# **Proposal for a common web service core**

## **Background**

The OGC currently provides a range of standards intended to allow users to query data whilst abstracting them from the underlying delivery platforms, tied together by an OGC OWS specification re-engineered from the existing standards, but it does not really provide a framework to build new services. Due to the complexity of the data accessed, the standards can be both difficult to implement and consume and the richer the functionality that the standard supports the harder interoperability is to achieve.

The existing standards based around KVP based queries and XML based responses rely heavily on a shared vocabulary to provide the interoperability. This has worked well for the WMS standard and the WFS standard but for different reasons. In the case of the WMS the query interface provided to the consumer is simple; the heavy lifting performed on the server converts the underlying data source into a geo-referenced image. The WFS service provides a richer query interface to the consumer but is based on the assumption that the underlying data can be mapped into a row and column table structure i.e. the data structure that is queried against is simple.

WCS is a very much more complicated beast, although it started as a mechanism to query 2D raster datasets it has grown by using extensions (e.g. MetOcean, EO) that are designed to copes with everything from point clouds through to N dimensional data cubes as exemplified by ensemble climate models. This is despite trying to support all of the different types of query that may be required. Interoperability achieved by the extensions inheriting and extending the WCS conformance classes and schemas that both complicate the job of the service implementers and the service consumers. Thus interoperability is difficult to achieve, with varying degrees of success. The user is expected to understand the domain of the data queried and that of WCS as well. It also becomes harder and harder to add new types of data and new query approaches as the impacts of adding functionality on the existing functionality have to be analysed and approved.

With the exception of WMS, the current OGC standards are probably closer to query languages than modern API’s. The general approach for an API now is to have a multitude of self-describing specialist endpoints that perform a single task rather than a single open-ended flexible endpoint that requires domain knowledge to construct. This single task, self-describing approach, makes both implementation and interoperability easier. It also opens the information to a much wider consumer audience with the focus on abstracting the consumer away from the underlying data storage. The design of the API, based on the users query requirements, provides the information that the users require to construct the query as clearly and simply as possible.

By building upon the OpenAPI specification and including HTML outputs that support the schema.org tags, the new OGC WFS 3.0 specification provides a useful pattern that helps improve the user experience for the application developer. It would be possible to apply the same approach to other OGC standards, WMS should be straightforward to map onto the OpenAPI specification but it would be better to replace WCS with a multitude of specific methods rather than recreating it as a single standard. To enable interoperability in this OpenAPI based multitude of endpoints the OGC could host a repository or registry of OpenAPI definitions and work with schema.org to add or improve HTML tag definitions, the schema.org tagging approach provides a better approach than catalogues for enabling discovery of API’s.

## Overview

**A prototype is under construction and is best demonstrated by the following:**

Consider a well-used use case of extracting a cube of date from a 4D cube. The underlying data model is based on coverages model. A coverage collection is a set of related coverages. For the MetOcean community each NWP simulation (aka “model run”) is a collection of coverages. The contained coverages all share the same horizontal and temporal domains, *but not* the vertical coordinate reference system.

The API will is called “getDataCube” and has the following structure:-

1. http://service/getcube/1.0.0/api/ will return the full OpenAPI specification of the getcube API. This will either be in json or yaml.
2. http://service/getcube/1.0.0/conformance. Will list the conformance classes implemented by this API
3. http://service/getcube/1.0.0/collections/ would return a json/html document that lists all the available coverage collections and their associated coverages. Each coverage has a description of its domain i.e the axes and a URL that points to the ITEMS contained within each coverage. An example of such a link would be [***http://serviceaddress/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_AGL/items/f.json***](#_http://labs.metoffice.gov.uk/getcub_1)***. (see point 5) and Example 1***
4. http://service/getDataCube/1.0.0/collections/{CollectionId). The collectionID are listed in the collections response to http://service/getcube/1.0.0/collections, for example the following would simply return the string items. <http://service/getcube/1.0.0/collections/EGloEGRR_EGloEGRR_2018-10-31T00.00.00Z_AGL/> would simply return the string **"items"**
5. http://service/getcube/1.0.0/collections/EGloEGRR\_EGloEGRR\_2018-10-31T00.00.00Z\_AGL/Items will return a list of the parameters needed to construct a data query on the AGL coverage (Above Ground Level) and the permissible ranges for each of the query parameters. It also includes the query URL, e.g. *http://serviceaddress/****getcube****/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_AGL/items/data*",

Note this pattern is easily extensible to many other data extraction patterns, e.g. trajectories

***Example 1 (http://service/getcube/1.0.0/collections/)***

***In this example there are two models that are being described, a global and limited area model.***

***The global model as (in this example) 7 coverages and each are described in detail, For each overage e.g. of GlobEGRR\_2018-10-31T00.00.00Z\_ISBL the following is available:-***

***The limited area model is described in the same way.***

***Notice that each have an embedded URL e.g. http://serviceaddress/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_ISBL/items/f.json***

***If that link is followed (see example 2) then the query parameters are described:-***

***{***

***"Global\_Coverages": [***

***{***

***"url": "http://serviceaddress/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_ISBL/items/f.json",***

***"name": "GlobEGRR\_2018-10-31T00.00.00Z\_ISBL",***

***"description": {***

***"summary": "Values which are measured at Standard ISOBaric levels",***

***"created": "2018-10-31T00:00:00Z",***

***"axes": {***

***"Lat": {***

***"units": "deg",***

***"lowerBound": "-90",***

***"upperBound": "90"***

***},***

***"Lon": {***

***"units": "deg",***

***"lowerBound": "-180",***

***"upperBound": "180"***

***},***

***"Time": {***

***"units": "ISO8601",***

***"lowerBound": "2018-10-31T00:00:00Z",***

***"upperBound": "2018-11-07T00:00:00Z"***

***}***

***}***

***}***

***}***

***],***

***"EURO4\_Coverages": [***

***{***

***"url": "http://serviceaddress/getcube/1.0.0/collections/EURO4\_EURO4\_2018-10-31T00.00.00Z\_ISBL/items/f.json",***

***"name": "EURO4\_2018-05-0T00.00.00Z\_ISBL",***

***"description": {***

***"summary": "Values which are measured at Standard ISOBaric levels",***

***"created": "2018-10-31T00:00:00Z",***

***"axes": {***

***"Lat": {***

***"units": "deg",***

***"lowerBound": "27",***

***"upperBound": "70"***

***},***

***"Lon": {***

***"units": "deg",***

***"lowerBound": "-38",***

***"upperBound": "61"***

***},***

***"Time": {***

***"units": "ISO8601",***

***"lowerBound": "2018-10-31T00:00:00Z",***

***"upperBound": "2018-11-03T12:00:00Z"***

***}***

***}***

***}***

***}***

***]***

***}***

***Example 2***

***A result of following this link http://serviceaddress/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_ISBL/items/f.json***

***{***

***"description": "UK Global data",***

***"query\_url": "http://serviceaddress/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_ISBL/items/data",***

***"timePeriod": "2018-10-31T00:00:00Z/2018-11-06T00:00:00Z",***

***"queryStringParameters": {***

***"yRange": {***

***"native\_grid\_resolution": "215",***

***"label": "Lat",***

***"uomLabel": "deg",***

***"lowerBound": "-89.576",***

***"upperBound": "89.583",***

***"interpolation\_options": [***

***"none",***

***"nearest-neighbor",***

***"linear",***

***"cubic"***

***],***

***"type": "regularAxis"***

***},***

***"xRange": {***

***"native\_grid\_resolution": "288",***

***"label": "Lon",***

***"uomLabel": "deg",***

***"lowerBound": "-180.000",***

***"upperBound": "180.000",***

***"interpolation\_options": [***

***"none",***

***"nearest-neighbor",***

***"linear",***

***"cubic"***

***],***

***"type": "regularAxis"***

***},***

***"tRange": {***

***"native\_grid\_resolution": "26",***

***"label": "Time",***

***"uomLabel": "ISO8601",***

***"values": [***

***"2018-10-31T00:00:00Z",***

***"2018-10-31T03:00:00Z",***

***"2018-10-31T06:00:00Z",***

***"2018-10-31T09:00:00Z",***

***"2018-10-31T12:00:00Z",***

***"2018-10-31T15:00:00Z",***

***"2018-10-31T18:00:00Z",***

***"2018-10-31T21:00:00Z",***

***"2018-11-01T00:00:00Z",***

***"2018-11-01T03:00:00Z",***

***"2018-11-01T06:00:00Z",***

***"2018-11-01T09:00:00Z",***

***"2018-11-01T12:00:00Z",***

***"2018-11-01T15:00:00Z",***

***"2018-11-01T18:00:00Z",***

***"2018-11-01T21:00:00Z",***

***"2018-11-02T00:00:00Z",***

***"2018-11-02T06:00:00Z",***

***"2018-11-02T12:00:00Z",***

***"2018-11-02T18:00:00Z",***

***"2018-11-03T00:00:00Z",***

***"2018-11-03T12:00:00Z",***

***"2018-11-04T00:00:00Z",***

***"2018-11-04T12:00:00Z",***

***"2018-11-05T00:00:00Z",***

***"2018-11-05T12:00:00Z",***

***"2018-11-06T00:00:00Z"***

***],***

***"interpolation\_options": [***

***"nearest-neighbor",***

***"linear",***

***"cubic"***

***],***

***"type": "irregularAxis"***

***},***

***"zRange": {***

***"native\_grid\_resolution": "15",***

***"label": "Isobaric\_Surface",***

***"uomLabel": "hPa",***

***"values": [***

***"1000",***

***"950",***

***"925",***

***"850",***

***"700",***

***"600",***

***"500",***

***"400",***

***"300",***

***"250",***

***"200",***

***"150",***

***"100",***

***"70",***

***"50",***

***"30"***

***],***

***"interpolation\_options": [],***

***"type": "irregularAxis"***

***},***

***"dataTypes": {***

***"relative-humidity": {***

***"type": "Parameter",***

***"description": {***

***"en": "Relative humidity"***

***},***

***"unit": {***

***"label": {***

***"en": "per cent"***

***},***

***"symbol": {***

***"value": "%",***

***"type": "http://www.opengis.net/def/uom/UCUM/"***

***}***

***},***

***"observedProperty": {***

***"id": "http://codes.wmo.int/grib2/codeflag/4.2/0-1-1",***

***"label": {***

***"en": "Relative humidity"***

***}***

***}***

***},***

***"temperature": {***

***"type": "Parameter",***

***"description": {***

***"en": "Temperature"***

***},***

***"unit": {***

***"label": {***

***"en": "kelvin"***

***},***

***"symbol": {***

***"value": "K",***

***"type": "http://www.opengis.net/def/uom/UCUM/"***

***}***

***},***

***"observedProperty": {***

***"id": "http://codes.wmo.int/grib2/codeflag/4.2/0-0-0",***

***"label": {***

***"en": "Temperature"***

***}***

***}***

***}***

***},***

***"outputFormat": [***

***"netcdf",***

***"geotiff",***

***"covjson"***

***]***

***}***

***}***

***Example 3***

**This a typical call for a get data cube,**

http://data.example.org/getcube/1.0.0/collections/GlobEGRR\_GlobEGRR\_2018-10-31T00.00.00Z\_ISBL/data?parameters=Temperature,Wind&xRange=Longitude,deg,0.0,10.0&yRange=Latitude,degs%2C50.0,60.0&tRange=Time,ISO8601,2018-10-31T00:00:00Z,2018-11-02T00:00:00Z &zRange=Pressue,hPa,200.0,1000.0&outputFormat=COVJson&outputCRS=epsg4326