KubeCon | CloudNativeCon

Europe 2023

KubeCon | CloudNativeCon

Europe 2023

Kubernetes Defensive Monitoring with Prometheus

David de Torres Huerta & Mirco De Zorzi (Sysdig)

KubeCon | CloudNativeCon

Europe 2023

David de Torres

Eng. Manager at Sysdig

Maintainer of PromCat.io

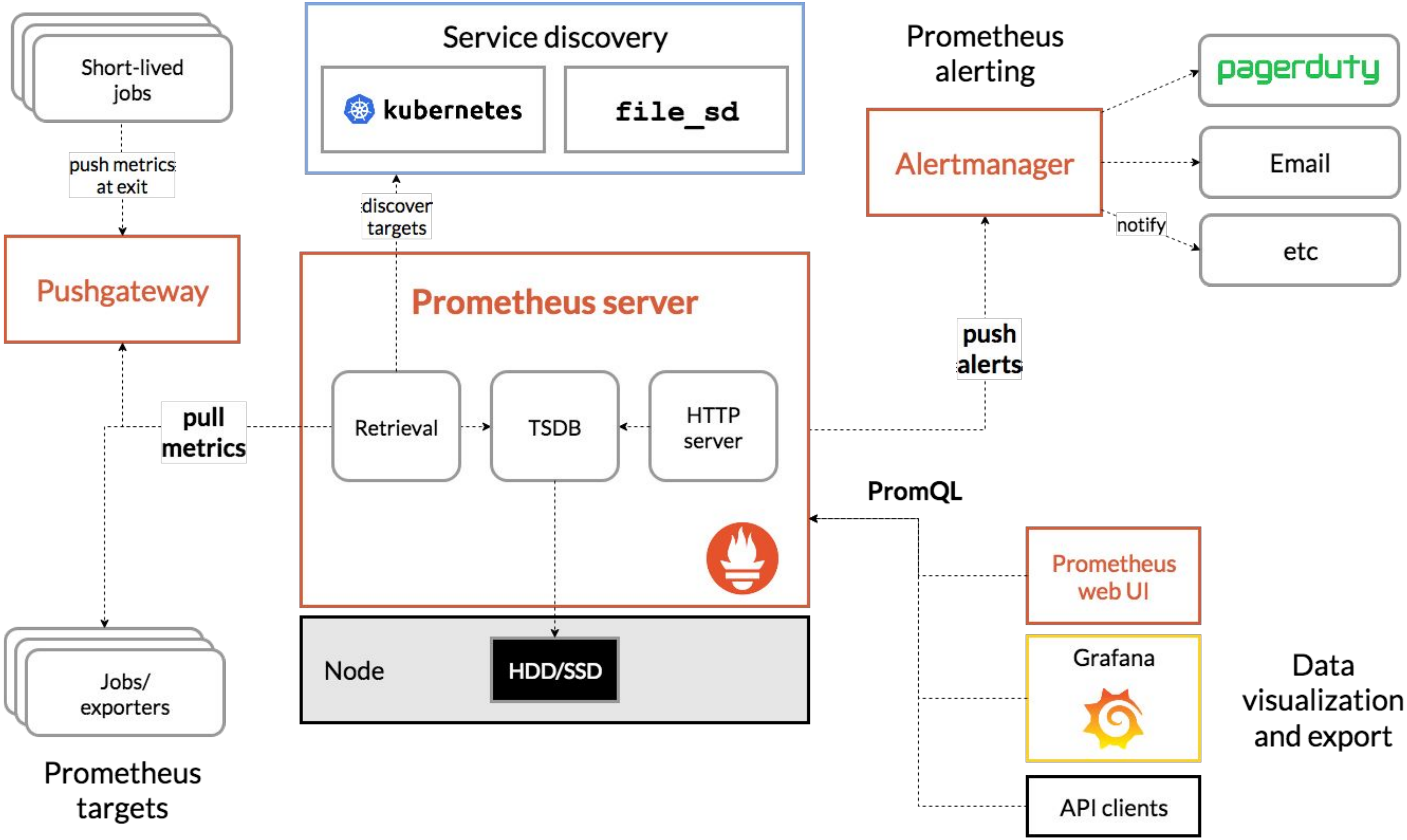
@maellyssa@mastodon.social

Mirco De Zorzi

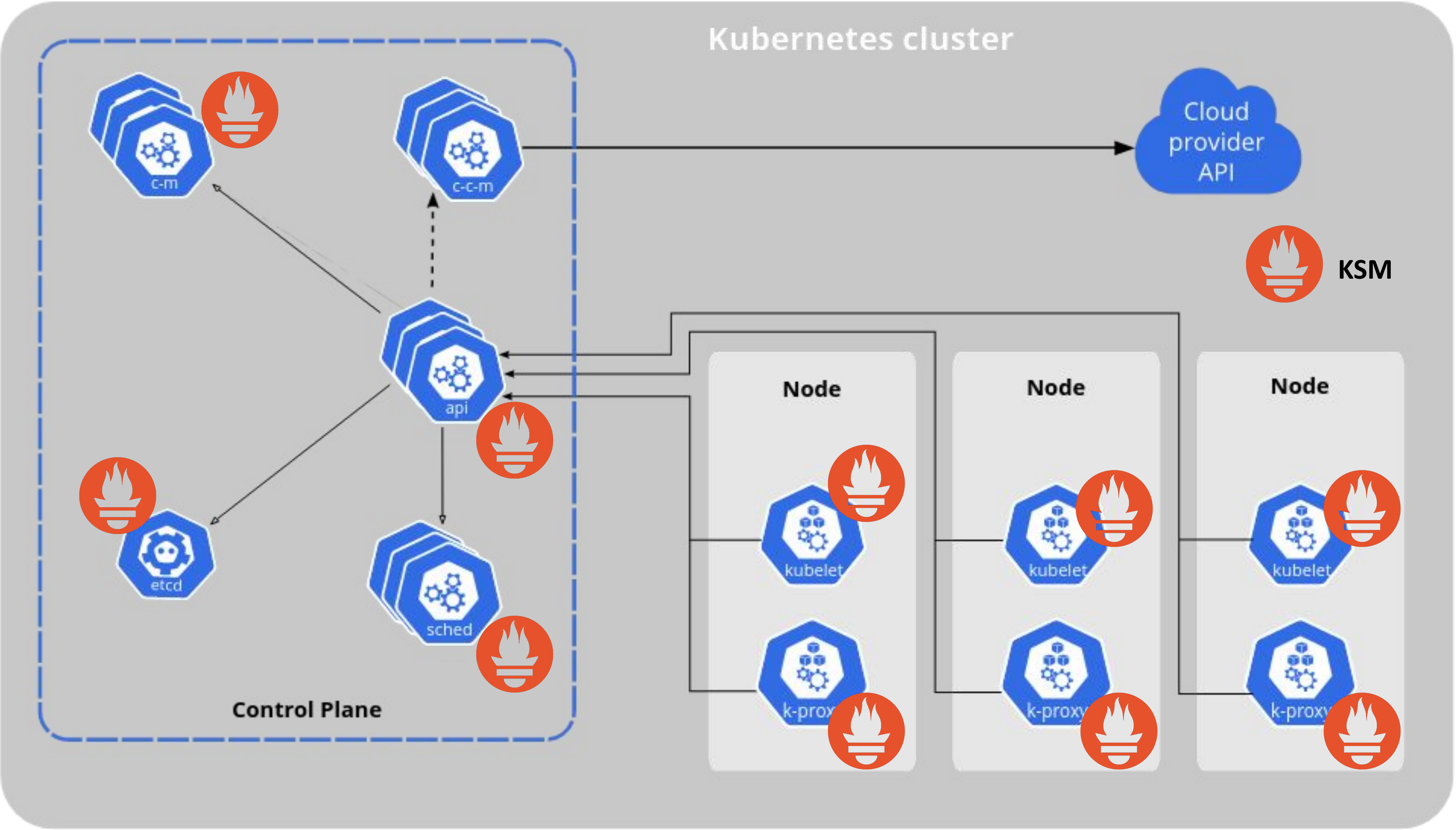
Engineer at Sysdig

@mircodezorzi

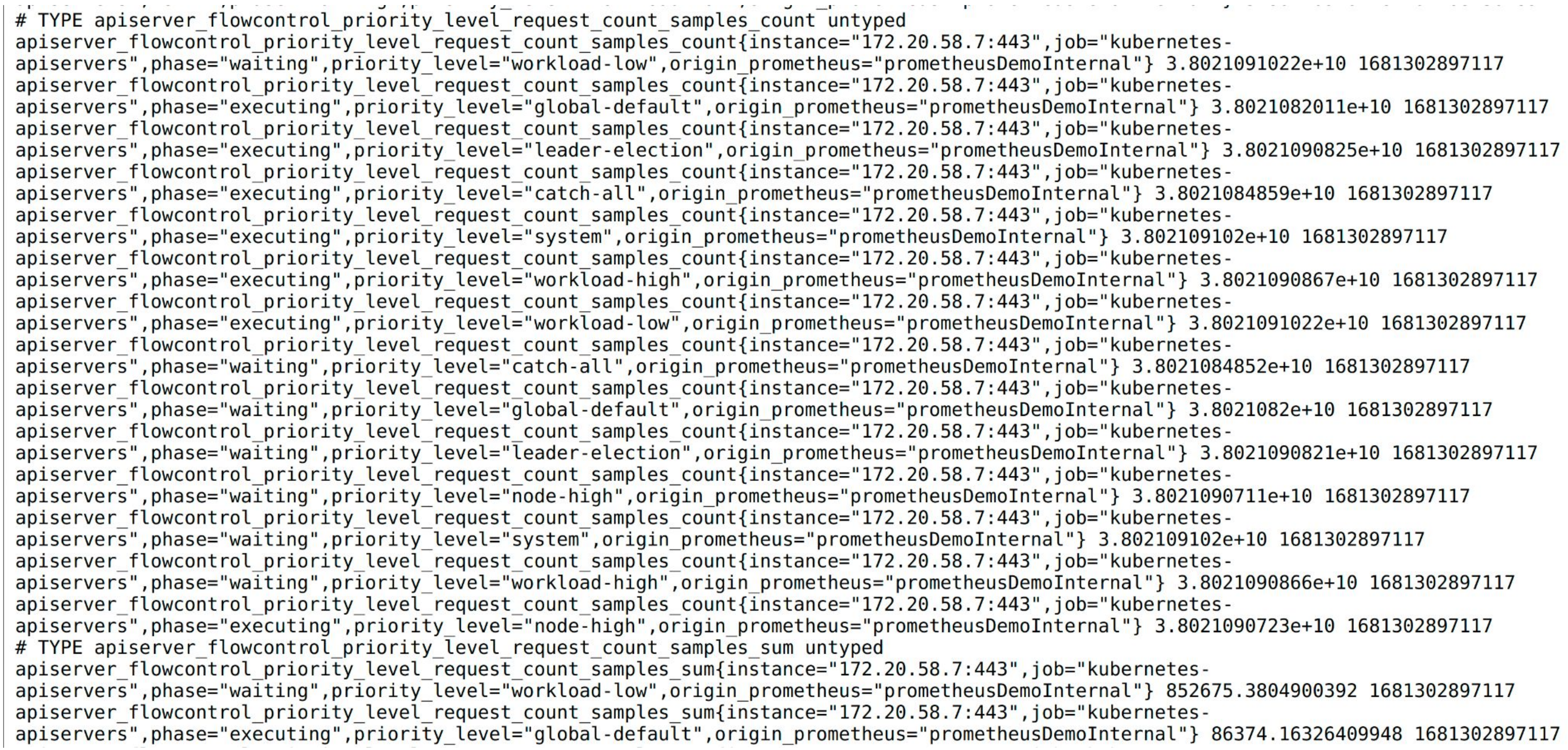
What is Prometheus



What is Prometheus

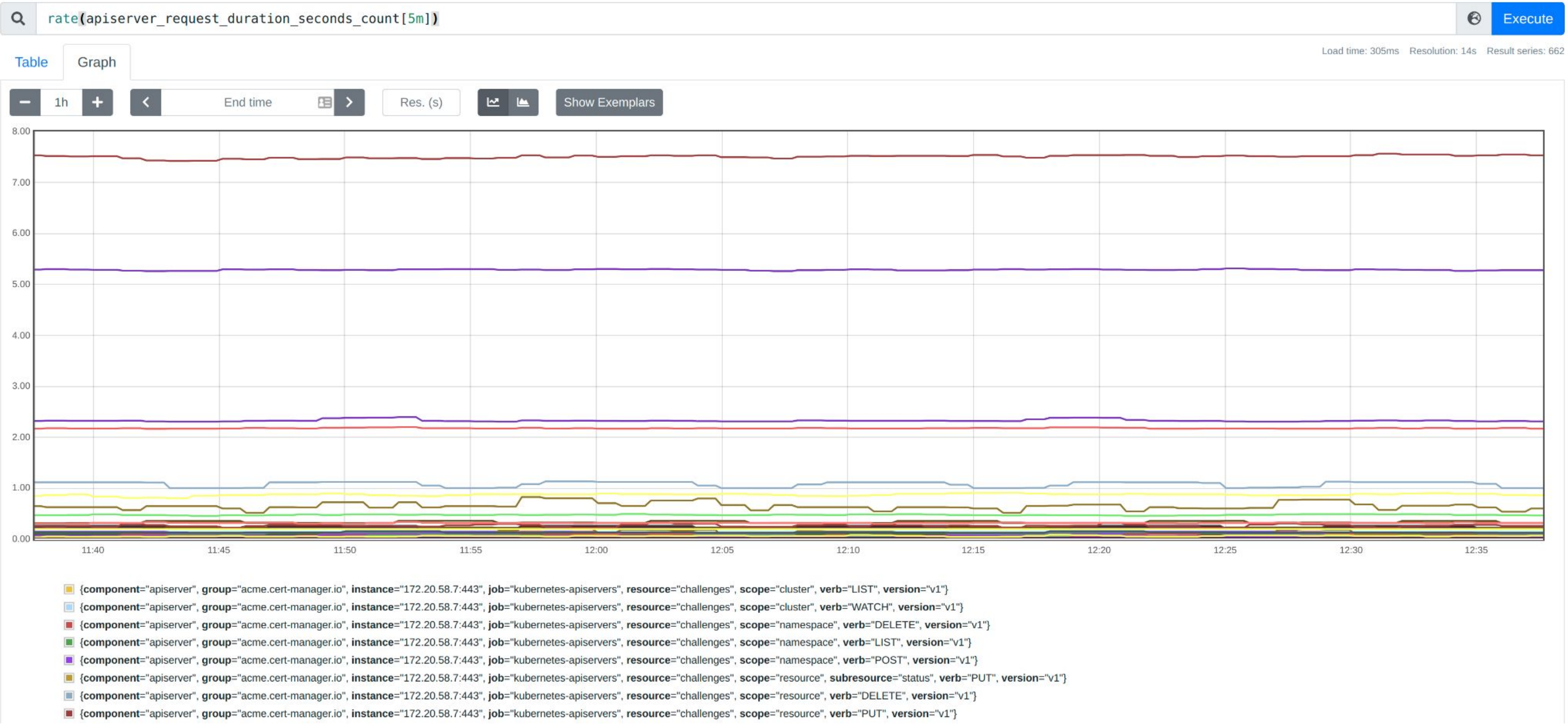


What is Prometheus



```
# TYPE apiserver_flowcontrol_priority_level_request_count_samples_count untyped
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="workload-low",origin_prometheus="prometheusDemoInternal"} 3.8021091022e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="global-default",origin_prometheus="prometheusDemoInternal"} 3.8021082011e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="leader-election",origin_prometheus="prometheusDemoInternal"} 3.8021090825e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="catch-all",origin_prometheus="prometheusDemoInternal"} 3.8021084859e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="system",origin_prometheus="prometheusDemoInternal"} 3.802109102e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="workload-high",origin_prometheus="prometheusDemoInternal"} 3.8021090867e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="workload-low",origin_prometheus="prometheusDemoInternal"} 3.8021091022e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="catch-all",origin_prometheus="prometheusDemoInternal"} 3.8021084852e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="global-default",origin_prometheus="prometheusDemoInternal"} 3.8021082e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="leader-election",origin_prometheus="prometheusDemoInternal"} 3.8021090821e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="node-high",origin_prometheus="prometheusDemoInternal"} 3.8021090711e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="system",origin_prometheus="prometheusDemoInternal"} 3.802109102e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="workload-high",origin_prometheus="prometheusDemoInternal"} 3.8021090866e+10 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_count{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="node-high",origin_prometheus="prometheusDemoInternal"} 3.8021090723e+10 1681302897117
# TYPE apiserver_flowcontrol_priority_level_request_count_samples_sum untyped
apiserver_flowcontrol_priority_level_request_count_samples_sum{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="waiting",priority_level="workload-low",origin_prometheus="prometheusDemoInternal"} 852675.3804900392 1681302897117
apiserver_flowcontrol_priority_level_request_count_samples_sum{instance="172.20.58.7:443",job="kubernetes-
apiservers",phase="executing",priority_level="global-default",origin_prometheus="prometheusDemoInternal"} 86374.1632640948 1681302897117
```

What is Prometheus



```
rate(apiserver_request_duration_seconds_count[5m])
```

Load time: 305ms Resolution: 14s Result series: 662

🟨 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="cluster", **verb**="LIST", **version**="v1"}

🟦 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="cluster", **verb**="WATCH", **version**="v1"}

🟥 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="namespace", **verb**="DELETE", **version**="v1"}

🟩 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="namespace", **verb**="LIST", **version**="v1"}

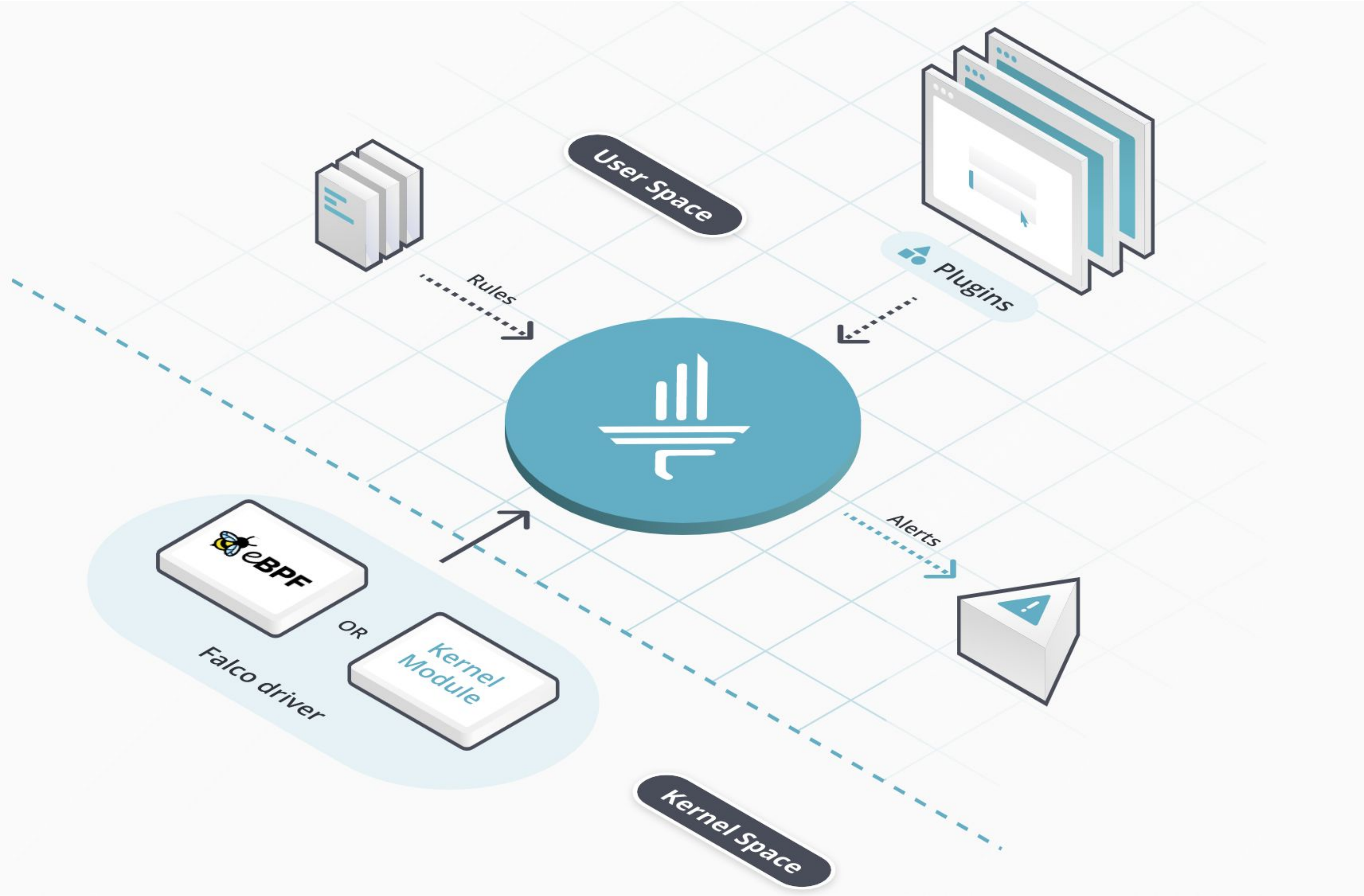
🟪 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="namespace", **verb**="POST", **version**="v1"}

🟫 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="resource", **subresource**="status", **verb**="PUT", **version**="v1"}

🟦 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="resource", **verb**="DELETE", **version**="v1"}

🟥 {**component**="apiserver", **group**="acme.cert-manager.io", **instance**="172.20.58.7:443", **job**="kubernetes-apiservers", **resource**="challenges", **scope**="resource", **verb**="PUT", **version**="v1"}

What is Falco



Preface

Preface

When the only tool you have is hammer, everything looks like a nail.

Preface

When the only tool you have is ~~hammer~~, everything looks like a ~~nail~~.

Prometheus *metrics*

Why metrics?

Capability	Metrics-based Monitoring	Runtime Security (Falco)	Image Scanning
Detect unusual behavior	✅	✅	❌
Monitor resource usage	✅	⚠️ Limited	❌
Alert on specific events	✅	✅	✅
Anomaly detection	✅	⚠️ Limited (Rule-based)	❌
Detect insecure configs	❌	✅	✅
Monitor network activity	⚠️ Limited	✅	❌
Detect vulnerable packages	⚠️ Limited (Labels)	❌	✅
Container visibility	⚠️ Limited (Metrics)	✅ (System calls)	❌
Real-time detection	✅	✅	❌ (Pre-deployment)
Incident investigation	⚠️ Limited (Metrics)	✅ (Detailed events)	⚠️ Limited (Scan results)
Historical data and past context	✅	⚠️ Limited	⚠️ Limited
Detect insider threats	⚠️ Limited	✅	⚠️ Limited
Detect application-level attacks	✅ (Custom metrics & alerting)	⚠️ Limited	❌

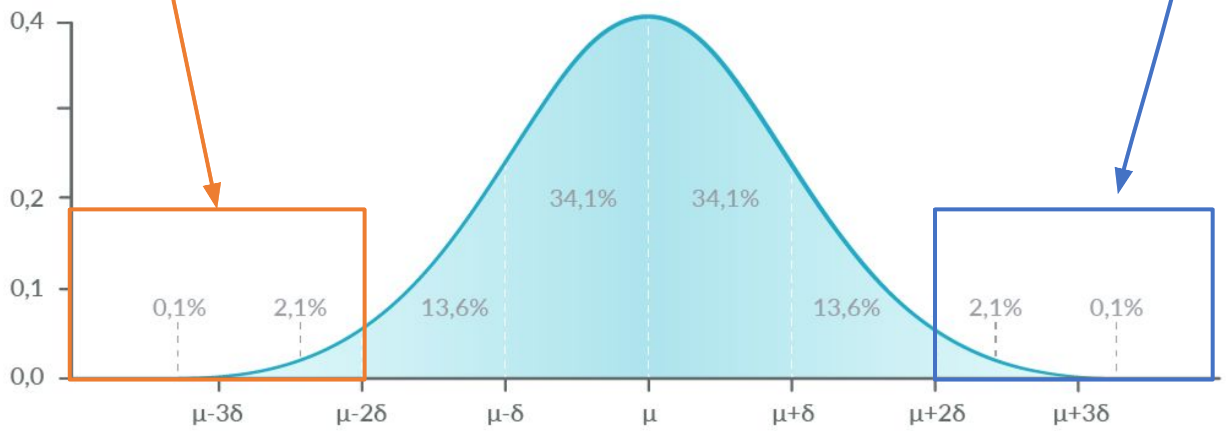
Quick introduction to Anomaly detection with Prometheus

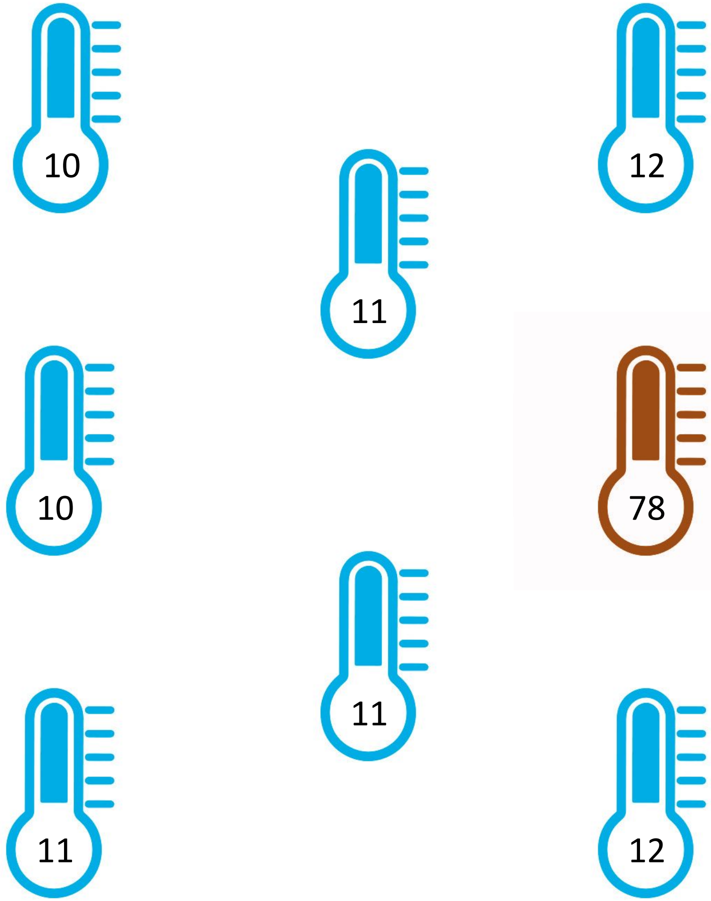
Anomaly detection

Group anomaly:

Detect 5% anomalies (top and bottom):

```
temp > avg (temp) + 2 * stddev(temp)
   OR temp < avg (temp) - 2 * stddev(temp)
```



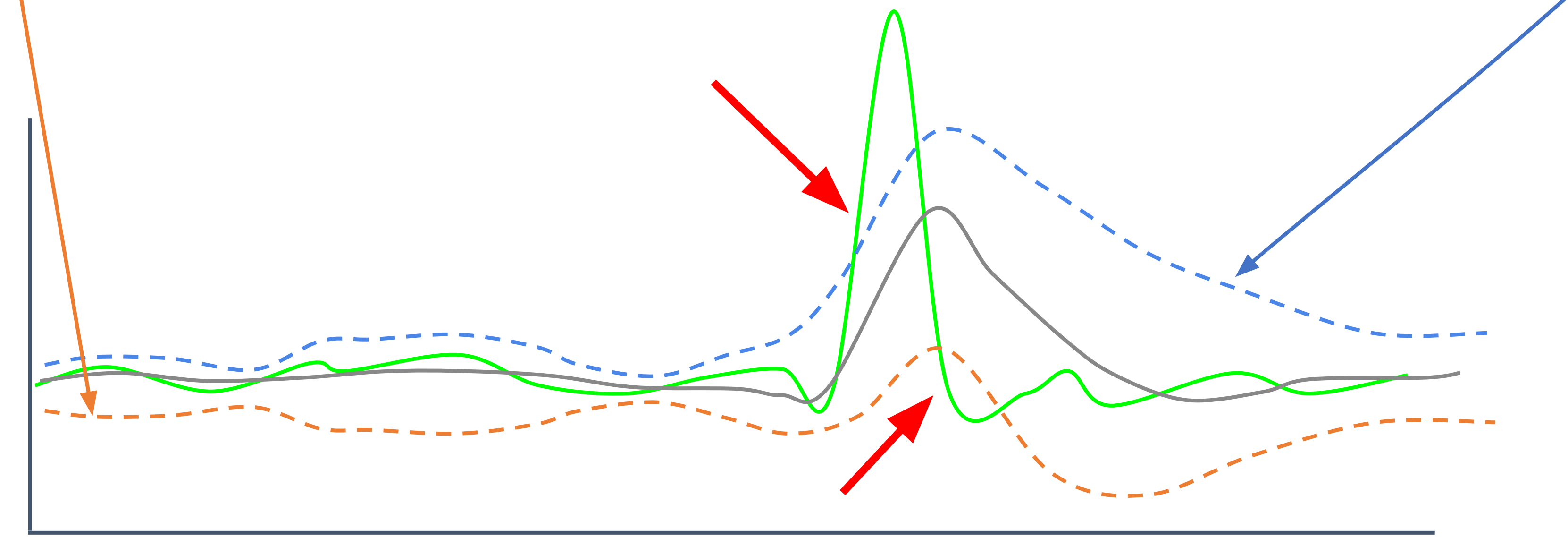


Anomaly detection

Simple time anomalies:

Detect samples that are "different" than the values in the last minutes (or hours, days…):

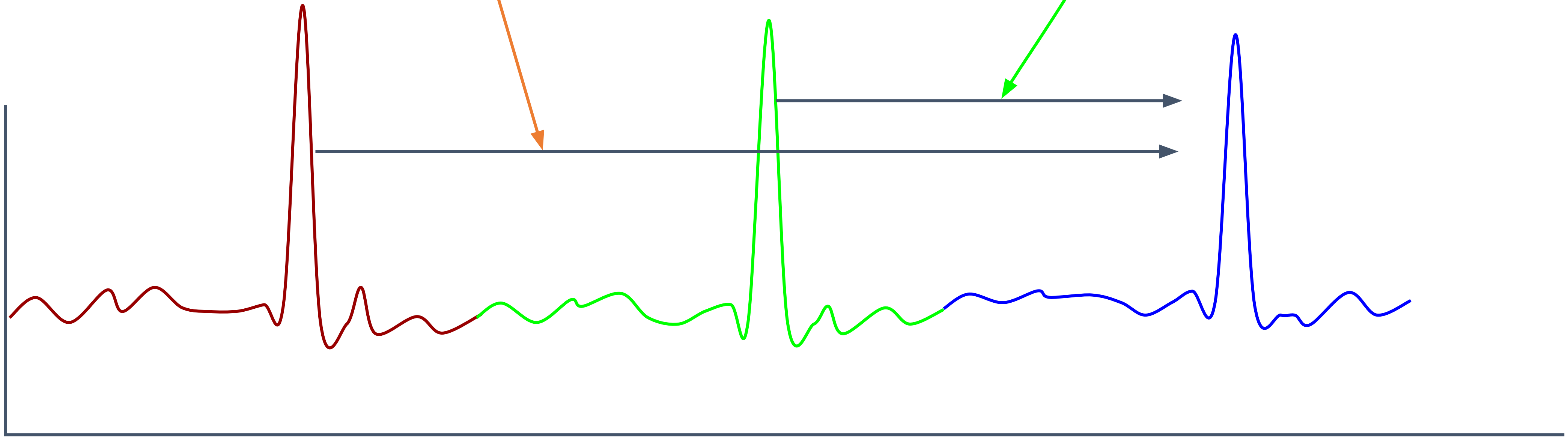
```
temp > avg_over_time (temp[5m]) + 2 * stddev_over_time(temp[5m])
  OR temp < avg_over_time (temp[5m]) - 2 * stddev_over_time(temp[5m])
```



Anomaly detection

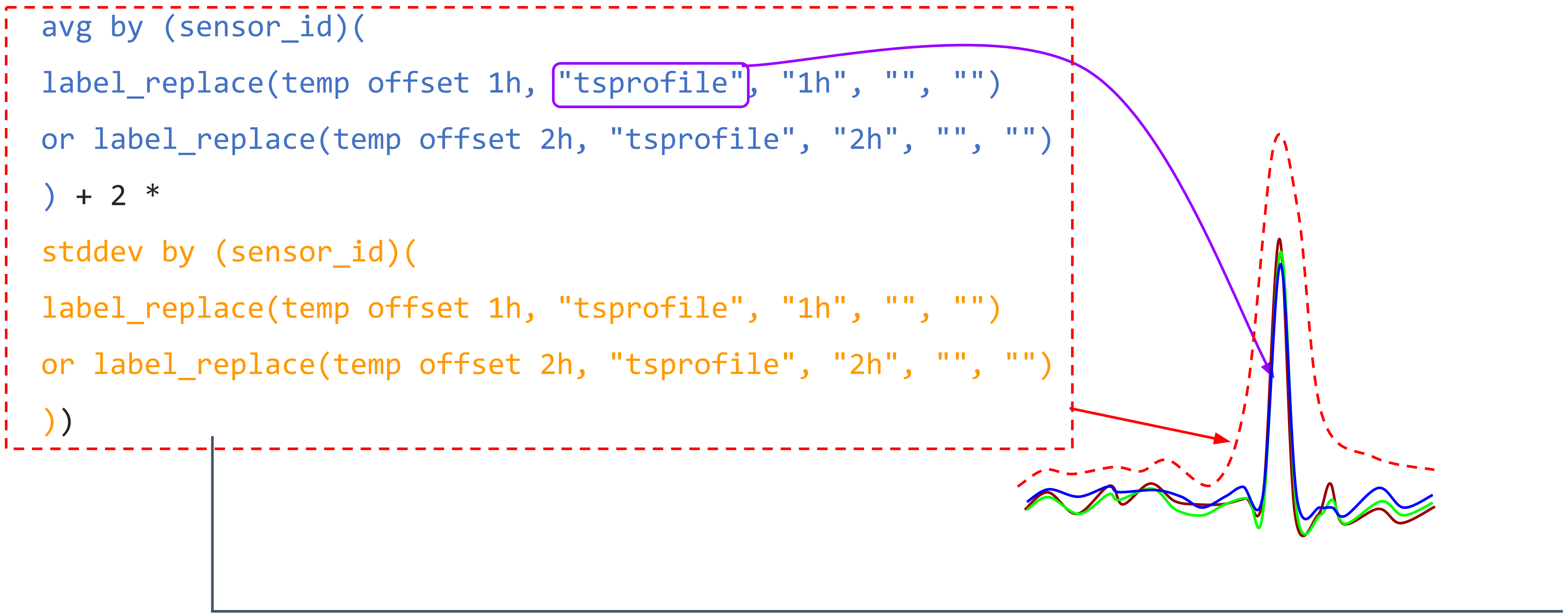
Seasonal time anomalies: We move the windows to the present with the `offset` modifier and create new auxiliary time series and adding new label `tsprofile` with `label_replace`

```
(
label_replace(temp offset 1h, "tsprofile", "1h", "", "")
or label_replace(temp offset 2h, "tsprofile", "2h", "", "")
)
```



Anomaly detection

Seasonal time anomalies: And when they overlap, we will treat them as a group anomaly



```
temp > (
avg by (sensor_id)(
label_replace(temp offset 1h, "tsprofile", "1h", "", "")
or label_replace(temp offset 2h, "tsprofile", "2h", "", "")
) + 2 *
stddev by (sensor_id)(
label_replace(temp offset 1h, "tsprofile", "1h", "", "")
or label_replace(temp offset 2h, "tsprofile", "2h", "", "")
))
```

Security threats with Prometheus

Security threats with Prometheus

Unauthorized attempts to access to API-Server:

Detect unauthorized attempts to access the API server:

```
sum by (client)(
    rate(apiserver_request_total{code=~"401|403"}[5m]))
```

Security threats with Prometheus

Unauthorized attempts to access to resources:

Unauthorized access attempts to sensitive resources:

```
sum by (username, resource) (
    rate(apiserver_request_total
        {resource=~"secrets|configmaps",
         code=~"401|403"}[5m]))
```

Cluster-wide changes (possible insider):

```
rate(apiserver_request_total
        {verb=~"CREATE|UPDATE|PATCH|DELETE",
         scope="cluster"}[5m])
```

Security threats with Prometheus

Creation of new ingress (possible back-door creation):

New ingress in the last 5 minutes:

```
count(kube_ingress_created)
- count(kube_ingress_created offset 5m)
> 0
```

Security threats with Prometheus

Certificates near to expire:

In Kubernetes elements:

```
histogram_quantile(0.01, sum by(job, instance, le)
   (rate(apiserver_client_certificate_expiration_seconds_bucket[5m])))
< 3600 * 24 * 7
```

In Apps:

```
certmanager_certificate_expiration_timestamp_seconds - time()
< 3600 * 24 * 7
```

Security threats with Prometheus

Fingerprinting attempts:

High 404 errors in ingress controller:

```
(sum by (host, path)
   (rate(nginx_ingress_controller_requests{status=~"404"}[5m]))

/ sum by (host, path)
   (rate(nginx_ingress_controller_requests{}[5m])))

> 0.3
```

Security threats with Prometheus

Force brute attacks:

High 403 errors in ingress controller:

```
(sum by (host, path)
   (rate(nginx_ingress_controller_requests{status=~"403"}[5m]))

/ sum by (host, path)
   (rate(nginx_ingress_controller_requests{}[5m])))

> 0.3
```

Security threats with Prometheus

SQL injection attacks:

High 500 errors in ingress controller:

```
(sum by (host, path)
   (rate(nginx_ingress_controller_requests{status=~"500"}[5m]))

/ sum by (host, path)
   (rate(nginx_ingress_controller_requests{}[5m])))

> 0.3
```

Security threats with Prometheus

Anomaly usage of volumes (possible symlink attack or DoS):

High inode usage in a volume compared with other volumes:

```
(sum by (instance)(kubelet_volume_stats_inodes_used)
/ sum by (instance)(kubelet_volume_stats_inodes))

>

scalar(
    avg (
        sum by (instance) (kubelet_volume_stats_inodes_used)
        / sum by (instance) (kubelet_volume_stats_inodes)))

+ 3 * scalar(
    stddev (
        sum by (instance) (kubelet_volume_stats_inodes_used)
        / sum by (instance) (kubelet_volume_stats_inodes)))
```

Security threats with Prometheus

Abnomal CPU consumption in container (possible crypto-mining):

Container consuming more CPU than the same containers in other pods of the same workload:

```
rate(container_cpu_usage_seconds_total{pod!="",cpu="total"}[5m]) * on (pod)
group_left(owner_name,owner_kind) kube_pod_owner

> on (owner_name,owner_kind,namespace,container) group_left

avg by (owner_name,container,namespace,owner_kind)
    (rate(container_cpu_usage_seconds_total{pod!="",cpu="total"}[5m]) * on (pod)
    group_left(owner_name,owner_kind) kube_pod_owner)

+ 3 * stddev by (owner_name,container,namespace,owner_kind)
    (rate(container_cpu_usage_seconds_total{pod!="",cpu="total"}[5m]) * on (pod)
    group_left(owner_name,owner_kind) kube_pod_owner)
```

Security threats with Prometheus

Abnomal network outbound bytes from a pod (possible data exfiltration):

Pod sending more bytes than the rest of the pods of the same workload:

```
sum by (pod,namespace)
    (rate(container_network_transmit_bytes_total{pod!=""}[5m])) * on (pod)
    group_left(owner_name,owner_kind) kube_pod_owner

> on (owner_name,owner_kind,namespace) group_left

avg by (owner_name,owner_kind,namespace)
    (sum by (pod,namespace)
        (rate(container_network_transmit_bytes_total{pod!=""}[5m])) * on (pod)
        group_left(owner_name,owner_kind) kube_pod_owner)

+ 3 * stddev by (owner_name,owner_kind,namespace)
    (sum by (pod,namespace)
        (rate(container_network_transmit_bytes_total{pod!=""}[5m])) * on (pod)
        group_left(owner_name,owner_kind) kube_pod_owner)
```

Security threats with Prometheus

Abnomal network outbound package size (possible massive extraction):

Abnormal packets response size in ingress controller:

```
    histogram_quantile(0.99,
        sum by (path,host, le)
        (rate(nginx_ingress_controller_response_size_bucket[5m])) )

    >

    avg_over_time(
        histogram_quantile(0.99, sum(rate(nginx_ingress_controller_response_size_bucket[5m])) by
        (path,host, le))[1h:5m])

    + 3 * stddev_over_time(
        histogram_quantile(0.99, sum(rate(nginx_ingress_controller_response_size_bucket[5m])) by
        (path,host, le))[1h:5m])
```

Time machine of kubernetes topology

Time machine of kubernetes topology

`kube-state-metrics` provides **`kube_pod_info`**, which can be used to infer some information about the cluster's topology

```
kube_pod_info{
    pod
    namespace
    host_ip
    pod_ip
    node
    created_by_kind
    created_by_name
    uid
    priority_class
    host_network
}
```

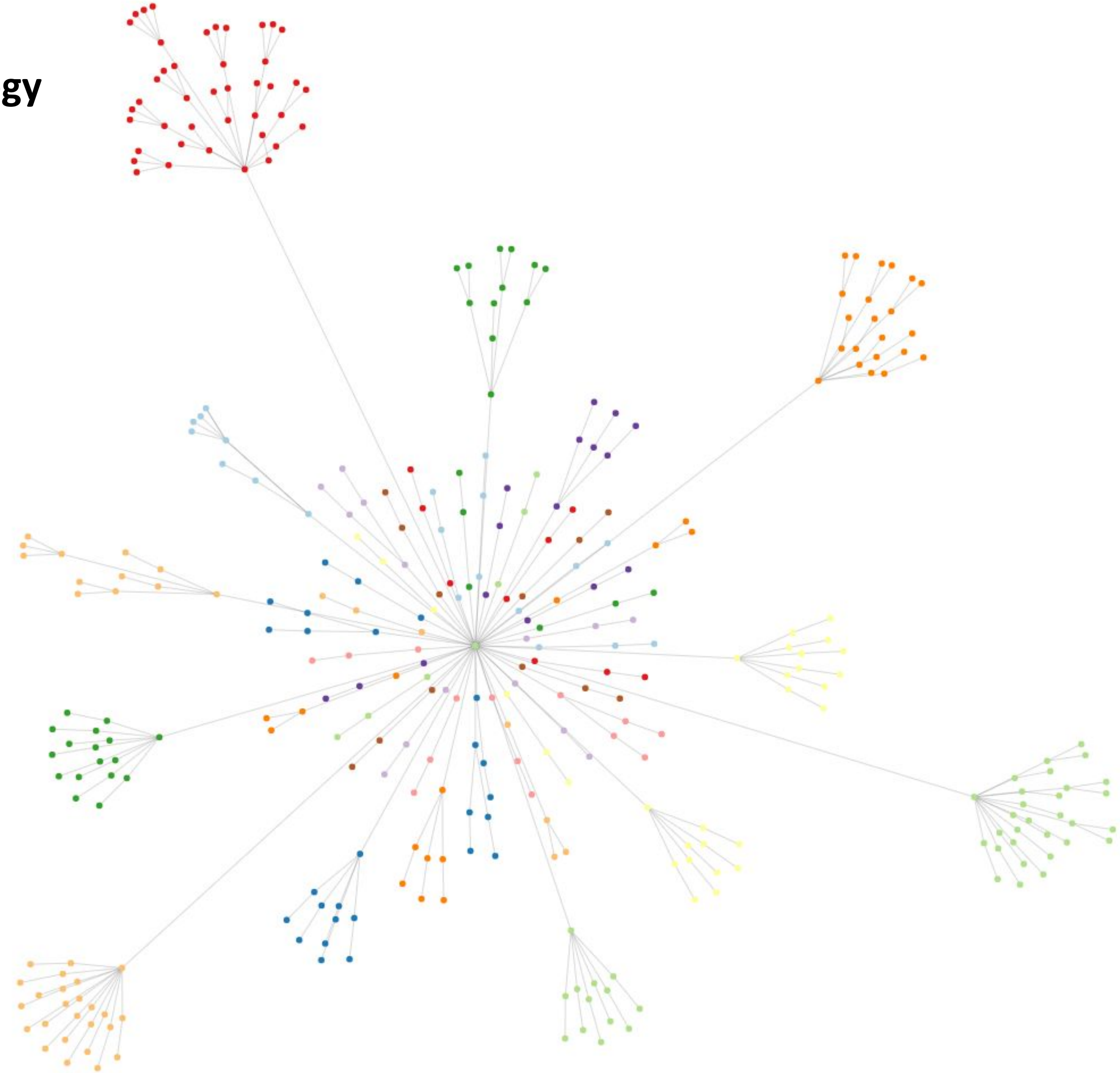
Deployment
StatefulSet
DaemonSet

The PromQL modifier `offset` allows us go back in time and query the value of the metric in the past:

```
kube_pod_info offset 5m

kube_pod_info offset 1h

kube_pod_info offset 2d

kube_pod_info offset 4w
```

Time machine of kubernetes topology

Visualizing Infrastructure Topology

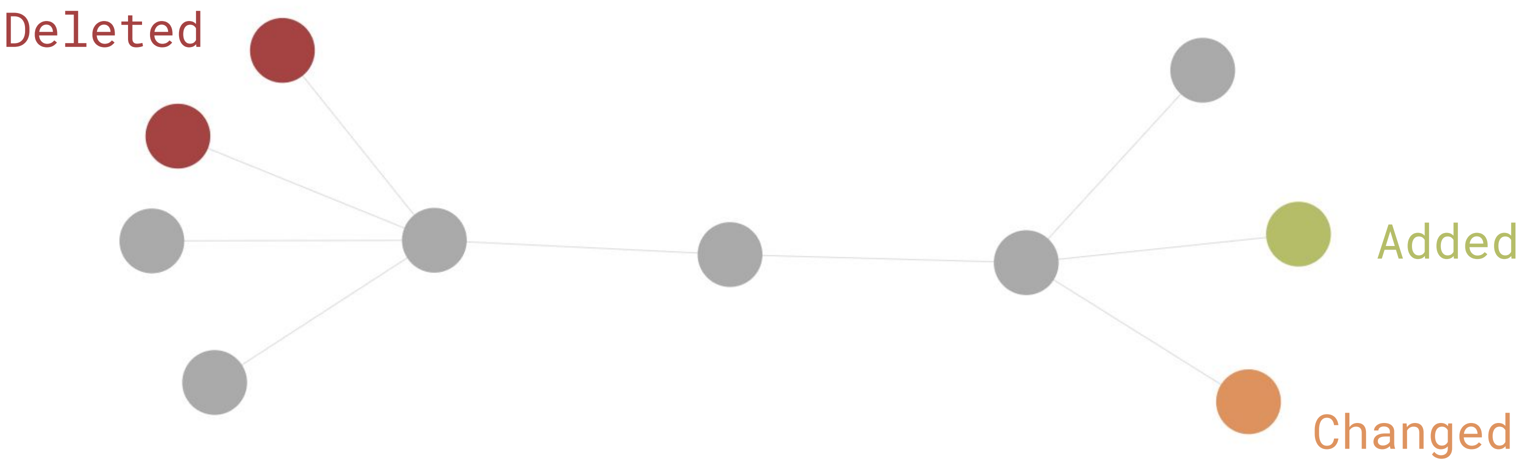


https://gist.github.com/mircodezorzi/fbcbd69319888d23717daaaa9361e98c

Time machine of kubernetes topology

KubeCon | CloudNativeCon Europe 2023

Visualizing Topology Over Time



Time machine of kubernetes topology

Service meshes can give more insight regarding network topology, emitting service level metrics with request source and destination.

Istio

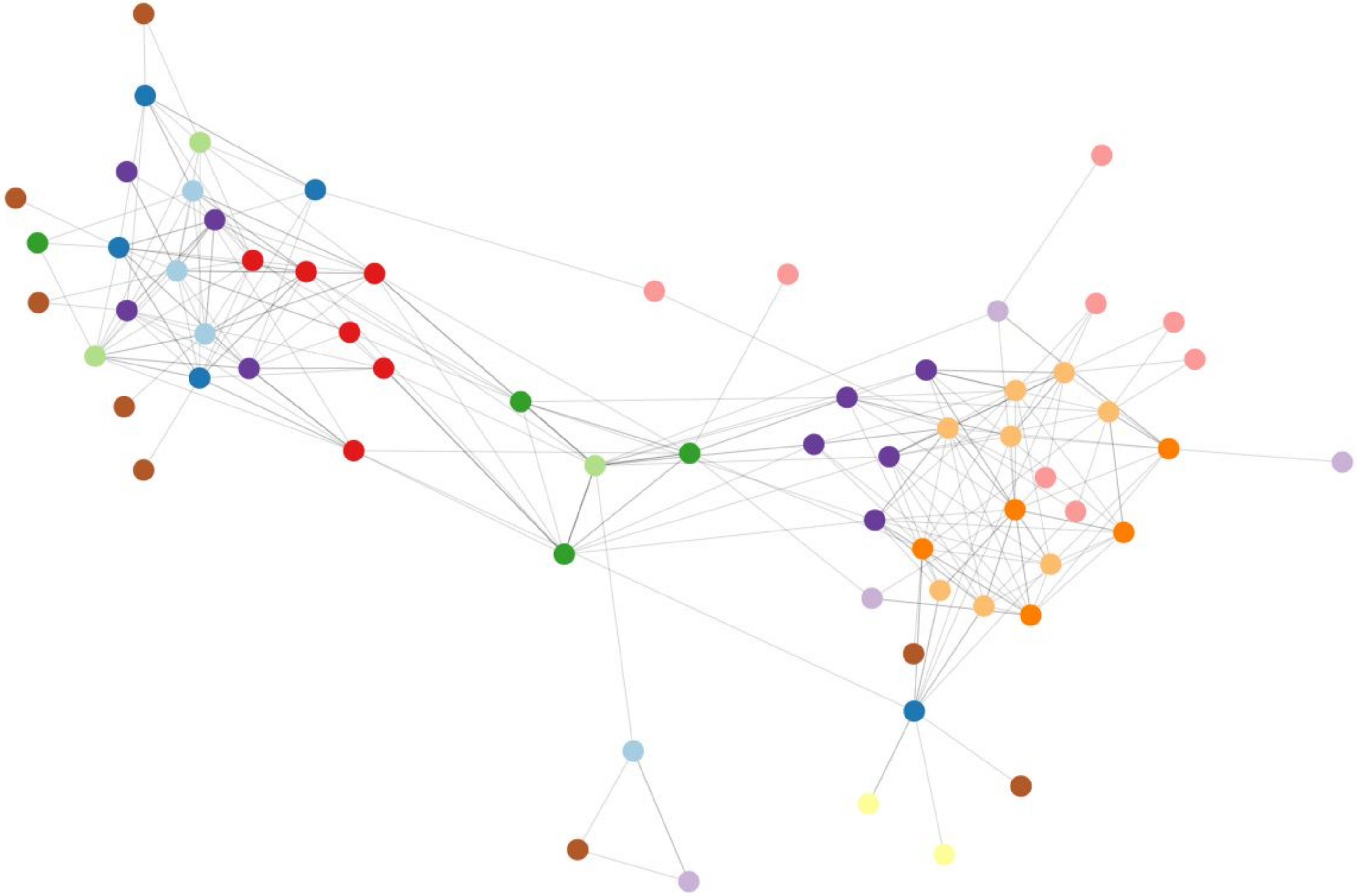
```
istio_requests_total{
   source_app
   source_workload
   destination_app
   destination_workload
   ...
}
```

Linkerd

```
request_total{
   pod
   deployment
   ...
   dst_deployment
}
```

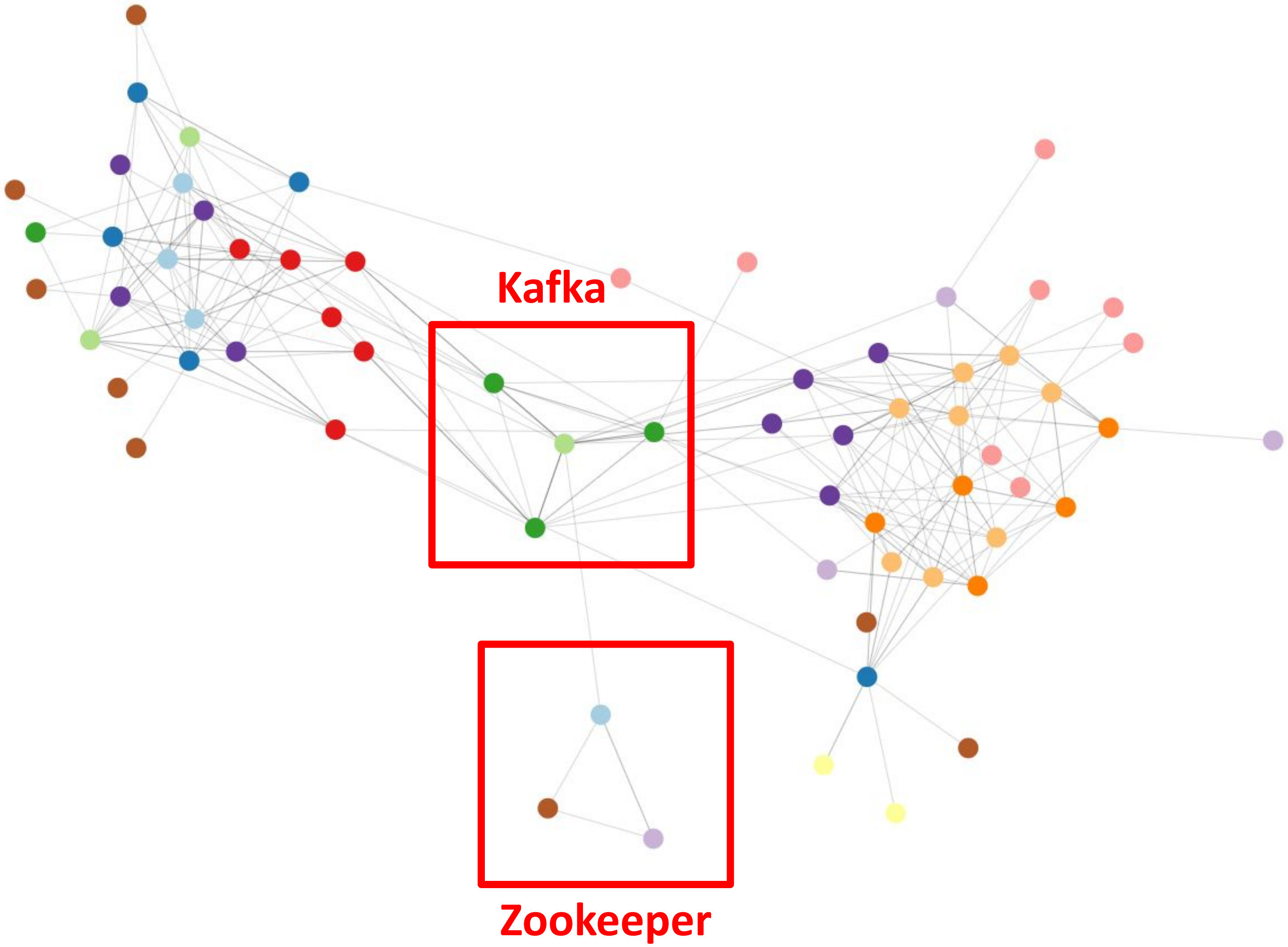
Time machine of kubernetes topology

Visualizing Network Traffic Between Pods



Time machine of kubernetes topology

Visualizing Network Traffic Between Pods



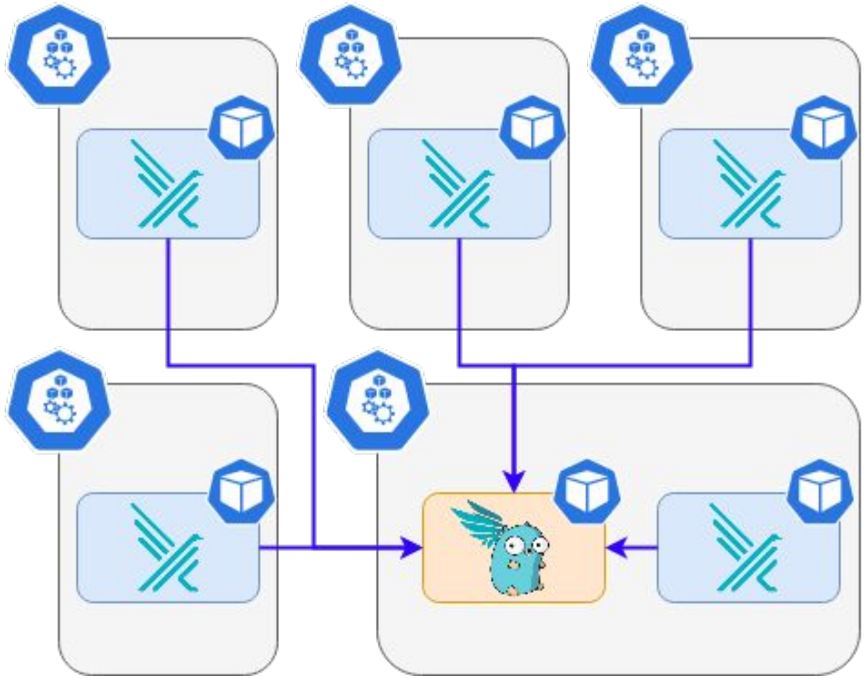
Kafka

Zookeeper

Monitoring Falco

Monitoring Falco

	Falco Exporter	Falco Sidekick
Type of deployment	Sidecar	Deployment
Cardinality	1 / node	1 / cluster
Metrics	Falco Events	Falco events Output (destination + status)
Extra labels	❌	✅



Monitoring Falco

High number of detections of a rule:

```
increase(falco_events{rule="Write below monitored dir"}[5m]) > 5
```

High number of critical events:

```
increase(falco_events{priority="Critical"}[5m]) > 10
```

High number of errors in outputs:

```
rate(falcosidekick_outputs{destination="webui", status="error"}[5m]) /
rate(falcosidekick_outputs{destination="webui"}[5m])
> 0.1
```

Please scan the QR Code above
to leave feedback on this session