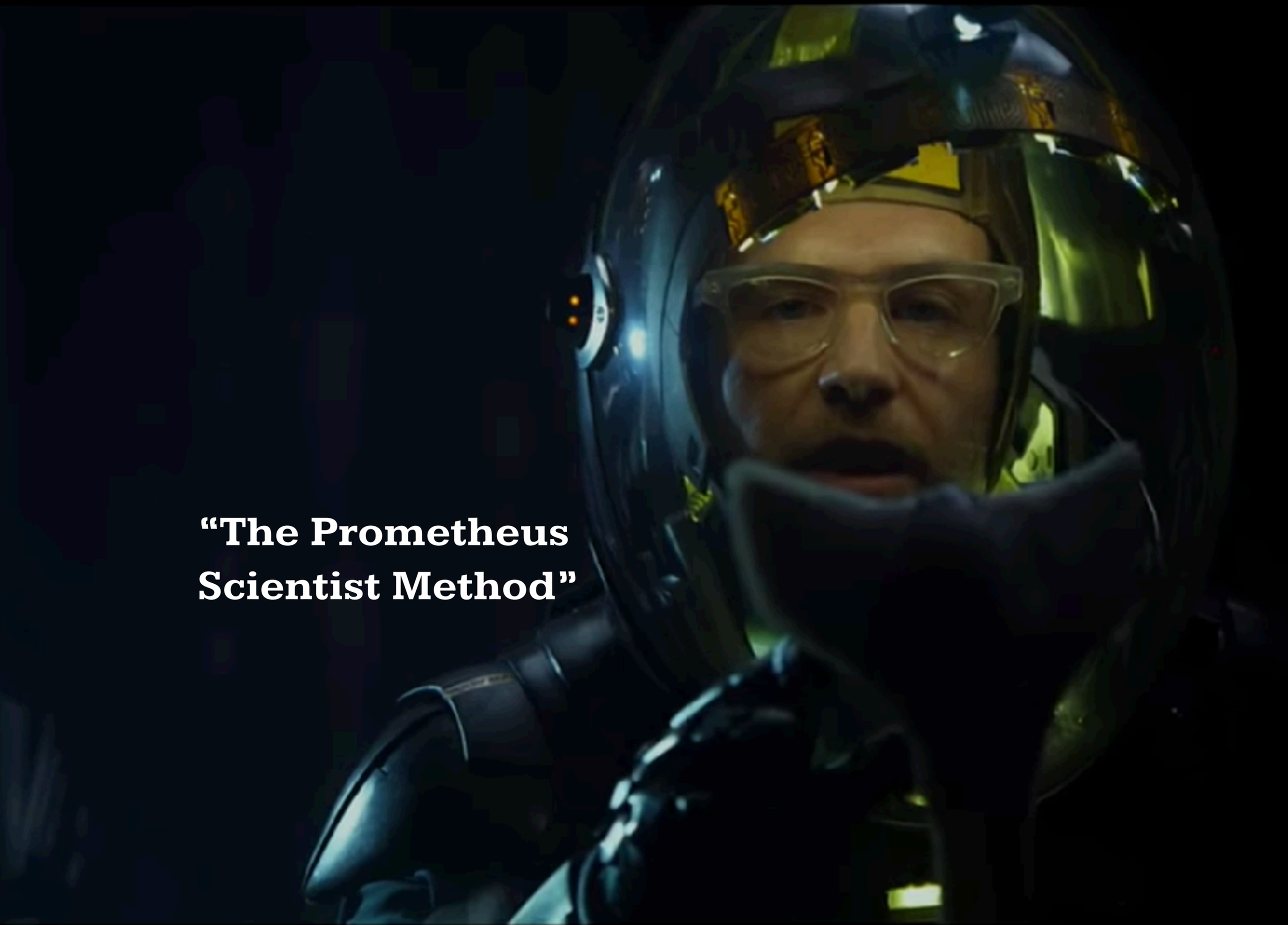
Application Metrics

with Prometheus

Rafael Dohms @rdohms

How do *you* do metrics?


“The Prometheus
Scientist Method”

I hope not.

Rafael Dohms

Staff Engineer

doh.ms rdohms

usabilla

jobs.usabilla.com

Rafael Dohms

Staff Engineer

doh.ms

rdohms

usabilla

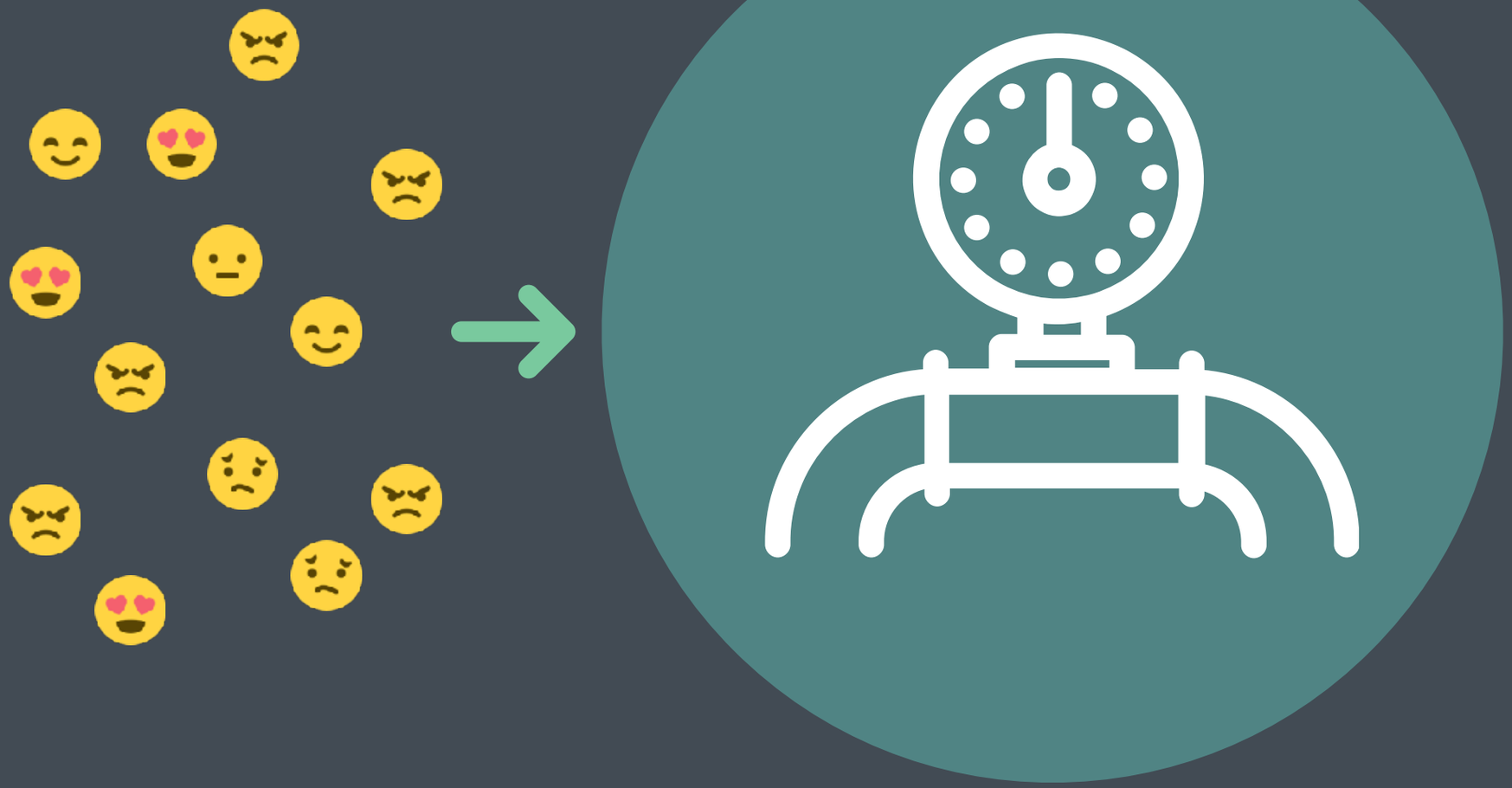
jobs.usabilla.com

Feedback

usabilla

Kafka / DDD / Autonomous Microservices / Monitoring

usabilla



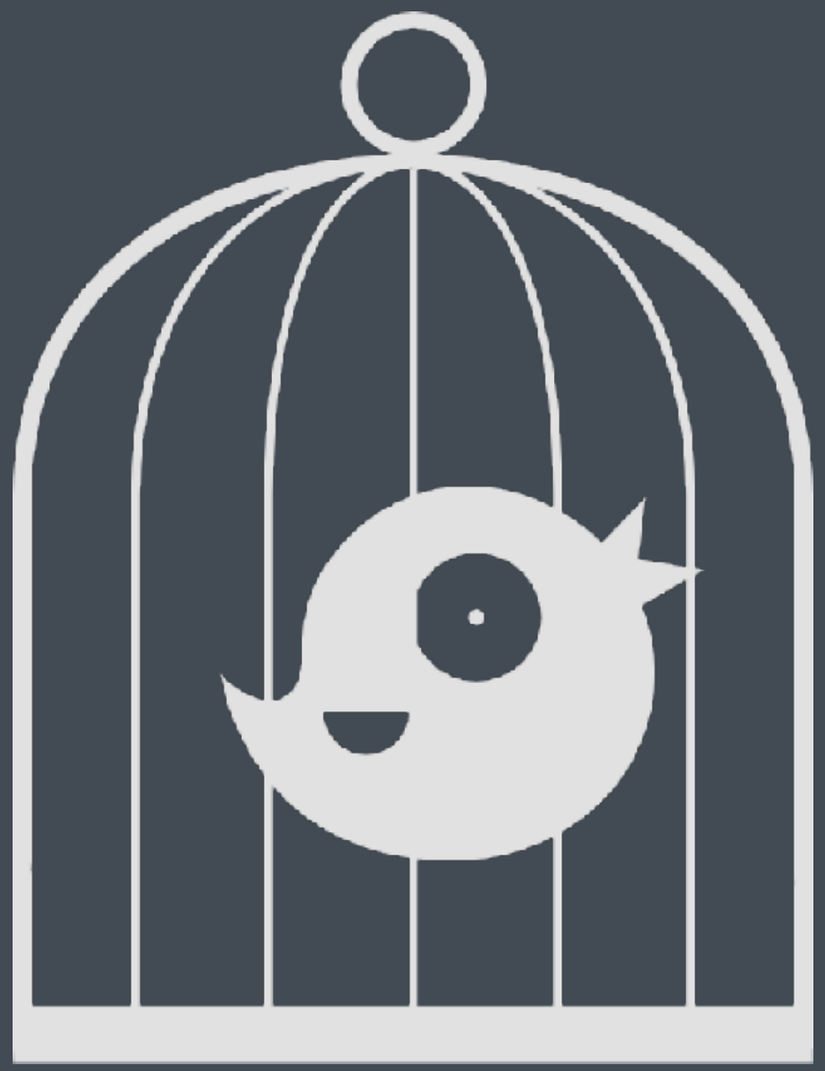
Kafka / DDD / Autonomous Microservices / Monitoring

usabilla

Kafka / DDD / Autonomous Microservices / Monitoring

Metrics are insights into the current state of your application.

Metrics tell you if your service is healthy.



Canary Deploys

Oksana Latysheva

Metrics tell you what is wrong.

Metrics tell you what is right.

Metrics tell you what will soon be wrong.

Metrics tell you where to start looking.

Site Reliability Engineering

SLIs

SLOs

SLAs

Service Level Indicators

“A quantitative measure of some aspect of your application”

The response time of a request was 150ms

Source: Site Reliability Engineering - O’Reilly

Service Level Objectives

“A target value or a range of values for something measured by an SLI”

Request response times should be below 200ms

Source: Site Reliability Engineering - O’Reilly

Help you drive architectural decisions, like optimisation

Response time SLO: 150 ms
95th Percentile of Processing time (PHP time): 5ms

As a result we decided to invest more time in exploring the problem domain and not optimising our stack.

Service Level Agreements

“An explicit or implicit contract with your customer, that includes consequences of missing their SLOs”

The 99th percentile of requests response times should meet our SLO, or we will refund users

Source: Site Reliability Engineering - O’Reilly

Measuring

“If it moves, we track it.”

–Etsy Engineering

https://codeascraft.com/2011/02/15/measure-anything-measure-everything/

Metrics

Telemetry

What is happening right now?

Statistics

How often does this happen?

Telemetry

“the process of recording and transmitting the readings of an instrument”

2016 Driver Standings

POS	DRIVER	NATIONALITY	CAR	PTS
1	Nico Rosberg	GER	MERCEDES	385
2	Lewis Hamilton	GBR	MERCEDES	380
3	Daniel Ricciardo	AUS	RED BULL RACING TAG HEUER	256
4	Sebastian Vettel	GER	FERRARI	212
5	Max Verstappen	NED	RED BULL RACING TAG HEUER	204
6	Kimi Räikkönen	FIN	FERRARI	186
7	Sergio Perez	MEX	FORCE INDIA MERCEDES	101
8	Valtteri Bottas	FIN	WILLIAMS MERCEDES	85
9	Nico Hulkenberg	GER	FORCE INDIA MERCEDES	72
10	Fernando Alonso	ESP	MCLAREN HONDA	54
11	Felipe Massa	BRA	WILLIAMS MERCEDES	53

Statistics / Analytics

“the practice of collecting and analysing numerical data in large quantities”

1991 Driver Standings

POS	DRIVER	NATIONALITY	CAR	PTS
1	Ayrton Senna	BRA	MCLAREN HONDA	96
2	Nigel Mansell	GBR	WILLIAMS RENAULT	72
3	Riccardo Patrese	ITA	WILLIAMS RENAULT	53
4	Gerhard Berger	AUT	MCLAREN HONDA	43
5	Alain Prost	FRA	FERRARI	34
6	Nelson Piquet	BRA	BENETTON FORD	26.5
7	Jean Alesi	FRA	FERRARI	21
8	Stefano Modena	ITA	TYRRELL HONDA	10
9	Andrea de Cesaris	ITA	JORDAN FORD	9
10	Roberto Moreno	BRA	MINARDI FERRARI	8
11	Pierluigi Martini	ITA	MINARDI FERRARI	6

Statistics / Analytics

“the practice of collecting and analysing numerical data in large quantities”

1991 Driver Standings

POS	DRIVER	NATIONALITY	CAR	PTS
1	Ayrton Senna	BRA		96
2	Nigel Mansell	GBR	WILLIAMS RENAULT	72
3	Riccardo Patrese	ITA	WILLIAMS RENAULT	53
4	Gerhard Berger	AUT	MCLAREN HONDA	43
5	Alain Prost	FRA	FERRARI	34
6	Nelson Piquet	BRA	BENETTON FORD	26.5
7	Jean Alesi	FRA	FERRARI	21
8	Stefano Modena	ITA	TYRRELL HONDA	10
9	Andrea de Cesaris	ITA	JORDAN FORD	9
10	Roberto Moreno	BRA	MINARDI FERRARI	8
11	Pierluigi Martini	ITA	MINARDI FERRARI	6

I really miss Ayrton Senna

Statistics / Analytics

“the practice of collecting and analysing numerical data in large quantities”

Telemetry

response time of public endpoints

Statistics

Incoming feedback items with origin information

“If it moves, we track it.”

System Throughput

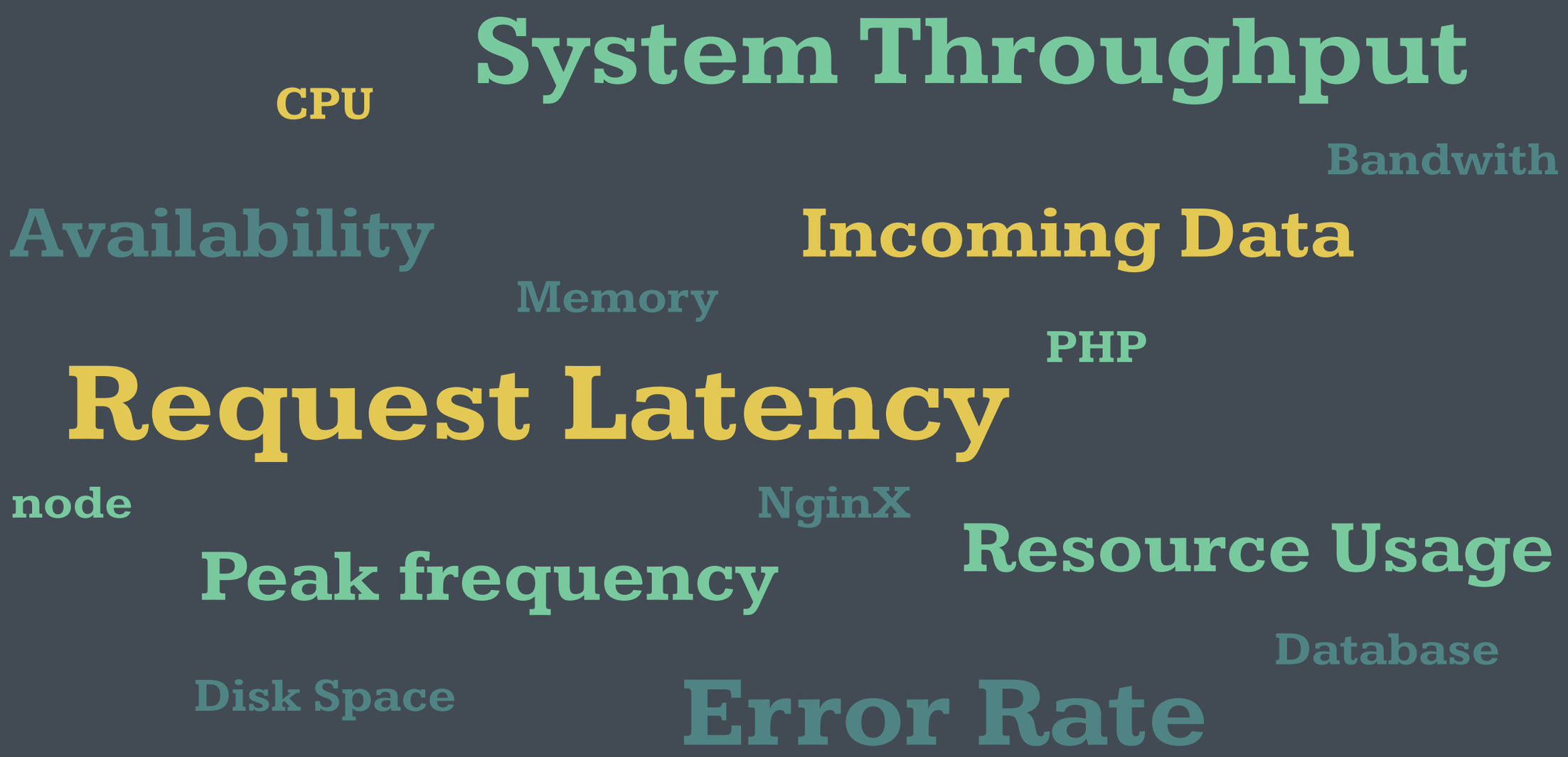
Availability

Request Latency

Resource Usage

Error Rate

“If it moves, we track it.”



“If it moves, we track it.”

Measure Monitoring

System Throughput

CPU

Bandwith

Availability

Incoming Data

Memory

PHP

Request Latency

node

NginX

Resource Usage

Peak frequency

Database

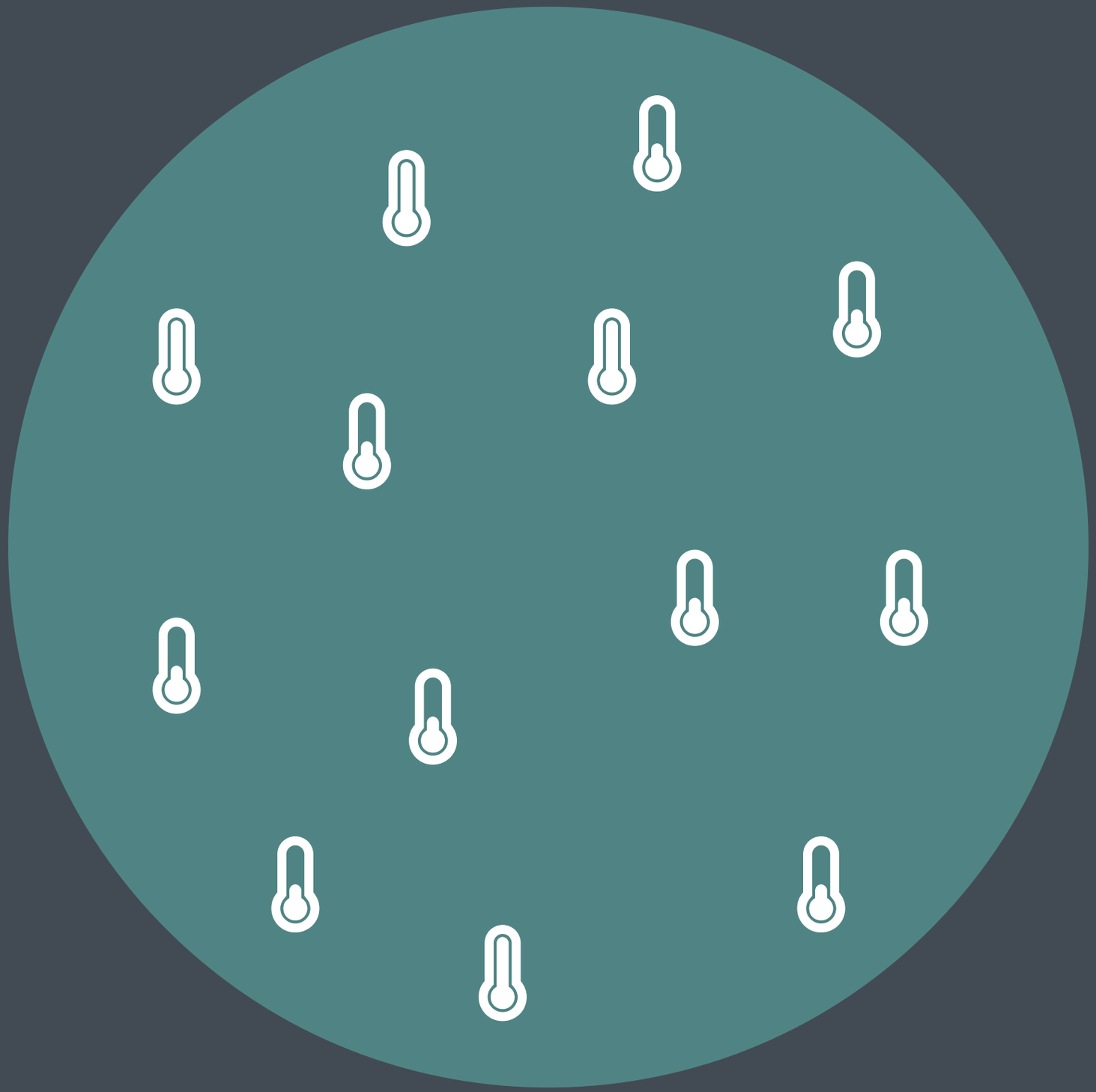
Disk Space

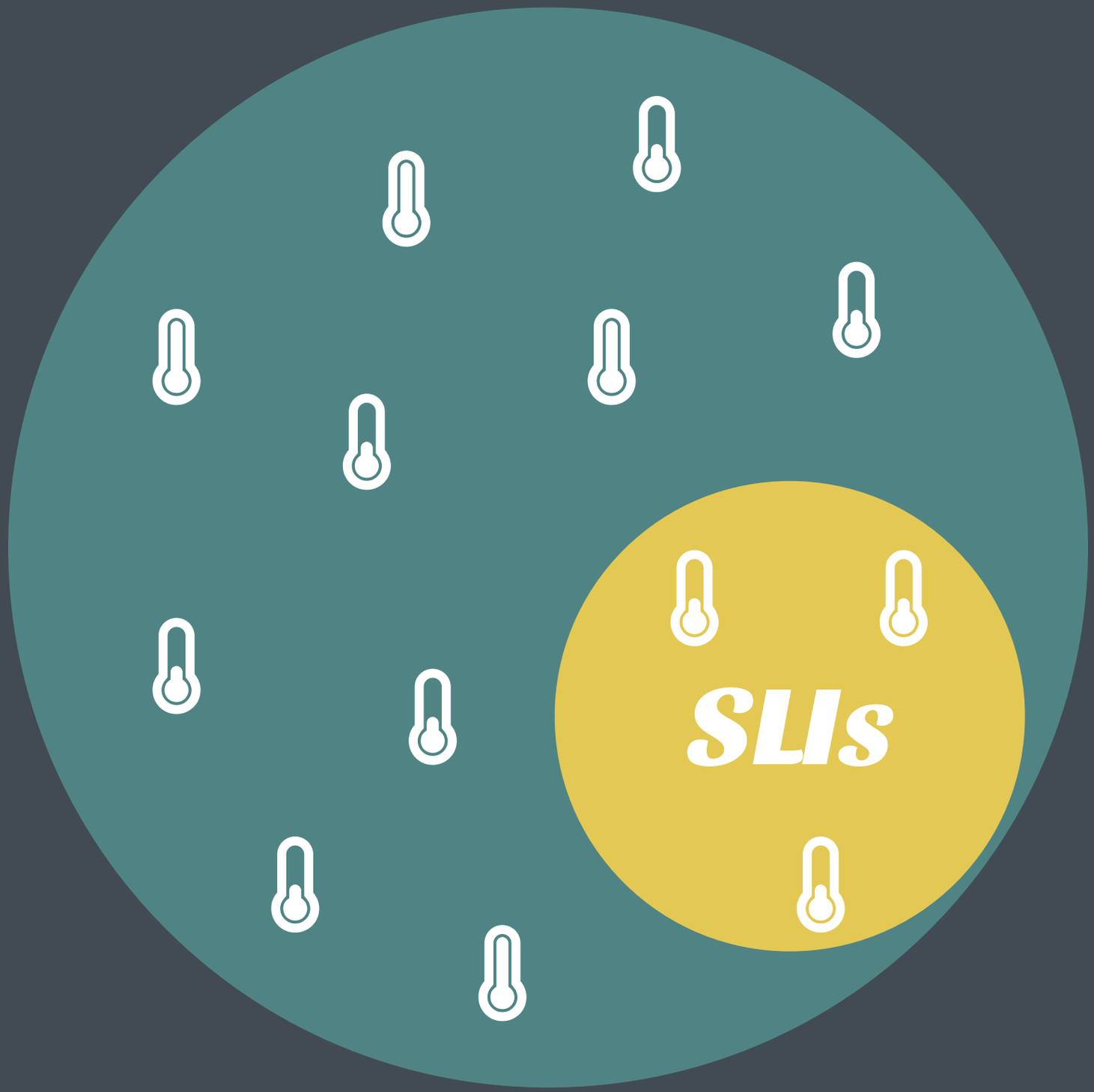
Error Rate

Measure measurements

“If it moves, we track it.”

Metrics, Everywhere.




SLIs

Picking good SLIs

SLIs may change according to who is looking at the data.

Understanding the nature of your system

User-Facing serving system?

availability, throughput, latency

Storage System?

availability, durability, latency

Big Data Systems?

throughput, end-to-end latency

User-Facing and **Big Data** Systems

User-Facing and **Big Data** Systems

- SLIs
  - **Response time** in the “receive” endpoint
  - **Turn around time**, from “receive” to “show”.
  - Individual **processing time per step**
  - Data counting: **how many**, what nature

User-Facing and **Big Data** Systems

More relevant to development team

- SLIs
  - **Response time** in the “receive” endpoint
  - **Turn around time**, from “receive” to “show”.
  - Individual **processing time per step**
  - Data counting: **how many**, what nature

User-Facing and **Big Data** Systems

More relevant to development team

- SLIs
  - **Response time** in the “receive” endpoint
  - **Turn around time**, from “receive” to “show”.
  - Individual **processing time per step**
  - Data counting: **how many**, what nature
- Other Metrics
  - node, nginx, php-fpm, java metrics
  - server metrics: cpu, memory, disk space
  - Size of cluster
  - Kafka health

User-Facing and **Big Data** Systems

- SLIs
  - **Response time** in the “receive” endpoint
  - **Turn around time**, from “receive” to “show”.
  - Individual **processing time per step**

More relevant to development team

- Other Metrics
  - node, nginx, php-fpm, java metrics
  - server metrics: cpu, memory, disk space
  - Size of cluster
  - Kafka health

More relevant to Infrastructure team

Picking Targets

Target value

SLI value >= target

Target Range

lower bound <= SLI value <= upper bound

Don't pick a target based on current performance

What is the business need?
What are users trying to achieve?
How much impact does it have on the user experience?

How long can it take between the user clicking submit and a confirmation that our servers received the data?

How long can it take between the user clicking submit and a confirmation that our servers received the data?

“Immediate"

“We sell as real time”

“500ms, too much HTML“

“I don’t know”

How long can it take between the user clicking submit and a confirmation that our servers received the data?

“Immediate"

“We sell as real time”

“500ms, too much HTML“

“I don’t know”

What is human perception of immediate? 100ms

Collection API should respond within 150ms

Some, but not too many.

can you settle an argument or priority based on it?

Don’t over achieve.

The Chubby example.

Adapt. Evolve.

re-define SLO's as your product evolves.

Meeting Expectations.

Attach consequences to your Objectives.

The night is dark and full of loopholes.

take a friend from legal with you.

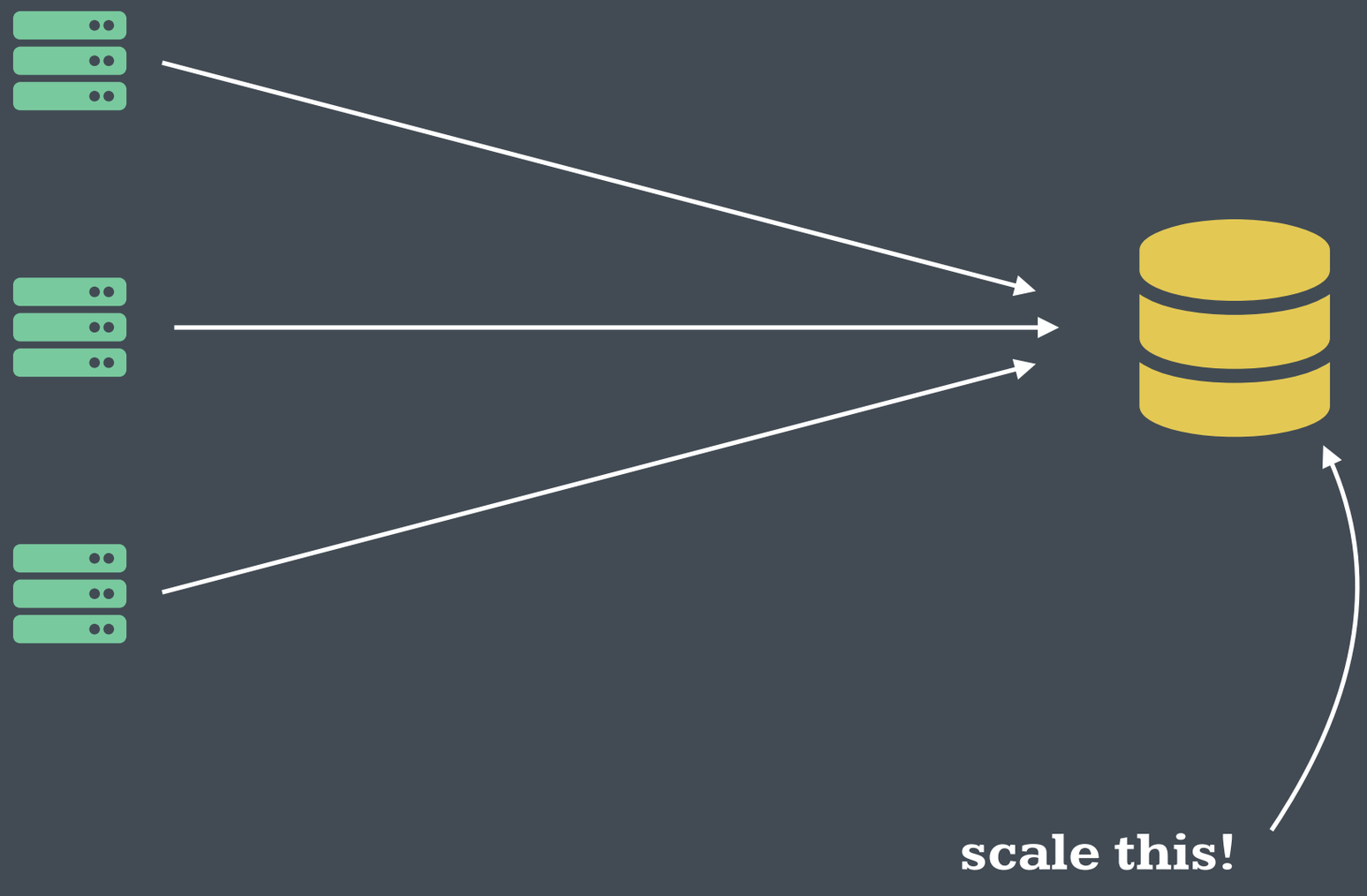
Safety Margins.

like setting the alarm 5 minutes before the meeting.

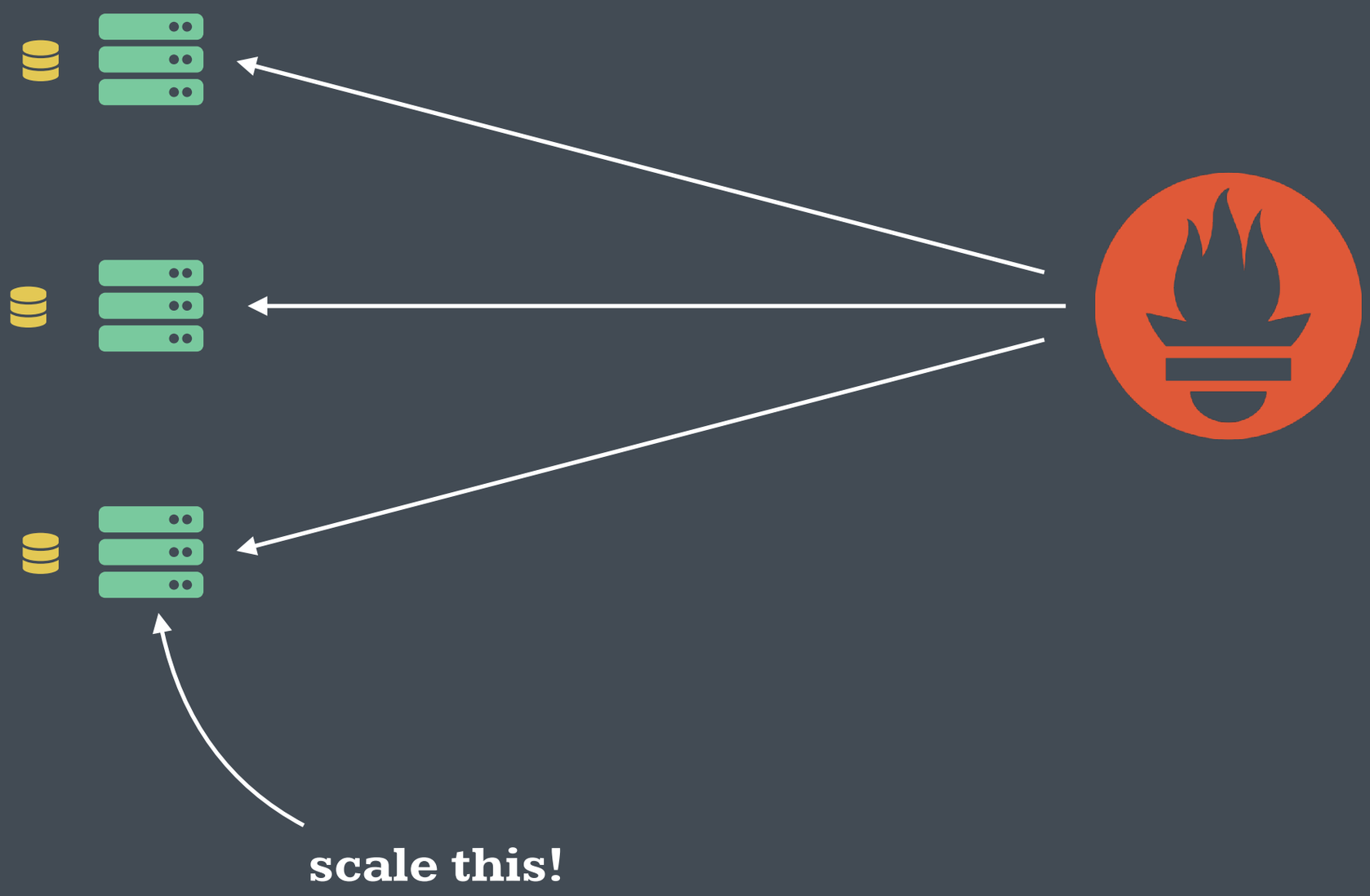
Metrics in Practice.

prometheus.io

Push Model



Pull Model



Telemetry

Prometheus

Statistics

Prometheus

+

StatsD, InfluxDB, etc...

Long Term Storage

Counter

Cumulative metric the represents a single number that only increases

Histogram

Samples and count of observations over time

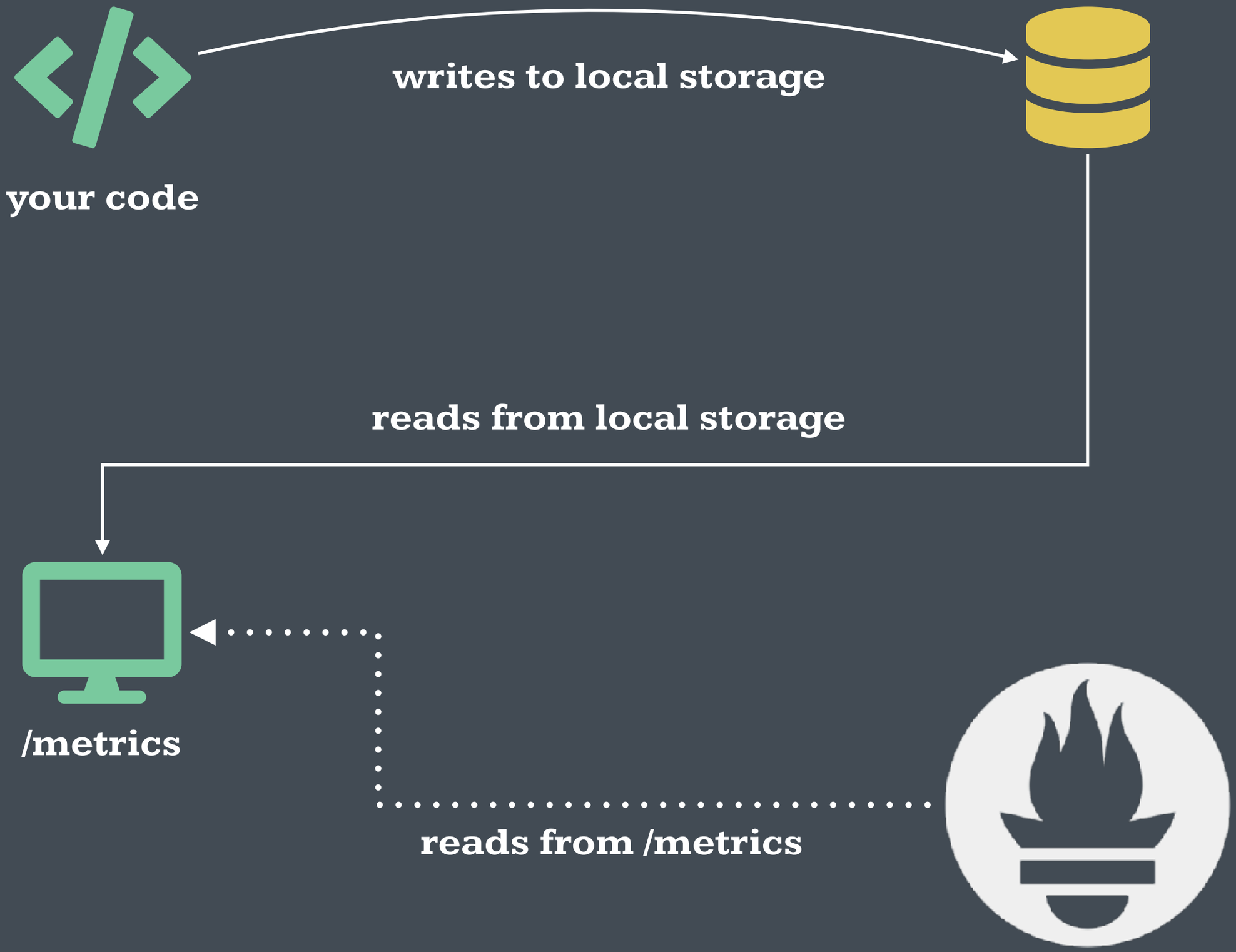
Gauge

A counter, that can go up or down

Summary

Same as a histogram but with stream of quantiles over a sliding window.

jimdo/prometheus_client_php


writes to local storage
your code
reads from local storage
/metrics
reads from /metrics

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();


$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);



$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();

$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();

$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

APC / APCu

Redis

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();


$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

namespace

metric name

help

label names

buckets

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();

$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);

$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

measurement

label values

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();

$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);

$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);

$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

namespace

metric name

help

labels

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();

$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);


$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

```php
<?php

use Prometheus\Counter;
use Prometheus\Histogram;
use Prometheus\Storage\APC;
require_once 'vendor/autoload.php';
$adapter   = new APC();


$histogram = new Histogram(
    $adapter,
    'my_app',
    'response_time_ms',
    'This measures ....',
    ['status', 'url'],
    [0, 10, 50, 100]
);



$histogram->observe(15, ['200', '/url']);


$counter = new Counter($adapter, 'my_app', 'count_total',
'How many...', ['status', 'url']);


$counter->inc(['200', '/url']);
$counter->incBy(5, ['200', '/url']);
```

```php
<?php

use Prometheus\RenderTextFormat;
use Prometheus\Storage\APC;

require_once 'vendor/autoload.php';

$adapter = new APC();


$renderer = new RenderTextFormat();
$result = $renderer->render($adapter->collect());

echo $result;
```

```php
<?php

use Prometheus\RenderTextFormat;
use Prometheus\Storage\APC;

require_once 'vendor/autoload.php';

$adapter = new APC();


$renderer = new RenderTextFormat();
$result = $renderer->render($adapter->collect());

echo $result;
```

```php
<?php

use Prometheus\RenderTextFormat;
use Prometheus\Storage\APC;

require_once 'vendor/autoload.php';

$adapter = new APC();


$renderer = new RenderTextFormat();
$result = $renderer->render($adapter->collect());

echo $result;
```

```
# HELP my_app_count_total How many...
# TYPE my_app_count_total counter
my_app_count_total{status="200",url="/url"} 6

# HELP my_app_response_time_ms This measures ....
# TYPE my_app_response_time_ms histogram
my_app_response_time_ms_bucket{status="200",url="/url",le="0"} 0
my_app_response_time_ms_bucket{status="200",url="/url",le="10"} 0
my_app_response_time_ms_bucket{status="200",url="/url",le="50"} 1
my_app_response_time_ms_bucket{status="200",url="/url",le="100"} 1
my_app_response_time_ms_bucket{status="200",url="/url",le="+Inf"} 1
my_app_response_time_ms_count{status="200",url="/url"} 1
my_app_response_time_ms_sum{status="200",url="/url"} 16
```

“Demos always fail.”

–Rafael (yesterday)

“I’ll just try this live demo again.”

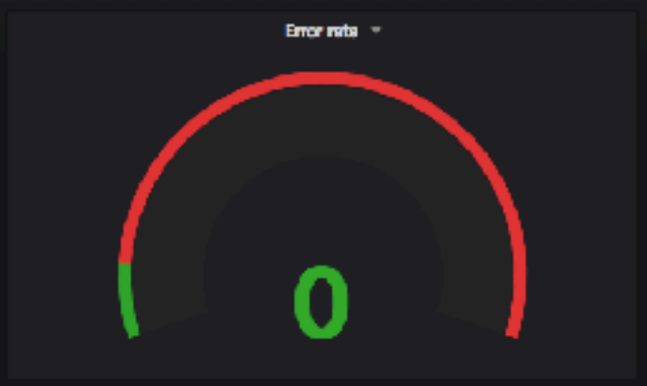
–Also Rafael (today)

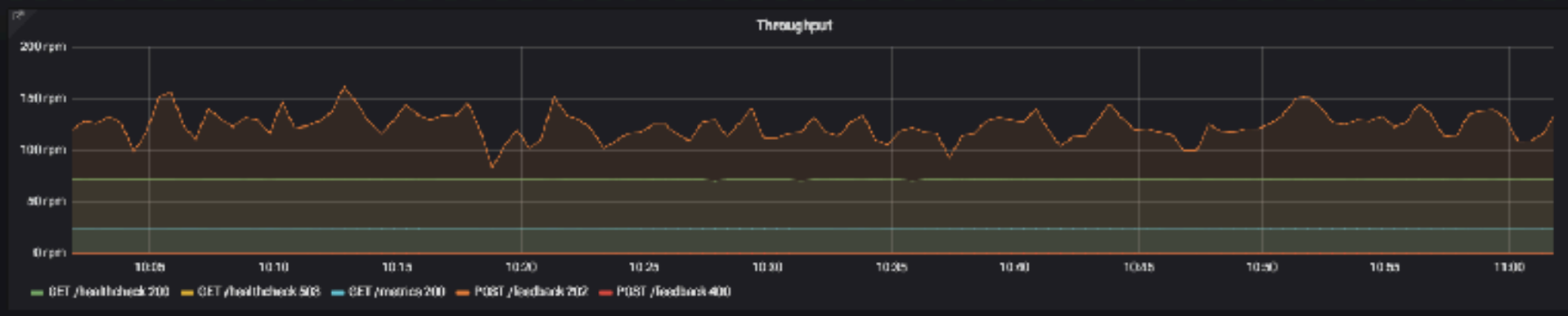
https://github.com/rdohms/talk-app-metrics

http://localhost:8180/index http://localhost:9090/graph http://localhost:8180/metrics

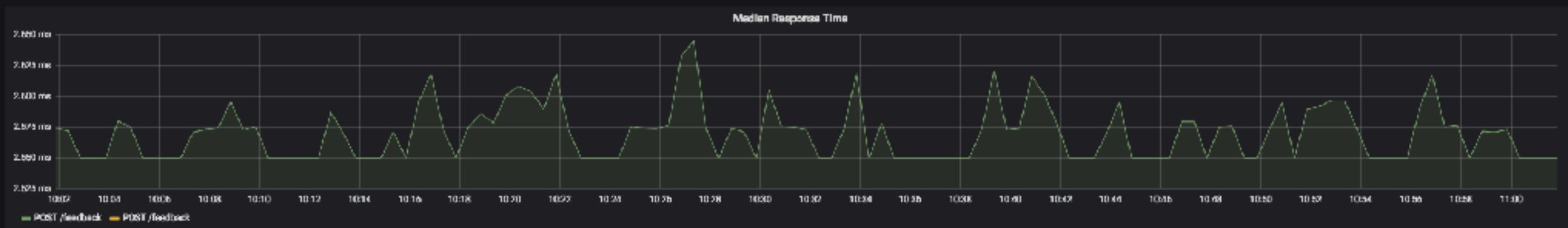
You can't act on what you can't see.

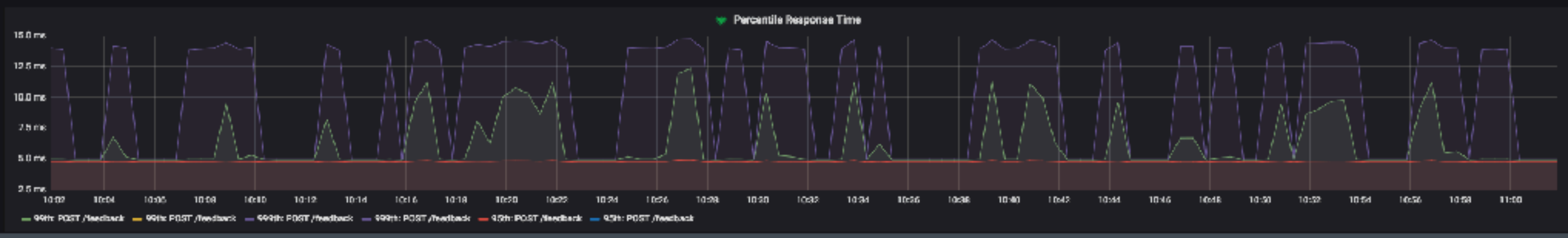
Throughput





Response time





web_passive

1.2/min

Last 24 hours

3.72%

web_active

72/min

Last 24 hours

75.18%

sdk_passive

7/min

Last 24 hours

1.74%

sdk_active

6.0/min

Last 24 hours

4.33%

mail

0/min

Last 24 hours

1.45%

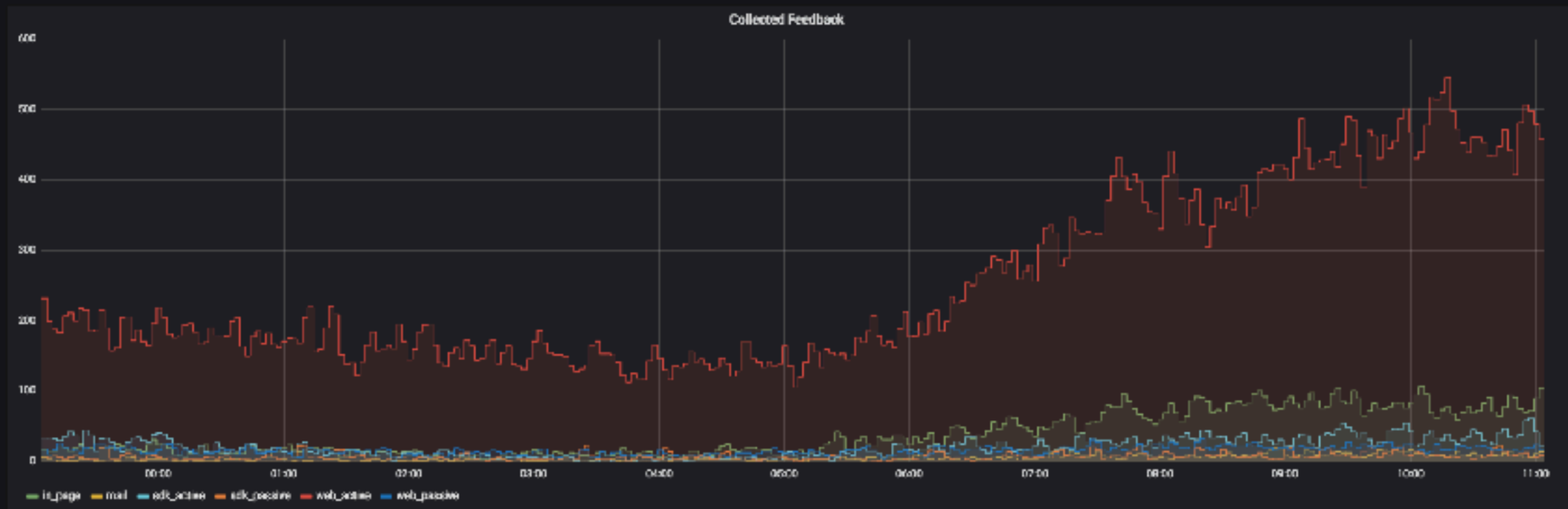
in_page

20/min

Last 24 hours

13.58%

Collected Feedback



Total Feedback

106/min

Metric Endpoints Up

100.0%

Metrics without actionability are just numbers on a screen.

Act as soon as an SLO is threatened.

Thank you.

Drop me some feedback at Usabilla and make this talk better.

@rdohms
http://slides.doh.ms