



LTB 2020
Load Testing &
Benchmarking

PERFORMANCE TESTING AUTOMATION FOR CI/CD PIPELINES.



Henrik Rexed
Partner Solution Evangelist
Neotys

Who is this speaker?



- Henrik Rexed (knightloader)*
- 14 years of Performance engineering



- Technical Evangelist @Neotys
- Lego Fan
- Focus on technologies for Performance engineers

Agenda



01

The current challenge with Performance Testing

02

Make sure your tests are relevant

03

Let's test earlier

04

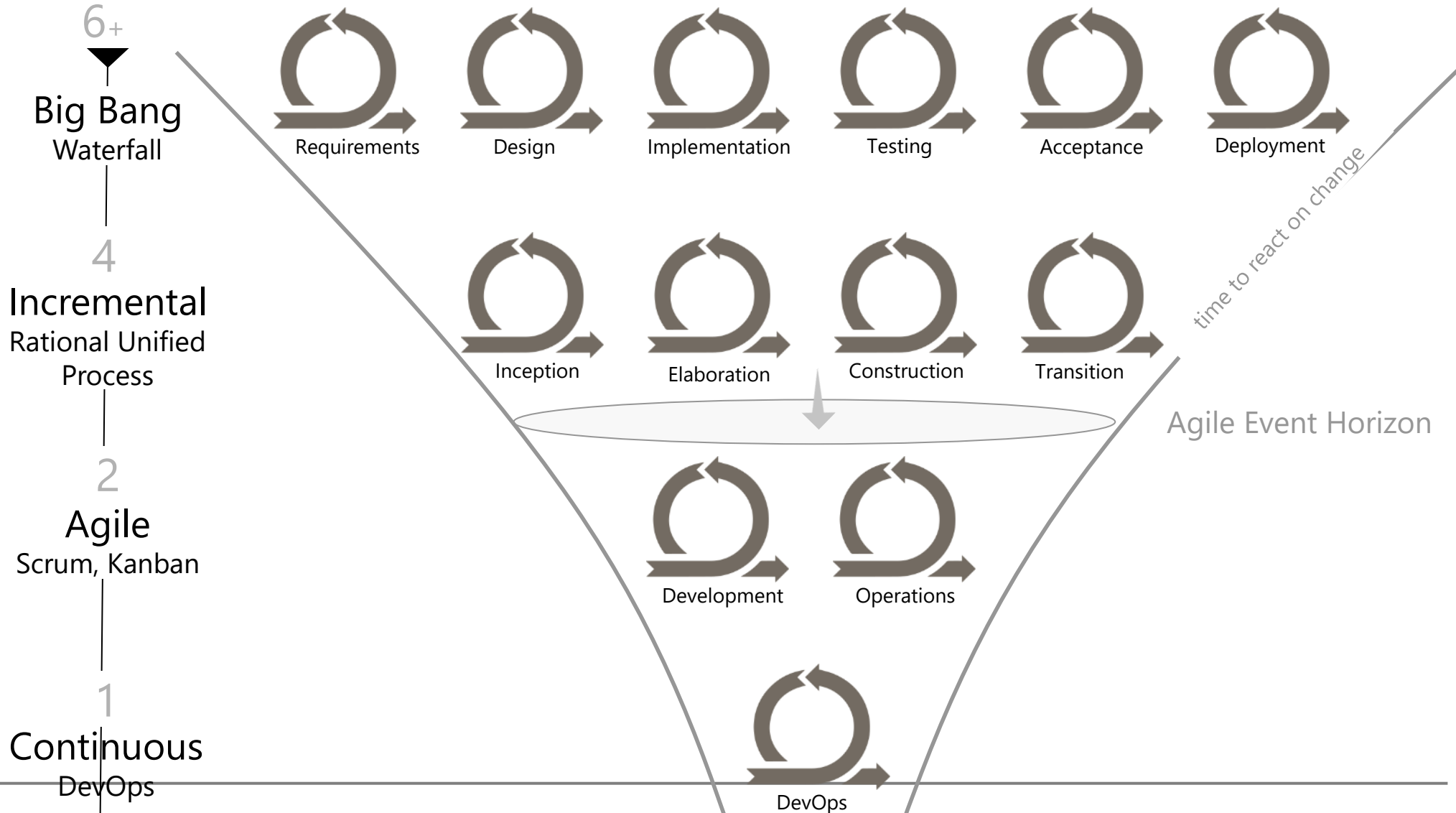
What are the SLI I should focus on?

05

Demo

We waited
30 min
NO SERVICE

Change of Methodology



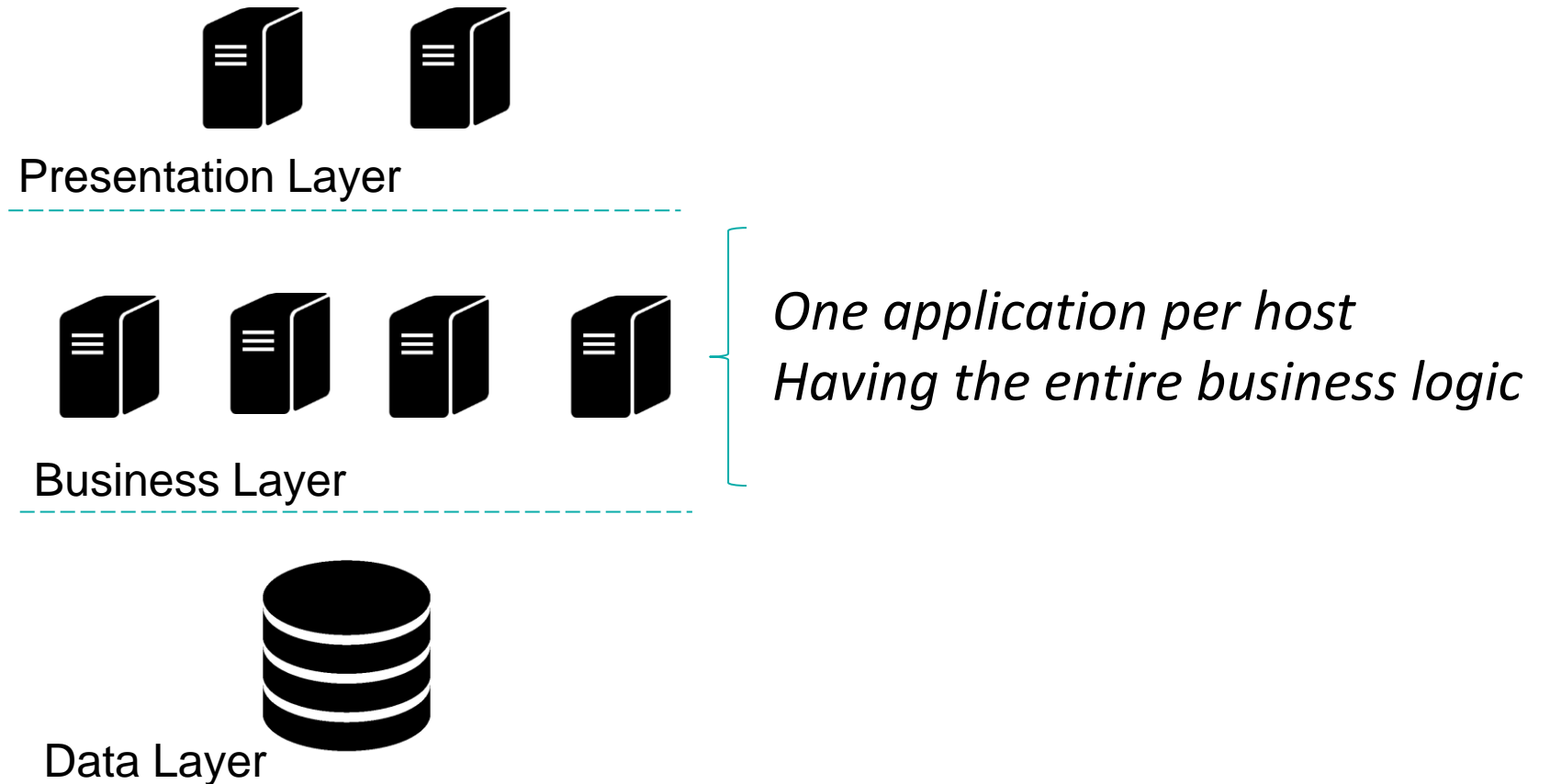
Performance testing is like going to the Gym

- Over 1 in 5 Americans belong to at least one U.S. health club or studio
- 12% gym members sign up in January
- Most of the people quit or stop going after 24 weeks
- 50% is going to find the soulmate



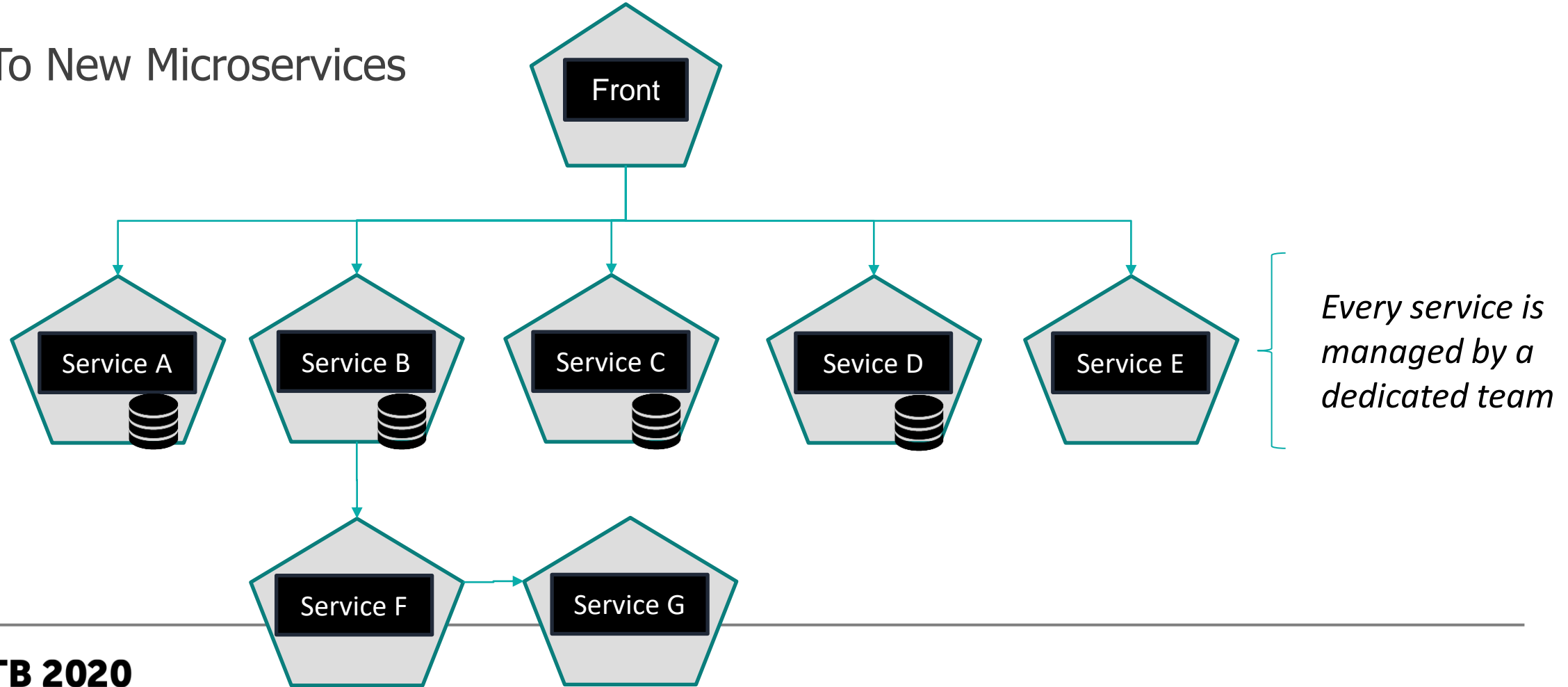
Ok but let's review our architecture first

- From Monolithic



Ok but let's review our architecture first

- To New Microservices



Let's move to the cloud

- Reduce the cost
- Increase the availability of the application
- Simplify, modernize

Yes! But



How can we resolve all those challenges?



Agenda



01

The current challenge with Performance Testing

02

Make sure your tests are relevant

03

Let's test earlier

04

What are the SLI I should focus on?

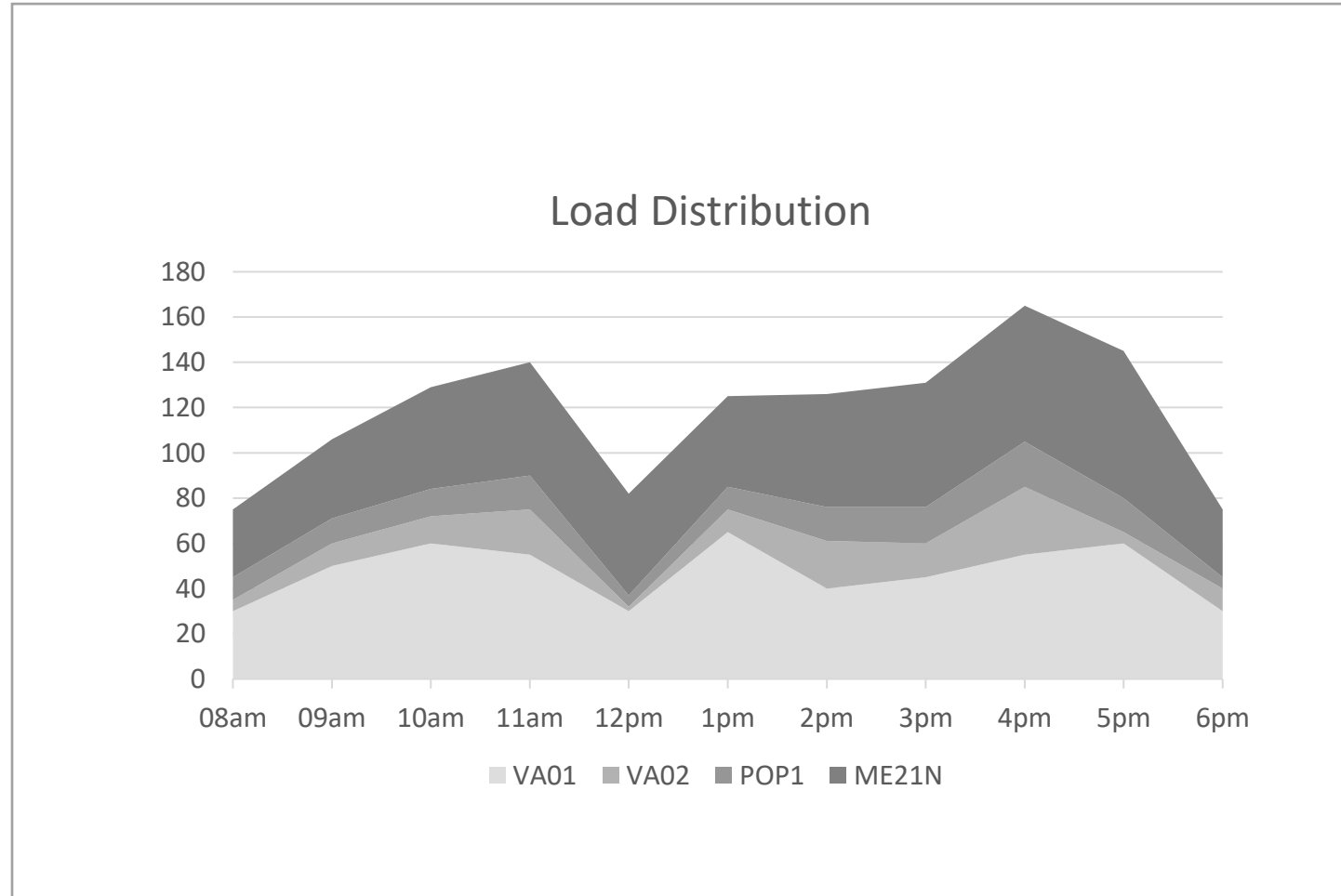
05

Q/A

Performance testing is like



Efficient and realistic workload model



Let's reuse monitoring data



Traditional
monitoring

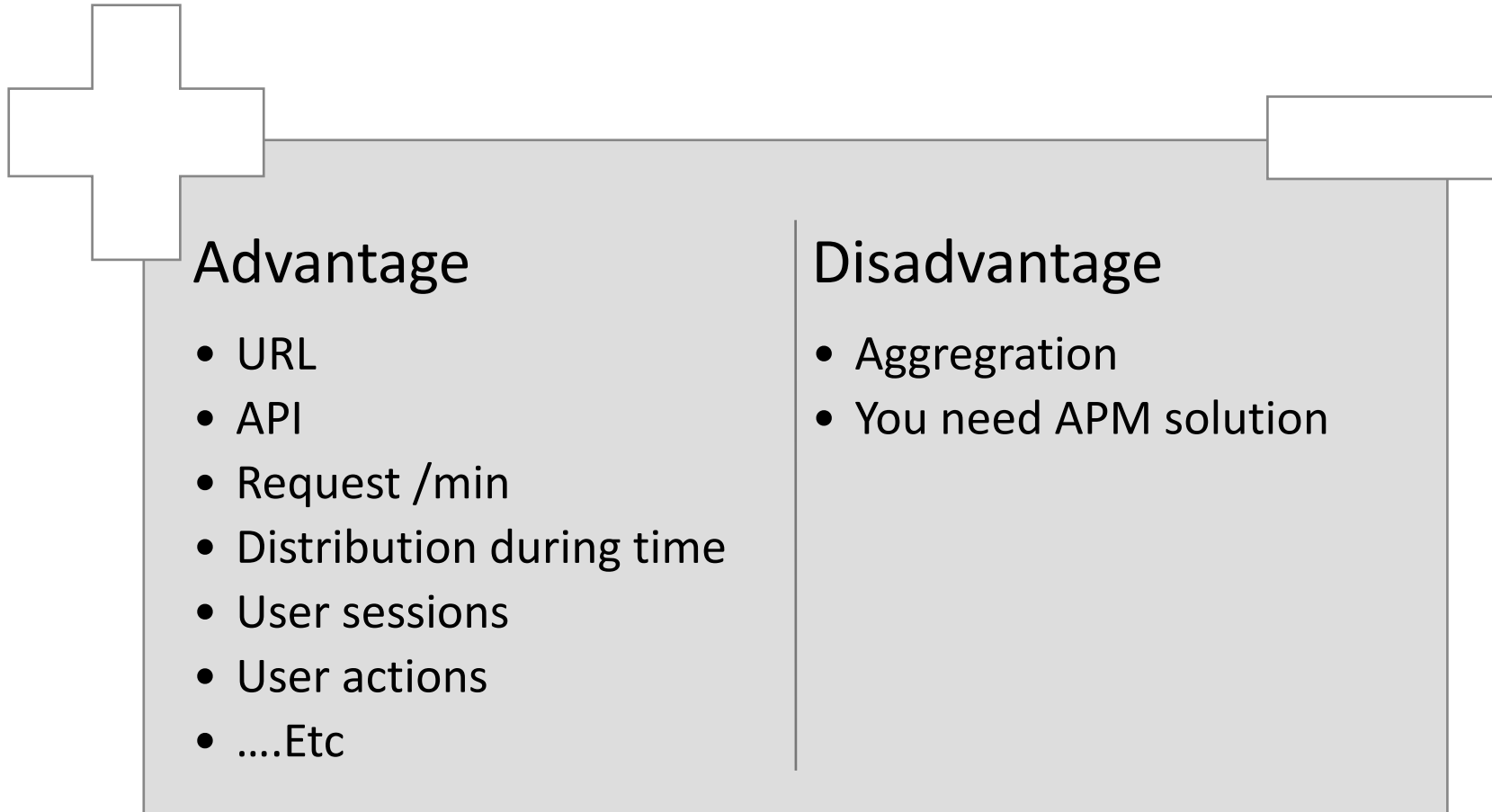


Tracking system



APM

Let's extract data from our APM to learn how our application is used



Agenda



01

The current challenge with Performance Testing

02

Make sure your tests are relevant

03

Let's test earlier

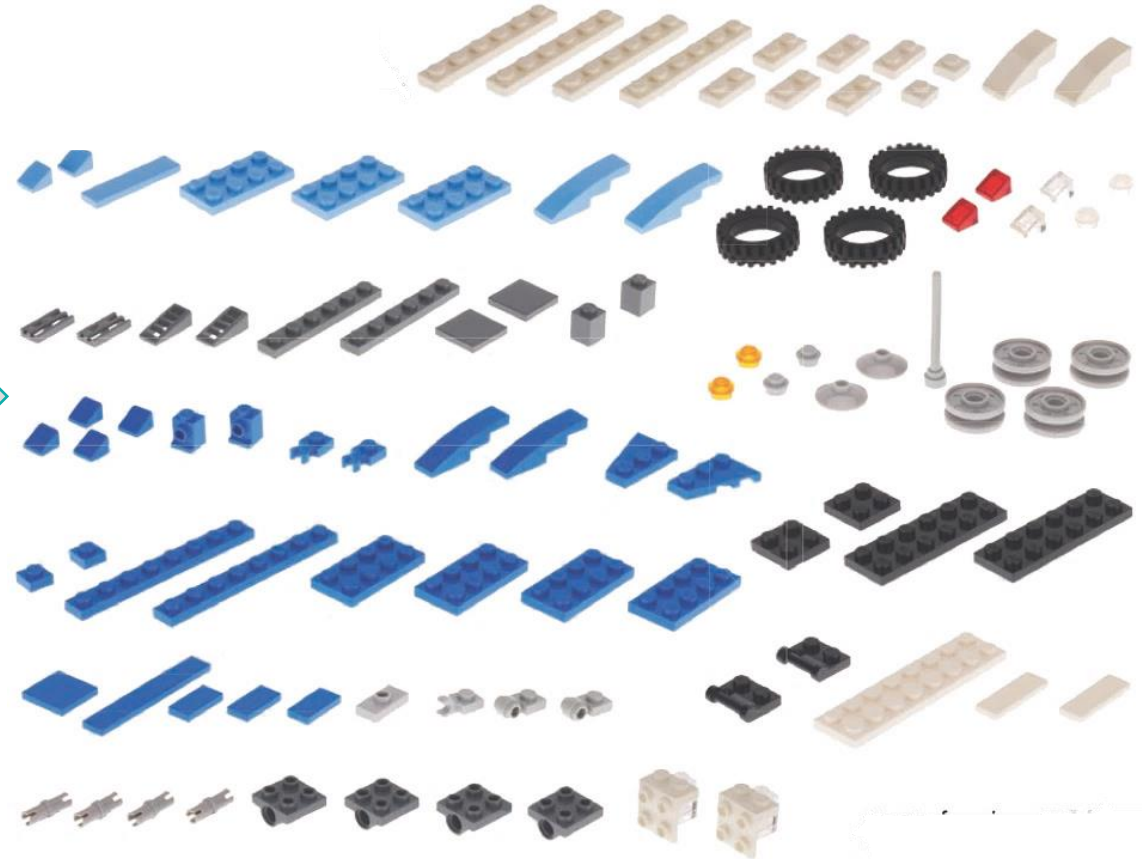
04

What are the SLI I should focus on?

05

Q/A

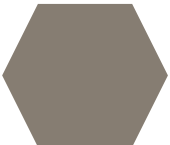
Architecture is Based on Components



Continuous Performance Testing

API Testing

Test a Component



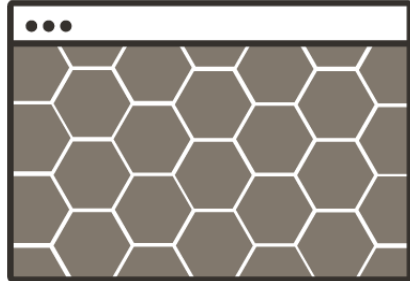
Integration Testing

Test a System

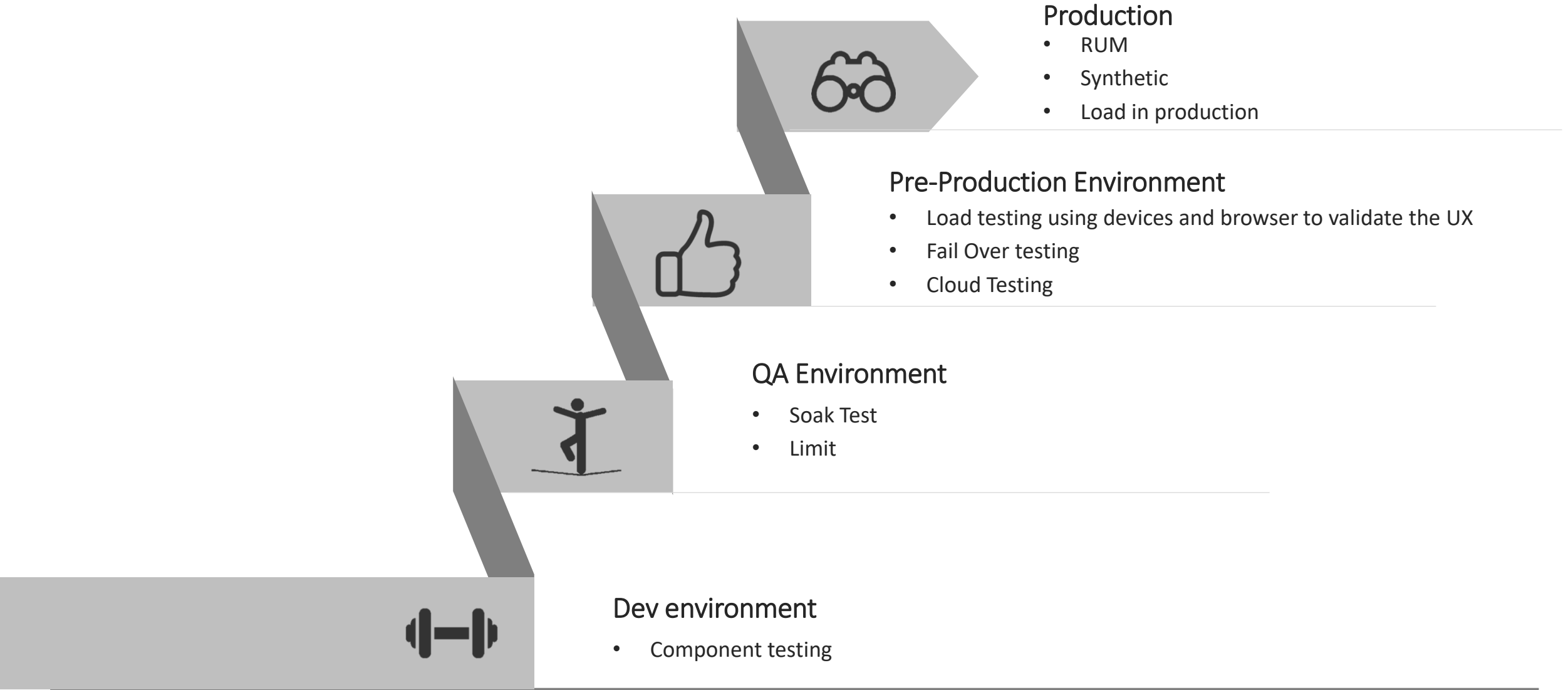


Application Testing

Test Real World



The complexity of the test depends of project life cycle



Agenda



01

The current challenge with Performance Testing

02

Make sure your tests are relevant

03

Let's test earlier

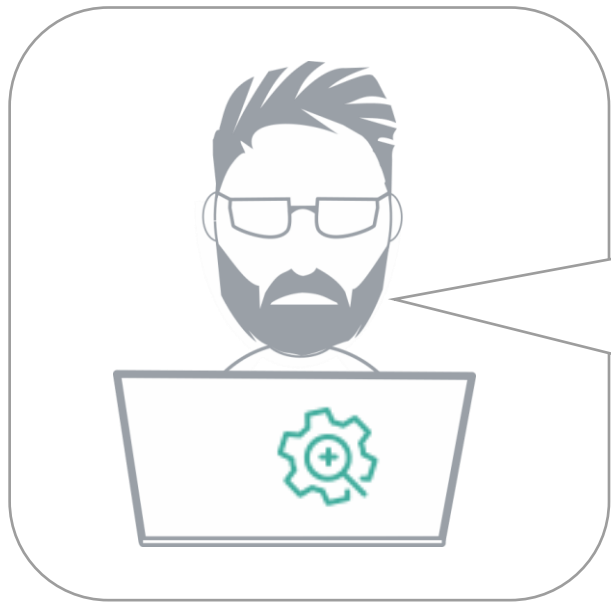
04

What are the SLI I should focus on?

05

Demo

Process to analyse



Test Failure



Error rate

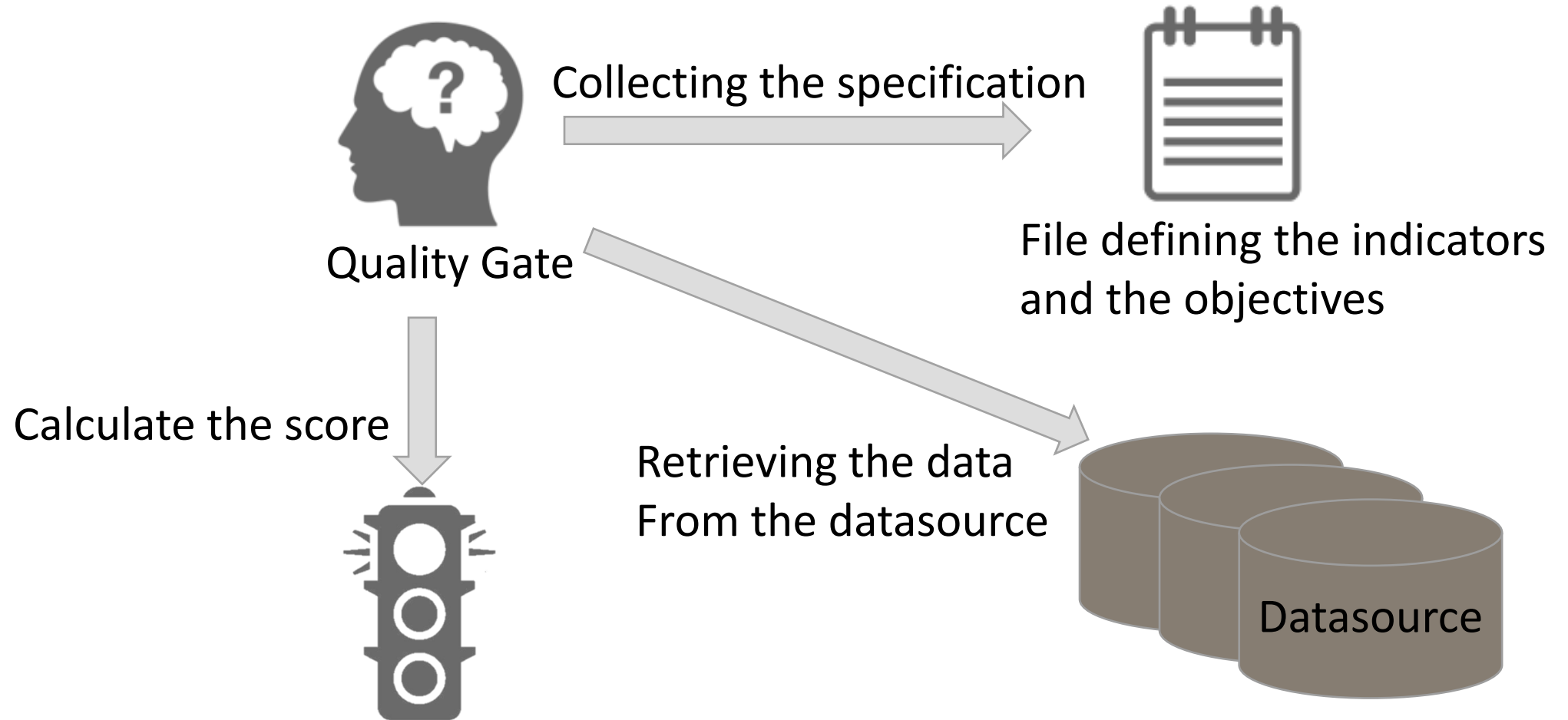


Throughput + Response Time versus objectives



Resources check

Quality Gate



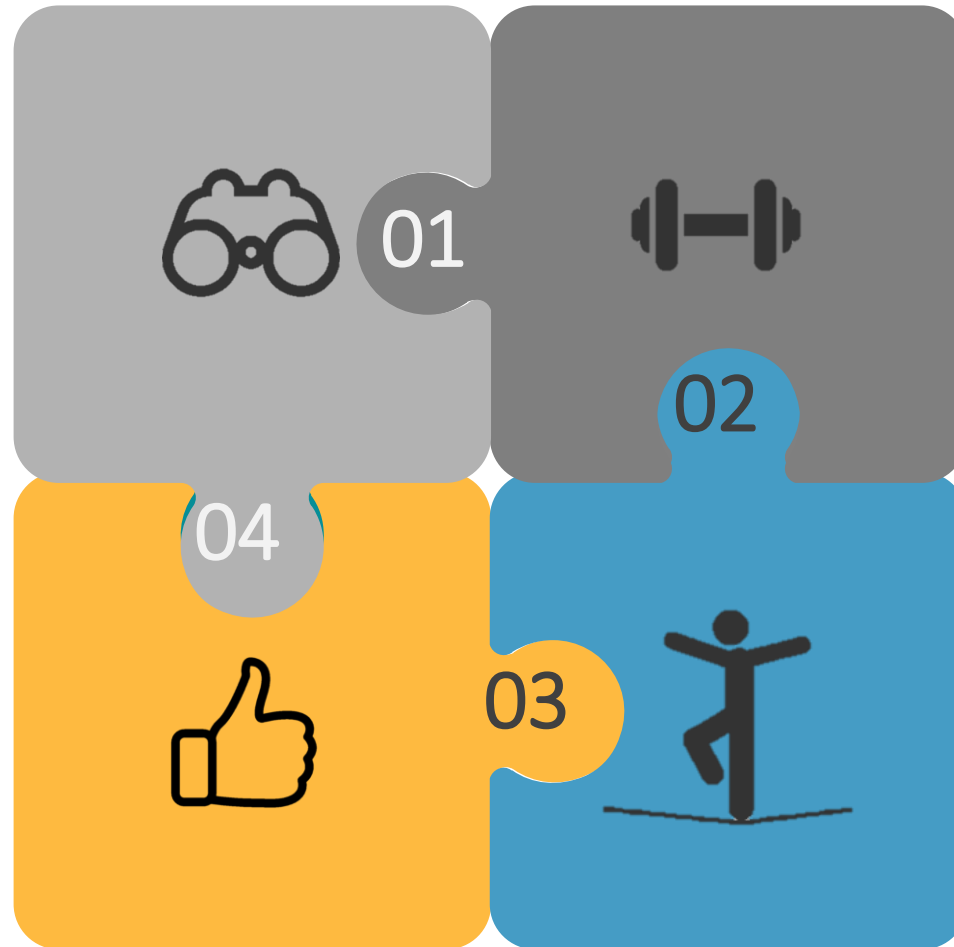
Not limit the evaluation only on response time

Production

- Rendering time
- Number of clicks to achieve an action
- Average visit time
- IO
- Number of Pods utilized

User Experience

- On a representative environment:
- ~~Response Time~~ Rendering time
- Number of container Pods/utilized
- The behavior of the network
- ..etc



Regression

Detecting regression by looking at:

- Response time
- Hit/s
- Memory used
- Cpu

Scalability

If it's a cloud native architecture :

- Number of containers/Pods utilized
- The memory used
- CPU
- Response time

Agenda



01

The current challenge with Performance Testing

02

Make sure your tests are relevant

03

Let's test earlier

04

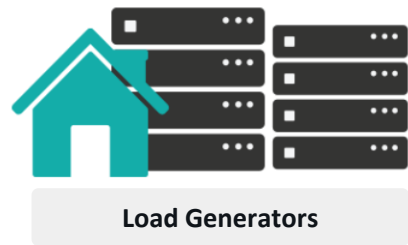
What are the SLI I should focus on?

05

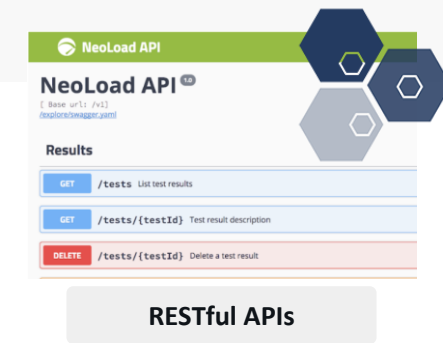
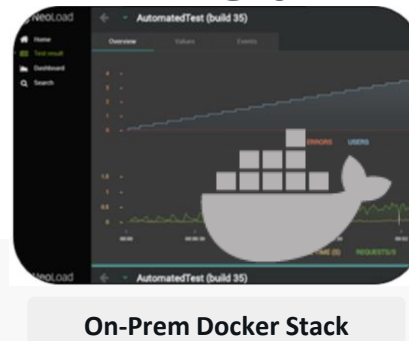
Demo

NeoLoad's platform

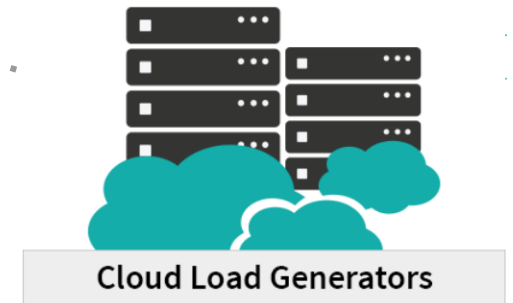
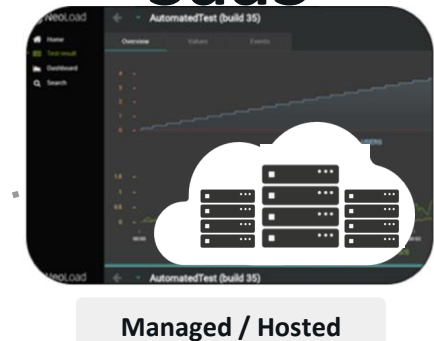
NeoLoad
Core



NeoLoad
Web



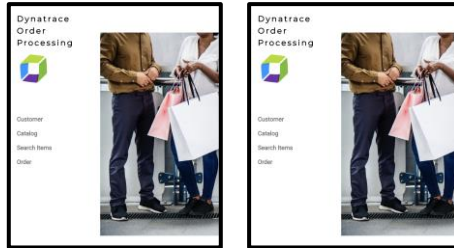
NeoLoad
SaaS



What we prepared for you today

Kubernetes cluster

Staging Production



<https://github.com/keptn-orders/>



Docker Registry with pre-built app images



GitHub Org

Keptn Repo for project files



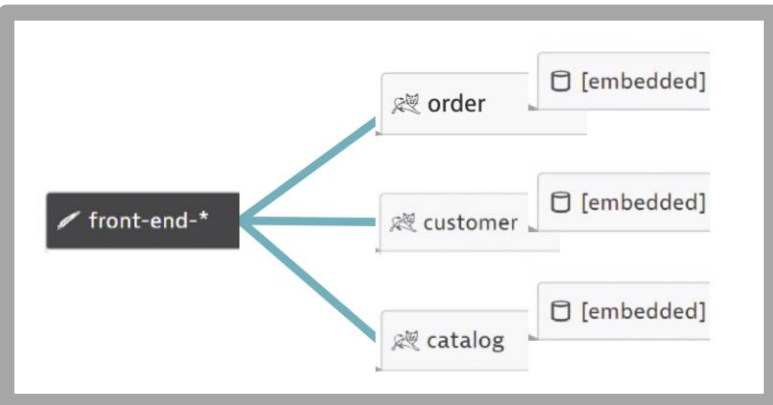
NeoLoad

Test Orchestration & Test History



dynatrace

SaaS Cluster



LTB 2020

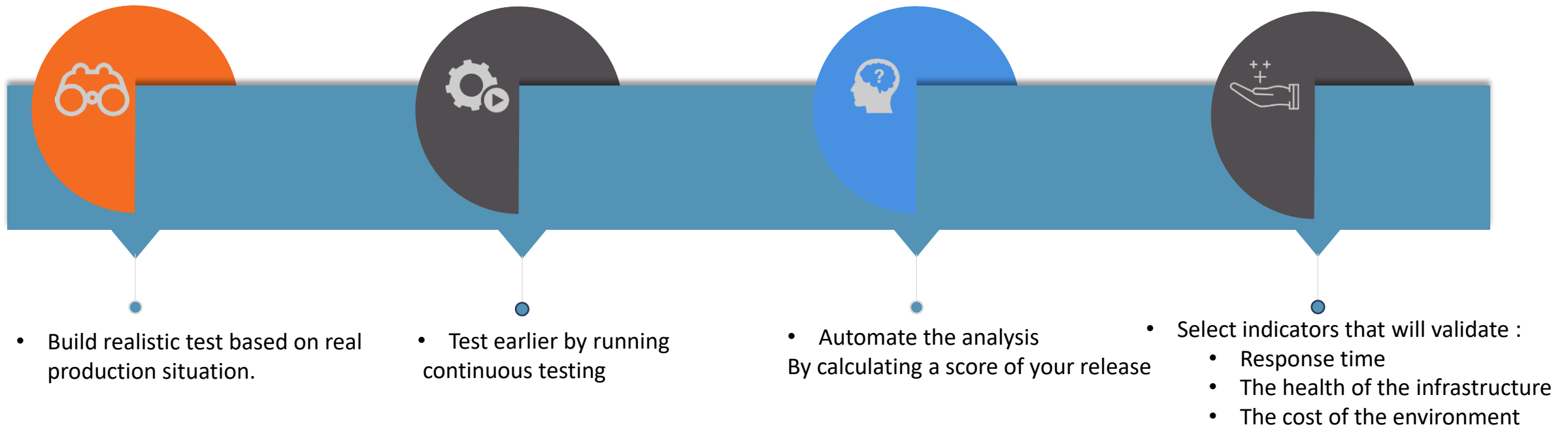
Load Testing & Benchmarking



Demo time



Take aways





LTB 2020
Load Testing &
Benchmarking

QA