

LINE

Prometheus as a (internal) service

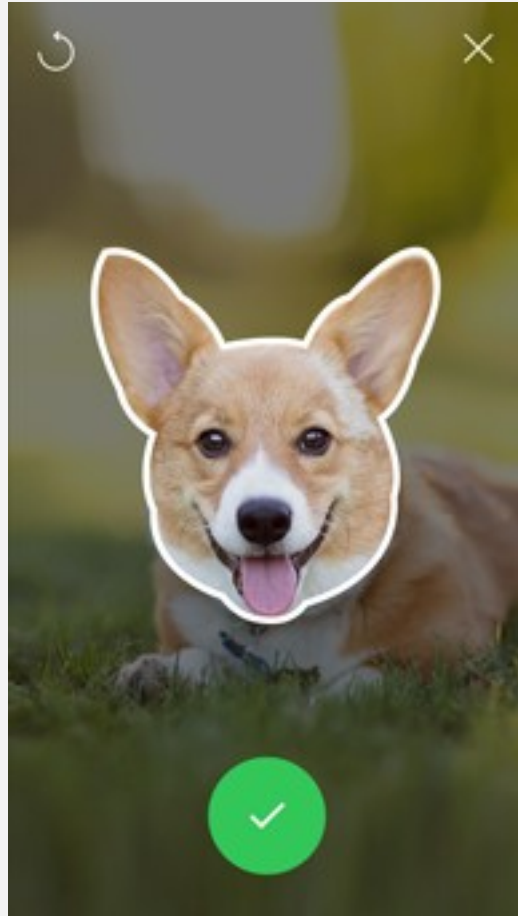
Paul Traylor
LINE Fukuoka

Self-Introduction

- Wanted to make games in high school
 - Worked on several mods creating levels
 - Decided games were hard, web development looked easier
- North Carolina State University - Computer Science
- Worked in San Francisco for ~7 years
 - First job primarily web development
 - Second job primarily devops
- LINE Fukuoka ~ 1 year
 - Focused primarily on upgrading monitoring tools

LINE Fukuoka – LINE Family Apps

- LINE Creators Studio
- LINE Fortune
- LINE Surveys
- LINE Part Time
- and more!

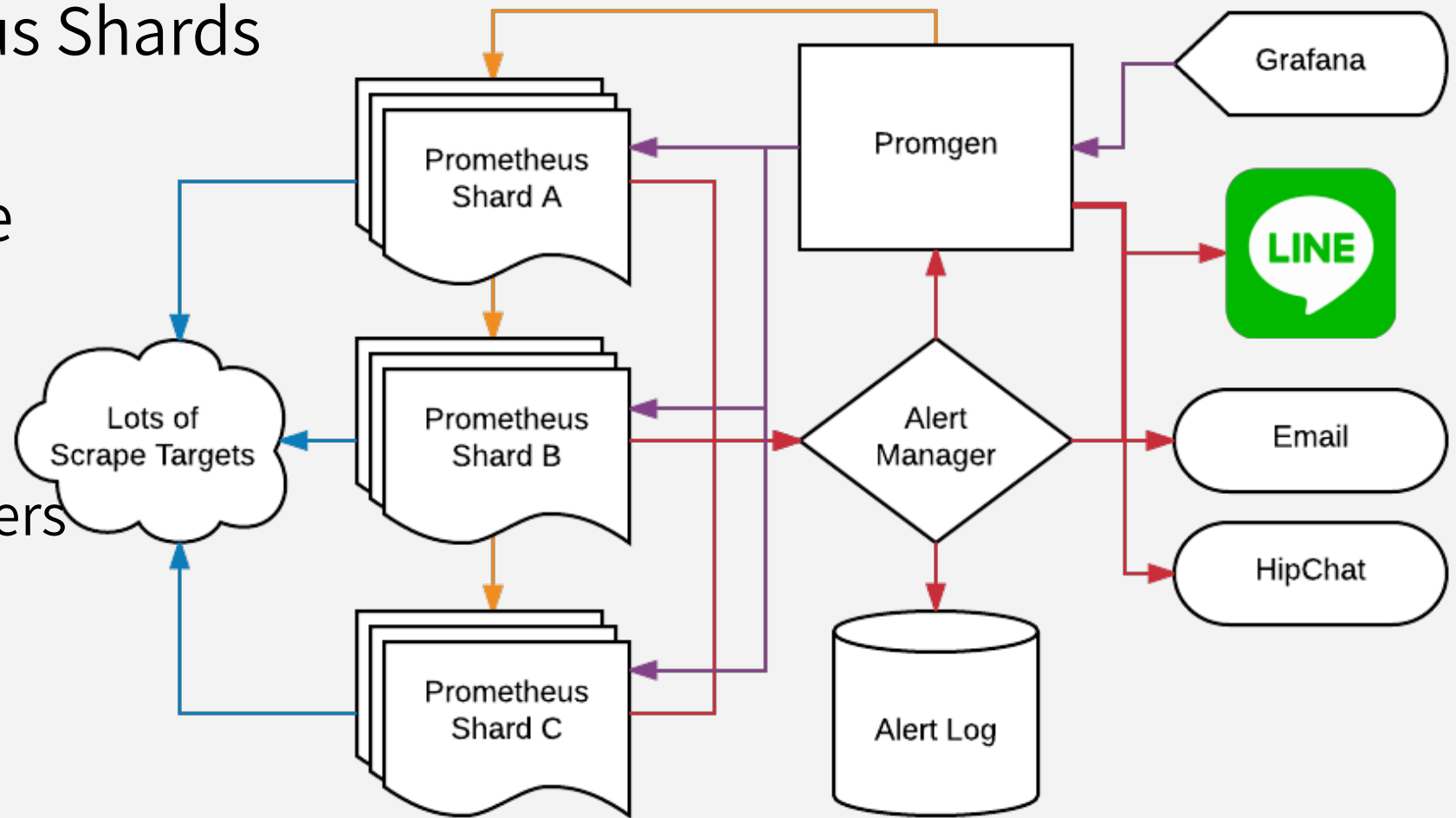


Current Responsibilities

- Continue development on Promgen
 - Introduced last year at Promcon ¹
 - Rewritten in Django
 - <https://github.com/line/promgen>
- Migrate legacy monitoring to Prometheus
 - Installing exporters
 - Setting Prometheus targets
 - Configuring rules

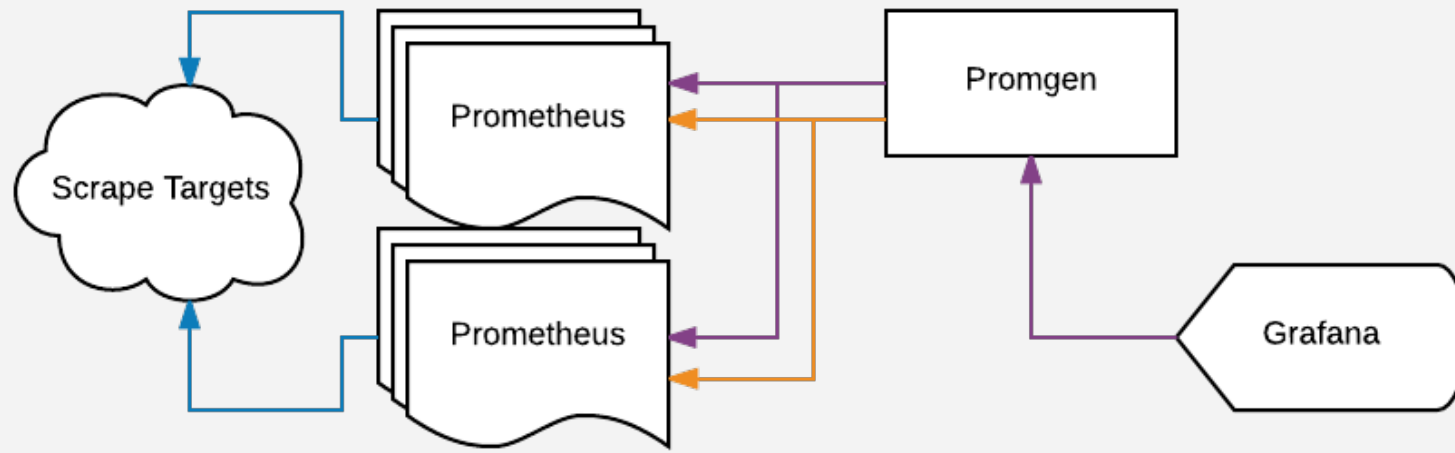
Environment

- HA Prometheus Shards
- LB Promgen
- LOTS of scrape targets
 - ~3.5 million samples
 - ~3000 exporters



Environment – Promgen + Prometheus

- Manages targets and rules for Prometheus
- All exporters go into a single json file
- All rules go into a single rules file
- Uses relabeling on the Prometheus side filter out unrelated shards



Environment – Promgen + Prometheus

```
global:
  scrape_interval:      15s
  evaluation_interval: 15s

  external_labels:
    cluster_name: 'shard-name'

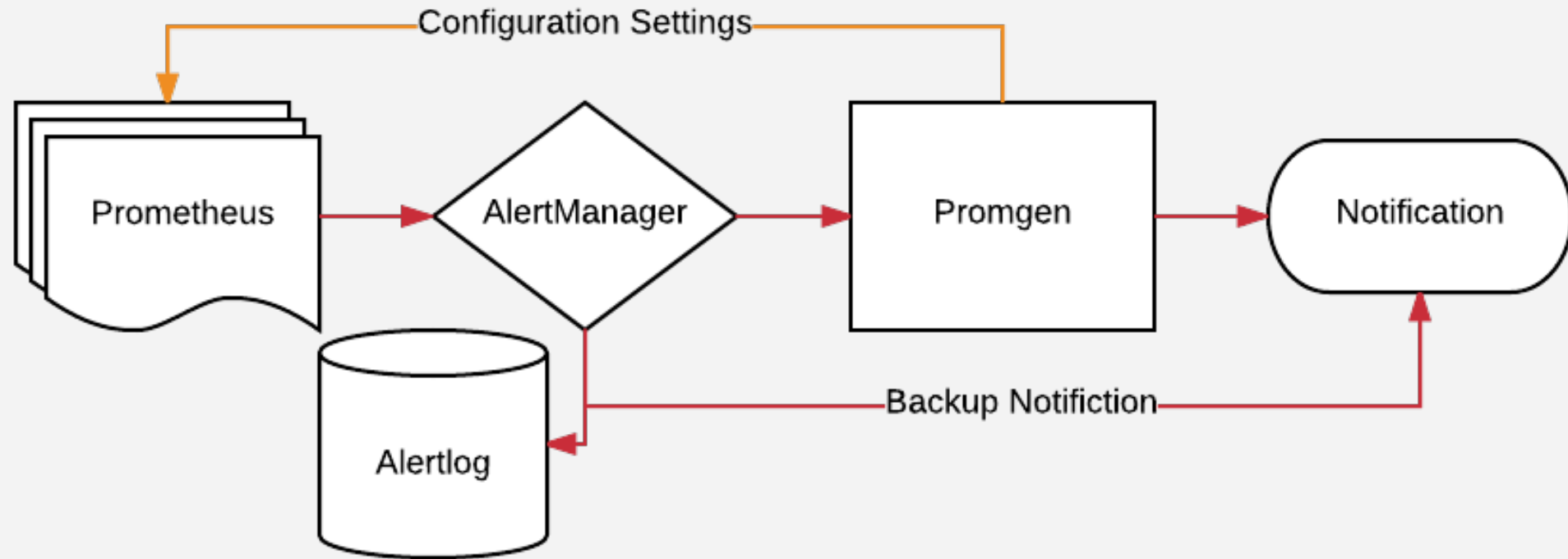
rule_files:
- "/path/to/common.rule"
- "/path/to/promgen.rule"

scrape_configs:
- job_name: 'promgen'
  file_sd_configs:
    - files:
      - "/path/to/promgen.json"

  relabel_configs:
- source_labels: [__shard]
  regex: {{ shard_keep }}
  action: keep
```


Environment – Promgen + Alertmanager

- Hard to build a dynamic routing tree in Alertmanager
- Route everything through Promgen
- But Provide a backup in case Promgen goes down



Environment – Promgen + Alertmanager

```
route:
  receiver: default
  group_by: ['service', 'project']
  routes:

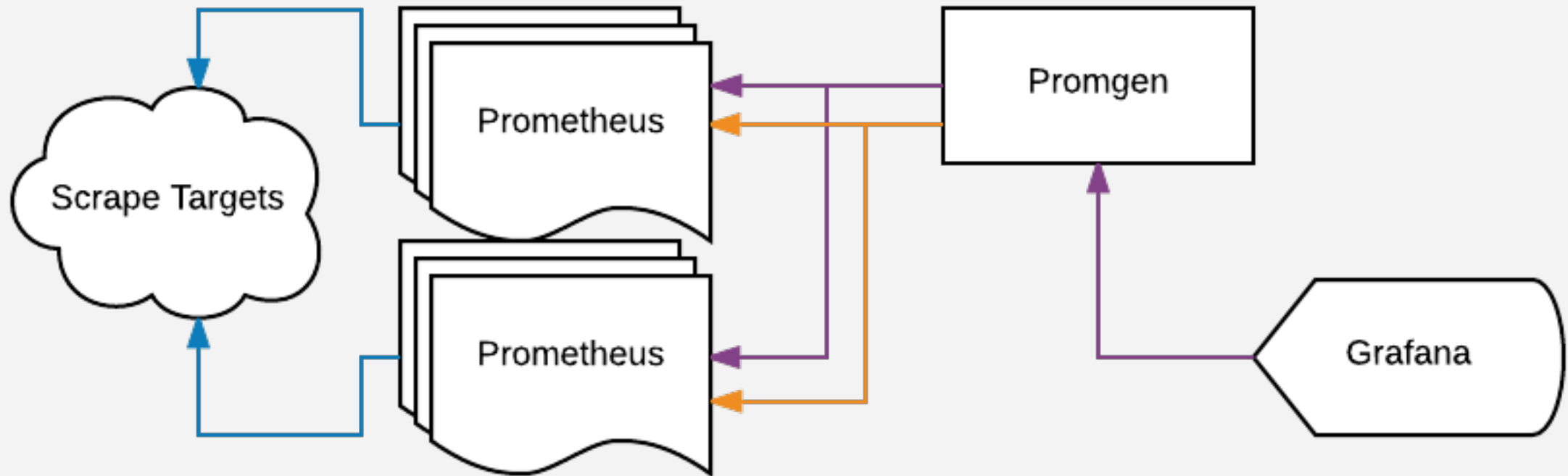
# Anything that matches the job
# alertmanager, should be routed
# directly since other parts of
# the system may not be working
# correctly
- receiver: backup
  match:
    job: alertmanager

- name: default
  webhook_configs:
  - url: http://alertlog.example.com
    send_resolved: true
  - url: http://promgen.example.com
    send_resolved: true

- name: backup
  email_configs:
  - to: backup@example.com
    send_resolved: true
```

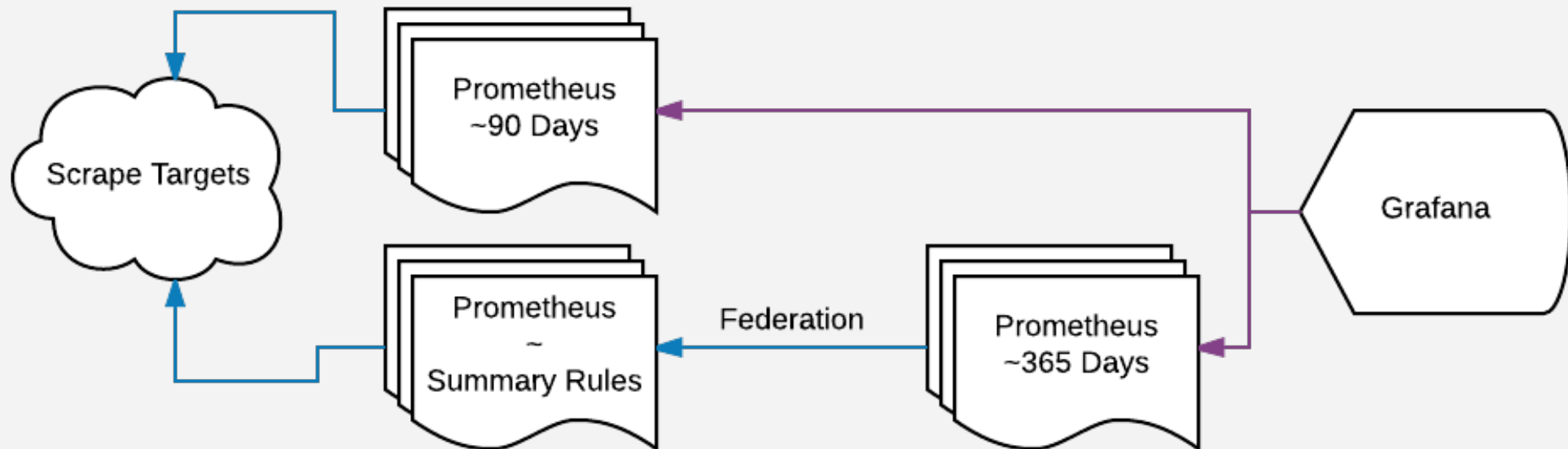
Operation – Where is my shard?

- Use Promgen to query each shard and then combine the result
- Still interested in native support for Prometheus remote_read



Operation – Retention

- Prometheus cares about the present (and recent past)
- Developer's mostly care about the present
- Sometimes want to compare with historical data
- Wanted to use the same queries for both recent and longterm queries



Operation – Retention

- Has not scaled well
- Thought we could co-locate both
 - Summary servers use a lot of network / cpu to fetch and summarize
 - Longterm servers only need to retain the data
- Lots of memory problems
 - Summary Prometheus eats resources due to twice the workload of a normal server
 - Sometimes longterm process dies sometimes summary process dies
 - Need various extra alert rules to watch both

Operation - Watching upstream projects

- Prometheus is generally fairly stable from a cli / api perspective but there are still times when things change
 - <https://github.com/prometheus/prometheus/pulse>
 - <https://github.com/prometheus/alertmanager/pulse>
- Federation instance labels #2488
 - meta:samples:sum{} = sum(scrape_samples_scraped) BY (service, project, job)*
 - meta:alerts:count{} = count(ALERTS == 1) BY (service, project, alertname, alertstate)*
 - meta:exporters:count{} = count(up) BY (service, project, job)*

Operation – Fixed Labels

Recording Rule

```
meta:samples:sum{} = sum(scrape_samples_scraped) BY  
  (service,project,job)
```

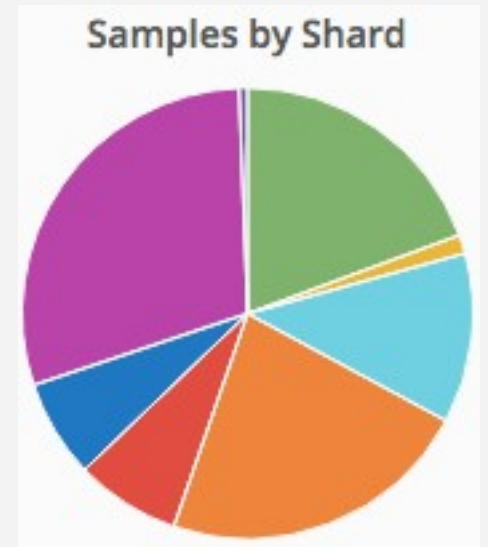
Prometheus Configuration

```
relabel_configs:
```

```
- source_labels: [__address__]  
  target_label: cluster_node  
  action: replace
```

Grafana Query

```
sum(avg(meta:samples:sum) without (cluster_node)) by  
  (cluster_name)
```



Education - Writing Queries is hard

- PromQL can be unfamiliar
 - <https://prometheus.io/docs/querying/functions/>
- Hard to remember labels at times
 - Lots of service labels
 - Lots of project labels
 - Lots of mountpoints for disk metrics
 - Lots of interfaces for network metrics
 - ...

Education – Writing Queries is hard

- without|by ?
- ignoring|on?

```
max by (instance) (node_filesystem_free / node_filesystem_size)
```

```
max by (instance, mountpoint) (node_filesystem_free{fstype!~"tmpfs|rootfs"} / node_filesystem_size{fstype!~"tmpfs|rootfs"})
```

Education - Writing rules is harder

- Is loadavg useful ?
- Global rule namespace
- Promgen uses Service and Project labels to route on, but easy to lose your label when writing queries

Global Rule excludes children

```
example_rule{service!~"A/B", }:
```

```
  # Service A override includes self
```

```
  - example_rule{service="A", }
```

```
  # Service B override includes self, but excludes children
```

```
  - example_rule{service="B", project!~"C"}:
```

```
    # Project Override
```

```
    - example_rule{project="C" }
```

Education - Grafana default dashboards

- Try to provide basic dashboards for the most common metrics
 - node_exporter / nginx_exporter / mysql_exporter / etc
- Use templating from Grafana to drill down
 - Shard -> Service -> Project -> Instance
 - Prototyping simple proxy so that developers can ignore shard
- Currently building most dashboards manually
 - Difficult to update when you want to change navigation
 - Considering ways to auto generate dashboards

Promgen – One Year Later

- Rewritten in Django to take advantage of ORM and admin page
- Better rule editor
- Sharding
- Prometheus Proxy

Promgen – Want to build a better rule editor

Clause:

Prometheus: [Basics](#) [Operators](#) [Functions](#) [Examples](#) Promgen: [Rules](#)

Example Query `node_load1{service="Default", <exclude>} > 5`

```
max(1 - (node_filesystem_free{<exclude>} / node_filesystem_size{<exclude>}))  
BY (mountpoint, instance, service, project) > 0.9
```

- insert metric at cursor -

[Test Query](#)

Promgen – Easier notification settings

Project: paul.project

[Home](#) / [docker-demo](#) / [paul.service](#) / [paul.project](#) / Register Notifier

promgen.notification.email

[promgen.notification.ikasan](#)

[promgen.notification.linenotify](#)

[promgen.notification.webhook](#)

Simple plaintext Email notification

Email Address:

Alias:

Register Notifier