Continuous Performance Testing

can your app still take it?

Christian Kühn

Software - Developer

always happy

interested:
Security, DevOops, Automation

dmTECH GmbH

100% dm subsidiary

complete corporate-IT
Hardware
Software
Operations

why performance-Tests?

better estimation of load balancing in productive environments

"preview" how the app might handle load

become confident in your app

(but mostly save money)

why performance-Tests?

find bottlenecks

save money (again)

increase customer happiness

measuring performance: what?

bad requests

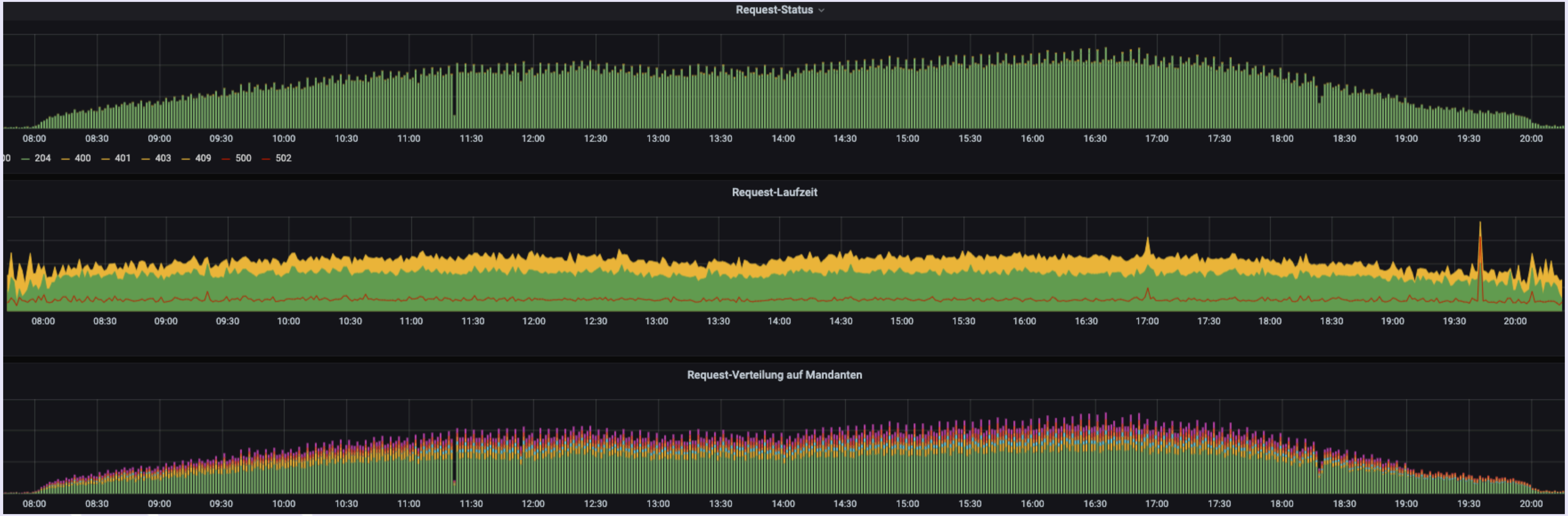
slow requests

app crashes

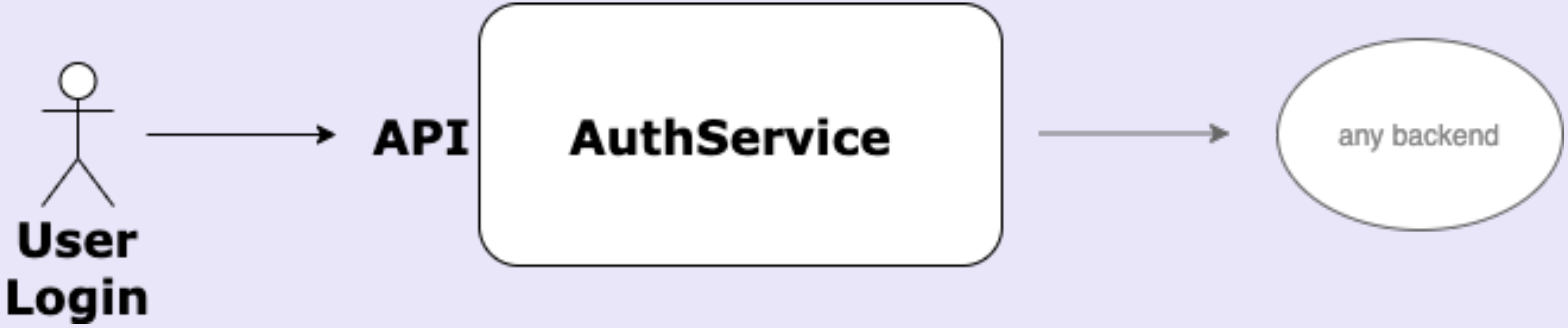
measuring performance: how?

Metriken!!

metrics - dashboards - alerting



imagine this application



Gatling

load test as code

alternatives

k6

JMeter

ab / yes

SOAPUI

...

Gatling

"Load Test as Code"

Scala - based (Akka)

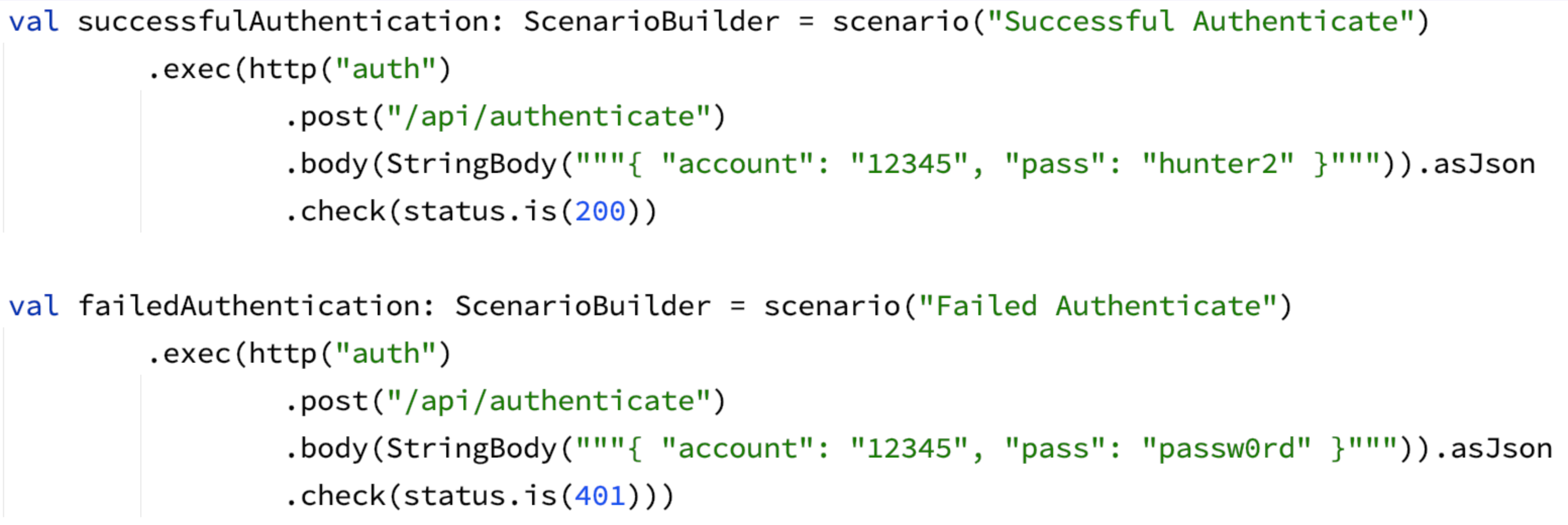
HTTP-perf testing with Netty (REST, SOAP)

can report metrics (graphite)

https://gatling.io

Scenario

test case description

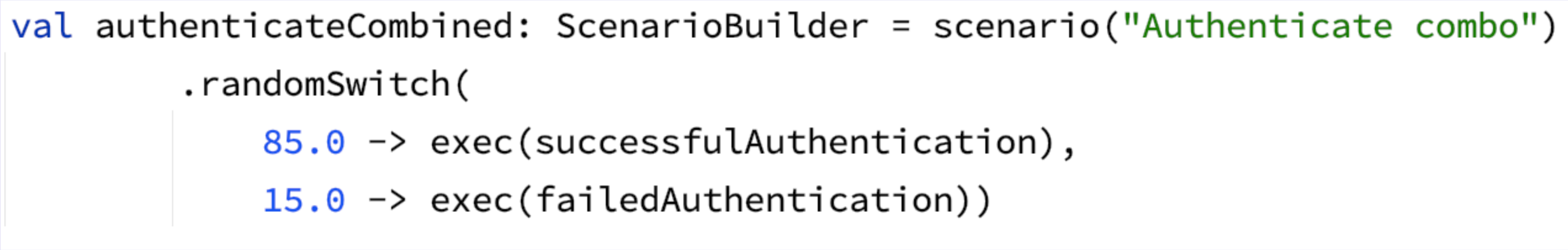


```
val successfulAuthentication: ScenarioBuilder = scenario("Successful Authenticate")
        .exec(http("auth")
                .post("/api/authenticate")
                .body(StringBody("""{ "account": "12345", "pass": "hunter2" }""")).asJson
                .check(status.is(200))

val failedAuthentication: ScenarioBuilder = scenario("Failed Authenticate")
        .exec(http("auth")
                .post("/api/authenticate")
                .body(StringBody("""{ "account": "12345", "pass": "passw0rd" }""")).asJson
                .check(status.is(401)))
```

Scenario (2)

combination of scenarios



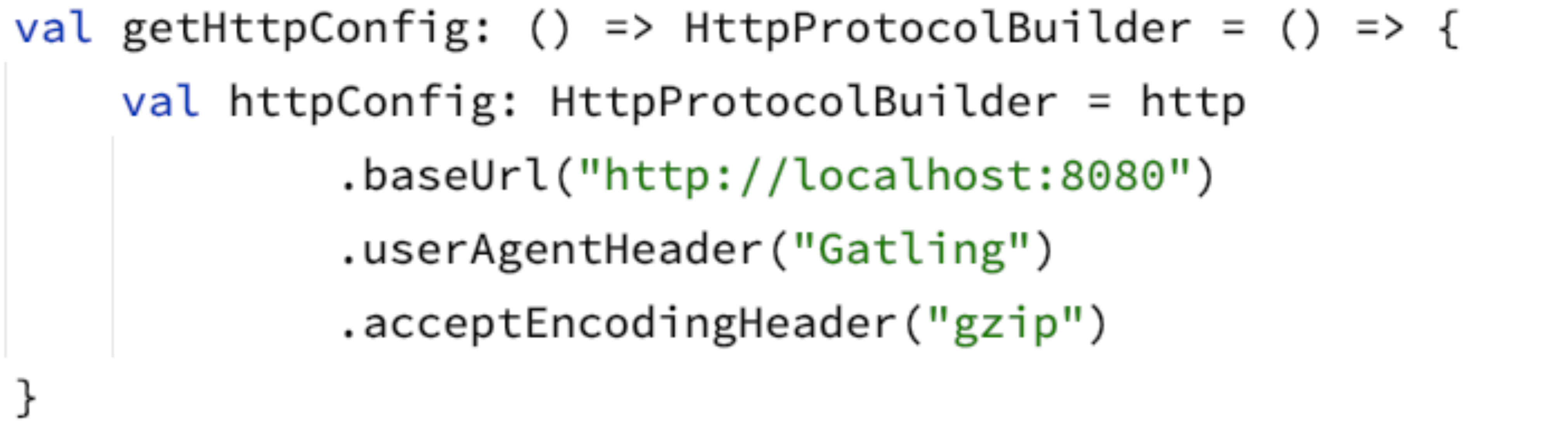
```
val authenticateCombined: ScenarioBuilder = scenario("Authenticate combo")
        .randomSwitch(
            85.0 -> exec(successfulAuthentication),
            15.0 -> exec(failedAuthentication))
```

Simulation

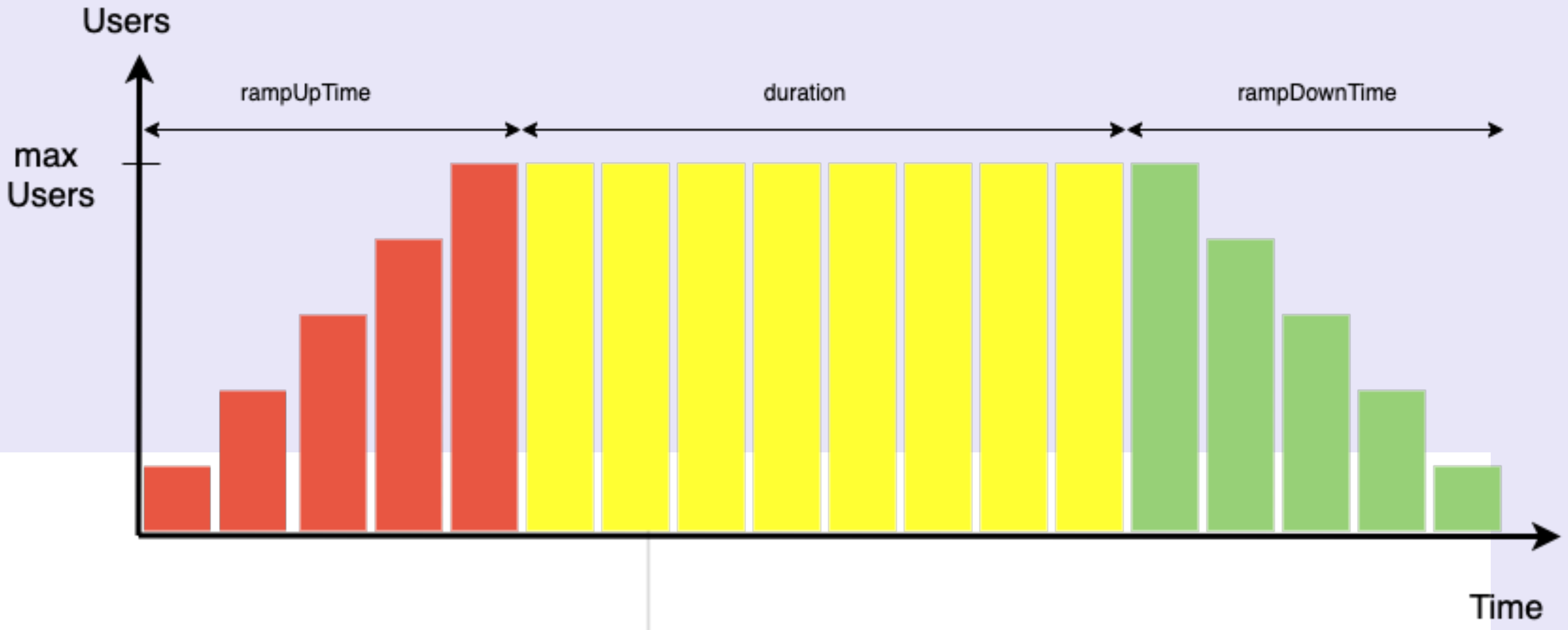
basic test setup

- amount of users / clients
- httpConfig
- rampUp / -down
- duration

Simulation (example)



```
val getHttpConfig: () => HttpProtocolBuilder = () => {
    val httpConfig: HttpProtocolBuilder = http
            .baseUrl("http://localhost:8080")
            .userAgentHeader("Gatling")
            .acceptEncodingHeader("gzip")
}
```



```
setUp(
  authenticateCombined.inject(
    rampUsersPerSec( rate1 = 0) to 30 during (10 seconds), // RAMP UP, 3 additional requests per second)
    constantUsersPerSec( rate = 30) during(30 seconds) randomized, // 20 requests per second for 30 seconds
    rampUsersPerSec( rate1 = 30) to 0 during (10 seconds)) // RAMP Down, 3 less requests per second
).protocols(httpConfig)
  .assertions(
    global.failedRequests.percent.lte( threshold = 0.05), // under 5% wrong answers
    global.responseTime.percentile4.lte( threshold = 1000) // 99% of requests answered quicker than 1000ms
  )
```

demo

report -> see html, grafana

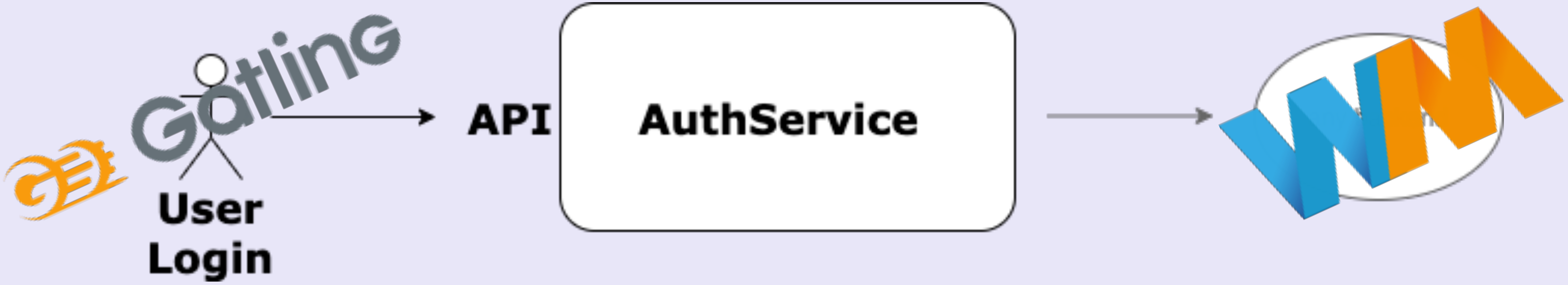
"continuous testing"

Merge-Requests -> always (?!)

"release"-System -> always :-) (and periodically)

Prod -> periodically if possible

app backed by third-party service



Wiremock

API - Simulator (HTTP)

prepared responses
depending on request

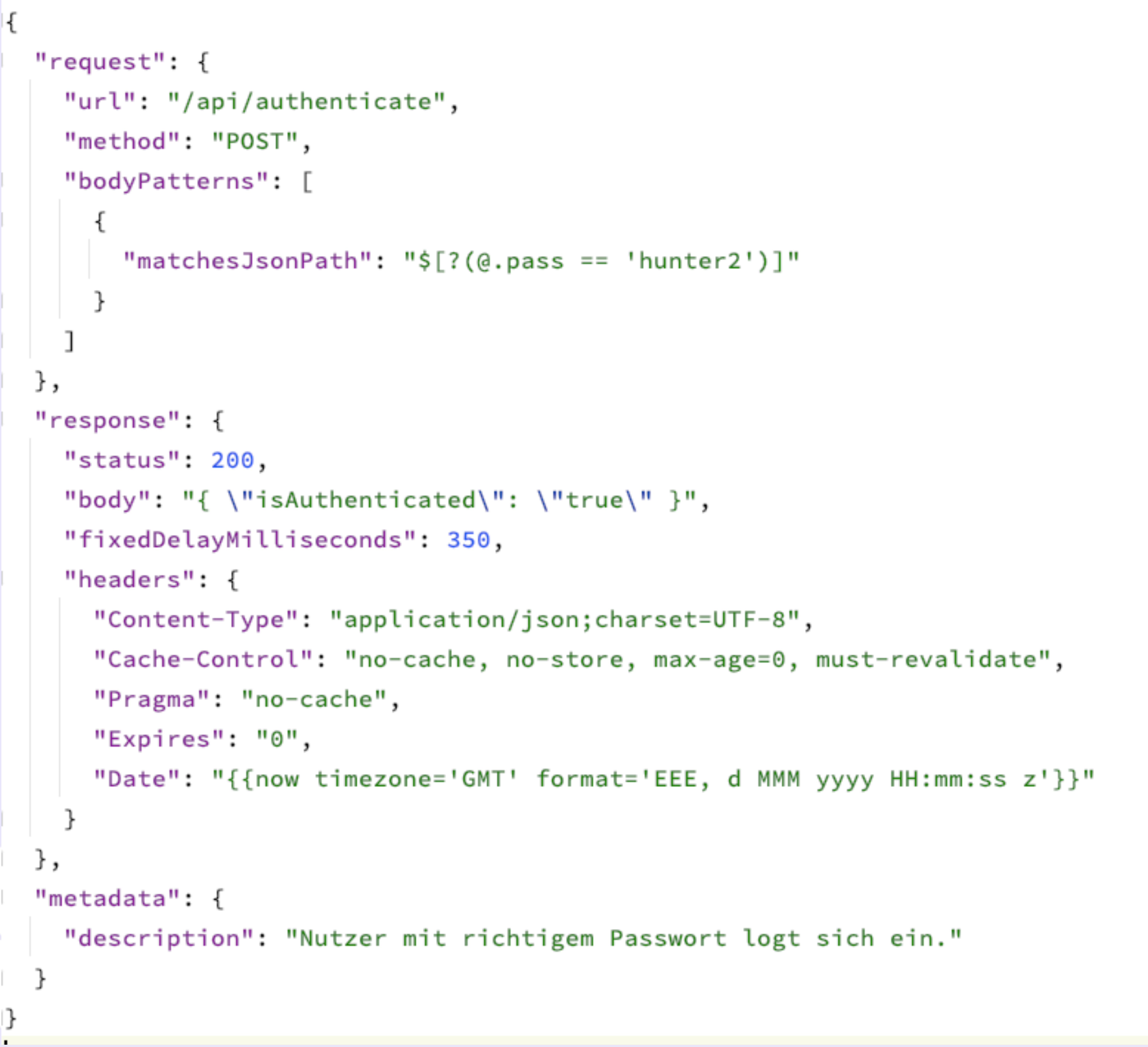
record / playback

JUnit or standalone

mock http-services

mapping on request-specifica
(e.g. certain headers)

configuration in java or json

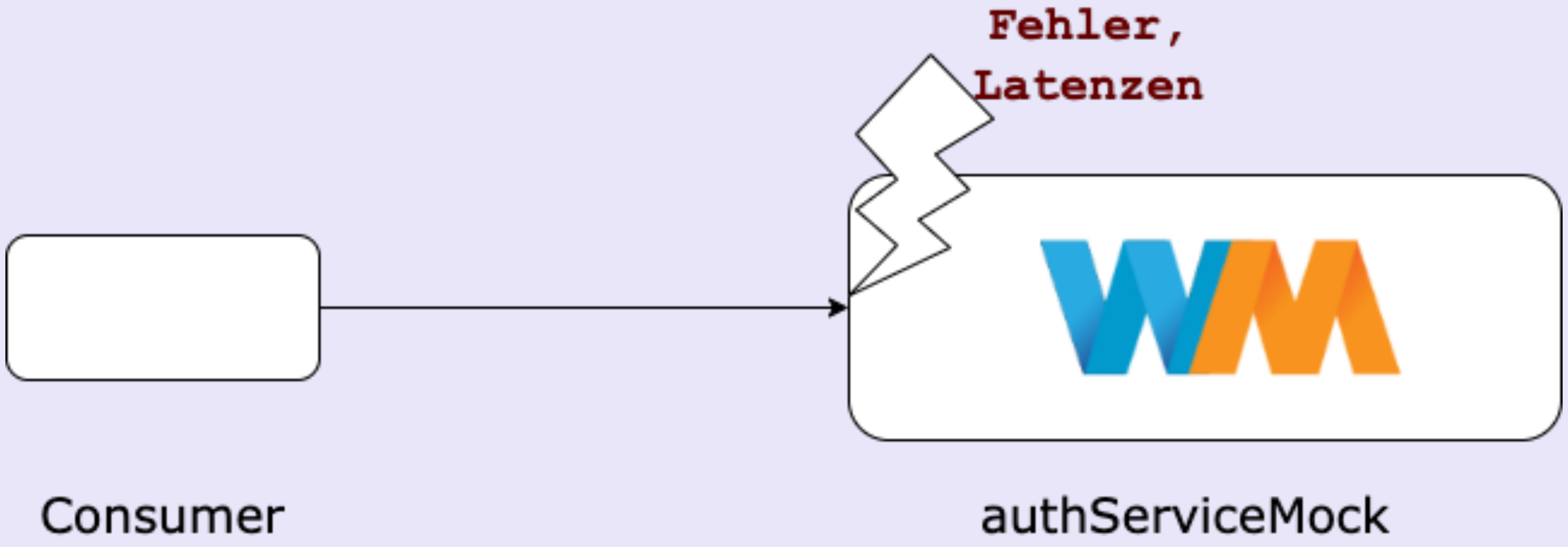


```
{
  "request": {
    "url": "/api/authenticate",
    "method": "POST",
    "bodyPatterns": [
      {
        "matchesJsonPath": "$[?(@.pass == 'hunter2')]"
      }
    ]
  },
  "response": {
    "status": 200,
    "body": "{ \"isAuthenticated\": \"true\" }",
    "fixedDelayMilliseconds": 350,
    "headers": {
      "Content-Type": "application/json;charset=UTF-8",
      "Cache-Control": "no-cache, no-store, max-age=0, must-revalidate",
      "Pragma": "no-cache",
      "Expires": "0",
      "Date": "{{now timezone='GMT' format='EEE, d MMM yyyy HH:mm:ss z'}}"
    }
  },
  "metadata": {
    "description": "Nutzer mit richtigem Passwort logt sich ein."
  }
}
```

test your own client for third-party APIs

simulate errors and latency, timeouts..

sandbox simulating your own service



for other teams/customers that consume your app scenarios can have multiple stages

"admin"-API for control

thanks for joining

questions?

happy about any Feedback #slack 🐦@ChristianKuehn