# Mastering Observability with OS Tools

**Luca Raveri**
Software Engineer at Talentware

# Who Am I

- **Hi** 👋 I'm Luca Raveri, Software Engineer and AWS Solutions Architect

💻 Backend Development

☁️ Cloud Architectures

🔍 Observability

📌 Venice, Italy

Talentware

# Talentware

- **Mission**: Helping companies build, retain, and grow talent through a skill-based approach.

- **Tech Stack**: Full-stack JavaScript, AWS, and an internal Data Science team.

📌 Milan, Italy

klyn Fitboxing Isola
Brooklyn Fitboxing
WAO Isola Coworking
Via Pietro Borsieri
Via Cola Montano
Via Carlo Farini
Via Guglielmo Pepe
Via Pastrengo
Isola
Via Federico Con
Via Gaetano de Ca
Monumentale
Milano Porta Garibaldi

Talentware

## Do you really know your system?

Talentware

## Common Scenario

***The day after the release...***

Talentware

## What are Monitoring and Observability?

- **Monitoring**: Continuous measurement of a system's health to detect anomalies, performance issues, or downtime → *tells us that something is wrong*
- **Observability**: The ability to understand the internal state of a complex system by examining the data it produces → *tells us why it is wrong*

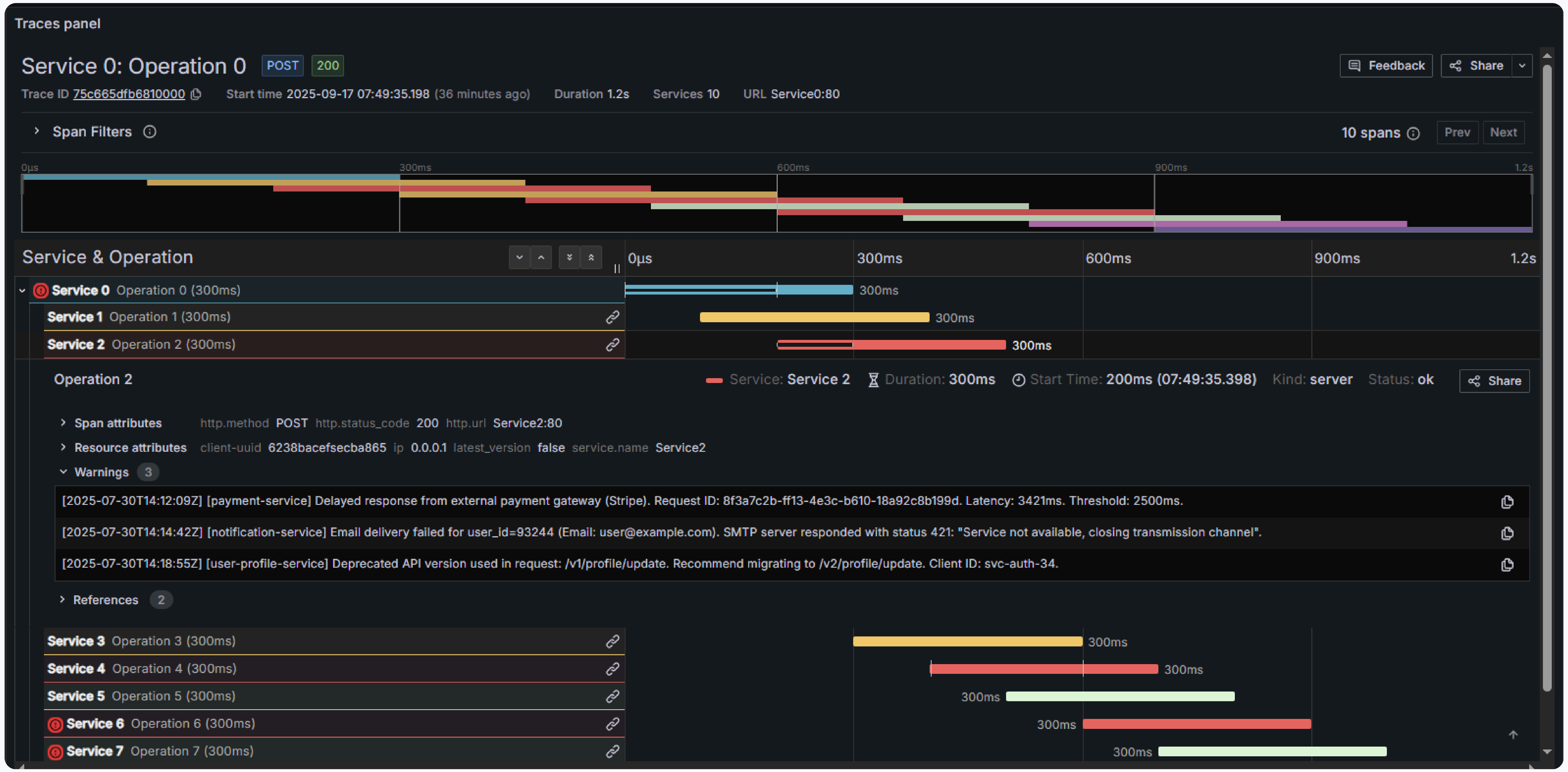
Talentware

# The Three Pillars Of Observability

- **Metrics**: Numeric measurements over time (e.g. CPU usage) → *What is happening*
- **Logs**: Discrete, timestamped records of events → *Why is happening*
- **Traces:** Request flow through the system → *Where is happening*

Talentware

# What is a Trace


Traces panel

Service 0: Operation 0 POST 200

Trace ID 75c665dfb6810000 Start time 2025-09-17 07:49:35.198 (36 minutes ago) Duration 1.2s Services 10 URL Service0:80

Span Filters 10 spans

| Service & Operation | Duration |
|---|---|
| Service 0 Operation 0 (300ms) | 300ms |
| Service 1 Operation 1 (300ms) | 300ms |
| Service 2 Operation 2 (300ms) | 300ms |

Operation 2 — Service: Service 2 Duration: 300ms Start Time: 200ms (07:49:35.398) Kind: server Status: ok

Span attributes http.method POST http.status_code 200 http.url Service2:80

Resource attributes client-uuid 6238bacefsecba865 ip 0.0.0.1 latest_version false service.name Service2

Warnings 3

```
[2025-07-30T14:12:09Z] [payment-service] Delayed response from external payment gateway (Stripe). Request ID: 8f3a7c2b-ff13-4e3c-b610-18a92c8b199d. Latency: 3421ms. Threshold: 2500ms.
[2025-07-30T14:14:42Z] [notification-service] Email delivery failed for user_id=93244 (Email: user@example.com). SMTP server responded with status 421: "Service not available, closing transmission channel".
[2025-07-30T14:18:55Z] [user-profile-service] Deprecated API version used in request: /v1/profile/update. Recommend migrating to /v2/profile/update. Client ID: svc-auth-34.
```

References 2

| Service & Operation | Duration |
|---|---|
| Service 3 Operation 3 (300ms) | 300ms |
| Service 4 Operation 4 (300ms) | 300ms |
| Service 5 Operation 5 (300ms) | 300ms |
| Service 6 Operation 6 (300ms) | 300ms |
| Service 7 Operation 7 (300ms) | 300ms |

Talentware

https://play.grafana.org

## Key Benefits of Observability

- **Lower MTTD (Mean Time to Detect)**
  Problems are identified quickly, often before users notice.

- **Lower MTTR (Mean Time to Resolve)**
  Root causes are found and fixed faster, reducing downtime and impact

Talentware

## Observability

## ===

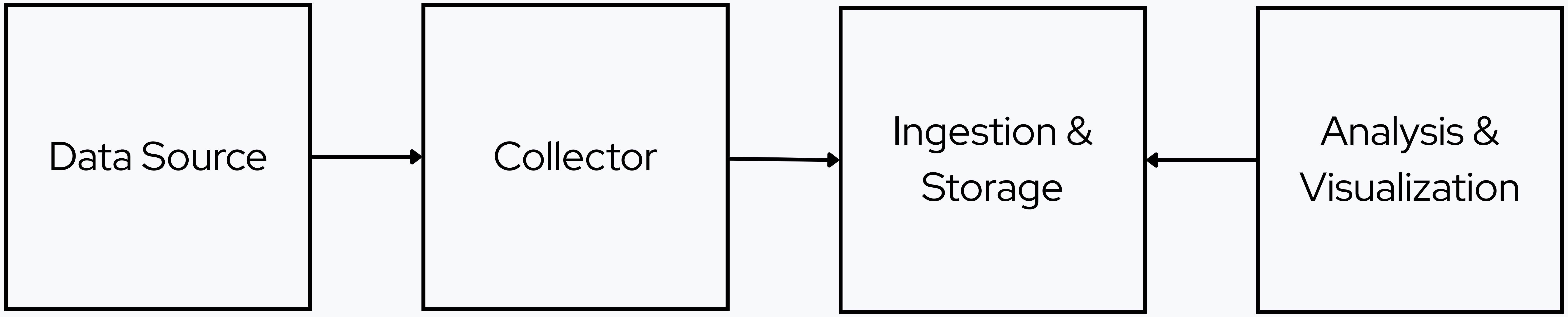
## Performance

Talentware

# Demo Time!

Talentware

# Let's Build an Observability System



## LGTM Stack

**Loki**

Database for logs

**Grafana**

Visualization tool

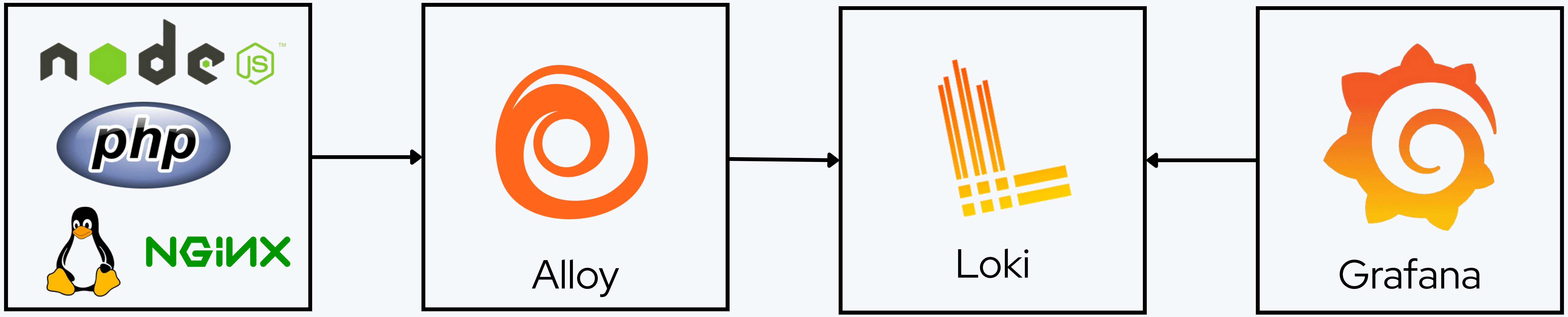
**Tempo**

Database for traces

**Mimir**

Database for metrics

# Let's build an Observability System



Talentware

## Let’s Start Logging

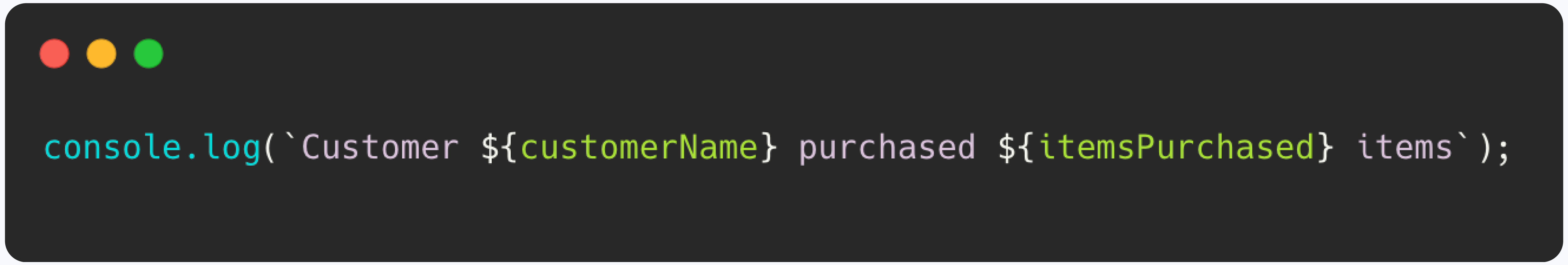
- Use a logger instead of native functions
- Apply log levels and timestamps
- Format logs (JSON)
- Never log secrets and PII
- Set retention policies (e.g., logrotate)

Talentware

## Logging Strategies

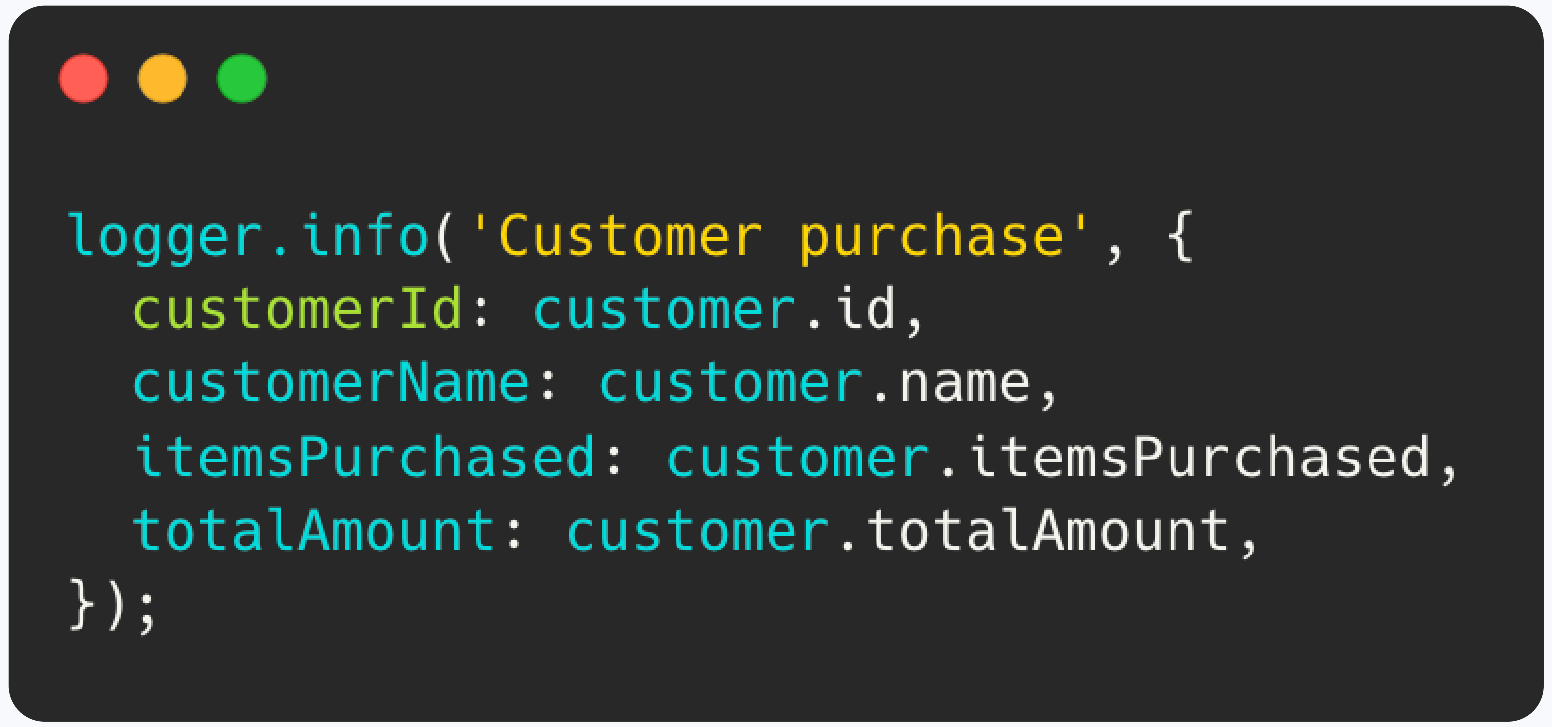
- Log system inputs and outputs
- Centralize logging in a single layer
- Use UUIDs to correlate related logs
- More logs means more insight, but higher overhead

Talentware



```javascript
console.log(`Customer ${customerName} purchased ${itemsPurchased} items`);
```

❌



```javascript
logger.info('Customer purchase', {
  customerId: customer.id,
  customerName: customer.name,
  itemsPurchased: customer.itemsPurchased,
  totalAmount: customer.totalAmount,
});
```

✓

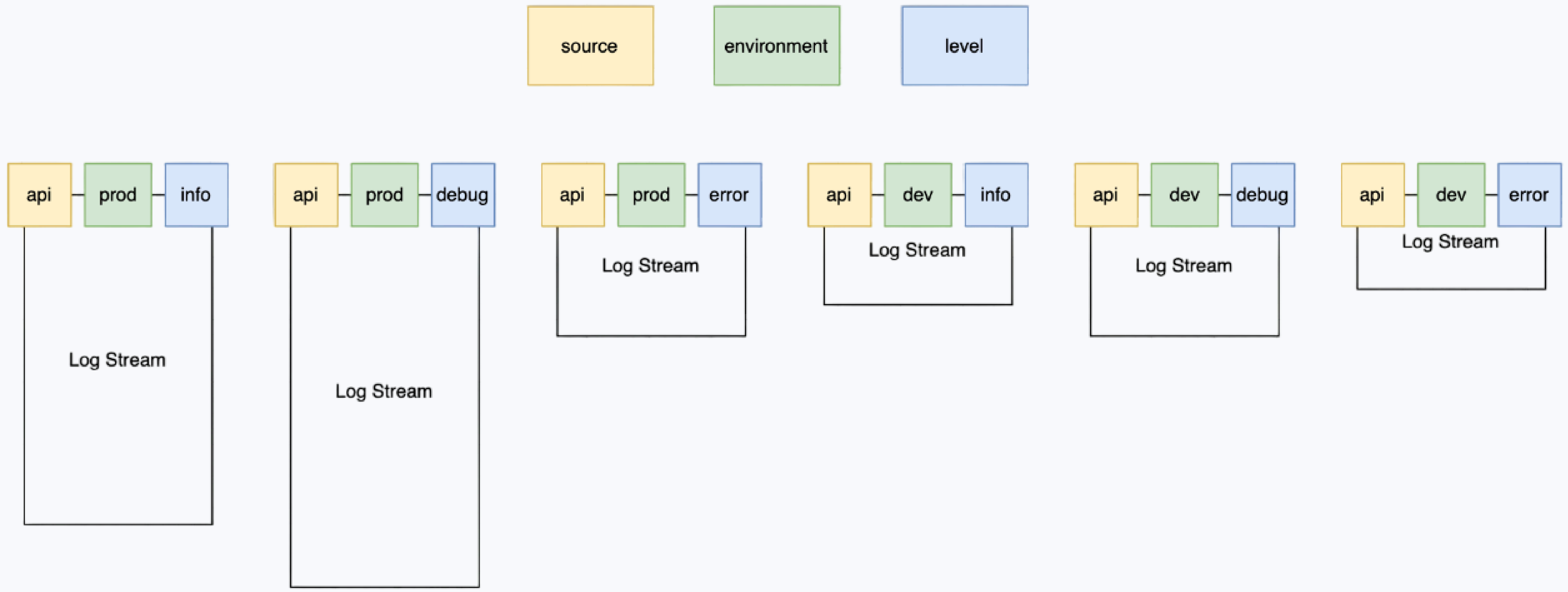
Talentware

## What is Loki

*“Loki is a horizontally scalable, highly available, multi-tenant log aggregation system inspired by Prometheus. It is designed to be very cost effective and easy to operate. It does not index the contents of the logs, but rather a set of labels for each log stream.”*

Talentware

# How Loki Works



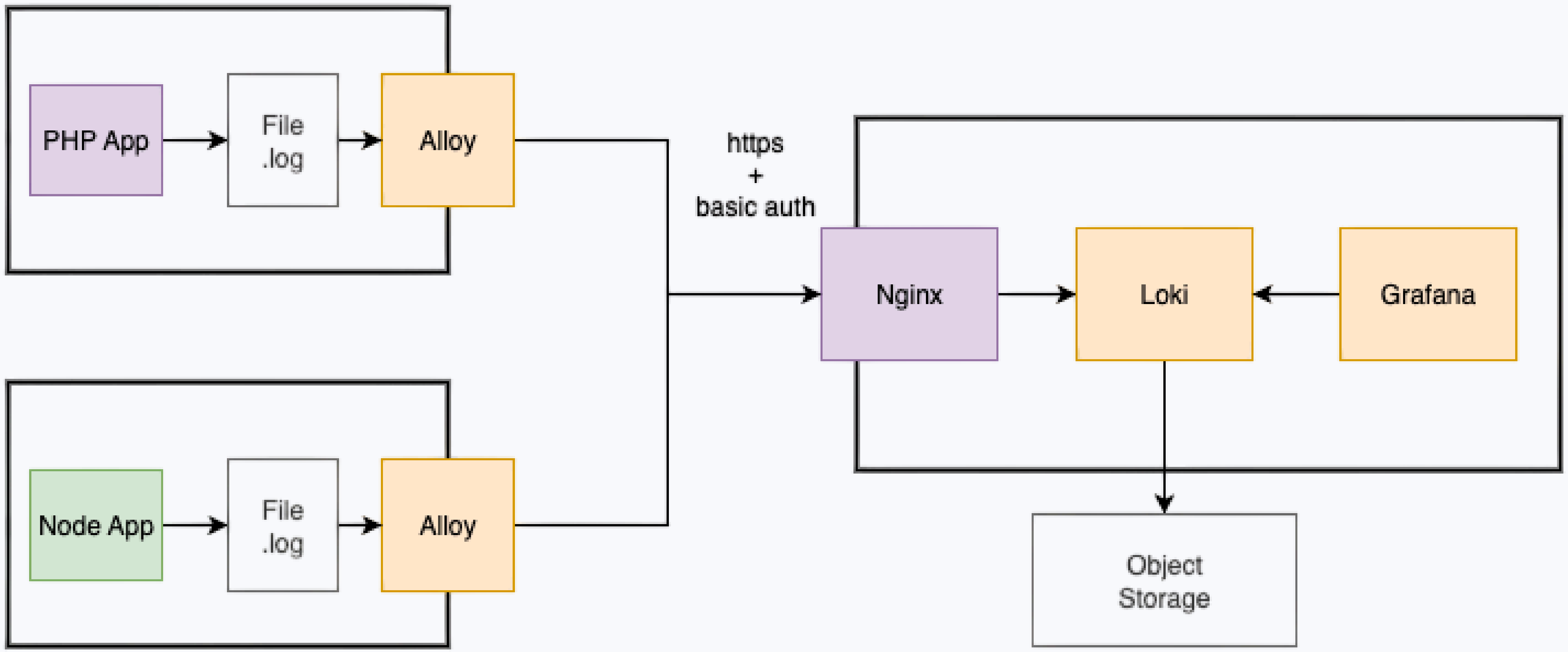
Talentware

# How to Send Logs to Loki

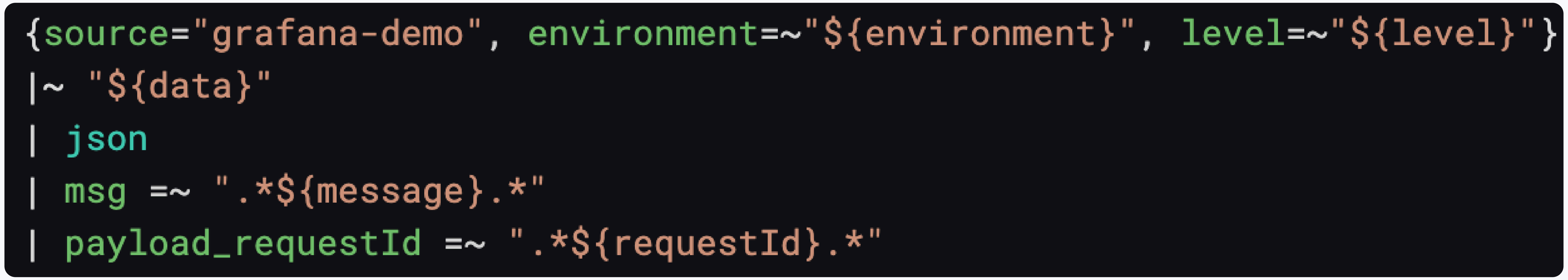
**Grafana Alloy** is a collector that automatically reads your log files and pushes them to Loki. It can:

- Apply **static labels** to all logs from a file
- Extract **dynamic labels** from log content
- Perform **transformations** on log data

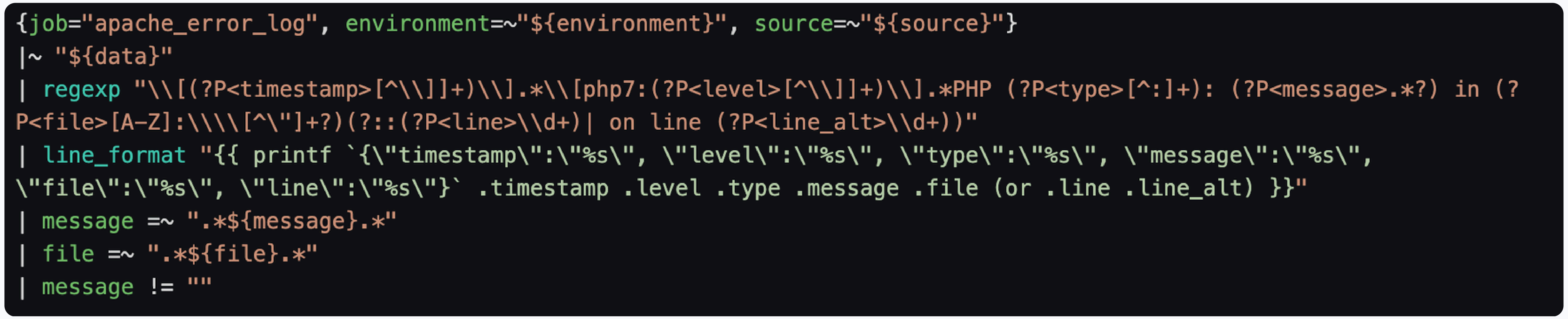
# Hosting Loki



# How to Query Logs



```
{source="grafana-demo", environment=~"${environment}", level=~"${level}"}
|~ "${data}"
| json
| msg =~ ".*${message}.*"
| payload_requestId =~ ".*${requestId}.*"
```

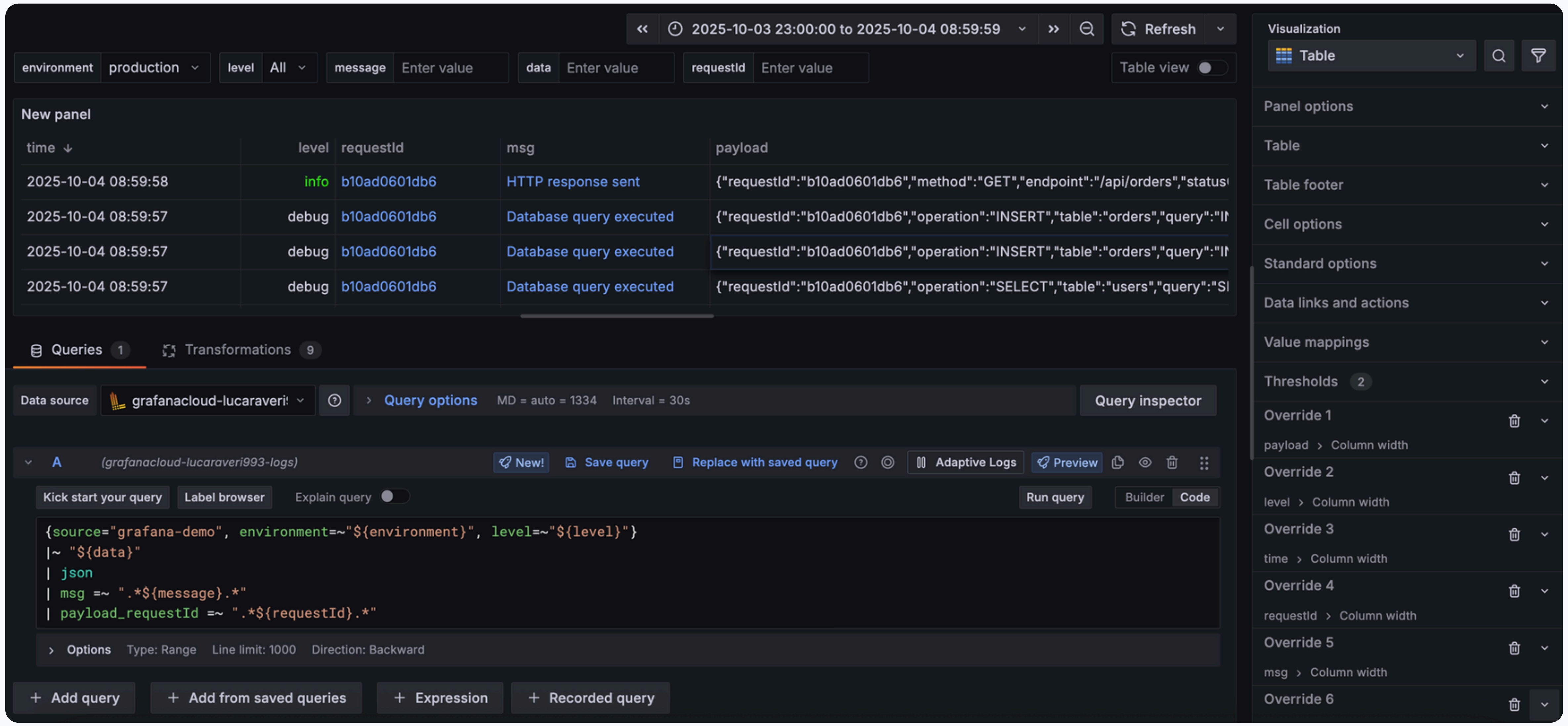


```
{job="apache_error_log", environment=~"${environment}", source=~"${source}"}
|~ "${data}"
| regexp "\\[(?P<timestamp>[^\\]]+)\\].*\\[php7:(?P<level>[^\\]]+)\\].*PHP (?P<type>[^:]+): (?P<message>.*?) in (?
P<file>[A-Z]:\\\\[^\"]+?)(?::(?P<line>\\d+)| on line (?P<line_alt>\\d+))"
| line_format "{{ printf `{\"timestamp\":\"%s\", \"level\":\"%s\", \"type\":\"%s\", \"message\":\"%s\",
\"file\":\"%s\", \"line\":\"%s\"}` .timestamp .level .type .message .file (or .line .line_alt) }}"
| message =~ ".*${message}.*"
| file =~ ".*${file}.*"
| message != ""
```

Talentware

# How to Visualize Logs

1. Write the query
2. Apply transformations
3. Create variables
4. Add overrides



| time ↓ | level | requestId | msg | payload |
|---|---|---|---|---|
| 2025-10-04 08:59:58 | info | b10ad0601db6 | HTTP response sent | {"requestId":"b10ad0601db6","method":"GET","endpoint":"/api/orders","status |
| 2025-10-04 08:59:57 | debug | b10ad0601db6 | Database query executed | {"requestId":"b10ad0601db6","operation":"INSERT","table":"orders","query":"IN |
| 2025-10-04 08:59:57 | debug | b10ad0601db6 | Database query executed | {"requestId":"b10ad0601db6","operation":"INSERT","table":"orders","query":"IN |
| 2025-10-04 08:59:57 | debug | b10ad0601db6 | Database query executed | {"requestId":"b10ad0601db6","operation":"SELECT","table":"users","query":"SI |

```
{source="grafana-demo", environment=~"${environment}", level=~"${level}"}
|~ "${data}"
| json
| msg =~ ".*${message}.*"
| payload_requestId =~ ".*${requestId}.*"
```

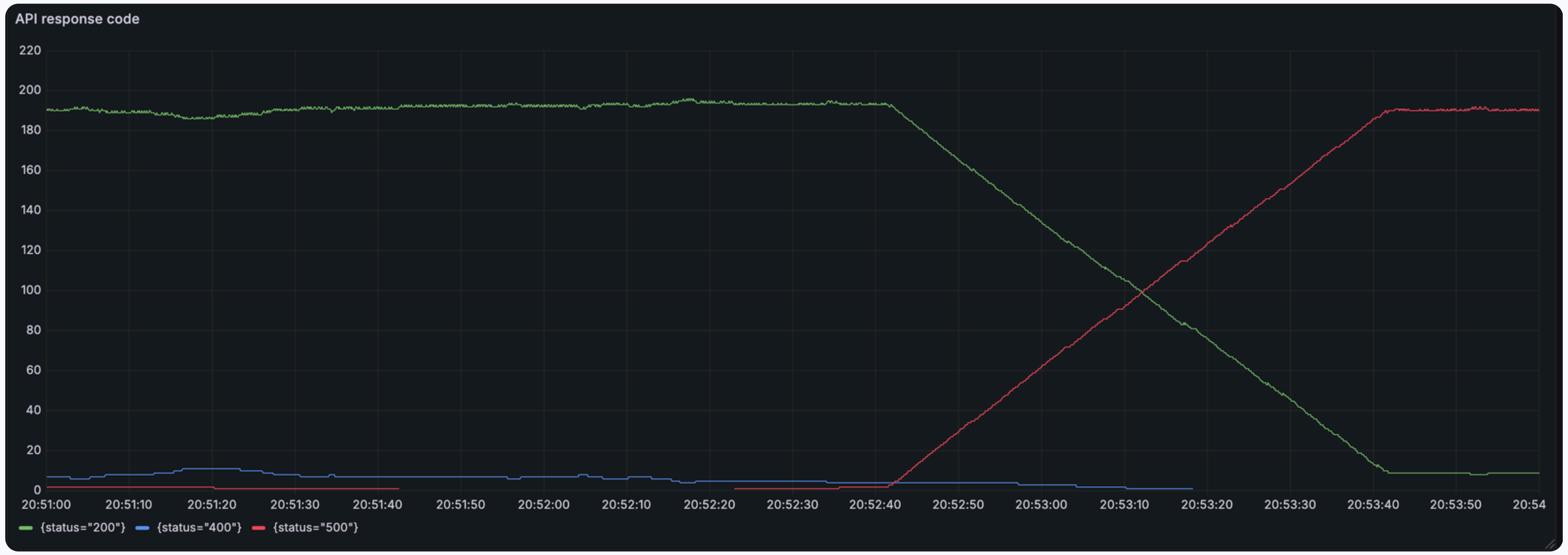
Options Type: Range Line limit: 1000 Direction: Backward

Override 1: payload > Column width
Override 2: level > Column width
Override 3: time > Column width
Override 4: requestId > Column width
Override 5: msg > Column width
Override 6

Talentware

# Alerting Strategies

- Single error detection → trigger on critical errors
- Error rate spikes (RED method) → alert on sudden increase in errors



Talentware

## Results

- Strong team adoption and enthusiasm.
- Significantly improved **MTTD** and reduced **MTTR** from 2 days to 2 hours, enhancing service quality.
- Handled 1 GB/day with 2-week retention (14 GB total), running smoothly on a €6/month instance.

## Thank you!

## Questions?

Talentware

# Demo Repository

Talentware