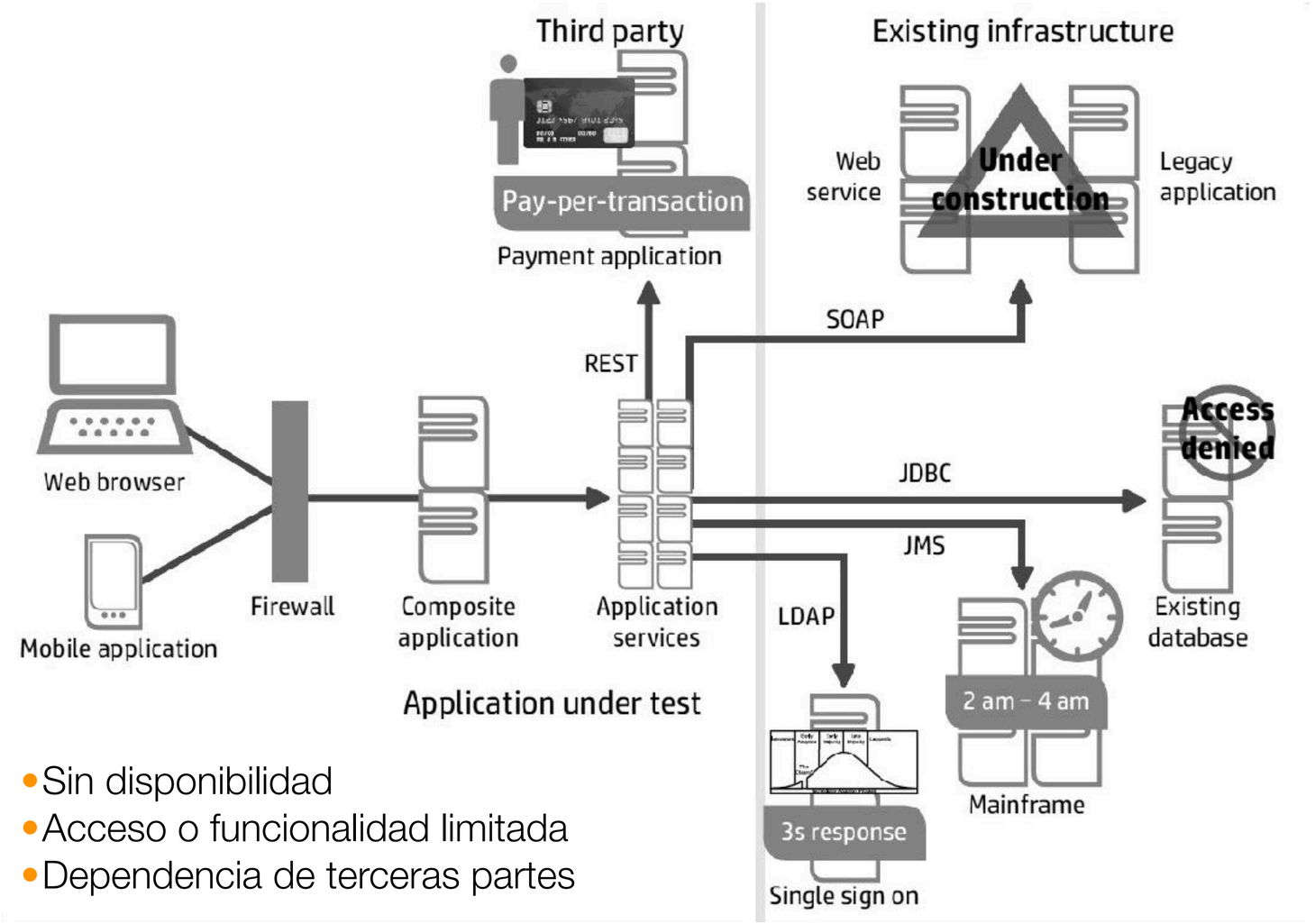
mountebank

Virtualización de Servicios - Testing APIs

Jul. 2019

wetek.io

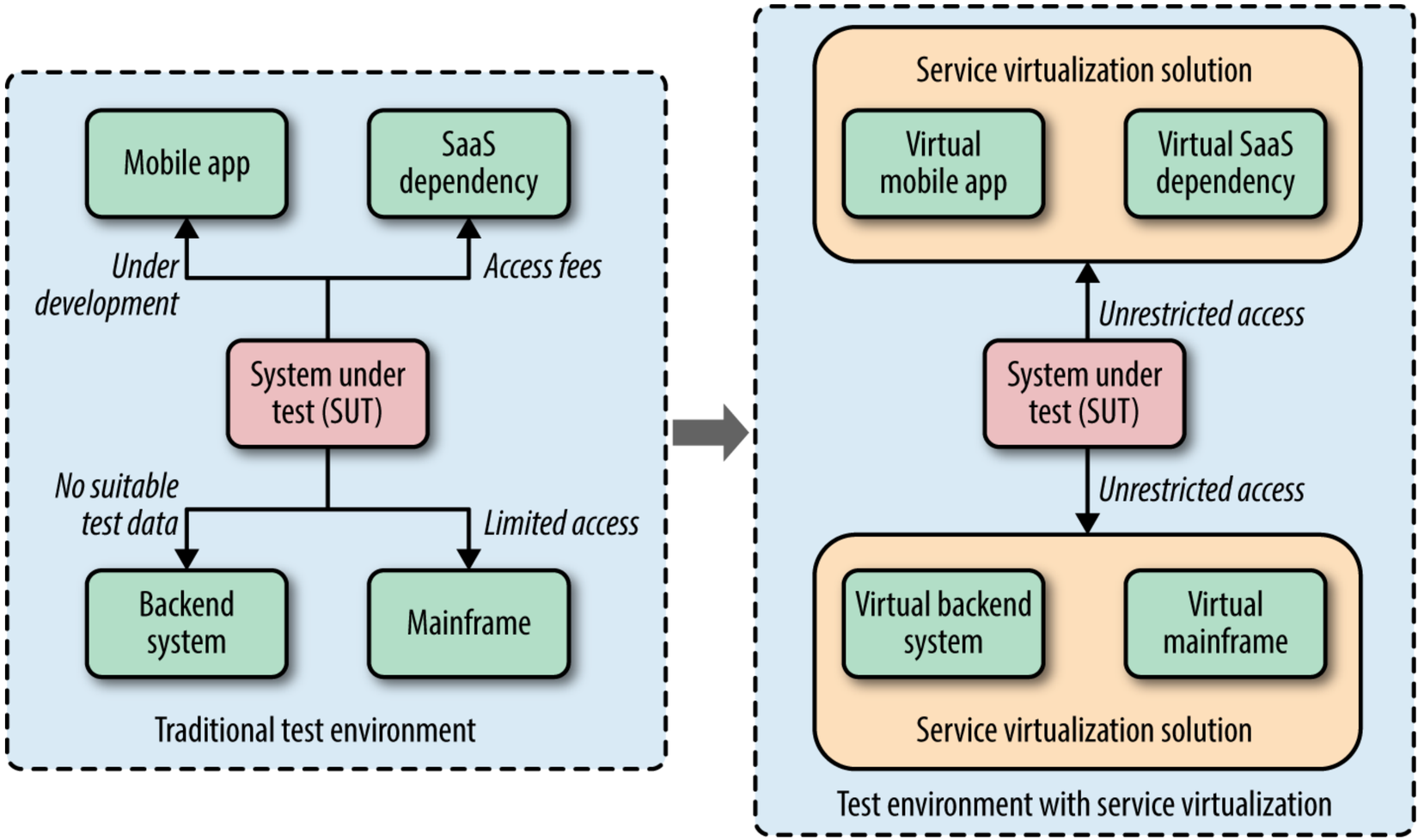
Testing de Servicios y APIs, problemática a resolver



- Sin disponibilidad
- Acceso o funcionalidad limitada
- Dependencia de terceras partes

Créditos: https://www.northwaysolutions.com/hp-service-virtualization-test-earlier-and-improve-time-to-market-ttm/

Virtualización de servicios, qué es?



Fuente: https://en.wikipedia.org/wiki/Service_virtualization
Ebook: https://www.oreilly.com/webops-perf/free/service-virtualization.csp

Virtualización de Servicios/APIs vs. Mocking

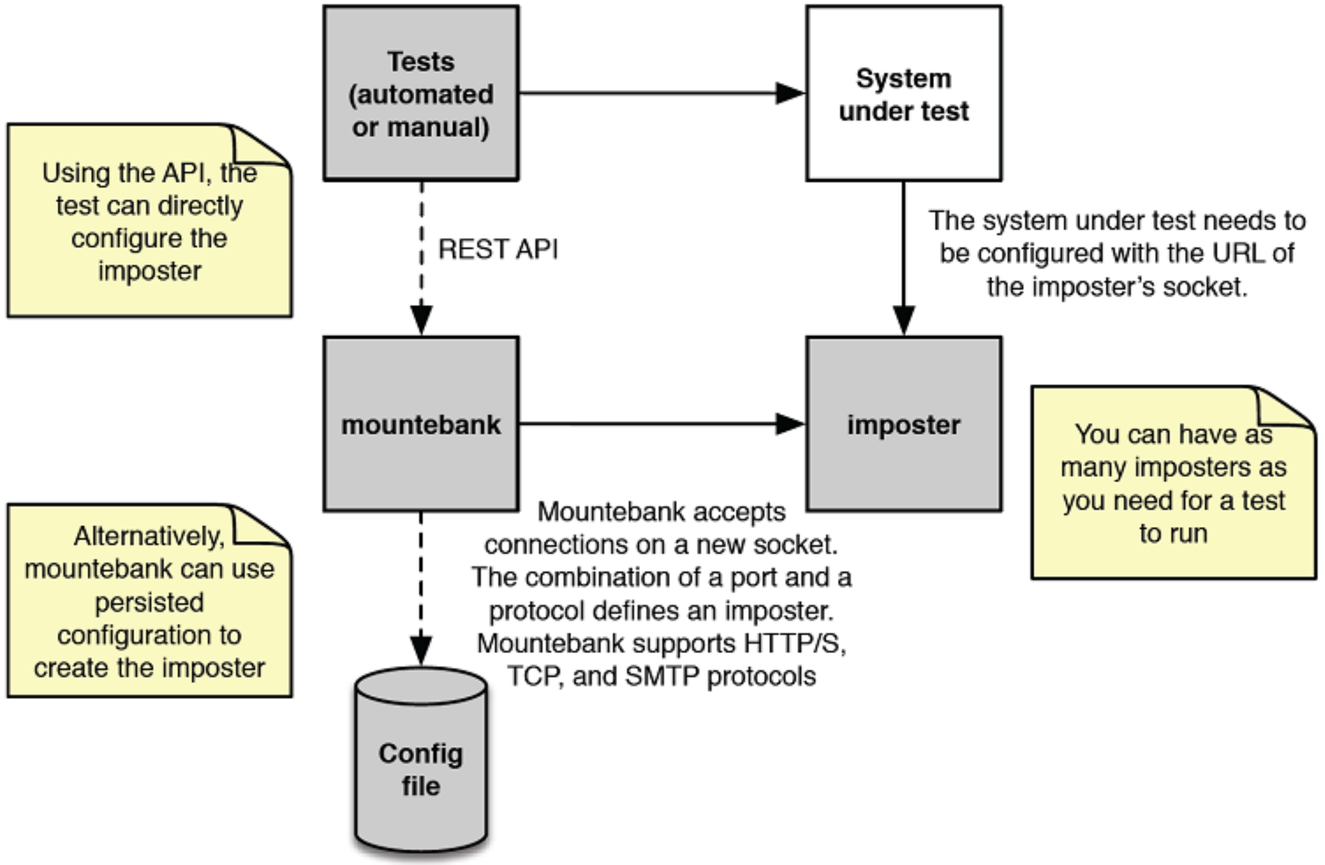
Stubbin o Mocking de Servicios / APIs	Virtualización de Servicios / APIs
• Se utiliza principalmente para soportar pruebas unitarias y de integración.	• Se utiliza principalmente para soportar las pruebas de sistemas, aceptación y rendimiento, aunque también se puede usar para pruebas unitarias y de integración.
• Creado y usado principalmente por desarrolladores	• Puede ser creado por cualquier persona autorizada y luego compartirse y usarse dentro del equipo o entre equipos.
• Se utiliza principalmente dentro de los límites de un solo equipo o proyecto	• Se puede utilizar a nivel de equipo individual, en todos los equipos dentro del mismo proyecto y en toda la organización; por ejemplo, a través de un centro de excelencia de virtualización de servicios dedicado
• No escala bien para soportar proyectos y escenarios de prueba más grandes	• Solución escalable y confiable, capaz de soportar grandes proyectos y escenarios de prueba.
• Sin soporte (o necesidad) para stubbing o mocking basada en datos	• Admite la creación de activos virtuales flexibles y basados en datos.
• Sin soporte (o necesidad) para una amplia gama de protocolos de mensajes y transporte	• Admite una amplia gama de protocolos de mensajes y transporte.

Fuente: https://www.oreilly.com/webops-perf/free/service-virtualization.csp

Mountebank, ¿Qué es?

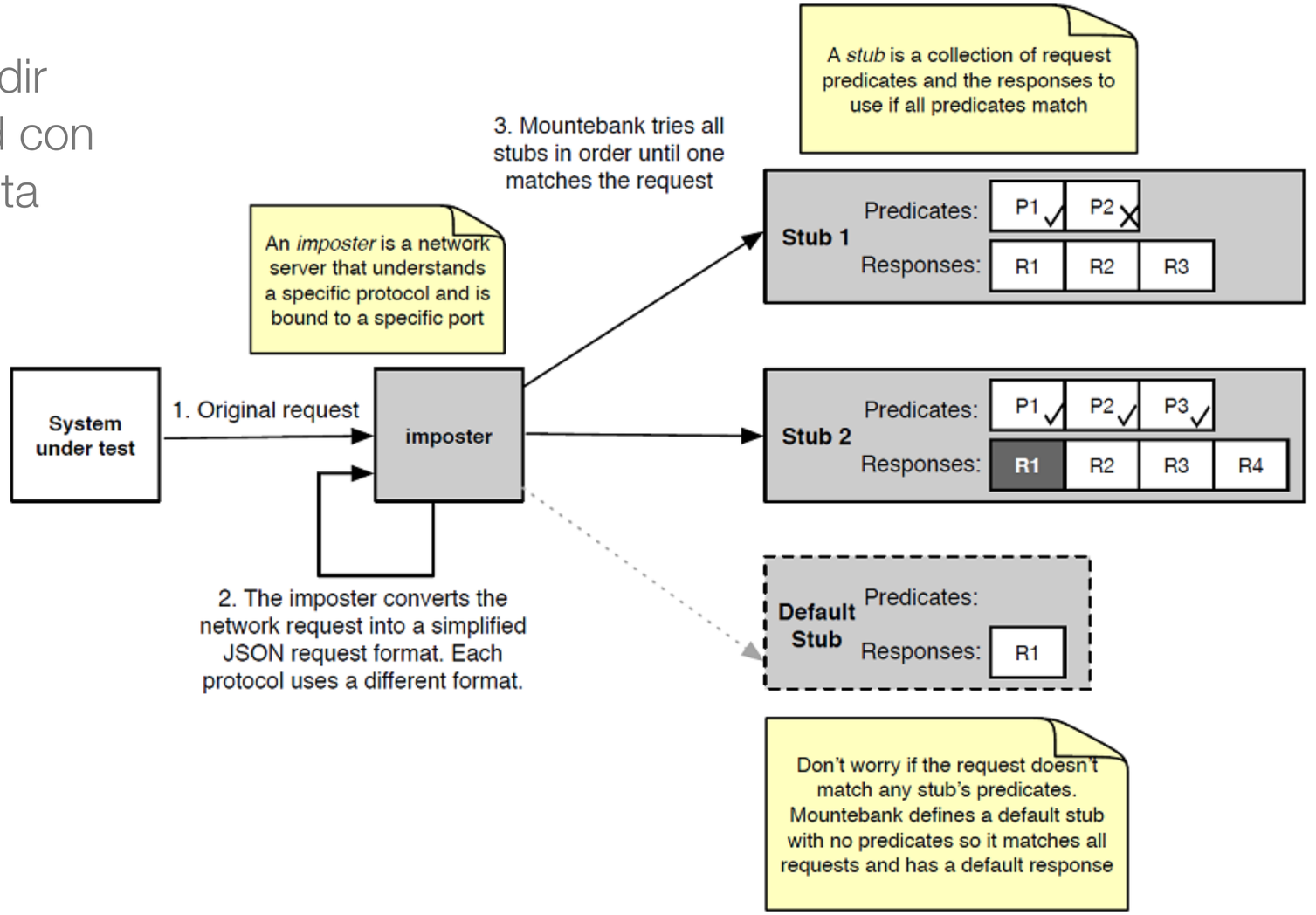
Mountebank es una herramienta liviana de código abierto para la virtualización de los servicios HTTP, HTTPS, SMTP y TCP, que cualquier aplicación bajo prueba puede usar, en lugar del servicio real http://www.mbtest.org/

Testing de Microservicios con Virtualización de Servicios



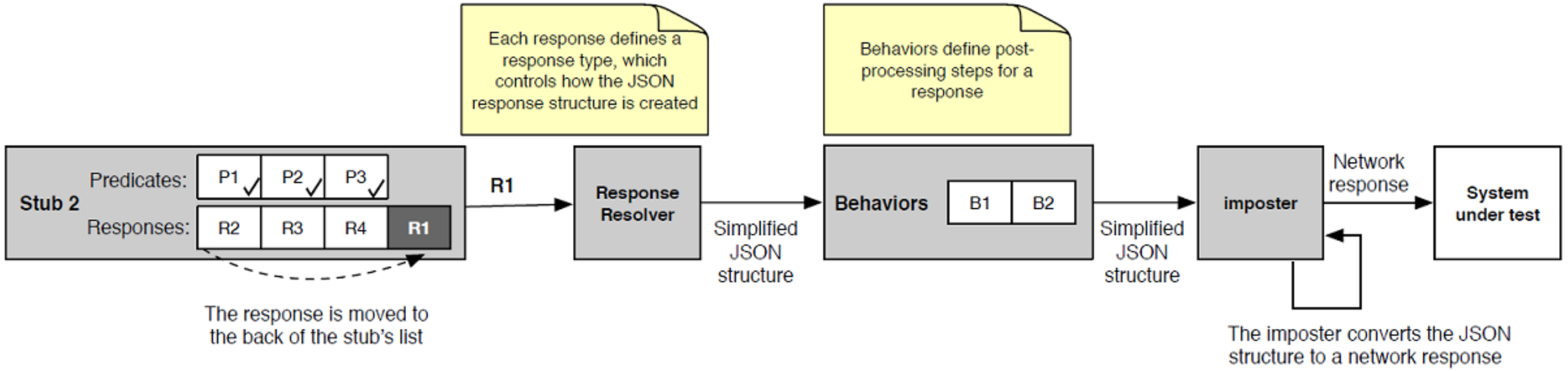
Mountebank, ¿cómo funciona?

hacer coincidir
una solicitud con
una respuesta



Mountebank, ¿cómo funciona?

generación de la respuesta



mountebank, instalación y ejecución

Instalación:

```
npm install -g mountebank
```

Ejecución:

```
mb &
```

```
info: [mb:2525] mountebank v1.12.0 now taking orders -
point your browser to http://localhost:2525 for help
```

```
mb start I stop
```

mountebank, ejemplo 1 - hola mundo

holaMundo.json

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

linux

```
curl -d@holaMundo.json http://localhost:2525/imposters
```

windows

```
Invoke-RestMethod -Method POST -Uri http://localhost:2525/imposters -InFile holaMundo.json
```

log consola

```
info: [mb:2525] POST /imposters
info: [http:3000] Open for business…
```

mountebank, ejemplo hola mundo

para verificar funcionamiento:

linux `curl -i http://localhost:3000`

windows `Invoke-RestMethod -Method GET -Uri http://localhost:3000`

log consola `info: [http:3000] ::1:63325 => GET /`

respuesta

versión protocolo — status

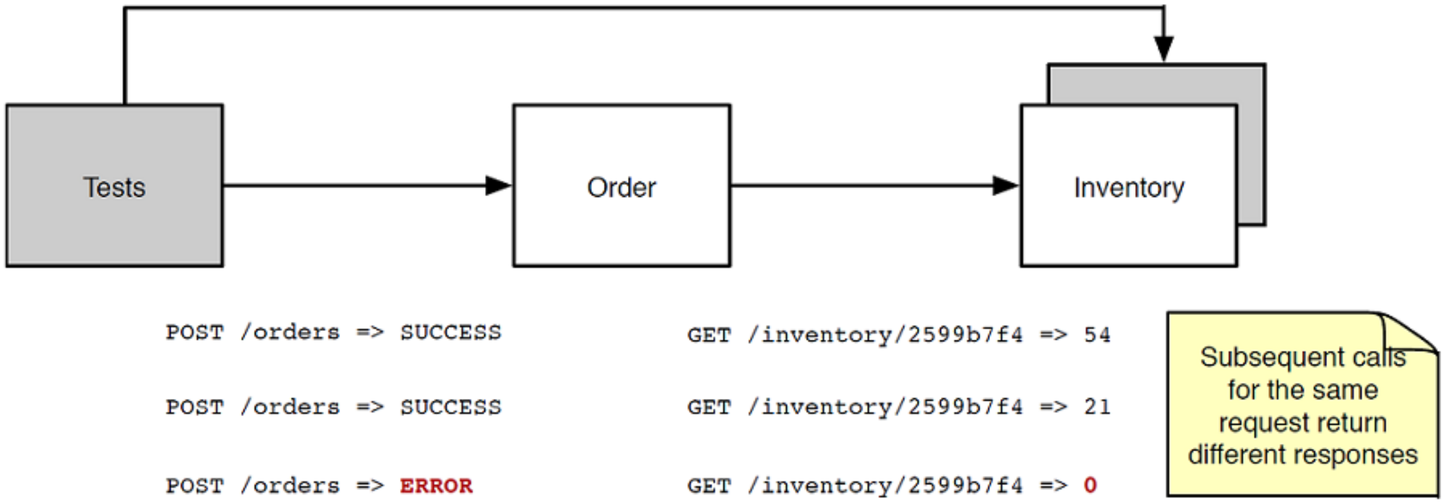
```
HTTP/1.1 200 OK
Content-Type: text/plain
Connection: close
Date: Thu, 25 Oct 2018 13:39:55 GMT
Transfer-Encoding: chunked
```

headers

body

```
Hola, Mundo!
```

mountebank, ejemplo 2 - respuestas circulares



circular.json

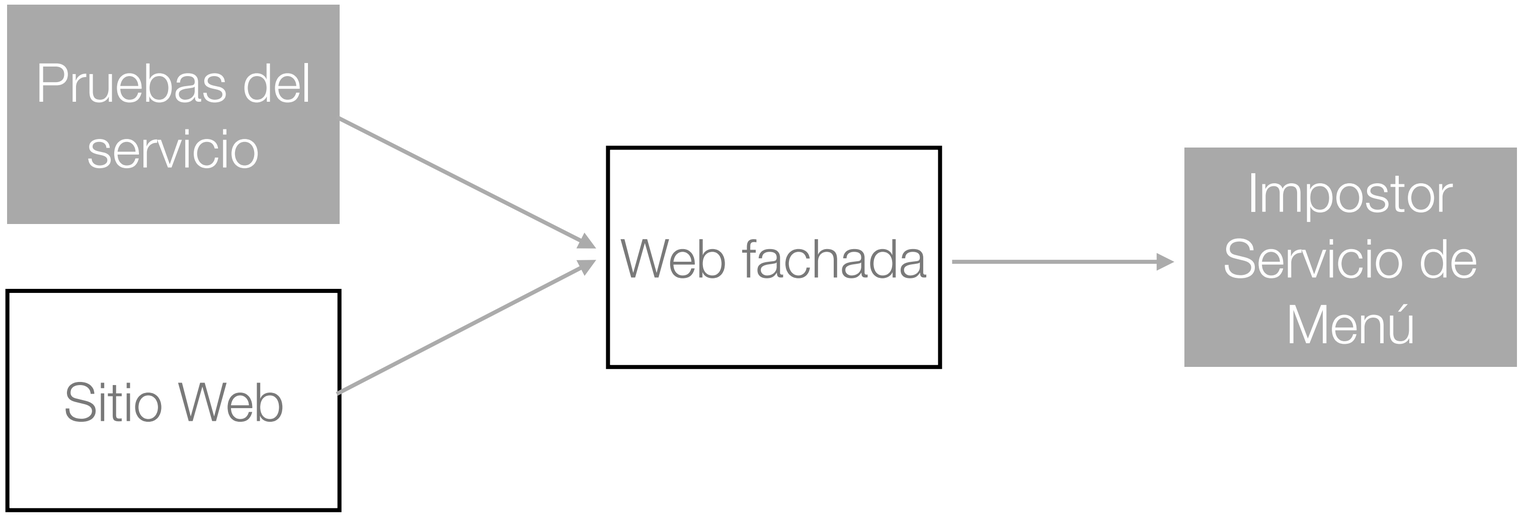
```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [
        { "is": { "body": "54" } },
        { "is": { "body": "21" } },
        { "is": { "body": "0" } }
    ]
  }]
}
```

curl -d@circular.json http://localhost:2525/imposters

curl -i http://localhost:3000

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

mountebank, ej. 3 virtualización servicio de menú de restaurante



mountebank, ej. 3 virtualización servicio de menú de restaurante

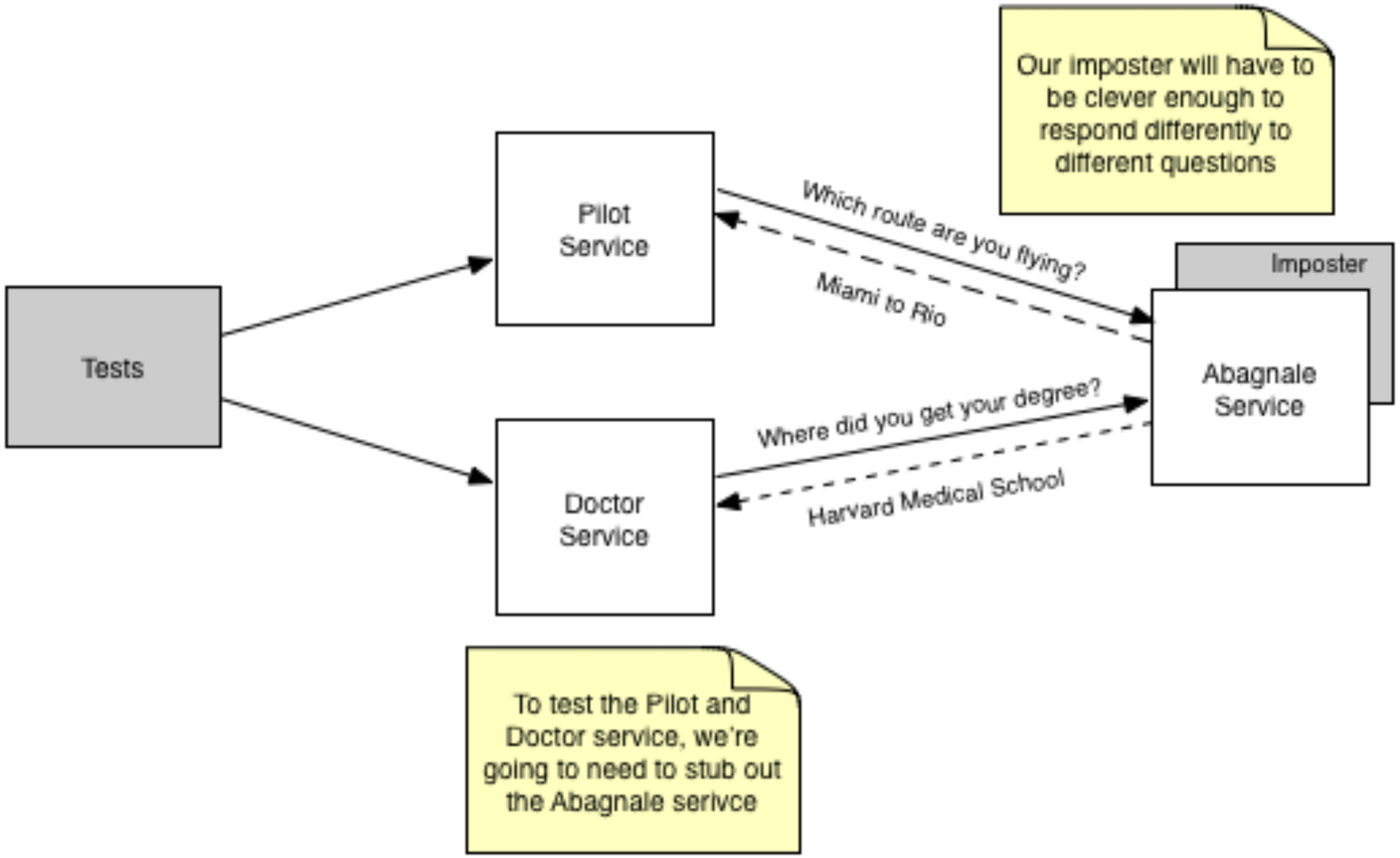
sin predicado

```
{
  "port": 3000,
  "protocol": "http",
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": {"Content-Type": "application/json"},
        "body": {
          "menues": [
              {
                "id": "menu1",
                "name": "La gran Mila",
                "description": "Milanesa con papas fritas"
              },
              {
                "id": "menu2",
                "name": "Super Pollo",
                "description": "Pollo al horno con papas"
              }
          ],
          "_links": {
            "next": "/menues?pagina=2&itemsPorPagina=2"
          }
        }
      }
    }]
  }]
}
```

con predicado

```
{
  "port": 3000,
  "protocol": "http",
  "stubs": [
    {
      "predicates": [{ "equals": { "query": { "pagina": "2" } } }],
      "responses": [{
        "is": {
          "statusCode": 200,
          "headers": { "Content-Type": "application/json" },
          "body": { "menues": [] }
        }
      }]
    },
    {
      "responses": [{
        "is": {
          "statusCode": 200,
          "headers": { "Content-Type": "application/json" },
          "body": {
            "menues": [
              {
                "id": "menu1",
                "name": "La gran Mila",
                "description": "Milanesa con papas fritas"
              },
              {
                "id": "menu2",
                "name": "Super Pollo",
                "description": "Pollo al horno con papas"
              },
            ],
            "_links": { "next": "/menues?pagina=2&itemsPorPagina=3" }
          }
        }
      }]
    }
  ]
```

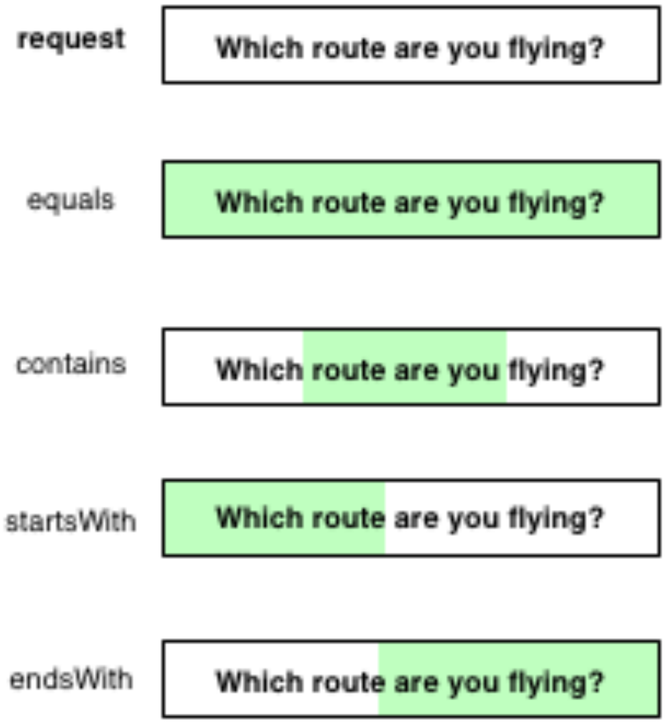
mountebank, ej. 4 Servicio Abagnale - servicio con predicado



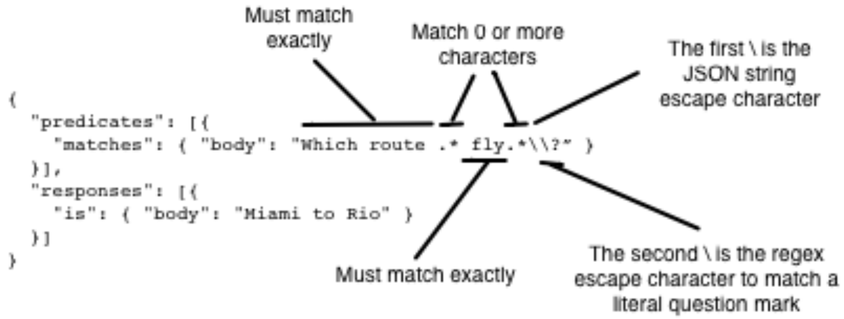
tipos de predicado

mountebank, ej. 3 Servicio Abagnale - tipos de predicado

tipos de predicado



The shaded area represents an example predicate to match against the entire request field



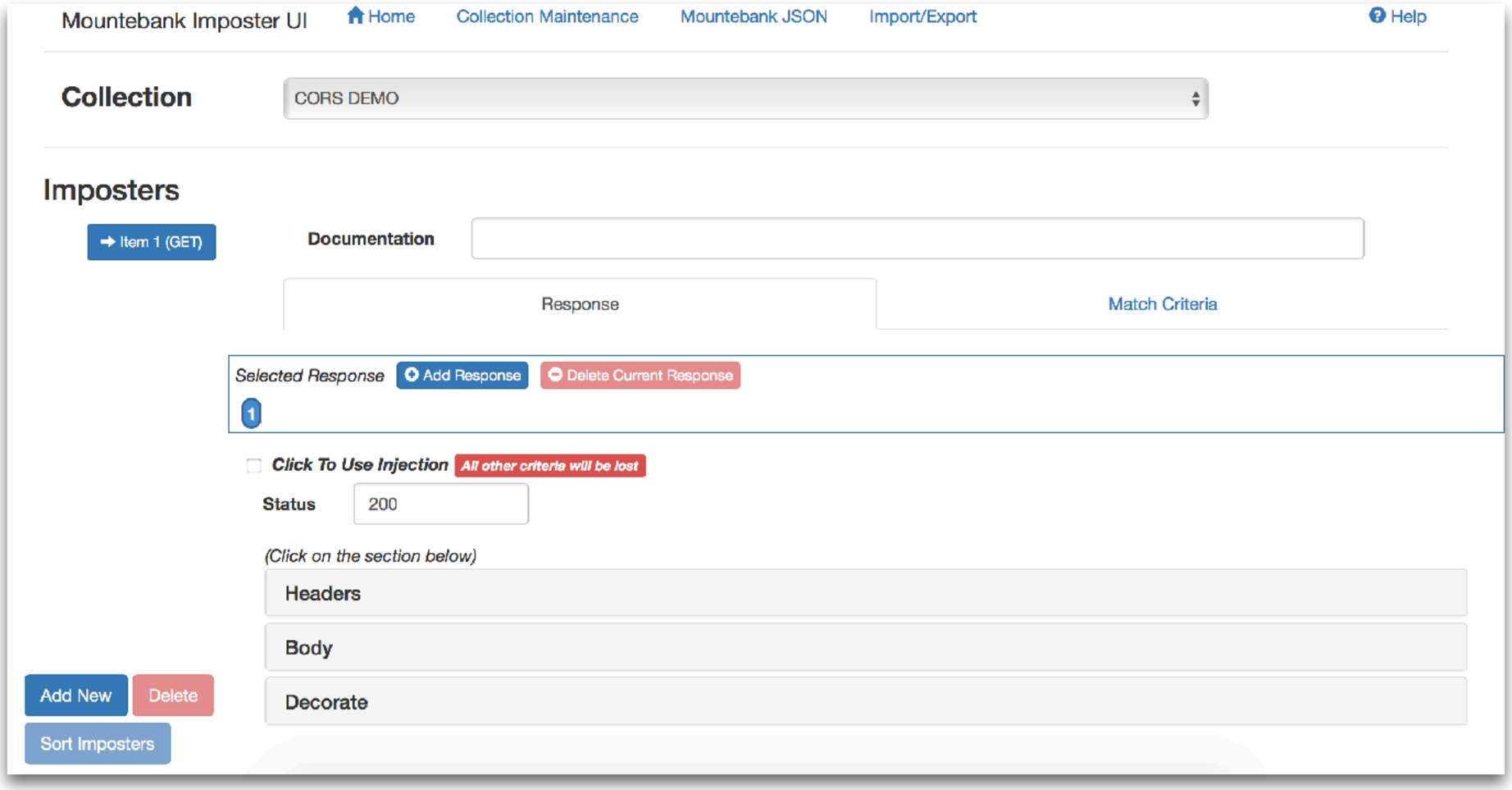
wetek.io

mountebank, tipos de predicados

Operador	Descripción
equals	Requiere que el campo de solicitud sea igual al valor de predicado
deepEquals	Realiza la igualdad de conjuntos anidados en los campos de solicitud de objetos.
contains	Requiere que el campo de solicitud contenga el valor del predicado
startsWith	Requiere que el campo de solicitud comience con el valor de predicado
endsWith	Requiere que el campo de solicitud termine con el valor de predicado
matches	Requiere que el campo de solicitud coincida con la expresión regular proporcionada como valor de predicado
exists	Requiere que el campo de solicitud exista como un valor no vacío (si es True) o no (si es False)
not	Invierte el sub-predicado.
or	Requiere cualquiera de los sub-predicados para ser satisfecho
and	Requiere que todos los sub-predicados estén satisfechos
inject	Requiere una función proporcionada por el usuario para devolver True

mountebank, Imposter UI

REST Testing with Mountebank, by Donaby Henton



https://github.com/donhenton/mountebank-UI

wetek.io

mountebank, Parte II: avanzado

- Requerimientos HTTPS
- Proxy (Play/Recording)
- Respuestas dinámicas: Javascript Injection (ej. OAUTH Authent/Authoriz.)
- Comportamientos: post-procesar un requerimiento programáticamente
- Mountebank y Continuous Delivery (CD)

práctica (hands-on)

mountebank, dockerizado

Instala la última versión de mountebank y expone el puerto 2525 listo para configuración y prueba.

```
docker run -p 2525:2525 -d jkris/mountebank
```

Si desea cargar sus impostores automáticamente, montarlos como un Volumen y enlazar los puertos necesarios:

```
docker run \
    -v ./imposters:/imposters \
    [-p IMPOSTER_1_PORT:IMPOSTER_1_PORT ...] \
    -p 2525:2525 \
    -d jkris/mountebank \
    --configfile /imposters/imposters.ejs --allowInjection
```

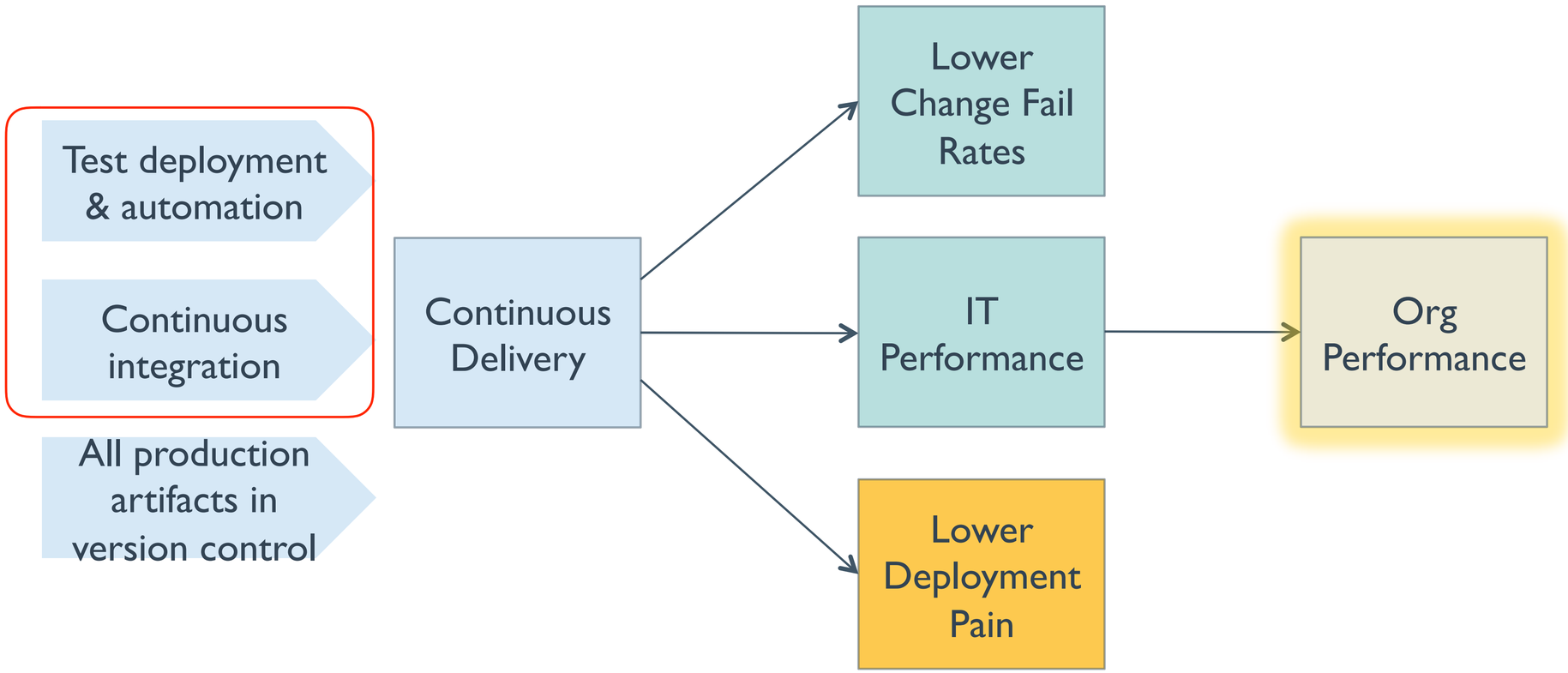
Links de interés

- https://www.mbtest.org/
- Mountebank UI - https://github.com/donhenton/mountebank-UI
- REST Testing with Mountebank and Donaby Henton https://www.youtube.com/watch?v=69usGV3uScI&list=WL&index=5
- https://tzinov15.github.io/swagger-bank/
- https://blog.codecentric.de/en/2015/06/mock-server-powered-by-mountebank-and-docker/#mountebank
- https://www.digitalocean.com/community/tutorials/how-to-mock-services-using-mountebank-and-node-js https://www.soapui.org/learn/mocking/what-is-api-virtualization.html

weteк.io

ANEXOS

IT/Organization Performance



NOTA: “The Role of Continuous Delivery in IT and Organizational Performance”, Dr. Nicole Forsgren y Jez Humble, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2681909