zalando

END-TO-END LOAD TESTING AT SCALE

OLIWIA ZAREMBA

CONTINUOUS TESTING MEETUP

10-09-2019

WHY LOAD TESTING?

BLACK FRIDAY

Source: giphy.com

BLACK FRIDAY

- **Auto-scaling is not a solution for huge spikes**

→ To handle a huge spike of traffic, all services need to be pre-scaled beforehand

- **Scaling into infinity costs infinite $$$**

→ The scaling configuration needs to be frugal

Problem statement:

For each service S_i find a minimum value of the scaling parameter k_i, at which the service can handle the expected load L.

WHY END-TO-END?

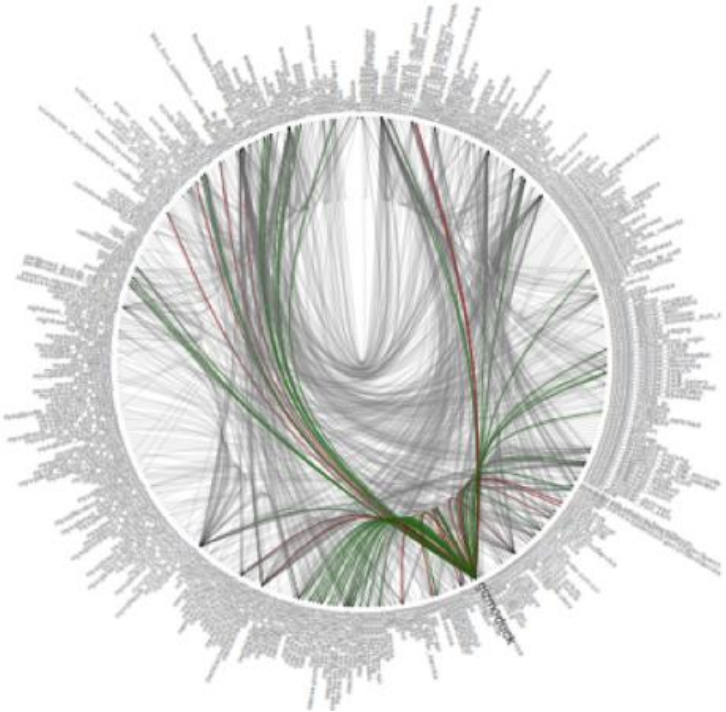
WHY NOT MAKING IT THIS SIMPLE?

```
curl --get https://www.zalando.de/
```

WHY NOT MAKING IT THIS SIMPLE?

- Real customers perform different actions: browsing, filtering, checking out, …
- These actions are served by many different services

MICROSERVICES



Source: https://thenewstack.io/history-service-mesh/

Definition of the task:

Generate an increasingly high **realistic** load to identify bottlenecks in the microservices.

AT WHAT SCALE?

BLACK FRIDAY 2017 NUMBERS

- **2,000** orders per minute (**1,500** in 2016)
- **100,000** new customers

 Source: https://corporate.zalando.com/en/newsroom/en/stories/zalando-celebrates-successful-black-friday-2017

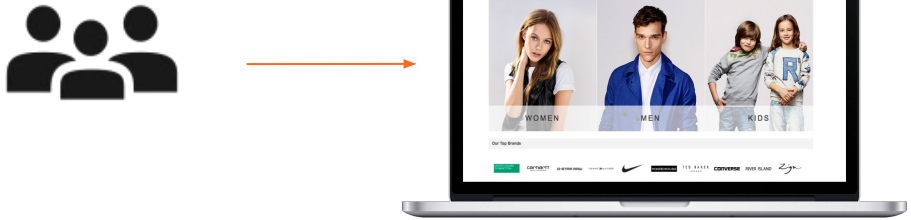
BLACK FRIDAY 2017 NUMBERS

- **2,000** orders per minute (**1,500** in 2016)
- **100,000** new customers

… and expectations for **bigger numbers** in 2018

HOW DO WE WANT TO ACHIEVE IT?

SIMULATE REAL USERS

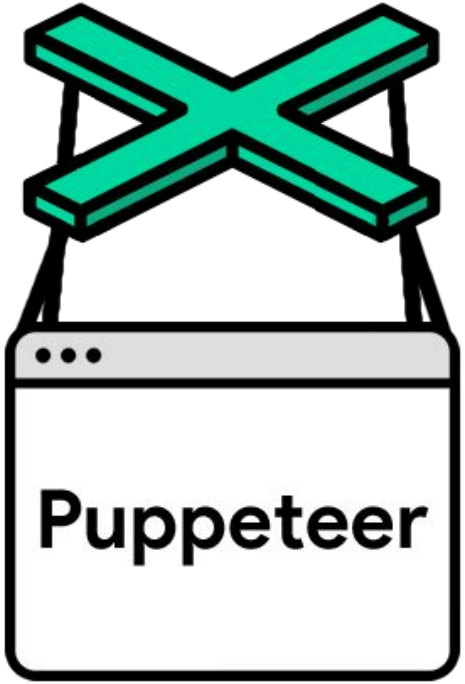


zalando

SIMULATE REAL USERS

zalando

SIMULATE REAL USERS - WITH PUPPETEER?

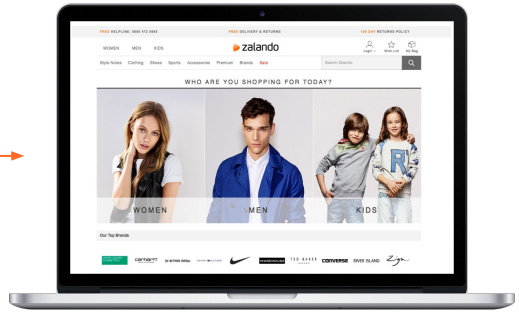


SIMULATE REAL USERS - WITH PUPPETEER?

SIMULATE REAL USERS - WITH PUPPETEER?

SIMULATE REAL USERS… AND MAKE IT CHEAP

team

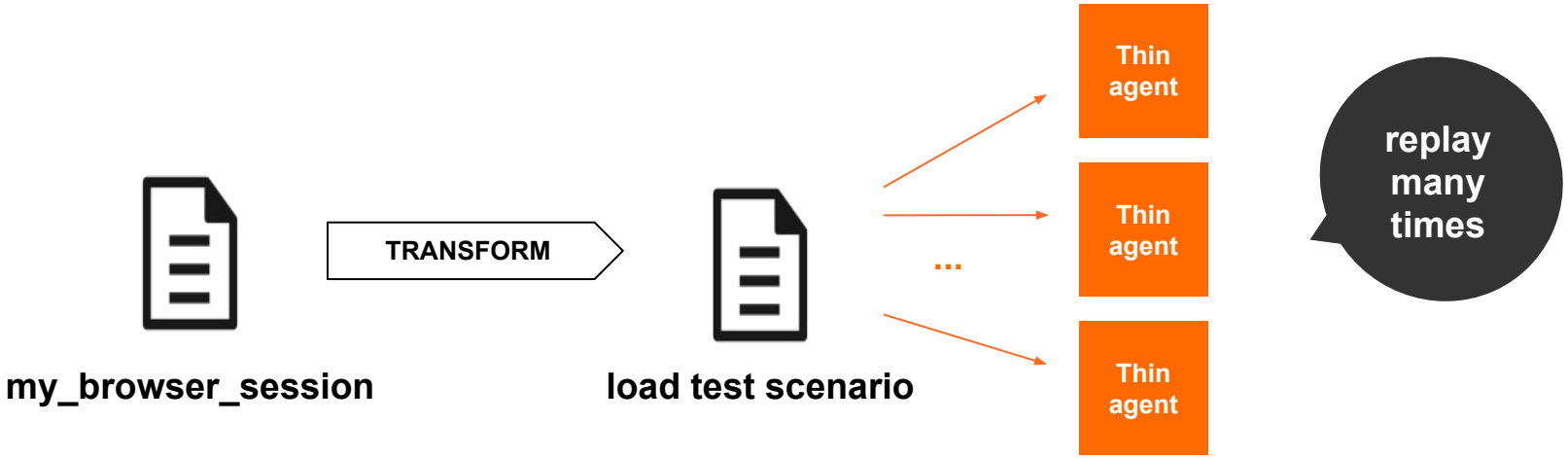


my_browser_session

record once

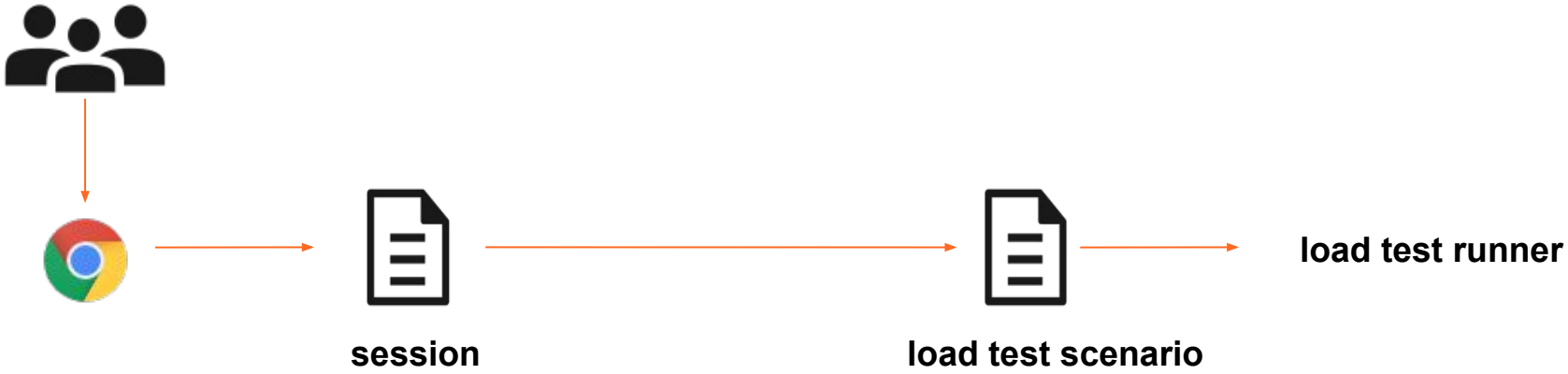
zalando

SIMULATE REAL USERS… AND MAKE IT CHEAP



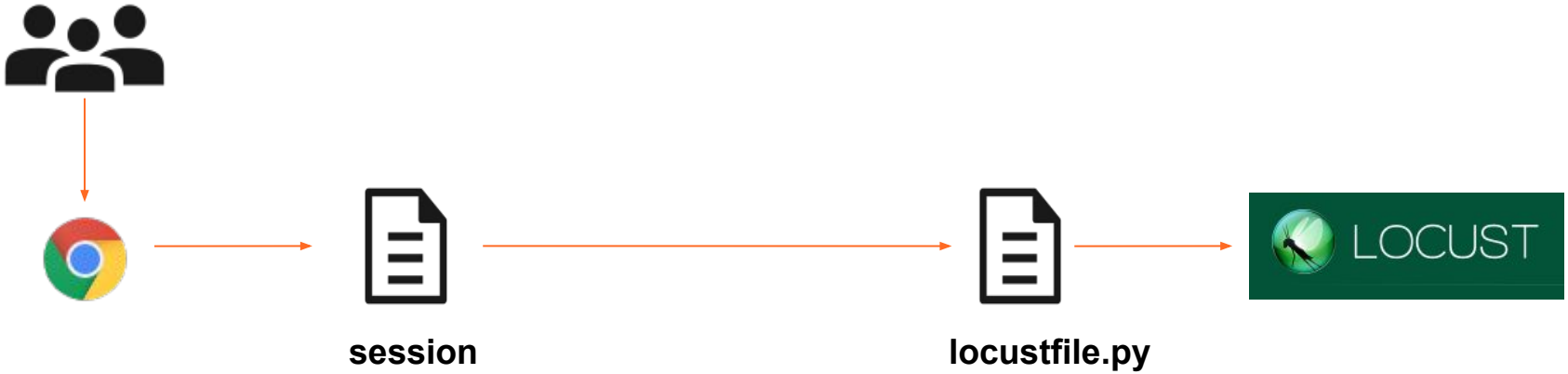
IMPLEMENTATION

RECORDING BY THE TEAM + REPLAYING BY THE LOAD TEST RUNNER

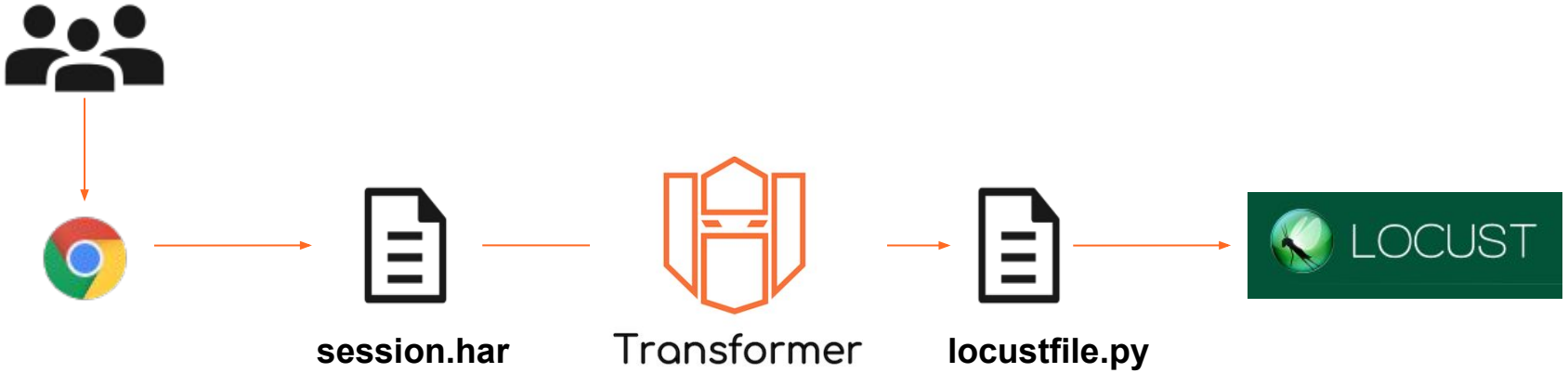


zalando

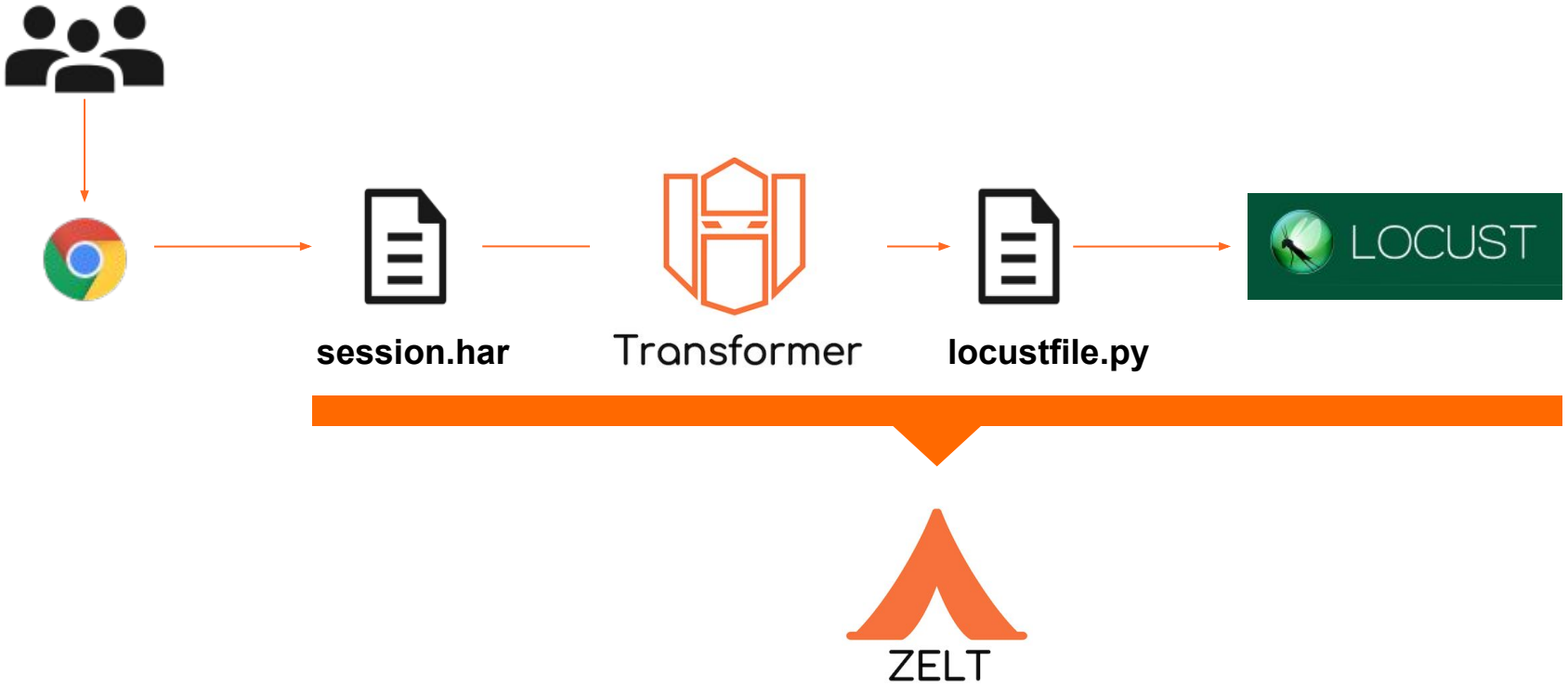
RECORDING BY THE TEAM + REPLAYING BY LOCUST



TRANSFORMER + LOCUST

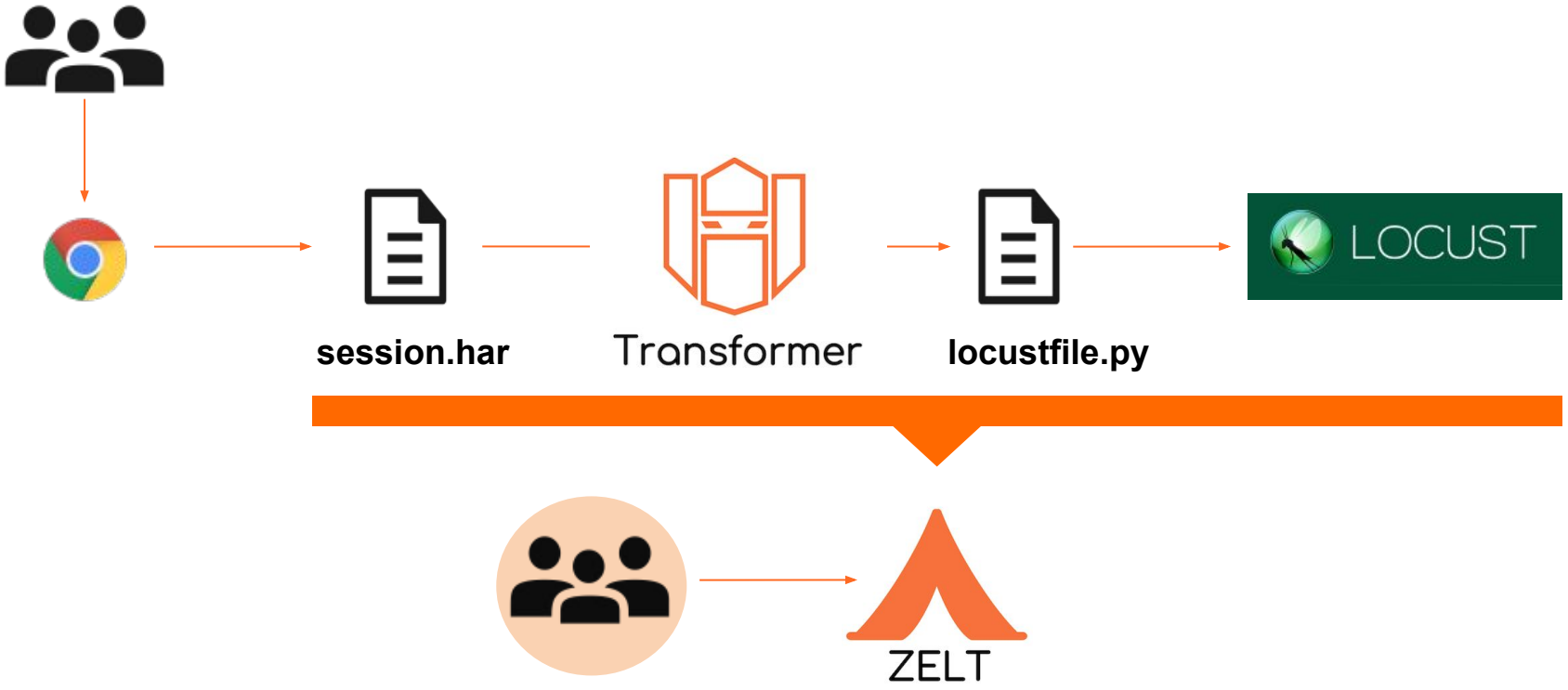


TRANSFORMER + ZELT + LOCUST

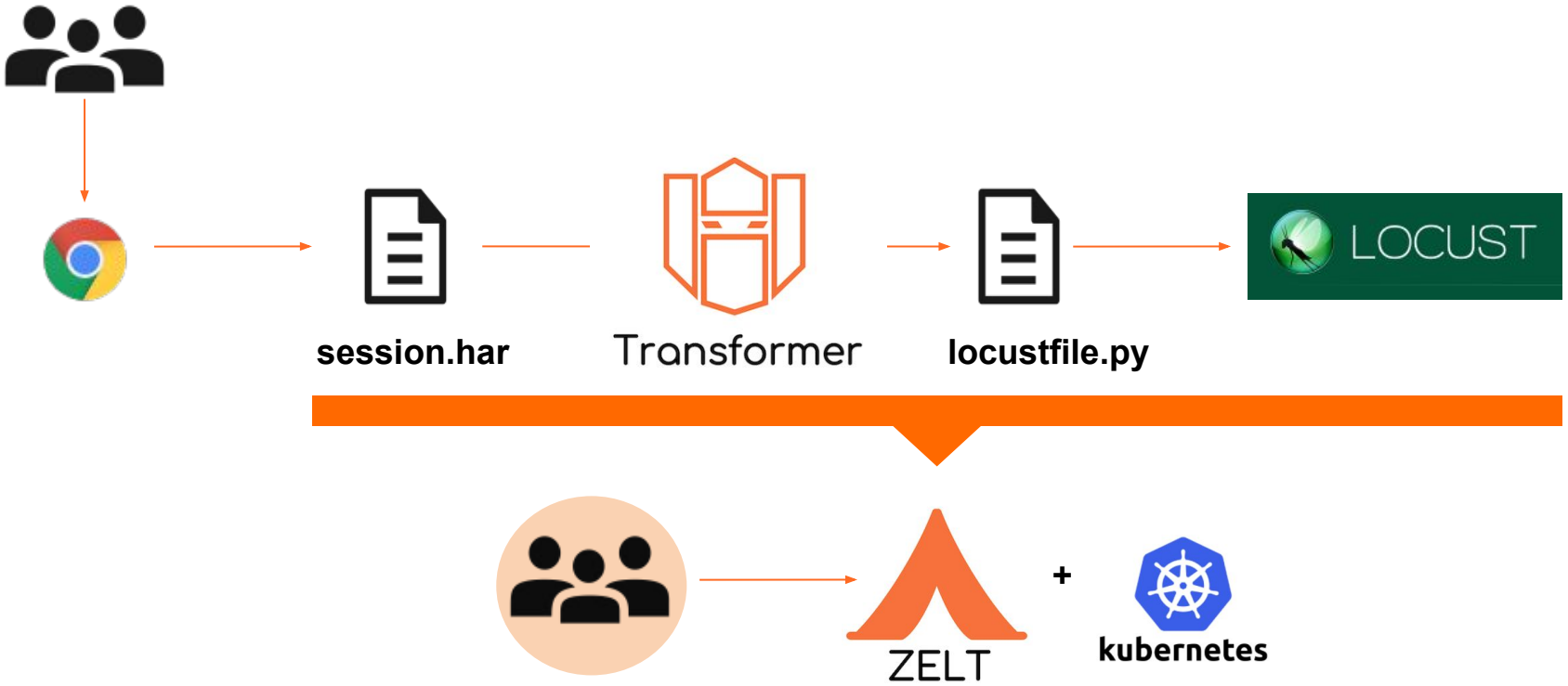


zalando

TRANSFORMER + ZELT + LOCUST



TRANSFORMER + ZELT + LOCUST + KUBERNETES



CHOOSING THE TECHNOLOGY

 Source: locust.io

CHOOSING THE TECHNOLOGY

Define user behaviour in code

No need for clunky UIs or bloated XML. Just plain code.

Distributed & scalable

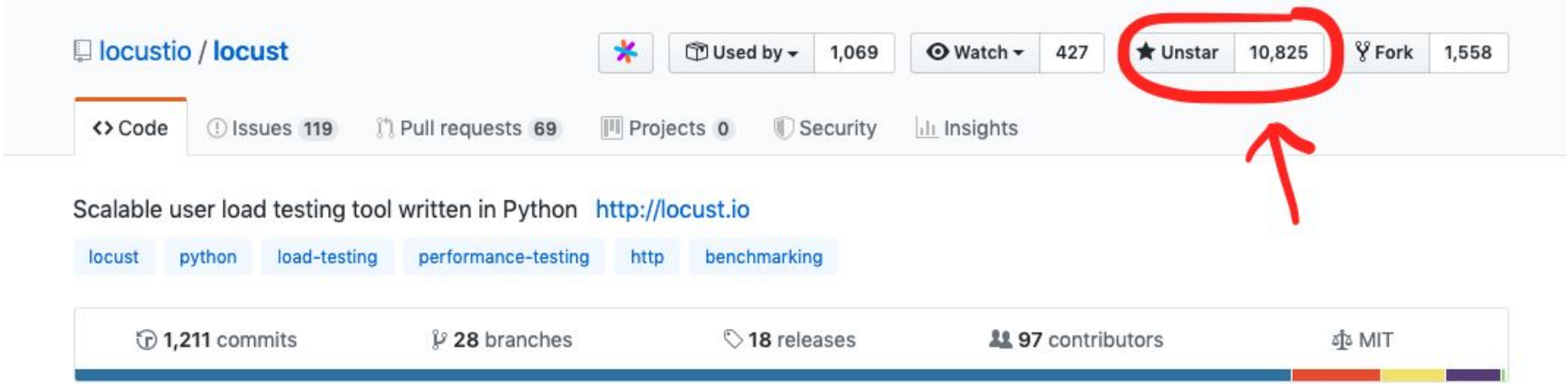
Locust supports running load tests distributed over multiple machines, and can therefore be used to simulate millions of simultaneous users

Proven & battle tested

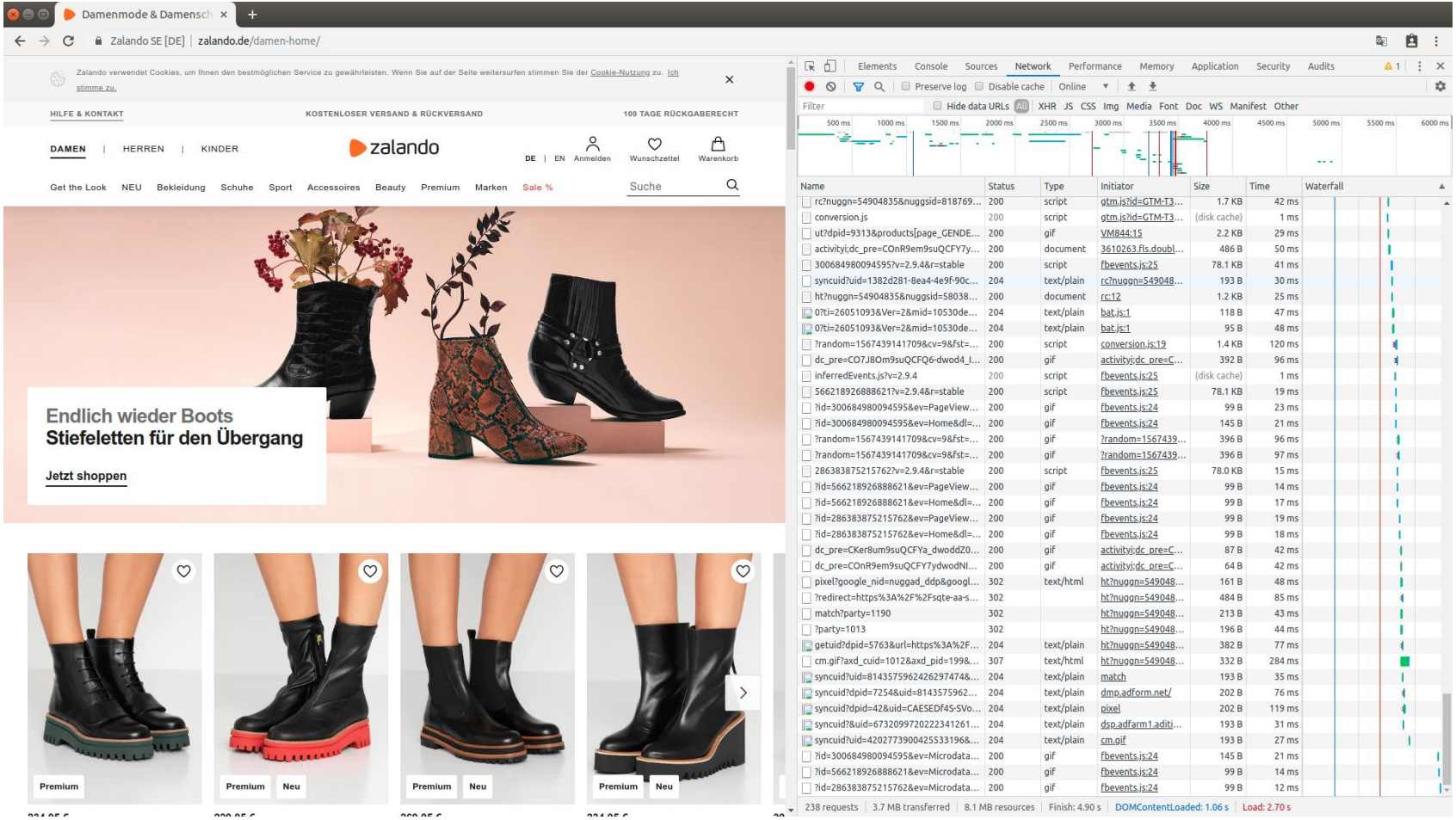
Locust has been used to simulate millions of simultaneous users. Battlelog, the web app for the Battlefield games, is load tested using Locust, so one can really say Locust is Battletested ;).

 Source: locust.io

CHOOSING THE TECHNOLOGY



 Source: github.com/locustio/locust


Damenmode & Damensch
Zalando SE [DE] | zalando.de/damen-home/
Zalando verwendet Cookies, um Ihnen den bestmöglichen Service zu gewährleisten. Wenn Sie auf der Seite weitersurfen stimmen Sie der Cookie-Nutzung zu. Ich stimme zu.
HILFE & KONTAKT
KOSTENLOSER VERSAND & RÜCKVERSAND
100 TAGE RÜCKGABERECHT
DAMEN | HERREN | KINDER
zalando
DE | EN
Anmelden
Wunschzettel
Warenkorb
Get the Look NEU Bekleidung Schuhe Sport Accessoires Beauty Premium Marken Sale %
Suche
Endlich wieder Boots
Stiefeletten für den Übergang
Jetzt shoppen
Premium
Premium Neu
Premium
Premium Neu
Elements Console Sources Network Performance Memory Application Security Audits
Preserve log
Disable cache
Online
Filter
Hide data URLs
All XHR JS CSS Img Media Font Doc WS Manifest Other
Name Status Type Initiator Size Time Waterfall
238 requests | 3.7 MB transferred | 8.1 MB resources | Finish: 4.90 s | DOMContentLoaded: 1.06 s | Load: 2.70 s

zalando

HOW IT WORKS: 1. RECORDING THE USER BEHAVIOUR

- **HAR** - **H**TTP **AR**chive
- File extension: .har
- Format: JSON

HOW IT WORKS: 1. RECORDING THE USER BEHAVIOUR


```
[
    {
        "url": "https://www.zalando.de/damen-home/",
        "method": "GET"
    },
    {
        "url": "https://www.google-analytics.com/analytics.js",
        "method": "GET"
    },
    {
        "url": "https://mosaic02.ztat.net/nvg/z-footer-fragment/production/svg-icons/de-DE/1565255942669/Discover.png",
        "method": "GET"
    }
]
```


zalando

HOW IT WORKS: 2. TRANSFORMING HAR INTO LOCUSTFILE

my_browser_session.har

```
[
    {
        "url": "http://abc.de",
        "method": "GET"
    },
    {
        "url": "http://x.yz",
        "method": "POST"
    }
]
```


locustfile.py

```
class Scenario(TaskSet):
    @task
    def abc_de():
        requests.get("http://abc.de")

    @task
    def x_yz():
        requests.post("http://x.yz")
```

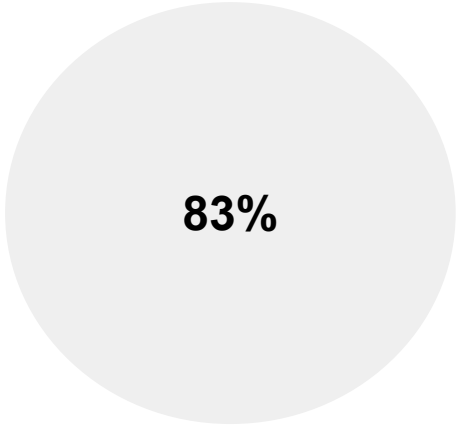
HOW IT WORKS: 2. TRANSFORMING SCENARIOS WITH WEIGHTS

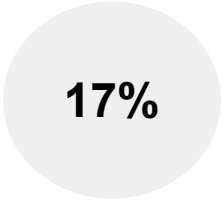
browsing_items_scenario.har

```
[
    {
        "url": "http://abc.de/item/very-nice-shoes",
        "method": "GET"
    },
    {
        "url": "http://abc.de/item/black-dress",
        "method": "GET"
    }
]
```

checkout_scenario.har

```
[
    {
        "url": "http://abc.de/item/very-nice-shoes",
        "method": "GET"
    },
    {
        "url": "http://abc.de/checkout",
        "method": "POST"
    }
]
```





zalando

HOW IT WORKS: 2. TRANSFORMING SCENARIOS WITH WEIGHTS

browsing_items_scenario.har

```
[
    {
        "url": "http://abc.de/item/very-nice-shoes",
        "method": "GET"
    },
    {
        "url": "http://abc.de/item/black-dress",
        "method": "GET"
    }
]
```

checkout_scenario.har

```
[
    {
        "url": "http://abc.de/item/very-nice-shoes",
        "method": "GET"
    },
    {
        "url": "http://abc.de/checkout",
        "method": "POST"
    }
]
```

locustfile.py

```
class GermanyScenarios(TaskSet):

    @task(83)
    class BrowsingScenario(TaskSequence):
        @seq_task(1)
        def get_nice_shoes():
            requests.get("http://abc.de/item/very-nice-shoes")

        @seq_task(2)
        def get_black_dress():
            requests.get("http://abc.de/item/black-dress")

    @task(17)
    class CheckoutScenario(TaskSequence):
        @seq_task(1)
        def get_nice_shoes():
            requests.get("http://abc.de/item/very-nice-shoes")

        @seq_task(2)
        def check_out():
            requests.post("http://abc.de/checkout")
```

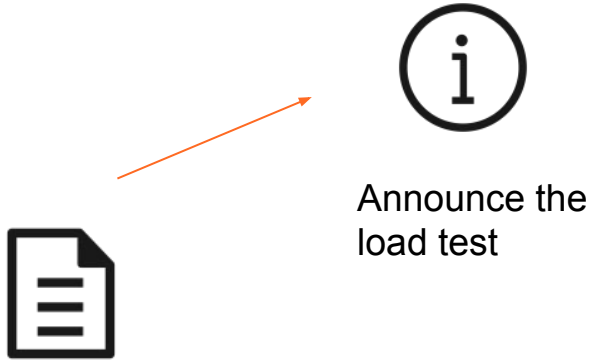
HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS

Plan and record the scenarios

Input: # RPS for each major service

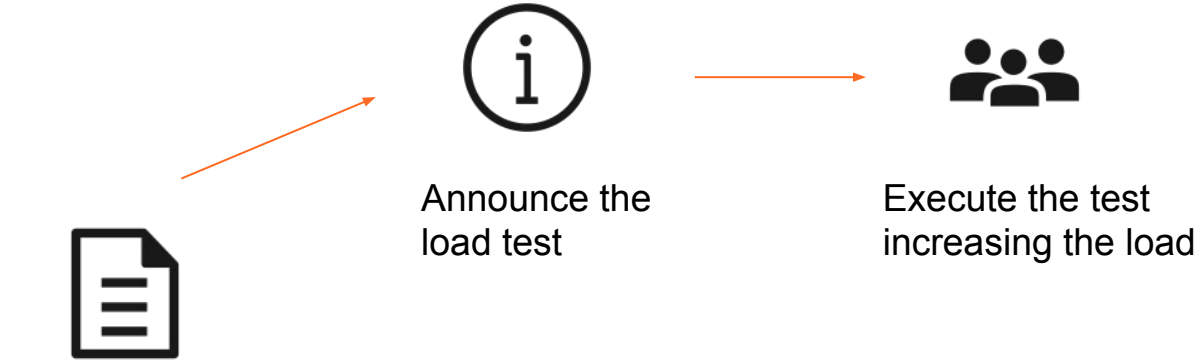
Output: HAR files with the scenarios recorded

HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS



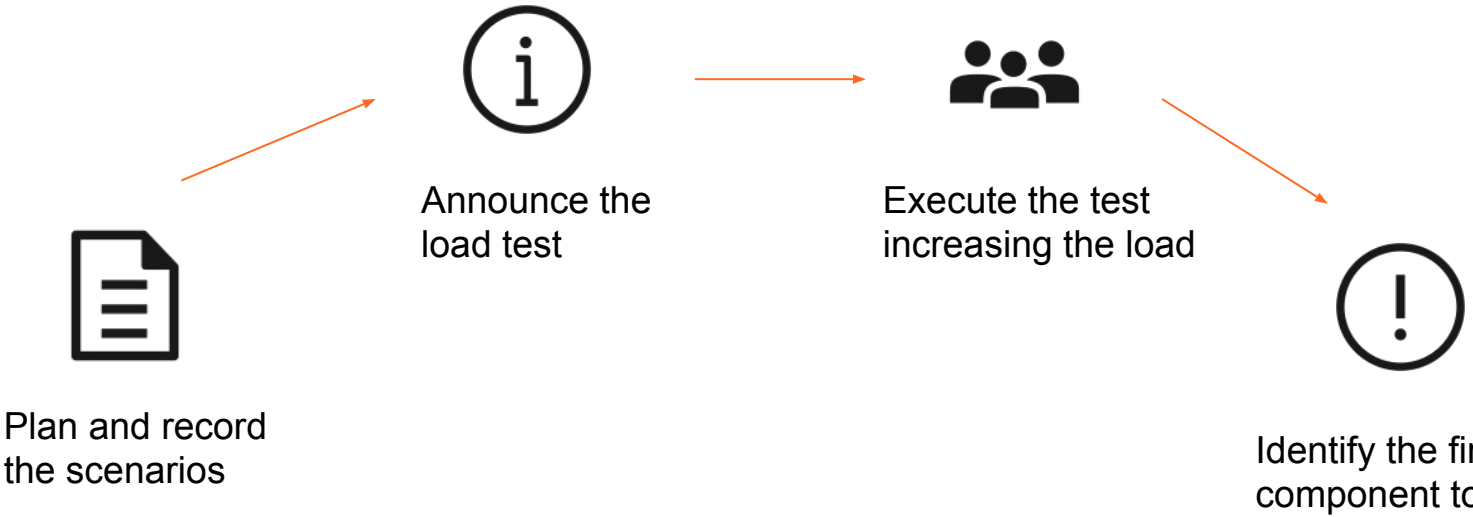
zalando

HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS



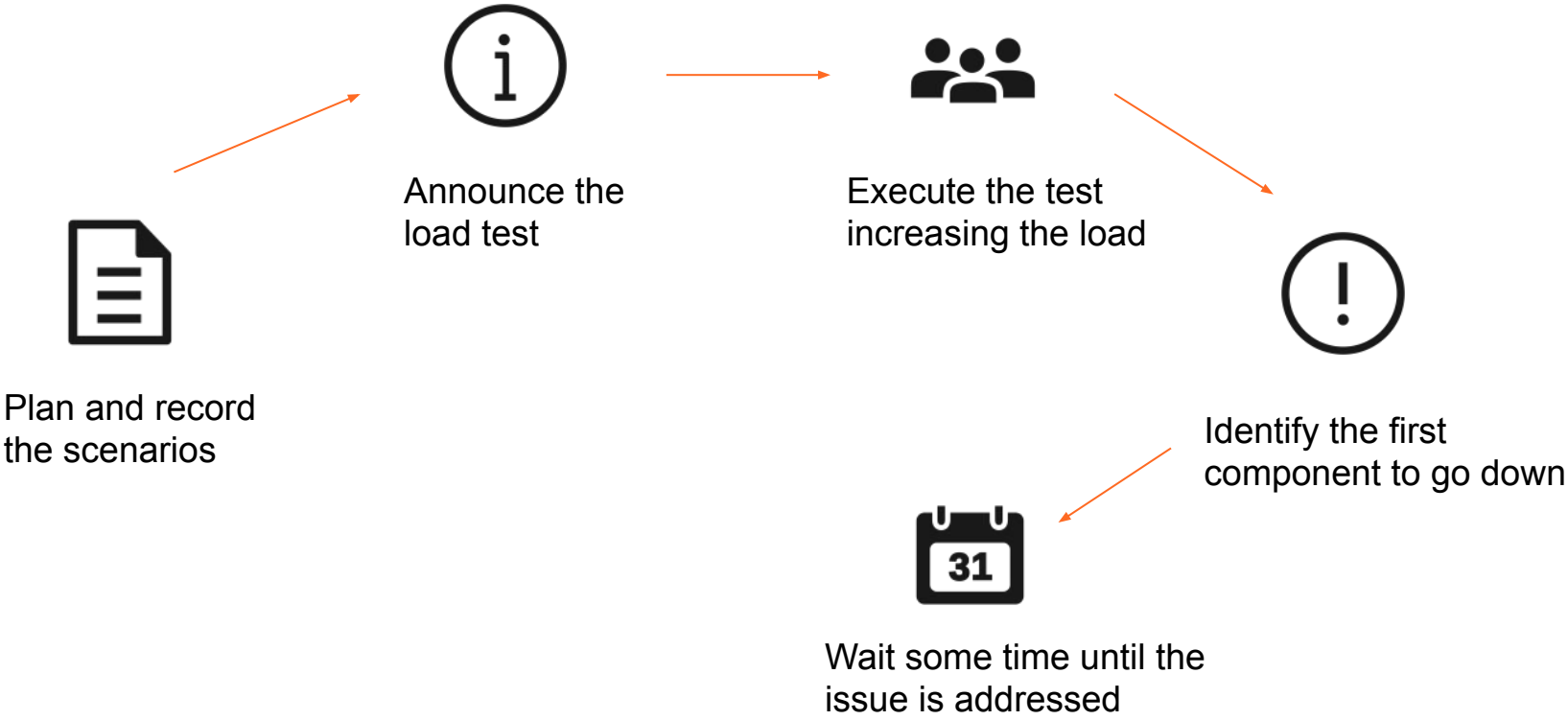
zalando

HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS



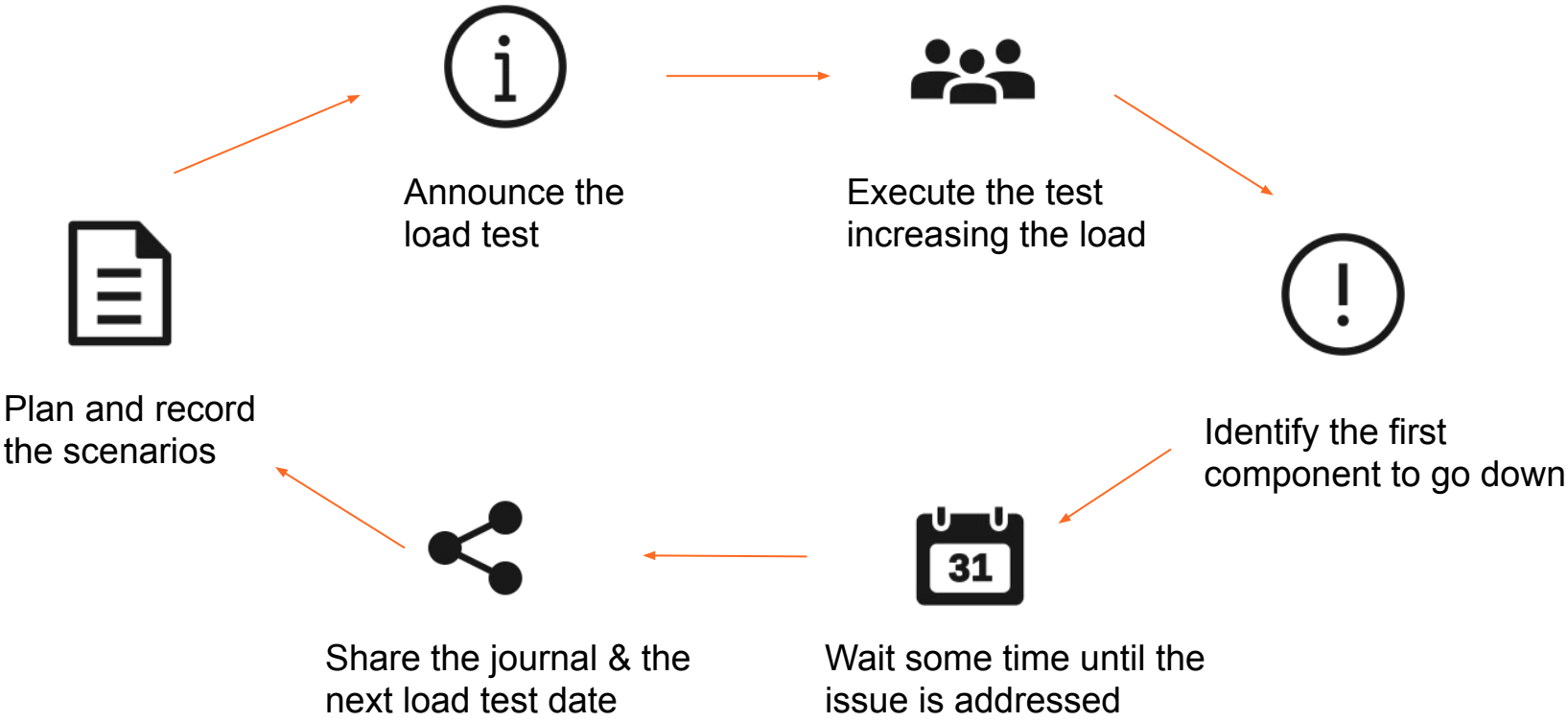
zalando

HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS



zalando

HOW IT WORKS: 3. EXECUTING THE END-TO-END LOAD TESTS



zalando

SOME OBSTACLES DOWN THE ROAD

OBSTACLE 1: SECURITY SYSTEM BLOCKED US

OBSTACLE 1: SECURITY SYSTEM BLOCKED US

- End-to-end load test is in reality a well-intended DoS attack

OBSTACLE 1: SECURITY SYSTEM BLOCKED US

- **Solution**: mark all requests coming from Zelt easily identifiable by the security system
- Analytics, machine learning models, A/B tests need to filter out Zelt traffic too!

OBSTACLE 2: COOKIES RECORDED IN THE HAR FILE ARE NOT VALID WHEN REPLAYING

- **Solution**: don’t process the cookies as recorded. Instead, let the cookies be set by response headers in the replay mode

```
"Set-Cookie: <cookie-name>=<cookie-value>"
```

OBSTACLE 3: WE CAN'T KEEP USING THE SAME TEST CUSTOMER ACCOUNT

- **Solution**: override the customer credentials in the registration/login step with test accounts
- Parameterize the scenarios: for each execution, choose a random account from a defined set

OBSTACLE 4: WE ONLY WANT TO TARGET ZALANDO, NOT GOOGLE ANALYTICS ENDPOINTS

- **Solution**: provide a blacklisting mechanisms for automatic filtering of the recorded requests

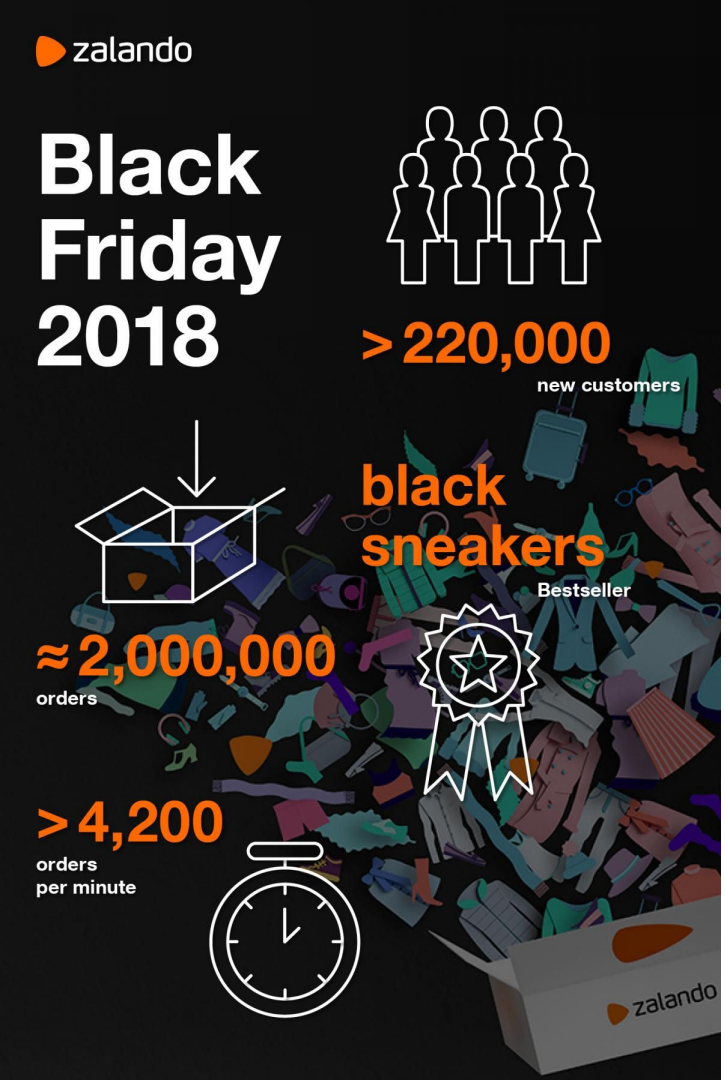
OBSTACLE 5: MORE AND MORE ZALANDO-SPECIFIC MECHANISMS NEED TO BE ADDRESSED

- **Solution**: introduce a system of plugins for Transformer
- Implement each Zalando-specific solution as a plugin

HOW DID WE DO?

FINAL CONFIGURATION

zalando



OFFICIAL RESULTS OF THE BLACK FRIDAY CAMPAIGN

Source: corporate.zalando.com

zalando

ONE MORE THING...

github.com/zalando-incubator/transformer

github.com/zalando-incubator/zelt

zalando

OLIWIA ZAREMBA

SOFTWARE ENGINEER

oliwia.zaremba@zalando.de

twitter.com/tortilato