A Field Guide to
Reliability Engineering at Zalando

goto; Amsterdam 2024 - Heinrich Hartmann

👋 I'm Heinrich - Reliability Engineer

Experience

zalando	Senior Principal SRE (2021)
CIRCONUS	Chief Data Scientist (2015)
universitätbonn	PhD in Mathematics (2011)

Talking Reliability since 2015

- SRECon - Statistics for Engineers
- DevOps Berlin - Zalando's quest to Operate 10K…
- SLOConf - The State of the Histogram
- P99 Conf - How to measure Latency
- FOSDEM - Latency SLOs Done Right
- Circllhist - A Histogram Data Structure… (arxiv)

Find me on

heinrichhartmann.com
LinkedIn, X

Menu

1. Principles
2. Context
3. Operations at Zalando
    a. Alerting
    b. Dashboards
    c. Observability
    d. Incident Process
    e. SLOs
    f. WORMs

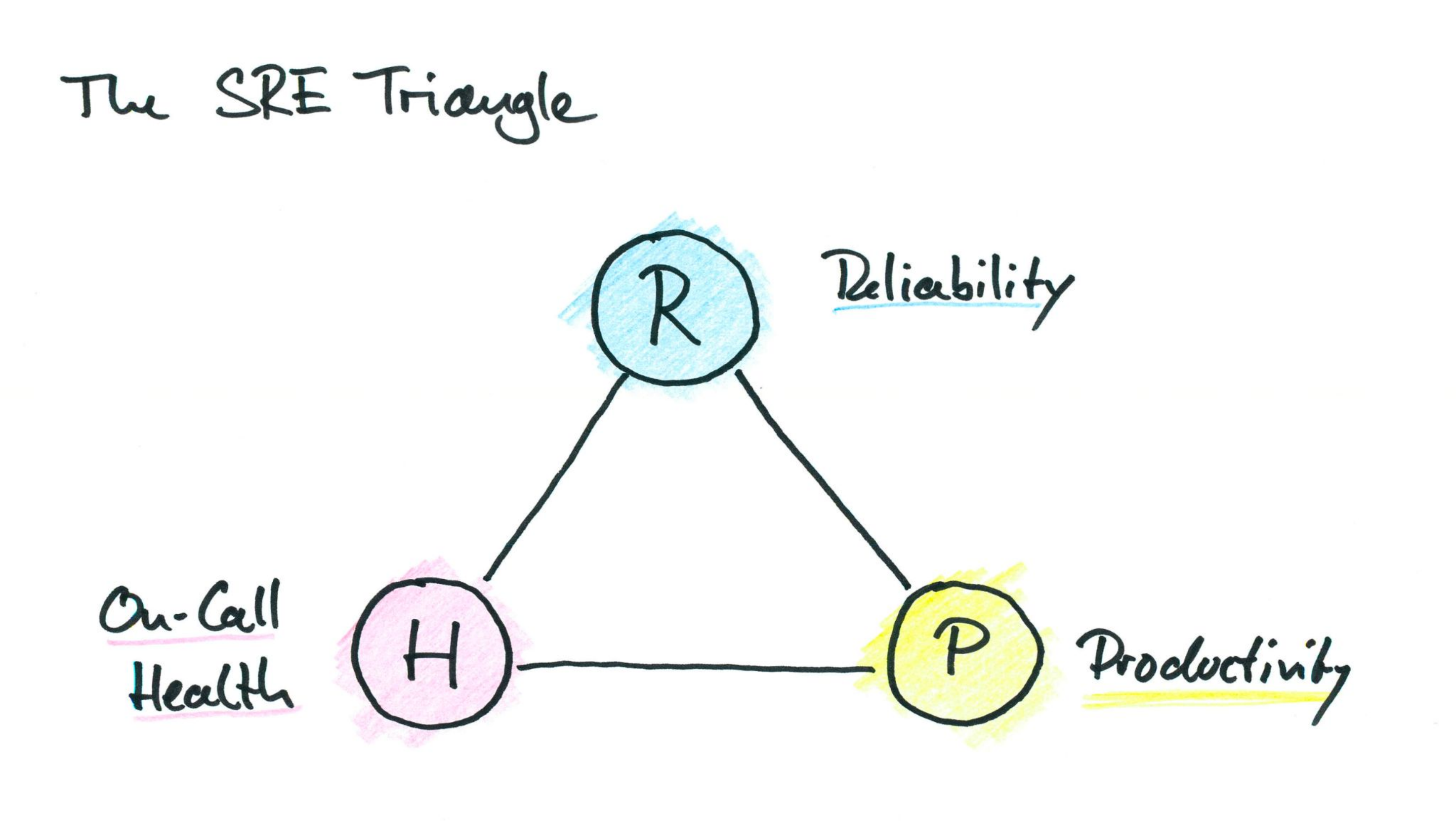
Principles

Mission

Protect the User Experience from operational failures while keeping an eye on (1) Developer Productivity and (2) On-Call Health.

#1 Rule of Operations

Obsess about User Experience.


The SRE Triangle
R
Reliability
H
On-Call
Health
P
Productivity

#2 Rule of Operations

Engineering for Reliability involves people as much as it involves technology.

Engineering Reliability at Scale

Small Company (~10 FTE)

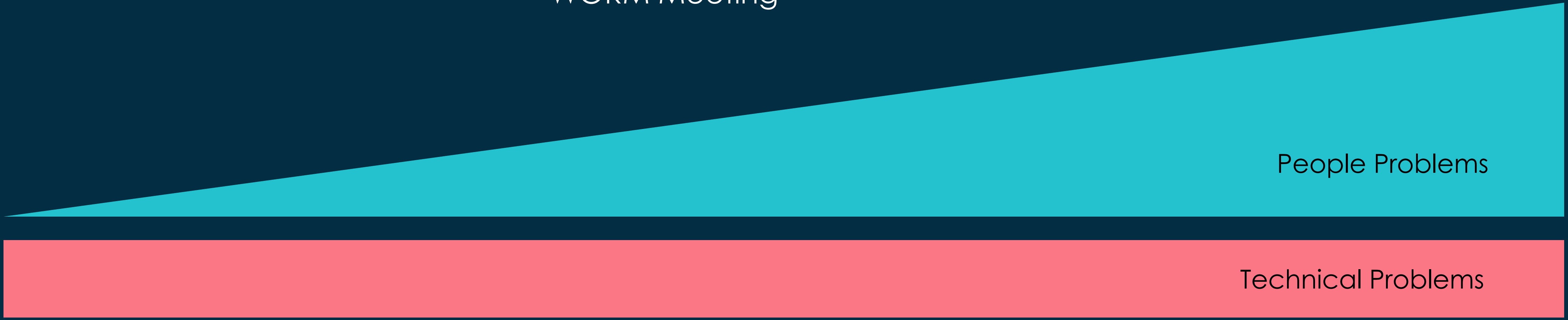
- Alerts & Dashboards
- Logging

Medium Company (~100 FTE)

- Incident Management
- Observability
- On-call rotations
- Playbooks
- WORM Meeting

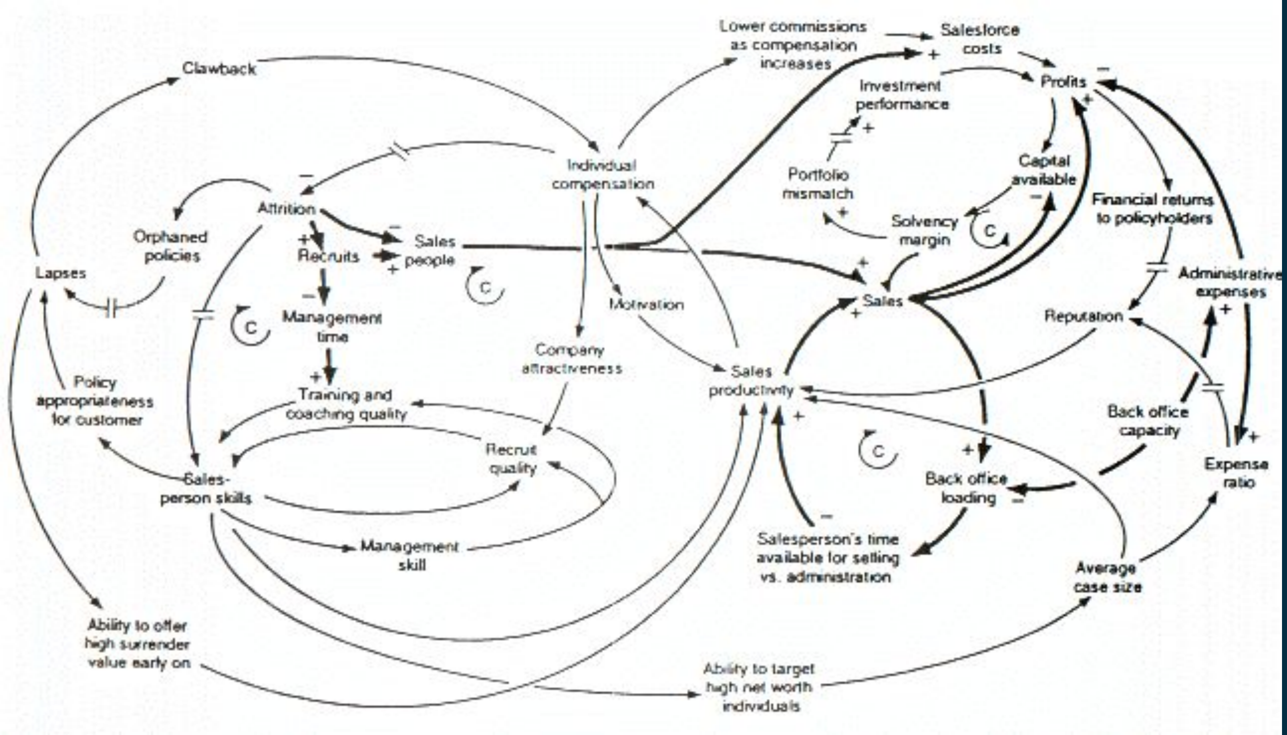
Large Company (>1k FTE)

- WORM Cascades
- Risk Management
- SRE Community & Guilds

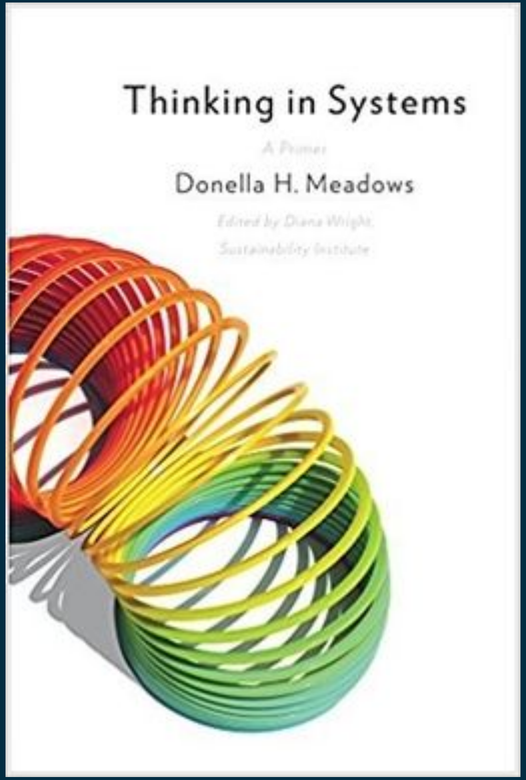


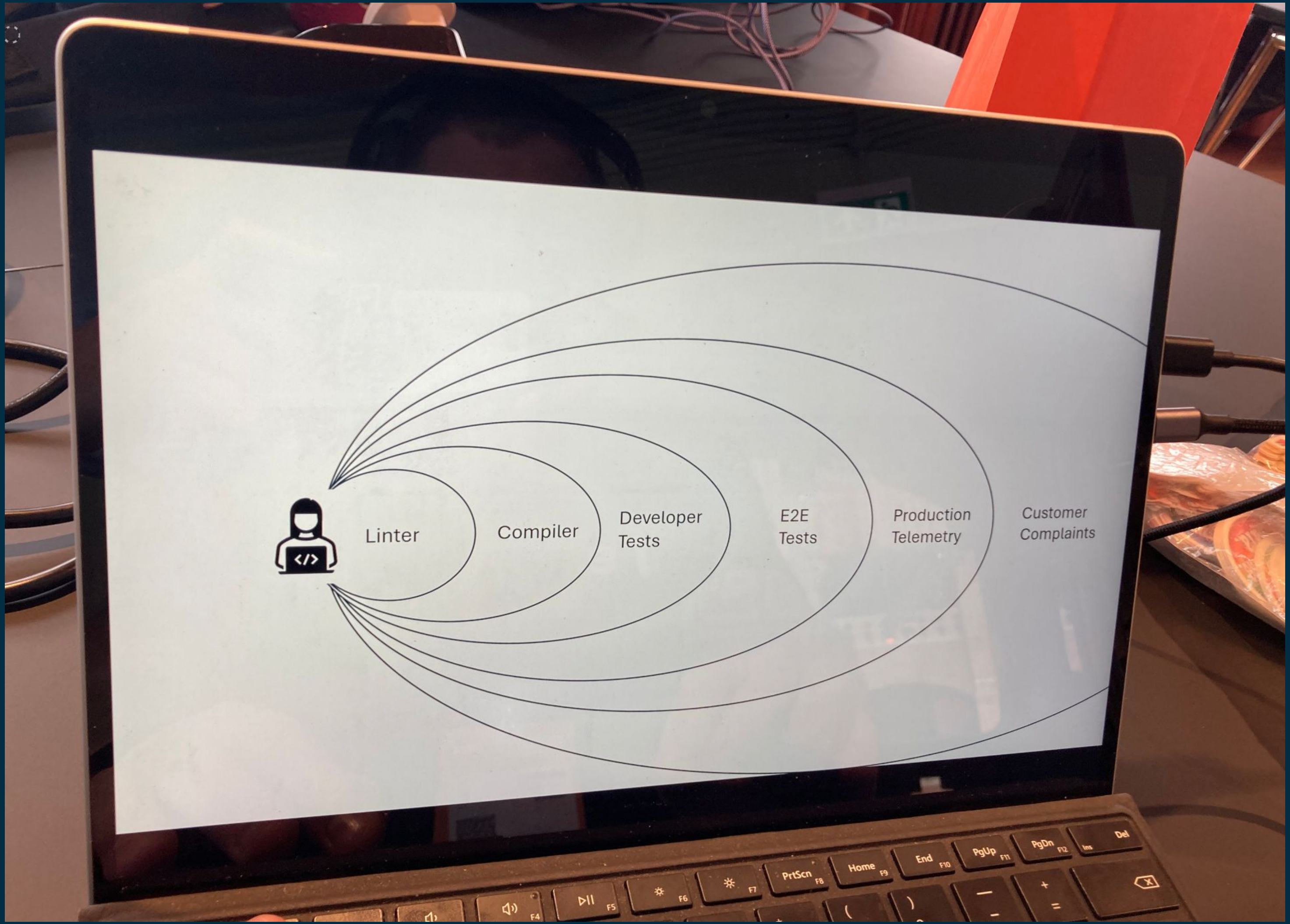
Engineering Socio-Technological Systems

with "Systems Theory"



Example: Causal Loop Diagram - source: wikipedia

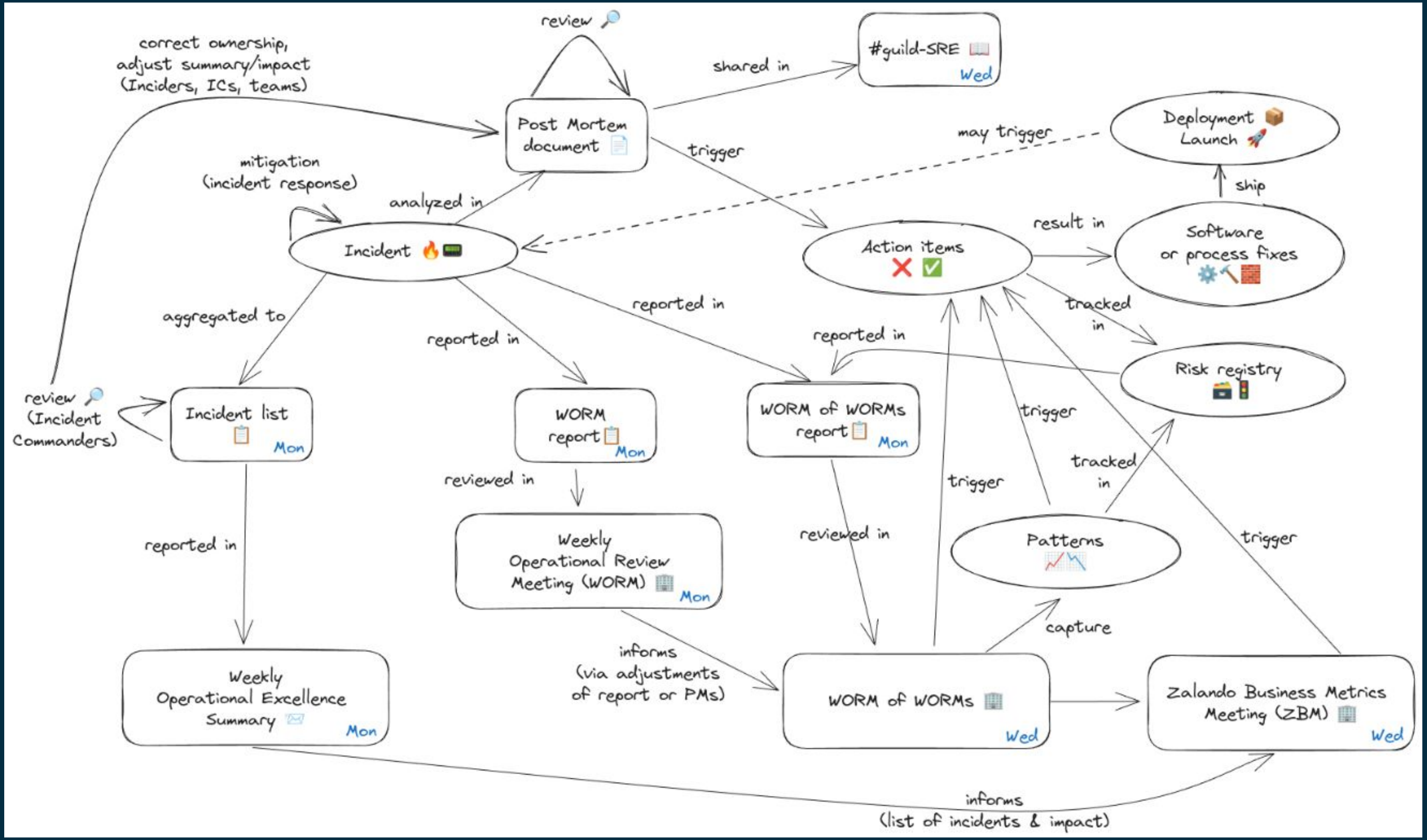




% Martin Thwaites @ Honeycomb GOTO 2024

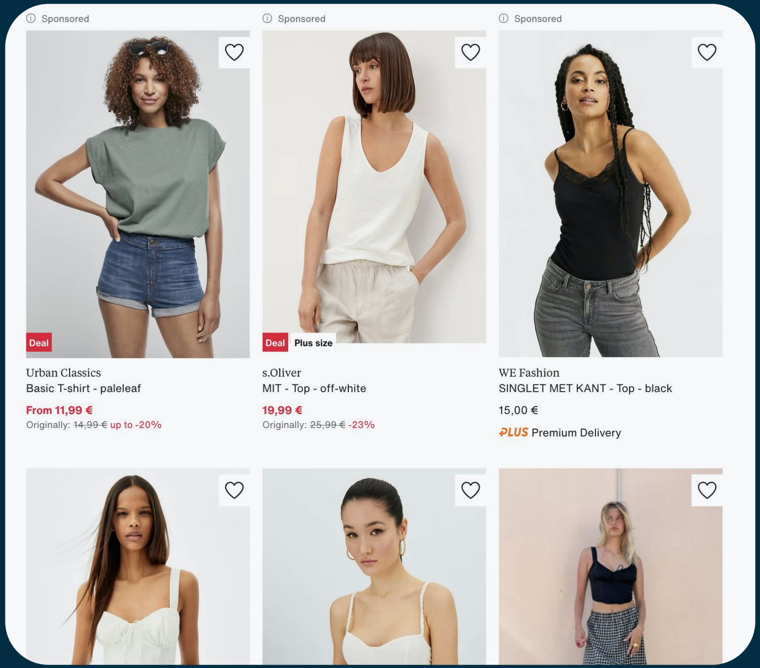
goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Reliability "Flywheel" at Zalando



Context

zalando



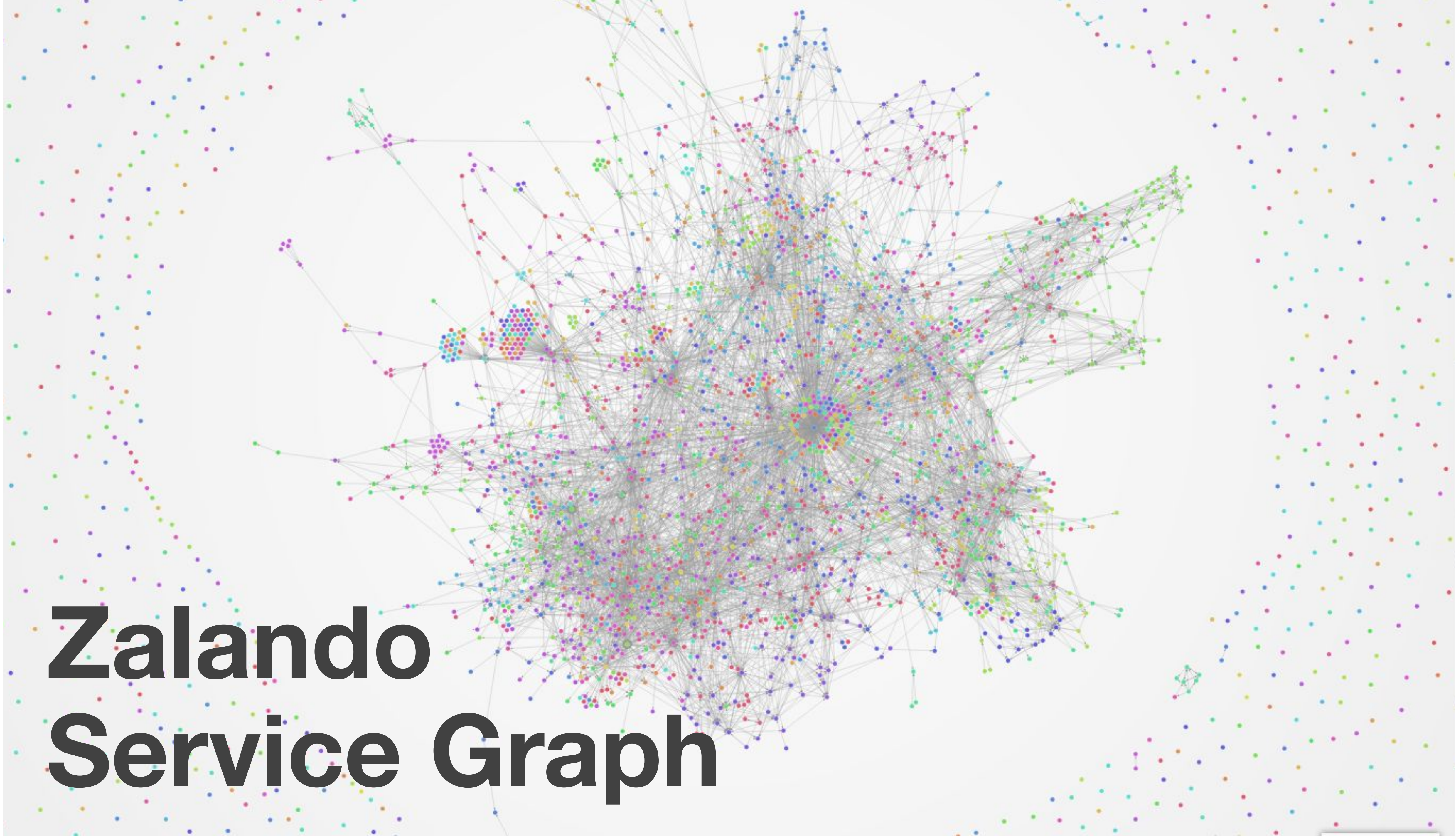
Urban Classics
Basic T-shirt - paleleaf
From 11,99 €
Originally: ~~14,99 €~~ up to -20%

s.Oliver
MIT - Top - off-white
19,99 €
Originally: ~~25,99 €~~ -23%

WE Fashion
SINGLET MET KANT - Top - black
15,00 €
PLUS Premium Delivery

- One of the leading fashion platforms in EU
- Founded in 2008
- 14.6 bn EUR Revenue / 50M+ active Customers
- 25 Countries
- 3K Tech Employees
- 3K+ Micro Services

Zalando Service Graph



Don't separate People and Technology

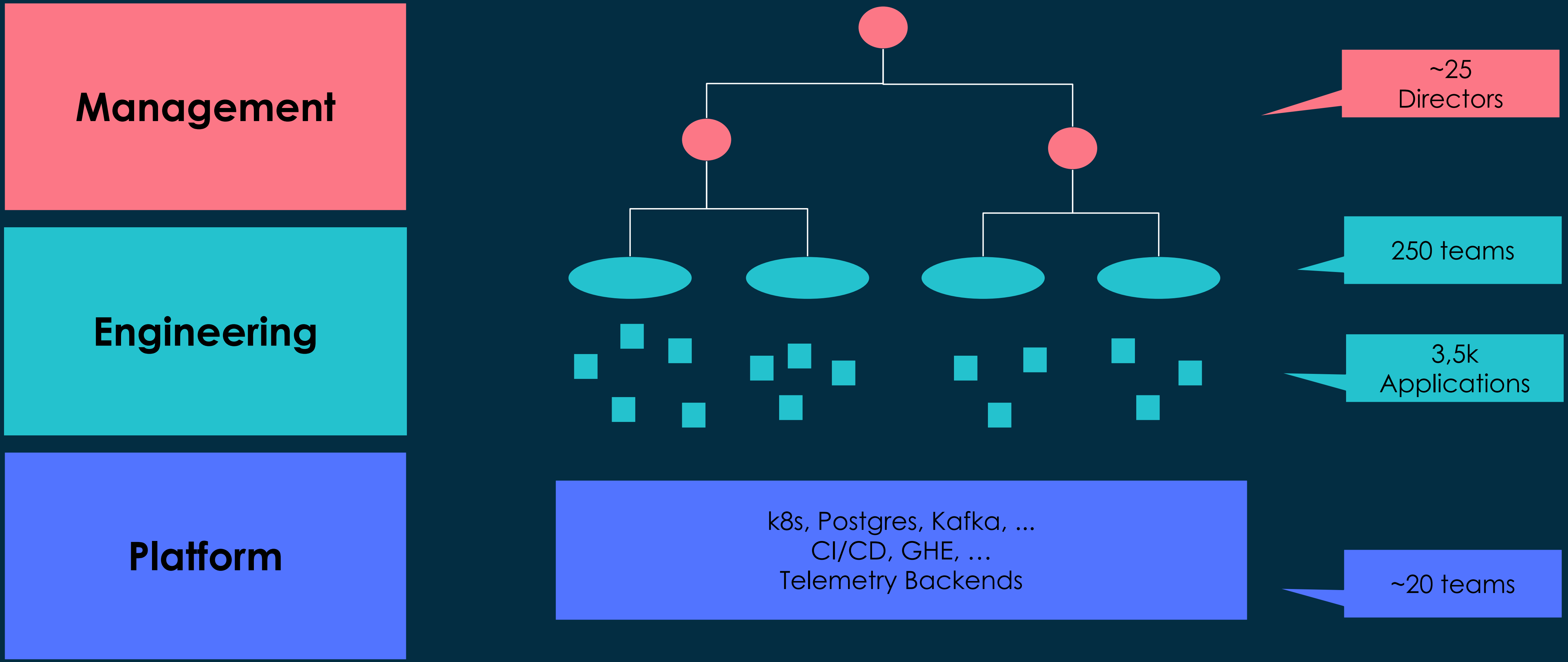
Conway's law

Technology Structures mirror People Structures.

Law of DevOps

You build it, you run it!

Systems Model of Zalando



goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Where do we stand?

+ Operating "transactional" Microservices
+ Protecting the Business
+ Preparing for High-Load Events

- Understanding User Experience
- Reliability of Data Systems / Business Processes

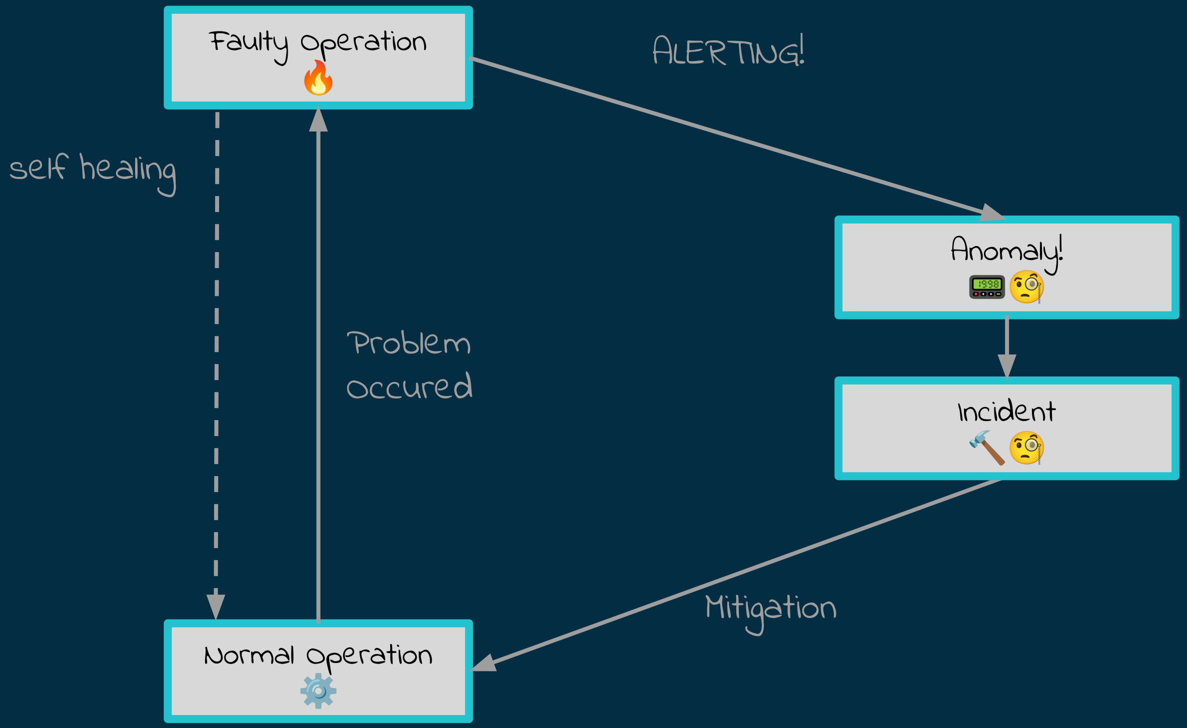
Operations at Zalando

Alerting

Why Alerting?

Reduce Time to Detect user-facing issues.

Alerting as Feedback Loop



#3 Rule of Operations

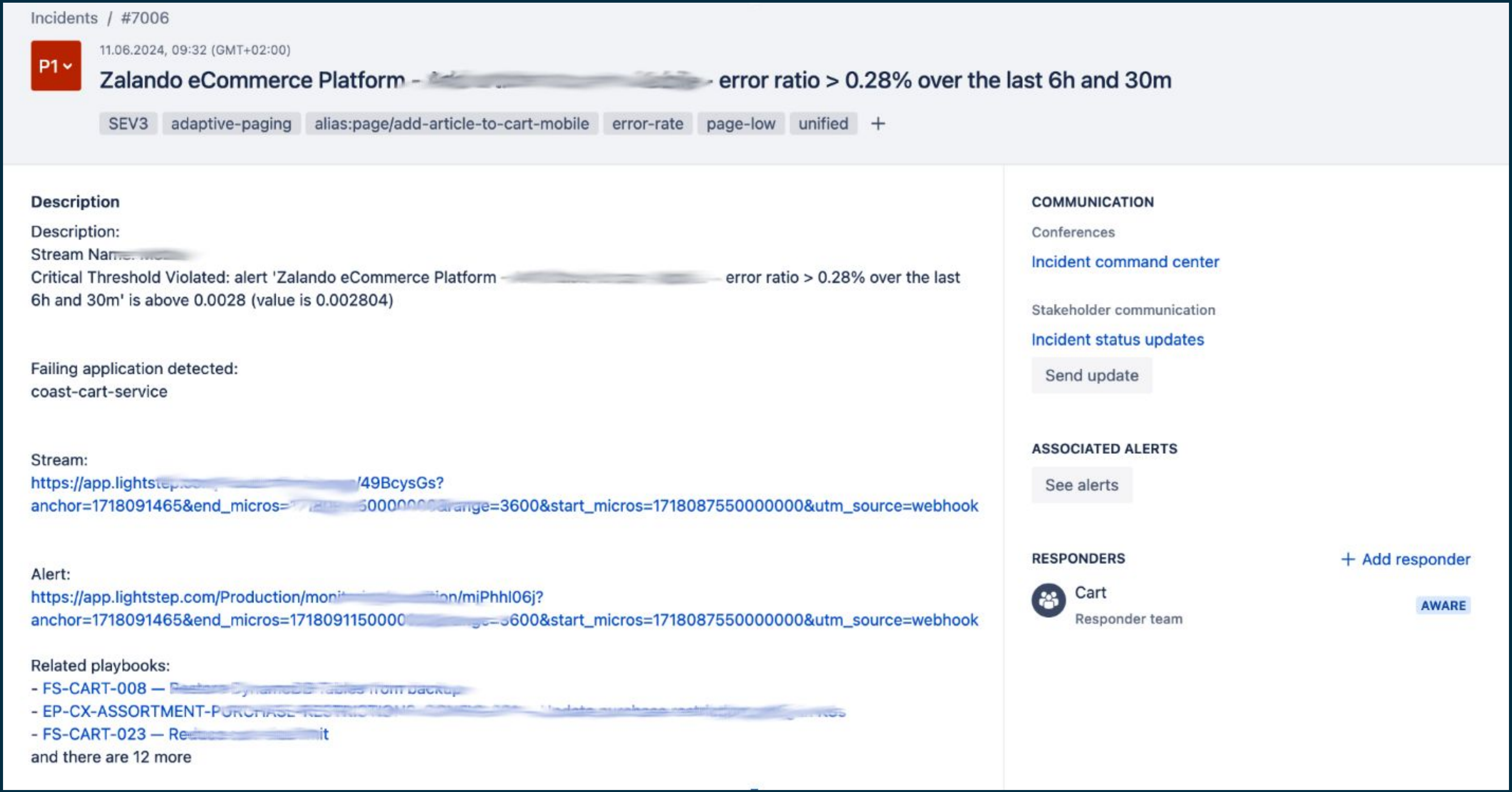
Alert on User Experience ("Symptoms") not on Server Experience ("Causes").

- Alert on error rates of user-facing "operations"

- Leverage SLO-based Alerting (if available)

- Don't alert on CPU Utilization

This is fine.



Incidents / #7006

P1

11.06.2024, 09:32 (GMT+02:00)

Zalando eCommerce Platform - [redacted] error ratio > 0.28% over the last 6h and 30m

SEV3 adaptive-paging alias:page/add-article-to-cart-mobile error-rate page-low unified +

Description

Description:
Stream Na[redacted]
Critical Threshold Violated: alert 'Zalando eCommerce Platform [redacted] error ratio > 0.28% over the last 6h and 30m' is above 0.0028 (value is 0.002804)

Failing application detected:
coast-cart-service

Stream:
https://app.lightst[redacted]/49BcysGs?anchor=1718091465&end_micros=[redacted]range=3600&start_micros=1718087550000000&utm_source=webhook

Alert:
https://app.lightstep.com/Production/moni[redacted]ion/miPhhl06j?anchor=1718091465&end_micros=1718091150000[redacted]600&start_micros=1718087550000000&utm_source=webhook

Related playbooks:
- FS-CART-008 — [redacted]
- EP-CX-ASSORTMENT-P[redacted]
- FS-CART-023 — Re[redacted]it

and there are 12 more

COMMUNICATION

Conferences

Incident command center

Stakeholder communication

Incident status updates

Send update

ASSOCIATED ALERTS

See alerts

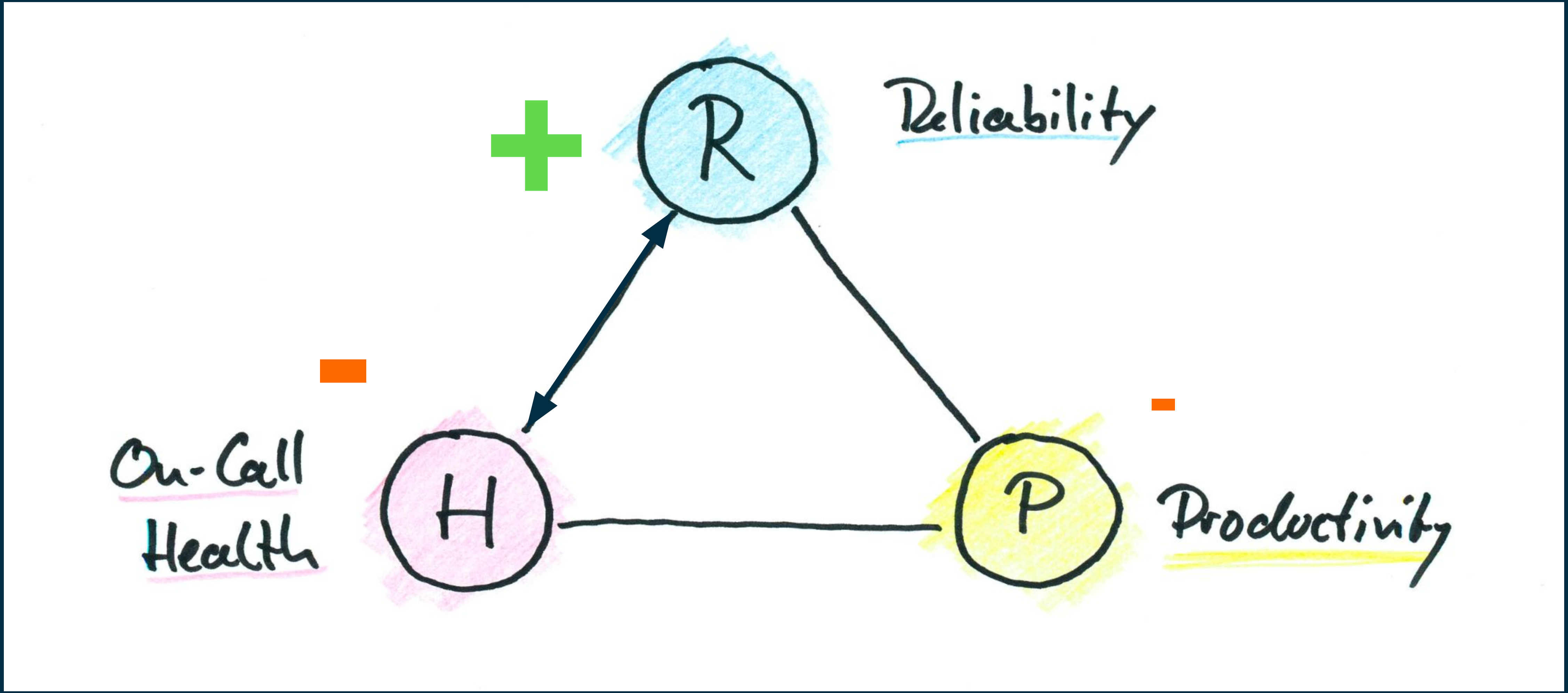
RESPONDERS

+ Add responder

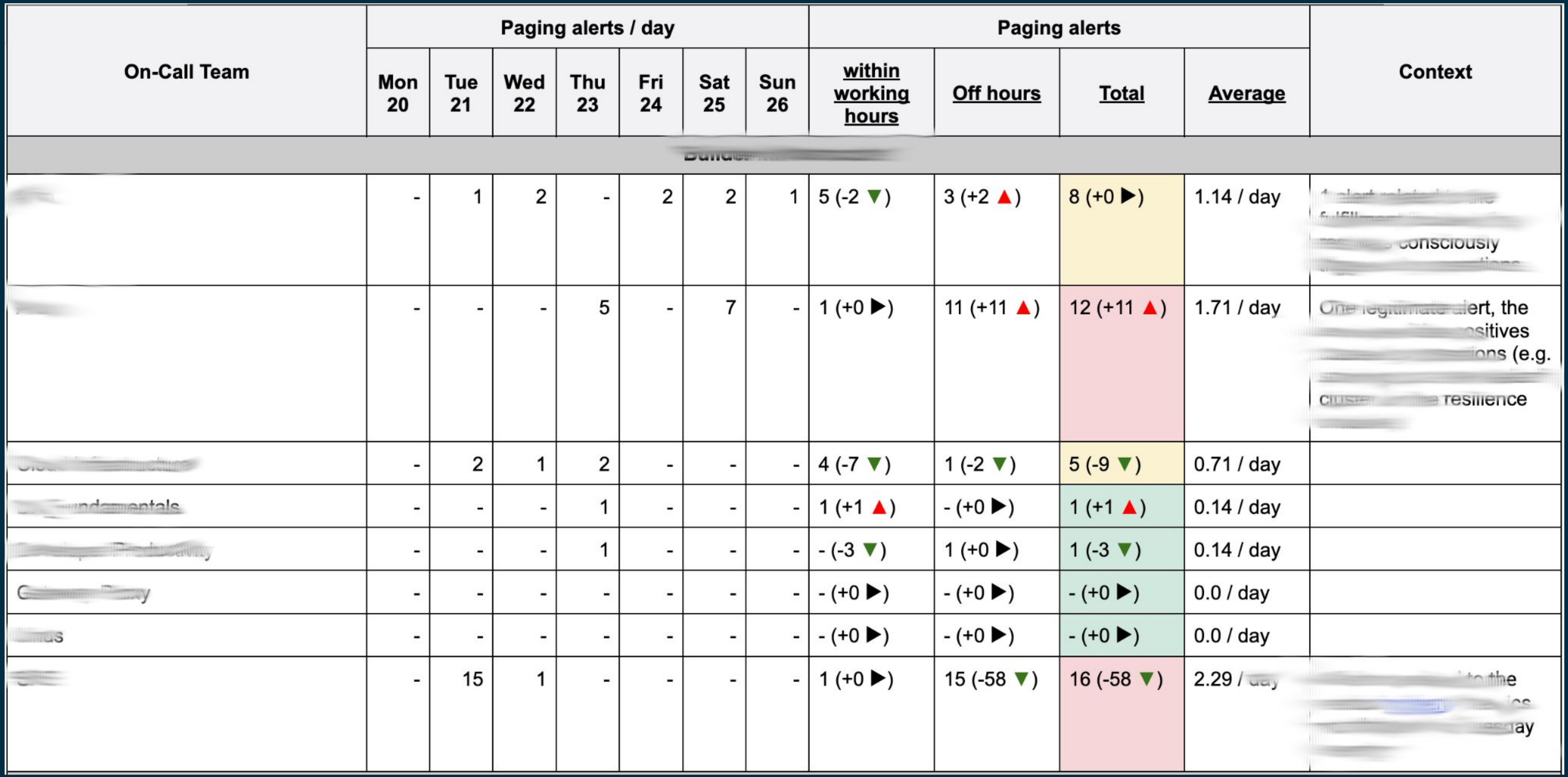
Cart
Responder team

AWARE

Adding alerts trades Reliability of On-Call Health



Review On-Call Health Weekly!



On-Call Team	Paging alerts / day							Paging alerts				Context
	Mon 20	Tue 21	Wed 22	Thu 23	Fri 24	Sat 25	Sun 26	within working hours	Off hours	Total	Average	
[redacted]												
[redacted]	-	1	2	-	2	2	1	5 (-2 ▼)	3 (+2 ▲)	8 (+0 ▶)	1.14 / day	[redacted]
[redacted]	-	-	-	5	-	7	-	1 (+0 ▶)	11 (+11 ▲)	12 (+11 ▲)	1.71 / day	[redacted]
[redacted]	-	2	1	2	-	-	-	4 (-7 ▼)	1 (-2 ▼)	5 (-9 ▼)	0.71 / day	
[redacted]	-	-	-	1	-	-	-	1 (+1 ▲)	- (+0 ▶)	1 (+1 ▲)	0.14 / day	
[redacted]	-	-	-	1	-	-	-	- (-3 ▼)	1 (+0 ▶)	1 (-3 ▼)	0.14 / day	
[redacted]	-	-	-	-	-	-	-	- (+0 ▶)	- (+0 ▶)	- (+0 ▶)	0.0 / day	
[redacted]	-	-	-	-	-	-	-	- (+0 ▶)	- (+0 ▶)	- (+0 ▶)	0.0 / day	
[redacted]	-	15	1	-	-	-	-	1 (+0 ▶)	15 (-58 ▼)	16 (-58 ▼)	2.29 / [redacted]	[redacted]

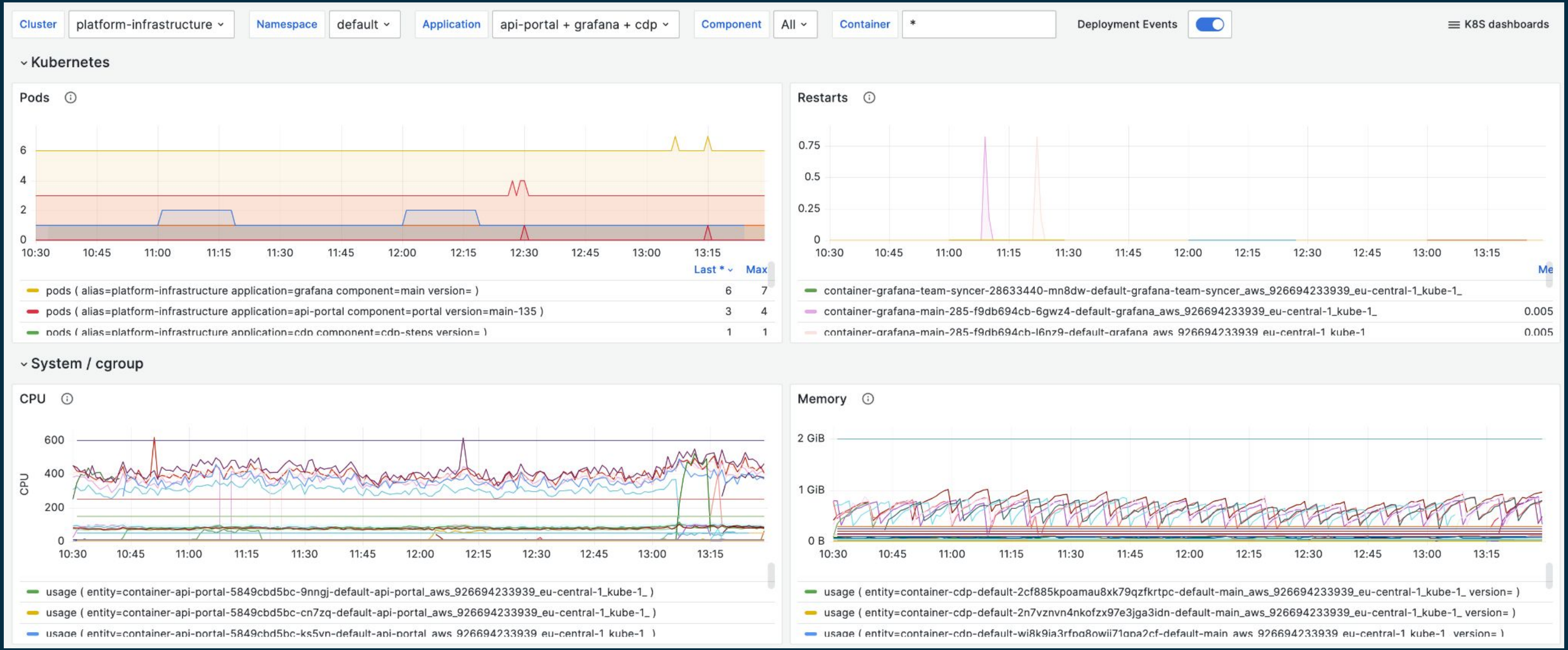
Dashboards

Why Dashboards?

- Reduce Time to Repair
- Look at them when you get alerted. Don't monitor dashboards.
- Starting point for understanding Service Health

- Every Application MUST have an Application Dashboard.
- Managed Services come with Managed Dashboards.

Managed Kubernetes Dashboard



Cluster: platform-infrastructure ⌄ · Namespace: default ⌄ · Application: api-portal + grafana + cdp ⌄ · Component: All ⌄ · Container: * · Deployment Events ● · ☰ K8S dashboards

⌄ Kubernetes

Pods ⓘ

	Last * ⌄	Max
🟨 pods (alias=platform-infrastructure application=grafana component=main version=)	6	7
🟥 pods (alias=platform-infrastructure application=api-portal component=portal version=main-135)	3	4
🟩 pods (alias=platform-infrastructure application=cdp component=cdp-steps version=)	1	1

Restarts ⓘ

	Me
🟩 container-grafana-team-syncer-28633440-mn8dw-default-grafana-team-syncer_aws_926694233939_eu-central-1_kube-1_	
🟪 container-grafana-main-285-f9db694cb-6gwz4-default-grafana_aws_926694233939_eu-central-1_kube-1_	0.005
container-grafana-main-285-f9db694cb-l6nz9-default-grafana_aws_926694233939_eu-central-1_kube-1	0.005

⌄ System / cgroup

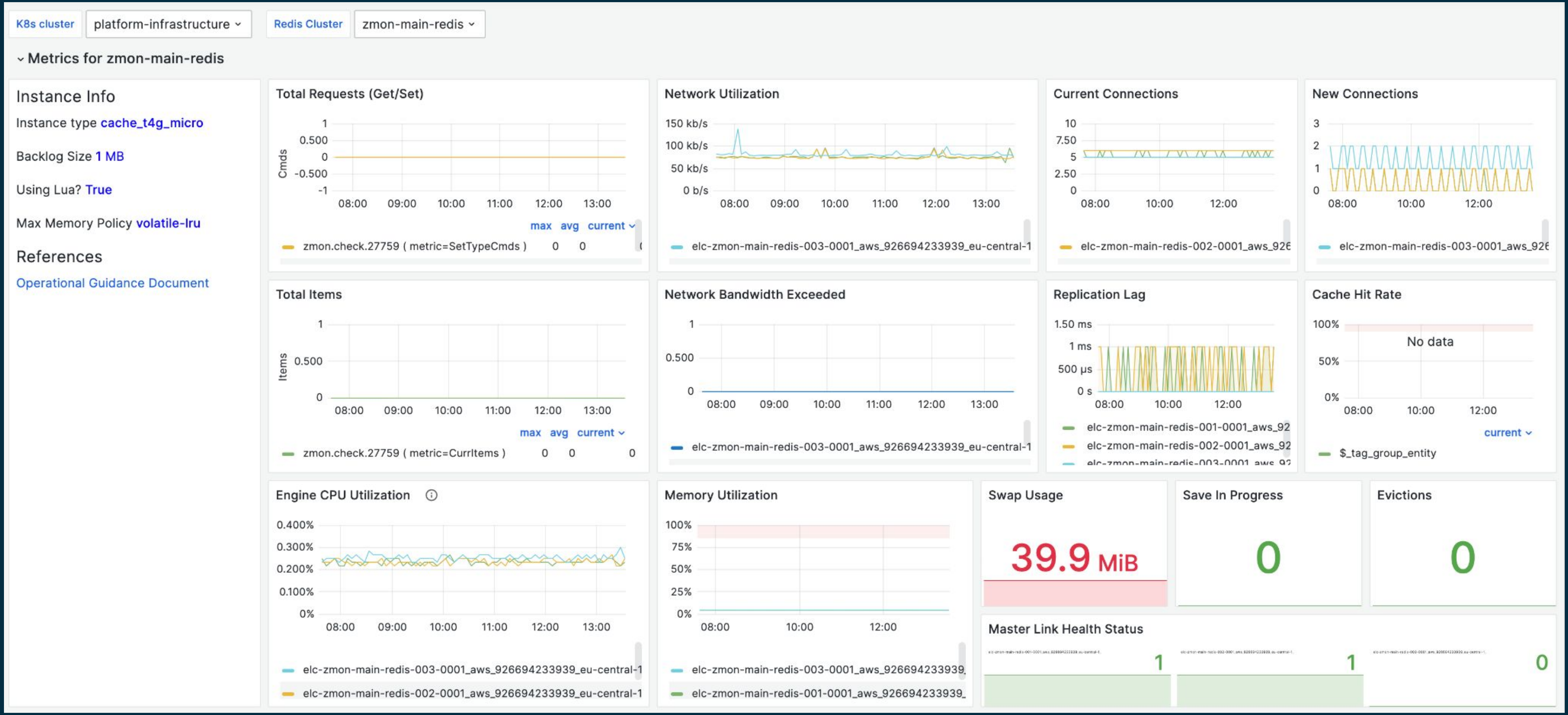
CPU ⓘ

- 🟩 usage (entity=container-api-portal-5849cbd5bc-9nngj-default-api-portal_aws_926694233939_eu-central-1_kube-1_)
- 🟨 usage (entity=container-api-portal-5849cbd5bc-cn7zq-default-api-portal_aws_926694233939_eu-central-1_kube-1_)
- 🟦 usage (entity=container-api-portal-5849cbd5bc-ks5vn-default-api-portal_aws_926694233939_eu-central-1_kube-1_)

Memory ⓘ

- 🟩 usage (entity=container-cdp-default-2cf885kpoamau8xk79qzfkrtpc-default-main_aws_926694233939_eu-central-1_kube-1_ version=)
- 🟨 usage (entity=container-cdp-default-2n7vznvn4nkofzx97e3jga3idn-default-main_aws_926694233939_eu-central-1_kube-1_ version=)
- 🟦 usage (entity=container-cdp-default-wi8k9ia3rfpg8owij71gpa2cf-default-main_aws_926694233939_eu-central-1_kube-1_ version=)

Managed REDIS Dashboard



K8s cluster: platform-infrastructure

Redis Cluster: zmon-main-redis

Metrics for zmon-main-redis

Instance Info

Instance type cache_t4g_micro

Backlog Size 1 MB

Using Lua? True

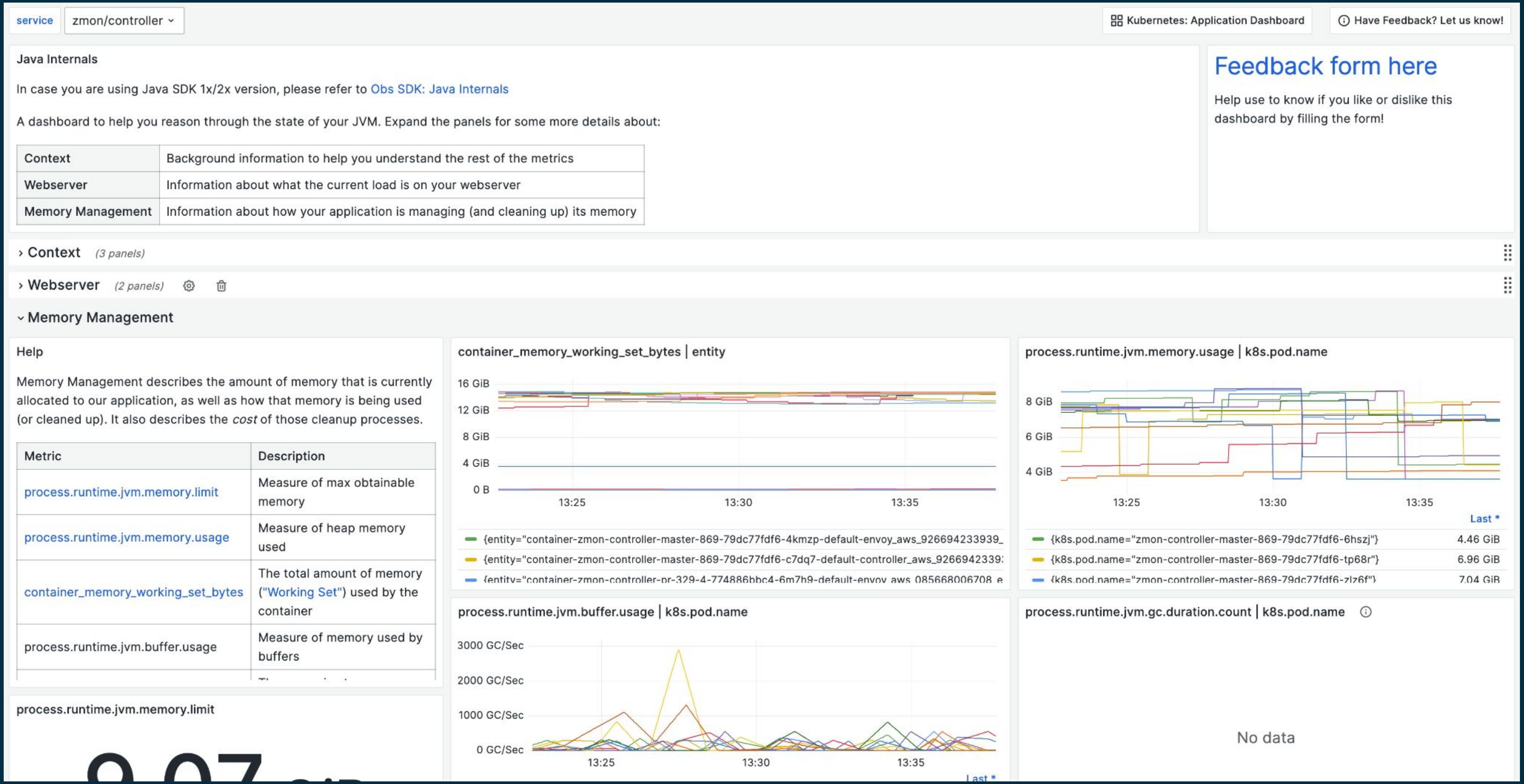
Max Memory Policy volatile-lru

References

Operational Guidance Document

Total Requests (Get/Set)
Cmds
1
0.500
0
-0.500
-1
08:00
09:00
10:00
11:00
12:00
13:00
max avg current
zmon.check.27759 (metric=SetTypeCmds) 0 0
Network Utilization
150 kb/s
100 kb/s
50 kb/s
0 b/s
elc-zmon-main-redis-003-0001_aws_926694233939_eu-central-1
Current Connections
10
7.50
5
2.50
0
elc-zmon-main-redis-002-0001_aws_926
New Connections
3
2
1
0
elc-zmon-main-redis-003-0001_aws_926
Total Items
Items
zmon.check.27759 (metric=CurrItems) 0 0 0
Network Bandwidth Exceeded
elc-zmon-main-redis-003-0001_aws_926694233939_eu-central-1
Replication Lag
1.50 ms
1 ms
500 µs
0 s
elc-zmon-main-redis-001-0001_aws_92
elc-zmon-main-redis-002-0001_aws_92
elc-zmon-main-redis-003-0001_aws_92
Cache Hit Rate
100%
50%
0%
No data
current
$_tag_group_entity
Engine CPU Utilization
0.400%
0.300%
0.200%
0.100%
0%
elc-zmon-main-redis-003-0001_aws_926694233939_eu-central-1
elc-zmon-main-redis-002-0001_aws_926694233939_eu-central-1
Memory Utilization
100%
75%
50%
25%
0%
elc-zmon-main-redis-003-0001_aws_926694233939_
elc-zmon-main-redis-001-0001_aws_926694233939_
Swap Usage
39.9 MiB
Save In Progress
0
Evictions
0
Master Link Health Status
1
1
0

Managed JVM Internals Dashboard



service
zmon/controller ˅
Kubernetes: Application Dashboard
Have Feedback? Let us know!

Java Internals

In case you are using Java SDK 1x/2x version, please refer to Obs SDK: Java Internals

A dashboard to help you reason through the state of your JVM. Expand the panels for some more details about:

Context	Background information to help you understand the rest of the metrics
Webserver	Information about what the current load is on your webserver
Memory Management	Information about how your application is managing (and cleaning up) its memory

Feedback form here

Help use to know if you like or dislike this dashboard by filling the form!

› Context *(3 panels)*

› Webserver *(2 panels)*

˅ Memory Management

Help

Memory Management describes the amount of memory that is currently allocated to our application, as well as how that memory is being used (or cleaned up). It also describes the *cost* of those cleanup processes.

Metric	Description
process.runtime.jvm.memory.limit	Measure of max obtainable memory
process.runtime.jvm.memory.usage	Measure of heap memory used
container_memory_working_set_bytes	The total amount of memory ("Working Set") used by the container
process.runtime.jvm.buffer.usage	Measure of memory used by buffers

process.runtime.jvm.memory.limit

container_memory_working_set_bytes | entity

- {entity="container-zmon-controller-master-869-79dc77fdf6-4kmzp-default-envoy_aws_926694233939_
- {entity="container-zmon-controller-master-869-79dc77fdf6-c7dq7-default-controller_aws_92669423393
- {entity="container-zmon-controller-pr-329-4-774886bbc4-6m7h9-default-envoy_aws_085668006708_e

process.runtime.jvm.memory.usage | k8s.pod.name

	Last *
{k8s.pod.name="zmon-controller-master-869-79dc77fdf6-6hszj"}	4.46 GiB
{k8s.pod.name="zmon-controller-master-869-79dc77fdf6-tp68r"}	6.96 GiB
{k8s.pod.name="zmon-controller-master-869-79dc77fdf6-zlz6f"}	7.04 GiB

process.runtime.jvm.buffer.usage | k8s.pod.name

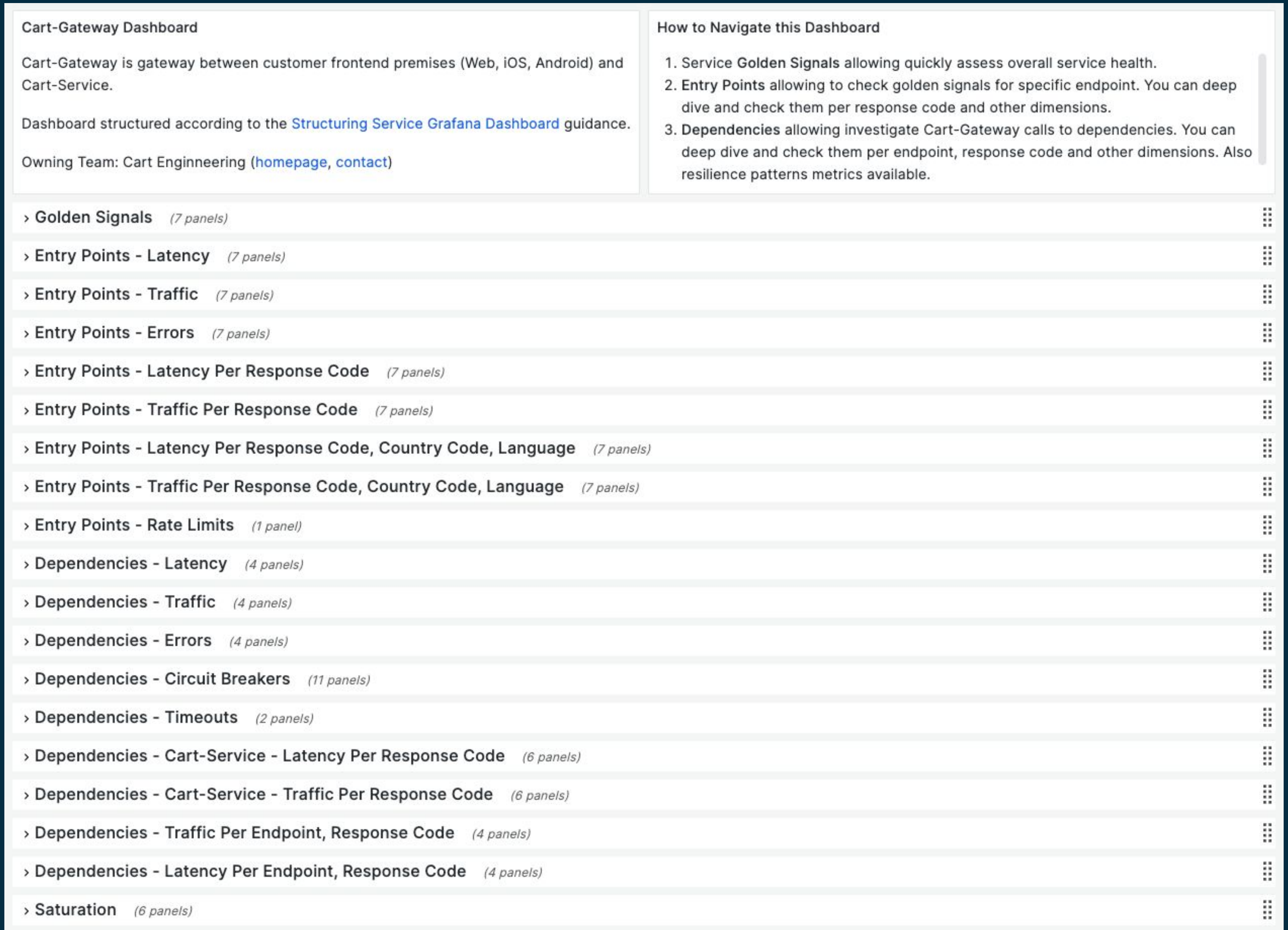
process.runtime.jvm.gc.duration.count | k8s.pod.name

No data

goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Zalando Application Dashboard Guidelines

1. Golden Signals
2. Entry Points
3. Dependencies
4. Saturation
5. Operational Insights
6. Storage



Cart-Gateway Dashboard

Cart-Gateway is gateway between customer frontend premises (Web, iOS, Android) and Cart-Service.

Dashboard structured according to the Structuring Service Grafana Dashboard guidance.

Owning Team: Cart Enginneering (homepage, contact)

How to Navigate this Dashboard

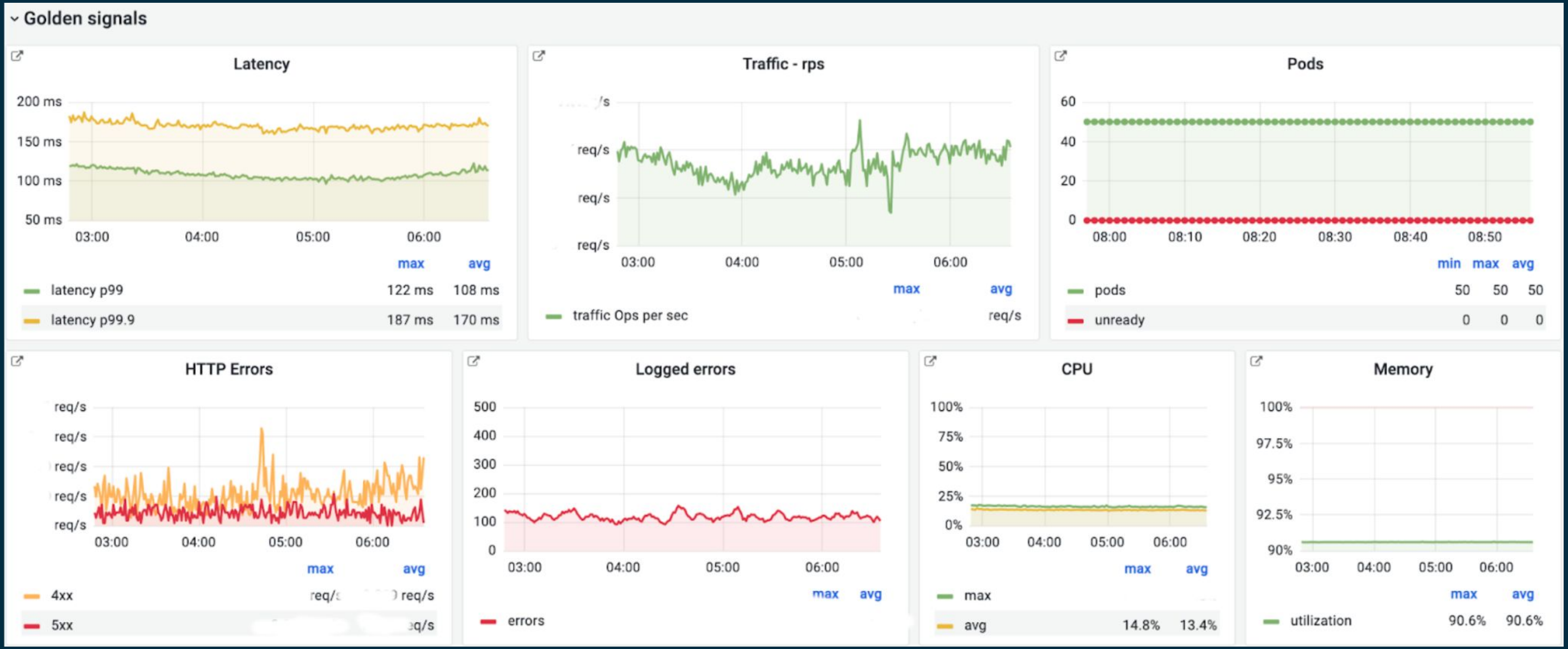
1. Service **Golden Signals** allowing quickly assess overall service health.
2. **Entry Points** allowing to check golden signals for specific endpoint. You can deep dive and check them per response code and other dimensions.
3. **Dependencies** allowing investigate Cart-Gateway calls to dependencies. You can deep dive and check them per endpoint, response code and other dimensions. Also resilience patterns metrics available.

- › Golden Signals *(7 panels)*
- › Entry Points - Latency *(7 panels)*
- › Entry Points - Traffic *(7 panels)*
- › Entry Points - Errors *(7 panels)*
- › Entry Points - Latency Per Response Code *(7 panels)*
- › Entry Points - Traffic Per Response Code *(7 panels)*
- › Entry Points - Latency Per Response Code, Country Code, Language *(7 panels)*
- › Entry Points - Traffic Per Response Code, Country Code, Language *(7 panels)*
- › Entry Points - Rate Limits *(1 panel)*
- › Dependencies - Latency *(4 panels)*
- › Dependencies - Traffic *(4 panels)*
- › Dependencies - Errors *(4 panels)*
- › Dependencies - Circuit Breakers *(11 panels)*
- › Dependencies - Timeouts *(2 panels)*
- › Dependencies - Cart-Service - Latency Per Response Code *(6 panels)*
- › Dependencies - Cart-Service - Traffic Per Response Code *(6 panels)*
- › Dependencies - Traffic Per Endpoint, Response Code *(4 panels)*
- › Dependencies - Latency Per Endpoint, Response Code *(4 panels)*
- › Saturation *(6 panels)*

courtesy of Evgeni Sokolov & Miha Lunar

Golden Signals Row - RED(S)

Duration **Requests** **Saturation**



	max	avg
🟩 latency p99	122 ms	108 ms
🟨 latency p99.9	187 ms	170 ms

	max	avg
🟩 traffic Ops per sec		req/s

	min	max	avg
🟩 pods	50	50	50
🟥 unready	0	0	0

	max	avg
🟨 4xx	req/s	req/s
🟥 5xx		req/s

	max	avg
🟥 errors		

	max	avg
🟩 max		
🟨 avg	14.8%	13.4%

	max	avg
🟩 utilization	90.6%	90.6%

Errors

goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

w/ Evgeni Sokolov & Miha Lunar

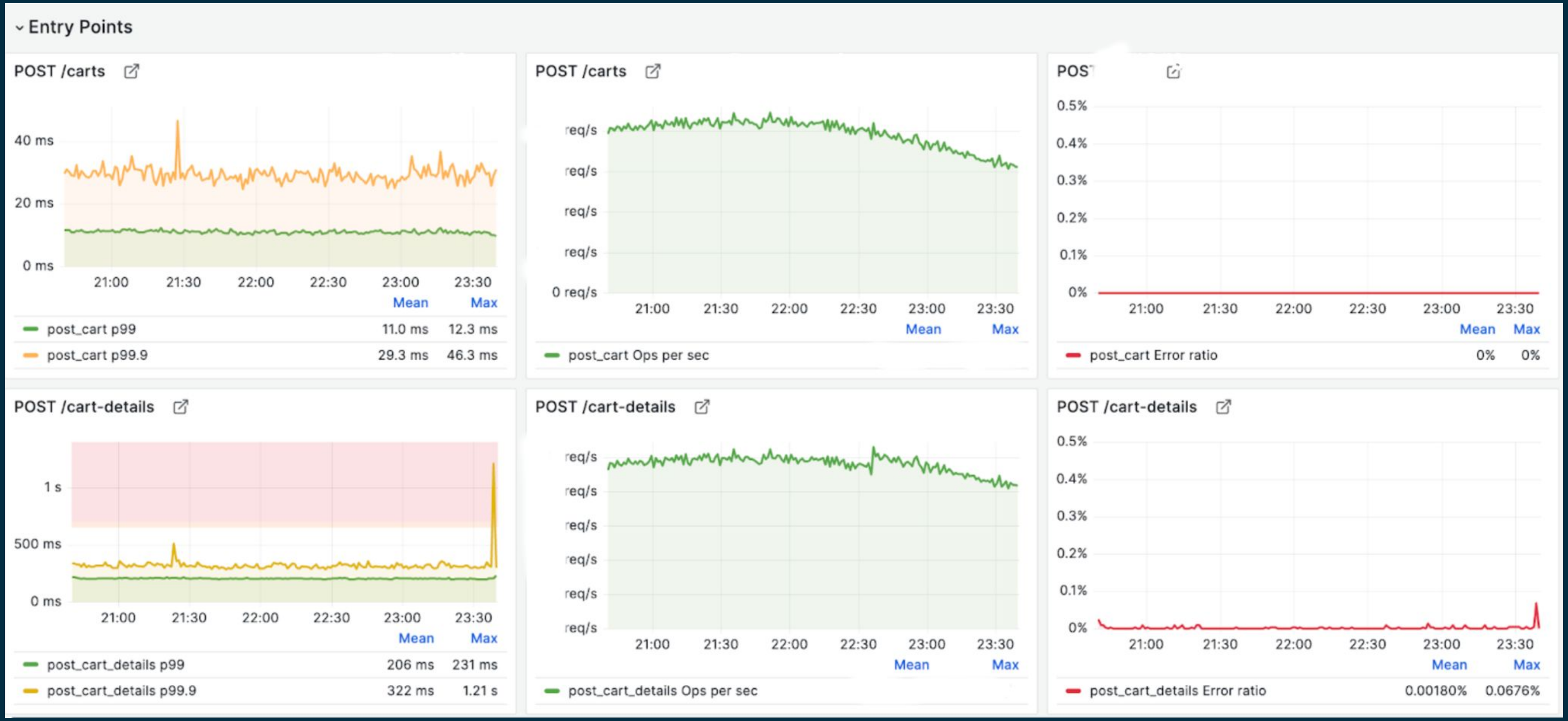
Entry Points Row

Golden Signals, again! - RED

Duration **Requests** **Errors**

POST /carts

POST /card-details



POST /carts

	Mean	Max
post_cart p99	11.0 ms	12.3 ms
post_cart p99.9	29.3 ms	46.3 ms

POST /carts

	Mean	Max
post_cart Ops per sec		

POS

	Mean	Max
post_cart Error ratio	0%	0%

POST /cart-details

	Mean	Max
post_cart_details p99	206 ms	231 ms
post_cart_details p99.9	322 ms	1.21 s

POST /cart-details

	Mean	Max
post_cart_details Ops per sec		

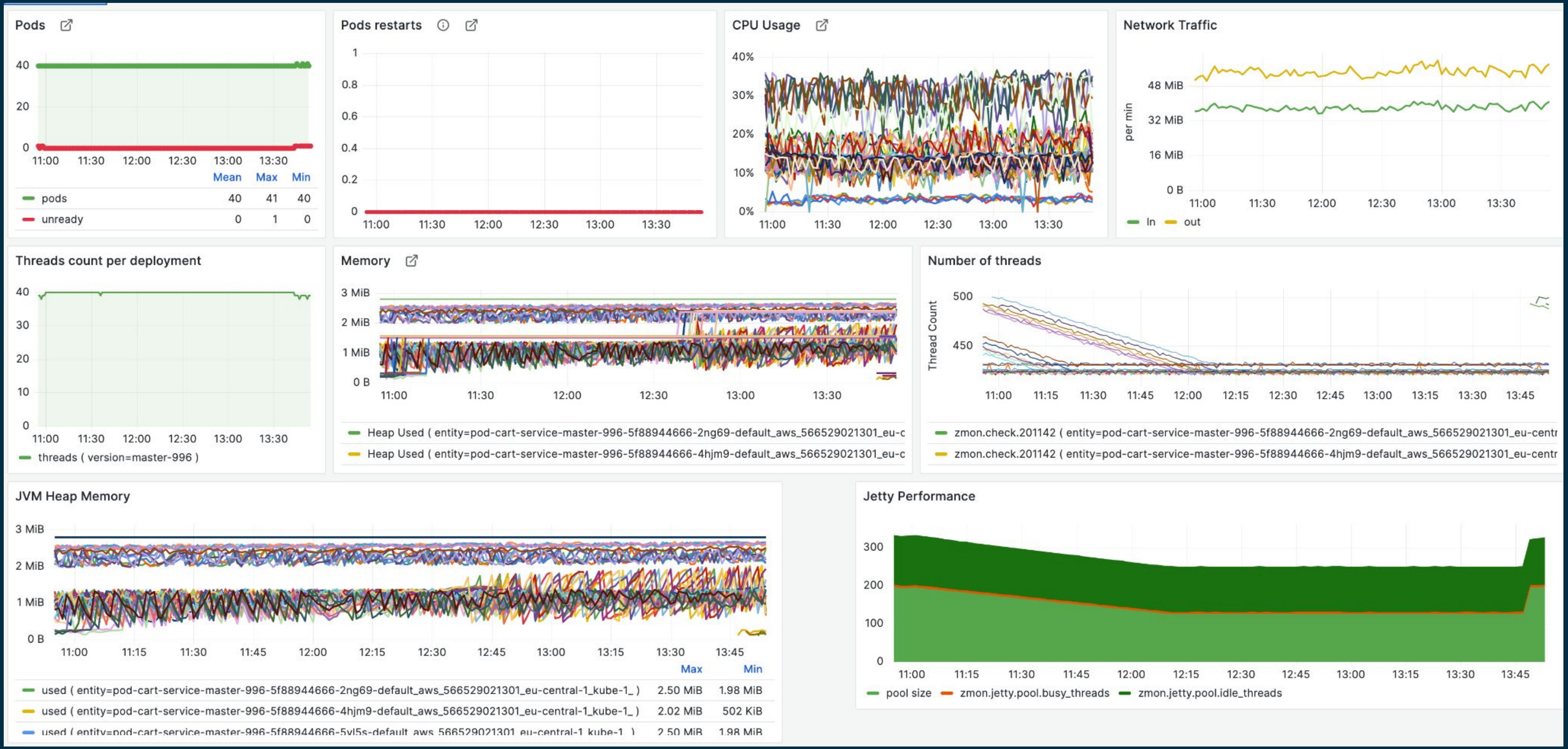
POST /cart-details

	Mean	Max
post_cart_details Error ratio	0.00180%	0.0676%

w/ Evgeni Sokolov & Miha Lunar

Saturation Row

… everything that can get saturated.



	Mean	Max	Min
pods	40	41	40
unready	0	1	0

threads (version=master-996)

Heap Used (entity=pod-cart-service-master-996-5f88944666-2ng69-default_aws_566529021301_eu-c

Heap Used (entity=pod-cart-service-master-996-5f88944666-4hjm9-default_aws_566529021301_eu-c

zmon.check.201142 (entity=pod-cart-service-master-996-5f88944666-2ng69-default_aws_566529021301_eu-centr

zmon.check.201142 (entity=pod-cart-service-master-996-5f88944666-4hjm9-default_aws_566529021301_eu-centr

	Max	Min
used (entity=pod-cart-service-master-996-5f88944666-2ng69-default_aws_566529021301_eu-central-1_kube-1_)	2.50 MiB	1.98 MiB
used (entity=pod-cart-service-master-996-5f88944666-4hjm9-default_aws_566529021301_eu-central-1_kube-1_)	2.02 MiB	502 KiB
used (entity=pod-cart-service-master-996-5f88944666-5vl5s-default_aws_566529021301_eu-central-1_kube-1_)	2.50 MiB	1.98 MiB

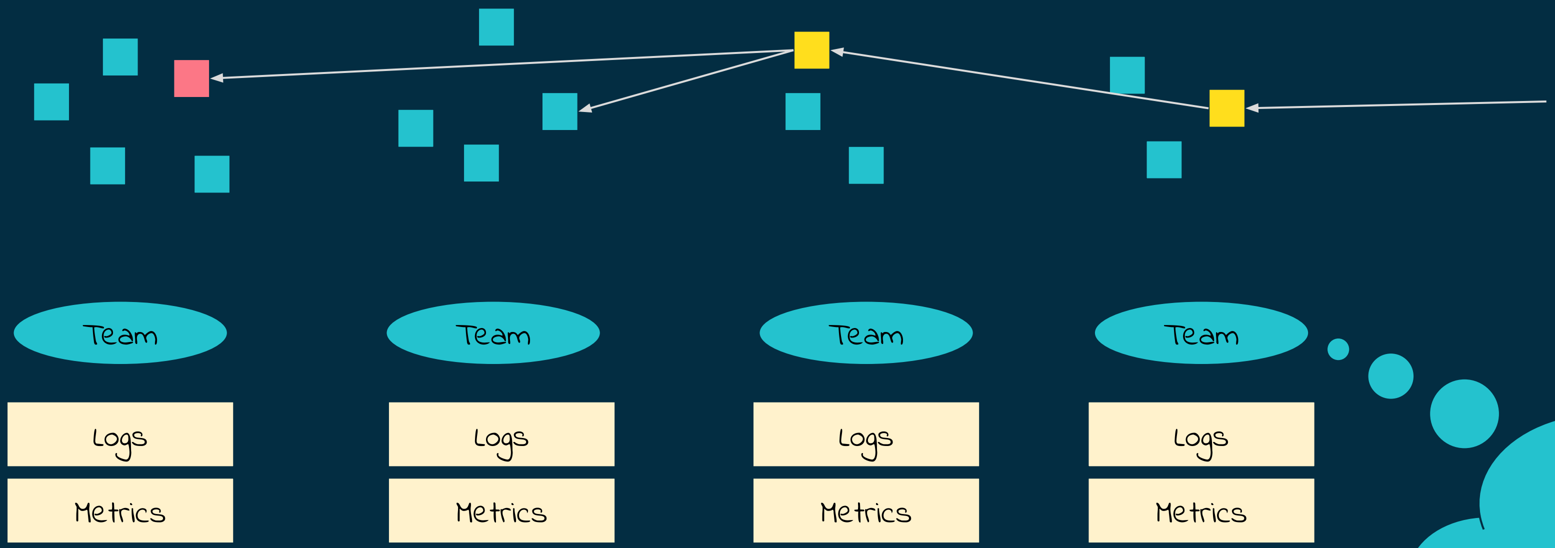
w/ Evgeni Sokolov & Miha Lunar

Observability

Why Observability?

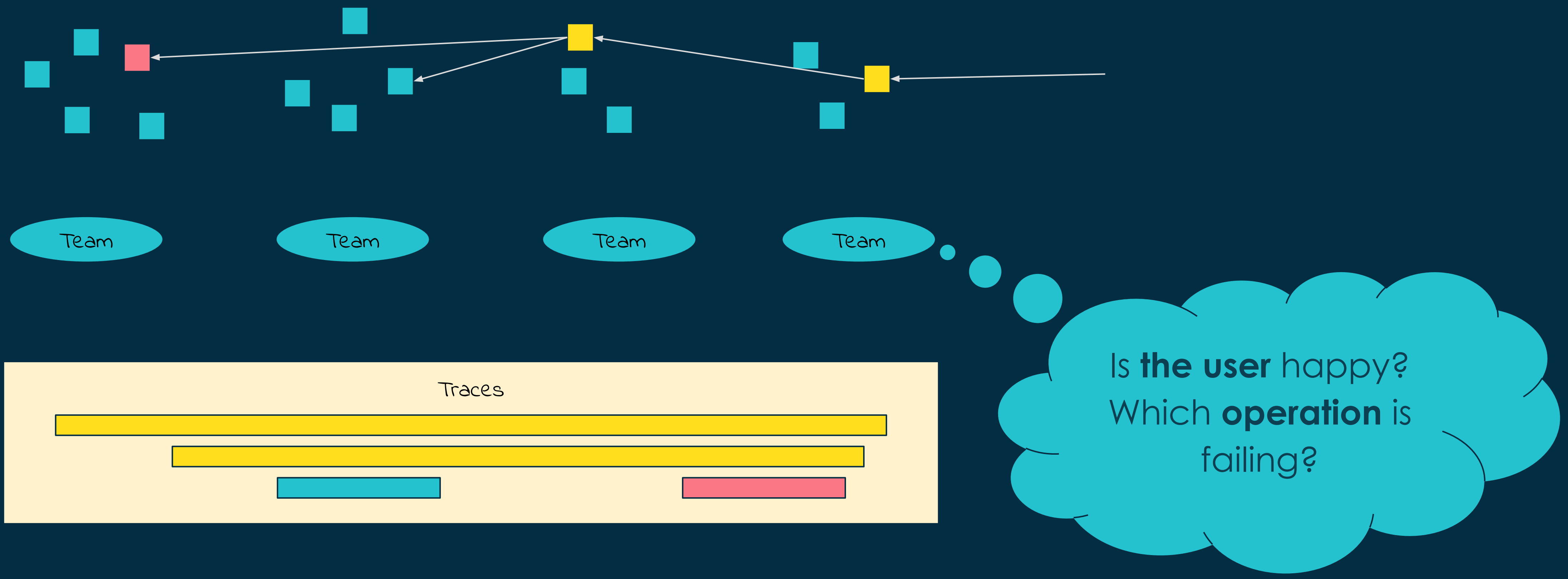
- Reduce Time to Repair
- Debug failures across team boundaries
- Understand User-Experience
- Basis for Alerting, Dashboards, Reporting, …

Traditional Monitoring



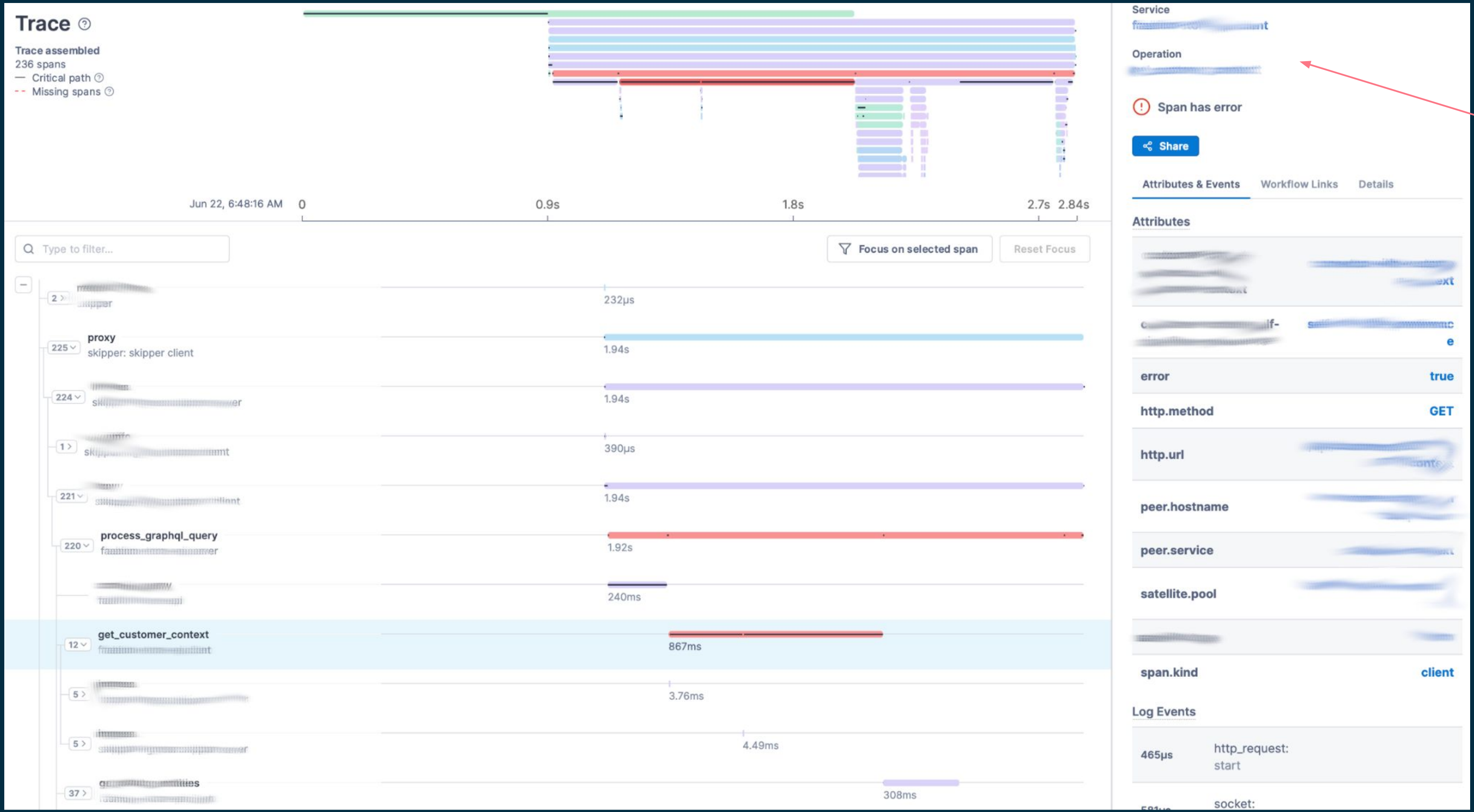
Is **my application** healthy? **Which errors** does it throw?

Observability



Is **the user** happy?
Which **operation** is failing?

Example Trace from Zalando Front Page



Trace ⓘ

Trace assembled
236 spans
— Critical path ⓘ
-- Missing spans ⓘ

Type to filter...

Focus on selected span

Reset Focus

```
2     [redacted]                         232µs
225   proxy
      skipper: skipper client            1.94s
224   [redacted]                         1.94s
1     [redacted]                         390µs
221   [redacted]                         1.94s
220   process_graphql_query
      [redacted]                         1.92s
      [redacted]                         240ms
12    get_customer_context
      [redacted]                         867ms
5     [redacted]                         3.76ms
5     [redacted]                         4.49ms
37    [redacted]                         308ms
```

Service
[redacted]

Operation
[redacted]

ⓘ Span has error

Share

Attributes & Events | Workflow Links | Details

Attributes

[redacted]	[redacted]
[redacted]	[redacted]
error	true
http.method	GET
http.url	[redacted]
peer.hostname	[redacted]
peer.service	[redacted]
satellite.pool	[redacted]
[redacted]	[redacted]
span.kind	client

Log Events

465µs	http_request: start
581µs	socket:

Team "CIA"

Application "CuCo"

ServiceNow
Cloud Observability

goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Zalando Developer Observability Guidelines

1. Use **OpenTelemetry** to instrument Applications.

2. Use **Distributed Tracing** to understand system behavior in the context of transactions (e.g. HTTP requests).

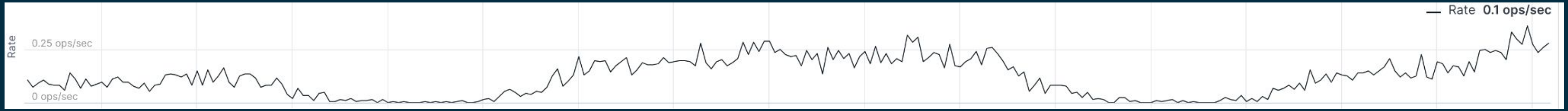
3. **Metrics** for precise counts & global resource statistics

4. Structured **Logging** for Lifecycle events

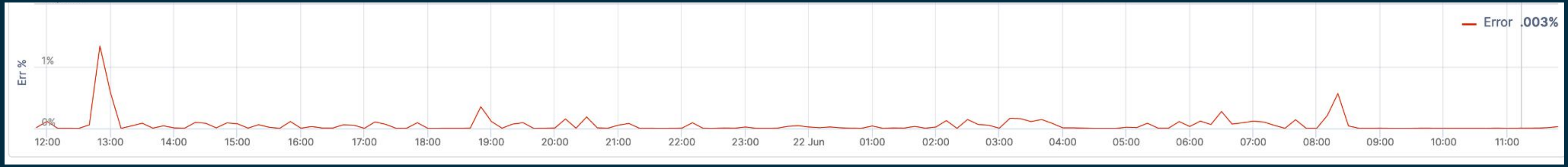
Monitor Reliability of Operations with "RED" Metrics

Operation: Reset Password

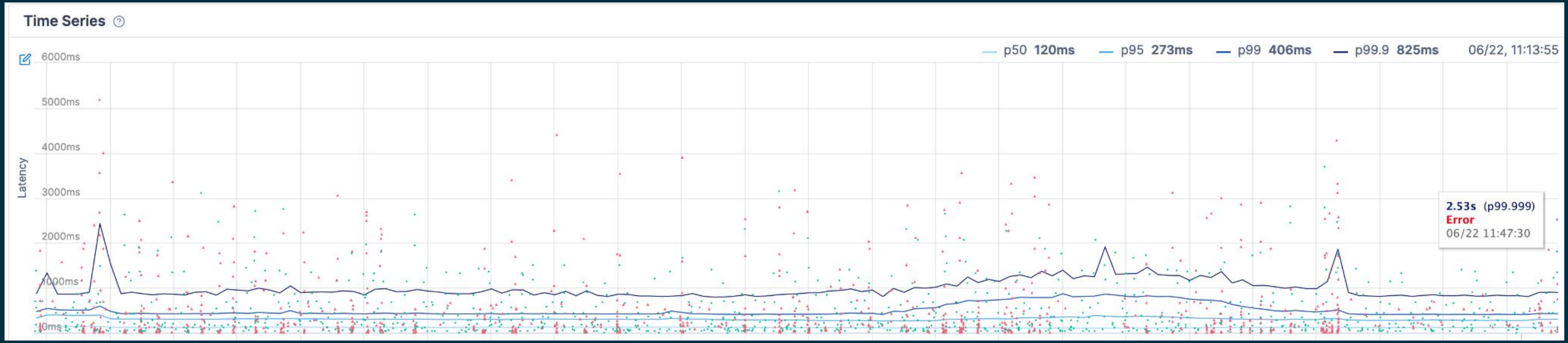
Requests



Errors



Duration



ServiceNow
Cloud Observability

goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Observability SDKs

based on Open Telemetry

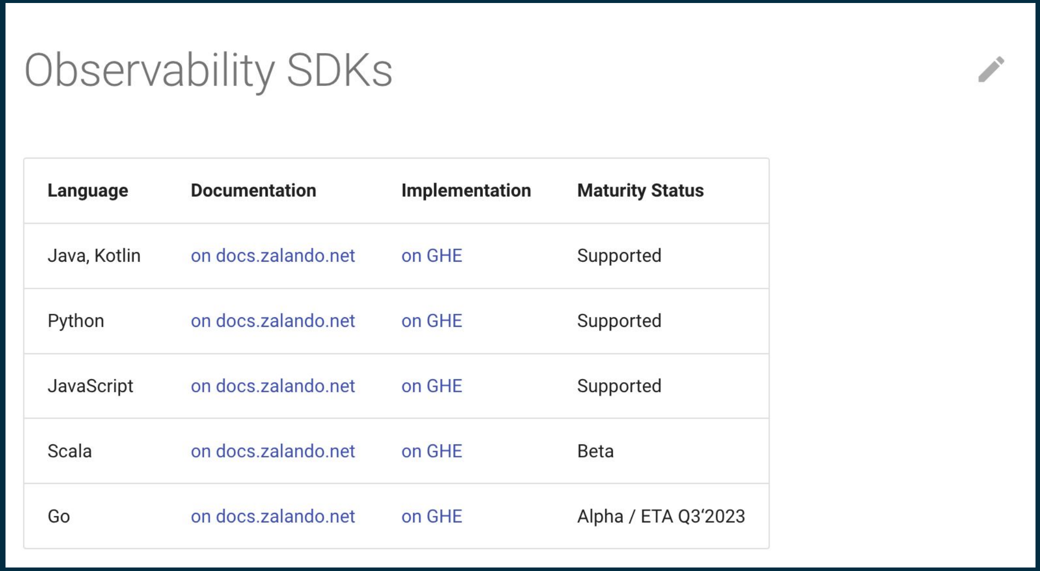
```
#!/usr/bin/env python3

import observability_sdk as obs

# Hook-up Zalando Backends
obs.initialize()

# Custom span
@obs.trace(name=..., attributes={...})
def add_to_cart():
       ...

# Custom metric
req_counter = obs.create_counter(
    name="total_requests",
    description="Total number of requests served",
    attributes = {...}
    unit="1",
    value_type=int,
)
def handle_request():
       req_counter.inc()
```



Observability SDKs

Language	Documentation	Implementation	Maturity Status
Java, Kotlin	on docs.zalando.net	on GHE	Supported
Python	on docs.zalando.net	on GHE	Supported
JavaScript	on docs.zalando.net	on GHE	Supported
Scala	on docs.zalando.net	on GHE	Beta
Go	on docs.zalando.net	on GHE	Alpha / ETA Q3'2023

OpenTelemetry

SLOs

Why SLOs?

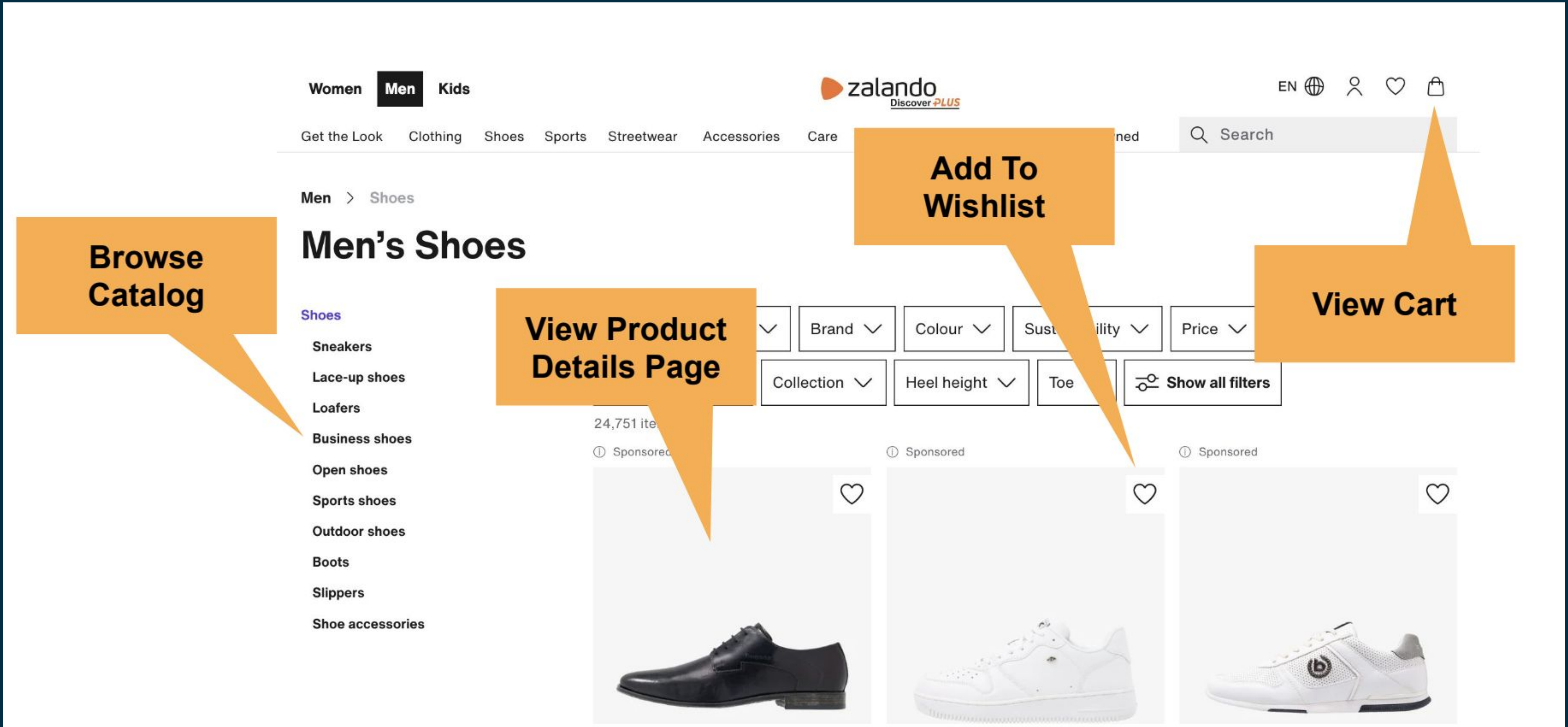
- Provide Top-Down understanding of Reliability provided to the user
- Steer engineering investments into Reliability
- Quantify impact of incidents
- … also derive high-quality alerting rules

#4 Rule of Operations

SLIs quantify the reliability of a User Experience.

SLOs are Reliability targets for managerial steering.

Zalando SLOs on Business Operations



Browse Catalog

View Product Details Page

Add To Wishlist

View Cart

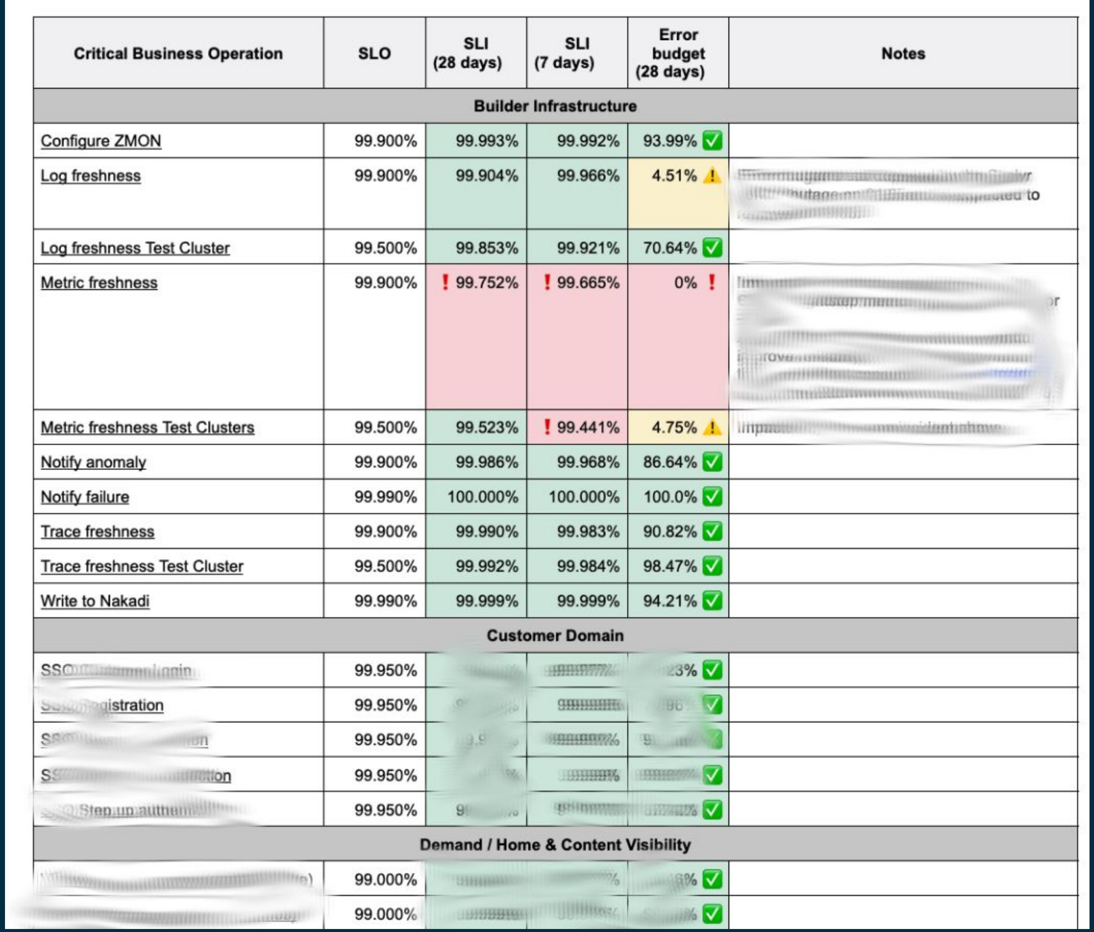
Women
Men
Kids
zalando
Discover PLUS
EN
Get the Look
Clothing
Shoes
Sports
Streetwear
Accessories
Care
ned
Search
Men > Shoes

Men’s Shoes

Shoes
Sneakers
Lace-up shoes
Loafers
Business shoes
Open shoes
Sports shoes
Outdoor shoes
Boots
Slippers
Shoe accessories

Brand
Colour
Sust ility
Price
Collection
Heel height
Toe
Show all filters
24,751 ite
Sponsored
Sponsored
Sponsored

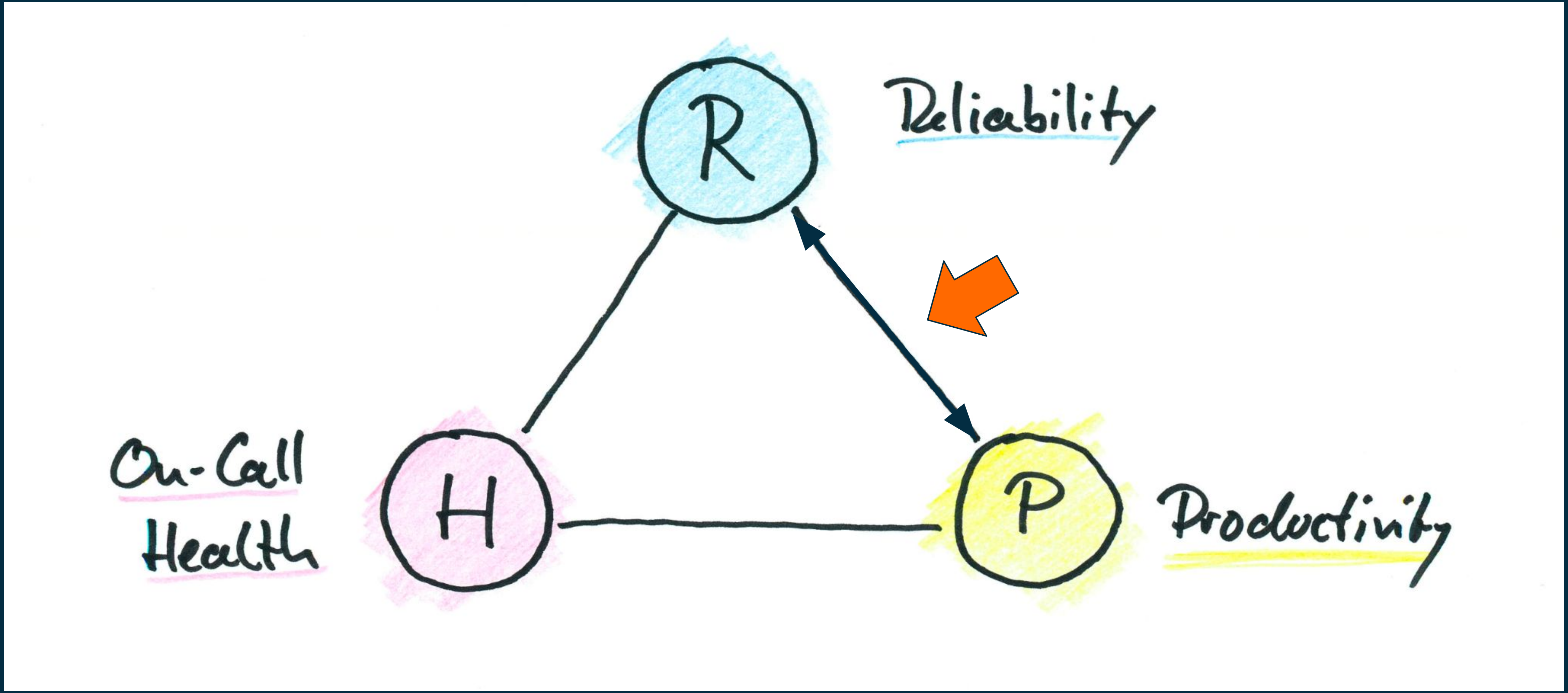
SLO Table Reviewed by Management



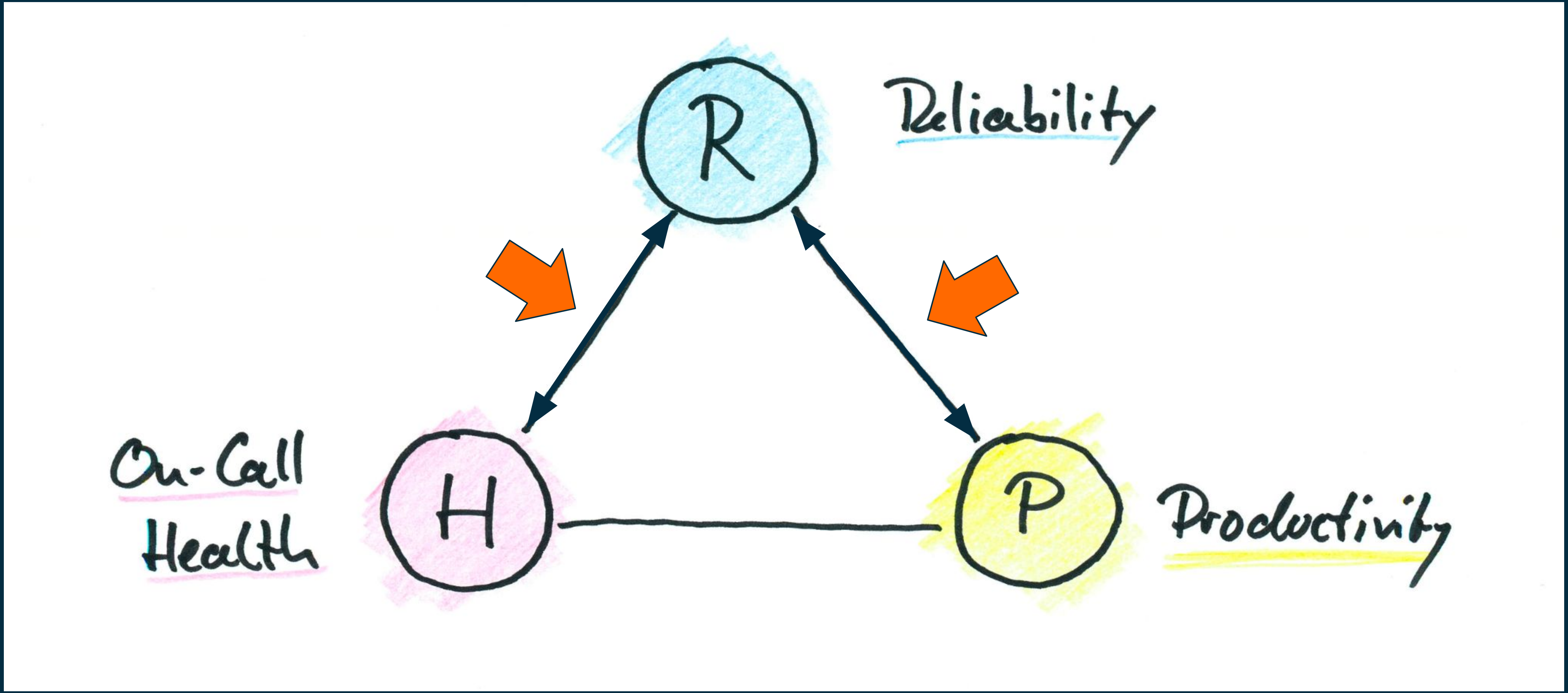
Critical Business Operation	SLO	SLI (28 days)	SLI (7 days)	Error budget (28 days)	Notes
Builder Infrastructure					
Configure ZMON	99.900%	99.993%	99.992%	93.99% ✅	
Log freshness	99.900%	99.904%	99.966%	4.51% ⚠	[redacted]
Log freshness Test Cluster	99.500%	99.853%	99.921%	70.64% ✅	
Metric freshness	99.900%	! 99.752%	! 99.665%	0% !	[redacted]
Metric freshness Test Clusters	99.500%	99.523%	! 99.441%	4.75% ⚠	[redacted]
Notify anomaly	99.900%	99.986%	99.968%	86.64% ✅	
Notify failure	99.990%	100.000%	100.000%	100.0% ✅	
Trace freshness	99.900%	99.990%	99.983%	90.82% ✅	
Trace freshness Test Cluster	99.500%	99.992%	99.984%	98.47% ✅	
Write to Nakadi	99.990%	99.999%	99.999%	94.21% ✅	
Customer Domain					
[redacted]	99.950%	[redacted]	[redacted]	[redacted] ✅	
[redacted]	99.950%	[redacted]	[redacted]	[redacted] ✅	
[redacted]	99.950%	[redacted]	[redacted]	[redacted] ✅	
[redacted]	99.950%	[redacted]	[redacted]	[redacted] ✅	
[redacted]	99.950%	[redacted]	[redacted]	[redacted] ✅	
Demand / Home & Content Visibility					
[redacted]	99.000%	[redacted]	[redacted]	[redacted] ✅	
[redacted]	99.000%	[redacted]	[redacted]	[redacted] ✅	

goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

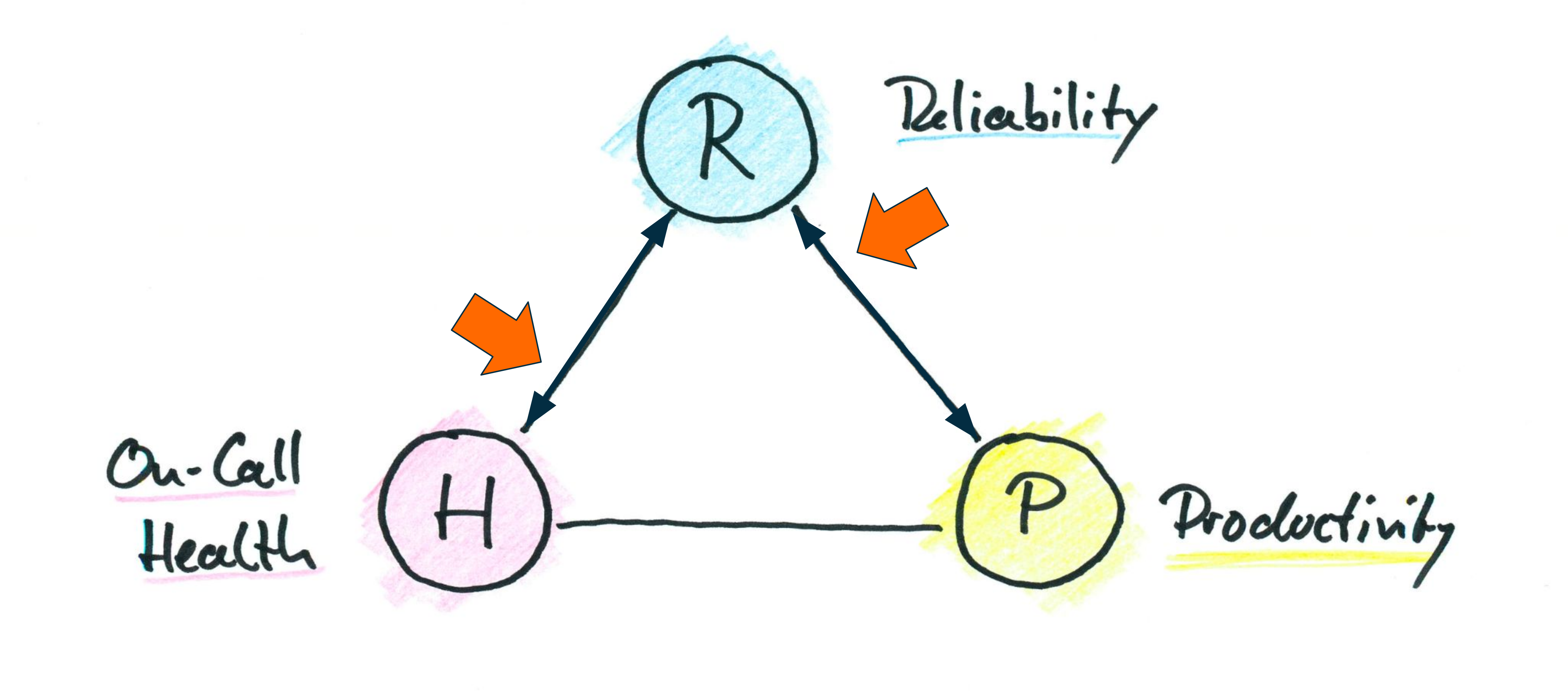
SLOs are used to Prioritize Engineering Investments



SLOs are also used to tune Alerting Sensitivity



Decouple Alerting/Reporting SLOs to get more value!

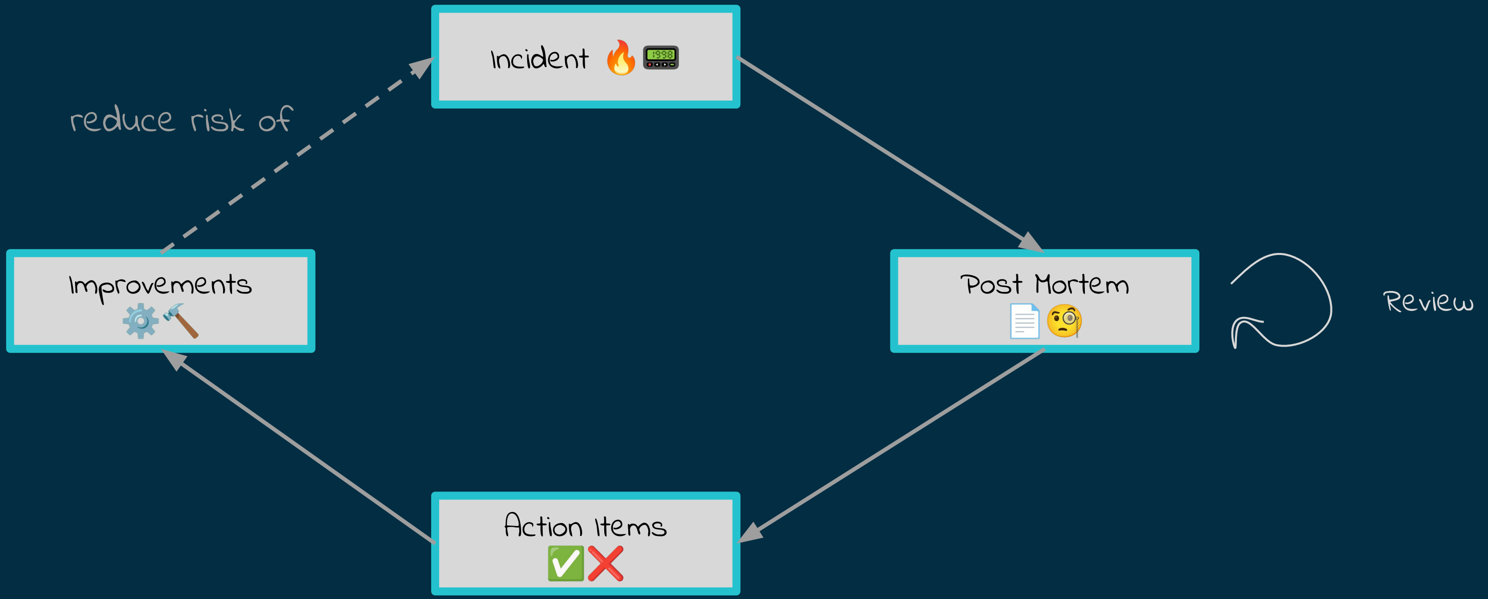


Incident Process

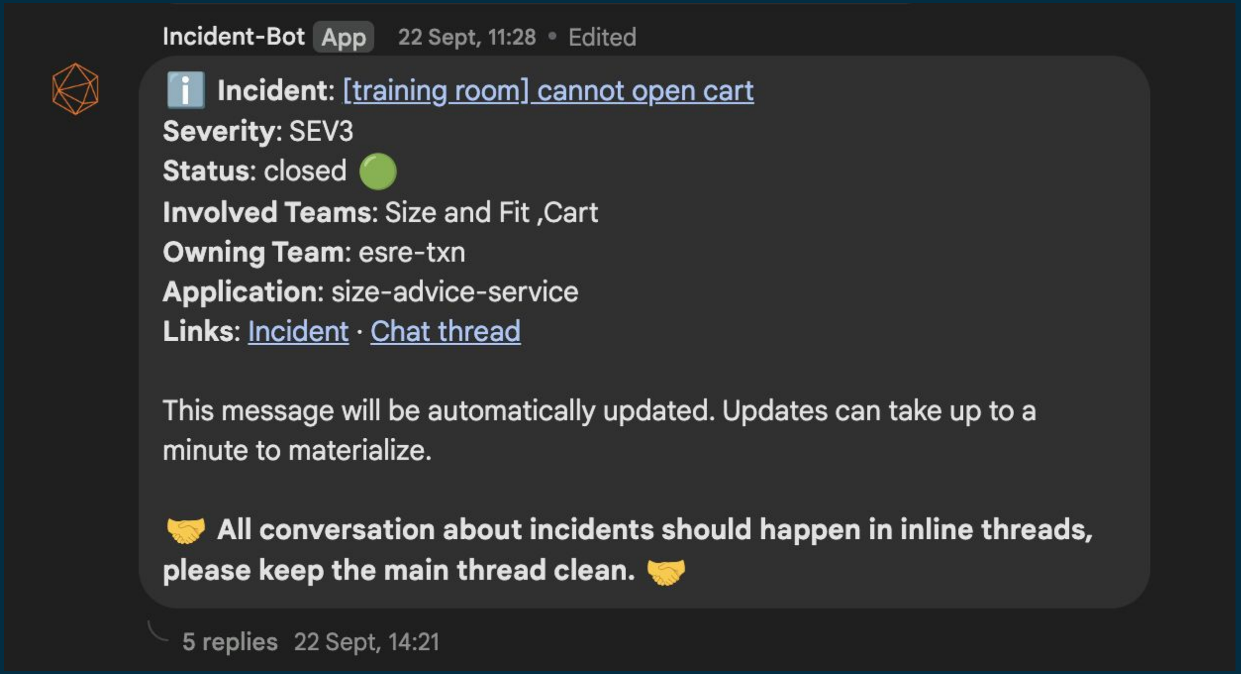
#5 Rule of Operations

Past Failures lead the way towards future Reliability.

Incident Process as Feedback Loop



Zalando Incident Process



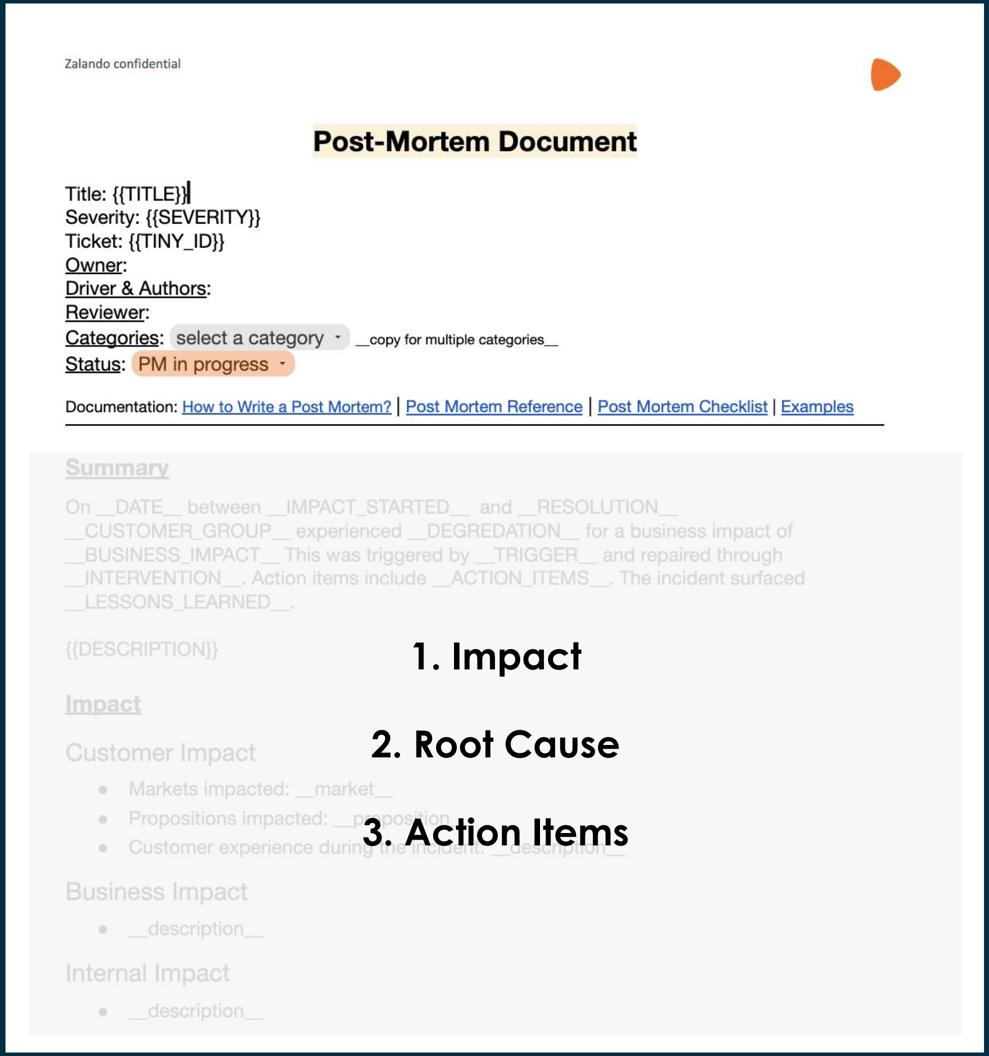
Incident-Bot App 22 Sept, 11:28 • Edited

ℹ️ **Incident:** [training room] cannot open cart
Severity: SEV3
Status: closed 🟢
Involved Teams: Size and Fit ,Cart
Owning Team: esre-txn
Application: size-advice-service
Links: Incident · Chat thread

This message will be automatically updated. Updates can take up to a minute to materialize.

🤝 **All conversation about incidents should happen in inline threads, please keep the main thread clean.** 🤝

5 replies 22 Sept, 14:21



Zalando confidential

Post-Mortem Document

Title: {{TITLE}}
Severity: {{SEVERITY}}
Ticket: {{TINY_ID}}
Owner:
Driver & Authors:
Reviewer:
Categories: select a category ▾ __copy for multiple categories__
Status: PM in progress ▾

Documentation: How to Write a Post Mortem? | Post Mortem Reference | Post Mortem Checklist | Examples

Summary

On __DATE__ between __IMPACT_STARTED__ and __RESOLUTION__ __CUSTOMER_GROUP__ experienced __DEGREDATION__ for a business impact of __BUSINESS_IMPACT__ This was triggered by __TRIGGER__ and repaired through __INTERVENTION__. Action items include __ACTION_ITEMS__. The incident surfaced __LESSONS_LEARNED__.

{{DESCRIPTION}}

Impact

Customer Impact

- Markets impacted: __market__
- Propositions impacted: __proposition__
- Customer experience during the incident: __description__

Business Impact

- __description__

Internal Impact

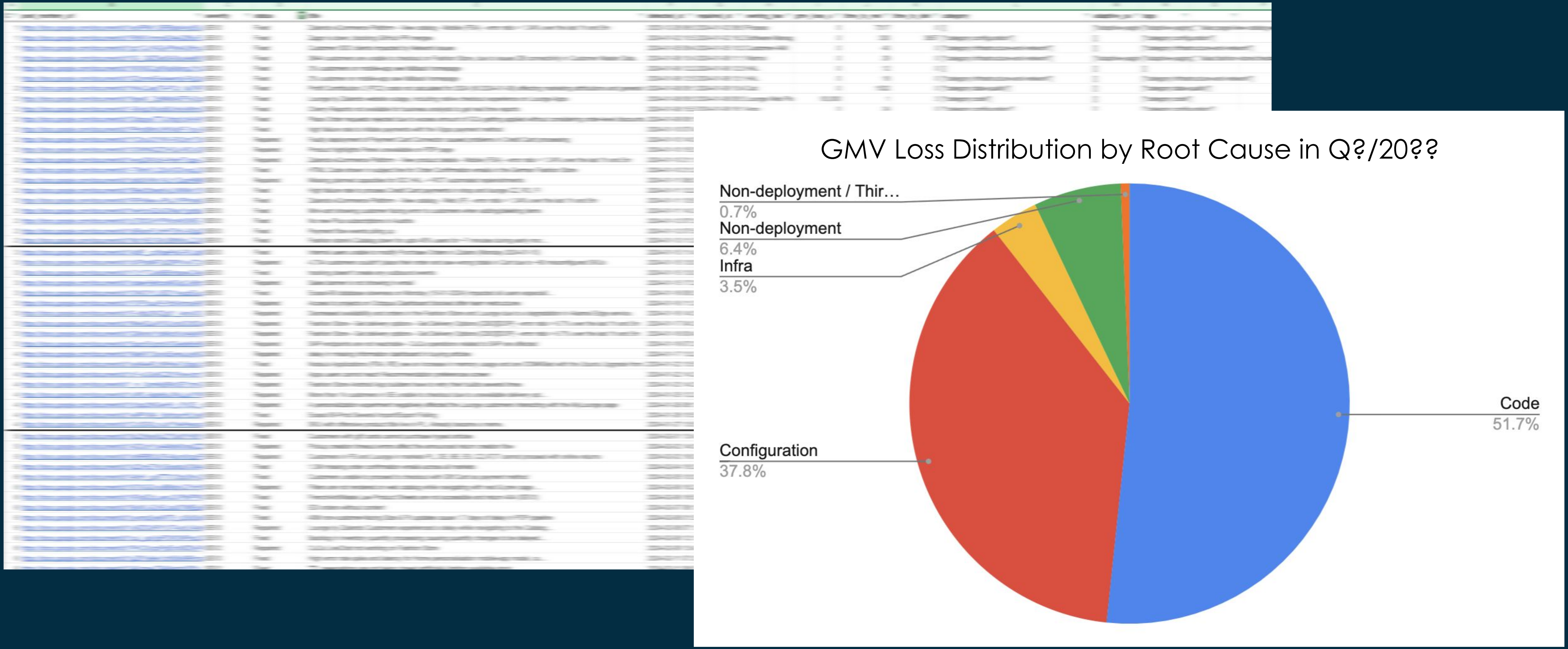
- __description__

1. Impact
2. Root Cause
3. Action Items

Zalando Severity Definitions

SEV1	**Example Incidents** ● Order Drop ● AWS Zone Outage	Ownership: Vice President
SEV2	**Example Incidents** ● Payments processor degraded ● Order confirmation emails delayed	Ownership: Director
SEV3	**Example Incidents** ● Users don't receive voucher ● Lounge users see not personalised articles	Ownership: Head of Engineering

Incident Insights every Quarter



GMV Loss Distribution by Root Cause in Q?/20??

Weekly Operational Review Meeting

#6 Rule of Operations

You get what you inspect.

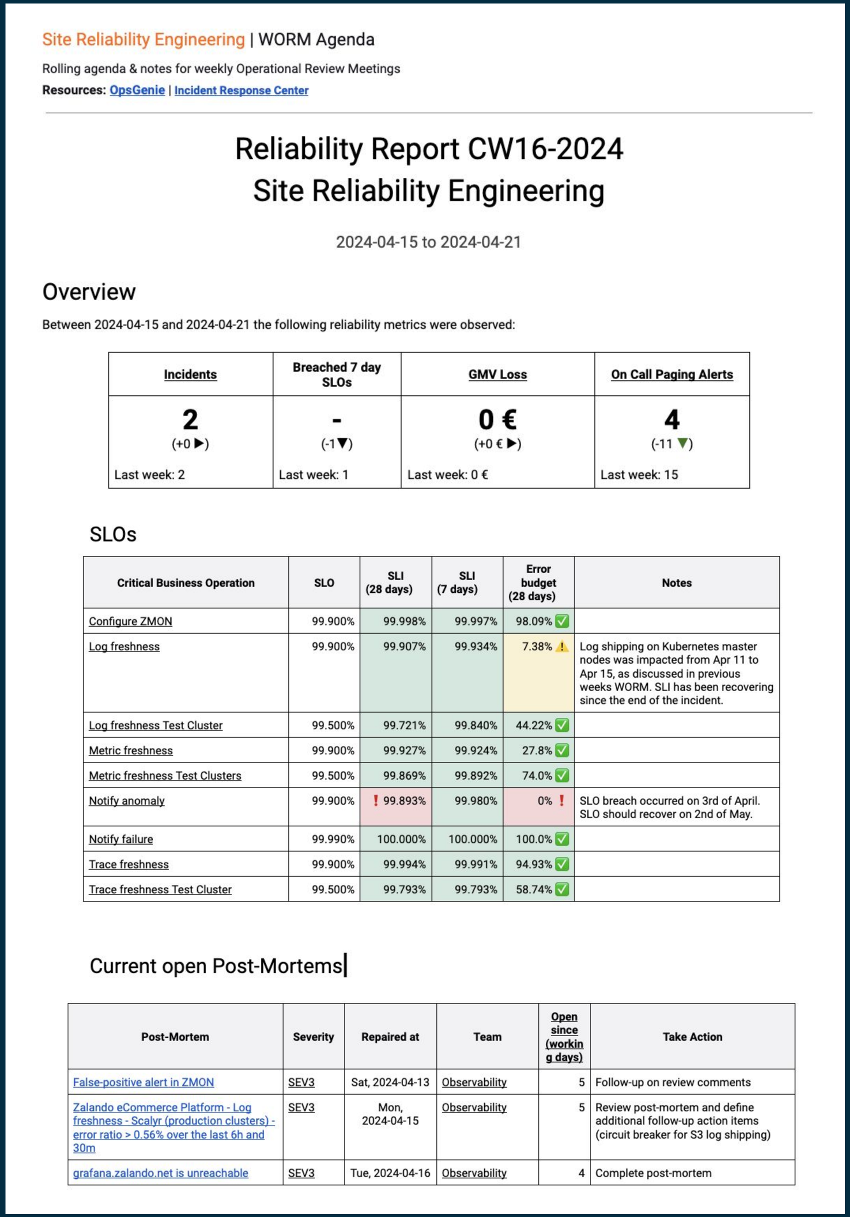
Reliability Reports

Supporting WORM Meetings on all Levels

Auto Generated Google Doc

WORM Agenda

- Incident Review -> Patterns?
- SLO Review
- Open Post Mortems
- On-Call Health



Site Reliability Engineering | WORM Agenda

Rolling agenda & notes for weekly Operational Review Meetings

Resources: OpsGenie | Incident Response Center

Reliability Report CW16-2024
Site Reliability Engineering

2024-04-15 to 2024-04-21

Overview

Between 2024-04-15 and 2024-04-21 the following reliability metrics were observed:

Incidents	Breached 7 day SLOs	GMV Loss	On Call Paging Alerts
2 (+0 ▶) Last week: 2	**-** (-1▼) Last week: 1	**0 €** (+0 € ▶) Last week: 0 €	**4** (-11 ▼) Last week: 15

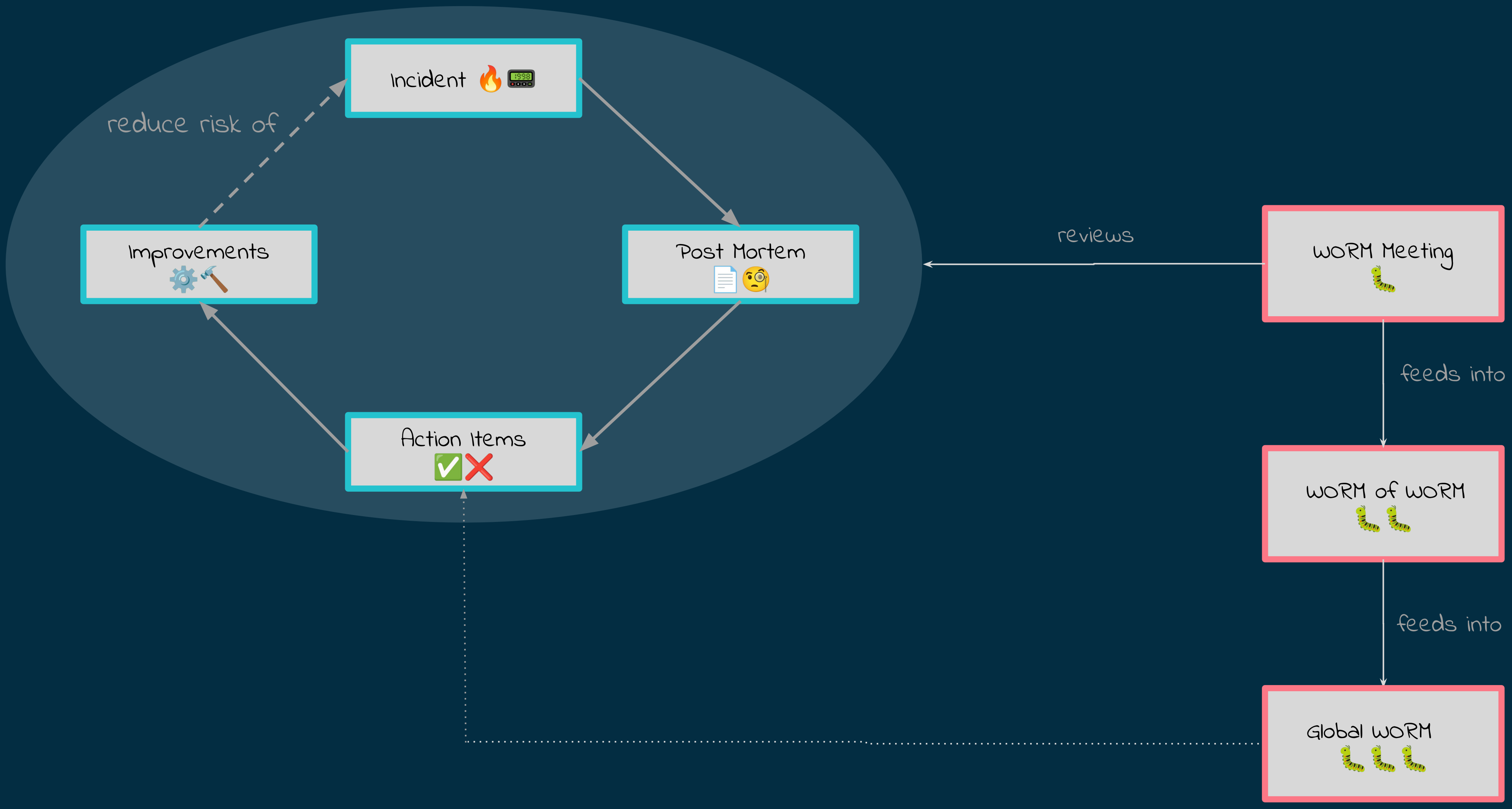
SLOs

Critical Business Operation	SLO	SLI (28 days)	SLI (7 days)	Error budget (28 days)	Notes
Configure ZMON	99.900%	99.998%	99.997%	98.09% ✅	
Log freshness	99.900%	99.907%	99.934%	7.38% ⚠️	Log shipping on Kubernetes master nodes was impacted from Apr 11 to Apr 15, as discussed in previous weeks WORM. SLI has been recovering since the end of the incident.
Log freshness Test Cluster	99.500%	99.721%	99.840%	44.22% ✅	
Metric freshness	99.900%	99.927%	99.924%	27.8% ✅	
Metric freshness Test Clusters	99.500%	99.869%	99.892%	74.0% ✅	
Notify anomaly	99.900%	❗ 99.893%	99.980%	0% ❗	SLO breach occurred on 3rd of April. SLO should recover on 2nd of May.
Notify failure	99.990%	100.000%	100.000%	100.0% ✅	
Trace freshness	99.900%	99.994%	99.991%	94.93% ✅	
Trace freshness Test Cluster	99.500%	99.793%	99.793%	58.74% ✅	

Current open Post-Mortems

Post-Mortem	Severity	Repaired at	Team	Open since (working days)	Take Action
False-positive alert in ZMON	SEV3	Sat, 2024-04-13	Observability	5	Follow-up on review comments
Zalando eCommerce Platform - Log freshness - Scalyr (production clusters) - error ratio > 0.56% over the last 6h and 30m	SEV3	Mon, 2024-04-15	Observability	5	Review post-mortem and define additional follow-up action items (circuit breaker for S3 log shipping)
grafana.zalando.net is unreachable	SEV3	Tue, 2024-04-16	Observability	4	Complete post-mortem

Zalando WORM Cascade



goto; Amsterdam 20204. Heinrich Hartmann @ Zalando

Rules of Operations

1. Obsess about User Experience.
2. Engineering for Reliability involves People & Technology.
3. Alert on User Pain ("Symptoms") not Server Pain ("Causes").
4. SLIs quantify the reliability of a User Experience.
5. Past Failures lead the way towards future Reliability.
6. You get what you inspect.

Thank you!

> Heinrich@HeinrichHartmann.com
#Let's talk Reliability! 💚