

*Hynek Schlawack*

# Get Instrumented

## How Prometheus Can Unify Your Metrics

---

# Goals

---

---

# Goals

---



---

# Goals

---



---

# Goals

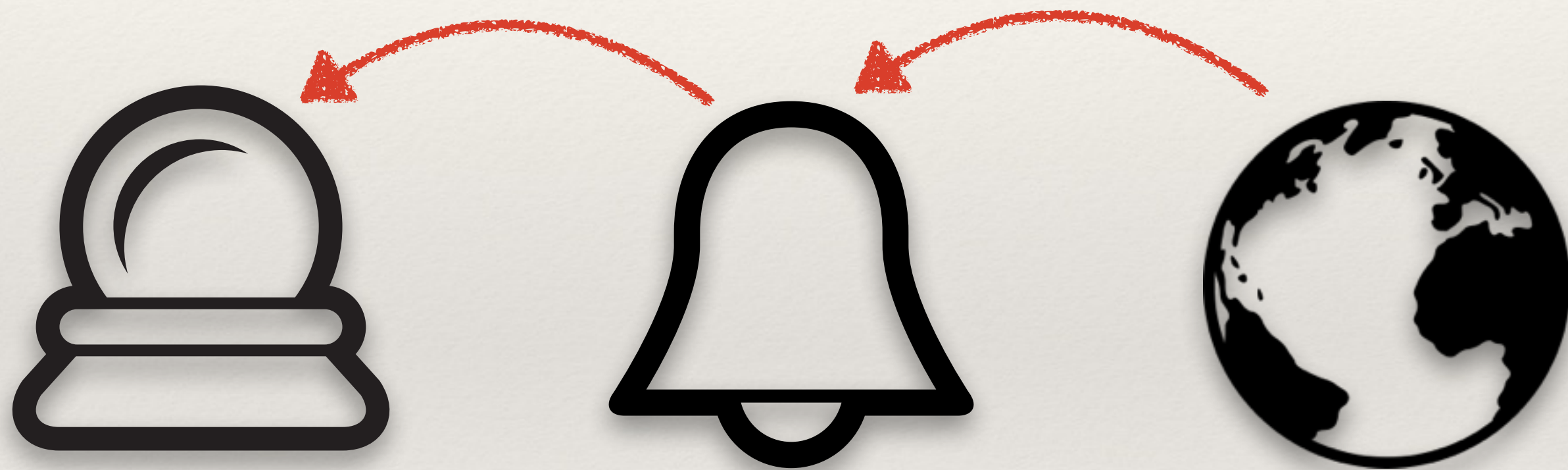
---



---

# Goals

---



# Service Level

# Service Level **Indicator**



Service Level **Indicator**

**Objective**

**Service Level Indicator**

**Objective**

**(Agreement)**

---

# Metrics

---

---

# Metrics

---

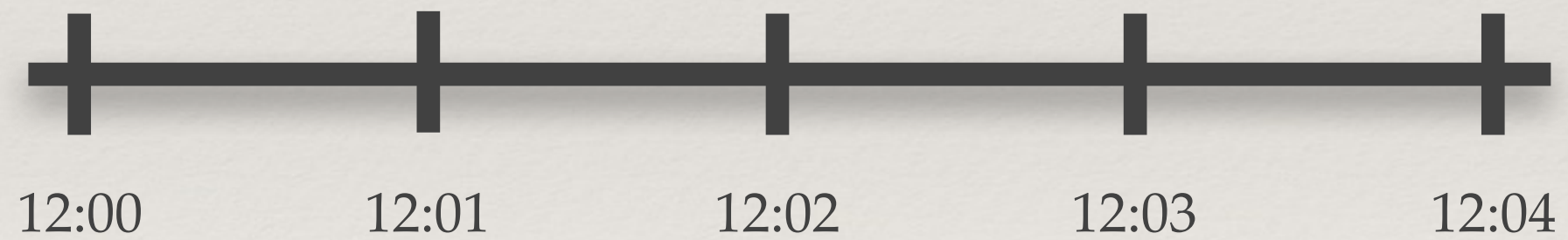
avg latency	0.3	0.5	0.8	1.1	2.6
-------------	-----	-----	-----	-----	-----

---

# Metrics

---

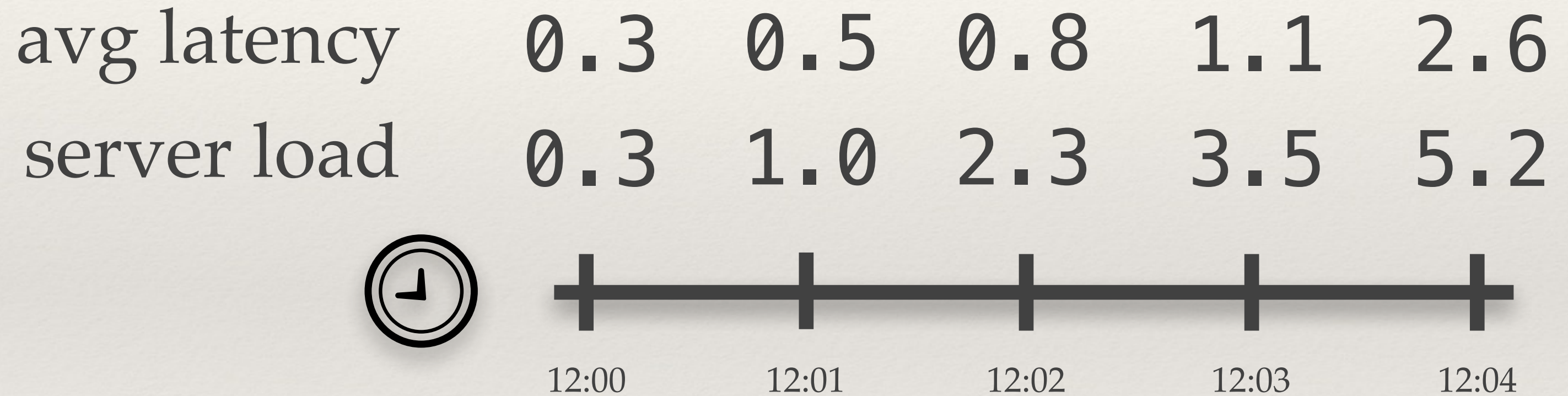
avg latency      0.3      0.5      0.8      1.1      2.6



---

# Metrics

---



F-WWOW | AJ-DM

F-WWOW | AJ-

CHRONO  
STICK PRIORITY

MACH SPD TRUE HDG YFS METER

SPD HDG TRK FPA ALT V/S

CSTR MPT UORD HDB RRPT  
-D- MK TERR TRRF -C- Std

1020 03000

LOC A/TW ALT APPR

ZOOM IN OUT

LS VV



SWITCHED ATT HDG

AIR DATA

FMS

L/O GRVY

OUT OF ORDER

THE XX THE XX THE XX THE XX

XX XX NI XX XX

XX XX EST XX XX

CAB PRESS SPO FAULT

WOMB UPPER BL CTL FAULT ACTUATOR FAULT

D -CABIN CREW .....ADVISE

ELEC C/B TRIPPED

L/O

AUTO BRK

BRK FAN

A-SKIP

UP DOWN

06:52:16

03:00



AIRSIDA

ACTIVE	POSITION	SEC	DATA
DISCONTINUITY			
D167A	88:41	338	FL288
INTCPT	88:42		195° 29
CLIFF	88:46		9° 425
SCOTT	81:44		
DISCONTINUITY			
IT/OI	82:26	338	FL288
ISPO1	82:32	208	FL188
DEST	YSSY	82:44	29.6T 1139 NM

CAB PRESS

DELTA P PSI

AUTO CAB ALT FT

AUTO V/S FT/PMIN

0.0 -150 0

CAB AIR EXTRACT

PACK 1

PACK 2

TAT +16 °C

SAT +16 °C

ISA +1 °C

065215 gph

Qn 376200 hPa

QnCS 37.2 hPa

FOB 68200 hPa

AIRSIDA

ACTIVE POSITION SEC DATA

AMPT IDENT YSCB

CANBERRA

35°18.45/149°11.7E

17 2600h

25 2600h LS

1 2 3 4

Q W E R T Y U I O P

A S D F G H J K L

Z X C V B N M

---

# Instrument

---



---

# Instrument

---



---

# Instrument

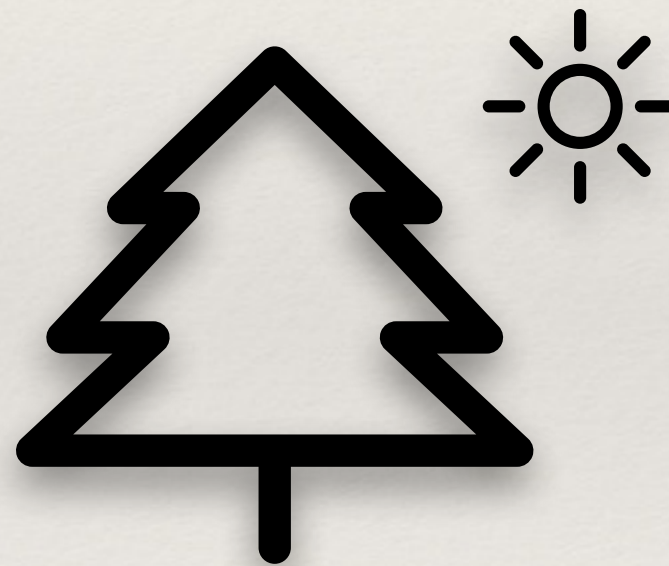
---



---

# Instrument

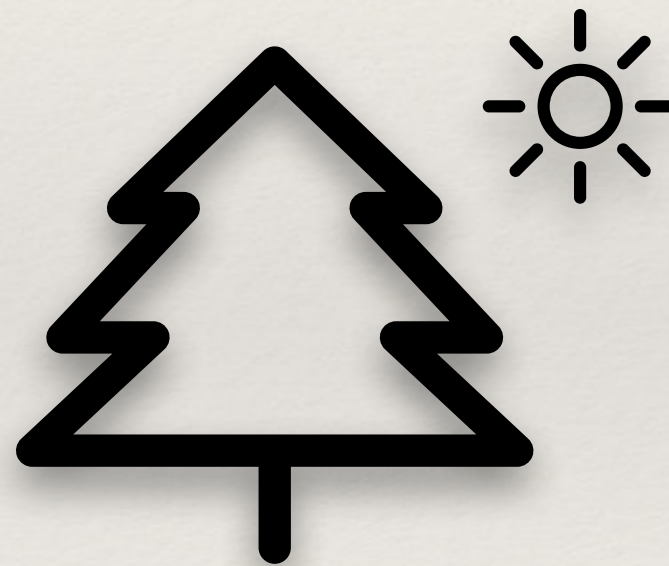
---



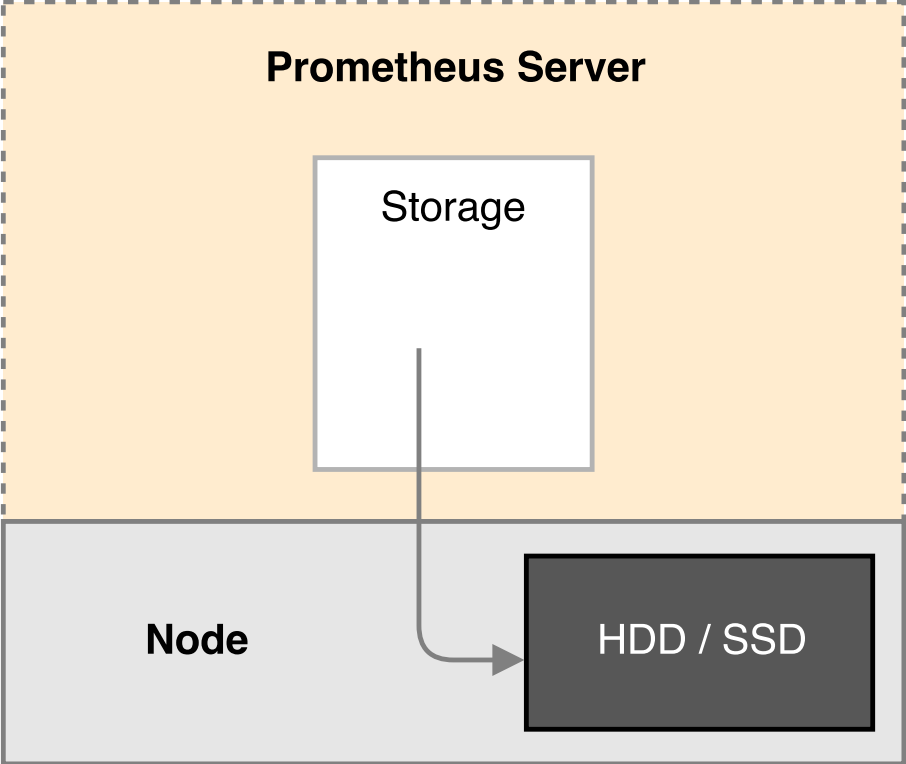
---

# Instrument

---







---

# Metric Types

---

---

# Metric Types

---

❖ counter



---

# Metric Types

---

❖ counter

❖ gauge

---

# Metric Types

---

❖ counter

❖ summary

❖ gauge

---

# Metric Types

---

❖ counter

❖ summary

❖ gauge

❖ histogram

---

# Metric Types

---

❖ counter

❖ summary

❖ gauge

❖ histogram

❖ buckets (1s,  
0.5s, 0.25, ...)

---

# Averages

---

---

# Averages

---

❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$

---

# Averages

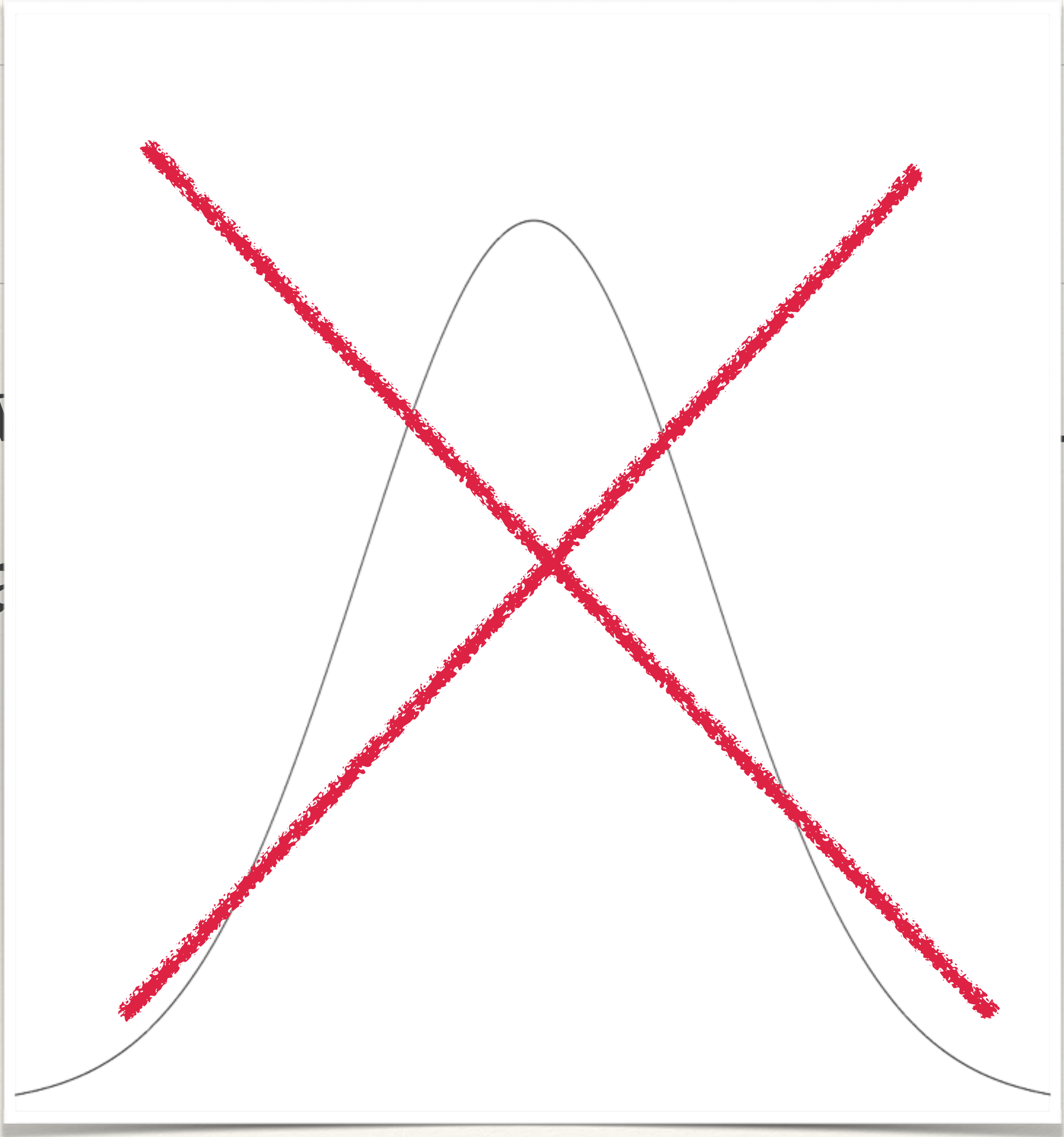
---

- ❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$
- ❖  $\text{avg}(\{1, 1, 1, 1, 10\}) = 2.8$

❖ av

❖ a

JX)





---

# Averages

---

- ❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$
- ❖  $\text{avg}(\{1, 1, 1, 1, 10\}) = 2.8$

---

# Averages

---

- ❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$
- ❖  $\text{avg}(\{1, 1, 1, 1, 10\}) = 2.8$
- ❖  $\text{median}(\{1, 1, 1, 1, 10\}) = \mathbf{1}$

---

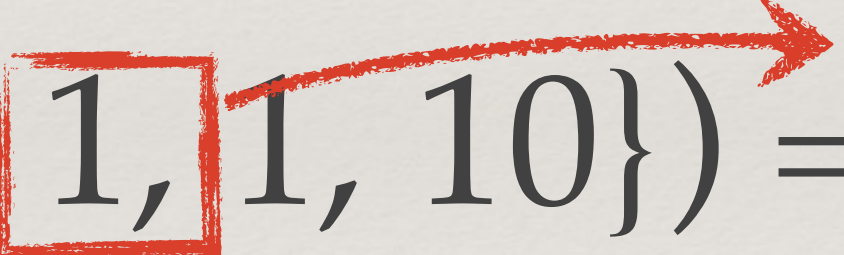
# Averages

---

❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$

❖  $\text{avg}(\{1, 1, 1, 1, 10\}) = 2.8$

❖  $\text{median}(\{1, 1, \boxed{1}, 1, 10\}) = \mathbf{1}$



---


# Averages

---


❖  $\text{avg}(\text{request time}) \neq \text{avg}(\text{UX})$

❖  $\text{avg}(\{1, 1, 1, 1, 10\}) = 2.8$

❖  $\text{median}(\{1, 1, \boxed{1}, 1, 10\}) = \mathbf{1}$



❖  $\text{median}(\{1, \boxed{1}, 100\_000\}) = \mathbf{1}$



---

# Percentiles

---

---

# Percentiles

---

$n^{\text{th}}$  percentile  $P$  of a data set

=

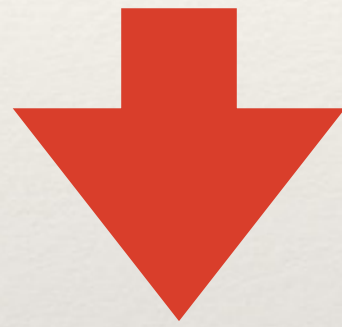
$P \geq n\%$  of values



50<sup>th</sup> percentile = **1 ms**



50<sup>th</sup> percentile = **1 ms**



50% of requests *done* by **1 ms**

---

# Percentiles

---

---

# Percentiles

---

P

{1, 1, 100\_000}

50<sup>th</sup>

1

---

# Percentiles

---

P

{1, 1, 100\_000}

50<sup>th</sup>

1

95<sup>th</sup>

90\_000

---

# Naming

---

---

# Naming

---

```
backend1_app_http_reqs_msgs_post  
backend1_app_http_reqs_msgs_get  
...
```

---

# Naming

---

~~backend1\_app\_http\_reqs\_msgs\_post~~  
~~backend1\_app\_http\_reqs\_msgs\_get~~  
...  
app\_http\_reqs\_total

---

# Naming

---

~~backend1\_app\_http\_reqs\_msgs\_post~~

~~backend1\_app\_http\_reqs\_msgs\_get~~

...

app\_http\_reqs\_total



---

# Naming

---

~~backend1\_app\_http\_reqs\_msgs\_post~~  
~~backend1\_app\_http\_reqs\_msgs\_get~~

...

app\_http\_reqs\_total

---

# Naming

---

~~backend1\_app\_http\_reqs\_msgs\_post~~

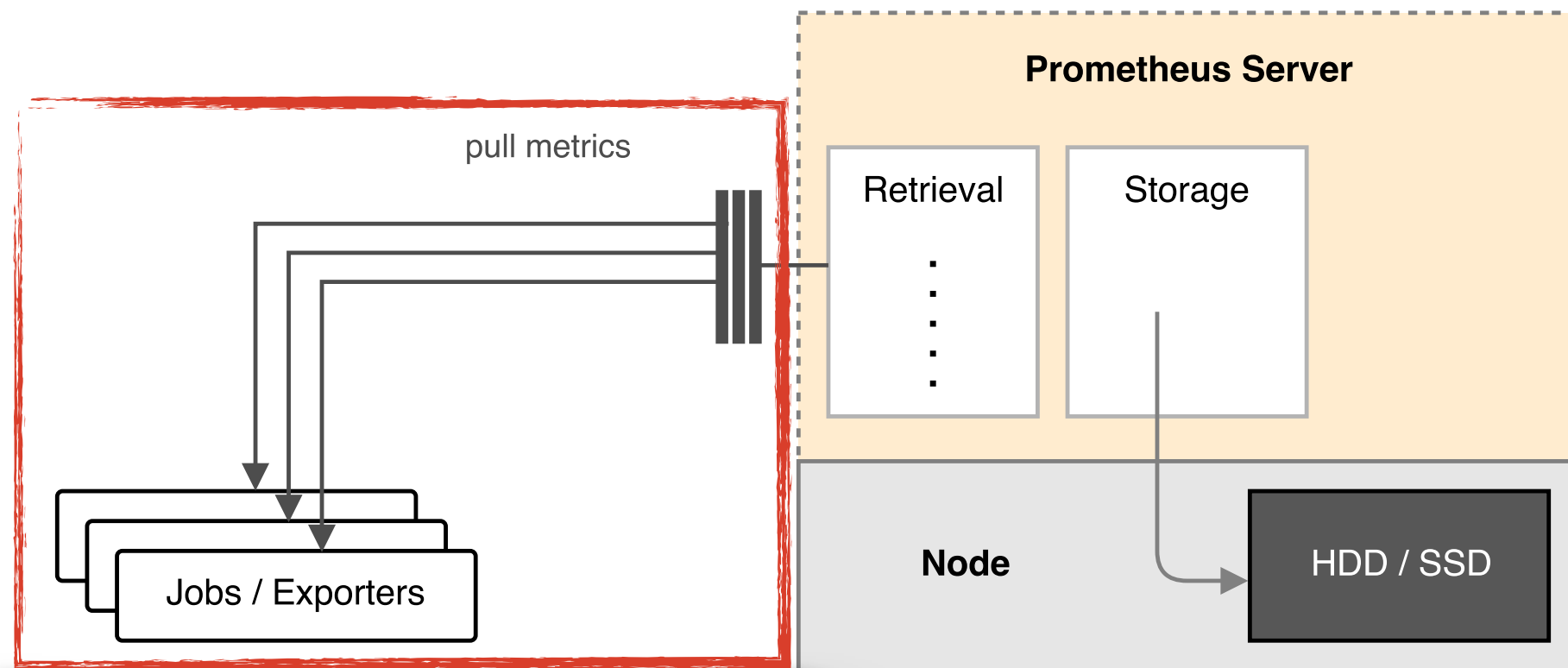
~~backend1\_app\_http\_reqs\_msgs\_get~~

...

app\_http\_reqs\_total{meth="POST", path="/msgs", backend="1"}

app\_http\_reqs\_total{meth="GET", path="/msgs", backend="1"}

...





1. resolution = scraping interval

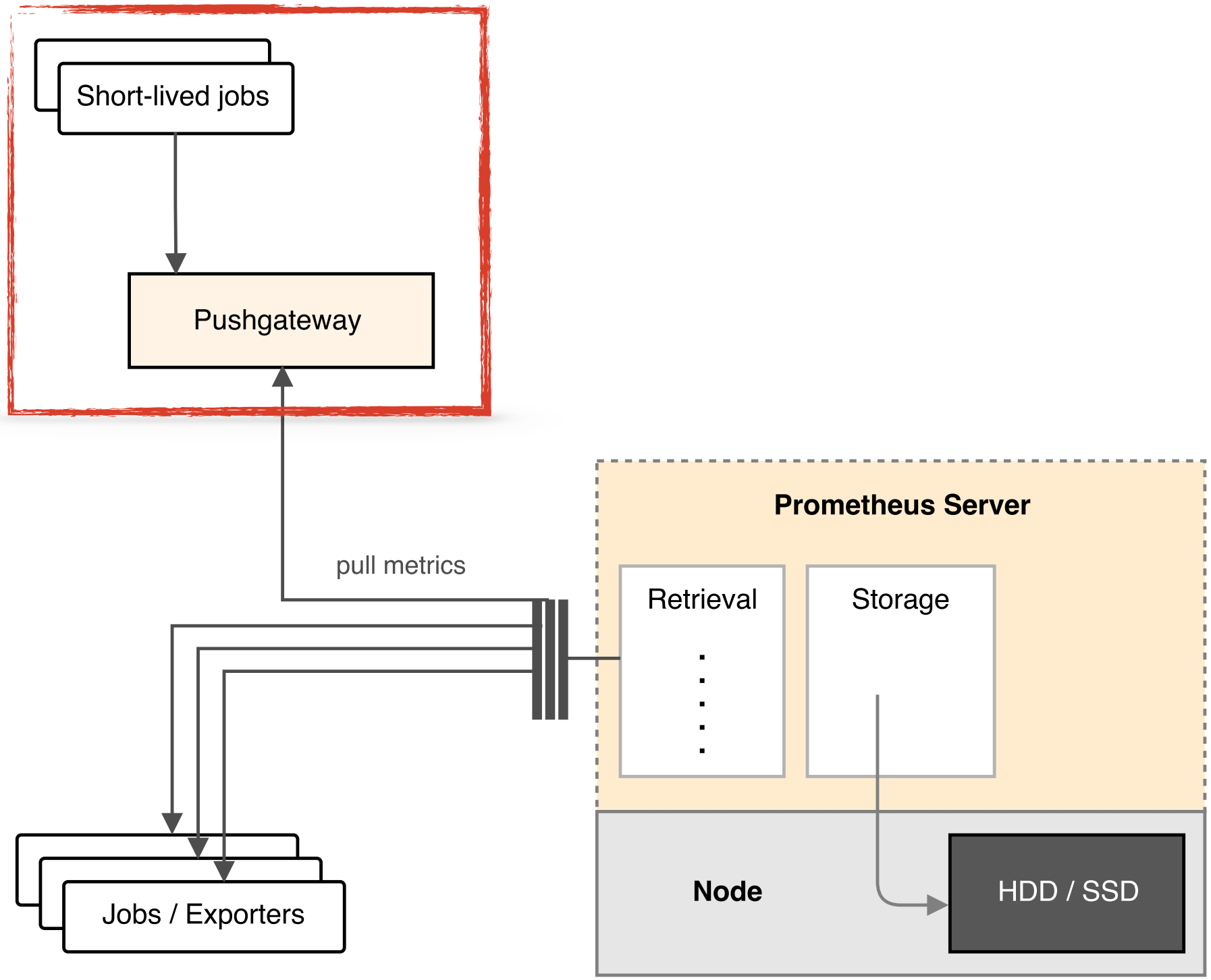
1. resolution = scraping interval
2. missing scrapes = less resolution

---

# Pull: Problems

---

- ❖ short lived jobs





---

# Pull: Problems

---

- ❖ short lived jobs
- ❖ target discovery

---

# Configuration

---

```
scrape_configs:  
  - job_name: 'prometheus'  
    static_configs:  
      - targets:  
        - 'localhost:9090'
```

---

# Configuration

---

```
scrape_configs:  
  - job_name: 'prometheus'  
    static_configs:  
      - targets:  
        - 'localhost:9090'
```

---

# Configuration

---

```
scrape_configs:
```

```
- job_name: 'prometheus'
```

```
  static_configs:
```

```
    - targets:
```

```
      - 'localhost:9090'
```

---

# Configuration

---

```
scrape_configs:
```

```
- job_name: 'prometheus'
```

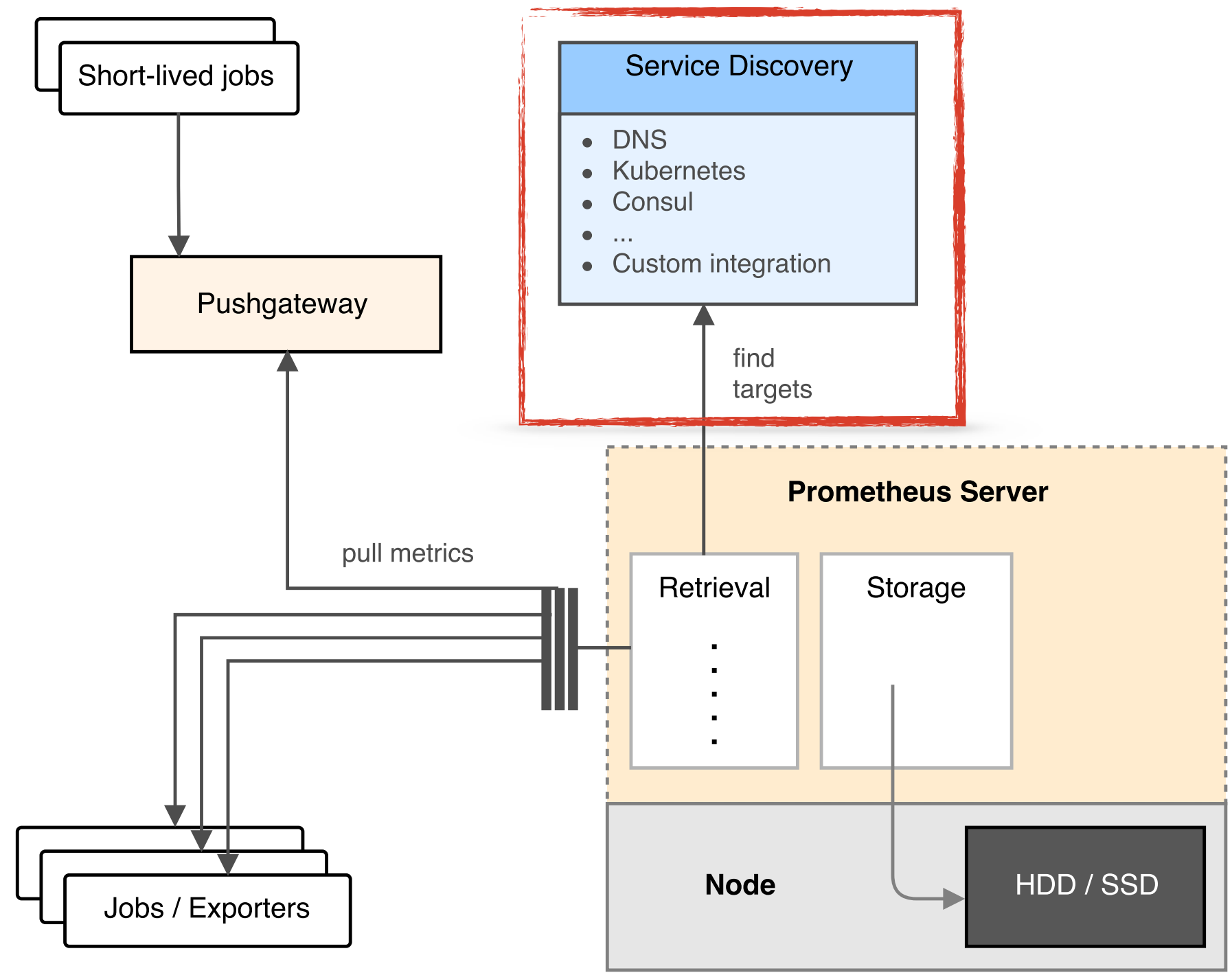
```
  static_configs:
```

```
    - targets:
```

```
      - 'localhost:9090'
```



```
{instance="localhost:9090", job="prometheus"}
```



---

# Pull: Problems

---

- ❖ target discovery
- ❖ short lived jobs
- ❖ Heroku / NATed systems

---

# Pull: *Advantages*

---



---

# Pull: Advantages

---

- ❖ multiple Prometheus easy

---

# Pull: Advantages

---

- ❖ multiple Prometheus easy
- ❖ outage detection

---

# Pull: Advantages

---

- ❖ multiple Prometheus easy
- ❖ outage detection
- ❖ predictable, no self-DoS

---

# Pull: Advantages

---

- ❖ multiple Prometheus easy
- ❖ outage detection
- ❖ predictable, no self-DoS
- ❖ easy to instrument 3<sup>rd</sup> parties

---

# Metrics Format

---

```
# HELP req_seconds Time spent \
processing a request in seconds.
# TYPE req_seconds histogram
req_seconds_count 390.0
req_seconds_sum 177.0319407
```

---

# Metrics Format

---

```
# HELP req_seconds Time spent \
processing a request in seconds.
# TYPE req_seconds histogram
req_seconds_count 390.0
req_seconds_sum 177.0319407
```

---

# Metrics Format

---

```
# HELP req_seconds Time spent \
processing a request in seconds.
# TYPE req_seconds histogram
req_seconds_count 390.0
req_seconds_sum 177.0319407
```

---

# Metrics Format

---

```
# HELP req_seconds Time spent \
processing a request in seconds.
# TYPE req_seconds histogram
req_seconds_count 390.0
req_seconds_sum 177.0319407
```



---

# Metrics Format

---

```
# HELP req_seconds Time spent \
processing a request in seconds.
# TYPE req_seconds histogram
req_seconds_count 390.0
req_seconds_sum 177.0319407
```

---

# Percentiles

---

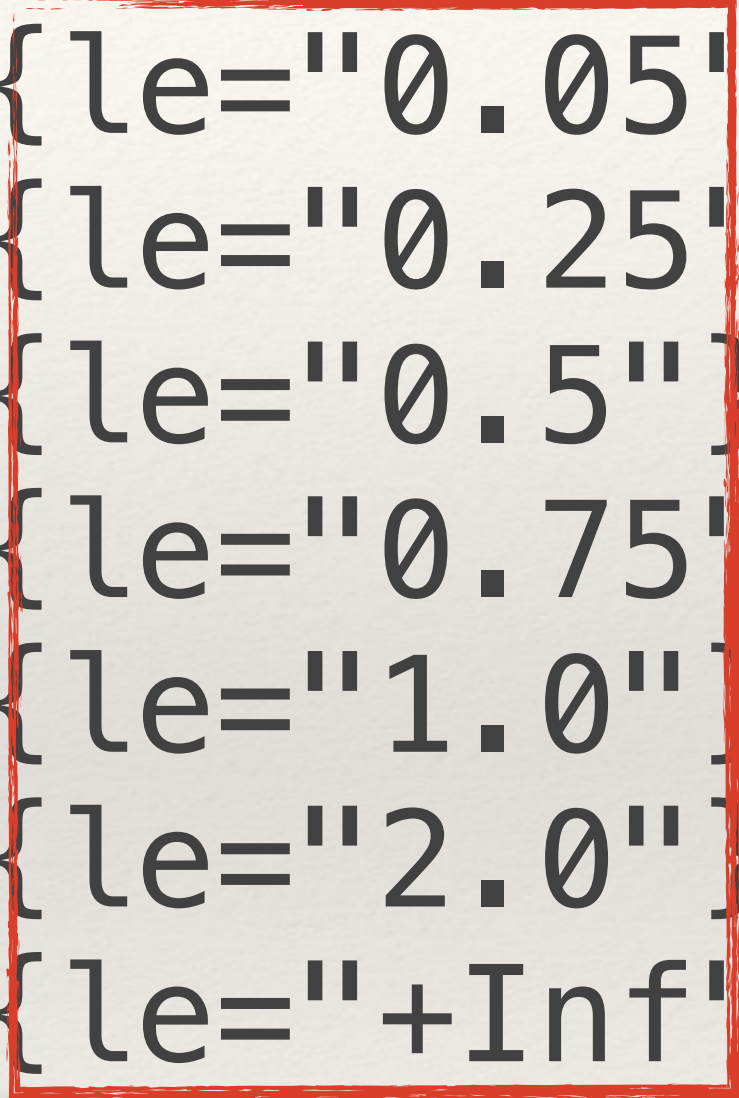
```
req_seconds_bucket{le="0.05"} 0.0
req_seconds_bucket{le="0.25"} 1.0
req_seconds_bucket{le="0.5"} 273.0
req_seconds_bucket{le="0.75"} 369.0
req_seconds_bucket{le="1.0"} 388.0
req_seconds_bucket{le="2.0"} 390.0
req_seconds_bucket{le="+Inf"} 390.0
```

---

# Percentiles

---

```
req_seconds_bucket{le="0.05"} 0.0  
req_seconds_bucket{le="0.25"} 1.0  
req_seconds_bucket{le="0.5"} 273.0  
req_seconds_bucket{le="0.75"} 369.0  
req_seconds_bucket{le="1.0"} 388.0  
req_seconds_bucket{le="2.0"} 390.0  
req_seconds_bucket{le="+Inf"} 390.0
```

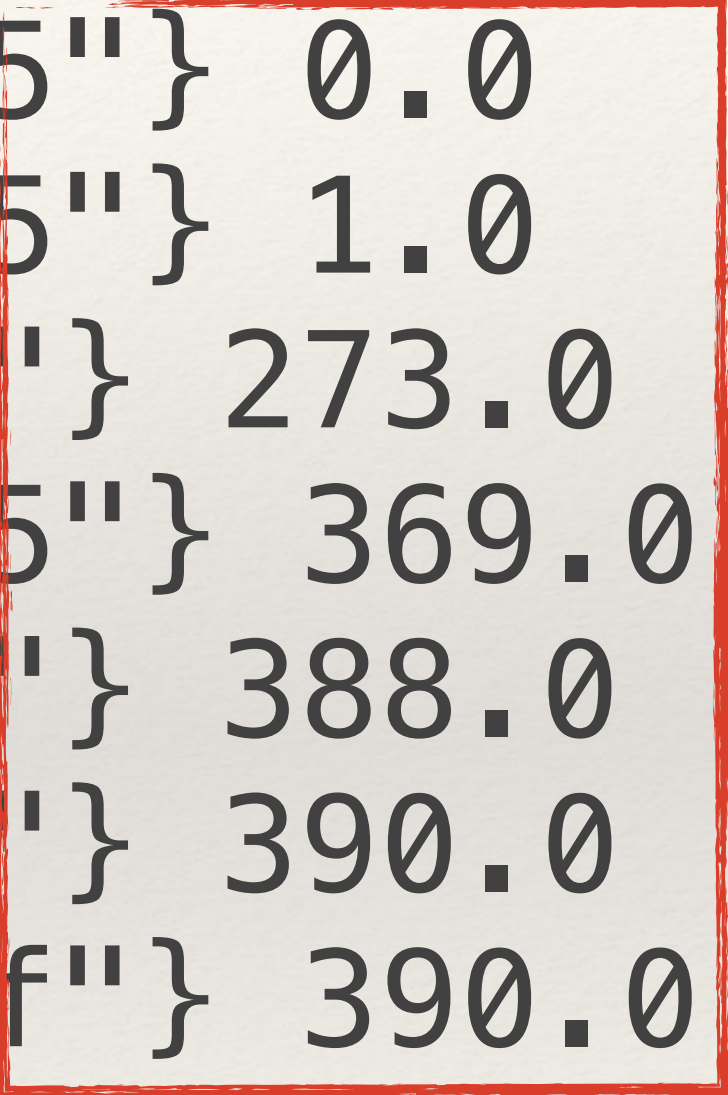


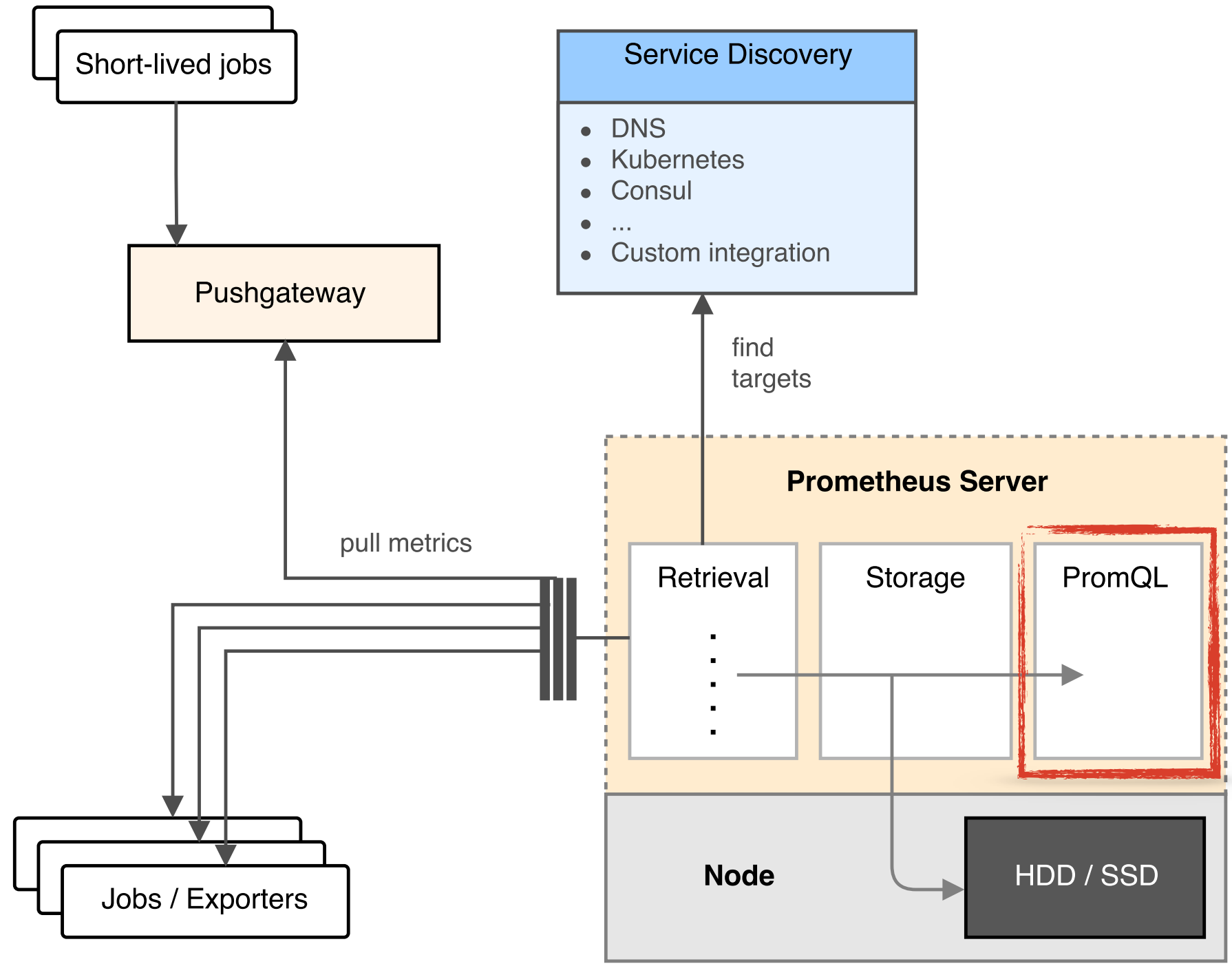
---

# Percentiles

---

```
req_seconds_bucket{le="0.05"} 0.0  
req_seconds_bucket{le="0.25"} 1.0  
req_seconds_bucket{le="0.5"} 273.0  
req_seconds_bucket{le="0.75"} 369.0  
req_seconds_bucket{le="1.0"} 388.0  
req_seconds_bucket{le="2.0"} 390.0  
req_seconds_bucket{le="+Inf"} 390.0
```





---

# Aggregation

---

---

# Aggregation

---

```
sum(  
  rate(  
    req_seconds_count[1m]  
  )  
)
```

---

# Aggregation

---

```
sum(  
  rate(  
    req_seconds_count[1m]  
  )  
)
```



---

# Aggregation

---

```
sum(  
  rate(  
    req_seconds_count[1m]  
  )  
)
```

---

# Aggregation

---

```
sum(  
  rate(  
    req_seconds_count[1m]  
  )  
)
```

---

# Aggregation

---

```
sum(  
  rate(  
    req_seconds_count{dc="west"}[1m]  
  )  
)
```

---

# Aggregation

---

```
sum(  
    rate(  
        req_seconds_count[1m]  
    )  
) by (dc)
```

---

# Percentiles

---

```
histogram_quantile(  
    0.9, rate(  
        req_seconds_bucket[10m]  
    ))
```

---

# Percentiles

---

```
histogram_quantile(  
    0.9, rate(  
        req_seconds_bucket[10m]  
    ))
```

---

# Percentiles

---

```
histogram_quantile(  
    0.9, rate(  
        req_seconds_bucket [10m]  
    ))
```

---

# Percentiles

---

```
histogram_quantile(  
    0.9, rate(  
        req_seconds_bucket[10m]  
    ))
```

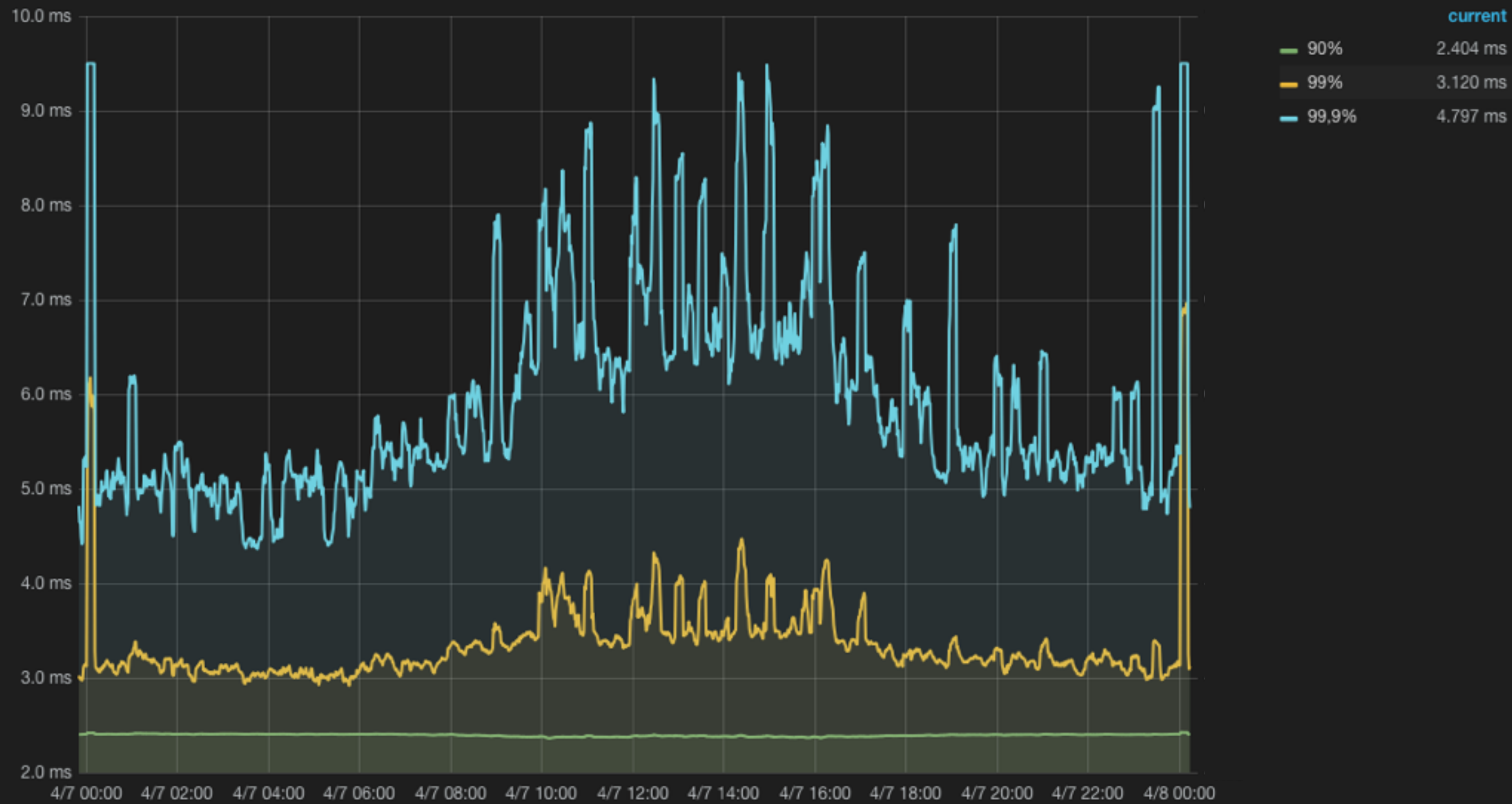


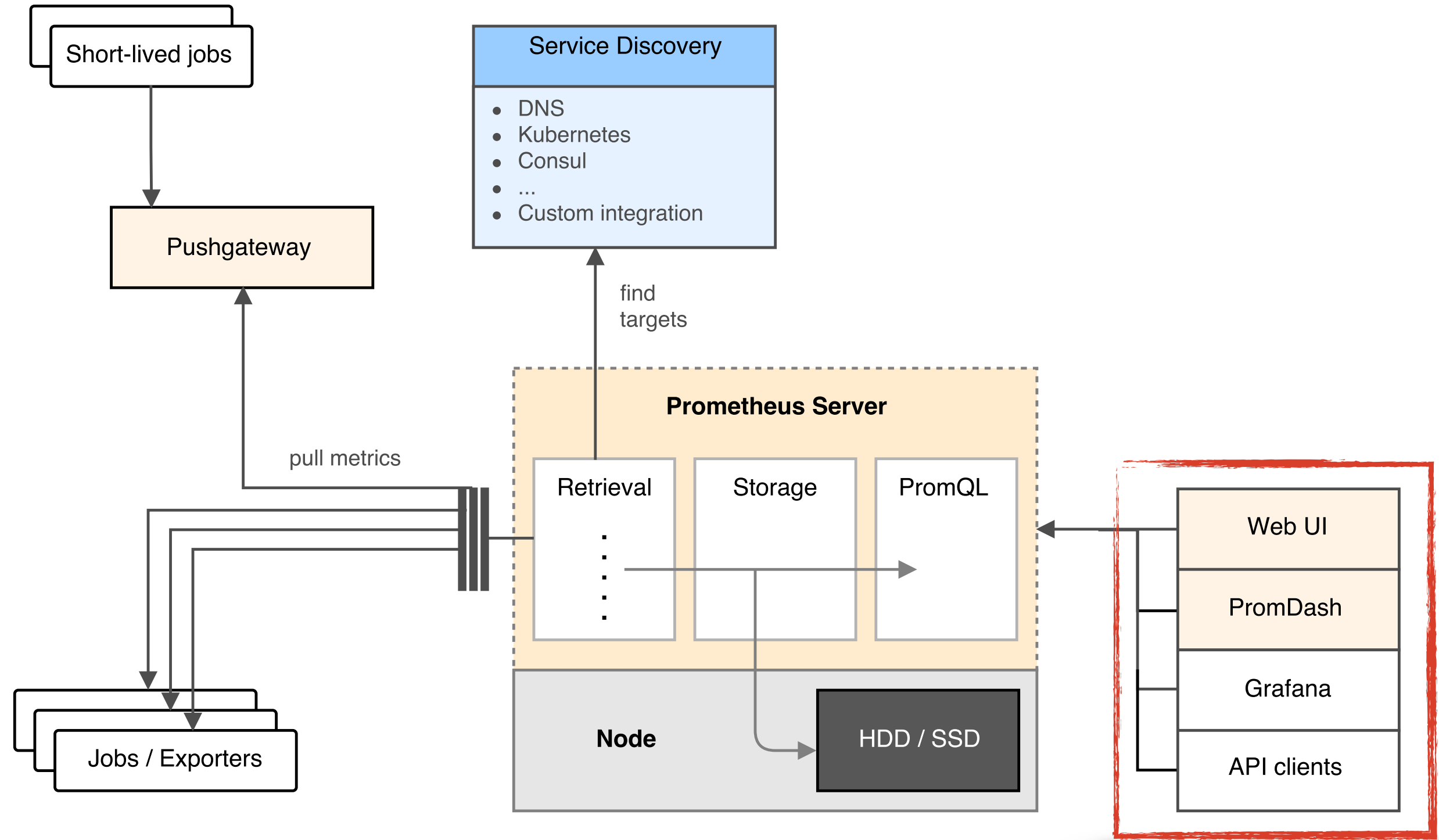
---

# Percentiles

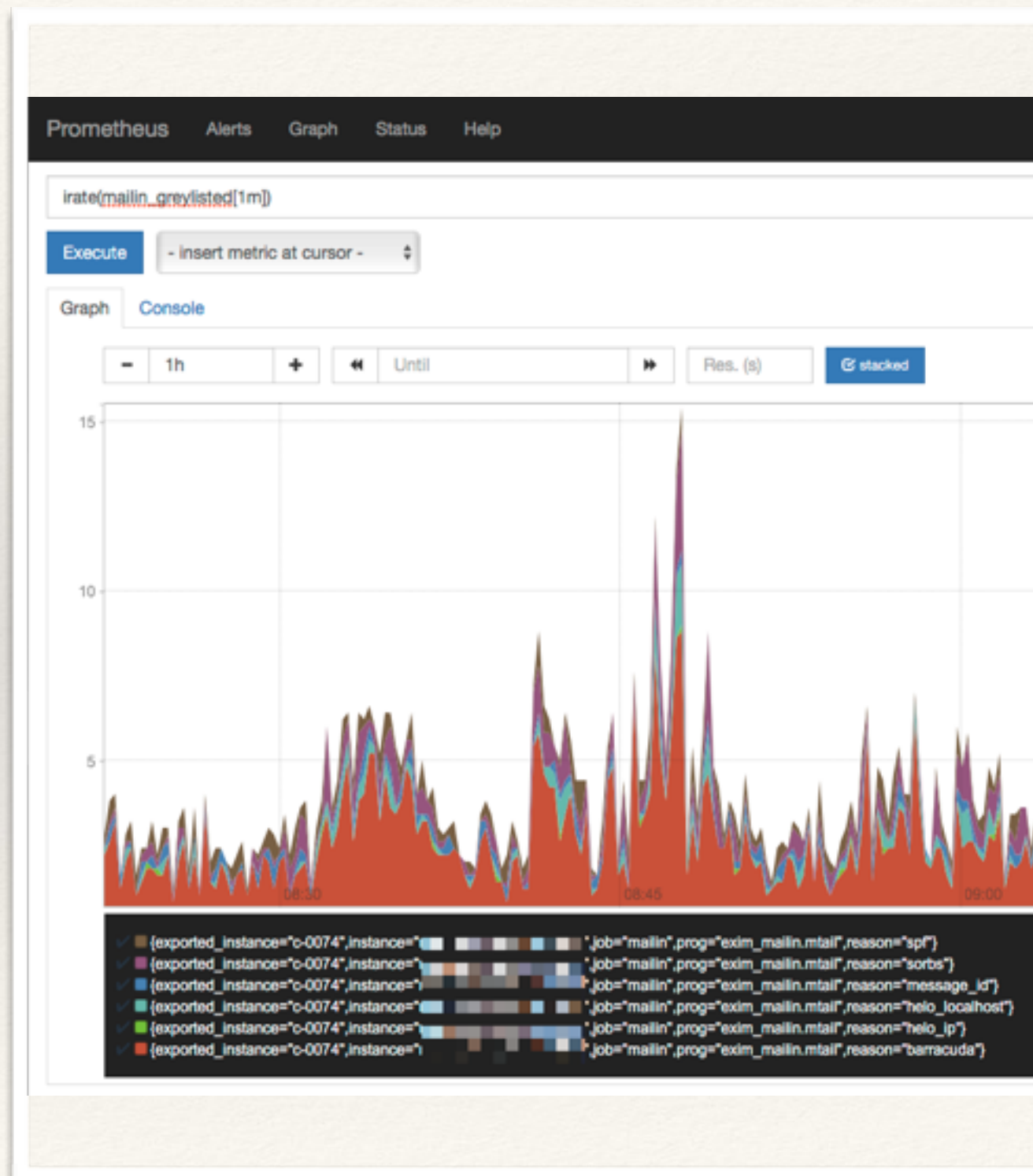
---

```
histogram_quantile(  
    0.9, rate(  
        req_seconds_bucket [10m]  
    ))
```



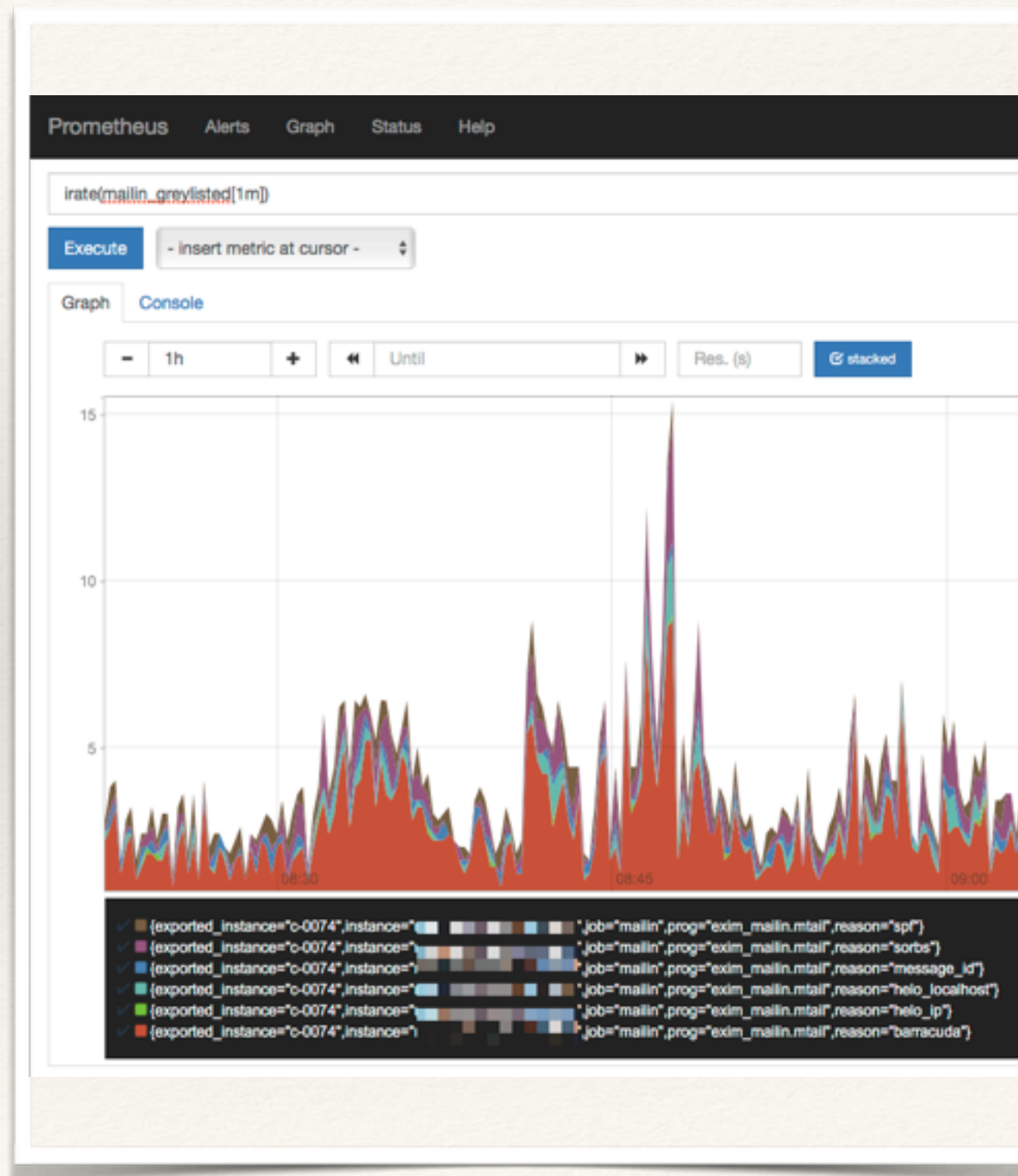


# Internal



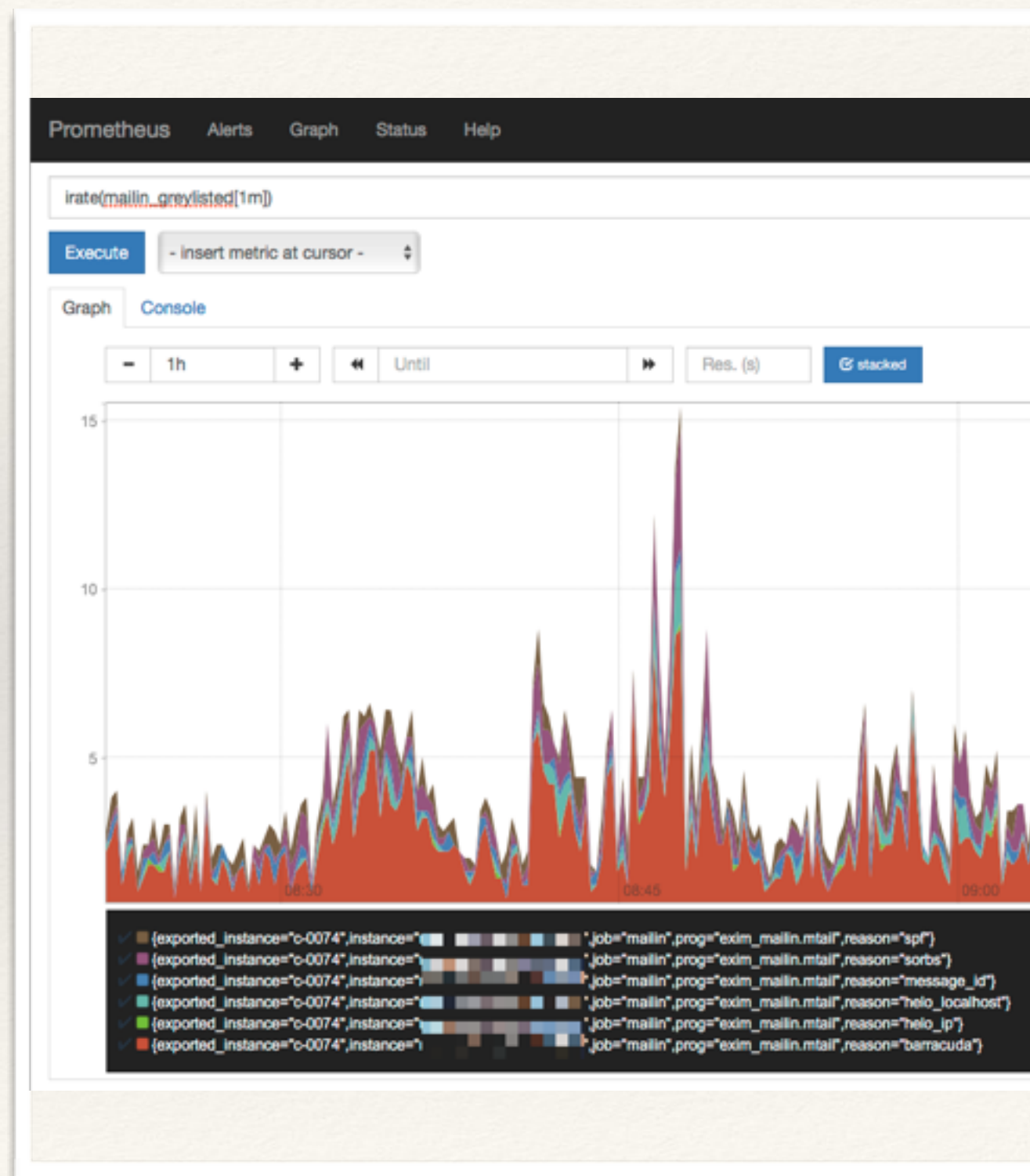
# Internal

❖ great for ad-hoc



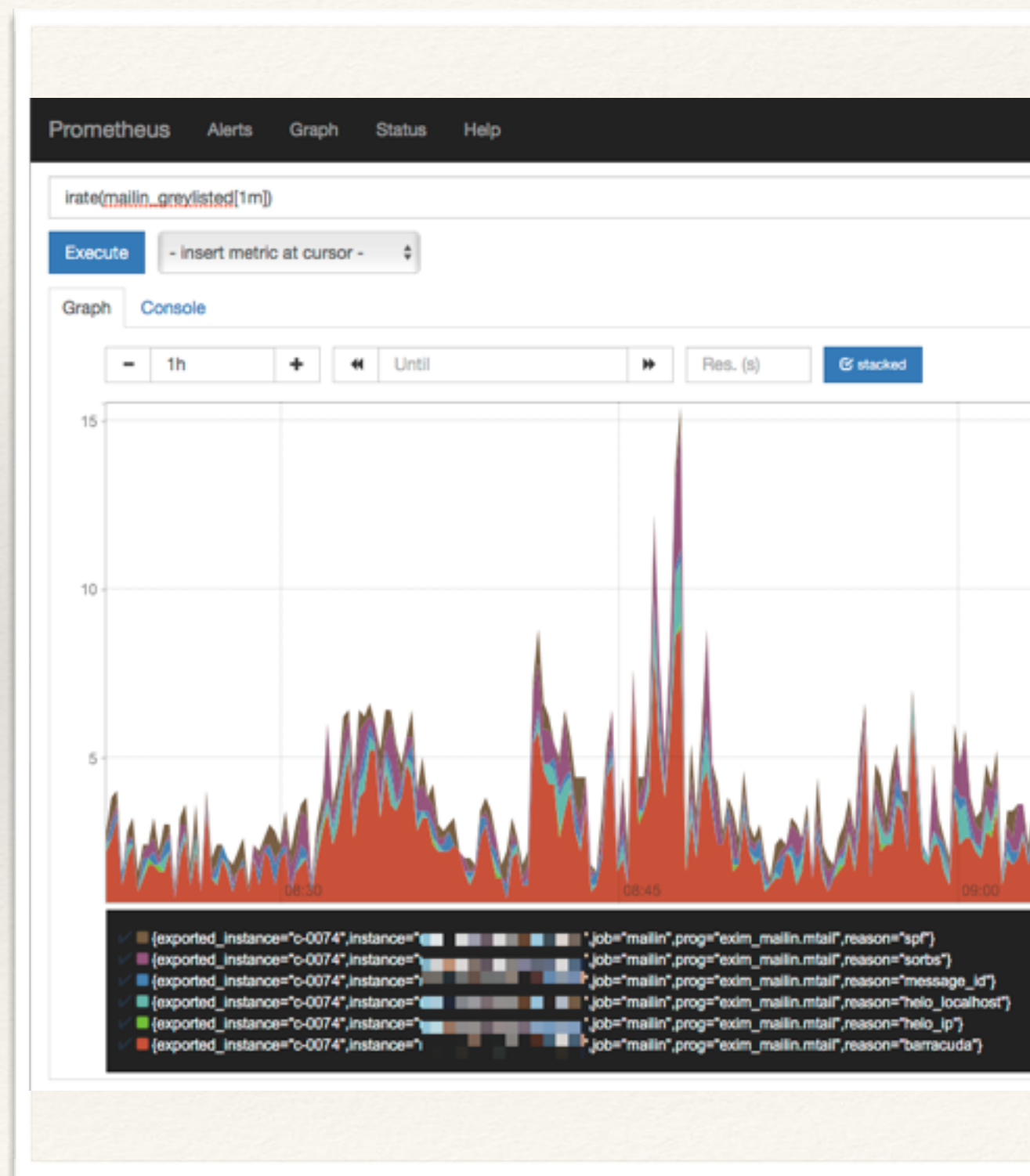
# Internal

- ❖ great for ad-hoc
- ❖ 1 expr per graph

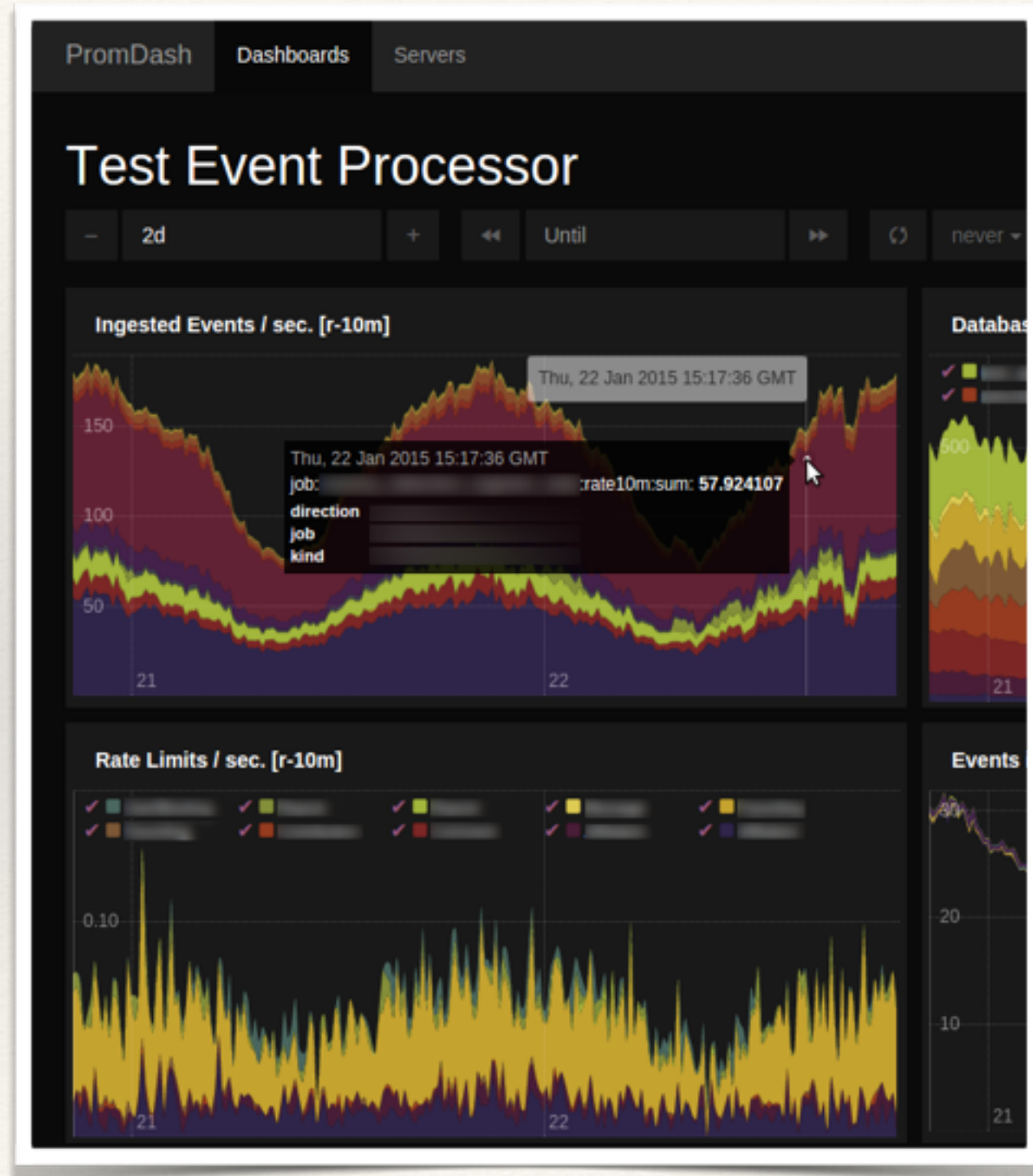


# Internal

- ❖ great for ad-hoc
- ❖ 1 expr per graph
- ❖ templating



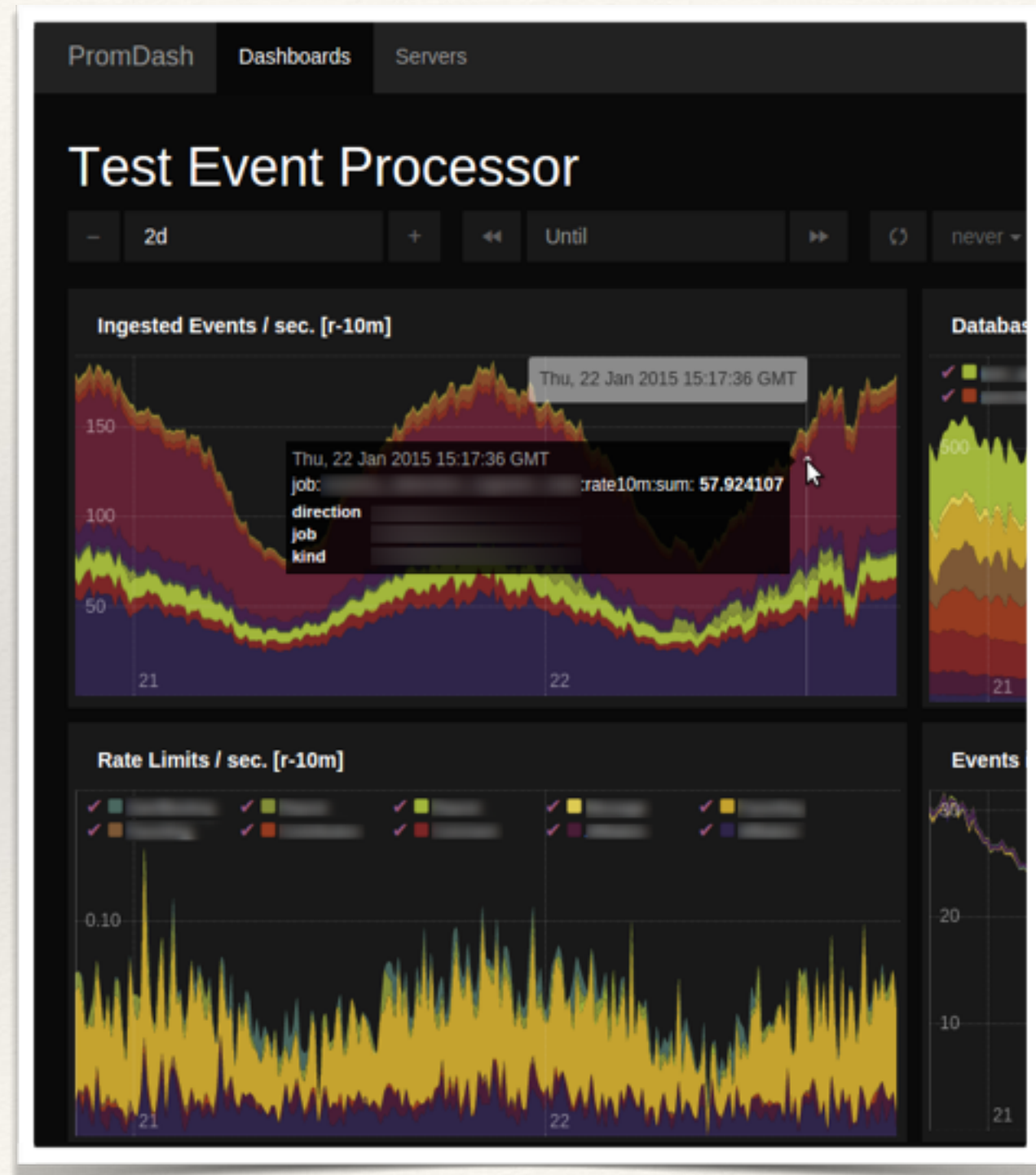
# PromDash





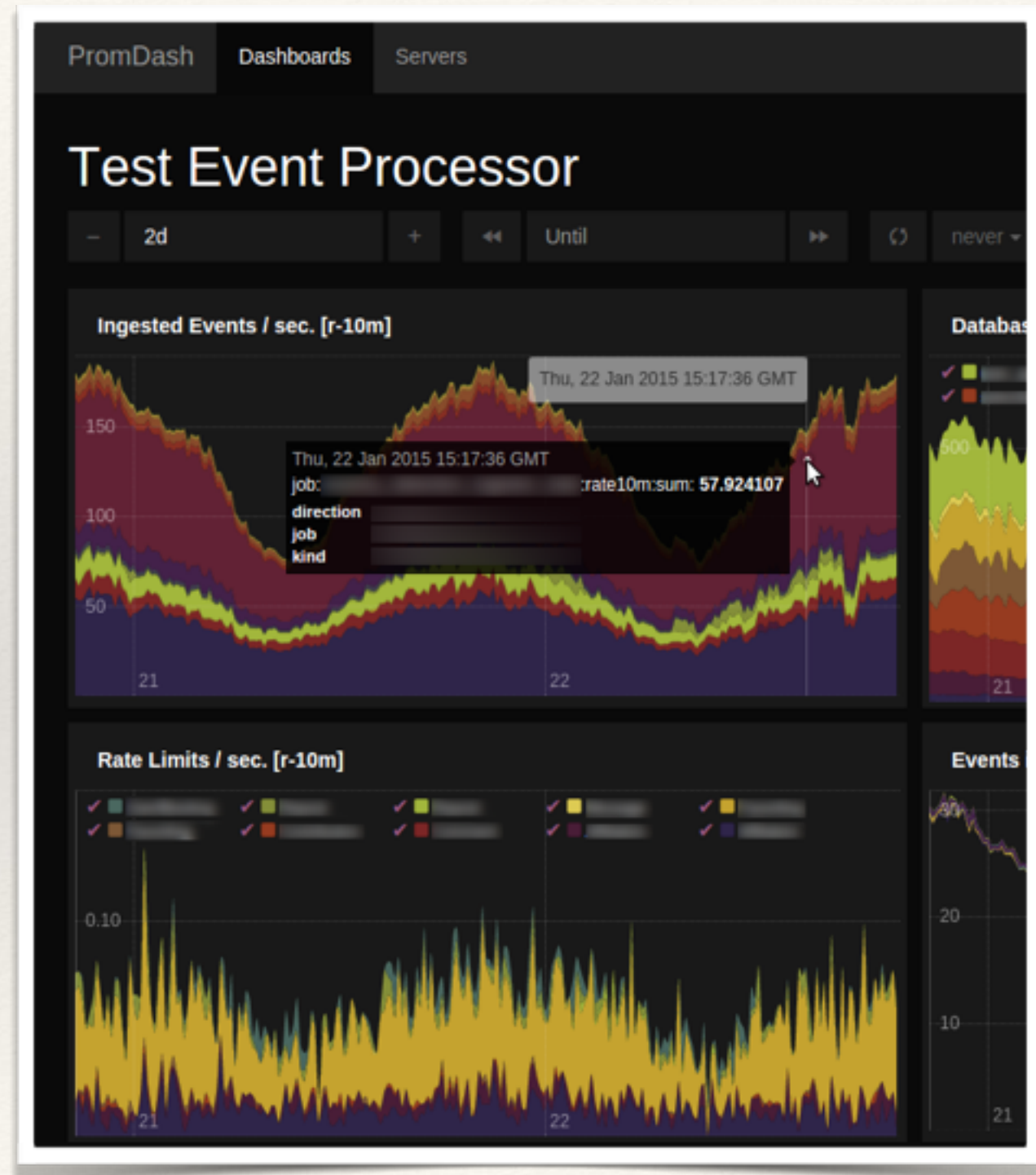
# PromDash

❖ best integration



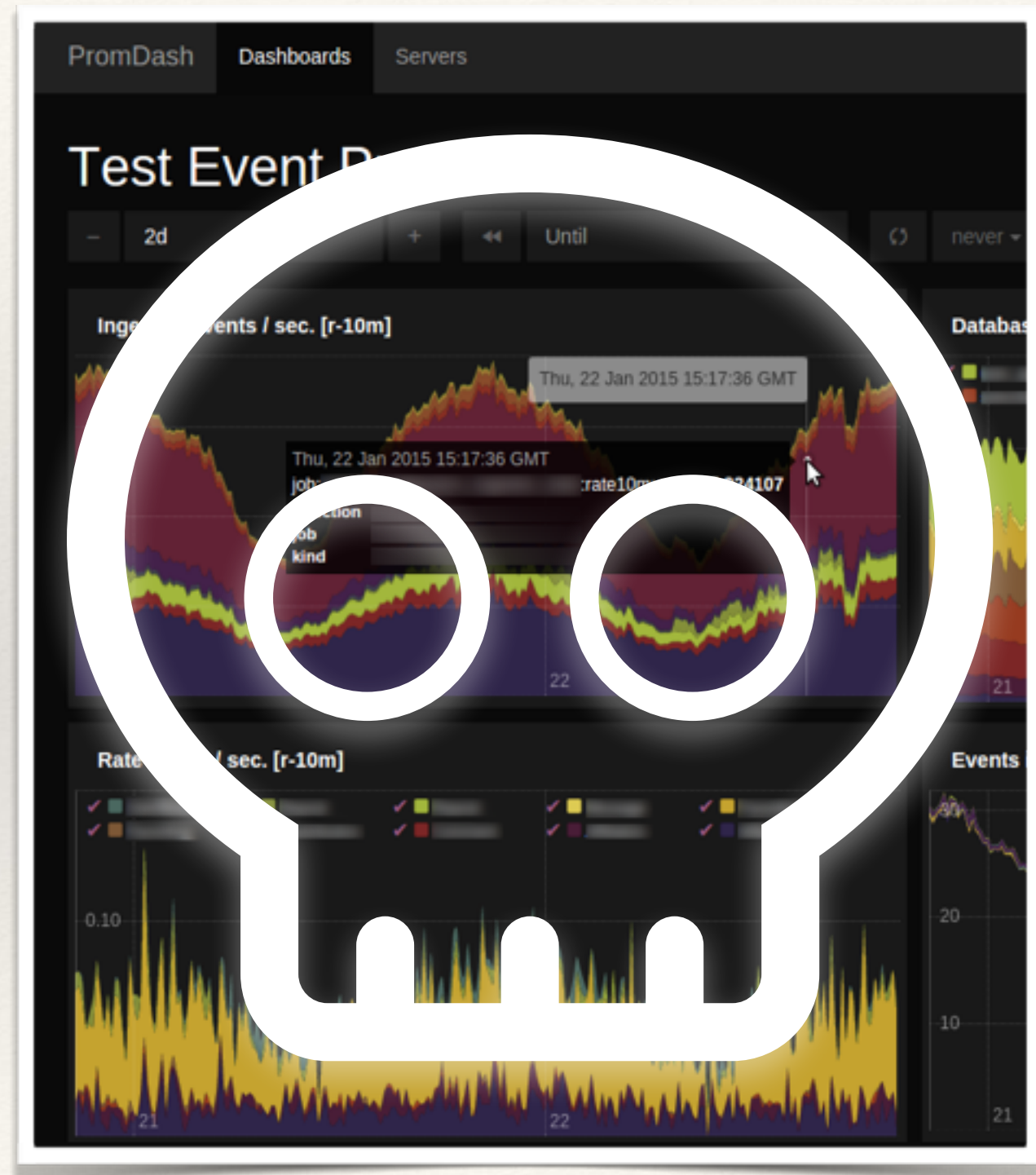
# PromDash

- ❖ best integration
- ❖ former official

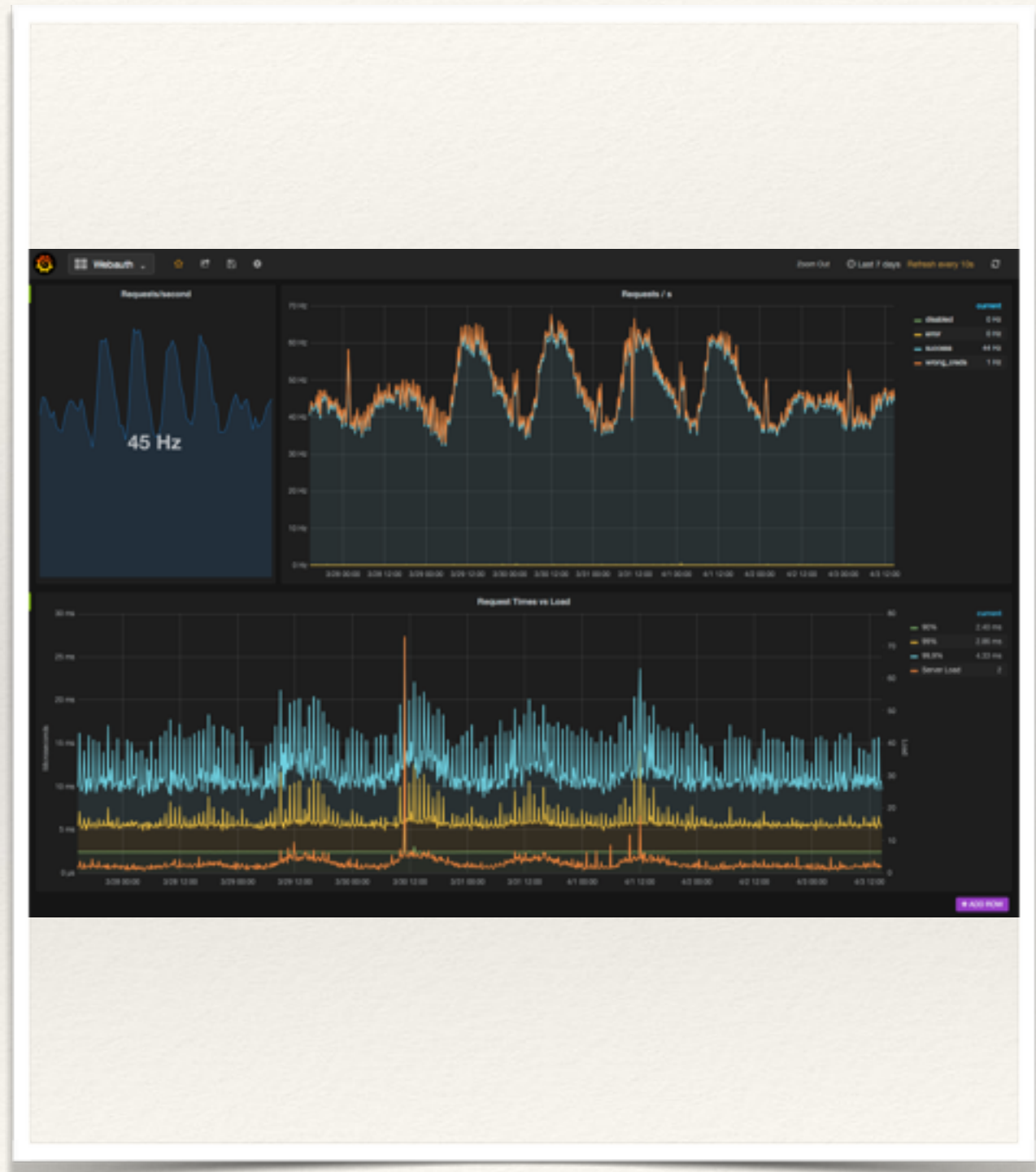


# PromDash

- ❖ best integration
- ❖ former official
- ❖ now deprecated
- ❖ don't bother

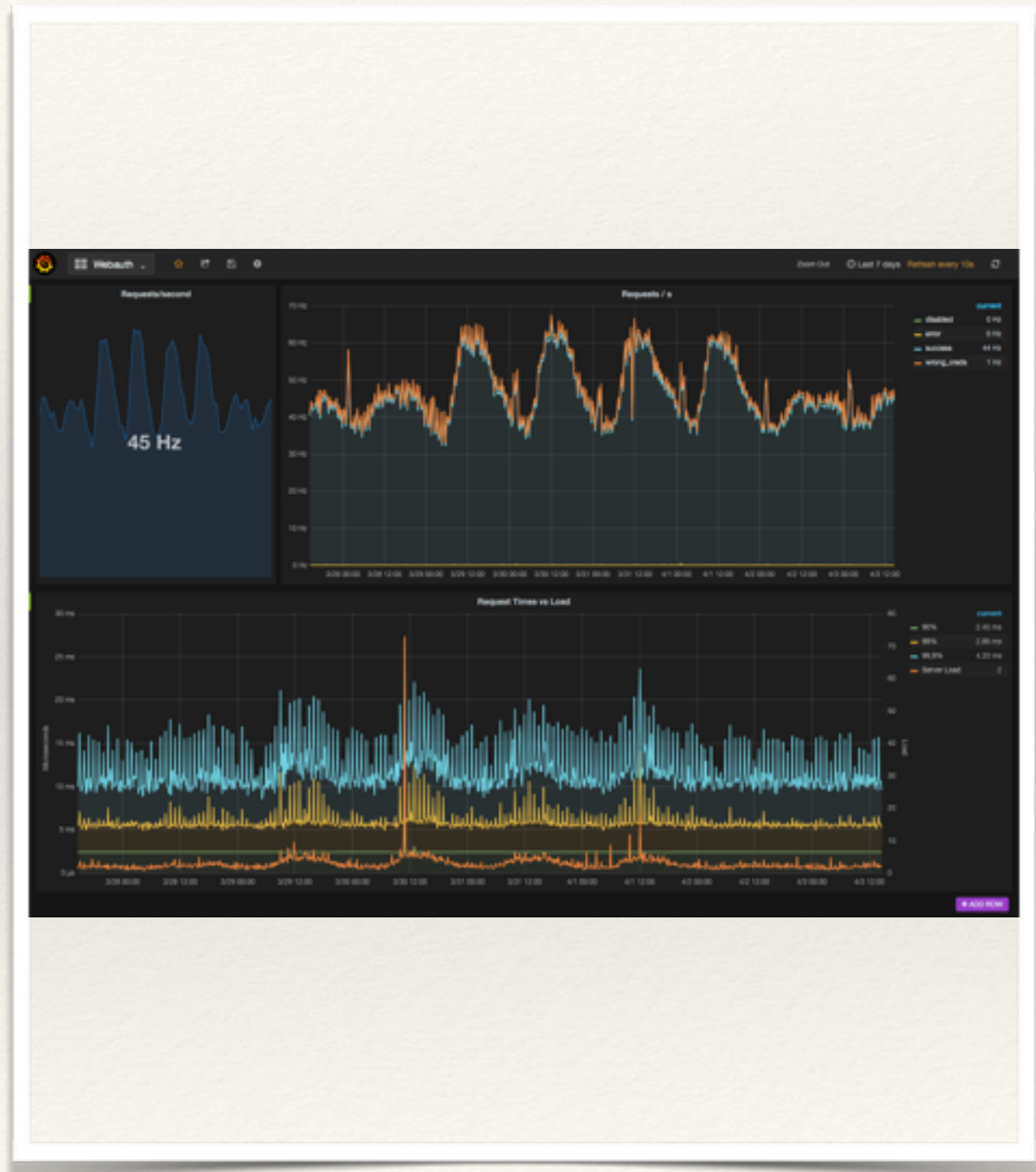


# Grafana



# Grafana

❖ pretty & powerful



# Grafana

- ❖ pretty & powerful
- ❖ *many* integrations



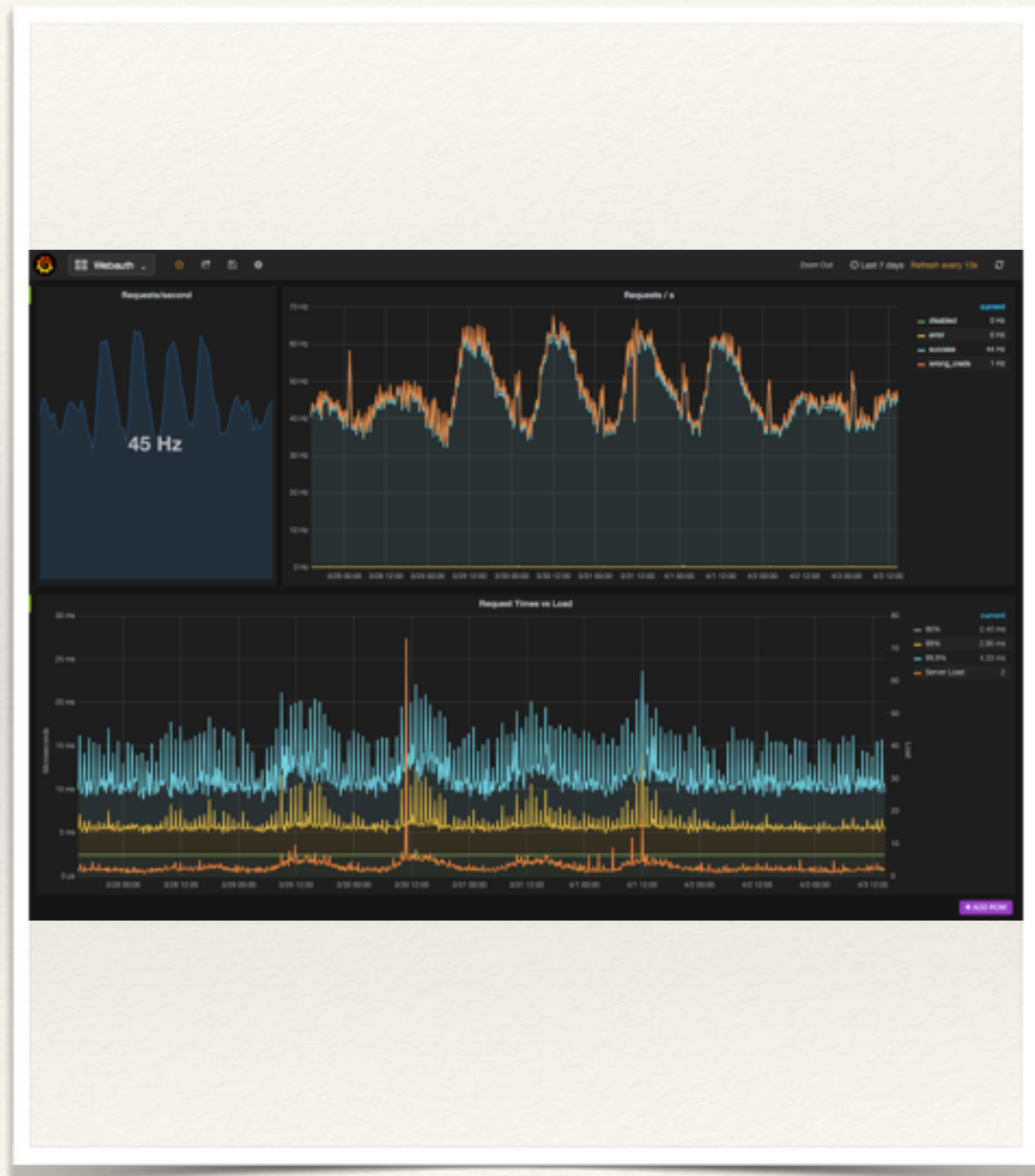
# Grafana

- ❖ pretty & powerful
- ❖ *many* integrations
- ❖ mix and match!

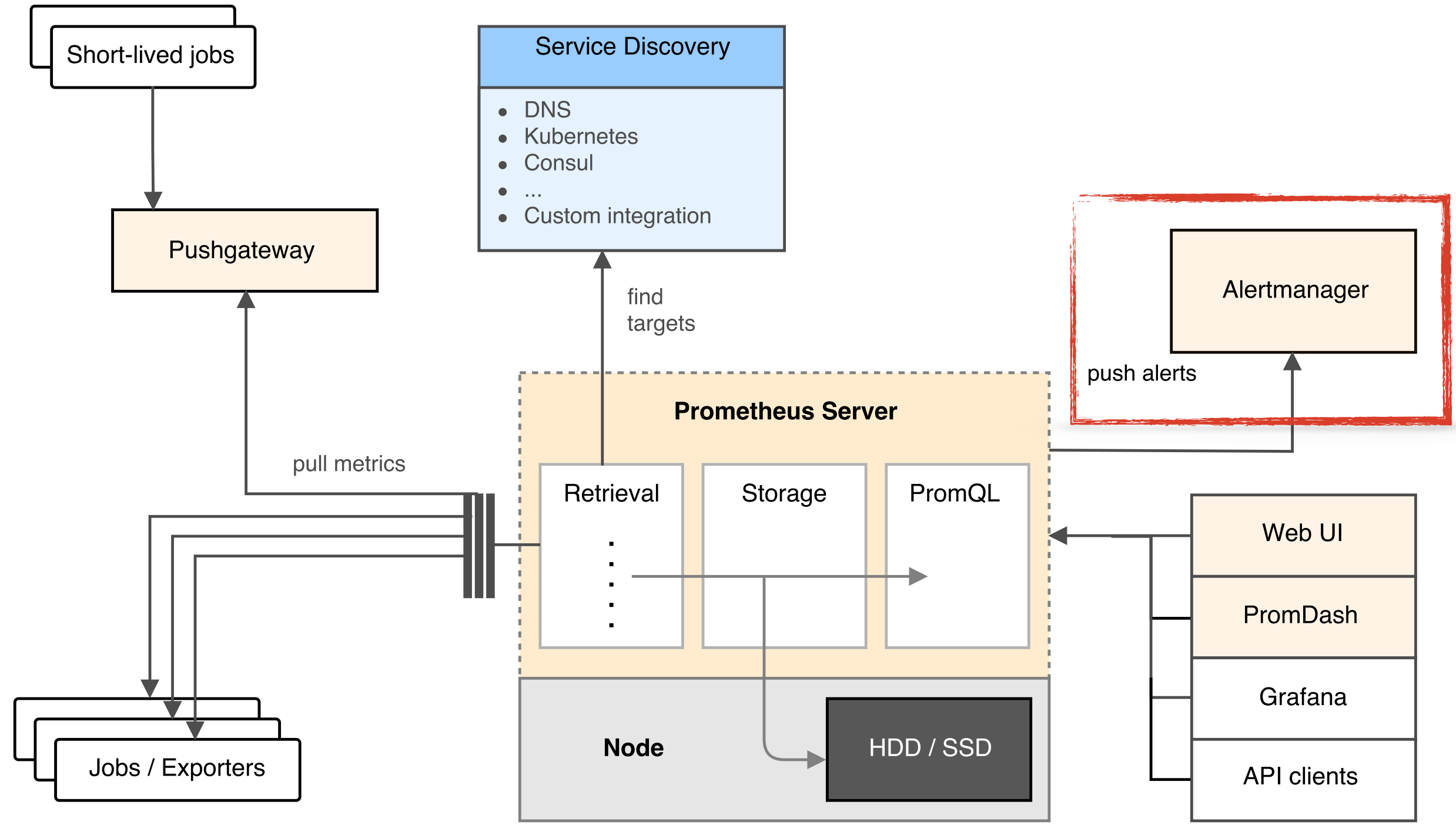


# Grafana

- ❖ pretty & powerful
- ❖ *many* integrations
- ❖ mix and match!
- ❖ use this!







---

# Alerts & Scrying

---

---

# Alerts & Scrying

---

```
ALERT DiskWillFillIn4Hours
```

```
IF predict_linear(
```

```
    node_filesystem_free[1h], 4*3600) < 0
```

```
FOR 5m
```

---

# Alerts & Scrying

---

```
ALERT DiskWillFillIn4Hours
  IF predict_linear(
    node_filesystem_free[1h], 4*3600) < 0
  FOR 5m
```

---

# Alerts & Scrying

---

```
ALERT DiskWillFillIn4Hours
  IF predict_linear(
    node_filesystem_free[1h], 4*3600) < 0
  FOR 5m
```

---

# Alerts & Scrying

---

```
ALERT DiskWillFillIn4Hours
  IF predict_linear(
    node_filesystem_free[1h], 4*3600) < 0
  FOR 5m
```

---

# Alerts & Scrying

---

```
ALERT DiskWillFillIn4Hours
  IF predict_linear(
    node_filesystem_free[1h], 4*3600) < 0
  FOR 5m
```

---

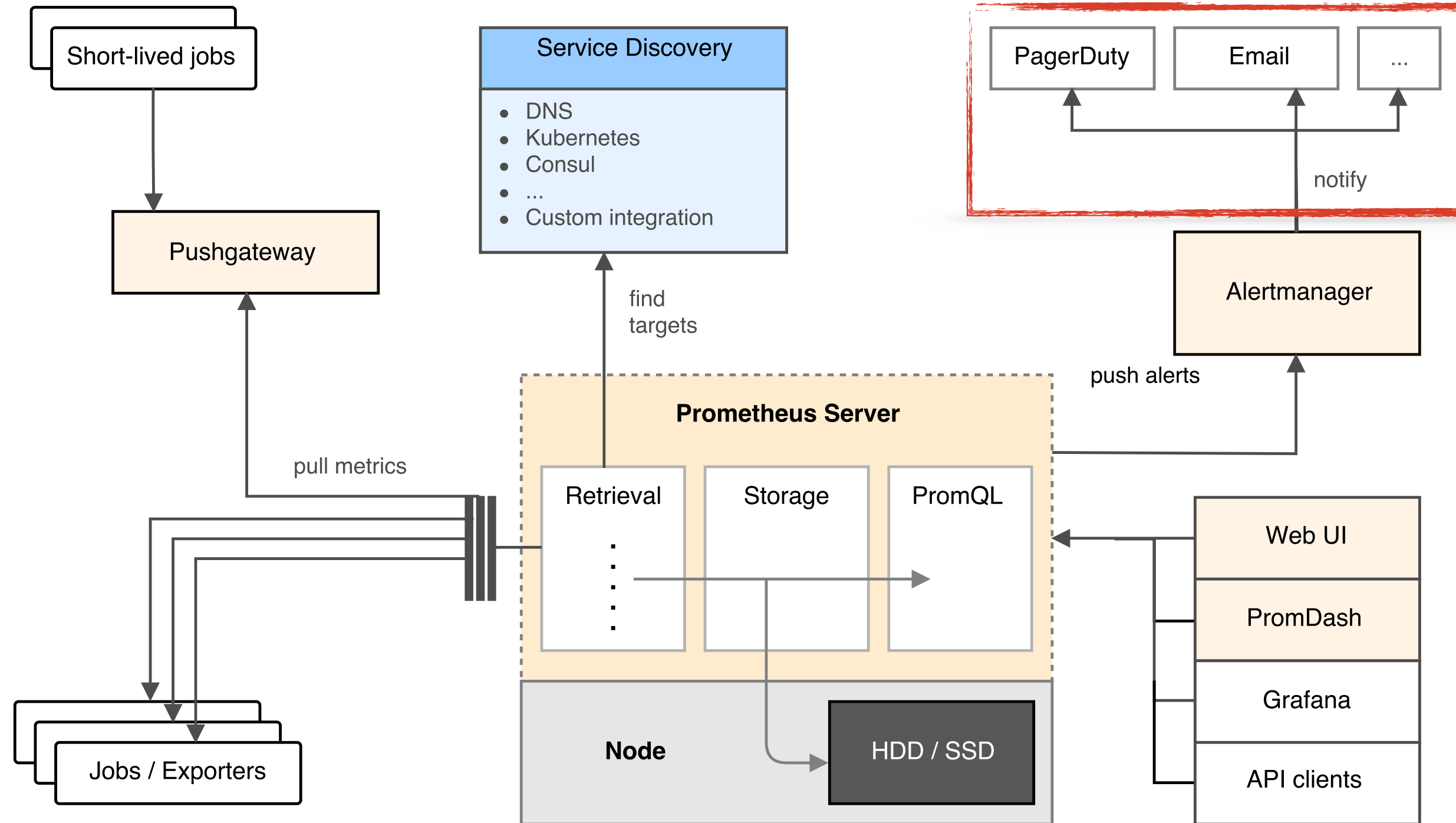
# Alerts & Scrying

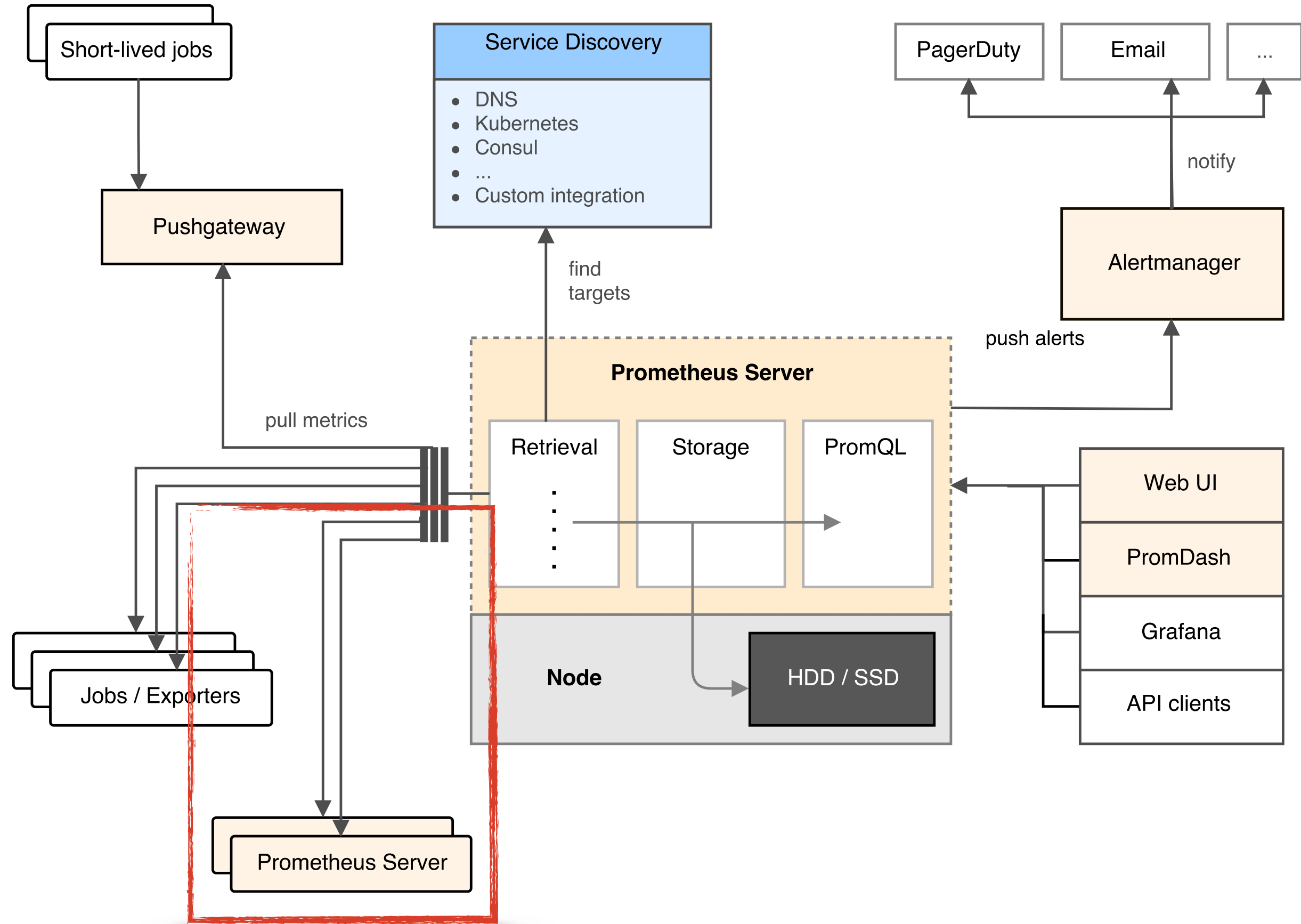
---

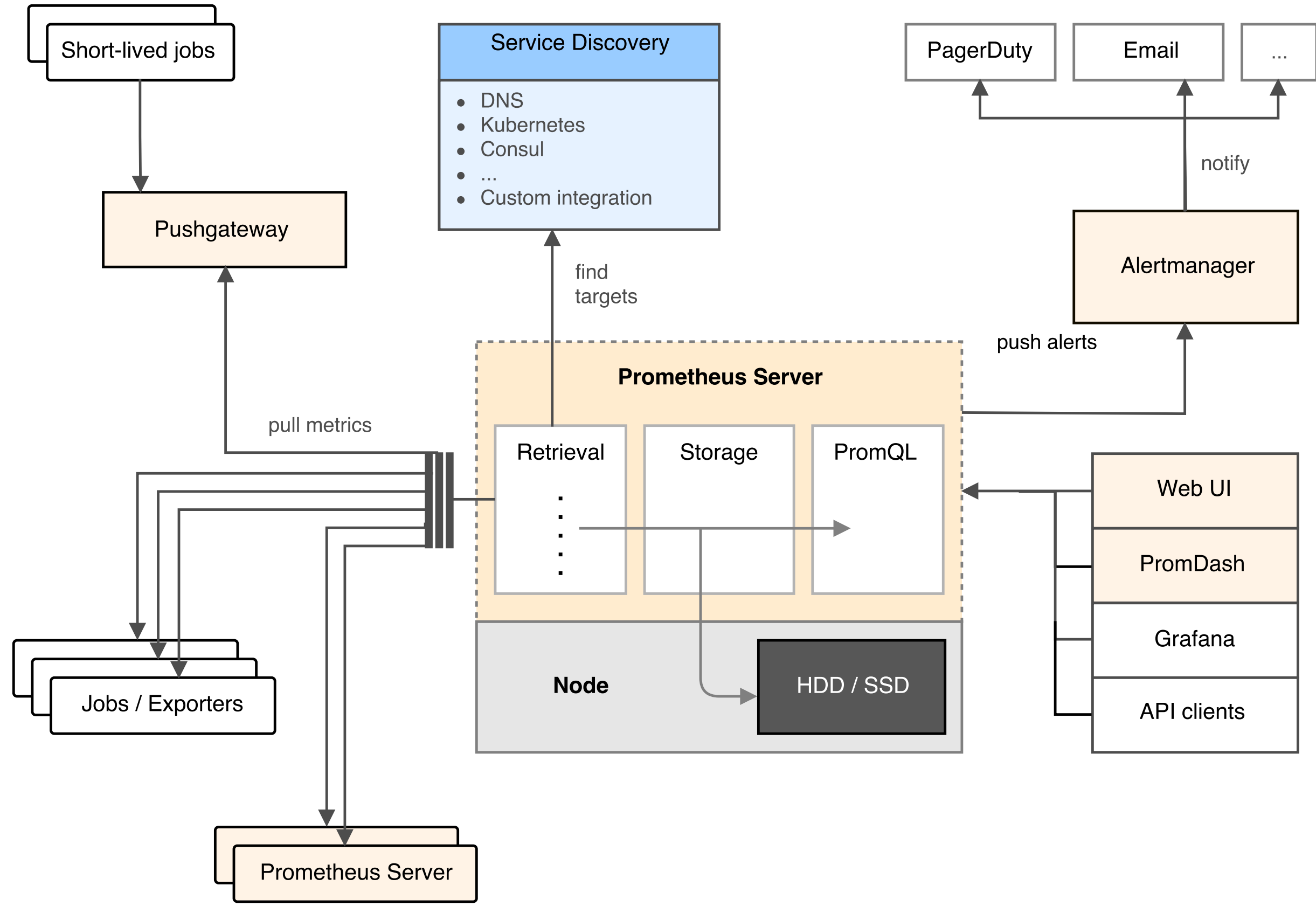
```
ALERT DiskWillFillIn4Hours
  IF predict_linear(
    node_filesystem_free[1h], 4*3600) < 0
```

```
FOR 5m
```





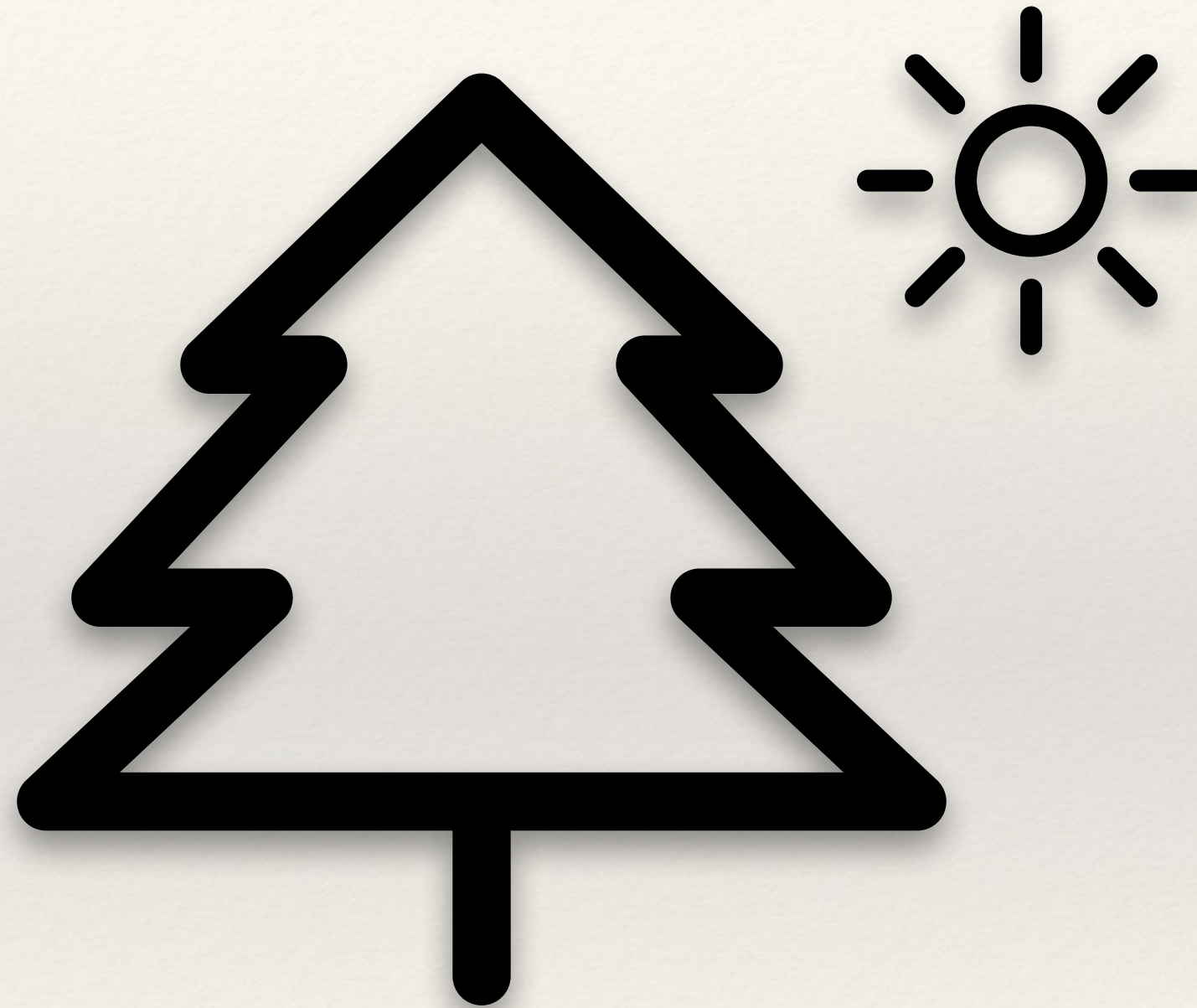




---

# Environment

---





HAProxy

MySQL

etcd

Consul

nginx

statsd

graphite

collectd

Django

redis

Kubernetes

PostgreSQL

SNMP

CouchDB

Varnish

InfluxDB

MongoDB

Apache

HAProxy

MySQL

etcd

Consul

nginx

statsd

graphite

collectd

Django

redis

Kubernetes

PostgreSQL

SNMP

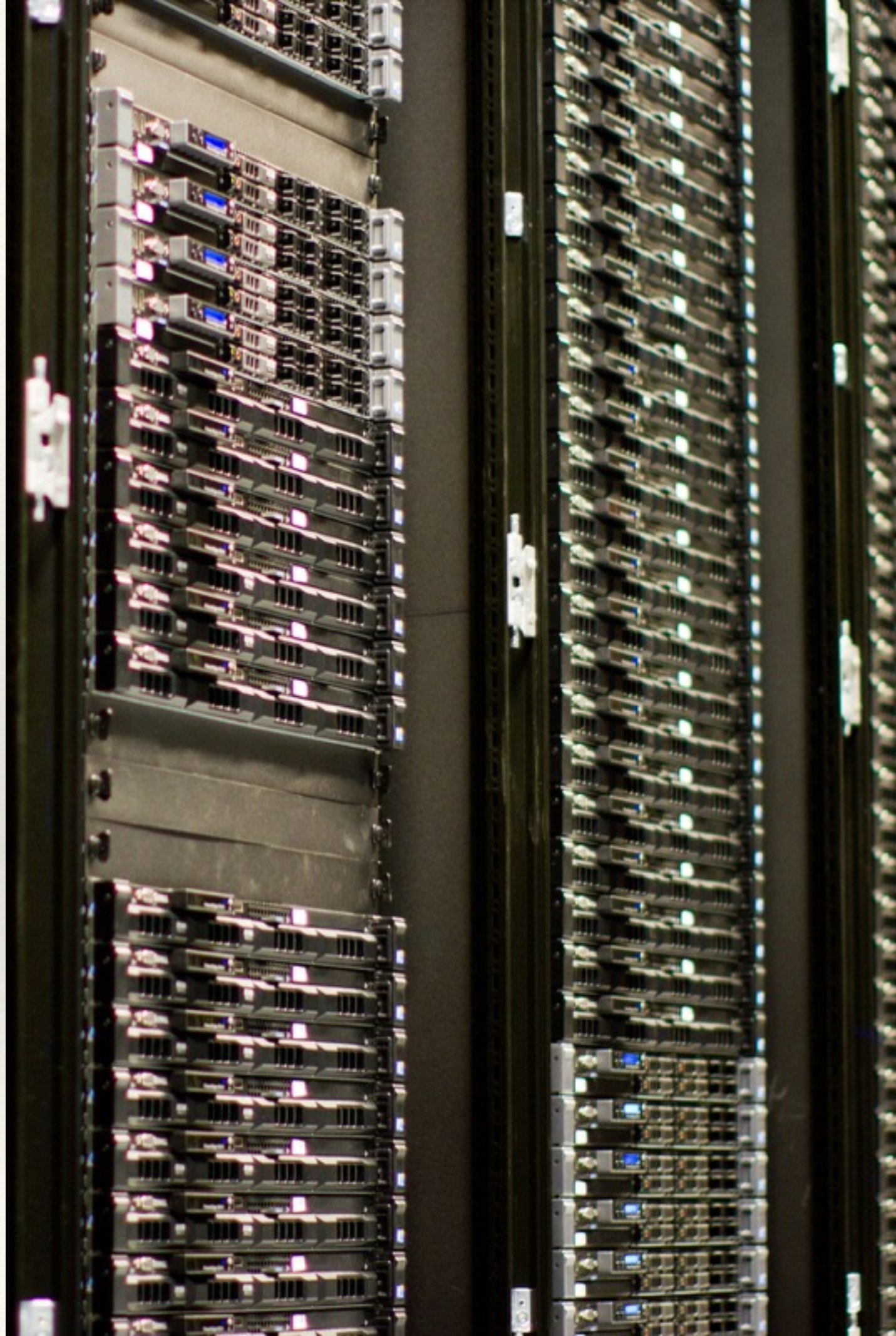
CouchDB

InfluxDB

Varnish

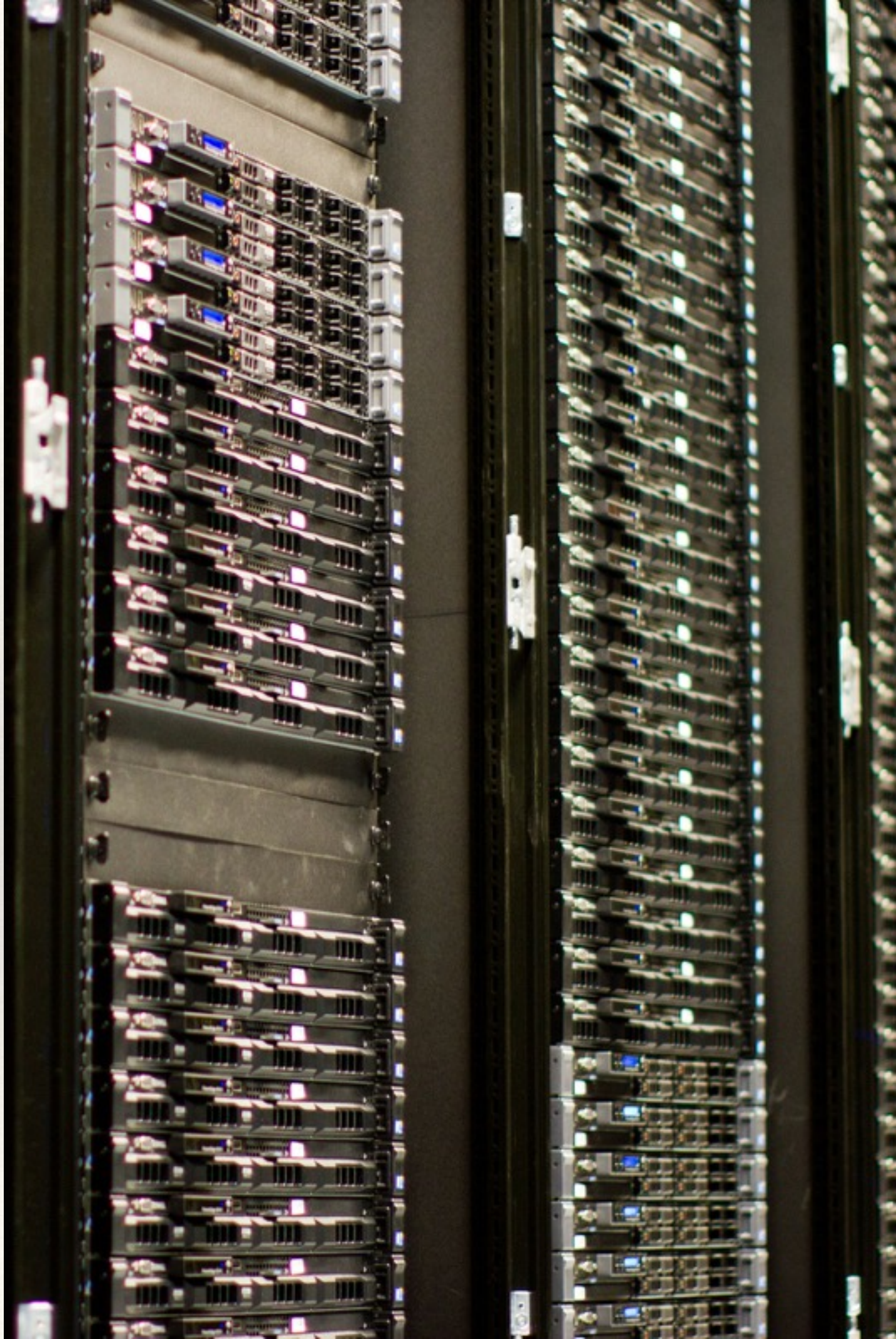
Apache

MongoDB



node\_exporter





cAdvisor

node\_exporter

# System Insight

❖ load

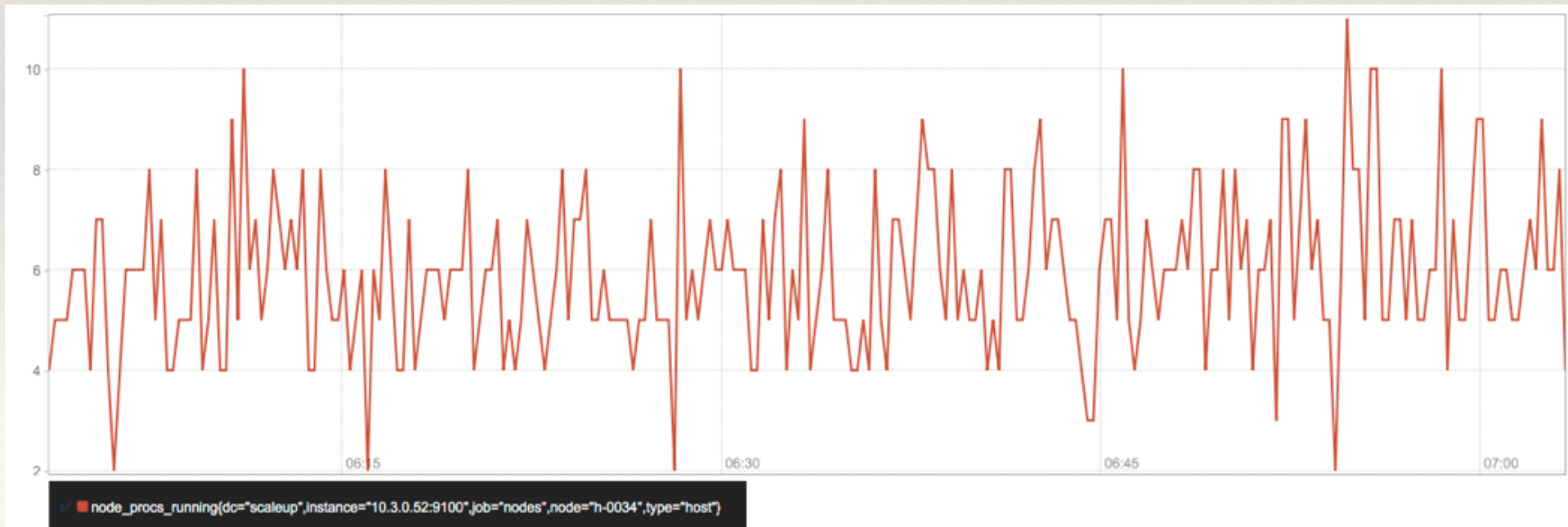
❖ memory

❖ disk

❖ procs

❖ network

❖ I/O



---

# mtail

---

---

# mtail

---

❖ follow (log) files

---

# mtail

---

- ❖ follow (log) files
- ❖ extract metrics using regex

---

# mtail

---

- ❖ follow (log) files
- ❖ extract metrics using regex
- ❖ can be better than direct

---

# Moar

---

---

# Moar

---

❖ **Edges:** web servers / HAProxy



---

# Moar

---

- ❖ **Edges:** web servers / HAProxy
- ❖ black box

---

# Moar

---

- ❖ **Edges: web servers / HAProxy**
- ❖ black box
- ❖ databases

---

# Moar

---

- ❖ **Edges: web servers / HAProxy**
- ❖ black box
- ❖ databases
- ❖ network

---

So Far

---

---

# So Far

---

❖ *system stats*

---

# So Far

---

- ❖ *system stats*
- ❖ *outside look*

---

# So Far

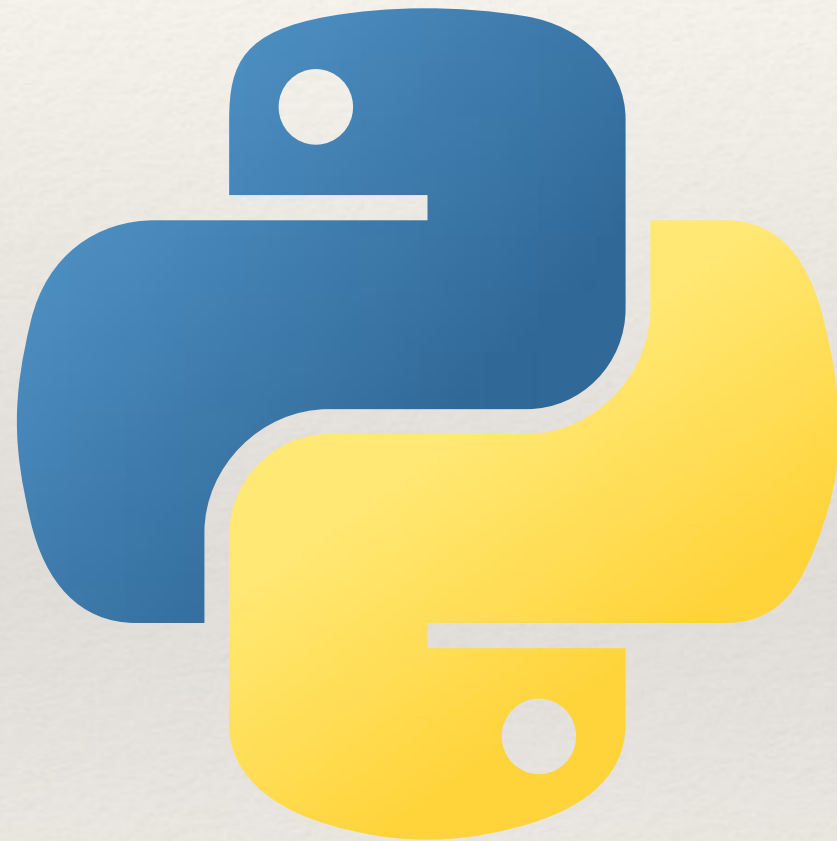
---

- ❖ system stats
- ❖ outside look
- ❖ 3rd party components

---

# Code

---





---

cat-or.not

---



---

# cat-or.not

---

❖ HTTP service



---

# cat-or.not

---

- ❖ HTTP service
- ❖ upload picture



---

# cat-or.not

---

- ❖ HTTP service
- ❖ upload picture
- ❖ meow! / nope



```
from flask import Flask, g, request
from cat_or_not import is_cat
```

```
app = Flask(__name__)
```

```
@app.route("/analyze", methods=["POST"])
```

```
def analyze():
```

```
    g.auth.check(request)
```

```
    return ("meow!"
```

```
           if is_cat(request.files["pic"])
```

```
           else "nope!")
```

```
if __name__ == "__main__":
```

```
    app.run()
```

```
from flask import Flask, g, request
from cat_or_not import is_cat
```

```
app = Flask(__name__)
```

```
@app.route("/analyze", methods=["POST"])
```

```
def analyze():
```

```
    g.auth.check(request)
```

```
    return ("meow!"
```

```
           if is_cat(request.files["pic"])
```

```
           else "nope!")
```

```
if __name__ == "__main__":
```

```
    app.run()
```

```
from flask import Flask, g, request
from cat_or_not import is_cat
```

```
app = Flask(__name__)
```

```
@app.route("/analyze", methods=["POST"])
```

```
def analyze():
```

```
    g.auth.check(request)
```

```
    return ("meow!"
```

```
            if is_cat(request.files["pic"])
```

```
            else "nope!")
```

```
if __name__ == "__main__":
```

```
    app.run()
```

```
pip install prometheus_client
```



```
from prometheus_client import \
    start_http_server
# ...
if __name__ == "__main__":
    start_http_server(8000)
    app.run()
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```

```
process_virtual_memory_bytes 156393472.0  
process_resident_memory_bytes 20480000.0  
process_start_time_seconds 1460214325.21  
process_cpu_seconds_total 0.169999999999998  
process_open_fds 8.0  
process_max_fds 1024.0
```





```
from prometheus_client import \  
    Histogram, Gauge
```

```
REQUEST_TIME = Histogram(  
    "cat_or_not_request_seconds",  
    "Time spent in HTTP requests.")
```

```
from prometheus_client import \  
    Histogram, Gauge
```

```
REQUEST_TIME = Histogram(  
    "cat_or_not_request_seconds",  
    "Time spent in HTTP requests.")  
ANALYZE_TIME = Histogram(  
    "cat_or_not_analyze_seconds",  
    "Time spent analyzing pictures.")
```

```
from prometheus_client import \  
    Histogram, Gauge
```

```
REQUEST_TIME = Histogram(  
    "cat_or_not_request_seconds",  
    "Time spent in HTTP requests.")
```

```
ANALYZE_TIME = Histogram(  
    "cat_or_not_analyze_seconds",  
    "Time spent analyzing pictures.")
```

```
IN_PROGRESS = Gauge(  
    "cat_or_not_in_progress_requests",  
    "Number of requests in progress.")
```

```
@app.route("/analyze", methods=["POST"])
@IN_PROGRESS.track_inprogress()
@REQUEST_TIME.time()
def analyze():
    g.auth.check(request)
    with ANALYZE_TIME.time():
        result = is_cat(
            request.files["pic"].stream)

    return "meow!" if result else "nope!"
```

```
@app.route("/analyze", methods=["POST"])
```

```
@IN_PROGRESS.track_inprogress()
```

```
@REQUEST_TIME.time()
```

```
def analyze():
```

```
    g.auth.check(request)
```

```
    with ANALYZE_TIME.time():
```

```
        result = is_cat(
```

```
            request.files["pic"].stream)
```

```
    return "meow!" if result else "nope!"
```

```
AUTH_TIME = Histogram("auth_seconds",  
                      "Time spent authenticating.")  
AUTH_ERRS = Counter("auth_errors_total",  
                   "Errors while authing.")  
AUTH_WRONG_CREDS = Counter("auth_wrong_creds_total",  
                           "Wrong credentials.")
```

```
class Auth:
```

```
    # ...
```

```
    @AUTH_TIME.time()
```

```
    def auth(self, request):
```

```
        while True:
```

```
            try:
```

```
                return self._auth(request)
```

```
            except WrongCredsError:
```

```
                AUTH_WRONG_CREDS.inc()
```

```
                raise
```

```
            except Exception:
```

```
                AUTH_ERRS.inc()
```

```
AUTH_TIME = Histogram("auth_seconds",  
                      "Time spent authenticating.")
```

```
AUTH_ERRS = Counter("auth_errors_total",  
                   "Errors while authing.")
```

```
AUTH_WRONG_CREDS = Counter("auth_wrong_creds_total",  
                           "Wrong credentials.")
```

```
class Auth:
```

```
    # ...
```

```
    @AUTH_TIME.time()
```

```
    def auth(self, request):
```

```
        while True:
```

```
            try:
```

```
                return self._auth(request)
```

```
            except WrongCredsError:
```

```
                AUTH_WRONG_CREDS.inc()
```

```
                raise
```

```
            except Exception:
```

```
                AUTH_ERRS.inc()
```

```
AUTH_TIME = Histogram("auth_seconds",  
                      "Time spent authenticating.")  
AUTH_ERRS = Counter("auth_errors_total",  
                   "Errors while authing.")  
AUTH_WRONG_CREDS = Counter("auth_wrong_creds_total",  
                           "Wrong credentials.")
```

```
class Auth:
```

```
    # ...
```

```
    @AUTH_TIME.time()
```

```
    def auth(self, request):
```

```
        while True:
```

```
            try:
```

```
                return self._auth(request)
```

```
            except WrongCredsError:
```

```
                AUTH_WRONG_CREDS.inc()
```

```
                raise
```

```
            except Exception:
```

```
                AUTH_ERRS.inc()
```



```
AUTH_TIME = Histogram("auth_seconds",  
                      "Time spent authenticating.")
```

```
AUTH_ERRS = Counter("auth_errors_total",  
                   "Errors while authing.")
```

```
AUTH_WRONG_CREDS = Counter("auth_wrong_creds_total",  
                           "Wrong credentials.")
```

```
class Auth:
```

```
    # ...
```

```
    @AUTH_TIME.time()
```

```
    def auth(self, request):
```

```
        while True:
```

```
            try:
```

```
                return self._auth(request)
```

```
            except WrongCredsError:
```

```
                AUTH_WRONG_CREDS.inc()
```

```
                raise
```

```
            except Exception:
```

```
                AUTH_ERRS.inc()
```

```
@app.route("/analyze", methods=["POST"])
def analyze():
    g.auth.check(request)
    with ANALYZE_TIME.time():
        result = is_cat(
            request.files["pic"].stream)

    return "meow!" if result else "nope!"
```

```
pip install prometheus_async
```

---

# Wrapper

---

```
from prometheus_async.aio import time
```

```
@time(REQUEST_TIME)
async def view(request):
    # ...
```

---

# Goodies

---

---

# Goodies

---

- ❖ aiohttp-based metrics export

---

# Goodies

---

- ❖ aiohttp-based metrics export
- ❖ also in thread!

---

# Goodies

---

- ❖ aiohttp-based metrics export
  - ❖ also in thread!
- ❖ Consul Agent integration



---

# Wrap Up

---

---

# Wrap Up

---



---

# Wrap Up

---



---

# Wrap Up

---



---

# Wrap Up

---



ox.cx /

p

@

hynek

vrrmd.de