

Intro to Prometheus

(application monitoring tool)

*The best way to learn
something
is to teach it*

Goal

Goal

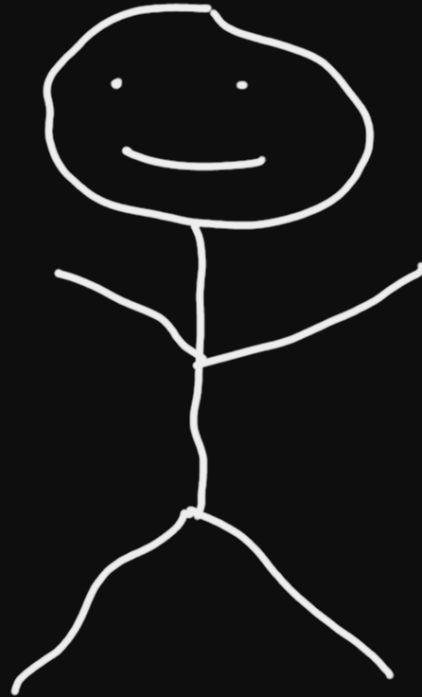
1. Give an overview of what Prometheus is

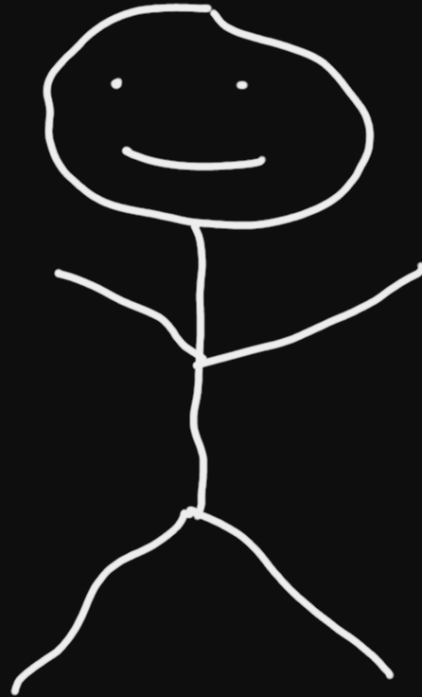
Goal

1. Give an overview of what Prometheus is
2. That's it

Goal

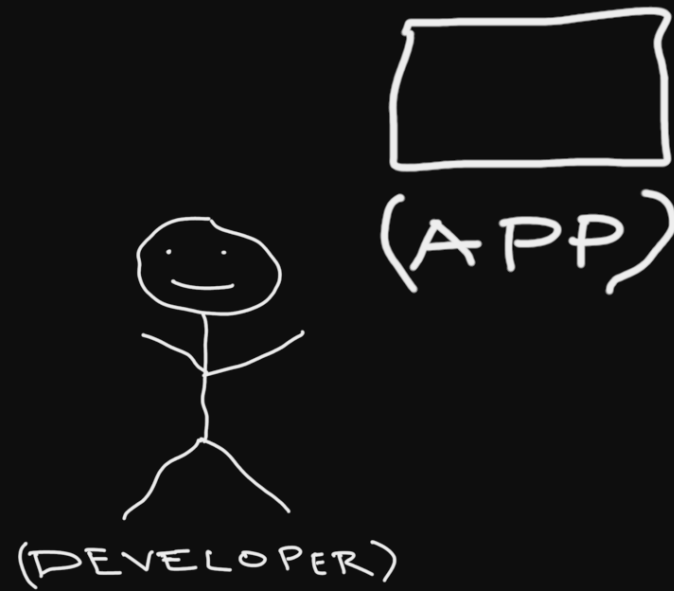
1. Give an overview of what Prometheus is
2. That's it
3. OK, maybe do a "demo": how to use it in the context of PHP/Symfony

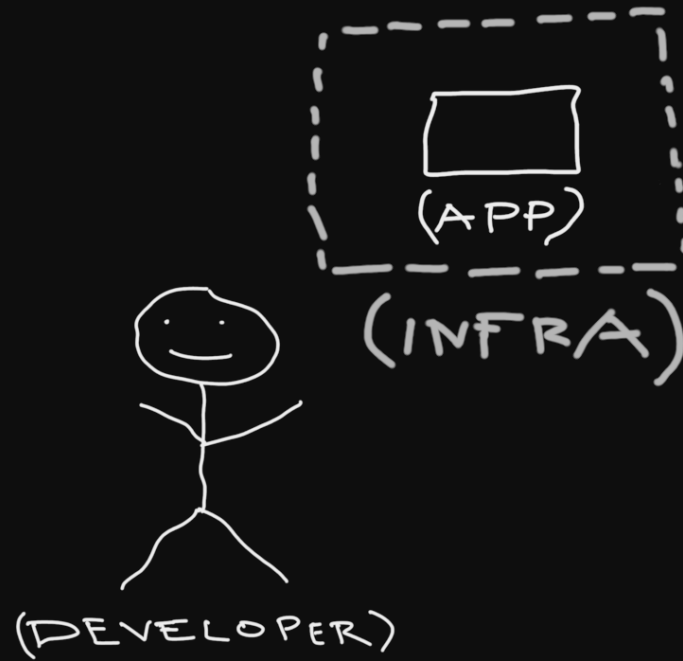


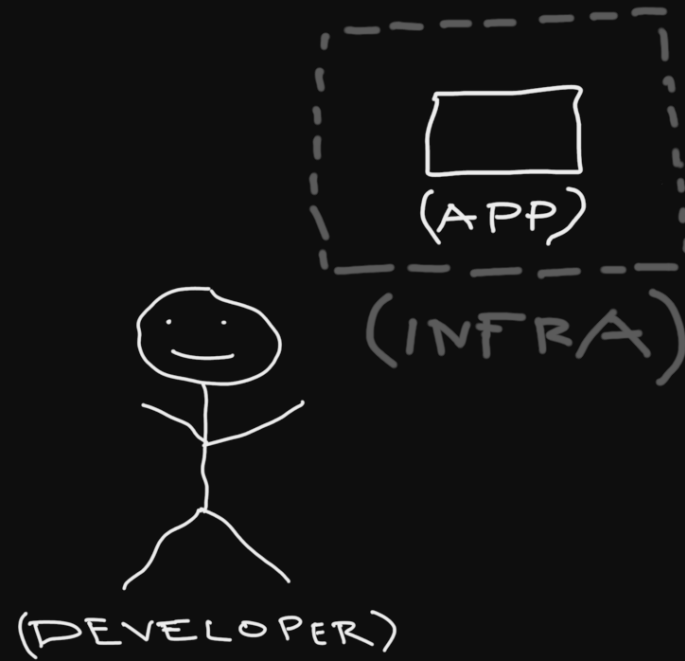


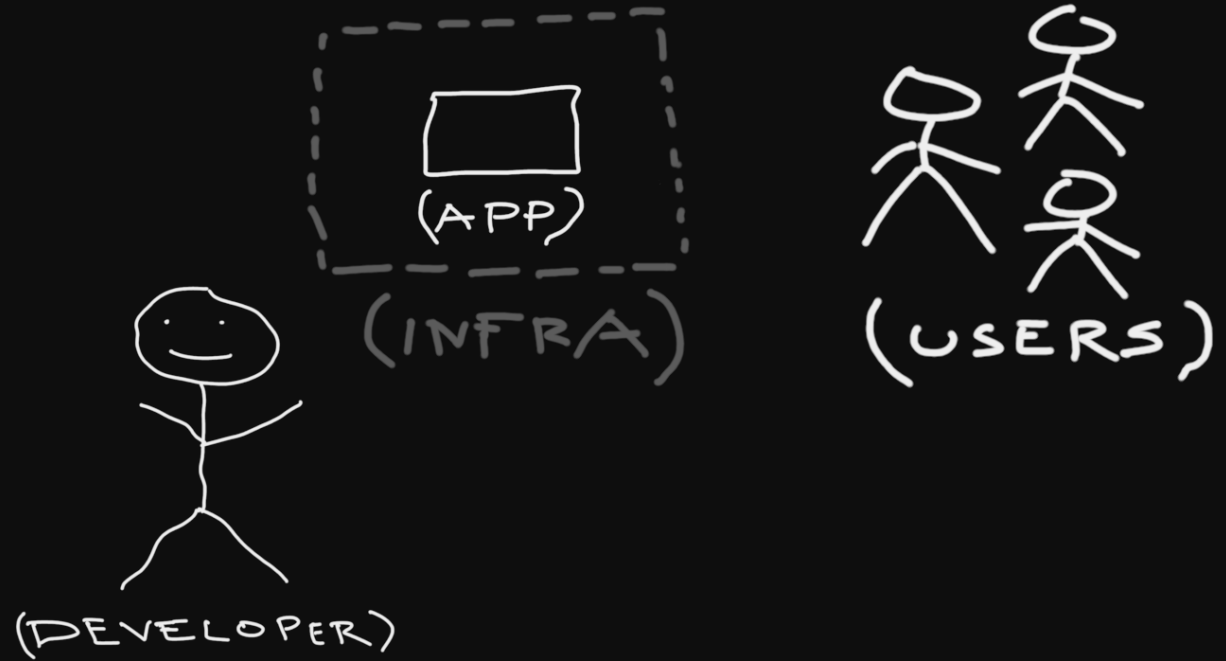
(DEVELOPER)

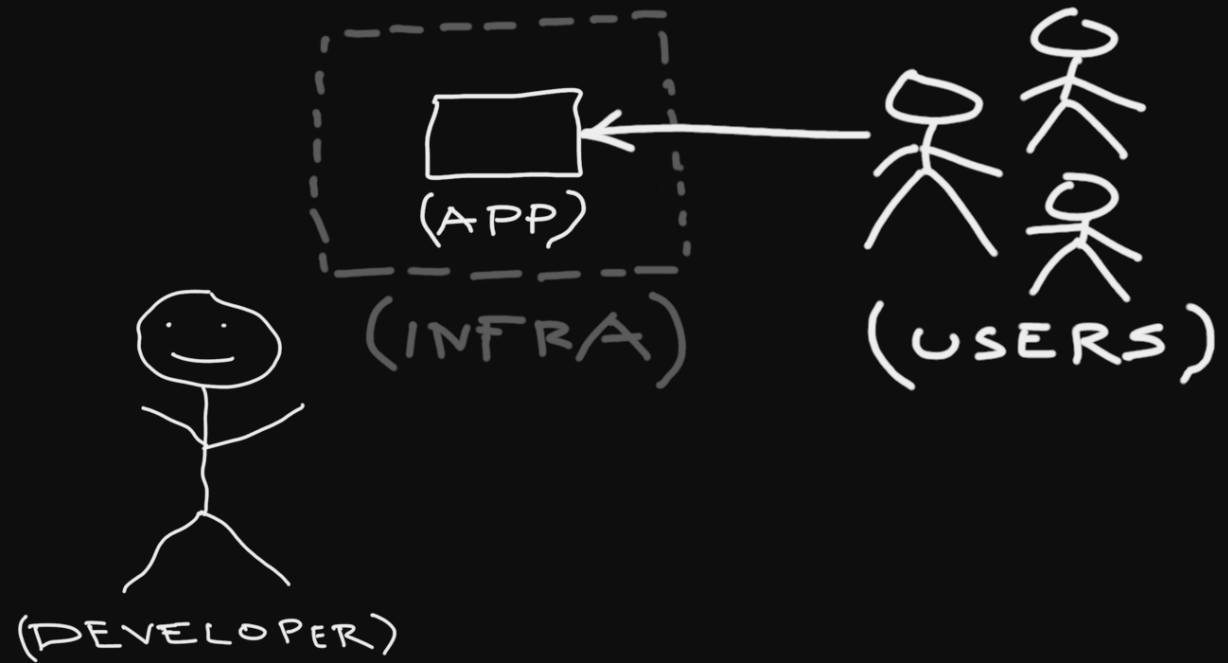


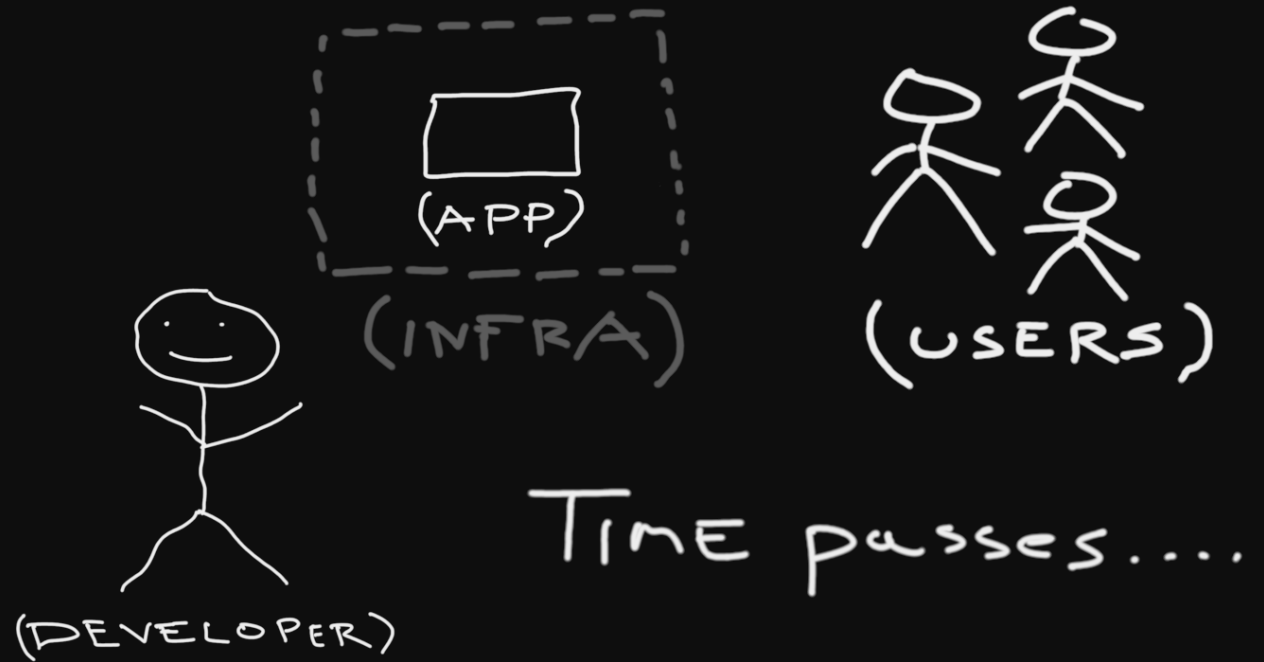


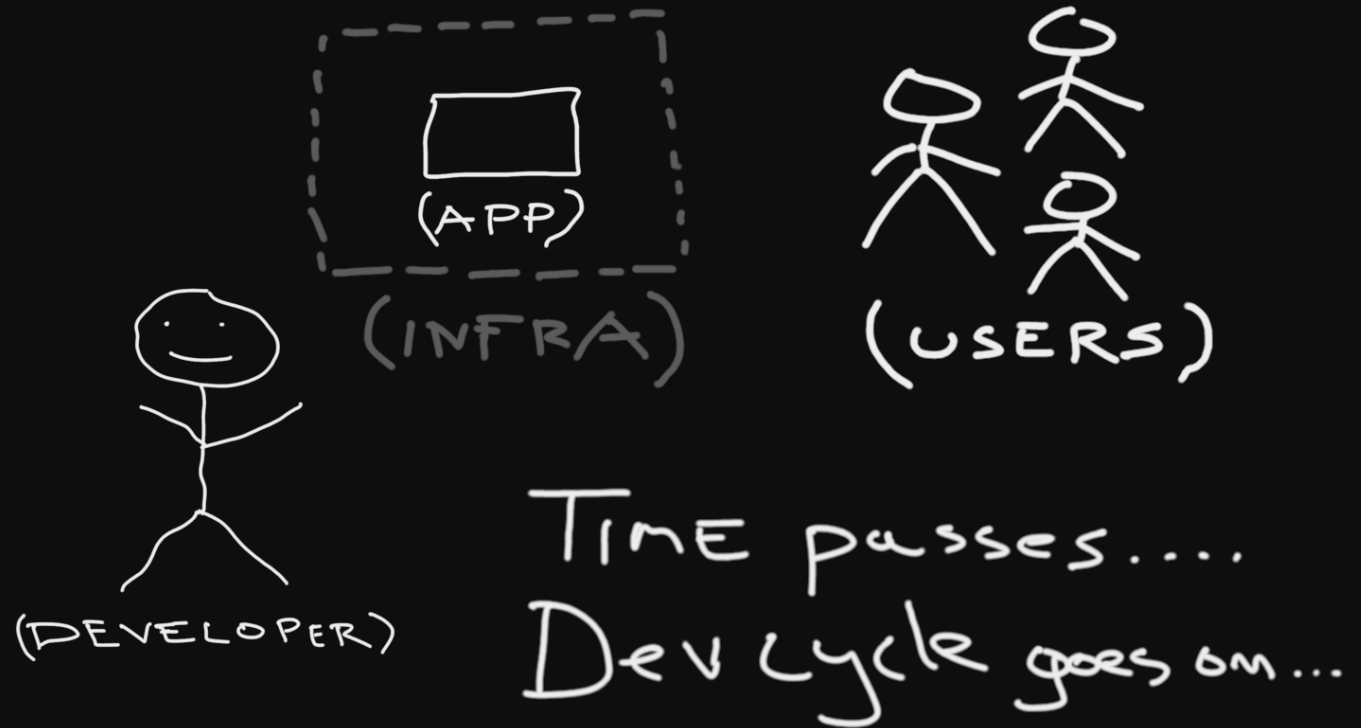




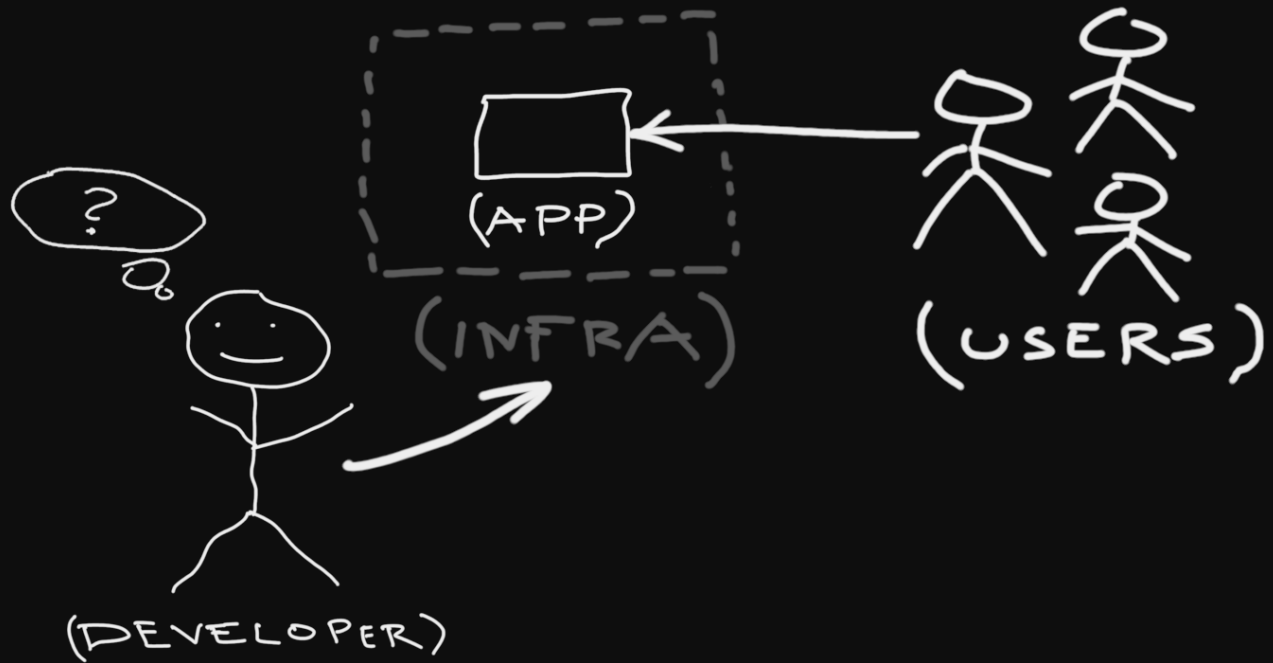


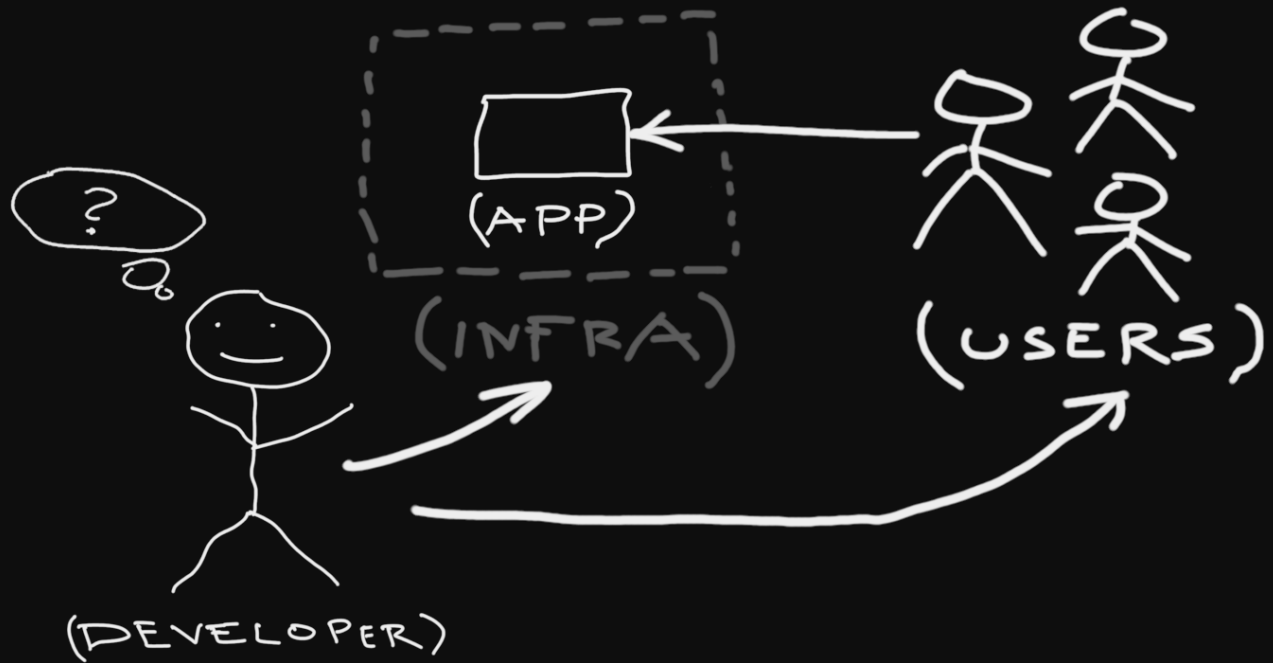


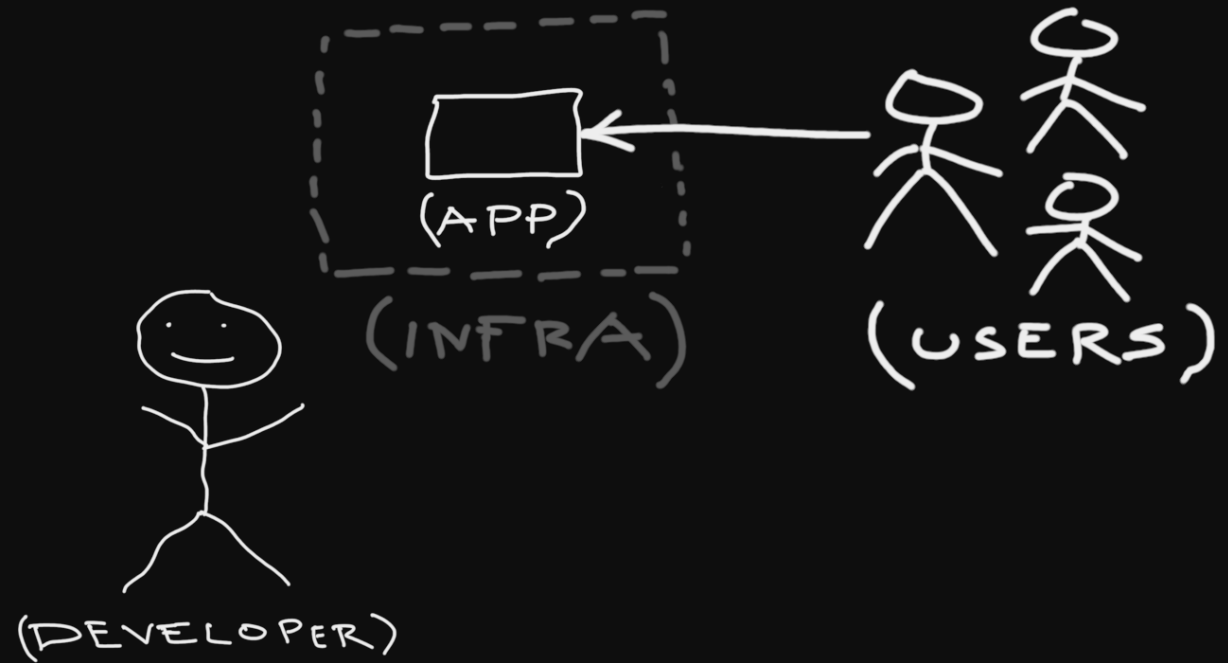


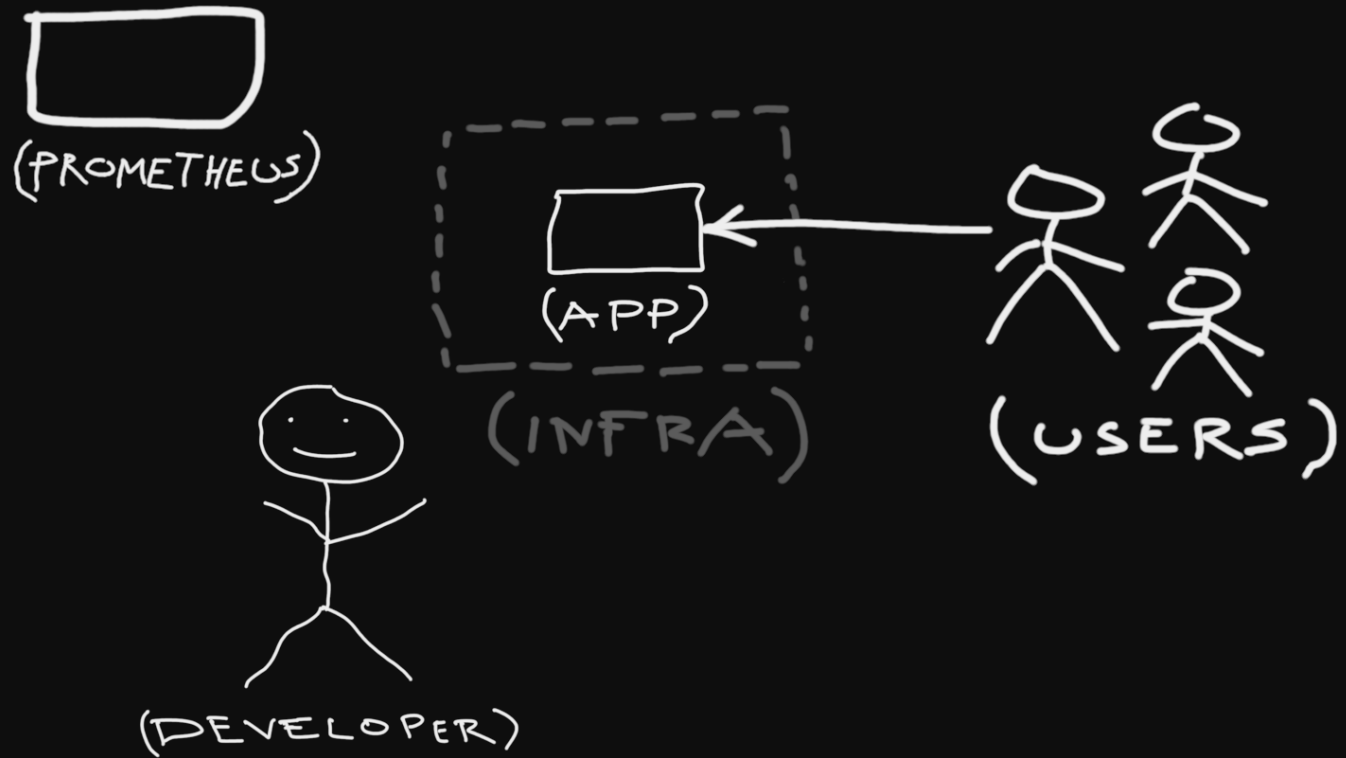


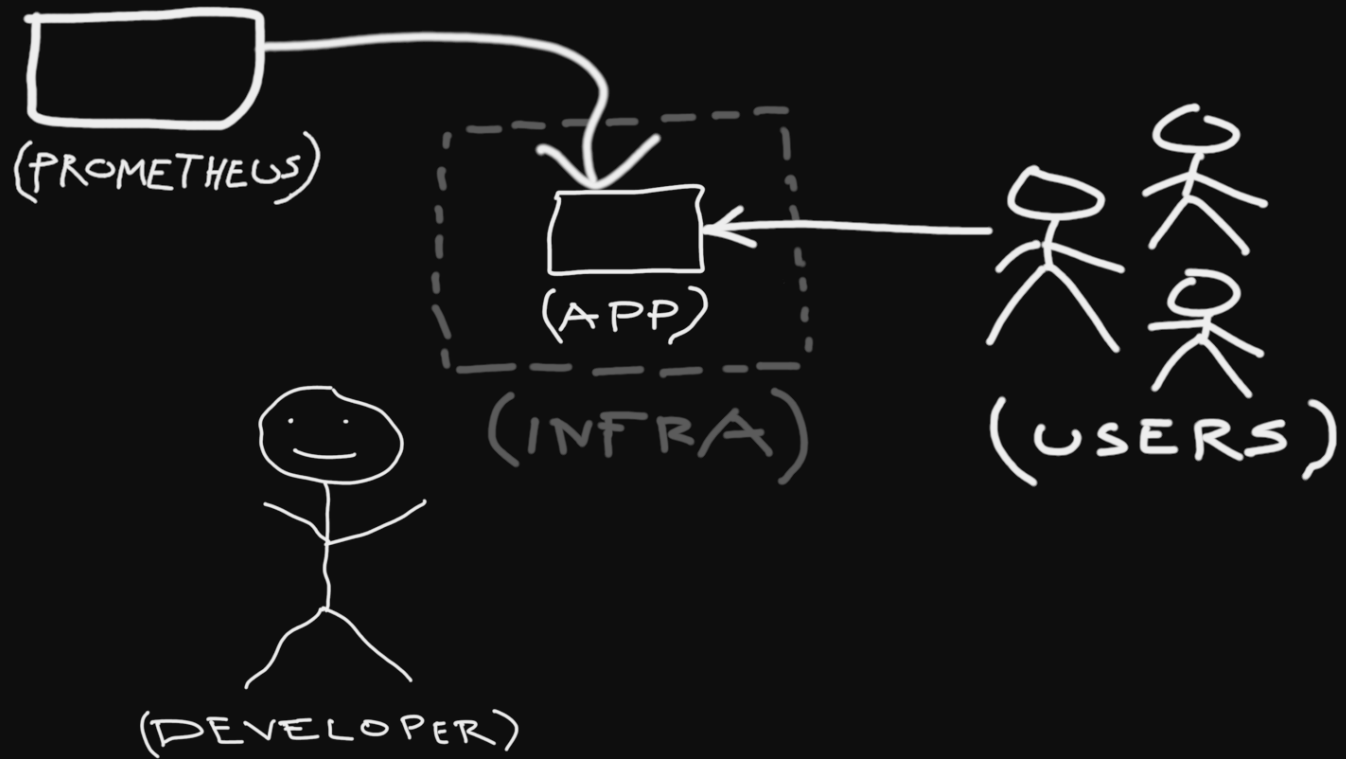


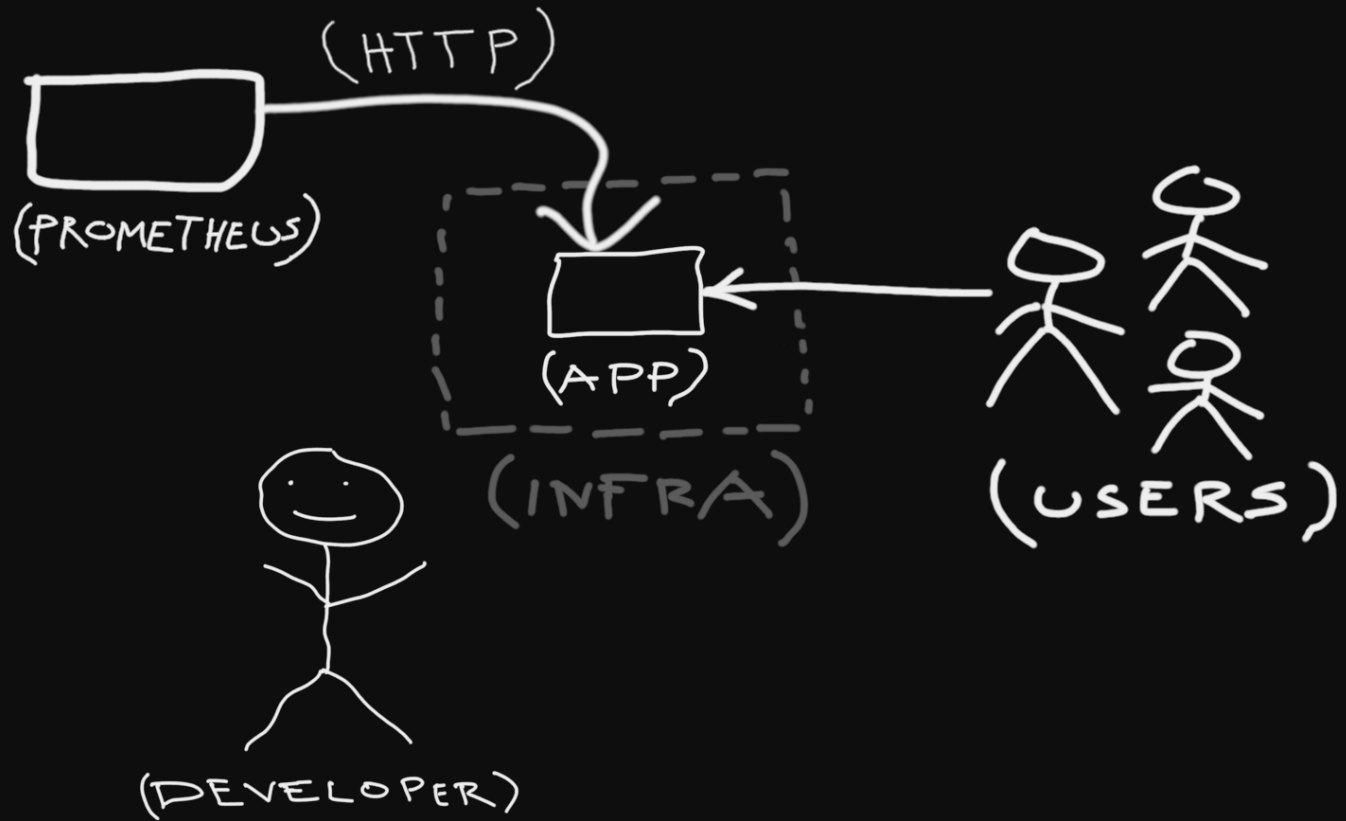


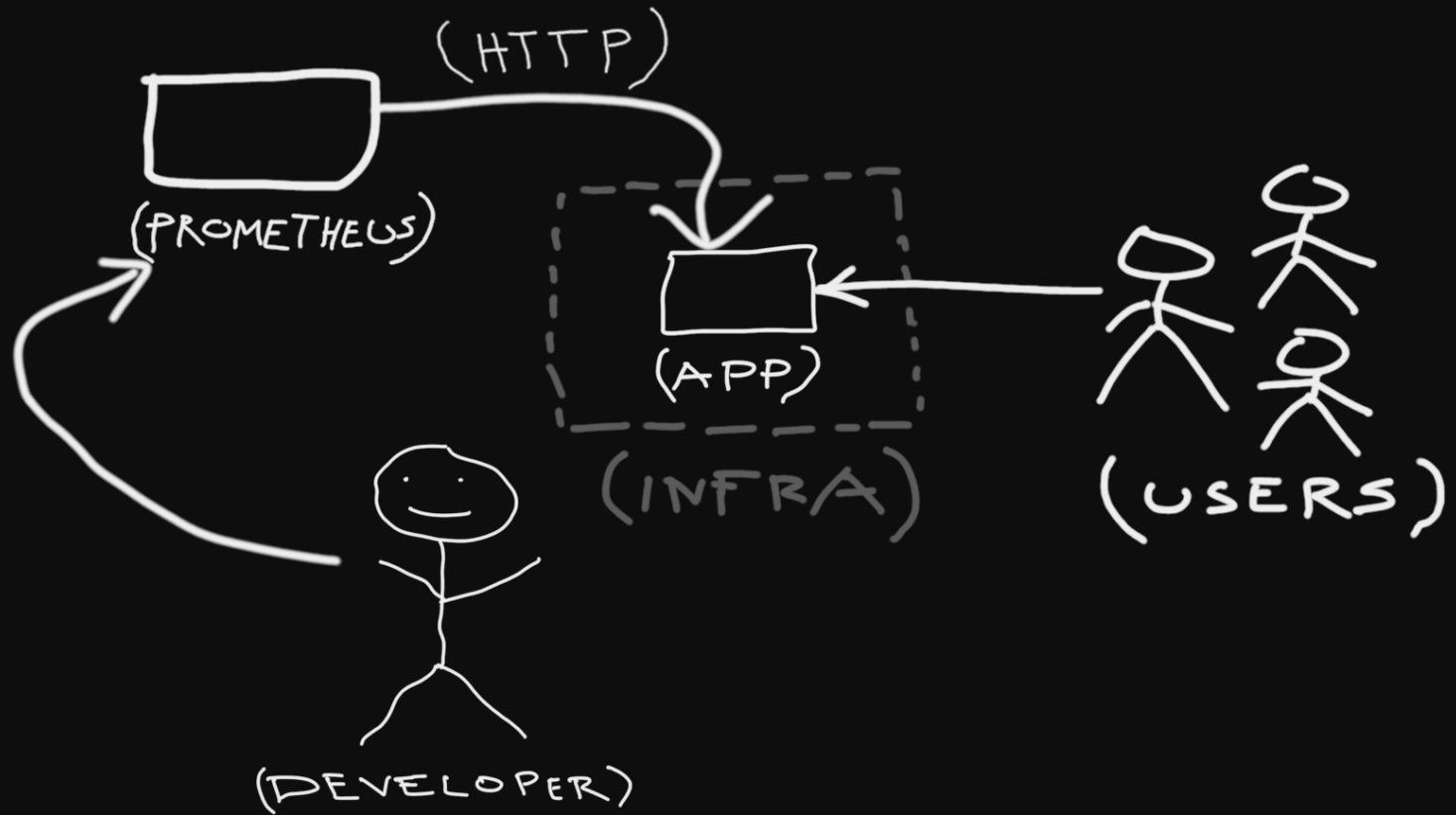


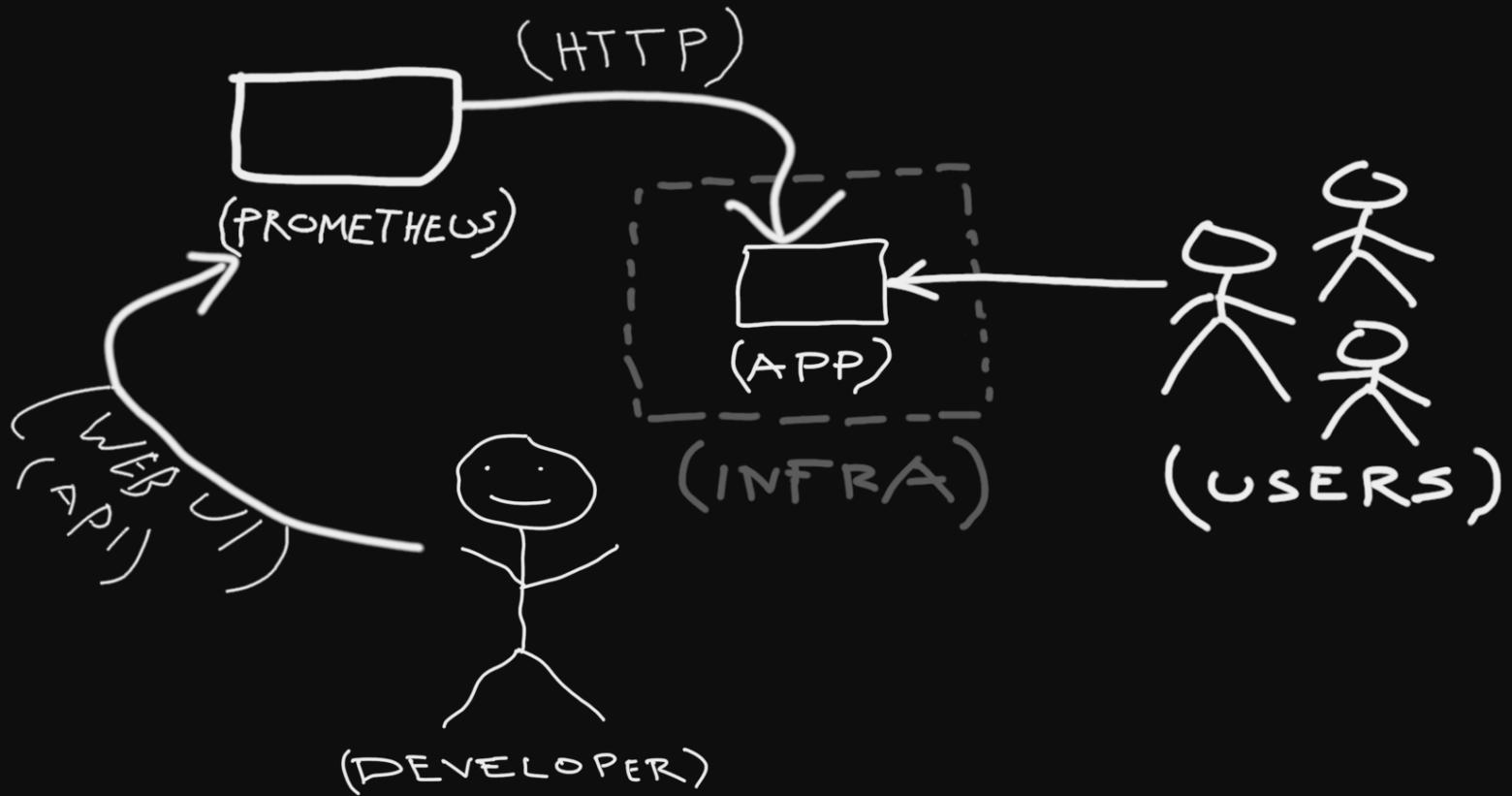










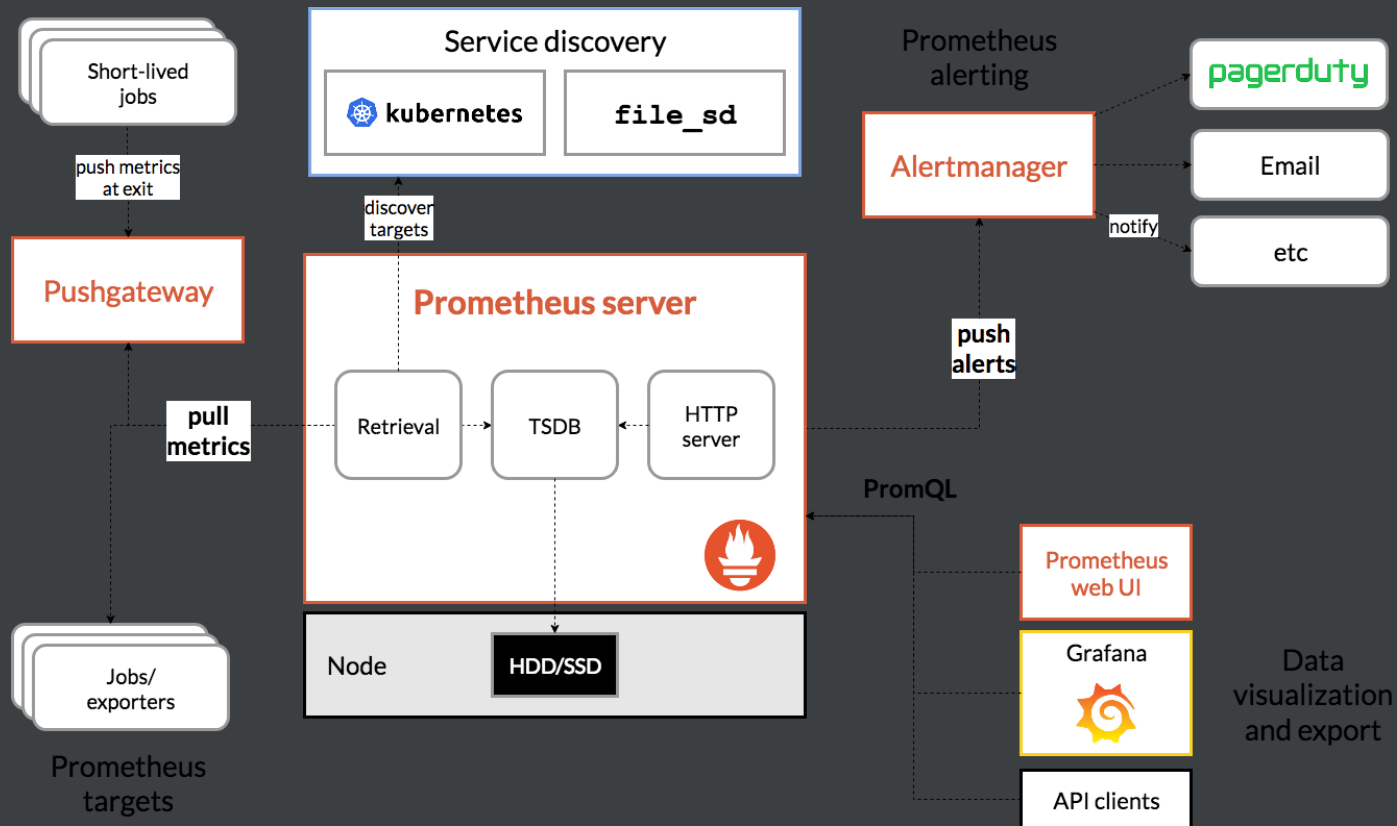


Prometheus scraps (gathers) metrics exposed via HTTP endpoint of your app. Scraping is done in regular intervals of time.

App is expected to have some long-lived state that represents metrics gathered since start of the app.

Prometheus

- Soundcloud
- 2012
- free software
- TSDB
- Go
- Twitter Bootstrap



source:

<https://prometheus.io/docs/introduction/overview/>

Prometheus \longleftrightarrow App communication





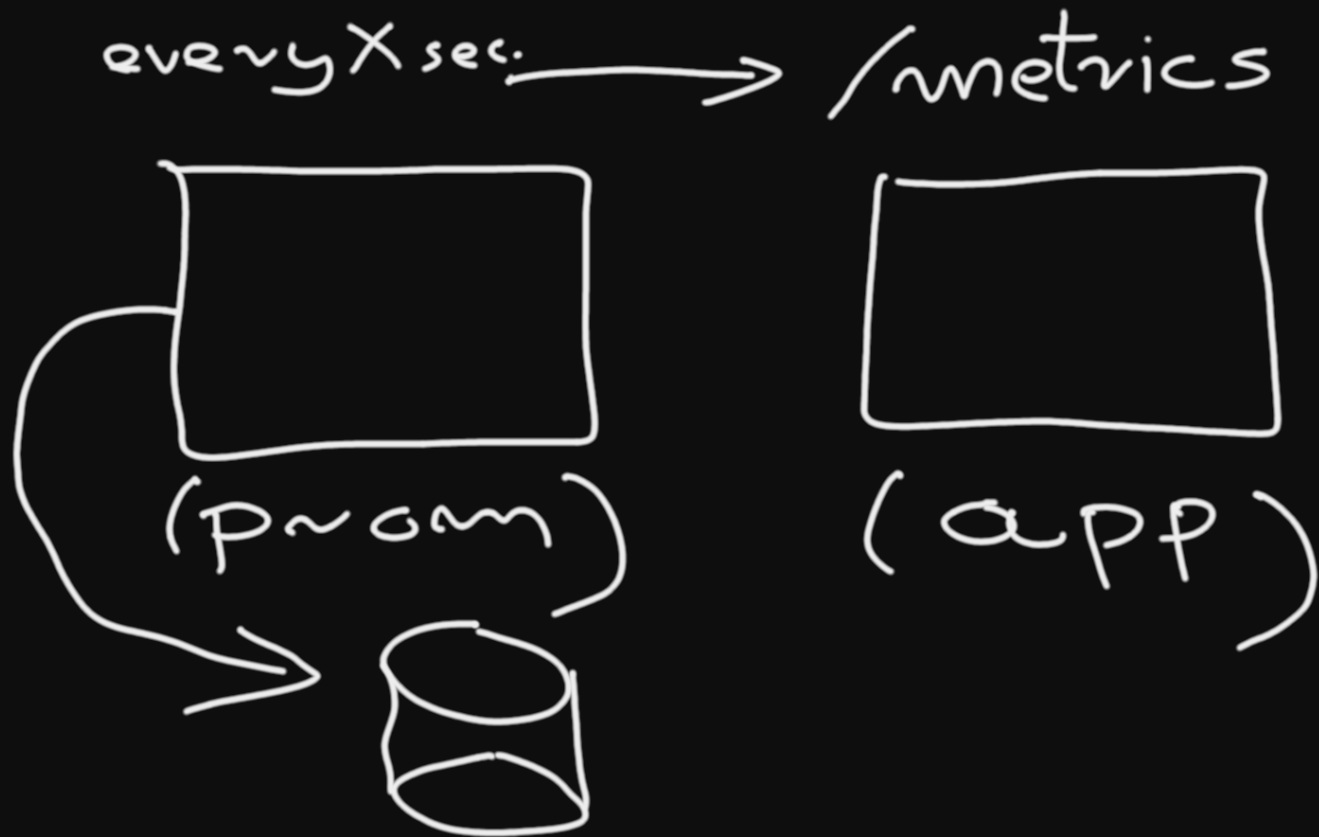
every X sec. \rightarrow /metrics

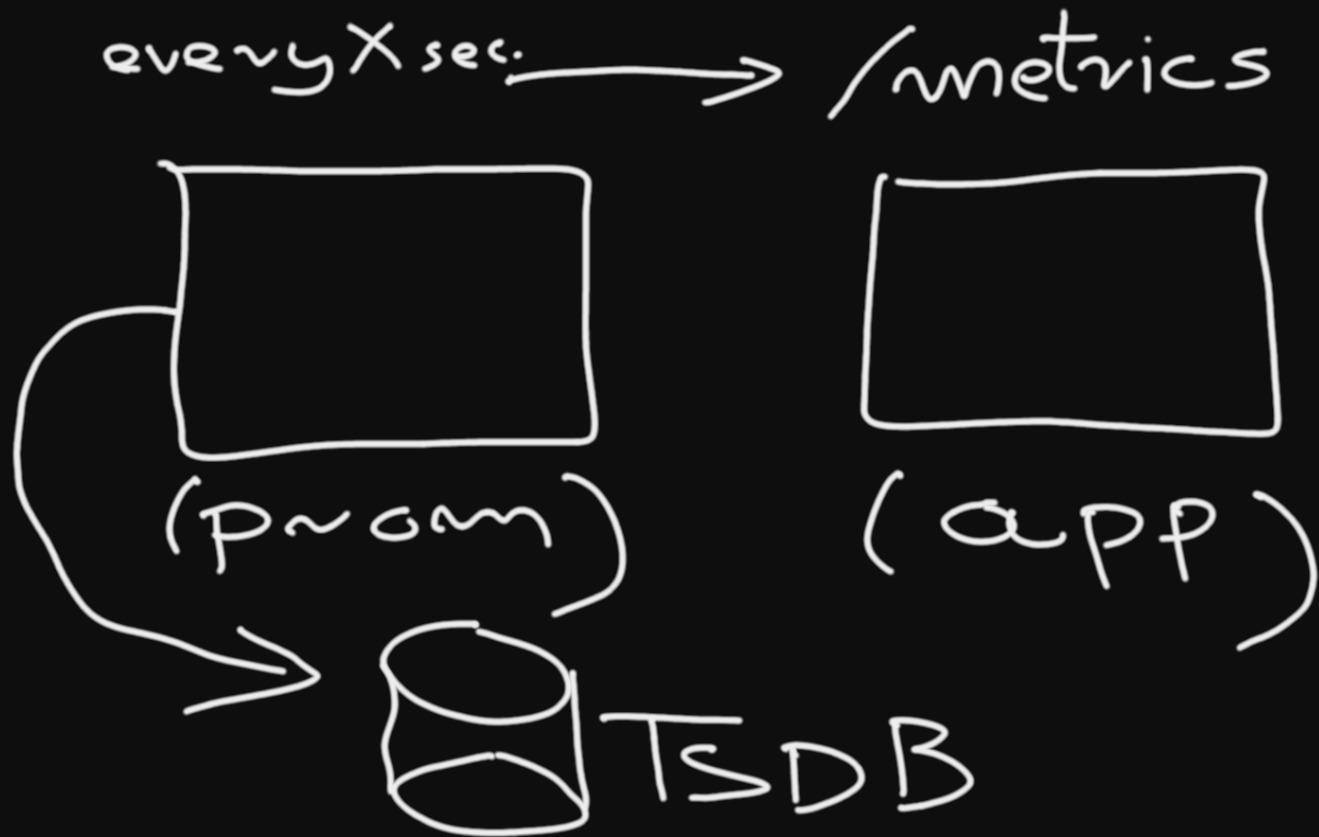


(prog)



(app)





/metrics

Metrics endpoint returns data in prometheus specific text format:

```
<metric-a-name>{<label name>=<label value>, ...} <value>  
<metric-b-name>{<label name>=<label value>, ...} <value>  
<metric-c-name>_<postfix> <value>  
<metric-c-name>_<postfix> <value>
```

/metrics

Metrics are just numeric values - numbers.

/metrics

Metrics are just numeric values - numbers.

Those numbers:

/metrics

Metrics are just numeric values - numbers.

Those numbers:

- can be categorized by set of label=value pairs

/metrics

Metrics are just numeric values - numbers.

Those numbers:

- can be categorized by set of label=value pairs
- can be accumulated into buckets (groups)

/metrics

Metrics are just numeric values - numbers.

Those numbers:

- can be categorized by set of label=value pairs
- can be accumulated into buckets
- are expected to change over time

/metrics

Metrics are just numeric values - numbers.

Those numbers:

- can be categorized by set of label=value pairs
- can be accumulated into buckets
- are expected to change over time
- can be analysed using PQL (PromQL)

/metrics

We have following types of metrics:

- Counter

ex: how many times given event occurred?

- Gauge

ex: how long is the background queue currently?

- Histogram

ex: what is distribution of request durations?

- Summary

ex: no idea how it differs from histogram

/metrics

We have following types of metrics:

- Counter

```
$counter = ... // setup for counter  
$counter->incBy(1);  
// or...  
$counter->incBy(1, ['example-label-value', ...]);
```

/metrics

We have following types of metrics:

- Gauge

```
$gauge = ... // setup for gauge  
$gauge->set(10)
```

/metrics

We have following types of metrics:

- Histogram

```
$histogram = ... // setup for histogram
$histogram->observe(10);
// or...
$histogram->observe(1, ['example-label-value', ...]);
```

introducing new
metric into app

introducing new metric into app

1. Think about: name, type, and labels for the metric

introducing new metric into app

1. Think about: name, type, and labels for the metric
2. Modify the app to add new metric

introducing new metric into app

1. Think about: name, type, and labels for the metric
2. Modify the app to add new metric:
 - declare the metric in the app using Prometheus library for your language

introducing new metric into app

1. Think about: name, type, and labels for the metric
2. Modify the app to add new metric:
 - declare the metric in the app using Prometheus library for your language
 - make sure metric is captured at appropriate places in the app

introducing new metric into app

1. Think about: name, type, and labels for the metric
2. Modify the app to add new metric:
 - declare the metric in the app using Prometheus library for your language
 - make sure metric is captured at appropriate places in the app
3. Deploy the app

introducing new metric into app

1. Think about: name, type, and labels for the metric
2. Modify the app to add new metric:
 - declare the metric in the app using Prometheus library for your language
 - make sure metric is captured at appropriate places in the app
3. Deploy the app
4. Query and visualise data in our tool of choice

querying metrics data

- https://prometheus.io/docs/tutorials/understanding_metric_types/
- <https://prometheus.io/docs/prometheus/latest/querying/examples/>

demo time (?)

