Journal of
open research software

Adaguc-Server: Interactive Access to Heterogeneous Meteorological and Climatological Datasets Using Open Standards

SOFTWARE
METAPAPER

]u[ubiquity press

MAARTEN PLIEGER

ERNST DE VREEDE

**Author affiliations can be found in the back matter of this article*

ABSTRACT

Adaguc-server is an open source geographical information system to visualize, combine, compare and share real-time meteorological, climatological and remote sensing data via OGC standards. During implementation of the server, we ran into the limits of what is possible with OGC standards. To overcome the found limits, extensions to the OGC standards were developed to achieve the required functionality and interoperability. The presented solution is used as a building block in a web based meteorological working station.

CORRESPONDING AUTHOR:

Maarten Plieger

Senior developer GeoICT, KNMI, NL

maarten.plieger@knmi.nl

KEYWORDS:

GIS; OGC; Web Map Services; Meteorology; Visualization; Web Services

TO CITE THIS ARTICLE:

Plieger M, de Vreede E 2021 Adaguc-Server: Interactive Access to Heterogeneous Meteorological and Climatological Datasets Using Open Standards. *Journal of Open Research Software,* 9: 33. DOI: *https://doi.org/10.5334/jors.382*

(1) OVERVIEW

1. INTRODUCTION

In this article we detail the Adaguc-server, a geographical information system dedicated to visualize meteorological, climatological and atmospheric datasets via web services. Visualization and data serving is based on the Web Map Service (WMS) [4] and Web Coverage Service (WCS) specifications from the Open Geospatial Consortium. This ensures that web services offered by Adaguc-server can be used in other geographical information systems, e.g. using and adhering to standards should allow for interoperability between systems. The Adaguc-server is designed to compare datasets in space and time by offering geographical maps, histograms, ensembles and time-series of atmospheric datasets. The Adaguc-server supports real time update and aggregation of data streams, making data from radars, satellites and operational weather models accessible as a webservice. The system accepts data in large or small chunks, the data can be given to the system in many smaller files if necessary. The server reprojects the geographical data to allow spatial comparisons of different geographical projections. Graphical representations of the data can be made using contour lines, shading, hillshading, nearest neighbour rendering and symbology, see ***Figure 1***. The Adaguc-server is used to provide open access to meteorological data and to provide visualizations for a web based meteorological working station.

2. IMPLEMENTATION AND ARCHITECTURE

Supported data formats and data types

In the atmospheric domain, datasets can be very heterogeneous. Meteorological and climatological datasets are often represented in more than 3 dimensions and is continuously updated. In many cases these datasets have four dimensions: time, elevation, latitude and longitude. To provide uniform access to these different datasets and data formats, an internal data model was developed. In the following sections we will detail this data model and data formats it supports. We will also describe how support for live data streams is achieved.

The ADAGUC common data model (ACDM)

The Adaguc-server uses a data model to index incoming data streams. This is mapped to the Adaguc common data model (ACDM). The data model resembles the data model from NetCDF [12]. NetCDF [5] can be seen as an array container where data is stored in arrays of N-dimensions. These are often X/Y/Z/T, but other dimensions are possible, such as members from an ensemble model, or precalculated thresholds for statistics. As with NetCDF, the Adaguc common data model uses dimensions to specify the length of arrays, variables to store the actual data (array container) and attributes to specify the properties of each variable. The data model supports subsetting data via dimensions, helping to reduce data transfer and to minimize memory footprint during data processing. The data model supports aggregating individual files to a single data model. An indexing database is used to allow for live updates of services, fast querying of subsets and support for virtually unlimited dataset sizes. Adapters exist for several formats for conversion into the data model. This allows downstream software components to be simpler, as they do not have to worry about the original data formats. The ACDM idea is inspired by the Common Data Model (CDM) from Unidata [12].

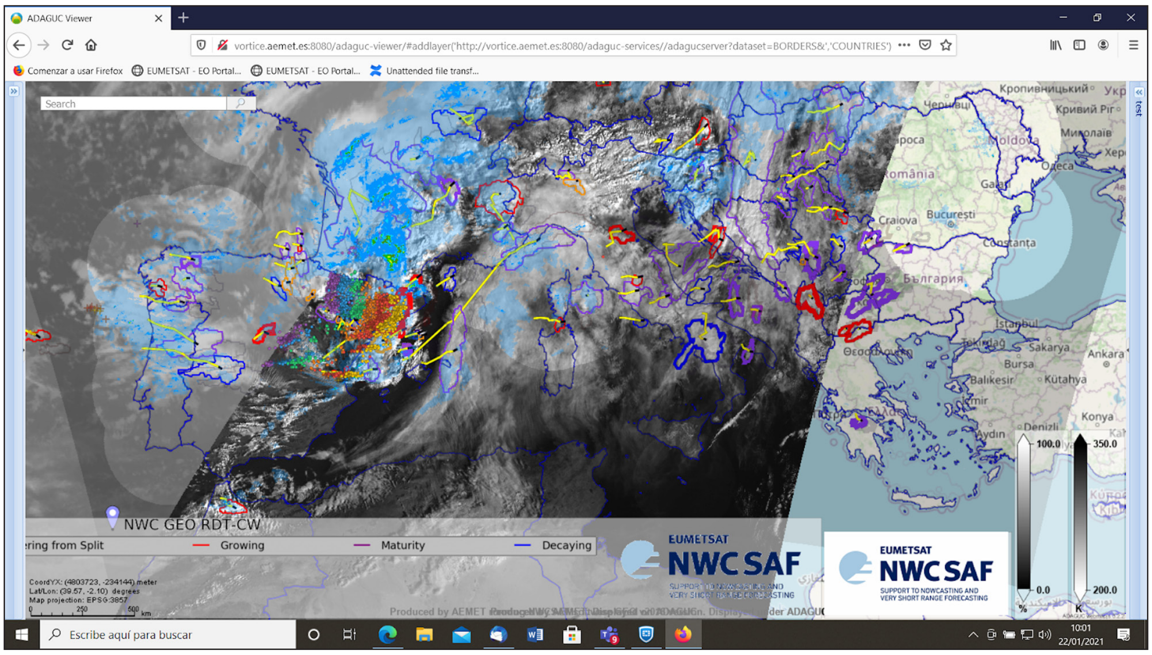

ADAGUC Viewer
vortice.aemet.es:8080/adaguc-viewer/#addlayer('http://vortice.aemet.es:8080/adaguc-services//adagucserver?dataset=BORDERS&','COUNTRIES')
Search
NWC GEO RDT-CW
Growing
Maturity
Decaying
EUMETSAT
NWC SAF
SUPPORT TO NOWCASTING AND VERY SHORT RANGE FORECASTING
100.0
350.0
0.0
200.0
%
K
Bucureşti
Constanta
Istanbul
Ankara
Bursa
Konya
Escribe aquí para buscar
22/01/2021

Figure 1 Application of Adaguc-server: Realtime WMS for NWCSAF in a Web application.

Supported Data formats
The server supports data types stored in the following data formats: NetCDF, HDF5, PNG, GeoJSON, CSV and OpenDAP [3]. OpenDAP is a data web service for scientific datasets [6]. The Adaguc-server reads these different data formats and data types into the same data model. For example the HDF5 data format can be read transparently into the same data model, as it is very similar to NetCDF. Both HDF5 as well as NetCDF use dimensions, variables and attributes to store data. Other data formats, such as PNG, GeoJSON and CSV, need more work to map to an uniform data model. For these data formats custom adapters have been written.

Data types
Adaguc-server can read regular grids, polygons and point data. Supported data types are for example model grids, structured and curvilinear grids, gridded data from satellites, radar observations, full color satellite imagery, satellite swath data, point observations, point time series and polygons. Adaguc-server is optimized to efficiently access data from remote OPeNDAP web services, like the services offered via Unidata's THREDDS server [2].

Aggregation of data files
Adaguc-server supports aggregations of multiple files into a single data and visualization service. This means that multiple files can be made available in a single WMS layer. The Adaguc-server allows for querying a sequence of files in a single webservice using the time dimension. In this case the files are concatenated using the time dimension, which means that the web service has one endpoint where you can query for different dates. From the user perspective it looks like there is one big file, but in practice several smaller files are internally used on the server. This means the individual data files can be small, speeding up access to huge datasets. The webservice can extend the dataset when new files arrive. New files can be added to the service at any moment without causing interruptions. This feature can be used for building real-time weather services, where new data is available constantly.

Database for indexing live data streams
To keep track of all the files and their dimensions, a relational database is used. This can either be PostgreSQL (preferred) or SQLite. This database is used to store an index of NetCDF files. These can contain many model levels and time steps in one file. Therefore it is useful to index a reference to the precise location of the data blocks inside the file. The server does this only for the non-geographical dimensions such as time and elevation (not the X and Y dimensions); allowing Adaguc-server to quickly find the corresponding file when a request for a certain time step is made. This database is used to support live updating of the service without interruptions or restarts.

Built-in data converters and post processors
Adaguc-server has a number of data converters and data post processors to support non-standard data conventions. It is for example possible to combine multiple variables into a single variable, to change the unit of measurement (e.g. Kelvin to degrees Celsius), or to change radar reflectivity (dBZ) into precipitation rate (mm/hour).

Adjusting metadata and data structure
The internal data model allows for tweaking the structure and metadata of the data. Adaguc-server can read a NetCDF file with bad metadata and then fix it by applying NCML which is a descriptive file in the NetCDF Markup language [10]. After the data file structure is read into the data model, it can be adjusted with NCML.

NCML describes how (meta)data needs to be changed in a standardized way. NCML allows dimensions, variables and attributes to be changed, removed or added. This means that a file which has incorrect metadata can be adjusted to become a file with correct metadata. This flexibility allows Adaguc-server to visualize many data files with bad or missing metadata. It can also be used to visualize existing datasets which cannot be fixed (regarding metadata). This feature is inspired by implementations in the Thredds data server, which has similar functionality.

Making meteorological data accessible via open standards
Adaguc-server can translate multi-dimensional datasets into a Web Map Service, enabling querying of the provided dimensions via the Web Map Service. The dimensions are advertised in the WMS GetCapabilities document. Viewers capable of using these dimensions can then be used to browse and visualize the data in a web browser. The WMS specification specifies two special dimensions, ELEVATION and TIME. Other dimensions are prefixed with DIM_< dimension name>, as specified in the WMS OGC standard). If needed, the dimension name can be adjusted via the configuration of Adaguc-server.

Extensions to the Web Map Service standard
The Adaguc-server provides extensions to the WMS standard to make meteorological data available in a convenient way. For a part these extensions are based on the NCWMS [1] extensions and the MetOcean Domain Working Group [8]. Others are the result of resolving difficulties that arose during implementation. This section summarizes the extensions.

JSON output format for GetFeatureInfo
In weather applications it is often desired to see a time series graph for a certain location on the map. Using WMS, we want to be able to query the WMS layer for a time sequence. The standard WMS GetFeatureInfo supports querying a single location for a single value and returns

either text, xml or html. For our application we need to efficiently get multiple values for a specific location, and output it in a web developer friendly format. Both requirements are not supported by the WMS standard specification, therefore extensions to the WMS standard were needed. This extension supports querying a single location for multiple timesteps and supports returning data in the JavaScript Object Notation (JSON) format. Querying multiple dates can be done by using wildcards or time range queries (start/end).

Wildcards for dimension querying using GetFeatureInfo

By default, WMS does not support querying multiple values at once. To get a timeseries, you have to query each data separately. As we found it cumbersome to compose multiple requests with the same dimension value, Adaguc-server supports the option to use wildcards for dimensions. This includes support for range querying (start/end) and wildcards such as an asterisk '*' returning all values for that dimension, see ***Figure 2***.

GetPointValue extension for GetFeatureInfo

The Adaguc-server further extends the standard by adding an extra request function called GetPointValue. Normally with GetFeatureInfo you query the map obtained with the GetMap request. This map has a coordinate system in screen space coordinates, and therefore the GetFeatureInfo also queries in screen space coordinates. The GetPointValue extension allows for querying a location in projection coordinates. Any projection known to the Pro.4j library is supported.

Support for weather models with forecast runs

Adaguc-server can serve weather model data as a Web Map Service. Weather models share a specific feature: model runs provide forecasts for the future and have an overlap with the previous run. That means that for most timestamps, multiple model runs are available. Often it is desired to see the latest run, but sometimes you want to be able to look at previous model runs, for example when you want to see how a forecast develops for the same timestamp.

The MetOcean domain working group came up with best practices [8] for serving weather models through WMS. Adaguc-server is following these. At the moment this is done by using a reference-time dimension in combination with time dimension. The reference-time value can be set to the model run you want to see.

This immediately raises the problem that it is impossible to know what the valid combinations are for time and reference time from only the GetCapabilities document, as shown in ***Figure 3***.

In the WMS GetCapabilities, two dimensions with ranges from 00H to 18H hour with 3 hourly increments and one from 00H to 12H with hourly increments are given. These form a matrix of combinations. The problem is that you cannot know which combinations are valid (marked yellow). It is not possible to make the time dimension dependent on the reference time dimension. It would be nice to have the possible time values for a given reference time value.

To overcome this issue, a heuristic is to use the difference between the end of the reference dimension and the time dimension. This gives an indication of the length of each model run. Smart viewers can take this into account when showing model data from such a webservice. The most practical solution would be to add extra functions extending the WMS standard to enable clients to obtain this information. Adaguc is experimenting with an extra call to get the list of model runs for a certain time step and with a call to get the available times for a given model run.

ncWMS Extensions: Change the color range of the legend

ncWMS is a Web Map Service implementation for environmental geospatial data. ncWMS is developed by

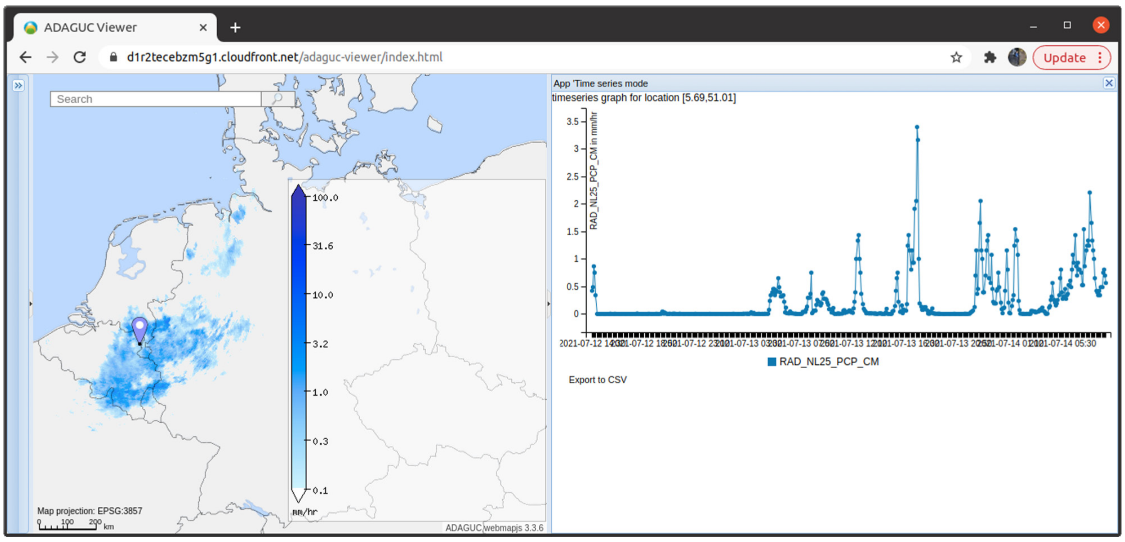


Figure 2 Example of using WMS extensions to display a timeseries graph.

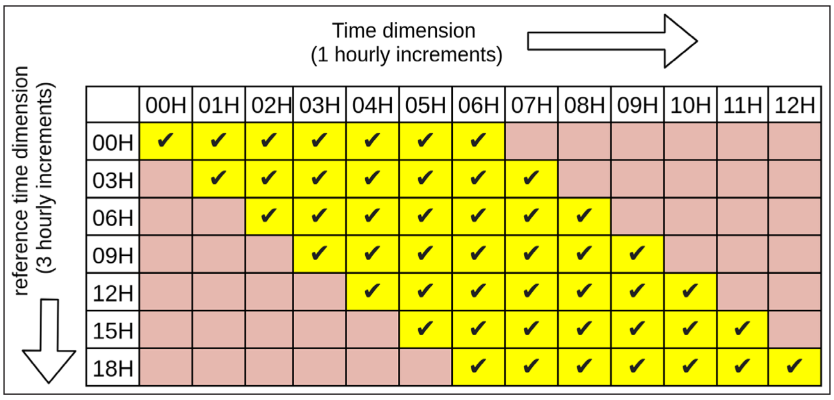
the Reading e-Science Centre at the University of Reading, UK [1]. ncWMS has designed new extensions to the WMS standard. Some of these extensions are very useful and the Adaguc-server implementation has already adopted some of these extensions. One very useful feature is the ability to change the color scale and range of a map.

Title, subtitle and legend extensions

During development of the Climate Atlas for the Netherlands, there was the requirement to generate web page publication ready images using a single WMS GetMap request. Therefore we needed the option to add a title, subtitle, legend, north arrow and scale-bar to the map. With these extensions it is possible to generate a map as shown in ***Figure 4*** using a single GetMap request.

GetHistogram extension

Certain projects at KNMI required to get the statistics for the viewed map. For this purpose the GetHistogram request has been added. It works very similar to the


Time dimension
(1 hourly increments)
reference time dimension
(3 hourly increments)

	00H	01H	02H	03H	04H	05H	06H	07H	08H	09H	10H	11H	12H
00H	✓	✓	✓	✓	✓	✓	✓						
03H		✓	✓	✓	✓	✓	✓	✓					
06H			✓	✓	✓	✓	✓	✓	✓				
09H				✓	✓	✓	✓	✓	✓	✓			
12H					✓	✓	✓	✓	✓	✓	✓		
15H						✓	✓	✓	✓	✓	✓	✓	
18H							✓	✓	✓	✓	✓	✓	✓

Figure 3 Weather model runs: Relation between reference time and time.

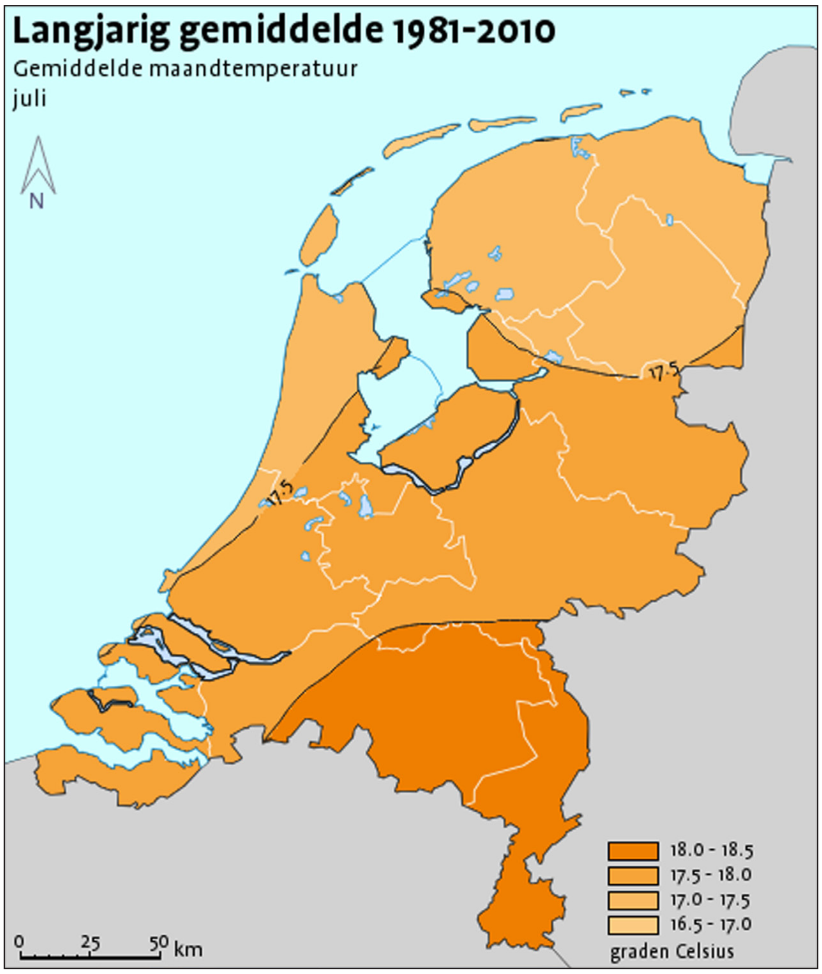


Figure 4 Image from *http://www.klimaatatlas.nl/* showing climate normals for monthly average temperature in July. This image is generated by a GetMap request. By using WMS extensions, it becomes possible to add a title, subtitle, scalebar, legend and north arrow in the map.

GetMap request, with the difference that it returns a histogram in the form of a JSON output. It gives the distribution of the actual values in the map, see ***Figure 5***.

WMS GetLegendGraphic extension

For the Dutch "Bosatlas for climate" we have long term climatological data containing the monthly average temperature in the Netherlands. In the Netherlands temperature varies from freezing during winter, to tropical temperatures in summer. The temporal difference in temperature between the months is big. However the spatial difference in temperature within a month is small. We want to use the same color scale to compare all twelve months with each other. Using the same colours for the classes allows for referencing the same temperatures between months. See ***Figure 6***. below.

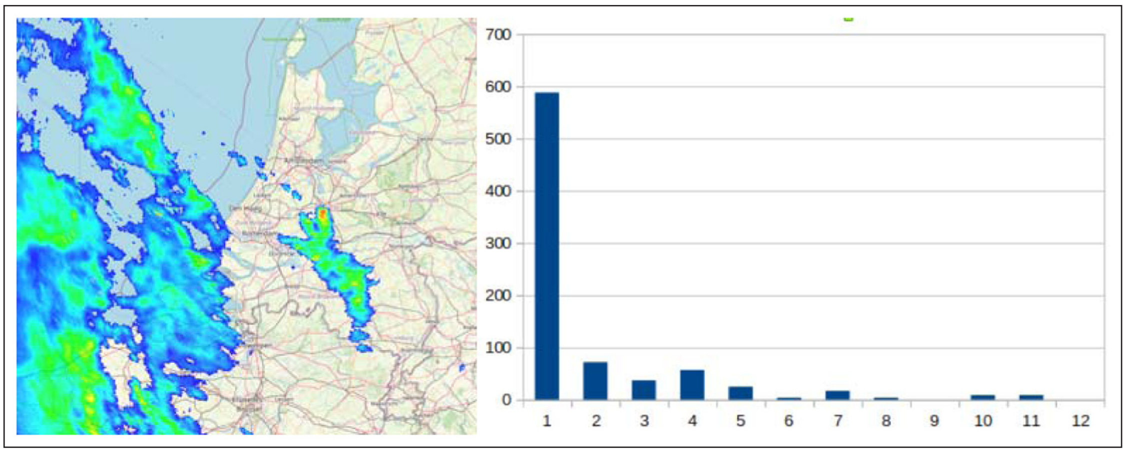


Figure 5 The GetHistogram request (on the right), similar to the GetMap request (on the left), provides statistics for the area in view using a web friendly format.

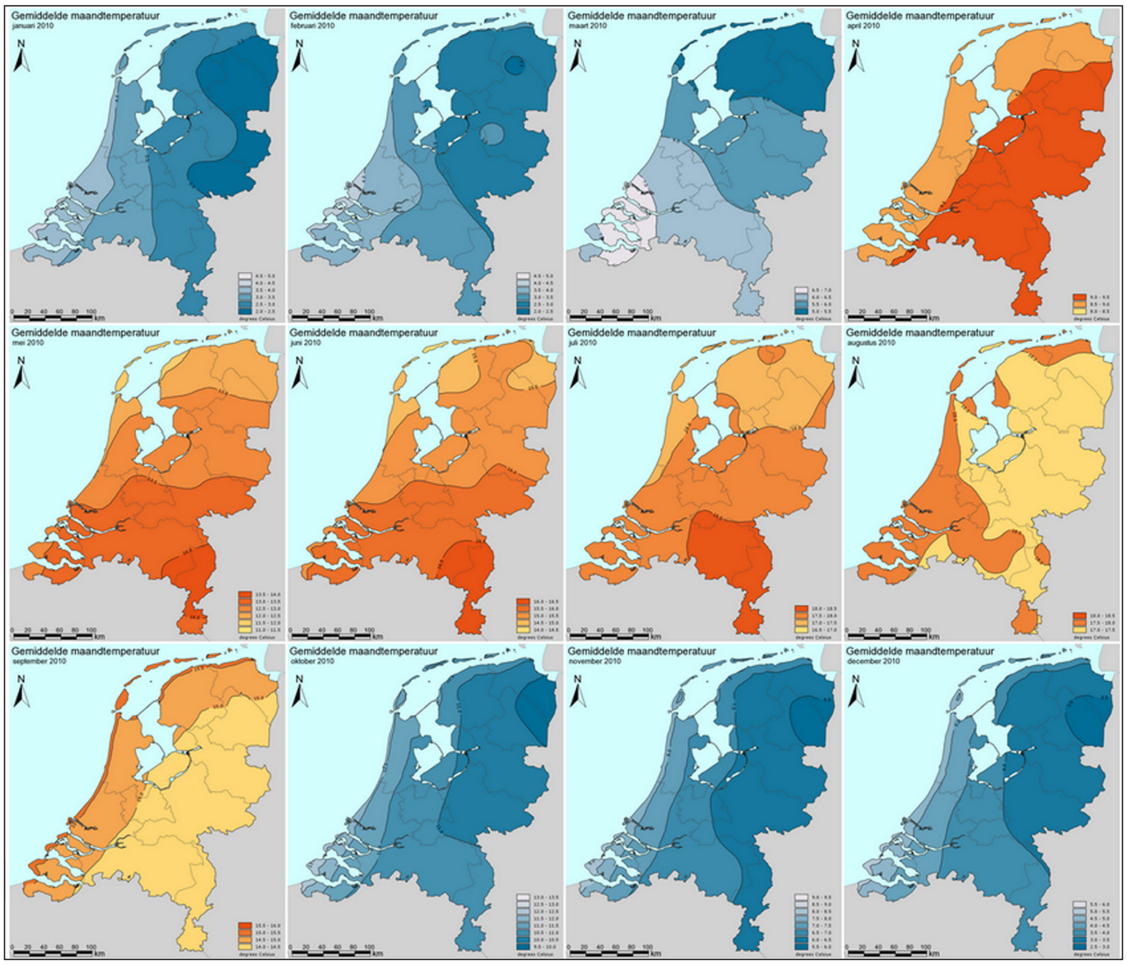


Figure 6 The same legend is used for all twelve months. Spatial difference in temperature for a month is low, temporal difference between maps is high.

As you can see, the legend for each month looks different, but it is based on the same colour classes. The red value in one map is the same red value as in the other. The legend image displays those classes which are available in the data (for that month). That means that the legendgraphic is different for each month and changes according to the time dimension. The Adaguc-server supports GetlegendGraphic in combination with any dimension. The server will figure out which classes are available on the map. This extension can be enabled by setting it in the style definition used for this layer. A web viewer needs to provide the same dimension information to the GetLegendGraphic as it provides to the GetMap request.

Vertical profiles: soundings for a location

Vertical profiles are more or less the same as time series, with the difference that the elevation dimension is queried for multiple values. Using the same wildcard extension as described previously, it is possible to easily obtain vertical profiles from a weather model. By simply using a wildcard to query for all elevations in combination with the option to return the data in a web friendly format, a JSON with the vertical profile is returned. This allows a frontend application to display the sounding from a weather model at any desired location as shown in ***Figure 7***. It is like querying a data cube [7] with weather model data.

Ensemble data for a location

Weather models are often run in parallel on supercomputers with slightly different variations of the start conditions. This results in many slightly different outcomes; an ensemble of model fields (called ensemble members) containing different weather forecast realisation for a certain period in the future. Statistics can be derived from the statistical distribution of the members. The closer the ensemble members relate to each other, the more certain the forecast is. When the ensembles differentiate from each other, the forecast is more uncertain. A typical application for these data is a visualization using a plume plot. The plume plot is composed from a set of ensembles and shows uncertainty over a given time period.

Within the Adaguc-server, the different model ensembles can be organized with an extra ensemble dimension, for example named DIM_ensemble in the request. It becomes possible to get the timeseries for a location of all ensembles by using the same GetFeatureInfo of GetPointValue extensions. It is also possible to use the wildcard for DIM_ensemble, resulting in the returning of all ensemble data available for that location. See ***Figure 8*** for an example.

Adaguc-server components

In this section we describe the architecture of the Adaguc-server. This section is not intended to go into the very details of the application.

Adaguc-server C++ application

The Adaguc-server is a visualization and data engine written in C++. C++ was chosen (in 2009) as this was one of the fastest higher level languages at the time. Advantages are that it is very popular and many libraries are available. The Adaguc-server component by itself is not a web server; it is just an application which needs to be executed to do something. It cannot be started and run as a web service without some extra tooling. In order to make it accessible as a web service it needs an application server. The Adaguc-server is made to comply

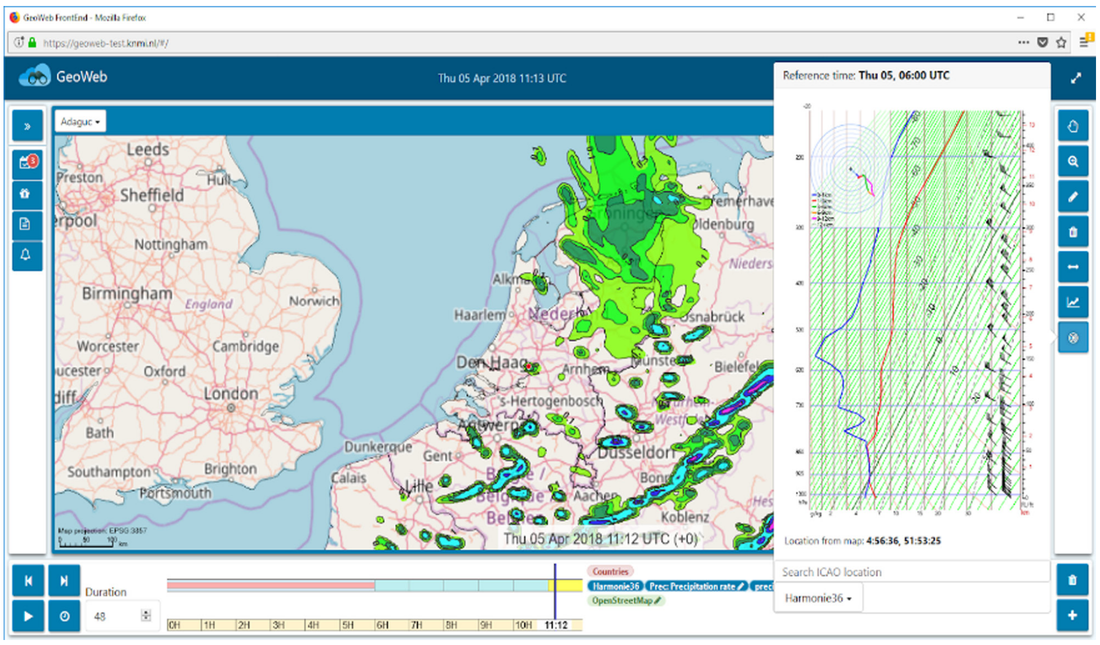

GeoWeb FrontEnd - Mozilla Firefox
https://geoweb-test.knmi.nl/#/
GeoWeb
Thu 05 Apr 2018 11:13 UTC
Reference time: Thu 05, 06:00 UTC
Adaguc
Thu 05 Apr 2018 11:12 UTC (+0)
Location from map: 4:56:36, 51:53:25
Search ICAO location
Harmonie36
Duration
48
Countries
Harmonie36
Prec: Precipitation rate
OpenStreetMap
0H 1H 2H 3H 4H 5H 6H 7H 8H 9H 10H 11:12

Figure 7 One of the earlier OpenGeoWeb applications, displaying the KNMI Harmonie weather model with a vertical sounding (Progtemp / Bijvoet diagram).

with the Common Gateway Interface (CGI) [13] standard. This makes it possible to run Adaguc-server directly from a server application like the apache httpd server.

The C++ application is executed once for each request. For each request a new process is made. This has both benefits and drawbacks. The main drawback is that the application has to be started when a new request comes in. This can cause overhead, especially in the time it takes to complete a request. We try to make the overhead minimal by making sure that no precious time is lost for initializing, reading configuration files or connecting the databases. In our experience the overhead is small in relation to time it takes overall to complete a WMS request. Data and image processing generally take more time than starting a process. There are also advantages: 1) the Adaguc-server can be easily run in parallel. The application server allows running as many Adaguc-server processes as desired in parallel, fully utilizing the available resources. 2) Each process stops after it has handled the request. Resources that were accidentally not freed are now automatically unallocated. For example connections to the database cannot be left open. The webserver can also kill processes which take too much time, making sure that system load does not cause a denial of service. 3) Dataset and configuration updates have a direct effect. This can be convenient when editing graphical styling of datasets, see ***Figure 9*** for an overview of the components.

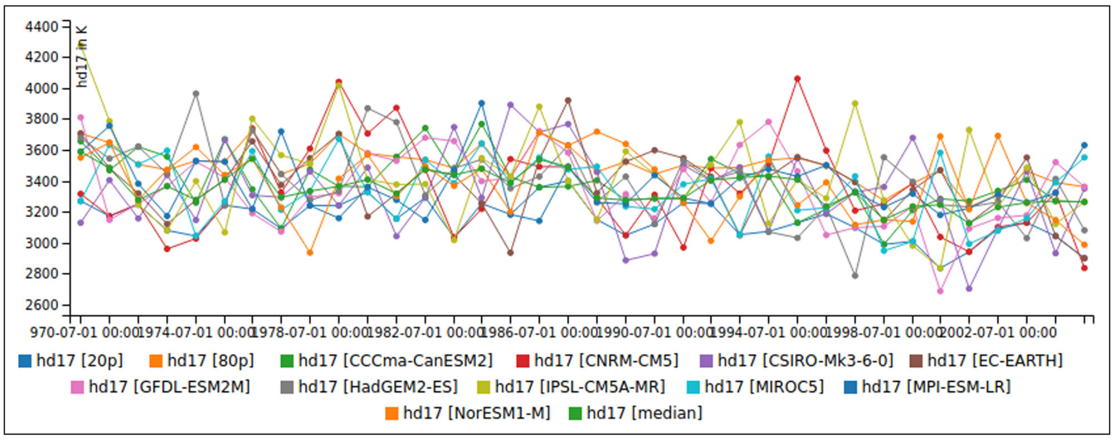


Figure 8 Wildcards for GetFeatureInfo. When using DIM_MEMBER=* in the request, all other members will be returned in the response. Smart clients can then show the different members or ensembles in a single plot.

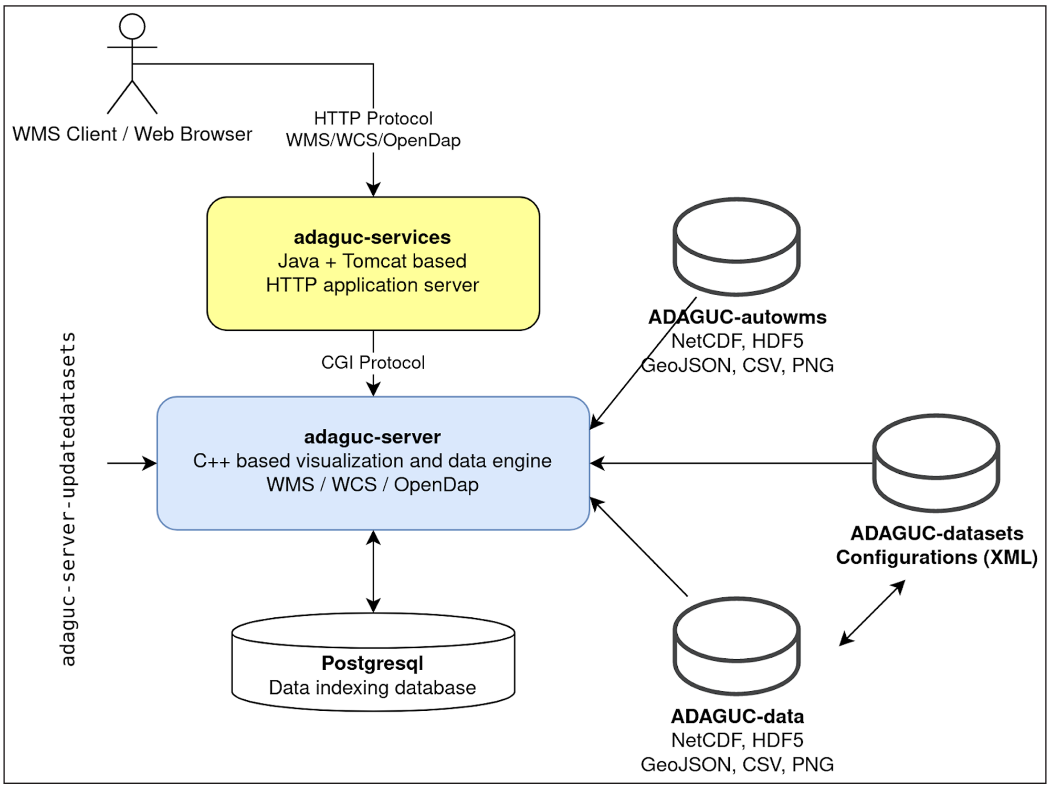


Figure 9 Overview of the Adaguc-server components.

Adaguc configuration

The Adaguc server has three important directory folders: adaguc-autowms, adaguc-datasets and adaguc-data.

The **adaguc-autowms** folder is meant to be used by developers and researchers. Any compatible file in this folder becomes automatically available as a WMS. NetCDF files in this folder are indexed and all dimension values in these NetCDF files become available in the WMS. The autowms can be configured to select graphical styling based on the variable and standard names in a NetCDF file. The autowms directory can interactively be browsed in the adaguc-viewer via the autowms web service. Here you can browse the files in your autowms directory. You can click on them and add them to the viewer. In practice it turns out to be useful for exploring various datasets in a web browser.

The **adaguc-datasets** folder is used for configuration of data via small XML files. These small files contain layer definitions, their graphical styling and configure how files should be concatenated in the webservice. This is often used for real time data streams. The data referenced in these files is normally stored in the adaguc-data folder. Files in the adaguc-datasets folder are called a dataset. They can be accessed in the WMS using the DATASET=<datasetname> parameter in the URL.

The **adaguc-data** folder contains the actual data. Usually different datastreams are divided into different folders. The files in adaguc-data can be live streams of meteorological data, like the previously mentioned precipitation radar data with an update frequency of five minutes. When a new file comes in, the Adaguc-server-updatedatasets script needs to be triggered. This will add an index of the new file in the database. After this step is completed, the new data file is available in the webservice.

Docker setup for Adaguc

A docker-compose file is available to start Adaguc-server and all its components with a single command. The docker has the same folders mentioned in the previous sections. It is very convenient to use the docker, as it allows for quick and easy setup of Adaguc-server on many different systems, see ***Figure 10*** for an overview of all components setup with docker-compose.

Graphical styling of datasets

The WMS GetMap request returns an image from the server to a client, for example a browser application, like OpenLayers or Leaflet. The GetMap image is styled according to the stylename given in the request. The style name corresponds to a style configuration on the server. For the WMS GetMap request the Adaguc-server can style data with different methods as described in the following sections.

Nearest neighbor interpolation

One of the fastest and simplest methods is nearest neighbor interpolation [14]. The pixels in the requested image simply resemble the values of the nearest grid cells from the source data, see ***Figure 11*** for an impression. Nearest neighbour interpolation is one of the fastest image rendering techniques and gives a good impression about the source data. It can be a natural way of showing

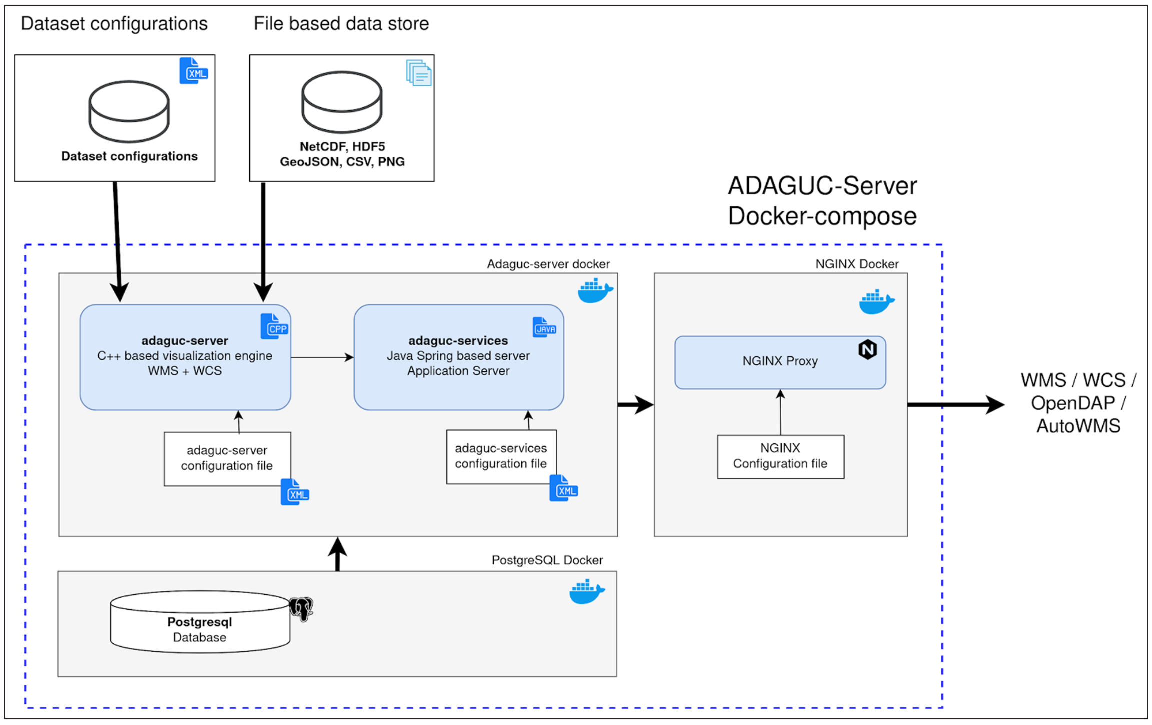


Figure 10 Overview of the provided docker-compose file, which starts all needed Adaguc components with a single command.

atmospheric datasets which is why it is used quite often in GIS applications. Adaguc-server makes use of the Proj library to support many different geographical projections. Adaguc is heavily optimized to achieve very fast image transformation. Adaguc-server starts with estimating what data is visible on the screen. It tries to transform only visible data. Second, multiple threads are used, each is responsible for a specific section of the screen. Third, when coordinates are close to each other, Adaguc-server can do an estimate for coordinates lying in between, without calling the proj4 library. By using threads, bypassing the proj4 library and doing good estimation of what is visible on the screen, image transformation in nearest neighbour mode is fast. Precipitation radar imagery in the polar stereographic projection can be warped and coloured to a mercator projection under 200ms on a modern system.

Bilinear interpolation

Bilinear interpolation [15] will generally result in smoother looking images, see ***Figure 12***. This can be used when the source data has a low geographical resolution.

In Adaguc-server the bilinear interpolation is done by splitting a quad into two triangles and filling these triangles using barycentric coordinates ***Figure 13*** shows. The picture on the right shows how different colours can be rendered with a smooth transition using barycentric coordinates, as explained by Lewis van Winkle in triangular interpolation using barycentric coordinates [20].

Contour Lines and shading

Contour Lines are useful to indicate the same value at given intervals. This results in a line. The line is plotted where the values in the map are the same. This is useful when there is a small gradient in the actual values, like monthly temperature maps. It is possible to define many contour definitions with custom colours, line-widths and text formatting.

Shading is similar to contour lines, but where a contour line is drawn exactly at a given value, shading will fill a surface between an upper and lower value. Between the given values the surface is shaded with the requested colour, see ***Figure 14***.

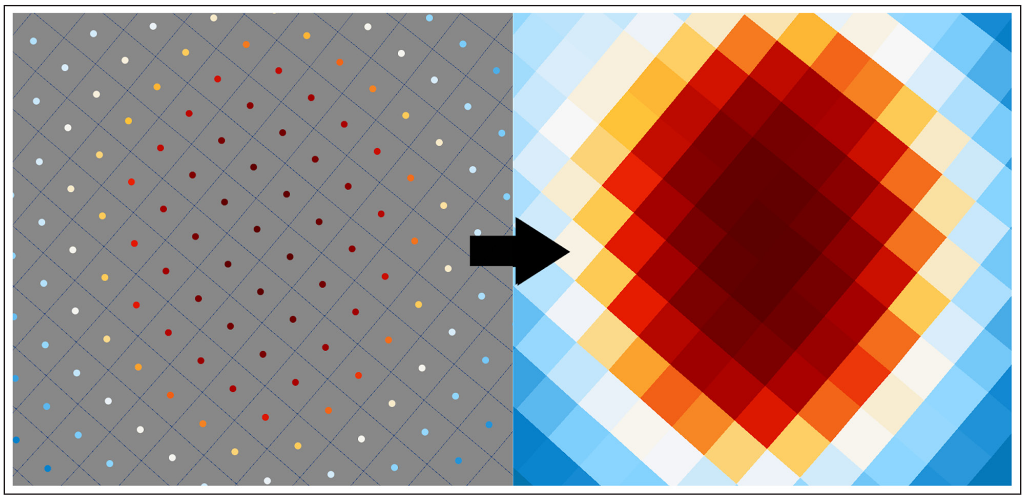


Figure 11 Impression of how Nearest Neighbour interpolation is used to fill the image. On the left, the center of the grid cells, on the right the filled grid cells.

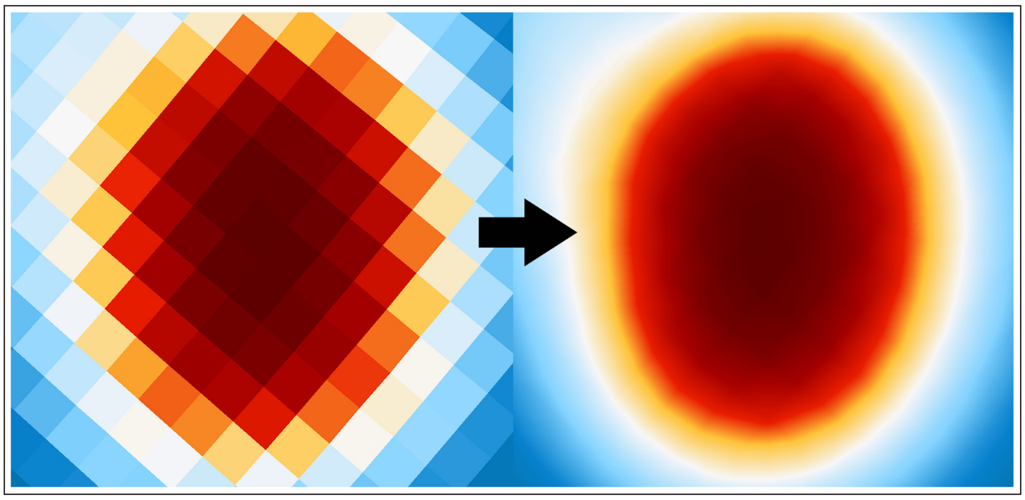


Figure 12 Comparison of nearest neighbour interpolation (left) versus bilinear interpolation (right).

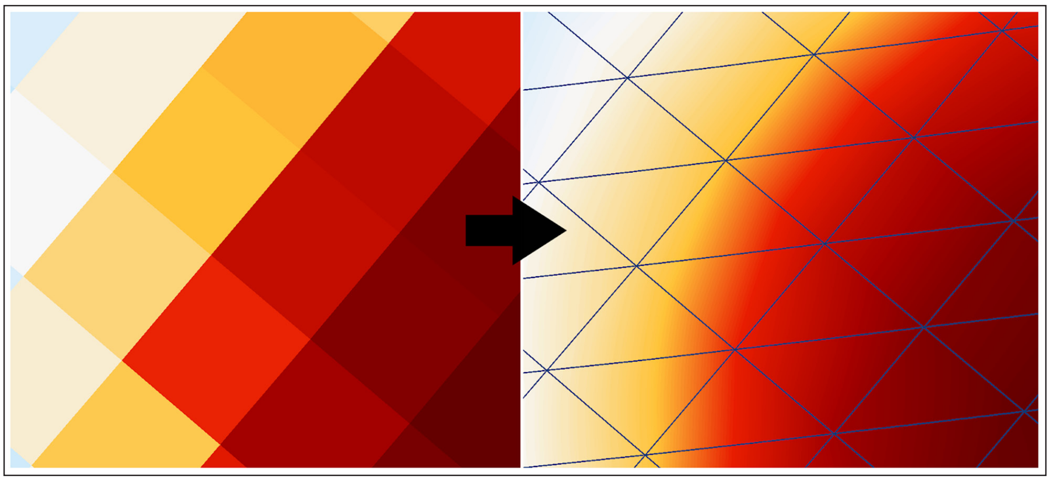


Figure 13 Triangles in combination with barycentric interpolation for achieving bilinear interpolation.

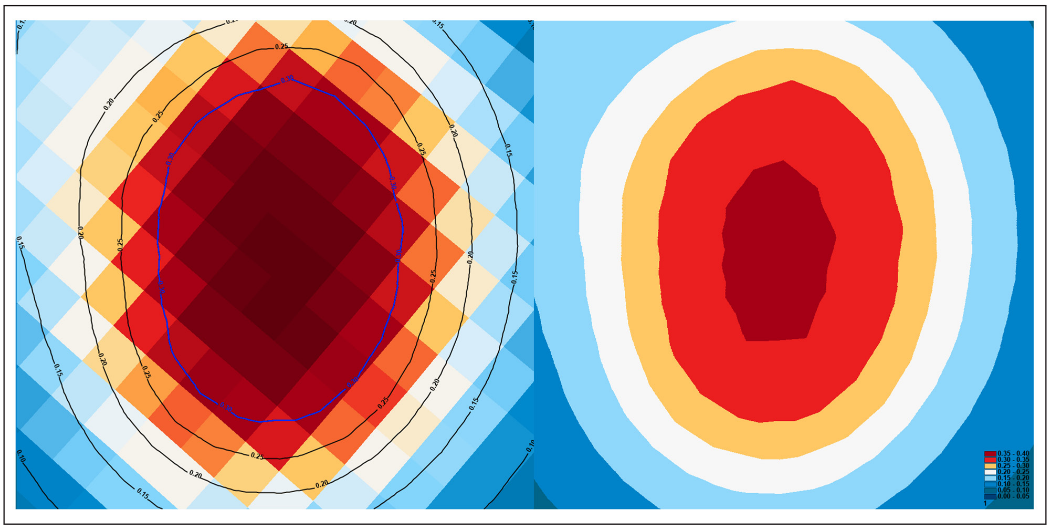


Figure 14 Application of contour lines (left) and shading (right).

Hill shading

Hill shading [18] is a technique to add a 3D effect to elevation imagery. Hill shading is usually applied to digital elevation models (DEM: [16] to achieve realistic impressions of hills. Hill shading is based on the digital elevation model and a lightsource. First the normal map is calculated. The normal map represents the angle of the grid cells [17] The second step is to combine the normal map with the light source. This will calculate how much light a grid cell receives from the light source. The lightsource is usually set on the top left corner of the map, as humans are trained with light sources from the upper-left [19]. The hill shaded map represents values between zero and one. Zero (0) means that the grid cell is on the darker side of the hill, while one (1) indicates that the grid cell is orientated towards the light source. Values in between are given, where 0.5 indicates a flat surface. These values can be used with a black to white gradient legend and alpha transparency to shade hills lighter and darker on the map, see ***Figure 15***.

This effect greatly enhances the impression of the terrain. Small differences in elevation are normally not visible, but with hill shading these differences become visible due to the added relief. The technique can be applied to any gridded dataset, such as flow data from a weather model to highlight eddy currents.

Stippling

Stippling is useful to indicate uncertainty. Weather forecasts and climate simulations often provide uncertainty information about the predicted values. By using stippling it is possible to indicate uncertainty as an unobtrusive overlay on the map, see ***Figure 16***.

Symbols – cloud, lightning, weather

With symbology it is possible to style data values using icons. Currently this is only available for point data. A typical use case is to display a weather map with weather symbols. Another application is to display cloud cover, which is usually measured in octa. Octa is indicated with dedicated symbols; a circle with a certain pattern. Adaguc can display symbols provided in the PNG image format. The symbol can be displayed for a given value range, as shown in ***Figure 17***.

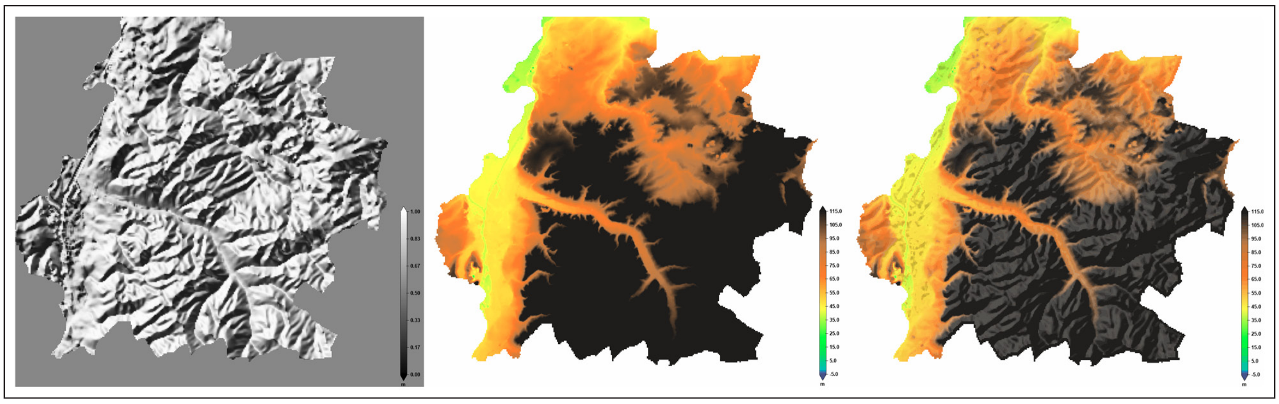


Figure 15 Visualisation of the Dutch elevation AHN dataset (Algemene Hoogtebestand Nederland) using hillshading. The left shows the raw hillshading layer, the middle shows standard rendering without hill shading and right shows rendering with hillshading.

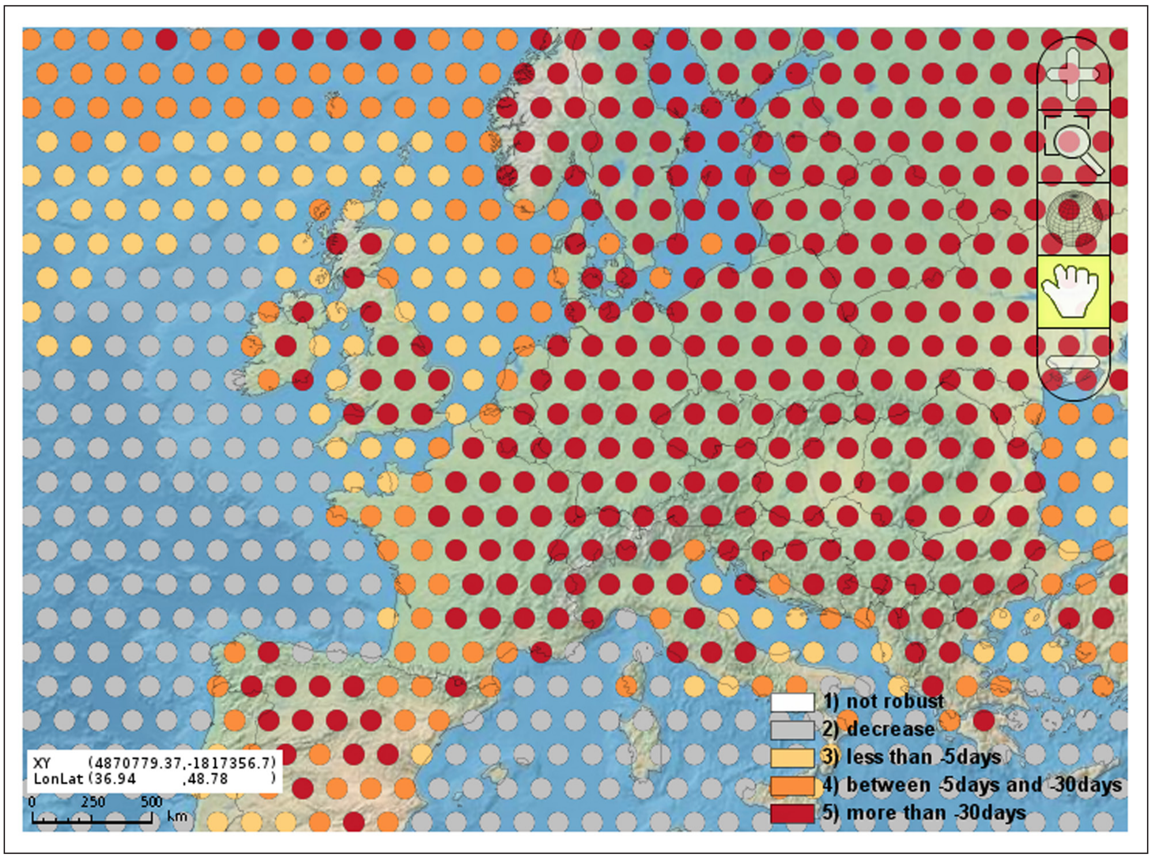


Figure 16 Stippling used to indicate uncertainty.

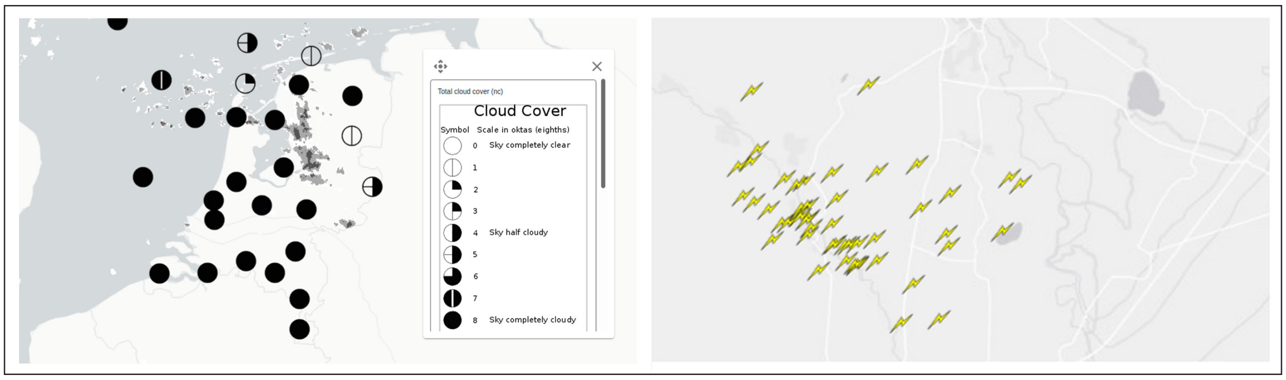


Figure 17 Left: Displaying cloud cover using symbols (unit in octa), right: symbols indicating cloud to ground lightning.

Wind vectors and Wind barbs

In the meteorological domain weather models generate output for various physical quantities such as wind. Wind is a vector composed from at least two values. These can be a horizontal and a vertical component, or it can be direction and speed. Based on this data, Adaguc can render wind vectors or meteorological wind barbs. Adaguc will rotate the barbs and vectors with respect to the current geographical projection. For some geographical projections the north pole is not always at the top of the map. Adaguc will rotate the wind vectors accordingly. See ***Figure 18*** for an example.

True color imagery

Adaguc has the capability to display imagery, ***Figure 19***. This can be topographic maps, satellite images or plain photographs. True color imagery can be encoded in four channels: red, green blue and alpha transparency. These channels are stored in image files like PNG or NetCDF files. In adaguc we support true color PNG and NetCDF files. For NetCDF we use a 32 bit unsigned integer (NC_UINT) to store 4 channels with 8 bits per channel. This way the same NetCDF format can be used to store true color data as well.

Conclusion

In this article we have described the purpose and capabilities of the Adaguc-server. With this software solution, datasets from the meteorological, atmospherical and climatological domains can be harmonized and visualized via open standards. Real time data streams with dimensions such as elevation, time or any other can be made available. Extensive visualizations options are available for data styling.

One of the aims is to make visualizations via Web Map Services interactive. We have put significant effort in making the Adaguc-server as fast as possible. This is achieved by loading the minimum required dataset to complete the request, and optimizing the reprojection and rendering pipelines.

For interoperability with other tools, we try to adhere to the standards from the Open GeoSpatial Consortium where possible. As described, during implementation we hit the limits with respect to making these kinds of data available via these standards. To overcome these limits, extensions were developed.

In the OpenGeoWeb project we are implementing a web based meteorological working station for the

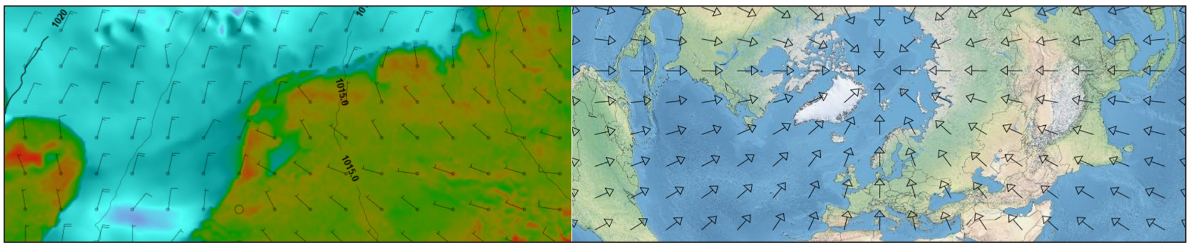


Figure 18 Left: Wind barbs for the KNMI Harmonie weather model, right: wind vectors all flowing northwards with geographical perspective correction.

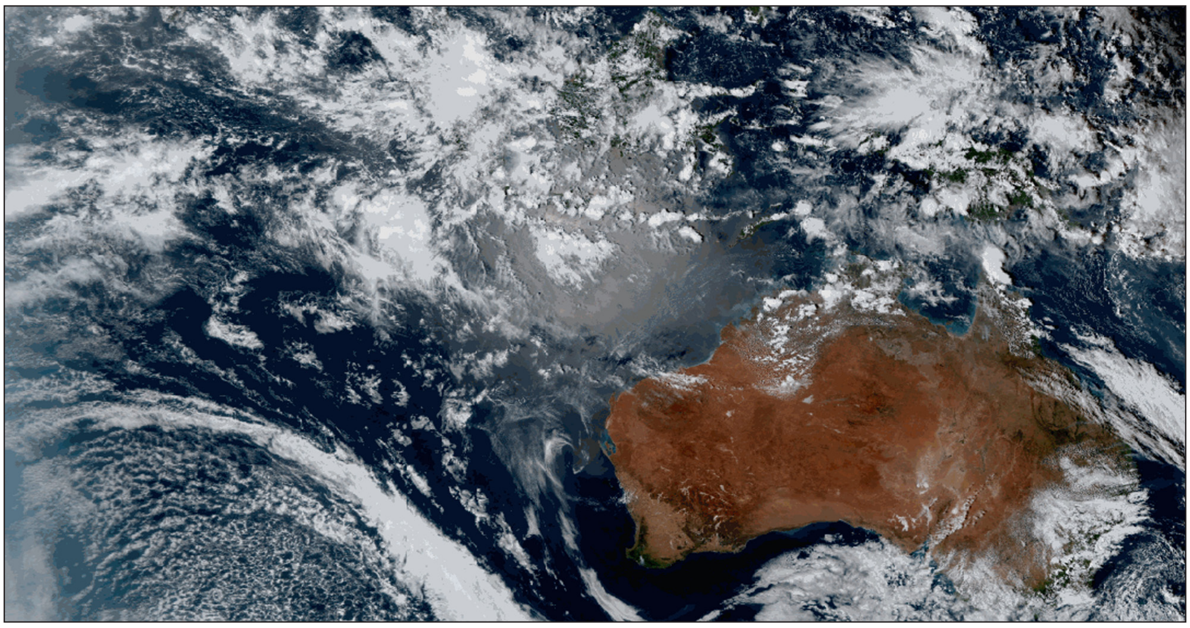


Figure 19 Natural view above Australia from the Himawari Satellite.

operational weather forecasters. Benefits can be found in the use of web technology, the option to combine datasets offered by different parties and the ability to work independently of location. Also we see opportunities to bring research faster and safely to the operational weather forecasting office.

Adaguc-server performance benefits from fast file systems which support seeking and reading small pieces from a file. Thanks to the indexing database, Adaguc-server can pinpoint the exact location and size to read from a file. This is often a fraction of the total, which makes the Adaguc-server efficient and responsive on fast block storage devices. In the near future, cloud storage solutions will play an important role in storing scientific datasets. The current way of using large NetCDF files is not a good match for object storage systems. These systems do provide minimal support for reading small chunks from an object. We foresee that research is needed to unlock the full potential of object storage systems in relation to Adaguc-server. We think that the indexing database can help with assembling objects to a dataset and provide transparent access to the original data.

3. QUALITY CONTROL

The Adaguc-server software is largely tested with functional tests and to some extent with unit tests. These functional tests do black-box testing of the software using test datasets and configuration files. For the functional tests, the Adaguc-server is started using PyTest. Within the testing framework, an instance of Adaguc-server is started and its response is evaluated and compared to expected results. These functional tests ensure that the behaviour of the software is left unchanged during development cycles. For each reported issue, we attempt to add a functional test which reproduces the issue. After fixing the issue, the test is updated. For many use cases there are functional tests made, the number of tests is increased for each added functionality.

(2) AVAILABILITY

OPERATING SYSTEM

The software is running on any recent Linux operating system (2020 and onwards).

PROGRAMMING LANGUAGE

C++ 11 and Python3.

ADDITIONAL SYSTEM REQUIREMENTS

Depends on how many users being served. For a local instance Adaguc-server runs well on a multicore machine with at least 2GB of ram. Adaguc-server relies on how fast the data can be read from disk into memory. A solid state disk will perform better than a traditional spinning hard drive.

DEPENDENCIES

PostgreSQL, Docker

LIST OF CONTRIBUTORS

Maarten Plieger, Ernst de Vreede, Saskia Wagenaar, Ian van der Neut, Simon van Bennekom, Jonas Matser, Maarten Terpstra, Friedrich Striewski, Loes Cornelis and Joao Macedo

SOFTWARE LOCATION

Archive: *https://github.com/KNMI/adaguc-server*
Name: Adaguc-server v2.5.11
Persistent identifier: *https://doi.org/10.5281/zenodo.5092194*
Licence: Apache 2.0
Publisher: Maarten Plieger - KNMI
Version published: *2.5.11*
Date published: 12/05/21
Code repository (e.g. SourceForge, GitHub etc.) (required)
Name: Adaguc-server
Identifier: *https://github.com/KNMI/adaguc-server*.
Licence: Apache 2.0
Date published: 12/05/21

LANGUAGE

C++ and Python

(3) REUSE POTENTIAL

GETTING STARTED WITH ADAGUC-SERVER

The easiest way to get started with Adaguc-server is by making use of the dockerized version. A docker-compose file is available to start the required infrastructure. With the docker-compose, the Adaguc-server, postgresql and nginx are started. This makes a ready to use system available. We encourage researchers to start the software locally on their workstation using docker. ***Figure 10*** gives an overview of the components. With a local docker version, it is possible to view your own local datasets in a public web application like opengeoweb.

APPLICATION OF ADAGUC-SERVER AT OTHER PLACES

The Adaguc-server is re-usable for anyone in the need to make climate or meteo datasets available via webservices. To illustrate how this is done in practice, we provide an overview of known institutes and users using Adaguc-server.

OPENGEOWEB

The OpenGeoWeb software is an integrated system that is currently being developed in a joint effort of different groups in and outside KNMI to realize the next generation tooling for monitoring the atmosphere by forecasters, scientists and developers. The OpenGeoWeb consortium is currently a formal collaboration between the Royal Netherlands Meteorological Office (KNMI), the Finish Meteorological office (FMI), and the Norwegian meteorological office (MetNo).

OpenGeoWeb (see ***Figure 20***) provides an integrated system that will provide functionality for real time monitoring of meteorological phenomena, analysis of related georeferenced data and the production of warnings and forecasts. OpenGeoWeb's key advantage is its applicability for both operations and research, enabling fast deployment of research results into meteorological operations. At KNMI the WMS services for the geoweb application are offered via the Adaguc-server. Besides real time weather data, Adaguc-server can also facilitate research. Researchers can view their own local WMS services in the shared OpenGeoWeb frontend application. In combination with the autowms functionality, it becomes possible to visualize local data from a researcher's workstation in the geoweb application. In the geoweb application, the researcher is able to compare his data with existing operational data streams. Benefits of a web based meteorological working station is that it can be used at other places then the weather office as well.

PRIMAVERA DATA VIEWER (H2020 PROJECT)

The PRIMAVERA Data Viewer (***Figure 21***) is a visual prototype that explores some of the PRIMAVERA project results obtained from different Global Climate Models. The aim of the prototype is to illustrate the climate model results obtained from different models and resolutions. In the viewer a few well known climate indices are selected to illustrate the data. At the moment, the Data Viewer provides results from the past and the future. The viewer allows to compare different datasets with a vertical slider which can be dragged to the sides of the screen. This provides a convenient way of comparing different datasets.

The climate datasets are served using Adaguc-server by the PRIMAVERA project. The services are also used in story maps (originating from ESRI's ArcGISOnline), a

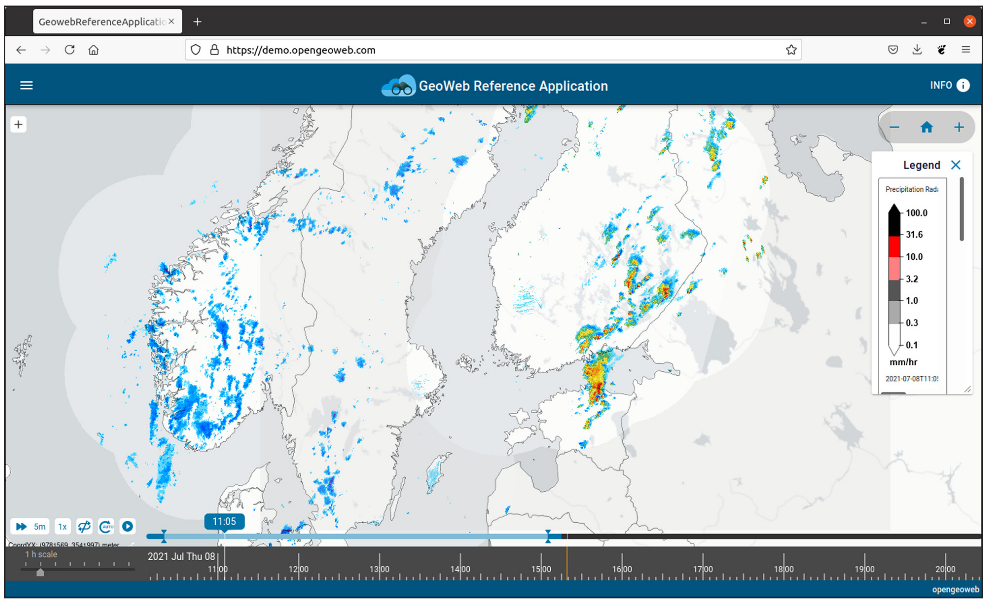


Figure 20 OpenGeoweb application showing real time precipitation data.

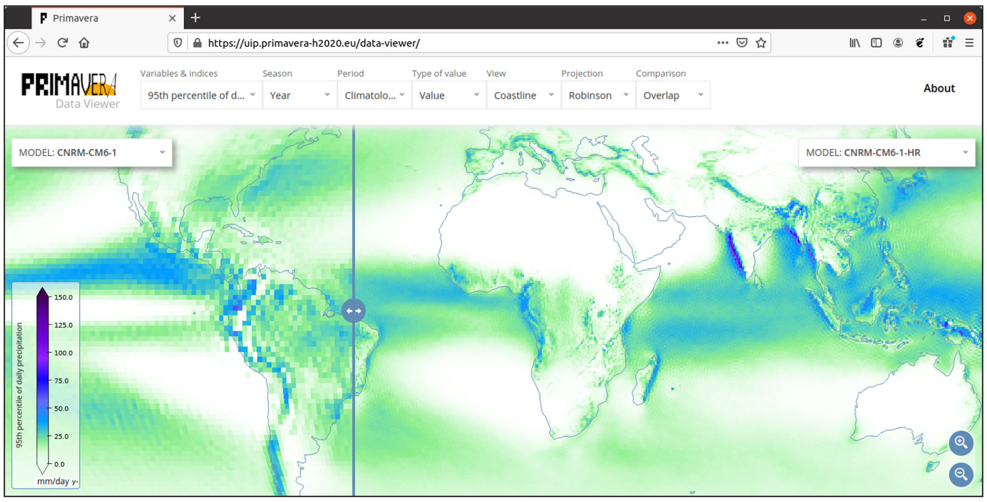


Figure 21 User Interface Platform of Primavera project (H2020), visualizations served via Adaguc-server. Web map application written with OpenLayers. *https://uip.primavera-h2020.eu/data-viewer*.

method to combine text and interactive maps to create inspiring and immersive stories. A link to the story maps from the Primavera project is available at the related software and sites sections.

CLIMATE CHANGE SCENARIO VIEWER FOR THE SPANISH CLIMATE CHANGE OFFICE

The climate change scenario viewer (*Figure 22*) is an easily accessible platform to view and download updated projections for future climate in Spain. The images are generated with Adaguc-server, the website makes use of leaflet. You can view historical data and future projections of climate changes for both observations and grids.

FIRE RISK VIEWER FROM PREDICTIA (SHOWCASE)

Predictia Intelligent Data Solutions is a private company in Spain making extensive use of the Adaguc-server. Predictia states the following on their site: "We have used this tool within several projects: the data viewer for the Spanish, the European project Primavera, with Madrid's Metro... It enables us to visualize climate data in a comprehensible and attractive way that is useful for the users of our tools. And it has an amazing community of developers and users around it that have been maintaining and expanding the project for more than 10 years." Examples of what they implement is for example a fire hazard viewer, making use of the stippling functionality, see *Figure 23*.

NWCSAF

The nowcasting Satellite Application Facility (NWCSAF) is using Adaguc-server and adaguc-viewer as a visualization tool to provide real time access to various satellite products. See *Figures 24* and *25* for examples.

ADAGUC AT METEO SPAIN (AEMET)

The meteorological institute of Spain (AEMET) is running the Adaguc-server and the Adaguc-viewer internally to display meteo information from various sources (*Figure 26*). They have integrated the display of Metars,

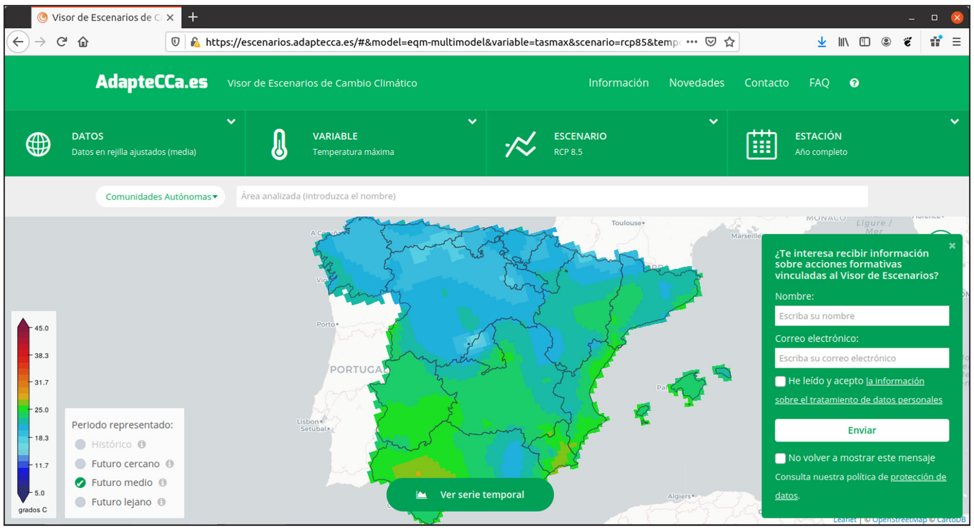

Visor de Escenarios de C
https://escenarios.adaptecca.es/#&model=eqm-multimodel&variable=tasmax&scenario=rcp85&temp
AdapteCCa.es
Visor de Escenarios de Cambio Climático
Información
Novedades
Contacto
FAQ
DATOS
Datos en rejilla ajustados (media)
VARIABLE
Temperatura máxima
ESCENARIO
RCP 8.5
ESTACIÓN
Año completo
Comunidades Autónomas
Área analizada (introduzca el nombre)
45.0
38.3
31.7
25.0
18.3
11.7
5.0
grados C
Periodo representado:
Histórico
Futuro cercano
Futuro medio
Futuro lejano
Ver serie temporal
PORTUGAL
Porto
Lisbon
Setúbal
Toulouse
Marseille
Algiers
¿Te interesa recibir información sobre acciones formativas vinculadas al Visor de Escenarios?
Nombre:
Escriba su nombre
Correo electrónico:
Escriba su correo electrónico
He leído y acepto la información sobre el tratamiento de datos personales
Enviar
No volver a mostrar este mensaje
Consulta nuestra política de protección de datos.
Leaflet | © OpenStreetMap © CartoDB

Figure 22 Climate change scenario viewer for the Spanish Climate Change Office – *https://escenarios.adaptecca.es/*.

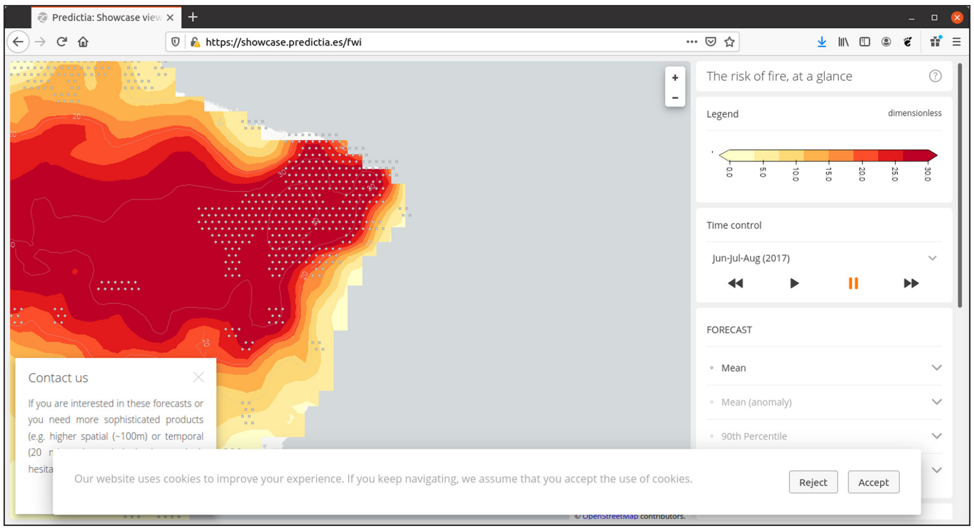

Predictia: Showcase view
https://showcase.predictia.es/fwi
The risk of fire, at a glance
Legend
dimensionless
0.0
5.0
10.0
15.0
20.0
25.0
30.0
Time control
Jun-Jul-Aug (2017)
FORECAST
Mean
Mean (anomaly)
90th Percentile
Contact us
If you are interested in these forecasts or you need more sophisticated products (e.g. higher spatial (~100m) or temporal
Our website uses cookies to improve your experience. If you keep navigating, we assume that you accept the use of cookies.
Reject
Accept
© OpenStreetMap contributors.

Figure 23 Fire hazard viewer with stippling applied, from *https://showcase.predictia.es/fwi*.

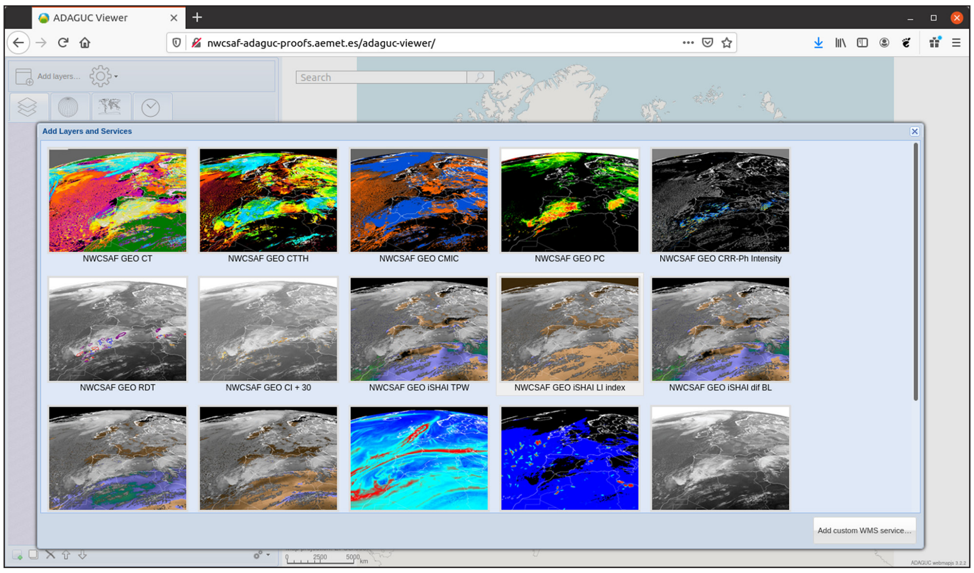


Figure 24 NWCSAF using adaguc-viewer, data selection menu has been extended with many products. From *http://nwcsaf-adaguc-proofs.aemet.es/adaguc-viewer/*.

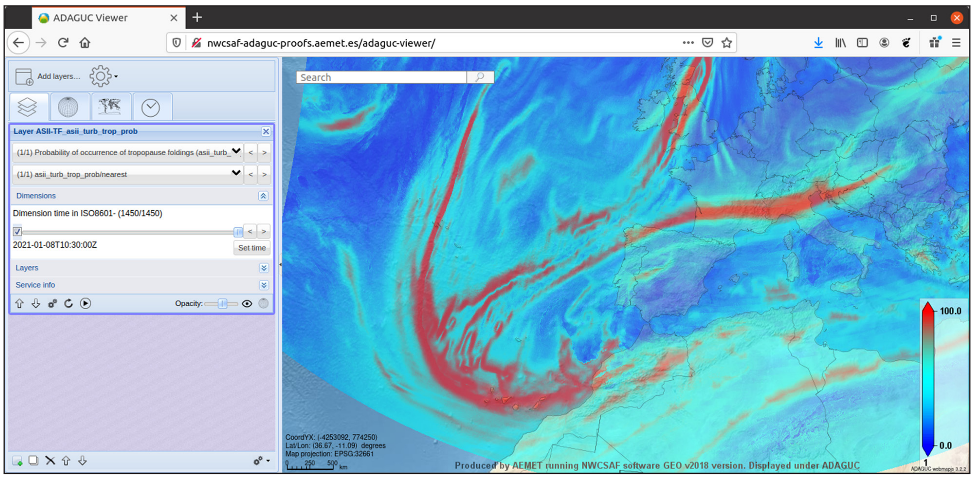

ADAGUC Viewer
nwcsaf-adaguc-proofs.aemet.es/adaguc-viewer/
Add layers...
Layer ASII-TF_asii_turb_trop_prob
(1/1) Probability of occurrence of tropopause foldings (asii_turb_
(1/1) asii_turb_trop_prob/nearest
Dimensions
Dimension time in ISO8601- (1450/1450)
2021-01-08T10:30:00Z
Set time
Layers
Service info
Opacity:
100.0
0.0
Produced by AEMET running NWCSAF software GEO v2018 version. Displayed under ADAGUC

Figure 25 NWCSAF displaying one of the live data streams in the adaguc-viewer. Here we selected “probability of occurrence of tropopause foldings”.

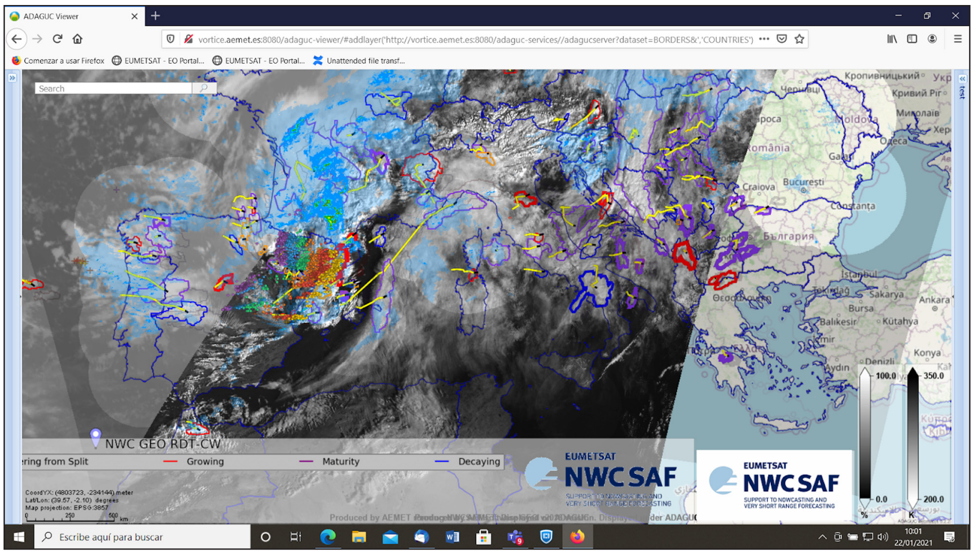

ADAGUC Viewer
vortice.aemet.es:8080/adaguc-viewer/#addlayer('http://vortice.aemet.es:8080/adaguc-services//adagucserver?dataset=BORDERS&','COUNTRIES')
NWC GEO RDT-CW
Growing
Maturity
Decaying
EUMETSAT NWC SAF
100.0
350.0
0.0
200.0
Escribe aquí para buscar

Figure 26 Adaguc displaying Satellite imagery from Meteosat (HRV, IR108, RDT-CW), Precipitation radars from OPERA and Lightning. A mix of polygons, points, and Netcdf CF is used in this application. Screenshot by AEMET. Contributors from NWCSAF, EUMETSAT, OPERA & NWCSAF2ADAGUC.

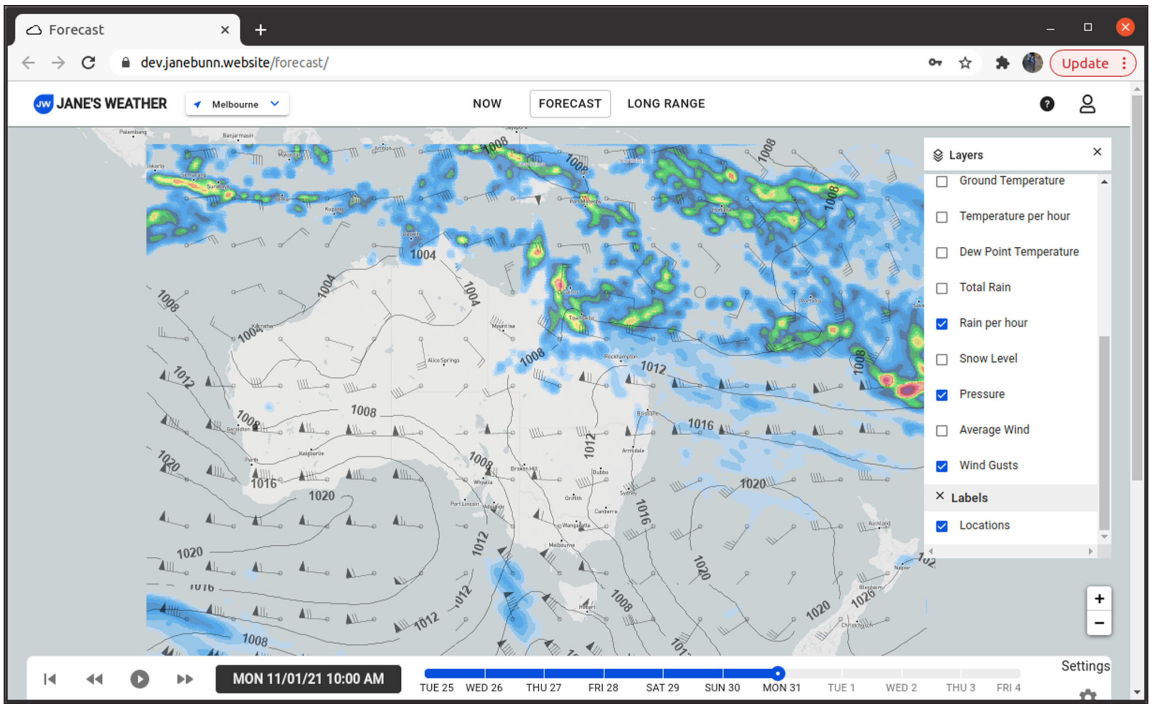


Figure 27 Jane's Weather - A website with weather and climate forecast. Visualization is done with the Adaguc-server. Contourlines, wind barbs and shading methods are applied.

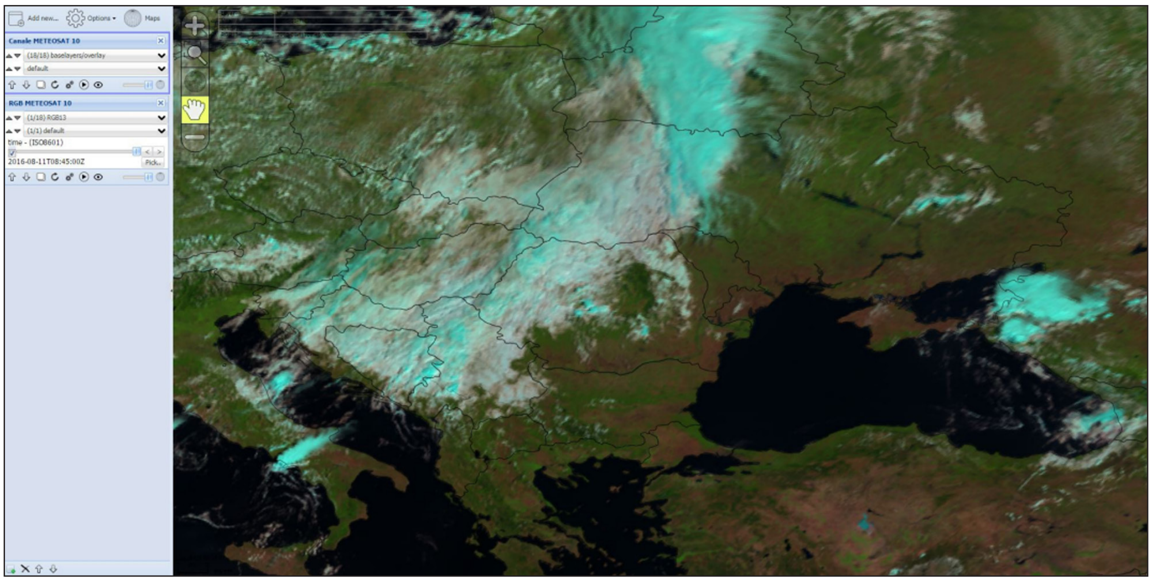


Figure 28 MeteoSat products visualization at NMA RO [11].

Radars, Satellite and Lightning. They combine data from Eumetsat and the NWCSAF in the same viewer.

JANE'S WEATHER – WEATHER AND CLIMATE FORECAST AUSTRALIA

A private company in Australia is building a weather and climate forecast viewer. You can see current weather, short term forecasts, and long term climate predictions (*Figure 27*).

APPLICATION AT METEO ROMANIA (NMA RO)

The National Meteorological Administration of Romania (NMA RO) uses Adaguc in their operational chain. It is used for products from the Now Casting Satellite Application Facility (NWCSAF) [11], Radar and MSG RGB/channels products [9].

NMA has implemented Land Surface Analysis products with Adaguc and is planning to develop other products such as SAF products visualisation (*Figure 28*).

ACKNOWLEDGEMENTS

Many thanks to Loes Cornelis and Friedrich Striewski for reviewing this document and providing valuable feedback. We also would like to thank Llorenc Liso, Xavier Calbet Alvarez, Pilar Rípodas, Oana Nicola, Andrei

Diamandi and Alina Ristea for sharing their work, participating in the workshops and applying the Adaguc stack at their institutes. Special thanks goes to Daniel San Martin from Predictia, you have applied the Adaguc-server in so many innovative applications that we lost count, it is a great inspiration for us. We also would like to thank Dave Hemming, thanks to you we have solved important issues inside the system and are thinking of interesting new features. We hope that we both benefit from this work and that we keep collaborating on these subjects in the future.

FUNDING STATEMENT

The software was funded by KNMI.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Maarten Plieger *orcid.org/0000-0002-7061-4070*
Senior developer GeoICT, KNMI, NL

Ernst de Vreede
Senior developer GeoICT, KNMI, NL

REFERENCES

1. **Blower JD, Gemmell AL, Griffiths K, Haines K, Santokhee A, Yang X.** A Web Map Service implementation for the visualization of multidimensional gridded environmental data; 2013. DOI: *https://doi.org/10.1016/j.envsoft.2013.04.002*
2. **Caron JL, Davis E, Ho Y, Kambic R.** "Unidata's THREDDS data server". 2006. *https://www.unidata.ucar.edu/software/tds/*.
3. **Cornillon P, Gallagher J, Sgouros T.** OPeNDAP: accessing data in a distributed, heterogeneous environment. *Data Science Journal*. 2003; 2: 164–174. DOI: *https://doi.org/10.2481/dsj.2.164*
4. **de la Beaujardiere, J.** OpenGIS Web Map Server Implementation Specification Version 1.3.0.; 2002; OGC® 06-042.
5. **Domenico B.** OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0; 2011. OGC 10-090r3.
6. **Gallagher J, Potter N, Sgouros T, Hankin S, Flierl G.** The Data Access Protocol — DAP 2.0; 2007. *https://cdn.earthdata.nasa.gov/conduit/upload/512/ESE-RFC-004v1.1.pdf*.
7. **Giuliani G, Chatenoux B, d. Bono A, Rodila D, Richard JP, Allenbach K, Dao H, Peduzzi P.** Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD); n.d. DOI: *https://doi.org/10.1080/20964471.2017.1398903*
8. **Little C.** e Meteorology and Oceanography Domain Working Group (Met Ocean DWG). 2018; *https://external.ogc.org/twiki_public/MetOceanDWG/WebHome*.
9. **Luca E, Nicola O, Craciunescu V, Andrei D, Ristea A.** The use of ADAGUC for MSG RGB products visualisation at the Romanian National Meteorological Administration; 2016.
10. **Nativi S, Caron J, Davis E, Domenico B.** Design and implementation of netCDF markup language (NcML) and its GML-based extension (NcML-GML); 2005. DOI: *https://doi.org/10.1016/j.cageo.2004.12.006*
11. **Nicola O.** Using ADAGUC for NWCSAF products visualization at NMA – a test case; 2015. *https://www.nwcsafrg/emetWebContents/WorkshopsSurveys Training/Workshops/2015UsersWorkshop/2015UsersWSPresentations/SESSION_VII/NWCSAF_Workshop_2015_Oana_Nicola_ADAGUC.pps*.
12. **Rew R, Davis G, Emmerson S, Davies H, Hartnett E, Heimbigner D, Fisher W.** The NetCDF data model; 2021. *https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_data_model.html*.
13. **Robinson D.** The Common Gateway Interface (CGI) Version 1.1; 2004. *tools.ietf.org*. DOI: *https://doi.org/10.17487/rfc3875*
14. **Wikipedia contributors.** Nearest-neighbor interpolation; 2017. *https://en.wikipedia.org/w/index.php?title=Nearest-neighbor_interpolation&oldid=763429489*.
15. **Wikipedia contributors.** Bilinear interpolation; 2021. *https://en.wikipedia.org/w/index.php?title=Bilinear_interpolation&oldid=1018246009*.
16. **Wikipedia contributors.** Digital elevation model; 2021. *https://en.wikipedia.org/wiki/Digital_elevation_model*.
17. **Wikipedia Contributors.** Normal Mapping; 2021. *https://en.wikipedia.org/wiki/Normal_mapping*.
18. **Wikipedia Contributors.** Terrain cartography; 2021. *https://en.wikipedia.org/w/index.php?title=Terrain_cartography&oldid=1020923614*.
19. **Wikipedia contributors.** Top left lighting; 2021. *https://en.wikipedia.org/wiki/Top-left_lighting*.
20. **Winkle LV.** Interpolating in a Triangle; 2016. *https://codeplea.com/triangular-interpolation*.

RELATED SOFTWARE AND SITES

PostgreSQL – Relational database: *https://www.postgresql.org/*.
Sqlite- File based relational database: *https://www.sqlite.org/*.
Proj – Geographical projection library: Proj *https://proj.org/*.
NCWMS – Web Map Service implementation for displaying environmental data: *https://reading-escience-centre.github.io/ncwms/*.
KNMI Klimaatatlas – Site with the climate normals for the climate in the Netherlands: *https://www.knmi.nl/klimaat-viewer*.
CLIPC – The CLIPC project from the European Union's Seventh Framework Programme *http://www.clipc.eu/*.
Tomcat – Java based application server: *http://tomcat.apache.org/*.
Apache httpd – Open-source HTTP server: *https://httpd.apache.org/*.

OpenLayers – OpenLayers for dynamic maps in any web page: *https://openlayers.org/en/latest/examples/wms-image.html*.
Leaflet – JavaScript library for interactive maps: *https://leafletjs.com/*.
AHN – Algemeen Hoogtebestand Nederland (AHN): *www.ahn.nl*.
Himwari – Japanese weather satellite: *https://himawari8.nict.go.jp/*.
MSGCPP – Meteosat Second Generation Cloud Physcial Products: *http://msgcpp.knmi.nl/*.
QGIS and WMS – Displaying WMS in QGIS: *https://www.qgistutorials.com/nl/docs/working_with_wms.htm*.
Adaguc-viewer – JavaScript application to display WMS with environmental datasets: *https://github.com/KNMI/adaguc-viewer*.
GeoWeb-core - A JavaScript library for interactive maps: *https://gitlab.com/opengeoweb/geoweb-core*.
H2020 Primavera – *https://uip.primavera-h2020.eu/data-viewer*.
H2020 Primavera storymaps – *https://uip.primavera-h2020.eu/storymaps*.

TO CITE THIS ARTICLE:
Plieger M, de Vreede E 2021 Adaguc-Server: Interactive Access to Heterogeneous Meteorological and Climatological Datasets Using Open Standards. *Journal of Open Research Software,* 9: 33. DOI: *https://doi.org/10.5334/jors.382*

Submitted: 24 June 2021 Accepted: 16 November 2021 Published: 09 December 2021

COPYRIGHT:
© 2021 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See *http://creativecommons.org/licenses/by/4.0/*.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.