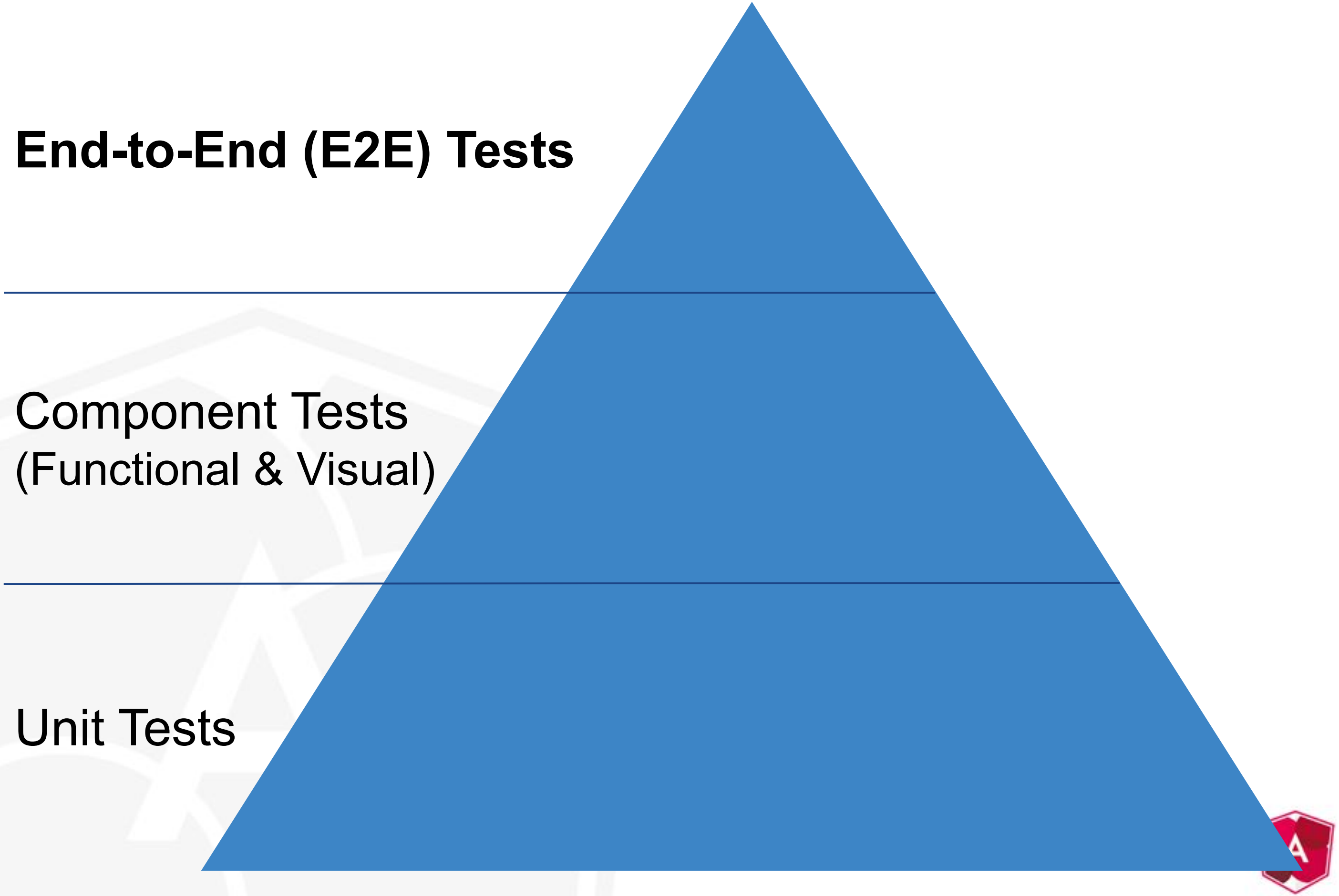
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Testing

End to End Tests



End-to-End (E2E) Tests

Component Tests
(Functional & Visual)

Unit Tests

2 Core Technologies for E2E Frameworks

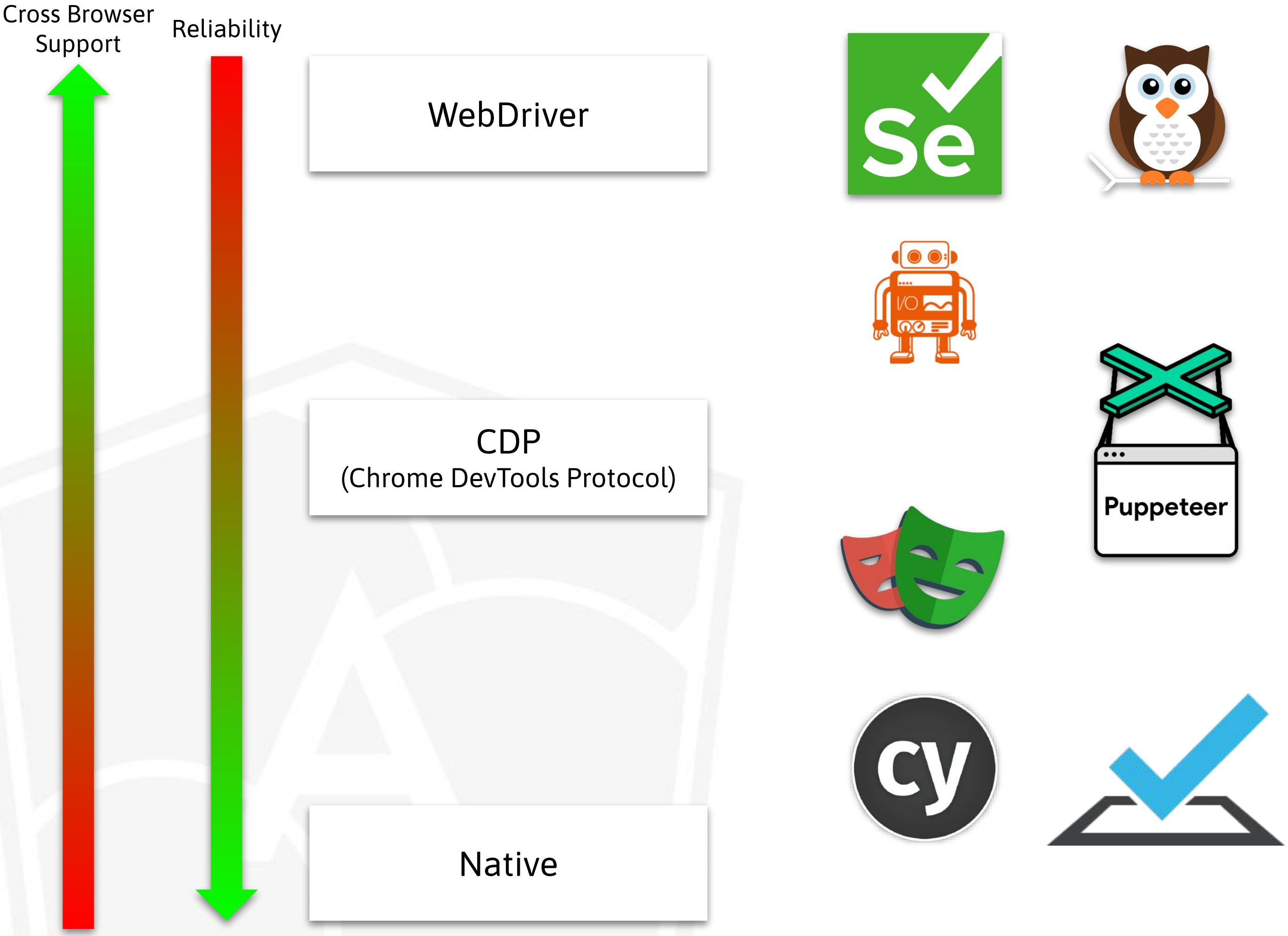
1. WebDriver
2. Chrome DevTool Protocol (CDP)

WebDriver

- W3C Standard
- Cross-Browser Support
- Known to be flaky
- WebDriver BiDi as successor
- Popular E2E Tools
  - Protractor
  - Selenium
  - NightWatch
  - WebDriverIO

CDP (Chrome DevTools Protocol)

- Debugging Tool for Chromium-based Browsers
- Puppeteer as primary library
- Playwright incl. Safari
- Much more stable
- Constrained to CDP-Browsers



Cross Browser Support

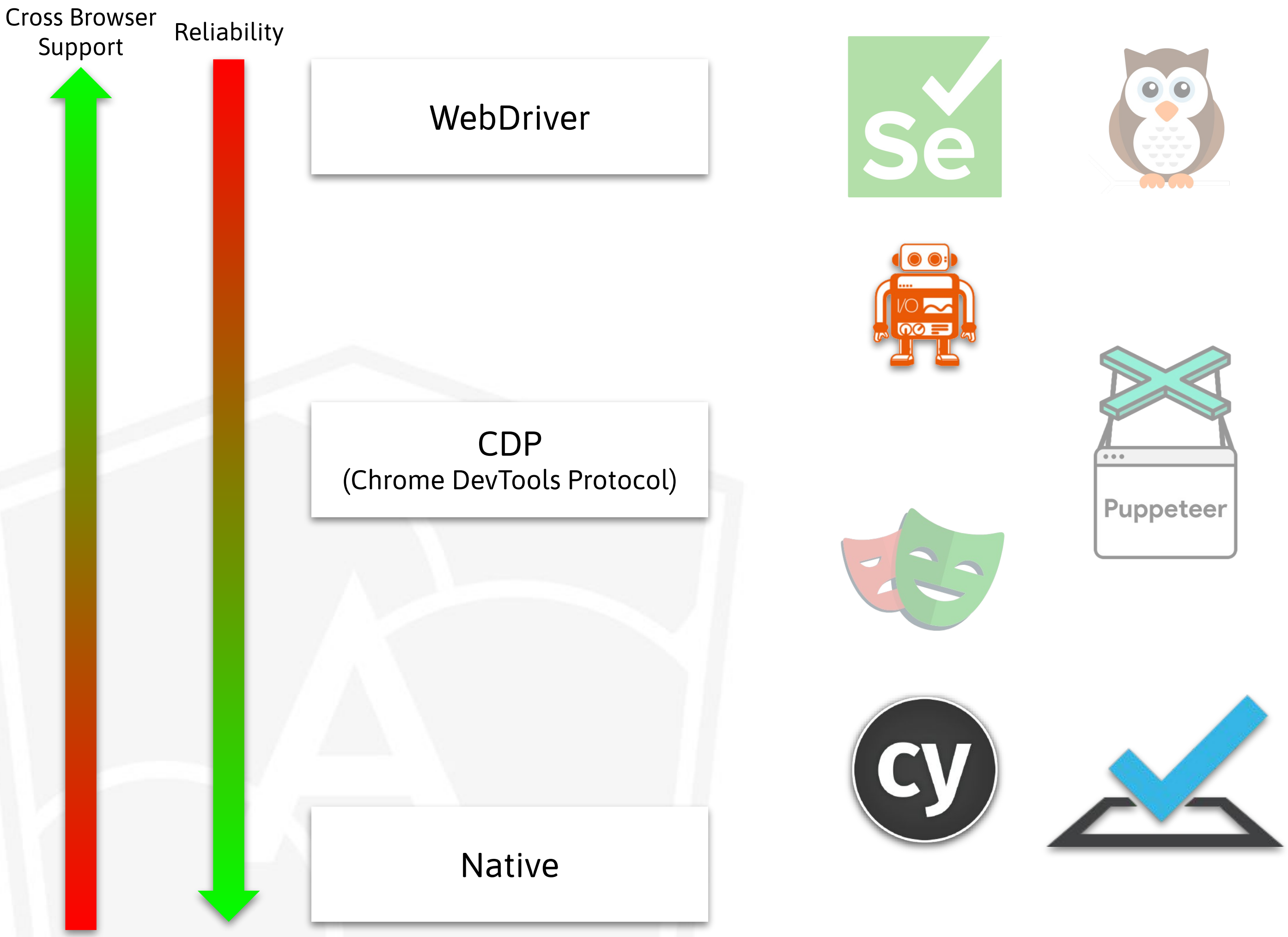
Reliability

WebDriver

CDP
(Chrome DevTools Protocol)

Native

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE



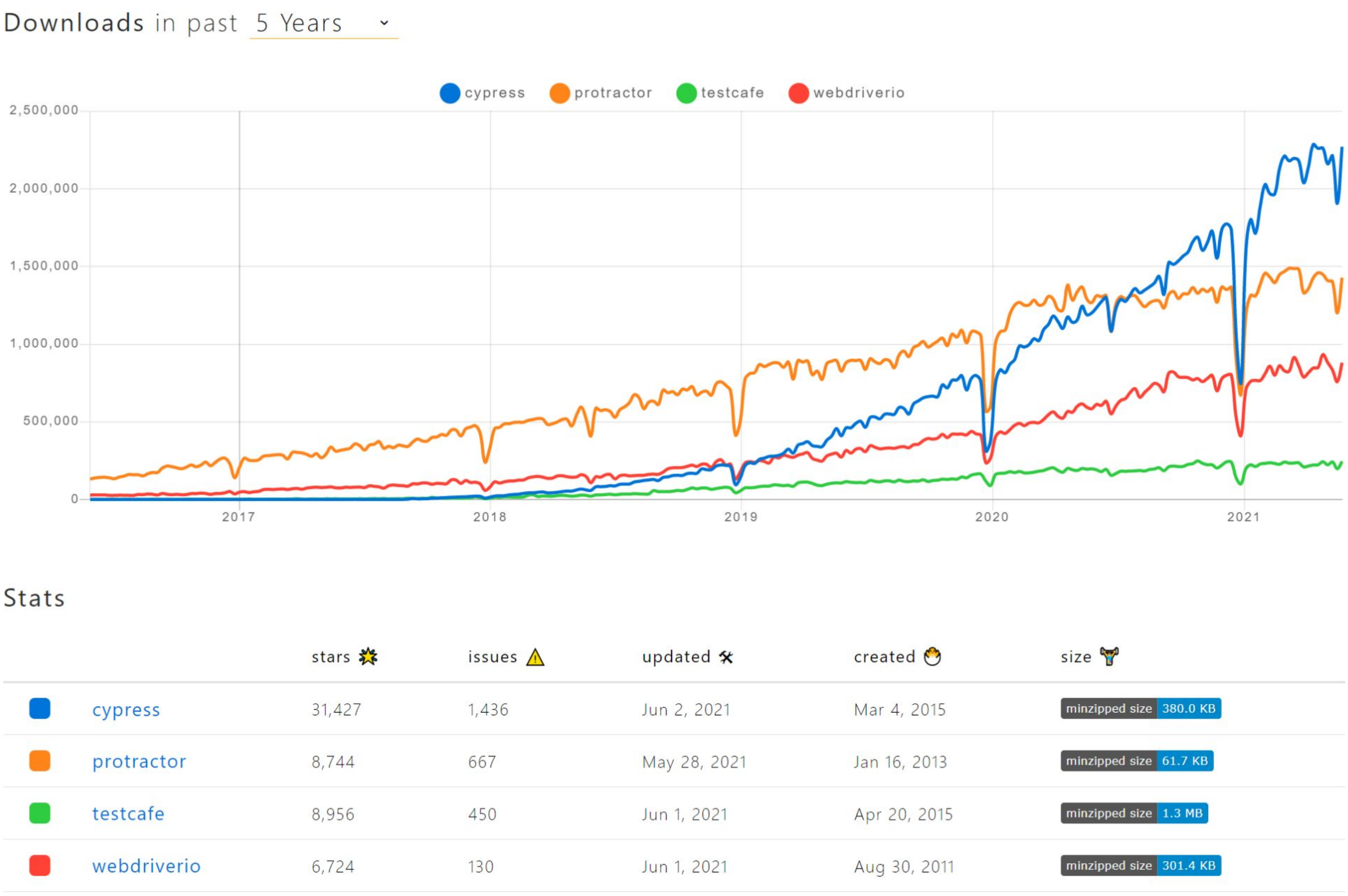
Cross Browser Support

Reliability

WebDriver

CDP
(Chrome DevTools Protocol)

Native



Downloads in past 5 Years ⌄

Stats

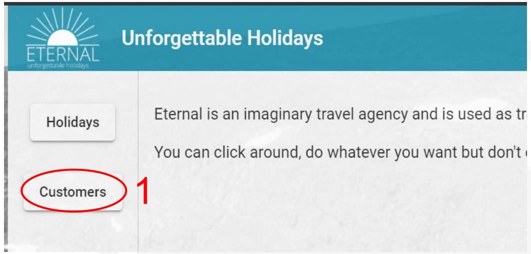
		stars 🌟	issues ⚠	updated 🛠	created 🐣	size 🏋
🟦	cypress	31,427	1,436	Jun 2, 2021	Mar 4, 2015	minzipped size 380.0 KB
🟧	protractor	8,744	667	May 28, 2021	Jan 16, 2013	minzipped size 61.7 KB
🟩	testcafe	8,956	450	Jun 1, 2021	Apr 20, 2015	minzipped size 1.3 MB
🟥	webdriverio	6,724	130	Jun 1, 2021	Aug 30, 2011	minzipped size 301.4 KB

Cypress

- Node.js Application
- Great Developer Experience
- Good Documentation
- Easy Setup
- Internal "IDE"
- CI Features like Videorecording and Screenshots

ANGULAR ARCHITECTS INSIDE KNOWLEDGE

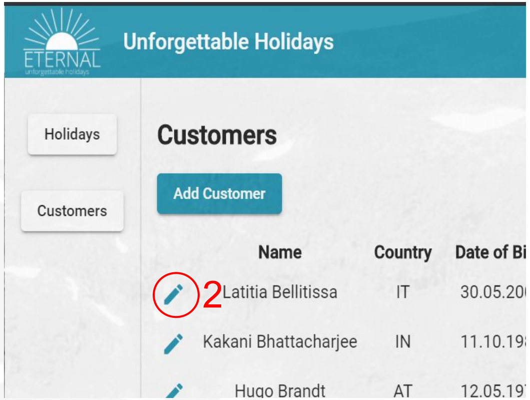
The Await Feature



Eternal is an imaginary travel agency and is used as tr

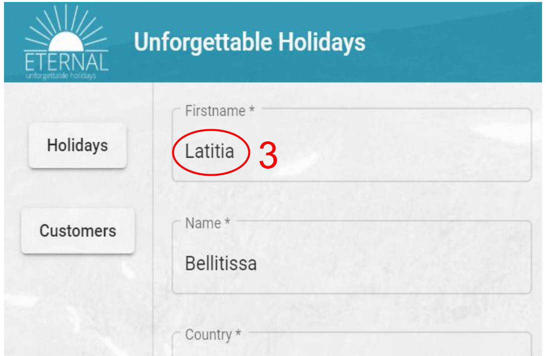
You can click around, do whatever you want but don't

1. Select Customers



Name	Country	Date of Bi
Latitia Bellitissa	IT	30.05.20
Kakani Bhattacharjee	IN	11.10.19
Hugo Brandt	AT	12.05.19

2. Select Customer

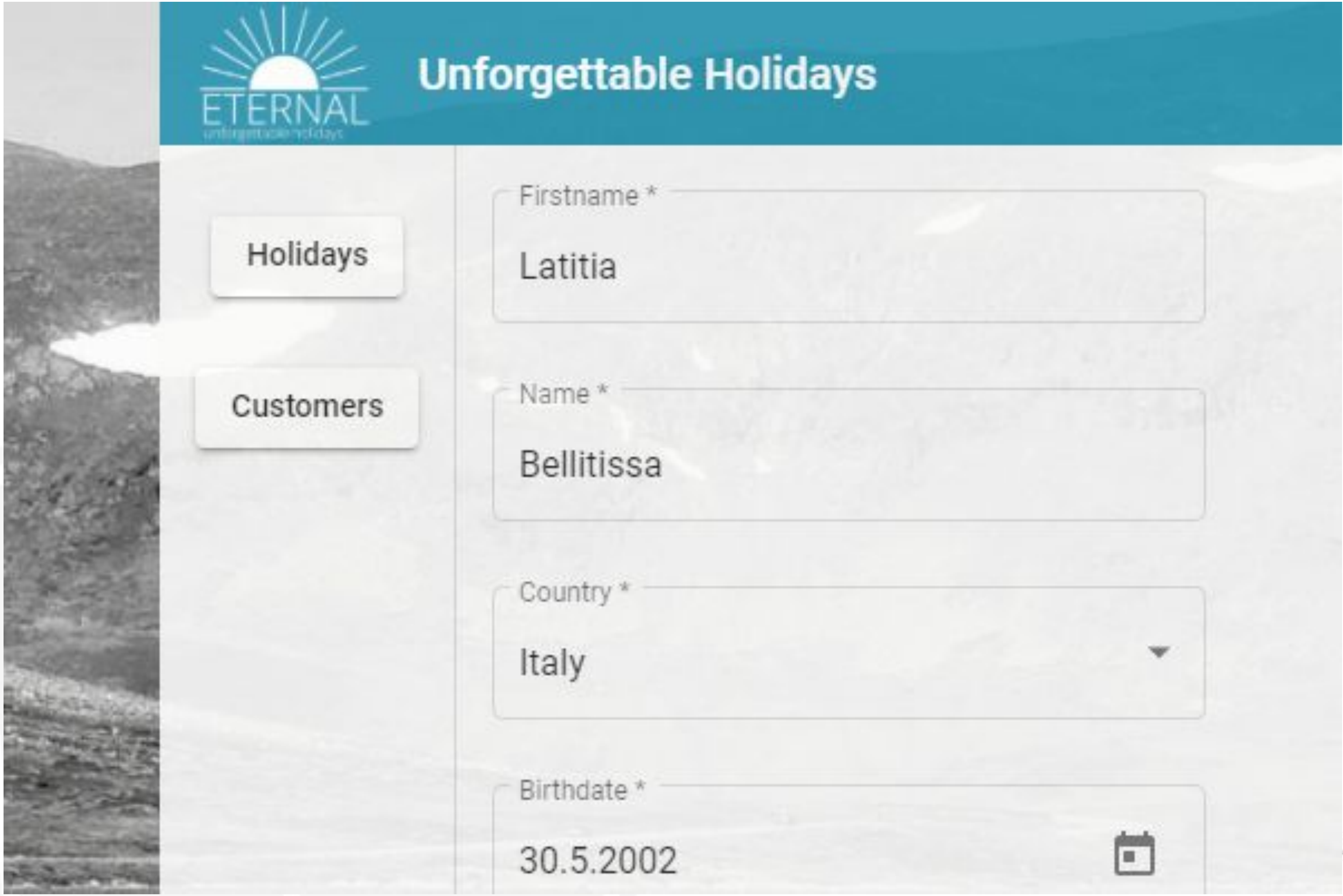


Firstname *
Latitia

Name *
Bellitissa

Country *

3. Edit Firstname



Unforgettable Holidays

Firstname *
Latitia

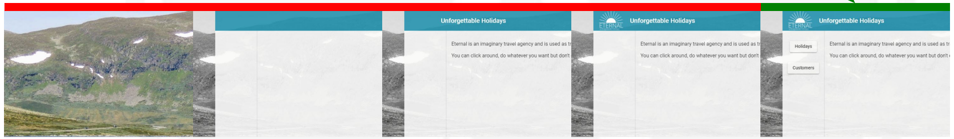
Name *
Bellitissa

Country *
Italy

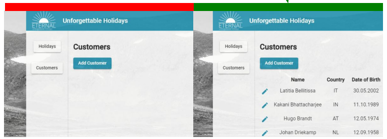
Birthdate *
30.5.2002

Non-Waiting Style

1. Select Customers


Unforgettable Holidays
Eternal is an imaginary travel agency and is used as tr
You can click around, do whatever you want but don't
Holidays
Customers

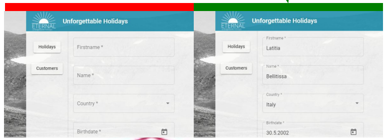
2. Select Customer



Customers

	Name	Country	Date of Birth
✎	Latitia Bellitissa	IT	30.05.2002
✎	Kakani Bhattacharjee	IN	11.10.1989
✎	Hugo Brandt	AT	12.05.1974
✎	Johan Driekamp	NL	12.09.1958

3. Edit Firstname



Firstname * Latitia

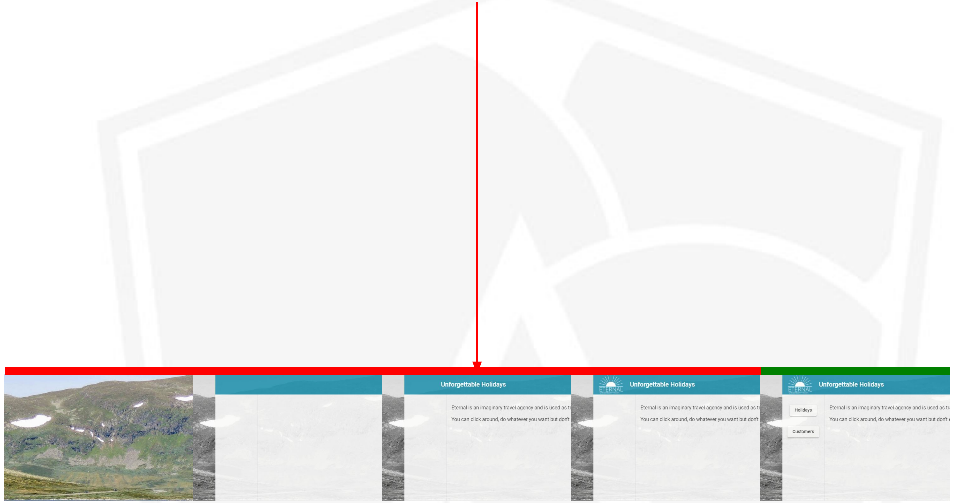
Name * Bellitissa

Country * Italy

Birthdate * 30.5.2002

Non-Waiting Style

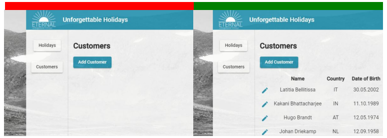
1. Select Customers



Eternal is an imaginary travel agency and is used as tr

You can click around, do whatever you want but don't

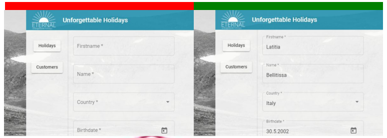
2. Select Customer



Customers

	Name	Country	Date of Birth
✎	Latitia Bellitissa	IT	30.05.2002
✎	Kakani Bhattacharjee	IN	11.10.1989
✎	Hugo Brandt	AT	12.05.1974
✎	Johan Driekamp	NL	12.09.1958

3. Edit Firstname



Firstname *

Name *

Country *

Birthdate *

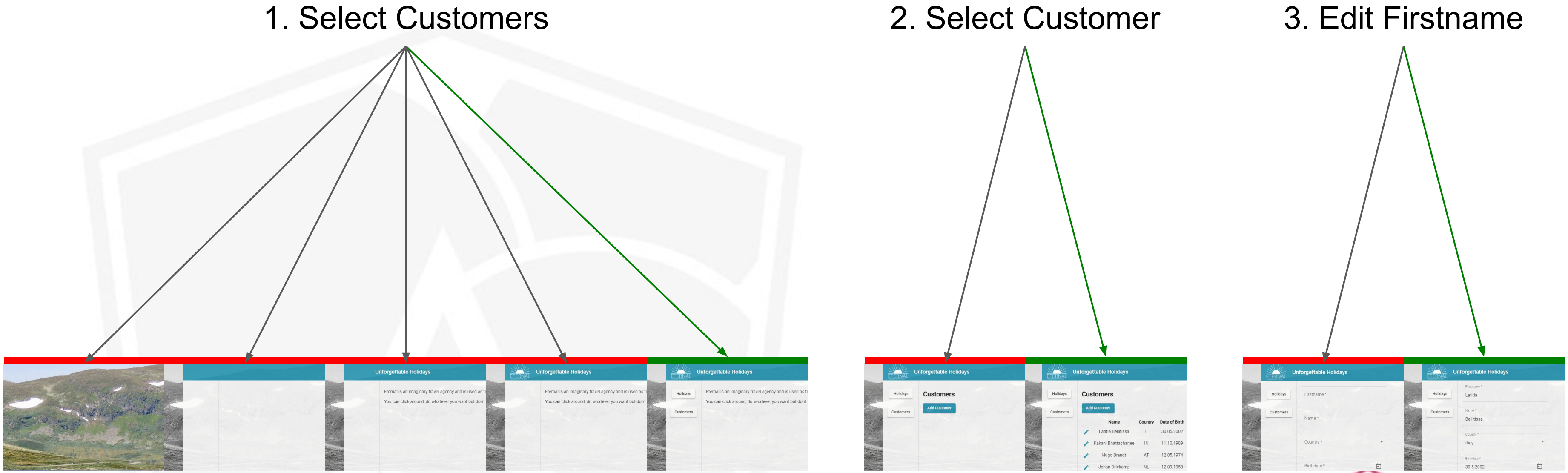
Firstname *: Latitia

Name *: Bellitissa

Country *: Italy

Birthdate *: 30.5.2002

Waiting Style



1. Select Customers

2. Select Customer

3. Edit Firstname

Eternal is an imaginary travel agency and is used as tr

You can click around, do whatever you want but don't

Customers

Name	Country	Date of Birth
Latitia Bellitissa	IT	30.05.2002
Kakani Bhattacharjee	IN	11.10.1989
Hugo Brandt	AT	12.05.1974
Johan Driekamp	NL	12.09.1958

Firstname *: Latitia

Name *: Bellitissa

Country *: Italy

Birthdate *: 30.5.2002

Basic Setup

- Cypress has no Angular integration
  - Autonomous setup
- yarn add -D cypress
- add tsconfig.json to newly cypress directory
- npx cypress open

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Commands

- Methods of "cy." object
- Run asynchronously
  - "Feels" like a Promise (thenable) but isn't
  - Don't use async/await
- Are chainable

Commands

- cy
  - .visit(url: string)
    - Can only be run at the beginning
    - Domain can't be changed
  - .get(selector)
    - Uses jQuery selectors
  - .contains
  - .click
  - .type

An E2E Test in Cypress

```
it('should rename Latitia to Laetitia', () => {

  cy.visit('');

  cy.get('mat-drawer a').contains('Customers').click();

  cy.get('div').contains('Latitia Bellitissa').siblings('.edit').click();

  cy.get('input:first').clear().type('Laetitia');

  cy.get('button[type=submit]').click();



  cy.get('div').contains('Bellitissa').should('have.text', 'Laetitia Bellitissa');

});
```

Assertions Implicit

- Behaves like a normal command
- Does waiting as well
- Good for single assertions

```
cy
  .get('h1')
  .should(
    'have.text',
    ' Unforgettable Holidays '
  );
```

Assertions Explicit

- More verbose
- Good for multiple assertions
- Massaging

```
cy.get('h1').should(($h1) => {

  expect($h1).to.have

    .text(' Unforgettable Holidays ');

});
```

Cypress Goodies

- lodash: Cypress._
- jQuery: Cypress.$
- ~~Moment: Cypress.moment~~
- Uses Mocha and Chai
  - Not Jest
  - Very similar commands

Demo

Flakiness / Unreliability

```
describe('Customers', () => {
  it('should add a customer', () => {
    cy.visit('');
    cy.get('a').contains('Customers').click();
    cy.get('a').contains('Add Customer').click();
  });
});
```

Flakiness / Unreliability

- Commands are not retried when they are successful
- Chains are multiple commands
- cy.get & cy.contains are commands

```
cy.get('a').contains('Add Customer').click();
    ↑           ↑
    1           2
```

Prevent Flakiness by Assertion

```
describe('Customers', () => {

  it('add customer', () => {

    cy.visit('');

    cy.get('a').contains('Customers').click();

    cy.get('a').should('contain', 'Add Customer');

    cy.get('a').contains('Add Customer').click();

  });

});
```

Prevent Flakiness by a better Selection

```
describe('Customers', () => {

  it('add customer', () => {

    cy.visit('');

    cy.get('mat-drawer a').contains('Customers').click();

    cy.get('mat-drawer-content a').contains('Add Customer').click();

  });

});
```

Prevent Flakiness by a much better Selection

```
describe('Customers', () => {
  it('add customer', () => {
    cy.visit('');
    cy.get('[data-test=btn-customers]').click();
    cy.get('[data-test=btn-add-customer]').click();
  });
});
```

Lab Time

Mocking Requests

```
cy.intercept('GET', '/holidays.json', {
  body: {
    holidays: [
      {
        title: 'Cambodia',
        teaser: 'Discover old temples and learn about the great Khmer Empire',
        imageUrl: 'https://eternal-app.s3.eu-central-1.amazonaws.com/assets/AngkorWatSmall.jpg',
        description:
          'Travel Siem Reap in Cambodia and visit the...'
      }
    ]
  }
});
```

3 Level Architecture

Tests

Page Object Models

Utility Functions

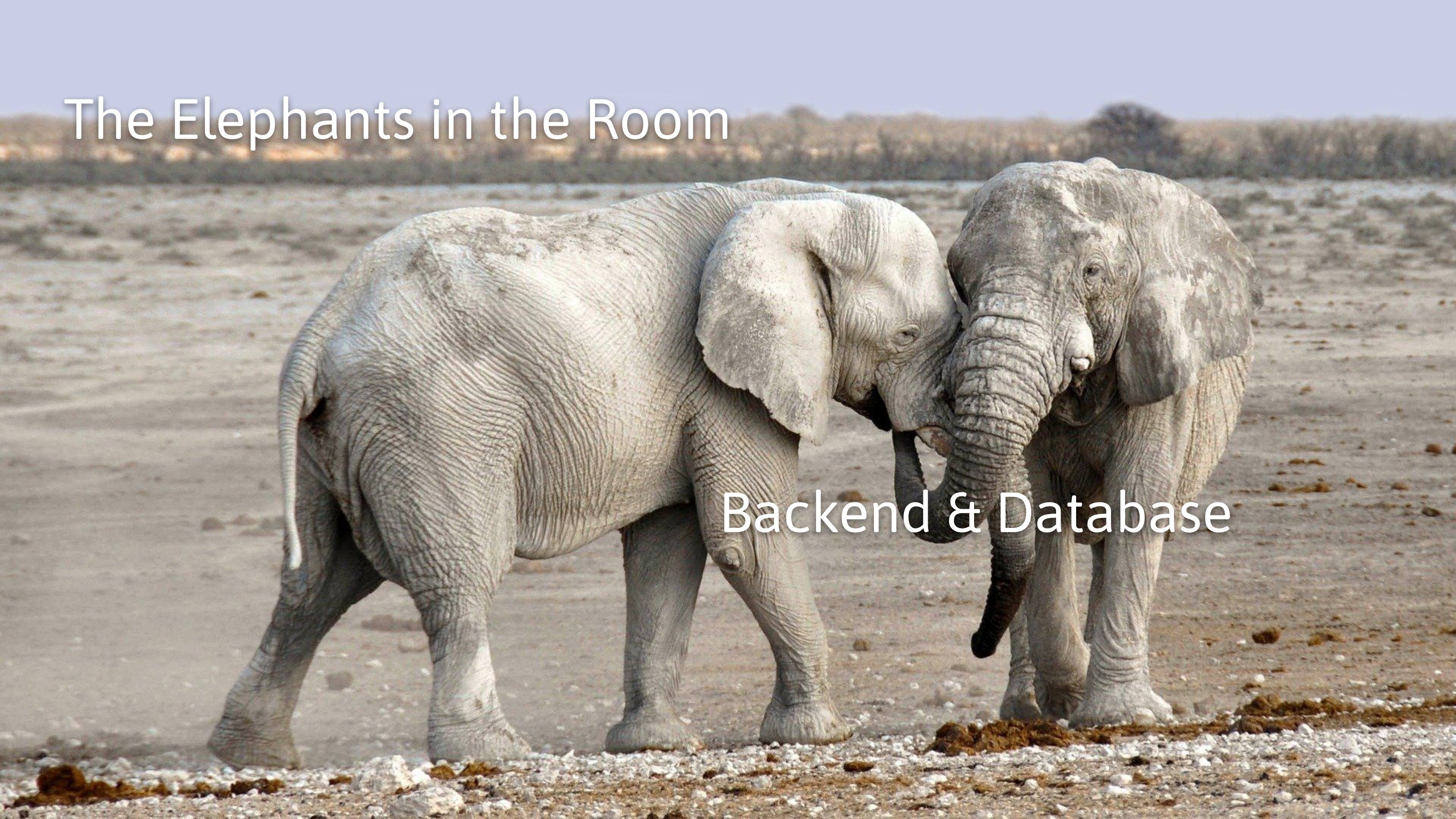
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Page Object Model

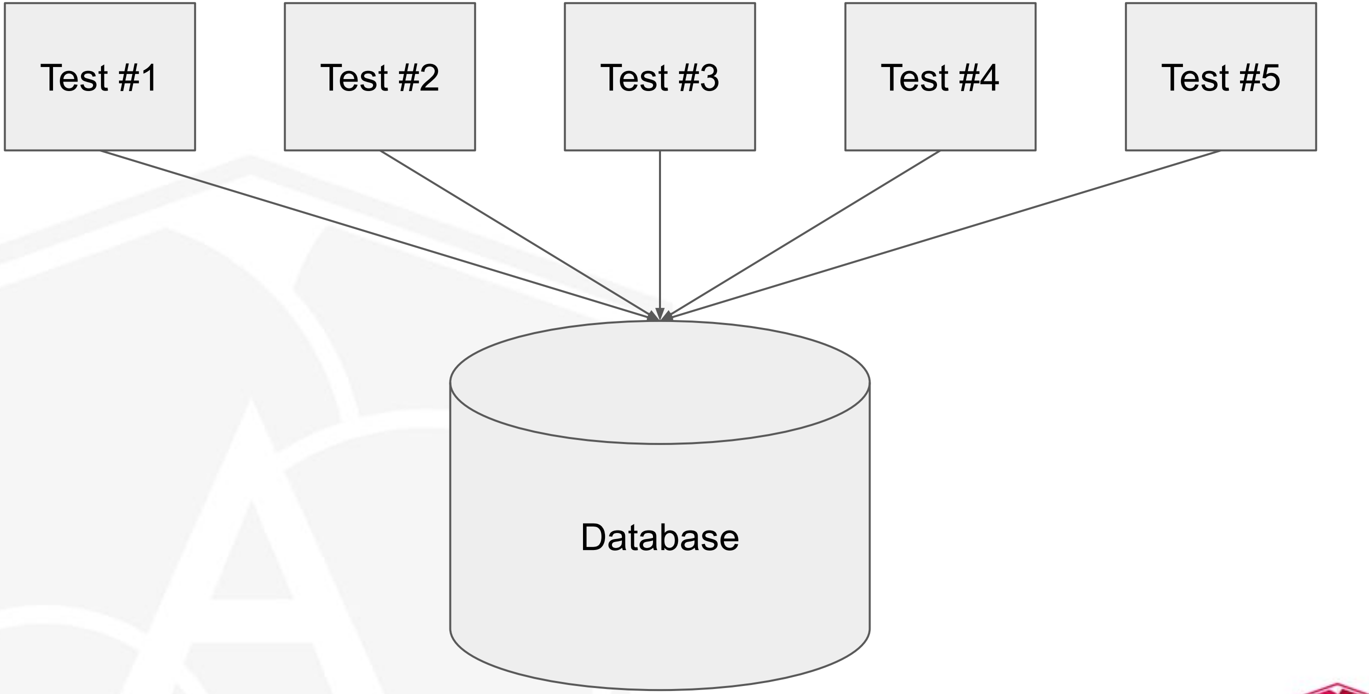
```
class Sidemenu {

  click(name: "Customers" | "Holidays"): Chainable {

    return cy.get("mat-drawer a").contains(name).click();

  }

}

export const sidemenu = new Sidemenu();
```

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

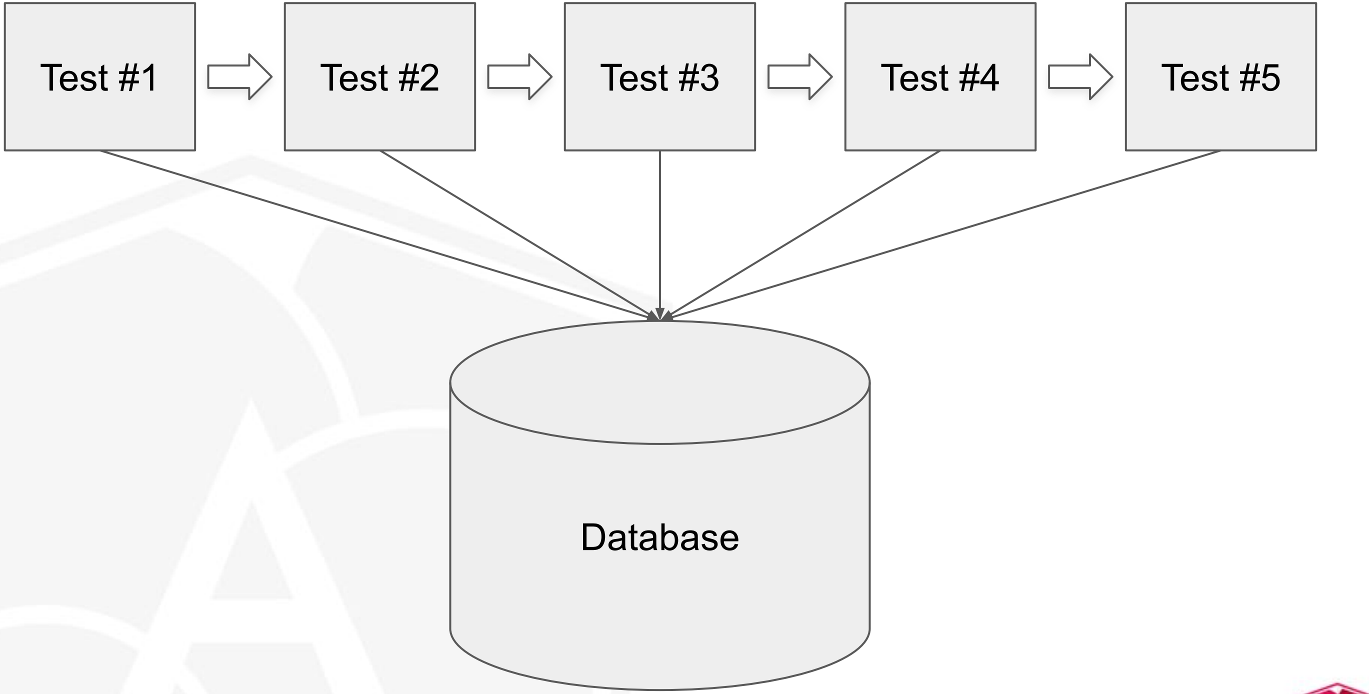
The Elephants in the Room



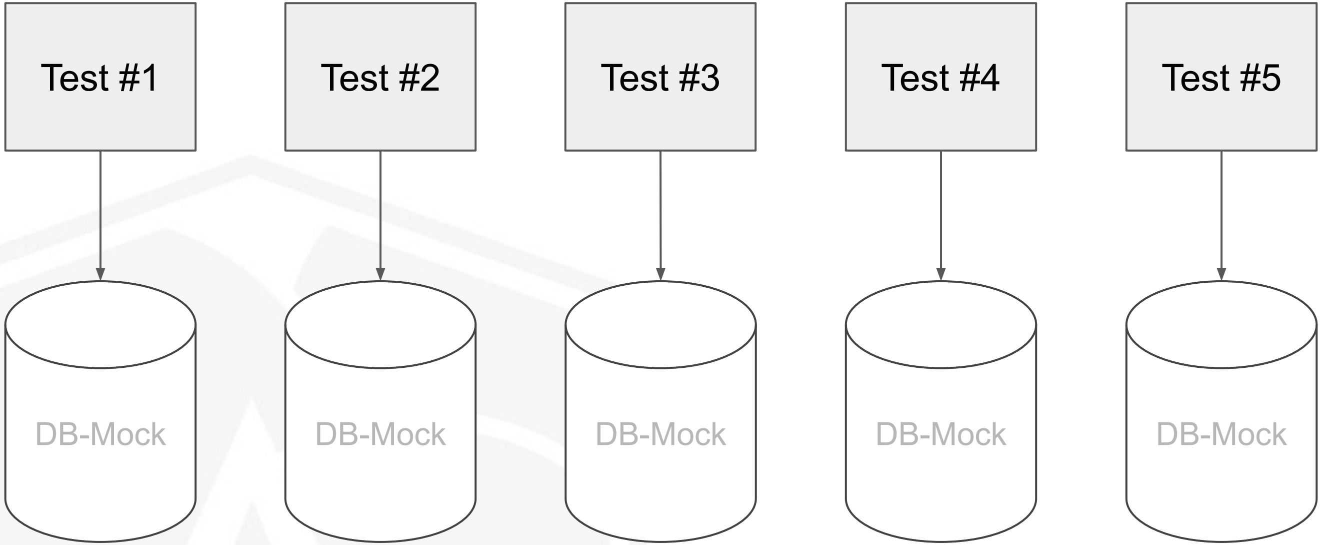
Backend & Database


Test #1
Test #2
Test #3
Test #4
Test #5
Database

Indirect "Transitive" Coupling



Copy Strategy from non-E2E???



Database in E2E Tests

Database in Non-E2E Tests

Test Seeded Database & done???

Issues with Test Seed

- "One size fits all" approach
- Tight Coupling → Not scalable
- Fast Reseeding not always possible
- Multiple Databases
- Data from External Systems → no Seeding possible

Individual Context - Best Case Scenario

- Data is referenced to a particular entity
  - User
  - Product
  - …
- Multi-Tenant Systems
- Customer-Centric Systems
  - Insurances
  - Banks

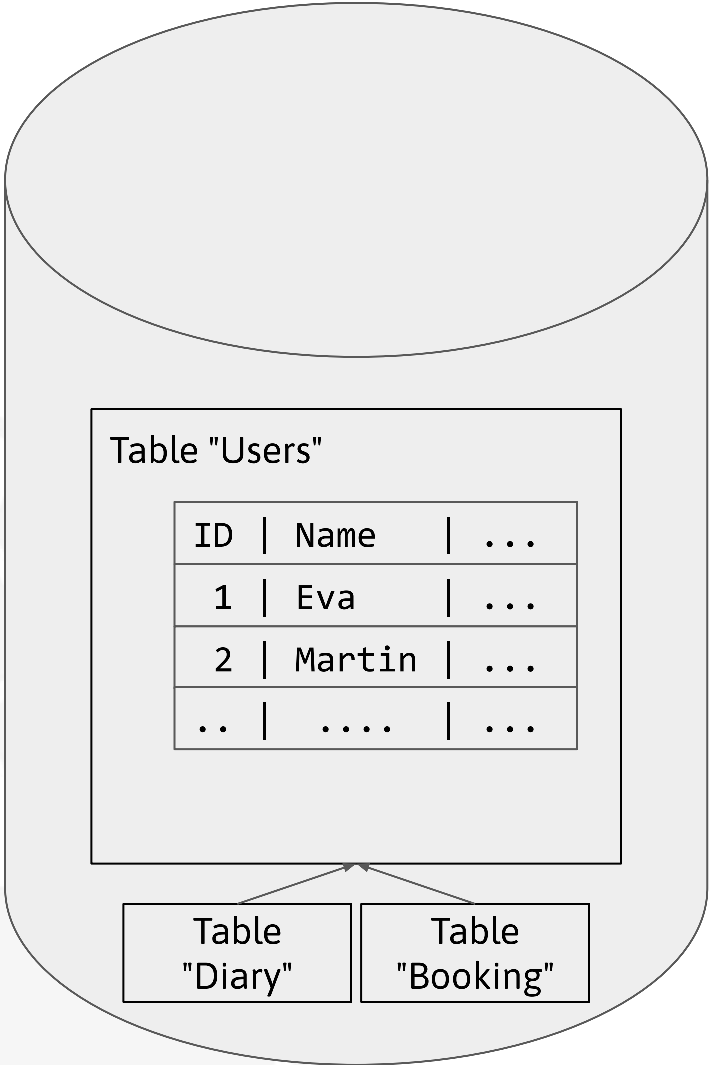
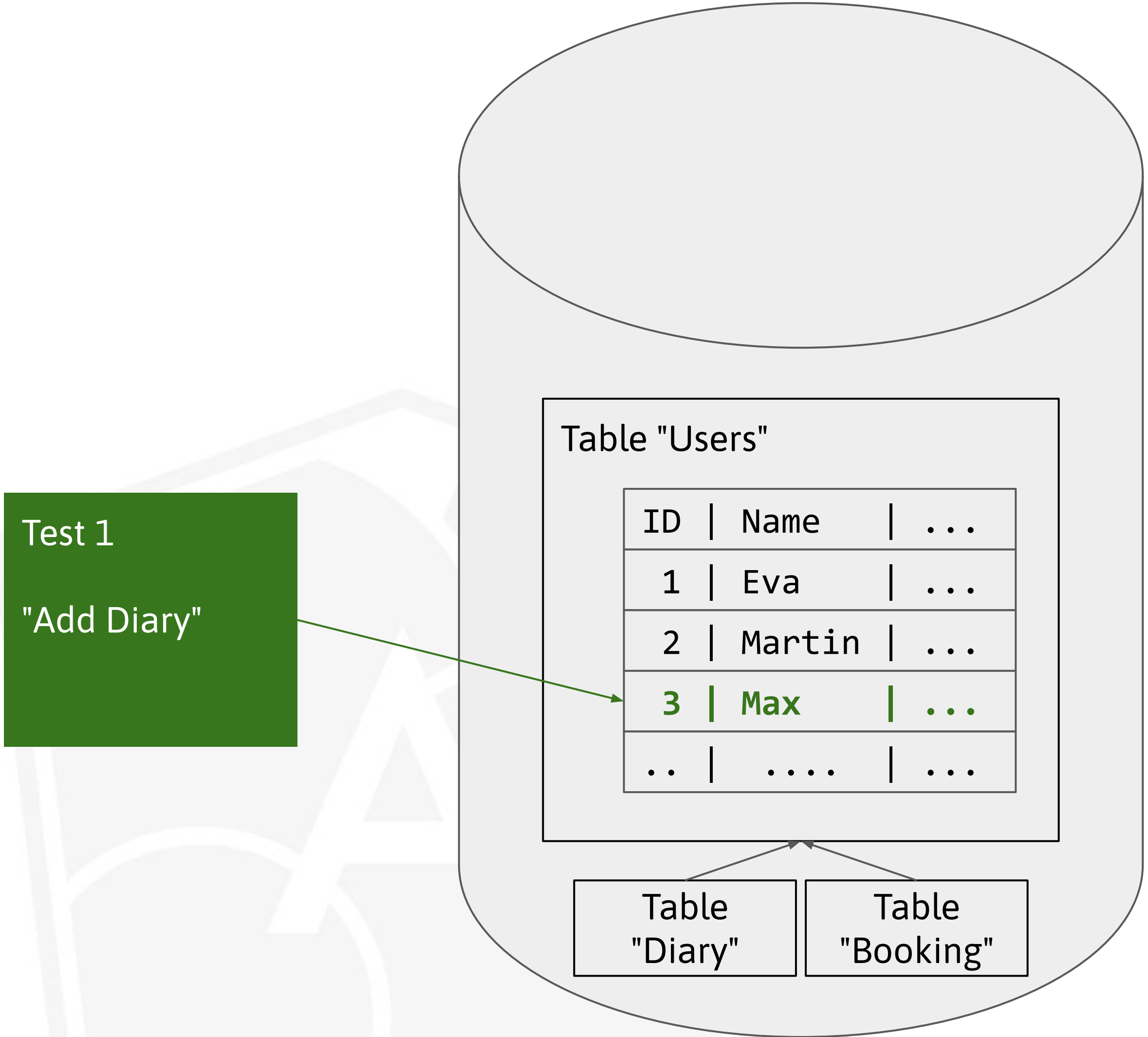


Table "Users"

ID	Name	...
1	Eva	...
2	Martin	...
..	....	...

Table "Diary"

Table "Booking"



Test 1

"Add Diary"

Table "Users"

ID	Name	...
1	Eva	...
2	Martin	...
3	Max	...
..	....	...

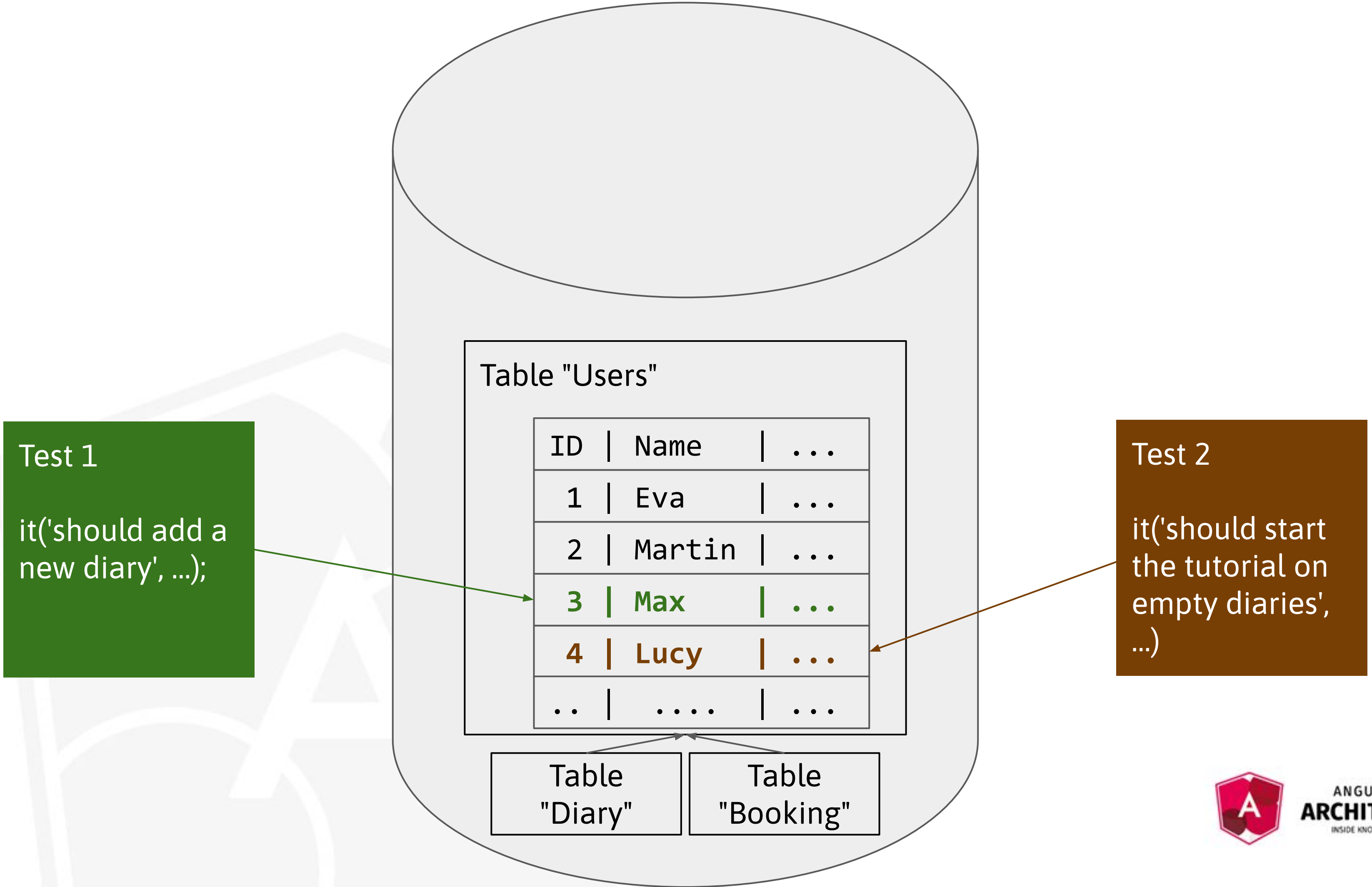


Table "Users"

ID	Name	...
1	Eva	...
2	Martin	...
3	Max	...
4	Lucy	...
..	....	...

Table "Diary"

Table "Booking"

Test 1

it('should add a new diary', ...);

Test 2

it('should start the tutorial on empty diaries', ...)

Test Setup !== Test

API Arrange Possibilities: Normal Requests

- Default Case
- Call same endpoints as the Frontend
- Don't use the frontend directly!
- cy.task() as alternative

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

API Arrange Possibilities: Dedicated Test API

- Backend provides special API for test mode
- Shortcuts possible, e.g.
  - merge chain of requests into one
  - Overcome Security Issues
- Best Option

Testing Scopes

- Individual Scope
  - All data depends on a certain ID
  - e.g. Personalised Data
  - Best Option in Combination with Test API (Sign Up & In)
- Global Scope
  - Tests Affect each other
  - Challenging Parallel Runs
  - Not so easy to solve...

Global Pattern I: Independent Tests

- Read-Only Character
- No Arranging required
- Rely on Test Seed
- Smoke Tests
- Tests for Static Elements

Global Pattern II: Intelligent Tests

- "I'll create and find it"
- Flexible
- Requires more code

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Global Pattern III: Dependent Test Group

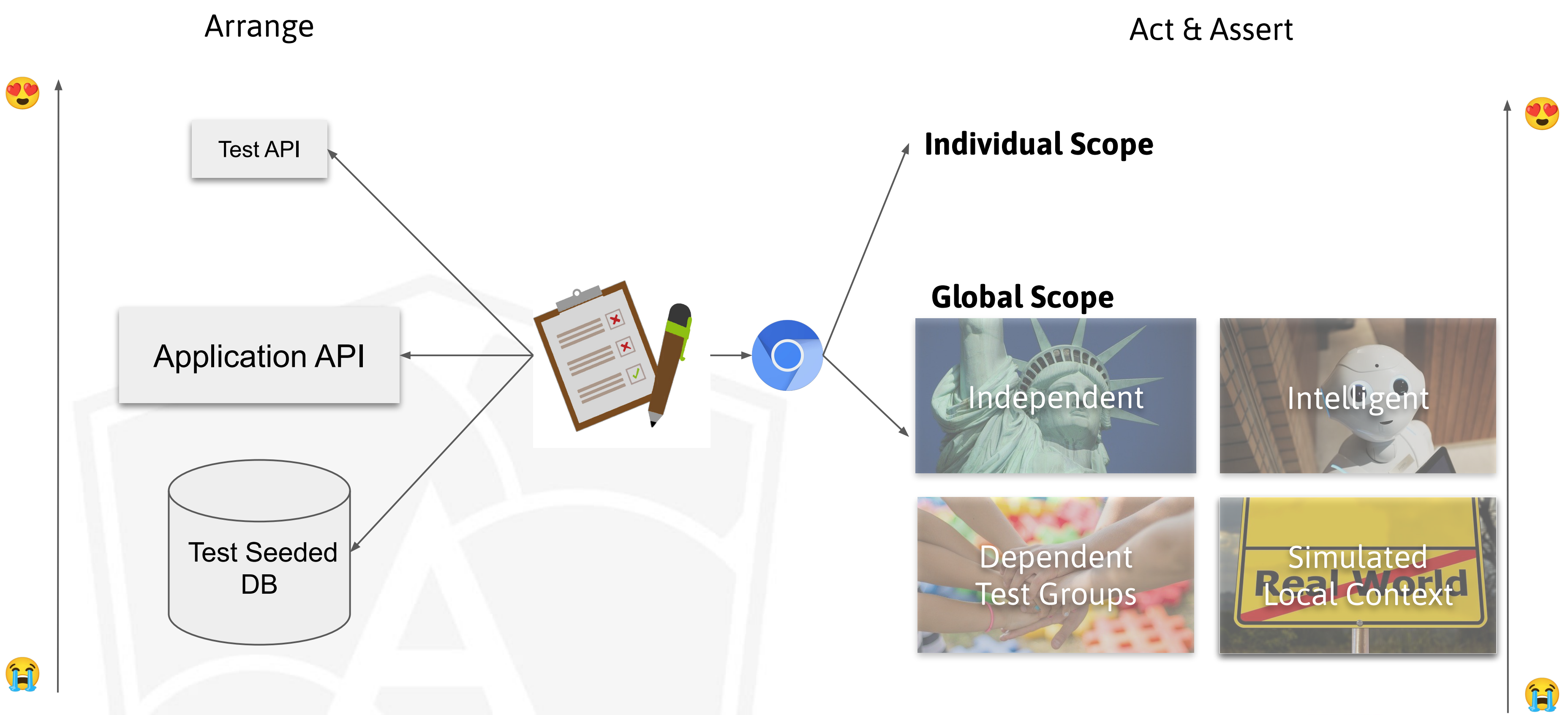
- Default Group
- Logical Group of Unit Tests
- Internal knowledge about other tests
- Order is important
- Database Reset after each Group Run

Global Scope IV: Simulated Individual Context

- Mock all APIs
- Transforms a global into an individual context

Global Scope V: Integration Tests

- Don't test it all and rely on integration tests



Arrange

Act & Assert

Test API

Application API

Test Seeded DB

Individual Scope

Global Scope

Independent

Intelligent

Dependent Test Groups

Simulated Local Context

Summary

- You will not have completely isolated tests
- Try to minimize loose coupling
- Always prefer Backend API over Test Seed
- Look out for opportunities with individual scope

What about Component Tests?

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Lab Time