EUROPEAN COMMISSION

Community research

Climate Information Portal

CLIPC DELIVERABLE (D -N°: 4.3)
Visualization Manual

File name: {CLIPC_Deliverable-4.3-final.pdf}

Dissemination level: PU (public)

Author(s): *Maarten Plieger, Ernst de Vreede, Wim Som de Cerff*

Reviewer(s): *Andrej Mihajlovski, KNMI Ag Stevens, STFC*

Reporting period: *01/12/2013 – 31/05/2015*

Release date for review: 28-10-2016

Final date of issue: xx-11-2016

Revision table			
Version	**Date**	**Name**	**Comments**
1	October	Draft	Circulated for discussion
2	October	Review	For internal review
3	November	Final copy	Final version

Abstract

The Visualization Manual describes the ADAGUC Web Map Service (WMS) server used in the CLIPC project. The WMS server is used for all map visualizations and plots in the CLIPC portal. In this document the ADAGUC WMS, the specific CLIPC functions implemented and the usage of the ADAGUC WMS server are explained. Note that for specific configurations and examples is referred to the public ADAGUC Wiki, as these are dynamic documents.

Project co-funded by the European Commission's Seventh Framework Programme (FP7; 2007-2013) under the grant agreement n°607418

Table of contents

Executive Summary

Visualization of climate indicators is one of the essential components in the CLIPC portal. This visualization needs to be standardized and highly performant, in order to meet the user requirements.

For the CLIPC project the OGC WMS version 1.3.0 standard, implemented in ADAGUC [3], is used for all map visualization, timeseries graphs and histograms. ADAGUC (Atmospheric Data Access for the Geospatial User Community) is an Open Source geographical information system to visualize netCDF, HDF5 and GeoJSON over the web. ADAGUC is used because of its high performance visualization of remote OPeNDAP URLs, enabling fast visualization on large datasets.

For this, data needs to be standardized to enable automated visualization, styling and geographical projections. NetCDF is the required file format. These files need to comply with the CF-conventions (NetCDF-CF). As many datasets from the CLIPC community are large datasets and available in NetCDF-CF, ADAGUC is an excellent choice for implementation.

The software consists of a server side C++ application and a client side JavaScript application. Web clients like GoogleMaps, OpenLayers and Leaflet are supported and can directly use the exposed webservices.

Features of ADAGUC are: fully compliant OGC WMS and WCS, integrated data converters and post processors to support various data conventions, aggregation support.

Features developed in CLIPC context are GeoJSON polygon/vector visualization for ranking, tiling functionality for high resolution datasets, ensemble visualization in the CLIPC portal, histogram visualization in the CLIPC portal and the NetCDF Web Coverage Service backend.

ADAGUC is an Open Source package and can be easily installed. A fully pre-configured Virtual Machine is available and also many online tutorials are available to support developers. As such it has a growing user community, using ADAGUC in both research and operational environments.

Introduction

According the Open Geospatial Consortium WMS is defined as follows:

> *The OpenGIS® Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.* [ref 1.]

For the CLIPC project WMS version 1.3.0 is implemented in ADAGUC. We use ADAGUC for its high performance visualization of remote OPeNDAP URLs, enabling fast visualization on large datasets. For this, data needs to be standardized to enable automated visualization, styling and geographical projections. NetCDF CF is the required format. As many datasets from the CLIPC community are large datasets, ADAGUC is an excellent choice for implementation.

CLIPC Visualization architecture

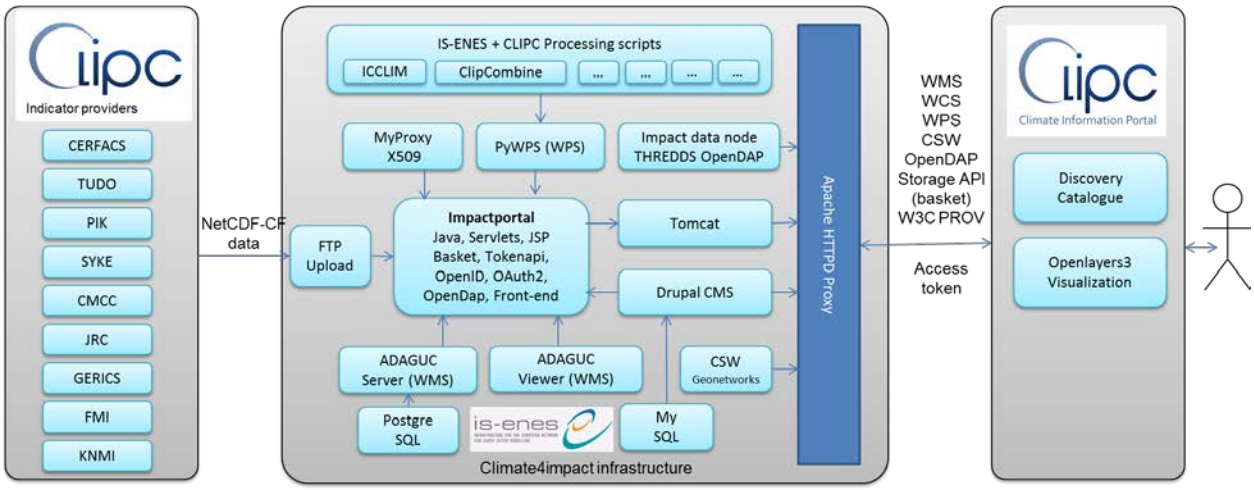

Figure 1 CLIPC architecture

The CLIPC architecture relies on Open Standards to provide well defined interfaces between data, the service backend and the end user portal. The architecture is depicted in Figure 1. On the left side all data providers are mentioned. They deliver the data in NetCDF-CF format. This enables easy integration in the backend, which provides services to the front-end portal. The data and backend services are hosted at KNMI and in the ESGF network; the frontend portal is run by Maris, at a different location.

The datasets uploaded to climate4impact at KNMI are checked for CLIPC DRS conformance. The DRS is a specification describing the metadata required in

CLIPC datasets. When correct, the metadata is automatically extracted and put into a metadata database serving a catalogue service (CSW). The CSW exposes the OPeNDAP URLs to the datasets as part of the metadata.

OPeNDAP, an acronym for "Open-source Project for a Network Data Access Protocol", is a data transport architecture and protocol widely used by earth scientists. The protocol is based on HTTP and the current specification is OPeNDAP 2.0 draft. OPeNDAP includes standards for encapsulating structured data, annotating the data with attributes and adding semantics that describe the data. The protocol is maintained by OPeNDAP.org, a publicly funded non-profit organization that also provides free reference implementations of OPeNDAP servers and clients [https://en.wikipedia.org/wiki/OPeNDAP][7].

The metadata is harvested by the CLIPC portal using the OGC CSW protocol. As the CLIPC portal now knows the OPeNDAP URL to the data, it can directly call the auto-generated WMS server on climate4impact to get an image, WCS to get data or WMS getFeatureInfo to get an (ensemble) timeseries plot.

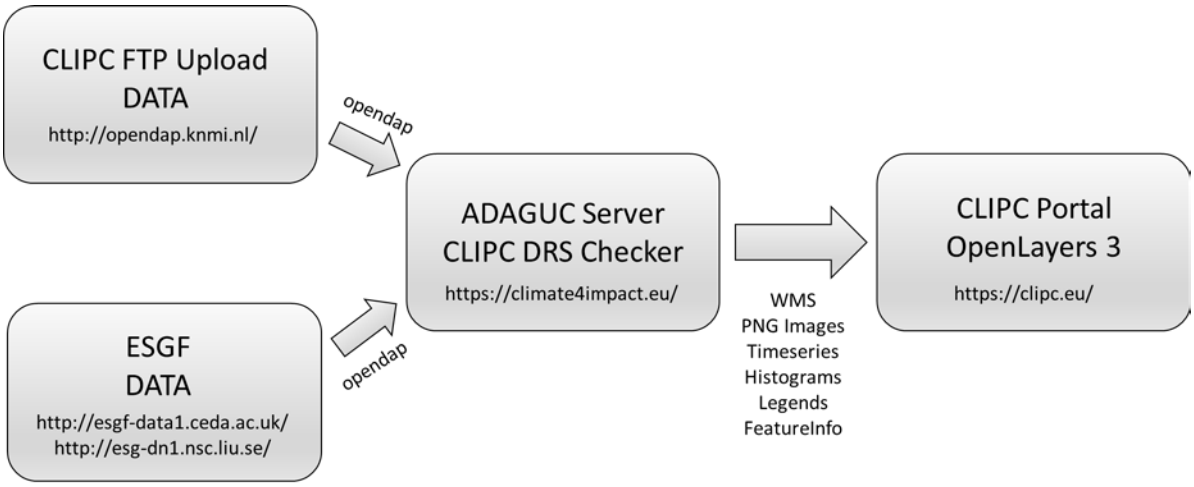

Figure 2 Simplified view from architecture overview: data, backend (climate4impact) and frontend (CLIPC portal)

ADAGUC software

ADAGUC (Atmospheric Data Access for the Geospatial User Community) is an Open Source geographical information system to visualize netCDF, HDF5 and GeoJSON over the web. The software consists of a server side C++ application and a client side JavaScript application. The software provides several features to access and visualize data over the web, it uses the OGC WMS and WCS standards for data dissemination. Web clients like GoogleMaps, OpenLayers and Leaflet are supported and can directly use the exposed webservices.

ADAGUC can visualize remotely published NetCDF files by adding the OPeNDAP resource as parameter to the webservice request. This enables direct

visualization of any OPeNDAP enabled resource over the web. Graphic styling of data is done by checking the variable standard_name and units.

OGC Web Coverage Services (WCS) are available and can be used for data re-projection, subsetting and conversion to other formats. Access to OPeNDAP services is done efficiently; multiple requests are aggregated into one and only the domain of interest is requested. This allows for easy, quick and interactive visualization of OPeNDAP-enabled datasets.

ADAGUC has a number of data converters and data post processors to support various data conventions. Supported file formats are “true color netCDF” for satellite imagery, structured grids, curvilinear grids, satellite swaths, point observations, point timeseries and polygons stored in GeoJSON.

Datasets consisting of several netCDF files can be aggregated into a single dataset and are offered over WMS, WCS and OPeNDAP. ADAGUC can be used as a component for Web Processing Services to subset data and convert GeoJSON to grids.

ADAGUC cannot only be used for visualisation, but it can also be used for data retrieval through the WCS 1.0 interface. The data retrieval can be done by processing components like the WPS-components in the processing suite. These processing components can use ADAGUC’s facilities for reprojecting and cutting out areas of data.

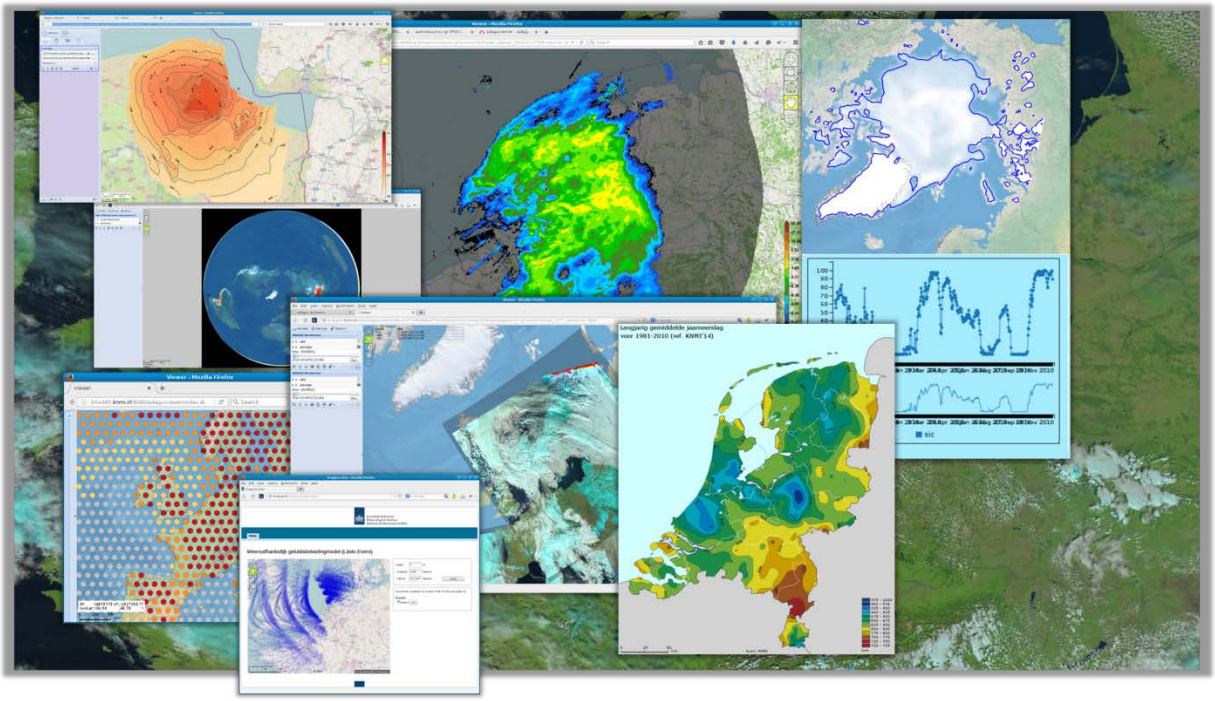

Figure 3 Visualizations created with ADAGUC software

ADAGUC Functionality developed in CLIPC.

In the CLIPC project ADAGUC has been extended to meet the requirements needed for indicator visualization. The following new functionalities have been added during the CLIPC project:

1. GeoJSON polygon/vector visualization for ranking
2. Tiling functionality for high resolution datasets
3. Ensemble visualization in the CLIPC portal
4. Histogram visualization in the CLIPC portal
5. Stippling style for uncertainty information
6. Implementation of extra CF-compliant projections
7. NetCDF Web Coverage Service backend

These functions are described in detail on the following pages.

GeoJSON polygon/vector visualization

Functionality for handling polygon information defined in the GeoJSON format has been added to ADAGUC. This polygon functionality enables ADAGUC to derive geographical grids from (multi)polygons. These grids can then be used for direct visualisation in WMS or as data fetched via WCS by processing tools.
This work enables the following tasks:

- (Partial) Map of Europe where each NUTS[1] area (levels 1,2,3) has a different colour.
- (Partial) Map of Europe where a NetCDF point file containing data for each NUTS area leads to an area on the map coloured according to its value. This provides a way to display data with one value per NUTS area as coloured NUTS areas on a map.
- Grid (for use in processing) where a NetCDF point file with data values for each NUTS area results in a grid where each cell has a value corresponding to the value for the NUTS area the cell belongs to.
- Grid (for use in processing)where each cell has a value defining which NUTS area it belongs to. Such grids can be used to group values in a data grid to generate grids (or tables) of derived parameters like min, max or average for a NUTS area. As a side effect this procedure can produce a set of derived data (min, max, average) for the NUTS regions in the GeoJSON.

[1] The current NUTS 2013 classification is valid from 1 January 2015 and lists 98 regions at NUTS 1, 276 regions at NUTS 2 and 1342 regions at NUTS 3 level. The NUTS classification (Nomenclature of territorial units for statistics) is a hierarchical system for dividing up the economic territory of the EU for the purpose of: the collection, development and harmonisation of European regional statistics and Socio-economic analyses of the regions

This can be useful for ranking NUTS areas by the min, max or average of their values in a grid.

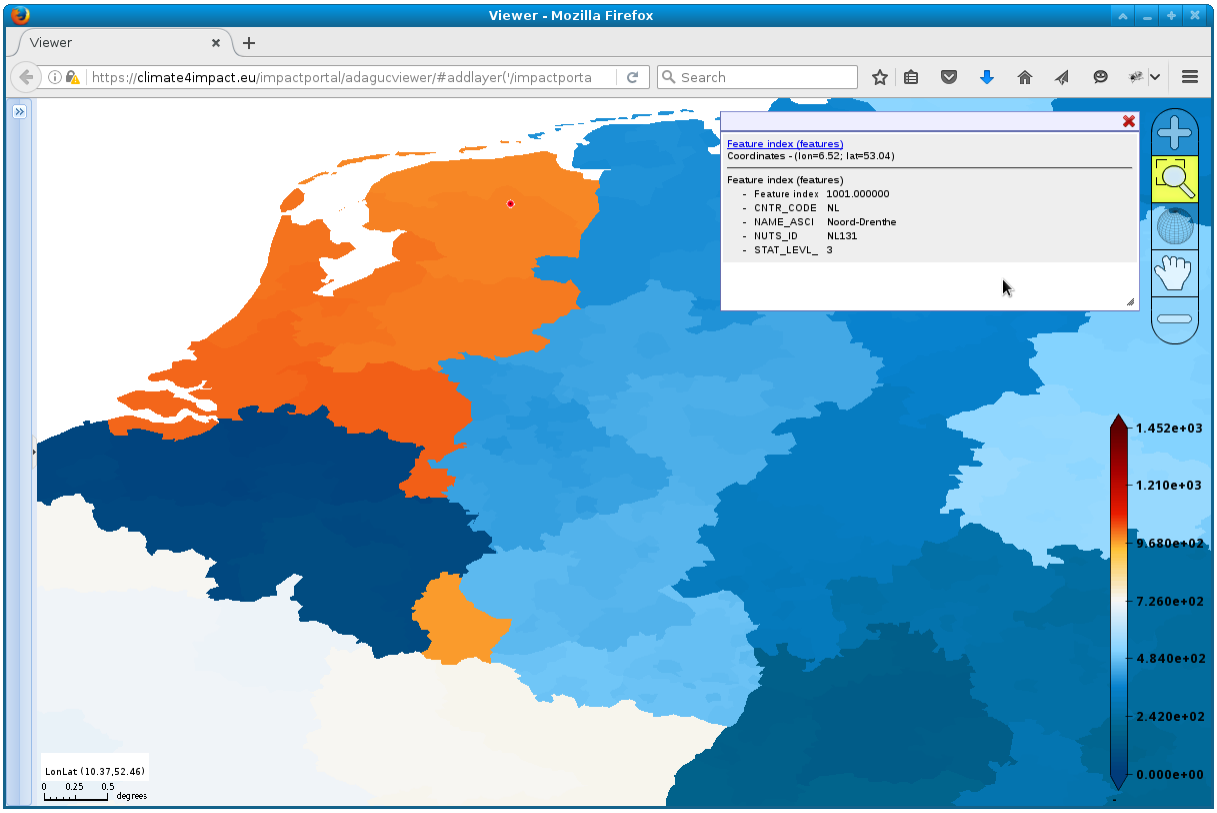

Figure 4 GeoJSON of the NUTS regions visualized in ADAGUC.

Tiling functionality for high resolution datasets

Some datasets in CLIPC are provided in such a high resolution that they cannot be stored in a single file. For example the flooding indicator from PIK is at 25 meter resolution along the coast in Europe. The data is organized in 6000 different NetCDF files, each covering its own area in 600x600 grid cells, which need to be “stitched” together. This can be compared with composing a large panoramic picture from multiple pictures taken along the horizon. The challenge here is to provide interactive visualization at several cartographic projections and zoom levels.

To achieve this, ADAGUC has been extended to allow building image pyramids at several zoom levels on coarser resolutions (see Figure 4). For CLIPC 7 different pyramid levels are used for the flooding indicator, the top level consists of three tiles, the bottom level of 6000 tiles. When zooming to a specific portion of the map, the level with the best matching resolution is used to render the requested image. That means that ADAGUC projects and combines several tiles into a single image suitable for WMS. This type of tiled datasets in ADAGUC behave like any other WMS, e.g. interactive zooming and panning at several cartographic map projections remain possible.

For CLIPC the flooding indicator in the cartographic projection lambert equal area is remapped on request to the Mercator projection used in the CLIPC portal.

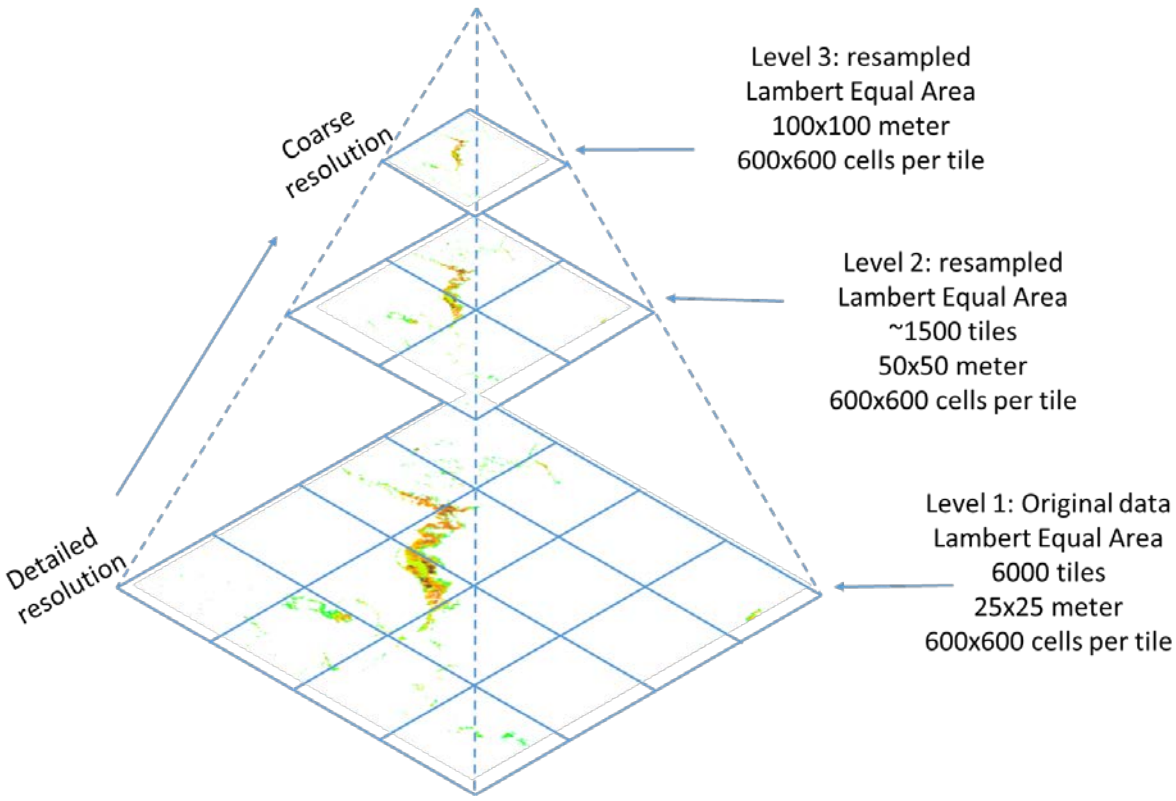

Figure 5: ADAGUC internal tile pyramids for high resolution data. ADAGUC stitches and warps the tiles for the required zoom level into a new image. Note that remapping to other cartographic projections remains possible.

Ensemble visualization in the CLIPC portal

The standard timeseries functionality in ADAGUC is achieved by requesting a time range from the server, using the WMS GetFeatureInfo request. This has been extended to return data for any other dimension(s). In CLIPC datasets are made available which include data for several members. ADAGUC is now able to return data for a specific location, for a range of members and dates.
This timeseries/ensemble can be returned in XML, GML, PNG and JSON. It is possible to use a wild card for any dimension. When using DIM_MEMBER=* in combination with a time range, a list of values for a set of all members is returned for the specified time range. This can be requested in JSON/JSONP, enabling JavaScript clients to use this functionality. To obtain a JSON for several members one has to add the following to the GetFeatureInfo request:

info_format=application/json
dim_member=*
time=1000-01-01T00:00:00Z/3000-01-01T00:00:00Z

This will return a JSON with all possible values within the specified time range for all available members. The JSON can then be visualized with the interactive JavaScript graph library named HighCharts (Figure 4), as done at the CLIPC portal.

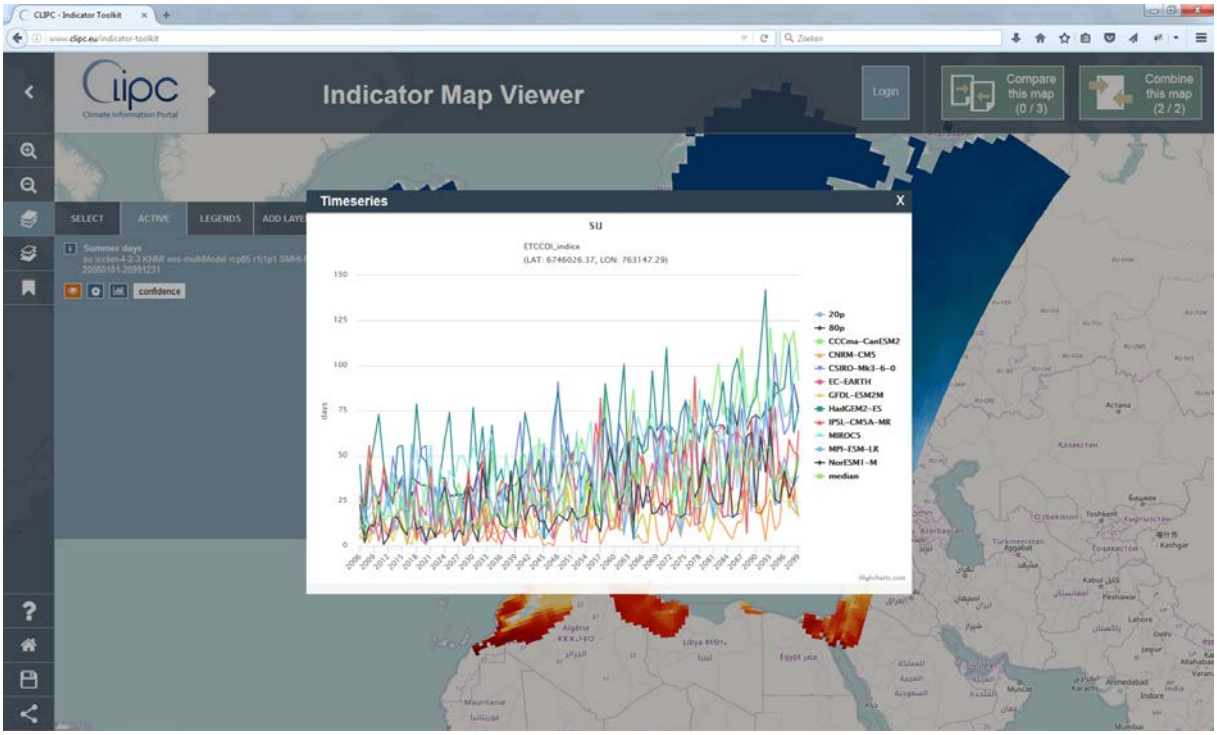

Figure 6 HighCharts graph in the CLIPC portal visualizing the JSON obtained with a GetFeatureInfo call to ADAGUC server.

Histogram visualization in the CLIPC portal

To get a histogram plot, we added the WMS getHistogram function. This is not part of the OGC specification. This extension works similar to the GetMap request, for the same area and dimensions it returns a JSON with histogram information, minimum, maximum and standard deviation. In the CLIPC portal this information is displayed in a graph (Figure 5).

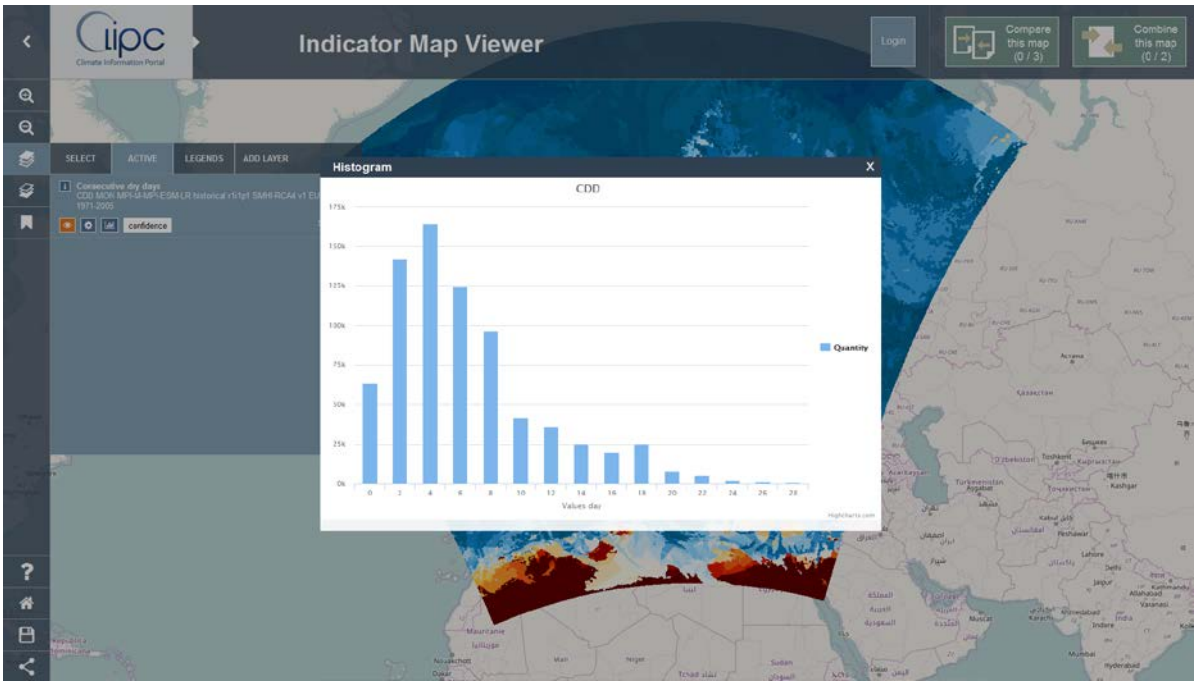

Figure 7 Histogram in CLIPC showing statistics for the specified area, for the indicator Consecutive Dry Days (CDD).

Stippling style for uncertainty information

For the CLIPC climate signal maps, a special visualization method named stippling was implemented in ADAGUC. The stippling method renders transparent equally spaced circles on a map. The values for the circles in the map are sampled from the base grid using nearest neighbour interpolation. The diameter of the circles and the distance of the circles can be configured. The stippling in ADAGUC can be enabled using the following style configuration:

```
<RenderMethod>stippling,nearest</RenderMethod>
<Stippling distancex="22" distancey="22" discradius="7"/>
```

The image in figure 6 displays the CLIPC climate signal maps with the stippling rendermethod. The NetCDF file uses CF status flags to store discrete values with texts, these texts are displayed in the legend accordingly.

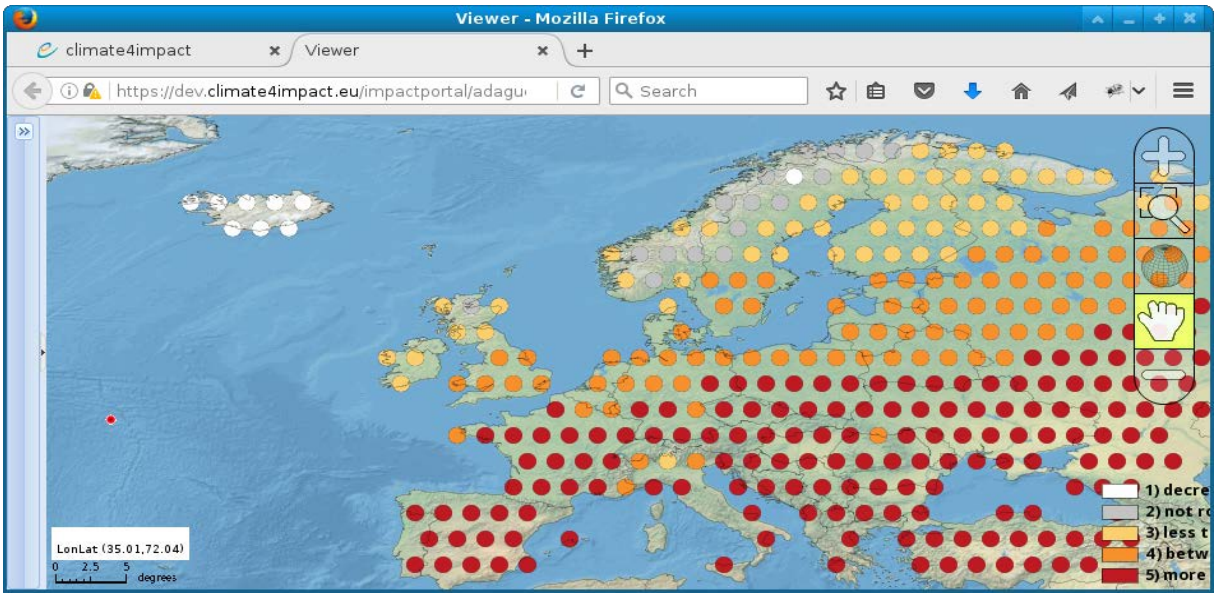

Figure 8 Stippling rendermethod in ADAGUC displaying climate signal maps.

Implementation of extra CF-compliant projections

During the CLIPC project many new datasets were created with several geographical projections. Some of these geographical projections were only specified using the Climate and Forecast conventions. For geographical projections, ADAGUC internally works with the proj library and converts CF projection parameters to proj4 strings. ADAGUC has been extend to recognize more CF projections, the following have been added to the existing ones:

- Mercator
- Lambert azimuthal equal area
- Lambert conformal conic

In the ADAGUC source code these changes are reflected in the CProj4ToCF.cpp file.

ADAGUC NetCDF Web Coverage Service backend

For CLIPC processing, extensive use of Web Coverage Services (WCS) is made. A web coverage service can lift the burden for other tools, as it provides a clean interface to gridded data in all sorts of cartographic projections; including curvilinear data, tiled datasets and the newly supported GeoJSON vector format. Normally, handling these types of data structure can be complex and is prone to errors. WCS solves this by converting data to regular CF compliant grids in any supported cartographic projection. Because WCS is implemented at a single location (in ADAGUC), support for new exotic datasets require only an update of the WCS implementation in ADAGUC. This enables this type of data in all processing services through the WCS interface.

For CLIPC a new NetCDF writer has been implemented in ADAGUC which writes NetCDF files compliant to the climate and forecast conventions using the WCS interface. Other output formats, like GeoTIFF and AAIGRID, are still possible but are being written using the GDAL API.
The NetCDF includes all metadata from the original file and has information about which software (ADAGUC) and version was used to create the files.

Another feature of the ADAGUC WCS is the ability to re-grid GeoJSON vector files to NetCDF CF grids while preserving polygon attributes like NUTS_ID and NUTS_NAME. This feature is applied in the NUTS Statistics extraction Web Processing Service. This WPS extracts statistics per region according to NUTS data. Two identical grids are obtained, one for the NUTS geojson and one for the indicator data. These grids have the same dimensions and can easily be manipulated with python using numpy matrix calculations.

Overall, the ADAGUC WCS is used to harmonize all sorts of data to regular grids which can easily be interpreted by other tools like Python.

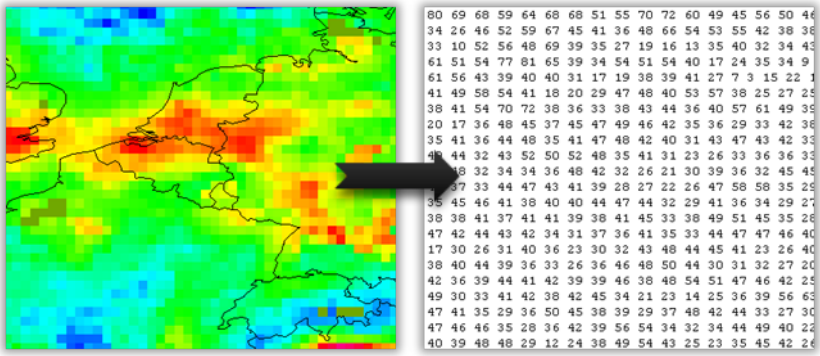

Figure 9 Web Coverage Services are used to obtain georeferenced data from a webservice.

How to use ADAGUC visualization functions

System overview and installation

An overview of the ADAGUC server is displayed below. The server and viewer are independent components and can be use standalone separated from each other. When a request to the server is made via the internet, at first a BASH shell script is called. This script sets a number of environment variables and finally calls the ADAGUC server executable.

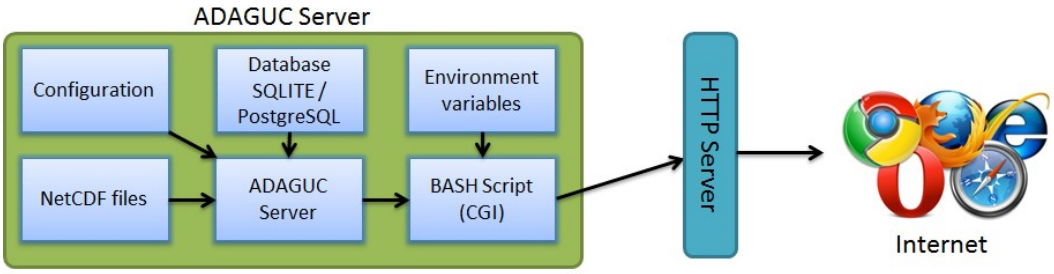

Figure 10 ADAGUC Server

The server on its turn reads out the environment variable (Config file, log file) and connects to the database and reads in NetCDF files. Finally the server returns an image or document. The required environment variables are:

- ADAGUC_CONFIG - pointer to the configuration file
- ADAGUC_LOGFILE - pointer where log messages should be stored, includes information logs and error logs
- ADAGUC_ERRORFILE - optional pointer which logs only error messages
- ADAGUC_FONT - Place where a TrueType font is stored, e.g. FreeSans.ttf
- ADAGUC_DATARESTRICTION - Optional pointer which controls access restrictions, by default set to FALSE, can be combinations of "ALLOW_WCS|ALLOW_GFI|ALLOW_METADATA|SHOW_QUERYINFO", separated with the | token.
    - FALSE: No restrictions (default, same as "ALLOW_WCS|ALLOW_GFI|ALLOW_METADATA")
    - ALLOW_WCS: Allows the Web Coverage Service, download of data
    - ALLOW_GFI: Allows GetFeatureInfo requests, e.g. getting information about a certain location
    - ALLOW_METADATA: Allows getting NetCDF header metadata information
    - SHOW_QUERYINFO: When a query has failed, the corresponding query will be presented to the user. This feature is disabled by default.
- ADAGUC_PATH - optional, is used as variable substitution {ADAGUC_PATH} in the configuration files, should point to the adaguc server installation
- ADAGUC_TMP - optional, is used as variable substitution {ADAGUC_TMP} in the configuration files, location where temporary files need to be written

- ADAGUC_ONLINERESOURCE - optional, specify the online resource in the CGI script itself, see OnlineResource to configure in the xml file.

The server requires the following libraries: HDF5, NetCDF4, UDUNITS, GDAL, LIBXML2, GD, Cairo, Proj.4, SQLITE and PostgreSQL.

Example bash script:

```
#!/bin/bash
export LD_LIBRARY_PATH=/build/lib/:$LD_LIBRARY_PATH
export PROJ_LIB=/build/share/proj/
export ADAGUC_CONFIG=/data/services/config/config.xml
export ADAGUC_LOGFILE=data/log/server.log
export ADAGUC_ERRORFILE=/data/log/server.errlog
export ADAGUC_FONT=/data/fonts/FreeSans.ttf
export ADAGUC_DATARESTRICTION="FALSE"
/data/build/bin/adagucserver
```

The server is able to read CF compliant files. In case of projected source data, a number of CF projections are supported. The server uses the ADAGUC standard for projected files, this means that a proj4_params attribute is used in the projection definition. See http://adaguc.knmi.nl/contents/documents/ADAGUC_Standard.html for details.

The server can read single files, or multiple files varying along a dimension stored together in directories. The server uses the database to lookup which dates belong to which file and time index. After changing the configuration, the database needs to be filled with data, this can be done with the following command:

```
adagucserver --updatedb --config myconfig.xml
```

Configuration:
Configuration of the ADAGUC server is described on the ADAGUC Wiki: https://dev.knmi.nl/projects/adagucserver/wiki/Configuration

Tutorials:
For ADAGUC several workshops were organized, outside the scope of CLIPC. But for these workshops many tutorials were created. The Tutorials can be found on the ADAGUC Wiki: https://dev.knmi.nl/projects/adagucserver/wiki/Tutorials

Subjects of the tutorials:

- Create a WMS service on a file
- Create a WMS service on a series of files with a time dimension
- Create a WMS service based on a remote dataset via OpenDAP
- Create a WMS service with multiple layers
- Styling of Layers with shading and contour lines
- Style a Layer with alpha transparency for cloud cover
- Configuration of an INSPIRE View Service

- Run ADAGUC Server from python

Virtual Machine

To make it even easier to start using the ADAGUC server a Virtual Machine is available. The Virtual Machine was created and tested under Oracle Virtual Box.

The Virtual Machine can be downloaded from the ADAGUC wiki:
https://dev.knmi.nl/projects/adagucserver/wiki/Preinstalled_virtual_environment

Conclusion and next steps

Visualization of indices is a critical component in order to convey information of indicators to the users. It needs to be highly performant and standardized. With ADAGUC we could provide a solid base for the CLIPC portal for visualization.

Within CLIPC the close cooperation between service developers (WP4), portal developers (WP3), data providers (WP6/7) and toolkit development (WP8) shaped the requirements needed for ADAGUC.

Based on these requirements, specific features were developed for CLIPC. These are the GeoJSON polygon/vector visualization for ranking, the tiling functionality for high resolution datasets, the ensemble visualization in the CLIPC portal, histogram visualization in the CLIPC portal and the NetCDF Web Coverage Service backend.

These features contributed to the success of the CLIPC portal and are also useful for the wider ADAGUC user community.

References

1. [OGC WMS specification] http://www.opengeospatial.org/standards/wms
2. [OGC WCS specification] http://www.opengeospatial.org/standards/wcs
3. [ADAGUC website] http://adaguc.knmi.nl
4. [ADAGUC workshops] https://dev.knmi.nl/projects/adagucserver/wiki/Workshops
5. [GDAL] http://www.gdal.org/
6. [HighCharts] http://www.highcharts.com/demo/
7. [OPeNDAP] https://www.opendap.org/
8. [PROJ] https://trac.**osgeo**.org/proj