

Machine Readable Metadata for Argo Platforms and Sensors

Eric Rehm (SBS), Brian King (NOC), Jean-Michel Lecount
(RBR Global)

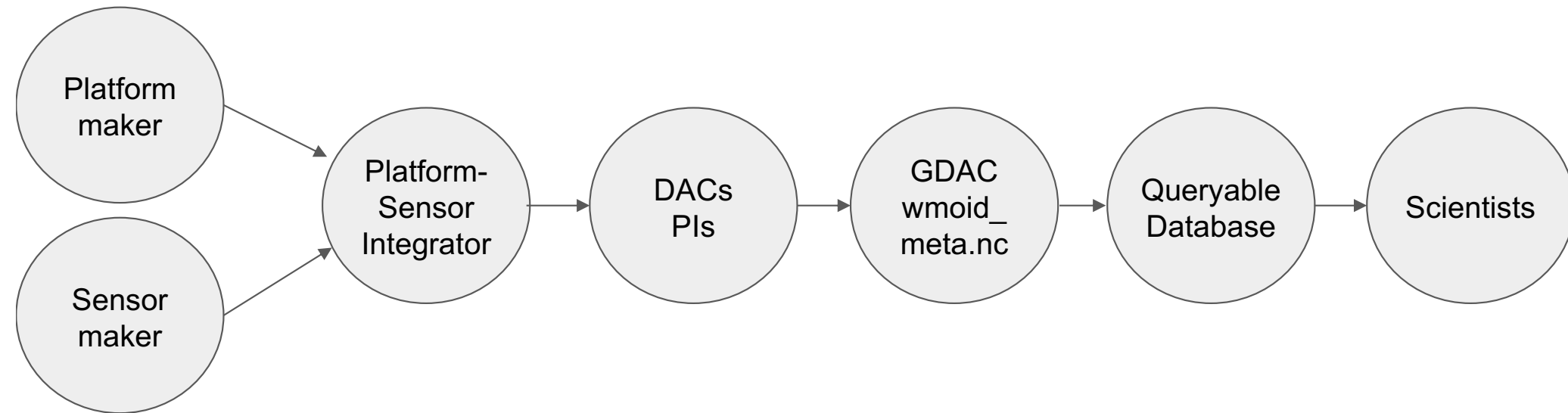
History

- ADMT-21: December 2020
 - RBR demonstrates schema-controlled JSON metadata for RBRargo³ CTDs via [RESTful API](#)
 - Recently updated to follow JSON Schema presented herein.
- ADMT-22, ADMT-23
 - Brian King (NOC) and Eric Rehm (SBS) work on vendor-extensible sensor, platform, and float metadata schemas and example instances
 - SBS demonstrates schema-compliant JSON metadata for SEAFET pH sensors
 - Preliminary feedback on example JSON instances from NOC, WHOI, MBARI
- ADMT-24, OS 2024
 - Email sent out prior to ADMT-24 to ARGO-DM list
 - Today's presentation
 - New schema based on ARGO User's Manual, vendor-controlled schema extensions, added context for validation of NERC Vocabulary Server (NVS) controlled terms
 - Demonstration of schema-based validation and validation of controlled terms using NVS RESTful API
 - Schema, examples, and code on GitHub: [euroargodev/json-sensor-metadata](#)
 - SBS, NOC, RBR and Tini Scientific submitted abstract to OS 2024

Metadata Users

Argo JSON metadata flow from sensor manufacturer and platform vendor to “users”.

- DAC creating a *wmoid_meta.nc* file,
- DACs and PIs who processing data from instrument units to ocean measurements) where correct metadata is crucial (esp. Calibration information)



What is JSON?

- JSON stands for JavaScript Object Notation
- JSON is a lightweight format for storing and transporting data
- Plain text: machine- and human-readable
- JSON is often used when data is sent from a server to a web page
- JSON is "self-describing" and easy to understand
- Consists of `"property" : value` pairs

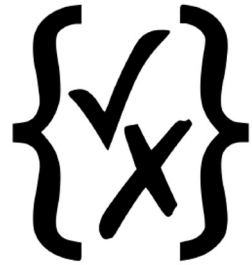
```
"SENSOR": "SDN:R25::FLUOROMETER_CHLA",
```

What is a schema?

An architecture or blueprint of how data will appear. A schema describes the shape of the data and how it relates to other models, tables and databases.

Argo NetCDF Schema

Name	Definition	Comment
SENSOR	<pre>char SENSOR(N_SENSOR, STRING32); SENSOR:long_name = "Name of the sensor mounted on the float"; SENSOR:conventions = "Argo reference table 25"; SENSOR:_FillValue = " ";</pre>	<p>Names of the sensors mounted on the float Example: CTD_PRES, CTD_TEMP, CTD_CNDC, OXYGEN_OPTCODE. See Argo reference table 25. Regular updates are made to an online version of this table available at: https://docs.google.com/spreadsheets/d/1Aw8B7FFUjG4e9MveD5qqvI7ZbB-z_x32IQnpHZbXVZA/edit#gid=1</p>



What is a JSON schema?

An architecture or blueprint of how JSON data will appear. A schema describes the shape of the data and how it relates to other models, tables and databases. **Instances instances (files) can be validated against schema(s).**

JSON Schema

```
"SENSOR": {  
  "type" : "string", "format" : "uri", "pattern" : "^SDN:R25::",  
  "description": "SENSOR string must be valid in current Argo reference table R25",  
  "validation-uri" : "https://vocab.nerc.ac.uk/collection/R25/current/"  
},
```

JSON Metadata Instance

```
"SENSOR": "SDN:R25::FLUOROMETER CHLA",
```

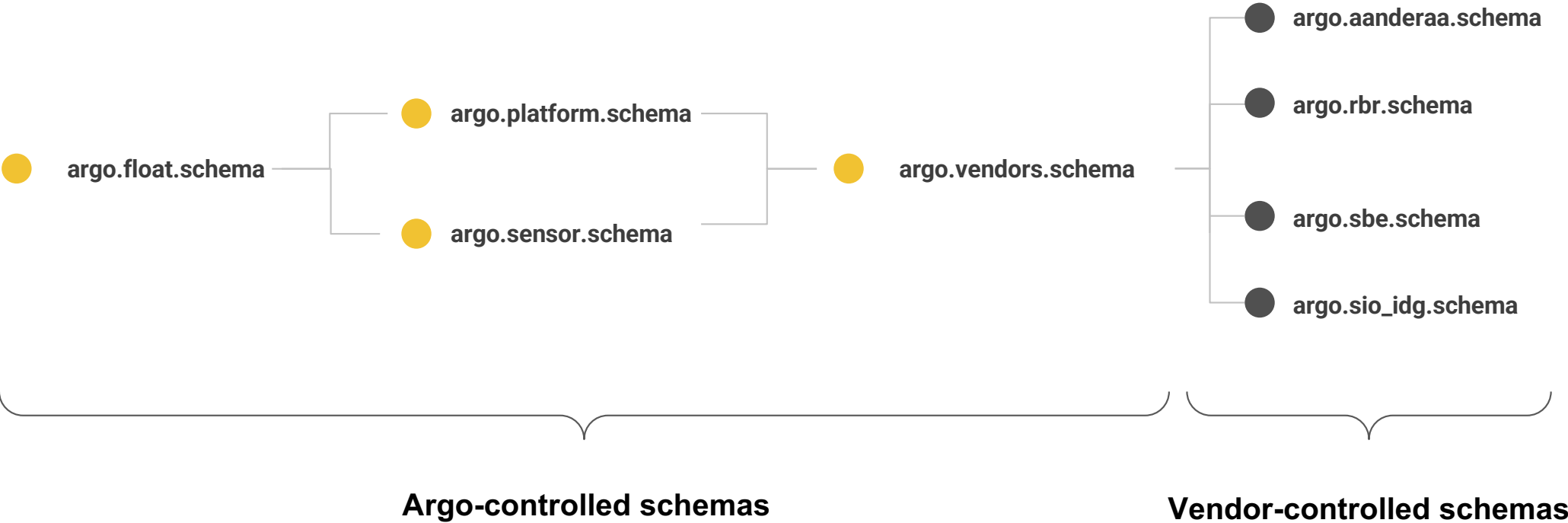
JSON Schema

```
{} argo.sensor.schema.json x {} sensor-WETLABS-MCOMS_FLBBCDx-0157.json U {} sensor-RBR-RBR_ARGO3-205908.json M
schema > {} argo.sensor.schema.json > ...
53 "SENSORS": {
54   "type": "array",
55   "items": {
56     "type": "object",
57     "properties": {
58       "SENSOR": {
59         "type": "string",
60         "format": "uri",
61         "pattern": "^SDN:R25::",
62         "description": "SENSOR string must be valid in current Argo reference table R25",
63         "validation-uri": "https://vocab.nerc.ac.uk/collection/R25/current/"
64       },
65       "SENSOR_MAKER": {
66         "type": "string",
67         "format": "uri",
68         "pattern": "^SDN:R26::",
69         "description": "SENSOR_MAKER string must be valid in current Argo reference table R26",
70         "validation-uri": "https://vocab.nerc.ac.uk/collection/R26/current/"
71       },
72       "SENSOR_MODEL": {
73         "type": "string",
74         "format": "uri",
75         "pattern": "^SDN:R27::",
76         "description": "SENSOR_MODEL string must be valid in current Argo reference table R27",
77         "validation-uri": "https://vocab.nerc.ac.uk/collection/R27/current/"
78       },
79       "SENSOR_SERIAL_NO": {
80         "type": "string",
81         "description": "Serial number of the sensor"
82       },
83       "SENSOR_FIRMWARE_VERSION": {
84         "type": "string"
85       },
86       "sensor_vendorinfo": {
87         "$ref": "../argo.vendors.schema.json#/defs/sensor_vendorinfo"
88       }
89     },
90     "required": [
91       "SENSOR",
92       "SENSOR_MAKER",
93       "SENSOR_MODEL",
94       "SENSOR_SERIAL_NO",
95       "SENSOR_FIRMWARE_VERSION"
```

JSON instance

```
{} argo.sensor.schema.json {} sensor-WETLABS-ECO_FLBBCD-3666.json x SBE-pHcalToJSON.py U {}
examples > {} sensor-WETLABS-ECO_FLBBCD-3666.json > {} @context > SDN:R25::
15 "SENSORS": [
16   {
17     "SENSOR": "SDN:R25::FLUOROMETER_CHLA",
18     "SENSOR_MAKER": "SDN:R26::WETLABS",
19     "SENSOR_MODEL": "SDN:R27::ECO_FLBBCD",
20     "SENSOR_SERIAL_NO": "3666",
21     "SENSOR_FIRMWARE_VERSION": " ",
22     "sensor_vendorinfo": {
23       "vendor_schema": "SBE",
24       "version": "0.1",
25       "SBE_manufacturing_date": "2014-07-17",
26       "CONFIG_EcoChlaFluorescenceExcitationWavelength_nm": 470,
27       "CONFIG_EcoChlaFluorescenceExcitationBandwidth_nm": 23,
28       "CONFIG_EcoChlaFluorescenceEmissionWavelength_nm": 695,
29       "CONFIG_EcoChlaFluorescenceEmissionBandwidth_nm": 70
30     }
31   },
32   {
33     "SENSOR": "SDN:R25::BACKSCATTERINGMETER_BBP700",
34     "SENSOR_MAKER": "SDN:R26::WETLABS",
35     "SENSOR_MODEL": "SDN:R27::ECO_FLBBCD",
36     "SENSOR_SERIAL_NO": "0157",
37     "SENSOR_FIRMWARE_VERSION": " ",
38     "sensor_vendorinfo": {
39       "vendor_schema": "SBE",
40       "version": "0.1",
41       "SBE_manufacturing_date": "2014-07-17",
42       "CONFIG_EcoBetaWavelength_nm": 700,
43       "CONFIG_EcoBetaBandwidth_nm": 25,
44       "CONFIG_EcoBetaAngle_angularDeg": 124
45     }
46   },
47   {
48     "SENSOR": "SDN:R25::FLUOROMETER_CDQM",
49     "SENSOR_MAKER": "SDN:R26::WETLABS",
50     "SENSOR_MODEL": "SDN:R27::ECO_FLBBCD",
51     "SENSOR_SERIAL_NO": "0157",
52     "SENSOR_FIRMWARE_VERSION": " ",
53     "sensor_vendorinfo": {
54       "vendor_schema": "SBE",
55       "version": "0.1",
56       "SBE_manufacturing_date": "2014-07-17",
57       "CONFIG_EcoCdFluorescenceExcitationWavelength_nm": 370
```

Schema organization



What's in a sensor file?

- **sensor.json**
 - sensor_info
 - *instrument_vendorinfo*
 - SENSORS (list)
 - SENSOR
 - SENSOR_MAKER
 - etc.
 - *sensor_vendor_info*
 - PARAMETERS (list)
 - PARAMETER
 - PARAMETER_SENSOR
 - etc.
 - *parameter_vendorinfo*
 - PREDEPLOYMENT_CALIB_EQUATION
 - PREDEPLOYMENT_CALIB_COEFFICIENT_LIST
 - etc.
 - *predeployment_vendorinfo*

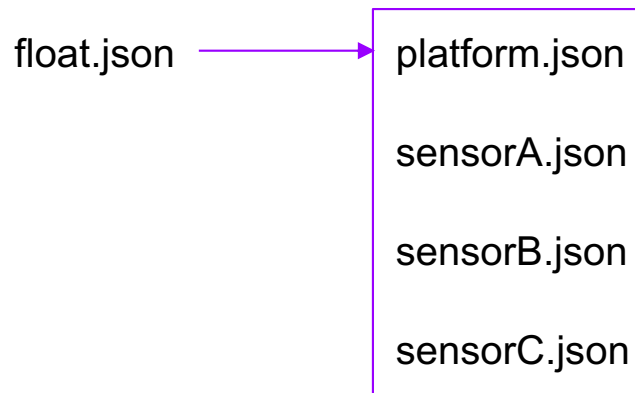
A platform file?

- **platform.json**
 - platform_info
 - *platform_vendorinfo*
 - PLATFORM (singleton)
 - PLATFORM_FAMILY
 - PLATFORM_TYPE
 - etc.
 - *platform_vendor_info*

```
"PREDEPLOYMENT_CALIB_COEFFICIENT_LIST": {  
  ...  
  "B0": "0.00624523",  
  "B1": "-0.00737614",  
  "B2": "-0.010341",  
  "B3": "-0.00817083",  
  "C0": "3.1201642E-3",  
  "PhaseCoef0": "-1.652",
```

What's in a float file?

It's an Argo-ready merge of platform + multiple sensor files:



- **float.json**
 - float_info
 - platform_info
 - Sensor_info_list (list)
 - sensorA_info
 - sensorB_info
 - sensorC_info
 - PLATFORM (singleton)
 - SENSORS (list)
 - sensor1
 - sensor2
 - sensor3
 - ...
 - PARAMETERS (list)
 - parameter1
 - parameter2
 - parameter3
 - ...

Meta-metadata describes JSON instance

- **date_creation:** date this file was created
- **link:** link to schema URL
- **format_version:** version of schema that is used to validate this instance
 - JSON format_version is encoded in the schema as a constant
 - JSON instance format_version assures “blueprint” matches schema
 - NOT used to track changes to JSON content, e.g. calibration ceoffs.
- *Identical for sensor_info, platform_info, float_info*

sensor_info meta-metadata
created_by
date_creation
link
format_version
contents
sensor_described

Linking an JSON metafile instance to schema

- JSON Schema provides only one formal way to link a schema to a JSON instance
 - Requires JSON delivered via HTTP
 - HTTP “Link” header points to schema. (Set in web server config file.)

Link: <"https://euro-argo.eu/schema/v0.2/argo.sensors.schema#"; rel="describedby">

- To be consistent, we provide hint in the **sensor_info/link** property
 - (\$schema property is reserved for defining the version of JSON Schema in the schema itself.)

```
"sensor_info": {  
  "created_by": "ECR test",  
  "date_creation": "2023-04-10T07:38:07-07:00",  
  "link" : "./argo.sensor.schema.json",  
}
```

JSON and Code Demo

- Schema, examples, sample code, etc. https://github.com/euroargodev/sensor_metadata_json
- Sea-Bird Demo

Main site:

<https://instrument.seabirdhub.com/>

Platform API example:

https://instrument.seabirdhub.com/api/argo-calibration?PLATFORM_SERIAL_NO=0068&PLATFORM_MODEL=NAVIS_EBR

Sensor API example:

https://instrument.seabirdhub.com/api/argo-calibration?SENSOR_SERIAL_NO=0490&SENSOR_MODEL=MCOMS_FLBBCD&SENSOR_TYPE=BACKSCATTERINGMETER_BB700

(SBS plans to have an API with just SENSOR_SERIAL_NO and SENSOR_MODEL that will give a complete JSON description of all SENSORS on that instrument.)