



ARKTALAS HOAVVA PROJECT

DELIVERABLE D-40: DESIGN OF THE ARKTALAS VISUALISATION AND ANALYSIS SYSTEM (AVS)

Customer	ESA
Author	Sylvain Herlédan, OceanDataLab
Distribution	Consortium and ESA
ESA Contract Number	4000127401/19/NL/LF
Document Reference	
SoW Deliverable Reference	D40
Version/Revision	1.2
Date of issue	2 July 2020

Approved by (NERSC)	Johnny A. Johannessen NERSC Project Manager	
Approved by (ESA)	Craig Donlon ESA Technical Officer	

Revision Change log

Issue	Date	Type	Change description
0.5	31 December 2019	Draft	Initial version
1.0	7 January 2020	1 st final version	Revised
1.1	9 January 2020	2 nd final version	Captions added to figures
1.2	2 July 2020	Final	<p>Update section 2.1 “Crawler for EO data”: the AVS will process data from a spool directory that is fed by the ADAS.</p> <p>Update section 3.3 “ADAS integration”: download links will be implemented as search queries to the ADAS.</p> <p>Annex 1 added to describe work progress and plan as requested during PM-2</p>

This Deliverable D-40: Technical Note (TN) is associated with Task 2b: Implementation of Analyses and Visualization System. It follows the structure and outline indicated in the Statement of Work (SoW) and the Arktalas Hoavva technical project proposal. The approval of this deliverable by the Agency will trigger the development of the Analyses and Visualization System and its deployment on the Arktalas infrastructure.

Table of Contents

1 Overview	3
2 Backend	5
2.1 Crawler for EO data	5
2.2 Processing chain.....	6
2.3 Web service.....	7
3 Frontend	9
3.1 Main features	9
3.2 Deployment.....	9
3.3 ADAS integration	10
Annex 1: Work progress	11

1 OVERVIEW

The goal of the Arktalas Hoavva Analysis and Visualisation System (AVS) is to display data representations at full resolution on an interactive map for the data sets stored in the Arktalas-Data Archive System (ADAS).

In order to achieve this objective, the AVS provides an intuitive web interface to browse and display the EO data sets collected for the Arktalas case studies, offering a graphical support for project members and external users to discuss their results, explore the synergy between available products and potentially discover new interesting cases.

The implementation of the AVS is based on the Syntool software suite developed by OceanDataLab, as it provides all the features required for rendering EO data on an interactive map integrated in a website.

As shown in **Figure 1**, the software suite is composed of:

- a web application which embeds an interactive map and controls to select products, dates, times and areas in an intuitive manner.
- a web service for searching data that match a set of criteria, using a database which contains the spatial and temporal coverage of all data available in the system.
- processing tools that transform EO data into web-compatible representations and extract the metadata required to populate the aforementioned database.

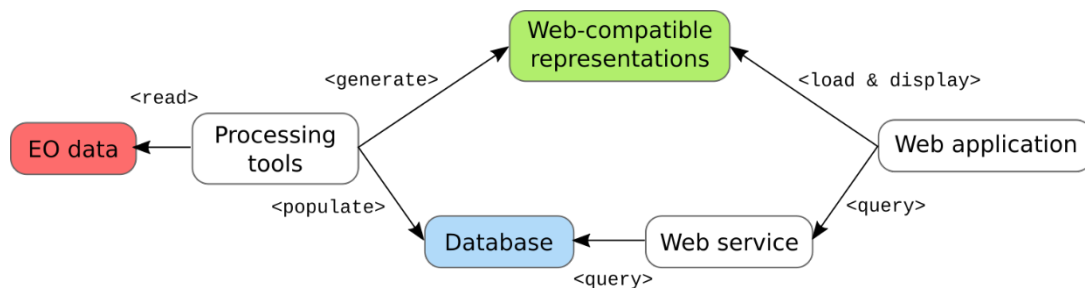


Figure 1: High level overview of a Syntool architecture

The web service and web application are already developed and can be deployed at any time, only requiring some configuration to define the list of products available in the data portal and to customize the options offered to users.

The web application features a mechanism to save its selected date, time, viewport and products in the form of a simple web link that can be embedded in an email, an article or a message in social media. Sharing these links allows other users to open the web application in the exact same state and can therefore be used as an interactive illustration for an article or as a graphical support during discussions with other members of the scientific community.

It also provides a way to create local (but shareable) bookmarks so that the web links appear with a descriptive title directly in the application. Bookmarks (or "hotspots" as they are called in Syntool) can be pre-configured, which could be used to give a quick access to cases of interest for the Arktalas studies.

Processing tools are the only components of the Syntool software suite that interact directly with EO data. Although these tools are already able to handle a variety of

products, some development will be needed to implement support for the Arktalas data sets: ad hoc data readers must be written to handle the specificities of these data sets so that they can be converted into a common format which can then be used as inputs by generic tools that intervene downstream in the processing chain.

The Syntool web application is also able to connect to several web services, which grants the possibility to distribute data among several hosting platforms if they allow public access to the web-compatible representations and to the Syntool web service.

In the context of the Arktalas project, this mechanism has several advantages:

- It removes the need to have all the data hosted in a single infrastructure, therefore reducing the amount of data transfers between partners and a central authority, also allowing several partners to participate actively in the data gathering process.
- It makes the AVS more resilient to failures, as the data portal will still be able to run in a degraded mode in case one of the hosting platforms is offline as data from the other platforms will still be available.
- It allows the AVS to grow in the future as connecting to new instances of the Syntool web service only requires minor changes in the configuration, making this solution both extensible (web services providing new data) and scalable (distributing data among a larger number of Syntool web service instances), as illustrated in **Figure 2**.

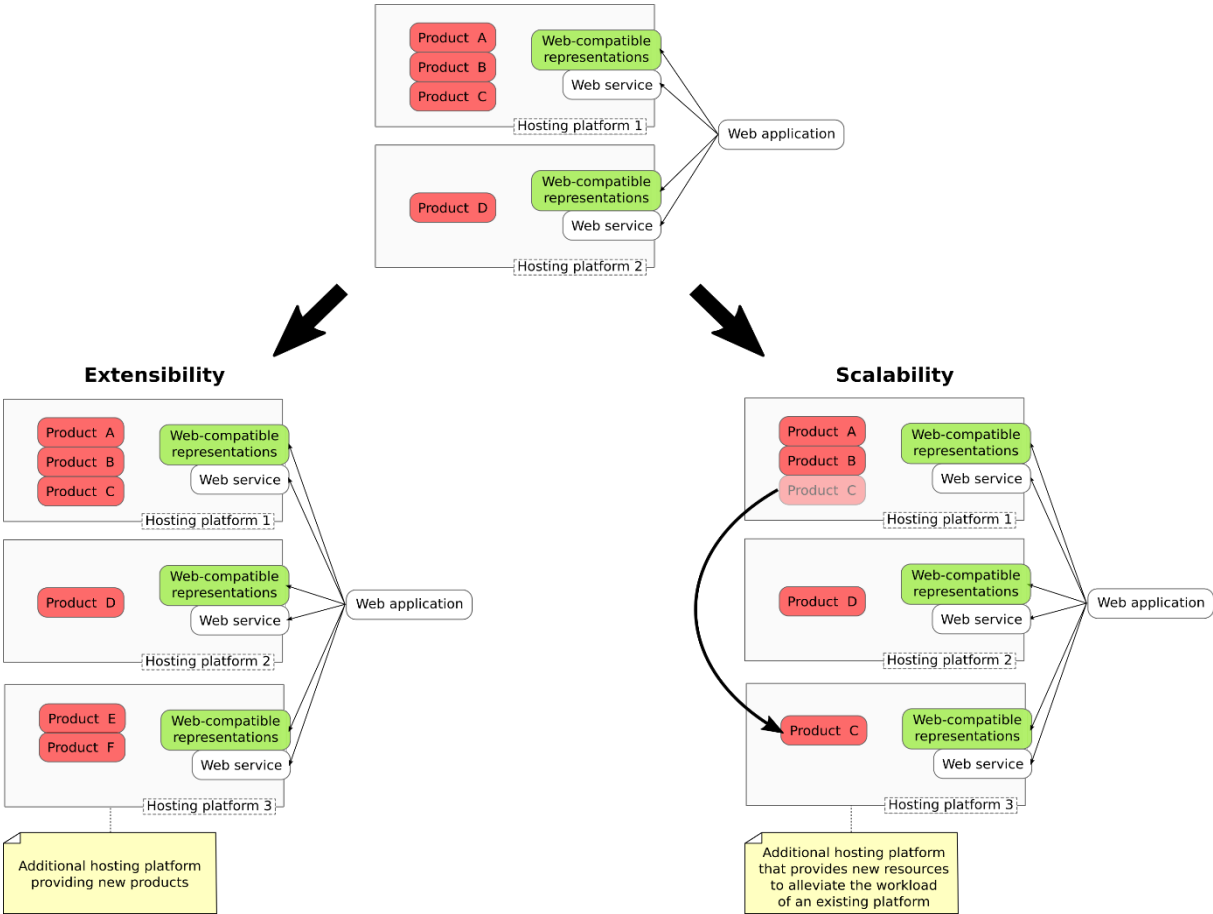


Figure 2: Extensibility and scalability of the system is achieved by integrating new backends

2 BACKEND

The backend of the AVS is composed of three main parts:

1. A crawler that will find the EO data (in raw format) to include in the AVS
2. Processing chains to produce web-compatible representations and extract metadata from EO data in order to populate the Syntool database
3. A Web service that can be queried from the Internet

Only the third part has to be permanently online: the AVS processes delayed-time data archives downloaded by the ADAS for temporal windows identified as being of interest for the case studies, not a permanent incoming feed of data. Processing resources are only required temporarily when the ADAS completes a download batch and resources that must be allocated permanently are thus limited to:

- storage for the web-compatible representations generated by the processing tools
- a database for Syntool metadata (MySQL database which can be mutualized with other services)
- a host (physical or virtual) for the web service
- a web server (which can be mutualized with other services) to expose the service endpoints and the web-compatible representations on the Internet

2.1 Crawler for EO data

The AVS backend will be installed on the same hosting platform as the ADAS, so finding the files that should be passed to the Syntool processing tools and accessing their content should be a straightforward task.

The ADAS is responsible for launching the production of web-compatible representations for the AVS.

Newly downloaded files will be stored in a spool directory and a crawling process will be executed periodically (CRON job) to scrutinize the content of the spool to detect files that need to be sent to the AVS.

As the ADAS manages the content of this directory, the AVS tools will only read files from it and won't delete them even after they have been processed.

It is therefore possible that the spool directory contains files that have already been processed for the AVS.

In order to differentiate between files that have been processed and files that have yet to be transformed into web-compatible representations, the crawler will keep track of the files that have already been used as inputs for the AVS processing chain simply by appending the path of these data at the end of a listing file.

The crawler will be implemented as a set of simple Shell (ksh, bash, etc...) scripts that produce a listing of files that will be passed as inputs to the Syntool processing chain.

2.2 Processing chain

The processing tools included in the Syntool software suite have been implemented for Python 2.7 and rely on several well-known libraries to handle map projections (Proj4, GDAL).

A Syntool processing chain is composed of three main tools:

- the `syntool-converter` command opens the EO data files using plugins (data readers) to support the native format of these files, extracts relevant information from their content and saves them in a format (GeoTIFF or netCDF depending on the situation) that the other tools are able to manipulate with ease.
- the `syntool-ingestor` command reads information from the converted files and produces web-compatible data representations as well as a file which only contains metadata (spatial and temporal coverage notably).
- the `syntool-meta2sql` command transforms the metadata file into a sequence of SQL statements that will be executed on the Syntool database to register the results of the processing chain and let the web service know that new data are available.

A typical workflow based on Syntool processing tools is represented in **Figure 3**.

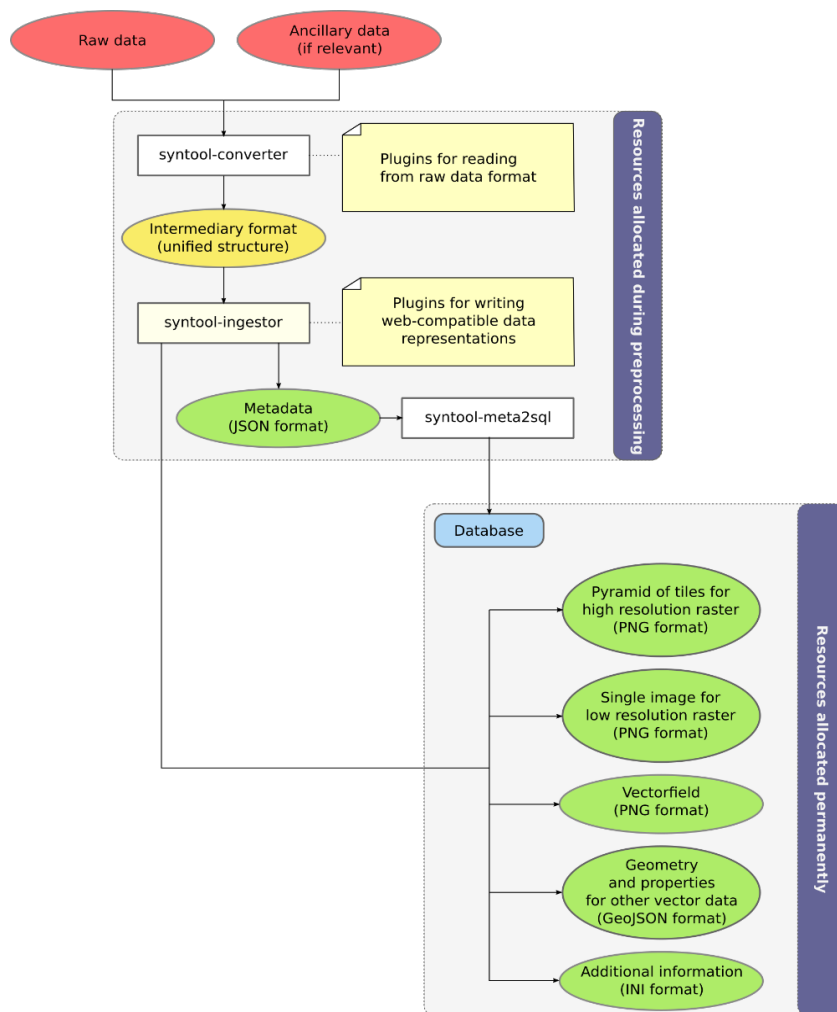


Figure 3: Workflow of a Syntool processing chain

The web-compatible representations produced by syntool-ingestor depend on the nature of the EO data and a single input file can be used to generate different representations if deemed useful.

For example, vector fields can be ingested as a raster with colors that represent the module of the vectors to focus on the amplitude of the geophysical phenomenon, but also as arrows so that directional information is also available for visualization.

The orchestration of this processing chain can once again be implemented using simple scripts that loop on the file paths listed by the crawler and run the sequence of commands.

2.3 Web service

The Syntool web service is a WSGI application (see <https://www.python.org/dev/peps/pep-3333/>) written for Python 2.7 or above (including Python 3.x), with minimal dependencies.

Python applications that follow the WSGI convention can be called by web servers to handle requests and produce responses dynamically: the Syntool web service extracts search parameters from the web request to query the AVS database, filter and format the results as a web response before sending it back to the web server which can then transfer it to the web client.

Application containers (also called "application servers") are middleware components that can be used to ease the management of web services. They act as wrappers that implement mechanisms that most applications require, such as setting environment variables, logging or the ability to spawn new application instances when the number of requests increases. uWSGI is a widely used application container that supports WSGI web services and provides many options to tune the behavior of the applications it manages.

The organization of the Web part of the AVS backend is described in **Figure 4**.

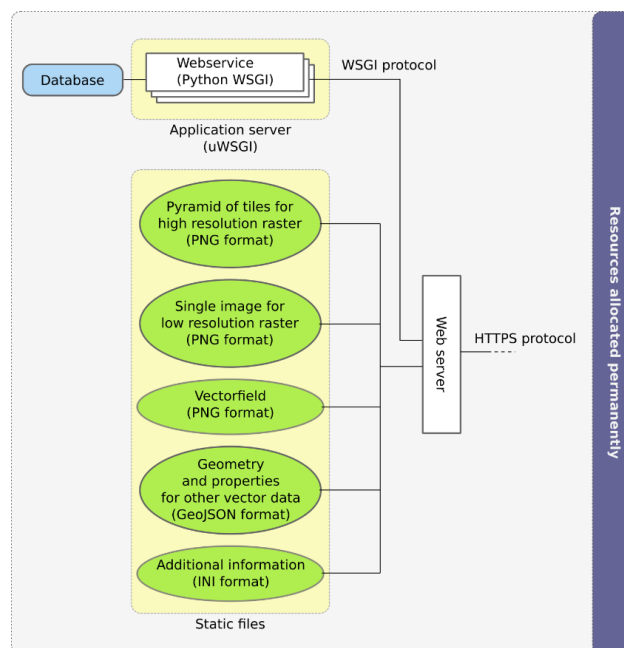


Figure 4: The AVS backend provides access to data representations and to the Syntool web service from the Internet

The web service provides three endpoints that the Syntool application requires to work properly:

- /coverage: given a list of product identifiers and a geographical area, this service returns the daily availability of each product within the specified area, encoded as a binary blob.
- /datasets: this service returns the metadata of all the registered EO data that match a filter composed of a product identifier, a geographical area and a time range.
- /getGranuleInfo: this service provides additional information related to a specific entry in the database (i.e. a granule). Only applicable if such information exists.

These three services only read information from the database, they don't create nor modify its content: granting a read-only access on the database to the Syntool web service allows it to work as intended and prevents security issues such as SQL injections. This setup has no downside and is therefore the recommended way to configure database access for the Syntool web service.

3 FRONTEND

The Syntool frontend is a web site composed of three main elements:

- an interactive map that users can zoom and pan: its viewport is used as geographical filter to find the relevant EO granules.
- a list of all the products available in Syntool, with checkboxes to select the ones that must be displayed on the map.
- a timeline which allows users to select a time range to filter to EO granules that must be displayed on the map and provides an intuitive way to modify this time range in order to browse the content of the AVS.

3.1 Main features

The Syntool web application relies on the OpenLayers JavaScript library for the map component, with some customization to improve its performance and add support for the variety of data representations that syntool-ingestor can produce.

The map is able to render several overlapping EO granules at the same time, with settings to change their transparency as well as the order in which they are rendered so that users can see the details provided by each of these granules to build a more complete representation of a geophysical phenomenon (synergy between products).

The timeline not only serves as a date/time selection control but also as an indicator for data availability: after each change in the products list or in the map viewport, the application calls the `/coverage` endpoint to update the content of the timeline component and to clearly show the days/months/years when data are available for the selected products over the visible area.

Details about geophysical variables, product version, algorithms applied during processing, flags used to mask invalid data, provider, etc... can be associated with each product and displayed either as a tooltip or a popup window that appears upon hovering or clicking on an item of the products list.

After each change of the map viewport, the selection in the products list or in the timeline, the application calls the `/datasets` endpoint to list all the EO granules that match the new set of constraints, build their footprint on the map and load the associated representation.

Selecting a granule by clicking on its footprint on the map triggers a call to the `/getGranuleInfo` endpoint: if additional information is available for the selected granule, it will be displayed in a contextual menu. This information can take several forms: it may be a simple text, an image or a link to an external resource.

3.2 Deployment

The Arktalas instance of the Web application does not have to be deployed on the same infrastructure as the AVS backend: all the processing work has been performed upstream and the results are accessible from the Internet so there is no restriction regarding the location of the Web application.

As this website is only composed of a few static files, the resources required to serve this application are minimal and any consumer-grade computer connected to the Internet would be suitable as long as it hosts a web server and has enough bandwidth.

The configuration of the web application is set in a JavaScript file that can either be included in the website main file or included dynamically as an external script when the user web browser loads the application. While the former method is more efficient and should be preferred once the AVS is complete, the later might be more practical during the early days of the AVS as being able to modify the configuration without requesting a new deployment of the website might be faster and more flexible depending on the deployment procedures defined for the hosting infrastructure.

3.3 ADAS integration

In order to streamline the workflow for users who browse data in the AVS and want to download the data represented on the map, the AVS will include links that lead to the ADAS web interface pre-configured with search parameters that match the displayed data.

These links will appear in the contextual menu displayed when an EO granule is selected, as Syntool can include such links in its interface if they are provided as external resources.

This can be done even once the preprocessing step has been completed without requiring a full reprocessing: additional information associated with an EO granule are expected to be any file located in a specific subdirectory of the data representation, so it can be added at a later stage with no side effect.

ANNEX 1: WORK IN PROGRESS

2020-07-01: PM-2

- Multi-server support has been added in Syntool at the end of 2019.
- Web components (portal webpage, webservice) are ready to be deployed.

Next steps:

- preparing a form (jointly with the ADAS development team) to collect details about data requirements and recommended rendering settings from science paper authors
- developing data readers for these data as they become available in the ADAS
- deploying web-services on IFREMER and NERSC servers
- deploying the portal webpage on OceanDataLab server

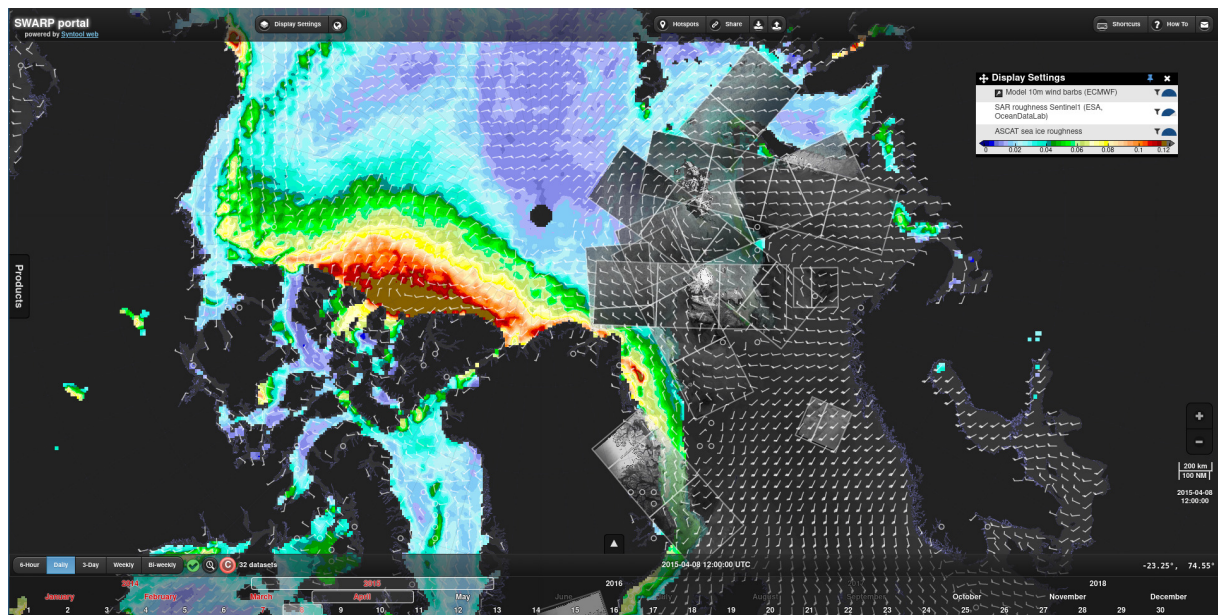


Figure 5: Screen capture of the SWARP Syntool portal that will be used as a basis for the AVS