

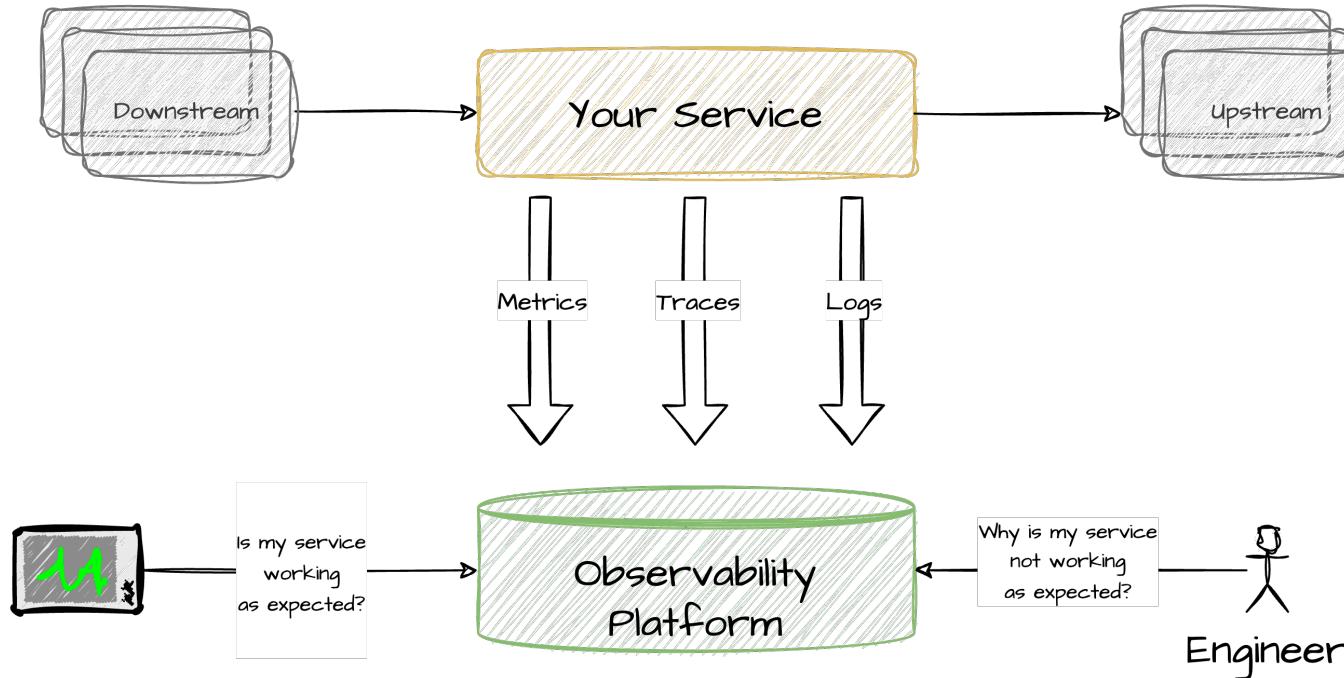
# Shift Into An Observability Mindset With OpenTelemetry

*Daniel Gomez Blanco*

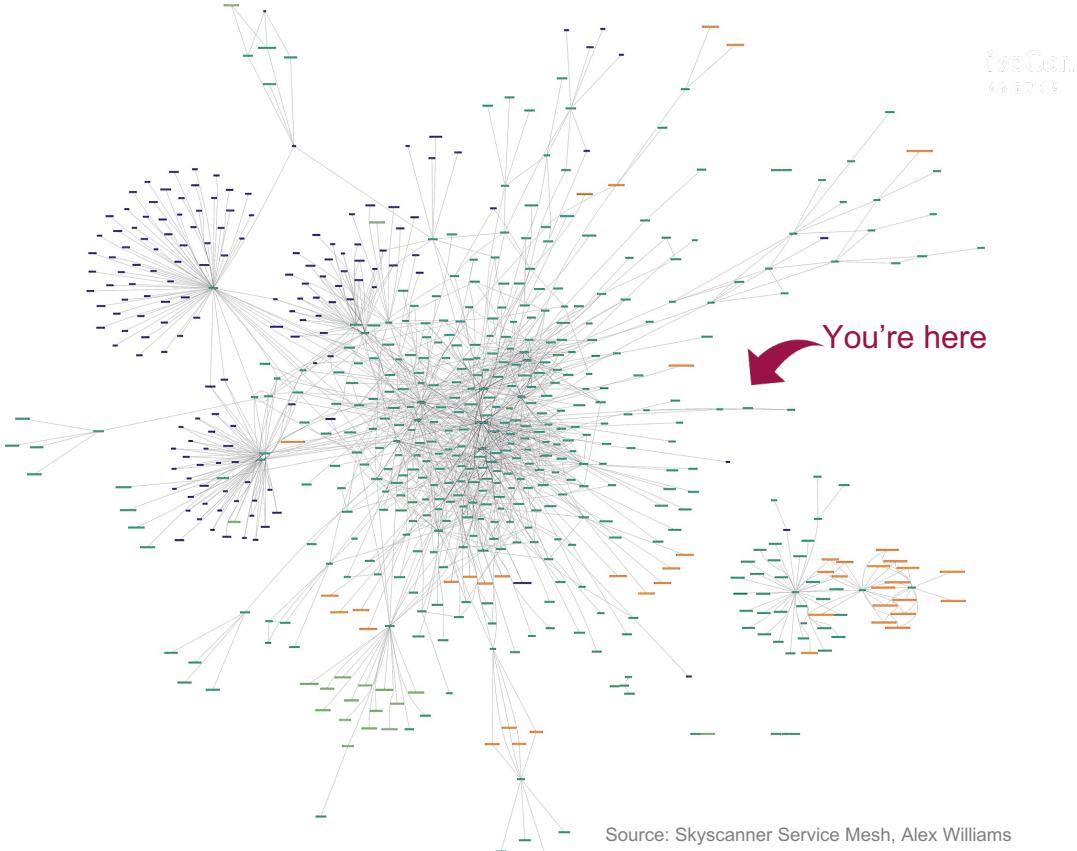
Principal Software Engineer, Skyscanner  
OpenTelemetry Governance Committee Member

To enable  
**effective observability**  
by making  
high-quality, portable  
telemetry ubiquitous

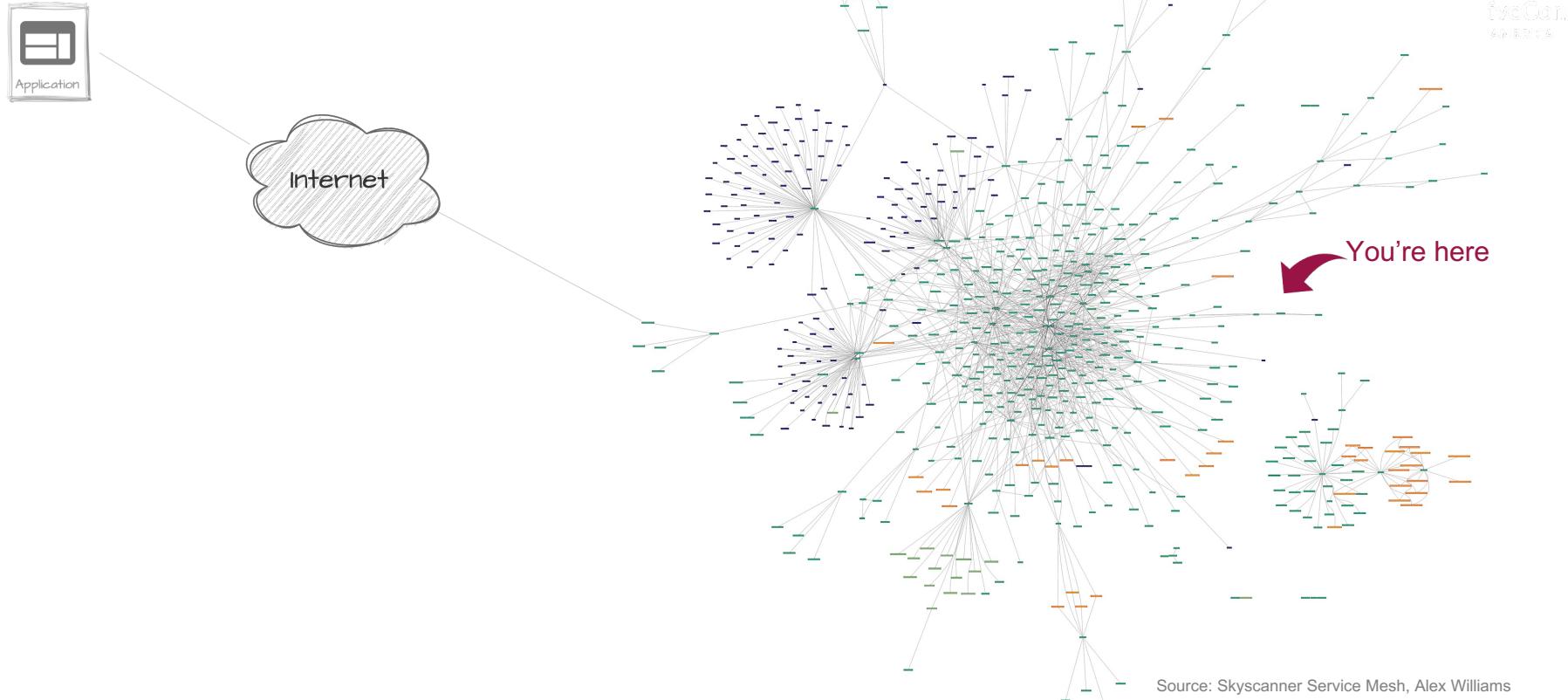
# We tend to focus on our service...



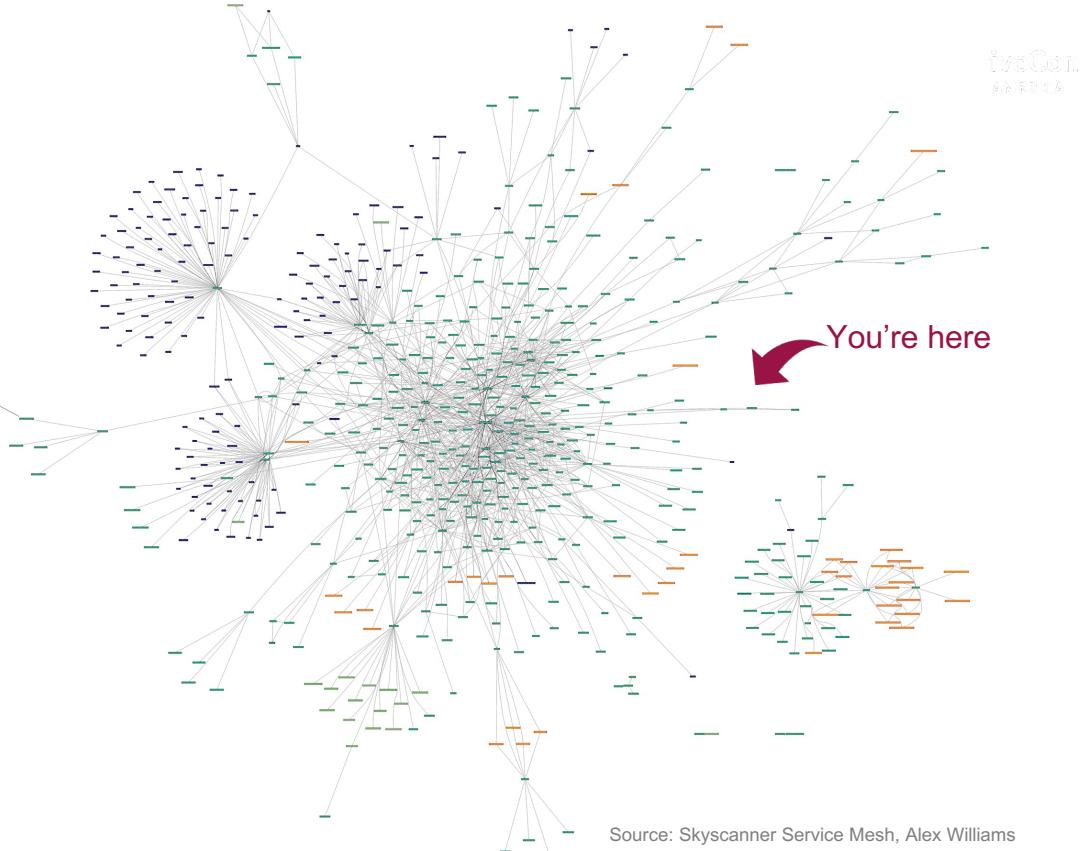
# ...and we forget the system



# ...and we forget the system



# ...and we forget the system

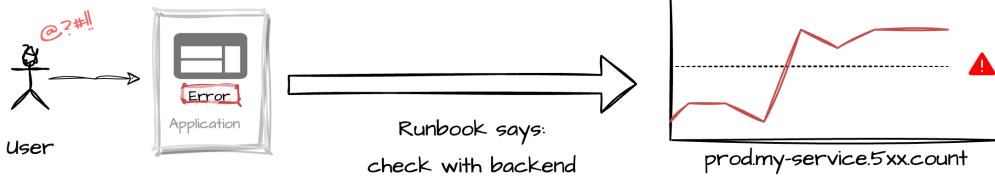


# Who am I?

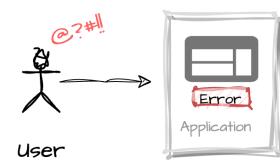
- Principal Engineer at Skyscanner leading observability since 2020
- Joined in 2018 to work on client-side performance and K8s resource optimisation
- Platform engineer in organisations of all sizes for 13 years
- OpenTelemetry Governance Committee member since November 2023
- Author of Practical OpenTelemetry: Adopting Open Observability Standards Across Your Organization (Apress, 2023)



# Debugging with no context → Intuition



# Debugging with no context → Intuition



Runbook says:  
check with backend



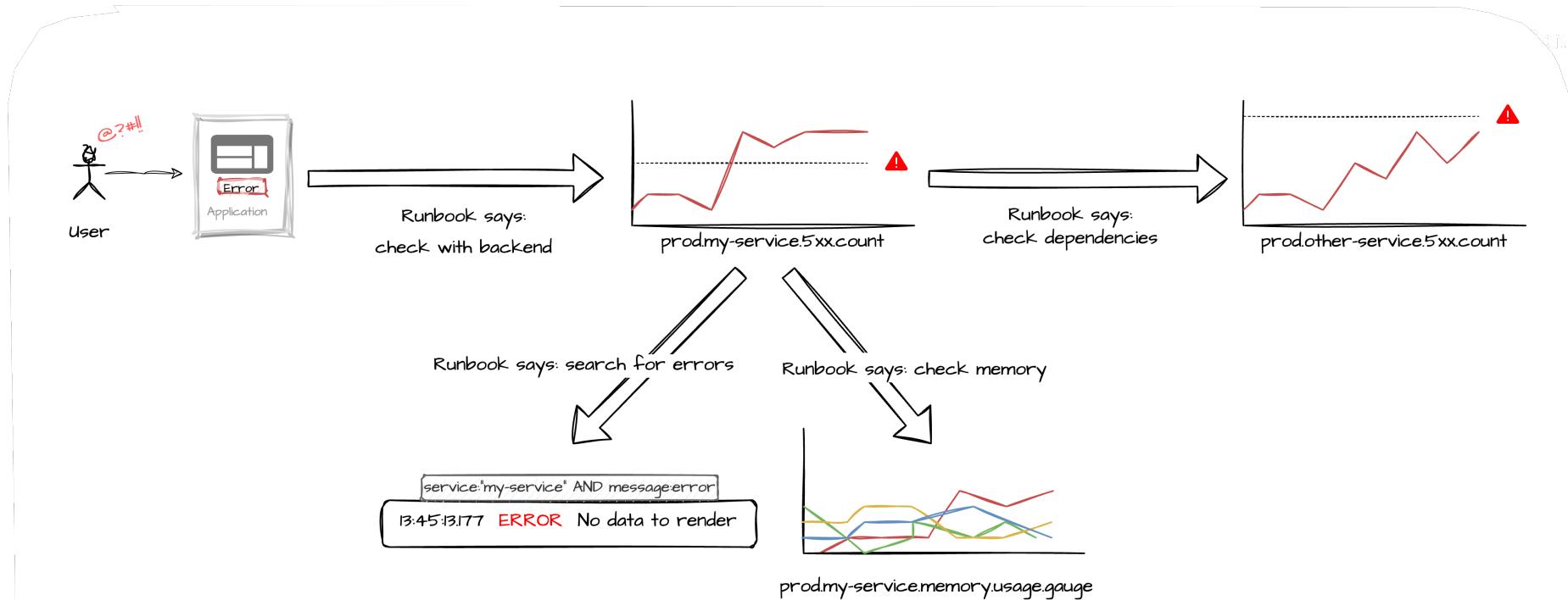
Runbook says: search for errors

service:'my-service' AND message:error  
13:45:13177 ERROR No data to render

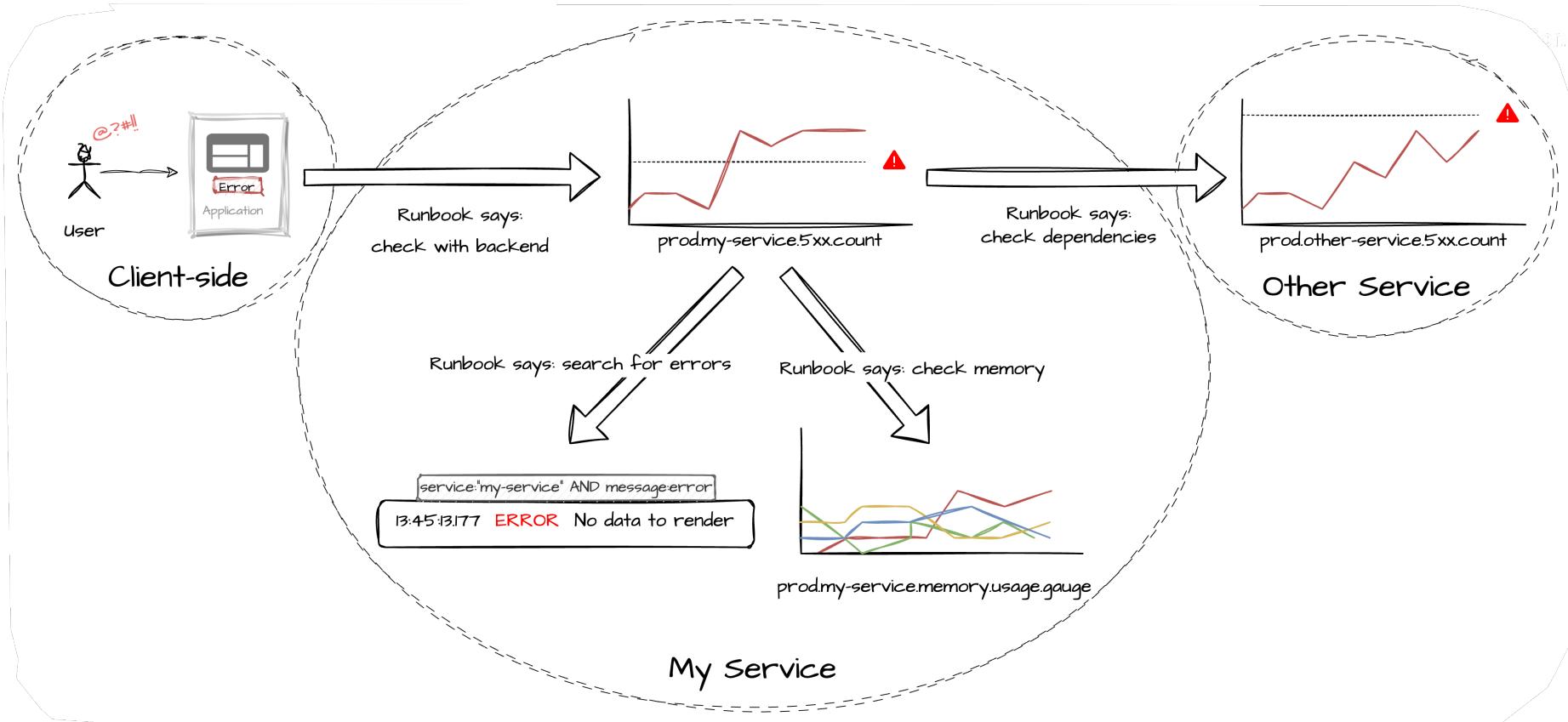
Runbook says: check memory



# Debugging with no context → Intuition



# Debugging with no context → Intuition

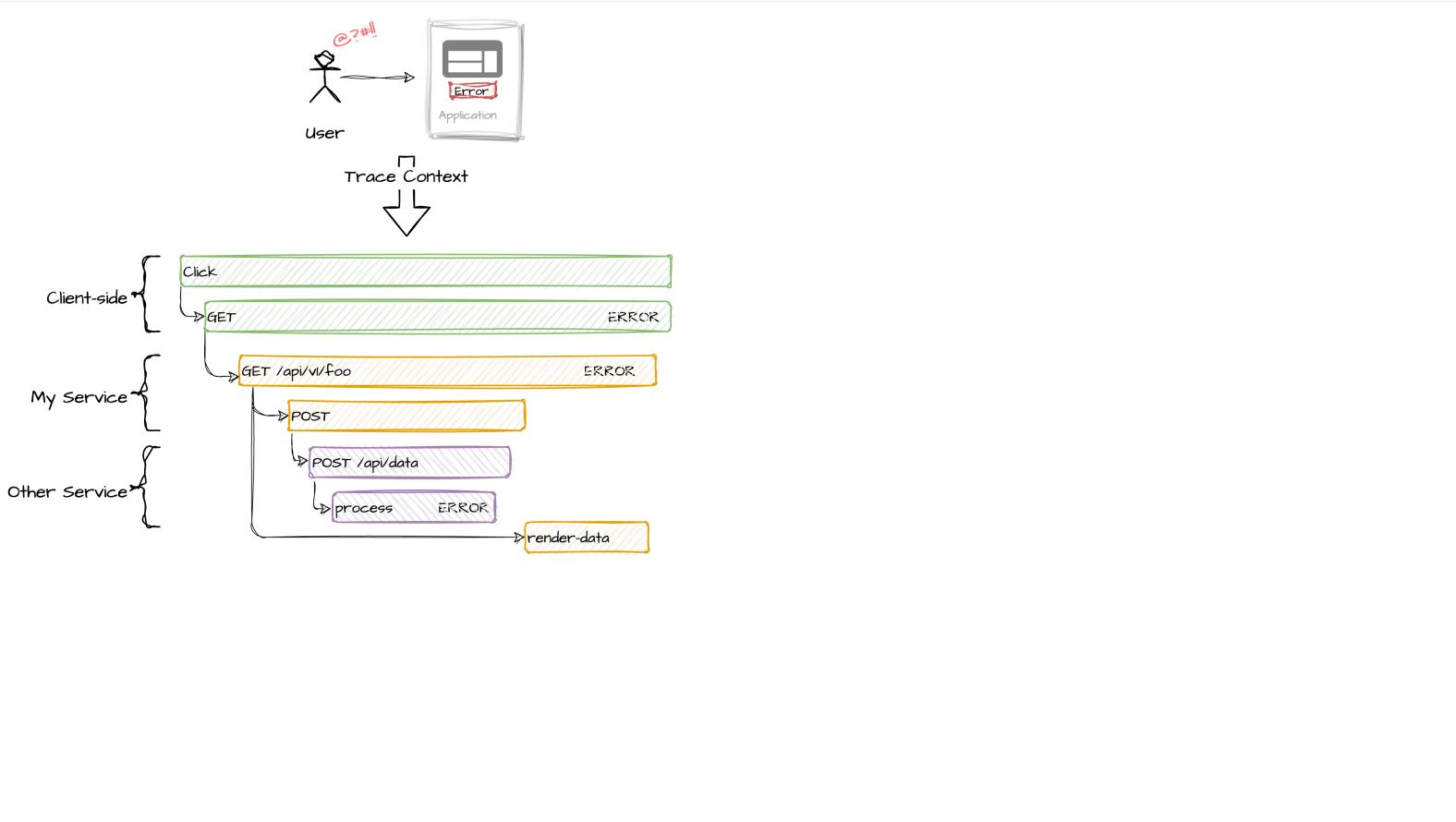


# Debugging with no context → Intuition

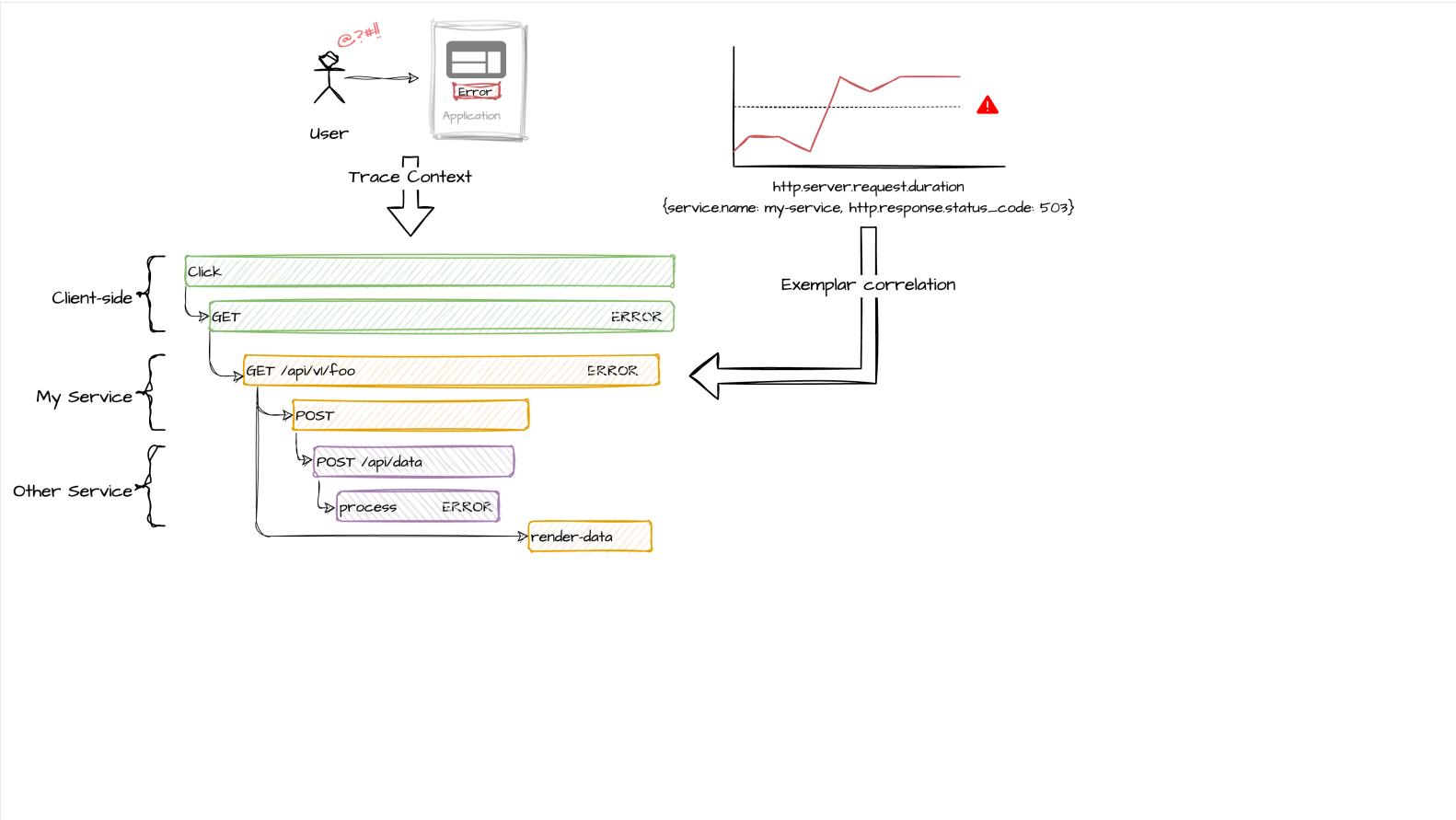


Source: <https://imgflip.com/i/8j8o12>

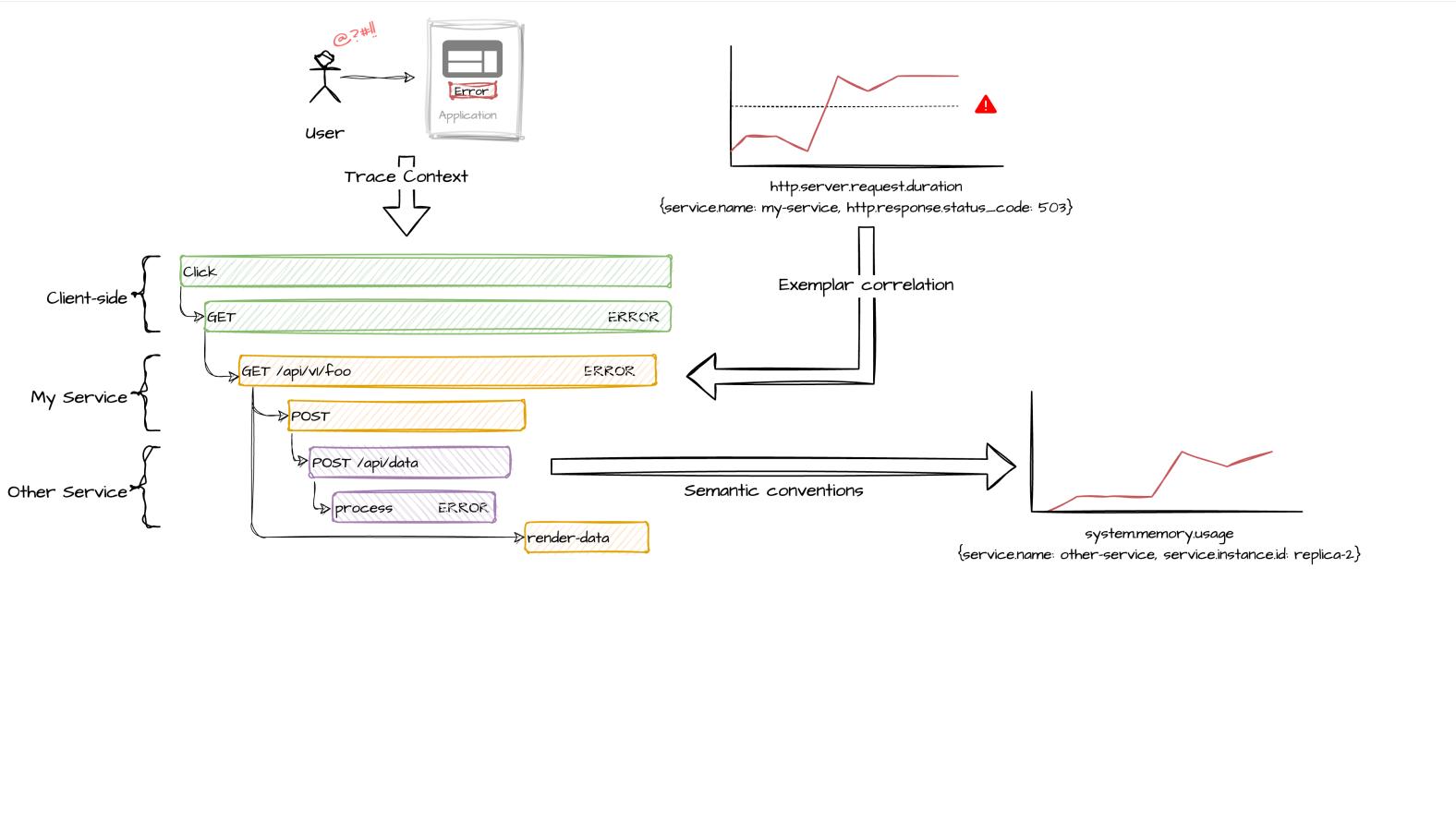
# Context & standards → Evidence



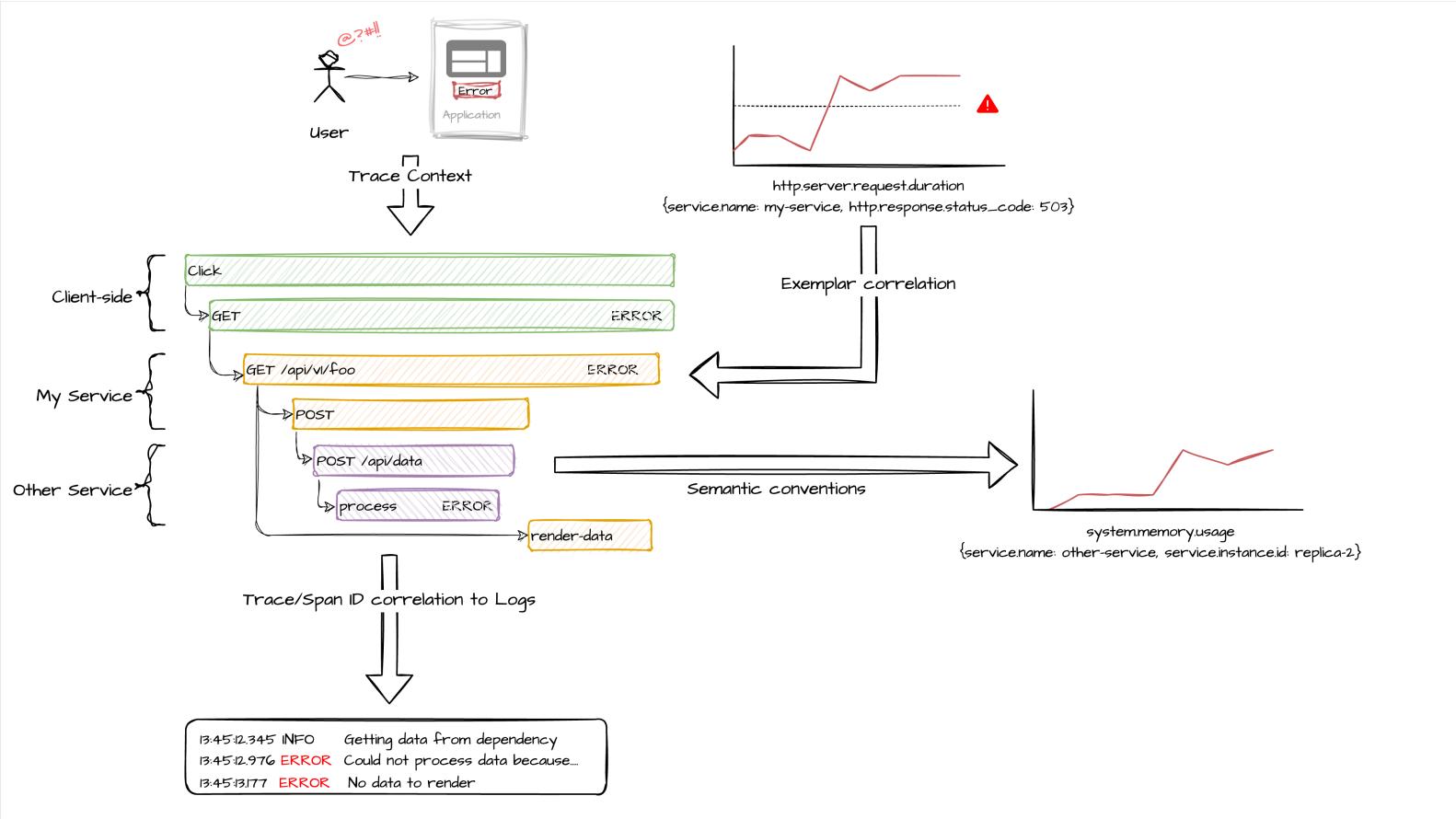
# Context & standards → Evidence



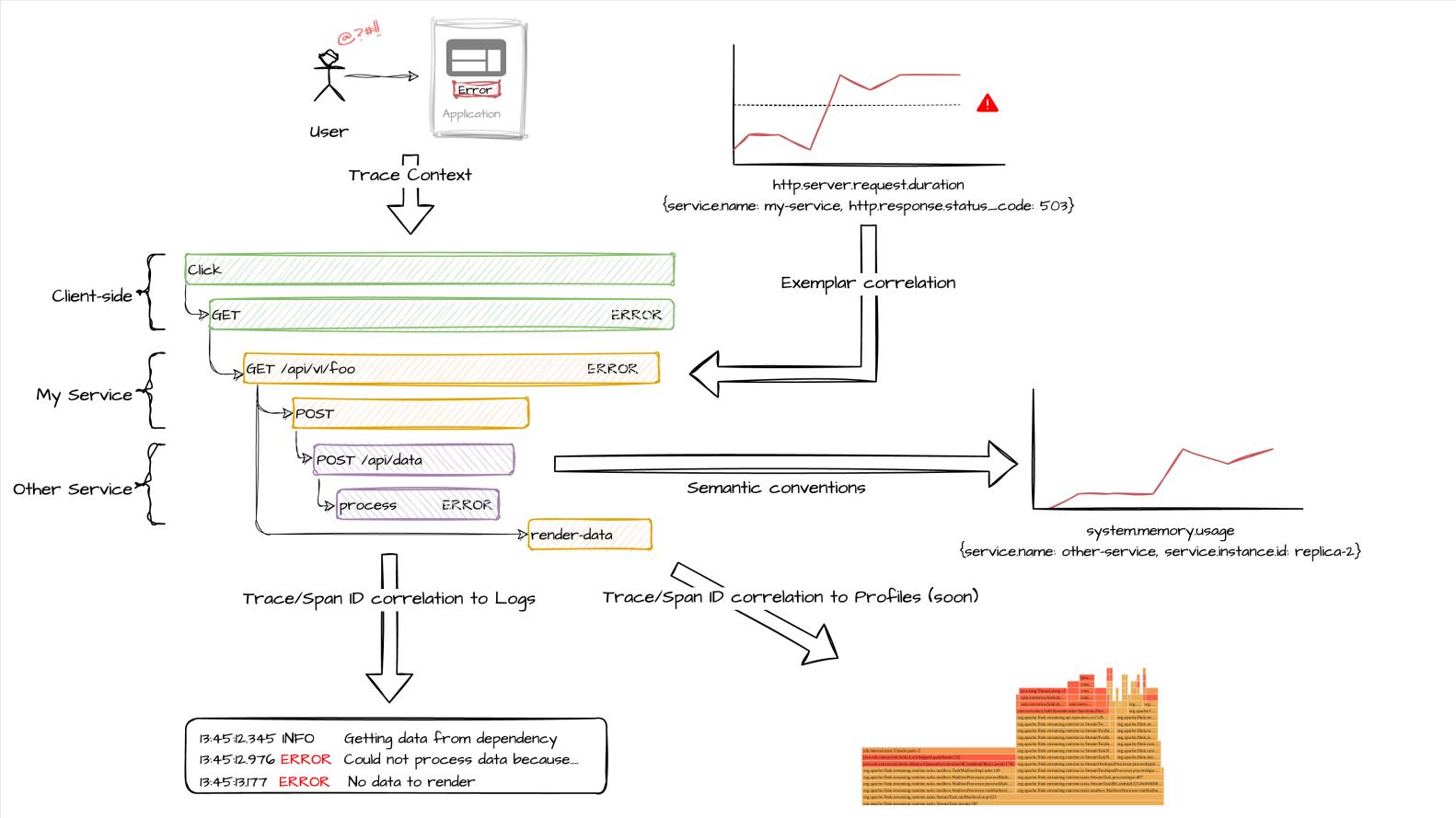
# Context & standards → Evidence



# Context & standards → Evidence

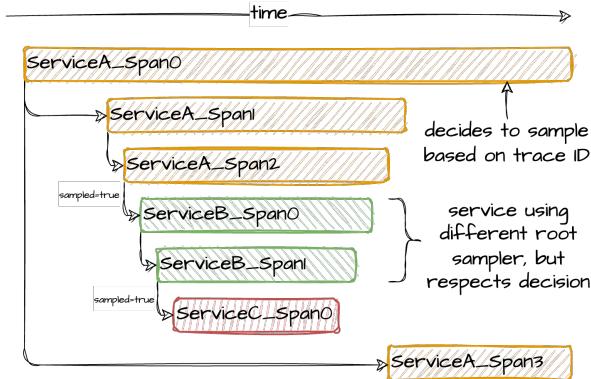


# Context & standards → Evidence



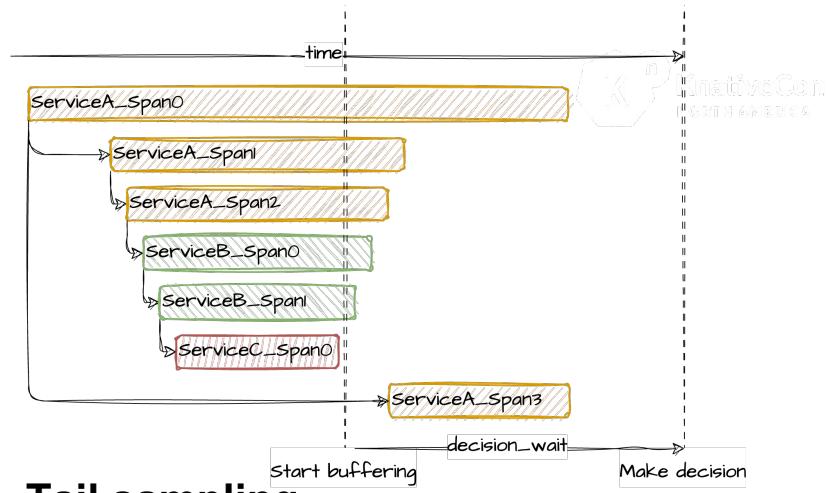
Most  
telemetry data  
gathered for  
debugging  
purposes  
is **never used**

# Trace sampling to keep the useful data



## Head sampling

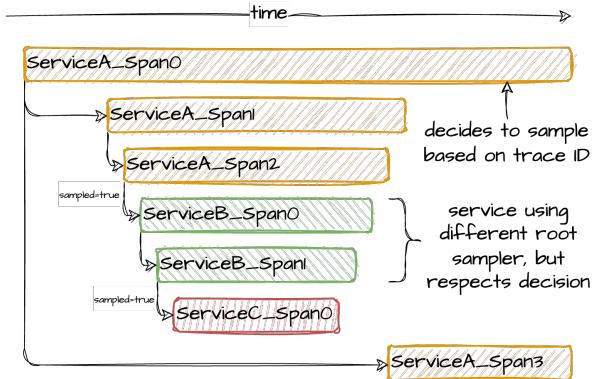
- Trace Context can propagate sampling decisions
- Easy configuration and maintenance
- Efficient resource utilisation
- Probabilistic (i.e. cannot consider whole trace)



## Tail sampling

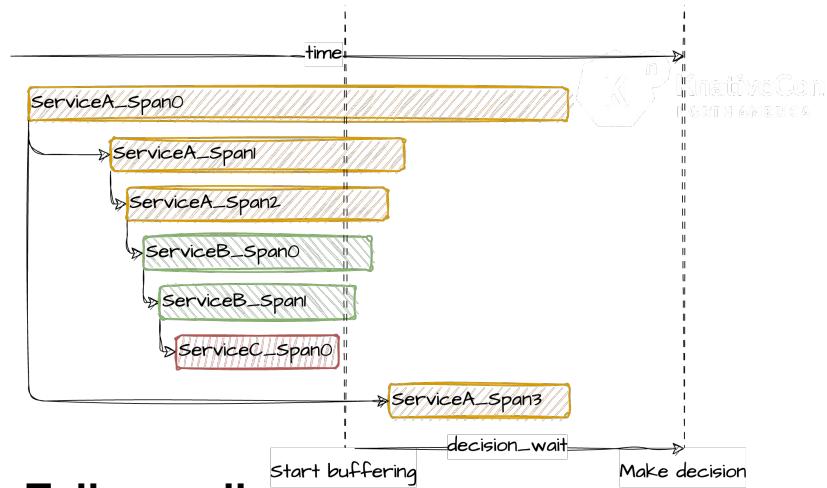
- More complex to implement and operate
- Can consider whole trace in sampling decision
- OpenTelemetry Collectors or vendor-specific features

# Trace sampling to keep the useful data



## Head sampling

- Trace Context can propagate sampling decisions
- Easy configuration and maintenance
- Efficient resource utilisation
- Probabilistic (i.e. cannot consider whole trace)



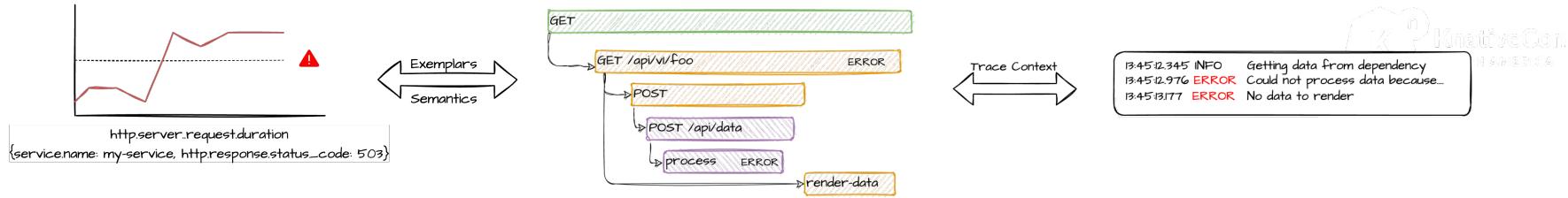
## Tail sampling

- More complex to implement and operate
- Can consider whole trace in sampling decision
- OpenTelemetry Collectors or vendor-specific features



Skyscanner keeps ~5% of the 7.5M traces and 150M spans produced each minute

# Context to rationalise data production



## Metrics

- Stable signal → alerting and long-term trend analysis
- Lower volume can make exports more resilient
- Keep cardinality low and correlate to traces for granularity

## Traces

- High granularity and context → debugging, optimisation, short-term analysis
- Queuing and retrying more expensive
- Possible to generate metrics from spans

## Logs

- High volume, low context → when you can't use anything else??
- Background tasks, legacy libs
- Please structure them! Or use OpenTelemetry appenders
- Events API Interface for non-application logs

# Metric Views to control streams



- Decouple measurements from their aggregation
  - OpenCensus concept
- Control of resulting metrics streams via SDK config
- Customise auto-instrumented metrics
  - Instrumentation libs
  - Library authors

```
meter_provider:  
  views:  
    # What instrument(s) does this apply to?  
    - selector:  
      instrument_name: http.server.request.duration  
      instrument_type: histogram  
    # What is the resulting stream(s)?  
    stream:  
      name: app.http.route.count  
      aggregation: sum  
      attribute_keys:  
        - http.route
```

Schema following <https://github.com/open-telemetry/opentelemetry-configuration>

# How do you change the mindset?



Observability  
engineer



Rest  
of the  
company

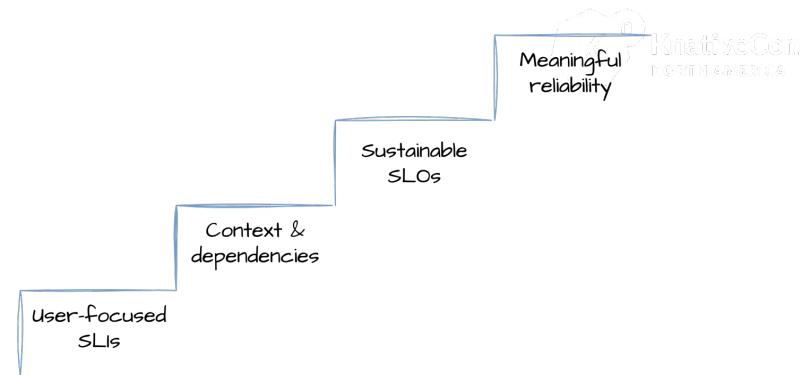
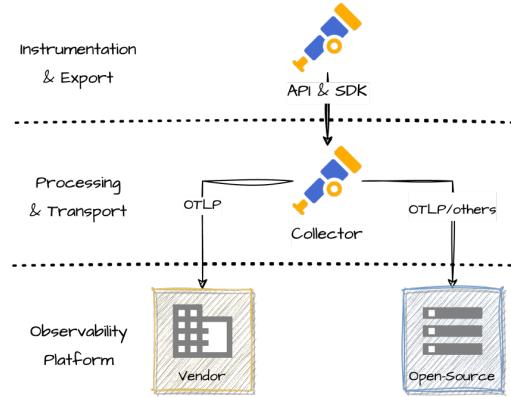


imgflip.com

Source: <https://imgflip.com/i/8j8m6f>



# Communicating value to leadership



## Simplification and future-proofing

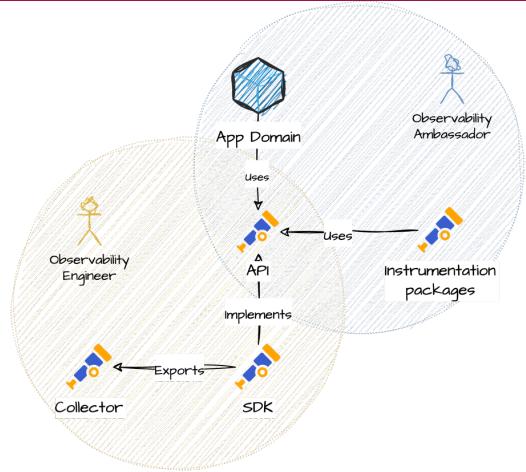
- Future-proof stable API, decoupled from SDK
- Standard and efficient tooling for transport and processing
- Integration with vendors and open-source platforms

## Correlation to business outcomes

- Instrument SLIs that directly relate to user experience
- Context to identify dependencies and set realistic targets
- Effective observability to reduce time-to-resolve

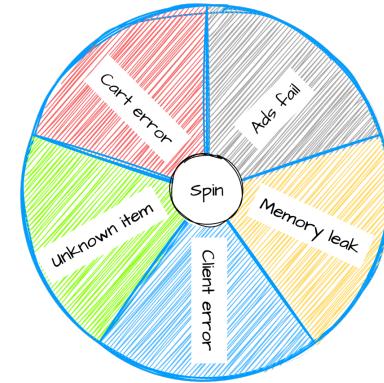
# Changing engineer mindset

Observability Day  
EUROPE



## Cross-organisation alignment

- Observability Engineers → Enablement
  - Implement company-wide standards out of the box
  - Maintain infra and SDK config
- Observability Ambassadors → Adoption
  - Evangelise and support engineers in their domain



## Make it fun!

- Debugging is a skill, and it can be trained
- Gamifying root cause analysis (wheel of misfortune)
- OpenTelemetry Demo feature flags to simulate incidents
- My experience: teams that use context always win!

- **Context over intuition**
- **Right tool for the right job**
- **Observability as a cross-functional discipline**

# Q&A

Catch me at OpenTelemetry Observatory in the Expo Hall!



Please leave feedback at <https://sched.co/1YFjB>