Day 1: Hello, World! mountebank

The Problem with End-to-End Testing

- Time
- Cost
- Slow feedback loop

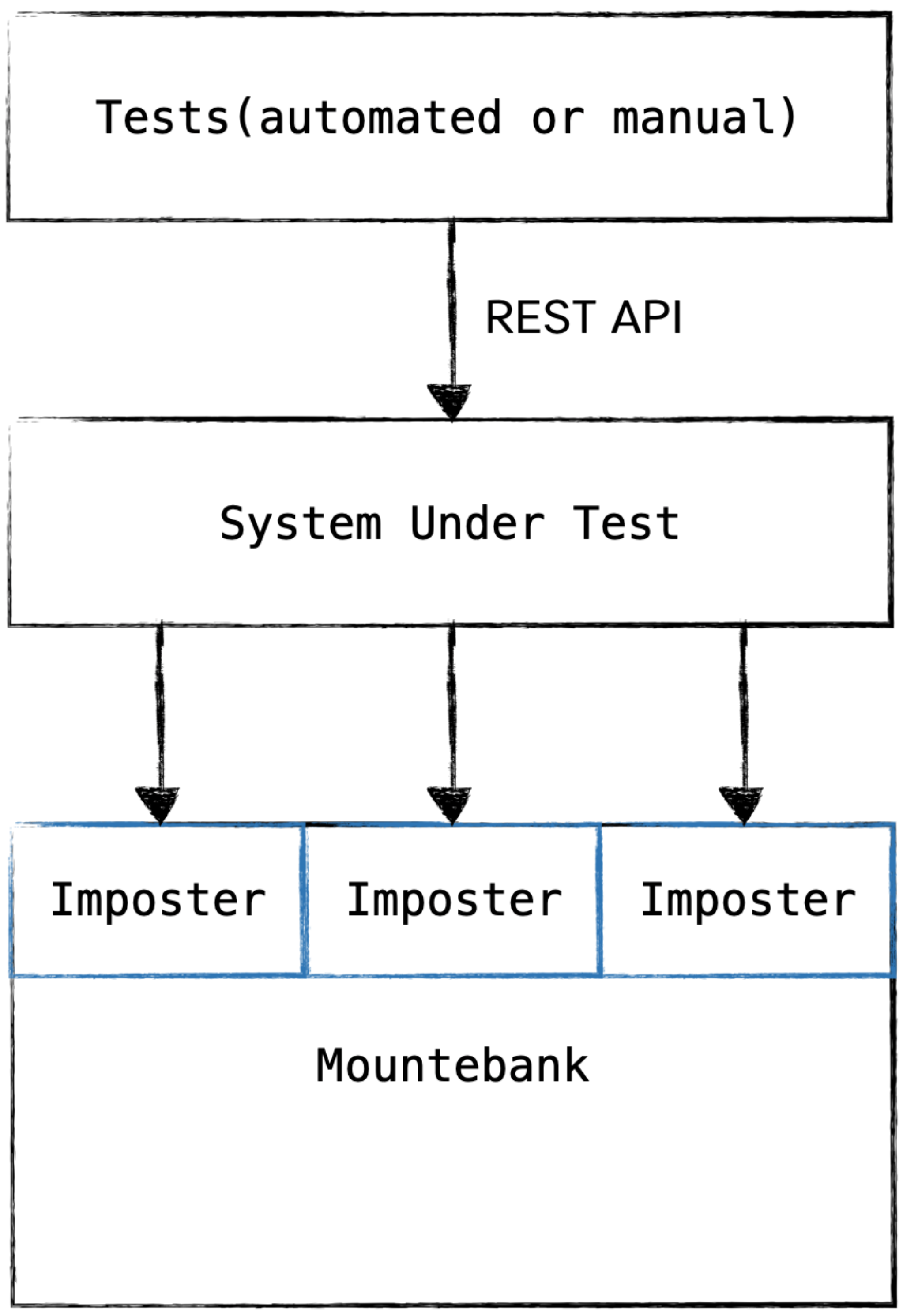
Setting up

3A pattern: Arrange, Act, Assert, Cleanup

- Setup data test
- Execute the system under test
- Assert the expected response
- Clean the state

Imposters: virtual services, a lightweight operation

- Mountebank provides configure virtual services, which are called imposters.
- Each imposter represents a socket that acts as the virtual service and accepts connections from the real service you are testing.
- Spinning up and shutting down imposters is a lightweight operation.



a simple Mountebank imposter

> ref: Testing Microservice with Mountebank page 17

How mountebank views an HTTP request

> ref: Testing Microservice with Mountebank page 26

```
POST /products?page=2&itemsPerPage=2 HTTP/1.1
Host: api.petstore.com
Content-Type: application/json

{
  "key": "asdul7890"
}
```

```
{
  "method": "POST",
  "path": "/products",
  "query": {
    "page": 2,
    "itemsPrePage": 2
  },
  "headers": {
    "Host": "api.petstore.com",
    "Content-Type": "application/json"
  },
  "body": "{\n  \"key\": \"asdul7890\"\n}"
}
```

How mountebank views an HTTP response

> ref: Testing Microservice with Mountebank page 27

```
HTTP/1.1 200 OK
Date: Sun, 05 Apr 2020 10:10:10 GMT
Content-Type: application/json

{
  "key": "asdul7890"
}
```

```
{
  "statusCode": 200,
  "headers": {
    "Date": "Sun, 05 Apr 2020 10:10:10 GMT",
    "Content-Type": "application/json"
  },
  "body": "{\n  \"key\": \"asdul7890\"\n}"
}
```

The HTTP response structure in JSON

```
{
  "statusCode": 200,
  "headers": { "Content-Type": "text/plain" },
  "body": "Hello, World!"
}
```

Hello, World Mountebank

Run: [Command line]

```
mb
```

> open web browser then go to http://localhost:2525

List imposters: [Command line, Web browser]

> curl

```
curl http://localhost:2525/imposters
```

> goto http://localhost:2525/imposters

Create First Imposters: [Command line]

> curl

```
curl -X POST http://localhost:2525/imposters --data '
{
  "protocol": "http",
  "port": 3000,
  "stubs": [
    {
      "responses": [
        {
          "is": {
            "statusCode": 200,
            "headers": { "Content-Type": "text/plain" },
            "body": "Hello, World!"
          }
        }
      ]
    }
  ]
}'
```

Delete: [Command line]

> curl

```
curl -X DELETE http://localhost:2525/imposters/3000
```

> response

```
info: [mb:2525] DELETE /imposters/3000
info: [http:3000] Ciao for now
```

> list imposters

```
curl http://localhost:2525/imposters
```

Delete all impostes

> curl

```
curl -X DELETE http://localhost:2525/imposters
```

Create First Imposters from Config File

> create file call hello-world.json

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [
    {
      "responses": [
        {
          "is": {
            "statusCode": 200,
            "headers": { "Content-Type": "text/plain" },
            "body": "Hello, World!"
          }
        }
      ]
    }
  ]
}
```

```
mb start --configfile hello-world.json
```

The Basic of Mountebank HTTP Responses

The ***is*** response type, which is the fundamental building block for a stub.

The HTTP response structure in JSON

```
{
  "statusCode": 200,
  "headers": { "Content-Type": "text/plain" },
  "body": "Hello, World!"
}
```

Default Response

```
{
  "protocol": "http",
  "port": 3001,
  "name": "Default Response",
  "defaultResponse": {
    "statusCode": 400,
    "headers": {
      "Connection": "Keep-Alive",
      "Content-Length": 0
    }
  },
  "stubs": [
    {
      "responses": [
        {
          "is": { "body": "BOOM!!!" }
        }
      ]
    }
  ]
}
```

Cycling through response

```
{
  "protocol": "http",
  "port": 3002,
  "stubs": [
    {
      "responses": [
        {
          "is": { "body": "1" }
        },
        {
          "is": { "body": "2" }
        },
        {
          "is": { "body": "3" }
        }
      ]
    }
  ]
}
```

Saving multiple Imposters in the config file

```
{
  "imposters": [
    {
      "protocol": "http",
      "port": 3000,
      "stubs": [{
          "responses": [{
              "is": {
                "statusCode": 200,
                "headers": { "Content-Type": "text/plain" },
                "body": "Hello, World!"
              }
          }]
      }]
    },
    {
      "protocol": "http",
      "port": 3001,
      "name": "Default Response",
      "defaultResponse": {
        "statusCode": 400,
        "headers": {
          "Connection": "Keep-Alive",
          "Content-Length": 0
        }
      },
      "stubs": [{
          "responses": [{
              "is": { "body": "BOOM!!!" }
          }]
      }]
    },
    {
      "protocol": "http",
      "port": 3002,
      "stubs": [{
          "responses": [
            { "is": { "body": "1" }},
            { "is": { "body": "2" }},
            { "is": { "body": "3" }}
          ]
      }]
    }
  ]
}
```

```
imposter.ejs
hello-world.json
default-response.json
cycling-through-response.json
```

```
{
  "imposters": [
    <%- include hello-world.json %>,
    <%- include default-response.json %>,
    <%- include cycling-through-response.json %>,
  ]
}
```

EJS: Embedded JavaScript

> Mountebank uses a templating language called EJS (https://ejs.co/)

Imposter Structure

```
{
  "port": <port>,
  "protocol": "http",
  "stubs": [
    {
      "predicates": [],
      "responses": [
        {
          "is": {
            "statusCode": <statusCode>,
            "headers": {},
            "body": {}
          }
        }
      ]
    }
  ]
}
```

Basic command and option

```
mb start --configfile hello-world.json --pidfile &
```

```
mb stop
```

```
mb restart  --configfile hello-world.json --pidfile &
```