Frontiers of Reliability Engineering

SRECon EMEA 2024 - Heinrich Hartmann

👋 I'm Heinrich - Reliability Engineer

Personal Experience

- Led Zalando SRE for 2.5 years
- Now Senior Principal SRE
- 10 years of Reliability Engineering
- Chief Data Scientist @ Circonus
- Math PhD

- A Leading Fashion Platform in Europe
- 3K Software Engineers
- 3K+ Micro services
- 250 Kubernetes Clusters
- 50M+ customers
- 14.6 bn EUR Revenue

Find me on LinkedIn

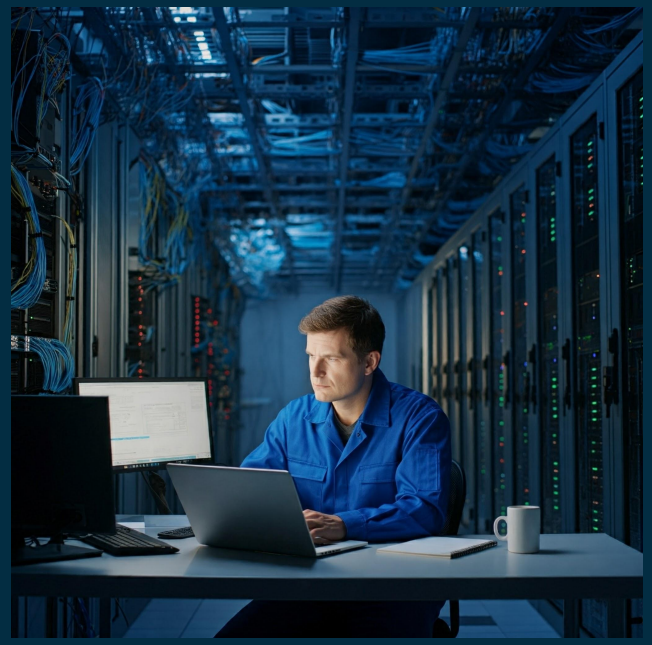
Menu

1. What have we achieved?
2. Principles
3. Where are we going?
   a. Managing for Reliability
   b. Mobile Observabity
   c. Data Operations

What have we Achieved?

Hardware Provisioning & Capacity Planning

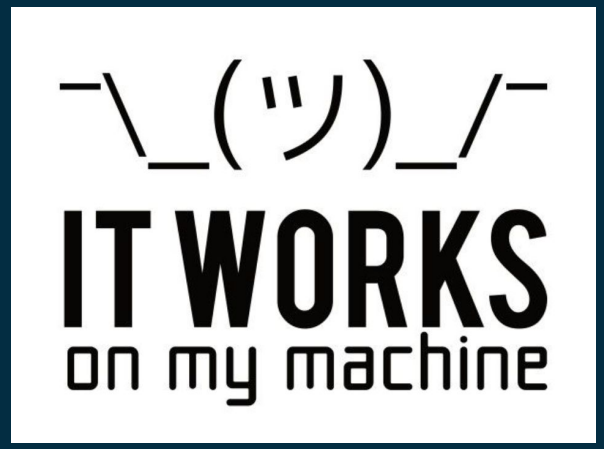
2014



2024

Packaging and Deployment

2014



2024

Monitoring

2014

graphite

StatsD chat on gitter docker pulls 11M

Ganglia

collectd
The system statistics collection daemon

SSH

2024

Prometheus

Observability

2014

Dapper at Google

Google Technical Report dapper-2010-1, April 2010

Dapper, a Large-Scale Distributed Systems Tracing Infrastructure

Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, Chandan Shanbhag

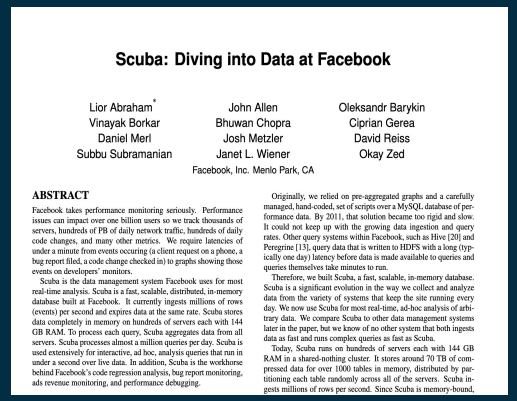
Abstract

Modern Internet services are often implemented as complex, large-scale distributed systems. These applications are constructed from collections of software modules that may be developed by different teams, perhaps in different programming languages, and could span many thousands of machines across multiple physical facilities. Tools that aid in understanding system behavior and reasoning about performance issues are invaluable in such an environment.

Here we introduce the design of Dapper, Google's production distributed systems tracing infrastructure, and describe how our design goals of low overhead, because large collections of small servers are a particularly cost-efficient platform for Internet services workloads [4]. Understanding system behavior in this context requires observing related activities across many different programs and machines.

A web-search example will illustrate some of the challenges such a system needs to address. A front-end service may distribute a web query to many hundreds of query servers, each searching within its own piece of the index. The query may also be sent to a number of other sub-systems that may process advertisements, check spelling, or look for specialized results, including images, videos, news, and so on. Results from all

Scuba at Facebook



Scuba: Diving into Data at Facebook

Lior Abraham* Vinayak Borkar Daniel Merl Subbu Subramanian

John Allen Bhuwan Chopra Josh Metzler Janet L. Wiener

Oleksandr Barykin Ciprian Gerea David Reiss Okay Zed

Facebook, Inc. Menlo Park, CA

ABSTRACT

Facebook takes performance monitoring seriously. Performance issues can impact over one billion users so we track thousands of servers, hundreds of PB of daily network traffic, hundreds of daily code changes, and many other metrics. We require latencies of under a minute from events occuring (a client request on a phone, a bug report filed, a code change checked in) to graphs showing those events on developers' monitors.

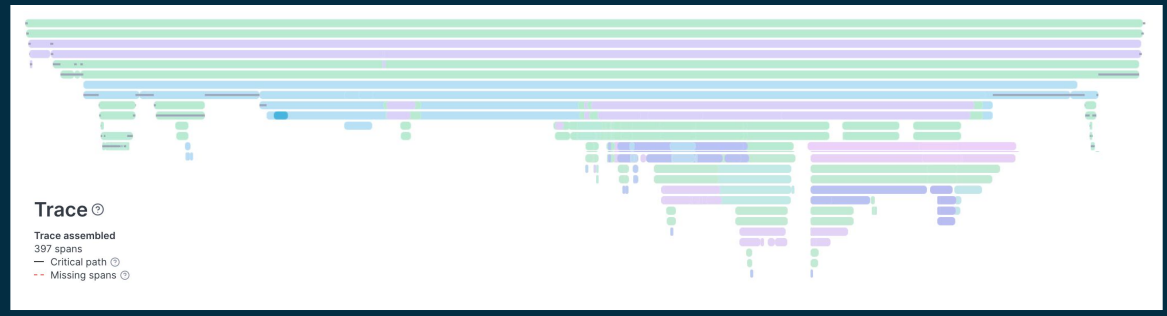
Scuba is the data management system Facebook uses for most real-time analysis. Scuba is a fast, scalable, distributed, in-memory database built at Facebook. It currently ingests millions of rows (events) per second and expires data at the same rate. Scuba stores data completely in memory on hundreds of servers each with 144 GB RAM. To process each query, Scuba aggregates data from all servers. Scuba processes almost a million queries per day. Scuba is used extensively for interactive, ad hoc, analysis queries that run in under a second over live data. In addition, Scuba is the workhorse behind Facebook's code regression analysis, bug report monitoring, ads revenue monitoring, and performance debugging.

Originally, we relied on pre-aggregated graphs and a carefully managed, hand-coded, set of scripts over a MySQL database of performance data. By 2011, that solution became too rigid and slow. It could not keep up with the growing data ingestion and query rates. Other query systems within Facebook, such as Hive [20] and Peregrine [13], query data that is written to HDFS with a long (typically one day) latency before data is made available to queries and queries themselves take minutes to run.

Therefore, we built Scuba, a fast, scalable, in-memory database. Scuba is a significant evolution in the way we collect and analyze data from the variety of systems that keep the site running every day. We now use Scuba for most real-time, ad-hoc analysis of arbitrary data. We compare Scuba to other data management systems later in the paper, but we know of no other system that both ingests data as fast and runs complex queries as fast as Scuba.

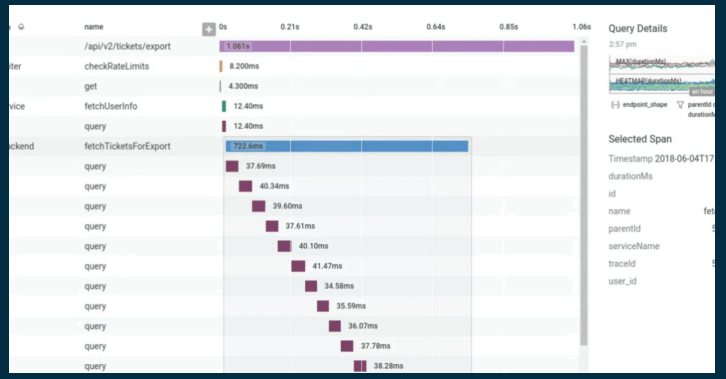
Today, Scuba runs on hundreds of servers each with 144 GB RAM in a shared-nothing cluster. It stores around 70 TB of compressed data for over 1000 tables in memory, distributed by partitioning each table randomly across all of the servers. Scuba ingests millions of rows per second. Since Scuba is memory-bound,

2024



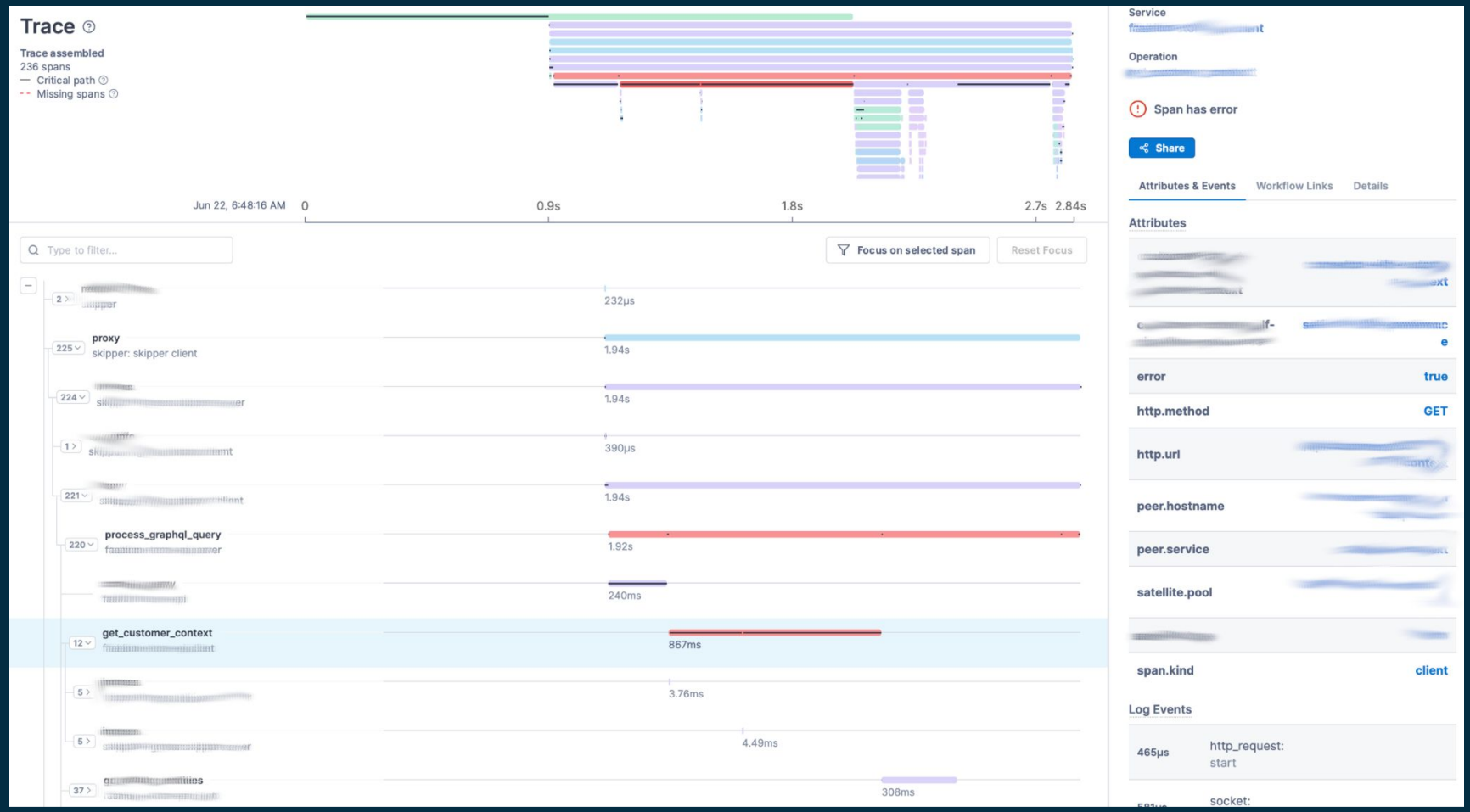
Lightstep from ServiceNow

ServiceNow Cloud Observability



honeycomb.io

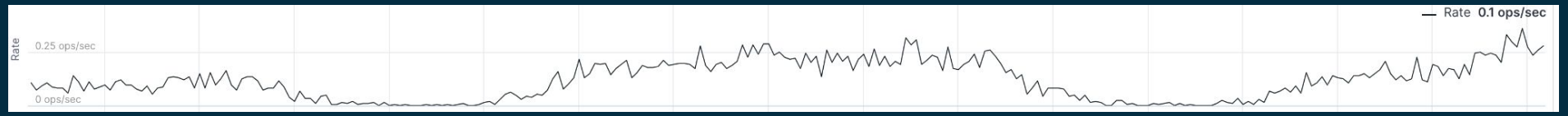
Microservice Observability w/ Tracing



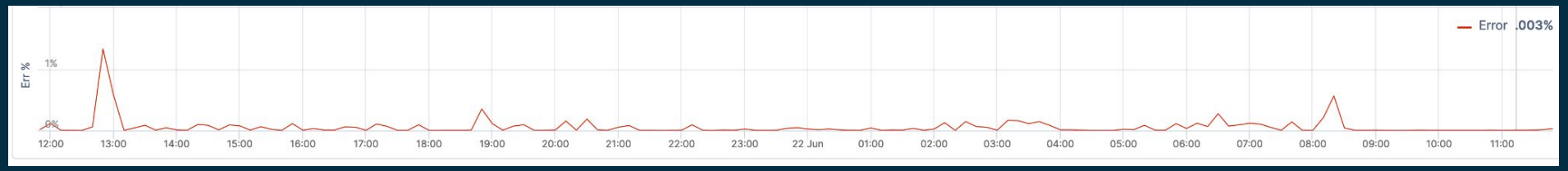
ServiceNow
Cloud Observability

Metrics & SLOs can be derived from Tracing Data

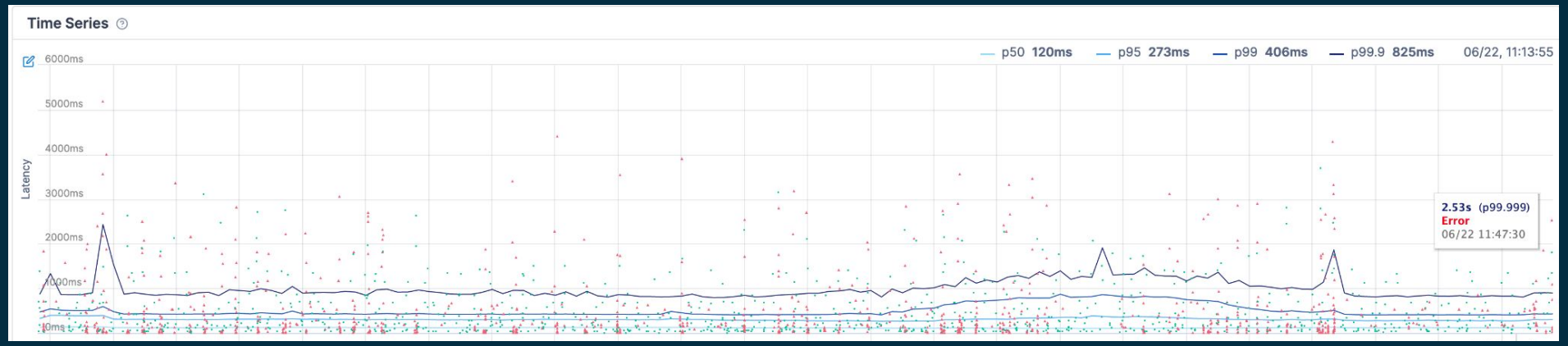
Requests



Errors



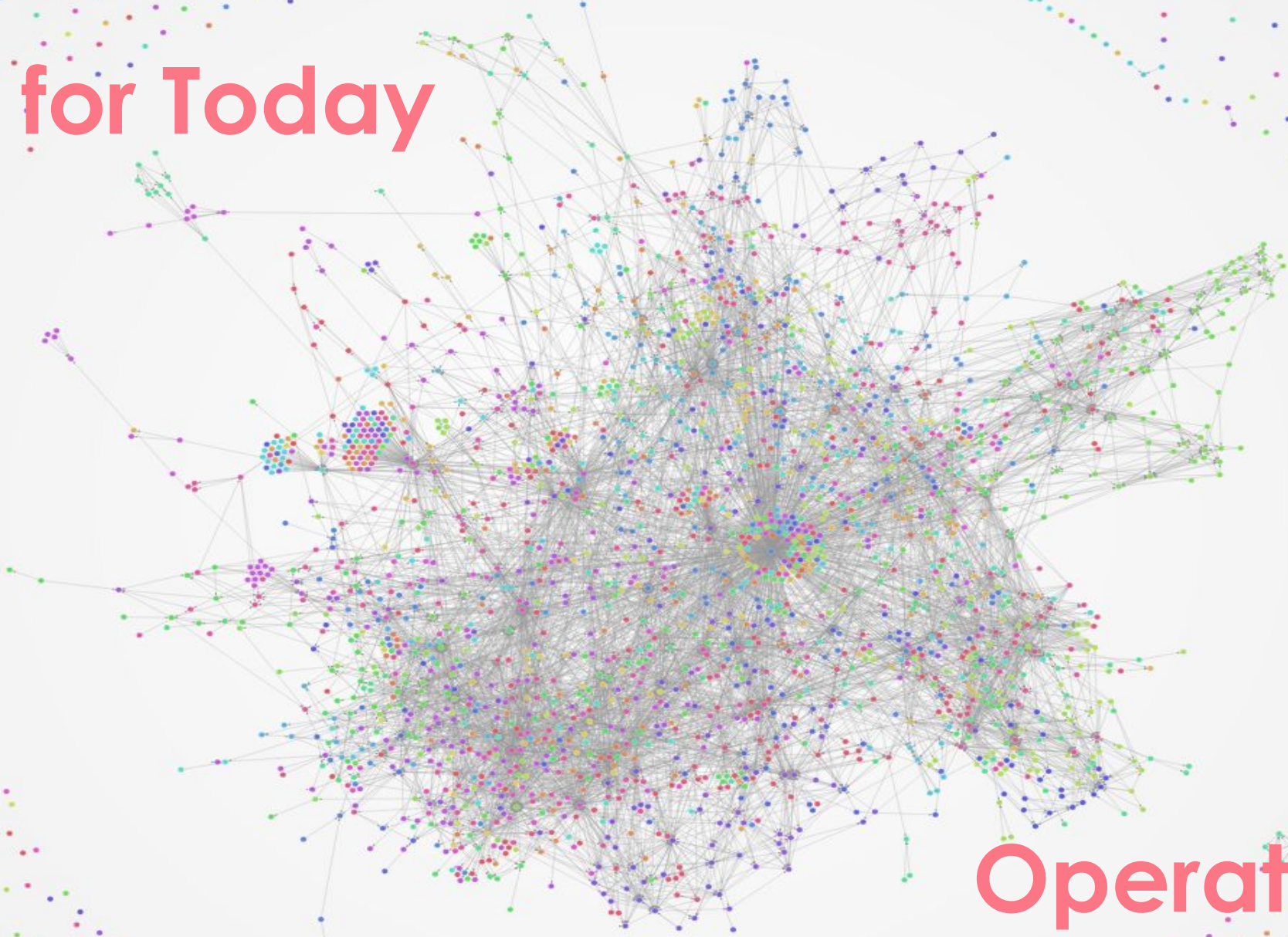
Duration



ServiceNow
Cloud Observability

Principles

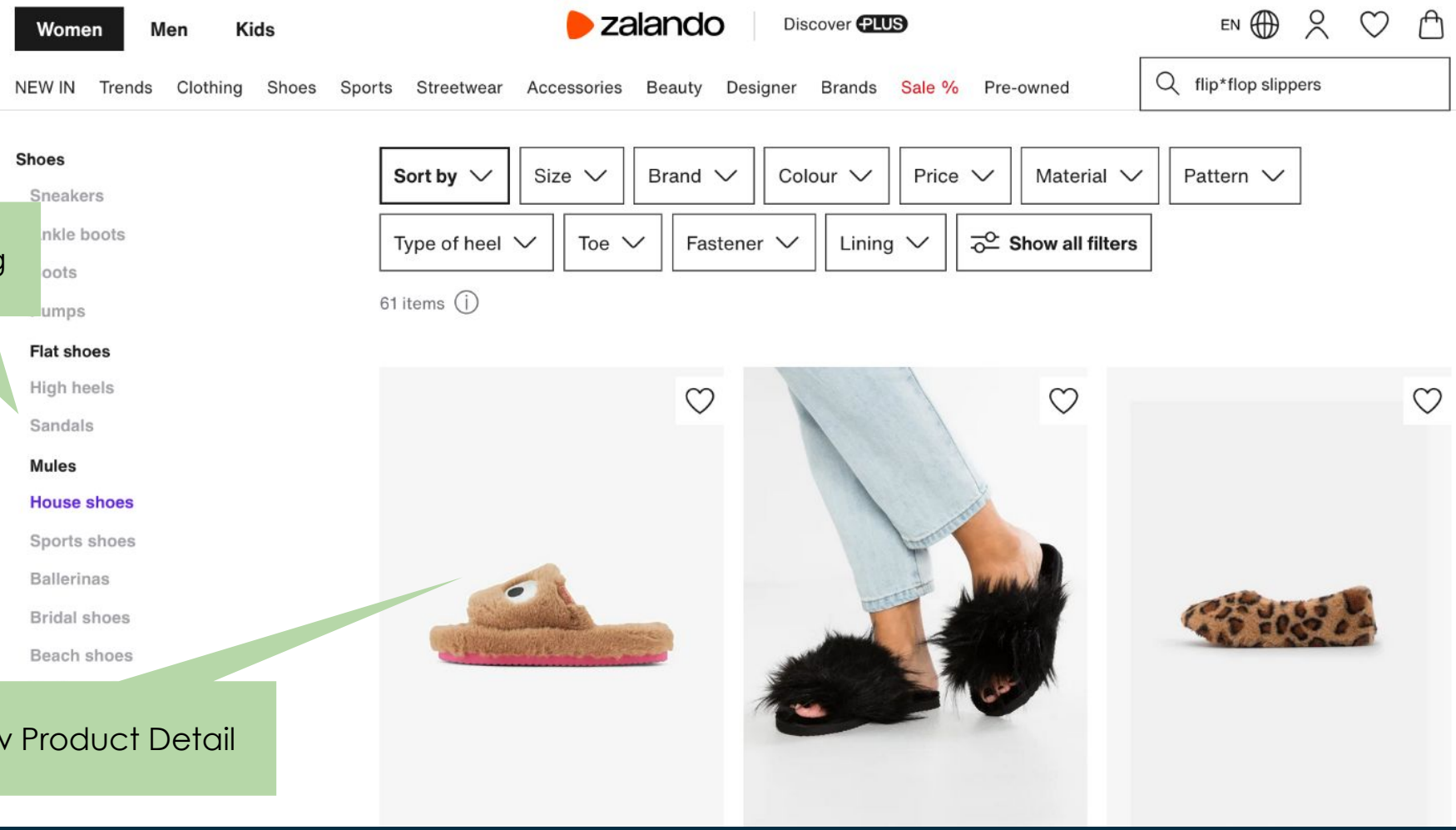
Quest for Today



Operate This!

Zalando Servie Graph - 2019

Protect the User Experience!



Browse Catalog

View Cart

View Product Detail


I don't care if your Data Center is on fire.
This is fine!

“

Reliability Engineering involves People as much as Technology.

Engineering Reliability at Scale

Small Company (~10 FTE)

- Alerts & Dashboards
- Logging
- On-call rotations

Medium Company (~100 FTE)

- Playbooks
- Incident Management
- Observability

Large Company (>1k FTE)

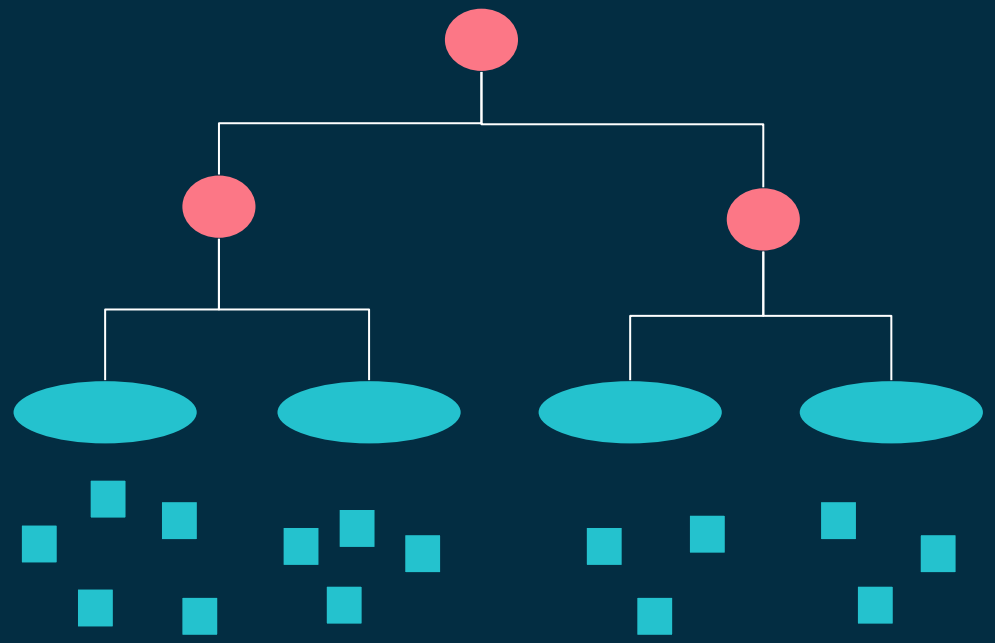
- WORM Meetings
- Risk Management
- SRE Community

Layerd Systems Model of Zalando

Management

Engineering

Platform

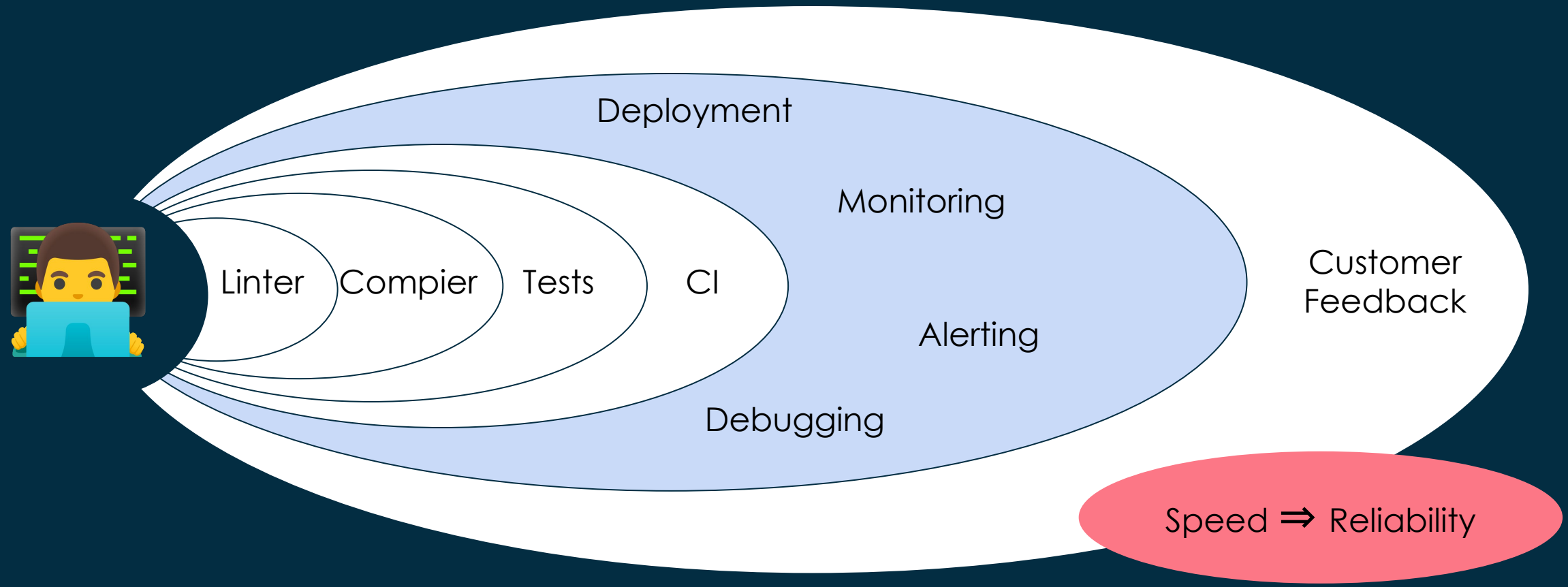


GitHub, AWS, K8S, CI/CD
Managed Postgres, Kafka, …

“

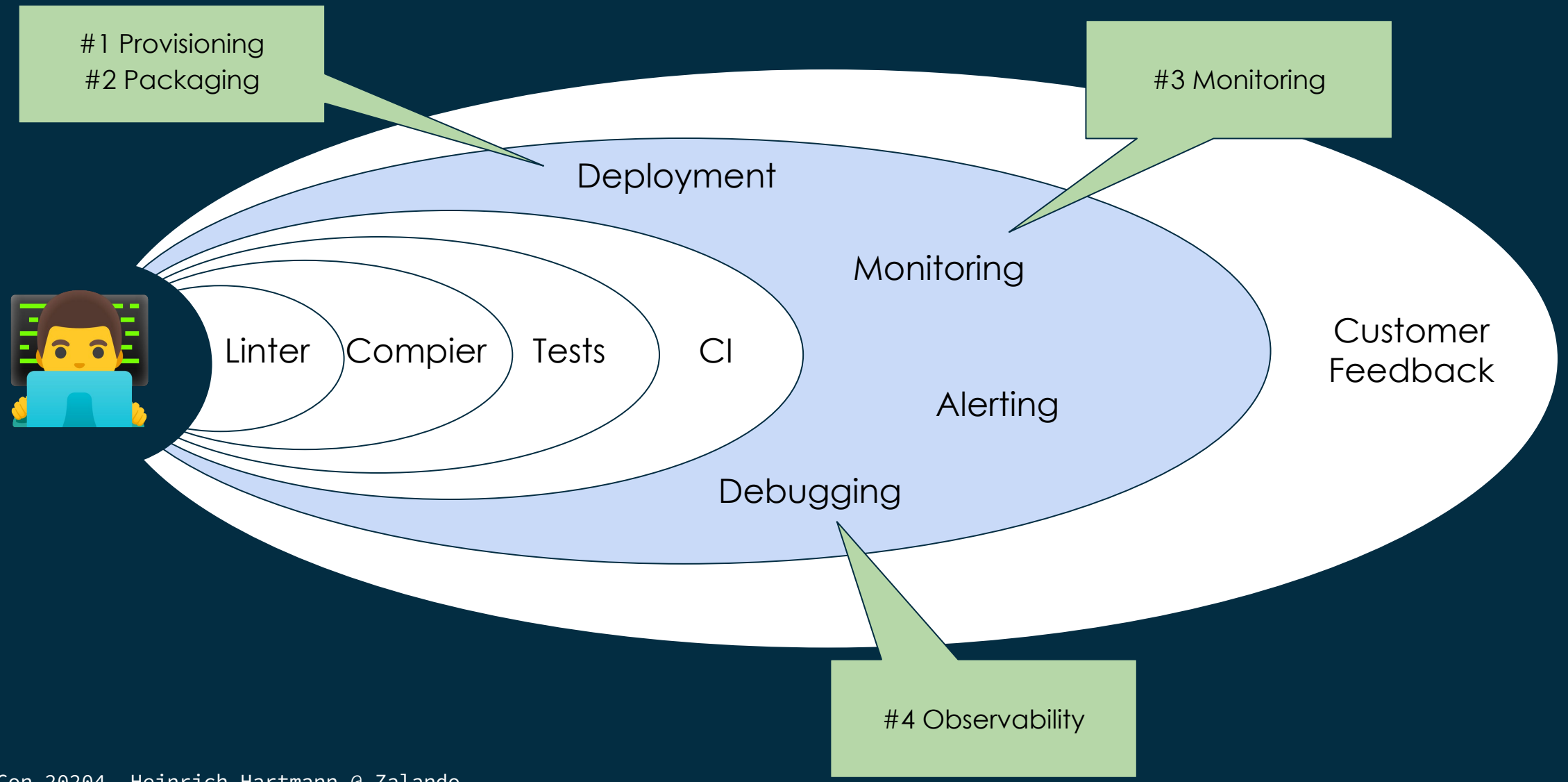
Reliability Engineering is all about Feedback Loops.

Reliability is driven by Feedback Loops

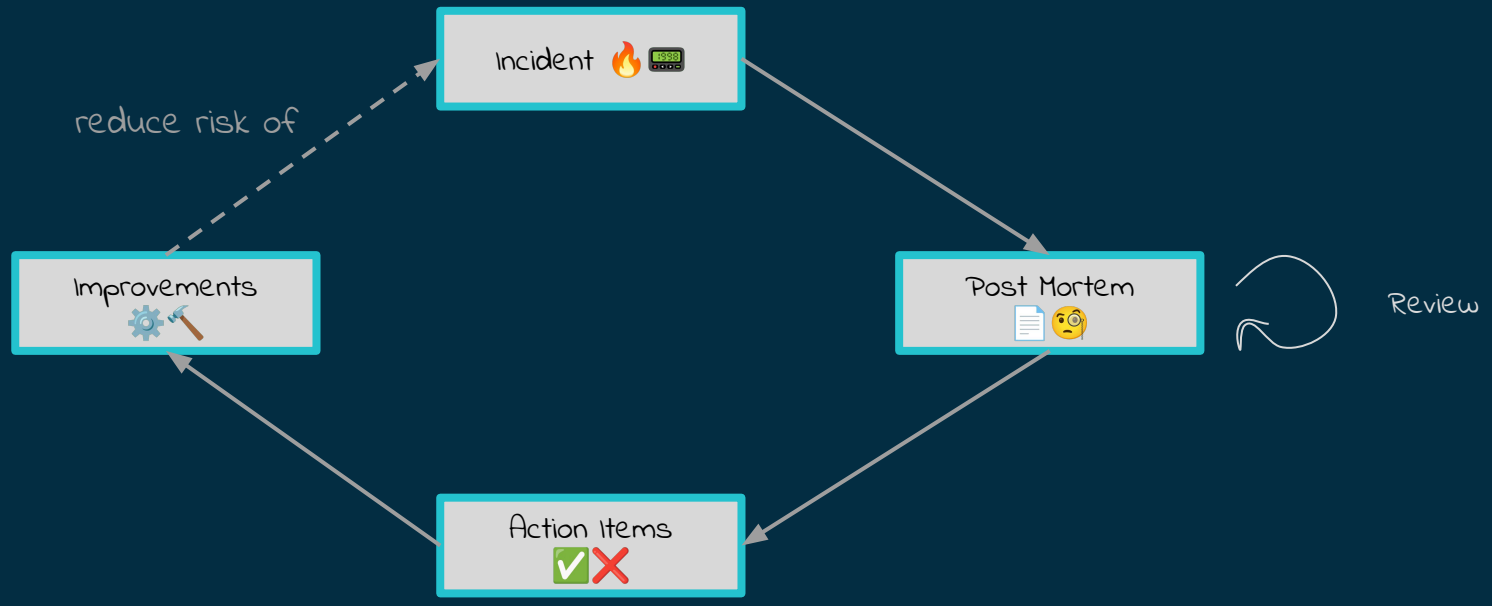


% Martin Thwaites

Achievements Accellerate Feedback Loop

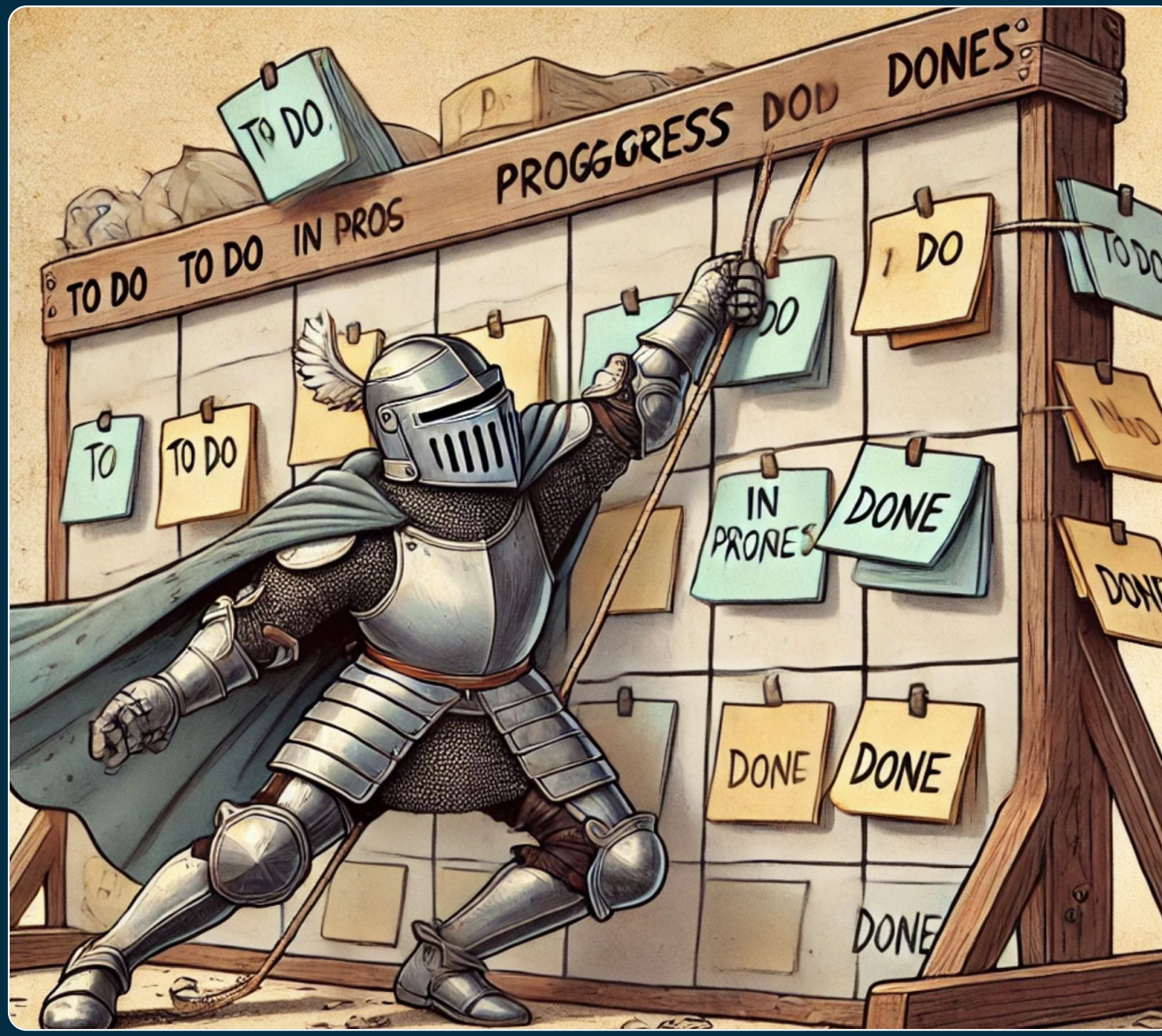


The Incident Process is a Meta Feedback Loop



Where are we going?

Managing for Reliability



How to enable on Managment to steer for reliability?

“

You get what you inspect.

Reliability Reports

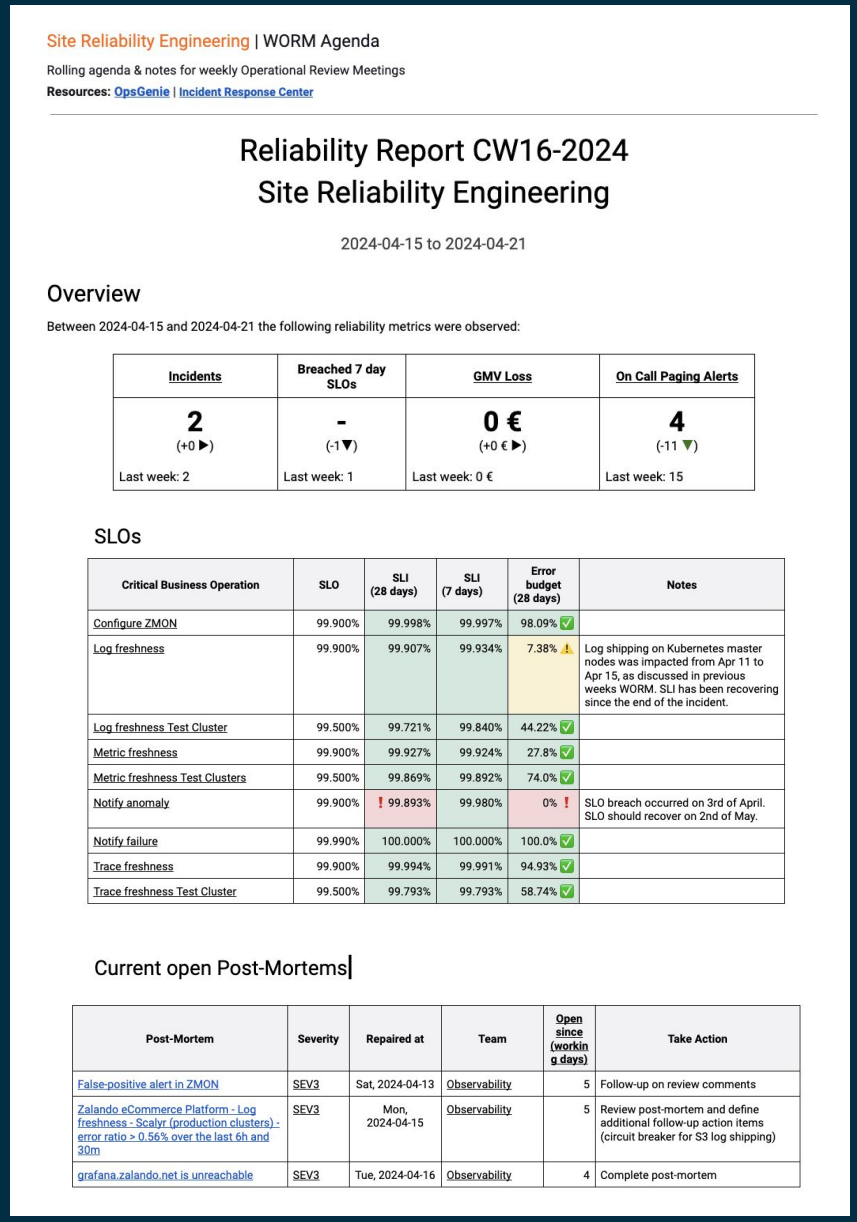
for Management on all Levels

Auto-generated Google Doc supporting

Weekly Operational Review Meetings (“WORM”).

Agenda

- Incidents
- SLO
- On-Call Health
- Open Post Mortems



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 ▶)	**-** (-1 ▼)	**0 €** (+0 € ▶)	**4** (-11 ▼)
Last week: 2	Last week: 1	Last week: 0 €	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

Incident Table shows where we failed the User

Title	Metadata	Impact	Take action
Zalando eCommerce Platform Engineering / Builder Infrastructure			
website unavailable	**Owning Team:** Cloud Governance **Detected at:** Wed, 2024-10-02 09:27 **TTD:** 1h 40m **TTR:** 35m	**Severity:** SEV3 **Categories:** Third-Party **Impact:** GMV Loss: 0 EBIT Loss: 0 Affected Customers: 0 Affected Partners: 0 Affected Employees: 0 Involved Applications:	**Effect:** still unclear **Cause:** Unknown, the website is managed by an external agency **Action:** Clarify ownership of the website
data-governance availability zone b in chrashloop backoff	**Owning Team:** Observability **Detected at:** Sun, 2024-10-06 08:28 **TTD:** 9m **TTR:** 22m	**Severity:** SEV3 **Categories:** **Impact:** GMV Loss: - EBIT Loss: - Affected Customers: Affected Partners: Affected Employees: Involved Applications:	**Effe** **Cause:** investigation ongoing **Action:**
ZMON backend `zmon.zalando.net` slow/unavailable leading to stale entities and false positive alerts	**Owning Team:** Observability **Detected at:** Wed, 2024-10-02 12:55 **TTD:** 16m **TTR:** 47m	**Severity:** SEV3 **Categories:** **Impact:** GMV Loss: - EBIT Loss: - Affected Customers: 0 Affected Partners: 0 Affected Employees: Involved Applications: - zmon	**Effect:** Creation of false positive alerts as well as delayed metrics ingestion. **Cause:** The incident was triggered by the traffic switch of ZMON entity management requests from the ZMON Controller to a new dedicated entity management service. The root cause for the observed effects is still under investigation. **Action:** Traffic was switched back to ZMON Controller. Additional testing is being conducted to understand why the traffic switch caused unexpected effects (sudden increase in total entity count) that were not observed before, when the new service was running in shadow mode.

Teams report on

1. **Impact**
2. **Cause**
3. **Actions**

SLOs provides top-down view on Reliability

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] ✅	

Teams report on

1. **Impact**
2. **Cause**
3. **Actions**

NOBL9 RELIABILITY CENTER

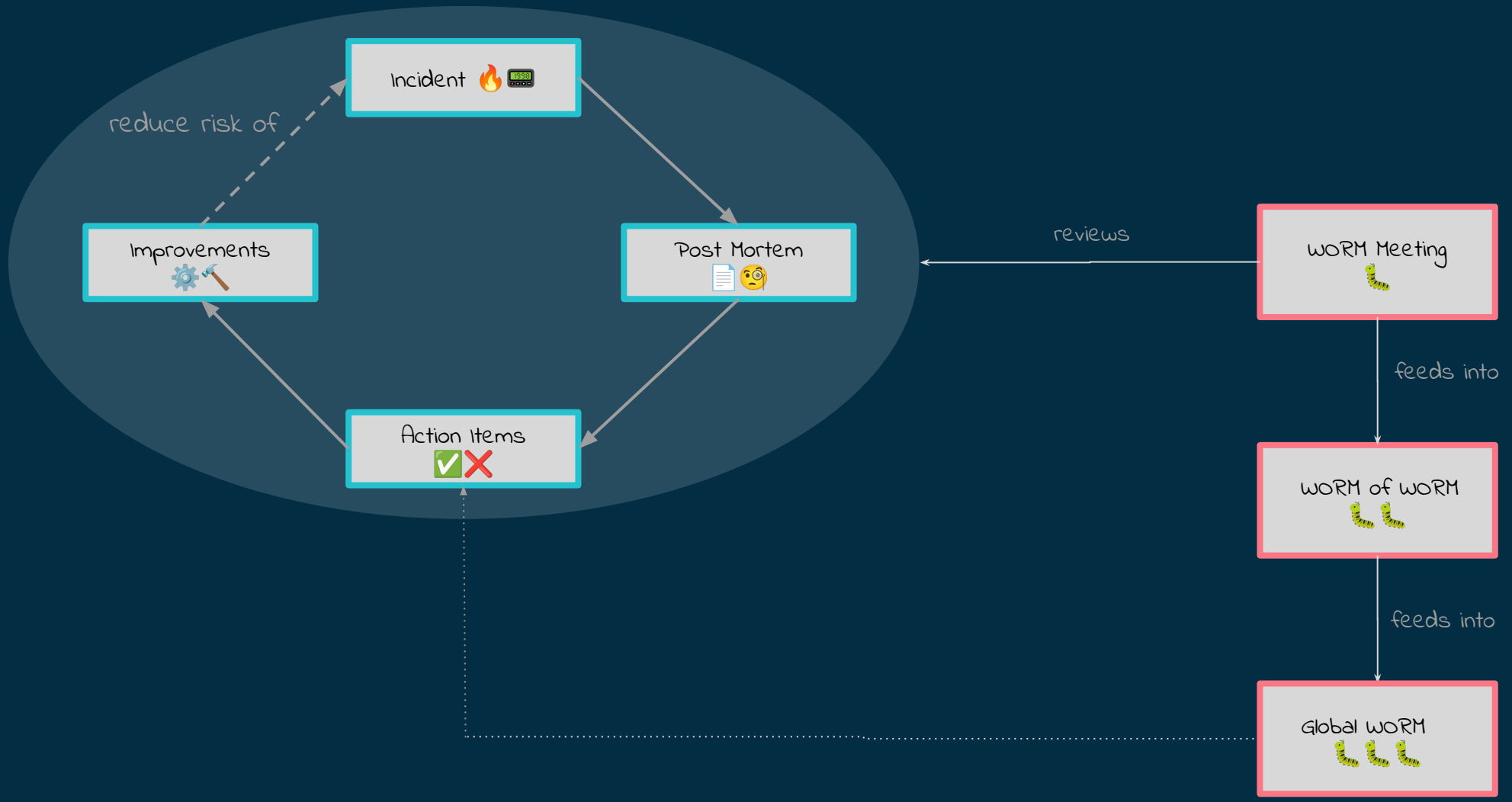
We check Alerting loads for All On-Call teams

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	consciously
[redacted]	-	-	-	5	-	7	-	1 (+0 ▶)	11 (+11 ▲)	12 (+11 ▲)	1.71 / day	ert, the sitives ons (e.g. resilience
[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]	to the ay

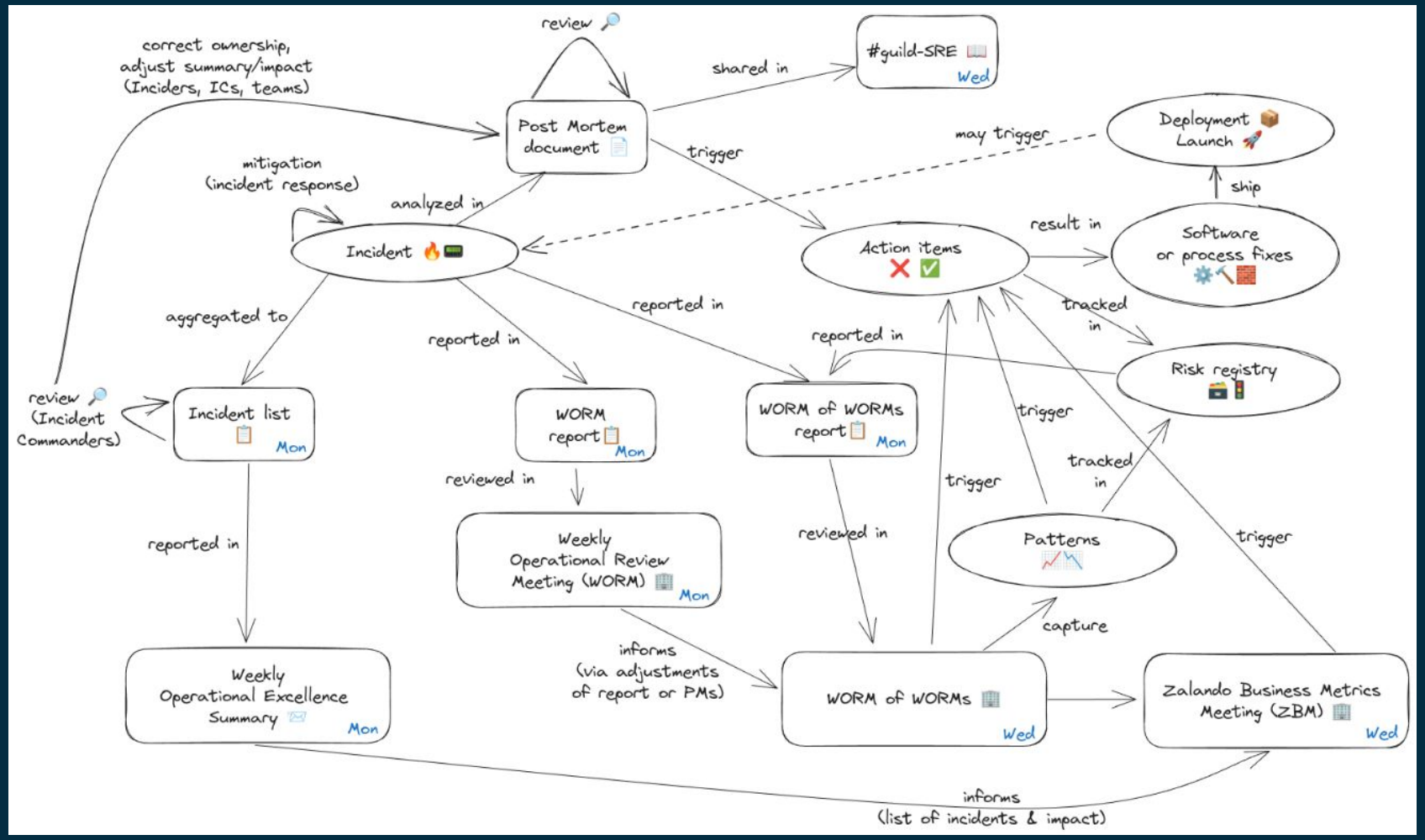
What caused the alerts?

What are we doing to prevent this?

Weekly Operationl Review Meeting (WORM) Cascade

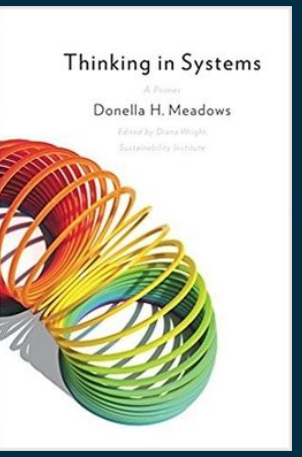


Further Augmentations of the Incident Process



The Quest Continues

- How to steer Managment Attention on Reliability?
  - Expand coverage of Reliability Reports
  - Apply “Systems Engineering” to Reliability
- How to drive Cross-Organisational Reliability Initiatives?
  - SRE/Champion Model
- How to increase value from Incident Process?
  - Post Motem AI Capabilities



Aptible

Mobile Observability

Incident Story

The undetected Order Drop

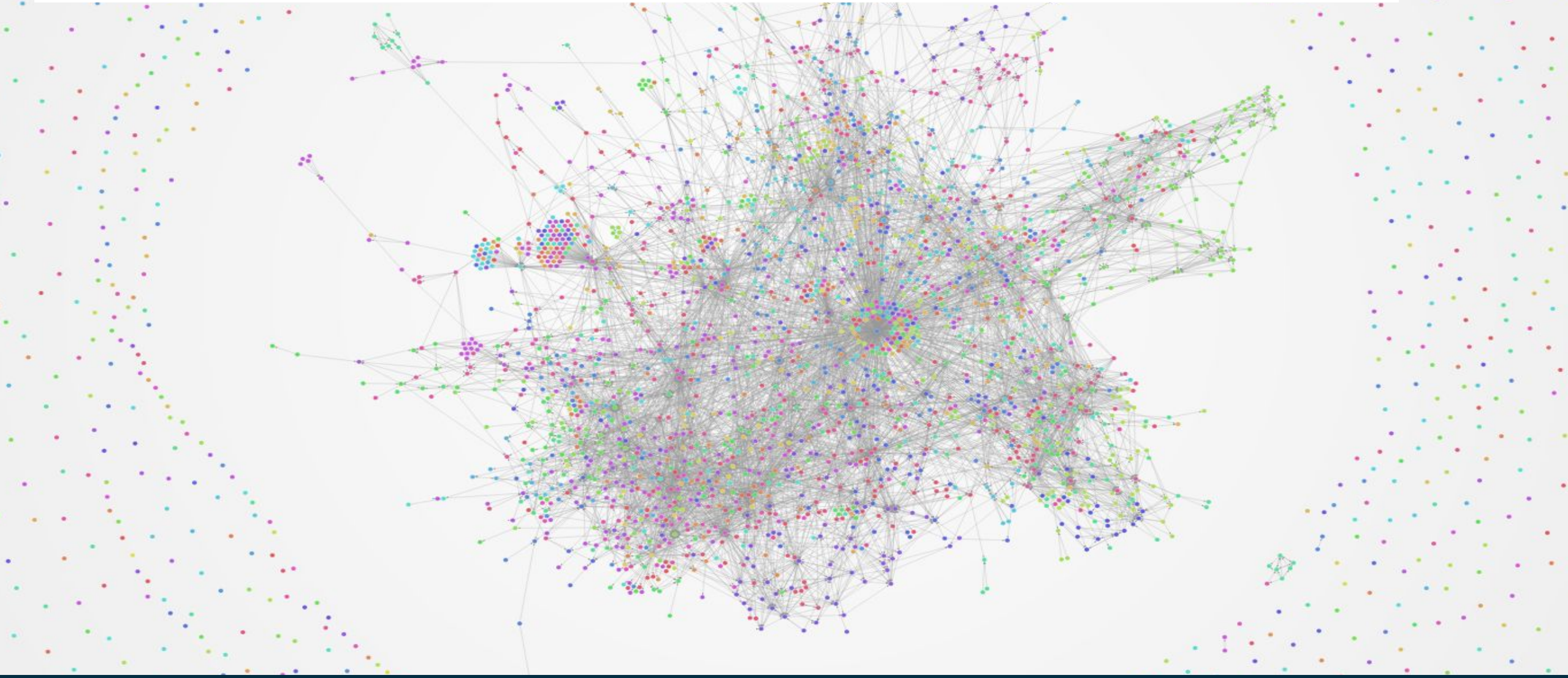
SLO Report the week after the Incident …

Place Order - Clients					
Redeem voucher / coupon				✅	
Remove article from cart				✅	
Request article invitation				✅	
Return articles				✅	
Select address				✅	
Select delivery option				✅	
Send Order Confirmation Email				✅	
Show order history				✅	
Sync Gift Card selection with payments				✅	
View cart				✅	
View checkout address				✅	
View checkout confirmation				✅	
View checkout payment option				✅	
View Hyped articles checkout address				✅	
View Hyped Articles checkout confirmation				✅	
View Order Confirmation				✅	

Incident Story

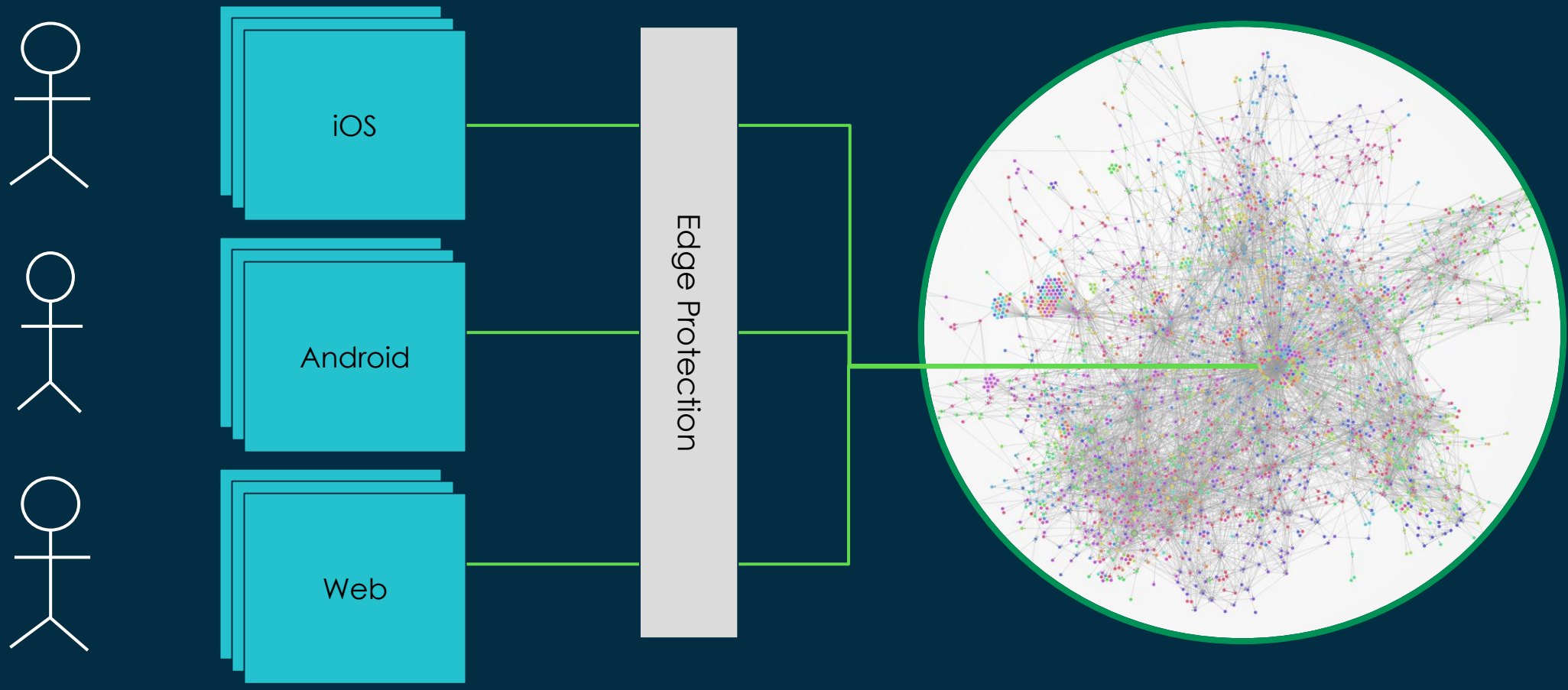
The lurking Add-To-Cart Failure

Is this even the right system to look at?



Zalando Servie Graph - 2019

To Protect UX, we have to look at the full system!



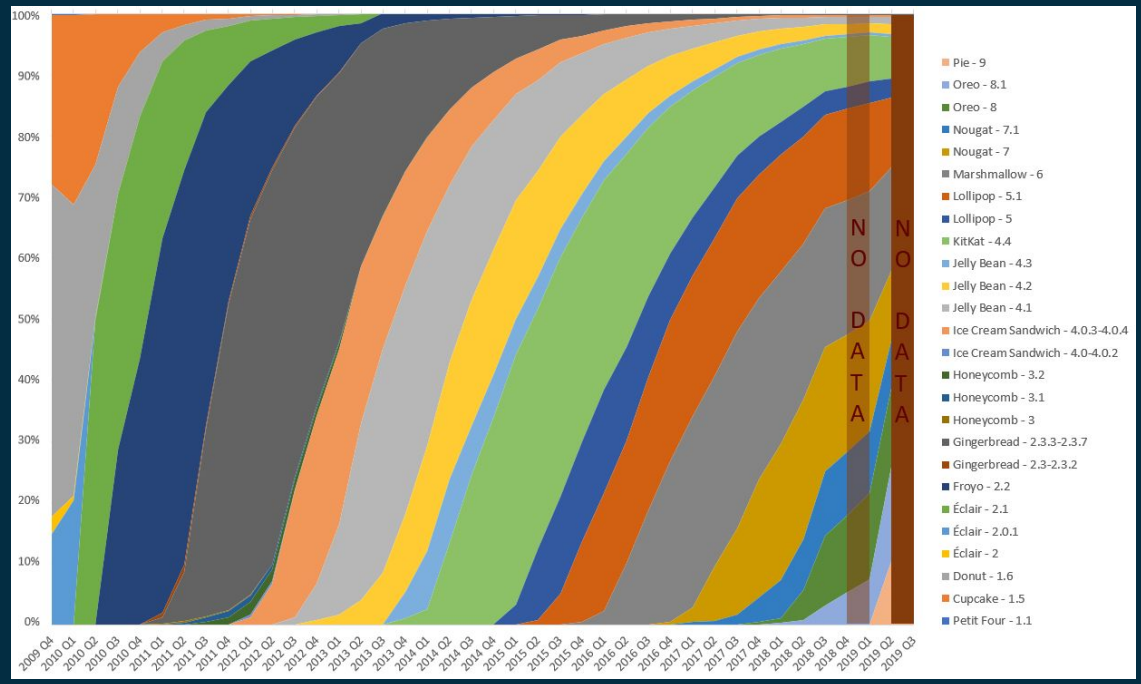
50 Million devices

3K Microservices

Challenges in Mobile Environment

1. **GLACIAL** deployment speed
   - One release every 2 weeks
   - 4 weeks to reach 80% penetration
2. Fragmented Platforms
3. Cellular Networks
4. Legal constraints
5. Automated UI Testing is HARD (i.e. missing)
6. Available Telemetry Data very limited

=> No DevOps culture in Mobile teams



Android versions deployed in the field (src: wikipedia)

Where are we investing?

* Distributed Tracing on Mobile + Web Clients

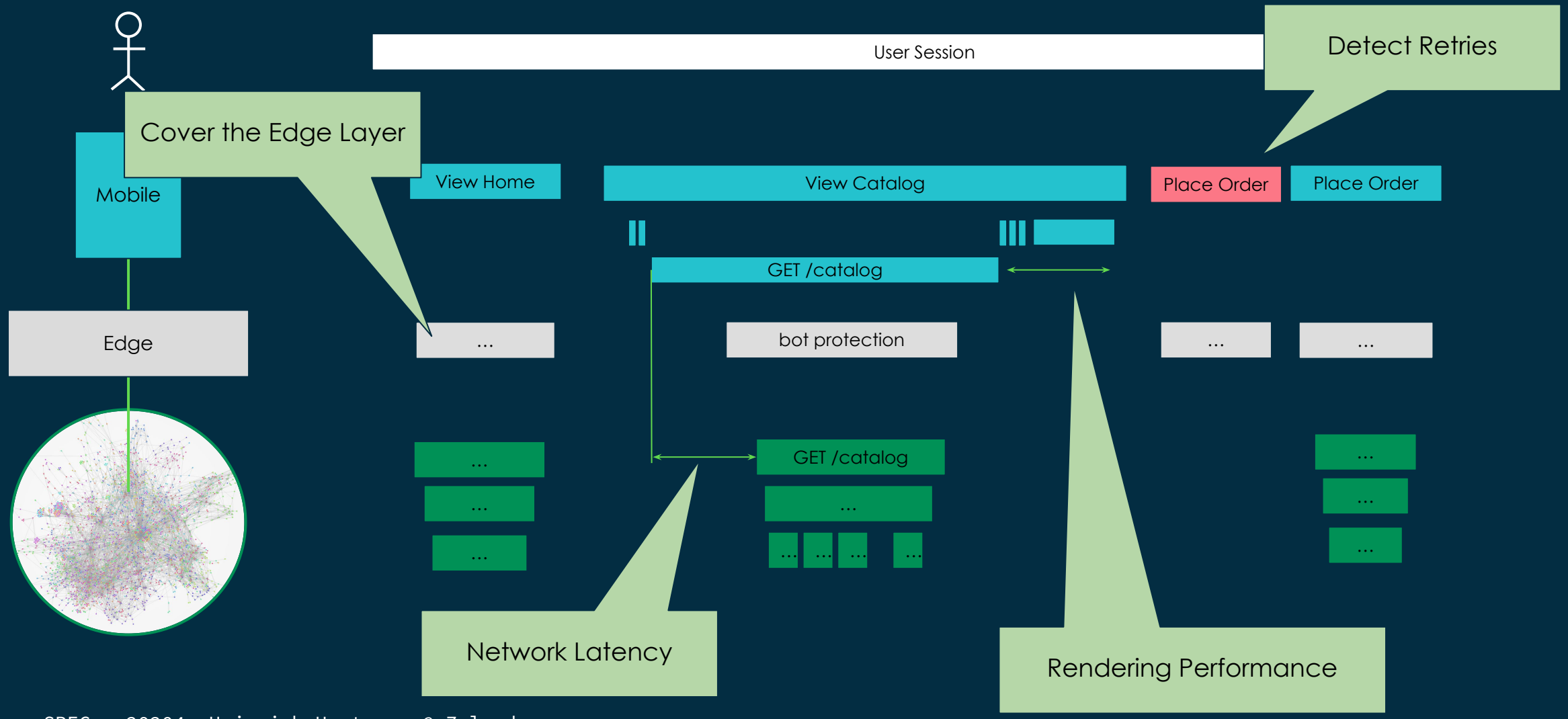
- Observability SDKs for Client Platforms
  - Browser (done)
  - Mobile (WIP) iOS / Android

- Client Side SLOs

  … complementing server-side measurements.

Goal: Expand Distributed Tracing to the Client!

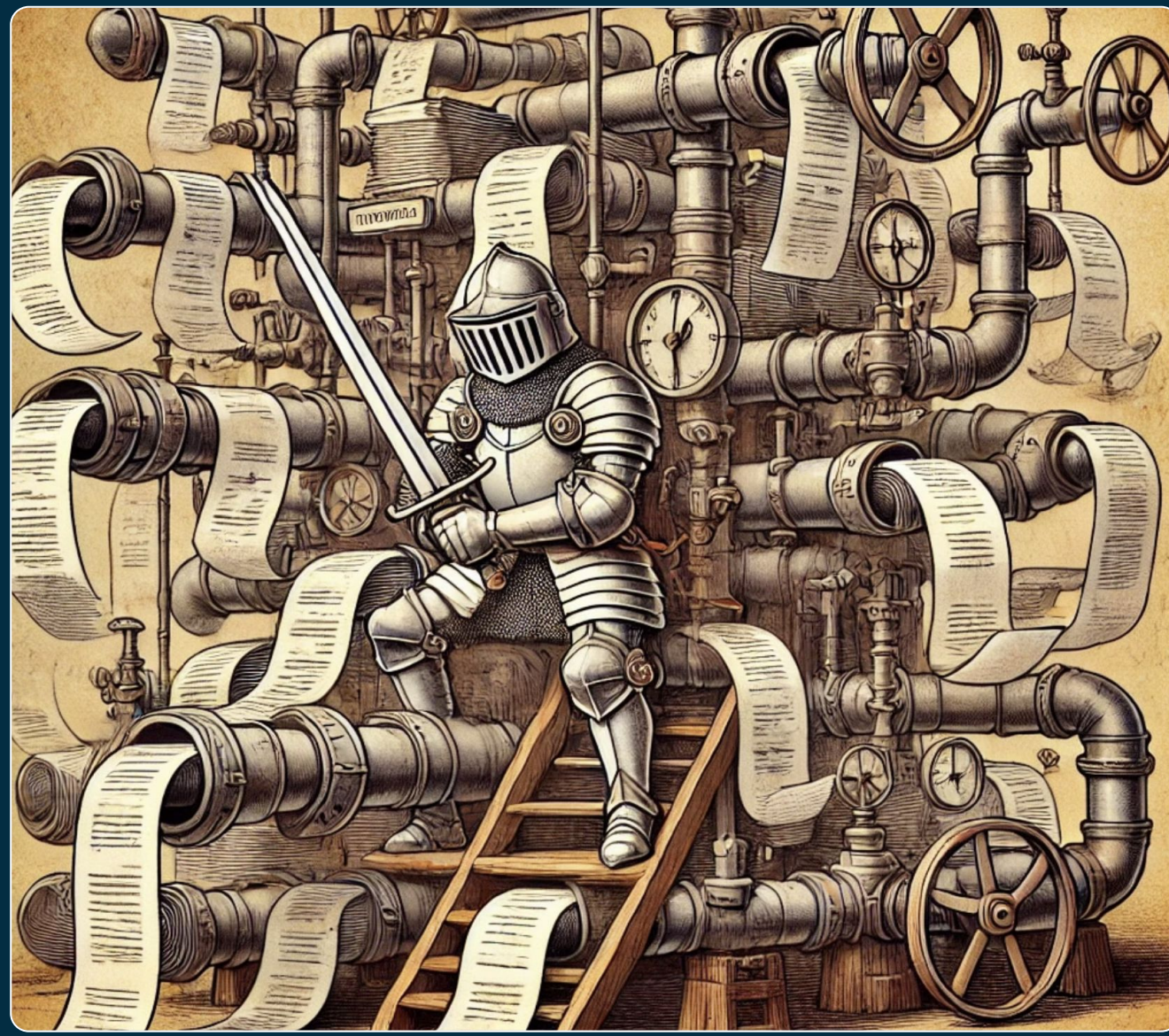
Goal: Expand Distributed Tracing to the Client!



Benefits from full Mobile Tracing Coverage

- Detect issues on the Client + Edge Layers
- Understand #impacted users
- Understand Retry behavior of users
- Understand Network latency
- Understand UI Performance
- Understand degradations of Business KPIs
- Detect general UX degradations

Data Operations



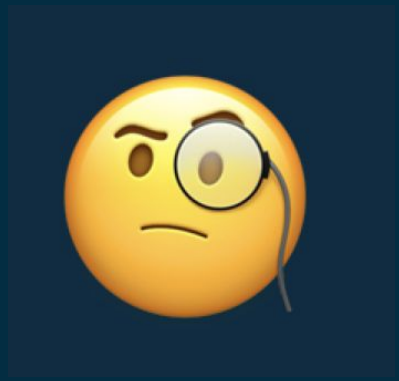
The “EURO” Incident in 2022

The Culprit

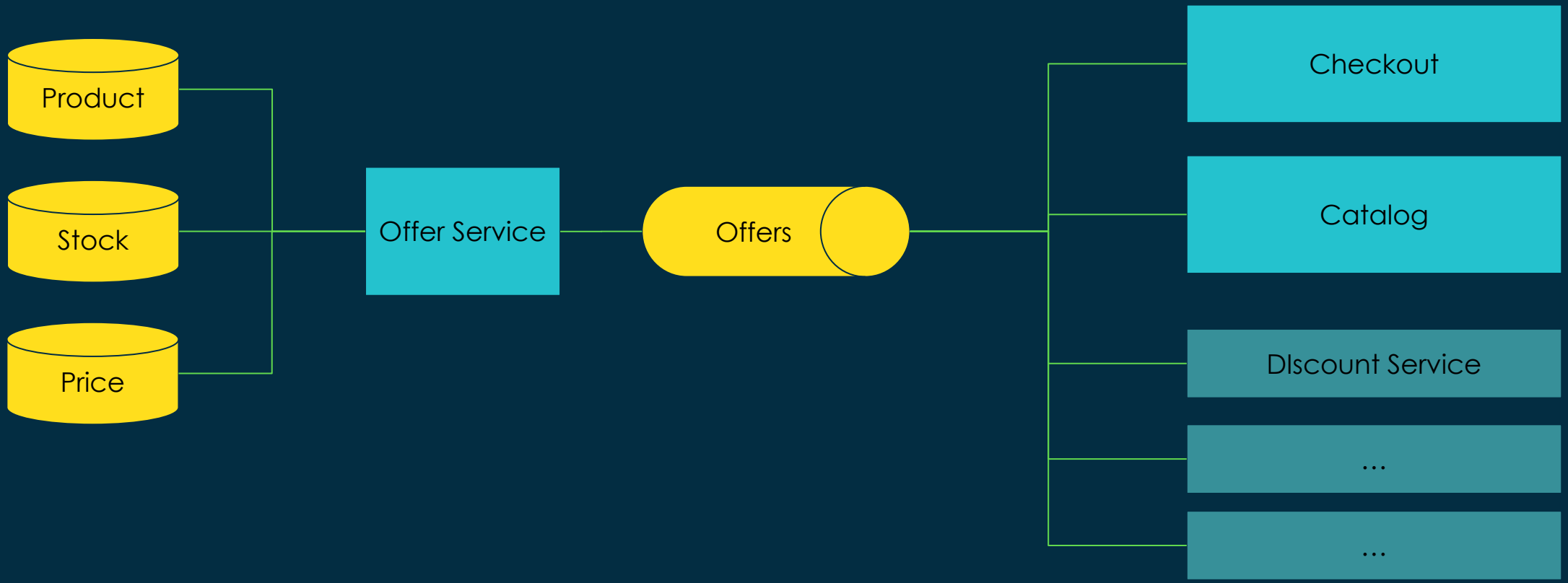
```
{
  "offer_id": "OF12345678",
  "product": {
      "product_id": "PR98765432",
      "name": "Men's Classic Leather Jacket",
      "category": "Men's Clothing > Jackets",
      "brand": "UrbanStyle",
      "description": "A premium classic leather jacket for men”
      "material": "Leather",
      "color": "Black",
      "size": ["S", "M", "L", "XL"],
      "images": [
      "https://alando.com/images/products/PR98765432_1.jpg",
      "https://alando.com/images/products/PR98765432_2.jpg"
      ],
      "tags": ["leather", "men's fashion", "jackets", "                style",
"winter"]
  },
  "price": {

      "currency": "EURO",
      "current_price": 129.99,
      "original_price": 159.99,
      "discount": {
      "percentage": 18,
      "description": "Autumn Sale"
      }
  },
  "stock": {
      "available": true,
      "quantity": 45,
      "locations": [
```

This should be “EUR”.



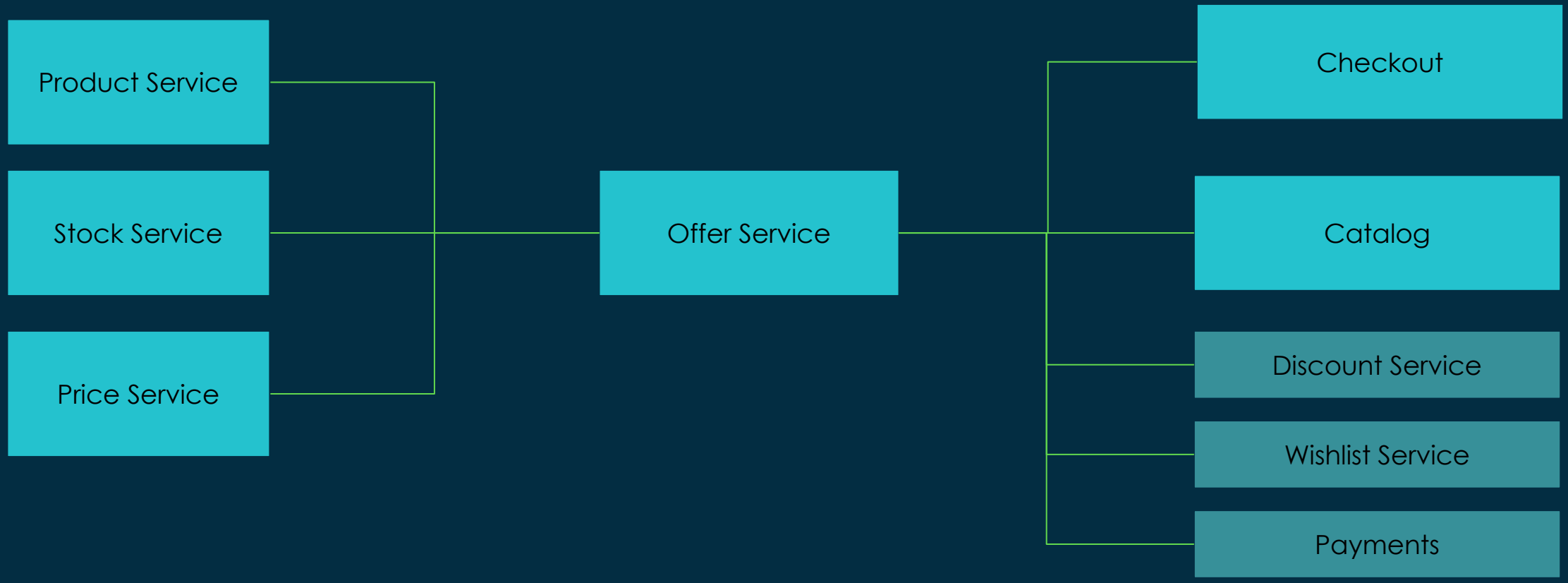
The EURO Incident - Data Architecture



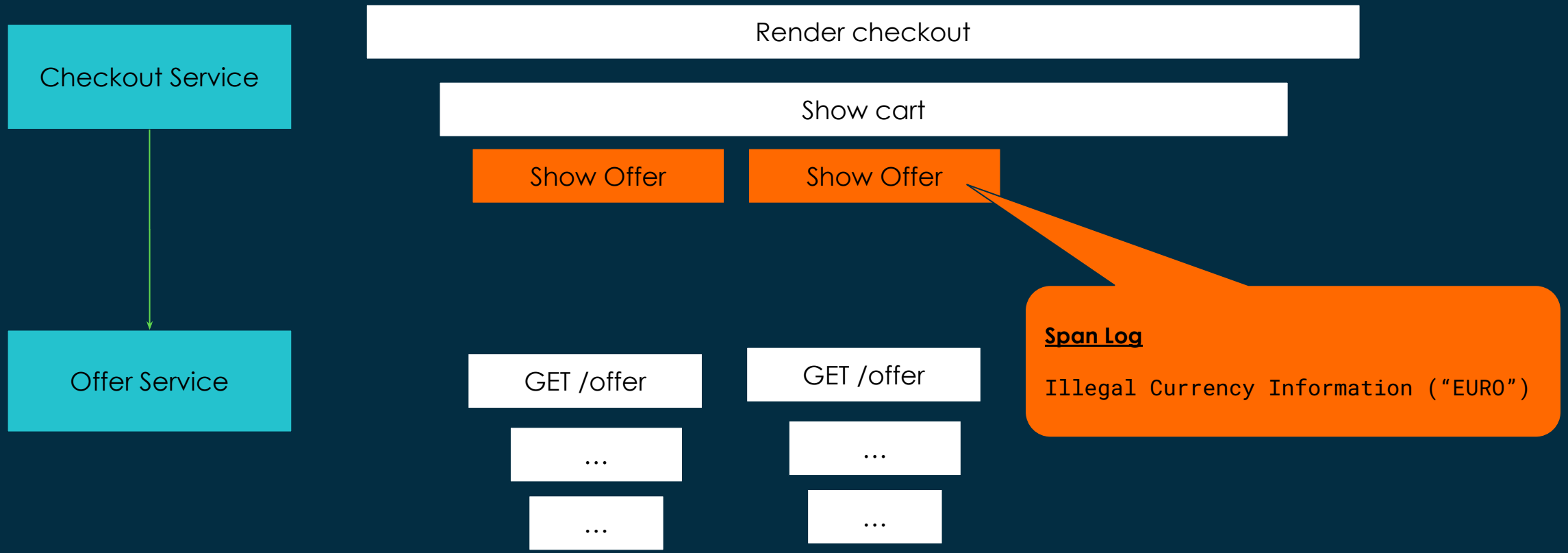
The EURO Incident - Take Aways

- Significant delay (30min) between cause (bad deployment) and impact.
- Significant delay (3h) between fix (rollback) and mitigation of symptoms.
- Limited help from Telemetry:
  - Metrics (Throughput, Backlog, Latency) - Not useful.
  - Tracing - Carried error information but no Causality Information

The “EURO Incident” with REST Architecture



Debuggin “EURO Incident” with Tracing



Data is of growing importance for …

1. AI
2. Business Processes
3. Business Intelligence

Patterns from ~70 Data Incidents in 2024

Backlogs / Delays / Capacity

Unclear Handover

Data Quality

Unstructured Data

Schema Changes

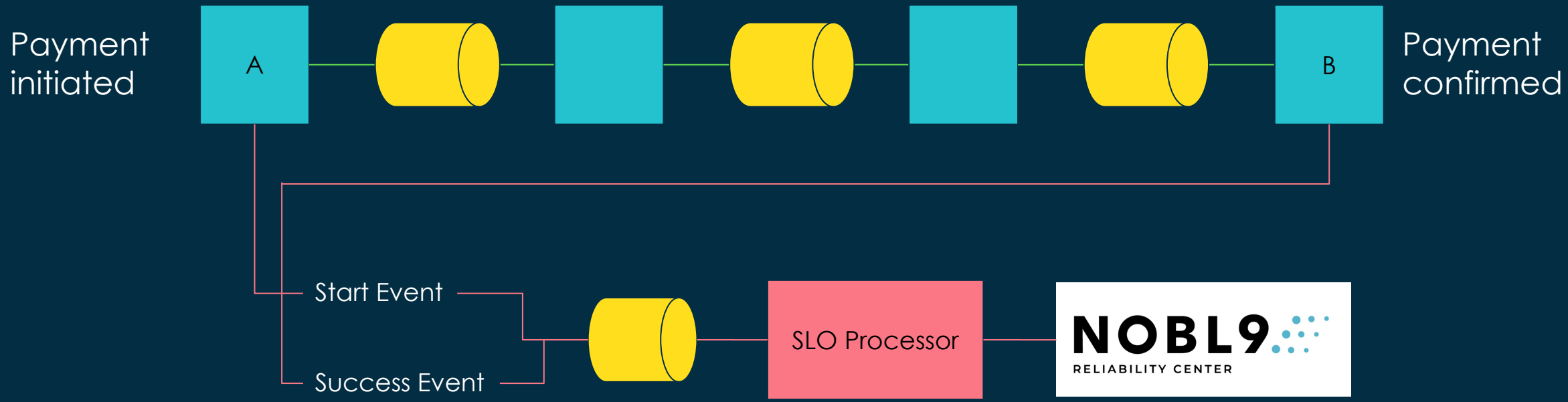
Long Data Chains

Unavailable Datasets

Integration w/ External Data Sources

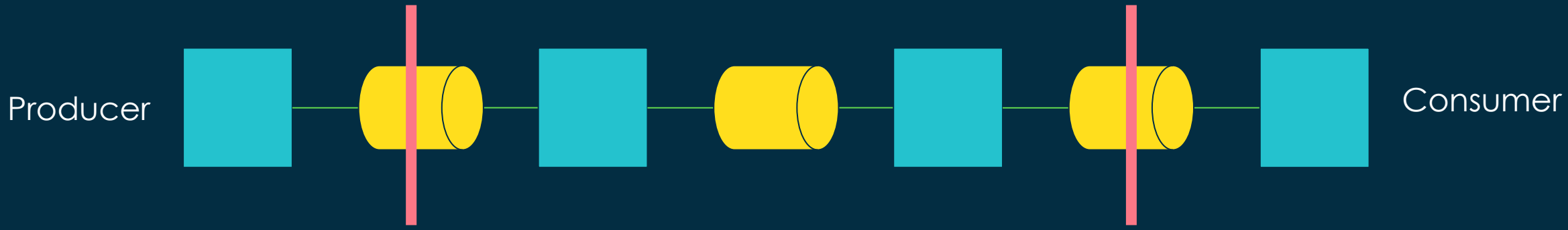
Investment Area - Data SLOs

- Check reliability of data processing pipelines end-2-end
- Check availability of datasets



Investment Area - Data Contracts

- Allow consumers to articulate expectations to data
- Detect Data Quality Problems earlier in the chain



UPRIVER

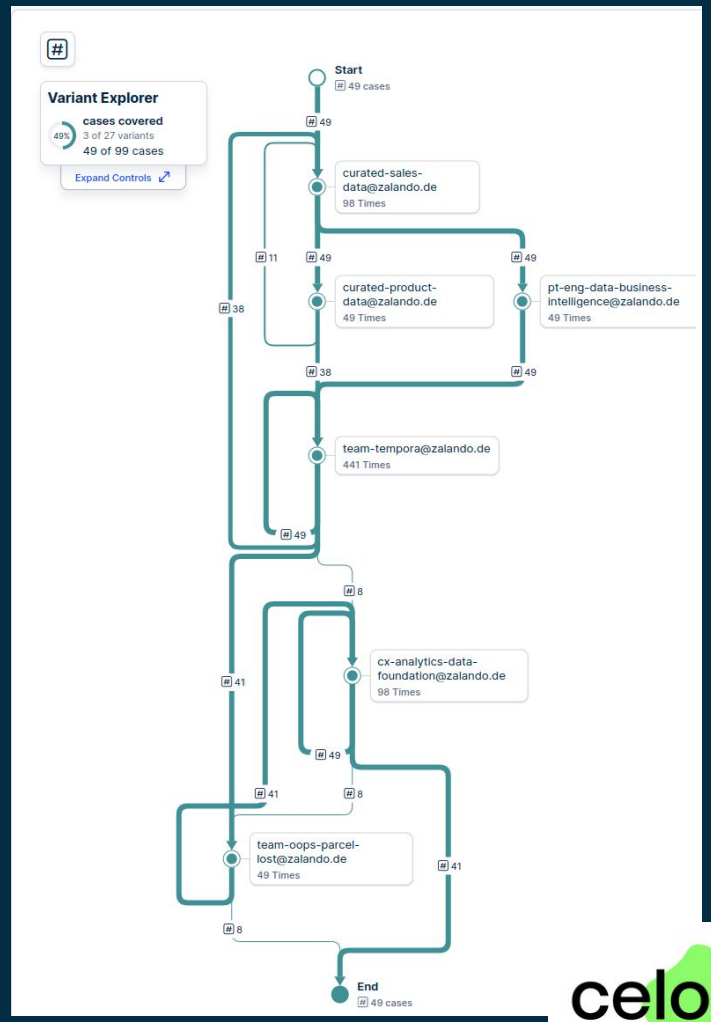
Anomalo

Investment Area - Data Lineage

- Map depenency chain of data products
- Upstream - Who produced/processed this data?
- Downstream - Who is depending on my data?

OpenLineage

AN OPEN FRAMEWORK FOR DATA LINEAGE COLLECTION AND ANALYSIS



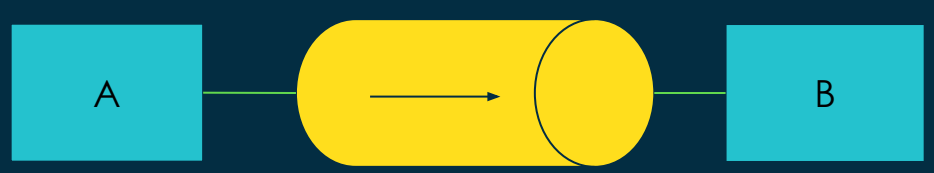
Experiment:

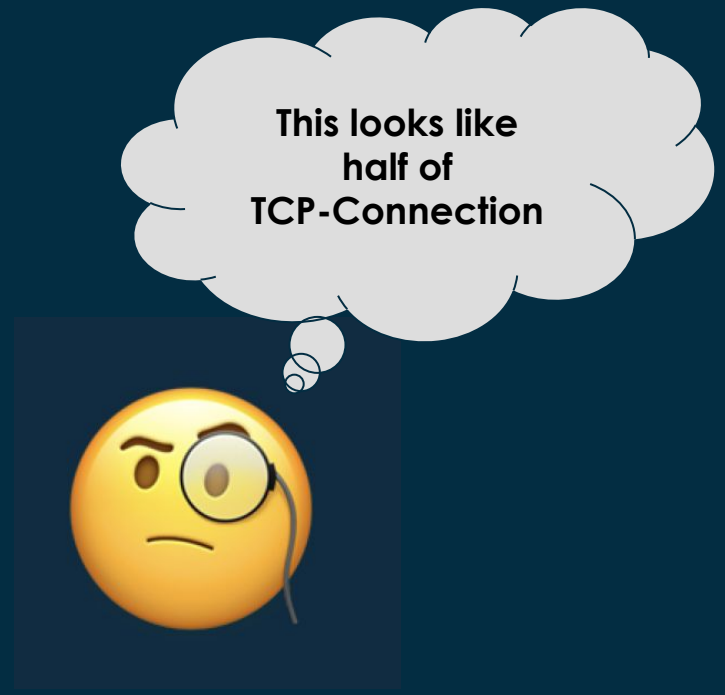
- Monitor timeliness of batch delivery for Business Analytics.
- Leverage Process Minig tool - Celonis.

celonis

Question: Do we have the right abstractions?

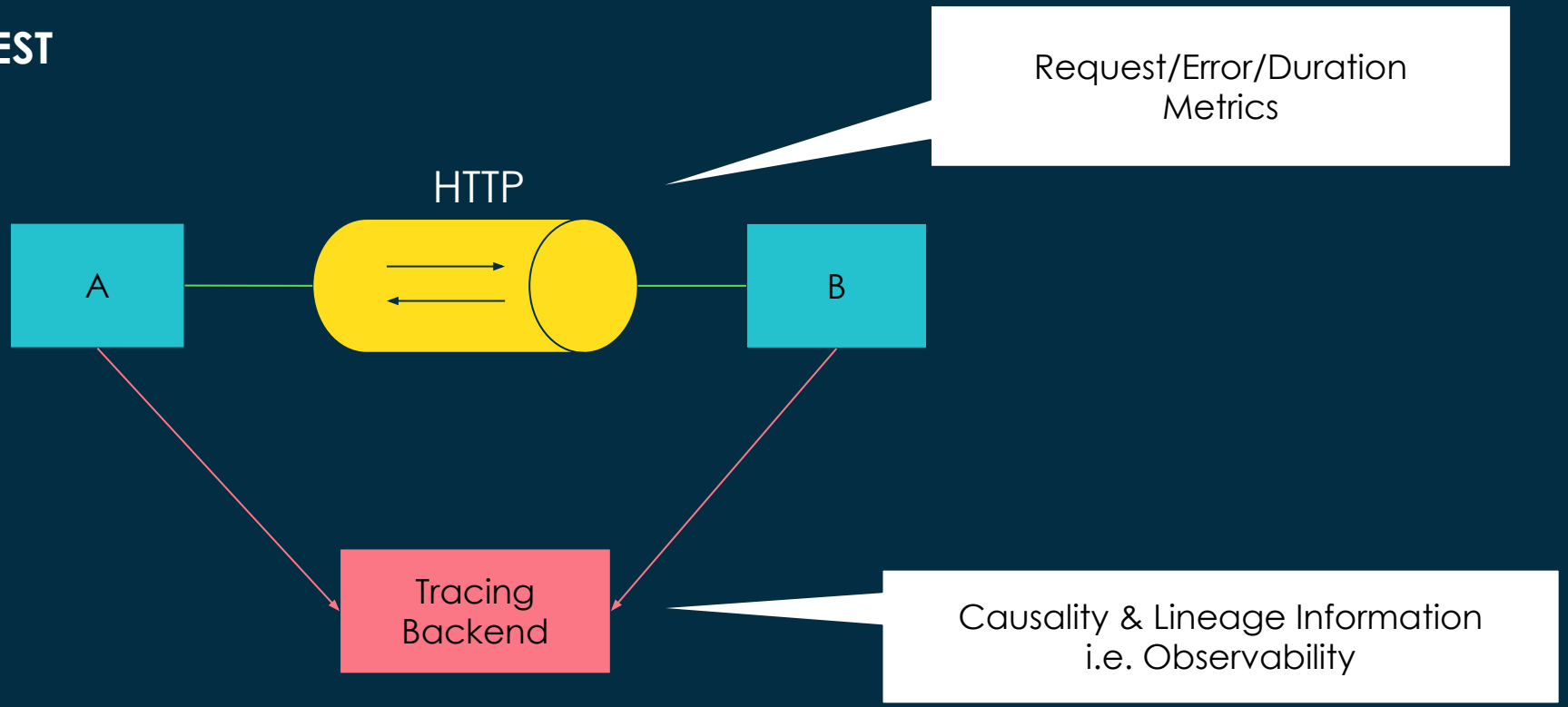
Data System Integration





Question: Do we have the right abstractions?

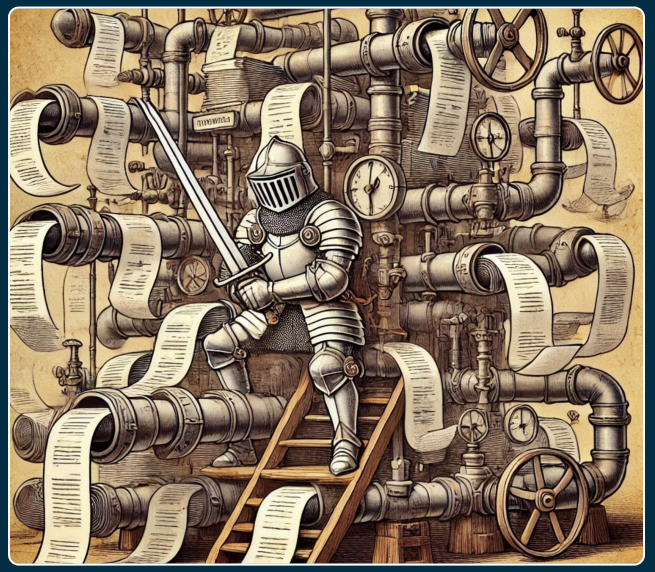
Microservice Integration / REST



Three Frontiers of Reliability Engineering

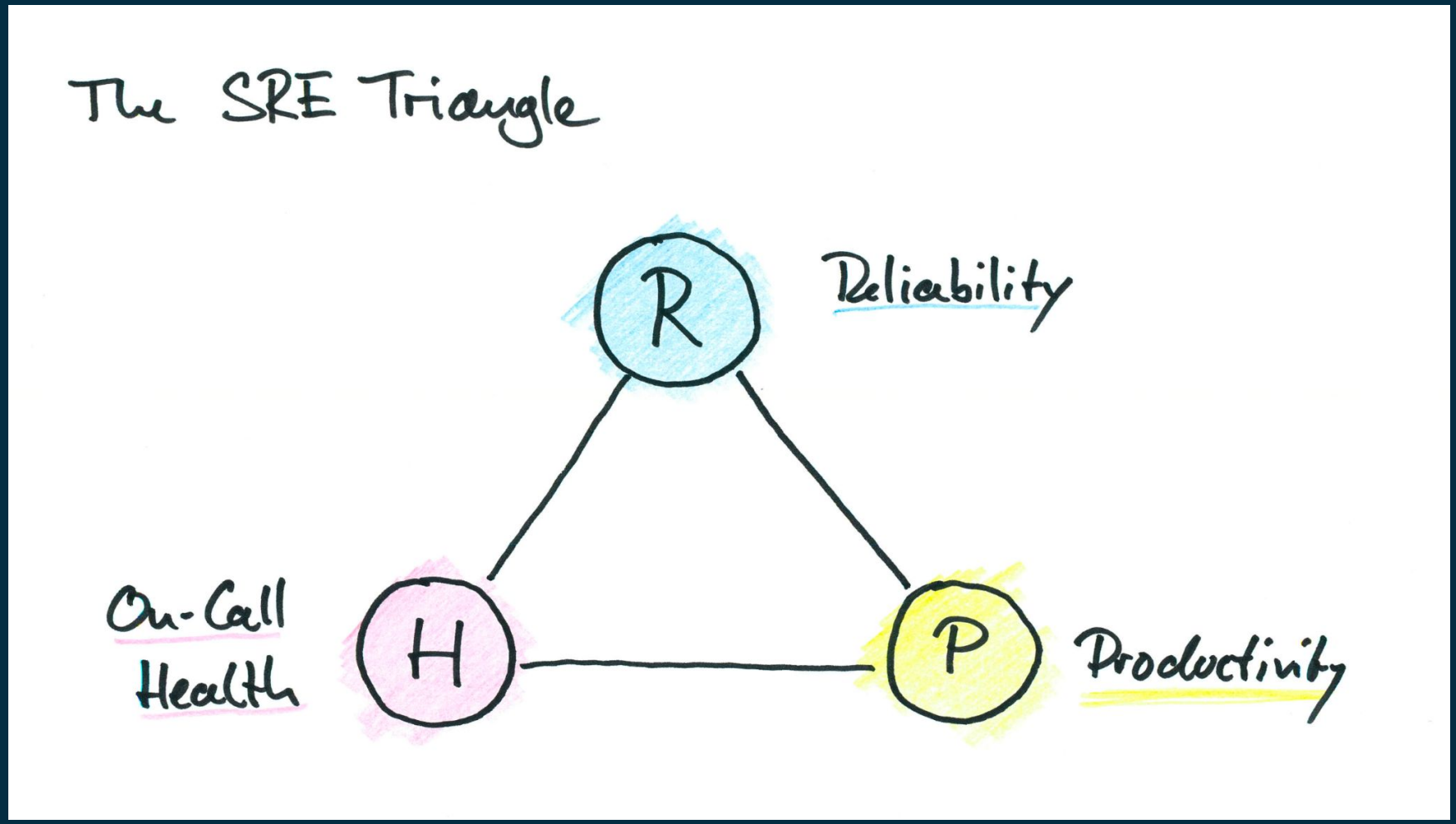
Managing for Reliability

Mobile Observability



Data Operations

> Heinrich Hartmann @ LinkedIn
#Let's talk Reliability! 💚


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