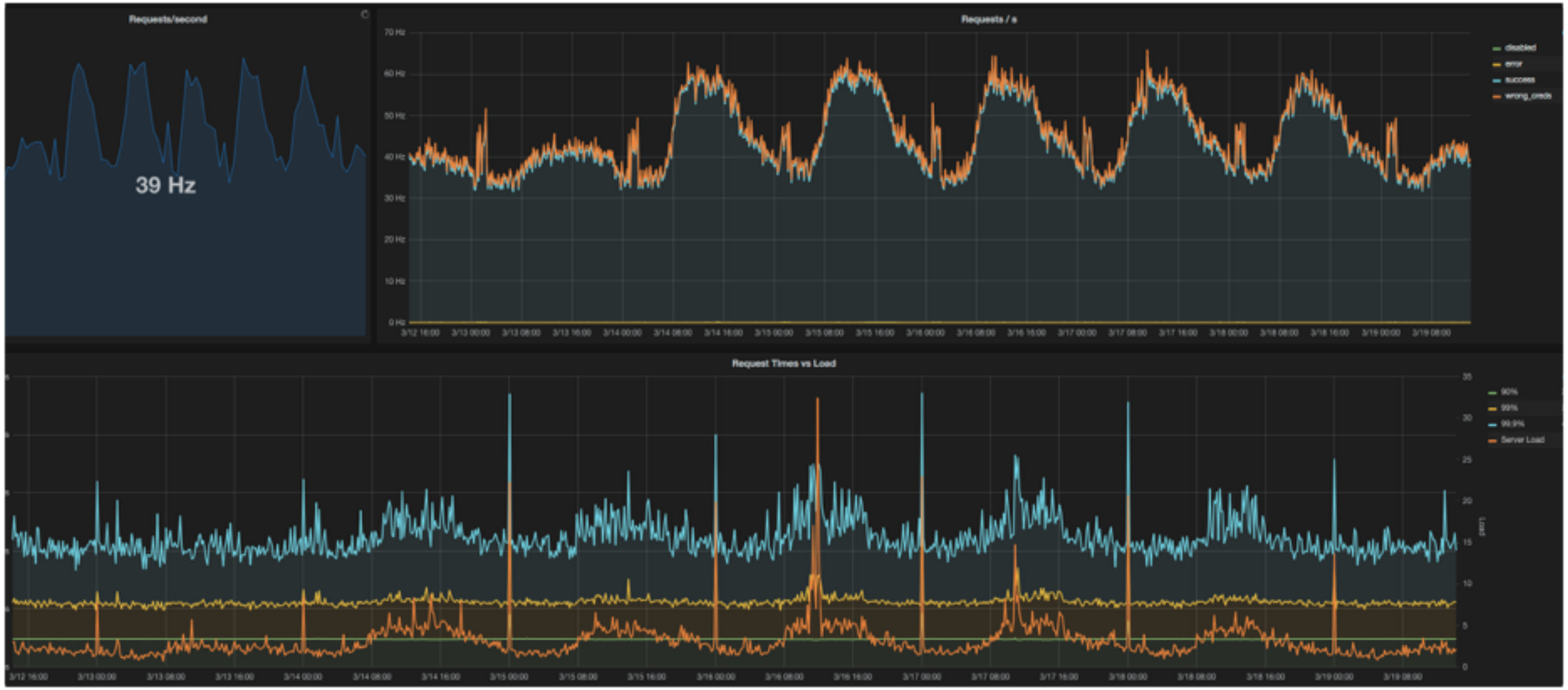


Hynek Schlawack

Get Instrumented

How Prometheus Can Unify Your Metrics

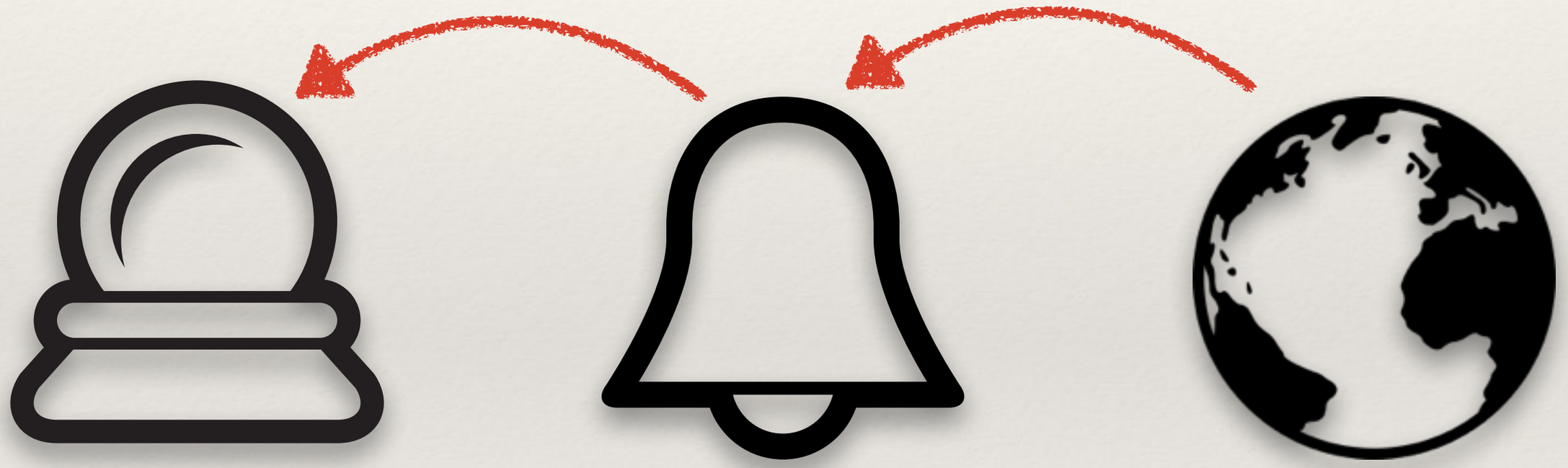
Goals

Goals

Goals

Goals

Goals



Service Level

Service Level **Indicator**

Service Level **Indicator**

Objective

Service Level **Indicator**

Objective

(Agreement)

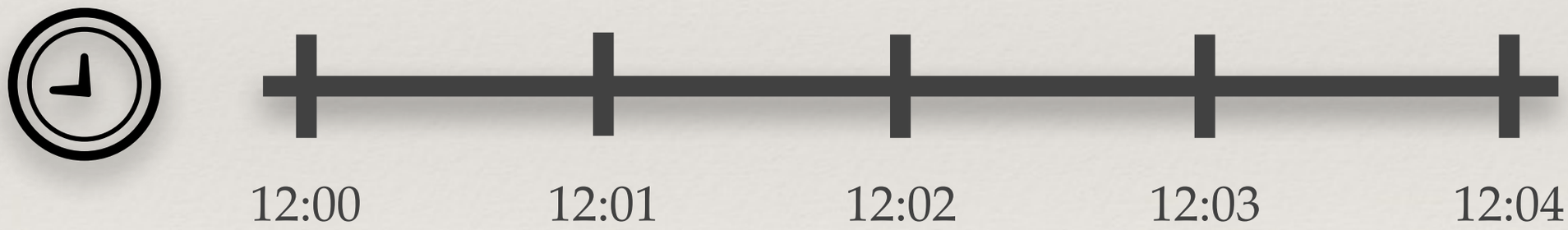
Metrics

Metrics

avg latency	0.3	0.5	0.8	1.1	2.6

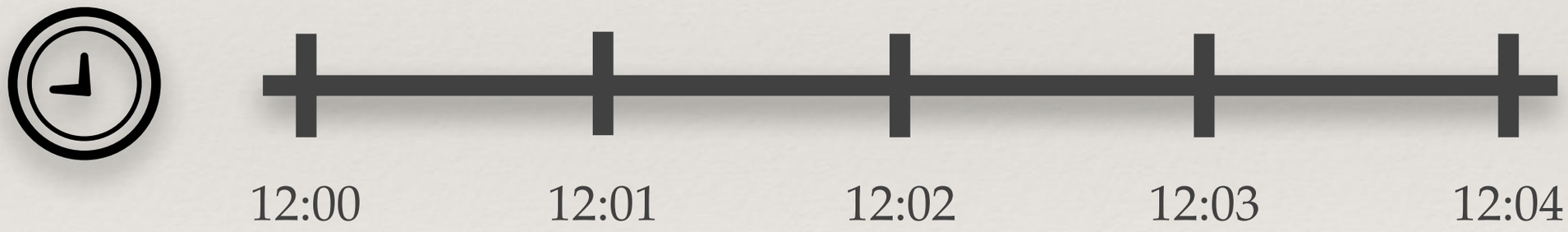
Metrics

avg latency `0.3` `0.5` `0.8` `1.1` `2.6`



Metrics

avg latency	0.3	0.5	0.8	1.1	2.6
server load	0.3	1.0	2.3	3.5	5.2

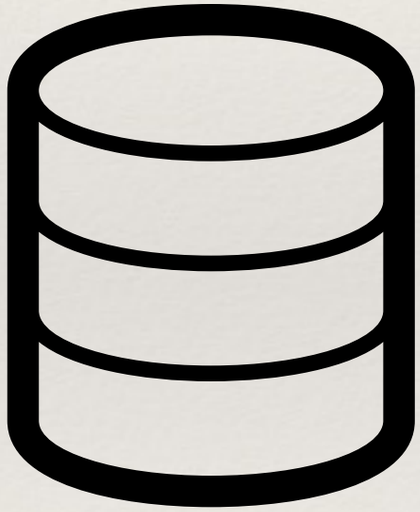



F-WWOW
AJ-DM
1020
03000
Std
DISCONTINUITY
DISCONTINUITY
CANBERRA
121.700
130.350

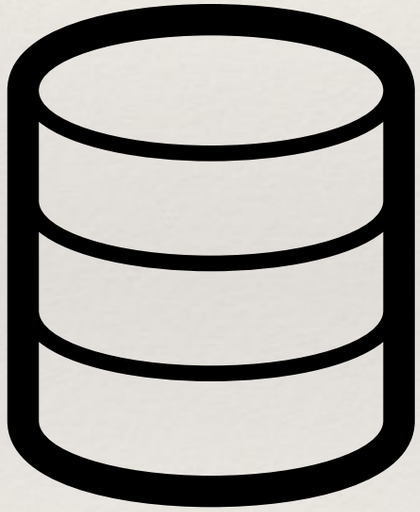
Instrument

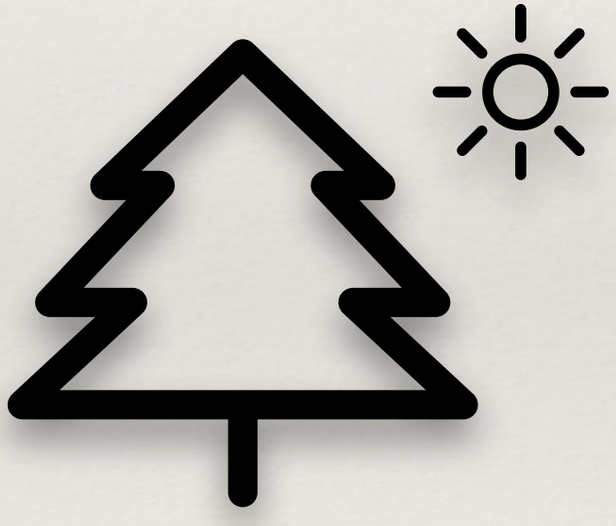
Instrument

Instrument

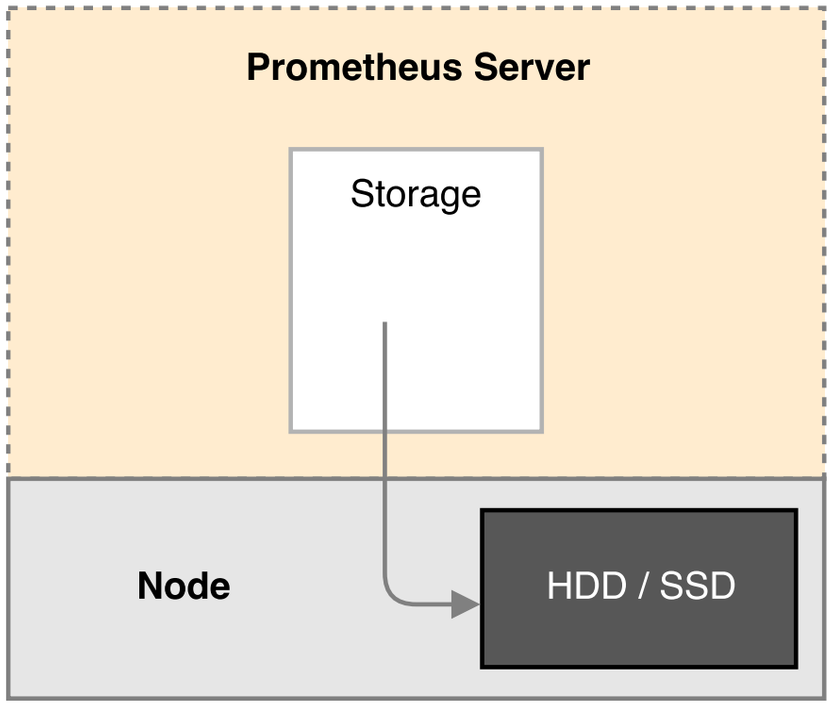


Instrument





Instrument


Prometheus Server
Storage
Node
HDD / SSD

Metric Types

Metric Types

- counter

Metric Types

- counter
- gauge

Metric Types

- counter
- summary
- gauge

Metric Types

- counter
- summary
- gauge
- histogram

Metric Types

- counter
- gauge
- summary
- histogram
  - buckets (1s, 0.5s, 0.25, …)

Averages

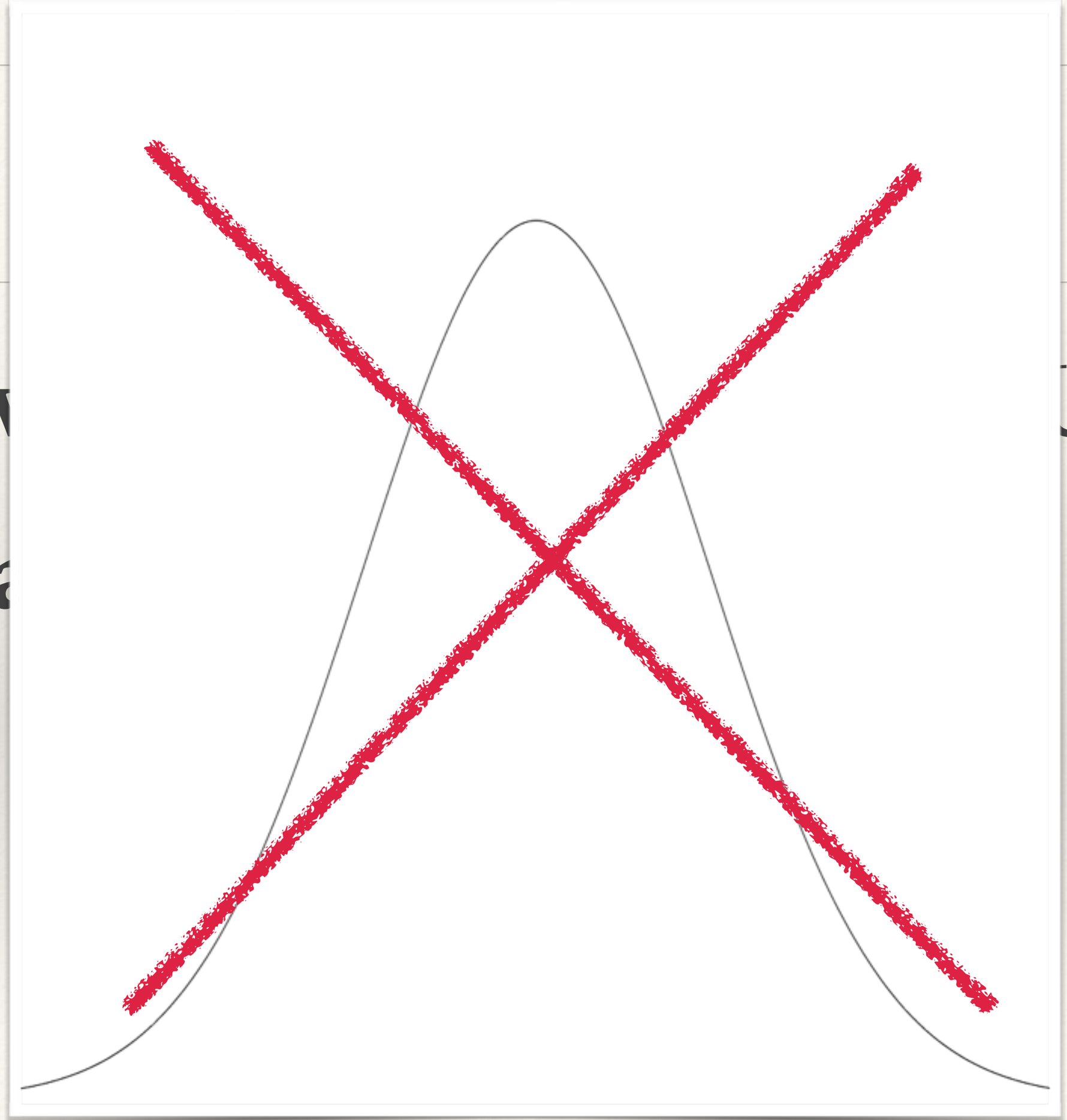
Averages

- **avg**(request time) ≠ **avg**(UX)

Averages

- **avg**(request time) ≠ **avg**(UX)
- **avg**($\{1, 1, 1, 1, 10\}$) $= 2.8$

- **av** UX)
- **a**



Averages

- **avg**(request time) ≠ **avg**(UX)
- **avg**($\{1, 1, 1, 1, 10\}$) $= 2.8$

Averages

- **avg**(request time) ≠ **avg**(UX)
  - **avg**($\{1, 1, 1, 1, 10\}$) $= 2.8$
- **median**($\{1, 1, 1, 1, 10\}$) $=$ **1**

Averages

- **avg**(request time) ≠ **avg**(UX)
  - **avg**({1, 1, 1, 1, 10}) = 2.8
- **median**({1, 1, 1, 1, 10}) = **1**

Averages

- **avg**(request time) ≠ **avg**(UX)
  - **avg**({1, 1, 1, 1, 10}) = 2.8
  - **median**({1, 1, 1, 1, 10}) = **1**
  - **median**({1, 1, 100_000}) = **1**

Percentiles

Percentiles

nth percentile P of a data set

=

P ≥ **n**% of values

50^{th} percentile = **1 ms**

50^{th} percentile = **1 ms**

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

Percentiles

Percentiles

P	{1, 1, 100_000}
50th	1

Percentiles

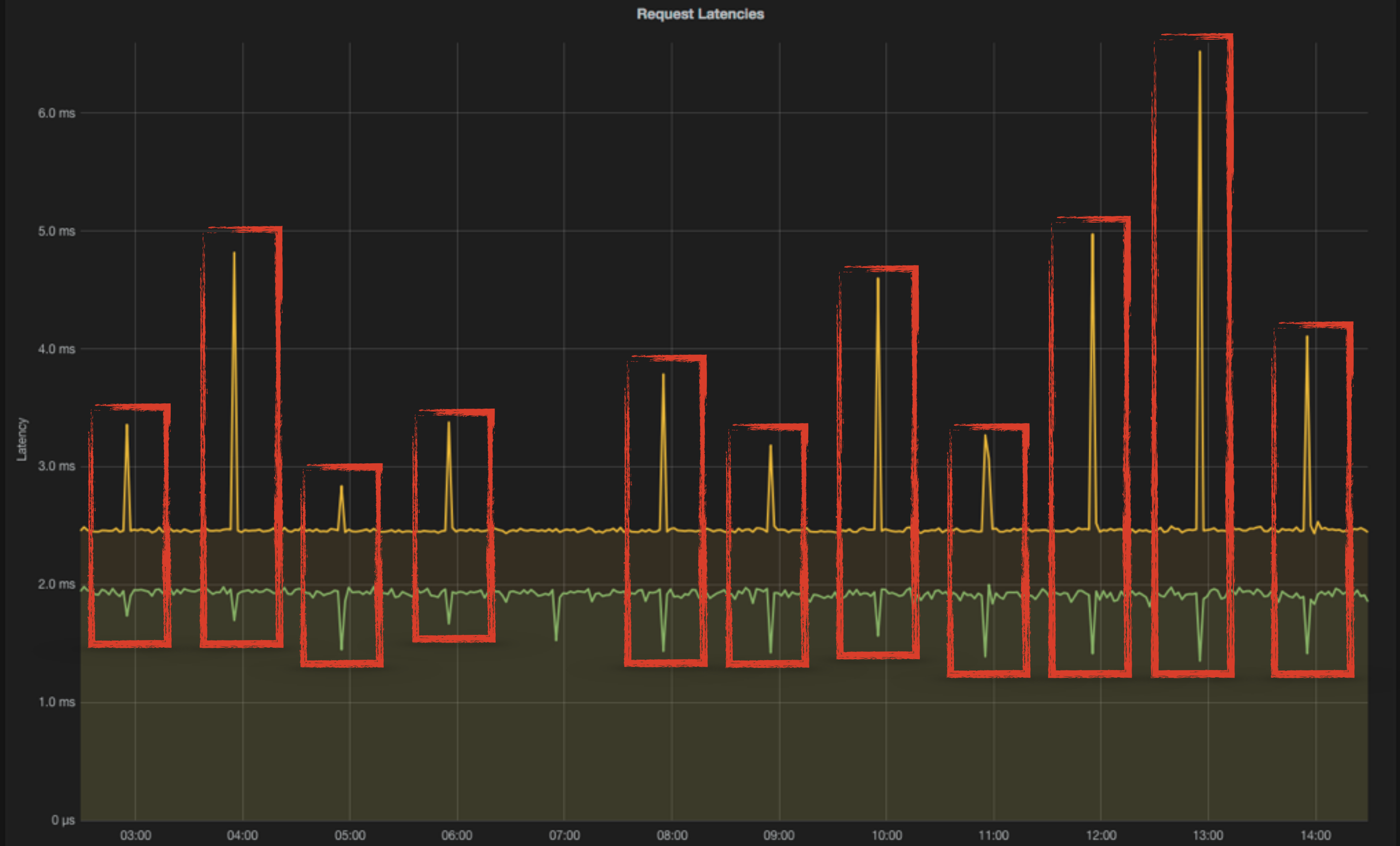
P	{1, 1, 100_000}
50th	1
95th	90_000

Request Latencies

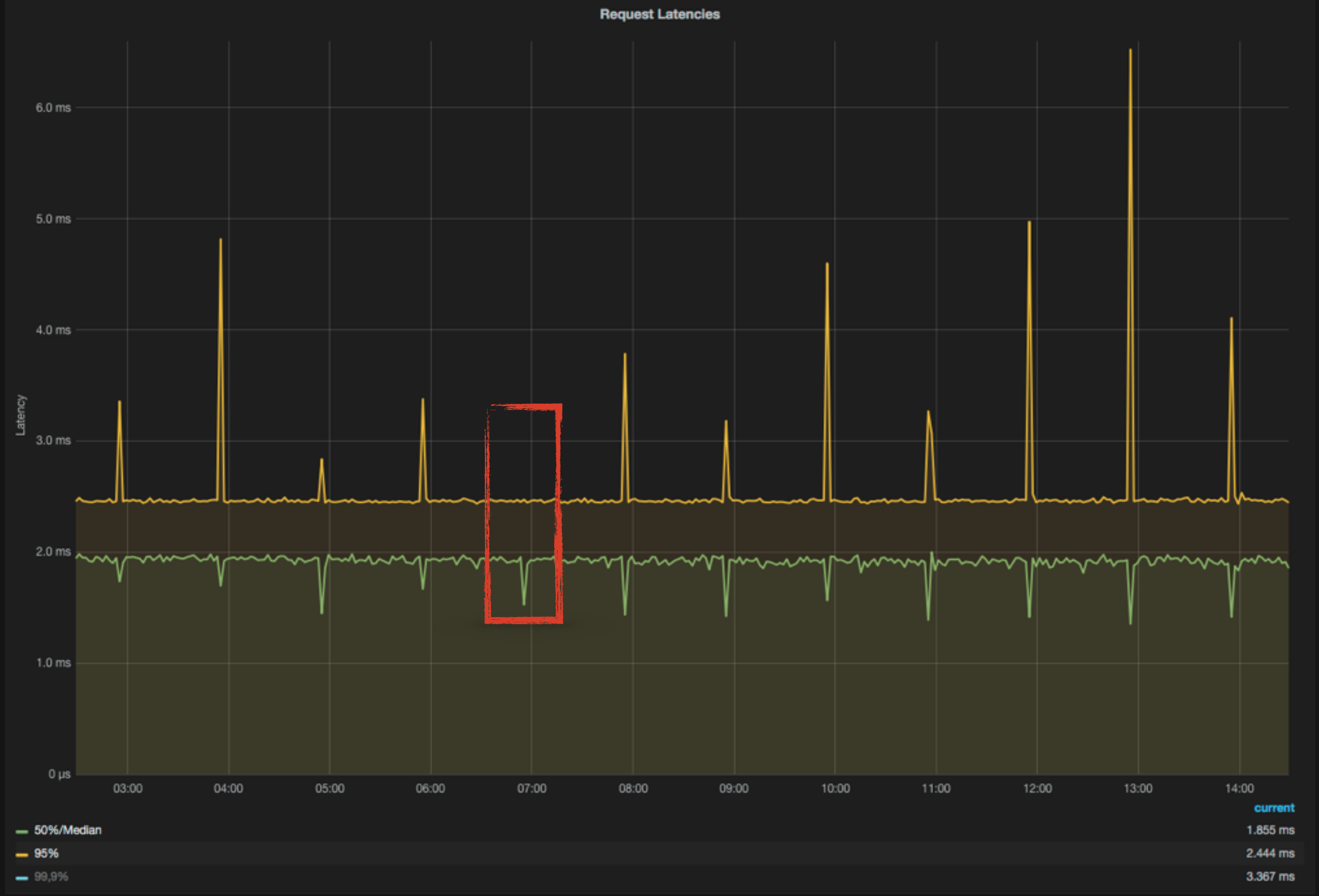
6.0 ms
5.0 ms
4.0 ms
3.0 ms
2.0 ms
1.0 ms
0 µs
Latency

03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00

	current
50%/Median	1.855 ms
95%	2.444 ms
99,9%	3.367 ms



	current
50%/Median	1.855 ms
95%	2.444 ms
99,9%	3.367 ms


Request Latencies
6.0 ms
5.0 ms
4.0 ms
3.0 ms
2.0 ms
1.0 ms
0 µs
Latency
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
current
50%/Median
1.855 ms
95%
2.444 ms
99,9%
3.367 ms

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

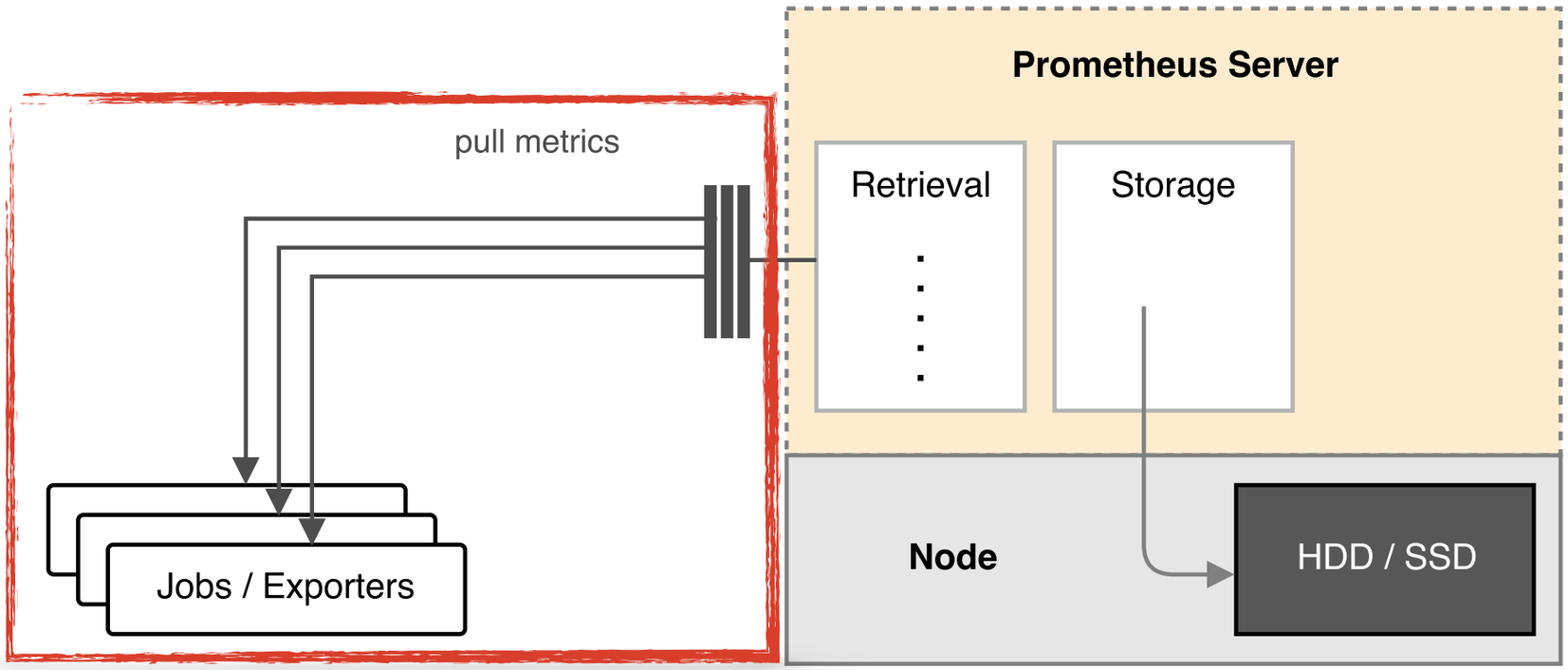
~~backend1_app_http_reqs_msgs_post~~
~~backend1_app_http_reqs_msgs_get~~
…
app_http_reqs_total

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"}
…

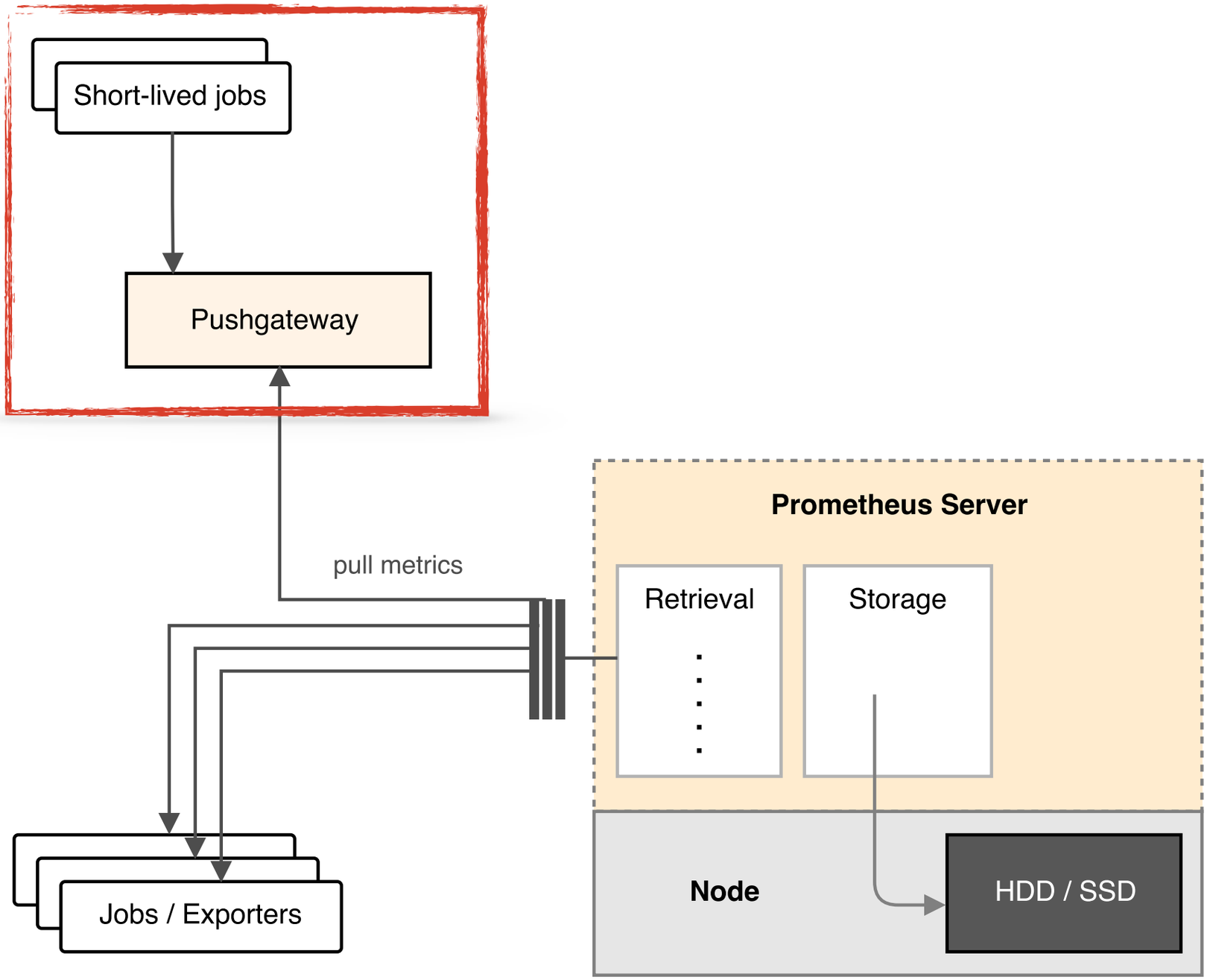

Prometheus Server
pull metrics
Retrieval
Storage
Jobs / Exporters
Node
HDD / SSD

1. resolution = scraping interval

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

Pull: Problems

- short lived jobs


Short-lived jobs
Pushgateway
pull metrics
Prometheus Server
Retrieval
Storage
Node
HDD / SSD
Jobs / Exporters

Pull: Problems

- short lived jobs
- target discovery

Configuration

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

Configuration

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

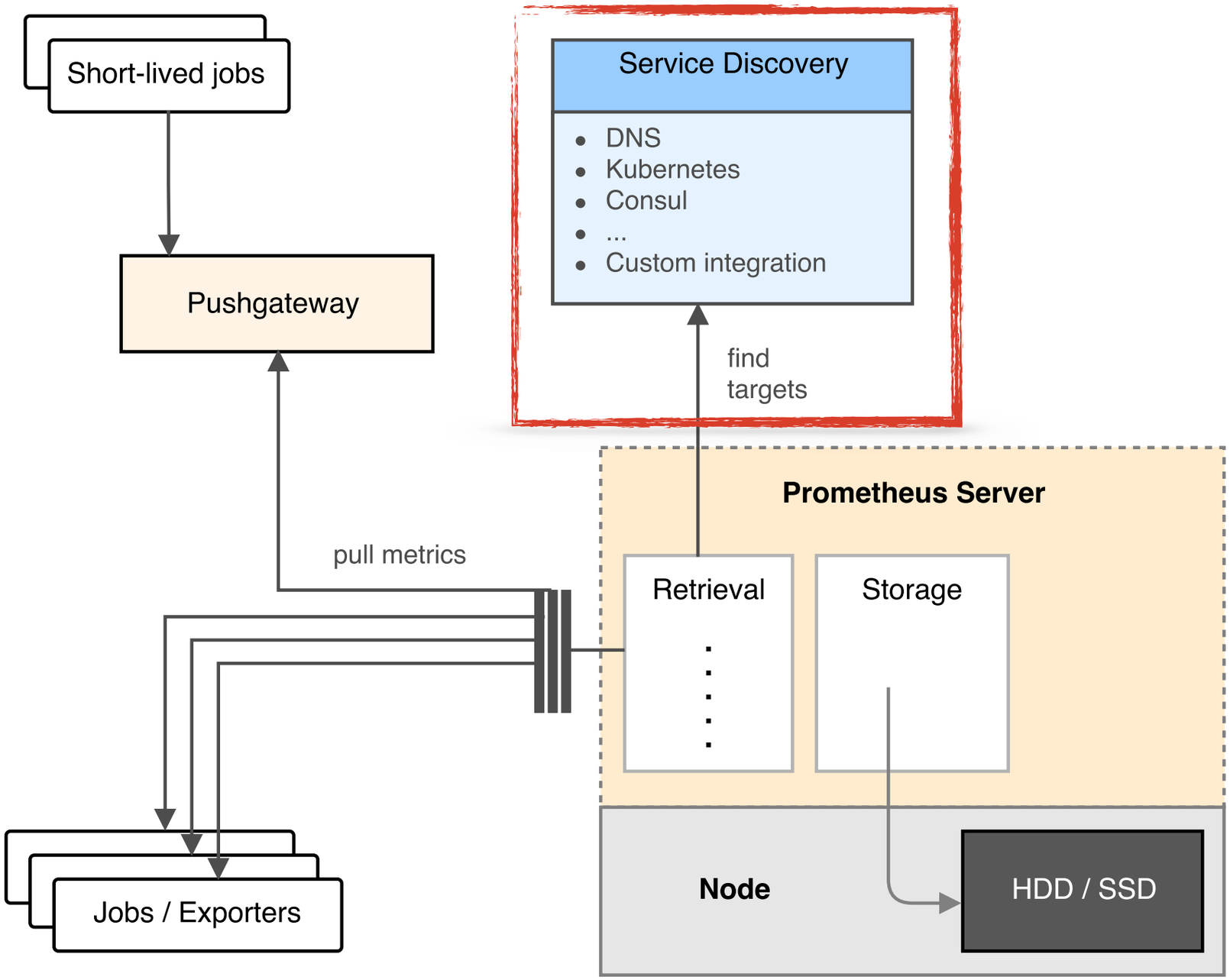
Configuration

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

Configuration

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

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


Short-lived jobs
Pushgateway
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
Prometheus Server
pull metrics
Retrieval
Storage
Node
HDD / SSD
Jobs / Exporters

Pull: Problems

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

Pull: Advantages

Pull: Advantages

- multiple Prometheis easy

Pull: Advantages

- multiple Prometheis easy
- outage detection

Pull: Advantages

- multiple Prometheis easy
- outage detection
- predictable, no self-DoS

Pull: Advantages

- multiple Prometheis easy
- outage detection
- predictable, no self-DoS
- easy to instrument 3rd 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

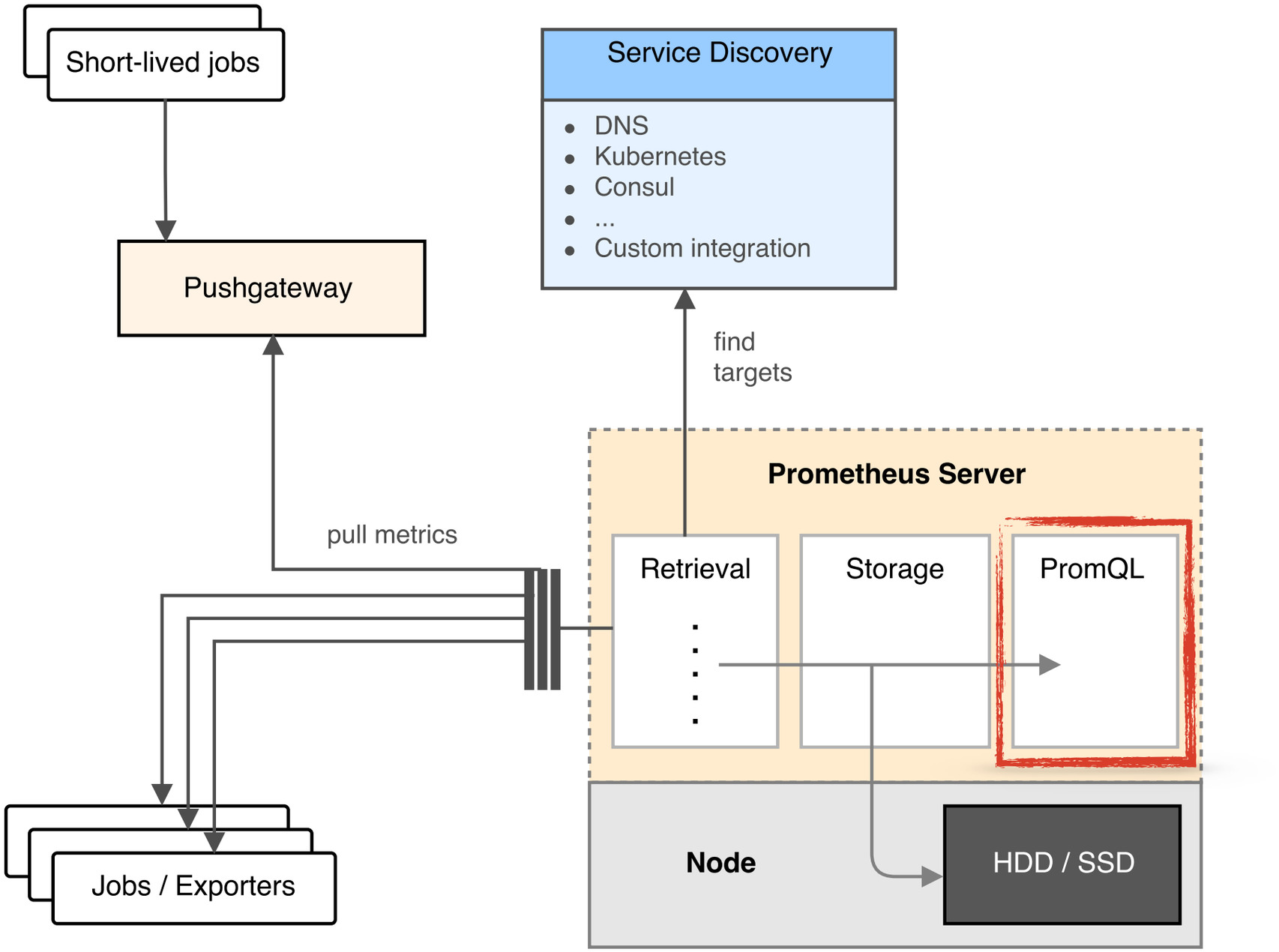
```
req_seconds_bucket{le="0.05"} 0.0
req_seconds_bucket{le="0.25"} 1.0
req_seconds_bucket{le="0.5"} 273.0
req_seconds_bucket{le="0.75"} 369.0
req_seconds_bucket{le="1.0"} 388.0
req_seconds_bucket{le="2.0"} 390.0
req_seconds_bucket{le="+Inf"} 390.0
```

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
```


Short-lived jobs
Pushgateway
pull metrics
Jobs / Exporters
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
Prometheus Server
Retrieval
Storage
PromQL
Node
HDD / SSD

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

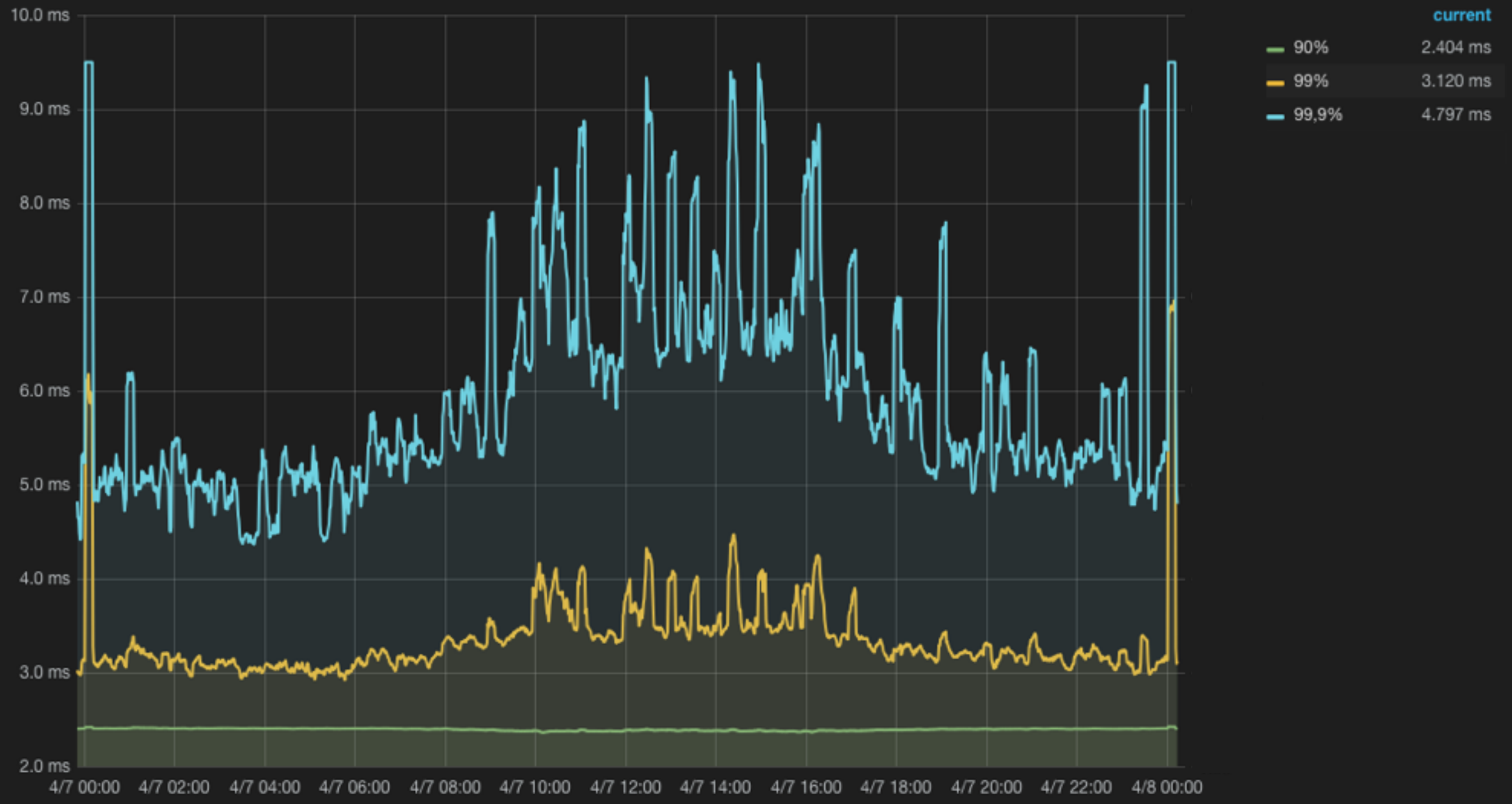
```
histogram_quantile(
  0.9, rate(
    req_seconds_bucket[10m]
))
```

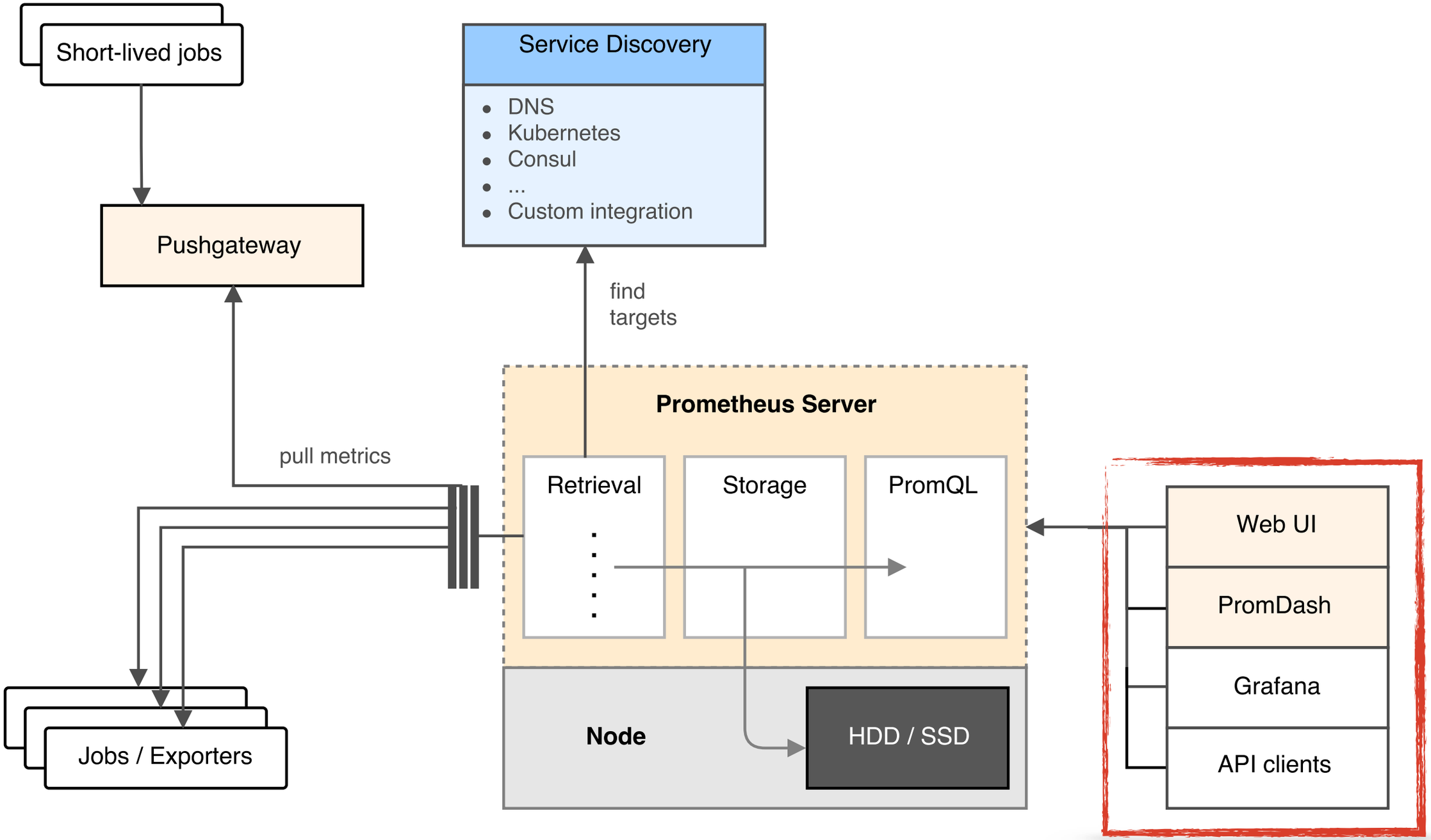
Percentiles

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

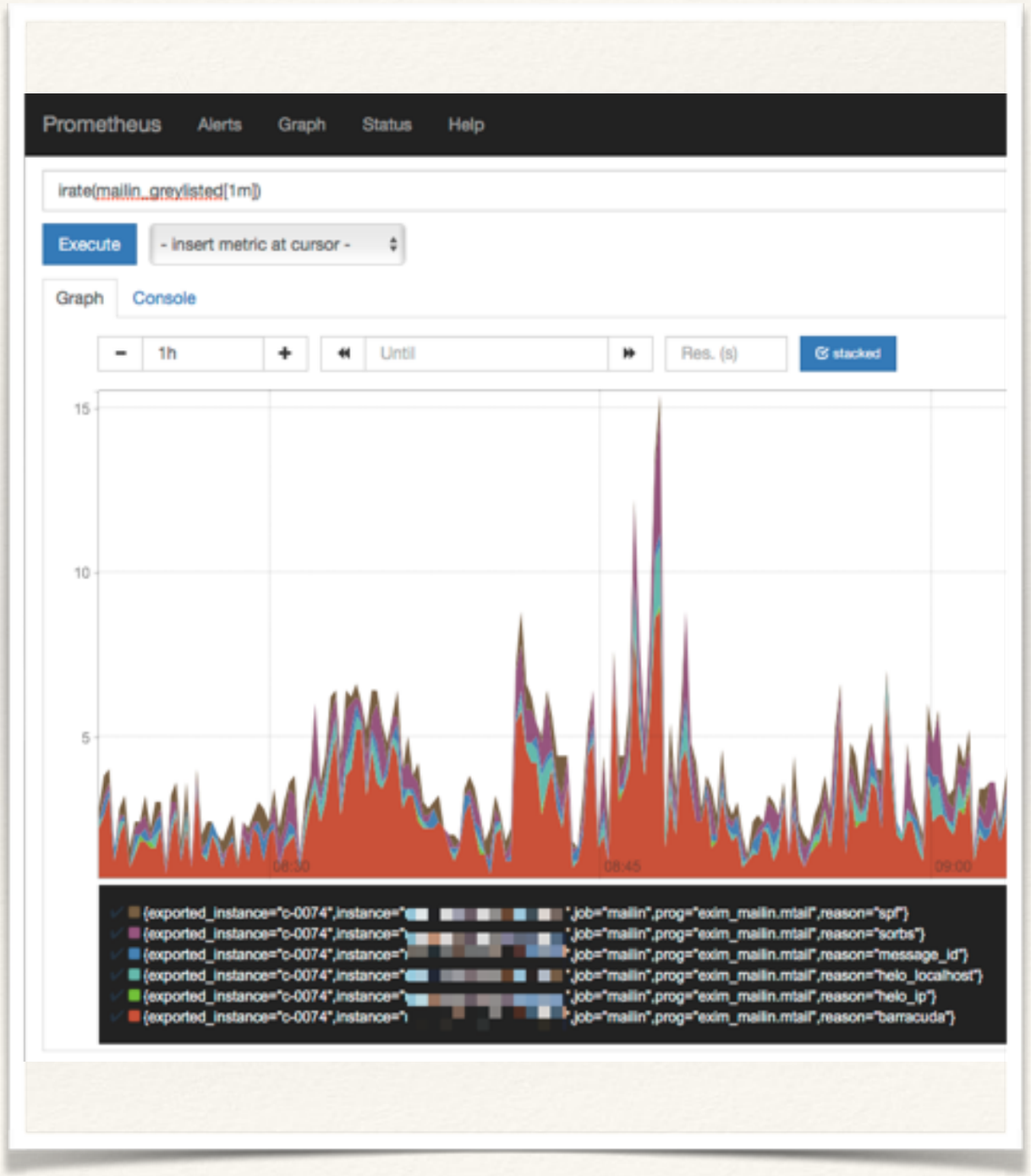
Percentiles

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


10.0 ms
9.0 ms
8.0 ms
7.0 ms
6.0 ms
5.0 ms
4.0 ms
3.0 ms
2.0 ms
4/7 00:00
4/7 02:00
4/7 04:00
4/7 06:00
4/7 08:00
4/7 10:00
4/7 12:00
4/7 14:00
4/7 16:00
4/7 18:00
4/7 20:00
4/7 22:00
4/8 00:00
current
90% 2.404 ms
99% 3.120 ms
99,9% 4.797 ms

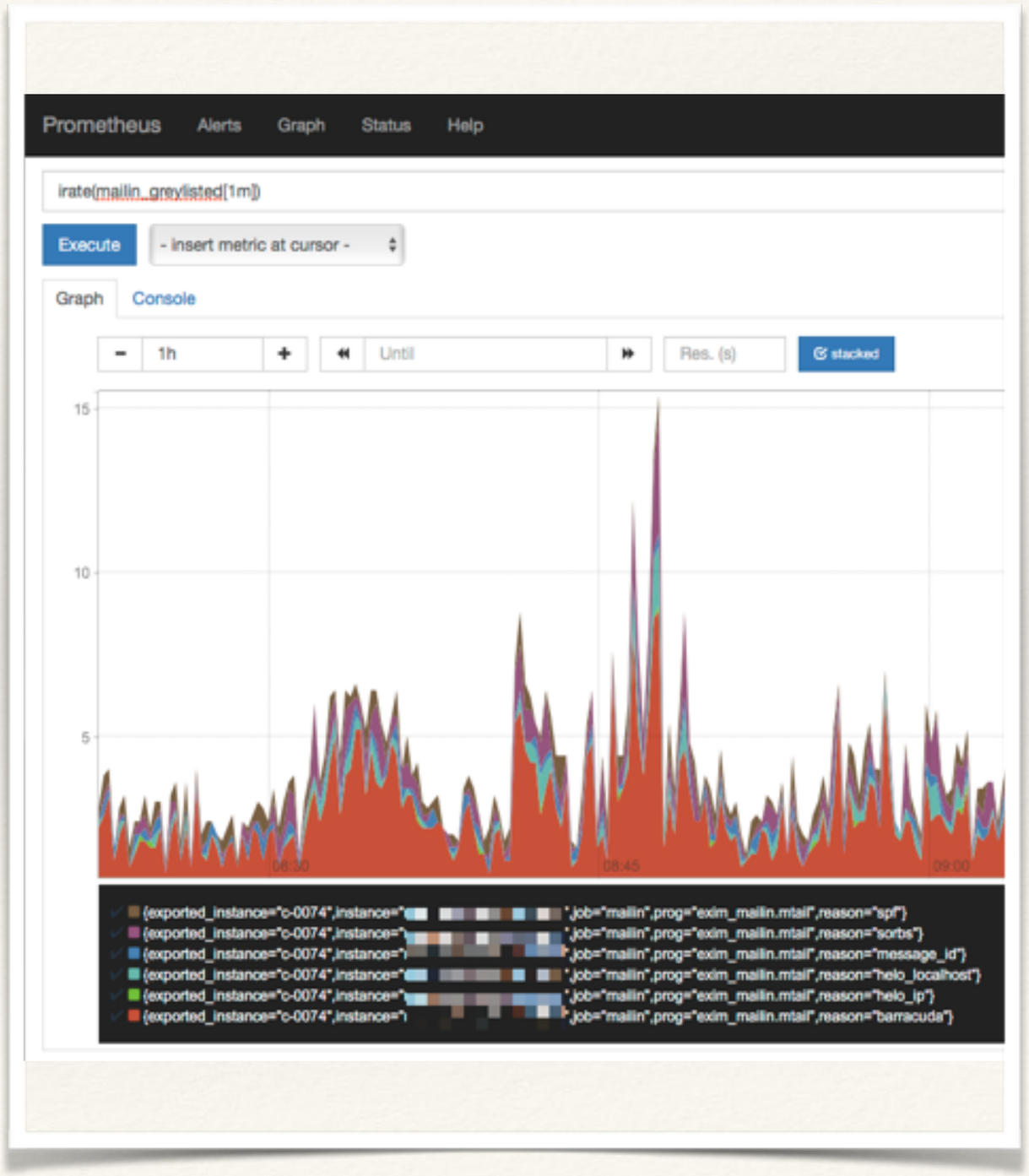

Short-lived jobs
Pushgateway
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
pull metrics
Prometheus Server
Retrieval
Storage
PromQL
Node
HDD / SSD
Jobs / Exporters
Web UI
PromDash
Grafana
API clients

Internal


Prometheus
Alerts
Graph
Status
Help
irate(mailin_greylisted[1m])
Execute
- insert metric at cursor -
Graph
Console
1h
Until
Res. (s)
stacked
15
10
5
08:30
08:45
09:00
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="spf"}
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="sorbs"}
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="message_id"}
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="helo_localhost"}
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="helo_ip"}
{exported_instance="c-0074",instance="
",job="mailin",prog="exim_mailin.mtail",reason="barracuda"}

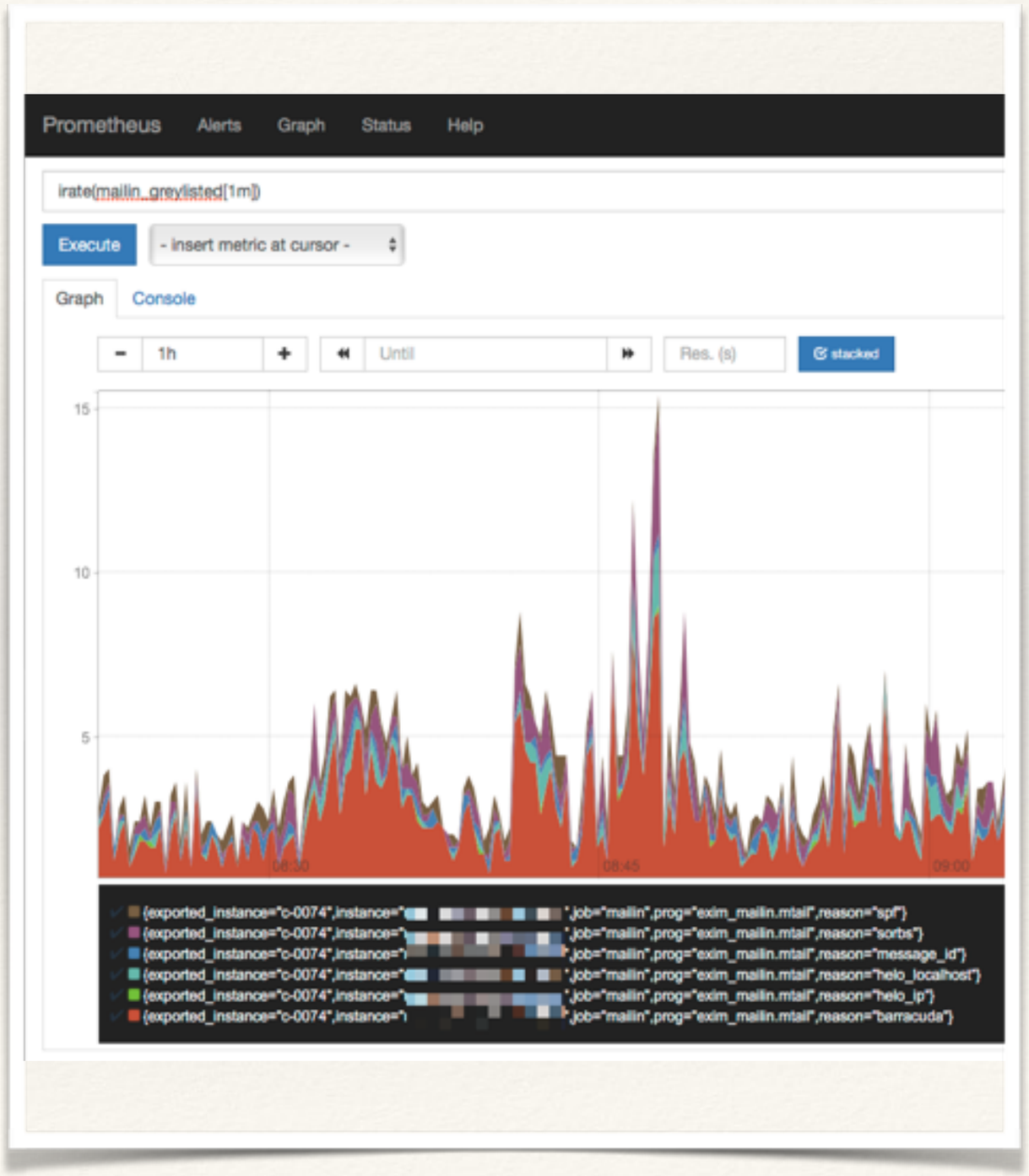
Internal

- great for ad-hoc



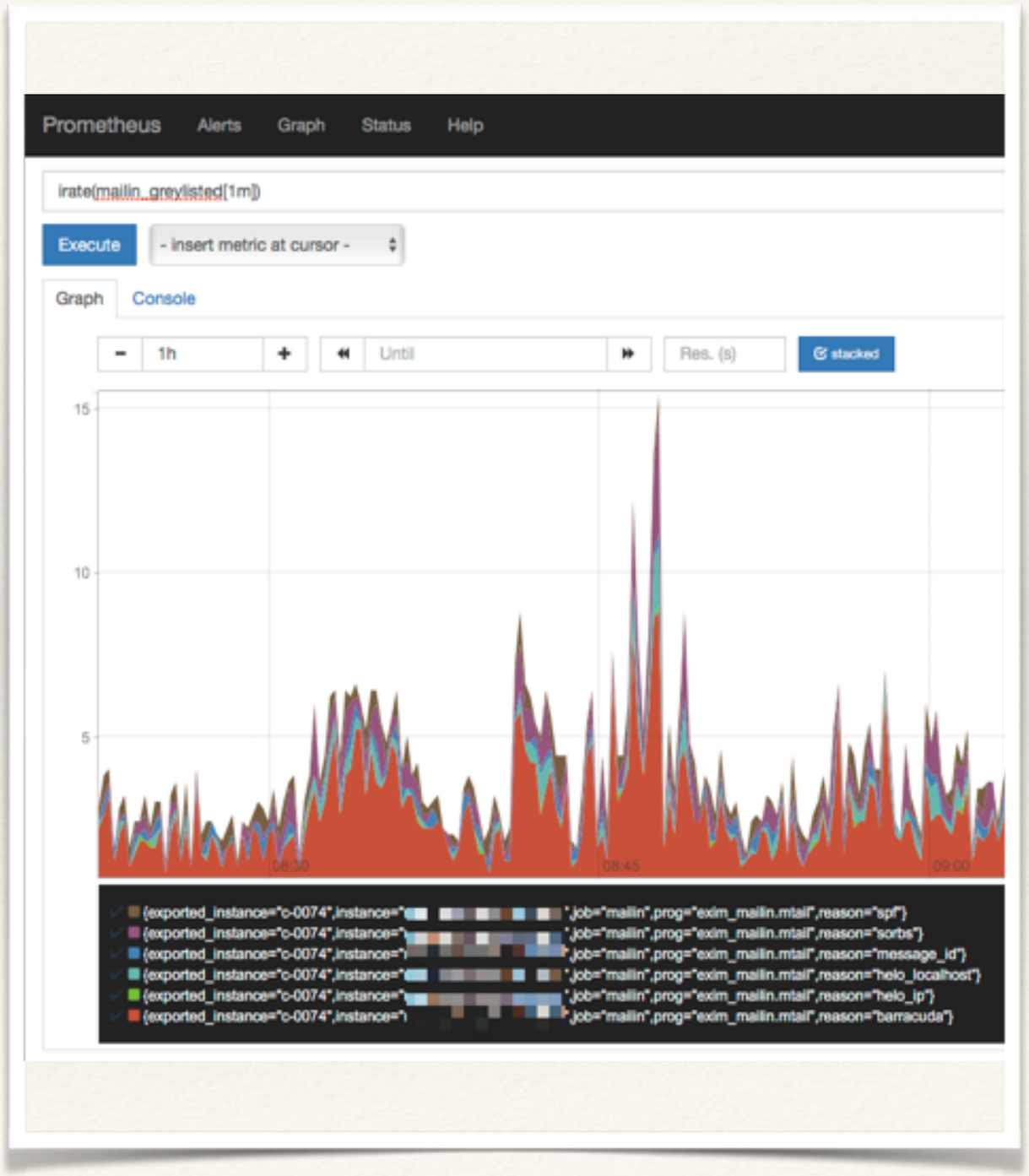
Internal

- great for ad-hoc
- 1 expr per graph



Internal

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



PromDash

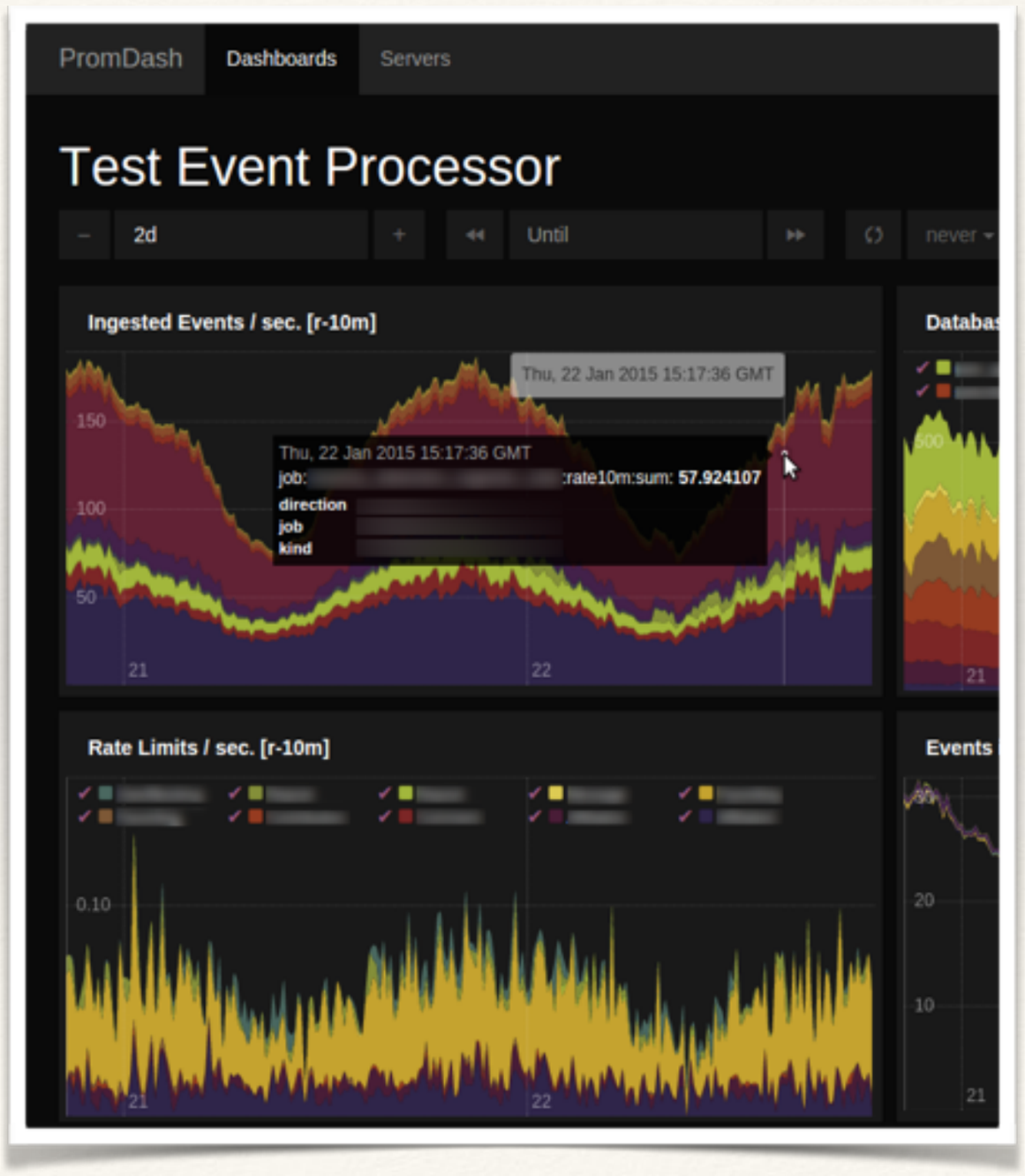

PromDash
Dashboards
Servers
Test Event Processor
2d
Until
never
Ingested Events / sec. [r-10m]
Thu, 22 Jan 2015 15:17:36 GMT
150
100
50
21
22
Thu, 22 Jan 2015 15:17:36 GMT
job:
:rate10m:sum: 57.924107
direction
job
kind
Rate Limits / sec. [r-10m]
0.10
Events
20
10

PromDash

- best integration

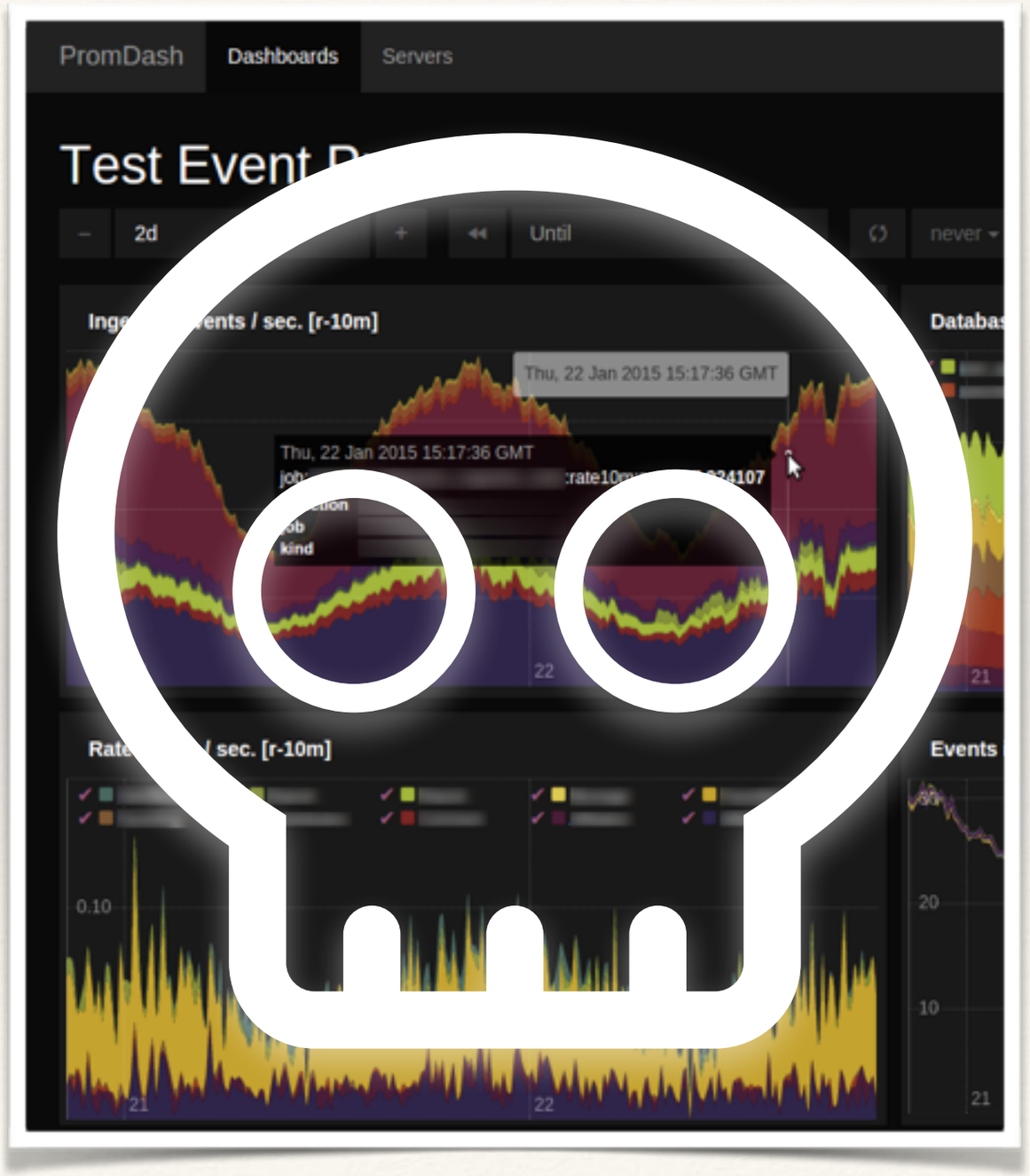
PromDash

- best integration
- former official



PromDash

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



Grafana


Webauth
Requests/second
45 Hz
Requests / s
Request Times vs Load

Grafana

- pretty & powerful

Grafana

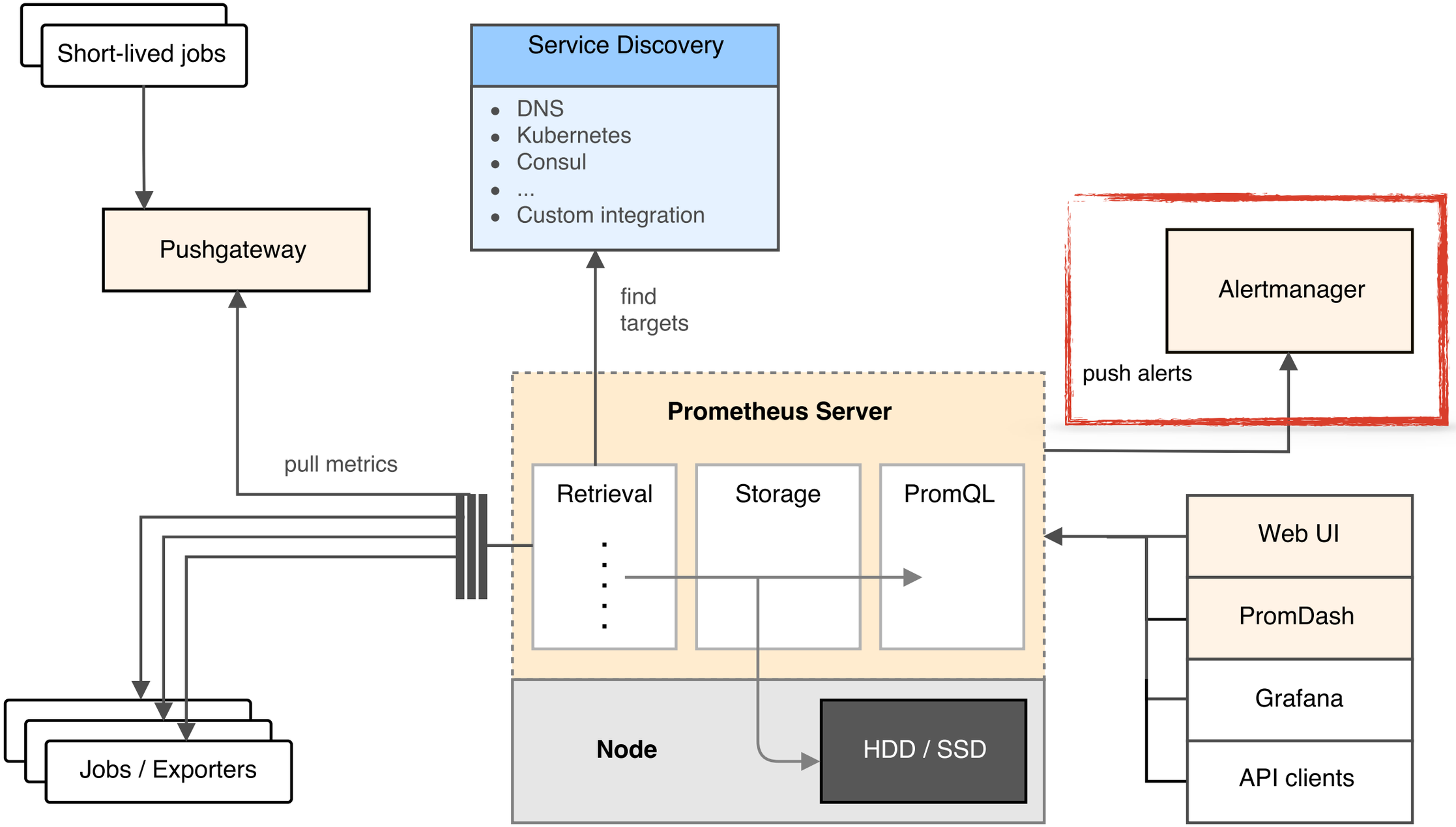
- pretty & powerful
- *many* integrations

Grafana

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

Grafana

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


Short-lived jobs
Pushgateway
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
Alertmanager
push alerts
Prometheus Server
pull metrics
Retrieval
Storage
PromQL
Web UI
PromDash
Grafana
API clients
Jobs / Exporters
Node
HDD / SSD

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

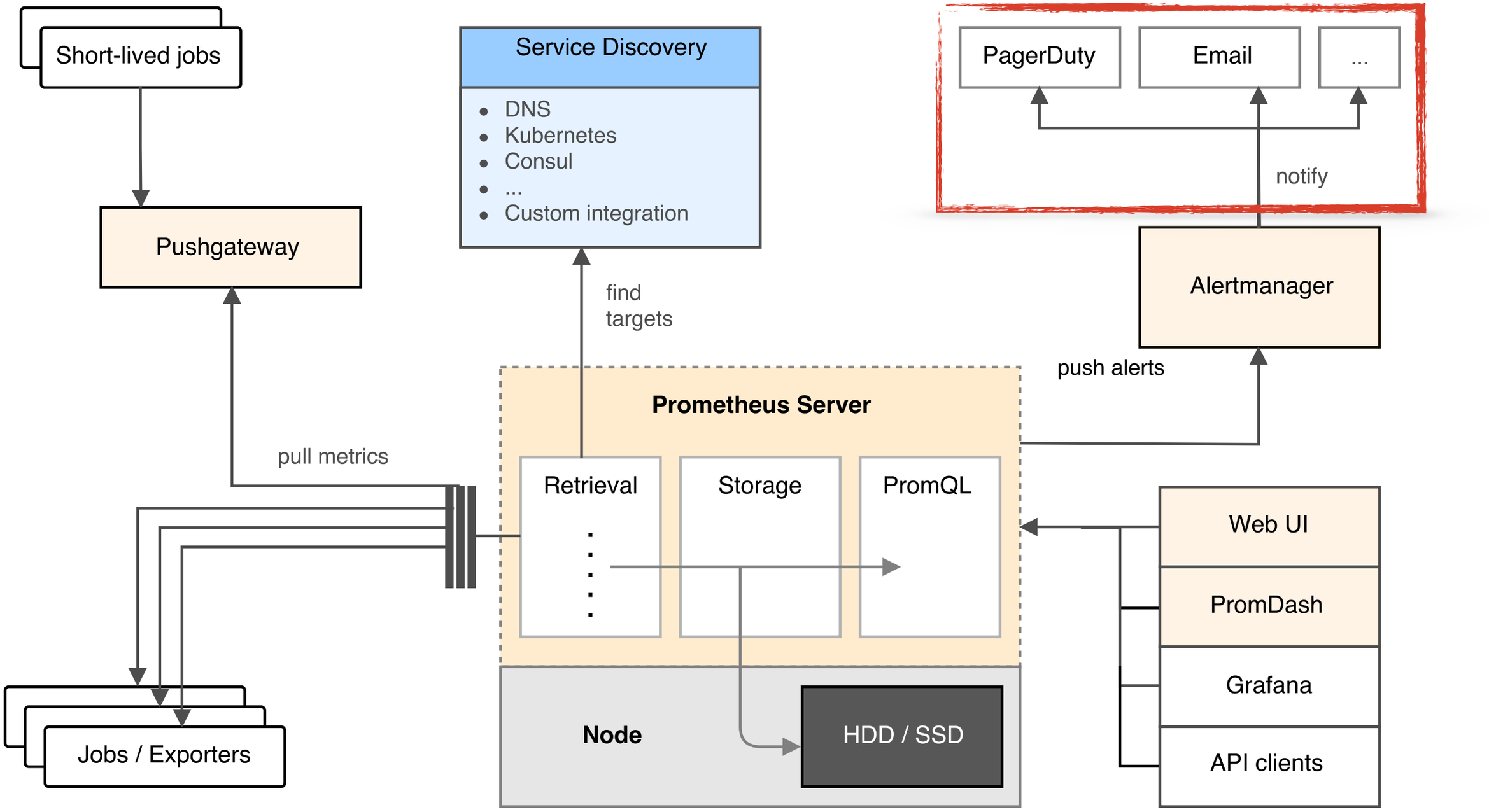
```
ALERT DiskWillFillIn4Hours
IF predict_linear(
  node_filesystem_free[1h], 4*3600) < 0
FOR 5m
```

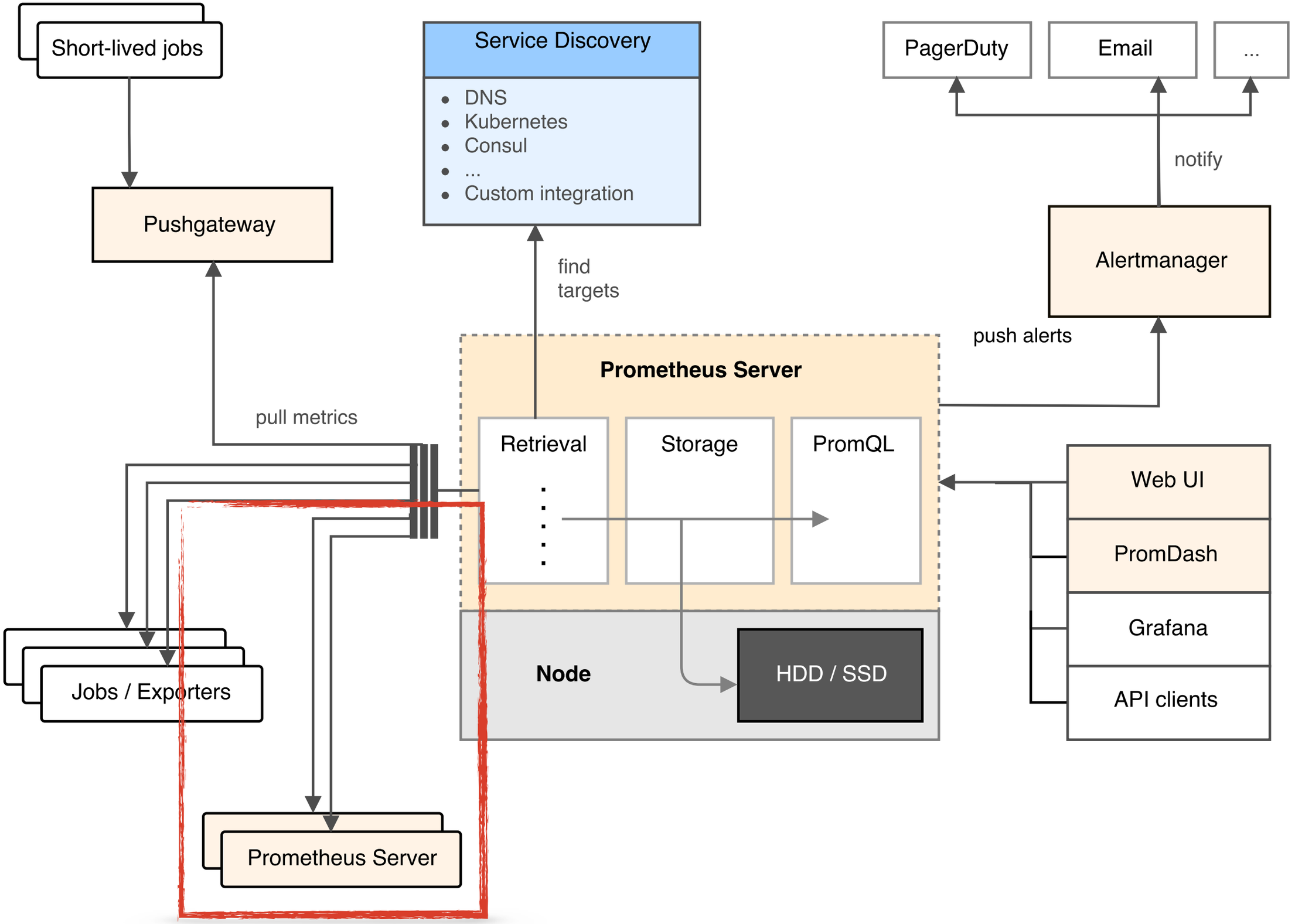
Alerts & Scrying

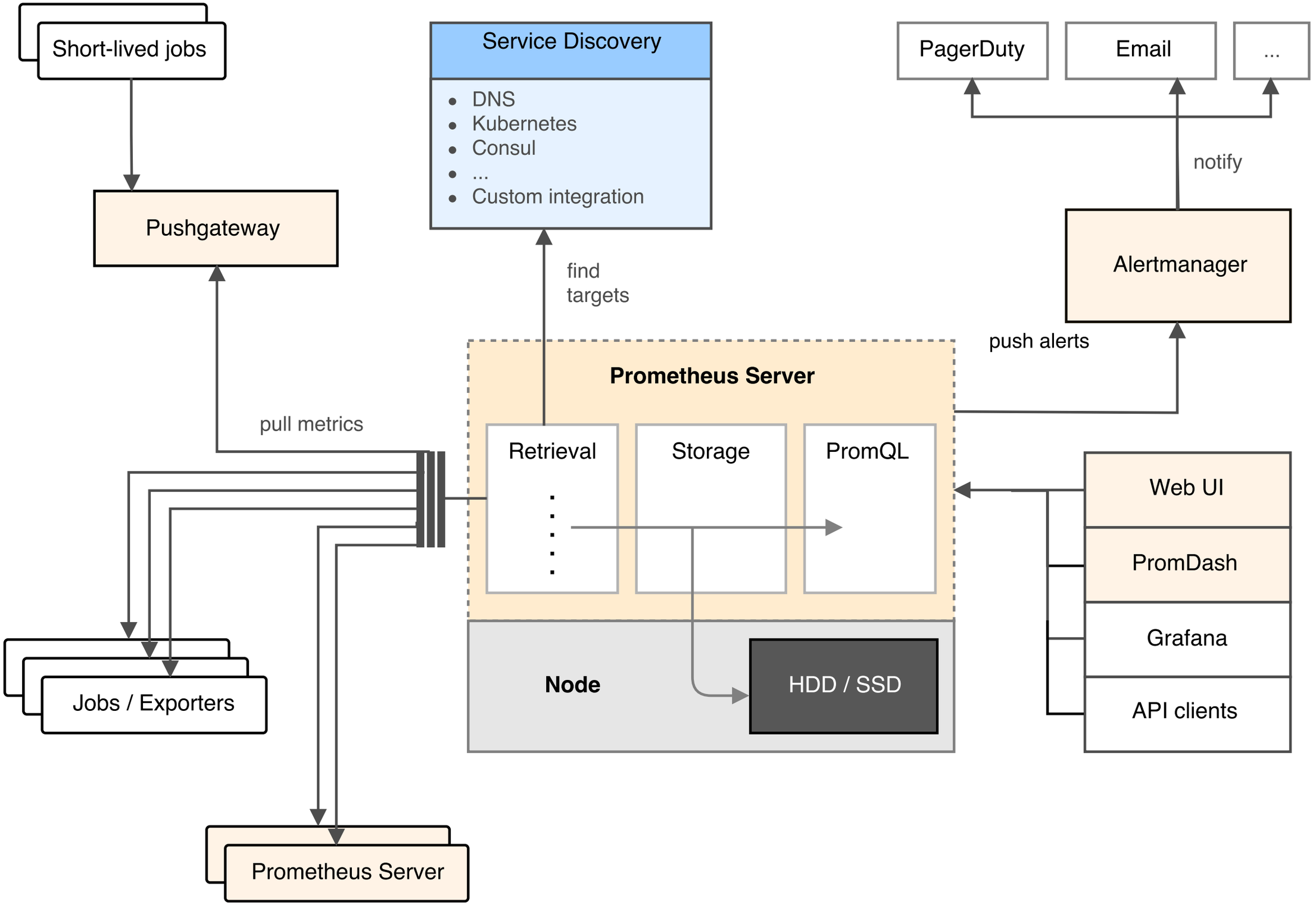
```
ALERT DiskWillFillIn4Hours
IF predict_linear(
  node_filesystem_free[1h], 4*3600) < 0
FOR 5m
```

Alerts & Scrying

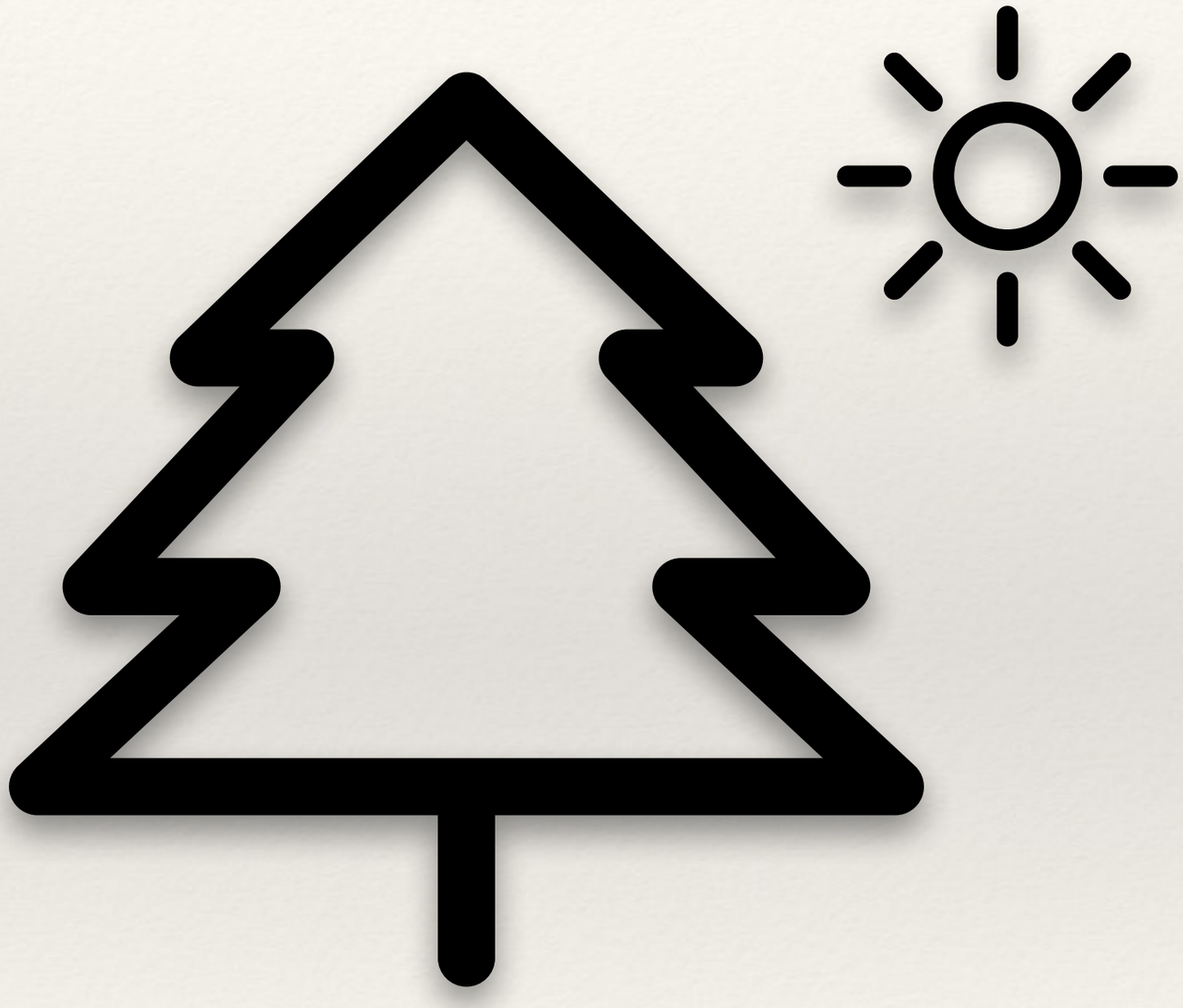
```
ALERT DiskWillFillIn4Hours
IF predict_linear(
   node_filesystem_free[1h], 4*3600) < 0
FOR 5m
```


Short-lived jobs
Pushgateway
pull metrics
Jobs / Exporters
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
Prometheus Server
Retrieval
Storage
PromQL
Node
HDD / SSD
PagerDuty
Email
...
notify
Alertmanager
push alerts
Web UI
PromDash
Grafana
API clients


Short-lived jobs
Pushgateway
pull metrics
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
PagerDuty
Email
...
notify
Alertmanager
push alerts
Prometheus Server
Retrieval
Storage
PromQL
Node
HDD / SSD
Web UI
PromDash
Grafana
API clients
Jobs / Exporters
Prometheus Server


Short-lived jobs
Pushgateway
pull metrics
Service Discovery
DNS
Kubernetes
Consul
...
Custom integration
find
targets
Prometheus Server
Retrieval
Storage
PromQL
Node
HDD / SSD
PagerDuty
Email
...
notify
Alertmanager
push alerts
Web UI
PromDash
Grafana
API clients
Jobs / Exporters
Prometheus Server

Environment



HAProxy

MySQL

etcd

Consul

nginx

statsd

graphite

collectd

Django

redis

Kubernetes

PostgreSQL

SNMP

CouchDB

Varnish

InfluxDB

Apache

MongoDB

HAProxy

MySQL

etcd

Consul

statsd

nginx

graphite

collectd

Django

redis

Kubernetes

PostgreSQL

SNMP

CouchDB

InfluxDB

Varnish

Apache

MongoDB

node_exporter

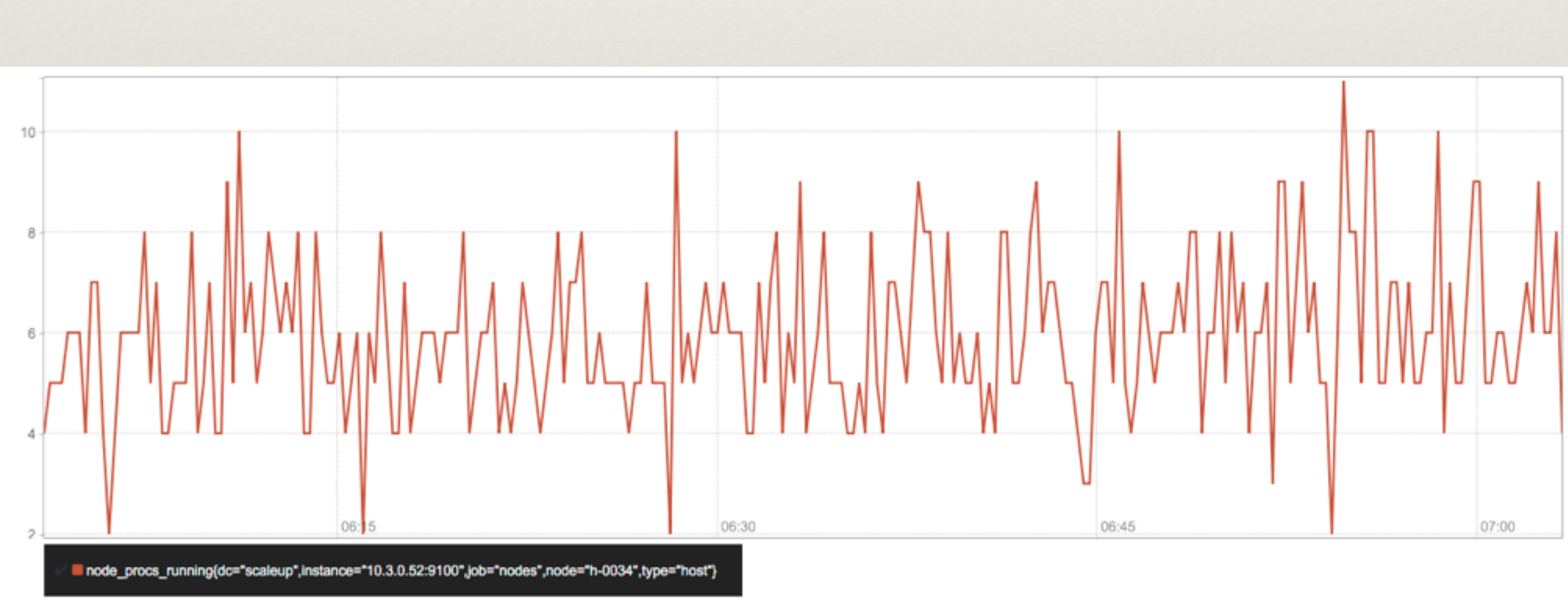
cAdvisor

node_exporter

System Insight

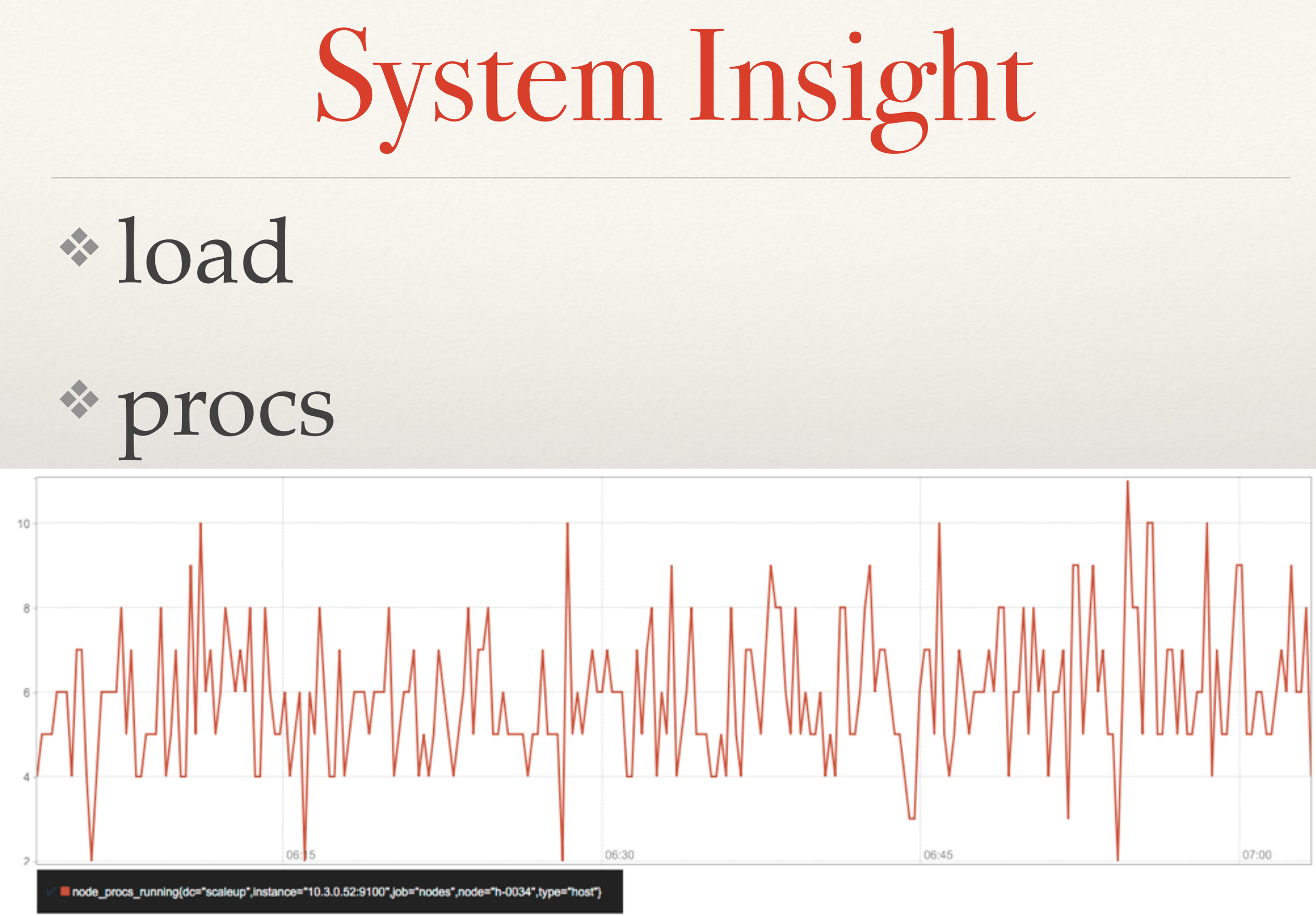
System Insight

- load



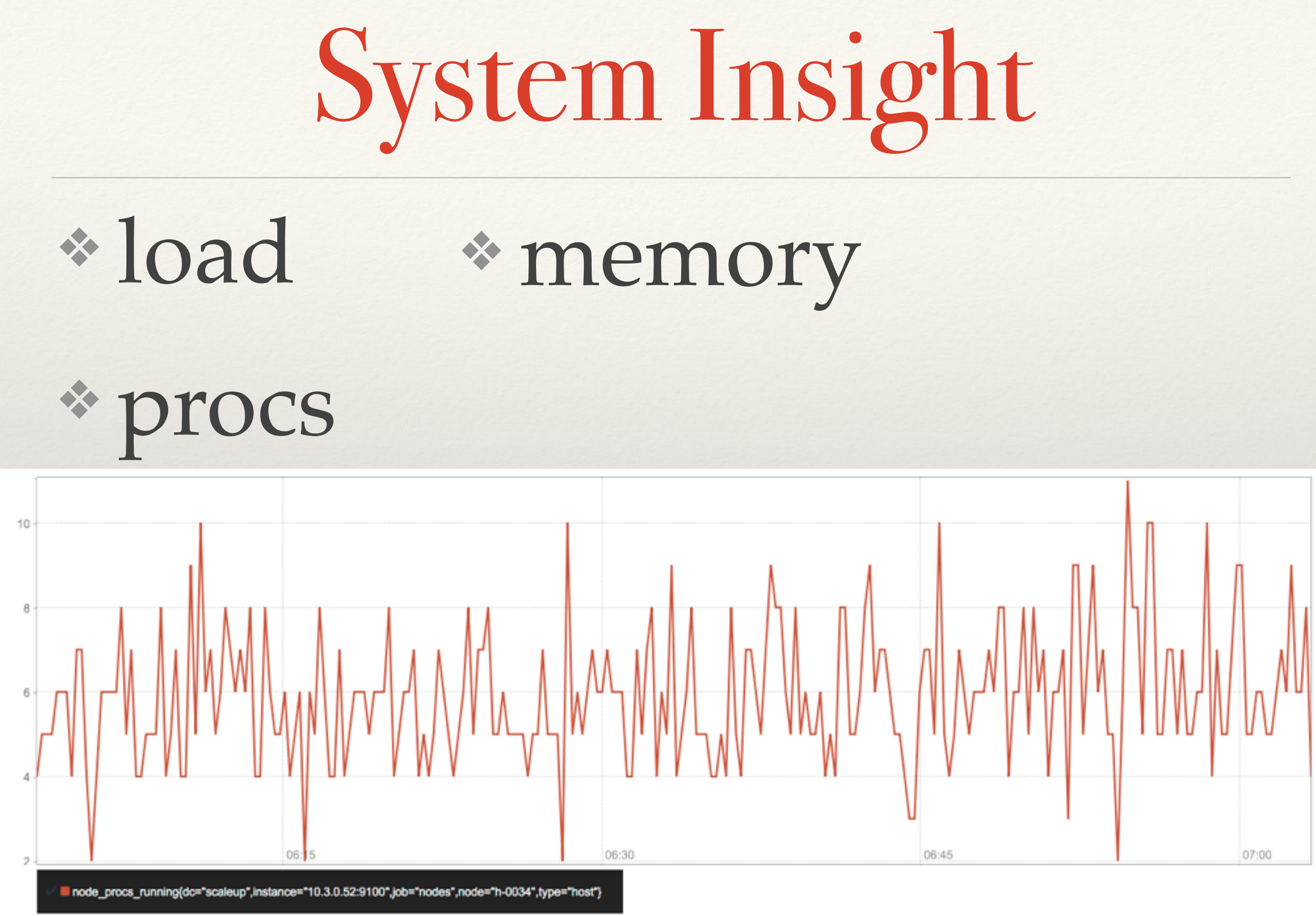
System Insight

- load
- procs



System Insight

- load
- memory
- procs



System Insight

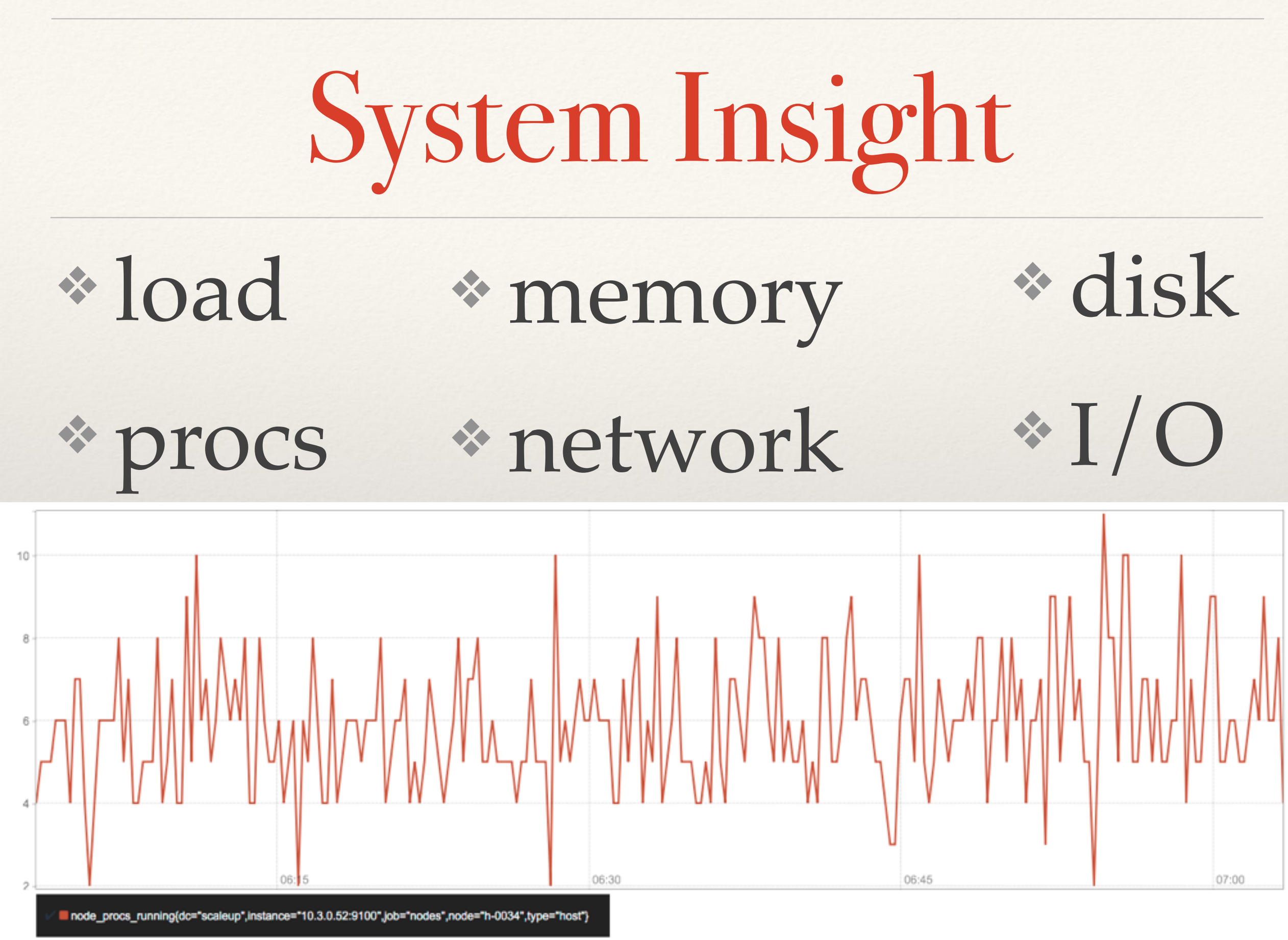
- load
- memory
- procs
- network

System Insight

- load
- memory
- disk
- procs
- network

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

```python
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()
```

```python
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()
```

```python
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 0.16999999998
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 0.16999999998
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 0.16999999998
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 0.16999999998
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 0.16999999998
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 0.16999999998
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.")
```

```python
@IN_PROGRESS.track_inprogress()
@REQUEST_TIME.time()
@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!"
```

```python
@IN_PROGRESS.track_inprogress()
@REQUEST_TIME.time()
@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!"
```

```
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()
```

```python
@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

vrmd.de