Science Demonstrator 3: SOS & SSN Ontology Based Data Acquisition & Near Real Time Quality Control (Use Case IC_14)

Overview

The Service allows to submit and publish raw observational (non-geophysical) environmental timeseries data in common standard formats (T-SOS XML and SSNO JSON) via a messaging API (EGI ARGO) that is used to perform Near Real Time (NRT) quality control procedures by an Apache Storm NRT QC Topology, which publishes the quality controlled and labelled data via a messaging output queue.

Scientific Objectives

Research Infrastructures, specifically observatories that build on environmental sensor networks, share a common problem: data acquisition services and, in particular, the preparation of data transfer prior to data transmission are often not yet sufficiently standardized. This hinders the operation of efficient, cross-RI data processing routines, e.g. for data quality checking.

The overall objective of this implementation case is to move the standardization level close to the sensors of RIs, thus allowing the implementation of common, generic data processing routines, e.g. for Near Real Time (NRT) Quality Control (QC).

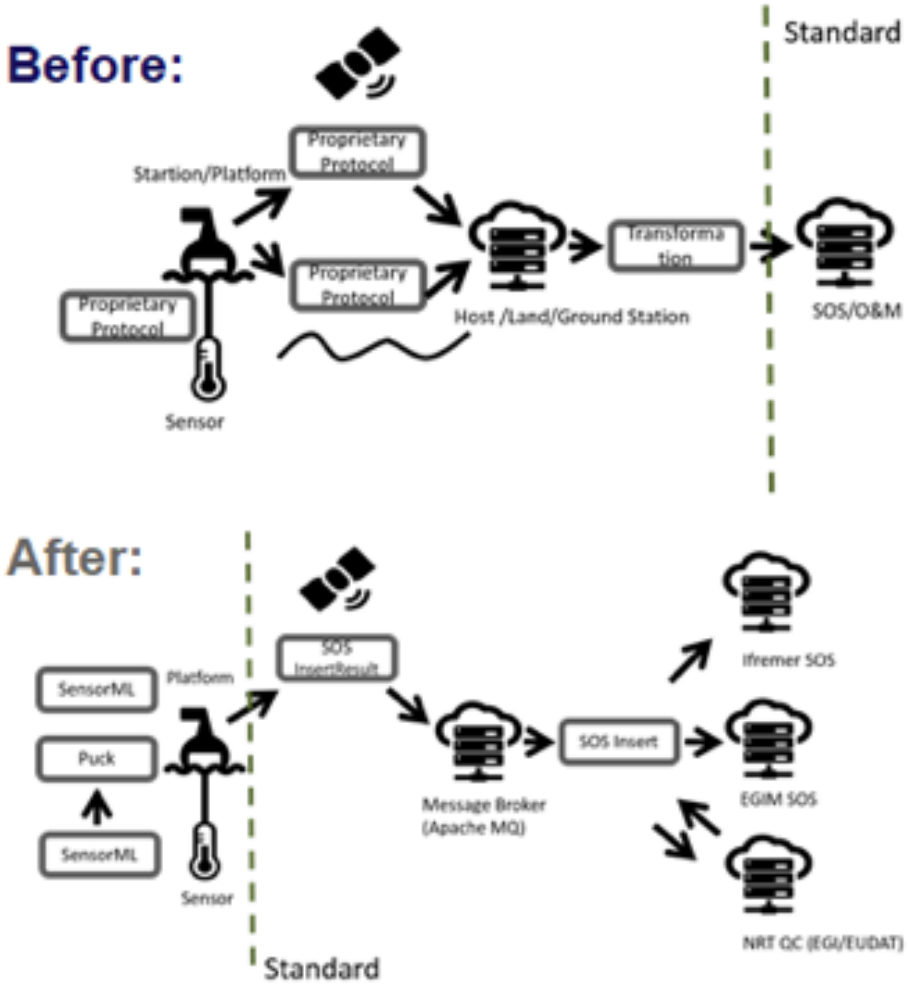

Figure 1. Moving Standards closer to the sensor. Today many RIs use proprietary protocols and formats within their data flow (Before). Standards such as SSNO (not shown here), SOS (Sensor Observation service) or O&M (Observation & measuremenT) frequently are only used to publish

data. Our approach was to use these standards as early as possible (After): use PUCK to expose SensorML metadata, transmit data as T-SOS Insertresult XML and issue these data at potentially multiple SOS servers as well as for consumption by a NRT QC service.

A further objective is to contribute to the harmonization of data transmission formats and protocols.

This science demonstrator use case aims to evaluate standardized data transmission using OGC SWE Transactional SOS (Sensor Observation Service) as a priority standard as well as using the Semantic Sensor Network (SSN) ontology. Both are implemented and tested. It will identify and implement common generic NRT QC routines suitable for multiple RIs (e.g. EMSO, EuroARGO, ANAEE, etc.) and deploy these on appropriate scalable cloud based technologies at own and/or EGI platforms.

Description

Figure 2. Architecture of the NRT QC service

The service is based on two main cloud based components: 1) an Apache Storm data processing unit responsible for near real time data quality control of a given real-time time series data and 2) a EGI ARGO messaging component responsible for the management and queuing of data sent from the sensor as well as for the delivery of quality controlled data.

As illustrated in Figure 2, the data processing and quality control builds on Apache Storm to support scalable NRT QC on streamed standardized sensor data. Apache Storm is a distributed real-time computation system. It specializes on reliable processing of data streams and is designed to support real-time analytics and continuous computation. Central to Apache Storm is the notion of Storm topology. Topology nodes are either spouts or bolts. Vertices are streams. A stream is an unbounded sequence of tuples. Tuples are data packages. A spout is a source of streams in a topology. Bolts perform computations (processing) on tuples.

Data from a sensor system using transactional SOS typically is sent in distinct intervals. We therefore chose to use EGI's ARGO messaging system which is built upon Kafka to manage such incoming messages. ARGO provides a well-documented REST API which is easy to use and based on JSON envelopes for messages wherein base64[1] encoded content can be sent. We have set up one message queue (topic) for incoming messages and another queue for delivery of quality labelled data.

Supported data messages are either base64 encoded T-SOS XML strings or SSNO based JSON strings. In order to ease things for processing we decided to transform these data messages into individual atomic observation objects based on SSNO.

Within the Storm topology we have implemented several nodes:

MessageReader is a BaseRichSpout which continuously reads messages from the ARGO message queue. As ARGO can send multiple messages within one API response, the spout splits theses messages into individual message objects and emits these as tuples for further processing within the topology.

MessageAtomizer Bolt is a BaseRichBolt which collects sensor metadata from sensor URLs given in the T-SOS or SSNO such as sensor specific measurement ranges. It recognizes the sent data format (SSNO or T-SOS), splits the message objects into atomic observation object tuples and emits these tuples.

RangeCheckController Bolt is a BaseRichBolt which takes these emitted tuples and checks if each numeric value is within the measurement range of the sensor specification. It adds a qualityOfObservation value (0=passed, 1=failed) to each atomic value and emits these as tuples.

OutlierController Bolt is a BaseWindowedBolt which collects a given number of atomic value tuples and performs a simple outlier check based on a modified z-score. The bolt again adds a qualityOfObservation value to each checked atomic value and emits these as tuples.

QualityControlledMessagePacker Bolt is a BaseWindowedBolt which collects a given number of quality checked atomic values and adds these into a JSON array which is the sent as payload to the ARGO messaging queue for QC'd messages where it is available for further consumption by an appropriate domain specific service to update or clean raw data holdings.

Advantages

Near real time quality control is a common problem for Research Infrastructures. Whereas domain specific NRT routines and standards exist, such as those for ICOS, ARGO or IAGOS, we have shown (see ENVRIplus deliverable D3.3) that clear communalities among those RIs with respect to NRT QC routines exist.

Figure 3. commonly used NRT quality routines within ENVRIplus RIs

Most commonly used are simple test such as outlier or spike detection, gradient or stuck value tests. A domain independent service able to perform these tests would therefore be an added value for all ENVRIplus RIs. RIs which have their own services in place could use it for

cross-validation of their QC results and it would give those RIs the opportunity to perform routine NRT QC checks which do not yet have own routines in place.

Unfortunately, data transmission formats used within RIs is very diverse. In general, data transmission at the sensor as well as platform level largely depends on community specific needs and habits or simply on manufacturer specifications. It is therefore difficult to offer generic, cross RI processing services in general and in particular services which allow NRT QC. It is therefore clearly advantageous for the scientific community to have access to standard supporting services. Further, such services potentially can strongly promote the use of these standards - one of the main objectives of this use case.

Link to the Demonstrator

Youtube video is at https://youtu.be/p3UQZkRRWlw

Contributors

- Dr Rober Huber, University of Bremen, rhuber@uni-bremen.de

Reference

[1] base64 is a group of binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation.