



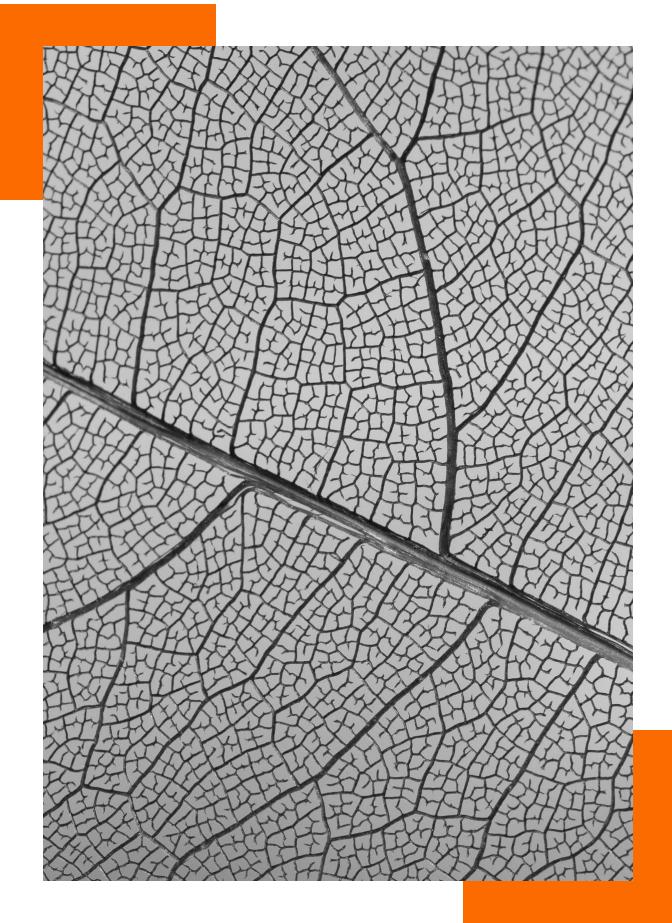
<https://github.com/romansegador/web-app> (TypeScript)
/testers-service (Java)
/books-service (Java)

Contract Testing

How to simplify your integration tests

While we start, if you want to practice at the end, clone the projects at the top and run the following commands, you will save some time at the end:

TypeScript-> npm install
Java -> mvn dependency:resolve



WHAT YOU'LL LEARN

- 01.** Understand how contract testing mitigates integration issues.
- 02.** Describe the benefits of Consumer-Driven Development.
- 03.** Explain the flow of contract testing and how it fits in the SDLC.

Let me introduce myself...



Román Segador

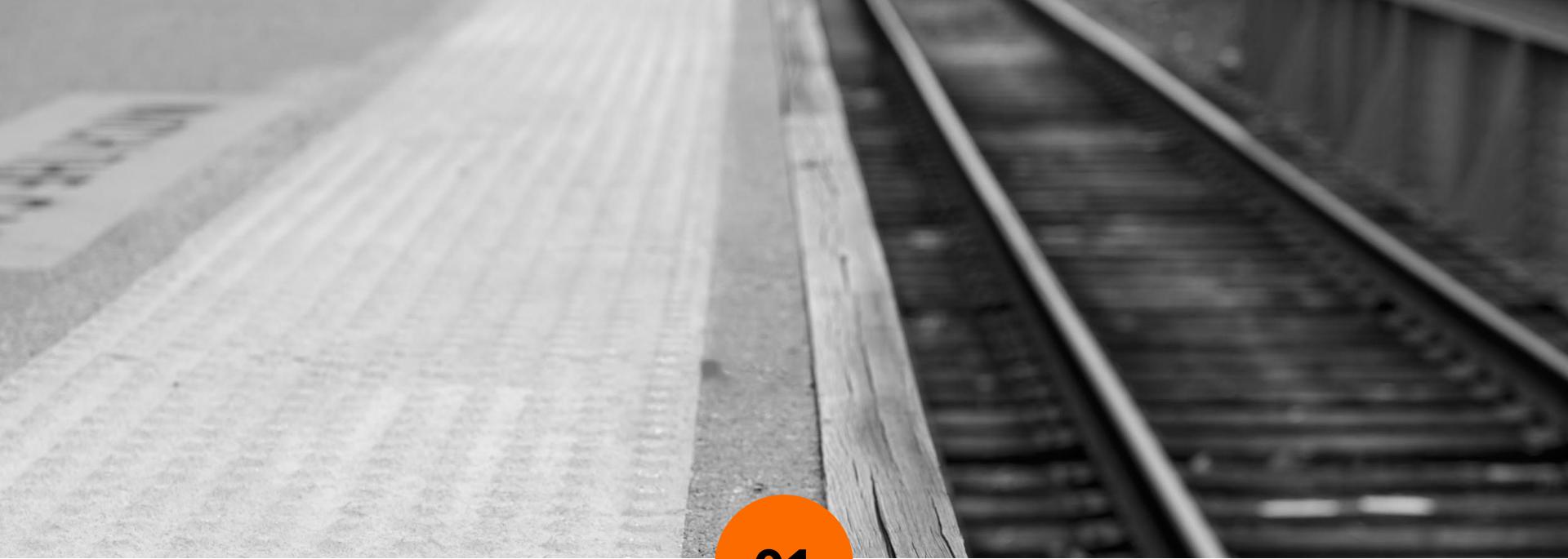
- +10 years working in **quality**.
- Quality Architect at **The Workshop**.
- **Agile Testing** Manifesto.
- **Modern testing** principles.
- Quality is a **whole team responsibility**.



<https://www.linkedin.com/in/romansegador/>



@roman_s_t



01

The Integration Testing Problem

And how they are maximized in the
microservices world



Team

Front-end

Back-end

Database

Web Team



vector created by stories - www.freepik.com

Customers Team



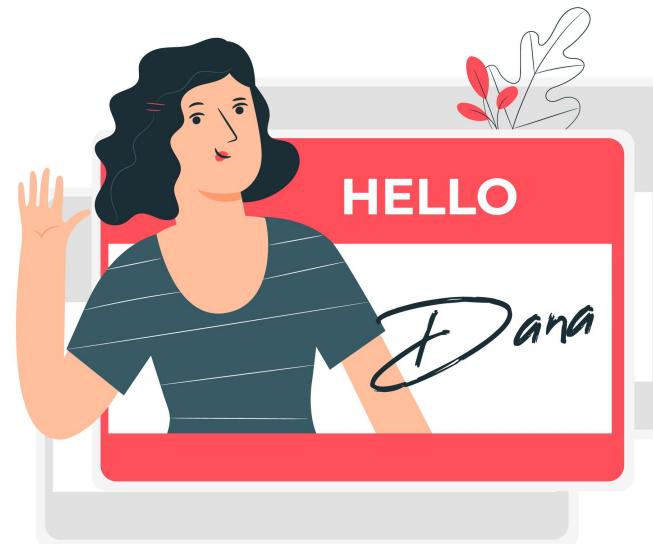
vector created by freepik - www.freepik.com

Web Team



vector created by stories - www.freepik.com

THE INTEGRATION

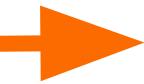


THE INTEGRATION



vector created by stories - www.freepik.com

Get User



vector created by freepik - www.freepik.com

THE INTEGRATION PROCESS



Talk

THE INTEGRATION PROCESS



Talk



Agree

THE INTEGRATION PROCESS



Talk



Agree



Coordinate

THE INTEGRATION PROCESS



Talk



Agree



Coordinate



Document

THE INTEGRATION



MOOD



vector created by stories - www.freepik.com



vector created by freepik - www.freepik.com



vector created by rawpixel.com - www.freepik.com



vector created by rawpixel.com - www.freepik.com

6 MONTHS LATER...



vector created by freepik - www.freepik.com

6 MONTHS LATER...

Customers Team



designed by freepik

Add Full Name!



designed by freepik

NEW FEATURE FINISHED

Team Customers



NEW FEATURE FINISHED

Team Customers



NEW FEATURE FINISHED

Team Customers



NEW FEATURE FINISHED

Team Customers



NEW FEATURE FINISHED

Team Customers

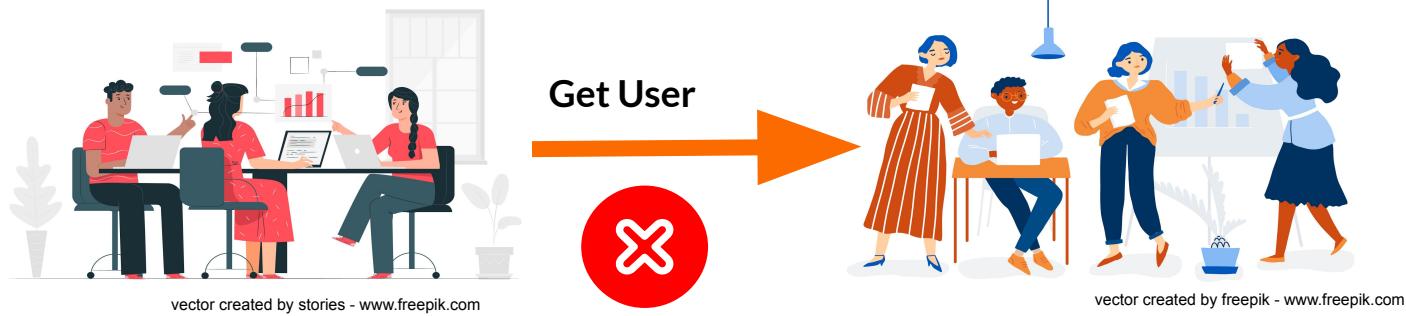


NEW FEATURE FINISHED

Team Customers



THE ROOT CAUSE



GET /profile/{id}

```
{  
  "Name": "Jane",  
  "Lastname": "Doe"  
}
```



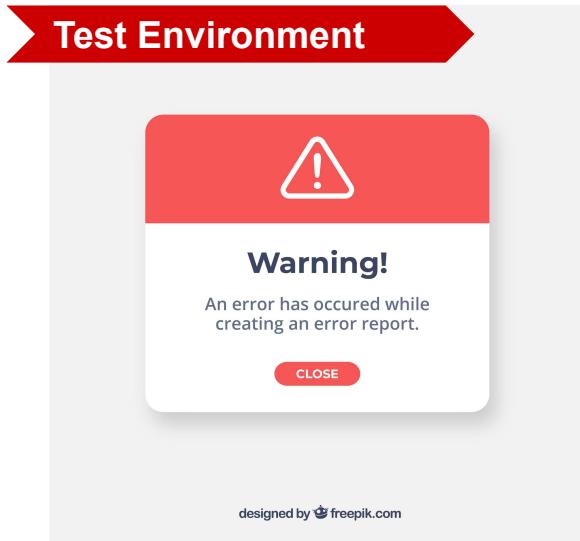
```
{  
  "Firstname": "Jane",  
  "Lastname": "Doe",  
  "Fullname": "Jane Doe"  
}
```

THE ISSUE

Production

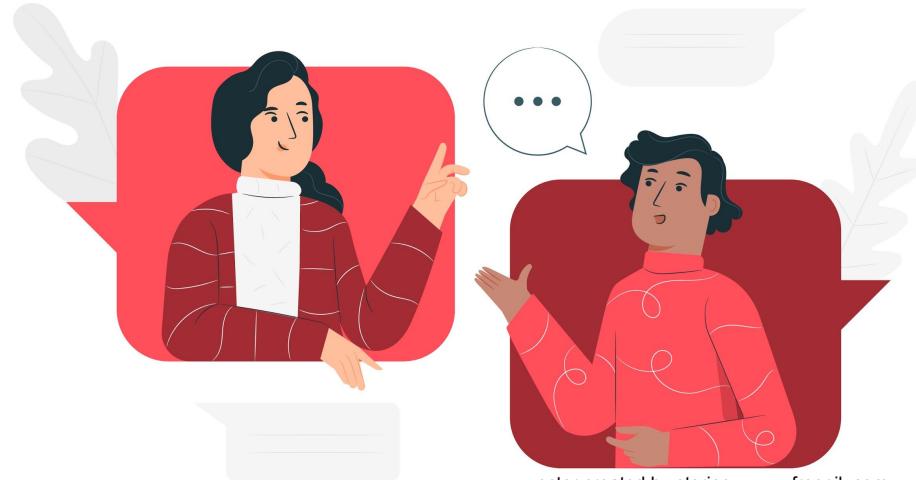


Test Environment



designed by freepik.com

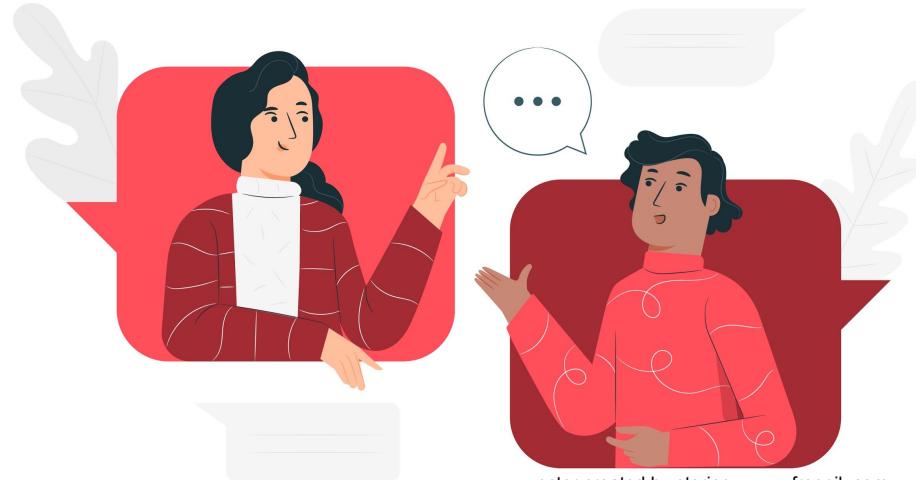
HOW TO FIX IT?



vector created by stories - www.freepik.com

- Rollback
- Customer Team Fix It
- Web Team Fix It

HOW TO FIX IT?

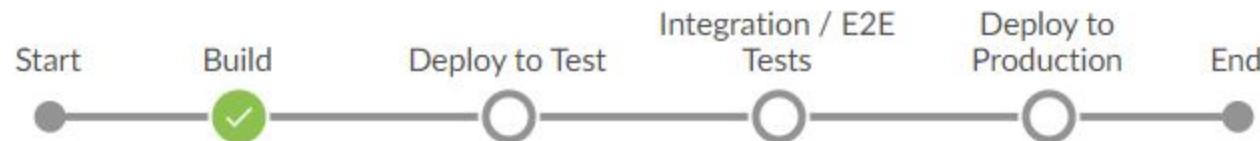


vector created by stories - www.freepik.com

Web Team Will Fix It

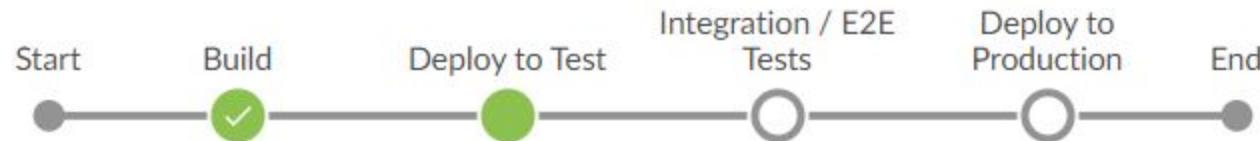
ISSUE FIXED BY WEB TEAM

Web Team



ISSUE FIXED BY WEB TEAM

Web Team



ISSUE FIXED BY WEB TEAM

Web Team



MOOD



vector created by stories - www.freepik.com



vector created by freepik - www.freepik.com



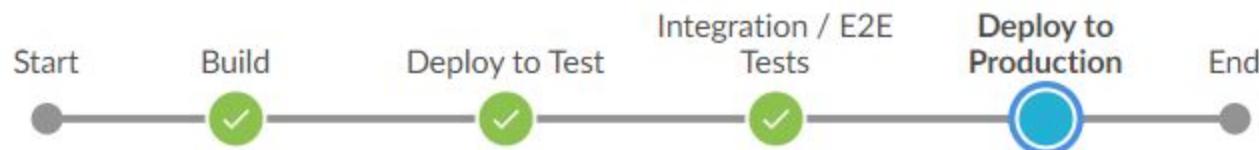
vector created by rawpixel.com - www.freepik.com



vector created by rawpixel.com - www.freepik.com

PROD DEPLOY

Customer Team



PROD DEPLOY

Customer Team



PROD DEPLOY

Customer Team



PROD DEPLOY

Customer Team



WebTeam



vector created by stories - www.freepik.com



Vector created by rawpixel.com - www.freepik.com

Customer Team



vector created by freepik - www.freepik.com



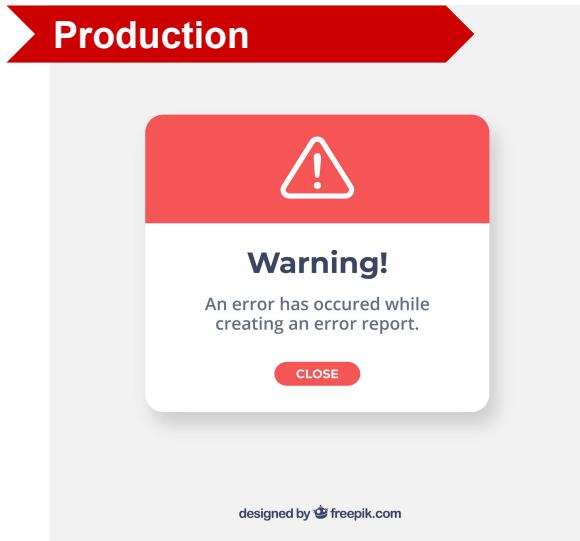
vector created by rawpixel.com - www.freepik.com

PRODUCTION INCIDENT

Test Environment

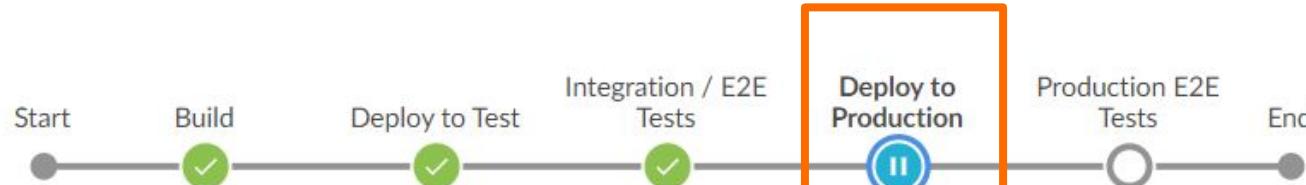


Production



PRODUCTION INCIDENT

Web Team



Customer Team



EMERGENCY RELEASE

Web Team



Customer Team



UNDERSTANDING THE PROBLEM



BROKEN ENVIRONMENTS

Errors found after deploy, so environment is already broken



RACE CONDITION

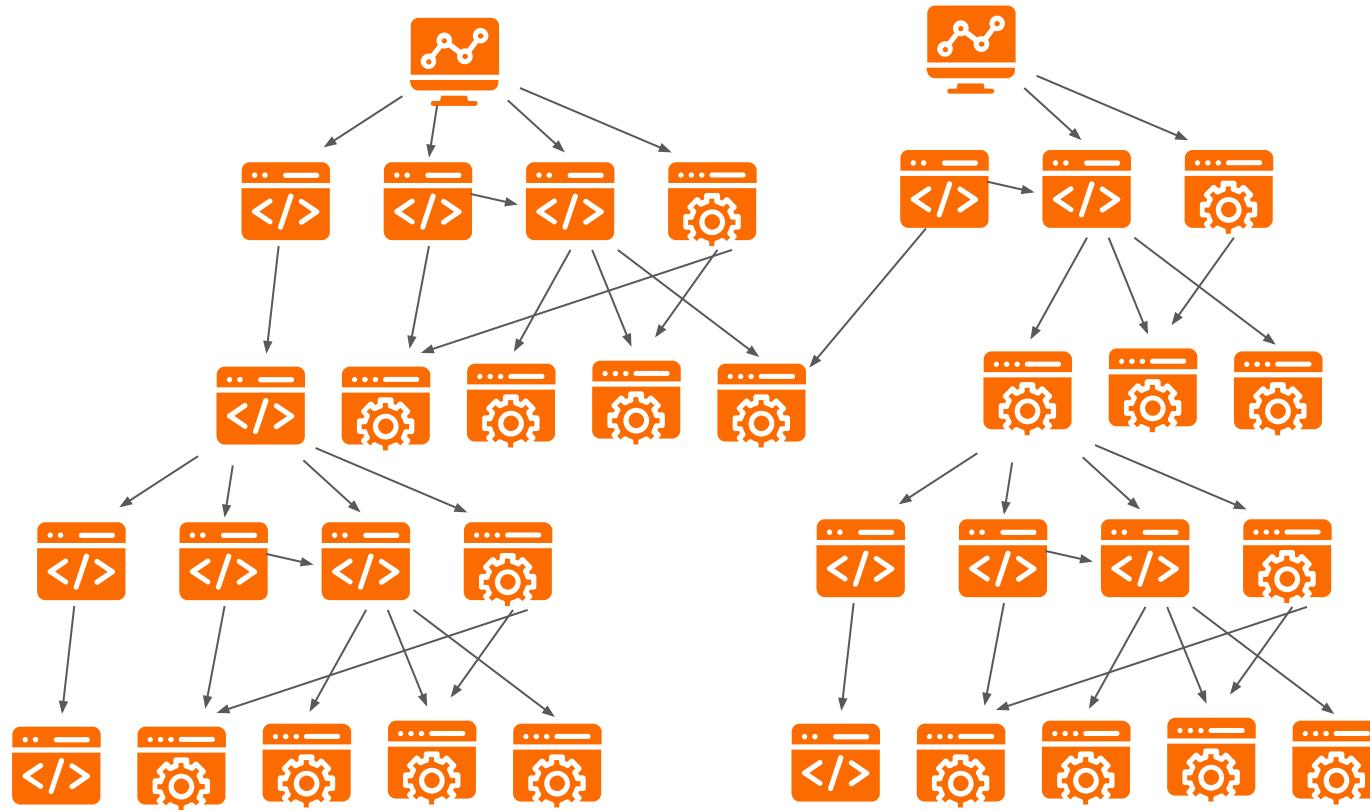
Testing the versions in a current environment which could differ from next environment versions



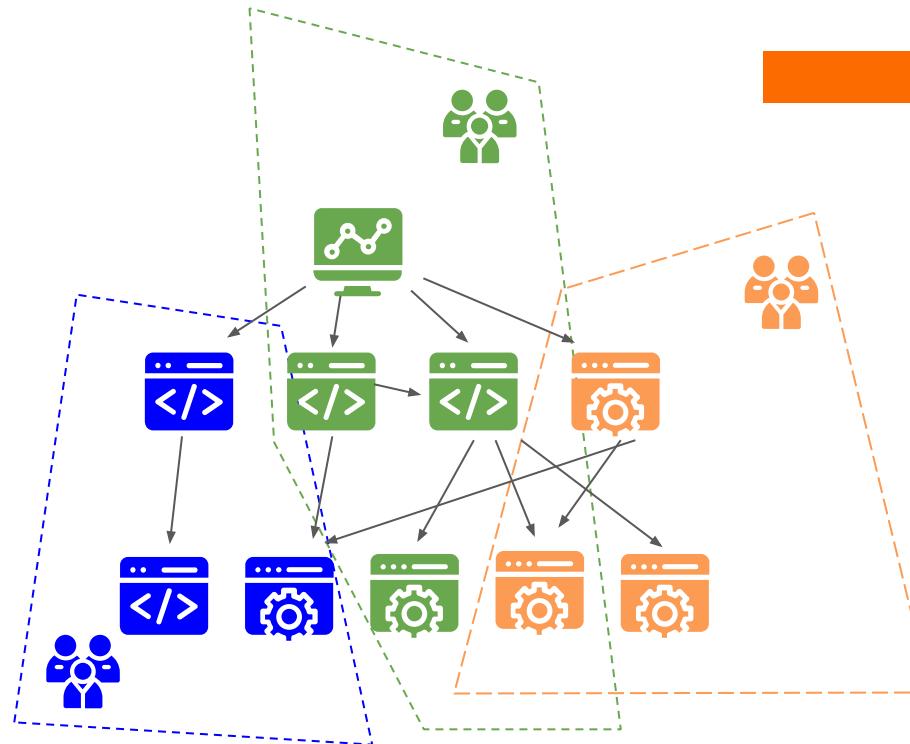
COORDINATION

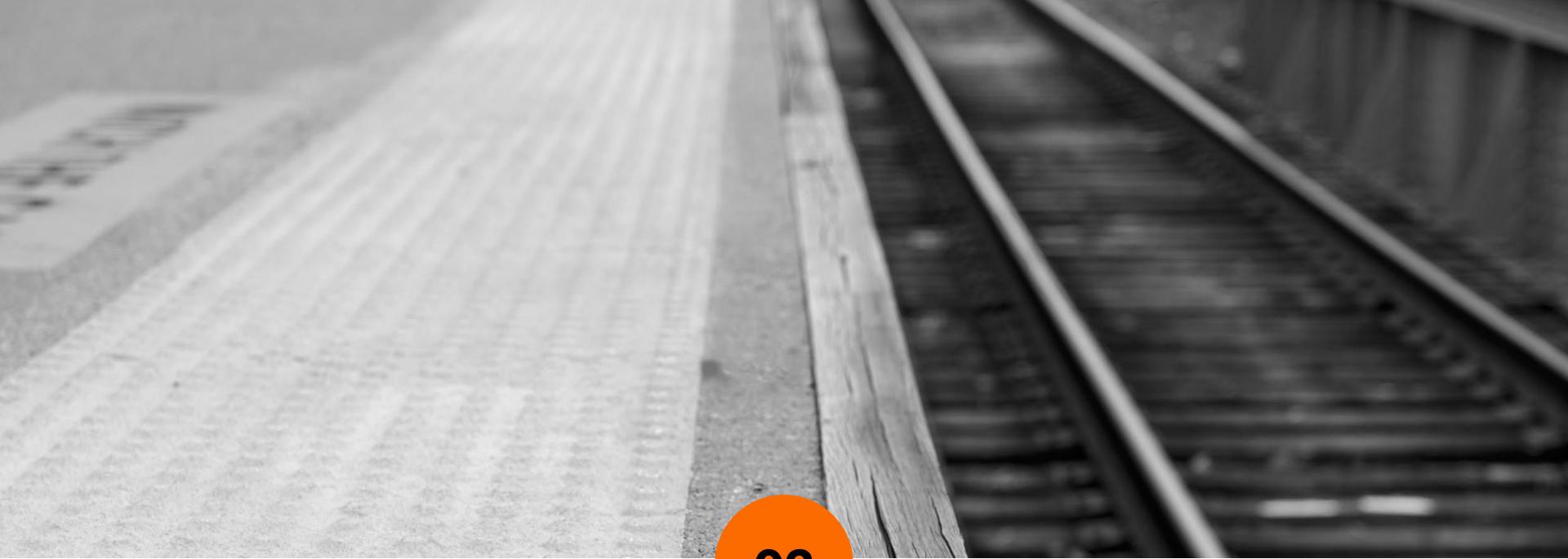
Documentation a Planification needed on the changes to don't block environments

MICROSERVICES



MICROSERVICES

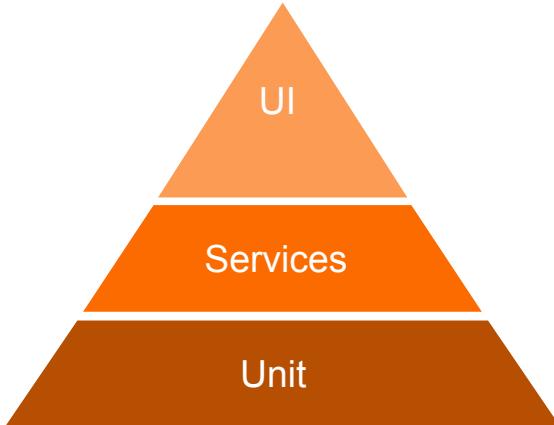




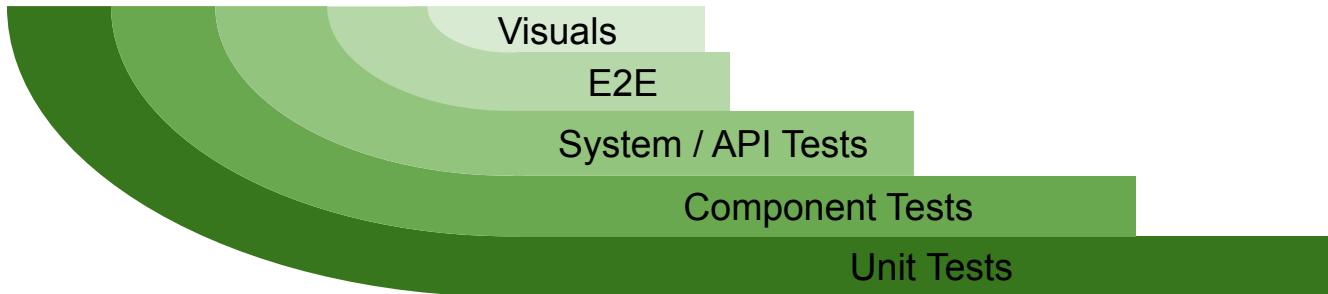
02

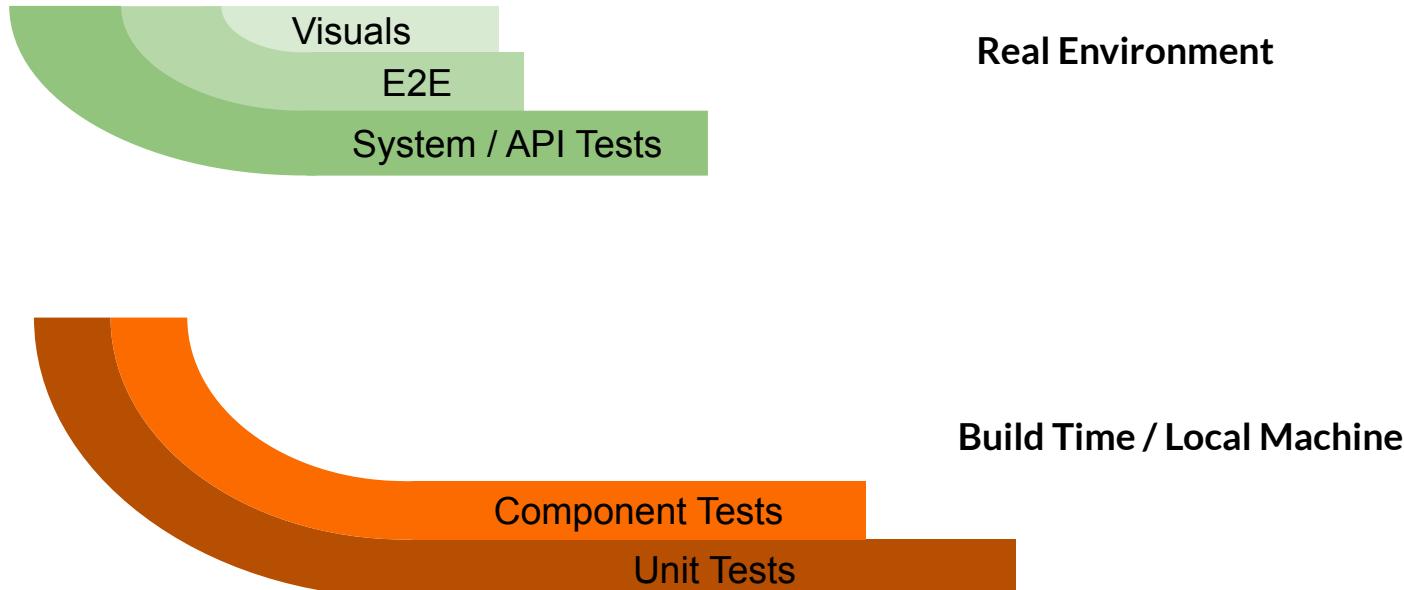
The Testing Pyramid

THE TESTING PYRAMID

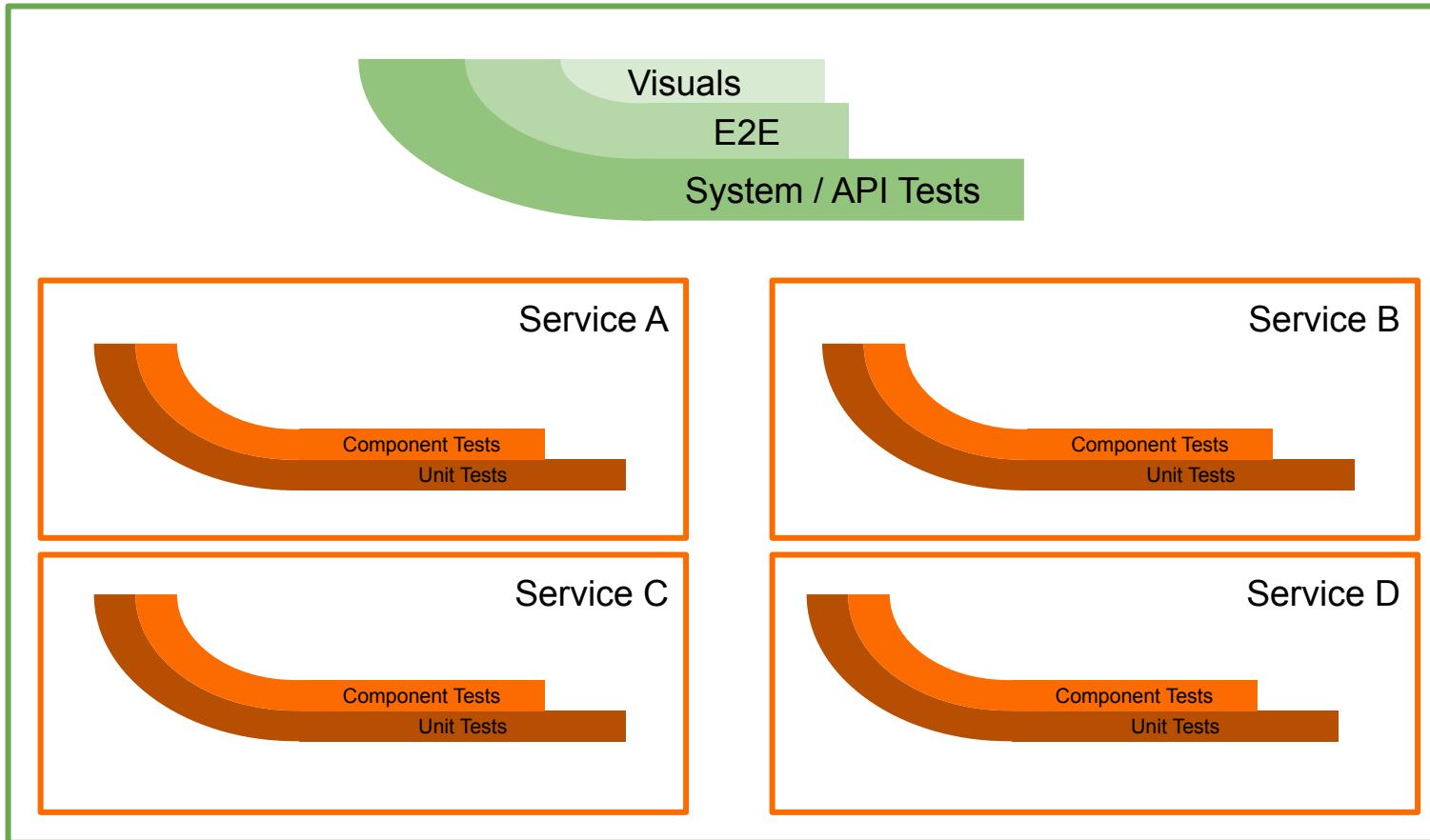


OUR TESTING PYRAMID





MICROSERVICES



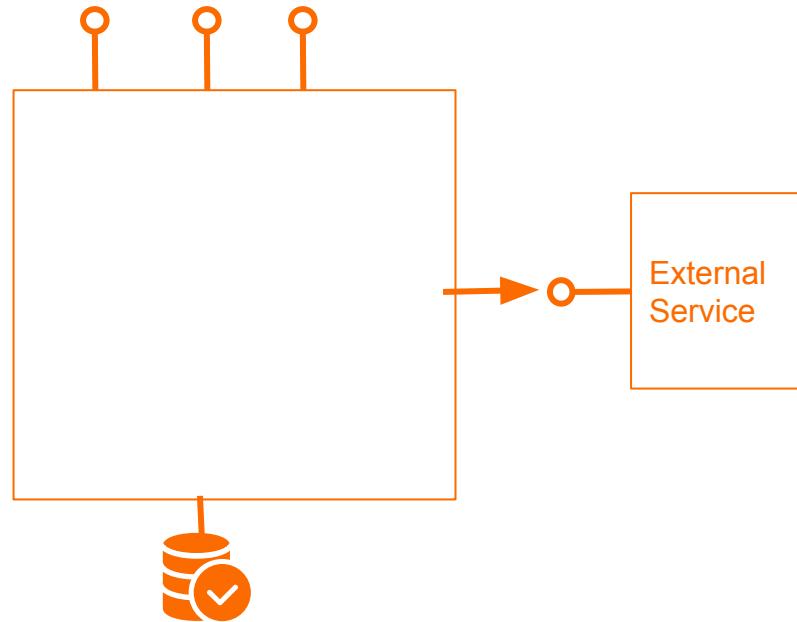


A black and white photograph showing a perspective view of railroad tracks. The tracks are made of metal rails and wooden sleepers, and they lead the eye towards a vanishing point in the distance. The lighting creates strong highlights and shadows on the rails and sleepers.

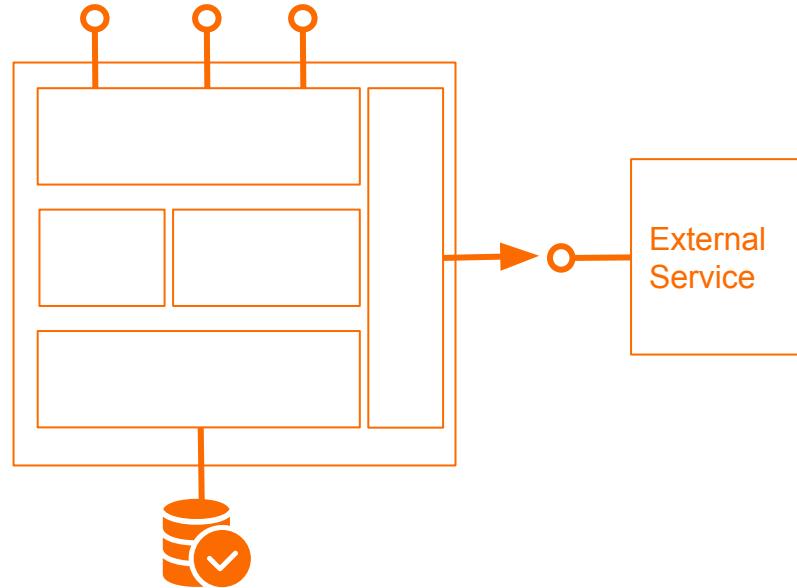
03

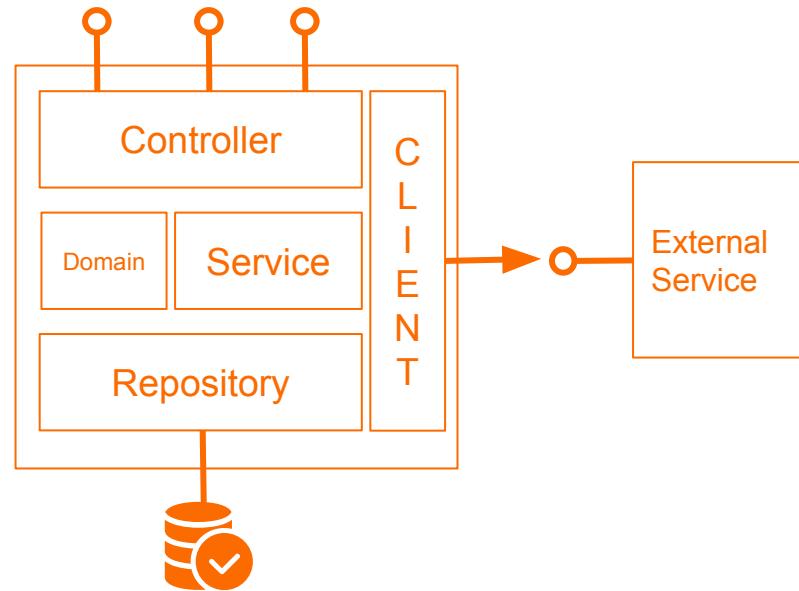
Contract Testing

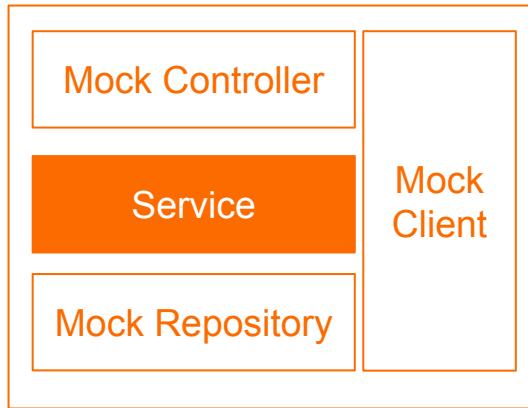
SERVICE LAYERS



SERVICE LAYERS



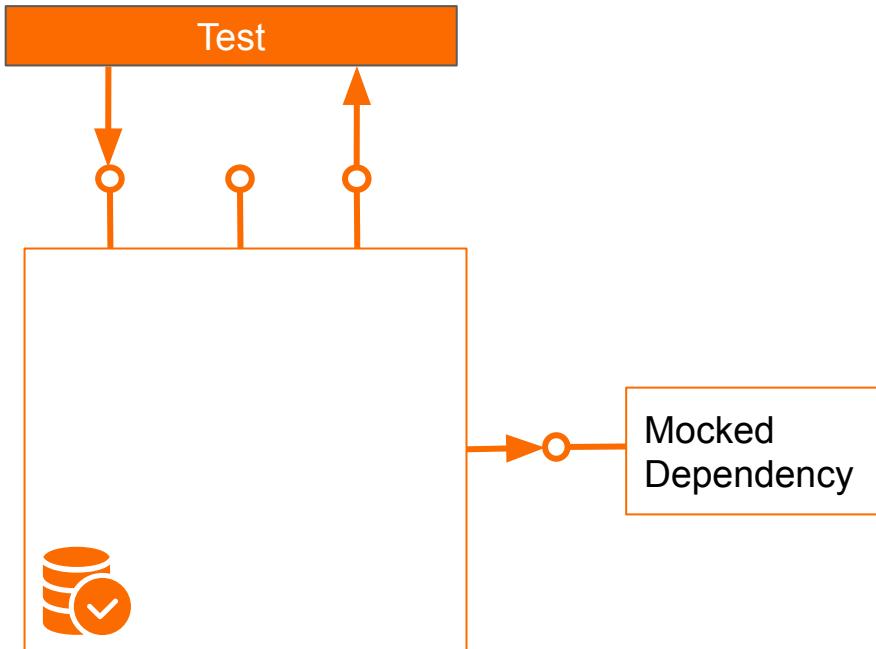




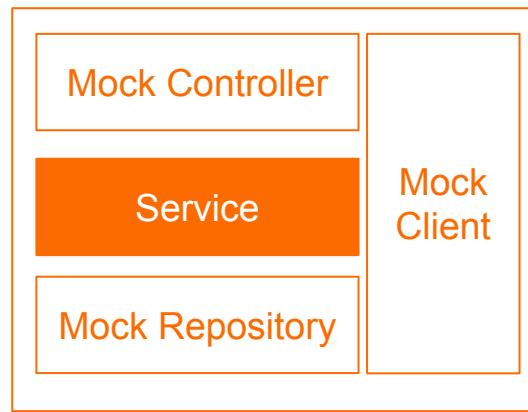
Mock

Fake implementation with the same public interface

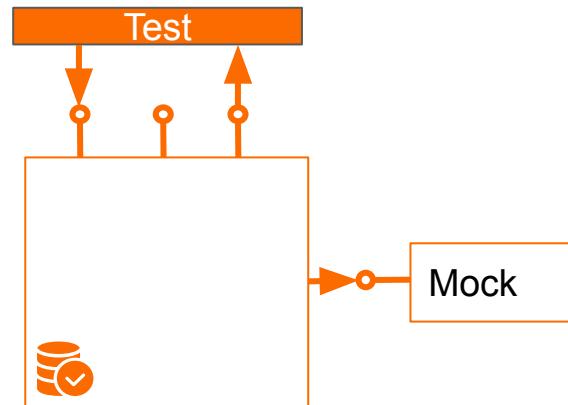
COMPONENT TESTS

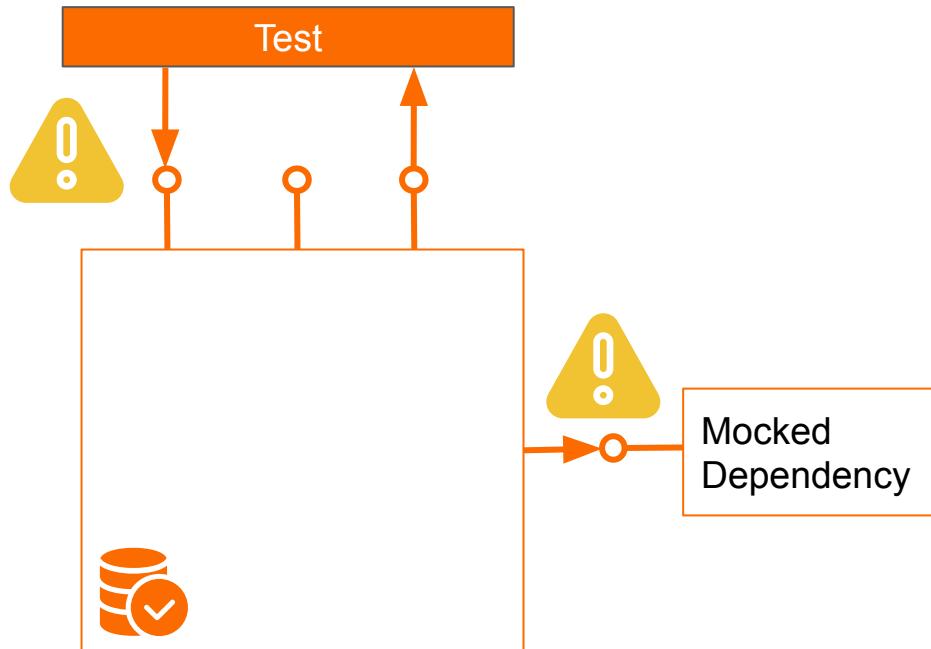


UNIT VS COMPONENT MOCKS

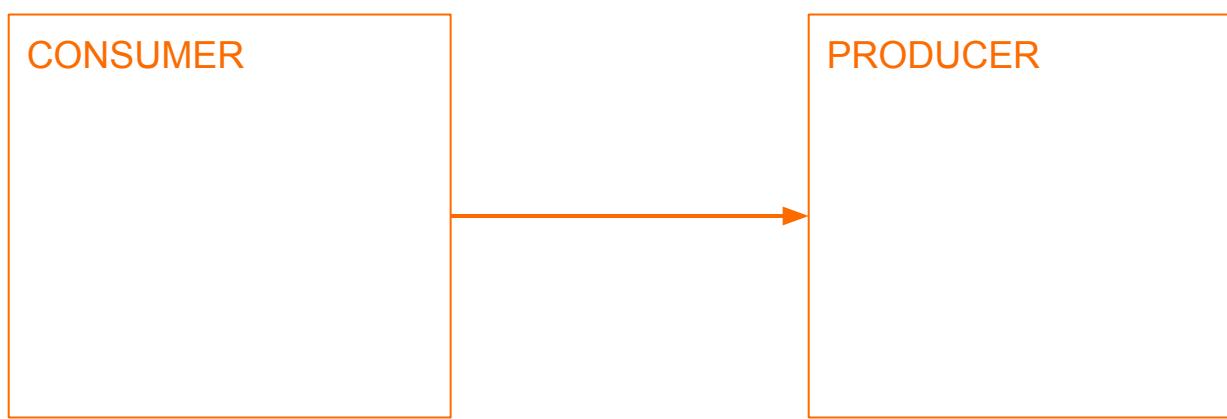


Mocks
Difference?





CONTRACTS



CONTRACTS

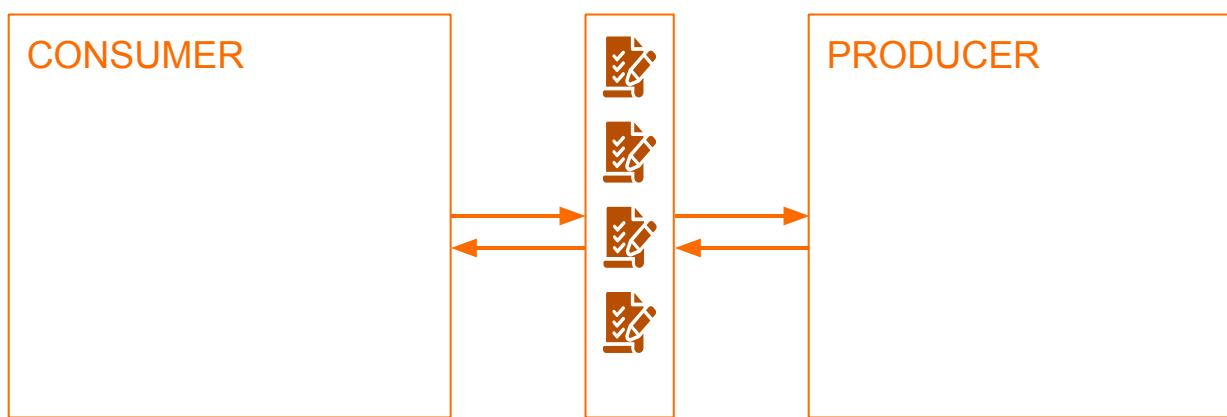
CONSUMER



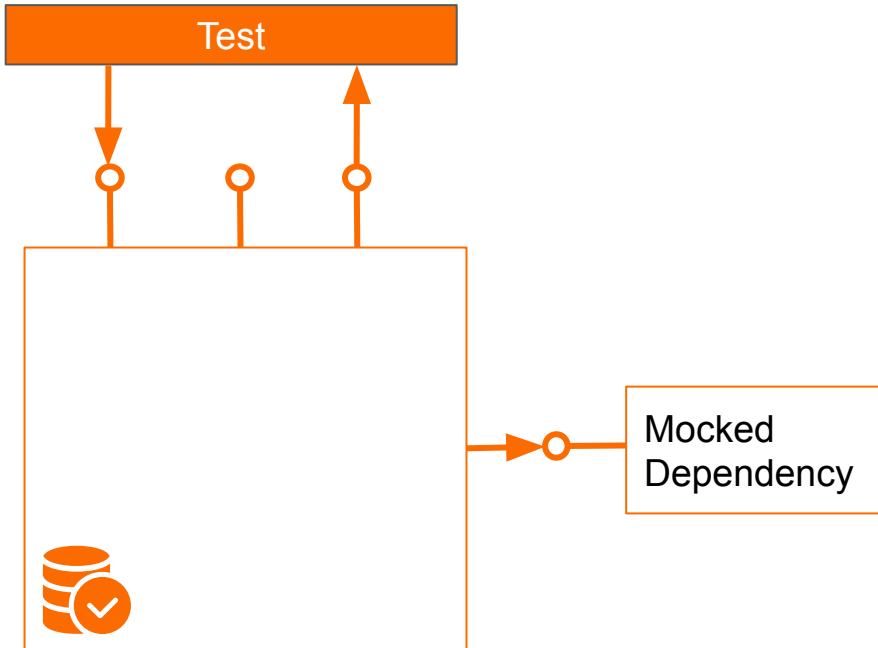
PRODUCER

```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]
```

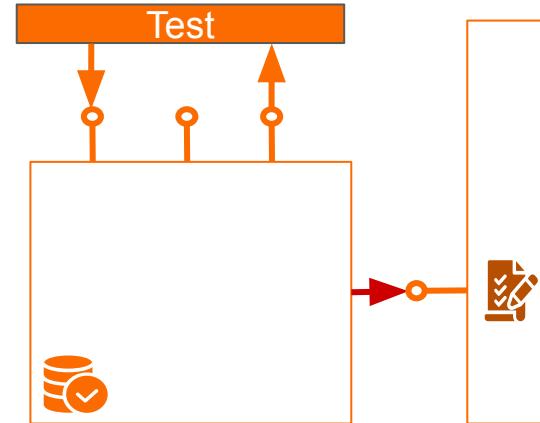




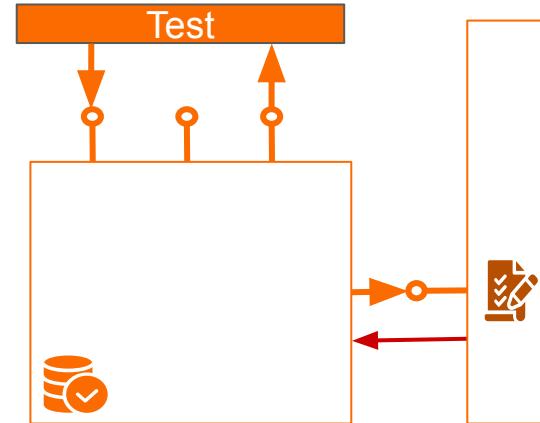
COMPONENT TESTS

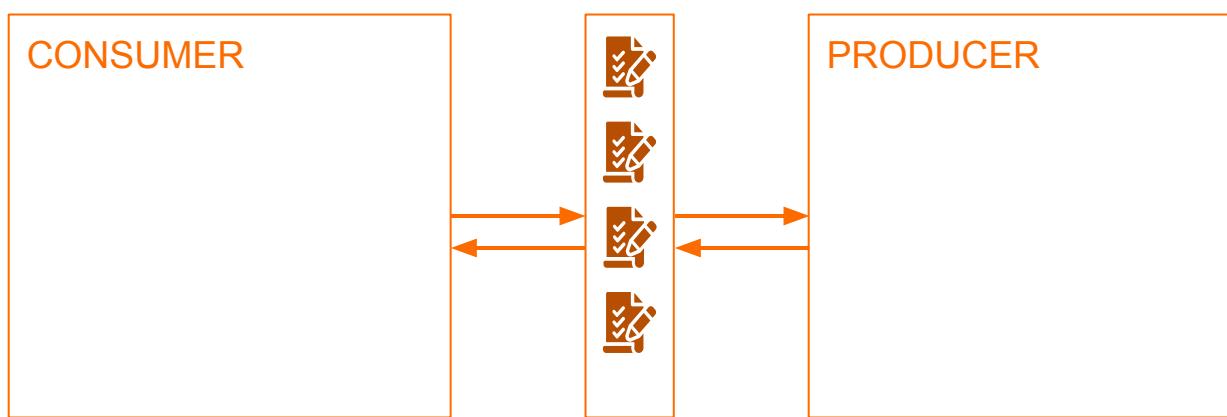


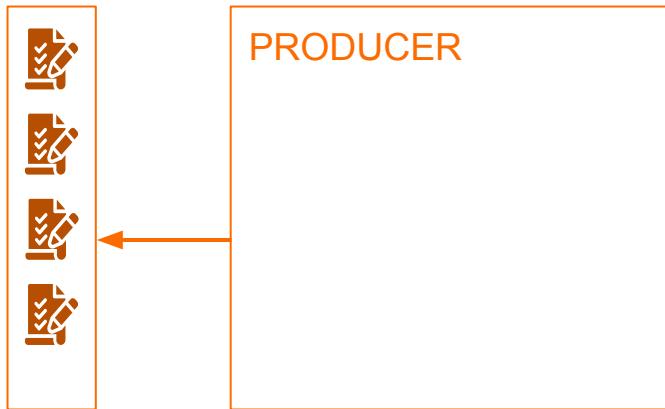
```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```

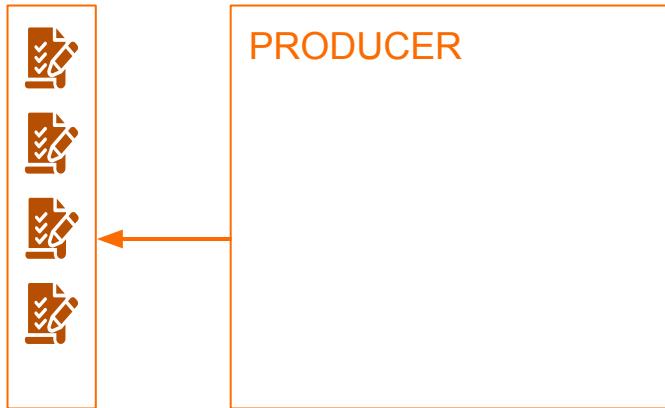


```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```

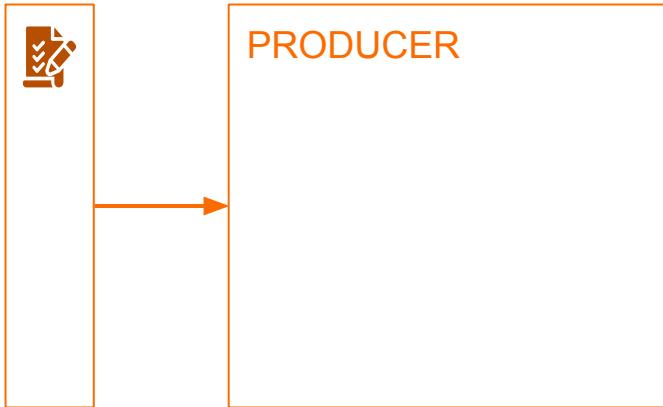




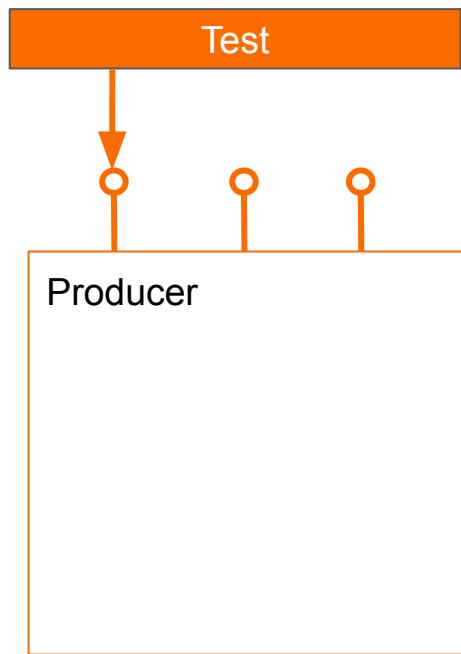




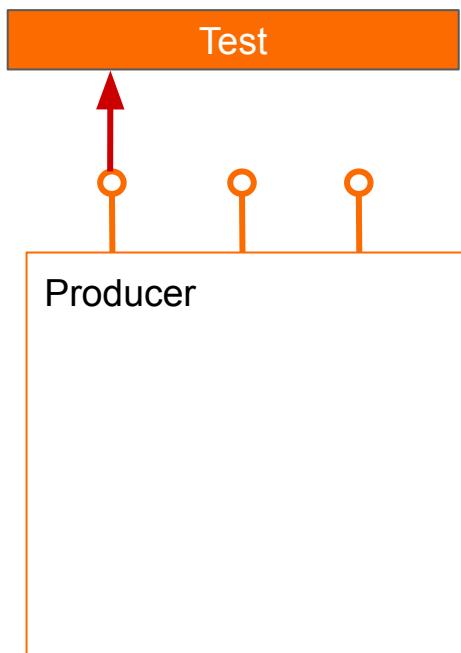
I'm the producer “Service B”
and I want to validate my
contracts



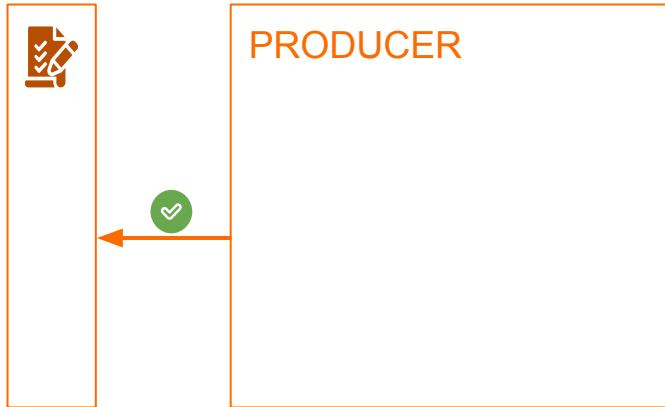
```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```



```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```

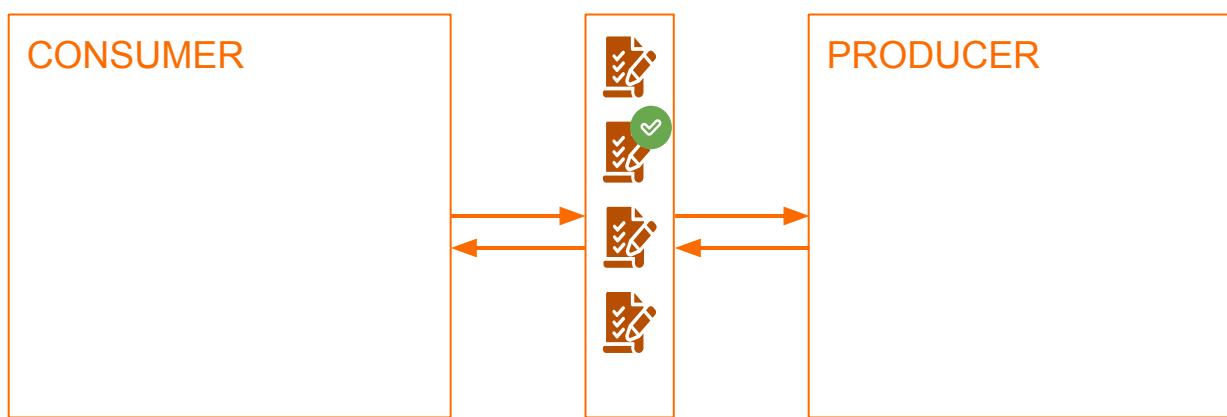


```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```

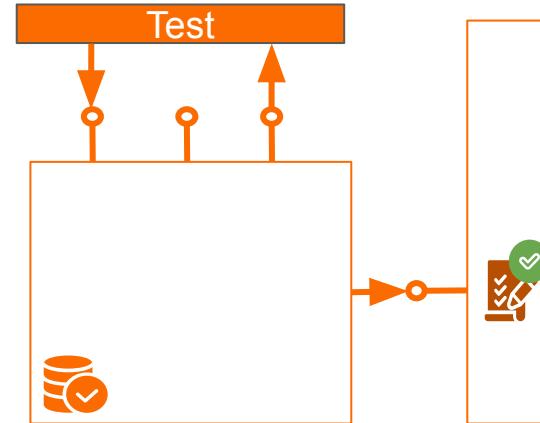


I'm the producer “Service B”, and I've validated this contract with “Service A”.

My Version is 0.0.1, which is already built.



```
{  
  "consumer": "Service A",  
  "provider": "Service B",  
  "interactions": [  
    {  
      "request": {  
        "method": "GET",  
        "path": "/profile/{id}"  
      },  
      "response": {  
        "status": 200,  
        "body": [  
          {  
            "name": "Jane",  
            "lastName": "Doe"  
          }  
        ]  
      }  
    }  
  ]  
}
```





Providers: test the integration with the consumers in the way they have defined.



Consumers: Reliable mocks for component tests. Mocks verified by the producer.



Healthy environments. Build is broken instead of Environment.



Test in local. No environments means that tests can run on developers machine, even before the code is committed to a repository

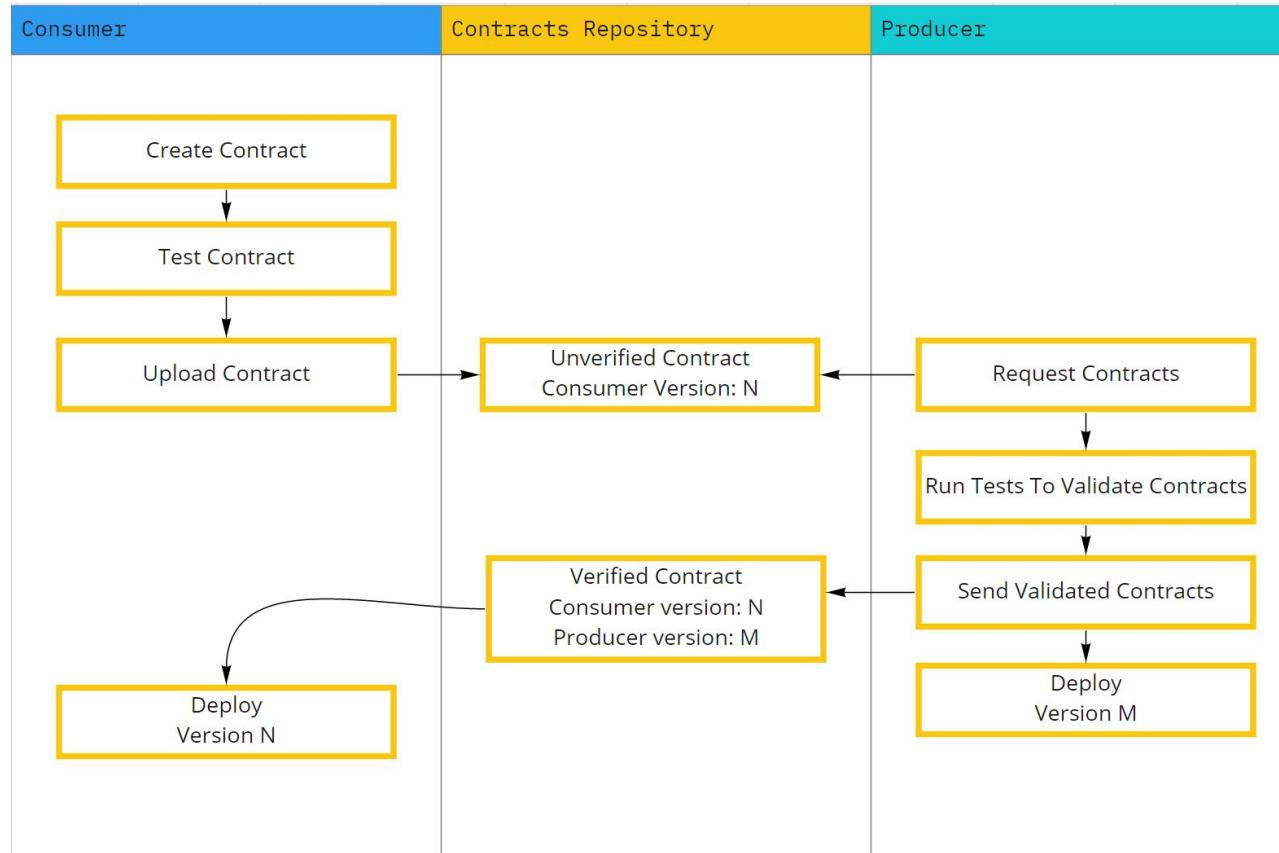


A black and white photograph of a railway track receding into the distance, with wooden sleepers visible along the left side.

04

Consumer Driven Development

CONSUMER DRIVEN DEVELOPMENT

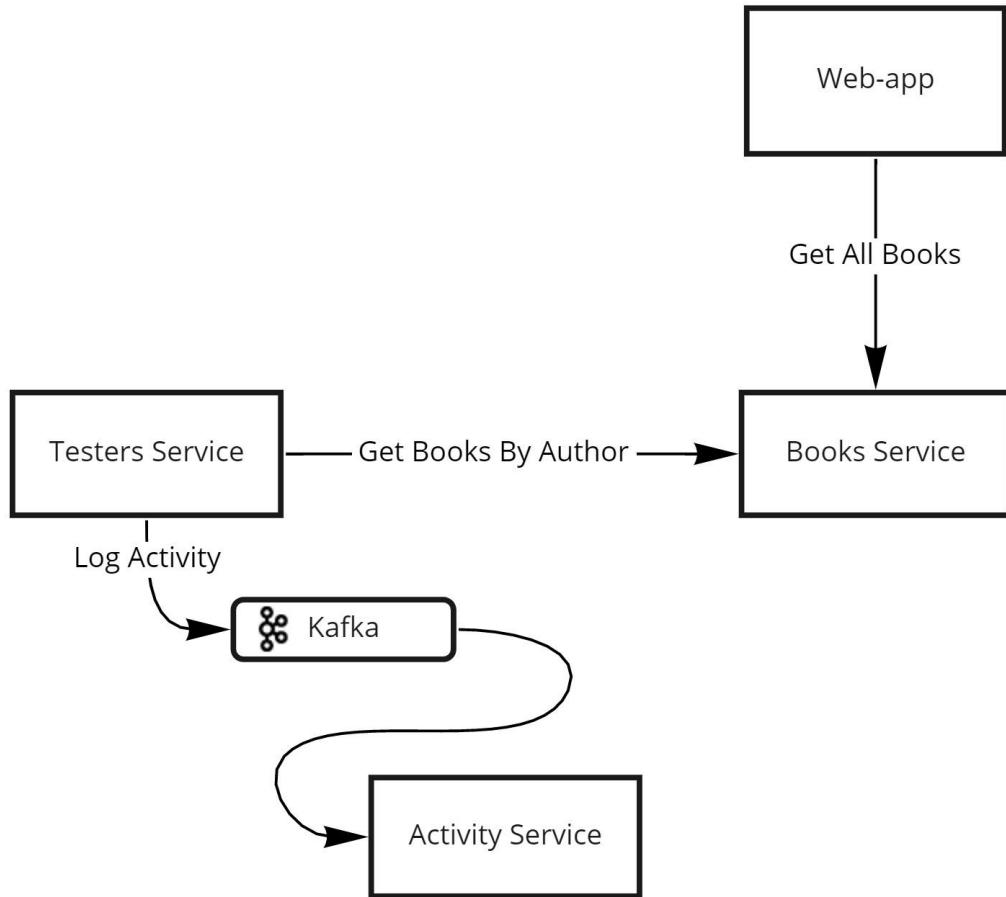


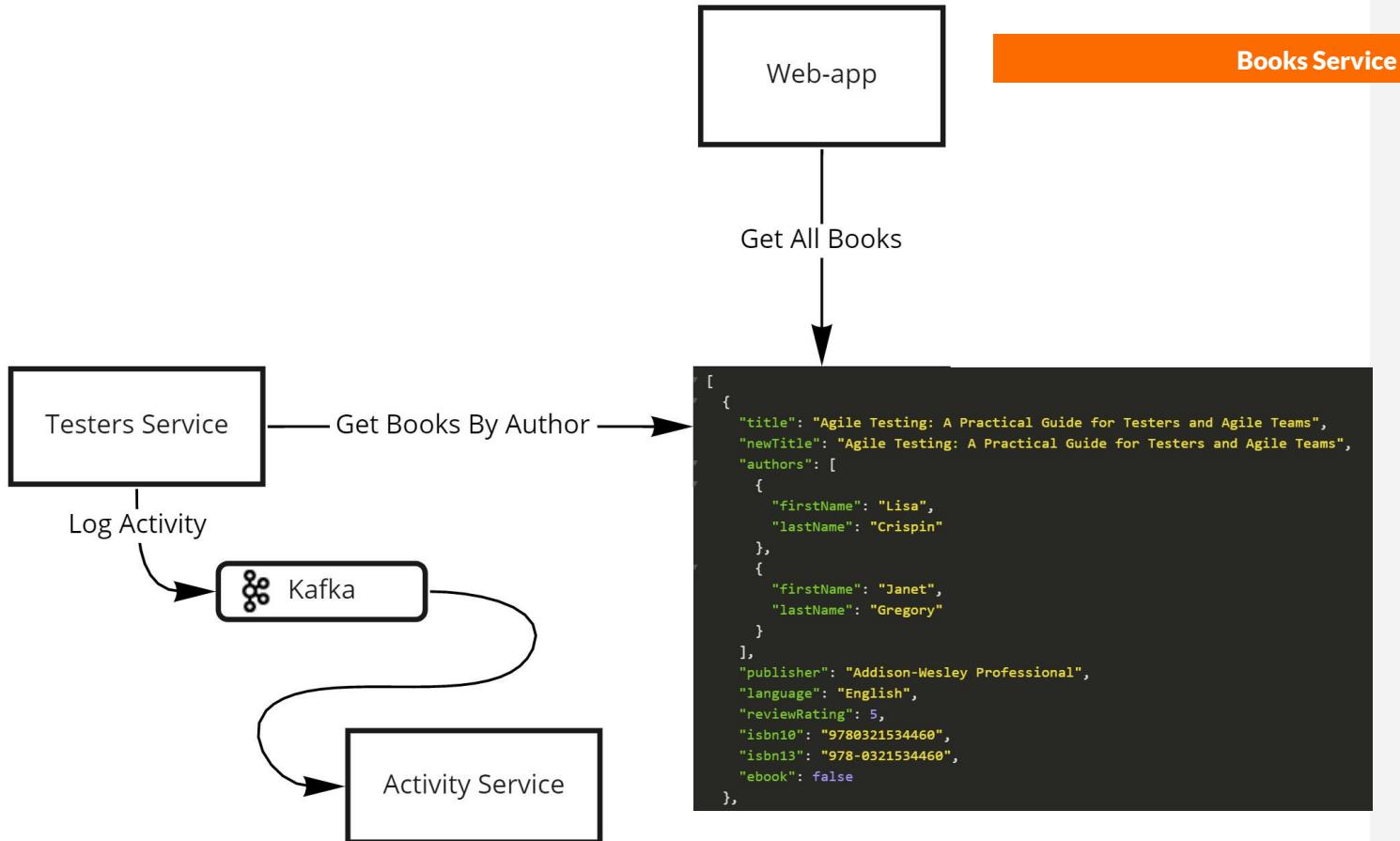


A black and white photograph of a railway track receding into the distance, with wooden sleepers visible along the left side.

05

Demo





Books

English Agile Testing: A Practical Guide for Testers and Agile Teams

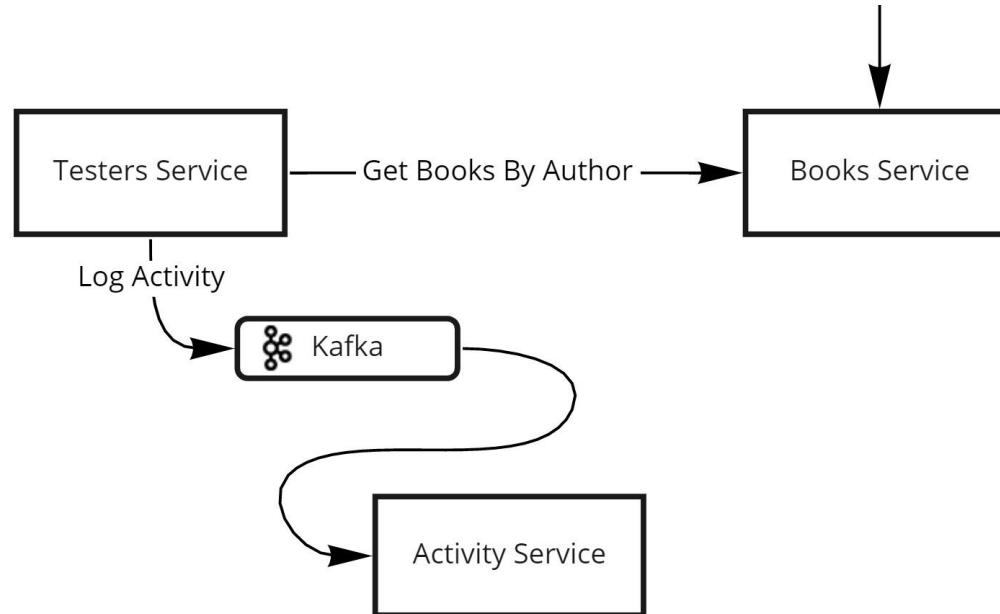
English More Agile Testing: Learning Journeys for the Whole Team

English Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing

Spanish Diseño Ágil con TDD

English A Practical Guide to Testing in DevOps

Web-App



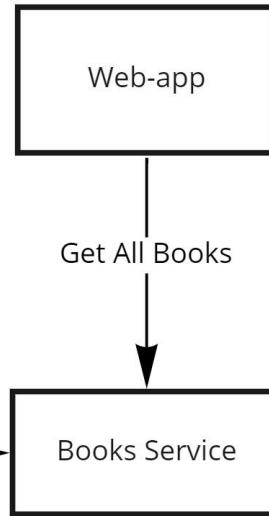
```
{  
  "firstName": "Katrina",  
  "lastName": "Clokie",  
  "twitter": "@katrinaclokie",  
  "books": [  
    {  
      "title": "A Practical Guide to Testing in DevOps"  
    }  
  ]  
,  
  {  
    "firstName": "Angie",  
    "lastName": "Jones",  
    "twitter": "@techgirl1908"  
  },  
  {  
    "firstName": "Kris",  
    "lastName": "Layton",  
    "twitter": "@krislayton"  
  },  
  {  
    "firstName": "Jen",  
    "lastName": "Strickland",  
    "twitter": "@jenstrickland"  
  },  
  {  
    "firstName": "Liz",  
    "lastName": "Carter",  
    "twitter": "@lizcarter"  
  },  
  {  
    "firstName": "Sarah",  
    "lastName": "Hollingshead",  
    "twitter": "@sarah_hollingshead"  
  },  
  {  
    "firstName": "Sara",  
    "lastName": "Layton",  
    "twitter": "@sara_layton"  
  },  
  {  
    "firstName": "Tina",  
    "lastName": "Layton",  
    "twitter": "@tina_layton"  
  },  
  {  
    "firstName": "Vivian",  
    "lastName": "Layton",  
    "twitter": "@vivian_layton"  
  },  
  {  
    "firstName": "Wendy",  
    "lastName": "Layton",  
    "twitter": "@wendy_layton"  
  },  
  {  
    "firstName": "Zoe",  
    "lastName": "Layton",  
    "twitter": "@zoe_layton"  
  }  
}
```

Log Activity



Kafka

Activity Service



Testers Service

Add “isbn10” field to Testers service

Add production code:

1. At dto/**Book** class add the field “String isbn10”

Update Tests:

2. At **BooksClientContractTest**:
 - a. Update contract at **booksAvailable** method. Add the new “isbn10” field below the existing “title” one.
 - b. Run the tests and see them **failing** (as test expectations are not updated to expect the isbn10 field)
 - c. Update **TestExpectations** at the bottom of the class, to include the isbn10 field in the expected book.
 - d. Run the tests and see them **passing**.

Change the Book DTO and see the producer fail

Add production code:

1. At dto/**Book** class rename the field “title” to “newTitle”.

Note: User your IDE refactor utility to rename all places where the field is used. If Not, you need to manually update it “BooksRepository” class and “ContractTest” class.

RunTests:

See how two tests are failed. Locate in the output the two integrations broken:

1. Integration with testers service.
2. Integration with Web-app.

Add “isbn10” field to Web-App

Add production code:

1. At src/app/**book.ts** add the field “isbn10: string”

Update Tests:

2. At src/app/**books.service.pact.spect.ts**:
 - a. Update **expectedBook** (around line 49), to include the isbn10 field in the expected book.
 - b. Run the tests and see them **failing** (as test contract is not updated with the new field)
 - c. Update contract at **provider.addInteraction**. Add the new “isbn10” field below the existing “title” one.
 - d. Run the tests and see them **passing**.



<https://github.com/romansegador/web-app>



<https://github.com/romansegador/testers-service>



<https://github.com/romansegador/books-service>



<https://github.com/romansegador/activity-service>



<https://github.com/romansegador/infrastructure>



THANKS!

Do you have any questions?



<https://www.linkedin.com/in/romansegador/>



@roman_s_t

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

Please keep this slide for attribution.