FROM USE CASE TO PRODUCTION

KAFKA STREAMS

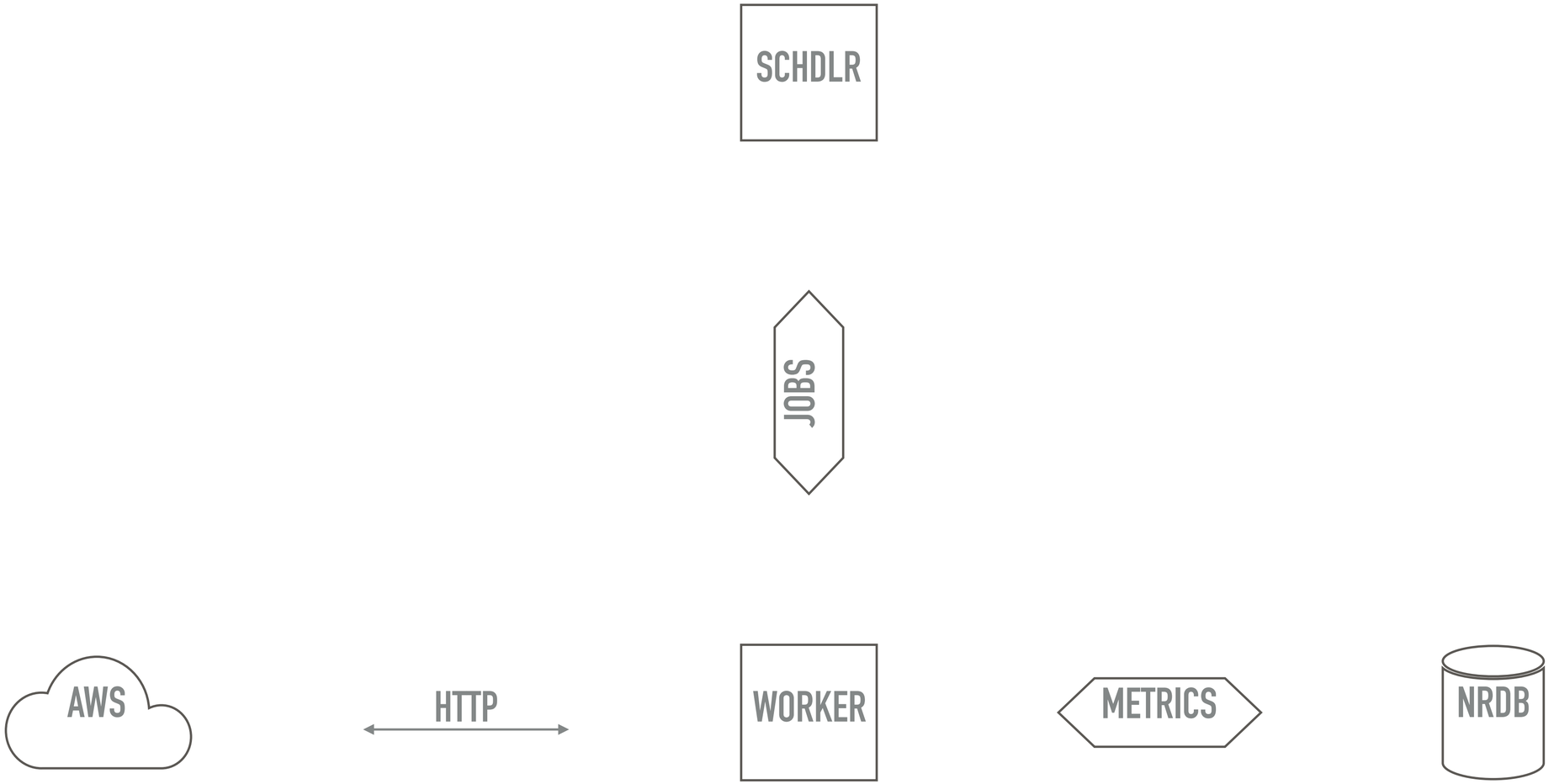
CLOUD MONITORING

CLOUD MONITORING

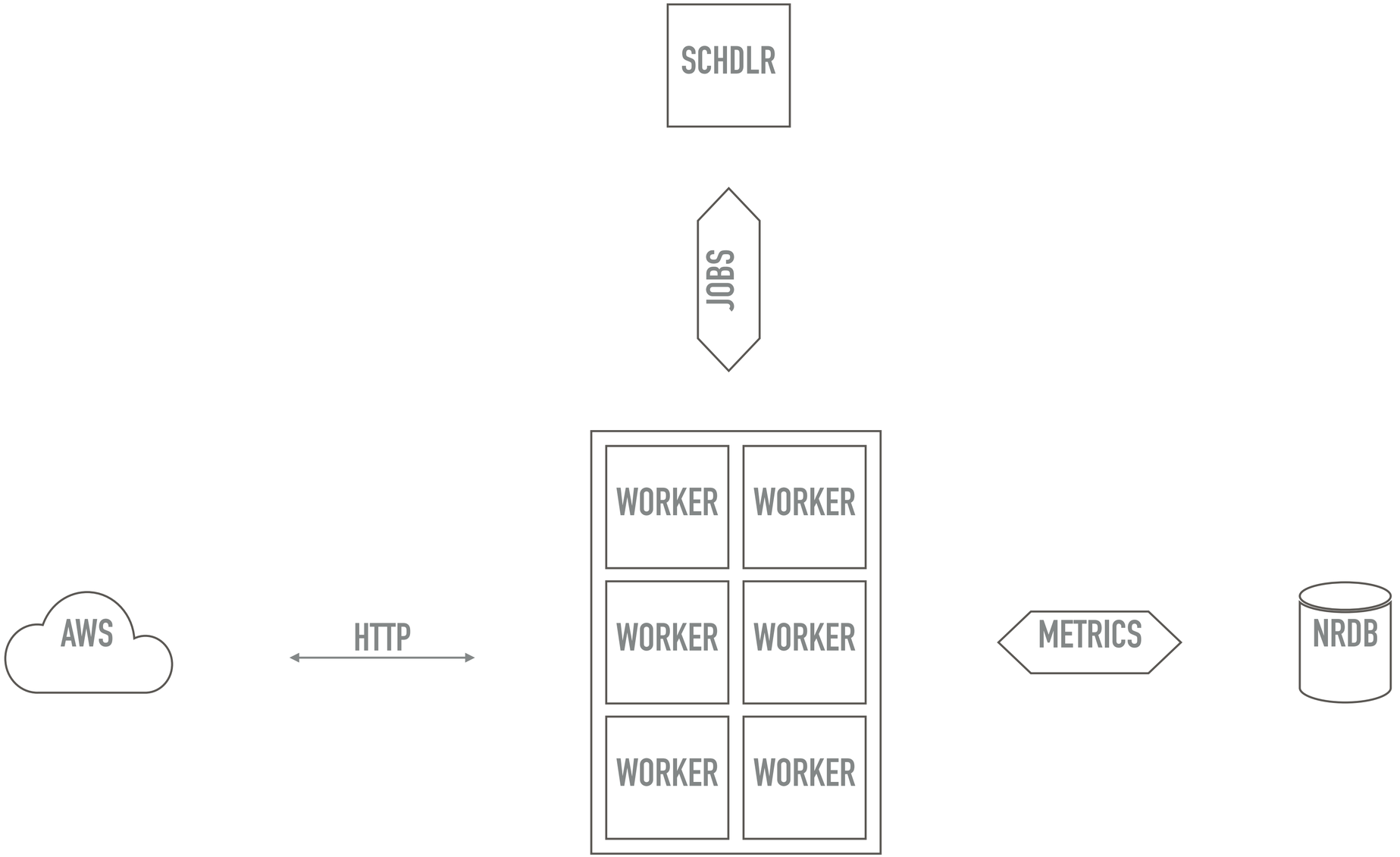
CLOUD MONITORING

CLOUD MONITORING

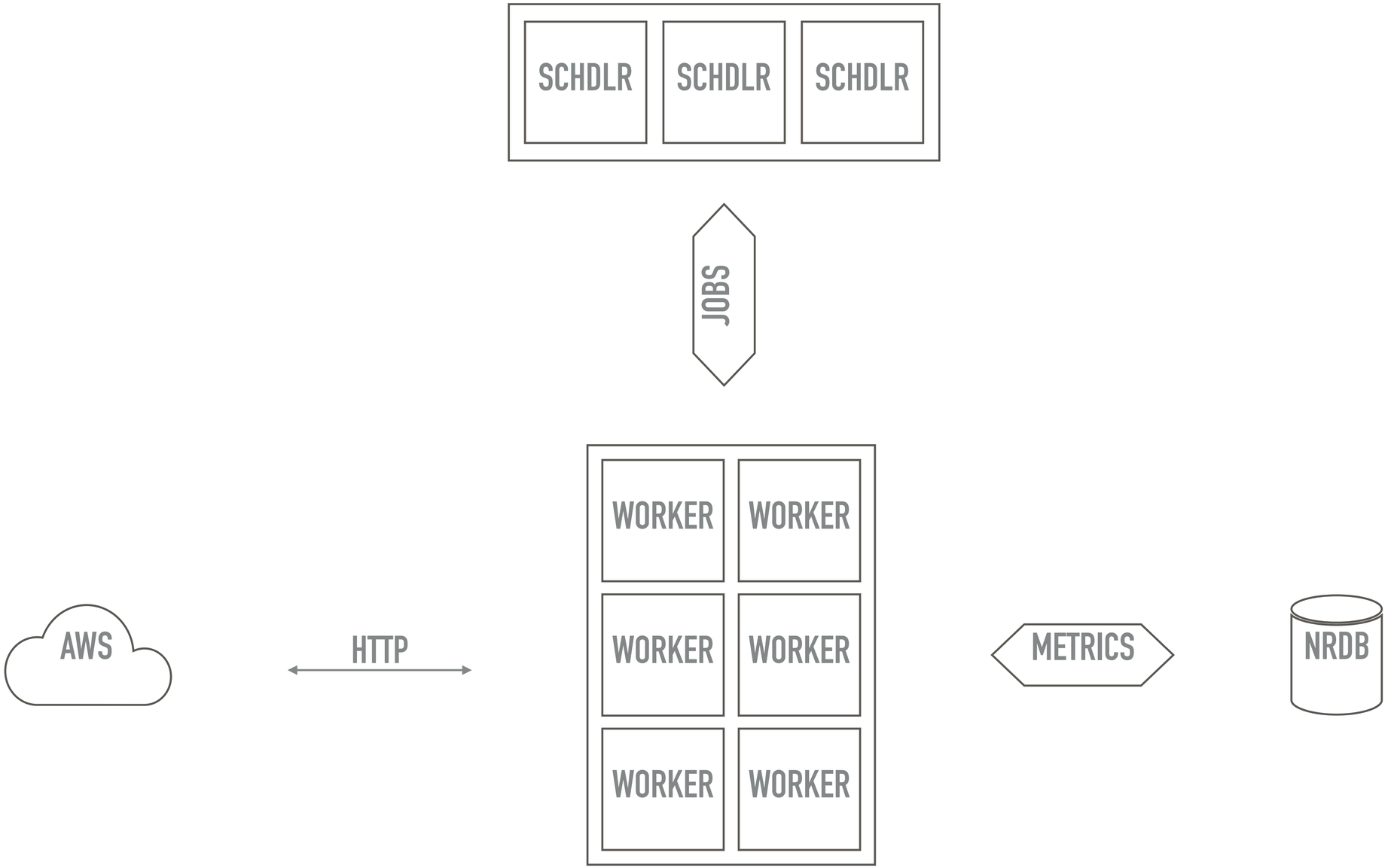
CLOUD MONITORING



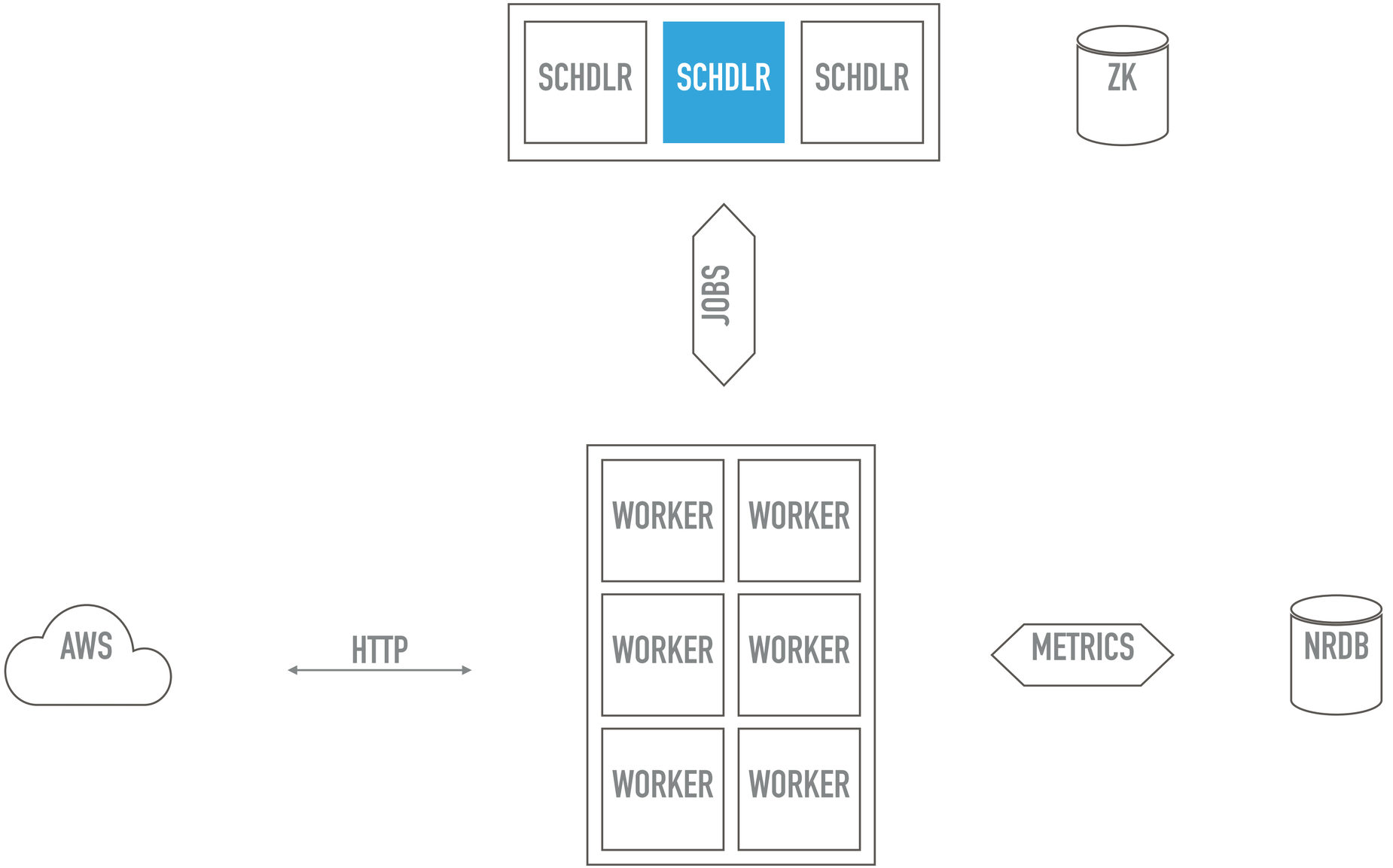
CLOUD MONITORING



CLOUD MONITORING

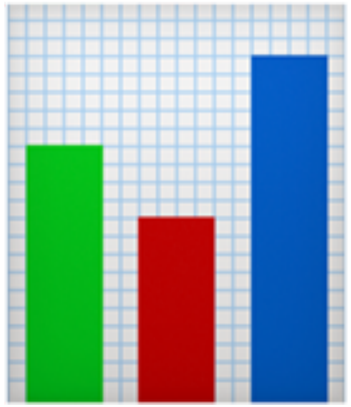


CLOUD MONITORING



CLOUD MONITORING

- horizontally scalable
- stateless
- failsafe
- a few Kafka topics



METRICS?

WHAT METRICS?

WHAT METRICS?

WHAT METRICS?

```
{
    "id": 1,
    “timestamp": 5,
    "max.cpu": 10
}
```

WHAT METRICS?

```
{
    "id": 1,
    “timestamp": 5,
    "max.cpu": 10
}
```

WHAT METRICS?

```
{
    "id": 1,
    “timestamp": 0,
    "max.cpu": 10
}
```

WHAT METRICS?

```
{
    "id": 1,
    “timestamp": 0,
    "max.cpu": 10
}

{
    "id": 1,
    "timestamp": 60,
    "max.cpu": 20
}
```

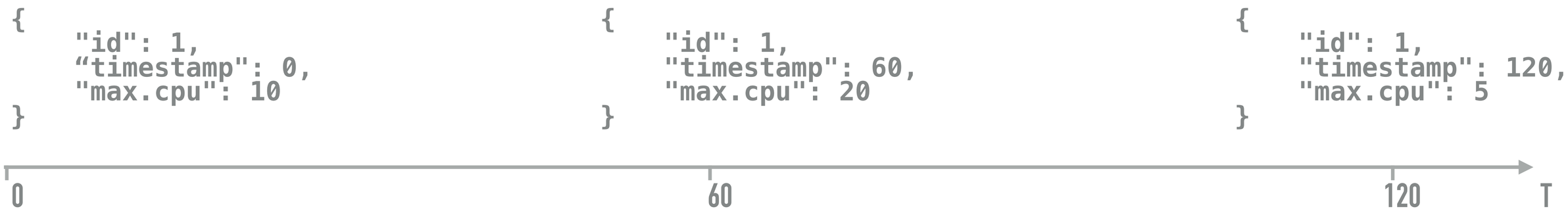
WHAT METRICS?

```
{
    "id": 1,
    “timestamp": 0,
    "max.cpu": 10
}

{
    "id": 1,
    "timestamp": 60,
    "max.cpu": 20
}

{
    "id": 1,
    "timestamp": 120,
    "max.cpu": 5
}
```

AGGREGATION



AGGREGATION

```
{
    "id": 1,
    "timestamp": 3600,
    "max.cpu": 20
}
```

STATE!

AGGREGATION

AGGREGATION

AGGREGATION

- batch?
- local storage?
- summon Redis?

APACHE kafka® Streams

AGGREGATION

✓ exactly once

✓ stateful (local state with failsafe mechanism)

? own cluster vs managed cluster

? framework vs library

AGGREGATION

AGGREGATION

AGGREGATION

AGGREGATION

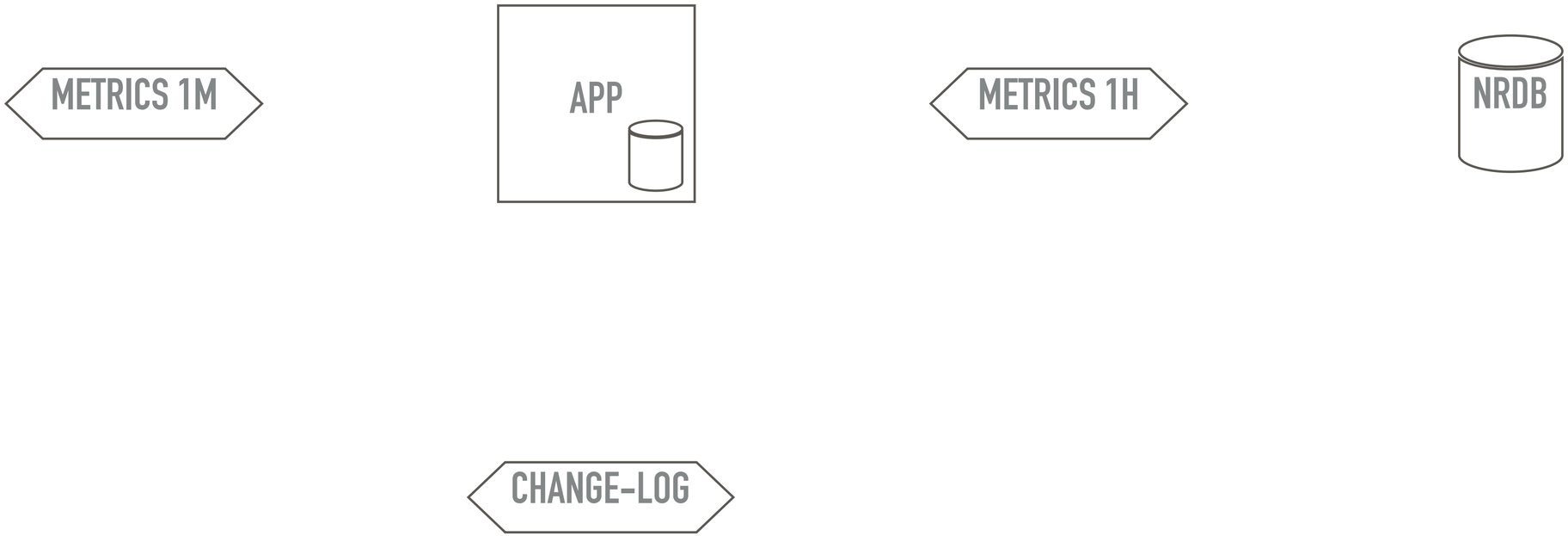
- same deployment mechanism
- no new external dependencies

STATE!

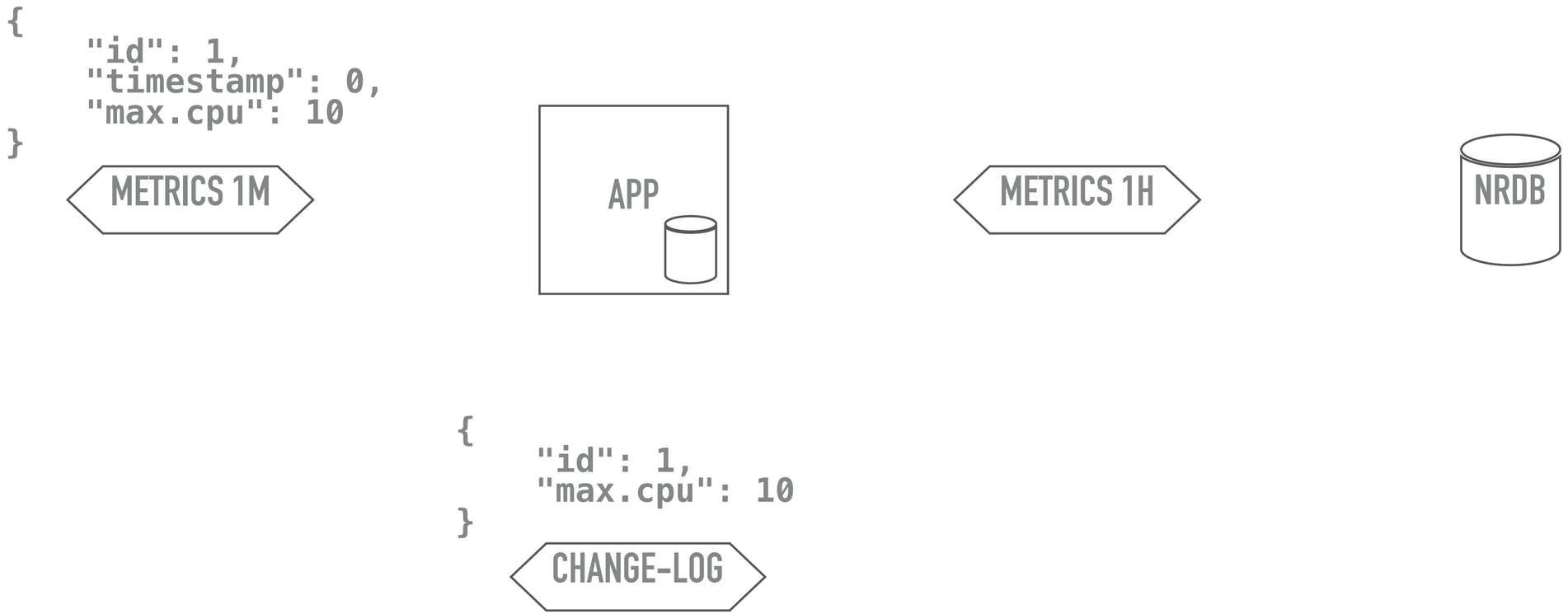
AGGREGATION

AGGREGATION

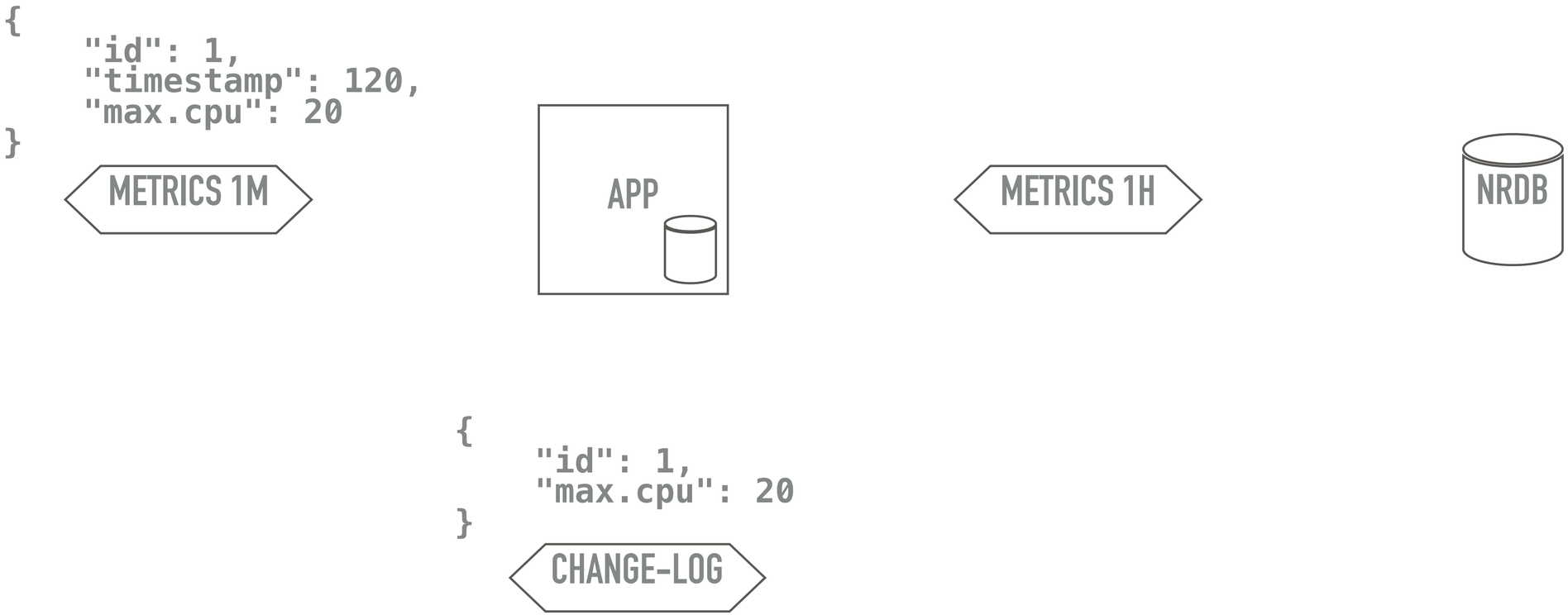
AGGREGATION



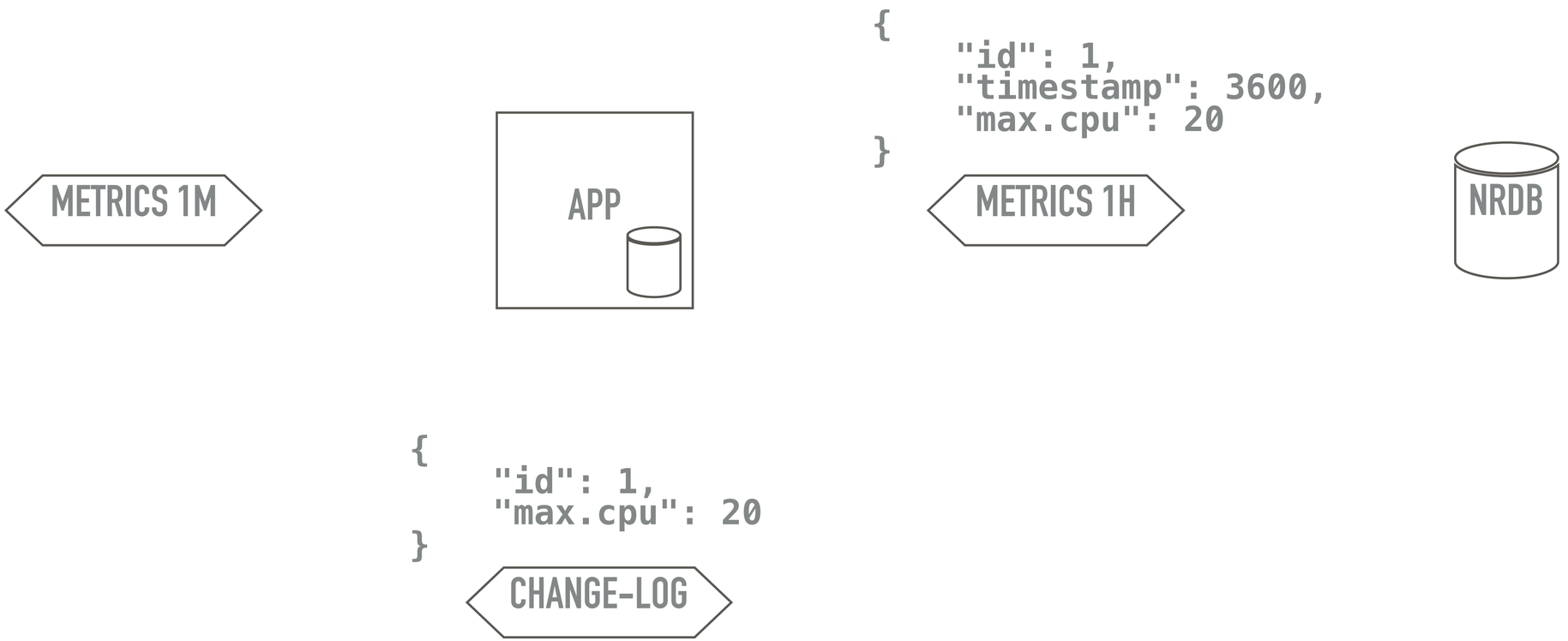
AGGREGATION



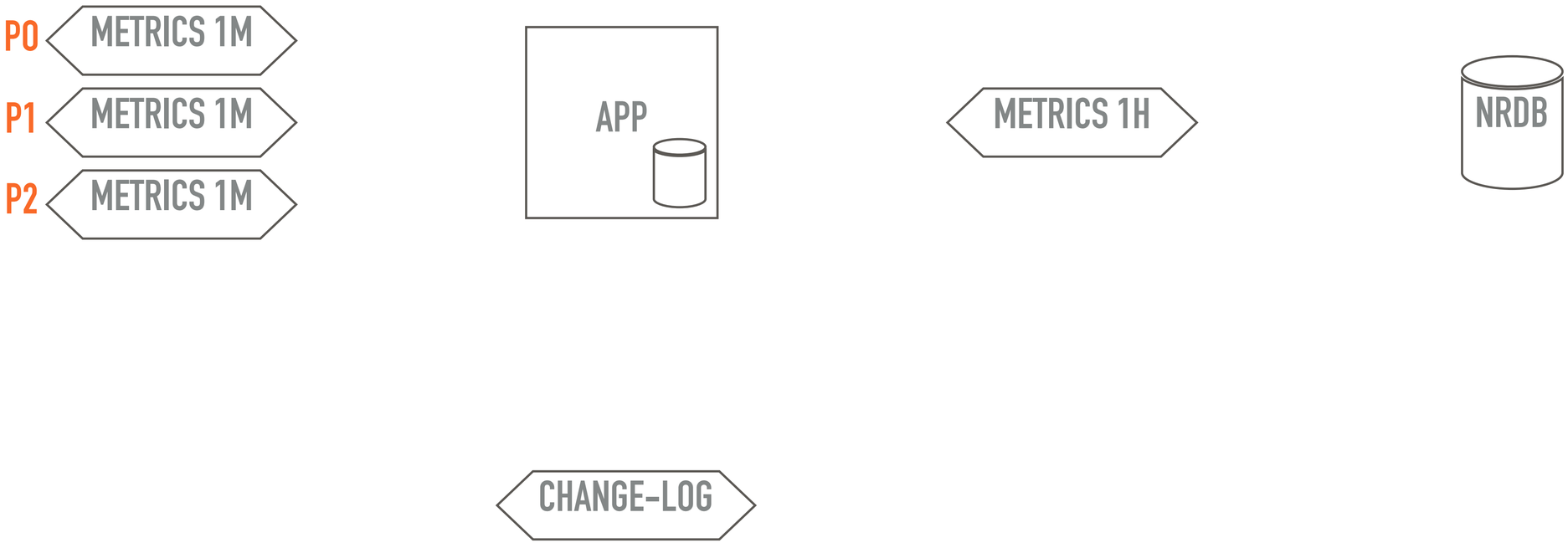
AGGREGATION



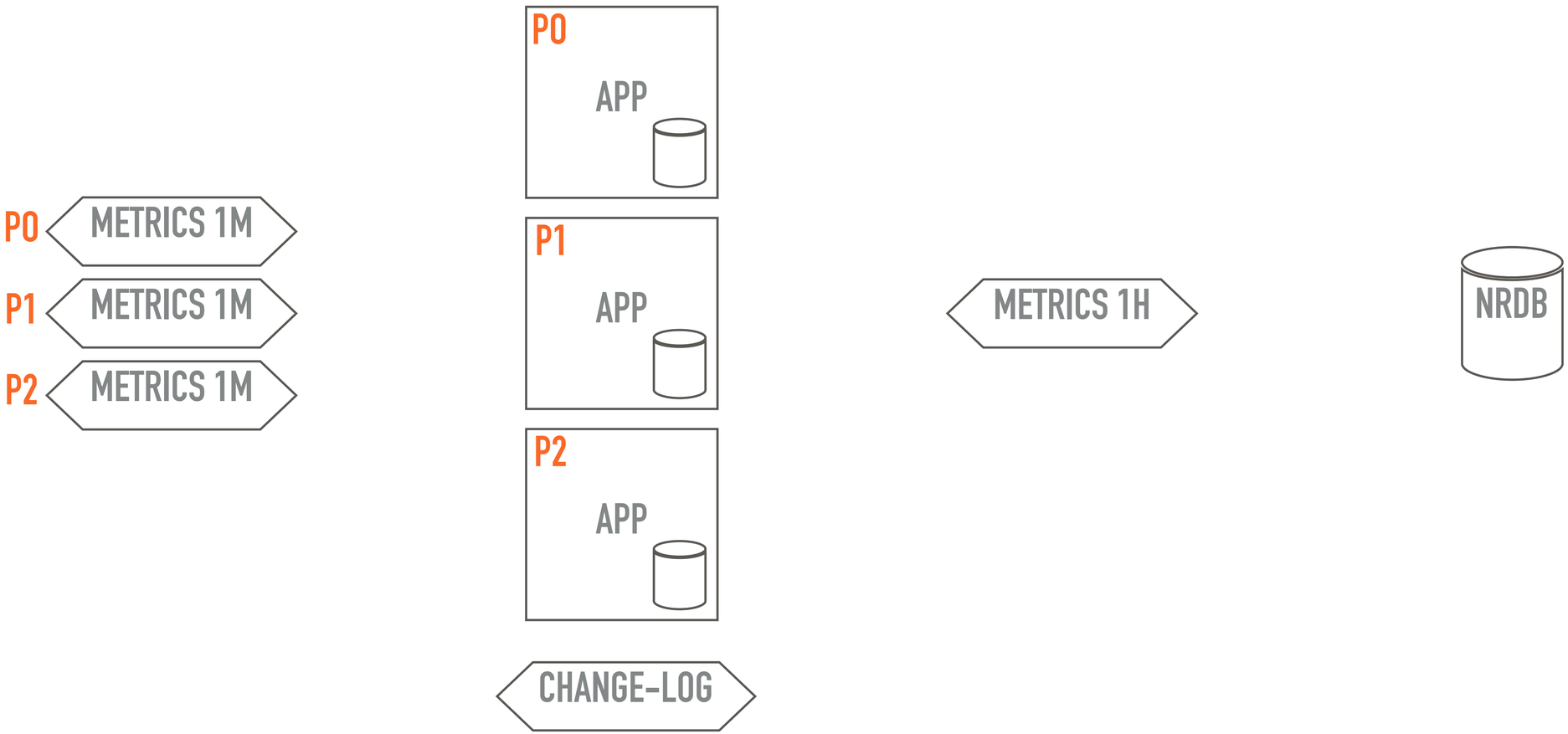
AGGREGATION



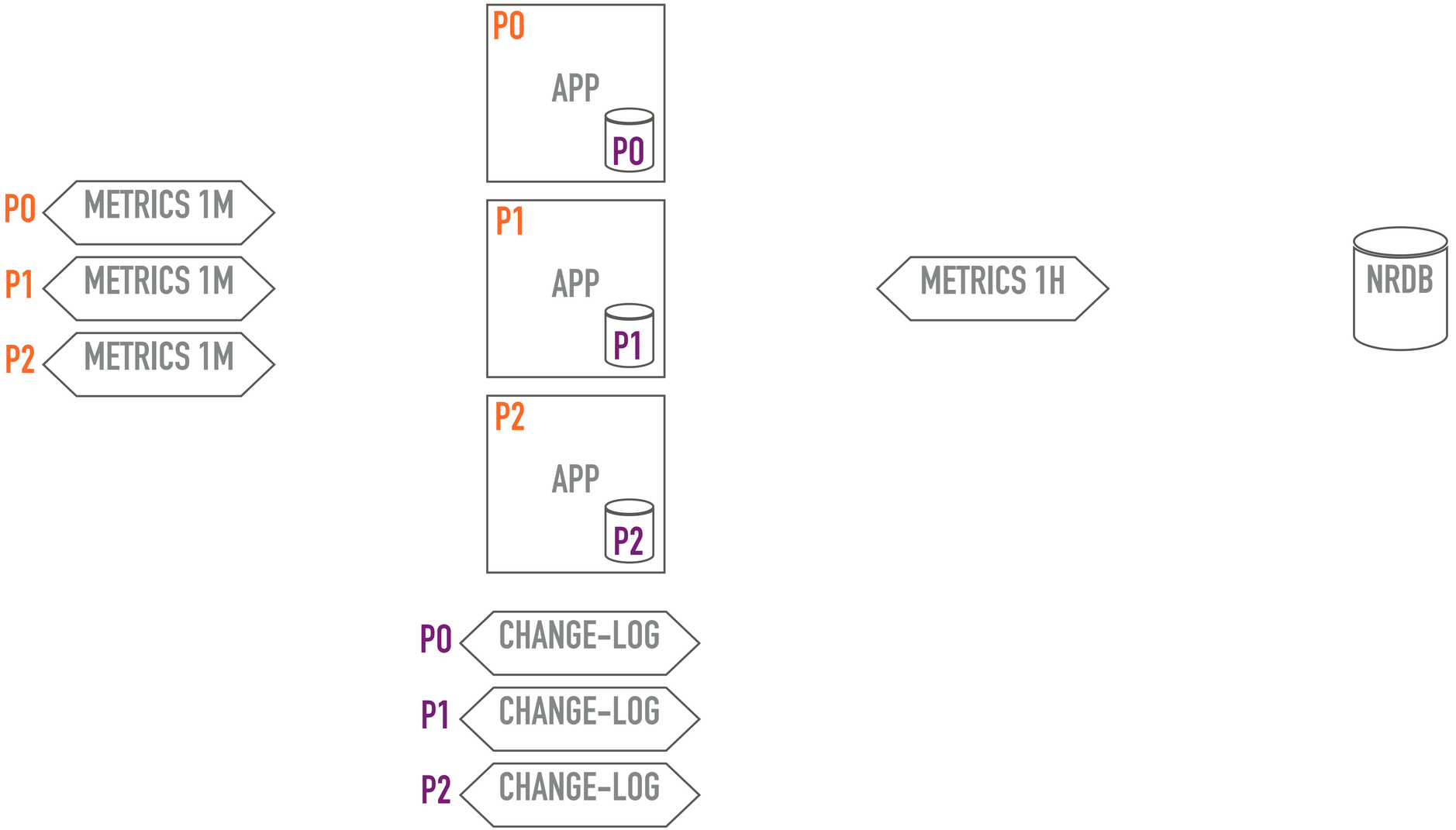
AGGREGATION



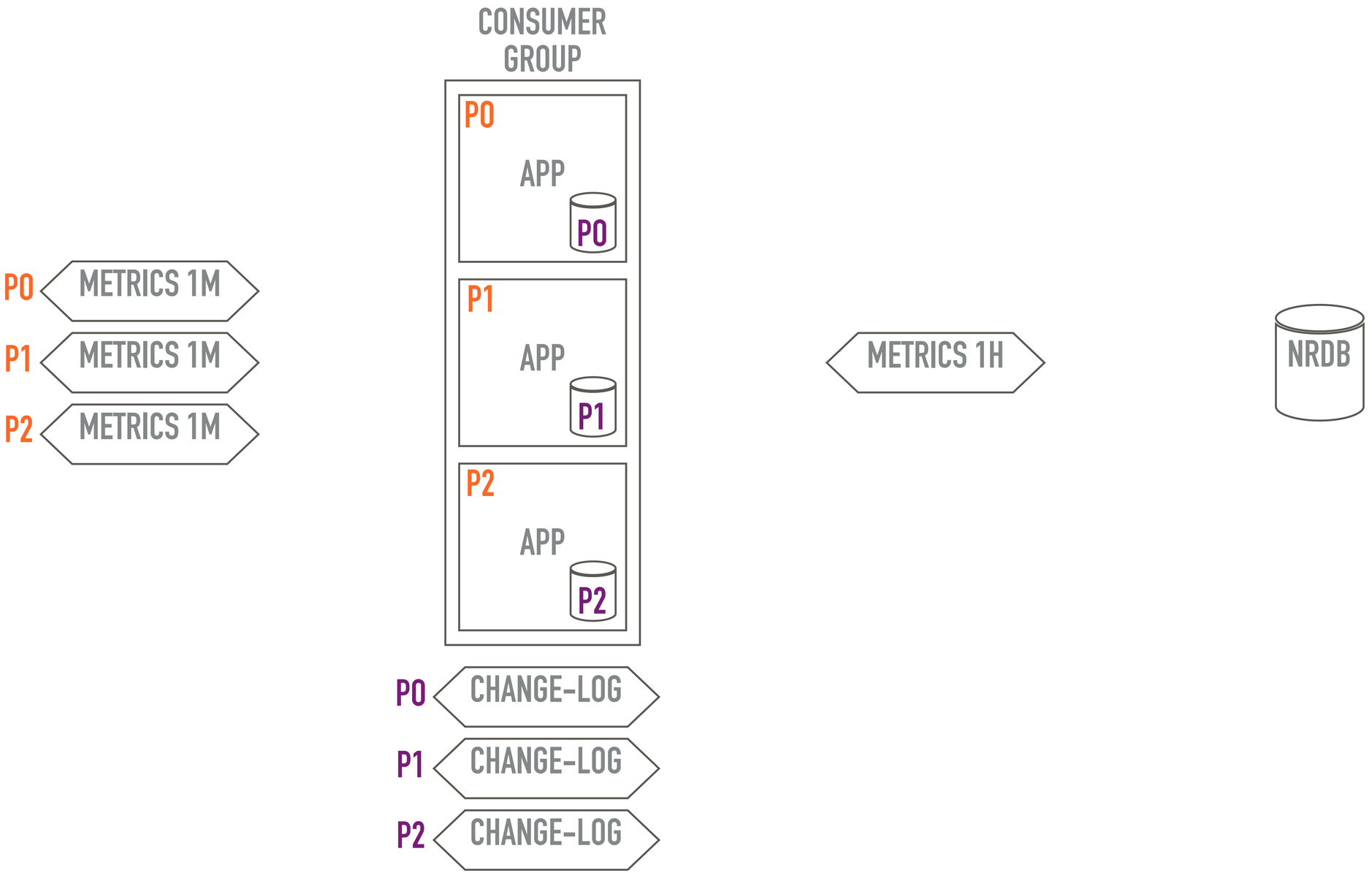
AGGREGATION



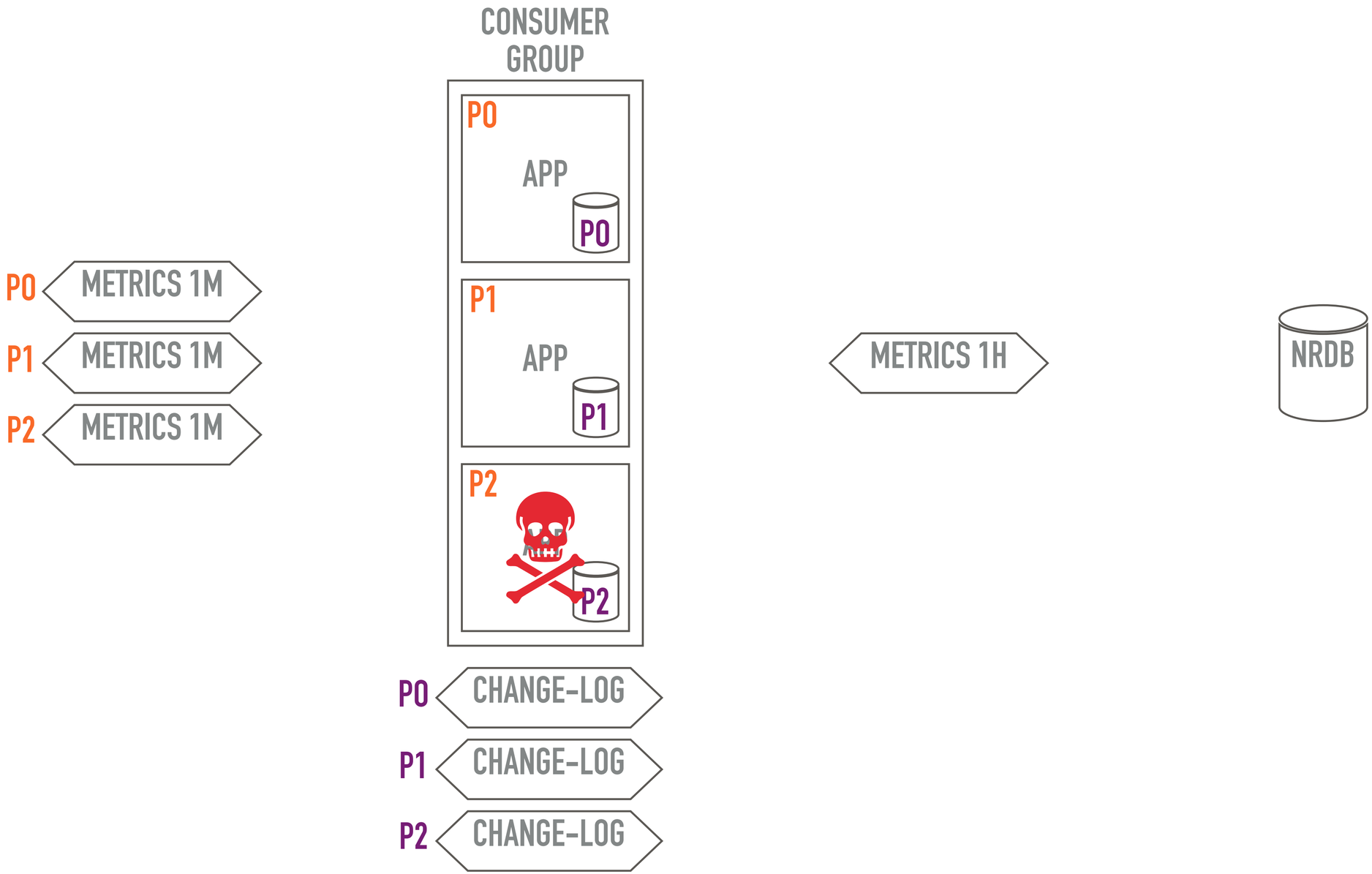
AGGREGATION



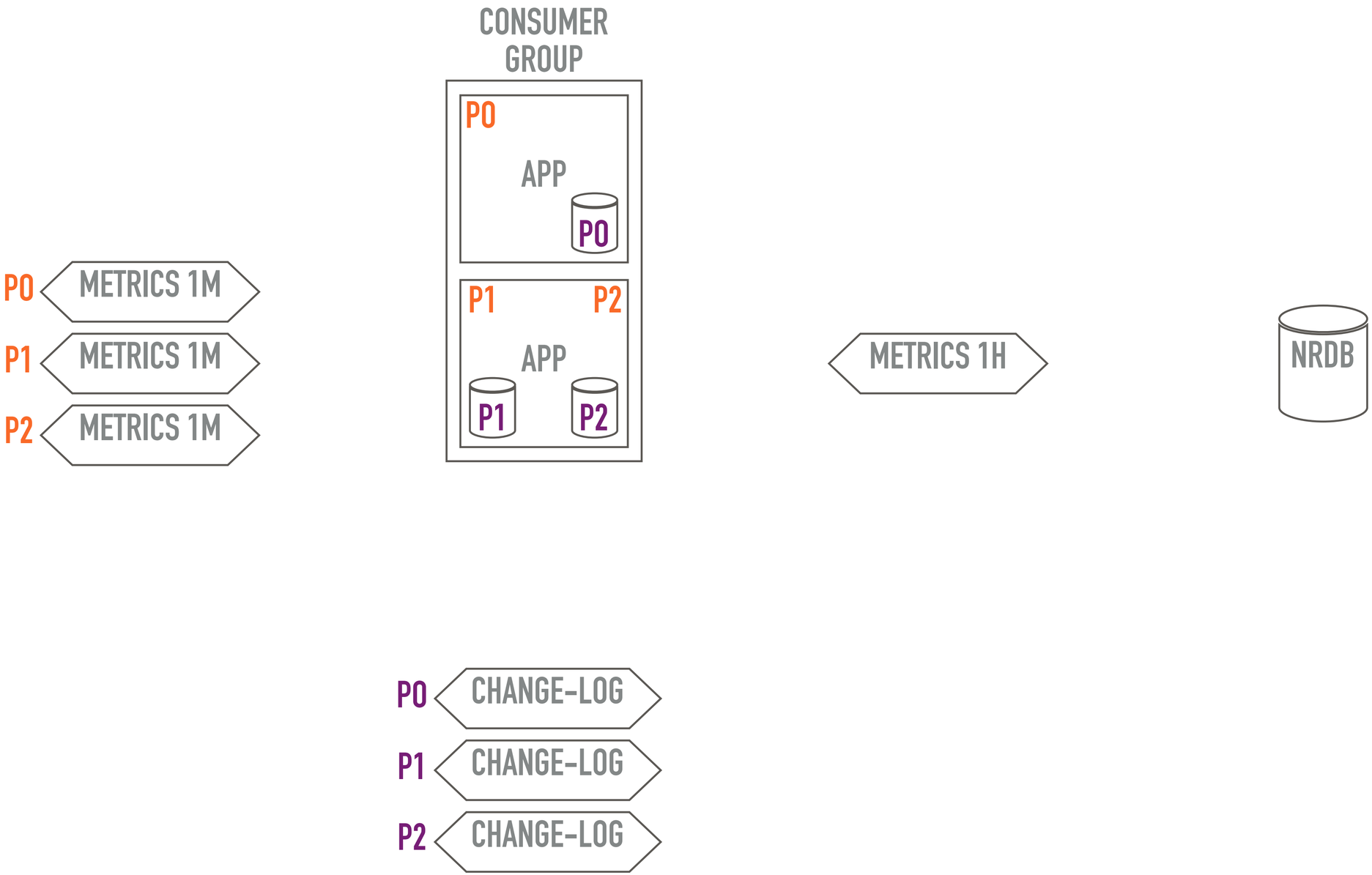
AGGREGATION



AGGREGATION



AGGREGATION

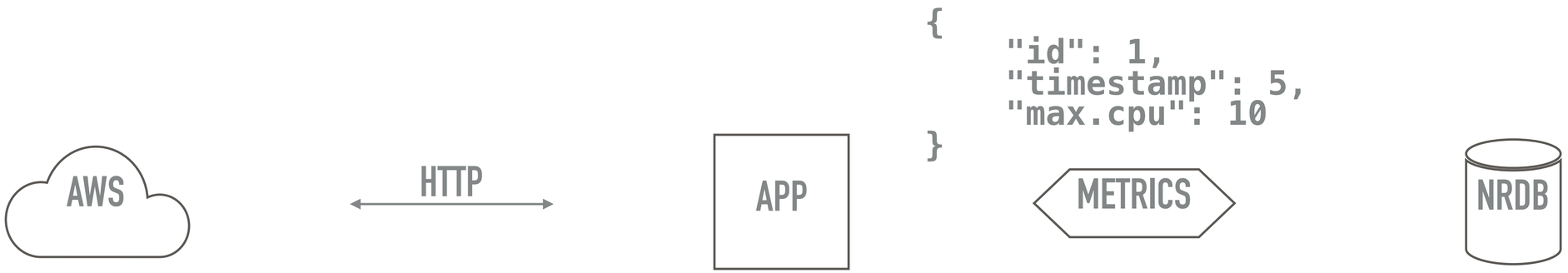


AGGREGATION

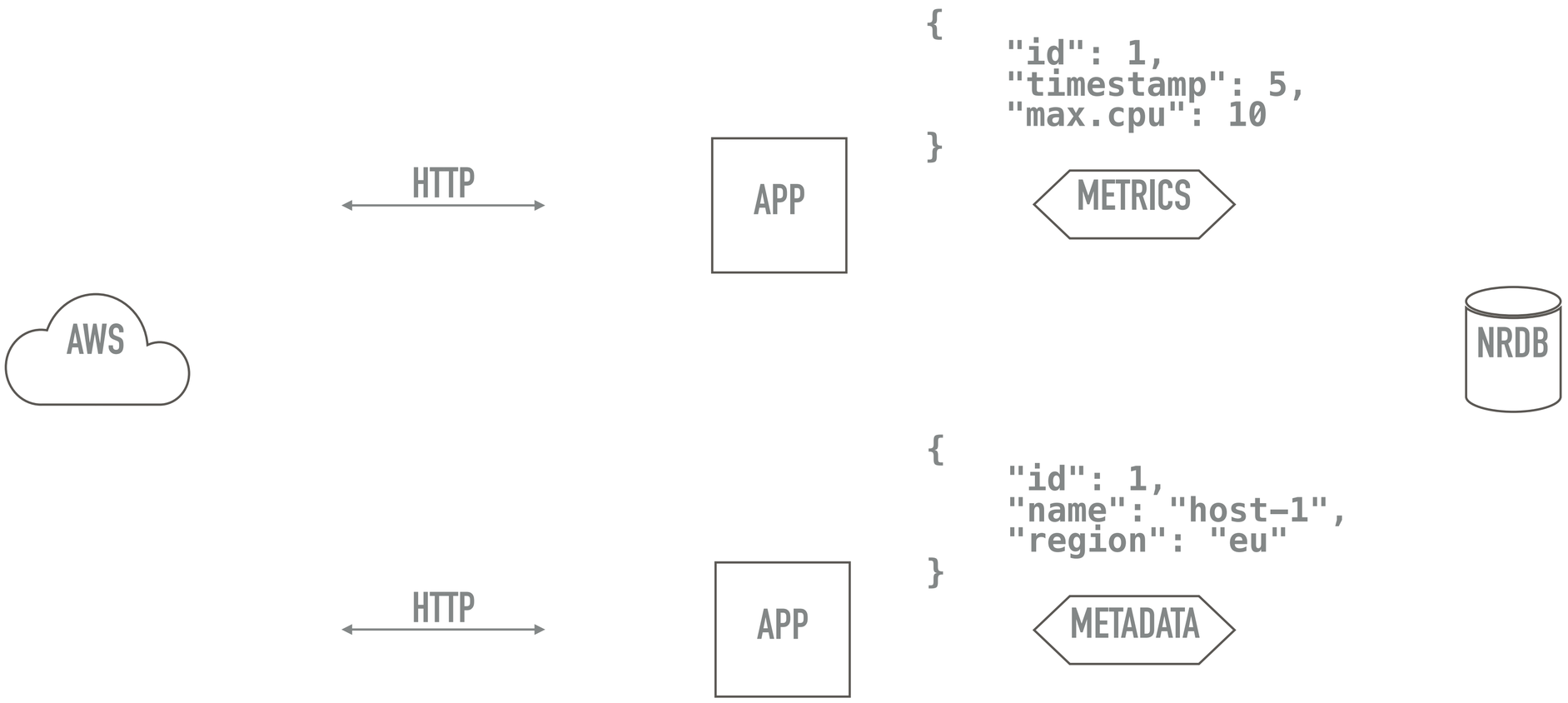
- key-value storage for stateful computations, failsafe
- time windows calculation
- scalable with number of partitions
- a bunch of new Kafka topics

SUCCESS

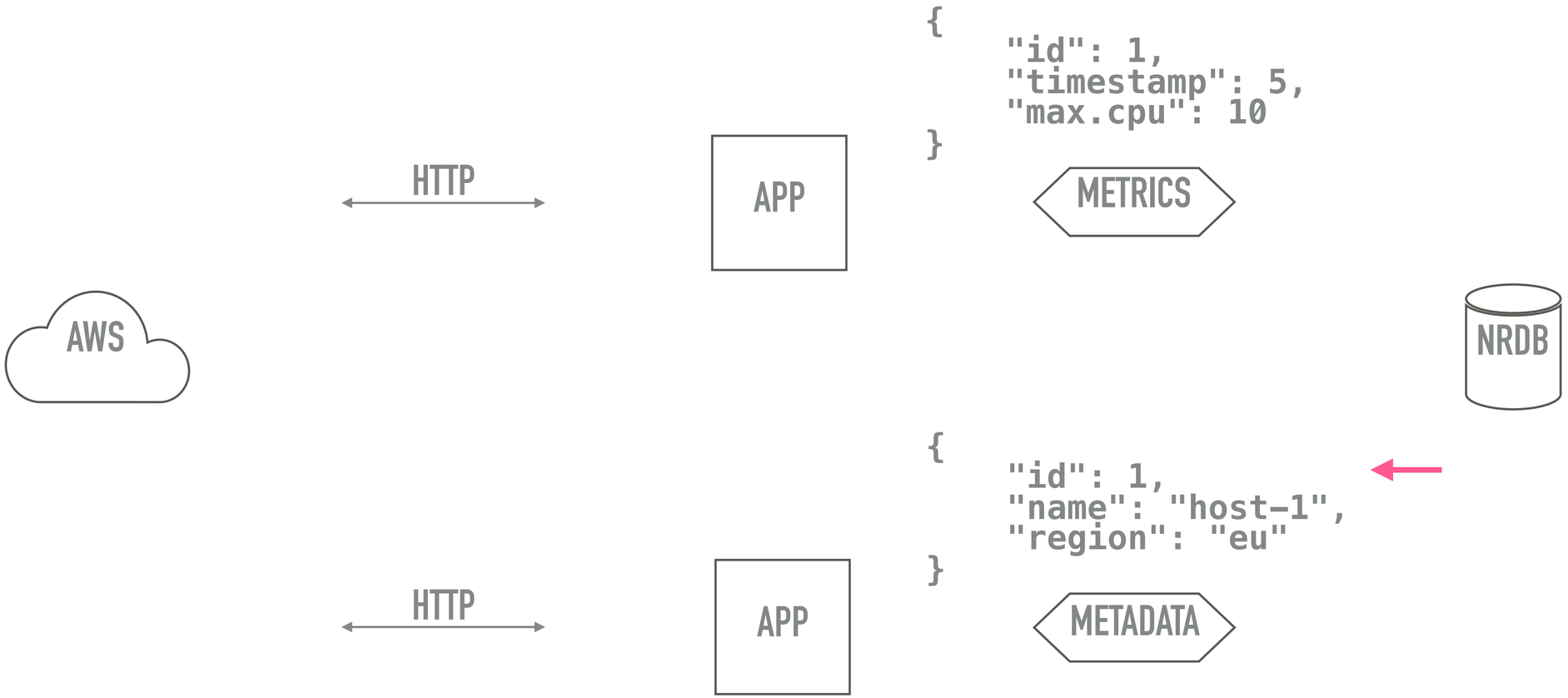
ENRICHMENT



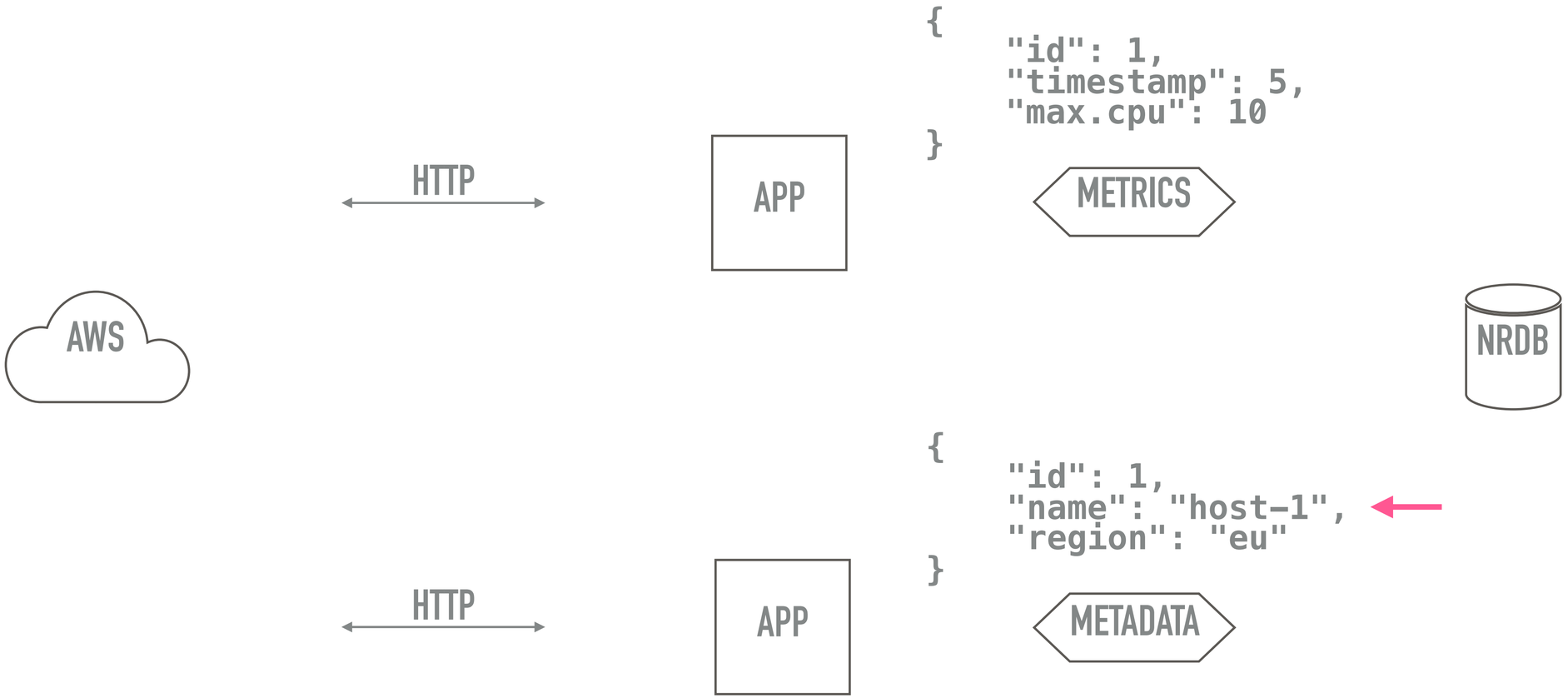
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
HTTP
APP
METRICS
AWS
NRDB
{
"id": 1,
"name": "host-1",
"region": "eu"
}
HTTP
APP
METADATA

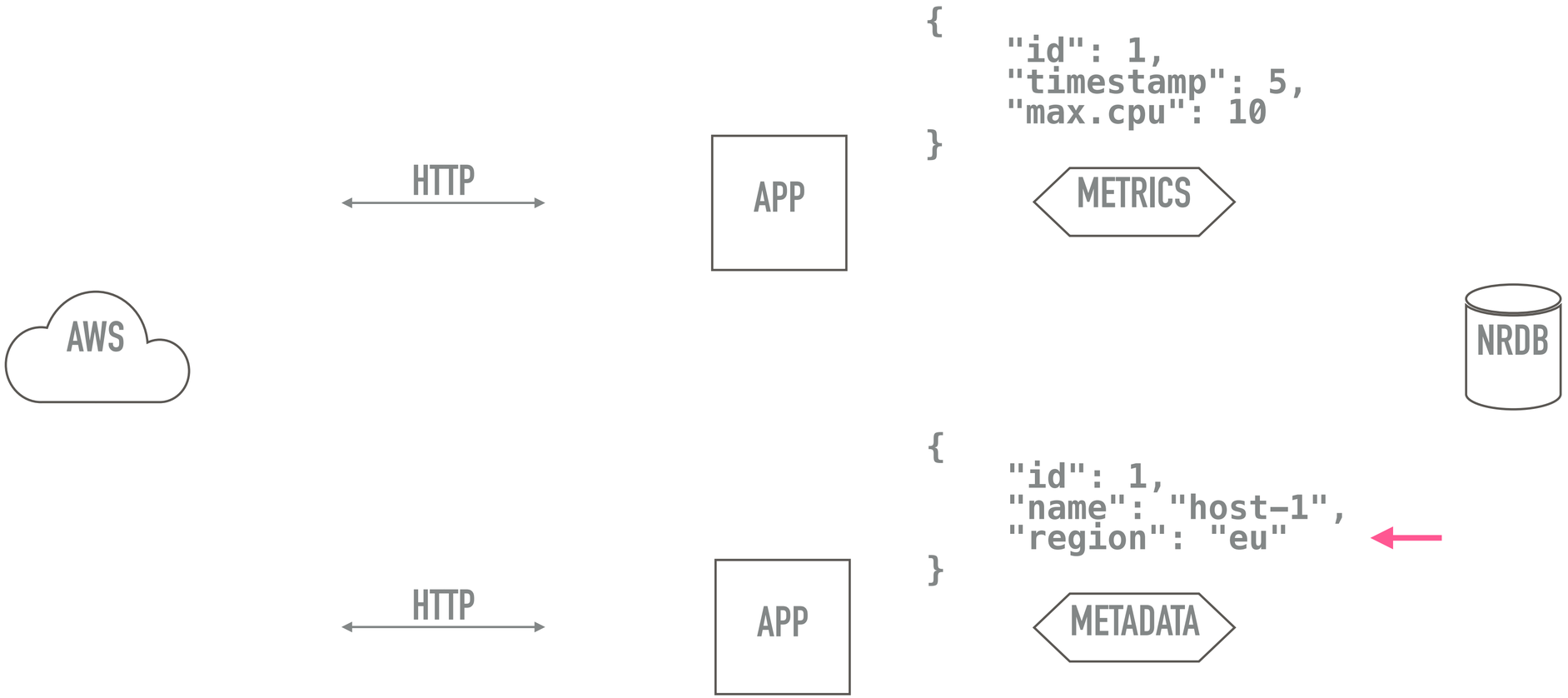
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
HTTP
APP
METRICS
AWS
NRDB
{
"id": 1,
"name": "host-1",
"region": "eu"
}
HTTP
APP
METADATA

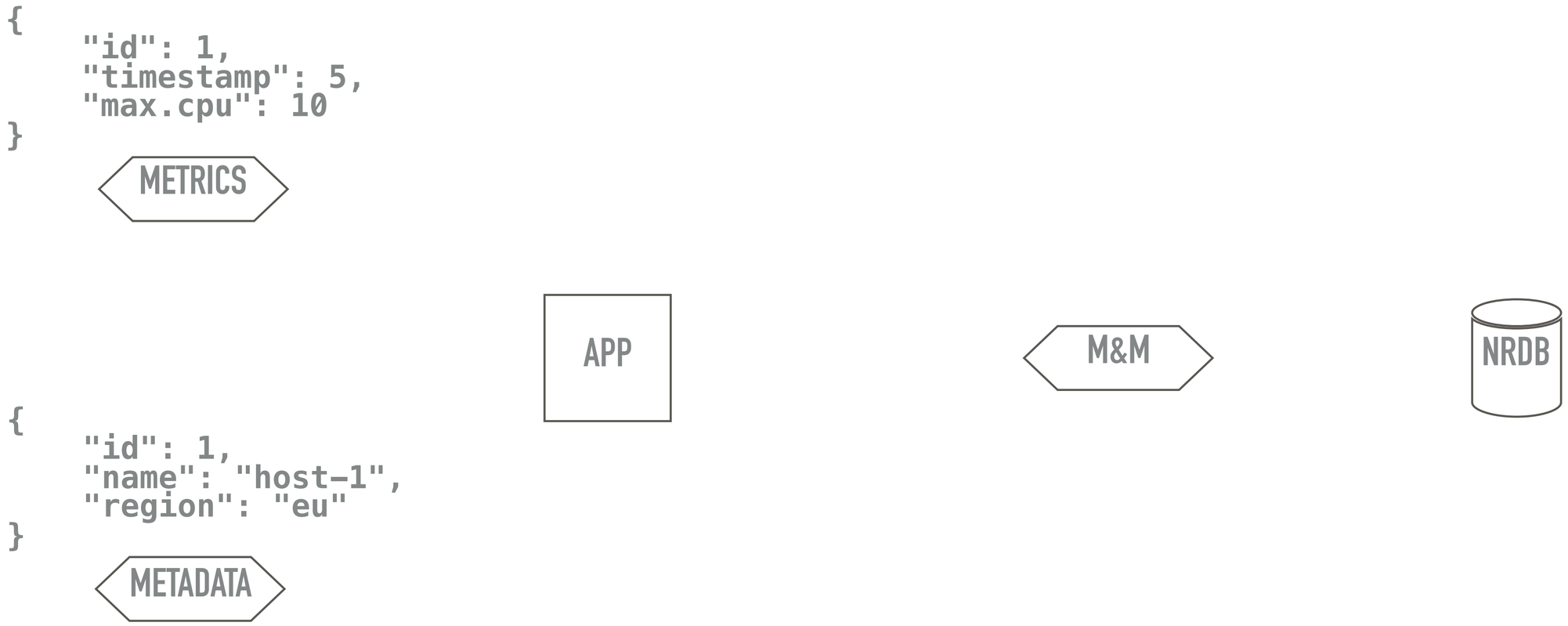
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
HTTP
APP
METRICS
AWS
NRDB
{
"id": 1,
"name": "host-1",
"region": "eu"
}
HTTP
APP
METADATA

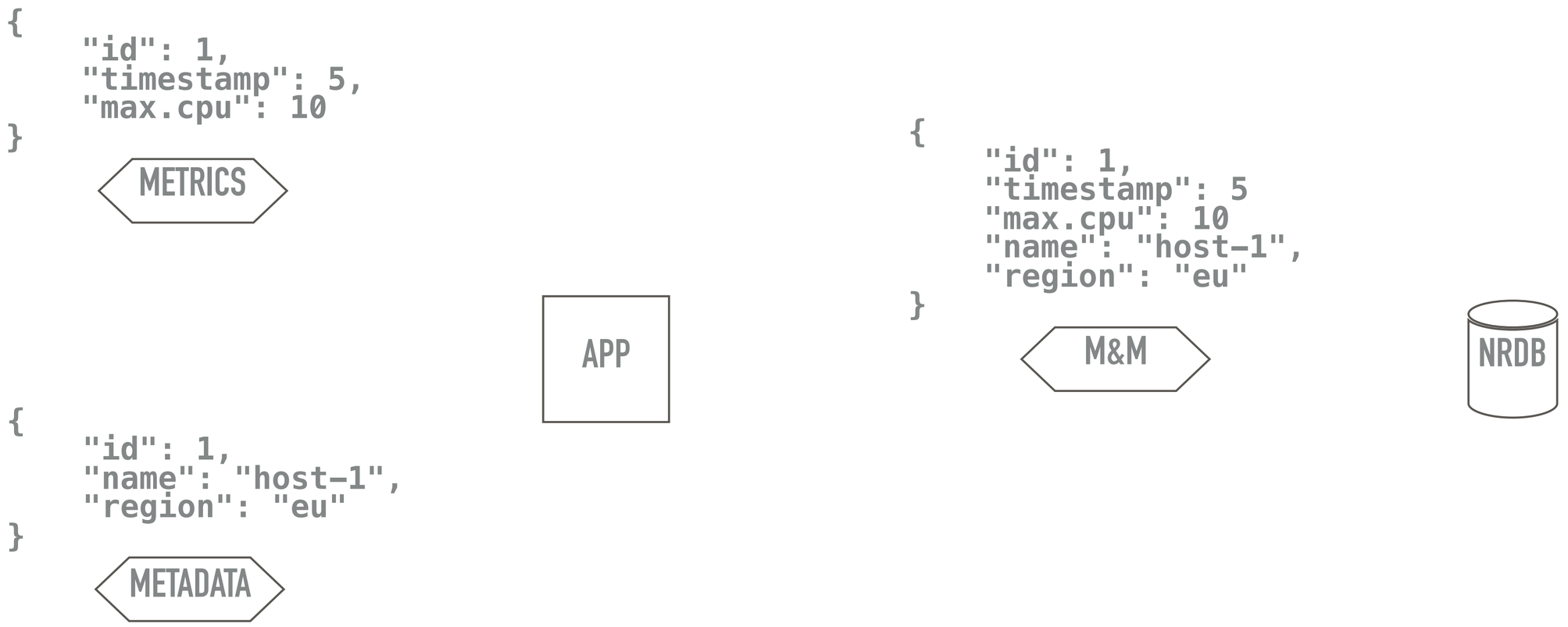
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
HTTP
APP
METRICS
AWS
NRDB
{
"id": 1,
"name": "host-1",
"region": "eu"
}
HTTP
APP
METADATA

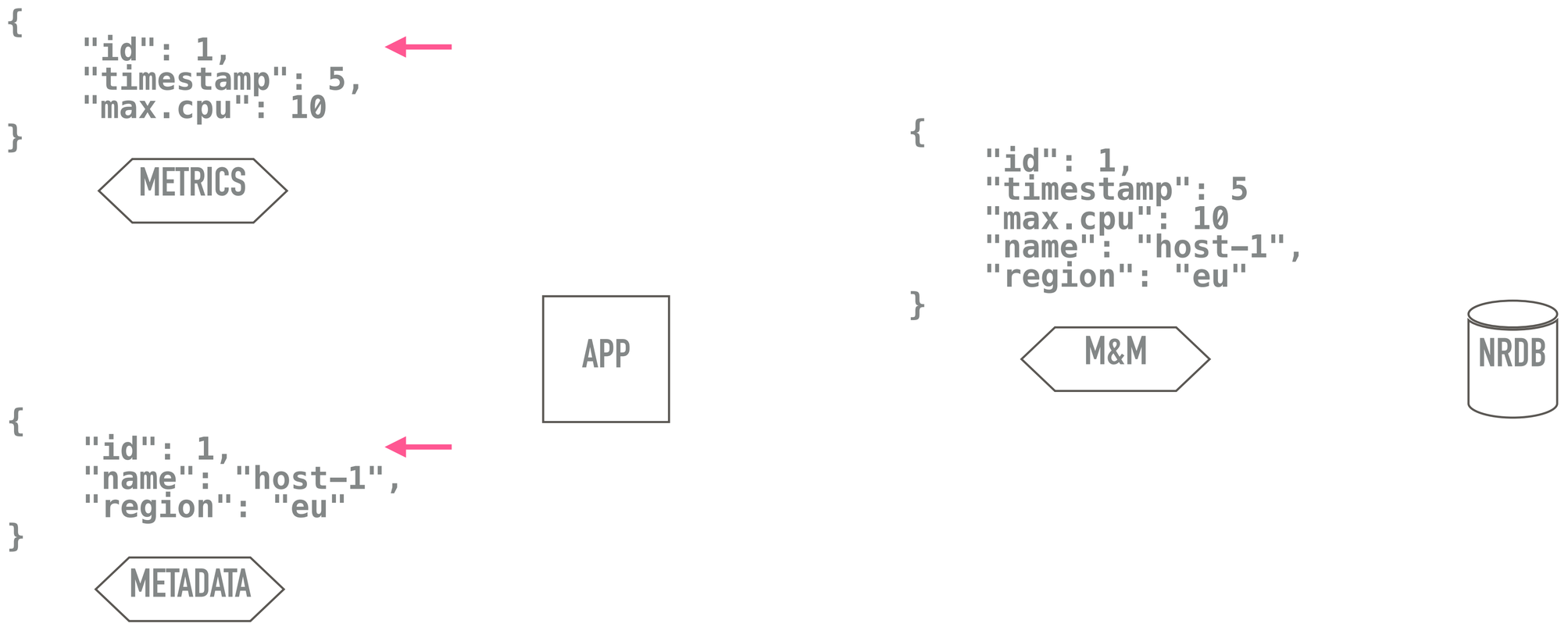
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
METRICS
APP
M&M
NRDB
{
"id": 1,
"name": "host-1",
"region": "eu"
}
METADATA

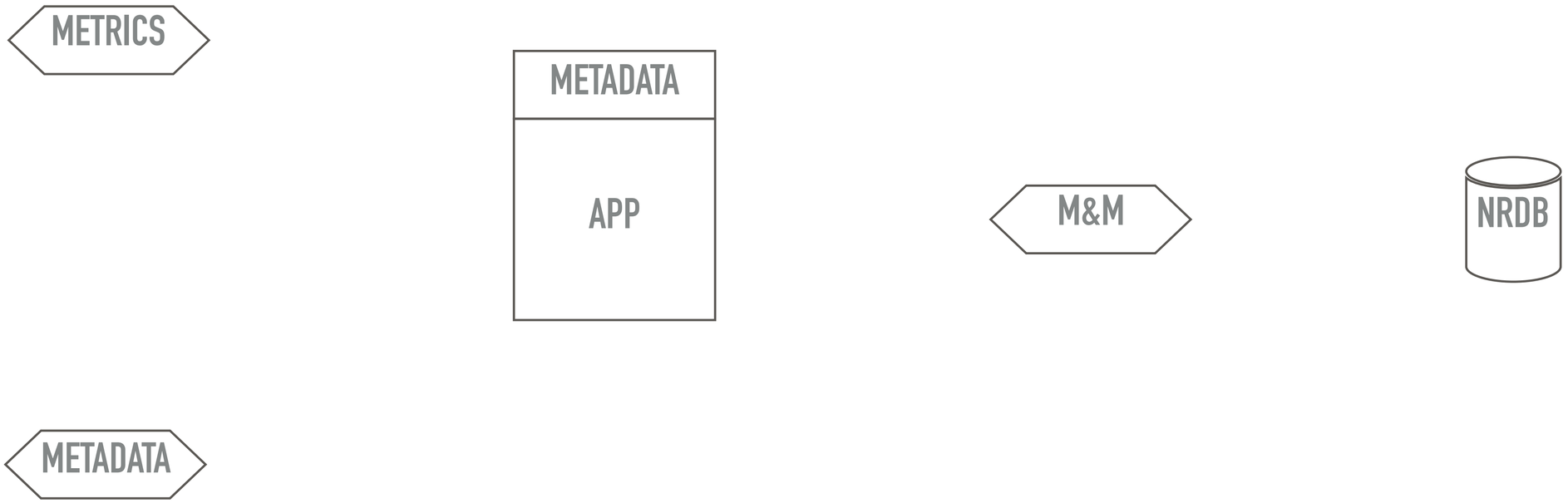
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
METRICS
{
"id": 1,
"name": "host-1",
"region": "eu"
}
METADATA
APP
{
"id": 1,
"timestamp": 5
"max.cpu": 10
"name": "host-1",
"region": "eu"
}
M&M
NRDB

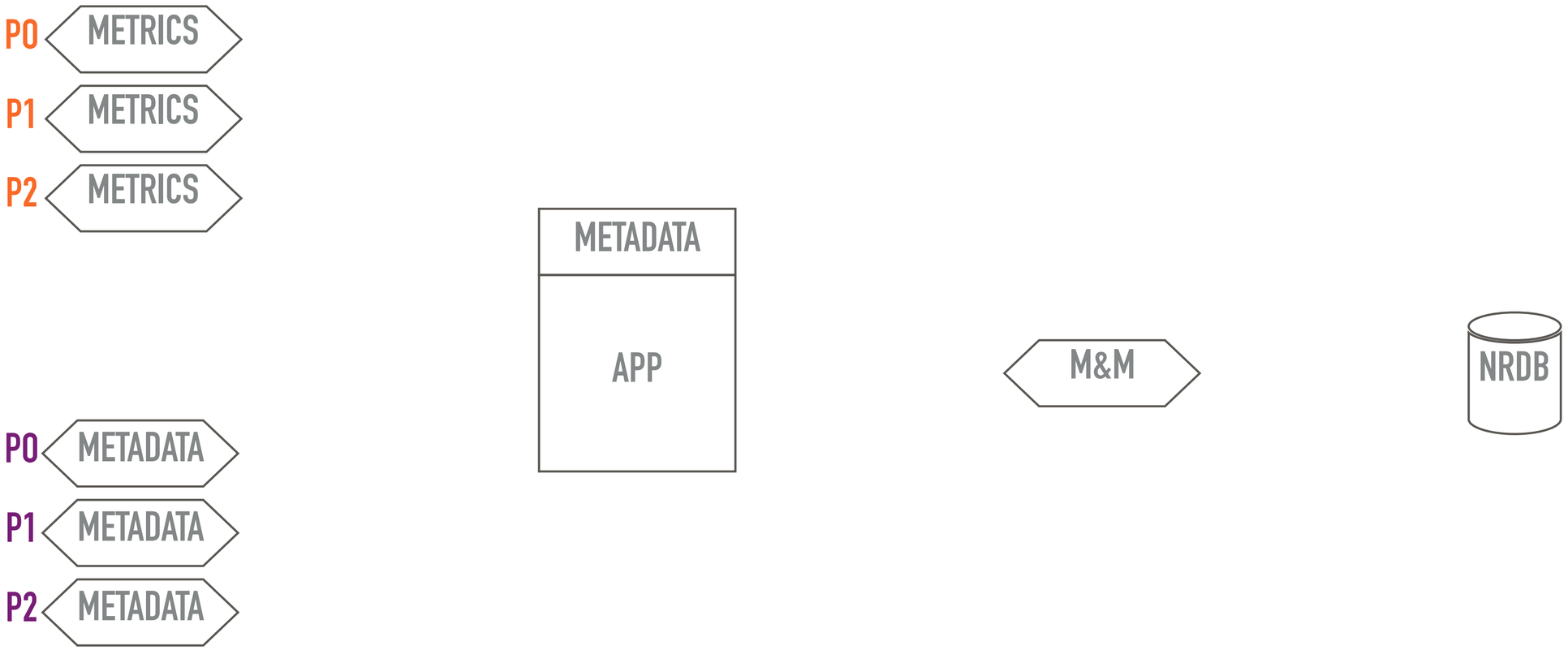
ENRICHMENT


{
"id": 1,
"timestamp": 5,
"max.cpu": 10
}
METRICS
APP
{
"id": 1,
"name": "host-1",
"region": "eu"
}
METADATA
{
"id": 1,
"timestamp": 5
"max.cpu": 10
"name": "host-1",
"region": "eu"
}
M&M
NRDB

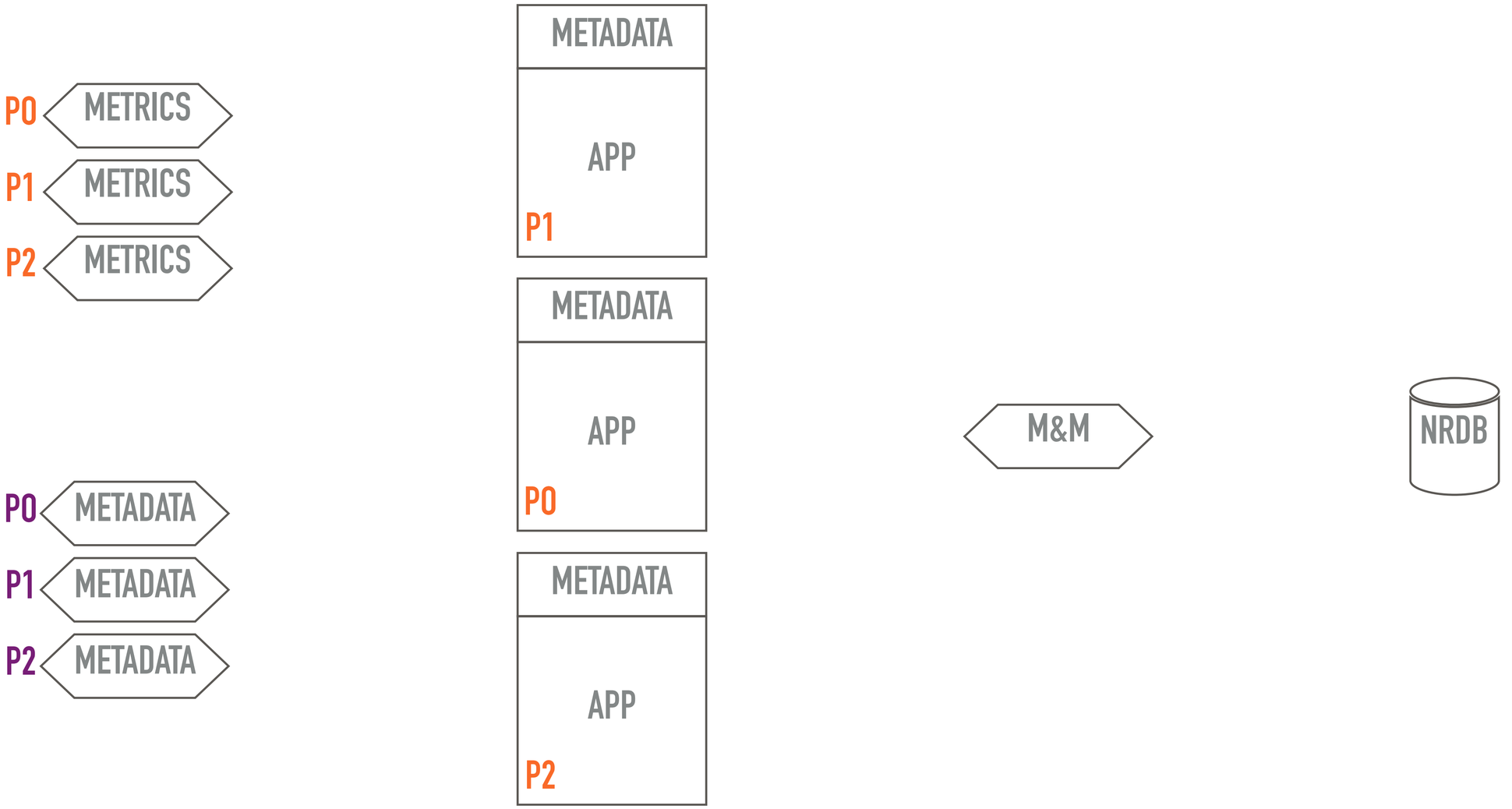
ENRICHMENT



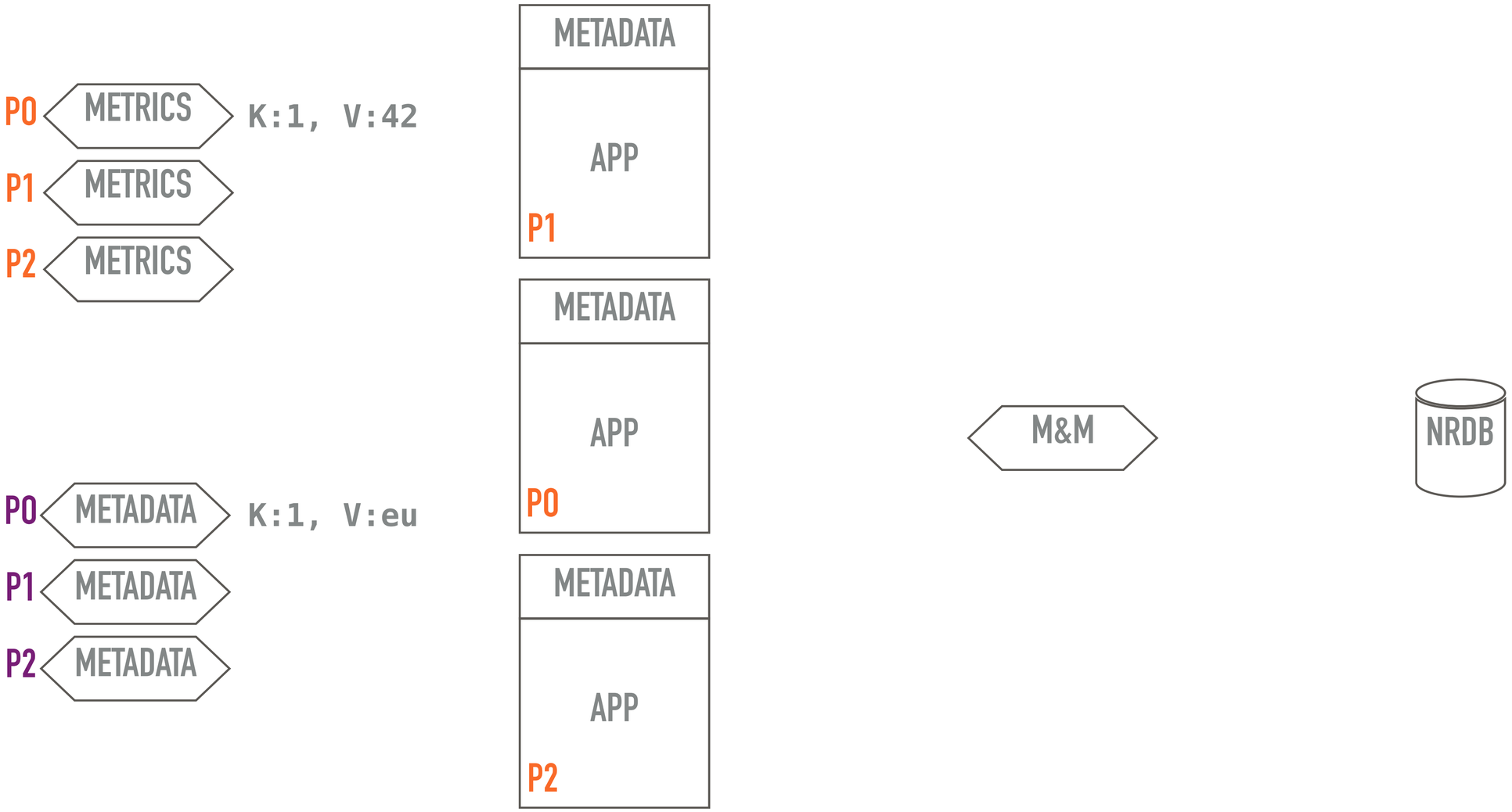
ENRICHMENT


P0
METRICS
P1
METRICS
P2
METRICS
METADATA
APP
M&M
NRDB
P0
METADATA
P1
METADATA
P2
METADATA

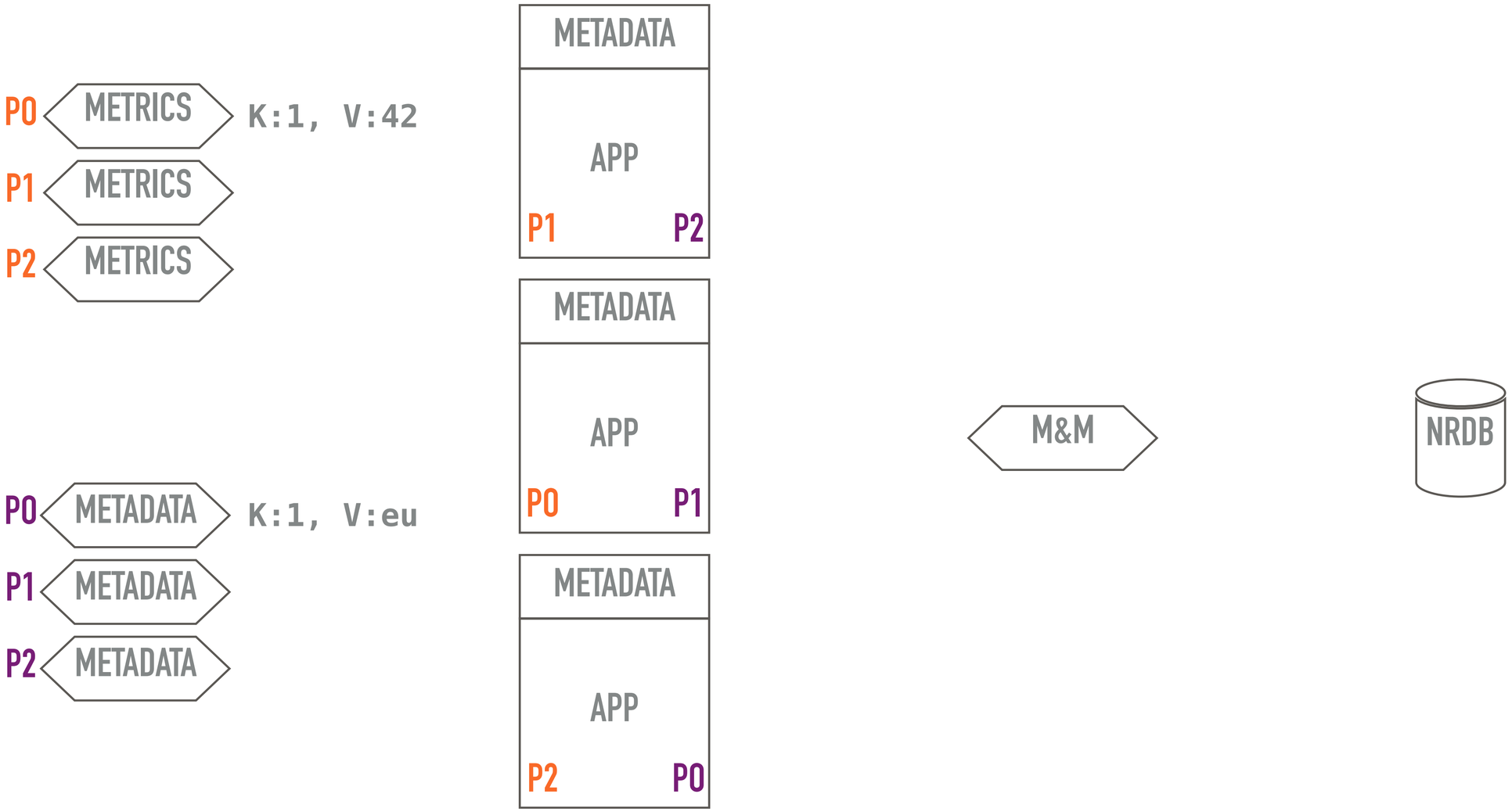
ENRICHMENT


P0
METRICS
P1
METRICS
P2
METRICS
P0
METADATA
P1
METADATA
P2
METADATA
METADATA
APP
P1
METADATA
APP
P0
METADATA
APP
P2
M&M
NRDB

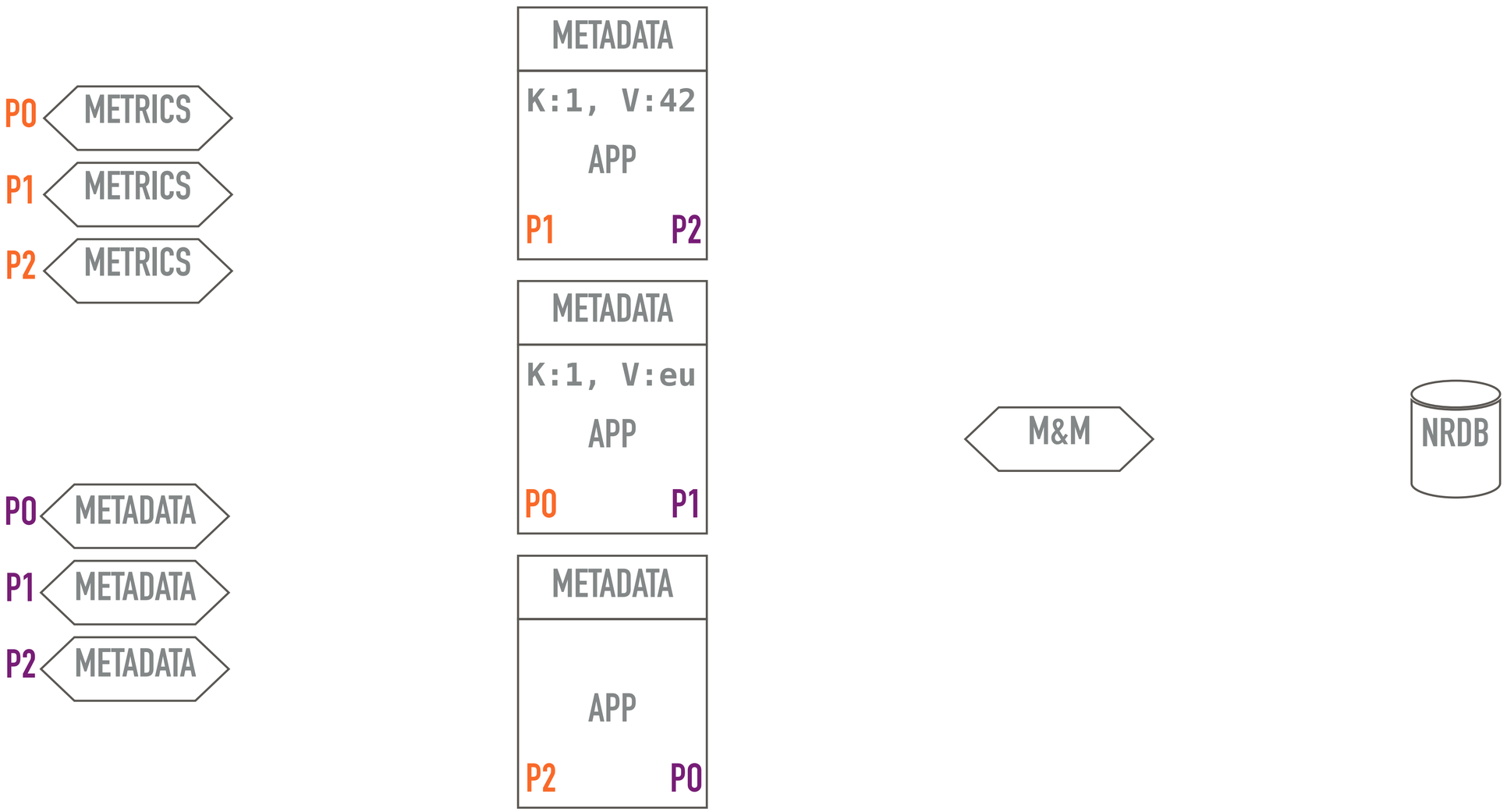
ENRICHMENT


P0
METRICS
K:1, V:42
P1
METRICS
P2
METRICS
P0
METADATA
K:1, V:eu
P1
METADATA
P2
METADATA
METADATA
APP
P1
METADATA
APP
P0
METADATA
APP
P2
M&M
NRDB

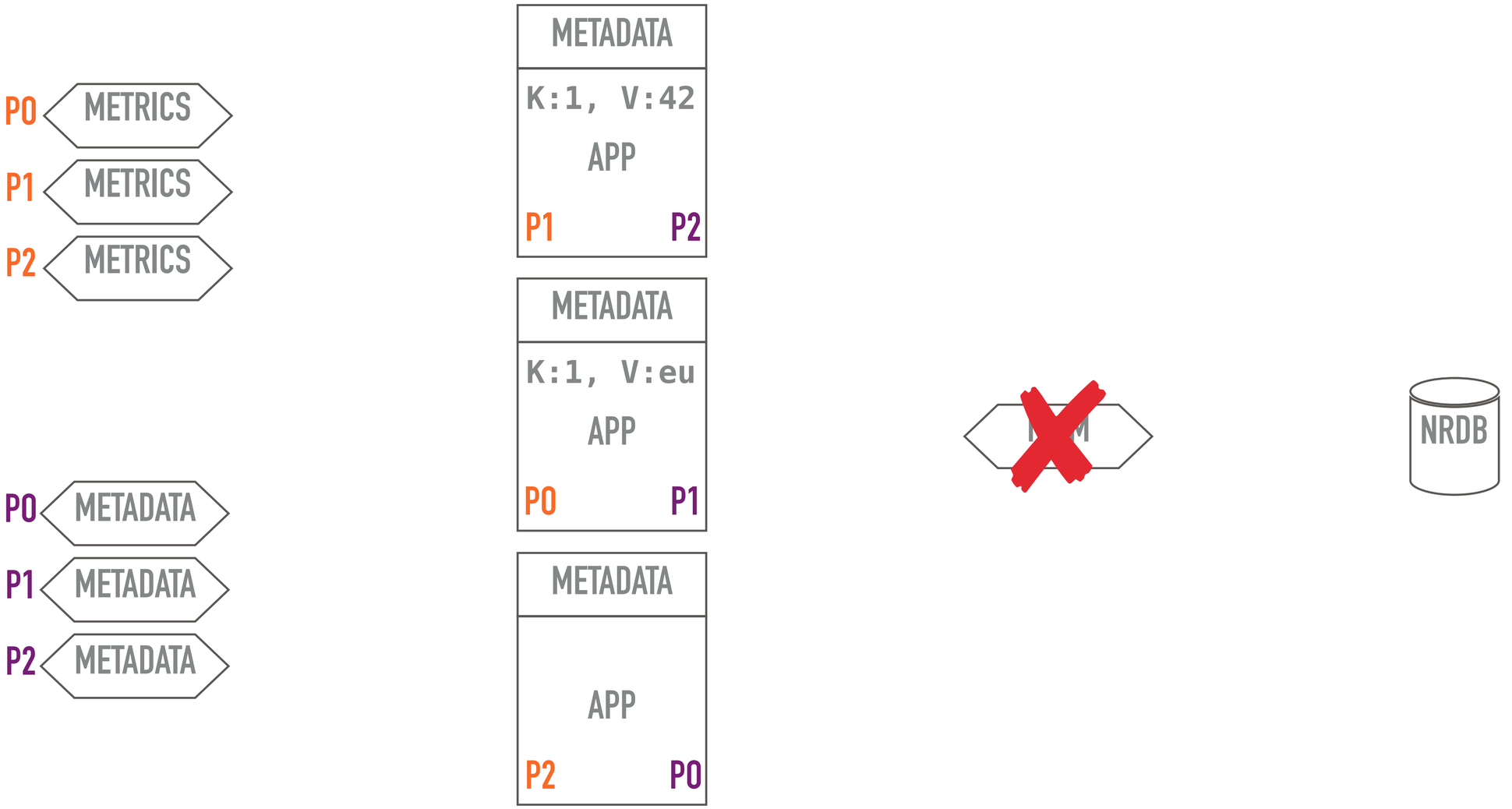
ENRICHMENT


P0
METRICS
K:1, V:42
P1
METRICS
P2
METRICS
P0
METADATA
K:1, V:eu
P1
METADATA
P2
METADATA
METADATA
APP
P1
P2
METADATA
APP
P0
P1
METADATA
APP
P2
P0
M&M
NRDB

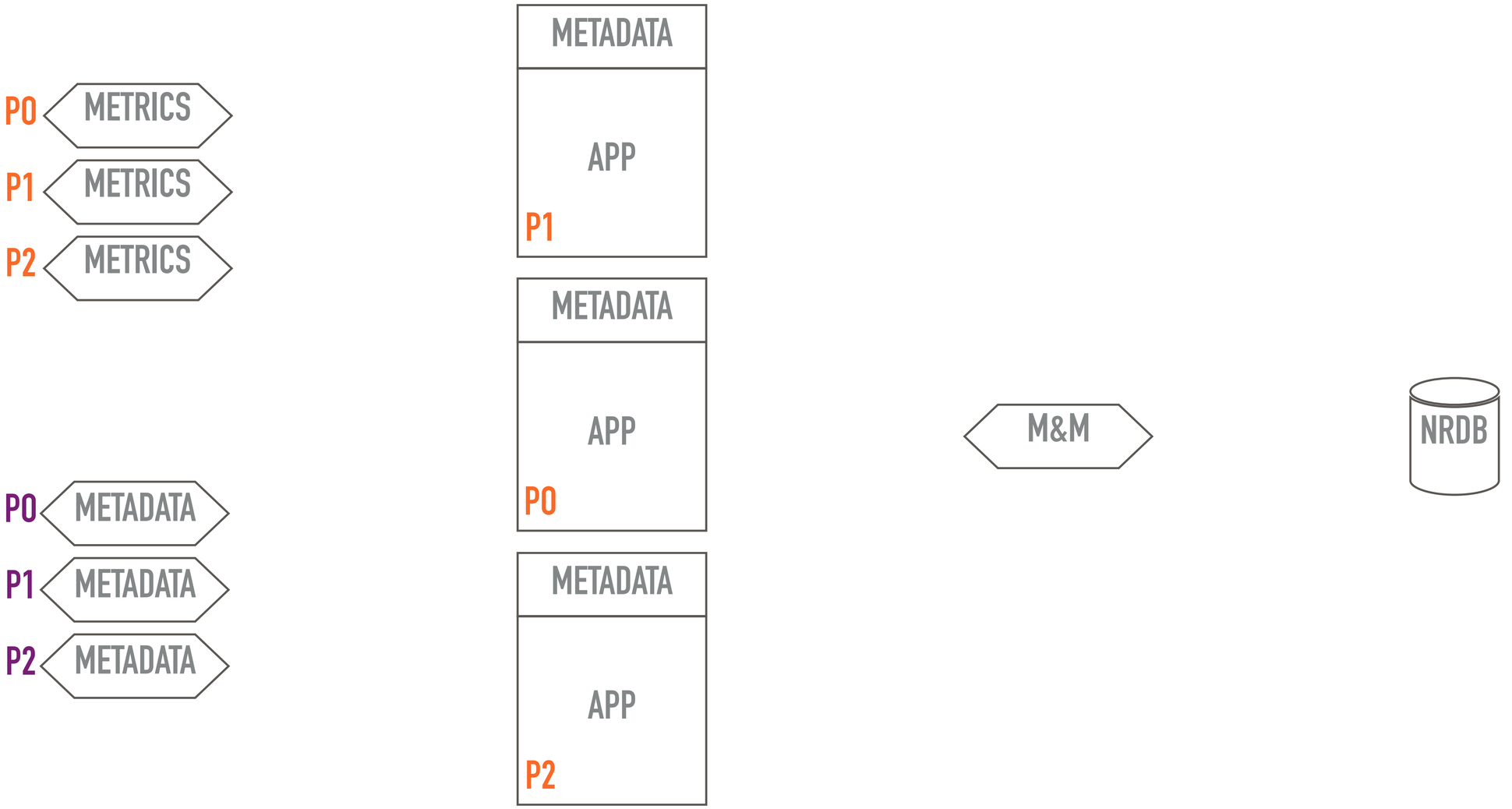
ENRICHMENT


P0
METRICS
P1
METRICS
P2
METRICS
P0
METADATA
P1
METADATA
P2
METADATA
METADATA
K:1, V:42
APP
P1
P2
METADATA
K:1, V:eu
APP
P0
P1
METADATA
APP
P2
P0
M&M
NRDB

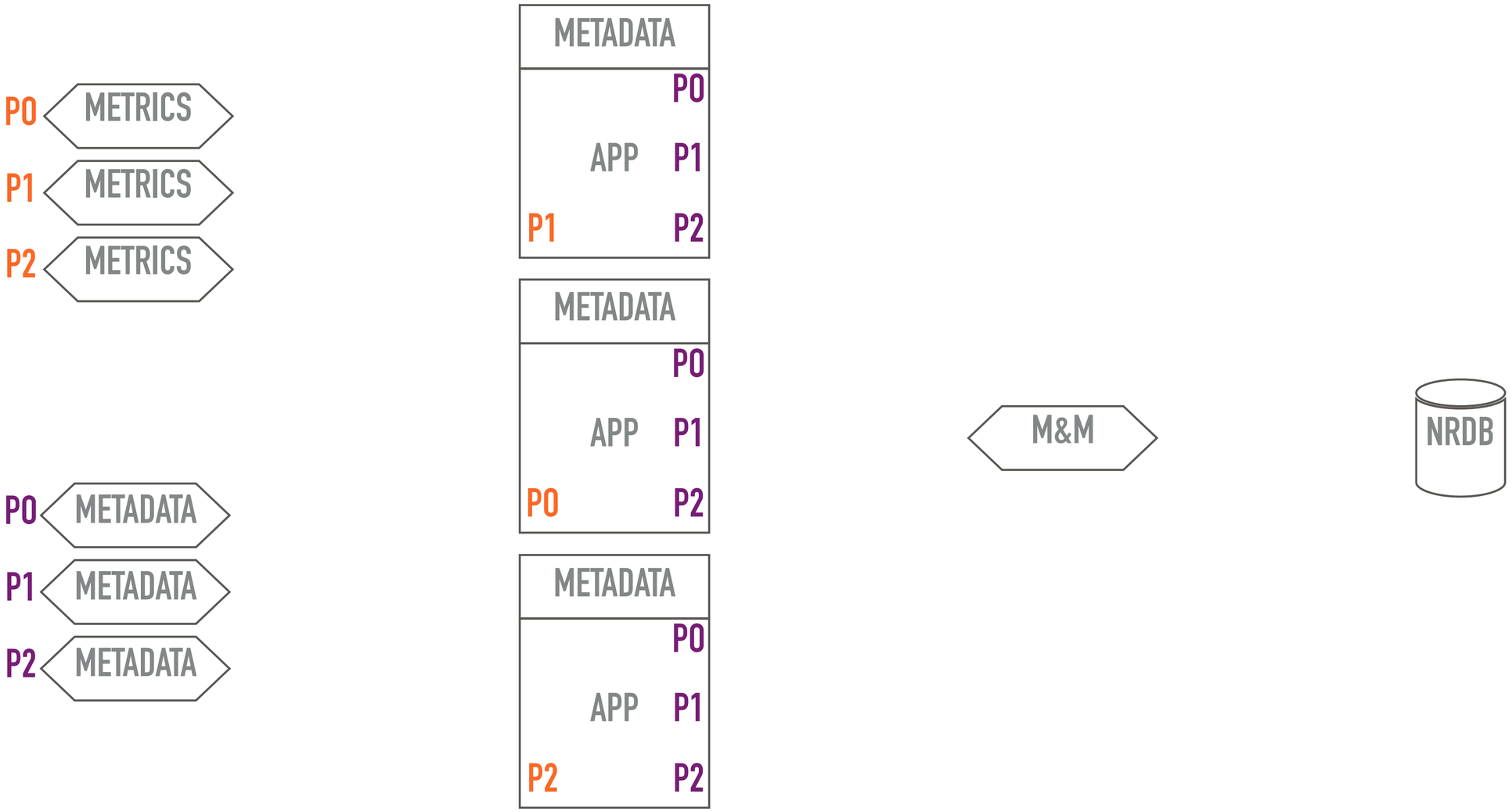
ENRICHMENT


P0 METRICS
P1 METRICS
P2 METRICS
P0 METADATA
P1 METADATA
P2 METADATA
METADATA
K:1, V:42
APP
P1 P2
METADATA
K:1, V:eu
APP
P0 P1
METADATA
APP
P2 P0
NRDB

ENRICHMENT

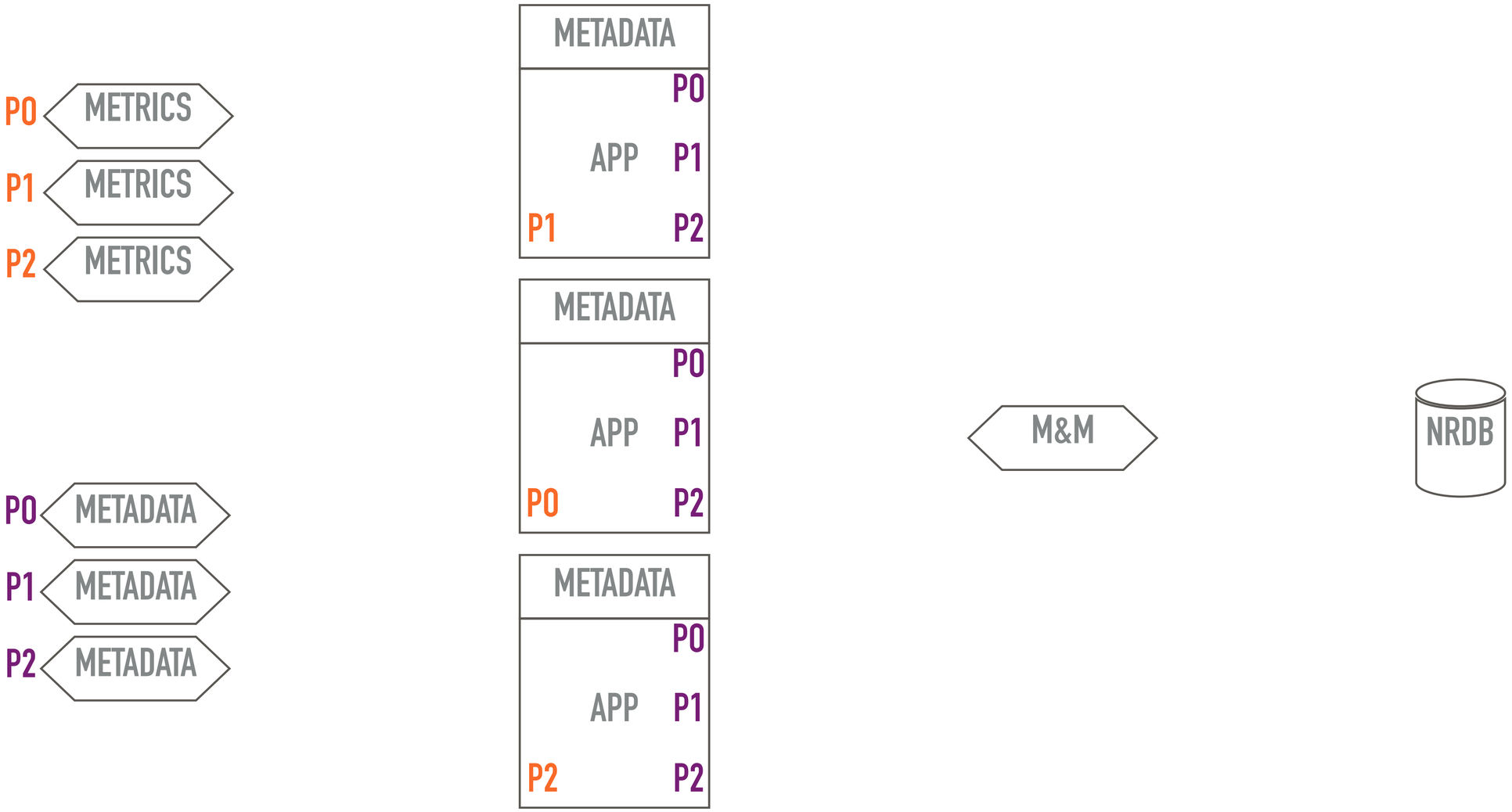

P0
METRICS
P1
METRICS
P2
METRICS
P0
METADATA
P1
METADATA
P2
METADATA
METADATA
APP
P1
METADATA
APP
P0
METADATA
APP
P2
M&M
NRDB

ENRICHMENT

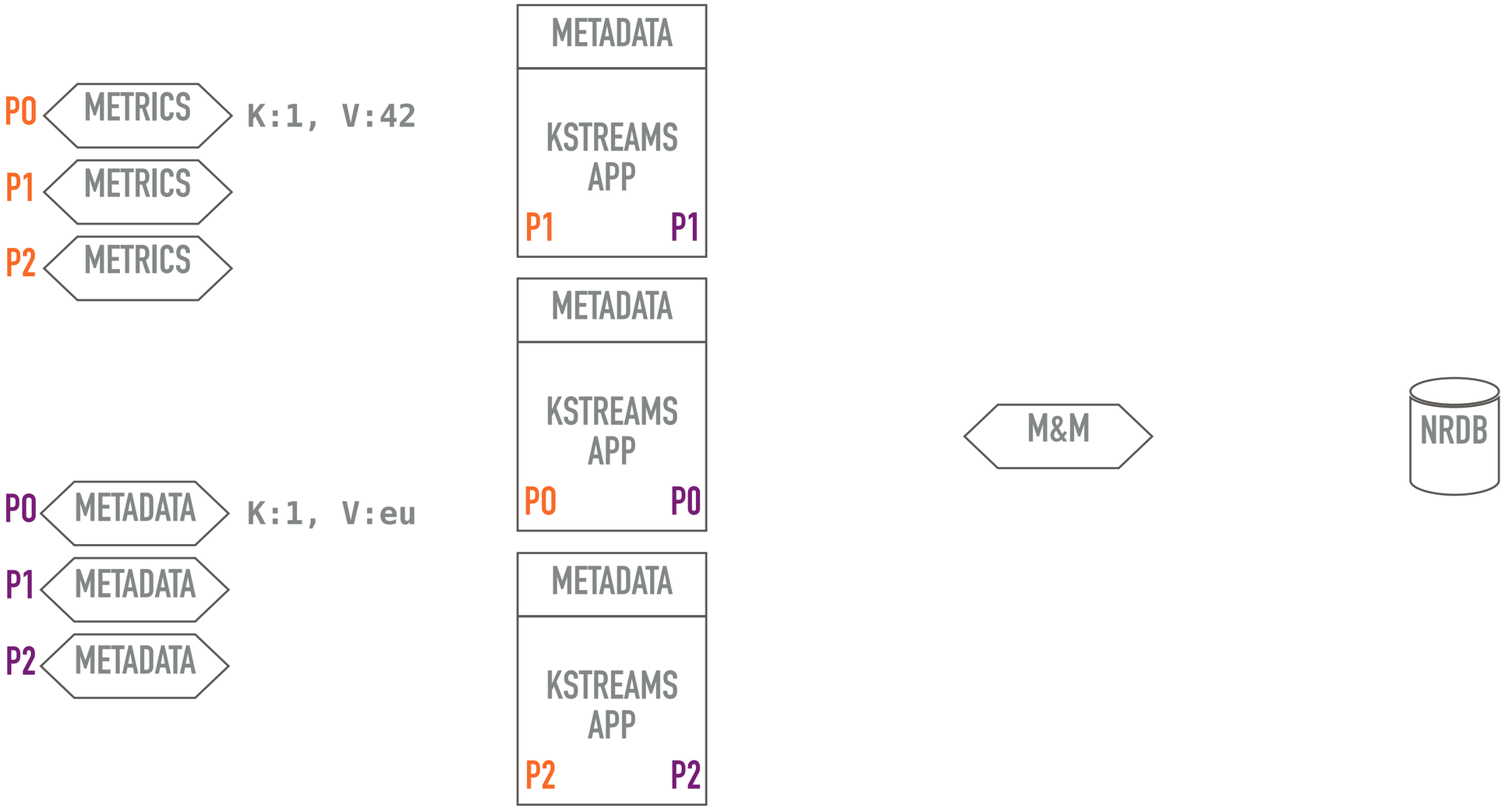

P0
METRICS
P1
METRICS
P2
METRICS
P0
METADATA
P1
METADATA
P2
METADATA
METADATA
P0
APP
P1
P1
P2
METADATA
P0
APP
P1
P0
P2
METADATA
P0
APP
P1
P2
P2
M&M
NRDB

CO-PARTITIONING

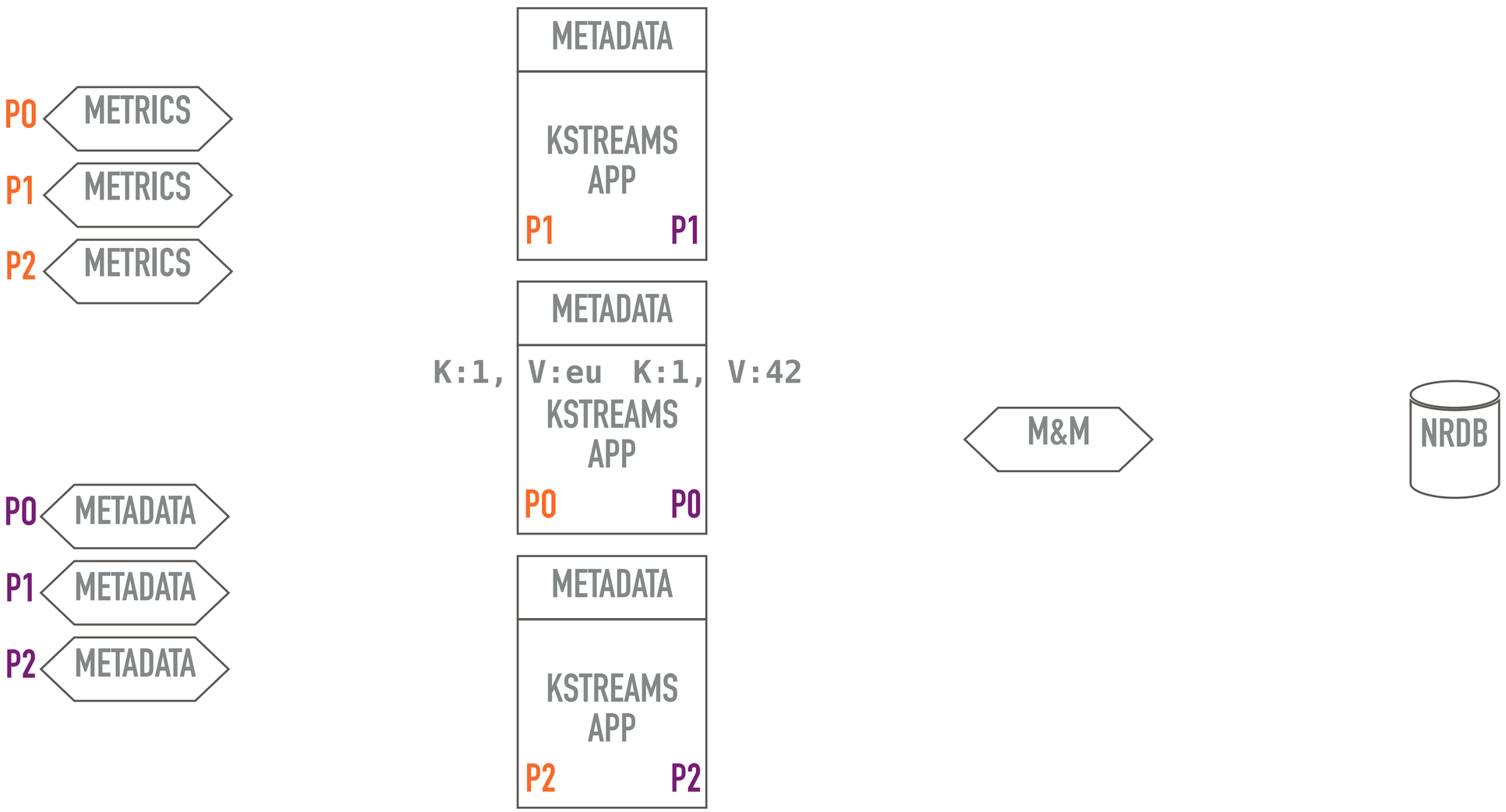
ENRICHMENT


P0
METRICS
P1
METRICS
P2
METRICS
P0
METADATA
P1
METADATA
P2
METADATA
METADATA
P0
APP
P1
P1
P2
METADATA
P0
APP
P1
P0
P2
METADATA
P0
APP
P1
P2
P2
M&M
NRDB

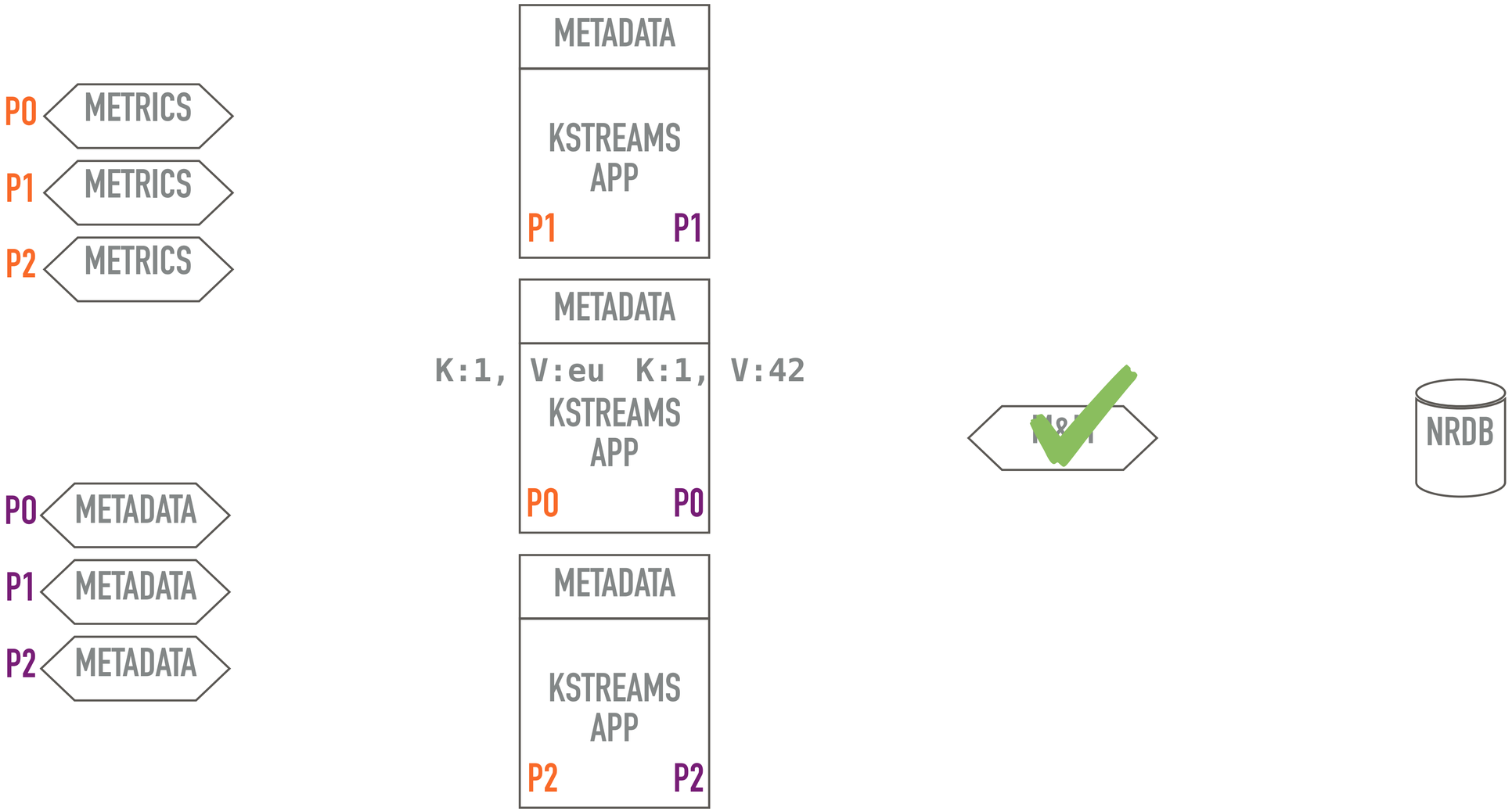
ENRICHMENT


P0
METRICS
K:1, V:42
P1
METRICS
P2
METRICS
P0
METADATA
K:1, V:eu
P1
METADATA
P2
METADATA
METADATA
KSTREAMS APP
P1
P1
METADATA
KSTREAMS APP
P0
P0
METADATA
KSTREAMS APP
P2
P2
M&M
NRDB

ENRICHMENT


P0 METRICS
P1 METRICS
P2 METRICS
P0 METADATA
P1 METADATA
P2 METADATA
METADATA
KSTREAMS APP
P1 P1
METADATA
K:1, V:eu K:1, V:42
KSTREAMS APP
P0 P0
METADATA
KSTREAMS APP
P2 P2
M&M
NRDB

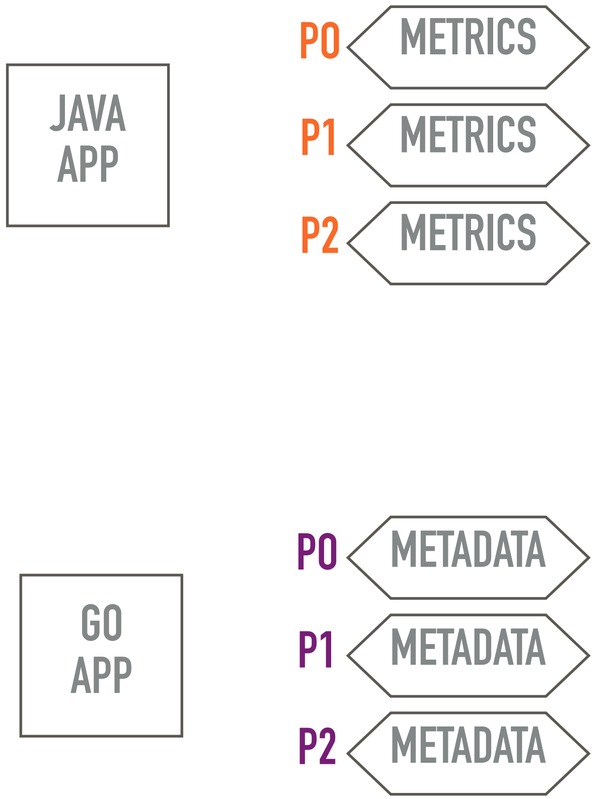
ENRICHMENT


P0 METRICS
P1 METRICS
P2 METRICS
P0 METADATA
P1 METADATA
P2 METADATA
METADATA
KSTREAMS APP
P1 P1
METADATA
K:1, V:eu K:1, V:42
KSTREAMS APP
P0 P0
METADATA
KSTREAMS APP
P2 P2
NRDB

ENRICHMENT

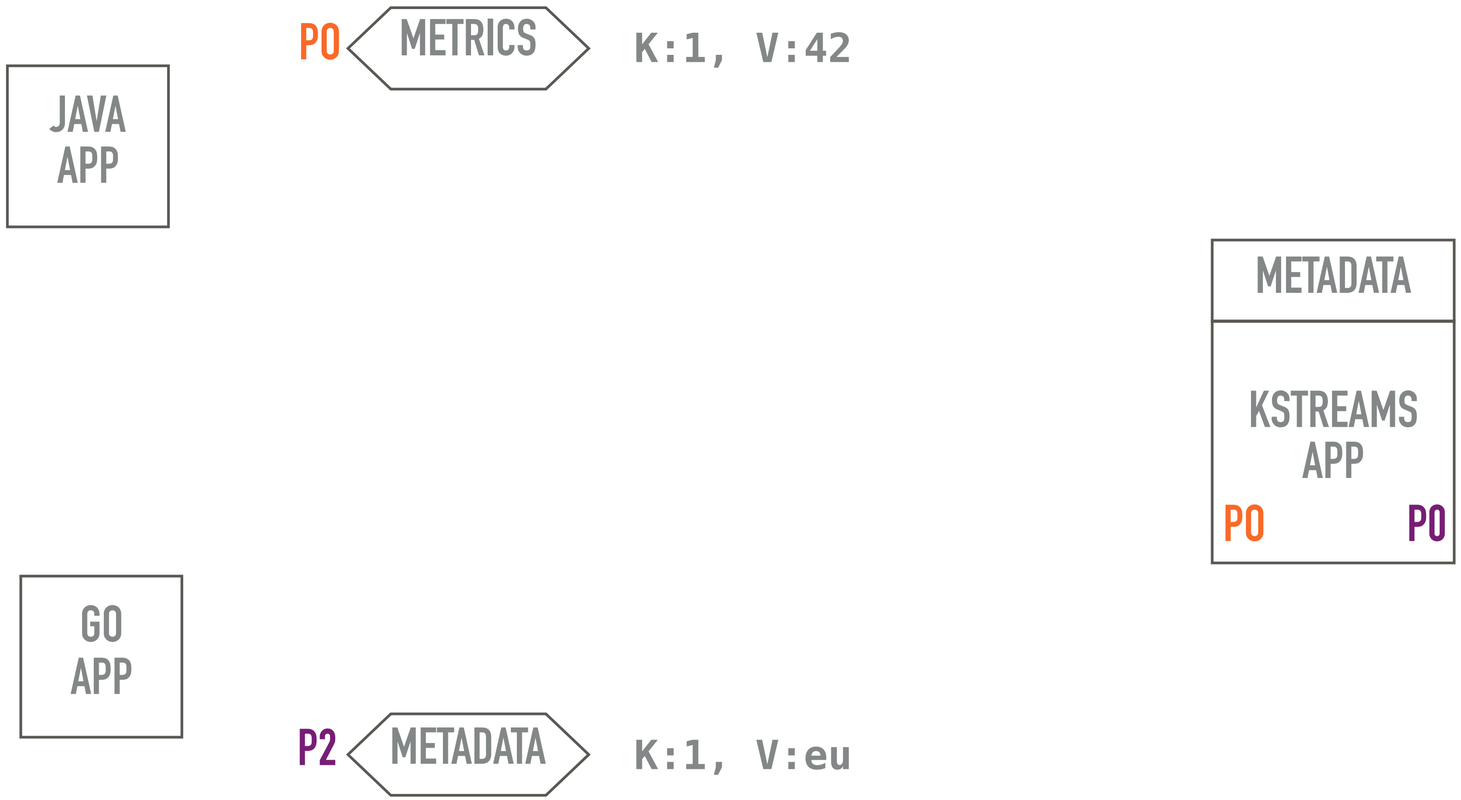
- same partitioning key
- same number of partitions
- same hash function of producer

ENRICHMENT

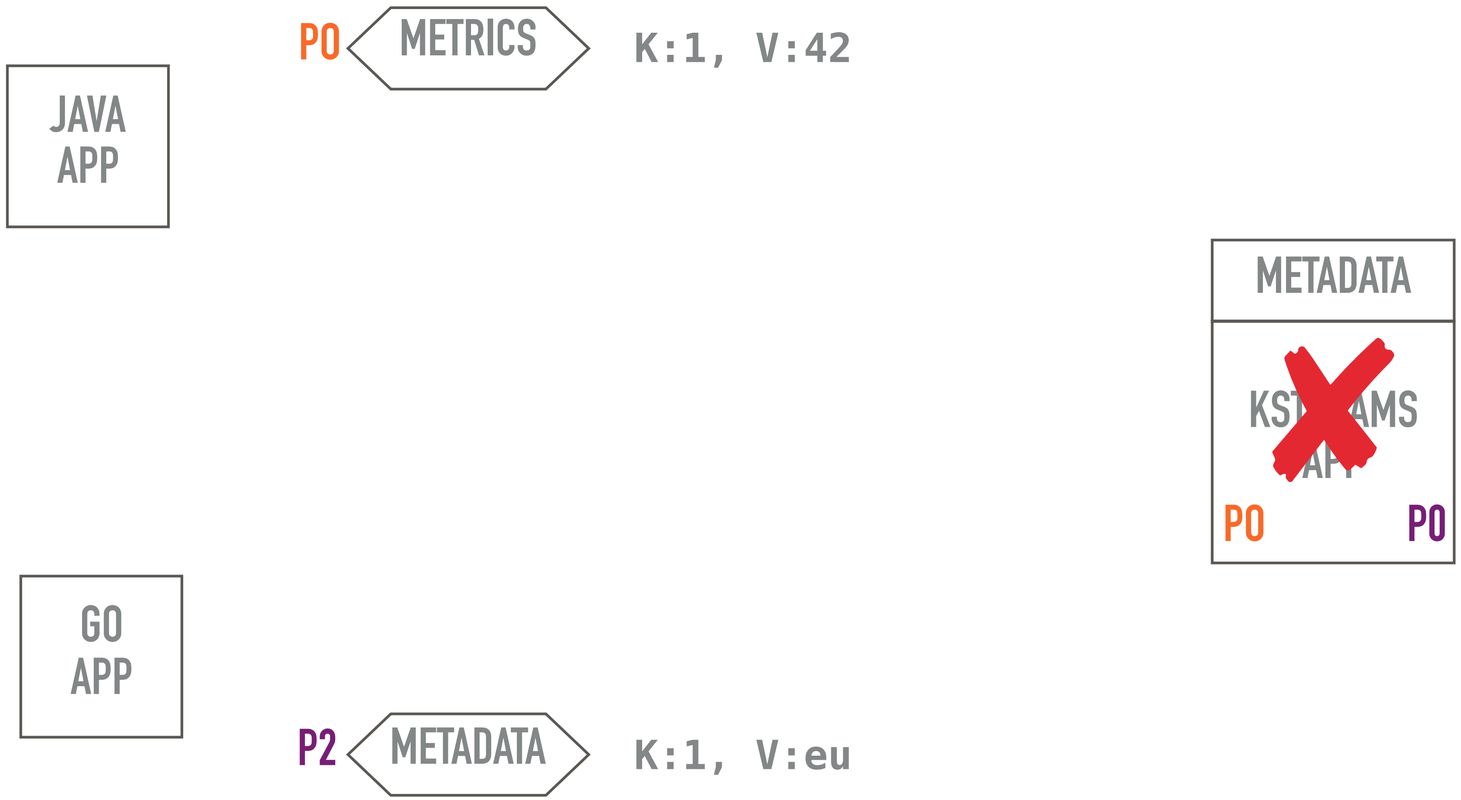

JAVA
APP
P0
METRICS
P1
METRICS
P2
METRICS
GO
APP
P0
METADATA
P1
METADATA
P2
METADATA

ENRICHMENT

ENRICHMENT


P0
METRICS
K:1, V:42
JAVA APP
METADATA
KSTREAMS APP
P0
P0
GO APP
P2
METADATA
K:1, V:eu

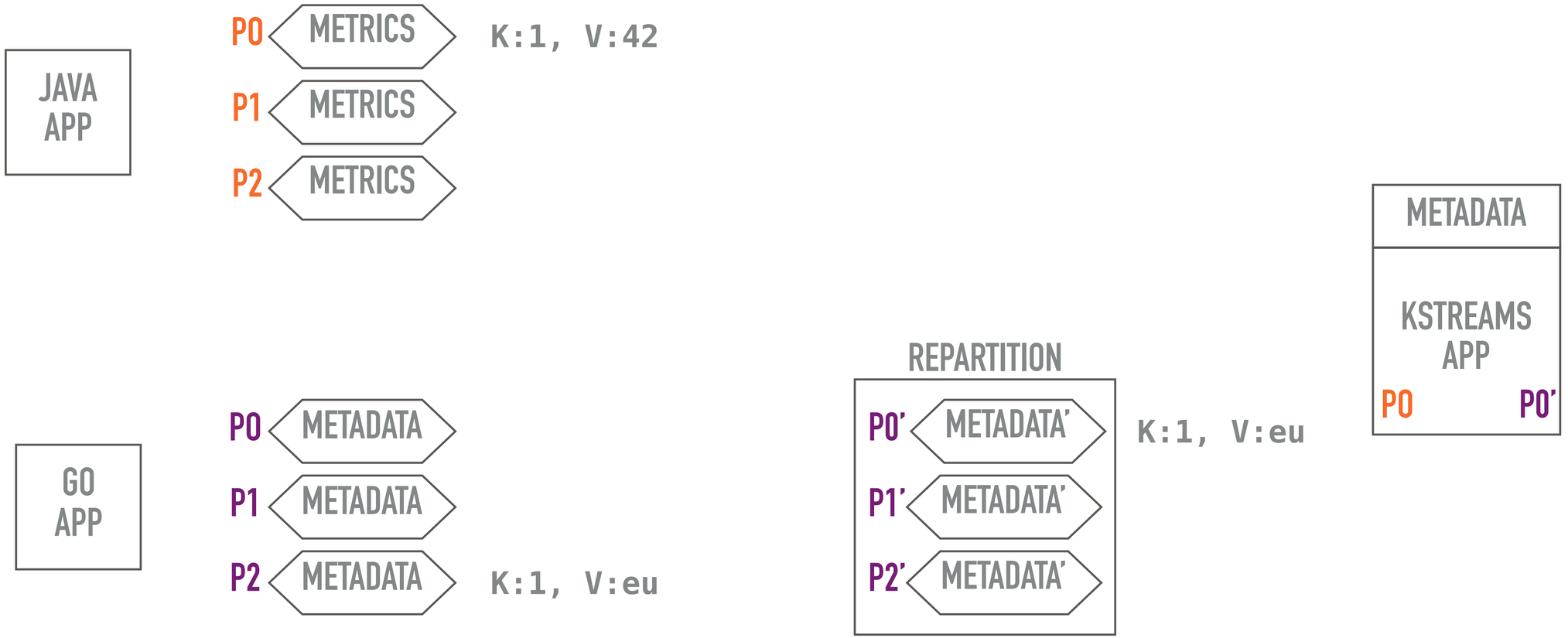
ENRICHMENT


P0
METRICS
K:1, V:42
JAVA APP
METADATA
KST AMS API
P0
P0
GO APP
P2
METADATA
K:1, V:eu

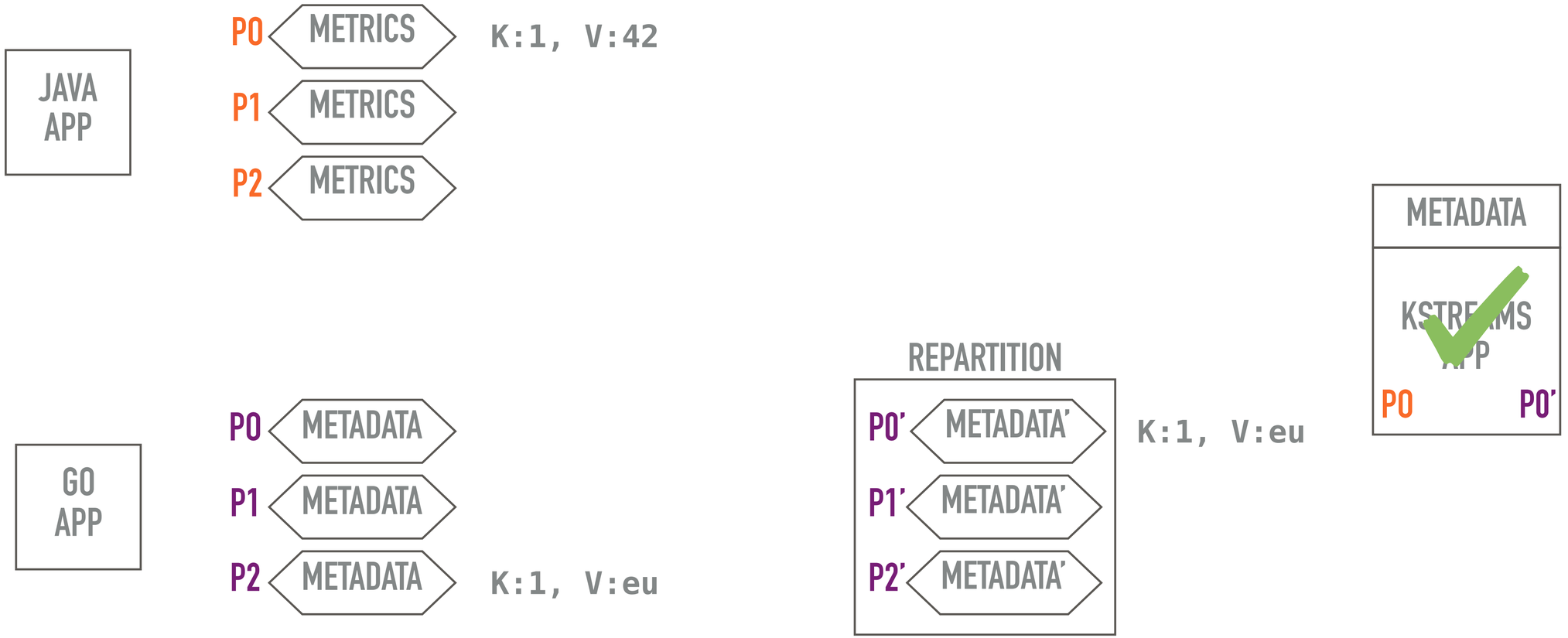
ENRICHMENT


MURMUR2
P0
METRICS
JAVA
APP
METADATA
KSTREAMS
APP
P0
P0
FNV-1A
GO
APP
P2
METADATA

ENRICHMENT


JAVA APP
P0
METRICS
K:1, V:42
P1
METRICS
P2
METRICS
GO APP
P0
METADATA
P1
METADATA
P2
METADATA
K:1, V:eu
REPARTITION
P0'
METADATA'
P1'
METADATA'
P2'
METADATA'
K:1, V:eu
METADATA
KSTREAMS APP
P0
P0'

ENRICHMENT

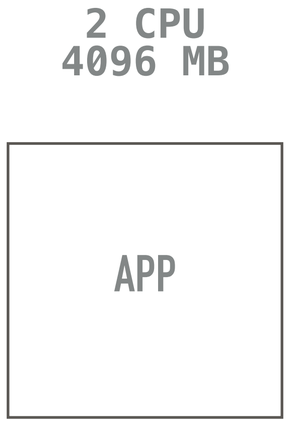

JAVA APP
P0 METRICS
P1 METRICS
P2 METRICS
K:1, V:42
GO APP
P0 METADATA
P1 METADATA
P2 METADATA
K:1, V:eu
REPARTITION
P0' METADATA'
P1' METADATA'
P2' METADATA'
K:1, V:eu
METADATA
KSTREAMS APP
P0
P0'

ENRICHMENT

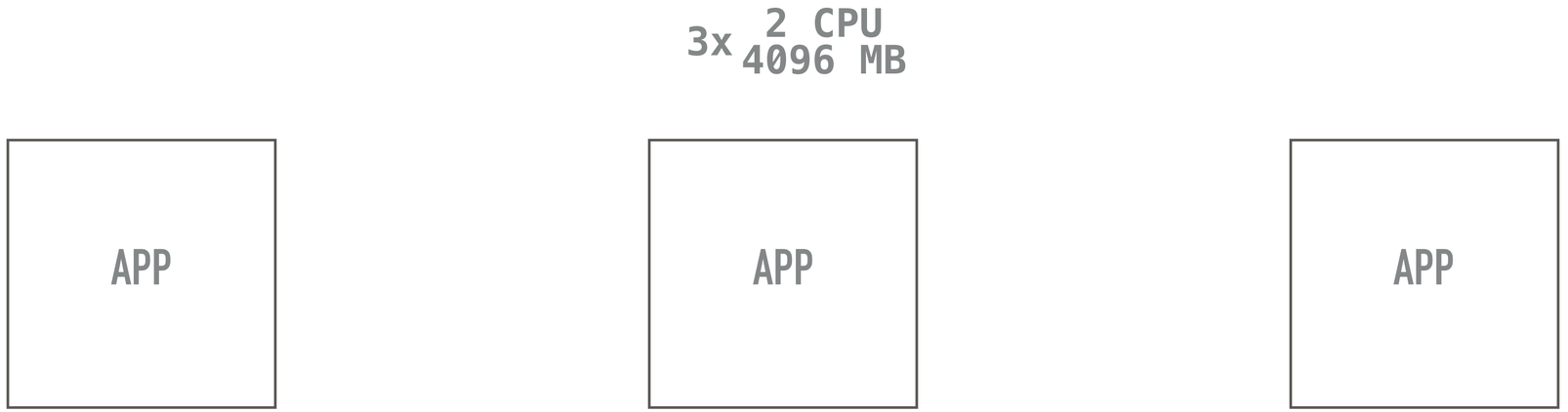
- enrichment at scale thanks to co-partitioning
- in case of doubt – repartition
- a bit more kafka topics

VACATIONS?

PRODUCTION READINESS



PRODUCTION READINESS



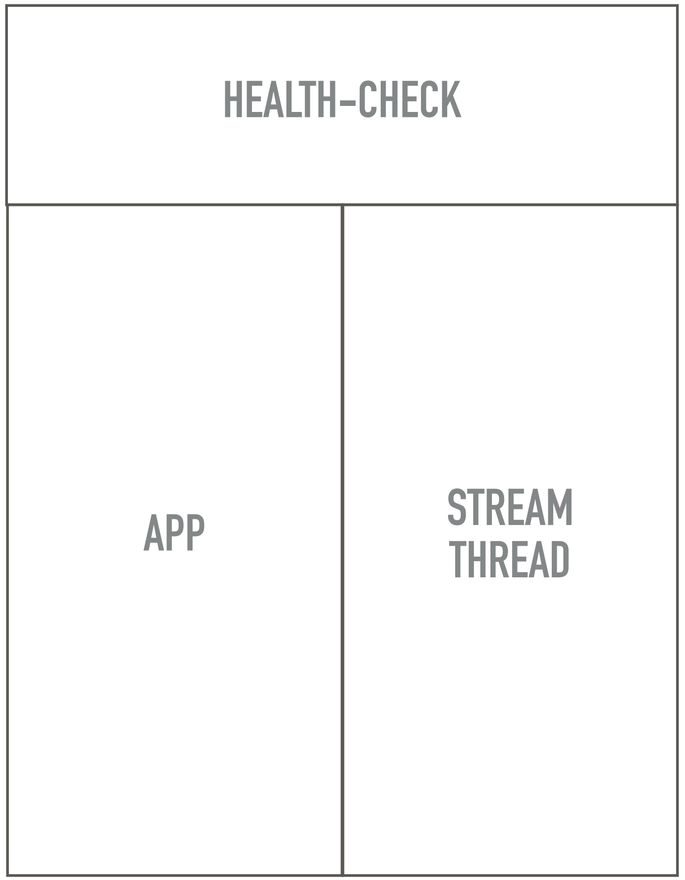

HEALTH-CHECK
APP
HEALTH-CHECK
APP
HEALTH-CHECK
APP

PRODUCTION READINESS

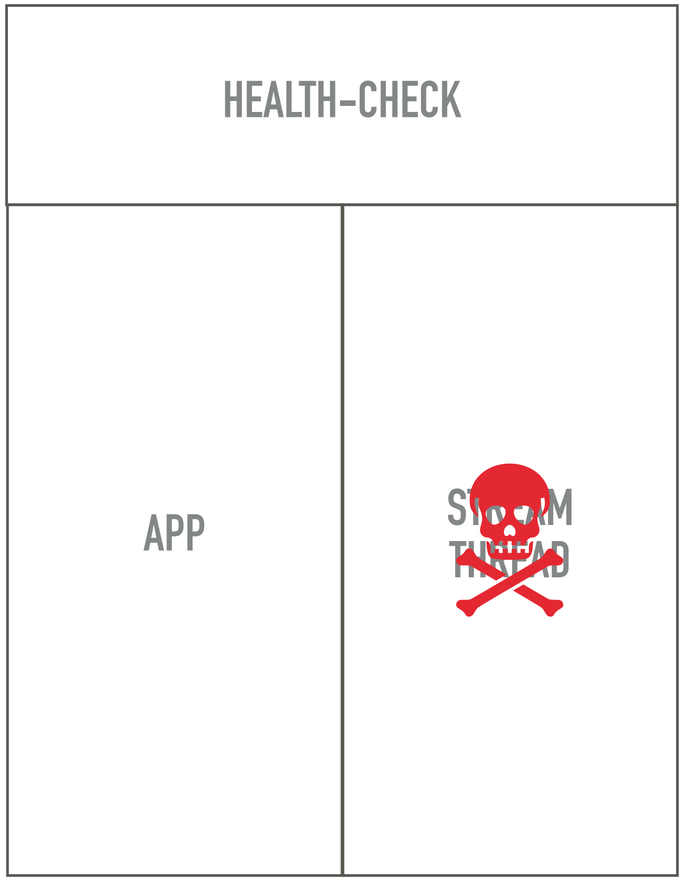

HEALTH-CHECK
APP
HEALTH-CHECK
APP


HEALTH-CHECK
APP
HEALTH-CHECK
APP
HEALTH-CHECK
APP

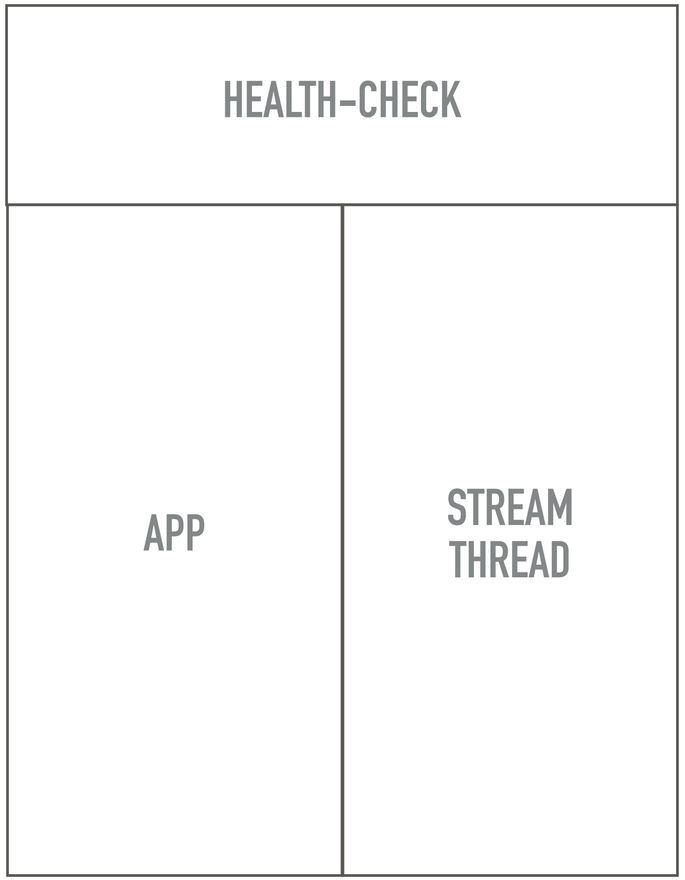
PRODUCTION READINESS


HEALTH-CHECK
APP
STREAM
THREAD

PRODUCTION READINESS



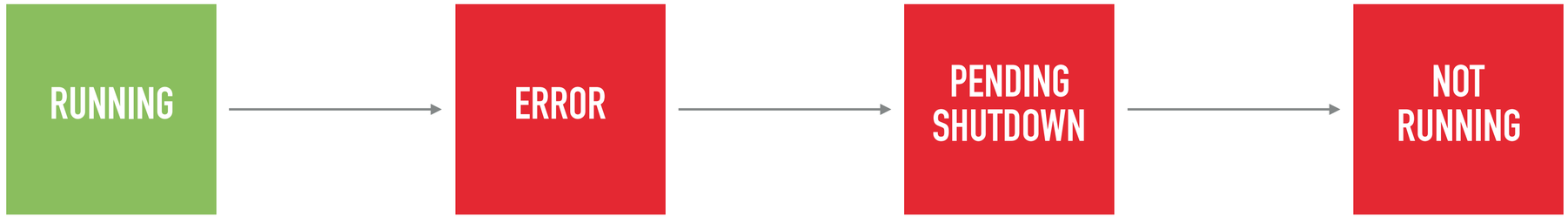
PRODUCTION READINESS

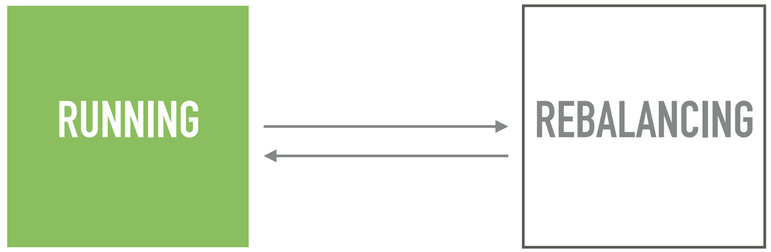

HEALTH-CHECK
APP
STREAM
THREAD

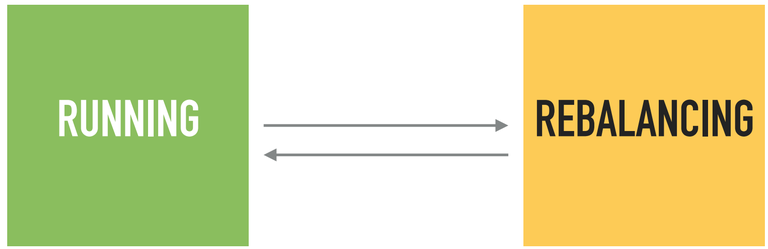
LISTENERS

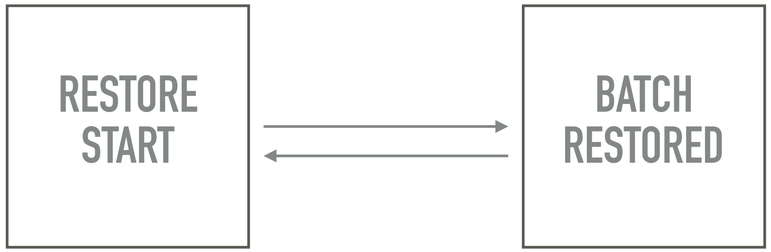
PRODUCTION READINESS

RUNNING

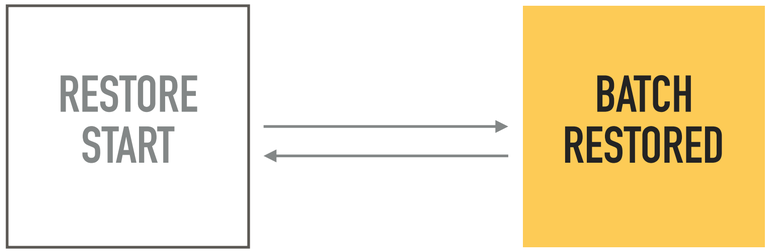

RUNNING
ERROR
PENDING SHUTDOWN
NOT RUNNING


RUNNING
REBALANCING


RUNNING
REBALANCING


RESTORE START
BATCH RESTORED


RESTORE START
BATCH RESTORED
RESTORE END


RESTORE START
BATCH RESTORED

PRODUCTION READINESS

- consuming/producing throughput
- missing joins
- rebalancing loops
- storage-specific metrics
- business-specific metrics

PRODUCTION READINESS

PRODUCTION READINESS

PRODUCTION READINESS

PRODUCTION READINESS

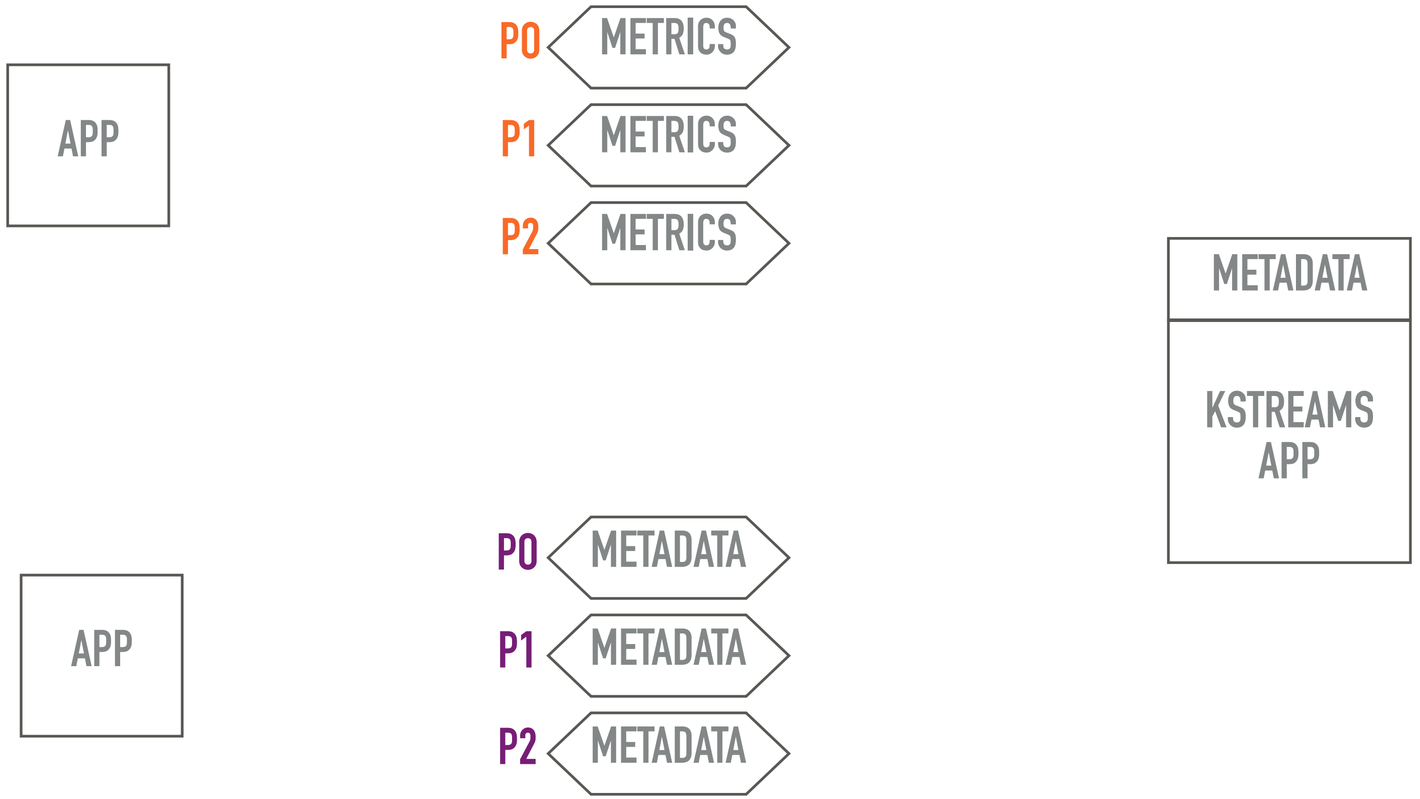
PRODUCTION READINESS

PRODUCTION READINESS

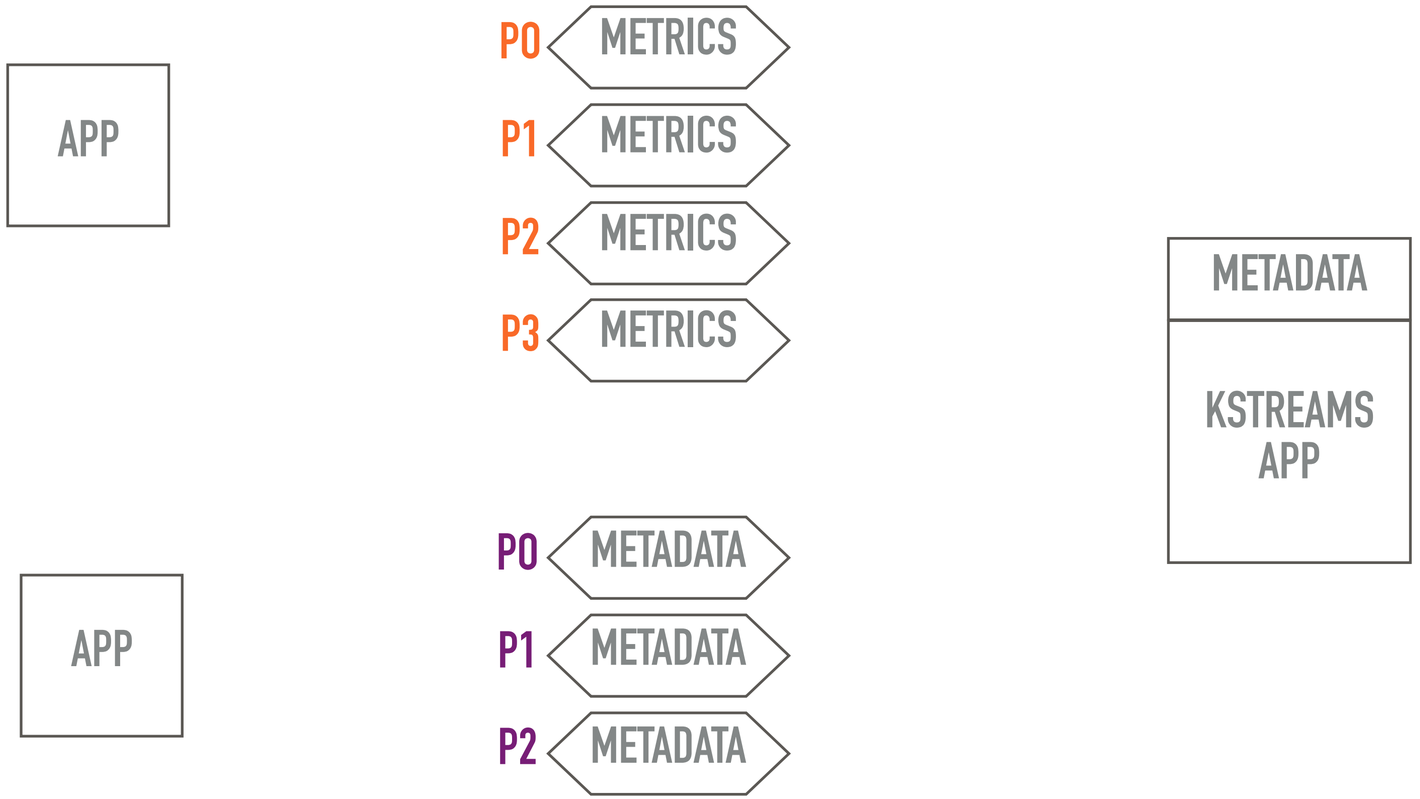
PRODUCTION READINESS

SCALE!

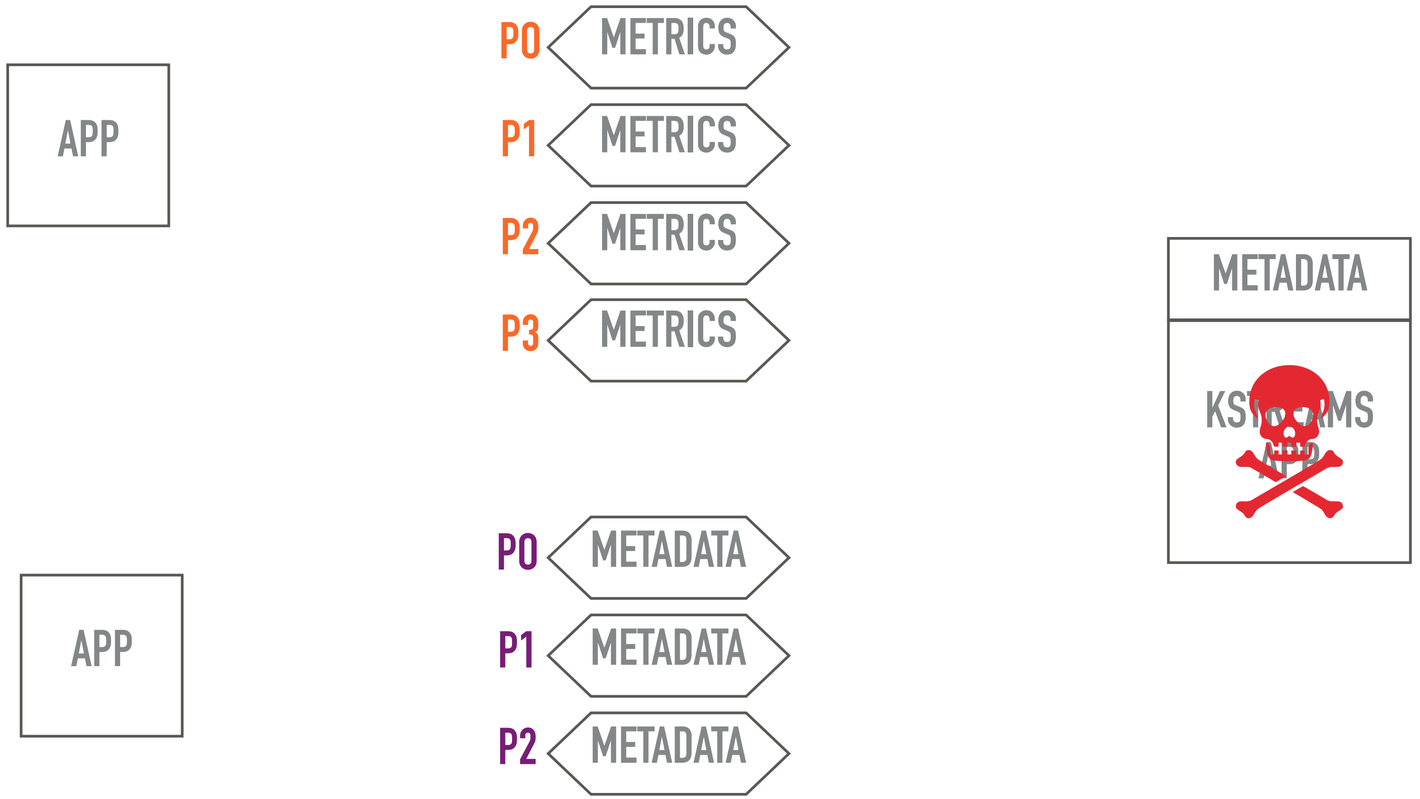
SCALE OUT


APP
P0
METRICS
P1
METRICS
P2
METRICS
METADATA
KSTREAMS
APP
APP
P0
METADATA
P1
METADATA
P2
METADATA

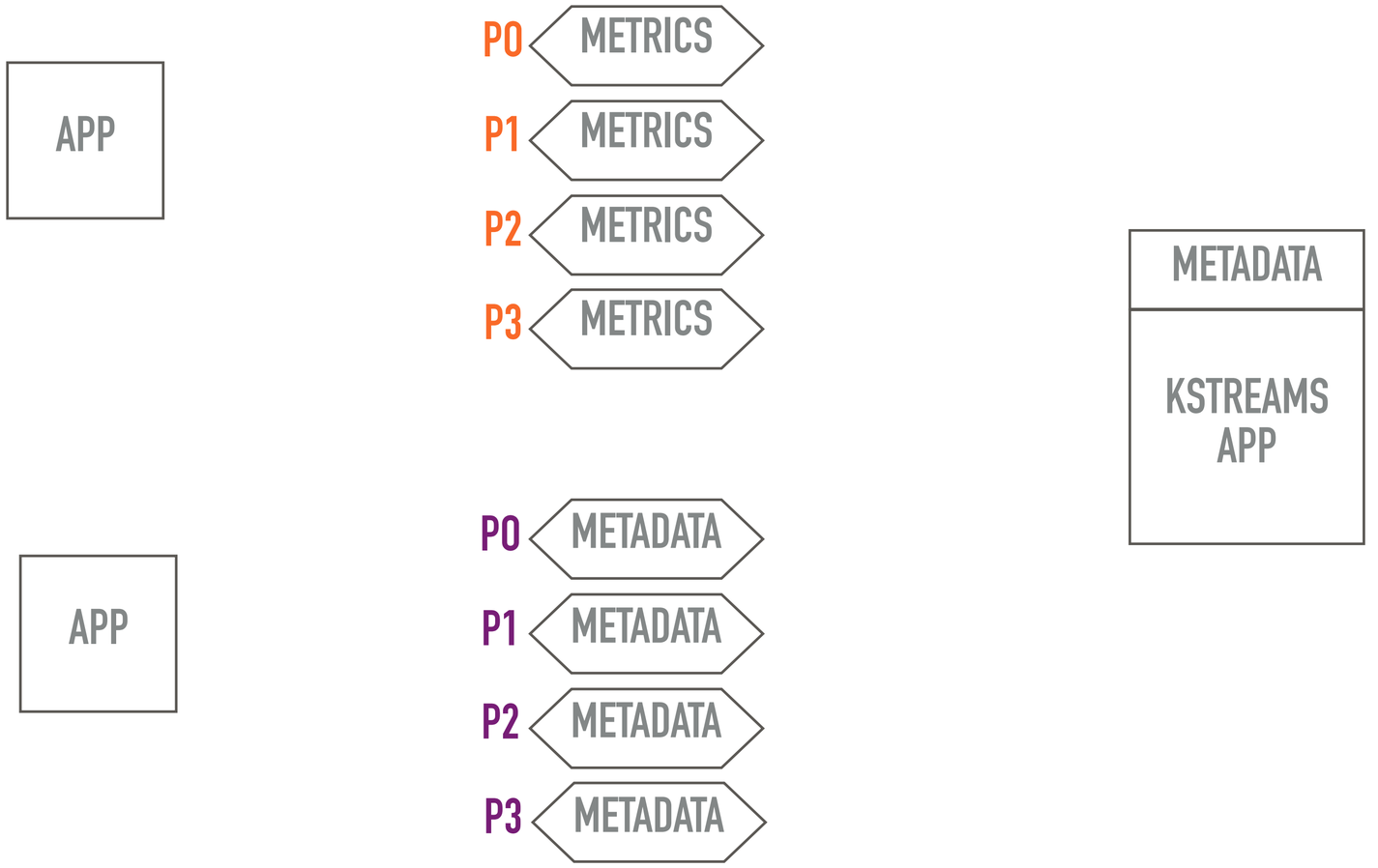
SCALE OUT


APP
P0
METRICS
P1
METRICS
P2
METRICS
P3
METRICS
METADATA
KSTREAMS
APP
APP
P0
METADATA
P1
METADATA
P2
METADATA

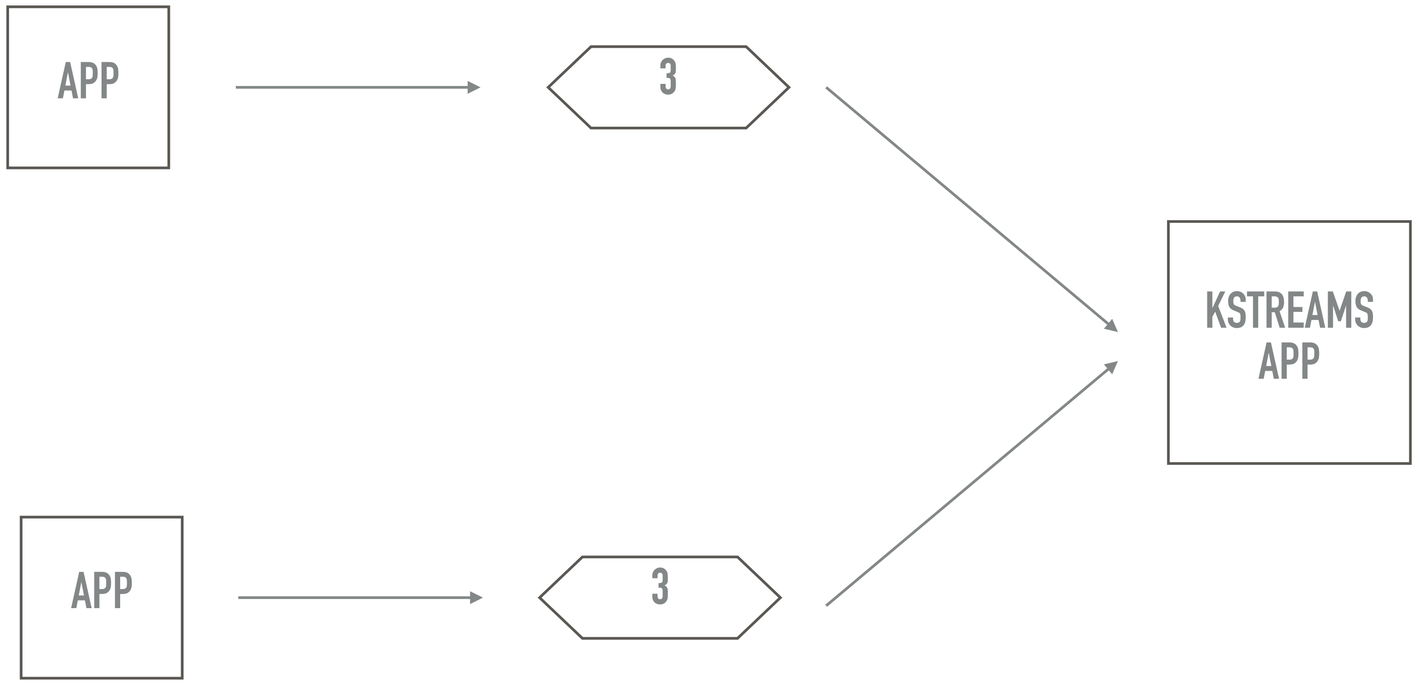
SCALE OUT


APP
P0 METRICS
P1 METRICS
P2 METRICS
P3 METRICS
METADATA
APP
P0 METADATA
P1 METADATA
P2 METADATA

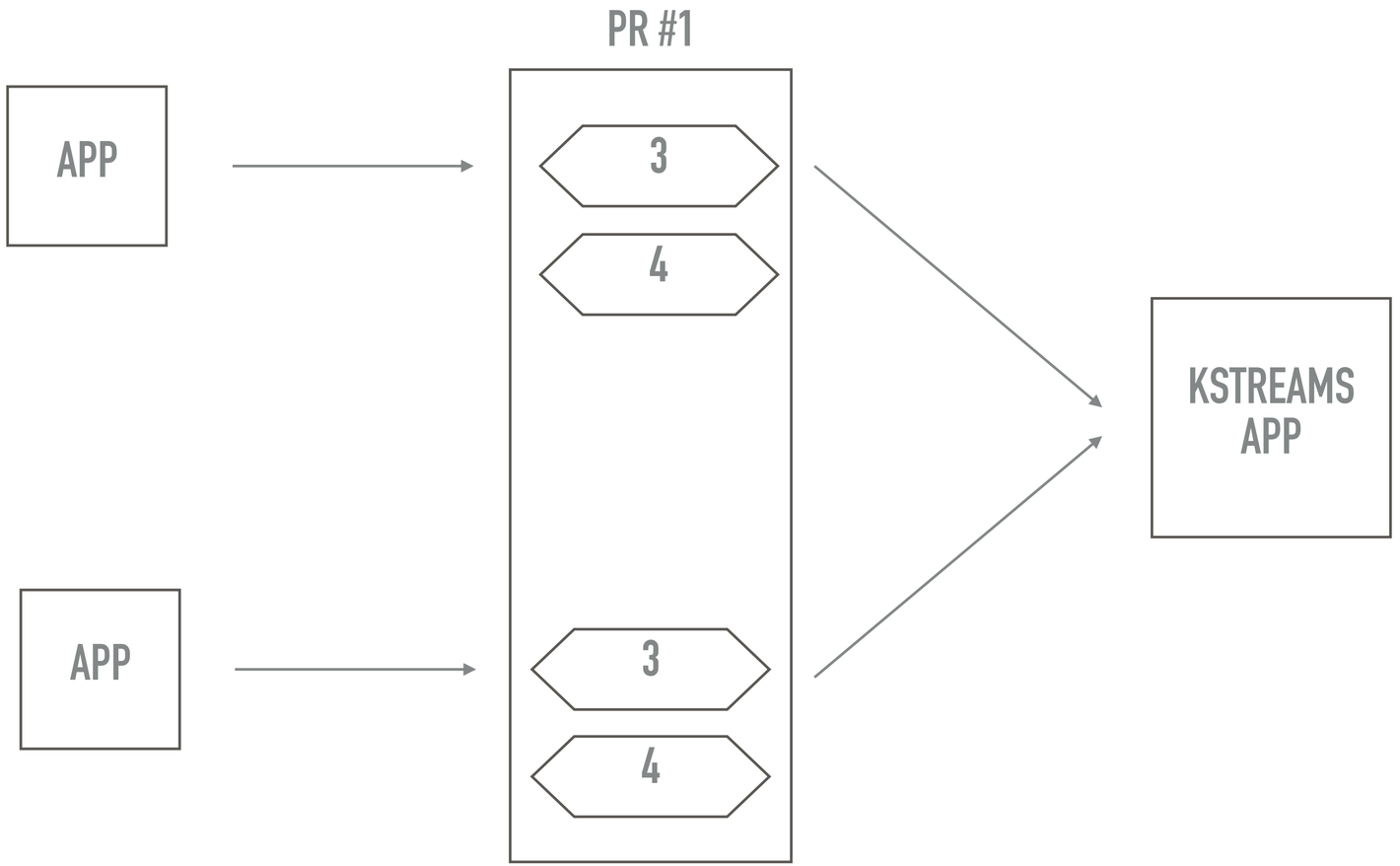
SCALE OUT


APP
P0 METRICS
P1 METRICS
P2 METRICS
P3 METRICS
METADATA
KSTREAMS APP
APP
P0 METADATA
P1 METADATA
P2 METADATA
P3 METADATA

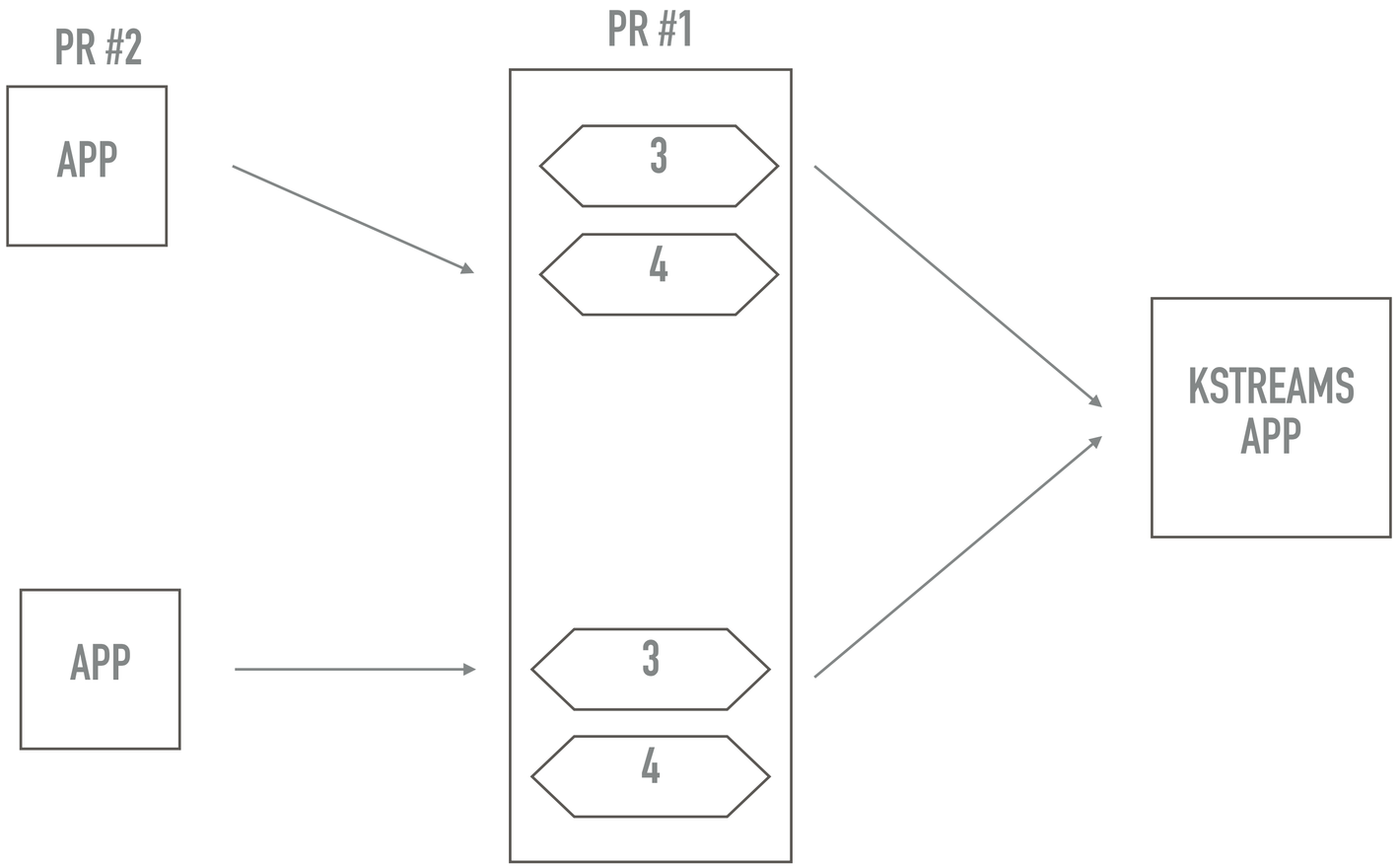
SCALE OUT



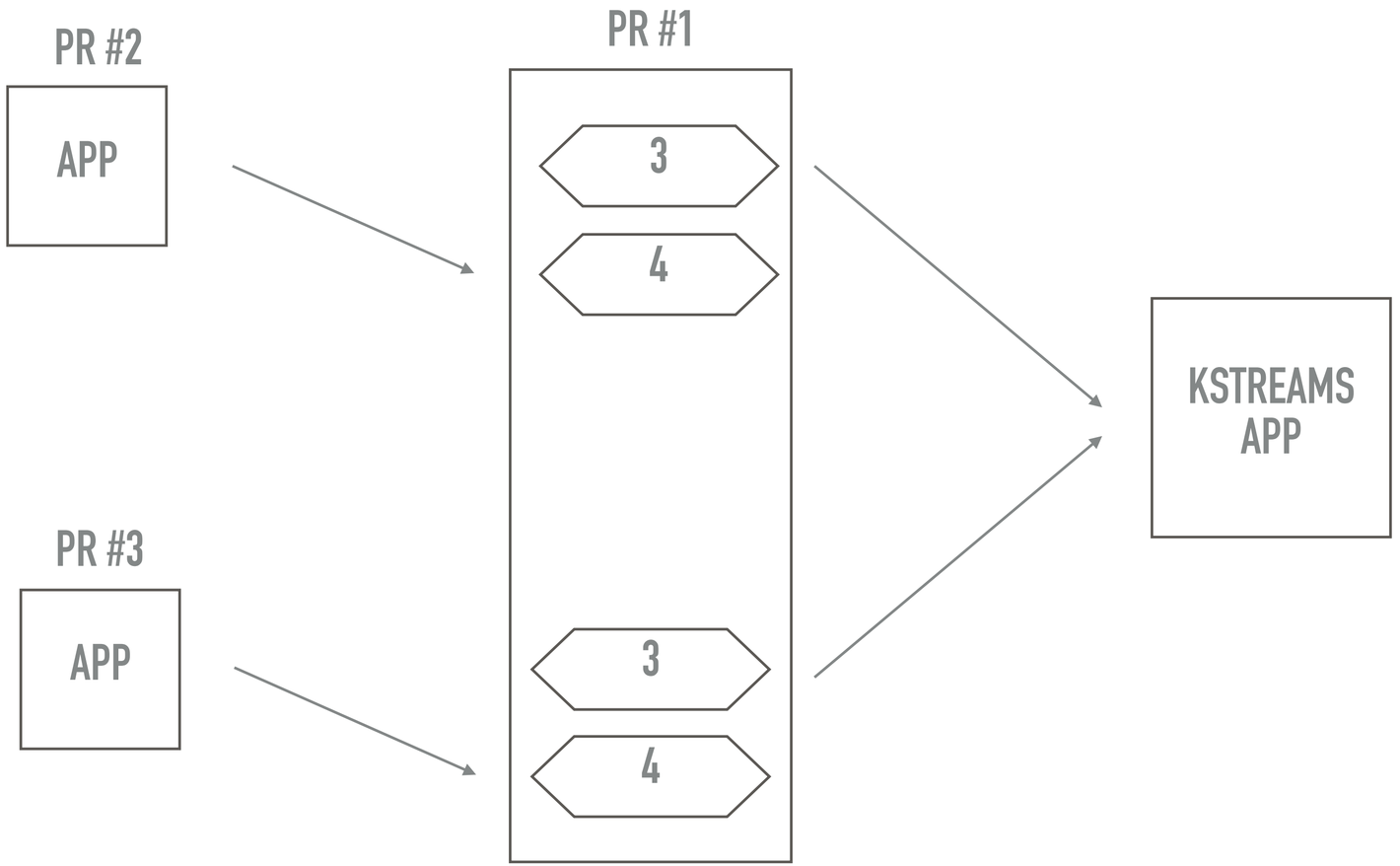
SCALE OUT



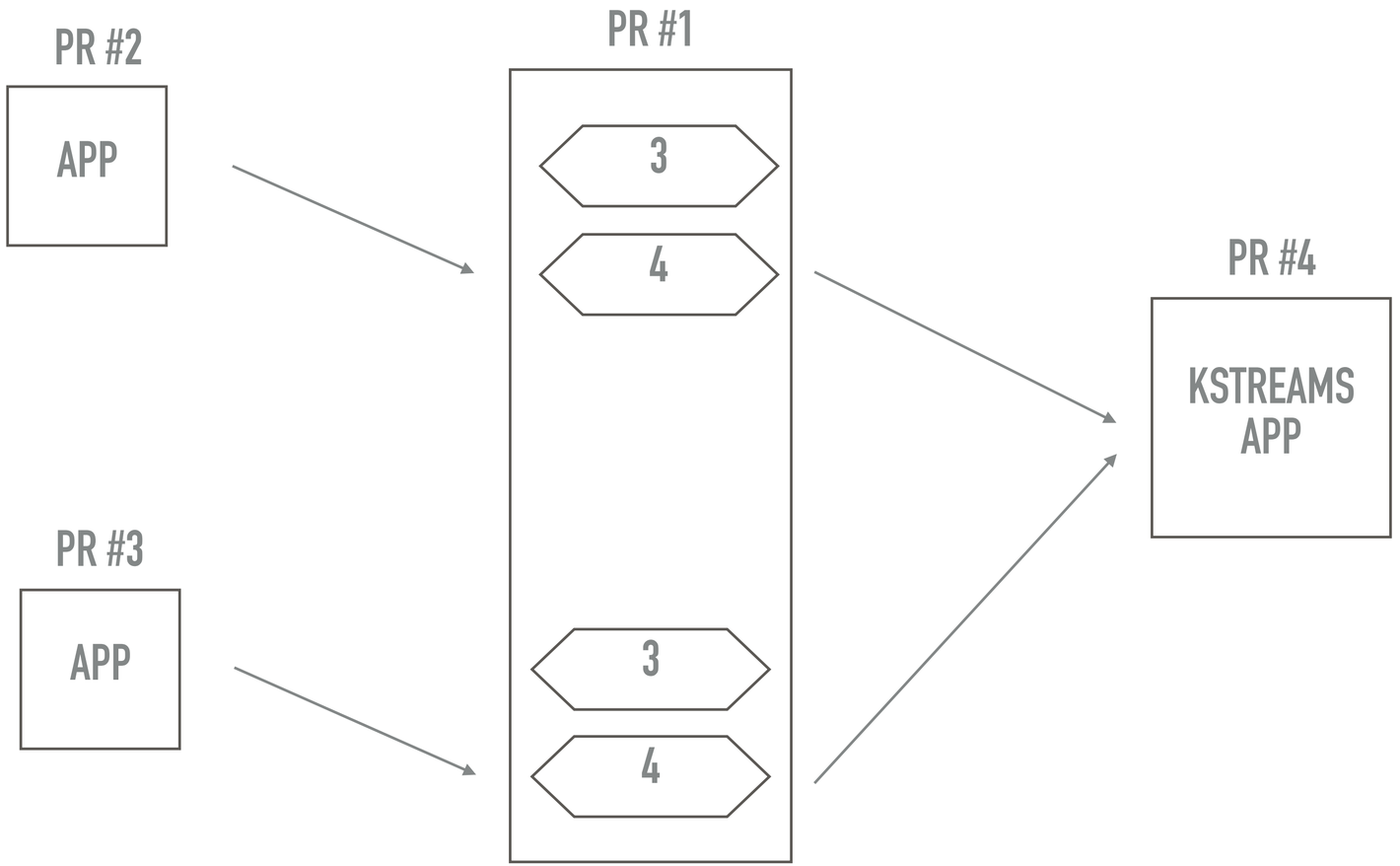
SCALE OUT



SCALE OUT

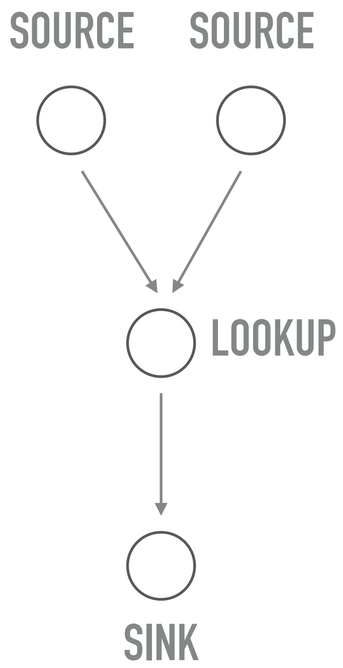


SCALE OUT

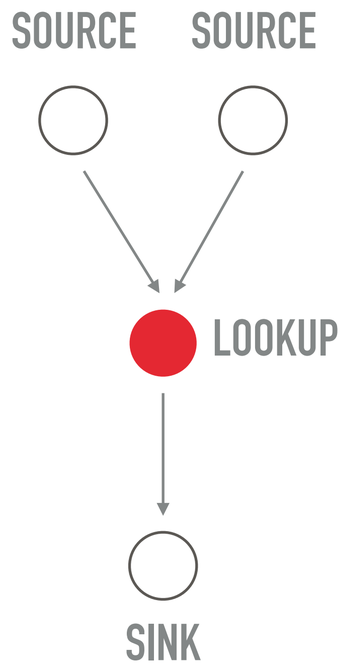


RUNBOOKS!

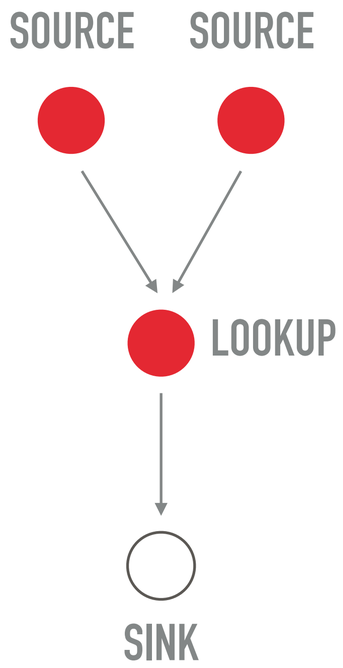
SCALE UP



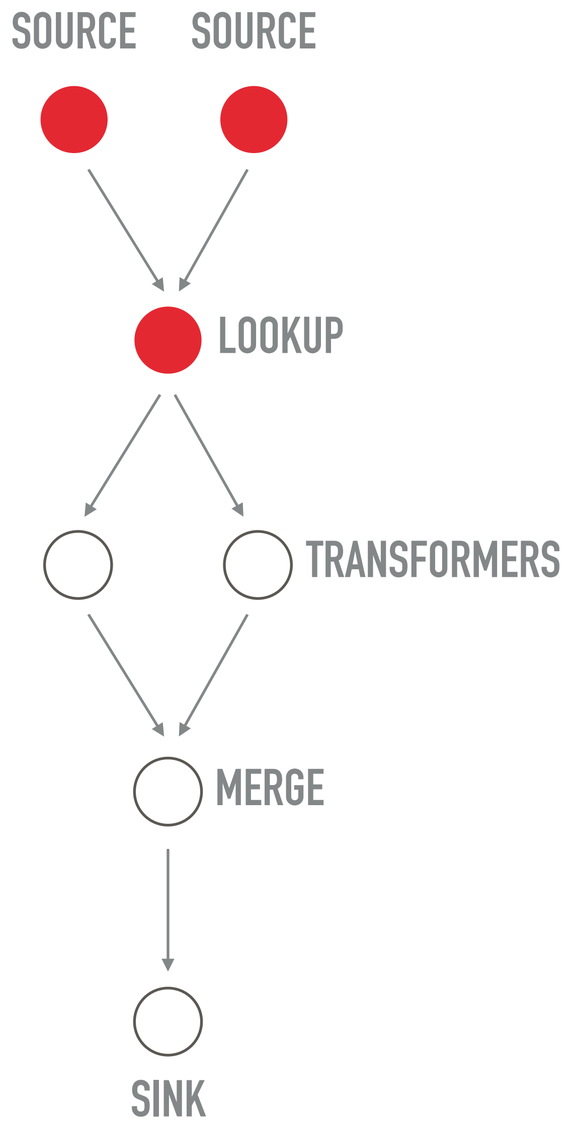
SCALE UP



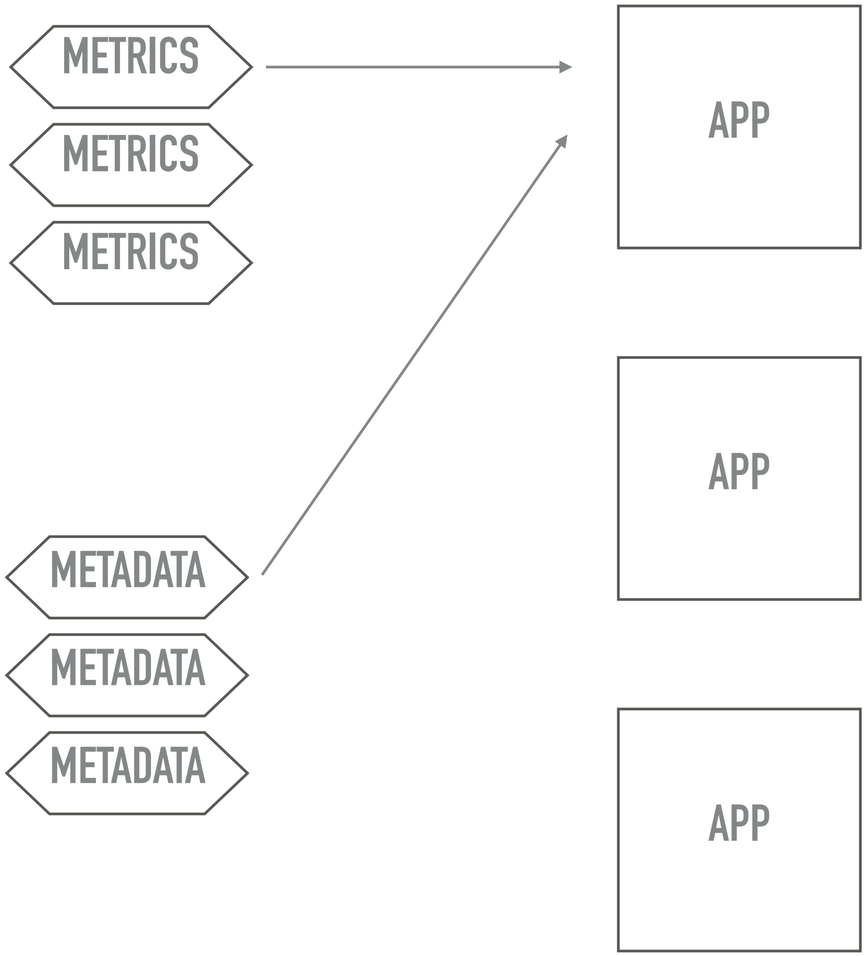
SCALE UP



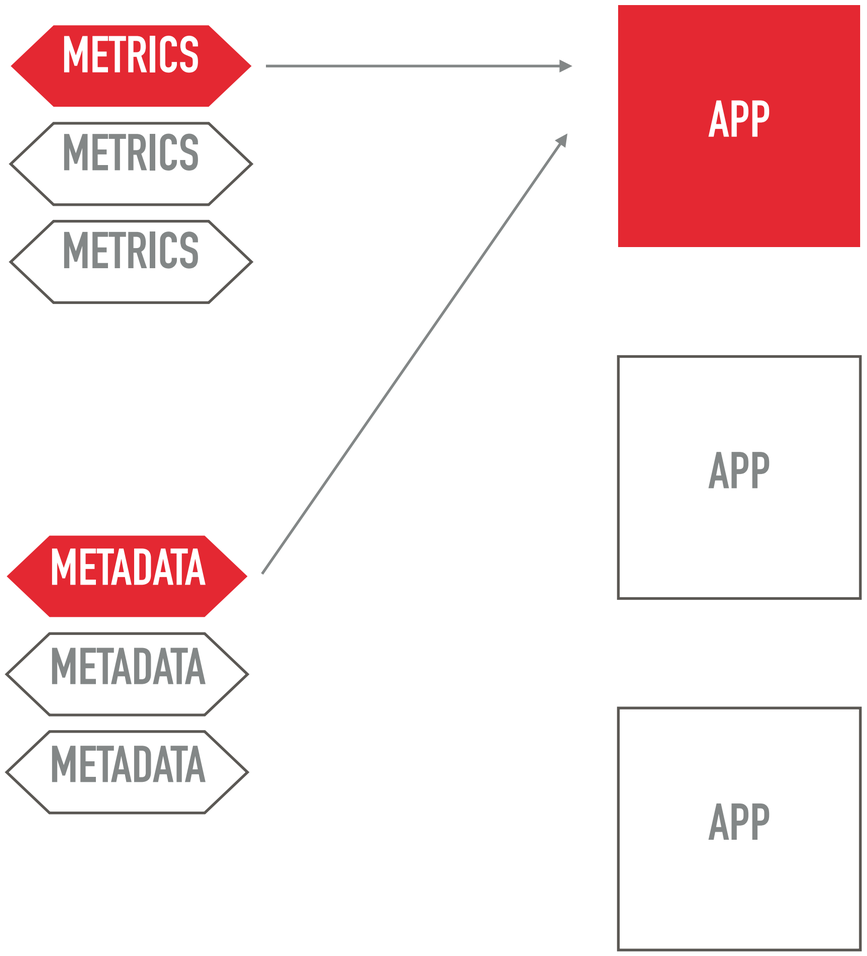
SCALE UP



SCALE UP


METRICS
METRICS
METRICS
METADATA
METADATA
METADATA
APP
APP
APP

SCALE UP


METRICS
METRICS
METRICS
METADATA
METADATA
METADATA
APP
APP
APP

SCALE UP

- don’t block
- if blocking, do it in a separate service
- distribute the load as uniform as possible (not by account id)

TOO GOOD TO BE TRUE…

- works reliably in production under load
- you really need to know Kafka, but that’s enough
- co-partitioning is the only magic dust
- watch new topics, there will be a lot

QUESTIONS?

CONCERNS?

PROTESTS?