

OGC Testbed-14
Swath Coverage Engineering Report

Table of Contents

1. Summary	4
1.1. Requirements & Research Motivation	4
1.2. Prior-After Comparison	4
1.3. Recommendations for Future Work	4
1.4. Document contributor contact points	5
1.5. Foreword	5
2. References	6
3. Terms and definitions	7
3.1. Abbreviated terms	7
4. Overview	9
5. Swath Coverage	10
5.1. Swath Coverage	10
5.2. Typical Types of Swath Coverages	10
5.2.1. Scanning Swath Coverage	10
5.2.2. Profiling Swath Coverage	11
5.2.3. Scanning-profiling Swath Coverage	13
5.3. Functional Requirements of Swath Coverage	13
5.3.1. Visualization	14
5.3.2. General 2D views	14
5.3.3. Time series views	15
5.3.4. Globe 2.5D/3D view	16
5.3.5. Analytical view	16
5.3.6. Analytics	16
6. Swath Coverage Models with Climate Forecast Conventions	18
6.1. Climate Forecast (CF) Conventions	18
6.2. Swath Coverage CF Coverage Models	18
6.2.1. VIIRS Swath Coverage	19
6.2.2. CloudSat Swath Coverage	25
6.2.3. MODIS Swath Coverage	26
6.2.4. CALIPSO Swath Coverage	27
7. Swath Coverage Models with OGC Coverage Implementation Schema	30
7.1. Coverage Implementation Schema (CIS)	30
7.2. Swath Coverage Models	30
7.3. CIS Modeling with Irregular Grid	31
7.3.1. MODIS MOD07 Swath Coverage	31
7.3.2. VIIRS Swath Coverage	33
7.3.3. CloudSat Swath Coverage	35
7.4. CIS Modeling with Regular Grid	37

7.4.1. VIIRS Swath Coverage	37
7.4.2. CloudSat Swath Coverage	39
7.4.3. CALIPSO Swath Coverage	41
8. Swath Coverage Encoding	43
8.1. Overview	43
8.2. GeoTIFF Encoding	43
8.3. GeoPackage Encoding	45
8.4. Encodings with GDAL drivers	46
9. WCS Swath Coverage Profile	51
9.1. Scope	51
9.2. Conformance	51
9.3. Swath Coverage Extension	51
9.4. Swath GeoTiff Coverage	52
9.5. Swath NetCDF Coverage	52
9.6. Swath GeoPackage Coverage	53
10. WCS REST API	54
10.1. Scope and Design Principle	54
10.2. Conformance	54
10.3. Main Paths	54
10.4. Service root path /	55
10.4.1. Get Operation	55
10.5. Service OpenAPI path /api	56
10.5.1. Get Operation	56
10.6. Service OpenAPI path /conformance	56
10.6.1. Get Operation	56
10.7. Service OpenAPI path /coverages	56
10.7.1. Get Operation	56
10.8. Service OpenAPI path /coverages/{coverageId}	57
10.8.1. Get Operation	57
10.9. Service OpenAPI path /coverages/{coverageId}/rangeSet	57
10.9.1. Get Operation	57
11. Swath Coverage Use Cases	59
11.1. Implementations of Swath WCS	59
11.1.1. CloudSat Level 2 Profile Swath Data	59
11.1.2. Modeling with CF Convention	61
11.1.3. CloudSat Swath WCS Implementation I	62
11.1.4. CloudSat Swath WCS Implementation II	67
11.1.5. VIIRS Swath Coverages and their Web Coverage Services	80
11.2. Implementations of Swath WCS Clients	80
11.2.1. CloudSat Profile Visualization	80
11.2.2. CloudSat High Dimension Viewers	82

11.2.3. Use Cases	83
12. Summary and Discussions	86
12.1. Summary	86
12.2. Discussions	86
12.2.1. Modeling Swath Coverages	86
12.2.2. WCS REST API for Swath Coverage	88
12.2.3. Visualization of Swath Coverage	89
12.3. Suggestions and Recommendations	89
12.4. Relation to other groups	89
Appendix A: Example Swath Coverage Services, their Requests and Responses	90
A.1. CloudSat Swath Coverage Service 1	90
A.1.1. Endpoints	90
A.1.2. Adjusted schema for swath coverage WCS	90
A.1.3. Example WCS requests and responses	95
A.2. CloudSat Profile WCS Implementation 2	107
A.2.1. Endpoints	107
A.2.2. Example WCS requests and responses	107
A.3. VIIRS VNP09 Swath Coverage Service WCS	110
A.3.1. Endpoints	110
A.3.2. Example WCS requests and responses	111
A.4. VIIRS VNP09 Swath Coverage Service WCS REST	112
A.4.1. Endpoints	112
A.4.2. Example WCS REST requests and responses	112
A.5. Swath Coverage Client	113
A.5.1. Client Implementation from MEEO	113
A.5.2. Examples	113
A.5.3. Client Implementation from EOX	114
Appendix B: WCS REST API Full Specification	116
B.1. Scope	116
B.2. Conformance	116
B.3. References	116
B.4. Terms and Definitions	117
B.4.1. coverage	117
B.4.2. dataset	117
B.4.3. dataset series	117
B.4.4. subsetting	117
B.4.5. trimming	117
B.4.6. slicing	117
B.5. Conventions	117
B.5.1. Base Standard	117
B.6. Core Requirement Class	118

B.6.1. Overview	118
B.6.2. OpenAPI Object	119
B.6.3. Service root path /	173
B.6.4. Service OpenAPI path /api	177
B.6.5. Service OpenAPI path /conformance	181
B.6.6. Service OpenAPI path /coverages	185
B.6.7. Service OpenAPI path /coverages/{coverageId}	188
B.6.8. Service OpenAPI path /coverages/{coverageId}/rangeSet	189
B.7. Range Subsetting Extension	192
B.7.1. Overview	192
B.7.2. OpenAPI Object	192
B.7.3. Service OpenAPI path /conformance	193
B.7.4. Service OpenAPI path /eocoverages/{eocoverageId}/rangeSet	193
B.8. Earth Observation Extension	194
B.8.1. Overview	194
B.8.2. OpenAPI Object	195
B.8.3. Service OpenAPI path /conformance	199
B.8.4. Service OpenAPI path /eocoverages	199
B.8.5. Service OpenAPI path /eocoverages/{eocoverageId}	200
B.8.6. Service OpenAPI path /eocoverages/{eocoverageId}/rangeSet	200
B.8.7. Service OpenAPI path /eocoverages/{eodataseriedId}	201
B.8.8. Service OpenAPI path /eocoverages/{eodataseriedId}/{eodatasetId}	202
B.8.9. Service OpenAPI path /eocoverages/{eodataseriedId}/{eodatasetId}/rangeSet	202
B.9. Transaction Insertion Deletion	203
B.9.1. Overview	203
B.9.2. OpenAPI Object	203
B.9.3. Service OpenAPI path /conformance	204
B.9.4. Service OpenAPI path /coverages/{coverageId}	205
B.9.5. Service OpenAPI path /coverages/{coverageId}/rangeSet	206
B.9.6. Service OpenAPI path /coverages/rangeSet	207
B.10. Transaction Update	207
B.10.1. Overview	207
B.10.2. OpenAPI Object	208
B.10.3. Service OpenAPI path /conformance	209
B.10.4. Service OpenAPI path /coverages/{coverageId}	209
B.10.5. Service OpenAPI path /coverages/{coverageId}/rangeSet	210
B.11. GeoTiff Coverage	210
B.11.1. Overview	210
B.11.2. OpenAPI Object	210
B.11.3. Service OpenAPI path /conformance	211
B.11.4. Service OpenAPI path /coverages/{coverageId}/rangeSet	211

B.12. NetCDF Coverage	212
B.12.1. Overview	212
B.12.2. OpenAPI Object	212
B.12.3. Service OpenAPI path /conformance	212
B.12.4. Service OpenAPI path /coverages/{coverageId}/rangeSet	213
B.13. GeoPackage Coverage	213
B.13.1. Overview	213
B.13.2. OpenAPI Object	213
B.13.3. Service OpenAPI path /conformance	214
B.13.4. Service OpenAPI path /coverages/{coverageId}/rangeSet	214
Appendix C: Revision History	216
Appendix D: Bibliography	218

Publication Date: 2019-02-07

Approval Date: 2018-12-13

Submission Date: 2018-11-08

Reference number of this document: OGC 18-047r3

Reference URL for this document: <http://www.opengis.net/doc/PER/t14-D007>

Category: Public Engineering Report

Editor: Eugene Genong Yu, Liping Di

Title: OGC Testbed-14: Swath Coverage Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright (c) 2019 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

This Engineering Report (ER) presents a summary, description and findings of the Swath Coverage task conducted by the OGC Testbed-14 initiative.

1.1. Requirements & Research Motivation

The goal of the Swath Coverage task was to improve access to National Aeronautics and Space Administration (NASA) data in Swath structure, specifically for those Earth Observations in Level 1 and 2.

The objectives of the Swath Coverage task were as follows:

- Modeling Swath data with Climate Forecast Convention
- Encodings in GeoTiff, NetCDF, and GeoPackage
- Serve typical swath data in WCS service
- Demonstrate the use of swath data through clients

1.2. Prior-After Comparison

The Swath Coverage has not been modeled and supported with existing standards and specifications. This OGC Testbed-14 task provided solutions through adapting the data models and services to better support access and visualization of swath coverages.

Achievements of the Swath Coverage Task

Items	Prior	After
Swath Coverage Model	Not modeled	Modeled by (1) Climate Forecast Convention and (2) CS 1.1
Swath Coverage Encoding	No standard encoding	Encodings: GeoTiff-Swath, NetCDF-Swath, GeoPackage-Swath
Swath Coverage Service	No support by WCS	WCS Swath Coverage Profile, WCS Swath Coverage REST API
Swath Coverage Client	Ad hoc	Standard interaction with WCS for data access and visualization

1.3. Recommendations for Future Work

The Swath Coverage is relevant to the standardization work of the Web Coverage Service (WCS) Standards Working Group (SWG). The task contributes to the development of open specifications on swath coverage as follows.

- WCS Swath Coverage Profile to adapt and specialize WCS and EO-WCS to model swath coverages
- Encoding of swath data in GeoTiff, NetCDF, and GeoPackage as supported formats under WCS Swath Coverage
- WCS Representational State Transfer (REST) Application Programming Interface (API) (OpenAPI 3.0.1) to facilitate stub code automatic generation for server and client

The ER makes the following recommendations for future testbed work:

- If the Geospatial Data Abstraction Library (GDAL) is still considered as the library to work on different swath coverage, the GPKG driver of GDAL needs to be enhanced to support non-tiled data. This could be done within an OGC testbed and then provided to the GDAL developer community as open source.
- Future testbeds could also look into how the Sensor Model Registry that was initiated in 2018 by the OGC Technical Committee could support the modeling of Swath Coverages. The Sensor Model Registry will be managed by the OGC Naming Authority.

1.4. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization
Eugene G. Yu	George Mason University
Liping Di	George Mason University
Stephan Meissl	EOX
Sergio Ferraresi	MEEO
Sizhe Wang	Arizona State University
Wenwen Li	Arizona State University
Ingo Simonis	OGC

1.5. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following normative documents are referenced in this document.

NOTE: Only normative standards are referenced here, e.g. OGC, ISO or other SDO standards. All other references are listed in the bibliography. Example:

- [OGC: OGC 06-121r9, OGC® Web Services Common Standard](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2]
- [OGC: OGC 17-089r1, OGC® Web Coverage Service \(WCS\) Interface Standard - Core, version 2.1](http://docs.opengeospatial.org/is/17-089r1/17-089r1.html) [http://docs.opengeospatial.org/is/17-089r1/17-089r1.html]
- [OGC: OGC 09-110r4, OGC® WCS 2.0 Interface Standard - Core, version 2.0.1](https://portal.opengeospatial.org/files/09-110r4) [https://portal.opengeospatial.org/files/09-110r4]
- [OGC: OGC 12-039, OGC® WCS Interface Standard – CRS Extension, version 1.0.0](https://portal.opengeospatial.org/files/54209) [https://portal.opengeospatial.org/files/54209]
- [OGC: OGC 09-147r3, OGC® WCS Interface Standard - KVP Protocol Binding Extension, version 1.0.1](https://portal.opengeospatial.org/files/09-147r3) [https://portal.opengeospatial.org/files/09-147r3]
- [OGC: OGC 10-140r2, OGC® WCS Interface Standard - Earth Observation Application Profile, version 1.1](http://docs.opengeospatial.org/is/10-140r2/10-140r2.html) [http://docs.opengeospatial.org/is/10-140r2/10-140r2.html]
- [OGC: OGC 09-146r6, Coverage Implementation Schema \("CIS"\)](http://docs.opengeospatial.org/is/09-146r6/09-146r6.html) [http://docs.opengeospatial.org/is/09-146r6/09-146r6.html]
- [OGC: OGC 12-100r1, OGC® GML Application Schema - Coverages - GeoTIFF Coverage Encoding Profile](https://portal.opengeospatial.org/files/?artifact_id=54813) [https://portal.opengeospatial.org/files/?artifact_id=54813]
- [OGC: OGC 14-100r2, CF-netCDF 3.0 encoding using GML Coverage Application Schema](http://docs.opengeospatial.org/is/14-100r2/14-100r2.html) [http://docs.opengeospatial.org/is/14-100r2/14-100r2.html]

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- swath coverage

unreferenced Earth Observation data that has the following characteristics of being time-ordered, having geolocation files with longitude and latitude at least, and having definite mapping relationship between geolocations and data fields.

- scanline | across-track axis

one of the two axes that a swath data has. In general, it is across the track of sensor movement.

- track | along-track axis

one of the two axes that a swath data has. In general, it is along the track of sensor movement.

- time

one of the dimensions often occurred in swath coverage. The common units are expressed in seconds, microseconds, or nanoseconds for sensor recordings. It can be explicitly expressed in Coordinated Universal Time (abbreviated to UTC) and International Atomic Time (TAI). UTC and TAI are different because UTC has leap seconds while TAI is continuous. UTC is based on Gregorian Calendar, the most commonly used calendar. In this context, if "gregorian" is the calendar, the time should be in UTC which has leap second. In computer, Unix epoch (or Unix time or POSIX time or Unix timestamp) may often be used which is the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT), not counting leap seconds (in ISO 8601: 1970-01-01T00:00:00Z). Proper library should be used to convert them into UTC or TAI.

3.1. Abbreviated terms

NOTE: The abbreviated terms clause gives a list of the abbreviated terms and the symbols necessary for understanding this document. All symbols should be listed in alphabetical order. Some more frequently used abbreviated terms are provided below as examples.

- CALIOP Cloud-Aerosol Lidar with Orthogonal Polarization
- CALIPSO Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations

- CIS Coverage Implementation Schema
- COM Component Object Model
- CORBA Common Object Request Broker Architecture
- COTS Commercial Off The Shelf
- DCE Distributed Computing Environment
- DCOM Distributed Component Object Model
- GDAL Geospatial Data Abstraction Library
- GIS Geographic Information System
- GMT Greenwich Mean Time
- IDL Interface Definition Language
- ISO International Organization for Standardization
- MODIS Moderate Resolution Imaging Spectroradiometer
- NASA National Aeronautics and Space Administration
- NOAA National Oceanic and Atmospheric Administration
- OGC Open Geospatial Consortium
- POSIX Portable Operating System Interface
- TAI International Atomic Time
- UTC Coordinated Universal Time
- VIIRS Visible Infrared Imaging Radiometer Suite

Chapter 4. Overview

Highlights

NOTE

This Chapter gives an overview of the complete document that helps the reader to better understand the various sections of the ER.

Section 5 introduces the problem of swath coverage access. It describes the situation prior to the testbed and discusses the requirements set by the sponsors.

Section 6 discusses the model using Climate Forecast Conventions. It provides recommendations on modeling typical types of swath coverages.

Section 7 discusses the model using OGC Coverage Implementation Schema. It provides recommendations on modeling typical types of swath coverages.

Section 8 presents other encodings or data models for Swath Coverage developed in this testbed in addition to netCDF and GML coverage.

Section 9 presents the Swath Coverage Profile under WCS. The Swath Coverage Profile is developed on CIS-1.1, WCS 1.1, and EO-WCS 1.1.

Section 10 presents the WCS REST API for Swath Coverage Profile. The version for OpenAPI is 3.0.1. The automatic generation of service and client stubs is supported with the full development of swath coverages under OpenAPI. The section only covers the design principles and major paths. Full details are included Appendix B.

Section 11 presents example implementations of Swath Coverage WCS services and clients, and typical use cases of Swath Coverages.

Section 12 provides a summary of the main findings and discusses links to other tasks such as WFS 3.0.

Annex A provides code snippets and examples that illustrate the functionality of the WCS Swath Coverage Service in requests and responses.

Annex B includes the complete draft specification for WCS REST API that follows OpenAPI 3.0.1. This is full document that is briefed in Section 10.

Chapter 5. Swath Coverage

Highlights

NOTE

This section explains some basic concepts for swath and swath coverage. Typical swath data are also described. General requirements are reviewed with the functional expectation of visualization and analytics.

5.1. Swath Coverage

In this context, a swath coverage is generally defined as the Earth Observation data that has the following characteristics [1] [2] [3]:

- They are time-ordered.
- There are geolocation files with longitude and latitude at least.
- There are definite mapping relationships between geolocations and data fields.

5.2. Typical Types of Swath Coverages

Swath coverages can be classified into different categories depending on the criteria. Based on the data fields, they may be grouped into two categories: radiometric swath data (Level 1) and geophysical swath data (Level 2) [4]. Based on observing geometry, they may be grouped into three categories: scanning (scan line), profiling (vertical), and scanning-profiling (curtain-like) [1], [4]. Depending on the mapping relationship between data fields and geolocations, they can be grouped into three: swath with time/Location, swath with Analytic fit, and swath with Attitude/Ephemeris data for spacecraft or airborne platforms [1].

5.2.1. Scanning Swath Coverage

A typical scanline swath coverage would look like the simplified sketch as shown in [Figure 1](#).

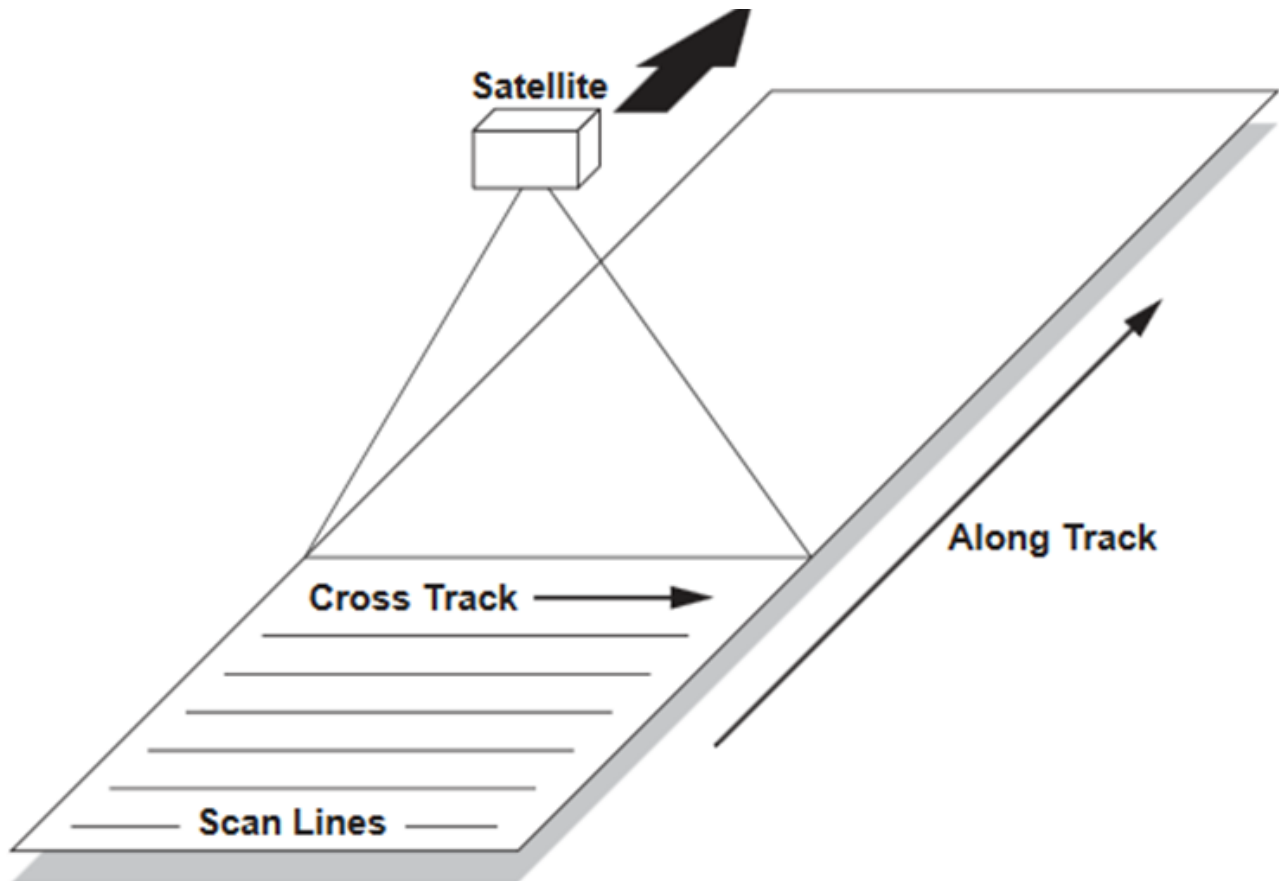


Figure 1. Scanning swath

In this testbed, the Level 1 and Level 2 products from the Visible Infrared Imaging Radiometer Suite (VIIRS) instrument are examples of scanning swath coverage datasets. The products of VIIRS are radiance at Level 1 and reflectance at Level 2 data. The product number is VNP09 for Suomi-NPP VIIRS produced by the Land Processes Distributed Active Archive Center (LP DAAC) of NASA. This is a typical scanline swath data. For the three I-Bands, the cross-track scan is expanded to 6400. For the M-Bands, the cross-track scan is expanded to 3200. The sensor has a constant scanning angular resolution while the footprint on the target Earth ground shows a "bow-tie" effect. To save the bandwidth in sending back the sensed data, certain tracks are skipped towards the tails along scanlines. Geolocation products are numbered VNP03 where VNP03IMGLL for I-Bands and VNP03MODLL for M-Bands. Three geolocation attributes are calculated for each pixel. They are latitude, longitude, and height. The mapping between the geolocation fields and data fields are one-to-one, i.e. each pixel has a unique geolocation value.

The operational products served at National Oceanic and Atmospheric Administration (NOAA) are numbered as SVI and SVM respectively. Each file has both radiance and reflectance data in one file. Geolocation files contain solar position (solar azimuth and solar zenith) and satellite position (satellite azimuth, satellite zenith, and satellite range) in addition to geometrical locations of latitude, longitude, and height. These parameters are useful in geometrical rectification and radiometric correction.

5.2.2. Profiling Swath Coverage

A typical profiling swath would look like these, as simply depicted in [Figure 2](#)

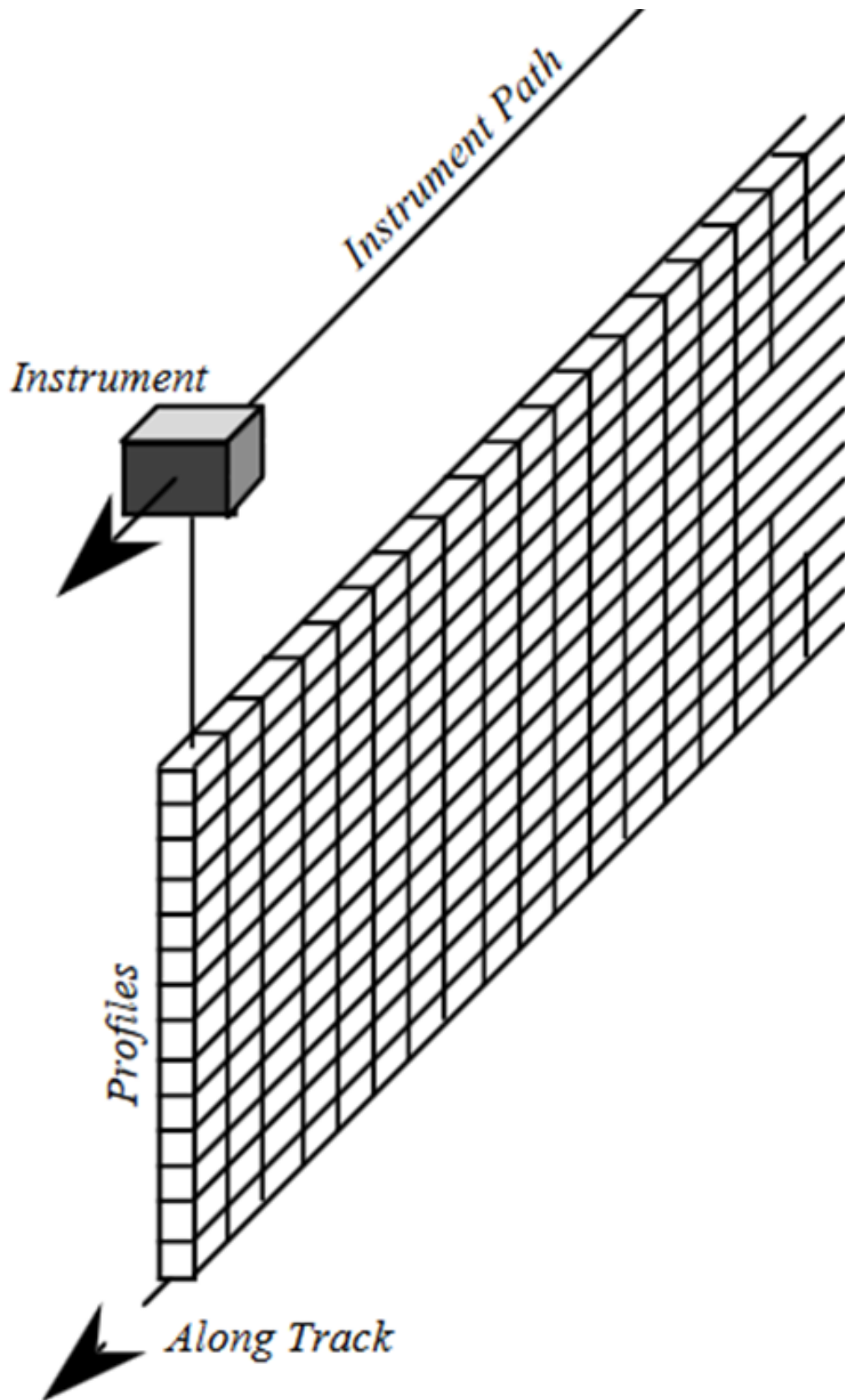


Figure 2. Profiling swath

The Level 1B and Level 2 backscatter products of Cloud-Aerosol LiDAR and Infrared Pathfinder Satellite Observations (CALIPSO) and the backscatter products of CloudSat are typical profiling swath coverages. The Level 1B of CALIOP (Cloud-Aerosol Lidar with Orthogonal Polarization) - the sensor of CALIPSO is a profiling swath coverage. The aggregation and averaging algorithm on board of the satellite convert the raw measurements into uniformly 583 elements vertically at each point.

There is no across-track scan. The resolution along the vertical direction varies from height to height due to the averaging scheme. The spatial resolution at each point is different from pixel to pixel due to its location at different record element (or altitude).

The level 1B and Level 2 product of CloudSat are also primarily backscatters of Radar and derived Cloud Geometric Profile respectively. They are also typical profiling swath coverage. At each point along the track, CloudSat has uniformly 125 bins which represents 240-meter vertical resolution. Each pixel has different spatial resolution due to their difference along vertical direction.

5.2.3. Scanning-profiling Swath Coverage

The scanning-profiling swath has both scan expansion with vertical profiles. To be general, this category of swath data includes those point clouds. It captures a volumetric description of the features.

A scanning-profiling product is the atmospheric profiling product from Moderate Resolution Imaging Spectroradiometer (MODIS). The product numbers are MOD07 and MYD07 for data collected on Terra and Aqua respectively. The MODIS Atmospheric Profile product consists of several parameters: they are total-ozone burden, atmospheric stability, temperature and moisture profiles, and atmospheric water vapor. All of these parameters are produced day and night for Level 2 at 5x5 1-km pixel resolution when at least 9 FOVs are cloud free. The MODIS Atmosphere Profile data product files are typical of scanning-profiling swath which profile data at different altitudes.

The MODIS total-ozone burden is an estimate of the total-column tropospheric and stratospheric ozone content. The MODIS atmospheric stability consists of three daily Level 2 atmospheric stability indices. The Total Totals (TT), the Lifted Index (LI), and the K index (K) are each computed using the infrared temperature- and moisture-profile data, also derived as part of MOD07.

The MODIS temperature and moisture profiles are produced at 20 vertical levels. A clear sky synthetic regression retrieval algorithm is used, where regression coefficients are derived by using a fast-radiative transfer model with atmospheric characteristics taken from a dataset of global (radiosonde and model) profiles. The MODIS atmospheric water-vapor product is an estimate of the total column water vapor made from integrated MODIS infrared retrievals of atmospheric moisture profiles in clear scenes.

5.3. Functional Requirements of Swath Coverage

The goal of this testbed effort was to better serve the swath coverage through a standard Web service that helps in extensively using swath data. To achieve the goal, the specific tasks included (1) modeling Swath data with Climate Forecast Convention, (2) encodings in GeoTiff, NetCDF, and GeoPackage, (3) serving typical swath data through a WCS service, and (4) demonstrating typical use cases of swath data through clients.

Two important aspects of using swath data are visualization and analytics. Visualization of the swath coverage in common GIS software packages allows visual exploration from different viewpoints or together with other features and properties in aligned geographical reference systems. Analytics include the use of actual data which puts a specific requirement on accessing the raw data as well as geospatially mapping to other properties and features.

5.3.1. Visualization

The visualization of the swath coverage may include a general 2D view, time series view, 3D view on a globe, or an analytical view.

5.3.2. General 2D views

The view of the swath coverage in 2D view may be shown by itself from different viewports or projections to a geographical reference system along with other geographical features. The view may be the result of sub-setting or slicing on a time dimension or (vertical) profile. Example views of CloudSat 2B-GEOPROF Level 2 (Swath) data are shown in Figure 3. The top shows the top view of the swath data over geographical regions in georeferenced projections. The lower view is a 2D view with axes of time and height (profile) along the track. The latter may not be projected in a standard spatial reference system, but the sensor coordinate system. Another example of a 2D profile view in a time-height coordinate system is shown in Figure 4.

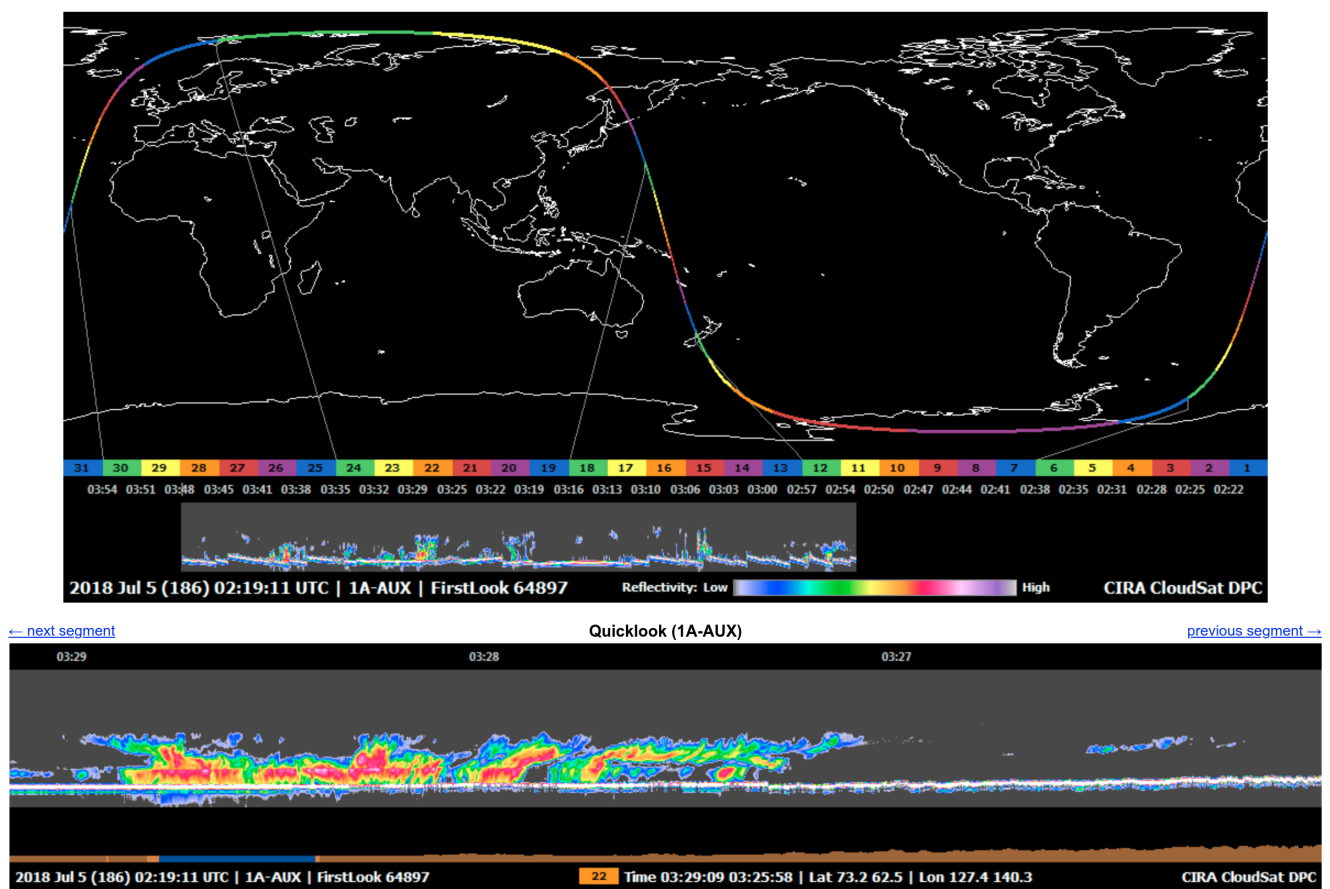


Figure 3. 2D Quickviews of CloudSat 2B-GEOPROF

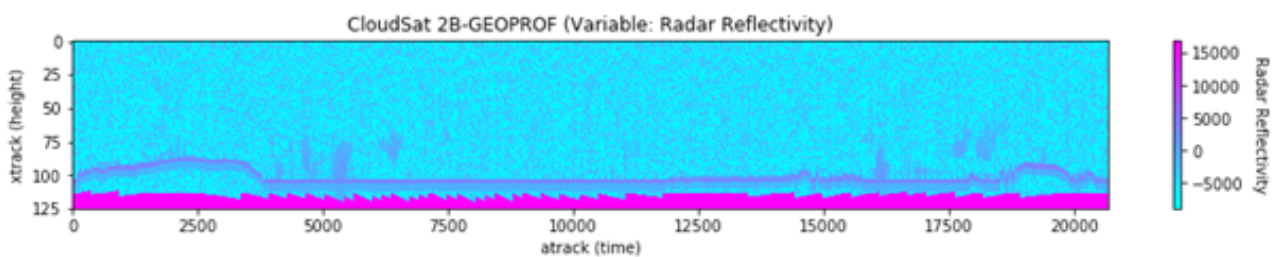


Figure 4. Profile View of CloudSat 2B-GEOPROF Radiance

5.3.3. Time series views

A swath coverage is a time-ordered observation of the ground sampled by satellite sensors. One of the requirements is to visualize the data with time series. This can be interactively controlled using a timeline or a timebar. One example time slider is shown in Figure 5 where the timeline is used to control the view of time series. In general, the timeline control allows the user to interactively zoom and pan in time as well as to jump to particular dates via the picker (top right corner of timeslider). While navigating the timeslider, the interface provides an indication of data availability (blue rectangles). Hovering over these indications shows the product id.

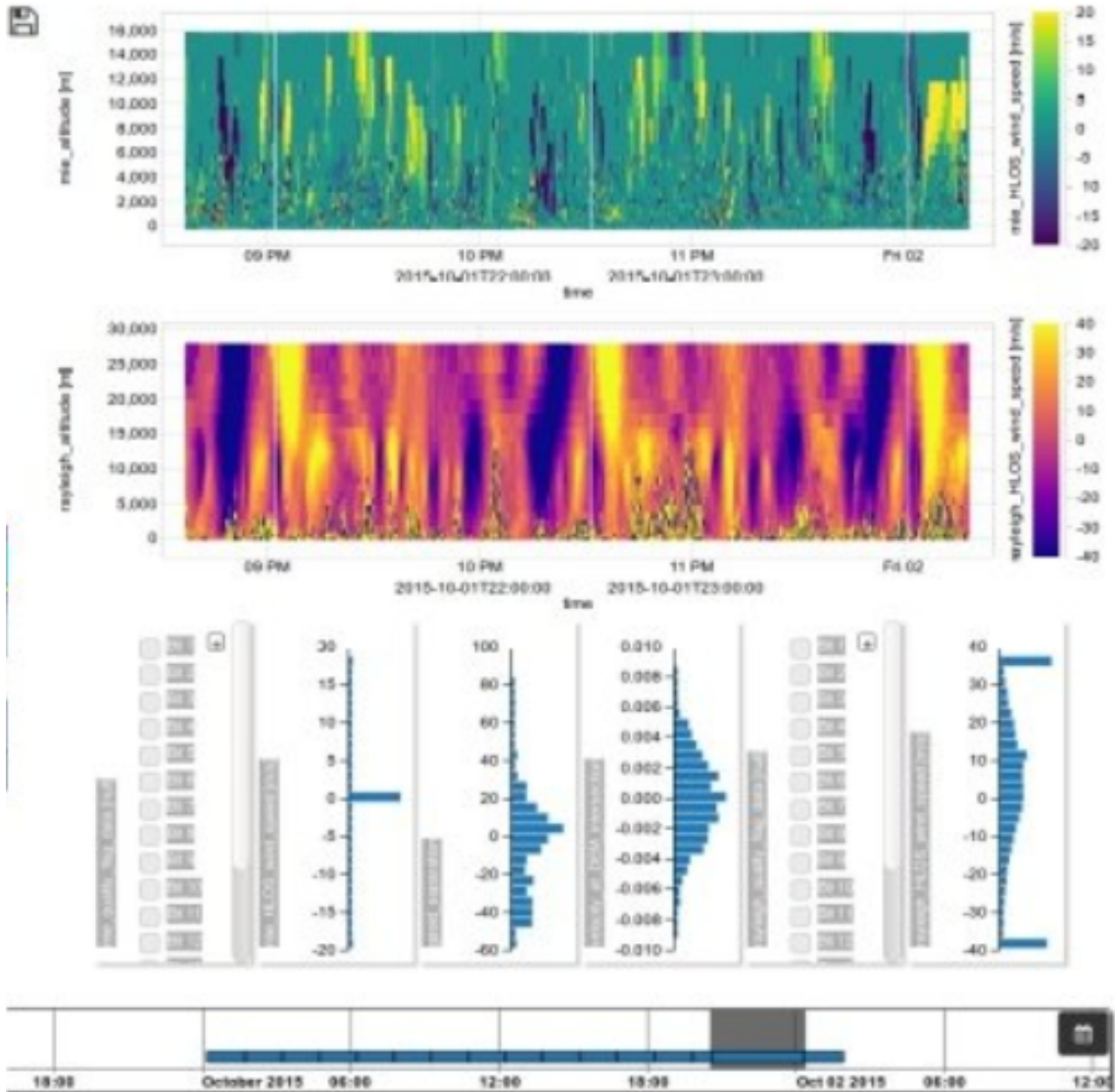


Figure 5. Timeseries Views of CloudSat 2B-GEOPROF Swath Data

Once the time of interest is visible the time slider allows the selection of the time interval for which data shall be visualized simply by clicking and holding on the timeslider. The current selection is highlighted with a gray background and can be adjusted at the ends as needed. Alternatively, the user can click on the availability indication for a quick selection. Any change of the time selection triggers a refresh of the data visualizations and prior data download as necessary.

Animation over a period with a given interval step may often be required to see the change. The timeline may support the selection of periods to be animated over.

5.3.4. Globe 2.5D/3D view

An example globe view is shown in [Figure 6](#). The view can also be changed to a map view. As on a standard map, the data would only be visible as footprint curves because the map view uses a 2.5D view. Both views are highly interactive allowing the user to zoom, pan, tilt, and rotate the globe or map.

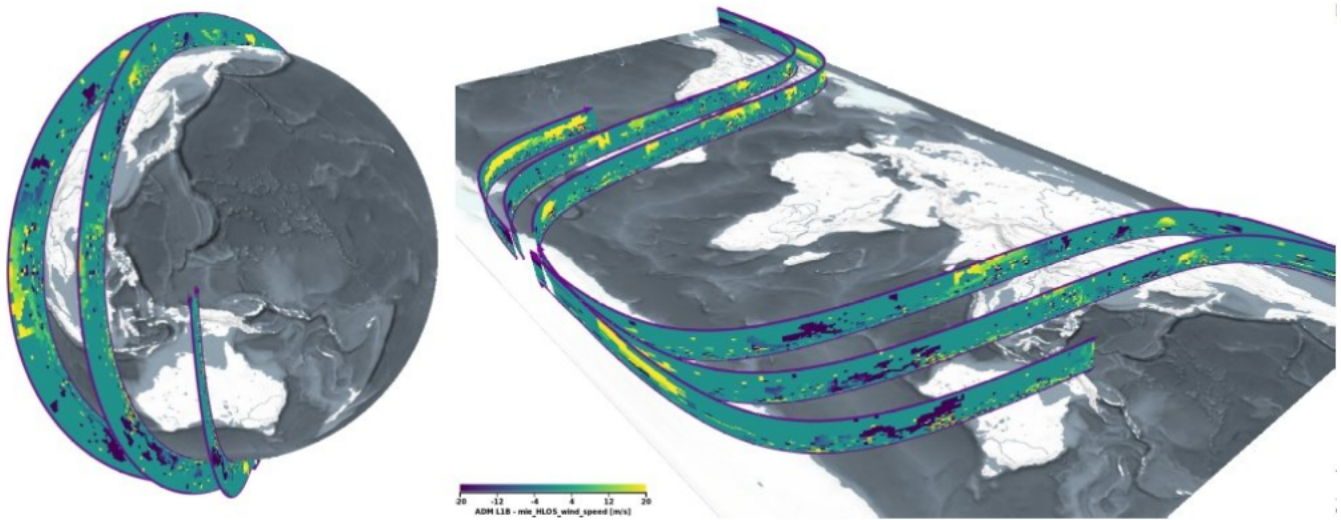


Figure 6. Timeseries Views of CloudSat 2B-GEOPROF Swath Data

The globe view may also be used as the basemap to select an area of interest for data access or detailed visual exploration of the data.

5.3.5. Analytical view

An analytical view may be required to show aggregated features of the coverage. The features may be the statistical distribution (e.g. histogram) or summation (e.g. statistic sum, mean, minimum, and maximum). The visuals may be rendered as maps or charts (e.g. bar charts for histogram). It is again highly interactive allowing a user to zoom and pan on both or individual axes. Linked views (connecting multiple views of the coverage) may be supported.

5.3.6. Analytics

The raw data may be required by users for further analysis and processing. In this context, the access of the swath data shall support the minimum content requirement for georeferencing as defined in standard [1].

The requirements to access the raw data of swath coverage imposes the following constraints on delivered data:

- Data should not be changed. Resampling should not happen.
- Integrity on the mapping relationship between data fields and geolocation fields should be maintained. The mapping is not just limited to just associate latitude/longitude/height to each

pixel, but also extended to relate to other location info (e.g. solar azimuth, solar zenith, satellite range, satellite azimuth, satellite zenith).

In general, swath data contains information: data fields, geolocation fields, and mapping between data and geolocation. The mapping between geolocation and data fields may be defined with Time/Location, Analytic fit, or Attitude/Ephemeris. Time is an important dimension in defining the mapping relationship between geolocation fields and data fields since a swath coverage is un-georeferenced or in the coordinate system of sensors. The time dimension may need to be distributed along with the swath coverage. This may be essential to maintain the integral correlation between geolocation fields and data fields when the swath coverage is subset spatially.

Chapter 6. Swath Coverage Models with Climate Forecast Conventions

Highlights

NOTE

This section describes the modeling of swath coverage using Climate Forecast (CF) conventions. The encoding of swath coverage in netCDF is the target.

6.1. Climate Forecast (CF) Conventions

The CF versions 1.7 and the in development version 1.8 were used in this testbed [5], [6]. The Swath-specific recommendation is adopted to define the swath coverage using CF [4].

6.2. Swath Coverage CF Coverage Models

The general decision tree as shown in the following is used to determine the CF models to be used to model swath coverage.

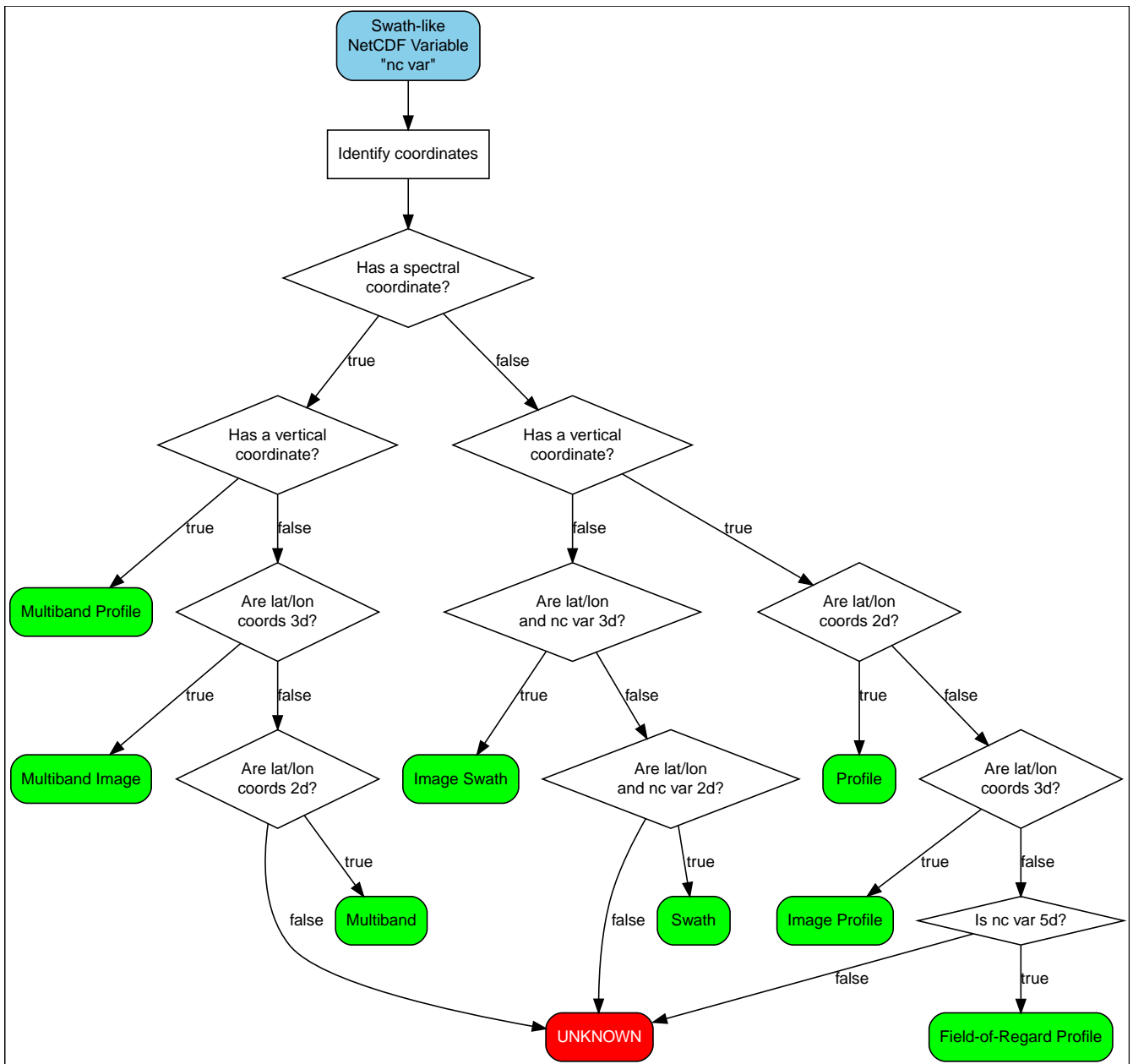


Figure 7. Decision tree for determining the encoding model in netCDF

6.2.1. VIIRS Swath Coverage

The data to be modeled is the Surface Reflectance (L2 Daily Swath product). These data are released as 6-minute swath data with two spatial resolutions of 750 m and 375 m at nadir [7] [8].

The decision flow on determining the model is shown in Figure 8. Model "multiband" may be used to all the bands of VIIRS.

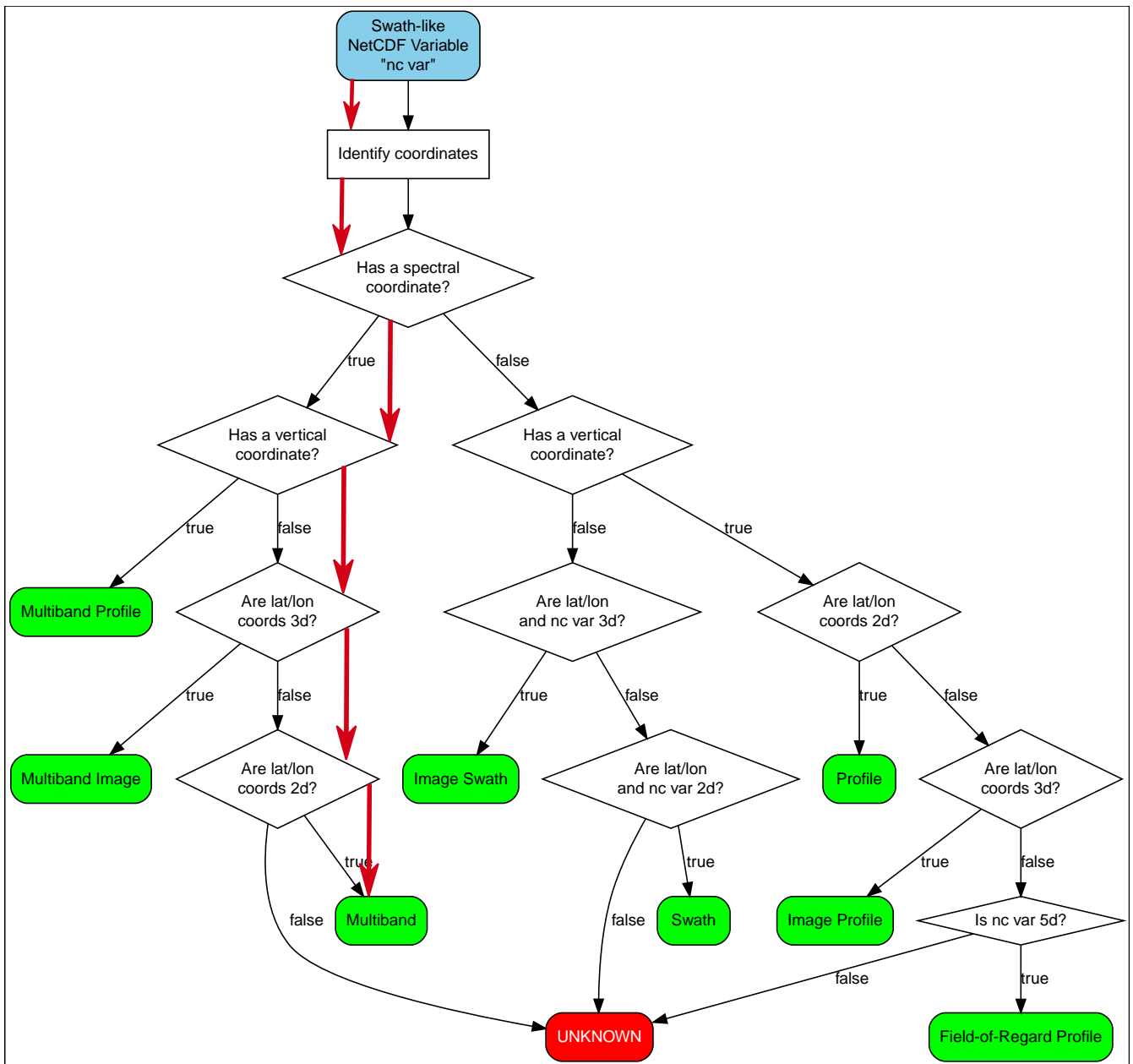


Figure 8. Decision tree for determining the encoding model in netCDF for VIIRS VNP09 Surface Reflectance 6-minute Swath Coverages

The I-Bands of VNP09 Surface Reflectance can be modeled as follows.

```
dimensions:
  time = 6496 ;
  scan = 6400 ;
  band = 3 ;
variables:
  float band(band) ;
    band:standard_name = "sensor_band_central_radiation_wavelength" ;
    band:units = "µm" ;
  float lat(time, scan) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;
  float lon(time, scan) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;
  double time(time) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;
  float height(time, scan) ;
    height:standard_name = "height" ;
    height:units = "meters" ;
  float swath_data(time, scan, band) ;
    swath_data:coordinates = "lon lat" ;
```

The M-Bands of VNP09 Surface Reflectance can be modeled as follows.

```
dimensions:
  time = 3248 ;
  scan = 3200 ;
  band = 3 ;
variables:
  float band(band) ;
    band:standard_name = "sensor_band_central_radiation_wavelength" ;
    band:units = "µm" ;
  float lat(time, scan) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;
  float lon(time, scan) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;
  double time(time) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;
  float height(time, scan) ;
    height:standard_name = "height" ;
    height:units = "meters" ;
  float swath_data(time, scan, band) ;
    swath_data:coordinates = "lon lat" ;
```

The operational release of daily VIIRS radiance and surface reflectance is released by NOAA. The release of VIIRS radiance and surface reflectance is archived band by band. The decision tree result for modeling the product is as shown in [Figure 9](#).

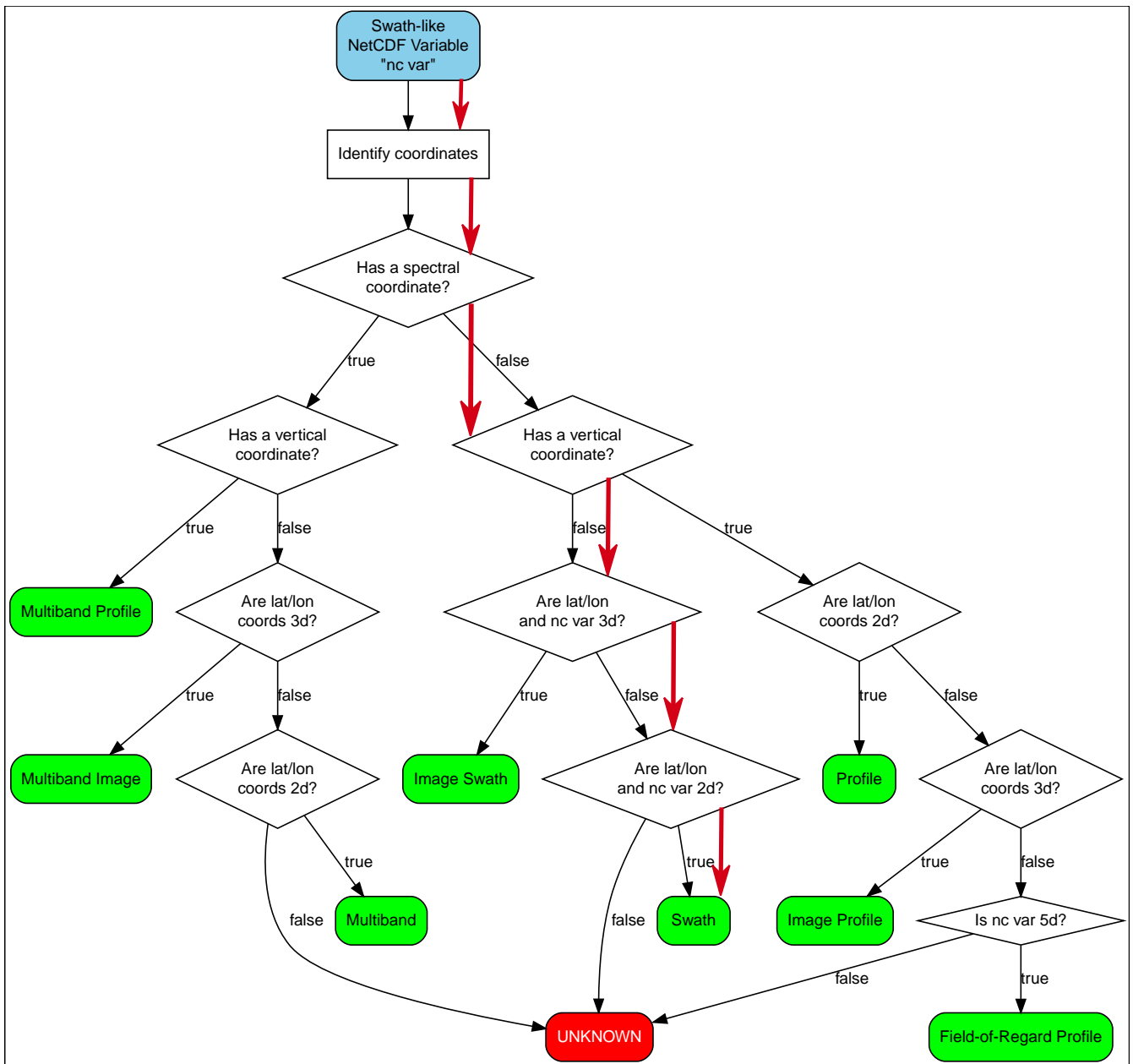


Figure 9. Decision tree for determining the encoding model in netCDF for VIIRS SVI/SVM Radiance and Surface Reflectance Swath Coverages

The SVI Bands of VIIRS (Radiance and Surface Reflectance) can be modeled as follows.

```
dimensions:
  time = 6144 ;
  scan = 6400 ;
variables:
  float lat(time, scan) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;
  float lon(time, scan) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;
  double time(time) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;
  float height(time, scan) ;
    height:standard_name = "height" ;
    height:units = "meters" ;
  float saza(time, scan) ;
    saza:standard_name = "SatelliteZenithAngle" ;
    saza:units = "degrees" ;
  float saaa(time, scan) ;
    saaa:standard_name = "SatelliteAzimuthAngle" ;
    saaa:units = "degrees" ;
  float soza(time, scan) ;
    soza:standard_name = "SolarZenithAngle" ;
    soza:units = "degrees" ;
  float soaa(time, scan) ;
    soaa:standard_name = "SolarAzimuthAngle" ;
    soaa:units = "degrees" ;
  float sar(time, scan) ;
    sar:standard_name = "SatelliteRange" ;
    sar:units = "meters" ;
  float swath_data(time, scan, band) ;
    swath_data:coordinates = "lon lat" ;
```

The SVM Bands of VIIRS (Radiance and Surface Reflectance) can be modeled as follows.

```
dimensions:
  time = 3072 ;
  scan = 3200 ;
variables:
  float lat(time, scan) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;
  float lon(time, scan) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;
  double time(time) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;
  float height(time, scan) ;
    height:standard_name = "height" ;
    height:units = "meters" ;
  float saza(time, scan) ;
    saza:standard_name = "SatelliteZenithAngle" ;
    saza:units = "degrees" ;
  float saaa(time, scan) ;
    saaa:standard_name = "SatelliteAzimuthAngle" ;
    saaa:units = "degrees" ;
  float soza(time, scan) ;
    soza:standard_name = "SolarZenithAngle" ;
    soza:units = "degrees" ;
  float soaa(time, scan) ;
    soaa:standard_name = "SolarAzimuthAngle" ;
    soaa:units = "degrees" ;
  float sar(time, scan) ;
    sar:standard_name = "SatelliteRange" ;
    sar:units = "meters" ;
  float swath_data(time, scan, band) ;
    swath_data:coordinates = "lon lat" ;
```

6.2.2. CloudSat Swath Coverage

Following through the decision tree, the model type is Profile, as shown in the section 2.5.3 in [4].

```
dimensions:  
  atrack = 20678;  
  xtrack = 125;  
  
variables:  
  
  double time(atriack) ;  
    time:standard_name = "time" ;  
    time:units = "<units> since <datetime string>" ;  
    time:calendar = "gregorian" ;  
  
  float lat(atriack) ;  
    lat:standard_name = "latitude" ;  
    lat:units = "degrees_north" ;  
  
  float lon(atriack) ;  
    lon:standard_name = "longitude" ;  
    lon:units = "degrees_east" ;  
  
  float swath_data(atriack, xtrack) ;  
    swath_data:coordinates = "time lon lat" ;
```

6.2.3. MODIS Swath Coverage

The modeling decision results in "Multiband Image" as described in section 2.4.2 [4].


```
dimensions:
  time = 1;
  nrows = 2330;
  ncols = 2030;
  band = 16;

variables:
  float band(band);
  band:standard_name = "Retrieved_Temperature_Profile";
  band:units = "K";

  float lat(time, nrows, ncols);
  lat:standard_name = "Geodetic Latitude";
  lat:units = "degrees_north";

  float lon(time, nrows, ncols);
  lon:standard_name = "Geodetic Longitude";
  lon:units = "degrees_east";

  double time(time);
  time:standard_name = "International Atomic Time at Start of Scan replicated
across the Swath";
  time:units = "s since 1 January 1993 00:00:00";
  time:calendar = "gregorian";

  float swath_data(time, nrows, ncols, band);
  swath_data:coordinates = "lon lat";
```

6.2.4. CALIPSO Swath Coverage

Following through the decision tree, the model type for CALIOP LIDAR swath coverage is Multiband Profile as shown in the section 2.5.3 in [4]. The decision tree result for modeling the product is as shown in [Figure 10](#).

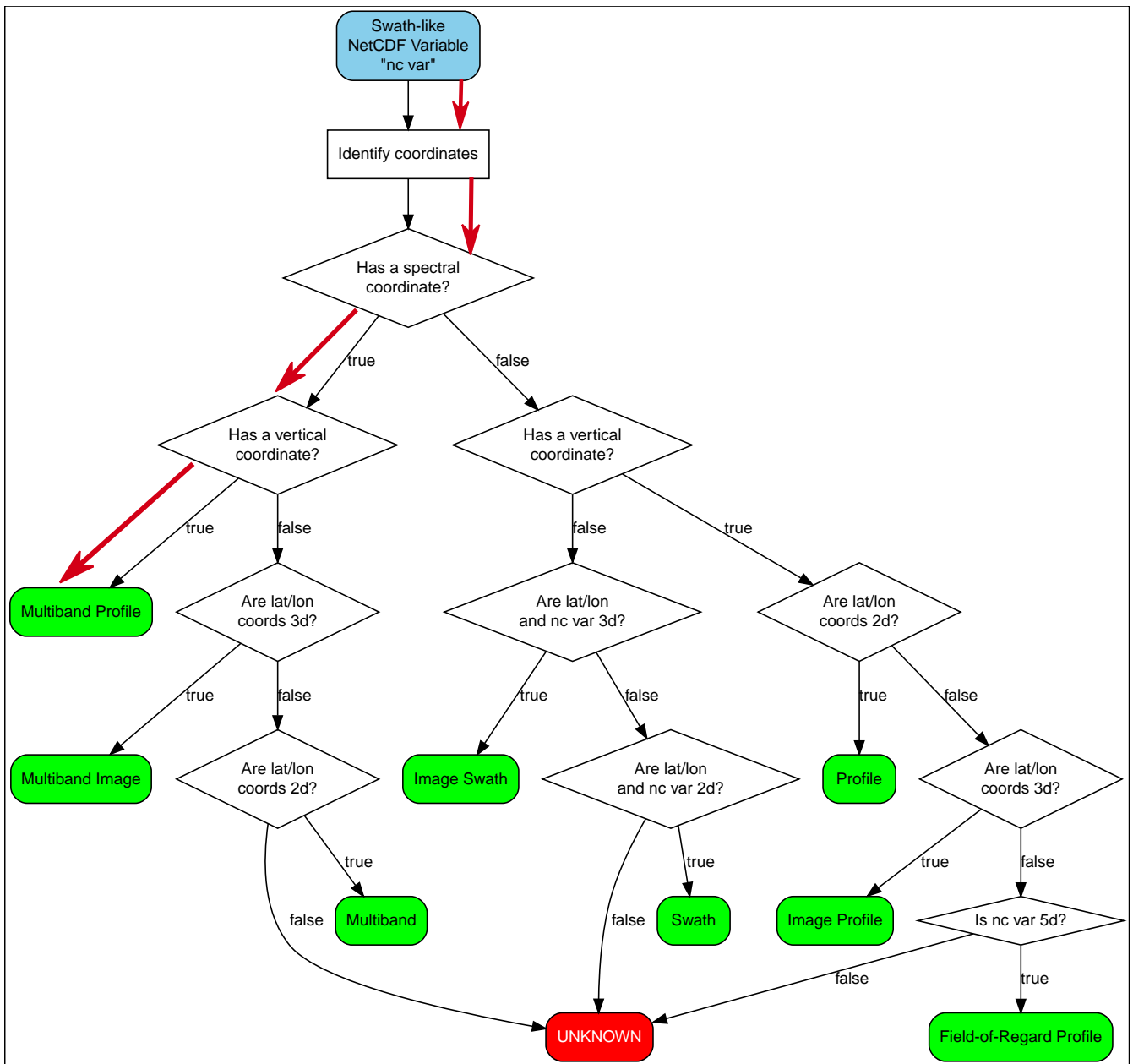


Figure 10. Decision tree for determining the encoding model in netCDF for VIIRS SVI/SVM Radiance and Surface Reflectance Swath Coverages

```
dimensions:
  atrack = 63330;
  z = 583

variables:
  float Altitude(z) ;
    alt:standard_name = "altitude";
    alt:long_name = "height above mean sea level" ;
    alt:units = "km" ;
    alt:positive = "up" ;
    alt:axis = "Z" ;
  double Profile_time(atrack) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;
  float Latitude(atrack) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;
  float Longitude(atrack) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;
  float Total_Attenuated_Backscatter_532(atrack, z) ;
    swath_data:coordinates = "lon lat alt" ;
  float Perpendicular_Attenuated_Backscatter_532(atrack, z) ;
    swath_data:coordinates = "lon lat alt" ;
  float Attenuated_Backscatter_1064(atrack, z) ;
    swath_data:coordinates = "lon lat alt" ;
```

Chapter 7. Swath Coverage Models with OGC Coverage Implementation Schema

Highlights

NOTE

This section describes the modeling of swath coverage using Coverage Implementation Schema (CIS). The encoding of swath coverage in GML coverage is the target.

7.1. Coverage Implementation Schema (CIS)

There are currently two versions of CIS, namely CIS version 1.0.1 and CIS version 1.1. In Testbed-14, CIS 1.1 was used to model the swath coverages. The reason for choosing version 1.1 is that it has more comprehensive capabilities in describing different coverages than version 1.0.1.

7.2. Swath Coverage Models

Two approaches were attempted to model swath coverages using CIS 1.1. One is based on extending the CIS definition of the Irregular grid class to support profiling swath. Another is use of the regular grid class, but leaving the geolocation attributes (including latitude/longitude/height/solar relative positions/satellite relative positions) to be mapped in *rangeset* fields.

In the subsequent subsections, different data types were attempted using both models. Example model results are provided. The pros and cons of the modeling approaches for different swath data types are discussed. In general, the following table summarizes the pros and cons of the two approaches. Method 1 (spatial domain) is the first model that directly model geographic coordinates as domain. Method 2 (spatial range) is the second model that model geographic reference coordinates in *rangeset* while the domain uses the coordinates of sensors.

Table 1. Pros and cons of two alternative swath coverage modeling methods

	Pros	Cons
Method 1 (spatial domain)	<ul style="list-style-type: none">• Direct support of spatial operations in domain	<ul style="list-style-type: none">• Irregular grid needs to be used• Irregular grid may explode the domain dataset with large amount of spatial coordinates
Method 2 (spatial range)	<ul style="list-style-type: none">• Regular grid may be sufficient in modeling sensor coordinates• Small size of domains dataset definition	<ul style="list-style-type: none">• Indirect support spatial operation through range subset extension

7.3. CIS Modeling with Irregular Grid

The irregular grid model directly uses longitude/latitude as the domain to be modeled as irregular grids in CIS.

7.3.1. MODIS MOD07 Swath Coverage

NOTE

MODIS MOD07 Swath regular grids on spatial coordinates while the irregular resolution along the altitude axis. The number of pressure levels along the altitude axis is limited. The use of Method 1 (spatial domain) works well.

Domainset

The following example shows a CIS modeling result.

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979%262=http://www.opengis.net/def/c
rs/OGC/0/AnsiDate" axisLabels="Lat Long h date">
    <RegularAxis axisLabel="Lat" uomLabel="deg" lowerBound="24.785248" upperBound
="27.685248" resolution="0.05"/>
    <RegularAxis axisLabel="Long" uomLabel="deg" lowerBound="0.673578"
upperBound="1.273578" resolution="0.05"/>
    <IrregularAxis axisLabel="t" uomLabel="s">
      <C>1514800500</C>
      <C>1514800500</C>
    </IrregularAxis>
    <IrregularAxis axisLabel="Pressure" uomLabel="hPa">
      <C>5.000000</C>
      <C>10.000000</C>
      <C>20.000000</C>
      <C>30.000000</C>
      <C>50.000000</C>
      <C>70.000000</C>
      <C>100.000000</C>
      <C>150.000000</C>
      <C>200.000000</C>
      <C>250.000000</C>
      <C>300.000000</C>
      <C>400.000000</C>
      <C>500.000000</C>
      <C>620.000000</C>
      <C>700.000000</C>
      <C>780.000000</C>
      <C>850.000000</C>
      <C>920.000000</C>
      <C>950.000000</C>
      <C>1000.000000</C>
    </IrregularAxis>
    <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index4D" axisLabels
="i j k l">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="11"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="57"/>
      <IndexAxis axisLabel="k" lowerBound="0" upperBound="0"/>
      <IndexAxis axisLabel="l" lowerBound="0" upperBound="1"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.

```
<RangeSet>
  <DataBlock>
    <V><tuple_list_5_hpa></V>
    <V>...</V>
    <V><tuple_list_1000_hpa></V>
  </DataBlock>
</RangeSet>
<RangeType>
  <swe:DataRecord>
    <swe:field name="panchromatic">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/unsignedInt">
        <swe:uom code="10^0"/>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>
```

7.3.2. VIIRS Swath Coverage

Domainset

NOTE

VIIRS Swath data are not geo-referenced. The spatial resolution for each pixel is irregular due to the aggregation measures used in the production of swath data. Each coordinate needs a pair of longitude and latitude to identify its location. The use of Method 1 (spatial domain) resulted in a domainset with large amount of geolocation data.

The following example shows a CIS modeling result.

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels=
"Lat Long h">
  <DisplacementAxisNest axisLabels="Lat Long Timescan" uomLabels="deg deg
sec">
    <P><C>-56.145405</C><C>69.62633</C><C>2018059113600000000</C></P>
<P><C>-56.15058</C><C>69.61555</C><C>2018059113600000001</C></P> <P><C>-56.155754
</C><C>69.60478</C><C>2018059113600000002</C></P>
    <P><C>-56.13966</C><C>69.61745</C><C>2018059113600010000</C></P>
<P><C>-56.14484</C><C>69.606674</C><C>2018059113600010001</C></P> <P><C>-56.150017
</C><C>69.59592</C><C>2018059113600010002</C></P>
    <P><C>-56.13392</C><C>69.608574</C><C>2018059113600020000</C></P>
<P><C>-56.1391</C><C>69.59781</C><C>2018059113600020001</C></P> <P><C>-56.14427</C>
<C>69.58705</C><C>2018059113600020002</C></P>
  </DisplacementAxisNest>
  <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index1D"
axisLabels="i">
    <IndexAxis axisLabel="i" lowerBound="0" upperBound="9"/>
  </GridLimits>
</GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.


```
<RangeType>
  <swe:DataRecord>
    <swe:field name="BandI1">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="BandI2">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="BandI3">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>
```

7.3.3. CloudSat Swath Coverage

NOTE

CloudSat Swath data are not geo-referenced profiles. Each pixel has unique values along all axes of longitude, latitude, and altitude. The use of Method 1 (spatial domain) resulted in a domainset with large amount of geolocation data.

Domainset

The following example shows a CIS modeling result.

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979&2=http://www.opengis.net/def
/crs/OGC/0/AnsiDate" axisLabels="Lat Long h date">

    <IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg d">
      <P><C>-69.35112</C><C>171.2182</C><C>46293.594+0.0</C></P>
      <P><C>-69.35994</C><C>171.20651</C><C>46293.594+0.15999997</C></P>

      ... 20753 <P> elements with 3 <C> elements each in total

    </IrregularAxisNest>

    <DisplacementAxisNest axisLabels="h" uomLabels="m">
      <P><C>24927</C></P>
      <P><C>24687</C></P>

      ... 125 x 20753 <P> elements with 1 <C> element each in total

    </DisplacementAxisNest>

    <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D" axisLabels
="i j">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="20752"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="124"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.

RangeType section of CloudSat 2B-GEOPROF Swath Coverage (irregular grid)

```
<RangeType>
  <swe:DataRecord>
    <swe:field name="Radar_Reflectivity">
      <swe:Quantity definition="TBD">
        <swe:description>TBD</swe:description>
        <swe:nilValues>
          <swe:nilValues>
            <swe:nilValue reason="fill value">15360</swe:nilValue>
          </swe:nilValues>
        </swe:nilValues>
        <swe:uom code="dBZe"/>
        <swe:constraint>
          <swe:AllowedValues>
```

```

        <swe:interval>-32768 32767</swe:interval>
        </swe:AllowedValues>
    </swe:constraint>
</swe:Quantity>
</swe:field>
<swe:field name="Gaseous_Attenuation">
    <swe:Quantity definition="TBD">
        <swe:description>TBD</swe:description>
        <swe:nilValues>
            <swe:nilValues>
                <swe:nilValue reason="fill value">15360</swe:nilValue>
            </swe:nilValues>
        </swe:nilValues>
        <swe:uom code="dBZe"/>
        <swe:constraint>
            <swe:AllowedValues>
                <swe:interval>-32768 32767</swe:interval>
            </swe:AllowedValues>
        </swe:constraint>
    </swe:Quantity>
</swe:field>
<swe:field name="CPR_Cloud_mask">
    <swe:Quantity definition="TBD">
        <swe:description>TBD</swe:description>
        <swe:nilValues>
            <swe:nilValues>
                <swe:nilValue reason="fill value">-99</swe:nilValue>
            </swe:nilValues>
        </swe:nilValues>
        <swe:uom code="various"/>
        <swe:constraint>
            <swe:AllowedValues>
                <swe:interval>-128 127</swe:interval>
            </swe:AllowedValues>
        </swe:constraint>
    </swe:Quantity>
</swe:field>
</swe:DataRecord>
</RangeType>

```

7.4. CIS Modeling with Regular Grid

The regular grid approach uses native along-track (normally time) and cross-track (normally scan) coordinates to define a grid while all geolocation fields are managed as ranges set fields.

7.4.1. VIIRS Swath Coverage

NOTE

VIIRS Swath data has a special rule for aggregation that resulted in different spatial resolution. The use of Method 2 (spatial range) needs specially predefined reference system that identify and reference the sensor coordinate systems with georeference systems.

Domainset

The following example shows a CIS modeling result.

Domainset section of VNP09 I-Bands Swath Coverage (regular grid)

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/eo/VIIRS" axisLabels="Time
Scan">
    <RegularAxis axisLabel="Time" lowerBound="201805911360000" upperBound=
"201805911360007" uomLabel="YYYYDDHMMSSSS" resolution="1"/>
    <RegularAxis axisLabel="Scan" lowerBound="0" upperBound="2" uomLabel=
"mgrad/sample" resolution="0.156"/>
    <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
axisLabels="i j">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="2"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="7"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.

```
<RangeType>
  <swe:DataRecord>
    <swe:field name="Latitude">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/float">
        <swe:uom code="deg"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="Longitude">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/float">
        <swe:uom code="deg"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="Height">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/float">
        <swe:uom code="m"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="BandI1">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="BandI2">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
    <swe:field name="BandI3">
      <swe:Quantity definition=
"http://www.opengis.net/def/dataType/OGC/0/signedInt">
        <swe:uom code="Reflectance/10000"/>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>
```

7.4.2. CloudSat Swath Coverage

NOTE

CloudSat Swath data are profiles. The use of Method 2 (spatial range) controls the size of domain dataset. However, the spatial operations of sub-setting can be demanding. Unique profile reference system for CloudSat needs to be defined for users to effectively use the service.

Domainset

The following example shows a CIS modeling result.

Domainset section of CloudSat 2B-GEOPROF Swath Coverage (regular grid)

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
    axisLabels="atrack xtrack">
    <RegularAxis axisLabel="atrack" lowerBound="0"
      upperBound="20677" uomLabel="scan" resolution="1" />
    <RegularAxis axisLabel="xtrack" lowerBound="0"
      upperBound="124" uomLabel="scan" resolution="1" />
  </GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.

RangeType section of CloudSat 2B-GEOPROF Swath Coverage (regular grid)

```
<RangeType>
  <swe:DataRecord>
    <swe:field name="Timestamp">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="second" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="Latitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="degree" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="Longitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="degree" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="SwathData">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="some_unit" />
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>
```

7.4.3. CALIPSO Swath Coverage

NOTE

CALIOP, the data from CALIPSO satellite, is a profile swath. Its modeling with Method 2 (spatial range) is similar to that for CloudSat. The further complexity for georeferencing and spatial sub-setting is resulted from the irregular coordinates due to the onboard aggregation scheme.

The following example shows a CIS modeling result.

Domainset section of CloudSat 2B-GEOPROF Swath Coverage (regular grid)

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
    axisLabels="atrack xtrack">
    <RegularAxis axisLabel="atrack" lowerBound="0"
      upperBound="63329" uomLabel="scan" resolution="1" />
    <RegularAxis axisLabel="z" lowerBound="0"
      upperBound="582" uomLabel="scan" resolution="1" />
  </GeneralGrid>
</DomainSet>
```

RangeType

The following example shows a CIS modeling result.

```
<RangeType>
  <swe:DataRecord>
    <swe:field name="Profile_time">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
          <swe:uom code="second" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Latitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/float32">
          <swe:uom code="degree" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Longitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/float32">
          <swe:uom code="degree" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Altitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/float32">
          <swe:uom code="km" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Total_Attenuated_Backscatter_532">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
          <swe:uom code="per kilometer per steradian" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Perpendicular_Attenuated_Backscatter_532">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
          <swe:uom code="per kilometer per steradian" />
        </swe:Quantity>
      </swe:field>
    <swe:field name="Attenuated_Backscatter_1064">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
          <swe:uom code="per kilometer per steradian" />
        </swe:Quantity>
      </swe:field>
    </swe:DataRecord>
  </RangeType>
```


Chapter 8. Swath Coverage Encoding

Highlights

NOTE

This section describes the encoding of a Swath Coverage in encodings other than CF and GML. The encoding formats include GeoTIFF, classic NetCDF, and GeoPackage.

8.1. Overview

The previous two sections describe encodings of swath coverage in CF and GML coverages. This section describes the following additional encodings of swath coverage experimented in Testbed 14. They are as follows.

- GeoTIFF
- GeoPackage

In the testbed, the participants also attempted to fully utilize the available formats in the well-known GDAL library to encode swath coverage. These included using its HDF4Image driver and NetCDF driver. The experience on encoding swath coverage is also discussed in this section.

8.2. GeoTIFF Encoding

In most cases, a swath coverage contains the following data or layers:

1. Geolocation layers, such as latitude, longitude, height, sensor angular position (e.g. satellite zenith angle, satellite azimuth, range to the satellite), and relative positions of radiance sources to be measured (e.g. solar zenith angle, solar azimuth, or (for night-time passive remote sensing) lunar zenith angle, lunar azimuth, and lunar phase);
2. Measurement layers or sensed digital numbers;
3. Mapping relationships between geolocations and measurements that relate the geolocations to each measurement.

These three reasons lead to using separate files to encode geolocations and measurements. Geolocations can be recorded in *float* or any other decimal numbers while measurement is recorded as digital numbers in integers (such as short integer). The data type for geolocation is generally different from that for measurements. Besides, geolocation layers may not have the same dimension as measurement layers. The most complex relationship between geolocations and measurements may need to have one-to-one mapping at the level of individual pixels. In other cases, one dimension of latitude and one dimension of longitude may be enough to form the axes to be mapped to two dimensions of measurement data. In addition, existing software packages may have limitations on the data layers. For example, GDAL, the commonly used raster data library, does support mixed types of layers in GeoTIFF files.

The following gives an example of encoding swath coverages for VIIRS Level 2 data where two files are generated to keep track. [Listing 1](#) shows the encoding for its geolocations. [Listing 2](#) shows the encoding for its data. A simple one-to-one mapping relationship is assumed to relate geolocations to measurements. To facilitate the cross-reference of relative locations of the pixels towards its nadir

in subsets of a data granule, the offset of each axis to the full scene origin is recorded and added as metadata.

Listing 1 An example encoding with GeoTIFF for geolocations of VIIRS swath coverage

```
river: GTiff/GeoTIFF
Files: VNP09.IBands.A2018115.1612.001_loc.tif
Size is 6400, 6464
Coordinate System is ``
Image Structure Metadata:
  COMPRESSION=LZW
  INTERLEAVE=PIXEL
Corner Coordinates:
Upper Left ( 0.0, 0.0)
Lower Left ( 0.0, 6464.0)
Upper Right ( 6400.0, 0.0)
Lower Right ( 6400.0, 6464.0)
Center ( 3200.0, 3232.0)
Band 1 Block=256x256 Type=Float32, ColorInterp=Gray
  Description = Latitude
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
Band 2 Block=256x256 Type=Float32, ColorInterp=Undefined
  Description = Longitude
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
Band 3 Block=256x256 Type=Float32, ColorInterp=Undefined
  Description = Height
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
```

```
Driver: GTiff/GeoTIFF
Files: VNP09.IBands.A2018115.1612.001_data.tif
Size is 6400, 6464
Coordinate System is ``
Image Structure Metadata:
  COMPRESSION=LZW
  INTERLEAVE=PIXEL
Corner Coordinates:
Upper Left  (   0.0,   0.0)
Lower Left  (   0.0, 6464.0)
Upper Right ( 6400.0,   0.0)
Lower Right ( 6400.0, 6464.0)
Center      ( 3200.0, 3232.0)
Band 1 Block=256x256 Type=Int16, ColorInterp=Gray
  Description = 375m Surface Reflectance Band I1
  Offset: 0,   Scale:0.00100000004749745
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
Band 2 Block=256x256 Type=Int16, ColorInterp=Undefined
  Description = 375m Surface Reflectance Band I2
  Offset: 0,   Scale:0.00100000004749745
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
Band 3 Block=256x256 Type=Int16, ColorInterp=Undefined
  Description = 375m Surface Reflectance Band I3
  Offset: 0,   Scale:0.00100000004749745
  Metadata:
    LineOffset=0
    ScanOffset=0
    TimeOffset=0
```

8.3. GeoPackage Encoding

Similarly, with the same consideration as those for GeoTIFF, different GeoPackages are used to manage different layers. Furthermore, being limited by the current support of GDAL on encoding non-integer data to one layer, the testbed participants experimented with the encoding of swath coverage with one layer in one GeoPackage file. [Listing 3](#) shows the encoding for the geolocations associated with the data layer. [Listing 4](#) shows the encoding for its data. However, the GDAL driver does not write out the metadata as programmed. An auxiliary file is needed to support the metadata. Besides, the GeoPackage driver defaults to creating tiled raster image. The un-georeferenced swath data is not supported for tiling. Further studies on encoding swath coverages are needed. If GDAL is still considered as the library to work on different swath coverage, the GPKG driver of GDAL needs to be enhanced to support non-tiled data; for example, through support for

the OGC GeoPackage Extension for Tiled Gridded Coverage Data (17-066r1). The encoding with GeoPackage needs to be further studied as well, to properly manage swath data for both visualization and data access.

Listing 3 An example encoding with GeoPackage for geolocations of VIIRS swath coverage

```
Driver: GPKG/GeoPackage
Files: VNP09.IBands.A2018115.1612.001_loc.gpkg.latitude.gpkg
Size is 6400, 6464
Coordinate System is ``
Origin = (0.000000000000000,6464.000000000000000)
Pixel Size = (1.000000000000000,-1.000000000000000)
Metadata:
  IDENTIFIER=VNP09.IBands.A2018115.1612.001_loc.gpkg.latitude
  ZOOM_LEVEL=5
Image Structure Metadata:
  INTERLEAVE=PIXEL
Corner Coordinates:
Upper Left ( 0.000, 6464.000)
Lower Left ( 0.000000, 0.000000)
Upper Right ( 6400.000, 6464.000)
Lower Right ( 6400.000, 0.000)
Center ( 3200.000, 3232.000)
Band 1 Block=256x256 Type=Float32, ColorInterp=Undefined
```

Listing 4 An example encoding with GeoPackage for data of VIIRS swath coverage

```
Driver: GPKG/GeoPackage
Files: VNP09.IBands.A2018115.1612.001_data.gpkg.band1.gpkg
Size is 6400, 6464
Coordinate System is ``
Origin = (0.000000000000000,6464.000000000000000)
Pixel Size = (1.000000000000000,-1.000000000000000)
Metadata:
  IDENTIFIER=VNP09.IBands.A2018115.1612.001_data.gpkg.band1
  ZOOM_LEVEL=5
Image Structure Metadata:
  INTERLEAVE=PIXEL
Corner Coordinates:
Upper Left ( 0.000, 6464.000)
Lower Left ( 0.000000, 0.000000)
Upper Right ( 6400.000, 6464.000)
Lower Right ( 6400.000, 0.000)
Center ( 3200.000, 3232.000)
Band 1 Block=256x256 Type=Int16, ColorInterp=Undefined
```

8.4. Encodings with GDAL drivers

Encoding Swath Coverage with GDAL HDF4Image Driver

HDFEOS2 and HDFEOS5 natively support the encoding of swath data since it has a special data type as swath in addition to grid and point [9] [10] [11] [12]. However, the data are encoded differently for different sensors and datasets. In this testbed, GDAL was used as the primary library to support data input and output. For writing out, GDAL currently only has an HDF4Image driver which does not have the full operation to encode swath coverage. This leads to this section on encoding swath coverage in HDF4.

Using the same three reasons as those stated for GeoTIFF - data type difference, dimension difference, and existing software support - the encoding of geolocation and measurements are stored in separate files. Similarly, metadata for keeping track of the offsets in each dimension for sub-setting are encoded and stored. Listing 5 shows the encoding for its geolocations. Listing 6 shows the encoding for its data. With the GDAL library used in the testbed, the metadata are not directly added to each layer, but saved as an auxiliary xml file, as shown as one example in Listing 7.

Listing 5 An example encoding with GDAL HDF4Image driver for geolocations of VIIRS swath coverage

```
Driver: HDF4Image/HDF4 Dataset
Files: VNP09.IBands.A2018115.1612.001_loc.hdf
       VNP09.IBands.A2018115.1612.001_loc.hdf.aux.xml
Size is 6400, 6464
Coordinate System is ``
Origin = (0.000000000000000,0.000000000000000)
Pixel Size = (1.000000000000000,1.000000000000000)
Metadata:
  BandDesc1=Latitude
  BandDesc2=Longitude
  BandDesc3=Height
  Signature=Created with GDAL (http://www.remotesensing.org/gdal/)
  TransformationMatrix=0.000000, 1.000000, 0.000000, 0.000000, 0.000000, 1.000000
Corner Coordinates:
Upper Left ( 0.000000, 0.000000)
Lower Left ( 0.000, 6464.000)
Upper Right ( 6400.000, 0.000)
Lower Right ( 6400.000, 6464.000)
Center ( 3200.000, 3232.000)
Band 1 Block=6400x156 Type=Float32, ColorInterp=Gray
  Description = Latitude
Band 2 Block=6400x156 Type=Float32, ColorInterp=Gray
  Description = Longitude
Band 3 Block=6400x156 Type=Float32, ColorInterp=Gray
  Description = Height
```

Listing 6 An example encoding with GDAL HDF4Image driver for data of VIIRS swath coverage

```
Driver: HDF4Image/HDF4 Dataset
Files: VNP09.IBands.A2018115.1612.001_data.hdf
      VNP09.IBands.A2018115.1612.001_data.hdf.aux.xml
Size is 6400, 6464
Coordinate System is ``
Origin = (0.000000000000000,0.000000000000000)
Pixel Size = (1.000000000000000,1.000000000000000)
Metadata:
  BandDesc1=375m Surface Reflectance Band I1
  BandDesc2=375m Surface Reflectance Band I2
  BandDesc3=375m Surface Reflectance Band I3
  Signature=Created with GDAL (http://www.remotesensing.org/gdal/)
  TransformationMatrix=0.000000, 1.000000, 0.000000, 0.000000, 0.000000, 1.000000
Corner Coordinates:
Upper Left ( 0.0000000, 0.0000000)
Lower Left ( 0.000, 6464.000)
Upper Right ( 6400.000, 0.000)
Lower Right ( 6400.000, 6464.000)
Center ( 3200.000, 3232.000)
Band 1 Block=6400x156 Type=Int16, ColorInterp=Gray
  Description = 375m Surface Reflectance Band I1
Band 2 Block=6400x156 Type=Int16, ColorInterp=Gray
  Description = 375m Surface Reflectance Band I2
Band 3 Block=6400x156 Type=Int16, ColorInterp=Gray
  Description = 375m Surface Reflectance Band I3
```

```
<PAMDataset>
  <PAMRasterBand band="1">
    <Description>375m Surface Reflectance Band I1</Description>
    <Scale>0.001000000047497451</Scale>
    <Metadata>
      <MDI key="LineOffset">0</MDI>
      <MDI key="ScanOffset">0</MDI>
      <MDI key="TimeOffset">0</MDI>
    </Metadata>
  </PAMRasterBand>
  <PAMRasterBand band="2">
    <Description>375m Surface Reflectance Band I2</Description>
    <Scale>0.001000000047497451</Scale>
    <Metadata>
      <MDI key="LineOffset">0</MDI>
      <MDI key="ScanOffset">0</MDI>
      <MDI key="TimeOffset">0</MDI>
    </Metadata>
  </PAMRasterBand>
  <PAMRasterBand band="3">
    <Description>375m Surface Reflectance Band I3</Description>
    <Scale>0.001000000047497451</Scale>
    <Metadata>
      <MDI key="LineOffset">0</MDI>
      <MDI key="ScanOffset">0</MDI>
      <MDI key="TimeOffset">0</MDI>
    </Metadata>
  </PAMRasterBand>
</PAMDataset>
```

Encoding Swath Coverage with GDAL netCDF Driver

The GDAL (version 2.3 used in the testbed) has an old netCDF driver. It supports CF 1.5. The writer of the driver has limited capabilities similar to HDF4Image driver. The output model of NetCDF is "classic", or commonly labeled as "NetCDF-3". The testbed participants experimented with encoding the swath coverage data using the driver by generating separate geolocation file and data file. [Listing 8](#) shows the encoding for its geolocations. [Listing 9](#) shows the encoding for its data. The metadata were also programmed to be written out. However, examining the output of netCDF file, the additional metadata are not written out. The metadata may be appended as separate files to be packed into one package to support the extra metadata.

Listing 8 An example encoding with GDAL netCDF driver for geolocations of VIIRS swath coverage

```
Driver: netCDF/Network Common Data Format
Files: VNP09.IBands.A2018115.1612.001_loc.nc
Size is 512, 512
Coordinate System is ``
Metadata:
  NC_GLOBAL#Conventions=CF-1.5
  NC_GLOBAL#GDAL=GDAL 2.3.2, released 2018/09/21
  NC_GLOBAL#history=Tue Oct 09 15:31:42 2018: GDAL Create(
/home/eugene/ogcwks/wksp/GMUWCS21/temp/VNP09.IBands.A2018115.1612.001_loc.nc, ... )
Subdatasets:
  SUBDATASET_1_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_loc.nc":Band1
  SUBDATASET_1_DESC=[6464x6400] Band1 (32-bit floating-point)
  SUBDATASET_2_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_loc.nc":Band2
  SUBDATASET_2_DESC=[6464x6400] Band2 (32-bit floating-point)
  SUBDATASET_3_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_loc.nc":Band3
  SUBDATASET_3_DESC=[6464x6400] Band3 (32-bit floating-point)
```

Listing 9 An example encoding with GDAL netCDF driver for data of VIIRS swath coverage

```
Driver: netCDF/Network Common Data Format
Files: VNP09.IBands.A2018115.1612.001_data.nc
Size is 512, 512
Coordinate System is ``
Metadata:
  NC_GLOBAL#Conventions=CF-1.5
  NC_GLOBAL#GDAL=GDAL 2.3.2, released 2018/09/21
  NC_GLOBAL#history=Tue Oct 09 15:31:25 2018: GDAL Create(
/home/eugene/ogcwks/wksp/GMUWCS21/temp/VNP09.IBands.A2018115.1612.001_data.nc, ... )
Subdatasets:
  SUBDATASET_1_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_data.nc":Band1
  SUBDATASET_1_DESC=[6464x6400] Band1 (16-bit integer)
  SUBDATASET_2_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_data.nc":Band2
  SUBDATASET_2_DESC=[6464x6400] Band2 (16-bit integer)
  SUBDATASET_3_NAME=NETCDF:"VNP09.IBands.A2018115.1612.001_data.nc":Band3
  SUBDATASET_3_DESC=[6464x6400] Band3 (16-bit integer)
```


Chapter 9. WCS Swath Coverage Profile

NOTE

Highlights

This section captures the special profiles needed for WCS to server Swath Coverage.

9.1. Scope

This section provides recommendations in implementing a Swath Coverage as a Profile (with extensions) to WCS to easily access Swath Coverages, based on the requirements, experiments, and tests in this OGC testbed.

These profiles and extensions were defined in reference to WCS 2.0.1 and CIS 1.1.

9.2. Conformance

The following are the conformance classes defined in this specification.

- Swath Coverage Extension
- Swath GeoTiff Coverage Conformance Class
- Swath NetCDF Coverage Conformance Class
- Swath GeoPackage Coverage Conformance Class

9.3. Swath Coverage Extension

Requirement SCE1 - Support of time dimension

Table 2. Requirement SCE1 - time dimension

Requirement	SCE1
Requirement Name	/req/core/sce1
Description	Swath coverage is often along the time axis. The support of sub-setting against time is required. The request imposes the explicit use of time as a bounding dimension in describing the coverage.

Requirement SCE2 - Encoding of Swath coverage

Table 3. Requirement SCE2 - CIS 1.1 encoding

Requirement	SCE2
Requirement Name	/req/core/sce2

Requirement	SCE2
Description	The encoding of swath coverage should assure the correct cross-referencing among time, cross-track scan, and relative positions among sensors, ground targets, and suns (or the primary source of radiance).

9.4. Swath GeoTiff Coverage

Table 4. Requirement SEEG1 - geotiff encoding

Requirement	SEEG1
Requirement Name	/req/ext/geotiff/seeg1
Description	The encoding of swath coverage in geotiff is supported with this extension. Considering the difference of data types in geolocation layers and reflectance layers, two geotiff files may be used to encode them separately. By default, the one-to-one relationship between pixel from geolocation layers and that from reflectance layers should be supported unless an explicit mapping relationship is defined.

9.5. Swath NetCDF Coverage

Table 5. Requirement SEEN1 - netCDF encoding

Requirement	SEEN1
Requirement Name	/req/ext/netcdf/seen1

Requirement	SEEN1
Description	<p>The encoding of swath coverage in netCDF is supported with this extension. Considering the difference of data types in geolocation layers and reflectance layers and readily available support from existing software packages, two NetCDF files may be used to encode them separately. By default, the one-to-one relationship between pixel from geolocation layers and that from reflectance layers should be supported unless an explicit mapping relationship is defined. The CF 1.8 along with the developing extension for swath coverage may be used to encode the coverage as a swath. If the swath coverage extension for CF is used, a single file would be sufficient to manage both geolocation properties and measurement properties.</p>

9.6. Swath GeoPackage Coverage

Table 6. Requirement SEEGPKG1 - GeoPackage encoding

Requirement	SEEGPKG1
Requirement Name	/req/ext/geopackage/seegpkg1
Description	<p>The encoding of swath coverage in geopackage is supported with this extension. Considering the difference of data types in geolocation layers and reflectance layers and readily available support of open-source program (such as GDAL), each layer may be encoded in different geopackage files. By default, the one-to-one relationship between pixel from all layers - either geolocation layers and data layers - should be supported.</p>

Chapter 10. WCS REST API

Highlights

NOTE This section brief describes the principles for designing the WCS REST API and the major paths. Full details can be found in Appendix B.

10.1. Scope and Design Principle

The following specification provides recommendations for implementing a RESTful API as an alternative interface service to a Web Coverage Service. The RESTful API is designed and documented following the OpenAPI 3.0.1 specification. The core resource is on coverage.

The REST API is based on OpenAPI version 3.0.1 (URL: <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md>). This leads to the following fixed value of the OpenAPI document. OpenAPI documents can be defined in either JSON or YAML ([Version 1.2](http://yaml.org/spec/1.2/spec.html) [http://yaml.org/spec/1.2/spec.html]).

The WFS 3.0 development team introduced their experience in developing RESTful API for WFS. This work formed the base for the development of the draft specification of a WCS REST API. The API is designed with modules and extensions. The API core defines the basic paths while all the other requirements are set as extensions or conformance classes. If the service claims to conform to certain modular class, the requirements of that compliant conformance class should be met by the service. The requirements are grouped into levels of conformance that the implementation of the service claims to conform to.

10.2. Conformance

The following mentions all the conformance classes to be defined in this specification.

- Core Conformance Class: This standard defines six requirements / conformance classes.
- Range Sub-setting Extension Conformance Class.
- Earth Observation Extension Conformance Class.
- Transaction Insertion Deletion Conformance Class.
- Transaction Update Conformance Class.
- GeoTiff Coverage Conformance Class.
- NetCDF Coverage Conformance Class.
- GeoPackage Coverage Conformance Class.

10.3. Main Paths

The resources managed by the draft WCS REST API are mainly grouped into two categories. One category is metadata that is more or less corresponding to that described in the WCS operations GetCapabilities and DescribeCoverage. Another category is the data itself. This is related to the

current WCS operation GetCoverage. The following table lists the major paths and their primary purpose.

Table 7. Major Paths in WCS REST API

PATH	METHOD	DESCRIPTION
/	GET	Gets the landing page in the requested representation
/api	GET	Gets the API description (i.e. OpenAPI document)
/conformance	GET	Gets conformance classes that server implements
/coverages	GET	Gets the description/metadata of all the coverages offered by the system
/coverages/{coverageId}	GET	Gets the description/metadata of the specific coverage
/coverages/{coverageId}/rangeSet	GET	Gets the coverage or a subset thereof

In the context of the WCS REST API design experiment, the resources are identified by paths. Each path uniquely identifies a resource. For example, the root path / identifies the entry point which returns a document containing links to other resources - i.e. specifications identified by "/api", conformance declaration (identified by "/conformance"), and list of coverages (identified by "/coverages"). The coverage description of each coverage is identified as "/coverages/{coverageId}" while the actual data of a coverage is identified as "/coverages/{coverageId}/rangeSet". The paths are unique identifiers for different resources.

The resources can be traversed through following the hyperlinks in the content from the root endpoint (root path). For example, the response from the root path would contain links to hyperlinks to API specification document, conformance, and list of coverages. Once transferred to the resource that lists coverages, its content provides hyperlinks allowing users to traverse and find resources for coverage descriptions and coverage data. When coverage descriptions are in media types other than HTML, users need to use the schema to follow through the resources.

10.4. Service root path /

10.4.1. Get Operation

Requirement	C1
Requirement Name	/req/core/rootpath-get
Path	/
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml

Requirement	C1
Description	The server SHALL support the HTTP GET operation at the path /. Three media types may be supported: application/json, application/xml, and text/html.

10.5. Service OpenAPI path /api

10.5.1. Get Operation

Requirement	C2
Requirement Name	/req/core/apipath-get
Path	/api
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /api.

10.6. Service OpenAPI path /conformance

10.6.1. Get Operation

Requirement	C3
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/core as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

10.7. Service OpenAPI path /coverages

10.7.1. Get Operation

Requirement	C4
Requirement Name	/req/core/coveragespath-get
Path	/coverages
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Parameters	q – in query, string, keywords search
	bbox – in path, comma-separated bounding box defined by { west, south, min. height, east, north, max. height}
	time – in path, comma-separated time bounded with earliest and latest.
	startIndex – in query, integer, the start index for returned coverages.
	count – in query, integer, number of coverages to be returned counting start from startIndex
Description	The server SHALL support the HTTP GET operation at the path /coverages. The default number of items is 10. If there are less than 10 coverages served in the WCS, the actual number of items are listed.

10.8. Service OpenAPI path /coverages/{coverageId}

10.8.1. Get Operation

Requirement	C4
Requirement Name	/req/core/coverages/coverageidpath-get
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.
Description	The server SHALL support the HTTP GET operation at the path /coverages/{coverageId}. It returns the metadata for the coverage identified by {coverageId}.

10.9. Service OpenAPI path /coverages/{coverageId}/rangeSet

10.9.1. Get Operation

Requirement	C4
Requirement Name	/req/core/coverages/coverageidpath/rangeSet-get
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.
	subset – in path, in the format of dimension(trimlow,trimhigh) or dimension(slicepoint)
Description	The server SHALL support the HTTP GET operation at the path /coverages/{coverageId}/rangeSet. It returns the metadata for the coverage identified by {coverageId}.

Chapter 11. Swath Coverage Use Cases

Highlights

NOTE This section describes the example implementations of Swath WCS and clients. Typical use cases will be covered here.

11.1. Implementations of Swath WCS

Several teams are engaged in implementing the draft WCS Swath Coverage Profile. The implementation environment and basic system take different forms.

11.1.1. CloudSat Level 2 Profile Swath Data

Data Processing Levels

EOSDIS data products are processed at various levels ranging from Level 0 to Level 4. Level 0 products are raw data at full instrument resolution. At higher levels, the data are converted into more useful parameters and formats. All EOS instruments must have Level 1 products. Most have products at Levels 2 and 3, and many have products at Level 4.

Level 2 data derived geophysical variables at the same resolution and location as Level 1 source data.

2B-GEOPROF Data

The testbed participants investigated a level 2 profile data named 2B GEOPROF. The 2B GEOPROF product identifies those levels in the vertical column sampled by CloudSat that contain significant radar echo from hydrometeors (rather than noise or radar clutter) and provides an estimate of the radar reflectivity factor for each of these volumes. Also included in the GEOPROF product is an estimate of the expected gaseous absorption loss for the observed reflectivity (which depends on water vapor fields from ECMWF), the MODIS cloud fraction (from MOD35) associated with the radar surface foot print, and several other flags the indicates the homogeneity of the MODIS data, and quality of the CloudSat data. The data fields are listed in the tables below:

Data Fields

Table 8. Data fields for CloudSat Level 2 2B-GEOPROF data

Popup	Type	Dimension s	Units	Range	Missing	
----- -----	-----	-----	-----	-----	-----	
Clutter_red uction_flag	INT(1)	nray				
CPR_Cloud_ mask	INT(1)	nbin,nray			-9	

Popup	Type	Dimensions	Units	Range	Missing	
CPR_Echo_Top	INT(1)	nray			-9	
Data_quality	UINT(1)	nray	—	0 to 127		
Data_status	UINT(1)	nray	—	0 to 127		
Data_target ID	UINT(1)	nray	—	0 to 81		
Gaseous_Atenuation	INT(2)	nbin,nray	dBZe		-99.99	
MODIS_cloud_flag	INT(1)	nray	None		99	
MODIS_Cloud_Fraction	INT(1)	nray			-99	
MODIS_scene_char	INT(1)	nray			-9	
MODIS_scene_var	INT(1)	nray			-9	
Navigation_land_sea_flag	UINT(1)	nray	—	1 to 3		
Radar_Reflectivity	INT(2)	nbin,nray	dBZe	-40 to 50	-88.88	
sem_NoiseFloor	REAL(4)	nray			0	
sem_NoiseFloorVar	REAL(4)	nray			0	
sem_NoiseGate	INT(1)	nray			0	
Sigma-Zero	INT(2)	nray	dB*100	-1000 to 4000	-9999	
SurfaceHeightBin	INT(1)	nray		1 to 125	-1	
SurfaceHeightBin_fraction	REAL(4)	nray	none		0	

Geolocation Fields

Table 9. Geolocation fields for CloudSat Level 2 2B-GEOPROF data

Popup	Type	Dimension s	Units	Range	Missing	
----- --	-----	-----	-----	-----	-----	
DEM_elevation	INT(2)	nray	meters	-9999 to 8850	9999	
Height	INT(2)	nbin,nray	m	-5000 to 30000	-9999	
Latitude	REAL(4)	nray	degrees	-90 to 90		
Longitude	REAL(4)	nray	degrees	-180 to 180		
Pitch_offset	REAL(4)	<scalar>	degrees	-90 to 90		
Profile_time	REAL(4)	nray	seconds	0 to 6000		
Range_to_intercept	REAL(4)	nray	km	600 to 800		
Roll_offset	REAL(4)	<scalar>	degrees	-90 to 90		
TAI_start ¹	REAL(8)	<scalar>	seconds	0 to 6e+008		
UTC_start ²	REAL(4)	<scalar>	seconds	0 to 86400		
Vertical_binsize	REAL(4)	<scalar>	m	to	-9999	

1. *TAI_Start*: The TAI timestamp for the first profile in the data file. TAI is International Atomic Time: seconds since 00:00:00 Jan 1 1993.
2. *UTC_Start*: The UTC seconds since 00:00 Z of the first profile in the data file.

For instance, a visualization for variable “Radar Reflectivity” of CloudSat 2B-GEOPROF is demonstrated in Figure 11.

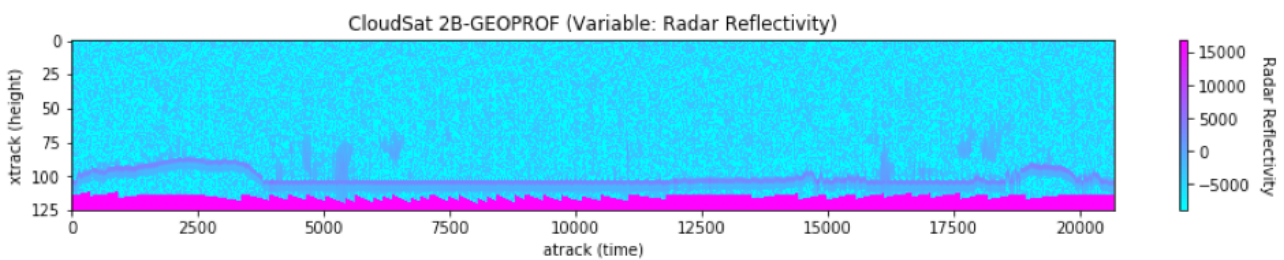


Figure 11. Scanning swath

11.1.2. Modeling with CF Convention

Climate and Forecast (CF) convention document is available at: <https://github.com/Unidata/EC-netCDF-CF/blob/master/swath/swath.adoc#image-profile>

According to section 2.5.3. Profile, this data is modeled as follows:

```

dimensions:
  atrack = 20678;
  xtrack = 125;

variables:

  double time(atriack) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;

  float lat(atriack) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;

  float lon(atriack) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;

  float swath_data(atriack, xtrack) ;
    swath_data:coordinates = "time lon lat" ;

```

11.1.3. CloudSat Swath WCS Implementation I

This section describes the implementation done by EOX. They identified the following requirements for implementing the swath WCS to serve CloudSat data:

- Service requirements - The service shall support:
 - retrieval of product metadata needed for visualization components, such as the timeslider view and the map view,
 - parameter subsetting, i.e., retrieving of individual parameters,
 - product subsetting in time,
 - spatial product subsetting (i.e., subsetting in latitude and longitude),
 - and, retrieval of "overviews" for performance optimization.
- Format requirements - The data format shall be:
 - self-describing to specify for example if the data is encoded as pixel-in-center or pixel-in-corner,
 - and, readable and interpretable in modern web browsers with existing libraries.

Modeling the CloudSat Swath Data Service using CIS 1.1

Retrieval of metadata

The standard WCS DescribeCoverage response contains all information needed to satisfy the requirement on metadata retrieval. However, there are two drawbacks. First, the `DomainSet` will be

quite huge or at least equal compared to the actual data and second the `DescribeCoverage` operation doesn't support searching in time and space.

The metadata needed by the client is ID, start, stop, and bounding box or better footprint. Additionally, often the envelope in physical dimensions will not be sufficient and clients may need to know the grid size or size of the mathematical dimensions of the `DomainSet`. In case of a `GeneralGrid` this information is present in the `GridLimits` element.

To overcome the second point the `DescribeEOCoverageSet` request of the Earth Observation application profile of WCS can be used. All 2B-GEOPROF products are grouped in a `DatasetSeries` which is then used in `DescribeEOCoverageSet` requests to get only the metadata to fill the currently visible time interval and spatial extent.

There is currently no way to retrieve coverage metadata restricting the potentially huge `DomainSet`. To avoid unnecessary large downloads, the client would either need to query a synchronized catalog or a change of the `DescribeCoverage` operation allowing limits to the response.

We propose providing the `DomainSet` axes information as coverages themselves via OGC Web Services Context (OWC) encoding. This way the `DomainSet` stays as small as possible while still providing the needed metadata. Additionally, by providing the axes information as coverages themselves allows direct subsetting of the domain. Thus, clients have a means to only download the needed information at the cost of additional `GetCoverage` requests to the `DomainSet` coverages.

A discarded idea was to add a boolean parameter named `restrictDomainSet` to the `DescribeCoverage` and `DescribeEOCoverageSet` operations. Omitting this new parameter or using a value of `False` yields the current behavior whereas a value of `True` only returns the `GridLimits` part of the `GeneralGrid` of the `DomainSet` omitting the remainder.

An additional idea, not used in the proposed solution, is to allow the same subsetting parameters in the `DescribeCoverage` operation than in the `GetCoverage` one subsetting the `DomainSet` itself. This would allow retrieving only the relevant parts of the `DomainSet` in the same way as the `RangeSet`.

Note that a general XPath extension to WCS was discussed and proposed by the WCS.SWG. This extension would allow clients to retrieve any subset of XML responses according to XPath queries. However, this proposed extension is being held back by the OGC Architecture Board as the proposal is seen as potentially beneficial to more OGC services and should get a broader scope.

Parameter subsetting

The requirement of parameter subsetting is simply satisfied by the band subsetting extension of WCS.

Temporal and spatial subsetting

Trimming and slicing of any axis defined in the `DomainSet` is standardized in the WCS core. The difficulty with the data at hand is to define a suitable `DomainSet` that contains time and spatial axes in order to support both temporal and spatial subsetting requirements.

The 2B-GEOPROF data provides the parameters in a 2D grid with size `125 x nray` where `nray` is typically around 20.000. Each of the ~20.000 columns has latitude, longitude, and time values that

are equal for the whole column. On the other side, each grid point has its own height value, i.e., there is no uniformity in the 125 height elements.

The challenge is to model the 2D data grid in the 4D CRS grid.

This is not possible using CIS 1.0, even including the ReferenceableGridCoverage extension, as there is no way to "get rid" of CRS axes in order to describe the 2D grid in the 4D CRS.

Using CIS 1.1 there is the `GeneralGrid` together with `DisplacementAxisNest` and `IrregularAxis`. However, only the `DisplacementAxisNest` supports to "get rid" of axes but which would unnecessarily 'blow up' (i.e. increasing the size to the point of being unmanagable) the `DomainSet` description as values for each individual 4D CRS grid element would have to explicitly be provided. Or in other words, the conformity of the 2D columns cannot be exploited and instead of providing latitude, longitude, and time each "only" ~20.000 times this would blow up to multiplying this 125 times:

`DomainSet` using `DisplacementAxisNest`

```
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979&2=http://www.opengis.net/def
/crs/OGC/0/AnsiDate" axisLabels="Lat Long h date">

    <DisplacementAxisNest axisLabels="Lat Long date h" uomLabels="deg deg d m">
      <P><C>-69.35112</C><C>171.2182</C><C>46293.594+0.0</C><C>24927</C></P>
      <P><C>-69.35112</C><C>171.2182</C><C>46293.594+0.0</C><C>24687</C></P>
      ...
      <P><C>-69.35112</C><C>171.2182</C><C>46293.594+0.0</C><C>-4812</C></P>

      <P><C>-69.35994</C><C>171.20651</C><C>46293.594+0.15999997</C><C>24929
</C></P>

      <P><C>-69.35994</C><C>171.20651</C><C>46293.594+0.15999997</C><C>24689
</C></P>

      ...
      <P><C>-69.35994</C><C>171.20651</C><C>46293.594+0.15999997</C><C>-4810
</C></P>

      ... 125 x 20753 <P> elements with 4 <C> elements each in total

    </DisplacementAxisNest>

    <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D" axisLabels
="i j">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="20752"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="124"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>
```

In order not to unnecessarily blow up the `DomainSet` an `IrregularAxisNest` element similar to `DisplacementAxisNest` and allowing to exploit the conformity of the columns is proposed as follows:

```

<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSG/0/4979&2=http://www.opengis.net/def
/crs/OGC/0/AnsiDate" axisLabels="Lat Long h date">

    <IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg d">
      <P><C>-69.35112</C><C>171.2182</C><C>46293.594+0.0</C></P>
      <P><C>-69.35994</C><C>171.20651</C><C>46293.594+0.15999997</C></P>

      ... 20753 <P> elements with 3 <C> elements each in total

    </IrregularAxisNest>

    <DisplacementAxisNest axisLabels="h" uomLabels="m">
      <P><C>24927</C></P>
      <P><C>24687</C></P>

      ... 125 x 20753 <P> elements with 1 <C> element each in total

    </DisplacementAxisNest>

    <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D" axisLabels
="i j">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="20752"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="124"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>

```

Using this `IrregularAxisNest` extension would allow instead of 125×20753 `P` elements with 4 `C` elements each to have only 20753 `P` with 3 `C` elements and 125×20753 `P` with one `C` elements, resulting in 2.656.384 instead of 10.376.500 elements or a reduction of almost 75%.

In combination with our proposal from the "Retrieval of metadata" using OWC, the resulted service achieved a really elegant solution available. The `DomainSet` axes are provided as coverages themselves via OWC mechanism. The contents of the `DisplacementAxisNest` element are provided as an `Index2D` coverage holding all the height values and the `IrregularAxisNest` element as three `Index1D` coverages corresponding to latitude, longitude, and date. This way the reported `DomainSet` stays small while supporting all required functionality even subsetting in the domain.

```

<DomainSet>
  <GeneralGrid axisLabels="Lat Long h date" srsName="http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979&amp;2=http://www.opengis.net
/def/crs/OGC/0/AnsiDate">
    <DisplacementAxisNest axisLabels="h" uomLabels="m">
      <owc:offering>
        <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&amp;version=2.1.0&amp;request=GetCoverage&amp;coverageId=
_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06_height&amp;format=image/tiff"
method="GET" type="image/tiff"/>
      </owc:offering>
    </DisplacementAxisNest>
    <IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg d">
      <owc:offering>
        <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&amp;version=2.1.0&amp;request=GetCoverage&amp;coverageId=
_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06_latitude&amp;format=text/csv"
method="GET" type="text/csv"/>
      </owc:offering>
      <owc:offering>
        <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&amp;version=2.1.0&amp;request=GetCoverage&amp;coverageId=
_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06_longitude&amp;format=text/csv"
method="GET" type="text/csv"/>
      </owc:offering>
      <owc:offering>
        <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&amp;version=2.1.0&amp;request=GetCoverage&amp;coverageId=
_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06_profile_time&amp;format=text/csv"
method="GET" type="text/csv"/>
      </owc:offering>
    </IrregularAxisNest>
    <GridLimits axisLabels="i j" srsName=
"http://www.opengis.net/def/crs/OGC/0/Index2D">
      <IndexAxis axisLabel="i" lowerBound="0" upperBound="124"/>
      <IndexAxis axisLabel="j" lowerBound="0" upperBound="20752"/>
    </GridLimits>
  </GeneralGrid>
</DomainSet>

```

Another potential option is to have only lat, long, and time in the `DomainSet` as subsetting in height is not needed on the service level. That would mean having a 1D data grid in a 3D CRS grid and height in the `RangeType`. However, this would result in 125×4 elements in the `RangeType`, 3 parameters and the height itself for 125 different heights, and a need to specify how they relate to each other, i.e., which parameter values correspond to which height value. This idea is discarded as impractical.

The adjusted schema to support the `IrregularAxisNest` extension can be found in Appendix A.

Overviews

The requirement on providing overviews targets performance optimization. A critical metric for performance is "First view" but also update after user interaction. Retrieving only overviews or lower resolution of the data reduces loading and rendering time. However, defining a general downsampling or downscaling algorithm suitable for any irregular grid is not a simple task. It seems easier to define an algorithm to map the irregular grid to a regular one and use the well-known scaling and interpolation extension on them.

The converted coverage has the number of axes from the data array or the mathematical dimensions, i.e., 2 in our case. The axes mapping from data array to CRS or physical dimensions needs to be defined in the `GetCoverage` operation.

The coverage description needs to include these details:

- resolution in all CRS axes, i.e., the minimum distance needed on each axis to correctly represent the data
- size of data array - provided in `GridLimits` element
- bounding box or maxima in each CRS axis - provided in `Envelope` element
- mapping from CRS to data array, i.e., how are the axes related - provided in axis elements like `IrregularAxisNest` or `DisplacementAxisNest` in `GeneralGrid`

The `GetCoverage` operation needs to be extended by these parameters:

- `changeType` - `boolean` to indicate type conversion
- `axesMapping` - mapping for each axis, i.e., `i` or `x` shall be `time` or `lat` or `long` and `j` or `y` shall be `height`, e.g., `i:date,j:h` there should be a default specified

By default, each axis will use its advertised resolution. This can be overwritten by using the mechanics of the scaling extension, e.g. via the `size` or `resolution` parameters.

Via this type conversion still the original potentially interpolated values are retrieved. Another optimization is to ask for complete renderings from the server but this is out of scope here and left to the Web Map Service (WMS). An integration like this between WCS and WMS could be addressed in one of the next testbeds.

CloudSat Swath WCS

The CloudSat Swath WCS was implemented with all the standard operations, including `GetCapabilities`, `DescribeCoverage`, `DescribeEOCoverageSet`, and `GetCoverage` with temporal, spatial, and/or other domain subsetting. Full example WCS requests and responses can be found in Appendix A.

11.1.4. CloudSat Swath WCS Implementation II

This section describes the implementation of CloudSat Swath WCS done by the team from Arizona State University.

Modeling CloudSat Swath Coverage with CIS 1.1

The CloudSat Swath Coverage can be further encoded in GML for publishing as follows, referring to version 3.2.1 of the OpenGIS Geography Markup Language (GML) Encoding Standard:

```
<CloudSatSwathProfile xmlns="http://www.opengis.net/app"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/app ./CoverageExamples.xsd">
  <gml:domainSet>
    <gml:Grid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>20678 125</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>atrack xtrack</gml:axisLabels>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponents>
            <Timestamp uom="urn:x-si:v1999:uom:seconds">template</Timestamp>
            <Longitude uom="urn:x-si:v1999:uom:degrees">template</Longitude>
            <Latitude uom="urn:x-si:v1999:uom:degrees">template</Latitude>
            <SwathData>template</SwathData>
          </gml:valueComponents>
        </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:tupleList>
        0.000000,167.553329,-69.744614,-2474.000000
        0.160000,167.541229,-69.753403,-8888.000000
        ... ..
        3308.320068,-106.911194,81.586121,16953.000000
      </gml:tupleList>
    </gml:DataBlock>
  </gml:rangeSet>
</CloudSatSwathProfile>
```

Or encoded in GML for publishing as follows, referring to version 1.1 of the CIS (Coverage Implementation Schema) encoding standard:

```
<GeneralGridCoverage
  xmlns="http://www.opengis.net/cis/1.1/gml"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:swe="http://www.opengis.net/swe/2.0"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/cis/1.1/gml ../cisAll.xsd">
<DomainSet>
  <GeneralGrid srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
    axisLabels="atrack xtrack">
    <RegularAxis axisLabel="atrack" lowerBound="0"
      upperBound="20677" uomLabel="scan" resolution="1" />
    <RegularAxis axisLabel="xtrack" lowerBound="0"
      upperBound="124" uomLabel="scan" resolution="1" />
  </GeneralGrid>
</DomainSet>

<RangeSet>
  <DataBlock>
    <CV>
      <V>0.000000</V>
      <V>-69.744614</V>
      <V>167.553329</V>
      <V>-2474.000000</V>
    </CV>

    ... ..

  </DataBlock>
</RangeSet>

<RangeType>
  <swe:DataRecord>
    <swe:field name="Timestamp">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="second" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="Latitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="degree" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="Longitude">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="degree" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="SwathData">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="some_unit" />
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>

```

```

    </swe:field>
  </swe:DataRecord>
</RangeType>
</GeneralGridCoverage>

```

Following the CIS 1.1 example http://schemas.opengis.net/cis/1.1/gml/examples-1.1/45_2D_distorted.xml, we can encode the swath coverage data as follows:

```

<GeneralGridCoverage xmlns="http://www.opengis.net/cis/1.1/gml"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:swe="http://www.opengis.net/swe/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/cis/1.1/gml ../cisAll.xsd">
  <DomainSet>
    <GeneralGrid srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
      axisLabels="Time xTrack Lat Long">
      <DisplacementAxisNest axisLabels="Time xTrack Lat Long"
        uomLabels="second unit deg deg">
        <P>
          <C>0.0000</C>
          <C>0</C>
          <C>-69.744614</C>
          <C>167.553329</C>
        </P>
        <P>
          <C>0.1600</C>
          <C>0</C>
          <C>-69.744614</C>
          <C>167.553329</C>
        </P>
        ... ..
        <P>
          <C>3308.0000</C>
          <C>124</C>
          <C>-69.744614</C>
          <C>167.553329</C>
        </P>
      </DisplacementAxisNest>
      <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
        axisLabels="i j">
        <IndexAxis axisLabel="i" lowerBound="0" upperBound="20677" />
        <IndexAxis axisLabel="j" lowerBound="0" upperBound="124" />
      </GridLimits>
    </GeneralGrid>
  </DomainSet>

  <RangeSet>
    <DataBlock>
      <CV>
        <V>1234.000000</V>
      </CV>
    </DataBlock>
  </RangeSet>

```

```

        <V>-2474.000000</V>
    </CV>
    ... ..
</DataBlock>

</RangeSet>
<RangeType>
  <swe:DataRecord>
    <swe:field name="Height">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="m" />
      </swe:Quantity>
    </swe:field>
    <swe:field name="SwathData">
      <swe:Quantity
        definition="http://www.opengis.net/def/dataType/OGC/0/double">
        <swe:uom code="some_unit" />
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</RangeType>
</GeneralGridCoverage>

```

Inspired by Stephan Meißl, below is a proposed version of data provision that reduces the redundancy of `DomainSet` in comparison to above versions, and provides all the needed information for visualization in one coverage/request.

```

<?xml version="1.0" encoding="UTF-8"?>
<wcs21:CoverageDescriptions
  xmlns:wcs21='http://www.opengis.net/wcs/2.1/gml'
  xmlns:wcs20='http://www.opengis.net/wcs/2.0'
  xmlns='http://www.opengis.net/cis/1.1/gml'
  xmlns:gml='http://www.opengis.net/gml/3.2'
  xmlns:swe='http://www.opengis.net/swe/2.0'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:schemaLocation='http://www.opengis.net/swe/2.0
http://schemas.opengis.net/sweCommon/2.0/swe.xsd
http://www.opengis.net/wcs/2.1/gml
http://schemas.opengis.net/wcs/2.1/gml/wcsAll.xsd
http://www.opengis.net/wcs/2.0
http://schemas.opengis.net/wcs/2.0/wcsAll.xsd'>
  <wcs21:CoverageDescription gml:id="_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
    <Envelope srsName="http://www.opengis.net/def/crs/OGC/0/Index2D" axisLabels=
"Lat Long date xtrack" srsDimension="4">
      <AxisExtent axisLabel="Lat" uomLabel="deg" lowerBound="-81.820435"
upperBound="81.82048"/>
      <AxisExtent axisLabel="Long" uomLabel="deg" lowerBound="-105.84433"
upperBound="171.2182"/>

```

```

    <AxisExtent axisLabel="date" uomLabel="d" lowerBound="46293.594"
upperBound="49613.914"/>
    <AxisExtent axisLabel="xtrack" uomLabel="" lowerBound="0" upperBound="124
"/>
  </Envelope>
  <DomainSet>
    <GeneralGrid srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
axisLabels="Lat Long date xtrack">
      <IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg d">
        <P><C/><C/><C/><C/></P>
      </IrregularAxisNest>
      <DisplacementAxisNest axisLabels="xtrack" uomLabels="">
        <P><C/></P>
      </DisplacementAxisNest>
      <GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index2D"
axisLabels="i j">
        <IndexAxis axisLabel="i" lowerBound="0" upperBound="20752"/>
        <IndexAxis axisLabel="j" lowerBound="0" upperBound="124"/>
      </GridLimits>
    </GeneralGrid>
  </DomainSet>
  <RangeType>
    <swe:DataRecord>
      <swe:field name="Height">
        <swe:Quantity definition="TBD">
          <swe:description>TBD</swe:description>
          <swe:nilValues>
            <swe:nilValues>
              <swe:nilValue reason="fill value">-8888</swe:nilValue>
            </swe:nilValues>
          </swe:nilValues>
          <swe:uom code="dBZe"/>
          <swe:constraint>
            <swe:AllowedValues>
              <swe:interval>-32768 32767</swe:interval>
            </swe:AllowedValues>
          </swe:constraint>
        </swe:Quantity>
      </swe:field>
      <swe:field name="Radar_Reflectivity">
        <swe:Quantity definition="TBD">
          <swe:description>TBD</swe:description>
          <swe:nilValues>
            <swe:nilValues>
              <swe:nilValue reason="fill value">15360</swe:nilValue>
            </swe:nilValues>
          </swe:nilValues>
          <swe:uom code="dBZe"/>
          <swe:constraint>
            <swe:AllowedValues>
              <swe:interval>-32768 32767</swe:interval>
            </swe:AllowedValues>
          </swe:constraint>
        </swe:Quantity>
      </swe:field>
    </swe:DataRecord>
  </RangeType>
</swe:DataRecord>

```

```

        </swe:AllowedValues>
    </swe:constraint>
</swe:Quantity>
</swe:field>
<swe:field name="Gaseous_Attenuation">
    <swe:Quantity definition="TBD">
        <swe:description>TBD</swe:description>
        <swe:nilValues>
            <swe:nilValues>
                <swe:nilValue reason="fill value">15360</swe:nilValue>
            </swe:nilValues>
        </swe:nilValues>
        <swe:uom code="dBZe"/>
        <swe:constraint>
            <swe:AllowedValues>
                <swe:interval>-32768 32767</swe:interval>
            </swe:AllowedValues>
        </swe:constraint>
    </swe:Quantity>
</swe:field>
<swe:field name="CPR_Cloud_mask">
    <swe:Quantity definition="TBD">
        <swe:description>TBD</swe:description>
        <swe:nilValues>
            <swe:nilValues>
                <swe:nilValue reason="fill value">-99</swe:nilValue>
            </swe:nilValues>
        </swe:nilValues>
        <swe:uom code="various"/>
        <swe:constraint>
            <swe:AllowedValues>
                <swe:interval>-128 127</swe:interval>
            </swe:AllowedValues>
        </swe:constraint>
    </swe:Quantity>
</swe:field>
</swe:DataRecord>
</RangeType>
<wcs20:ServiceParameters>
    <wcs20:CoverageSubtype>GeneralGridCoverage</wcs20:CoverageSubtype>
    <wcs20:nativeFormat>application/x-hdf</wcs20:nativeFormat>
</wcs20:ServiceParameters>
</wcs21:CoverageDescription>
</wcs21:CoverageDescriptions>

```

Operation of Subsetting

For example, the testbed participants investigated the 2B-GEOPROF data starting at 12:44:29 pm, Jan 1, 2017 (UTC), and lasting for 3308.320068 seconds.

Subsetting along atrack:

Figure 12 is a visualization of locations of data bins (atrack):

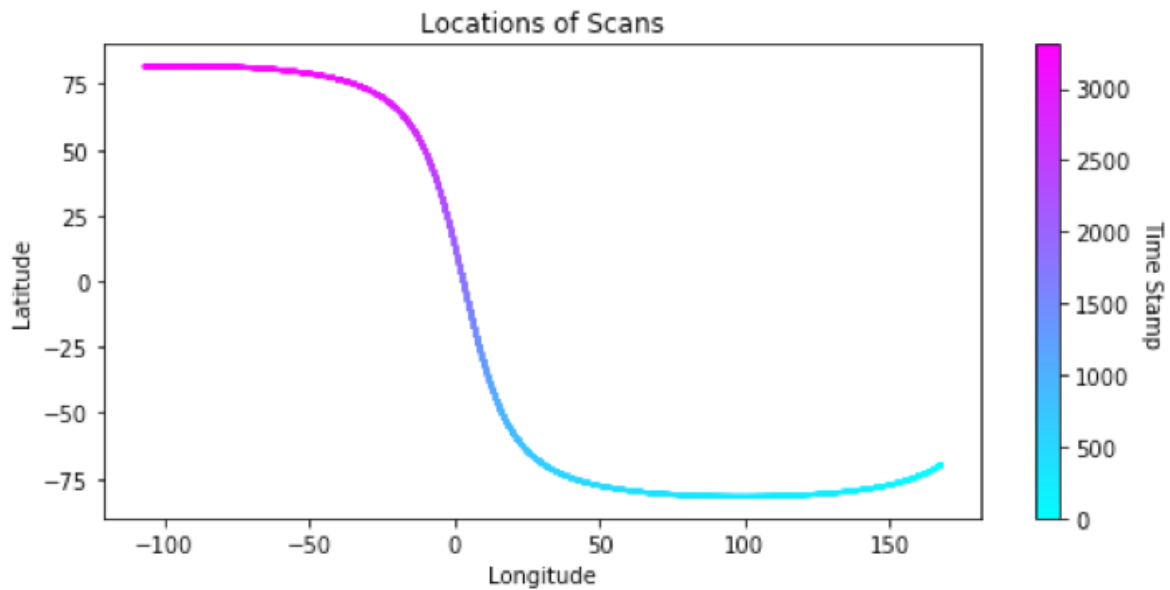


Figure 12. Scanning swath

From the visualization, the locations of the profile were found to be consistent with timestamps. So, by providing the start time and end time, the data can be easily subset. However, since the locations present a sine-function-like pattern, subsetting by specifying the minimum and maximum longitude and/or latitude may split the original grid into multiple ones. This may require some additional handling. One method to solve the problem is to sort the subset grids by time stamp and merge them sequentially along horizontal (atrack) direction, because they still share the same xtrack-dimension.

Subsetting along xtrack:

The easiest way to subset xtrack is to provide a range of xtrack number. However, it might not be meaningful for real applications. A better choice could be to assign a range of height of profile. Whereas, the heights in different data bins that share the same xtrack can vary. A visualization of top height of data bins along atrack is presented in Figure 13.

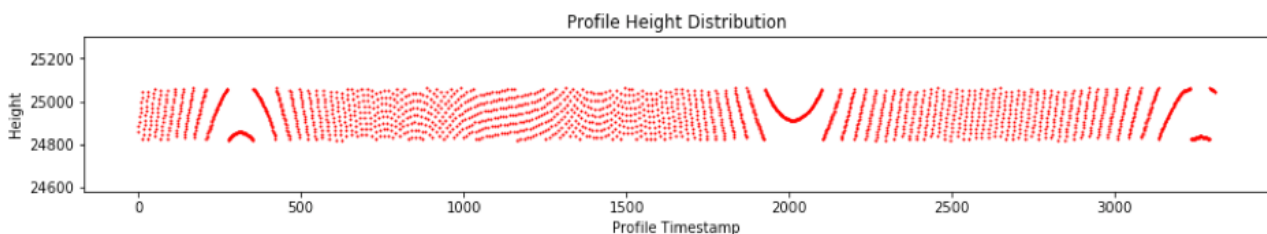


Figure 13. Scanning swath

Since data points are not neatly aligned in the same xtrack, subsetting by height may result in irregular grids. The solution is described by the following steps:

1. Keep the size of the original grid
2. Fill the filtered data points by missing value

3. Remove the top and bottom xtracks that only consist of missing value.

CloudSat Profile WCS

Data Provision (Version 1)

CloudSat Profile GEOPROF-2B is published, based on the coverage model based on the CF convention proposed earlier:

```
dimensions:
  atrack = 20678;
  xtrack = 125;

variables:
  double time(atriack) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;

  float lat(atriack) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;

  float lon(atriack) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;

  float swath_data(atriack, xtrack) ;
    swath_data:coordinates = "time lon lat" ;
```

This Data is published by WCS 2.0.1 whose GetCapabilities request is available at <http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=GetCapabilities>. The XML snippet below shows the EOCoverageSet information of the published data:

```
<wcseo:DatasetSeriesSummary>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-106.91119384765625 -81.81893920898438</ows:LowerCorner>
    <ows:UpperCorner>168.80332946777344 81.82908416748047</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <wcseo:DatasetSeriesId>swath__cs_2b_geoprof_dss</wcseo:DatasetSeriesId>
  <gml:TimePeriod gml:id="swath__cs_2b_geoprof_dss__timeperiod">
    <gml:beginPosition>2017-01-01T13:04:14.558Z</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:59:22.878Z</gml:endPosition>
  </gml:TimePeriod>
</wcseo:DatasetSeriesSummary>
```

Since the data is indexed by *atriack*. It has been split into 20678 granules. The information of each granule is described in DescribeEOCoverageSet XML document. Its DescribeEOCoverageSet request

goes

as

follows

http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swath__cs_2b_geoprof_dss. In the beginning part of each granule (see the XML snippet below), latitude and longitude is contained in label `<gml:lowerCorner>` respectively, and time is contained in label `<gml:beginPosition>`. The swath data (Radar_Reflectivity for this showcase) can be obtained by setting the CoverageId contained in label `<wcs:CoverageId>` and sending GetCoverage request accordingly. In CoverageId, the number right after the dot/period indicates the value of *atrack*.

```
<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>-69.89391326904297 167.34603881835938</gml:lowerCorner>
    <gml:upperCorner>-69.88391326904296 168.59603881835938</gml:upperCorner>
    <gml:beginPosition>2017-01-01T13:04:17.278Z</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:04:17.278Z</gml:endPosition>
  </gml:EnvelopeWithTimePeriod>
</gml:boundedBy>
<wcs:CoverageId>swath__cs_2b_geoprof_1K_granule_cs_2b_geoprof_1K.1</wcs:CoverageId>
```

Here is an example of GetCoverage request to obtain the swath data in GML format (GeoTIFF by default) with *atrack* being 20678: http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=GetCoverage&CoverageId=swath__cs_2b_geoprof_granule_cs_2b_geoprof.20678&format=gml, which contains 125 values following the *xtrack*. The values can be found in label `<gml:DataBlock>`.

While it takes a lot of processing time to describe all the information of 20678 scans in a single DescribeEOCoverageSet XML document, it could be very slow to use your browser to view the request directly. There is a subsampling version of the original dataset for quick inspection purpose with 1,000 scans (dimension of *atrack* = 1000). Its DescribeEOCoverageSet and GetCoverage requests are listed as follows:

http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swath__cs_2b_geoprof_1K_dss

http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=GetCoverage&CoverageId=swath__cs_2b_geoprof_1K_granule_cs_2b_geoprof_1K.1&format=gml

Data Provision (Version 2)

The strategy of publishing the CloudSat coverage model has been changed. Instead of publishing the data in many granules, they are provided as a whole piece this time. This greatly improved the performance of data transferring and processing in both back-end and front-end.

It follows the CF convention as well:

```

dimensions:
  atrack = 20678;
  xtrack = 125;

variables:
  double time(atriack) ;
    time:standard_name = "time" ;
    time:units = "<units> since <datetime string>" ;
    time:calendar = "gregorian" ;

  float lat(atriack) ;
    lat:standard_name = "latitude" ;
    lat:units = "degrees_north" ;

  float lon(atriack) ;
    lon:standard_name = "longitude" ;
    lon:units = "degrees_east" ;

  float height(atriack, xtrack) ;
    time:standard_name = "height" ;
    time:units = "m" ;

  float swath_data(atriack, xtrack) ;
    swath_data:coordinates = "time lon lat" ;

```

This Data is published by a WCS 2.0.1 with a GetCapabilities request that is available at <http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=GetCapabilities>. The XML snippet below shows one of the variable information of the published data:

```

<wcs:CoverageSummary>
  <wcs:CoverageId>cs_2b_geoprof:CPR_Cloud_mask</wcs:CoverageId>
  <wcs:CoverageSubtype>RectifiedGridCoverage</wcs:CoverageSubtype>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-106.91119384765625 -81.81893920898438</ows:LowerCorner>
    <ows:UpperCorner>167.55332946777344 81.81908416748047</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <ows:BoundingBox crs="http://www.opengis.net/def/crs/OGC/0/Index2D">
    <ows:LowerCorner>0 0</ows:LowerCorner>
    <ows:UpperCorner>20677 124</ows:UpperCorner>
  </ows:BoundingBox>
  <gml:TimePeriod gml:id="cs_2b_geoprof:CPR_Cloud_mask__timeperiod">
    <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
  </gml:TimePeriod>
</wcs:CoverageSummary>

```

The coverageId (<wcs:CoverageId>), WGS84 bounding box (<ows:WGS84BoundingBox>), data grid bounding box (<ows:BoundingBox>), and time coverage period (<gml:TimePeriod>) are present in the

GetCapabilities document. CoverageId is constructed by the dataset name and the variable name, connected with a colon. The WGS84 bounding box is only to help people identify the coverage of the satellite scanning trail on the Earth's surface. The data grid itself, a vertical profile, could not correctly project onto the Earth. Therefore the data grid coverage/size is directly recorded in the `<ows:BoundingBox>` tag. `<gml:TimePeriod>` contains the time information of first scanned and last scanned columns of the dataset.

Using the coverageId, the DescribeCoverage document of `cs_2b_geoprof:CPR_Cloud_mask` can be retrieved by http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=DescribeCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask, in which more data grid information is contained. Beside the data range, data unit, and missing value are declared in the document too, presenting in the following XML snippet:

```
<gml:rangeType>
  <swe:DataRecord>
    <swe:field name="CPR_Cloud_mask">
      <swe:Quantity>
        <swe:description>CPR_Cloud_mask</swe:description>
        <swe:nilValues>
          <swe:nilValue reason="
http://www.opengis.net/def/nil/OGC/0/unknown">-99</swe:nilValue>
        </swe:nilValues>
        <swe:uom code="" />
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>-128 127</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</gml:rangeType>
```

Missing value is contained in `<swe:nilValue>`, data unit is contained in code attribute of tag `<swe:uom>` (missing for this variable), while data range can be found in `<swe:interval>`.

Finally, the data can be retrieved through the GetCoverage request as http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask. It is available as GML 3.2, following the CIS 1.1 standard. The values can be found in tag `<gml:DataBlock>`.

One more thing that needs to be mentioned is that not all the auxiliary/geographic information, like height, time, latitude, and longitude, has been provided along with the swath variable in a single GetCoverage request. Instead, they are available as different variables that are listed in GetCapabilities document. For example, the XML snippet below shows the Height information:

```

<wcs:CoverageSummary>
  <wcs:CoverageId>cs_2b_geoprof:Height</wcs:CoverageId>
  <wcs:CoverageSubtype>RectifiedGridCoverage</wcs:CoverageSubtype>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-106.91119384765625 -81.81893920898438</ows:LowerCorner>
    <ows:UpperCorner>167.55332946777344 81.81908416748047</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <ows:BoundingBox crs="http://www.opengis.net/def/crs/OGC/0/Index2D">
    <ows:LowerCorner>0 0</ows:LowerCorner>
    <ows:UpperCorner>20677 124</ows:UpperCorner>
  </ows:BoundingBox>
  <gml:TimePeriod gml:id="cs_2b_geoprof:Height__timeperiod">
    <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
  </gml:TimePeriod>
</wcs:CoverageSummary>

```

Data Subsetting

Data subsetting is implemented based on the original data axes. Users can find the names of axes and corresponding ranges by sending the DescribeCoverage request. For example, below is presented an XML snippet from http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=DescribeCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask:

```

<gml:EnvelopeWithTimePeriod axisLabels="nray nbin" srsDimension="2"
  srsName="http://www.opengis.net/def/crs/OGC/0/Index2D">
  <gml:lowerCorner>0 0</gml:lowerCorner>
  <gml:upperCorner>20677 124</gml:upperCorner>
  <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
  <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
</gml:EnvelopeWithTimePeriod>

```

The axes names contained in the attribute `axisLabels` of tag `gml:EnvelopeWithTimePeriod`, and corresponding lower bound and upper bound can be found in tag `<gml:lowerCorner>` and `<gml:upperCorner>`. The `<gml:beginPosition>` and `<gml:endPosition>` elements define the time span of the scans in this dataset.

According to [OGC® Web Coverage Service 2.0 Interface Standard - KVP Protocol Binding Extension](<https://portal.opengeospatial.org/files/09-147r1>), the subsetting request is made by adding Key-Value Pairs (KVPs) in GetCoverage request. For each axis that needs to be subset, add a KVP like `subset=axisLabel(lowerBound, upperBound)`. For example, to get the subset from 50 to 100 in *nray* and from 10 to 20 in *nbin*, the request will be as follows: [http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask&subset=nray\(50,100\)&subset=nbin\(10,20\)](http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask&subset=nray(50,100)&subset=nbin(10,20))

The XML snippet below shows the new spatial and temporal bounds of retrieved dataset:

```

<gml:EnvelopeWithTimePeriod axisLabels="nray nbin" srsDimension="2"
  srsName="http://www.opengis.net/def/crs/OGC/0/Index2D">
  <gml:lowerCorner>50 10</gml:lowerCorner>
  <gml:upperCorner>100 20</gml:upperCorner>
  <gml:beginPosition>2017-01-01T13:04:22.558</gml:beginPosition>
  <gml:endPosition>2017-01-01T13:04:30.558</gml:endPosition>
</gml:EnvelopeWithTimePeriod>

```

Notes: the lowerBound and upperBound in the KVP are both inclusive.

11.1.5. VIIRS Swath Coverages and their Web Coverage Services

Both an extended WCS 2.1 and a WCS REST service for serving Visible Infrared Imaging Radiometer (VIIRS) swath data were implemented by the team from George Mason University.

Data

The surface reflectance (VNP09) of VIIRS was chosen for experiments on serving swath coverages. The VNP09 data are produced by NASA and distributed by the Land Processes Distributed Active Archive Center (LP DAAC). The surface reflectance data in swath structure are separately stored from its associated geolocation data. The VNP09 collection has a granule of 6-minute swath coverage. The data are available from January 2012 to present.

WCS 2.1 Service

The swath data was modeled in CIS 1.1 with a regular grid of the sensor-reference system. It allows subsetting along different dimensions, including spatial and temporal dimensions. Full details and examples can be found in Appendix A.

WCS REST Service

The RESTful WCS service was implemented following the draft specification developed in the testbed with the assistance from the WFS 3 standards team. The WCS REST service allows subsetting along different dimensions, including spatial and temporal dimensions. The responses are in the similar contents as those from WCS 2.1. Full details and examples can be found in Appendix A.

11.2. Implementations of Swath WCS Clients

Several implementations of clients to access WCS Swath Coverages are made available.

11.2.1. CloudSat Profile Visualization

The draft implementation of the EO-WCS Swath client for CloudSat data relies on the **Implementation 1** of the CloudSat WCS server. The table with pros and cons of the two available WCS implementations for CloudSat as follows.

Table 10. Data fields for CloudSat Level 2 2B-GEOPROF data

WCS Implementation	Pros	Cons	
Implementation #1	<ul style="list-style-type: none"> • vertical granules (1 x 125) harvesting • TIFF visualization • georeferencing available 	<ul style="list-style-type: none"> • it is not possible to get the full CloudSat strip (i.e. 20678 x 125); the [swathcs_2b_geoprof_dss](http://ci.ci.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swathcs_2b_geoprof_dss) document is too big (~115M) to dynamically get all the granules information to place them on the virtual globe • format=image/PNG is available but there are visualization issues 	
Implementation #2	<ul style="list-style-type: none"> • both full CloudSat strip and/or subsetting granules (via &nray, &nbin) are available 	<ul style="list-style-type: none"> • no TIFF/PNG support • no geolocation info available (i.e. <gml:pos>) 	

The client supports discovery and GetCoverage to access the implemented CloudSat services. The first step is about discovery and selection of the EO Swath coverages of interest from the EO Swath products catalogue. The search interface is shown in [Figure 14](#).

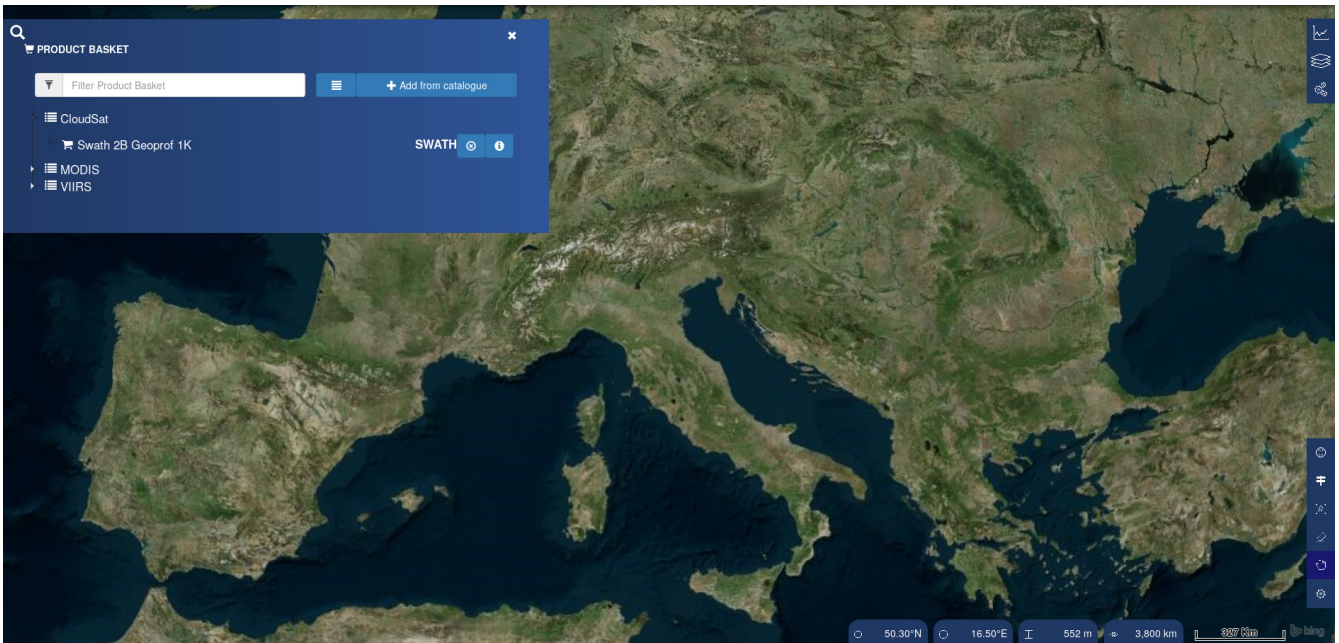


Figure 14. MEO Client Discovery User Interface

11.2.2. CloudSat High Dimension Viewers

A client with capabilities to view high dimension data was implemented by the group from EOX. It provides multiple views. A general view allows subsetting of the data to be shown. A timeslider view allows temporal control. A globe view allows a 2.5D map view, based on Cesium and geotiff.js. An analytics view allows view charts and statistics, based on D3, graphly, and geotiff.js. The launch view of the client is shown in Figure 15. Further details about the client can be found in Appendix A.

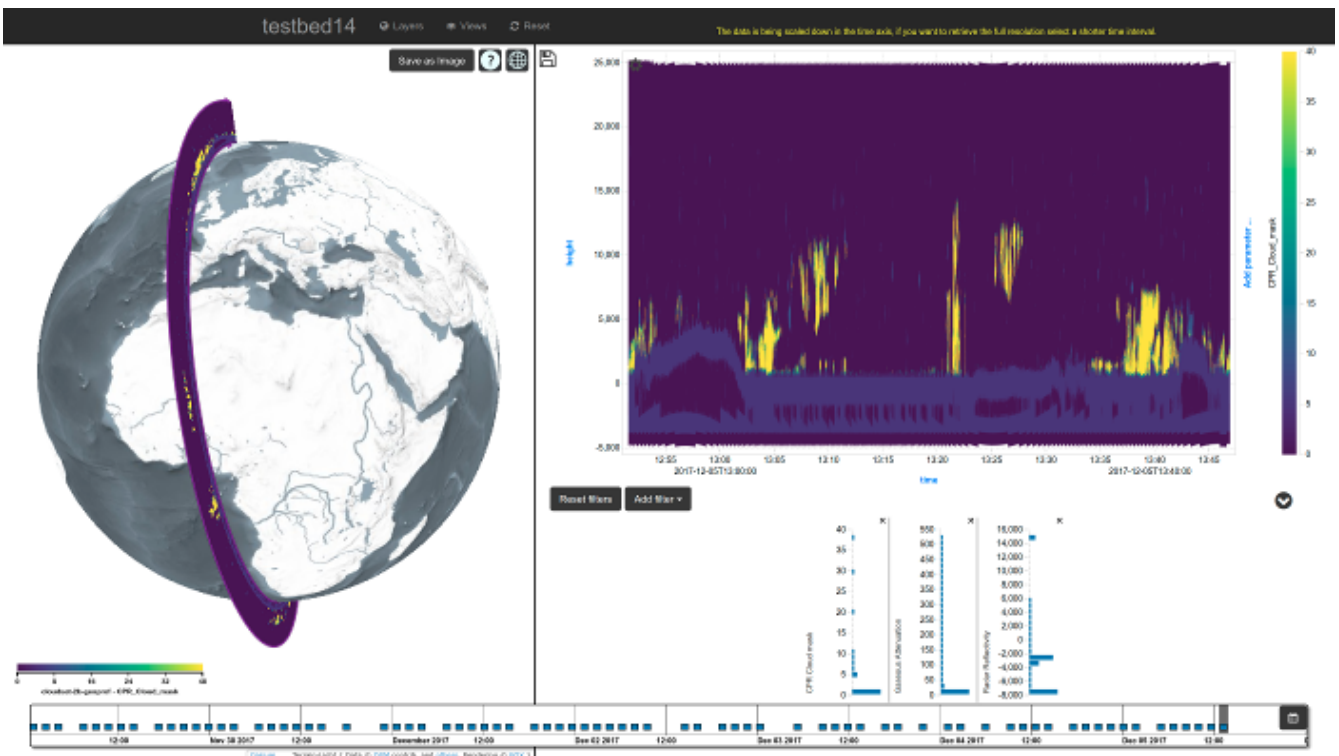


Figure 15. EOX Client User Interface

11.2.3. Use Cases

Access Swath Coverages

Using the client, we can retrieve the whole CloudSat dataset (e.g. http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swath_cs_2b_geoprof_dss). The resulted visualization of the complete coverage without subsetting is shown in [Figure 16](#).

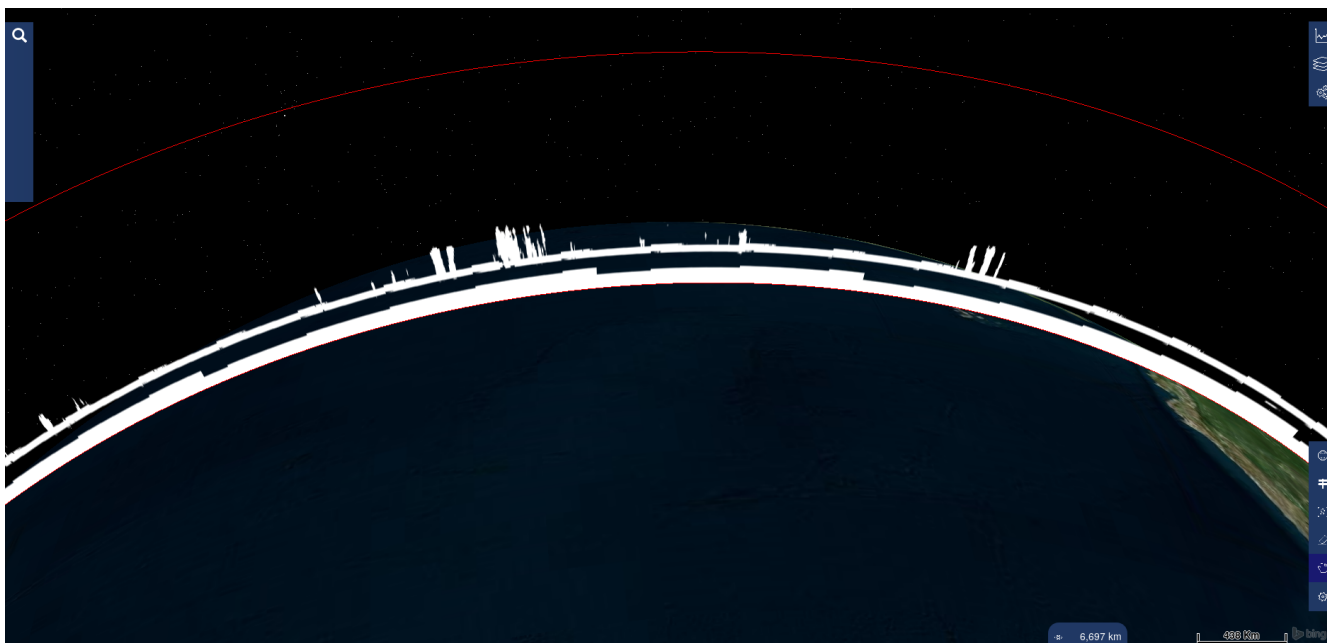


Figure 16. View of a CloudSat swath coverage with MEEO Client

Trimming Swath Coverage

When the EO Swath coverage is selected, a subset of the coverage available at http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swath_cs_2b_geoprof_dss data can be retrieved and the geometries of the first 25 granules are drawn on the virtual globe as shown in [Figure 17](#).

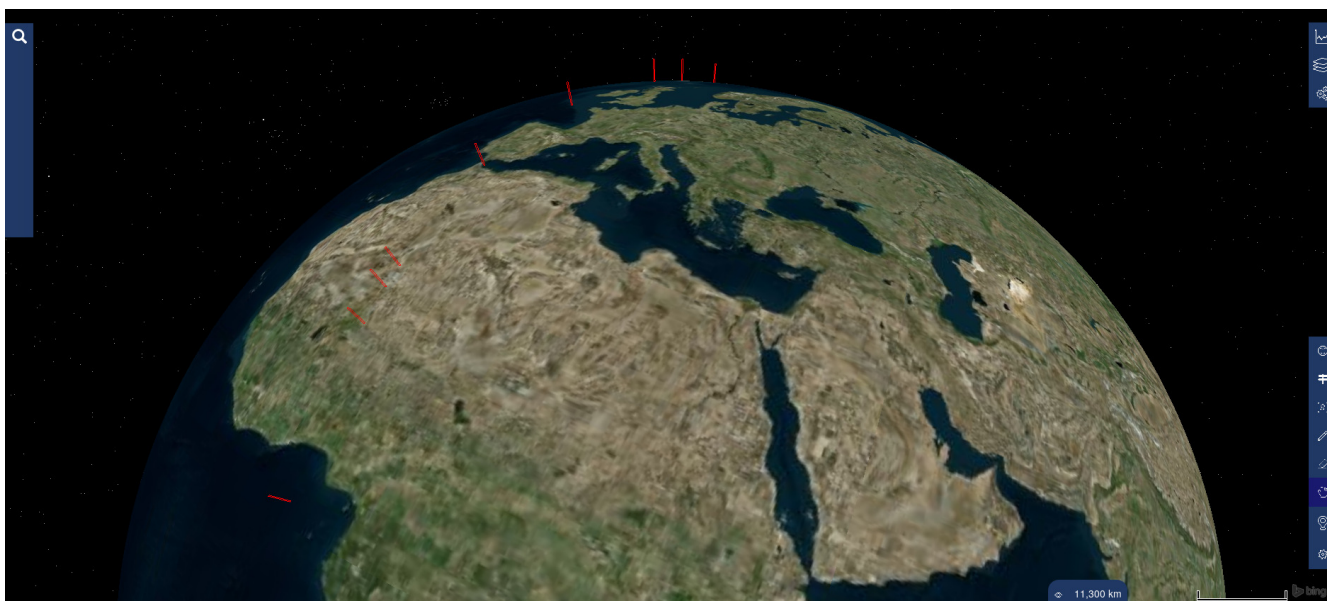


Figure 17. A subset of CloudSat data displayed with a client software

Slicing Swath Coverage

The data access can be sliced along a dimension. One of the example sliced subset of CloudSat swath coverage is shown in [Figure 18](#).

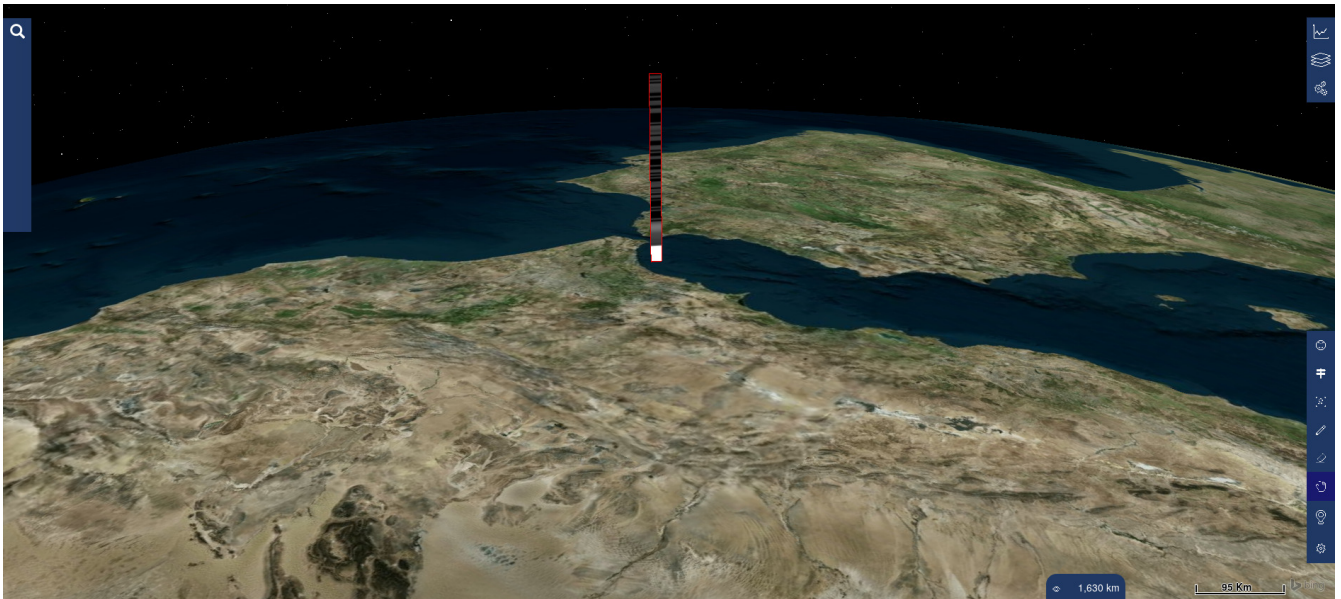


Figure 18. A sliced subset of CloudSat data displayed with a client software

Linked Views of Swath Coverage

The data access and visualization can be done with a high dimension client. Visuals of data, image, and statistics can be displayed and cross-linked, which enable users to interactively access and analyze the data. An example of linked multiple views for a CloudSat swath coverage is shown in [Figure 19](#). An example of a 2.5D view of a CloudSat swath coverage is shown in [Figure 20](#).

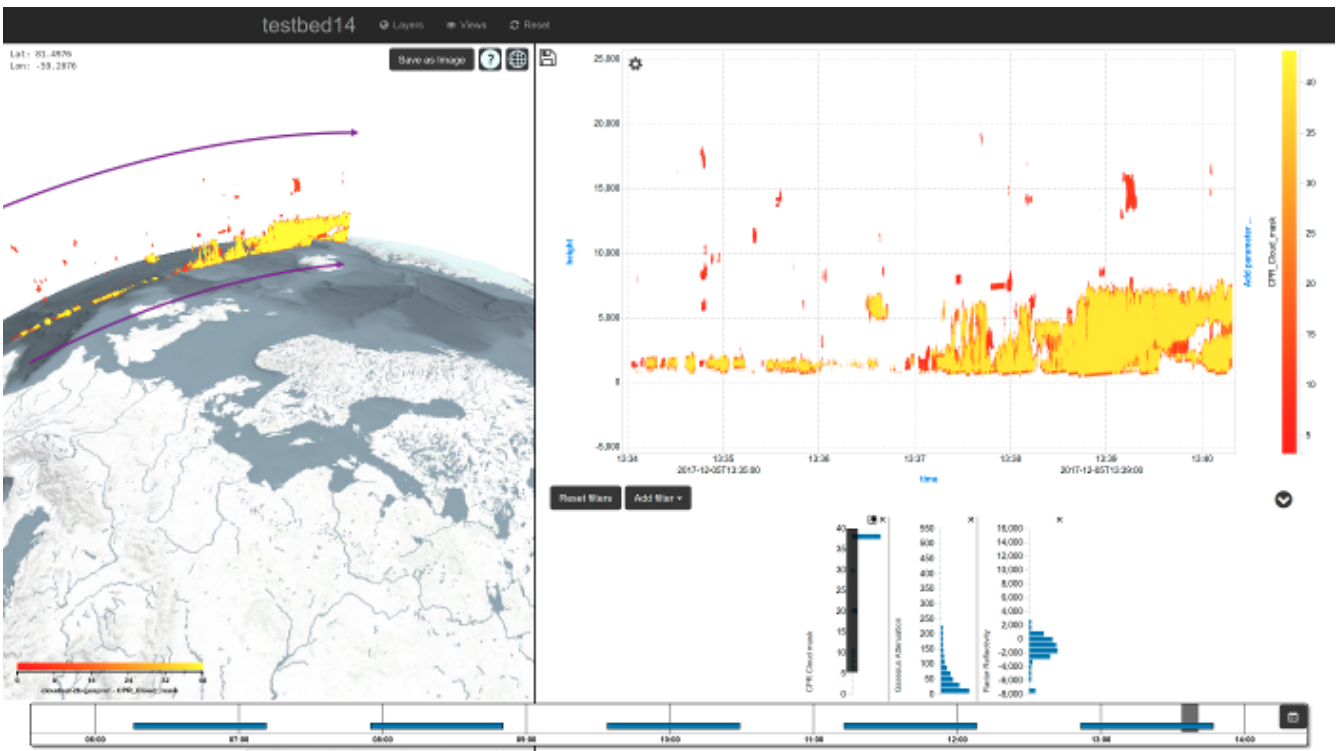


Figure 19. Linked Multiple Views of a CloudSat Swath Coverage using the EOX Client

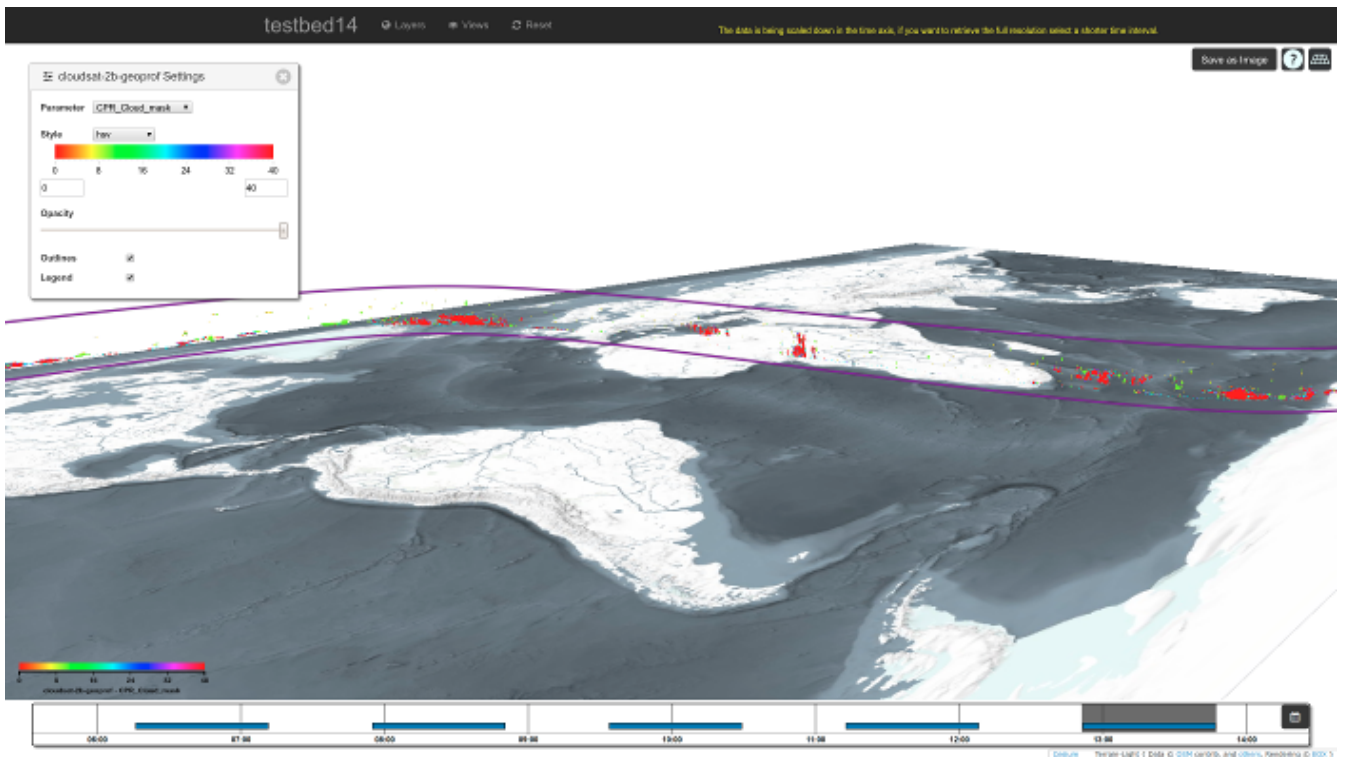


Figure 20. A 2.5D View of a CloudSat Swath Coverage using the EOX Client

Chapter 12. Summary and Discussions

Highlights

NOTE

This section summarizes the experiments done during Testbed 14 for swath coverages. It also covers some of discussion points in depth.

12.1. Summary

The achievements of the Swath Coverage task in this testbed:

- Development of a new WCS Swath Coverage Profile: A set of extension recommendations have been proposed to support the proper description, modeling, and servicing of swath coverages. The services for typical classes of swath coverages have been prototyped to demonstrate the possible extensions to WCS specification. CIS 1.1 was found to be most suitable in extending its capability to fully support different types of swath coverages.
- Development of a new REST API for WCS and WCS Swath Profile: A set of REST APIs were developed following OpenAPI 3.0.1. A prototype REST service has been implemented for serving Swath coverage.
- Development of new Swath Coverage encoding in GeoTiff, NetCDF, and GeoPackage: In addition to modeling of swath coverages with CIS 1.1, CF and NetCDF, GeoTiff, and GeoPackage were also experimented in modeling and encoding typical swath coverages. The swath effort of CF was found effective in modeling swath coverages with its modeling decision tree. The encoding of swath coverages with GeoTIFF and GeoPackages was found to be feasible with extensions.

12.2. Discussions

This section covers some of the key discussion topics.

12.2.1. Modeling Swath Coverages

The first question to be answered for transmitting swath coverages is the model of swath coverage. In this testbed, several commonly accepted standard formats are considered in modeling swath coverage. They are CIS 1.1, CF 1.8, and GeoPackage. The following characteristics of swath coverage summarizes the challenges facing the modeling of swath coverage:

1. The swath data is not georeferenced. The swath data is normally level 1 or level 2 data that is relatively unprocessed. It keeps the original values sensors measured except some of necessary systematic corrections. The requirement of keeping originally sensed values untouched leads to the use of the coordinate system of each sensor to mapping each to a specific location. Each sensor has different coordinate systems to map each pixel to the ground location. Different pixels may have different sizes of ground footprint. The definition of geospatial axes for swath coverage cannot be easily expressed as a simple model as that in simple grid imagery where each axis can be mapped with one resolution step.
2. Different sensors have different coordinate reference systems. Different sensors have different patterns of ground tracks due to their operational mode. For example, each of the MOD07 has a

different footprint resolution of pixels due to the angle of measurement being taken from different positions. VIIRS Level 2 product VNP09 has different sampling rates due to the systematic design of dropping some measurements at the outskirts of the scan and aggregating with different sampling rates depending on the relative location along the scanline towards the nadir.

3. Higher dimension of data is measured with swath coverage. The measurement of the sensor may target at a different height, which leads to the resulting data being in higher dimensions than 2D. The look angle of the sensor further complicates the representation of height dimension. CloudSat data are example of such complicated mapping between pixel and its locations in terms of 3D coordinate (latitude, longitude, and height).

These intrinsic characteristics of sensor coordinate systems project that there is no simple mapping relationship between sensor pixel and its geospatial location. The result may be in a complex mapping relation that cannot be expressed as one simple equation, but a one-to-one lookup table. Each pixel may be represented as a unique coordinate in 2D (Latitude, Longitude) or 3D (latitude, longitude, and height).

In modeling these swath coverages, different approaches may be adopted. Natively, HDF may be used in encoding the swath coverage with different adaptations to fit for different sensors. There is a fundamental difference between the OGC coverage model and the HDF or NetCDF format when it comes to describing the domain. Particularly the mapping from physical dimensions (Latitude, Longitude, Time, Height, etc.) to mathematical dimensions (nray, nbin, i, j, etc.) is quite different.

HDF defines dimensions like nray, nbin, and scalar in the CloudSat example. These dimensions are used by variables or fields which are grouped as GeoField or DataField. Each variable simply defines which dimensions it uses. For example, Latitude uses nray whereas Height uses nray and nbin. Two variables like Latitude and Longitude can simply use the same dimension like nray.

The OGC coverage model, on the other hand, explicitly defines a domain as DomainSet. A coverage provides a value from the RangeType for each coordinate of the domain, the RangeSet. Historically there has been a 1-to-1 mapping between domain axis or physical dimensions and data axis or mathematical dimensions. Only the recently introduced CIS 1.1 has limited support for alternate mappings using the DisplacementAxisNest concept. However, there is currently no way to directly map multiple physical dimensions to one mathematical dimension which is rather simple in HDF.

To overcome this limitation in the OGC coverage model the following approaches have been analyzed:

- Extend GeneralGrid to support dimension mapping
 - Pros
 - Fully compatible to OGC coverage model and thus OGC services
 - Cons
 - Significantly increased size, e.g., about 10MB HDF including three data variables results in about 50MB uncompressed or 7MB compressed XML coverage description
 - Added value of complex coverage description not yet clear
- Use all variables including domain ones in indexed coverages

- Pros
 - Readily supported in existing server software
- Cons
 - Only index based subsetting supported excluding "real" spatial and time subsetting
 - Need to define convention for relation of coverages, e.g., coverage with id ID:lat holds latitude values and the one with ID:long longitude values for the coverage with id ID:Radar_Reflectivity, where axisLabels like nray are matching.
- Use RangeType for irregular domain variables like Height
 - Pros
 - Simpler DomainSet
 - Cons
 - Explosion of RangeType including the need for a convention to describe relations between them, e.g., which variable values correspond to which height value.

It should be noted that the 2018 OGC Technical Committee (TC) meeting held in Orleans, France passed a motion to develop a Sensor Model Registry (SMR) that would include registers for sensor models and their parameters. The SMR will be built and controlled by the OGC Naming Authority (OGC-NA). The purpose of the SMR will be to provide information that can assist applications with transformation of observations from sensor coordinate systems to geodetic/projected coordinate reference systems. Future work could look into how the Sensor Model Registry could support the modeling of Swath Coverages.

12.2.2. WCS REST API for Swath Coverage

At the time of this testbed, there is no official REST API specification for WCS. The quick approach is to develop a REST API following OpenAPI and experience from the WFS 3.0 SWG. Their modularized approach is adopted in developing the WCS REST API. The resources are simplified to consider two types - metadata and coverages (at granule level). The current version of WCS REST API leaves most of WCS operations on coverages as they are. Therefore, the coverage subsetting (trimming and slicing) are left to be parameters to be operated on the coverage resource. This approach may not fully follow the protocol of REST services, but it easily bridges the existing WCS services and the WCS REST services. The mapping of operations on coverage between the conventional WCS and the REST WCS is straightforward. It is more or less like the implementation of KVP binding.

The WCS REST specification was used in implementation of the swath coverage service. The interface for swath coverage is not much different from that for non-swath and rectified coverage. The content passed to parameters is different since swath coverage has its unique properties. One of the unique properties of Swath Coverage is its own coordinate reference system referred to its relative positions among sensors, ground target, and power source (e.g. sun). The relative geometry among these components form the base of the swath coordinate reference system. Another important aspect of swath coverage is that it has a time dimension. These dimensions may be supported in subsetting operations. Therefore, these parameters may be passed in through the subset parameter on operations to each swath coverage.

12.2.3. Visualization of Swath Coverage

The visualization of swath coverage may require the swath coverage to be gridded or geo-rectified to be shown properly along with other georeferenced data. One issue noticed during the use case experiment is impact of the large size of swath coverage encoded as GML. For interactive visualization, it is impossible to retrieve the large file. For example, The swath coverage [http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swathcs_2b_geoprof_dss\[swathcs_2b_geoprof_dss\]](http://cici.lab.asu.edu:8080/geoserver/ows?service=WCS&version=2.0.1&request=DescribeEOCoverageSet&eoId=swathcs_2b_geoprof_dss[swathcs_2b_geoprof_dss]) is too big to allow dynamic harvesting of the strips information (position, coverageId, time etc.). The coverage in gridded, geo-rectified, and compressed format is significantly small in size of data to be transmitted. If tiled service is supported, the interactive preview can be easily supported. The experiments in this testbed identified the differences on requirements for visualization and data access. For data access to swath coverage, it is important to keep the data untouched and all the associated parameters cross-referencable. On the contrary, for visualization to swath coverage, it is important to have the results sent to be in some geo-referenced projections that can be aligned with other georeferenced data easily. Tiling also requires the georectification of swath coverage. Quick views of swath coverage are important for visualization, especially at the stage of browsing many coverages at the same time. The interaction of retrieving raw data value from the visuals needs to have a defined relationship to reach a consensus in converting on-screen location to the point in the coordinate reference system of swath coverage (or sensor).

12.3. Suggestions and Recommendations

The following are recommendations for extensions and/or enhancements to WCS and related OGC standards to better support swath data.

- Allow providing the DomainSet axes information as coverages themselves via OGC Web Services Context (OWC)
- Add IrregularAxisNest element to GeneralGrid in CIS 1.1
- Add parameters to the GetCoverage operation to ask for a type conversion from referenceable to rectified coverage
- Add a special namespace for sensor-specific coordinate systems to model unique swath coordinate reference systems. The referencing computation model may be pre-defined with sensor Web process.

12.4. Relation to other groups

Related groups:

- WFS 3.0: Used their model as the base and example to develop the REST API specification for WCS and WCS Swath Coverage Profile
- NetCDF group: provide examples of encoding different Swath Coverages
- OpenAPI group: provide examples of WCS REST API

Appendix A: Example Swath Coverage Services, their Requests and Responses

This appendix lists some of the example services and their example requests and responses. It does not follow exactly the Test Suite of OGC, but an attempt to test the services and demonstrate their capabilities.

The appendix gives the actual endpoints used to demonstrate the request.

A.1. CloudSat Swath Coverage Service 1

A.1.1. Endpoints

Table 11. Endpoints for CloudSat Swath Coverage WCS

Endpoint	http://ows.eox.at/testbed-14/eoxserver/ows
Implementation Organization:	EOX
Developer or Point of Contact:	Stephan Meißl
Email:	

A.1.2. Adjusted schema for swath coverage WCS

The extended schema for the Swath Coverage WCS implementation is provided here for future use. It should be noted that the examples provided in this section can be validated against these extended XML schemas using `xmllint`.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://www.opengis.net/cis/1.1/gml"
  xmlns:cis="http://www.opengis.net/cis/1.1/gml"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:sml="http://www.opengis.net/sensorml/2.0"
  xmlns:owc="http://www.opengis.net/owc/1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xml:lang="en"
  version="1.1.0">
  <annotation>
    <appinfo>grid.xsd</appinfo>
    <documentation>
      This XML Schema Document encodes grid definitions for the OGC Coverage
      Implementation Schema (CIS) 1.1.
      It consists of the GMLCOV 1.0 definitions plus additions introduced by CIS
      1.1.
      Last updated: 2017-jan-16
      Copyright (c) 2016, 2017 Open Geospatial Consortium.
```


To obtain additional rights of use, visit
<http://www.opengeospatial.org/legal/>.

```
</documentation>
</annotation>

<!-- ===== -->
<!-- Includes and imports -->
<!-- ===== -->
<import namespace="http://www.opengis.net/gml/3.2" schemaLocation=
"http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
<import namespace="http://www.opengis.net/sensorml/2.0" schemaLocation=
"http://schemas.opengis.net/sensorML/2.0/sensorML.xsd"/>
<import namespace="http://www.opengis.net/owc/1.0" schemaLocation=
"http://schemas.opengis.net/owc/1.0/owc.xsd"/>
<include schemaLocation="coverage.xsd"/>

<!-- ===== -->
<!-- Body of this schema -->
<!-- ===== -->
<element name="GeneralGrid" type="cis:GeneralGridType">
  <annotation>
    <documentation>A general n-D grid is defined through a sequence of axes,
each of which can be of a particular axis type.</documentation>
  </annotation>
</element>
<complexType name="GeneralGridType">
  <sequence>
    <element name="Origin" type="cis:DirectPositionType" minOccurs="0"/>
    <choice maxOccurs="unbounded">
      <element ref="cis:Axis"/>
      <element ref="cis:DisplacementAxisNest"/>
      <element ref="cis:IrregularAxisNest"/>
      <element ref="cis:Model"/>
    </choice>
    <element ref="cis:GridLimits" minOccurs="0"/>
  </sequence>
  <attribute name="srsName" type="anyURI" use="required"/>
  <attribute name="axisLabels" type="cis:NameListType" use="required"/>
</complexType>
<!-- ===== -->
<simpleType name="NameListType">
  <annotation>
    <documentation>A name list is an ordered sequence of names. Due to
technical reasons of GML, names are restricted to NCName.</documentation>
  </annotation>
  <list itemType="NCName"/>
</simpleType>
<!-- ===== -->
<complexType name="DirectPositionType">
  <annotation>
    <documentation>Direct positions resemble n-D coordinates indicating where
```

a coverage provides values. For efficiency reason, frequently repeated tag names are kept terse: "dp" for "directPosition", "c" for "directCoordinate". To accommodate all possible coordinate types (such as "1.2" for degrees, "2016-02-29" for dates, etc.) the type is anySimpleType.</documentation>

```

    </annotation>
    <sequence>
      <element name="C" type="anySimpleType" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
<!-- ===== -->
<element name="Axis" type="cis:AxisType"/>
<complexType name="AxisType">
  <attribute name="axisLabel" type="NCName" use="required"/>
</complexType>
<!-- ===== -->
<element name="GridLimits" type="cis:GridLimitsType">
  <annotation>
    <documentation>This is the boundary of the array underlying the grid,
given by its diagonal corner points in integer coordinates. The grid limits can be
omitted in case all axes are of type index axis, because then it repeats the grid
information in a redundant way. The purpose of the axisLabels attribute, which lists
the axis labels of all axisExtent elements in proper sequence, is to enforce axis
sequence also in XML systems which do not preserve document order.</documentation>
  </annotation>
</element>
<complexType name="GridLimitsType">
  <sequence>
    <element ref="cis:IndexAxis" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="srsName" type="anyURI" use="required"/>
  <attribute name="axisLabels" type="cis:NameListType" use="required"/>
</complexType>
<!-- ===== -->
<element name="IndexAxis" type="cis:IndexAxisType" substitutionGroup="cis:Axis">
  <annotation>
    <documentation>An Index Axis is an axis with only integer positions
allowed.</documentation>
  </annotation>
</element>
<complexType name="IndexAxisType">
  <complexContent>
    <extension base="cis:AxisType">
      <attribute name="lowerBound" type="anySimpleType" use="required"/>
      <attribute name="upperBound" type="anySimpleType" use="required"/>
    </extension>
  </complexContent>
</complexType>
<!-- ===== -->
<element name="RegularAxis" type="cis:RegularAxisType" substitutionGroup="
cis:Axis">
  <annotation>

```

```

    <documentation>A Regular Axis is an axis where all direct coordinates are
    at a common distance from its immediate neighbors.</documentation>
  </annotation>
</element>
<complexType name="RegularAxisType">
  <complexContent>
    <extension base="cis:AxisType">
      <sequence>
        <element name="Offset" type="cis:DirectPositionType" minOccurs="0
"/>
      </sequence>
      <attribute name="lowerBound" type="anySimpleType" use="required"/>
      <attribute name="upperBound" type="anySimpleType" use="required"/>
      <attribute name="resolution" type="anySimpleType" use="required"/>
      <attribute name="uomLabel" type="NCName" use="required"/>
    </extension>
  </complexContent>
</complexType>
<!-- ===== -->
<element name="IrregularAxis" type="cis:IrregularAxisType" substitutionGroup=
"cis:Axis">
  <annotation>
    <documentation>An irregular axis enumerates all possible direct position
coordinates. In order to keep frequently repeated tag names short they have been
abbreviated: "dp" for "DirectPosition".</documentation>
  </annotation>
</element>
<complexType name="IrregularAxisType">
  <complexContent>
    <extension base="cis:AxisType">
      <sequence>
        <choice>
          <element name="C" type="anySimpleType" maxOccurs="unbounded"/>
          <element ref="owc:offering"/>
        </choice>
      </sequence>
      <attribute name="uomLabel" type="NCName" use="required"/>
    </extension>
  </complexContent>
</complexType>
<!-- ===== -->
<element name="DisplacementAxisNest" type="cis:DisplacementAxisNestType">
  <annotation>
    <documentation>A DisplacementAxisNest is a group of warped axes where
points on the grid all have their individual direct position coordinates. The
sequenceRule element describes linearization order.</documentation>
  </annotation>
</element>
<complexType name="DisplacementAxisNestType">
  <sequence>
    <choice maxOccurs="unbounded">

```

```

        <element name="P" type="cis:DirectPositionType"/>
        <element name="C" type="anySimpleType"/>
        <element ref="owc:offering"/>
    </choice>
    <element name="SequenceRule" type="gml:SequenceRuleType" minOccurs="0"/>
</sequence>
<attribute name="axisLabels" type="cis:NameListType" use="required"/>
<attribute name="uomLabels" type="cis:NameListType" use="required"/>
</complexType>
<!-- ===== -->
<element name="IrregularAxisNest" type="cis:IrregularAxisNestType">
    <annotation>
        <documentation>A IrregularAxisNest is a group of warped axes where points
on the grid all have their individual direct position coordinates. The sequenceRule
element describes linearization order.</documentation>
    </annotation>
</element>
<complexType name="IrregularAxisNestType">
    <sequence>
        <choice maxOccurs="unbounded">
            <element name="P" type="cis:DirectPositionType"/>
            <element name="C" type="anySimpleType"/>
            <element ref="owc:offering"/>
        </choice>
        <element name="SequenceRule" type="gml:SequenceRuleType" minOccurs="0"/>
    </sequence>
    <attribute name="axisLabels" type="cis:NameListType" use="required"/>
    <attribute name="uomLabels" type="cis:NameListType" use="required"/>
</complexType>
<!-- ===== -->
<element name="Model" type="cis:TransformationModelType" abstract="true">
    <annotation>
        <documentation>A Transformation Model is a grid transformation definition,
given by some algorithm which is not specified further.</documentation>
    </annotation>
</element>
<complexType name="TransformationModelType">
    <attribute name="axisLabels" type="cis:NameListType" use="required"/>
    <attribute name="uomLabels" type="cis:NameListType" use="required"/>
</complexType>
<!-- ===== -->
<element name="TransformationBySensorModel" type=
"cis:TransformationBySensorModelType" substitutionGroup="cis:Model">
    <annotation>
        <documentation>A TransformationBySensorModel is a transformation
definition which is given by a SensorML 2.0 transformation
specification.</documentation>
    </annotation>
</element>
<complexType name="TransformationBySensorModelType">
    <complexContent>

```

```

    <extension base="cis:TransformationModelType">
      <sequence>
        <element name="SensorModel" type="sml:AbstractProcessPropertyType"
maxOccurs="unbounded"/>
        <element name="SensorInstance" type=
"sml:AbstractProcessPropertyType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</schema>

```

A.1.3. Example WCS requests and responses

All examples are based on the selected test data [2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06.hdf](https://github.com/opengeospatial/D007-Swath_Coverage_Engineering_Report/wiki/Use-case-CloudSat_examples/2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06.hdf) [https://github.com/opengeospatial/D007-Swath_Coverage_Engineering_Report/wiki/Use-case-CloudSat_examples/2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06.hdf] from 5 December 2017, the 339th day of 2017.

GetCapabilities

Request

<http://ows.eox.at/testbed-14/eoxserver/ows?service=wcs&request=getcapabilities>

Response

Note that only one coverage is directly visible here, all others are only listed in the advertised [DatasetSeries](#).

GetCapabilities response - Contents part

```
<wcs:Contents>
  <wcs:CoverageSummary>
    <wcs:CoverageId>_2017339123204_61738_CS_2B-
    GEOPROF_GRANULE_P_R04_E06</wcs:CoverageId>
    <wcs:CoverageSubtype>ReferenceableDataset</wcs:CoverageSubtype>
  </wcs:CoverageSummary>
  <wcs:Extension>
    <wcseo:DatasetSeriesSummary>
      <ows:WGS84BoundingBox>
        <ows:LowerCorner>-81.826141 -179.999786</ows:LowerCorner>
        <ows:UpperCorner>81.825691 180.000000</ows:UpperCorner>
      </ows:WGS84BoundingBox>
      <wcseo:DatasetSeriesId>ccloudsat-2b-geoprof</wcseo:DatasetSeriesId>
      <gml:TimePeriod gml:id="ccloudsat-2b-geoprof_timeperiod">
        <gml:beginPosition>2017-11-26T01:26:03Z</gml:beginPosition>
        <gml:endPosition>2017-12-05T13:46:54Z</gml:endPosition>
      </gml:TimePeriod>
    </wcseo:DatasetSeriesSummary>
  </wcs:Extension>
</wcs:Contents>
```

DescribeCoverage

Request

http://ows.eox.at/testbed-14/eoxserver/ows?service=wcs&request=describecoverage&version=2.1.0&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06

Response

- With OGC Web Services Context (OWC): [responseDescribeCoverage.xml](#)
[https://raw.githubusercontent.com/wiki/opengeospatial/D007-Swath_Coverage_Engineering_Report/Use-case-CloudSat_examples/responseDescribeCoverage.xml?token=ARSKhTtA9QxjgZbUJszZjX10WzN6KwHhks5b_OiswA%3D%3D]
- Full response without OWC (Note that this file contains over 40.000 lines and expands to 48MB):
[responseDescribeCoverage_full.xml.gz](#) [https://github.com/opengeospatial/D007-Swath_Coverage_Engineering_Report/wiki/Use-case-CloudSat_examples/responseDescribeCoverage_full.xml.gz]

DescribeCoverage response with OWC

```
<?xml version='1.0' encoding='iso-8859-1'?>
<wcs21:CoverageDescriptions xmlns:cis10="http://www.opengis.net/gmlcov/1.0"
xmlns:cis11="http://www.opengis.net/cis/1.1/gml" xmlns:crs=
"http://www.opengis.net/wcs/crs/1.0" xmlns:eop="http://www.opengis.net/eop/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:int=
"http://www.opengis.net/wcs/interpolation/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:om="http://www.opengis.net/om/2.0" xmlns:owc="http://www.opengis.net/owc/1.0"
xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:rsub=
"http://www.opengis.net/wcs/range-subsetting/1.0" xmlns:scal=
```

```

"http://www.opengis.net/wcs/scaling/1.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:wcs20="http://www.opengis.net/wcs/2.0" xmlns:wcs21=
"http://www.opengis.net/wcs/2.1/gml" xmlns:wcese="
http://www.opengis.net/wcs/wcese/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/wcs/2.1/gml http://schemas.opengis.net/wcs/2.1/gml/wcsAll.xsd
http://www.opengis.net/wcs/wcese/1.1
http://schemas.opengis.net/wcs/wcese/1.1/wcsEOAll.xsd">
  <wcs21:CoverageDescription gml:id="_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
    <cis11:Envelope axisLabels="Lat Long h date" srsDimension="4" srsName=
"http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSG/0/4979&amp;2=http://www.opengis.net
/def/crs/OGC/0/AnsiDate">
      <cis11:AxisExtent axisLabel="Lat" lowerBound="-81.804718005" uomLabel="
deg" upperBound="81.819176005"/>
      <cis11:AxisExtent axisLabel="Long" lowerBound="-105.844330005" uomLabel=
"deg" upperBound="171.218201005"/>
      <cis11:AxisExtent axisLabel="h" lowerBound="-4917" uomLabel="m"
upperBound="25062"/>
      <cis11:AxisExtent axisLabel="date" lowerBound="2017-12-05 12:51:34+00:00"
uomLabel="d" upperBound="2017-12-05 13:46:54+00:00"/>
    </cis11:Envelope>
    <cis11:Metadata>
      <wcese:EOMetadata>
        <eop:EarthObservation gml:id="eop__2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
          <om:phenomenonTime>
            <gml:TimePeriod gml:id="phen_time__2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
              <gml:beginPosition>2017-12-
05T12:51:34Z</gml:beginPosition>
              <gml:endPosition>2017-12-05T13:46:54Z</gml:endPosition>
            </gml:TimePeriod>
          </om:phenomenonTime>
          <om:resultTime>
            <gml:TimeInstant gml:id="res_time__2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
              <gml:timePosition>2017-12-05T13:46:54Z</gml:timePosition>
            </gml:TimeInstant>
          </om:resultTime>
          <om:procedure/>
          <om:observedProperty/>
          <om:featureOfInterest>
            <eop:Footprint gml:id="footprint__2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
              <eop:multiExtentOf>
                <gml:MultiGeometry gml:id=
"multi_geom__2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06">
                  <gml:geometryMember>
                    <gml:LineString gml:id=

```

```

"line_string__2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06">
    <gml:posList>-69.35112000 171.21820100
-72.20041700 166.82487500 -74.59602400 161.78387500 -76.56911500 156.05653400
-78.20853400 149.39813200 -79.54218300 141.63066100 -80.56026500 132.82167100
-81.29612700 122.66144600 -81.71841400 111.47950700 -81.80471800 98.71207400
-81.50271600 86.33461000 -80.84779400 75.18617200 -79.84094200 65.24035600
-78.51689100 56.79996100 -76.86551700 49.63352200 -74.81656600 43.41619500
-72.37244400 38.13395300 -70.43242600 34.94118900 -68.27170600 32.06703200
-65.81163800 29.40571200 -63.02435700 26.94187900 -59.90953100 24.67566100
-56.42108200 22.56763300 -52.50247200 20.58208100 -48.13398400 18.70537400
-38.92371700 15.49581600 -27.36452300 12.29165700 -14.59577400 9.28638400 21.07567600
1.50471200 31.00457800 -0.96263000 39.27206000 -3.33139700 47.68225900 -6.25287200
54.66089600 -9.35745900 57.69982900 -11.01993600 60.43932300 -12.75421800 62.92651000
-14.58539600 65.21673600 -16.55822400 67.70516200 -19.12450800 69.87683900
-21.86276600 71.83847800 -24.90299400 73.58805100 -28.26239800 75.13224800
-31.95952200 76.50169400 -36.07752200 77.67155500 -40.52152600 78.70601700
-45.53061700 79.62034600 -51.29109200 80.37684600 -57.65340000 81.41947200
-72.14802600 81.81917600 -89.00888100 81.51007800 -105.84433000</gml:posList>
    </gml:LineString>
    </gml:geometryMember>
    </gml:MultiGeometry>
    </eop:multiExtentOf>
    </eop:Footprint>
    </om:featureOfInterest>
    <om:result/>
    <eop:metaDataProperty>
    <eop:EarthObservationMetaData>
    <eop:identifier>_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06</eop:identifier>
    <eop:acquisitionType>NOMINAL</eop:acquisitionType>
    <eop:status>ARCHIVED</eop:status>
    </eop:EarthObservationMetaData>
    </eop:metaDataProperty>
    </eop:EarthObservation>
    </wcseo:EOMetadata>
    </cis11:Metadata>
    <cis11:DomainSet>
    <cis11:GeneralGrid axisLabels="Lat Long h date" srsName=
"http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSG/0/4979&amp;2=http://www.opengis.net
/def/crs/OGC/0/AnsiDate">
    <cis11:DisplacementAxisNest axisLabels="h" uomLabels="m">
    <owc:offering>
    <owc:operation code="GetCoverage" href=
"http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&amp;version=2.1.0
&amp;request=GetCoverage&amp;coverageId=_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06_height&amp;format=image/tiff" method="GET" type="image/tiff
"/>
    </owc:offering>
    </cis11:DisplacementAxisNest>
    <cis11:IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg

```



```

d">
    <owc:offering>
        <owc:operation code="GetCoverage" href=
"http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0
&request=GetCoverage&coverageId=_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06_latitude&format=text/csv" method="GET" type="text/csv"/>
    </owc:offering>
    <owc:offering>
        <owc:operation code="GetCoverage" href=
"http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0
&request=GetCoverage&coverageId=_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06_longitude&format=text/csv" method="GET" type="text/csv
"/>
    </owc:offering>
    <owc:offering>
        <owc:operation code="GetCoverage" href=
"http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0
&request=GetCoverage&coverageId=_2017339123204_61738_CS_2B-
GEOPROF_GRANULE_P_R04_E06_profile_time&format=text/csv" method="GET" type=
"text/csv"/>
    </owc:offering>
    </cis11:IrregularAxisNest>
    <cis11:GridLimits axisLabels="i j" srsName=
"http://www.opengis.net/def/crs/OGC/0/Index2D">
        <cis11:IndexAxis axisLabel="i" lowerBound="0" upperBound="124"/>
        <cis11:IndexAxis axisLabel="j" lowerBound="0" upperBound="20752"/>
    </cis11:GridLimits>
    </cis11:GeneralGrid>
</cis11:DomainSet>
<cis11:RangeType>
    <swe:DataRecord>
        <swe:field name="Radar_Reflectivity">
            <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/short">
                <swe:description>Radar Reflectivity Factor</swe:description>
                <swe:nilValues>
                    <swe:nilValues>
                        <swe:nilValue reason="fill value">15360</swe:nilValue>
                        <swe:nilValue reason="missing">-8888</swe:nilValue>
                    </swe:nilValues>
                </swe:nilValues>
                <swe:uom code="dBZe"/>
                <swe:constraint>
                    <swe:AllowedValues>
                        <swe:interval>-32768 32767</swe:interval>
                        <swe:significantFigures>5</swe:significantFigures>
                    </swe:AllowedValues>
                </swe:constraint>
            </swe:Quantity>
        </swe:field>
        <swe:field name="Gaseous_Attenuation">

```

```

    <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/short">
      <swe:description>Gaseous Attenuation</swe:description>
      <swe:nilValues>
        <swe:nilValues>
          <swe:nilValue reason="fill value">15360</swe:nilValue>
          <swe:nilValue reason="missing">-9999</swe:nilValue>
        </swe:nilValues>
      </swe:nilValues>
      <swe:uom code="dBZe"/>
      <swe:constraint>
        <swe:AllowedValues>
          <swe:interval>-32768 32767</swe:interval>
          <swe:significantFigures>5</swe:significantFigures>
        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
  <swe:field name="CPR_Cloud_mask">
    <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/byte">
      <swe:description>CPR Cloud mask</swe:description>
      <swe:nilValues>
        <swe:nilValues>
          <swe:nilValue reason="fill value">-99</swe:nilValue>
          <swe:nilValue reason="missing">-9</swe:nilValue>
        </swe:nilValues>
      </swe:nilValues>
      <swe:uom code="various"/>
      <swe:constraint>
        <swe:AllowedValues>
          <swe:interval>-128 127</swe:interval>
          <swe:significantFigures>3</swe:significantFigures>
        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
</swe:DataRecord>
</cis11:RangeType>
<wcs20:ServiceParameters>
  <wcs20:CoverageSubtype>ReferenceableDataset</wcs20:CoverageSubtype>
  <wcs20:nativeFormat/>
</wcs20:ServiceParameters>
</wcs21:CoverageDescription>
</wcs21:CoverageDescriptions>

```

DescribeEOCoverageSet

Request

<http://ows.eox.at/testbed-14/eoxserver/ows?service=wcs&request=describeocoverageset&>

version=2.1.0&eoid=cloudsat-2b-geoprof&count=2

Response

- [responseDescribeEOCoverageSet.xml](https://github.com/opengeospatial/D007-Swath_Coverage_Engineering_Report/wiki/Use-case-CloudSat_examples/responseDescribeEOCoverageSet.xml) [https://github.com/opengeospatial/D007-Swath_Coverage_Engineering_Report/wiki/Use-case-CloudSat_examples/responseDescribeEOCoverageSet.xml]

DescribeEOCoverageSet response

```
<?xml version='1.0' encoding='iso-8859-1'?>
<wcseo:EOCoverageSetDescription numberMatched="110" numberReturned="2" xmlns:cis10=
"http://www.opengis.net/gmlcov/1.0" xmlns:cis11="http://www.opengis.net/cis/1.1/gml"
xmlns:crs="http://www.opengis.net/wcs/crs/1.0" xmlns:eop=
"http://www.opengis.net/eop/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:int
="http://www.opengis.net/wcs/interpolation/1.0" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:om="http://www.opengis.net/om/2.0" xmlns:owc="http://www.opengis.net/owc/1.0"
xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:rsub=
"http://www.opengis.net/wcs/range-subsetting/1.0" xmlns:scal=
"http://www.opengis.net/wcs/scaling/1.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:wcs20="http://www.opengis.net/wcs/2.0" xmlns:wcs21=
"http://www.opengis.net/wcs/2.1/gml" xmlns:wcseo="
http://www.opengis.net/wcs/wcseo/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/wcs/2.1/gml http://schemas.opengis.net/wcs/2.1/gml/wcsAll.xsd
http://www.opengis.net/wcs/wcseo/1.1
http://schemas.opengis.net/wcs/wcseo/1.1/wcsEOAll.xsd">
  <wcs21:CoverageDescriptions>
    <wcs21:CoverageDescription gml:id="_2017330010611_61600_CS_2B-
    GEOPROF_GRANULE_P_R04_E06">
      <cis11:Envelope axisLabels="Lat Long h date" srsDimension="4" srsName=
      "http://www.opengis.net/def/crs-
      compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979&amp;2=http://www.opengis.net
      /def/crs/OGC/0/AnsiDate">
        <cis11:AxisExtent axisLabel="Lat" lowerBound="-81.809029005" uomLabel="deg"
        upperBound="81.747520005"/>
        <cis11:AxisExtent axisLabel="Long" lowerBound="-179.997971005" uomLabel="deg"
        upperBound="179.999924005"/>
        <cis11:AxisExtent axisLabel="h" lowerBound="-4917" uomLabel="m" upperBound=
        "25062"/>
        <cis11:AxisExtent axisLabel="date" lowerBound="2017-11-26 01:26:03+00:00"
        uomLabel="d" upperBound="2017-11-26 02:21:23+00:00"/>
      </cis11:Envelope>
      <cis11:Metadata>
        <wcseo:EOMetadata>
          <eop:EarthObservation gml:id="eop__2017330010611_61600_CS_2B-
          GEOPROF_GRANULE_P_R04_E06">
            <om:phenomenonTime>
              <gml:TimePeriod gml:id="phen_time__2017330010611_61600_CS_2B-
              GEOPROF_GRANULE_P_R04_E06">
                <gml:beginPosition>2017-11-26T01:26:03Z</gml:beginPosition>
                <gml:endPosition>2017-11-26T02:21:23Z</gml:endPosition>
              </gml:TimePeriod>
            </om:phenomenonTime>
          </eop:EarthObservation>
        </wcseo:EOMetadata>
      </cis11:Metadata>
    </wcs21:CoverageDescription>
  </wcs21:CoverageDescriptions>
</wcseo:EOCoverageSetDescription>
```

```

</om:phenomenonTime>
<om:resultTime>
  <gml:TimeInstant gml:id="res_time__2017330010611_61600_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
    <gml:timePosition>2017-11-26T02:21:23Z</gml:timePosition>
  </gml:TimeInstant>
</om:resultTime>
<om:procedure/>
<om:observedProperty/>
<om:featureOfInterest>
  <eop:Footprint gml:id="footprint__2017330010611_61600_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
    <eop:multiExtentOf>
      <gml:MultiGeometry gml:id="multi_geom__2017330010611_61600_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
        <gml:geometryMember>
          <gml:LineString gml:id="line_string__2017330010611_61600_CS_2B-
GEOPROF_GRANULE_P_R04_E06">
            <gml:posList>-70.57107500 -19.02876500 -72.95608500
-23.12391300 -74.96965800 -27.68832600 -76.68452500 -32.86287300 -78.14092300
-38.79295700 -79.36860700 -45.68719900 -80.32445500 -53.28407300 -81.05862400
-61.91464200 -81.55524400 -71.47969100 -81.80902900 -82.72042100 -81.74195900
-94.42089800 -81.35762000 -105.44789900 -80.67722300 -115.45625300 -79.72912600
-124.18916300 -78.50637100 -131.81683300 -76.99546800 -138.44665500 -75.19721200
-144.13668800 -73.72017700 -147.72169500 -72.03339400 -151.02590900 -70.14587400
-154.02436800 -68.04177100 -156.75308200 -65.67740600 -159.27084400 -63.00725200
-161.61686700 -60.04938500 -163.77551300 -56.72987400 -165.80380200 -53.06691700
-167.69465600 -48.99454100 -169.48965500 -44.41706800 -171.23025500 -39.31382000
-172.92388900 -27.32359500 -176.25831600 -11.21291900 -179.99797100 -11.20326200
179.99992400 13.26330500 174.74099700 25.79866400 171.83317600 36.86013800
168.86434900 45.82065600 165.92021200 49.93270500 164.29847700 53.61832000
162.62858600 56.95401000 160.87780800 59.97777200 159.02427700 62.72701600
157.04064900 65.18359400 154.94178800 67.39399000 152.69561800 69.40323600
150.25389100 71.32357800 147.43067900 73.04538700 144.33277900 74.59246100
140.90319800 75.95484200 137.15957600 77.14787300 133.07644700 78.19279500
128.59262100 79.10068500 123.66085100 79.89111300 118.13572700 81.11864500
104.80731200 81.74752000 89.32199900 81.73171200 72.74892400 81.05502300
57.08217600</gml:posList>
          </gml:LineString>
        </gml:geometryMember>
      </gml:MultiGeometry>
    </eop:multiExtentOf>
  </eop:Footprint>
</om:featureOfInterest>
<om:result/>
<eop:metaDataProperty>
  <eop:EarthObservationMetaData>
    <eop:identifier>_2017330010611_61600_CS_2B-
GEOPROF_GRANULE_P_R04_E06</eop:identifier>
    <eop:acquisitionType>NOMINAL</eop:acquisitionType>
    <eop:status>ARCHIVED</eop:status>
  </eop:EarthObservationMetaData>
</eop:metaDataProperty>

```

```

    </eop:EarthObservationMetaData>
  </eop:metaDataProperty>
</eop:EarthObservation>
</wcseo:EOMetadata>
</cis11:Metadata>
<cis11:DomainSet>
  <cis11:GeneralGrid axisLabels="Lat Long h date" srsName=
"http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSG/0/4979&2=http://www.opengis.net
/def/crs/OGC/0/AnsiDate">
  <cis11:DisplacementAxisNest axisLabels="h" uomLabels="m">
    <owc:offering>
      <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=
_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_height&format=image/tiff"
method="GET" type="image/tiff"/>
    </owc:offering>
  </cis11:DisplacementAxisNest>
  <cis11:IrregularAxisNest axisLabels="Lat Long date" uomLabels="deg deg d">
    <owc:offering>
      <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=
_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_latitude&format=text/csv"
method="GET" type="text/csv"/>
    </owc:offering>
    <owc:offering>
      <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=
_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_longitude&format=text/csv"
method="GET" type="text/csv"/>
    </owc:offering>
    <owc:offering>
      <owc:operation code="GetCoverage" href="http://ows.eox.at/testbed-
14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=
_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_profile_time&format=text/csv"
method="GET" type="text/csv"/>
    </owc:offering>
  </cis11:IrregularAxisNest>
  <cis11:GridLimits axisLabels="i j" srsName=
"http://www.opengis.net/def/crs/OGC/0/Index2D">
    <cis11:IndexAxis axisLabel="i" lowerBound="0" upperBound="124"/>
    <cis11:IndexAxis axisLabel="j" lowerBound="0" upperBound="20752"/>
  </cis11:GridLimits>
</cis11:GeneralGrid>
</cis11:DomainSet>
<cis11:RangeType>
  <swe:DataRecord>
    <swe:field name="Radar_Reflectivity">
      <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/short">
        <swe:description>Radar Reflectivity Factor</swe:description>

```

```

<swe:nilValues>
  <swe:NilValues>
    <swe:nilValue reason="fill value">15360</swe:nilValue>
    <swe:nilValue reason="missing">-8888</swe:nilValue>
  </swe:NilValues>
</swe:nilValues>
<swe:uom code="dBZe"/>
<swe:constraint>
  <swe:AllowedValues>
    <swe:interval>-32768 32767</swe:interval>
    <swe:significantFigures>5</swe:significantFigures>
  </swe:AllowedValues>
</swe:constraint>
</swe:Quantity>
</swe:field>
<swe:field name="Gaseous_Attenuation">
  <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/short">
    <swe:description>Gaseous Attenuation</swe:description>
    <swe:nilValues>
      <swe:NilValues>
        <swe:nilValue reason="fill value">15360</swe:nilValue>
        <swe:nilValue reason="missing">-9999</swe:nilValue>
      </swe:NilValues>
    </swe:nilValues>
    <swe:uom code="dBZe"/>
    <swe:constraint>
      <swe:AllowedValues>
        <swe:interval>-32768 32767</swe:interval>
        <swe:significantFigures>5</swe:significantFigures>
      </swe:AllowedValues>
    </swe:constraint>
  </swe:Quantity>
</swe:field>
<swe:field name="CPR_Cloud_mask">
  <swe:Quantity definition=
"http://www.opengis.net/def/property/netcdf/1.0/byte">
    <swe:description>CPR Cloud mask</swe:description>
    <swe:nilValues>
      <swe:NilValues>
        <swe:nilValue reason="fill value">-99</swe:nilValue>
        <swe:nilValue reason="missing">-9</swe:nilValue>
      </swe:NilValues>
    </swe:nilValues>
    <swe:uom code="various"/>
    <swe:constraint>
      <swe:AllowedValues>
        <swe:interval>-128 127</swe:interval>
        <swe:significantFigures>3</swe:significantFigures>
      </swe:AllowedValues>
    </swe:constraint>
  </swe:Quantity>
</swe:field>

```

```

        </swe:Quantity>
    </swe:field>
</swe:DataRecord>
</cis11:RangeType>
<wcs20:ServiceParameters>
    <wcs20:CoverageSubtype>ReferenceableDataset</wcs20:CoverageSubtype>
    <wcs20:nativeFormat/>
</wcs20:ServiceParameters>
</wcs21:CoverageDescription>
</wcs21:CoverageDescriptions>
<wcseo:DatasetSeriesDescriptions>
    <wcseo:DatasetSeriesDescription gml:id="cloudsat-2b-geoprof">
        <cis11:Envelope axisLabels="Lat Long h date" srsDimension="4" srsName=
"http://www.opengis.net/def/crs-
compound?1=http://www.opengis.net/def/crs/EPSSG/0/4979&amp;2=http://www.opengis.net
/def/crs/OGC/0/AnsiDate">
            <cis11:AxisExtent axisLabel="Lat" lowerBound="-81.826141005" uomLabel="deg"
upperBound="81.825691005"/>
            <cis11:AxisExtent axisLabel="Long" lowerBound="-179.999786005" uomLabel="deg"
upperBound="180.000000005"/>
            <cis11:AxisExtent axisLabel="h" lowerBound="-4917" uomLabel="m" upperBound=
"25062"/>
            <cis11:AxisExtent axisLabel="date" lowerBound="2017-11-26 01:26:03+00:00"
uomLabel="d" upperBound="2017-12-05 13:46:54+00:00"/>
        </cis11:Envelope>
        <wcseo:DatasetSeriesId>cloudsat-2b-geoprof</wcseo:DatasetSeriesId>
        <gml:TimePeriod gml:id="cloudsat-2b-geoprof_timeperiod">
            <gml:beginPosition>2017-11-26T01:26:03Z</gml:beginPosition>
            <gml:endPosition>2017-12-05T13:46:54Z</gml:endPosition>
        </gml:TimePeriod>
    </wcseo:DatasetSeriesDescription>
</wcseo:DatasetSeriesDescriptions>
</wcseo:EOCoverageSetDescription>

```

GetCoverage for DomainSet coverages

- Height
 - http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_height&format=image/tiff
- Latitude
 - http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_latitude&format=text/csv
- Longitude
 - http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_longitude&format=text/csv
- Profile time

- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_profile_time&format=text/csv

- Height subset

- [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_height&subset=x\(0,10\)&subset=y\(0,100\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageId=_2017330010611_61600_CS_2B-GEOPROF_GRANULE_P_R04_E06_height&subset=x(0,10)&subset=y(0,100)&format=image/tiff)

GetCoverage

A list of various **GetCoverage** requests needed for the use case is provided below. The resulting files are provided in the data formats section afterwards.

- Full

- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff

- Band subsetting

- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&rangesubset=CPR_Cloud_mask
- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&rangesubset=Gaseous_Attenuation
- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&rangesubset=Radar_Reflectivity
- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&rangesubset=Radar_Reflectivity:CPR_Cloud_mask

- Subsetting

- Subset along y axis

- [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=y\(0,100\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=y(0,100)&format=image/tiff)

- Subset along x and y axis

- [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=x\(0,10\)&subset=y\(0,100\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=x(0,10)&subset=y(0,100)&format=image/tiff)

- Slice on the y axis

- [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=y\(100\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&subset=y(100)&format=image/tiff)

- Scaling

- Scale all axes equally 1

- http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&scalefactor=0.1&format=image/tiff

- Scale all axes equally 2
 - [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&scalexes=x\(0.1\),y\(0.1\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&scalexes=x(0.1),y(0.1)&format=image/tiff)
- Scale axes individually
 - [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&scalexes=x\(1\),y\(0.1\)&format=image/tiff](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&scalexes=x(1),y(0.1)&format=image/tiff)
- Scale axes to specific size
 - [http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&scalesize=x\(100\),y\(2000\)](http://ows.eox.at/testbed-14/eoxserver/ows?service=WCS&version=2.1.0&request=GetCoverage&coverageid=_2017339123204_61738_CS_2B-GEOPROF_GRANULE_P_R04_E06&format=image/tiff&scalesize=x(100),y(2000))

A.2. CloudSat Profile WCS Implementation 2

A.2.1. Endpoints

Table 12. Endpoints for CloudSat Swath Coverage WCS Implementation 2

Endpoint	http://cici.lab.asu.edu/swathserver
Implementation Organization:	Arizona State University
Developer or Point of Contact:	<ul style="list-style-type: none"> • Sizhe Wang • Wenwen Li
Email:	

A.2.2. Example WCS requests and responses

This Data is published by WCS 2.0.1 whose GetCapabilities request is available as <http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=GetCapabilities>. The XML snippet below shows one of the variable information of the published data:

```

<wcs:CoverageSummary>
  <wcs:CoverageId>cs_2b_geoprof:CPR_Cloud_mask</wcs:CoverageId>
  <wcs:CoverageSubtype>RectifiedGridCoverage</wcs:CoverageSubtype>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-106.91119384765625 -81.81893920898438</ows:LowerCorner>
    <ows:UpperCorner>167.55332946777344 81.81908416748047</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <ows:BoundingBox crs="http://www.opengis.net/def/crs/OGC/0/Index2D">
    <ows:LowerCorner>0 0</ows:LowerCorner>
    <ows:UpperCorner>20677 124</ows:UpperCorner>
  </ows:BoundingBox>
  <gml:TimePeriod gml:id="cs_2b_geoprof:CPR_Cloud_mask__timeperiod">
    <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
  </gml:TimePeriod>
</wcs:CoverageSummary>

```

The coverageId (<wcs:CoverageId>), WGS84 bounding box (<ows:WGS84BoundingBox>), data grid bounding box (<ows:BoundingBox>), and time coverage period (<gml:TimePeriod>) are present in GetCapabilities document. The CoverageId is constructed by the dataset name and the variable name, connected with a colon. The WGS84 bounding box is only to help people identify the coverage of the satellite scanning trail on the Earth's surface. The data grid itself, a vertical profile, could not correctly project onto the Earth. Therefore the data grid coverage/size is directly recorded in <ows:BoundingBox> tag. <gml:TimePeriod> contains the time information of first scanned and last scanned column of the dataset.

Using the coverageId, DescribeCoverage document of *cs_2b_geoprof:CPR_Cloud_mask* can be retrieved by http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=DescribeCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask, in which more data grid information is contained. Besides, the data range, data unit, and missing value are declared in the document too, as presented in the following XML snippet:

```

<gml:rangeType>
  <swe:DataRecord>
    <swe:field name="CPR_Cloud_mask">
      <swe:Quantity>
        <swe:description>CPR_Cloud_mask</swe:description>
        <swe:nilValues>
          <swe:nilValue reason="
http://www.opengis.net/def/nil/OGC/0/unknown">-99</swe:nilValue>
        </swe:nilValues>
        <swe:uom code="" />
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>-128 127</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</gml:rangeType>

```

The missing value is contained in `<swe:nilValue>`, the data unit is contained in the code attribute of tag `<swe:uom>` (missing for this variable), while data range can be found in `<swe:interval>`.

Finally, the data can be retrieved by GetCoverage request as http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask. It is available as GML 3.2, following CIS 1.1 standard. The values can be found in tag `<gml:DataBlock>`.

One more thing that needs to be mentioned is that not all the auxiliary/geographic information, like height, time, latitude, and longitude, has been provided along with a swath variable in a single GetCoverage request, instead, they are available as different variables that are listed in GetCapabilities document. For example, the XML snippet below shows the Height information:

```

<wcs:CoverageSummary>
  <wcs:CoverageId>cs_2b_geoprof:Height</wcs:CoverageId>
  <wcs:CoverageSubtype>RectifiedGridCoverage</wcs:CoverageSubtype>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-106.91119384765625 -81.81893920898438</ows:LowerCorner>
    <ows:UpperCorner>167.55332946777344 81.81908416748047</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <ows:BoundingBox crs="http://www.opengis.net/def/crs/OGC/0/Index2D">
    <ows:LowerCorner>0 0</ows:LowerCorner>
    <ows:UpperCorner>20677 124</ows:UpperCorner>
  </ows:BoundingBox>
  <gml:TimePeriod gml:id="cs_2b_geoprof:Height__timeperiod">
    <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
    <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
  </gml:TimePeriod>
</wcs:CoverageSummary>

```

Data Subsetting

Data subsetting is implemented based on the original data axes. Users can find the names of axes and corresponding ranges by sending the DescribeCoverage request. For example, below is a presentation of an XML snippet from http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=DescribeCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask:

```
<gml:EnvelopeWithTimePeriod axisLabels="nray nbin" srsDimension="2"
  srsName="http://www.opengis.net/def/crs/OGC/0/Index2D">
  <gml:lowerCorner>0 0</gml:lowerCorner>
  <gml:upperCorner>20677 124</gml:upperCorner>
  <gml:beginPosition>2017-01-01T13:04:14.558</gml:beginPosition>
  <gml:endPosition>2017-01-01T13:59:22.878068359</gml:endPosition>
</gml:EnvelopeWithTimePeriod>
```

The axes names are contained in the attribute `axisLabels` of tag `gml:EnvelopeWithTimePeriod`, and corresponding lower bound and upper bound can be found in tag `<gml:lowerCorner>` and `<gml:upperCorner>`. The `<gml:beginPosition>` and `<gml:endPosition>` elements define the time span of the scans in this dataset.

According to [OGC® Web Coverage Service 2.0 Interface Standard - KVP Protocol Binding Extension](<https://portal.opengeospatial.org/files/09-147r1>), the subsetting request is made by adding KVPs in GetCoverage request. For each axis that need to be subset, add a KVP like `subset=axisLabel(lowerBound, upperBound)`. For example, to get the subset from 50 to 100 in `nray` and from 10 to 20 in `nbin`, the request will be as follows: [http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask&subset=nray\(50,100\)&subset=nbin\(10,20\)](http://cici.lab.asu.edu/swathserver/?service=wcs&version=2.0.1&request=getCoverage&coverageId=cs_2b_geoprof:CPR_Cloud_mask&subset=nray(50,100)&subset=nbin(10,20))

The XML snippet below shows the new spatial and temporal bounds of retrieved dataset:

```
<gml:EnvelopeWithTimePeriod axisLabels="nray nbin" srsDimension="2"
  srsName="http://www.opengis.net/def/crs/OGC/0/Index2D">
  <gml:lowerCorner>50 10</gml:lowerCorner>
  <gml:upperCorner>100 20</gml:upperCorner>
  <gml:beginPosition>2017-01-01T13:04:22.558</gml:beginPosition>
  <gml:endPosition>2017-01-01T13:04:30.558</gml:endPosition>
</gml:EnvelopeWithTimePeriod>
```

A.3. VIIRS VNP09 Swath Coverage Service WCS

A.3.1. Endpoints

Table 13. Endpoints for VIIRS Swath Coverage WCS

Endpoint	http://geo.csiss.gmu.edu/gmuwcs
Implementation Organization:	George Mason University

Developer or Point of Contact:	Eugene G. Yu
Email:	gyu@gmu.edu [mailto:gyu@gmu.edu]

A.3.2. Example WCS requests and responses

The following are example requests:

- GetCapabilities
 - <http://geo.csiss.gmu.edu/gmuwcs?service=WCS&request=GetCapabilities>
- Describe coverage
 - <http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=DescribeCoverage&coverageId=VNP09.IBands.A2018115.1612.001>
- GetCoverage (supported formats: application/x-hdf, application/netcdf, application/gml+xml, image/tiff, application/geopackage+sqlite3)
 - <http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml>
- GetCoverage with subsetting/slicing
 - Subsetting
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Longitude,http://www.opengis.net/def/crs/EPSG/0/4326\(-40,-30\)&subset=Latitude,http://www.opengis.net/def/crs/EPSG/0/4326\(11,20\)&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:12:00.000,2018-04-25%2016:18:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Longitude,http://www.opengis.net/def/crs/EPSG/0/4326(-40,-30)&subset=Latitude,http://www.opengis.net/def/crs/EPSG/0/4326(11,20)&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:12:00.000,2018-04-25%2016:18:00.000))
 - Slicing
 - Time at 2018-04-25 16:12:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:12:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:12:00.000))
 - Time at 2018-04-25 16:13:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:13:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:13:00.000))
 - Time at 2018-04-25 16:14:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:14:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:14:00.000))

- Time at 2018-04-25 16:15:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:15:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:15:00.000))
- Time at 2018-04-25 16:16:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:16:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:16:00.000))
- Time at 2018-04-25 16:17:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:17:00.000\)](http://geo.csiss.gmu.edu/gmuwcs?service=WCS&version=2.0.1&request=GetCoverage&coverageId=VNP09.IBands.A2018115.1612.001&format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:17:00.000))

A.4. VIIRS VNP09 Swath Coverage Service WCS REST

A.4.1. Endpoints

Table 14. Endpoints for VIIRS Swath Coverage WCS REST

Endpoint	http://geo.csiss.gmu.edu/gmuwcsrest
Implementation Organization:	George Mason University
Developer or Point of Contact:	Eugene G. Yu
Email:	gyu@gmu.edu [mailto:gyu@gmu.edu]

A.4.2. Example WCS REST requests and responses

The following are example requests:

- `/coverages/{coverageId}/rangeSet` with subsetting/slicing
 - Subsetting
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Longitude,http://www.opengis.net/def/crs/EPSSG/0/4326\(-40,-30\)&subset=Latitude,http://www.opengis.net/def/crs/EPSSG/0/4326\(11,20\)&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:12:00.000,2018-04-25%2016:18:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Longitude,http://www.opengis.net/def/crs/EPSSG/0/4326(-40,-30)&subset=Latitude,http://www.opengis.net/def/crs/EPSSG/0/4326(11,20)&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:12:00.000,2018-04-25%2016:18:00.000))
 - Note: This implementation supports both the explicit "format" parameter, or the header "Accept" media type.
 - Slicing
 - Time at 2018-04-25 16:12:00.000:

- [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:12:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:12:00.000))
- Time at 2018-04-25 16:13:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:13:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:13:00.000))
- Time at 2018-04-25 16:14:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:14:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:14:00.000))
- Time at 2018-04-25 16:15:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:15:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:15:00.000))
- Time at 2018-04-25 16:16:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:16:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:16:00.000))
- Time at 2018-04-25 16:17:00.000:
 - [http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC\(2018-04-25%2016:17:00.000\)](http://geo.csiss.gmu.edu/gmuwcsrest/coverages/VNP09.IBands.A2018115.1612.001/rangeSet?format=application/gml+xml&subset=Time,http://www.opengis.net/def/trs/ISO-8601/0/Gregorian+UTC(2018-04-25%2016:17:00.000))

A.5. Swath Coverage Client

A.5.1. Client Implementation from MEEO

Table 15. Swath Coverage Client 1 from MEEO

Website	https://cloud.services.meeo.it
Implementation Organization:	MEEO
Developer or Point of Contact:	Sergio Ferraresi
Email:	

A.5.2. Examples

Example screen captures can be seen from the following links:

- Discovery
 - Catalogue (<https://cloud.services.meeo.it/f/0986b645874c41abbf75/?dl=1>)

- Granule
 - Granules Polygons (<https://cloud.services.meeo.it/f/06516a3e290141fbb0bb/?dl=1>)
 - Granule Polygon (<https://cloud.services.meeo.it/f/98aad25e047b4a22a0fd/?dl=1>)
 - Granule Visualization(<https://cloud.services.meeo.it/f/719a9f375b524095b788/?dl=1>)
 - Full Strip(<https://cloud.services.meeo.it/f/af1c80846fb0462790e8/?dl=1>)

A.5.3. Client Implementation from EOX

Table 16. Swath Coverage Client 2 from EOX

Website	http://ows.eox.at/testbed-14/eoxc/
Implementation Organization:	EOX
Developer or Point of Contact:	Stephan Meißl
Email:	stephan.meissl@eox.at [mailto:stephan.meissl@eox.at]

The functions of the client can be grouped into five groups: general view, timeslider view, global view, analytic view, and product setting.

General view

The main functions to adjust the visualization are the subsetting of the data to be shown. Subsetting can be performed in time via the timeslider (selected time interval in bottom in screenshots above), spatially via globe or map view (left part), and in actual parameter values via analytics view (selection on histograms in lower part of analytics view on the right).

Timeslider view

The timeslider is a control that allows a user to interactively zoom and pan in time as well as to jump to particular dates via the picker (top right corner of timeslider). While navigating the timeslider, it gives an indication of data availability (blue rectangles). Hovering over these indicators shows the product id.

Once the time of interest is visible it allows to select the time interval for which data shall be visualized simply by click and hold on the timeslider. The current selection is highlighted with a gray background and can be adjusted at the ends as needed. Alternatively, the user can click on the availability indication for a quick selection.

Any change of the time selection triggers a refresh of the data visualizations and prior data download as necessary.

Globe view

The globe view is a 2.5D map viewer. As on a standard map the data would only be visible as footprint curves, the map view uses a 2.5D view. Both views are highly interactive allowing to zoom, pan, tilt, and rotate the globe or map.

All data available in the selected time interval is shown. The rendering itself (like selected parameter, color-scale, or opacity) is configured using the product settings accessibly via the Layers button in the top menu.

The globe view is developed, based on Cesium and geotiff.js.

Analytics view

The analytics view also shows the currently selected parameter of the product like on the globe view. It is again highly interactive allowing to zoom and pan on both or individual axes.

The parameter histograms on the bottom allow subsetting in actual parameter values. The subsetting is applied to all views particularly globe and analytics view.

By default, the axes show time horizontally and height vertically. This can be changed to show for example the parameter values horizontally.

While it is recommended to perform the time and spatial subsets by the data interface, i.e., on the server, the parameter value subset can be performed directly in the client.

The analytics view is based on D3, graphly, and geotiff.js.

Product settings

The product settings allow a user to configure the rendering.

The first thing is to select the parameter to visualize. In the case of CloudSat 2B-GEOPROF, this is either radar reflectivity, gaseous attenuation, or CPR cloud mask.

Second the color-scale and value range to use can be adjusted. A slider allows to configure the opacity.

In the globe view the outline showing the footprint and direction of the curtain, as well as the legend can be enabled or disabled.

The analytics view can be configured separately if needed using the cog icon.

Appendix B: WCS REST API Full Specification

This appendix includes a full specification attempted during the Testbed-14. It is provided here as was developed. It provided the baseline for implementing the REST API of WCS to be used in serving Swath Coverages.

B.1. Scope

This specification provides recommendations for implementing a RESTful API as an alternative interface service to a Web Coverage Service. The RESTful API is designed following the guidelines and specifications of OpenAPI. The core resource is on coverage.

Resource discovery and simple subsetting/trimming operations allow for the retrieval of specific subset of coverage resources that serve the user requirements. The requirements are group into levels of conformances as the implementation of the service to claim what they conform.

B.2. Conformance

The following are the conformance classes defined in this specification.

- Core Conformance Class: This standard defines six requirements / conformance classes.
- Range Subsetting Extension Conformance Class.
- Earth Observation Extension Conformance Class.
- Transaction Insertion Deletion Conformance Class.
- Transaction Update Conformance Class.
- GeoTiff Coverage Conformance Class.
- NetCDF Coverage Conformance Class.
- GeoPackage Coverage Conformance Class.

B.3. References

The following normative documents are served as bases in developing this specification.

- Open API Initiative: OpenAPI Specification 3.0.1, <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md>
- IETF: RFC7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. <https://tools.ietf.org/html/rfc7231>
- W3C: HTML5, W3C Recommendation, <http://www.w3.org/TR/html5/>
- OGC: OGC 06-121r3, OGC Web Services Common Specification, 2010.
- OGC: OGC 09-110r3, OGC® WCS 2.0 Interface Standard – Core, 2012.

- OpenSearch: OpenSearch 1.1, - <https://github.com/dewitt/opensearch/blob/master/opensearch-1-1-draft-6.md>

B.4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r3], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

B.4.1. coverage

feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain

B.4.2. dataset

Earth Observation that forms the minimum accessible/downloadable data that may contain several coverages

B.4.3. dataset series

collection of Earth Observation datasets that have some common characteristics

B.4.4. subsetting

operation on coverage which, for a coverage provided, extracts part or all of its cell/value pairs and returns a sub-coverage containing these cell/value pairs

B.4.5. trimming

operation on coverage as subsetting operation which returns a coverage with the same number of dimensions as the input

B.4.6. slicing

operation on coverage as subsetting operation which returns a coverage with a reduced number of dimensions as compared to the input coverage

B.5. Conventions

B.5.1. Base Standard

The REST API is based on OpenAPI version 3.0.1 (URL: <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md>). This leads to the following fixed value of the OpenAPI document. OpenAPI documents are allowed to be defined in either JSON or YAML ([Version 1.2](#)

[<http://yaml.org/spec/1.2/spec.html>]).

JSON

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "A sample API conforming to the OGC Web Coverage Service standard",
    "version": "0.0.1"
  },
  "paths":{
  }
}
```

YAML

```
openapi: 3.0.1

info:
  title: A sample API conforming to the OGC Web Coverage Service standard
  version: 0.0.1

paths:
```

B.6. Core Requirement Class

B.6.1. Overview

The core requirement of the WCS REST API specification task is to implement the following paths and basic operations on resource coverages and coverage data (rangeSet).

Table 1 – Paths

PATH	METHOD	DESCRIPTION
/	GET	Gets the landing page in the requested representation

PATH	METHOD	DESCRIPTION
/api	GET	Gets the API description (i.e. OpenAPI document)
/conformance	GET	Gets conformance classes that server implements
/coverages	GET	Gets the description/metadata of all the coverages offered by the system
/coverages/{coverageId}	GET	Gets the description/metadata of the specific coverage
/coverages/{coverageId}/rangeSet	GET	Gets the coverage or a subset thereof

In the context of the WCS REST API design experiment, the resources are identified by paths. Each path uniquely identifies a resource. For example, the root path / identifies the entry point which returns a document containing links to other resources - i.e. specifications identified by "/api", conformance declaration (identified by "/conformance"), and list of coverages (identified by "/coverages"). The coverage description of each coverage is identified as "/coverages/{coverageId}" while the actual data of a coverage is identified as "/coverages/{coverageId}/rangeSet". The paths are unique identifiers for different resources.

The resources can be traversed through following the hyperlinks in the content from the root endpoint (root path). For example, the response from the root path would contain links to hyperlinks to API specification document, conformance, and list of coverages. Once transfer to the resource for list of coverages, its content contains hyperlinks allowing users to traverse and find resources for coverage description and coverage data. When coverage descriptions are in media types other than HTML, users need to use the schema to follow through the resources.

B.6.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field openapi

Requirement	O1
Requirement Name	/req/core/openapi
Description	The value should be fixed as "3.0.1".

JSON

```
{
  "openapi": "3.0.1"
}
```

YAML

```
openapi: 3.0.1
```

Field info object

Requirement	O2
Requirement Name	/req/core/openapi-info
Description	The field version of info object should be fixed as "0.0.1" to comply with this WCS REST API specification.

JSON

```
{  
  "openapi": "3.0.1",  
  "info": {  
    "title": "A sample API conforming to the OGC Web Coverage Service standard",  
    "version": "0.0.1"  
  },  
  "paths": {  
  }  
}
```

YAML

```
openapi: 3.0.1  
  
info:  
  title: A sample API conforming to the OGC Web Coverage Service standard  
  version: 0.0.1  
  
paths:
```

Field servers object

Requirement	O3
Requirement Name	/req/core/openapi-servers
Description	At least one server should be defined.

JSON

```
"servers": [{  
  "title": "https://dev.example.org/",  
  "version": "0.0.1"  
}]
```

YAML

```
servers:  
- url: 'https://dev.example.org/'
```

Field paths object

Requirement	O4
Requirement Name	/req/core/openapi-paths
Description	The following paths should be defined: /, /api, /conformance

JSON

```
"paths":{  
  "/": {},  
  "/api": {},  
  "/conformance": {}  
}
```

YAML

```

paths:

/:

/api:

/conformance:

```

Field components object

Requirement	O5
Requirement Name	/req/core/openapi-components
Description	This defines reusable objects. They are only effective when they are referenced outside of the components object. Types of reusable components for WCS REST API include parameters and schemas.

JSON

```

"components": {
"parameters": {
},
"schemas": {
}
}

```

YAML

```

components:
parameters:
schemas:

```

Parameter bbox

Requirement	O5P.1
Requirement Name	/req/core/openapi-components-bbox

Requirement	O5P.1
Description	This is the spatial parameter used in query for subsetting.

JSON

```
"bbox":
{
  "name": "bbox",
  "in": "query",
  "description":
    "Only coverages that have a geometry that intersects the bounding box are selected. \
    The bounding box is provided as four or six numbers, depending on whether the \
    coordinate reference system includes a vertical axis (elevation or depth): \
    * Lower left corner, coordinate axis 1 \
    * Lower left corner, coordinate axis 2 \
    * Lower left corner, coordinate axis 3(optional) \
    * Upper right corner, coordinate axis 1 \
    * Upper right corner, coordinate axis 2 \
    * Upper right corner, coordinate axis 3 (optional) \
    The coordinate reference system of the values is WGS84 longitude/latitude \
    (http://www.opengis.net/def/crs/OGC/1.3/CRS84) unless a different coordinate \
    reference system is specified in the parameter `bbox-crs`. \
    For WGS84 longitude/latitude the values are in most cases the sequence of \
    minimum longitude, minimum latitude, maximum longitude and maximum latitude. \
    However, in cases where the box spans the antimeridian the first value \
    (west-most box edge) is larger than the third value (east-most box edge). \
    If a coverage has multiple spatial geometry properties, it is the decision of the \
```

```
server whether only a single spatial geometry property is used to determine \
the extent or all relevant geometries.",
"required": false,
"schema":
{
"type": "array",
"minItems": 4,
"maxItems": 6
},
"items":
{
"type": "number"
},
"style": "form",
"explode": false
}
```

YAML

```
bbox:
name: bbox
in: query
description: >
Only coverages that have a geometry that intersects the bounding box are selected.
The bounding box is provided as four or six numbers, depending on whether the
coordinate reference system includes a vertical axis (elevation or depth):
* Lower left corner, coordinate axis 1
```

- * Lower left corner, coordinate axis 2
- * Lower left corner, coordinate axis 3(optional)
- * Upper right corner, coordinate axis 1
- * Upper right corner, coordinate axis 2
- * Upper right corner, coordinate axis 3 (optional)

The coordinate reference system of the values is WGS84 longitude/latitude (<http://www.opengis.net/def/crs/OGC/1.3/CRS84>) unless a different coordinate reference system is specified in the parameter `bbox-crs`.

For WGS84 longitude/latitude the values are in most cases the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude.

However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge).

If a coverage has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries.

`required: false`

`schema:`

`type: array`

`minItems: 4`

`maxItems: 6`

`items:`

`type: number`

`style: form`

`explode: false`

Parameter time

Requirement	O5P.2
Requirement Name	/req/core/openapi-components-time
Description	The definition of temporal parameter used in query should be given in common components.

JSON

```
"time":
{
  "name": "time",
  "in": "query",
  "description":
    "Either a date-time or a period string that adheres to RFC 3339. Examples: \
    * A date-time: \"2018-02-12T23:20:50Z\" \
    * A period: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" or \"2018-02-12T00:00:00Z/P1M6\
    DT12H31M12S\" \
    Only coverages that have a temporal property that intersects the value of \
    `time` are selected. \
    If a coverage has multiple temporal properties, it is the decision of the \
    server whether only a single temporal property is used to determine \
    the extent or all relevant temporal properties.",
  "required": false,
  "schema":
  {
    "type": "string"
  },
  "style": "form",
  "explode": false
}
```

YAML

time:

name: time

in: query

description: >

Either a date-time or a period string that adheres to RFC 3339. Examples:

* A date-time: "2018-02-12T23:20:50Z"

* A period: "2018-02-12T00:00:00Z/2018-03-18T12:31:12Z" or "2018-02-12T00:00:00Z/P1M6DT12H31M12S"

Only coverages that have a temporal property that intersects the value of `time` are selected.

If a coverage has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.

required: false

schema:

type: string

style: form

explode: false

Parameter coverageId

Requirement	O5P.3
Requirement Name	/req/core/openapi-components-coverageid
Description	This defines the identifier of coverage.

JSON

```

"coverageId":
{
  "name": "coverageId",
  "in": "path",
  "required": true,
  "description": "Identifier (name) of a specific coverage",
  "schema":
  {
    "type": "string"
  }
}

```

YAML

```

coverageId:
  name: coverageId
  in: path
  required: true
  description: Identifier (name) of a specific coverage
  schema:
    type: string

```

Schema exception

Requirement	O5S.1
Requirement Name	/req/core/openapi-components-exception
Description	This defines exception object that can be used in responses.

JSON

```
"exception": {
  "type": "object",
  "required": [
    "code"
  ],
  "properties": {
    "code": {
      "type": "string"
    },
    "description": {
      "type": "string"
    }
  }
}
```

YAML

```
exception:
  type: object
  required:
  - code
  properties:
    code:
      type: string
    description:
      type: string
```


Schema link

Requirement	O5S.2
Requirement Name	/req/core/openapi-components-link
Description	This defines link object. This is a different definition from the link object of OpenAPI.

JSON

```
"link": {
  "type": "object",
  "required": [
    "href"
  ],
  "properties": {
    "href": {
      "type": "string"
    },
    "rel": {
      "type": "string"
    },
    "type": {
      "type": "string"
    },
    "hreflang": {
      "type": "string"
    }
  },
  "xml": {
    "name": "link"
  }
}
```

YAML

```

link:
  type: object
  required:
    - href
  properties:
    href:
      type: string
    rel:
      type: string
    type:
      type: string
    hreflang:
      type: string
    xml:
      name: link

```

Schema root

Requirement	O5S.3
Requirement Name	/req/core/openapi-components-root
Description	This defines root object that can be used in the response to GET operation towards a WCS REST API root path.

JSON

```

"root": {
  "type": "object",
  "required": [
    "link"

```

```
],  
  
"properties": {  
  
  "link": {  
  
    "type": "array",  
  
    "items": {  
  
      "$ref": "#/components/schemas/link"  
  
    },  
  
    "example": [  
  
      {  
  
        "href": 'http://data.example.org/',  
  
        "rel": "self",  
  
        "type": "application/json",  
  
        "title": "this document"  
  
      },  
  
      {  
  
        "href": 'http://data.example.org/api',  
  
        "rel": "service",  
  
        "type": "application/json",  
  
        "title": "the list of API definitions"  
  
      },  
  
      {  
  
        "href": 'http://data.example.org/conformance',  
  
        "rel": "conformance",  
  
        "type": "application/json",  
  
        "title": "WCS 3.0 conformance classes implemented by this server"  
  
      },  
  
    ]  
  
  }  
  
}
```

```
{
  "href": 'http://data.example.org/coverages',
  "rel": "metadata",
  "type": "application/json",
  "title": "Metadata about the coverage set"
}
]
},
"xml": {
  "wrapped": true
}
}
```

YAML

```
root:
  type: object
  required:
  - link
  properties:
    link:
      type: array
      items:
        $ref: '#/components/schemas/link'
      example:
        - href: 'http://data.example.org/'
```

```

rel: self

type: application/json

title: this document

- href: 'http://data.example.org/api'

rel: service

type: application/json

title: the list of API definitions

- href: 'http://data.example.org/conformance'

rel: conformance

type: application/json

title: WCS 3.0 conformance classes implemented by this server

- href: 'http://data.example.org/coverages'

rel: metadata

type: application/json

title: Metadata about the coverage set

```

Schema openapi

Requirement	O5S.4
Requirement Name	/req/core/openapi-components-openapi
Description	This defines an OpenAPI object that can be used in the response to GET operation towards a WCS REST API api path.

JSON

```

"openapi": {
  "type": "object",
  "required": [
    "apispecs"

```

```
],  
  
"properties": {  
  "apispecs": {  
    "type": "array",  
    "items": {  
      "$ref": "#/components/schemas/link"  
    },  
    "example": {  
      {  
        "href": "http://data.example.org/api/openapi.yaml",  
        "rel": "service",  
        "type": "application/openapi+yaml;version=3.0",  
        "title": "The API YAML definition"  
      },  
      {  
        "href": "http://data.example.org/api/openapi.json",  
        "rel": "service",  
        "type": "application/openapi+json;version=3.0",  
        "title": "The API JSON definition"  
      }  
    }  
  }  
}
```

YAML

```

openapi:
  type: object
  required:
    - apispecs
  properties:
    apispecs:
      type: array
      items:
        $ref: '#/components/schemas/link'
      example:
        - href: 'http://data.example.org/api/openapi.yaml'
        rel: service
        type: application/openapi+yaml;version=3.0
        title: The API YAML definition
        - href: 'http://data.example.org/api/openapi.json'
        rel: service
        type: application/openapi+json;version=3.0
        title: The API JSON definition

```

Schema reqclasses

Requirement	O5S.5
Requirement Name	/req/core/openapi-components-reqclasses
Description	This defines the conformance classes that the service complies to.

JSON


```

"reqclasses": {
  "type": "object",
  "required": [
    "conformsTo"
  ],
  "properties": {
    "conformsTo": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
  example: [
    "http://www.opengis.net/spec/wcs-2/3.0/req/core",
    "http://www.opengis.net/spec/wcs-2/3.0/req/eo",
    "http://www.opengis.net/spec/wcs-2/3.0/req/update",
    "http://www.opengis.net/spec/wcs-2/3.0/req/insertdelete",
    "http://www.opengis.net/spec/wcs-2/3.0/req/geotiff",
    "http://www.opengis.net/spec/wcs-2/3.0/req/netcdf",
    "http://www.opengis.net/spec/wcs-2/3.0/req/geopackage"
  ]
}
}
}

```

YAML

```

reqclasses:
  type: object
  required:
    - conformsTo
  properties:
    conformsTo:
      type: array
      items:
        type: string
  example:
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/core'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/eo'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/update'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/insertdelete'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/geotiff'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/netcdf'
    - 'http://www.opengis.net/spec/wcs-2/3.0/req/geopackage'

```

Schema contents

Requirement	O5S.6
Requirement Name	/req/core/openapi-components-contents
Description	This defines the contents for coverages that the service return on listing coverage resources.

YAML

```

contents:
  type: object
  required:

```

```
- total
- startIndex
- items
- coverages

properties:
total:
type: integer
startIndex:
type: integer
items:
type: integer
coverages:
type: array
items:
$ref: '#/components/schemas/coverageSummary'

coverageSummary:
type: object
properties:
coverageId:
type: string
coverageSubType:
type: string
wgs84BoundingBox:
$ref: '#/components/schemas/boundingBox'

boundingBox:
```

```
$ref: '#/components/schemas/boundingBox'
```

```
metaData:
```

```
$ref: '#/components/schemas/OWSMetadata'
```

```
OWSMetadata:
```

```
type: object
```

```
properties:
```

```
identifier:
```

```
type: string
```

```
title:
```

```
type: string
```

```
abstract:
```

```
type: string
```

```
keywords:
```

```
type: string
```

```
boundingBox:
```

```
$ref: '#/components/schemas/boundingBox'
```

```
outputFormat:
```

```
type: string
```

```
metadata:
```

```
type: string
```

```
availableCRS:
```

```
type: string
```

```
accessConstraints:
```

```
type: string
```

```
fees:
```

```
type: string
```

```
pointOfContact:
  $ref: '#/components/schemas/CI_ResponsibleParty'
  language:
    type: string
  CI_ResponsibleParty:
    type: object
    properties:
      individualName:
        type: string
      organizationName:
        type: string
      positionName:
        type: string
      role:
        type: string
      boundingBox:
        type: object
        required:
          - lowerCorner
          - upperCorner
        properties:
          lowrCorner:
            type: array
            items:
              type: number
```

```

upperCorner:
  type: array
  items:
    type: number
  crs:
    type: string
    default: urn:ogc:def:crs:OGC:1.3:CRS84
  dimensions:
    type: integer
    default: 2

```

Schema coverageDescription

Requirement	O5S.7
Requirement Name	/req/core/openapi-components-coverageDescription
Description	This defines the description to describe a coverage when one coverage is identified and retrieved for metadata.

YAML

```

coverageDescription:
  type: object
  properties:
    coverageId:
      type: string
    coverageFunction:
      $ref: '#/components/schemas/coverageFunction'
  metadata:
    type: array

```

```
items:
  type: object
  domainSet:
    $ref: '#/components/schemas/domainSet'
  rangeType:
    $ref: '#/components/schemas/DataRecord'
  serviceParameters:
    $ref: '#/components/schemas/ServiceParameters'
  ServiceParameters:
    type: object
  properties:
    coverageSubtype:
      type: string
    extension:
      type: array
      items:
        type: object
    DataRecord:
      $ref: '#/components/schemas/DataRecordType'
    DataRecordType:
      allOf:
        - $ref: '#/components/schemas/AbstractDataComponentType'
        - type: object
      properties:
        field:
          type: array
```

```

items:

allof:

- $ref: '#/components/schemas/AbstractDataComponentType'

# - $ref: '#/components/schemas/AbstractDataComponentPropertyType' # There is
PropertyType

- type: object

properties:

name:

$ref: '#/components/schemas/NCName'

AbstractDataComponentPropertyType:

type: object

# No definition was found in SWE Common [OGC 08-094r1]

AbstractDataComponentType:

allof:

- $ref: '#/components/schemas/AbstractSWEIdentifiableType'

- type: object

properties:

definition:

type: string

updatable:

type: boolean

optional:

type: boolean

AbstractSWEIdentifiableType:

allof:

- $ref: '#/components/schemas/AbstractSWEType'

```


- type: object

properties:

identifier:

type: string

label:

type: string

description:

type: string

AbstractSWEType:

type: object

properties:

extension:

type: array

items:

type: object

id:

type: string

domainSet:

\$ref: '#/components/schemas/DomainSetType'

DomainSetType:

allOf:

- oneOf:

- \$ref: '#/components/schemas/AbstractGeometry'

- \$ref: '#/components/schemas/AbstractTimeObject'

- \$ref: '#/components/schemas/OwnershipAttributeGroup'

- \$ref: '#/components/schemas/AssociationAttributeGroup'

AbstractTimeObject:

\$ref: '#/components/schemas/AbstractTimeObjectType'

AbstractTimeObjectType:

allOf:

- \$ref: '#/components/schemas/AbstractGMLType'

AbstractGeometry:

\$ref: '#/components/schemas/AbstractGeometryType'

AbstractGeometryType:

allOf:

- \$ref: '#/components/schemas/AbstractGMLType'

- \$ref: '#/components/schemas/SRSReferenceGroup'

SRSReferenceGroup:

type: object

properties:

srsName:

type: string

srsDimension:

type: integer

minimum: 1

SRSInformationGroup:

\$ref: '#/components/schemas/SRSInformationGroup'

SRSInformationGroup:

type: object

properties:

axisLabels:

```

$ref: '#/components/schemas/NCNameList'

uomLabels:

$ref: '#/components/schemas/NCNameList'

NCNameList:

type: array

items:

$ref: '#/components/schemas/NCName'

NCName:

type: string

# pattern: '[\i-[:]][\c-[:]]*'

AbstractGMLType:

type: object

properties:

StandardObjectProperties:

$ref: '#/components/schemas/StandardObjectProperties'

id:

$ref: '#/components/schemas/ID'

StandardObjectProperties:

type: object

properties:

metaDataProperty:

$ref: '#/components/schemas/metaDataProperty'

description:

$ref: '#/components/schemas/StringOrRefType'

descriptionReference:

```

```

$ref: '#/components/schemas/ReferenceType'

identifier:

$ref: '#/components/schemas/CodeWithAuthorityType'

name:

$ref: '#/components/schemas/CodeType'

CodeType:

allOf:

- type: string

- type: object

properties:

codeSpace:

type: string

CodeWithAuthorityType:

allOf:

- $ref: '#/components/schemas/CodeType'

- type: object

required: ["codeSpace"]

StringOrRefType:

allOf:

- type: string

- $ref: '#/components/schemas/AssociationAttributeGroup'

metaDataProperty:

$ref: '#/components/schemas/MetaDataPropertyType'

MetaDataPropertyType:

type: object

properties:

```

AbstractMetaData:

\$ref: '#/components/schemas/AbstractMetaData'

AssociationAttributeGroup:

\$ref: '#/components/schemas/AssociationAttributeGroup'

AbstractMetaData:

\$ref: '#/components/schemas/AbstractMetaDataType'

AbstractMetaDataType:

type: object

properties:

id:

\$ref: '#/components/schemas/ID'

ID:

type: string

pattern: '([\i-[:]][\c-[:]]*)'

coverageFunction:

\$ref: '#/components/schemas/coverageFunctionType'

coverageFunctionType:

oneOf:

- \$ref: '#/components/schemas/mappingRule'
- \$ref: '#/components/schemas/coverageMappingRule'
- \$ref: '#/components/schemas/gridFunction'

mappingRule:

\$ref: '#/components/schemas/coverageMappingRule'

coverageMappingRule:

oneOf:

```

- $ref: '#/components/schemas/ruleDefinition'

- $ref: '#/components/schemas/ruleReference'

ruleDefinition:

type: string

ruleReference:

$ref: '#/components/schemas/ReferenceType'

ReferenceType:

type: object

properties:

OwnershipAttributeGroup:

$ref: '#/components/schemas/OwnershipAttributeGroup'

AssociationAttributeGroup:

$ref: '#/components/schemas/AssociationAttributeGroup'

OwnershipAttributeGroup:

type: object

properties:

owns:

type: boolean

default: false

AssociationAttributeGroup:

type: object

properties:

owns:

type: boolean

default: false

simpleLink:

```

```
$ref: '#/components/schemas/simpleLink'  
  
nilReason:  
  
$ref: '#/components/schemas/NilReasonType'  
  
remoteSchema:  
  
type: string  
  
simpleLink:  
  
type: object  
  
properties:  
  
type:  
  
type: string  
  
default: "simple"  
  
href:  
  
type: string  
  
role:  
  
type: string  
  
arcrole:  
  
type: string  
  
title:  
  
type: string  
  
show:  
  
type: string  
  
enum:  
  
- new  
  
- replace  
  
- embed
```

```
- other

- none

actuate:

type: string

enum:

- onLoad

- onRequest

- other

- none

NilReasonType:

allof:

- $ref: '#/components/schemas/NilReasonEnumeration'

- type: string

NilReasonEnumeration:

allof:

- type: string

enum:

- inapplicable

- missing

- template

- unknown

- withheld

- type: string

pattern: 'other:\w{2,}'

gridFunction:

$ref: '#/components/schemas/gridFunctionType'
```



```
integerList:
  type: array
  items:
    type: integer
gridFunctionType:
  type: object
properties:
  sequenceRule:
    $ref: '#/components/schemas/SequenceRuleType'
  startPoint:
    $ref: '#/components/schemas/integerList'
SequenceRuleType:
  allOf:
    - $ref: '#/components/schemas/SequenceRuleEnumeration'
    - type: object
  properties:
    order:
      $ref: '#/components/schemas/IncrementOrder'
    axisOrder:
      $ref: '#/components/schemas/AxisDirectionList'
  default: "Linear"
SequenceRuleEnumeration:
  type: string
  enum:
    - Linear
```

- Boustrophedonic
- Cantor-diagonal
- Spiral
- Morton
- Hilbert

IncrementOrder:

type: string

enum:

- "+x+y"
- "+y+x"
- "+x-y"
- "-x-y"

AxisDirectionList:

type: array

items:

\$ref: '#/components/schemas/AxisDirection'

AxisDirection:

type: string

pattern: '[\+\-][1-9][0-9]*'

Schema TrimDimensions, TrimLows, and TrimHighs

Requirement	O5S.8
Requirement Name	/req/core/openapi-components-trimming-request
Description	These define the request parameters that are used in query for coverage trimming

YAML

TrimDimensions:

name: TrimDimensions

in: query

description: >

This parameter is a string that holds a list of comma separated dimension names for subsetting. The dimension name should match axislabels that are available in the offered coverage. The order should match the order in {TrimLows} and {TrimHighs}. If the corresponding position in {TrimLows} or {TrimHighs} has no value - no space or white space, it the corresponding dimension is treated with no value entered.

For example, the coverage has the following dimensions

```
{latitude, longitude, time, band}
```

```
latitude = [-90,90]
```

```
longitude = [-180, 180]
```

```
time = [1983, 2018]
```

```
band = [1, 15]
```

```
Given TrimDimensions=latitude,time
```

```
TrimLows=-70,,1988
```

```
TrimHighs=80,170,1998
```

This is a subsetting operation that retrieve the data bounding by latitude = [-70, 80], longitude = [-180, 170], time=[1983, 1998], and band = [1, 15].

required: false

allowEmptyValue: true

schema:

type: string

TrimLows:

name: TrimLows

in: query

description: >

This parameter is a string that holds a list of comma separated values for subsetting.

The values should be only applicable to the dimensions that listed in {TrimDimensions}. The order should match the order in {TrimDimensions}. If the corresponding position has no value to be give, it should be left with no value - no space or white space. This would be equivalent to the case where the lowest value along that dimension is used instead.

If empty, it would be treated that all dimensions have no restriction on lows.

For example, the coverage has the following dimensions

```
{latitude, longitude, time, band}
```

```
latitude = [-90,90]
```

```
longitude = [-180, 180]
```

```
time = [1983, 2018]
```

```
band = [1, 15]
```

```
Given TrimDimensions=latitude,time
```

```
TrimLows=-70,,1988
```

```
TrimHighs=80,170,1998
```

This is a subsetting operation that retrieve the data bounding by latitude = [-70, 80], longitude = [-180, 170], time=[1983, 1998], and band = [1, 15].

```
required: false
```

```
allowEmptyValue: true
```

```
schema:
```

```
type: string
```

```
TrimHighs:
```

```
name: TrimLows
```

```
in: query
```

```
description: >
```

This parameter is a string that holds a list of comma separated values for subsetting. The values should be only applicable to the dimensions that listed in {TrimDimensions}. The order should match the order in {TrimDimensions}. If the corresponding position has no value to be give, it should be left with no value - no space or white space. This would be equivalent to the case where the highest value along that dimension is used instead.

If empty, it would be treated that all dimensions have no restriction on highs.

For example, the coverage has the following dimensions

```
{latitude, longitude, time, band}
```

```
latitude = [-90,90]
```

```
longitude = [-180, 180]
```

```
time = [1983, 2018]
```

```
band = [1, 15]
```

```
Given TrimDimensions=latitude,longitude,time
```

```
TrimLows=-70,,1988
```

```
TrimHighs=80,170,1998
```

This is a slicing operation that retrieve the data bounding by latitude = [-70, -70], longitude = [-180, 180], time=[1983, 2018], and band = [1, 15]. The output dimensions would be reduced to {longitude, band}. Two dimensions are reduced.

```
required: false
```

```
allowEmptyValue: true
```

```
schema:
```

```
type: string
```

Schema SliceDimensions and SlicePoints

Requirement	O5S.9
Requirement Name	/req/core/openapi-components-slicing-request
Description	These define the request parameters that are used in query for coverage slicing.

YAML

```
SliceDimensions:  
  name: slicedimlabels  
  in: query  
  description: >
```

This parameter is a string that holds a list of comma separated dimension names for slicing. The dimension name should match axislabels that are available in the offered coverage. The order should match the order in {SlicePoints}.

For example, the coverage has the following dimensions

```
{latitude, longitude, time, band}
```

```
latitude = [-90,90]
```

```
longitude = [-180, 180]
```

```
time = [1983, 2018]
```

```
band = [1, 15]
```

Given SliceDimensions=latitude,time

```
SlicePoints=-70,1988
```

This is a subsetting operation that retrieve the data bounding by latitude = [-70, 80], longitude = [-180, 170], time=[1983, 1998], and band = [1, 15]

required: false

allowEmptyValue: true

schema:

type: string

SlicePoints:

name: SlicePoints

in: query

description: >

This parameter is a string that holds a list of comma separated dimension names for slicing. The dimension name should match axislabels that are available in the offered coverage. The order should match the order in {SlicePoints}.

For example, the coverage has the following dimensions

```
{latitude, longitude, time, band}
```

```
latitude = [-90,90]
```

```
longitude = [-180, 180]
```

```
time = [1983, 2018]
```

```
band = [1, 15]
```

```
Given SlicePoints=latitude,time
```

```
TrimLows=-70,1988
```

This is a subsetting operation that retrieve the data bounding by latitude = [-70, 80], longitude = [-180, 170], time=[1983, 1998], and band = [1, 15]

```
required: false
```

```
allowEmptyValue: true
```

```
schema:
```

```
type: string
```

Schema Coverage

Requirement	O5S.10
Requirement Name	/req/core/openapi-components-coverage-response
Description	This defines the Coverage (in GML) returned from the WCS.

YAML

```
Coverage:
```

```
type: object
```

```
properties:
```

```
coverageFunction:
```

```
$ref: '#/components/schemas/coverageFunction'
```

```
metadata:
```

```
type: object
```

```
domainSet:
```

```
$ref: '#/components/schemas/domainSet'
```

```
rangeType:
```

```
$ref: '#/components/schemas/DataRecord'

rangeSet:

$ref: '#/components/schemas/rangeSet'

CoverageGML:

$ref: '#/components/schemas/Coverage'

rangeSet:

type: object

properties:

rangeSet:

$ref: '#/components/schemas/RangeSetType'

RangeSetType:

oneOf:

- type: array

items:

$ref: '#/components/schemas/ValueArray'

- type: array

items:

$ref: '#/components/schemas/AbstractScalarValueList'

- $ref: '#/components/schemas/DataBlock'

- type: object

properties:

File:

$ref: '#/components/schemas/FileType'

AbstractValue:

oneOf:
```


- \$ref: '#/components/schemas/AbstractScalarValue'
- \$ref: '#/components/schemas/AbstractScalarValueList'
- \$ref: '#/components/schemas/CompositeValue'
- \$ref: '#/components/schemas/ValueExtent'

ValueExtent:

oneOf:

- \$ref: '#/components/schemas/CategoryExtent'
- \$ref: '#/components/schemas/CountExtent'
- \$ref: '#/components/schemas/QuantityExtent'

type: object

CategoryExtent:

type: object

properties:

CategoryExtent:

\$ref: '#/components/schemas/CategoryExtentType'

CategoryExtentType:

allOf:

- \$ref: '#/components/schemas/CodeOrNilReasonListType'

CodeOrNilReasonListType:

allOf:

- \$ref: '#/components/schemas/NameOrNilReasonList'
- type: object

properties:

codeSpace:

type: string

NameOrNilReasonList:

```

type: array

items:

$ref: '#/components/schemas/NameOrNilReason'

NameOrNilReason:

allof:

- $ref: '#/components/schemas/NilReasonEnumeration'

- type: string

- type: string

CountExtent:

type: object

properties:

CountExtent:

$ref: '#/components/schemas/CountExtentType'

CountExtentType:

allof:

- $ref: '#/components/schemas/integerOrNilReasonList'

integerOrNilReasonList:

type: array

items:

$ref: '#/components/schemas/integerOrNilReason'

integerOrNilReason:

allof:

- $ref: '#/components/schemas/NilReasonEnumeration'

- type: integer

- type: string

```

```

QuantityExtent:
  type: object
  properties:
    QuantityExtent:
      $ref: '#/components/schemas/QuantityExtentType'
    QuantityExtentType:
      allOf:
        - $ref: '#/components/schemas/MeasureOrNilReasonListType'
    MeasureOrNilReasonListType:
      allOf:
        - $ref: '#/components/schemas/doubleOrNilReasonList'
        - type: object
      required: [
        'uom' ]
      properties:
        uom:
          $ref: '#/components/schemas/UomIdentifier'
        doubleOrNilReasonList:
          type: array
          items:
            $ref: '#/components/schemas/doubleOrNilReason'
        doubleOrNilReason:
          allOf:
            - $ref: '#/components/schemas/NilReasonEnumeration'
            - type: number
            - type: string

```

AbstractScalarValue:

oneOf:

- type: object

properties:

Quantity:

type: number

- type: object

properties:

Category:

type: string

- type: object

properties:

Count:

type: integer

- type: object

properties:

Boolean:

type: boolean

AbstractScalarValueList:

type: array

items:

\$ref: '#/components/schemas/AbstractScalarValue'

ValueArray:

type: object

properties:

ValueArray:

\$ref: '#/components/schemas/ValueArrayType'

ValueArrayType:

allOf:

- \$ref: '#/components/schemas/AbstractScalarValue'
- type: object

properties:

referenceSystem:

\$ref: '#/components/schemas/referenceSystem'

referenceSystem:

type: object

properties:

codeSpace:

type: string

uom:

\$ref: '#/components/schemas/UomIdentifier'

UomIdentifier:

allOf:

- type: object

properties:

UomSymbol:

type: string

pattern: '^:\n\r\tj'

- type: object

properties:

UomURI:

```

type: string

pattern: '([a-zA-Z][a-zA-Z0-9\-\+\.\]*:\|\.\/\|\.\/|#).*'

CompositeValue:

type: object

properties:

CompositeValue:

$ref: '#/components/schemas/CompositeValueType'

CompositeValueType:

allOf:

- $ref: '#/components/schemas/AbstractGMLType'

- type: array

items:

type: object

properties:

valueComponent:

$ref: '#/components/schemas/valueComponent'

- $ref: '#/components/schemas/valueComponents'

- $ref: '#/components/schemas/AggregationAttributeGroup'

AggregationAttributeGroup:

type: object

properties:

aggregationType:

$ref: '#/components/schemas/AggregationType'

AggregationType:

type: string

```

enum:

- set
- bag
- sequence
- array
- record
- table

valueComponents:

type: object

properties:

valueComponents:

\$ref: '#/components/schemas/ValueArrayType'

ValueArrayType:

allOf:

- type: array

items:

\$ref: '#/components/schemas/Value'

- \$ref: '#/components/schemas/OwnershipAttributeGroup'

valueComponent:

\$ref: '#/components/schemas/ValuePropertyType'

ValuePropertyType:

allOf:

- type: array

items:

\$ref: '#/components/schemas/Value'

Value:

oneOf:

- \$ref: '#/components/schemas/AbstractValue'
- \$ref: '#/components/schemas/AbstractGeometry'
- \$ref: '#/components/schemas/AbstractTimeObject'
- \$ref: '#/components/schemas/GMLNull'

GMLNull:

\$ref: '#/components/schemas/NilReasonType'

DataBlock:

type: object

title: gml-Datablock

properties:

DataBlock:

\$ref: '#/components/schemas/DataBlockType'

DataBlockType:

allof:

- type: object

title: gml-rangeParameters

properties:

rangeParameters:

\$ref: '#/components/schemas/AssociationRoleType'

File:

type: object

title: gmlFile

properties:

File:


```
$ref: '#/components/schemas/FileType'  
  
FileType:  
  
allOf:  
  
- type: object  
  
title: rangeParameters  
  
properties:  
  
rangeParameters:  
  
$ref: '#/components/schemas/AssociationRoleType'  
  
- oneOf:  
  
- type: object  
  
title: fileName  
  
properties:  
  
fileName:  
  
type: string  
  
- type: object  
  
title: fileReference  
  
properties:  
  
fileReference:  
  
type: string  
  
- type: object  
  
title: fileStructure  
  
properties:  
  
fileStructure:  
  
$ref: '#/components/schemas/CodeType'  
  
- type: object  
  
properties:
```

```

mimeType:
type: string
- type: object
properties:
compression:
type: string
AssociationRoleType:
allof:
- type: array
items:
type: object
- $ref: '#/components/schemas/OwnershipAttributeGroup'
- $ref: '#/components/schemas/AssociationAttributeGroup'

```

Field tags object

Requirement	O6
Requirement Name	/req/core/openapi-components-tags
Description	A set of predefined tags to group operations available to paths.

JSON

```

"tags":
[
  {
    "name": "Capabilities",
    "description":
      "Essential characteristics of this API including information about the data."
  },
  {
    "name": "Coverages",
    "description":
      "Access to data (coverages)."
  }
]

```

YAML

```

tags:
- name: Capabilities
  description: >-
    Essential characteristics of this API including information about the
    data.
- name: Coverages
  description: >-
    Access to data (coverages).

```

B.6.3. Service root path /

Get Operation

Requirement	C1
Requirement Name	/req/core/rootpath-get
Path	/

Requirement	C1
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /. Three media types may be supported: application/json, application/xml, and text/html.

JSON

```
"/":
{
  "get":
  {
    "summary": "landing page of this API",
    "description":
      "The landing page provides links to the API definition, the Conformance \
statements and the metadata about the coverage data in this dataset.",
    "operationId": "getLandingPage",
    "tags":
      ["Capabilities"],
    "responses":
      {
        "200":
          {
            "description": "links to the API capabilities",
            "content":
              {
                "application/json":
                  {
```

```

"schema":
{
"$ref": "#/components/schemas/root"
}
},
"text/html":
{
"schema":
{
"type": "string",
"example": "<!DOCTYPE html><html><head><title>landing page of this WCS REST API</title></head><body><h1/><ul><li><a href='http://data.example.org/' rel='self' type='application/json' title='this document' lang='en'></a></li><li><a href='http://data.example.org/api' rel='spec' type='application/json' title='the list of API specifications' lang='en'></a></li><li><a href='http://data.example.org/conformance' rel='self' type='application/json' title='WCS 3.0 conformance classes implemented by this server' lang='en'></a></li><li><a href='http://data.example.org/' rel='metadata' type='application/json' title='Metadata about the coverage set' lang='en'></a></li></ul></body></html>"
}
},
"application/xml":
{
"schema":
{
"$ref": "#/components/schemas/root"
}
}
}
}

```

```
}  
  
}
```

YAML

```
'/':  
  get:  
    summary: landing page of this API  
    description: >  
      The landing page provides links to the API definition, the Conformance  
      statements and the metadata about the coverage data in this dataset.  
    operationId: getLandingPage  
    tags:  
      - Capabilities  
    responses:  
      '200':  
        description: links to the API capabilities  
        content:  
          application/json:  
            schema:  
              $ref: '#/components/schemas/root'  
          text/html:  
            schema:  
              type: string  
        example: <!DOCTYPE html><html><head><title>landing page of this WCS REST  
API</title></head><body><h1><ul><li><a href='http://data.example.org/' rel='self'  
type='application/json' title="this document" lang='en'></a></li><li><a  
href='http://data.example.org/api' rel='spec' type='application/json' title="the list  
of API specifications" lang='en'></a></li><li><a
```

```
href='http://data.example.org/conformance' rel='self' type='application/json'
title="WCS 3.0 conformance classes implemented by this server"
lang='en'></a></li><li><a href='http://data.example.org/' rel='metadata'
type='application/json' title="Metadata about the coverage set"
lang='en'></a></li></ul></body></html>
```

application/xml:

schema:

\$ref: '#/components/schemas/root'

B.6.4. Service OpenAPI path /api

Get Operation

Requirement	C2
Requirement Name	/req/core/apipath-get
Path	/api
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /api.

JSON

```
"/api": {
  "get": {
    "summary": "List of the API specifications in OpenAPI",
    "description":
      "List the API specification documents.",
    "operationId": "getOpenApi",
    "tags": [
      "Capabilities"
    ],
    "responses": {
```

```

"200": {
  "description": "link to API description file.",
  "content": {
    "application/json": {
      "schema":
        "$ref": "#/components/schemas/openapi"
    },
    "text/html": {
      schema: {
        "type": "string",
        "example": "<!DOCTYPE html><html><head><title>List of this WCS REST API OpenAPI
specifications</title></head><body><h1><ul><li><a
href='http://data.example.org/api/openapi.json' rel='spec'
type='application/openapi+json;version=3.0' title='The API JSON definition'
lang='en'></a></li><li><a href='http://data.example.org/api/openapi.yaml' rel='spec'
type='application/openapi+yaml;version=3.0' title='The API YAML definition'
lang='en'></a></li></ul></body></html>"
      }
    },
    "application/xml": {
      "schema": {
        "$ref": "#/components/schemas/openapi"
      }
    }
  }
}

```



```
'/api':
  get:
    summary: list of the API specifications in OpenAPI
    description: >-
      List the API specification documents.
    operationId: getOpenApi
    tags:
      - Capabilities
    responses:
      '200':
        description: link to API description file.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/openapi'
          text/html:
            schema:
              type: string
        example: <!DOCTYPE html><html><head><title>List of this WCS REST API OpenAPI
        specifications</title></head><body><h1><ul><li><a
        href='http://data.example.org/api/openapi.json' rel='spec'
        type='application/openapi+json;version=3.0' title='The API JSON definition'
        lang='en'></a></li><li><a href='http://data.example.org/api/openapi.yaml' rel='spec'
        type='application/openapi+yaml;version=3.0' title='The API YAML definition'
        lang='en'></a></li></ul></body></html>
          application/xml:
            schema:
              $ref: '#/components/schemas/openapi'
```

B.6.5. Service OpenAPI path /conformance

Get Operation

Requirement	C3
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/core as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

JSON

```
"/conformance": {
  "get": {
    "summary": "information about standards that this API conforms to",
    "description":
      "list all requirements classes specified in a standard (e.g., WCS 2 OpenAPI 3.0 ) that
      the server conforms to",
    "operationId": "getRequirementsClasses",
    "tags": [
      "Capabilities"
    ],
    "responses": {
      "200": {
        "description": "the URIs of all requirements classes supported by the server",
        "content": {
          "application/json": {
```

```

"schema": {
  "$ref": "#/components/schemas/reqclasses"
}
},
"text/html": {
  "schema": {
    "type": "string",
    "example": "<!DOCTYPE html><html><head><title>Conformance of this WCS REST
API</title></head><body><h1/><ul><li>http://www.opengis.net/spec/wcs-
2/3.0/req/core</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/eo</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/update</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/insertdelete</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/geotiff</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/netcdf</li><li>http://www.opengis.net/spec/wcs-
2/3.0/req/geopackage</li></ul></body></html>"
  }
}
"application/xml": {
  "schema": {
    "$ref": "#/components/schemas/reqclasses"
  }
}
},
"default": {
  "description": "An error occurred.",
  "content": {
    "application/json": {

```


OpenAPI 3.0) that the server conforms to

operationId: `getRequirementsClasses`

tags:

- `Capabilities`

responses:

'200':

description: `the URIs of all requirements classes supported by the server`

content:

application/json:

schema:

\$ref: '#/components/schemas/reqclasses'

text/html:

schema:

type: `string`

example: `<!DOCTYPE html><html><head><title>Conformance of this WCS REST API</title></head><body><h1>http://www.opengis.net/spec/wcs-2/3.0/req/corehttp://www.opengis.net/spec/wcs-2/3.0/req/eohttp://www.opengis.net/spec/wcs-2/3.0/req/updatehttp://www.opengis.net/spec/wcs-2/3.0/req/insertdeletehttp://www.opengis.net/spec/wcs-2/3.0/req/geotiffhttp://www.opengis.net/spec/wcs-2/3.0/req/netcdfhttp://www.opengis.net/spec/wcs-2/3.0/req/geopackage</body></html>`

application/xml:

schema:

\$ref: '#/components/schemas/reqclasses'

default:

description: `An error occured.`

content:

application/json:

```

schema:
  $ref: '#/components/schemas/exception'

text/html:
  schema:
    type: string

application/xml:
  schema:
    $ref: '#/components/schemas/exception'

```

B.6.6. Service OpenAPI path /coverages

Get Operation

Requirement	C4
Requirement Name	/req/core/coveragespath-get
Path	/coverages
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Parameters	q – in query, string, keywords search
	bbox – in path, comma-separated bounding box defined by {west, south, min. height, east, north, max. height}
	time – in path, comma-separated time bounded with earliest and latest.
	startIndex – in query, integer, the start index for returned coverages.
	count – in query, integer, number of coverages to be returned counting start from startIndex
Description	The server SHALL support the HTTP GET operation at the path /coverages. The default number of items is 10. If there are less than 10 coverages served in the WCS, the actual number of items are listed.

YAML

```

'/coverages':

```

```
get:
summary: describe coverages available in the WCS
operationId: describeCoverages
tags:
- Capabilities
responses:
'200':
description: List metadata about the coverages.
content:
application/json:
schema:
$ref: '#/components/schemas/contents'
text/html:
schema:
type: string
application/xml:
schema:
$ref: '#/components/schemas/contents'
default:
description: An error occurred.
content:
application/json:
schema:
$ref: '#/components/schemas/exception'
text/html:
```



```
schema:
  type: string
  application/xml:
    schema:
      $ref: '#/components/schemas/exception'
  parameters:
    - name: q
      in: query
      description: keywords to search against the coverages.
      required: false
      schema:
        type: string
        - $ref: '#/components/parameters/bbox'
        - $ref: '#/components/parameters/time'
        - name: startIndex
      in: query
      description: StartIndex of the (matching criteria if q is allowed and set ) results to
        be returned. The default is 1-based index.
      required: false
      schema:
        type: integer
        - name: count
      in: query
      description: Number of the (matching criteria if q is allowed and set ) results to be
        returned.
      required: false
      schema:
```

```
type: integer
minimum: 0
maximum: 10000
default: 10
```

B.6.7. Service OpenAPI path /coverages/{coverageId}

Get Operation

Requirement	C4
Requirement Name	/req/core/coverages/coverageidpath-get
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.
Description	The server SHALL support the HTTP GET operation at the path /coverages/{coverageId}. It returns the metadata for the coverage identified by {coverageId}.

YAML

```
'/coverages/{coverageId}':
  get:
    summary: 'describe the {coverageId} coverage'
    operationId: describeCoverage
    tags:
      - Capabilities
    parameters:
      - $ref: '#/components/parameters/coverageId'
    responses:
      '200':
        description: 'Metadata about the {coverageId} coverage shared by this API.'
        content:
```

```

application/json:
  schema:
    $ref: '#/components/schemas/coverageDescription'
text/html:
  schema:
    type: string
application/xml:
  schema:
    $ref: '#/components/schemas/coverageDescription'
  default:
    description: An error occurred.
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/exception'
    text/html:
      schema:
        type: string

```

B.6.8. Service OpenAPI path /coverages/{coverageId}/rangeSet

Get Operation

Requirement	C4
Requirement Name	/req/core/coverages/coverageidpath/rangeSet-get
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	C4
	TrimDimensions – in path, comma-separated string, dimensions for trimming
	TrimLows – in path, comma-separated string, low bounds of dimensions for trimming
	TrimHighs – in path, comma-separated string, high bounds of dimensions for trimming
	SliceDimensions – in path, comma-separated string, dimensions for slicing
	SlicePoints – in path, comma-separated string, point values along dimensions for slicing
Description	The server SHALL support the HTTP GET operation at the path /coverages/{coverageId}/rangeSet. It returns the metadata for the coverage identified by {coverageId}.

YAML

```

'/coverages/{coverageId}/rangeSet':
  get:
    summary: 'retrieve coverage {coverageId} within specific range'
    description: >-
      A coverage can often be viewed as a single layer. A dataset may contain several
      coverages.
    operationId: getCoverages
    tags:
      - Coverages
    parameters:
      - $ref: '#/components/parameters/coverageId'
      - $ref: '#/components/parameters/TrimDimensions'
      - $ref: '#/components/parameters/TrimLows'
      - $ref: '#/components/parameters/TrimHighs'
      - $ref: '#/components/parameters/SliceDimensions'

```

- \$ref: '#/components/parameters/SlicePoints'

responses:

'200':

description: >-

Information about the coverages in a given dataset.

content:

application/gml+xml:

schema:

\$ref: '#/components/schemas/CoverageGML'

application/json:

schema:

\$ref: '#/components/schemas/CoverageGML'

default:

description: An error occured.

content:

application/json:

schema:

\$ref: '#/components/schemas/exception'

text/html:

schema:

type: string

application/gml+xml:

schema:

\$ref: '#/components/schemas/exception'

B.7. Range Subsetting Extension

B.7.1. Overview

The Range Subsetting Extension requirement of the WCS REST API specification is required to enable parameters for rangesubset of path `/*/rangeSet`. The following table gives the path in the core class that would be affected.

Table 2 – Paths for WCS with Range Subsetting Support

PATH	METHOD	DESCRIPTION
<code>/eocoverages/{eocoverageId}/rangeSet</code>	GET	Gets the accessible coverage data.

B.7.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field paths object

Field components object

Parameter RANGESUBSET

Requirement	EOP.1
Requirement Name	<code>/req/rsub/openapi-components-rangesubset</code>
Description	This defines rangesubset parameter.

Example 1 The following KVP request snippet extracts the component named “red” from the coverage addressed (assuming that these components are defined in the coverage’s range type):

```
...&RANGESUBSET=red&...
```

Example 2 The following KVP request snippet extracts the three components named “nir”, “red”, and “green” from the coverage addressed (assuming that these components are defined in the coverage’s range type):

```
...&RANGESUBSET=nir,red,green&...
```

Example 3 The following KVP request snippet extracts the three components named “red”, “green”, and “blue” from the coverage addressed (assuming that these components are defined in the coverage’s range type), but changes its sequence over the original coverage range type definition:

```
...&RANGESUBSET=green,red,blue&...
```

Example 4 The following KVP request snippet extracts a number of components from the coverage addressed, starting from component “nir” up to component “green”. Assuming that the coverage’s

range type contains, in contiguous XML document order, “nir”, “red”, and “green”, the resulting coverage will contain these three range components “nir”, “red”, and “green”:

...&RANGESUBSET=nir:green&...

Example 5 The following KVP request snippet shows mixing of the previously exemplified component selection; for a range type consisting of band01 to band36 (with the obvious order of bands), the request results in a record containing the components band01, band03, band04, band05, band19, band20, band21:

...&RANGESUBSET=band01,band03:band05,band19:band21&...

B.7.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/rsub as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.7.4. Service OpenAPI path /eocoverages/{eocoverageId}/rangeSet

This path is an example path to be affected. Any path ended rangeSet may be affected.

Get Operation

Requirement	EO5
Requirement	/req/eo/eocoverages/eocoverageidpath/rangeSet-get
Path	/eocoverages/{eocoverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	
	RANGESUBSET – in path, comma-separated string (interval is colon-separated)

Requirement	EO5
Description	The server SHALL support the HTTP GET operation at the path /eocoverages/{eocoverageId}/rangeSet. It returns the data for the EO coverage identified by {eocoverageId}. If no parameters are given, the whole EO coverage is returned.

B.8. Earth Observation Extension

B.8.1. Overview

The Earth Observation Extension requirement of the WCS REST API specification is required to implement the following paths and basic operations on resource EO coverages and EO coverage data (rangeSet).

Table 2 – Paths for EOWCS

PATH	METHOD	DESCRIPTION
/eocoverages	GET	Gets the description/metadata of all the EO-coverages (i.e. dataset series, dataset, or stitched mosaics) offered by the service
/eocoverages/{eocoverageId}	GET	Gets the description/metadata of the EO coverages (i.e. dataset series, dataset, or stitched mosaics)
/eocoverages/{eocoverageId}/rangeSet	GET	Gets the accessible EO coverage or a subset thereof (i.e. dataset series, dataset, or stitched mosaics). If the EO coverage is a dataset series (collection of datasets), this path may be disabled when there is no support of accessing packed data for the data series.
/eocoverages/{eodataserieId}/{eodatasetId}	GET	Gets the description/metadata of the EO coverage as a specific dataset when eodataserieId indicate a data series that consist of datasets (smallest accessible data). This is a convenient path to manage EO coverage resources in the structure of two level hierarchs.

PATH	METHOD	DESCRIPTION
/eocoverages/{eodataseriedId}/{eodatasetId}/rangeSet	GET	Gets the accessible EO coverage or a subset thereof (i.e. dataset). This is a data access support for the convenient path to a dataset resource in a data series when the data are managed in two level hierarchs.

B.8.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field paths object

Requirement	EO1
Requirement Name	/req/eo/openapi-paths
Description	The following paths should be defined: /eocoverages, /eocoverages/{eocoverageId}, /eocoverages/{eocoverageId}/rangeSet, in addition to /, /api, /conformance

JSON

```

"paths":{
"/": {},
"/api": {},
"/conformance": {}
"/eocoverages": {}
"/eocoverages/{eocoverageId}": {}
"/eocoverages/{eocoverageId}/rangeSet": {}
}

```

YAML

```

paths:
  /:
  /api:
  /conformance:
  /eocoverages:
  /eocoverages/{eocoverageId}:
  /eocoverages/{eocoverageId}/rangeSet:

```

Field components object

Parameter eocoverageId

Requirement	EOP.1
Requirement Name	/req/eo/openapi-components-eocoverageid
Description	This defines the identifier of EO coverage.

YAML

```

eocoverageId:
  name: eocoverageId
  in: path
  required: true
  description: Identifier (name) of a specific EO coverage
  schema:
  type: string

```

Parameter eodataseriessid

Requirement	EOP.2
Requirement Name	/req/eo/openapi-components-eodataseriessid
Description	This defines the identifier of EO data series.

YAML

```

eodataserId:
  name: eodataserId
  in: path
  required: true
  description: Identifier (name) of a specific EO dataseries.
  schema:
  type: string

```

Parameter eodataserId

Requirement	EOP.3
Requirement Name	/req/eo/openapi-components-eodataserid
Description	This defines the identifier of EO dataset.

YAML

```

eodataserId:
  name: eodataserId
  in: path
  required: true
  description: Identifier (name) of a specific EO dataset.
  schema:
  type: string

```

Schema EOCoverage

Requirement EOS.1	Requirement Name
/req/eo/openapi-components-eocoverage-response	Description

Field tags object

Requirement EOT	Requirement Name
/req/eo/openapi-components-tags	Description

JSON

```
"tags":  
[  
  {  
    "name": "Capabilities",  
    "description":  
      "Essential characteristics of this API including information about the data."  
  },  
  {  
    "name": "Coverages",  
    "description":  
      "Access to data (coverages)."  
  },  
  {  
    "name": "EOCoverages",  
    "description":  
      "Access to data (EO coverages)."  
  }  
]
```

YAML

tags:

- name: Capabilities

description: >-

Essential characteristics of this API including information about the data.

- name: Coverages

description: >-

Access to data (coverages).

- name: EOCoverages

description: >-

Access to data (EO coverages).

B.8.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/eo as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.8.4. Service OpenAPI path /eocoverages

Get Operation

Requirement	EO3
Requirement Name	/req/eo/eocoveragespath-get

Requirement	EO3
Path	/eocoverages
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Parameters	q – in query, string, keywords search
	bbox – in path, comma-separated bounding box defined by {west, south, min. height, east, north, max. height}
	time – in path, comma-separated time bounded with earliest and latest.
	startIndex – in query, integer, the start index for returned EO coverages.
	count – in query, integer, number of EO coverages to be returned counting start from startIndex
Description	The server SHALL support the HTTP GET operation at the path /coverages. The default number of items is 10. If there are less than 10 EO coverages served in the WCS, the actual number of items are listed.

B.8.5. Service OpenAPI path /eocoverages/{eocoverageId}

Get Operation

Requirement	EO4
Requirement Name	/req/eo/eocoverages/eocoverageidpath-get
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.
Description	The server SHALL support the HTTP GET operation at the path /eocoverages/{eocoverageId}. It returns the metadata for the EO coverage identified by {eocoverageId}.

B.8.6. Service OpenAPI path /eocoverages/{eocoverageId}/rangeSet

Get Operation

Requirement	EO5
Requirement Name	/req/eo/eocoverages/eocoverageidpath/rangeSet-get

Requirement	EO5
Path	/eocoverages/{eocoverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	eocoverageId – in path, string, identifier of the coverage.
	TrimDimensions – in path, comma-separated string, dimensions for trimming
	TrimLows – in path, comma-separated string, low bounds of dimensions for trimming
	TrimHighs – in path, comma-separated string, high bounds of dimensions for trimming
Description	The server SHALL support the HTTP GET operation at the path /eocoverages/{eocoverageId}/rangeSet. It returns the data for the EO coverage identified by {eocoverageId}. If no parameters are given, the whole EO coverage is returned.

B.8.7. Service OpenAPI path /eocoverages/{eodataseriesId}

Get Operation

Requirement	EO3
Requirement Name	/req/eo/eocoverages/eodataseriespath-get
Path	/eocoverages/{eodataseriesId}
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Parameters	eodataseriesId – in path, identifier of EO data series or identifier of EO coverage when type is Data Series.
	bbox – in query, comma-separated bounding box defined by {west, south, min. height, east, north, max. height}
	time – in query, comma-separated time bounded with earliest and latest.
	startIndex – in query, integer, the start index for returned EO coverages.
	count – in query, integer, number of EO coverages to be returned counting start from startIndex

Requirement	EO3
Description	The server SHALL support the HTTP GET operation at the path /coverages. The default number of items is 10. If there are less than 10 EO coverages served in the WCS, the actual number of items are listed.

B.8.8. Service OpenAPI path /eocoverages/{eodataseriesId}/{eodatasetId}

Get Operation

Requirement	EO6
Requirement Name	/req/eo/eocoverages/eodataseriesId/eodatasetidpath-get
Path	/eocoverages/{eodataseriesId}/{eodatasetId}
Parameters	eodataseriesId – in path, string, identifier of the EO data series.
	eodatasetId – in path, string, identifier of the EO dataset
Description	This is optional for datasets to be organized in two levels. The server SHALL support the HTTP GET operation at the path /eocoverages/{eodataseriesId}. It returns the metadata for the EO coverage (dataset) identified by {eodataseriesId}.

B.8.9. Service OpenAPI path /eocoverages/{eodataseriesId}/{eodatasetId}/rangeSet

Get Operation

Requirement	EO5
Requirement Name	/req/eo/eocoverages/eocoverageidpath/rangeSet-get
Path	/eocoverages/{eocoverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	eodataseriesId – in path, string, identifier of the EO data series.
	eodatasetId – in path, string, identifier of the EO dataset
	TrimDimensions – in path, comma-separated string, dimensions for trimming
	TrimLows – in path, comma-separated string, low bounds of dimensions for trimming

Requirement	EO5
	TrimHighs – in path, comma-separated string, high bounds of dimensions for trimming
	bbox – in path, comma-separated bounding box defined by {west, south, min. height, east, north, max. height}
	time – in path, comma-separated time bounded with earliest and latest.
Description	The server SHALL support the HTTP GET operation at the path /eocoverages/{eodataserie s}/{eodatase tId}/rangeSet. It returns the data for the EO dataset identified by {eodatase tId}. If no parameters are given, the whole EO dataset without subsetting is returned.

B.9. Transaction Insertion Deletion

B.9.1. Overview

The Transaction Insertion Deletion Conformance requirement of the WCS REST API specification is required to implement the following paths and basic operations on resource EO coverages and EO coverage data (rangeSet).

Table 4 – Operations of Insertion and Deletion

PATH	METHOD	DESCRIPTION
/coverages/{coverageId}	POST	Create the metadata for coverage identified by {coverageId}
	DELETE	Delete the metadata for coverage identified by {coverageId}
/coverages/{coverageId}/rangeSet	POST	Create the data for coverage identified by {coverageId}. Failed if {coverageId} is used.
	DELETE	Delete the data for coverage identified by {coverageId}.
/coverages/rangeSet	POST	Create the data for coverage identified and assigned a unique coverageId to be returned.

B.9.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or

definitions are given under components.

Field paths object

Requirement EO1	Requirement Name
/req/eo/openapi-paths	Description

JSON

```
"paths":{  
  "/": {},  
  "/api": {},  
  "/conformance": {}  
  "/coverages": {}  
  "/coverages/{coverageId}": {}  
  "/coverages/rangeSet": {}  
  "/coverages/{coverageId}/rangeSet": {}  
}
```

YAML

```
paths:  
  /:  
  /api:  
  /conformance:  
  /eocoverages:  
  /eocoverages/{eocoverageId}:  
  /eocoverages/{eocoverageId}/rangeSet:
```

Field tags object

B.9.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/insertdelete as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.9.4. Service OpenAPI path /coverages/{coverageId}

POST Operation

Requirement	ID1
Requirement Name	/req/insertdelete/coverages/coverageidpath-post
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.
Description	<p>The server SHALL support the HTTP POST operation at the path /coverages/{coverageId} that creates the metadata for the coverage.</p> <p>HTTP code: 423 – when the resource with identifier {coverageId} exists.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

DELETE Operation

Requirement	ID2
Requirement Name	/req/insertdelete/coverages/coverageidpath-delete
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	ID2
Description	<p>The server SHALL support the HTTP DELETE operation at the path /coverages/{coverageId} that delete the metadata for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.9.5. Service OpenAPI path /coverages/{coverageId}/rangeSet

POST Operation

Requirement	ID3
Requirement Name	/req/insertdelete/coverages/coverageidpath/rangeSet-post
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.
Description	<p>The server SHALL support the HTTP POST operation at the path /coverages/{coverageId}/rangeSet that creates the data for the coverage.</p> <p>HTTP code: 423 – when the resource with identifier {coverageId} exists.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

DELETE Operation

Requirement	ID4
Requirement Name	/req/insertdelete/coverages/coverageidpath/rangeSet-delete
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	ID4
Description	<p>The server SHALL support the HTTP DELETE operation at the path /coverages/{coverageId}/rangeSet that deletes the data for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.9.6. Service OpenAPI path /coverages/rangeSet

POST Operation

Requirement	ID5
Requirement Name	/req/insertdelete/coverages/coverageidpath/rangeSet-post
Path	/coverages/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.
Description	<p>The server SHALL support the HTTP POST operation at the path /coverages/rangeSet that creates the data for the coverage by generating a unique {coverageId} which does not have both metadata and data resources with the generated id {coverageId}.</p> <p>HTTP code: 500 – when the resource with identifier {coverageId} cannot be created by the server due to internal error.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.10. Transaction Update

B.10.1. Overview

The Transaction Update Conformance requirement of the WCS REST API specification is required to implement the following paths and basic operations on resource EO coverages and EO coverage data (rangeSet).

Table 4 – Operations of Insertion and Deletion

PATH	METHOD	DESCRIPTION
/coverages/{coverageId}	PUT	Update the metadata for coverage identified by {coverageId}
/coverages/{coverageId}/rangeSet	PUT	Update the data for coverage identified by {coverageId}. Failed if {coverageId} is not used.

B.10.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field paths object

Requirement	EO1
Requirement Name	/req/eo/openapi-paths
Description	The following paths should be added: /coverages/rangeSet

JSON

```

"paths":{
  "/": {},
  "/api": {},
  "/conformance": {}
  "/coverages": {}
  "/coverages/{coverageId}": {}
  "/coverages/rangeSet": {}
  "/coverages/{coverageId}/rangeSet": {}
}

```

YAML

```

paths:
  /:
  /api:
  /conformance:
  /eocoverages:
  /eocoverages/{eocoverageId}:
  /eocoverages/{eocoverageId}/rangeSet:

```

B.10.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/core/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/update as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.10.4. Service OpenAPI path /coverages/{coverageId}

UPDATE Operation

Requirement	U1
Requirement Name	/req/insertdelete/coverages/coverageidpath-update
Path	/coverages/{coverageId}
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	U1
Description	<p>The server SHALL support the HTTP UPDATE operation at the path /coverages/{coverageId} that creates the metadata for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.10.5. Service OpenAPI path /coverages/{coverageId}/rangeSet

UPDATE Operation

Requirement	U2
Requirement Name	/req/insertdelete/coverages/coverageidpath/rangeSet-update
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/gml+xml
Parameters	coverageId – in path, string, identifier of the coverage.
Description	<p>The server SHALL support the HTTP UPDATE operation at the path /coverages/{coverageId}/rangeSet that updates the data for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.11. GeoTiff Coverage

B.11.1. Overview

This enables the WCS to support coverage encoded in GeoTiff.

B.11.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field components object

Schema geotiffcoverage

Requirement	GC1
Requirement Name	/req/geotiff/openapi-components-geotiffcoverage
Description	This defines the coverage in the encoding of GeoTiff.

B.11.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/geotiff/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/geotiff as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.11.4. Service OpenAPI path /coverages/{coverageId}/rangeSet

This change applies to rangeSet on all operation. Media type image/tiff should be supported to enable the support of GeoTiff as coverage encoding.

Requirement	GC2
Requirement Name	/req/geotiff/coverages/coverageidpath/rangeSet-encoding
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: image/tiff
	Content-Type: image/tiff
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	GC2
Description	<p>The server SHALL support the HTTP UPDATE operation at the path /coverages/{coverageId}/rangeSet that updates the data for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.12. NetCDF Coverage

B.12.1. Overview

This enables the WCS to support coverage encoded in NetCDF.

B.12.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field components object

Schema geotiffcoverage

Requirement	NC1
Requirement Name	/req/netcdf/openapi-components-netcdfcoverage
Description	This defines the coverage in the encoding of NetCDF.

B.12.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/geotiff/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml

Requirement	EO2
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/netcdf as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.12.4. Service OpenAPI path /coverages/{coverageId}/rangeSet

This change applies to rangeSet on all operation. Media type application/x-netcdf should be supported to enable the support of GeoTiff as coverage encoding.

Requirement	NC2
Requirement Name	/req/geotiff/coverages/coverageidpath/rangeSet-encoding
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/x-netcdf
	Content-Type: application/x-netcdf
Parameters	coverageId – in path, string, identifier of the coverage.
Description	<p>The server SHALL support the HTTP UPDATE operation at the path /coverages/{coverageId}/rangeSet that updates the data for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

B.13. GeoPackage Coverage

B.13.1. Overview

This enables the WCS to support coverage encoded in GeoPackage.

B.13.2. OpenAPI Object

This section defines the top-level attributes of the WCS OpenAPI object. Shared components or definitions are given under components.

Field components object

Schema geotiffcoverage

Requirement	GPKG1
Requirement Name	/req/geopackage/openapi-components-geopackagecoverage
Description	This defines the coverage in the encoding of GeoPackage.

B.13.3. Service OpenAPI path /conformance

Get Operation

Requirement	EO2
Requirement Name	/req/geotiff/conformancepath-get
Path	/conformance
Headers	Accept: application/json
	Accept: text/html
	Accept: application/xml
Description	The server SHALL support the HTTP GET operation at the path /conformance. The response shall list http://www.opengis.net/spec/wcs-2/3.0/req/geopackage as the conformance class when the core class is implemented, where wcs-2 identifies the version of WCS and 3.0 identifies the version of OpenAPI.

B.13.4. Service OpenAPI path /coverages/{coverageId}/rangeSet

This change applies to rangeSet on all operation. Media type application/x-geopackage should be supported to enable the support of GeoTiff as coverage encoding.

Requirement	GPKG2
Requirement Name	/req/geotiff/coverages/coverageidpath/rangeSet-encoding
Path	/coverages/{coverageId}/rangeSet
Headers	Accept: application/x-geopackage
	Content-Type: application/x-geopackage
Parameters	coverageId – in path, string, identifier of the coverage.

Requirement	GPKG2
Description	<p>The server SHALL support the HTTP UPDATE operation at the path /coverages/{coverageId}/rangeSet that updates the data for the coverage.</p> <p>HTTP code: 404 – when the resource with identifier {coverageId} does not exist.</p> <p>HTTP code: 200 – success. The coverageId is returned.</p>

Appendix C: Revision History

NOTE

Highlights

This appendix tracks the revisions.

Table 17. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
August 3, 2018	Eguene G. Yu	.1	all	initial version
October 16, 2018	Eugene G. Yu	.2	all	Re-format & add contents
October 24, 2018	Eugene G. Yu	.3	summary, references, terms, revision history, discussions	Incorporated part of the contents provided by Stephan Meißl and Sergio Ferraresi
October 25, 2018	Eugene G. Yu	.4	summary, profile, use cases, discussions, revision history, discussions, coverage encoding	Incorporated the contents provided by Sizhe Wang and Sergio Ferraresi
October 26, 2018	Eugene G. Yu	.5	cf modeling, cis modeling	Added CALIOP swath coverage modeling.
October 30, 2018	Eugene G. Yu	.6	WCS REST API, Appendix A and B	Reformatting and moved full specification of WCS REST API as an appendix. Add examples.
November 1, 2018	Eugene G. Yu	.6.1	terms, encoding	Incorporated typo fixes and suggested clarifications from Chris Little (OGC).
November 14, 2018	Eugene G. Yu	.6.2	Section 7, appendix B	Incorporating comments from James Passmore and Ingo Simonis.

Date	Editor	Release	Primary clauses modified	Descriptions
November 19, 2018	Eugene G. Yu	6.3	all sections	Revising the ER with comments from Dr. Gobe Hobona
November 20, 2018	Eugene G. Yu	6.4	Chapter 11, Chapter 12, and Appendix A.	Adding details from implementations and use cases.
November 20, 2018	Eugene G. Yu	6.4.1	Appendix A.	One minor fix.
December 10, 2018	Eugene G. Yu	6.4.2	Appendix A.	Revisions with inputs and comments from Dr. Carl Reed.

Appendix D: Bibliography

1. FGDC: Content Standard for Remote Sensing Swath Data. Federal Geographic Data Committee, Reston, Virginia, USA (1999).
2. Di, L., Kobler, B.: NASA Standards for Earth Remote Sensing Data. In: Dowman, I. and Janssen, L. (eds.) ISPRS Archives – Volume XXXIII Part B2, 2000. pp. 147–155. ISPRS, Amsterdam, The Netherlands (2000).
3. Di, L.: The Open GIS Web Service Specifications for Interoperable Access and Services of NASA EOS Data. In: Earth Science Satellite Remote Sensing. pp. 220–232. Springer, Berlin, Heidelberg (2006).
4. Jelenak, A., Santek, D., Yang, K., Carp, R.: Encoding of Swath Data in the Climate and Forecast Convention. UniData, Boulder, Colorado, USA (2018).
5. Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Blower, J., Caron, J., Signell, R., Bentley, P., Rappa, G., Håfjck, H., Pamment, A., Jckes, M., Raspaud, M.: NetCDF Climate and Forecast (CF) Metadata Conventions. CF Conventions (2014).
6. Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Blower, J., Caron, J., Signell, R., Bentley, P., Rappa, G., Håfjck, H., Pamment, A., Jckes, M., Raspaud, M.: NetCDF Climate and Forecast (CF) Metadata Conventions. CF Conventions (2018).
7. Franch, B., Roger, J.C., Vermote, E.F.: Suomi-NPP VIIRS Surface Reflectance Algorithm Theoretical Basis Document (ATBD). (2016).
8. Roger, J.C., Vermote, E.F., Devadiga, S., Ray, J.P.: Suomi -NPP VIIRS Surface Reflectance User's Guide. (2016).
9. Taaheri, A., Rodrigues, K.: HDF-EOS Library User's Guide Volume 1: Overview and Examples. Raytheon Company, Riverdale , Maryland (2017).
10. Taaheri, A., Rodrigues, K.: HDF-EOS Library User's Guide Volume 2: Function Reference Guide. Raytheon Company, Riverdale , Maryland (2017).
11. Taaheri, A., Rodrigues, K.: HDF - EOS Interface Based on HDF5, Volume 1: Overview and Examples. Raytheon Company, Riverdale , Maryland (2017).
12. Taaheri, A., Rodrigues, K.: HDF-EOS Interface Based on HDF5, Volume 2: Function Reference Guide. Raytheon Company, Riverdale , Maryland (2017).