# Maritime Track Fusion

Additional Tags:

Tags to be removed:

Operator Key: streaming:maritime\_fusion

Group: infore\_project/partner\_services

## Description

This operator uses the Maritime Track Fusion Service to perform a fusion of different streamed maritime input data streams in a streaming analytic workflow.

To utilize this operators functionality an Akka cluster with the Track Fusion Algorithm has to be set up.  
The Kafka cluster and the names of the input and output topics used by the Track Fusion has to be provided to the operator.

The operator starts the Track Fusion on the Akka cluster (connection information through Maritime Connection).  
Then the input data events received at the different *input stream* ports are pushed to the data topics of the Kafka cluster.  
Any number of different input data streams can be connected to the input ports of the operator.   
These different input streams can be for example the AIS data stream of ship positions, sensor data or similar.  
All input data events needs to have the mandatory fields (t, lon, lat) describing timestamp, longitude and latitude.   
If the data events contain the field "ship" it is assumed that the data stream is coming from AIS, otherwise from another source/sensor.

The Maritime Event Detection will perform the fusion which combines the data from the different input streams and creates one output stream of fused events.  
These fused events are pushed to the *output topic* of the Kafka cluster.  
The operator reads from the output topic of the Kafka cluster and pushes the fused events further downstream (to the *output stream* port).

This is a streaming operator and needs to be placed inside a Streaming Nest or a Streaming Optimization operator.  
The operator defines the logical functionality and can be used in all streaming analytic workflow for any supported streaming platform (currently Flink and Spark).  
The actual implementation used depends on the type of connection connected to the Streaming Nest operator in which this operator is placed.

## Tutorial Process

#### Tutorial 1 (Simple Maritime Track Fusion)

In this tutorial process the usage of the Maritime Track Fusion operator is demonstrated.

<?xml version="1.0" encoding="UTF-8"?><process version="9.10.001">  
 <context>  
 <input/>  
 <output/>  
 <macros/>  
 </context>  
 <operator activated="true" class="process" compatibility="9.10.001" expanded="true" name="Process">  
 <parameter key="logverbosity" value="init"/>  
 <parameter key="random\_seed" value="2001"/>  
 <parameter key="send\_mail" value="never"/>  
 <parameter key="notification\_email" value=""/>  
 <parameter key="process\_duration\_for\_mail" value="30"/>  
 <parameter key="encoding" value="SYSTEM"/>  
 <process expanded="true">  
 <operator activated="true" class="retrieve" compatibility="9.10.001" expanded="true" height="68" name="Retrieve Freya Flink" width="90" x="112" y="34">  
 <parameter key="repository\_entry" value="/Connections/Freya Flink"/>  
 </operator>  
 <operator activated="true" class="streaming:streaming\_nest" compatibility="0.6.002-SNAPSHOT" expanded="true" height="82" name="Streaming Nest" width="90" x="313" y="34">  
 <parameter key="job\_name" value="Track Fusion Test"/>  
 <process expanded="true">  
 <operator activated="true" class="retrieve" compatibility="9.10.001" expanded="true" height="68" name="Retrieve Freya Kafka" width="90" x="112" y="136">  
 <parameter key="repository\_entry" value="/Connections/Freya Kafka"/>  
 </operator>  
 <operator activated="true" class="retrieve" compatibility="9.10.001" expanded="true" height="68" name="Retrieve Maritime Cluster" width="90" x="112" y="34">  
 <parameter key="repository\_entry" value="/Connections/Maritime Cluster"/>  
 </operator>  
 <operator activated="true" class="multiply" compatibility="9.10.001" expanded="true" height="166" name="Multiply" width="90" x="246" y="85"/>  
 <operator activated="true" class="streaming:kafka\_source" compatibility="0.6.002-SNAPSHOT" expanded="true" height="82" name="Kafka Source (3)" width="90" x="380" y="340">  
 <parameter key="kafka\_topic" value="uav"/>  
 <parameter key="start\_from\_earliest" value="true"/>  
 </operator>  
 <operator activated="true" class="streaming:kafka\_source" compatibility="0.6.002-SNAPSHOT" expanded="true" height="82" name="Kafka Source (2)" width="90" x="380" y="238">  
 <parameter key="kafka\_topic" value="satelitte"/>  
 <parameter key="start\_from\_earliest" value="true"/>  
 </operator>  
 <operator activated="true" class="streaming:kafka\_source" compatibility="0.6.002-SNAPSHOT" expanded="true" height="82" name="Kafka Source" width="90" x="380" y="136">  
 <parameter key="kafka\_topic" value="ais"/>  
 <parameter key="start\_from\_earliest" value="true"/>  
 </operator>  
 <operator activated="true" class="streaming:maritime\_fusion" compatibility="0.6.002-SNAPSHOT" expanded="true" height="166" name="Maritime Track Fusion" width="90" x="514" y="34">  
 <enumeration key="input\_topics">  
 <parameter key="topic\_name" value="ais"/>  
 <parameter key="topic\_name" value="satelitte"/>  
 </enumeration>  
 <parameter key="output\_topic" value="output"/>  
 </operator>  
 <operator activated="true" class="streaming:kafka\_sink" compatibility="0.6.002-SNAPSHOT" expanded="true" height="82" name="Kafka Sink" width="90" x="715" y="85">  
 <parameter key="kafka\_topic" value="output"/>  
 </operator>  
 <connect from\_op="Retrieve Freya Kafka" from\_port="output" to\_op="Multiply" to\_port="input"/>  
 <connect from\_op="Retrieve Maritime Cluster" from\_port="output" to\_op="Maritime Track Fusion" to\_port="maritime-connection"/>  
 <connect from\_op="Multiply" from\_port="output 1" to\_op="Maritime Track Fusion" to\_port="kafka-connection"/>  
 <connect from\_op="Multiply" from\_port="output 2" to\_op="Kafka Source" to\_port="connection"/>  
 <connect from\_op="Multiply" from\_port="output 3" to\_op="Kafka Source (2)" to\_port="connection"/>  
 <connect from\_op="Multiply" from\_port="output 4" to\_op="Kafka Source (3)" to\_port="connection"/>  
 <connect from\_op="Multiply" from\_port="output 5" to\_op="Kafka Sink" to\_port="connection"/>  
 <connect from\_op="Kafka Source (3)" from\_port="output stream" to\_op="Maritime Track Fusion" to\_port="input stream 3"/>  
 <connect from\_op="Kafka Source (2)" from\_port="output stream" to\_op="Maritime Track Fusion" to\_port="input stream 2"/>  
 <connect from\_op="Kafka Source" from\_port="output stream" to\_op="Maritime Track Fusion" to\_port="input stream 1"/>  
 <connect from\_op="Maritime Track Fusion" from\_port="output stream" to\_op="Kafka Sink" to\_port="input stream"/>  
 <portSpacing port="source\_in 1" spacing="0"/>  
 <portSpacing port="sink\_out 1" spacing="0"/>  
 </process>  
 </operator>  
 <connect from\_op="Retrieve Freya Flink" from\_port="output" to\_op="Streaming Nest" to\_port="connection"/>  
 <portSpacing port="source\_input 1" spacing="0"/>  
 <portSpacing port="sink\_result 1" spacing="0"/>  
 </process>  
 </operator>  
</process>

## Parameters

#### input topics

Names of the Kafka topics used by the Track Fusion algorithm to receive input data events.  
If there are more input data streams connected, than topic names provided, the operator automatically creates generic topic names for the additional input data streams.

#### output topic

Name of the Kafka topic used by the Track Fusion algorithm to push fused output data events to.

#### key field

Name of key field to use for Kafka records.

## Input

#### maritime-connection (Connection)

The connection to the Maritime Akka Cluster to start the Maritime Track Fusion algorithm.

#### kafka-connection (Connection)

The connection to the Kafka Cluster which is used by the Maritime Track Fusion Service for communication.

#### input streams (Stream Data Container)

The input of this streaming operation.  
It needs to receive the output of a preceding streaming operator, to define the flow of data events in the streaming analytic workflow.  
Any number of input data streams can be connected and will be fused together.

## Output

#### output stream (Stream Data Container)

The output of this streaming operation.  
Connect it to the next Streaming operator to define the flow of the data events in the designed streaming analytic workflow.