A COMMON, OPEN SOURCE INTERFACE BETWEEN EARTH OBSERVATION DATA INFRASTRUCTURES AND FRONT-END APPLICATIONS

Deliverable 05

Version 1.0 from 2018/03/27

Overview document about back offices metadata standards and interfaces

Change history

Issue	Date	Author(s)	Description
0.0	2018/03/01	openEO consortium	Answers to questionnaire, Appendix
0.1	2018/03/07	Edzer Pebesma, Matthias Mohr, WWU	First draft.
0.2	2018/03/23	Alexander Jacob, EURAC	Review Editing
1.0	2018/03/27	Matthias Schramm, TU Wien	Final review and creation of final version

For any clarifications please contact openEO@list.tuwien.ac.at.

Number of pages: **29**

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776242. Any dissemination of results reflects only the author's view(s) and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

Table of Contents

List of Tables

List of Figures

List of Acronyms

API Application Programming Interface, a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components (wikipedia)

EO Earth Observation

OGC Open Geospatial Consortium, a standardisation body

openEO name of the project for which this report was written

1. Executive Summary

This “overview document about back offices metadata standards and interfaces” report complements the “openEO proof-of-concept” deliverables D3.1, D4.2 and D5.1 (all demonstrators) by describing differences and correspondences of computational back office (or back-end, or server) properties relevant to the design choices of the openEO API. It focuses on available standards for describing image collection data sets and processes, describes which openEO partners (and/or back-ends) already use particular standards, and gives example queries and responses. It then discusses the value of these standards for the openEO API. It concludes that for describing processes, existing standards are of little use. For data discovery, the recent OpenSearch seems to have most potential.

2. Introduction

The EU funded project openEO [1] develops a single interface (API) for connecting multiple clients (user-side software) to multiple back-ends (server-side software, also referred to as *back office*). With such an API, a client that can operate one back-end can also operate all other back-ends supporting the same API (Fig. 2.1). The advantage of this is *interoperability*, meaning that more systems can work with each other. As a consequence, not only less software needs to be written, but users can compare, *using a single client*, offerings of different back-ends, in terms of pricing but also in terms of whether identical requests to different servers return identical results.

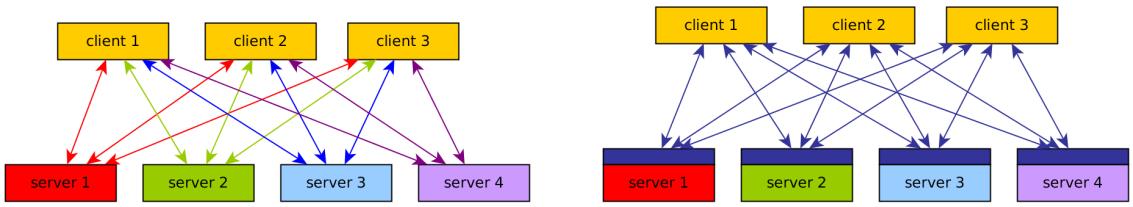

Figure 2.1: Left: each back-end has its own API, for full interoperability each of m clients would have to support APIs of n servers; openEO (right) develops a single API (dark blue) for which only n server-side API drivers are needed

Typically, remote sensing data is distributed by space agencies in the form of granules (or scenes): files (or sets of files) with georeferenced grid (raster) data for a rectangular region that give observed pixel values obtained during a single overpass of the satellite (e.g. Sentinel and Landsat) or aggregated over a short time frame (MODIS). When using remote sensing e.g. for change analysis over a larger region, all granules covering the region and time period need to be identified, organised, and processed.

As we argue in [2], users are most helped when the organisation of the data in granules can be abstracted away, but can directly operate on *image collections*. This means that the analysis is file agnostic, and analysis addresses space and time dimensions directly, *without the need to explicitly select granules*[1].

[1]The canonical example of a platform that successfully implemented this approach is Google Earth Engine

The openEO API will enable researchers to work on a “data cube view” of the EO imagery and directly filter, aggregate or map functions over dimensions of a cube (e.g. spatial, temporal, spectral) without being concerned about how the data in the processing platform is organised (by granule, as collections of GRIB or NetCDF files, as one or several arrays in an array database, etc.).

To reach this goal, existing back-end systems need to be adapted (or interfaced) to fit the new API. In its first phase, the openEO project has designed an API and implemented this for a number of clients and back-ends in a *proof of concept*, reported on in a set of demonstrator deliverables (D3.1, D4.2 and D5.1) but also demonstrated on the project home page at http://openeo.org/. This document describes which metadata standards and interfaces are used by existing back-ends. This information is meant to help the development of the openEO API.

This report is organised as follows: section 3 briefly describes the back-ends used in the proof-of-concept. Section 4 describes existing standards used by the respective back-ends. Sections 5 and 6 discuss the options, and give tentative conclusions. The appendix contains answers to the questionnaire given by all back-ends considered as well as by Google Earth Engine (appendix A), and example queries and responses of the various services (appendix B).

3. Overview of back-ends

The back-ends for which we will discuss metadata standards and interfaces are those listed in table 3.1. The openEO proof-of-concept also uses an R back-end (“R Back-end WWU”), which was developed by WWU for the purpose of openEO development and reference testing. As such, it does not support metadata standards or interfaces other than demanded for openEO, and for that reason it is left out of the discussion in this report.

Back-end name	Partner	Software
Grass GIS Mundialis	Mundialis	Grass GIS
OpenShift EODC	EODC	various, open source
GeoPySpark VITO	VITO	GeoTrellis
Sentinel Hub Sinergise	Sinergise	proprietary
WCPS EURAC	EURAC	mostly Rasdaman community

Table 3.1: Back-end drivers considered in this report.

A detailed description of the back-ends is found in openEO Deliverable 4.2 [3].

4. Standards involved

4.1. Discovery

4.1.1. CSW

CSW is an abbreviation for *catalogue service*, an open standard for spatial data catalogues. It can be used to discover and query spatial metadata.

CSW (catalogue service) specifications are found here: http://www.opengeospatial.org/standards/cat.

Two example CSW queries are given in appendix B.1

4.1.2. WCS

WCS (web coverages service) is an OGC standard for querying and downloading coverage data [4].

Coverages are phenomena that have a single (scalar, or functional) value for each space and time combination; they can be seen as a function

$$S \times T \times ... \rightarrow Q \tag{1}$$

mapping from *domain* space and time $(S \times T)$ and potentially further dimensions onto *range* Q. Q might be a scalar (like temperature of air) or a function like energy distribution (E) as a function of wavelength L; $L \rightarrow E$. In the latter case, we could also define the coverage as $S \times T \times L \rightarrow E$, in which case the range variable is a scalar, see [5, 6] for longer discussions on this.

Coverages are often represented by raster data for 2-D spatial datasets and arrays for 2-D spatial + time, or 2-D spatial + time + spectral. When we have sensor data, e.g. air quality measurements from monitoring stations, the spatial locations are no longer regularly distributed over space and a representation by irregular points is more appropriate. In case we have categorical values such as soil type or land cover type, polygons are often used to delineate the areas with constant value. Finally, contour lines are a way to describe the sets of points where a coverage value is constant. The Coverage Implementation Schema ("CIS") OGC document attempts to reconcile these different views on coverages.

All OGC WCS specifications are found here: http://www.opengeospatial.org/standards/wcs. The web coverage standard considers space and time as dimensions but spectral band as an attribute, not as a spectral dimension. The query parameter to obtain the description of WCS coverages is `GetCoverageDescription`

The OGC WCS profile for Earth Observation [7] describes “Dataset Series” as

> “ A Dataset Series is a collection of coverages and/or Dataset Series; cf. Subclause 6.6. A Dataset Series can refer to any number of Datasets, Stitched Mosaics, and Dataset Series. A Dataset Series is not a coverage itself”

which corresponds to our earlier use of *image collection*. According to WCS specifications, image collections (dataset series) are not coverages. According to our definition in 1 an image collection is a coverage, and [5] shows how it can be handled properly in an array database.

4.1.3. OpenSearch

OpenSearch (web site: http://www.opensearch.org/Home) is “a collection of simple formats for the sharing of search results”.

The Committee on Earth Observation Satellites (CEOS) has developed best practices[2] of OpenSearch in order to allow for standardized and harmonized access to metadata and data

[2] http://ceos.org/ourwork/workinggroups/wgiss/access/opensearch/

of CEOS agencies, including CWIC and FedEO communities. Their web site points to a best practices document, and a developer guide.

The CEOS best practice document [8] comes up with the definitions of *granule* and *collection* that openEO also adopts:

> "A granule is the finest granularity of data that can be independently managed. A granule usually matches the individual file of EO satellite data."
>
> "A collection is an aggregation of granules sharing the same product specification. A collection typically corresponds to the series of products derived from data acquired by a sensor on board a satellite and having the same mode of operation."

OGC provides a document for OpenSearch Geo and Time extensions [9]. It mostly attempts to tie up the search concepts into other OGC standard documents; the CEOS best practice advices to comply with this, as well as with [OGC 13-026r8]: OpenSearch Extension for Earth Observation Products [10]:

> "The Geo and Time extensions [OGC 10-032r8] specify a series of query parameters that can be used to geographically and temporally constrain search results. Parameters include: bounding box, geometry, and start/end of a temporal extent. It also defines a set of response elements to be used to express geographical and temporal context in the search results."
>
> "The extension for EO Products [OGC 13-026r8] defines query parameters that allow the filtering of search results with the fields that are unique to EO products, e.g. platform (satellite), sensor, processing center, etc. The OpenSearch query parameters defined in this document are aligned with O&M EO Profile [OGC 10-157r4] that describes EO products metadata."

The first best practice is: **CEOS-BP-001 - Support of two step search [Recommended]**:

> "One serious hurdle to overcome in searching for data is the great number of data items to account for in responses, as well as the expected number of successful "hits" for a query. In ordinary web searches, the searcher is usually looking for a small number of web pages or documents. Relevance ranking typically does a good job of presenting these successful hits near the top of the returned list, followed by single point-and-click retrievals. However, when searching for Earth science data covering large time periods or spatial areas, a user will often specify a set of constraints to find an appropriate data collection together with space-time criteria for files within that data collection. Often, the precision of the data collections returned for the search is low, with many spurious hits. However, the space-time precision of the files is often quite high: that is, the user truly wants to use all the data files of a desirable data collection set that fall within the spacetime region of interest. Thus, searching for all data satisfying both dataset content and space-time region at the same time can produce a great many spurious hits, i.e., all the files for data collections that are not desired.
>
> The 2-step search consists of a collection level search and the subsequent granule level search (or file-level search) as Figure 4 shows. "

More extensions and best practice documents related to the OpenSearch standard are available, including but not limited to:

- Geo to search for geographic extents or location names.
- Referrer to reveal the source of the search to the service.
- The software project resto is adding earth observation related capabilities for their search engine.
- SRU to make the SRU (Search/Retrieve via URL) standard available for OpenSearch.
- Time to search for temporal extents.
- Earth Data discovery OpenSearch Best Practices published by ESIP.

Table 4.1 shows an overview regarding supported OpenSearch capabilities by the back offices.

		VITO	Sentinel Hub	EODC
OpenSearch Version		1.1	1.1	1.1
Output Formats	XML			✓
	Atom	✓	✓	✓
	Others		HTML, JSON	
Extensions	EO	✓	✓	
	Geo	✓	✓	✓
	Referrer	✓		
	Resto		✓	
	SRU	✓		
	Time	✓	✓	✓

Table 4.1: Supported OpenSearch versions, output formats and extensions by back offices.

4.2. Process description

We will briefly describe WPS and WCPS, two OGC standards for processing data.

4.2.1. WPS

The OGC Web Processing Service [11] allows for arbitrary processing being done by a web processing service. The WPS can describe the processes it offers, and allows for their execution, where input and execution parameters need to be passed, and process output is returned. In openEO we allow users to define a process graph, which represents an arbitrarily complex expression formed from primitive functions. This would not fit the WPS well, since it wants to advertise, and describe, each process it offers. Chaining WPS's providing primitive functions does not seem to be a realistic proposition. The WPS model assumes that the user submits process inputs with the execute request; for openEO the data is, and will remain, on the server.

4.2.2. WCPS

A standardised service for processing of coverages is the OGC WCPS [12].

WCPS (web coverages processing service) [12] does not return its processing capability because the processing capabilities are part of, and by that defined by any implementation of, the standard (section 7; appendix B.2 gives the syntax in Backus-Naur form). It does allow for expressing processes as expressions, e.g. `a.red + a.green` would add the red and green attributes of coverage `a`.

5. Discussion

In [2] we (the openEO consortium) argue that users are most helped when they get the ability to analyse the data in a *cube view*, meaning that the data are analysed in a certain spatial and temporal resolution, but that they don't have to be stored in that resolution. In case they are not stored in a data cube, they may be resampled on-the-fly to the data cube requested. This approach has the advantage that no data gets lost, and that complete flexibility is retained as to *which* cube a user wants to analyse image collections.

According to WCS, image collections are *not* coverages [7]. This is neither justified by the theoretical model (1) nor by implementation in an array database [5], and [7] remains unclear about what motivated this choice. If one would follow the WCS, one would have to make so-called "stitched mozaics", which requires abandoning of overlaps of tiles and exact time of taking the image. In addition, the WCS specifies that different bands need to be different attributes in the domain, rather than a dimension of its own. This choice also seems arbitrary and unnecessarily limiting. Altogether, although WCS gives one possibility, it seems too restrictive for the openEO API.

Processing is specified by the WCPS [12] and WPS [11]. To fulfill the openEO goals, WCPS seems too restricted in the datamodel assumed and in the operations allowed (which are hard-coded in the standards document). WPS seems to be too generic and course grained: it is not suited to the model where *process graphs* (or more generally: chained processes) are passed on to the service; each process graph would have to be advertised as a process, limiting the flexibility required by users.

For the data discovery part, OpenSearch and its extensions for space and time [9] and Earth Observation [10] seem promising. Recommendations for OpenSearch by CEOS [8] provide guidelines how to advertise image collections as search items.

6. Conclusions

This document describes the standards and services supported by different existing back-ends for discovering data and processes, in the context of their potential adoption in the openEO API. None of the services allows for automated discovery of available processes in a way that is useful for openEO. Services for discovering data include CSW, WCS, and OpenSearch. OpenSearch with its EO and time and geo extension can describe image collections. Among existing standards, OpenSearch and it extensions seems so far the most promising candidate for discovering and describing image collections.

7. References

[1] E. Pebesma, W. Wagner, M. Schramm, A. Von Beringe, C. Paulik, M. Neteler, J. Reiche, J. Verbesselt, J. Dries, E. Goor, and et al., "openEO – a common, open source interface between earth observation data infrastructures and front-end applications," *Zenodo*, Nov 2017. [Online]. Available: https://zenodo.org/record/1065474

[2] E. Pebesma, W. Wagner, P. Soille, M. Kadunc, N. Gorelick, M. Schramm, J. Verbesselt, J. Reiche, M. Appel, J. Dries, A. Jacob, M. Neteler, S. Gebbert, C. Briese, and P. Kempeneers, "openEO: an open API for cloud-based big earth observation processing platforms," *EGU2018-4957 ESSI2.2 - Data cubes of Big Earth Data - a new paradigm for accessing and processing Earth Science Data*, 2018. [Online]. Available: https://meetingorganizer.copernicus.org/EGU2018/EGU2018-4957.pdf

[3] G. Busch and openEO consortium, *Two early prototype back-ends, openEO Deliverable 4.2*. openEO consortium, 2018. [Online]. Available: http://openeo.org/links/

[4] OGC, *OGC WCS 2.0 Interface Standard- Core: Corrigendum; 09-110r4*. Open Geospatial Consortium, 2012. [Online]. Available: http://www.opengeospatial.org/standards/wcs

[5] M. Appel, F. Lahn, W. Buytaert, and E. Pebesma, "Open and scalable analytics of large earth observation datasets: From scenes to multidimensional arrays using SciDB and GDAL," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 138, pp. 47 – 56, 2018 (open access). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271617300898

[6] S. Scheider, B. Gräler, E. Pebesma, and C. Stasch, "Modeling spatiotemporal information generation," *International Journal of Geographical Information Science*, vol. 30, no. 10, pp. 1980–2008, 2016. [Online]. Available: http://pebesma.staff.ifgi.de/generativealgebra.pdf

[7] OGC, *OGC WCS Interface Standard - Earth Observation Application Profile, version 1.0.0; 10-140r1*. Open Geospatial Consortium, 2014. [Online]. Available: http://www.opengeospatial.org/standards/wcs

[8] CEOS, *CEOS OpenSearch Best Practice Document, Version 1.2*. CEOS, 2014. [Online]. Available: http://ceos.org/document_management/Working_Groups/WGISS/Interest_Groups/OpenSearch/CEOS-OPENSEARCH-BP-V1.2.pdf

[9] OGC, *OpenSearch Geo and Time Extensions; 10-032r8*. Open Geospatial Consortium, 2014. [Online]. Available: http://www.opengeospatial.org/standards/opensearchgeo

[10] ——, *OGC OpenSearch Extension for Earth Observation; 13-026r8*. Open Geospatial Consortium, 2016. [Online]. Available: http://www.opengeospatial.org/standards/opensearch-eo

[11] ——, *OGC WPS 2.0.2 Interface Standard Corrigendum 2; 14-065r2*. Open Geospatial Consortium, 2018. [Online]. Available: http://www.opengeospatial.org/standards/wps

[12] ——, *Web Coverage Processing Service (WCPS) Standard; 08-068r2*. Open Geospatial Consortium, 2009. [Online]. Available: http://www.opengeospatial.org/standards/wcps

Appendices

A. Answers to questions regarding back ends

D 4.1 Overview document about back offices metadata standards and interfaces
Back-ends to consider (handling partner):

- A.1. GRASS GIS (Mundialis)
- A.2. File-based (EODC)
- A.3. GeoTrellis (VITO)
- A.4. Sentinel Hub (Sinergise)
- A.5. WCPS (EURAC)
- A.6. Google Earth Engine (WWU)

Questions to be answered for every back-end:

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?
2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)
3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)
4. Can you provide an example query, and point to the documentation and/or API specification?
5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?
6. Does the back-end provide a structured way to obtain metadata about available processes?
7. If so, which standards does it use for this (if any)?
8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

A.1. GRASS GIS back-end by mundialis

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?

   The back-end itself provide multiple endpoints to query metadata about the GRASS GIS database. These endpoints will be used by the openEO Core API wrapper that exposes the GRASS GIS back-end. The wrapper will support the openEO `/data` endpoint to expose raster layer and image collection (in GRASS GIS called STRDS, for SpatioTemporal

Raster DataSet) in the GRASS GIS database that uses the v0.01 openEO Core API standard with enhanced metadata.

2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)

   The back-end uses JSON to describe the metadata of raster layers, vector layers and raster time series that is fully documented using swagger 2.0. We use the v0.01 openEO Core API standard to make the GRASS GIS database information available through openEO API. There are no plans to provide an OGC based endpoint for metadata access

3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)

   The back-end supports time range as well as extreme value filters for time series data as well as pattern based filter of raster and vector layer names. The filter will not be supported in the openEO Core API wrapper.

4. Can you provide an example query, and point to the documentation and/or API specification?

   https://actinia.mundialis.de/apidocs/ https://actinia.mundialis.de/api/swagger.json

5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?

   Yes, but in a limited form, based on the GRASS GIS time series metadata. There are no band descriptions available for image collections. Bands must be specified as single time series.

6. Does the back-end provide a structured way to obtain metadata about available processes?

   Yes, and this will be exposed using openEO Core API version 0.01 to describe jobs

7. If so, which standards does it use for this (if any)?

   JSON is used. The data format is fully documented using swagger 2.0.

8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   No.

A.2. File-based back end, EODC

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?

   https://csw.eodc.eu

2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)

   - OGC CSW 2.0.2 & 3.0.0
   - OGC Filter 1.1.0

- OGC OWS Common 1.0.0
- OGC GML 3.1.1
- OGC SFSQL 1.2.1
- Dublin Core 1.1
- SOAP 1.2
- ISO 19115 2003
- ISO 19139 2007
- ISO 19119 2005
- NASA DIF 9.7
- FGDC CSDGM 1998
- SRU 1.1
- A9 OpenSearch 1.1

3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)

   Filter capabilities can be obtained at: https://csw.eodc.eu/?service=CSW&version=2.0.2&request=GetCapabilities&sections=Filter_Capabilities

   - Full Text: csw:AnyText
   - Geometry Operands: gml:Point, gml:LineString, gml:Polygon, gml:Envelope
   - Spatial Operators: BBOX, Beyond, Contains, Crosses, Disjoint, DWithin, Equals, Intersects, Overlaps, Touches, Within
   - Logical Operators: Between, , EqualTo, GreaterThan, GreaterThanEqualTo, LessThan, LessThanEqualTo, Like, , NotEqualTo, NullCheck
   - Functions: length, lower, ltrim, rtrim, trim, upper
   - Record queries constraints:
   - apiso: Abstract, AccessConstraints, AlternateTitle, AnyText, BoundingBox, Classification , ConditionApplyingToAccessAndUse , CRS, Contributor , CouplingType, Creator, CreationDate, Degree , Denominator, DistanceUOM, DistanceValue, Format, GeographicDescriptionCode, HasSecurityConstraints, Identifier, KeywordType, Language, Lineage, Modified, OperatesOn, OperatesOnIdentifier, OperatesOnName, Operation, Operation, OtherConstraints , ParentIdentifier, Publisher, PublicationDate, ResourceLanguage, Relation, RevisionDate, ServiceType, ServiceTypeVersion, ResponsiblePartyRole, SpecificationDate, SpecificationDateType, SpecificationTitle, Subject, TempExtent_begin, TempExtent_end, Title, TopicCategory, Type
   - csw: AnyText
   - dc: contributor, creator, date, format, identifier, language, publisher, relation, rights, source, subject, title, type, abstract, alternative, modified, spatial
   - ows: BoundingBox

4. Can you provide an example query, and point to the documentation and/or API specification?
   - Example Query: https://csw.eodc.eu/?service=CSW&version=2.0.2&request=GetRecords&typenames=csw:Record&elementSetName=full&resultType=results&constraintLanguage=CQL_TEXT&startposition=1&constraint=apiso:TempExtent_begin%20>%20'2018-02-02'
   - Capabilities: https://csw.eodc.eu/?service=CSW&version=2.0.2&request=GetCapabilities
   - pycsw documentation of: http://docs.pycsw.org/en/latest/introduction.html
5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?
   - Yes, descriptions about available image collections can be obtained at the URI: https://csw.eodc.eu/?service=CSW&version=2.0.2&request=GetRecords&typenames=csw:Record&elementSetName=full&resultType=results&constraintLanguage=CQL_TEXT&startposition=1&constraint=apiso:Type%20=%20%27series%27
   - Record Attributes:
     - dc:identifier, dc:title, dc:type, dc:subject, dc:format, dct:references, dct:modified, dct:abstract, dc:date, dc:creator, dc:publisher, dc:language, dc:rights, dct:alternative, dct:spatial, ows:BoundingBox
6. Does the back-end provide a structured way to obtain metadata about available processes?

   No, we currently do not offer processes.
7. If so, which standards does it use for this (if any)?

   See 6.
8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   We additionally provide dct:references on GetRecords operations to retrieve the file paths of the records returned, that are pointing to the location of the files in the EODC storage:

```
<dct:references scheme="offlineAccess">
    /eodc/products/copernicus.eu/s1b_csar_grdh_iw/2018/02/02/S1B_IW_GRDH_
    1SDV_20180202T150315_20180202T150340_009446_010FB8_3334.zip
</dct:references>
```

A.3. VITO Geotrellis back-end

Summary: Overall geotrellis does not implement any metadata standard, so we will have to implement things ourselves. At VITO we do expose a lot of metadata using standardized interfaces, (WCS/Opensearch/CSW/Inspire XML/...), so we may be able to reuse some of that.

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?

- http://openeo.vgt.vito.be/openeo/data returns a list of available layers. Included metadata is still limited but should be expanded to available dates, bounding boxes.
- http://www.vito-eodata.be/openSearch_all/description.xml List collections available for file-based access. (Currently not planned to support by the back-end.)
- WCS interface for direct download of base layers: https://proba-v-mep.esa.int/applications/geo-viewer/app/geoserver/ows?SERVICE=WCS&REQUEST=GetCapabilities
- This interfaces currently only works on static, predefined datasets. We do not (yet) have a WCS interface on geotrellis layers, but it may become available in the upstream open source project.

2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)
   - OpenSearch: esipdiscovery:version="1.2"
   - WCS = 2.0.1 older versions are supported by the software, but not recommended for use
3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)
   - In opensearch this is possible
   - In Geotrellis: metadata is returned at a layer (collection) level, and not per tile, so this question does not apply there.
4. Can you provide an example query, and point to the documentation and/or API specification?

   http://openeo.vgt.vito.be/openeo/data gives result:


```
{
   "product_id": "CGLS_NDVI_V1",
   "zoom": 0
},
{
   "product_id": "CGLS_NDVI_V2",
   "zoom": 0
},
{
   "product_id": "CGS_SENTINEL2_RADIOMETRY_V101",
   "zoom": 0
},
```


5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?

   Yes, that's the only thing it currently provides. (see above)
6. Does the back-end provide a structured way to obtain metadata about available processes?

No, geotrellis does not have this concept, so we will need to implement and maintain our own description database for OpenEO

7. If so, which standards does it use for this (if any)?

   Not applicable

8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   No

A.4. Sentinel Hub back-end

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?

   Yes

2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)

   - For Sentinel-2 and Landsat 8 there is OpenSearch http://opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/describe.xml http://opensearch.sentinel-hub.com/resto/api/collections/Landsat8/describe.xml

     RESTO OpenSearch version 2.3

   - For other data sources (S-1, S-2 L1C, S-2 L2A, S-3, Landsat 5-7-8, ENVISAT MERIS, MODIS) there is WCS 1.0.0 and WFS 2.0.0

3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)

   Yes:

   - OpenSearch: boundingBox, geometry (region of interest), product_id, startDate / completionDate /publishedDate/updatedDate, cloudCover, spacecraft (S2A, S2B), orbitNumber
   - WFS: date, time, path, crs, mbr, cloudCoverPercentage, geometry

4. Can you provide an example query, and point to the documentation and/or API specification?

   Example: http://opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/search.json?maxRecords=2&cloudCover=100

   Docs: https://github.com/jjrom/resto

   Example: https://services.sentinel-hub.com/ogc/wfs/ef60cfb1-53db-4766-9069-c5369c3161e6?service=WFS&request=GetCapabilities

   Docs: http://www.opengeospatial.org/standards/wfs

5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?

   No, except listing of collections

6. Does the back-end provide a structured way to obtain metadata about available processes?

   No

7. If so, which standards does it use for this (if any)?

   N/A

8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   No other metadata is accessible through the back-end services, but Sentinel2 on AWS keeps SAFE and inspire metadata XMLs in 'product' folders, e.g. http://sentinel-s2-l1c.s3-website.eu-central-1.amazonaws.com/#products/2016/4/19/S2A_OPER_PRD_MSIL1C_PDMC_20160419T042500_R017_V20160419T014649_20160419T014649/

A.5. EURAC WC(P)S back-end (openEO api 0.0.2)

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)? YES
   a. (http://sdi.eurac.edu/geonetwork/srv/eng/main.home) web viewing only however
   b. (http://saocompute.eurac.edu/openEO_WCPS_Driver/openeo/data) openEO API 0.0.2 POC

2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)
   a. csw
   b. uses openEO minimum information requirements from POC, extracted from WCS 2.0 GML data (getCoverageDescription), which could also be exposed completely if desired.

3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)
   a. YES
   b. NO

4. Can you provide an example query, and point to the documentation and/or API specification?
   a. NO
   b. http://saocompute.eurac.edu/openEO_WCPS_Driver/openeo/data/S2_L2A_T32TPS_10M

5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?
   a. YES
   b. NO

6. Does the back-end provide a structured way to obtain metadata about available processes?
   a. NO
   b. YES
      - (http://saocompute.eurac.edu/openEO_WCPS_Driver/openeo/processes)
      - (http://saocompute.eurac.edu/openEO_WCPS_Driver/openeo/processes/max_time)
7. If so, which standards does it use for this (if any)?

   openEO api 0.0.2 POC
8. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   N/A

A.6. Google Earth Engine

1. Does the back-end provide a public endpoint (URL) that can be queried to obtain metadata about available datasets (granules, image collections)?

   Yes, however it requires authentication (for no reason other than the whole API requires it). https://earthengine.googleapis.com/api/search?q=*

   This end-point doesn't return a lot of detail about each dataset as its primary purpose is keyword search. Details (like granules) about individual datasets are accessed through the /value servlet (see below).
2. If so, which standards are used for this (if any)? (e.g. CSW, OpenSearch, WCS; mention versions)

   OAuth2 for authentication. Non-standard JSON response (JSON with custom schema).
3. Can, in the request for metadata, filters be applied, and if yes on which criteria (e.g. bounding box, time, max. cloud cover)

   At the top level (/search), you can only filter on keywords and data type (raster or vector). For an individual dataset, you can filter on any metadata to find specific images/granules.
4. Can you provide an example query, and point to the documentation and/or API specification?

   All elevation rasters: https://code.earthengine.google.com/search/rasters?q=elevation

   This is an internal search API, which means we don't currently expose documentation for it externally.
5. Does the back-end provide descriptions of image collections (e.g. L8, Sentinel 1)?

   Yes. Image collections are included in the /search endpoint, but you can also query them directly via a /value call.

6. Does the back-end provide a structured way to obtain metadata about available processes?

   Yes. https://earthengine.googleapis.com/api/algorithms

   If so, which standards does it use for this (if any)?

   OAuth2. Custom JSON response.

7. Does the back-end provide access to any other metadata, possibly using standards, and if so, which ones and how?

   No.

B. Example service queries

B.1. CSW

To get the number of records, use GET (e.g. by clicking) on: http://pycsw.eodc.eu/?service=CSW&version=2.0.2&request=GetRecords&typenames=csw:Record&elementsetname=brief
this returns as response:

```
<csw:GetRecordsResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2"
    xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dct="http://purl.org/
        dc/terms/"
    xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gml="http://www.
        opengis.net/gml"
    xmlns:ows="http://www.opengis.net/ows" xmlns:xs="http://www.w3.org/2001/
        XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0.2"
    xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2 http://schemas.
        opengis.net/csw/2.0.2/CSW-discovery.xsd">
<csw:SearchStatus timestamp="2018-03-12T21:04:29Z" />
<csw:SearchResults nextRecord="1001" numberOfRecordsMatched="3447007"
   numberOfRecordsReturned="1000" recordSchema="http://www.opengis.net/cat/
       csw/2.0.2"
   elementSet="brief" />
</csw:GetRecordsResponse>
```

As a second example, we query all records younger than 2018-02-02:
https://csw.eodc.eu/?service=CSW&version=2.0.2&request=GetRecords&typenames=csw:Record&elementSetName=full&resultType=results&constraintLanguage=CQL_TEXT&startposition=1&constraint=apiso:TempExtent_begin%20%3E%20%272018-02-02%27 returns the first 1000 records of around 200,000, the first one of which is

```
<csw:Record>
<dc:identifier>S1B_IW_GRDH_1SDV_20180202T150020_20180202T150045_009446_010FB8_7123</dc:identifier>
<dc:title>S1B IW GRDH 1SDV 20180202T150020 20180202T150045 009446 010FB8 7123</dc:title>
<dc:type>dataset</dc:type>
<dc:subject>Sentinel-1</dc:subject>
<dc:subject>Sentinel-1B</dc:subject>
<dc:subject>GRD</dc:subject>
<dc:subject>ascending</dc:subject>
<dc:subject>Synthetic Aperture Radar</dc:subject>
<dc:subject>SAR</dc:subject>
<dc:subject>High Resolution</dc:subject>
<dc:subject>HR</dc:subject>
<dc:subject>Interferometric Wide swath</dc:subject>
<dc:subject>IW</dc:subject>
<dc:subject>VV+VH</dc:subject>
<dc:subject>s1b_csar_grdh_iw</dc:subject>
```

```
<dc:subject scheme="http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_TopicCategoryCode">
      imageryBaseMapsEarthCover</dc:subject>
<dc:format>SAFE</dc:format>
<dct:references scheme="offlineAccess">
/eodc/products/copernicus.eu/s1b_csar_grdh_iw/2018/02/02/S1B_IW_GRDH_1SDV_20180202T150020_20180202T150045_009446_010FB8_7123.
      zip
</dct:references>
<dc:relation />
<dct:modified>2018-02-02</dct:modified>
<dct:abstract>
This dataset contains a High Resolution (HR) Sentinel-1 Interferometric Wide swath (IW) Level-1 Ground Range Detected (GRD)
      product. It was obtained by Synthetic Aperture Radar (SAR) on board Sentinel-1B, in VV+VH polarisation and ascending
      orbit. Sentinel-1 is an imaging radar mission providing continuous all-weather, day-and-night imagery at C-band. The
      Sentinel-1 constellation provides high reliability, improved revisit time, geographical coverage and rapid data
      dissemination to support operational applications in the priority areas of marine monitoring, land monitoring and
      emergency services.
</dct:abstract>
<dc:date>2018-02-02T19:29:17Z</dc:date>
<dc:creator>ESA</dc:creator>
<dc:publisher>ESA</dc:publisher>
<dc:language>eng</dc:language>
<dc:rights>otherRestrictions</dc:rights>
<dct:alternative>S1B_IW_GRDH_1SDV_20180202T150020_20180202T150045_009446_010FB8_7123</dct:alternative>
<dct:spatial scheme="http://www.opengis.net/def/crs">http://www.opengis.net/def/crs/EPSG/0/4326</dct:spatial>
<ows:BoundingBox crs="urn:x-ogc:def:crs:EPSG:6.11:4326" dimensions="2"></csw:Record>
```

B.2. OpenSearch

B.2.1. VITO

The OpenSearch description can be found at http://www.vito-eodata.be/openSearch_all/description.xml. The URL template to find collections is http://www.vito-eodata.be/openSearch_all/findCollections.atom?lineage={eo:lineage?}&parentIdentifier={eo:parentIdentifier?}&language={eo:language?}&specificationdateType={eo:specificationdateType?}&instrument={eo:instrument?}&title={eo:title?}&resolution={eo:resolution?}&otherConstraint={eo:otherConstraint?}&platform={eo:platform?}&distanceUOM={eo:distanceUOM?}&uid={geo:uid?}&specificationTitle={eo:specificationTitle?}&sensorType={eo:sensorType?}&useLimitation={eo:useLimitation?}&keyword={eo:keyword?}&productType={eo:productType?}&organizationName={eo:organizationName?}&degree={eo:degree?}&topicCategory={eo:topicCategory?}&abstract={eo:abstract?}&classification={eo:classification?}&denominator={eo:denominator?}&organizationRole={eo:organizationRole?}&distanceValue={eo:distanceValue?}&accessConstraint={eo:accessConstraint?}&specificationDate={eo:specificationDate?}&compositeType={eo:compositeType?}&httpAccept={sru:httpAccept?}&recordSchema={sru:recordSchema?}&startIndex={os:startIndex?}&count={os:count?}&startPage={os:startPage?}&clientId={referrer:source?}&q={os:searchTerms?}&start={time:start?}&end={time:end?}&bbox={geo:box?}&geometry={geo:geometry?}&name={geo:name?}. It uses the OpenSearch extensions: eo, geo, sru, referrer, time.

Example results for an OpenSearch query for the keyword ”FAPAR“. Response from a GET of http://www.vito-eodata.be/openSearch_all/findCollections.atom?q=fapar:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:feed xmlns:atom="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:georss="http://www.
      georss.org/georss/11" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:score="http://a9.com/-/opensearch/extensions/
      relevance/1.0/" xmlns:media="http://tools.search.yahoo.com/mrss/" xmlns:ns1="http://purl.org/dc/terms/" xmlns:xhtml="
      http://www.w3.org/1999/xhtml" xmlns:param="http://a9.com/-/spec/opensearch/extensions/parameters/1.0/" xmlns:os="http://
      a9.com/-/spec/opensearch/1.1/" xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/" xmlns:geo="http://a9.com/-/
      opensearch/extensions/geo/1.0/" xmlns:eo="http://a9.com/-/opensearch/extensions/eo/1.0/" xmlns:vito="http://www.opengis.
      net/vito/1.0" xmlns:sru="http://a9.com/-/opensearch/extensions/sru/2.0/" xmlns:esipdiscovery="http://commons.esipfed.org
      /ns/discovery/1.2/" xmlns:referrer="http://www.opensearch.org/Specifications/OpenSearch/Extensions/Referrer/1.0"
      esipdiscovery:version="1.2">
    <atom:title>Search results feed for root Collection.</atom:title>
    <atom:subtitle>Number of results: 1</atom:subtitle>
    <atom:id>http://www.vito-eodata.be/openSearch_all/findCollections.atom</atom:id>
    <atom:generator uri="http://www.vito-eodata.be/openSearch_all/findCollections.atom?q=fapar" version="0.1">root collection<
        /atom:generator>
    <atom:updated>2018-03-15T11:08:16.768Z</atom:updated>
    <os:totalResults>1</os:totalResults>
    <os:startIndex>1</os:startIndex>
    <os:itemsPerPage>1</os:itemsPerPage>
    <os:Query role="request" os:searchTerms="fapar" os:count="20"/>
```

```
    <atom:link href="http://www.vito-eodata.be/openSearch_all/findCollections.atom?q=fapar" rel="self" type="application/atom+
        xml" title="self"/>
    <atom:link href="http://www.vito-eodata.be/openSearch_all/description.xml" rel="search" type="application/
        opensearchdescription+xml" title="Collection description"/>
    <atom:link href="http://www.vito-eodata.be/openSearch_all/findCollections.atom?q=fapar&amp;startIndex=1" rel="first" type=
        "application/atom+xml" title="first"/>
    <atom:link href="http://www.vito-eodata.be/openSearch_all/findCollections.atom?q=fapar&amp;startIndex=1" rel="last" type="
        application/atom+xml" title="last"/>
    <atom:entry>
        <atom:title>Sentinel-2 FAPAR</atom:title>
        <atom:id>urn:eop:VITO:CGS_S2_FAPAR</atom:id>
        <dc:identifier>urn:eop:VITO:CGS_S2_FAPAR</dc:identifier>
        <atom:published>2015-06-30T23:00:00.0Z</atom:published>
        <atom:updated>2018-12-31T00:00:00Z</atom:updated>
        <atom:summary>Collection of CGS_S2_FAPAR products</atom:summary>
        <atom:link href="http://www.vito-eodata.be/collections/srv/eng/main.home?uuid=urn:eop:VITO:CGS_S2_FAPAR" rel="
            describedBy" type="text/html"/>
        <atom:link href="http://pdfintegration.vgt.vito.be/PDF/image/defaultimages/Sentinel_2_Belgium.jpg" rel="icon" type="
            image/jpeg" title="Illustration image"/>
        <atom:link href="http://www.vito-eodata.be/openSearch_all/description.xml?collection=urn:eop:VITO:CGS_S2_FAPAR" rel="
            search" type="application/opensearchdescription+xml"/>
        <atom:link href="http://www.vito-eodata.be/collections/srv/eng/xml_iso19139?uuid=urn:eop:VITO:CGS_S2_FAPAR" rel="via"
            type="application/vnd.iso.19139+xml" title="Collection Catalogue Entry"/>
    </atom:entry>
</atom:feed>
```

B.2.2. Sinergise Sentinel Hub

The OpenSearch description for Sentinel 2 collections can be found at http://opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/describe.xml. The URL template to find collections is http://opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/search.atom?q={searchTerms?}&maxRecords={count?}&index={startIndex?}&page={startPage?}&lang={language?}&identifier={geo:uid?}&geometry={geo:geometry?}&box={geo:box?}&name={geo:name?}&lon={geo:lon?}&lat={geo:lat?}&radius={geo:radius?}&startDate={time:start?}&completionDate={time:end?}&parentIdentifier={eo:parentIdentifier?}&productType={eo:productType?}&processingLevel={eo:processingLevel?}&platform={eo:platform?}&instrument={eo:instrument?}&resolution={eo:resolution?}&organisationName={eo:organisationName?}&orbitNumber={eo:orbitNumber?}&sensorMode={eo:sensorMode?}&cloudCover={eo:cloudCover?}&snowCover={eo:snowCover?}&cultivatedCover={resto:cultivatedCover?}&desertCover={resto:desertCover?}&floodedCover={resto:floodedCover?}&forestCover={resto:forestCover?}&herbaceousCover={resto:herbaceousCover?}&iceCover={resto:iceCover?}&urbanCover={resto:urbanCover?}&waterCover={resto:waterCover?}&updated={dc:date?}&publishedAfter={resto:publishedAfter?}&publishedBefore={resto:publishedBefore?}&sortParam={resto:sortParam?}&sortOrder={resto:sortOrder?}&spacecraft={resto:spacecraft?}&sgsId={resto:sgsId?}. It uses the OpenSearch extensions: geo, time, eo, resto.

Example results for an OpenSearch query without search term, but with a cloud coverage of 0% and orbit number 14201 at the location 94.89 18.08. Response from a GET of http://opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/search.atom?cloudCover=0&orbitNumber=14201&lat=18.08&lon=94.89:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xml:lang="en" xmlns="http://www.w3.org/2005/Atom" xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/" xmlns:os=
    "http://a9.com/-/spec/opensearch/1.1/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:georss="http://www.georss.org/
    georss" xmlns:gml="http://www.opengis.net/gml" xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/" xmlns:eo="
    http://a9.com/-/opensearch/extensions/eo/1.0/" xmlns:metalink="urn:ietf:params:xml:ns:metalink" xmlns:xlink="http://www.
    w3.org/1999/xlink" xmlns:media="http://search.yahoo.com/mrss/" xmlns:om="http://www.opengis.net/om/2.0" xmlns:eop="http:
    //a9.com/-/opensearch/extensions/eo/1.0/" xmlns:opt="http://www.opengis.net/opt/2.1">
 <author>
  <name>Sentinel Hub</name>
  <email>info@sentinel-hub.com</email>
 </author>
 <title>resto</title>
 <subtitle type="html">2 results&amp;nbsp;|&amp;nbsp;_pagination</subtitle>
 <generator uri="http://github.com/jjrom/resto" version="2.3">resto</generator>
 <updated>2018-03-15T12:09:24Z</updated>
 <id>96d46663-7735-5ba4-9ed8-47eb9ffe8546</id>
 <link rel="self" title="self" type="application/atom+xml" href="http://opensearch.sentinel-hub.com/resto/api/collections/
     Sentinel2/search.atom?&amp;cloudCover=0&amp;orbitNumber=14201&amp;lat=18.08&amp;lon=94.89"/>
```

```
<link rel="search" title="OpenSearch Description Document" type="application/opensearchdescription+xml" href="http://
      opensearch.sentinel-hub.com/resto/api/collections/Sentinel2/describe.xml"/>
<os:totalResults>2</os:totalResults>
<os:startIndex>1</os:startIndex>
<os:itemsPerPage>2</os:itemsPerPage>
<os:Query role="request" eo:cloudCover="0" eo:orbitNumber="14201" geo:lat="18.08" geo:lon="94.89"/>
<entry>
 <title>S2A_OPER_MSI_L1C_TL_SGS__20180312T074221_A014201_T46QGE_N02.06</title>
 <updated>2018-03-15T10:11:16.439173Z</updated>
 <id>http://opensearch.sentinel-hub.com/resto/collections/Sentinel2/c8626cb7-154c-52bd-818a-76512d02499d.atom</id>
 <summary type="text">Sentinel-2/MSI acquired on 2018-03-12T04:22:28Z</summary>
 <link rel="self" type="application/atom+xml" title="ATOM link for c8626cb7-154c-52bd-818a-76512d02499d" href="http://
       opensearch.sentinel-hub.com/resto/collections/Sentinel2/c8626cb7-154c-52bd-818a-76512d02499d.atom?&amp;lang=en"/>
 <link rel="enclosure" type="text/html" length="0" title="File for c8626cb7-154c-52bd-818a-76512d02499d product"
       metalink:priority="50" href="http://sentinel-s2-l1c.s3-website.eu-central-1.amazonaws.com#tiles/46/Q/GE/2018/3/12/0/"/
       >
 <link rel="enclosure" type="text/directory" title="Download" href="s3://sentinel-s2-l1c/tiles/46/Q/GE/2018/3/12/0/"/>
 <dc:identifier>c8626cb7-154c-52bd-818a-76512d02499d</dc:identifier>
 <dc:date>2018-03-12T04:22:28Z/2018-03-12T04:22:28Z</dc:date>
 <opt:EarthObservation>
  <eop:metaDataProperty>
   <eop:EarthObservationMetaData>
    <eop:productType>S2MSI1C</eop:productType>
    <eop:processing>
     <eop:ProcessingInformation>
      <eop:processingLevel>1C</eop:processingLevel>
     </eop:ProcessingInformation>
    </eop:processing>
   </eop:EarthObservationMetaData>
  </eop:metaDataProperty>
  <om:procedure>
   <eop:EarthObservationEquipment>
    <eop:platform>
     <eop:Platform>
      <eop:shortName>Sentinel</eop:shortName>
      <eop:serialIdentifier>2</eop:serialIdentifier>
     </eop:Platform>
    </eop:platform>
    <eop:instrument>
     <eop:Instrument>
      <eop:shortName>MSI</eop:shortName>
     </eop:Instrument>
    </eop:instrument>
    <eop:sensor>
     <eop:Sensor>
      <eop:sensorType>OPTICAL</eop:sensorType>
      <eop:resolution uom="m">10</eop:resolution>
     </eop:Sensor>
    </eop:sensor>
    <eop:acquisitionParameters>
     <eop:Acquisition>
      <eop:orbitNumber>14201</eop:orbitNumber>
     </eop:Acquisition>
    </eop:acquisitionParameters>
    <eop:platform>
     <eop:Platform>
      <eop:shortName>Sentinel-2</eop:shortName>
     </eop:Platform>
    </eop:platform>
   </eop:EarthObservationEquipment>
  </om:procedure>
  <om:phenomenonTime>
   <gml:validTime>
    <gml:TimePeriod>
     <gml:beginPosition>2018-03-12T04:22:28Z</gml:beginPosition>
     <gml:endPosition>2018-03-12T04:22:28Z</gml:endPosition>
    </gml:TimePeriod>
   </gml:validTime>
  </om:phenomenonTime>
  <om:result>
   <opt:EarthObservationResult>
    <opt:cloudCoverPercentage>0</opt:cloudCoverPercentage>
   </opt:EarthObservationResult>
  </om:result>
 </opt:EarthObservation>
 <s3Path>tiles/46/Q/GE/2018/3/12/0</s3Path>
 <georss:where>
  <gml:MultiPolygon>
   <gml:polygonMember>
    <gml:Polygon>
     <gml:exterior>
      <gml:LinearRing>
       <gml:posList srsDimensions="2">18.079810667 94.889446073 18.067585364 95.878944351 17.762276205 95.80471211
            17.466004584 95.732750546 17.079417319 95.638800455 17.087883781 94.879166833 18.079810667 94.889446073</
            gml:posList>
      </gml:LinearRing>
     </gml:exterior>
    </gml:Polygon>
   </gml:polygonMember>
```

```
    </gml:MultiPolygon>
   </georss:where>
  </entry>
  <entry>
   <title>S2A_OPER_MSI_L1C_TL_SGS__20180312T074221_A014201_T46QGF_N02.06</title>
   <updated>2018-03-15T09:53:39.5368Z</updated>
   <id>http://opensearch.sentinel-hub.com/resto/collections/Sentinel2/acade6d7-9029-5528-9bcf-e0ab1f16fe15.atom</id>
   <summary type="text">Sentinel-2/MSI acquired on 2018-03-12T04:22:28Z</summary>
   <link rel="self" type="application/atom+xml" title="ATOM link for acade6d7-9029-5528-9bcf-e0ab1f16fe15" href="http://
         opensearch.sentinel-hub.com/resto/collections/Sentinel2/acade6d7-9029-5528-9bcf-e0ab1f16fe15.atom?&amp;lang=en"/>
   <link rel="enclosure" type="text/html" length="0" title="File for acade6d7-9029-5528-9bcf-e0ab1f16fe15_product"
         metalink:priority="50" href="http://sentinel-s2-l1c.s3-website.eu-central-1.amazonaws.com#tiles/46/Q/GF/2018/3/12/0/"/
         >
   <link rel="enclosure" type="text/directory" title="Download" href="s3://sentinel-s2-l1c/tiles/46/Q/GF/2018/3/12/0/"/>
   <dc:identifier>acade6d7-9029-5528-9bcf-e0ab1f16fe15</dc:identifier>
   <dc:date>2018-03-12T04:22:28Z/2018-03-12T04:22:28Z</dc:date>
   <opt:EarthObservation>
    <eop:metaDataProperty>
     <eop:EarthObservationMetaData>
      <eop:productType>S2MSI1C</eop:productType>
      <eop:processing>
       <eop:ProcessingInformation>
        <eop:processingLevel>1C</eop:processingLevel>
       </eop:ProcessingInformation>
      </eop:processing>
     </eop:EarthObservationMetaData>
    </eop:metaDataProperty>
    <om:procedure>
     <eop:EarthObservationEquipment>
      <eop:platform>
       <eop:Platform>
        <eop:shortName>Sentinel</eop:shortName>
        <eop:serialIdentifier>2</eop:serialIdentifier>
       </eop:Platform>
      </eop:platform>
      <eop:instrument>
       <eop:Instrument>
        <eop:shortName>MSI</eop:shortName>
       </eop:Instrument>
      </eop:instrument>
      <eop:sensor>
       <eop:Sensor>
        <eop:sensorType>OPTICAL</eop:sensorType>
        <eop:resolution uom="m">10</eop:resolution>
       </eop:Sensor>
      </eop:sensor>
      <eop:acquisitionParameters>
       <eop:Acquisition>
        <eop:orbitNumber>14201</eop:orbitNumber>
       </eop:Acquisition>
      </eop:acquisitionParameters>
      <eop:platform>
       <eop:Platform>
        <eop:shortName>Sentinel-2</eop:shortName>
       </eop:Platform>
      </eop:platform>
     </eop:EarthObservationEquipment>
    </om:procedure>
    <om:phenomenonTime>
     <gml:validTime>
      <gml:TimePeriod>
       <gml:beginPosition>2018-03-12T04:22:28Z</gml:beginPosition>
       <gml:endPosition>2018-03-12T04:22:28Z</gml:endPosition>
      </gml:TimePeriod>
     </gml:validTime>
    </om:phenomenonTime>
    <om:result>
     <opt:EarthObservationResult>
      <opt:cloudCoverPercentage>0</opt:cloudCoverPercentage>
     </opt:EarthObservationResult>
    </om:result>
   </opt:EarthObservation>
   <s3Path>tiles/46/Q/GF/2018/3/12/0</s3Path>
   <georss:where>
    <gml:MultiPolygon>
     <gml:polygonMember>
      <gml:Polygon>
       <gml:exterior>
        <gml:LinearRing>
         <gml:posList srsDimensions="2">18.98276638 94.899397254 18.969138769 95.941512903 18.276520637 95.92961585
               18.206671564 95.912853011 17.979073786 95.857389163 17.990937442 94.888497472 18.98276638 94.899397254</
               gml:posList>
        </gml:LinearRing>
       </gml:exterior>
      </gml:Polygon>
     </gml:polygonMember>
    </gml:MultiPolygon>
   </georss:where>
  </entry>
```

```
</feed>
```

B.2.3. EODC

The OpenSearch description can be found at https://csw.eodc.eu/?mode=opensearch&service=CSW&version=3.0.0&request=GetCapabilities. The URL template to find collections is https://csw.eodc.eu?mode=opensearch&service=CSW&version=3.0.0&request=GetRecords&elementsetname=full&typenames=csw:Record&resulttype=results&q={searchTerms?}&bbox={geo:box?}&time={time:start?}/{time:end?}&outputformat=application/atom%2Bxml&startposition={startIndex?}&maxrecords={count?}&recordids={geo:uid}. It uses the OpenSearch extensions: geo, time.

Example results for an OpenSearch query for the keyword ”Sentinel-2“ with a maximum of two records younger than March 2018. Response from a GET of https://csw.eodc.eu/?mode=opensearch&service=CSW&version=3.0.0&request=GetRecords&elementsetname=full&typenames=csw:Record&resulttype=results&q=Sentinel-2&outputformat=application/atom%2Bxml&time=2018-03-01/&maxrecords=2:

```
<?xml version="1.0" encoding="UTF-8"?>
<atom:feed xmlns:atom="http://www.w3.org/2005/Atom" xmlns:os="http://a9.com/-/spec/opensearch/1.1/">
  <atom:id>https://csw.eodc.eu</atom:id>
  <atom:title>EODC Data Catalogue</atom:title>
  <atom:author>
    <atom:name>EODC</atom:name>
  </atom:author>
  <atom:link href="https://csw.eodc.eu?mode=opensearch&amp;service=CSW&amp;version=3.0.0&amp;request=GetCapabilities" rel="
      search" type="application/opensearchdescription+xml" />
  <atom:updated>2018-03-15T10:43:36Z</atom:updated>
  <os:Query role="request" />
  <os:totalResults>72002</os:totalResults>
  <os:startIndex>1</os:startIndex>
  <os:itemsPerPage>2</os:itemsPerPage>
  <atom:entry xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:georss="http://www.georss.org/georss" xmlns:gml="http://www.
      opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2005/Atom
      http://www.kbcafe.com/rss/atom.xsd.xml">
    <atom:author>
      <atom:name>ESA</atom:name>
    </atom:author>
    <atom:category term="Sentinel-2" />
    <atom:category term="Sentinel-2A" />
    <atom:category term="descending" />
    <atom:category term="Multi-Spectral Instrument" />
    <atom:category term="MSI" />
    <atom:category term="s2a_prd_msil1c" />
    <atom:id>S2A_MSIL1C_20180228T000611_N0206_R016_T59VND_20180228T015123</atom:id>
    <dc:identifier>S2A_MSIL1C_20180228T000611_N0206_R016_T59VND_20180228T015123</dc:identifier>
    <atom:link href="/eodc/products/copernicus.eu/s2a_prd_msil1c/2018/02/28/
        S2A_MSIL1C_20180228T000611_N0206_R016_T59VND_20180228T015123.zip" title="Path to the file location on the EODC
        archive" type="offlineAccess" />
    <atom:link href="https://csw.eodc.eu?service=CSW&amp;version=2.0.2&amp;request=GetRepositoryItem&amp;id=
        S2A_MSIL1C_20180228T000611_N0206_R016_T59VND_20180228T015123" />
    <atom:title>S2A MSIL1C 20180228T000611 N0206 R016 T59VND 20180228T015123</atom:title>
    <atom:updated>2018-03-01</atom:updated>
    <atom:published>2018-03-01</atom:published>
    <atom:rights>otherRestrictions</atom:rights>
    <atom:summary>This dataset contains a Sentinel-2 Multi-Spectral Instrument (MSI) Level-1C product (descending orbit), with
        a cloud coverage of 100.0\%. Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission,
        supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well
        as observation of inland waterways and coastal areas.</atom:summary>
    <georss:where>
      <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <gml:lowerCorner>57.3626141979 170.999664053</gml:lowerCorner>
        <gml:upperCorner>57.7422981528 171.259802549</gml:upperCorner>
      </gml:Envelope>
    </georss:where>
  </atom:entry>
  <atom:entry xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:georss="http://www.georss.org/georss" xmlns:gml="http://www.
      opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2005/Atom
      http://www.kbcafe.com/rss/atom.xsd.xml">
    <atom:author>
      <atom:name>ESA</atom:name>
    </atom:author>
    <atom:category term="Sentinel-2" />
    <atom:category term="Sentinel-2A" />
    <atom:category term="descending" />
    <atom:category term="Multi-Spectral Instrument" />
    <atom:category term="MSI" />
```

```
    <atom:category term="s2a_prd_msil1c" />
    <atom:id>S2A_MSIL1C_20180228T002701_N0206_R016_T55KBP_20180228T015456</atom:id>
    <dc:identifier>S2A_MSIL1C_20180228T002701_N0206_R016_T55KBP_20180228T015456</dc:identifier>
    <atom:link href="/eodc/products/copernicus.eu/s2a_prd_msil1c/2018/02/28/
        S2A_MSIL1C_20180228T002701_N0206_R016_T55KBP_20180228T015456.zip" title="Path␣to␣the␣file␣location␣on␣the␣EODC␣
        archive" type="offlineAccess" />
    <atom:link href="https://csw.eodc.eu?service=CSW&amp;version=2.0.2&amp;request=GetRepositoryItem&amp;id=
        S2A_MSIL1C_20180228T002701_N0206_R016_T55KBP_20180228T015456" />
    <atom:title>S2A MSIL1C 20180228T002701 N0206 R016 T55KBP 20180228T015456</atom:title>
    <atom:updated>2018-03-01</atom:updated>
    <atom:published>2018-03-01</atom:published>
    <atom:rights>otherRestrictions</atom:rights>
    <atom:summary>This dataset contains a Sentinel-2 Multi-Spectral Instrument (MSI) Level-1C product (descending orbit), with
        a cloud coverage of 0.0\%. Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission, supporting
        Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as
        observation of inland waterways and coastal areas.</atom:summary>
    <georss:where>
      <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <gml:lowerCorner>-24.4900892108 144.040227075</gml:lowerCorner>
        <gml:upperCorner>-23.4823186052 145.137169091</gml:upperCorner>
      </gml:Envelope>
    </georss:where>
  </atom:entry>
</atom:feed>
```

B.3. WCS

Coverage description: response from a GET of https://proba-v-mep.esa.int/applications/geo-viewer/app/geoserver/ows?SERVICE=WCS&REQUEST=DescribeCoverage&CoverageId=PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY&version=2.0.0

```
<wcs:CoverageDescriptions xmlns:wcs="http://www.opengis.net/wcs/2.0" xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:gml="
    http://www.opengis.net/gml/3.2" xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0" xmlns:xlink="http://www.w3.org/1999/
    xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:wcsgs="
    http://www.geoserver.org/wcsgs/2.0" xmlns:wcseo="http://www.opengis.net/wcseo/1.0" xmlns:eop="http://www.opengis.net/eop
    /2.0" xmlns:om="http://www.opengis.net/om/2.0" xsi:schemaLocation="http://www.opengis.net/wcs/2.0␣http://schemas.opengis
    .net/wcs/2.0/wcsDescribeCoverage.xsd␣http://www.opengis.net/wcseo/1.0␣https://proba-v-mep.esa.int/applications/geo-
    viewer/app/geoserver/schemas/wcseo/1.0/wcsEOCoverage.xsd">
<wcs:CoverageDescription gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY">
<gml:boundedBy>
<gml:EnvelopeWithTimePeriod srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels="Lat␣Long␣time" uomLabels="Deg␣Deg
    ␣s" srsDimension="2">
<gml:lowerCorner>-64.991 -180.0</gml:lowerCorner>
<gml:upperCorner>75.0 179.991</gml:upperCorner>
<gml:beginPosition>2013-10-11T00:00:00.000Z</gml:beginPosition>
<gml:endPosition>2018-03-01T00:00:00.000Z</gml:endPosition>
</gml:EnvelopeWithTimePeriod>
</gml:boundedBy>
<wcs:CoverageId>PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY</wcs:CoverageId>
<gml:coverageFunction>
<gml:GridFunction>
<gml:sequenceRule axisOrder="+2␣+1">Linear</gml:sequenceRule>
<gml:startPoint>0 0</gml:startPoint>
</gml:GridFunction>
</gml:coverageFunction>
<gmlcov:metadata>
<gmlcov:Extension>
<wcsgs:TimeDomain default="2018-03-01T00:00:00.000Z">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_0">
<gml:timePosition>2013-10-11T00:00:00.000Z</gml:timePosition>
</gml:TimeInstant>
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_1">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_2">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_3">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_4">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_5">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_6">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_7">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_8">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_9">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_10">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_11">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_12">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_13">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_14">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_15">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_16">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_17">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_18">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_19">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_20">
```

<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_21">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_22">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_23">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_24">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_25">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_26">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_27">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_28">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_29">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_30">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_31">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_32">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_33">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_34">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_35">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_36">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_37">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_38">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_39">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_40">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_41">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_42">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_43">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_44">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_45">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_46">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_47">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_48">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_49">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_50">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_51">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_52">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_53">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_54">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_55">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_56">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_57">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_58">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_59">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_60">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_61">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_62">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_63">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_64">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_65">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_66">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_67">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_68">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_69">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_70">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_71">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_72">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_73">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_74">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_75">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_76">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_77">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_78">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_79">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_80">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_81">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_82">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_83">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_84">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_85">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_86">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_87">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_88">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_89">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_90">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_91">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_92">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_93">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_94">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_95">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_96">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_97">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_98">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_99">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_100">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_101">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_102">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_103">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_104">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_105">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_106">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_107">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_108">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_109">

```
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_110">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_111">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_112">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_113">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_114">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_115">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_116">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_117">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_118">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_119">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_120">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_121">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_122">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_123">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_124">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_125">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_126">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_127">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_128">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_129">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_130">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_131">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_132">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_133">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_134">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_135">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_136">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_137">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_138">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_139">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_140">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_141">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_142">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_143">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_144">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_145">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_146">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_147">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_148">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_149">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_150">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_151">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_152">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_153">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_154">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_155">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_156">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_157">
<gml:TimeInstant gml:id="PV_MEP__PROBAV_S10_TOC_1KM_GEOMETRY_td_158">
</wcsgs:TimeDomain>
<wcseo:EOMetadata></gmlcov:Extension>
</gmlcov:metadata>
<gml:domainSet>
<gmlcov:rangeType>
<wcs:ServiceParameters></wcs:CoverageDescription>
</wcs:CoverageDescriptions>
```