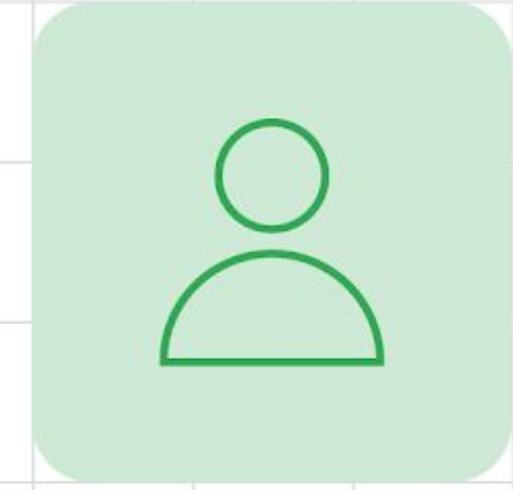


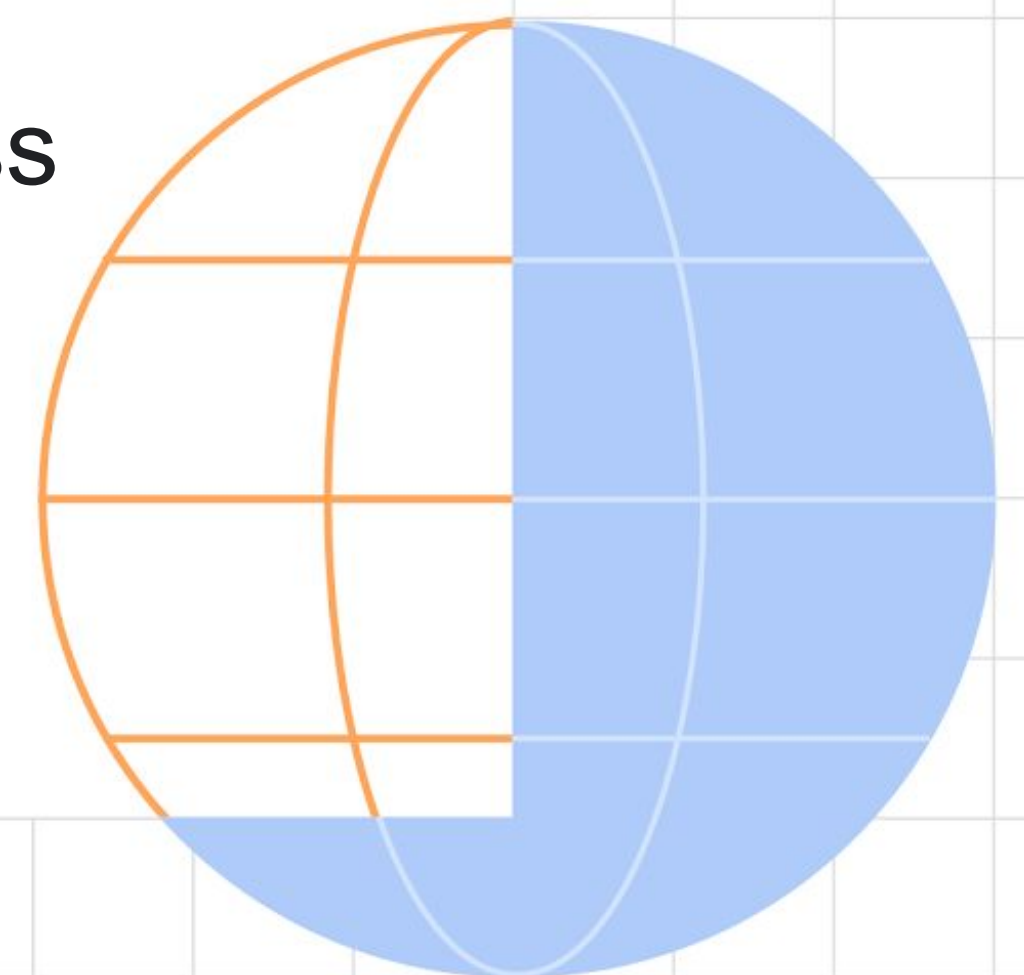


Testing Components with Angular CDK

Introduction to Angular CDK component harness



Aristeidis Bampakos
GDE Greece
@abampakos



Agenda

What we will see today

- Component unit testing in Angular
- Why use component harness
- Use harness with Angular Material
- Write harness for your Angular library

That's me! 😊

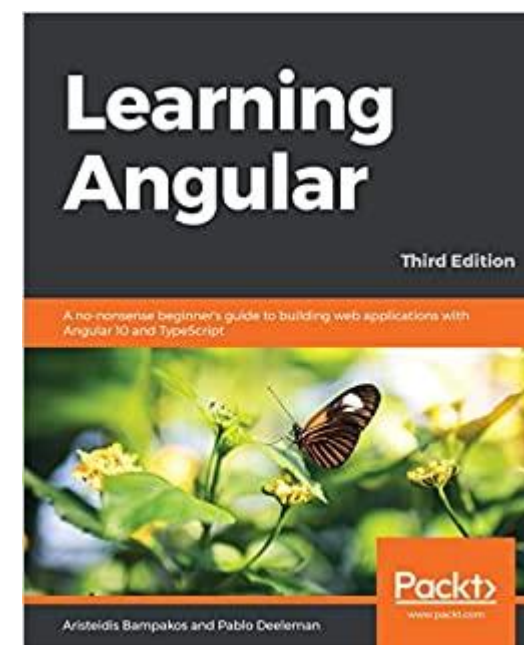
Web Development Team Lead at Plex-Earth

- Live in Greece
- Angular GDE
- Senior Angular Tech Instructor at Code.Hub
- Book author at Packt

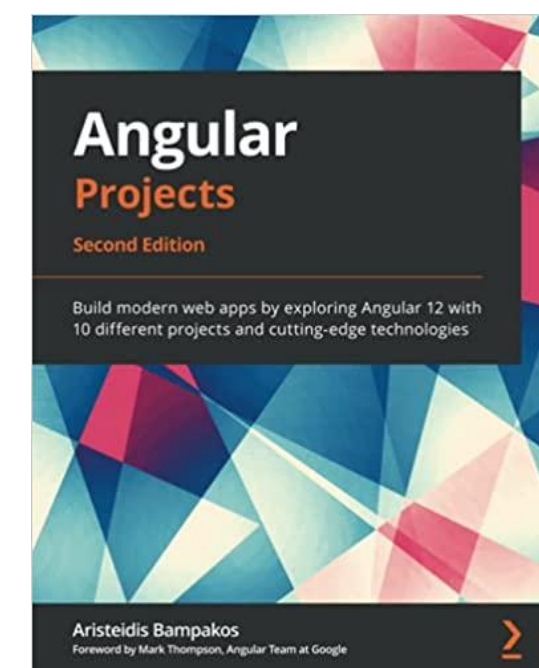
Book author

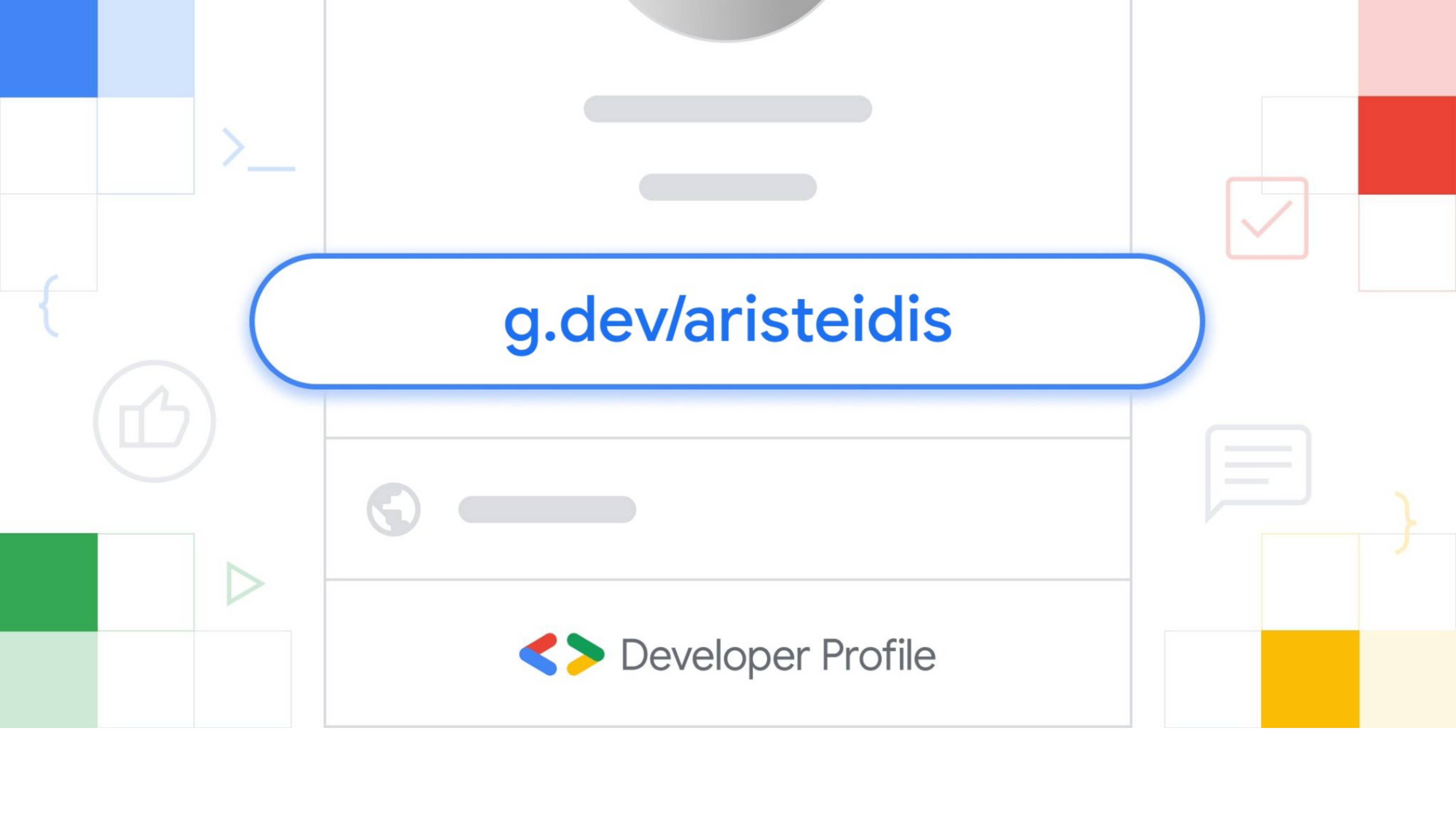
Two books in two years 🤖

September 2020



July 2021



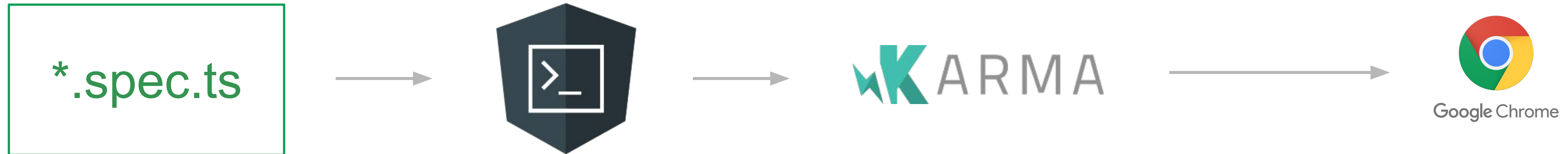


g.dev/aristeidis

 Developer Profile

Unit testing in Angular

ng test



Unit testing in Angular

Testing components

A component unit test checks the interaction of the TypeScript class with the HTML template in the context of the Angular framework



Angular TestBed

Component DOM interaction

Create the component in the browser DOM and validate its behavior and state.



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

```
@Component({  
  selector: 'app-hello-world',  
  template: '<span><ng-content></ng-content></span>',  
})
```

```
export class HelloWorldComponent {}
```

```
@Component({  
  selector: 'app-root',  
  template: '<app-hello-world>{{name}}</app-hello-world>',  
})
```

```
export class AppComponent {  
  name = 'Hello Angular CDK';  
}
```

Angular components

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';
import { HelloWorldComponent } from './hello-world.component';

describe('AppComponent', () => {
  let fixture: ComponentFixture<AppComponent>;
  let component: AppComponent;

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [AppComponent, HelloWorldComponent]
    });

    fixture = TestBed.createComponent(AppComponent);
    component = fixture.componentInstance;
  });
});
```

Test suite setup

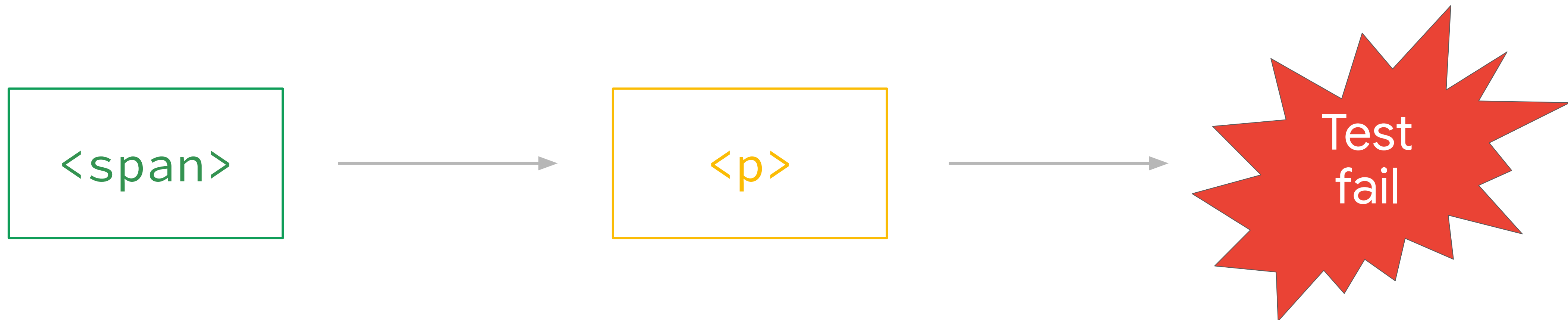
```
it('should create', () => {  
  expect(component).toBeTruthy();  
});
```

```
it('should display a greeting', () => {  
  component.name = 'Angular Nation';  
  fixture.detectChanges();  
  const nameDisplay: HTMLElement = fixture.nativeElement.querySelector('span');  
  expect(nameDisplay.textContent).toEqual('Angular Nation');  
});
```

Component tests

Unit test failure

Relying on component internals may break your tests



Angular CDK

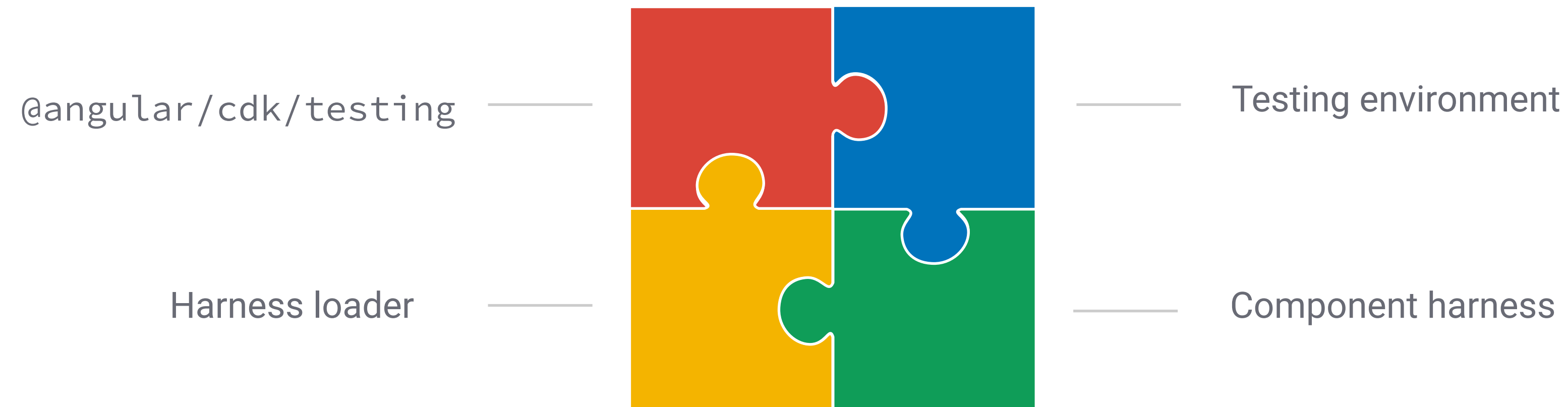
Component harness

A set of utilities that allow a test to interact with a component over a public testing API.



Angular CDK testing

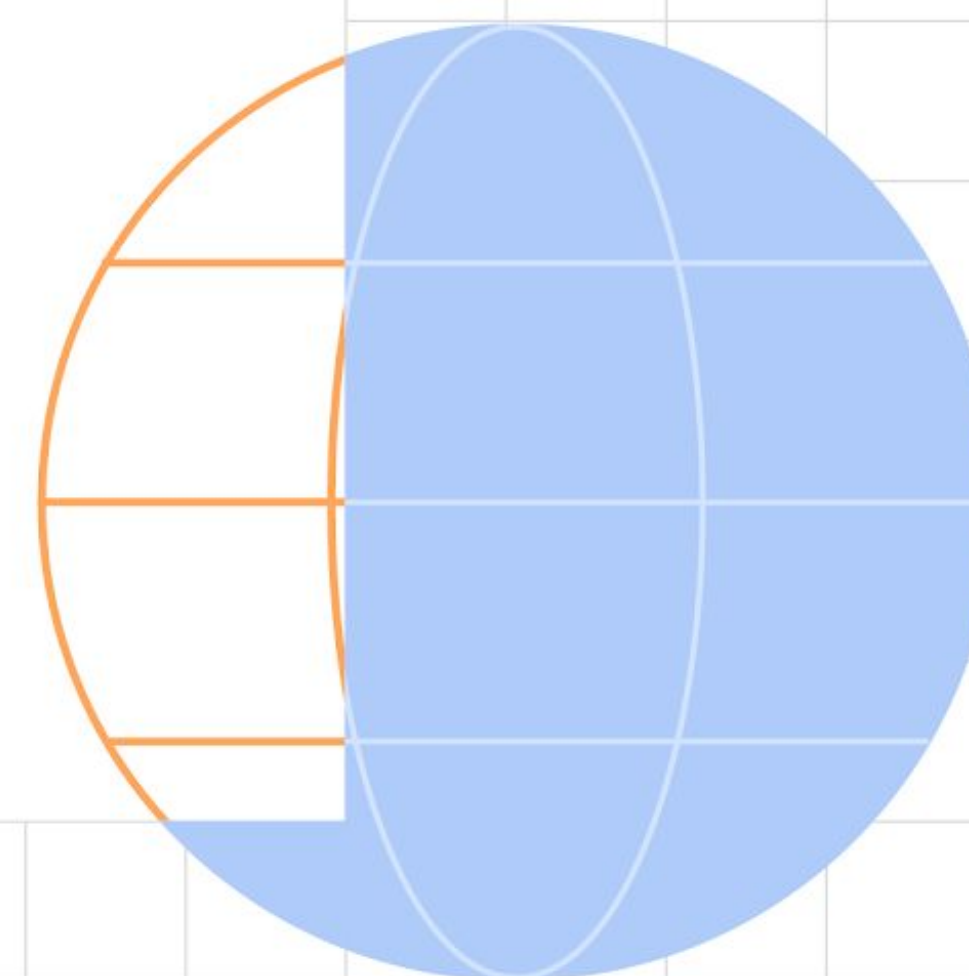
Anatomy of a component harness test





Harness cooking time 🍴

Let's write an Angular Material test!



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

@Component({
  selector: 'app-button',
  template: '<button mat-button>{{ title }}</button>',
})
export class ButtonComponent {
  title = 'I am a Material button';
}
```

Angular Material component

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ButtonComponent } from './button.component';

import { HarnessLoader } from '@angular/cdk/testing';
import { TestbedHarnessEnvironment } from '@angular/cdk/testing/testbed';
import { MatButtonHarness } from '@angular/material/button/testing';

describe('ButtonComponent', () => {
  let fixture: ComponentFixture<ButtonComponent>;
  let component: ButtonComponent;
  let loader: HarnessLoader;

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [ButtonComponent]
    });

    fixture = TestBed.createComponent(ButtonComponent);
    component = fixture.componentInstance;
    loader = TestbedHarnessEnvironment.loader(fixture);
  });
});
```

Test suite setup

```
it('should display button text', async () => {  
  component.title = 'Hello Angular Material';  
  const buttonHarness = await loader.getHarness(MatButtonHarness);  
  expect(await buttonHarness.getText()).toBe('Hello Angular Material');  
});
```

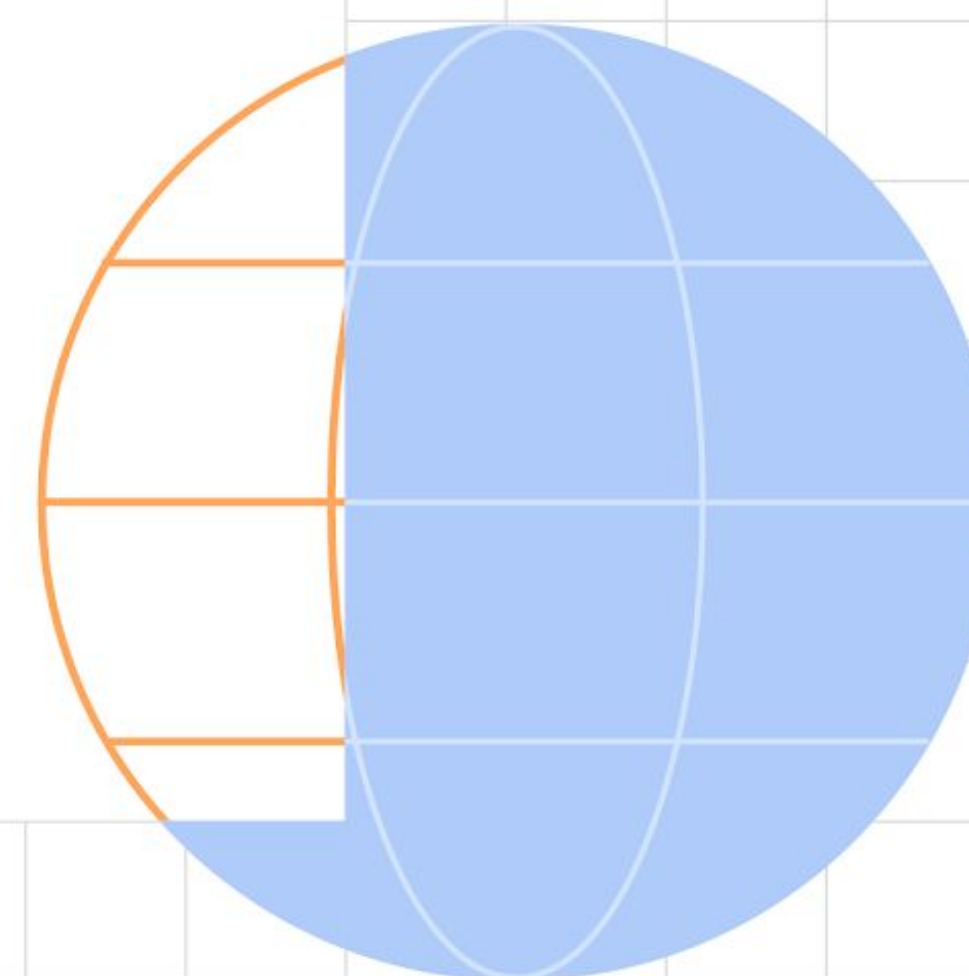
```
it('should display button text with predicate', async () => {  
  component.title = 'Click me';  
  const buttonHarness = await loader.getHarness(  
    MatButtonHarness.with({ text: 'Click me' })  
  );  
  expect(buttonHarness).not.toBeNull();  
});
```

Component tests



Harness cooking time 🍴

Let's write a component harness!



```
import { ComponentHarness } from '@angular/cdk/testing';

export class HelloWorldHarness extends ComponentHarness {
  static hostSelector = 'app-hello-world';

  getContainerElement = this.locatorFor('span');

  async getText() {
    const container = await this.getContainerElement();
    return container.text();
  }
}
```

Component harness

```
it('should display a greeting', async () => {  
  component.name = 'Angular Nation';  
  const nameHarness = await loader.getHarness>HelloWorldHarness);  
  expect(await nameHarness.getText()).toEqual('Angular Nation');  
});
```

```
it('should display a greeting with predicate', async () => {  
  component.name = 'Angular Nation with predicate';  
  const nameHarness = await loader.getHarness(  
   >HelloWorldHarness.with({ text: 'Angular Nation with predicate' })  
  );  
  expect(nameHarness).not.toBeNull();  
});
```

Component tests

```
import { BaseHarnessFilters, HarnessPredicate } from '@angular/cdk/testing';

interface HelloWorldHarnessFilters extends BaseHarnessFilters {
  text?: string | RegExp;
}

static with(options: HelloWorldHarnessFilters): HarnessPredicate<HelloWorldHarness> {
  return new HarnessPredicate(HelloWorldHarness, options)
    .addOption('text', options.text,
      (harness, text) => HarnessPredicate.stringMatches(harness.getText(), text));
}
```

Component harness predicate

Recap

Go write your unit tests with Angular CDK!

- How to properly test an Angular component
- What is component harness
- Test Angular Material components with harness
- Create test infrastructure for your Angular library

Useful links

Resources about testing and component harness

- <https://stackblitz.com/edit/angular-testing-cdk>
- <https://angular.io/guide/testing-components-basics>
- <https://material.angular.io/cdk/test-harnesses>
- <https://material.angular.io/guide/using-component-harnesses>

Google Developers



Thank You!



Aristeidis Bampakos
GDE Greece
@abampakos