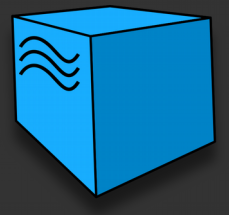
Efficient Selenium Infrastructure with Selenoid

by Ivan Krutov and Roman Orlov



Ivan Krutov

Java & Golang developer

Selenoid core maintainer

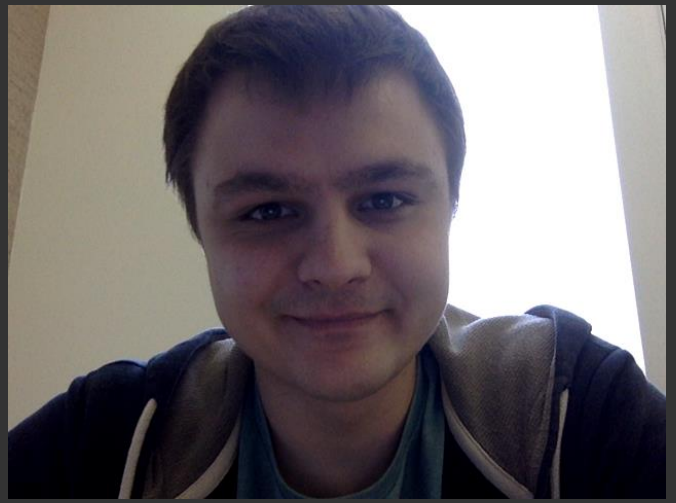
Devops

Roman Orlov

Python and Golang developer

Devops and SRE

Big Selenium cluster



Selenium clusters

Typical Selenium Grid	My cluster
50 browsers	5000+ browsers
10000 sessions / day	2.5M sessions / day
1 data center	5 data centers
5 rps	4000 rps
500 Kbit/s	> 2 Gbit/s
Sometimes down	24/7/365

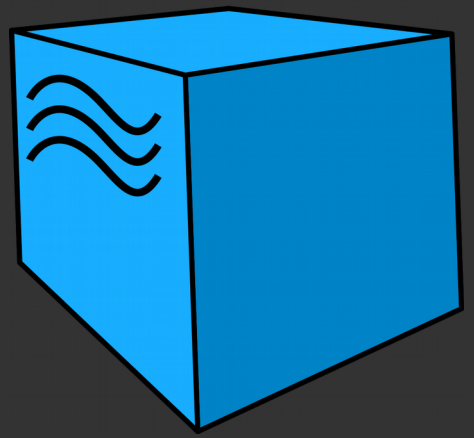
Available browsers








Selenium Tools



Selenoid

Moon

Browsers

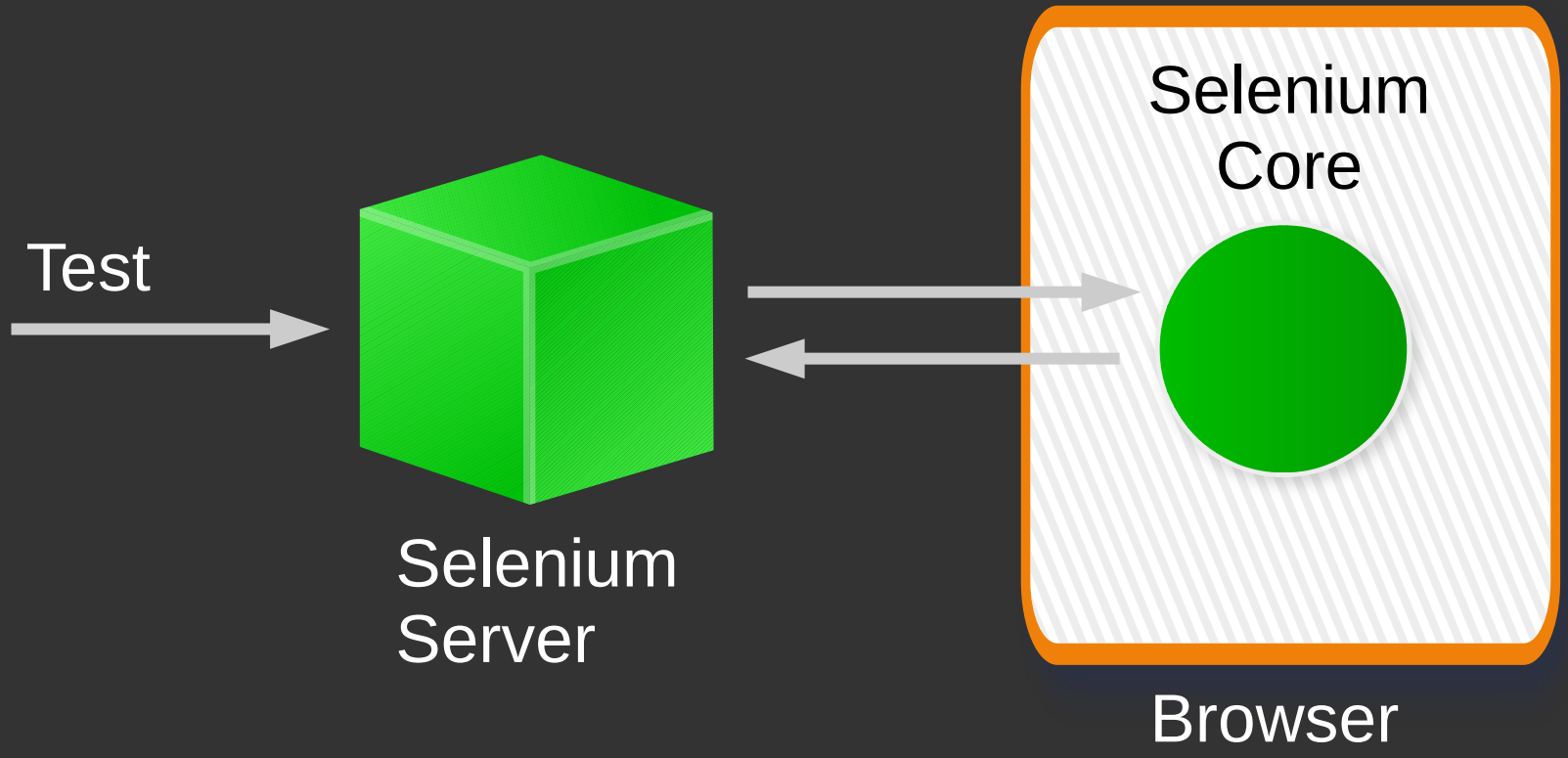
Agenda

Local browser tests development

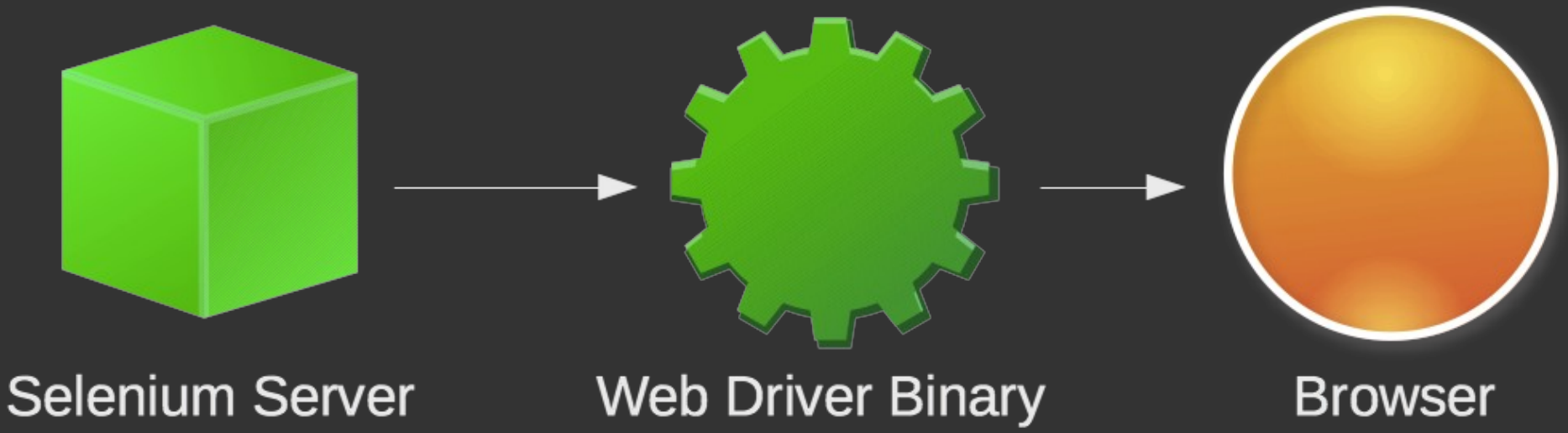
Creating Selenium cluster

Advanced Selenoid features for big clusters

Selenium RC architecture



Selenium WebDriver architecture



Installing Selenium Webdriver

Manually install Java

Manually download Selenium JAR

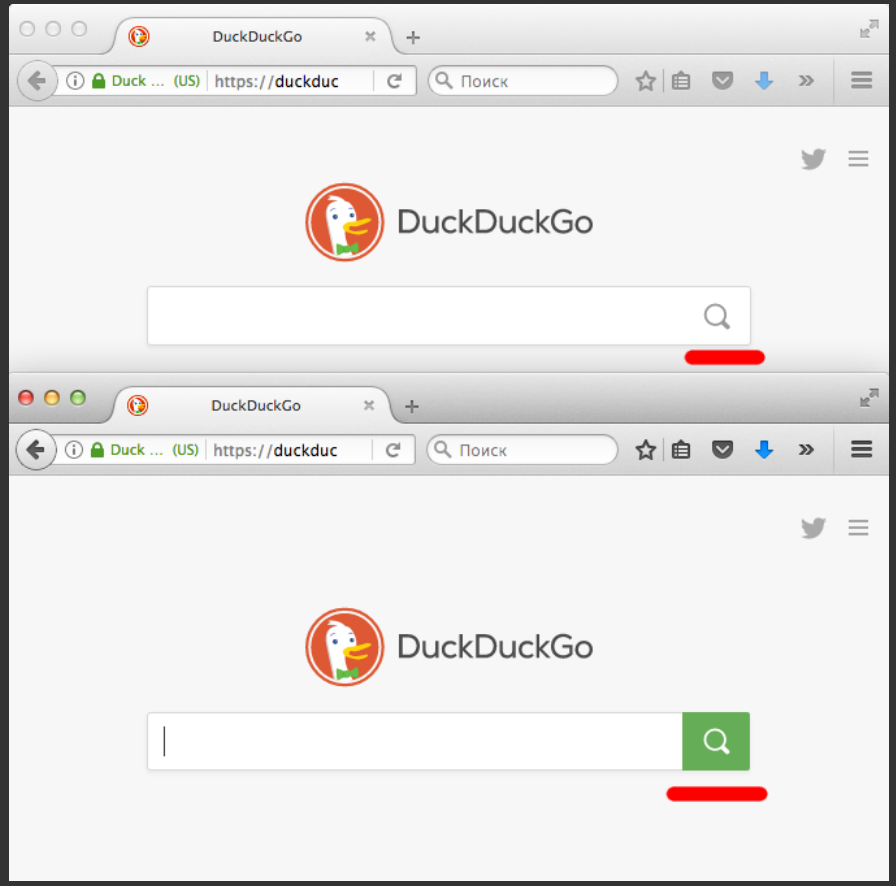
Manually download web-driver binaries

Manually unpack web-drivers from archives

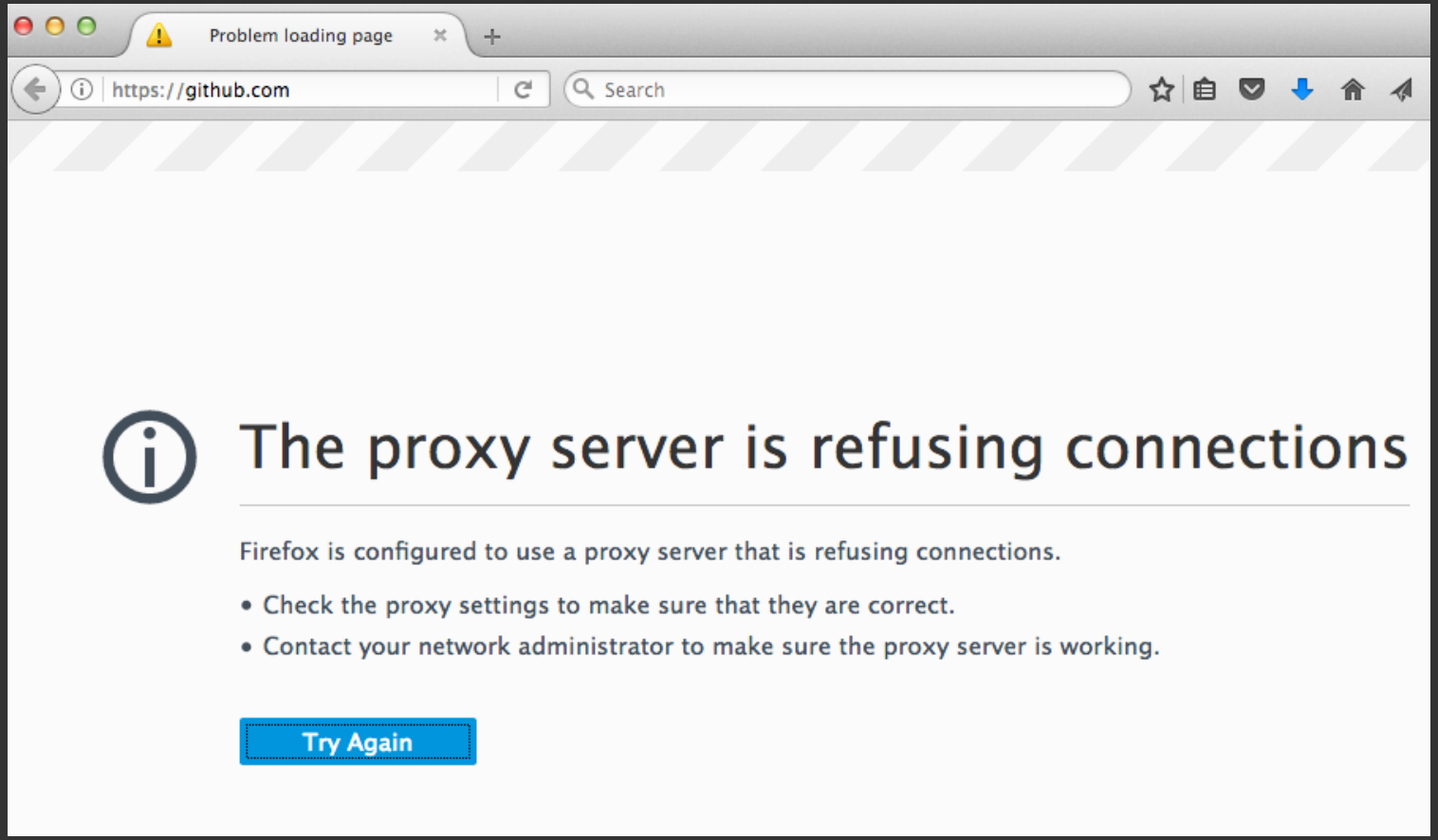
Manually install required browsers

Manually type complicated start command

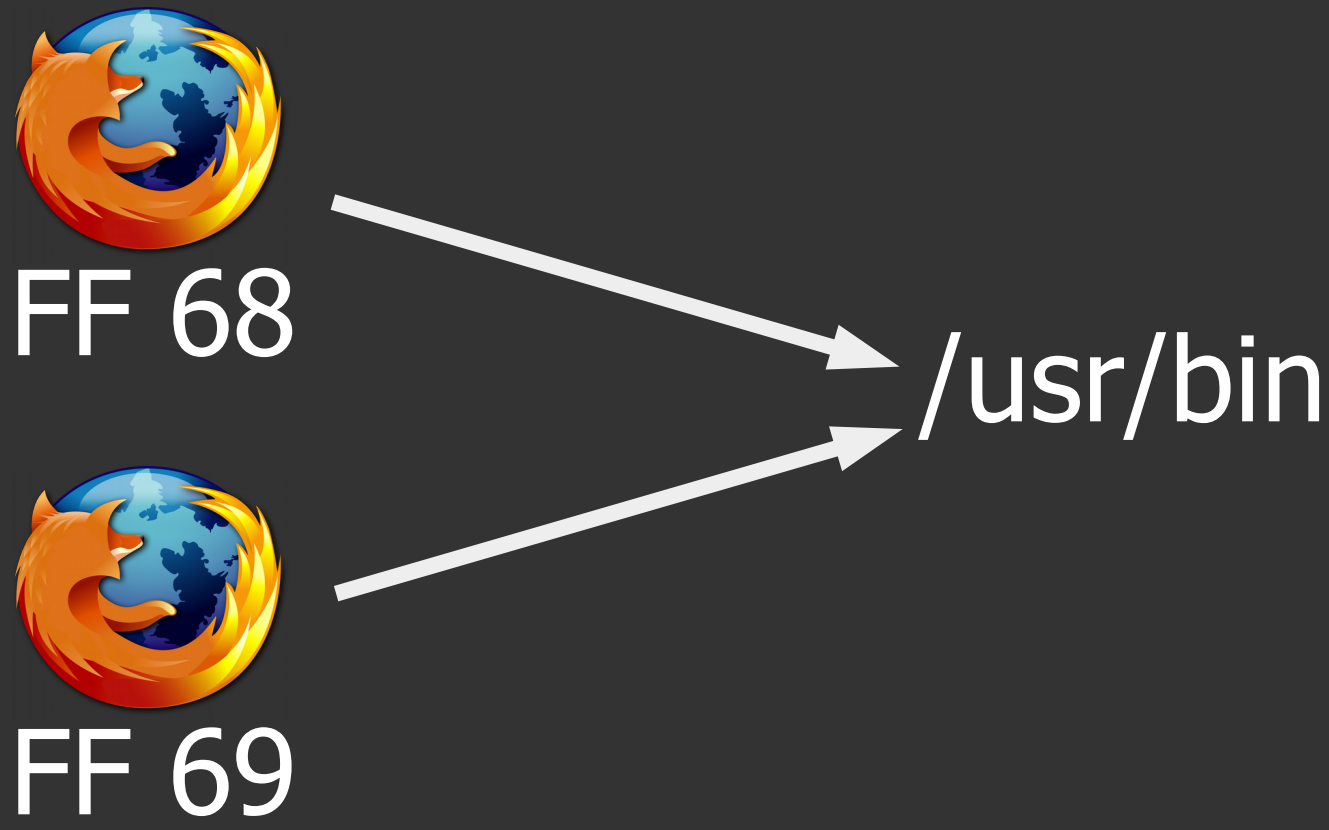
Selenium window focus problem



Selenium browser settings problem

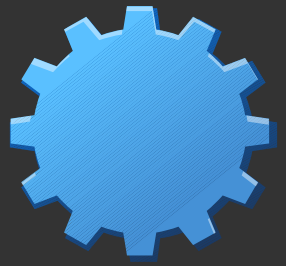


Installing two Firefox versions



Selenium compatibility issues

75



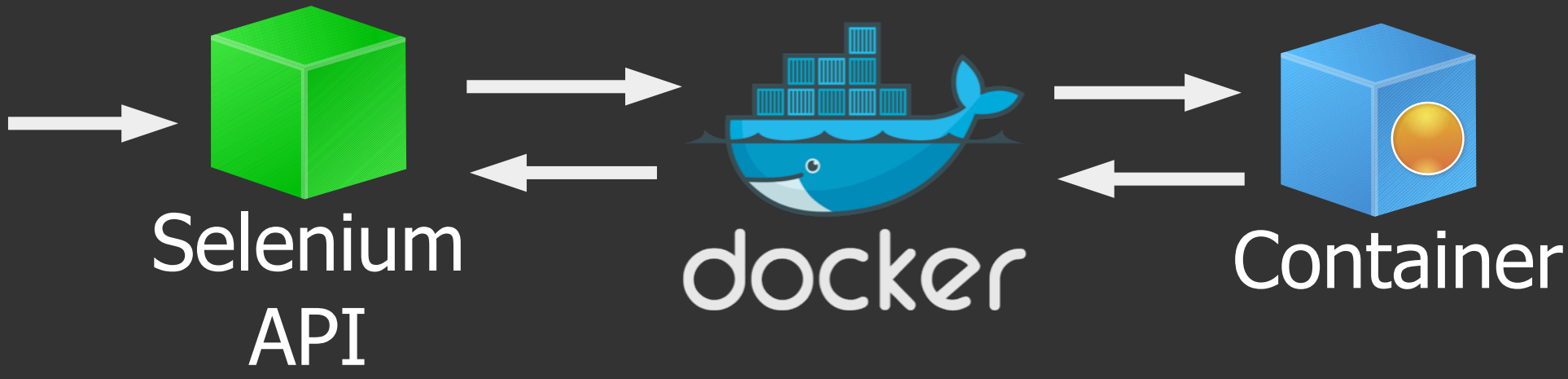
Chromedriver ???

Bad Foundation

Linux covers 80% of browsers

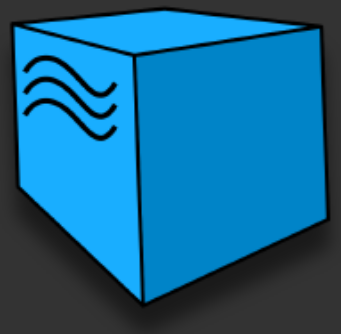
Let’s use containers where possible!

How it should work

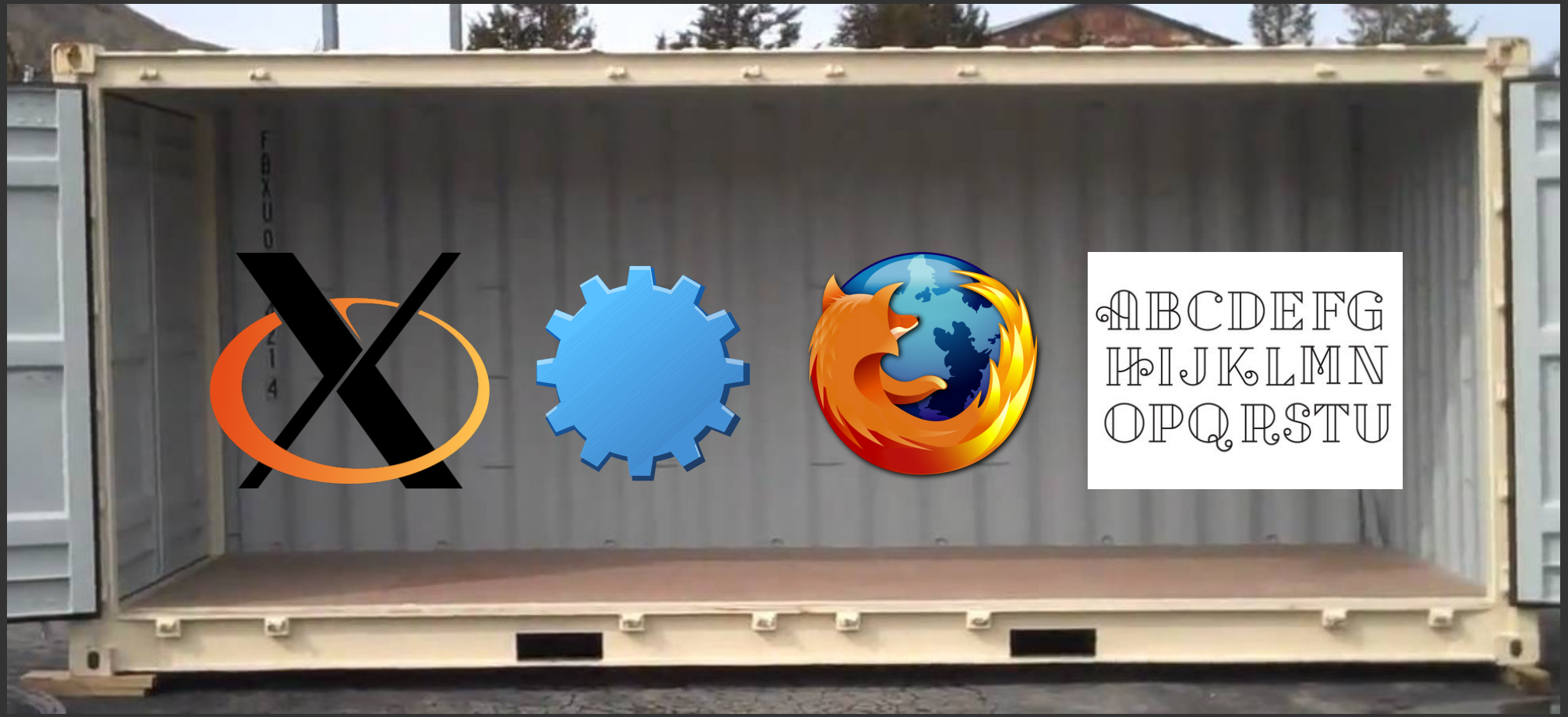


Selenium

Selenoid



Selenoid runs isolated browsers



Images for all recent browsers

3.6+

48.0+

33.0+
12.16

https://bit.ly/35sK7Jv

https://bit.ly/testconworkshop

Chrome Devtools Protocol

JSON over WS

Chrome Devtools Protocol

DOM tree manipulation

CSS style manipulation

Tracking network activities

Inspecting accessibility tree

Page tracing and profiling

Subscribe to page events

Scaling Selenium for small team

Workstation

Server or VM

Fault Tolerance

DC1

DC2

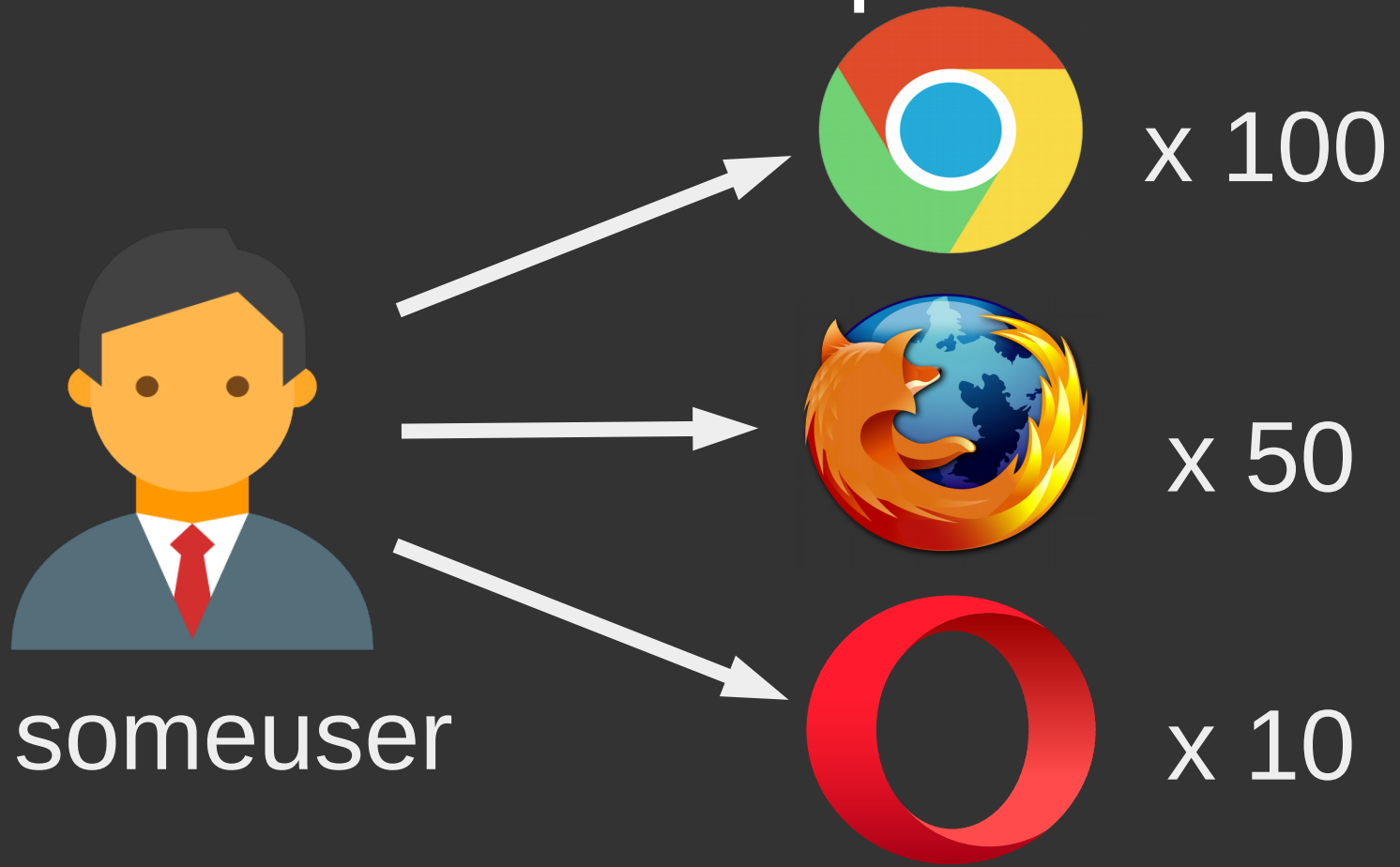
Scalability

DC1 DC2 DC3 DC4

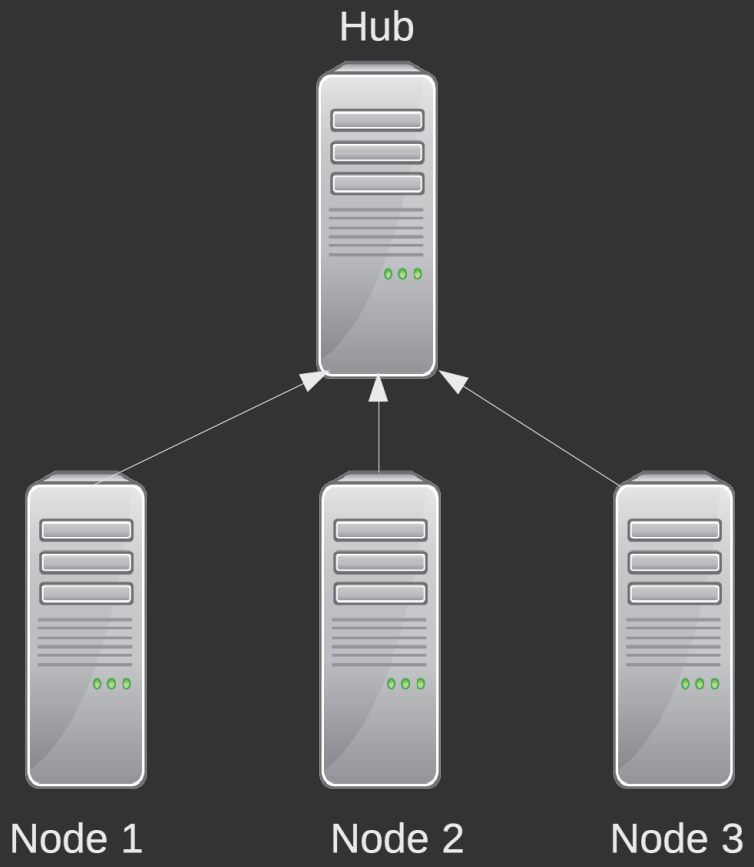
Standard API

Access control

Browser consumption control



Selenium Grid



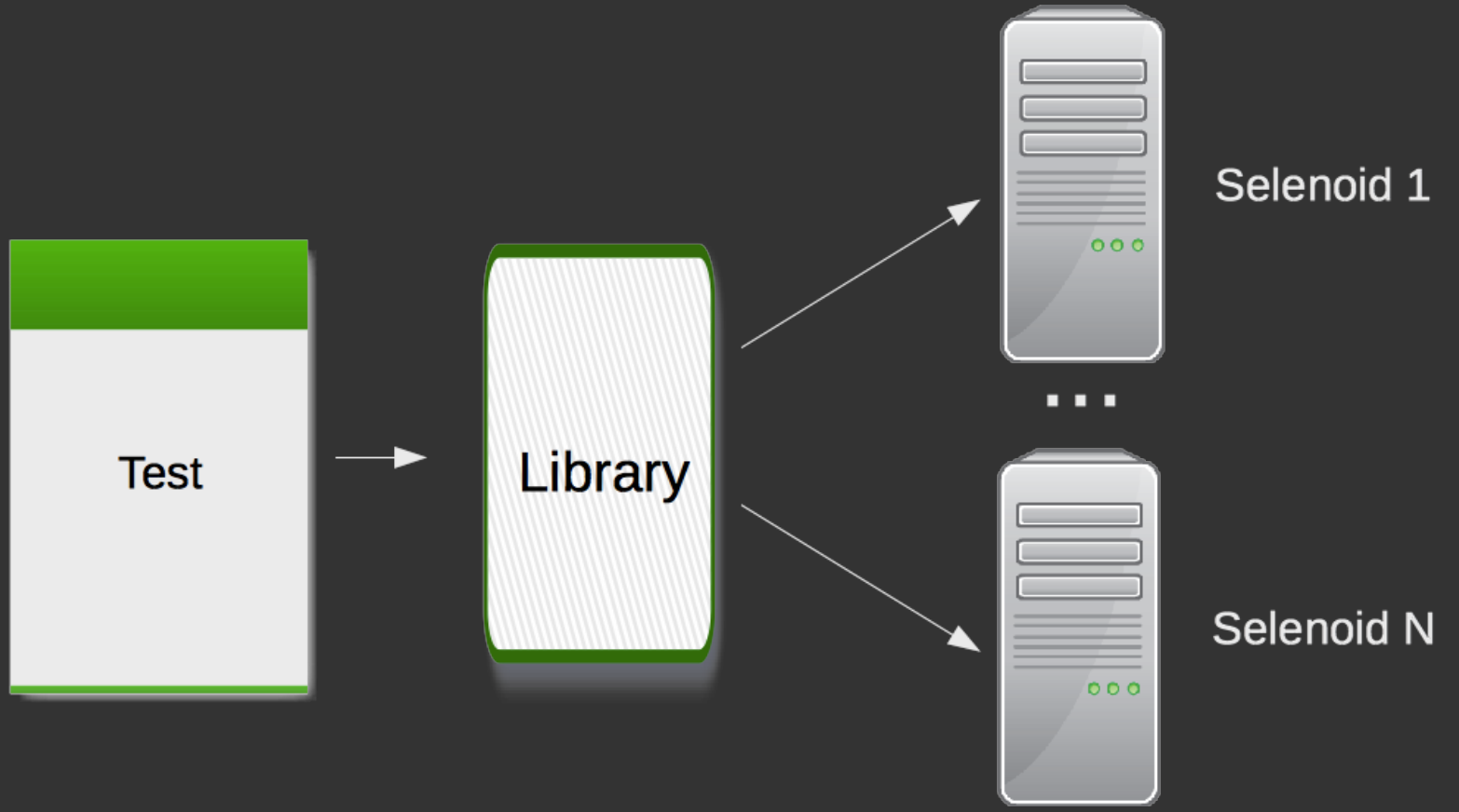
Selenium Grid

Not fault-tolerant

Not really scalable

No access control

Client-side Load Balancing



Hosts Information File

```
<qa:browsers xmlns:qa="urn:config.gridrouter.qatools.ru">
    <browser name="firefox" defaultVersion="45.0">
        <version number="45.0">
            <region name="us-east">
                <host name="host1.example.com" port="4444" count="5"/>
                <host name="host2.example.com" port="4444" count="5"/>
            </region>
            <region name="us-west">
                <host name="host3.example.com" port="4444" count="3"/>
            </region>
        </version>
        <version number="46.0">
            ...
        </version>
    </browser>
    <browser name="chrome" defaultVersion="59.0">
        ...
    </browser>
</qa:browsers>
```

Client-side load balancing

Fault-tolerant

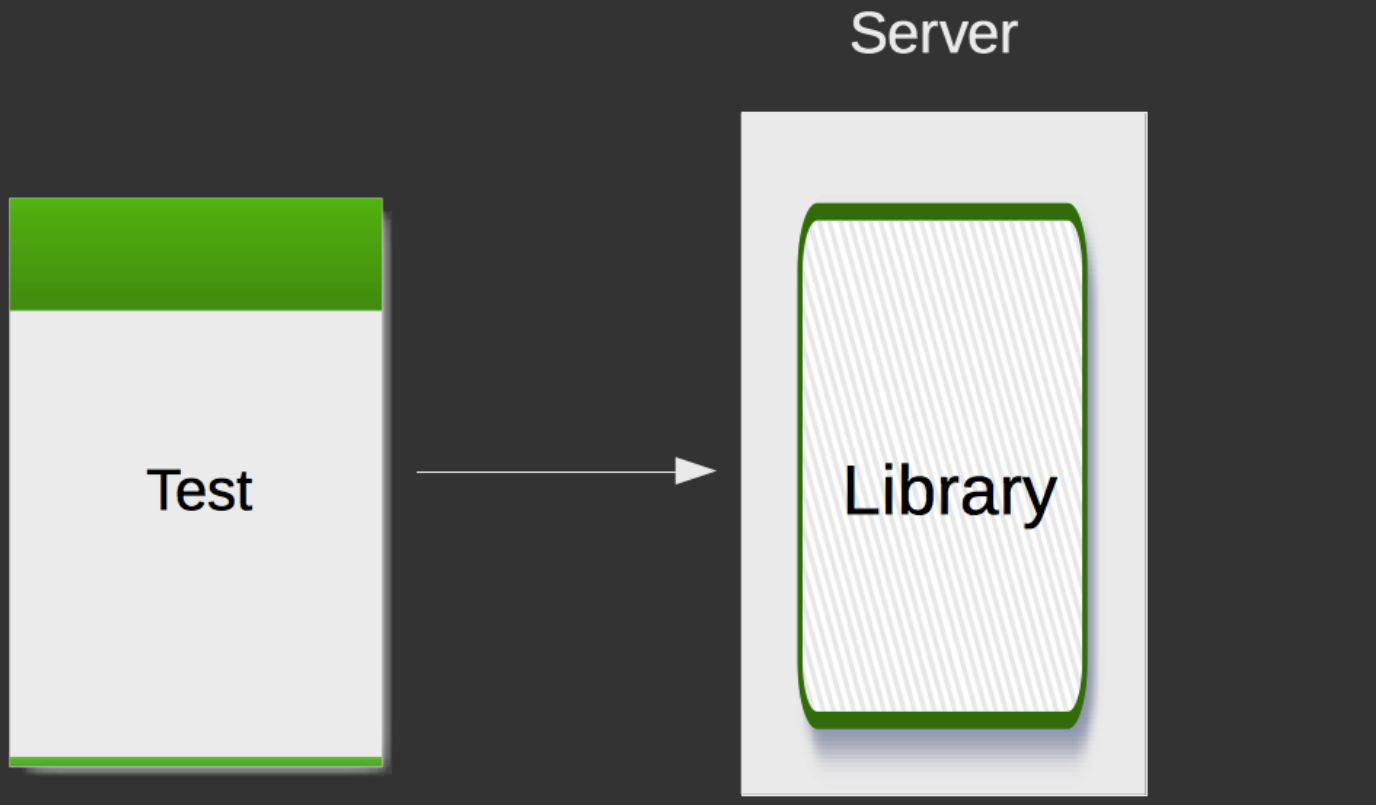
Scalable

No access control

Not standard Selenium

A lot of teams

Server-side Load Balancing



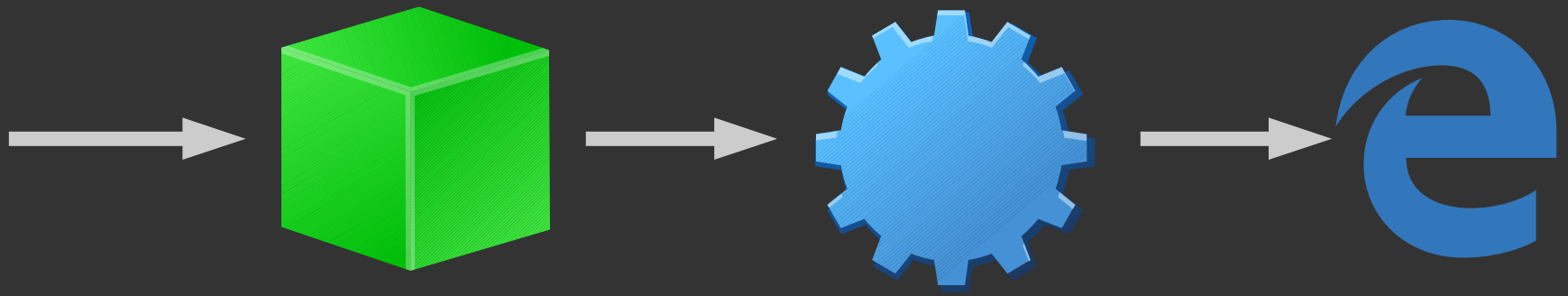
GoGridRouter (aka Ggr)

100% Golang

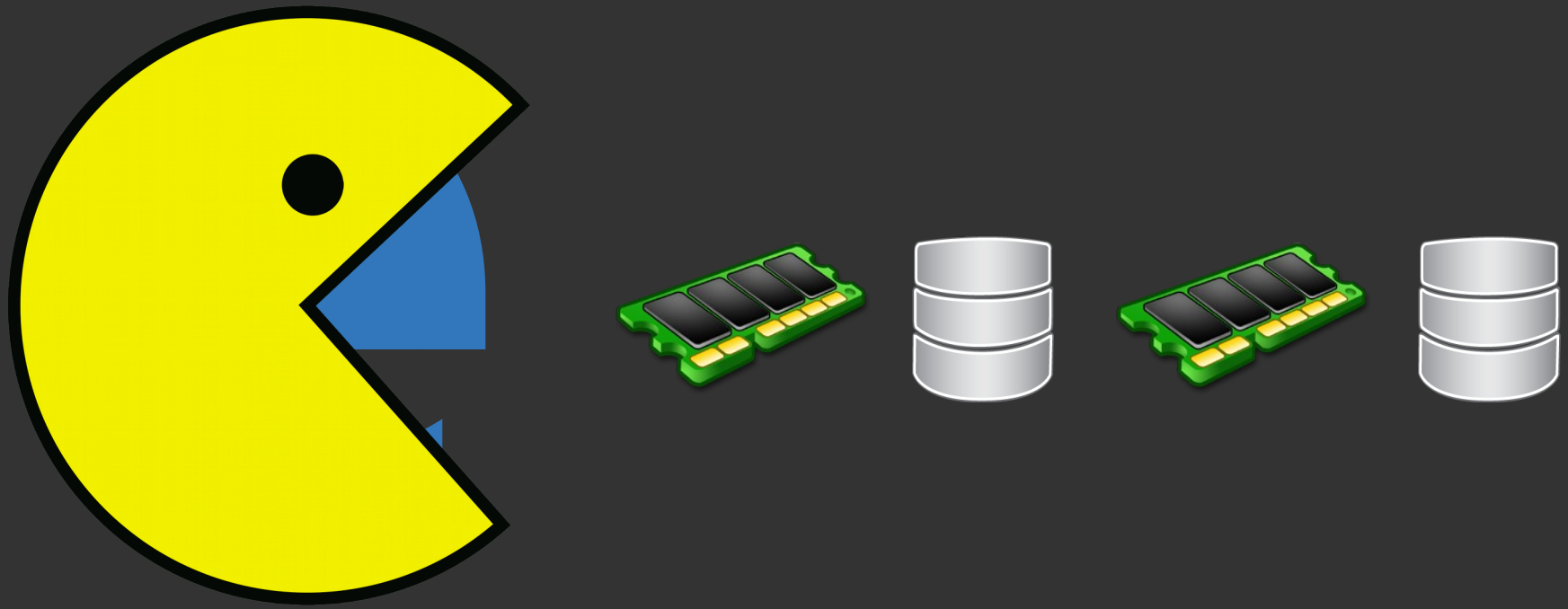
6 Mb binary

400 rps → 200 Mb RAM

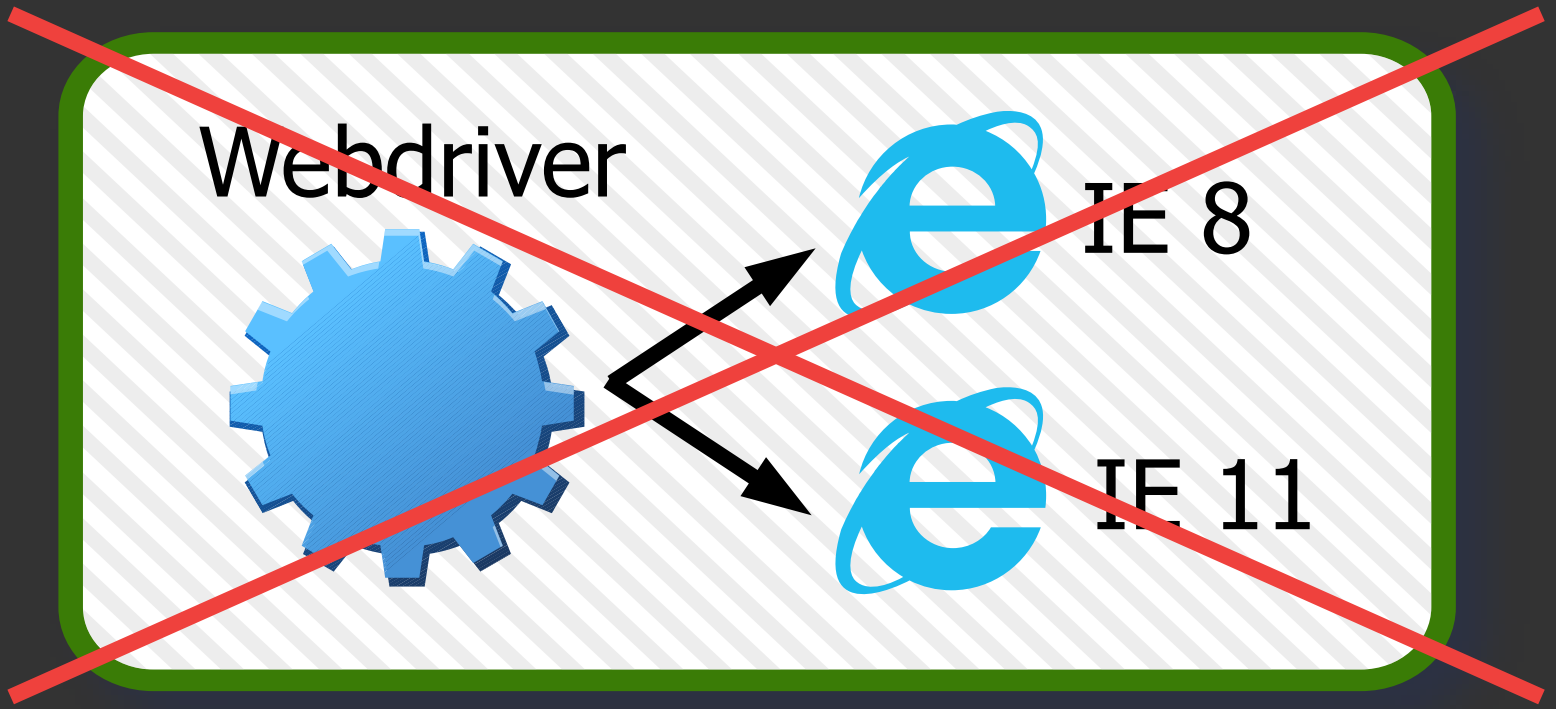
Traditional Selenium testing



Traditional Selenium testing



Traditional Selenium testing



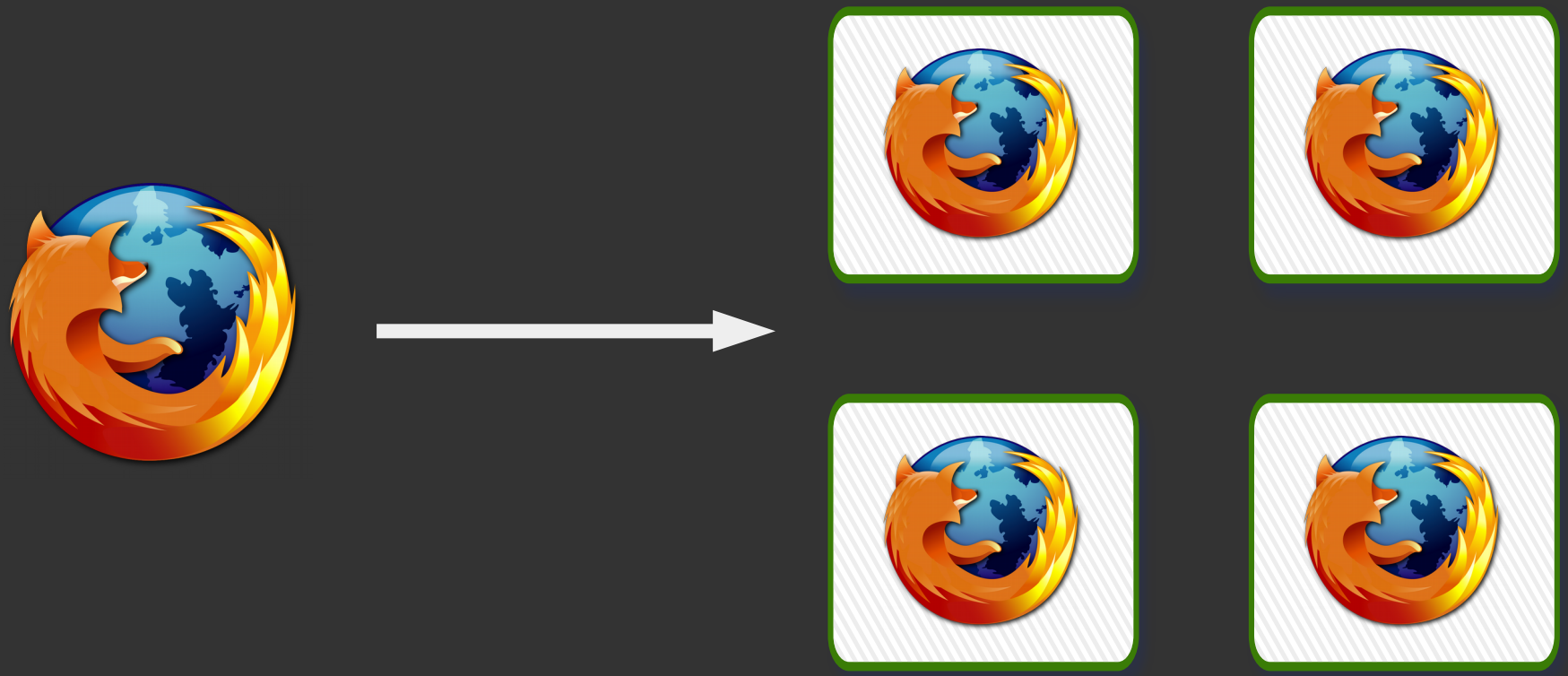
Traditional Selenium testing

Jeremy C. Jan 24, 2019

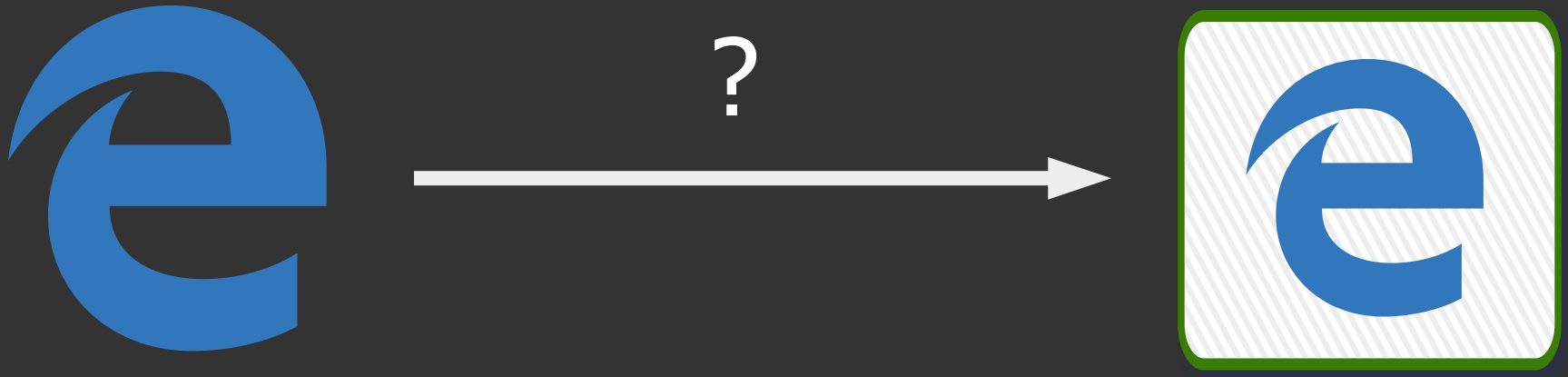
How is it possible that this is still an issue when it was reported over 3 years ago? It took a year and a half for MS to acknowledge it was an issue, and now it's the end of January 2019 and there is 0 update or communication.

Traditional Selenium testing

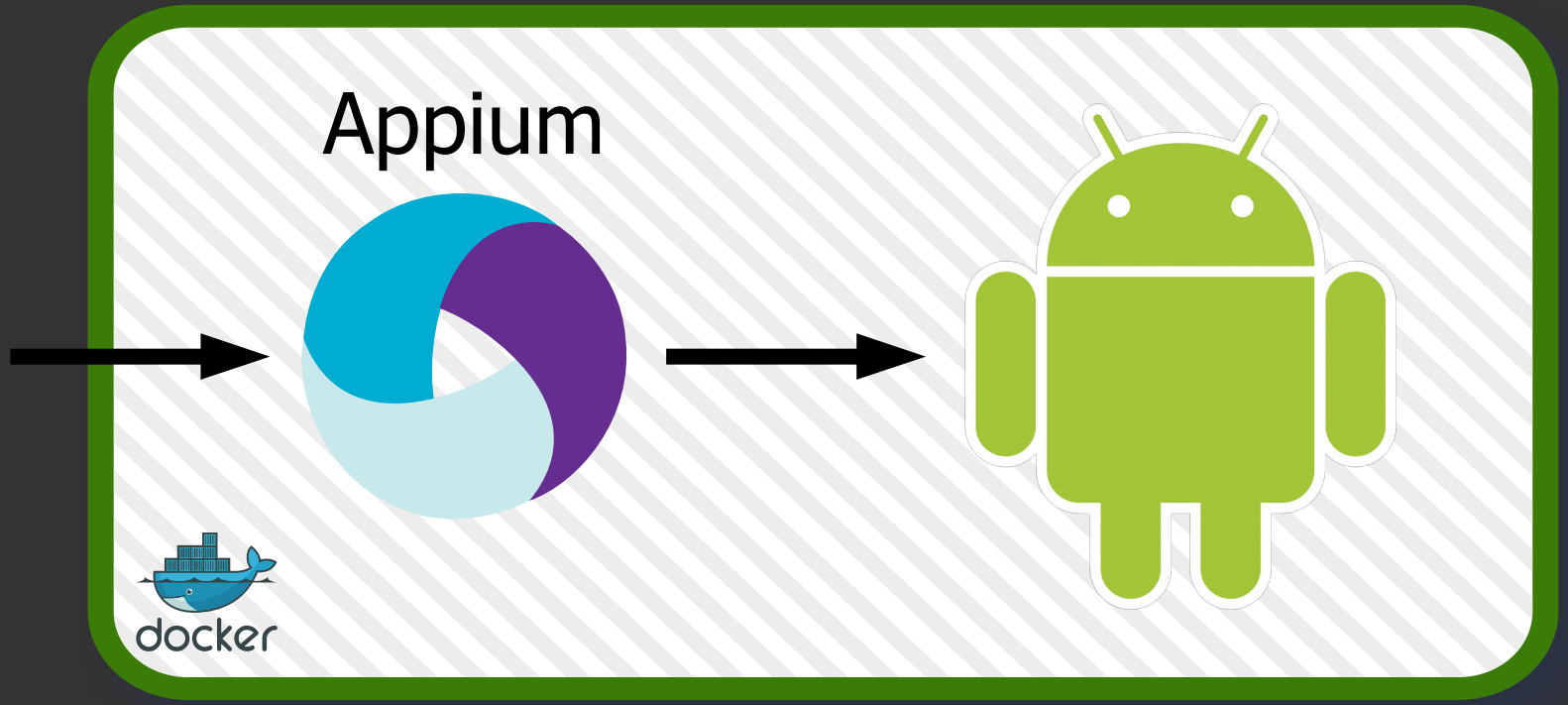
Browsers in containers



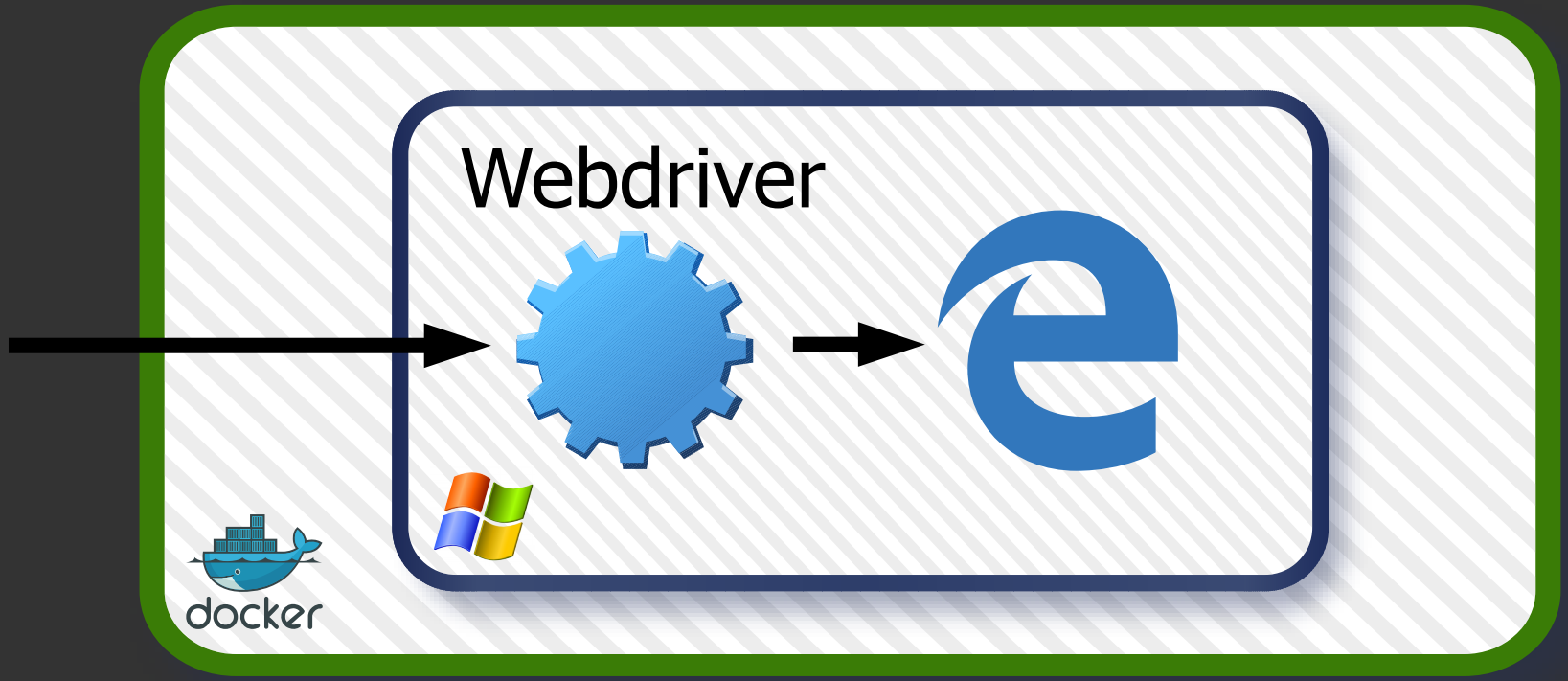
Windows in Docker container



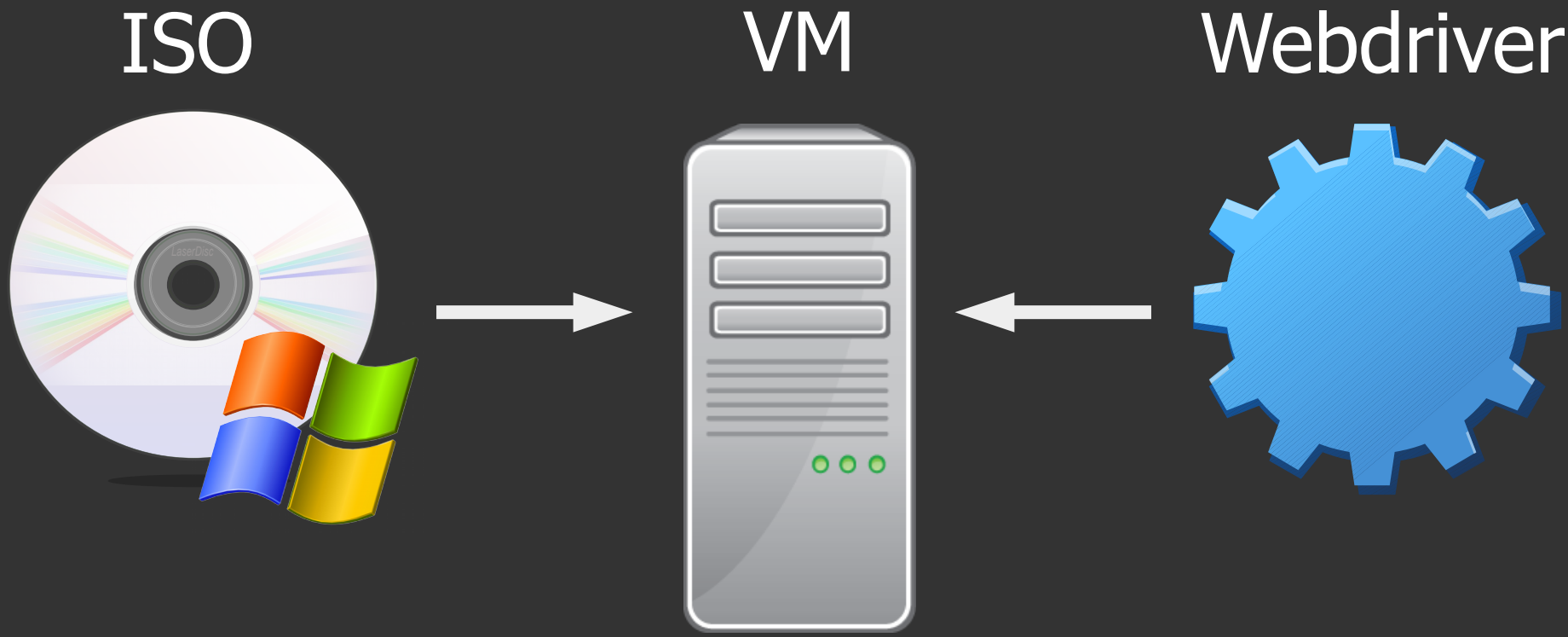
Android in Docker container



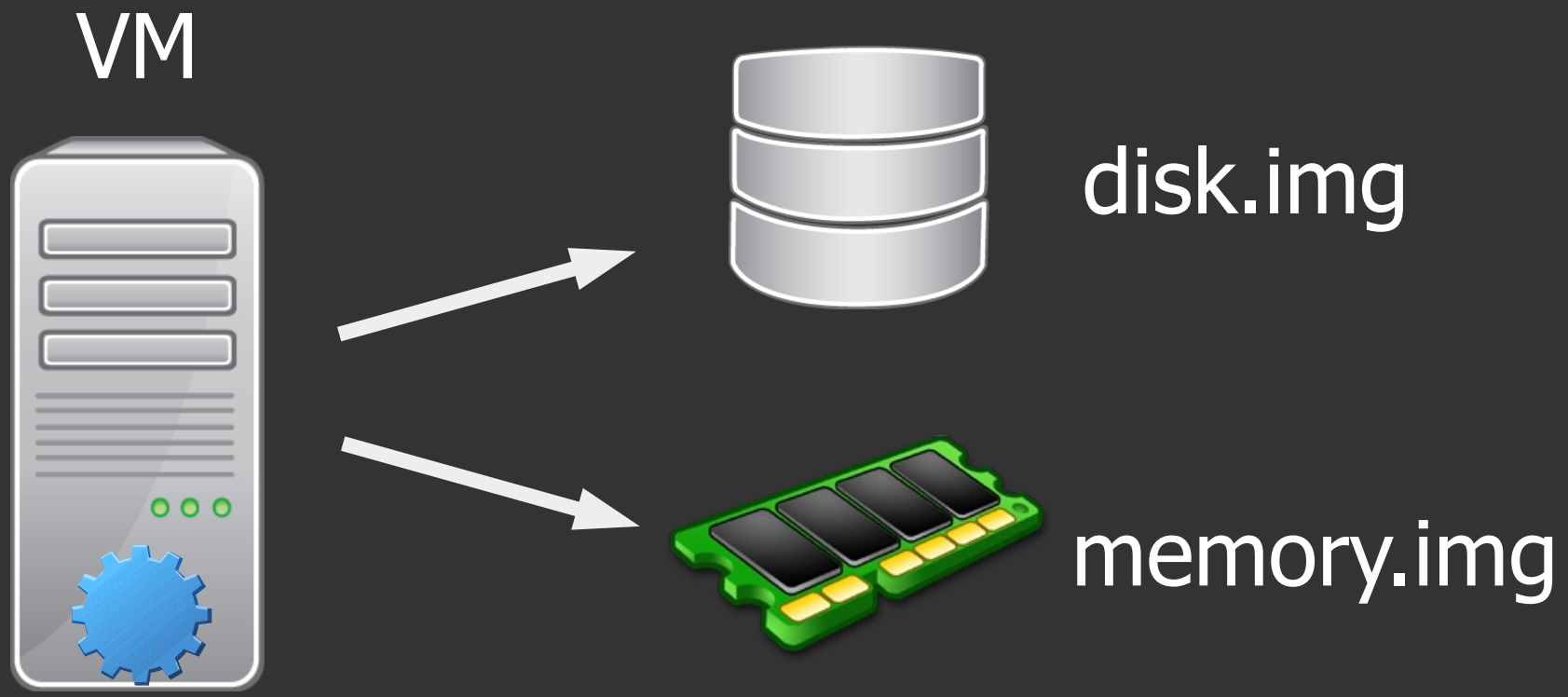
Windows in Docker container



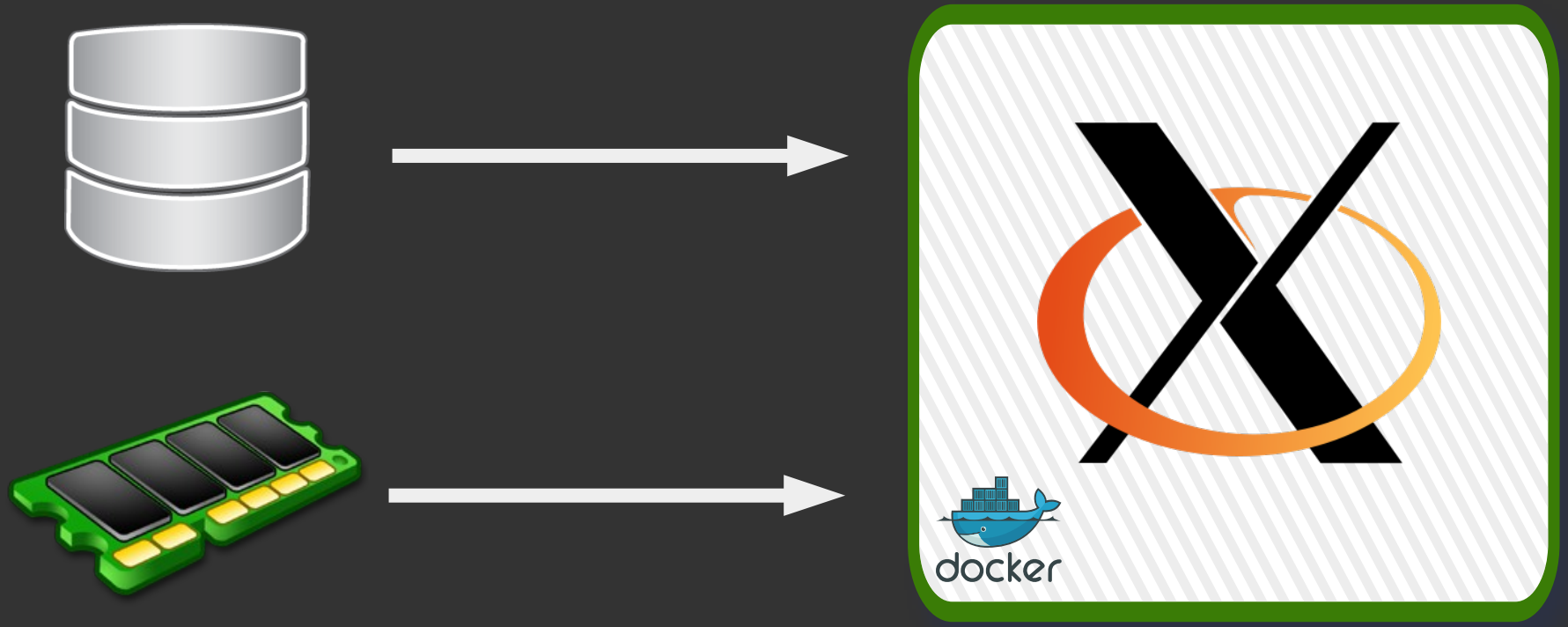
Building Windows image (step 1/3)



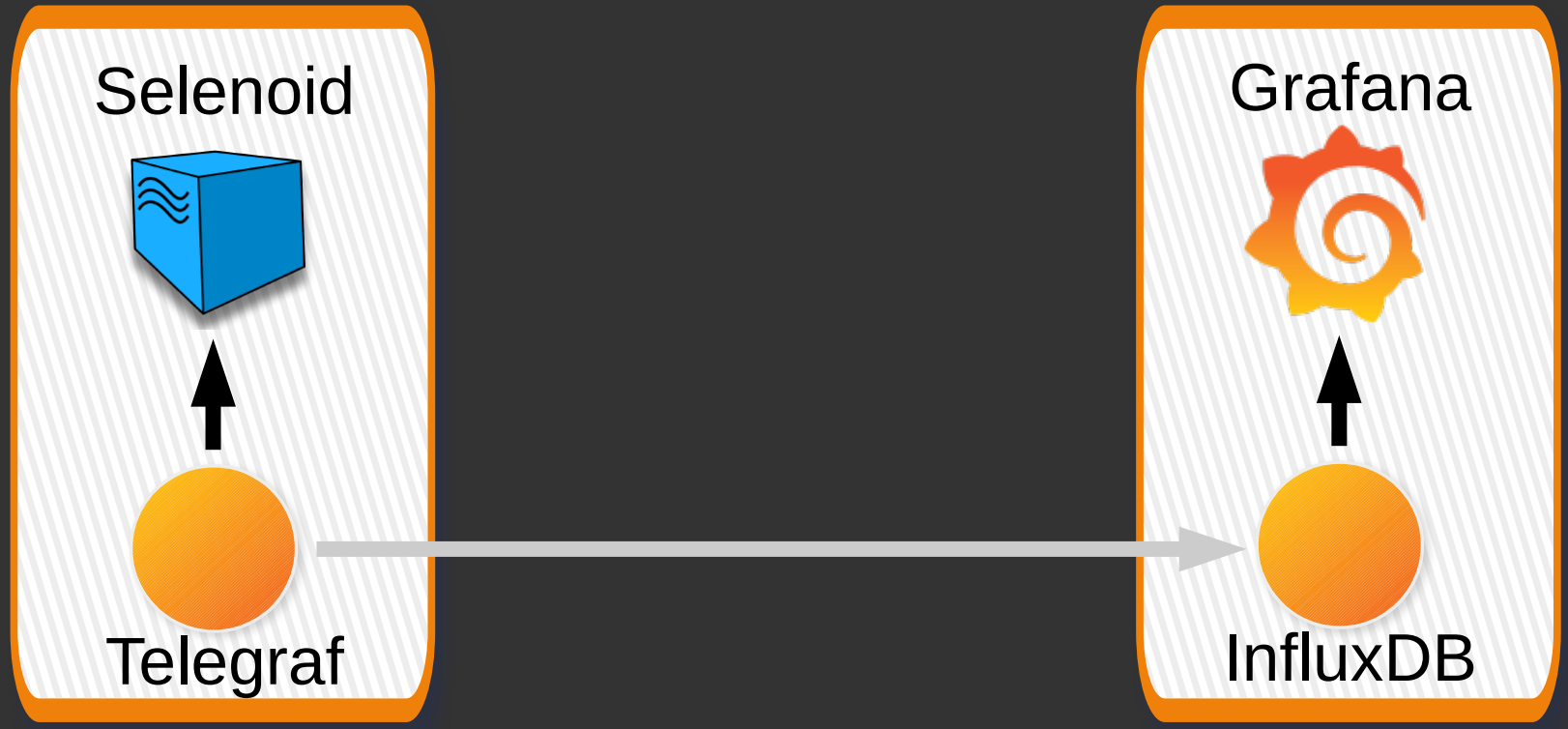
Building Windows image (step 2/3)

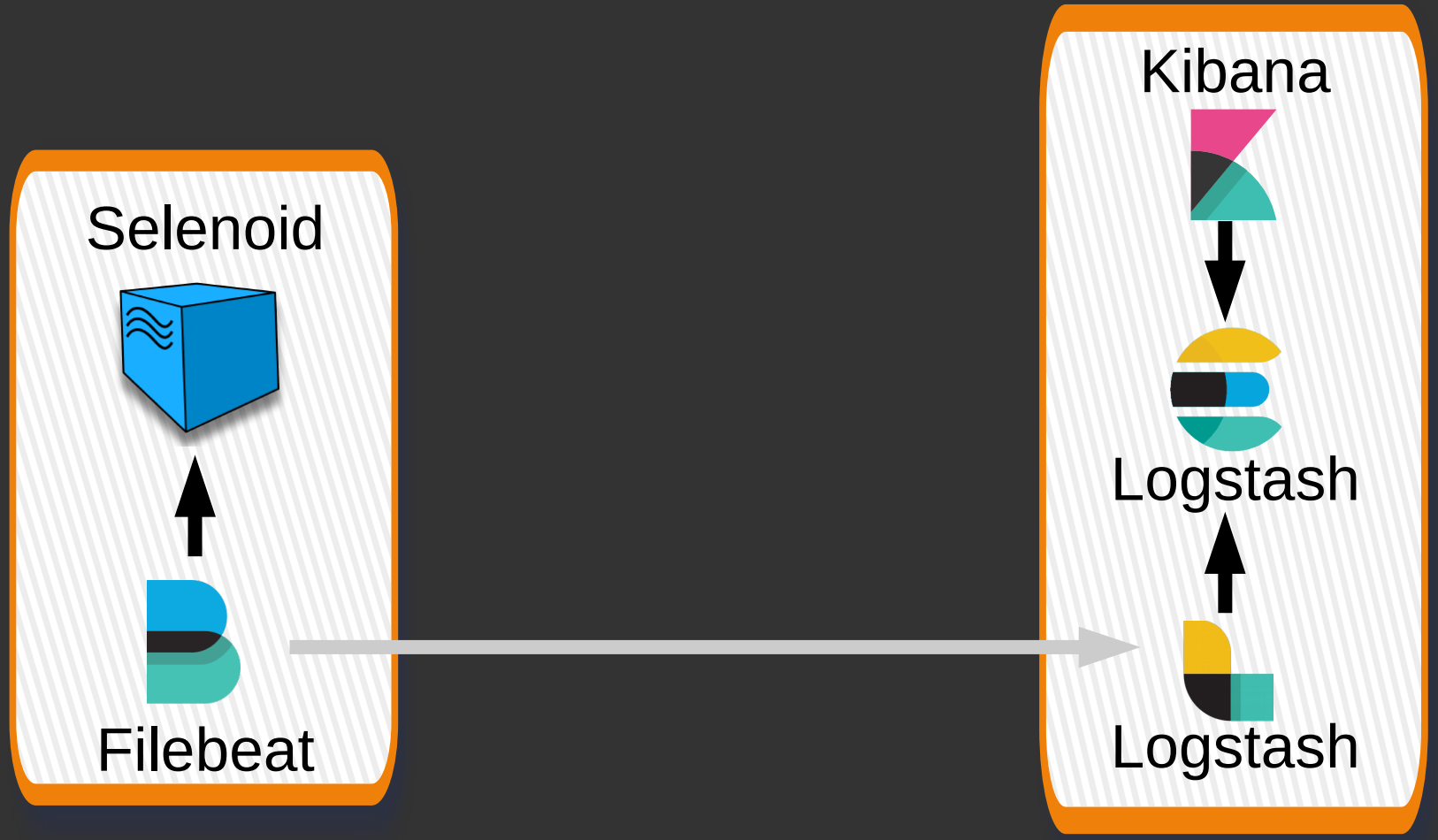


Building Windows image (step 3/3)



http://github.com/aerokube/windows-images


Selenoid
Grafana
Telegraf
InfluxDB


Selenoid
Filebeat
Kibana
Logstash
Logstash