Grafana Labs

k6

Load Testing

And shifting left

Dougal Matthews

Senior Engineer

dougal@grafana.com

Today's Goal:

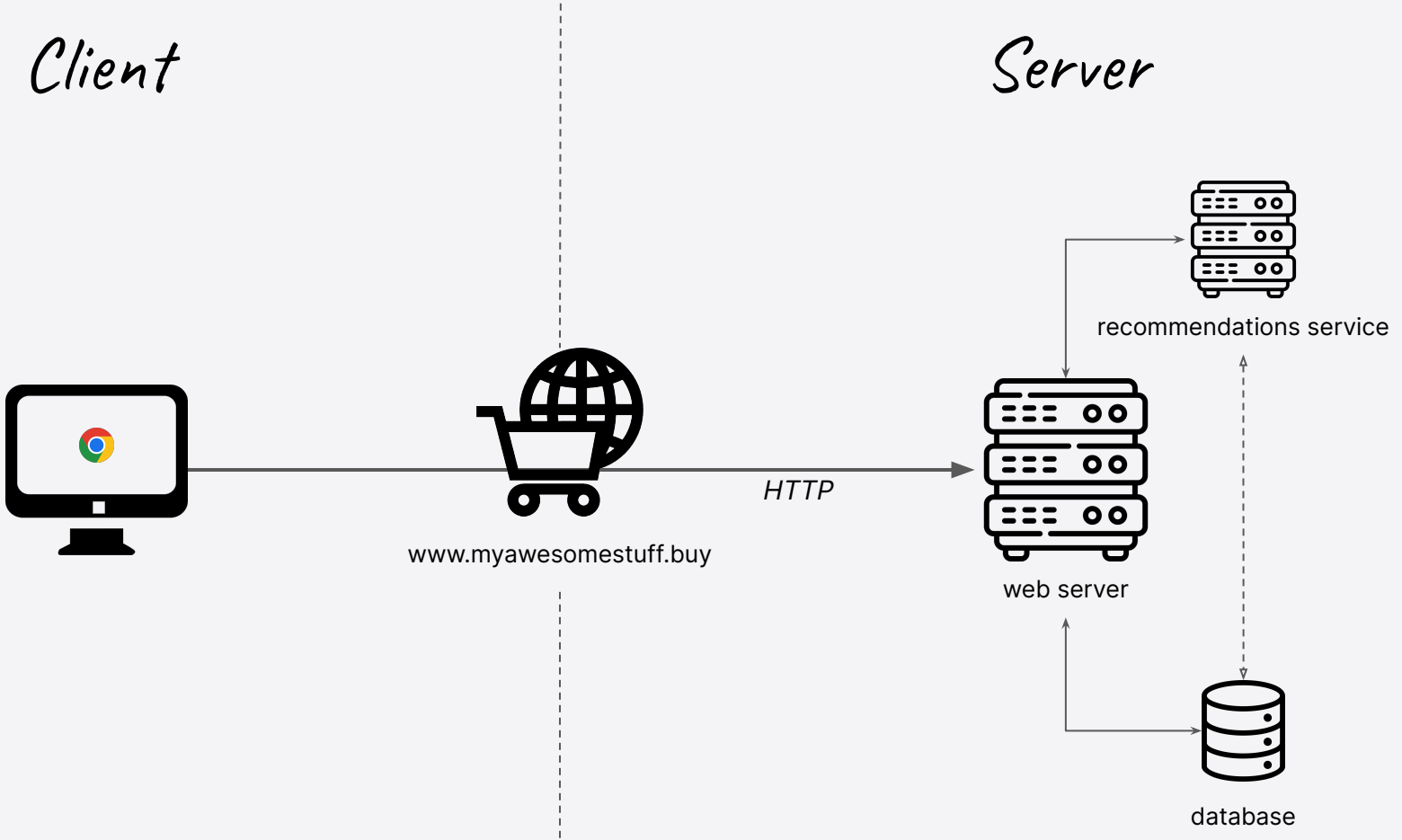
- Load testing introduction
- k6 introduction

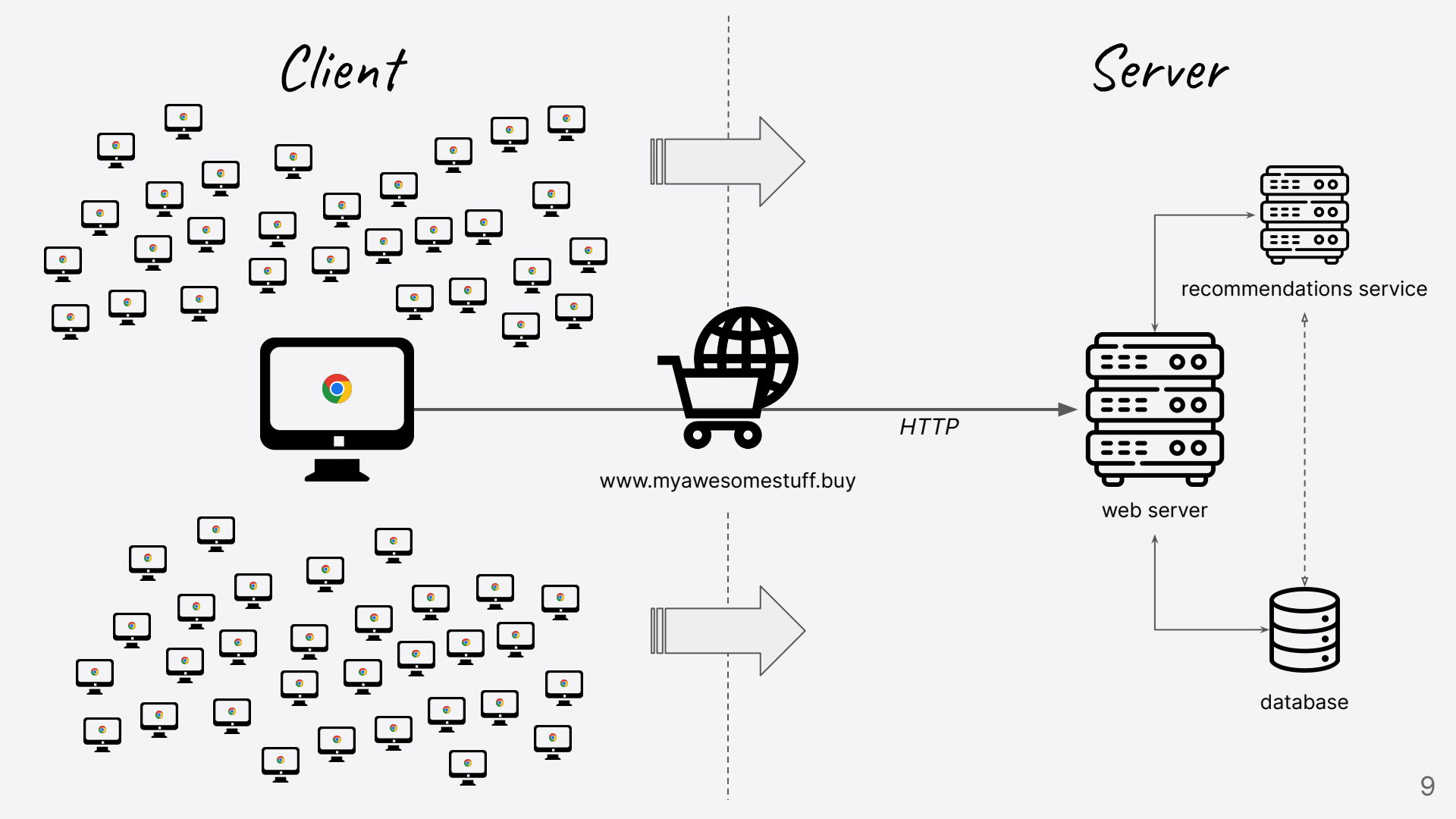
Does your company do load testing?

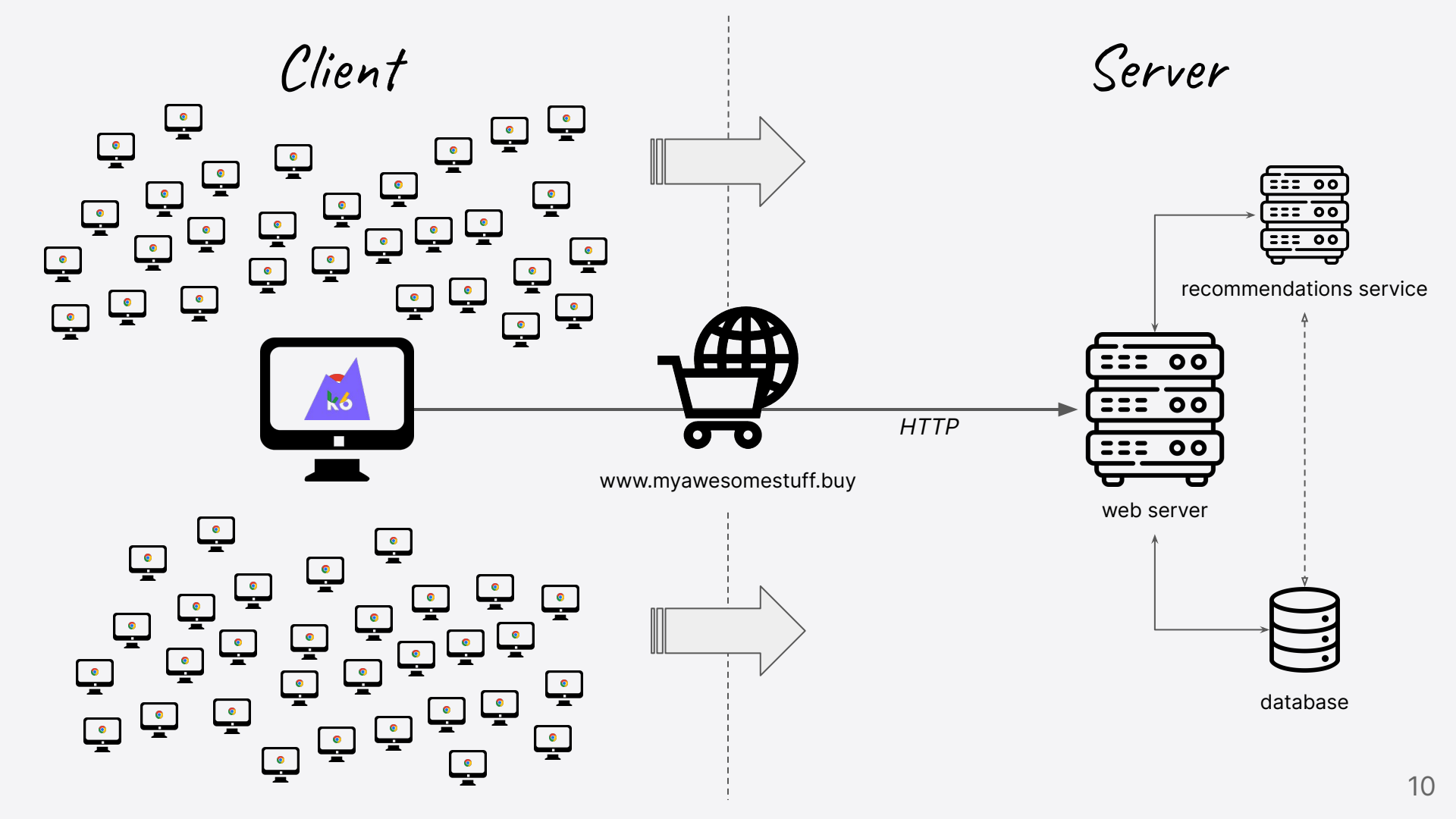
What is load testing?

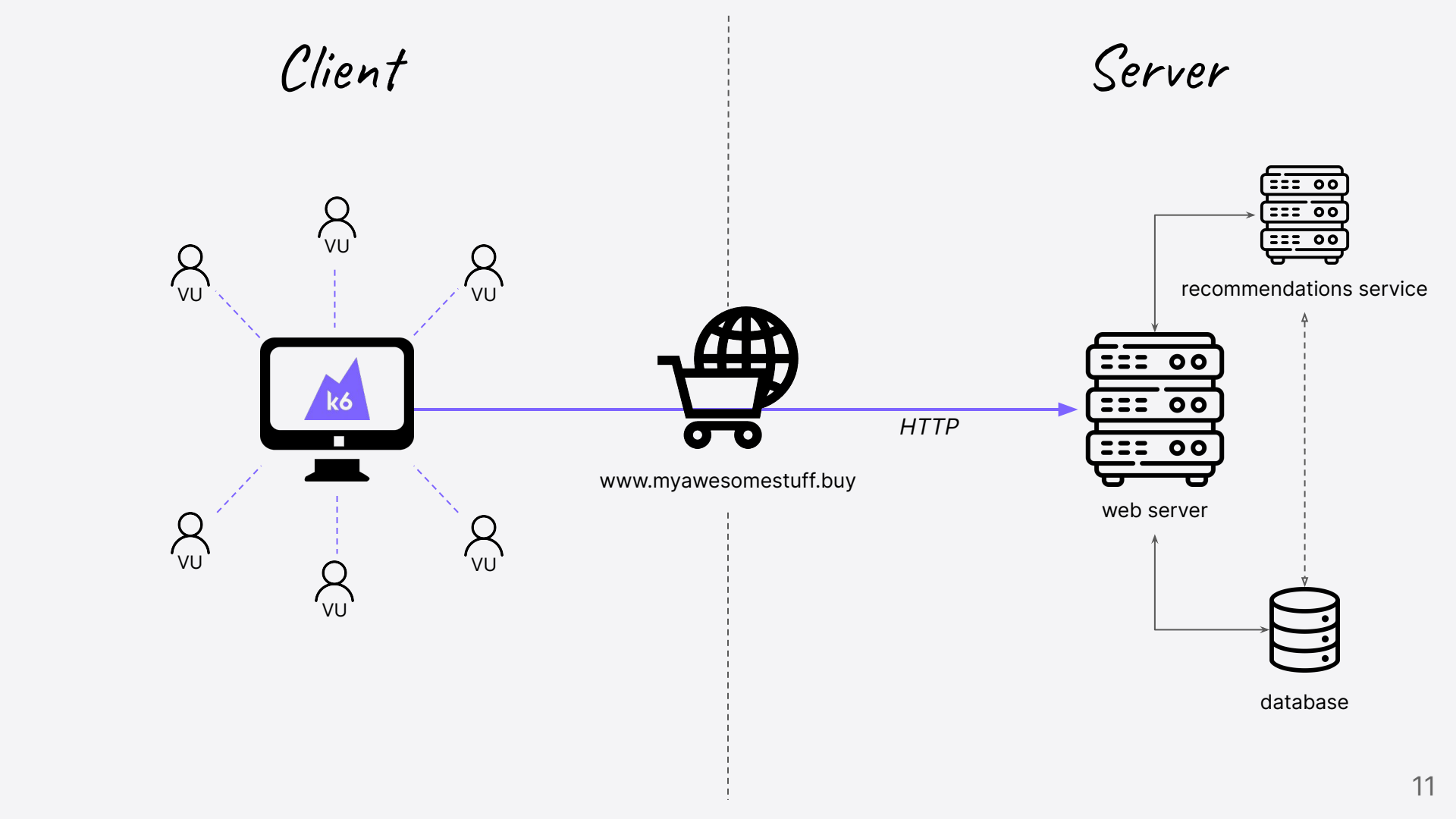
Load = concurrency

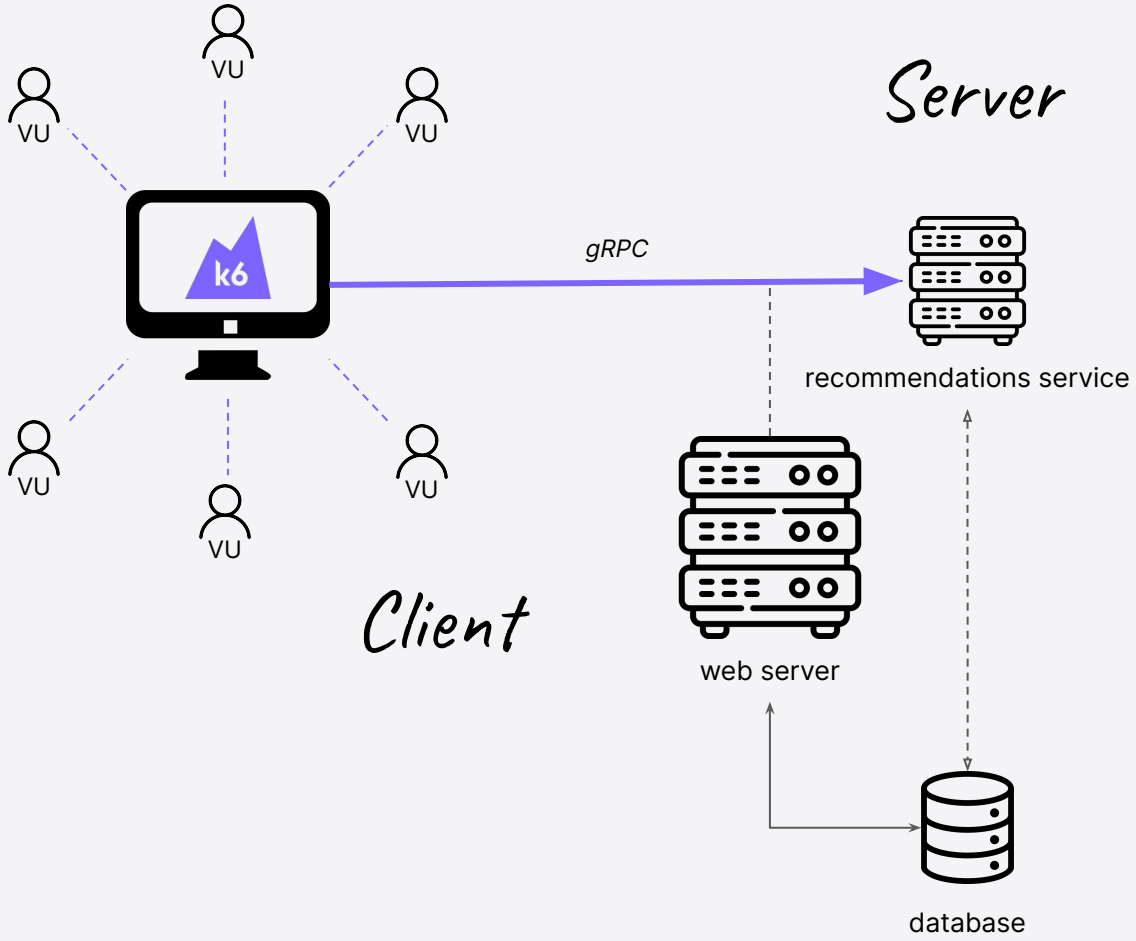
Who does performance testing?


Client
Server
recommendations service
HTTP
www.myawesomestuff.buy
web server
database


Client
Server
recommendations service
HTTP
www.myawesomestuff.buy
web server
database


Client
Server
recommendations service
HTTP
www.myawesomestuff.buy
web server
database


Client
Server
VU
VU
VU
VU
VU
VU
k6
www.myawesomestuff.buy
HTTP
recommendations service
web server
database


VU
VU
VU
VU
VU
VU
k6
gRPC
Server
recommendations service
web server
database
Client

How?

in a nutshell...

- Open source
- Written in Go; single cross-platform binary
- Scripts written in JavaScript
- Command-Line Interface (CLI)
- Supports modern client/server protocols (HTTP, WebSocket, gRPC)

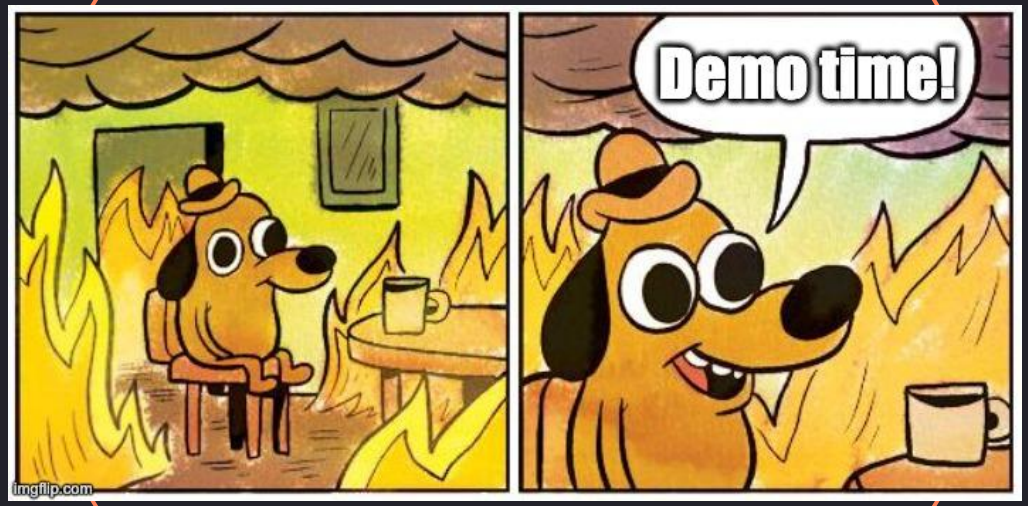
```
// script.js
```

```
import http from 'k6/http'

export default function () {
  http.get('https://test-api.k6.io')
}
```

```
$ k6 run script.js
```

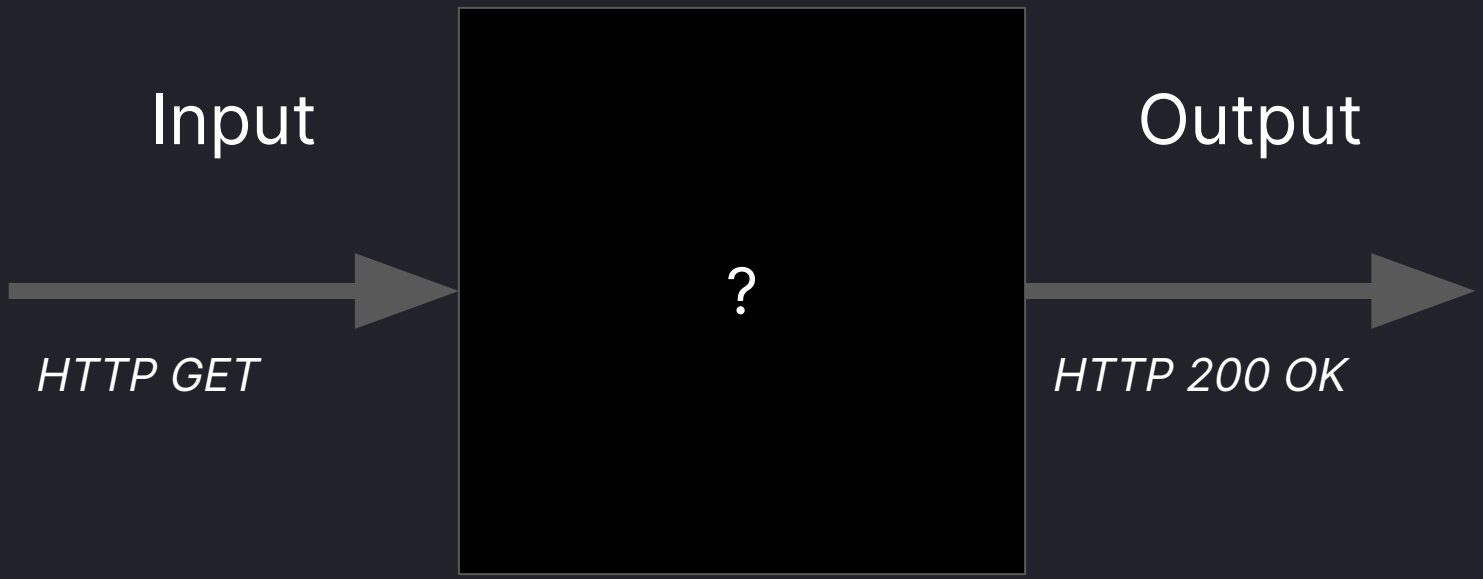
```
     /\      |‾‾| /‾‾/   /‾‾/
  /\  /  \     |  |/  /   /  /
 /  \/    \    |     (   /   ‾‾\
/          \   |  |\  \ |  (‾)  |
/ __________ \  |__| \__\ \_____/ .io

execution: local
   script: temp.js
   output: -

scenarios: (100.00%) 1 scenario, 200 max VUs, 3m30s max duration (incl. graceful stop):
         * default: Up to 200 looping VUs for 3m0s over 2 stages (gracefulRampDown: 30s, gracefulStop: 30s)


   ✓ API response HTTP 200

   checks.........................: 100.00% ✓ 524       ✗ 0
   data_received..................: 1.9 MB  65 kB/s
   data_sent......................: 95 kB   3.2 kB/s
   http_req_blocked...............: avg=8.32ms   min=0s      med=0s       max=96.22ms  p(90)=41.18ms  p(95)=43.84ms
   http_req_connecting............: avg=3.32ms   min=0s      med=0s       max=62.91ms  p(90)=16.83ms  p(95)=17.86ms
 ✓ http_req_duration..............: avg=122.76ms min=79.22ms med=101.05ms max=481.54ms p(90)=174.03ms p(95)=239.19ms
     { expected_response:true }...: avg=122.76ms min=79.22ms med=101.05ms max=481.54ms p(90)=174.03ms p(95)=239.19ms
 ✓ http_req_failed................: 0.00%   ✓ 0         ✗ 524
   http_req_receiving.............: avg=2.45ms   min=0s      med=1.4ms    max=16.99ms  p(90)=5.85ms   p(95)=10.5ms
   http_req_sending...............: avg=66.76µs  min=0s      med=0s       max=705µs    p(90)=375.28µs p(95)=540.49µs
   http_req_tls_handshaking.......: avg=4.63ms   min=0s      med=0s       max=47.34ms  p(90)=23.73ms  p(95)=25.73ms
   http_req_waiting...............: avg=120.24ms min=71.44ms med=98.65ms  max=479.72ms p(90)=171.89ms p(95)=237.33ms
   http_reqs......................: 524     17.487089/s
   iteration_duration.............: avg=3.12s    min=1.08s   med=3.1s     max=5.53s    p(90)=5.12s    p(95)=5.15s
   iterations.....................: 427     14.249976/s
   vus............................: 97      min=4       max=97
   vus_max........................: 200     min=200     max=200
```

```
import http from 'k6/http'
import { check } from 'k6'

export const options = {
  stages: [
    { duration: '1m', target: 200 }, // ramp up to 200 VUs over 1 minute
    { duration: '2m', target: 200 }, // hold steady at 200 VUs for 2 minutes
  ],
  thresholds: {
    http_req_failed: ['rate<0.01'], // http errors should be less than 1%
    http_req_duration: ['p(95)<5000'], // 95% of requests should be below 5000ms
  }
}

export default function () {
  const response = http.get('https://test-api.k6.io') // make GET request and store the response

  // verify that the response status code was HTTP 200
  check(response, {
    'API response HTTP 200': (r) => r.status === 200,
  })
}
```


Demo time!
imgflip.com

Instrumentation and Load Testing

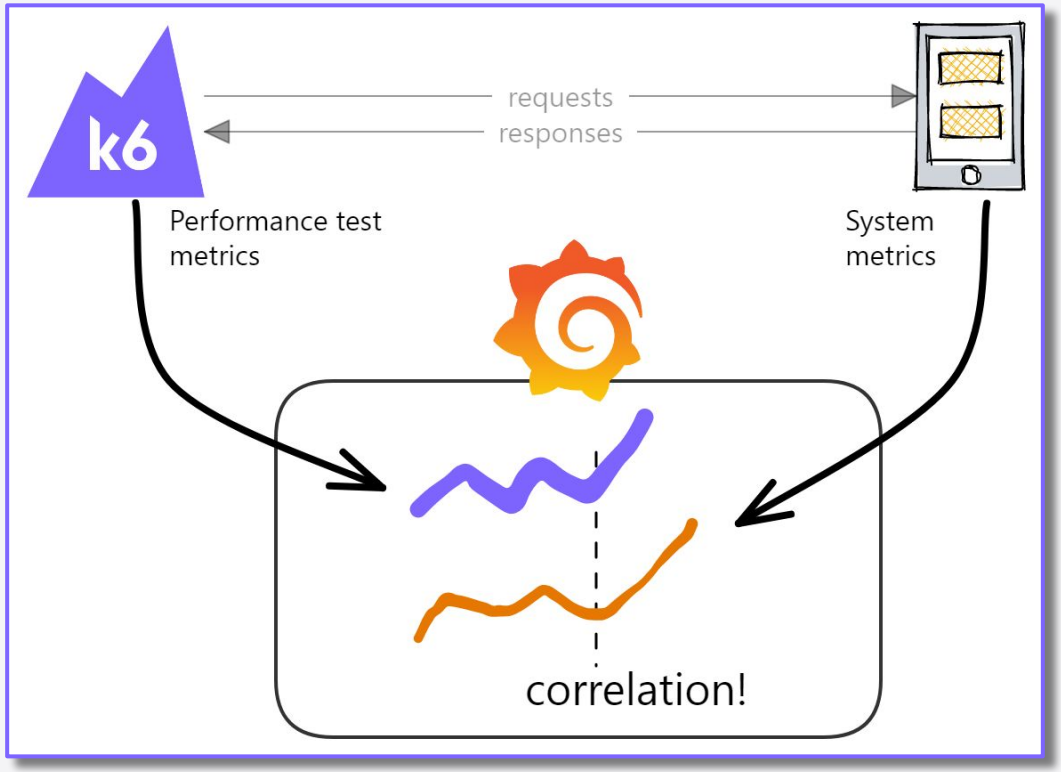
The Black Box



The Black Box

The solution: **Instrumentation!**

Metric correlation



Shifting Left

There is more

- K6 Extensions: https://k6.io/docs/extensions/
- K6 browser recorder
- k6 operator: https://github.com/grafana/k6-operator
- k6 cloud: https://grafana.com/products/cloud/k6/

Further learning

- Self-paced learning: https://github.com/grafana/k6-learn
- k6 OSS docs: https://docs.k6.io

Q&A

Thank you!