Data

OpenEarth.nl - OpenEarth.eu:	**Home**	**Data**	**Products**	**Tools&tutorials**	**Forum**	**Search**	**Join@LinkedIn**

The OpenEarth philosophy aims to collect and disseminate environmental and lab datasets in a project-superseding manner rather than on a project-by-project basis. We believe that science and engineering have become so **data-intensive** that data management is beyond the capabilities of individual researchers. Data management needs to migrate from artisanal methods to **21st century technology**. This implies data management needs to team up with IT-professionals, and vice versa: the 4th paradigm.

All data online

Sustainable solutions to manage data are **web-based** and involve **communities**. OpenEarth aims to be a 4th paradigm workflow solution to let scientist and engineers collaborate in communities over the web. The need for teaming up of science and IT is clearly illustrated in a recent Nature article. Anno 2020 the preferred solutions are in the **Cloud** (Signell & Pothina, 2019).

Community

At the bottom of this wiki page you can see a movie of the community activity in our raw data repository, a tool we got from the IT-world. Such communities should not only deal with data, but deal with numerical models and analysis tools as well. Data cannot be treated separately from the rest of science. Therefore OpenEarth aims to be an **integral workflow for data, models and tools**. For hosting such a the workflow we advocate collaboration with professional data centres such as 3TU datacentre, DANS and Pangea. Some data centers are member of DataCite, and can give you a DOI for published data under conditions, enabling anyone to cite your web-based data.

Standards

To be an effective and sustainable 4th paradigm solution, OpenEarth has identified the most promising international standards for exchange of data over the web. We have chosen standards to be both *de iure* standards and *de facto* standards, meaning they have been approved by an international, recognized standards body: OGC or ISO and that they are used by a community of practice. The latter implies that the standard is not just a paper blue-print standards, but has been implemented. The most important criterion is that are these standards a web-based, meaning that they rely on a division of work between a server and a client (your cluster, desktop, laptop, AR/VR set, touchtable or smart phone).

Leveling up datasets

Depending on how demanding and experienced you are as a user (scientists tend to known things better than you, you are on equal terms with professionals and the general public is often less experienced). This means that some kinds of users require heavy servers (general public) whereas scientist are already happen when you dump the raw data on them (under version control). We identified 5 data processing levels to serve all potential users: raw data, standards data, tailored data, visualized data and catalogs of data (the number was is also used by Tim Berners-Lee in his 5stardata) The most difficult transition is that from raw to standard data. In the database world this step is known as ETL (Extract, Transform, Load). When proper open standards + open implementations have been chosen and vendor-lock-in to commercial packages has been avoided, all subsequent levels can simply be obtained by installing and configuring the right off-the-shelf open-source software.

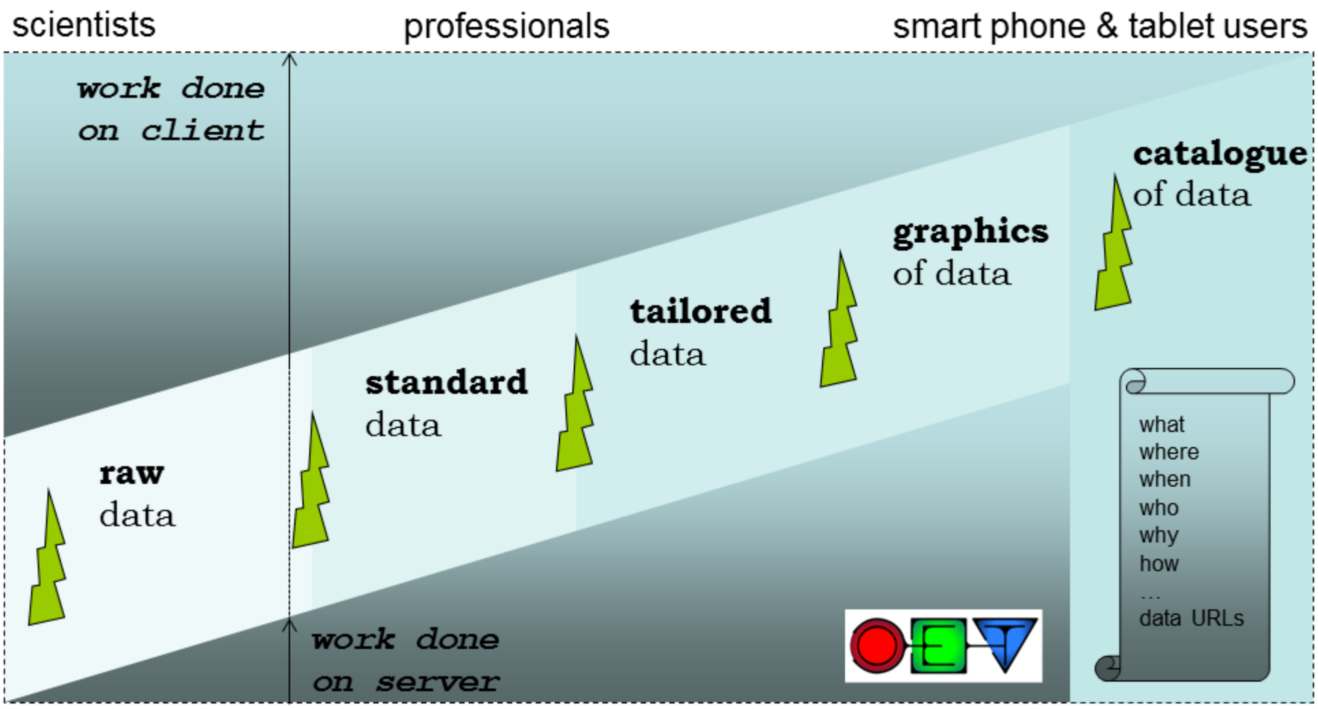

Figure: the client-server division of work for a range of users.

Raw data: Versioning data

These chosen OpenEarth standards are shown in the scheme below, they come from different realms. We aim to work with all of these standards, but currently only use the bold ones on a daily basis. These include the subversion tool to store not only the **raw data**, but also the processing software (scripts, settings) under **version control** using the web 2.0 Wikipedia approach: everyone can sign up for write access. This allows us to naturally attribute versions to data, an aspect that lacks in most of today's data management solutions (known as provenance or lineage) with remote sensing as a noteworthy exception to this rule [1] . Next to SubVersion also GIT-LFS can be used for scripts and configurations. Anno 2020 the trend is to use commercial cloud storage services (Amazon, S3, Azure Blob storage, Google Cloud) with new file types that allow remote subsetting, like zarr or cloud optimized geotiff.

Standard data: grids versus features

For **standardized data** we use two approaches: one for voluminous gridded data and one for relational data. For grids the netCDF format (NASA and OGC standard) with the CF vocabularies and EPSG codes [2] , [3] forms a very powerful data stack as described in an OceanObs'09 paper. We place the netCDF files on a OPeNDAP server for dissemination of TBs of netCDF data over the web. OPeNDAP is available in many user software applications. It is for instance built-in for MATLAB since 2012, and it is optionally available for the R, python, ArcGis and many other netCDF programs [4] , [5] . On top of an OPeNDAP servet you can place various servlets that tailor the data to specific needs, e.g. remap to a different grids in a different projection (WCS) or make an image of a subset of the grid (WMS). The OGC WPS and OGC WCPS standards also allows you to perform remote calculations on the data. Increasingly server side micro services are used that simply serve geojsons of aggregated data via a REST API.

Standard data: relational database

For ecological data, which have an overwhelming amount of meta-data, we use a plain-vanilla Relational DataBase Management System (RDBMS). We chose the powerful, open source PostgreSQL implementation with PostGIS spatio-temporal add-on. We are working on adopting dedicated spatio-temporal standard as well. These standards allow for live server-side processing on the data to meet the demands of the user. They deliver **tailored data**. The OGC consortium is *the* international body for specifications of these standards. The EU INSPIRE directive prescribes these standards. For typical GIS data (flat, 2D or 2.5D) we already work with postgis, geoserver and geonetwork. However, these so-called `WxS` protocols still lack implementation in operational software for many specific demands of time-dependent, 3D, curvi-linear data products in our field. We do not develop `WxS` software ourselves, but just wait for the open source implementations, most under OSGeo umbrella, to cover the demands of our our field. By far the most promising `WxS` client and server implementation we indentified is ADAGUC by the Royal Dutch met office KNMI. ADAGUC not only implemented the WCS standard to request data tailoring over the web very fast, but also the WMS standard to request imagery. For exchange of **graphics of data**, we chose to start working with the KML standards, the standard behind Google Earth that was also adopted as standard by OGC, but we will adopt WMS as well.

Tailored data, graphics of data and catalogues

Once data has been upgraded from raw data to standard data, off-the-shelf (OTS) server packages can be used to provide additional service on the data. Merging data form different data sets, reprojecting data into different coordinate systems or interpolating data can be done using the OGC WFS standard for feature (vector) data using geoserver. The OGC WCS standard for grid data can be served using the THREDD ADAGUC WCS service. More complex manipulations cna be performed with the OGC WPS standard (Web Processing Service), using for instance pyWPS. For more convenient use, the data can also be turned into geo-referended graphics using again geoserver for feature(vector) data and THREDD ADAGUC WMS service. Many OTS webGIS viewer exist to visualize such graphics on map, for instance HERON, ADAGUC viewer, Google maps and Google Earth. All the tailoring and graphics services can be cataloged using the OGC CSW standard using geonetwork or CKAN. OpenEarth has bundled all this software into a seamless software stack for server installation. An OTS software component has been chosen for each of the 5 data levels, with separate components for feature and grid data if necessary. Our aim is to make installation this stack a 10 min effort for commercial cloud hosting platform such as Amazon Web Services, Microsoft Azure or Google Cloud Platform. But also R&D clouds should be part of the scope of course, e.g. EOSC.

Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'
Unknown macro: 'lozenge'

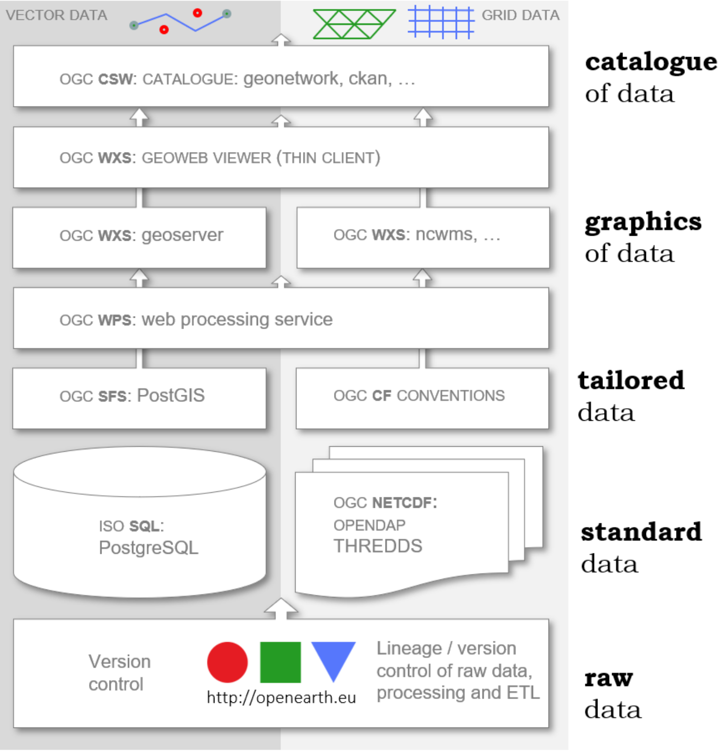

Figure: the open standards (ISO or OGC) that OpenEarth chose as overall solution, and the open source implementations (components) of the open ststack, using OTS open source components.

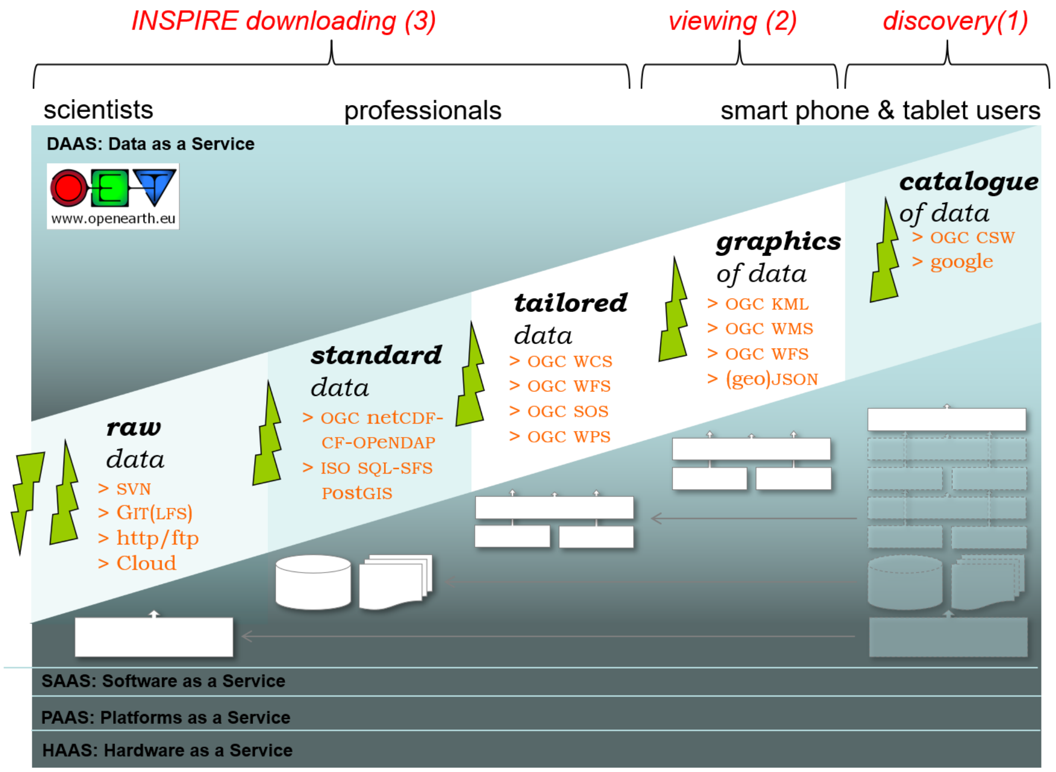

Figure: the open standards (ISO or OGC) that OpenEarth chose as overall solution, and the open source implementations (components) of the open standards that OpenEarth chose as overall solution for a server stack.

Extract Transform Load

The data collection procedure and the relation between those standards is explained in the OpenEarth Data Standards document, developed in the framework of the EU FP7 Project MICORE and Building with Nature. The basis the 3-step ETL procedure well-known in the database world. ETL describes the process to Extract data from somewhere, Transform it to the strict database datamodel requirements, and Load it into the database. We extended ETL with one crucial extra step: **provide** the data again to users via the web. We believe that any effective data management solution should include users at the start of the ETL process *and* and the end. Loading data into the database and using data from the database should be possible from the work environment of the user. In the sketch above we explicitly included client and server to highlight the paramount importance of easy and immediate web-based Provide mechanisms of the data, that are not covered by ETL.

Step by step

ETL contains the followings steps:

- data is not just numbers and meta-information, but consists of raw data produced by the measuring equipment (e.g. volts) + processing scripts.
- raw data + scripts should be stored in the OpenEarthRawData repository enabling version control
- raw data should then be enriched with metadata and processed into useful data products (netCDF, PostgreSQL table) using transformation scripts that should also be put under version control in a repository
- resulting data products should conform to the best open source semantic standards available, e.g. CF, WoRMS
- data products should be made available easily via webbased interfaces (OPeNDAP, ODBC ore dedicated DB-APIs, WxS) but also with automatable procedures for widely-used data processing languages such as matlab, IDL, python, fortran, C and java (OpenEarth Tools).
- data products are primarily meant for dissemination, raw data and scripts are primarily meant for archiving.
- meta-data should be gathered and inserted into a central catalogue.

Other initiatives

Numerous other datasets have been or are being uploaded continually in the MICORE and Building with Nature research programs. And OpenEarth is not the only initiative to share and disseminate government-paid Earth science data freely on the web using open standards. We made an inventory of related initiatives. Our aim is to spread the use of the open standards and make them stick in our everyday work.