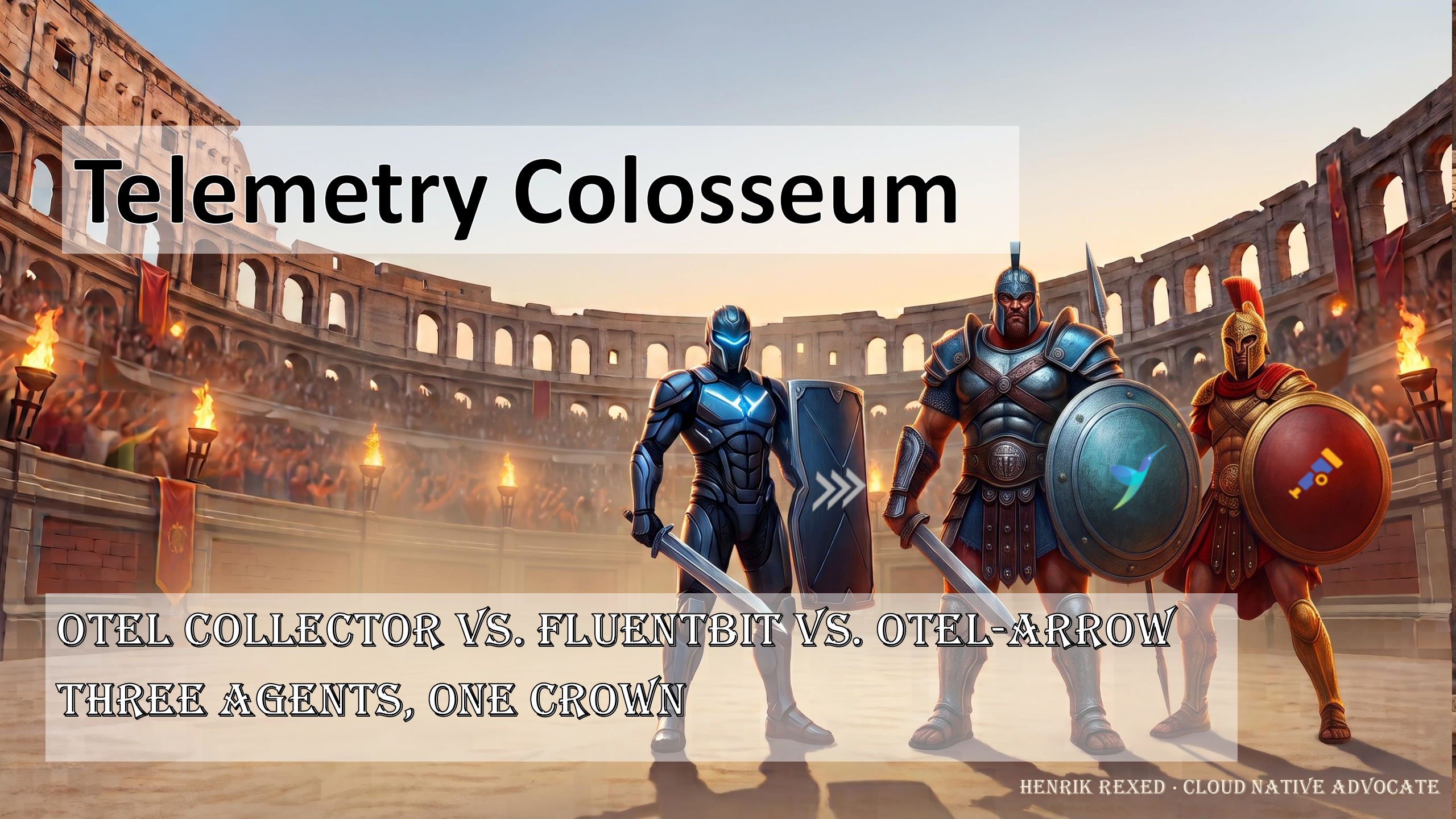
Telemetry Colosseum



OTEL COLLECTOR VS. FLUENTBIT VS. OTEL-ARROW

THREE AGENTS, ONE CROWN

HENRIK REXED · CLOUD NATIVE ADVOCATE

Henrik Rexed

Cloud Native Advocate, CNCF Ambassador, AAIF Ambassador



- **15+ years of Performance engineering**
- **Owner of IsitObservable**

  Producer of Perfbytes

DISCLAIMER

- **No birds or telescopes were harmed in the making of this presentation**
- **The intention behind this talk track is not to assign blame to any CNCF project.**
- **This session is made to help the community in choosing their telemetry agent.**

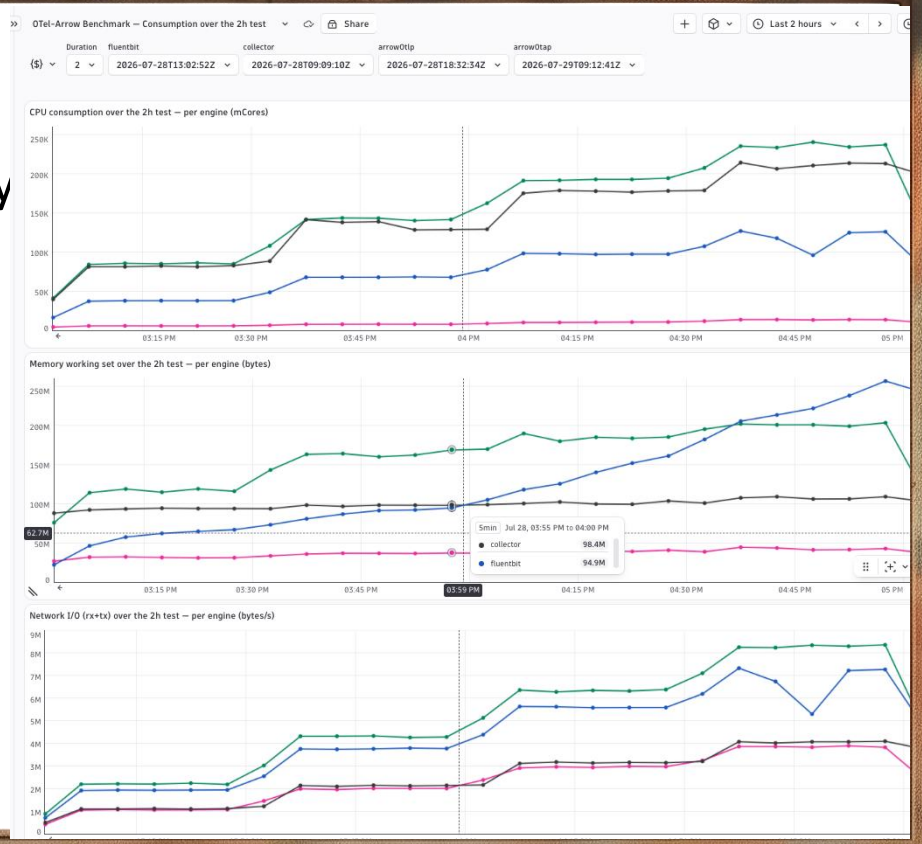


OUR 3 GLADIATOR



If you stay with me you will …

- Explain the value of OTAP
- Understand the difference between each engine by looking at :
  - Desing Experience
  - The plugins for :
    - Logs
    - Metrics
    - Traces
- See various Benchmarking results
- Recommendation on which agents needs to used under specific conditions




The Protocol

Protocol Compatible for all signals

OTLP



OTAP



OTLP

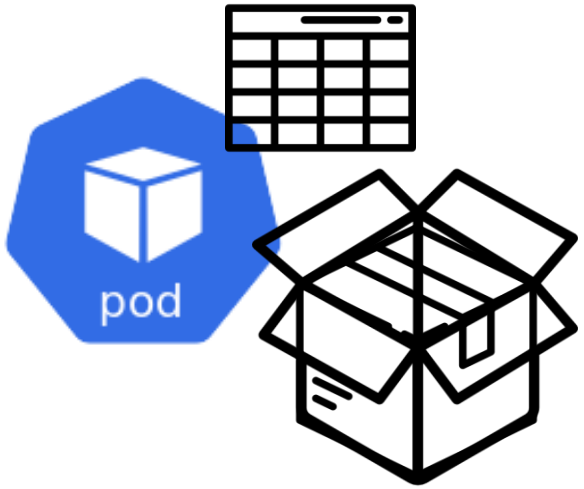
Service.name	http.response.code	http.method	Duration
Checkout	200	GET	100ms

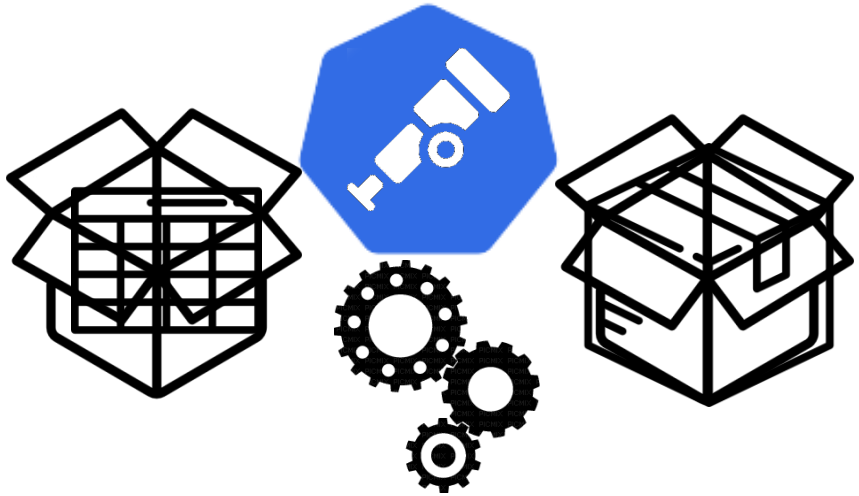
Service.name	http.response.code	http.method	Duration
Paiement	200	POST	200ms

Service.name	http.response.code	http.method	Duration
Checkout	200	GET	90ms

OTLP

- Data encoded as Protobuf message





OTAP

Service.name	http.response.code	http.method	duration
Checkout	200	GET	100ms
Paiment	200	POST	200ms
Checkout	200	GET	90ms

OTAP

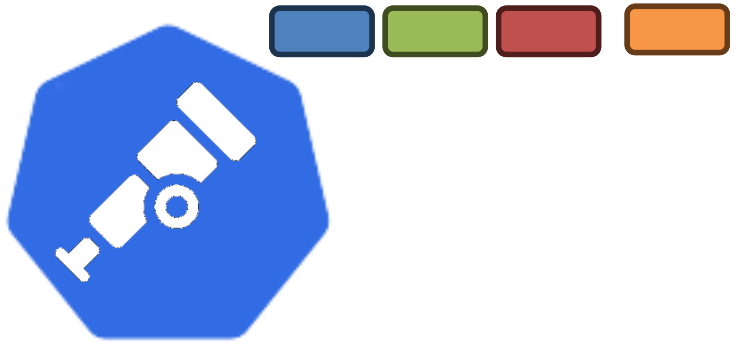
dictionary

Id	Service
1	checkout
2	paiement

record

Service id	http.responose.code	http..method	duration
1	200	GET	100ms

OTAP

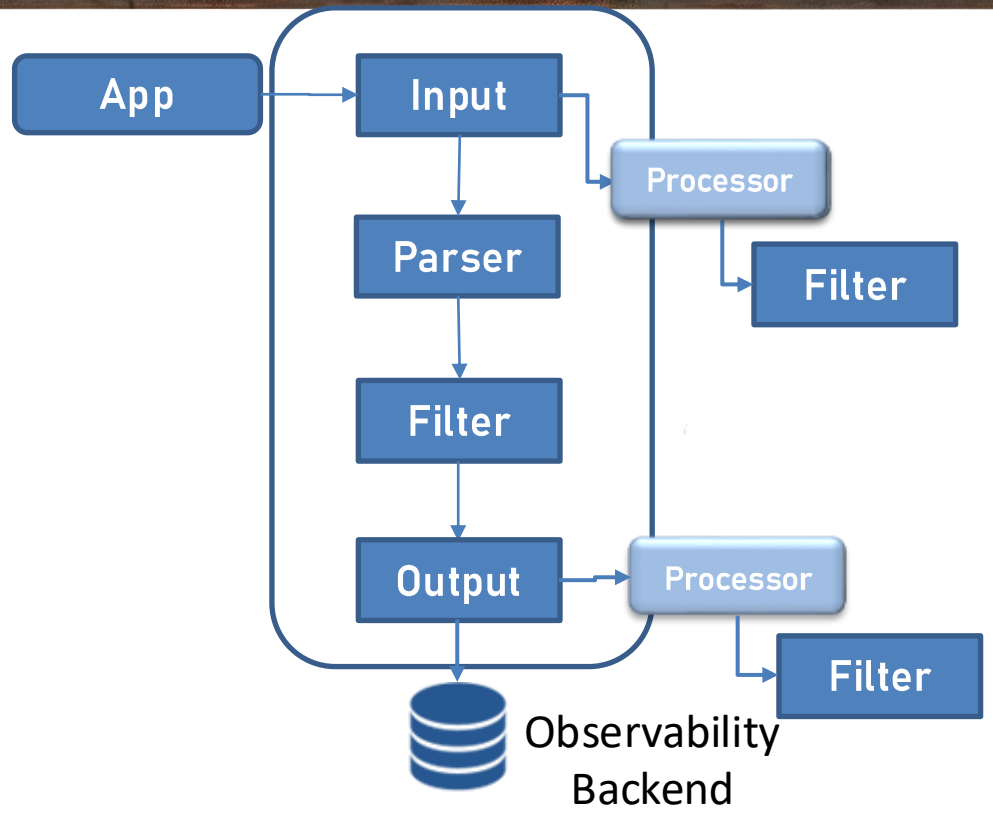




Design

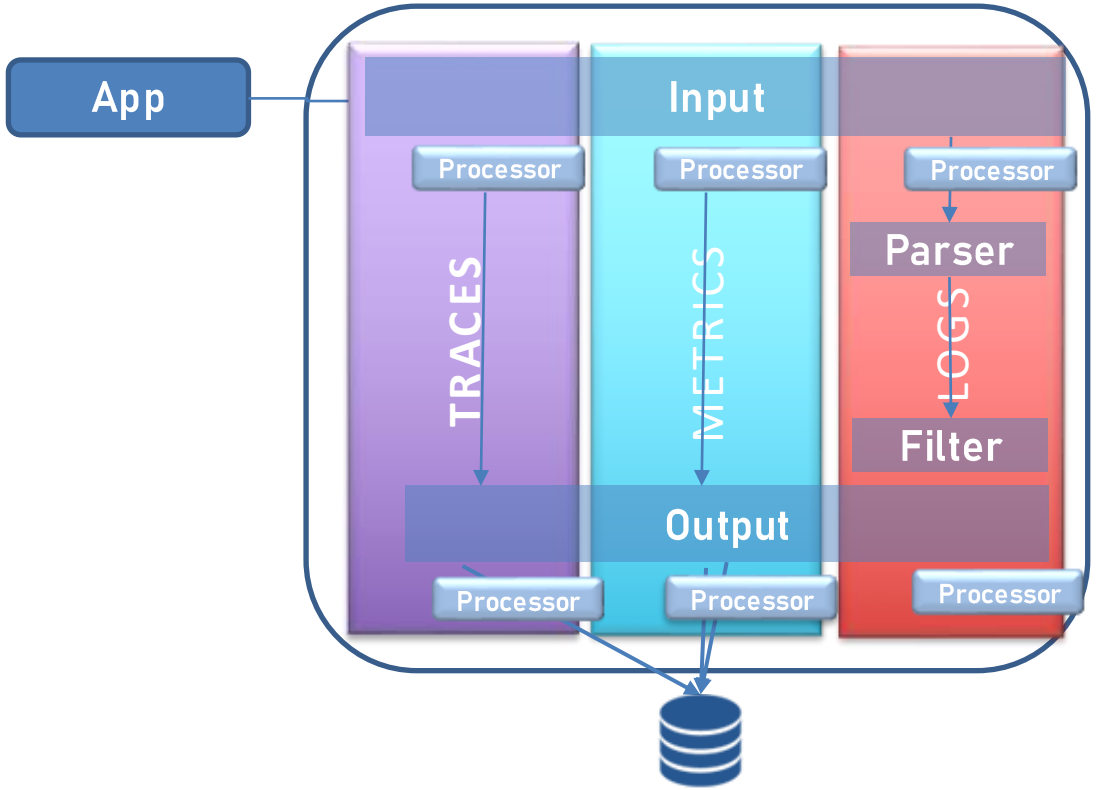

CHAPTER 2: THE WEAPON
(THE GHOSEN BLADE)
PRECOHON, BUTTERS & PROCESSING
OTLP
Latancy
Throughpot
Latergy

Fluentbit



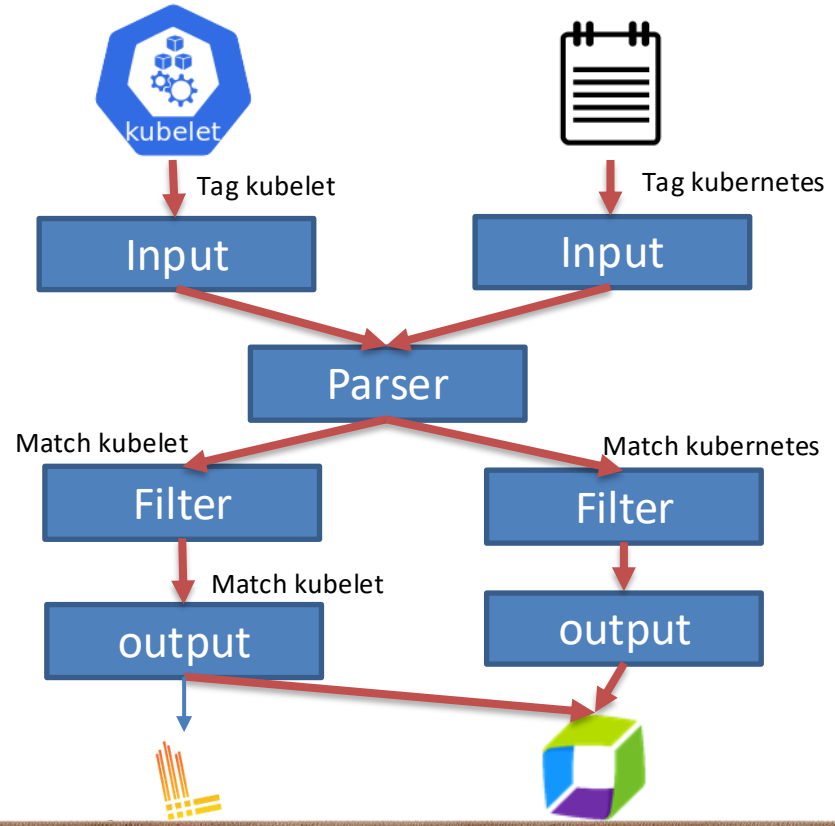

```
pipeline:
  inputs:
    - name: tail
      path: /var/log/containers/*.log
      multiline.parser: docker, cri
      tag: kube.*
      mem_Buf_Limit: 5MB
      skip_Long_Lines: On
      processors:
        logs:
          - name: modify
            match: "kube.*"
            add:
              - k8s.cluster.name ${CLUSTERNAME}
              - dt.kubernetes.cluster.id ${CLUSTER_ID}
  filters:
    - name: kubernetes
      match: kube.*
      merge_log: on
      keep_log: off
      k8s-logging.parser : on
      k8S-logging.exclude: on
    - name: modify
      match: kube.*
      rename:
        - log content
        - kubernetes_pod_name k8s.pod.name
        - kubernetes_namespace_name k8s.namespace.name
      remove:
        - kubernetes_container_image
        - kubernetes_container_hash
  outputs:
    - name: http
      host: ${DT_ENDPOINT_HOST}
      port: 443
      match: “kube.*"
```

Fluentbit design pattern

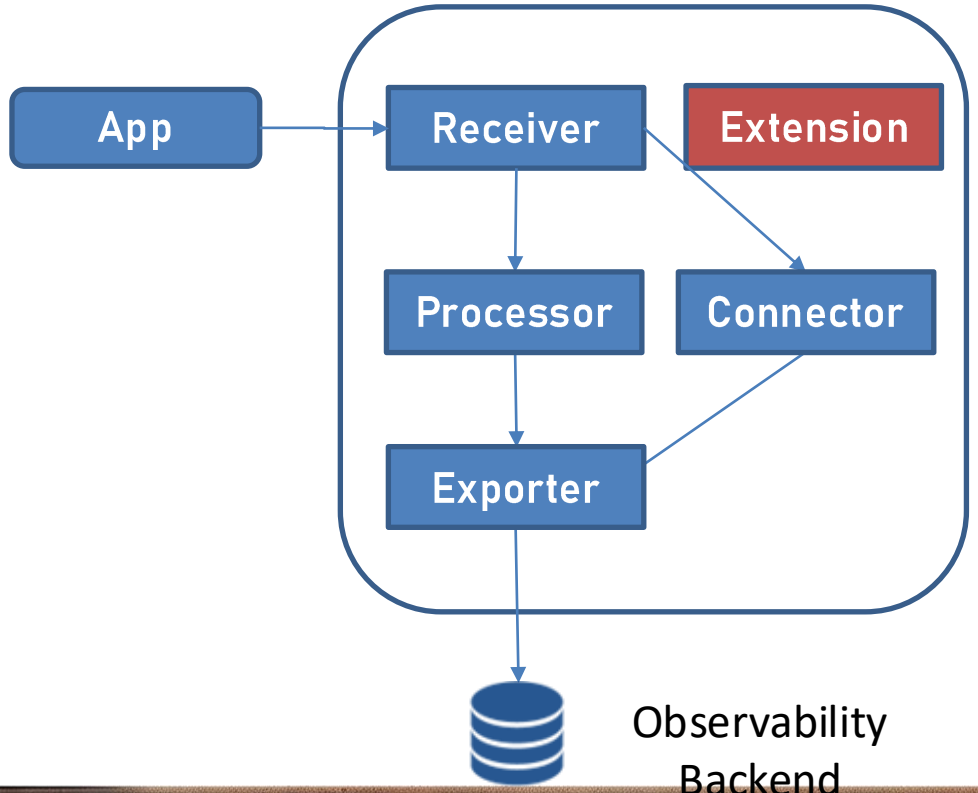


Fluentbit : Tag is key

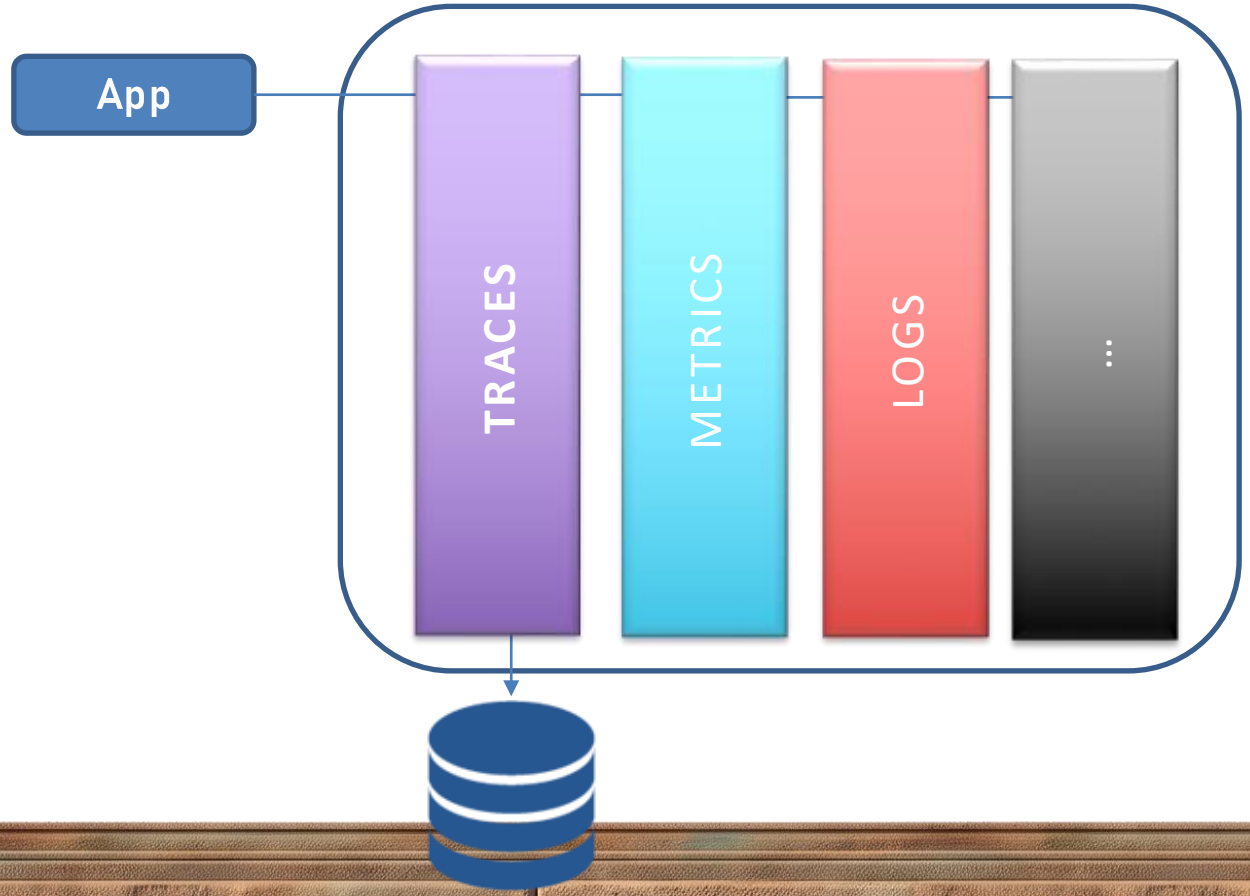
- **In Fluentbit the pipeline flow rely on Tag**
- **Input plugins will assign a Tag**
- **Tag will determine which content will be routed to specific filter, parser, output plugins**
- **Tag is the key concept to define your pipeline flow**



The Collector

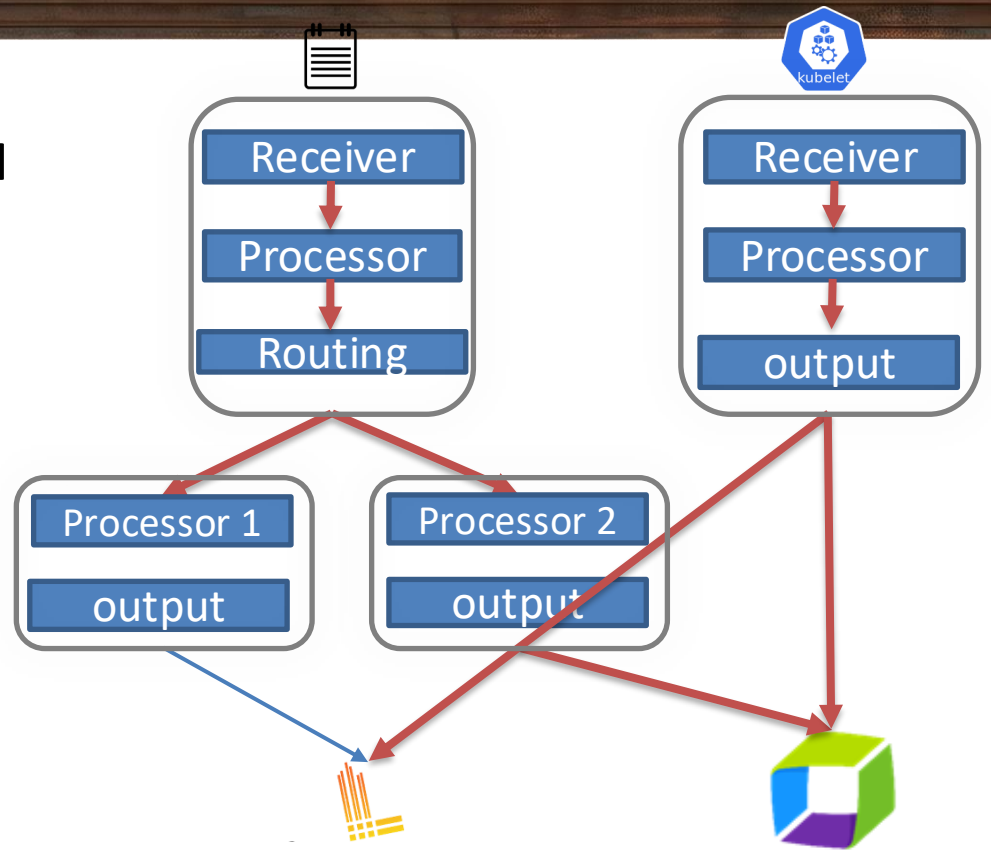



```
receivers:
filelog:
  include:
   - /var/log/pods/*/*/*.log
  start_at: beginning
  include_file_path: true
  include_file_name: false
processors:
k8sattributes:
  auth_type: "serviceAccount"
  passthrough: false
  filter:
    node_from_env_var: K8S_NODE_NAME
 memory_limiter:
  check_interval: 1s
  limit_percentage: 70
  spike_limit_percentage: 30
exporters:
 otlphttp:
  endpoint: $DT_ENDPOINT/api/v2/otlp
  headers:
   Authorization: "Api-Token $DT_API_TOKEN"
service:
 pipelines:
  logs:
   receivers: filelog
   processors: memory_limiter,k8sattributes batch
   exporters: otlphttp
```

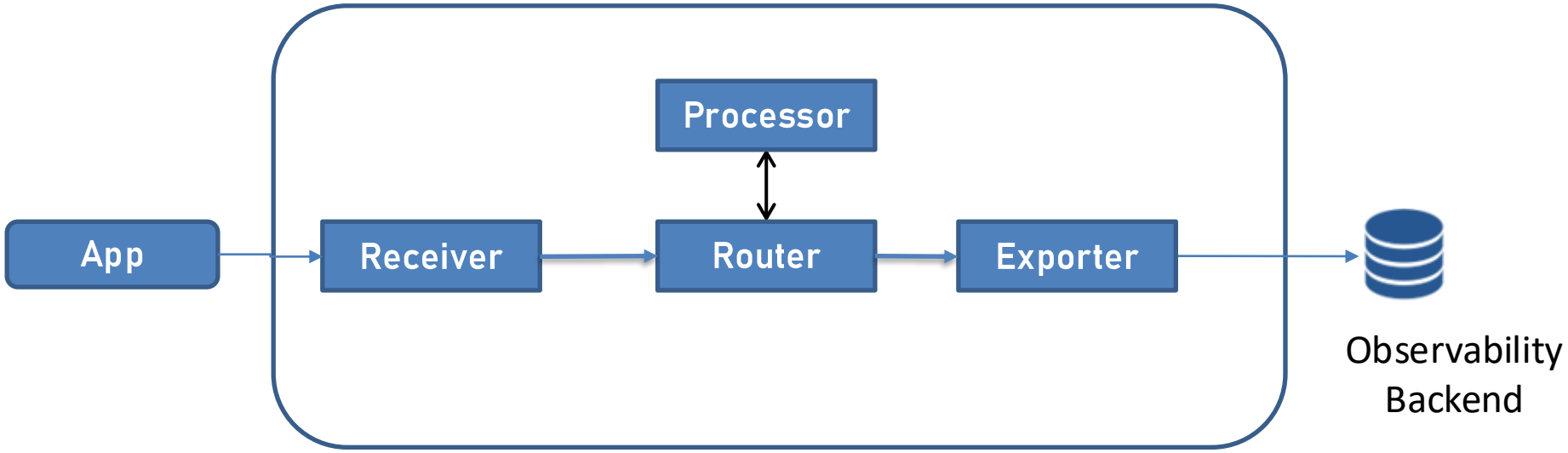

App
TRACES
METRICS
LOGS
...

OpenTelemetry Collector: Connector

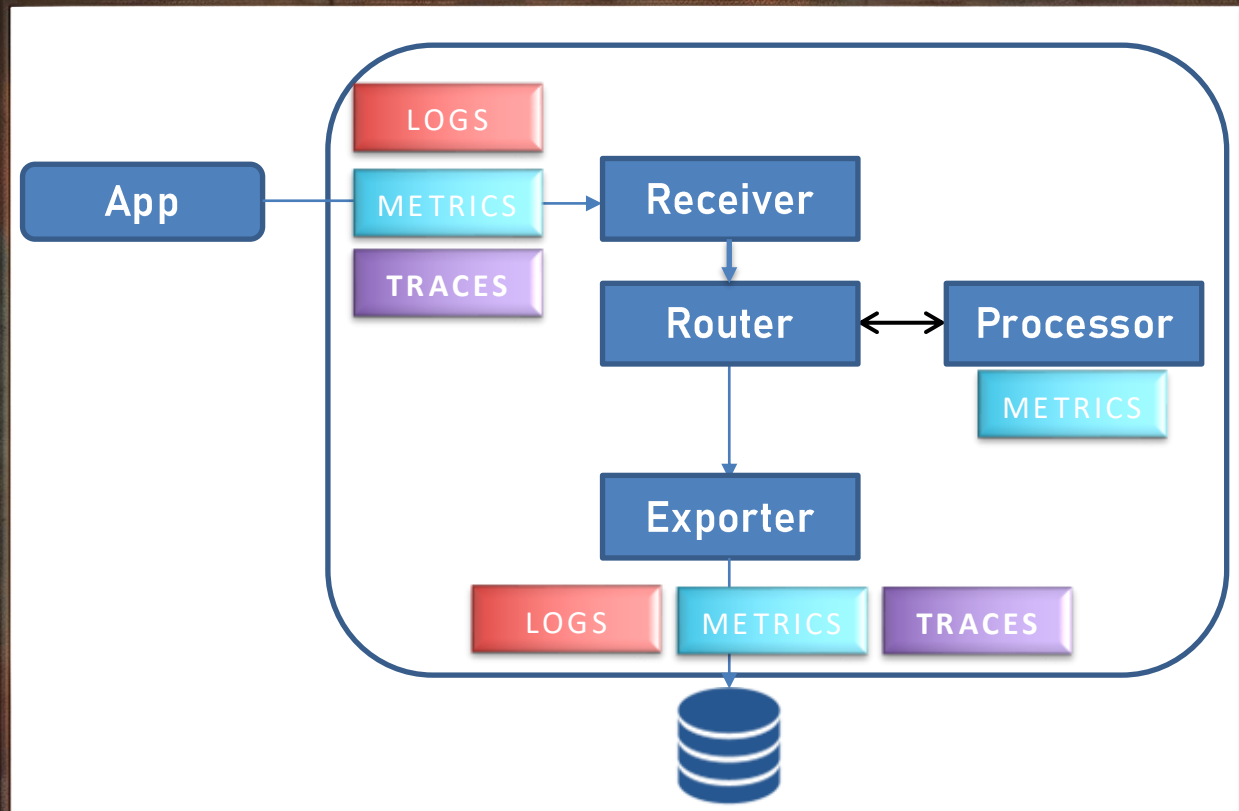
- **The collector provides features helping us to build complex pipeline. The flow would be different based on the content by :**
  - **Building separate pipelines for specific format**
  - **Or taking advantage of the routing connector (and probably transform)**



Otel-arrow


Processor
App
Receiver
Router
Exporter
Observability
Backend

Otel-arrow - Design Patterns




```
default:
    pipelines:
      main:
        type: otap
        nodes:
          otlp/ingest:
            type: receiver:otlp
            ...
          attributes/enrich:
            type: processor:attribute
            config:
              actions:
                ...
          router/by_signal:
            type: processor:type_router
            outputs: ["logs", "metrics", "traces"]
            config: {}

          batch/logs:    { type: processor:batch, config: { send_batch_size: 1000, timeout: 5s } }
          batch/metrics: { type: processor:batch, config: { send_batch_size: 1000, timeout: 5s } }
          batch/traces:  { type: processor:batch, config: { send_batch_size: 1000, timeout: 5s } }

          otlp_http/dynatrace:
            type: exporter:otlp_http
            config:
              ...
        connections:
          - { from: otlp/ingest,       to: attributes/enrich }
          - { from: attributes/enrich to: router/by_signal }
          - { from: [batch/logs, batch/metrics, batch/traces], to: otlp_http/dynatrace }
```


LOGS
[ERROR] PUMY1 - MICROTIMESTAMPS
[ERROR] PUMY1 - MIRS-TIMESTERRRT
[ERROR] PUMY1 - MICROTIMESTINHTS
[ERROR] PUMY1 - L ESTIMESTARPRT
[ERROR] PUMY1 - MICRO-TIMESORYDT
[ERROR] PUMY1 - ERTRKROCK ERROR
[ERROR] PUMY1 - MICROTIMESTEFMUS
[ERROR] PUMY1 - MICRO-TIMESTEFMUS

Plugins Required to receive Logs

- **To receive logs we need the following plugins :**
  - **UDP**
  - **TCP**
  - **FluentForward**
  - **OpenTelemetry**
  - **Syslog**
  - **Kafka**
  - **And ofcourse Read from a file**

UDP	✓	✓	
TCP	✓	✓	
Fluent	✓	✓	
Syslog	✓	✓	✓
File	✓	✓	✓
OpenTelemetry	✓	✓	✓
Kafka	✓	✓	✓
Otap		✓	✓

Plugins Required to Process Logs

- **To process logs we need the following plugins :**
  - **Enrich the logs with:**
    - **Extra resources**
    - **Kubernetes metadata**
  - **Parse the content**
  - **Drop**
  - **Batch the logs**
  - **…etc**

			
Resource	Modify	resource	attributes
Kubernetes	✓	✓	
Filter	grep	filter	filter
Parse	Regexp, json, parser…etc	Transform (with OTTL)	Transform (OPL)
Batch	Throttle	Batch	Batch

metrics



Plugins required to receive metrics

- **To receive metrics we mainly need the following plugins :**
  - **Collectd**
  - **Statsd**
  - **OpenTelemetry**
  - **Prometheus**
  - **Host Metrics**

			
Collectd	✓	✓	
Statsd	✓	✓	
Prometheus	✓	✓	
OpenTelemetry	✓	✓	✓
Host	✓	✓	✓
OTAP		✓	✓

Plugins required to Process metrics

- **To process metrics we mainly need the following plugins :**
  - **Enrich the metric with:**
    - **Extra resources**
    - **Kubernetes metadata**
  - **Drop**
  - **Convert**

			
Ressource	labels	✓	attributes
Transform		✓	✓
Filter	metric_selector	✓	✓
CumulativetoDelta	✓	✓	✓
DeltaToRate		✓	
kubernetes		✓	


TRACES

Plugins required to receive traces

- **To receive traces we mainly need the following plugins :**
  - **OpenTelemetry**
  - **Zipkin**
  - **Kafka**
  - **…Etc**

			
OpenTelemetry	✓	✓	✓
zipkin		✓	
Kafka		✓	
OTAP		✓	✓

Plugins required to Process traces

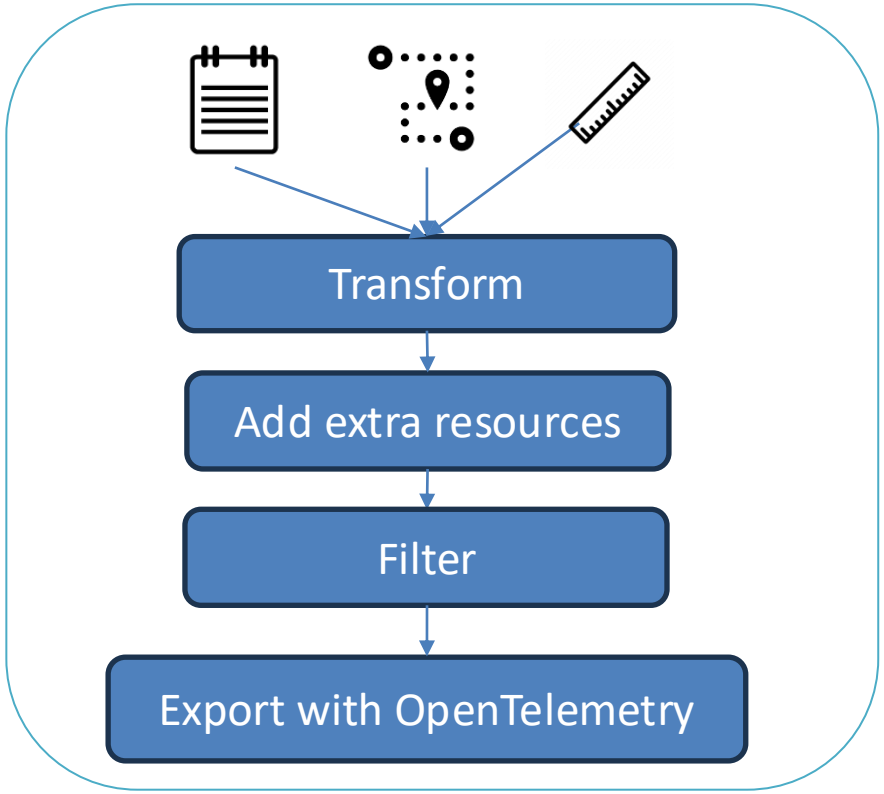
- **To process traces we mainly need the following plugins :**
  - **Enrich the trace with:**
    - **Extra resources**
    - **Kubernetes metadata**
  - **Drop**
  - **Sample**
  - **Filter**

Kubernetes		✓	
Ressource	Content_modifier	✓	✓
TailSampling	✓	✓	
probabilisticSampling	✓	✓	
Filter		✓	✓
Transform		✓	✓

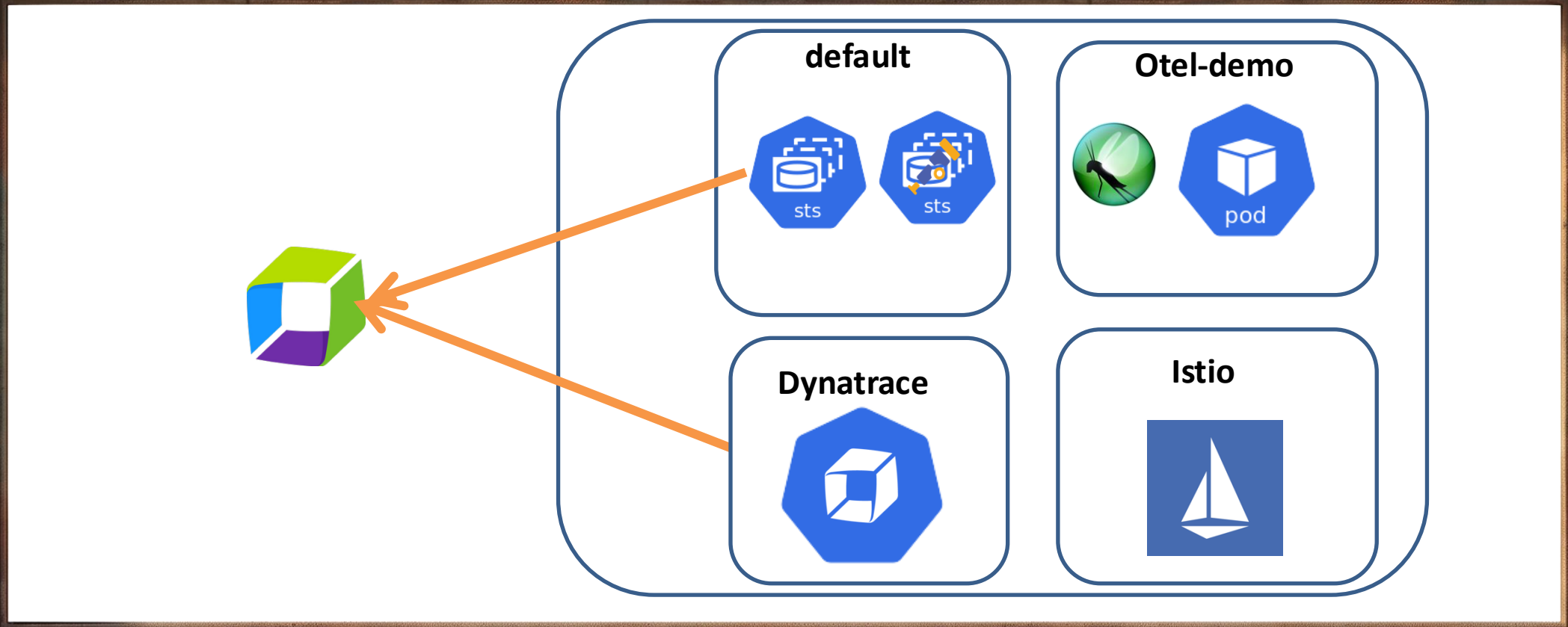

benchMark
SENATVS POPVLVSQVE ROMANVS
Throughput (msgs/s)
CLASSIC OTLP
ARROW_OTLP
Latency (ms)
95th Percentile
0.12ms
1.2ms
Arrow
original
CPU Usage (%)
Metric
Baseline OTEP (Classic)
Arrow-OTLP (Neol)
Throughput (msgs/s)
1.1M
2.4M
Arrow-OTLP (New)
2.4M
Performance Delta
+118%
METRIC
Baceline
Mlx
New
Max
12.0B
2.4M
2.4M
2.6M
9.2B
1.0M
1.1M
2.4M
0.0B
1.7M
1.6M
2.4M
12.B
1.7M
2.5M
2.4M
1.1B
1.2M
2.4M
2.4M
1.0B
3.9M
8.7M
+118%

The pipeline

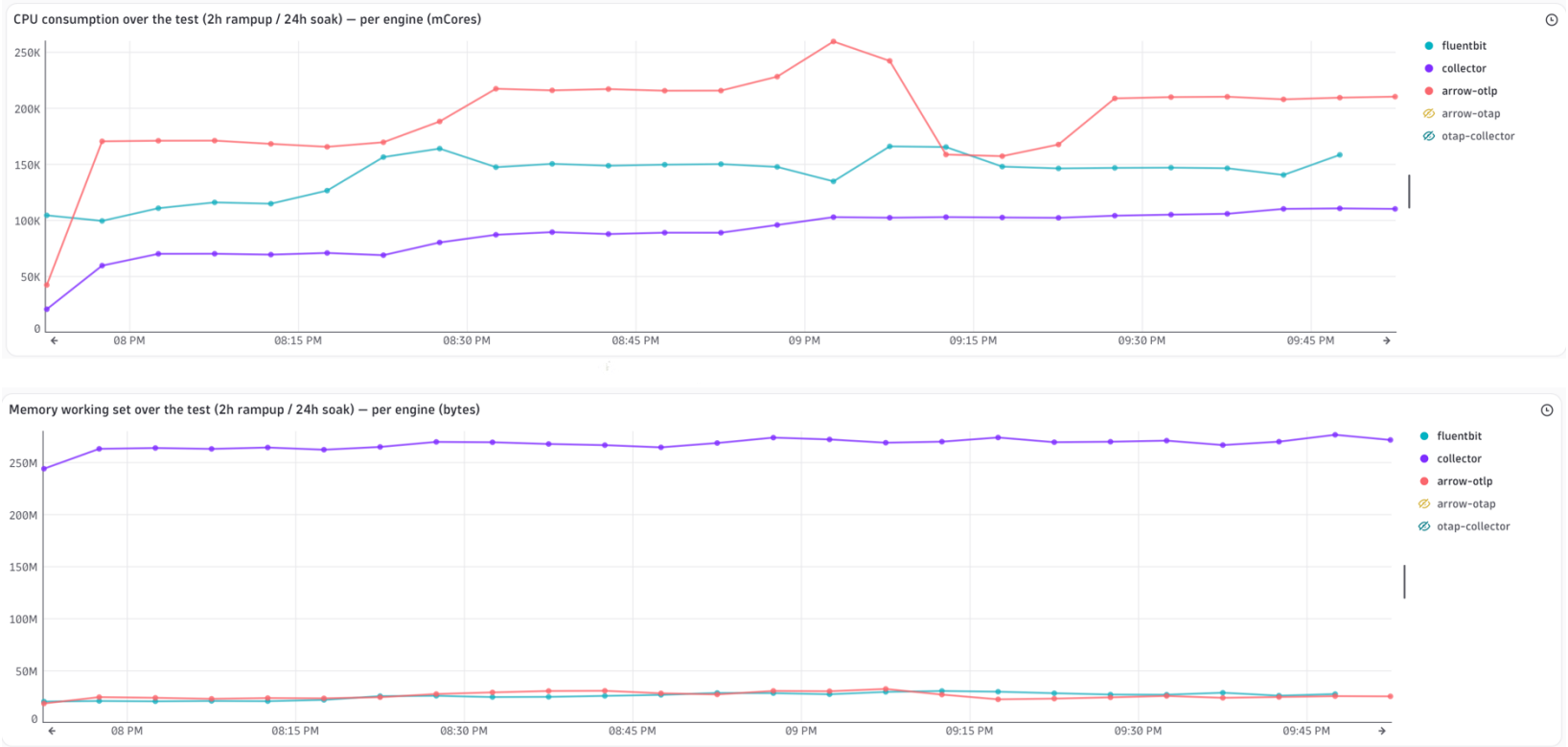
- Use features supported by each engine:
  - Receive otlp
  - Transform
  - Filter
  - Enrich
  - Export



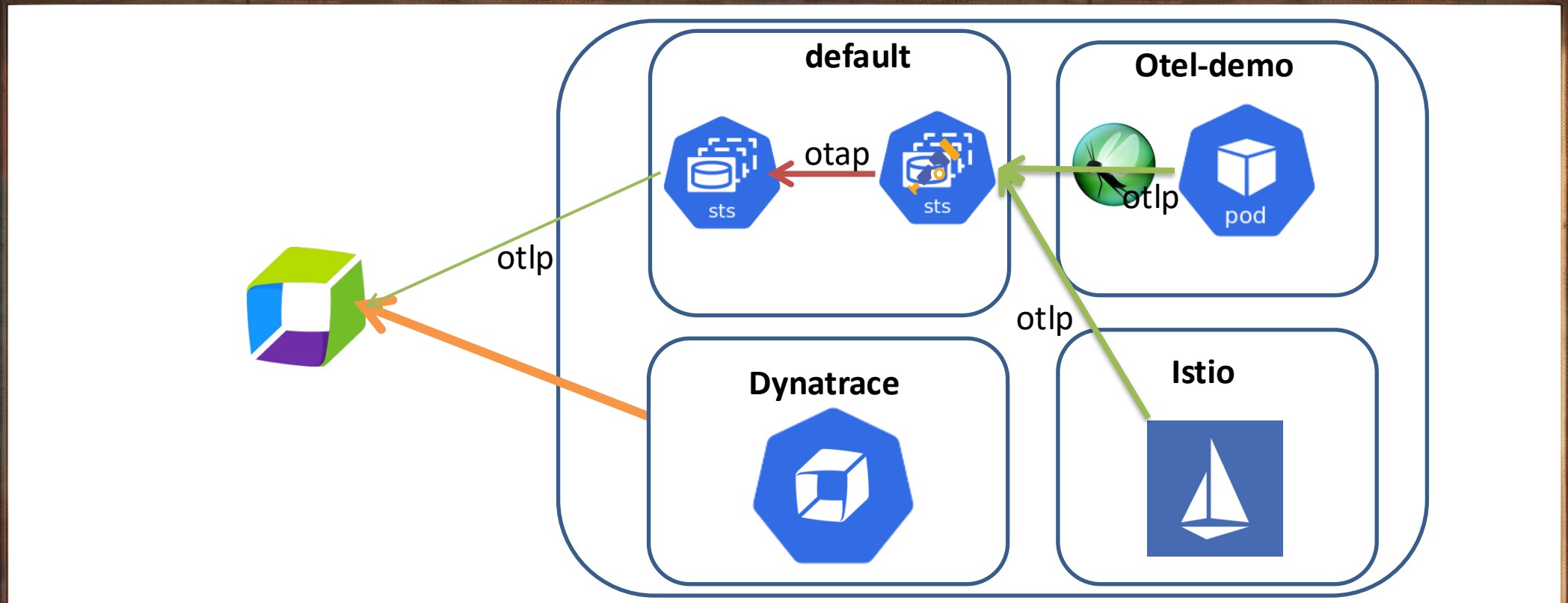
Architecture



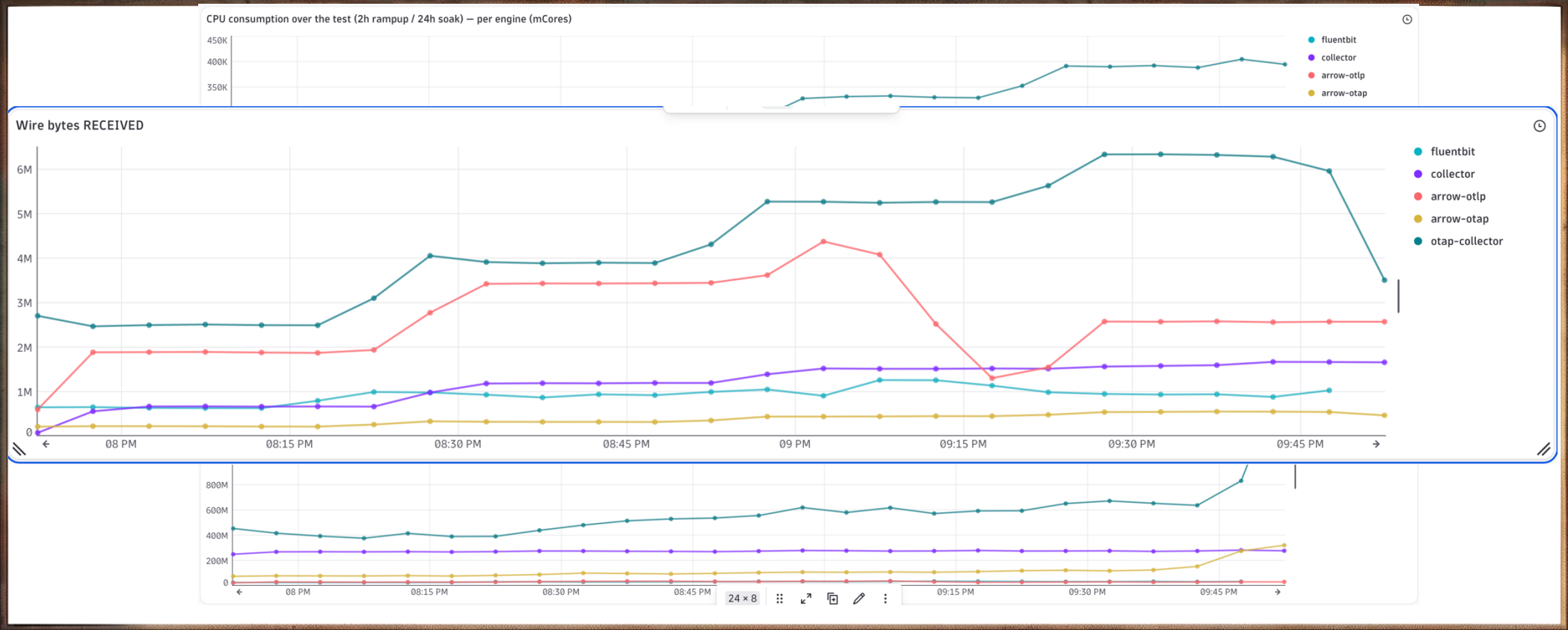
Rampup Tests



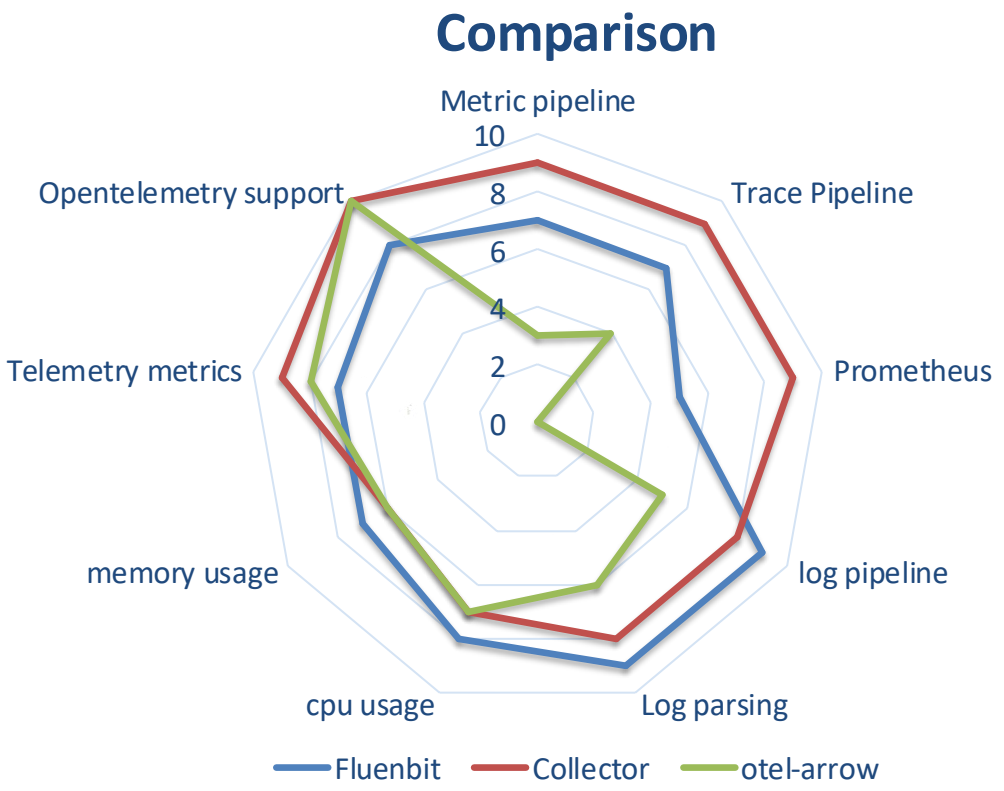
Tests using OTAP



Rampup Tests



Conclusion


Comparison
Metric pipeline
Trace Pipeline
Prometheus
log pipeline
Log parsing
cpu usage
memory usage
Telemetry metrics
Opentelemetry support
10
8
6
4
2
0
Fluenbit
Collector
otel-arrow

Is It Observable

- Looking for educational content on Observability , Checkout the YouTube Channel :













IS IT OBSERVABLE?

Thank you

