zalando

ALERTING MONITORING

AND ALL THAT JAZZ

Luis Mineiro @voidmaze
SRE @ Zalando

Coding Serbia, 16.05.2019

ZALANDO AT A GLANCE

~ 5.4 billion EUR

revenue 2018

> 300 million visits per month

> 27 million

active customers

> 400,000

product choices

> 15,500

employees in Europe

> 80%

of visits via mobile devices

~ 2,000

brands

17

countries

as of March 2019

ZALANDO OFFICES

1 BERLIN **HEADQUARTERS**

2 ERFURT **TECH OFFICE**

3 MÖNCHENGLADBACH **TECH OFFICE**

4 DORTMUND **TECH HUB**

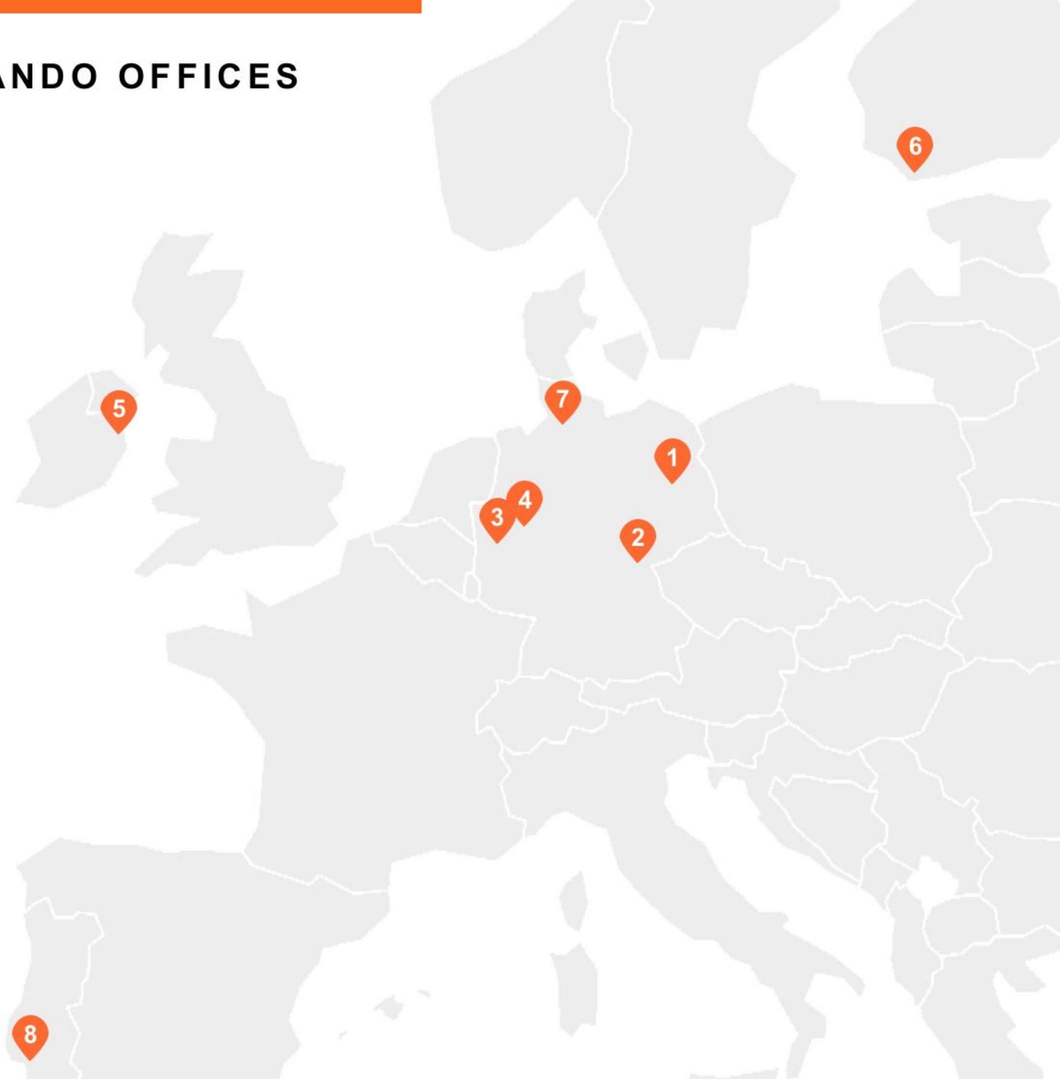
5 DUBLIN **TECH HUB**

6 HELSINKI **TECH HUB**

7 HAMBURG **ADTECH LAB**

8 LISBON **TECH HUB**

as of March 2019



WE ARE CONSTANTLY INNOVATING TECHNOLOGY

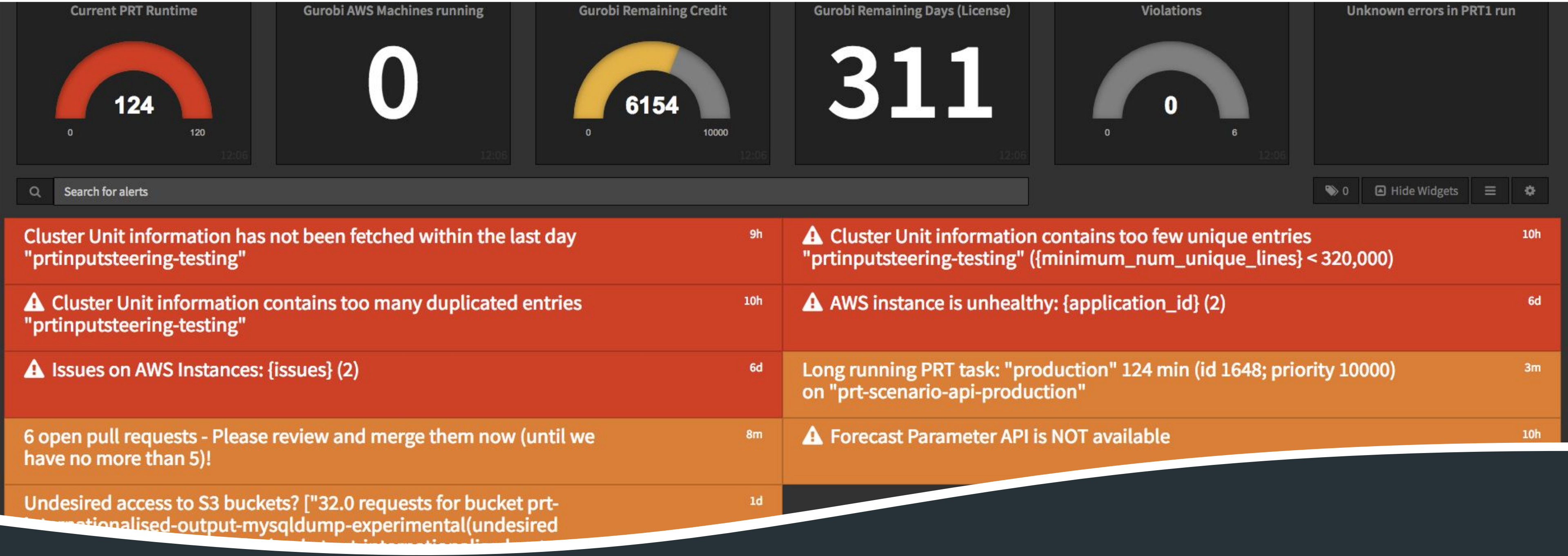
HOME-BREWED, CUTTING-EDGE & SCALABLE
technology solutions

help our brand to
WIN ONLINE

> 2,000
employees at

8 international
tech locations

HQs
in Berlin



Current PRT Runtime: 124 (0–120)

Gurobi AWS Machines running: 0

Gurobi Remaining Credit: 6154 (0–10000)

Gurobi Remaining Days (License): 311

Violations: 0 (0–6)

Unknown errors in PRT1 run

Cluster Unit information has not been fetched within the last day "prtinputsteering-testing" 9h

⚠ Cluster Unit information contains too few unique entries "prtinputsteering-testing" ({minimum_num_unique_lines} < 320,000) 10h

⚠ Cluster Unit information contains too many duplicated entries "prtinputsteering-testing" 10h

⚠ AWS instance is unhealthy: {application_id} (2) 6d

⚠ Issues on AWS Instances: {issues} (2) 6d

Long running PRT task: "production" 124 min (id 1648; priority 10000) on "prt-scenario-api-production" 3m

6 open pull requests - Please review and merge them now (until we have no more than 5)! 8m

⚠ Forecast Parameter API is NOT available 10h

Undesired access to S3 buckets? ["32.0 requests for bucket prt-internationalised-output-mysqldump-experimental(undesired 1d

Looks familiar?

TERMINOLOGY

MONITORING

Collecting, processing, aggregating, and displaying real-time quantitative data about a system, such as query counts and types, error counts and types, processing times, and server lifetimes.

ALERT

A notification intended to be read by a human and that is pushed to a system such as a bug or ticket queue, an email alias, or a pager.

SRE Book, Chapter 6: Monitoring Distributed Systems

MONITORING

Your monitoring system should address two questions: **what’s broken**, and **why**?

The "what’s broken" indicates the **symptom**; the "why" indicates a (possibly intermediate) **cause**.

"**What**" versus "**why**" is one of the most important distinctions in writing good monitoring with maximum signal and minimum noise.

SRE Book, Chapter 6: Monitoring Distributed Systems

ALERTING CLASSIFICATION

Urgency	**Name**	**Delivery**
Will be addressed... eventually	Report	Dashboards or nowhere (/dev/null)
Predicted to fail "soon"	Ticket	An issue tracker or *cough*, Email
Urgently and actively get the attention of a specific human	Page	A pager, cell phone or something going *beep* *beep*

WHAT TO ALERT ON

Alerting should be both **hard failure–centric** and **human-centric**.

Distributed Systems Observability e-Book, Chapter 2: Monitoring and Observability

Symptoms are a better way to capture more problems more comprehensively and robustly with less effort - "**symptom-based monitoring**," in contrast to **"cause-based monitoring"**.

Rob Ewaschuk, "My Philosophy on Alerting"

Keep alerting simple, **alert on symptoms**. Aim to **have as few alerts as possible**, by alerting on symptoms that are associated with end-user pain rather than trying to catch every possible way that pain could be caused.

Prometheus Best Practices, https://prometheus.io/docs/practices/alerting/

SERVICE LEVEL OBJECTIVES

You should pick SLOs that represent the **most critical aspects of the user experience**.

Google Cloud Platform Blog, Building good SLOs - CRE life lessons

Start by thinking about (or finding out!) **what your users care about**, not what you can measure.

Choose **just enough SLOs to provide good coverage** of your system’s attributes. Defend the SLOs you pick: if you can’t ever win a conversation about priorities by quoting a particular SLO, it’s probably not worth having that SLO.

SRE Book, Chapter 4 - Service Level Objectives

ALERTING STRATEGY

What to alert on:

"hard failure–centric and human-centric"

ALERTING STRATEGY

What to alert on:

"hard failure–centric and human-centric"

"symptom-based monitoring"

ALERTING STRATEGY

What to alert on:

"hard failure–centric and human-centric"

"symptom-based monitoring"

"alert on symptoms"

ALERTING STRATEGY

What to alert on:

"hard failure–centric and human-centric"

"symptom-based monitoring"

"alert on symptoms"

"symptoms that are associated with end-user pain"

ALERTING STRATEGY

Service Level Objectives:

"most critical aspects of the user experience"

ALERTING STRATEGY

Service Level Objectives:

"most critical aspects of the user experience"

"what your users care about"

ALERTING STRATEGY

"hard failure–centric and human-centric"

"symptom-based monitoring"

"alert on symptoms"

"symptoms that are associated with end-user pain"

=

"most critical aspects of the user experience"

"what your users care about"

zalando

ALERTING STRATEGY

What to alert on:

"Keep alerting simple"

ALERTING STRATEGY

What to alert on:

"Keep alerting simple"

"Aim to have as few alerts as possible"

ALERTING STRATEGY

Service Level Objectives:

"just enough [...] to provide good coverage"

ALERTING STRATEGY

"Keep alerting simple"

"Aim to have as few alerts as possible"

=

"just enough SLOs to provide good coverage"

ALERTING STRATEGY

Service Level Objective = Symptom + Threshold

ALERTING STRATEGY

Page only when your SLO is missed or in danger of being missed

ALERTING CHECKLIST

1. Does this rule detect **an otherwise undetected condition** that is urgent, actionable, and actively or imminently **user-visible**?

ALERTING CHECKLIST

1. Does this rule detect **an otherwise undetected condition** that is urgent, actionable, and actively or imminently **user-visible**?
2. **Will I ever be able to ignore this alert**, knowing it’s benign?

ALERTING CHECKLIST

1. Does this rule detect **an otherwise undetected condition** that is urgent, actionable, and actively or imminently **user-visible**?
2. **Will I ever be able to ignore this alert**, knowing it’s benign?
3. Does this alert **definitely indicate** that users are being **negatively affected**?

ALERTING CHECKLIST

1. Does this rule detect **an otherwise undetected condition** that is urgent, actionable, and actively or imminently **user-visible**?
2. **Will I ever be able to ignore this alert**, knowing it’s benign?
3. Does this alert **definitely indicate** that users are being **negatively affected**?
4. **Can I take action in response to this alert**?

ALERTING CHECKLIST

1. Does this rule detect **an otherwise undetected condition** that is urgent, actionable, and actively or imminently **user-visible**?
2. **Will I ever be able to ignore this alert**, knowing it’s benign?
3. Does this alert **definitely indicate** that users are being **negatively affected**?
4. **Can I take action in response to this alert**?
5. **Are other people getting paged for this issue**?

SRE Book, Chapter 6: Monitoring Distributed Systems

ALERTING EXAMPLES

"Load average is high"

ALERTING EXAMPLES

"Cassandra node is down"

ALERTING EXAMPLES

"EC2 instance is unhealthy"

CREDIT

The majority of these slides were inspired or contained references to the excellent work from many industry experts and publications:

People:

- Rob Ewaschuk
- Björn Rabenstein
- Cindy Sridharan
- Charity Majors
- And many more...

Publications:

- Site Reliability Engineering (Book)
- The Site Reliability Workbook (Book)
- Distributed Systems Observability (e-Book)

ХВАЛА

QUESTIONS?

Don't miss my next talk tomorrow at 11:30
"Are we all on the same page? Let's fix that"

Luis Mineiro @voidmaze

We're Hiring!
https://jobs.zalando.com