

# **SST System Interface Guide**

## **Ecosystem interface guide**

Document version: 1.1

Document release date: 06/02/2020

Subject		
<b>SST Reference Guide [1.1]</b>		
Writer	Sign off and Date	Deadline for Comments
XpandIT	05/02/2020	

<b>RECIPIENTS</b>		
Name	Organization	Sign-off
Oscar	EMSA	

## TABLE OF CONTENTS

<b>STAR STREAMING OVERVIEW.....</b>	<b>7</b>
<b>1 AIVDM TCP SOURCE INTERFACE.....</b>	<b>8</b>
1.1 FUNCTIONALITY.....	8
1.2 AIVDM MESSAGES.....	8
1.2.1 <i>Message flow</i> .....	8
1.2.2 <i>Message description</i> .....	8
1.2.3 <i>Example message</i> .....	10
<b>2 JMS WEBLOGIC SINK CONNECTOR .....</b>	<b>11</b>
2.1 FUNCTIONALITY.....	11
2.2 FORMATTED AIS MESSAGES.....	11
2.2.1 <i>Message flow</i> .....	11
2.2.2 <i>Message description</i> .....	11
2.2.3 <i>Example message</i> .....	12
<b>3 REST ZIP SINK CONNECTOR .....</b>	<b>13</b>
3.1 FUNCTIONALITY.....	13
3.2 FORMATTED AIS MESSAGES.....	13
3.2.1 <i>Message flows</i> .....	13
3.2.2 <i>Message description</i> .....	13
3.2.3 <i>Example message</i> .....	14
<b>4 MONITOR STREAM .....</b>	<b>16</b>
4.1 FUNCTIONALITY.....	16
4.2 FORMATTED AIS MESSAGES.....	16
4.2.1 <i>Message flow</i> .....	16
4.2.2 <i>Message description (Consumed Messages)</i> .....	16
4.2.3 <i>Example message (Consumed Messages)</i> .....	17
4.2.4 <i>Message description (Produced Message)</i> .....	17
4.2.5 <i>Example message (Produced Message)</i> .....	18
<b>5 MIRROR MAKER.....</b>	<b>19</b>
5.1 FUNCTIONALITY.....	19
5.2 FORMATTED AIS MESSAGES.....	19
5.2.1 <i>Message flow</i> .....	19
5.2.2 <i>Message description</i> .....	19
5.2.3 <i>Example message</i> .....	20
<b>6 KAFKA TOPIC CONSUMER .....</b>	<b>21</b>
6.1 FUNCTIONALITY.....	21
6.2 CONSUMED MESSAGES .....	21
6.2.1 <i>Message flow</i> .....	21
6.3 IMPLEMENTATION.....	21
6.3.1 <i>Properties</i> .....	21
6.3.2 <i>Implementation</i> .....	23
<b>7 KSQL .....</b>	<b>26</b>
7.1 FUNCTIONALITY.....	26
7.2 KSQL WORKFLOW .....	26

7.2.1	Communication flow.....	26
7.2.2	KSQL Query Structure.....	26
7.2.3	KSQL Funcionality.....	26
7.2.4	Example queries.....	27
<b>8</b>	<b>AIVDM TCP SINK INTERFACE .....</b>	<b>29</b>
8.1	FUNCTIONALITY .....	29
8.2	TCP SINK WORKFLOW .....	29
8.2.1	Message flow.....	29
8.2.2	Message description (Kafka Topic to TCP Sink) .....	29
8.2.3	Example message (Kafka Topic to TCP Sink).....	30
8.2.4	Message description (TCP Sink to TCP Client) .....	30
8.2.5	Example message (TCP Sink to TCP Client) .....	30
<b>9</b>	<b>FILE SINK.....</b>	<b>31</b>
9.1	FUNCTIONALITY .....	31
9.2	FILE SINK WORKFLOW .....	31
9.2.1	Message flow.....	31
9.2.2	Message description (Kafka Topic to File Sink) .....	31
9.2.3	Example message (Kafka Topic to File Sink) .....	32
9.2.4	Message description (File Sink to File).....	32
9.2.5	Example message (File Sink to File) .....	32
<b>10</b>	<b>MONITOR METRICS SINK.....</b>	<b>33</b>
10.1	FUNCTIONALITY .....	33
10.2	MONITOR METRICS SINK WORKFLOW .....	33
10.2.1	Message flow.....	33
10.2.2	Message description (Kafka Topic to Monitor Metrics Sink) .....	33
10.2.3	Example message (Obtained Message).....	34
10.2.4	File Content (Originator count) .....	34
10.2.5	Example content (Produced by Monitor Metrics Sink).....	34
<b>11</b>	<b>AIS BACKLOG FILE SOURCE .....</b>	<b>36</b>
11.1	FUNCTIONALITY .....	36
11.2	AIS BACKLOG FILE SOURCE WORKFLOW .....	36
11.2.1	Message flow.....	36
11.2.2	Backlog file description .....	36
11.2.3	Backlog file example .....	36
11.2.4	TCP output description.....	36
11.2.5	Example TCP output.....	36
<b>12</b>	<b>JMS SOURCE CONNECTOR.....</b>	<b>37</b>
12.1	FUNCTIONALITY .....	37
12.2	JMS SOURCE WORKFLOW .....	37
12.2.1	Message flow.....	37
12.2.2	Example message (Obtained Message).....	37
12.2.3	Avro Schema .....	38
12.2.4	Example Avro Message.....	40
<b>13</b>	<b>REAL-TIME DASHBOARD INTERFACES .....</b>	<b>53</b>
13.1	FUNCTIONALITY .....	53
13.2	KAFKA TO ELASTICSEARCH WORKFLOW.....	53
13.2.1	Message Description (ValidMessage Avro).....	53
13.2.2	Example message (ValidMessage Avro) .....	53
13.2.3	Message Description (sst-events template).....	54

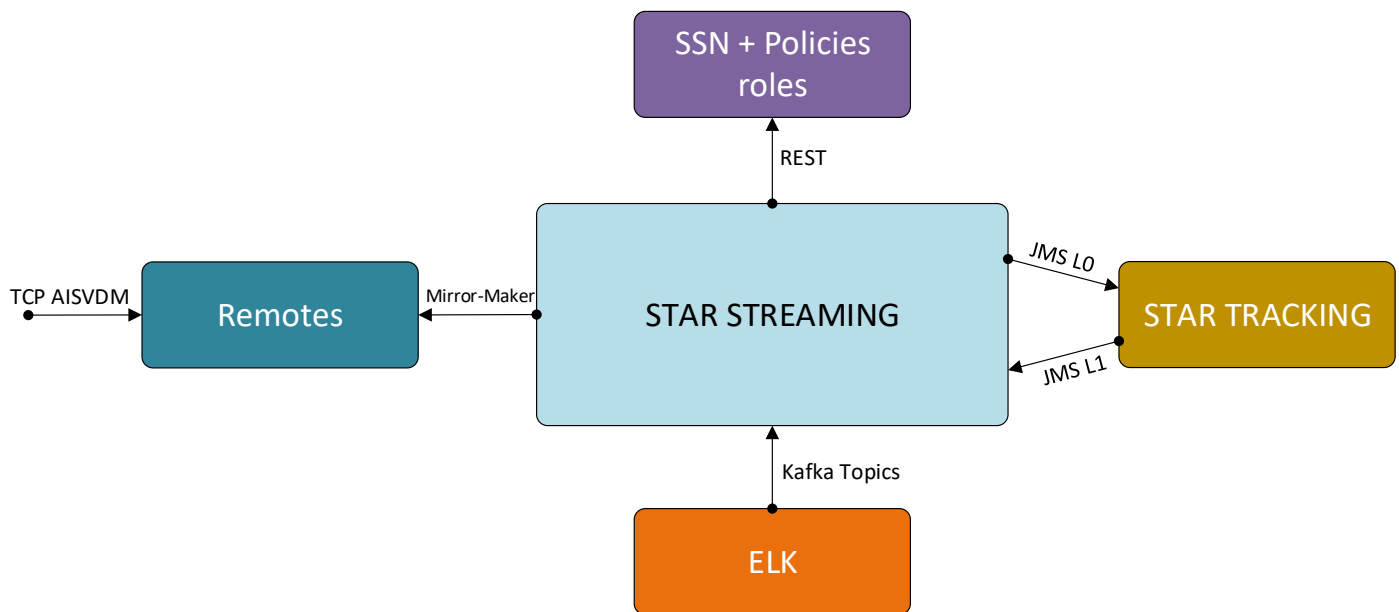
13.2.4	Example message (sst-events template).....	54
13.3	VISUALIZATION DATA FLOW .....	56
13.3.1	Elasticsearch REST endpoint .....	56
13.3.2	Request Body Description (json) .....	56
13.3.3	Request body Examples (json) .....	56
13.3.4	Response Description (json) .....	93
13.3.5	Response Examples (json) .....	94



## STAR STREAMING OVERVIEW

---

The Star Streaming platform communicates with other external systems through various interfaces (explained in detail further below). An overview of those interactions are presented in the following high level diagram:



In terms of usage of Kafka lingo, connectors that are called "sinks" are output interfaces and connectors that are called "sources" are input interfaces.

# 1 AIVDM TCP SOURCE INTERFACE

## 1.1 Functionality

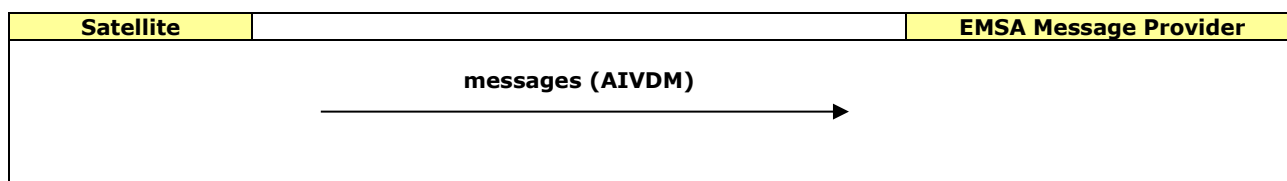
The AIS provider connects to the AIVDM TCP Source connector and sends AIVDM messages for transmission to EMSA.

## 1.2 AIVDM messages

Messages received by the TCP Connector are in the AIVDM format. The comment block extension defined in the current document allows providing additional information not encodable in NMEA format.

### 1.2.1 Message flow

The following figure outlines the expected synchronous flow of the raw message.



### 1.2.2 Message description

#### 1.2.2.1 AIVDM sentences following standard Comment Block

The information "i:" tag in the comment block has been extended in order to allow specifying additional information

The following table describes the raw message received by the TCP Connector.

Array	Value	Occ
vdmSentence		1
commentBlock		0-1
	time	0-1
	destination	0-1
	source	0-1
	information	0-1
	source	0-1
	dataQuality	0-1
	xml tag O	0-1
	xml tag S	0-1
	Xml tag Q	0-1
	dataProcessing	0-1
	qualityLevel	0-1

		dataOriginator		0-1
			rcc	1
			subs	0-3
			countryCode	1
		usagePolicy		0-1
			transmittedDataSensitivity	0-1
			billingChargeSystem	0-1
		recipients		0-10
			rcc	1
			subs	0-3
			countryCode	1
		satInfo		0-1
			groundStaionAcquisitionTs	0-1
			dataCentreIngestionTs	0-1
			dataCentreDeliveryTs	0-1
			satelliteId	0-1
			groundStationId	0-1
			foa	0-1
			toa	0-1
		positionValidationMethod		0-1
			kalmanFiltering	0-1
			correlations	0-1
			dopplers	0-1
		positionComputationMethod		0-1
			method	1
			values	1

The following XML schema describes completes the table above and allows to validate comment block contents.

```
<!-- W3C Schema generated by XMLSpy v2019 sp1 (x64) (http://www.altova.com) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="O">
    <!--Originator-->
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minInclusive value="1">
        <xs:maxInclusive value="3">
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Q">
    <xs:simpleType>
      <xs:restriction base="xs:byte">
        <xs:enumeration value="12"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

```

</xs:element>
<xs:element name="S">
  <!--Source-->
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="S"/><!--Satellite AIS-->
        <xs:enumeration value="T"/><!--Terrestrial AIS-->
        <xs:enumeration value="R"/><!--RPAS AIS-->
        <xs:enumeration value="V"/><!--Ship AIS-->
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="comment_block">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="S" minOccurs="0"/>
        <xs:element ref="Q" minOccurs="0"/>
        <xs:element ref="0" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

### 1.2.3 Example message

Container includes raw AIS Message.

```
!AIVDM,1,1,,B,7050QlAD:Go<E2Ush5@aPEQu;;;@,0*56
```

Container includes raw multiline AIS Message.

```
!AIVDM,2,1,3,A,577NrV02CT9QI81;B21<P4v1<P4r3N222222216H0Q:F6080B5Dp1k4p888,0*5D
!AIVDM,2,2,3,A,888888888880,2*27
```

Container includes AIVDM multiline AIS Message with comment block.

```
\\g:1-2-
1932,c:1474364472,s:106,i:<S>S</S><Q>12</Q><O>XLS</O>*31\\!AIVDM,2,1,3,A,577NrV02CT9QI81;B21<P4v
1<P4r3N222222216H0Q:F6080B5Dp1k4p888,0*5D
\\g:2-2-1932*54\\!AIVDM,2,2,3,A,888888888880,2*27
```

## 2 JMS WEBLOGIC SINK CONNECTOR

### 2.1 Functionality

The JMS Weblogic is waiting for new messages to be received at a certain topic, when these arrive are sent to the pre-defined Weblogic Server JMS queues. The destination queue is defined based on the type of the message that is consumed from the topic.

### 2.2 Formatted AIS messages

Messages that are obtained from the topics in the Central EMSA cluster can be in one of three different types:

- Raw
- JSON
- XML

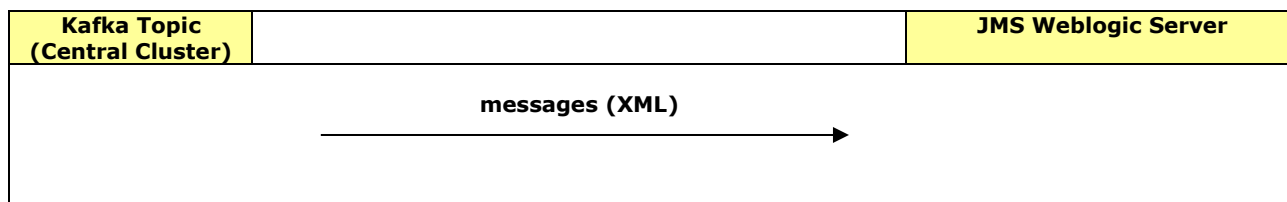
Since the desired functionality of the JMS Weblogic Connector is to send messages to a JMS queue in the XML format then the obtained format from the Avro Valid Message will be XML.

The XML Ais Messages can be one of two types, either Position (1,2,3,18,19 and 27) or Voyage (5 and 24).

Only the value obtained from the "formatted\_message" field will be sent to the JMS queue.

#### 2.2.1 Message flow

The following figure outlines the expected synchronous flow of the XML message.



#### 2.2.2 Message description

The following table describes the Valid Message Avro format obtained from the JMS Weblogic Connector from the topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1

mmsi	1
message_id	1
hub_id	1

### 2.2.3 Example message

Container includes avro AIS Message with the formatted message converted to XML.

```
{
  "unique_id": "861919401@1546970330000",
  "originator": "RUS",
  "source": "source-xpa",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "processed_timestamp": 1546970330002,
  "prev_topic_timestamp": 1546970330001,
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
  "formatted_message": "XML MESSAGE",
  "mmsi": "356091000",
  "message_id": 1,
  "hub_id": "hub-1"
}
```

Container includes the XML message from the example above that is sent to the JMS Queue.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EMSA source="Sat-AIS" id="1" timestamp="2019-01-08T18:00:15.613Z"
xmlns="http://schemas.emsa.europa.eu/cdf/position" xmlns:ns2="http://schemas.emsa.europa.eu/cdf"
xmlns:ns4="http://schemas.emsa.europa.eu/cdf/ais"
xmlns:ns3="http://schemas.emsa.europa.eu/cdf/voyage">
  <PositionMessage>
    <DataAccessRights>
      <ns2:Originator>RUS</ns2:Originator>
    </DataAccessRights>
    <ShipParticulars>
      <ns2:MMSI>356091000</ns2:MMSI>
    </ShipParticulars>
    <PositionReport timestamp="2019-01-08T17:58:50.000Z">
      <ns2:Latitude>30.4328666666666665622642540256492793560028076171875</ns2:Latitude>
      <ns2:Longitude>-177.4579500000000010959411156363785266876220703125</ns2:Longitude>
      <SOG unit="knots">11.4000000000000003552713678800500929355621337890625</SOG>
      <COG unit="decimal degrees">67</COG>
      <NavigationalStatus>0</NavigationalStatus>
      <TrueHeading unit="decimal degrees">65</TrueHeading>
      <RateOfTurn unit="decimal degrees">0</RateOfTurn>
    </PositionReport>
  </PositionMessage>
</EMSA>
```

## 3 REST ZIP SINK CONNECTOR

### 3.1 Functionality

The purpose of this interface is distributing AIS to legacy SSN Streaming Interface systems (SSN SI procies).

The Rest Zip Sink Connector is waiting for new messages to be received at a certain topic, when these arrive and a certain requirement is fulfilled these are grouped, zipped and sent to an endpoint. The requirements for the messages to be zipped are:

- Max number of messages obtained.
- Max interval of time between ZIP's has been reached.

### 3.2 Formatted AIS messages

Messages that are obtained from the topics in the Central EMSA cluster can be in one of three different types:

- Raw
- JSON
- XML

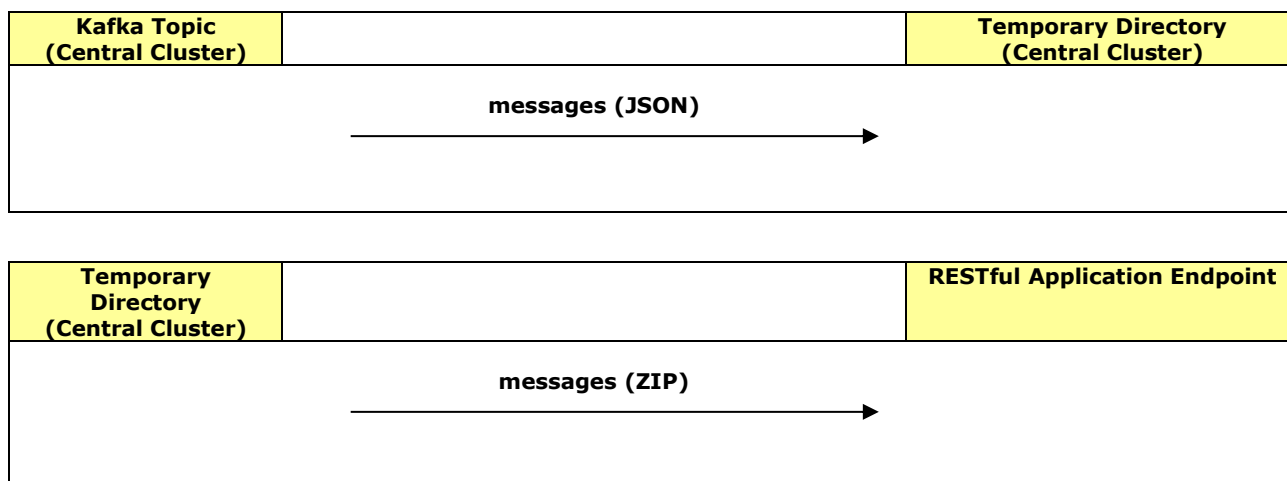
Since the desired functionality of the Rest Zip Connector is to group AIS Messages in the JSON format and then send the zip to an endpoint, then the desired format type is JSON.

The JSON Ais Messages can be one of the 27 types.

Only the value obtained from the "formatted\_message" field will be sent to the JMS queue.

#### 3.2.1 Message flows

The following figure outlines the expected synchronous flows of the JSON and Zipped messages.



#### 3.2.2 Message description

The following table describes the Valid Message Avro format obtained from the Rest Zip Connector from the topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

### 3.2.3 Example message

Container includes Avro AIS Message with the formatted message converted to JSON.

```
{
  "unique_id": "861919401@1546970330000",
  "originator": "RUS",
  "source": "source-xpa",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "processed_timestamp": 1546970330002,
  "prev_topic_timestamp": 1546970330001,
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
  "formatted_message": "JSON MESSAGE",
  "mmsi": "356091000",
  "message_id": 1,
  "hub_id": "hub-1"
}
```

Container includes the JSON message from the example above that is sent to the temporary directory to be zipped.

```
{
  "ssnsiName": "SSNSI_NAME",
  "requestor": "MEDSERVER",
  "messages": [
    {
      "reportedVessel": {
        "mmsi": 356091000, "flagRegistryCode": "PA"
      },
      "positionReport": {
        "rateOfTurn": 0,
        "aisMessageType": "1",
        "heading": 65,
        "latitude": 30.432866666666666,
        "courseOverGround": 67.0,
        "epfdStatus": 3,
        "speedOverGround": 11.4,
        "timeL": 1547032332,
        "source": "source-xpa",
        "navigationalStatus": 0,
        "longitude": -177.45795,
        "raimFlag": false
      },
      "originalNmeas": [
        "!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63"
      ],
      "originator": "RU",
    }
  ]
}
```

```
    "vdmSentence": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
    "commentBlock":
    {
        "information":
        {
            "dataOriginator":
            {
                "rcc": "RUS",
                "countryCode": "RU"
            }
        }
    },
    "originator": "RUS",
    "source": "source-xpa"
}
```

## 4 MONITOR STREAM

### 4.1 Functionality

The monitor stream consumes messages from Kafka Topics and produces messages with the Monitor Avro Schema to the monitor topic. Messages are processed and generate metrics that will be used as fields of the monitor message.

Monitor streams can consume from topics with either the Generic Avro Schema or Valid Avro Schema.

### 4.2 Formatted AIS messages

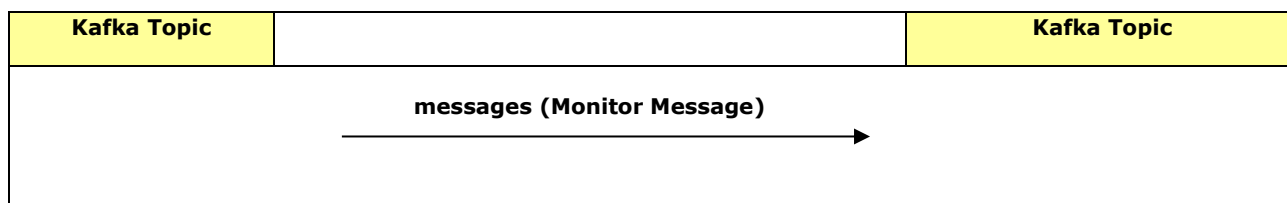
Message obtained are in one of two formats:

- Generic Avro Format
- Valid Avro Format

Messages produced will have the Monitor Avro Schema, and if the topic from where the monitor stream consumed the message has the Generic Avro Format both the MMSI or the Message Id can't be obtained since the message has not yet been validated, therefore both fields will be set with pre-defined values.

#### 4.2.1 Message flow

The following figure outlines the expected synchronous flow of the Avro Monitor Message.



#### 4.2.2 Message description (Consumed Messages)

The following table describes the Avro Generic Message consumed from the kafka topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
hub_id	1

The following table describes the Avro Valid Message consumed from the kafka topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

### 4.2.3 Example message (Consumed Messages)

Container includes Avro Generic Message consumed from the topic.

```
{
  "unique_id": "861919401@1546970330000",
  "originator": "RUS",
  "source": "source-xpa",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "processed_timestamp": 1546970330002,
  "prev_topic_timestamp": 1546970330001,
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
  "hub_id": "hub-1"
}
```

Container includes Avro Valid Message consumed from the topic.

```
{
  "unique_id": "861919401@1546970330000",
  "originator": "RUS",
  "source": "source-xpa",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "processed_timestamp": 1546970330002,
  "prev_topic_timestamp": 1546970330001,
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
  "formatted_message": "",
  "mmsi": "356091000",
  "message_id": 1,
  "hub_id": "hub-1"
}
```

### 4.2.4 Message description (Produced Message)

The following table describes the Avro Monitor Message produced from the kafka topic.

Value	Occ
unique_id	1
timestamp	1
creation_timestamp	1
received_timestamp	1

prev_topic_timestamp	1
source	1
originator	1
input_topic	1
mmsi	1
message_id	1
hub_id	1

### 4.2.5 Example message (Produced Message)

Container includes Avro Monitor Message producer after consuming a message with the Generic Avro Schema.

```
{ "unique_id": "861919401@1546970330000",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "received_timestamp": 1546970360001,
  "prev_topic_timestamp": 1546970330000,
  "source": "source-xpa",
  "originator": "RUS",
  "input_topic": "input-topic",
  "mmsi": "0000000000",
  "message_id": 0,
  "hub_id": "hub-1" }
```

Container includes Avro Monitor Message producer after consuming a message with the Valid Avro Schema.

```
{ "unique_id": "861919401@1546970330000",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "received_timestamp": 1546970360002,
  "prev_topic_timestamp": 1546970330001,
  "source": "source-xpa",
  "originator": "RUS",
  "input_topic": "valid-topic",
  "mmsi": "356091000",
  "message_id": 1,
  "hub_id": "hub-1" }
```

## 5 MIRROR MAKER

### 5.1 Functionality

The Mirror Maker will copy the messages on a specific topic and send them to a topic on another host. In terms of functionality it will work like a Kafka Consumer and a Producer.

### 5.2 Formatted AIS messages

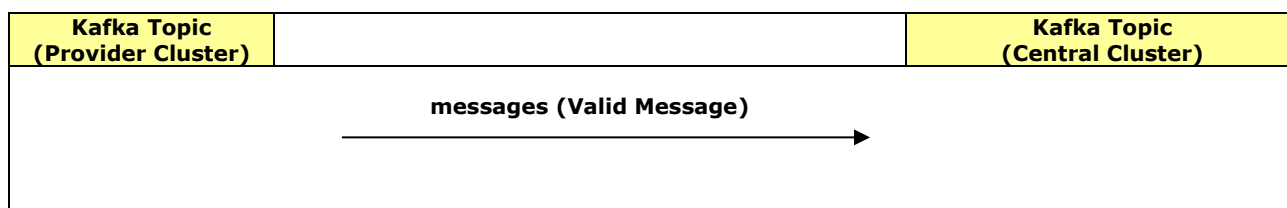
The replicated message between hosts has the Valid Message Avro Format.

Messages that are obtained from the topic in the provider host to be replicated will have the formatted message empty since it has not yet entered a filtering process.

The format passed is still empty since this will not add overhead to the process and most times the message is not yet ready to be delivered.

#### 5.2.1 Message flow

The following figure outlines the expected synchronous flow of the Avro Valid Message.



#### 5.2.2 Message description

The following table describes the Avro Valid Message from the topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

### 5.2.3 Example message

Container includes Avro Valid Message.

```
{ "unique_id": "861919401@1546970330000",  
  "originator": "RUS",  
  "source": "source-xpa",  
  "timestamp": 1546970330000,  
  "creation_timestamp": 1546970330001,  
  "processed_timestamp": 1546970330002,  
  "prev_topic_timestamp": 1546970330001,  
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",  
  "formatted_message": "",  
  "mmsi": "356091000",  
  "message_id": 1,  
  "hub_id": "hub-1" }
```

## 6 KAFKA TOPIC CONSUMER

---

### 6.1 Functionality

A JVM Kafka Consumer that is able to connect to a topic a consume messages from it.

The consumer can be a simple type key/value consumer and use the apache.kafka.common deserializers. This type of consumer can be used in most platforms that support kafka.

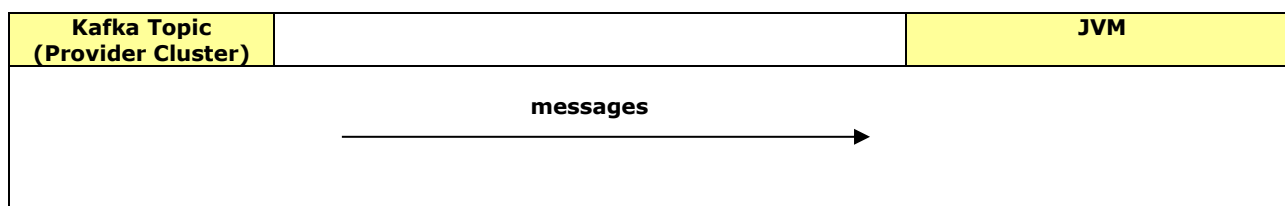
It can be a more complex consumer and use avro classes, in that case the consumer can only be used in the confluent platform (needs access to the confluent schema registry and also uses the confluent avro deserializers).

### 6.2 Consumed messages

The consumed messages format depends on the implementation of the consumer, it can be a plaintext key/value pair or it can be an avro key/value pair.

#### 6.2.1 Message flow

The following figure outlines the expected synchronous flow of the consumed messages.



### 6.3 Implementation

#### 6.3.1 Properties

For the consumer to be able to consume the messages on a desired topic a few properties have to be passed, either implicitly in the java code or via a properties file.

##### 6.3.1.1 Required Properties

##### **bootstrap.servers**

Is a comma-separated list of host and port pairs that are the addresses of the Kafka brokers.

##### **group.id**

This property defines the consumer group id, multiple consumers can be combined in a group and consume messages in a parallelized manner. The level of parallelization depends on the partitions of the Kafka topic consumed.

##### **key.deserializer**

The java class for deserialization of the message key. For example if the key is defined as a String the class passed should be "org.apache.kafka.common.serialization.StringDeserializer".

**(This value is defined in the java code as it should not change)**

##### **value.deserializer**

The java class for deserialization of the message value. For example if the value is defined as an Integer the class passed should be "org.apache.kafka.common.serialization.IntegerDeserializer".

**(This value is defined in the java code as it should not change)**

### **6.3.1.2 Optional Properties**

#### **schema.registry.url**

The host and port address of the confluent schema registry. This property is used when consuming from a topic with a schema associated (in the confluent platform), the consumer will not be able to consume messages from the topic if this property is not passed.

#### **specific.avro.reader**

If we are consuming from a topic with an avro schema associated this property should be set as true. In that case the deserializer class for the key/value should be set accordingly. For example if the value of the message is an avro class the "value.deserializer" should be set to "io.confluent.kafka.serializers.KafkaAvroDeserializer".

**(This value is defined in the java code as it should not change)**

#### **enable.auto.commit**

If this property is set as true the consumer will automatically commit the offsets of the messages consumed to the Kafka brokers, otherwise the offsets must be committed manually.

#### **auto.offset.reset**

This property enable the consumer to read either from a pre-defined offset:

- The "latest" committed offset will enable the consumer to read only "new" messages (default value).
- The "earliest" offset committed will read all the available messages in the topic from the oldest committed offset.
- "none" will throw exception to the consumer if no previous offset is found for the consumer's group.

### **6.3.1.3 Security Properties**

This is also considered an optional set of properties and should only be used if the brokers are using the SSL security protocol to access the topics.

#### **security.protocol**

Should be filled with "ssl" if it is indeed the security protocol being used, otherwise ignore the property or set it as "plaintext".

#### **ssl.truststore.location**

The location path to the ssl truststore (should only be filled when the security.protocol=ssl).

#### **ssl.truststore.password**

The ssl truststore password (should only be filled when the security.protocol=ssl).

#### **ssl.keystore.location**

The location path to the ssl keystore (should only be filled when the security.protocol=ssl).

#### **ssl.keystore.password**

The ssl keystore password (should only be filled when the security.protocol=ssl).

#### **ssl.key.password**

The ssl key password (should only be filled when the security.protocol=ssl).

### **6.3.1.4 Example of a properties file**

Properties file passed to the Java Kafka Consumer.

```
#application
group.id=test-group

#services
bootstrap.servers=remote-hub.emsa.europa.eu:9092
schema.registry.url=http://remote-hub.emsa.europa.eu:8081

#channels
topic=test-topic

#options
enable.auto.commit=true
auto.offset.reset=latest

#Security
security.protocol=ssl
ssl.truststore.location=C:/Users/Xpand-it/IdeaProjects/main/ais-kafka-
consumer/src/main/resources/kafka.truststore.jks
ssl.truststore.password=6qVcjL8TeKraQFRH
ssl.keystore.location=C:/Users/Xpand-it/IdeaProjects/main/ais-kafka-
consumer/src/main/resources/kafka.keystore.jks
ssl.keystore.password=6qVcjL8TeKraQFRH
ssl.key.password=6qVcjL8TeKraQFRH
```

## **6.3.2 Implementation**

### **6.3.2.1 Creating the Java Kafka Consumer**

To create a java kafka consumer we can pass as an argument to the JVM the properties file and with this file construct a Configuration class that the consumer will use.

Then we can simply access the properties in the Configuration file to use as the properties of the consumer.

Creating a Java Kafka Consumer with the provided properties.

```
public AvroConsumer(Configuration configuration){
    this.configuration = configuration;
}

public void run(){
    //consumer properties
    Properties props = new Properties();
```

```
props.put("bootstrap.servers", this.configuration.bootstrapServers);
props.put("schema.registry.url", this.configuration.schemaRegistryUrl);
props.put("group.id", this.configuration.groupId);
//string inputs and outputs deserializers
props.put("key.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");
props.put("value.deserializer", "io.confluent.kafka.serializers.KafkaAvroDeserializer");
props.put("specific.avro.reader", true);

...

//kafka consumer object
KafkaConsumer<String, ValidMessage> consumer = new KafkaConsumer<String, ValidMessage>(props);
```

In the provided snippet the Kafka Consumer will consume message with a key with a common type (String) and a value as a complex type (Valid Message Avro Class). The deserialization properties were set accordingly.

**Note:** The ValidMessage class was generated via the maven avro plugin, the avsc schema is placed in a pre-defined directory and the maven plugin will generate the java classes from the schema.

### 6.3.2.2 Subscribing to the Kafka Topic

The above implementation only creates a Kafka Consumer class with pre-defined key and value types.

The consumer will still have to subscribe to a kafka topic.

A Java Kafka Consumer subscribing to a Kafka Topic.

```
public void run(){

    ...

    //kafka consumer object
    KafkaConsumer<String, ValidMessage> consumer = new KafkaConsumer<>(props);

    Consumer.subscribe(Arrays.asList(this.configuration.inputTopic));
```

The subscribe method can take several topics and when subscribed to a particular set of topics each time the poll method is called the consumer will obtain the available records in each of these topics.

The signature of the consumer must be specified taking into consideration the topics being consumed. If the defined type for the consumer key/value is not

### 6.3.2.1 Consuming from the Kafka Topics

A Java Kafka Consumer doing poll to the subscribed topics (infinite loop)

```
while (true) {
    ConsumerRecords<String, ValidMessage> records = consumer.poll(100);
    for (ConsumerRecord<String, ValidMessage> record : records){
        String key = record.key();
        ValidMessage value = record.value();

        ...

    }
}
```

The poll method will fetch all the available messages in the subscribed topics, these messages are defined as a ConsumerRecord and each one of these is composed by a key (accessed via the ".key()" method) and a value (accessed via the ".value()" method).

If either the key or value is defined as an avro class the return values can be directly converted to the correspondent java class.

The process done to the received records at each call of poll is defined by the developer.

### **6.3.2.2 Running the Kafka Consumer JVM**

The main class of the JVM Kafka Consumer.

```
public static void main(String[] args) {  
  
    Configuration config = loadConfiguration(args[0]);  
  
    AvroConsumer consumer = new AvroConsumer(config);  
    consumer.run();  
  
}
```

The load configurations method should receive a properties file as an argument and produce a Configurations object where each property is mapped to the corresponding value.

The ".run()" method will then start the connector and poll the topics until the JVM process is stopped.

Running the JVM Kafka Consumer on a terminal.

```
$ java -jar ais-kafka-consumer-1.0.0.jar consumer.properties
```

## 7 KSQL

### 7.1 Functionality

The KSQL client allows the user to query the topics with common SQL queries and obtain the desired information. The KSQL server provides a REST API at port 8088/tcp that allow the user to make queries, these can be done directly via the rest endpoint or via the KSQL Client that can be accessed via the terminal.

### 7.2 KSQL Workflow

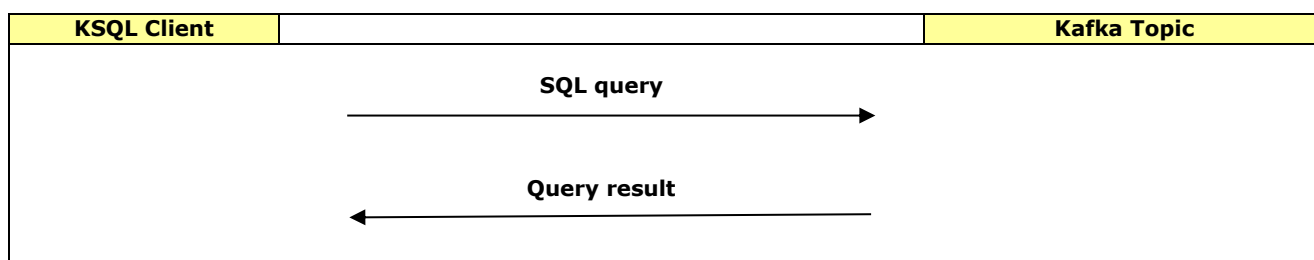
The KSQL client can query any topic and obtain information based on the returned output.

Unlike the common SQL queries that are applied Relational Databases the KSQL will only return information based on the records contained in the kafka topic. Since new records may arrive to the kafka topic at any point in time the KSQL query will return the computed values for each query from time to time.

Each kafka topic has a retention time that defines the total time that a record will reside in a topic, after this time has passed the record will be removed from the topic and will no longer be used to compute que query results.

#### 7.2.1 Communication flow

The following figure outlines the expected synchronous flow of the KSQL Client.



#### 7.2.2 KSQL Query Structure

KSQL Query
<pre> SELECT select_expr [, ...] FROM from_item [ LEFT JOIN join_table ON join_criteria ] [ WINDOW window_expression ] [ WHERE condition ] [ GROUP BY grouping_expression ] [ HAVING having_expression ] [ LIMIT count ]; </pre>

#### 7.2.3 KSQL Funcionality

A **LIMIT** can be used to limit the number of rows returned. Once the limit is reached the query will terminate.

If you want to select older data, you can configure KSQL to query the stream from the beginning. You must run this configuration before running the query:

```
SET 'auto.offset.reset' = 'earliest';
```

The command above will allow KSQL to query every record that resides in a Kafka Topic, if the property is not set to 'earliest' it will only present results for records that arrive at the topic after que KSQL query is deployed.

KSQL supports the following WINDOW types:

- **TUMBLING**: Tumbling windows group input records into fixed-sized, non-overlapping windows based on the records' timestamps. You must specify the window size for tumbling windows. Note: Tumbling windows are a special case of hopping windows where the window size is equal to the advance interval.
- **HOPPING**: Hopping windows group input records into fixed-sized, (possibly) overlapping windows based on the records' timestamps. You must specify the window size and the advance interval for hopping windows.
- **SESSION**: Session windows group input records into so-called sessions. You must specify the session inactivity gap parameter for session windows. For example, imagine you set the inactivity gap to 5 minutes. If, for a given record key such as "provider\_#", no new input data arrives for more than 5 minutes, then the current session for "provider\_#" is closed, and any newly arriving data for "provider\_#" in the future will mark the beginning of a new session.

## 7.2.4 Example queries

Container includes KSQL query.

```
SELECT * FROM pageviews_enriched;
```

Container includes KSQL response.

```
1519746861328 | User_4 | User_4 | Page_58 | Region_5 | OTHER
1519746861794 | User_9 | User_9 | Page_94 | Region_9 | MALE
1519746862164 | User_1 | User_1 | Page_90 | Region_7 | FEMALE
^CQuery terminated
```

Container includes KSQL query with LIMIT.

```
SELECT gender, regionid, numusers FROM pageviews_regions LIMIT 5;
```

Container includes KSQL response.

```
FEMALE | Region_6 | 3
FEMALE | Region_1 | 4
FEMALE | Region_9 | 6
MALE | Region_8 | 2
```

OTHER | Region\_5 | 4  
LIMIT reached  
Query terminated

**Container includes KSQL query with WINDOW.**

```
SELECT gender, regionid , COUNT(*) AS numusers FROM pageviews_enriched
WINDOW TUMBLING (size 30 second)
GROUP BY gender, regionid
HAVING COUNT(*) > 1;
```

## 8 AIVDM TCP SINK INTERFACE

### 8.1 Functionality

The TCP sink is designed to consume messages from a pre-defined topic and send them to client that connect to the port where the TCP Sink is waiting for connections.

When a client connects to the port it will send all the messages that appear from that moment onwards on the topic that is defined in the properties.

The TCP Sink will only work for topics that have the Valid Message Avro Format.

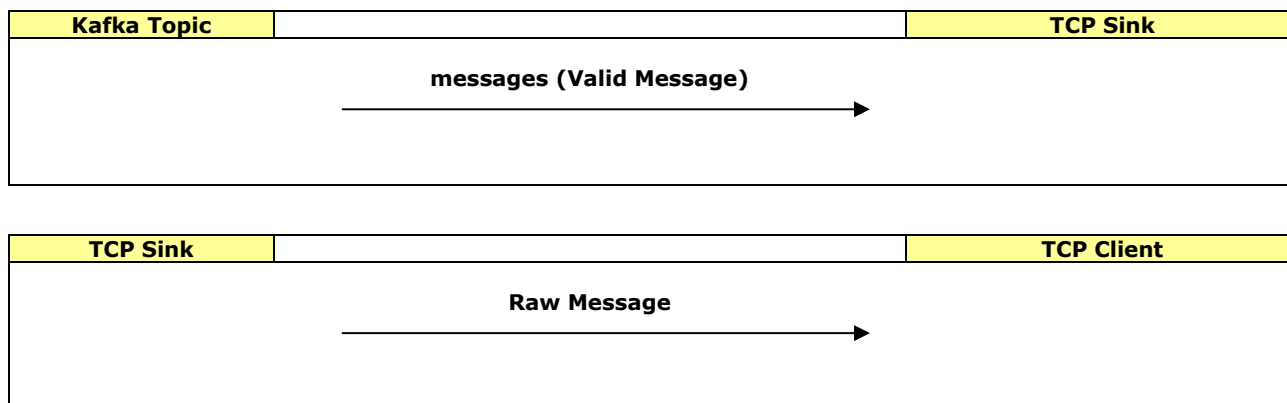
### 8.2 TCP Sink workflow

TCP Sink waits for connection, when client connects consumes from the Kafka Topic messages with the Valid Message Avro Format.

The message consumed will then be processed and the result will be sent to the client that connected via TCP.

#### 8.2.1 Message flow

The following figure outlines the expected synchronous flow of the Avro Valid Message.



#### 8.2.2 Message description (Kafka Topic to TCP Sink)

The following table describes the Avro Valid Message that is consumed from the topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1

message_id	1
hub_id	1

### 8.2.3 Example message (Kafka Topic to TCP Sink)

Container includes Avro Valid Message.
<pre>{   "unique_id": "861919401@1546970330000",   "originator": "RUS",   "source": "source-xpa",   "timestamp": 1546970330000,   "creation_timestamp": 1546970330001,   "processed_timestamp": 1546970330002,   "prev_topic_timestamp": 1546970330001,   "raw_message": "\\i:&lt;O&gt;RUS&lt;/O&gt;*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",   "formatted_message": "",   "mmsi": "356091000",   "message_id": 1,   "hub_id": "hub-1"} </pre>

### 8.2.4 Message description (TCP Sink to TCP Client)

The following table describes the Message that is sent to the TCP Client.

Value	Occ
raw_message	1

### 8.2.5 Example message (TCP Sink to TCP Client)

Container includes a sample Message of the message sent to the TCP Client.
\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63

## 9 FILE SINK

### 9.1 Functionality

The File Sink Connector is designed to consumed messages in the Generic Message Avro Format and write them in files located in a pre-defined directory. The timeframe of the file creation can be defined as:

- Minute
- Hour
- Day

The File Sink will obtain the raw message from the avro schema and save it in a file grouped with the current timestamp.

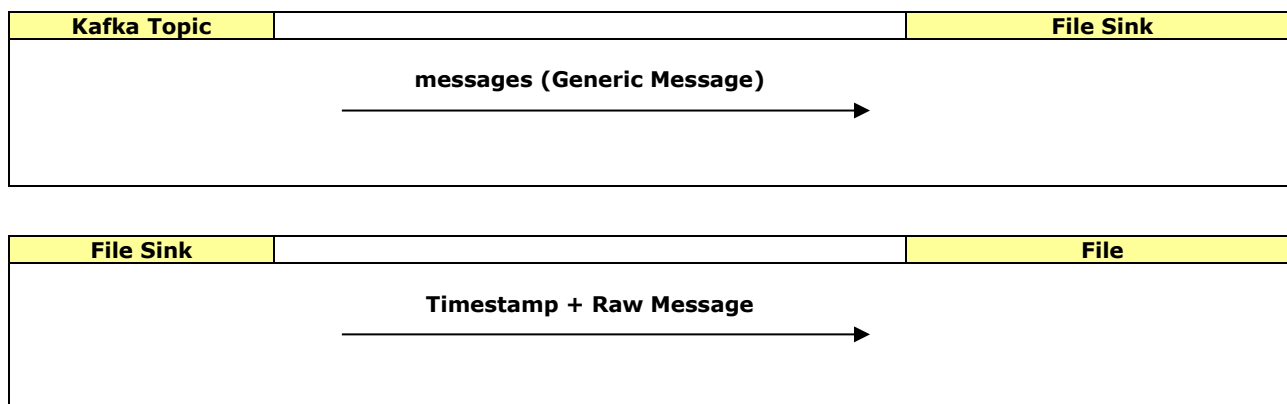
### 9.2 File Sink workflow

Messages are consumed from a topic with the Generic Message Avro Format by the File Sink Connector.

The message consumed will then be processed and later written in a file located in the directory defined when the connector was installed.

#### 9.2.1 Message flow

The following figure outlines the expected synchronous flow of the Avro Valid Message.



#### 9.2.2 Message description (Kafka Topic to File Sink)

The following table describes the Avro Generic Message that is consumed from the topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1

raw_message	1
hub_id	1

### 9.2.3 Example message (Kafka Topic to File Sink)

Container includes Avro Generic Message.

```
{ "unique_id": "861919401@1546970330000",
  "originator": "RUS",
  "source": "source-xpa",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "processed_timestamp": 1546970330002,
  "prev_topic_timestamp": 1546970330001,
  "raw_message": "\\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63",
  "hub_id": "hub-1" }
```

### 9.2.4 Message description (File Sink to File)

The following table describes the Message that is sent to the TCP Client.

Value	Occ
timestamp	1
raw_message	1

**Note:** The timestamp sent is the actual timestamp and not the original from the message.

### 9.2.5 Example message (File Sink to File)

Container includes a sample Message of the message that is saved in a file.

```
1546970330 + \t + \\i:<O>RUS</O>*28\\!AIVDM,1,1,,1,15CV4N001jCCbGLAJ022WR220H0e,0*63
```

## 10 MONITOR METRICS SINK

### 10.1 Functionality

The Monitor Metrics Sink Connector is designed to consumed messages in the Monitor Message Avro Format produce a count of originator at the end of each time window and write the final count to a file located in a pre-defined directory. The timeframe of the originator count write is defined as milliseconds.

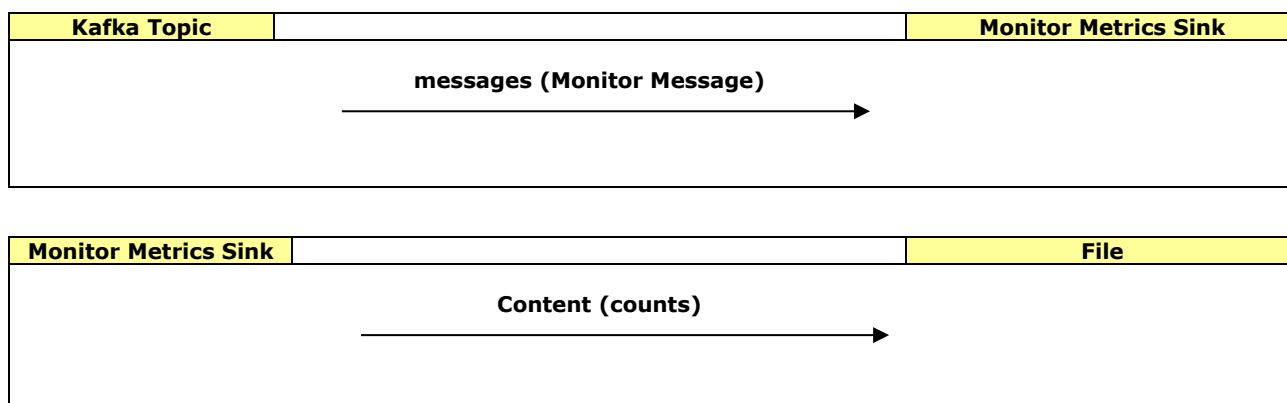
The Monitor Metrics Sink will obtain all the messages that were produced to the monitor from a specified source topic and will count the originator until the time window has ended, when it's end the connector will write the final count to a specified file.

### 10.2 Monitor Metrics Sink workflow

Messages are consumed from a topic with the Monitor Message Avro Format by the Monitor Metrics Sink Connector.

#### 10.2.1 Message flow

The following figure outlines the expected synchronous flow of the Avro Monitor Message.



#### 10.2.2 Message description (Kafka Topic to Monitor Metrics Sink)

The following table describes the Avro Monitor Message obtained from the kafka topic.

Value	Occ
unique_id	1
timestamp	
creation_timestamp	1
received_timestamp	1
prev_topic_timestamp	
source	1
originator	1
input_topic	1
mmsi	1
message_id	1
hub_id	1

### 10.2.3 Example message (Obtained Message)

Container includes Avro Monitor Message producer after consuming a message with the GenericMessage Avro Schema.

```
{ "unique_id": "861919401@1546970330000",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "received_timestamp": 1546970360001,
  "prev_topic_timestamp": 1546970330000,
  "source": "source-xpa",
  "originator": "POR",
  "input_topic": "input-topic",
  "mmsi": "000000000",
  "message_id": 0,
  "hub_id": "hub-1" }
```

Container includes Avro Monitor Message producer after consuming a message with the ValidMessage Avro Schema.

```
{ "unique_id": "861919401@1546970330000",
  "timestamp": 1546970330000,
  "creation_timestamp": 1546970330001,
  "received_timestamp": 1546970360002,
  "prev_topic_timestamp": 1546970330000,
  "source": "source-xpa",
  "originator": "POR",
  "input_topic": "valid-topic",
  "mmsi": "356091000",
  "message_id": 1,
  "hub_id": "hub-1" }
```

### 10.2.4 File Content (Originator count)

The following table describes the file line content that is produced by the connector.

Value	Occ
timestamp	1
source	0-1
originator	0-1
count	0-1

### 10.2.5 Example content (Produced by Monitor Metrics Sink)

Container includes a file content produced by multiple iterations of the connector.

```
1553695638737
1553695668743   POR:11210
1553695698771   POR:11119
1553695728791   POR:11199
1553695773597   POR:7927
1553695833601
1553695893598
1553695933818
1553695963838   POR:11224,FRA:11224
1553695993851   POR:11153,FRA:11153
1553696023869   POR:11131,FRA:11131
1553696053891   POR:6665,FRA:4486,UGA:2179
1553696083917   POR:11125,FRA:5562,UGA:5563
1553696133598   POR:6620,FRA:3310,UGA:3310
1553696193598
1553696253598
```



## 11 AIS BACKLOG FILE SOURCE

### 11.1 Functionality

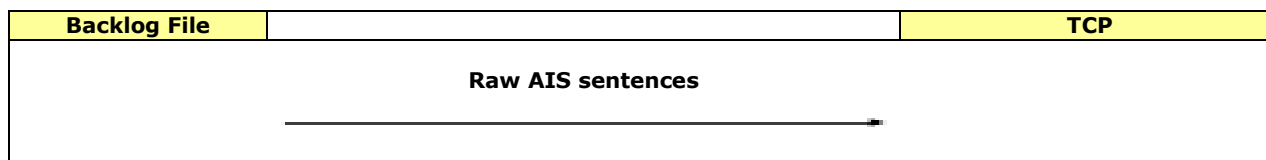
The AIS Backlog File Source Application is designed to constantly scan a specified directory for new AIS backlog files and send their contents to a TCP connection.

### 11.2 AIS Backlog File Source workflow

AIS sentences are read from the backlog file line by line and sent to a TCP server without validation.

#### 11.2.1 Message flow

The following figure outlines the expected flow of Raw AIS messages.



#### 11.2.2 Backlog file description

The backlog file must be a new line separated list of AIS sentences. Sentences will be read from top to bottom and multi-sentence messages must be ordered.

#### 11.2.3 Backlog file example

File containing multiple AIS messages.

```
\i:<O>DNK</O>,c:1560526125*4D\!BSVDM,1,1,,A,13BDwr001nQ3SQf0@Vvk;BSL00RO,0*6E
\i:<O>DNK</O>,c:1560526125*4D\!BSVDM,2,1,2,B,53uNqUT00003M<7;S;50thU>3GVo0?OC2222220D0`522t0Ht0
000000,0*01
\i:<O>DNK</O>,c:1560526125*4D\!BSVDM,2,2,2,B,0000000000000000,2*3C
```

#### 11.2.4 TCP output description

The output consists of a sequence of raw AIS sentences new line separated.

#### 11.2.5 Example TCP output

AIS sentences sent to the TCP server following the backlog file example.

```
1546970330 + \t +
\i:<O>DNK</O>,c:1560526125*4D\!BSVDM,2,1,2,B,53uNqUT00003M<7;S;50thU>3GVo0?OC2222220D0`522t0Ht0
000000,0*01\n\i:<O>DNK</O>,c:1560526125*4D\!BSVDM,2,2,2,B,0000000000000000,2*3C\n
```

## 12 JMS SOURCE CONNECTOR

### 12.1 Functionality

A Kafka Source Connector with the purpose of consuming messages from a JMS Topic with CDF messages and produce corresponding avro messages in a format yet to be validated. The connector obtains each corresponding field from the CDF message and maps it to a field in the avro schema.

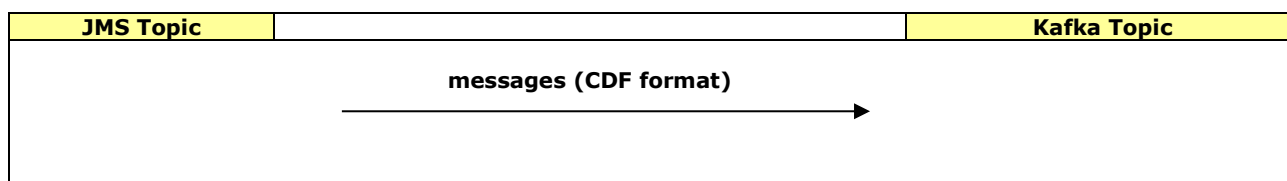
### 12.2 JMS Source workflow

Messages are consumed from a JMS topic and will be placed on a topic with a custom schema similar to the one defined by EMSA as the main schema to be used thought the enrichment process.

This step is required since kafka connect does not use the same data structure libraries to produce the avro message that regular kafka applications use.

#### 12.2.1 Message flow

The following figure outlines the expected synchronous flow of the JMS topic messages.



#### 12.2.2 Example message (Obtained Message)

Container includes an example of a CDF message obtained from the topic.

```
<ns2:EMSA
  xmlns:ns3="http://schemas.emsa.europa.eu/cdf/preprocessing"
  xmlns:ns2="http://schemas.emsa.europa.eu/cdf/position"
  xmlns="http://schemas.emsa.europa.eu/cdf" source="T-AIS" id="295170634794987646"
  timestamp="2019-03-26T12:18:10Z">
  <DataAccessRights>
    <Originator>ISL</Originator>
  </DataAccessRights>
  <ns2:PositionMessage>
    <ns2:ShipParticulars>
      <EMSAId>23342334</EMSAId>
      <MMSI>251011000</MMSI>
    </ns2:ShipParticulars>
    <ns2:PositionReport positionID="295170634794987646" timestamp="2019-03-26T12:18:10Z"
source="T-AIS">
      <Latitude>63.447964</Latitude>
      <Longitude>-20.031631</Longitude>
      <ns2:SOG>0.8</ns2:SOG>
      <ns2:COG>39.0</ns2:COG>
      <ns2:TrueHeading>329.0</ns2:TrueHeading>
      <ns2:RateOfTurn>720.0</ns2:RateOfTurn>
      <ns2:AdditionalInfo source="T-AIS">
        <ns4:AisSpecific
          xmlns:ns1="http://schemas.emsa.europa.eu/cdf"
          xmlns:ns0="http://schemas.emsa.europa.eu/cdf/position"
          xmlns:ns4="http://schemas.emsa.europa.eu/cdf/ais" msgId="1">
            <ns4:RawMessage>\i:<O>ISL</O>,c:1553602690*56
\!AIVDM,1,1,,A,13gHof?Oh8NTCFrTCR:QQbBH00S?,0*14</ns4:RawMessage>
          </ns4:AisSpecific>
```

```

        <ns3:PreProcessingSpecific>
          <ns3:KalmanValidationOutput>

<ns3:KalmanOutput>{"errorVU":0,"errorLatLon":0,"errorLonLon":100,"errorLonV":0,"errorLonU":0,
"errorLatU":0,"errorLatLat":100,"errorVV":6.616327160493828,"errorLatV":0,"errorUU":6.616327160493828,
"version":1}</ns3:KalmanOutput>
          <ns3:ValidityFlag>U</ns3:ValidityFlag>
        </ns3:KalmanValidationOutput>
      </ns3:PreProcessingSpecific>
    </ns2:AdditionalInfo>
  </ns2:PositionReport>
  <ns2:PositionEnrichmentInfo>
    <ns2:ShipInfo source="T-AIS" timestamp="2019-03-26T12:22:43Z">
      <ShipParticulars CSDID="163606">
        <EMSAId>1585292</EMSAId>
        <IMO>8303410</IMO>
        <MMSI>251011000</MMSI>
        <Name>BRYNJOLFUR</Name>
        <CallSign>TFTV</CallSign>
      </ShipParticulars>
      <ShipType>315</ShipType>
      <ShipDimensions>
        <A>12.0</A>
        <B>30.0</B>
        <C>4.0</C>
        <D>4.0</D>
      </ShipDimensions>
    </ns2:ShipInfo>
  </ns2:PositionEnrichmentInfo>
</ns2:PositionMessage>
</ns2:EMSA>

```

## 12.2.3 Avro Schema

The following section describes the avro schema of the topic where the messages are placed after being consumed and processed by the connector.

The following table describes the schema of the Position Report Avro Message.

Value		Occ
source		1
timestamp		1
originator		1
requestor		0-1
longitude		1
latitude		1
speed_over_ground		0-1
courser_over_ground		0-1
navigational_status		0-1
true_heading		0-1
heading		0-1
rate_of_turn		0-1
particulars		0-1
	csd_id	0-1
	emsa_id	0-1
	imo	0-1
	mmsi	0-1
	ir	0-1

	name		0-1	
	call_sign		0-1	
	flag_state		0-1	
ship_enrichment			0-1	
	source		1	
	timestamp		0-1	
	particulars		1	
		csd_id		0-1
		emsa_id		0-1
		imo		0-1
		mmsi		0-1
		ir		0-1
		name		0-1
		call_sign		0-1
		flag_state		0-1
	ship_type		0-1	
	banned		0-1	
	single_hull_tanker		0-1	
	position_fixing_device		0-1	
	detained		0-1	
	ship_risk_profile		0-1	
	priority_of_inspection		0-1	
	eligible_for_esp_inspection		0-1	
	eligible_for_banning		0-1	
	dimensions		1	
		a		0-1
		b		0-1
		c		0-1
		d		0-1
		beam		0-1
		length_overal		0-1
source_specific			1	
ais_specific			1	
	message_type		1	
	position_accuracy		1	
	raim_flag		1	
	nmea		1	
sat_ais_specific			1	
	message_type		1	
	position_accuracy		1	
	raim_flag		1	
	nmea		1	
	satellite_id		0-1	

	ground_station_identifier_ts	0-1
	ground_station_acquisition_ts	0-1
	data_centre_ingestion_ts	0-1
	data_centre_delivery_ts	0-1
	frequency_of_arrival	0-1
	time_of_arrival	0-1
	doppler_signal_level	0-1
	doppler_signal_noise_ratio	0-1
	doppler_channel_id	0-1
	data_flow_id	0-1
lrit_specific		1
	message_type	1
	response_type	1
	reference_id	1
	message_id	1
vms_specific		1
	naf	1
	fishing_serial_trip_number	0-1

## 12.2.4 Example Avro Message

An example of a message that was processed by the connector and placed in the kafka topic.

```
{
  "ais_specific": {
    "message_type": "CLASS_A",
    "nmea": "
\\i:<O>ISL</O>,c:1553602690*56\\!AIVDM,1,1,,A,13gHOf?Oh8NTCFrTCR:QQbBH00S?,0*14",
    "position_accuracy": "LOW",
    "raim_flag": false
  },
  "course_over_ground": {
    "float": 39.0
  },
  "heading": null,
  "latitude": 63.447964,
  "longitude": -20.031631,
  "lrit_specific": {
    "message_id": "",
    "message_type": 0,
    "reference_id": "",
    "response_type": 0
  },
  "navigational_status": null,
  "originator": "ISL",
  "particulars": {
    "call_sign": null,
    "csd_id": null,
    "emsa_id": {
      "string": "23342334"
    },
    "flag_state": null,
    "imo": null,
    "ir": null,
    "mmsi": {
      "long": 251011000
    },
    "name": null
  }
}
```

```

    },
    "position_id": 295170634794987646,
    "rate_of_turn": {
        "double": 720.0
    },
    "requestor": null,
    "sat_ais_specific": {
        "data_centre_delivery_ts": null,
        "data_centre_ingestion_ts": null,
        "data_flow_id": null,
        "doppler_channel_id": null,
        "doppler_signal_level": null,
        "doppler_signal_noise_ratio": null,
        "frequency_of_arrival": null,
        "ground_station_acquisition_ts": null,
        "ground_station_identifier": null,
        "message_type": "",
        "nmea": "",
        "position_accuracy": "LOW",
        "raim_flag": false,
        "satellite_id": null,
        "time_of_arrival": null
    },
    "ship_enrichment": {
        "banned": null,
        "detained": null,
        "dimensions": {
            "a": {
                "float": 12.0
            },
            "b": {
                "float": 30.0
            },
            "beam": null,
            "c": {
                "float": 4.0
            },
            "d": {
                "float": 4.0
            },
            "length_overall": null
        },
        "eligible_for_banning": null,
        "eligible_for_esp_inspection": null,
        "particulars": {
            "call_sign": {
                "string": "TFTV"
            },
            "csd_id": {
                "long": 163606
            },
            "emsa_id": {
                "string": "1585292"
            },
            "flag_state": null,
            "imo": {
                "string": "8303410"
            },
            "ir": null,
            "mmsi": {
                "long": 251011000
            },
            "name": {
                "string": "BRYNJOLFUR"
            }
        },
        "position_fixing_device": null,
        "priority_of_inspection": null,
        "ship_risk_profile": null,
        "ship_type": {
            "string": "315"
        },
        "single_hull_tanker": null,

```

```
    "source": "T-AIS",
    "timestamp": {
      "long": 1553602963000
    }
  },
  "source": "T-AIS",
  "source_specific": "T-AIS",
  "speed_over_ground": {
    "float": 0.8
  },
  "timestamp": 1553602690000,
  "true_heading": {
    "float": 329.0
  },
  "validity_flag": "U",
  "vms_specific": {
    "fishing_serial_trip_number": null,
    "naf": ""
  }
}
```

## 13 MULTISOURCE STREAM PROCESSOR

### 13.1 VMS Conversion Functionality

This Kafka Streams application is designed to consume from a Kafka topic with the SimpleMessage avro schema associated and should obtain the "raw\_message" field and convert it to CDF format, placing the conversion result in the field "formatted\_message". This conversion will only happen with VMS sources and this setting can be configured on the application configuration file.

### 13.2 VMS Conversion Workflow



#### 13.2.1 Message description (SimpleMessage Avro)

Below it is described the avro format of the messages that will be received from the desired Kafka Topics.

Value	Occ
unique_id	1
timestamp	1
raw_message	1
formatted_message	0-1

#### 13.2.2 Example Message (Received from Kafka)

A kafka message received to the topic after this message was consumed from a JMS queue/topic with the JMS Simple Source Connector.

```
{
  "unique_id": "861919401@1546970330",
  "timestamp": 1546970330,
  "raw_message": "//SR//AD/IMDATA//FR/XFA//TM/POS//RC/GGCR//IR/GBR000A13184//XR/PZ51//LT/50.455//LG/-8.433//DA/20190611//TI/1926//SP/89//CO/264//NA/GOVENEK OF LADRAM//FS/GBR//ER//",
  "formatted_message": ""
}
```

#### 13.2.3 Message description (SimpleMessage Avro)

Below it is described the avro format of the messages that will be sent to the desired Kafka Topic.

Value	Occ
unique_id	1
timestamp	1
raw_message	1
formatted_message	1

### 13.2.4 Example Message (Sent to Kafka)

A kafka message sent to the topic after the CDF conversion.

```
{ "unique_id": "861919401@1546970330",
  "timestamp": 1546970330,
  "raw_message": "//SR//AD/IMDATA//FR/XFA//TM/POS//RC/GGCR//IR/GBR000A13184//XR/PZ51//LT/50.455//LG/-8.433//DA/20190611//TI/1926//SP/89//CO/264//NA/GOVENEK OF LADRAM//FS/GBR//ER//",
  "formatted_message": "<?xml version='1.0' encoding='UTF-8' standalone='yes'?><ns2:EMSA source='VMS' id='idTest' timestamp='2019-08-02T14:52:06.288+01:00' xmlns='http://schemas.emsa.europa.eu/cdf' xmlns:ns5='http://schemas.emsa.europa.eu/cdf/vms' xmlns:ns2='http://schemas.emsa.europa.eu/cdf/position' xmlns:ns4='http://schemas.emsa.europa.eu/cdf/ais' xmlns:ns3='http://schemas.emsa.europa.eu/cdf/voyage'><ns2:PositionMessage><ns2:DataAccessRights><Originator>GBR</Originator></ns2:DataAccessRights><ns2:ShipParticulars><IR>GBR000A13184</IR><Name>GOVENEK OF LADRAM</Name><CallSign>GGCR</CallSign><OtherId name='XR'>PZ51</OtherId><FlagState>GB</FlagState></ns2:ShipParticulars><ns2:PositionReport timestamp='2019-07-11T19:26:00.000Z'><Latitude>50.455</Latitude><Longitude>-8.432</Longitude><ns2:SOG unit='knots'>89</ns2:SOG><ns2:COG unit='decimal degrees'>264</ns2:COG><ns2:AdditionalInfo source='VMS'><ns5:VmsSpecific><ns5:RawMessage>//SR//AD/IMDATA//FR/XFA//TM/POS//RC/GGCR//IR/GBR000A13184//XR/PZ51//LT/50.455//LG/-8.433//DA/20190611//TI/1926//SP/89//CO/264//NA/GOVENEK OF LADRAM//FS/GBR//ER//</ns5:RawMessage></ns5:VmsSpecific></ns2:AdditionalInfo></ns2:PositionReport></ns2:PositionMessage></ns2:EMSA>" }
```

## 13.3 Other Sources Routing Functionality

This Kafka Streams application is designed to consume from Kafka topic with the SimpleMessage avro schema associated and will route the message to the specified output topic.

## 13.4 Other Sources Routing Workflow



### 13.4.1 Message description (SimpleMessage Avro)

Below it is described the avro format of the messages that will be received from the desired Kafka Topics.

Value	Occ
unique_id	1
timestamp	1
raw_message	1
formatted_message	0-1

### 13.4.2 Example Message (Received from Kafka)

A kafka message received to the topic after this message was consumed from a JMS queue/topic with the JMS Simple Source Connector.

```
{ "unique_id": "861919401@1546970330",
  "timestamp": 1546970330,
  "raw_message": "//SR//AD/IMDATA//FR/XFA//TM/POS//RC/GGCR//IR/GBR000A13184//XR/PZ51//LT/50.455//LG/-8.433//DA/20190611//TI/1926//SP/89//CO/264//NA/GOVENEK OF LADRAM//FS/GBR//ER//",
  "formatted_message": ""
}
```

### 13.4.3 Message description (SimpleMessage Avro)

Below it is described the avro format of the messages that will be sent to the desired Kafka Topic.

Value	Occ
unique_id	1
timestamp	1
raw_message	1
formatted_message	1

### 13.4.4 Example Message (Sent to Kafka)

A kafka message sent to the topic after the routing.

```
{ "unique_id": "861919401@1546970330",
  "timestamp": 1546970330,
  "raw_message": "//SR//AD/IMDATA//FR/XFA//TM/POS//RC/GGCR//IR/GBR000A13184//XR/PZ51//LT/50.455//LG/-8.433//DA/20190611//TI/1926//SP/89//CO/264//NA/GOVENEK OF LADRAM//FS/GBR//ER//",
  "formatted_message": ""
}
```

## 14 AIS ORACLE STREAM PROCESSING

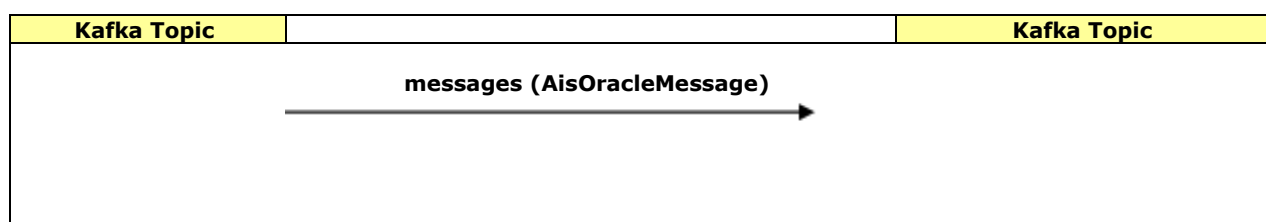
### 14.1 Functionality

A Kafka stream application designed to consume from two Kafka topics, one containing the Valid messages and the other the Invalid/Discarded ones.

This stream application will consume Valid Messages and given its source will produce to a different topic. The same happens to the Discarded messages that will be sent to the Kafka topic that contains this type of messages.

A JDBC Sink Connector will be in place to consume from these topics and send the messages to the Oracle database specified for the given topic.

### 14.2 Oracle Stream Processing Workflow



#### 14.2.1 Message description (ValidMessage Avro)

Below it is described the avro format of the messages that will be received from the valid Kafka Topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

#### 14.2.2 Example Message (Received from Kafka)

A kafka message received from the topic of valid messages
<pre>{   "unique_id": "363017035@1546970330000",   "originator": "originator-xpa",   "source": "SAT-AIS",   "timestamp": 1546970330000,   "creation_timestamp": 1546970330001,   "processed_timestamp": 1546970330002, }</pre>

```
"prev_topic_timestamp":1546970330001,
"raw_message":"\\i:<S>SAT-AIS</S><T> A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5
F:+975000 T:+123000</T>*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37",
"formatted_message":"",
"mmsi":"9106342",
"message_id":18,
"hub_id":"hub-1"}
```

### 14.2.3 Message Description (DiscardedMessage Avro)

Below it is described the avro format of the messages that will be received from the discarded Kafka Topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
discarded_timestamp	1
raw_message	1
reason	1
hub_id	1

### 14.2.4 Message Description (AisOracleMessageTAIS Avro)

Below it is described the avro format of the messages that will be sent to the TAIS Kafka topic.

Value	Occ
message_timestamp	1
timestamp_delivery_rh	1
timestamp_insert_db	1
remote_hub_id	1
originator	1
mmsi	1
coordinates	1
raw_message	0-1

### 14.2.5 Example AISOracleMessageTAIS message (Sent from Kafka)

A kafka message sent to topic after conversion

```
{"timestamp_insert_db":1564157474947,"timestamp":1564155463000,"unique_id":1827312404,"tag":"origi
nator-xpa","mmsi":"9106342","raw_message":"\\i:<S>SAT-AIS</S><T> A:123456999949 I:123456789
D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000
T:+123000</T>*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37","message_id":18,"sat_id":{"stri
ng":"SAT-AIS"},"ground_id":{"string":"Svalbard-5"},"timestamp_sat":{"long":975000}}
```

### 14.2.6 Message Description (AISOracleMessageShipAISAvro)

Below it is described the avro format of the messages that will be sent to the ShipAIS Kafka topic.

Value	Occ
message_timestamp	1
timestamp_delivery_rh	1
timestamp_insert_db	1
remote_hub_id	1
originator	1
mmsi	1
coordinates	1
raw_message	0-1

### 14.2.7 Example AISOracleMessageShipAIS message (Sent from Kafka)

A kafka message sent to topic after conversion
<pre>{   "timestamp_insert_db":1564157474947,   "timestamp":1564155463000,   "unique_id":1827312404,   "tag":"originator-xpa",   "mmsi":"9106342",   "raw_message":"\\i:&lt;S&gt;SAT-AIS&lt;/S&gt;&lt;T&gt; A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000 T:+123000&lt;/T&gt;*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37",   "message_id":18,   "sat_id":{"string":"SAT-AIS"},   "ground_id":{"string":"Svalbard-5"},   "timestamp_sat":{"long":975000} }</pre>

### 14.2.8 Message Description (AISOracleMessageSatAIS Avro)

Below it is described the avro format of the messages that will be sent to the SatAIS Kafka topic.

Value	Occ
message_timestamp	1
timestamp_delivery_rh	1
timestamp_insert_db	1
remote_hub_id	1
originator	1
satellite_id	1
ground_station_id	1
foa	1
toa	1
doppler_freq_channel	1
mmsi	1
coordinates	0-1
raw_message	1

### 14.2.9 Example AISOracleMessageSatAIS message (Sent from Kafka)

A kafka message sent to topic after conversion
<pre>{   "timestamp_insert_db":1564157474947,   "timestamp":1564155463000,   "unique_id":1827312404,   "tag":"originator-xpa",   "mmsi":"9106342",   "raw_message":"\\i:&lt;S&gt;SAT-AIS&lt;/S&gt;&lt;T&gt; A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000 T:+123000&lt;/T&gt;*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37",   "message_id":18,   "sat_id":{"string":"SAT-AIS"},   "ground_id":{"string":"Svalbard-5"},   "timestamp_sat":{"long":975000} }</pre>

### 14.2.10 Message Description (AISOracleMessageRPAS AIS Avro)

Below it is described the avro format of the messages that will be sent to the RPAS AIS Kafka topic.

Value	Occ
message_timestamp	1
timestamp_delivery_rh	1
timestamp_insert_db	1
remote_hub_id	1
originator	1
rpas_id	1
mmsi	1
coordinates	0-1
raw_message	1

### 14.2.11 Example AISOracleMessageRPAS AIS message (Sent from Kafka)

A kafka message sent to topic after conversion
<pre>{   "timestamp_insert_db":1564157474947,   "timestamp":1564155463000,   "unique_id":1827312404,   "tag":"originator-xpa",   "mmsi":"9106342",   "raw_message":"\\i:&lt;S&gt;SAT-AIS&lt;/S&gt;&lt;T&gt; A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000 T:+123000&lt;/T&gt;*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37",   "message_id":18,   "sat_id":{"string":"SAT-AIS"},   "ground_id":{"string":"Svalbard-5"},   "timestamp_sat":{"long":975000} }</pre>

### 14.2.12 Message description (AISOracleMessageInvalid Avro)

Below it is described the avro format of the messages that will be sent to the Invalid Kafka topic.

Value	Occ
message_timestamp	1
timestamp_delivery_rh	1
timestamp_insert_db	1
remote_hub_id	1
originator	1
mmsi	1
raw_message	1

reason_rejection	1
------------------	---

### 14.2.13 Example AISOracleMessageInvalid message (Sent from Kafka)

A kafka message sent to topic after conversion

```
{ "timestamp_insert_db":1564157474947,"timestamp":1564155463000,"unique_id":1827312404,"tag":"originator-xpa", "mmsi":"9106342", "raw_message":"\\i:<S>SAT-AIS</S><T> A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000 T:+123000</T>*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37", "message_id":18, "sat_id":{"string":"SAT-AIS"}, "ground_id":{"string":"Svalbard-5"}, "timestamp_sat":{"long":975000}}
```

## 15 SSN ENRICHMENT PROCESSOR WORKFLOW

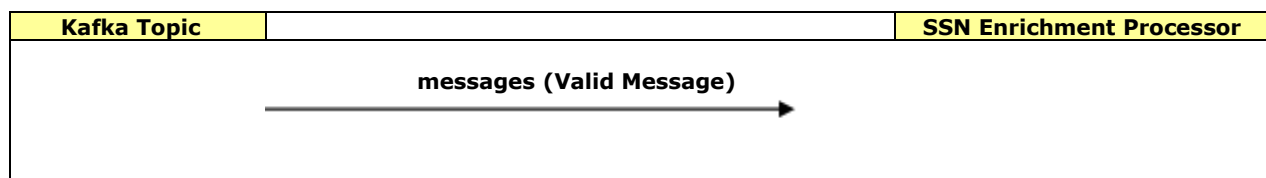
### 15.1 Functionality

The SSN Enrichment Processor is designed to consume messages with a Valid Message format, extract the MMSI of each record and then enrich the Comment Block of that message with information, relative to that MMSI, from multiple XML files containing relevant information. These XML files will be constantly updating their information, so this reading process of the XML files will occur frequently (e.g: 5 min). After the enrichment of the comment block is completed, the new valid message will be written to a Kafka topic.

### 15.2 SSN Enrichment Processor Workflow

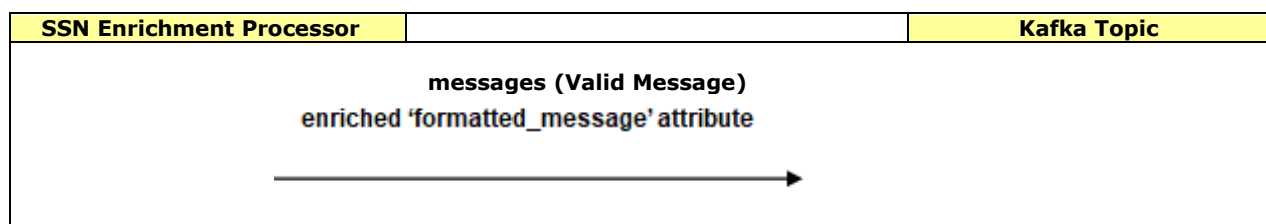
#### 15.2.1 Message Flow

Messages are consumed from a topic with the Valid Message format by the SSN Enrichment Processor.



Given the message, the SSN Enrichment Processor will enrich the message (in this case given the 'raw\_message' field, the enrichment processor will populate the 'formatted\_message' attribute) with SSN relevant information.

After the processing and enrichment of the message is done, the processor will write the enriched message into an output Kafka topic.



#### 15.2.2 Message description (Kafka Topic to SSN Enrichment Processor)

The following table describes the Valid Message obtained from the kafka topic.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp_	1

processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

### 15.2.3 Example message (Obtained message)

The following table describes the raw message:

Container includes the raw_message attribute obtained from a message that was read from the Kafka Topic
<pre>\g:1-2-1954,s:43931,c:1572447200*7F\\i:&lt;T&gt;A:1572447200 G:1 I:1572462335 D:1572465191 F:0.001000009298324585&lt;/T&gt;*2D\\i:&lt;Q&gt;12&lt;/Q&gt;*7F\\AIVDM,2,1,2,,59NSS;T2E2UHCd1KN20&lt;50F0d E=A8Dj222222216OhVN?6d@0JDhCS10,0*2B \g:2-2-1954*54\\!AIVDM,2,2,2,,CR0H0PFH8888880,2*60</pre>

### 15.2.4 Example of an enriched message (Produced by SSN Enrichment Processor)

The table below shows the result of the enrichment (the new 'formatted\_message' attribute):

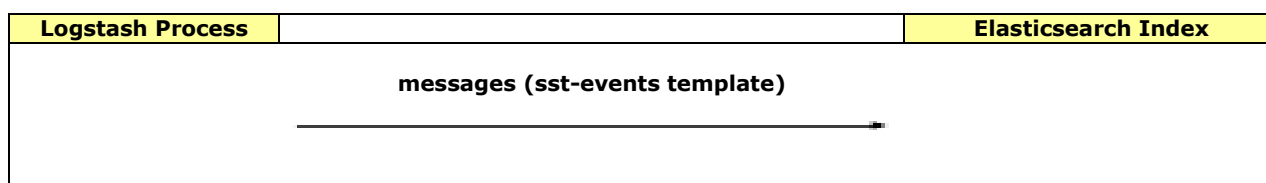
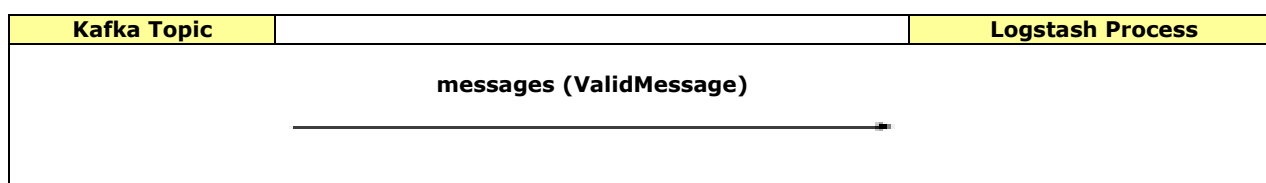
Container includes the formatted_message attribute obtained from a message enriched by the SSN Enrichment Processor
<pre>\1G4:1954,s:43931,c:1572447200,i:&lt;T&gt;A:1572447200 G:1 I:1572462335 D:15724651*30\\!AIVDM,2,1,2,,59NSS;T2E2UHCd1KN20&lt;50F0dE=A8Dj222222216OhVN?6d@0JDhCS10, 0*2B \2G4:1954,i:91 F:0.001000009298324585&lt;/T&gt;&lt;Q&gt;12&lt;/Q&gt;&lt;E&gt;C K Z ES&lt;/E&gt;&lt;I&gt;I:931629*3F\\!AIVDM,2,2,2,,CR0H0PFH8888880,2*60 \3G4:1954,i:4 M:247160401 C:C6UR4 F:BS N:"THAMES HIGHWAY"&lt;/I&gt;&lt;L&gt;SEMMA&lt;/L&gt;&lt;P&gt;*2C\ \4G4:1954,i:EEPLS&lt;/P&gt;*61\</pre>

## 16 REAL-TIME DASHBOARD INTERFACES

### 16.1 Functionality

Provides real-time analytic data and visualizations of events in Kafka topics. A Logstash process is used to consume and process records from Kafka topics into Elasticsearch indexes from which analytical data can be queried. Queries are sent to an Elasticsearch REST endpoint that Kibana uses to provide real-time visualizations organized in dashboards.

### 16.2 Kafka to Elasticsearch workflow



#### 16.2.1 Message Description (ValidMessage Avro)

Below is described the avro format of the messages that will be received from the desired Kafka Topics.

Value	Occ
unique_id	1
originator	1
source	1
timestamp	1
creation_timestamp	1
processed_timestamp	1
prev_topic_timestamp	1
raw_message	1
formatted_message	1
mmsi	1
message_id	1
hub_id	1

#### 16.2.2 Example message (ValidMessage Avro)

A kafka message received from the valid messages topic

```
{
  "unique_id": "363017035@1564155463000",
  "originator": "originator-xpa",
  "source": "SAT-AIS",
  "timestamp": 1564155463000,
  "creation_timestamp": 1564155463001,
  "processed_timestamp": 1564155463002,
  "prev_topic_timestamp": 1564155463001,
  "raw_message": "\\i:<S>SAT-AIS</S><T> A:123456999949 I:123456789 D:123456789 L:SAT-AIS G:Svalbard-5 F:+975000 T:+123000</T>*38\\!AIVDM,1,1,,B,B08ckaP02h;S8bNV72;HRe143P06,0*37",
  "formatted_message": "",
  "mmsi": "9106342",
  "message_id": 18,
  "hub_id": "hub-1"
}
```

### 16.2.3 Message Description (sst-events template)

Below is described the avro format of the messages that will be received from the desired Kafka Topics.

Value			Occ
@timestamp			1
@es_source			1
@es_source_reference			1
timestamps	message	unix_ms	1
		date	1
	delivered	unix_ms	1
		date	1
	current_topic	unix_ms	1
		date	1
	previous_topic	unix_ms	1
		date	1
latency	message_to_delivered_ms		1
	delivered_to_current_ms		1
	previous_to_current_ms		1
geo	longitude		0-1
	latitude		0-1
	position		0-1
source			1
originator			1
mmsi			1
message_id			1
hub_id			1
unique_id			1
raw_message			1
formatted_message			1
sat-ais	satellite_id		0-1

### 16.2.4 Example message (sst-events template)

Content from an sst-events index document

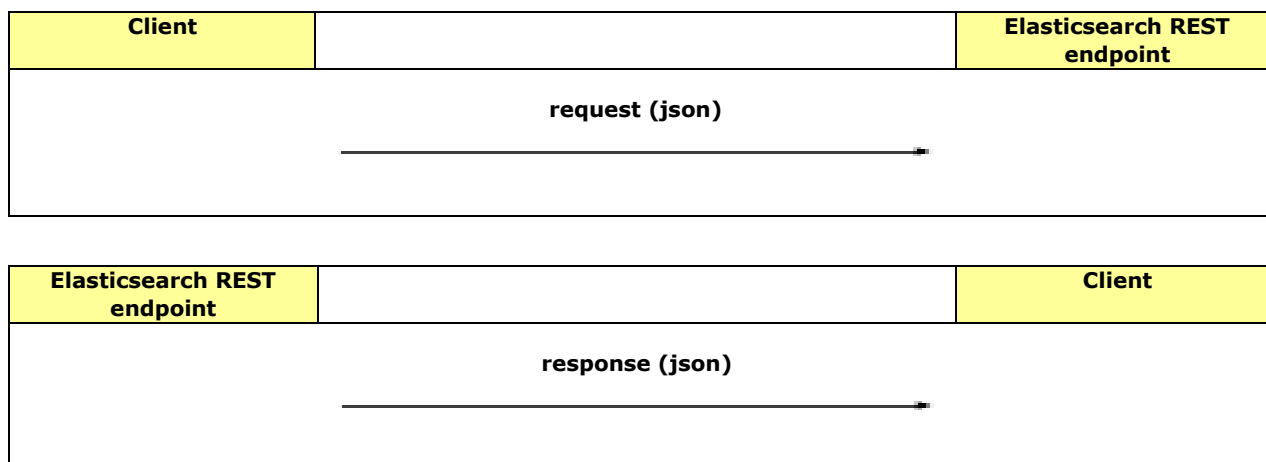
```
{
  "geo": {
```

```

        "longitude": 12.624725000000002,
        "position": [
            12.624725000000002,
            66.02132
        ],
        "latitude": 66.02132
    },
    "originator": "DNK",
    "@es_source_reference": "valid-type-position-topic",
    "mmsi": "257450500",
    "message_id": 1,
    "@es_source": "kafka",
    "raw_message":
    "\\i:<S>X</S><O>DNK</O>,c:1576265778,s:Mallard*0D\\!BSVDM,1,1,,B,13mQR10P000qjaVUiiN8ewvJ0@8@,0*18",
    "formatted_message": "<?xml version='1.0' encoding='UTF-8'
standalone='yes'?>\\n<ns2:EMSA source='Sat-AIS' id='695124677@1576265778000'\\n
timestamp='2019-12-13T19:36:18.673Z'\\n  xmlns='http://schemas.emsa.europa.eu/cdf'\\n
xmlns:ns2='http://schemas.emsa.europa.eu/cdf/position'\\n
xmlns:ns4='http://schemas.emsa.europa.eu/cdf/ais'\\n
xmlns:ns3='http://schemas.emsa.europa.eu/cdf/voyage'\\n    <ns2:PositionMessage>\\n
<ns2:DataAccessRights>\\n        <Originator>DNK</Originator>\\n
</ns2:DataAccessRights>\\n        <ns2:ShipParticulars>\\n            <MMSI>257450500</MMSI>\\n
</ns2:ShipParticulars>\\n            <ns2:PositionReport timestamp='2019-12-13T19:36:18.000Z'>\\n
<Latitude>66.02132000000000289219315163791179656982421875</Latitude>\\n
<Longitude>12.624725000000001529087967355735599994659423828125</Longitude>\\n            <ns2:SOG
unit='knots'>0</ns2:SOG>\\n            <ns2:COG unit='decimal
degrees'>223.09999999999994315658113919198513031005859375</ns2:COG>\\n
<ns2:NavigationalStatus>0</ns2:NavigationalStatus>\\n            <ns2:AdditionalInfo
source='Sat-AIS'>\\n                <ns4:SatAisSpecific dopplerChannelID='B' msgId='1'>/>\\n
</ns2:AdditionalInfo>\\n            </ns2:PositionReport>\\n
</ns2:PositionMessage>\\n</ns2:EMSA>\\n",
    "hub_id": "hub-1",
    "latency": {
        "previous_to_current_ms": 35,
        "delivered_to_current_ms": 37,
        "message_to_delivered_ms": 642
    },
    "unique_id": "695124677@1576265778000",
    "timestamps": {
        "current_topic": {
            "date": "2019-12-13T19:36:18.679Z",
            "unix_ms": 1576265778679
        },
        "delivered": {
            "date": "2019-12-13T19:36:18.642Z",
            "unix_ms": 1576265778642
        },
        "previous_topic": {
            "date": "2019-12-13T19:36:18.644Z",
            "unix_ms": 1576265778644
        },
        "message": {
            "date": "2019-12-13T19:36:18.000Z",
            "unix_ms": 1576265778000
        }
    },
    "@timestamp": "2019-12-13T19:36:18.686Z",
    "source": "X"
}

```

## 16.3 Visualization data flow



### 16.3.1 Elasticsearch REST endpoint

The Elasticsearch [Search API](#) is used to selectively pull data from multiple indexes. Both GET and POST requests can be used at the `/_search` endpoint to the same effect. Parameters can be passed directly in the endpoint URI or in the [request body](#). The latter option is easier to work with for complex queries and will be used in all examples.

Request structure example:

```
curl -X POST "http://elasticsearch.url:9200/_search" -H 'Content-Type: application/json' -d '{ request body }'
```

### 16.3.2 Request Body Description (json)

The request body includes the aggregations to perform on data, variable ranges and filters.

General information about the Search API using a request body can be found here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-body.html>

Information about Bucket Aggregations can be found here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket.html>

And information about the Query DSL that is used to select variable ranges and apply filters can be found here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html>

### 16.3.3 Request body Examples (json)

#### 16.3.3.1 AIS Delivery Volume Metrics

Reported Position Density Map
{ "size": 0,

```
"aggs": {
  "grid": {
    "geotile_grid": {
      "field": "geo.position",
      "precision": 8
    },
    "aggs": {
      "1": {
        "geo_centroid": {
          "field": "geo.position"
        }
      }
    }
  },
  "_source": {
    "excludes": []
  },
  "stored_fields": [
    "*"
  ],
  "script_fields": {},
  "docvalue_fields": [
    {
      "field": "@timestamp",
      "format": "date_time"
    },
    {
      "field": "timestamps.current_topic.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.delivered.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.message.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.previous_topic.date",
      "format": "date_time"
    }
  ],
  "query": {
```

```

"bool": {
  "must": [
    {
      "geo_bounding_box": {
        "geo.position": {
          "top_left": [
            -25.53952,
            64.00553
          ],
          "bottom_right": [
            41.7457200000000006,
            45.44813
          ]
        }
      }
    },
    {
      "range": {
        "@timestamp": {
          "format": "strict_date_optional_time",
          "gte": "2020-02-06T10:46:30.000Z",
          "lte": "2020-02-06T10:51:30.000Z"
        }
      }
    }
  ],
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}

```

#### Volume Overview

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "@es_source_reference",
        "order": {
          "_count": "desc"
        },

```

```

    "missing": "__missing__",
    "size": 5
  },
  "aggs": {
    "2": {
      "date_histogram": {
        "field": "timestamps.delivered.date",
        "calendar_interval": "1m",
        "time_zone": "Europe/Lisbon",
        "min_doc_count": 1
      }
    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],

```

```
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      },
      {
        "bool": {
          "should": [
            {
              "match_phrase": {
                "@es_source_reference": "valid-messages"
              }
            },
            {
              "match_phrase": {
                "@es_source_reference": "valid-type-position-topic"
              }
            },
            {
              "match_phrase": {
                "@es_source_reference": "valid-type-voyage-topic"
              }
            }
          ],
          "minimum_should_match": 1
        }
      },
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ],
    "filter": [
      {
        "match_all": {}
      }
    ]
  }
}
```

<pre>         }       ],       "should": [],       "must_not": []     }   } } </pre>
Message Volume by Hub Over Time
<pre> {   "aggs": {     "3": {       "terms": {         "field": "hub_id",         "order": {           "_count": "desc"         },         "missing": "__missing__",         "size": 5       },       "aggs": {         "2": {           "date_histogram": {             "field": "timestamps.delivered.date",             "calendar_interval": "1m",             "time_zone": "Europe/Lisbon",             "min_doc_count": 1           }         }       }     }   },   "size": 0,   "_source": {     "excludes": []   },   "stored_fields": [     "*"   ],   "script_fields": {},   "docvalue_fields": [     {       "field": "@timestamp",       "format": "date_time"     },     { </pre>

```

    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  },
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
}

```

Sat-AIS Message Volume by Satellite over Time

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "sat-ais.satellite_id",

```

```

    "order": {
      "_key": "desc"
    },
    "missing": "__missing__",
    "size": 5
  },
  "aggs": {
    "2": {
      "date_histogram": {
        "field": "timestamps.delivered.date",
        "calendar_interval": "1m",
        "time_zone": "Europe/Lisbon",
        "min_doc_count": 0
      }
    }
  },
  "size": 0,
  "_source": {
    "excludes": []
  },
  "stored_fields": [
    "*"
  ],
  "script_fields": {},
  "docvalue_fields": [
    {
      "field": "@timestamp",
      "format": "date_time"
    },
    {
      "field": "timestamps.current_topic.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.delivered.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.message.date",
      "format": "date_time"
    },
    {
      "field": "timestamps.previous_topic.date",

```

```

    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      },
      {
        "match_phrase": {
          "source": {
            "query": "Sat-AIS"
          }
        }
      }
    ],
    {
      "range": {
        "@timestamp": {
          "format": "strict_date_optional_time",
          "gte": "2020-02-06T10:46:30.000Z",
          "lte": "2020-02-06T10:51:30.000Z"
        }
      }
    }
  ],
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
}

```

Message Type distribution by Hub

```

{
  "aggs": {
    "2": {
      "terms": {

```

```

    "field": "hub_id",
    "order": {
      "_count": "desc"
    },
    "missing": "__missing__",
    "size": 20
  },
  "aggs": {
    "3": {
      "terms": {
        "field": "message_id",
        "order": {
          "_count": "desc"
        },
        "missing": "__missing__",
        "size": 20
      }
    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  }
]

```

```

    },
    {
      "field": "timestamps.previous_topic.date",
      "format": "date_time"
    }
  ],
  "query": {
    "bool": {
      "must": [
        {
          "range": {
            "@timestamp": {
              "format": "strict_date_optional_time",
              "gte": "2020-02-06T10:46:30.000Z",
              "lte": "2020-02-06T10:51:30.000Z"
            }
          }
        }
      ]
    },
    "filter": [
      {
        "match_all": {}
      }
    ],
    "should": [],
    "must_not": []
  }
}

```

Sat-AIS Distribution by Satellite by Originator

```

{
  "aggs": {
    "2": {
      "terms": {
        "field": "originator",
        "order": {
          "_count": "desc"
        },
        "missing": "__missing__",
        "size": 20
      },
      "aggs": {
        "3": {
          "terms": {
            "field": "sat-ais.satellite_id",

```

```

        "order": {
            "_count": "desc"
        },
        "missing": "__missing__",
        "size": 20
    }
}
},
"size": 0,
"_source": {
    "excludes": []
},
"stored_fields": [
    "*"
],
"script_fields": {},
"docvalue_fields": [
    {
        "field": "@timestamp",
        "format": "date_time"
    },
    {
        "field": "timestamps.current_topic.date",
        "format": "date_time"
    },
    {
        "field": "timestamps.delivered.date",
        "format": "date_time"
    },
    {
        "field": "timestamps.message.date",
        "format": "date_time"
    },
    {
        "field": "timestamps.previous_topic.date",
        "format": "date_time"
    }
],
"query": {
    "bool": {
        "must": [
            {
                "match_phrase": {

```

```

    "source": {
      "query": "Sat-AIS"
    }
  },
  {
    "range": {
      "@timestamp": {
        "format": "strict_date_optional_time",
        "gte": "2020-02-06T10:46:30.000Z",
        "lte": "2020-02-06T10:51:30.000Z"
      }
    }
  },
  ],
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
}
}

```

#### Message Volume Hub-Originator Heatmap

```

{
  "aggs": {
    "2": {
      "terms": {
        "field": "hub_id",
        "order": {
          "_count": "desc"
        },
        "missing": "__missing__",
        "size": 10
      },
      "aggs": {
        "3": {
          "terms": {
            "field": "originator",
            "order": {
              "_count": "desc"
            },
            "missing": "__missing__",

```

```

        "size": 20
      }
    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  }
}

```

```

    }
  }
},
"filter": [
  {
    "match_all": {}
  }
],
"should": [],
"must_not": []
}
}
}

```

#### Message Volume Hub-Source Heatmap

```

{
  "aggs": {
    "2": {
      "terms": {
        "field": "hub_id",
        "order": {
          "_count": "desc"
        },
        "missing": "__missing__",
        "size": 10
      },
      "aggs": {
        "3": {
          "terms": {
            "field": "source",
            "order": {
              "_count": "desc"
            },
            "missing": "__missing__",
            "size": 20
          }
        }
      }
    },
    "size": 0,
    "_source": {
      "excludes": []
    },
    "stored_fields": [

```

```
"*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  },
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
```

```
}
}
}
```

### 16.3.3.2 AIS Delivery Latency Metrics

#### Average External Latency Map

```
{
  "size": 0,
  "aggs": {
    "grid": {
      "geotile_grid": {
        "field": "geo.position",
        "precision": 5
      },
      "aggs": {
        "1": {
          "geo_centroid": {
            "field": "geo.position"
          }
        },
        "avg_of_latency.message_to_delivered_ms": {
          "avg": {
            "field": "latency.message_to_delivered_ms"
          }
        }
      }
    },
    "_source": {
      "excludes": []
    },
    "stored_fields": [
      "*"
    ],
    "script_fields": {},
    "docvalue_fields": [
      {
        "field": "@timestamp",
        "format": "date_time"
      },
      {
        "field": "timestamps.current_topic.date",
        "format": "date_time"
      }
    ]
  }
}
```

```
{
  "field": "timestamps.delivered.date",
  "format": "date_time"
},
{
  "field": "timestamps.message.date",
  "format": "date_time"
},
{
  "field": "timestamps.previous_topic.date",
  "format": "date_time"
}
],
"query": {
  "bool": {
    "must": [
      {
        "geo_bounding_box": {
          "geo.position": {
            "top_left": [
              -180,
              90
            ],
            "bottom_right": [
              180,
              -10.9082800000000001
            ]
          }
        }
      }
    ],
    "range": {
      "@timestamp": {
        "format": "strict_date_optional_time",
        "gte": "2020-02-06T10:46:30.000Z",
        "lte": "2020-02-06T10:51:30.000Z"
      }
    }
  }
},
"filter": [
  {
    "match_all": {}
  }
],

```

<pre> "should": [], "must_not": [] } } } </pre>
Latency Overview
<pre> {   "aggs": {     "4": {       "date_histogram": {         "field": "timestamps.delivered.date",         "fixed_interval": "1s",         "time_zone": "Europe/Lisbon",         "min_doc_count": 1       },       "aggs": {         "1": {           "avg": {             "field": "latency.message_to_current_ms"           }         },         "2": {           "avg": {             "field": "latency.message_to_delivered_ms"           }         },         "3": {           "avg": {             "field": "latency.delivered_to_current_ms"           }         }       }     }   },   "size": 0,   "_source": {     "excludes": []   },   "stored_fields": [     "*"   ],   "script_fields": {},   "docvalue_fields": [     {       "field": "@timestamp", </pre>

```

    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "bool": {
          "should": [
            {
              "match_phrase": {
                "@es_source_reference": "valid-type-position-topic"
              }
            },
            {
              "match_phrase": {
                "@es_source_reference": "valid-type-voyage-topic"
              }
            }
          ]
        }
      },
      {
        "minimum_should_match": 1
      }
    ],
    "minimum_should_match": 1
  },
  {
    "range": {
      "@timestamp": {
        "format": "strict_date_optional_time",
        "gte": "2020-02-06T10:46:30.000Z",
        "lte": "2020-02-06T10:51:30.000Z"
      }
    }
  }
}

```

```

    }
  }
},
"filter": [
  {
    "match_all": {}
  }
],
"should": [],
"must_not": []
}
}
}

```

#### Average External Latency by Hub over Time

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "hub_id",
        "order": {
          "1": "desc"
        },
        "missing": "__missing__",
        "size": 10
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.message_to_delivered_ms"
          }
        },
        "2": {
          "date_histogram": {
            "field": "timestamps.delivered.date",
            "fixed_interval": "1s",
            "time_zone": "Europe/Lisbon",
            "min_doc_count": 1
          },
          "aggs": {
            "1": {
              "avg": {
                "field": "latency.message_to_delivered_ms"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  }
}

```

```

    }
  }
],
"filter": [
  {
    "match_all": {}
  }
],
"should": [],
"must_not": []
}
}
}

```

Average External Latency by Originator over Time

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "originator",
        "order": {
          "1": "desc"
        },
        "size": 5
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.message_to_delivered_ms"
          }
        },
        "2": {
          "date_histogram": {
            "field": "timestamps.delivered.date",
            "fixed_interval": "1s",
            "time_zone": "Europe/Lisbon",
            "min_doc_count": 1
          },
          "aggs": {
            "1": {
              "avg": {
                "field": "latency.message_to_delivered_ms"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  }
}

```

```

    ],
    "filter": [
      {
        "match_all": {}
      }
    ],
    "should": [],
    "must_not": []
  }
}

```

#### Average External Latency Hub-Source Heatmap

```

{
  "aggs": {
    "2": {
      "terms": {
        "field": "hub_id",
        "order": {
          "1": "desc"
        },
        "missing": "__missing__",
        "size": 10
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.delivered_to_current_ms"
          }
        },
        "3": {
          "terms": {
            "field": "source",
            "order": {
              "1": "desc"
            },
            "missing": "__missing__",
            "size": 20
          },
          "aggs": {
            "1": {
              "avg": {
                "field": "latency.delivered_to_current_ms"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T11:37:58.816Z",
            "lte": "2020-02-06T11:52:58.816Z"
          }
        }
      }
    ]
  }
}

```

```

    }
  ],
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
}
}

```

Average External Latency Hub-Originator Heatmap

```

{
  "aggs": {
    "2": {
      "terms": {
        "field": "hub_id",
        "order": {
          "1": "desc"
        },
        "missing": "__missing__",
        "size": 10
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.delivered_to_current_ms"
          }
        },
        "3": {
          "terms": {
            "field": "originator",
            "order": {
              "1": "desc"
            },
            "missing": "__missing__",
            "size": 20
          },
          "aggs": {
            "1": {
              "avg": {
                "field": "latency.delivered_to_current_ms"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ]
  }
}

```

```

    }
  }
],
"filter": [
  {
    "match_all": {}
  }
],
"should": [],
"must_not": []
}
}
}

```

#### Average Internal Latency over Time

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "@es_source_reference",
        "order": {
          "1": "desc"
        },
        "size": 5
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.delivered_to_current_ms"
          }
        },
        "2": {
          "date_histogram": {
            "field": "timestamps.delivered.date",
            "fixed_interval": "1s",
            "time_zone": "Europe/Lisbon",
            "min_doc_count": 1
          },
          "aggs": {
            "1": {
              "avg": {
                "field": "latency.delivered_to_current_ms"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      }
    ],
    {
      "bool": {

```

<pre> "should": [   {     "match_phrase": {       "@es_source_reference": "valid-type-position-topic"     }   },   {     "match_phrase": {       "@es_source_reference": "valid-type-voyage-topic"     }   } ], "minimum_should_match": 1 }, {   "range": {     "@timestamp": {       "format": "strict_date_optional_time",       "gte": "2020-02-06T10:46:30.000Z",       "lte": "2020-02-06T10:51:30.000Z"     }   } }, {   "filter": [     {       "match_all": {}     }   ],   "should": [],   "must_not": [] } } </pre>
Average Total Latency over Time
<pre> {   "aggs": {     "2": {       "date_histogram": {         "field": "timestamps.delivered.date",         "fixed_interval": "1s",         "time_zone": "Europe/Lisbon",         "min_doc_count": 1       }, </pre>

```

"aggs": {
  "3": {
    "terms": {
      "field": "hub_id",
      "order": {
        "1": "desc"
      },
      "missing": "__missing__",
      "size": 5
    },
    "aggs": {
      "1": {
        "avg": {
          "field": "latency.message_to_current_ms"
        }
      },
      "4": {
        "terms": {
          "field": "@es_source_reference",
          "order": {
            "1": "desc"
          },
          "missing": "__missing__",
          "size": 5
        },
        "aggs": {
          "1": {
            "avg": {
              "field": "latency.message_to_current_ms"
            }
          }
        }
      }
    }
  },
  "size": 0,
  "_source": {
    "excludes": []
  },
  "stored_fields": [
    "*"
  ],

```

```

"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      }
    ],
    {
      "bool": {
        "should": [
          {
            "match_phrase": {
              "@es_source_reference": "valid-type-position-topic"
            }
          },
          {
            "match_phrase": {
              "@es_source_reference": "valid-type-voyage-topic"
            }
          }
        ]
      }
    }
  }
}

```

```

    }
  ],
  "minimum_should_match": 1
}
},
{
  "range": {
    "@timestamp": {
      "format": "strict_date_optional_time",
      "gte": "2020-02-06T10:46:30.000Z",
      "lte": "2020-02-06T10:51:30.000Z"
    }
  }
}
],
"filter": [
  {
    "match_all": {}
  }
],
"should": [],
"must_not": []
}
}
}

```

### 16.3.3.3 Kafka Topic Metrics

Message Volume by Kafka Topic over Time

```

{
  "aggs": {
    "3": {
      "terms": {
        "field": "@es_source_reference",
        "order": {
          "_key": "desc"
        },
        "size": 10
      },
      "aggs": {
        "2": {
          "date_histogram": {
            "field": "timestamps.current_topic.date",
            "calendar_interval": "1m",
            "time_zone": "Europe/Lisbon",

```

```
    "min_doc_count": 0
  },
  "aggs": {
    "1": {
      "moving_fn": {
        "buckets_path": "_count",
        "window": 5,
        "script": "MovingFunctions.unweightedAvg(values)"
      }
    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.previous_topic.date",
    "format": "date_time"
  }
],
```

```
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      },
      {
        "range": {
          "@timestamp": {
            "format": "strict_date_optional_time",
            "gte": "2020-02-06T10:46:30.000Z",
            "lte": "2020-02-06T10:51:30.000Z"
          }
        }
      }
    ],
    "filter": [
      {
        "match_all": {}
      }
    ],
    "should": [],
    "must_not": []
  }
}
```

Kafka Average Processing Latency by Topic over Time

```
{
  "aggs": {
    "3": {
      "terms": {
        "field": "@es_source_reference",
        "order": {
          "1": "desc"
        },
        "size": 5
      },
      "aggs": {
        "1": {
          "avg": {
            "field": "latency.previous_to_current_ms"
          }
        }
      }
    }
  }
}
```

```

    }
  },
  "2": {
    "date_histogram": {
      "field": "timestamps.current_topic.date",
      "fixed_interval": "1s",
      "time_zone": "Europe/Lisbon",
      "min_doc_count": 1
    },
    "aggs": {
      "1": {
        "avg": {
          "field": "latency.previous_to_current_ms"
        }
      }
    }
  }
},
"size": 0,
"_source": {
  "excludes": []
},
"stored_fields": [
  "*"
],
"script_fields": {},
"docvalue_fields": [
  {
    "field": "@timestamp",
    "format": "date_time"
  },
  {
    "field": "timestamps.current_topic.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.delivered.date",
    "format": "date_time"
  },
  {
    "field": "timestamps.message.date",
    "format": "date_time"
  },

```

```
{
  "field": "timestamps.previous_topic.date",
  "format": "date_time"
},
],
"query": {
  "bool": {
    "must": [
      {
        "match_phrase": {
          "@es_source": {
            "query": "kafka"
          }
        }
      }
    ],
    {
      "range": {
        "@timestamp": {
          "format": "strict_date_optional_time",
          "gte": "2020-02-06T10:46:30.000Z",
          "lte": "2020-02-06T10:51:30.000Z"
        }
      }
    }
  ],
  "filter": [
    {
      "match_all": {}
    }
  ],
  "should": [],
  "must_not": []
}
}
```

### 16.3.4 Response Description (json)

The response body returns the query results. Its general structure is explained here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html#search-api-response-body>

The aggregations field contains the data of interest in the form of one or multiple data series in buckets of one or multiple variables bound by the ranges defined in the query. More information about Bucket Aggregations can be found here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket.html>

## 16.3.5 Response Examples (json)

### 16.3.5.1 AIS Delivery Volume Metrics

Reported Position Density Map
<pre>{   "took": 3,   "timed_out": false,   "_shards": {     "total": 4,     "successful": 4,     "skipped": 0,     "failed": 0   },   "hits": {     "total": 235,     "max_score": null,     "hits": []   },   "aggregations": {     "grid": {       "buckets": [         {           "1": {             "location": {               "lat": 55.38336533489392,               "lon": 10.771771176277023             },             "count": 19           },           "key": "8/135/80",           "doc_count": 19         },         ...         {           "1": {             "location": {               "lat": 51.69883331749588,               "lon": -9.533333340659738             },             "count": 1           }         },       ]     }   } }</pre>

```

    "key": "8/121/84",
    "doc_count": 1
  }
]
},
"status": 200
}

```

#### Volume Overview

```

{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 6,
    "successful": 6,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "2": {
            "buckets": [
              {
                "key_as_string": "2020-02-06T10:46:00.000Z",
                "key": 1580985960000,
                "doc_count": 54
              },
              ...
              {
                "key_as_string": "2020-02-06T10:51:00.000Z",
                "key": 1580986260000,
                "doc_count": 57
              }
            ]
          },
          ...
        }
      ],
      "key": "valid-messages",

```

```

    "doc_count": 576
  },
  {
    "2": {
      "buckets": [
        {
          "key_as_string": "2020-02-06T10:46:00.000Z",
          "key": 1580985960000,
          "doc_count": 54
        },
        ...
        {
          "key_as_string": "2020-02-06T10:51:00.000Z",
          "key": 1580986260000,
          "doc_count": 57
        }
      ]
    },
    "key": "valid-type-position-topic",
    "doc_count": 367
  }
]
}
},
"status": 200
}

```

Message volume by Hub over Time

```

{
  "took": 8,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,

```

```

"buckets": [
  {
    "2": {
      "buckets": [
        {
          "key_as_string": "2020-02-06T10:46:00.000Z",
          "key": 1580985960000,
          "doc_count": 108
        },
        {
          ...
        }
      ]
    },
    "key": "hub-1",
    "doc_count": 1152
  },
  ...
  {
    "2": {
      "buckets": [
        {
          "key_as_string": "2020-02-06T10:46:00.000Z",
          "key": 1580985960000,
          "doc_count": 108
        },
        {
          ...
        }
      ]
    },
    "key": "hub-2",
    "doc_count": 1589
  }
]
}

```

<pre>     },     "status": 200   } </pre>
Sat-AIS Message Volume by Satellite over Time
<pre> {   "took": 1,   "timed_out": false,   "_shards": {     "total": 4,     "successful": 4,     "skipped": 0,     "failed": 0   },   "hits": {     "total": 58,     "max_score": null,     "hits": []   },   "aggregations": {     "3": {       "doc_count_error_upper_bound": 0,       "sum_other_doc_count": 0,       "buckets": [         {           "2": {             "buckets": [               {                 "key_as_string": "2020-02-06T10:46:00.000Z",                 "key": 1580985960000,                 "doc_count": 2               },               ...               {                 "key_as_string": "2020-02-06T10:51:00.000Z",                 "key": 1580986260000,                 "doc_count": 2               }             ]           },           "key": "__missing__",           "doc_count": 58         }       ]     }   }, } </pre>

"status": 200 }
Message Type distribution by Hub
<pre> {   "took": 1,   "timed_out": false,   "_shards": {     "total": 4,     "successful": 4,     "skipped": 0,     "failed": 0   },   "hits": {     "total": 1152,     "max_score": null,     "hits": []   },   "aggregations": {     "2": {       "doc_count_error_upper_bound": 0,       "sum_other_doc_count": 0,       "buckets": [         {           "3": {             "doc_count_error_upper_bound": 0,             "sum_other_doc_count": 0,             "buckets": [               {                 "key": "5",                 "doc_count": 216               },               ...               {                 "key": "2",                 "doc_count": 32               }             ]           },           "key": "hub-1",           "doc_count": 1152         },         ...         {           "3": {             "doc_count_error_upper_bound": 0, </pre>

```

        "sum_other_doc_count": 0,
        "buckets": [
          {
            "key": "5",
            "doc_count": 216
          },
          ...
          {
            "key": "2",
            "doc_count": 32
          }
        ]
      },
      "key": "hub-2",
      "doc_count": 1568
    }
  ]
}
},
"status": 200
}

```

#### Sat-AIS Distribution by Satellite by Originator

```

{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 58,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "2": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "3": {
            "doc_count_error_upper_bound": 0,

```

```

    "sum_other_doc_count": 0,
    "buckets": [
      {
        "key": "__missing__",
        "doc_count": 10
      }
    ]
  },
  ...
  {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": "__missing__",
          "doc_count": 2
        }
      ]
    },
    "key": "SWE",
    "doc_count": 2
  }
]
}
},
"status": 200
}

```

#### Message Volume Hub-Originator Heatmap

```

{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "2": {

```

```
"doc_count_error_upper_bound": 0,
"sum_other_doc_count": 0,
"buckets": [
  {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": "FRA",
          "doc_count": 108
        },
        ...
        {
          "key": "BEL",
          "doc_count": 70
        }
      ]
    },
    "key": "hub-1",
    "doc_count": 1152
  },
  {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": "FRA",
          "doc_count": 95
        },
        ...
        {
          "key": "BEL",
          "doc_count": 45
        }
      ]
    },
    "key": "hub-2",
    "doc_count": 1152
  }
]
}
```

"status": 200 }
Message Volume Hub-Source Heatmap
<pre> {   "took": 1,   "timed_out": false,   "_shards": {     "total": 4,     "successful": 4,     "skipped": 0,     "failed": 0   },   "hits": {     "total": 1152,     "max_score": null,     "hits": []   },   "aggregations": {     "2": {       "doc_count_error_upper_bound": 0,       "sum_other_doc_count": 0,       "buckets": [         {           "3": {             "doc_count_error_upper_bound": 0,             "sum_other_doc_count": 0,             "buckets": [               {                 "key": "T-AIS",                 "doc_count": 88               },               {                 "key": "Sat-AIS",                 "doc_count": 48               }             ]           }         ]       },       "key": "hub-1",       "doc_count": 1152     },     ...     {       "3": {         "doc_count_error_upper_bound": 0,         "sum_other_doc_count": 0, </pre>

```

    "buckets": [
      {
        "key": "Sat-AIS",
        "doc_count": 85
      },
      {
        "key": "T-AIS",
        "doc_count": 41
      }
    ]
  },
  "key": "hub-2",
  "doc_count": 1152
}
]
}
},
"status": 200
}

```

### 16.3.5.2 AIS Delivery Latency Metrics

#### Average External Latency Map

```

{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 387,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "grid": {
      "buckets": [
        {
          "1": {
            "location": {
              "lat": 52.85581906600409,

```

```

        "lon": 6.066794765806974
      },
      "count": 73
    },
    "key": "5/16/10",
    "doc_count": 73,
    "avg_of_latency.message_to_delivered_ms": {
      "value": 1561.5616438356165
    }
  },
  ...
  {
    "1": {
      "location": {
        "lat": 46.37166664469987,
        "lon": -124.67000003904104
      },
      "count": 1
    },
    "key": "5/4/11",
    "doc_count": 1,
    "avg_of_latency.message_to_delivered_ms": {
      "value": 3325
    }
  }
]
}
},
"status": 200
}

```

#### Latency Overview

```

{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 6,
    "successful": 6,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
}

```

```
"aggregations": {
  "4": {
    "buckets": [
      {
        "1": {
          "value": 2771.5
        },
        "2": {
          "value": 2760
        },
        "3": {
          "value": 11.5
        },
        "key_as_string": "2020-02-06T10:46:32.000Z",
        "key": 1580985992000,
        "doc_count": 2
      },
      ...
      {
        "1": {
          "value": 479.5
        },
        "2": {
          "value": 463
        },
        "3": {
          "value": 16.5
        },
        "key_as_string": "2020-02-06T10:51:29.000Z",
        "key": 1580986289000,
        "doc_count": 4
      }
    ]
  }
},
"status": 200
}
```

Average External Latency by Hub over Time

```
{
  "took": 8,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
```

```
"failed": 0
},
"hits": {
  "total": 1152,
  "max_score": null,
  "hits": []
},
"aggregations": {
  "3": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 0,
    "buckets": [
      {
        "1": {
          "value": 1620.1475694444443
        },
        "2": {
          "buckets": [
            {
              "1": {
                "value": 2760
              },
              "key_as_string": "2020-02-06T10:46:32.000Z",
              "key": 1580985992000,
              "doc_count": 3
            },
            ...
            {
              "1": {
                "value": 463
              },
              "key_as_string": "2020-02-06T10:51:29.000Z",
              "key": 1580986289000,
              "doc_count": 4
            }
          ]
        },
        "key": "hub-1",
        "doc_count": 1152
      },
      {
        "1": {
          "value": 1620.1475694444443
        },
        "2": {
```

```

    "buckets": [
      {
        "1": {
          "value": 2760
        },
        "key_as_string": "2020-02-06T10:46:32.000Z",
        "key": 1580985992000,
        "doc_count": 2
      },
      ...
      {
        "1": {
          "value": 463
        },
        "key_as_string": "2020-02-06T10:51:29.000Z",
        "key": 1580986289000,
        "doc_count": 7
      }
    ]
  },
  "key": "hub-2",
  "doc_count": 1152
}
]
}
},
"status": 200
}

```

#### Average External Latency by Originator over Time

```

{
  "took": 21,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "3": {

```

```
"doc_count_error_upper_bound": 0,
"sum_other_doc_count": 714,
"buckets": [
  {
    "1": {
      "value": 1933.8085106382978
    },
    "2": {
      "buckets": [
        {
          "1": {
            "value": 3265
          },
          "key_as_string": "2020-02-06T10:46:33.000Z",
          "key": 1580985993000,
          "doc_count": 2
        },
        ...
        {
          "1": {
            "value": 713
          },
          "key_as_string": "2020-02-06T10:51:29.000Z",
          "key": 1580986289000,
          "doc_count": 2
        }
      ]
    },
    "key": "RUS",
    "doc_count": 94
  },
  ...
  {
    "1": {
      "value": 1820.5217391304348
    },
    "2": {
      "buckets": [
        {
          "1": {
            "value": 2766
          },
          "key_as_string": "2020-02-06T10:46:40.000Z",
          "key": 1580986000000,
          "doc_count": 2
        }
      ]
    }
  }
]
```

```

    },
    ...
    {
      "1": {
        "value": 207
      },
      "key_as_string": "2020-02-06T10:51:19.000Z",
      "key": 1580986279000,
      "doc_count": 2
    }
  ]
},
"key": "NOR",
"doc_count": 74
}
]
}
},
"status": 200
}

```

#### Average External Latency Hub-Source Heatmap

```

{
  "took": 8,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 3412,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "2": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "1": {
            "value": 19.422332942555684
          },
          "3": {

```

```
"doc_count_error_upper_bound": 0,
"sum_other_doc_count": 0,
"buckets": [
  {
    "1": {
      "value": 21.7816091954023
    },
    "key": "T-AIS",
    "doc_count": 154
  },
  ...
  {
    "1": {
      "value": 18.063157894736843
    },
    "key": "Sat-AIS",
    "doc_count": 180
  }
]
},
"key": "hub-1",
"doc_count": 3412
},
...
{
  "1": {
    "value": 19.422332942555684
  },
  "3": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 0,
    "buckets": [
      {
        "1": {
          "value": 21.7816091954023
        },
        "key": "T-AIS",
        "doc_count": 176
      },
      ...
      {
        "1": {
          "value": 18.063157894736843
        },
        "key": "Sat-AIS",
```

```

        "doc_count": 190
      }
    ]
  },
  "key": "hub-2",
  "doc_count": 3666
}
]
}
},
"status": 200
}

```

#### Average External Latency Hub-Originator Heatmap

```

{
  "took": 5,
  "timed_out": false,
  "_shards": {
    "total": 4,
    "successful": 4,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "2": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "1": {
            "value": 23.61892361111111
          },
          "3": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
              {
                "1": {
                  "value": 28.488888888888887
                },

```

```

        "doc_count": 90
      },
      ...
    {
      "1": {
        "value": 20.68918918918919
      },
      "key": "SWE",
      "doc_count": 74
    }
  ]
},
"key": "hub-1",
"doc_count": 1152
},
...
{
  "1": {
    "value": 23.61892361111111
  },
  "3": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 0,
    "buckets": [
      {
        "1": {
          "value": 28.488888888888887
        },
        "key": "FIN",
        "doc_count": 81
      },
      ...
    ]
  },
  "1": {
    "value": 20.68918918918919
  },
  "key": "SWE",
  "doc_count": 71
}
]
},
"key": "hub-2",
"doc_count": 1077
}
]

```

<pre> } }, "status": 200 } </pre>
Average Internal Latency over Time
<pre> {   "took": 1,   "timed_out": false,   "_shards": {     "total": 6,     "successful": 6,     "skipped": 0,     "failed": 0   },   "hits": {     "total": 1152,     "max_score": null,     "hits": []   },   "aggregations": {     "3": {       "doc_count_error_upper_bound": 0,       "sum_other_doc_count": 0,       "buckets": [         {           "1": {             "value": 32.91317484444444           },           "2": {             "buckets": [               {                 "1": {                   "value": 21                 },                 "key_as_string": "2020-02-06T10:46:32.000Z",                 "key": 1580985992000,                 "doc_count": 1               },               ...               {                 "1": {                   "value": 27.5                 },                 "key_as_string": "2020-02-06T10:51:29.000Z",                 "key": 1580986289000, </pre>

<pre>         "doc_count": 2       }     ]   },   "key": "valid-messages",   "doc_count": 789 }, ... {   "1": {     "value": 5.324652777777778   },   "2": {     "buckets": [       {         "1": {           "value": 2         },         "key_as_string": "2020-02-06T10:46:32.000Z",         "key": 1580985992000,         "doc_count": 1       },       ...       {         "1": {           "value": 5.5         },         "key_as_string": "2020-02-06T10:51:29.000Z",         "key": 1580986289000,         "doc_count": 2       }     ]   },   "key": "valid-type-position-topic",   "doc_count": 576 } ] } }, "status": 200 } </pre>
Average Total Latency over Time
<pre> {   "took": 10,   "timed_out": false, </pre>

```
"_shards": {
  "total": 6,
  "successful": 6,
  "skipped": 0,
  "failed": 0
},
"hits": {
  "total": 1152,
  "max_score": null,
  "hits": []
},
"aggregations": {
  "2": {
    "buckets": [
      {
        "3": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "1": {
                "value": 2771.5
              },
              "4": {
                "doc_count_error_upper_bound": 0,
                "sum_other_doc_count": 0,
                "buckets": [
                  {
                    "1": {
                      "value": 2781
                    },
                    "key": "valid-messages",
                    "doc_count": 1
                  },
                  ...
                  {
                    "1": {
                      "value": 2762
                    },
                    "key": "valid-type-position-topic",
                    "doc_count": 1
                  }
                ]
              },
              "key": "hub-1",
```

```
        "doc_count": 2
      }
    ]
  },
  "key_as_string": "2020-02-06T10:46:32.000Z",
  "key": 1580985992000,
  "doc_count": 2
},
...
{
  "3": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 0,
    "buckets": [
      {
        "1": {
          "value": 479.5
        },
        "4": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "1": {
                "value": 490.5
              },
              "key": "valid-messages",
              "doc_count": 2
            },
            ...
            {
              "1": {
                "value": 468.5
              },
              "key": "valid-type-position-topic",
              "doc_count": 2
            }
          ]
        },
        "key": "hub-1",
        "doc_count": 4
      },
      ...
      "1": {
        "value": 356.5
```

```

    },
    "4": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "1": {
            "value": 490.5
          },
          "key": "valid-messages",
          "doc_count": 2
        },
        ...
        {
          "1": {
            "value": 468.5
          },
          "key": "valid-type-position-topic",
          "doc_count": 2
        }
      ]
    },
    "key": "hub-2",
    "doc_count": 4
  }
]
},
"key_as_string": "2020-02-06T10:51:29.000Z",
"key": 1580986289000,
"doc_count": 4
}
]
}
},
"status": 200
}

```

### 16.3.5.3 Kafka Topic Metrics

Message Volume by Kafka Topic over Time

```

{
  "took": 14,
  "timed_out": false,
  "_shards": {
    "total": 6,

```

```

"successful": 6,
"skipped": 0,
"failed": 0
},
"hits": {
  "total": 1152,
  "max_score": null,
  "hits": []
},
"aggregations": {
  "3": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 0,
    "buckets": [
      {
        "2": {
          "buckets": [
            {
              "1": {
                "value": null
              },
              "key_as_string": "2020-02-06T10:46:00.000Z",
              "key": 1580985960000,
              "doc_count": 94
            },
            ...
            {
              "1": {
                "value": 103.8
              },
              "key_as_string": "2020-02-06T10:51:00.000Z",
              "key": 1580986260000,
              "doc_count": 78
            }
          ]
        },
        "key": "valid-messages",
        "doc_count": 879
      },
      ...
      {
        "2": {
          "buckets": [
            {
              "1": {

```

```

        "value": null
      },
      "key_as_string": "2020-02-06T10:46:00.000Z",
      "key": 1580985960000,
      "doc_count": 54
    },
    ...
    {
      "1": {
        "value": 103.8
      },
      "key_as_string": "2020-02-06T10:51:00.000Z",
      "key": 1580986260000,
      "doc_count": 57
    }
  ]
},
"key": "valid-type-position-topic",
"doc_count": 576
}
]
}
},
"status": 200
}

```

#### Kafka Average Processing Latency by Topic over Time

```

{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 6,
    "successful": 6,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1152,
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "3": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [

```

```
{
  "1": {
    "value": 36.588541666666664
  },
  "2": {
    "buckets": [
      {
        "1": {
          "value": 19
        },
        "key_as_string": "2020-02-06T10:46:32.000Z",
        "key": 1580985992000,
        "doc_count": 1
      },
      ...
    ]
  },
  "key": "valid-messages",
  "doc_count": 576
},
{
  "1": {
    "value": 2.5034722222222223
  },
  "2": {
    "buckets": [
      {
        "1": {
          "value": 1
        },
        "key_as_string": "2020-02-06T10:46:32.000Z",
        "key": 1580985992000,
        "doc_count": 1
      },
      ...
    ]
  },
  "1": {
```

```
        "value": 1.5
      },
      "key_as_string": "2020-02-06T10:51:29.000Z",
      "key": 1580986289000,
      "doc_count": 2
    }
  ]
},
"key": "valid-type-position-topic",
"doc_count": 345
}
]
}
},
"status": 200
}
```