AdriaClim

Climate change information, monitoring and management tools for adaptation strategies in Adriatic coastal areas

Project ID: 10252001

D.4.1.1 Report on system architecture design and technological stack

LP – Arpae

Final version

Public document

Table of contents

1. Aims and content of the document

This deliverable describes the architecture of the AdriaClim Information System as agreed during the first part of the project. This will drive the subsequent phase of the project, in which the system will be implemented and deployed.

2. Overview of the system architecture

The AdriaClim Information System consists mainly of a number of **big-data repository nodes**, hosted by different institutes being project partners, and of a **geoportal**, acting as an interface between the big data repository and the final users of the project outcome. The big-data repositories will host the output of model simulations performed within the project, climatic indicators based on these simulations, and possibly other indicators as well as observational data. Another component of the system is the Climate Literacy Toolkit, which is a separate divulgative platform which will however make use of the services provided by the geoportal (e.g. access to data and graphics, user registration and authentication).

3. The Big-data repository nodes

The internal architecture of the big-data repository nodes is described in detail in deliverable 4.3.1. Here we will just briefly report that the repository will be distributed among a number of different physical servers hosted by some institutes being project partners. The standard repository nodes are based on the ERDDAP server software (https://coastwatch.pfeg.noaa.gov/erddap/index.html) from NOAA (USA) and they will be joined together in a so-called Erddap Federation, so that they will appear as a single node from outside, e.g. from the point of view of the geoportal. Thanks to the interoperability capabilities of ERDDAP, other data archives based on networked technologies such as OPeNDAP, SOS or WMS could be interfaced by an ERDDAP server and join the Adriaclim Erddap Federation almost transparently.

The ERDDAP system consists of an archive (back end) and an http front end. The front end provides many services for access to raw data, but it is limited in terms of visualisation of the selected data and has also limited capabilities for authentication.

For this reason, the data stored in the distributed nodes will be mainly available through the geoportal, which will provide a more user-friendly interface, customised to the Adriaclim needs for data access and visualisation, and an authentication management system. However, the institutes

hosting the single nodes may decide to make the repository public and/or define additional local credentials for granting to specific users direct access to the data.

4. Climate impact indicators

Within the AdriaClim project, following an accepted classification, the relevant indicators have been subdivided into the following categories: “Indicators for changes in the climate systems”, “Indicators relevant for climate change impacts” and “Indicators relevant for adaptation”. Indicators of the first category are mostly based on hydrometeorological data, usually produced by Earth system models, thus they are suitable for being computed on the basis of datasets of climatic simulations. Indicators of the second and third categories are more specific to some particular natural or economic sector and require the knowledge of specific statistical data for their computation, in addition to model datasets or indicators, for obtaining a projection in the future.

Indicators of the first category, that will be computed on the basis of model datasets available on the Adriaclim platform (regardless of whether they will be produced within the project or externally), can be further divided into the following categories:

1. Indicators as a single, time-dependent global value

e.g. Increasing rates of global mean sea-level rise

2. Linear indicators, computable as a time series of gridded fields, point-by-point, and suitable for an a posteriori geographical subsetting

e.g. Atmospheric water vapour

3. Nonlinear indicators, whose operator does not commute with the area-averaging operator

e.g. Number of summer nights with T_{min} > 20C°.

4.1 Collection and distribution of indicators computed outside WP4

Since some of the “Pilots areas” defined in the project have already developed own methodologies for computing climatic indicators, the project strategy for collecting the indicators will likely follow these steps:

- distribute new model datasets allowing Partners to apply their indicator-computing methodologies to these datasets
- collect the indicator datasets already existing at each Pilot or computed on the basis of the new Adriaclim model datasets

- classify and homogeneize the indicators received, associate proper metadata, making them suitable for being stored on the Adriaclim platform repository
- store the indicators in the available repository nodes and redistribute them to the partners in order to allow the subsequent computation of more specific indicators
- collect the new indicators computed and iterate the process of distributing data and collecting new indicators, if needed.

It will be appreciated if, especially for the more generic indicators, such as those classified as "Indicators for changes in the climate systems", the computation done by each Pilot area will be extended also to the geographical areas belonging to other Pilot areas, or, in general, to all the area covered by the available data, if this extension will have only marginal additional costs. This will allow an exchange of data between partners and avoid double work.

4.2 Computation of indicators within WP4

If the above indicators are not considered enough for the project purposes, an indicator computing engine will have to be implemented within the task 4.2 of WP4, performing computation of indicators on the basis of the Adriaclim model datasets stored in the distributed repository nodes. Due to limitation in the computing power available at the repository nodes, the computation will not be performed online at each user request, but it will be performed once for all at the time of publication of each dataset (and repeated later, if necessary, for a limited number of times), on a reasonable set of geographical areas, if applicable, and stored in the repository in a proper dataset, together with the model dataset(s) that generated them. In the absence of a High Performance Computing system, this computation may require long times, but it guarantees that the data will be timely available when users will request them and they will be suitable for an interactive use.

For implementing the computation of these indicators we can identify the following subtasks:

1. Computation of the indicators
    - 1.1. Evaluate the list of suitable indicators.
    - 1.2. For each indicator:
        - 1.2.1. define the input data (e.g. type of input model, variables, vertical levels, time frequency) and express this list in terms of a data query on the Adriaclim repository
        - 1.2.2. define the output (indicator) data structure and associated metadata
        - 1.2.3. implement the computing algorithm, fed by the data query defined above

1.2.4. code the output result according to the rules defined above (data structure and metadata)

There are no limitations on the computer languages to be used for the implementation of the computation, but it must be kept in mind that the system must be easily reproducible by means of a standard installation of open source software available on public GNU/Linux repositories, plus some specifically developed software, and the computation should be executable in a non interactive way by a script.

2. Packaging of the system

2.1. Implement a script for integrating the indicator computing process for a single model dataset. The script should receive in input the dataset(s) to use, the list or the group of indicators to be computed, it should launch the computation optimising the local resources (CPU, memory), collect the output data and archive them in the proper dataset(s) on the local repository node, managing the possible error conditions (e.g. some indicators may not be computable due to insufficient input data, etc.).

2.2. Implement a local user interface (e.g. command-line or simple web interface) available to operators, to execute the computation script when a new model dataset becomes available on the local repository node.

2.3. Create a container/virtual machine including all the indicator computing software stack.

All the additional software for computing the indicators and for packaging the system specifically developed within the project should be made available on a public repository with an open source licence which guarantees its free redistribution.

4.3 Additional guidelines regarding Indicators for changes in the climate systems

Due to the distributed nature of the data repository, it has been decided that the Adriaclim platform will provide only precomputed indicators and not tools and algorithms for computing them online; linear indicators could, in principle, be computed and archived on the same grid as the input dataset and space-aggregated on user request at the time of retrieval, while nonlinear indicators need to be space-aggregated once for all on all the expected user areas at computation time and stored as a sparse point dataset.

The indicators based on a specific model dataset should be ideally grouped into a single dataset of indicators (or more than one if required by the topological nature of different indicators, e.g. gridded, sparse point, etc.). The metadata associated with each indicator should reference the model dataset(s) they are based on and should contain all the information needed for identifying

unambiguously the algorithm used, the indicator time and space extent and any other required information.

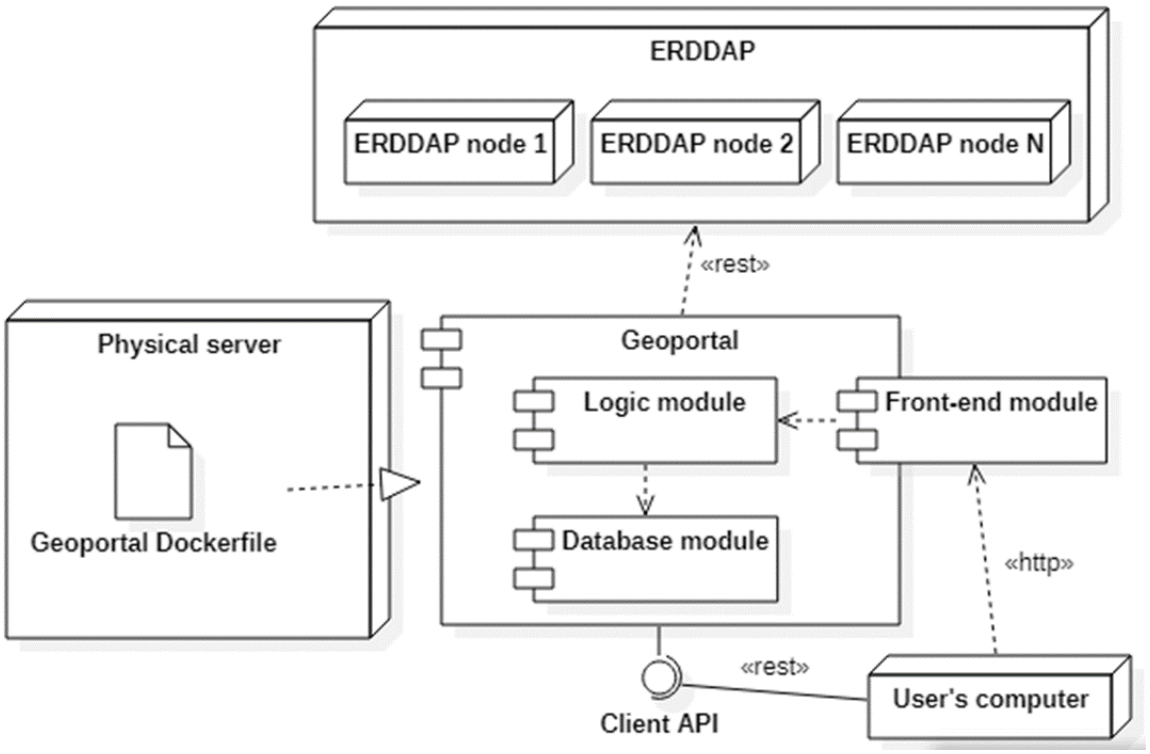
5. The Adriaclim Geoportal

The geoportal is the web application that will expose the output datasets of Adriaclim work packages in such a way as to provide maximum benefit to users and decision makers. The potential output of Adriaclim WPs consists of hundreds of datasets and indicators, each comprising a large amount of gridded or ungridded data such as polygonal shapes used frequently for planning purposes and contained in formats such as SHP (shape) or TIFF. It is only through abstraction and presentation that data can become information and then knowledge.

The geoportal architecture necessarily stems from its underlying mission. This mission can be summarised as follows:

- Abstraction of big data ERDDAP nodes, providing the user with a single point of entry to all project datasets and removing the need to interact directly with a specialist domain tool such as ERDDAP.
- Visualisation and simplification, exposing a coherent interface that hides low-level minutiae and allows users with non-technical backgrounds to have intuitive access to all project data. ERDDAP exposes an enormous amount of project data; if these data are simply offered as-is, the user can feel lost and struggle to find value.
- Performance and automation, delivering methods for advanced users to filter and download the data with the full power and options provided by ERDDAP.
- Technology support. The geoportal provides a layer with technology missing from ERDDAP, such as customised authentication and additional rendering strategies.

5.1 Deployment structure



The geoportal interacts with an ERDDAP federation by querying its master node through calls to its REST APIs. The master node in turn is a repository for all datasets present in the federated nodes, making it the single point of access to the collection. The geoportal, as far as ERDDAP is concerned, is a user like any other, and it can be the system's only user if the appropriate steps are taken to block the federation from outside access. As long as the presence of other users and roles does not affect the geoportal' ability to access the metadata, it does not need to have exclusive access to the federation. In fact, the dispatcher model as described in the next section relies on users having direct access to the ERDDAP federation.

The geoportal itself is deployed out of a Docker image file and consists of:

- a logic module, written in Python using the Django framework, which contains all the server-side functionality required to drive the system: this includes user management, proxy features to ERDDAP, serving web content to clients, and interacting with the database module.
- a database module, which is deployed from the Docker image of a PostgreSQL installation. This stores all geoportal-specific data, such as user information (since geoportal users are

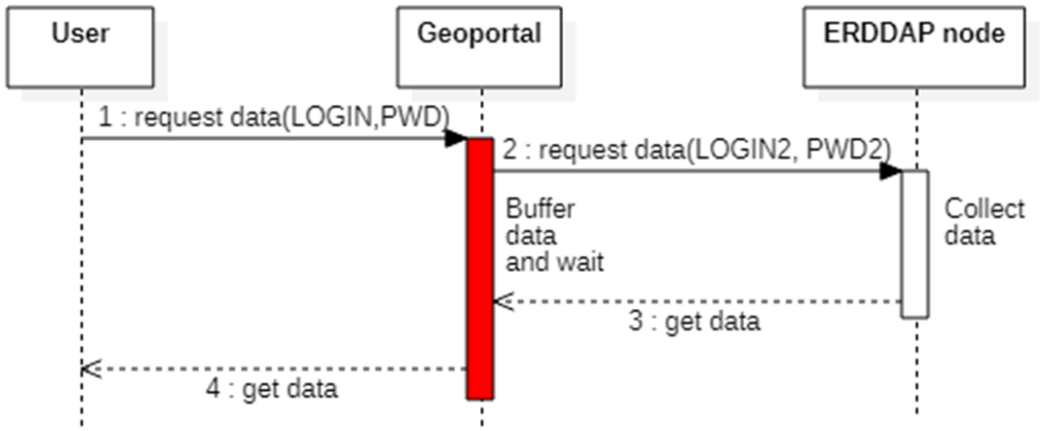
not the same as ERDDAP users), any cached data that may prove necessary for performance reasons, configuration and environment-specific settings.

- a front-end module, written in Javascript with HTML and CSS files. This is the actual user interface that will run on all modern browsers and uses robust technology such as Leaflet to display WMS and vector data on a map.

5.2 Interaction with the repository nodes

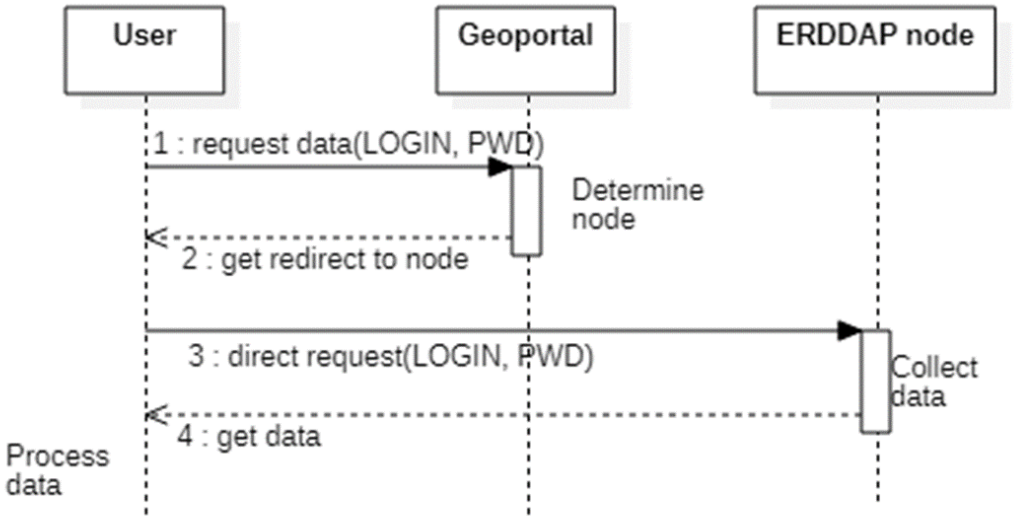
Because the ERDDAP repositories host a very large amount of data, it raises the important question of how that data is going to reach the end user. There are basically two approaches to this data flow, and either one can be used exclusively or they can be mixed and matched depending on circumstances.

The proxy model



In this model, the user never has any interaction with the ERDDAP node: every request is mediated by the geoportal, which also collects all the data queries, forwards them to ERDDAP, and provides the result. It is architecturally simple, but puts more strain on the geoportal's network infrastructure. Under this model, no user is required on the ERDDAP nodes except a single one for the geoportal itself. The geoportal manages users and roles locally, if necessary limiting access to specific resources to certain roles based on agreed-upon metadata stored with the datasets.

The dispatcher model



Following this model, the geoportal does not directly query the ERDDAP node, but instead constructs an URL for the client to perform the query with its own resources. This frees up network and computation resources for the geoportal, but requires one of the following: either that all data on ERDDAP be public and accessible by anonymous users, or that geoportal users be synchronised with ERDDAP users through a periodical batch activity.

5.3 Authentication and authorization

It has been noted that ERDDAP authentication strategies are either not secure enough (password hash) for modern standards or too inconvenient (google ID / orcid) for general user access in AdriaClim. For this reason authentication will be handled by the geoportal, providing a user/password scheme in compliance with current security regulations.

At least two user roles will be present on the system: user and administrator. If specific datasets require finer-grained access to ERDDAP resources, they will be marked as such in their metadata and the geoportal will honour those restrictions (if in proxy mode), or the datasets will be made physically unavailable to the specific user in ERDDAP (if in dispatcher mode).

5.4 Visualising gridded data

Gridded data will have at least two display modes.

- First, taking advantage of ERDDAP's built-in ability to output WMS data, the geoportal plugs directly into a dataset and renders it as a layer. This is the easier way of displaying data, but it can only be as advanced as ERDDAP's own WMS capabilities, which are quite basic. In

particular, ERDDAP maps all datasets onto the same linear spectrum of colours, regardless of value distribution, and offers no ability to customise or style a layer. Also, it does not support publishing vector-valued datasets as WMS layers.

- Secondly, the logic and front-end modules of the geoportal could provide an alternate data rendering experience. In particular, the logic module queries ERDDAP, potentially joining the results of multiple queries (e.g. for vector data in which dimensions are separated into multiple variables), and delivers it to the front-end, which translates it into styled geometry to be rendered on screen. This can, for example, visualise directional wind or current data. It could also be used to display scalar datasets that render poorly under ERDDAP's WMS, by rendering them as grids of colored rectangles.

5.5 Visualising point data (indicators)

For one-dimensional data, i.e. data without geographical coordinates and therefore difficult to show on a map, the best approach is to have a separate panel with a list of such datasets. When queried, it will show the same modal window that is used to analyse the time series for gridded data at a specific location, and will provide a time graph of the data, tabular values as well as export options (mapping ERDDAP's own exporting capabilities.)

After having collected the end users requirements, this visualisation can be enhanced by integrating suitable data analytics packages in the front end.