



# Prometheus at Prezi

## Replacing 10 years of anti-patterns



**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich





# What's **Prezi**?





# What's **Prezi**?



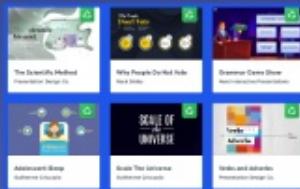
Create presentations



# What's **Prezi**?



Create presentations



Browse presentations

# Our scale



# Our scale

100+ million users



# Our scale

100+ million users

~15 engineering teams



# Our scale

100+ million users

~15 engineering teams

~100 micro services

 python™  Scala  GO



# Our scale

100+ million users

~15 engineering teams

~100 micro services



Web, desktop & mobile clients



# Monitoring...



# Monitoring...

10 years of  
organic build up



Lack of best  
practices



# Monitoring...

10 years of  
organic build up

+

Lack of best  
practices



Services without dashboards or alerts  
**Thousands of alerts**  
**Alert fatigue**  
Cause based alerts  
**Unstable**  
Many moving parts  
High maintenance cost  
Not ready for Kubernetes  
**Log based**  
Icinga (nagios) based  
**1 minute resolution**  
**2+ minutes of delay**  
Poor availability  
Very slow dashboards

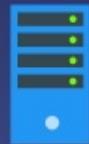


# How did we get there?

*Our previous architecture, from service to alert*

# How did we get there?

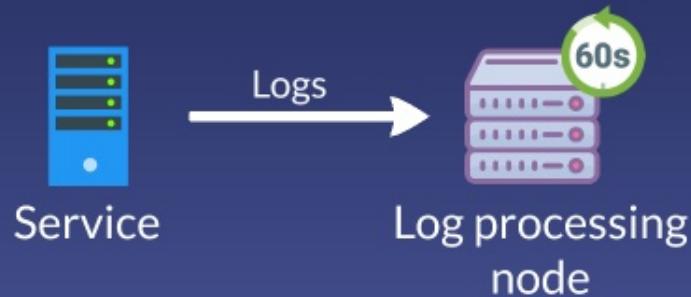
*Our previous architecture, from service to alert*



Service

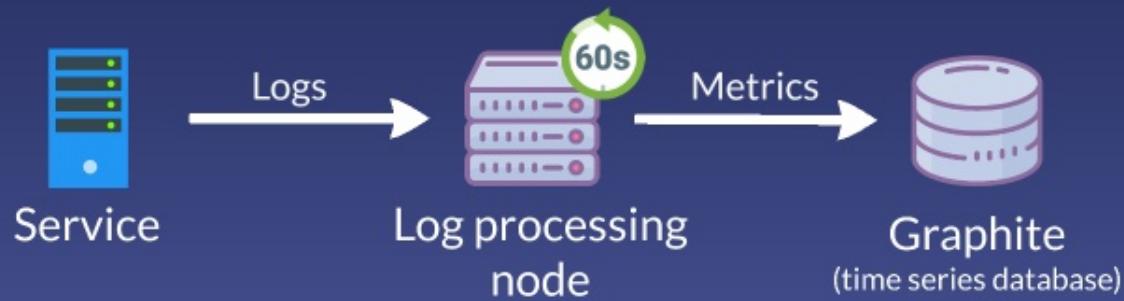
# How did we get there?

*Our previous architecture, from service to alert*



# How did we get there?

*Our previous architecture, from service to alert*



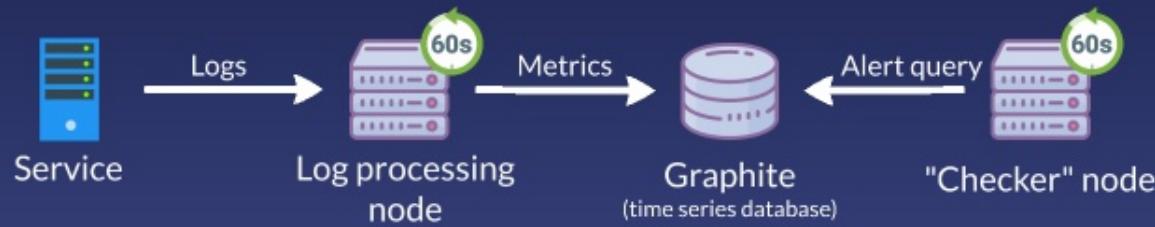
# How did we get there?

*Our previous architecture, from service to alert*



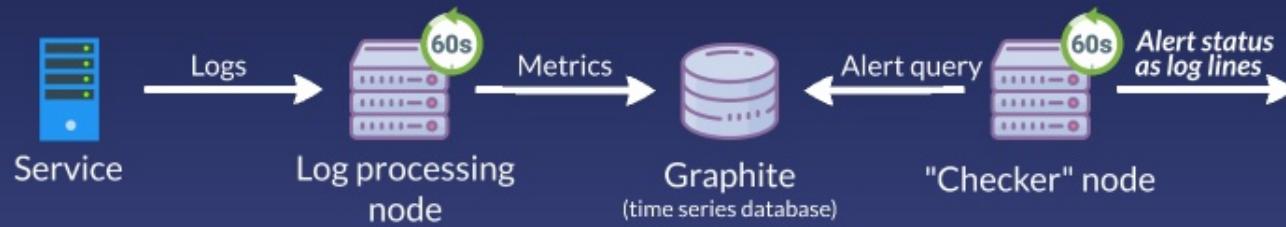
# How did we get there?

*Our previous architecture, from service to alert*



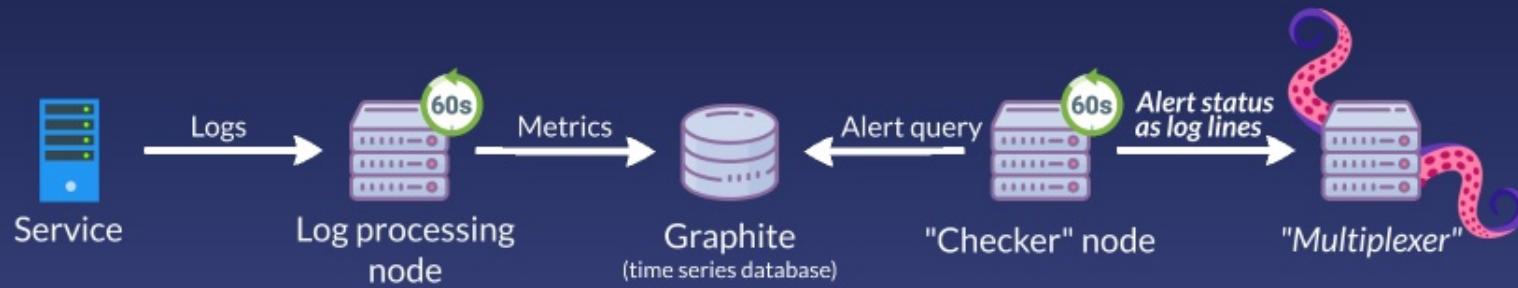
# How did we get there?

*Our previous architecture, from service to alert*



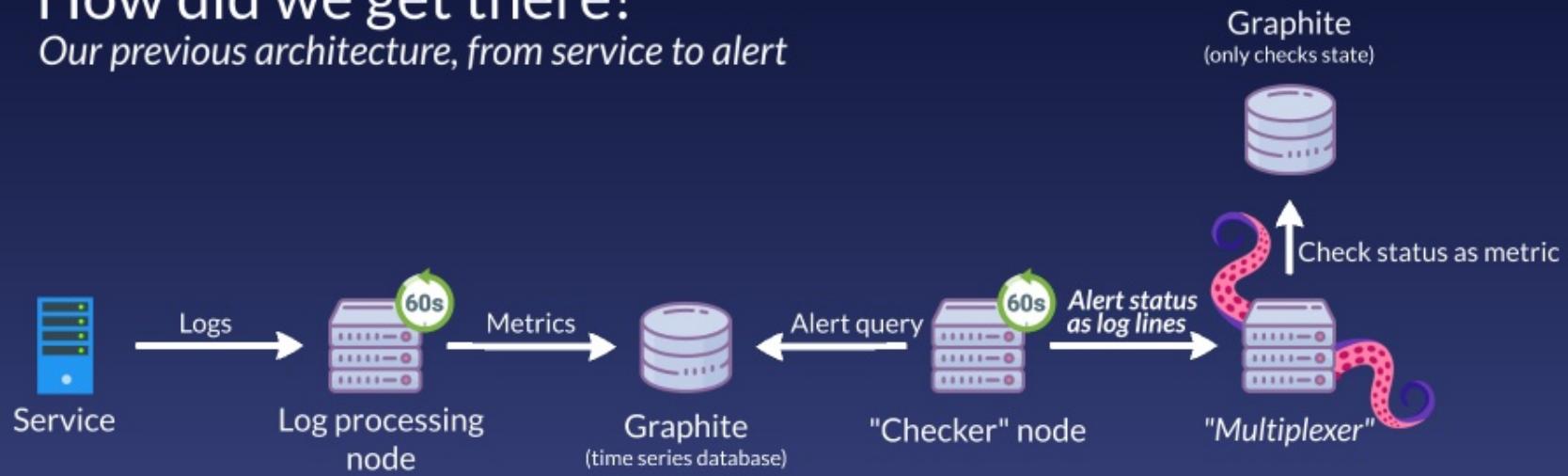
# How did we get there?

*Our previous architecture, from service to alert*



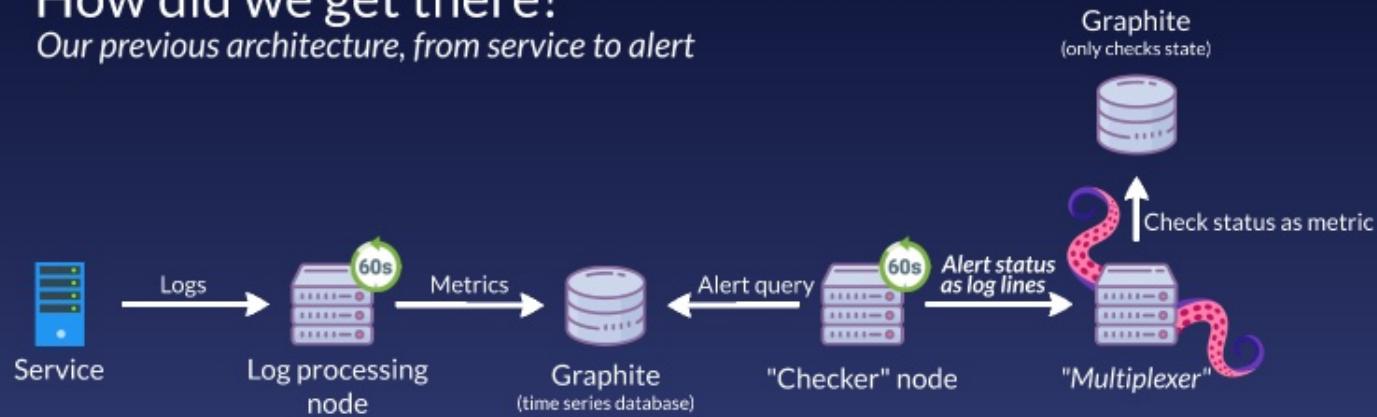
# How did we get there?

*Our previous architecture, from service to alert*



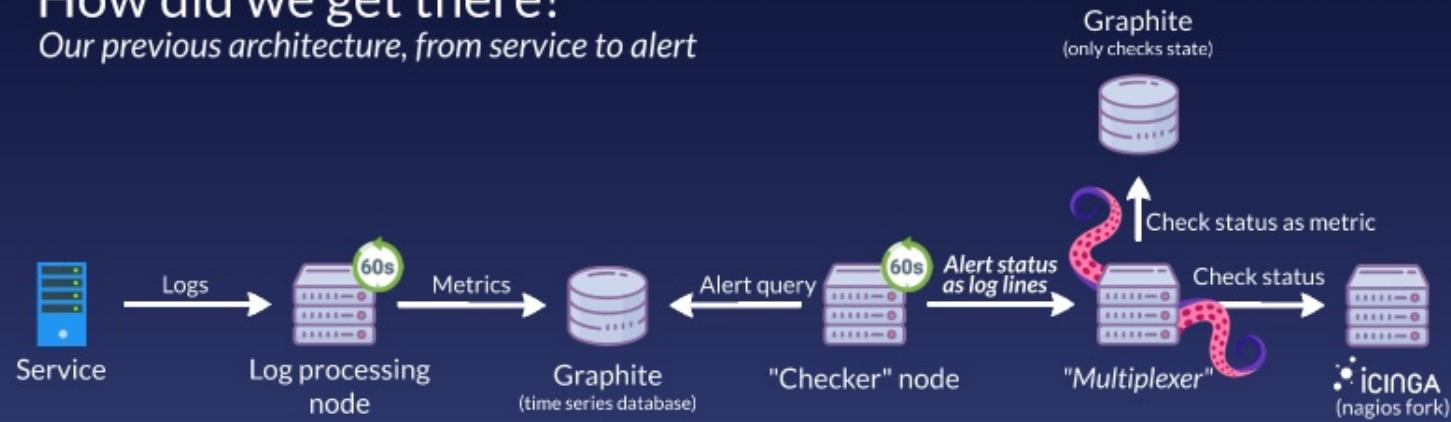
# How did we get there?

*Our previous architecture, from service to alert*



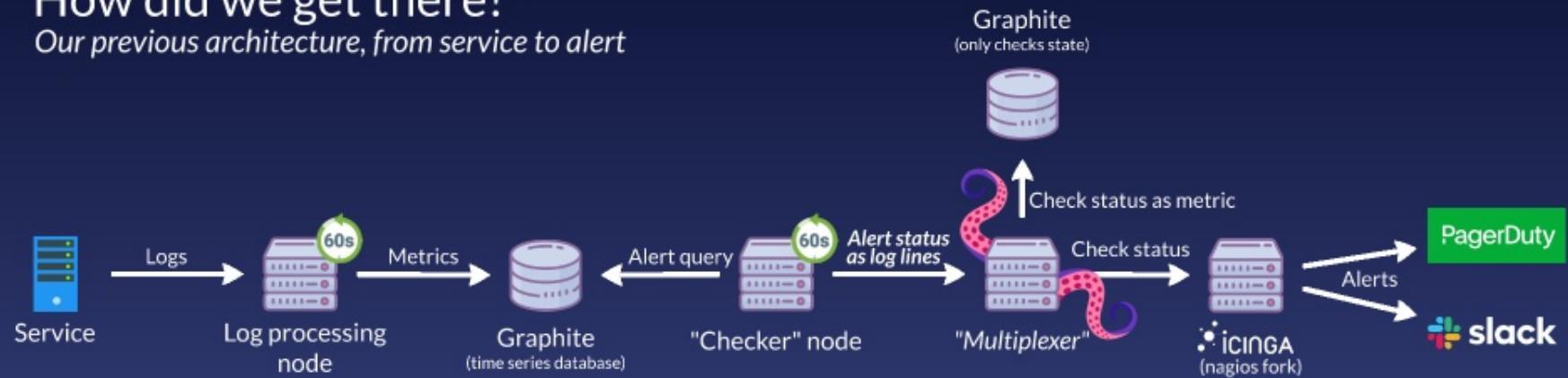
# How did we get there?

*Our previous architecture, from service to alert*



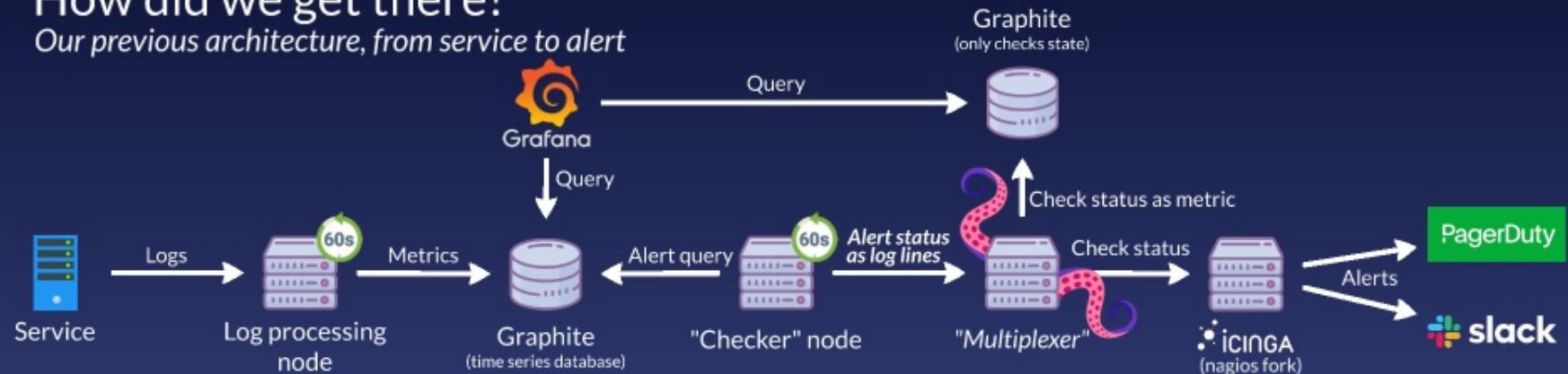
# How did we get there?

*Our previous architecture, from service to alert*



# How did we get there?

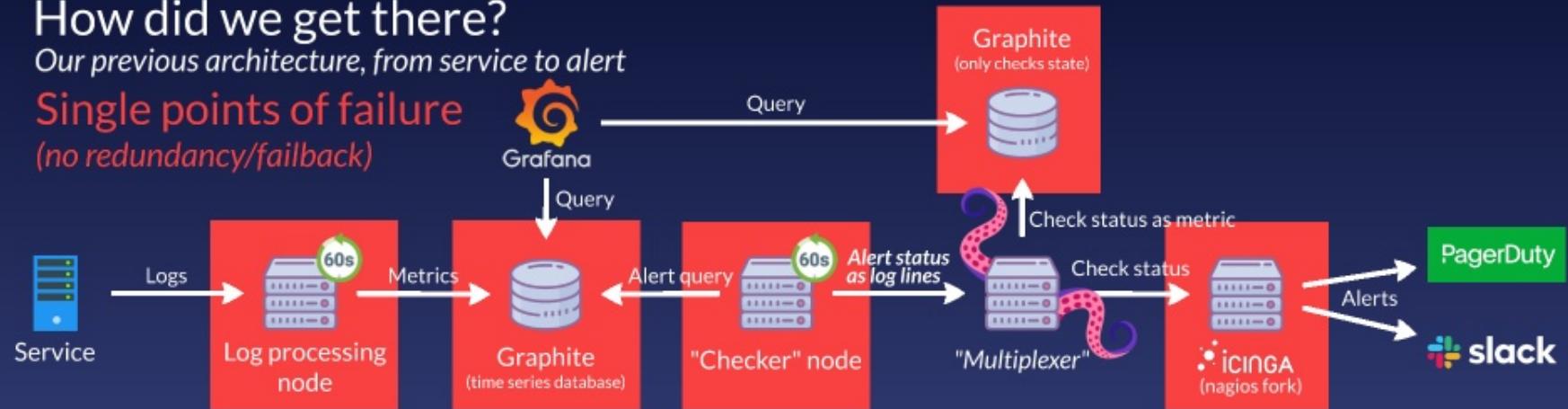
*Our previous architecture, from service to alert*



# How did we get there?

*Our previous architecture, from service to alert*

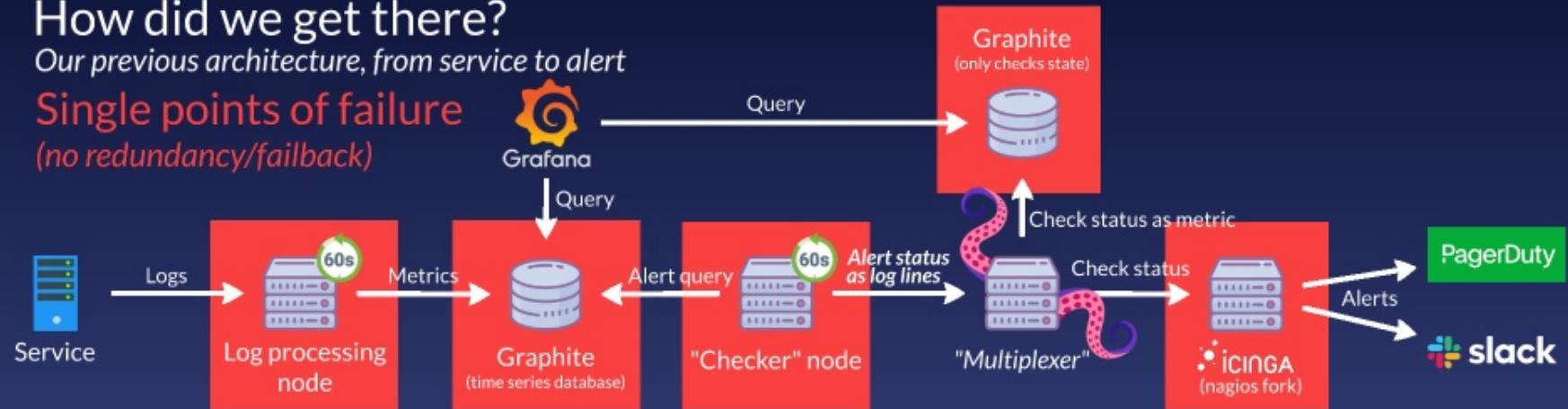
**Single points of failure**  
*(no redundancy/fallback)*



## How did we get there?

*Our previous architecture, from service to alert*

**Single points of failure**  
*(no redundancy/fallback)*

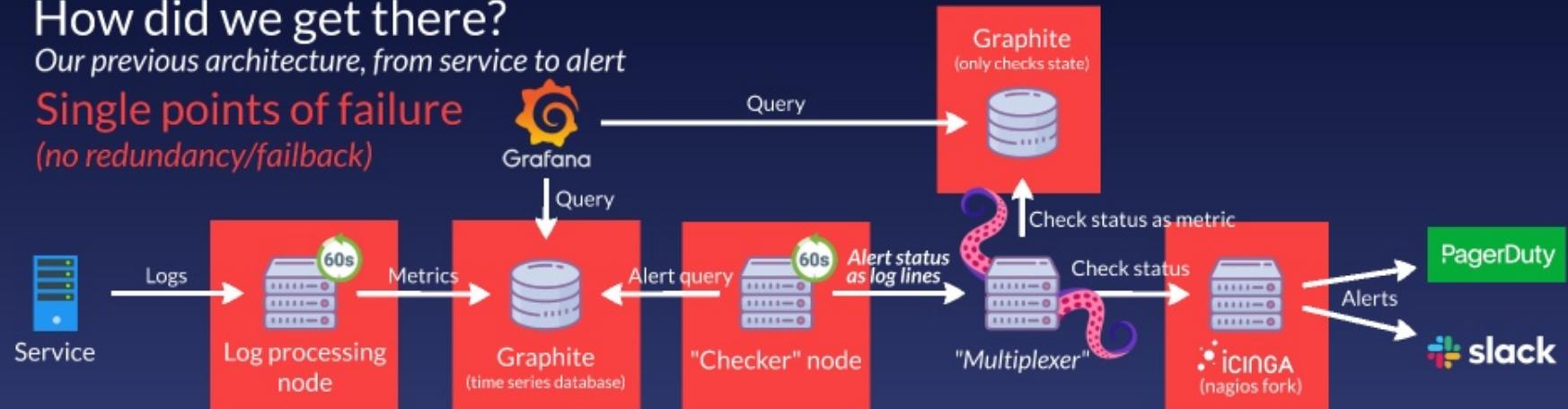


**New generation**  
High availability setup

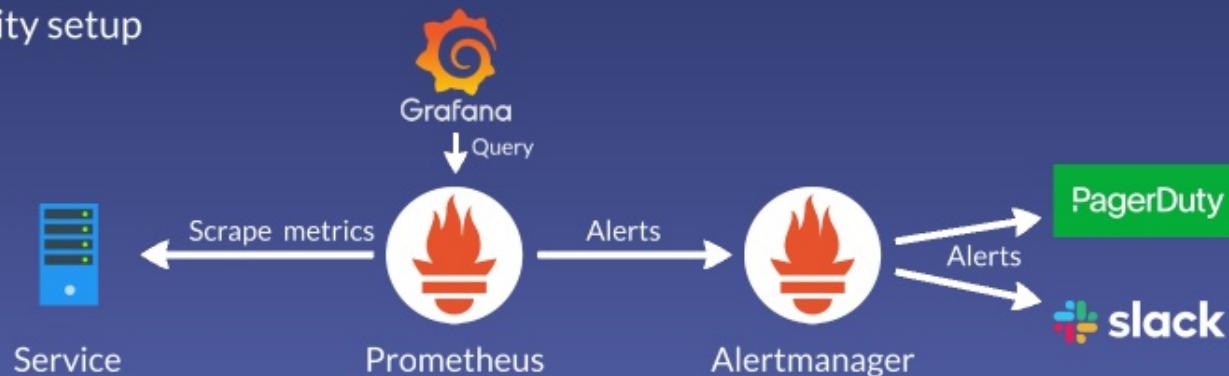
## How did we get there?

*Our previous architecture, from service to alert*

**Single points of failure**  
*(no redundancy/fallback)*



**New generation**  
High availability setup



# Lack of best practices

# Lack of best practices



No dashboard guidelines

# Lack of best practices



No dashboard guidelines



No alerting guidelines



# Prometheus at Prezi

## Replacing 10 years of anti-patterns



**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich





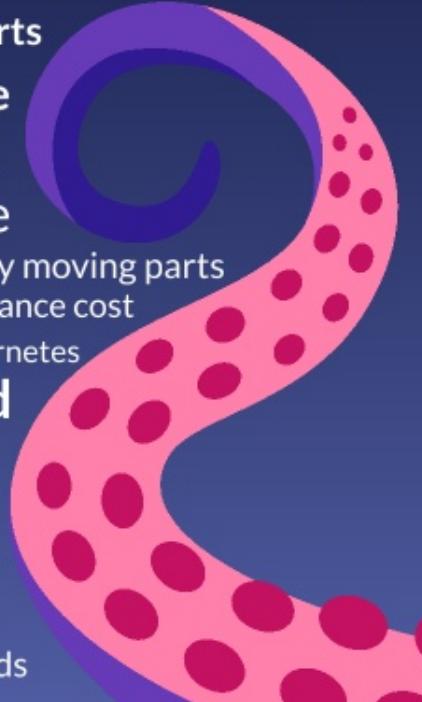
# Selling Prometheus

- ▶ Set clear goals
- ▶ Experience and planning
- ▶ Measure performance

► Set clear goals

# ► Set clear goals

Services without dashboards or alerts  
**Thousands of alerts**  
**Alert fatigue**  
Cause based alerts  
**Unstable**  
Many moving parts  
High maintenance cost  
Not ready for Kubernetes  
**Log based**  
Icinga (nagios) based  
**1 minute resolution**  
**2+ minutes of delay**  
Poor availability  
Very slow dashboards



# ► Set clear goals



## Alerting quality



# ► Set clear goals



## Alerting quality



## User experience



Services without dashboards or alerts  
**Thousands of alerts**  
**Alert fatigue**  
Cause based alerts  
Unstable  
Many moving parts  
High maintenance cost  
Not ready for Kubernetes  
**Log based**  
Icinga (nagios) based  
**1 minute resolution**  
**2+ minutes of delay**  
Poor availability  
Very slow dashboards

# ► Set clear goals



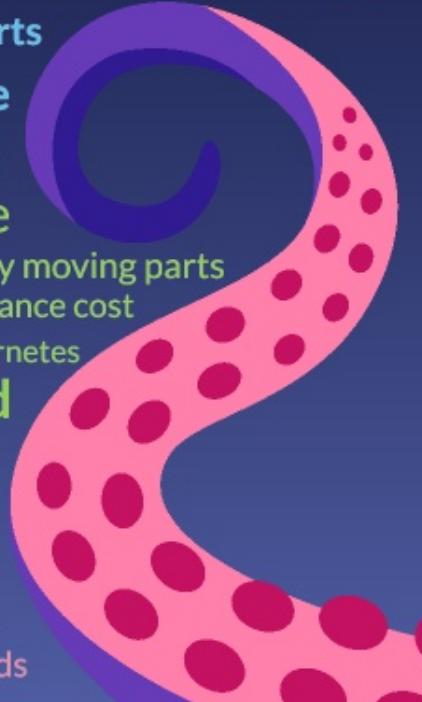
**Alerting quality**



**User experience**



**Uniformity and  
future proofing**



Services without dashboards or alerts  
**Thousands of alerts**  
**Alert fatigue**  
Cause based alerts  
**Unstable**  
Many moving parts  
High maintenance cost  
Not ready for Kubernetes  
**Log based**  
Icinga (nagios) based  
**1 minute resolution**  
**2+ minutes of delay**  
Poor availability  
Very slow dashboards

# ► Experience and planning

# ► Experience and planning



# ► Experience and planning



**Champion**

Have an expert with a vision



**Experiment**

Limited experiment on our CI in 2017

# ► Experience and planning



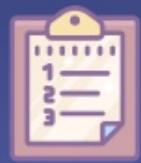
## Champion

Have an expert with a vision



## Experiment

Limited experiment on our CI in 2017



## Scope & plan

Estimated 6 months to transition

Full time for 5 engineers



# ► Measure performance

The  **Alerting quality** example

► Identify key metrics

► Instrument

► Measure



# Identify key metrics

# ► Identify key metrics

— Total pages

# ► Identify key metrics



Total pages

Acknowledgement rate

Neither ignored nor auto-resolved

# ► Identify key metrics



Total pages

Acknowledgement rate

Neither ignored nor auto-resolved

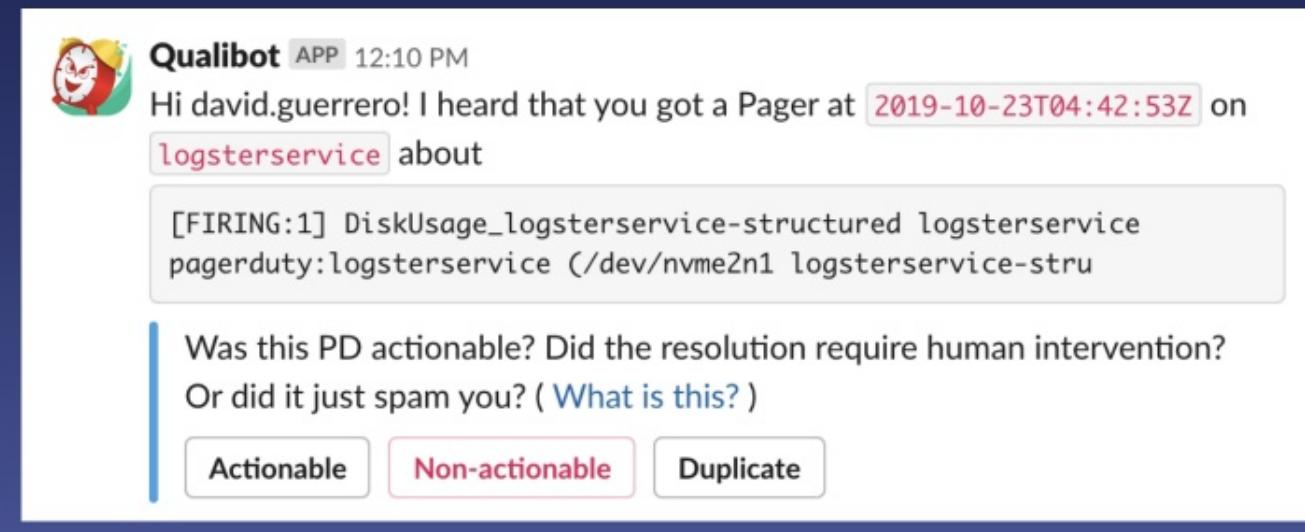
Actionable rate

Real and requires urgent human intervention



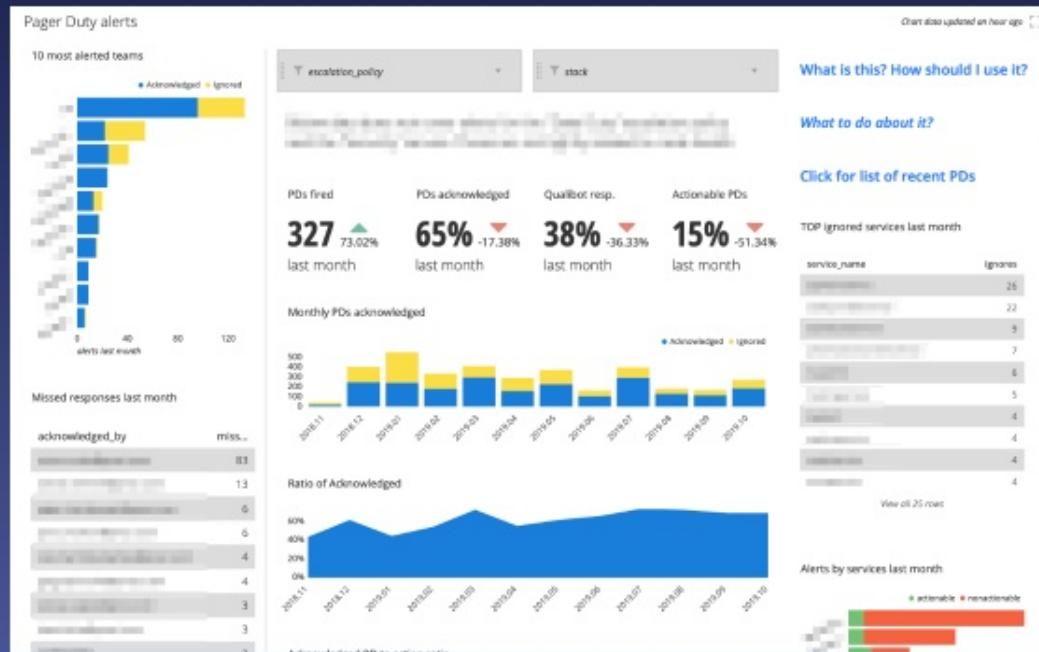
# Instrument

# ► Instrument



# ► Measure

# ► Measure



# ► Measure performance

## The Alerting quality example

### ► Identify key metrics



Total pages

Acknowledgement rate

Neither ignored nor auto-resolved

Actionable rate

Real and requires urgent human intervention

### ► Instrument



### ► Measure





# Prometheus at Prezi

## Replacing 10 years of anti-patterns



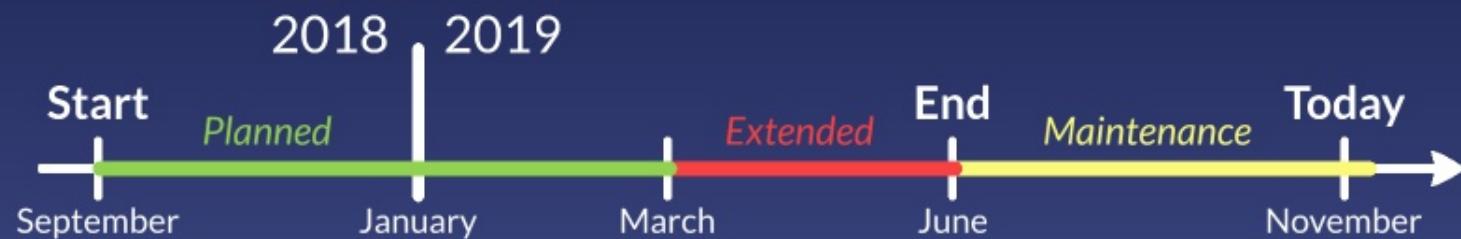
**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich



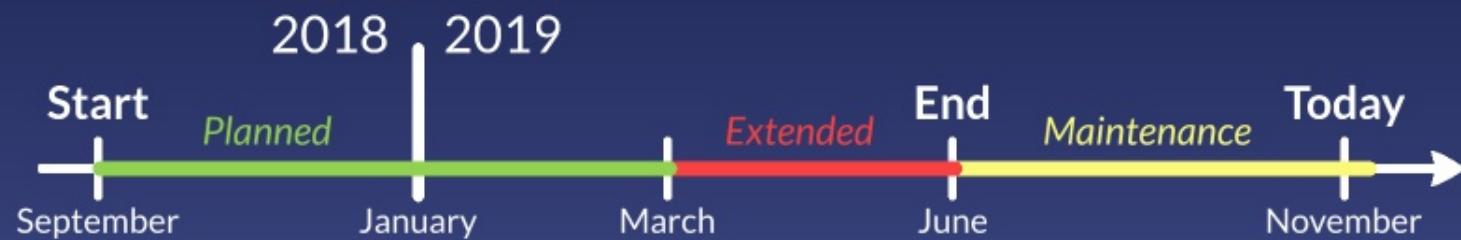
# Replacing 10 years of anti-patterns

## In practice



# Replacing 10 years of anti-patterns

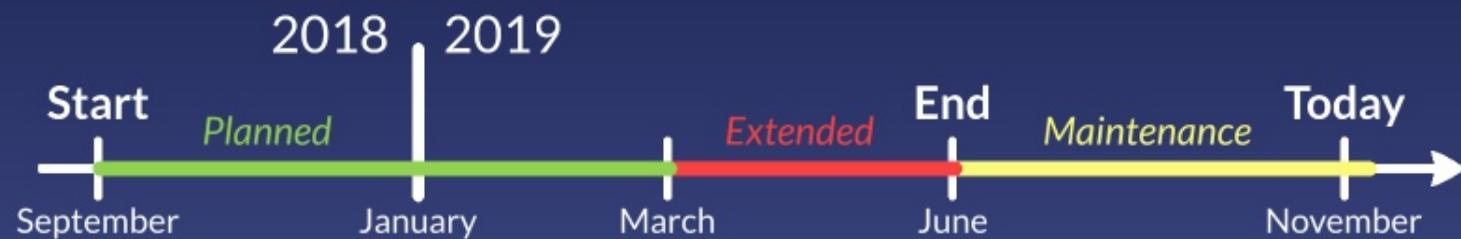
## In practice



## Implementation

# Replacing 10 years of anti-patterns

## In practice



Implementation



Education



# Implementation



# Implementation



Configuration generation

```
src/main/java/com/example/service/HelloService.java
src/main/resources/application.properties
src/main/resources/db/migration/V1__Initial.sql
src/test/java/com/example/service/HelloServiceTest.java
src/test/resources/application.properties
src/test/resources/db/migration/V1__InitialTest.sql
```

```
service_name: fold erservice

configs:
- prometheus_stack: prometheus-classic
  monitoring:
    - include: django
    - include: node
  alerting:
    - environment_name: fold erservice-prod
      groups:
        - default-security
      alerts:
        - include: http/status_ratio
          overrides:
            threshold: 0.2
            target: pagerduty

  alerting_targets:
    - name: pagerduty
      type: pagerduty
      target: fold erservice
```

```
service_name: folderservice

configs:
- prometheus_stack: prometheus-classic
  monitoring:
    - include: django
    - include: node
  alerting:
    - environment_name: folderservice-prod
      groups:
        - default-security
      alerts:
        - include: http/status_ratio
          overrides:
            threshold: 0.2
            target: pagerduty

  alerting_targets:
    - name: pagerduty
      type: pagerduty
      target: folderservice
```

Scraping config

```
service_name: fold erservice

configs:
- prometheus_stack: prometheus-classic
  monitoring:
    - include: django
    - include: node
  alerting:
    - environment_name: fold erservice-prod
      groups:
        - default-security
      alerts:
        - include: http/status_ratio
          overrides:
            threshold: 0.2
            target: pagerduty

  alerting_targets:
    - name: pagerduty
      type: pagerduty
      target: fold erservice
```



Scraping config



Group of alerts

```
service_name: fold erservice

configs:
- prometheus_stack: prometheus-classic
  monitoring:
    - include: django
    - include: node
  alerting:
    - environment_name: fold erservice-prod
      groups:
        - default-security
      alerts:
        - include: http/status_ratio
          overrides:
            threshold: 0.2
            target: pagerduty

  alerting_targets:
- name: pagerduty
  type: pagerduty
  target: fold erservice
```

Scraping config

Group of alerts

Alert template

Threshold override

```
service_name: fold erservice

configs:
- prometheus_stack: prometheus-classic
  monitoring:
    - include: django
    - include: node
  alerting:
    - environment_name: fold erservice-prod
    groups:
      - default-security
    alerts:
      - include: http/status_ratio
        overrides:
          threshold: 0.2
          target: pagerduty
    alerting_targets:
      - name: pagerduty
        type: pagerduty
        target: fold erservice
```

Scraping config

Group of alerts

Alert template

Threshold override

Target management



# Implementation



Configuration generation





# Implementation

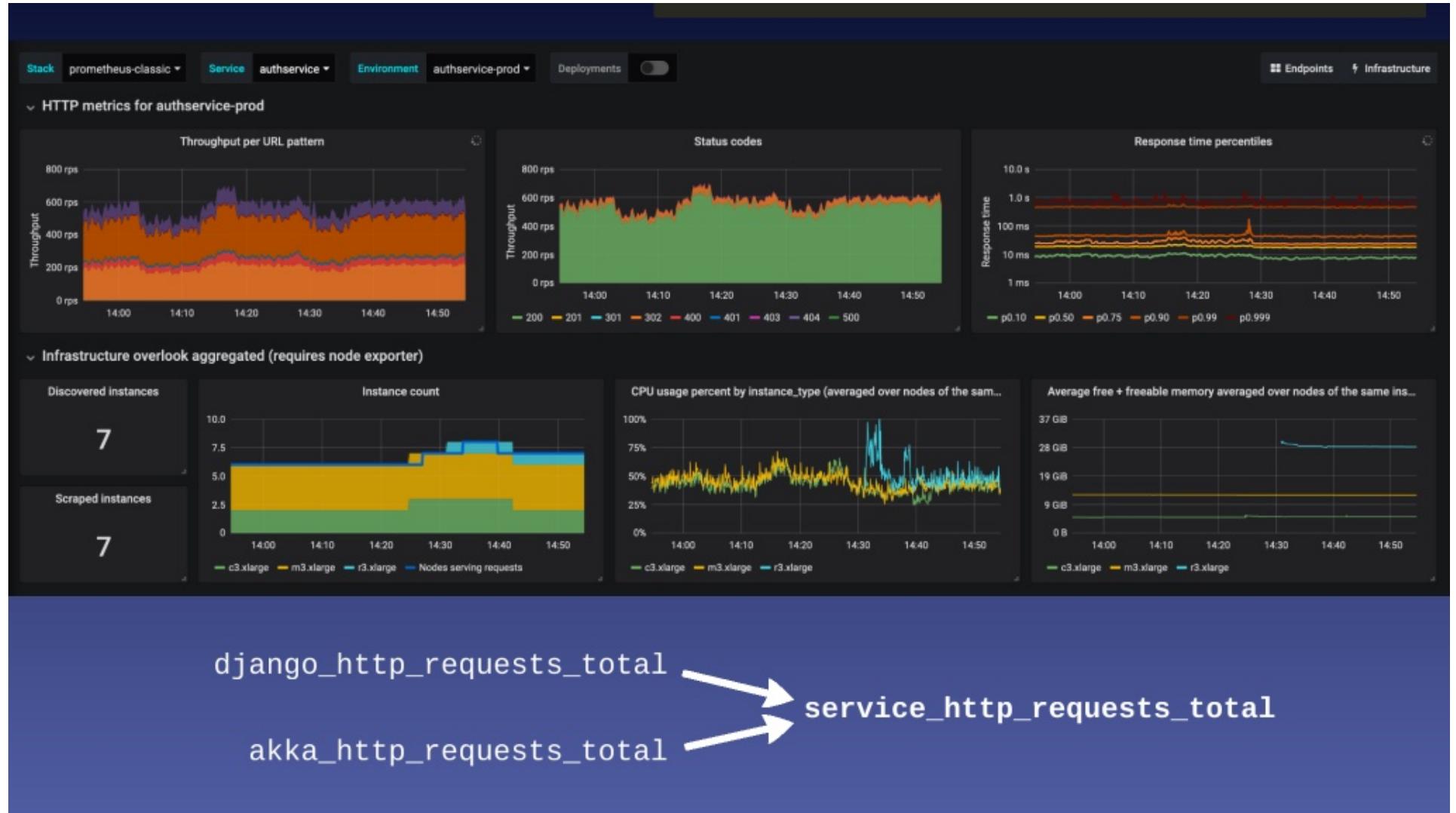


Configuration generation



Generic dashboards







# Implementation



Configuration generation



Generic dashboards





# Implementation

 Configuration generation



 Generic dashboards



 Migration tooling



# Implementation



Configuration generation



Generic dashboards



Migration tooling



Django/Python  
prezi/django-exporter



# Implementation



Configuration generation



Generic dashboards



Migration tooling



Django/Python  
prezi/django-exporter



Client side monitoring



# Implementation



Configuration generation



Generic dashboards



Migration tooling



Django/Python  
prezi/django-exporter



Client side monitoring



Retention & scaling



# Education



# Education



"My Philosophy On Alerting"



# Education



"My Philosophy On Alerting"



Presentations & sessions



# Education

✓ "My Philosophy On Alerting"

✓ Presentations & sessions

✓ Continuous support



# Education

-  "My Philosophy On Alerting"
-  Presentations & sessions
-  Continuous support
-  Extensive documentation



# Education



"My Philosophy On Alerting"



Presentations & sessions



Continuous support



Extensive documentation



Attachment to old alerts



# Education



"My Philosophy On Alerting"



Presentations & sessions



Continuous support



Extensive documentation



Attachment to old alerts



Lack of awareness

It could be much better!



# Education



"My Philosophy On Alerting"



Presentations & sessions



Continuous support



Extensive documentation

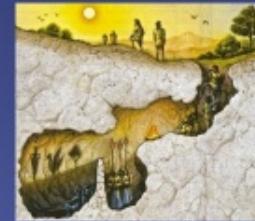


Attachment to old alerts



Lack of awareness

It could be much better!



*Plato's allegory of the cave*



# Lack of awareness

It could be much better!



*Plato's allegory of the cave*



# Prometheus at Prezi

## Replacing 10 years of anti-patterns



**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich





# Prometheus at Prezi



User experience  
& uniformity



Alerting quality



Key takeaways



# User experience & uniformity



# User experience & uniformity



Simple setup / configuration



# User experience & uniformity

- ✓ Simple setup / configuration
- ✓ Generic dashboards



# User experience & uniformity

- ✓ Simple setup / configuration
- ✓ Generic dashboards
- ✓ Monitoring as a platform



# User experience & uniformity

- Simple setup / configuration
- Generic dashboards
- Monitoring as a platform
- Real time dashboards & alerts



# User experience & uniformity

95% of use cases

- Simple setup / configuration
- Generic dashboards
- Monitoring as a platform
- Real time dashboards & alerts



# User experience & uniformity

95% of use cases

Remaining 5%

-  Simple setup / configuration
-  Generic dashboards
-  Monitoring as a platform
-  Real time dashboards & alerts



# User experience & uniformity

95% of use cases

- Simple setup / configuration
- Generic dashboards
- Monitoring as a platform
- Real time dashboards & alerts

Remaining 5%

- Anything custom is difficult



# Alerting quality

On-call experience improvements



# Alerting quality

## On-call experience improvements



Total pages

Acknowledgement rate

Neither ignored nor auto-resolved

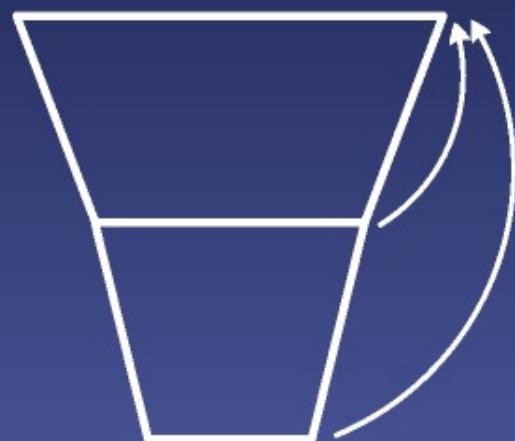
Actionable rate

Real and requires urgent human intervention



# Alerting quality

On-call experience improvements



Total pages

-50%

Acknowledgement rate

Neither ignored nor auto-resolved

Actionable rate

Real and requires urgent human intervention



# Alerting quality

## On-call experience improvements



Total pages

-50%

Acknowledgement rate  
Neither ignored nor auto-resolved

x2

Actionable rate  
Real and requires urgent human intervention



# Alerting quality

## On-call experience improvements



Total pages

-50%

Acknowledgement rate  
Neither ignored nor auto-resolved

x2

Actionable rate  
Real and requires urgent human intervention

x2



# Key takeaways



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



Education is a constant effort



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



Education is a constant effort



Selling the project is part of it



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



Education is a constant effort



Selling the project is part of it



**Great success!**



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



Education is a constant effort



Selling the project is part of it



Great success!



Open source  
contributions



# Key takeaways



Migrating and **cleaning up** is a lot of effort

Still not 100% done



Education is a constant effort



Selling the project is part of it



Great success!



Open source  
contributions



Easier K8s  
migration



# Prometheus at Prezi

## Replacing 10 years of anti-patterns



**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich



## David Guerrero

Site Reliability Engineer

david.guerrero@prezi.com

heldroe@gmail.com

<https://linkedin.com/in/heldroe/en>

<https://github.com/heldroe>



# Prometheus at Prezi

## Thank you for listening!



<https://prezi.com/jobs>



<https://github.com/prezi/django-exporter>



<https://engineering.prezi.com/prometheus-unchained-331b7ab8737>

### "My Philosophy on Alerting"

<https://docs.google.com/document/d/199PqyG3UsyXlwieHaqbGiWVa8eMWi8zzAn0YfcApr8Q/edit>

This presentation



Dashboard  
browsing demo

00:00:00.000

Before

After

**00:00:00.000**

**Before**

**After**



# Prometheus at Prezi

## Replacing 10 years of anti-patterns



**David Guerrero**  
Site Reliability Engineer  
[david.guerrero@prezi.com](mailto:david.guerrero@prezi.com)

PromCon EU 2019  
November 08  
Munich

