

Feature Project Proposal——Akraino Profiling

Helloway He

(helloway.wewe@gmail.com)

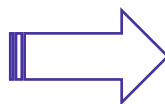
05/02/2019



Background - Requirement for Unified Metrics Exposition Format

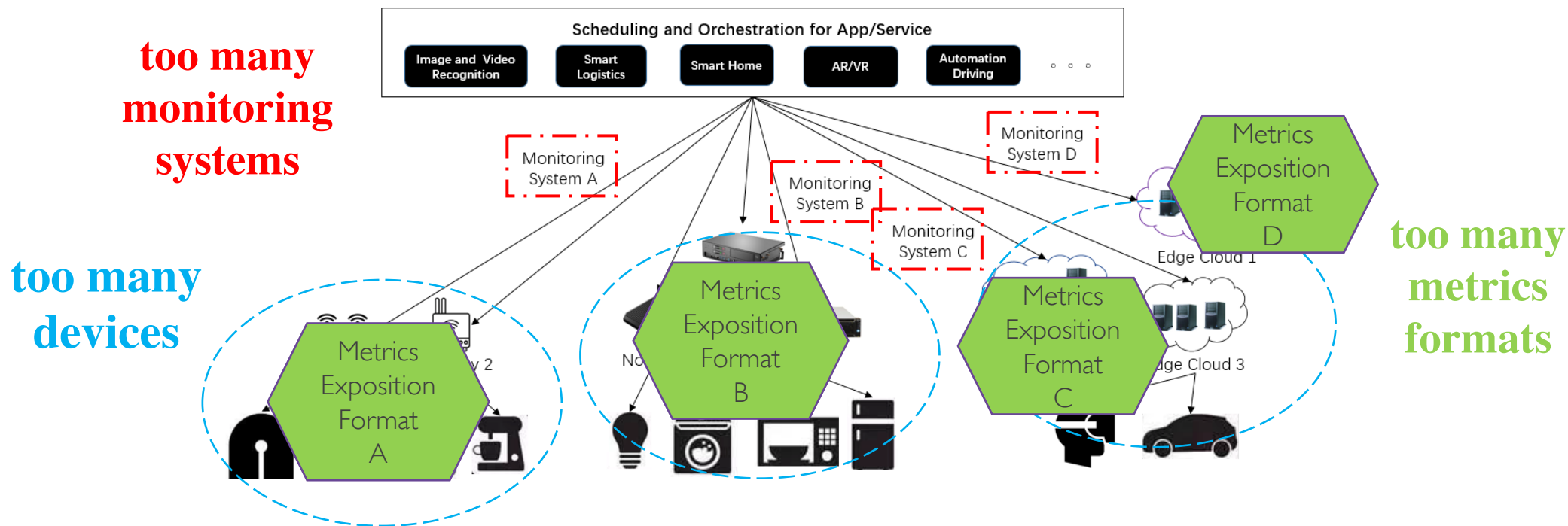
Requirement

- › Have the unified metrics exposition format standard at the edge



Advantages

- Reduce development complexity
- Vendors only use define metrics exposition format
- Develop one format conversion tool at most
- Gather more comprehensive metrics, do better scheduling



- Hard to choose the optimal running nodes/devices
- High development complexity
- How to seamlessly connect all edge nodes/devices

Background – OpenMetrics Metrics Exposition Format Standard

- A Metrics Exposition Format Standard
- CNCF sandbox project launched by the Prometheus community
- Prometheus has become a de-facto standard in metric monitoring
- Take out Prometheus' metrics format alone to set up OpenMetrics project
- Main contributors
 - Google, Prometheus, InfluxData, SolarWinds, Open Census, Uber, Data Dog, etc.

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="post",code="200"} 1027 1395066363000
http_requests_total{method="post",code="400"} 3 1395066363000

# Escaping in label values:
msdos_file_access_time_seconds{path="C:\\DIR\\FILE.TXT",error="Cannot find file:\\n\\FILE.TXT\\"} 1.458

# Minimalistic line:
metric_without_timestamp_and_labels 12.47

# A weird metric from before the epoch:
something_weird{problem="division by zero"} +Inf -3982045

# A histogram, which has a pretty complex representation in the text format:
# HELP http_request_duration_seconds A histogram of the request duration.
# TYPE http_request_duration_seconds histogram
http_request_duration_seconds_bucket{le="0.05"} 24054
http_request_duration_seconds_bucket{le="0.1"} 33444
http_request_duration_seconds_bucket{le="0.2"} 100392
http_request_duration_seconds_bucket{le="0.5"} 129389
http_request_duration_seconds_bucket{le="1"} 133988
http_request_duration_seconds_bucket{le="+Inf"} 144320
http_request_duration_seconds_sum 53423
http_request_duration_seconds_count 144320
```

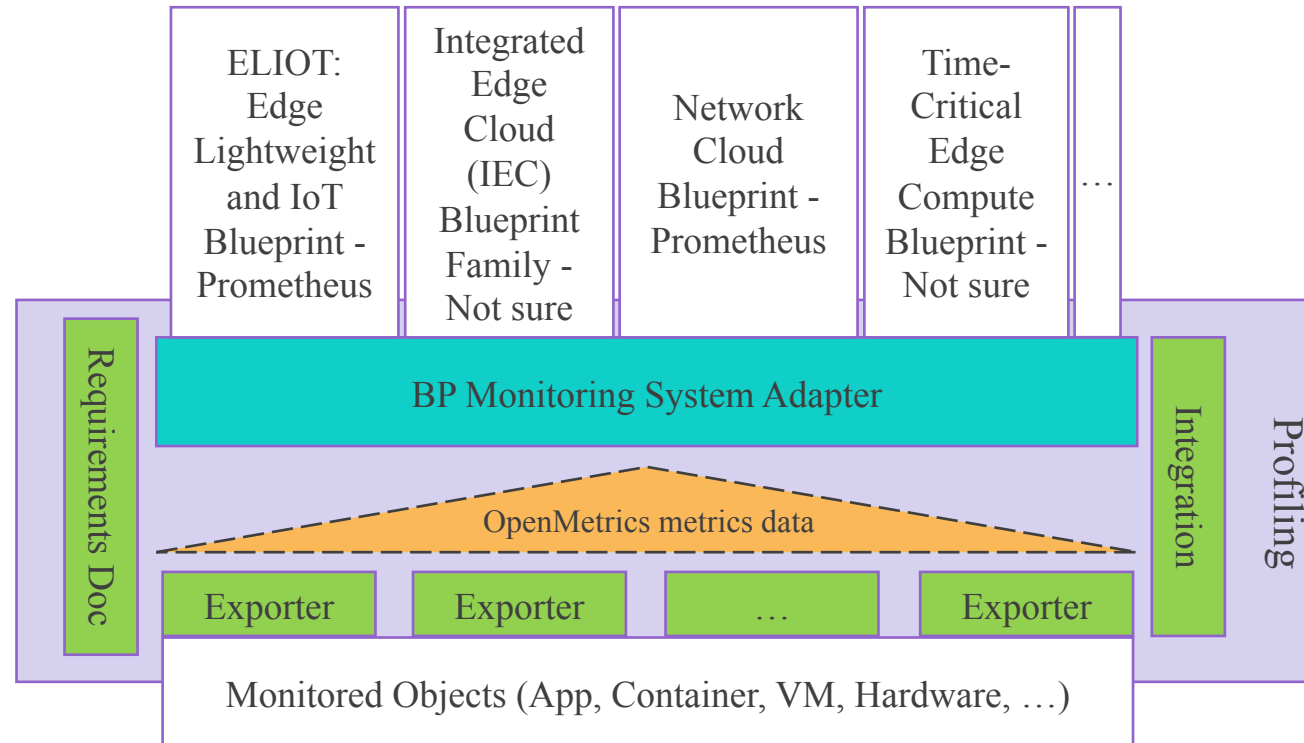
OpenMetrics Metrics Data Format



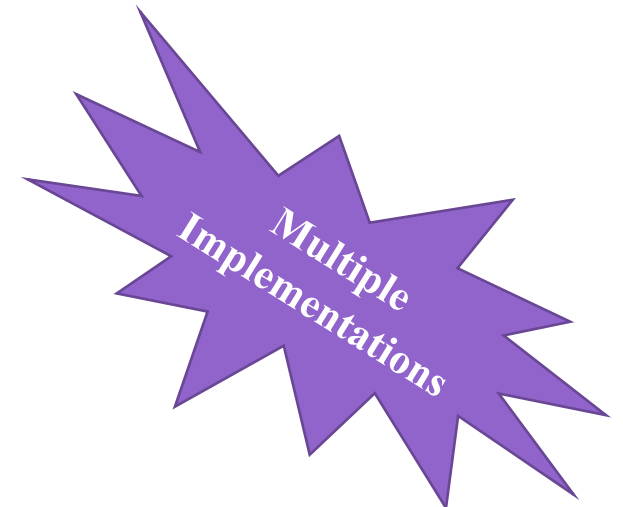
Akraino Profiling Feature Project & Framework

Akraino Profiling Goals - Use OpenMetrics to unify the end-to-end metrics exposition format for the edge computing,

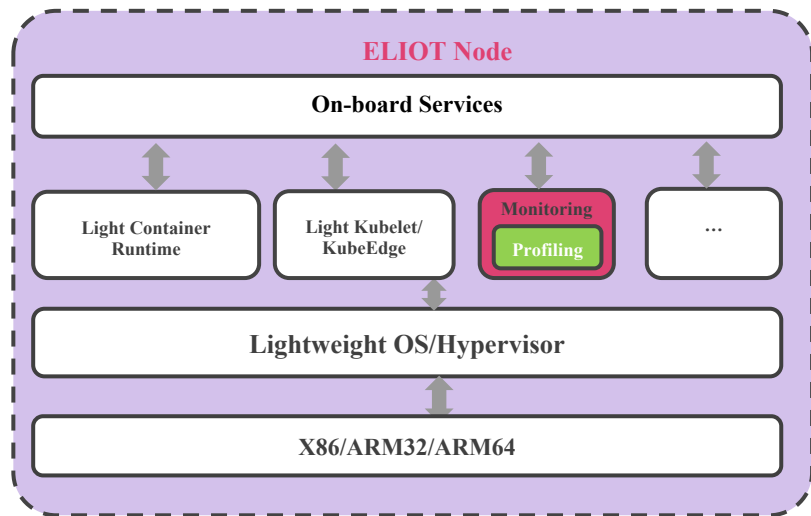
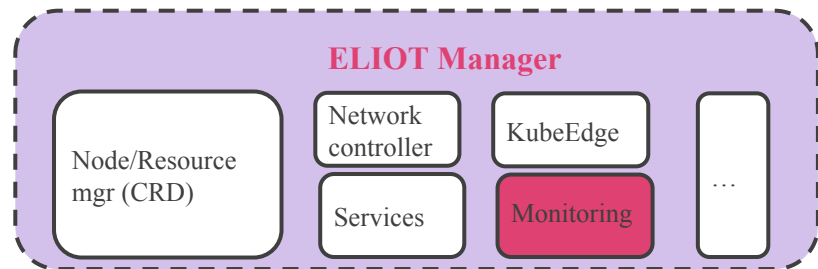
1. Requirement Doc
 - Analyze which metrics data need to be monitored, such as: node_memory_total, reponse_lantency, transmission_bandwidth.
2. Exporter
 - Collect metrics data from the monitored object
 - Expose those metrics in OpenMetrics format
3. Adapter (optional, pluggable and dynamically loaded component)
 - Do format conversion work
 - Analyze whether the BP monitoring systems need to develop the adapter or not
 - OpenMetrics compliance
 - OpenMetrics incompliance
4. Integration



- Requirement Doc – Required;
- Exporter – Required;
- Adapter – Optional;
- Integration – Required;



Profiling Implementation - OpenMetrics Compliant



Monitoring (developer/user tools)

Using prometheus and cadvisor to monitoring resources

Profiling

Add/update **exporters** to collect and expose the metric data to Prometheus server on the Eliot Manager in the OpenMetrics format

- **Pre-analysis**
Monitoring Metrics Data: eliot_node1_sound, eliot_node1_temperature,... The requirements document is roughly as follows.
- **Exporter**
Add a new node exporter/update an existing node exporter to collect and expose metrics data which are compliant with OpenMetrics standard,
- **Adapter (Eliot Not required)**
Eliot uses Prometheus as the monitoring system that can directly process the OpenMetrics metric data collected;
- **Integration**
Develop scripts based on ansible/helm to integrate Profiling with Eliot.

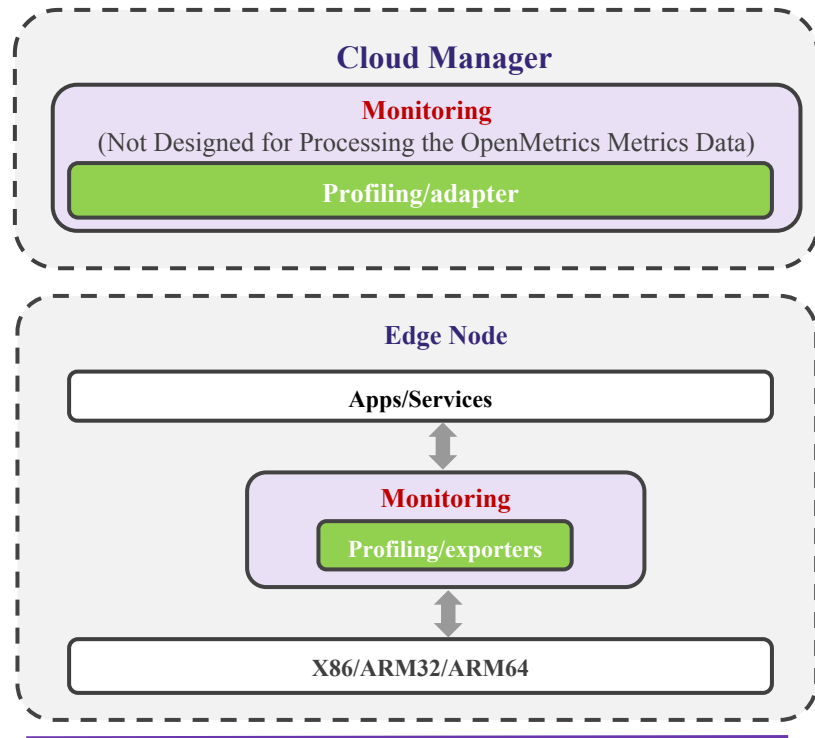
```
# HELP eliot_node1_sound Sound (noise) level.
# TYPE eliot_node1_sound gauge
eliot_node1_sound 1
# HELP eliot_node1_temperature Temperature in C and F.
# TYPE eliot_node1_temperature gauge
eliot_node1_temperature{metric="celsius"} 19.093447311567388
eliot_node1_temperature{metric="fahrenheit"} 66.3682051608213
...
```

Monitoring Metric Data

Metric Name	Metric Type	Metric Description
eliot_node1_sound	gauge	Sound (noise) level
eliot_node1_temperature	gauge	Temperature in C and F



Profiling Implementation - OpenMetrics Noncompliant



Monitoring (developer/user tools)

Using Ganglia, Nagios, etc., and their own collector daemon to monitoring resources

Profiling

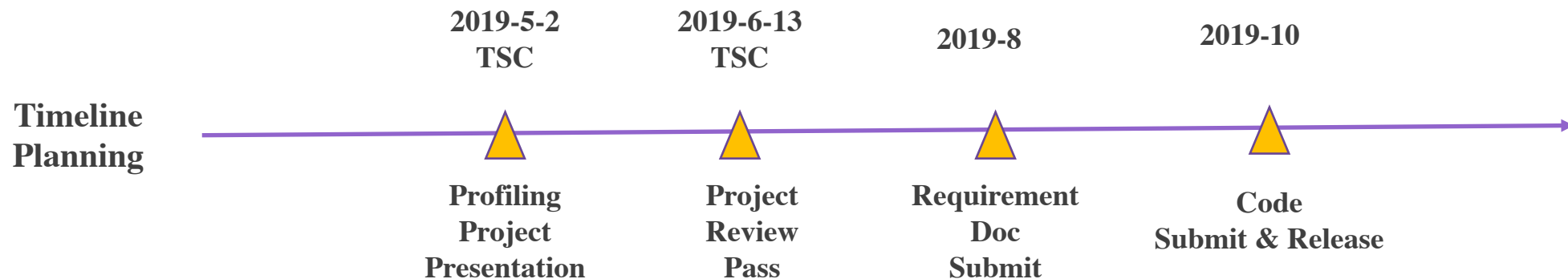
1. **Edge Node** Add/redevelop **exporters** to collect and expose the metric data to the monitoring system on the Cloud Manager in the OpenMetrics format.
2. **Cloud Manager** Develop **adapter** to convert OpenMetrics metric data format to proprietary format that they can process.

Take Ganglia (on the cloud) and gmond (Ganglia collector daemon at the edge) as an example:

- **Pre-analysis**
Determine the monitoring metrics;
- **Exporter**
Redevelop **gmond_exporter** to serve OpenMetrics metrics for ganglia on the Cloud;
- **Adapter (Ganglia required)**
Develop **ganglia_adapter** for ganglia to convert the OpenMetrics metric to ganglia metrics as it can't directly process the OpenMetrics metric collected by gmond_exporter;
- **Integration**
Develop scripts based on ansible/helm to integrate Profiling which includes cloud part and edge part.

Next Steps

- Preliminarily plan to focus on IOT scenario - Eliot
 - Complete Metrics Requirement Analysis
 - Add/Update a exporter at least
 - Develop scripts to integrate with Eliot
 - Expected to be released with Eliot in R2
- Future plans to extend to more Akraino Blueprint Families/BPs



Akraino Profiling Summary

Feature	Description	Companies Participating / Committers	Requested Release / Timeline	Informational
Akraino Profiling	<p>Akraino Profiling to use OpenMetrics to unify the end-to-end metrics exposition format for the edge computing, i.e., make all the edge nodes/devices/apps/services expose their metrics data in OpenMetrics format.</p> <ol style="list-style-type: none"> Pre-analysis Analyze which metrics data require to be monitored, then aggregate them into the requirement document; Exporter Development Analyze which exporters need to be add/redevelop/updated according to the metrics requirement document and develop them; Adapter Development (optional, pluggable and dynamically loaded component) Analyze which BP monitoring systems need to develop the adapter and develop them; Integration To develop scripts based on ansible/helm to integrate Profiling into each BP. 	Huawei ARM Dell	R2	<p>Impacted Blueprint Family – Applies to all BP Families and Blueprints</p> <p>See next slide for additional details</p>

Q & A

Thanks