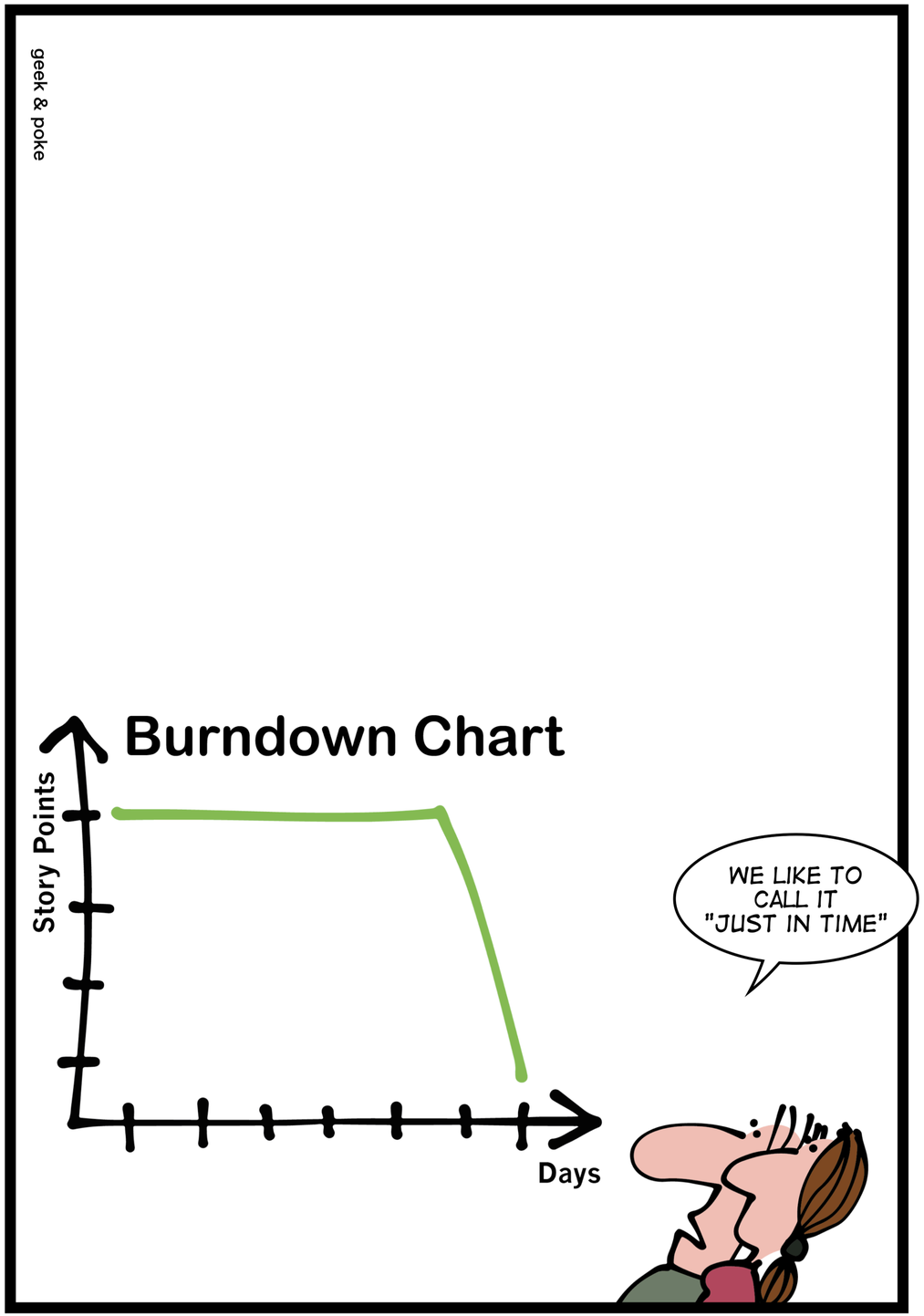
# Deployen op vrijdagmiddag zonder stress

## Using Testcontainers, Playwright & Azure DevOps

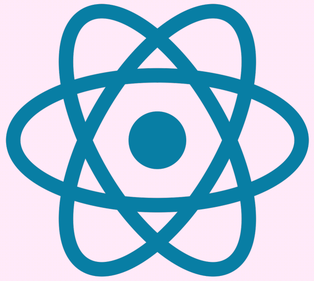
Bas Stoker, 11 april 2024

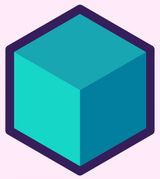

geek & poke
Burndown Chart
Story Points
Days
WE LIKE TO CALL IT "JUST IN TIME"

# About me

DEV TALENTS

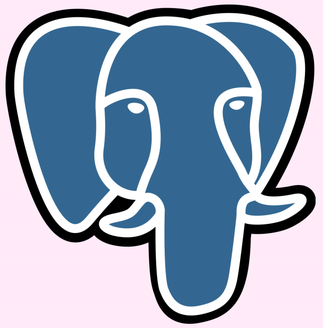
- Fullstack developer bij de NS
- 20 jaar ervaring met software development
- Enthousiast over toekomst van frontend-development en test-technieken in het bijzonder

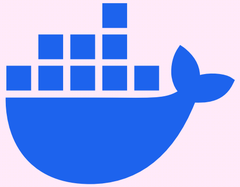




Testcontainers

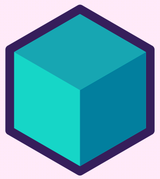
Java™



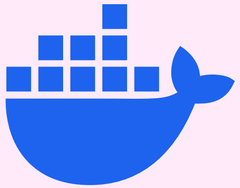


docker®


Playwright



Testcontainers



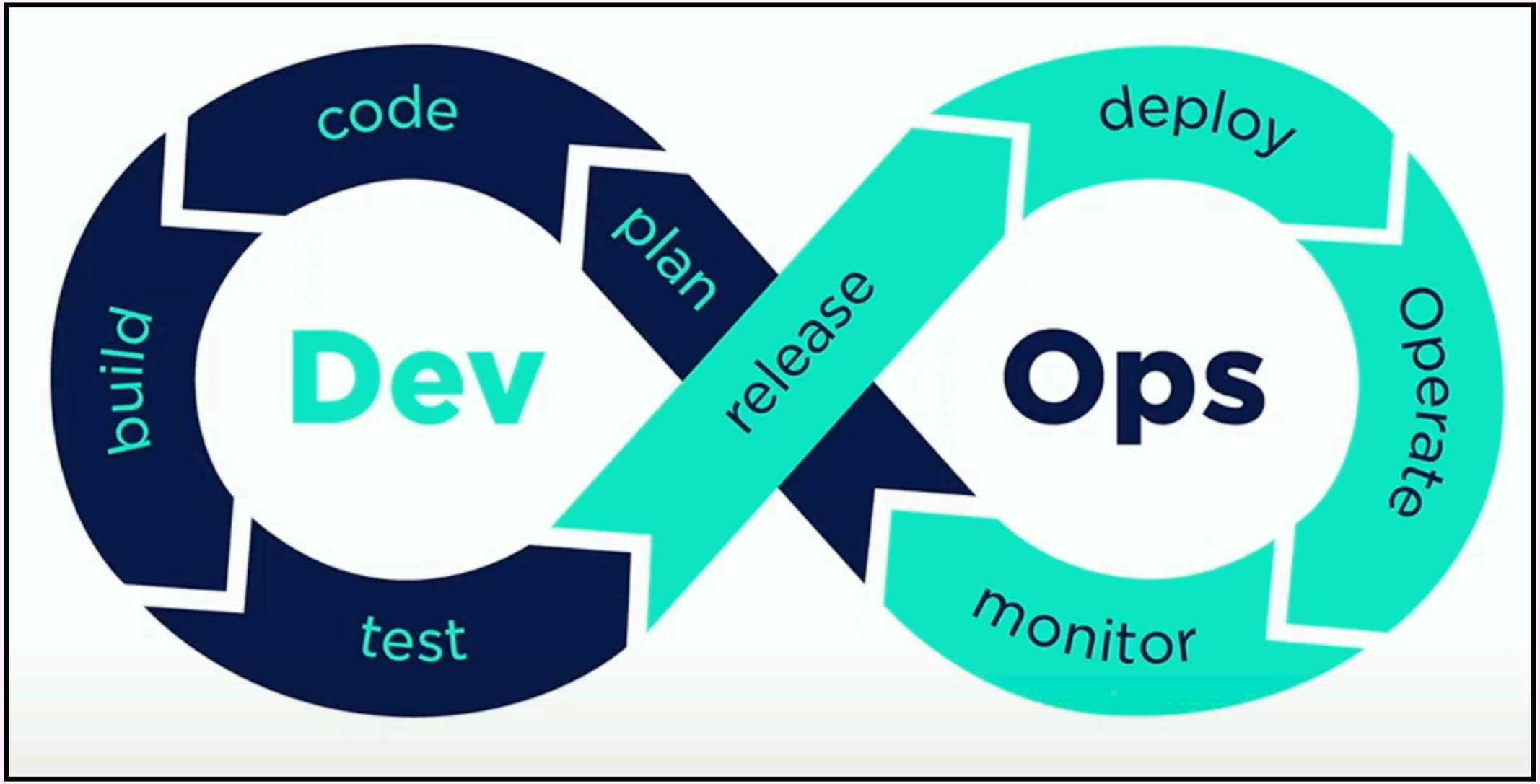
docker®

Java™

Playwright

# Inhoud

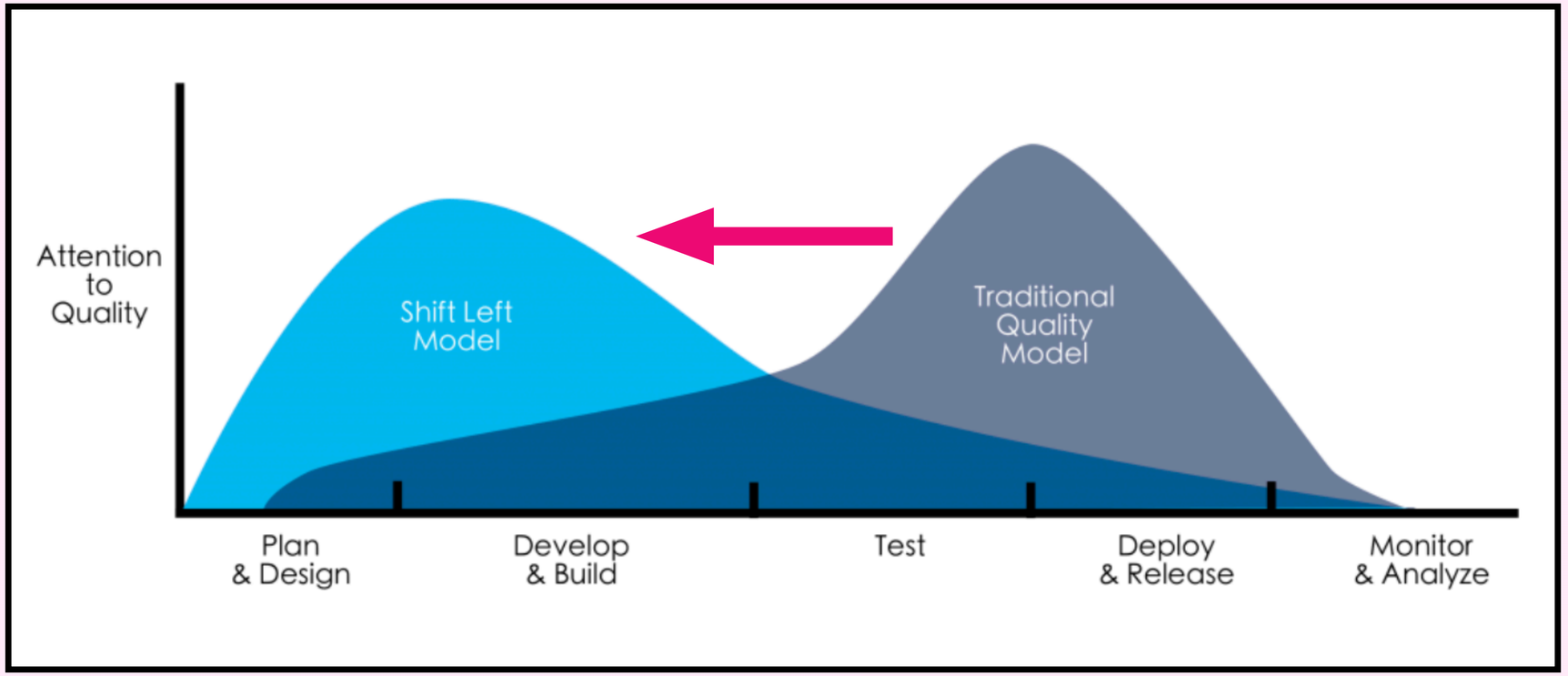
- Trends in testing
  - CI/CD en DevOps
  - Testcontainers
  - Playwright
- Case-study: NS Bijsturing Materieel (BAM)
- Live demo systeemtest ⚡
- Tips & Tricks

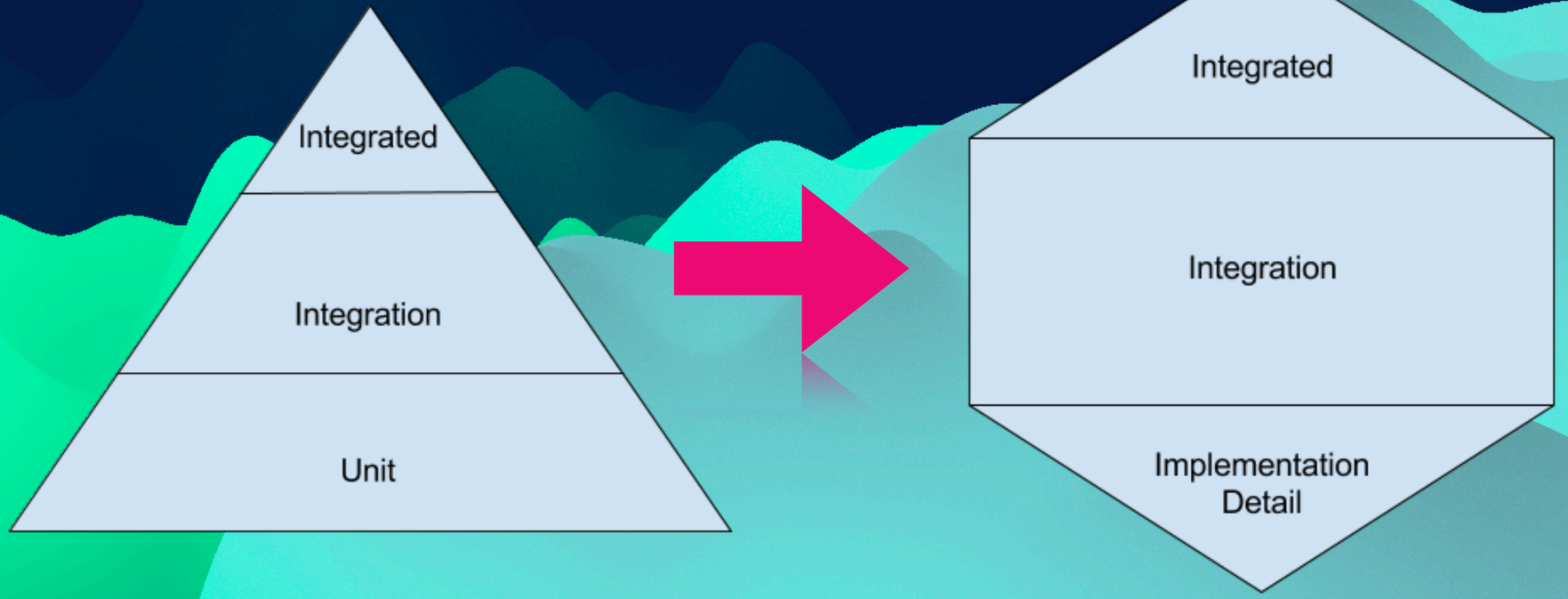

code
plan
build
Dev
test
release
deploy
Ops
operate
monitor

# Continuous Deployment

## Continuous testing?

- Veel verantwoordelijkheid bij het development-team
- Vaak geen dedicated testers meer
- Resultaat: Alle testen zijn geautomatiseerd


Attention
to
Quality
Shift Left
Model
Traditional
Quality
Model
Plan
& Design
Develop
& Build
Test
Deploy
& Release
Monitor
& Analyze

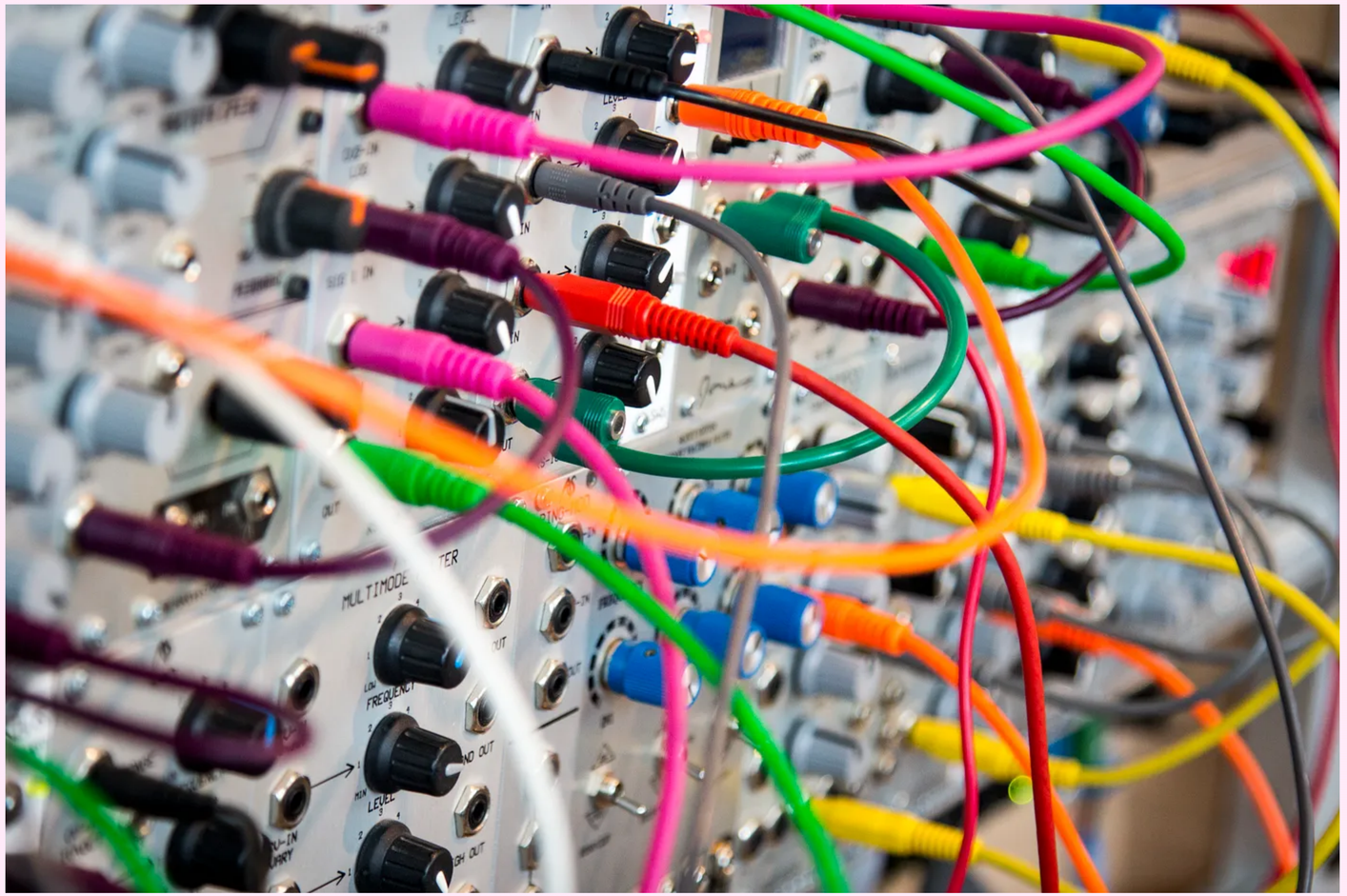

Integrated
Integration
Unit
Integrated
Integration
Implementation
Detail

# “Write tests. Not too many. Mostly Integration”

- Quote van Kent C. Dodds
- Hoger in de testpyramide wordt het vertrouwen in de test hoger
- Bestaande aannames dat e2e-tests duur zijn om te onderhouden kloppen niet meer

https://kentcdodds.com/blog/write-tests

# “You’re probably mocking too much”




https://clayshentrup.medium.com/tests-dependencies-65f592a46529

# Testcontainers

**Alternatief voor…**

- In-memory alternatieven zoals H2 als database
- Shell scripts
- Docker compose
- Zelf vanuit Java Docker-api aanroepen (veel werk)

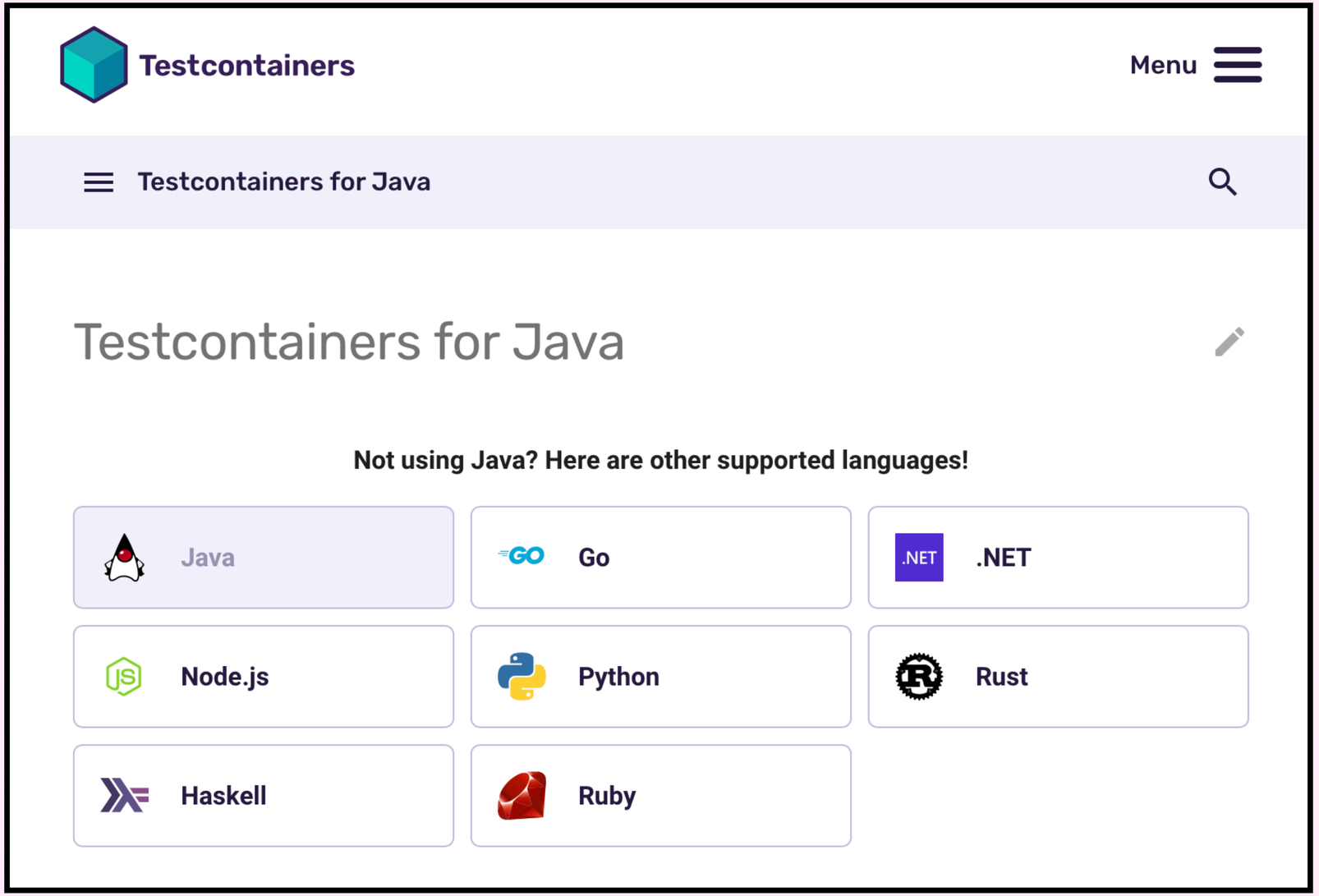
# Testcontainers

**Testen met echte dependencies**

- Docker Containers kennen we al van o.a. Kubernetes
- Open Source library, begonnen als Java library
- Totale lifecycle van Docker containers:
  - Startup
  - Cleanup

```
public static PostgreSQLContainer<?> POSTGRES_CONTAINER
  = new PostgreSQLContainer<>(DockerImageName.parse("postgres:14"))
        .withDatabaseName(
        "integration-tests-db")
        .withUsername("admin")
        .withPassword("admin")
        .withInitScript("create_schema_and_users.sql")
        .withNetworkAliases("postgres")
        .withCommand(new String[]{"postgres", "-c", "fsync=off", "-c", "wal_level=logical", "-c",
                                  "max_replication_slots=1"});

static {
    POSTGRES_CONTAINER.start();
}

@Override
public void initialize(final @NotNull ConfigurableApplicationContext applicationContext) {
    TestPropertySourceUtils.addInlinedPropertiesToEnvironment(applicationContext,
            "spring.datasource.url=" + POSTGRES_CONTAINER.getJdbcUrl(),
            "spring.datasource.username=" + POSTGRES_CONTAINER.getUsername(),
            "spring.datasource.password=" + POSTGRES_CONTAINER.getPassword());
}
```

# Testcontainers

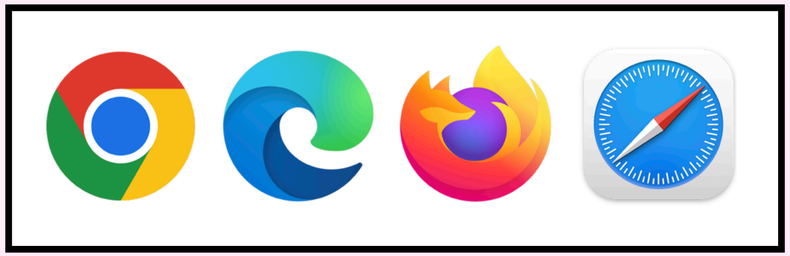


# Veel bestaande containers

Testcontainers →

Docker →

# Playwright



## Any browser • Any platform • One API

**Cross-browser.** Playwright supports all modern rendering engines including Chromium, WebKit, and Firefox.

**Cross-platform.** Test on Windows, Linux, and macOS, locally or on CI, headless or headed.

**Cross-language.** Use the Playwright API in TypeScript, JavaScript, Python, .NET, Java.

**Test Mobile Web.** Native mobile emulation of Google Chrome for Android and Mobile Safari. The same rendering engine works on your Desktop and in the Cloud.

## Resilient • No flaky tests

**Auto-wait.** Playwright waits for elements to be actionable prior to performing actions. It also has a rich set of introspection events. The combination of the two eliminates the need for artificial timeouts - the primary cause of flaky tests.

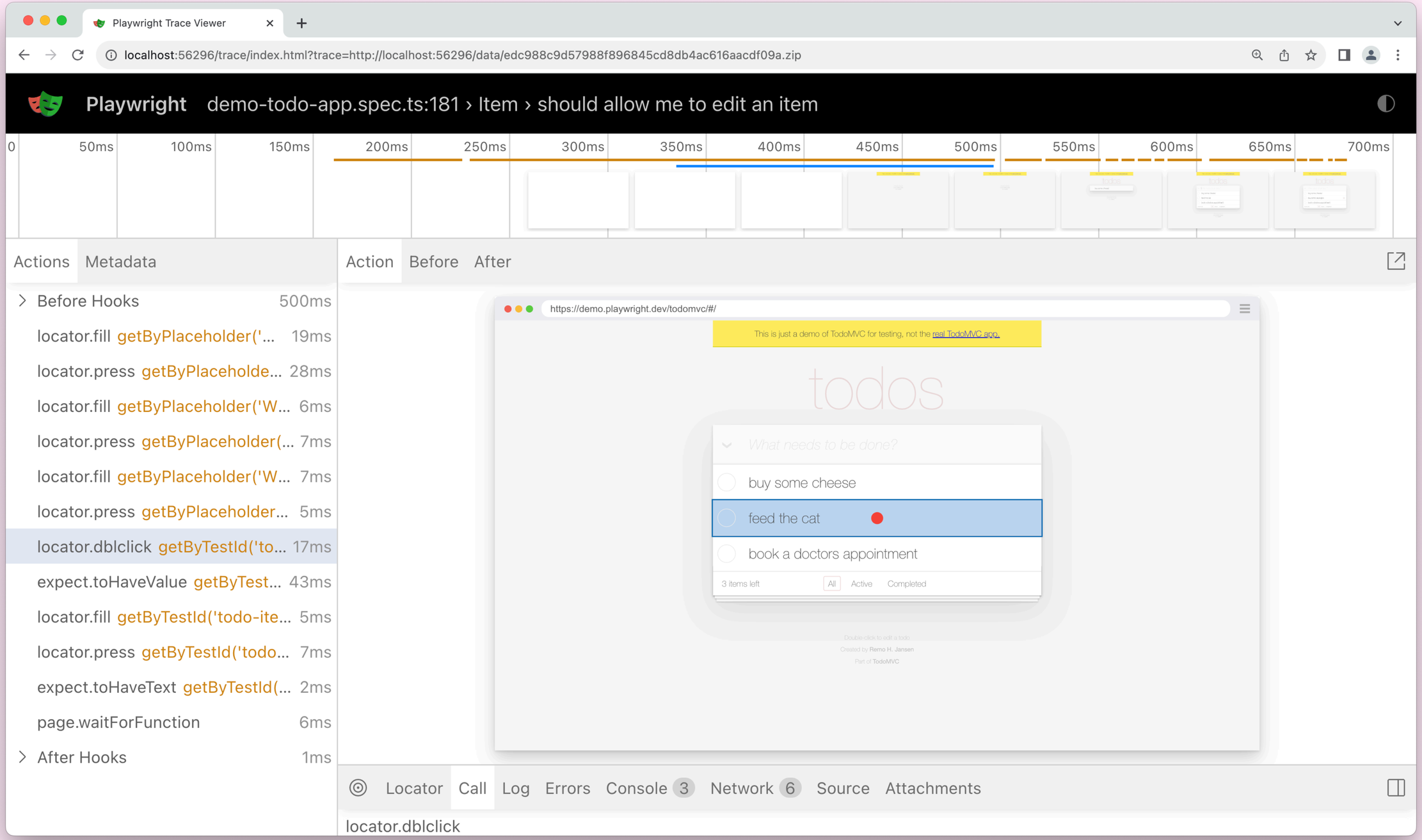
**Web-first assertions.** Playwright assertions are created specifically for the dynamic web. Checks are automatically retried until the necessary conditions are met.

**Tracing.** Configure test retry strategy, capture execution trace, videos, screenshots to eliminate flakes.

# Playwright

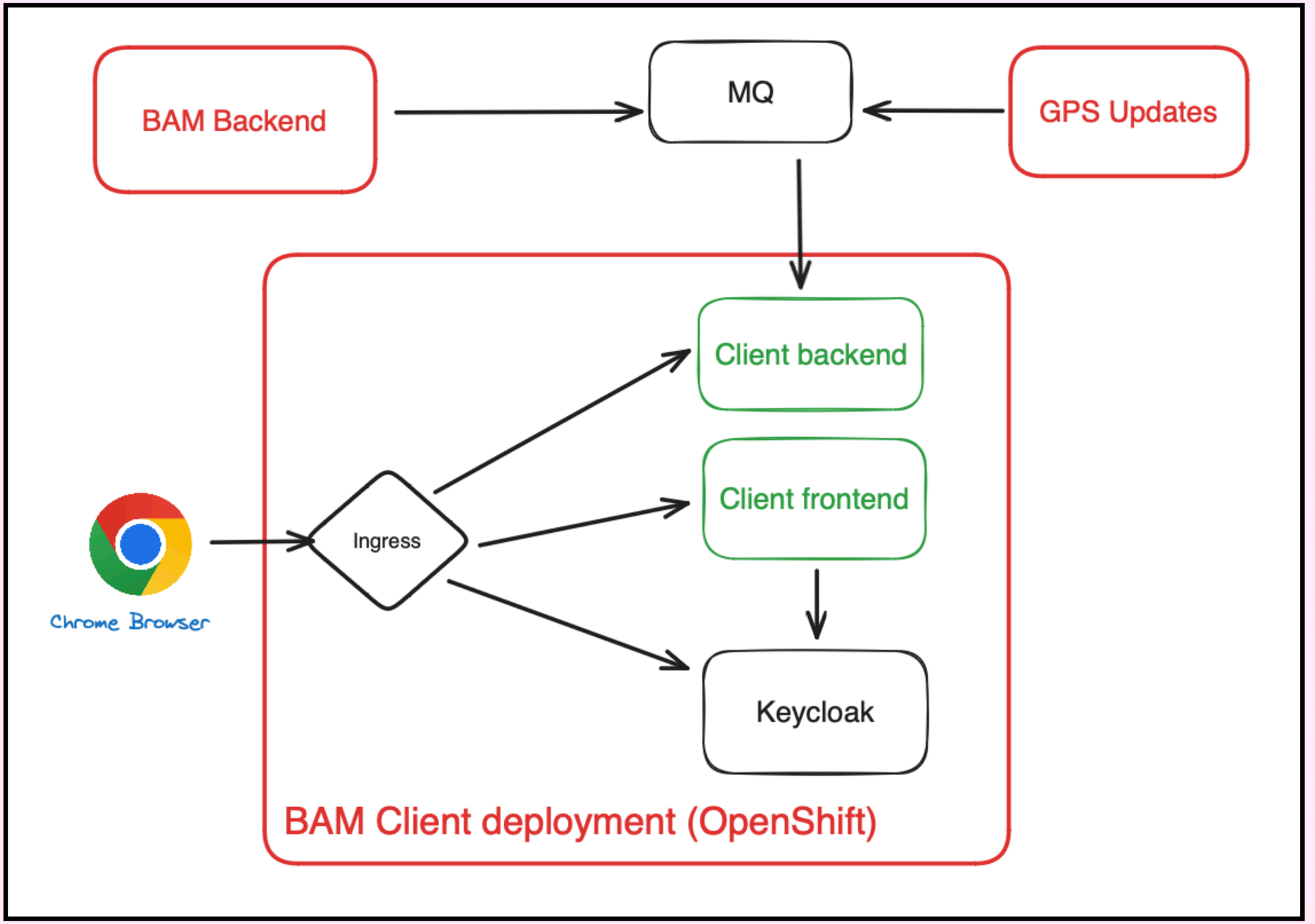
- Zit een supergoed team achter van Microsoft
- Support voor MacOS / Linux / Windows
  - Browsers worden automatisch gemanaged
- Sneller en minder flaky dan bijv. Selenium
- Handige functies zoals trace, slomo, en video/screenshot opnames

```
BrowserType.LaunchOptions chromiumlaunchOptions
  = new BrowserType.LaunchOptions()
        .setProxy(new Proxy(proxyUrl))
        .setHeadless(true)
       .setTracesDir(tracesPath);

Browser browser =
Playwright.create().chromium().launch(chromiumlaunchOptions);
BrowserContext context = browser.newContext(new Browser.NewContextOptions()
        .setViewportSize(BROWSER_WIDTH, BROWSER_HEIGHT)
        .setTimezoneId("Europe/Amsterdam"));

Page page = context.newPage();

page.navigate("http://localhost:8080");

//

@Test
void statusBecomesSubmitted() {
  page.locator("#submit-button").click();
  assertThat(page.locator(".status")).hasText("Submitted"); // auto-waits
}
```


Playwright Trace Viewer
localhost:56296/trace/index.html?trace=http://localhost:56296/data/edc988c9d57988f896845cd8db4ac616aacdf09a.zip
Playwright
demo-todo-app.spec.ts:181 › Item › should allow me to edit an item
0
50ms
100ms
150ms
200ms
250ms
300ms
350ms
400ms
450ms
500ms
550ms
600ms
650ms
700ms
Actions
Metadata
Before Hooks 500ms
locator.fill getByPlaceholder('... 19ms
locator.press getByPlaceholde... 28ms
locator.fill getByPlaceholder('W... 6ms
locator.press getByPlaceholder(... 7ms
locator.fill getByPlaceholder('W... 7ms
locator.press getByPlaceholder... 5ms
locator.dblclick getByTestId('to... 17ms
expect.toHaveValue getByTest... 43ms
locator.fill getByTestId('todo-ite... 5ms
locator.press getByTestId('todo... 7ms
expect.toHaveText getByTestId(... 2ms
page.waitForFunction 6ms
After Hooks 1ms
Action
Before
After
https://demo.playwright.dev/todomvc/#/
This is just a demo of TodoMVC for testing, not the real TodoMVC app.
todos
What needs to be done?
buy some cheese
feed the cat
book a doctors appointment
3 items left
All
Active
Completed
Locator
Call
Log
Errors
Console 3
Network 6
Source
Attachments
locator.dblclick

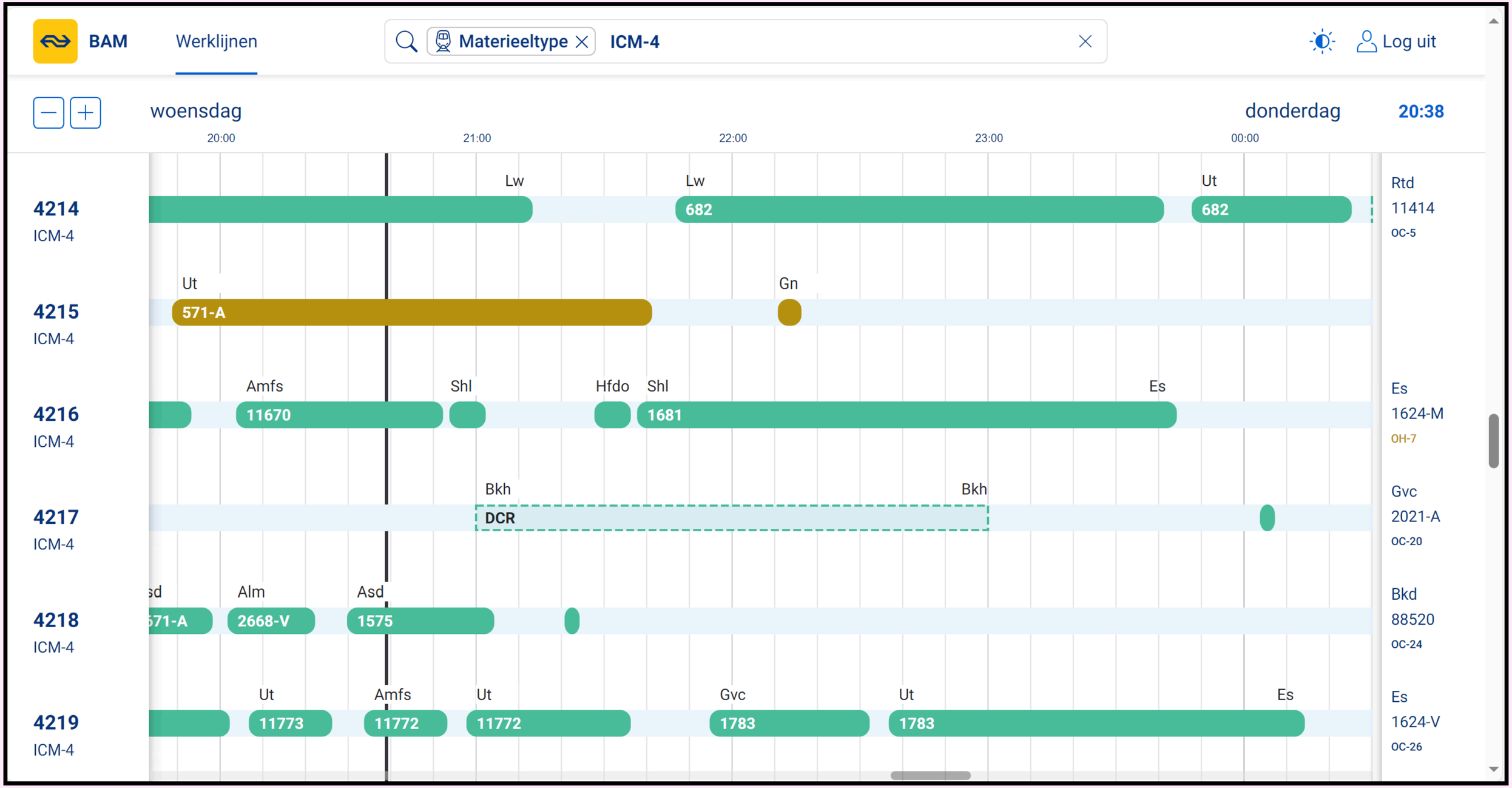
# Project: NS nieuwbouw BAM

- Realtime bijsturen van al het materieel (Intercity / Sprinters / Internationaal)
- Project van vier agile DevOps teams
- Microservices-architectuur
- Meeste data asynchroon via queues:
  - GPS-vertragingsdata van treinen
  - Wijzigingen in de materieelplanning


BAM Backend
MQ
GPS Updates
Client backend
Client frontend
Keycloak
Ingress
Chrome Browser
BAM Client deployment (OpenShift)

# BAM Client

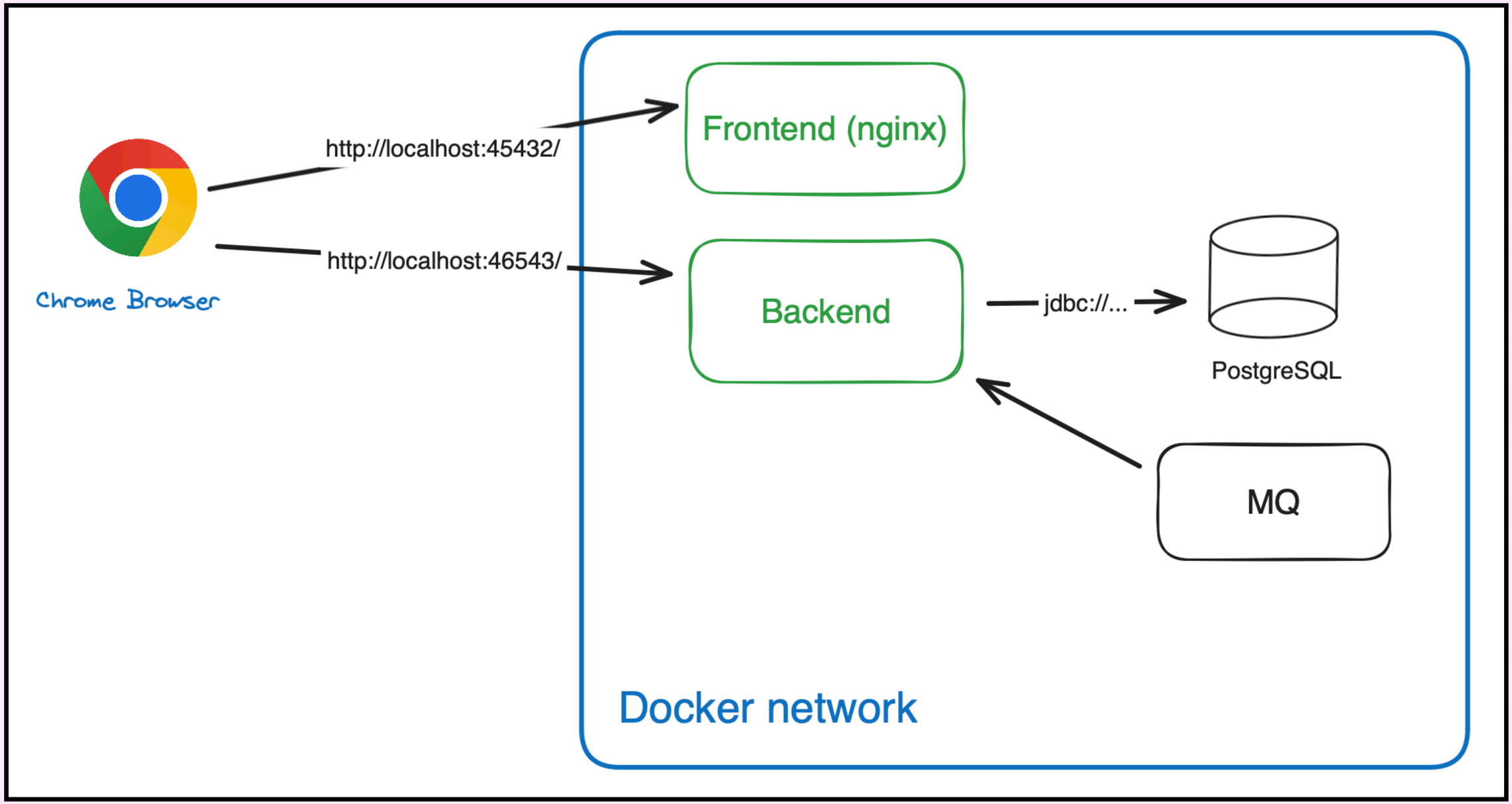
- Verantwoordelijkheid van 1 team
- Bestaat uit twee microservices
  - Backend for frontend (ontvangen MQ-berichten, caching etc…)
  - Frontend (ReactJS SPA, ontsloten via NGINX web server)

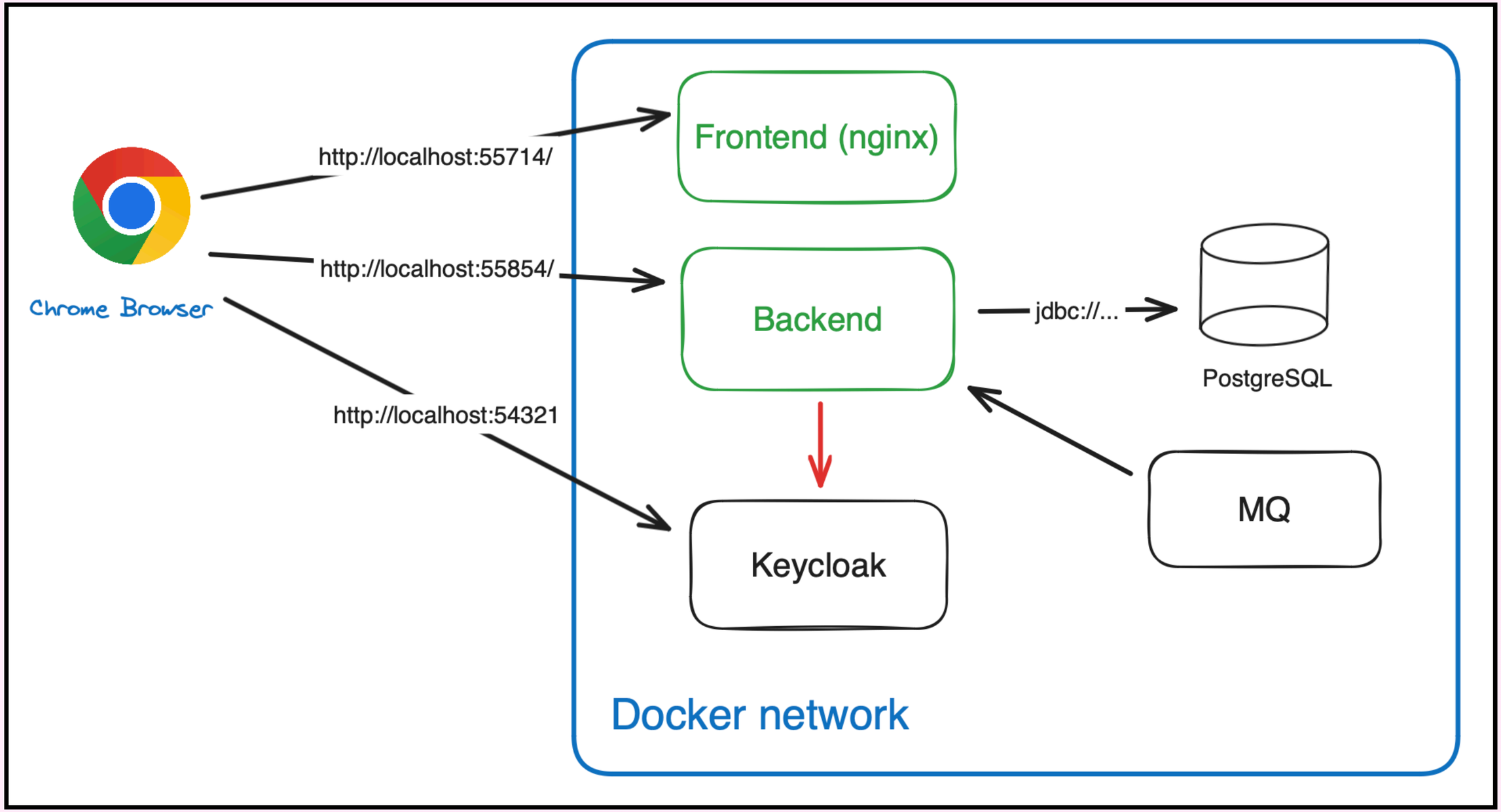

BAM
Werklijnen
Materieeltype
ICM-4
Log uit
woensdag
donderdag
20:38
20:00
21:00
22:00
23:00
00:00
4214
ICM-4
Lw
Lw
Ut
682
682
Rtd
11414
OC-5
4215
ICM-4
Ut
Gn
571-A
4216
ICM-4
Amfs
Shl
Hfdo
Shl
Es
11670
1681
Es
1624-M
OH-7
4217
ICM-4
Bkh
Bkh
DCR
Gvc
2021-A
OC-20
4218
ICM-4
Alm
Asd
2668-V
1575
Bkd
88520
OC-24
4219
ICM-4
Ut
Amfs
Ut
Gvc
Ut
Es
11773
11772
11772
1783
1783
Es
1624-V
OC-26

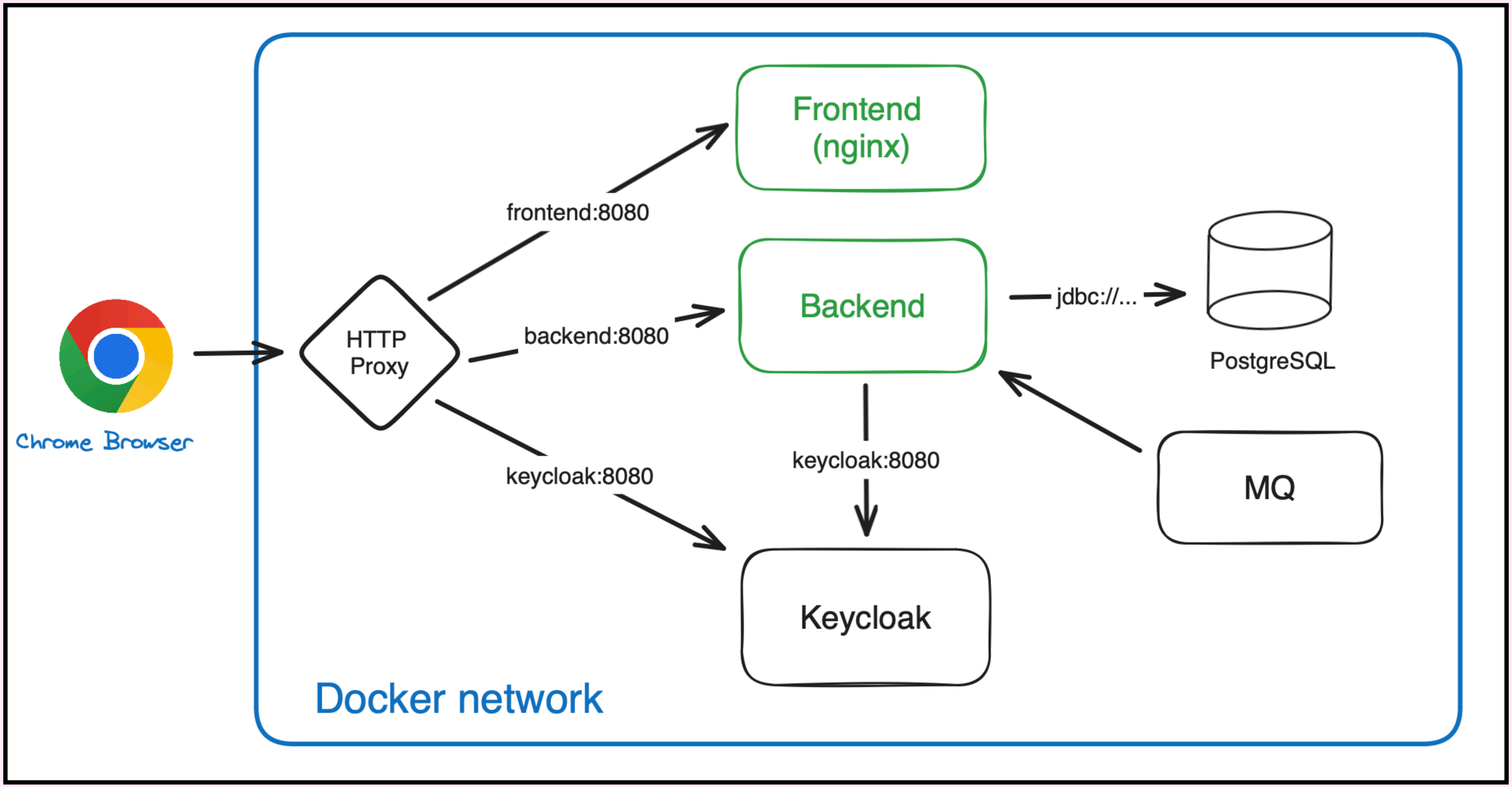
# BAM Client Systeemtest

- We testen exact de images die ook naar Productie gaan
  - Config o.b.v. Env Variables (12 factor app)
- Alleen de testcode en de browser draaien niet in Docker
- Alleen Docker en een JVM nodig, Playwright installeert de browsers
- Lokaal draai je precies dezelfde test als in de CI/CD-pipeline

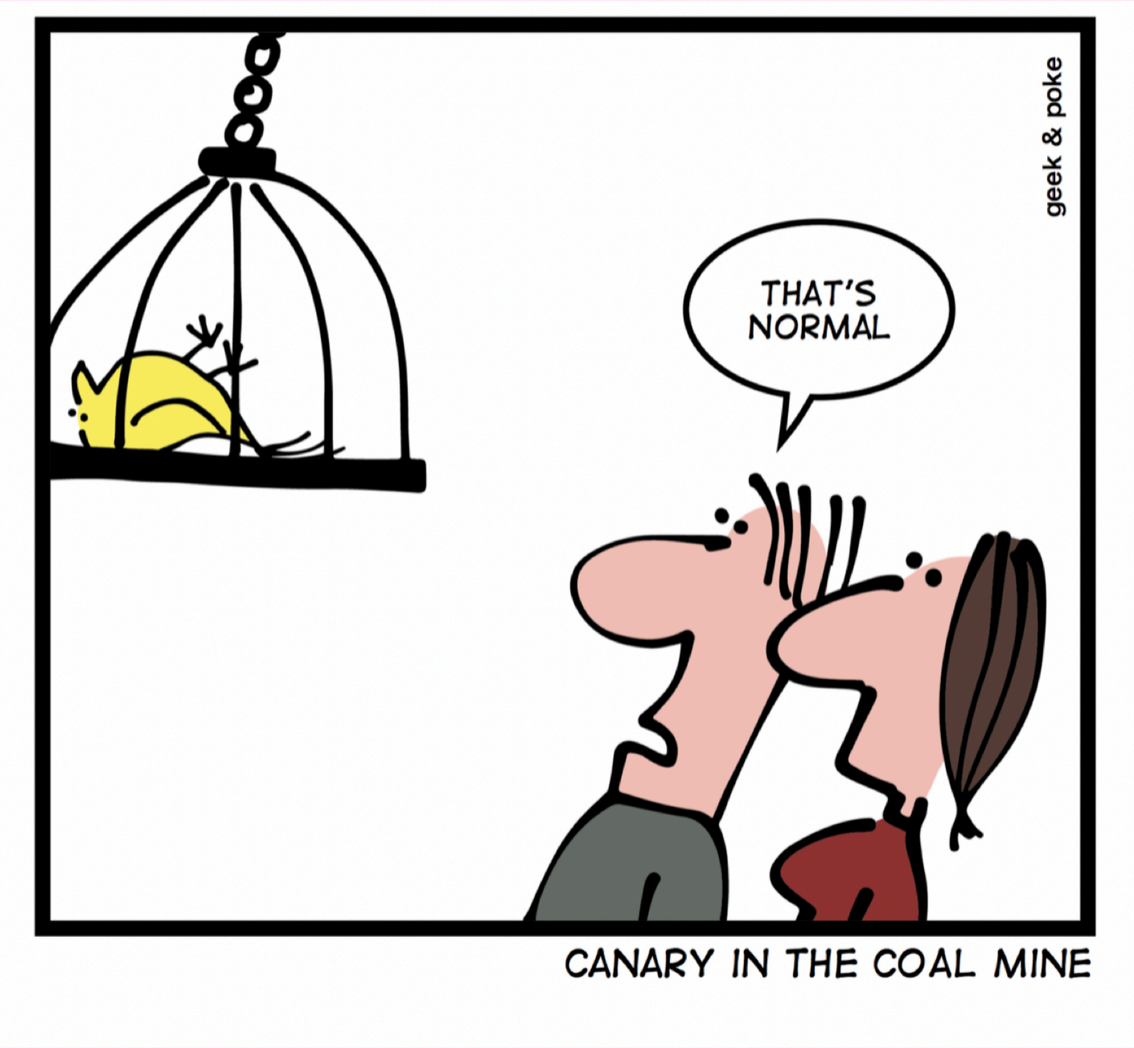
https://12factor.net/


Chrome Browser
http://localhost:45432/
http://localhost:46543/
Frontend (nginx)
Backend
jdbc://...
PostgreSQL
MQ
Docker network


Chrome Browser
http://localhost:55714/
http://localhost:55854/
http://localhost:54321
Frontend (nginx)
Backend
jdbc://...
PostgreSQL
Keycloak
MQ
Docker network


Frontend
(nginx)
frontend:8080
Backend
backend:8080
HTTP
Proxy
jdbc://...
PostgreSQL
Chrome Browser
keycloak:8080
keycloak:8080
MQ
Keycloak
Docker network

# Demo time



# Benefits

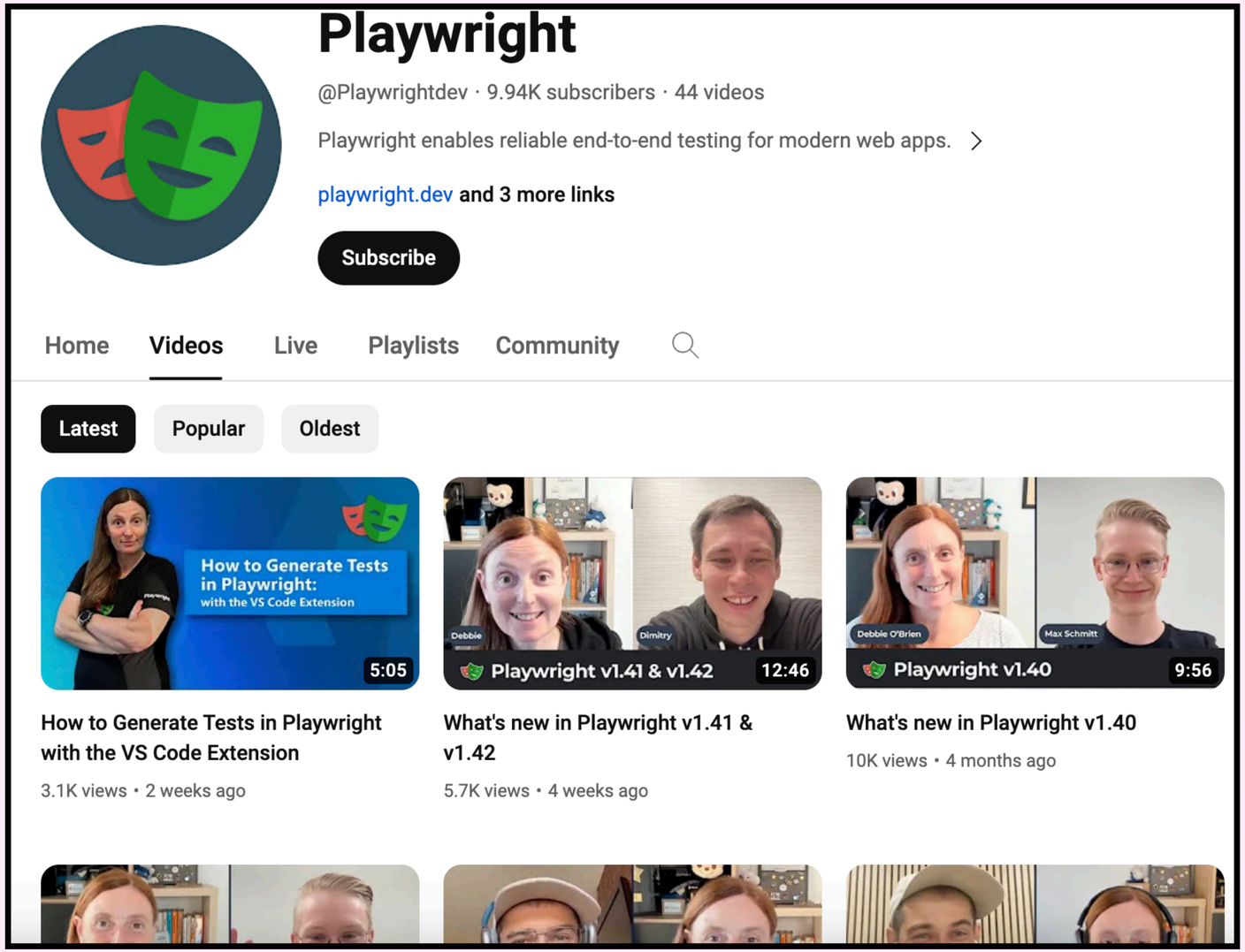
- Als team geeft deze test als extra vangnet veel vertrouwen
- Ook handig om lokaal de app te draaien tijdens development / debugging
- Testen van bijv. een Keycloak-upgrade kan in 5 minuten
- Genereren van documentatie die altijd up-to-date is met screenshots etc.

# Next steps

- Opleidingsdoeleinden
- Genereren van documentatie die altijd up-to-date is met screenshots etc.
- Duurtest toevoegen (bijv. ’s nachts uitvoeren)

# Tips & Tricks Playwright

- Demo’s op YouTube
- playwright.dev

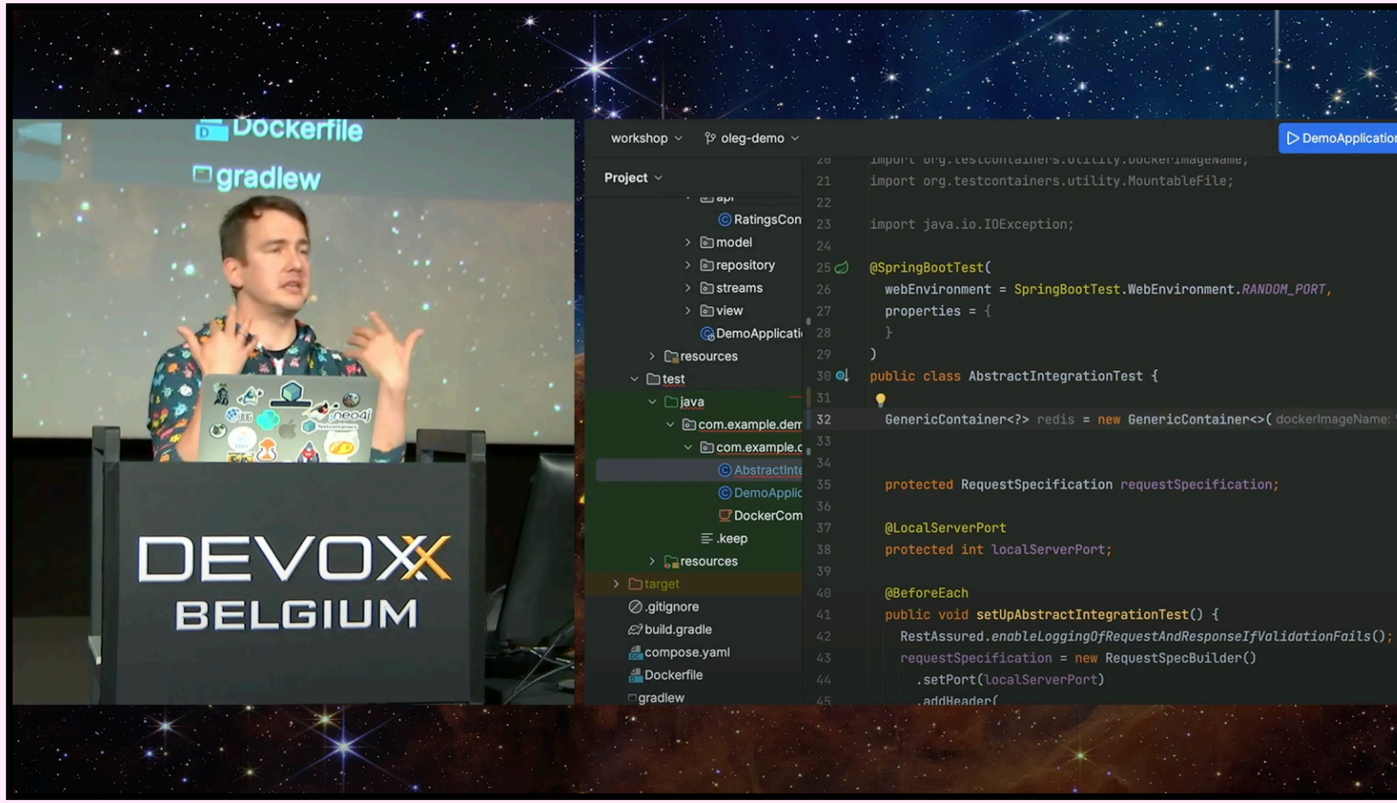


# Tips & Tricks Testcontainers

- Presentatie’s op Youtube (Devoxx)
- https://github.com/testcontainers
- Localstack voor AWS services

LocalStack

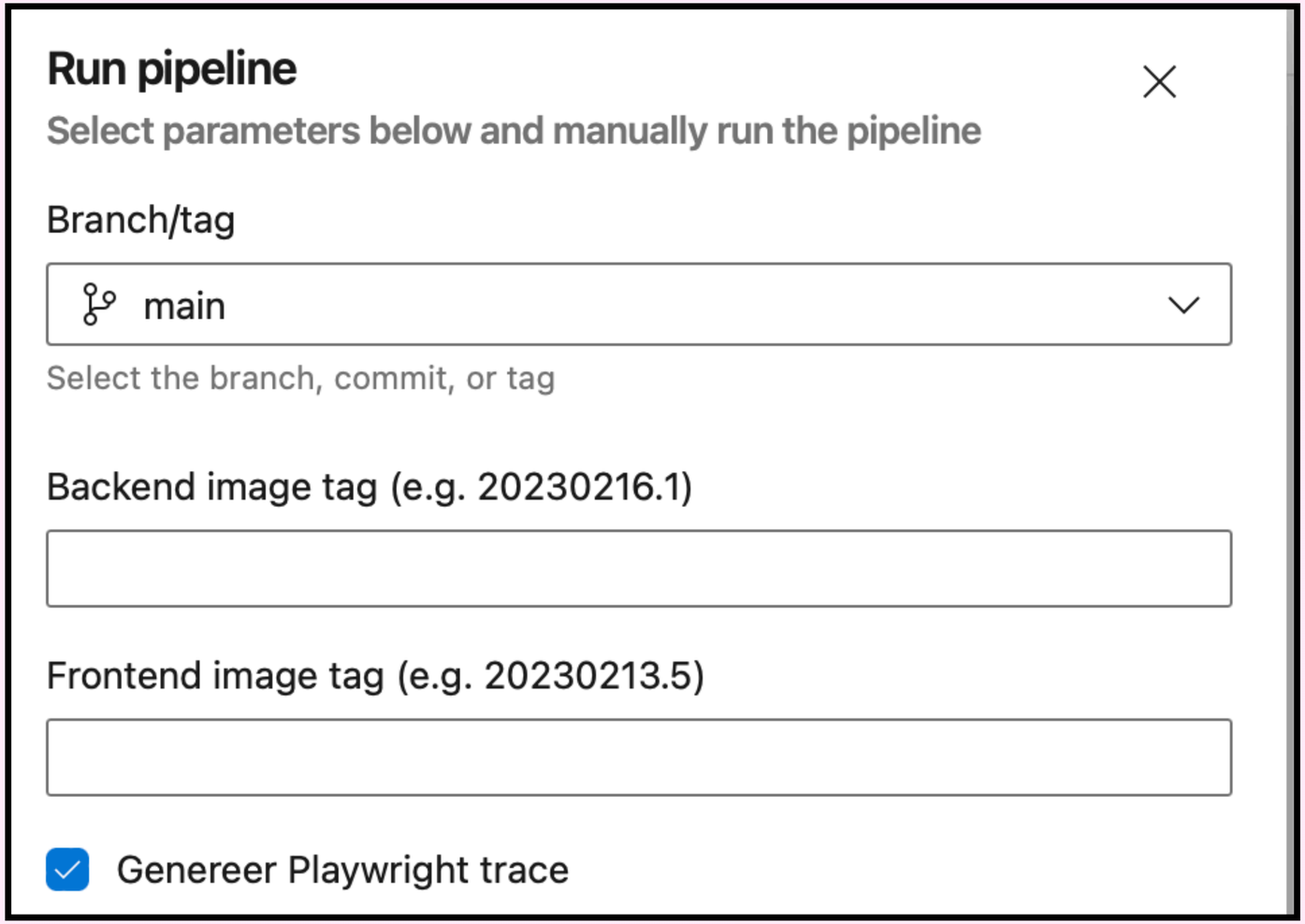
Run Locally, Deploy Globally




https://www.youtube.com/watch?v=0kXEwo0XFaY

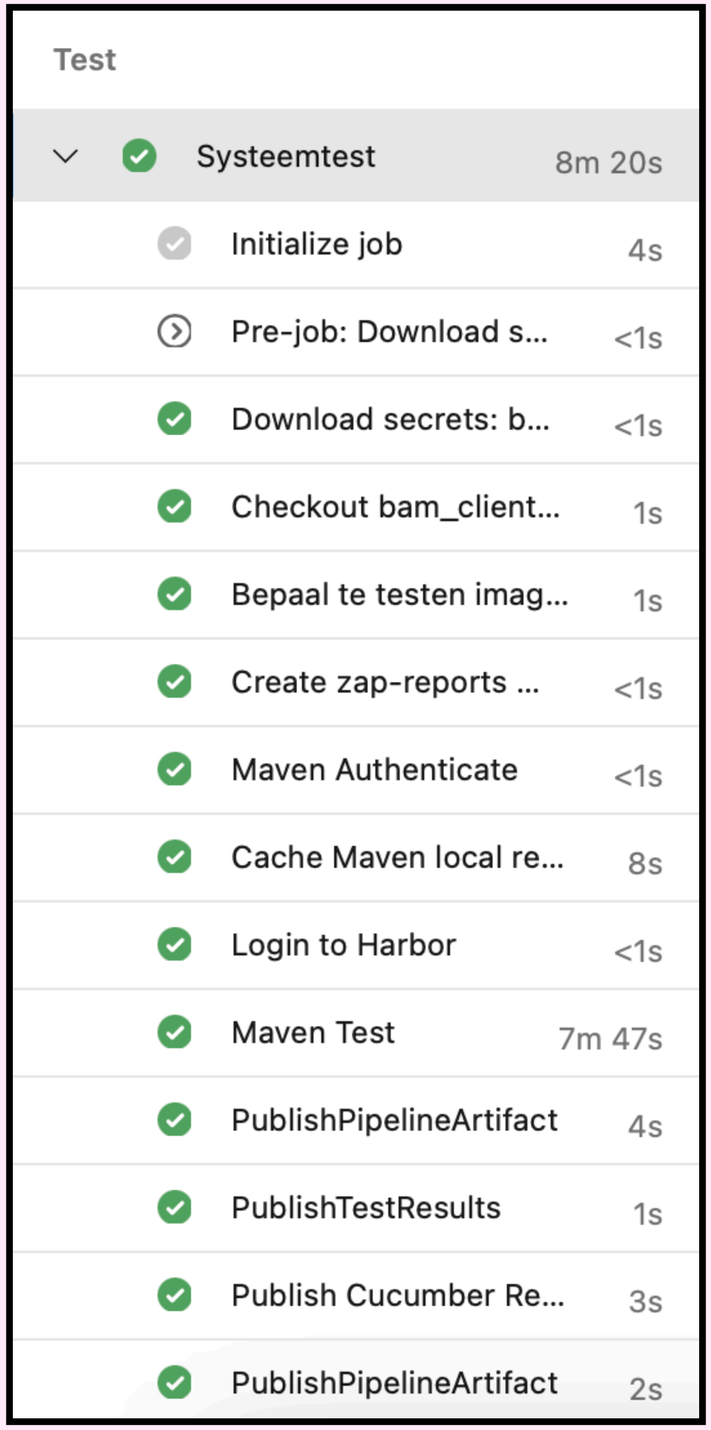
# Azure DevOps

## Pipelines



# Azure DevOps

## Pipelines



# Vragen?


POST OFFICE
POST OFFICE
Bureau de Change

# Introducing SafeTest: A Novel Approach to Front End Testing

Netflix Technology Blog · Follow
Published in Netflix TechBlog · 7 min read · Feb 13, 2024



by Moshe Kolodny

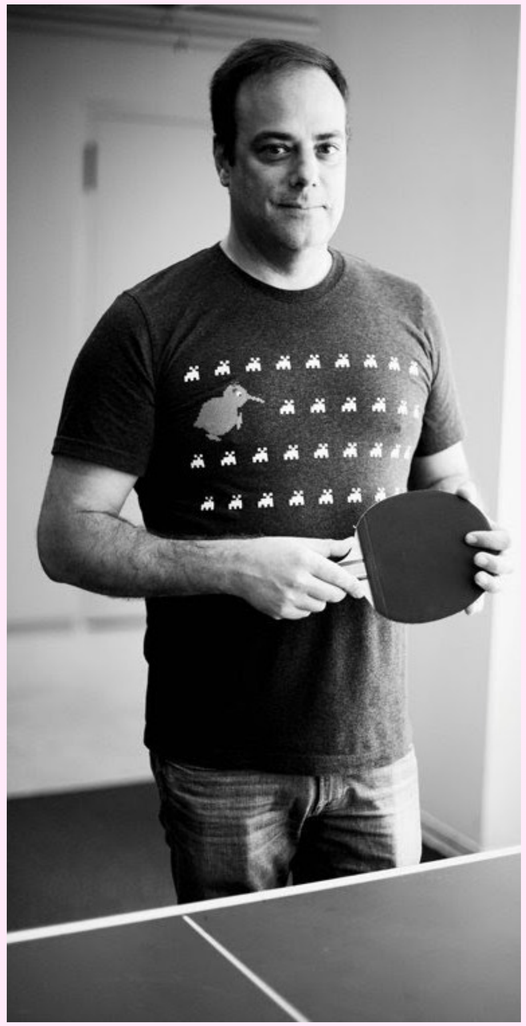
In this post, we're excited to introduce SafeTest, a revolutionary library that offers a fresh perspective on End-To-End (E2E) tests for web-based User Interface (UI) applications.

## The Challenges of Traditional UI Testing


TEST
SLOPES


1965
RALLYE MONTE-CARLO
52
AJB 44B

# Controlling Your Environment Makes You Happy



APRIL 10, 2000 *by* JOEL SPOLSKY

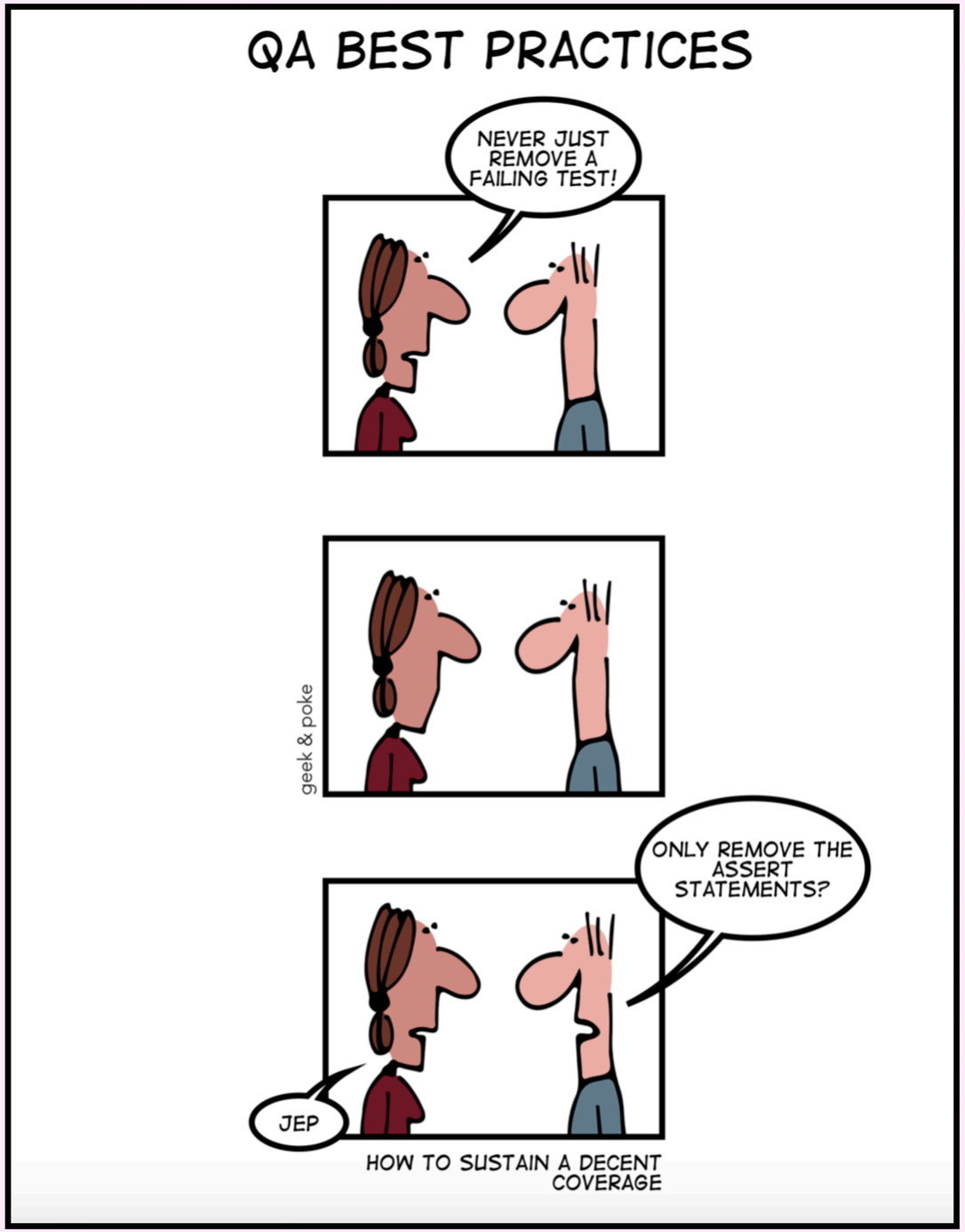
## Controlling Your Environment Makes You Happy

SOFTWARE DESIGNER, UIBOOK

Most of the hard core C++ programmers I know *hate* user interface programming. This surprises me, because I find UI programming to be quintessentially easy, straightforward, and fun.

It's *easy* because you usually don't need algorithms more sophisticated than how to center one rectangle in another. It's *straightforward*

# Coverage 🚫


QA BEST PRACTICES
NEVER JUST REMOVE A FAILING TEST!
geek & poke
ONLY REMOVE THE ASSERT STATEMENTS?
JEP
HOW TO SUSTAIN A DECENT COVERAGE