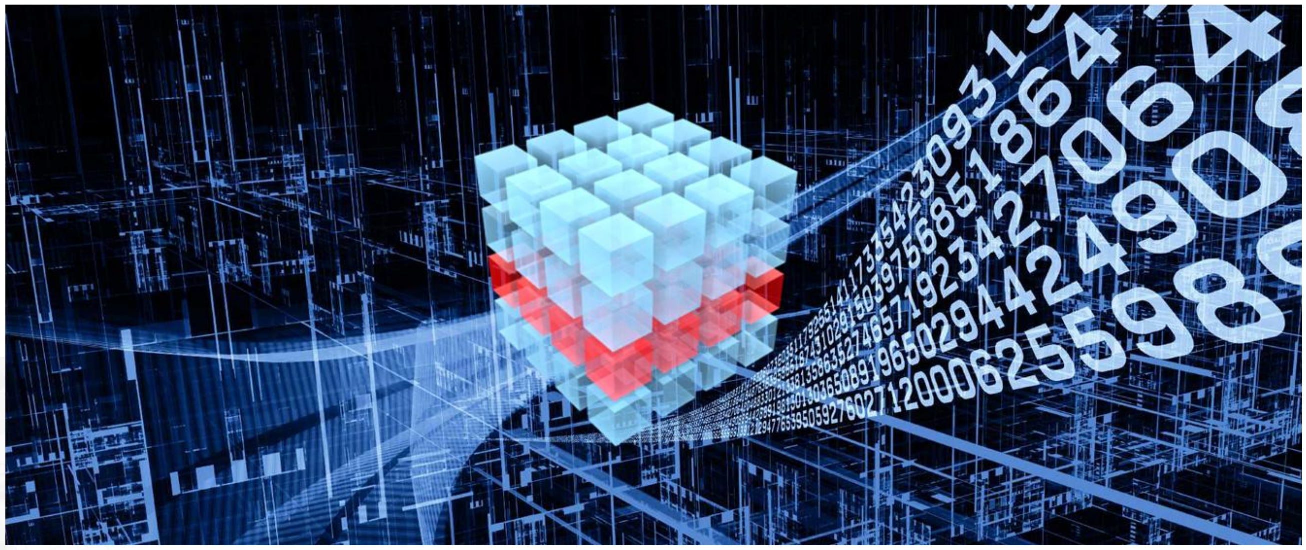

The Datacube Manifesto

Recently, the term datacube is receiving increasing attention as it has the potential of greatly simplifying “Big Earth Data” services for users by providing massive spatio-temporal data in an analysis-ready way. However, there is considerable confusion about the data and service model of such datacubes.

With this Manifesto we provide a concise, crisp definition of datacubes, based on our two decades of experience in datacube modeling, query languages, architectures, distributed processing, standards development, and active deployment of Petascale datacube services at some of the largest data centers worldwide.

Definition. A **datacube** is a massive multi-dimensional array, also called “raster data” or “gridded data”; “massive” entails that we talk about sizes significantly beyond the main memory resources of the server hardware. Data values, all of the same data type, sit at grid points as defined by the *d* axes of the *d*-dimensional datacube. Coordinates along these axes allow addressing data values unambiguously.

A *d*-dimensional **grid** is characterized by the fact that each inner grid point has exactly two neighbors along each direction; border grid points have just one (see Fig. 1). Point clouds, e.g., are not grids.

The Six Principles of Datacube Services

Requirement 1: Datacubes shall support gridded data of at least one through four spatial, temporal, or other dimensions.

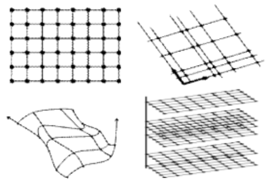

Fig. 1: sample grids [OGC CIS 1.1]

Although the name “datacube” suggests a 3-D model, the datacube model is not restricted to that. The spectrum includes 1-D sensor timeseries, 2-D imagery, 3-D x/y/t image timeseries and x/y/z subsurface voxel data, 4-D x/y/z/t climate and ocean datacubes, and even 5-D atmospheric data with two time dimensions, as mandated by climate modelers. Besides these use cases support for lower dimensions is already a must because slices extracted from, say, a 3-D x/y/t image timeseries can be 2-D x/y images or 1-D timeseries. Typically, one datacube is constructed from a large number of files, such as all scenes from one satellite instrument. By ornamenting these axes with domain-specific metadata, spatio-temporal and other semantics can be expressed. The grid can be regular, irregular, or even mixed (such as regular x/y and irregular t), cf. Fig.1.

Ideally, the datacube is based on the OGC Coverage Implementation Schema *(CIS) standard, adopted also by EU INSPIRE and under adoption by ISO.*

Requirement 2: Datacubes shall treat all axes alike, irrespective of an axis having a spatial, temporal, or other semantics.

In the past, most standards as well as implementations had dedicated methods for spatial extraction, and diff-

erent ones for temporal extraction. Datacubes, on the contrary, offer the single concept of an axis *which, through ornamenting metadata, becomes spatial, temporal, or anything else. After such a unifying step, trimming and slicing (see Fig. 2) can follow the same syntax along all axes. This is not withstanding different units of measure – Latitude might be measured in degrees like 42°30', time in days like 2017-06-06, height in flight levels like FL100.*

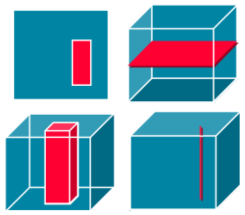
Fig. 2: datacube trimming (left) and slicing (right) [OGC WCS 2]

Requirement 3: Datacubes shall allow efficient trimming and slicing along any number of axes from a datacube in a single request.

Analysis-ready means: ready for analysis along all dimensions. *Inevitably, the first analysis task is spotting the relevant areas in space and time; extracting axis by axis, maybe even transporting intermediate results to the client, leads to excessive copying and transport of data "too big to transport".*

Ideally, the service interface conforms with OGC Web Coverage Service *(WCS) 2 standard – adopted also by EU INSPIRE and foreseen by ISO – as a modular, open, interoperable service interface supporting many protocols (such as GET/KVP and SOAP) and formats (from GeoTIFF over NetCDF to GRIB2).*

Requirement 4: Datacubes shall convey similar extraction performance along any datacube axis.

Obviously, all datacube trimming and slicing should come at good performance – why is this worth mentioning? Because classic storage is optimized for horizontal access, penalizing temporal and vertical extraction. A stack of GeoTIFF scenes representing a timeseries is not yet a datacube, even when offered through a datacube API – it will be fast in extracting image slices, but convey disastrous performance on timeseries analysis.

Requirement 5: Datacubes shall allow adaptive partitioning, invisible to the user when performing access and analysis.

It is common sense that partitioning is a must for large datacubes. It does not add functionality, but speeds up performance tremendously if done right. Notably, there is not one single "good" partitioning; it depends on the individual mix of client access patterns, such as purely horizontal access, purely vertical access, pure timeseries access, or any combination (and variation, over time) of patterns. A powerful datacube tool will offer partitioning as a tuning option (Fig. 3).

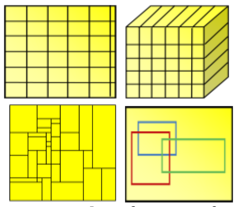
Fig. 3: Classification of partitioning schemes [Furtado, 1999]

As such, partitioning should remain invisible – compare it to a database index. Having to consider partitions makes analysis queries terribly complicated, in particular on more involved operations like convolution kernels and differential analysis. Offering single data objects analysis-ready is a key concept of datacubes.

Requirement 6: Datacubes shall support a language allowing clients to submit simple as well as composite extraction, processing, filtering, and fusion tasks in an ad-hoc fashion.

Analysis-ready data is one side of the coin, analysis functionality the other; "any query, any time, on any size" is a must. Shipping procedural code to the cube is undesirable from several perspectives, including user friendliness, parallelizability, and security. Therefore, some high-level datacube language must be supported where users describe what they want to get, not the detailed algorithm.

Ideally, the service supports the OGC Web Coverage Processing Service *(WCPS) standard as an open, interoperable geo datacube language also adopted by EU INSPIRE.*

Conclusion. Datacube services built along the six requirements of this Manifesto have proven to be significantly more user-friendly, faster, and scalable than typical classical services – massive gridded spatio-temporal data become analysis-ready (Fig. 4). The OGC datacube standards, CIS and WCS/WCPS, are embraced by open-source and proprietary implementers, coming with compliance tests enabling interoperability down to the level of single pixels. Availability of datacube standards and tools is heralding a new era of service quality and, ultimately, better data insights.

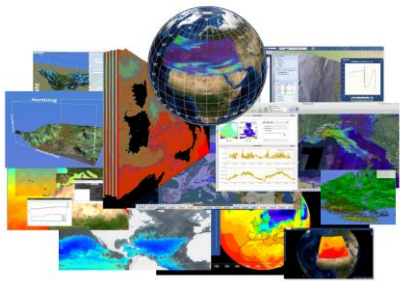
Fig. 4: Kaleidoscope of WCS/WCPS based datacube portals [EarthServer, 2017]