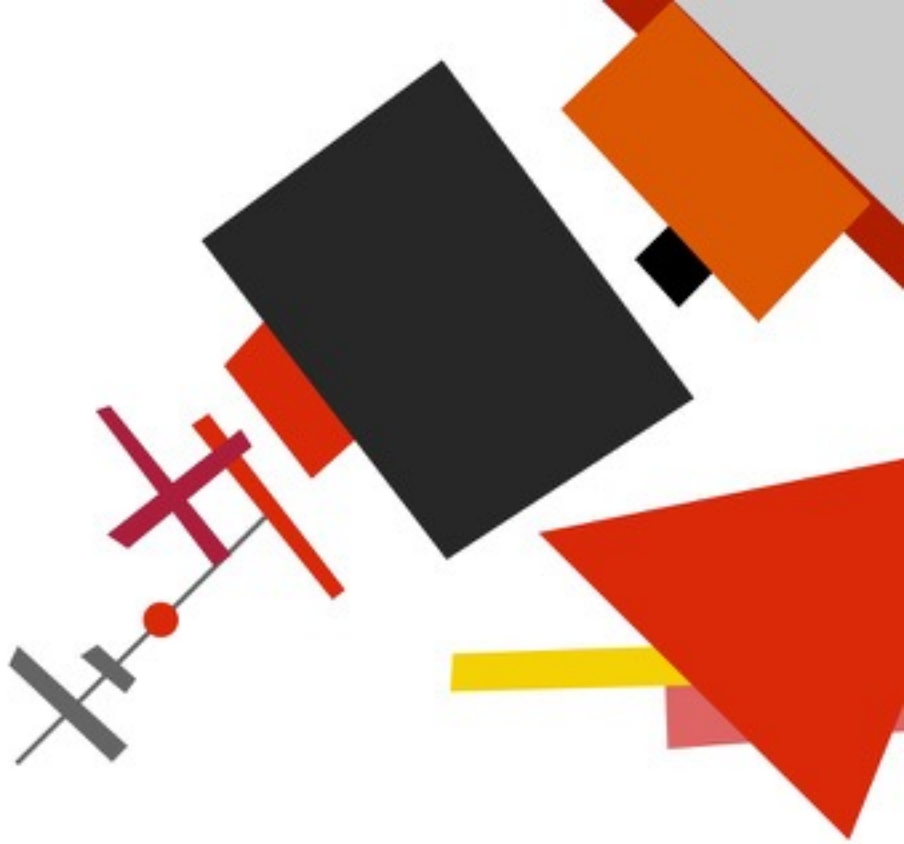
Превышаем скоростные лимиты с Angular 2

Алексей Охрименко - IPONWEB



HighLoad++

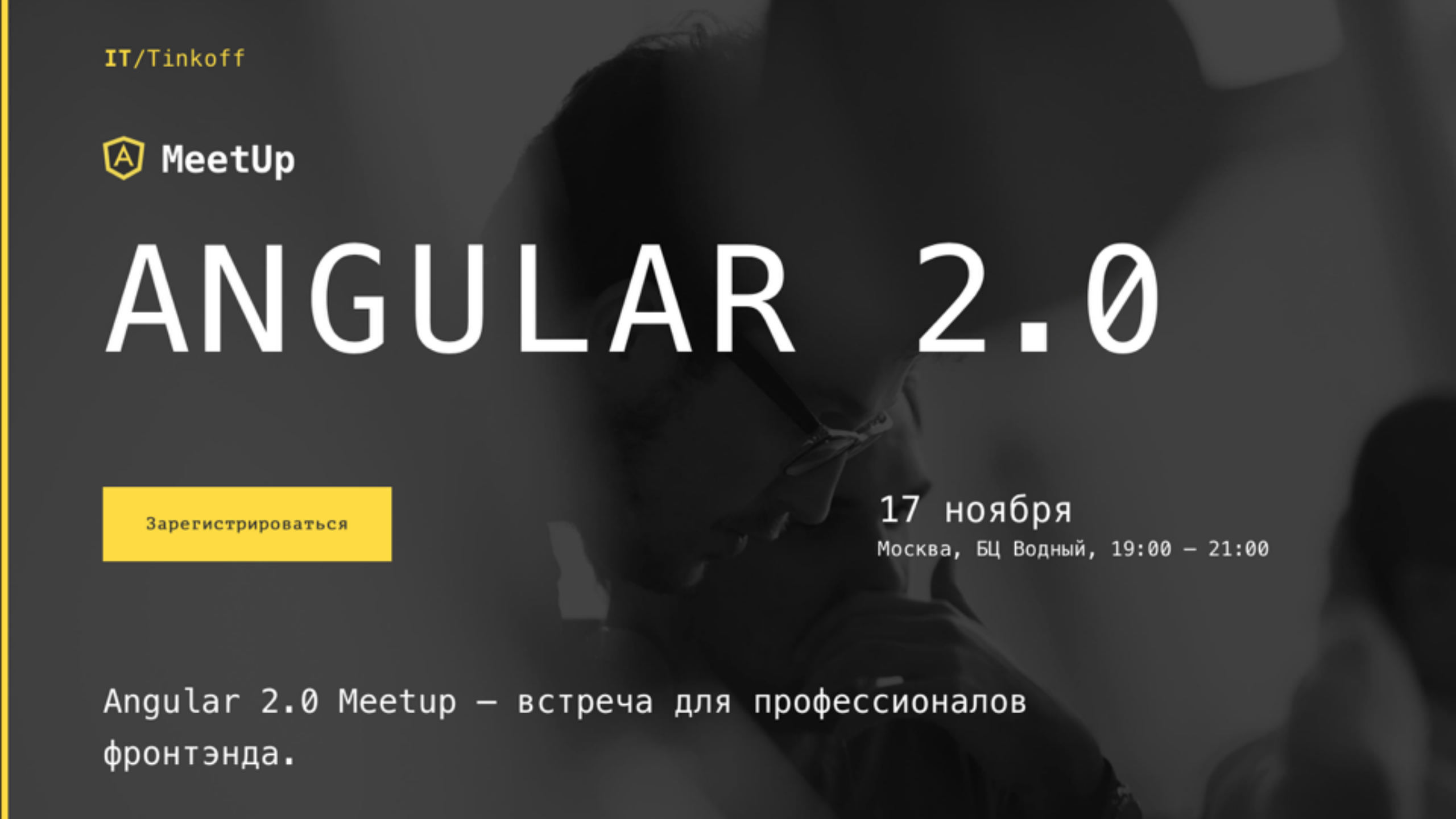
Профессиональная конференция разработчиков высоконагруженных систем

Алексей Охрименко

Twitter: @Ai_boy

IPONWEB


WINTER IS COMING


IT/Tinkoff
MeetUp
ANGULAR 2.0
Зарегистрироваться
17 ноября
Москва, БЦ Водный, 19:00 – 21:00
Angular 2.0 Meetup – встреча для профессионалов фронтэнда.

IPONWEB

RTB

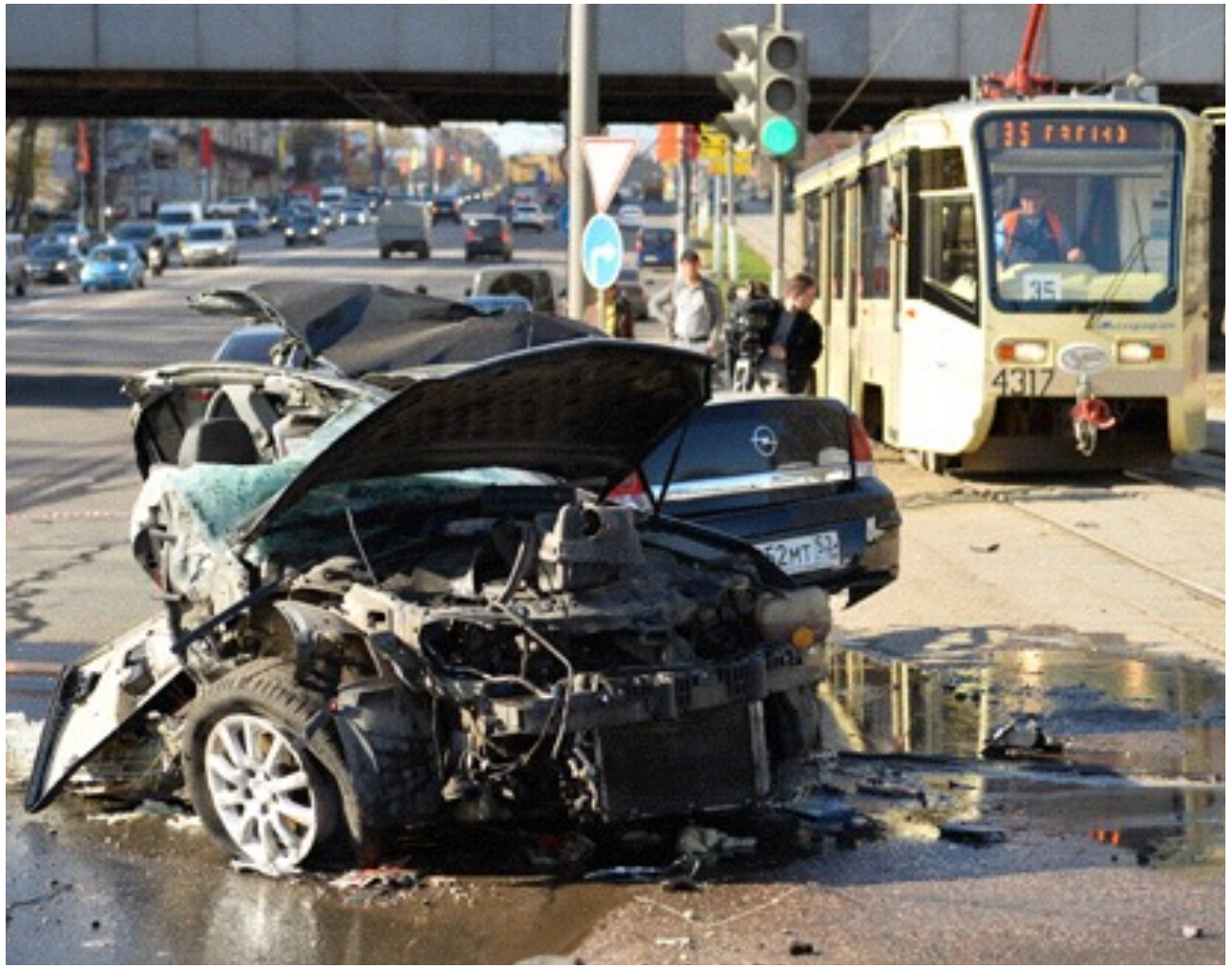
DSP

SSP


SPOILER
ALERT!

последствия превышения скорости (в реальной жизни)

последствия превышения скорости*



* - поищите в Google Image - 4-ый результат

Всегда успеете…

А где тогда скорость превышать?

Angular 2, Angular 2… нас и [НАШ_FRAMEWORK] неплохо кормит


20
30
40
50

А что значит «скорость»?

Скорость загрузки

Скорость загрузки

Размер

Скорость загрузки

Размер

LazyLoading

Скорость загрузки

Размер

LazyLoading

Скорость работы

Скорость загрузки

Размер

LazyLoading

Скорость работы

Обьем работы

Скорость загрузки

Размер

LazyLoading

Скорость работы

Обьем работы

Производительность

Скорость загрузки

Размер

LazyLoading

Скорость работы

Обьем работы

Производительность

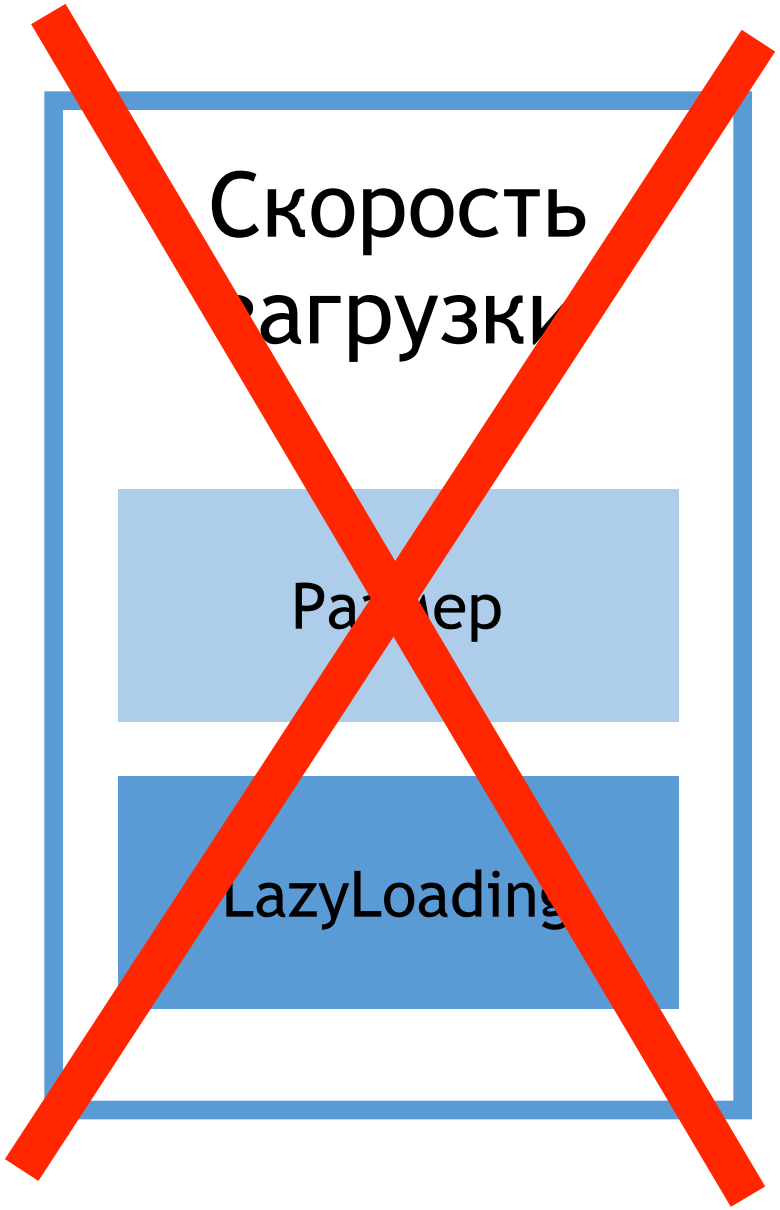
Память

Скорость загрузки

- Размер
- LazyLoading

Скорость работы

- Обьем работы
- Производительность
- Многопоточность
- Память



Скорость работы

- Обьем работы
- Производительность
- Многопоточность
- Память

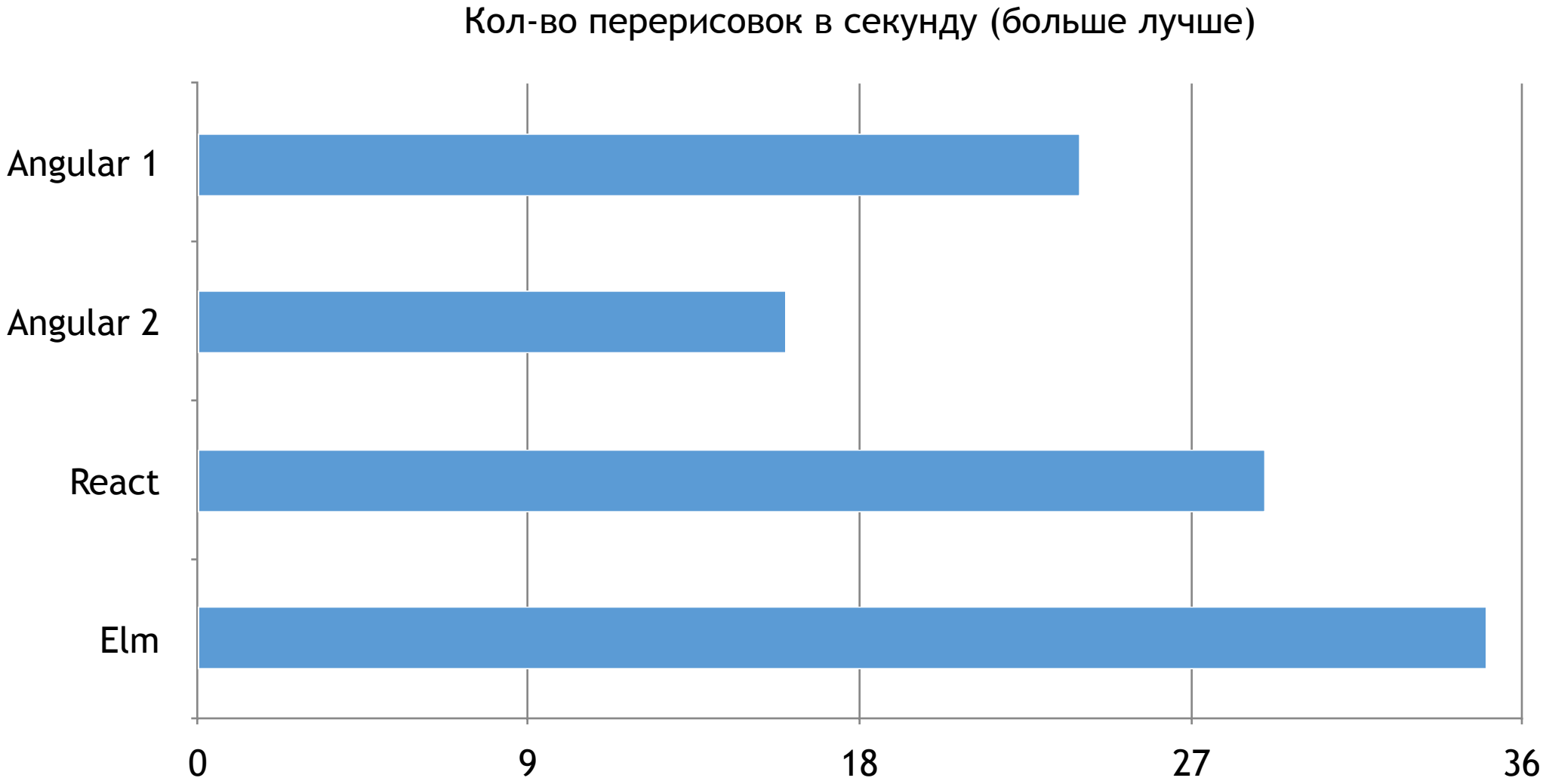
Скорость работы

https://github.com/krausest/js-framework-benchmark

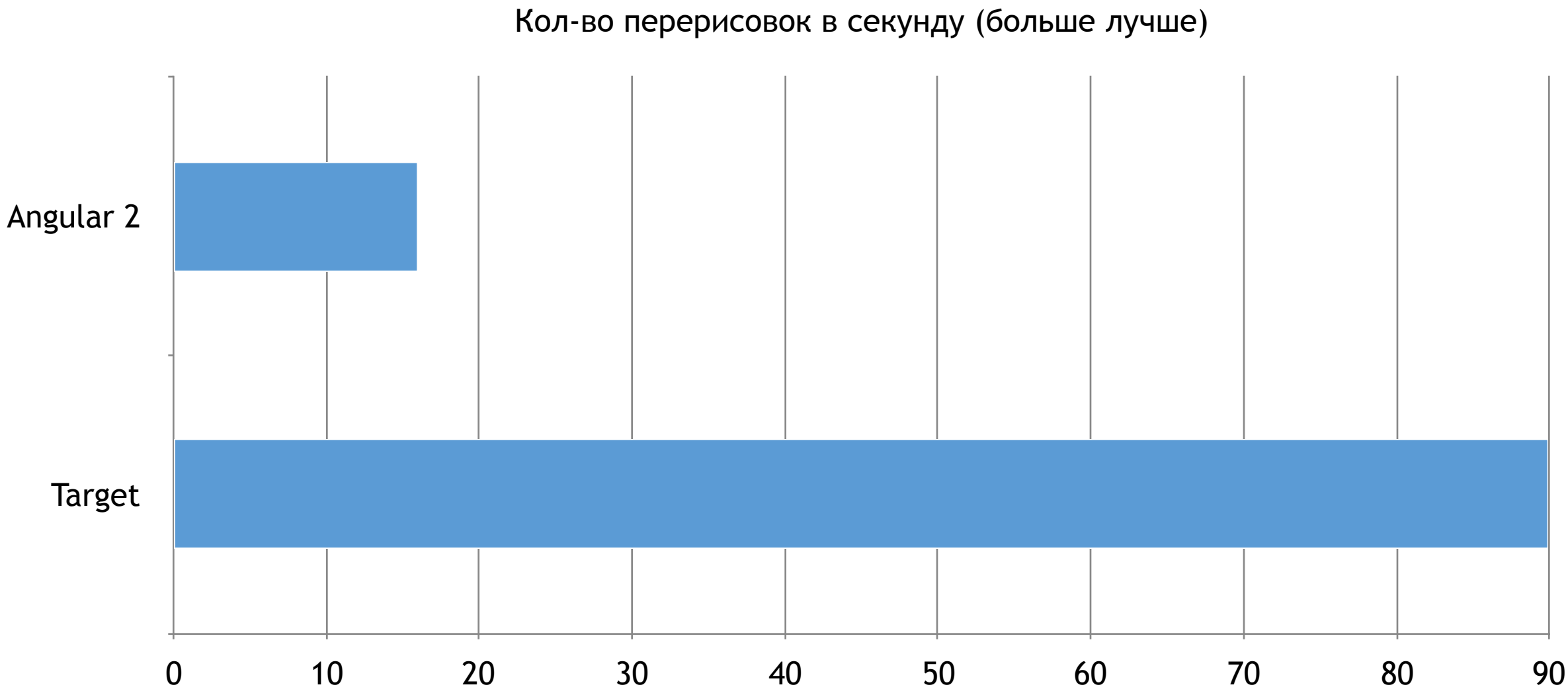
https://github.com/mathieuancelin/js-repaint-perfs

cluster1 slave	5	1.25	3.06
cluster2	5	13.49	14.00
cluster2 slave	5	12.14	1.79
cluster3	7	8.38	3.78
cluster3 slave	9	7.12	13.05
cluster4	1	2.66	6.49
cluster4 slave	5	6.94	8.46
cluster5	8	10.55	2.38
cluster5 slave		5.08	1.77
cluster6		9.70	12.44
cluster6 slave	3	3.30	4.32
cluster7	2	3.63	14.32
cluster7 slave	5	9.15	12.95
cluster8	2	3.35	7.28
cluster8 slave	10	8.95	4.04
cluster9	8	5.15	2.32
cluster9 slave	10	4.32	3.39
cluster10	6	1.85	7.85

SELECT blah FROM something



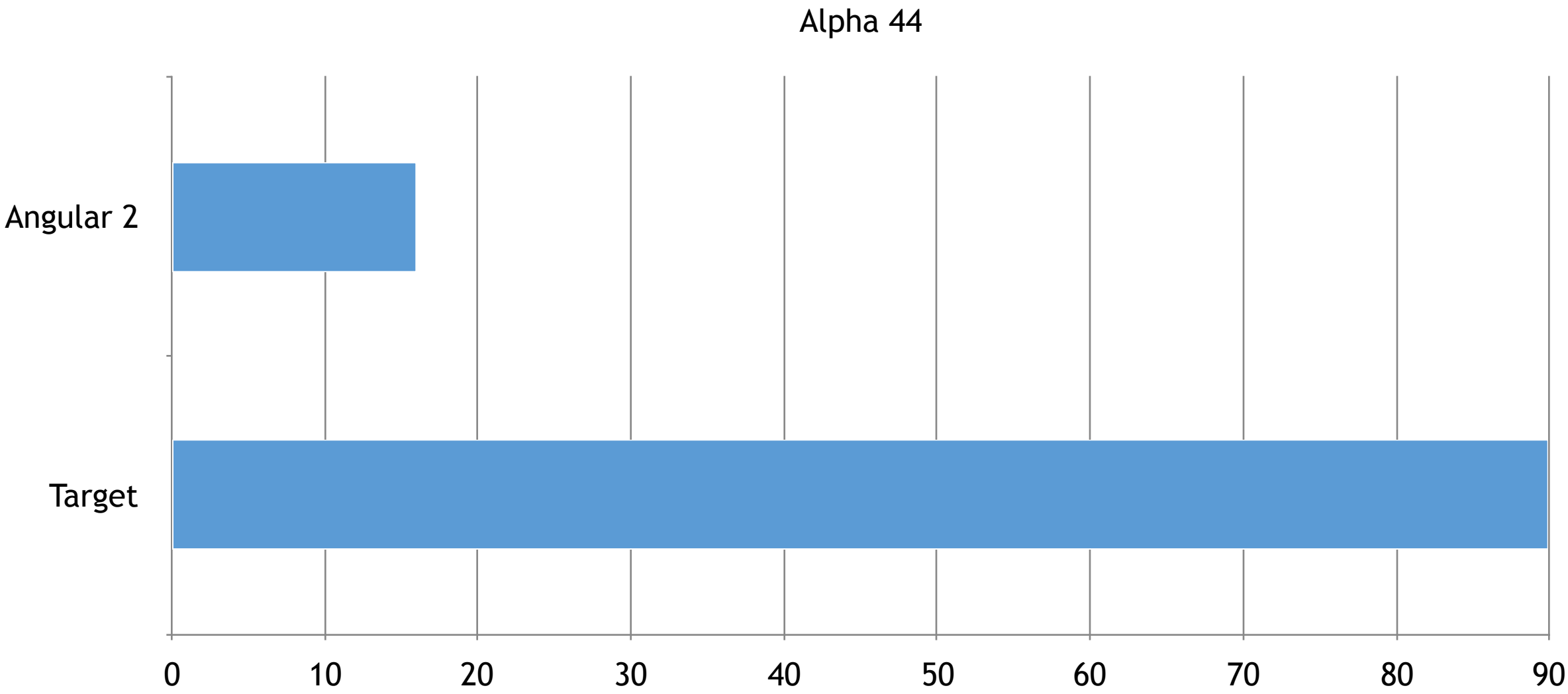
Наша цель … 90 RR

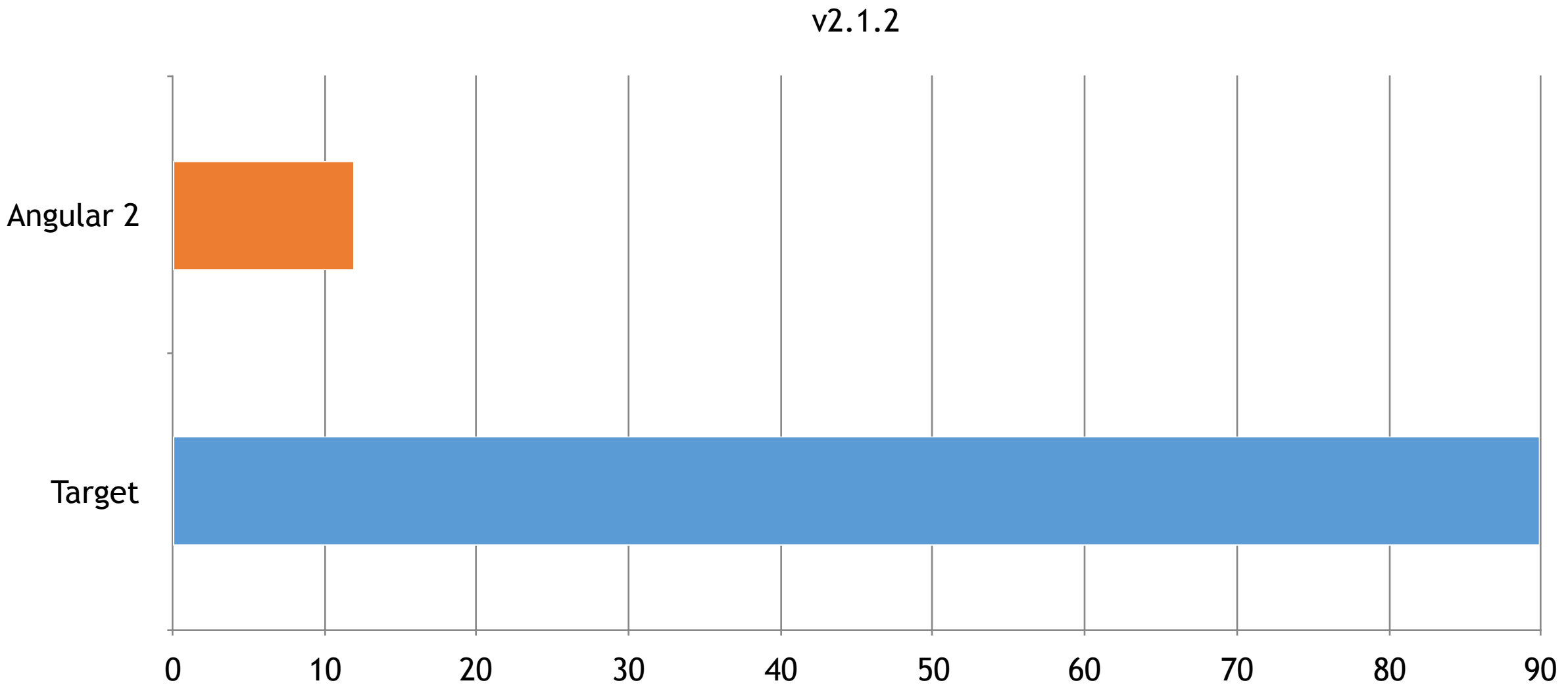

Кол-во перерисовок в секунду (больше лучше)
Angular 2
Target
0
10
20
30
40
50
60
70
80
90

Старая версия Angular 2

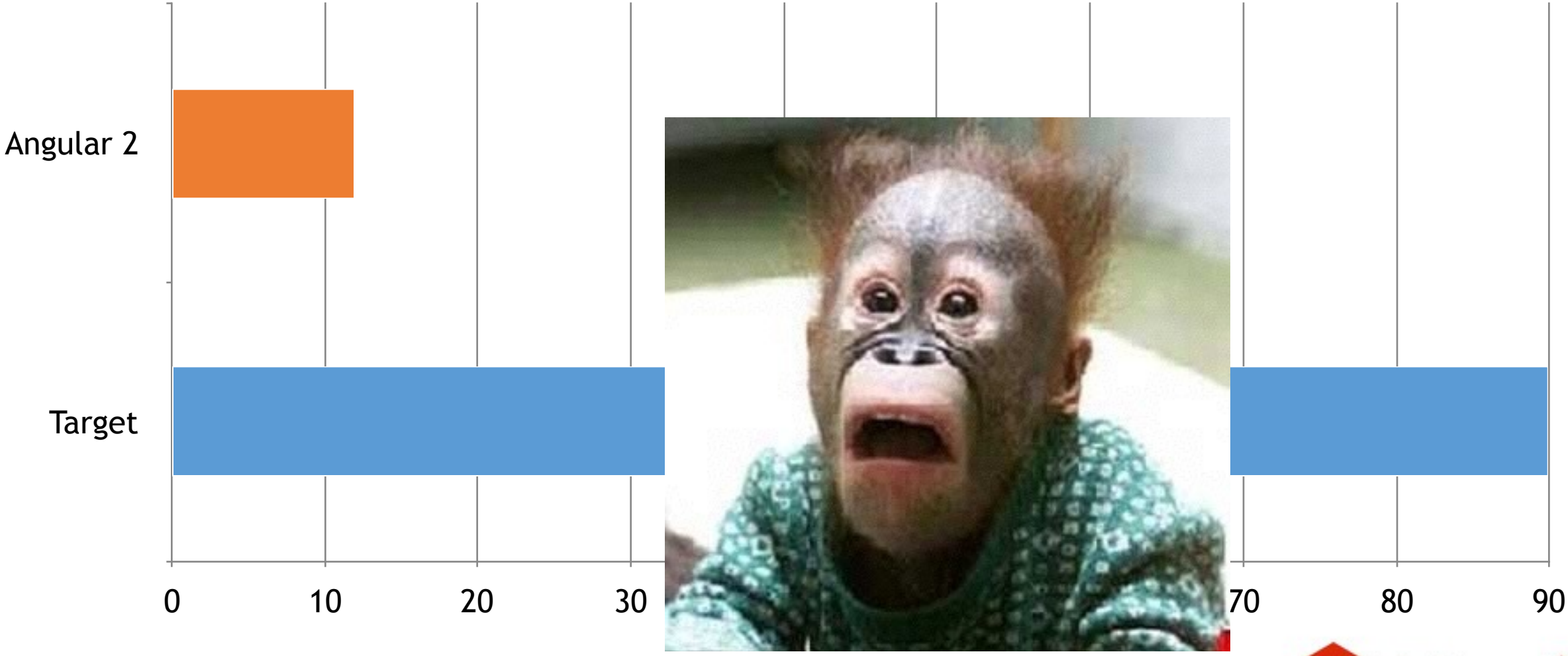
Alpha 44

Alpha 44 —> v2.1.2


Alpha 44
Angular 2
Target
0
10
20
30
40
50
60
70
80
90

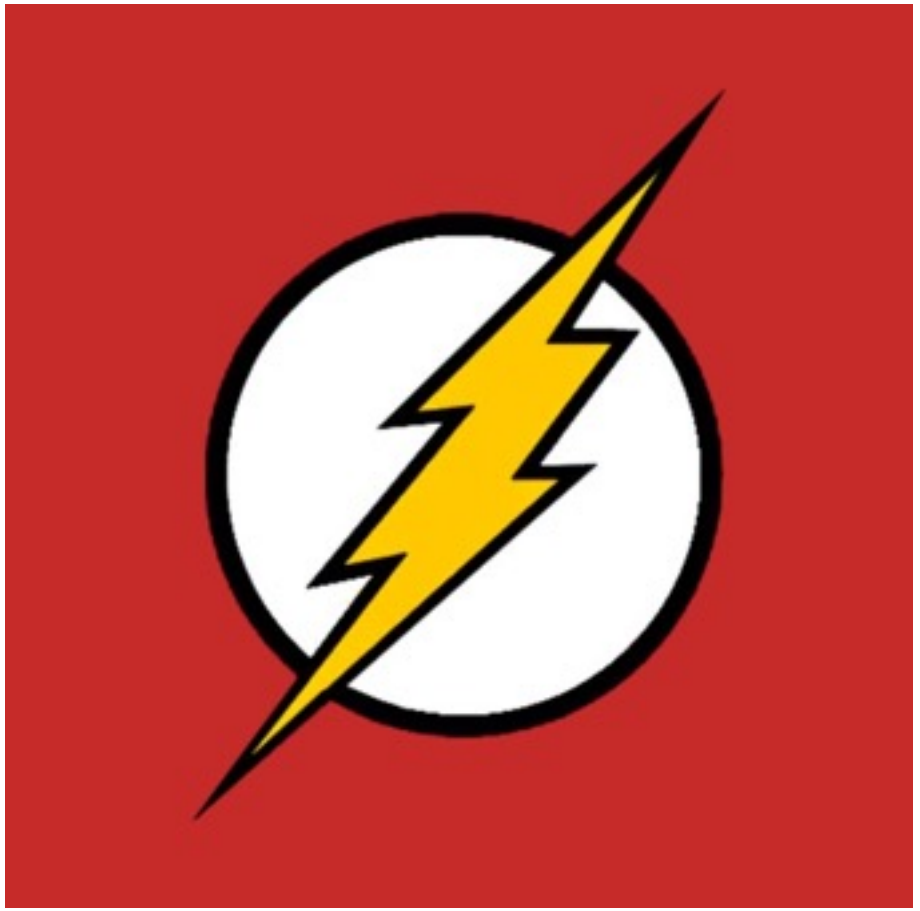

v2.1.2
Angular 2
Target
0
10
20
30
40
50
60
70
80
90

v2.1.2



На самом деле все просто…

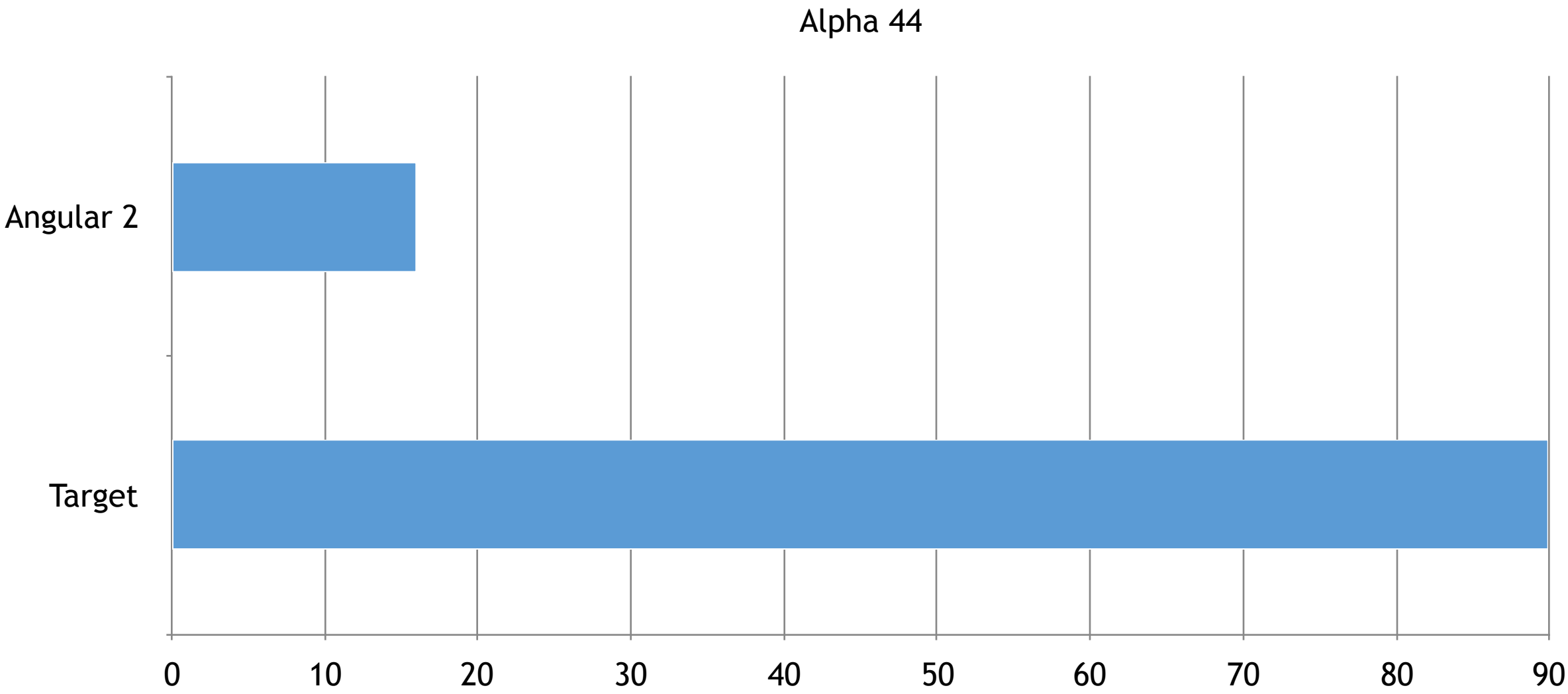
Angular 2 Performance Checklist

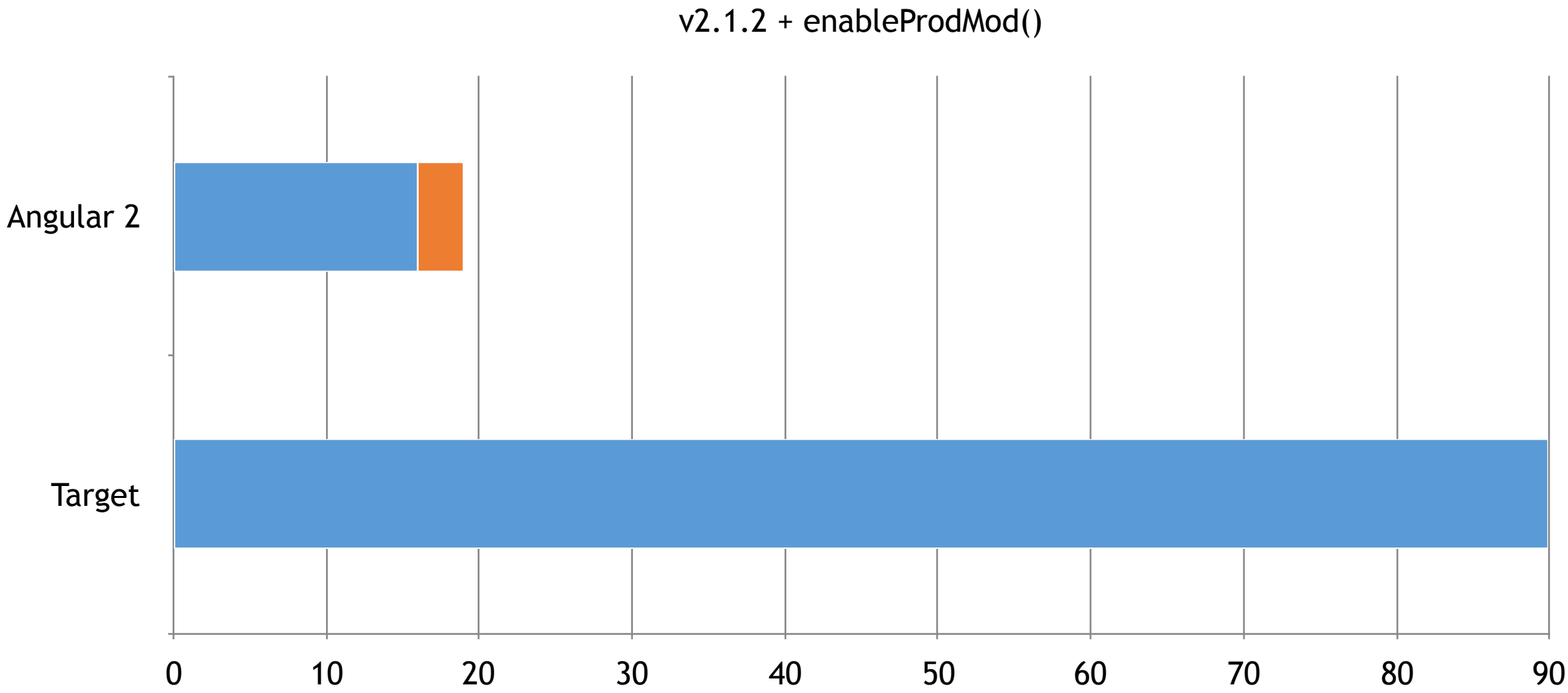


```
import {enableProdMode} from '@angular/core';

enableProdMode();
```

300% в Edge

enableProdMode()

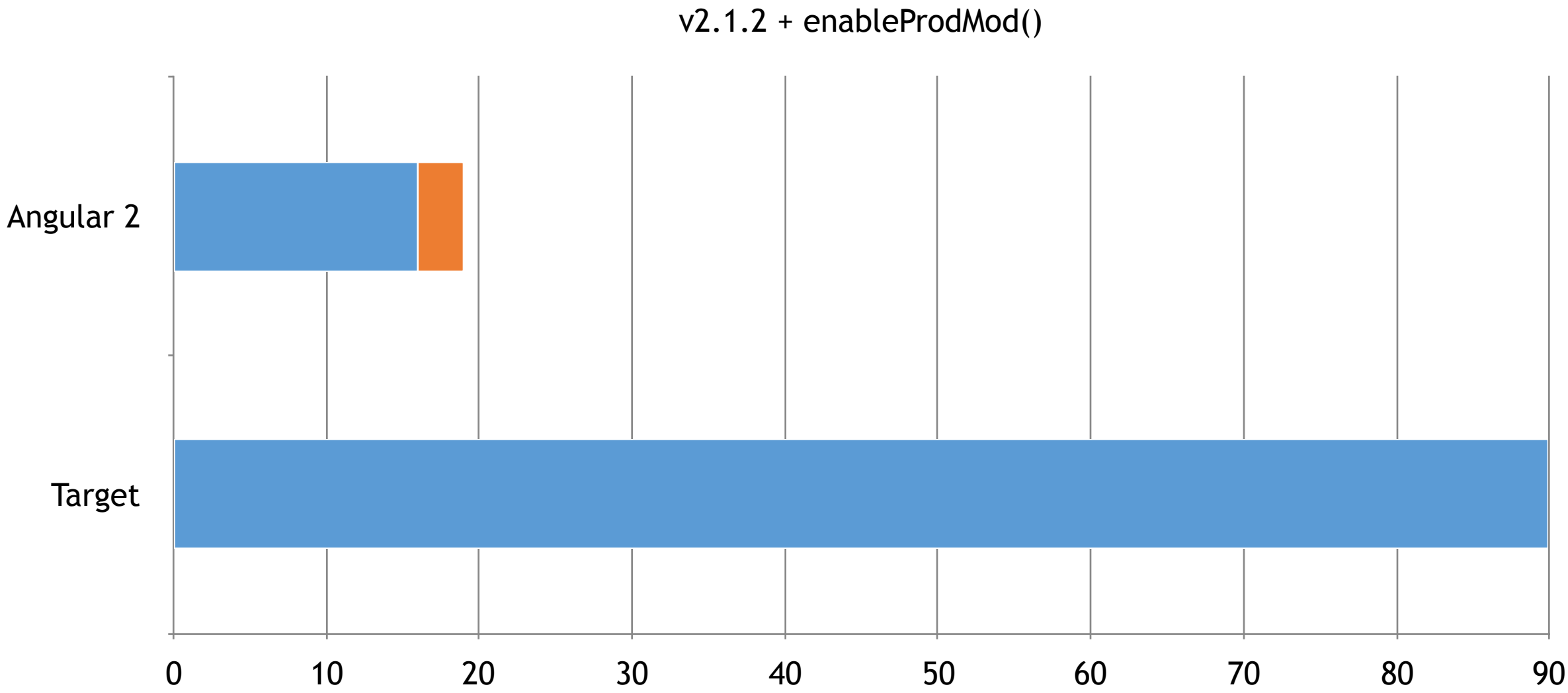

Alpha 44
Angular 2
Target
0
10
20
30
40
50
60
70
80
90

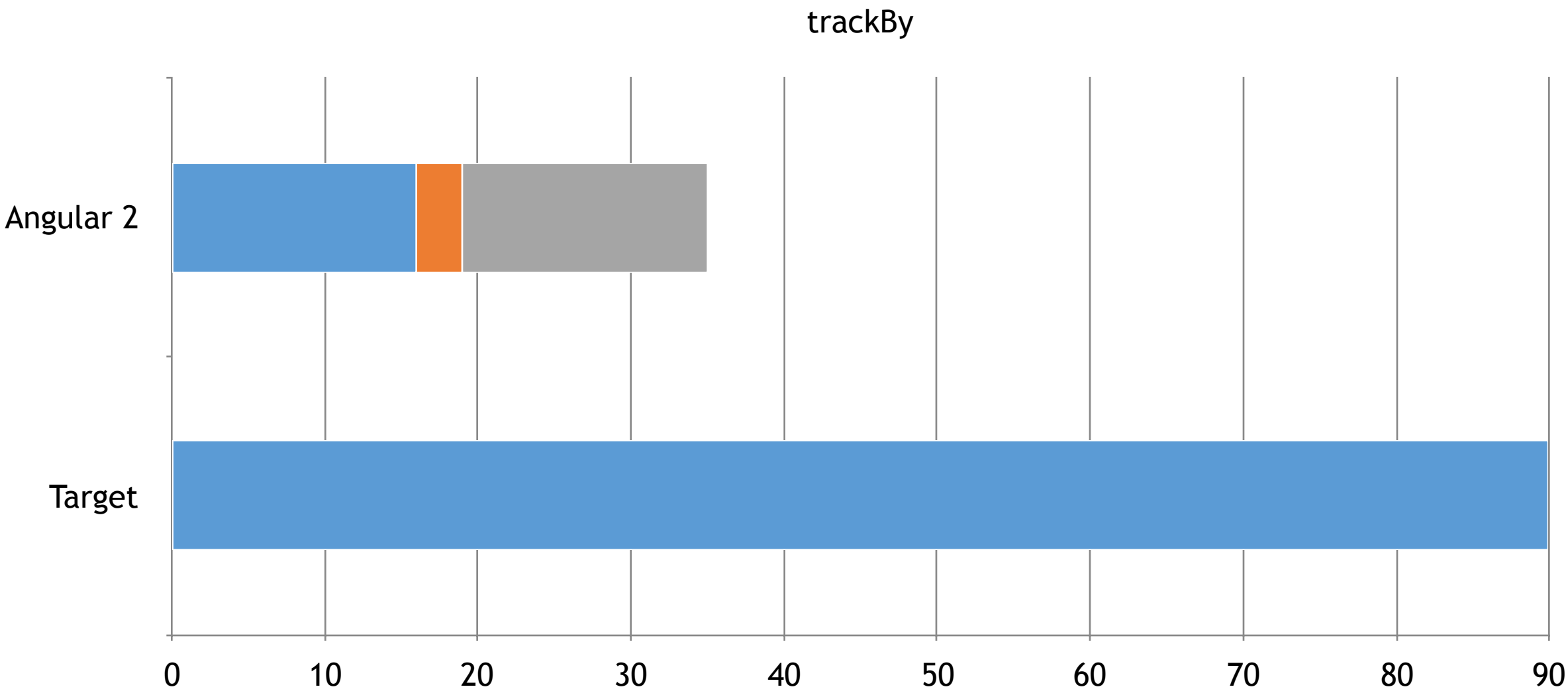

v2.1.2 + enableProdMod()
Angular 2
Target
0
10
20
30
40
50
60
70
80
90

```
function getData(keepIdentity) {
    var oldData = data;
    if (!keepIdentity) { // reset for each tick
      data = [];
      for (var i = 1; i <= ENV.rows; i++) {
        data.push({  …  });
        data.push({  …  });
      }
    }
}
```

```
@Page({
    template: `
        <div
          *ngFor="let post of posts;trackBy:identify">
            {{post.data}}
        </div>
    `
})
export class SomeConponent {
    identify(index,item){
        return post.id
    }
}
```

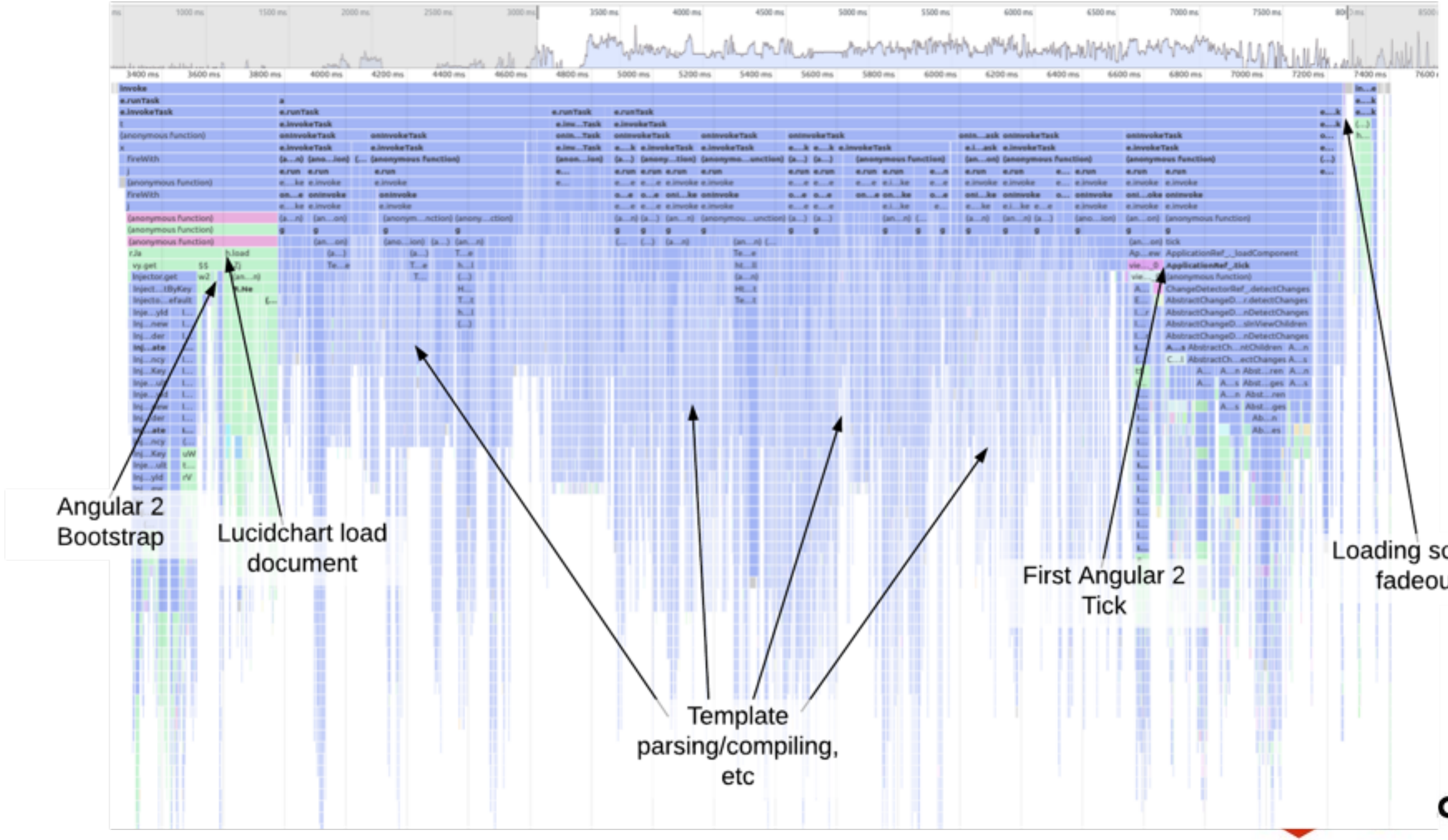
```
@Page({
    template: `
        <div
          *ngFor="let post of posts;trackBy:identify">
            {{post.data}}
        </div>
    `
})
export class SomeConponent {
    identify(index,item){
        return post.id
    }
}
```

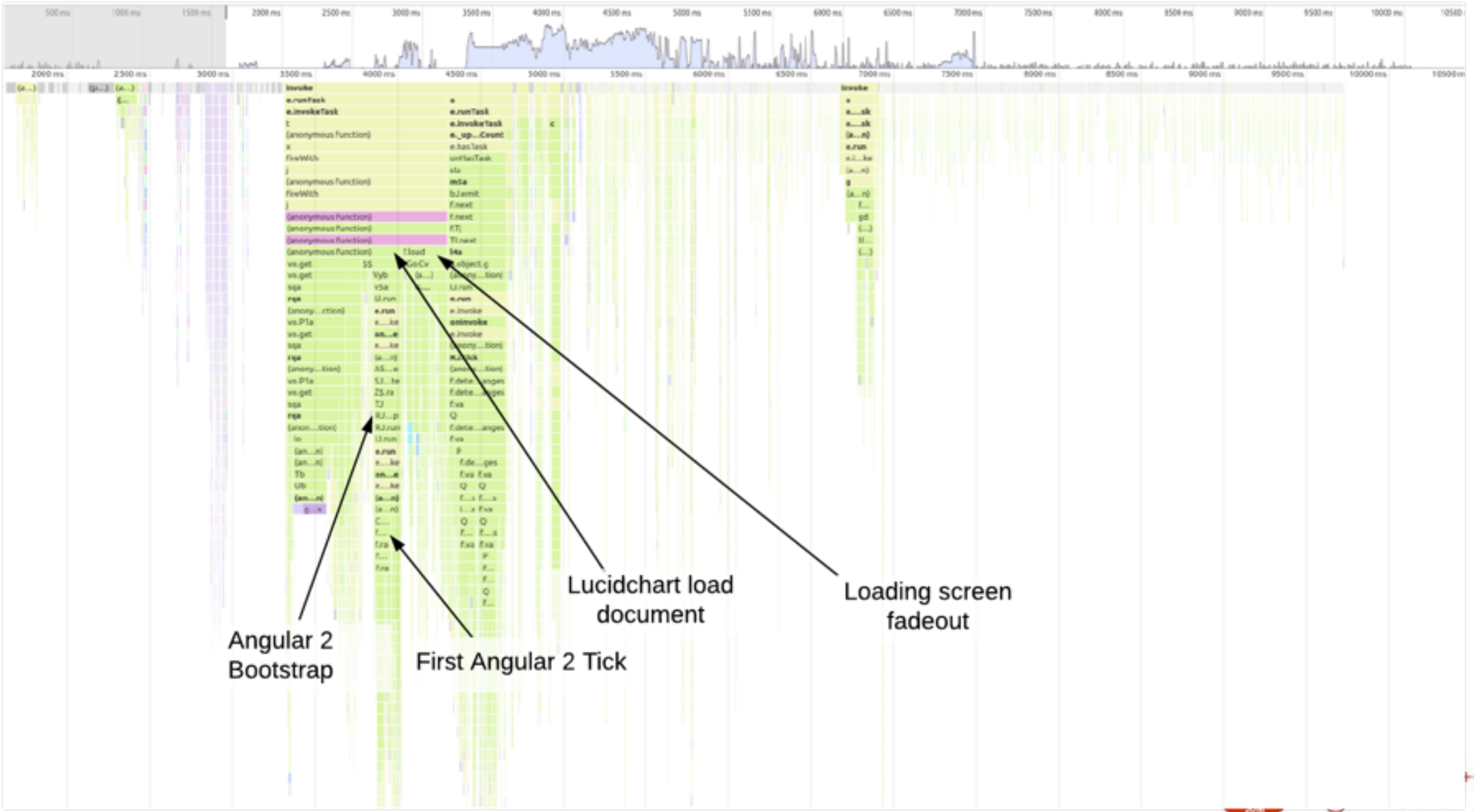

v2.1.2 + enableProdMod()
Angular 2
Target
0
10
20
30
40
50
60
70
80
90



AOT

Ahead Of Time template compilation

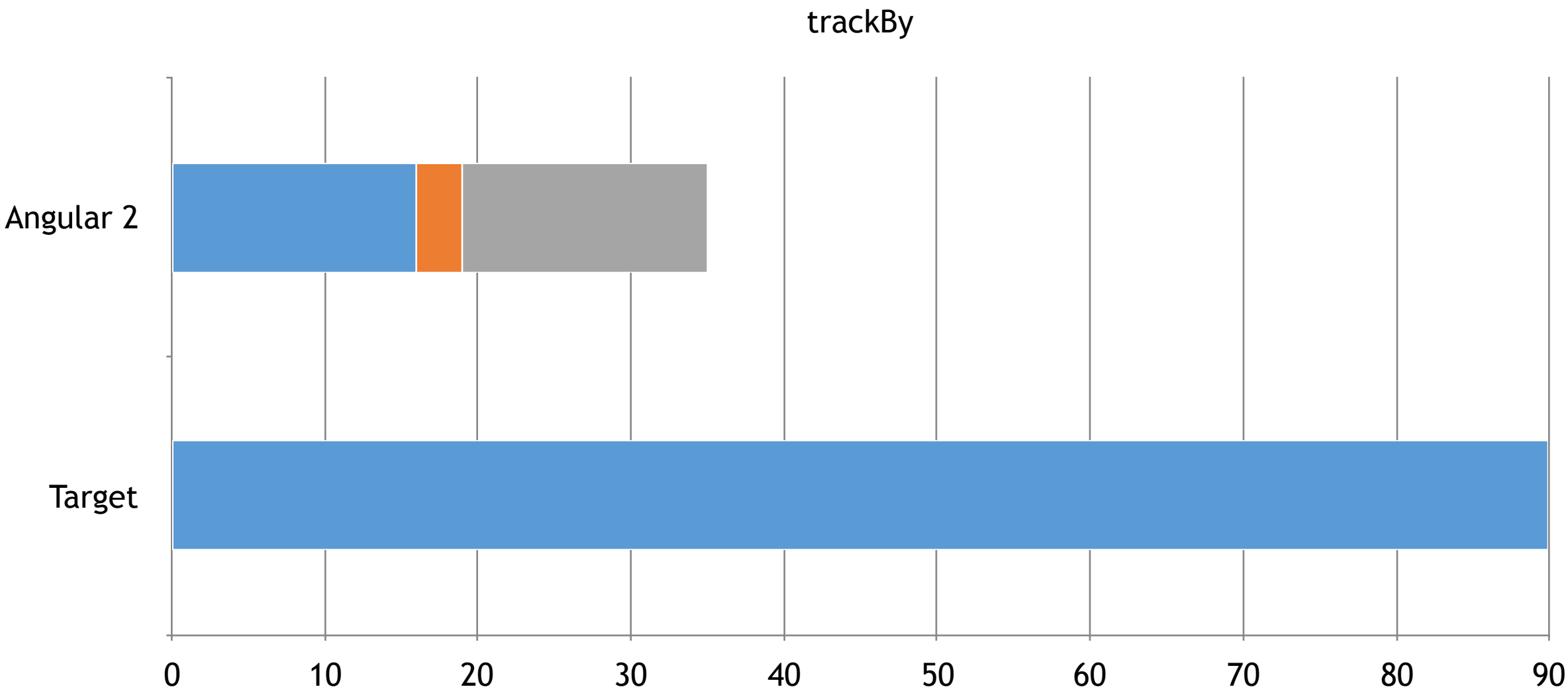

Angular 2
Bootstrap
Lucidchart load
document
Template
parsing/compiling,
etc
First Angular 2
Tick
Loading screen
fadeout

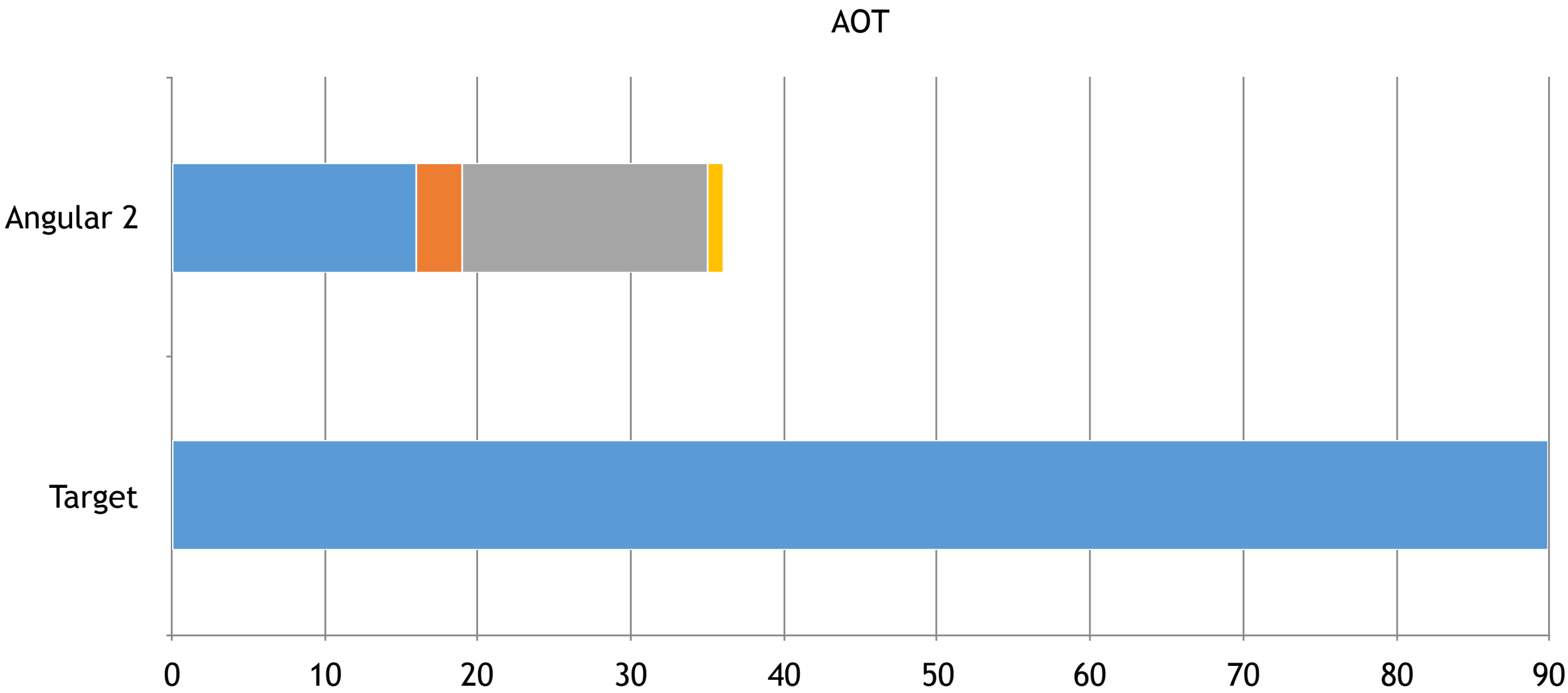

Angular 2
Bootstrap
First Angular 2 Tick
Lucidchart load
document
Loading screen
fadeout

Angular CLI

```
ng serve —–aot
```

```
ng build —–aot
```





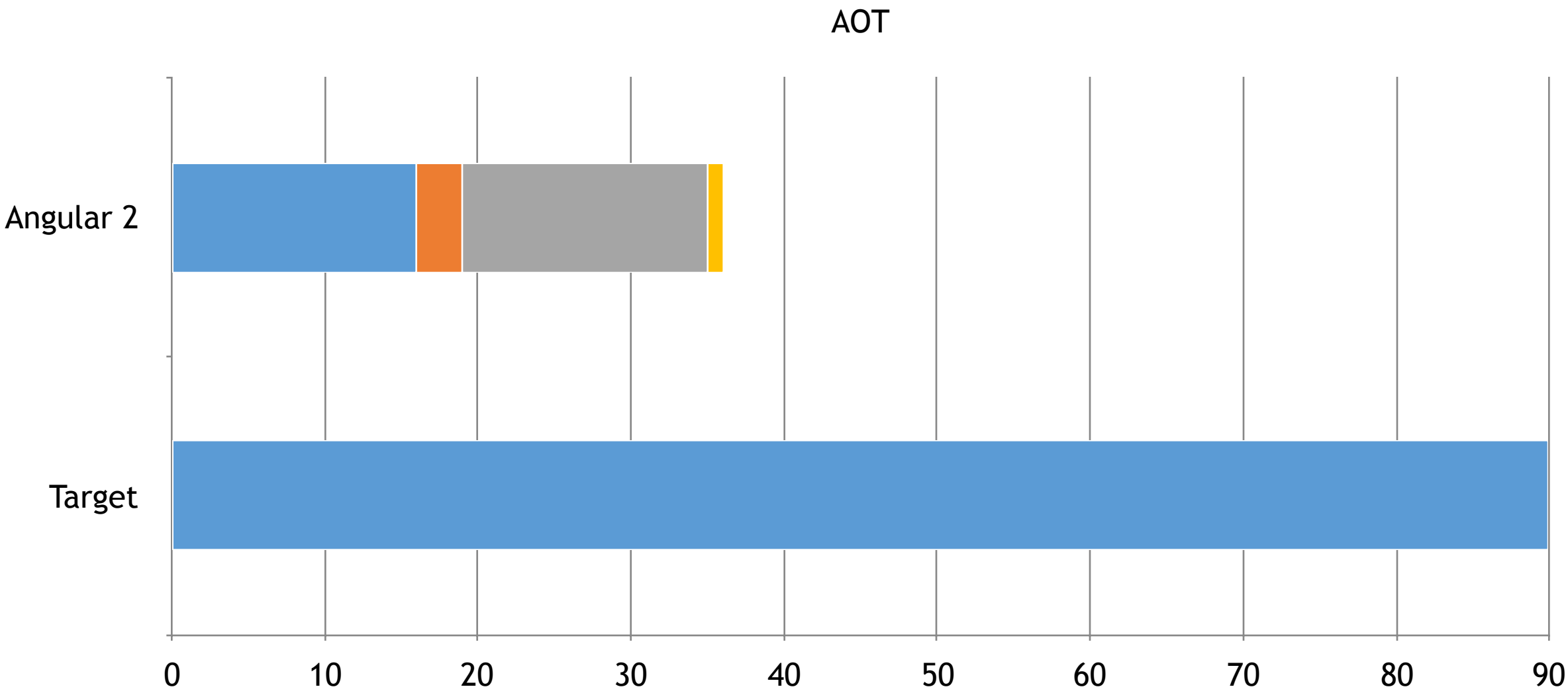
WebWorkers

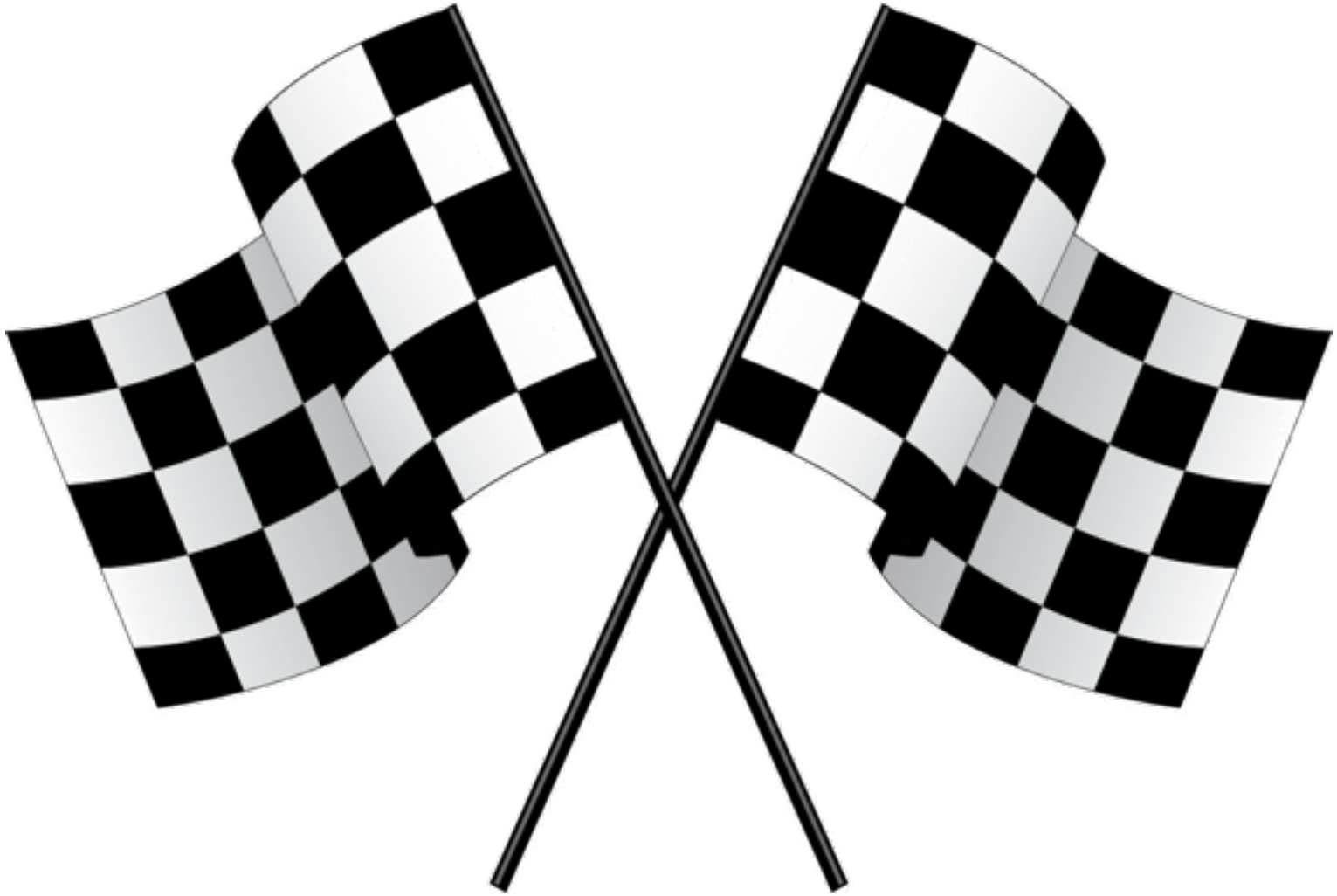
```
import {bootstrapWorkerUi} from '@angular/platform-webworker';
import {enableProdMode} from '@angular/core';

export function main() {
  enableProdMode();
  bootstrapWorkerUi('loader.js');
}
```

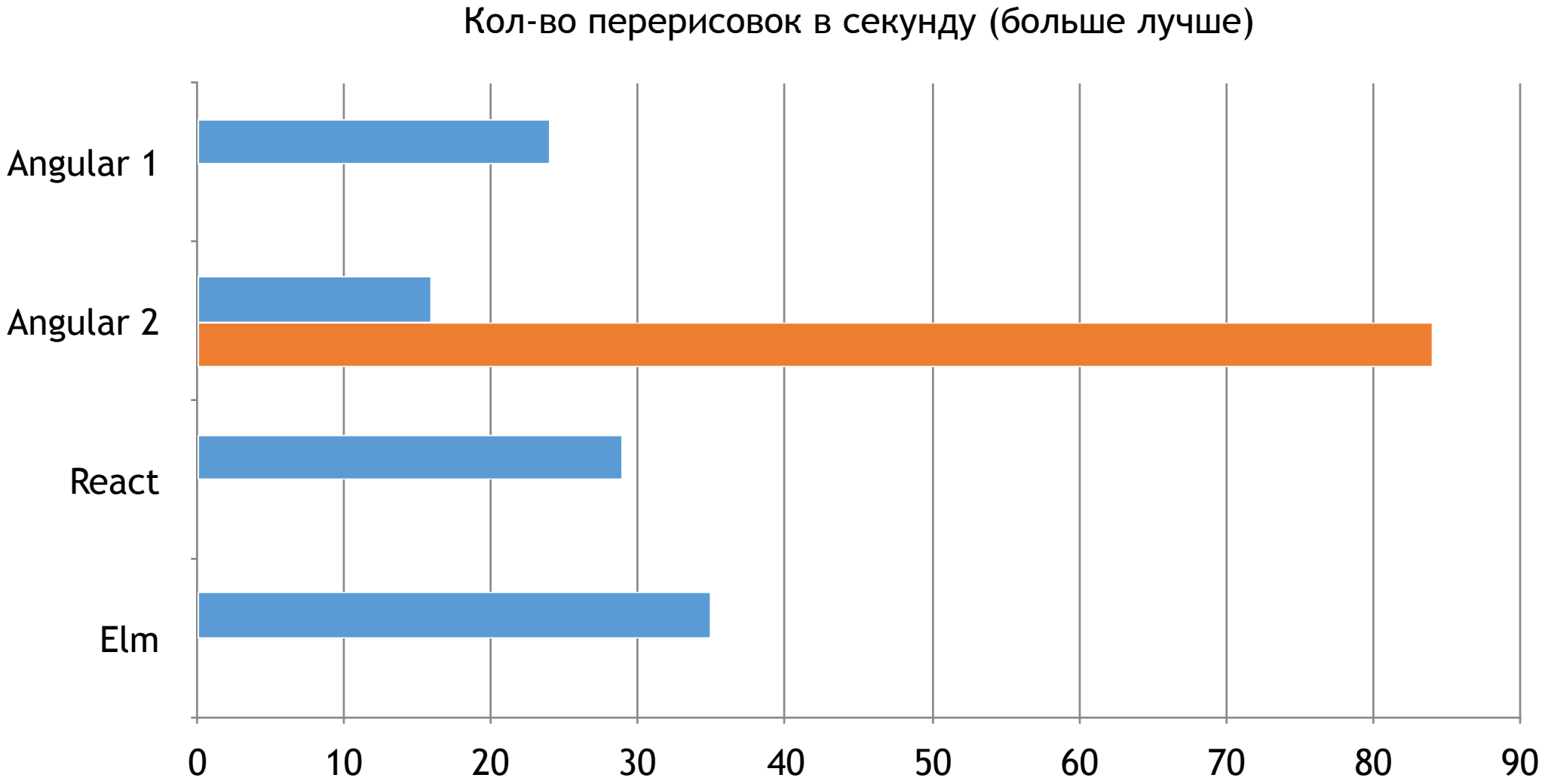
```
import {bootstrapWorkerUi} from '@angular/platform-webworker';
import {enableProdMode} from '@angular/core';

export function main() {
  enableProdMode();
  bootstrapWorkerUi('loader.js');
}
```

```
@NgModule({
  imports: [WorkerAppModule],
  bootstrap: [AppComponent],
  declarations: [AppComponent]
})
class WebWorkerModule {}

export function main() {
  enableProdMode();
  platformWorkerAppDynamic().bootstrapModule(WebWorkerModule);
}
```

```
@NgModule({
  imports: [WorkerAppModule],
  bootstrap: [AppComponent],
  declarations: [AppComponent]
})
class WebWorkerModule {}

export function main() {
  enableProdMode();
  platformWorkerAppDynamic().bootstrapModule(WebWorkerModule);
}
```



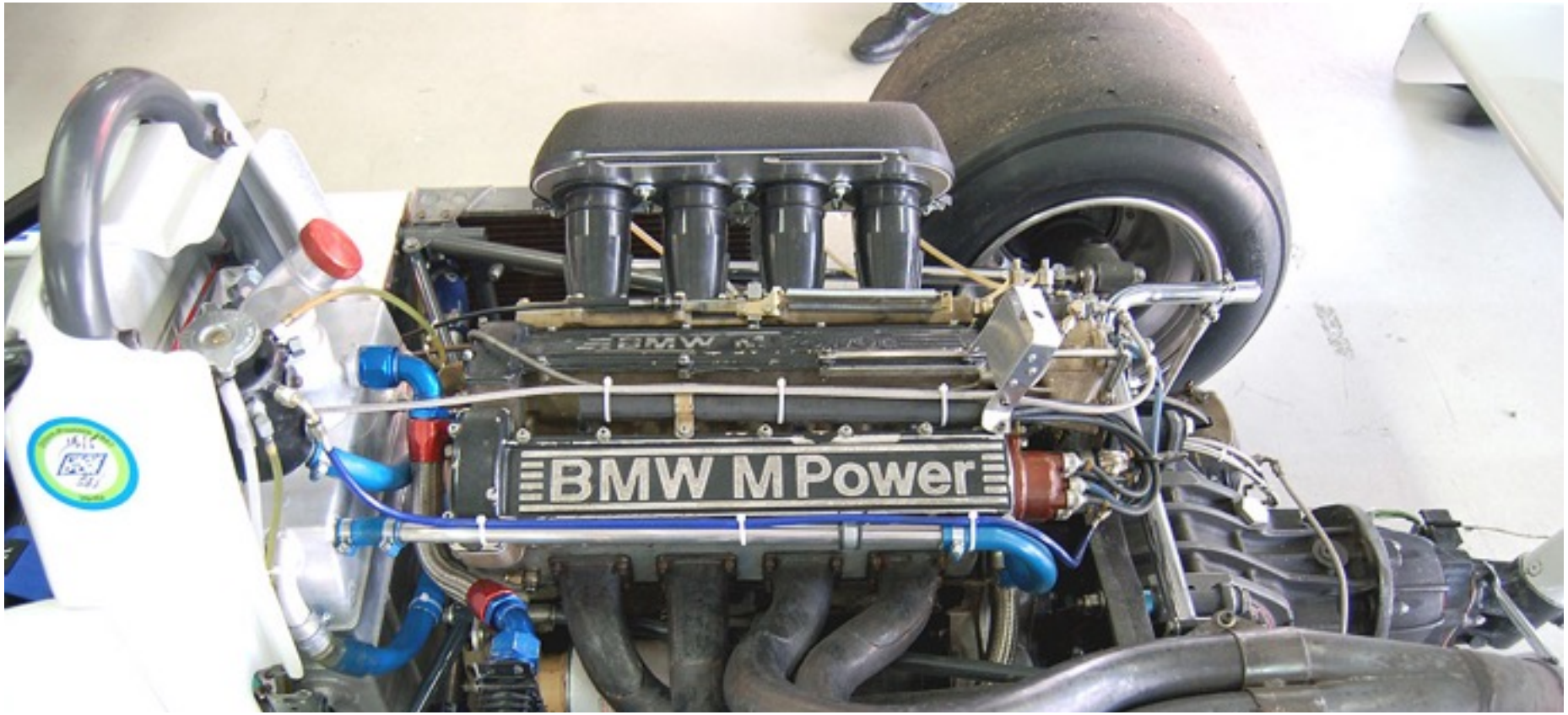



Кол-во перерисовок в секунду (больше лучше)
Angular 1
Angular 2
React
Elm
0
10
20
30
40
50
60
70
80
90

Еще раз

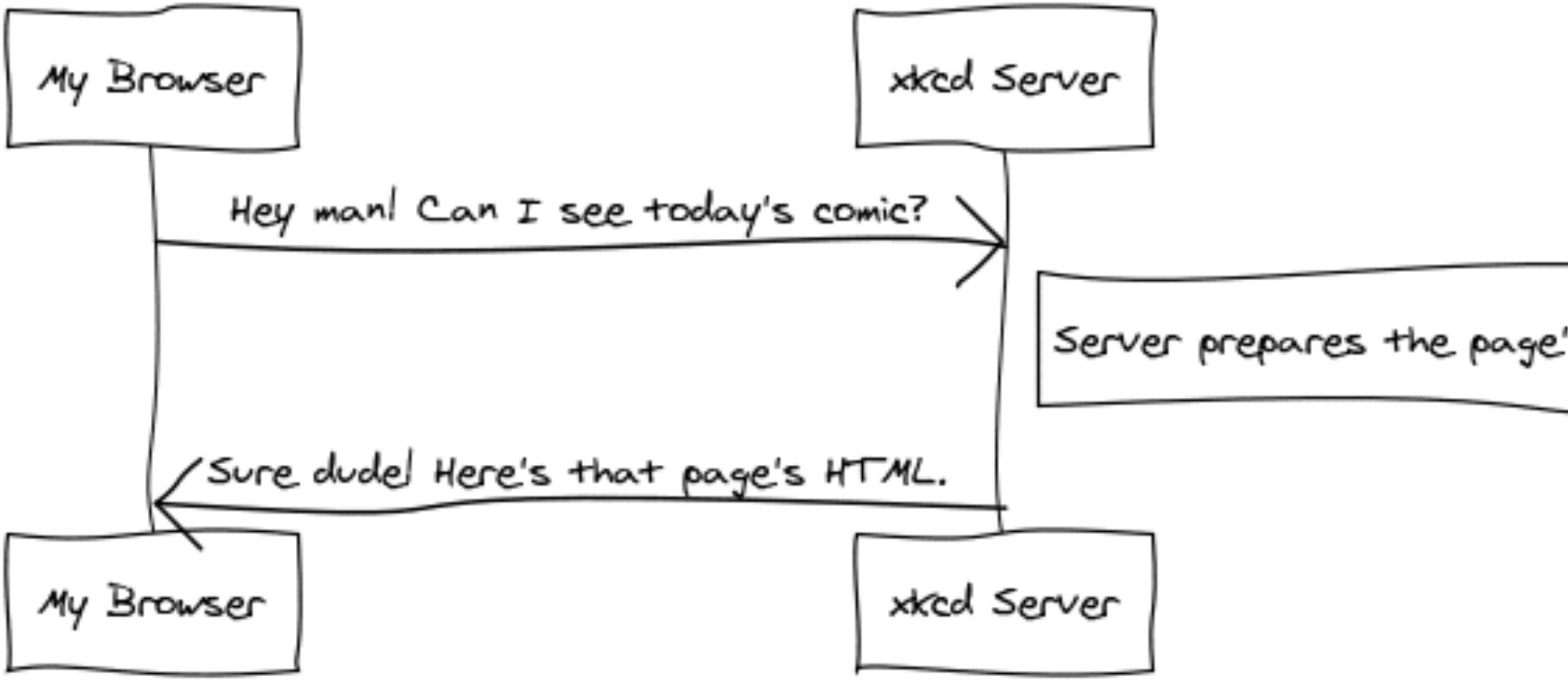
- enableProd()
- trackBy
- AOT
- WebWorkers

Заглянем под капот



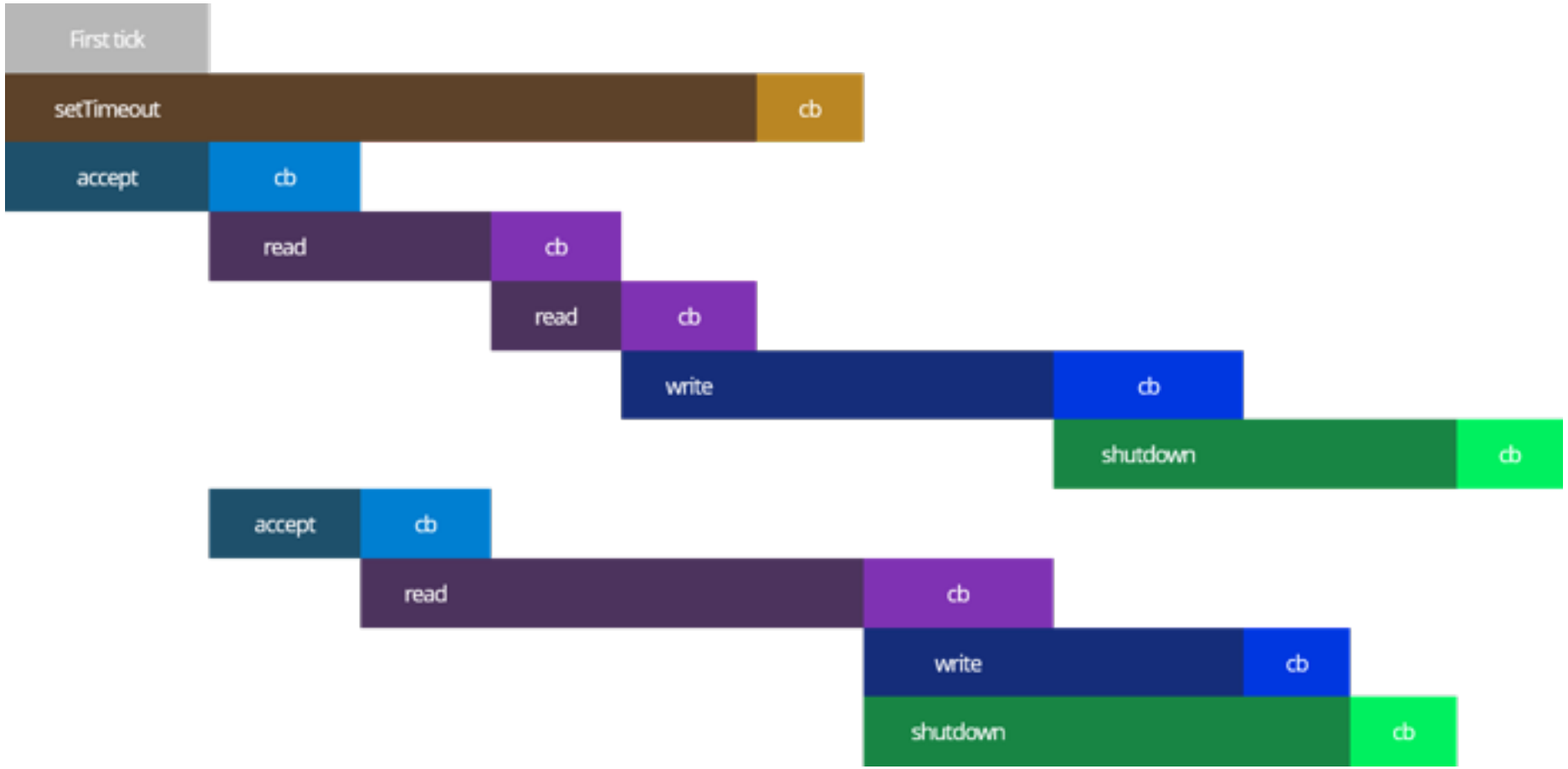
Scott Hanselman

ZoneJs



```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

```
process.on('uncaughtException', (err) => {
  console.log(`Caught exception: ${err}`);
});
```



```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone);
}

setTimeout(someCallback, 0);
```

```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone);
}

setTimeout(someCallback, 0);
```

```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone);
}

setTimeout(someCallback, 0);
```

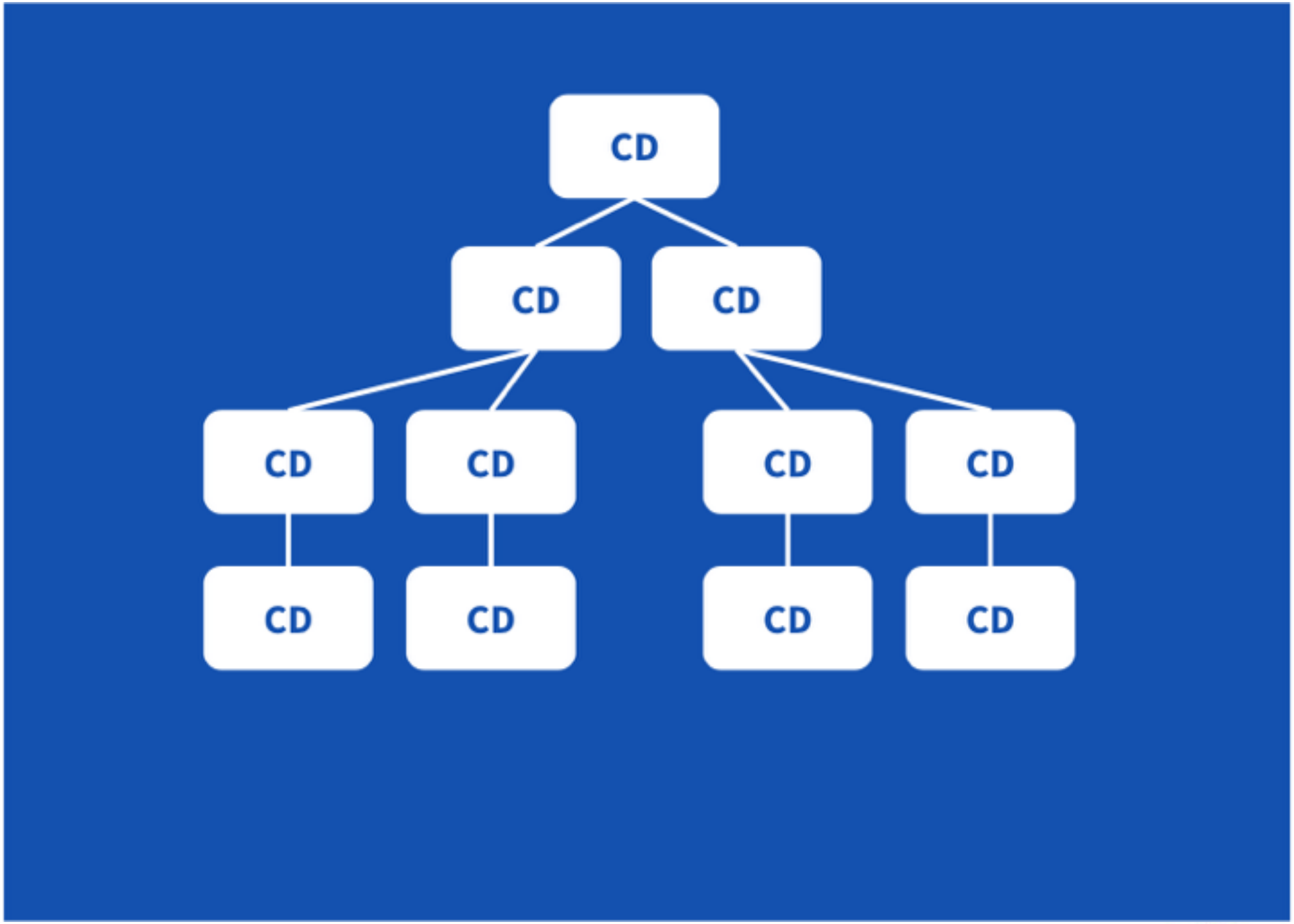
```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone);
}

setTimeout(someCallback, 0);
```

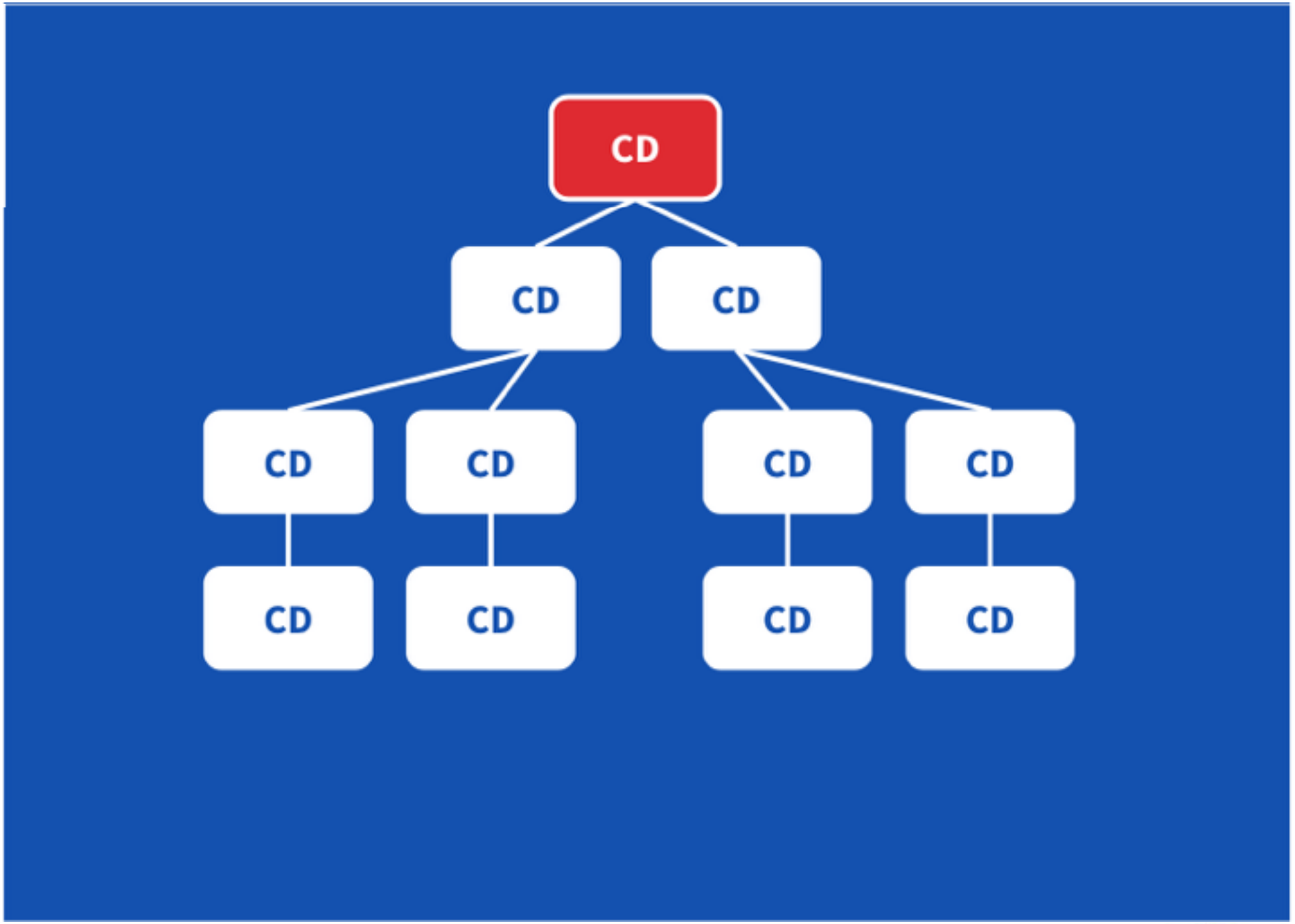
```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone); // TRUE
}

setTimeout(someCallback, 0);
```

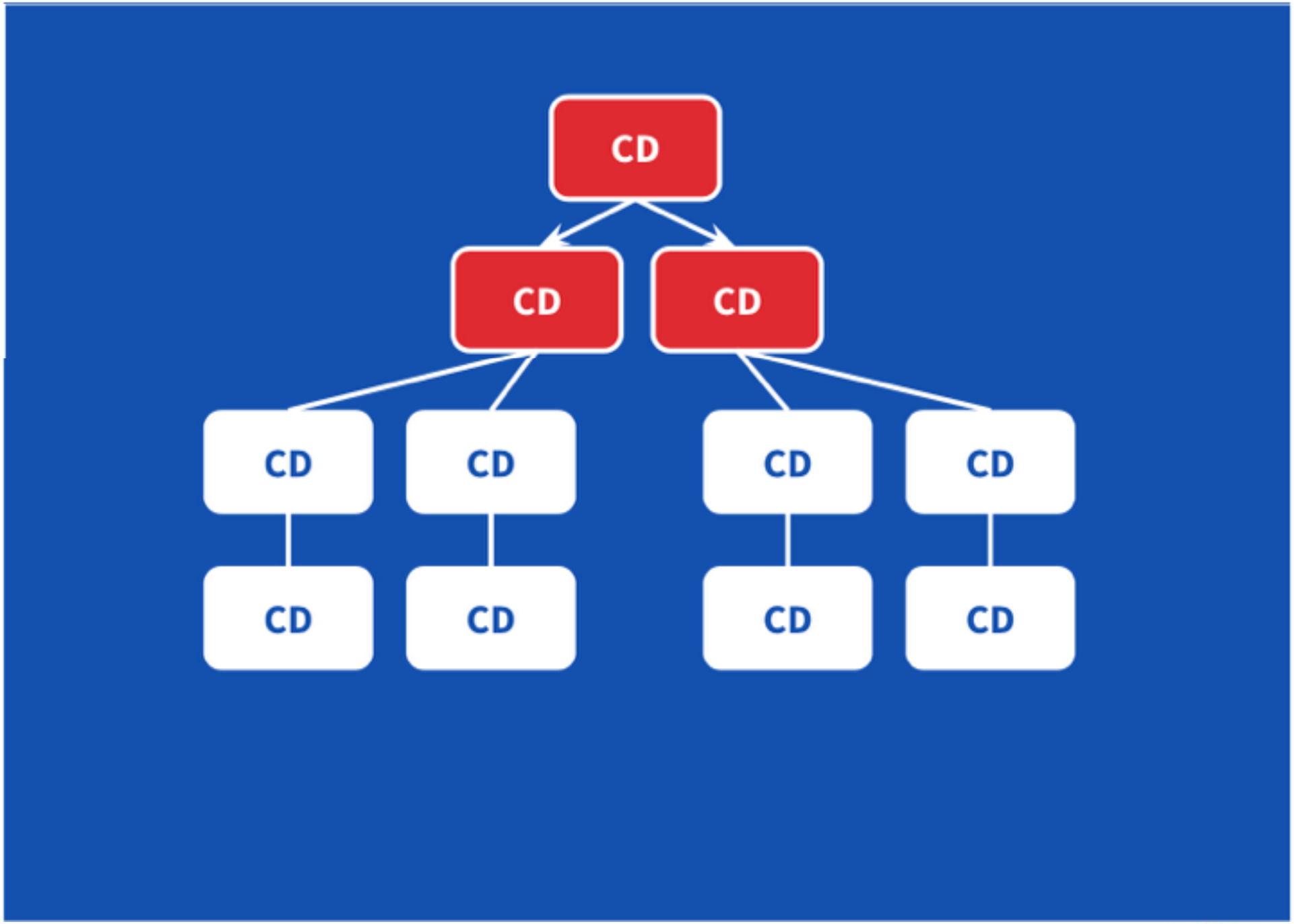
```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone);
}

setTimeout(someCallback, 0);
```

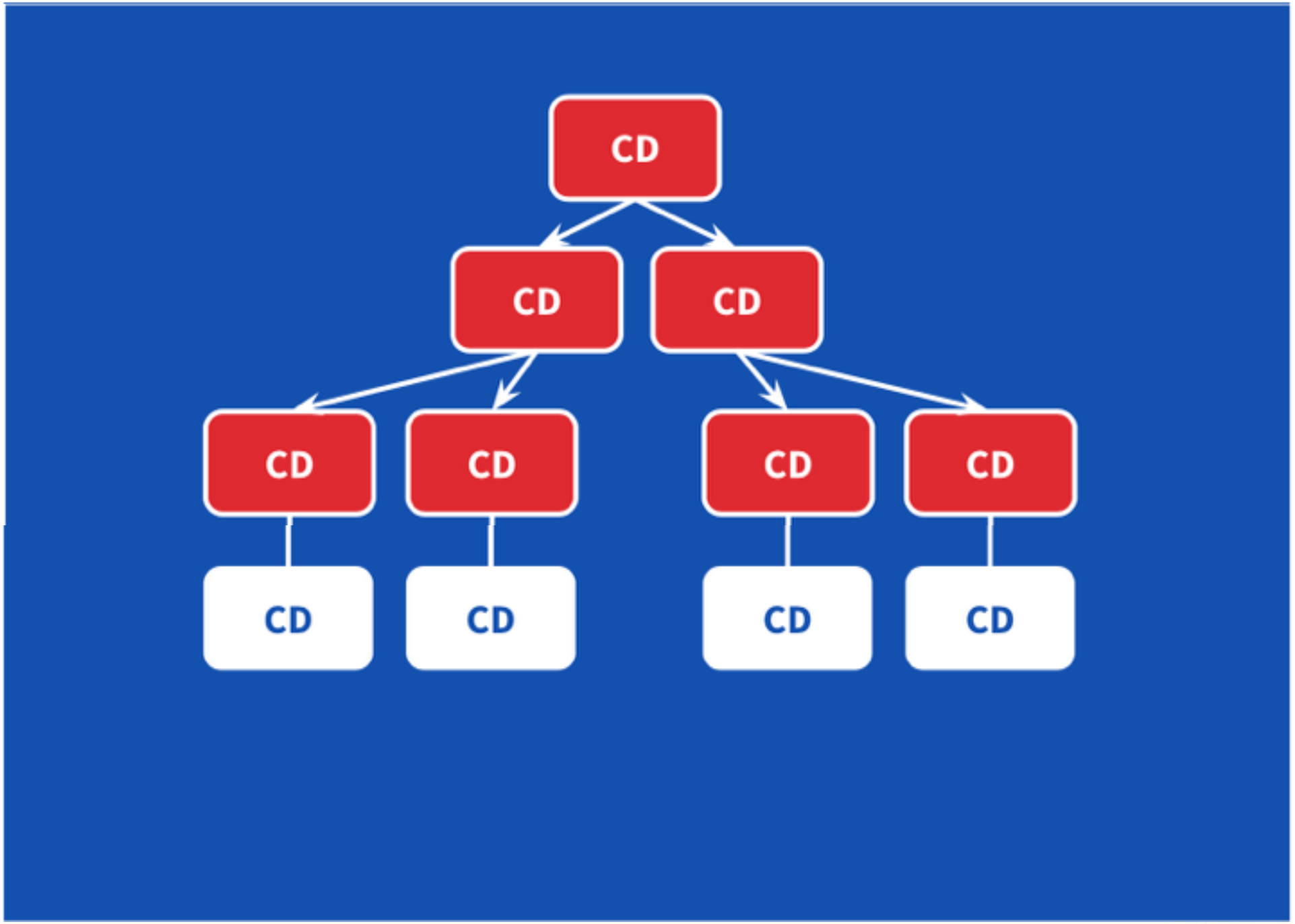
```
Zone.current.fork({}).run(function () {
    Zone.current.inTheZone = true;
    setTimeout(someCallback, 0);
});

function someCallback() {
    console.log(Zone.current.inTheZone); // FALSE
}

setTimeout(someCallback, 0);
```

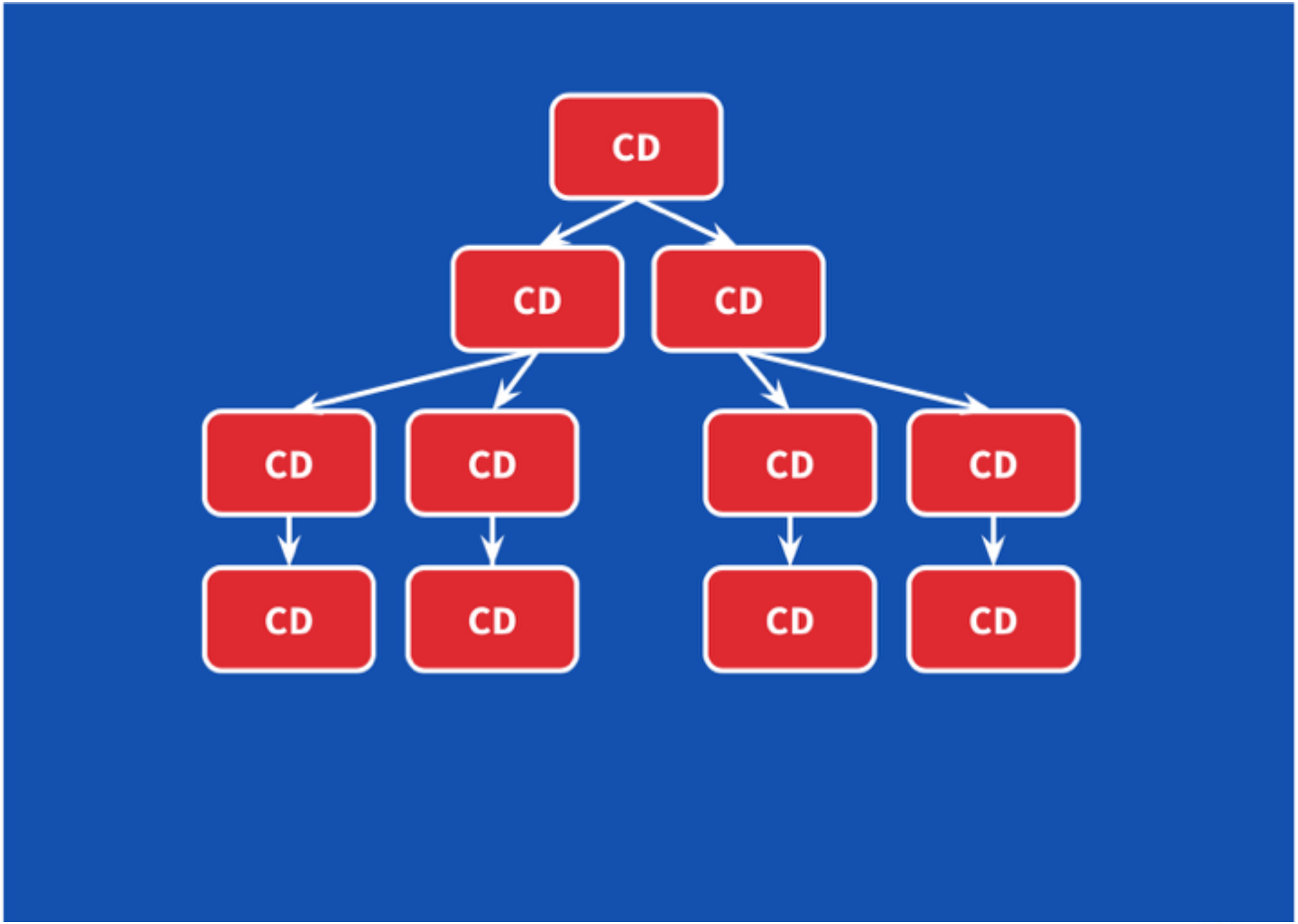
Change Detection


CD
CD
CD
CD
CD
CD
CD
CD
CD
CD


CD
CD
CD
CD
CD
CD
CD
CD
CD
CD


CD
CD
CD
CD
CD
CD
CD
CD
CD
CD


CD
CD
CD
CD
CD
CD
CD
CD
CD
CD

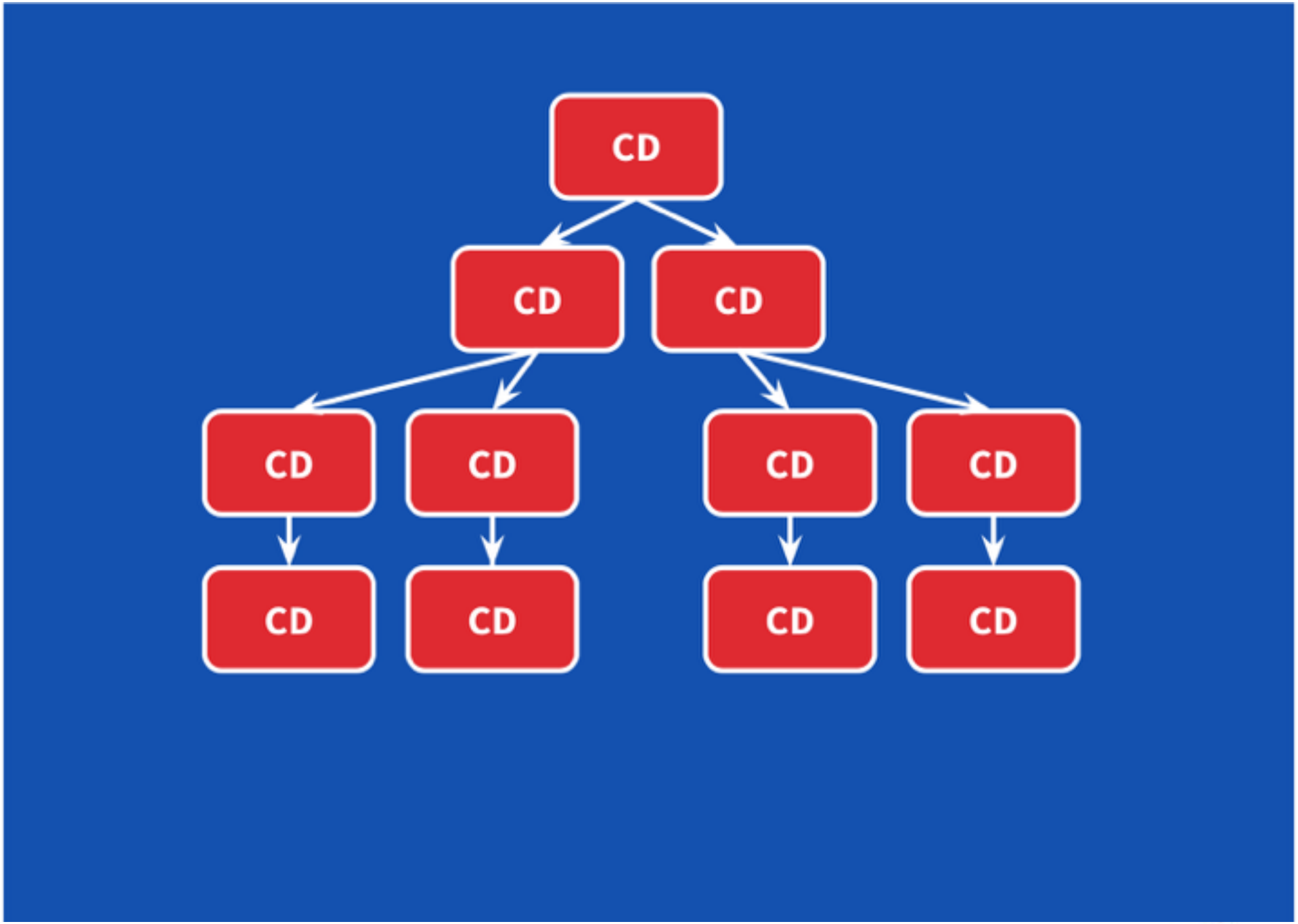

CD
CD
CD
CD
CD
CD
CD
CD
CD
CD

```
// very simplified version of actual source
class ApplicationRef {

  changeDetectorRefs:ChangeDetectorRef[] = [];

  constructor(private zone: NgZone) {
    this.zone.onTurnDone
        .subscribe(() => {
                this.zone.run(() => this.tick()
            });
  }

  tick() {
    this.changeDetectorRefs
      .forEach((ref) => ref.detectChanges());
  }
}
```

```
// very simplified version of actual source
class ApplicationRef {

  changeDetectorRefs:ChangeDetectorRef[] = [];

  constructor(private zone: NgZone) {
    this.zone.onTurnDone
        .subscribe(() => {
                this.zone.run(() => this.tick()
            });
  }

  tick() {
    this.changeDetectorRefs
      .forEach((ref) => ref.detectChanges());
  }
}
```

```
// very simplified version of actual source
class ApplicationRef {

  changeDetectorRefs:ChangeDetectorRef[] = [];

  constructor(private zone: NgZone) {
    this.zone.onTurnDone
        .subscribe(() => {
                this.zone.run(() => this.tick()
            });
  }

  tick() {
    this.changeDetectorRefs
      .forEach((ref) => ref.detectChanges());
  }
}
```

```
// very simplified version of actual source
class ApplicationRef {

  changeDetectorRefs:ChangeDetectorRef[] = [];

  constructor(private zone: NgZone) {
    this.zone.onTurnDone
        .subscribe(() => {
                this.zone.run(() => this.tick()
            });
  }

  tick() {
    this.changeDetectorRefs
      .forEach((ref) => ref.detectChanges());
  }
}
```

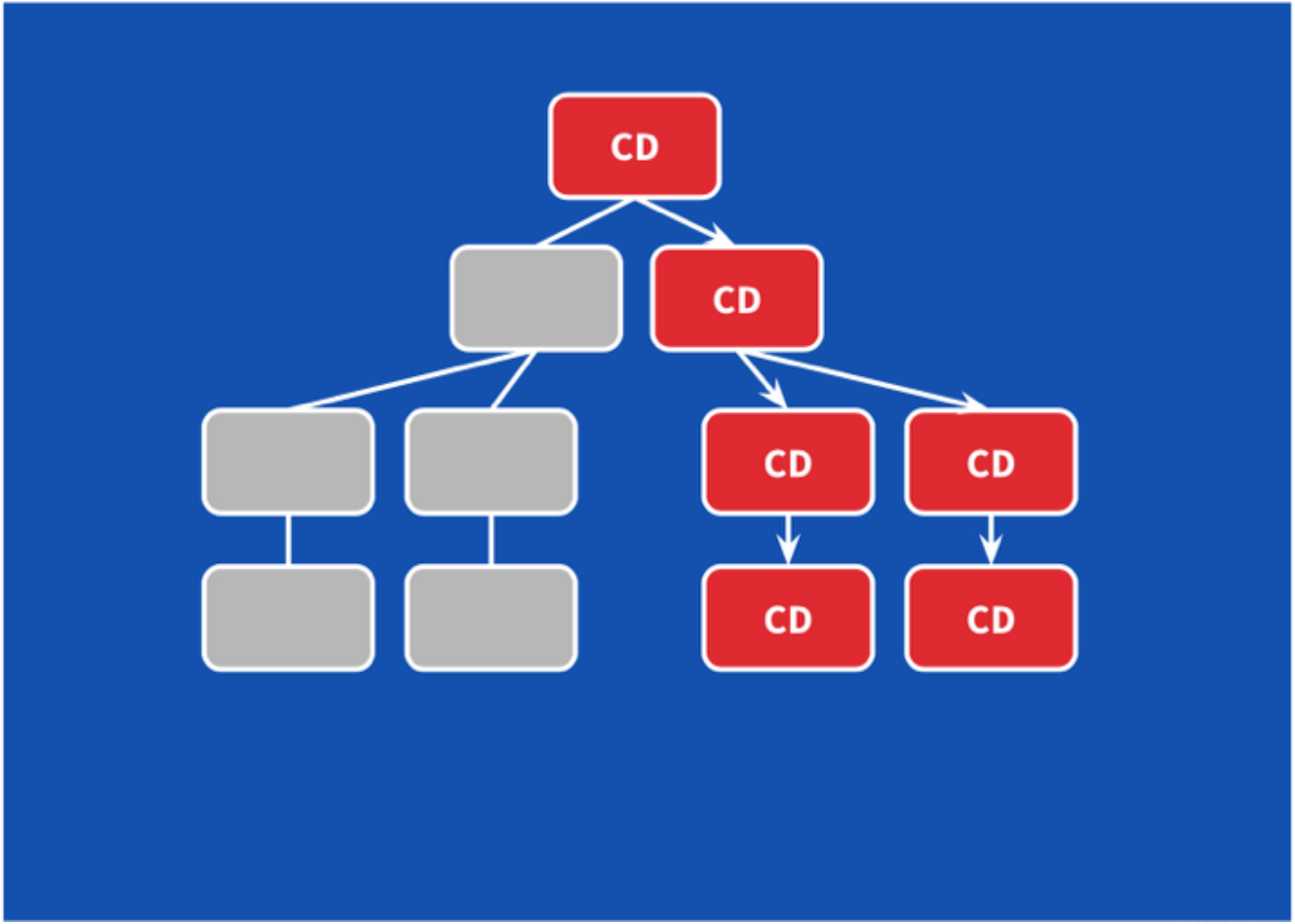

CD
CD
CD
CD
CD
CD
CD
CD
CD
CD

```
@Component({
  template: '<v-card [vData]="vData"></v-card>'
})
class VCardApp {

  constructor() {
    this.vData = {
      name: 'Christoph Burgdorf',
      email: 'christoph@thoughtram.io'
    }
  }

  changeData() {
    this.vData.name = 'Pascal Precht';
  }
}
```

```
@Component({
  template: `
    <h2>{{vData.name}}</h2>
    <span>{{vData.email}}</span>
  `
})
class VCardCmp {
  @Input() vData;
}
```

```
@Component({
  template: '<v-card [vData]="vData"></v-card>'
})
class VCardApp {

  constructor() {
    this.vData = {
      name: 'Christoph Burgdorf',
      email: 'christoph@thoughtram.io'
    }
  }

  changeData() {
    this.vData = { name: 'Pascal Precht' };
  }
}
```

```
@Component({
  template: `
    <h2>{{vData.name}}</h2>
    <span>{{vData.email}}</span>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush
})
class VCardCmp {
  @Input() vData;
}
```


CD
CD
CD
CD
CD
CD

Управляем Zone и CD

```
constructor(private zone: NgZone) {}
```

```
processOutsideAngularZone() {
  this.progress = 0;
  this.zone.runOutsideAngular(() => {
    this.increaseProgress(() => {
      this.zone.run(() => {
        console.log('Outside Done!');
      });
    });
  });
}
```

```
processOutsideAngularZone() {
  this.progress = 0;
  this.zone.runOutsideAngular(() => {
    this.increaseProgress(() => {
      this.zone.run(() => {
        console.log('Outside Done!');
      });
    });
  });
}
```

```
constructor(private cd: ChangeDetectorRef) {}
```

```
ngOnInit() {
    this.addItemStream.subscribe(() => {
      this.counter++; // application state changed
      this.cd.markForCheck(); // marks path
    })
  }
}
```

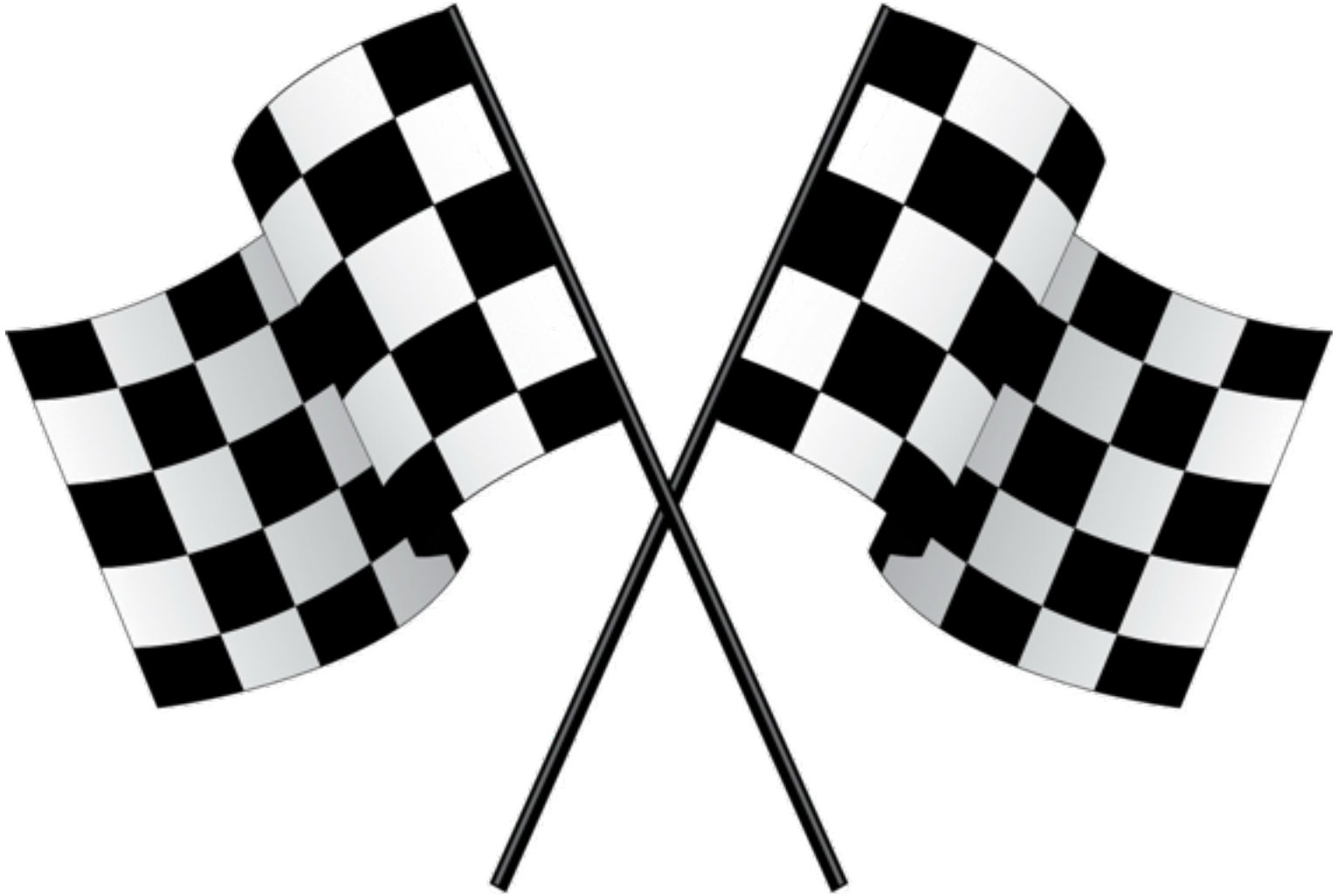
А можно как-то попроще?

Redux

ng2-redux
ngrx/store

Mobx

ng2-mobx



Улучшаем Perceived Performance

- Увеличивая реальную производительность

Улучшаем Perceived Performance

- Увеличивая реальную производительность
- Замедляя реальную производительность

Улучшаем Perceived Performance

- Увеличивая реальную производительность
- Замедляя реальную производительность
- Грамотно перераспределяя нагрузку и ресурсы

NAS, Predictions, Preloading, Presudo-Isomorphism

Производительность фронтенда

Доклад принят в Программу конференции

Алексей Охрименко

IPONWEB

Senior JavaScript Developer.

Видео

FRP

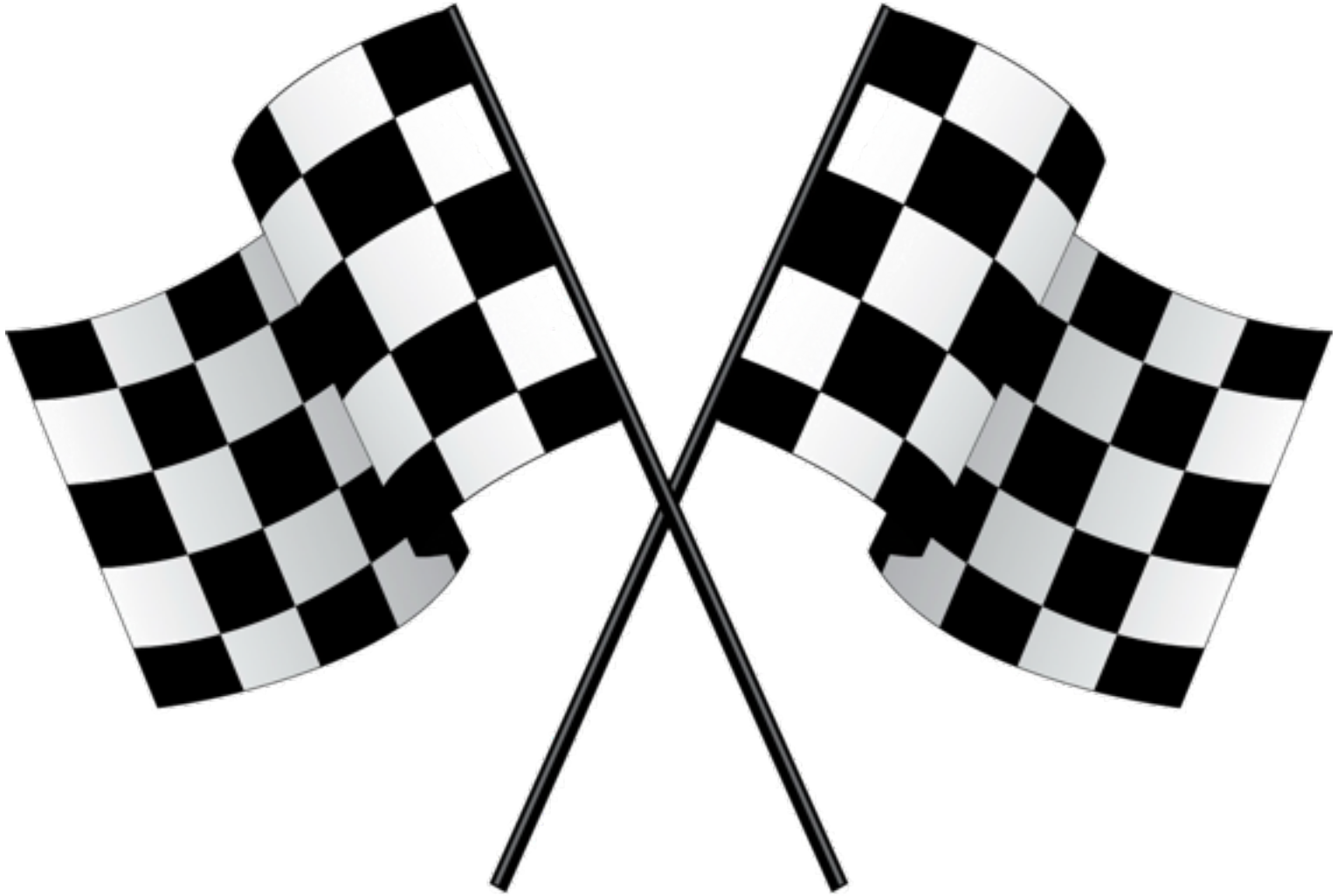
functional reactive programming

```
this.form.valueChanges
    .filter((value) => this.form.valid)
    .switchMap((value) => {
       return http.post('/api', value)
    });
```

```
this.form.valueChanges
    .debounce(500)
    .filter((value) => this.form.valid)
    .switchMap((value) => {
       return http.post('/api', value)
    });
```

```
this.form.valueChanges
    .debounce(500)
    .distinctUntilChanged()
    .filter((value) => this.form.valid)
    .switchMap((value) => {
       return http.post('/api', value)
    });
```

```
this.form.valueChanges
    .debounce(500)
    .distinctUntilChanged()
    .filter((value) => this.form.valid)
    .switchMap((value) => {
       return http.post('/api', value)
    }).retryWhen(attempts =>
       attempts
         .zip(Observable.range(1, 3), (_, i) => i)
         .flatMap((i: number) => {
           return Observable.timer(i * 1000);
         })
    ))
```



Алексей
Охрименко

http://bit.ly/2eMOBjm

Twitter: @Ai_boy

IPONWEB