Out of Sight, Out of Control

Why Observability Matters

JUG Saxony 26.02.2026 | Nora Schöner | wolkencode.de

About me

AWS HEROES

aws User Group Leader

AWS Hero

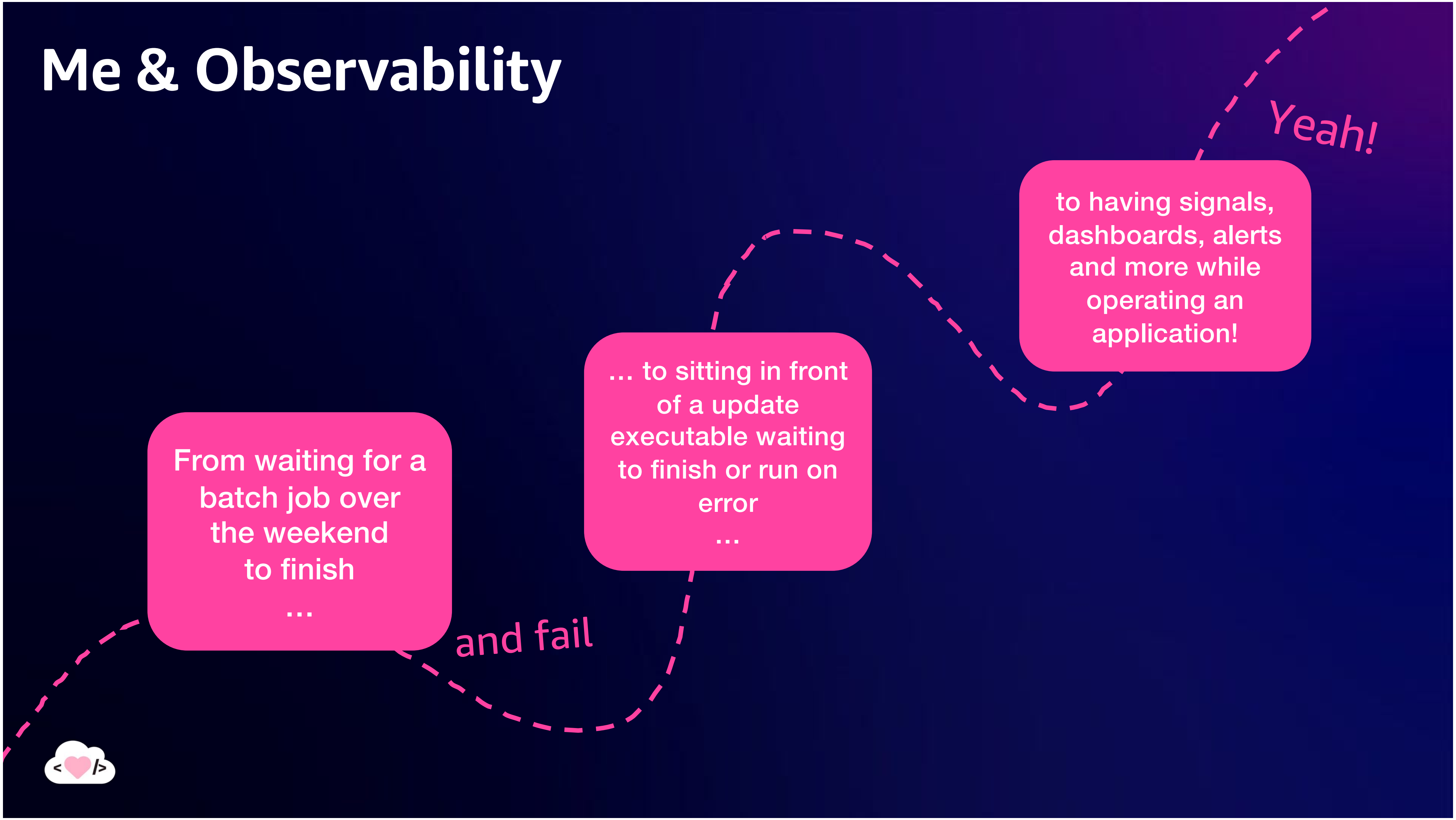
& AWS User Group Leader for Nürnberg

Senior 🎙️👩‍💻
Cloud Engineer &
Solutions Architect

graduated
in Maths 🧮

currently
obsession:
crocheting

Me & Observability



From waiting for a batch job over the weekend to finish
...

and fail

... to sitting in front of a update executable waiting to finish or run on error
...

to having signals, dashboards, alerts and more while operating an application!

Yeah!

What we’ll cover today

1️⃣ The Evolution And The Basics

2️⃣ Common Pitfalls And How To Avoid

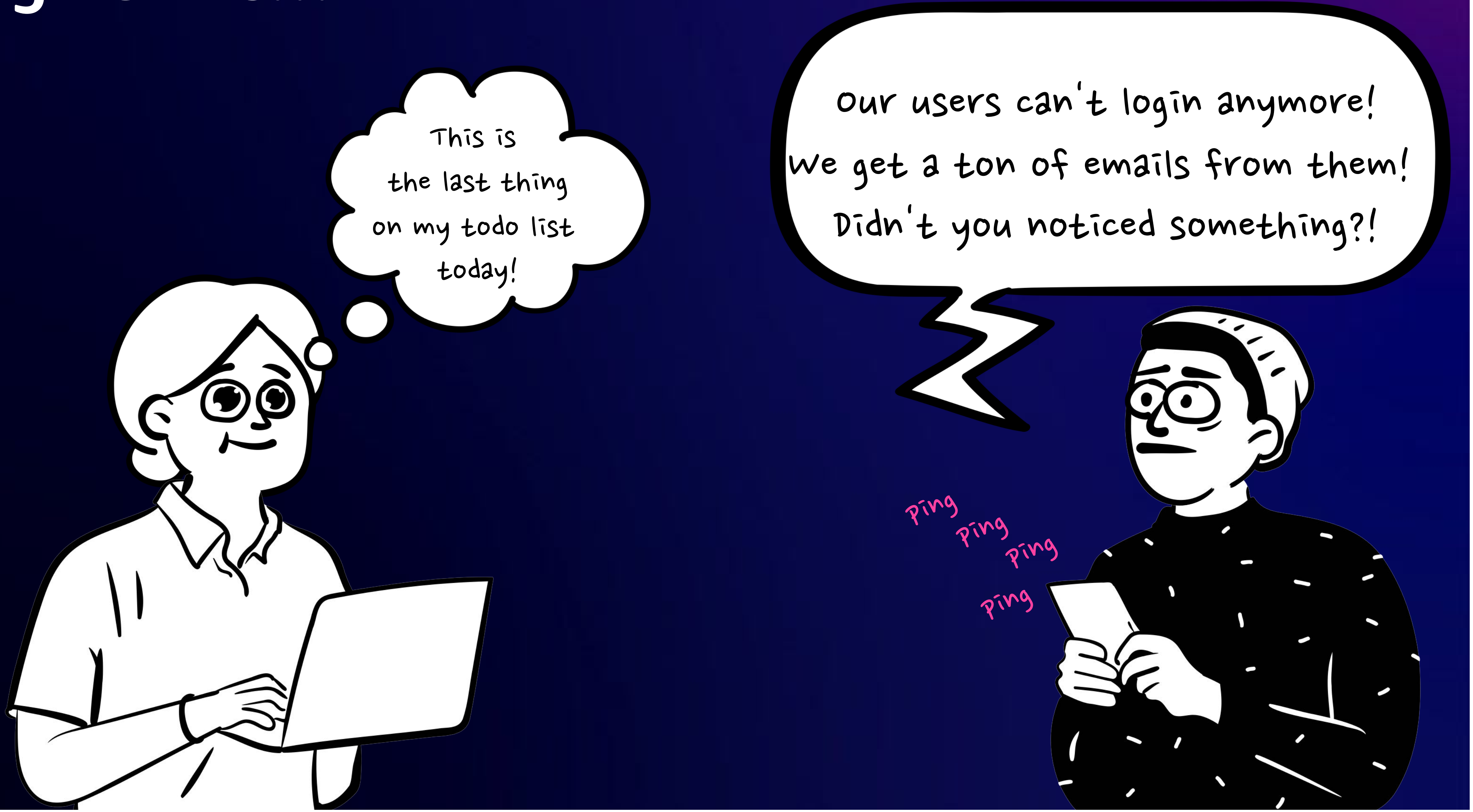
3️⃣ Practical Implementation Steps

4️⃣ Checking The O11y Maturity Level

🤖 How AI Can Support Us

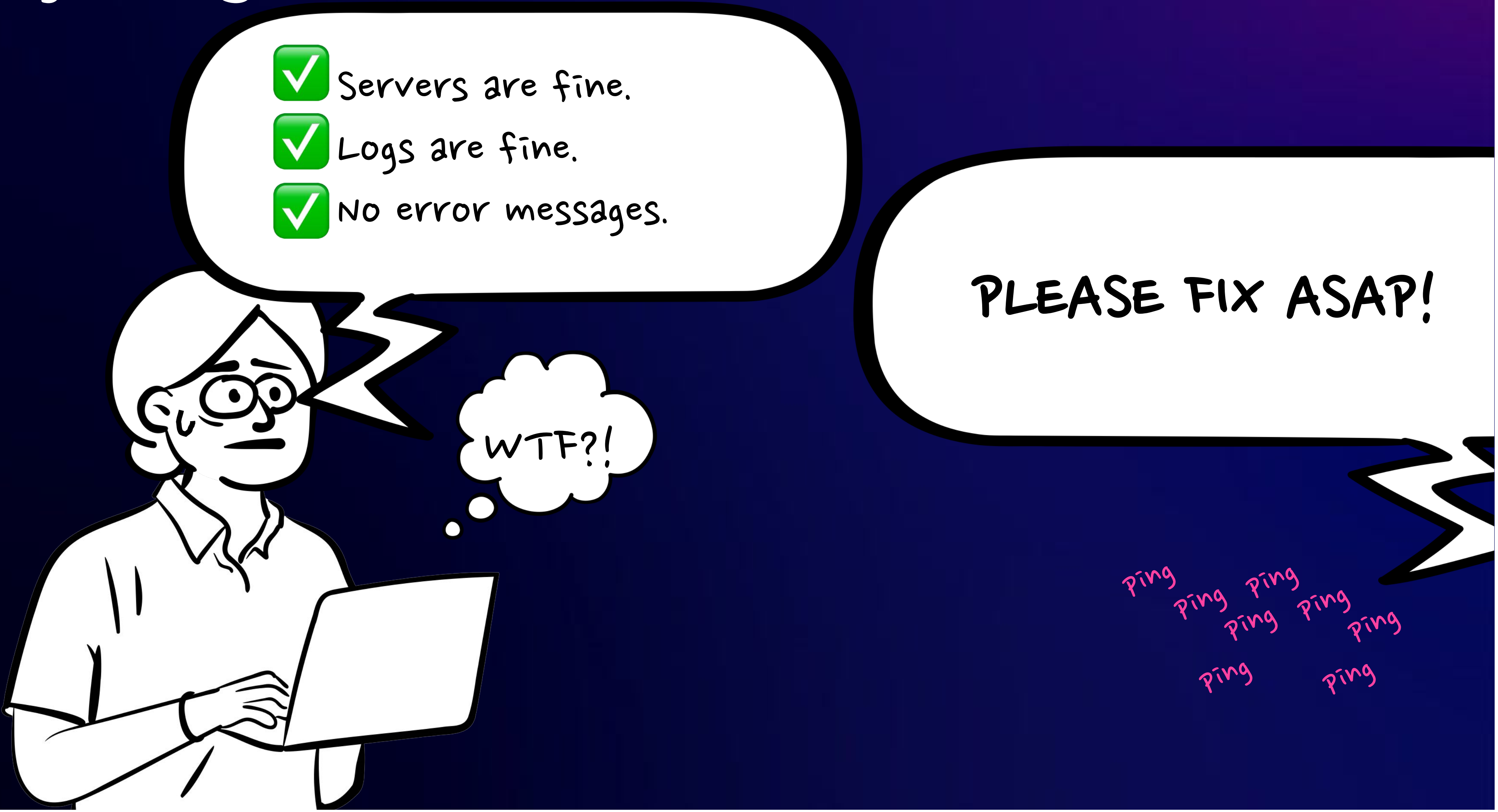
Hello, O11y!

Imagine this...


This is the last thing on my todo list today!
Our users can't login anymore! We get a ton of emails from them! Didn't you noticed something?!
Ping
Ping
Ping
Ping

Imagine this…

Everything seems fine…

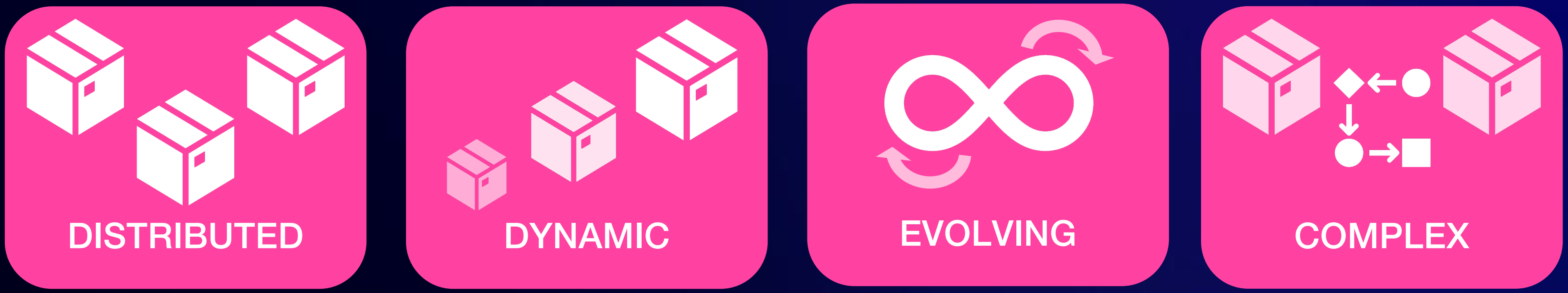


What actually happened?

- "What's ACTUALLY happening inside our system?"

The Modern Reality

Systems are more and more...

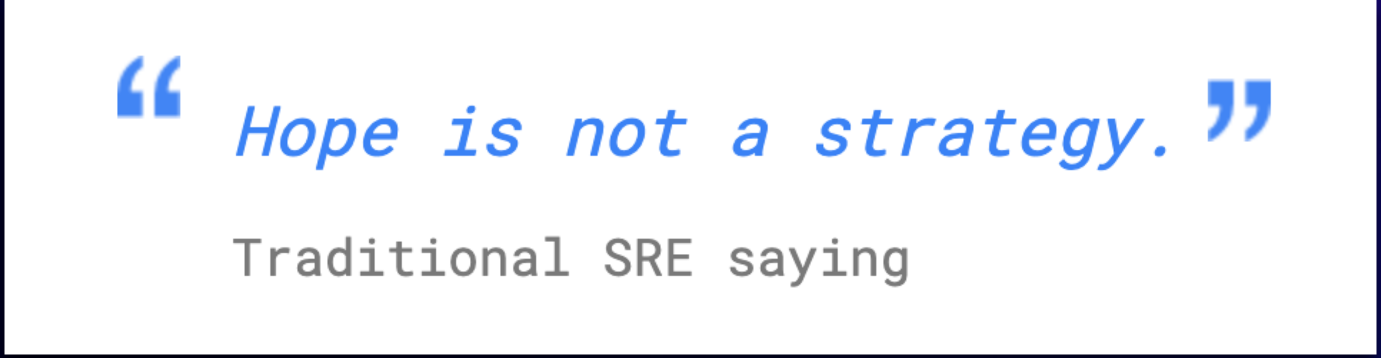


DISTRIBUTED

DYNAMIC

EVOLVING

COMPLEX

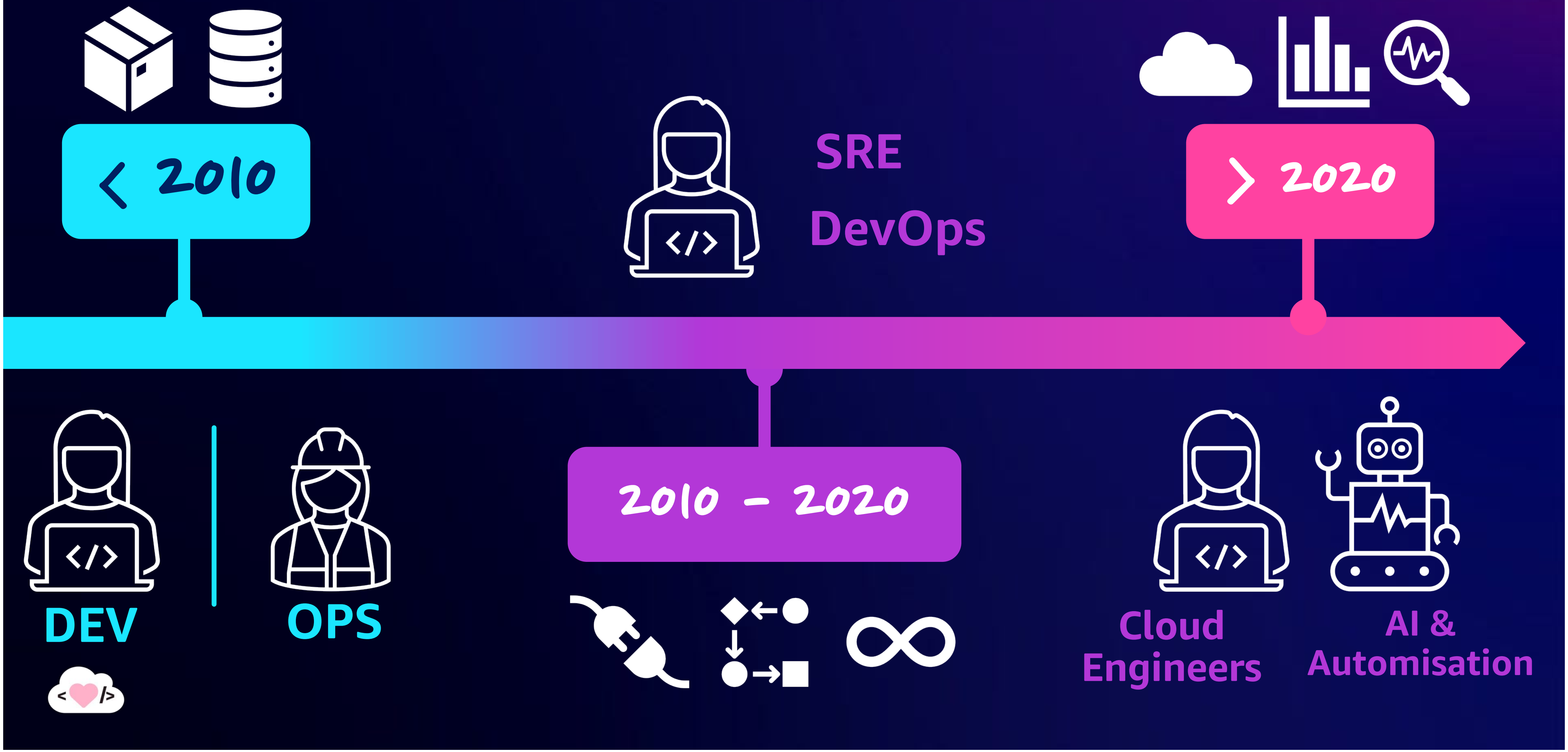


“*Hope is not a strategy.*”

Traditional SRE saying

screenshot taken from https://sre.google/sre-book/introduction/

The evolution from monitoring to observability: A brief timeline



< 2010

DEV

OPS

2010 – 2020

SRE

DevOps

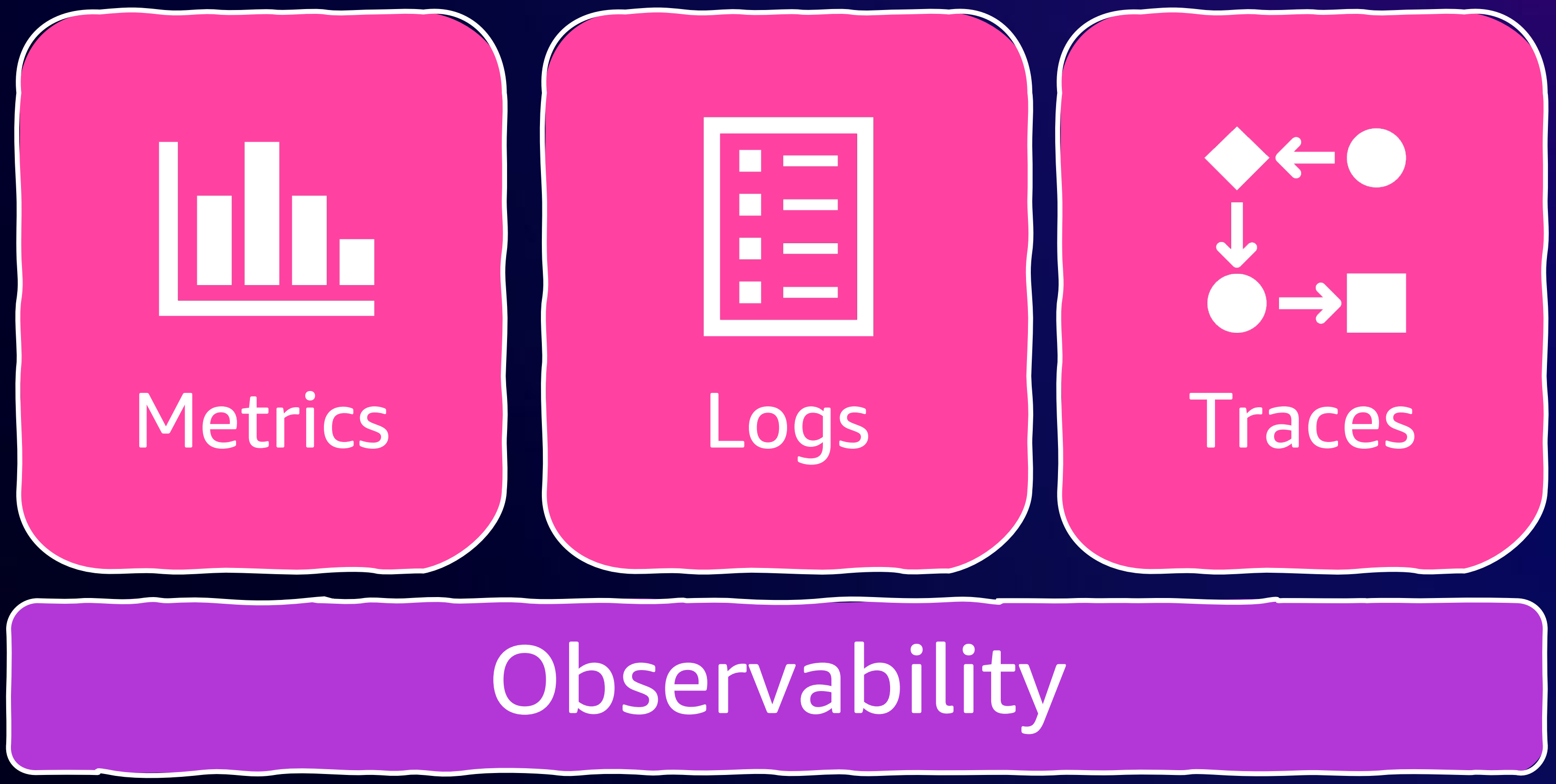
> 2020

Cloud Engineers

AI & Automisation

Three Signals

The Three Signals: An Overview



Metrics

Logs

Traces

Observability

The Three Signals: Metrics

Metrics

"What's the pattern?" – times-series data

- an error rate
- CPU utilization
- response time value

Common Tools, e.g. Prometheus, OpenTelemetry, Beats, …

The Three Signals: Logs

Logs

"What happend in detail?" – verbose data

- error messages
- request context
- user action

Common Tools, e.g. Loki, ElasticSearch, OpenTelemetry, …

The Three Signals: Traces

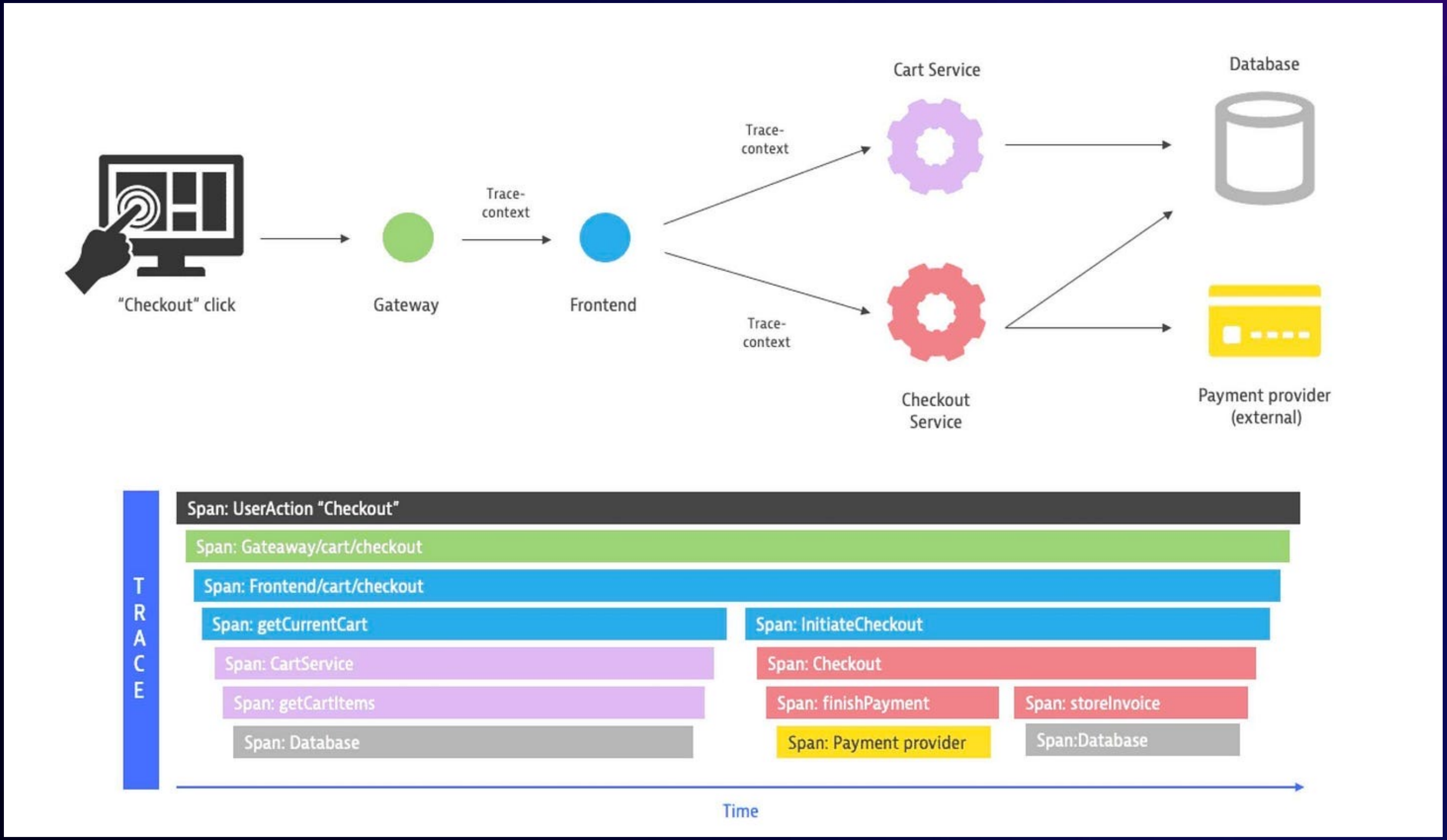
Traces

"Where did my request go?" – flow data

- grouped by transactions & spans
- show correlation between requests

Common Tools, e.g. Jaeger, Grafana Tempo, OpenTelemetry, …

The Three Signals :Traces & Spans



"Checkout" click

Gateway

Trace-context

Frontend

Trace-context

Trace-context

Cart Service

Checkout Service

Database

Payment provider (external)

T R A C E

Span: UserAction "Checkout"

Span: Gateaway/cart/checkout

Span: Frontend/cart/checkout

Span: getCurrentCart

Span: CartService

Span: getCartItems

Span: Database

Span: InitiateCheckout

Span: Checkout

Span: finishPayment

Span: Payment provider

Span: storeInvoice

Span:Database

Time

Traces

Spans

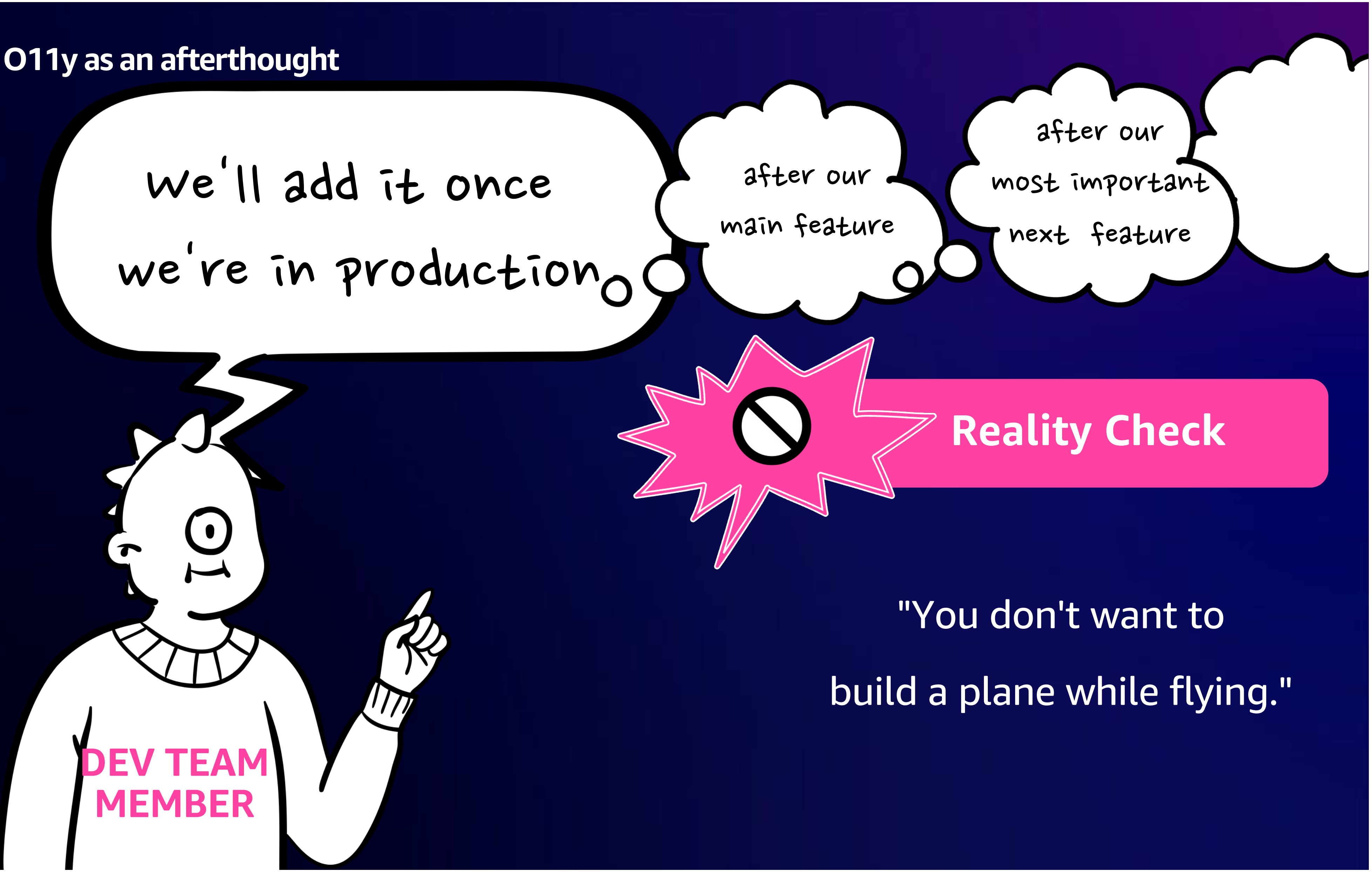
https://www.architecture-weekly.com/p/making-your-system-observability

COMMON PITFALLS

Pitfall #1

O11y as an Afterthought

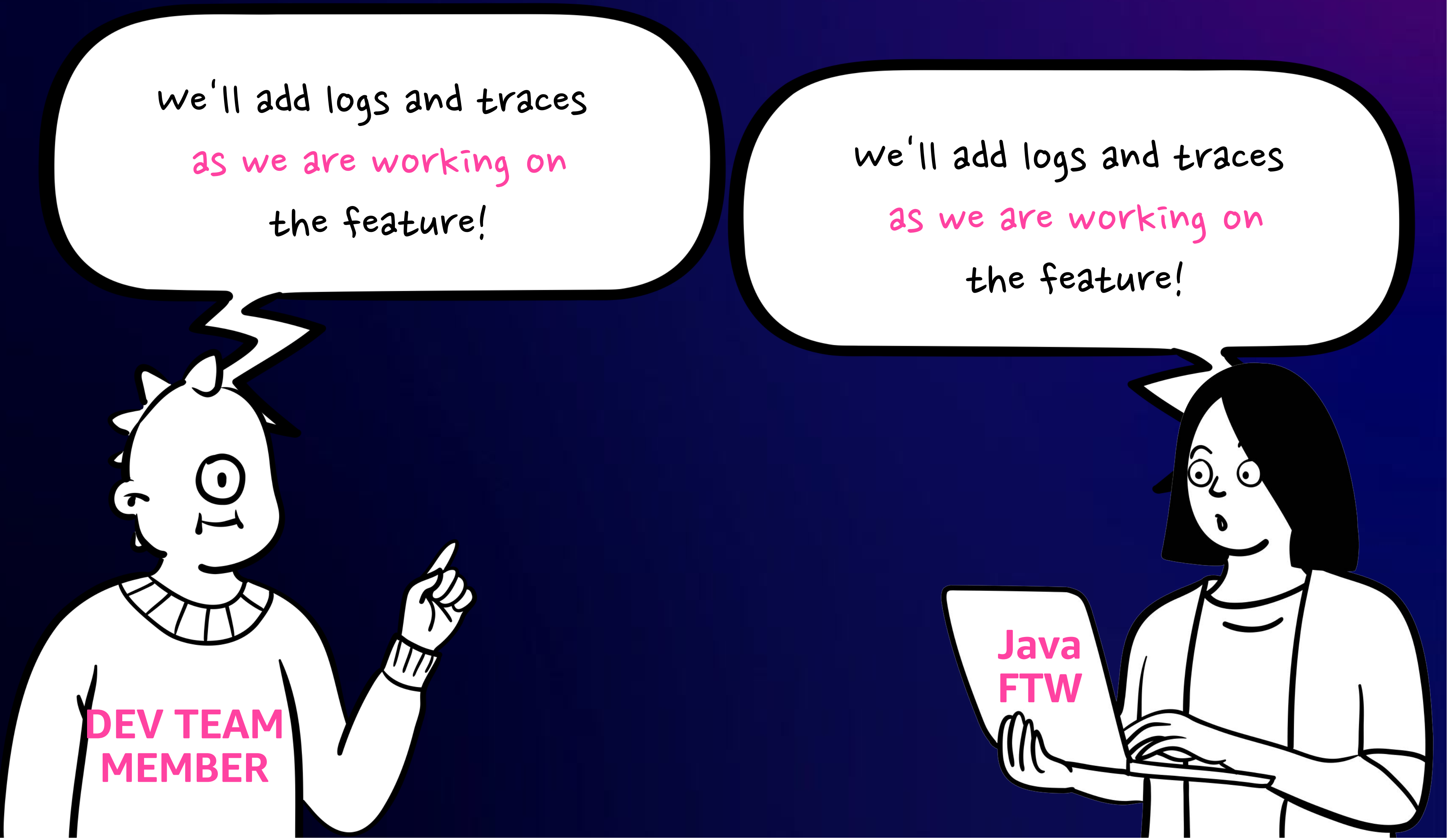
Pitfall #1: O11y as an afterthought

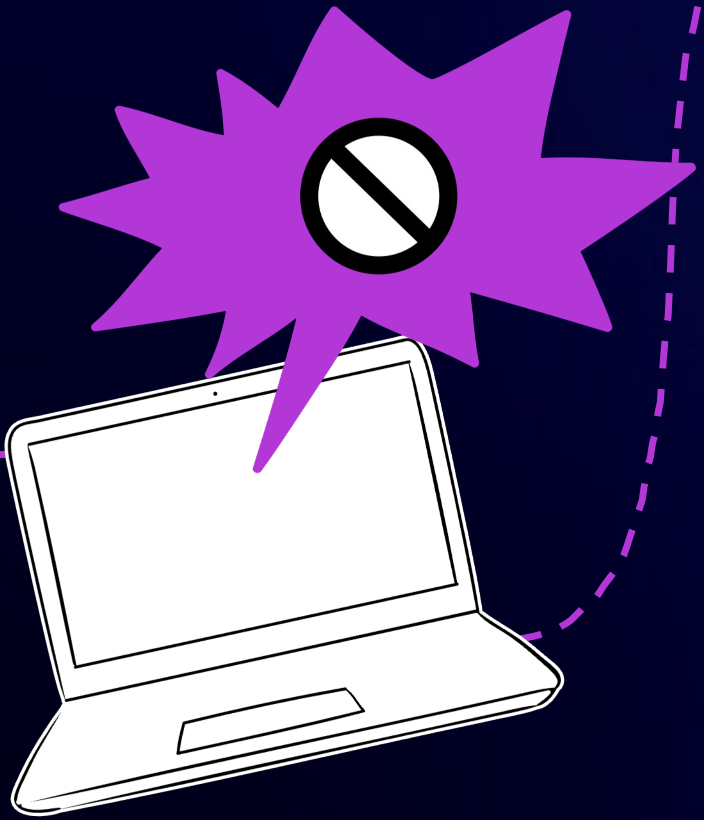

we'll add it once we're in production
after our main feature
after our most important next feature
DEV TEAM MEMBER

Reality Check

"You don't want to build a plane while flying."

Pitfall #1: O11y as an afterthought: The Lesson


we'll add logs and traces as we are working on the feature!
we'll add logs and traces as we are working on the feature!
DEV TEAM MEMBER
Java FTW



Pitfall #2

The Data Deluge

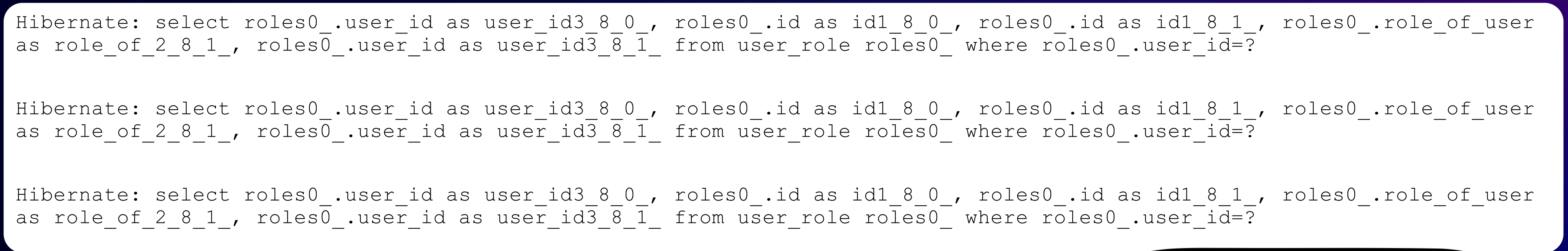
Pitfall #2: The Data Deluge

Log as many as possible, but when it's enough?

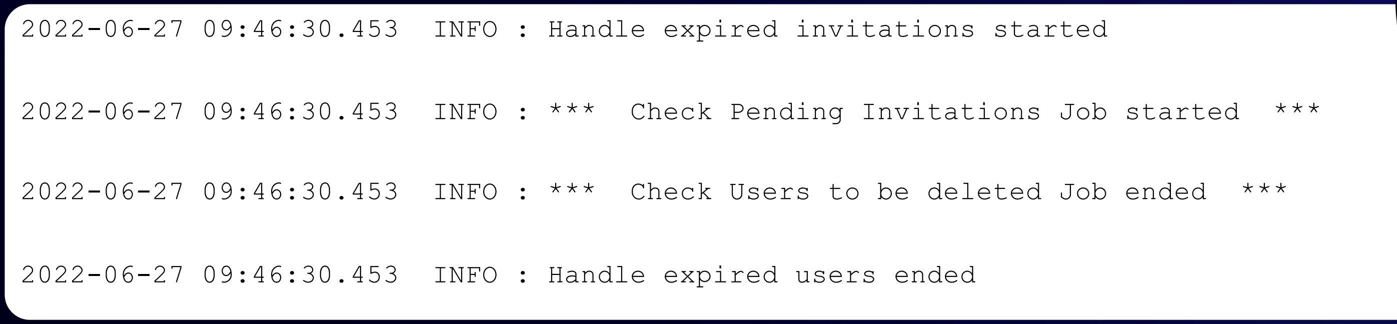
Pitfall #2: The Data Deluge: Don't stop logging

```
Starting XYZ app...


Spring profiles: dev





2021-09-10T16:21:29,679 [main] INFO  c.c.xyz.BusinessRestApplication - Starting BusinessRestApplication using Java
11.0.11 on pbf-1875-xyz-ifb-be-business-6f8945875b-sq2hp with PID 1 (/app.jar started by root in /)


2021-09-10T16:21:30,363 [main] INFO  c.c.xyz.BusinessRestApplication - The following profiles are active: dev


2021-09-10T16:21:30,495 [main] WARN  o.s.b.c.c.ConfigDataEnvironment - Property 'spring.profiles' is invalid and should
be replaced with 'spring.config.activate.on-profile' [origin: System Environment Property "SPRING_PROFILES"]


2021-09-10T16:21:37,898 [main] WARN  i.o.c.s.t.c.TracerAutoConfiguration - Tracer bean is not configured! Switching to
NoopTracer


2021-09-10T16:21:54,212 [main] WARN  o.s.b.a.o.j.JpaBaseConfiguration$JpaWebConfiguration - spring.jpa.open-in-view is
enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure
spring.jpa.open-in-view to disable this warning


2021-09-10T16:22:01,392 [main] INFO  c.c.xyz.BusinessRestApplication - Started BusinessRestApplication in 33.468 seconds
(JVM running for 34.936)
```

Pitfall #2: The Data Deluge: Being precise with your logs



```
Hibernate: select roles0_.user_id as user_id3_8_0_, roles0_.id as id1_8_0_, roles0_.id as id1_8_1_, roles0_.role_of_user
as role_of_2_8_1_, roles0_.user_id as user_id3_8_1_ from user_role roles0_ where roles0_.user_id=?

Hibernate: select roles0_.user_id as user_id3_8_0_, roles0_.id as id1_8_0_, roles0_.id as id1_8_1_, roles0_.role_of_user
as role_of_2_8_1_, roles0_.user_id as user_id3_8_1_ from user_role roles0_ where roles0_.user_id=?

Hibernate: select roles0_.user_id as user_id3_8_0_, roles0_.id as id1_8_0_, roles0_.id as id1_8_1_, roles0_.role_of_user
as role_of_2_8_1_, roles0_.user_id as user_id3_8_1_ from user_role roles0_ where roles0_.user_id=?
```



```
2022-06-27 09:46:30.453  INFO : Handle expired invitations started

2022-06-27 09:46:30.453  INFO : ***  Check Pending Invitations Job started  ***

2022-06-27 09:46:30.453  INFO : ***  Check Users to be deleted Job ended  ***

2022-06-27 09:46:30.453  INFO : Handle expired users ended
```

Pitfall #2: The Data Deluge: Using correlation IDs

```
2022-06-27T10:13:53 - User Login called at /auth/user

2022-06-27T10:13:53 - User Login called at /auth/user

2022-06-27T10:13:53 - User login succeeded

2022-06-27T10:13:53 - User Login failed
```

```
2022-06-27T10:13:53 - User Login called at /auth/user (id: 3675f647-xx-17c13b67e08b)

2022-06-27T10:13:53 - User Login called at /auth/user (id: 81ebf877-xx-f20e6e00bc80)

2022-06-27T10:13:53 - User login succeeded (id: 81ebf877-xx-f20e6e00bc80)

2022-06-27T10:13:53 - User Login failed (id: 3675f647-xx-17c13b67e08b)
```

Pitfall #2: The Data Deluge: Just Nope

```
2022-06-27T10:13:53 - User Login called (username: karl password: geheim123!)
```

Pitfall #2: The Data Deluge: The Lesson

Log whats **important** for your application

Log what's **help** you **solving incidents**

Hide what shouldn't be logged

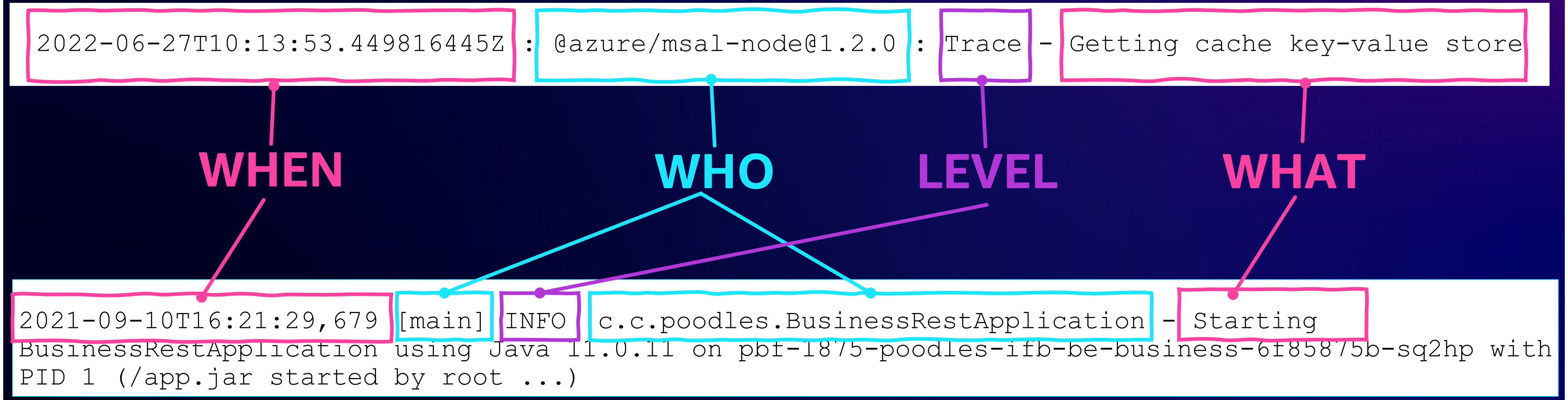
WHEN

WHO

WHAT

WHY

Pitfall #2: The Data Deluge: The Lesson



```
2022-06-27T10:13:53.449816445Z : @azure/msal-node@1.2.0 : Trace - Getting cache key-value store
```

```
2021-09-10T16:21:29,679 [main] INFO  c.c.poodles.BusinessRestApplication - Starting
BusinessRestApplication using Java 11.0.11 on pbf-1875-poodles-ifb-be-business-6f85875b-sq2hp with
PID 1 (/app.jar started by root ...)
```

```
ts="2021-09-10T16:21:29,679" package="main" level="INFO"
module="c.c.poodles.BusinessRestApplication" type="startup" java_version="11.0.11"
jar="app.jar" message="Starting BusinessRestApplication using Java 11.0.11 on pbf-1875-
poodles-ifb-be-business-6f8945875b-sq2hp with PID 1 (/app.jar started by root in /)"
```

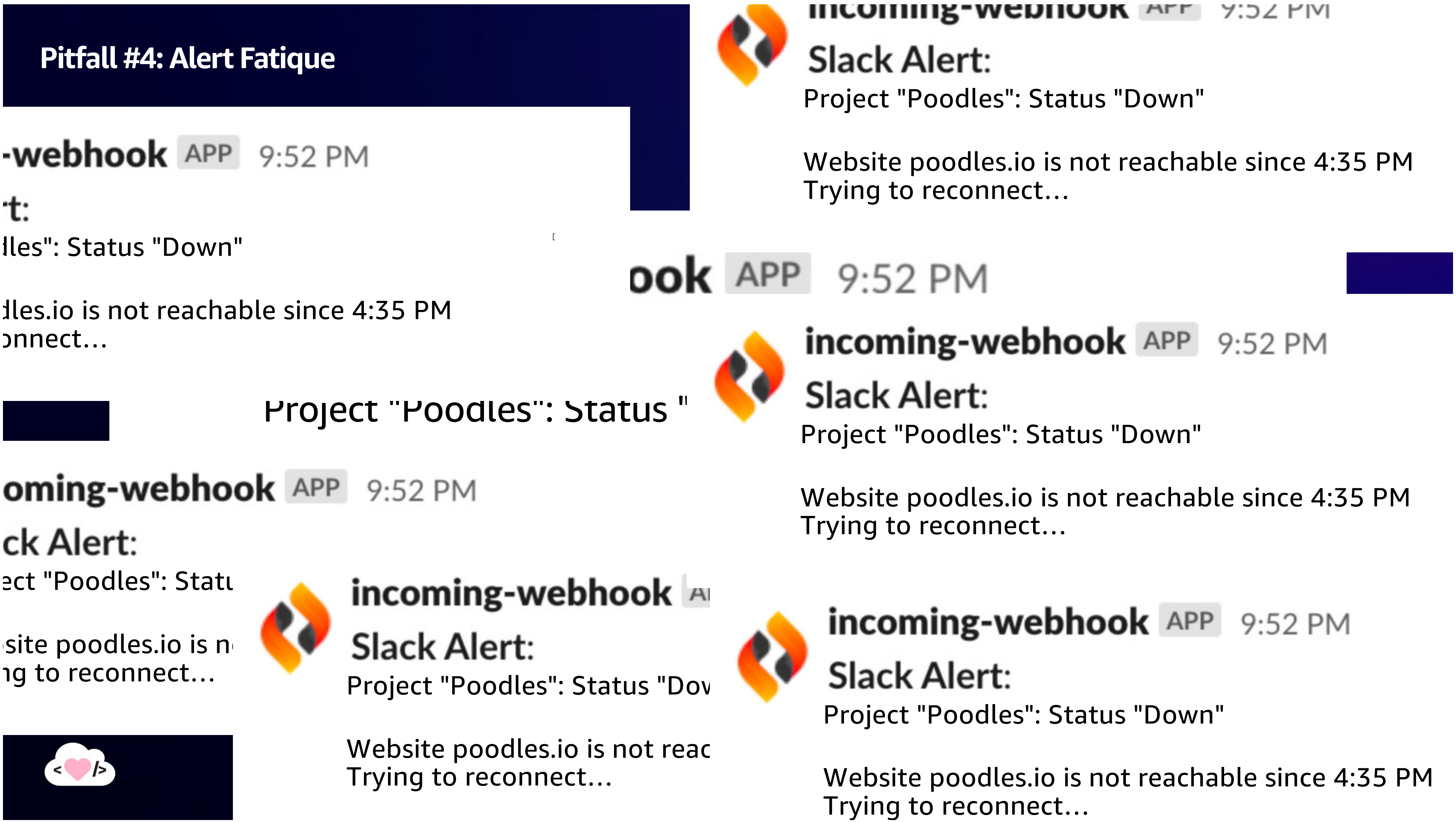
Pitfall #3

Alert Fatique

Pitfall #4: Alert Fatique

These are just those usual warnings ...
I'll clean up the alerts next sprint

Pitfall #4: Alert Fatique



incoming-webhook APP 9:52 PM

Slack Alert:

Project "Poodles": Status "Down"

Website poodles.io is not reachable since 4:35 PM
Trying to reconnect...

-webhook APP 9:52 PM

t:

lles": Status "Down"

lles.io is not reachable since 4:35 PM
onnect...

ook APP 9:52 PM

incoming-webhook APP 9:52 PM

Slack Alert:

Project "Poodles": Status "Down"

Website poodles.io is not reachable since 4:35 PM
Trying to reconnect...

Project "Poodles": Status "

oming-webhook APP 9:52 PM

ck Alert:

ect "Poodles": Statu

site poodles.io is n
ng to reconnect...

incoming-webhook AI

Slack Alert:

Project "Poodles": Status "Dov

Website poodles.io is not reac
Trying to reconnect...

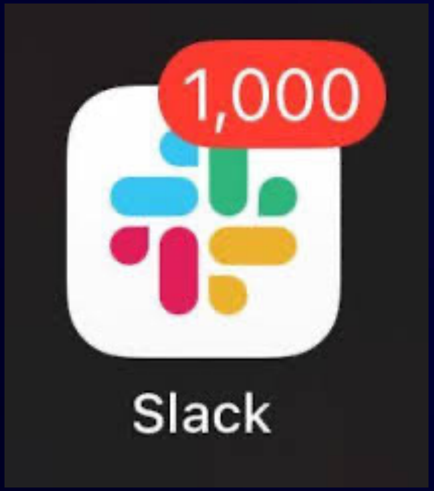
incoming-webhook APP 9:52 PM

Slack Alert:

Project "Poodles": Status "Down"

Website poodles.io is not reachable since 4:35 PM
Trying to reconnect...

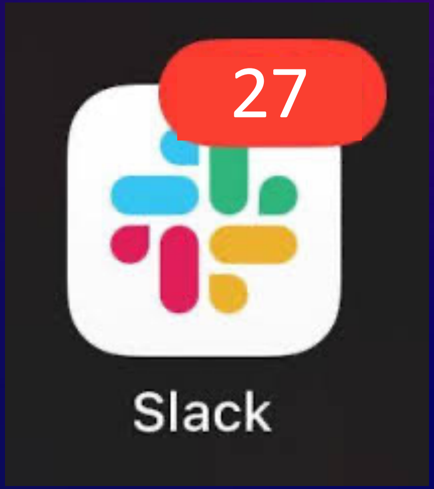
Pitfall #4: Alert Fatique: The Lessons

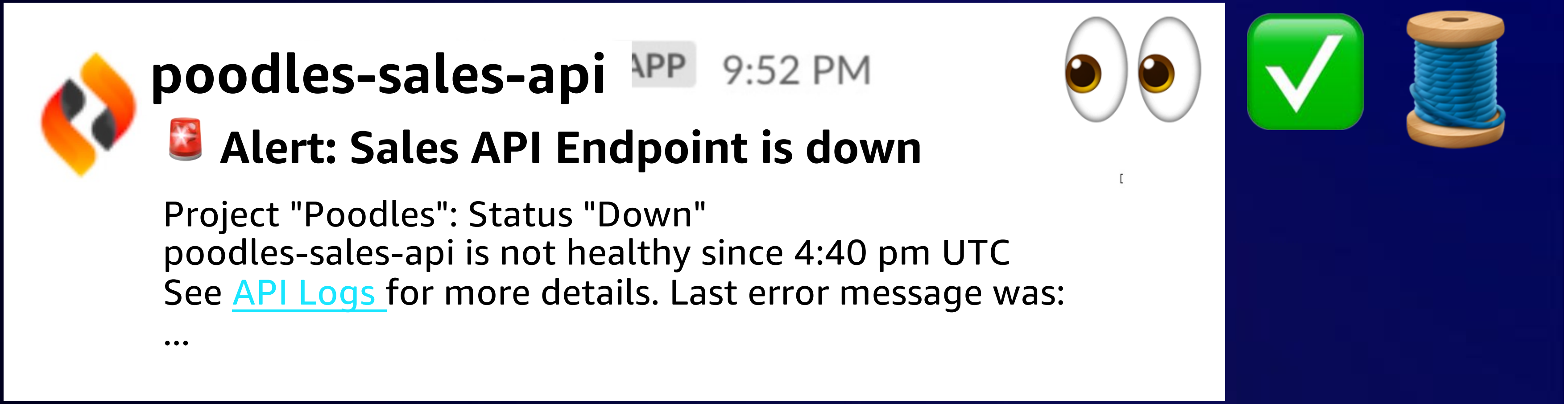


only alert on important things

→

that require action

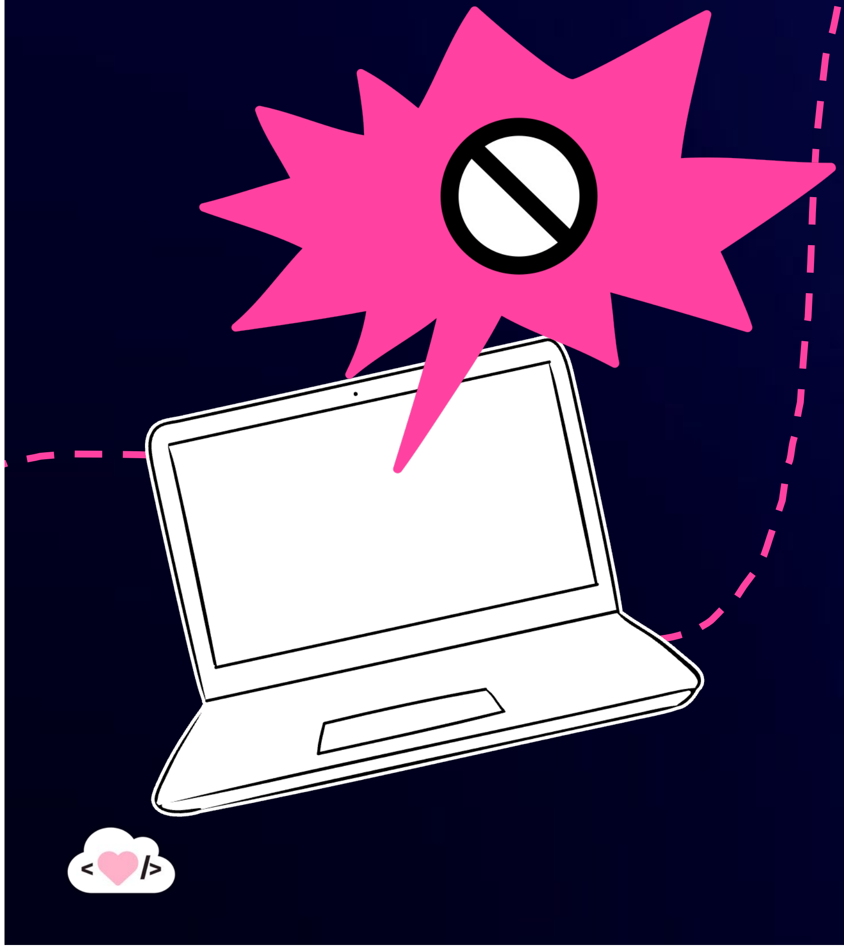




poodles-sales-api APP 9:52 PM

🚨 **Alert: Sales API Endpoint is down**

Project "Poodles": Status "Down"
poodles-sales-api is not healthy since 4:40 pm UTC
See API Logs for more details. Last error message was:
...



Pitfall #4

The Dashboard Theater

Pitfall #3: The Dashboard Theater

https://www.freepik.com/free-photo/close-up-person-using-coding-application-write-test-code_416763469.htm
https://www.freepik.com/free-photo/desktop-pc-wooden-desk-showcases-infographics-client-reach-data_418374016.htm

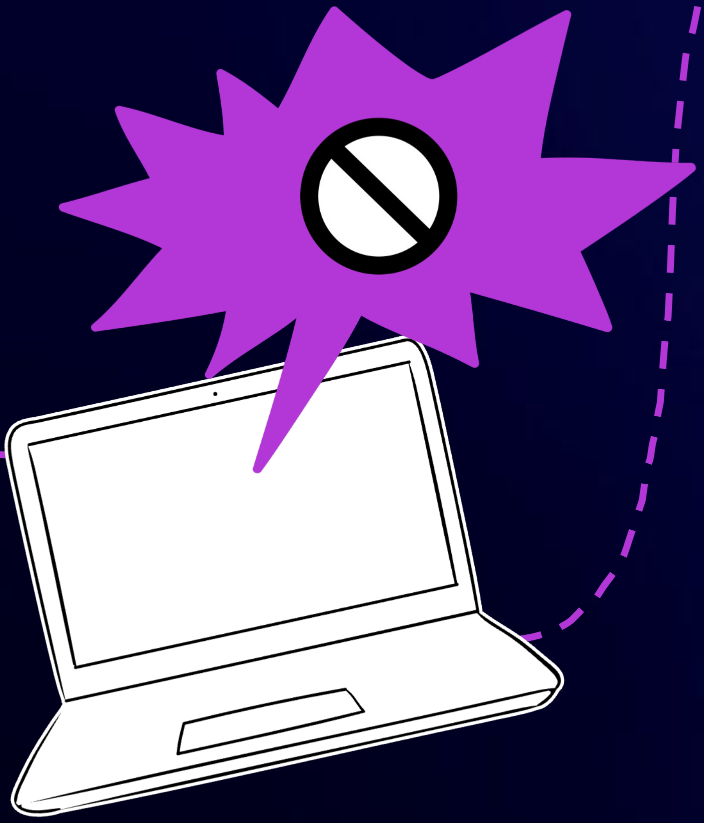
Pitfall #3: The Dashboard Theater

Dashboards are great for

- Known issues
- Trends
- Sharing

Pitfall #2: The Dashboard Theater: The Lesson

But O11y need explorative tools,
to put things in perspective.



Pitfall #5

The "Not My Job" Syndrome

Pitfall #5: "Not my Job Syndrome"

observability is what the OPS team does
I'm just caring about my features
Nahh, they have their dashboards.
Should we add an alert for service downtime?
DEV TEAM MEMBER
O11y <3

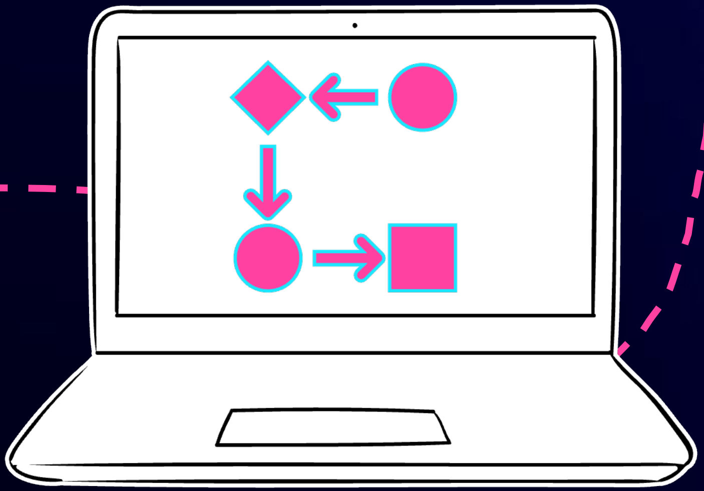
Pitfall #5: "Not my Job Syndrome": The Lesson

observability is shared responsibility

Add to Definition of Done

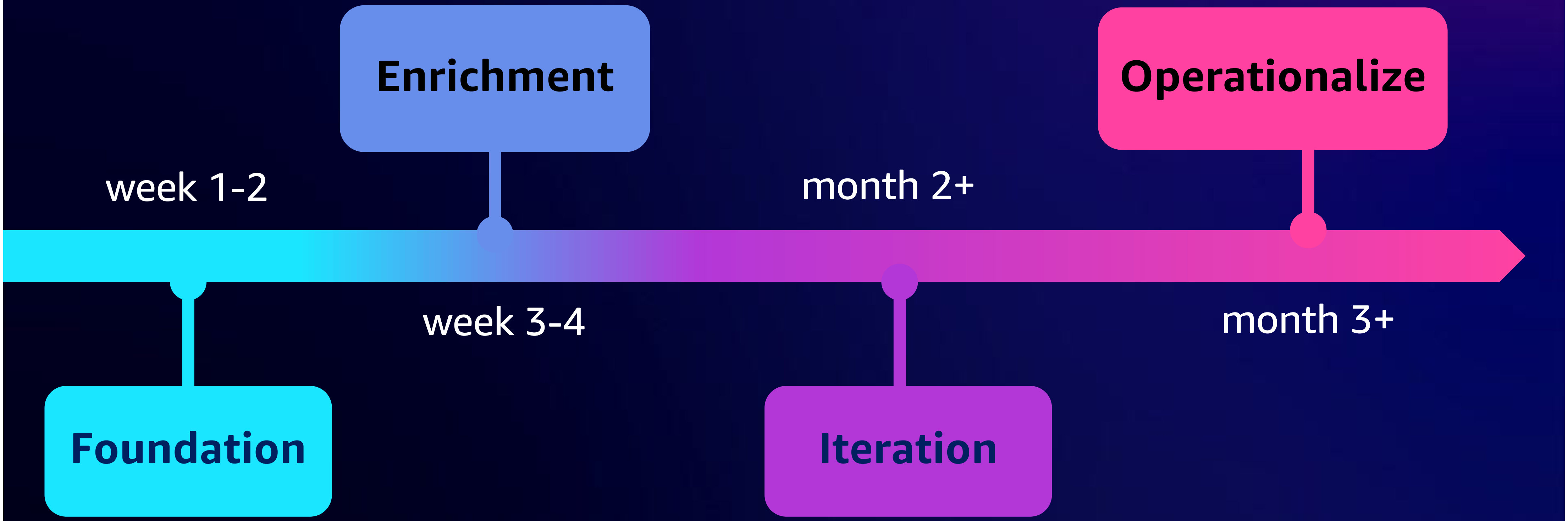
Part of your Code Review

Review in your Weeklies



Taking Action

Taking Action: The 4-Phase Approach



Foundation

week 1-2

Enrichment

week 3-4

Iteration

month 2+

Operationalize

month 3+

Taking Action: Setting The Foundation (Week 1-2)

Foundation

▶️ Pick ONE critical user journey

▶️ The three signals to add:

- Structured logging (e.g. template for log library)
- Basic metrics (e.g. error rate, latency, user access rate etc.)
- Distributed tracing (e.g. OpenTelemetry)

Taking Action: Enrich your setup (Week 3-4)

Enrichment

▶ Enhance your signals:

- Golden Metrics (Traffic, Latency, Erros, Saturation)
- USE (Usage, Saturation, Errors)

▶ Add more context (correlation IDs, labels):

- Business Contest (User ID, Tenant ID, …)
- versions, features, environment
- tag your services with metadata

Taking Action: Iterate (Month 2+)

Iteration

▶️ Apply learnings to next critical user journey

- Instrument dependencies:
  - Databases
  - Caches
  - External APIs
  - …

▶️ Add custom business metrics (KPIs, …)

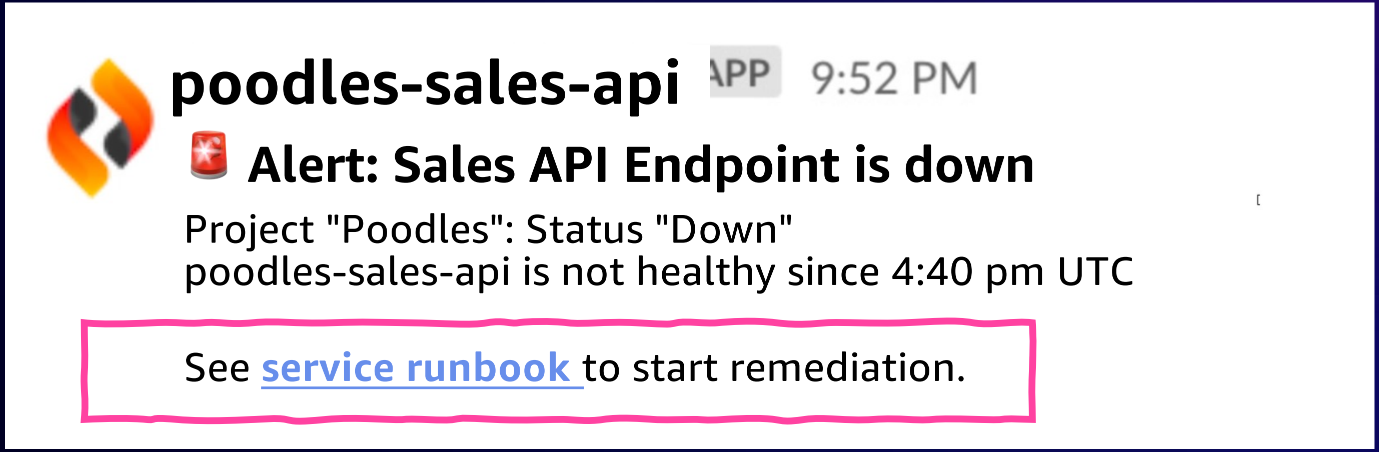
Taking Action: Making it operation ready (Month 3+)

Operationalize

- ▶️ Build runbooks referencing observability data
- ▶️ Create debugging templates
- ▶️ Train team on exploratory investigation
- ▶️ Make observability part of incident postmortems

Taking Action: Making it operation ready (Month 3+)

Operationalize

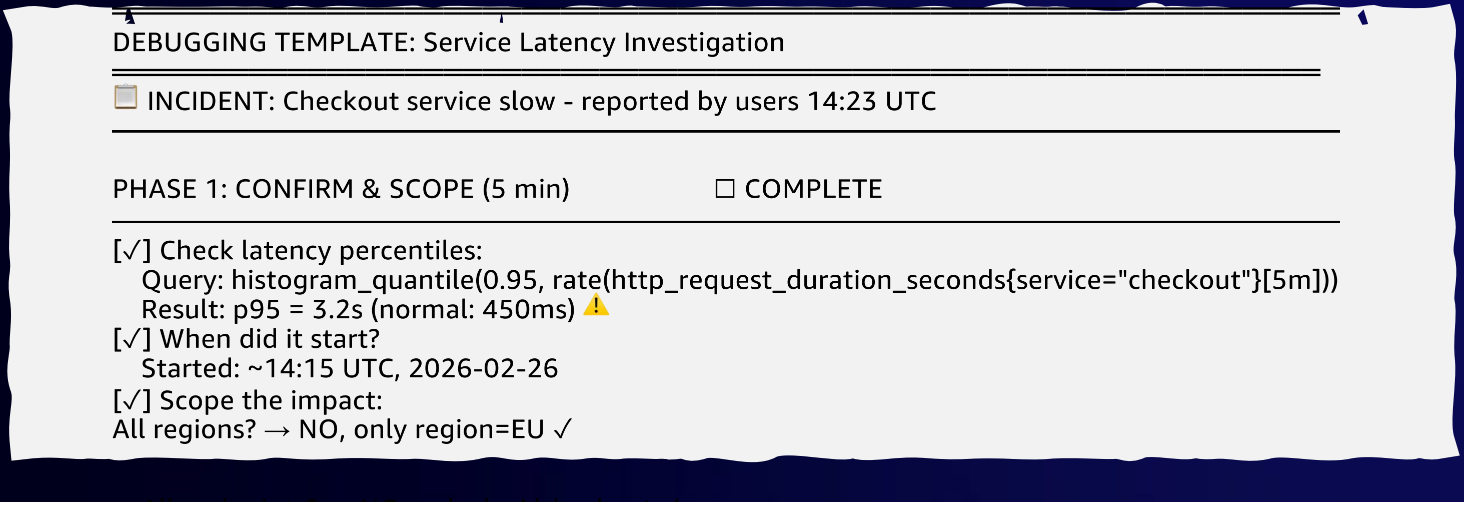


poodles-sales-api APP 9:52 PM

🚨 **Alert: Sales API Endpoint is down**

Project "Poodles": Status "Down"
poodles-sales-api is not healthy since 4:40 pm UTC

See service runbook to start remediation.



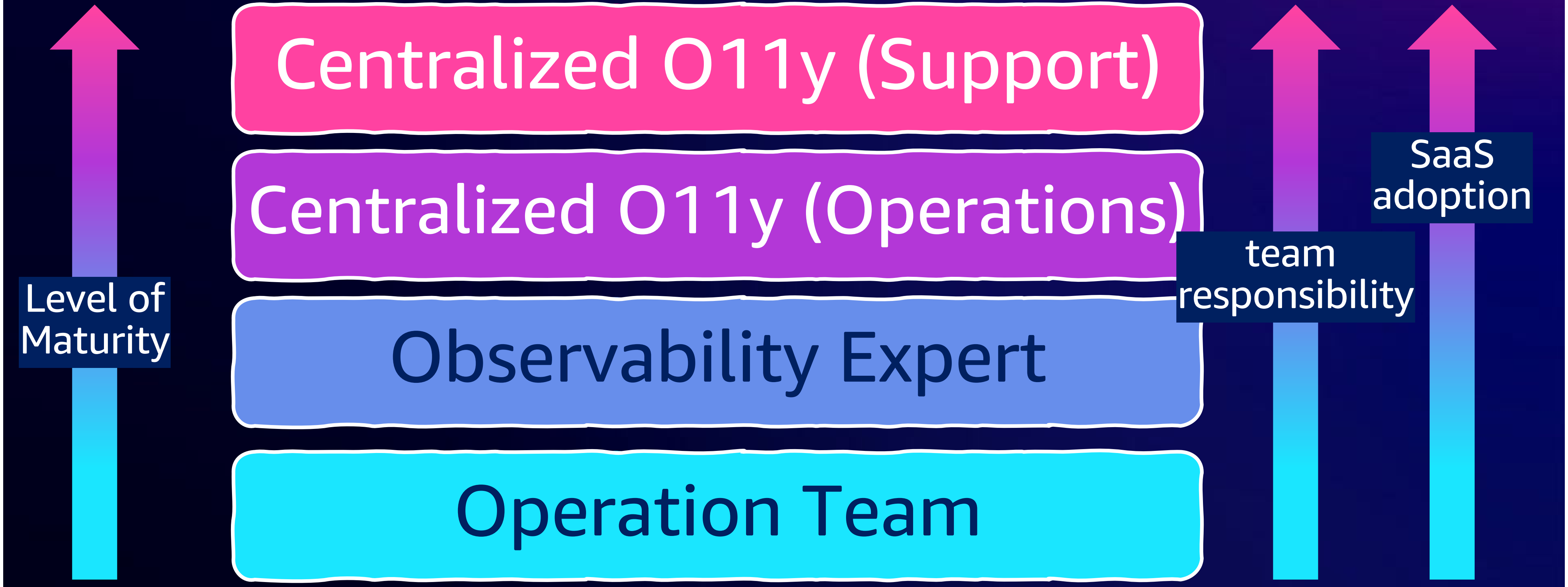
```
DEBUGGING TEMPLATE: Service Latency Investigation

📋 INCIDENT: Checkout service slow - reported by users 14:23 UTC

PHASE 1: CONFIRM & SCOPE (5 min)              ☐ COMPLETE

[√] Check latency percentiles:
    Query: histogram_quantile(0.95, rate(http_request_duration_seconds{service="checkout"}[5m]))
    Result: p95 = 3.2s (normal: 450ms) ⚠️
[√] When did it start?
    Started: ~14:15 UTC, 2026-02-26
[√] Scope the impact:
All regions? → NO, only region=EU ✓
```

Taking Action: Key Enabler For Your Success

- 🌐 Use Open Standards (e.g. OpenTelemetry)
- 🤖 Automate Instrumentation
- 🎯 Measure What Matters to Users
- ✅ Part of Definition of Done
- 💻 Start Local, Think Production

The Observability Maturity Level

The Maturity Level: Key Groups



Centralized O11y (Support)

Centralized O11y (Operations)

Observability Expert

Operation Team

Level of Maturity

team responsibility

SaaS adoption

The Maturity Level: Key Groups

Operation Team

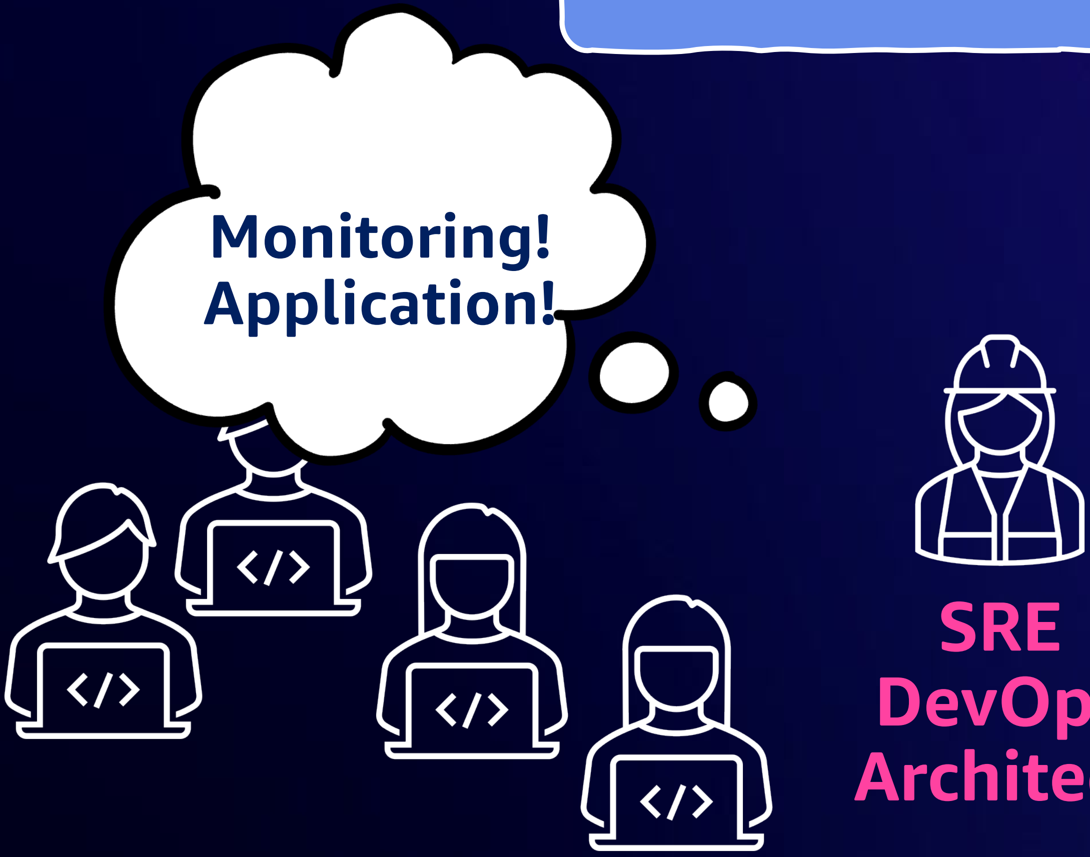
Infrastructure?
Monitoring?
Application?
Product?

Product Team

Operation Team

The Maturity Level: Key Groups

Observability Expert



SRE
DevOps
Architect

Product Team

The Maturity Level: Key Groups

Centralized O11y (Operations)

Monitoring?
Alerting?
We provide for the teams!

Product Team

Central Observabilty Team

The Maturity Level: Key Groups

Centralized O11y (Support)

Monitoring!
Alerting!
Produkt KPIs

Product Team

We support the teams!

Central Observabilty Team

Self-Assessment

Can you trace a single user request?

Do teams use the same standards?

Do you have an o11y support team?

Do you review your observability features regularly?

Do you have runbooks or remediation in place?

Where Should You Start?

> Maturity isn't about tools. It's about how effectively you **understand system behaviour**.

AI-Assited Observability

AI Use Cases: Overview

- 🔍 Anomaly Detection
- 🎯 Root Cause Analysis

- 🔮 Predictive Insights
- 🔔 Intelligent Alerting
- 📖 Automated Runbooks

Use Cases 1: 🔍 Anomaly Detection

🔍 Anomaly Detection

The Challenge

Can't watch every metric 24/7

The Example

15% gradual increase over 3 days in latency
→ some queries running longer than usual

Tools

e.g. Grafana Sift, Datadog Watchdog,
honeycomb Intelligence …

🎯 Root Cause Analysis

The Challenge

50 services, 200 metrics, 1000 log lines

The Example

Payment failures correlated with inventory service deployment + DB query change

Tools

e.g. Datadog Watchdog, Grafana Cloud, Agent0 by dash0, …

Use Cases 3: 🔮 Predictive Insights

🔮 Predictive Insights

The Challenge

Disks fill, certificates expire, … (predictable surprises 🕵🏻‍♀️)

The Example

DB storage will hit 90% in 3 days → proactive scaling

Over the last 3 years, user access went up by 70% → proactive scaling

Tools

e.g. AWS DevOps Agent, honeycomb capabilities,…

Use Cases 4: 🔔 Intelligent Alerting

🔔 Intelligent Alerting

The Challenge

100 alerts, 3 real issues (alert fatigue!)

The Example

15 alerts → 1 intelligent incident with impact assessment
AI grouped related alerts and suppressed noise

Tools

e.g. Agent0 by dash0, …

Use Cases 5: 📖 Automated Runbooks

📖 Automated Runbooks

The Challenge

"What do I do now?" paralysis

The Example

AI suggests relevant runbooks, guides investigation, e.g.
"Similar pattern on 2024-12-03. Check Redis connection pool"

Tools

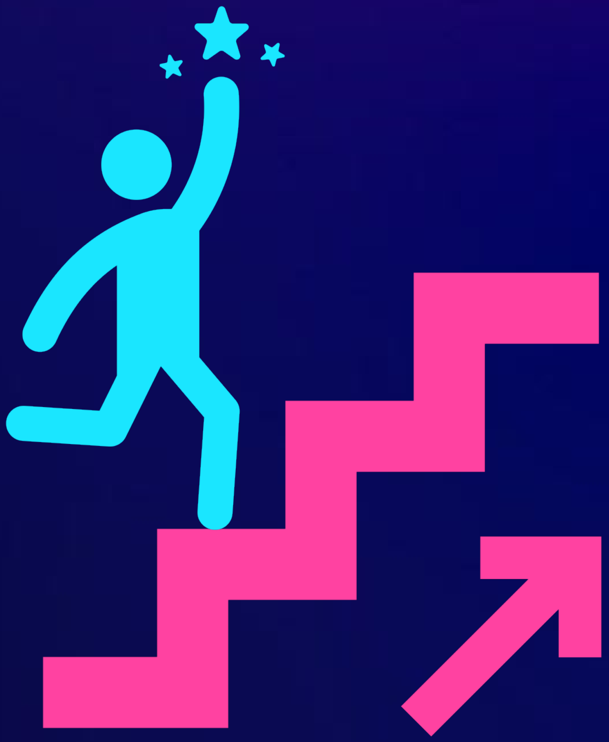
e.g. resolveAI, Grafana Cloud Assistant,…

The Realistic Path with AI

Don't boil the ocean!

Some Practical Steps

- Enable built-in AI features
- Feed the model (30-90 days data)
- Start in shadow mode
- Integrate with workflows



Human + AI Partnership

AI doesn't replace engineers, it amplifies them.

The best workflow

- AI detects → Human investigates
- AI suggests → Human decides
- AI learns → System improves

Four Considerations

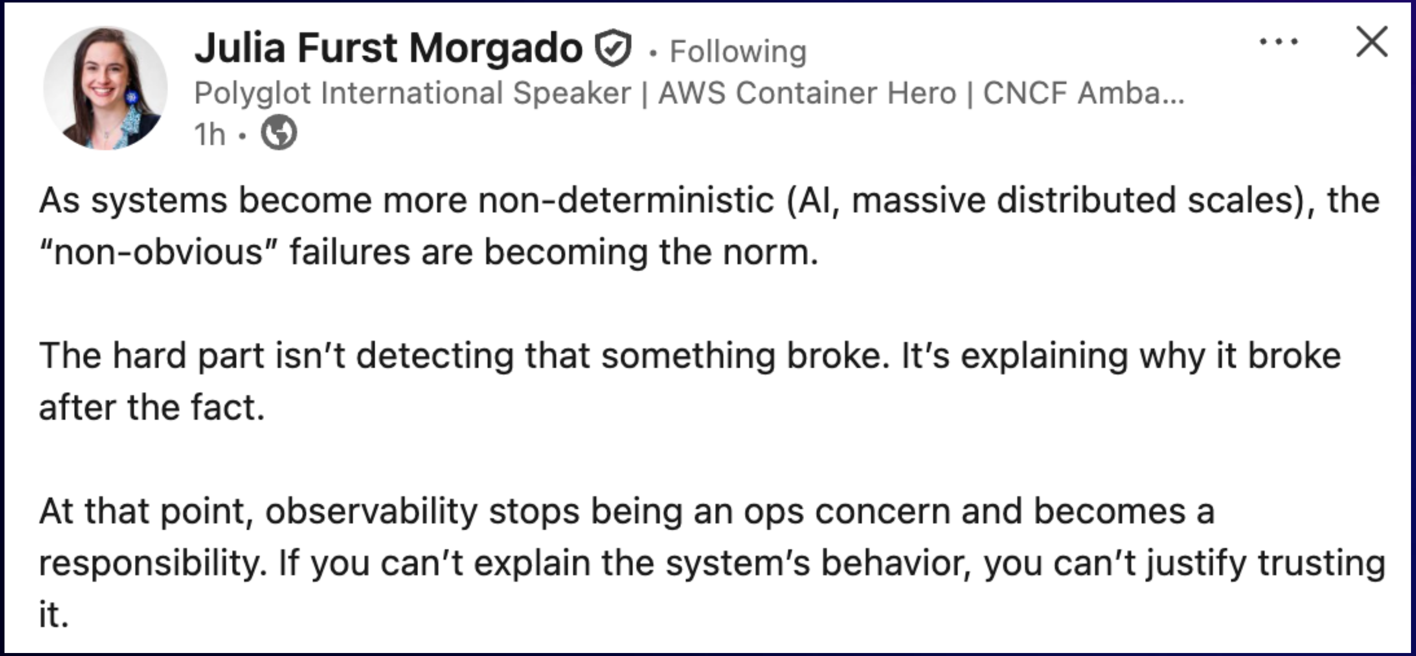
🗑️ Garbage in, garbage out (fix data quality first)

📦 Black box risk (understand recommendations)

💰 Cost (start small, measure ROI)

🧠 Cultural readiness (trust requires training)

So, what's now? 🕵️‍♀️



Julia Furst Morgado · Following

Polyglot International Speaker | AWS Container Hero | CNCF Amba...

1h ·

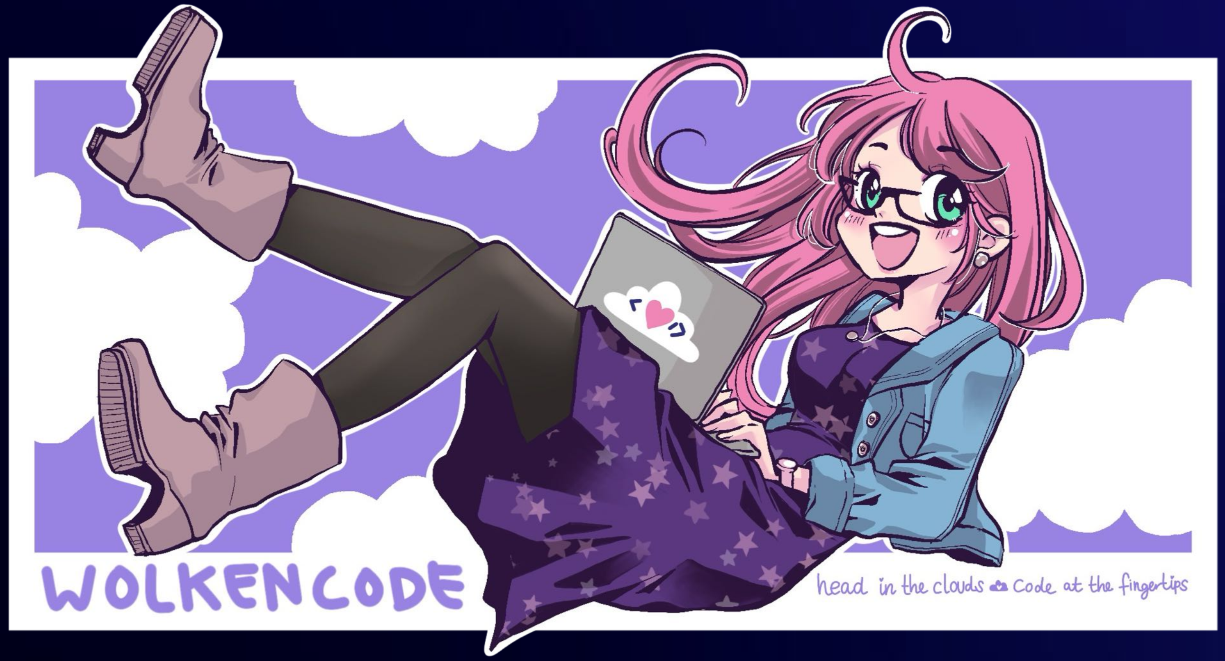
As systems become more non-deterministic (AI, massive distributed scales), the "non-obvious" failures are becoming the norm.

The hard part isn't detecting that something broke. It's explaining why it broke after the fact.

At that point, observability stops being an ops concern and becomes a responsibility. If you can't explain the system's behavior, you can't justify trusting it.

Screenshot taken from Julia Furst Morgado's LinkedIn Page on February, 2026

T H A N K S



linktree.wolkencode.de

Some more things to explore:

- OpenTelemetry Java Examples
- OpenTelemetry Demo (Docker, K8s)
- Charity Major's Blog (worth reading)
- Grafana'S 3rd annual o11y report
- Hands-on Blog Post (causely): Observability for slow queries
- Google's SRE Book (Free Readable Online Version)

Additional Sources used

- Cartoon People: https://www.openpeeps.com/
- Icons used are provided by the icon assests of MS PowerPoint

Disclaimer: The Manga Drawings for the slide "About me" and "Thanks" are drawn by the presenter, Nora Schöner. You are **not allowed** to copy, alter or use them neither commercially nor for your personal usage!