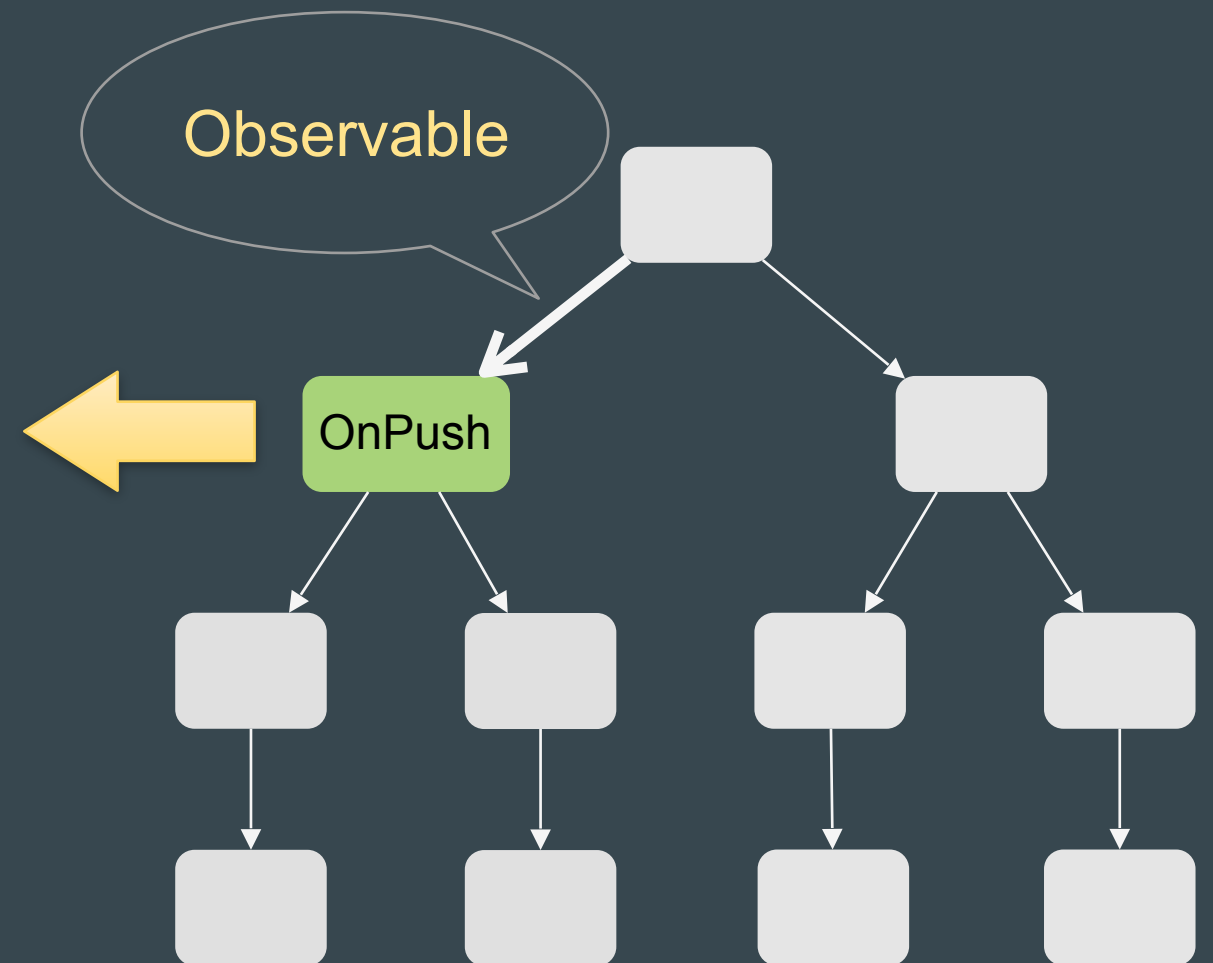
`$scope.$digest()`

```

export class BranchComponent {
  curValue: string;
  constructor(
    private cd: ChangeDetectorRef
  ) { }
  @Input() stream: Observable<string>;
  ngOnInit() {
    this.stream.subscribe(value => {
      this.curValue = value;
      this.cd.markForCheck();
    })
  }
}

```

手动触发一次变化更新



其他性能优化点

enableProdMode()

✓ 可关闭运行时的特殊检查

```
import { enableProdMode } from '@angular/core';

if (ENV === 'production') {
  enableProdMode();
}
```

ngFor-trackBy

✓ 减少元素的销毁重建

```
// template
```

```
<li *ngFor="let item of items; let i = index; trackBy: trackByFn">...</li>
```

```
// class
```

```
class AppComponent {  
  trackByFn(idx, item) {  
    return item.id;  
  }  
}
```

< ng-container />

✓ 避免DOM嵌套过深

```
<ng-container>  
  <h1>GF</h1>  
  <span>I'm in ng container</span>  
</ng-container>
```



```
<h1>GF</h1>  
<span>I'm in ng container</span>
```

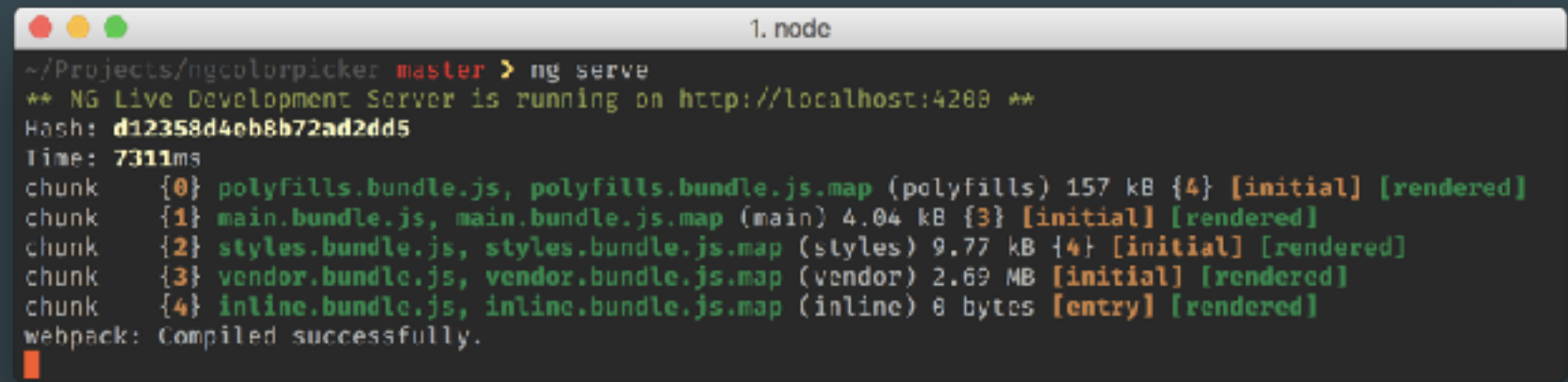
(二) 工具与工程化

Angular CLI

1. auto generate

2. webpack

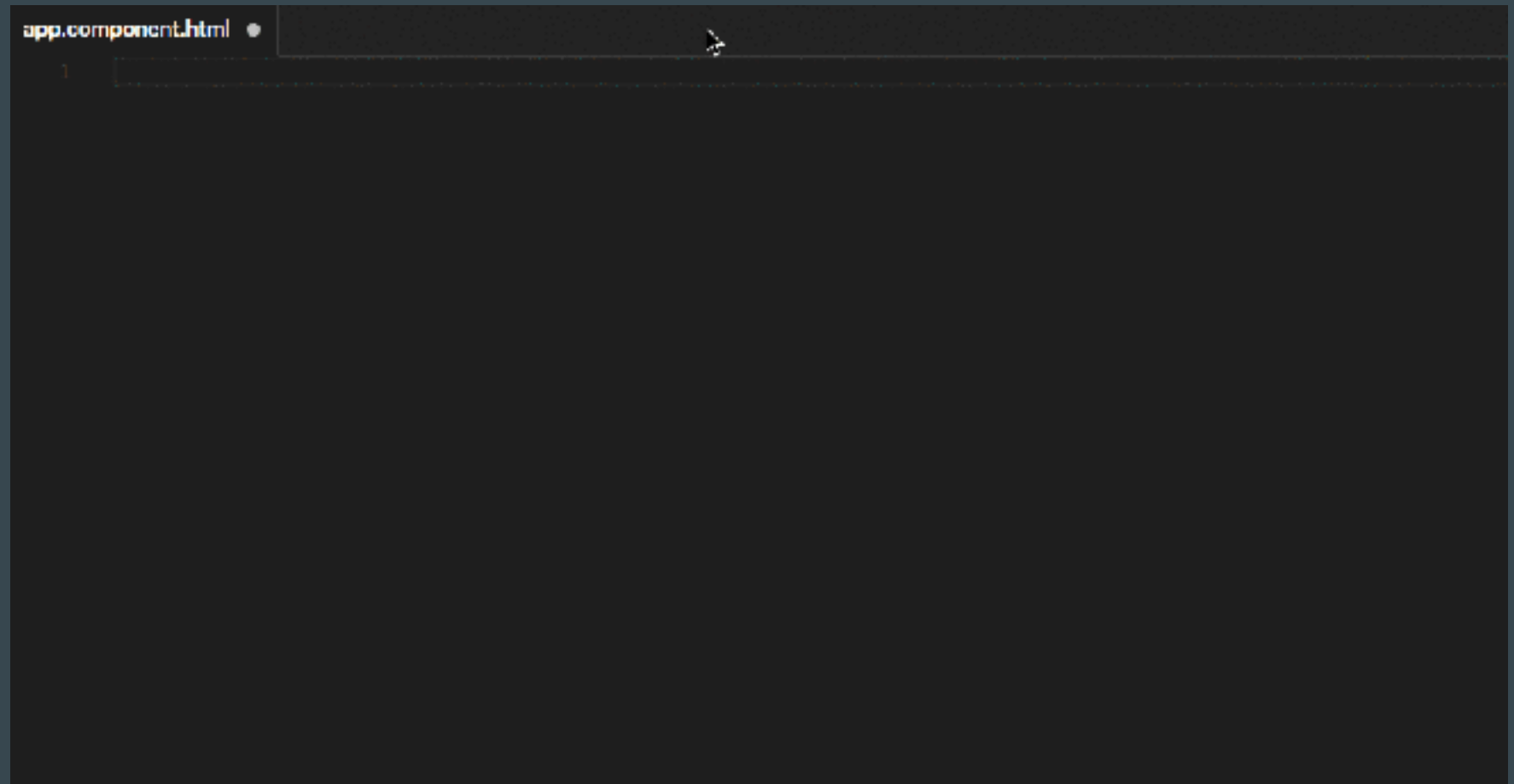
3. ngtools



```
1. node
~/Projects/ngcolorpicker master > ng serve
** NG Live Development Server is running on http://localhost:4200 **
Hash: d12358d4eb8b72ad2dd5
Time: 7311ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4} [initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 4.04 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 9.77 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.69 MB [initial] [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 8 bytes [entry] [rendered]
webpack: Compiled successfully.
```

Angular Language Service

- Completions lists
- AoT Diagnostic messages
- Quick info
- Go to definition



其他工具

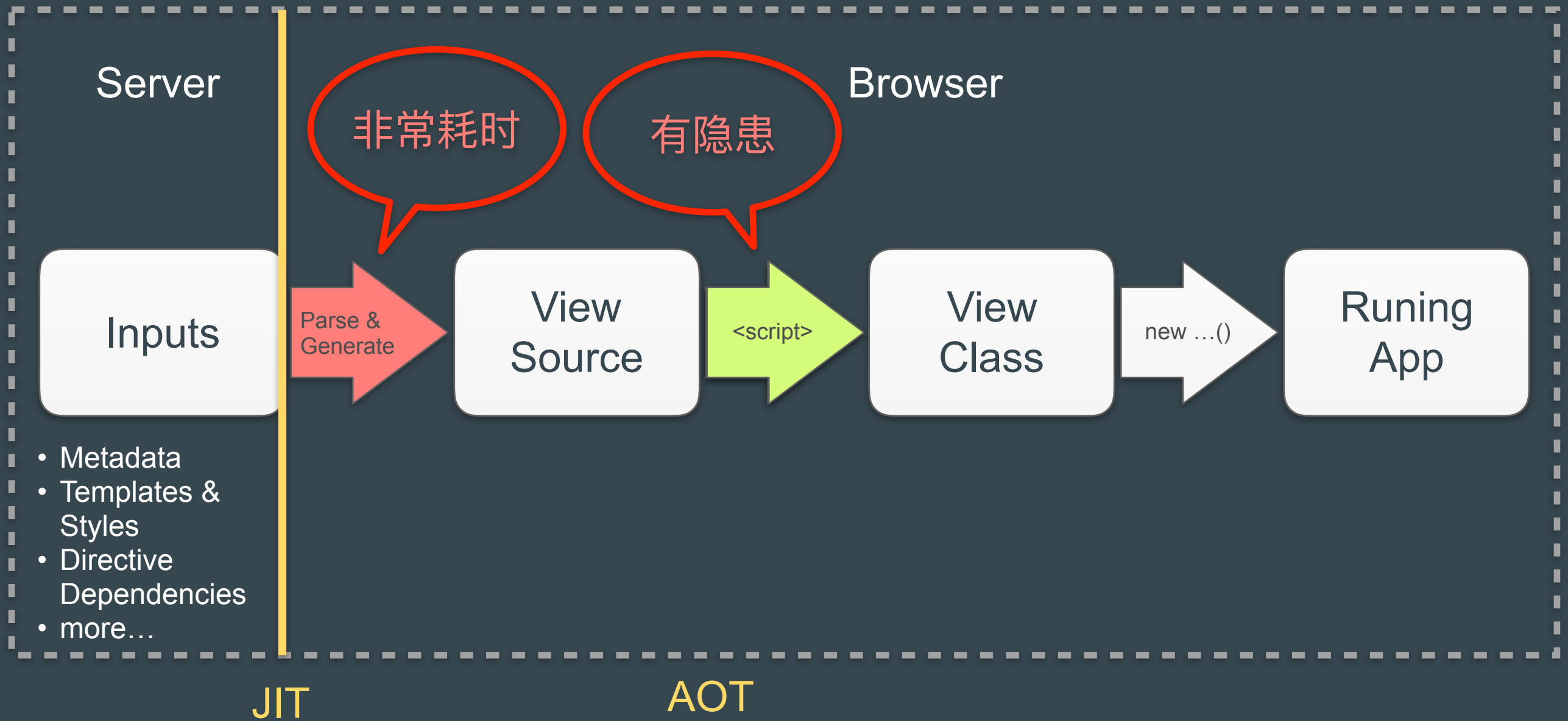
- Codelyzer
- Augury
- Compodoc & ngd
- Snippets
- Ngrev

更好玩的

- ngworld

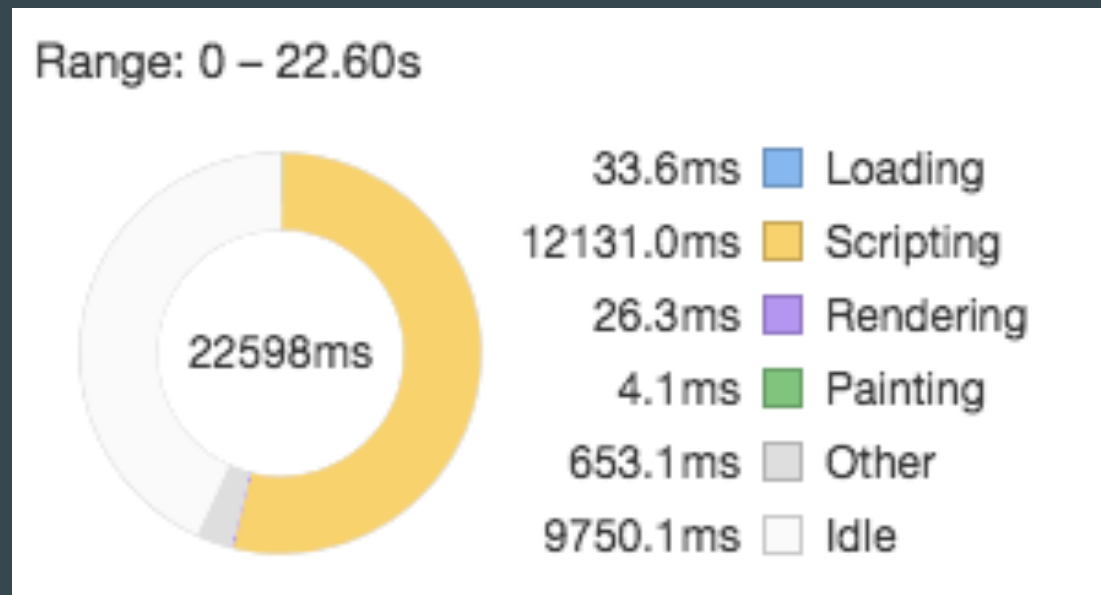
构建

AoT



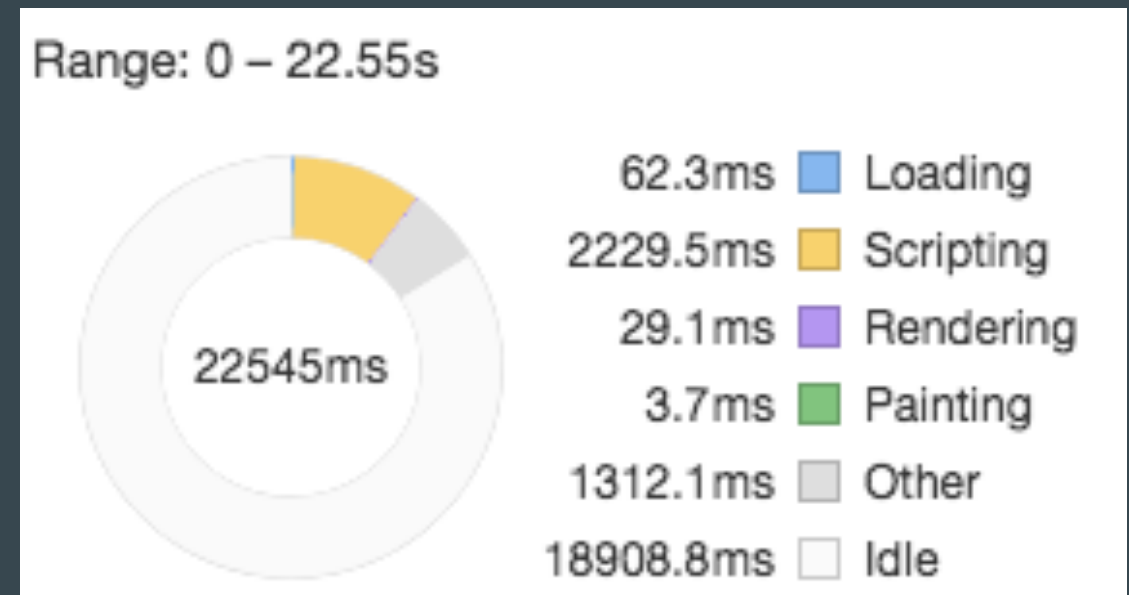
首屏Scripting指标

JiT



≈12s

AoT



≈2s

AoT构建

✓ ngc 命令

✓ @ngtools/webpack

1. Angular CLI:

ng build --aot

2. webpack:

```
const { AotPlugin } = require('@ngtools/webpack');
module.exports = {
  // ...
  module: {
    rules: [
      // ...
      // 1. 添加必要的loader
      {
        "test": /\.html$/,
        "loader": "raw-loader"
      },
      {
        "test": /\.ts$/,
        "loader": "@ngtools/webpack"
      }
    ]
  },
  plugins: [
    // ...
    // 2. 在插件栏添加 AotPlugin 实例
    new AotPlugin({
      "mainPath": "main.ts",
      "tsConfigPath": "tsconfig.json"
    })
  ]
}
```

AOT vs JIT

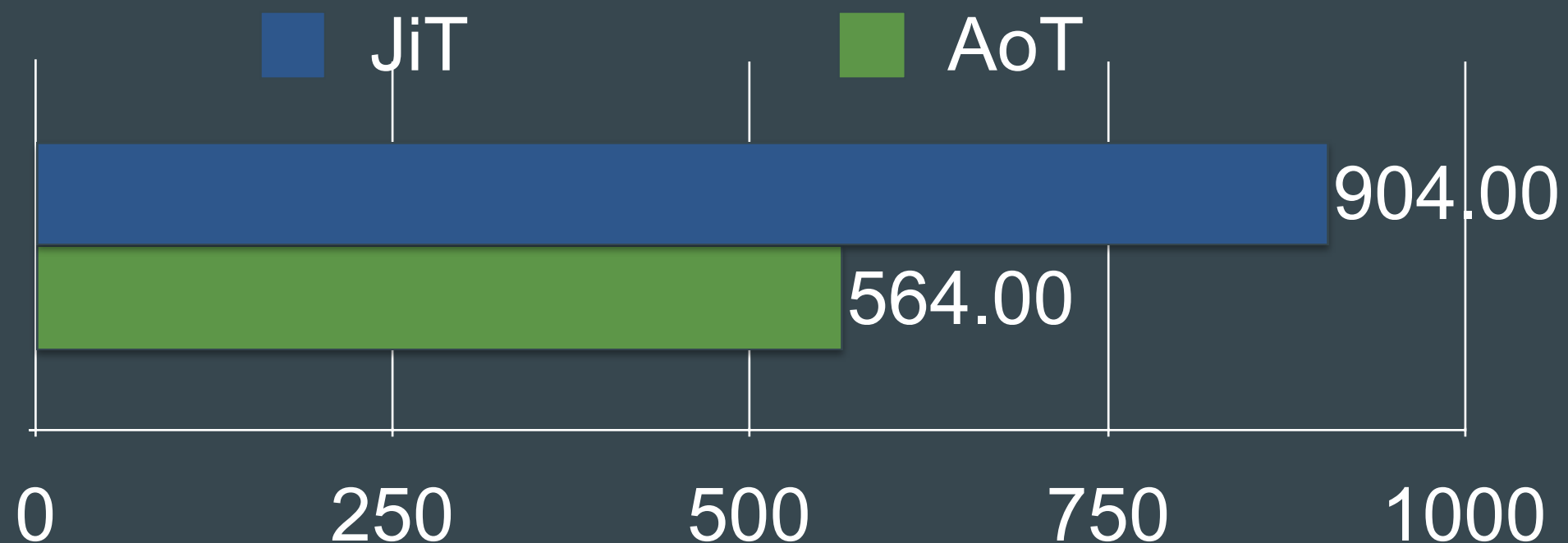
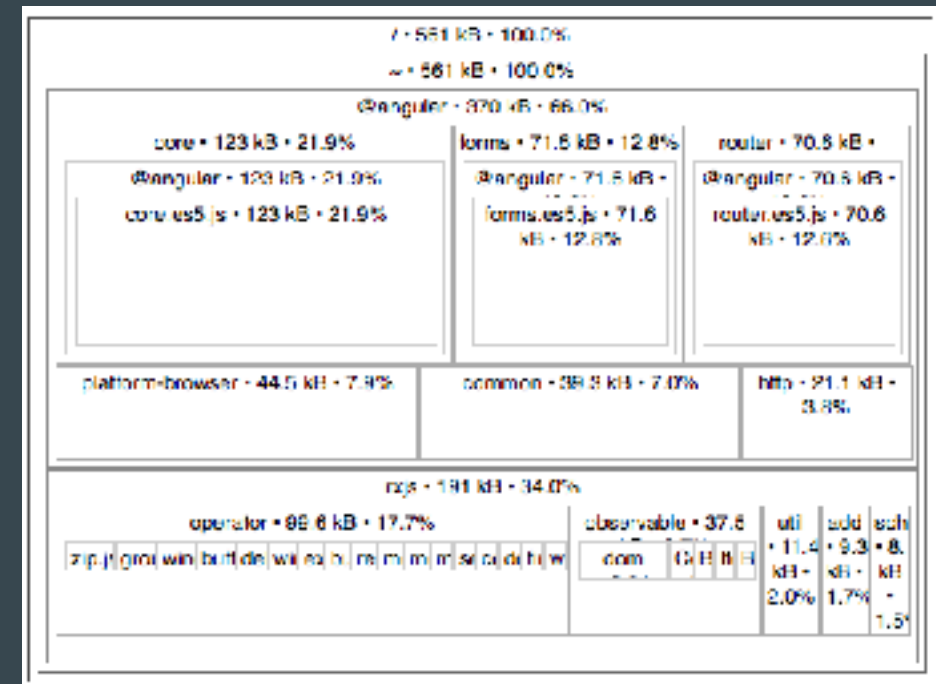
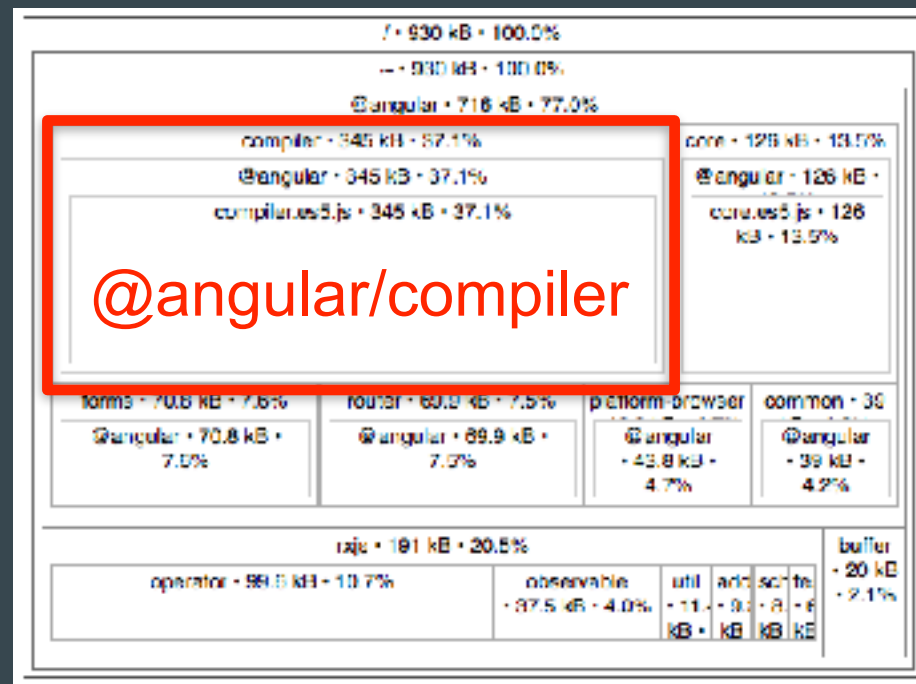
1.5K	Jun	7	16:17	dist/inline.0c78aaa422d913680ae2.bundle.js
1.1M	Jun	7	16:17	dist/main.1b103fa4436dc404.bundle.js
92K	Jun	7	16:17	dist/polyfill.e6bfff948a1e6d9adad.bundle.js
109K	Jun	7	16:17	dist/scripts.152047673bd29fd5e.bundle.js
904K	Jun	7	16:17	dist/vendor.670498ea7c73e897d75a.bundle.js

JiT

1.5K	Jun	7	16:26	dist/inline.7359b583f4299410e979.bundle.js
1.4M	Jun	7	16:26	dist/main.45675b1b74216f303.bundle.js
92K	Jun	7	16:26	dist/polyfill.e6bfff948a1e6d9adad.bundle.js
109K	Jun	7	16:26	dist/scripts.152047673bd29fd5e.bundle.js
564K	Jun	7	16:26	dist/vendor.02abd91054bc216b76cd.bundle.js

AoT

vendor.js



main.js

```
<!-- toy.component.html -->
<div
  [attr.dataId]=data.id
  [style.left]="data.x + 'px'"
  [style.top]="data.y + 'px'"
>
</div>
```



View_ToyComponent_0()

View_ToyComponent_Host_0()

ToyComponentNgFactory()

```
function View_ToyComponent_0(l) {
  return import1.evid(0, [
    (l)(), import1.eeld(0, null, null, 1, 'div', [], [
      [
        1,      Binding
        'dataId',
        0
      ],
      [
        4,
        'left',
        null
      ],
      [
        4,
        'top',
        null
      ]
    ]), null, null, null, null)),
    (l)(), import1.eted(null, ['\n'])),
    (l)(), import1.eted(null, ['\n'])),
    ], null, function (ck, v) {
      var co = v.component;
      var currVal_0 = co.data.id;
      var currVal_1 = (co.data.x + 'px');
      var currVal_2 = (co.data.y + 'px');
      ck(v, 0, 0, currVal_0, currVal_1, currVal_2);
    });
}
```

数据比较

AoT-friendly

```
@Component({  
  template: `  
    <cmp [name]="name"></cmp>  
  `,  
  // ...  
})
```

```
// Don't  
export class NameComponent {  
  @Input() private name: string;  
};  
  
// Do  
export class NameComponent {  
  /** @internal */  
  @Input() name: string;  
};
```

Test	AoT With ngc	AoT With @ngtools/webpack	JIT
control	✓	✓	✓
form-control	✓	✓	✓
func-in-string-ctrlg	✓	✓	✓
jquery	✓	✓	✓
template-variable	✓	✓	✓
template-expression	✓	✓	✓
mut-property-decorator	✓	✗	✓
namut-property-decorator	✓	✗	✓
angular-redux-store	✓	✓	✓
ngrx	✓	✓	✓
ngrx-compose	✓	✓	✓
arrow-function-exports	✗	✗	✓
default-exports	✗	✗	✓
form-control-error	✗	✗	✓
func-as-variable-export	✗	✗	✓
func-declaration-export	✓	✓	✓
func-in-declarations	✗	✗	✓
func-in-providers	✗	✗	✓
func-in-providers-useFactory	✗	✗	✓
func-in-providers-useValue	✗	✗	✓
func-in-routes	✗	✗	✓
interpolated-es6	✗	✗	✓

<https://github.com/rangle/angular-2-aot-sandbox>

Lazy Load

太大了!

```
1.5K Jun 7 16:26 dist/inline.7359b583f4299410e979.bundle.js
1.4M Jun 7 16:26 dist/main.4fbe8675b1b74216f303.bundle.js
92K Jun 7 16:26 dist/polyfills.81e6bfff948a1e6d9adad.bundle.js
109K Jun 7 16:26 dist/scripts.800152047673bd29fd5e.bundle.js
564K Jun 7 16:26 dist/vendor.02abd91054bc216b76cd.bundle.js
```

Lazy Load

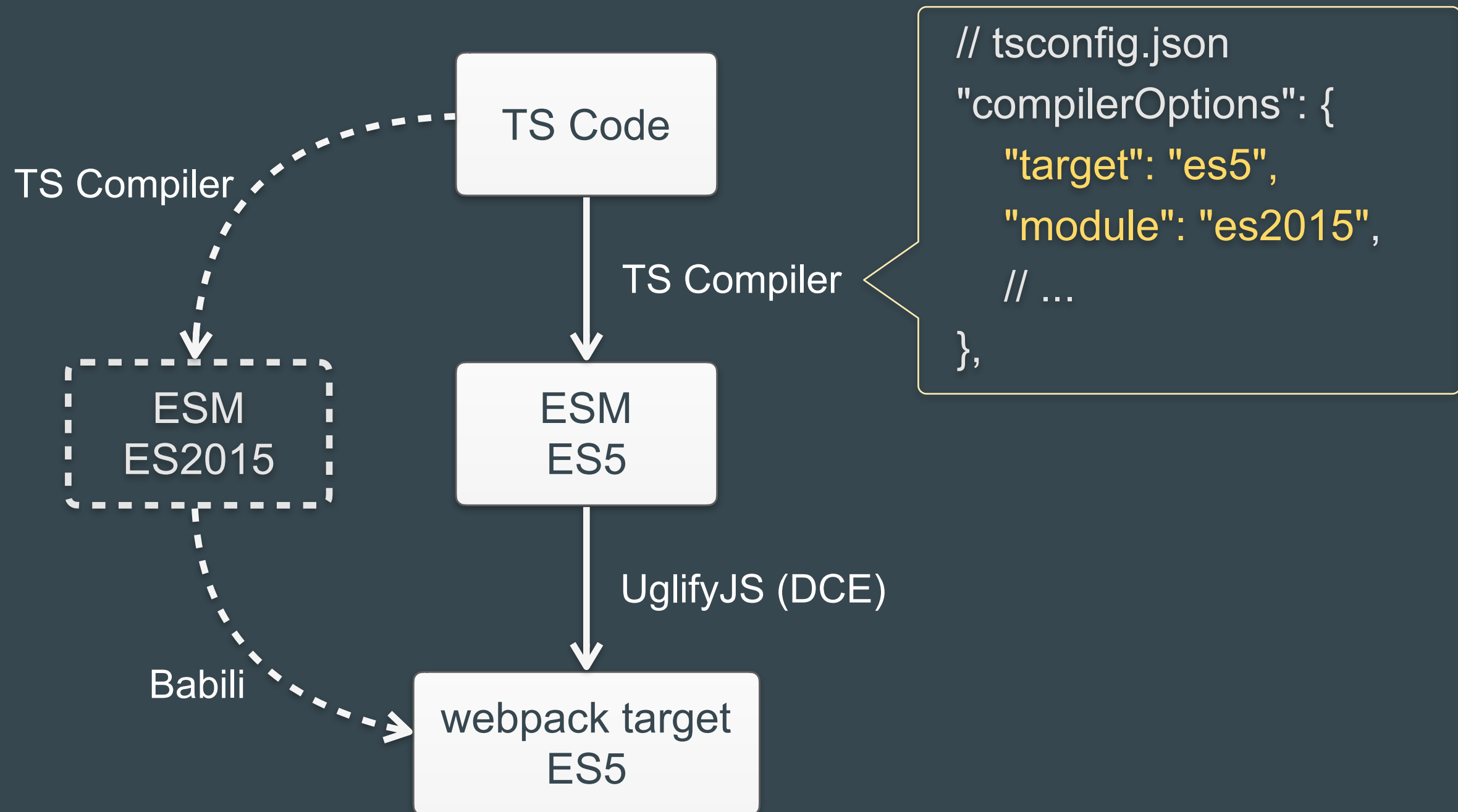
黑魔法?

```
{  
  path: 'lazy-path',  
  loadChildren: 'path/to/lazy.module#LazyModule'  
}
```

#模块类名

```
593K Jun 7 16:31 dist/0.dcdf06fae3350088cf77.chunk.js  
709K Jun 7 16:31 dist/1.de1c0ea71dabc83d5962.chunk.js  
1.5K Jun 7 16:31 dist/inline.dd19266503b163ec790f.bundle.js  
161K Jun 7 16:31 dist/main.376f835d6cfd7b4e2a76.bundle.js  
92K Jun 7 16:31 dist/polyfills.849365358db48be666ab.bundle.js  
109K Jun 7 16:31 dist/scripts.bd4a33734d5be9a548d0.bundle.js  
564K Jun 7 16:31 dist/vendor.274456b18e6f37bcfd2b.bundle.js
```

Tree Shaking (webpack2)





- ✓ 尽量使用带ES Module版本的第三方库
- ✓ 如果不支持Tree Shaking，尽量使用小粒度导入

小粒度导入

// Bad

```
import 'rxjs';  
import { Observable } from 'rxjs';  
import { Subject } from 'rxjs/Rx';
```

// Good

```
import { Observable } from 'rxjs/Observable';  
import { Subject } from 'rxjs/Observer';  
import 'rxjs/add/operator/switchMap.js';
```

593K	Jun	7	16:31	dist/0.dcdf06fae3350088cf77.chunk.js	593K	Jun	7	16:37	dist/0.dcdf06fae3350088cf77.chunk.js
709K	Jun	7	16:31	dist/1.de1c0ea71dabc83d5962.chunk.js	709K	Jun	7	16:37	dist/1.de1c0ea71dabc83d5962.chunk.js
1.5K	Jun	7	16:31	dist/inline.62bb17d84f3d912dddf1.bundle.js	1.5K	Jun	7	16:37	dist/inline.62bb17d84f3d912dddf1.bundle.js
161K	Jun	7	16:31	dist/main.1d8da44e5fb39ff7e631.bundle.js	161K	Jun	7	16:37	dist/main.1d8da44e5fb39ff7e631.bundle.js
92K	Jun	7	16:31	dist/polyfills.849365358db48be666ab.bundle.js	92K	Jun	7	16:37	dist/polyfills.849365358db48be666ab.bundle.js
109K	Jun	7	16:31	dist/scripts.bd4a33734d5be9a548d0.bundle.js	109K	Jun	7	16:37	dist/scripts.bd4a33734d5be9a548d0.bundle.js
564K	Jun	7	16:31	dist/vendor.471563162b63087ba997.bundle.js	422K	Jun	7	16:37	dist/vendor.471563162b63087ba997.bundle.js

(三) 常见问题解决

3.1 依赖导入

Error: Can't bind to 'x' since it isn't a known property of 'y'

✓ imports: [XModule]

✓ declarations: [XComponent]

3.2 CDN

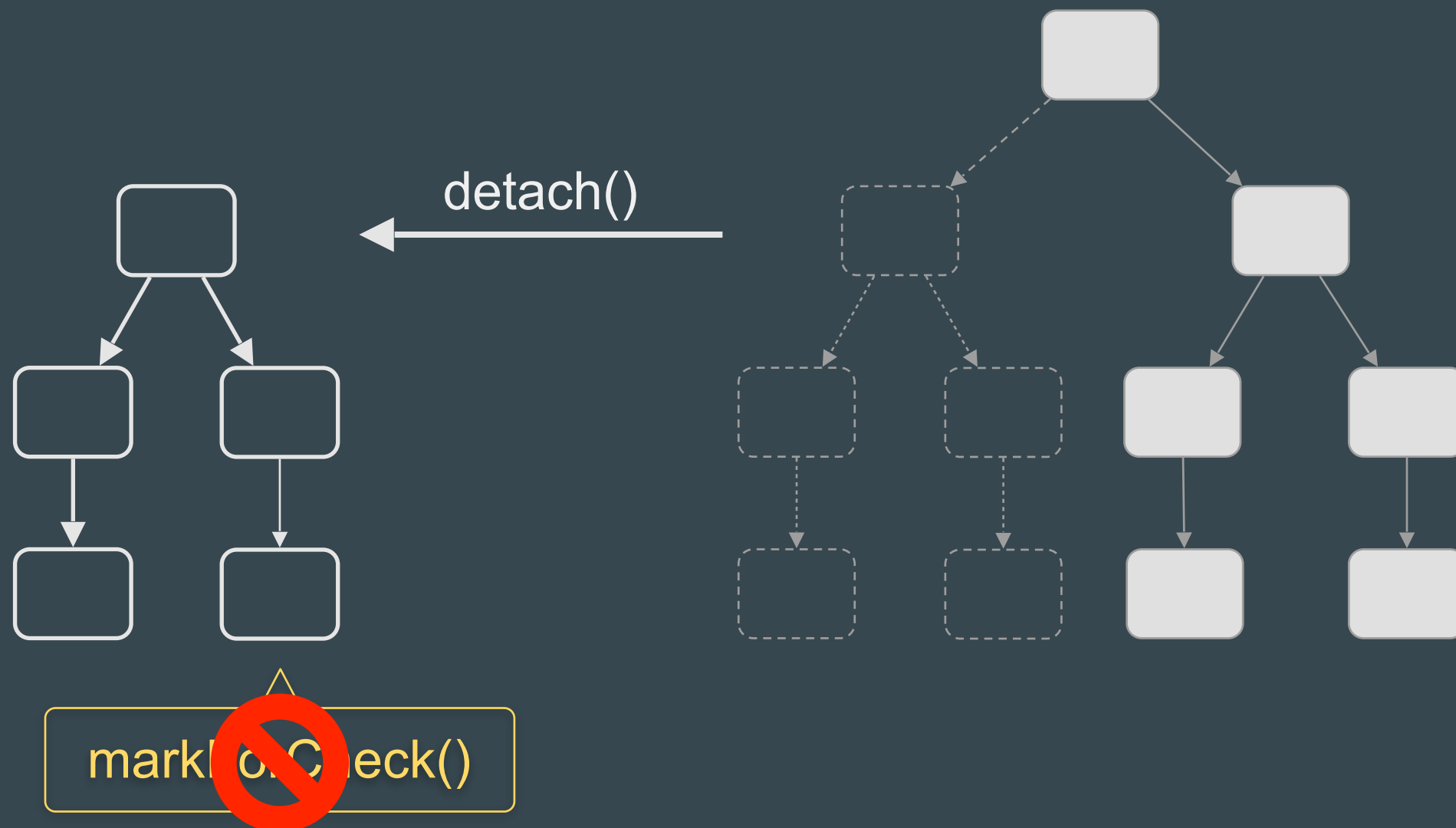
使用的 Angular CLI 提供的 deployUrl 功能

```
ng build --deploy-url=https://cdn.gfzq.cn/pah/
```

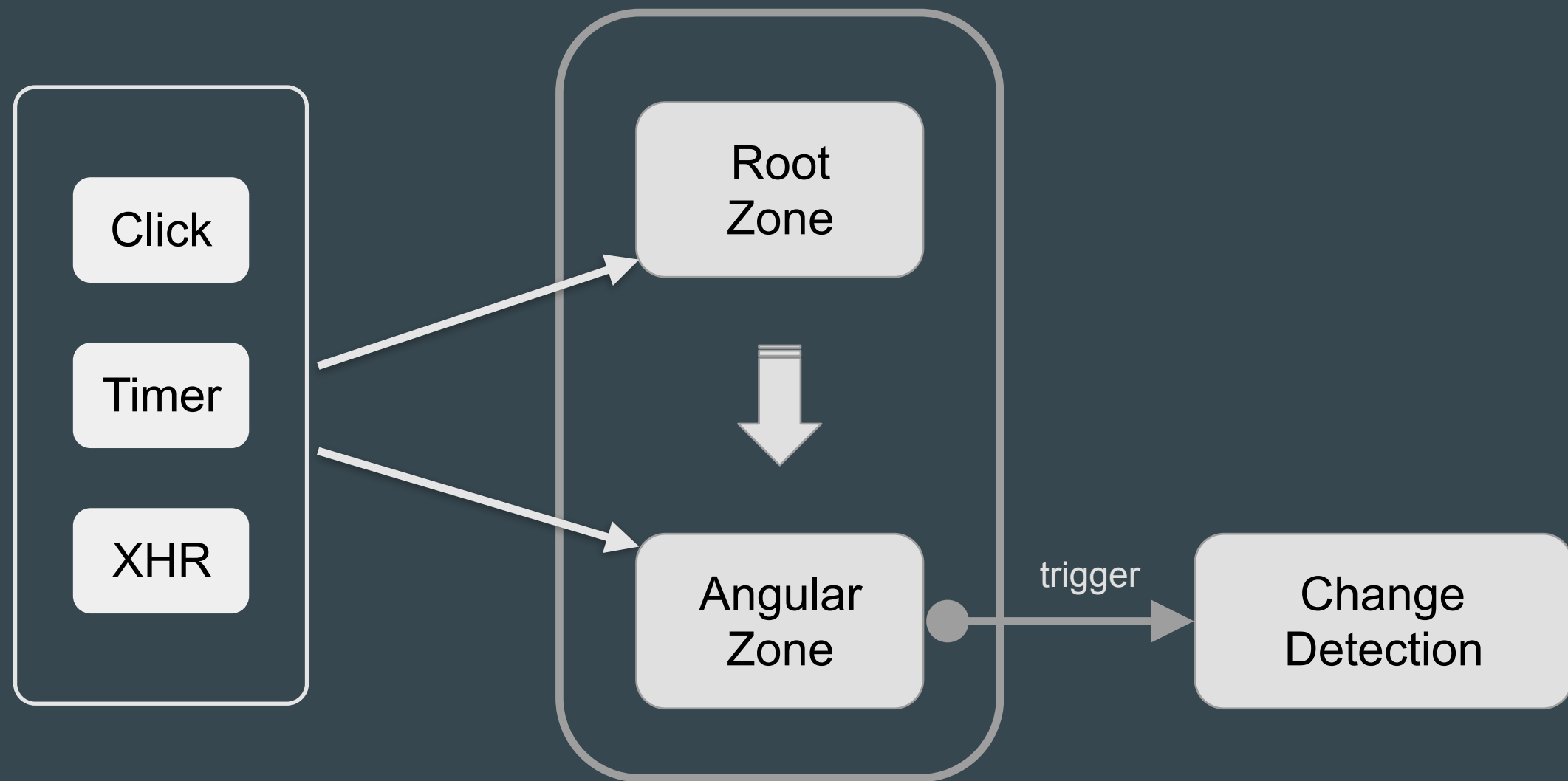
注意：

1. 必须以 ``/`` 结尾
2. 暂不支持省略协议头，如 ``//cdn.gfzq.cn/pah/``

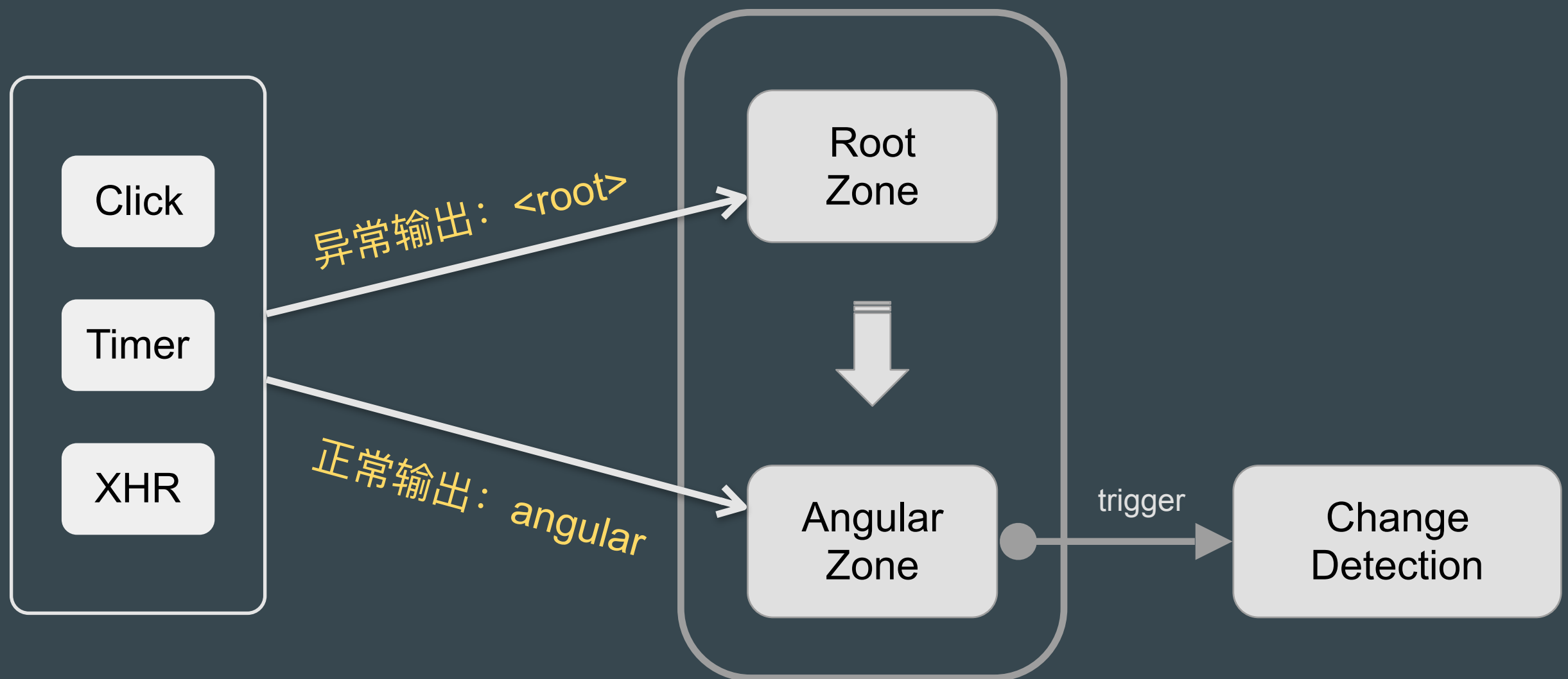
3.3 detach & markForCheck



3.4 视图更新失效



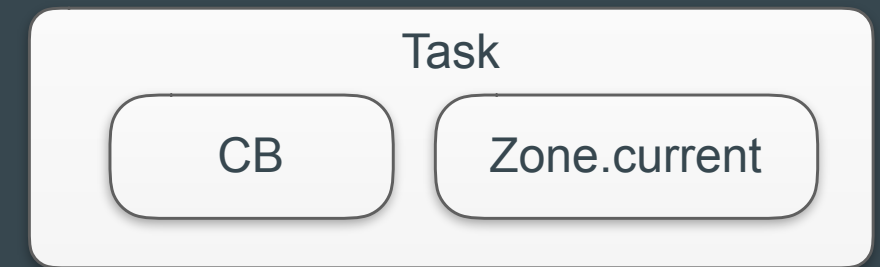
```
function asyncFunc() {  
    console.log(Zone.current.name);  
}
```



JSBridge

```
JSB.call( name, params, successCB, errorCallback )
```

```
// angular zone
function btnClick() {
  console.log(Zone.current.name);
  callJSBridgeFunction(() => {
    console.log(Zone.current.name);
    setTimeout(() => {
      console.log(Zone.current.name);
    });
  });
}
```

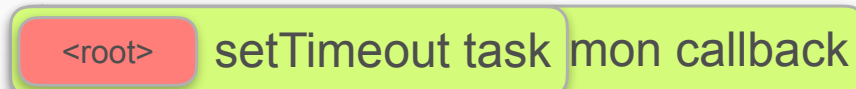


```
Zone.current = task.zone;
try {
  invokeCBWithZone();
}
finally {
  Zone.current = task.zone.parent;
}
```

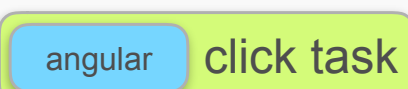
current Zone



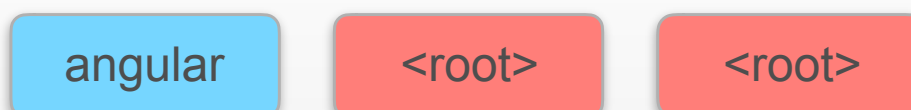
Tasks



Stack



Log



解决方法

```
// angular zone
console.log(Zone.current.name);
JSBridgeAsyncMethod(() => {
  console.log(Zone.current.name);
  setTimeout(() => {
    console.log(Zone.current.name);
    this.cd.detectChanges();
  });
  this.cd.detectChanges();
});
```

推荐解决方法

```
// angular zone
console.log(Zone.current.name);
JSBridgeAsyncMethod(() => {
  this.ngZone.run(() => {
    console.log(Zone.current.name);
    setTimeout(() => {
      console.log(Zone.current.name);
    });
  });
});
```

`$scope.$apply()`

谢谢大家

Q&A

lzhspace@gmail.com