ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

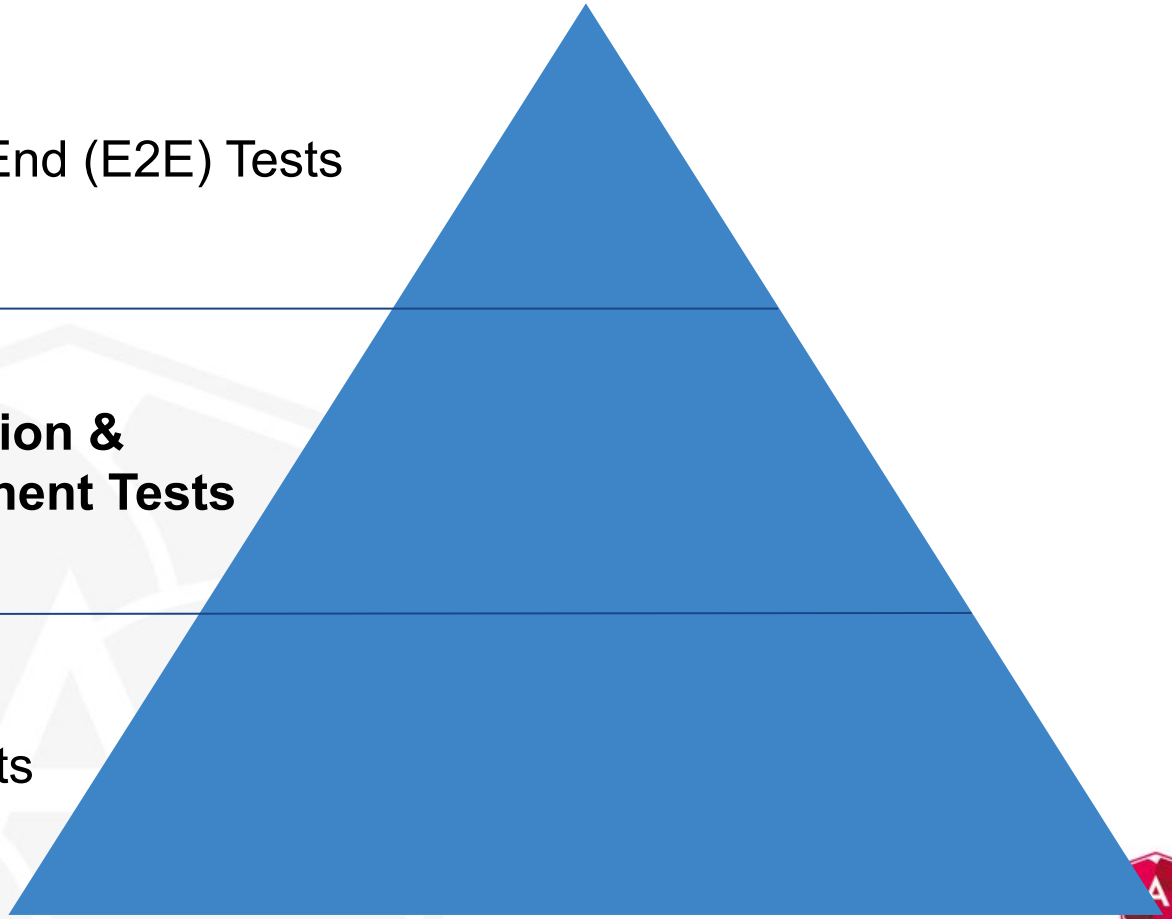
Angular Testing

3 - Component & Integration Tests Basic

End-to-End (E2E) Tests

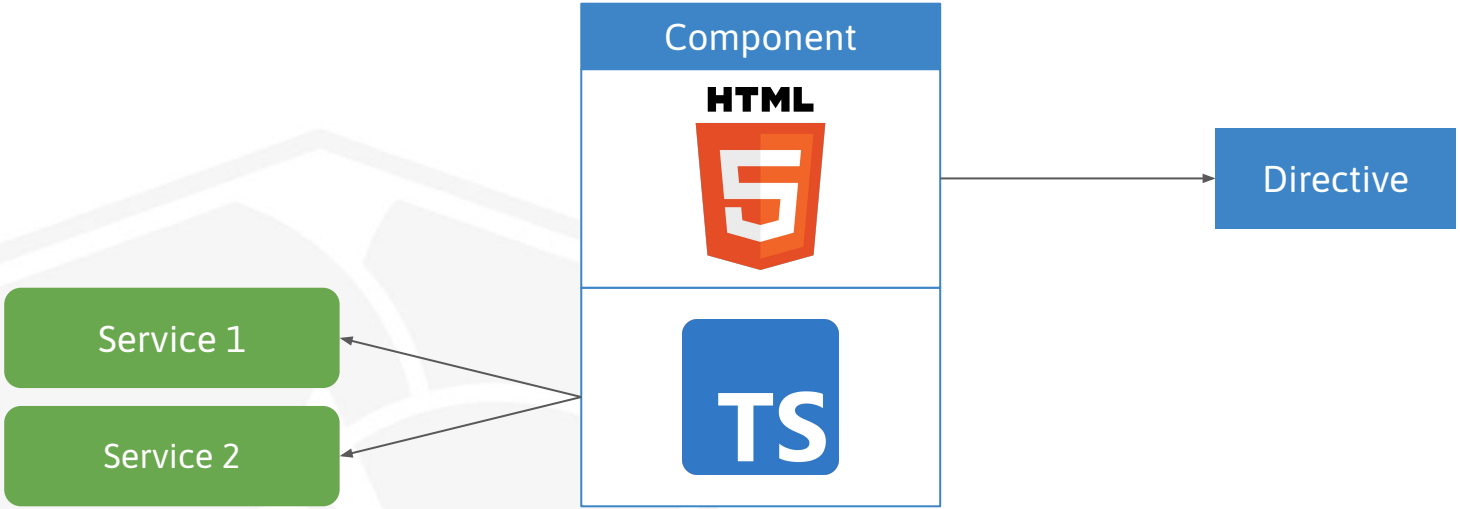
Integration & Component Tests

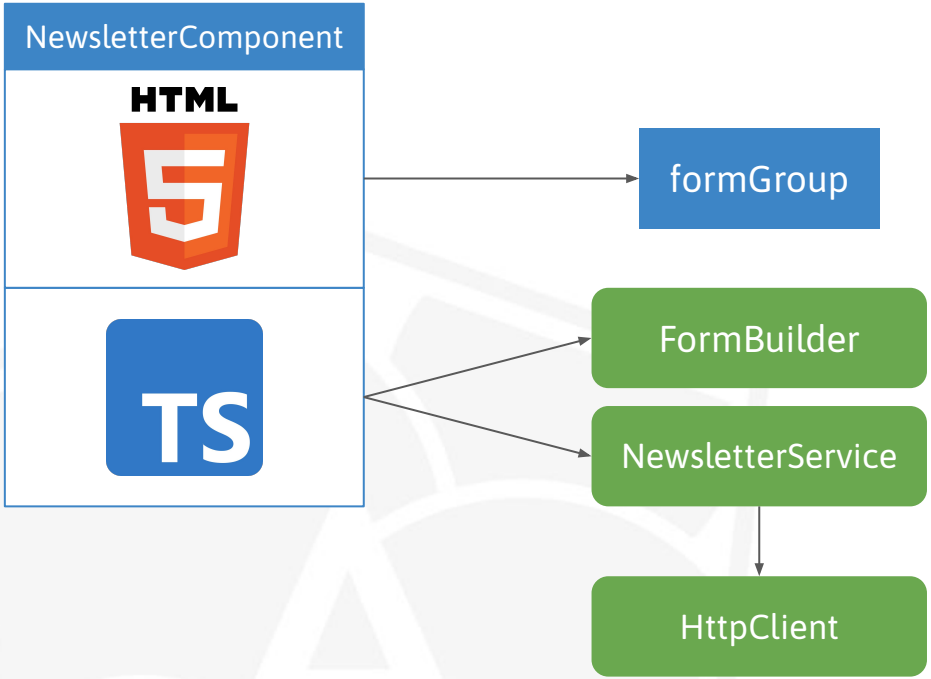
Unit Tests

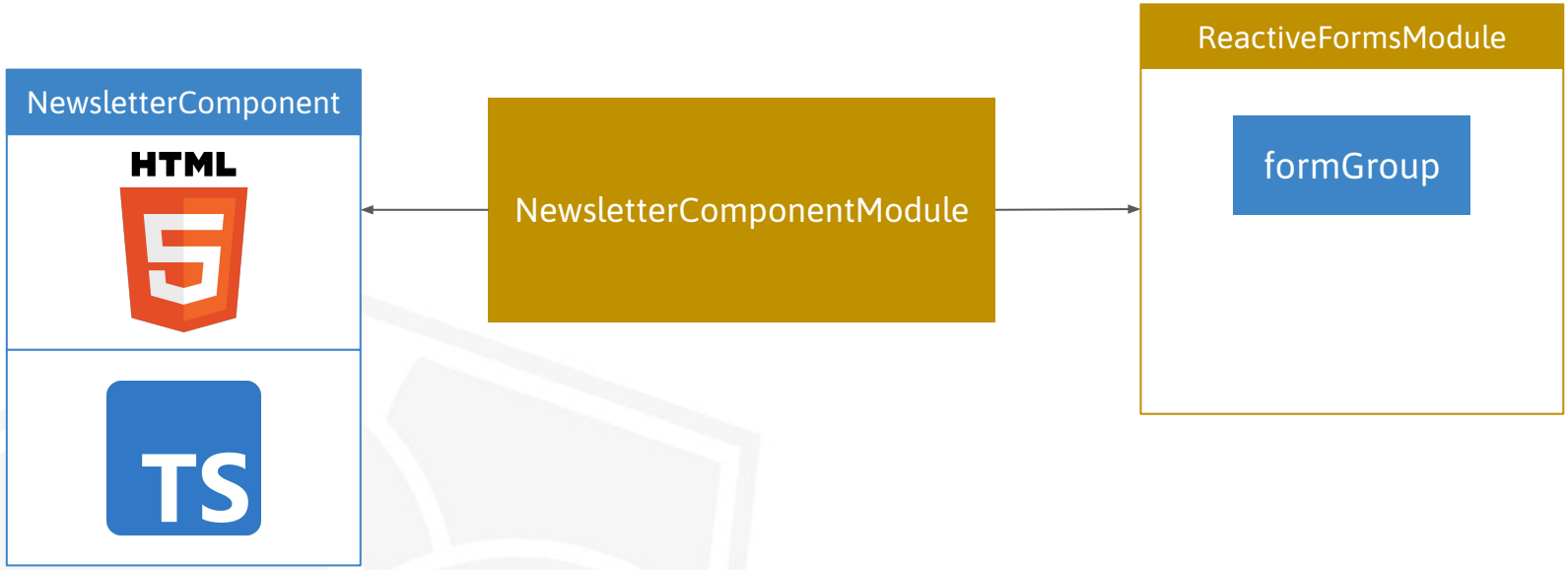


ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Configuring the TestingModule

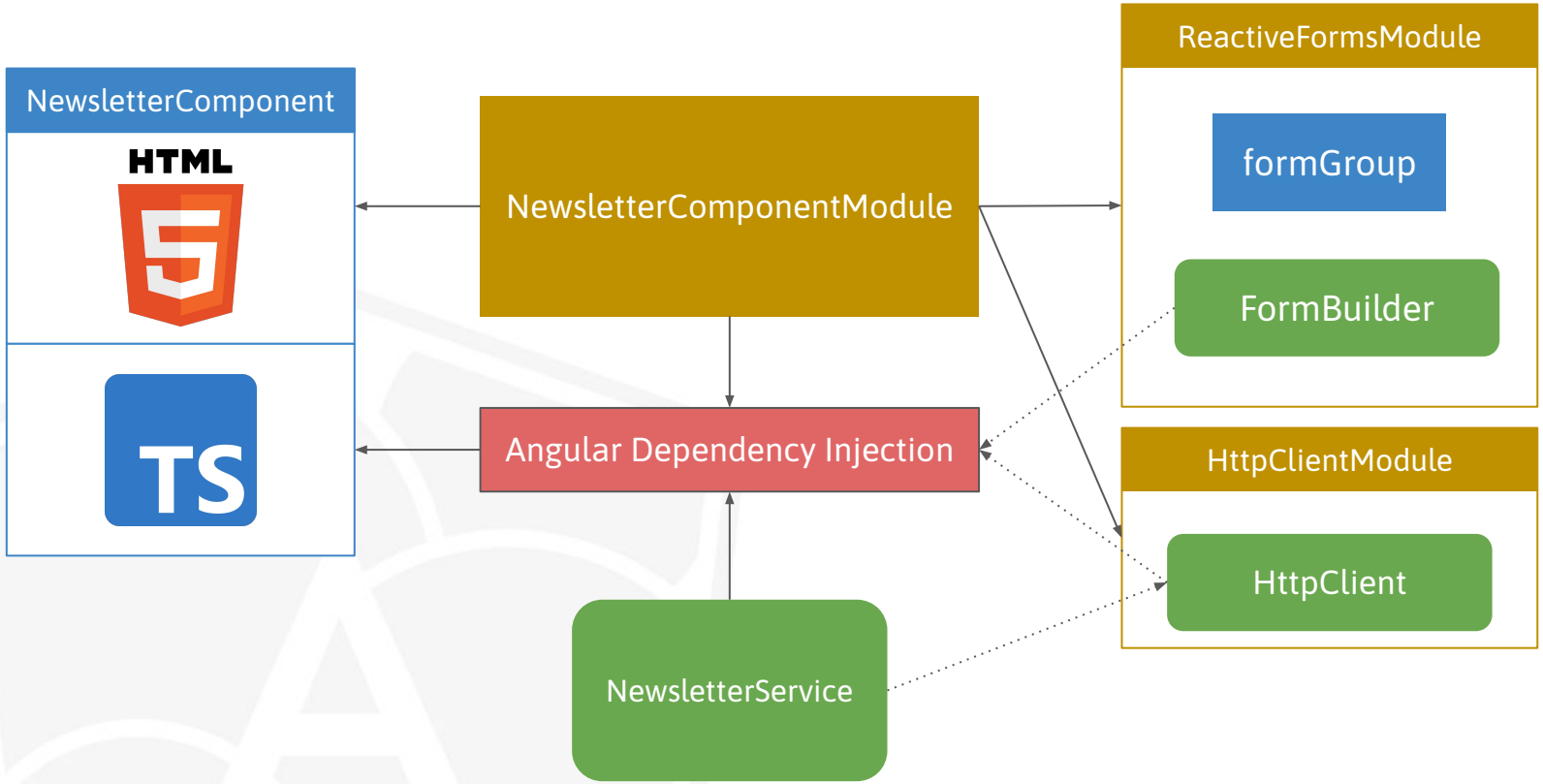






Different Ways of providing Services

1. @Injectable({providedIn: 'root'})
2. `providers` property of an NgModule
3. `providers` property of a Component/Pipe/Directive

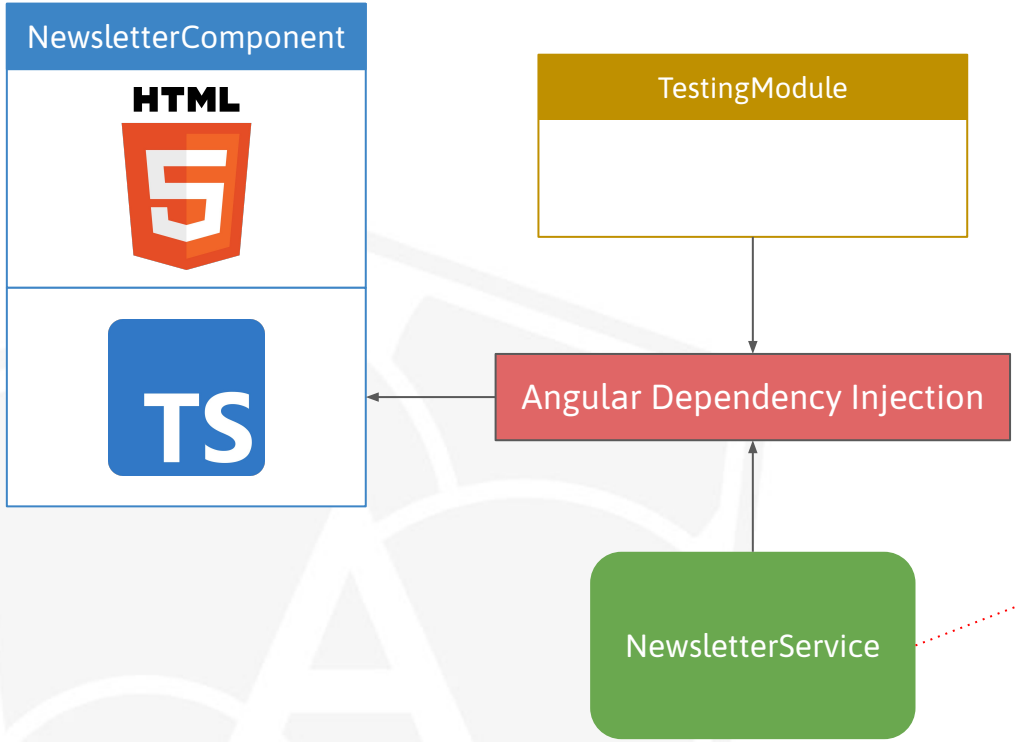


ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Testing Module

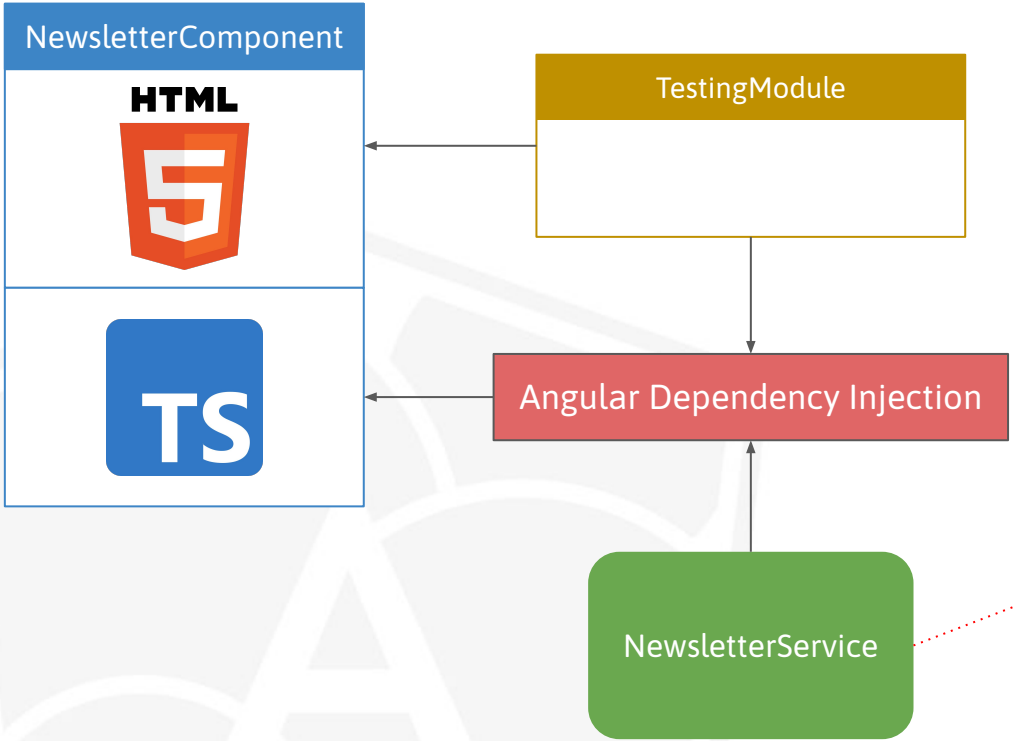
```
const fixture = TestBed.configureTestingModule({})
```


ANGULAR ARCHITECTS
INSIDE KNOWLEDGE



Testing Module

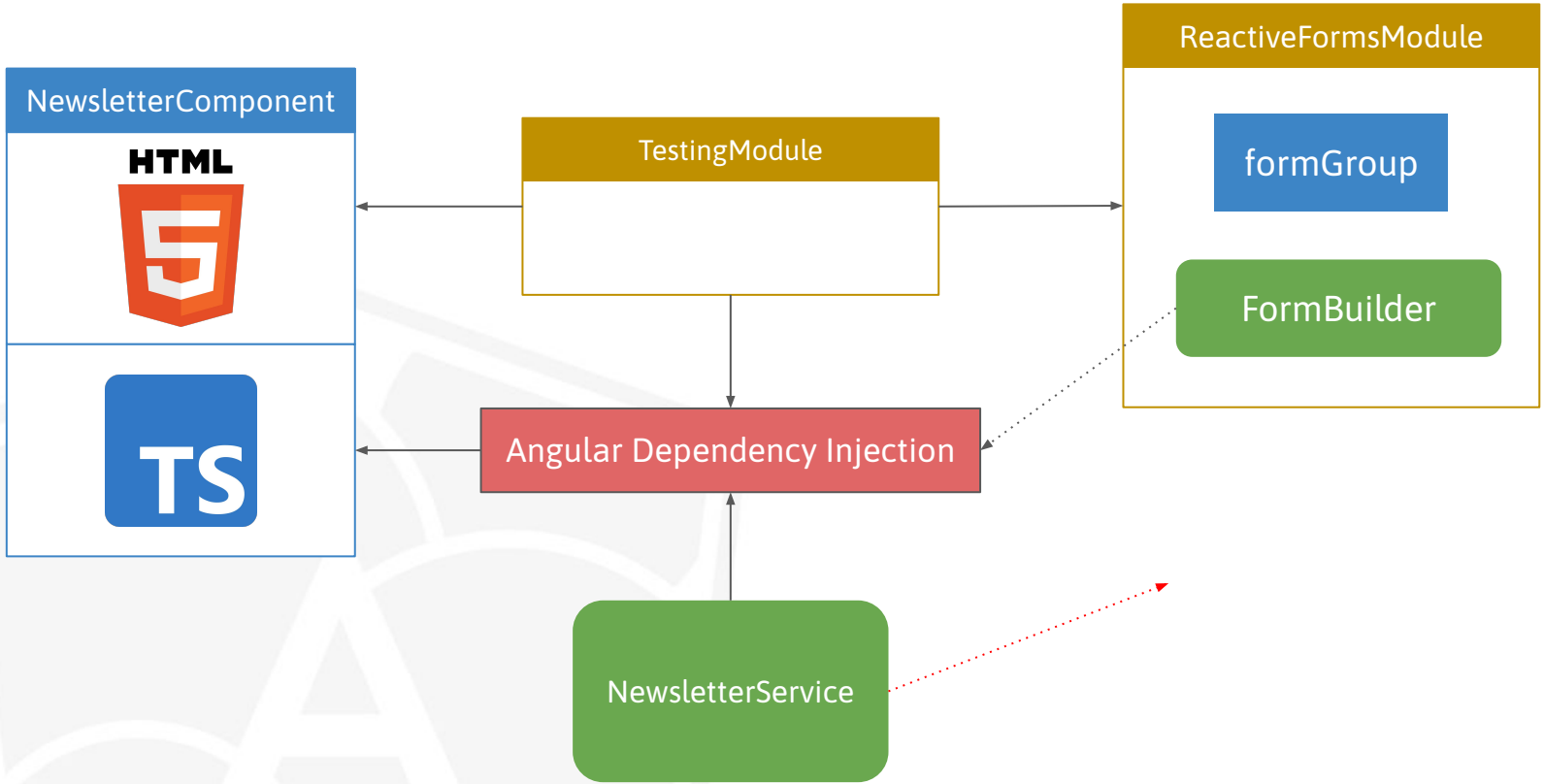
```
const fixture = TestBed.configureTestingModule({

  declarations: [NewsletterComponent]

})
```

Testing Module

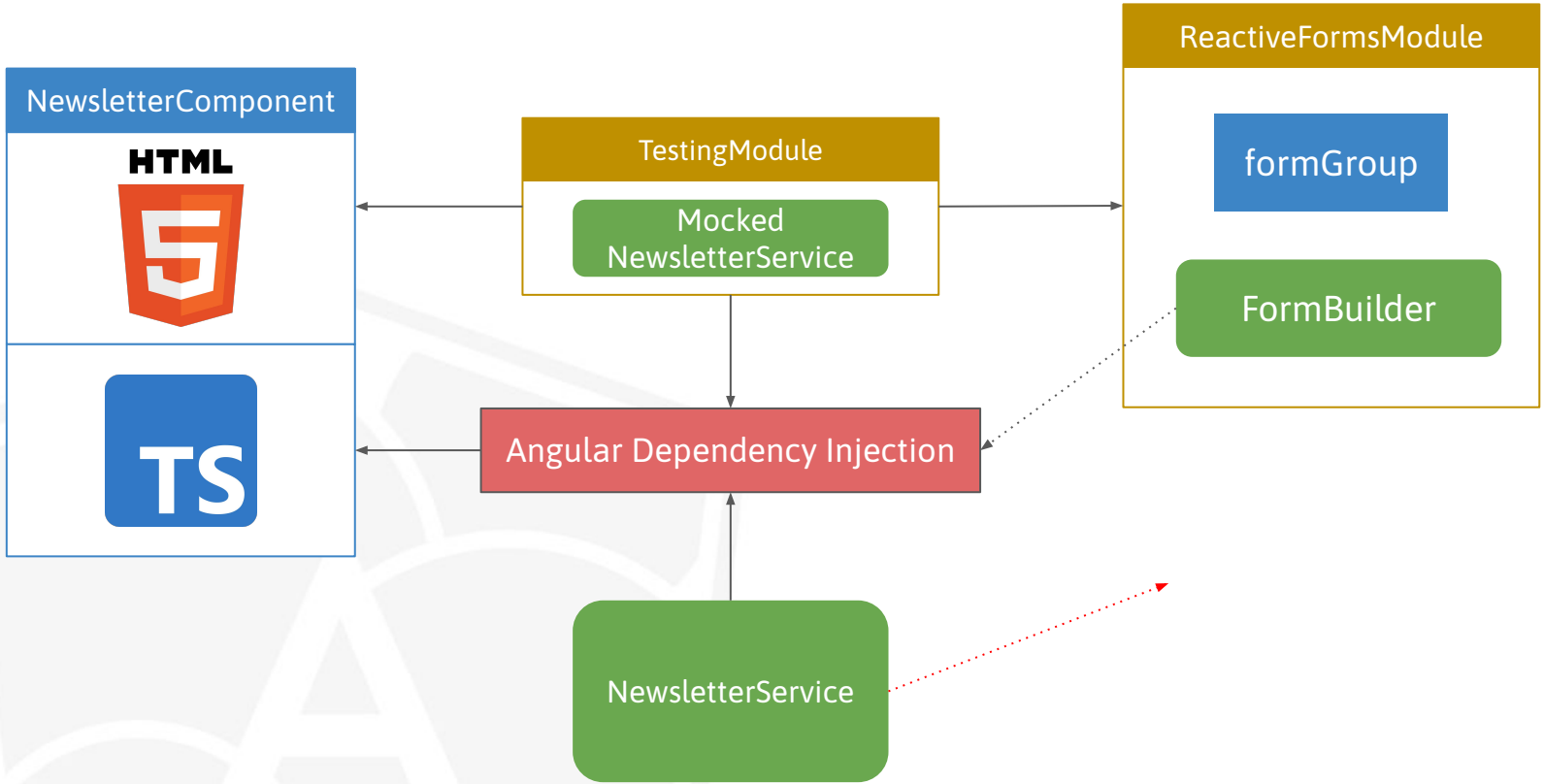
```
const fixture = TestBed.configureTestingModule({
  declarations: [NewsletterComponent],
  imports: [ReactiveFormsModule]
})
```

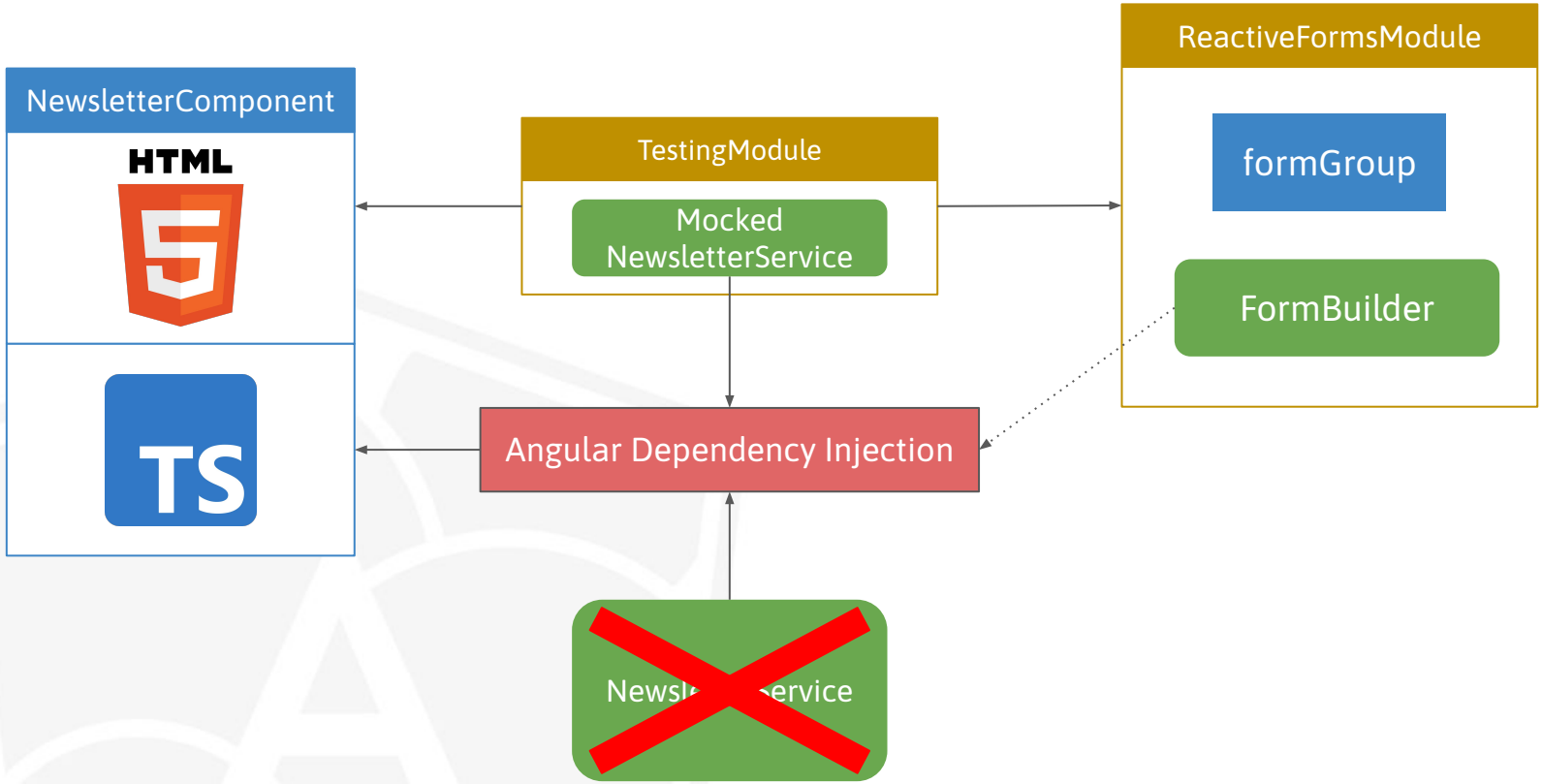
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



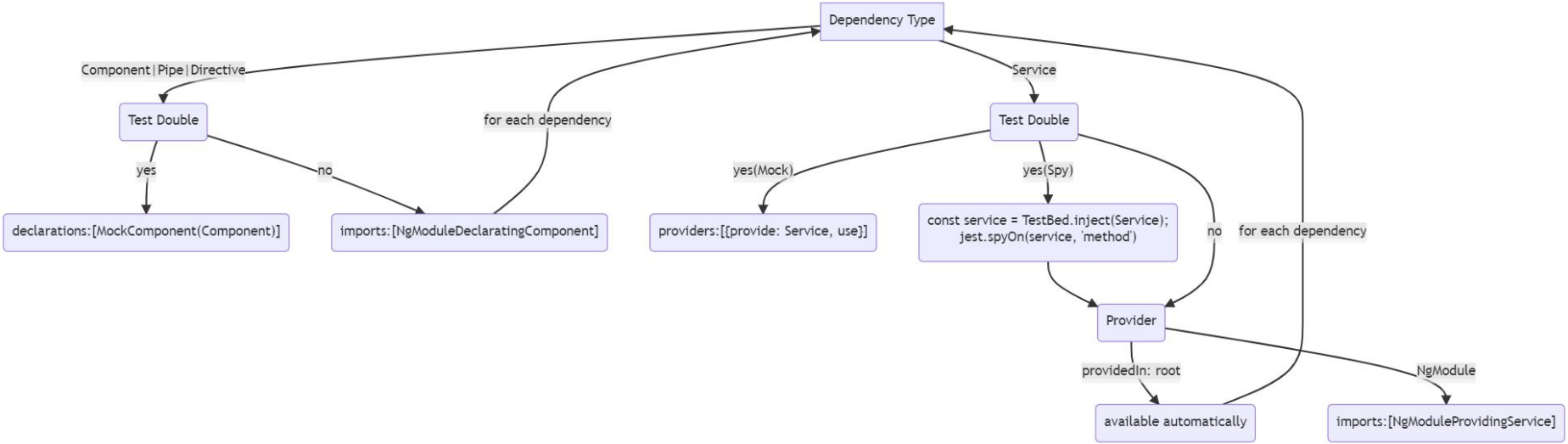
Adding Dependency Injection

```
const fixture = TestBed.configureTestingModule({
  declarations: [NewsletterComponent],
  providers: [{ provide: NewsletterService, useValue: null }]
})
```





Decision Tree: TestingModule



ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Mocking Components

1. "Three Monkeys"
2. Component Stubs
3. ng-mocks
4. Don't mock!

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Mocking Components - Three Monkeys

```
it('should mock components in 🙈🙉🙊 style, () => {
  const fixture = TestBed.configureTestingModule({
    declarations: [RequestInfoComponent],
    schemas: [NO_ERRORS_SCHEMA]
  }).createComponent(RequestInfoComponent);
  fixture.detectChanges();
  expect(true).toBe(true);
});
```

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Mocking Components - Manually

```
it('should stub the components', () => {
  @Component({ selector: 'mat-form-field', template: '' })
  class MatFormField {}

  const fixture = TestBed.configureTestingModule({
    declarations: [RequestInfoComponent, MatFormField],
    imports: [ReactiveFormsModule]
  }).createComponent(RequestInfoComponent);

  fixture.detectChanges();
  expect(true).toBe(true);
});
```

Mocking Components - ng-mocks

```
it('should stub the components', () => {
  const fixture = TestBed.configureTestingModule({
    declarations: [RequestInfoComponent, MockComponent(MatFormField)],
    imports: [ReactiveFormsModule]
  }).createComponent(RequestInfoComponent);

  fixture.detectChanges();
  expect(true).toBe(true);
});
```

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Mocking Components - Don't

```
it('should import the modules', () => {
  const fixture = TestBed.configureTestingModule({
    declarations: [RequestInfoComponent],
    imports: [ReactiveFormsModule, MatFormFieldModule, MatHintModule, MatLabelModule]
}).createComponent(RequestInfoComponent);

  fixture.detectChanges();

  expect(true).toBe(true);
});
```

HttpTest

```
it("should use Angular's http mock", () => {
  TestBed.configureTestingModule({
    declarations: [RequestInfoComponent],
    imports: [ReactiveFormsModule, HttpClientTestingModule],
  });
  const fixture = TestBed.createComponent(RequestInfoComponent);
  // queries for DOM Elements button and messageBox
  const httpController = TestBed.inject(HttpTestingController);

  button.click()

  const request = httpController.expectOne((req) => !!req.url.match(/nominatim/));
  request.flush([{ street: "Domgasse", streetNumber: 5 }]);
  expect(lookupResult.textContent).toBe("Address found");
});
```

Instead of HttpClientModule

Runs AFTER http request

RoutingTest

- RouterTestingModule provides routing functionality for tests
- Location can verify the expected url
- RoutingConfiguration is required

DOM Interaction

Interacting with the DOM

```
fixture.detectChanges();

const button = fixture.debugElement

  .query(By.css('button'))

  .nativeElement as HTMLButtonElement;

button.click();
```

Remember fakeAsync?

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Proper Handling of Event Listeners

```
it("should trigger search on click", fakeAsync(() => {
  fixture.detectChanges();
  const button = fixture.debugElement.query(By.css("button"))
    .nativeElement as HTMLButtonElement;
  button.click();
  tick();

  expect(lookuper.lookup).toHaveBeenCalled();
}));
```

Snapshot

- Serialization Feature
- Good fit when it comes to rough template checks
- No CSS
- Is not an image, e.g. png or jpg.

Lab Time