Prometheus for Practioners

Migrating to Prometheus at Fastly

Monitorama EU 2018 | Marcus Barczak
@ickymettle

fastly
R

lonely planet

Pinterest

iHeartMEDIA

yelp

wayfair

DAILY NEWS

the guardian

imgur

Rakuten.com

Boots

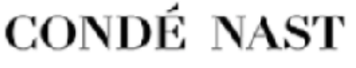
A&E

GitHub

shazam

FAST COMPANY

BBC


CONDÉ NAST

BuzzFeed

RollingStone

BUSINESS INSIDER

KICKSTARTER

Hotel Tonight

FOURSQUARE

W
WENNER MEDIA

New Relic.

imgIX

GOV.UK

vimeo

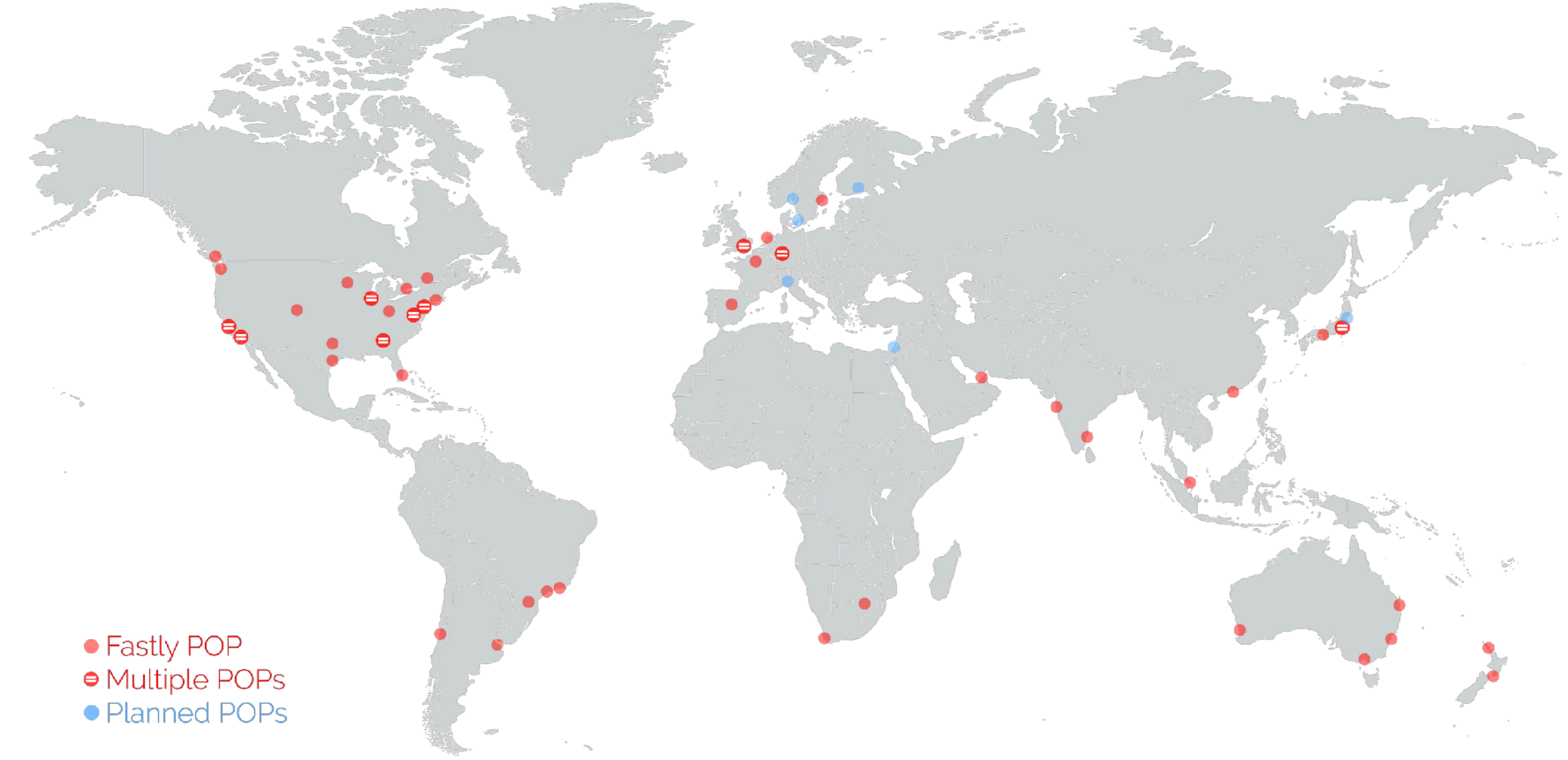
shopify

1stdibs.

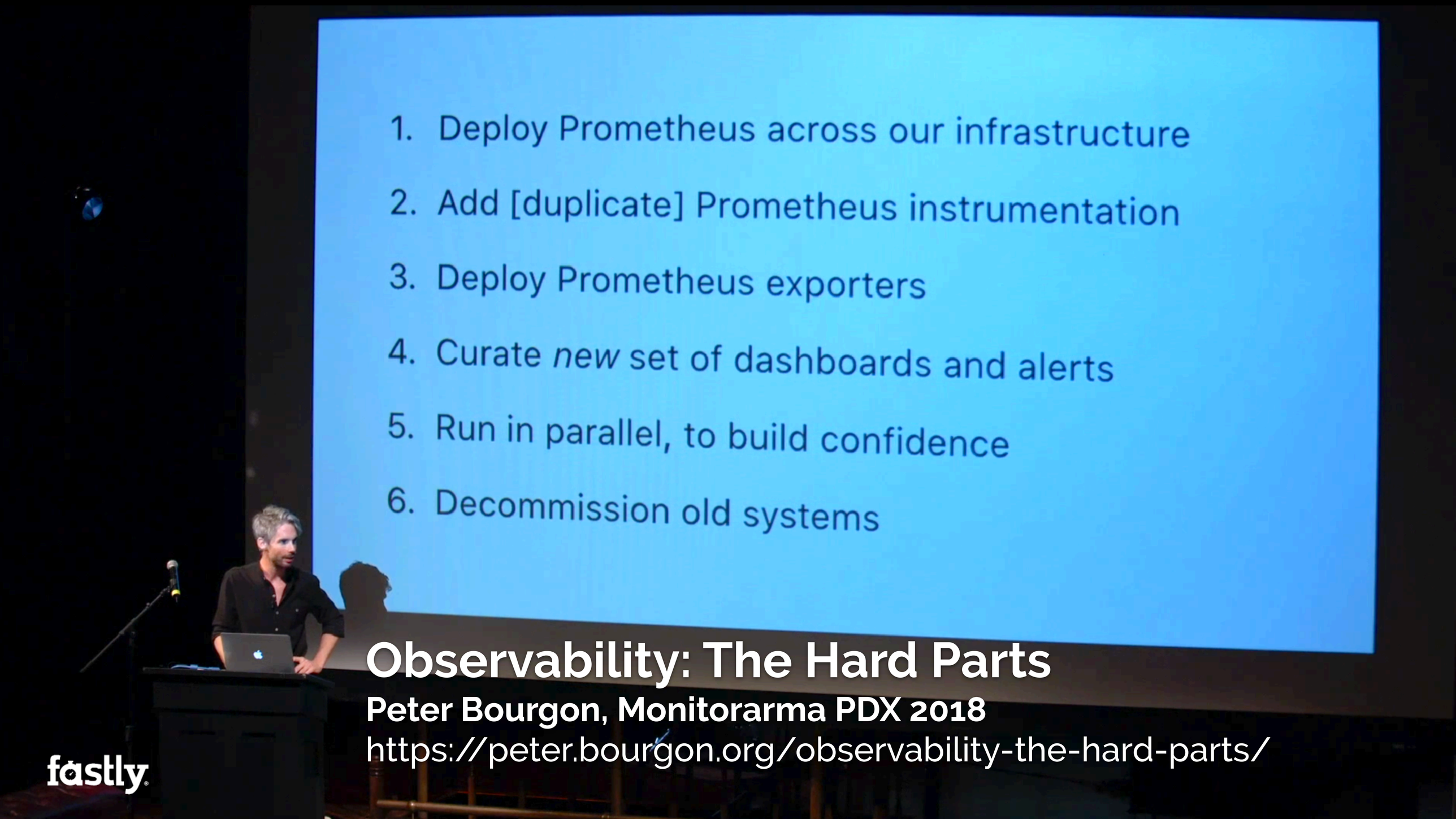
KAYAK

Etsy

fastly



fastly


1. Deploy Prometheus across our infrastructure
2. Add [duplicate] Prometheus instrumentation
3. Deploy Prometheus exporters
4. Curate *new* set of dashboards and alerts
5. Run in parallel, to build confidence
6. Decommission old systems
Observability: The Hard Parts
Peter Bourgon, Monitorarma PDX 2018
https://peter.bourgon.org/observability-the-hard-parts/
fastly

Prometheus for Practioners

~~Migrating to Prometheus at Fastly~~

Observability: The "Easy" Parts

Monitorama EU 2018 | Marcus Barczak

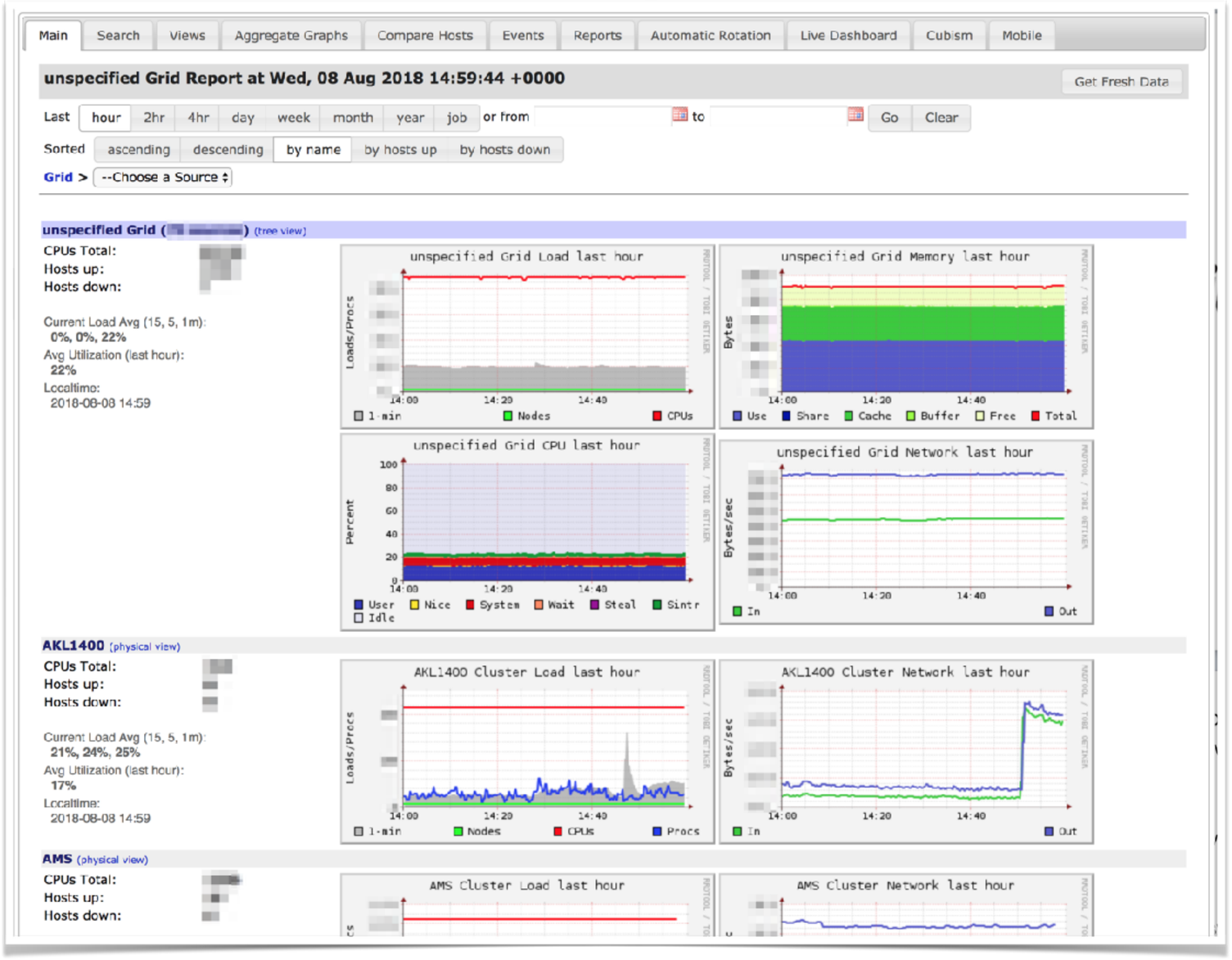
@ickymettle

How **were** we monitoring Fastly?

Ganglia

+

ICINGA



fastly

Growing pains with Ganglia

- Operational overhead.
- Limited graphing functions.
- No alerting support,
- No real API for consuming metric data.

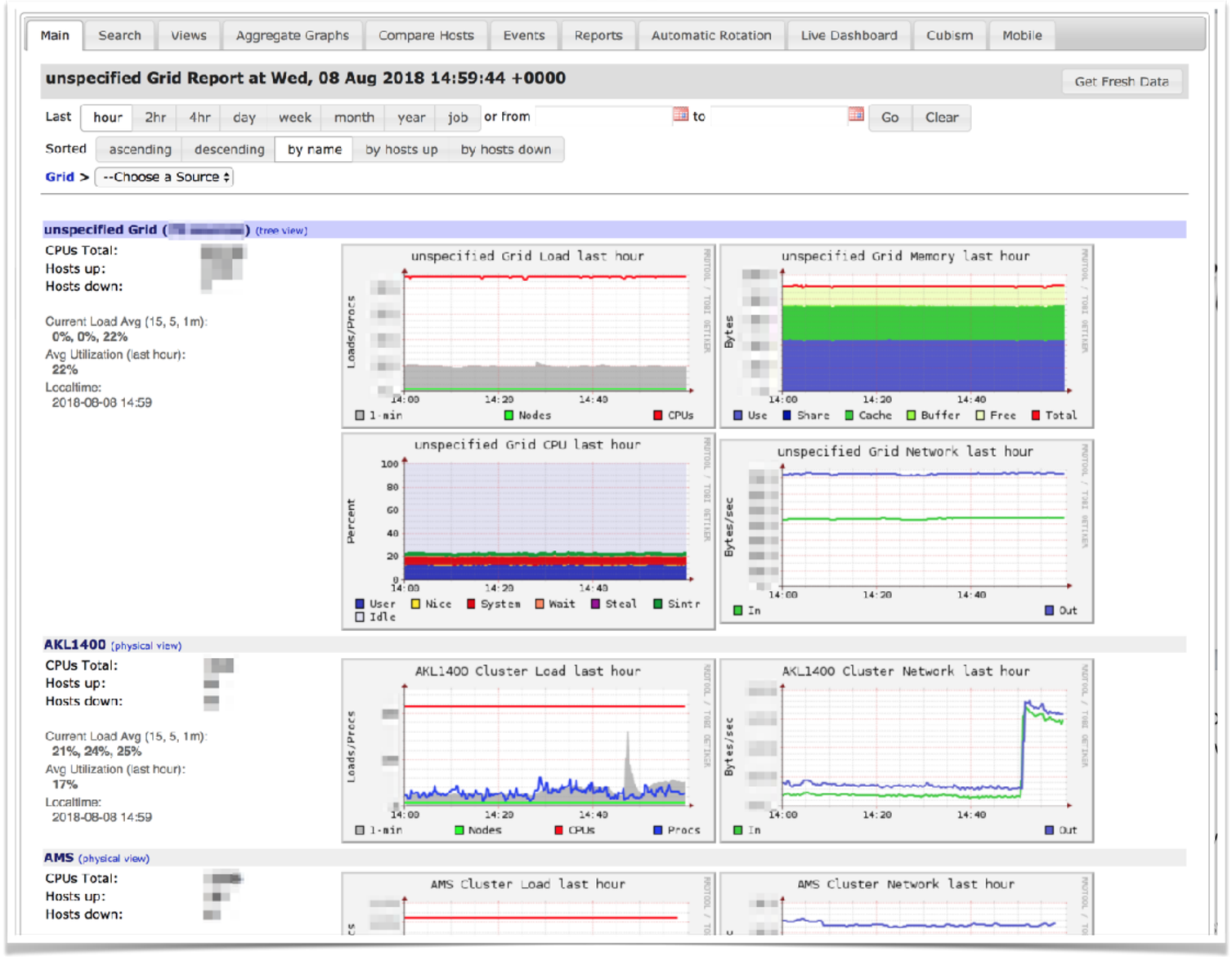
Ganglia

+

ICINGA

+

$aaS



fastly

Growing pains doubled

- Now supporting **two** systems.
- Where do I put my metrics?
- Still writing external plugins and agents.
- Monitoring treated as a "post-release" phase.

Scaling our infrastructure

horizontally

Required scaling our monitoring

vertically

Third time lucky

Project goals

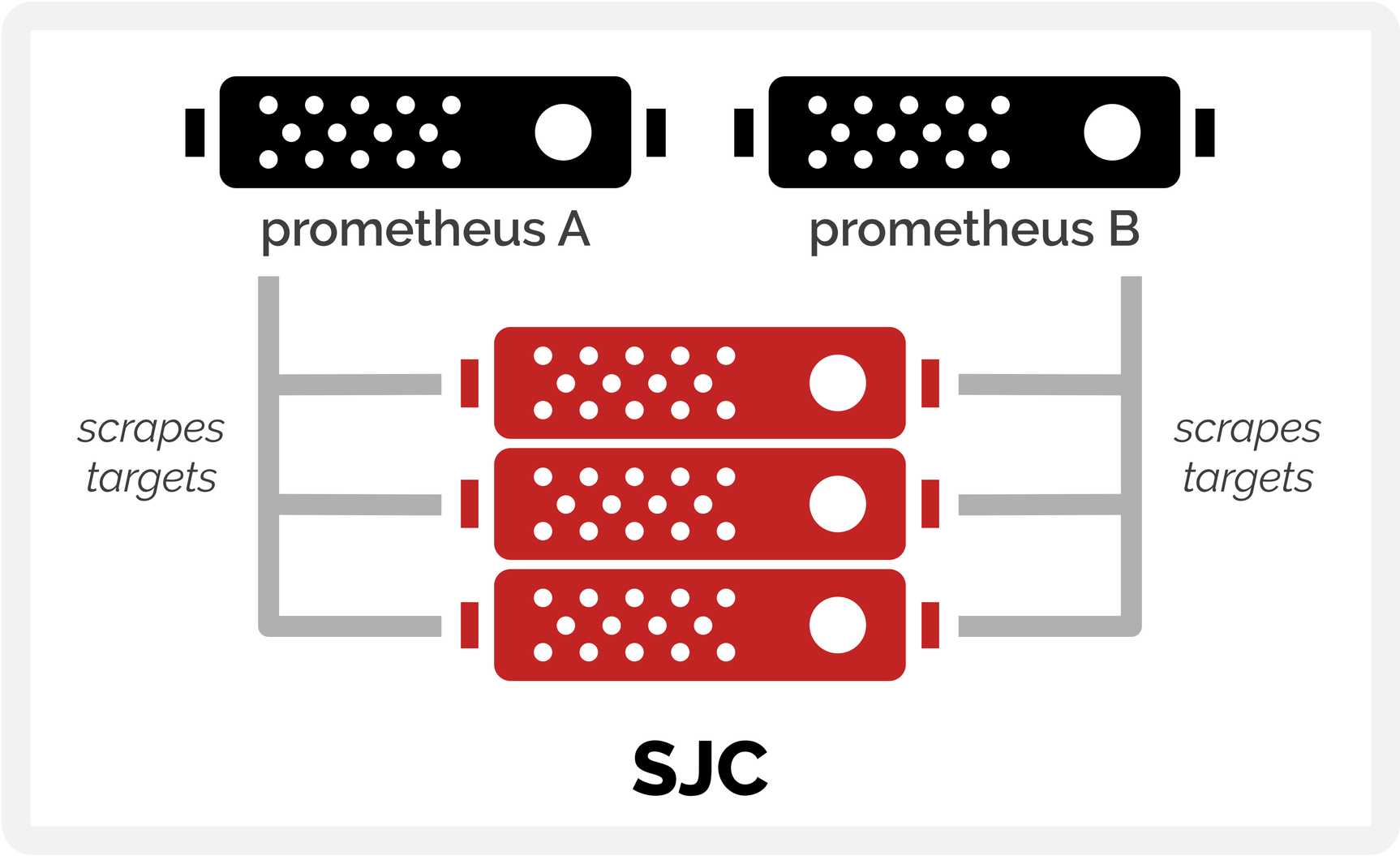
- Scale with our infrastructure growth,
- Be easy to deploy and operate.
- Engineer friendly instrumentation libraries.
- First class API support for data access.
- To reboot our monitoring culture.

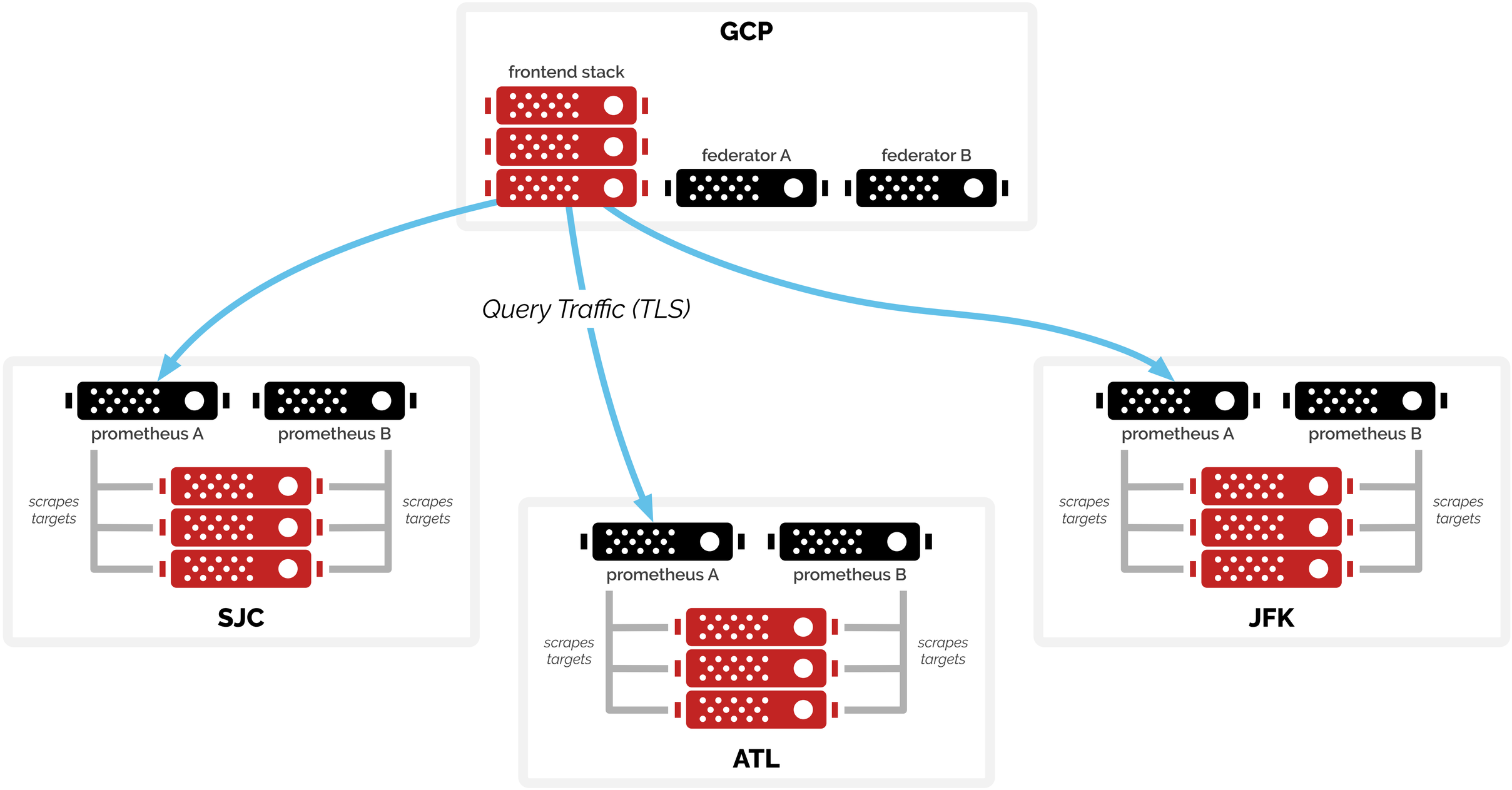
fastly

Getting started

- Build a proof of concept.
- Pair with pilot team to instrument their services.
- Iterate through the rest.
- Run both systems in parallel.
- Decommission SaaS system and Ganglia.

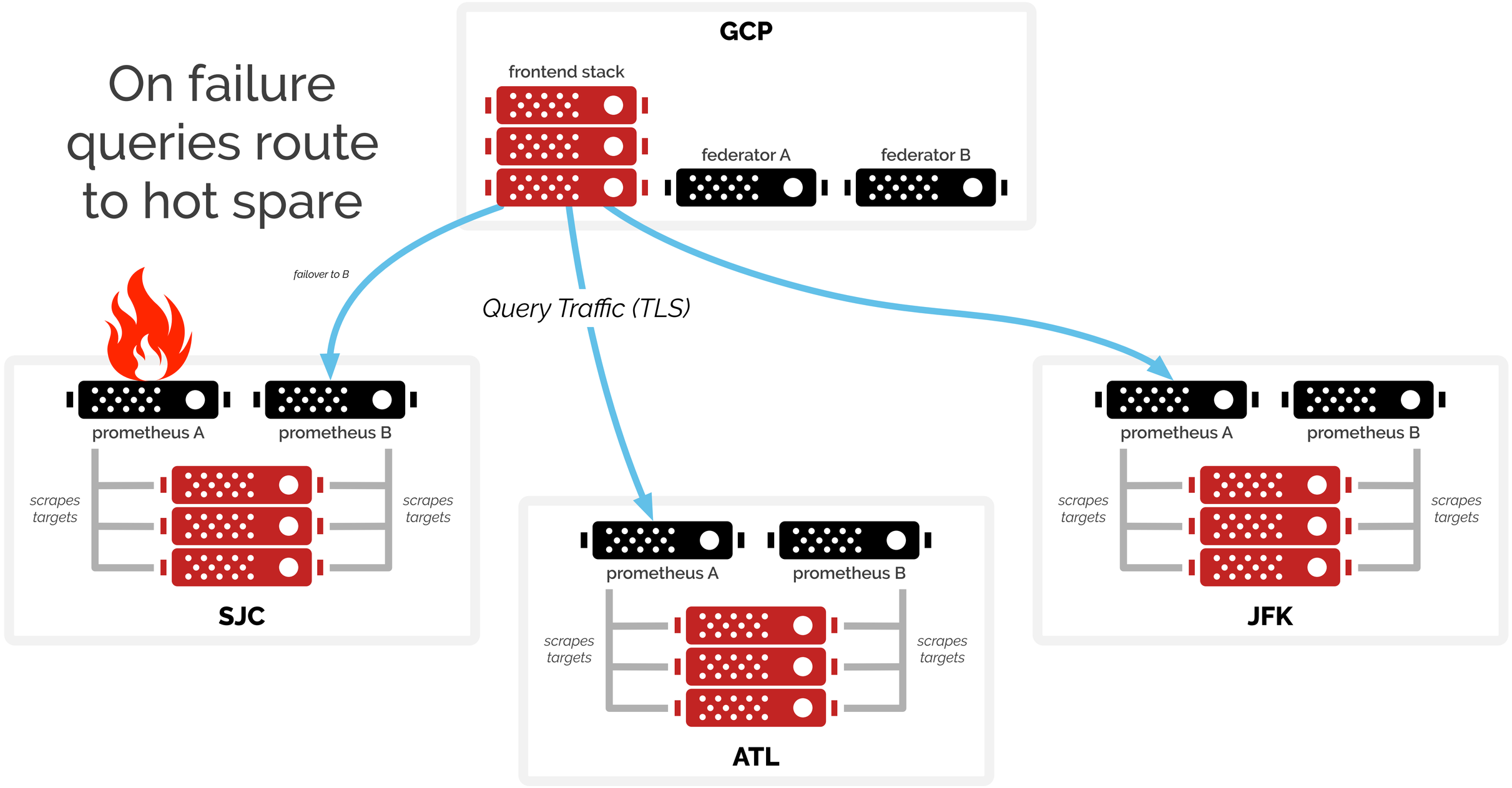
Infrastructure build


prometheus A
prometheus B
scrapes targets
scrapes targets
SJC


GCP
frontend stack
federator A
federator B
Query Traffic (TLS)
prometheus A
prometheus B
scrapes targets
scrapes targets
SJC
prometheus A
prometheus B
scrapes targets
scrapes targets
ATL
prometheus A
prometheus B
scrapes targets
scrapes targets
JFK

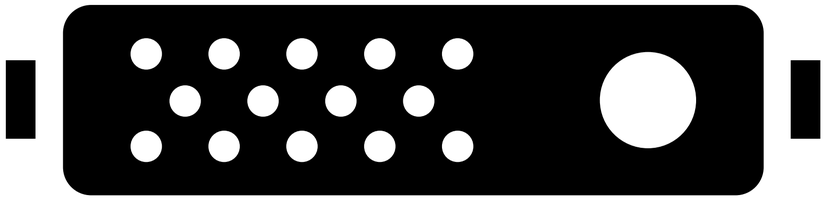
fastly

On failure queries route to hot spare



fastly

Prometheus Server Software Stack



Ghost Tunnel
TLS termination and auth.

Service Discovery Sidecar
Target configuration

Rules Loader
Recording and Alert rules

Prometheus

fastly

Prometheus Server Software Stack

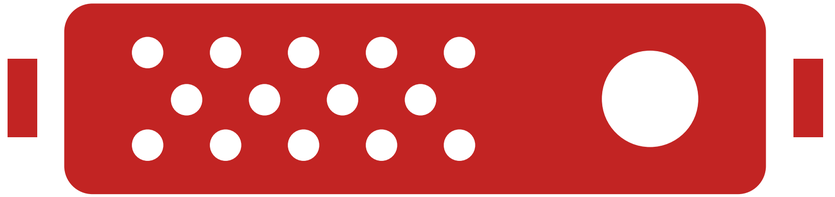
Ghost Tunnel
TLS termination and auth.

Service Discovery Sidecar
Target configuration

Rules Loader
Recording and Alert rules

Prometheus

Typical Server Software Stack



Exporters
Built into services or sidecar

Service Discovery Proxy
Service discovery and
TLS exporter proxy

fastly

Build your own service discovery?

Fastly's infrastructure
is bare metal hardware
no cloud conveniences

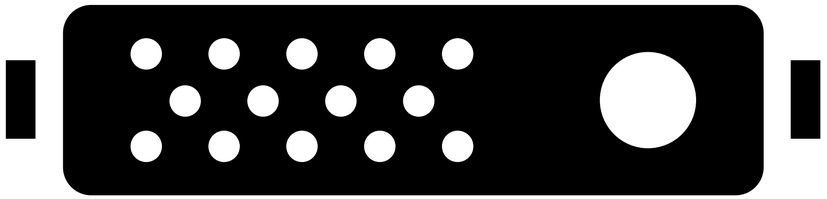
fastly

Service discovery requirements

- Automatic discovery of targets.
- Self-service registration of exporter endpoints,
- TLS encryption for all exporter traffic.
- Minimal exposure of exporter TCP ports.

fastly

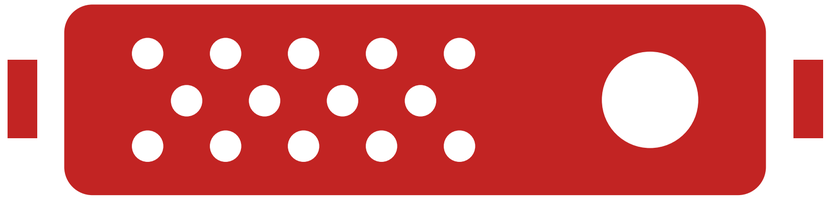
Prometheus Server Software Stack



Ghost Tunnel

TLS termination and auth.

Typical Server Software Stack



Exporters

Built into services or sidecar

PromSD Sidecar

Target configuration

queries for available targets

PromSD Proxy

Service discovery and TLS exporter proxy

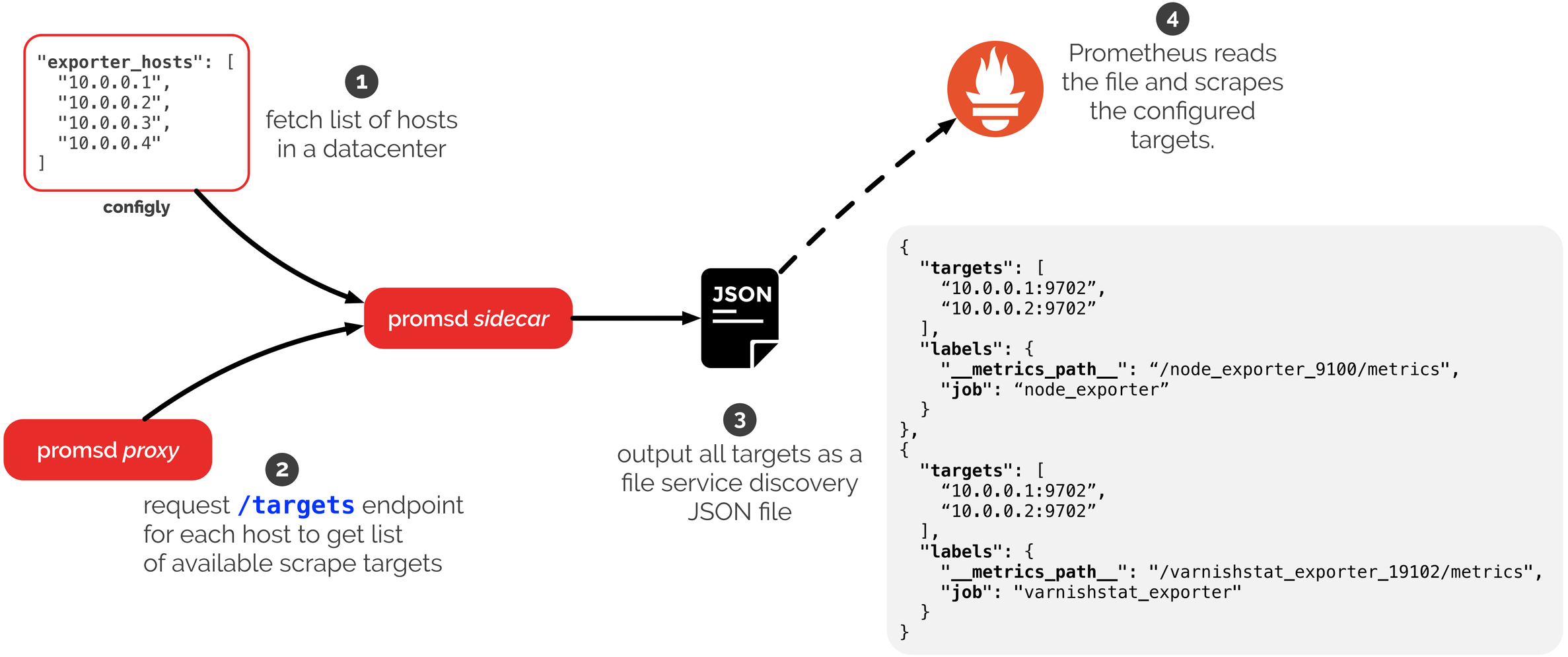
generates config for prometheus

Prometheus

scrapes proxied targets over TLS

fastly

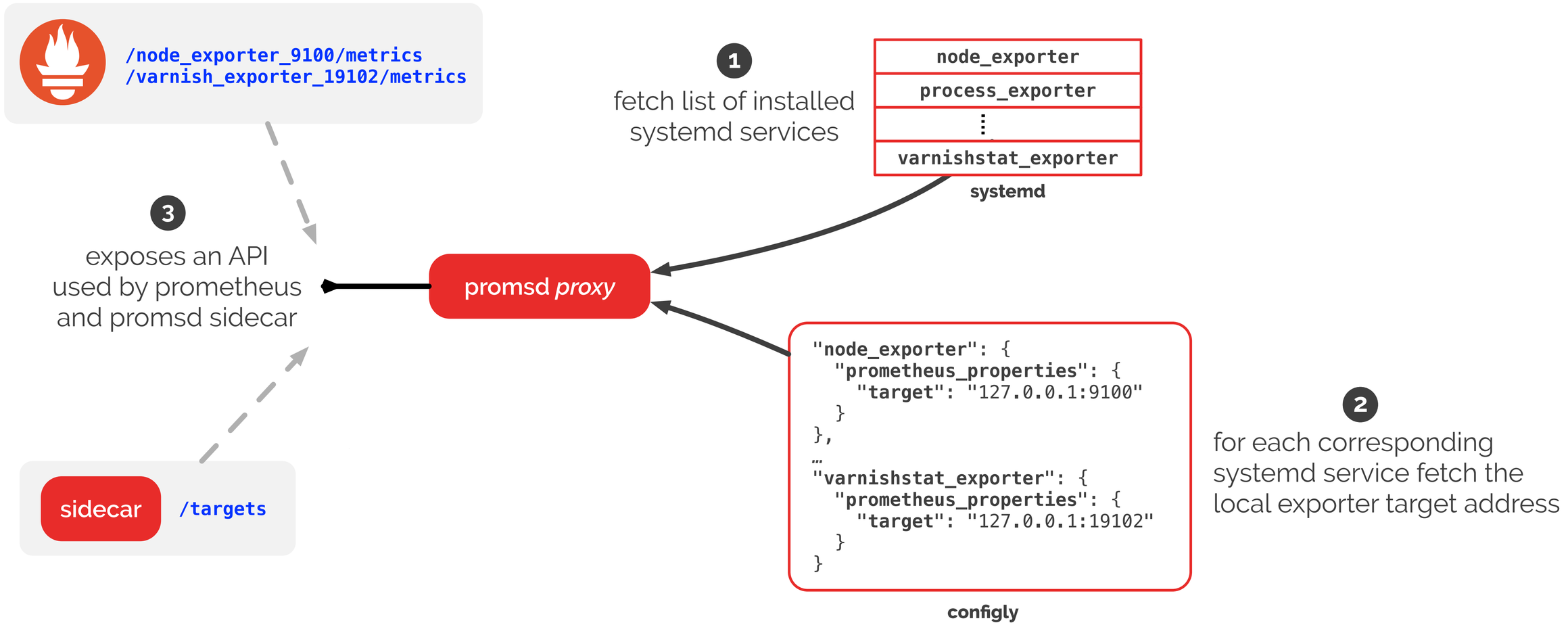
PromSD sidecar




```
{
  "targets": [
    “10.0.0.1:9702”,
    “10.0.0.2:9702”
  ],
  "labels": {
    "__metrics_path__": “/node_exporter_9100/metrics",
    "job": “node_exporter”
  }
},
{
  "targets": [
    “10.0.0.1:9702”,
    “10.0.0.2:9702”
  ],
  "labels": {
    "__metrics_path__": "/varnishstat_exporter_19102/metrics",
    "job": "varnishstat_exporter"
  }
}
```

fastly

PromSD proxy



fastly

It worked!

- Really easy to leverage the file SD mechanism.
- New targets can be added with one line of config.
- TLS and authentication everywhere.
- Single exporter port open per host.

Prometheus Adoption

Prometheus at Scale at Fastly

114 Prometheus servers globally

28.4M time series

2.2M million samples/second

... and growing!

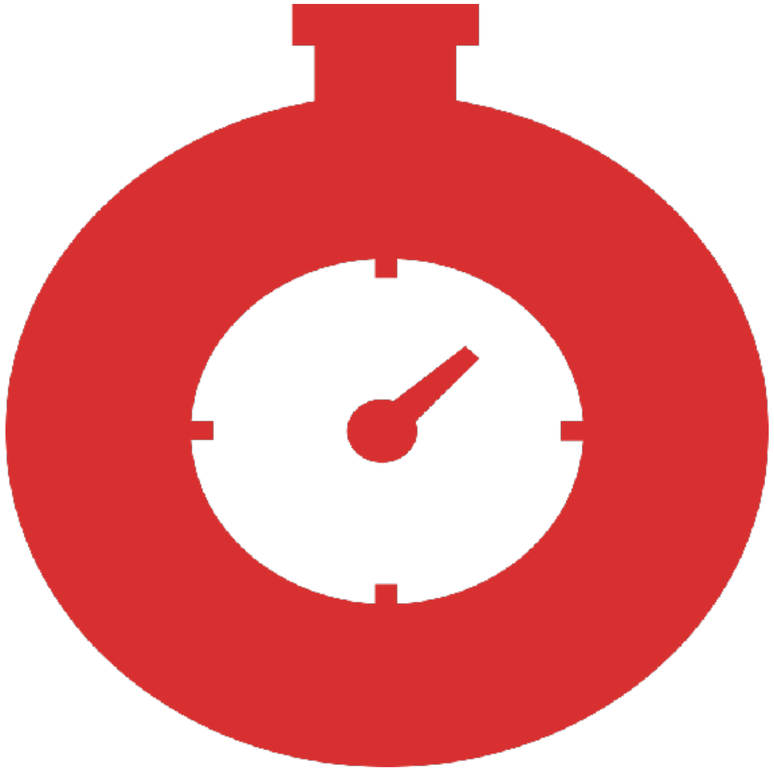
fastly

Prometheus wins

- Engineers love it.
- Dashboard and alert quality have increased.
- PromQL enables some deep insights.
- Scaling linearly with our infrastructure growth.

Still some rough edges.

- Metrics exploration without prior knowledge.
- Alertmanager's flexibility.
- Federation and global views.
- Long term storage still an open question.



fastly

Thanks!

@ickymettle

fastly.com

monitorama slack
#talk-marcus-barczak

fastly