



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Playwright Workshop



4 - Beyond E2E

Agenda

- Visual Regression
- Component tests
- API tests
- Isolated tests with database



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



Main Challenges

1. Pixel Matching
2. Component Isolation

Introducing Storybook



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Frontend Application



Headless Browser



Tests



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Storybook



↓

Code
(Components)



2

Headless Browser



1

3



4



Tests



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Storybook

- Allows to isolate Components
- Not Angular Specific
- Configure a Component for various states
- Can also used for visual widget library (not just testing)

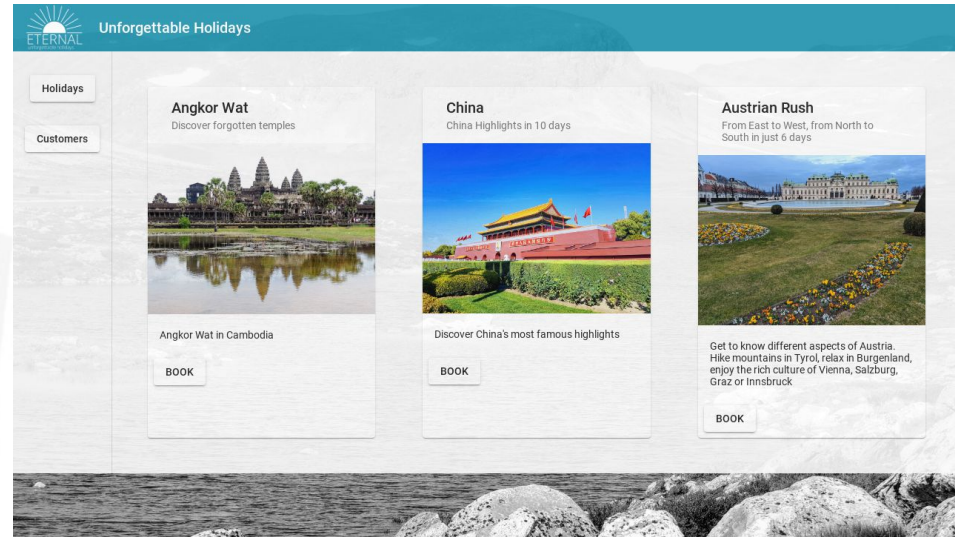
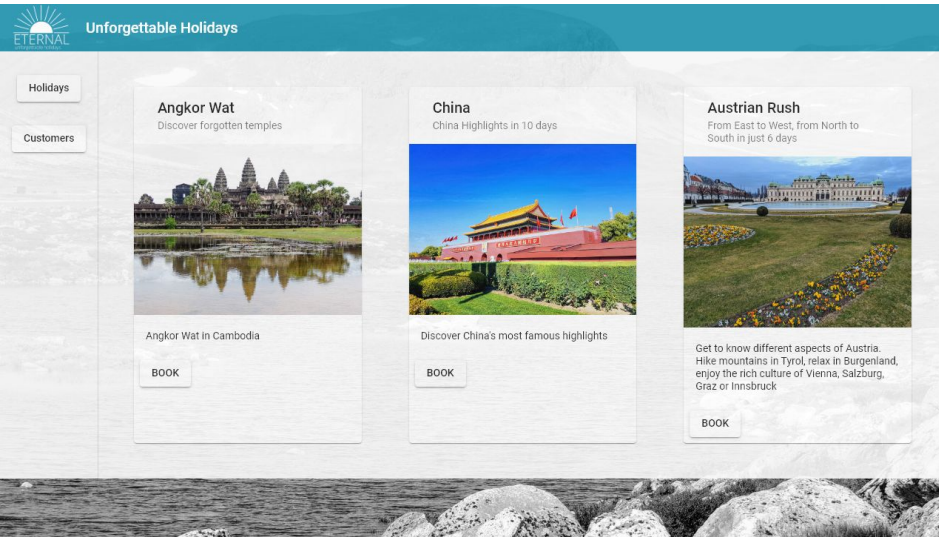


What about
Pixel Matching???

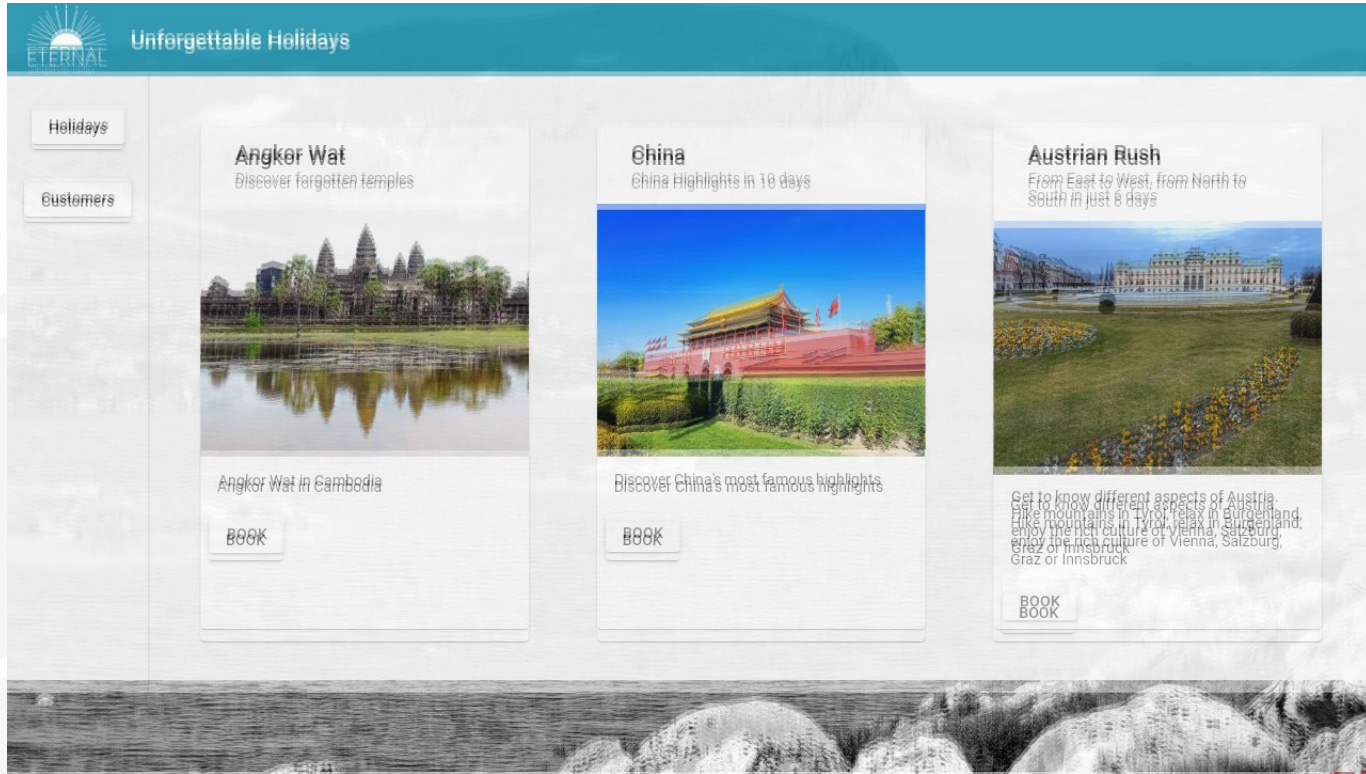


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Spot the difference



Spot the difference

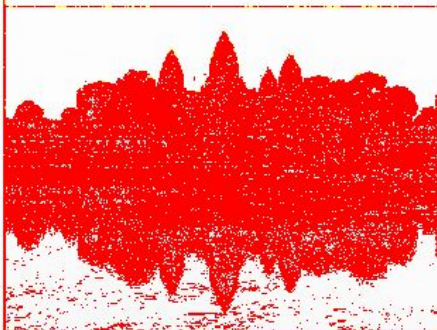


Holidays

Customers

Angkor Wat

Discover forgotten temples

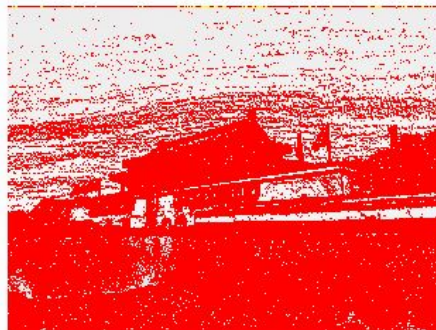


Angkor Wat in Cambodia

BOOK

China

China Highlights in 10 days

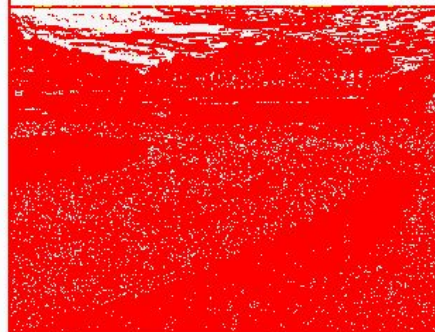


Discover China's most famous highlights

BOOK

Austrian Rush

From East to West, from North to South in just 6 days



Get to know different aspects of Austria. Hike mountains in Tyrol, relax in Burgenland, enjoy the rich culture of Vienna, Salzburg, Graz or Innsbruck

BOOK

Storybook



↓

Code
(Components)



2

Headless Browser



1

3



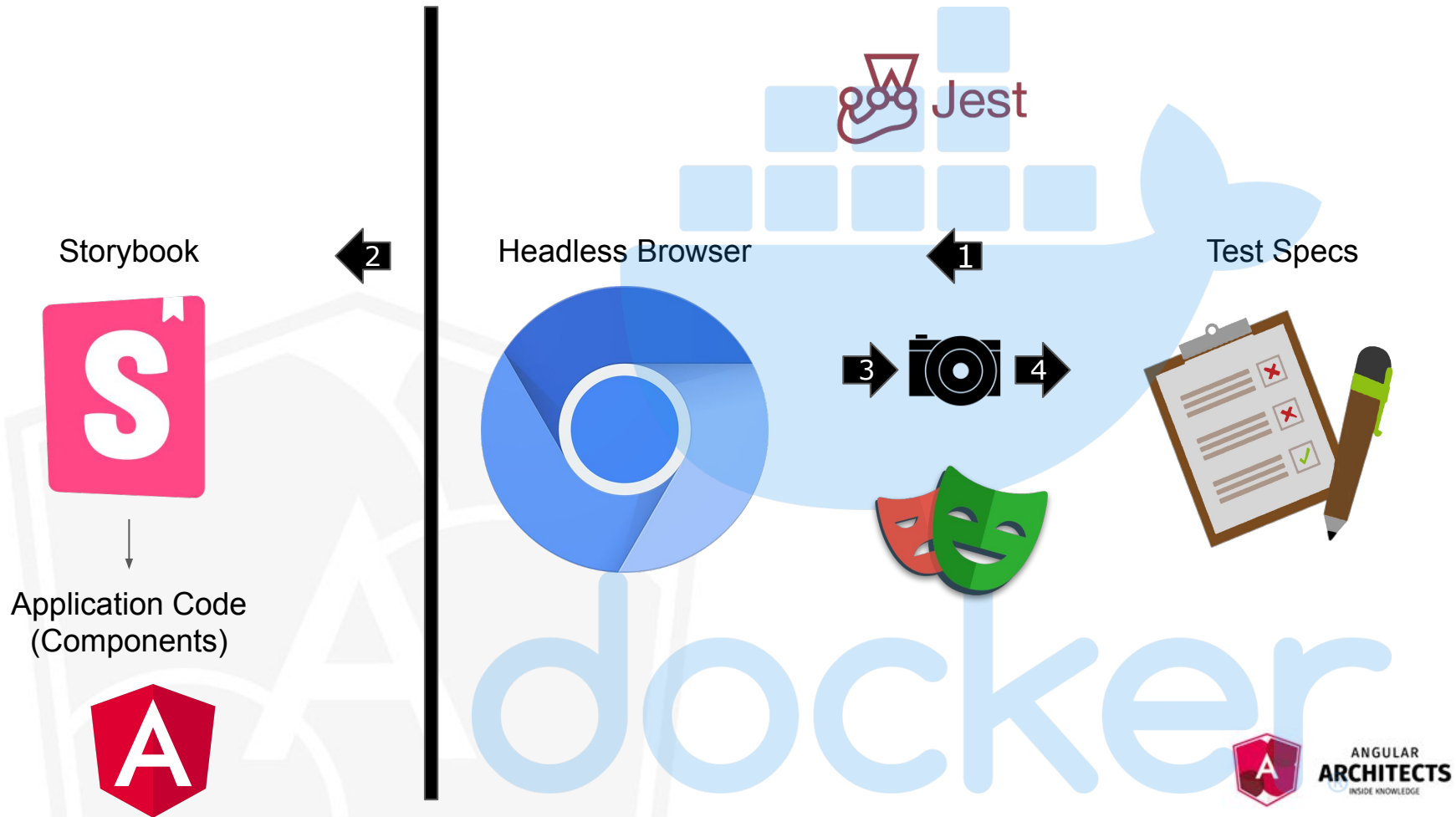
4



Tests



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



Summary

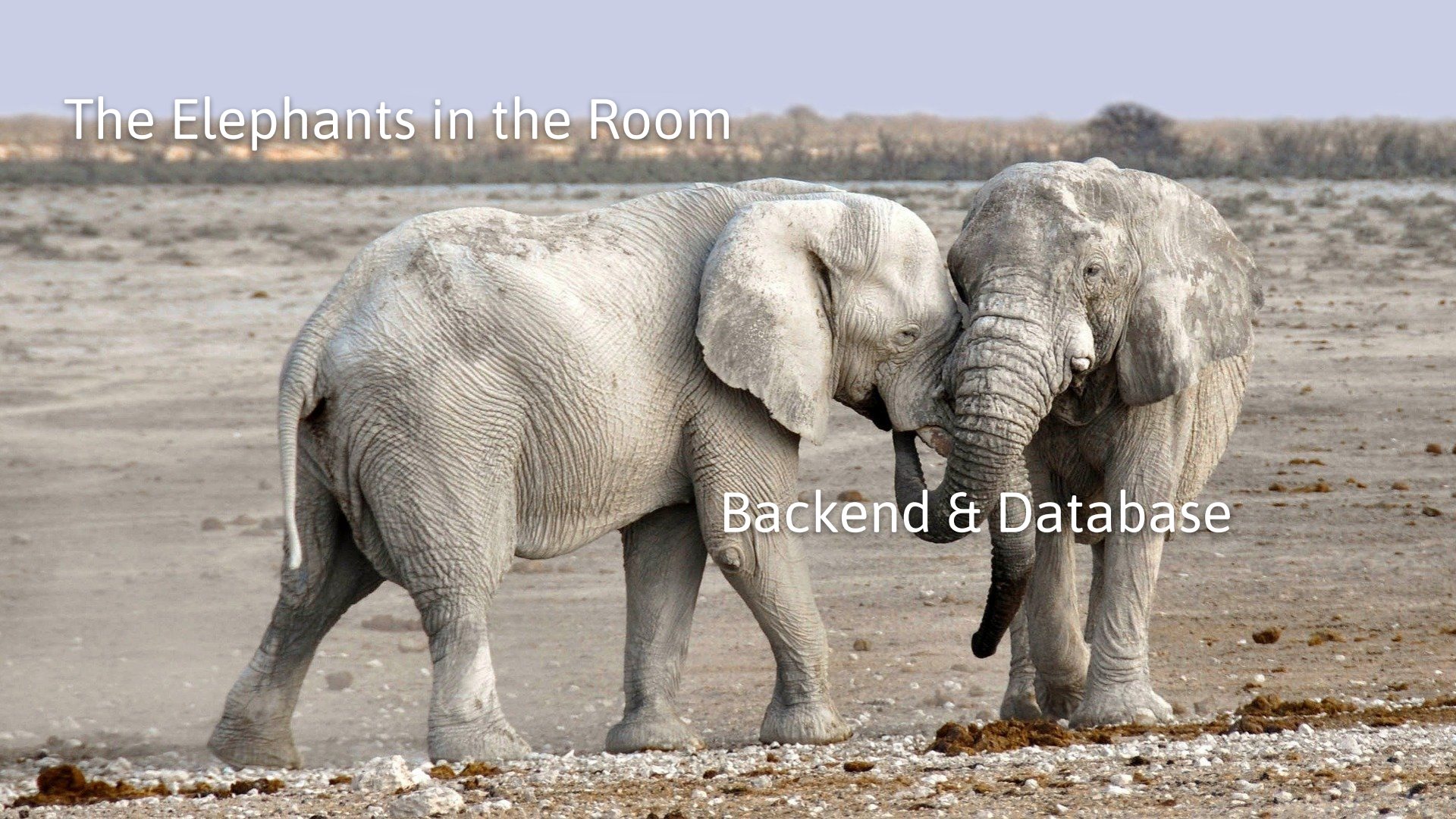
- Visual Regression is an important testing technique
- Special Requirements for Infrastructure
 - Docker
 - Cloudprovider

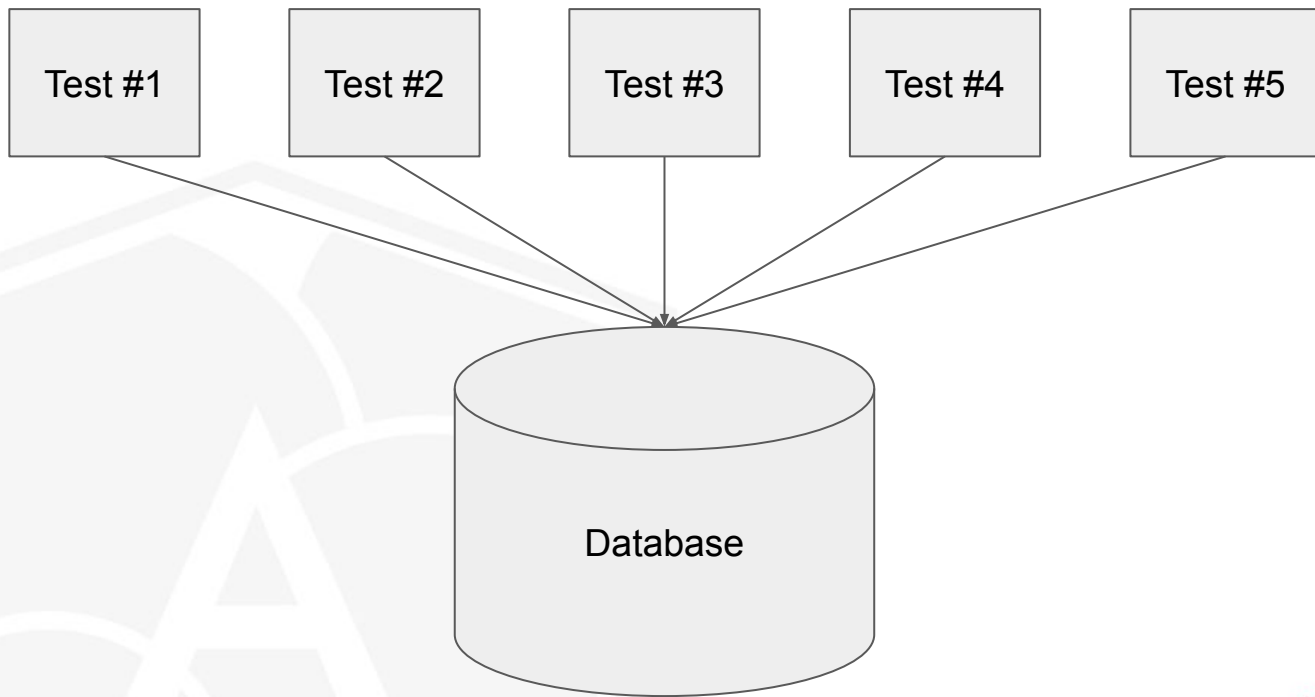


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

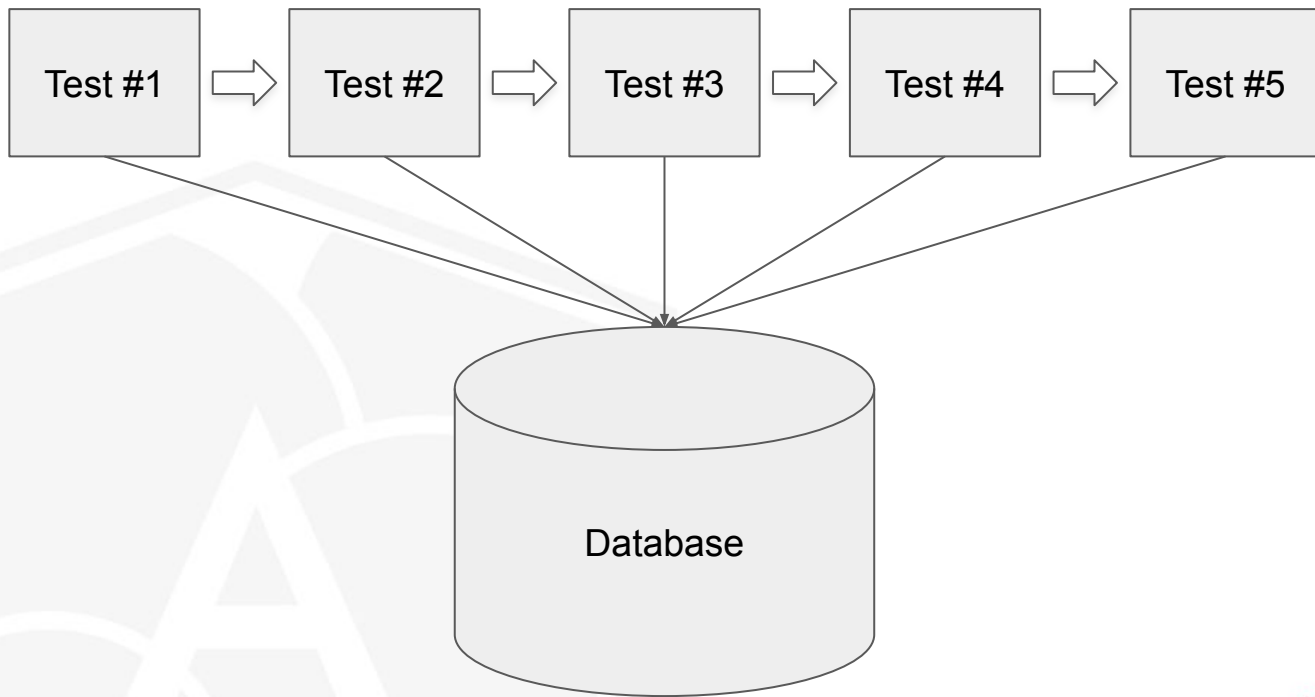
The Elephants in the Room

Backend & Database

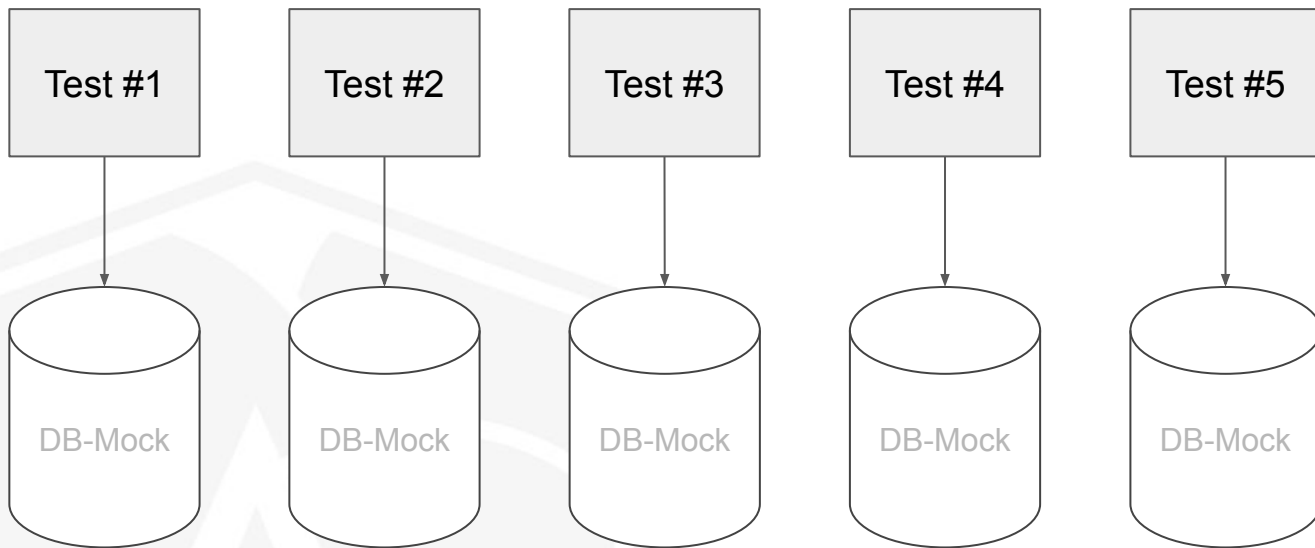




Indirect "Transitive" Coupling



Copy Strategy from non-E2E???



Database in E2E Tests



Database in Non-E2E Tests

Test Seeded Database & done???



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Issues with Test Seed

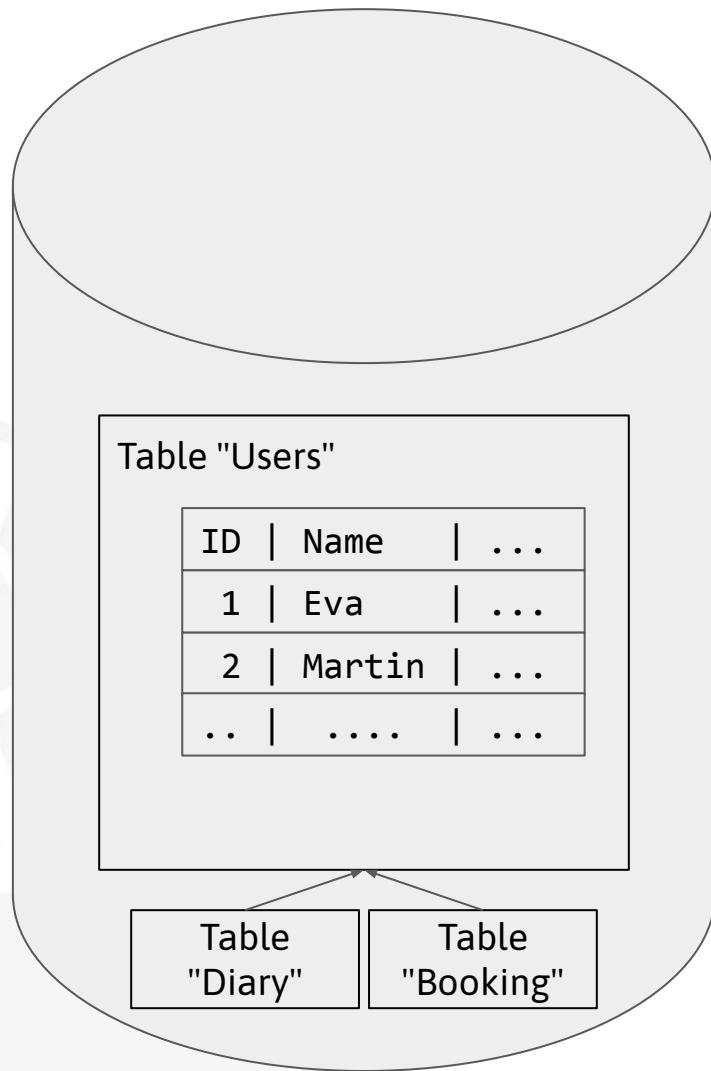
- "One size fits all" approach
- Tight Coupling → Not scalable
- Fast Reseeding not always possible
- Multiple Databases
- Data from External Systems → no Seeding possible



Individual Scope - Best Case Scenario

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





Test 1

"Add Diary"

Table "Users"

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

Table
"Diary"

Table
"Booking"



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Test 1

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

Table "Users"

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

Table
"Diary"

Table
"Booking"

Test 2

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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Test Setup !== Test



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

API Arrange Possibilities: Normal Requests

- Default Case
- Call same endpoints as the Frontend
- Don't use the frontend directly!
- use `request.[get|post|...]()`



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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Arrange

Act & Assert



Test API

Application API

Test Seeded
DB



Individual Scope

Global Scope



Dependent
Test Groups



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

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

