Telefónica

ALTO integration with Kafka and future plans

Work in progress

Luis M. Contreras, Alejandro Muñiz

19/03/24 – IETF 119, Side meeting on “Information Exposure for Edge Computing”

ALTO integration with Kafka – Starting point

-

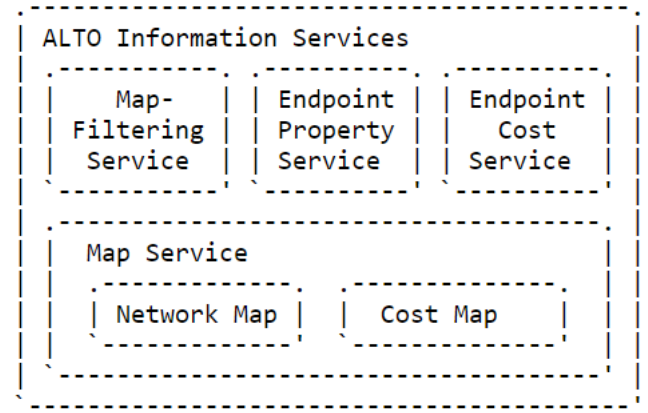


Figure 2: ALTO Information Service Framework

→

- Network Map
- Cost Map
- Endpoint Property Service
- Endpoint Cost Service
- Map-Filtering Service

To expose information:

One topic per service and maybe one cost-map topic per metric.

-

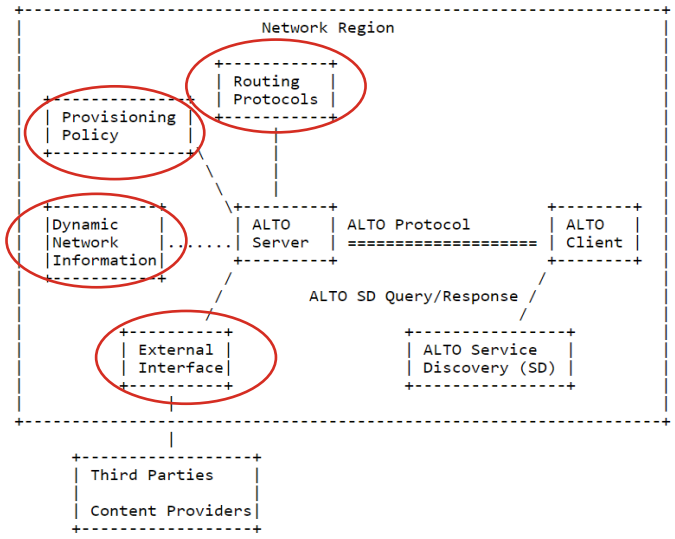


Figure 1: Basic ALTO Architecture

→

- Routing Protocol 1
- Dynamic Net Information
- Provisioning Policy
- Routing Protocol 2
- External Interface X

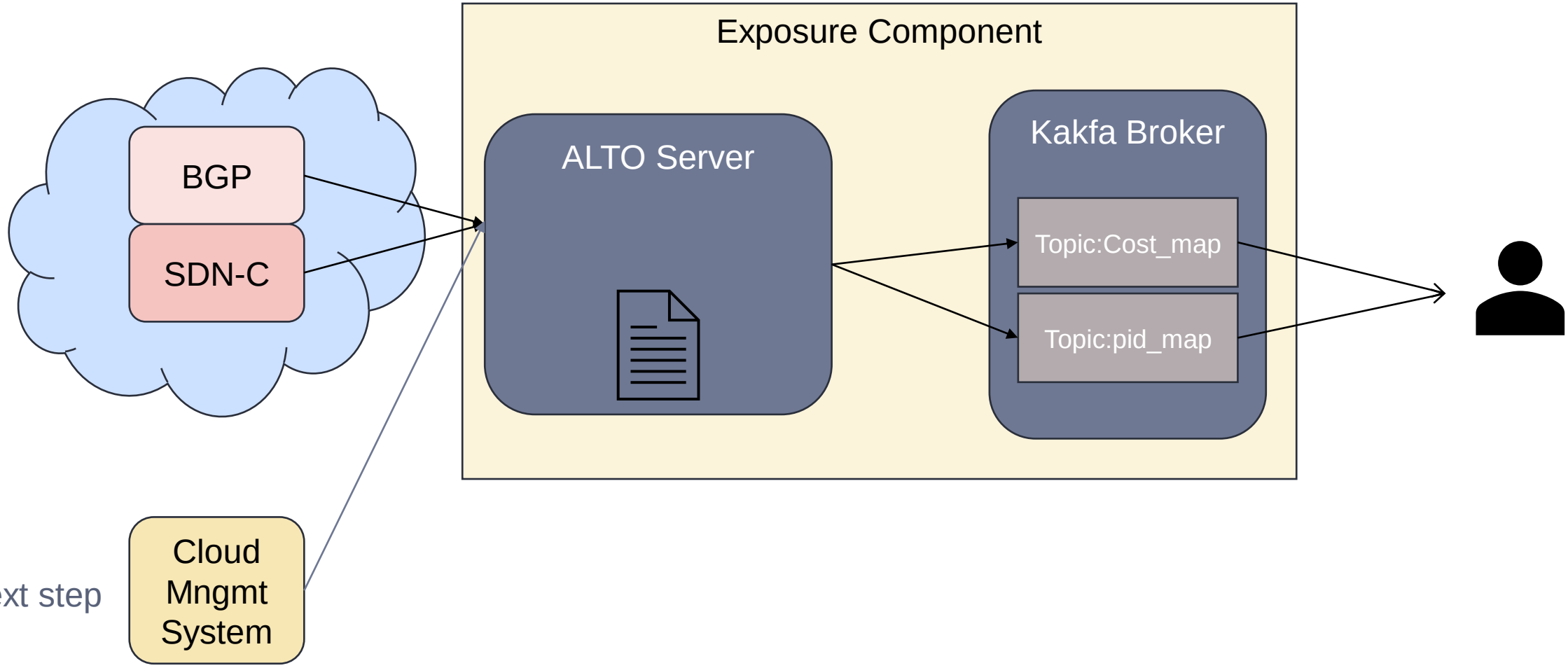
To receive information:

One topic per each type of source with one or more producers per network.

Kafka as tool for expossing information

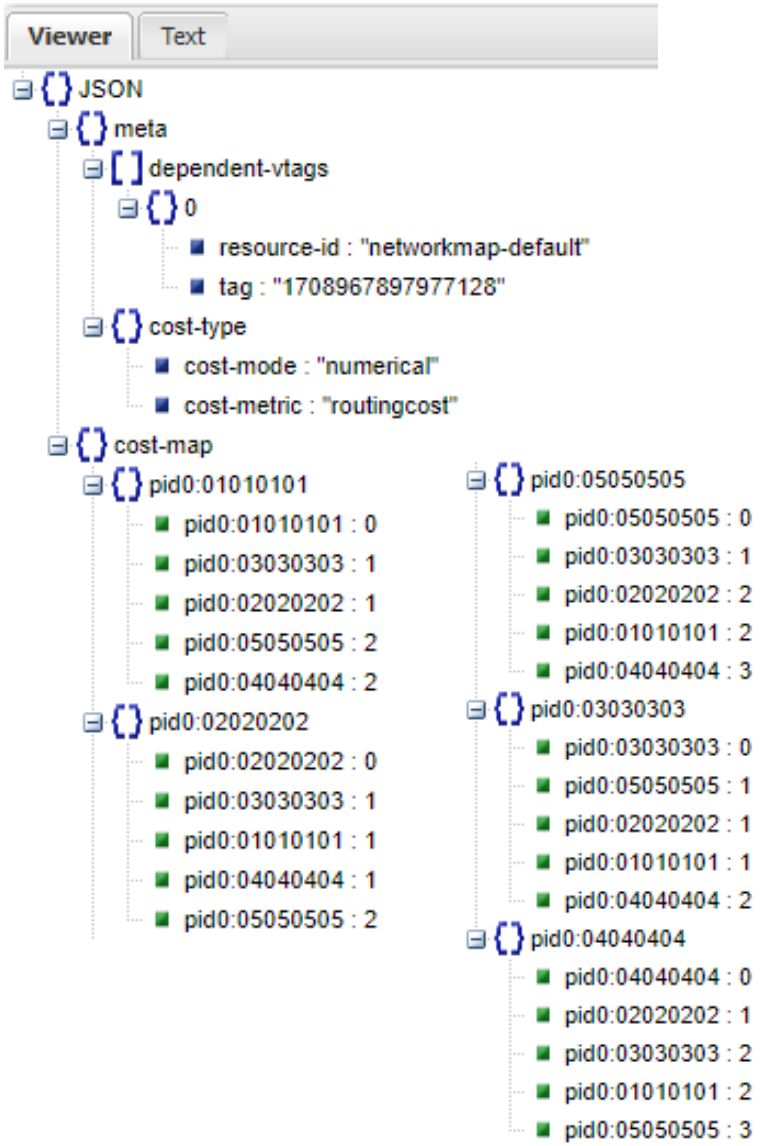
Work in progress

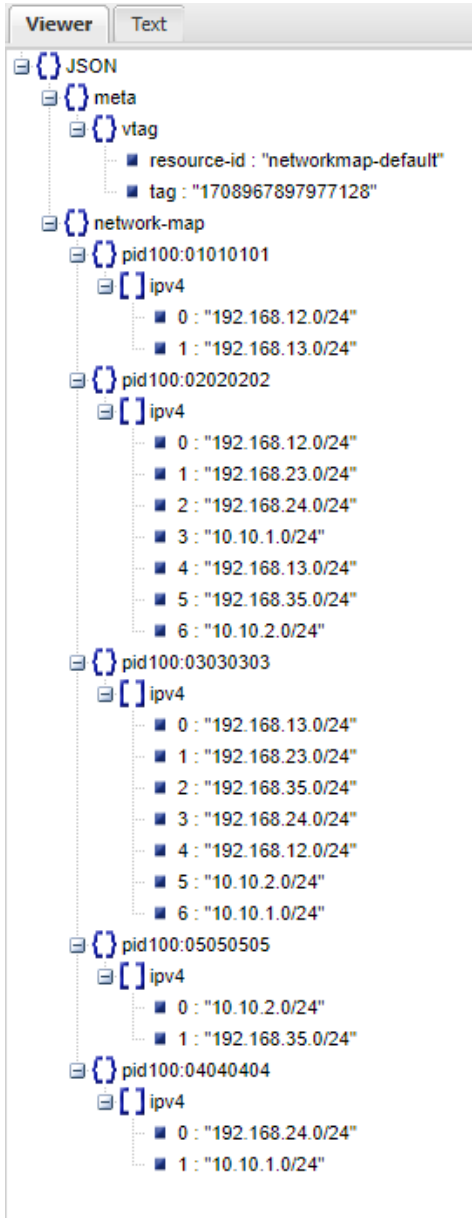
Next step



Scenario

Results:





First Step: Launching the environment

- Starting zookeeper and launching Kafka broker

```
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$ cat launcher
#!/bin/bash
# © 2024 Telefónica Innovación Digital, All rights reserved

bin/zookeeper-server-start.sh config/zookeeper.properties & #1>logs/launcher-ultima-actividad.log 2>>logs/launcher.log &
sleep 3
bin/kafka-server-start.sh config/server.properties #1>>logs/launcher-ultima-actividad.log 2>>logs/launcher.log &
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$
```

- Creating the services

```
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$ bin/kafka-topics.sh --create --topic metwork_maps --bootstrap-server localhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but
 not both.
Created topic metwork_maps.
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$ bin/kafka-topics.sh --create --topic cost_maps --bootstrap-server localhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but
 not both.
Created topic cost_maps.
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$
```

Second Step: Feeding the queues

- From ALTO code we launch a Kafka Producer and using the BGP based information obtained we feed the two queues created.

```
Received:1192 Bytes
1710335985882405
RecordMetadata(topic='network_maps', partition=0, topic_partition=TopicPartition(topic='network_maps', partition=0), offset=9,
 timestamp=1710335985883, log_start_offset=0, checksum=None, serialized_key_size=-1, serialized_value_size=521, serialized_hea
der_size=-1)
RecordMetadata(topic='cost_maps', partition=0, topic_partition=TopicPartition(topic='cost_maps', partition=0), offset=9, times
tamp=1710335985887, log_start_offset=0, checksum=None, serialized_key_size=-1, serialized_value_size=776, serialized_header_si
ze=-1)
Received:1192 Bytes
1710335985894872
RecordMetadata(topic='network_maps', partition=0, topic_partition=TopicPartition(topic='network_maps', partition=0), offset=10
, timestamp=1710335985895, log_start_offset=0, checksum=None, serialized_key_size=-1, serialized_value_size=521, serialized_he
ader_size=-1)
RecordMetadata(topic='cost_maps', partition=0, topic_partition=TopicPartition(topic='cost_maps', partition=0), offset=10, time
stamp=1710335985902, log_start_offset=0, checksum=None, serialized_key_size=-1, serialized_value_size=776, serialized_header_s
ize=-1)
```

Third Step: Connecting Consumers

- The consumer realizes the connection and starts receiving the data.

```
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$ bin/kafka-console-consumer.sh --topic cost_maps --bootstrap-server localhost:9092
{'meta':{'dependent-vtags': [{'resource-id': 'networkmap-default', 'tag': '1710336520127427'}], 'cost-type': {'cost-mode': 'nu
merical', 'cost-metric': 'routingcost'}}, 'cost-map': {'pid0:04040404': {'pid0:04040404': 0, 'pid0:02020202': 1, 'pid0:0101010
1': 2, 'pid0:03030303': 2, 'pid0:05050505': 3}, 'pid0:01010101': {'pid0:01010101': 0, 'pid0:02020202': 1, 'pid0:03030303': 1,
'pid0:04040404': 2, 'pid0:05050505': 2}, 'pid0:02020202': {'pid0:02020202': 0, 'pid0:01010101': 1, 'pid0:03030303': 1, 'pid0:0
4040404': 1, 'pid0:05050505': 2}, 'pid0:03030303': {'pid0:03030303': 0, 'pid0:01010101': 1, 'pid0:02020202': 1, 'pid0:05050505
': 1, 'pid0:04040404': 2}, 'pid0:05050505': {'pid0:05050505': 0, 'pid0:03030303': 1, 'pid0:01010101': 2, 'pid0:02020202': 2, '
pid0:04040404': 3}}}
{'meta':{'dependent-vtags': [{'resource-id': 'networkmap-default', 'tag': '1710336520135151'}], 'cost-type': {'cost-mode': 'nu
merical', 'cost-metric': 'routingcost'}}, 'cost-map': {'pid0:04040404': {'pid0:04040404': 0, 'pid0:02020202': 1, 'pid0:0101010
1': 2, 'pid0:03030303': 2, 'pid0:05050505': 3}, 'pid0:01010101': {'pid0:01010101': 0, 'pid0:02020202': 1, 'pid0:03030303': 1,
'pid0:04040404': 2, 'pid0:05050505': 2}, 'pid0:02020202': {'pid0:02020202': 0, 'pid0:01010101': 1, 'pid0:03030303': 1, 'pid0:0
4040404': 1, 'pid0:05050505': 2}, 'pid0:03030303': {'pid0:03030303': 0, 'pid0:01010101': 1, 'pid0:02020202': 1, 'pid0:05050505
': 1, 'pid0:04040404': 2}, 'pid0:05050505': {'pid0:05050505': 0, 'pid0:03030303': 1, 'pid0:01010101': 2, 'pid0:02020202': 2, '
pid0:04040404': 3}}}
Processed a total of 10 messages
```

Third Step: Connecting Consumers

```
ubuntu@alto-server:~/tests/alto-kafka/api/kafka_ale$ bin/kafka-console-consumer.sh --topic network_maps --bootstrap-server localhost:9092
{'meta':{'vtag': {'resource-id': 'networkmap-default', 'tag': '1710338351656189'}}, 'network-map': {'pid100:01010101': {'ipv4'
: ['192.168.12.0/24', '192.168.13.0/24']}, 'pid100:02020202': {'ipv4': ['192.168.12.0/24', '192.168.23.0/24', '192.168.24.0/24
', '192.168.13.0/24', '192.168.35.0/24']}, 'pid100:03030303': {'ipv4': ['192.168.13.0/24', '192.168.23.0/24', '192.168.35.0/24
', '192.168.24.0/24', '192.168.12.0/24']}, 'pid100:04040404': {'ipv4': ['192.168.24.0/24']}, 'pid100:05050505': {'ipv4': ['192
.168.35.0/24']}}}
{'meta':{'vtag': {'resource-id': 'networkmap-default', 'tag': '1710338351664834'}}, 'network-map': {'pid100:01010101': {'ipv4'
: ['192.168.12.0/24', '192.168.13.0/24']}, 'pid100:02020202': {'ipv4': ['192.168.12.0/24', '192.168.23.0/24', '192.168.24.0/24
', '192.168.13.0/24', '192.168.35.0/24']}, 'pid100:03030303': {'ipv4': ['192.168.13.0/24', '192.168.23.0/24', '192.168.35.0/24
', '192.168.24.0/24', '192.168.12.0/24']}, 'pid100:04040404': {'ipv4': ['192.168.24.0/24']}, 'pid100:05050505': {'ipv4': ['192
.168.35.0/24']}}}
Processed a total of 11 messages
```

Next step

- Integration of compute related information
  - First approach = compute resources

Ex. Openstack information


```
{
  "compute5":{
    "disk_available":"1833",
    "ram_available":"130520",
    "vcpu_available":"80",
    "disk_used":"984",
    "ram_used":"127296",
    "vcpu_used":"70",
    "instances":"29"
  },
  "compute4":{
    "disk_available":"1833",
    "ram_available":"130520",
    "vcpu_available":"80",
    "disk_used":"980",
    "ram_used":"190588",
    "vcpu_used":"86",
    "instances":"32"
  },
  "compute6":{
    "disk_available":"1833",
    "ram_available":"130520",
    "vcpu_available":"80",
    "disk_used":"966",
    "ram_used":"169472",
    "vcpu_used":"88",
    "instances":"43"
  },
……
```


- Open points
  - What kind of topics define: resources, metrics on resources, etc
  - How far the topics can be common per Compute Management System
  - How to integrate compute information on ALTO artifacts (PIDs, End-Point properties, etc)
  - How to deal with multi-homing for cloud environments connected to more than one PE
  - How to automatize the collection of compute information
  - etc

For further details, please contact:
luismiguel.contrerasmurillo@telefonica.com


Telefónica