

Open Geospatial Consortium

Submission Date: 2021-10-06

Approval Date: 2021-10-06

Publication Date: 2021-10-07

External identifier of this OGC® document: <http://www.opengis.net/doc/DP/GDC>

Internal reference number of this OGC® document: 21-067

Category: OGC® Discussion Paper

Editor: Ingo Simonis (OGC)

OGC: Towards Data Cube Interoperability

Copyright notice

Copyright © 2021 Open Geospatial Consortium

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

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore 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, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Discussion Paper

Document subtype:

Document stage: Final

Document language: English

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 NON-INFRINGEMENT 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.

Table of Contents

1. What is a Data Cube?	5
2. Do we need an Abstract Model for Data Cubes?	8
3. User's Perspective	9
4. Data Cube Provider Perspective	11
5. Discussion	13
6. Commercial Perspective	15
7. Next Activities	16
Annex A: Revision History	17

i. Abstract

Data cubes, multidimensional arrays of data, are used frequently these days, but differences in design, interfaces, and handling of temporal characteristics are causing interoperability challenges for anyone interacting with more than one solution. To address these challenges, the Open Geospatial Consortium (OGC) and the Group on Earth Observation (GEO) invited global data cube experts to discuss state-of-the-art and way forward at the “Towards Data Cube Interoperability” workshop. The two-day workshop, conducted in late April 2021, started with a series of pre-recorded position statements by data cube providers and data cube users. These videos served as the entry points for intense discussions that not only produced a new definition of the term ‘data cube’ (by condensing and shifting emphasize on what is known as the [six faces model](#)), but also pointed out a wide variety of expectations with regards to data cube behaviour and characteristics as well as data cube usage patterns. This report summarizes the various perspectives and discusses the next steps towards efficient usage of data cubes. It starts with the new definition of the term Data Cube, as this new understanding drives several recommendations discussed later in this report. The report includes further discussion that followed the actual workshop, mainly conducted in the context of the Geo Data Cube task in [OGC Testbed-17](#).

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

Geo Data Cube, Interoperability, Software architecture

iii. Preface

NOTE

This document has originated in work undertaken by OGC staff together with the Group on Earth Observation (GEO) and workshop participants. 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.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

OGC

v. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Ingo Simonis (OGC)

Chapter 1. What is a Data Cube?

Existing definitions coming from the (geospatial) computer science domain often focus on data structure aspects exclusively. Here, a data cube is defined as an multi-dimensional ("n-D") array of values, with emphasis on the fact that "cube" is just a metaphor to help illustrate a data structure that can in fact be 1- dimensional, 2-dimensional, 3-dimensional, or higher-dimensional. The dimensions may be coordinates or enumerations, e.g., categories.

This workshop emphasized the need to leave these computer-science based definitions behind and focus on the user perspective instead. What is a data cube from the user's perspective? We currently observe a general shift from data centric to user centric perspectives. Users don't care if data is stored in a relational database, in a cloud-based object store, or a file server. They are interested in the access mechanisms to the data and the processing algorithms they can apply.

The workshop was conducted within the location or geo community. Thus, the terms geo data cube and data cube are used interchangeably. It is therefore assumed that each cube has some spatial characteristics. Though no formal consensus process was applied, the following definition describes the tenor of the vast majority of workshop participants:

"A (geo) data cube is a discretized model of the earth that offers estimated values of certain variables for each partition of the Earth's surface called a cell. A data cube instance may provide data for the whole Earth or a subset thereof. Ideally, a data cube is dense (i.e., does not include empty cells) with regular cell distance for its spatial and temporal dimensions. A data cube describes its basic structure, i.e., its spatial and temporal characteristics and its supported variables (also known as 'properties'), as metadata. It is further defined by a set of functions. These functions describe the available discovery, access, view, analytical, and processing methods that are supported to interact with the data cube."

As it becomes apparent, the cube is described for the user, not the data. It does not matter if the cube contains one, two, or three spatial dimensions. Time can be modeled as a set of additional dimensions, though in most cases, there is likely only a single temporal dimension that describes the time of observation. Other temporal dimensions include for example 'validity time' (often used in the simulation community to describe when and how long a projected value is valid).

Spatially, the ideal data cube is dense with no gaps between cells. Each cell represents an area in the real world and the set of all cells represents a continuous area without holes. Such a cube allows the retrieval of property values for any location within the bounds of the data cube. Broader definitions of a data cube support cube structures that are not spatially dense. In most cases, these are point-oriented data cubes. Here, individual data points are distributed in space and regularly ordered in the data cube. In this case, the data cubes do not contain values for locations in between data points, but may offer interpolation methods to calculate property values at any location. Examples include data cubes with a set of measuring stations that line up all stations in a single dimension. The stations provide data for discrete point locations and any property value for locations in between stations needs to be interpolated. Whereas the data cube purists insisted on the spatially dense criterion as an essential characteristic for a data cube, the majority of the workshop participants accepted the broader definition as long as the user is sufficiently informed about the applied interpolation methods.

Still, the issue could not be fully solved at the workshop, which is why the definition provided herein speaks of the “ideal data cube” being spatially dense. The following figure shows different implementations of data cubes. Each cell may contain 1 to many variables. Time can be among these variables.

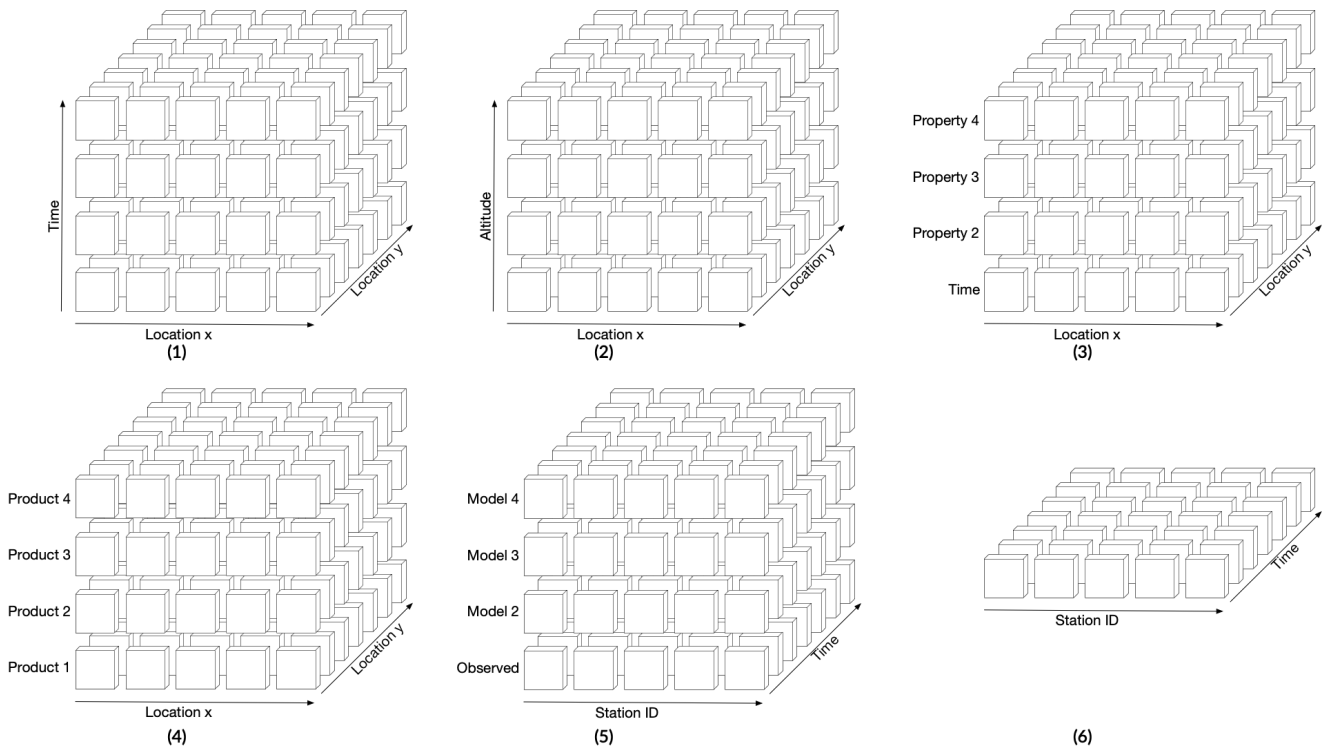


Figure 1. Different implementations of data cubes

In the figure above, cube (1) organizes cells along two spatial and one temporal dimension. Cube (2) adds altitude as a third spatial dimension. Here, time could be handled as a fourth dimension or becomes part of the variables expressed in each cell. The property versus dimension pattern is further illustrated in cube (3), which organizes time similar to other variables (properties) in a specific dimension. Technically, any dimension can be transformed into a property of a cell and vice versa. It depends on the specific set of questions that users post against the data cube. Thus, property versus dimension is not a technical challenge, but rather a decision to be made to provide the best user experience to the data cube customer, which could lead to an even stronger user-oriented definition of data cube: ‘The ideal data cube follows the mental model of its user group’.

The following cube implementations (4) to (6) illustrate further possible implementations; all following the ‘broader’ definition as discussed above. Cube (4) uses two spatial dimensions and represents different products in the third dimension. Here, cells along the product axis may have different variables. Cube (5) and cube (6) represent a set of stations.

A cube does not need to support temporal dimensions. Temporal characteristics might be expressed as property values within the attribute vector per cell. Alternatively, a cube can provide any number of temporal dimensions, i.e., treat temporal characteristics as first class citizens, which allows efficient exploration of the time dimension(s) via data cube functions.

The further a data cube implementation departs from the ideal data cube with its spatially dense characteristics, the closer the data cube aligns with a general database. There is another element that influences that blurry line between a data cube and a general database. User experience is to a good extent determined by the knowledge the user has about the cube, and the functions offered by

the data cube to access and process the data. Combining these two brings in another aspect that has not been discussed in detail yet: The role of metadata. Integration and processing of data in multiple steps leads to new products after each step. With the right metadata, users can understand what constitutes each of these products in detail. Thus, a data cube represents a specific, documented product within a data integration and/or processing workflow. The data cube is constituted by a database with access and processing functions that is documented with metadata to sufficiently understand the offered data for further processing, analysis, or decision making. Depending on their position within the value-adding workflow, data cubes may offer raw data as delivered by sensors or models, analysis ready data (ARD), or decision ready information (DRI).

It is the metadata elements and functions that primarily differentiate the data cube definition taken here over other definitions that define a cube from the computer science perspective. For the user, it matters what functions are offered by a data cube instance. A user needs to understand what questions can be asked to access data that fulfills specific filter criteria, how to visualize (sub-) sets of data, or how to execute analytical functions and other processes on the data cube. If supported, the user needs to understand how to add additional processes to the data cube so that they can be executed directly on the data cube and do not require previous download of data.

All other characteristics, such as spatial and temporal details (e.g., being dense or sparse, overlapping or perfectly aligned, area- or point-based), and property details (scales of measurements, handling of incomplete data, interpolation methods, error values, etc.) are provided as cube metadata. Metadata can provide different levels of detail. In this context, it needs to be emphasized that many observations include simplifications or other decisions that influence the properties of the observations without being described as metadata or otherwise easily noticed. As examples, the individual pixels of camera sensors are often read out sequentially, or a push-broom sensor on a satellite cannot keep time still during a full swing. Regardless of the small temporal differences, a single observation time value is usually assigned in both cases.

The data cube definition provided herein does not define any characteristics of the physical storage model of the data on disk or in memory. Being fully independent of the selected data storage model, both on-demand ad-hoc created as well as long term physical read-only memory is supported. For the consumer of the data cube, it is usually irrelevant to know if the data is stored in a relational database, a cloud-based object store, or a file server. These aspects become more important in complex scenarios where for example data from several cubes needs to be fused or different security models are enforced. The situation is different for ad-hoc created data cubes. Given that these are usually produced by processes that use some other data and possibly specific parameterization, reproducibility of results might be affected due to changing content of the cube.

Chapter 2. Do we need an Abstract Model for Data Cubes?

One discussion thread circled around mathematical foundations and formal abstractions of data cubes. Though there is common agreement that a mathematical foundation would eventually lead to enhanced interoperability, there was no majority among the workshop participants for such an approach. This further underpins the basic stance towards a user-centric, flexible design for data cubes being favored over fundamentally more solid but inflexible and restrictive approaches. Instead, interoperability is ensured by a unique method to describe various cube designs, leaving more levels of freedom and better acknowledging the fact that flexibility is required to adapt to the variety of user requirements and usage scenarios.

Chapter 3. User's Perspective

In [his presentation](#), Amruth Kiran from the Indian Institute for Human Settlements described experiences and lessons learned with the Indian data cube to analyse Earth observation data together with census and sample surveys data. Using the Open Data Cube framework, they built a multi-sensor data cube with support for time series analysis. The data cube serves as a baseline for statistical analysis and machine learning model development to understand land cover changes, population development and other indicators over time at varying spatial and temporal resolutions for the whole country. The challenge, he noticed, is not so much with the setup of the data cube, but with the integration of several cubes that provide different data sets. As soon as these data cubes are based on different underlying software, APIs and additional technologies such as STAC (Spatio-Temporal Access Catalog) need to be investigated to facilitate the integration.

Gregory Giuliani, University of Geneva, presented the [Swiss Data Cube Platform as a Service](#). Serving mostly Landsat and Sentinel data, the Swiss data cube uses the Open Data Cube framework to index the data and serves it to a series of clients such as Jupyter Notebooks, Web Services, and Web APIs. Gregory emphasized the importance of libraries being served together with the actual data cube (i.e., to support the functions described in the data cube definition section above). These libraries that in particular contain algorithms that work on the cube data are an essential part of user experience for a variety of users. For the future, Swiss Data Cube 2.0 plans to use COG (Cloud Optimized GeoTiff) for enhanced user experience with support for multiple CRS (Coordinate Reference Systems) and spatial resolutions. There is high confidence that the combination of STAC and OGC APIs (and corresponding ISO standards) will further address outstanding interoperability issues. The key challenge currently remaining is the “application-to-the-data” paradigm and “distributed application” paradigm, where new applications can be added to data cube platforms and different data cubes can be fused to produce analytical results collaboratively. Understanding the efforts and the complexity to produce data cubes, the extra value of federated cube analytics is certainly worth the investment into a common API and data cube description approach. Good experiences have been made with a metadata site that describes the various products and services offered by the data cube platform.

Jimena Juárez, National Institute of Statistics and Geography in Mexico, [shared experiences](#) with the Mexican Geospatial Data Cube. The cube is currently explored in three thematic areas: Urban growth, vegetation and deforestation, and water availability.

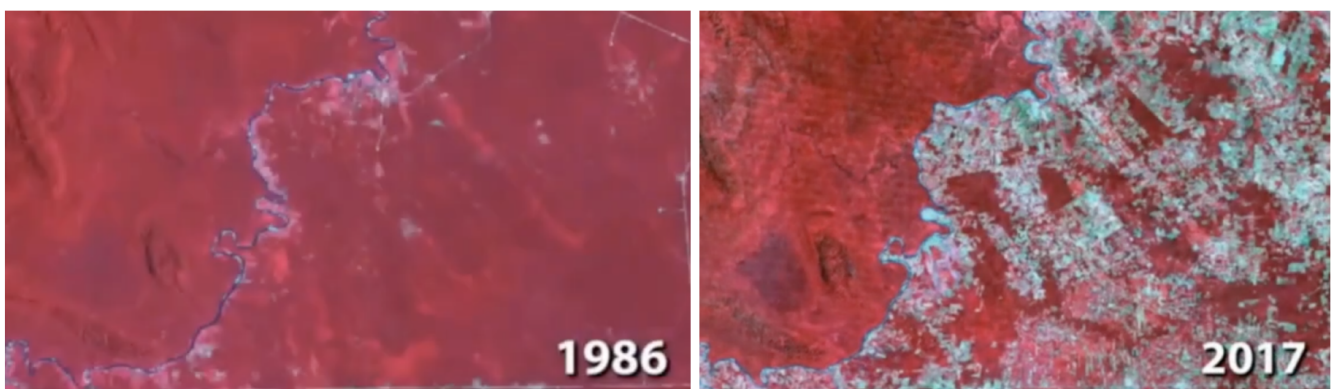


Figure 2. Change in vegetation over time in Mexico. The border line is a river that separates the protected area West of the river from the deforested area in the East. The analysis shows the effectiveness of land protection policies and procedures.

Similar to other presentations, Jimena emphasized the integration of data across data cubes. Only the federation of cubes allows to derive maximum value out of the data stored in each individual data cube. Common APIs to these cubes would even simplify the use of sub-cubes that can be easily instantiated in the form of software containers. Sub-cubes bear great potential for multi-cube analysis if data transport constraints prevent integration at the full cube level. Similarly, sub-cubes simplify the process of making data subsets available to different audiences.

Nataliia Kussul and Andrii Shelestov, both with the Space Research Institute at the National Academy of Science in Ukraine, [presented the Ukrainian Open Data Cube](#). The data cube is particularly useful in exploring scientific challenges in the agricultural and environmental domain. With the organization of data in data cubes, computationally expensive processes such as crop type identification and land cover classification and change detection over time based on machine learning become manageable. Based on the Open Data Cube framework, the Ukrainian data cube makes use of parallelization mechanisms to analyze large areas in separate chunks of data.

Chapter 4. Data Cube Provider Perspective

Alex Leith from Geoscience Australia represented the Open Data Cube (ODC) Steering Council. In his [Open Data Cube presentation](#), he introduced a framework that builds the foundation for many data cube instances that are currently in use. Key points he made are reflected in the cube definition, including his emphasis on metadata about data cubes.

Baudouin Raoult from the European Centre for Medium-Range Weather Forecasts (ECMWF) presented [the Hypercube](#), an impressive multi-cube setup with more than 300PB of data, serving, shared between the numerical weather prediction and observation cube (MARS) and the climate data store, up to 300TB/day to thousands of users. Baudouin emphasized the importance of API support for continuous and categorical dimensions. He further stressed the fact that we need to go beyond cubes that are generated from satellite imagery. As an example, he described typical use cases from meteorology, where three different types of time are usually provided (i.e., forecast time, lead time, hindcast time), and the fact that cubes usually use parameters, levels, and timesteps as leading dimensions.

Edzer Pebesma from Münster University Germany [emphasized the importance of creating reproducible Earth observation science](#) with data cubes. With openEO, he introduced an API and processing environment that adopts a user centric view with data cubes generated on the fly based on users' needs. openEO targets interoperability and reusability aspects across different cloud platforms that are all implementing data cubes following their own design principles and concepts.

Peter Baumann with Rasdaman [shared experiences](#) about data cube design and usage in various communities, including the OGC/ISO coverage model, in particular the grid coverage type, INSPIRE, ISO SQL and Multi Dimensional Arrays (MDA), as well as Multidimensional Expressions (MDX) queries in the OLAP cube. He emphasized the existence of well-matured and standardized models, e.g., the OGC Coverage Implementation Schema (CIS) in combination with the Web Coverage Service (WCS) and the importance of operations that are often domain specific. As an example, reprojection as a typical use case in the geo community does not exist in other communities.

Grega Milcinski from Sinergise described how [the Euro Data Cube](#) developed an API with the user in focus to avoid any possible constraints coming from existing standards. Grega favored an approach that optimizes data according to the specific requirements set by the user community. Instead of trying to harmonize cube models and approaches, emphasis should be placed on making all functions available to the user with sufficient parameterization options. Typical functions include stitching scenes, reprojection, scaling, mosaicking, backscatter calibration, orthorectification and others. Availability of tools is another important factor, as the combination of tools and cubes/cube APIs allow for powerful analytics and visualization, though with a single cube type only. In addition to the typical raster data processing, Grega stressed the importance of vector data support for applications such as machine learning, where vector data is used for e.g. labelling.

Gilberto Queiroz and Karine Ferreira from INPE, Brazil, [explained the importance of the various processing steps](#) required to eventually incorporate data into a data cube to satisfy the specific requirements set by machine learning applications. These are particularly complex for the temporal composition, where multiple strategies exist to select the best pixel available for a given period of time. They presented the API for data cube classification of the SITS (Satellite Image Time

Series) R package developed in the Brazil Data Cube project. The SITS API includes functions to access data cubes from distinct sources and to classify them using machine and deep learning methods to produce land use and cover information.

Peter Strobl, JRC, Italy, [emphasizes the fundamental differences between the user's perspective \(how data is stored and organised on disk\) and the user's perspective \(how it is presented to users through APIs\)](#). Both aspects are in fact fairly independent and have their own design criteria. At the same time, underlying differences in data storage, differences in APIs and the combination of both may lead to different query strategies from one cube to another to obtain the exact same data. Reversively, two data cube instances may deliver different results for identical queries due to underlying differences in design and functionality. The issue becomes problematic when users expect consistent behavior and interoperability between several data cubes. As an example, multiple re-sampling steps in processing chains within or across data cubes inevitably lead to differences in results depending on their sequence of execution. Another important aspect of data cubes distinguishing them from random layer stacks should be minimum requirements regarding e.g. reference system (axis), discretization, and topology necessary to establish a 'dimension' in a data cube and clear criteria that allow assessing the 'interoperability' between the dimensions of different data cubes.

Stefano Natali and Simone Mantovani from MEEO [discussed the data provider perspective](#) as implemented in their data cube technology developed in-house. They stressed the fact that some data cube challenges are actually heritage from the type of data discussed here, for example with satellite data not natively made for multi-temporal data analysis. Both further shared positive experiences with data better stored in its original format with as much processing as possible being executed on the fly. This approach even allows making use of multiple data centers simultaneously in an efficient way.

Stephan Meißl together with Stefan Achtsnit, EOX, Austria, [shared their approach implemented in the Euro Data Cube EOxHub Workspace](#). The approach features deferred execution and lazy evaluation, with only structure and metadata being loaded at interaction start with a data cube. Both further discussed additional strategies to enhance performance for data processing in data cubes. In addition to these strategies, Meißl and Achtsnit emphasized the importance of user functions that can be dynamically loaded into the data cube processing environment. This approach, successfully tested in the [OGC Earth Observation Applications Pilot](#), allows to reverse the classical data to processor concept towards applications to the data or data-proximate computing. Executing applications close to the physical location of the data allows processing even big data in reasonable time.

Chapter 5. Discussion

One key result of the workshop is the new definition of data cube. This new definition goes beyond classical computer science, but sees data cubes as views on data paired with processing functions that can be executed on these data. As such, it condenses the [six faces model as developed by Strobl et al](#), and puts more emphasis on the data cube customer perspective. One aspect that was discussed heavily is the value of abstract models. There was no doubt that a data cube standardization approach with a conceptual model, logical model, and derived physical models would be a valuable contribution to the data cube discussion. Just, is it efficient? What value would it have for the data cube user community these days, where underlying infrastructure and systems change every year?

Interoperability, because of the cloud, is changing. Standards will continue to be important. But it is best practices and recipes that become even more important. It is the sharing of code in combination with open source and the recipes, experiences, and best practices that make the difference. In particular, what works at scale? The traditional approaches to interoperability fade. In a world where you can generate a replica of a database in the cloud within minutes and thus produce a fully functional copy of the snapshot you just worked with, moving XML or JSON or any other code around becomes less important. The success of technologies such as Cloud Optimized GeoTIFF (COG) should be used as sign posts. Sure, technologies such as AWS S3 provide an API, but essentially, all it is is HTTP RANGE GET operations. It is the recipe that matters now.

STAC, the Spatio-Temporal Asset Catalog, is basically a JSON-based manifest index that allows us to understand what is in an object. This object can be a data exchange bucket.

Many data cubes are the result of a long series of processing steps that had been applied to the data. Complex workflows and toolchains have been executed before data was incorporated into the cube. What are the best practices that make us understand what these workflows and toolchains are, how to describe them to ensure reproducible processes and knowledge derivation processes? How do you go from the sensor (or model) data to the view you offer as a data cube so that you can expose data at scale in a repeatable way?

The workshop analysed the current data cube and processing landscape. Simplified, we can observe three different approaches. First, Google Earth Engine (GEE) represents a platform with software; or, more explicit, a computing environment with data and software to process, analyze, and visualize earth observation data. Microsoft with its [Planetary Computer](#) is moving into the same direction. Second, Amazon Web Services (AWS) represents a platform without software. Third, there is software without a platform (openEO and Open Data Cube, ODC). Assuming that these different approaches will coexist for some foreseeable time in future, two possible paths to more efficient and powerful data processing can be outlined.

Taking Google Earth Engine as a starting point, we can understand what it offers; and what we are missing. We can then stepwise add these missing elements. Alternatively, or probably complementary, we could go through the ingredients we believe need to be supported for a sustainable solution, and evaluate each component separately. After reducing the list by removing unnecessary elements, we end up with the Geo Data Cube solution that we need.

There is a path towards new findings, new insights, and new discoveries. With different

approaches coexisting, it boils down to full understanding of the available data (or better, resources, including their complete history), and services to operate on these data. Any cube is a combination of these two. Instead of trying to find the ultimate definition and robust model, it is important to stepwise integrate what is currently existing, i.e., openEO, Open Data Cube, Euro Data Cube, Google Earth Engine, Microsoft Planetary Computer, AWS platforms and other platforms and approaches via well defined APIs. At the same time, new approaches need to be expanded that allow the deployment of new applications into the various processing environments. Important contributions are currently being made by [OGC Testbed-17](#), which continues the work of the [Earth Observation Application Pilot](#) that brought applications-to-the-data in a standards-based way. Standardization is still important, as every step towards more homogeneous APIs and resource definitions allows more efficient and error-free data processing and thus leads to better and faster results. The applications-to-the-data architecture is embedded in the [Earth Observation Exploitation Platform Common Architecture](#), which contributes to an open network of resources and greater interoperability between Earth observation platforms.

In summary, data cubes combine data and services. They enable easier access to data and more efficient exploitation of the rapidly growing body of EO and other data. As a path forward, it is important to stepwise integrate available components, while continuing the development of homogeneous approaches to boost interoperability and accelerate the path to new insights as a base for better decision making. Both paths should co-exist, not one replacing the other.

Chapter 6. Commercial Perspective

The way forward needs to further explore the different operational and commercial models for data cubes. Data cubes can be offered as Data-as-a-Service with only simple data access functionality, or as Platform-as-a-Service, which provides users a cloud environment in which they can develop, manage, and deliver applications. Data cube operators can further choose from various Infrastructure-as-a-Service models.

Consumers need to fully understand what costs are involved with data being available as “Analysis Ready Data” (ARD), “Analysis Ready Cloud Optimized” (ARCO), or “Decision Ready Information” (DRI). Is all data pre-computed and can be directly loaded? Is it computed on the fly and causing additional processing costs? How to handle on-the-fly processing costs for repeatedly executed processes? How to build a user-specific data cube based on existing data most cost-efficiently?

Chapter 7. Next Activities

The research on Data Cubes has been continued in the [OGC Innovation initiative Testbed-17](#). The [Data Cube task](#) moved on from its original planning and already shows very impressive first results. The final report is expected to be released to the public shortly after the end of Testbed-17 in late December or early January 2022.

This report has been released as [OGC document 21-067](#) available as PDF document [here](#).

Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2021-10-06	1.0	Ingo Simonis	all	initial version