Consumer Driven Contracts
A hands-on guide to Spring Cloud Contract

About me

Spring Cloud developer at Pivotal
working mostly on:

- Spring Cloud Sleuth
- Spring Cloud Contract
- Spring Cloud Pipelines

Connect with me

- Twitter: **@mgrzejszczak**
- Blog: http://toomuchcoding.com

Why Contract Tests Matter

Let’s get it started!

start.spring.io

Poll

Have you ever used WireMock?

Answers:

- yes
- no

Poll

Will the tests…

Answers:

- pass
- fail
- I don’t care

Thumbs up / down

Does it make sense?

Thumbs up / down

Does it make sense?

QUESTIONS ?

BREAK

Doing Consumer Driven Contracts with Spring Cloud Contract

What we will NOT be talking about?

- Schema
- WSDL
- ESB
- XSD
- XSLT

THE IDEA OF SPRING CLOUD CONTRACT IS NOT TO INTRODUCE UNNECESSARY COUPLING OR REPLICATE OLD MISTAKES

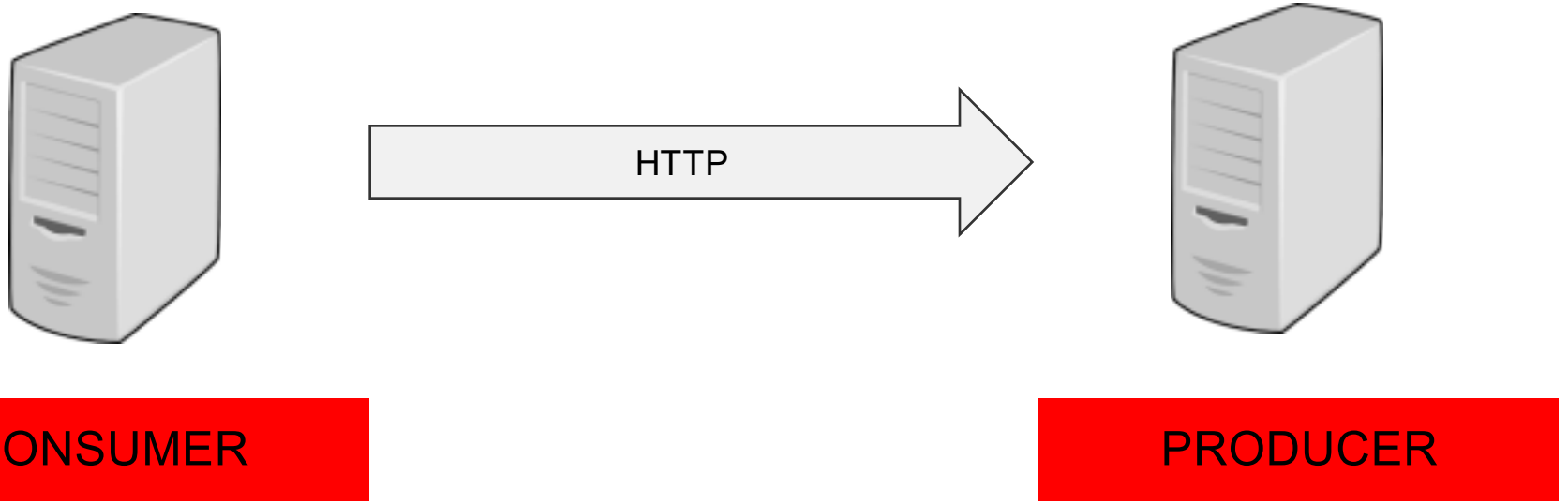
Terminology

- Producer
  - service that exposes an API / sends a message
- Consumer
  - service that consumes the API of the producer / listens to a message from the producer
- Contract
  - agreement between producer and consumer how the API / message will look like
- Consumer Driven Contracts
  - approach where the consumer drives the changes of the API of the producer

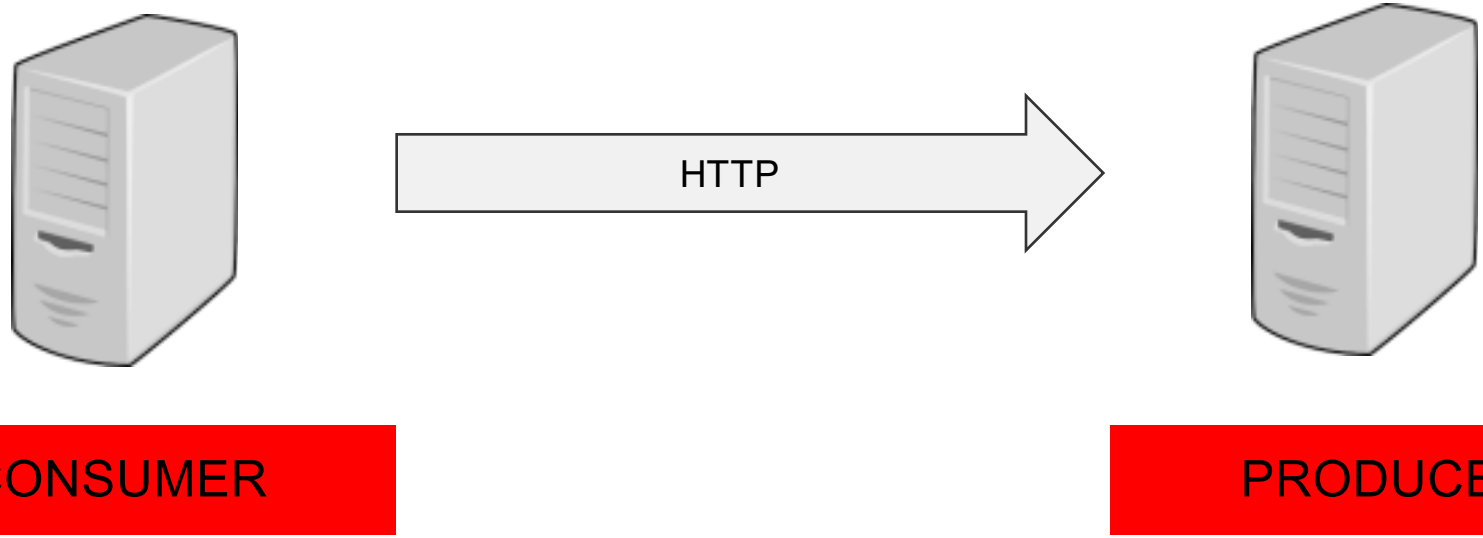
What problems are we trying to solve?

- Stub validity & reusability in the integration tests
- Nice API creation

Typical situation



How to write a test for it?



How to write a test for it?

HTTP

CONSUMER

PRODUCER

Stub validity & reusability

pivotal-calculator-service / stub / src / main / resources / testdata / **mappings** /

This branch is even with develop.

..	
post.json	Integration test with history service
post_bad_request.json	Update testdata
put.json	
put_not_found.json	Integration test with history service

CONSUMER OWNS THE STUB DEFINITIONS

Stub validity & reusability

```
{
  "priority": 99,
  "request": {
    "method": "POST",
    "url": "/history"
  },
  "response": {
    "status": 201,
    "headers": {
      "Content-Type": "application/json"
    }
  }
}
```

```
{
  "priority": 1,
  "request": {
    "method": "POST",
    "url": "/history",
    "bodyPatterns": [
      {
        "contains": "6+6"
      }
    ]

  },
  "response": {
    "status": 400
  }
}
```

Stub validity & reusability

Let's assume that we're introducing a new endpoint...

Stub validity & reusability

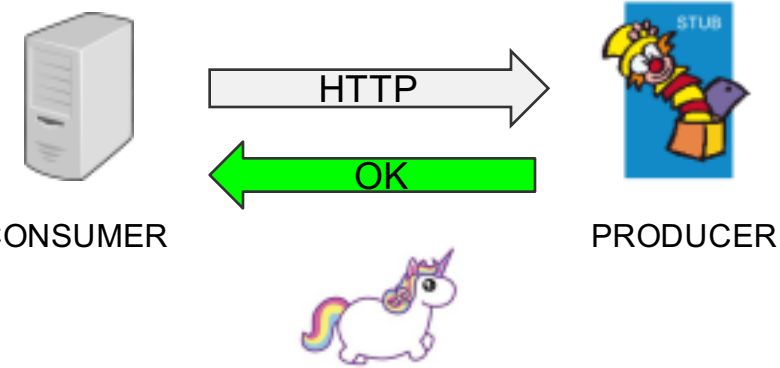
```
{
  "priority": 99,
  "request": {
    "method": "POST",
    "url": "/someNonExistantUrl"
  },
  "response": {
    "status": 201,
    "headers": {
      "Content-Type": "application/json"
    }
  }
}
```

Stub validity & reusability

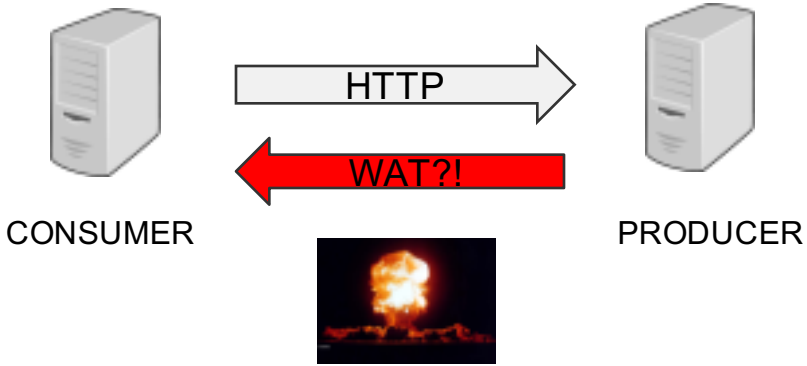
- I write the missing code to work with the new URL
- The unit and integration tests pass
- Now we deploy to an environment where real integrations take place...

Stub validity & reusability

TEST



REALITY



Stub validity & reusability

TEST

topic:
bar

PRODUCER

CONSUMER

REALITY

topic:
foo

topic:
bar

PRODUCER

CONSUMER

Stub validity & reusability

- Stubs reside with the consumer
- Consumer controls the stubbing process
- How are you sure that the stubs are valid?
- What if other teams want to reuse those stubs?
- Do you even know who is calling your API?
- How are you sure you're listening to the proper topic / queue?
- Are you sure you'll deserialize your message properly?

Nice API Creation

- It’s the consumer that uses the API
- Consumers should take part in the creation of the API of the producer
- The producer’s API change should be driven by consumers

Nice API Creation

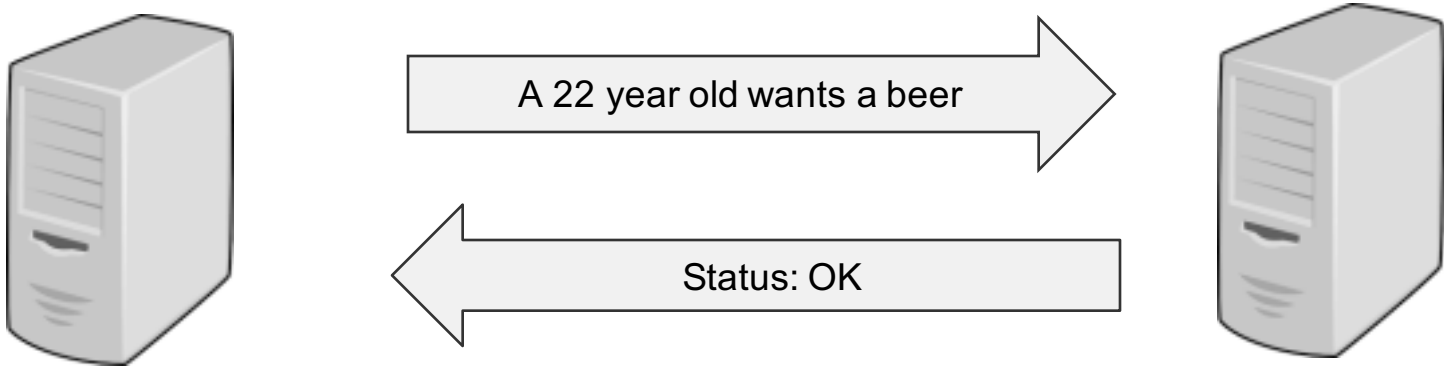
Thumbs up / down

- Does it make sense?

What are we going to code?

- Consumer
  - service that gets beer requests
  - has to ask another service if the client can get the beer
- Producer
  - service that checks if the client is old enough to buy beer
- Feature
  - if the user is too young - the beer will not be sold
  - otherwise the beer will be granted

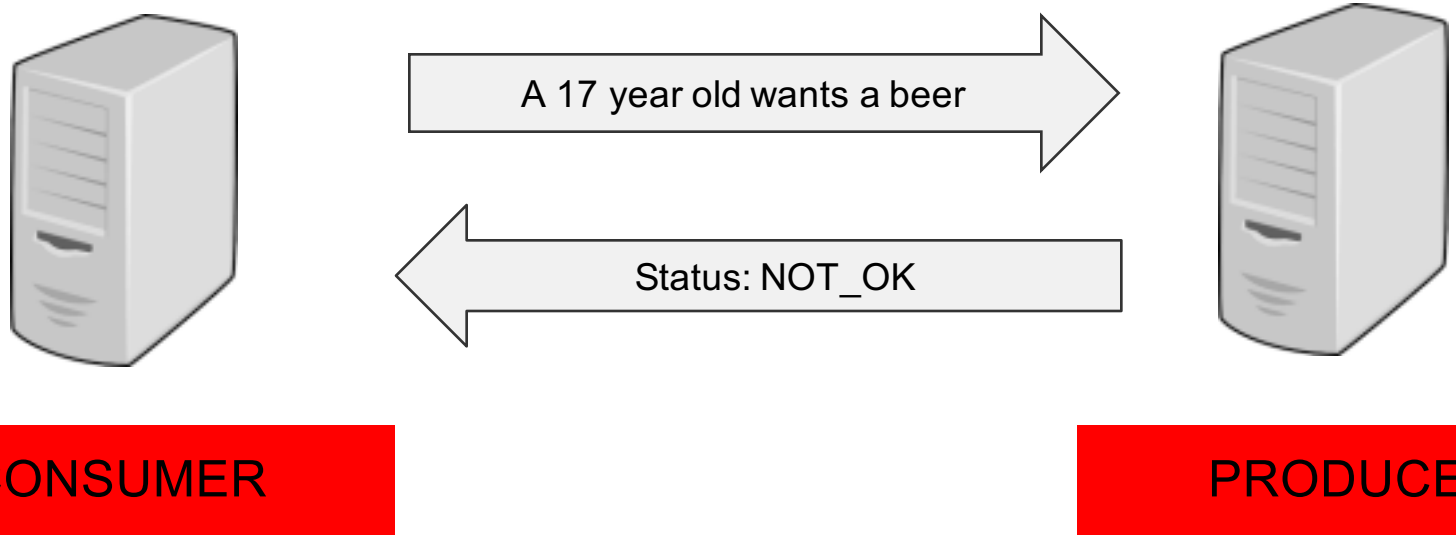
What are we going to code?



CONSUMER

PRODUCER

What are we going to code?



What are we going to code?

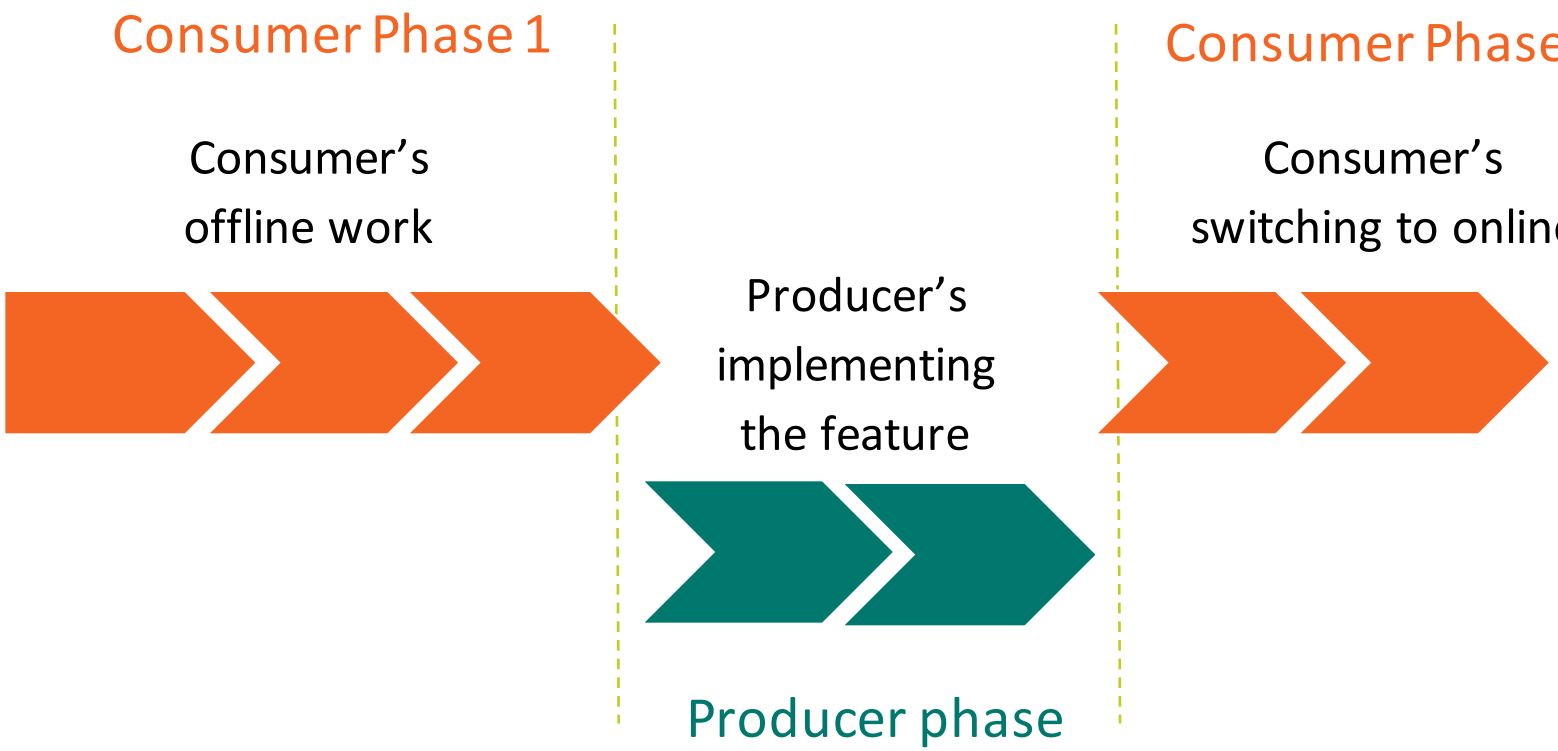
CONSUMER

WHITE TERMINAL
WHITE IDE

PRODUCER

BLACK TERMINAL
BLACK IDE

What are we going to code?

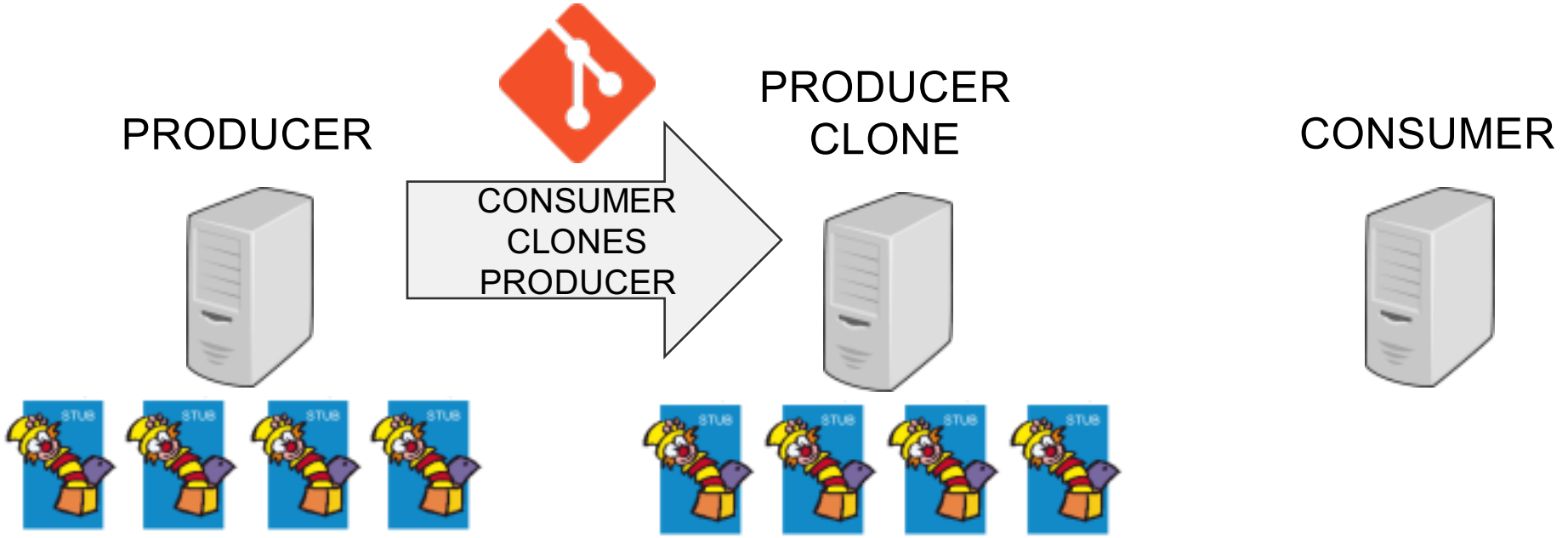


Consumer phase 1

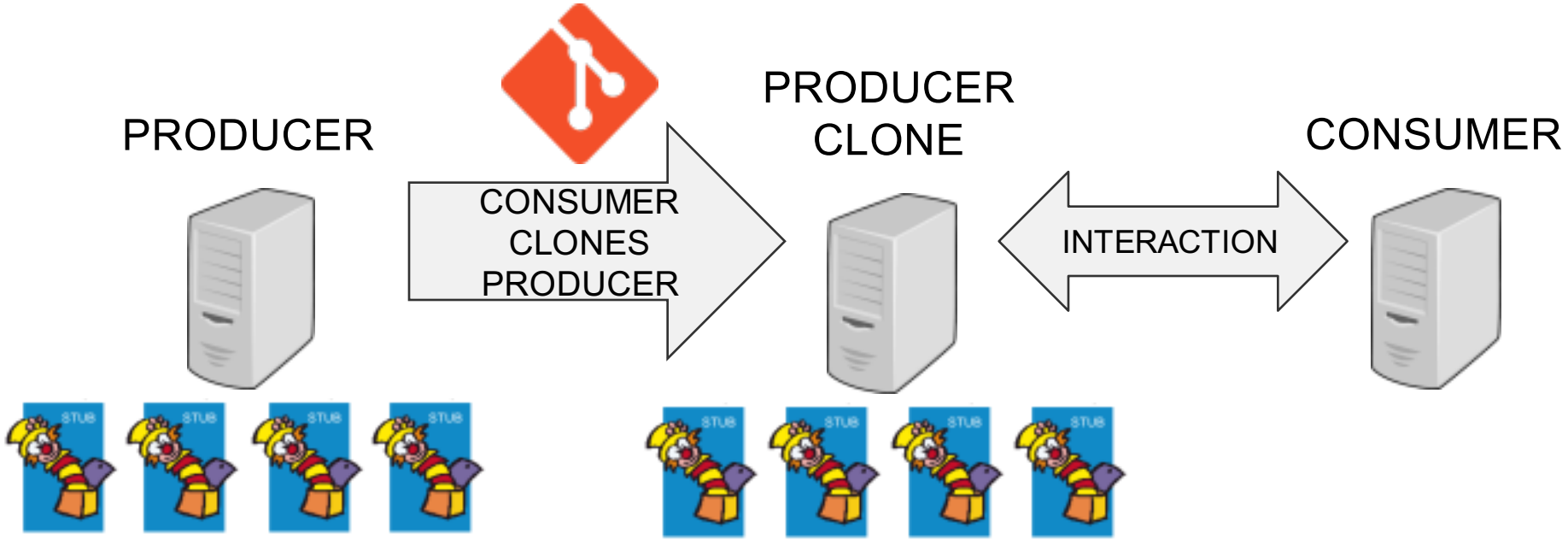
PRODUCER

CONSUMER

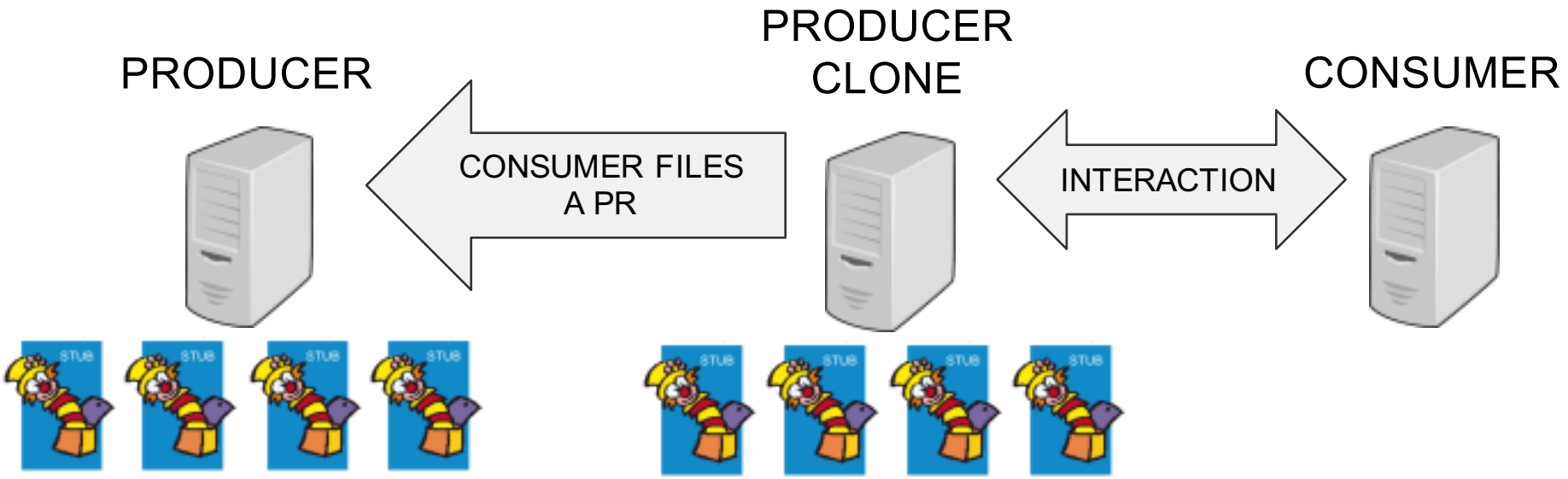
Consumer phase 1



Consumer phase 1



Consumer phase 1



Thumbs up / down

- Does it make sense?

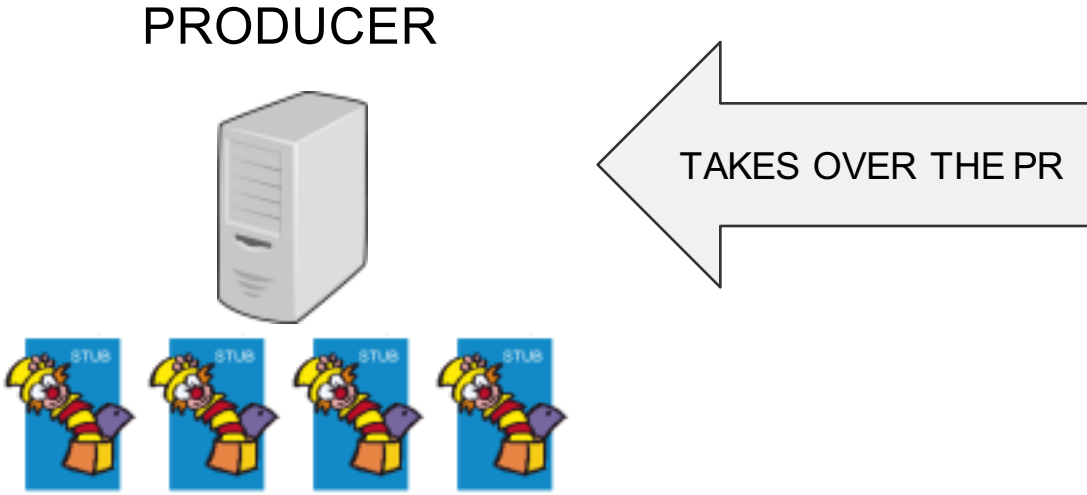
Consumer phase 1

CODE

Thumbs up / down

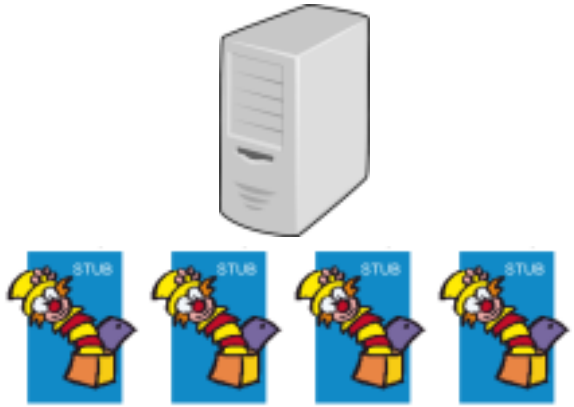
- Does it make sense?

Producer phase



Producer phase

PRODUCER



TAKES OVER THE PR

WRITES THE MISSING IMPLEMENTATION

Producer phase

Nexus

JFrog Artifactory

PACTS

PRODUCER

MERGES TO MASTER
CI UPLOADS THE
STUBS

TAKES OVER THE PR

WRITES THE MISSING
IMPLEMENTATION

Thumbs up / down

- Does it make sense?

Producer phase

CODE

Thumbs up / down

- Does it make sense?

Consumer phase 2

Nexus

JFrog Artifactory

PACT

DOWNLOADS STUBS OF COLLABORATORS DURING BUILD PHASE

CONSUMER

SWITCHES TO ONLINE MODE

Thumbs up / down

- Does it make sense?

Summary

- we've created an API that suits the consumer and the producer
- expectations were defined by readable contracts
- expectations were tested against the producer
- producer stubs can be reused by consumers
- starting and setting stubs is fully automated

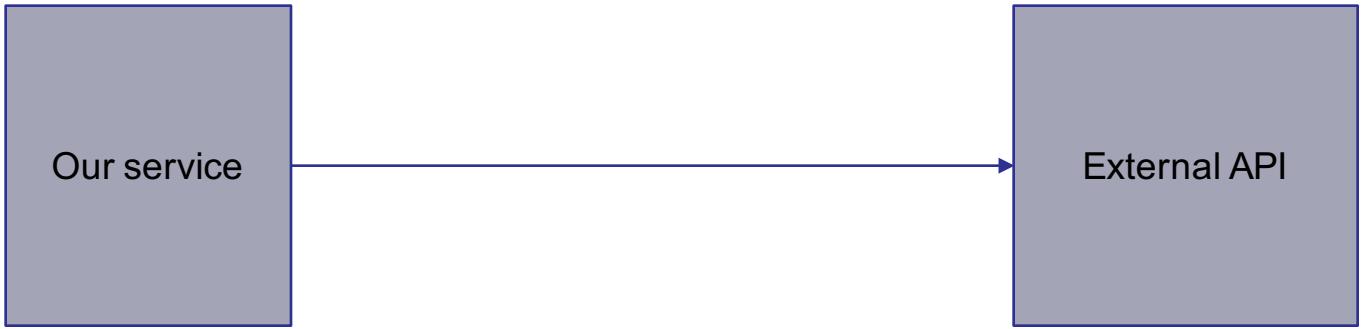
QUESTIONS ?

BREAK

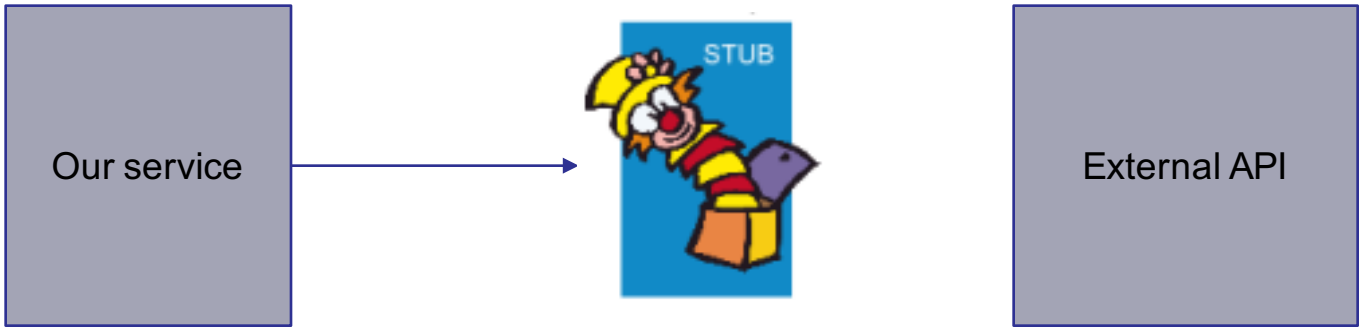
Spring Cloud Contract Advanced

Stubbing Services You Don’t Own

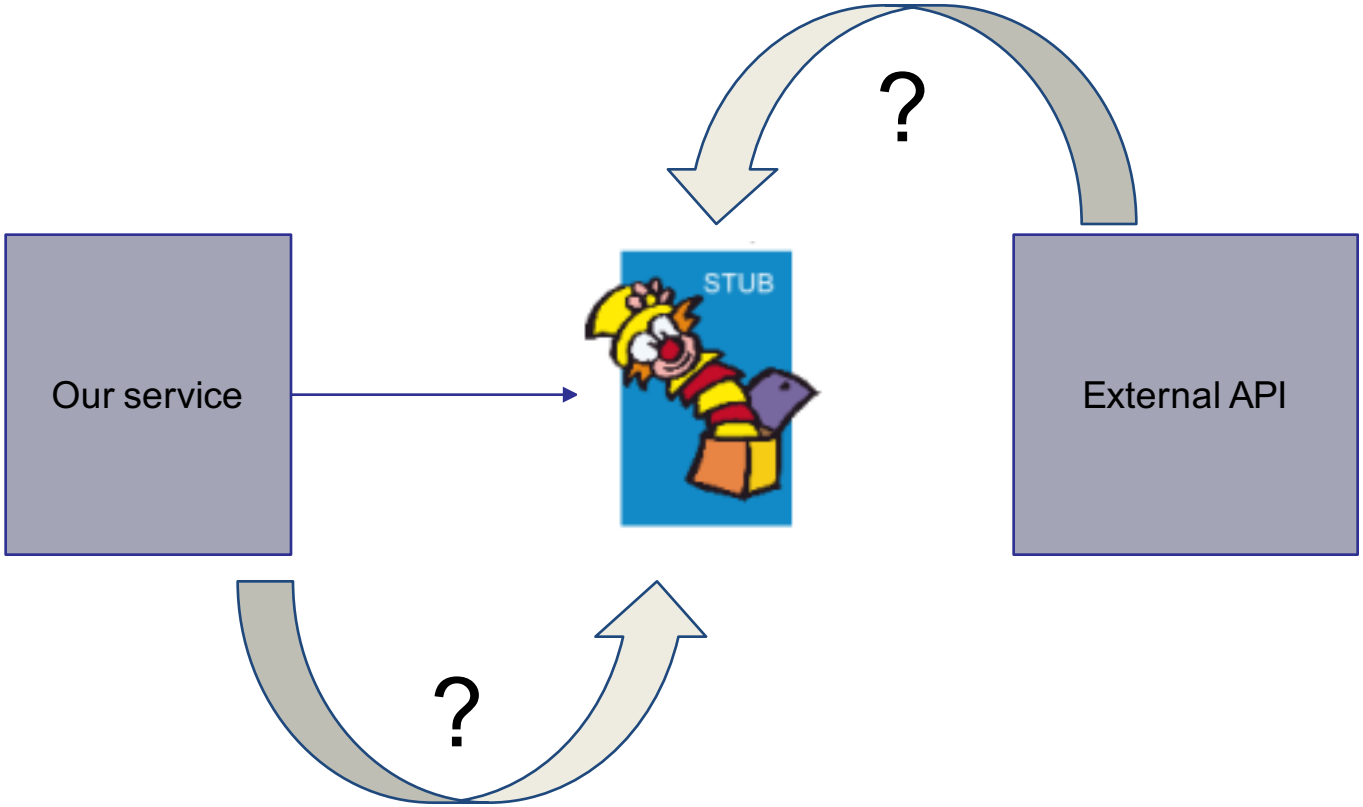
Stubbing services you don’t own



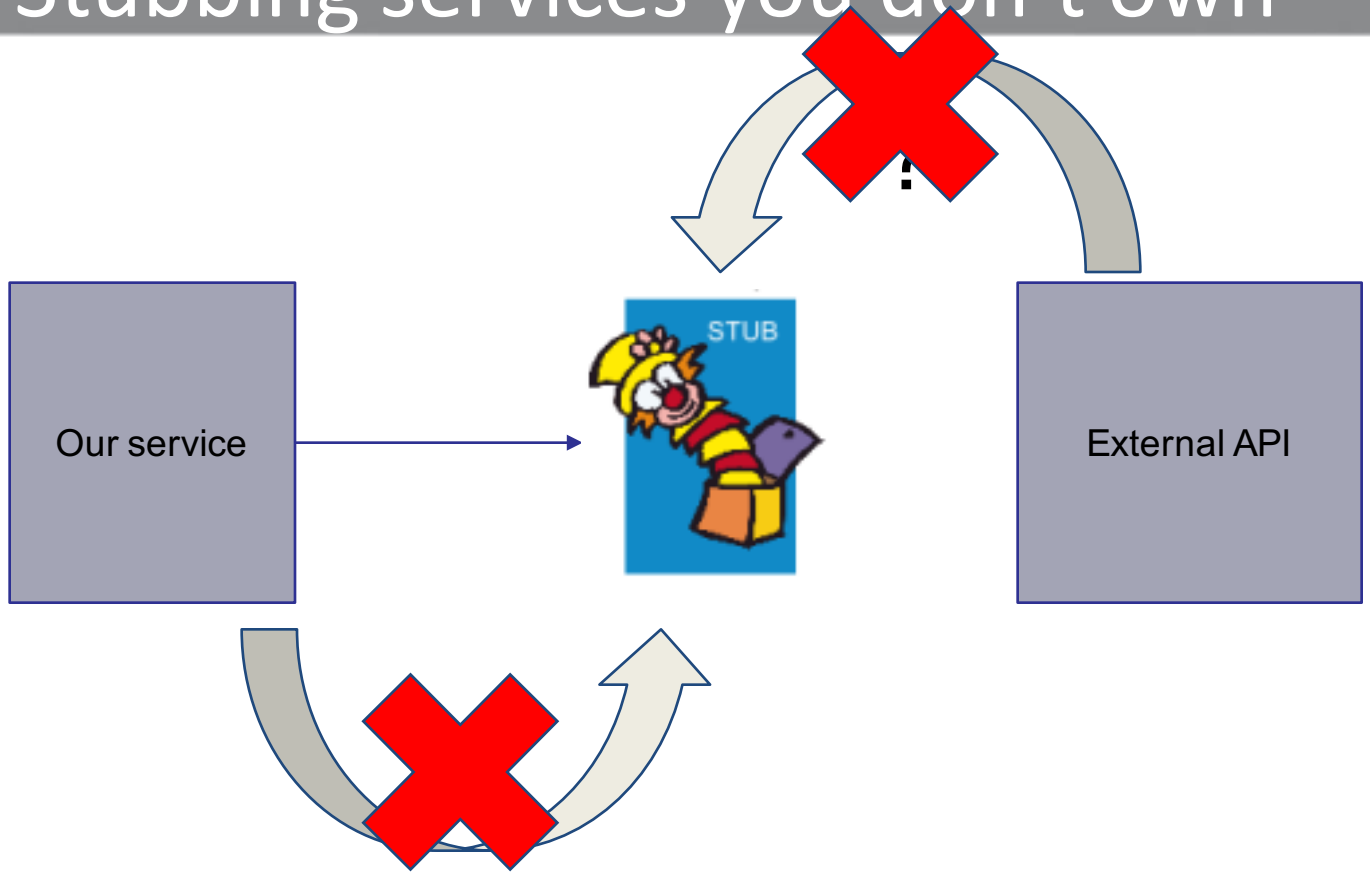
Stubbing services you don’t own



Stubbing services you don’t own



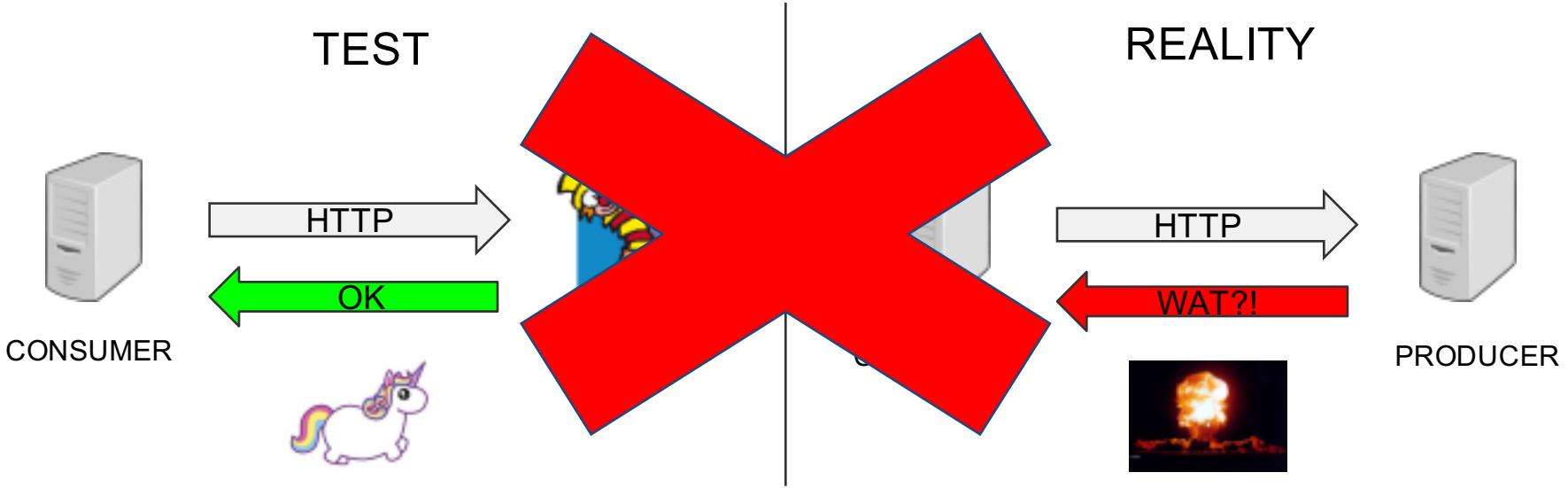
Stubbing services you don't own



Stubbing services you don’t own

How about writing the stub manually?

Stubbing services you don't own



Stubbing services you don’t own

What is the source of truth?

Poll

Client app
The external system
Neither

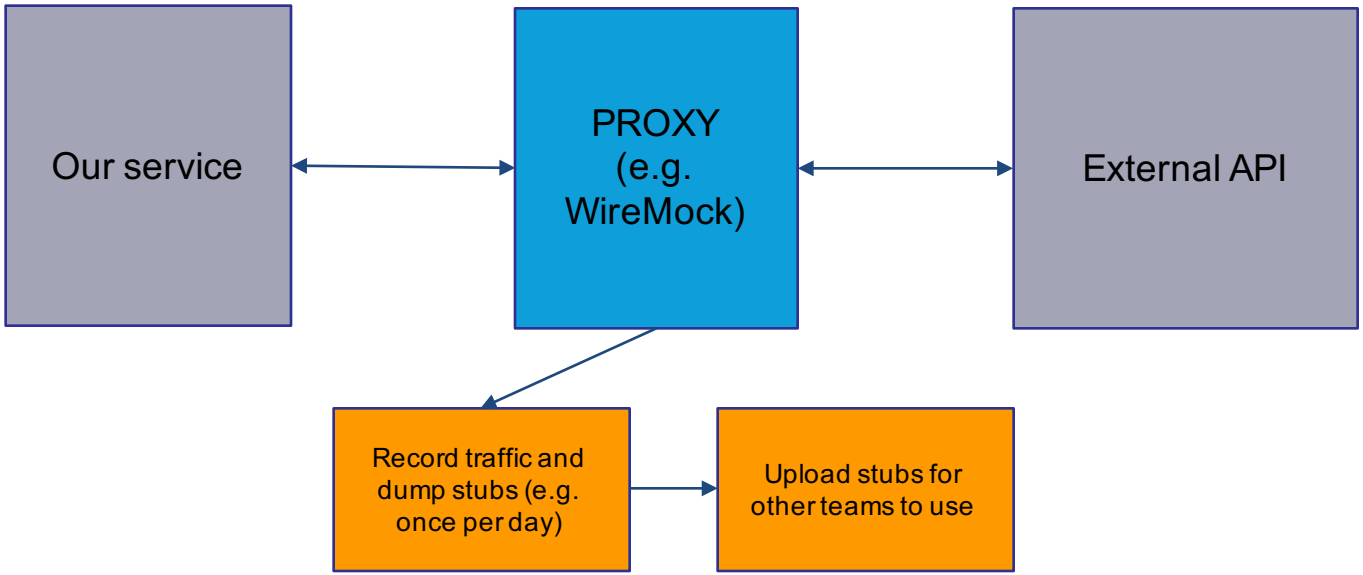
Stubbing services you don’t own

The external system!

Stubbing services you don’t own

Let’s call the external service and record the conversation

Stubbing services you don’t own



Stubbing services you don’t own

CODE

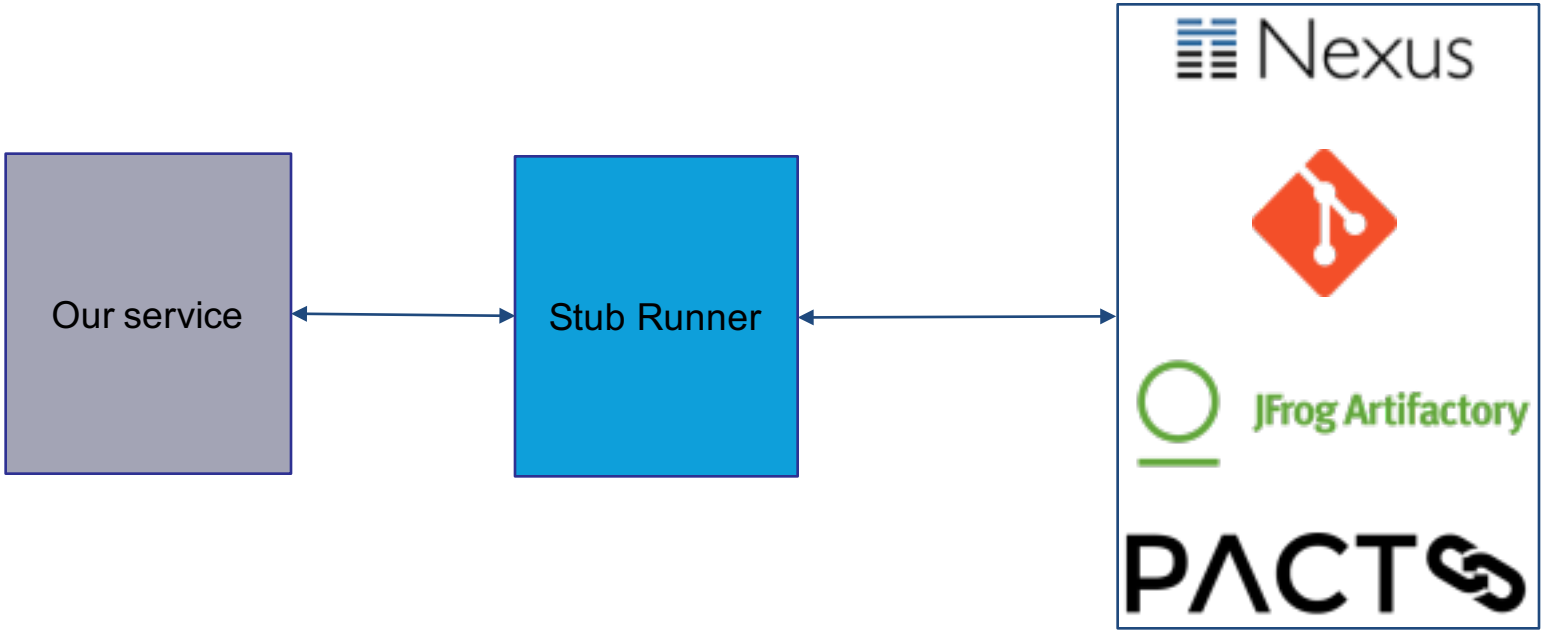
Thumbs up / down

- Does it make sense?

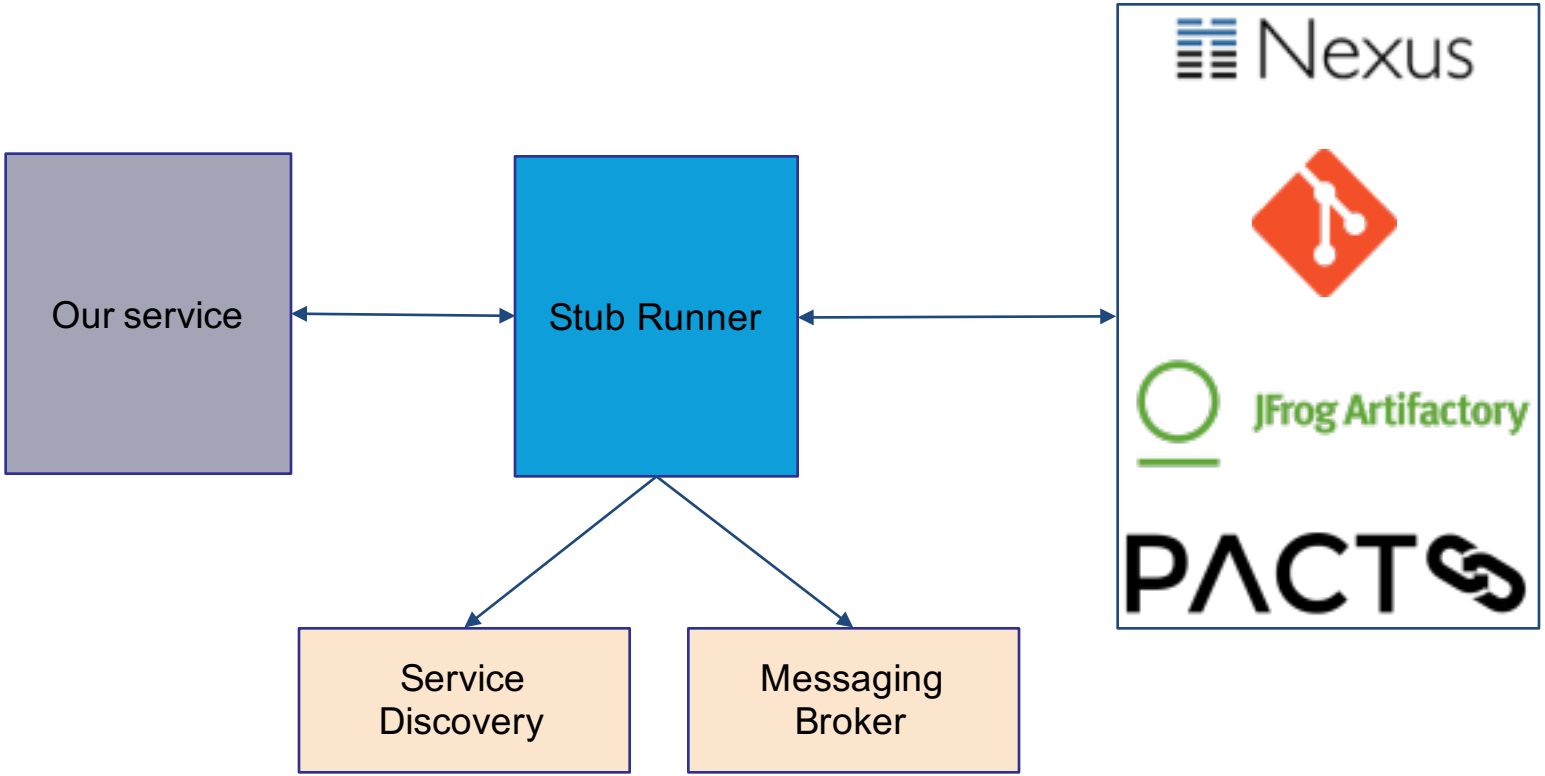
Running stubs in standalone mode

How to use the stubs if you don’t use Java?
How to use the stubs against a running app ?

Running stubs in standalone mode



Running stubs in standalone mode



Running stubs in standalone mode

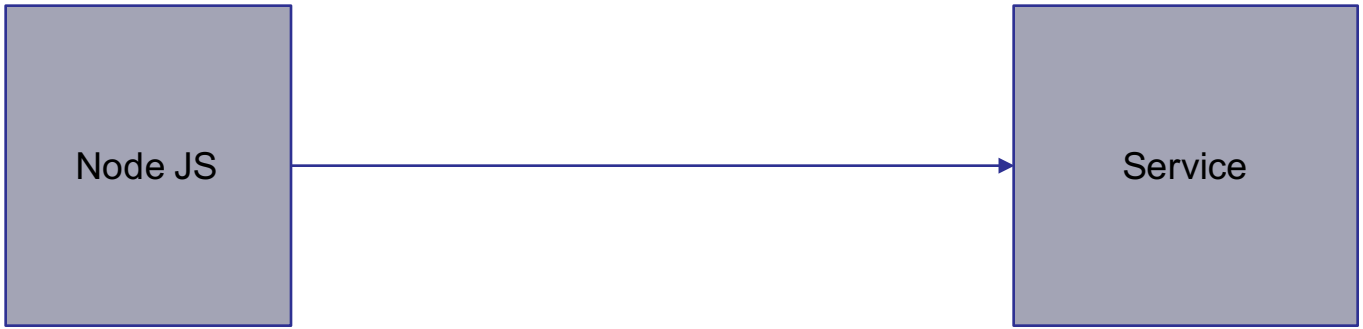
CODE

Starting stubs for a polyglot application

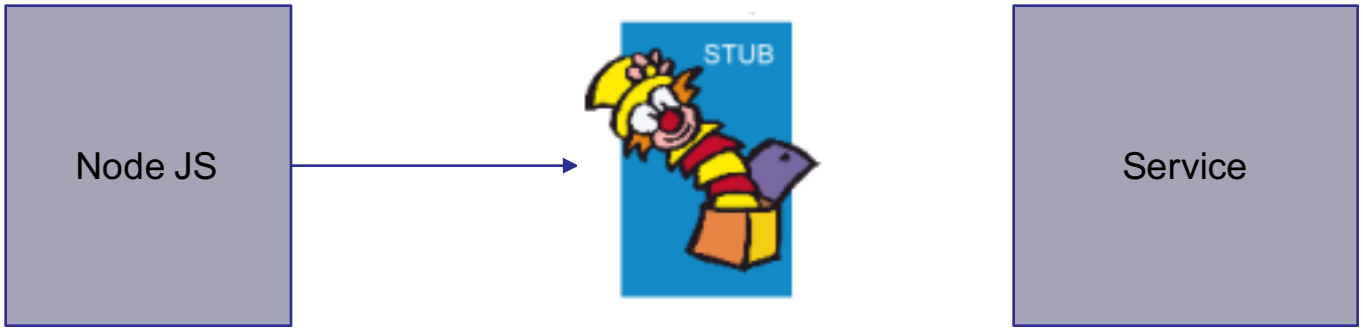
Poll

- Do you know NodeJS ?

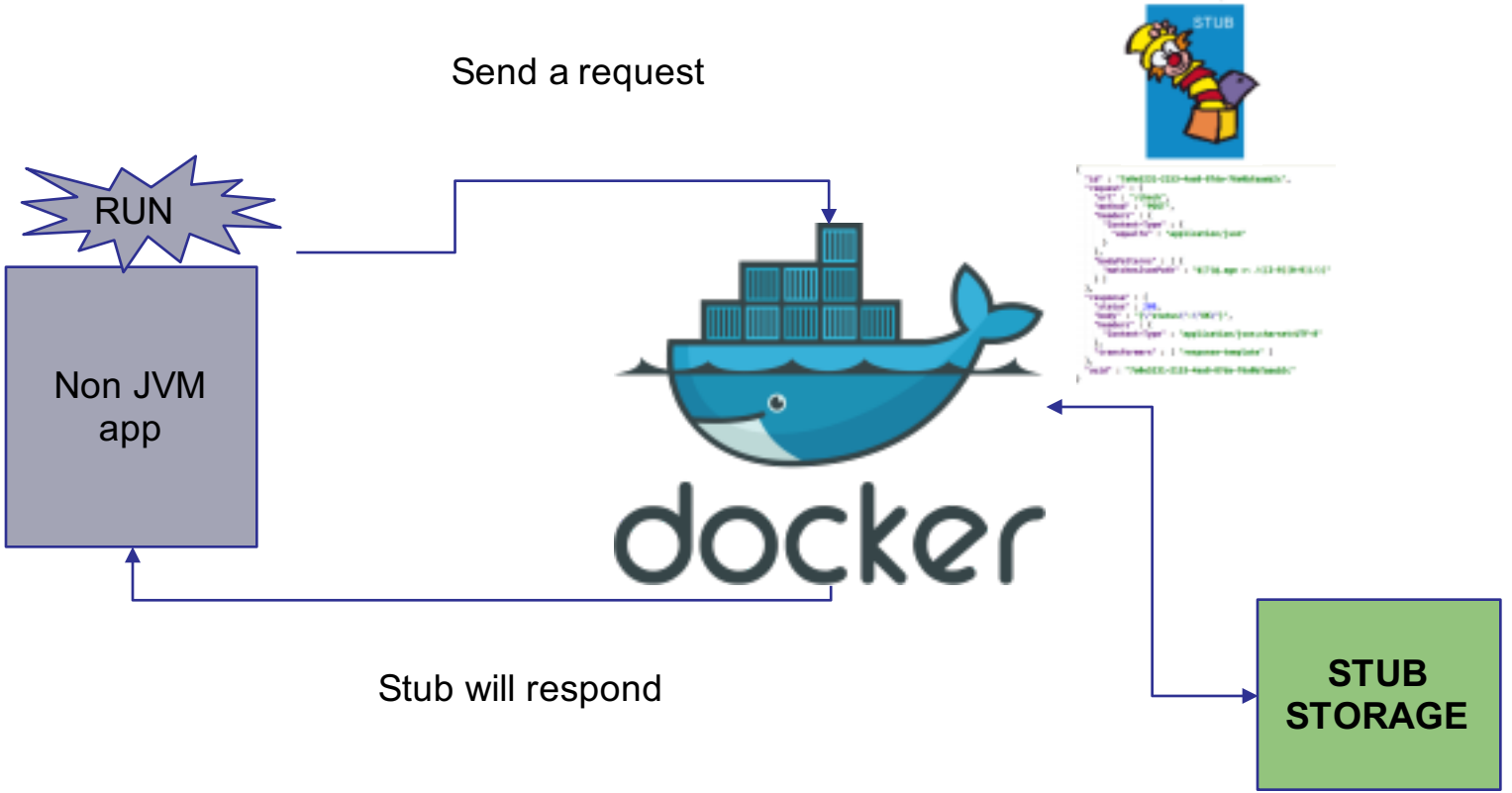
Starting stubs for a polyglot application



Starting stubs for a polyglot application



Starting stubs for a polyglot application



Starting stubs for a polyglot application

CODE

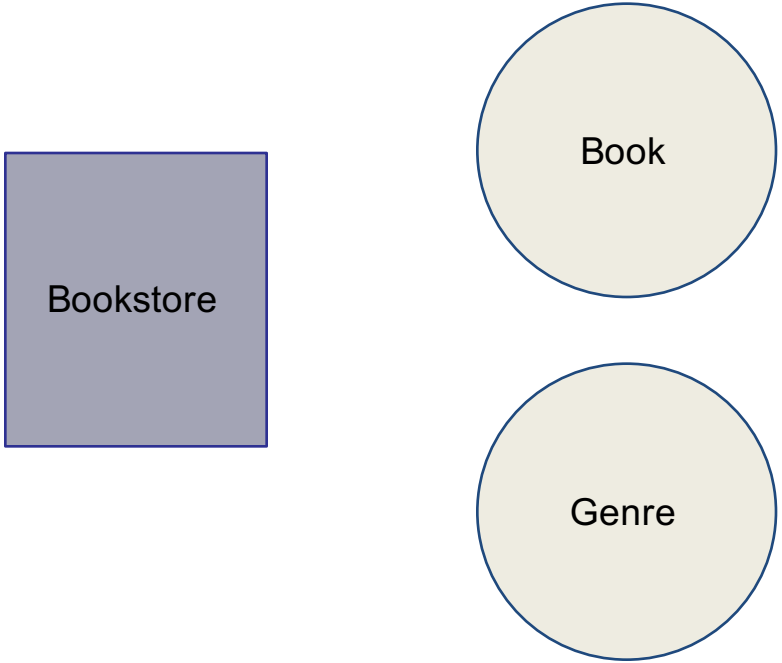
Poll

Which option do you prefer?

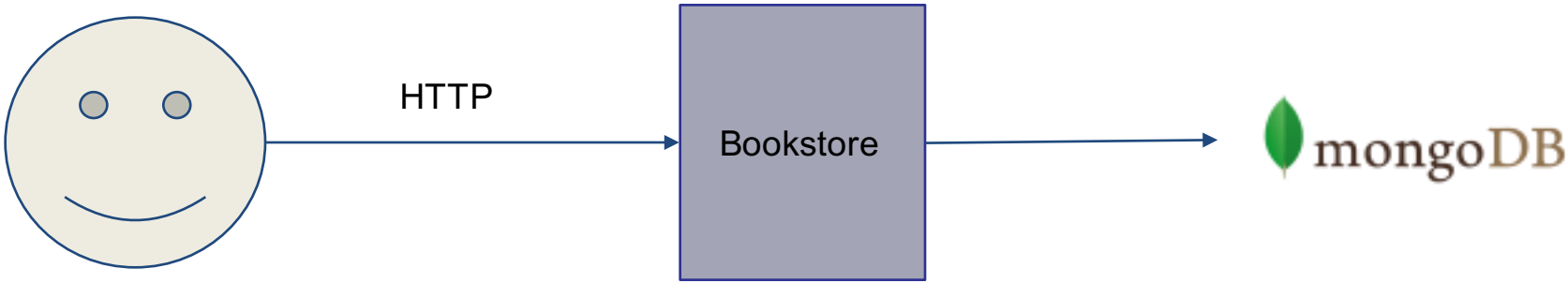
- Spring Cloud CLI
- Running `java -jar`
- Docker image

Generating tests for polyglot applications

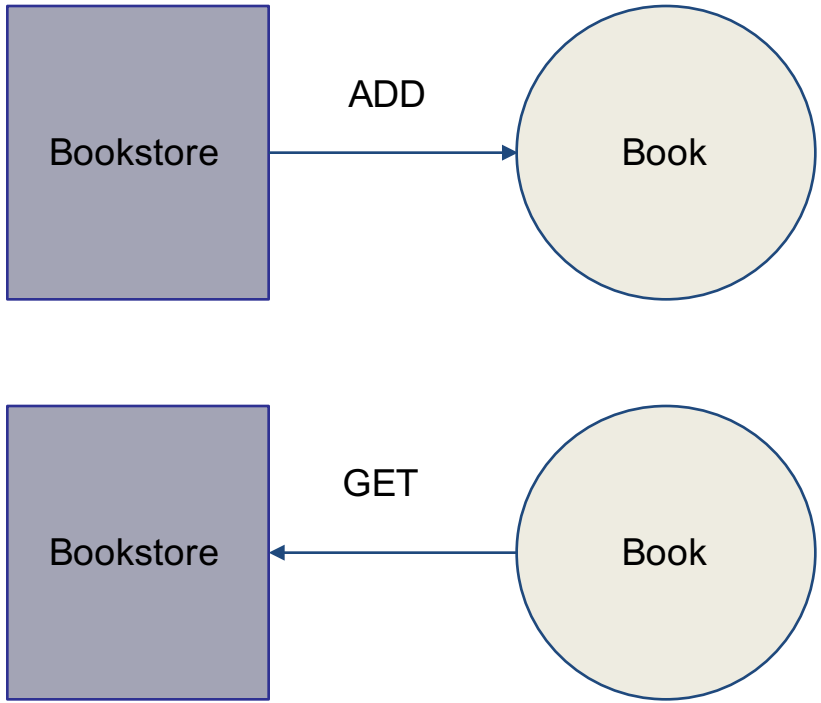
Generating tests for polyglot application



Generating tests for polyglot application

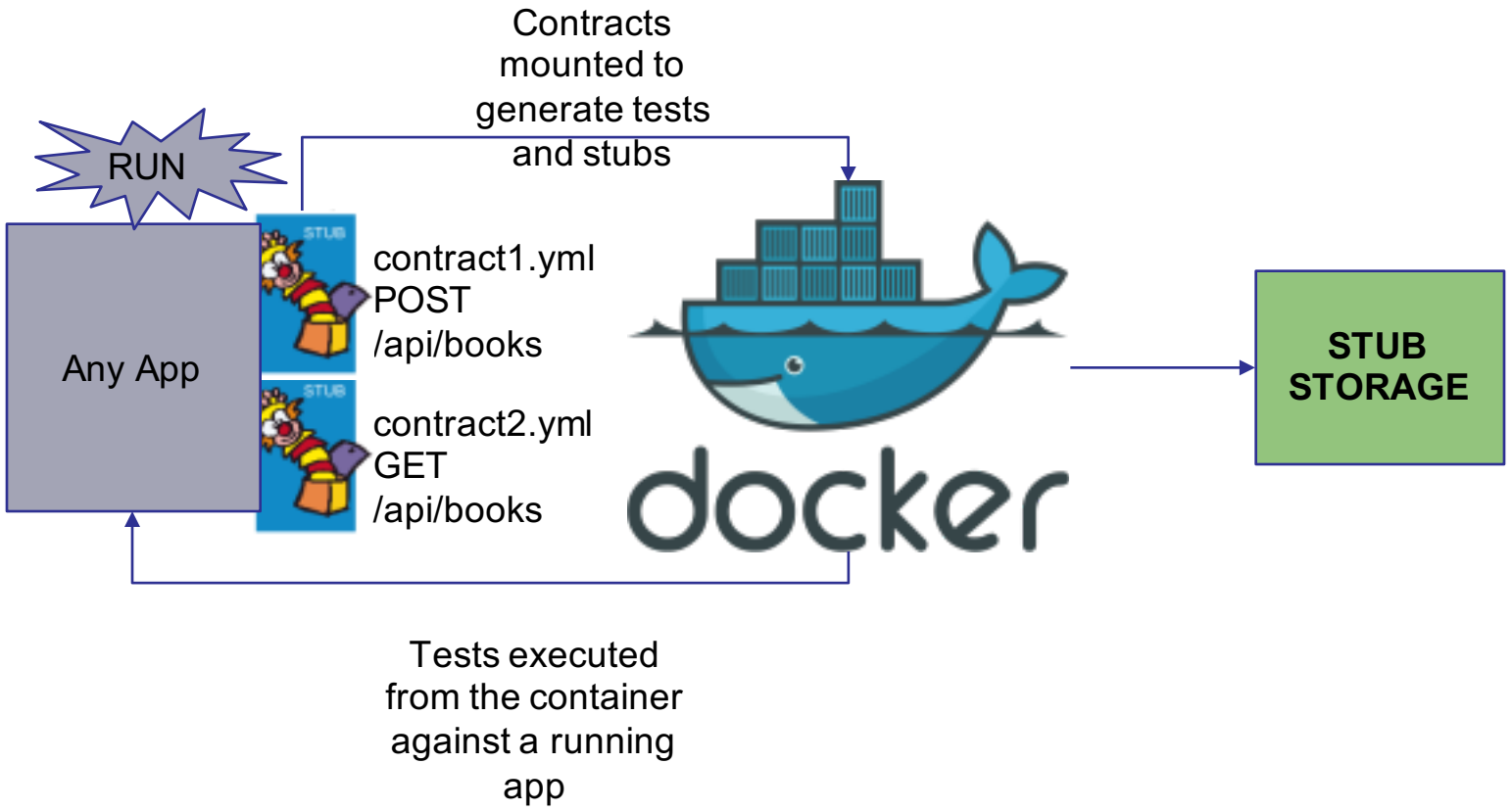


Generating tests for polyglot application



Spring Cloud Contract Docker

Generating tests for polyglot application



Thumbs up / down

- Does it make sense?

QUESTIONS ?