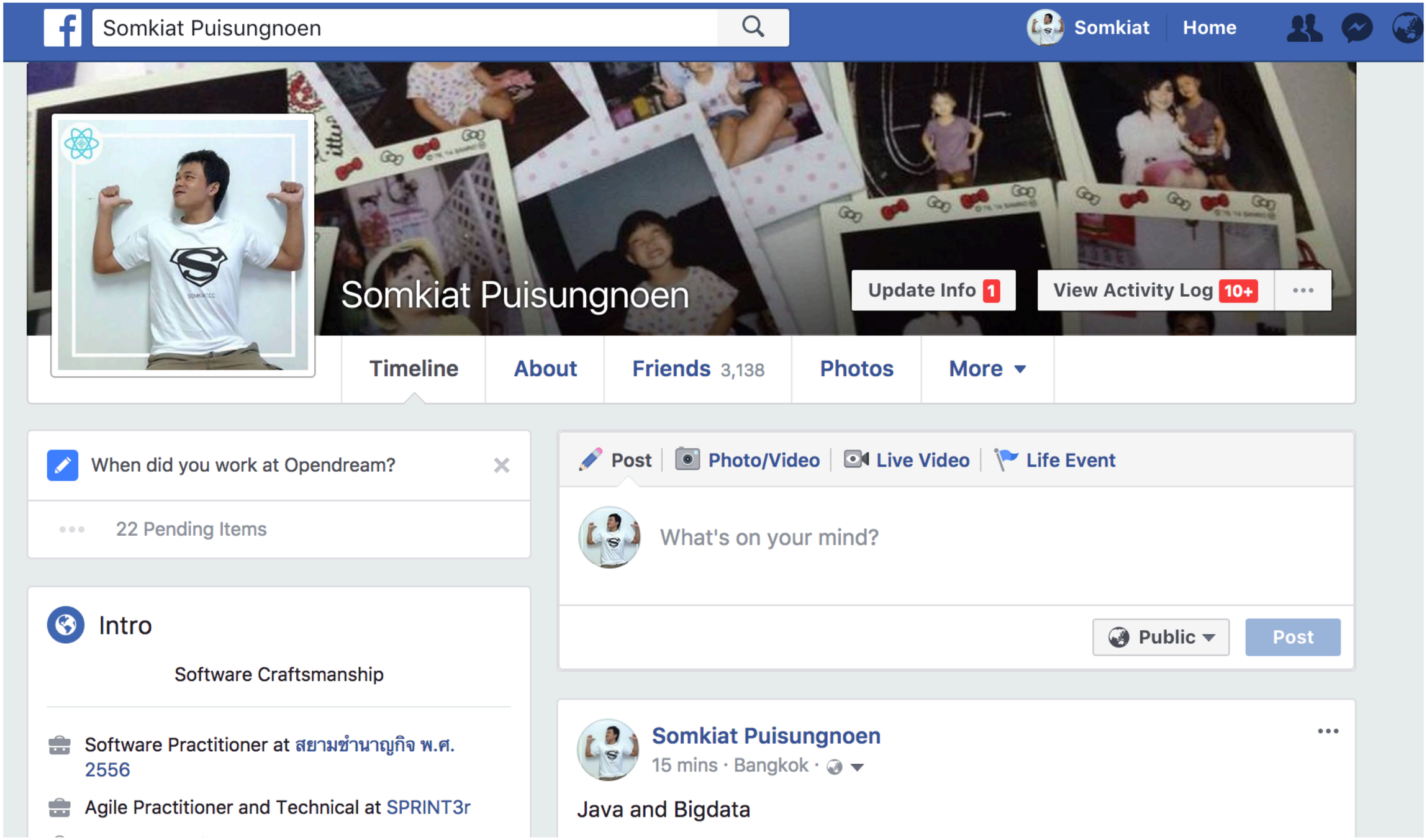
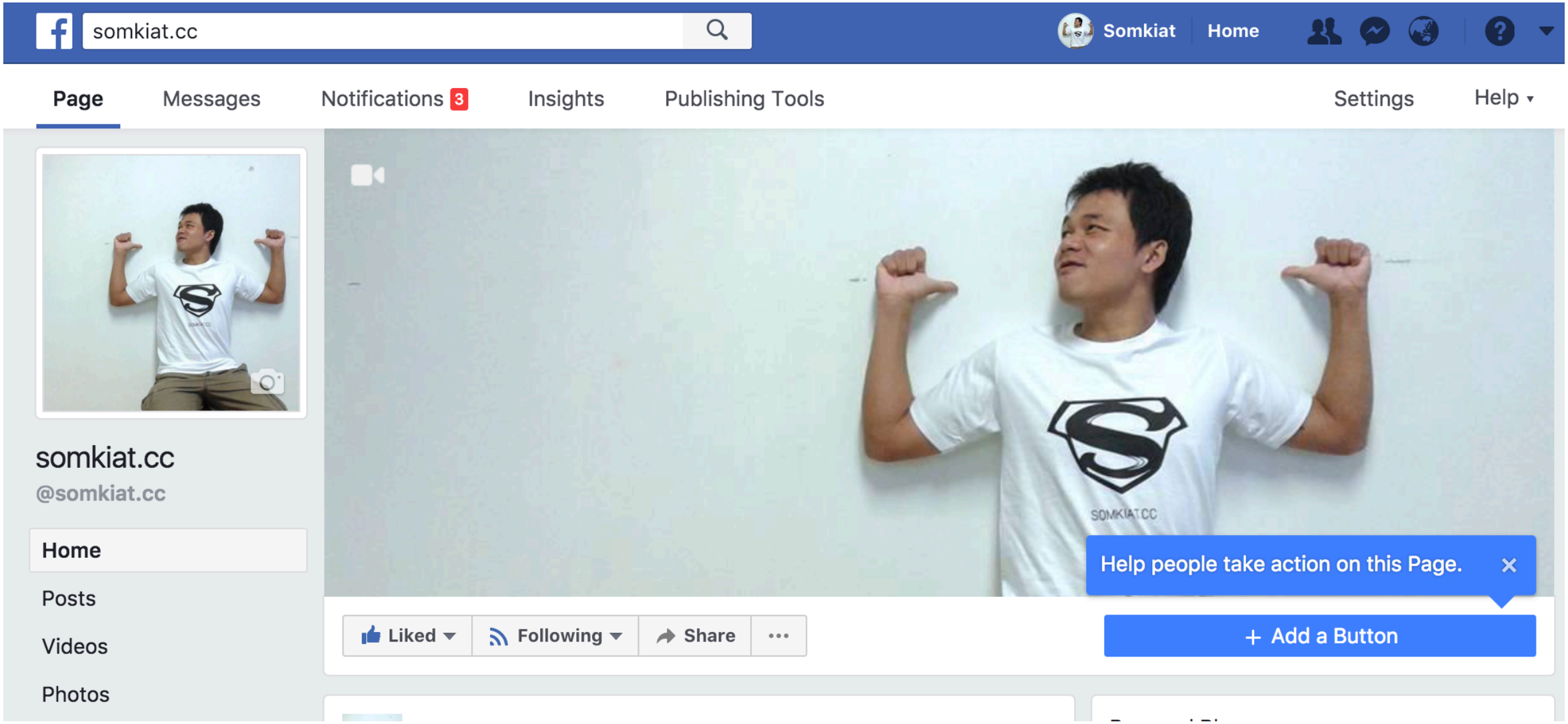
สยามชำนาญกิจ
SIAM CHAMNAN KIT

Test-Driven Development with JAVA



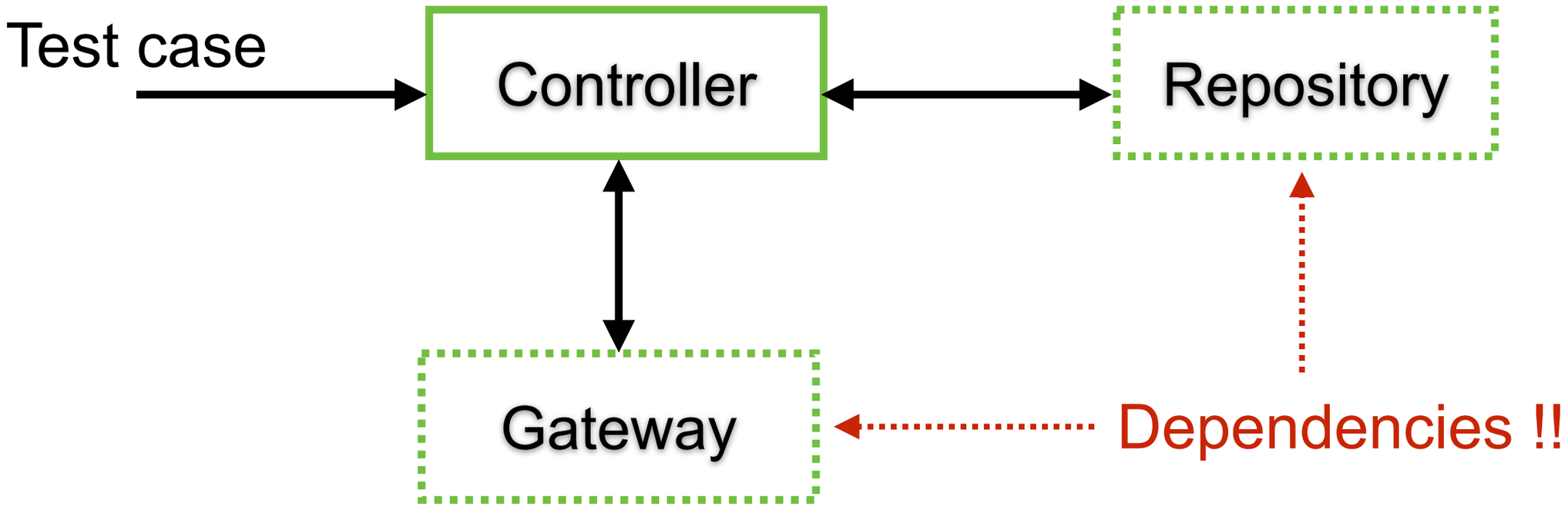


Manage your dependencies

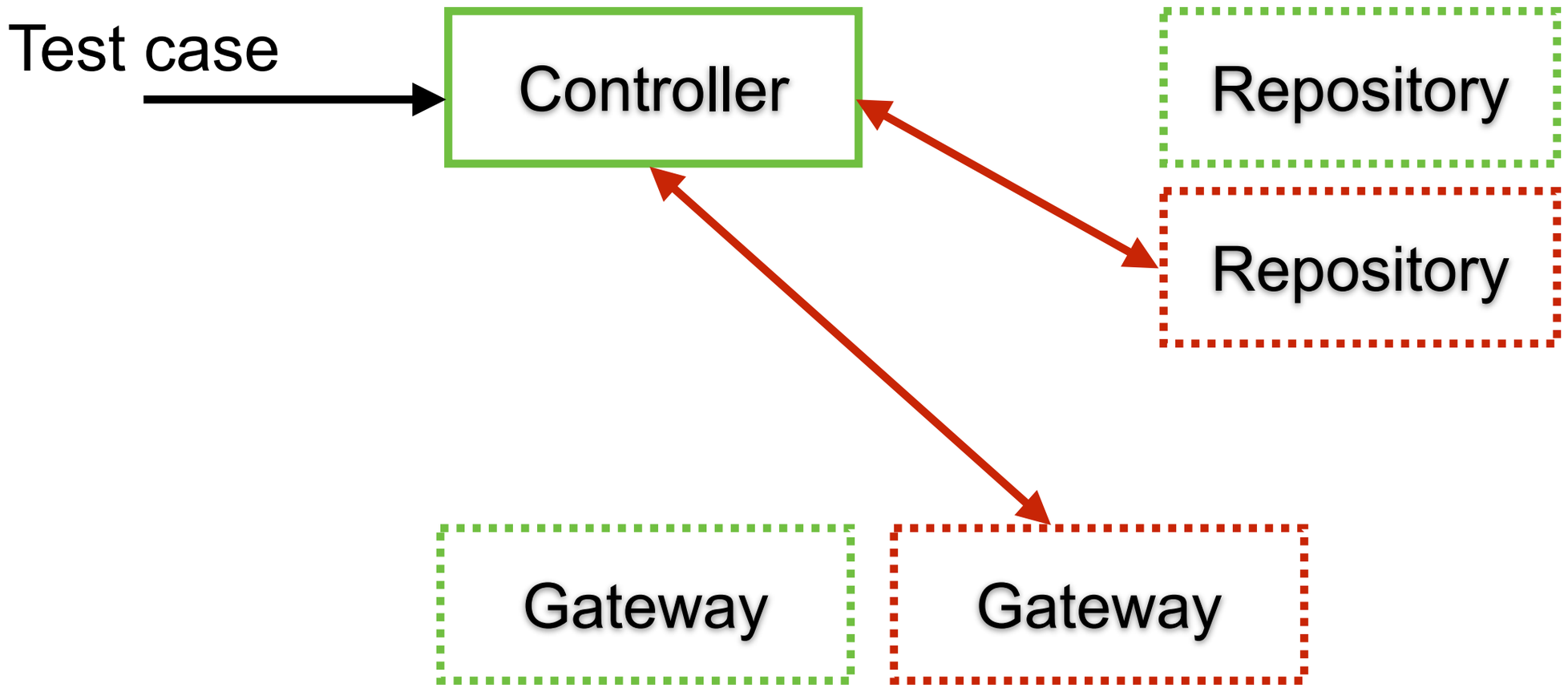
Good Unit Tests (F.I.R.S.T)

Fast
Independent/Isolate
Repeat
Self-validate
Thorough/Timely

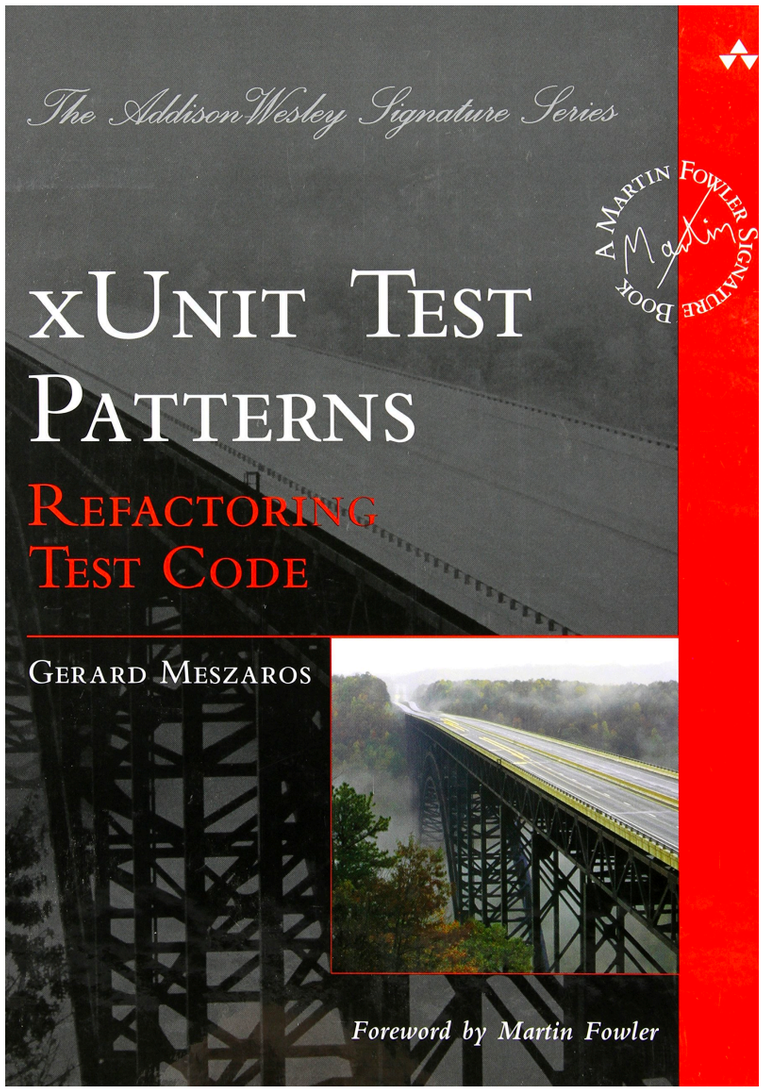
How to test Controller ?



Working with dependencies



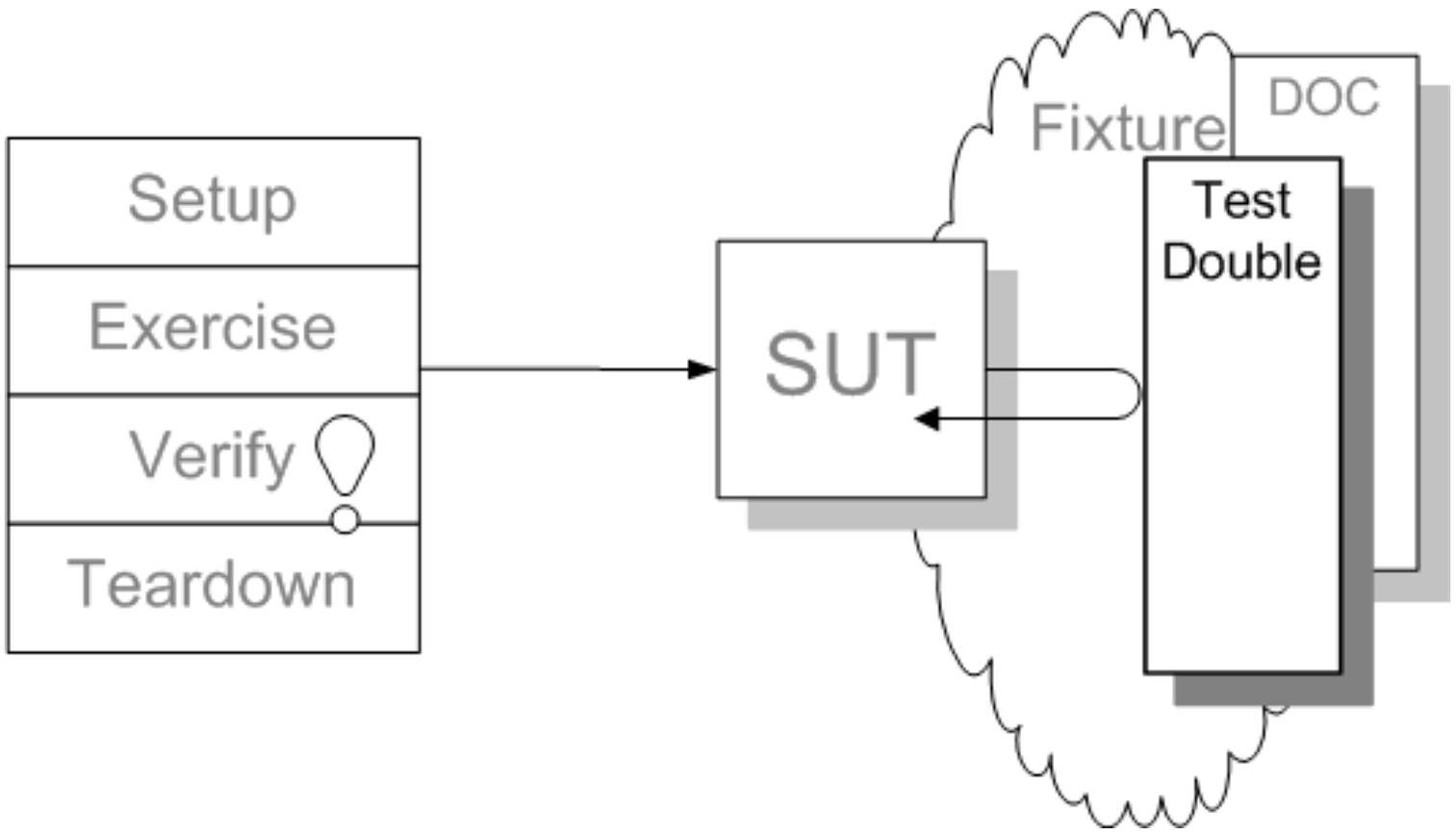
Test double ?



http://xunitpatterns.com

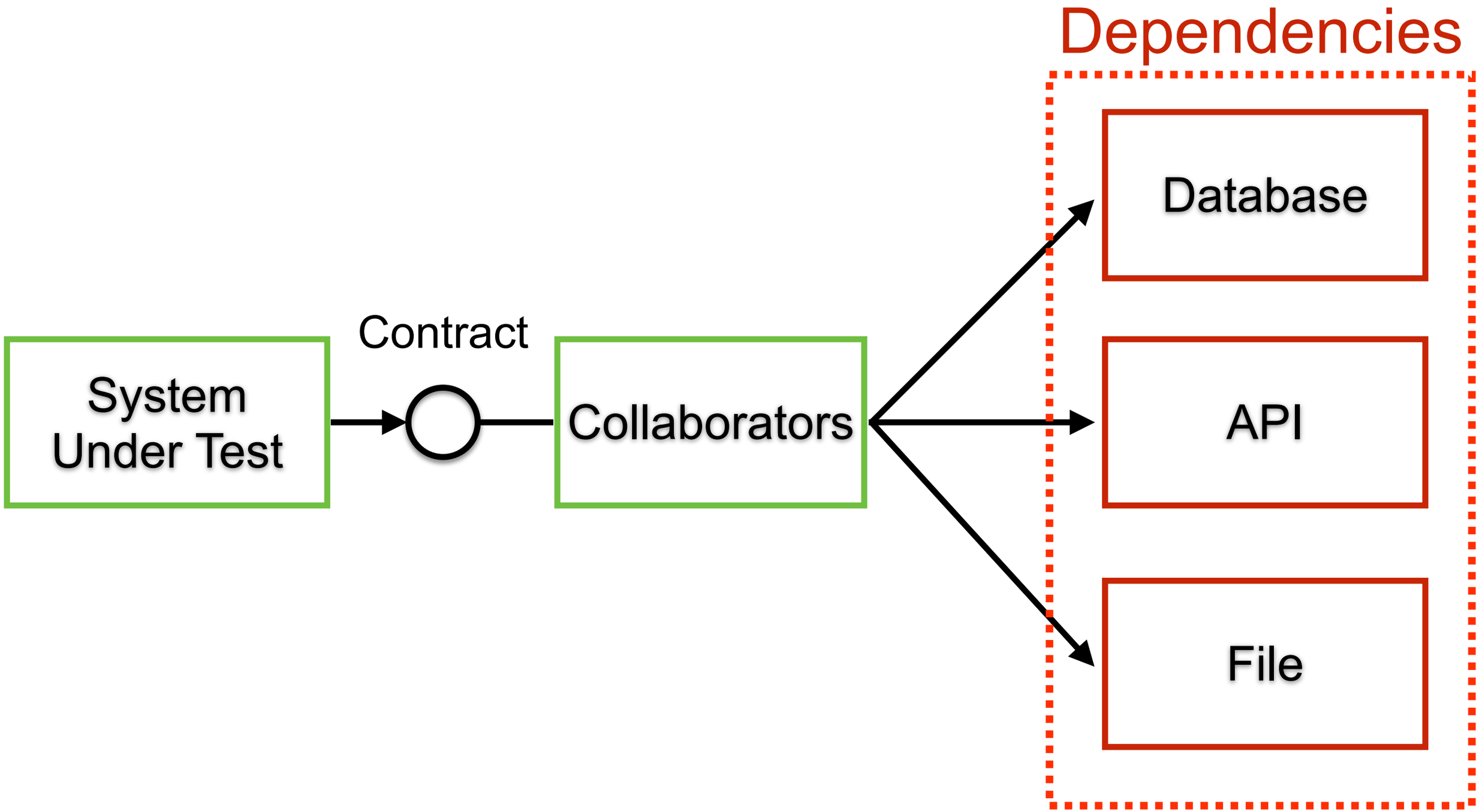
Test double ?

Stand-in for something that would be otherwise real in the execution of your program

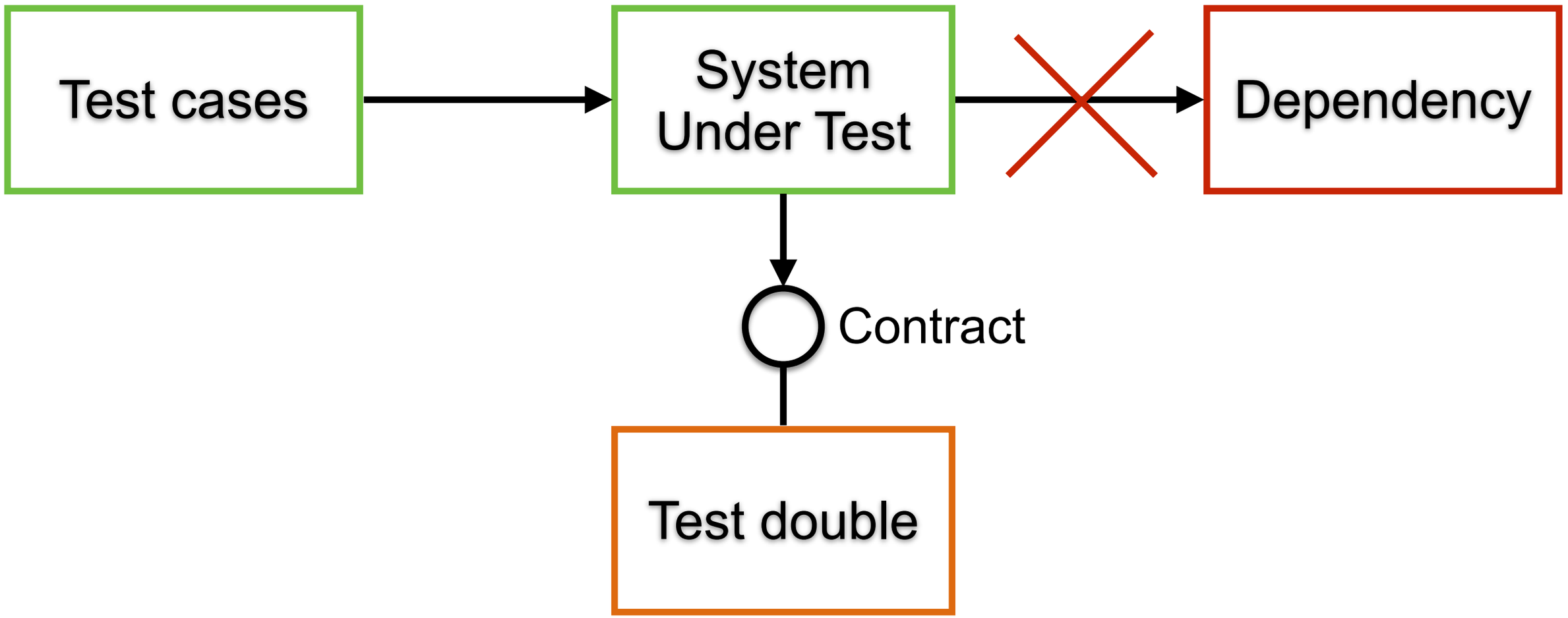


http://xunitpatterns.com/Test%20Double.html

Real implementation



With Test double



Types of Test double

Dummy object	Stub	Spy	Mock object	Fake object

Types of Test double

Dummy object

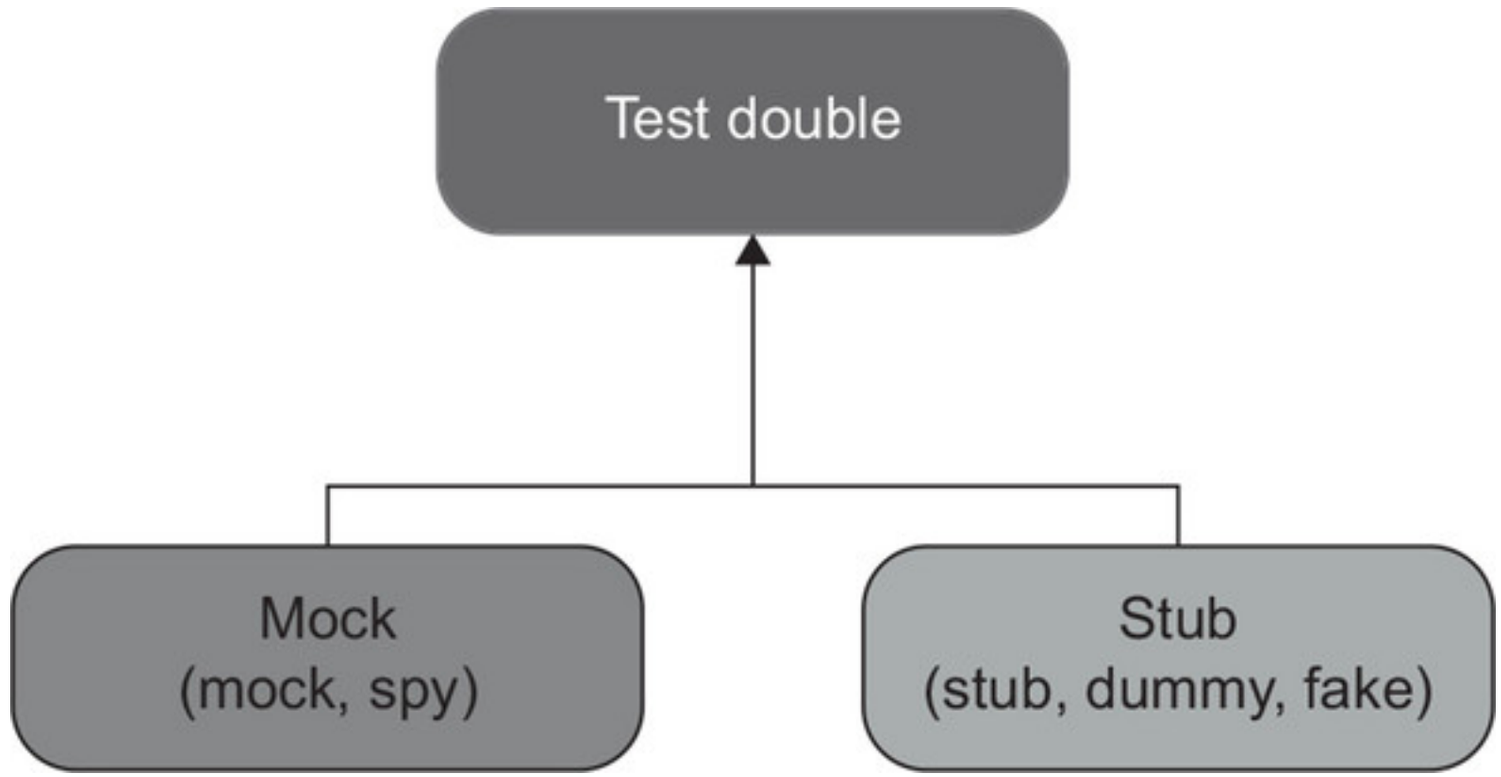
Stub

Fake object

Spy

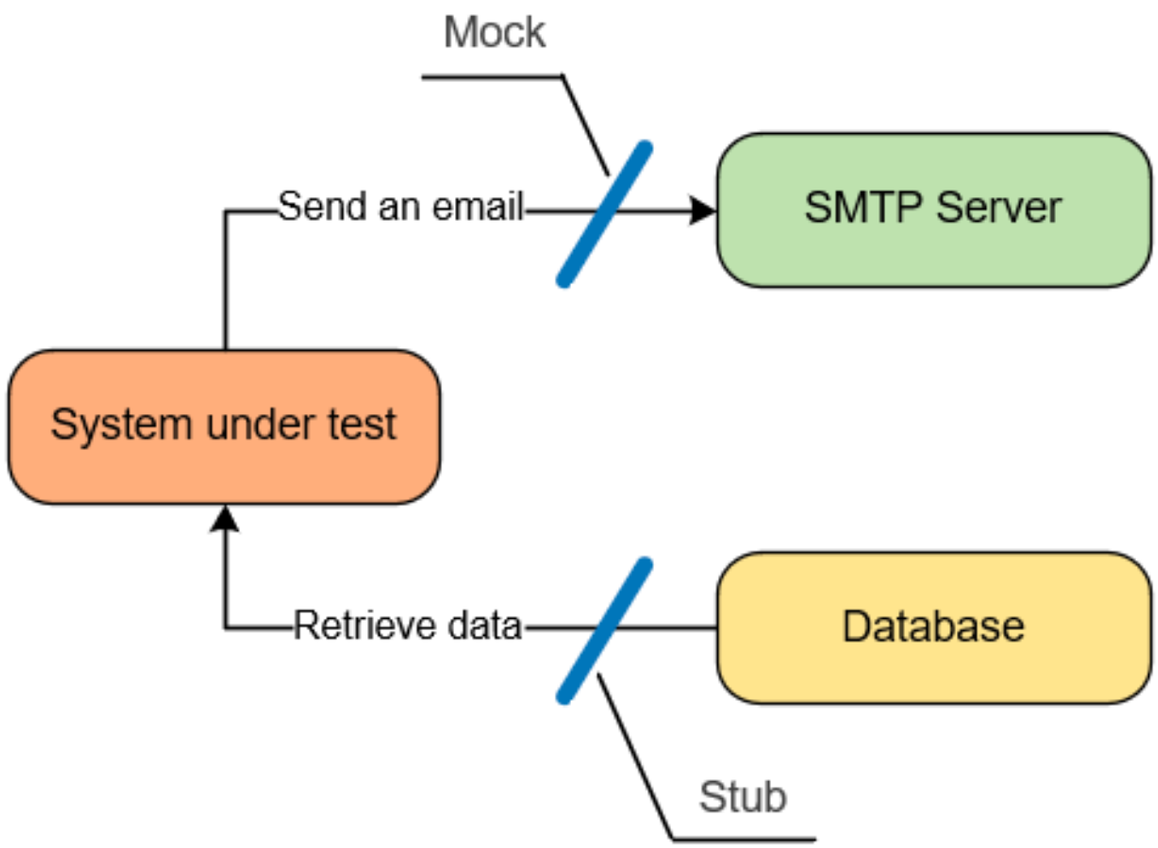
Mock object

Types of Test double

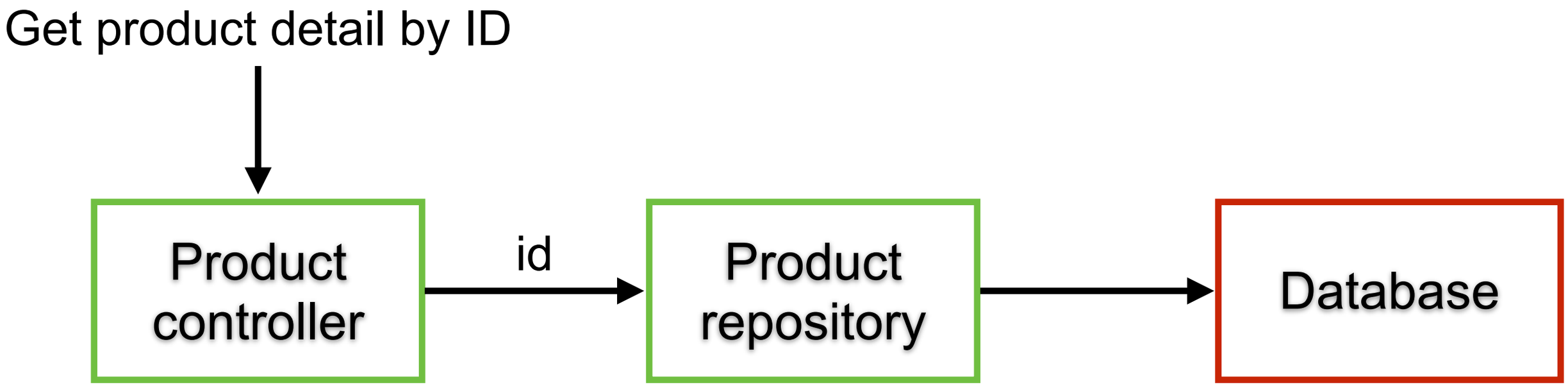


Mock vs Stub

Mock for outcoming interaction
Stub for incoming interaction



Example



Fake

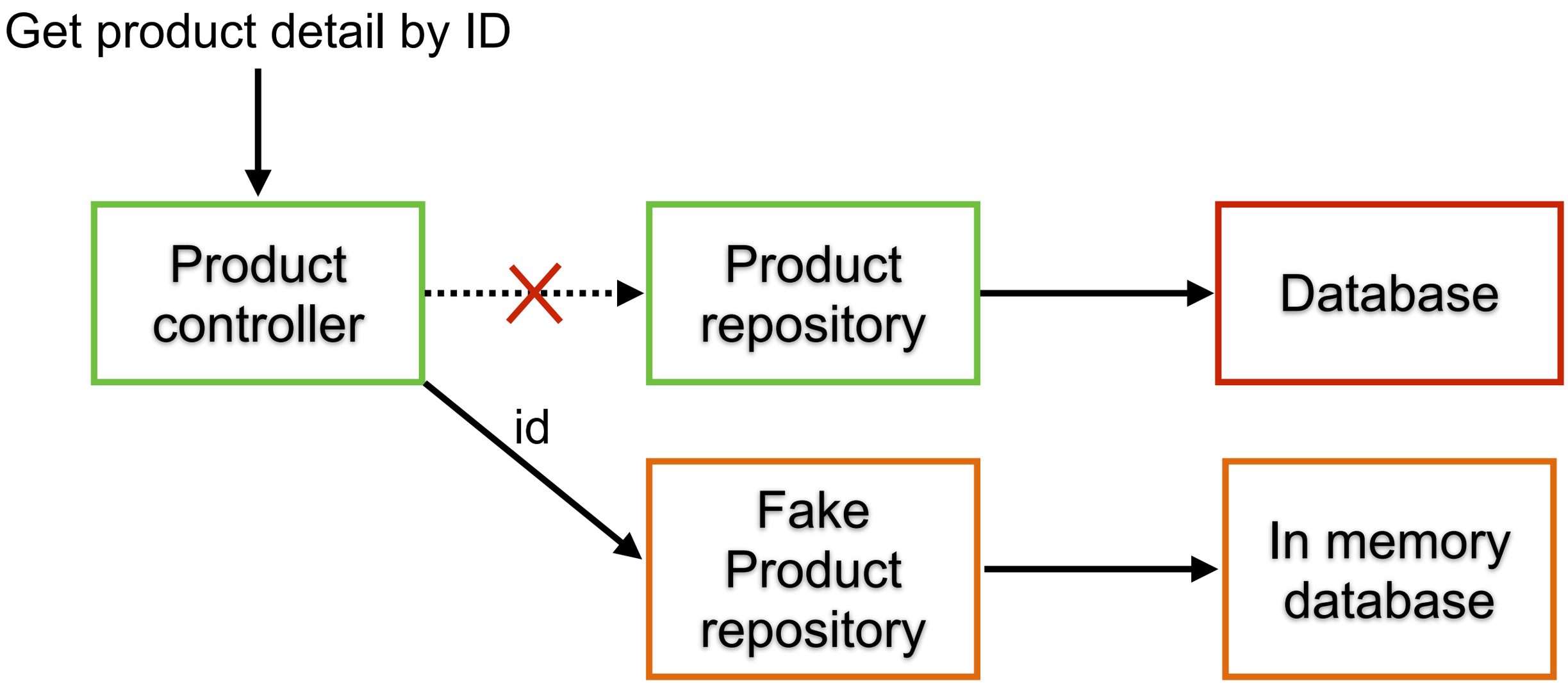
Working with implementation but take some shortcut

Not suitable for production

Use with read and write operations

E.g. In-memory database, Fake API server

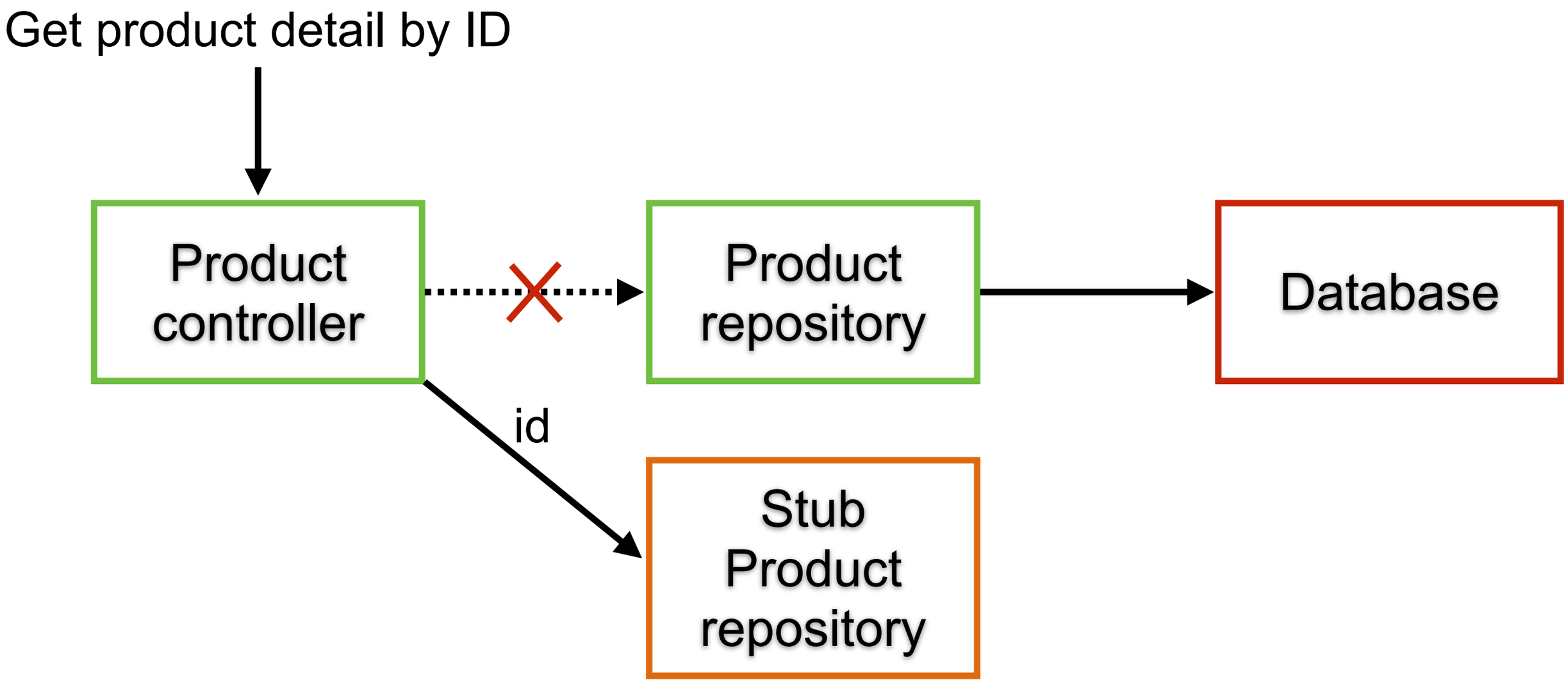
Fake



Stub

Provide answers to calls made during the test
A double with hardcode return values

Stub

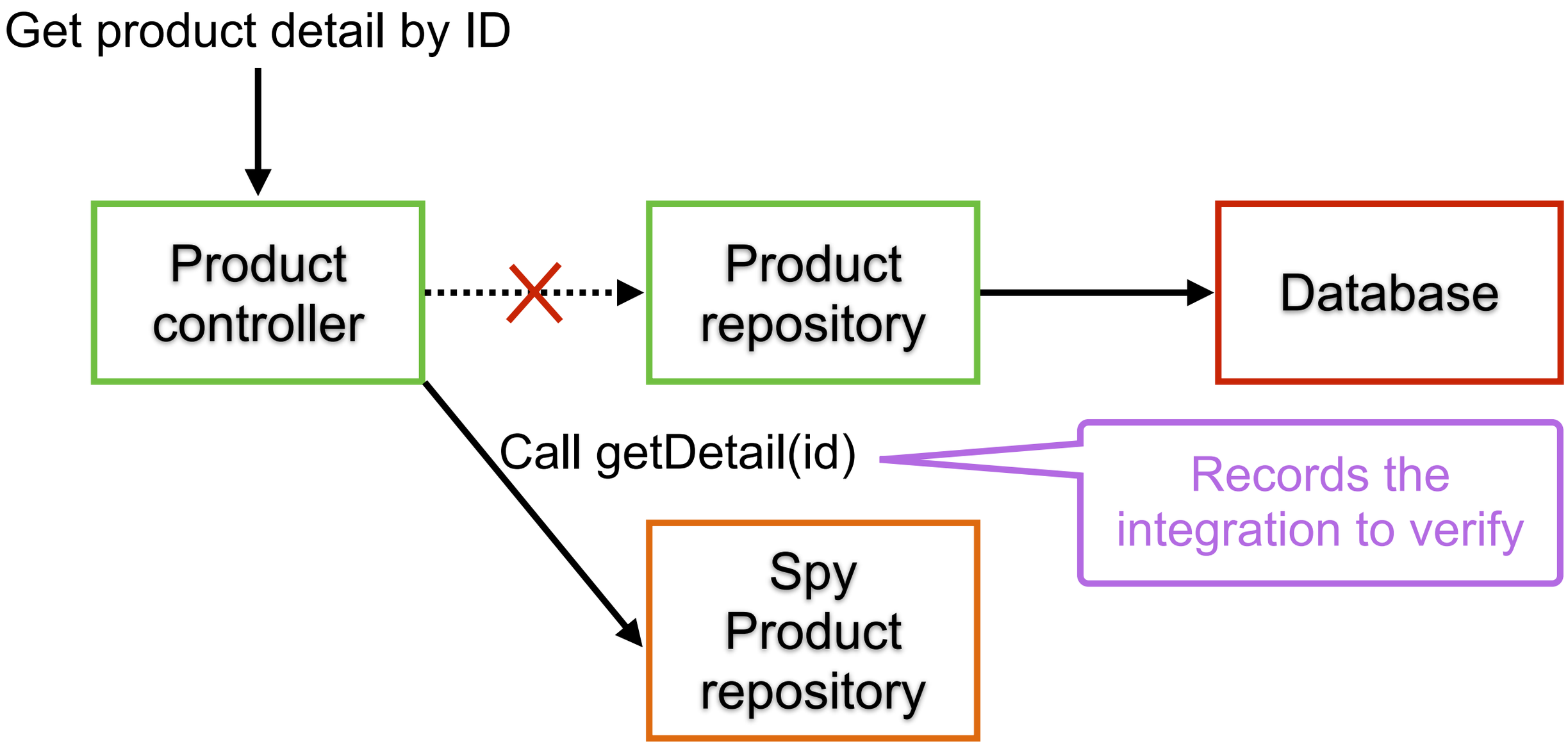


Spy

Like stub
Record some information based on how its called

E.g. how many message it was sent via email service

Spy

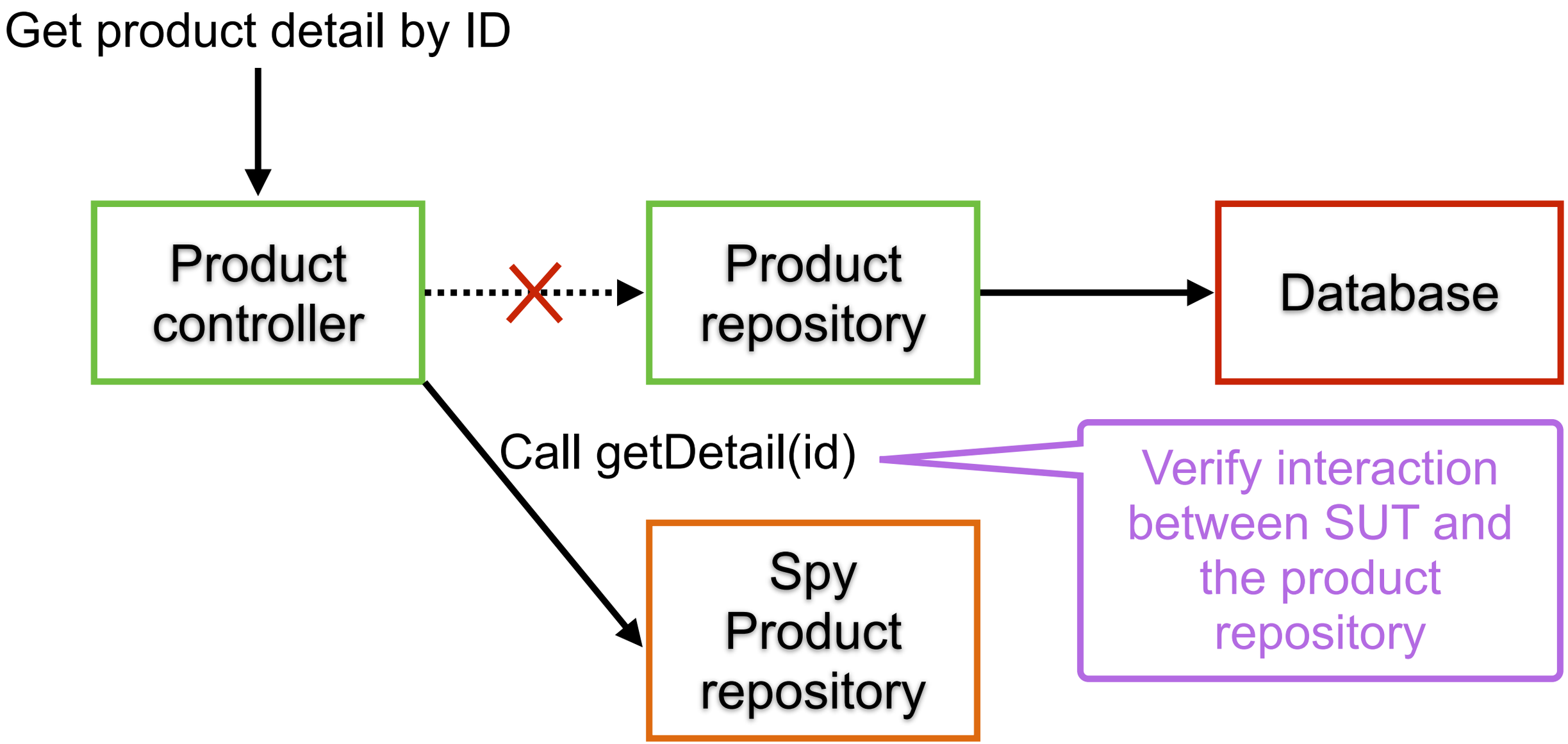


Mock object

Pre-programmed with expectations with spec

Mock object can throw an exception if receive a call that don’t expect

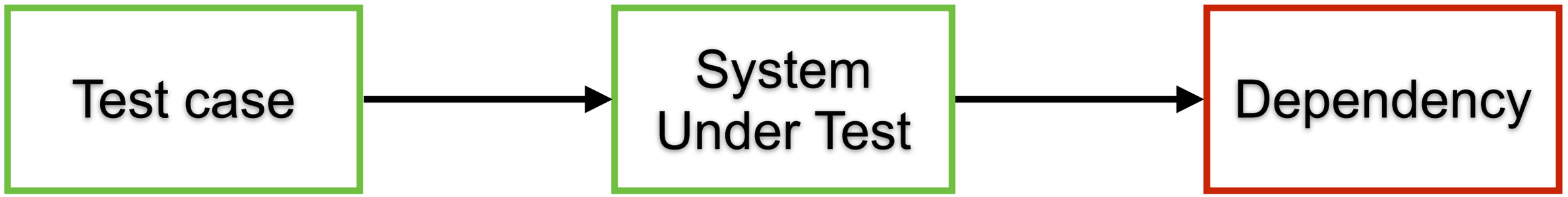
Mock object



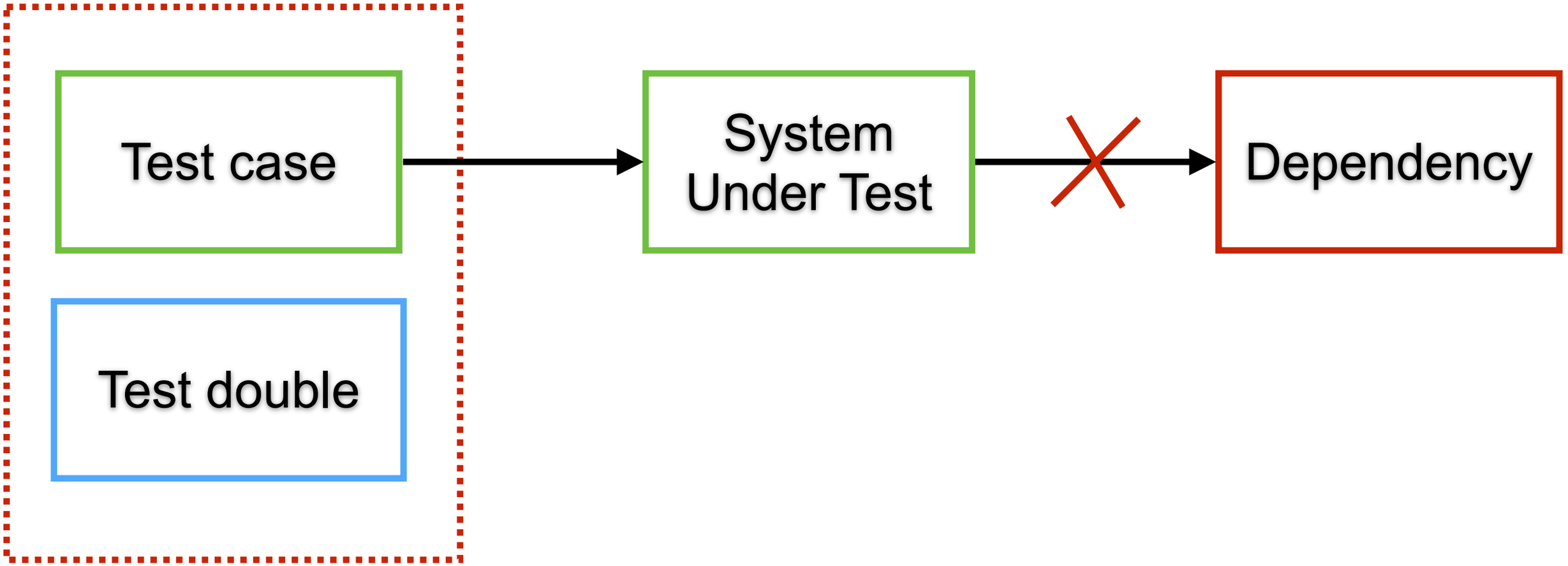
Dummy object

Passed around but never actually used
Used to fill parameter lists

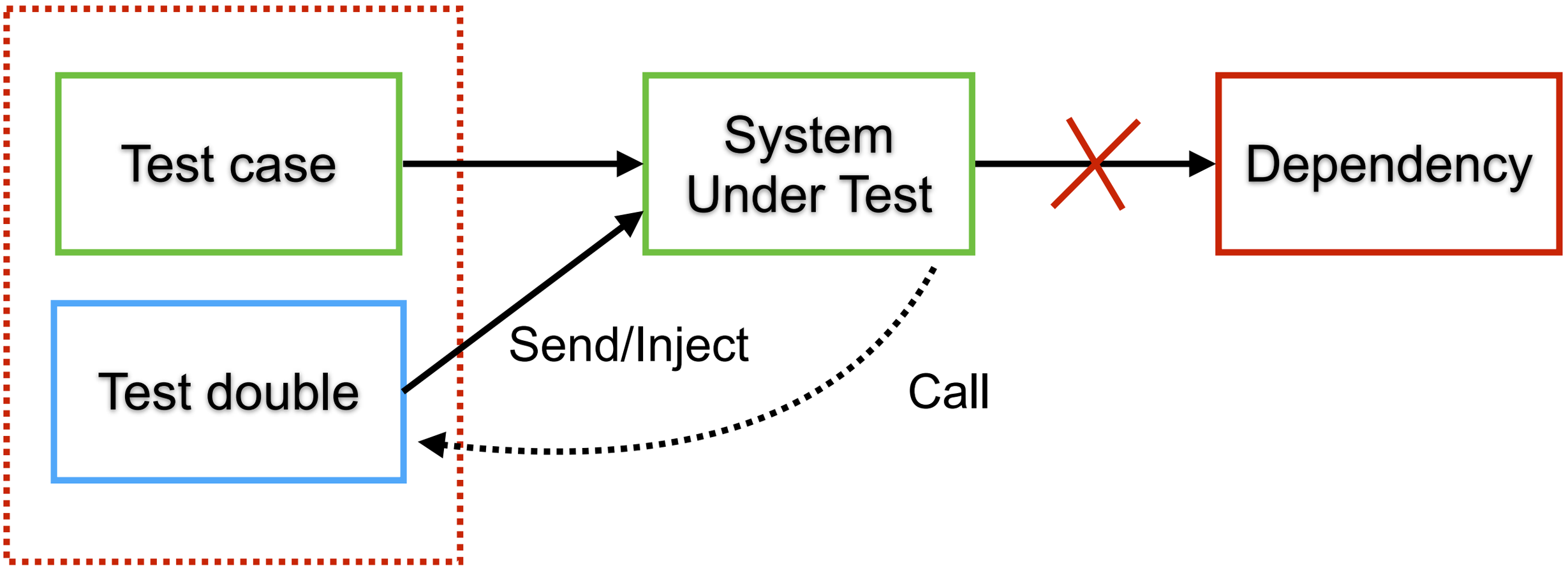
Test double ?



Create test double



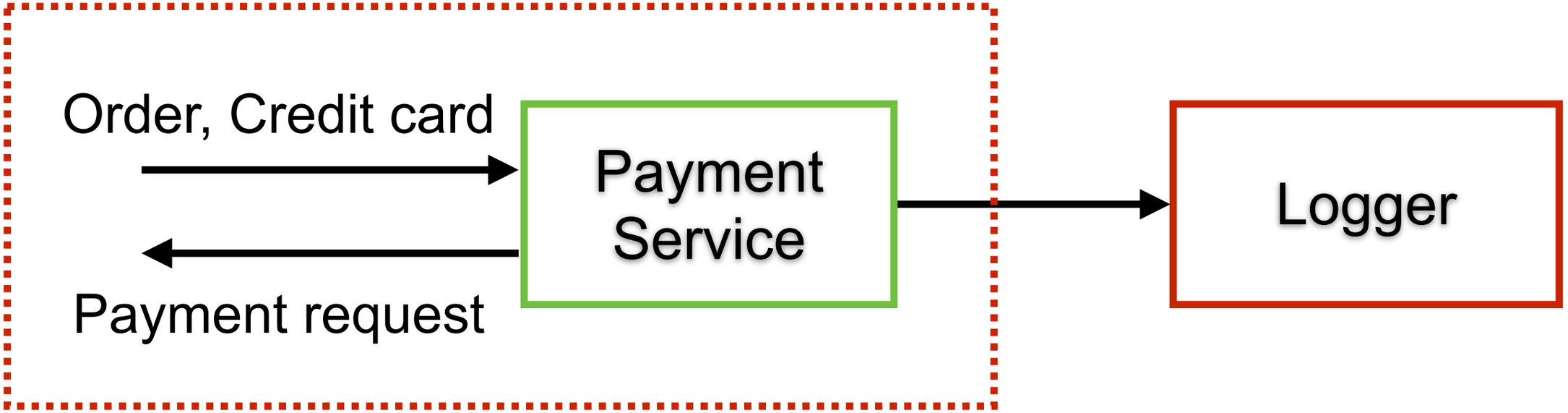
Send/inject test double to SUT



Workshop of Test Double

1. Dummy

Something pass to a constructor or method
but it not use !!



Example of Dummy

How to working with Logger ?

```
interface Logger {
    void append(String message);
}

public class PaymentService {

    private final Logger logger;

    public PaymentService(Logger logger) {
        this.logger = logger;
    }

    public PaymentRequest createPayment(Order order, CreditCard creditCard) {
        logger.append("Create payment for order " + order.toString());
        throw new RuntimeException("TODO Next");
    }
}
```

Create Logger Dummy

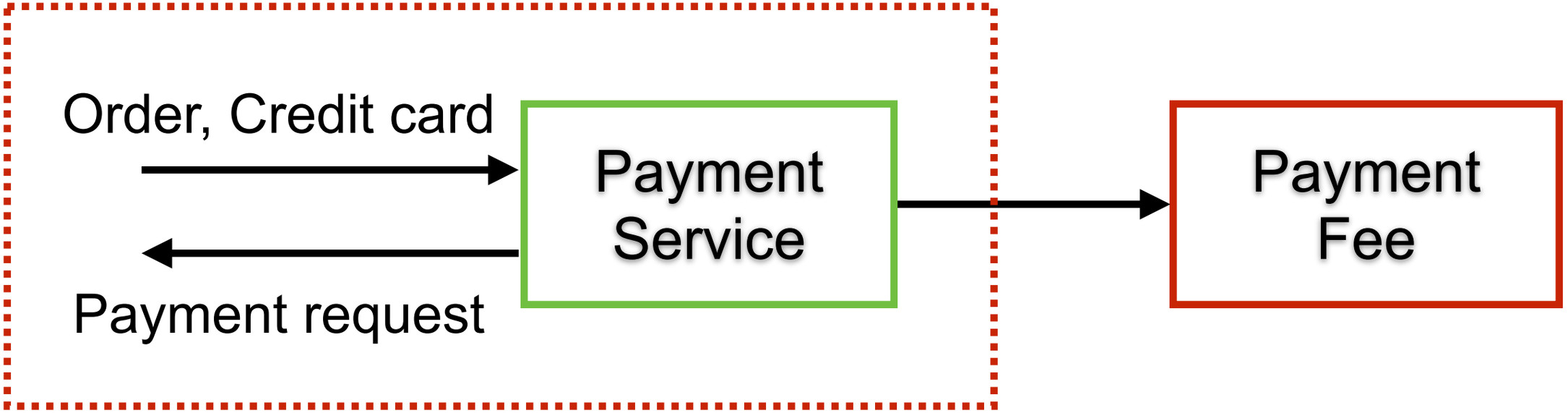
```
class LoggerDummy implements Logger {

    @Override
    public void append(String message) {
        // Do not thing
    }
}
```

Use Logger Dummy in test

```
class PaymentServiceTest {

    @Test
    public void working_with_dummy() {
        // Arrange
        Logger logger = new LoggerDummy();
        PaymentService paymentService = new PaymentService(logger);

        // Act and assert

    }

}
```

2. Stub

Provide answers for our call, don’t have logic
Return pre-defined value

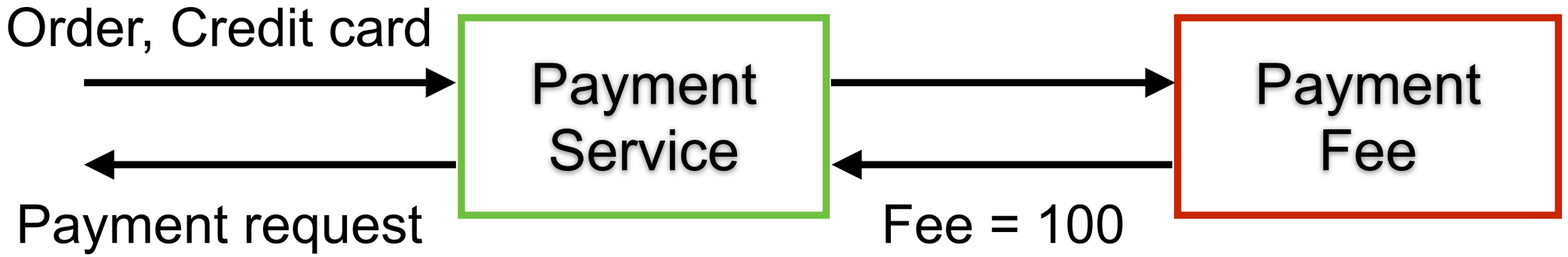


Example of Stub

How to working with Payment Fee ?

```
interface PaymentFee {
    int feeRate(String name);
}

public class PaymentService {

    private final Logger logger;
    private final PaymentFee paymentFee;

    public PaymentService(Logger logger, PaymentFee paymentFee) {
        this.logger = logger;
        this.paymentFee = paymentFee;
    }

    public PaymentRequest createPayment(Order order, CreditCard creditCard) {
        logger.append("Create payment for order " + order.toString());
        int fee = paymentFee.feeRate(creditCard.getName());
        return new PaymentRequest(order.getPrice(), fee);
    }
}
```

Example of Stub



Total price = 200

Create PaymentFee Stub

Return fee = 100

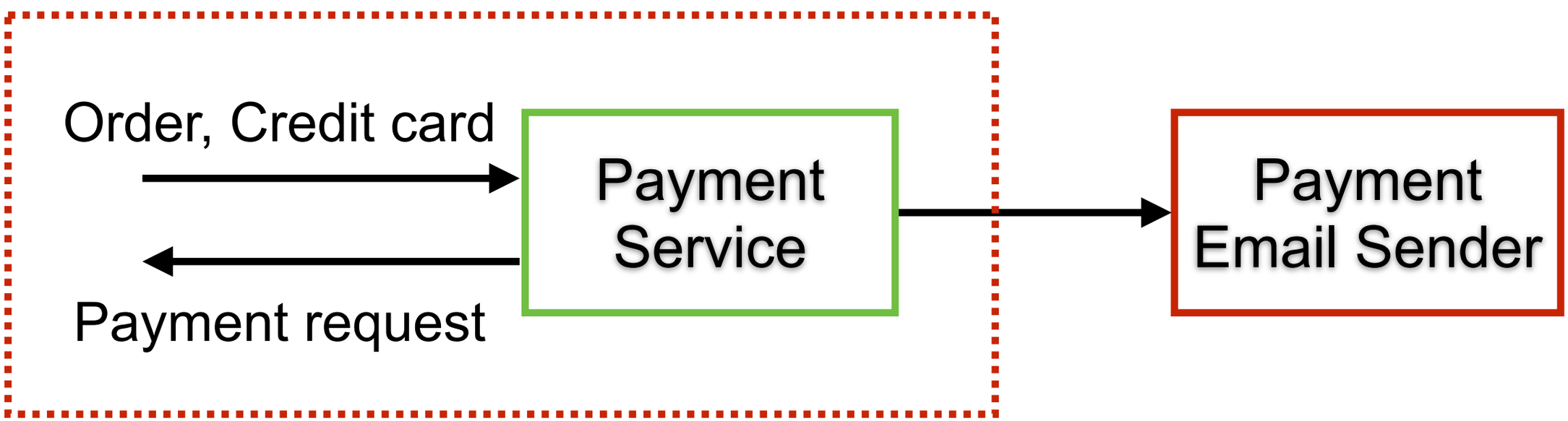
```
class PaymentFeeStub implements PaymentFee {

    @Override
    public int feeRate(String name) {
        return 100;
    }
}
```

Use Stub in test

```
@Test
public void working_with_stub() {
    // Arrange
    Logger logger = new LoggerDummy();
    PaymentFee paymentFee = new PaymentFeeStub();
    PaymentService paymentService =
            new PaymentService(logger, paymentFee);

    // Act
    PaymentRequest paymentRequest =
            paymentService.createPayment(new Order(100),
                                         new CreditCard());

    // Assert
    assertEquals(200, paymentRequest.getTotalPrice());
}
```

3. Spy

Try to record all movements just like a movie spy
Spy is silent
You can assert based on the data of spy

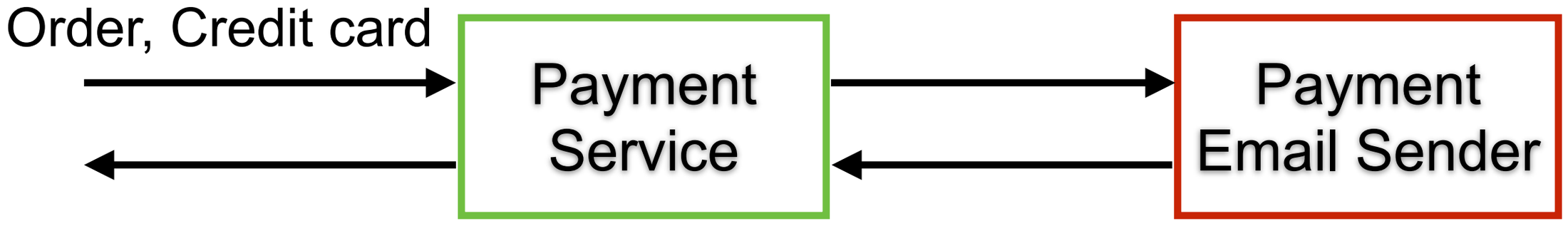


Example of Spy

How to working with Payment Email Sender ?

```
interface PaymentEmailSender {
    void send(PaymentRequest paymentRequest);
}

public class PaymentService {

    private final Logger logger;
    private final PaymentFee paymentFee;
    private final PaymentEmailSender paymentEmailSender;

    public PaymentRequest createPayment(Order order, CreditCard creditCard) {
        logger.append("Create payment for order " + order.toString());
        int fee = paymentFee.feeRate(creditCard.getName());
        PaymentRequest paymentRequest =
                        new PaymentRequest(order.getPrice(), fee);

        paymentEmailSender.send(paymentRequest);

        return paymentRequest;
    }
}
```

Example of Spy



Call method send() ?

Create PaymentEmailSender Spy

Recording how many time to called send()

```
class PaymentEmailSenderSpy implements PaymentEmailSender {

    private boolean called = false;

    @Override
    public void send(PaymentRequest paymentRequest) {
        called = true;
    }

    public boolean calledSend() {
        return called;
    }
}
```

Create PaymentEmailSender Spy

Recording how many time to called send()

```
class PaymentEmailSenderSpy implements PaymentEmailSender {

    List<PaymentRequest> paymentRequestList = new ArrayList<>();

    @Override
    public void send(PaymentRequest paymentRequest) {
        paymentRequestList.add(paymentRequest);
    }

    public int timesCalled() {
        return paymentRequestList.size();
    }

    public boolean calledWithPaymentRequest(PaymentRequest paymentRequest) {
        return paymentRequestList.contains(paymentRequest);
    }
}
```

Use Spy in test

Spy vs Stub ?

```
@Test
public void working_with_spy() {
    // Arrange
    Logger logger = new LoggerDummy();
    PaymentFee paymentFee = new PaymentFeeStub();
    PaymentService paymentService = new PaymentService(logger, paymentFee);
    PaymentEmailSenderSpy emailSender = new PaymentEmailSenderSpy();

    // Act
    PaymentRequest paymentRequest =
            paymentService.createPayment(new Order(100), new CreditCard());

    // Assert
    assertEquals(1, emailSender.timesCalled());
    assertTrue(emailSender.calledWithPaymentRequest(paymentRequest));
}
```

Use Spy in test

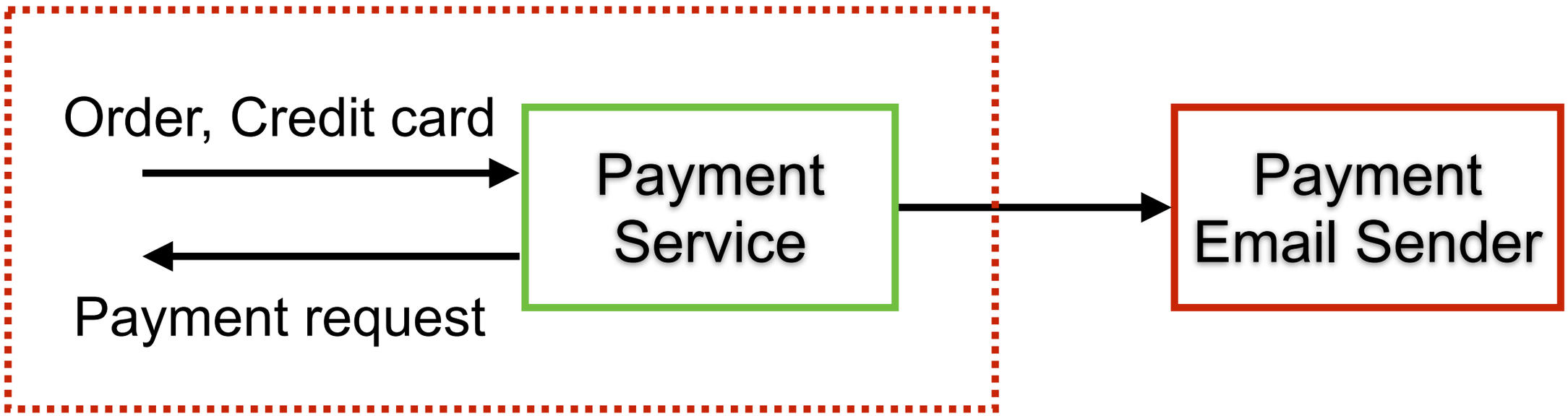
Spy vs Stub ?

```
@Test
public void working_with_spy() {
    // Arrange
    Logger logger = new LoggerDummy();
    PaymentFee paymentFee = new PaymentFeeStub();
    PaymentService paymentService = new PaymentService(logger, paymentFee);
    PaymentEmailSenderSpy emailSender = new PaymentEmailSenderSpy();

    // Act
    PaymentRequest paymentRequest =
            paymentService.createPayment(new Order(100), new CreditCard());

    // Assert
    assertEquals(1, emailSender.timesCalled());
    assertTrue(emailSender.calledWithPaymentRequest(paymentRequest));
}
```

4. Mock

Use to verify the behaviour between SUT and its collaborators

Mock have a own logic and assertion



Test case

ทำการส่ง email เมื่อราคารวม order มากกว่า 1,000 บาท
โดยที่ email ที่ส่งคือ order ที่ถูกสั่งซื้อ

Create EmailSender Mock

Try to record the behaviour of dependencies

```
class PaymentEmailSenderMock implements PaymentEmailSender {

    List<PaymentRequest> paymentRequestList = new ArrayList<>();
    List<PaymentRequest> expectedRequestList = new ArrayList<>();

    @Override
    public void send(PaymentRequest paymentRequest) {
        paymentRequestList.add(paymentRequest);
    }

    public void setExpectedRequestList(PaymentRequest paymentRequest) {
        expectedRequestList.add(paymentRequest);
    }

    public void verify() {
        assertEquals(expectedRequestList, paymentRequestList);
    }
}
```

Use Mock in test

```
@Test
public void working_with_mock() {
    // Arrange
    …
    PaymentEmailSenderMock emailSender = new PaymentEmailSenderMock();
    PaymentRequest expectedRequest = new PaymentRequest(1000, 10);

    // Act
    PaymentRequest paymentRequest =
            paymentService.createPayment(new Order(1000), new CreditCard());

    // Assert
    emailSender.setExpectedRequestList(expectedRequest);
    emailSender.verify();
}
```

Spy vs Mock ?

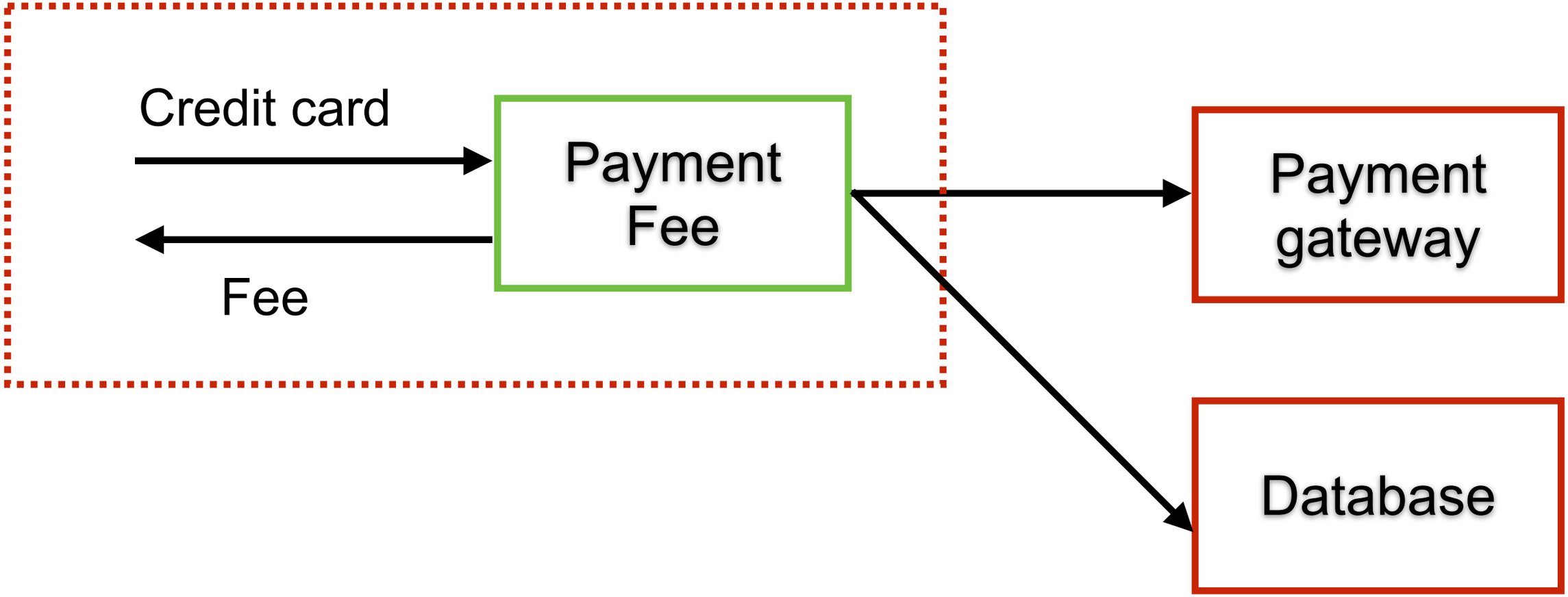
Why use Spy ?

When not sure about what SUT will call from the collaborator.

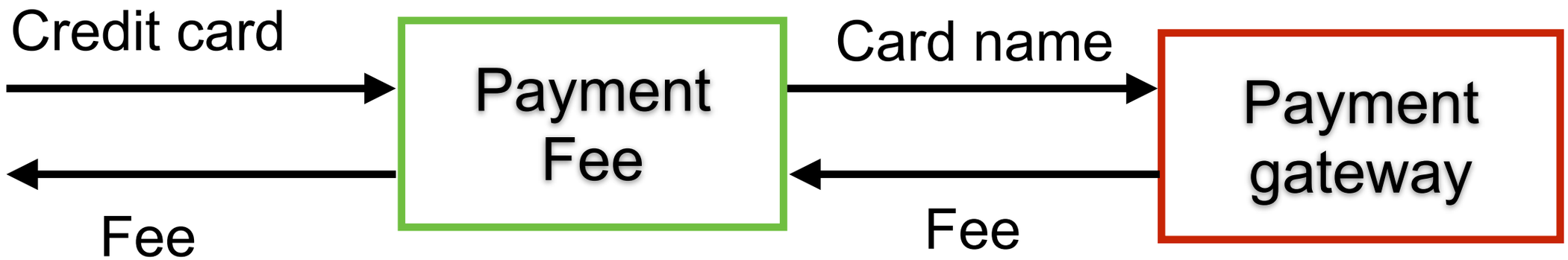
Try to record everything and assert

5. Fake

Different from others
Like the production system but its not real !!
Simplified version of the business logic



Create PaymentGateway Fake



CreditCardFee

```
public class CreditCardRate implements PaymentFee {

    private final PaymentGateway paymentGateway;

    public CreditCardRate(PaymentGateway paymentGateway) {
        this.paymentGateway = paymentGateway;
    }

    @Override
    public int feeRate(String name) {
        String feeRate = paymentGateway.rateFor(name);
        return Integer.parseInt(feeRate);
    }
}
```

Create PaymentGateway Fake

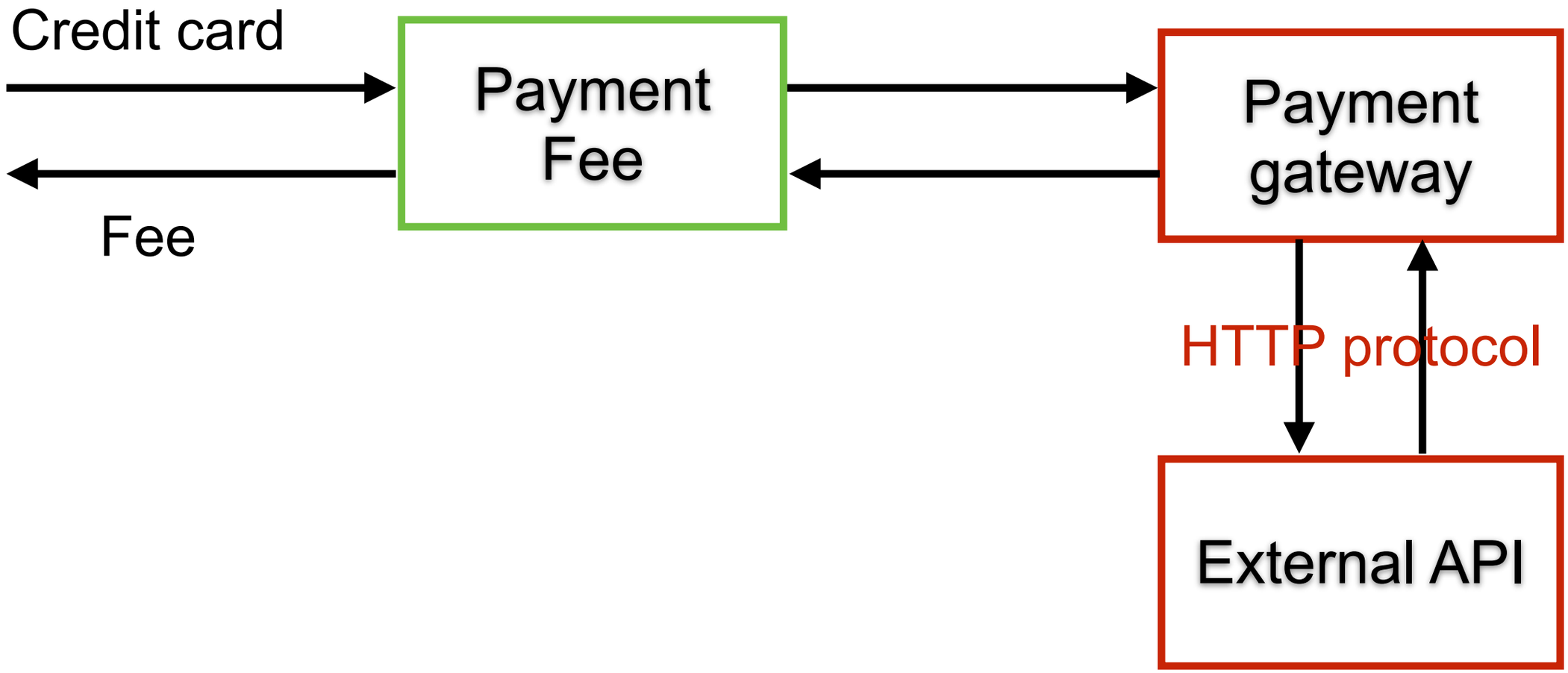
```
class PaymentGatewayFake implements PaymentGateway {
    @Override
    public String rateFor(String name) {
        if("VISA".equals(name)) {
            return "4";
        } else if("MASTER".equals(name)) {
            return "3";
        }
        return "2";
    }
}
```

Use Fake in test

```
class CreditCardRateTest {

    @Test
    public void working_with_fake() {
        // Arrange
        PaymentGatewayFake paymentGatewayFake =
                           new PaymentGatewayFake();
        CreditCardRate creditCardRate =
                           new CreditCardRate(paymentGatewayFake);

        // Act
        int result = creditCardRate.feeRate("VISA");

        // Assert
        assertEquals(4, result);
    }

}
```

Create PaymentGateway Fake

Working with Network protocol

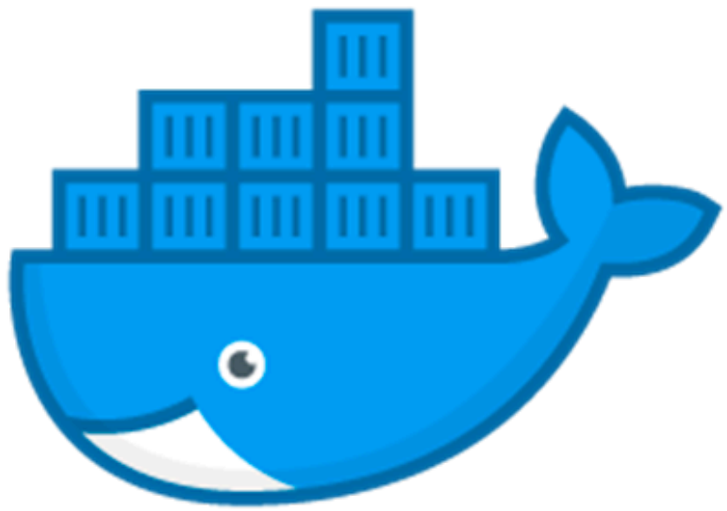


Fake API Server

mountebank - over the wire test doubles

home imposters

Container



Working with WireMock

https://github.com/tomakehurst/wiremock

https://github.com/lanwen/wiremock-junit5

Add dependencies in Gradle

```
dependencies {
    testImplementation "com.github.tomakehurst:wiremock:2.27.2"
    testImplementation “ru.lanwen.wiremock:wiremock-junit5:1.3.1"
    testImplementation "io.rest-assured:rest-assured:4.3.1"
}
```

REST-assured

https://rest-assured.io/

Setup WireMock server in test

```
class CreditCardRateWithAPIServerTest {

    WireMockServer wireMockServer;

    @BeforeEach
    public void setup() {
        wireMockServer = new WireMockServer(8090);
        wireMockServer.start();
        setupStub();
    }

    @AfterEach
    public void teardown() {
        wireMockServer.stop();
    }

    public void setupStub() {
        wireMockServer.stubFor(get(urlEqualTo("/users/1"))
                .willReturn(aResponse()
                .withHeader("Content-Type", "application/json")
                .withStatus(200)
                .withBodyFile("json/hello.json")));
    }
}
```

Setup WireMock server in test

```
class CreditCardRateWithAPIServerTest {

    WireMockServer wireMockServer;

    @BeforeEach
    public void setup() {
        wireMockServer = new WireMockServer(8090);
        wireMockServer.start();
        setupStub();
    }

    @AfterEach
    public void teardown() {
        wireMockServer.stop();
    }

    public void setupStub() {
        wireMockServer.stubFor(get(urlEqualTo("/users/1"))
                .willReturn(aResponse()
                .withHeader("Content-Type", "application/json")
                .withStatus(200)
                .withBodyFile("json/hello.json")));
    }
}
```

Start and stop server in each test

Setup WireMock server in test

```
class CreditCardRateWithAPIServerTest {

    WireMockServer wireMockServer;

    @BeforeEach
    public void setup() {
        wireMockServer = new WireMockServer(8090);
        wireMockServer.start();
        setupStub();
    }

    @AfterEach
    public void teardown() {
        wireMockServer.stop();
    }

    public void setupStub() {
        wireMockServer.stubFor(get(urlEqualTo("/users/1"))
                .willReturn(aResponse()
                .withHeader("Content-Type", "application/json")
                .withStatus(200)
                .withBodyFile("json/hello.json")));
    }
}
```

Setup request and response of API

Create test case to call server

```
@Test
public void working_with_wiremock_200() {
    given().
            when().
            get("http://localhost:8090/users/1").
            then().
            assertThat().statusCode(200);
}
```

```
@Test
public void working_with_wiremock_404() {
    given().
            when().
            get("http://localhost:8090/users/2").
            then().
            assertThat().statusCode(404);
}
```