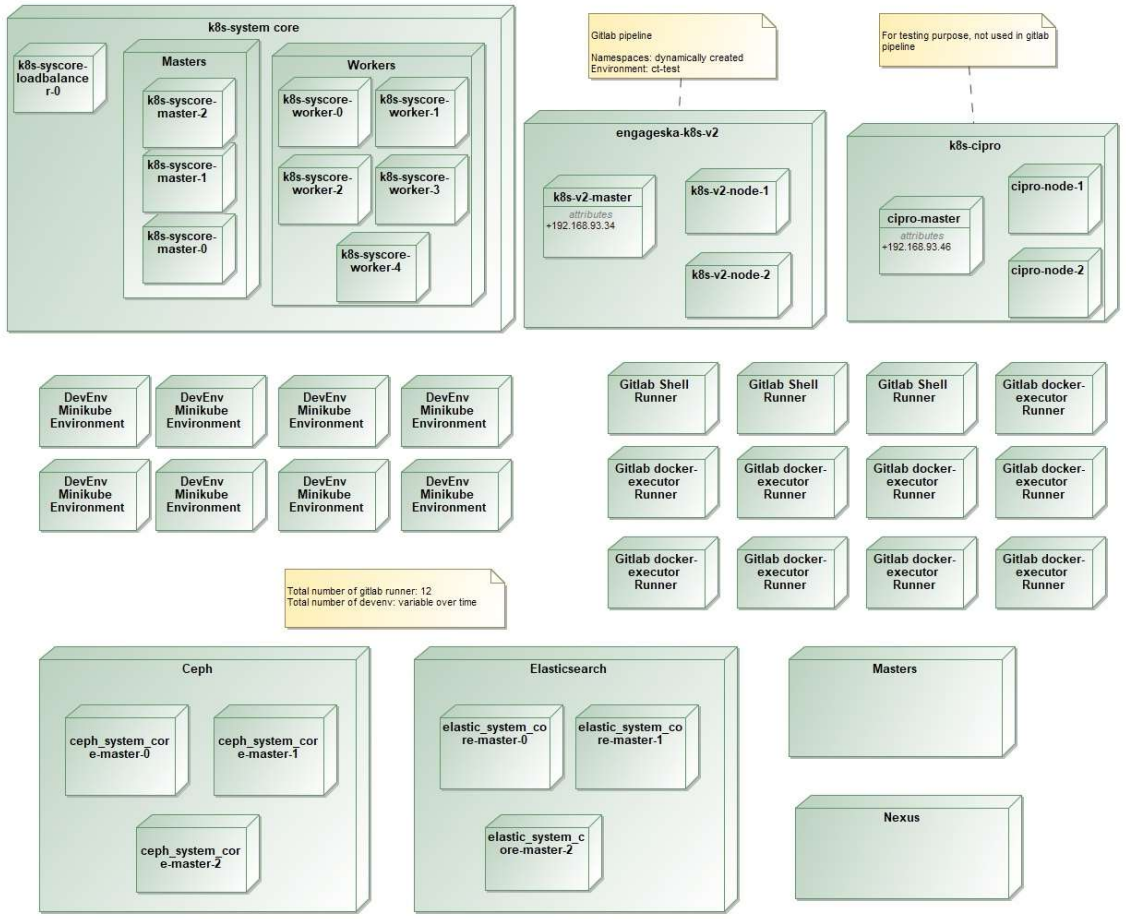
# SKA Web interfaces

for infrastructure/testing environment

Matteo Di Carlo

# SKA Infrastructure

- Openstack virtualization



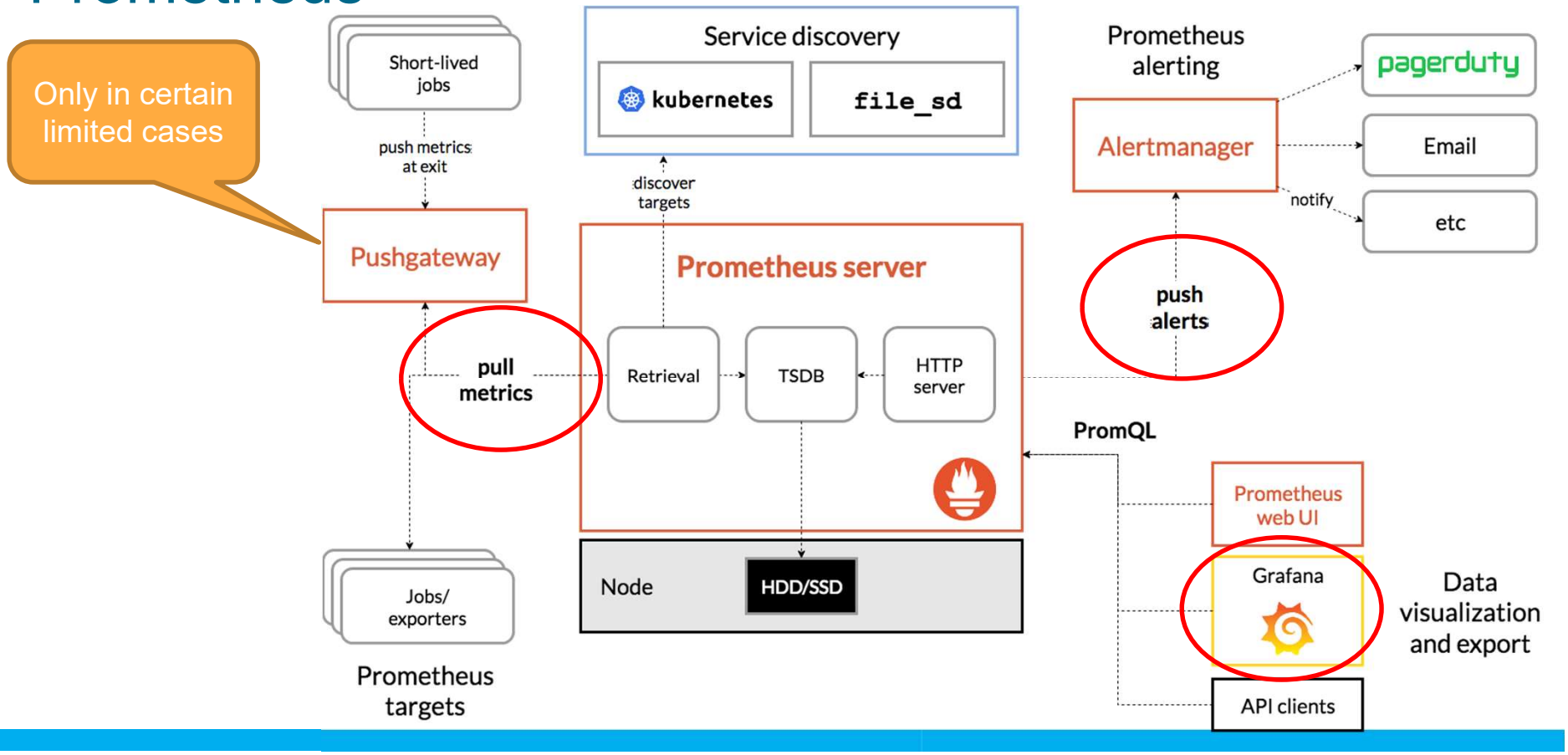
# Prometheus

Client (Prometheus) - server (targets/exporters) application

All data are *time series*

Modifiability === easily add *new exporters* and alerts

# Prometheus


Only in certain limited cases
Short-lived jobs
push metrics at exit
Pushgateway
Service discovery
kubernetes
file_sd
discover targets
Prometheus server
pull metrics
Retrieval
TSDB
HTTP server
Node
HDD/SSD
Jobs/ exporters
Prometheus targets
Prometheus alerting
Alertmanager
pagerduty
Email
etc
notify
push alerts
PromQL
Prometheus web UI
Grafana
API clients
Data visualization and export

# Prometheus exporter - metrics

Every time series is uniquely identified by:

- its metric name and
- optional key-value pairs called

```
<metric name>{<label name>=<label value>, ..
```

*It must have all the information rela*

4 types of metrics:

- Counter: cumulative metric
- Gauge: single numerical value that can arbitrarily go up and down
- Histogram: samples observations (usually things like request durations or response sizes) and counts them in configurable buckets
- Summary: samples observations (similar to a *histogram*)

# Prometheus - alerts

```
prometheus_alert_rules:
- alert: CriticalCPULoad
  expr: 'instance:node_cpu:load > 98'
  for: 30m
  labels:
    severity: critical
  annotations:
    description: "{% raw %}{{ $labels.instance }} of job {{ $labels.job }} ha
s Critical CPU load for more than 30 minutes.{% endraw %}"
    summary: "{% raw %}Instance {{ $labels.instance }} -
 Critical CPU load{% endraw %}"
```

# Grafana

Engine for displaying data on web coming from many data sources

- Working with grafana means to create dashboards which allows to understand something

# Grafana

Plugin architecture where a plugin can be:

- A panel
- A data source
- An app (combination of data source and panels for a specific purpose)

# Demo 1: infrastructure and testing