Docker /Maven / Cucumber

pump up your productivity !!

Illia Izotov

Nicolas Bobo

Who are we ?

Illia Izotov
i.izotov@criteo.com

Nicolas Bobo
n.bobo@criteo.com

Who are you ?

We will see

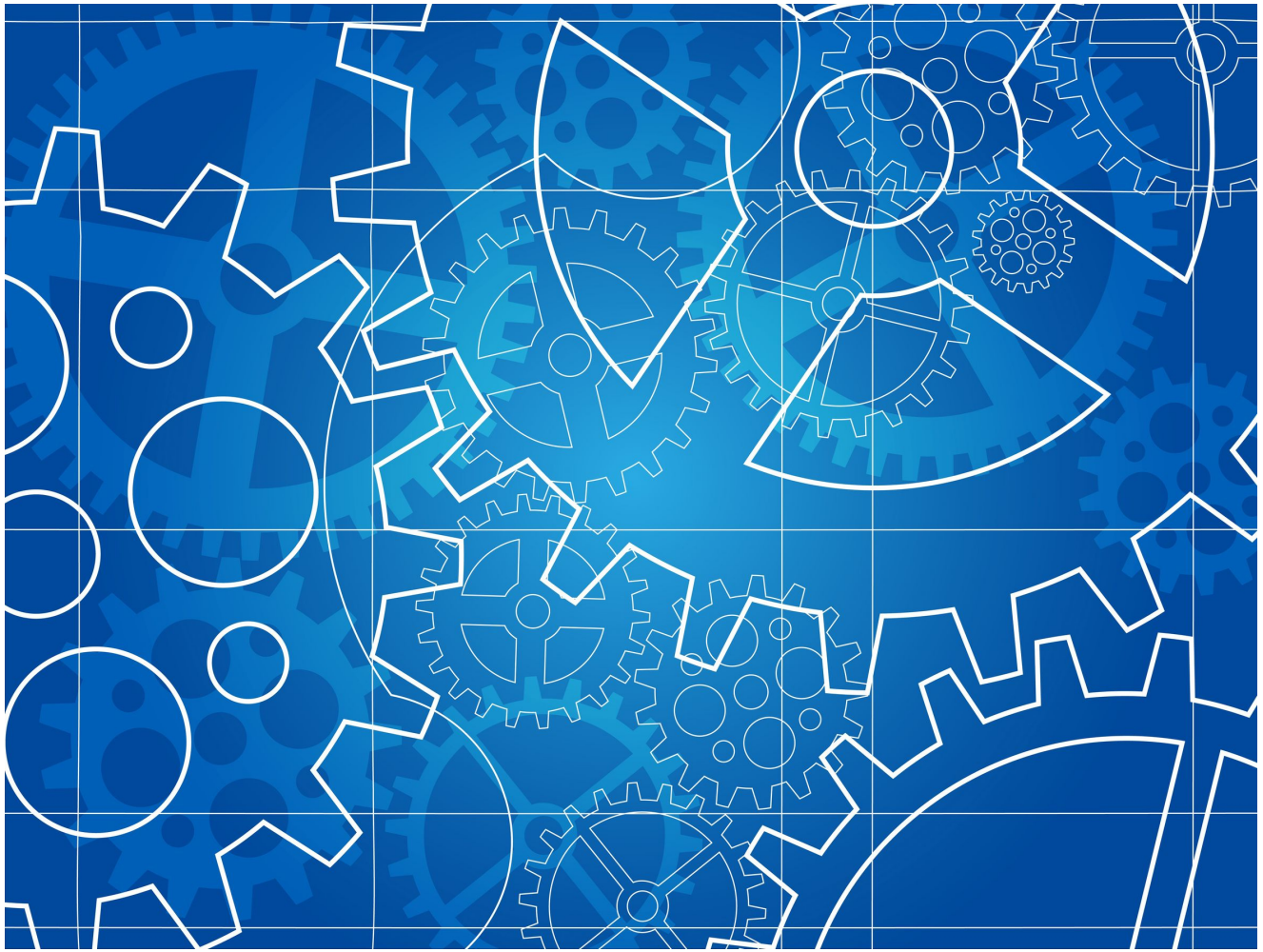
- technical stuff
- orchestrate deployement
- reciepes

We will not see

- maven / docker training
- docker on production
- how to do a backlog

Origin of this project

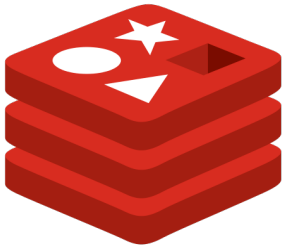
Technical stack



Frameworks

Symfony

APACHE
STORM™
Distributed • Resilient • Real-time

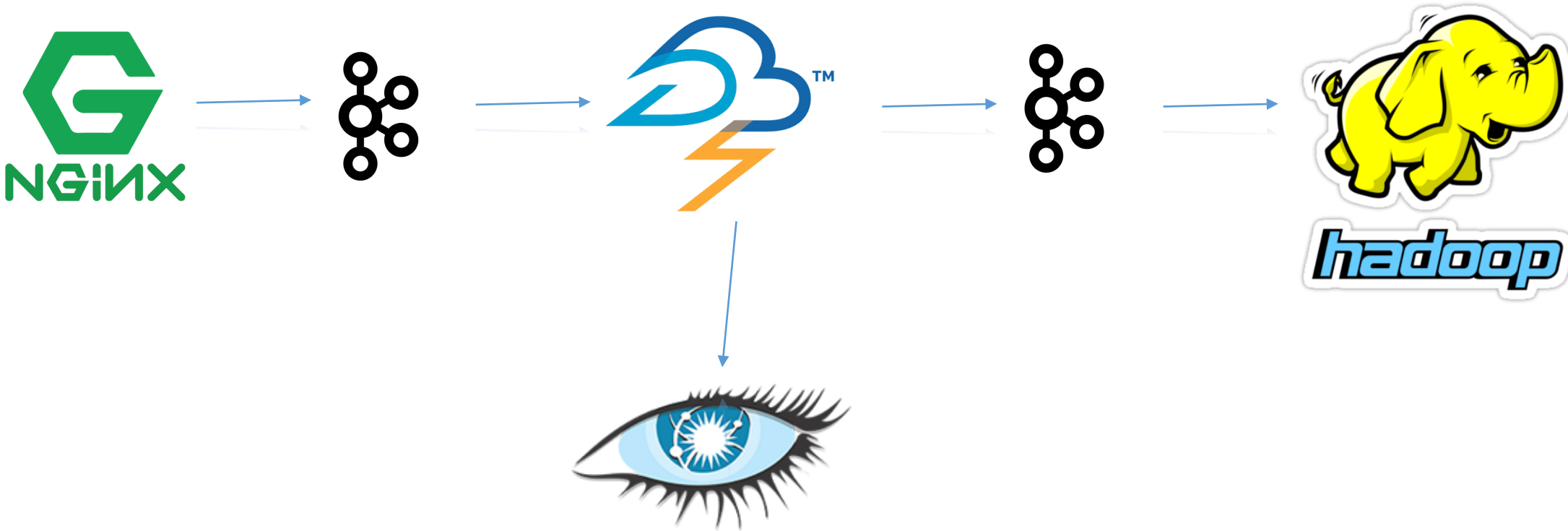


redis cassandra

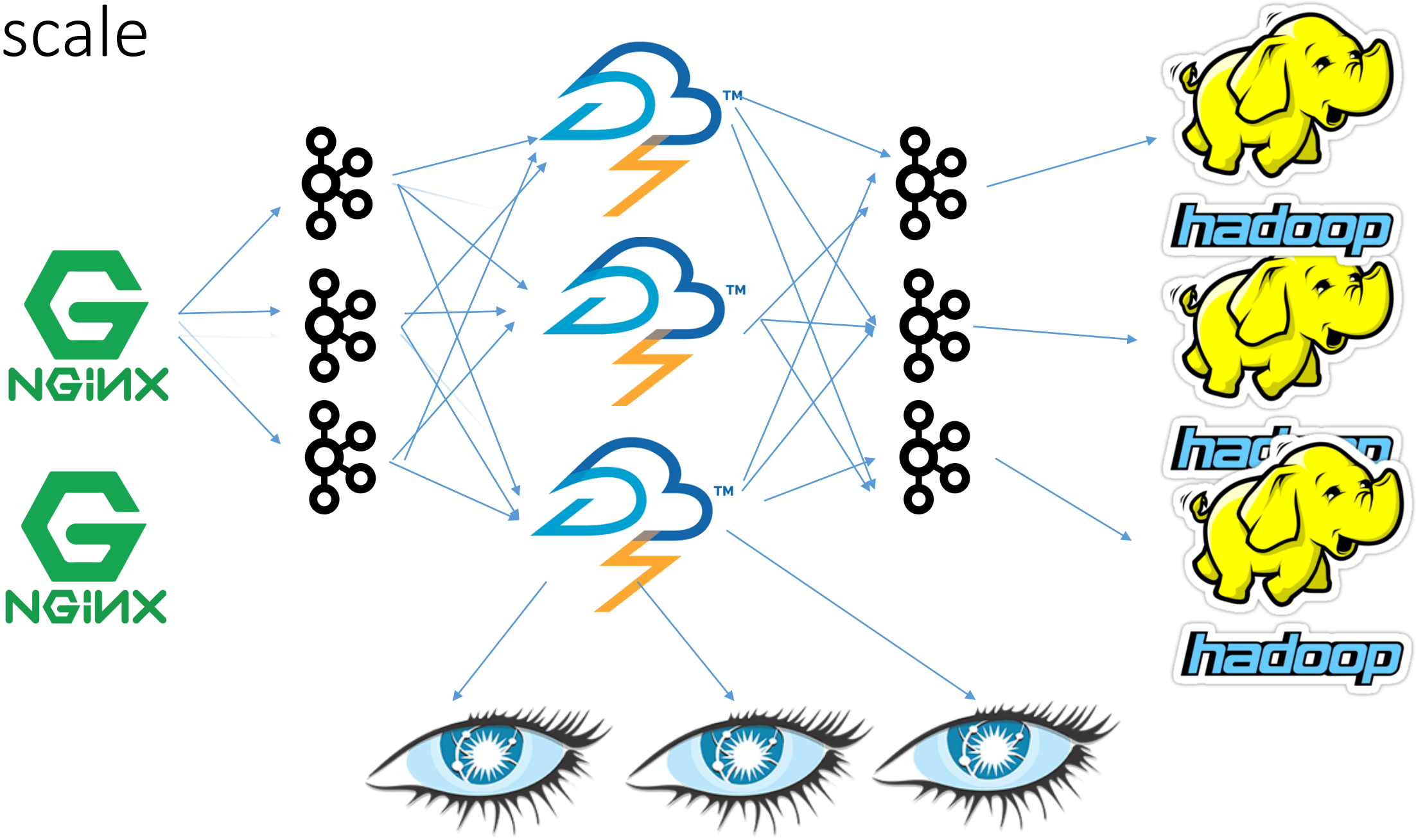
jetty://

Languages


INFRASTRUCTURE


NGINX
hadoop

@scale



Problematic

- ✓ My code base is growing fast (many developpers)
- ✓ Manual integration tests – Less and less maintainable
- ✓ Manual db setup – More and more side effects
- ✓ Flaky tests

- ✓ Difficulties to release
- ✓ Miss of confidence in our releases
- ✓ Big infra
- ✓ Difficulty to test and debug locally


KEEP
CALM
AND
STOP
THE
LINE

We need a strong test harness

- ✓ Unit tests
- ✓ Integration tests
- ✓ Component tests
- ✓ End to end tests

Integration tests vs End to end tests vs Component tests

https://martinfowler.com/articles/microservice-testing/

By Toby Clemson

Unit testing *alone doesn't provide guarantees about the behaviour of the system*

https://martinfowler.com/articles/microservice-testing/

*An **integration test** verifies the communication paths and interactions between components to detect interface defects.*

https://martinfowler.com/articles/microservice-testing/

*A **component test** limits the scope of the exercised software to a portion of the system under test, manipulating the system through internal code interfaces and using test doubles to isolate the code under test from other components.*

https://martinfowler.com/articles/microservice-testing/

*An **end-to-end test** verifies that a system meets external requirements and achieves its goals, testing the entire system, from end to end*

https://martinfowler.com/articles/microservice-testing/


YOU TAKE
THE BLUE PILL
THE STORY ENDS
YOU WAKE UP IN YOUR BED
AND BELIEVE WHATEVER
YOU WANT TO BELIEVE.
YOU TAKE
THE RED PILL
YOU STAY IN WONDERLAND
AND I SHOW YOU
HOW DEEP THE
THE RABBIT-HOLE GOES.

4 categories of usecases identified

- ✓ Must
- ✓ Should
- ✓ Nice to have
- ✓ Must not

Must

- ✓ Give ability to developer to run integration tests locally
- ✓ Give ability to developer to debug locally
- ✓ Give back right feedback
- ✓ Allow to easily run a single test and not a whole test suite run with
  Not flaky

Should

- ✓ Report the errors as close as possible to their root cause
- ✓ Easy to use
- ✓ Run test quickly, make them parallelizible
- ✓ Give ability to developer to run e2e tests locally
- ✓ Allow any kind of test framework (junit, testNG, cucumber, scalatest...)
- ✓ Send alert in case of failure

Should

- ✓ Run integration test from CI manually
- ✓ Run only nominal integration test
- ✓ Be able to deploy remote cluster
- ✓ Easy to deploy locally
- ✓ Do not duplicate docker file

Nice to have

- ✓ Generate fancy reports (HTML...)
- ✓ Coverage

MUST NOT DO

- ✓ Be unmaintainable
- ✓ Be not used by developper
- ✓ Dump a big dataset
- ✓ Be flaky by design
- ✓ Use more ram than what we have on our laptop

Solution

We are craftmen

E2e, ComponentTests and IT should be written by developpers

We need tools

Docker / Maven / fabric8 / Cucumber match the Usecases identified

In summary

- ✓ Stop the line
- ✓ Developper agreement
- ✓ Define use case
- ✓ Choose the right technologies

A bit of theory


Maven™

PHASE A Build Lifecycle is Made Up of Phases

GOAL A Build Phase is Made Up of Plugin Goals

PROFILE Override pom confirguration adapt pom at runtime for one environment

PLUGINS plugins are where much of the real action is performed

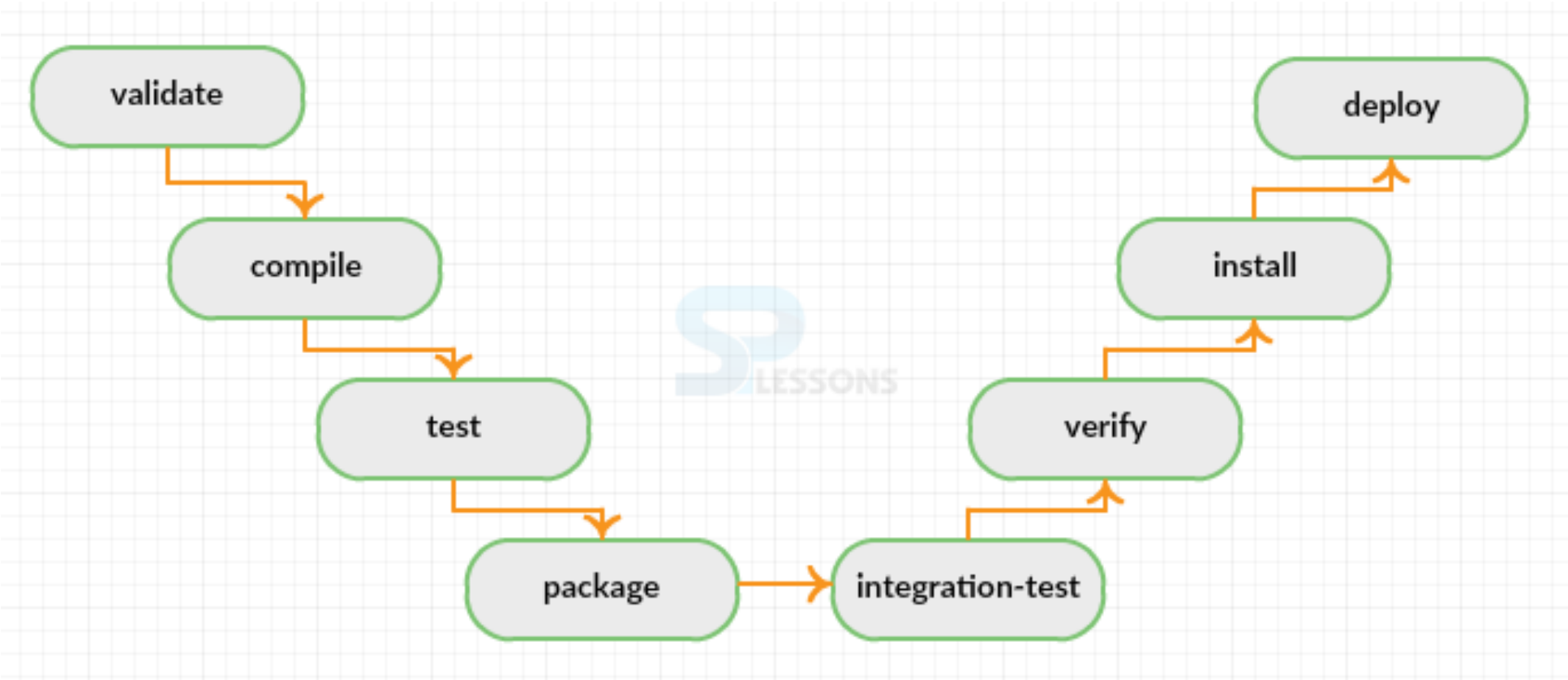
create jar files

compile code

unit test code

create project documentation

PHASES


validate
compile
test
package
integration-test
verify
install
deploy

We can invoke phase or plugin

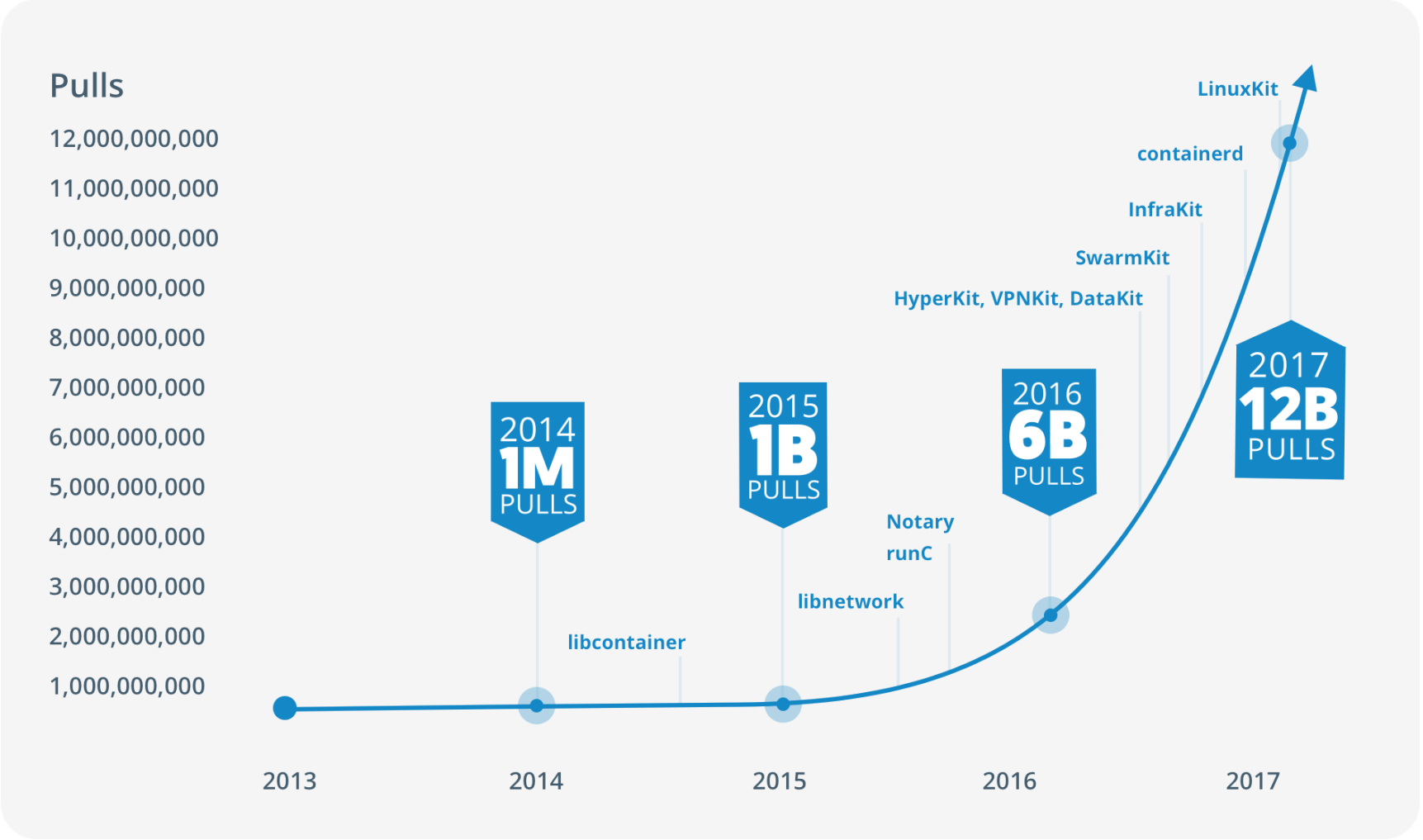
mvn *pluginId*:*goalId*

ie : mvn surefire:test

goal test of the plugin surfire
it is attached to the « test » phase


docker

Docker is a software technology providing containers, promoted by the company Docker, Inc.



Docker – what's inside?

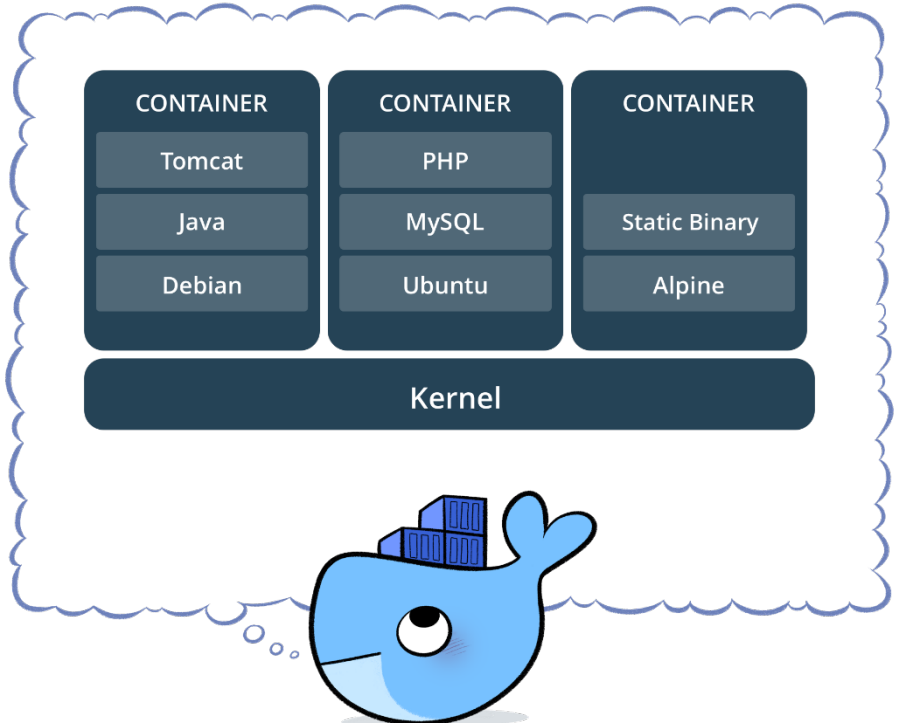
cgroups

LXC

Containerd

Docker

Docker-compose / Kubernetes / etc.



Basic terms

- Dockerfile
- Image
- DockerHub
- Container
- Volume
- Network
- Docker-compose

Container vs VM

Container

- Containers are an abstraction at the app layer that packages code and dependencies together.
- Lightweight

VM

- Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.
- Full OS emulation

fabric8

Being devOps doesn’t suck anymore!

Fabric8 maven plugin

fabric8io/docker-maven-plugin

This is a Maven plugin for managing Docker images and containers. It focuses on two major aspects for a Docker build integration:

- Building images
- Running containers

It provides seamless integration between low-level docker API and maven build phases.

Fabric8 docker maven plugin

Maven plugin invocation

mvn docker:build

mvn docker:start

mvn docker:stop


cucumber

“Behaviour” is a more useful word than “test”

Dan North

cucumber

A single source of truth

Cucumber merges specification and test documentation into one cohesive whole.

Living documentation

Because they're automatically tested by Cucumber, your specifications are always bang up-to-date.

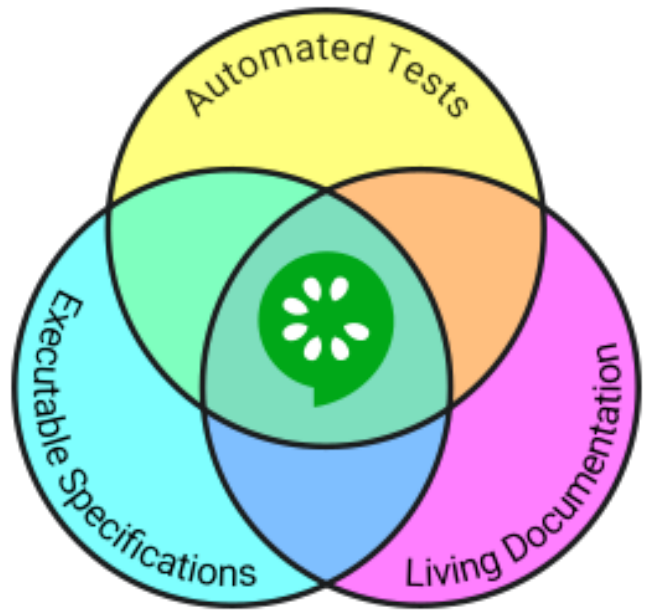
Focus on the customer

Business and IT don't always understand each other. Cucumber's *executable specifications* encourage closer collaboration, helping teams keep the business goal in mind at all times.

Less rework

When automated testing is this much fun, teams can easily protect themselves from costly regressions.

cucumber

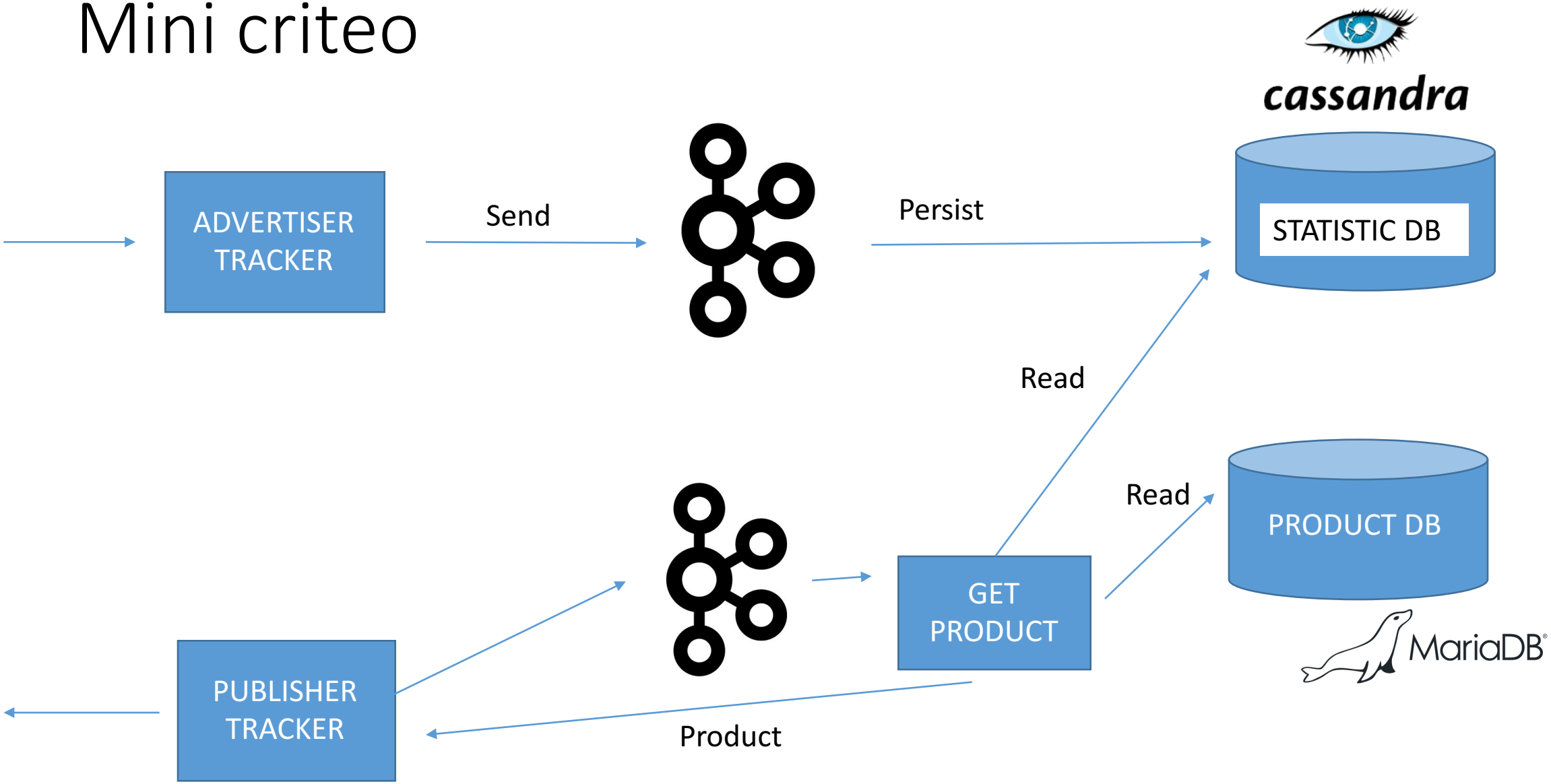

Automated Tests
Executable Specifications
Living Documentation

Jenkins

support building, deploying and automating any project.

Let’s practice

Mini criteo


ADVERTISER TRACKER
Send
Persist
cassandra
STATISTIC DB
Read
Read
PRODUCT DB
MariaDB
GET PRODUCT
PUBLISHER TRACKER
Product

Demo

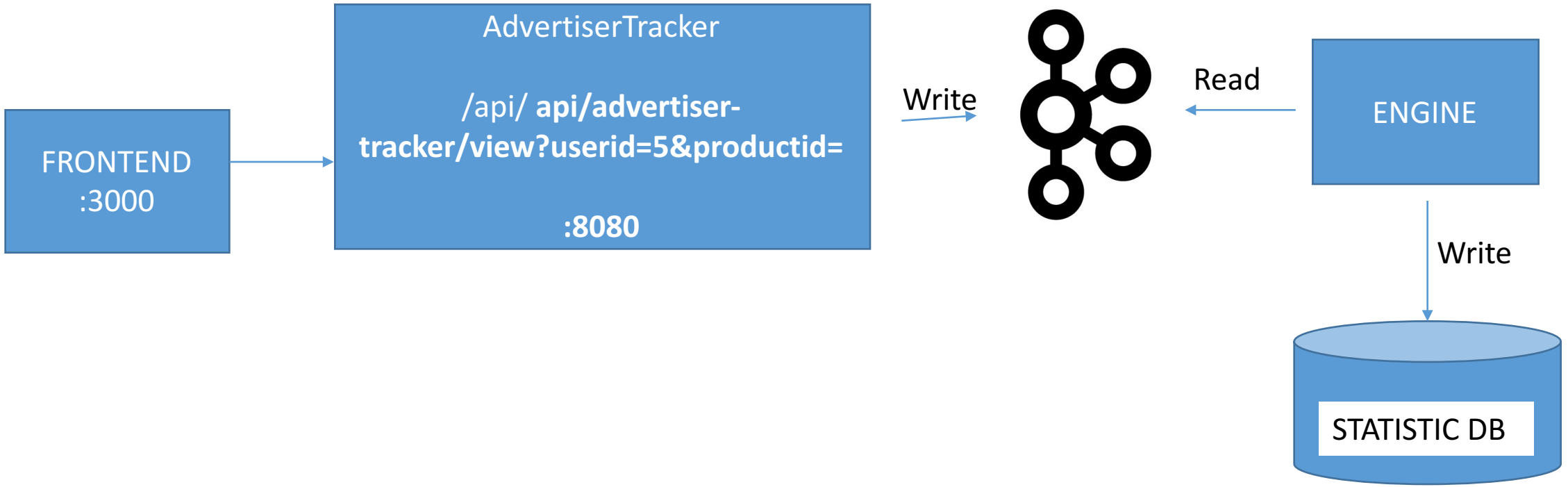
FRAC / Categories / ORDINATEURS PORTABLES

Retour

Dell XPS 13 9360 - 13.3" - Core i7 7500U - 8 Go RAM - 256 Go SSD - français

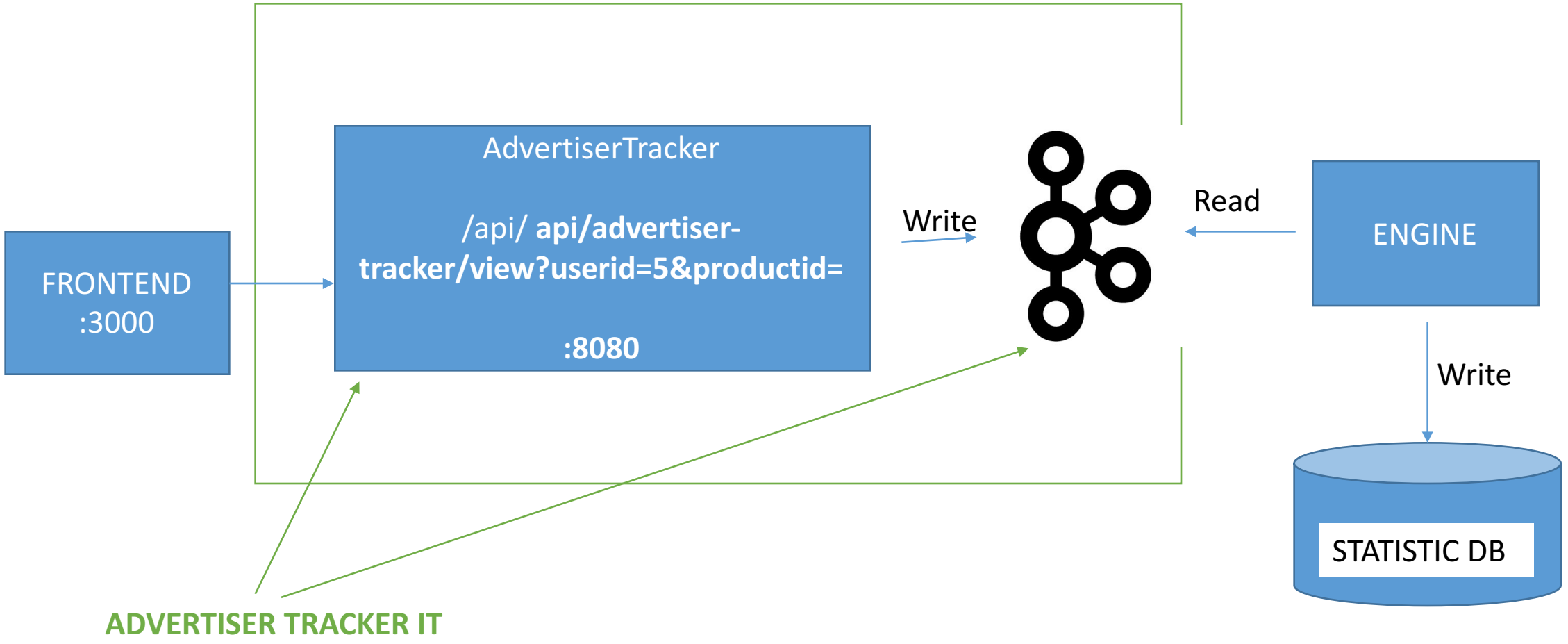
UseCase 1 : I want to run and debug my project on my dev machine

Let’s focus on advertiser

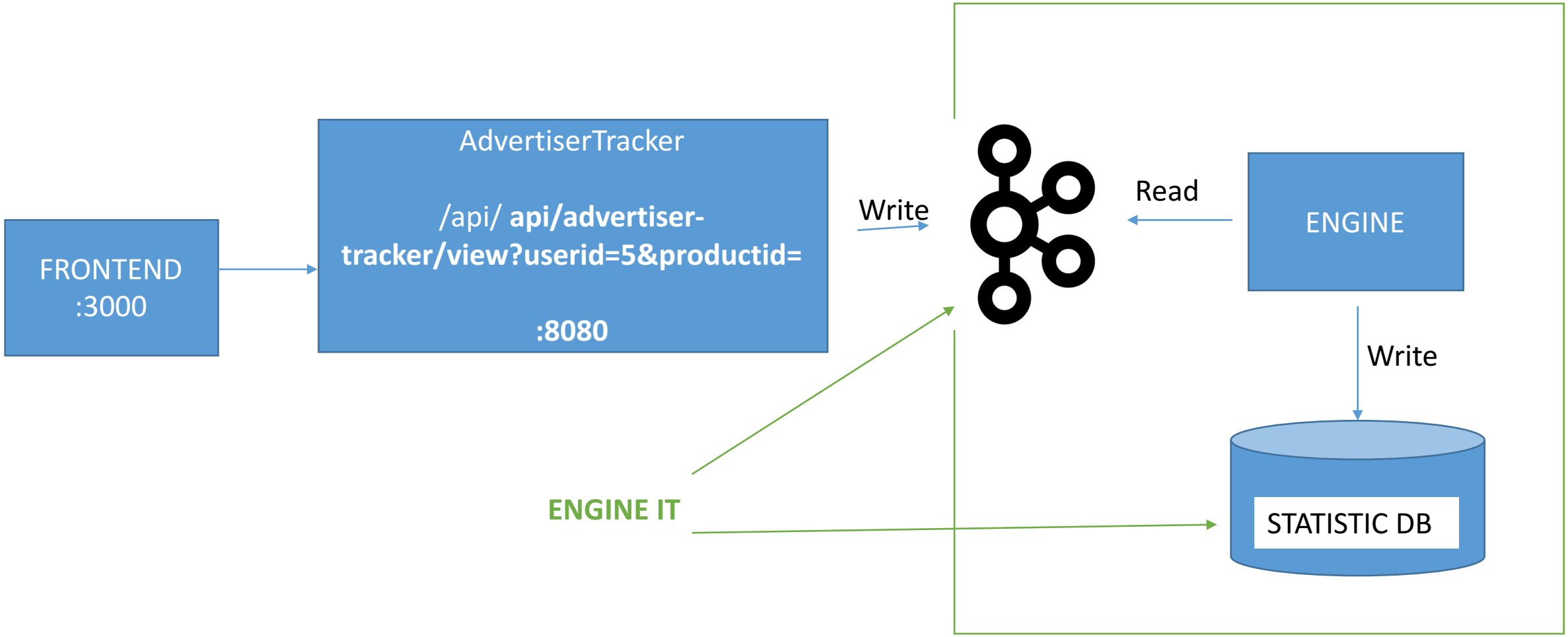


UseCase 2 : I want automatic Integration tests

Integration Test - AdvertiserTracker

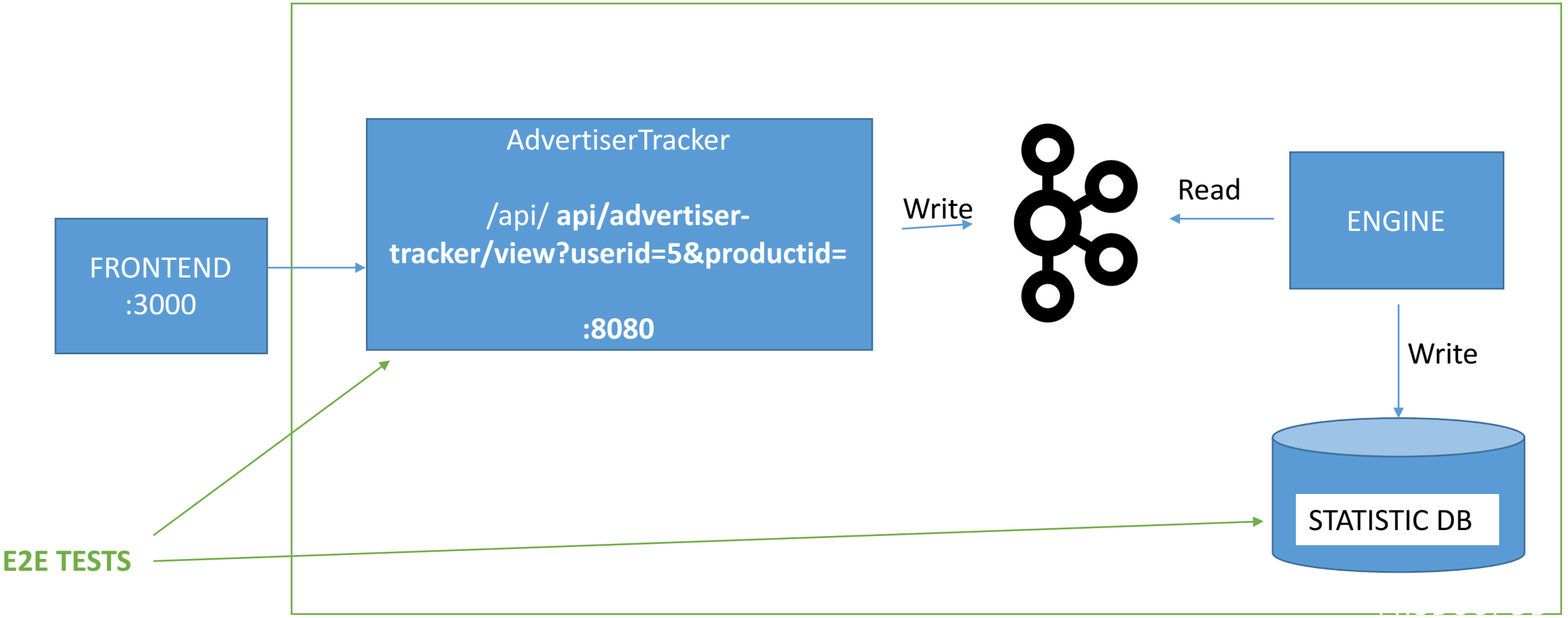


Integration Test - Engine



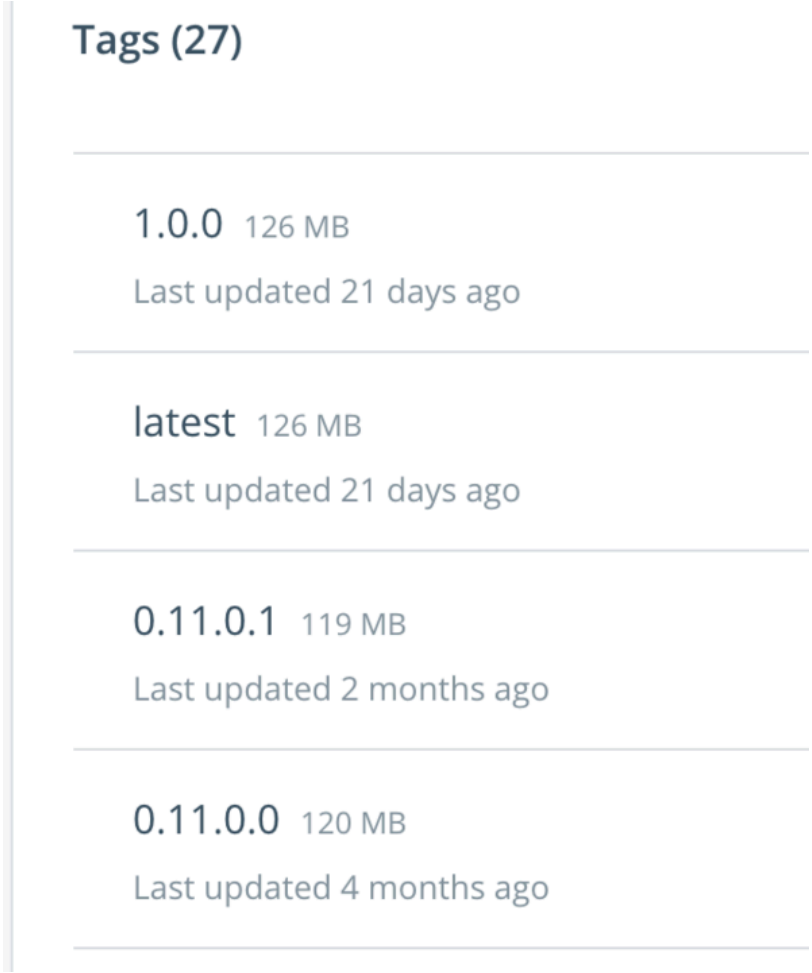
UseCase 3 : I want automatic e2e tests

E2E Test – AdvertiserTracker - Engine



UseCase 4 : component version is changing

https://store.docker.com/



https://github.com/astrohome/agile-grenoble-presentation-demo

Take away

Give ability to developer to debug locally

mvn docker:build

mvn docker:start

mvn docker:stop

Take away

Give ability to developer to run integration tests locally

mvn clean install

mvn verify

Take away

Allow to easily run a single test and not a whole test suite

```
mvn -Dit.test=ITIntegrationTest#testmyFirstIntegrationIT verify
```

http://maven.apache.org/surefire/maven-failsafe-plugin/examples/single-test.html

Thank you !!