# ARKTALAS HOAVVA PROJECT

## DELIVERABLE D-50: VERIFICATION REPORT FOR ADAS AND AVS IMPLEMENTATION

| | |
|---|---|
| ***Customer*** | ESA |
| ***Author*** | Consortium |
| ***Distribution*** | Consortium and ESA |
| ***ESA Contract Number*** | 4000127401/19/NL/LF |
| ***Document Reference*** | |
| ***SoW Deliverable Reference*** | D50 |
| ***Version/Revision*** | 1.0 |
| ***Date of issue*** | 13 April 2021 |

| | | |
|---|---|---|
| ***Approved by (NERSC)*** | Johnny A. Johannessen<br>NERSC Project Manager | |
| ***Approved by (ESA)*** | Craig Donlon<br>ESA Technical Officer | Digitally signed by Craig James Donlon<br>Date: 2023.10.22 08:13:59 +02'00' |

# Revision Change log

| Issue | Date | Type | Change description |
|---|---|---|---|
| 0.5 | 19 March 2021 | Draft | Initial version |
| 1.0 | 13 April 2021 | Final | Final version |

*This Deliverable D-50: 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.*

# Table of Contents

# 1 OVERVIEW

Arktalas data are processed by a system structured around two main components: the Arktalas-Data Archive System (ADAS) which is responsible for listing, downloading and storing data of interest for the studies, and the Arktalas Hoavva Analysis and Visualisation System (AVS) which transforms some of these data into representations that can be displayed at full resolution on an interactive map.

The ADAS and the AVS must be hosted on the same infrastructure to avoid unnecessary data transfers and to simplify the orchestration of data processing operations. It does not mean that data discovery, storage and processing are limited to a single infrastructure: multiple systems, each comprised of one ADAS and one AVS, can be deployed on different infrastructures and the Arktalas visualisation portal will be able to access and display the data available in each AVS on the same map, as shown in Figure 1. This distributed architecture makes the overall solution easy to extend, the addition of a new ADAS-AVS couple requiring only a few changes in the portal configuration file.

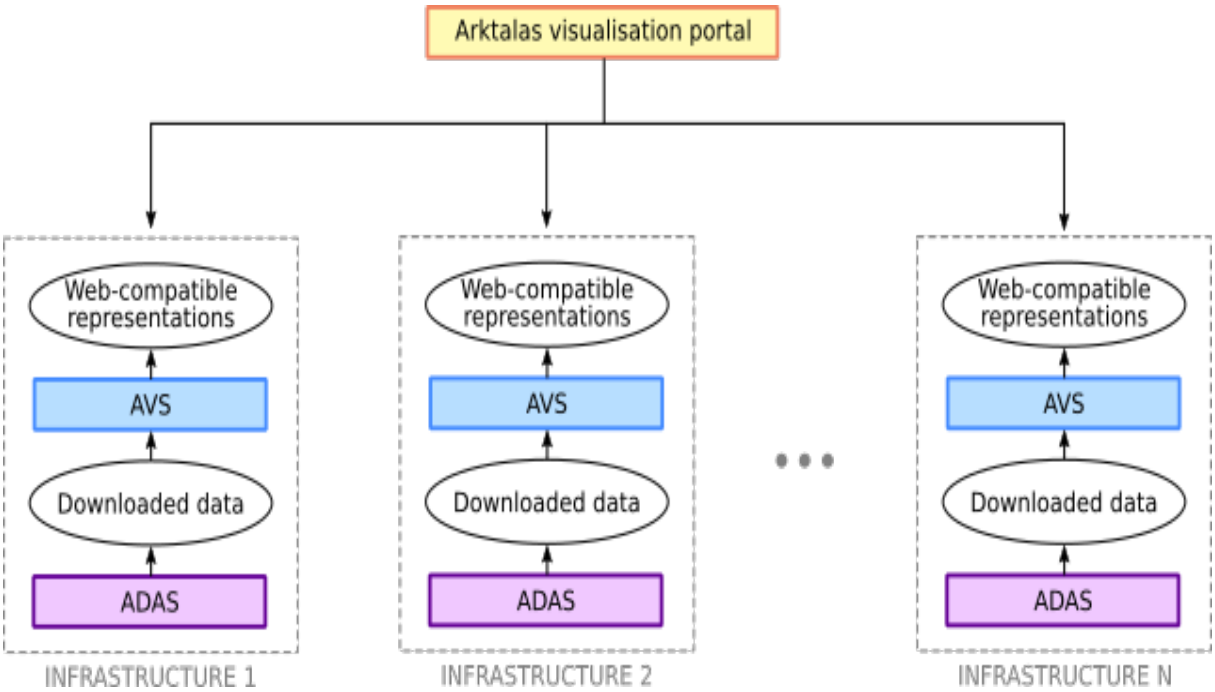


***Figure 1:*** *Distributed architecture for the backend of the Arktalas visualisation portal*

Given these requirements, the communication between the ADAS and the AVS must be as independent from the underlying infrastructure as possible.

# 2 COMMUNICATION BETWEEN ADAS AND AVS

## 2.1 Design

The communication between the ADAS and the AVS is based on files that the ADAS creates in a shared directory which is readable by the AVS. If the ADAS and the AVS are not running on the same machine, this directory must be either shared over the network or replicated on the AVS machine using a synchronization mechanism.

This directory, designated as the ADAS spool directory, contains two types of files:

- the data downloaded by the ADAS, or a symbolic link to the actual files to avoid data duplication
- flag files that contain metadata about the downloaded data, so that the AVS can identify the nature of the files and pass them to the adequate processing chain

The ADAS harvests information from data providers (such as OSISAF and the Copernicus Open Access Hub) periodically to update a database that keeps track of the files available for download and the associated metadata. The files that are relevant for the Arktalas studies are then downloaded automatically. Upon completion of a download, the ADAS first copies (or create a symbolic link to) the data in the ADAS spool directory, then it creates another file named like the data file with an additional *.flag* extension and puts metadata related to the downloaded file in this "flag file".

The metadata inside the flag file include at least one URL where the file can be downloaded from, a "type" field which identifies the product (or data set) which the file belongs to, and potentially other fields describing the origin and contents of the file.

A script is then responsible for periodically inspecting the content of the ADAS spool directory and listing all the files that end with the .flag extension: since the ADAS creates the .flag files only after copying the data files (or creating the symbolic links), listing the .flag files is a safe way to detect the data that are ready for processing.

The detected flag files are then registered in a local database, the AVS jobs database, to keep track of their processing status: a new entry is created for each flag file that was not already registered and all new entries are marked as "requiring processing". The script then queries the jobs database for all entries that yet need to be processed, creates a processing job file in a dedicated directory (AVS spool directory), and finally marks these entries as "dispatched" so that they are not sent to the AVS processing chains multiple times.

A service monitors the content of the AVS spool directory and triggers a processing script each time a new processing job file is created in the spool. This script reads the path of the flag file to process from the processing job file, selects a processing chain using the "type" field extracted from the flag file and then executes this chain on the input file (located by removing the .flag extension from the path of the flag file).

Upon completion the processing chain marks the flag file as "processed" in the jobs database and the script removes the processing job file from the AVS spool directory.

The ADAS periodically removes the files it created in the ADAS spool directory after a delay whose duration exceeds by a fair amount the time it takes for the AVS to process the data. The

ADAS-AVS system is not meant to be used for operational services so there is no real time error reporting. Failures can be detected quite easily though, either by looking for processing jobs that have been dispatched but not completed in the AVS jobs database, or by searching for files older than a threshold in the AVS spool directory. The data flow starting from the files downloaded by the ADAS and ending with their publication in the Arktalas visualisation portal is described in Figure 2.

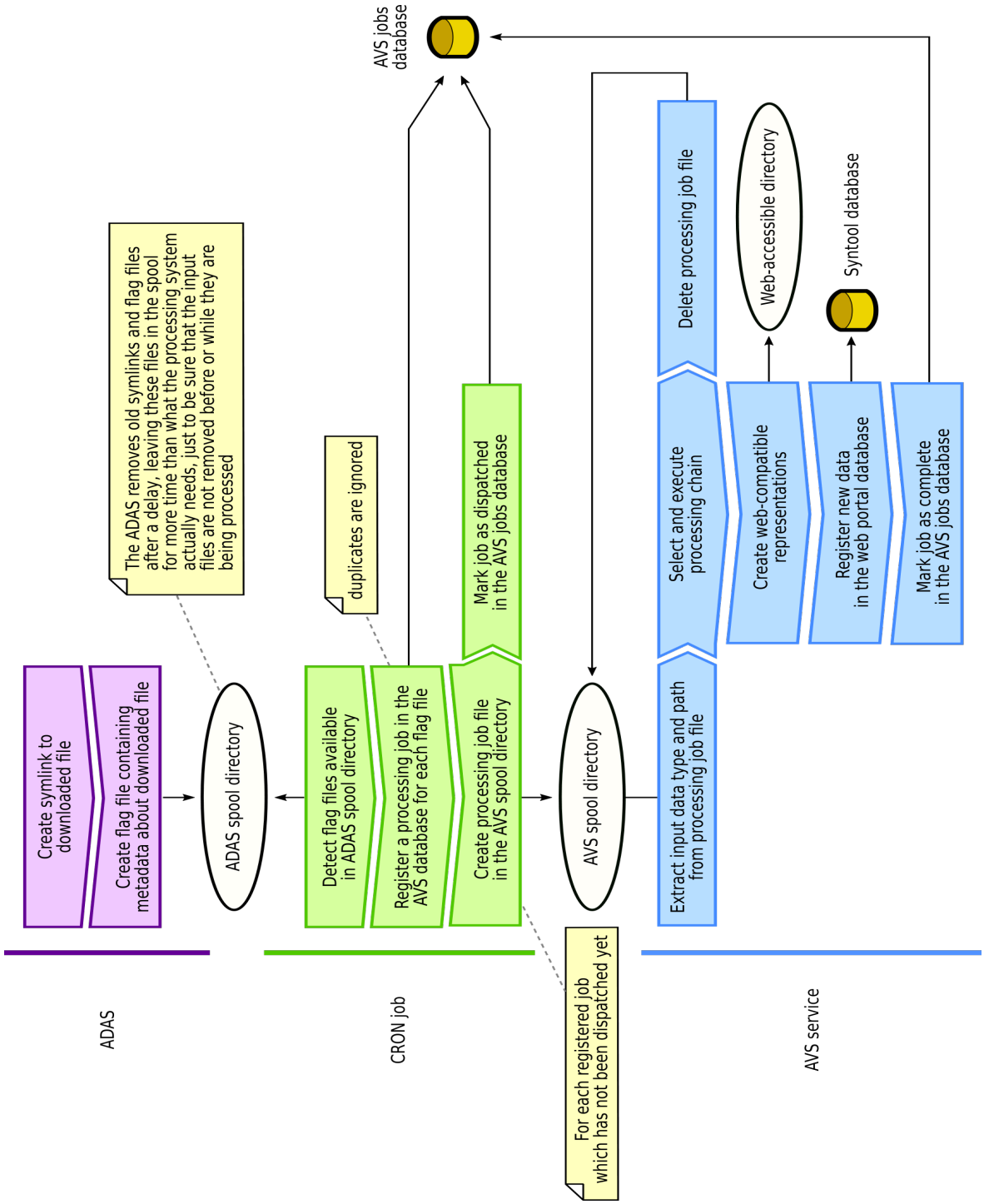


***Figure 2.*** *Design for the ADAS-AVS communication mechanism*

## 2.2 Implementation on NERSC infrastructure

An instance of the ADAS-AVS system has been deployed on the NERSC infrastructure. The initial design has been slightly adapted to be compatible with this platform and its management policy:

- the ADAS spool directory, as well as the storage for downloaded data, is shared with the machine hosting the AVS service over the network using the NFS protocol
- when the AVS service starts, it marks all the incomplete processing jobs (i.e. jobs that were dispatched but whose processing was interrupted due to the AVS service shutdown) as "requiring processing" so that they are dispatched again and completely processed during the next execution of the CRON job. This mechanism allows the system to recover by itself after a crash or a reboot.
- the AVS component has been split into two parts, each running on a separate virtual machine:
  - the Web part comprised of the Syntool database, the Syntool webservice and the Web server (Nginx)
  - the processing part which includes the CRON job, the AVS jobs database and the AVS service in charge of managing the processing chains
- once the data required by the Arkatalas studies have all been ingested in the AVS, the processing part will not be necessary anymore, so the virtual machine hosting the processing system can be shut down to save resources
- instead of sending results directly to the Web virtual machine, the processing chains store all the results they generate (including SQL statements) in a shared directory and the Web virtual machine imports these results periodically. This solution adds a delay before data can be published but it removes coupling between the two virtual machines, thus allowing processing to occur even when the Web virtual machine is offline.

This implementation has been tested successfully with OSISAF sea ice concentration products and Sentinel-1 SAR roughness data, as agreed during Progress Meeting 3. The screenshot in Figure 3 shows how the data processed during these tests are rendered in the Arktalas visualisation portal.

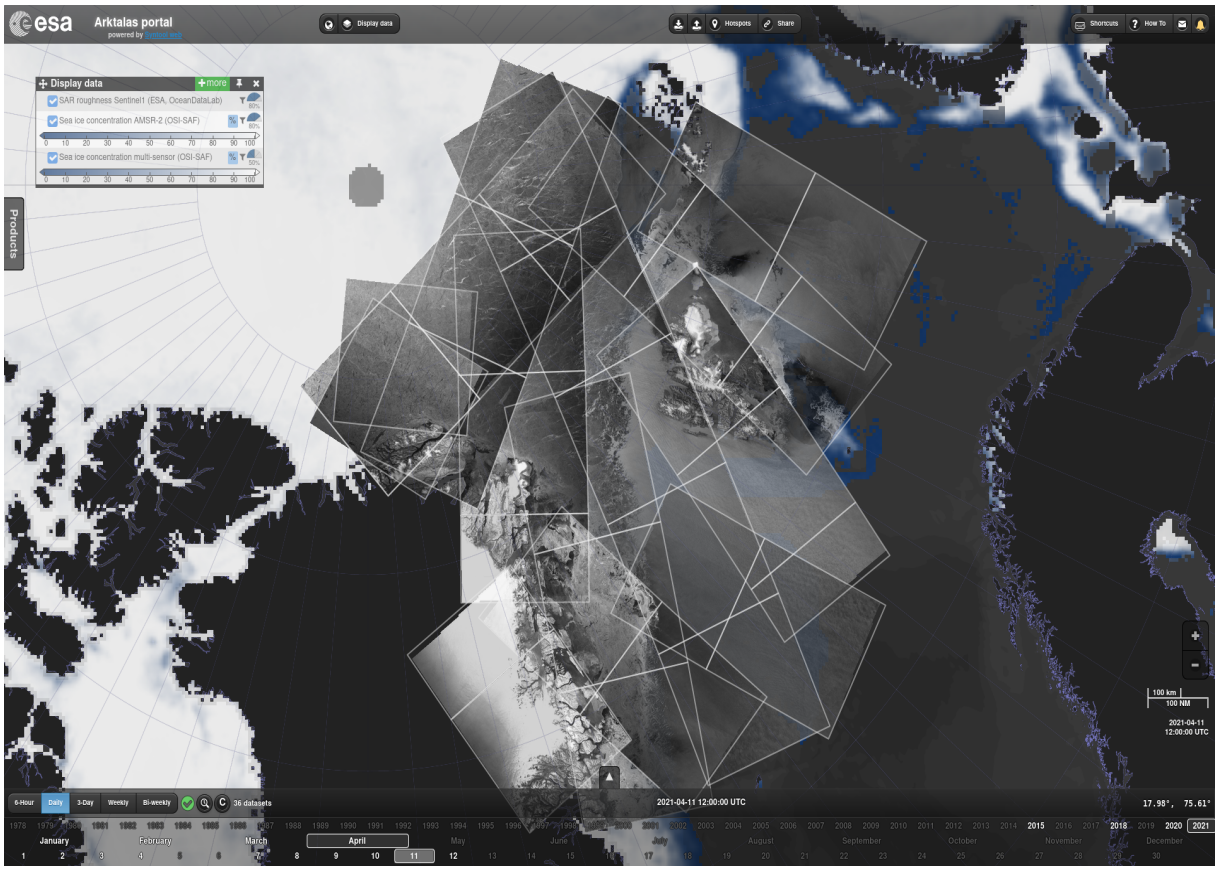


***Figure 3:*** *OSISAF sea ice concentration (AMSR-2 and multi-sensor) and SAR roughness from Sentinel-1 displayed in the Arkatalas visualisation portal*