SpectoLabs

Effective Testing with API Simulation and (Micro)Service Virtualisation

Andrew Morgan | Daniel Bryant
@mogronalol | @danielbryantuk

@Spectolabs

tl;dr

SpectoLabs

- Testing distributed systems is hard
- Many of us building distributed systems (microservices)
- API simulation is a new (old) tool to help with testing

About Us

SpectoLabs



Andrew Morgan

@mogronalol

- Independent Consultant
- Microservices / Cloud / Continuous Delivery
- InfoQ Editor



Daniel Bryant

@danielbryantuk

- Consultant/CTO SpectoLabs
- Cloud, containers, and microservices
- O'Reilly book on the way!


SpectoLabs

Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

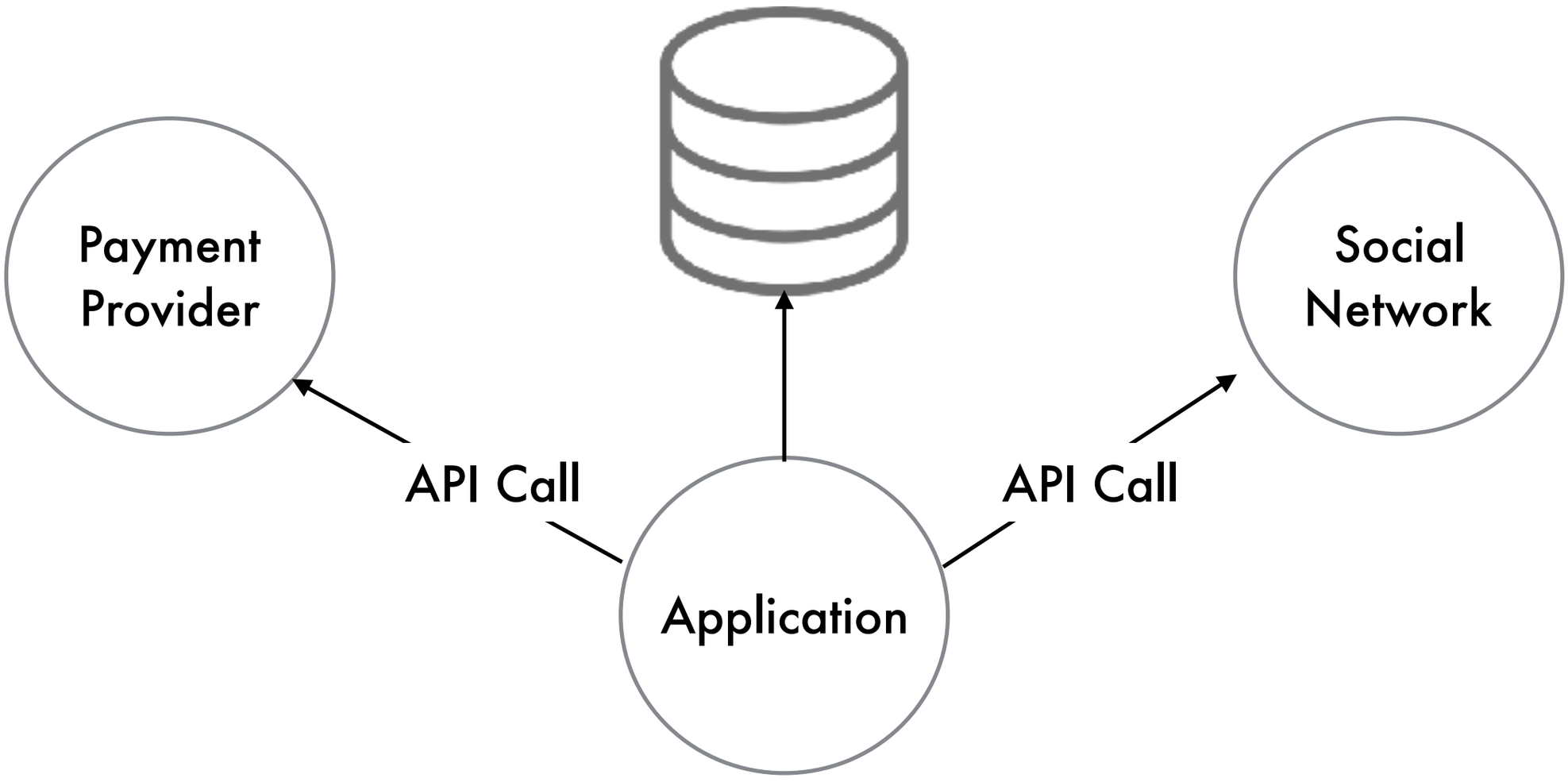
Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

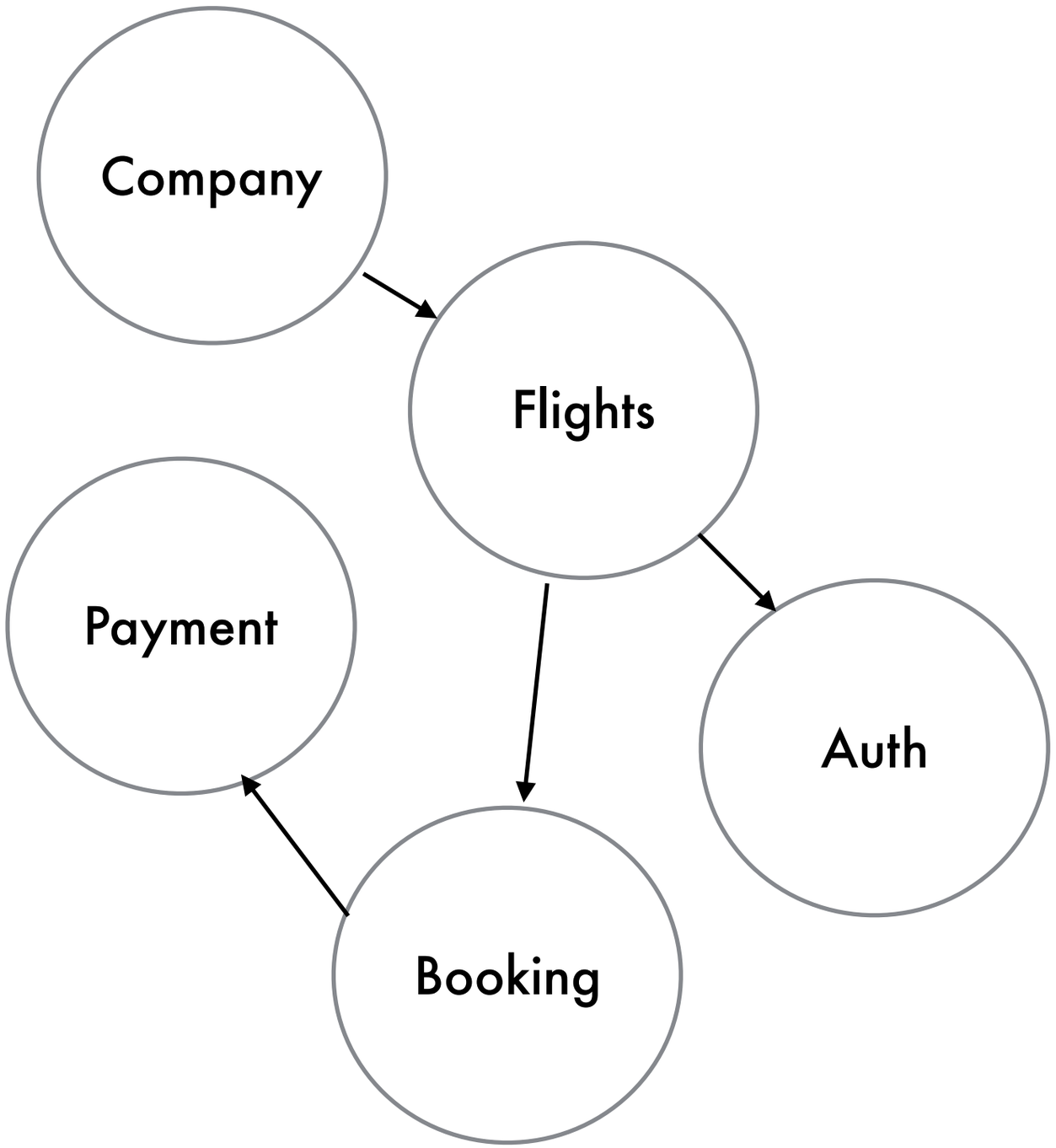
API-Driven Architecture

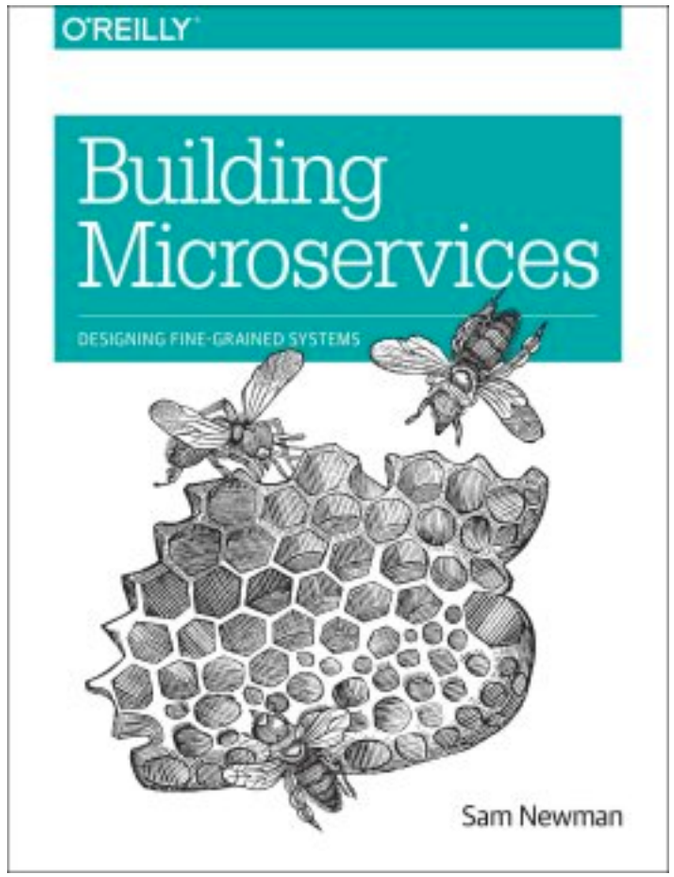
SpectoLabs



Microservices

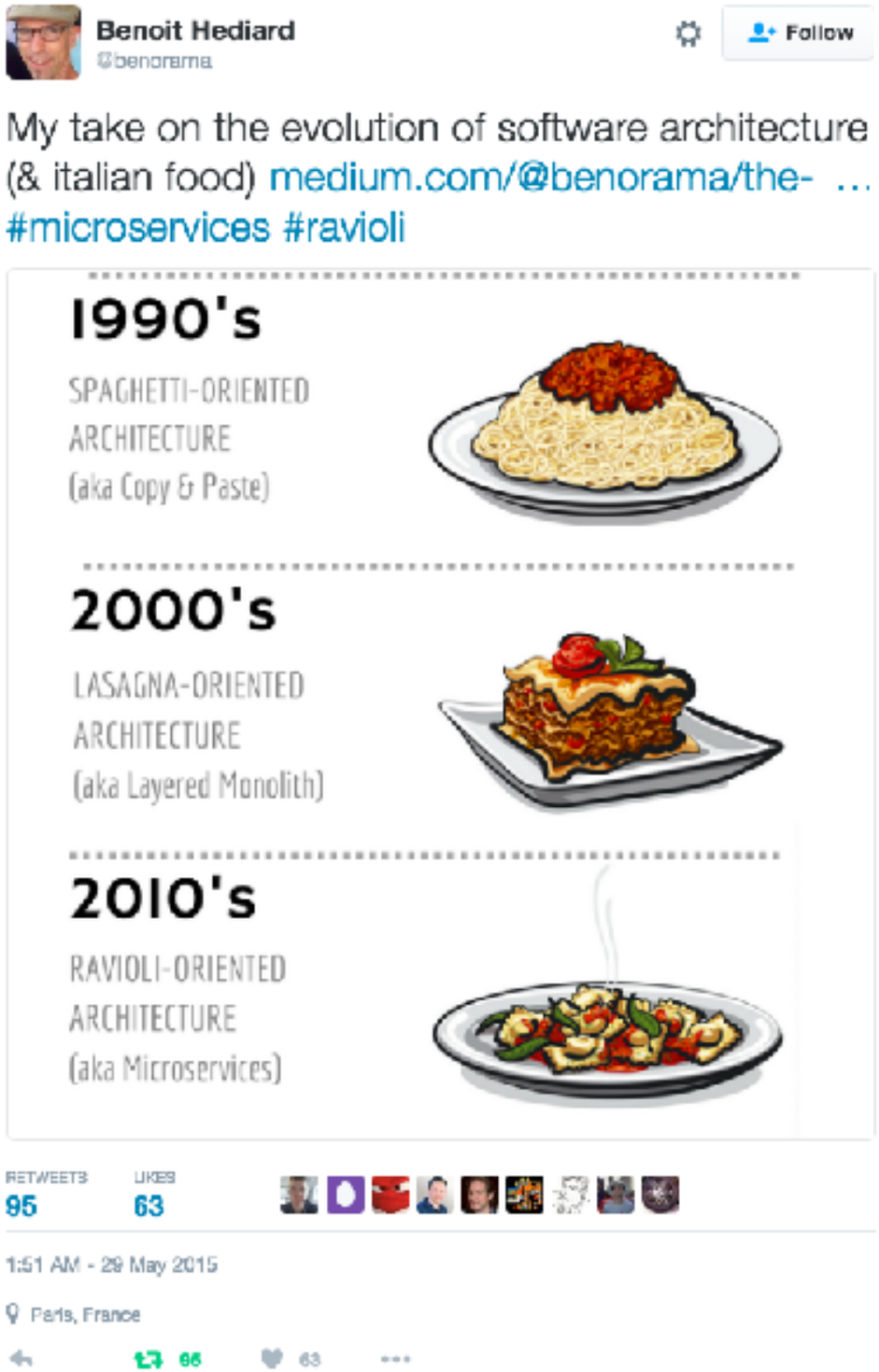
SpectoLabs





A natural evolution...

SpectoLabs



Challenges...

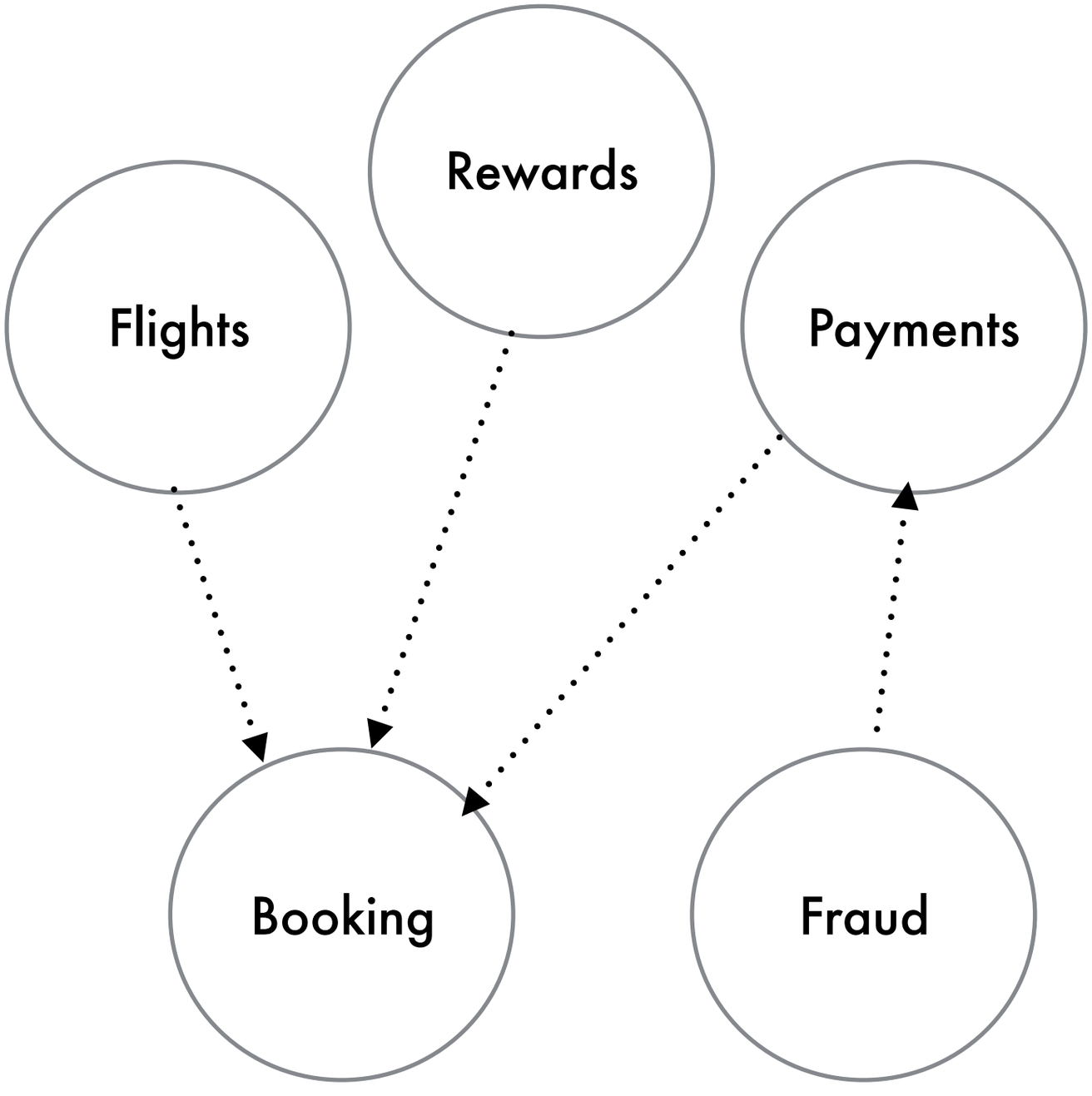
Airline Booking Application

SpectoLabs

- Microservice architecture
- Integration with third party services
- Integration with legacy internal services
- Heavy resource consumption
- A standard CI/CD pipeline

Resources

SpectoLabs

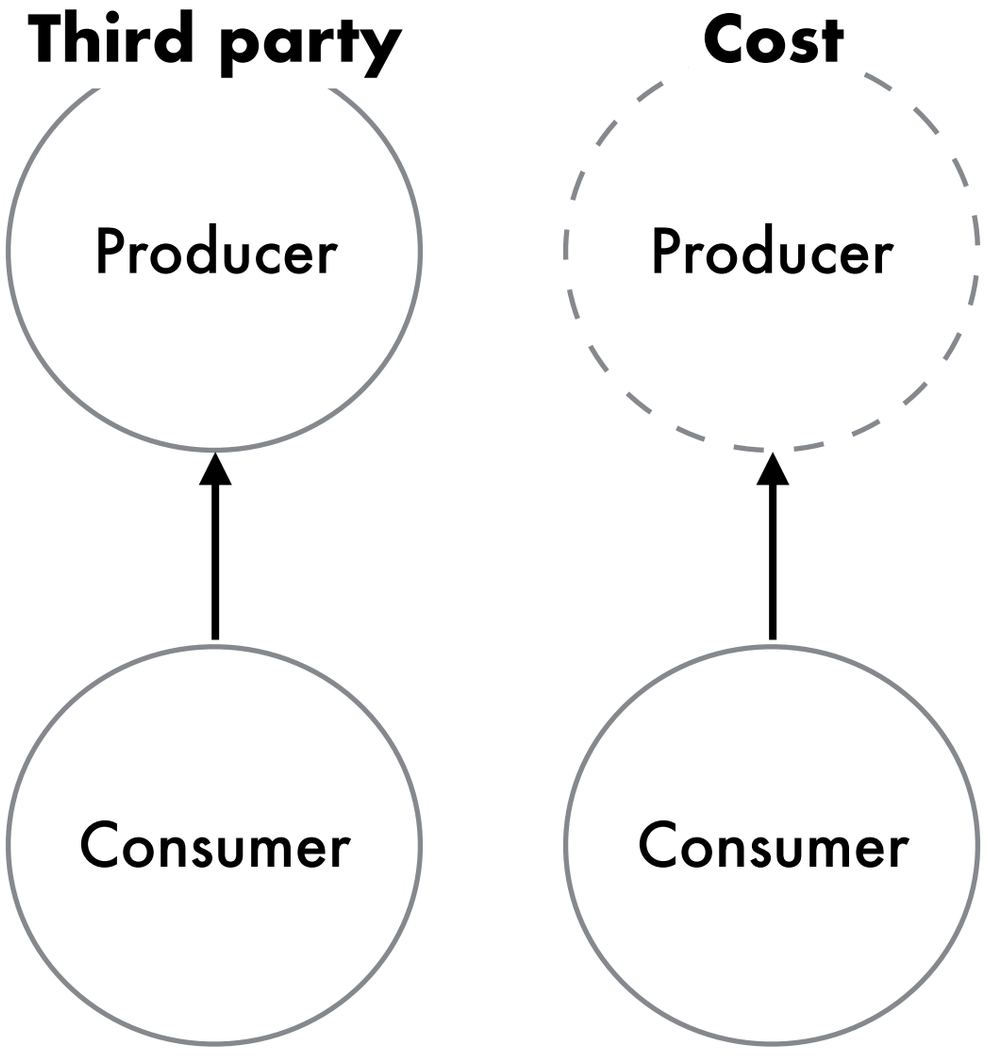


amazon web services

Cost

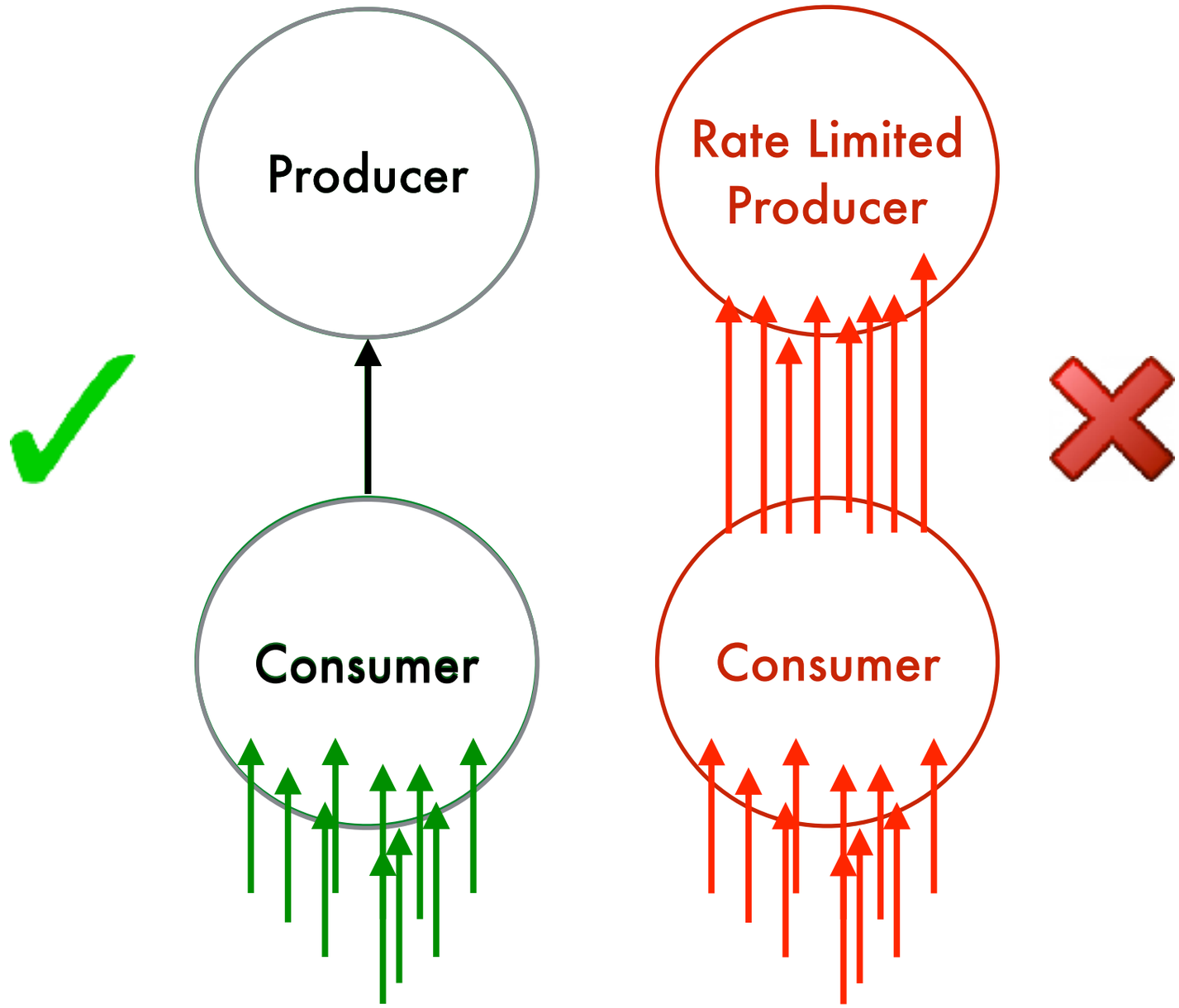


Licensing



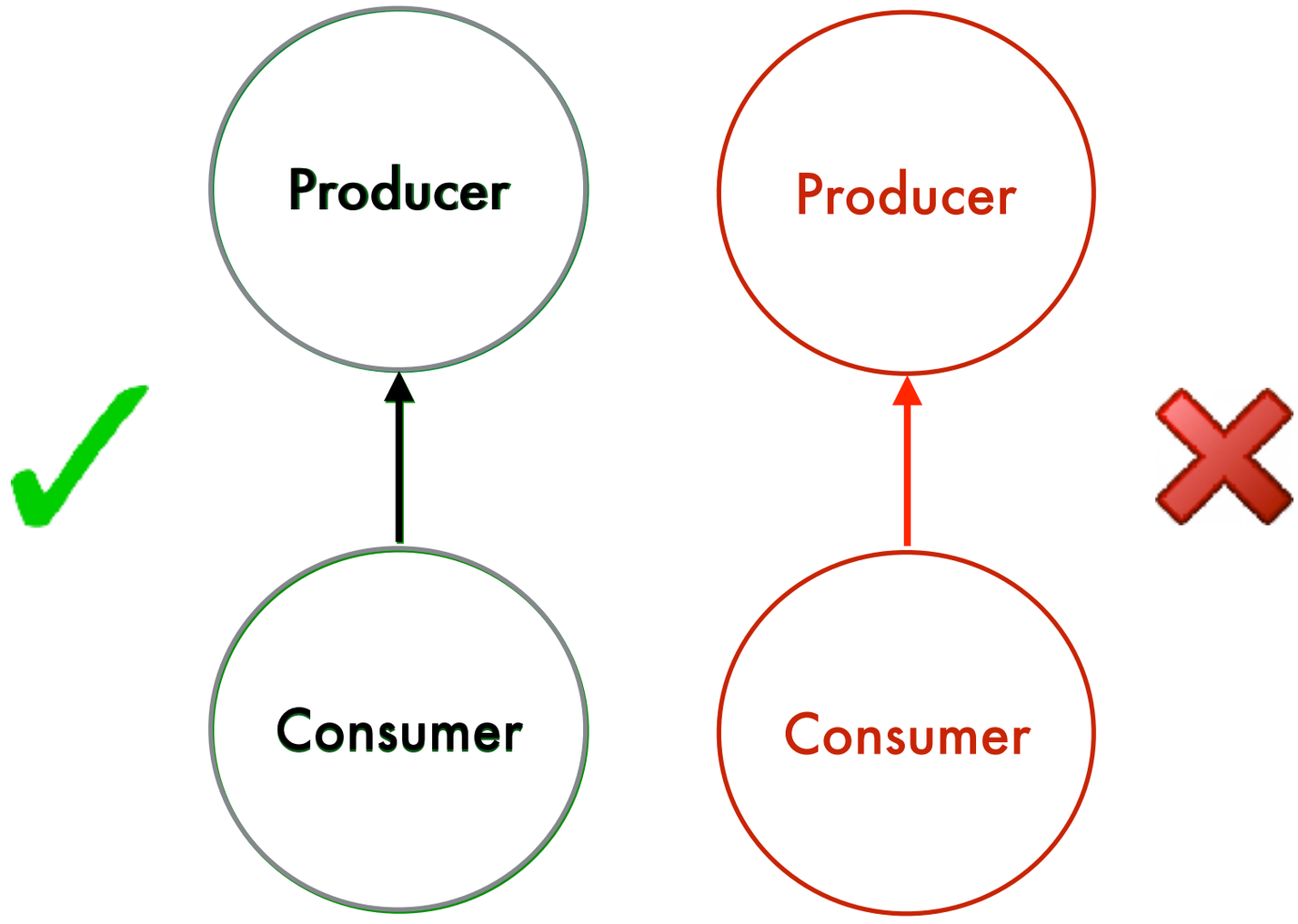
Rate Limiting

SpectoLabs



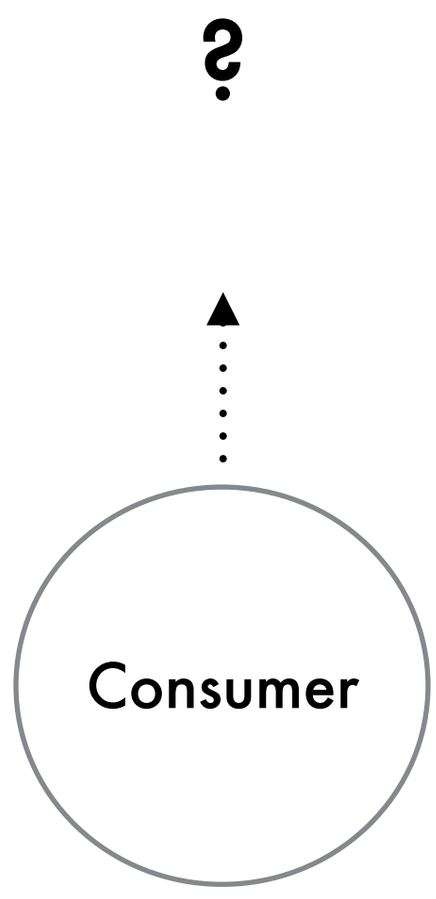
Flakey Dependencies

SpectoLabs



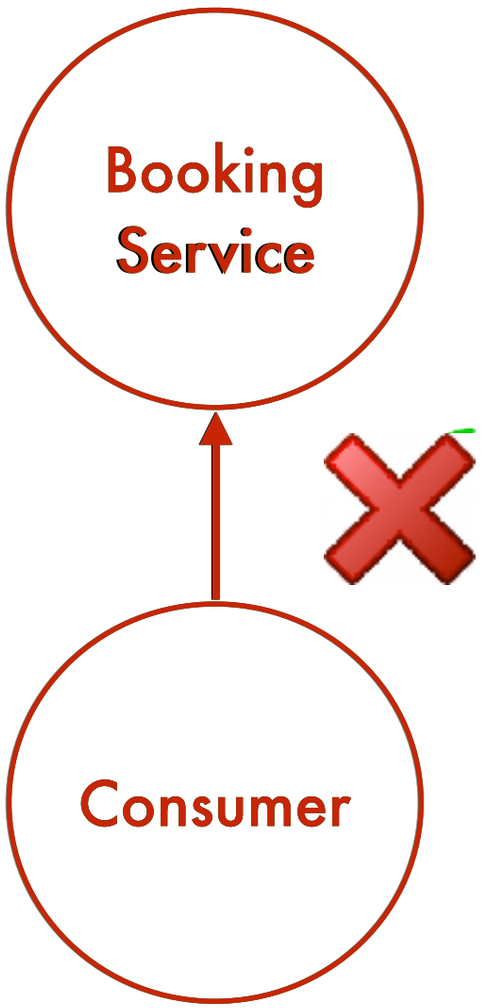
Doesn't exist

SpectoLabs



Non Deterministic

SpectoLabs

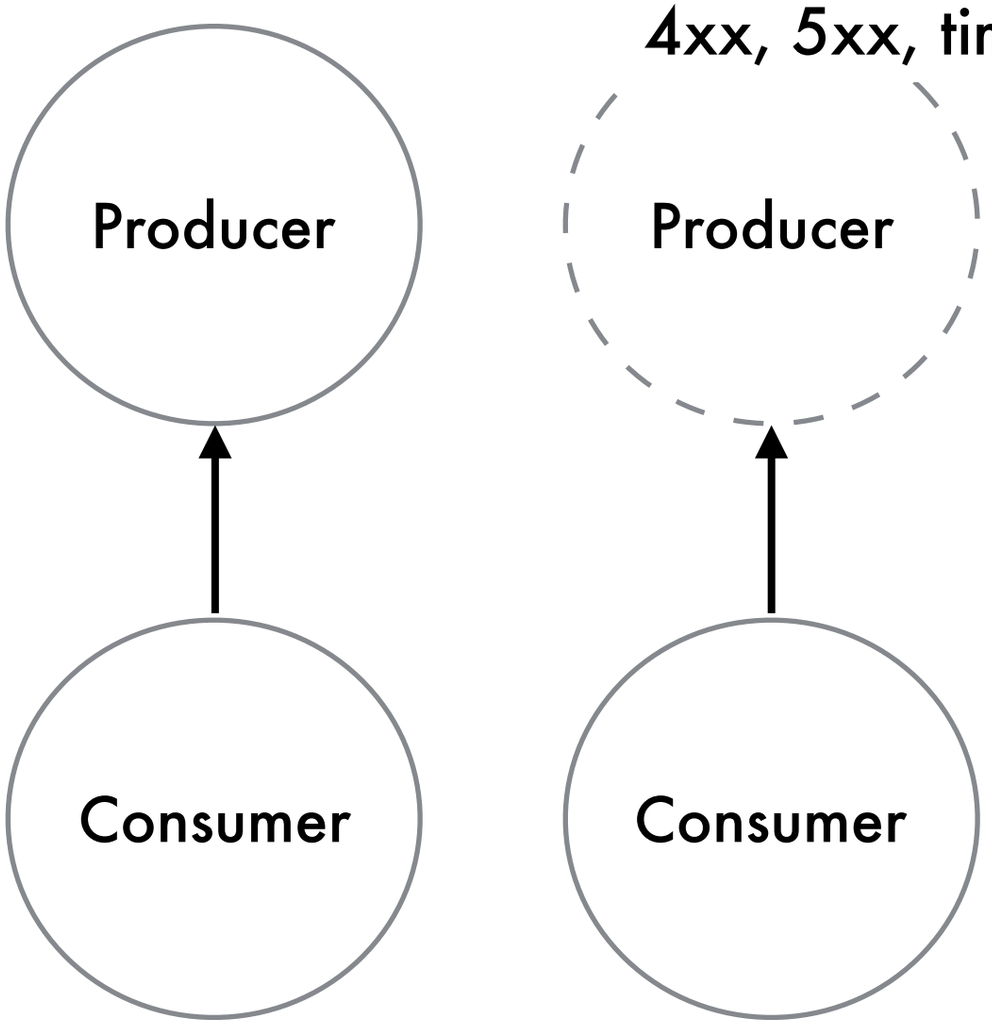



```
{
    “date” : “2013-01-08”,
    …
}
```

```
assertThat(booking.getDate())
.isEqualTo(LocalDate.of(2013,1,8));
```

Fault Injection

SpectoLabs



Operationally challenging to trigger faults

Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

API Simulation

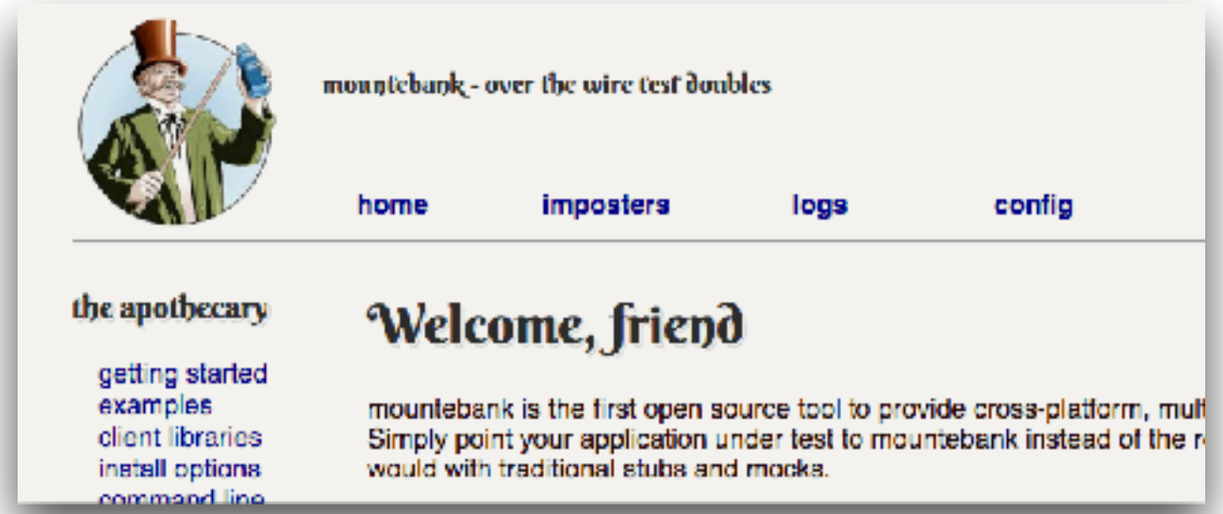
SpectoLabs

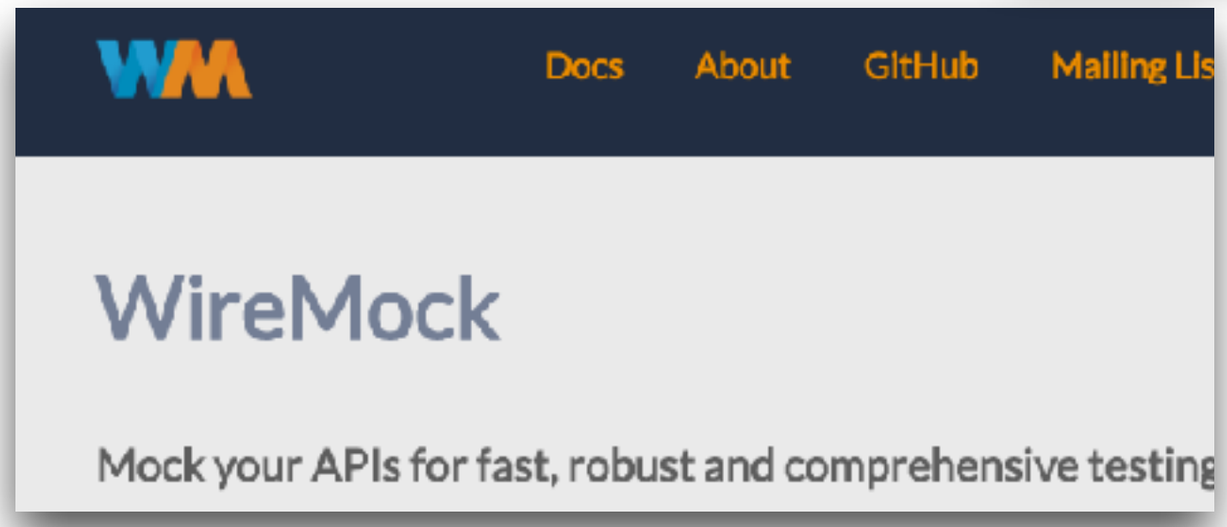
- Emulation of real API
- Non-intrusive
- Lightweight - or should be

Open Source Tooling

SpectoLabs



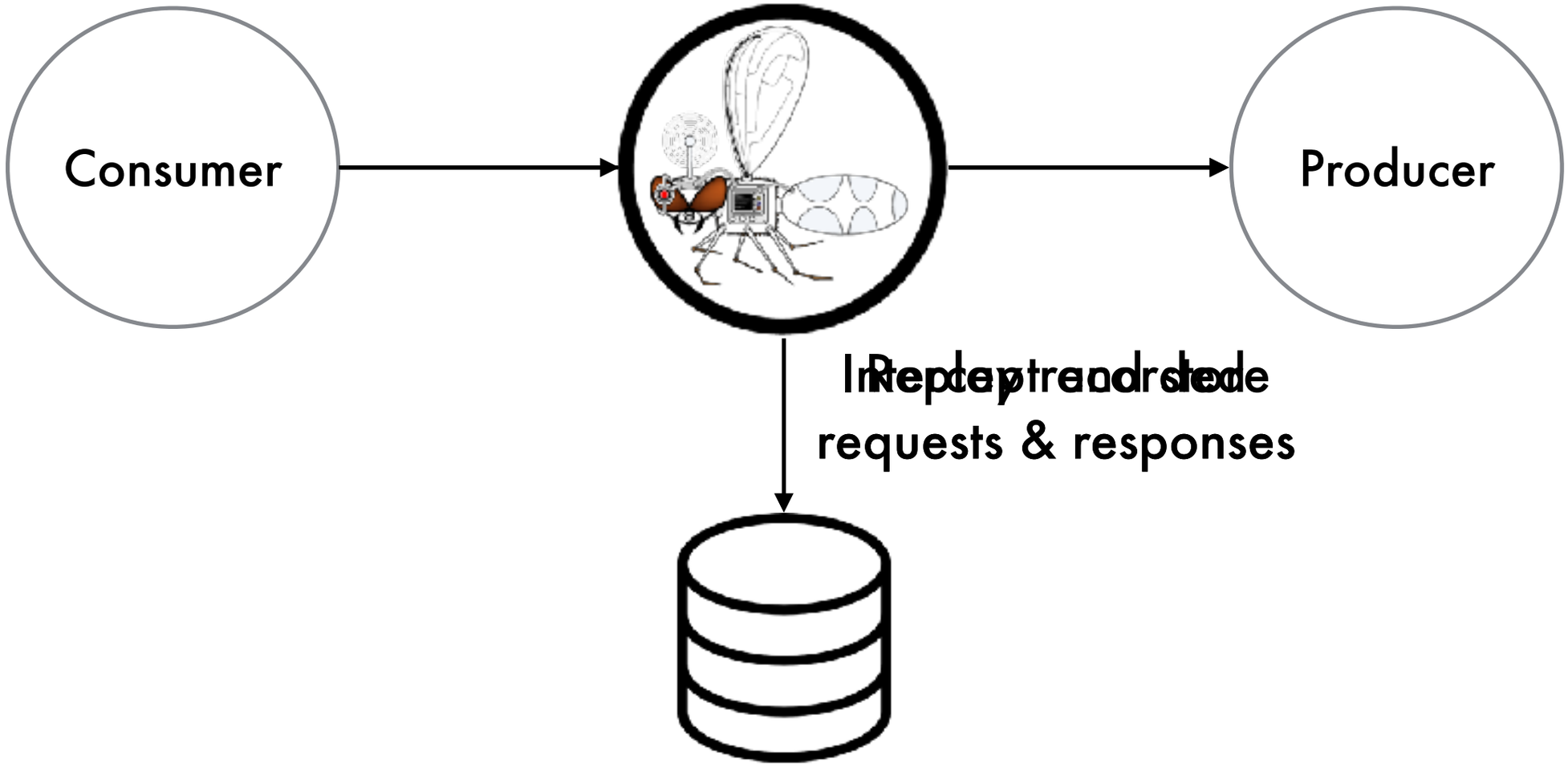




Architecture

SpectoLabs

Snapshot Mode



Creating Data

SpectoLabs

```
@ClassRule
public static HoverflyRule hoverflyRule = HoverflyRule.inSimulateMode(dsl(
    service("www.booking-service.com")
        .get("/api/bookings/1")
        .willReturn(success(json(new Booking(1))))
        .post("/api/bookings")
        .body(json(new Booking("myBooking")))
        .willReturn(created("http://www.booking-service.com/api/bookings/1")),
    service("www.payment-service.com")
        .get("/api/payments/1")
        .willReturn(success("{\"amount\": \"1.25\"\"}", "application/json"))
));
```

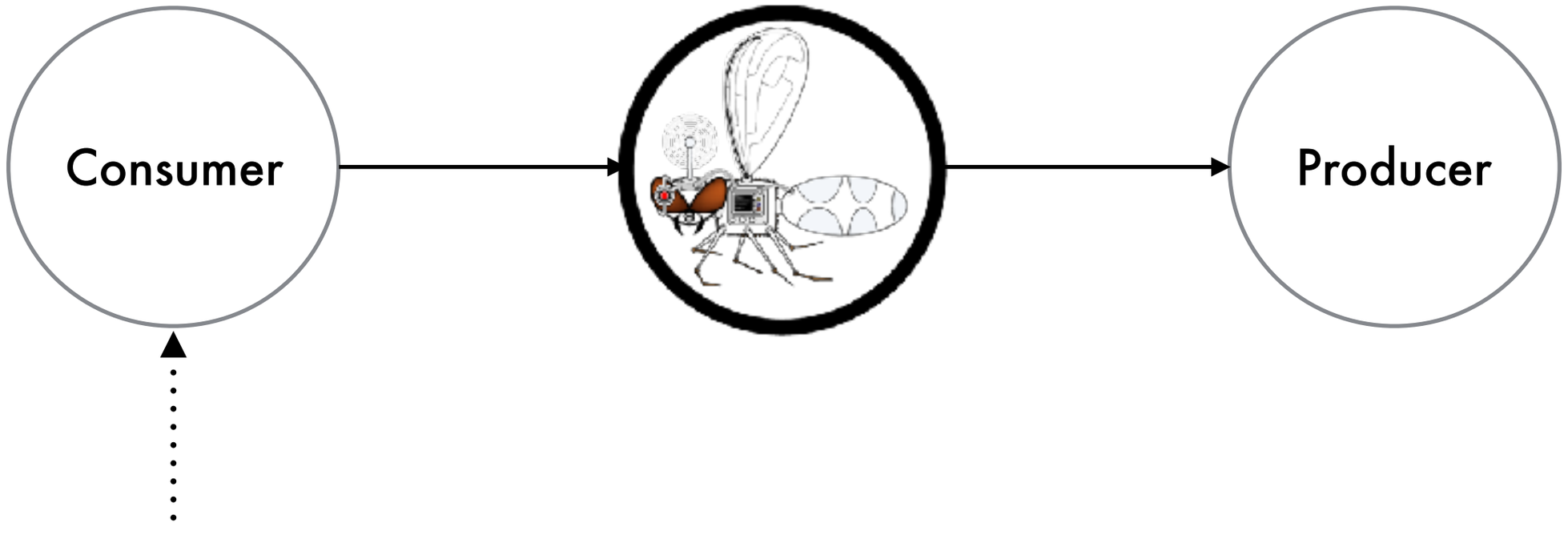
Creating Data

SpectoLabs


```
{
  "data": {
    "pairs": [
      {
        "response": {
          "status": 200,
          "body": "{\"stats\":{\"counters\":{\"capture\":0,\"modify\":0,\"simulate\
        },
        "request": {
          "requestType": "recording",
          "path": "/api/stats",
          "method": "GET",
          "destination": "localhost:60603",
          "scheme": "http",
          "query": "",
          "body": "",
          "headers": {
            "Accept": [
              "text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2"
            ],
            "Proxy-Connection": [
              "keep-alive"
            ],
            "User-Agent": [
              "Java/1.8.0_40"
            ]
          }
        }
```

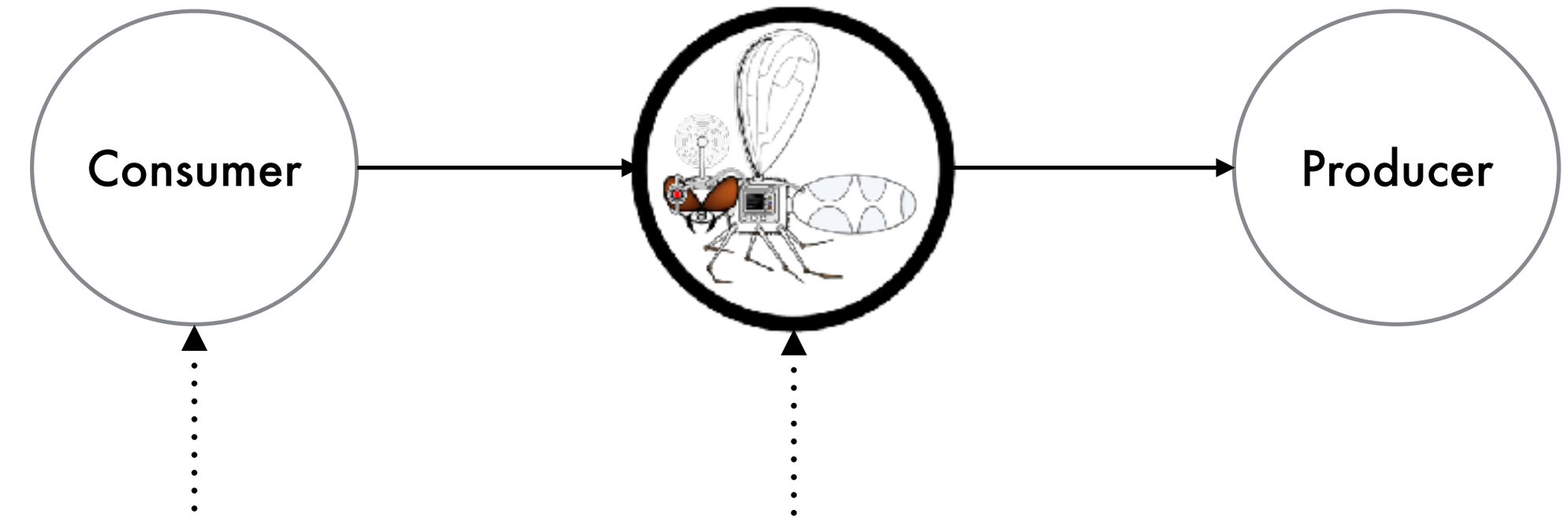
Forward Proxy

SpectoLabs



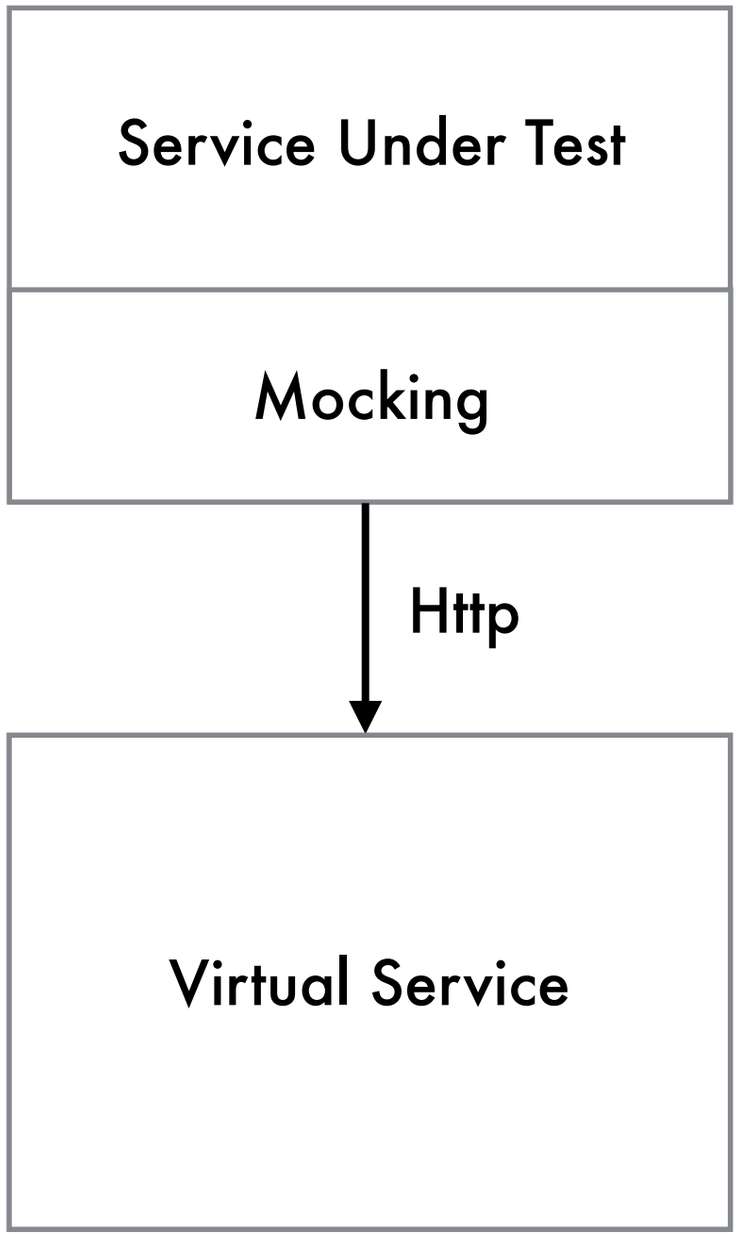
Reverse Proxy

SpectoLabs



Not Mocking

SpectoLabs



Mocks bleed into the architectural boundary of your application

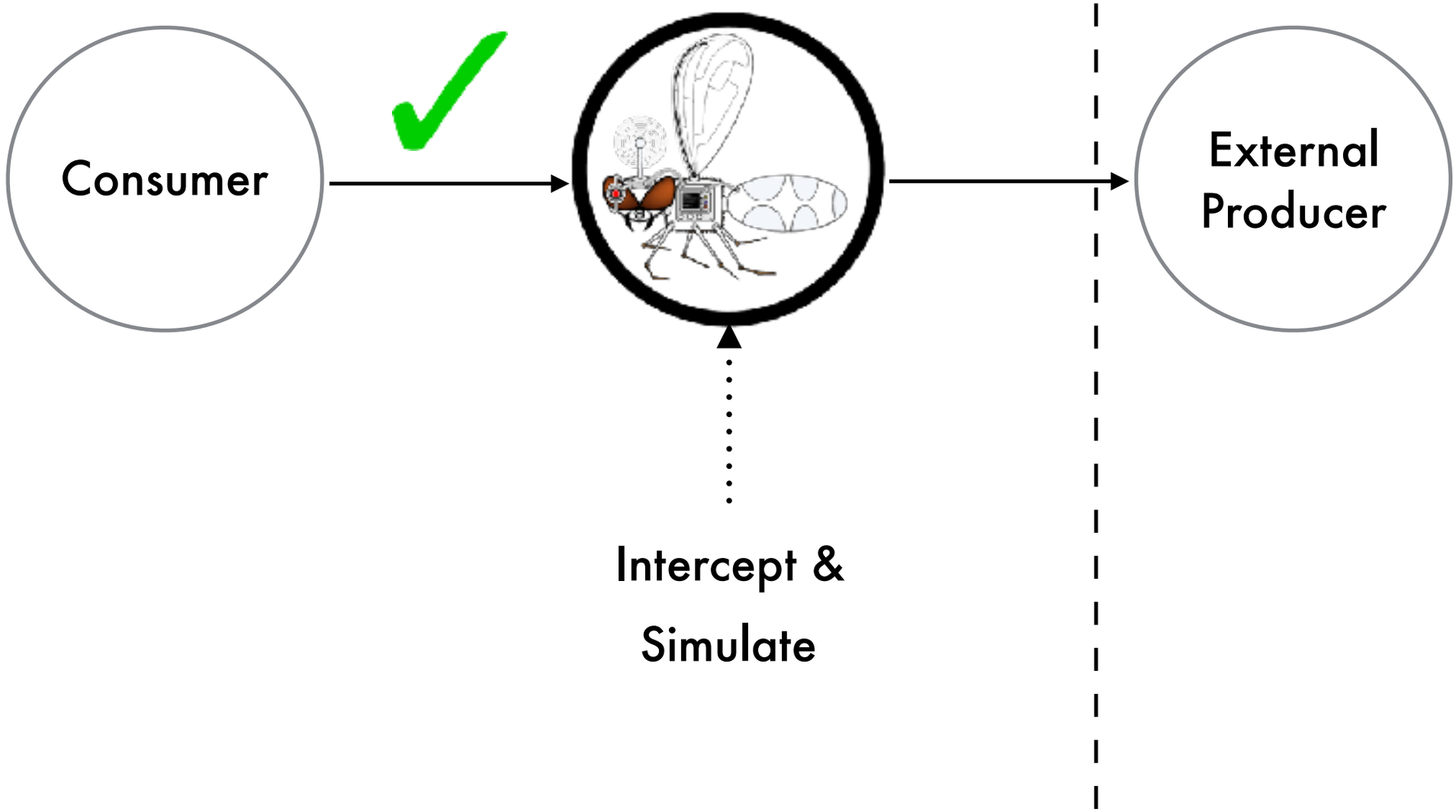
Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

Record and replay

SpectoLabs



JUnit Example

SpectoLabs

```
@Rule
public HoverflyRule hoverflyRule = HoverflyRule.inCaptureMode("src/test/
resources/external.json");

@Test
public void shouldBeAbleToRetrieveABooking() throws URISyntaxException {
    // Given
    final int bookingId = 1;

    // When
    final Booking booking = bookingServiceGateway.getBooking(bookingId);

    // Then
    assertThat(booking.getCustomer()).isEqualTo("Andrew Morgan");
    assertThat(booking.getDate()).isEqualTo(LocalDate.of(2016,3,1));
}
```

JUnit Example

SpectoLabs


```
{
  "data": [
    {
      "request": {
        "path": "/api/bookings/1",
        "method": "GET",
        "destination": "www.my-test.com",
        "query": null,
        "body": ""
      },
      "response": {
        "status": 200,
        "body": “{\”customer\”:\”Andrew Morgan\", \”date\”:
\”2016-03-01\”}}}”,
        "headers": {
          "Content-Type": [
            "application/json"
          ]
        }
      }
    }
  ]
}
```

JUnit Example

SpectoLabs

```
@Rule
public HoverflyRule hoverflyRule =
HoverflyRule.inSimulationMode(classpath("external.json"));

@Test
public void shouldBeAbleToRetrieveABooking() throws URISyntaxException {
    // Given
    final int bookingId = 1;

    // When
    final Booking booking = bookingServiceGateway.getBooking(bookingId);

    // Then
    assertThat(booking.getCustomer()).isEqualTo("Andrew Morgan");
    assertThat(booking.getDate()).isEqualTo(LocalDate.of(2016,3,1));
}
```

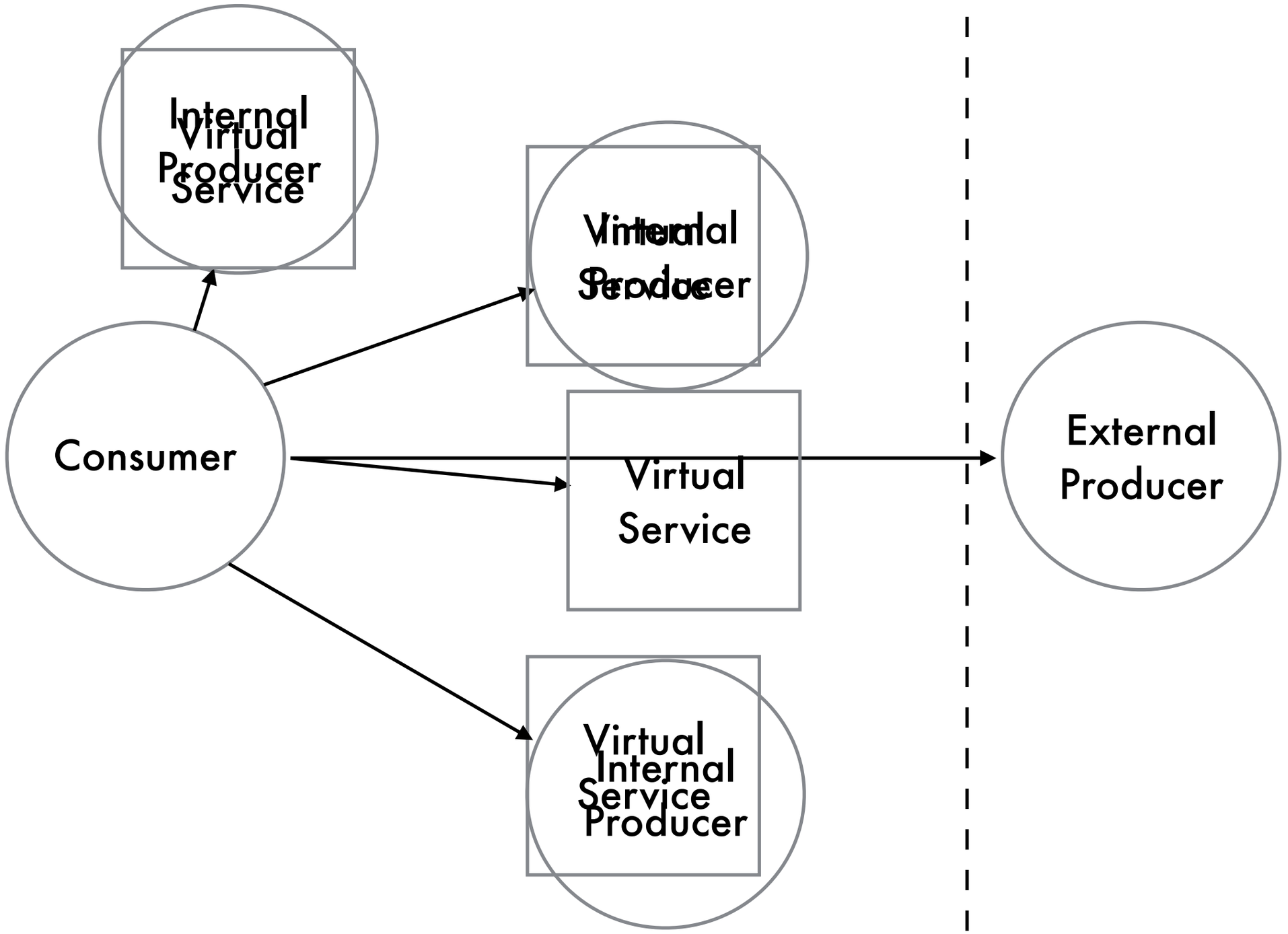
Advantages

SpectoLabs

- Isolation
- Flaky dependencies
- Rate limiting
- Deterministic
- Licensing

Operations

SpectoLabs



CLI Example

SpectoLabs

```
$ hoverctl start –proxy-port 8080
$ hoverctl import accounts-service-simulation.json

$ hoverctl start –proxy-port 8081
$ hoverctl import payments-service-simulation.json

$ curl http://www.account-service.com/api/v1/accounts/1 –proxy http://
localhost:8080

[{"name":"foo"}]

$ curl http://www.payment-service.com/api/v1/payments/1 –proxy http://
localhost:8081

[{“amount":"£23.27"}]
```



Matching

SpectoLabs

Matchers			
Method	Destination	Path	Query
GET	www.foo.com	/miss	?token=foo
GET	www.foo.com	/login	?token=foo
Request			
GET	www.foo.com	/login	?token=foo
✓	✓	✓	✓

Why Looser Matching?

SpectoLabs

Matcher			
Method	Destination	Path	Query
GET	www.foo.com	/login	?token=foo
Request			
GET	www.foo.com	/login	???
✓	✓	✓	✗

Field Omission

SpectoLabs

Matcher			
Method	Destination	Path	
GET	www.foo.com	/login	
Request			
GET	www.foo.com	/login	?token=bar
✓	✓	✓	✗

Regex Matching

SpectoLabs

Matcher		
Method	Destination	Path Regex
GET	www.foo.com	name\\/(.*)
Request		
GET	www.foo.com	/name/tim
✓	✓	✓

JSON Matching

SpectoLabs

Matcher
Body JSON
{“foo” : “bar”}
Request
{“foo” : “bar”}


JSON Matching

SpectoLabs

Matcher
Body JSON loose=true
{“foo” : “bar”, “not” : “important”}
Request
{“foo” : “bar”}


Additional Matchers

SpectoLabs

- XML (Strict and Loose)
- JSONPath / XPath
- ...Anything

Scoring

SpectoLabs

Request					
GET	www.foo.com	/name/tim	?token=bar		
Matchers				**Match?**	**Score**
GET	www.foo.com			✓	2
GET	www.foo.com	/name/tim		✓	3
GET	www.foo.com	/name/tim	?token=bar	✓	4

Debugging

SpectoLabs

Request						
GET	www.foo.com	/name/tim	?token=bar			
Matchers				**Score**	**Match?**	**Closest?**
MISS	www.foo.com	miss		1	✗	
MISS	www.foo.com	/name/tim		2	✗	
GET	www.foo.com	/name/tim	miss	3	✗	✓

Templating

SpectoLabs

Path Matcher	Response Body
/name\\/(.*)	{“name”:”{{ Request.Path.[1] }}”}

Request Path	Response Body
/name/john	{“name”:”john”}
/name/peter	{“name”:”peter”}

The Problem of State

SpectoLabs

Request			Response
GET	www.shopping.com/basket		{}
POST	www.shopping.com/basket	{"item":"eggs"}	200
GET	www.shopping.com/basket		{}

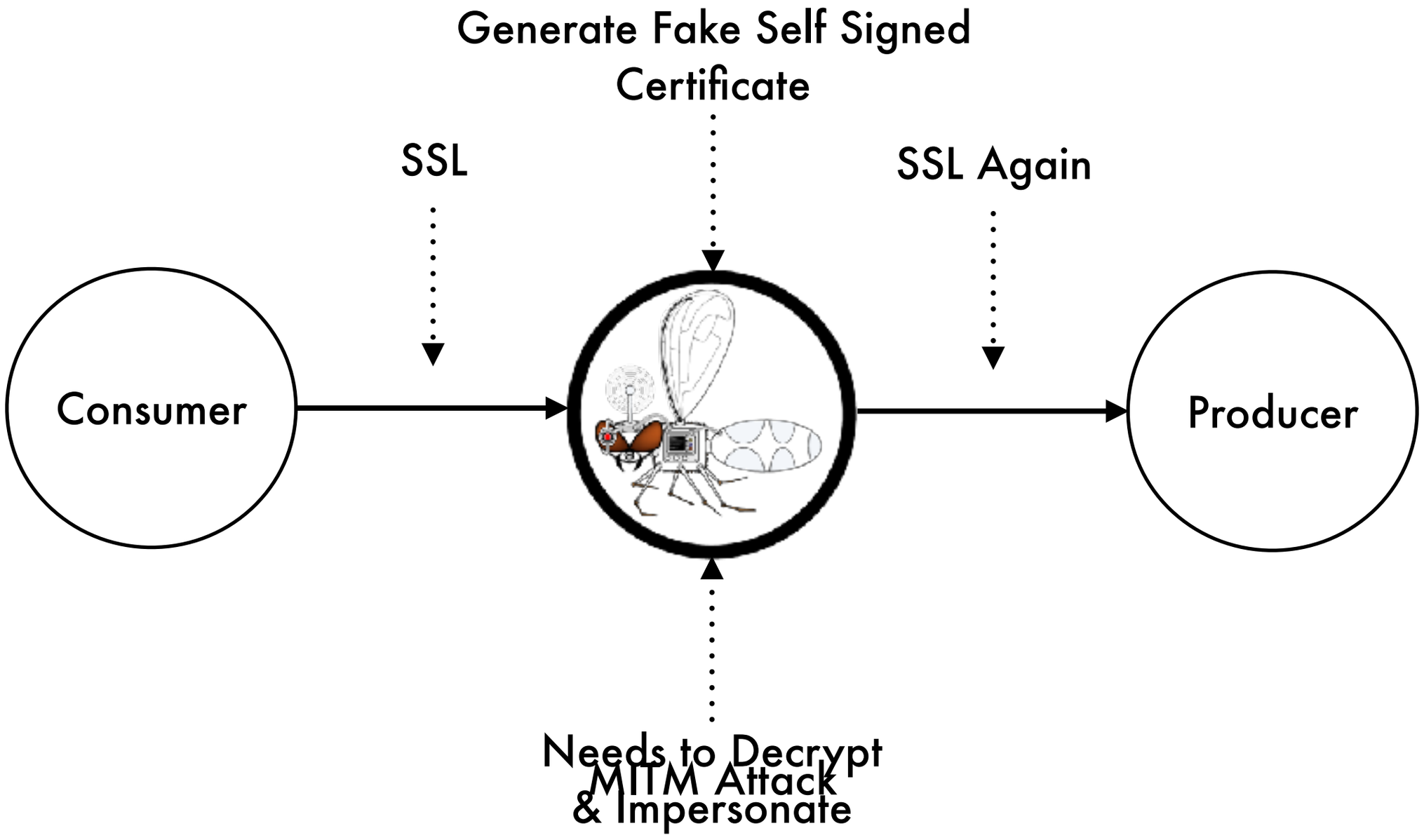
State

SpectoLabs

Request			Response	Requires State	Transitions State
GET	www.shopping.com/basket		{}	basket == empty	
POST	www.shopping.com/basket	{"item":"eggs"}	200		basket = eggs
GET	www.shopping.com/basket		{"item":"eggs"}	basket == eggs	

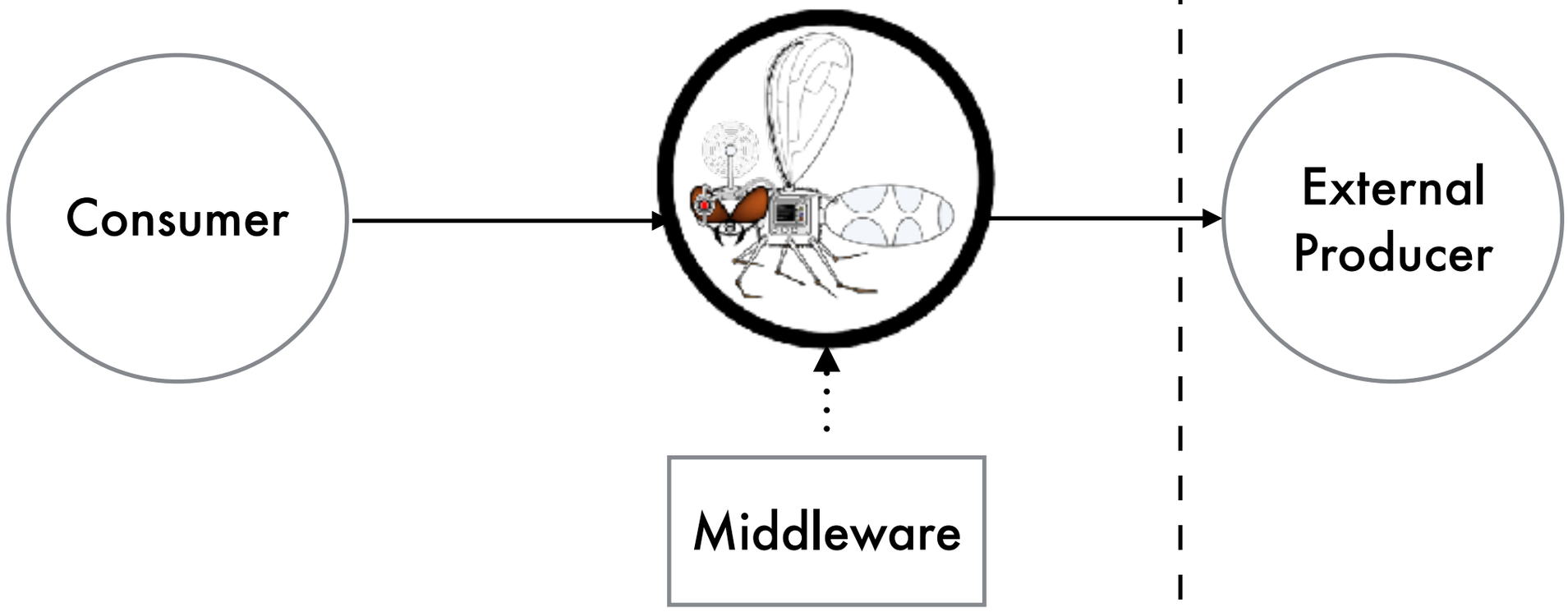
SSL Bump

SpectoLabs



Fault Injection

SpectoLabs



Fault Injection

SpectoLabs

```
def main():
    data = sys.stdin.readlines()
    # this is a json string in one line so we are interested in that one line
    payload = data[0]
    logging.debug(payload)


    payload_dict = json.loads(payload)


    payload_dict['response']['status'] = 201
    payload_dict['response']['body'] = "body was replaced by middleware"

    # now let' sleep for 2 seconds
    sleep(2)

    # returning new payload
    print(json.dumps(payload_dict))
```

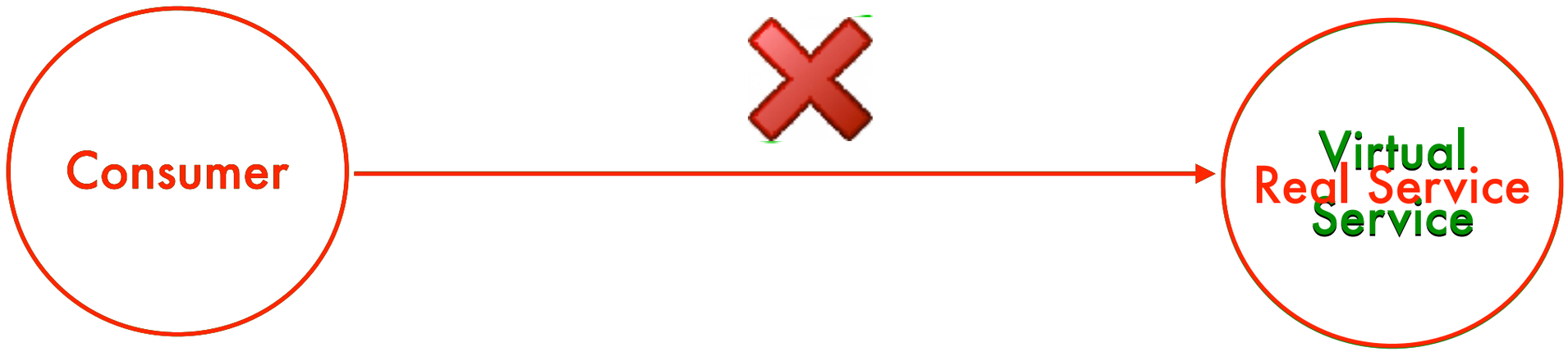
Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

Stale Data

SpectoLabs



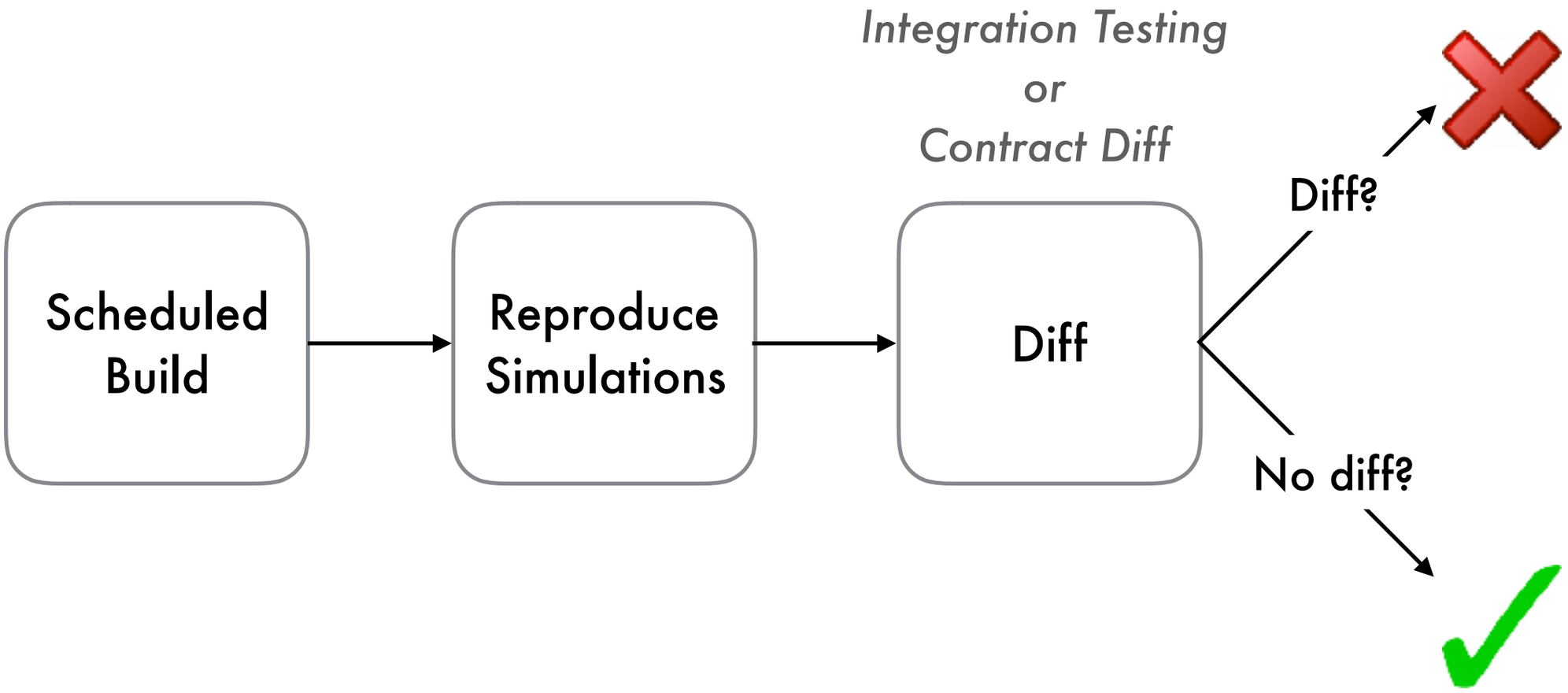
Ownership

SpectoLabs

Own Consumer	Own Producer	Strategy
✓	✗	Integration Contract Testing
✗	✓	
✓	✓	

Integration Contract Testing

SpectoLabs



http://martinfowler.com/bliki/IntegrationContractTest.html

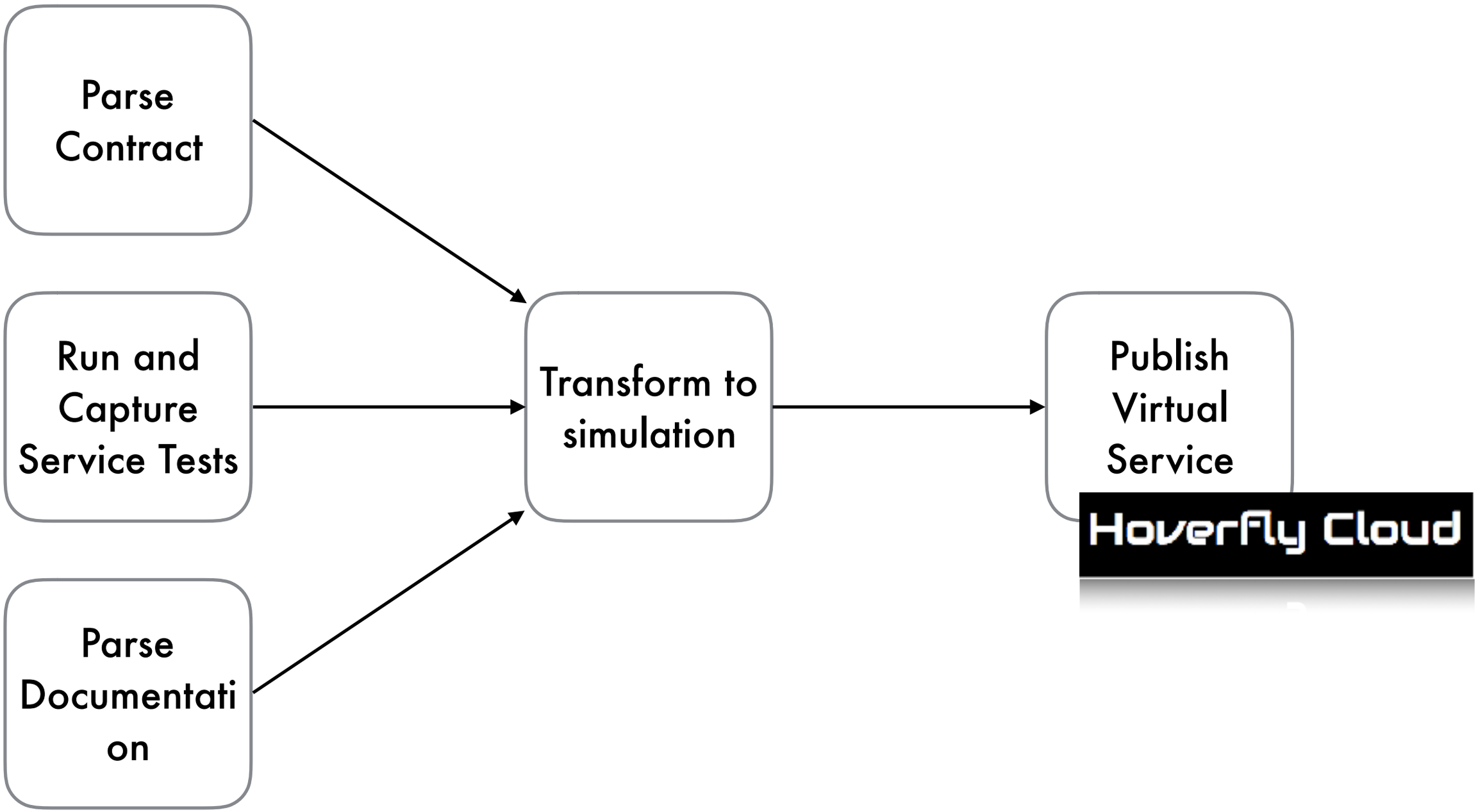
Ownership

SpectoLabs

Own Consumer	Own Producer	Strategy
✓	✗	Integration Contract Testing
✗	✓	Simulation as First Class Citizen
✓	✓	

First Class Simulations

SpectoLabs



Ownership

SpectoLabs

Own Consumer	Own Producer	Strategy
✓	✗	Integration Contract Testing
✗	✓	Virtual Service as First Class Citizen
✓	✓	Consumer Driven Contract Testing

Consumer Driven Contracts

SpectoLabs

Consumer Defines Contract

Generate Producer Stubs

Implement Consumer



Generate Producer Contract Tests

Implement Producer

Run Consumer Contract Tests



http://www.martinfowler.com/articles/consumerDrivenContracts.html

Consumer Driven Contracts

SpectoLabs

Change producer

Run Consumer Contract Tests

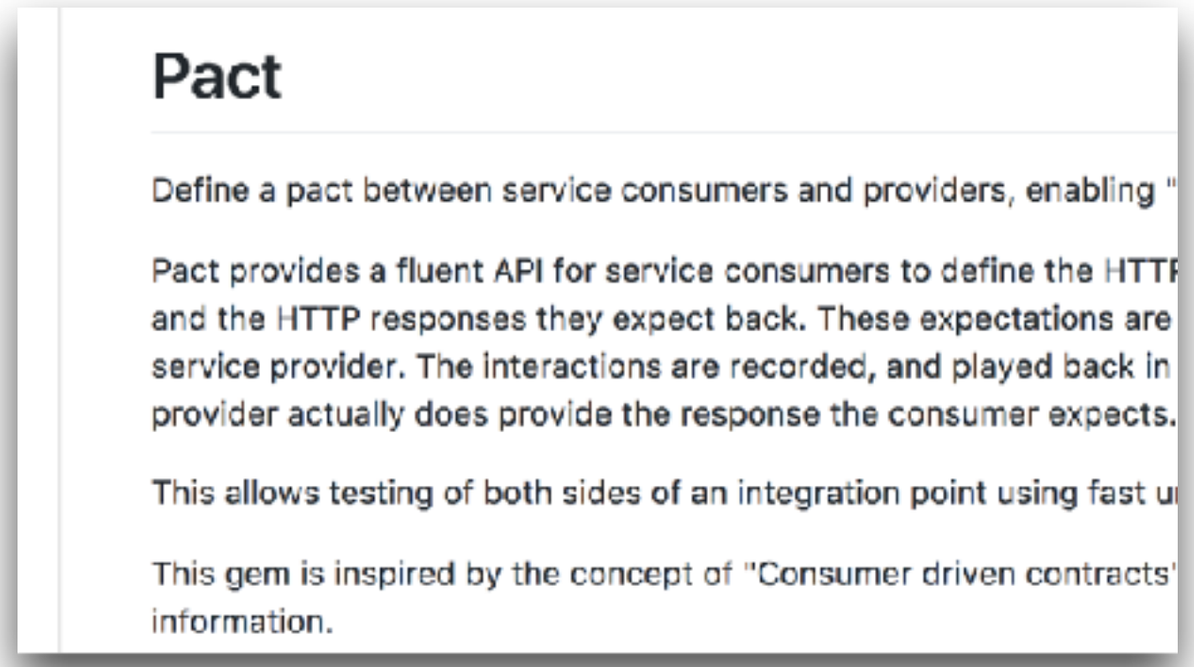


http://www.martinfowler.com/articles/consumerDrivenContracts.html

Tools

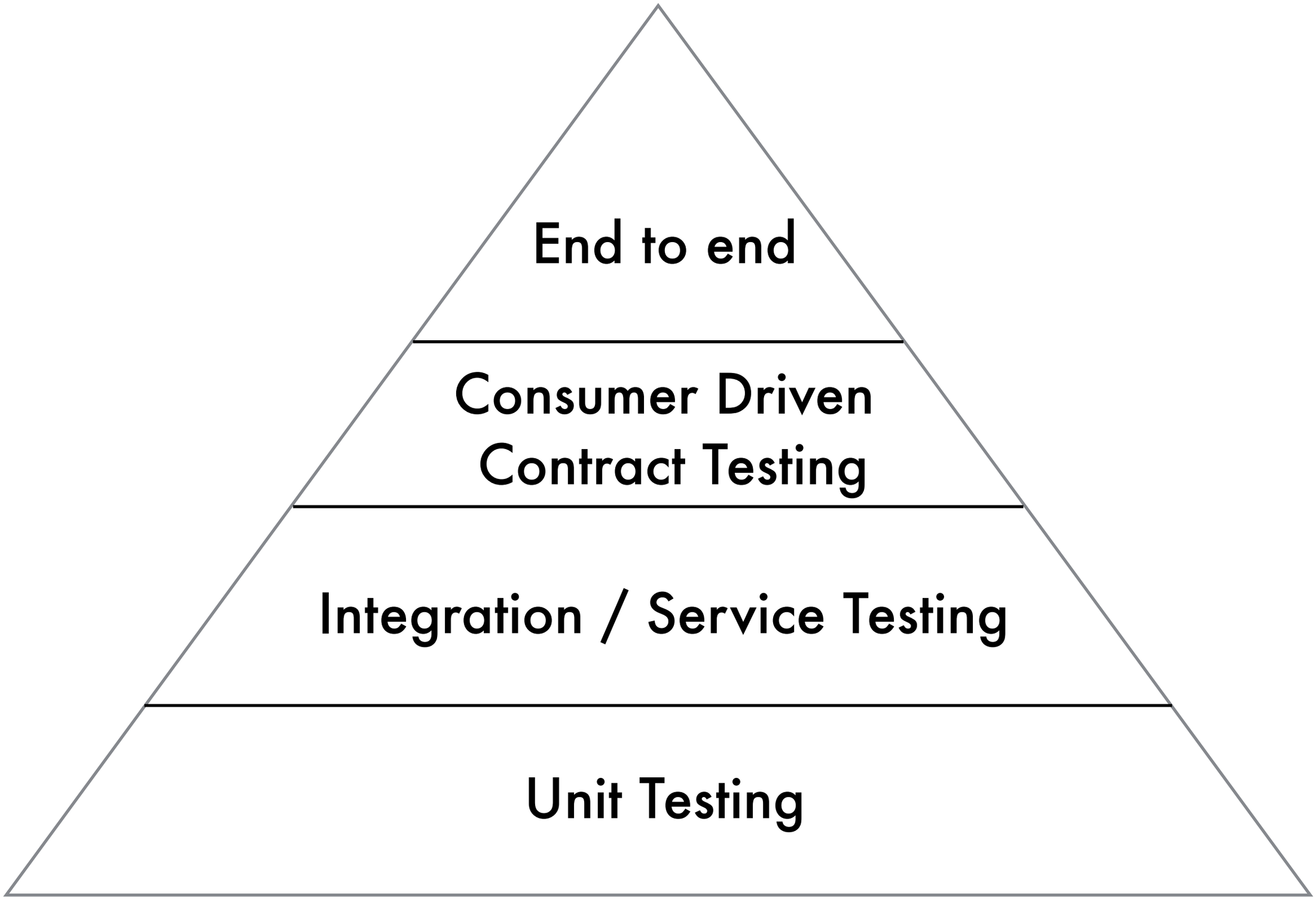
SpectoLabs





Testing Pyramid

SpectoLabs



Agenda

SpectoLabs

- The problems with testing API driven architectures
- Introduction to API simulation
- Applying API simulation patterns
- Data management
- Conclusion

Conclusion

SpectoLabs

- Reduced cost be replacing real resources
- Improved deployment time with API simulations
- Less dependance on end-to-end testing
- Advanced patterns for simplifying API simulations
- Improved feedback cycle

Questions?

@mogronalol | @danielbryantuk
andrew.morgan@specto.io | daniel.bryant@specto.io
https://github.com/mogronalol
hoverfly.io