Where Big Data meets Linked Data: Applying standard data models to environmental data streams

Adam Leadbetter,
Damian Smyth,
Robert Fuller
and Eoin O'Grady
Marine Institute
Rinville, Oranmore, Galway, H91 R673, Ireland
Email: adam.leadbetter@marine.ie

Adam Shepherd
Biological and Chemical Oceanography Data Management Office
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 96678-2391
Email: ashepherd@whoi.edu

***Abstract*—In August 2015, a new seafloor observatory was deployed in Galway Bay, Ireland. The sensors on the observatory platform are connected by fibre-optic cable to a shore station, where a broadband connection allows data transfer to the Marine Institute's data centre. This setup involved the development of a new data acquisition system which takes advantage of open source streaming data solutions developed in response to the Big Data paradigm, in particular the Velocity aspect. This activity merges concepts from the arenas of both Big Data and Internet of Things where data standardisation is not normally considered. This paper considers the architecture implemented to stream marine data from instrument to end user and offers suggestions on how to standardise these data streams.**

***Keywords*—Linked Data, Big Data, streaming data, oceanographic data management, Earth Science Informatics.**

I. Introduction

The traditional oceanographic data management paradigm of small-volume, complex datasets heavily annotated in a delayed-mode has been challenged in recent years through the proliferation of new instrument platforms and communications models. An oceanographic research expedition or monitoring programme often relied on the availability of a research vessel and the deployment of discrete sampling sensors from that platform at a number of stations, and the data were made available only after the return to shore and extensive processing of the results. The development of the Argo float network, the global fleet of Autonomous Underwater Vehicles and networks of seafloor observatories has led to a growth in ocean research and observation [1]. This has led to a move into the Big Data paradigm for oceanographic data managers. In particular, as marine data is more and more required to be accessible in real-time, or near-real-time, to aid decision making processes the "Velocity" component of the Big Data paradigm has emerged as being of particular importance.

The connection of these platforms and other oceanographic instrumentation to a range of communications networks which allows the observation data to be published to the World Wide Web in real-time also means that there is an opportunity to align oceanographic data management practices with the emerging Internet of Things paradigm.

This paper introduces achievements made in connecting the Big Data and Internet of Things models with established oceanographic data management practices to allow data to stream from instrument to Web, arriving at the final destination in a structured, connected format.

II. Big Data "Velocity" for Ocean Observations

The common definition of Big Data incorporates the characteristics of [2]:

- **Volume** The quantity of generated and stored data
- **Variety** The type and nature of the data
- **Velocity** The speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development
- **Variability** Inconsistency of the data set can hamper processes to handle and manage it
- **Veracity** The quality of captured data can vary greatly, affecting accurate analysis

Within the Earth Sciences domain, the problem of Volume has been addressed by projects such as eReefs [3] which provides both a Semantic Web ontology [4], [5] to describe the datasets available to the system and a brokering layer to bring those datasets into common processing and display tools. Many projects have looked to provide a variety of oceanographic information through single points, such as the SeaDataNet [6] project in Europe and the Biological and Chemical Oceanography-Data Management Office [7] project in the United States. A common theme of the approach to addressing the issue of variety in these projects, as with eReefs, is to use Semantic Web techniques (in particular well-managed controlled vocabularies published online) to ensure the data descriptions are interoperable. SeaDataNet has also included controlled vocabularies to describe the veracity of data which have been quality controlled in *post hoc* procedures by national oceanographic data centres. The contents of these vocabularies have been incorporated by the Intergovernmental Oceanographic Commission's (IOC) International Oceanographic Data and Information Exchange and the Joint IOC-World Meteorological Organisation Technical Commission for Oceanography and Marine Meteorology (JCOMM) into the Ocean Data Standards process [8]. However, the problem of the velocity of ocean data has been historically restricted to, at

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

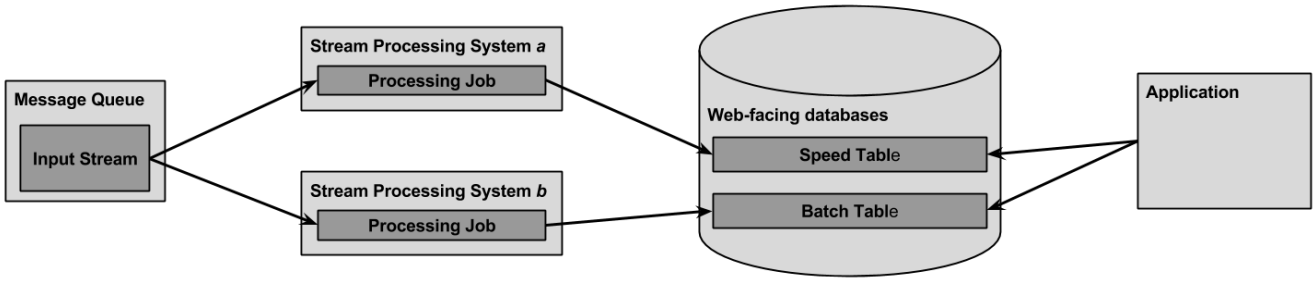

Fig. 1. The lambda-architecture

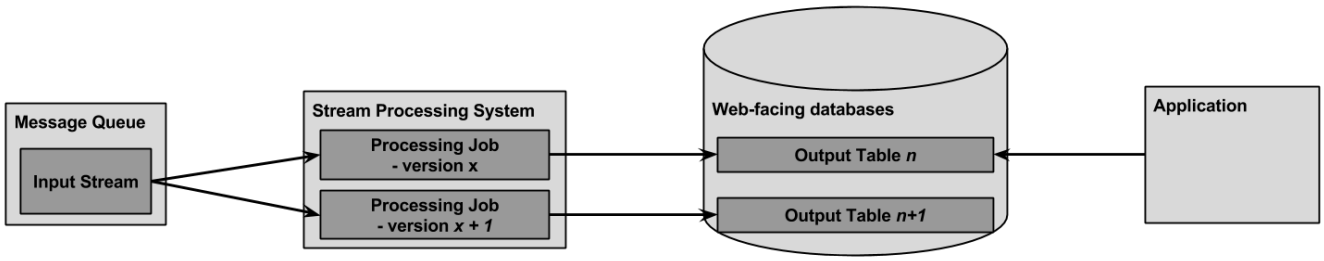

Fig. 2. The kappa-architecture

best, near-real time applications due to the constraints placed on systems by the harshness of the operating environment. The installation of a communications and power cable in Galway Bay for research into ocean energy development has provided an opportunity to address some of the issues of streaming data in real-time, with at the slowest a 2-second delay, from the seabed to the World Wide Web, which may have applications in future deployments of more power- and bandwidth-limited situations.

A. Architectural Considerations

As the demand for data to be delivered in real-time from a range of internet applications, new architectures for software processing such as the lambda- and kappa-architectures, have been developed. In lambda-architecture, an immutable sequence of records is captured and fed into a batch system and a stream processing system in parallel (see Figure 1). The transformation logic is, however, implemented twice, once in the batch system and once in the stream processing system. Results from both systems are stitched together at query time to produce a complete answer [9]. However, particularly in ocean science scenarios, the desired workflow is to process the data in some rapid manner as close to the time of collection as possible using the *a priori* knowledge of the dataset, and then re-process once the *post hoc* knowledge base is increased. This fits well within the basis of the kappa-architecture, which was proposed by [10] as an alternative to the lambda-architecture (Figure 2). The lambda architecture concentrates on the ability to reprocess the full data stream at a later date through storing the full data in a message queue which allows for multiple subscribers (for example, Apache Kafka, http://kafka.apache.org/). When reprocessing is required, new processing job code is introduced to the stream processing system which can be run against the entire message queue to generate an n+1 version of the output.

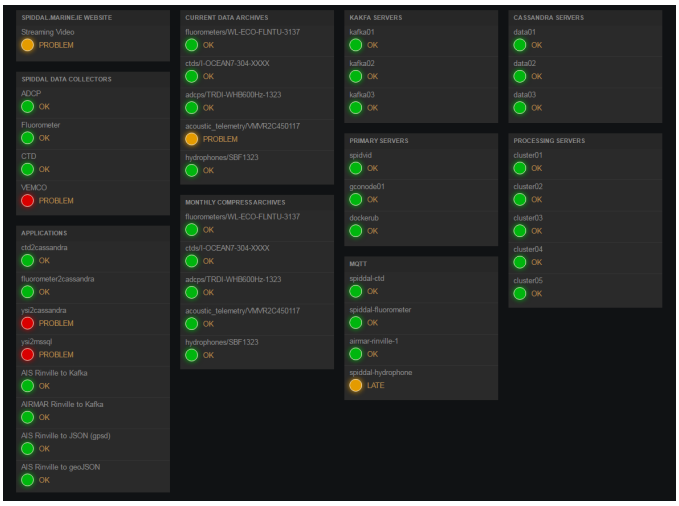

Fig. 3. The streaming data system monitoring dashboard enabled by microservices with exposed HTTP interfaces.

B. Implementation

When implementing a streaming data system to deliver the Galway Bay cable observatory data to end users in real-time, the following considerations guided the architectural design.

The instruments deployed at the Galway Bay cable observatory are connected for both electrical power consumption and data delivery by a combination of a fibre-optic and copper cable which runs around 1600 metres off shore. The data are exposed from the cable termination equipment in a shore

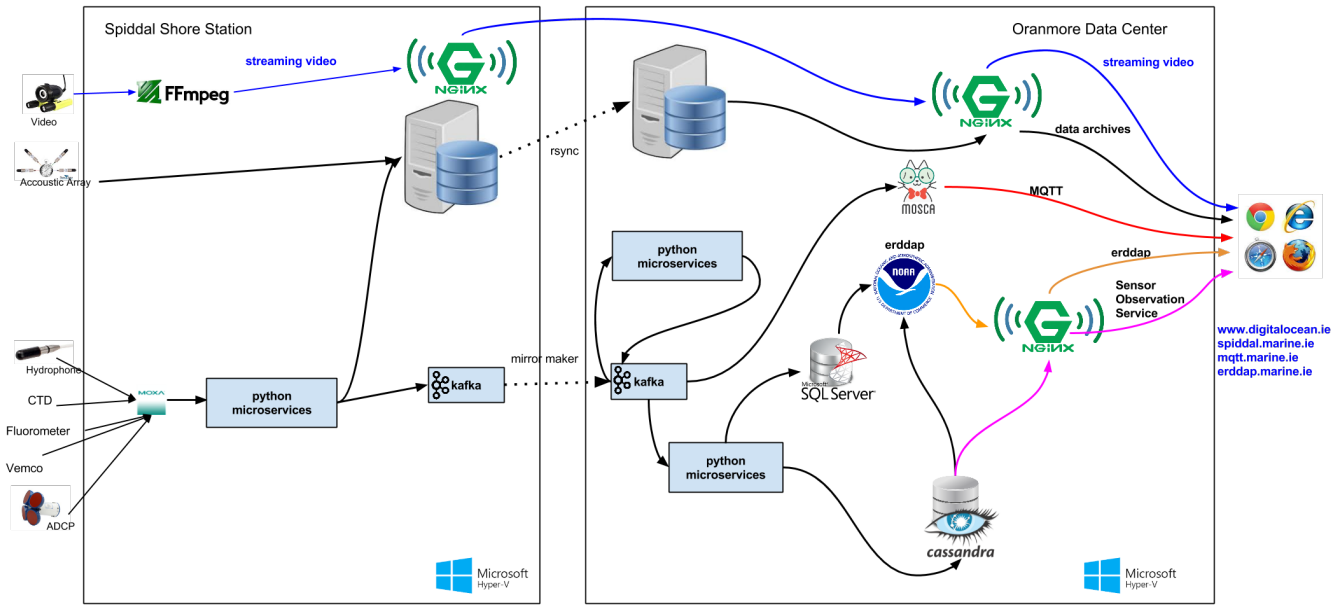

Fig. 4. The streaming data system monitoring dashboard enabled by microservices with exposed HTTP interfaces.

station to a corporate network over TCP/IP via a serial device server. In order to receive, process, archive and publish the data the microservices [11] approach has been adopted, meaning that at each stage of the data flow, a small piece of code performing one task takes a data input and does one job with it (e.g. pushing it to an Apache Kafka messaging queue; writing from a queue to disc). It is worth noting that a microservice pushing the data to a message queue will not do any other processing on those data, including archiving them elsewhere. Each of these microservices has been written to expose a hypertext transfer protocol interface which has allowed the production of simple dashboards for monitoring the health of the data system (see Figure 3).

The first layer of microservices encountered acquire the data feeds from the serial to Ethernet server and pushes them, raw with some additional metadata comprising a Global Positioning System timestamp and a unique identifier for the instrument, to an Apache Kafka message queue. This message queue supports a two-week archive of the data from the observatory in case of a loss of connectivity between the shore station and the data centre. The complete Kafka message queue, consisting of one topic per instrument type, is mirrored from the shore station to the data centre using the Kafka MirrorMaker. At the instrument interface, microservices also write the raw output to disc, which is then mirrored to the data centre using rsync. This job also removes the files from the shore station on success. This approach ensures that the data can be retrieved even with a network outage and also cleans up disc space in the shore station.

Once received in the data centre, the mirrored Kafka message queue is subscribed to by a number of microservices. Some perform immediate data cleansing, enrichment (such as standard unit conversions), or filtering whilst others may perform data aggregation. However, as there is a queue buffer in the shore station and a disc archive of the raw data, these processing tasks may be re-run at any stage in line with the principals of the kappa architecture. These microservices push their results to new Kafka topics, and from these data may be moved to a permanent storage location in either a traditional SQL database or in an Apache Cassandra data store. These data stores are exposed in a semi-RESTful way to end users via an instance of the ERDDAP data server[12], which sits behind an NGINX web server. The NGINX configuration has been extended to provide access to the Apache Cassandra data store via a simplified OGC Sensor Observation Service (see Section IV-B below). Ultimately, the data files, the ERDDAP server and the Sensor Observation Service all allow users batch access to the data archives. Further, the latest messages in the Apache Kafka message queues have been made available via MQTT, the Internet of Things data access standard, through the Mosca MQTT broker set in read-only mode.

Each of the microservices has been constructed to exit if it receives no data in a given time period. This frees up the TCP/IP connections made to the data streams, but there remains the issue of what to do with the microservice once it has exited. Manual intervention on this is not desirable as it requires a constantly on call member of staff, and is also not scalable. To this end, the microservices are monitored by the supervisord process control system, and the microservices are restarted by supervisord after exit. This paradigm of ”exit the process if no data has been received within the configurable time”, together with ”restart the process if it has exited”, has proven a robust system requiring little manual intervention.

The video stream is captured to disc in one-minute chunks via a video capture card. A Python job monitoring the GPS time from the shore station’s writes the time to disc and this is used as an overlay on the video stream via FFMPEG. Internally, the video is available over the User Datagram Protocol which allows real-time monitoring of the video. The video stream is delivered over the web to users via HTTP Live

```
2016-08-18T11:42:59.782Z|I-OCEAN7-304-XXXX|  24.34  16.118  37.630  29.376 1504.0682 11:45:22.86M
2016-08-18T11:45:05.628Z|WL-ECO-FLNTU-3137|08/18/16 11:43:37  695 47  700 120 539
```

Listing 1. Raw outputs from oceanographic instrumentation - line 1: a conductivity-temperature-depth sensor; line 2: a fluorometer.

```
#Grok patterns for Idronaut instruments
IDRONAUT_OCEAN7_304 %{NUMBER:pressure:float}%{SPACE}%{NUMBER:temperature:float}%{NUMBER:
    conductivity:float}%{SPACE}%{NUMBER:salinity:float}%{SPACE}%{NUMBER:sound_velocity:float}%{
    SPACE}%{TIME:raw_time}%{SPACE}
#Grok patterns for Wetlabs instrument outputs
WETLABS_ECO_FLNTU_3137 %{MONTHNUM}\/%{MONTHDAY}\/%{YEAR}%{SPACE}%{TIME}%{SPACE}%{NUMBER:
    wavelength_fluorescence}%{SPACE}%{INT:chlorophyll_counts}%{SPACE}%{NUMBER:
    wavelength_turbidity}%{SPACE}%{INT:turbidity_counts}%{SPACE}%{NUMBER:thermistor}
#Grok patterns for the prefix added by the Marine Institute Spiddal Observatory
MI_INST [a-zA-Z0-9._-]+
MI_PREFIX %{TIMESTAMP_ISO8601:timestamp}\|%{MI_INST:instrument}\|
```

Listing 2. Grok patterns to parse the raw data outputs of Listing 1.

```
{
  "MI_PREFIX": [["2016-08-18T11:42:59.782Z|I-OCEAN7-304-XXXX|"]],
  "TIMESTAMP_ISO8601": [["2016-08-18T11:42:59.782Z"]],
  "YEAR": [[2016]],
  "MONTHNUM": [[8]],
  "MONTHDAY": [[18]],
  "HOUR": [[11,null,11]],
  "MINUTE": [[42,null,45]],
  "SECOND": [[59.782,22.86]],
  "ISO8601_TIMEZONE": [["Z"]],
  "MI_INST": [["I-OCEAN7-304-XXXX"]],
  "SPACE": [["  ","  ","  ","  ","  "," "," "]],
  "IDRONAUT_OCEAN7_304": [["24.34  16.118  37.630  29.376 1504.0682 11:45:22.86"]],
  "pressure": [[24.34]],
  "BASE10NUM": [[24.34,16.118,37.630,29.376,1504.0682]],
  "temperature": [[16.118]],
  "conductivity": [[37.630]],
  "salinity": [[29.376]],
  "sound_velocity": [[1504.0682]],
  "TIME": [["11:45:22.86"]]
}
```

Listing 3. Output of the grok pattern %{MI_PREFIX}%{SPACE}%{IDRONAUT_OCEAN7_304} as defined in Listing 2 and run against line 1 of Listing 1

Streaming.

Where possible, automation techniques have been employed. For example, a Rundeck [1] instance is operating with access to the data system architecture and has been successfully employed in configuration of instrumentation attached to the subsea observatory.

The complete architecture is shown in overview in Figure 4.

III. Linked Data and the Semantic Web in Ocean Sciences

Much effort has been put into structured descriptions of oceanographic data in order to allow for machine-to-machine interoperability of datasets [13]. The focus of this effort has been around the Semantic Web [4] and Linked Data [14] standards introduced by the World Wide Web Commission. This work has broadly encompassed four areas, namely:

- Standardising vocabulary
- Faceted browsing of data
- Enhanced data discovery
- Machine-to-machine interoperability

The standardised vocabularies for oceanographic data began with the code tables of the GF3 data format [15], and have migrated to full Semantic Web documents, published as serialisations of the Resource Description Framework, such as on the NERC Vocabulary Server [16], the Marine Metadata Interoperability Ontology Register and Repository [17], or the Australian SISSVoc server [18].

Faceted browsing of data has been enabled by adding hierarchical information to these controlled vocabularies, allowing data clients such as those deployed by the SeaDataNet project [6] to drill down through "Physical oceanography" through to "Currents" and then to "Horizontal flow in the water column" to discover datasets which relate to latitudinal or longitudinal current speeds in the marine environment. The

[1] http://rundeck.org/

NETMAR project extended this approach beyond the observed environmental property to include the measuring instrument or sensor, the platform from which an observation was made, the project responsible for funding an observing campaign and the vertical zone in which the measurement was made [16]. Enhanced data discovery, such as "fuzzy" search, has been explored in some of these applications, the traditional example being that a search for the term "precipitation" would yield results for "rainfall" and "snow" data, but this has been only truly explored as an extension of the faceted search use case until now as the horizontal connections between facets requires the development of a deeper Knowledge Organisation System than has been traditionally used in the ocean sciences.

The publication of many of the resources described above in World Wide Web Consortium standards has been a first step towards machine-to-machine interoperability. To further this, the EarthCube cyber-infrastructure programme in the United States has funded the GeoLink project[2] which brings together experts from the geosciences, computer science, and library science in an effort to develop Semantic Web components that support discovery and reuse of data and knowledge. GeoLink's participating repositories include content from field expeditions, laboratory analyses, journal publications, conference presentations, theses/reports, and funding awards that span scientific studies from marine geology to marine ecosystems and biogeochemistry to paleoclimatology. GeoLink is building a set of reusable ontology design patterns (ODPs) that describe core geoscience concepts, a network of Linked Data published by participating repositories using those ODPs, and tools to facilitate discovery of related content in multiple repositories.

IV. Streaming Linked Data observations

In order to make Earth Science data collected from a range of sources interoperable, there have been many efforts to create a standard model for environmental observations and measurements. An information model agreed by both the International Standards Organisation (ISO) and Open Geospatial Consortium (OGC) for observations, measurements and physical samples, called Observations and Measurements (O&M) has been defined [19], [20], [21]. As part of the OGC's effort in Sensor Web Enablement, O&M has been extensively used as the model in XML serialisations of sensor observations on the World Wide Web. However, XML has fallen out of favour in current web development practices [22] and the JavaScript Object Notation (JSON) serialization of data is preferred. This has led to the introduction of JSON representations of the O&M information model [23] (OM-JSON), and Sensor Observations Services offering the JSON representation of O&M as their payload. The emergence of both a Linked Data model for JSON (JSON-LD, [24]) and a Semantic Web representation of O&M [25] allows for a merger between the streaming data paradigm and the Linked Data paradigm described above. This merger is explored in further detail below.

A. Beyond Plain Old MQTT

The first challenge in standardising the output of various MQTT feeds, with a range of instrument manufacturers responsible for the raw data streams, is to provide an extensible system for harmonisation of the heterogeneous raw data inputs to homogeneous outputs.

One popular tool for systems administrators to harmonise log files from various services is Logstash[3]. Logstash has been designed to centralise data processing of all types, and to normalise data from various schema and formats. If we consider the raw instrument output from, for example, conductivity-temperature-depth sensor or fluorometers to be log files, then Logstash is a promising method for processing those logs to normalised schema. Logstash has access to a large number of inputs (including MQTT), filters for processing data, and outputs (again, including MQTT). Of the filters, the grok filter[4] is designed to parse arbitrary text to a structure. Listing 1 illustrates the relatively structured nature of the data emitted from the oceanographic instruments, but *a priori* knowledge of the fields is required in order to parse the data. This *a priori* knowledge can be encoded as a grok pattern, which is an enhanced regular expression which decodes the unstructured data into a JSON object. Custom grok patterns can be registered into the configuration of a Logstash instance. Examples of these grok patterns are shown in Listing 2 and may also be accessed online[5].

This method is not reliant on the monolithic Logstash application, as the grok filter is available as a module for a number of programming languages, including Python[6] and JavaScript[7]. However, whichever implementation of grok is used, further manipulation of the resulting-JSON object (Listing 3) is required in order to produce a recognised standard data format. Logstash also provides a "mutate" filter which can be used to perform this kind of manipulation, and an lightweight-JavaScript implementation of analogous functionality also exists [8]. Using a JavaScript mutate pattern such as that in Listing 4 over the JSON object in Listing 3 yields a full OM-JSON document (Listing 5).

OM-JSON is the JSON implementation of the O&M data model. O&M defines an observation as an action whose result is an estimate of the value of some property of the feature-of-interest within a discrete time instant or period obtained using a specified procedure. The key notions are the feature-of-interest (the geographic point 53.233,-9.285 off the coast of Spiddal, Ireland), the property being measured (temperature), the estimated value (16.118), its units of measure (degrees Celsius), the temporal element (2016-08-18T11:42:59.782Z), and the specified procedure (the use of a CTD rosette). Requirements of the OM-JSON specification dictate how this information is to be encoded. For example, Listing 6 demonstrates how time instants must be declared - as a JSON object with a single property called "instant" whose character string value matches one of the following XML Schema Definition (XSD)[9] data types: dateTime, date, gYearMonth, gYear.

These requirements are formalised in the OM-JSON schema, a set of JSON Schema [26] documents which define

[2]http://www.geolink.org/

[3]https://www.elastic.co/products/logstash

[4]https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html

[5]https://github.com/IrishMarineInstitute/grok-raw-inst/tree/master/patterns

[6]https://github.com/garyelephant/pygrok

[7]https://github.com/phillip/node-grok

[8]https://github.com/adamml/mutate.js

[9]http://www.w3.org/TR/xmlschema11-2/

```
{
  "id": "foo",
  "phenomenonTime": {"instant": "\$.TIMESTAMP_ISO8601[0][0]"},
  "featureOfInterest":
    {"href": "http://linked.marine.ie/feature/exampleURI"},
  "member": [
    {
      "id": "#temperature",
      "type": "Measurement",
      "result":
        {
          "value": "\$.temperature[0][0]",
          "uom": "http://vocab.nerc.ac.uk/collection/P06/current/UPAA/"
        },
      "observedProperty":
        {"href":"http://vocab.nerc.ac.uk/collection/P01/current/TEMPPR01/"},
      "procedure":
        {"href":"http://vocab.nerc.ac.uk/collection/L22/current/TOOL0861/"},
      "resultTime": "\$.TIMESTAMP_ISO8601[0][0]"
    }
  ]
}
```

Listing 4. A javascript template, for use with the mutate.js package, which can be used with the data in Listing 1 to create an OM-JSON document

```
{
    "id":"foo",
    "phenomenonTime":{"instant":"2016-08-18T11:42:59.782Z"},
    "featureOfInterest":{"href":"http://linked.marine.ie/feature/exampleURI"},
    "member":[
        {
            "id":"#temperature",
            "type":"Measurement",
            "result":
            {
                "value":16.118,
                "uom":"http://vocab.nerc.ac.uk/collection/P06/current/UPAA/"
            },
            "observedProperty":{
                "href":"http://vocab.nerc.ac.uk/collection/P01/current/TEMPPR01/"
            },
            "procedure":{
                "href":"http://vocab.nerc.ac.uk/collection/L22/current/TOOL0861/"
            },
            "resultTime":"2016-08-18T11:42:59.782Z"
        }
    ]
}
```

Listing 5. The resulting OM-JSON from taking the JSON structure at Listing 3 through the mutate.js package using the template in Listing 4

the structure of valid OM-JSON data. With a JSON data document and the OM-JSON schema, documents can be validated as proper representations of the OM data model. Other requirements such as how to define estimated values as seen in Listing 7 state that units of measure shall either be a symbol from the Unified Code for Units of Measure [27] or a URI denoting a unit-of-measure defined in a web resource.

A full example of a valid OM-JSON document can be seen in Listing 5. Now, by allowing for URIs to denote meaning, OM-JSON schema employs Linked Data principles for data interoperability and disambiguation. Taking the OM-JSON from Listing 5, one can resolve the URIs to understand the meaning of a particular datum, and a machine, trying to integrate OM-JSON documents can perform co-reference resolution techniques with high precision. Naturally, the next progression is to enhance OM-JSON schema to validate JSON-LD documents, or as we will discuss later, OM-JSON-LD.

B. *OM-JSON as a Sensor Observation Service*

The OGC's Sensor Observation Service (SOS) is a web service to query real-time sensor data and sensor data time series and is part of the Sensor Web. The sensor data offered by an SOS implementation comprises descriptions of sensors themselves, which are encoded in the Sensor Model Language (SensorML), and the measured values in an O&M serialisation. Taking this definition of the SOS, the payload may be modelled

as OM-JSON. To this end, a simple SOS has been developed which delivers an OM-JSON document in response to the standard OGC GetObservation call [10]. This SOS implementation has been added to the architecture described above as a series of configuration statements in the NGINX web server which call one of a number Lua scripts [11] which query the Cassandra data store to return the appropriate values. The GetCapabilities and DescribeSensor operations of the SOS retain their XML payloads as per the standard OGC specification. Future work would complete functionality of this SOS implementation to fully meet the OGC specification.

V. Towards OM-JSON-LD

JSON-LD is designed around the concept of a "context" to provide additional mappings from JSON to the Resource Description Framework (RDF)[12] model. The context links object properties in a JSON document to concepts in an ontology. As an ontology for O&M already exists in OWL [25] it is possible to do this for OM-JSON by inserting URIs from the O&M ontology into the OM-JSON context. Because JSON-LD has been designed to be minimally invasive to pre-existing JSON documents, there is only one necessary modification to the OM-JSON schema for it to accept JSON-LD documents as valid OM-JSON. JSON-LD's '@context' element is where terms are mapped to IRIs, or better yet, dereferenceable URIs. Tim Berners-Lee's notion of 5-Star Linked Data[13] is formalized by adding JSON-LD's '@context' element to the OM-JSON schema. This simple extension allows for an OM-JSON document to be translated into other RDF representations, and therefore, enabling interoperability with other RDF documents. Without it, OM-JSON could still be employing 5-Star Linked Data as the URIs may be linking to RDF, but this requirement is not enforced, and for good reason. The initial motivation for the development of OM-JSON was purely focused on the encoding of Observations and Measurements records, and not the encoding of Linked Data representations of those records. However, by adding JSON-LD compliance, OM-JSON implementers decide in the '@context' definition, which elements of the OM-JSON document are RDF compliant. Or, implementers can ignore completely the '@context' element and still produce valid OM-JSON documents. This extension of the OM-JSON schema is purely an enhancement to the implementation that does not affect pre-existing OM-JSON documents, nor does it alter the goals or intent of the O&M implementation. One strategy for creating an OM-JSON context is given in Listing 8.

The modification to the OM-JSON schema is given in a fork of the master OM-JSON GitHub repository[14].

The JSON-LD '@context' given in Listing 8 can be added to the OM-JSON document given in Listing 5 to produce on OM-JSON-LD document. The JSON-LD context developed here gives rise to the RDF triples as shown in Listing 9.

[10]https://github.com/IrishMarineInstitute/spiddal.marine.ie

[11]https://github.com/IrishMarineInstitute/spiddal.marine.ie/tree/master/lua

[12]https://www.w3.org/RDF/

[13]http://5stardata.info/en/

[14]https://github.com/adamml/om-json

VI. "Born Connected"

The idea that data may flow from the oceanographic instrument to the Web fully semantically annotated has gained ground in recent years [28], [13]. The major hurdles in advancing the "Born Connected" approach have been in getting involvement from instrument manufacturers in the process of assigning URLs to parameters to create Linked Data from the point of collection and in creating an extensible approach to implementing "Born Connected" ideals. The approach explored in this paper delegates some of the responsibility for developing the Linked Data representations of the instrument output as any responsible party can register the grok patterns and jq filters for publishing Linked Data in this process. Indeed, this is the key progress in "Born Connected" systems which this paper represents: an extensible system, with components that can be registered, for the structuring of raw instrument outputs first as plain-old JSON and thereafter as OM-JSON. The weakness is that this approach retains the "fog-computing" [29] approach of earlier "Born Connected" approaches, and in the long-term it would be ideal to move the composition of the OM-JSON documents closer to the devices.

VII. Future Work

As yet, the patterns for harmonising raw data outputs to JSON do not deal with instruments which either output multi-line data formats or binary data formats. The former may be tackled through grok's ability to handle multi-line data, but this has not been proven. However, a strategy for dealing with binary logfiles must be properly developed. This could involve a microservice emitting a text-based representation of the binary output which can be polled by either Logstash or a second microservice written in a language with bindings to grok.

The JSON-LD context for OM-JSON remains incomplete, and there is some work to be done, potentially including recommending updates to the JSON-LD spec, to allow a full translation of OM-JSON to Linked Data in this manner. Continuing the argument of Section VI, another area of future work would be the implementation of grok and jq on power-limited devices for deployment in data loggers which could herald the movement of this approach to structuring data closer to the instruments themselves.

Another area in which initial investigations have been undertaken, but more work is required, is in indexing the data for textual and spatio-temporal search. Elasticsearch provides some support for these features, and its capabilities have been demonstrated in initial tests but not yet put into production.

While built around a distributed architecture, the initial implementation of this data architecture was delivered within a single hypervisor. Recent work has separated the Cassandra cluster from this hypervisor, and future work will extract the Kafka cluster and Erddap to their own hypervisors to benefit from the high availability principles of the architecture's components.

VIII. Conclusion

In conclusion, this paper has demonstrated that it is possible to make use of the emerging Big Data "Velocity"

```
{
      "instant": "2016-08-18T11:42:59.782+01:00"
}
```

Listing 6. OM-JSON encoding of a time instant as specified by the associated JSON Schema document

```
{
    "value": 16.118,
    "uom": "http://vocab.nerc.ac.uk/collection/P06/current/UPAA/"
}
```

Listing 7. OM-JSON encoding of the value estimated by a measurement observation, showing the unit of measure defined by a remote web resource

paradigm which has enabled the real-time sharing of sub-sea environmental observations via the World Wide Web. The architecture shown in this paper enables the use of both Internet of Things standards, via the exposure of an MQTT interface to the data, and also extends the streaming data paradigm to produce output which is domain standards compliant. This latter activity has extended the "Born Connected" idea which has proposed that environmental data be collected as Linked Data from as close to their point of origin as possible. The approach taken in this paper has begun to pave the way for an extensible approach to "Born Connected" activity in the marine sciences domain, which was not the case in past implementations of solutions to this problem. As shown above, there is still work to do, but the groundwork laid out here will allow further development of these ideas and may inform future answers and greater interoperability of environmental data from the point of collection.

On issue which remains is the migration of the presented data system from a DevOps to a truly operational environment. This is being promoted through a migration of the low-level Python based microservices to the StreamSets[15] platform. StreamSets allows for a visual composition of the streaming data chain which lowers the barrier to adoption. The code for the system has been made available on GitHub for re-use[16], and this could further be promoted by providing Docker images for the virtual machines which make up the distributed architecture. This would move the system closer to a "puch button" install model.

Acknowledgements

The authors would like to thank the Science Foundation of Ireland and the Sustainable Energy Authority of Ireland for their funding used to install the subsea observatory infrastructure in Galway Bay, against which the work in this paper was developed. HEAnet (Ireland's National Research and Education Network) have also provided valuable infrastructure resources.

References

[1] P. J. Durack, T. Lee, N. T. Vinogradova, and D. Stammer, "Keeping the lights on for global ocean salinity observation," *Nature Climate Change*, vol. 6, no. 3, pp. 228–231, 2016.

[2] M. Hilbert, "Big data for development: a review of promises and challenges," *Development Policy Review*, vol. 34, no. 1, pp. 135–174, 2016.

[15] https://streamsets.com/

[16] https://github.com/IrishMarineInstitute/uwobs

[3] J. Yu, B. Leighton, N. Car, S. Seaton, and J. Hodge, "The eReefs data brokering layer for hydrological and environmental data," *Journal of Hydroinformatics*, vol. 18, no. 2, pp. 152–167, 2016.

[4] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The Semantic Web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[5] D. L. McGuinness, "Ontologies come of age," in *Spinning the Semantic Web: bringing the World Wide Web to its full potential*, D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, Eds. Cambridge, MA: MIT Press, 2003, ch. 6, pp. 171–195.

[6] D. M. Schaap and R. K. Lowry, "SeaDataNet–Pan-European infrastructure for marine and ocean data management: unified access to distributed data sets," *International Journal of Digital Earth*, vol. 3, no. S1, pp. 50–69, 2010.

[7] M. Allison, C. Chandler, R. Groman, P. Wiebe, D. Glover, and S. Gegg, "The Biological and Chemical Oceanography Data Management Office," in *American Geophysical Union Fall Meeting Abstracts*, vol. 1, 2011, p. 1602.

[8] *Ocean Data Standards, Vol.3: Recommendation for a Quality Flag Scheme for the Exchange of Oceanographic and Marine Meteorological Data*, Intergovernmental Oceanographic Commission of UNESCO, Paris, 2013.

[9] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable realtime data systems*. Greenwich, CT: Manning Publications, 2015.

[10] J. Kreps. (2014) Questioning the Lambda Architecture. [Online]. Available: https://www.oreilly.com/ideas/questioning-the-lambda-architecture

[11] D. Namiot and M. Sneps-Sneppe, "On micro-services architecture," *International Journal of Open Information Technologies*, vol. 2, no. 9, 2014.

[12] R. Mendelssohn and R. Simons, "ERDDAP-an easier way for diverse clients to access scientific data from diverse sources," in *AGU Fall Meeting Abstracts*, vol. 1, 2008, p. 09.

[13] A. M. Leadbetter, "Linked Ocean Data," in *The Semantic Web in Earth and Space Science. Current Status and Future Directions*, T. Narock and P. Fox, Eds. Amsterdam: IOS Press, 2015, ch. 2, pp. 11–31.

[14] T. Berners-Lee. (2009) Linked Data design issues. [Online]. Available: https://www.w3.org/DesignIssues/LinkedData.html

[15] Intergovernmental Oceanographic Commission, "GF3: a general formatting system for geo-referenced data. Vol. 2. Technical description of the GF3 format and code tables." UNESCO, Tech. Rep., 1987.

[16] A. M. Leadbetter, R. K. Lowry, and D. O. Clements, "Putting meaning into netmar–the open service network for marine environmental data," *International Journal of Digital Earth*, vol. 7, no. 10, pp. 811–828, 2014.

[17] J. Graybeal, A. W. Isenor, and C. Rueda, "Semantic mediation of vocabularies for ocean observing systems," *Computers & Geosciences*, vol. 40, pp. 120–131, 2012.

[18] S. J. Cox, J. Yu, and T. Rankine, "SISSVoc: A Linked Data API for access to SKOS vocabularies," *Semantic Web*, vol. 7, no. 1, pp. 9–24, 2014.

[19] S. J. D. Cox, "Observations and Measurements Part 1 Observation schema," Open Geospatial Consortium, Tech. Rep.

```
"@context": {
    "observedProperty": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
        observedProperty",
    "href": "@id",
    "featureOfInterest": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
        featureOfInterest",
    "id": "@id",
    "phenomenonTime": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
        phenomenonTime",
    "instant": "@value",
    "member": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#relatedObservation"
        ,
    "procedure": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#Process",
    "resultTime": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#resultTime",
    "result": "http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#result"
}
```

Listing 8. One approach for creating a truly Linked Data Sensor Observation Service involves the output in Listing 5 overlain with a JSON-LD Context document of the type shown here.

```
<http://bar.baz/foo#temperature> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    Process> <http://vocab.nerc.ac.uk/collection/L22/current/TOOL0861/> .
<http://bar.baz/foo#temperature> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    observedProperty> <http://vocab.nerc.ac.uk/collection/P01/current/TEMPPR01/> .
<http://bar.baz/foo#temperature> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    result> _:c14n0 .
<http://bar.baz/foo#temperature> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    resultTime> "2016-08-18T11:42:59.782Z" .
<http://bar.baz/foo> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    featureOfInterest> <http://linked.marine.ie/feature/exampleURI> .
<http://bar.baz/foo> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    phenomenonTime> "2016-08-18T11:42:59.782Z" .
<http://bar.baz/foo> <http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation#
    relatedObservation> <http://bar.baz/foo#temperature> .
```

Listing 9. Resource Description Framework triples generated from applying the JSON-LD context document in Listing 8 to the OM-JSON document in Listing 5

Implementation Standard 07-022r1, 2007. [Online]. Available: http://portal.opengeospatial.org/files/22466

[20] ——, "Observations and Measurements Part 2 Sampling features," Open Geospatial Consortium, Tech. Rep. Implementation Standard 07-002r3, 2007. [Online]. Available: http://portal.opengeospatial.org/files/22467

[21] *ISO 19156: Geographic Information Observations and Measurements*, International Organization for Standardization, Geneva, 2011. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm\\?csnumber=32574

[22] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of JSON and XML data interchange formats: A case study," in *Computer Applications in Industry and Engineering*, 2009, pp. 157–162.

[23] S. J. D. Cox and P. A. Taylor, "OGC Observations and Measurements JSON implementation," Open Geospatial Consortium, Tech. Rep. Discussion Paper 15-100r1, 2015. [Online]. Available: http://www.opengis.net/doc/dp/om-json/

[24] M. Lanthaler and C. Gütl, "On using JSON-LD to create evolvable RESTful services," in *Proceedings of the Third International Workshop on RESTful Design*. Association for Computing Machinery, 2012, pp. 25–32.

[25] S. J. D. Cox, "An explicit OWL representation of ISO/OGC Observations and Measurements," in *Proceedings of the 6th International Conference on Semantic Sensor Networks*. CEUR-WS.org, 2013, pp. 1–18.

[26] F. Galiegue and K. Zyp, "JSON Schema: Core definitions and terminology," *Internet Engineering Task Force (IETF)*, 2013.

[27] G. Schadow and C. J. McDonald, "The unified code for units of measure," *Regenstrief Institute and UCUM Organization: Indianapolis, IN, USA*, 2009.

[28] J. Fredericks, "Persistence of knowledge across layered architectures," in *Collaborative Knowledge in Scientific Research Networks*, P. Diviacco, P. Fox, C. Pshenichny, and A. Leadbetter, Eds. Hershey, PA: IGI Global, 2015, ch. 13, pp. 262–282.

[29] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.