MUTATION AND CONTRACT TESTING

CodeteCON #4
Kraków 2018
Mateusz Bryła

AGENDA

Mutation tests

- Concept
- Principle
- Setup
- Summary

Contract tests

- Concept
- Principle
- Setup
- Summary

Can’t make it?

https://medium.com/@mateusz.bryla

https://codete.com/blog/mutation-and-contract-testing/

https://github.com/mbryla

Why?

MUTATION TESTING

first **proposed** in 1971 (Richard Lipton)

A. Jefferson Offutt and R. H. Untch “Mutation 2000: Uniting the Orthogonal”

first **implemented** in 1980 (Timothy Budd)

T. A. Budd, “Mutation Analysis of Program Test Data”

BETONDEMONTAGETECHNI
LOVE& PEACE
IN
WEST& EAST
Test
West

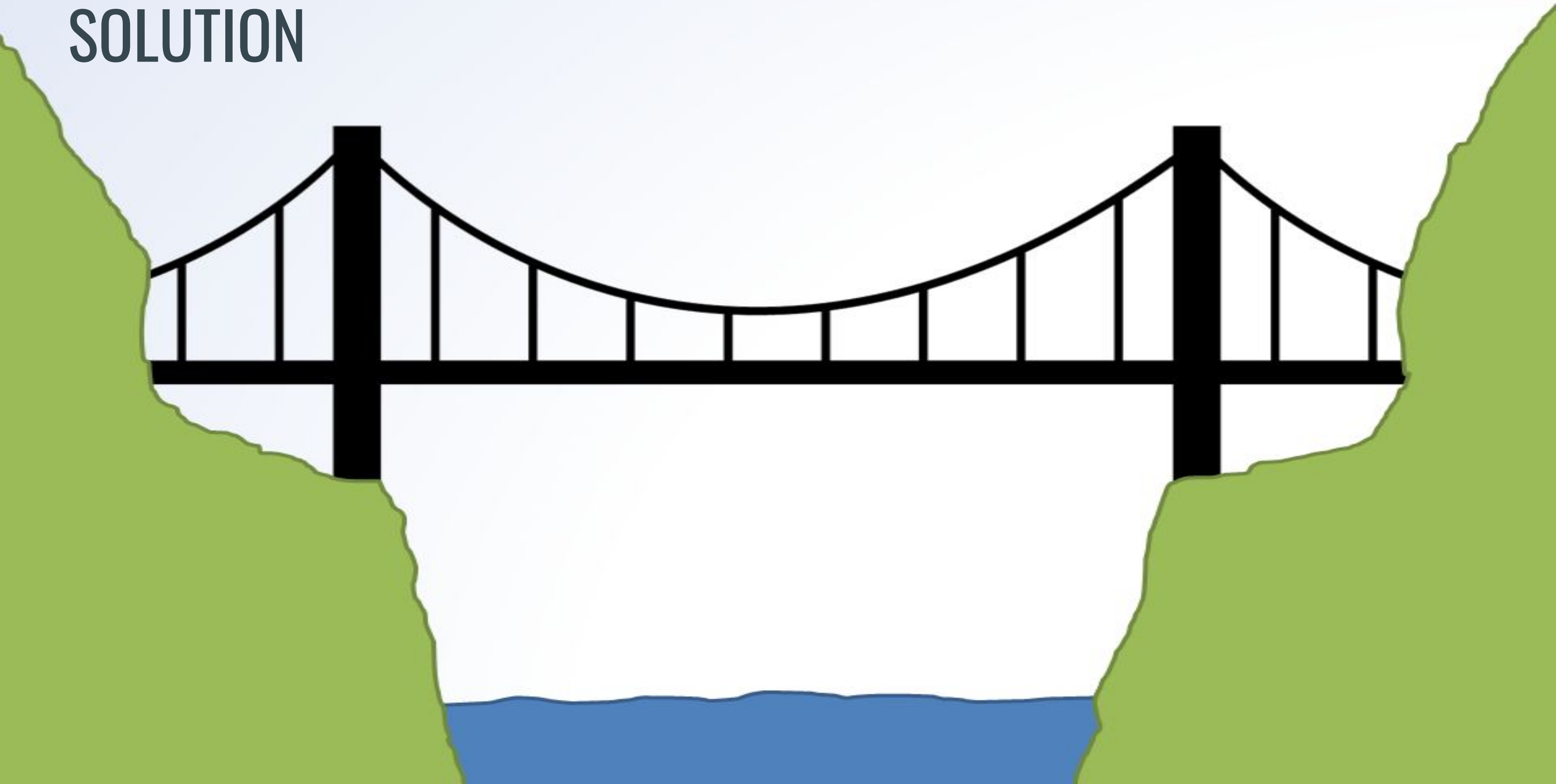
PUSH TO OPEN
LASER BEAM
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ERROR
PROGR
TRACK
PAUSE
REPEAT
TRACK PROGRAM SELECTOR
ON/OFF
PLAY
PAUSE
STOP
NEXT
CM
SELECT
STORE
CANCEL
REPEAT
REV
FWD
SEARCH
STEREO
Symphonien Nos 8 & 9 · Aus der Neuen Welt
»From the New World« · Du Nouveau Monde
BERLINER PHILHARMONIKER · RAFAEL KUBELIK

MUTATION TESTING

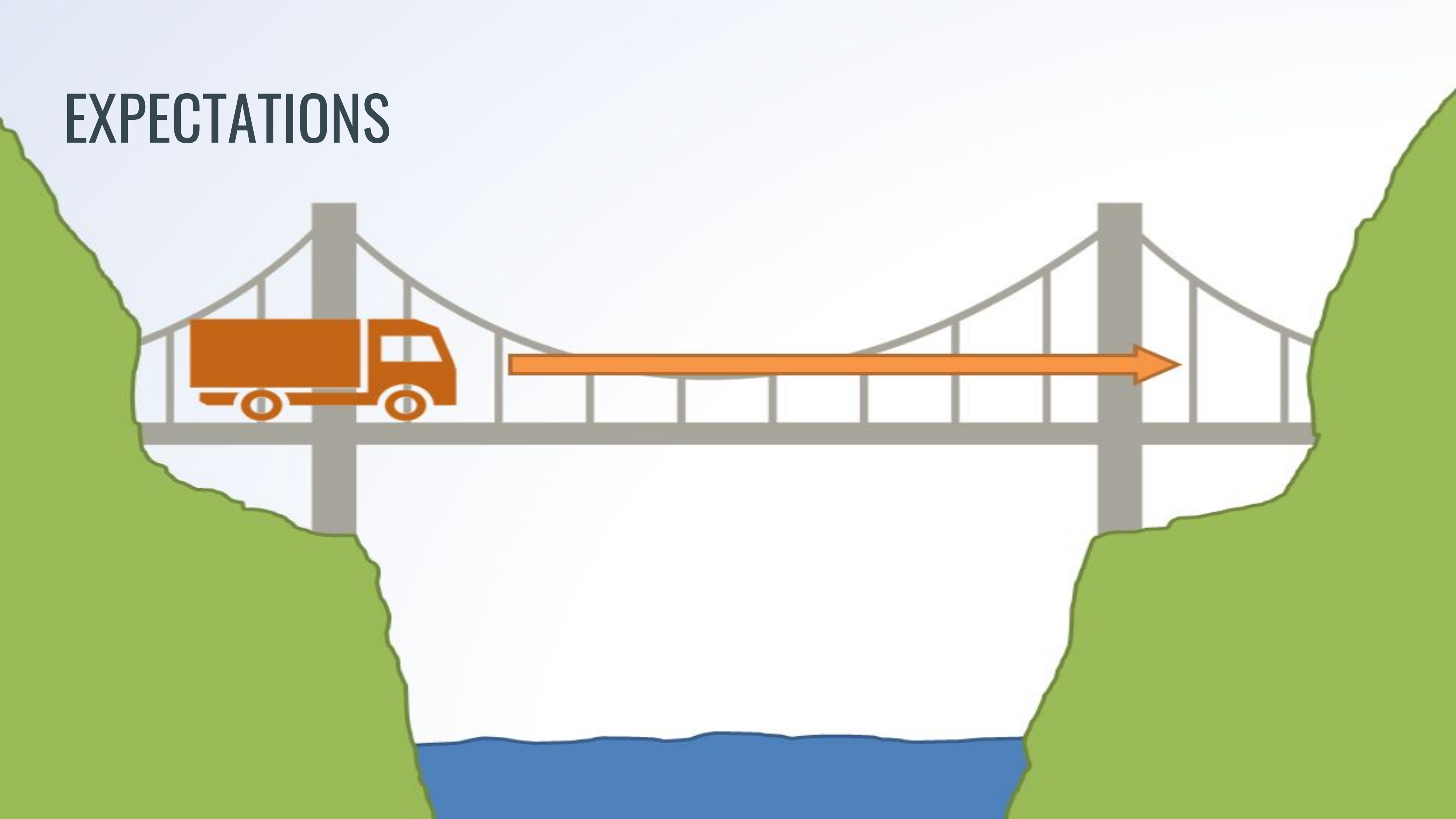
concept

PROBLEM

SOLUTION



EXPECTATIONS

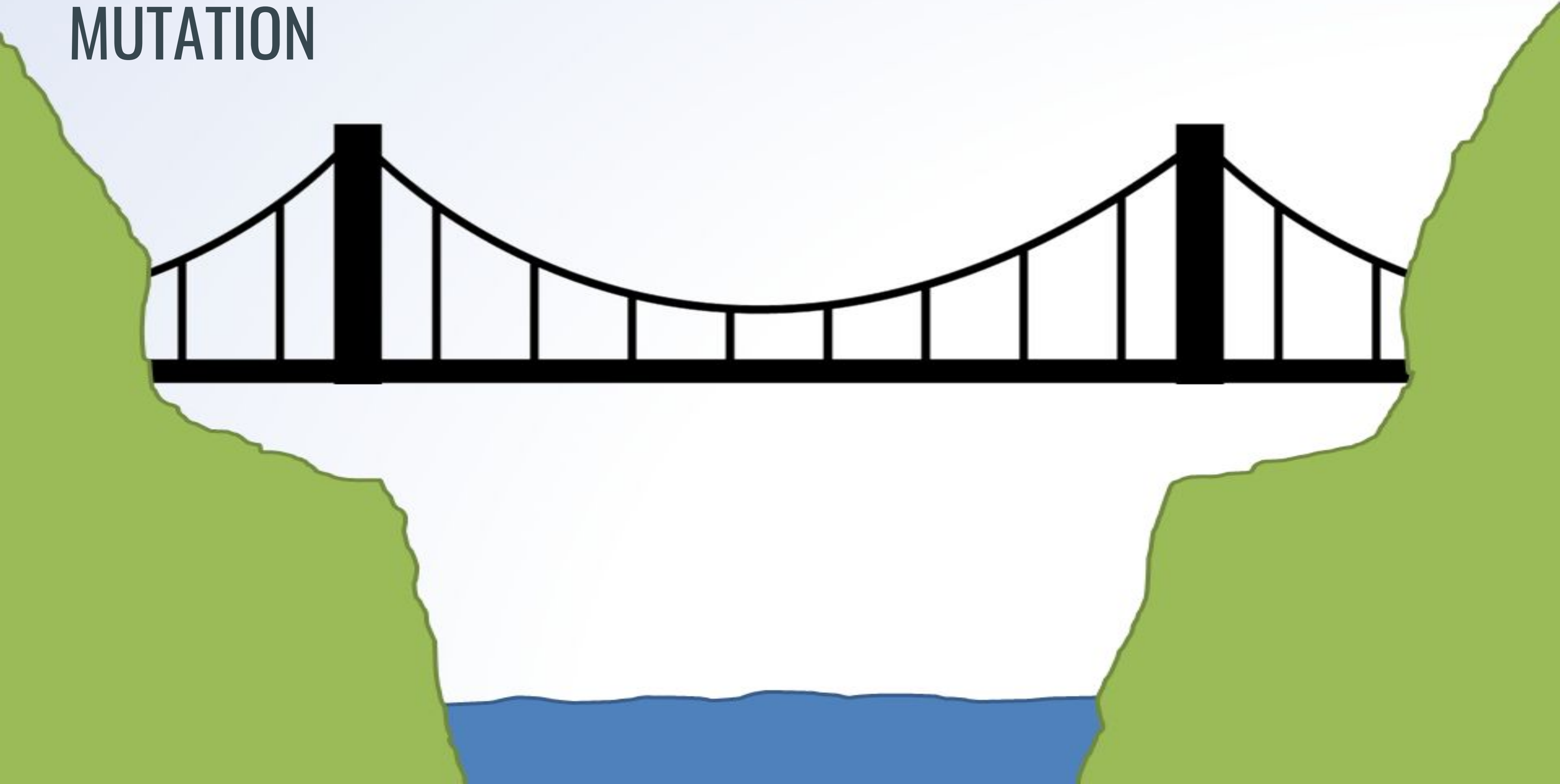


TESTS - TDD?

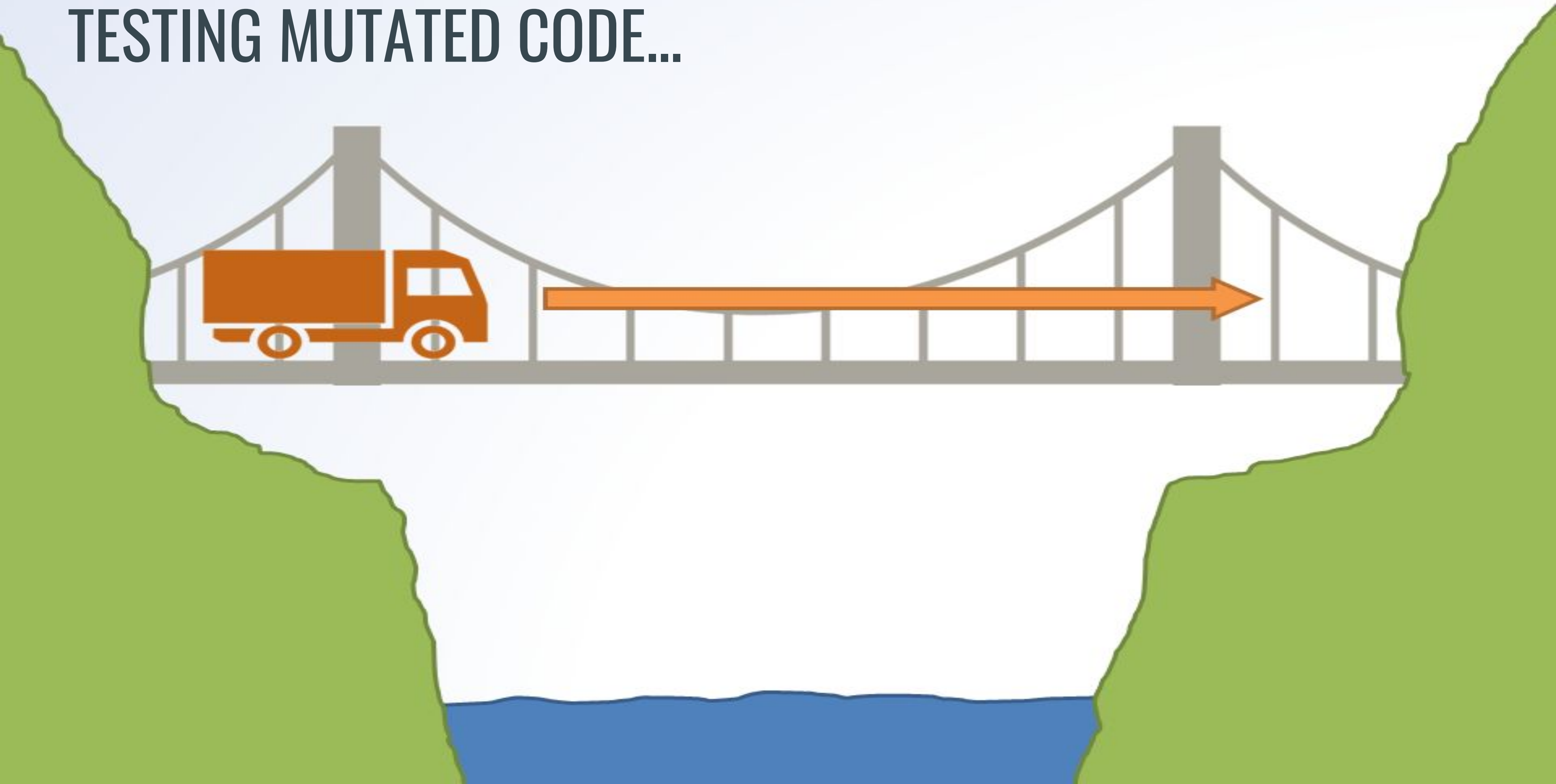
TESTS - NO TDD?

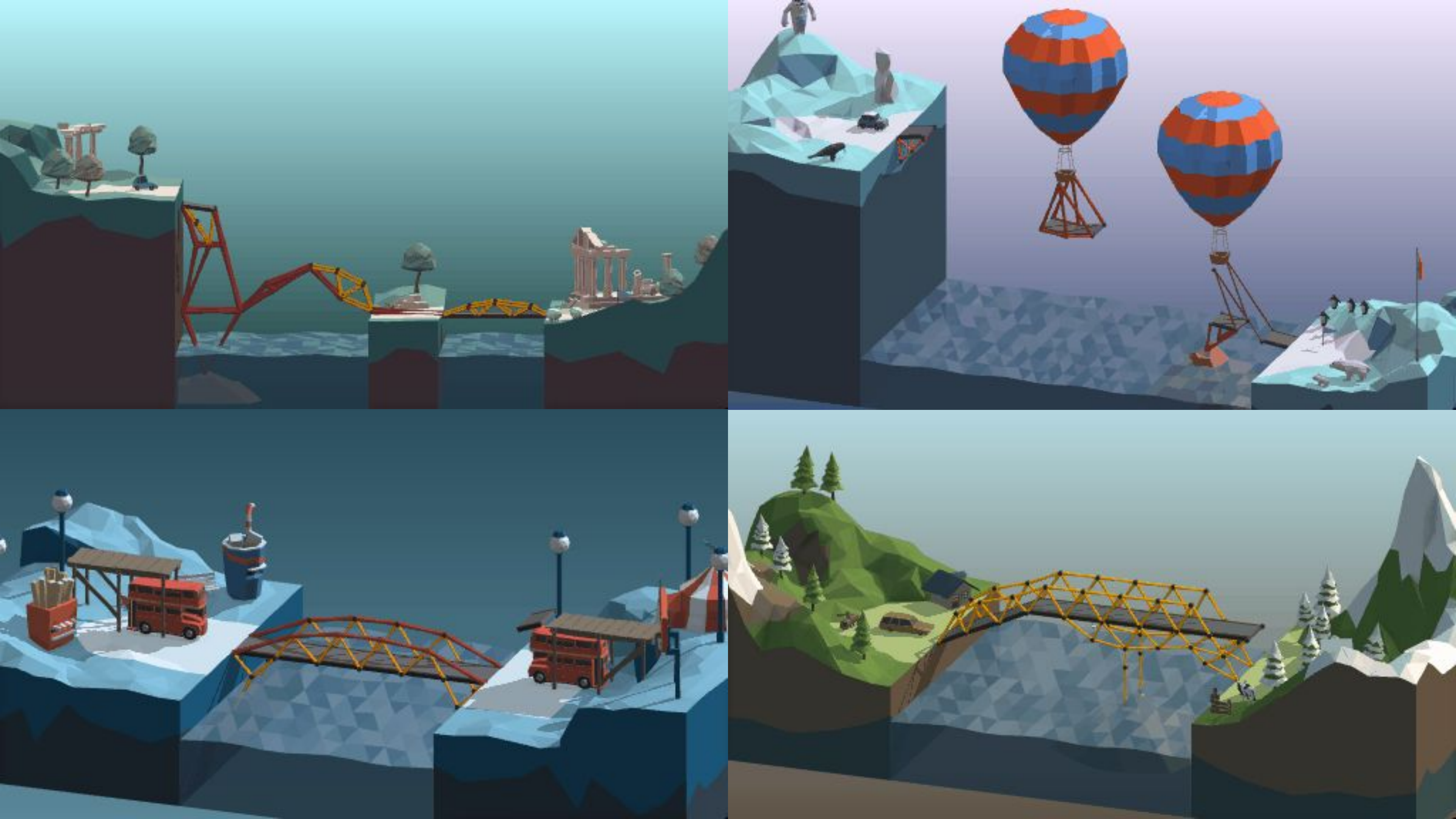
TEST

MUTATION

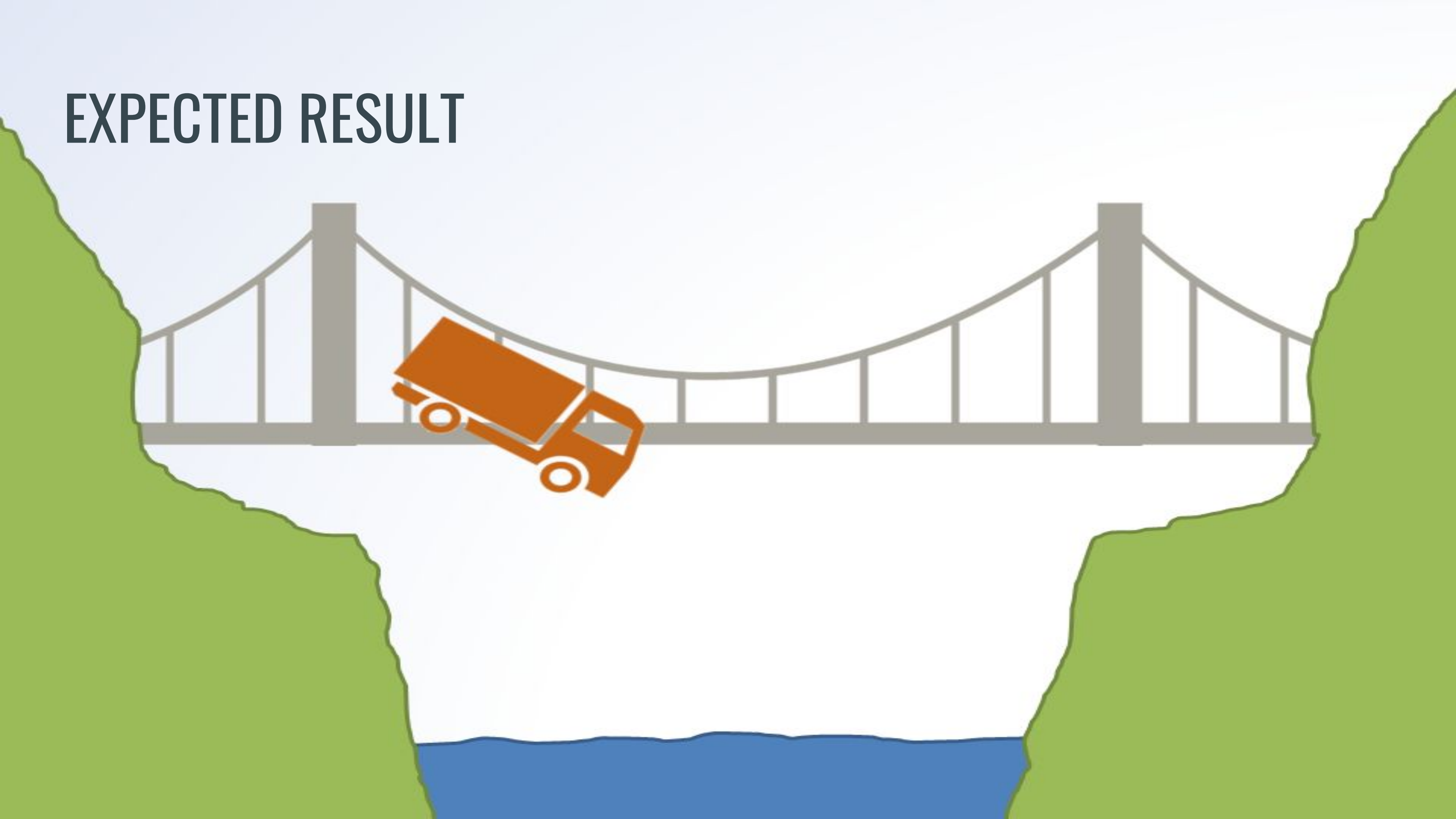


TESTING MUTATED CODE…

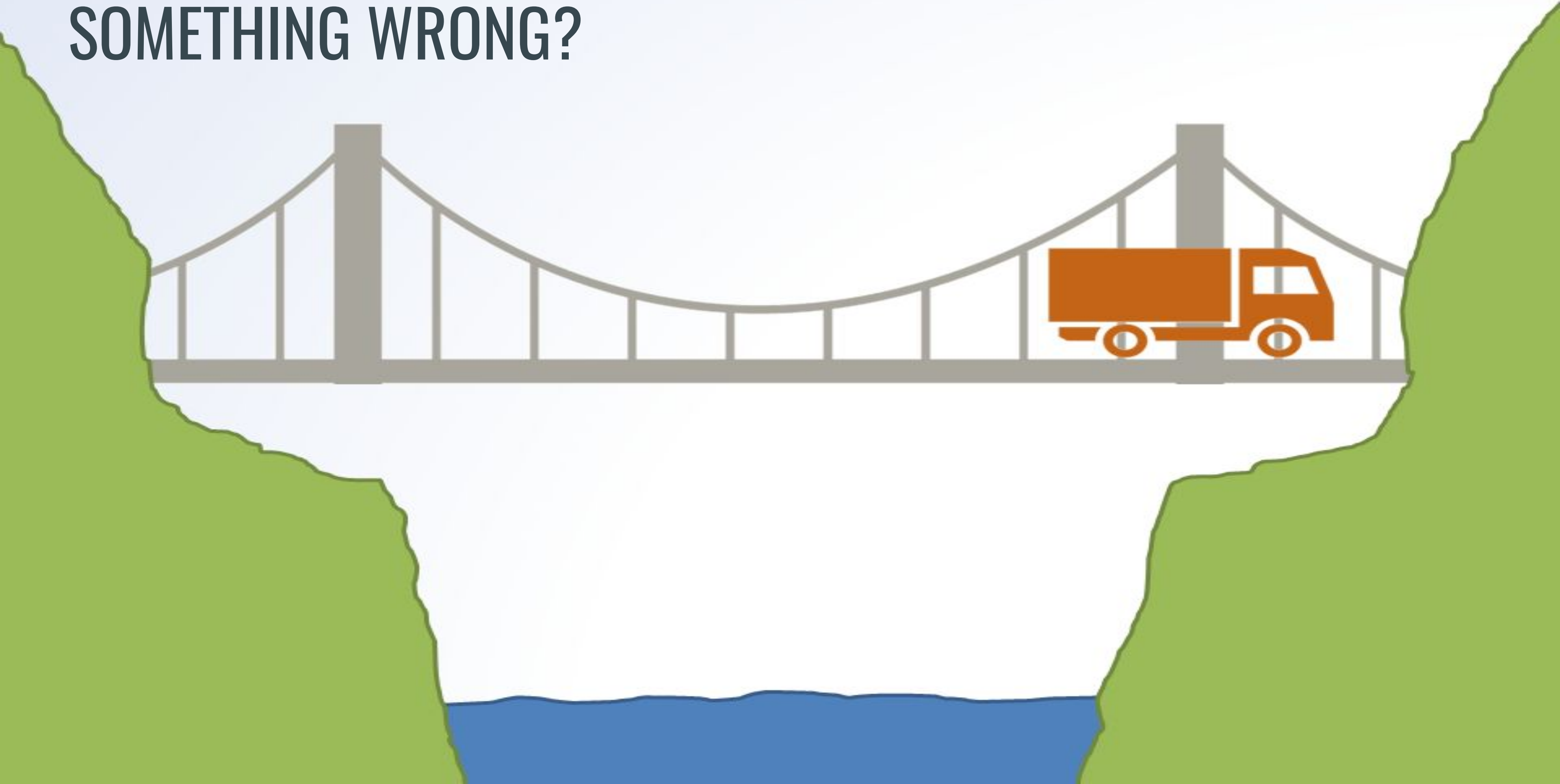




EXPECTED RESULT



SOMETHING WRONG?



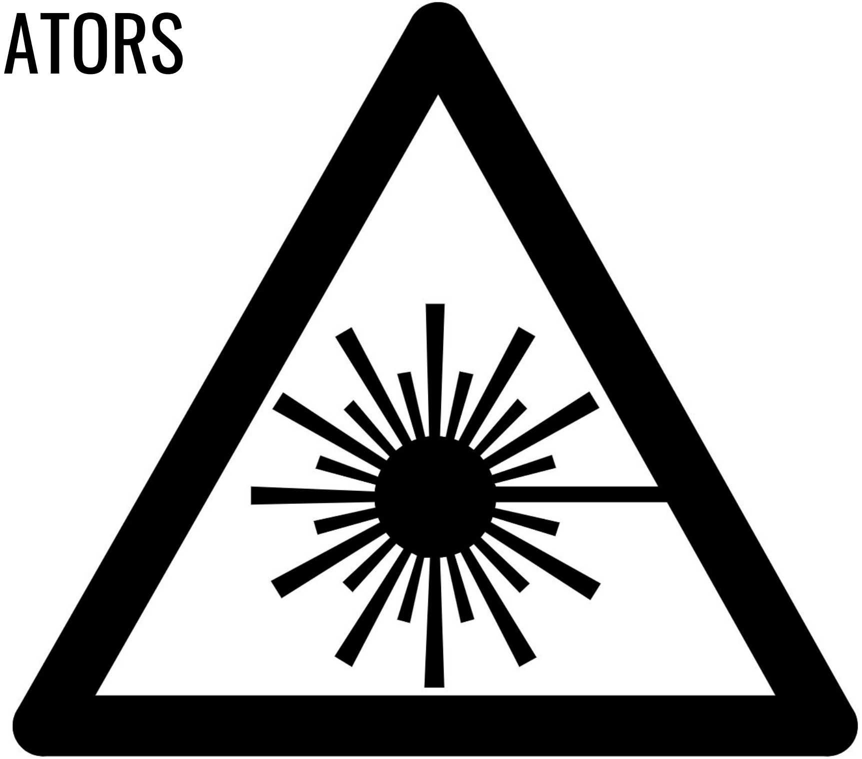
SO. WHAT EXACTLY HAPPENED...

MUTATION TESTING

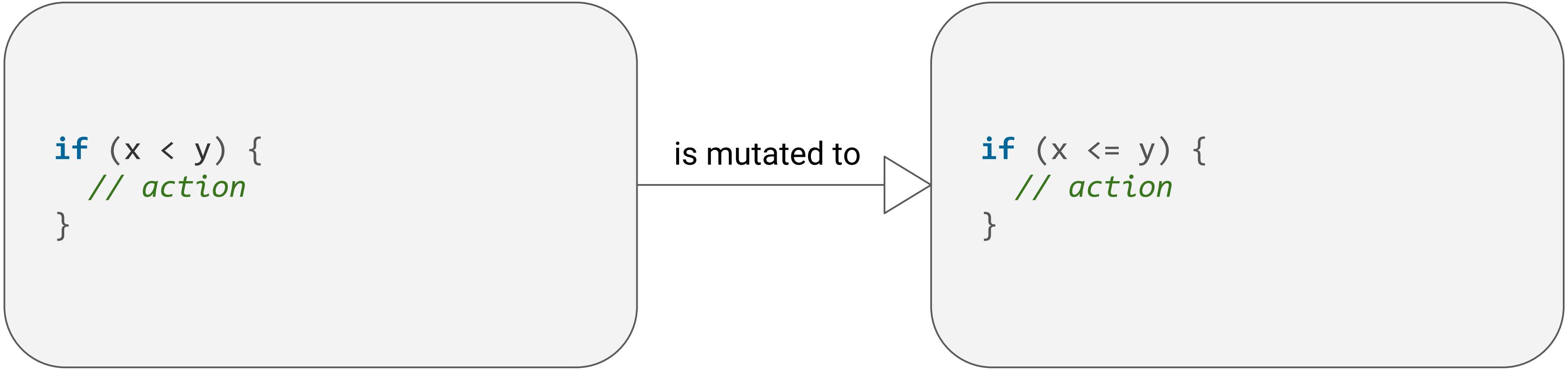
principle

1. Run your test suites
2. Mutate a piece of production code
3. Run your tests again
4. Find tests that did not fail
5. Repeat steps 2 - 4 with different mutations

MUTATION OPERATORS



BOUNDARY

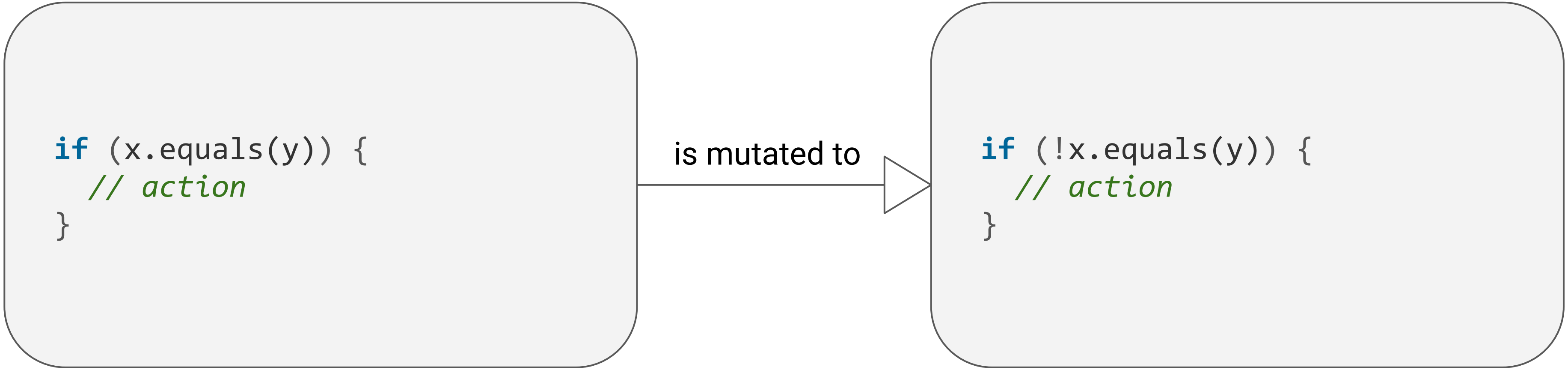


```
if (x < y) {
  // action
}
```

is mutated to

```
if (x <= y) {
  // action
}
```

NEGATION

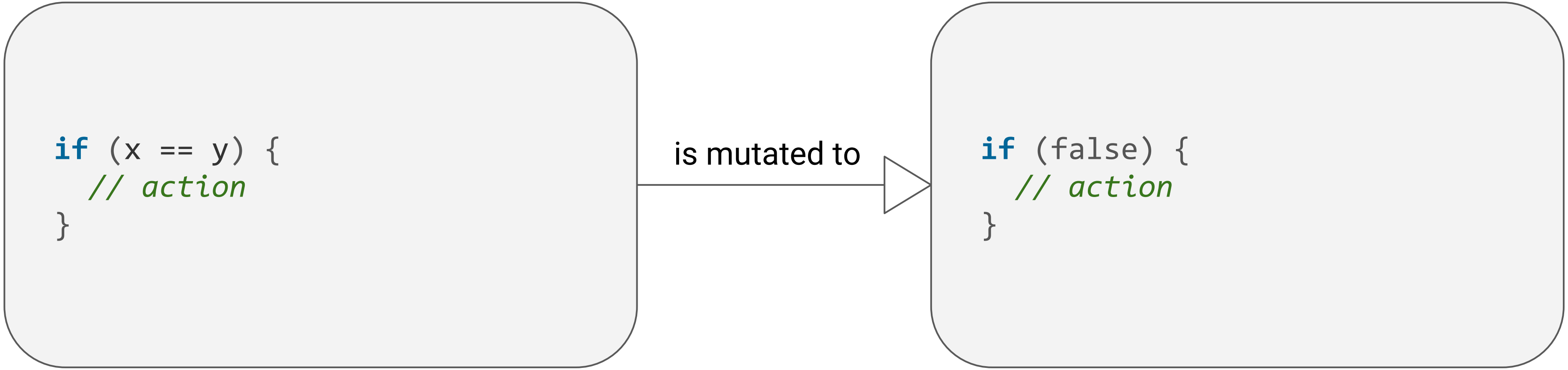


```
if (x.equals(y)) {
  // action
}
```

is mutated to

```
if (!x.equals(y)) {
  // action
}
```

CONDITIONALS

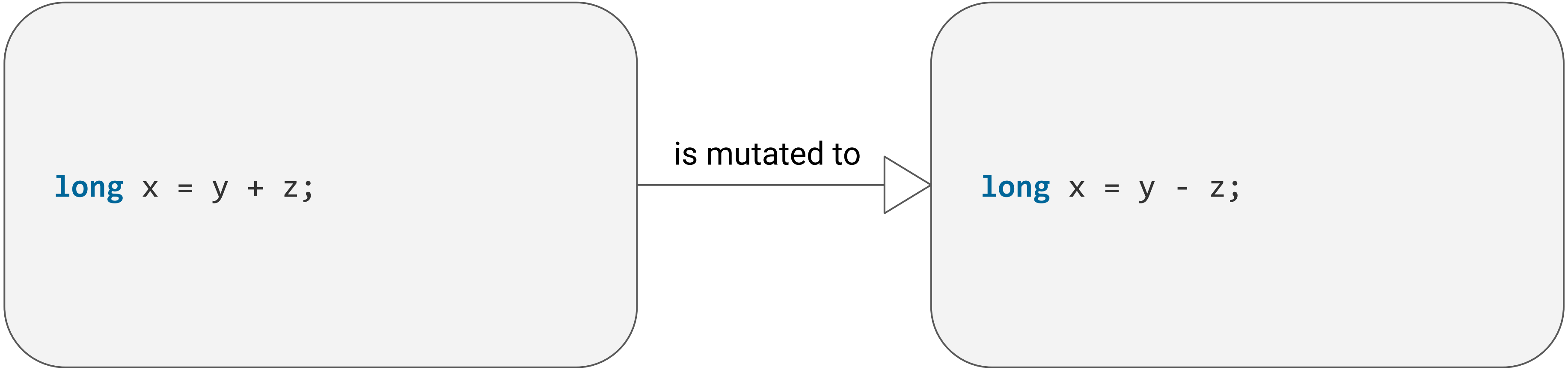


```
if (x == y) {
  // action
}
```

is mutated to

```
if (false) {
  // action
}
```

MATH

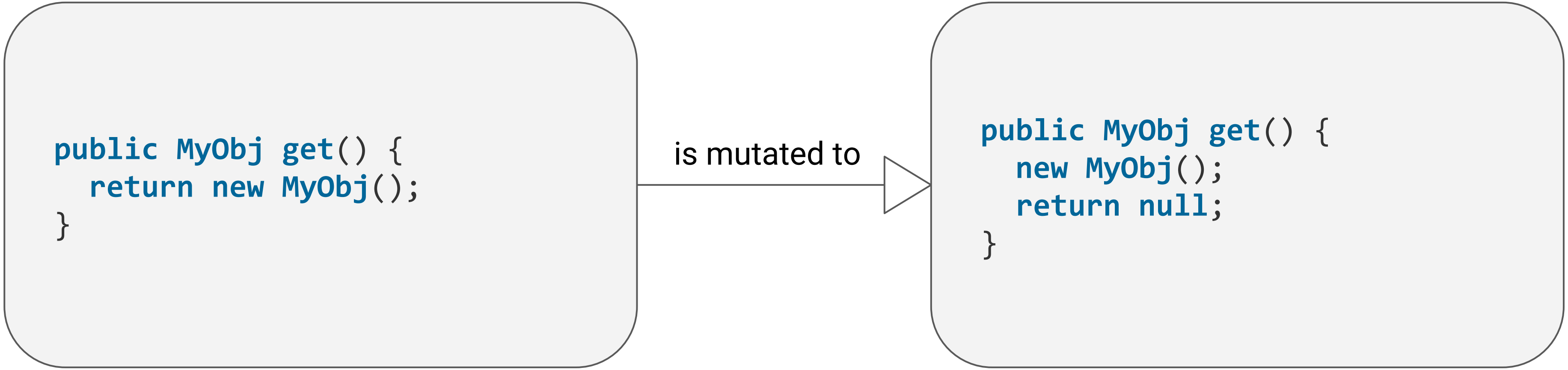


```
long x = y + z;
```

is mutated to

```
long x = y - z;
```

RETURN

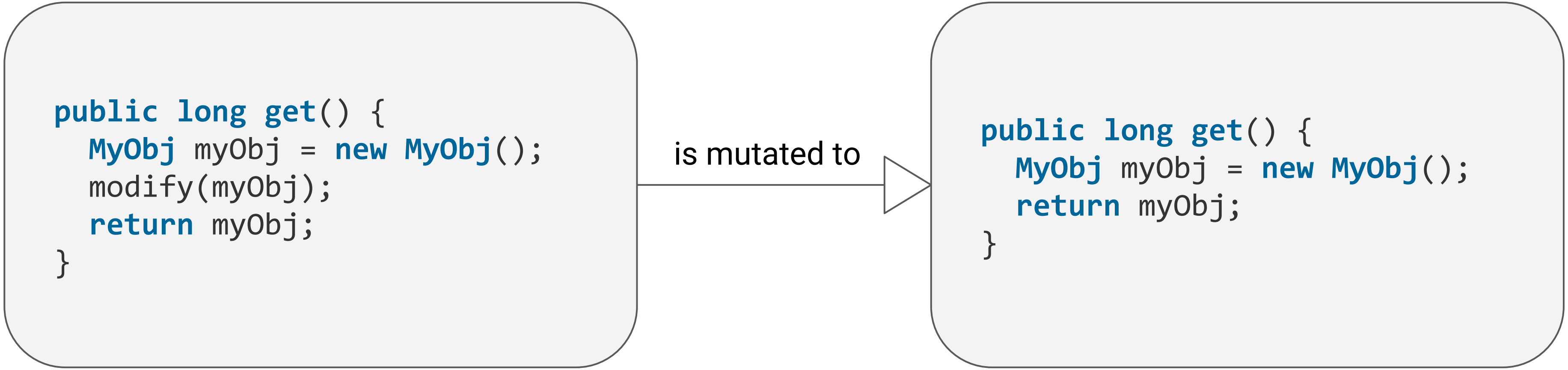


```
public MyObj get() {
  return new MyObj();
}
```

is mutated to

```
public MyObj get() {
  new MyObj();
  return null;
}
```

VOID

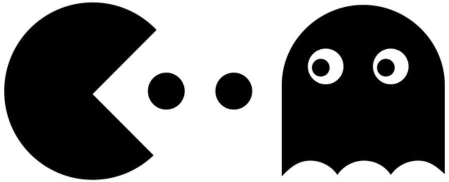


```java
public long get() {
  MyObj myObj = new MyObj();
  modify(myObj);
  return myObj;
}
```

is mutated to

```java
public long get() {
  MyObj myObj = new MyObj();
  return myObj;
}
```

SURVIVORS



SETUP

Pitest (Java)

```
<plugin>
    <groupId>org.pitest</groupId>
    <artifactId>pitest-maven</artifactId>
    <version>LATEST</version>
 </plugin>
```

```
mvn org.pitest:pitest-maven:mutationCoverage
```

Stryker (JavaScript, Scala, others)

MUTATION TESTING

summary

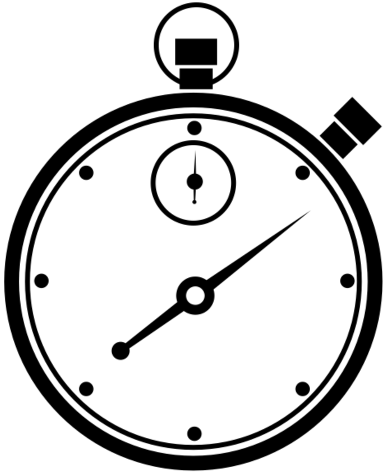
EXECUTION

100 classes

10 tests / class

1 ms / test

10 mutations / class (100 classes)



EXECUTION

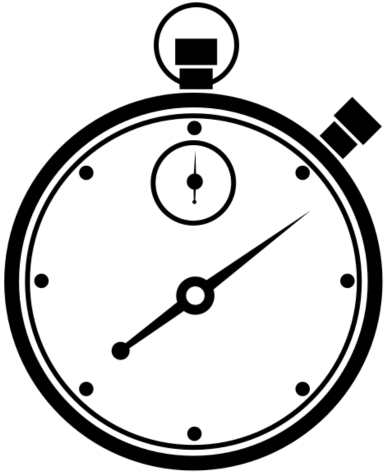
100 classes

10 tests / class

1 ms / test

10 mutations / class (100 classes)

~ 17 minutes



EXECUTION

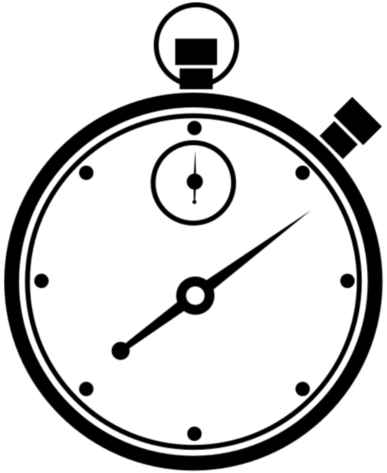
100**0** classes

10 tests / class

1 ms / test

10 mutations / class (1000 classes)

~ 28 hours



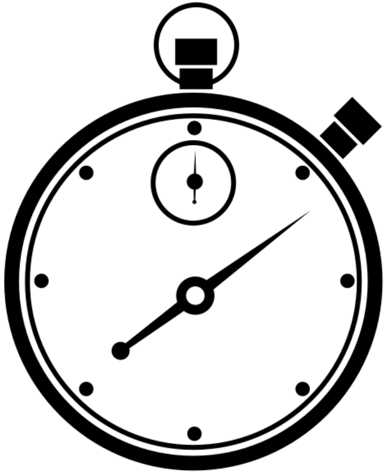
SMART EXECUTION

1000 classes

10 tests / class

1 ms / test

10 mutations / class (1000 classes)



~ 2 minutes

real life example: 481 test cases fully mutated on 16GB i7 -> 03:24 min

SURVIVOR STORY

CONTRACT TESTING

first article in 2006 (Ian Robinson)

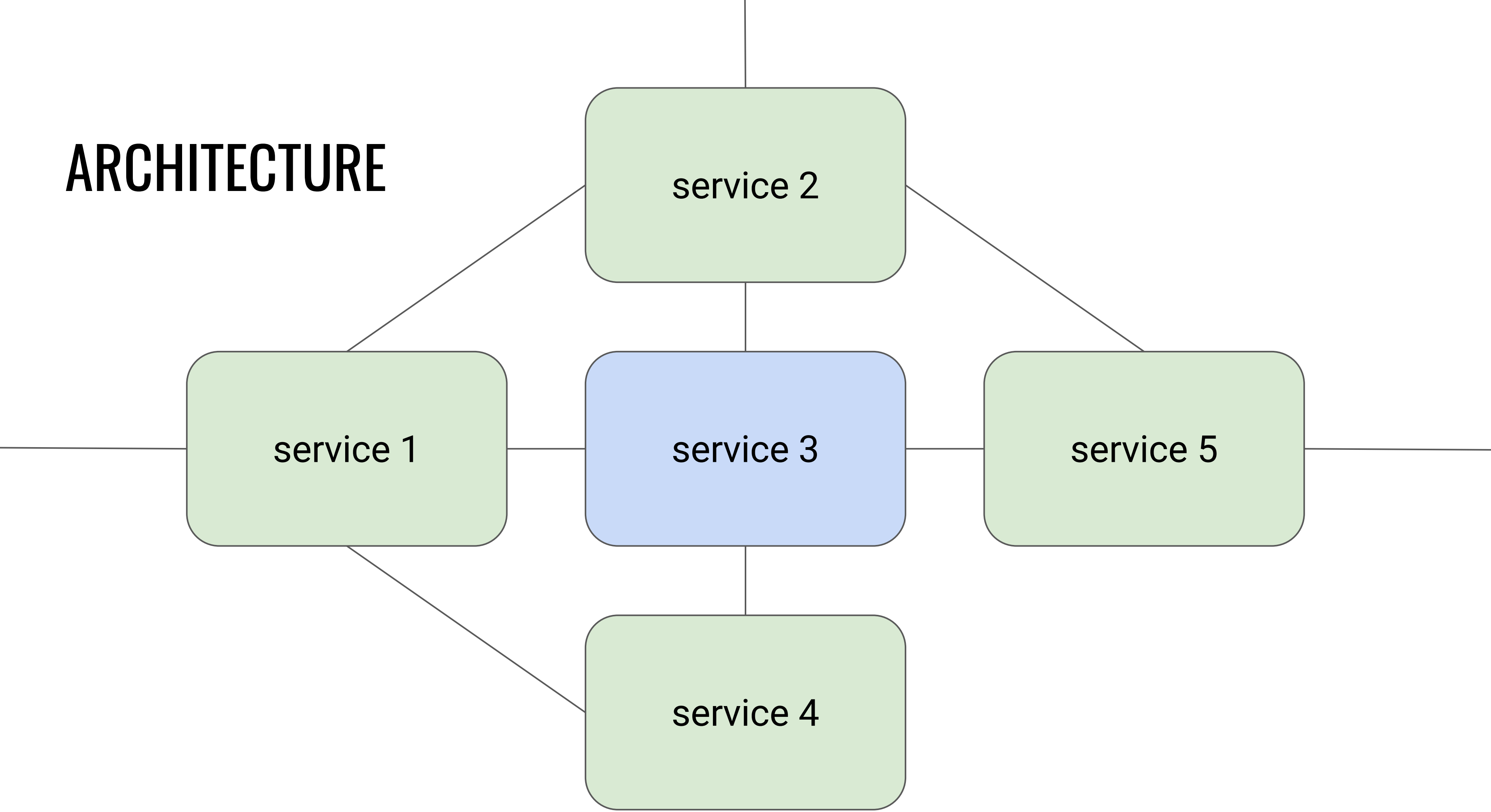
I. Robinson “Consumer-Driven Contracts: A Service Evolution Pattern”

CONTRACT TESTING

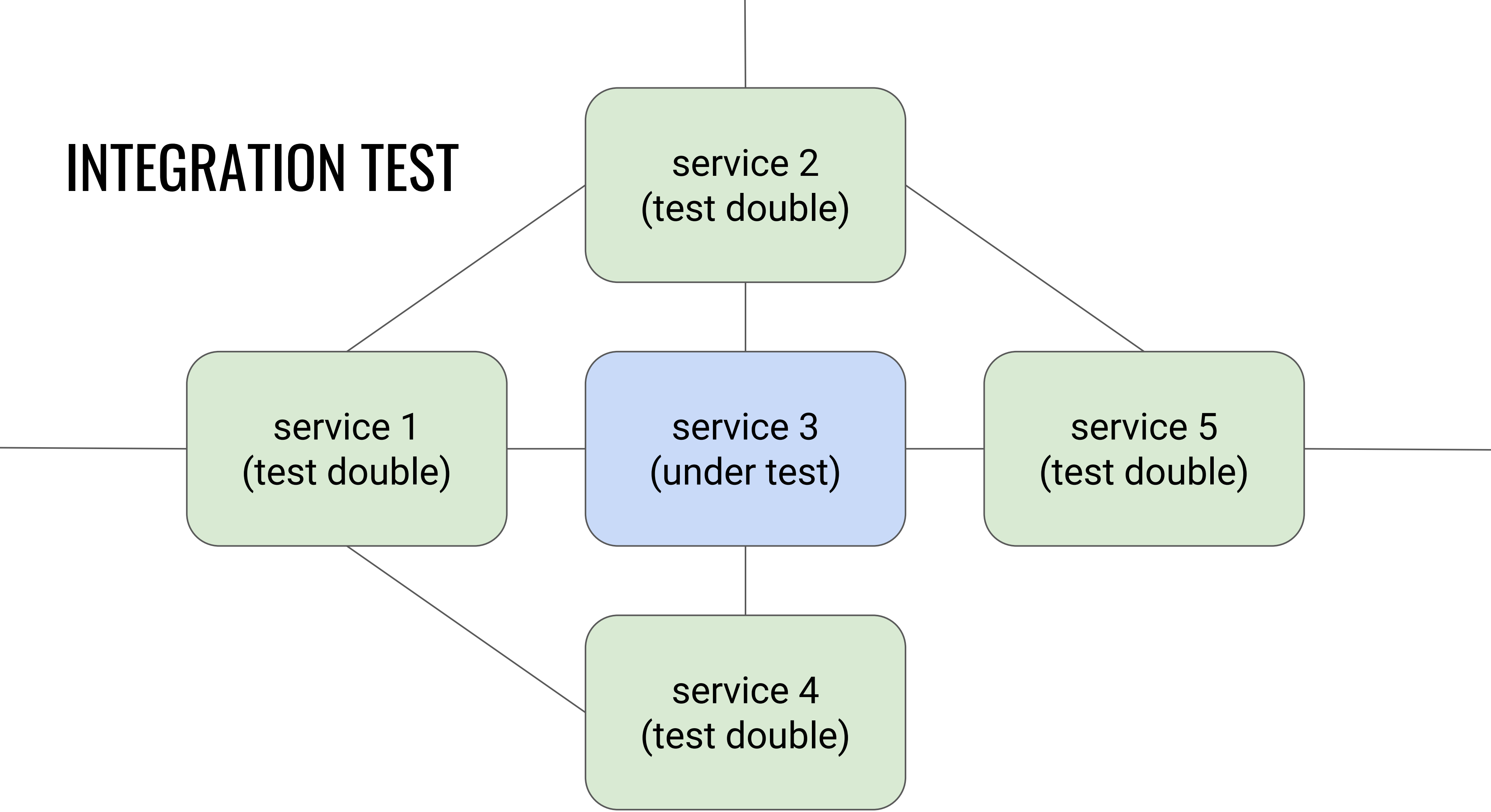
concept

Why?

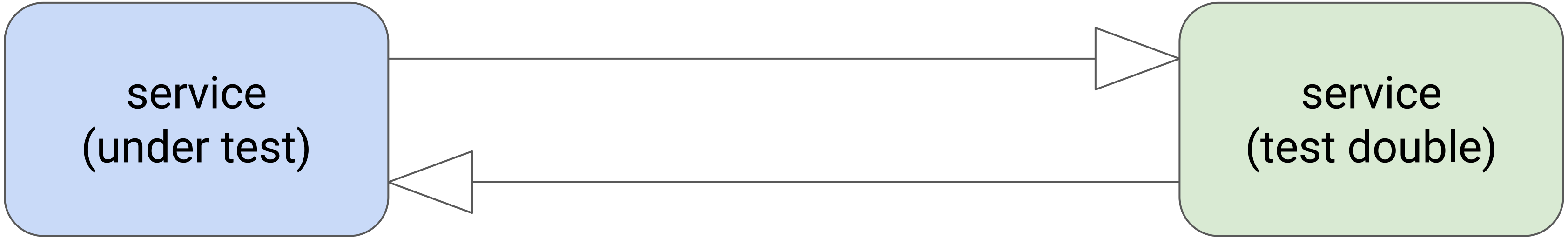
ARCHITECTURE



INTEGRATION TEST



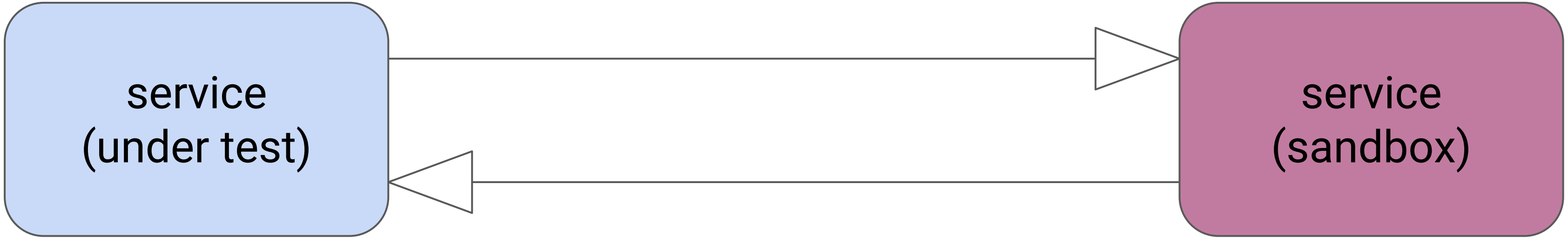
SINGLE INTEGRATION



service
(under test)

service
(test double)

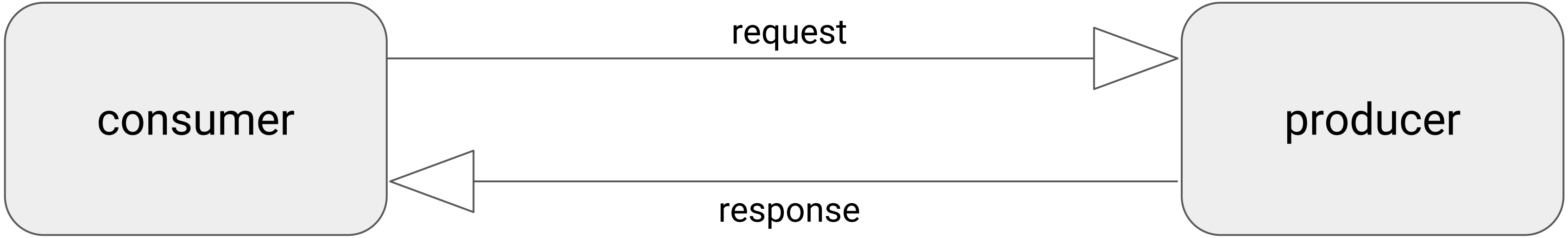
SINGLE INTEGRATION



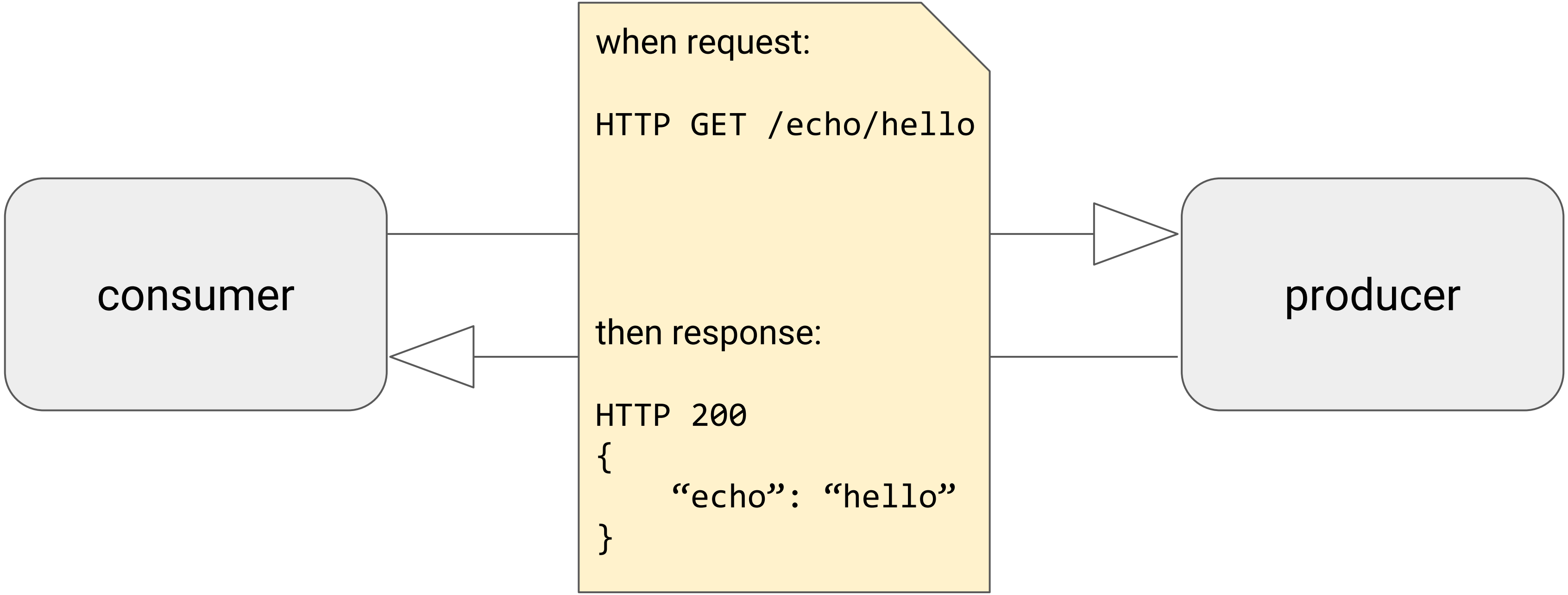
service
(under test)

service
(sandbox)

INTERFACE



CONTRACT



when request:

```
HTTP GET /echo/hello
```

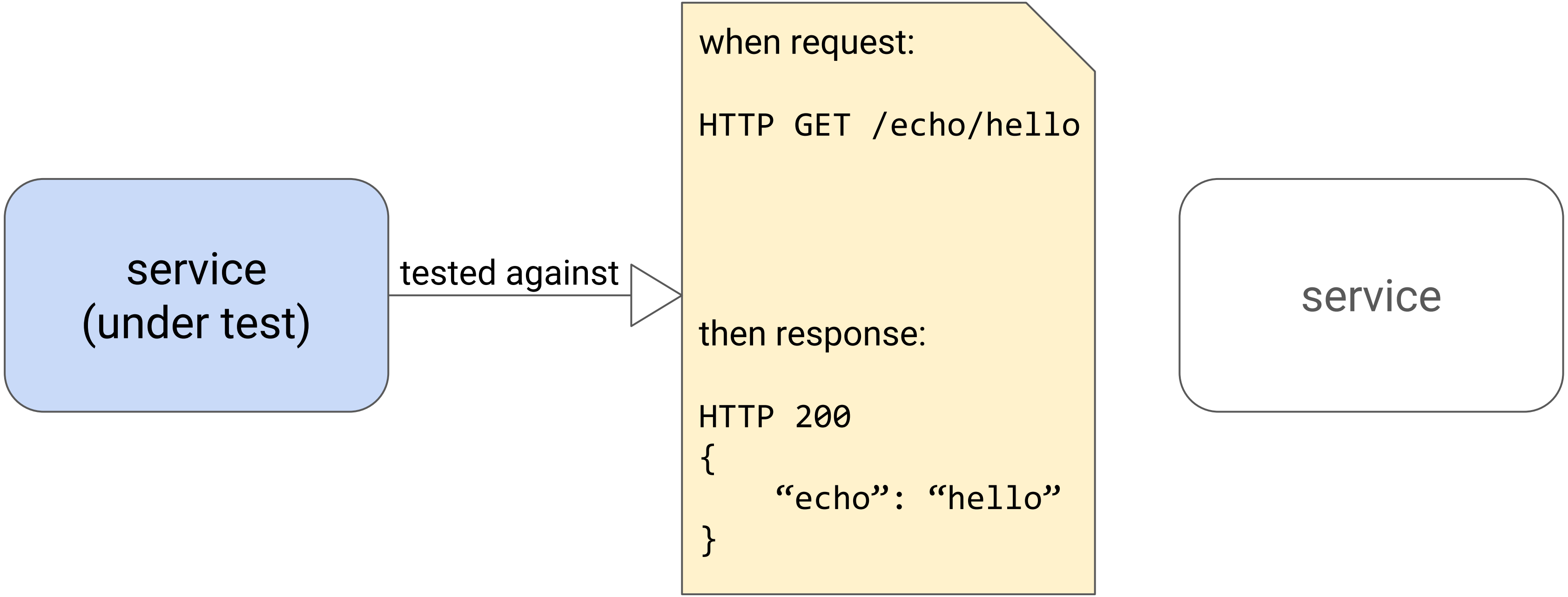
then response:

```
HTTP 200
{
    “echo”: “hello”
}
```

CONTRACT TESTING

principle

CONTRACT TEST



service
(under test)

tested against

when request:

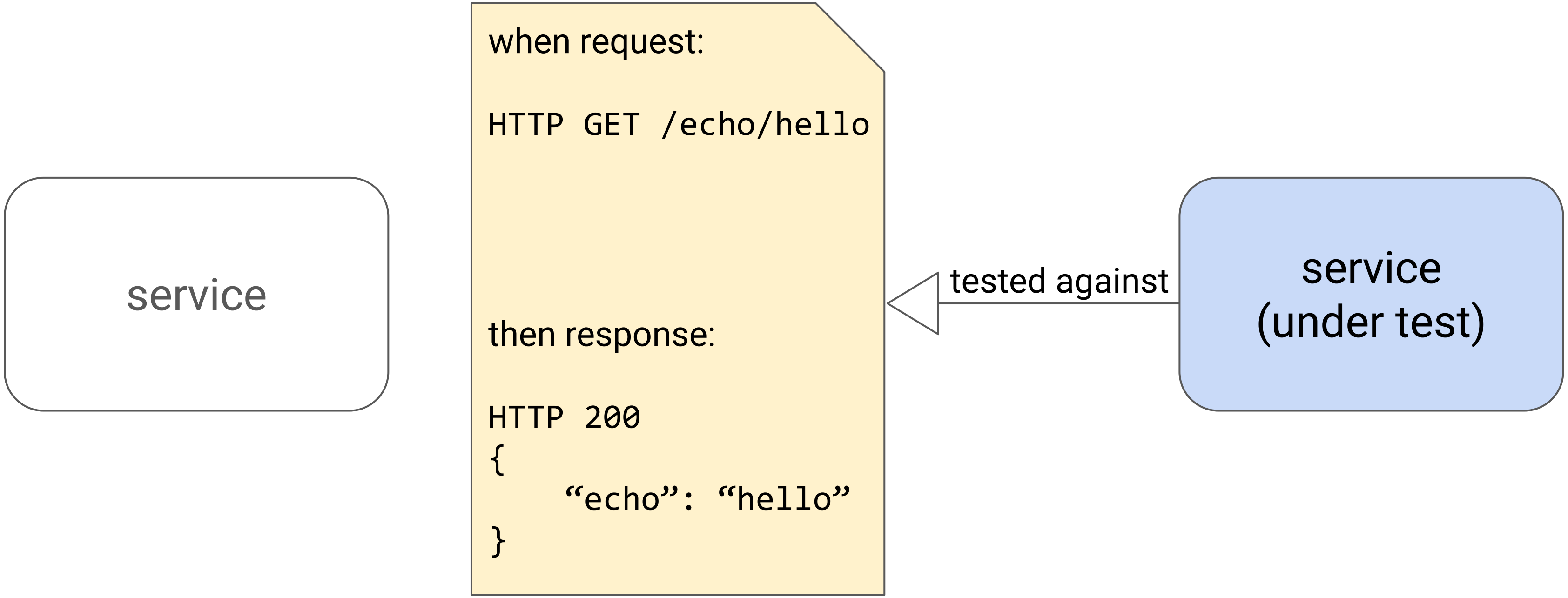
```
HTTP GET /echo/hello
```

then response:

```
HTTP 200
{
    “echo”: “hello”
}
```

service

CONTRACT TEST



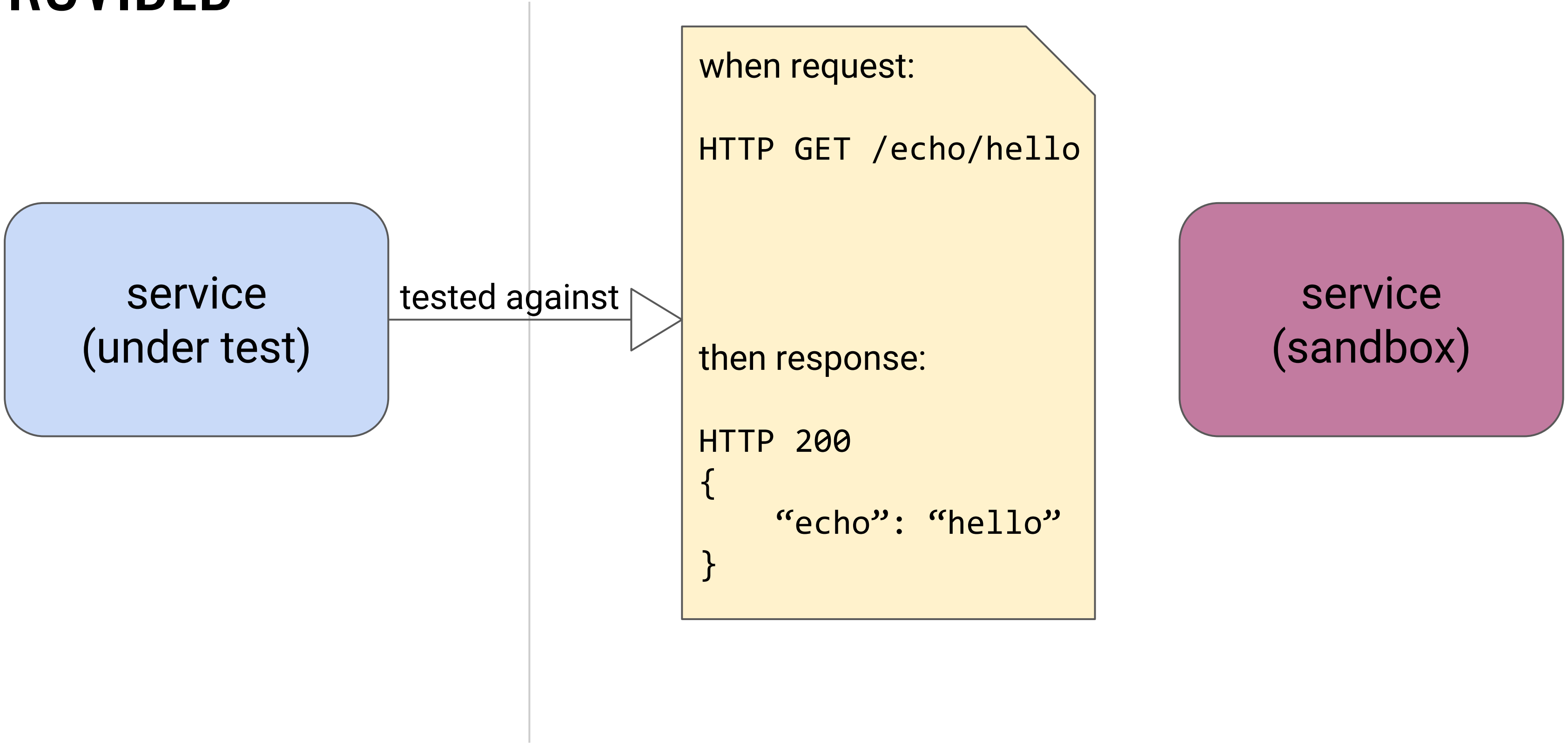
when request:

```
HTTP GET /echo/hello
```

then response:

```
HTTP 200
{
    “echo”: “hello”
}
```

PROVIDED



service
(under test)

tested against

when request:

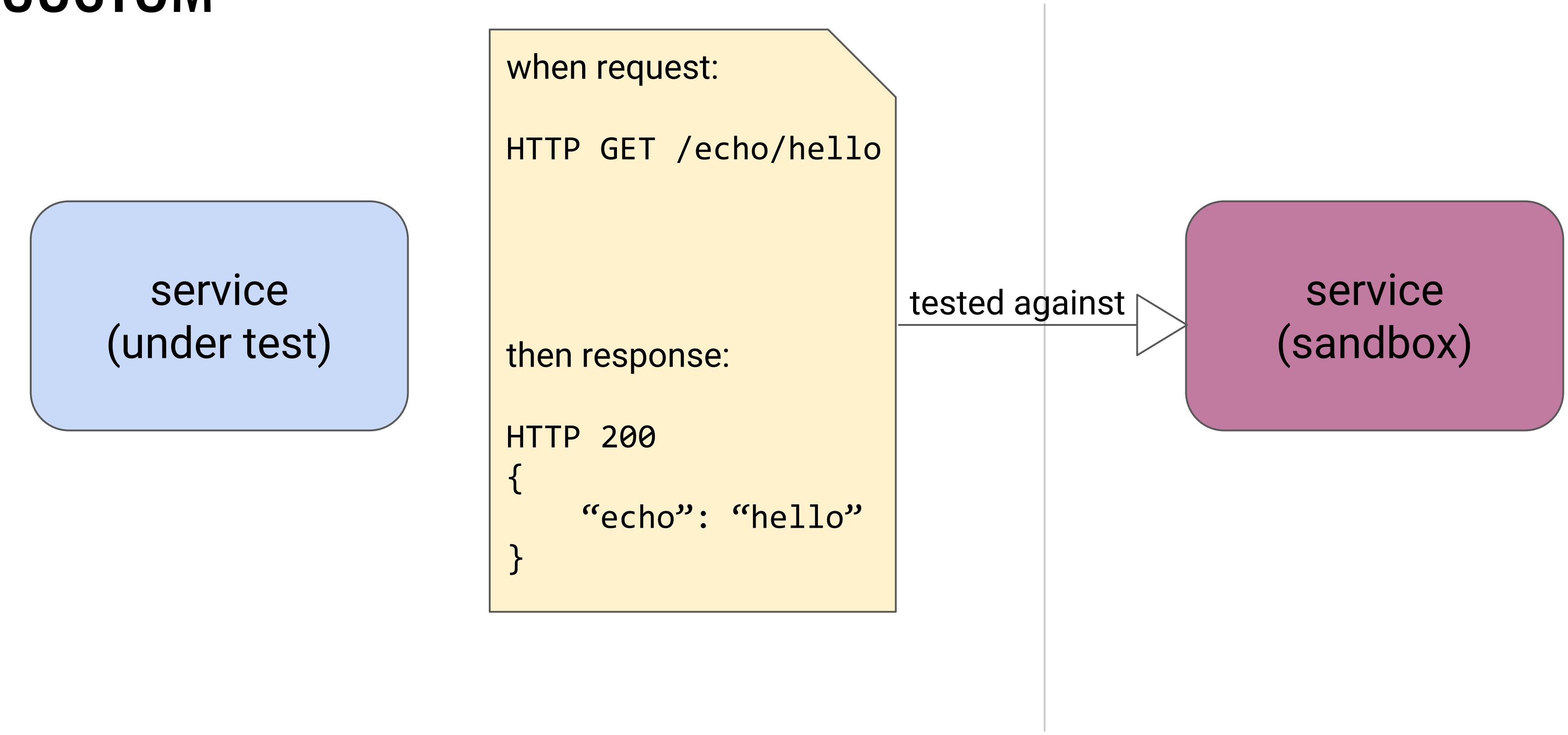
```
HTTP GET /echo/hello
```

then response:

```
HTTP 200
{
    “echo”: “hello”
}
```

service
(sandbox)

CUSTOM



when request:

```
HTTP GET /echo/hello
```

then response:

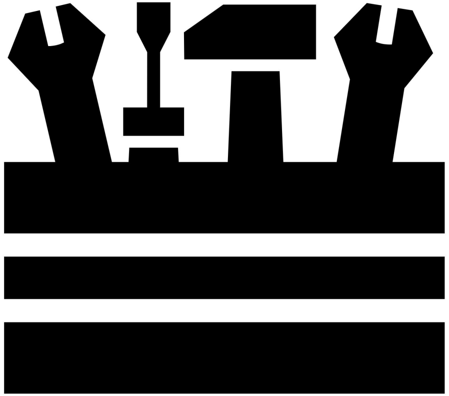
```
HTTP 200
{
    “echo”: “hello”
}
```

SETUP

Spring Cloud Contract (Java)

https://github.com/mbryla/contract-testing

```
<plugin>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-contract-maven-plugin</artifactId>
    <version>2.0.0.M7</version>
    <extensions>true</extensions>
    <configuration>
        <!-- Provide the base class for your auto-generated tests →
        <baseClassForTests>com.mbryla.contract.producer.BaseTest</baseClassForTests>
    </configuration>
</plugin>
```



Pact (Ruby, JavaScript, Go, others)

SETUP

```
package contracts
import org.springframework.cloud.contract.spec.Contract
Contract.make {
    request {
        method 'GET'
        url '/echo/hello'
        headers {
            header('Content-Type', 'application/json;charset=UTF-8')
        }
    }

    response {
        status 200
        body("{\"echo\": \"hello\"}")
        headers {
            header('Content-Type': 'application/json;charset=UTF-8')
        }
    }
}
```

CONTRACT TESTING

summary

FINAL REMARKS

ALL IN ONE - take photos ;)

This presentation - “Mutation and Contract testing” at CodeteCON #4, Kraków 2018

`https://medium.com/@mateusz.bryla`

`https://codete.com/blog/mutation-and-contract-testing/`

`https://github.com/mbryla`

Mutation tests - help verify the quality of your tests

Pitest (Java), Stryker (JavaScript, Scala, others)

```
mvn org.pitest:pitest-maven:mutationCoverage
```

Contract tests - protect you from unstable mocks / sandboxes

Spring Cloud Contract (Java), Pact (Ruby, JavaScript, Go, others)

`https://github.com/mbryla/contract-testing`

ATTRIBUTIONS

Hands Up - Avel Chuklanov - https://unsplash.com/photos/9cx4-QowgLc (CC0 1.0)
Bridge - Franc - https://thenounproject.com/search/?q=bridge&i=122542 (CC BY 3.0 US)
Cargo Truck - No way - https://thenounproject.com/search/?q=truck&i=154864 (CC0 1.0)
Berlin Wall - Unknown - https://commons.wikimedia.org/wiki/File:West_and_East_Germans_at_the_Brandenburg_Gate_in_1989.jpg (CC 1.0)
Linus Torvalds - Unknown - https://commons.wikimedia.org/wiki/Linux#/media/File:Linus_Torvalds.jpeg (CC 3.0)
Poly Bridge - Imgur (respective owners)
https://imgur.com/r/PolyBridge/hgyZIeO
https://imgur.com/r/PolyBridge/yVHKUpz
https://imgur.com/WYpPKbs
https://imgur.com/r/PolyBridge/2hTYZno
Blown bridge - Giphy - https://giphy.com/gifs/explosion-bridge-miFVHkNdJoowM - respective owner
Hilarious little girl - Giphy - https://giphy.com/gifs/AuIvUrZpzBl04 - respective owner
Laser - Yohan Berger - https://thenounproject.com/search/?q=laser&i=36677 (CC BY 3.0 US)
Resurrection - Isabel Martinez Isabel - https://thenounproject.com/search/?q=survived&i=202842 (CC BY 3.0 US)
Pac Man - iconoci - https://thenounproject.com/search/?q=pac+man&i=9215 (CC BY 3.0 US)
Stopwatch - Nick Holroyd - https://thenounproject.com/search/?q=stopwatch&i=12666 (CC BY 3.0 US)
Toolbox - Georgiana Ionescu - https://thenounproject.com/search/?q=toolbox&i=1348711 (CC BY 3.0 US)
Survivors - Gerald Wildmoser - https://thenounproject.com/search/?q=drown&i=208787 (CC BY 3.0 US)
Sandbox - Giphy - https://media.giphy.com/media/AAHUIzS2Oo8UM/giphy.gif - respective owner
Thank you - Gan Khoon Lay - https://thenounproject.com/search/?q=thank%20you&i=854669 (CC BY 3.0 US)

Q/A

LINGmates

Mateusz Bryła
mateusz.bryla@lingmates.com