

Monitoring at Scale

Migrating to Prometheus at Fastly



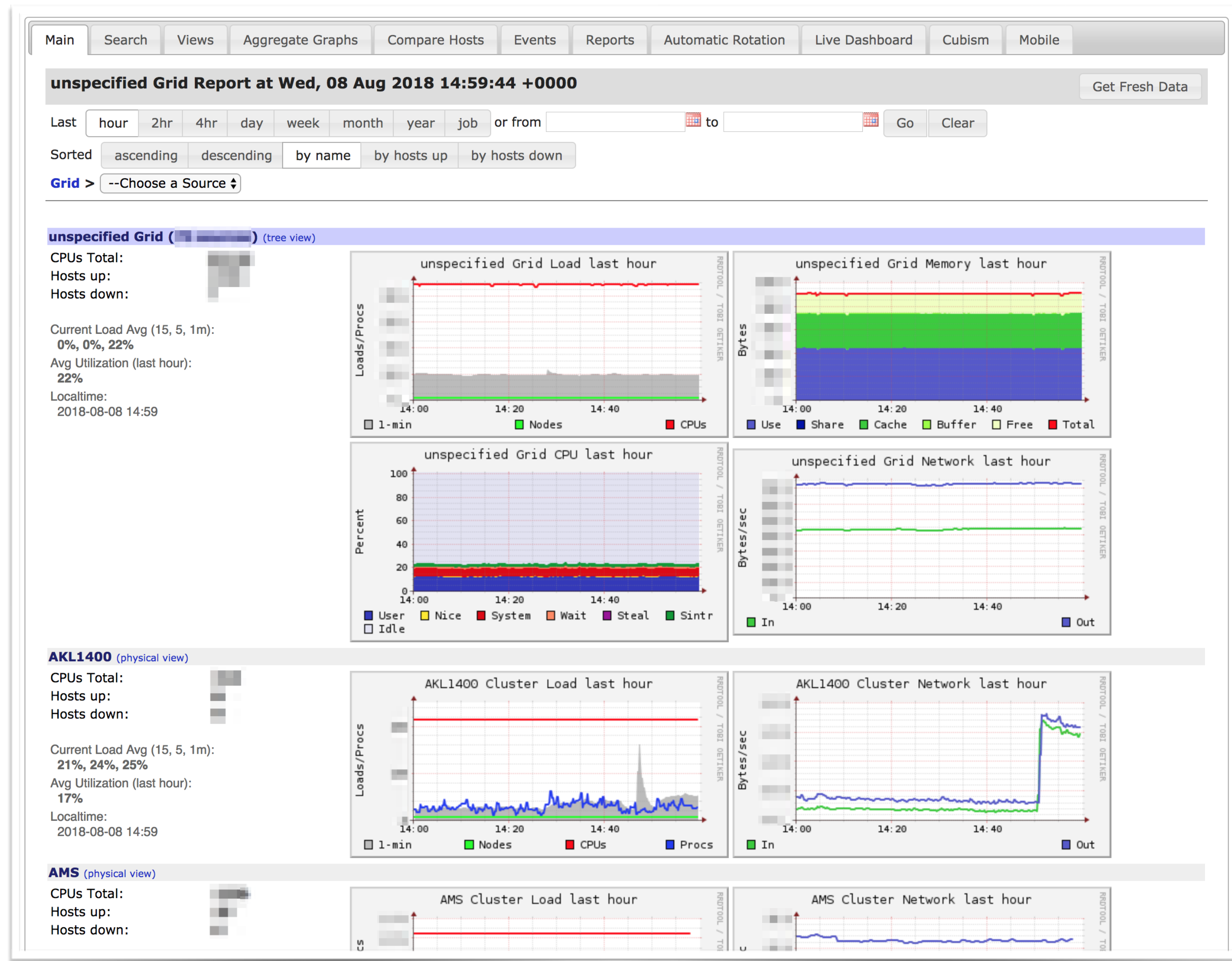
PROMCON 2018 | Marcus Barczak
@ickymettle

fastly[®]





How **were** we
monitoring Fastly?



Growing pains with Ganglia

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



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 reinvigorate our monitoring culture.

See: <https://peter.bourgon.org/observability-the-hard-parts/>

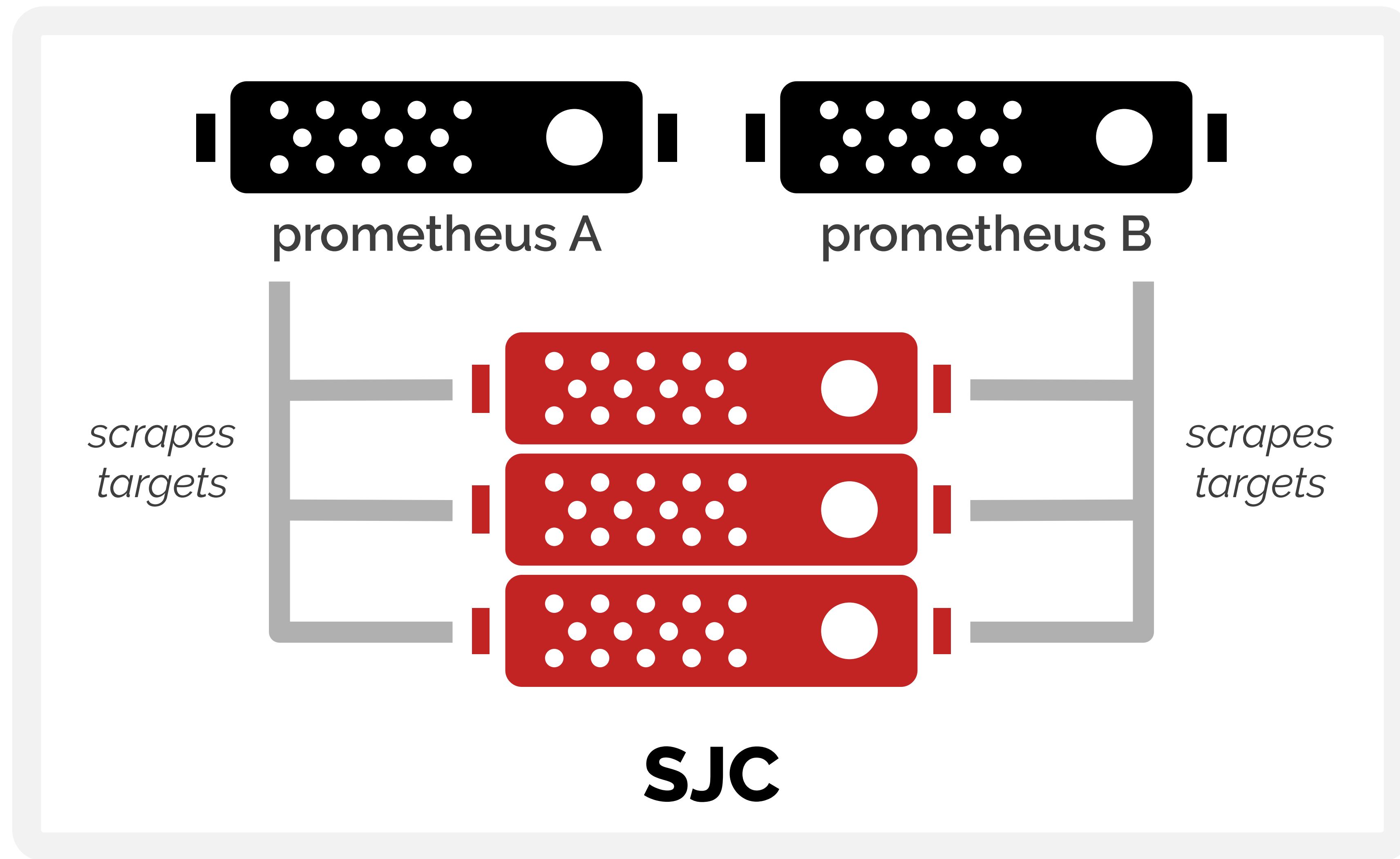


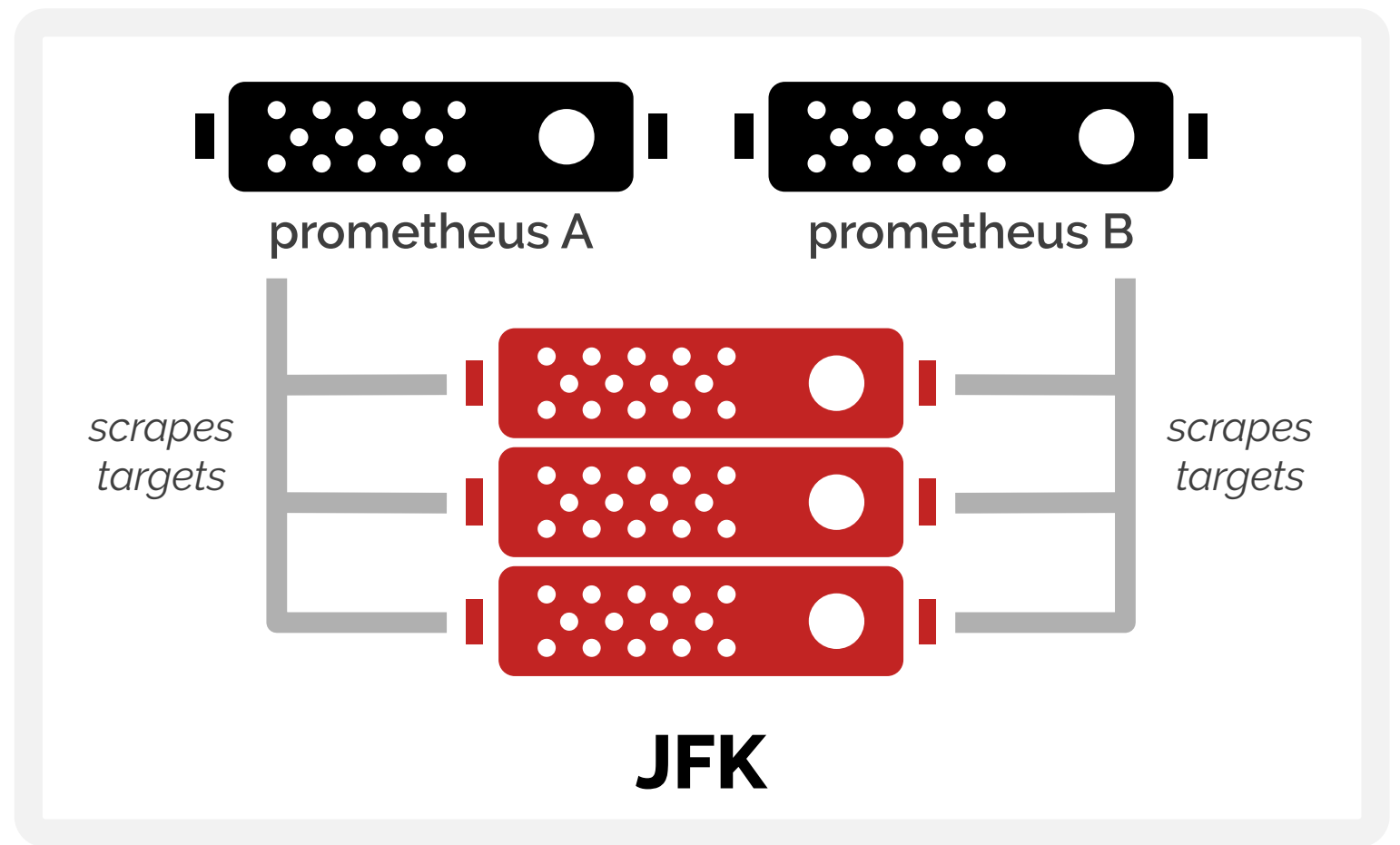
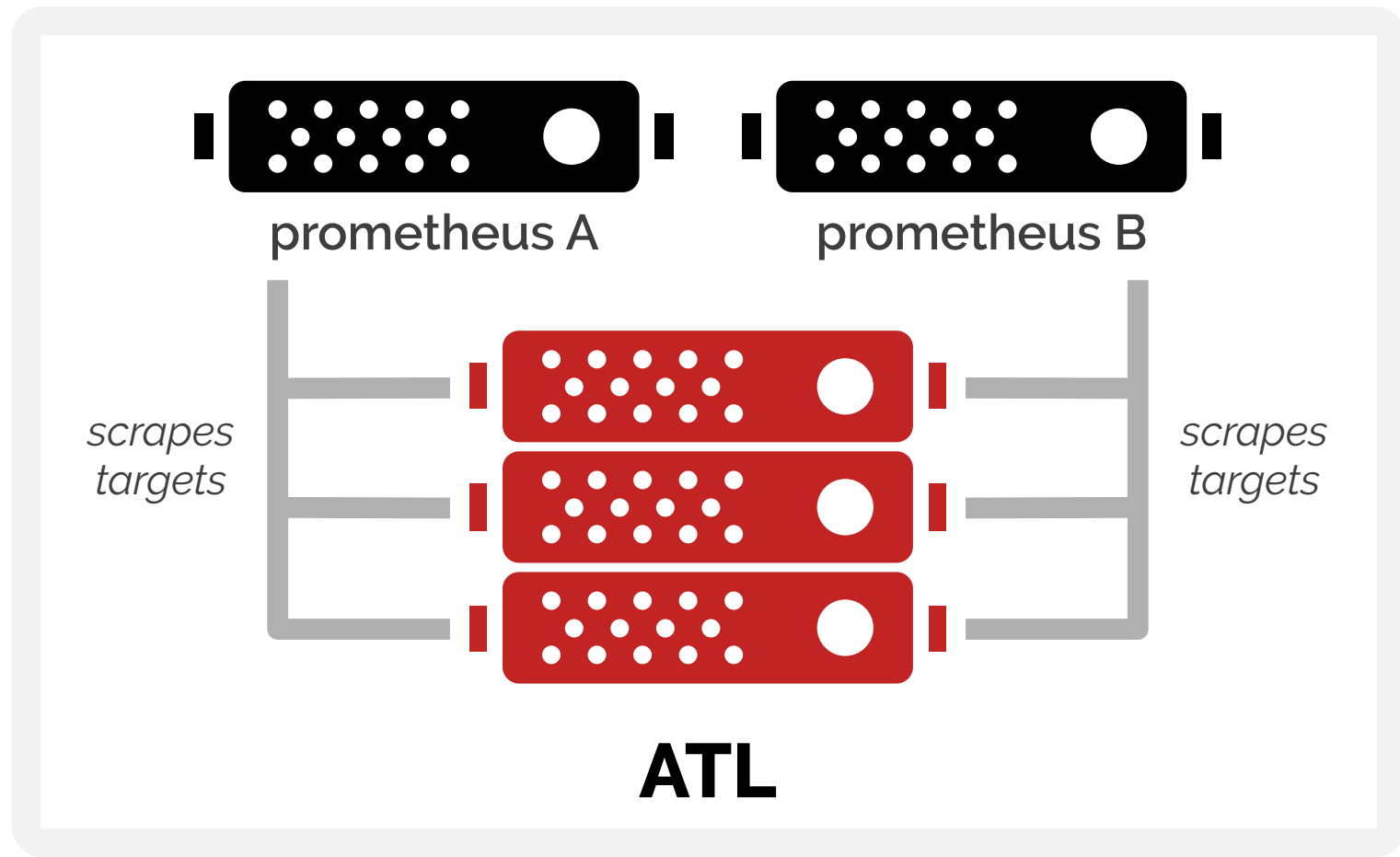
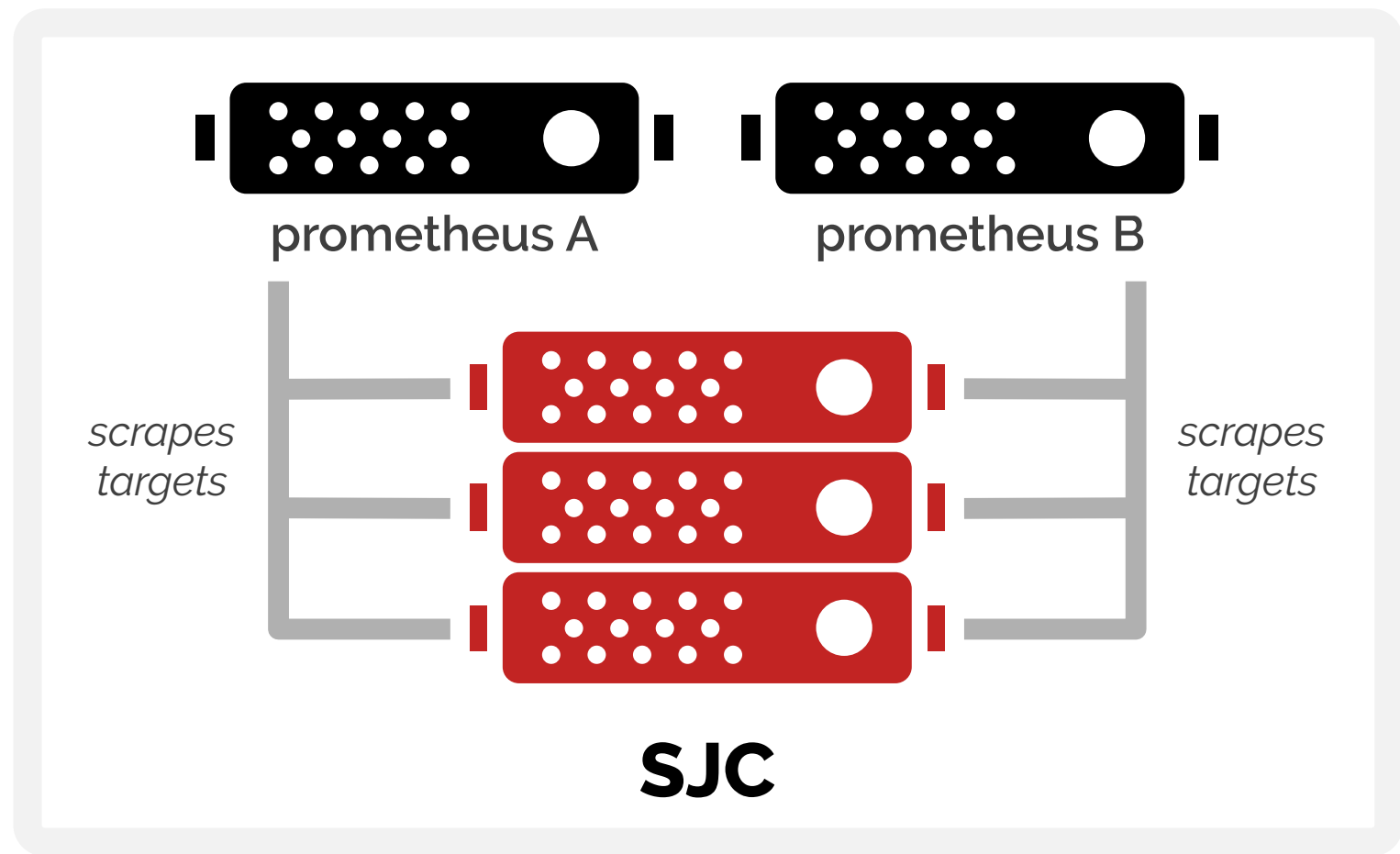


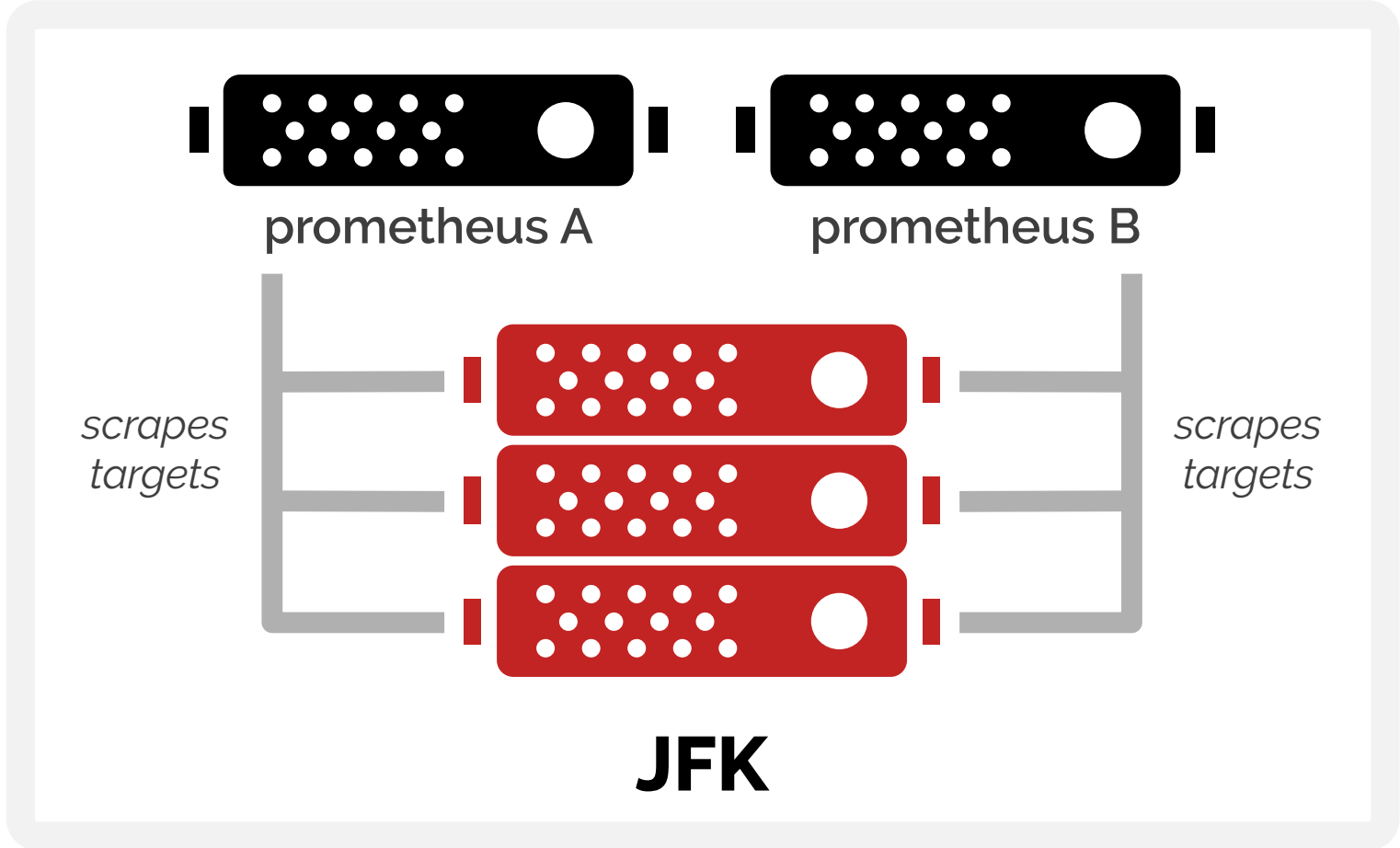
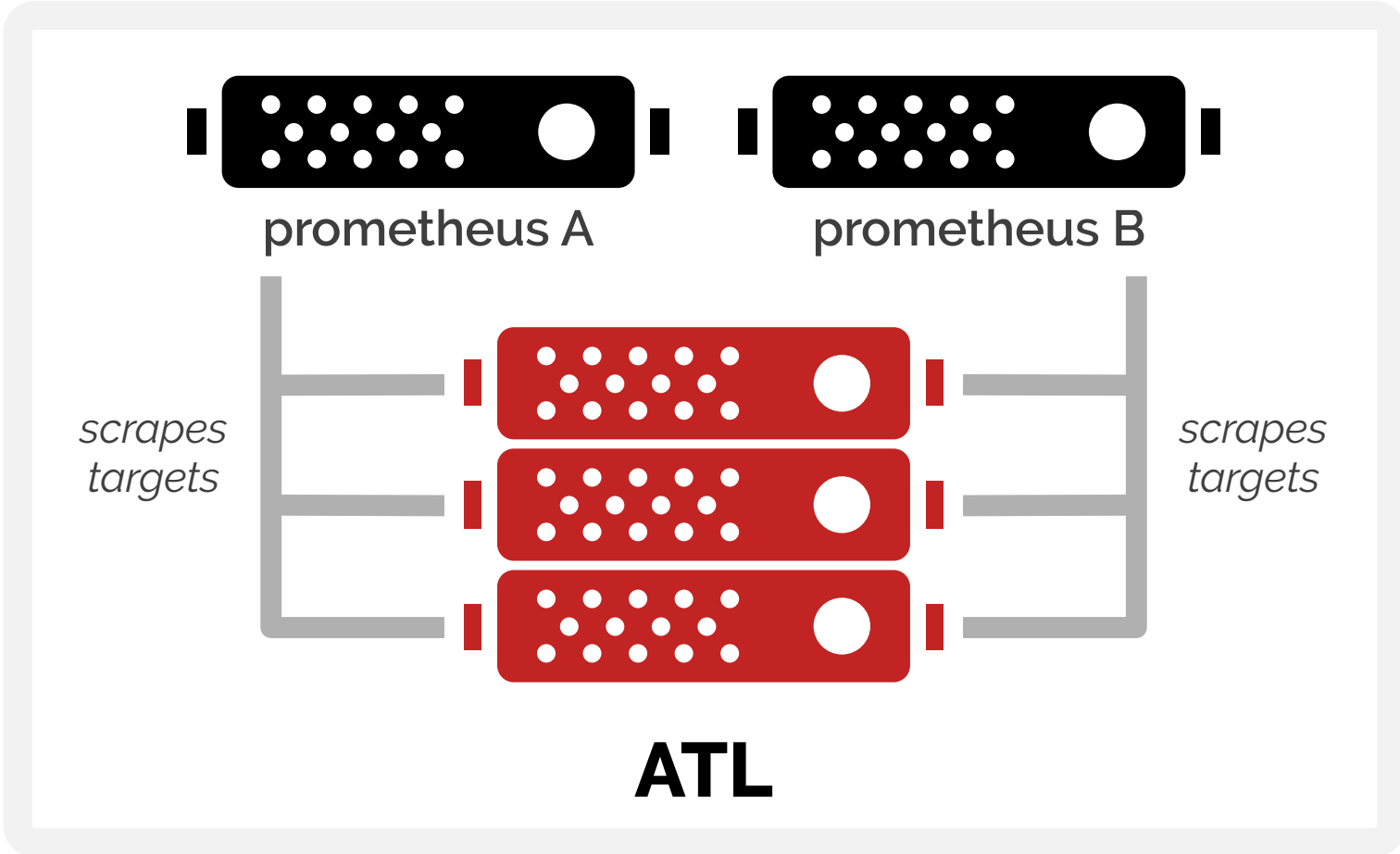
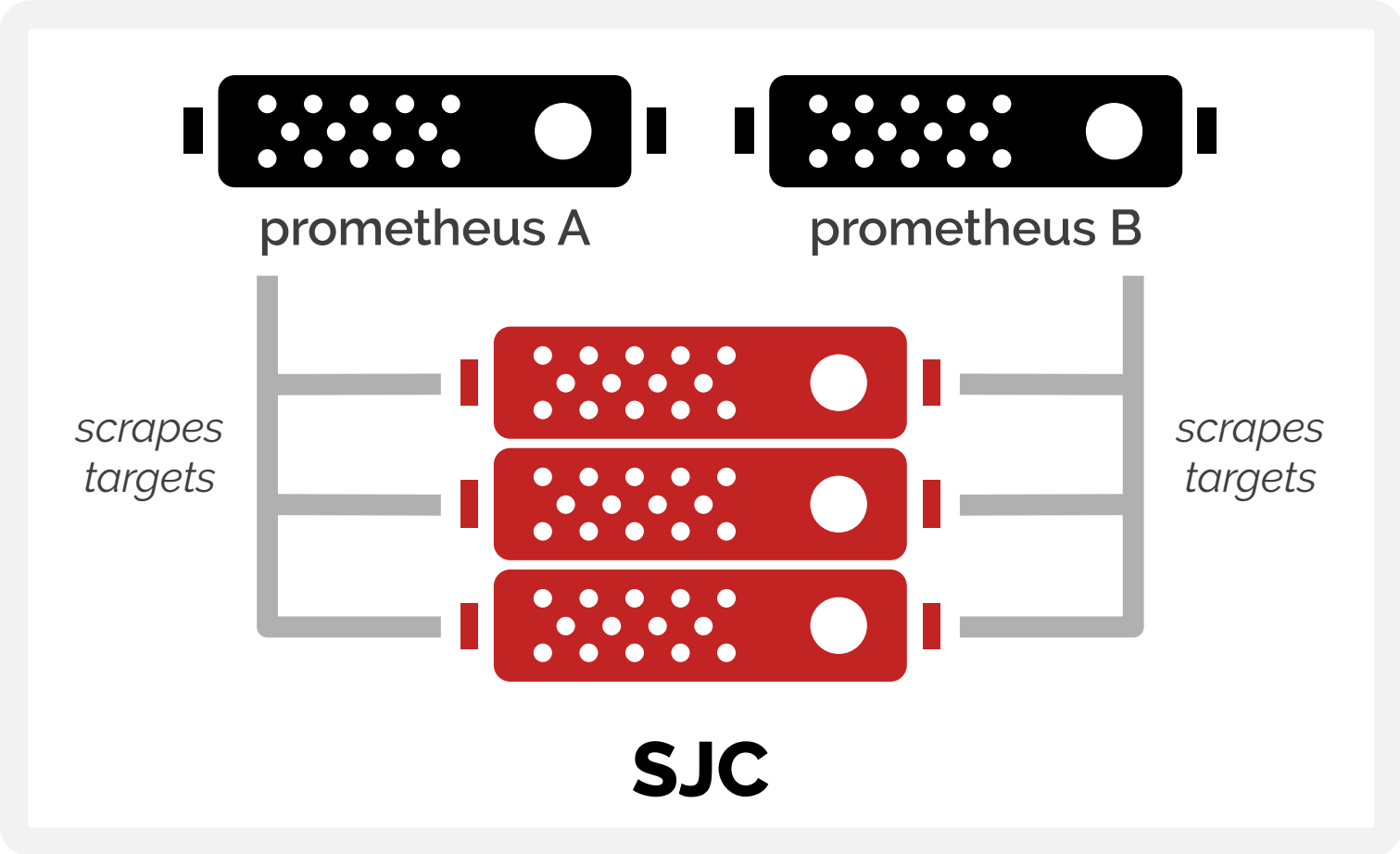
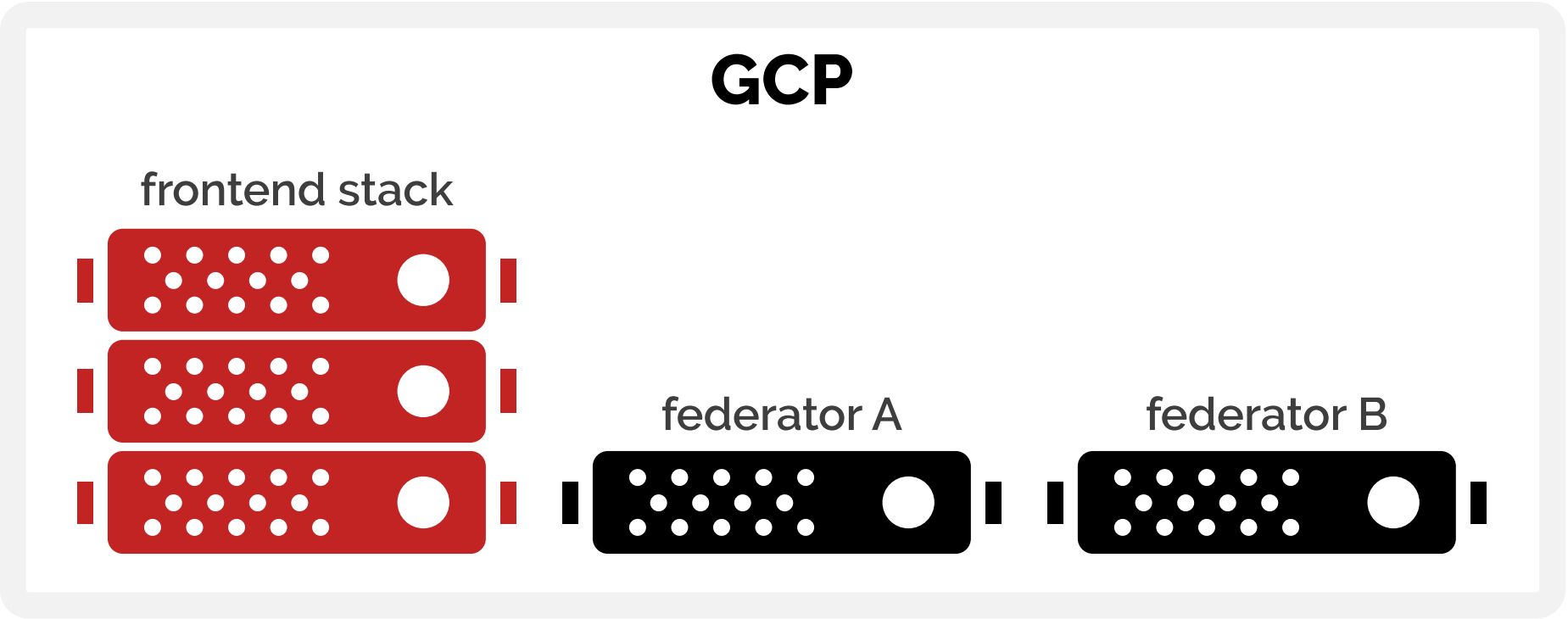
Getting started

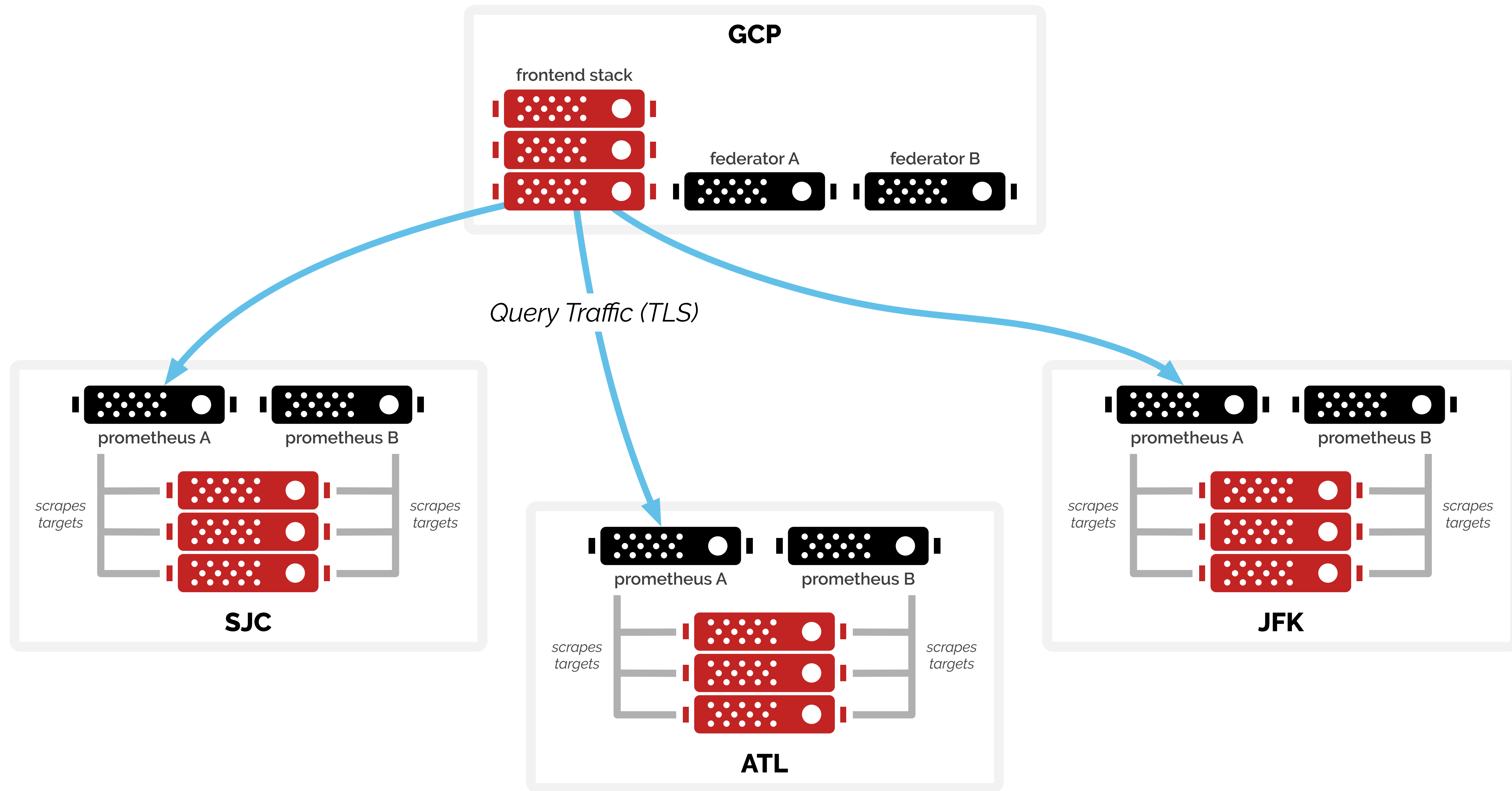
- ① Build a proof of concept.
- ① Pair with pilot team to instrument their services.
- ① Iterate through the rest.
- ① Run both systems in parallel.
- ① Decommission SaaS system and Ganglia.

Infrastructure build

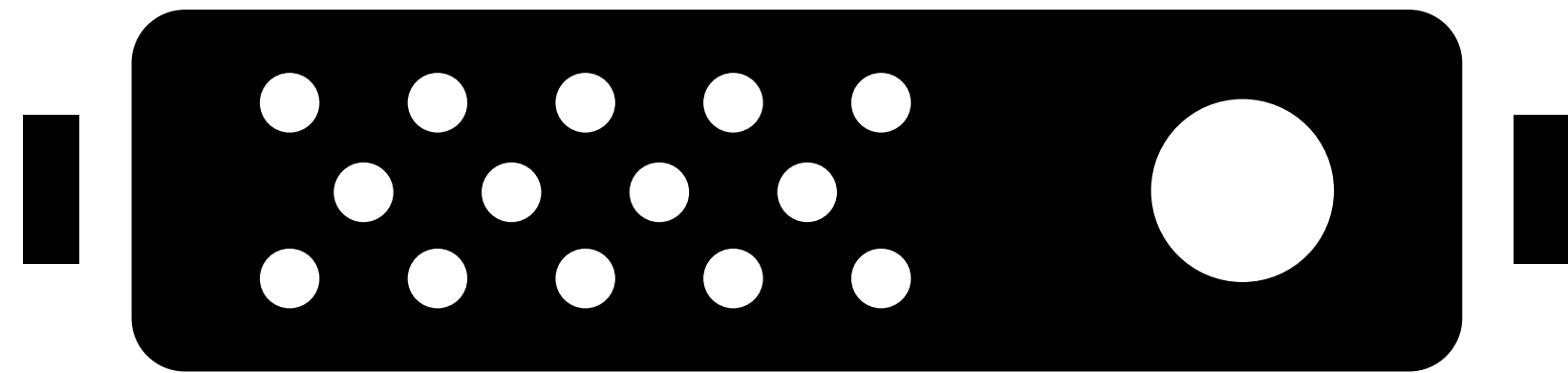








Prometheus Server Software Stack



Ghost Tunnel

TLS termination and auth.

Service Discovery Sidecar

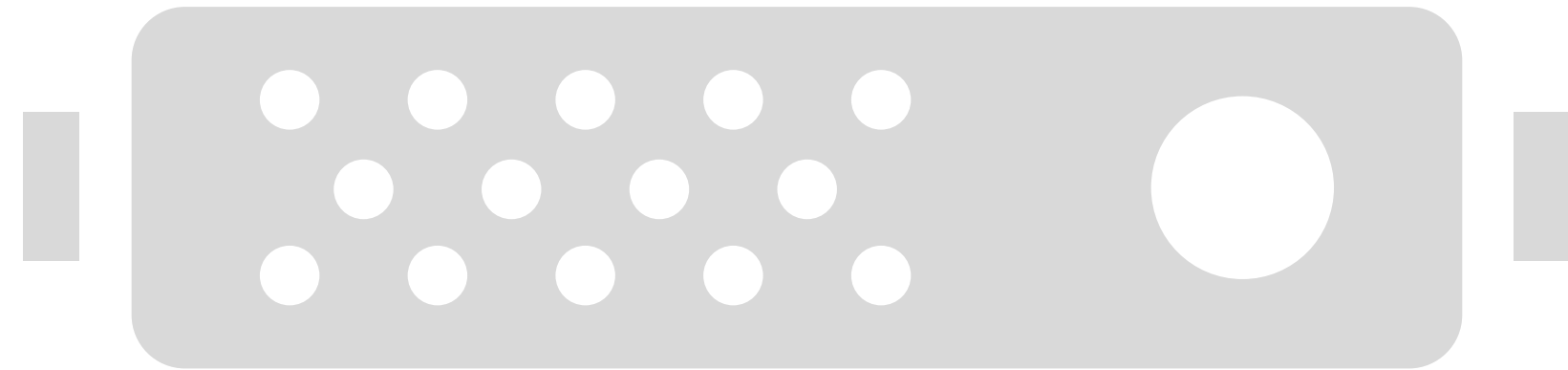
Target configuration

Rules Loader

Recording and Alert rules

Prometheus

Prometheus Server Software Stack



Ghost Tunnel

TLS termination and auth.

Service Discovery Sidecar

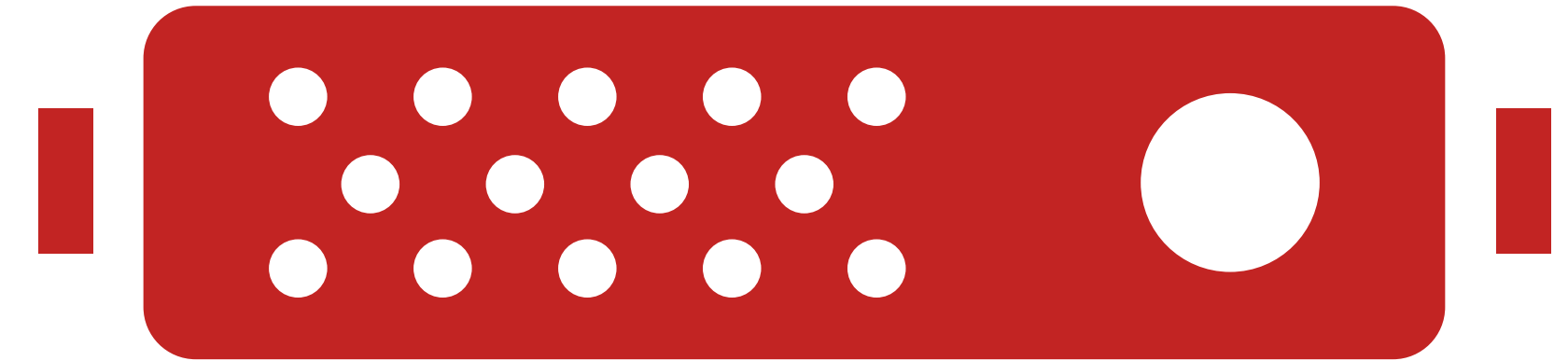
Target configuration

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

**Build your own
service discovery?**

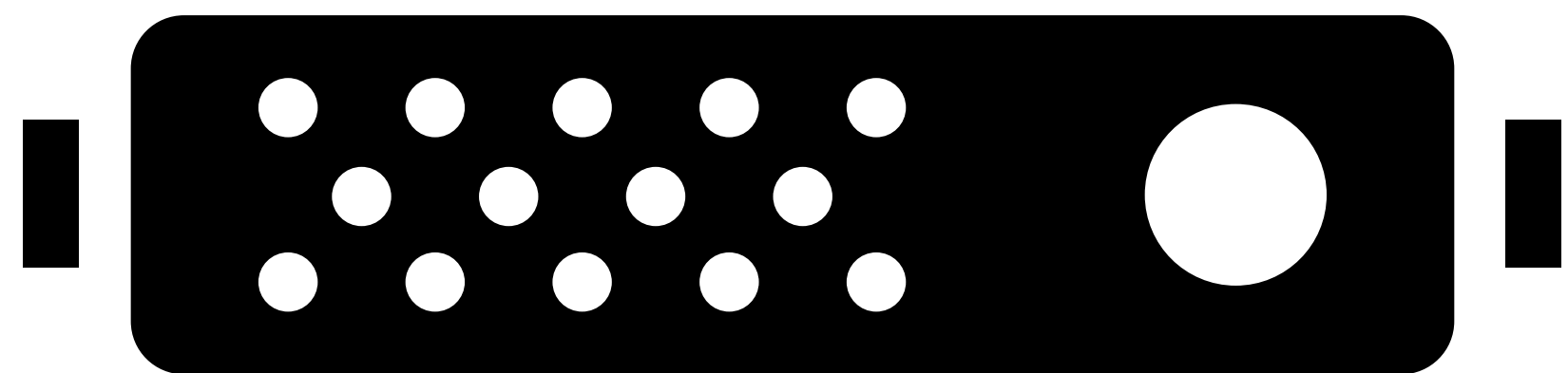


Fastly's infrastructure
is bare metal hardware
no cloud conveniences

Service discovery requirements

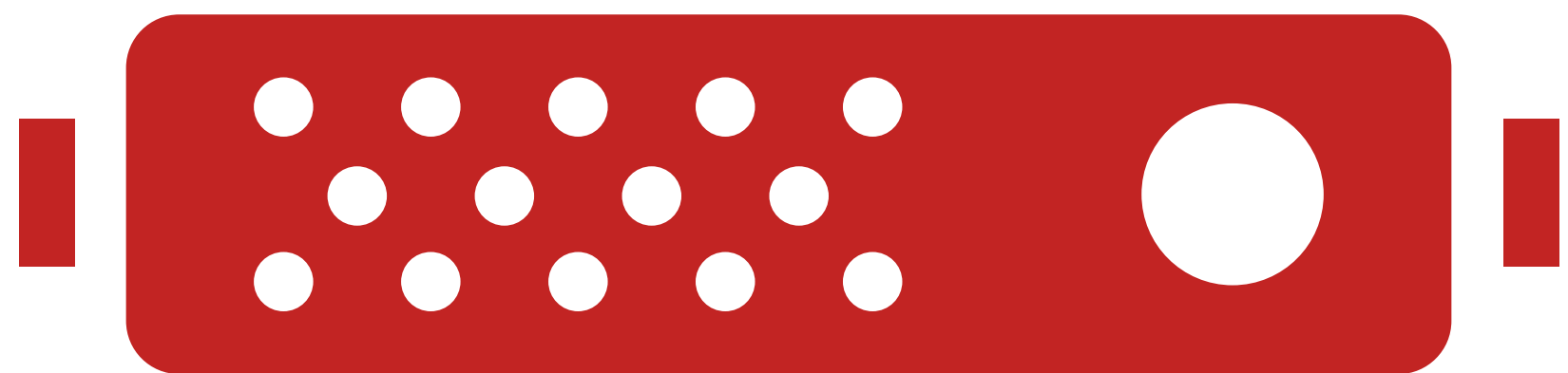
- ⦿ Automatic discovery of targets.
- ⦿ Self-service registration of exporter endpoints,
- ⦿ TLS encryption for all exporter traffic.
- ⦿ Minimal exposure of exporter TCP ports.

Prometheus Server Software Stack



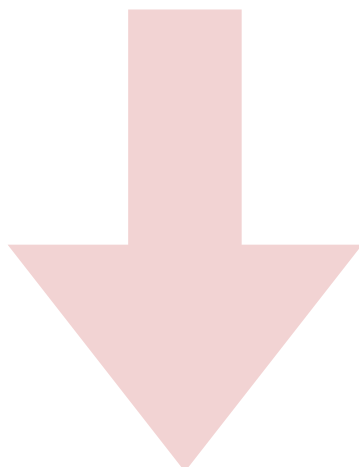
Ghost Tunnel
TLS termination and auth.

Typical Server Software Stack



Exporters
Built into services or sidecar

PromSD Sidecar
Target configuration



generates config
for prometheus

Prometheus



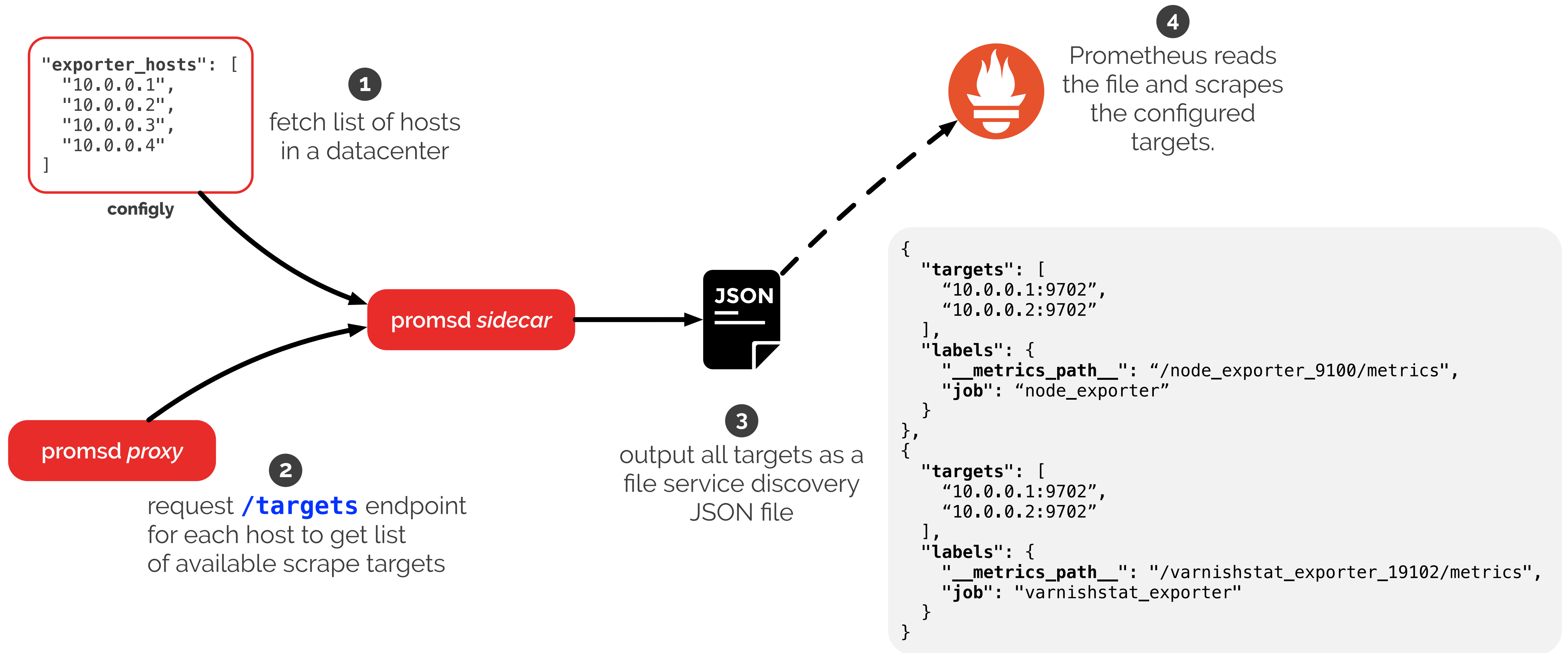
queries for
available targets

PromSD Proxy
Service discovery and
TLS exporter proxy

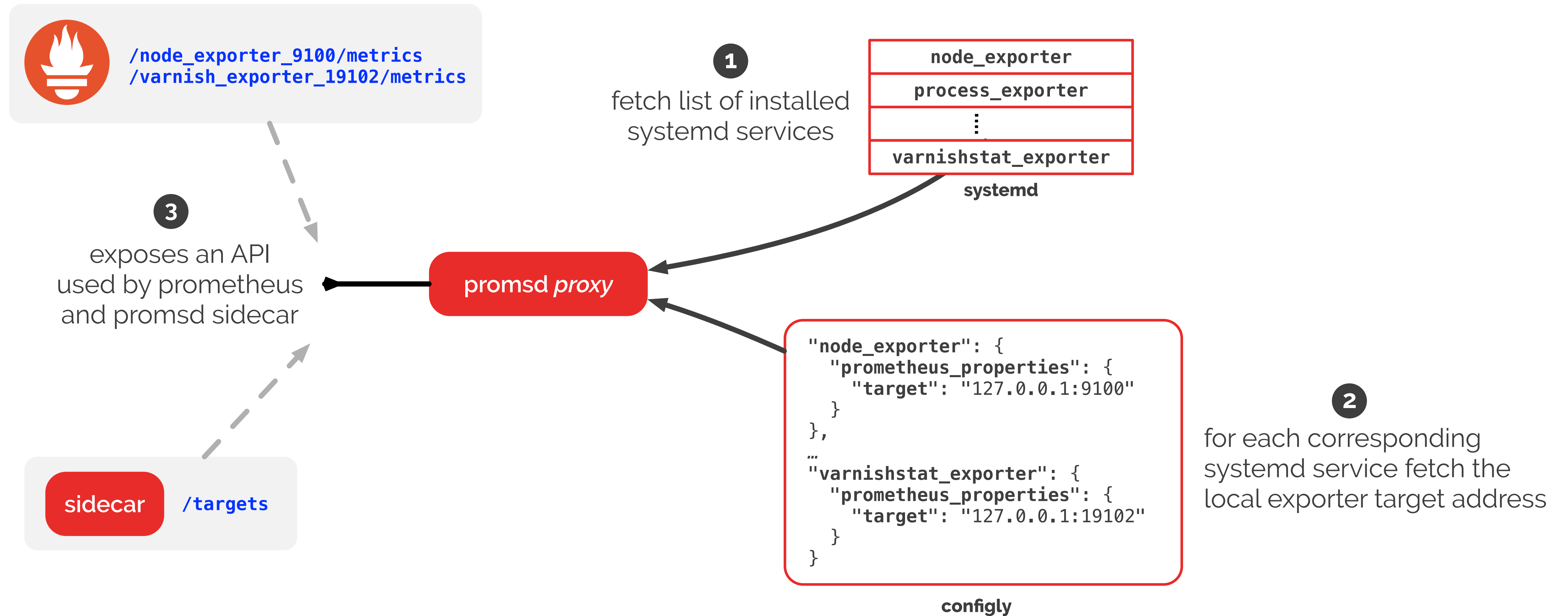


scrapes proxied targets over TLS

PromSD sidecar



PromSD proxy



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



developer

I'm still looking at the functions available in Prometheus and it seems even more constrained than **vendor**

... a few hours later



developer

it's *much* better than **vendor** 😄 and so much faster!

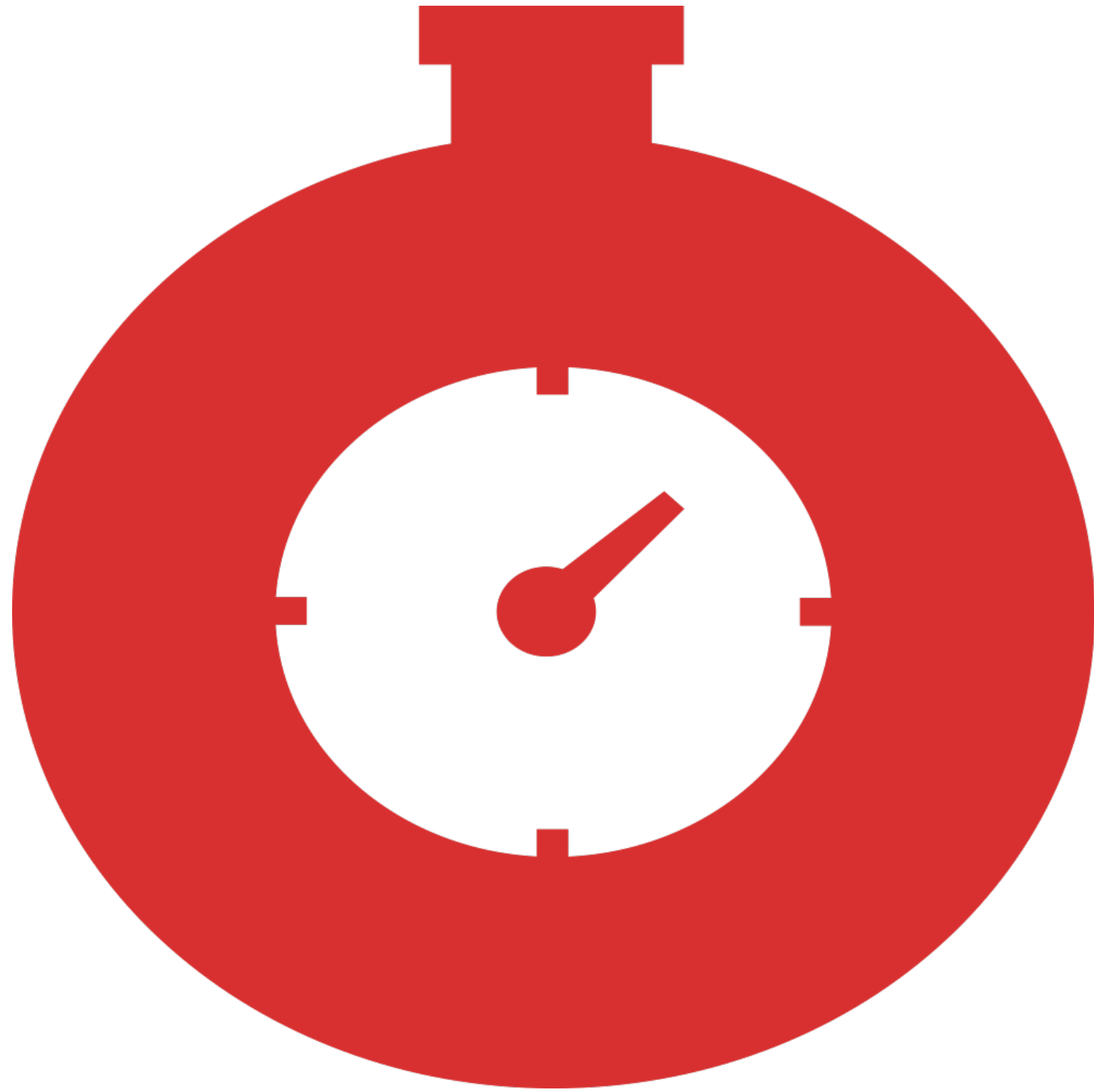


Prometheus wins

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

Still some rough edges.

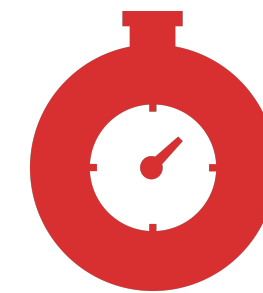
- ⦿ Metrics exploration without prior knowledge.
- ⦿ Alertmanager's flexibility.
- ⦿ Federation and global views.
- ⦿ Long term storage still an open question.



Thanks!



@ickymettle



fastly.com