Collaborative Data Hub Software - Maintenance and Evolution Services - Ready for Digital Twin Earth

DHuS Administration Manual

Role/Title	Name	Signature	Date
Authors	Copernicus IVV Team		19/08/2022
Verified/Approved	Matteo Cortese; Barbara Borgia		19/08/2022

Change register

Version /Rev.	Date	Description
3.0	24-06-2015	First issue of the document in the frame of the Collaborative Data Hub Software Maintenance and Evolution Services for Digital Twin Earth project.
3.1	19-08-2022	The following sections have been added or modified: - 1.4 Reference documents: updated - 1.5 Acronyms: updated - 2.4 DHuS users System Role section: Federation_User role added - 5.4 Product Synchronization section: title modified and new Intelligent Synchronizer mechanism added, together with Product Sources section - 5.5. Monitoring section: added - 7.4 Synchronizer section: updated according to new Intelliget Synchronizer configuration and outdated paragraphs removed - 7.5 Product Sources "how to" sections added - 7.8.1.1 section: added a note concerning Federation_User role - 8.1.10 synchronizer section: synchronizer part of dhus.xml file updated according to new Intelligent Synchronizers configuration - 8.1.11 product sources section: productSource part of dhus.xml added - 8.3 start.sh: parameters concerning skipped products and monitoring added - 8.4 Monitoring.xml section added - 8.5 log4j2.xml section: new debug lines added - APPENDIX A updated - APPENDIX B and C added

Table of Contents

Table of Figures

Table Index

1. Introduction

1.1 Scope

This document applies to the Data Hub Software ("DHuS") and is maintained within the "*Collaborative Data Hub Software Maintenance and Evolution Services for Digital Twin Earth*" contract by Serco.

1.2 Purpose

The purpose of this document is to describe all essential information to make full use of the DHuS software. The target audience of this document therefore are System administrators that will install the DHuS software and manage the DHuS.

This manual includes a description of the DHuS system functions and alternate modes of operation, and step-by-step procedures for system access and use.

In particular, it describes (and dedicates a specific section to):

- How to install the DHuS software;
- How to administer, manage and operate the installed DHuS instance.

The User-oriented manual of the Data Hub is available here https://scihub.copernicus.eu/userguide/.

1.3 Document applicability

Here below we provide a cross check between the SW version and the version of this document. In addition, the new features and performance improvements of each release can be found in the corresponding Software Release Note.

DHuS version	DHuS Administration Manual version	SRN reference
3.1.4-osf	v3.0	COPE-SERCO-TN-22-1366

Table 1 DHuS version & Documents reference Mapping Matrix

1.4 Reference documents

Reference	Document name, Reference, issue number, revision number
RD 1	GitHub open source framework https://github.com/SentinelDataHub/DataHubSystem
RD 2	OData System Design Document, GAEL-P286-SDD-003-01-00, v1.2, 2016-04-25

RD 3	Open Search Description Technical Note, GAEL-P286-TCN-013, v1.0, 2017-06-12
RD 4	DHuS Architectural Design Document, GAEL-P286-ADD-001, v2.1 2014-06-03
RD 5	DHuS OData and Open Search Interface Control Document, COPE-SERCO-IF-17-0018, v2.0
RD 6	Cache management GAEL-P286-TCN-014, v1.0
RD 7	Solr documentation: http://lucene.apache.org/solr/
RD 8	COPE-SERCO-TN-17-0140 DHuS external DB Installation and Configuration Manual v1.2
RD 9	Quartz Enterprise Job Scheduler: http://www.quartz-scheduler.org/documentation/
RD 10	DHuS User Guide: https://scihub.copernicus.eu/userguide/7ODataAPI
RD 11	GAEL-P286-TCN-031-GDPR-1-9.pdf
RD 12	COPE-SERCO-TN-21-1229 - Keycloak Installation and Configuration Manual, v1.1
RD 13	GAEL-P286-TCN-032-Intelligent-Sync-1-4.pdf
RD 14	GAEL-P286-TCN-033-Metrics-1.3.3.pdf

Table 2 Reference Documents

1.5 Acronyms

Acronym	Description
API	Application Programming Interface
BE	Back-End
DHuS	Data Hub Service
FE	Front-End
GUI	Graphical User Interface
SRN	Software Release Note
GDPR	General Data Protection Regulation
COTS	Commercial Off-the-Shelf

1.6 Definitions and Glossary

Acronym	Description
API	Application program interface, is a set of routines, protocols, and tools for building software applications.
Back-End	DHuS instance that has products in its storage (i.e. that has ingested or remote-synchronized products)
Checksum	A checksum is a digit representing the sum of the correct digits in a piece of stored or transmitted digital data, against which later comparisons can be made to detect errors in the data.
DHuS Node	A single DHuS instance, which may be networked with one or more other DHuS instances. Any DHuS node is potentially a source of data products, accessible from any other node in the network.
Front-End	DHuS instance that synchronizes product metadata from a Back-End
Procedures	A set of written instructions to be followed by an operator to perform his required tasks.

2. System overview

The Data Hub Software (DHuS) is an open source software developed by a Serco/Gael consortium to the purpose of supporting the ESA Copernicus data access.

The DHuS provides a web interface (UI) to allow interactive data discovery and download, and a powerful Application Programming Interface (API) that allows users to access the data via computer programs/scripts thereby automating/integrating the download within their workflow.

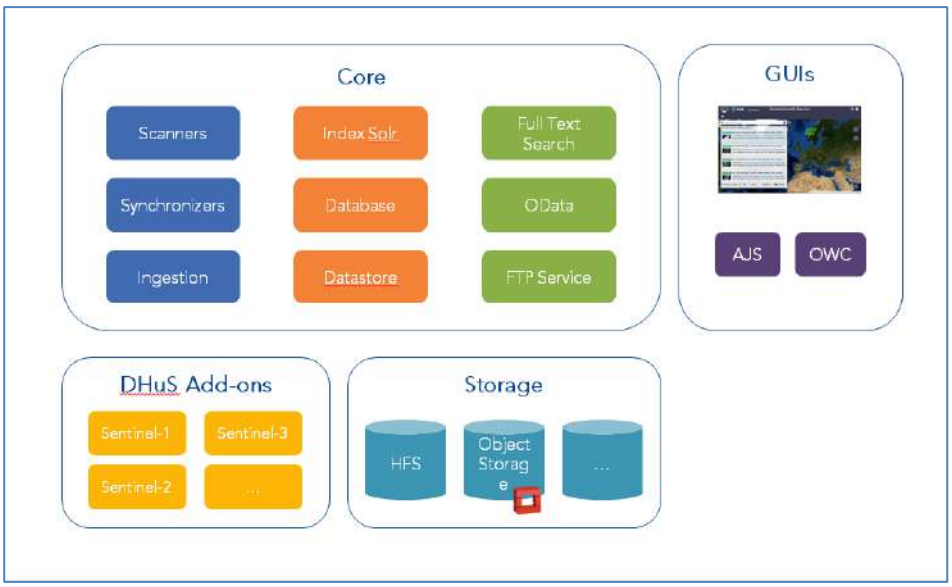

Figure 1.1 DHuS High Level Architecture

2.1 DHuS Functionalities

The major functionalities of the Data Hub Software are schematically represented in figure below.

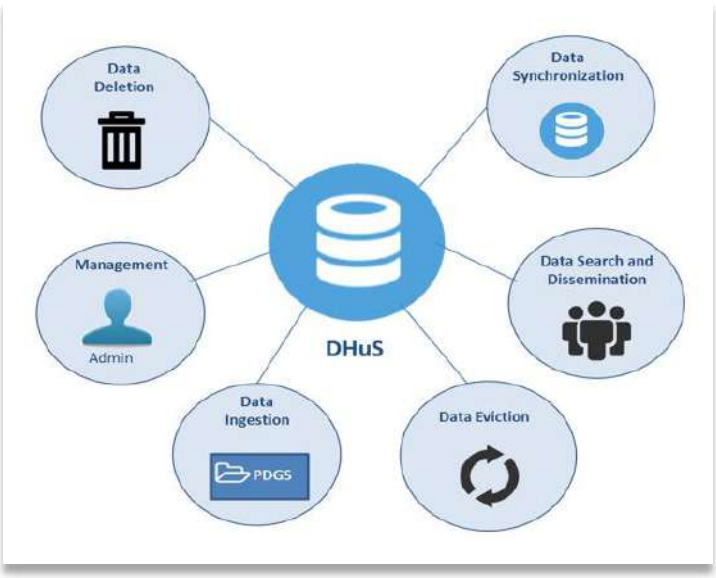

Figure 1.2 DHuS functionalities

Data Ingestion (DHuS→external data provider, e.g. PDGS): This function is activated by DHuS, which retrieves Sentinels' products from an external data source.

Data Search (Users->DHuS): This function is activated by external users who perform searches via four possible interfaces (described in Section 2.2):

- Web-based Graphical User Interface (**AJS**),
- Open Data Protocol (**OData**) interface (see [RD 5]),
- **OpenSearch** interface (see [RD 3]).

Data Dissemination (DHuS→Users): The dissemination function manages the product dissemination, retrieval and access by the user. It also allows disseminating/retrieving sub-components of products. This is considered of particular relevance for the voluminous data products managed in the Sentinels mission.

Data Synchronization (DHuS 1→ DHus 2): This interaction, based on the OData protocol, allows a DHuS Instance (DHuS Front-End) to retrieve products and their metadata from another DHuS instance (DHuS Back-End).

Data deletion (Administators->DHuS): This functionality permits to the administrators to delete products from the archive and eventually to include a record in the DHuS Database which can be read by OData protocol in order to let operators and user to identify the deleted products and the deletion cause.

Management (Administrators->DHuS and Users): This function is in charge of managing the user accounts for access to the DHuS. It enables the definition of the roles and permissions of the users and prevents uncontrolled accesses. Moreover, it controls the data eviction and the system configuration.

Data Eviction (Administators->DHuS): The DHuS rolling archive is managed through control of the overall disk space available for EO products, the management of a general eviction policy and the creation of specific eviction rules governing the period during which products and collections remain available for dissemination prior to eviction.

2.2 DHuS Interfaces

2.2.1 Web-based Graphical User Interface (AJS)

The "default" DHuS graphical interface is the **AJS UI**: it is in charge of providing the user with an interface for the discovery, and downloading of products and for the visualization of the relevant metadata. AJS is accessible by the default at: http://[DHUS_IP_ADDRESS].

2.2.2 Open Data Protocol (OData)

OData (see [RD 5]) is a standardized protocol for creating and consuming data APIs. OData builds on core protocols like HTTP and commonly accepted methodologies like REST thus facilitating access via a large set of client tools as simple as common Web browsers, download-managers or command line tools such as cURL or wget. The result is a uniform way to expose full-featured data APIs.

2.2.3 OpenSearch

OpenSearch (see [RD 3] and [RD 5]) is a collection of technologies that allow publishing of search results in a format suitable for association and aggregation. It is a way for websites and search engines to publish search results in a standard and accessible format.

2.3 DHuS system databases

DHuS application supports both embedded and externalized databases (since version 0.14.1+).

An embedded database means that the database is integrated as an inseparable part of an application software. The database engine runs inside the same JVM while the application is running.

An externalized database means that the database is separate from the application software. It is installed as service on a machine different from the ones hosting DHuS services.

The following databases are available in DHuS:

- Relational database
- Non-relational database

HSQL DB and Solr are the databases supported for the embedded case.

PostgreSQL DB and Solr are the databases supported for the externalized case.

2.4 DHuS users System Roles

The following roles are allowed for DHuS users:

- Search: this role allows to the user to search for products available in the catalogue of the DHuS instance. This role allows also the management of saved searches.
- Download: this role allows to the user to download products from the DHuS instance. In particular, the download from the his/her own Cart is ensured.
- Upload: this role allows to the user to ingest products in the DHuS instance via Manual Upload, OData or FileScanner (if he/she is also Data Manager).
- User Manager: this role allows to the user to manage user accounts. In particular, this user can create, modify, lock, unlock and delete user accounts.
- Archive Manager: this role allows to the user to manage the archive of the DHuS instance. This role is mandatory to perform a metadata synchronization.
- Data Manager: this role allows to the user to delete products and deleted products. Moreover, he/she is also able to manage 'COLLECTIONS' section in the 'Management' panel. In particular, the user can create, update and delete a collection.
- System Manager: this role allows to the user to see Network information and to manage DataStores, Evictions, Orders, Scanners, Synchronizers, Transformations and to repair Products via OData.
- Event Manager: this role allows to the user to manage Event OData entity. In particular, the user can create, modify and delete an Event.
- Statistics: this role allows to the user to see Network information and Metrics via OData.
- Authed: this is a default User Roles needed for DHuS internal mechanisms.
- Federation User: this role allows user to access the *odata/v2/Synchronizers* entity and the *odata/v2/Evictions* entity with only reading permission.

All these User Roles can be assigned to users by administrator at user creation or via "Edit User" function.

Please note that Federation_User role can be created only via OData and not via GUI.

Search and Download are the roles assigned by DHUS when user performs the Sign-Up on the DHuS instance.

3. DHuS deployment modes

DHuS can be installed and configured to run in three different modes. The purpose of having different modes is to cope with increasing user load and amount of data to be managed. The modes, in order of performances (and requirements) are:

- *Single instance mode (Section 3.1);*
- *Front-End/Back-End Mode (Section 3.2);*
- *Scalability 2.0 mode (Section 3.3).*

3.1 Single Instance mode

The deploy in Single Instance mode foresees the set-up of one Data Hub instance exposing its catalogue, i.e. the list of Sentinel' products available in the instance, to the users.

The following functionalities are managed by this instance:

- data harvesting (ingestion);
- management of user requests;
- products publication to end users.

Resources allocation for the single instance depends on the user load and the ingestion rate. Values provided for single instance apply (refer to Section 4.1.1).

3.2 Front-End/Back-End Mode

Deploy in Front-End/Back-End mode foresees the set-up of two (or more) Data Hub instances connected via OData synchronizers (Section 7.4). Depending on the kind of OData synchronizer active among the Data Hub instances, this deployment can be set up in two different ways:

- the data harvesting functionality (ingestion) is managed by one or more DHuS instances, called "*Back-Ends*";
- the management of user requests and products publication to end users is managed by **one** DHuS instance called "*Front-End*".

Resources allocation for the FE and BEs instances depends on the user load and the ingestion rate. Values provided for single instance apply (refer to Section 4.1.1).

3.3 Scalability 2.0 Mode

The objective of the configuration in externalized mode is to have several DHuS instances acting as one to share the user load and the products information: the deployment in externalized mode is completely transparent to the user.

Here in after we list the necessary actors for the deployment of DHuS in scalability 2.0, however resources allocation for all the actors depends on the user load and the product harvesting (ingestion/synchronization) rate.

- **Proxy**: A proxy is needed for load balancing among the nodes. It must be configured to redirect incoming traffic to the DHuS nodes based on a roundrobin load balancing algorithm. Please refer to the proxy documentation for instructions on how to implement this.
- **Relational Database:** it is installed as service on a machine different from the ones hosting DHuS services. It is the only product and user information (e.g. profile changes) data source and it is shared among all DHuS nodes. At now, only PostgreSQL software is supported. Additional PostgreSQL instances can be used for redundancy (please refer to software documentation for instruction on how implement this).

- **Non-Relational Database:** it is installed as service on a machine different from the ones hosting DHuS services. It is the only product indexes data source. So far, only Solr software is supported. Additional Solr instances can be used for redundancy (please refer to [RD 7] for instruction on how implement this).
- **Non-Relational Database orchestrator**: it is installed as service on a machine different from the the ones hosting DHuS services. It is in charge of configuration files management for the non-relational database and redundancy management in case of additional Solr instances. So far, only Zookeeper software is supported.
- **Master:** The DHuS master is the one in charge of the ingestion/synchronization of products. Please note that the DHuS master is functionally equivalent to the DHuS nodes.
- **Nodes:** The DHuS nodes are DHuS instances towards which the user traffic is redirected from proxy. It is mandatory that master and nodes share the same DataStores to allow access to ingested/synchronized products (please refer to 6.1.2 for details about the DataStore configuration). The product deletion and eviction can be executed on all nodes.

Tomcat session and EhCache replication mechanism shall be configured properly in the **dhus.xml** file of all DHuS instances, as described in [RD 8].

Here follows a representation of the deployment in scalability mode:

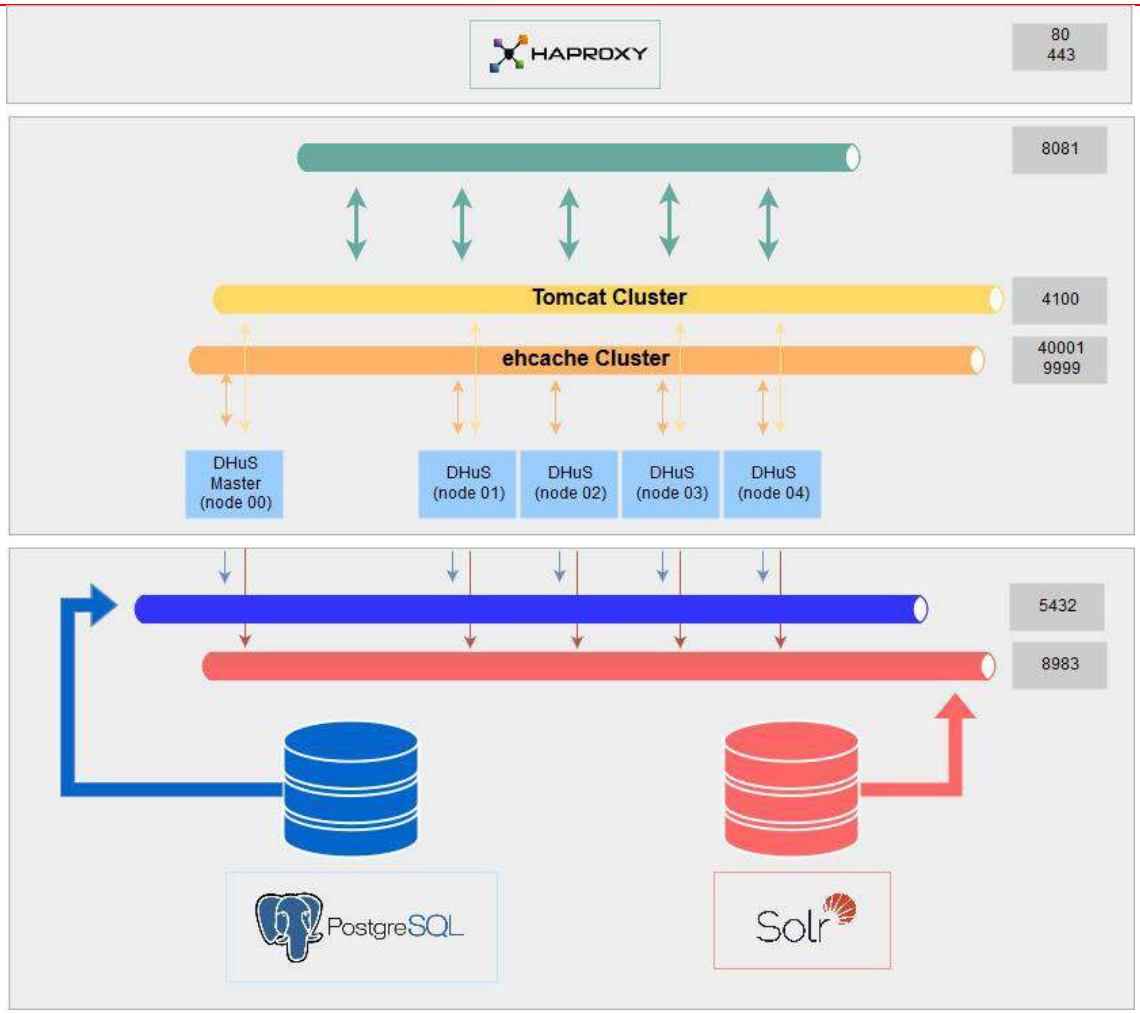

Figure 1.3 DHuS Scalabiliy 2.0 deploy

4. Installation pre-requisites

4.1 Infrastructure Requirements

4.1.1 Single instance and FE/BE Mode

The technical specifications of each machine hosting the DHuS software (both in Single Instance and FE/BE Mode) are provided in the following table:

	MINIMUM	MEDIUM	HIGH
CPU Core Number	4	24	32
RAM	8 GB	32 GB	48 GB
LOCAL DISK	> 1 Gb		

Table 3 Singe Instance & FE/BE Mode Infrastructure Requirements

4.1.2 Scalability 2.0

The requirements for the Scalability 2.0 mode are described in the [RD 8] document, that provides a detailed step-by-step DHuS scalability 2.0 installation guide.

4.2 Network Requirements

DHuS is accessed primarily via HTTP and FTP interface. The Installation procedure of the DHuS Software must be performed using a non-privileged user (not root); application installed in this way cannot start services listening on ports numbers smaller than 1024. By default, the HTTP interface is reachable on 8081 port that must be opened for inbound requests. The DHuS FTP service is reachable, by default, on 2121.The DHuS requires for some of his functions a mailing service based on an external SMTP server. Following table describes the default DHuS network ports configuration:

SERVICES	INBOUND	OUTBOUND
HTTP	8081	External map server port External Nominatim service port
HTTPS	443	-
FTP	2121	-
SMTP	-	25

Table 4 Network Requirements

Please note that, since the UI Map is retrieved from an external service (and not internally from DHuS), the machine where DHuS is installed shall be able to access a map server (e.g. OpenStreetMap (https://www.openstreetmap.org). Moreover, DHuS provides also an interface with Nominatim (http://nominatim.openstreetmap.org/), consequently the machine where DHuS is installed shall be able to reach the service.

4.3 Software Requirements

DHuS software is fully written in Java and can be considered portable to any platform providing the JRE (Java Runtime Environment). The DHuS supports:

- Java RE version 8 (since 0.13.4.21-1 DHuS is no longer compatible with Java 7 version)

Please note that a Java minor version greater than 255 might cause a failure at DHuS start. To solve this issue, the package `icu4j-3.4.4.jar` in the lib/ folder shall be substituted with `icu4j-4.8.jar`.

For what regards the operating system, the distribution is specifically tailored for Linux based systems and has been tested on the following distributions:

- Debian >= 7.7
- Red Hat >=6.7
- Centos >= 6.6
- Mac OS > =El Capitan (run but not tested)

It is recommended to use a Linux Operating System working on a multithread environment running in 64bit.

5. Administration Functionalities

5.1 Login

Once the installation package has been successfully installed (as described in Section 4), the DHuS server can be accessed online (https://[DHUS_IP_ADDRESS]) or on local URL (https://localhost:PORT/). To access the administrator panels, it is first necessary to login as root, using the default settings.

5.2 Product Ingestion

The DHuS allows the ingestion of Sentinels products using the following methods:

- Ad hoc upload (Section 7.2.1);
- Ingestion via File Scanner (Section 7.2.2).

5.2.1 Ad hoc product upload

Ad hoc upload feature is available only to the administrator. DHuS system makes available an incoming space to let the user upload a product via HTTP/HTTPS protocols. Once uploaded, data is processed to be referenced by DHuS clients. This panel gathers all the information necessary to perform the upload (at least the path to the product).

Optional: *Assignation of a product to a collection is manually set by the uploader. A product can be included in any collection.*

Ad hoc upload is not cumulative: only one product can be uploaded at a time.

5.2.2 Upload via File Scanner

Cumulative / parallel uploads can be performed creating a FileScanner (see Section 7.2.2).

The following properties are used to describe and configure a FileScanner:

- **FtpScanner =** Boolean property that allows to insert username and password in case of products to upload stored on an external data provider.
- **SourceRemove =** Boolean property that allows to remove or not successfully ingested products from the Inbox folder, i.e. the folder containing products to be uploaded.
- **Url to scan =** Path of the Inbox folder.
- **Pattern =** Regular expression to select only a specific set of products to be ingested among those present in the Inbox folder.
- **Schedule =** Following the crontab syntax, it allows to schedule each run of the FileScanner.

- **isActive =** Boolean property that allows to activate or deactivate the next runs of a configured FileScanner.

Optional: *Assignation of a product to a collection is manually set by the uploader. A product can be included in any collection.*

5.2.2.1 Actions

It is possible to perform the following actions on a FileScanner:

- **Creation:** Setting at least the Url to scan, it is possible to create a FileScanner in order to ingest products present in the selected Inbox folder. This action can be performed both via AJS GUI and OData v4.
- **Start:** This action allows to scan the configured Inbox folder and ingest products stored in it. At the end of the upload process, the FileScanner stops. This action can be performed both via AJS GUI and OData v4.
- **Stop:** This action allows to stop a FileScanner and it can be performed both via AJS GUI and OData v4. Please note that the stop action does not affect the FileSchedule scheduling: if a FileScanner is scheduled and it is stopped, the nexts runs will start according to the scheduling set.
- **Update:** It is possible to update an existing FileScanner modifying any of its property previously set. This action can be performed both via AJS GUI and OData v4. Please note that before any update, the FileScanner shall be stopped.
- **Schedule/Deschedule:** It is possible to schedule each run of a FileScanner. According to the schedule set, the FileScanner scans the Inbox folder and upload products stored in it. This action can be performed both via AJS GUI and OData v4.
- **Deletion:** This action allows to delete an existing FileScanner. The action can be performed both via AJS GUI and OData v4.

5.3 Product deletion

Product deletion of a product can be triggered via UI or directly via OData API.

In particular, it permits to:

1) **Delete a product via UI**. The deletion is accepted after the declaration of a cause description. At the end of the process, the product:
    - is not present in Product OData entity;
    - is present in DeletedProducts OData entity with the description of the delete cause;
    - is not present in OpenSearch;
    - is not present in the GUI.
2) **Delete a product via OData - No cause specified**. The deletion is launched via command line. At the end of the process, the product:
    - is not present in Product OData entity;

- is present in DeletedProducts OData entity;
- is not present in OpenSearch;
- is not present in the GUI.

3) **Delete a product via OData - Cause specified**. The deletion is launched via command line. At the end of the process, the product:
   - is not present in Product OData entity;
   - is present in DeletedProducts OData entity with the description of the delete cause;
   - is not present in OpenSearch;
   - is not present in the GUI.
4) **Delete a product via OData - Complete deletion**. The deletion is launched via command line. At the end of the process, the product:
   - is not present in Product OData entities;
   - is not present in DeletedProducts OData entities;
   - is not present in OpenSearch;
   - is not present in the GUI;

Please note that in the "Front-End/Back-End" deploy, once activated, the deletion acts on databases and archive according to the instance on which it is triggered:

- If the deletion is launched on a DHuS FE instance connected to a BE via OData metadata synchronizer, the process will only erase data from the Relational database and the Non-Relational database of the FE. The BE will not be affected by any change, in fact the product will not be deleted from the BE databases and archive.
- If the deletion is triggered on a BE instance, the process will erase data from databases and archive. Consequently, if the BE is connected to a FE with an OData metadata synchronizer, to avoid inconsistencies in the FE, it will be necessary to delete the same data also from this instance (otherwise product downloads and product inspection will not work).

5.4 Product Synchronization

The DHuS provides end users an OData synchronizer service able to populate a DHuS instance with the data stored on the rolling archive of another DHuS instance.

Two different kinds of synchronization can be performed:

1) **Metadata synchronization**: Copy of product metadata from a DHuS instance to another. The copy is performed according to predefined synchronization selection criteria

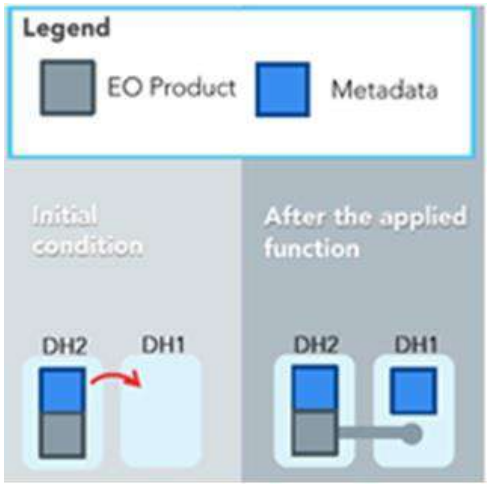

Figure 5.1 Metadata Synchronization

2) **Product/Remote Synchronization**: Copy of product and metadata from a DHuS instance to another. This is different from a product re-ingestion since during the product synchronization the UUID of the product is not re-generated but it is synchronized from the BE.

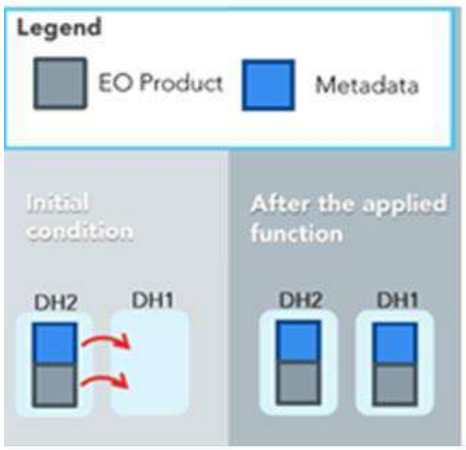

Figure 5.2 Products Synchronization

In case the rolling archive of the Back End contains some products not present in the Front End, once the synchronization runs, the synchronization mirrors products present in the Back End instance respecting the synchronization specification (according to the chosen configuration) and that are not in the database of the Front End instance (in case of metadata synchronization, only the metadata will be mirrored).

Warning: Please note that it is not possible to do metadata synchronization between two Front-End instances; in fact, the metadata synchronizer needs to access the LocalPath property of products to complete the synchronization of metadata with success. Indeed, in this case, quicklook and thumbnail will be not shown in the DHuS instance requesting metadata.

Note: Please note that only backward compatibility is supported for synchronization purposes; in order to have a correct functioning of the Synchronizer feature the Front-End running DHuS version shall be higher or equal to the Back-End running DHuS version.

It is possible to configure multiple synchronizers fetching products from different data sources. In case of more than one synchronizer set, they will be triggered consequentially.

With DHuS 3.0.X branch, the Synchronization mechanism has been enhanced in order to include offline products during the run. In particular, by activating the related parameter, it is possible to synchronize both online and offline products from the Data Source. Offline synchronized products will be offline on the Target DHuS instance and can be retrieved via the dedicated process.

5.4.1 Intelligent Synchronizers

A further improvement made to the synchronizers concerns the possible use of multiple sources for the same synchronizer (therefore called Intelligent Synchronizer).

The sources available for the Intelligent Synchronizer are sorted according to some pre-defined parameters, and the best one is used to synchronize the products.

At start, the synchronizer checks first if it has one or more sources. In case of one source, it is working as today and the new 'listable', 'rankingSchedule', 'lastDateSourceUsed' and 'retryingSourceDelay' parameters values are ignored. In case of multiple sources, the first step is to generate the source ranking. Please note that if some sources are not set as listable, they will simply not be used to extract the list of products to download, but those sources will still be used to download the products (if selected as best source).

In the generation of source ranking, some parameters are collected to compare and choose the best source. In particular, the synchronizer first checks if all sources are available for download products, and then the following parameters are collected for each available source:

- Source synchronization rate, i.e. time weighted information about the size of bytes transferred during synchronization; this value is given by metrics by the field `transferSizeFifteenMinutesRate`, as explained in RD 14 and Section 0.
- Source synchronization outcome, given by `successFifteenMinutesRate/(successFifteenMinutesRate+failureFifteenMinutesRate)`

Subsequently, the highest values for the rate and the outcome between all the different sources are taken as maxRate and maxOutcome, and a value of ratio is computed for each source as:

- SourceRatio= `((sourceOutcome/maxOutcome) + (sourceRate/maxRate))/2`

After that, sources are ranked by their SourceRatio and the source with the highest value is chosen as the best one. The synchronizer ranking is performed according to the ranking schedule of the synchronizer, and it is important to be sure that ranking schedule and schedule of the synchronizer run will not overlap, as detailed better here below.

Clearly, by configuring only one source, it remains possible to use a synchronizer as previously done.

To configure the DataSources that can be used by Intelligent Synchronizers, an ad hoc section has been introduced in the dhus.xml as detailed in the following.

As can be deduced from what has been said and from the algorithm used by the Intelligent Synchronizers shown here above, Intelligent Synchronizers and Reporting features are strictly linked and metrics shall be configured to have Intelligent synchronizers working, as per Section 5.5.

A synchronizer is characterized by the following parameters that shall be configured when the synchronizer is created:

- **Id**= Identifier of the Synchronizer. Read-only and automatically generated at creation
- **Label**= Name of the synchronizer. Must be unique.
- **Schedule**= how often the synchronizer shall be running. This shall be configured according to the crontab syntax. A schedule is configured according to the following pattern (refer to RD 9):
  - `Seconds Minutes Hours Day-of-month Month Day-of-week [Year]`
- **Active=** "true" or "false", it represents the status of the synchronizer, i.e. if it is running or if it is stopped and therefore not synchronizing products
- **Created=** time of the creation of the Synchronizer
- **Modified=** time of the modification of the Synchronizer
- **Target Collection**= is a nullable and modifiable property with no default value. If specified, the synchronized products will be filled in the referenced Collection. Example: `collection_name`.
- **Page size**= number of products synchronized at each synchronizer run. The PageSize parameter for the OData product synchronization with copy is the number of downloads happening in parallel, however, to avoid idling download threads, the download task queue is overfed, thus you may have between 1 and PageSize * (2-1) download tasks submitted. It is suggested to set this parameter lower than the total number of threads allowed in the system.

  Please note that this value represents the `$top` parameter passed to the BE DHuS instance togheter with the `$filter` parameter built using the LastCreationDate (using binary operator 'ge') and FilterParam properties. Due to the 'ge' operator, the beginning of the page contains at least one already synchronized product. So, if you want to synchronize exactly 100 products per run, you have to set a PageSize of 101.
- **Copy product**= this parameter determinates if the synchronization is a metadata synchronization or is a products synchronization:
  - **False**: the synchronizer will synchronize only the metadata of the products (the products are stored only in the Back End incoming folder).
  - **True**: the synchronizer will make a copy of the products in the Front-End incoming folder during the synchronization.

Note that if this parameter is set as true, the "Remote Incoming" field in *productsSource* section shall be set as empty.

- **Filter Parameter**= following the OData filters syntax (see [RD 5]), it filters the products and synchronizes only the ones respecting the filter. Example: `substringof('S1A_',Name)`. For reference see [RD 10].
- **Geofilter** = Geographical filter to synchronize only products belonging to a specific area from the BE. This filter is a post-filter, which means that it is executed client-side, not server-side.

  This Property is a String with a specific syntax: <OPERATOR> <Well-Known-Text Shape>.

  - <OPERATOR> can be:
    - *within* (a product's footprint in inside the given shape)
    - *contains* (a product's footprint contains the given shape)
    - *disjoint* (a product's footprint and the shape don't overlap)
    - *intersects* (a product's footprint and the shape overlap)
  - <Well-Known-Text Shape> is a topologically valid shape written in WKT (See: https://en.wikipedia.org/wiki/Well-known_text)

  Example

  To synchronise products whose footprint overlap with Iceland, it could be possible using the following GeoFilter value:

```
intersects POLYGON((-24.61903123097289 63.345943833554685, -
13.125342536439836 63.345943833554685, -13.125342536439836
66.61007811487349, -24.61903123097289 66.61007811487349, -24.61903123097289
63.345943833554685))
```

- **SkipOnError:** it means if a product is skipped or not when a synchronization error occurs. Default value: true
- **SyncOfflineProducts =** Boolean parameter that allows to synchronize offline products. Its default value is false. In details:
  - **False:** Only products online in the Data Source DHuS instance will be synchronized.
  - **True:** Both online and offline products in the Data Source DHuS instance will be synchronized.
- **RetriesForSkippedProducts** = this parameter allows setting the number of retries that should be put in place in case of skipped products during synchronization run. Its default value is 3.

  When a product is skipped, it will be retried to be downloaded other times. The number of attempts is given by this parameter. The synchronizer run when the product will be retried, is determined by a parameter in the *start.sh* file, that is `-Ddhus.sync.skipped` (details about this parameter are reported in 8.3). In case more than one source is configured for the synchronizer, the skipped product will be retried to be downloaded from the best source selected in that synchronizer run (no matter which source was used during the first download attempt). When the number of retries exceeds the RetriesForSkippedProducts parameter, the product is abandoned and is no longer downloaded.

Please note that the retry mechanism is general, and so each synchronizer checks the global map of skipped products and build its OData filter.

- **TimeoutSkippedProducts** = this parameter defines the timeout in milliseconds of each request of synchronization for skipped products. Its default value is 60000.
- **RankingSchedule=** Schedule configuration of the Intelligent Synchronizer source ranking refresh. This parameter should be used only with Intelligent Synchronizers configuration.

  To ensure Intelligent Synchronizer functioning, ranking schedule and synchronizer schedule shall be shifted. For example:

  - `<sync:schedule>0 */2 * * * ?</sync:schedule>`, i.e. every 2 mins on second 0 for synchronizer schedule
  - `<sync:rankingSchedule>30 */5 * * * ?</sync:rankingSchedule>`, i.e. every 5 mins on second 30 for ranking schedule
- **RetryingSourceDelay=** this value is reported in minutes and represents the delay after which a source will be retried, in order to check its ranking, even if other sources have better performances
- **Sources=** here the sources among which the synchronizer can download data. The parameters for the reported sources are:
  - **ReferencedId=** it represents the Id of the referenced source
  - **LastCreationDate=** Last successfully synchronized product Creation Date. Products with creation date >= of the one here indicated will be synchronized. This not-nullable property is used by the OData product synchronizer to fetch pages of products in the right order. Once the synchronizer ends its run, this field is updated with the creation date of the last synchronized product.

    Please note that, in case of multiple sources for an intelligent synchronizer, the LastCreationDate field is updated for the source used to list the products, and not for the one used to download them.

    Please also note that the field CreationDate is not synchronized from BE to FE, so:

    - the "Creation Date" of a product in the BE represents the availability date on the BE instance of the DHuS during the ingestion process (its value is set to the current Date when the row is inserted in the Database slightly antecedent to the ingestion date)
    - The "Creation Date" of a product in the FE represents the moment in which the product has been published in the FE instance.

    The syntax for the Last Creation Date parameter is the following:

    ```
    YYYY-MM-DDTHH:MM:SS.sss
    ```

    Note: Until 0.12.5-6, the OData product synchronizer used the LastIngestionDate instead of LastCreationDate (using the same approach described above).

This parameter is used to list the pages of products to be synchronized only if the Source is defined as listable in *productSources* section of the dhus.xml

- **LastDateSourceUsed=** the last date the source was used. It is used to understand when reusing a source because it was not used in the last period, as configured in the "RetryingSourceDelay" parameter
- **SourceCollection=** DataSource Collection filter: only the products on the source which are belonging to the configured collection will be synchronized. Example: `Collections('collection_name')`. If this field is left empty, the synchronizer will synchronize every product without any check on their DataSource collection

5.4.2 Product Sources

Product sources are the sources from which a synchronizer can synchronize products. These are listed in an ad hoc section within the dhus.xml.

They can be used individually or even together within the configuration of a synchronizer (in the case of 2 or more product sources associated with the same synchronizer, you will have an Intelligent Synchronizer with all the functions explained in the Section 225.4.1).

A product source is characterized by the following parameters that shall be configured when it is created:

- **Id=** Identifier of the Source. Read-only and automatically generated at creation
- **Url=** https://[DHuS_address]/odata/v1
- **Login=** Username of a user registered in the DataSource. In case of metadata synchronization, the archive manager rights enabled are requested for this user. In case of product synchronization, a normal user (meaning with search and download rights) can be used.
- **Password=** password used to login the user to the Source
- **RemoteIncoming=** path of the incoming folder configured for the DHuS installed as the Back End/DataSource instance. The role of the remote incoming is linked to the kind of synchronizer: metadata synchronizer ("Remote Incoming" field shall be set) or product synchronizer (empty "Remote Incoming" field).

  Moreover, the configuration of such depends also on the DataStore used by the BE/DataSource instance, please find below a summary of the different possible configurations:

	Synchronizer type	Remote incoming	Notes
BE incoming in HFS	Product	N	
	Metadata	Y	Absolute path of the incoming folder configured for the DHuS installed as the Back End instance. The FE will have access to products, quicklook and thumbnails that remain in the BE incoming folder.

BE incoming in openStack	Product	N	
	Metadata	N	The dhus.xml <DataStore> section of the FE instance shall report the description of the openstack DataStore of the BE (where the products are located) in readOnly=true (see 7.16 dhus.xml DataStore section). The FE will have access to products, quicklook and thumbnails that remain in the BE openstack container.

Table 5 Synchronizer type vs Remote incoming configuration

One last possible configuration for the remote incoming is the following: "Copy product" field is set as false (metadata synchronizer) and the remote incoming is not set. In this case, the FE will basically be a catalogue of metadata. This means that it will not have access to products, quicklook and thumbnails.

- **Listable=** Defines if the Source will be used to define the list of products to be synchronized. If set as false, this source will be only used to download and so to synchronize products, but not used to list them. Its default value is false.

  Please note that if a synchronizer with only one source is configured, and the source is set with listable=false, the synchronizer will not be able to synchronize any products

  Please note that if a synchronizer with only one source is configured, and the source is set with listable=false, the synchronizer will not be able to synchronize any product.

5.5 Monitoring

The purpose of the monitoring feature is to collect metrics for performances of DHuS. This task is done by collecting at a fixed rate the performance of certain components of DHuS, like:

- product count
- synchronizer performance
- eviction performance
- ingestion
- HTTP requests (queries and downloads)
- others

Metrics are reported periodically, according to the configuration, and they are exposed on `/odata/v2/Metrics` entity. They can be preserved or not after DHuS stop and it is possible to configure a purge period for the metrics shown in `/odata/v2/Metrics`.

In the *start.sh* file, it is also possible to add a filter on metrics to collect only desired data. In particular:

- `-Ddhus.metrics.filter=ingestion` will collect and show only metrics concerning ingestion
- `-Ddhus.metrics.filter=prod_sync` will collect and show only metrics concerning synchronization

- if no filter is configured into *start.sh*, then all metrics are reported

As explained in Section 5.4, metrics are the basis of the Intelligent Synchronizer mechanism. In particular, here below are reported the measurements collected by metrics and reported in `/odata/v2/Metrics` entity that are used by the ranking mechanism through their "fifteenMinutesRate" field:

- `prod_sync.sync<synchId>.source<sourceId>.meters.transferSize`, size of bytes transfered
- `prod_sync.sync<synchId>.source<sourceId>.counters.success`, number of synchronized products
- `prod_sync.sync<synchId>.source<sourceId>.counters.failure`, number of failures obtained during synchronization

Please note that the "fifteenMinutesRate" values are computed by metric's library used in the software as per RD 9 , and are process meters weighted with time.

In addition to the "fifteenMinutesRate", such exposed OData metrics reports also the count of products matching the meaning of the metric, in the "Count" field. In particular:

- `Count`: for the `prod_sync.sync<synchId>.source<sourceId>.counters.success` and `prod_sync.sync<synchId>.source<sourceId>.counters.failure` metrics is the number of successfully synchronized or failed products, respectively. For the `prod_sync.sync<synchId>.source<sourceId>.meters.transferSize` metric it represents the cumulative sum of bytes for all products successfully synchronized.

The synchronization rate calculated in bytes/s and processed with the following formula:

- `sizeProduct / (timeEndInMillis – timeStartInMillis))*1000`

It is shown in OData via the `prod_sync.sync<synchId>.source<sourceId>.histogram.transferRate` metric, where *`timeEndInMillis`* and `timeStartInMillis` represent the timestamp of end and start of the synchronization, as reported in log file for the debug (for example `"Transfert rate calculation - timeStart: 1656934648460 - timeEnd: 1656934675372 - Size: 2462816615 - TransferRate: 91513000 (bytes/s)")` line.

Unlike `prod_sync.sync<synchId>.source<sourceId>.meters.transferSize`, in this case it is possible to check the "Minimum", "Maximum" and "Mean" values of the synchronization rate of the batch of products synchronized and taken into account, as per "Count" field, getting measurements not weighted with time.

For more metrics details, please refers to RD 14 .

In order to configure the metrics, two configuration files are provided within the DHuS distribution:

- *monitoring.xml-dhus* to configure an embeded database (HSQL DB in file or in memory)
- *monitoring.xml-influx* to configure an InfluxDb for metrics

In order to work correctly, the file must be renamed as *monitoring.xml*. If monitoring.xml file does not exist, monitoring and so metrics are disabled.

For details about the configuration please refer to RD 14 , Section 8.4 and to Appendix B and Appendix C.

The file reported in Appendix C refers to the monitoring file for metrics configured with an externalized database.

The file reported in Appendix B refers to monitoring file for metrics configured with an embedded database.

In particular, as explained in Section 8.4, there are two possible configurations for embedded metrics:

- the first one is with a database in memory: in this case, metrics are stored in HSQL database loaded in-memory. When the DHuS is stopped, metrics are lost, and the database cancelled.
- the second one is in file, i.e. metrics are stored into a file, with a path specified into the configuration. A directory is so created, and here different information regarding metrics is reported, such as a file with all the metrics extracted according to the configured parameter. When the DHuS is stopped, metrics are not lost and they are kept in OData and in file as well, according to the configuration.

5.6 User Synchronizer

The DHuS provides an OData User synchronizer service able to populate a DHuS instance with the users stored on the rolling archive of another DHuS instance. The user synchronizer shall be created in the instance where the users are going to be imported (Front-End) and it can be activated by users with "user manager" rights.

The user synchronizer retrieves users (from a Back-End) following the user creation date criteria, meaning that it retrieves users from the oldest to the most recent. Here follows an example of the query performed by the User synchronizer:

```
http://DHuS_IP:8081/odata/v1/Users?$top=100&$orderby=Created
```

Refer to Section 7.5 for the management of User Synchronizers.

5.7 User Management

DHuS implements a user management system that prevents uncontrolled accesses and manipulations from unauthorized users. DHuS proposes a user authentication and authorization strategy defined in its internal Database. Users are able to register or sign-in and the administrators are able to configure the user/group permissions from the Web user interface.

By means of the dedicated management panels in the UI (Section 7.8.1), the administrator can:

- create and delete any user;
- lock and unlock any user;
- update a user profile, including password reset;
- authorize the user to access a list of services.

User management is also possible via OData API (Section 7.8.2); the administrator can perfrom the following action for any users:

- creation and deletion;
- update profile;
- lock and unlock.

Due to the introduction of the GDPR, the User Management in DHuS 3.0.X branch has been delegated to an external Identity Access Management (IAM), Keycloak. This means that no User information are more stored on DHuS database and all Administrators' actions previously listed are redirected to Keycloak.

5.8 Collection management

DHuS service allows gathering products into collections. The administrator can manage collections, creating or deleting them by means of the dedicated Collection Management Panel.

5.9 System management

The System management is used to configure basic information in the system. By means of the dedicated management panels in the UI (Section 7.10), it is possible to:

- configure the SMTP server address, the username, password and e-mail account details to send communications to the users;
- configure the DHuS Support Team address for any support information;
- change the administrator password;
- configure a periodical dump of the database.

5.10 Product Eviction

The Data Eviction Service is responsible for removing data to keep to the Data Store compliant to sizing or data-offer constraints.

DHuS provides several tools to evict products or product metadata in the form of customizable eviction.

5.10.1 Customizable Evictions

This feature allows fine data management on a DHuS instance by declaring and customizing evictions with several properties such as OData filters.

The configurable **Hard** eviction affects both physical products, product metadata, quicklooks and thumbnails.

Multiple evictions can be declared in a DHuS instance and it is possible to schedule different cron schedules for each eviction configured. If multiple evictions are set, they will be put in a queue of evictions.

The following properties are used to describe and configure an eviction:

- **Name**
  The name of the eviction, used as a reference in OData. Must be unique, cannot be null, and cannot be updated. White space cannot be included in the Name.
- **MaxEvictedProducts**
  The maximum number of products that can be evicted during a single eviction run.

- **KeepPeriod**
  The minimal time after which a product can be evicted. The unit of measure is specified in the parameter "KeepPeriodUnit". It is computed according to `ModificationDate` parameter.
- **KeepPeriodUnit**
  The time unit of the KeepPeriod property. It is possible to define the minimal keeping period for a product on sub-daily basis (e.g. hourly, by minutes). Default value is "DAYS".
- **Filter**
  OData filter used to determine which products can be evicted by this eviction. This filter works the same way as the ones used on the Products entity set.
- **OrderBy**
  An OData ordering clause used to determine in which order products will be evicted by the eviction.
- **TargetCollection**
  The collection targeted by this eviction. Only products that are part of this collection can be evicted by this eviction. If the TargetCollection property is set to null, products will be evicted regardless of their collection.
- **SoftEviction**
  Boolean parameter that shall be set to false.
- **Status**

  The current status of this eviction. The "Status" of Evictions is a read-only property and it can only be changed by the underlying system. Possible status are:
  - STARTED if the eviction is running;
  - QUEUED if the eviction was triggered but another one is currently running;
  - STOPPED if the eviction is not running.
- **Active**

  Boolean property to indicate whether the eviction will be triggered on schedule.
- **Schedule**

  This property represents the schedule configured using the CRON syntax. This CRON property will affect only the associated eviction.
- **SafeMode**

  Boolean parameter that shall be set to false.
- **TargetDataStore**

  The DataStore targeted by the Eviction. Only products stored in the selected DataStore will be processed by the Eviction run.

  This parameter is taken into account only if the Eviction is triggered by a cron.
- **BaseDate**

  This parameter indicates the datetime metadatum with respect to the Eviction run will be performed. The allowed values are `creationDate` and `modificationDate`.

In the dhus.xml file coming from the distribution, an eviction named "DefaultEviction" is set and it is generated automatically by the DHuS at the start-up. This "DefaultEviction" does not have a declared schedule; it can be configured by declaring an eviction CRON in the dhus.xml configuration file or via AJS UI.

5.10.1.1 Actions

It is possible to perform the following actions on a Customizable eviction:

- **Creation:** allows creating a customizable eviction. It can be performed via AJS GUI and OData API.
- **Update:** allows modifying the eviction properties already set. All the eviction properties, except for Name and Status, can be updated. This action can be performed via AJS GUI and OData API.
- **Stop:** allows stopping the current run of the eviction. The Eviction will act again at the next run if a schedule property is set. Stop can be performed via OData API.
- **Deletion:** action to perform in order to remove an eviction rule set. If an eviction run is on-going at the time of the deletion, the run will end and then the eviction will be deleted. The eviction rule will be removed both from the `/Evictions` OData entity (refer to [RD 5]), from GUI and from dhus.xml as well.
- **Queued:** allows manually triggering an eviction run. When this action is called, the concerned eviction is started and its Status becomes "STARTED". If another eviction was already running, the concerned eviction is put in a queue and its Status becomes "QUEUED", it will then be started once the previously running eviction is finished.

All the listed actions can be performed within the DHuS at runtime. Please refer to Section 7.11 for further details.

5.11 OSM Search Engine Interface

DHuS implements an interface with OpenStreetMap search engine Nominatim, available at:

- http://nominatim.openstreetmap.org

This interface allows users to to search OSM data by name and address (geocoding) and to generate synthetic addresses of OSM points (reverse geocoding).

It is configurable via dhus.xml file.

5.12 GDPR compliancy

The European Union General Data Protection Regulation (GDPR) is a data protection ruling that took effect in 2018. It creates one set of guidance and authority to protect the personal data of all EU citizens and applies to any organization that manages data of EU residents.

Since DHuS SW manages data of its own registered Users coming from all over the world, it has been updated to cope with the GDPR requirements.

The DHuS GDPR compliancy is ensured by the externalization of User Management from the SW itself and the deletion of Users info stored on DHuS databases.

Indeed, the entire User Management has been delegated to an external Identity Access Management (IAM) Keycloak.

Please refer to [RD 12] for details on how to install and configure this COTS to properly interface a DHuS instance.

On the other hand, also DHuS SW configuration has been updated to be able to interface Keycloak and allow Users there registered to properly access DHuS instance and perform all currently allowed functionalities.

Except the Login and Logout, all the following User functionalities have been redirected to the IAM:

- Self-Registration
- Edit Profile
- Forgot Password

For Administrator Users, also the following actions on Users are substituted by actions on Keycloak:

- Creation/Update/Deletion
- Lock/Unlock
- Change Password

Please refer to Section 7.8.3 for details on how these actions have to be executed.

To be fully compliant with GDPR requirements, all Users info have to be removed from DHuS instances. To cope with this, a script to delete Users info is provided within DHuS distribution. Please refer to Chapter 12 for more details on script usage.

6. Installation and Configuration Procedures

6.1 Software configuration

DHuS configuration files are:

- **dhus.xml**; located in the `etc` directory
- **server.xml**; located in the `etc` directory
- **start.sh**; located in the installation directory
- **restart.sh**; located in the installation directory
- **stop.sh**; located in the installation directory
- **log4j2.xml**; located in the `etc` directory
- **suggester.dic**; located in the `etc` directory
- **opensearch-description-file.xml**; located in the `etc` directory
- **dhus_ehcache_distributed.xml**; located in the `etc` directory
- **l2aOnDemand.properties**; located in the `etc` folder
- **temporary_files_ehcahe.xml**; located in the `etc` directory

- **AJS GUI**
    - **appconfig.json**; located in `[DHUSDIR]/etc/conf` directory
    - **styles.json**; located in `[DHUSDIR]/var/tomcat/webapps/ROOT/conf` directory

The following subsection describes in details how to configure the major DHuS functionalities; for the complete list of configuration parameters, please refer to Sections 7.20 and 9.

6.1.1 Storage configuration

This section describes how to configure the DHuS data storage, i.e. to use a File System (HFS) or an object storage (OpenStack Swift); this configuration shall be performed within the dhus.xml file.

Since DHuS v0.13.x, it is possible to configure DHuS to interface its data storage (physical data location) by using the `[DataStore]/DataStore` configuration: this configuration allows DHuS to interface both data storage on file system and Object storage.

Please note that the `[System]/archive incoming` parameter is not supported anymore since 0.14.7-2 version.

Thus, starting from 0.14.7-2 DHuS version, the DHuS data storage configuration shall be managed according to the following possible scenarios:

- Database coming from previous DHuS version
- Empty Database

Empty Database (no products)		
HFSDataStore configuration can be used in order to store product in a Hierarchical File System, as well as Object Storage configuration.		
	File System (HFS)	**Object storage (OpenStack Swift)**
Products stored via DataStore	HFS e configuration can be used in order to store product into Object storage. Refer to Section 7.16.1 for instruction about this configuration.	OpenStackDataStore configuration can be used in order to store product into Object storage. Refer to Section 7.16.2 for instruction about this configuration.
Database coming from DHuS previous DHuS version		
Since DHuS 0.14.7-2, OldIncoming configuration is no longer supported. An automatic script, described in Section 10, is provided in order to automatically convert the old incoming configuration in the HFSDataStore configuration in order to store product in a Hierarchical File System.		
	File System (HFS)	**Object storage (OpenStack Swift)**
Old products stored via OldIncoming	The script `updateConfiguration.sh` is in charge to convert the configuration automatically: during the migration from previous DHuS versions to the 0.14.7-2 one, products coming from Oldincoming configuration will be stored in a HFS DataStores. The `<DataStore></DataStore>` section of the DHuS.xml will be automatically filled.	N/A
Products stored via DataStore	The HFSDataStore configuration already in use must be used in the dhus.xml file.	The OpenStackDataStore configuration already in use must be used in the dhus.xml file.

Table 6 Storage configuration on dhus.xml

6.1.2 DataStore properties and configuration

The DataStore Entity is an administration tool inaccessible to regular users and it is available through the OData v4 interface. A DataStore is a layer used by the DHuS to interface its data storage/s.

All the three installation modes (see Section 2.4) can be configured to use a Hierarchical File System storage or an Object storage. Currently DHuS supports the OpenStack Swift Object storage. Swift is a highly available, distributed, eventually consistent object/blob store (https://docs.openstack.org/developer/swift/).

Warning: multiple DataStores can coexist in DHuS; each DataStore is independent from the others. During the ingestion process, products are stored in all DataStores with write access. Please refer to `[DataStore]/DataStore` in [8.1] for details about how to configure DataStores.

Users with the role SYSTEM_MANAGER can access the DataStore entity and update its entries at this address:

- `[DHUS_HOST]/odata/v2/DataStores`

HfsDataStore andOpenStackDataStore inherit from DataStore. Every DataStore, regardless of its subtype, is located in the DataStores entity set.

Table 7 details properties communal to each DataStore:

PROPERTY	TYPE	DESCRIPTION
Name	String	The reference property. It is unique, cannot be null, and cannot be updated.
Restriction	String	Determines whether data can be written on this DataStore. The allowed values are: - none : it allows to the DataStore to do modifications on both products metadata and physical products; - referencesOnly : it allows to the DataStore to do modifications only on the products metadata and not on the physical products; - readOnly : it not allow to do modifications neither on the products metadata nor physical products. Its default value is 'none'. Note: In all DHuS instance acting as Node, this parameter shall be set 'readOnly'.
Priority	Int32	Represents the priority with which DataStores are accessed within the system. DataStores with the smallest priority are accessed first. Its default value is 100.
MaximumSize	Int64	Represents the threshold beyond which a DataStore will attempt to free enough space for that new product. Its default value is -1.
CurrentSize	Int64	Counter of the current size of a DataStore. Its default value is 0.
AutoEviction	Boolean	Parameter that allows to active an Automatic On-Insert Eviction on the DataStore. Its default value is 'false'.

Table 7 Common DataStore entities properties

Properties present in the HFSDataStore are specified here in after:

PROPERTY	TYPE	DESCRIPTION
Path	String	Local storage path.
MaxFileNo	Int32	Maximum number of sub-folder for each stage. Its default value is 10. Please note that this field is labelled as `MaxFileDepth` in OData.
MaxItems	Int32	Maximum number of file stored in each folder. Its default value is 1024.

Table 8 HFSDataStore entity properties

Properties present in the OpenStackDataStore are specified here in after:

PROPERTY	TYPE	DESCRIPTION
Provider	String	Provider service to use, only "openstack-swift" is currently available.
Identity	String	Identifier for the authentication service. The syntax is the following: • KeyStone v2: *project_name:username* • KeyStone v3: *Default:project_name:username*
Credential	String	Credential of the Openstack account to be used. The syntax is the following: *password*
Url	String	URL of Openstack authentication service.
Region	String	Each Region has its own full Openstack deployment, including its own API endpoints and networks. The region is linked to the Openstack account. The default value is *regionOne*.

Container	String	Name of the container to be used. Please note that the container shall be created manually before configuring this parameter. Please check 7.17 for further details about how to create a container in OpenStack.

Table 9 OpenStackDataStore entity properties

6.1.2.1 Actions

The DataStores entity set supports CRUD (Create, Read, Update, Delete) operations on DataStores and allows to manage the configuration of DataStores within the DHuS at runtime.

Here after the list of the allowed actions:

- **Creation:** allows creating a DataStore.
- **Update:** allows updating DataStore properties.
- **Delete:** action to perform in order to remove a DataStore previously created. It will be removed from `/DataStores` OData entity (see [RD 5]) and from dhus.xml.
- **List products contained in the Datasore:** allows listing the total stored products in the DataStore.

Please refer to Section 7.16 for details about how to manage DataStores.

6.1.3 NetCDF Temporary Cache mechanism

DHuS implements a temporary cache feature whose objective is to allow the support for the inspection function of NetCDF-based files (Sentinel-3, Sentinel-5P, COSMO-SkyMed). This mechanism can be applied both in Single Instance and Scalability 2.0 modes. Please refer to Section 7.18 for details about its configuration.

6.1.3.1 Design

The first time NetCDF files are accessed within a product during ingestion, they are extracted and stored as temporary .dat cache files so that subsequent accesses can be done faster without having to perform the extraction again. Temporary cache files have been introduced to make sure this extraction is only performed once per process that is once per ingestion.

In addition, in a Front-End scenario, please note the First Node inspection does not open NetCDF files and no .dat file is created when inspecting the first node of a NetCDF product, i.e the cache mechanism is not involved in this case; cached files are created when the NetCDF node itself is inspected. The temporary file is generated the first time a user browses nodes inside of a NetCDF node (allowing faster browsing of child nodes).

The size of temporary cache files is proportional to the related products and usually larger since zipped products are compressed. So a dedicated physical space on the Virtual Machine hosting the DHuS instance should be dedicated to the temporary cache.

As described in the following sections, this cache feature can be:

- disabled configuring a proper option in the `start.sh` file (see Section 7.18.1 and 8.3);

- customized about location and size configuring proper options in the `temporary_files_ehcache.xml` and `start.sh` (see Section 7.18.2.2 and 8.3);
- shared between the DHuS nodes deployed in Scalability 2.0 configuring properly the `temporary_files_ehcache.xml` file (see Section 7.18.2.3).

6.1.3.2 Approach

NetCDF cache can be configured for both ingestion and inspection purposes. Please find below suggestion on how to configure the cached files location in an efficient way.

- **Ingestion (Back-End side)**

  The netCDF cache should be configured (on NAS or on local partition) and it should be able to help the ingestion process.

- **Inspection (Front-End side)**

  The netCDF cache should be configured only on local partition. If it is not possible to configure the netCDF cache on local partition, the suggestion is to not configure it at all. In this scenario, the response times experienced with a cache configured on a NAS are worst w.r.t. having a disabled cache.

6.2 Single Instance installation

Before staring the installation of DHuS Software, the following recommendations should be taken into account:

- Stop the old DHuS version present in the VM.
- Read available documentation (i.e. Administration Manual and Software Release Note).

6.2.1 Installation starting from an empty database

Procedure

- **Step 1.** Download and install the new DHuS version:
  - Create a user named `dhus`. Every step in the installation procedure, if not explicitly mentioned, shall be performed as dhus user.
  - Create the installation directory: `mkdir -p [install-dir]`
  - Download the DHuS package (shar package) and save it into the installation directory
  - Change the permissions on the file: `chmod +x dhus-XX.XX.XX.shar`
  - Launch `./dhus-XX.XX.XX.shar` (the package will autoinstall).
- **Step 2:** Backup configuration files (start.sh, dhus.xml, server.xml, etc) coming from the distribution
- **Step 3:** Copy configuration files from previous DHuS version in the installation folder. Modify the `etc/dhus.xml` removing the configuration sections filled from the previous version. e.g: `<conf:dataStores>`, `<conf:synchronizers>`, `<conf:evictions>`, `<conf:sources/>`.

Make you sure that the varFolder variable have an absolute path of your choice. This directory will contain the databases folders (if deployed in embedded mode), the dump folders, the tomcat folder, etc.

Example: `<!ENTITY varFolder "/home/dhus/local_dhus">`

- **Step 4:** Create the var folder following the path configured in the `dhus.xml`.
- **Step 5:** Run the `updateConfiguration.sh` script.
- **Step 6**: Start DHuS via the following command in the installation directory:

```
nohup /bin/bash ./start.sh &
```

The log files will be created in the logs directory after the installation.

6.2.2 Installation starting from a database coming from previous versions

Procedure

- **Step 1.** Perform a backup of SQL database and Solr (contained in the var folder of the DHus version running) before migrating to new DHuS versions.
- **Step 2.** Download and install the new DHuS version:
  - Create a user named `dhus`. Every step in the installation procedure, if not explicitly mentioned, shall be performed as dhus user.
  - Create the installation directory: `mkdir -p [install-dir]`
  - Download the DHuS package (shar package) and save it into the installation directory
  - Change the permissions on the file: `chmod +x dhus-XX.XX.XX.shar`
  - Launch `./dhus-XX.XX.XX.shar` (the package will autoinstall).
- **Step 3:** Backup configuration files (start.sh, dhus.xml, server.xml, etc) coming from the distribution
- **Step 4:** Copy configuration files from previous DHuS version in the installation folder
- **Step 5:** Create the varfolder and copy in it the backuppedSQL database and Solr.
- **Step 6:** Run the `updateConfiguration.sh` script.
- **Step 7:** Run the `updateDatabase.sh` script.
- **Step 8**: Start DHuS via the following command in the installation directory: `nohup /bin/bash ./start.sh &`

The log files will be created in the logs directory after the installation.

6.3 Scalability 2.0 Mode installation

Please refer to [RD 7] for details about how install DHuS in externalized mode. In the manual the following information are detailed:

- Installation and Configuration procedures
- Migration of existing embedded HSQLDB to external PostgreSQL
- Migration of existing embedded Solr to external Solr
- Export/Import of external PostgreSQL DB and Solr

7. Administrator Procedures

7.1 Login

7.1.1 How to login as administrator

7.1.1.1 Via Graphical interfaces

The "Login" button is displayed in the upper right side of the DHuS Home page.

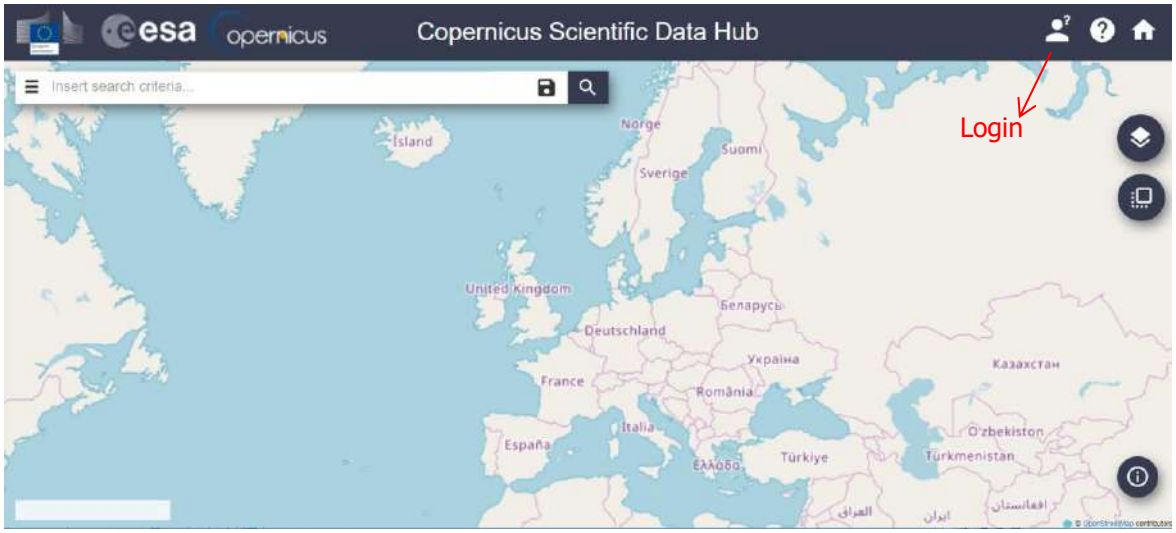

Figure 7.1 DHuS Login (AJS)

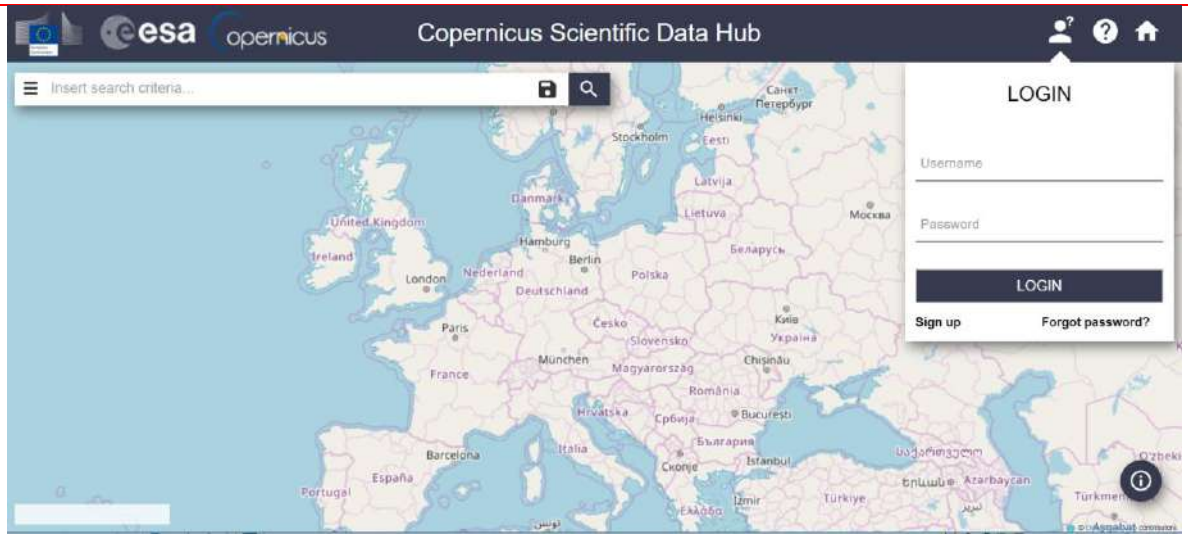

Figure 7.2 DHuS Login Panel (AJS)

Once the user has logged in, the Login badge displays information related to the logged user (email, allowed functions, etc…), refer to Figure 7.3.

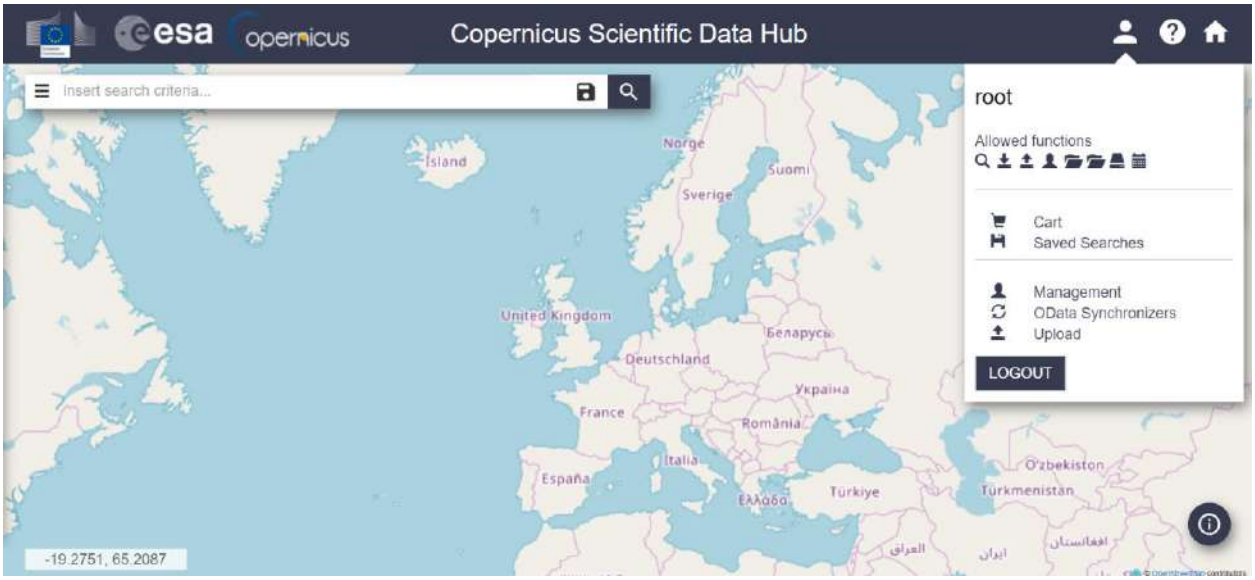

Figure 7.3 User Information display (AJS)

In case of GDPR activated, the Login is still managed by the UI, but username and email related to logged User are no more shown in the Login badge.

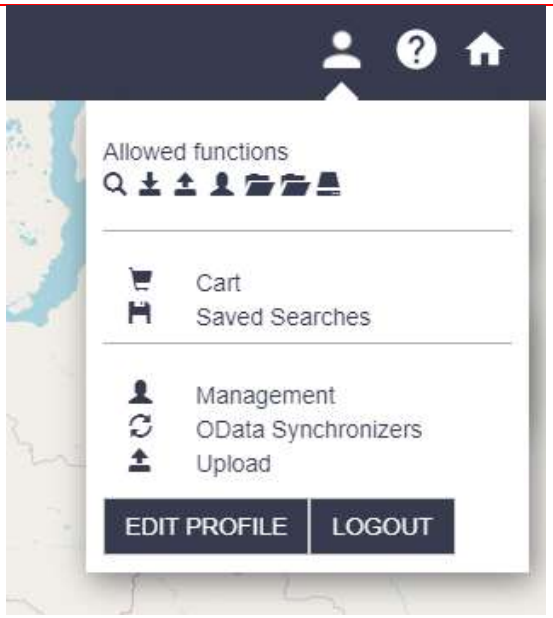

Figure 4: User information in case of GDPR activate

7.2 Product Ingestion

7.2.1 How to upload ad hoc product

7.2.1.1 Via Graphical interface

The product upload is accessible by the "Upload" menu:

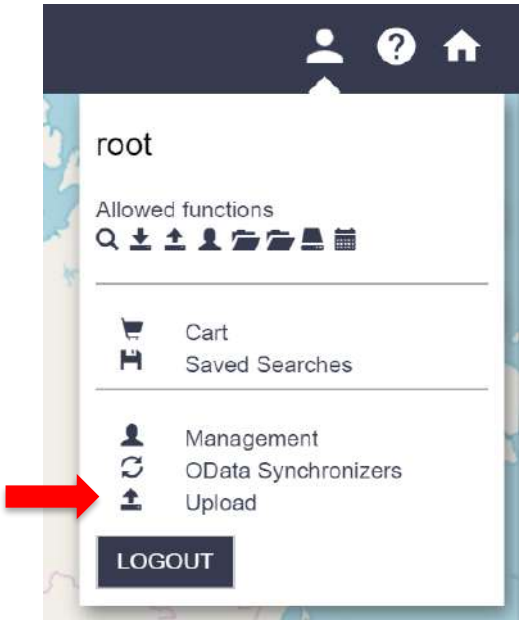

Figure 7.5 Products Upload (AJS)

Once in the "Upload panel", it is possible to perform the upload of a product: select the input products, then (optionally) select a collection in the list of collections and click on the "Upload" button. The upload will start and at the end of it, a pop up will notify that the upload is over.

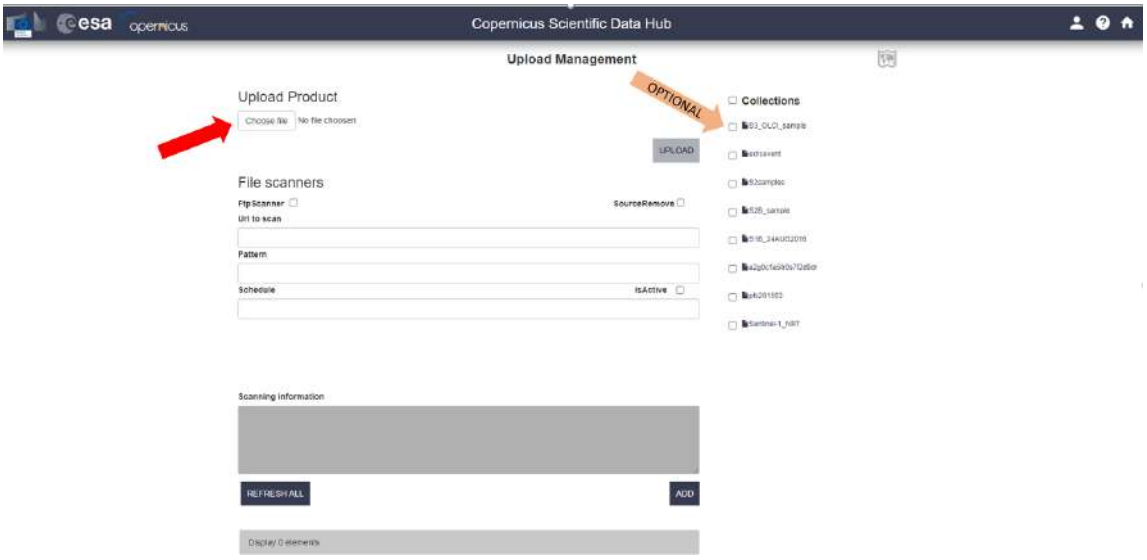

Figure 7.6 Upload Panel (AJS)

7.2.2 How to upload via File Scanner

7.2.2.1 Via Graphical Interfaces

7.2.2.1.1 Creation

In order to create a FileScanner perform the following steps:

- **Step 1:** Access the Upload panel.
- **Step 2:** Configure the following properties:
  - *FtpScanner:* Tick this box, if products to be ingested are located on an external data provider. Then insert username and password to access data.
  - *Url to scan:* Insert the path of the Inbox folder, i.e. the folder containg products to be uploaded.
  - *SourceRemove:* Tick this box in order to remove successfully uploaded products from the Inbox folder.
  - *Pattern:* Insert a regular expression in this field if you want to upload only a specific subset of products stored in the Inbox folder.
  - *Schedule:* Using the crontab syntax, insert the scheduling according to which FileScanner runs starts.
  - *isActive:* Tick this box in order to schedule the FileScanner.

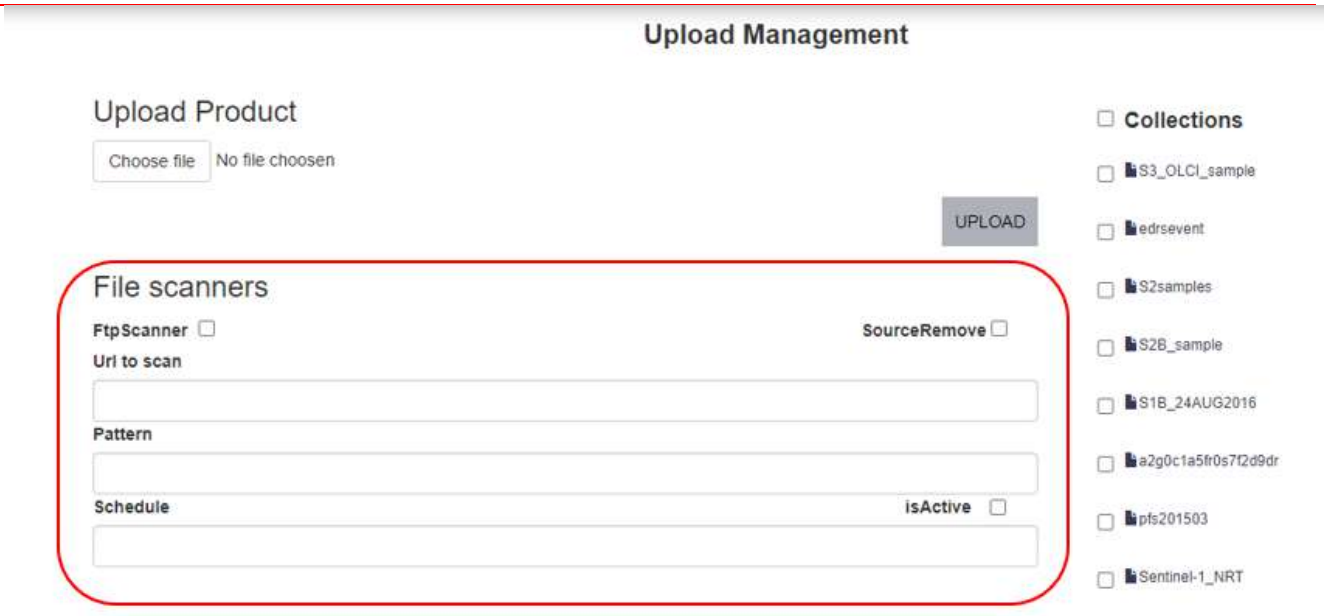

Figure 7.7: FileScanner Creation via GUI

- **Step 3:** If you want to upload products in an existing collection, tick the corresponding box among the available Collections shown on the left of the Upload panel.

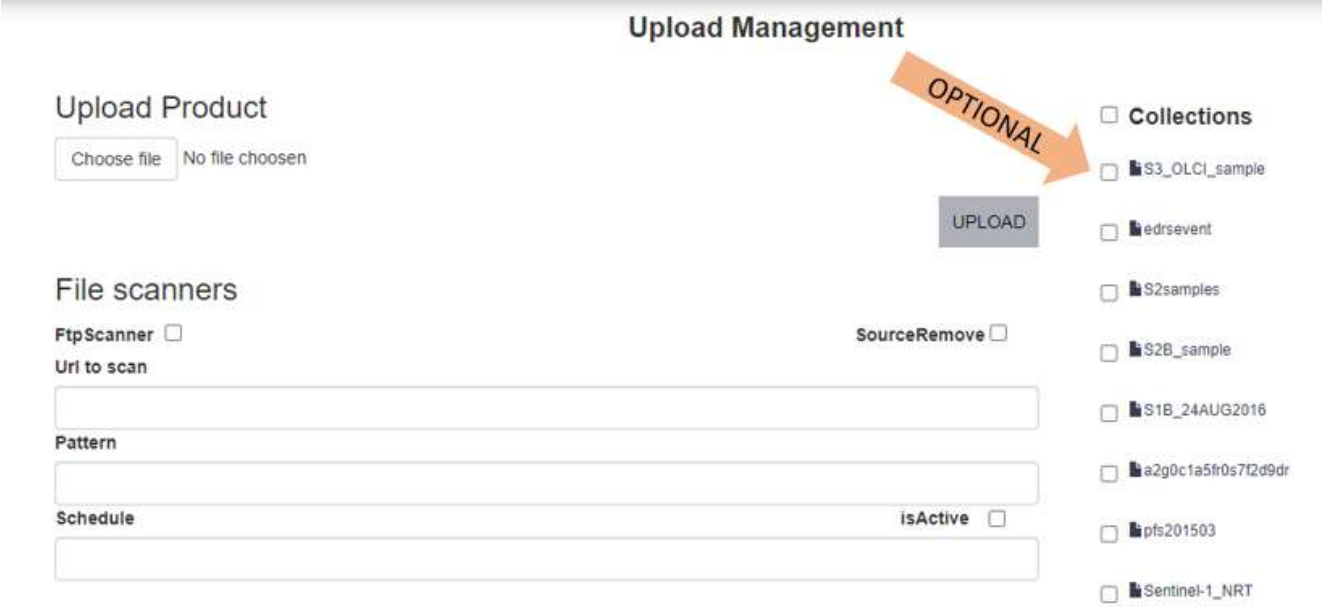

Figure 7.8: How to upload in Collection via GUI

- **Step 4:** Click on 'ADD' button and check that the message 'FILE SCANNER ADDED' is displayed. After the FileScanner creation, the path of the related Inbox folder is reported in the bottom of the Upload panel.

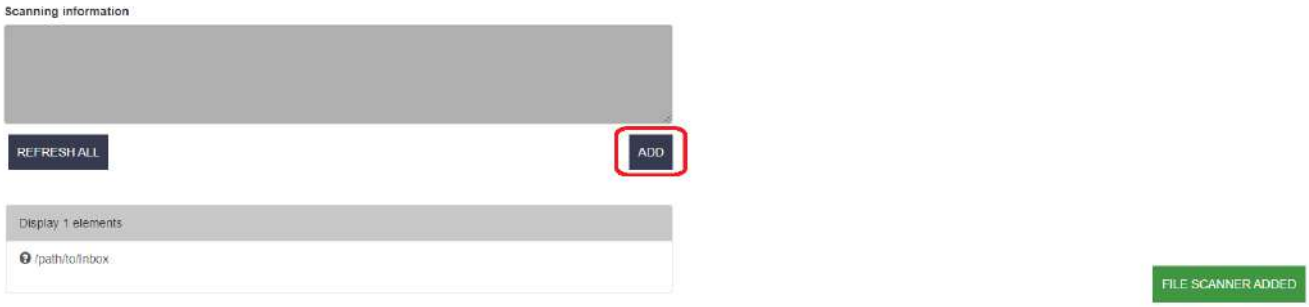

Figure 7.9: Upload panel after FileScanner creation via GUI

7.2.2.1.2 Start

In order to start a FileScanner perform the following steps:

- **Step 1:** Access the Upload panel.
- **Step 2:** In the bottom of the Upload panel, put the mouse over the Inbox path corresponding to the FileScanner to be started in order to show the 'Play File Scanner' icon.

Figure 7.10: GUI icon to start a FileScanner

- **Step 3:** Click on the 'Play File Scanner' icon and check that the message 'FILE SCANNER PLAYED' is displayed.

Figure 7.11: FileScanner start message via GUI

7.2.2.1.3 Stop

In order to stop a FileScanner perform the following steps:

- **Step 1:** Access the Upload panel.
- **Step 2:** In the bottom of the Upload panel, put the mouse over the Inbox path corresponding to the FileScanner to be started in order to show the 'Stop File Scanner' icon.

Figure 7.12: GUI icon to stop a FileScanner

- **Step 3:** Click on the 'Stop File Scanner' icon and check that the message 'FILE SCANNER STOPPED' is displayed.

Figure 7.13: FileScanner stop message via GUI

7.2.2.1.4 Update

In order to update a File Scanner perform the following actions:

- **Step 1:** Access the Upload panel.
- **Step 2:** In the bottom of the Upload panel, click on the Inbox path corresponding to the File Scanner to be updated in order to display the information related to it. Then modify some parameters.

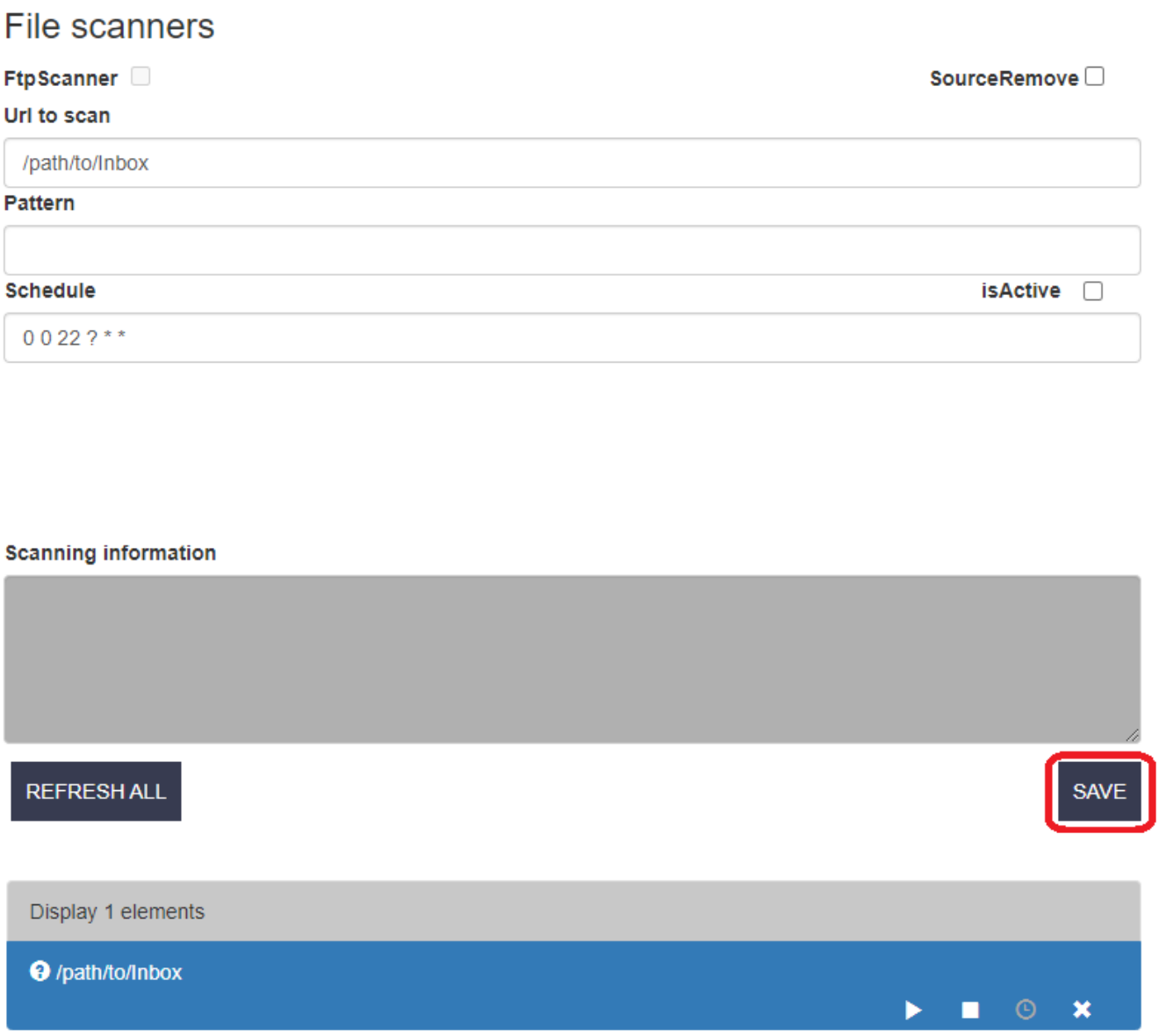

Figure 7.14: Update FileScanner via GUI

- **Step 3:** Click on 'SAVE' button and check that the message 'SCANNER UPDATED' is displayed.

Figure 7.15: FileScanner update message via GUI

7.2.2.1.5 Schedule/Deschedule

In order to schedule or deschedule a FileScanner it is possible to update it ticking or unticking the corresponding 'isActive' box. Please refer to Section 7.2.2.1.4 for details on how update a FileScanner.

It is also possible to schedule or deschedule a FileScanner via the dedicated clock icon. In order to do this, perform the following actions:

- **Step 1:** Access the Upload panel.
- **Step 2:** In the bottom of the Upload panel, put the mouse over the Inbox path corresponding to the FileScanner to be started in order to show the 'Schedule File Scanner' icon. Please note that in case of active FileScanner scheduling the clock icon is blue, otherwise it is grey.

Figure 7.16: GUI icon to schedule/deschedule a FileScanner

- **Step 3:** Click on the 'Schedule File Scanner' icon and check that the message 'SCANNER SCHEDULE UPDATED' is displayed.

Figure 7.17: FileScanner schedule/deschedule message via GUI

7.2.2.1.6 Deletion

In order to delete a FileScanner perform the following actions:

- **Step 1:** Access the Upload panel.
- **Step 2:** In the bottom of the Upload panel, put the mouse over the Inbox path corresponding to the FileScanner to be started in order to show the 'Play File Scanner' icon.

Figure 7.18: GUI icon to remove a FileScanner

- **Step 3:** Click on 'Remove File Scanner' icon and check that the message '' is displayed.

Figure 7.19: FileScanner deletion message via GUI

7.2.2.2 Via OData

DHuS software supports the management of FileScanner using OData v4 API. FileScanners can be inspected and configured using the Scanners entity set on the DHuS OData v4 API:

```
http://SERVICE_URL/odata/v2/Scanners
```

Only users with the Upload role on OData can access Scanners.

7.2.2.2.1 JSON ODATA representation for File Scanner

Here after a JSON ODATA representation as an example for configuring the File Scanner:

```
{
"@odata.type":"#OData.DHuS.FileScanner",
"Url": "/path/to/INBOX_folder",
"Pattern": "^S2[A|B]_MSIL1C.*zip$",
"SourceRemove": true,
"Cron": {"Active": true, "Schedule": "0 0/1 * * * ?"}
}
```

7.2.2.2.2 Creation

The following instruction has to be performed via command line in order to create a FileScanner:

```
curl -H "Content-Type: application/json" -X POST --data "@/path/to/scanner.json" -u username:password "http://localhost:8081/odata/v2/Scanners"
```

where the "`scanner.json`" is the name of the JSON file containing the proper configuration for the creation of the FileScanner.

7.2.2.2.3 Start

The following instruction has to be performed via command line in order to start a selected FileScanner:

```
curl -H "Content-Type: application/json" -X POST --data "" -u username:password "http://localhost:8081/odata/v2/Scanners(ID)/OData.DHuS.StartScanner"
```

where `ID` is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

7.2.2.2.4 Stop

The following instruction has to be performed via command line in order to stop a selected FileScanner:

```
curl -H "Content-Type: application/json" -X POST --data "" -u username:password "http://localhost:8081/odata/v2/Scanners(ID)/OData.DHuS.StopScanner"
```

where `ID` is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

7.2.2.2.5 Update

The following instruction has to be performed via command line in order to update a FileScanner:

```
curl -H "Content-Type: application/json" -X PATCH --data "@/path/to/scanner.json" -u username:password "http://localhost:8081/odata/v2/Scanners(ID)"
```

where:

- "scanner.json" is the name of the JSON file containing the proper configuration for the update of the FileScanner;
- ID is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

Please note that before any update, the FileScanner shall be stopped.

7.2.2.2.6 Check Status

The following instruction has to be performed via command line in order to check the status of a selected FileScanner:

```
curl -X GET -u username:password
"http://localhost:8081/odata/v2/Scanners(ID)/ScannerStatus"
```

where ID is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

7.2.2.2.7 Upload in Collection

In order to ingest products in an existing Collections, the following Json file shall be configured:

```
{
"@odata.id":"http://localhost:8081/odata/v2/Collections('Collection_name')"
}
```

Then, the following instruction has to be performed via command line in order to link the Collection to the selected FileScanner:

```
curl -H "Content-Type: application/json" -X POST --data
"@/path/to/Collections.json" -u username:password
"http://localhost:8081/odata/v2/Scanners(ID)/Collections/\$ref"
```

where:

- "Collections.json" is the name of the JSON file described above;
- ID is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

7.2.2.2.8 Deletion

The following instruction has to be performed via command line in order to delete a selected FileScanner:

```
curl -X DELETE -u username:password "http://localhost:8081/odata/v2/Scanners(ID")
```

where `ID` is the identifier of the FileScanner to update. It can be read from the corresponding Scanners OData Entity.

7.3 Product deletion

7.3.1 How to delete a product

7.3.1.1 Via Graphical Interfaces

Step 1: Select a product to be deleted from the UI list/search panel. It is possible to select set of products also using the "select all" button (bulk deletion);

Step 2: Delete products clicking on the trashbin icon in the lower side of the panel itself

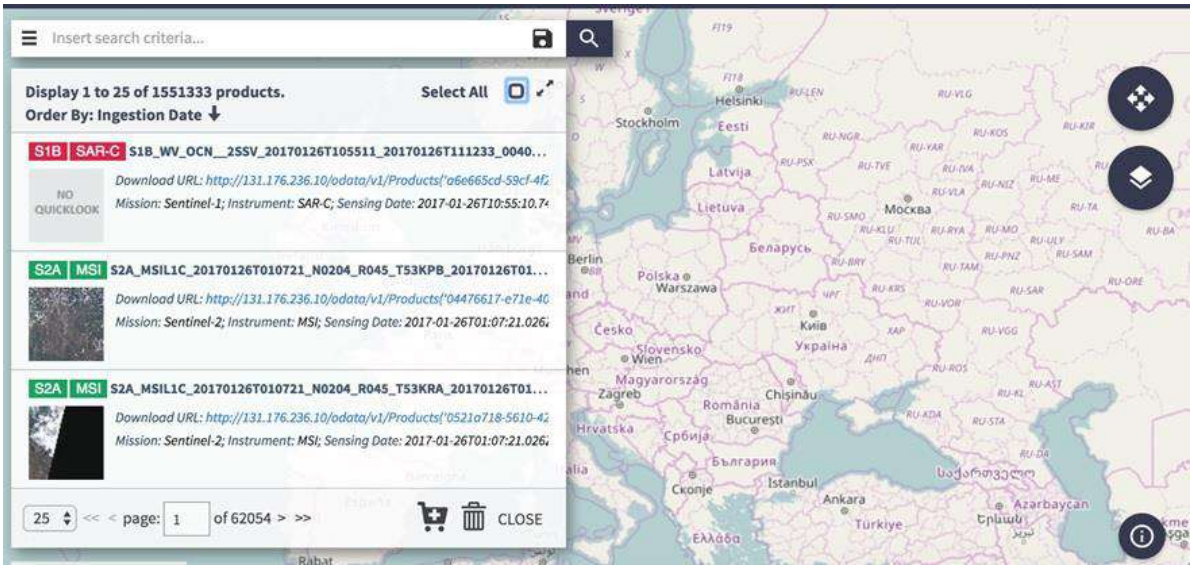

Figure 7.20 Deletion via OData API

A prompt message is showed up to allow inserting a deletion cause:

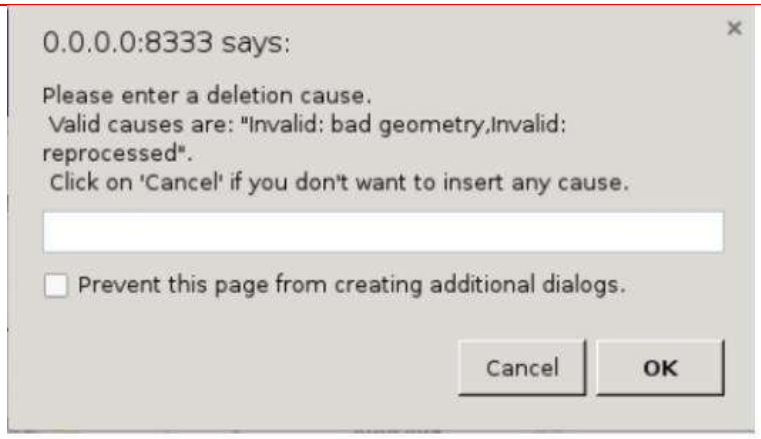

Figure 7.21 Deletion cause panel

The set of deletion causes shown in the panel can be configured as explained in Section 9.1 (AJS GUI `appconfig.json`).

If an operator does not choose one of the configured deletion causes, the DHuS set the deletion cause value as "Invalid".

7.3.1.2 Via OData

The Data Hub provides to administrators the possibility to delete products via OData.

All deleted (and evicted) products are accessible to all kind of users at:

http://[DHUS_IP_ADDRESS]/odata/v1/DeletedProducts

How to delete a product via OData?

- With no cause:

```
curl -u ${DLOGIN}:${DPASS} X DELETE
"http://${DHOST}:${DPORT}/odata/v1/Products('<UUID>')"
```

  In this case, the DeletionCause is « null »

- With a generic cause:

```
curl -u ${DLOGIN}:${DPASS} X DELETE
"http://${DHOST}:${DPORT}/odata/v1/Products('<UUID>')?cause=Size%20Needed"
```

- Determining it as "Invalid":

```
curl -u ${DLOGIN}:${DPASS} -X DELETE
"http://${DHOST}:${DPORT}/odata/v1/Products('<UUID>')?cause=Invalid"
```

- Removing it from the Hub and the archive:

```
curl -X DELETE -u ${DLOGIN}:${DPASS}
"http://${DHOST}:${DPORT}/odata/v1/Products('<UUID>')?purge=true"
```

It is strongly recommended to identify invalid products by starting their deletionCause with "Invalid". Such practice will make easier to set up a regular expression in future (for example: "Invalid, bad geometry" or "Invalid, reprocessed").

Please note that evicted products have their DeletionCause automatically set to "Automatic Eviction".

7.4 Synchronizer

7.4.1 How to create a new Synchronizer

7.4.1.1 Preconditions

The source/target instances shall be configured as follows:

1) source instance: DHuS instance with a proper quota limitation.
2) target instance: having the synchronization functionality enabled, meaning that the dhus.xml shall contain the following setting:

   ```
   <executor enabled="true" batchModeEnabled="false">
   ```

   and having a user with the 'archive management' function enabled (in the case of metadata synchronization).
3) before creating the intelligent synchronizers, the metrics must be activated by configuring the *monitoring.xml* file, otherwise they cannot work

In case a proxy has been set up between Source and Target instances, please change the network properties in order to include the source in the list of IP that can access without going through the proxy.

7.4.1.2 Via OData

Since the Synchronizers GUI panel is not updated to the last DHuS version (in particular, Intelligent Synchronizers work with OData v2, while GUI is still using OData v1), it is possible to create synchronizers only using OData, with the following commands:

- Log in the machine where the DHuS to fill with products is installed
- To create the synchronizer, use the command:

  ```
  curl -H "Content-Type: application/json" -X POST --data
  "@/path/to/Synchronizer.json" -u username:password http://
  [DHuS_URL]/odata/v2/Synchronizers
  ```

  where "Synchronizer.json" is the name of the JSON file that contains the parameters for the synchronizer. An example of this file is given below:

  ```
  {
   "@odata.type": "#OData.DHuS.ProductSynchronizer",
   "Id": 0,
   "Label": "S1",
  ```

```
 "Cron":{
 "Active":false,
 "Schedule":"0 */2 * * * ?"
 },
 "GeoFilter":  "intersects  POLYGON((-24.61903123097289  63.345943833554685,-
13.125342536439836                    63.345943833554685,-13.125342536439836
66.61007811487349,-24.61903123097289     66.61007811487349,-24.61903123097289
63.345943833554685))",
 "PageSize": 2,
 "RetriesForSkippedProducts": 1,
 "TimeoutSkippedProducts": 60000,
 "RetryingSourceDelay": 20,
 "SyncOfflineProducts": false,
 "SkipOnError": true,
 "CopyProduct":true,
 "FilterParam":"startswith(Name,'S2A')",
 "RankingSchedule":"30 */5 * * * ?",
 "TargetCollection": [{
 "@odata.type": "#OData.DHuS.Collection",
 "Name": "S2A" }],
 "ReferencedSources": [{
 "@odata.type": "#OData.DHuS.ReferencedSource",
 "ReferenceId":0,
 "LastCreationDate": "2022-06-27T00:00:00.001Z" }, {
 "@odata.type": "#OData.DHuS.ReferencedSource",
 "ReferenceId":1,
 "LastCreationDate": "2022-06-27T00:00:00.001Z" }]
}
```

7.4.2 How to update a Synchronizer

7.4.2.1 Via OData

Since the Synchronizers GUI panel is not updated to the last DHuS version (in particular, intelligent synchronizers work with OData v2, while GUI is still using OData v1), it is possible to update intelligent synchronizers only via OData with the command reported below:

```
curl -H "Content-Type: application/json" -X PATCH --data
"@/path/to/Synchronizer.json" -u username:password
"http://[DHuS_URL]/odata/v2/Synchronizers(<ID>)"
```

where:

- <ID> is the Synchronizer identifier to be updated.
- "Synchronizer.json" is the name of the JSON file that contains the parameters for the synchronizer. The file structure is the same of the "Synchronizer.json" reported in 7.4.1.2

7.4.3 How to delete a Synchronizer

7.4.3.1 Via OData

With the following command it is possible to delete a synchronizer, removing it both in OData and in the dhus.xml file.

- Log in the machine where the DHuS to fill with products is installed
- To remove the synchronizer, use the command:

  ```
  curl -H "Content-Type: application/json" -X DELETE -u username:password "http://[DHuS_URL]/odata/v2/Synchronizers(<ID>)"
  ```

  where <ID> is the Synchronizer identifier to be deleted.

7.5 Product Sources

7.5.1 How to create a new product source

7.5.1.1 Via OData

Since the Synchronizers GUI panel is not updated to the last DHuS version, it is possible to create product sources only using OData, with the following commands:

- Log in the machine where the DHuS to fill with products is installed
- To create the product source, use the command:

  ```
  curl -H "Content-Type: application/json" -X POST --data "@/path/to/ProductSource.json" -u username:password http://[DHuS_URL]/odata/v2/ProductSources
  ```

  where "ProductSource.json" is the name of the JSON file that contains the parameters for the product source. An example of this file is given below:

  ```
  {
   "@odata.type": "#OData.DHuS.ProductSource",
   "Url":"https://scihub.copernicus.eu/dhus/odata/v1/Products",
   "Login": "***",
   "Password": "***",
   "RemoteIncoming": "/my/directory/for/products",
   "Listable": true
  }
  ```

7.5.2 How to update a product source

7.5.2.1 Via OData

Since the Synchronizers GUI panel is not updated to the last DHuS version, it is possible to update product sources only via OData with the command reported below:

```
curl -H "Content-Type: application/json" -X PATCH --data
"@/path/to/ProductSource.json" -u username:password
"http://[DHuS_URL]/odata/v2/ProductSources(<ID>)"
```

where:

- <ID> is the Product Source identifier to be updated.
- "ProductSource.json" is the name of the JSON file that contains the parameters for the product source. The file structure is the same of the "ProductSource.json" reported in 7.5.1.1

7.5.3 How to delete a product source

7.5.3.1 Via OData

With the following command it is possible to delete a product source, removing it both in OData and in the dhus.xml file.

- Log in the machine where the DHuS to fill with products is installed
- To remove the product source, use the command:

  ```
  curl -H "Content-Type: application/json" -X DELETE -u username:password
  "http://[DHuS_URL]/odata/v2/ProductSources(<ID>)"
  ```

  where <ID> is the product source identifier to be removed.

7.6 User Synchronizer

7.6.1 How to create a User Synchronizer

- **Step 1:** Log in the machine where the FE DHuS is installed. This DHuS instance shall be filled with users registered in the BE.
- **Step 2:** On the DHuS acting as FE, set the environment variables (export <variable>=value) that configure the DHuS:
  - DHOST DHuS Host address eg: localhost:8080, scihub.copernicus.eu/dhus/
  - DLOGIN DHuS login username to connect to the $DHOST DHuS

- DPASS DHuS login password to connect to the $DHOST DHuS

- **Step 3:** Install M4 Macro Language package running
  - `apt-get install m4` (on Debian platforms)
  - `yum install m4` (on RedHat/CentOS platforms)
- **Step 4:** Create a working folder containing the scripts available here https://github.com/SentinelDataHub/Scripts/tree/master/dhus_sync_script
- **Step 5:** To create the user synchronizer use the command:

```
./createSynchronizer <-D_SCHEDULE=…> <-D_REQUEST=start|stop> <-
D_SERVICEURL=...> [-D_LABEL=...] [-D_SERVICELOGIN=...] [-
D_SERVICEPASSWORD=...]
```

Example:

```
. /createSynchronizer -D_SCHEDULE="0 */3 * * * ?" -D_REQUEST=stop \
-D_SERVICEURL="http://192.168.0.105:8080/odata/v1" -D_LABEL=my_user_syncer \
-D_SERVICELOGIN=root -D_SERVICEPASSWORD=a
```

Where:

- **Schedule**= how often the synchronizer shall be running. This shall be configured according to the crontab syntax
- **Request**= "start" or "stop"
- **Label**= Name of the synchronizer
- **Service URL**= https://[Back-End_DHuS_address]/odata/v1
- **Service Login Username**= User name of a user registered in the Back end with user manager role.
- **Service Login Password**= password of the user in the previous step
- **Page size**= number of users synchronized at each synchronizer run
- **Cursor=** if the cursors is set as "i" (where i is such that 0<i<total amount of users) the synchronizer will import in the FE all the users of the BE from the i-th.
- **Force=**true/false. This parameter defines if this UserSynchronizer should overwrite namesakes. If Forced is set to true, namesakes will be overwritten thus after a complete synchronisation of all users from the data source, there will be no namesakes on the synchronising DHuS.

7.6.2 How to start/stop a User Synchronizer

To start an existing synchronizer execute the following command:

```
./updateSynchronizer <id> -D_REQUEST=start|stop
```

Example:

```
./updateSynchronizer 0 -D_REQUEST=start
```

7.6.3 How to update a User Synchronizer

To update an existing synchronizer, execute the following command:

```
./updateSynchronizer <id> [-D_SCHEDULE=...] [-D_REQUEST=start|stop] \
[-D_SERVICEURL=...] [-D_LABEL=...] [-D_SERVICELOGIN=...] [-D_SERVICEPASSWORD=...]
```

Example:

```
./updateSynchronizer 0 -D_SCHEDULE='0 24 * * * ?'
```

7.6.4 How to delete a User Synchronizer

To delete an existing user synchronizer launch the following command:

```
./deleteSynchronizer <id>
```

7.7 Management Panels

The section describes the management panels implemented in DHuS and accessible via AJS UI.

For the AJS UI, the first step is to access the "Management" area by means of the button at the right side of the Home web page.

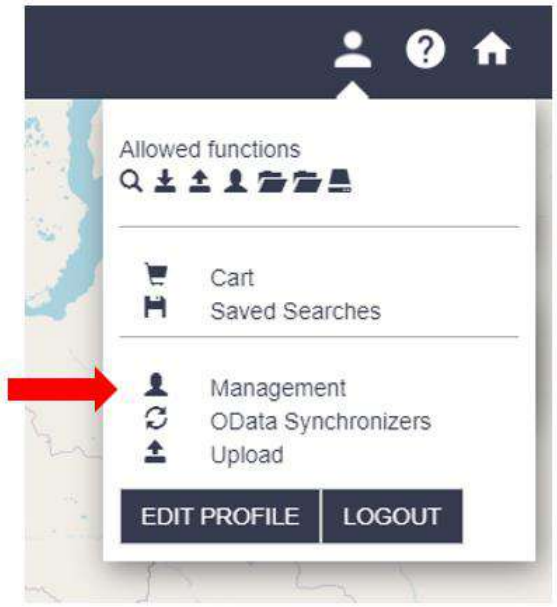

Figure 7.22 Management area (AJS)

The following four subpanels are present:

- Users
- Collections

- System
- Eviction
- Data Stores

Figure 7.23 Management Subpanel (AJS)

7.8 User Management

7.8.1 How to manage users via GUI - GDPR disabled

Access the administrator management panel as described in Section 7.7 and go to the Users panel.

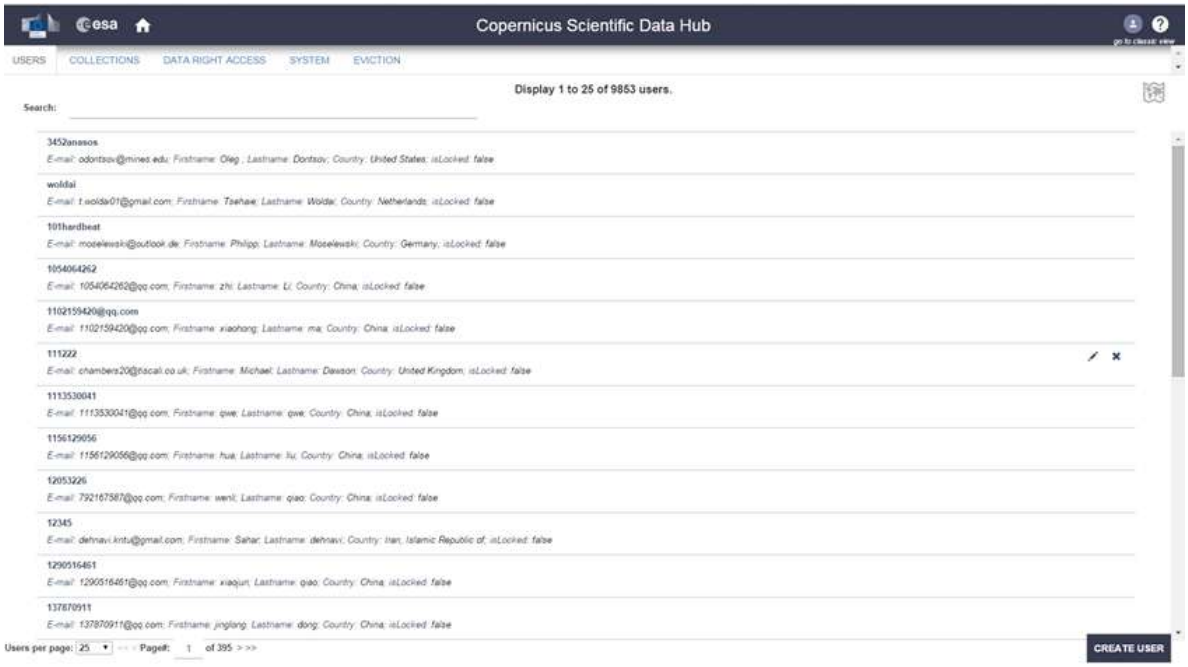

Figure 7.24 DHuS User Management Panel (AJS)

7.8.1.1 Create a new user

The Administrator shall:

- **Step 1**: Access the DHuS page;
- **Step 2**: Perform the login;
- **Step 3**: Select the Management Panel and then select the Users management panel;

- **Step 4**: Click on the "Create" button in the lower part of the User management page, which will enable the form here below.

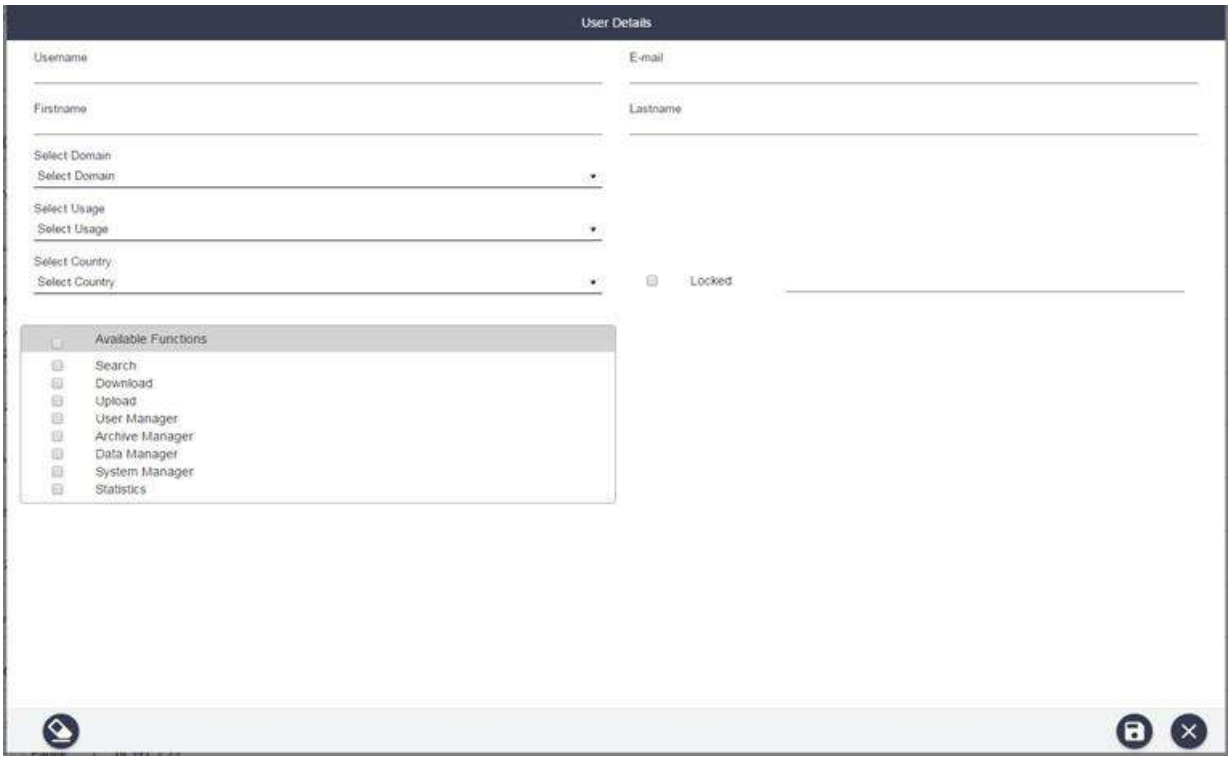

Figure 7.25 User Creation Form (AJS)

The administrator has the possibility to modify user's authorization settings and information. To modify whatever authorization setting or user information, the Administrator, before executing the following "how to" procedures, has to:

- **Step 1**: Access the DHuS page
- **Step 2**: Perform the login
- **Step 3**: Select the Management Panel
- **Step 4**: Select the Users Management Panel
- **Step 5**: Select the name of the user in the users list on the left side of the user management panel

Warning: Please note that it is not possible to create, delete or edit the profile of a user with the new "Federation User" role via GUI; instead, it is possible via OData, as shown in 7.8.2.

7.8.1.2 Lock selected user

The Administrator shall:

- **Step 1:** Click on the "locked" checkbox under the Registration form in the right side of the panel;
- **Step 2:** Indicate the reason of this locking process in the box on the right;
- **Step 3**: Click on the "save" button to complete;
- **Step 4**: The email notification service will send an e-mail to the user with his profile information (login, first name, last name, available services...) including locking notification and its relative reason, if it has been indicated.

Warning: the locking restriction and its relative reason are registered in the OData "Restriction" EntityType, under the corresponding "User". Please refer to [RD 5 RD 5] for more details.

7.8.1.3 Delete selected user

Click on the "Delete" button to delete

Figure 7.26 Update, delete users, modify user password on AJS

The email notification service will send an e-mail to the deleted user with the communication of the deletion process.

7.8.1.4 Change user password

Click on the "edit user password" checkbox and the following panel will appear:

Figure 7.27 Update, edit user password and delete users AJS

Including the new password and, then, clicking on the save button, the user password will be updated.

7.8.2 How to manage users via API – GDPR disabled

7.8.2.1 JSON ODATA representation for user configuration

Here after a json file showing the needed parameters to manage users, as an example:

```
{
"Username":"User_Name",
"Email":"E-mail_address",
"FirstName":"User_FirstName",
"LastName":"User_LastName",
"Country":"Italy",
"Phone":null,
"Address":null,
"Domain":"Climate",
"SubDomain":null,
"Usage":"Commercial",
"SubUsage":null,
"Password":"password",
"SystemRoles":{"__deferred":{"uri":"SystemRoles('DOWNLOAD')"}},
"SystemRoles":{"__deferred":{"uri": "SystemRoles('SEARCH')"}},
"SystemRoles":{"__deferred":{"uri":"SystemRoles('ARCHIVE_MANAGER')"}}
}
```

7.8.2.2 Create a new user

```
curl -H "Content-Type: application/json" -X POST --data "@/path/to/user.json" -u
${DLOGIN}:${DPASS} "http://${DHOST}/odata/v1/Users"
```

where the "`user.json`" is the name of the JSON file containing the proper configuration for the creation of the user, detailed in Section 7.8.2.1.

7.8.2.3 Lock selected user

```
curl -H "Content-Type: application/json" -X POST -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v1/LockUser?username='User_Name'&reason='This%20is%20a%20re
ason%20for%20lock.'"
```

Please note that "%20" characters should be used instead of white spaces in the description of the reason for the lock action.

7.8.2.4 Unlock selected user

```
curl -H "Content-Type: application/json" -X POST -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v1/UnlockUser?username='User_Name'&restriction='RESTRICTION
_UUID'"
```

Please note that the `RESTRICTION_UUID` to be set in the restriction field can be retrieved via the following OData query for locked users:

```
http://${DHOST}/odata/v1/Users('User_Name')/Restrictions
```

7.8.2.5 Update user profile

```
curl -H "Content-Type: application/json" -X PUT --data "@/path/to/user.json" -u
${DLOGIN}:${DPASS} "http://${DHOST}/odata/v1/Users('User_Name')"
```

where the "`user.json`" is the name of the JSON file containing the proper configuration for the creation of the user, detailed in Section 7.8.2.1. All the fields in the json file can be updated.

7.8.2.6 Delete selected user

```
curl -H "Content-Type: application/json" -X DELETE -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v1/Users('User_Name')"
```

7.8.3 How to manage users – GDPR enabled

As stated in Section 5.12, all User Management is redirected to Keycloak in case of GDPR enabled.

Despite this, the 'User Management' panel is still accessible via Graphical Interface, but all buttons are not working according to what described in Section 7.8.1. They redirect to Keycloak Administrator Console.

How to manage Users in case of GDPR activate is described in next Sections.

7.8.3.1 Create a new User

In case of GDPR activate, the following actions shall be executed by an Administrator to create a new User:

- **Step 1:** Access the Keycloak Administration Console and login as Administrator User.
- **Step 2:** Access the Realm related to the DHuS instance.
- **Step 3:** Access the 'Users' page and click on 'Add user' button.

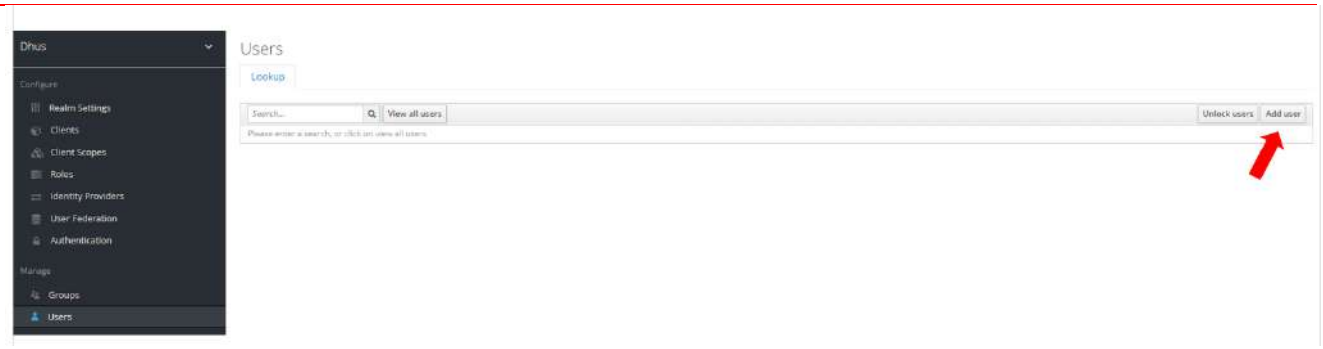

Figure 28: Keycloak 'Add user' button

- **Step 4:** In the 'Add user' panel that appears, fill the following fields and click on 'Save':
  - Username
  - Email
  - First Name
  - Last Name

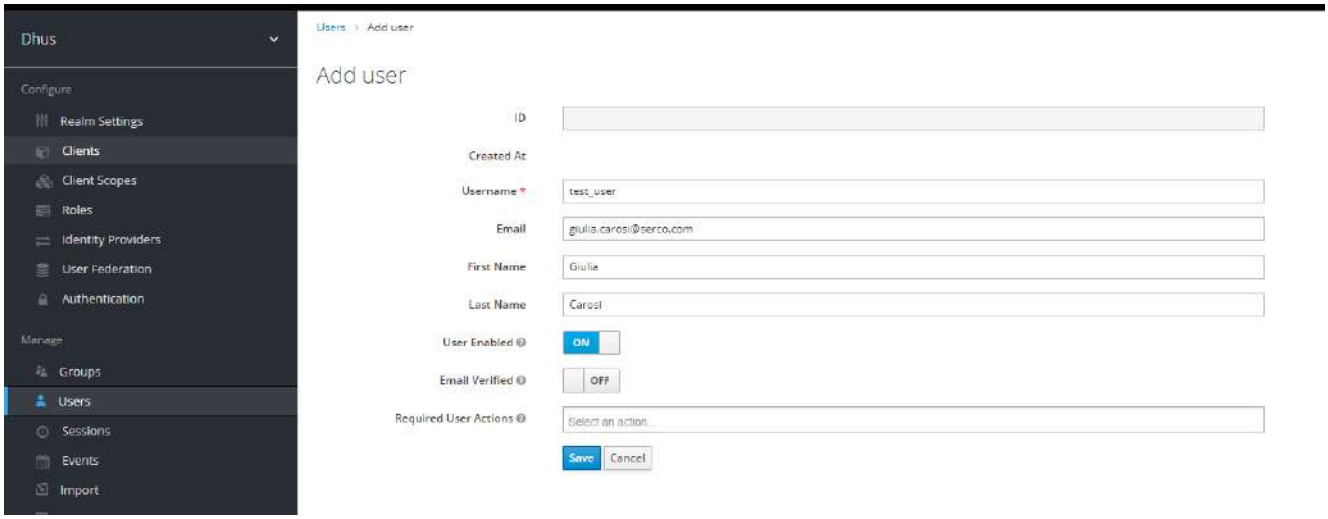

Figure 29: Keycloak 'Add user' panel

- **Step 5:** A message to confirm the User creation is displayed and the just created User's section is shown.

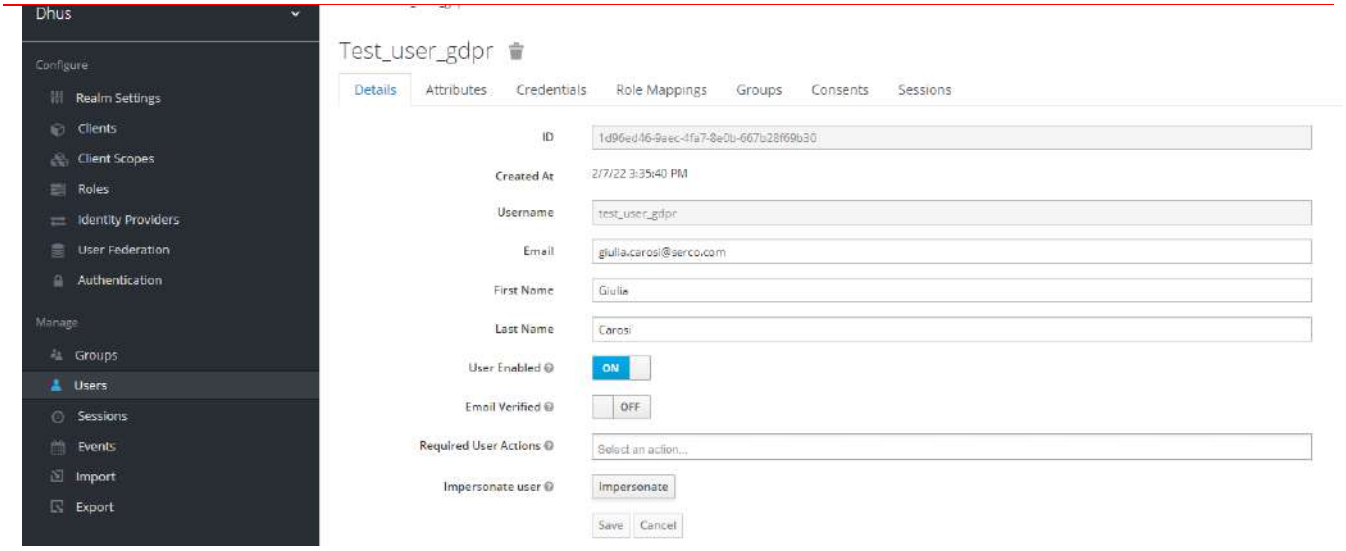

Figure 30: Keycloak User's section

- **Step 6:** Access the User's 'Attributes' tab and add the following keys setting the corresponding value:
  - country
  - domain
  - usage

  Then click on 'Save'.

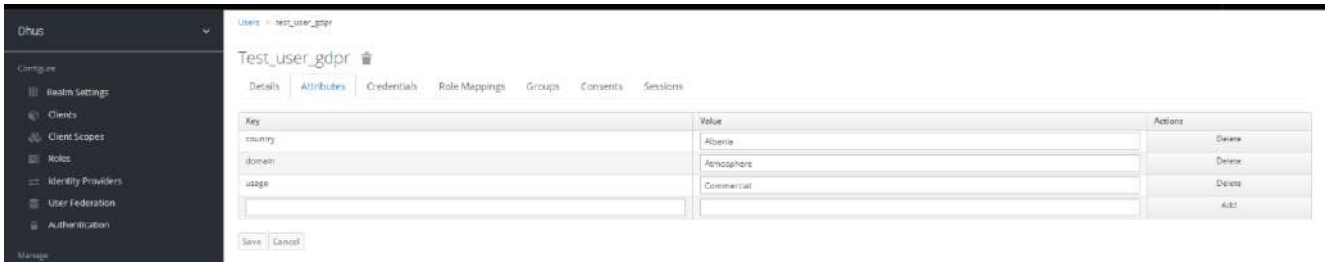

Figure 31: Keycloak User's Attributes

- **Step 7:** Access the User's 'Credentials' tab and set the User's password. Enable or not the change password at the first login, according to the Operational needs.

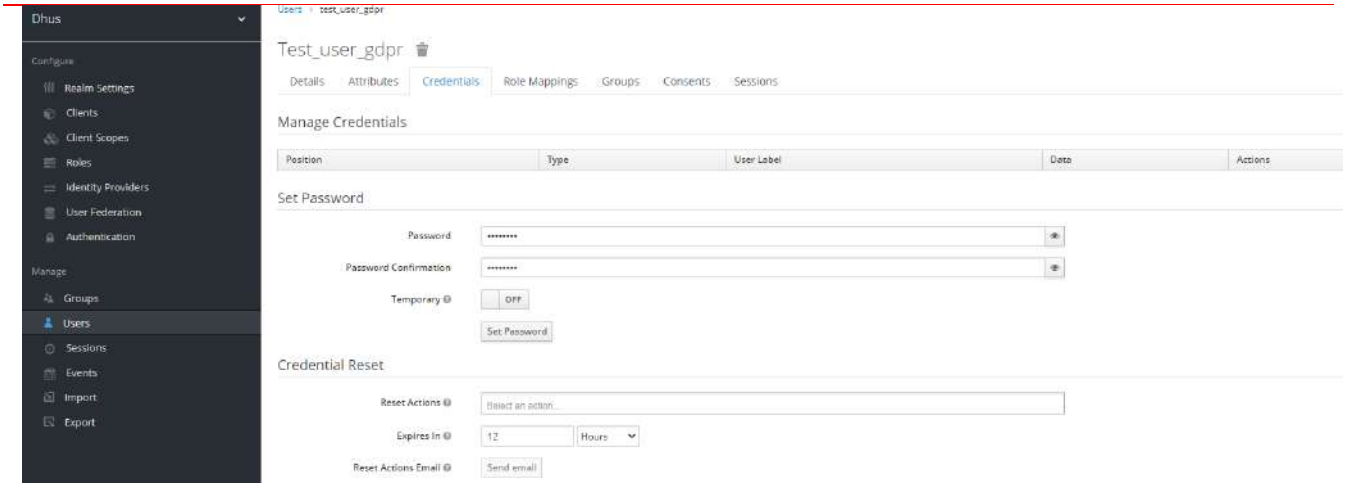

Figure 32: Keycloak User's password setup

7.8.3.2 Edit registered User

In case of GDPR activate, the following actions shall be executed by an Administrator to edit a registered User:

- **Step 1:** Access the Keycloak Administration Console and login as Administrator User.
- **Step 2:** Access the Realm related to the DHuS instance.
- **Step 3:** Access the 'Users' page and search for the User to be modified.
- **Step 4:** Click on 'Edit' button.

Figure 33: How to edit a User on Keycloak

- **Step 5:** According to which User's info has to be modified, access the related tab. In detail:
  - Details: User's email, First Name and Last Name can be modified from this panel. After the change has been applied, click on 'Save'.

    Please note that the User's username cannot be changed.
  - Attributes: The User's attributes 'country', 'usage' and 'domain' can be modified from this tab. In order to update an Attribute, its previous value has to be deleted and recreate.

- Role Mappings: This tab allows to assign to a User an additional Role w.r.t the default ones (SEARCH and DOWNLOAD). To perform this change, select the 'client' to which applied the change and assign to the User the selected role.

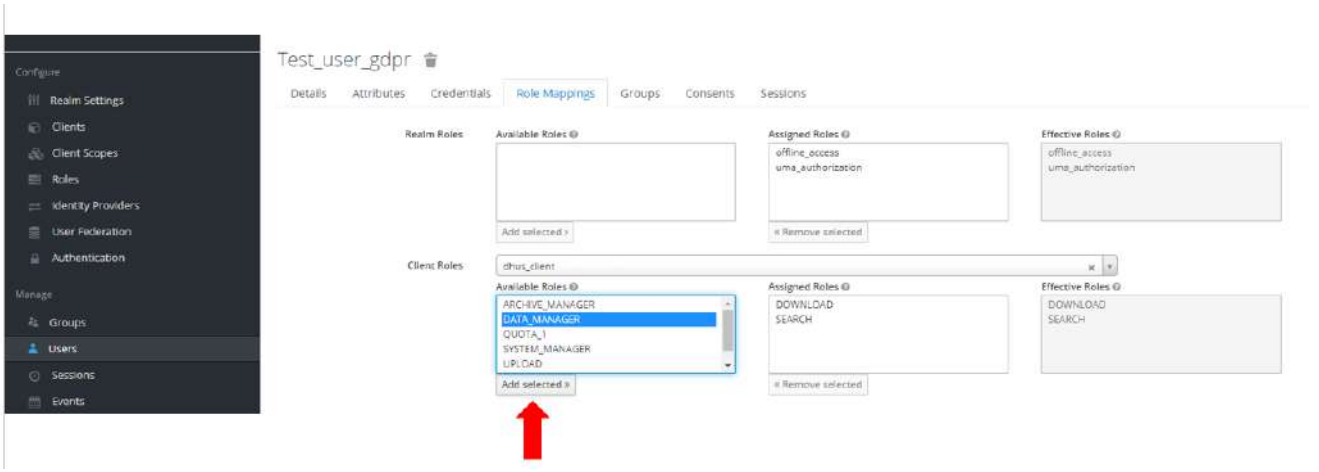

Figure 34: How to add User Role on Keycloak

7.8.3.3 Lock/Unlock selected User

In case of GDPR activate, the following actions shall be executed by an Administrator to lock/unlock a registered User:

- **Step 1:** Access the Keycloak Administration Console and login as Administrator User.
- **Step 2:** Access the Realm related to the DHuS instance.
- **Step 3:** Access the 'Users' page and search for the User to be locked.
- **Step 4:** Click on User's ID and disable the 'User Enabled' button. Then click on 'Save'.

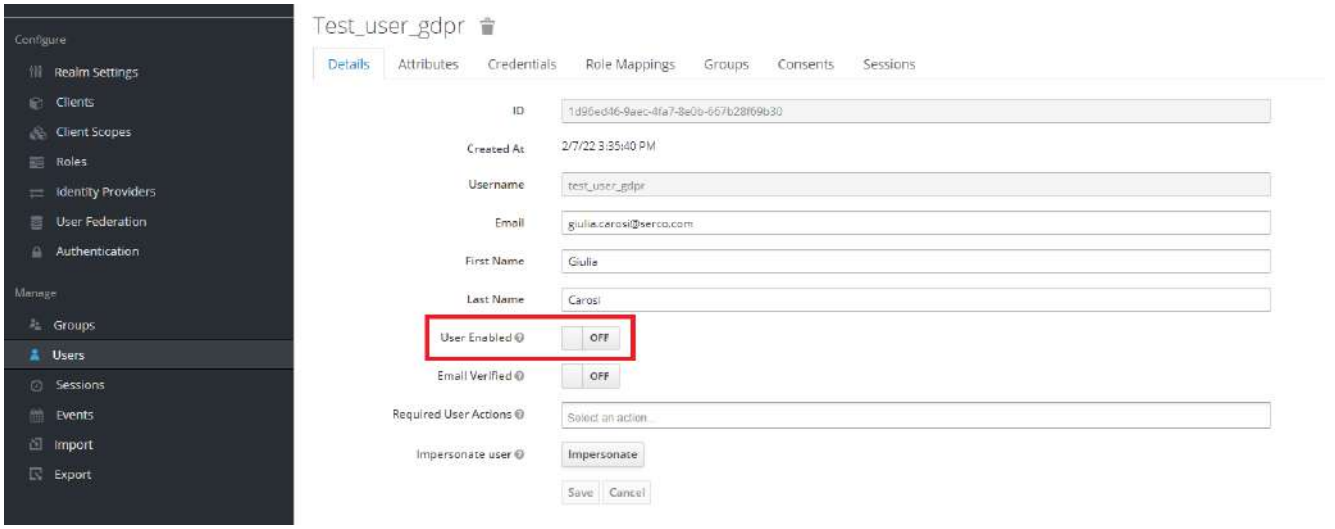

Figure 35: How to lock a User on Keycloak

- **Step 5:** The button shall be enabled, in case of User's unlock. Then click on 'Save'.

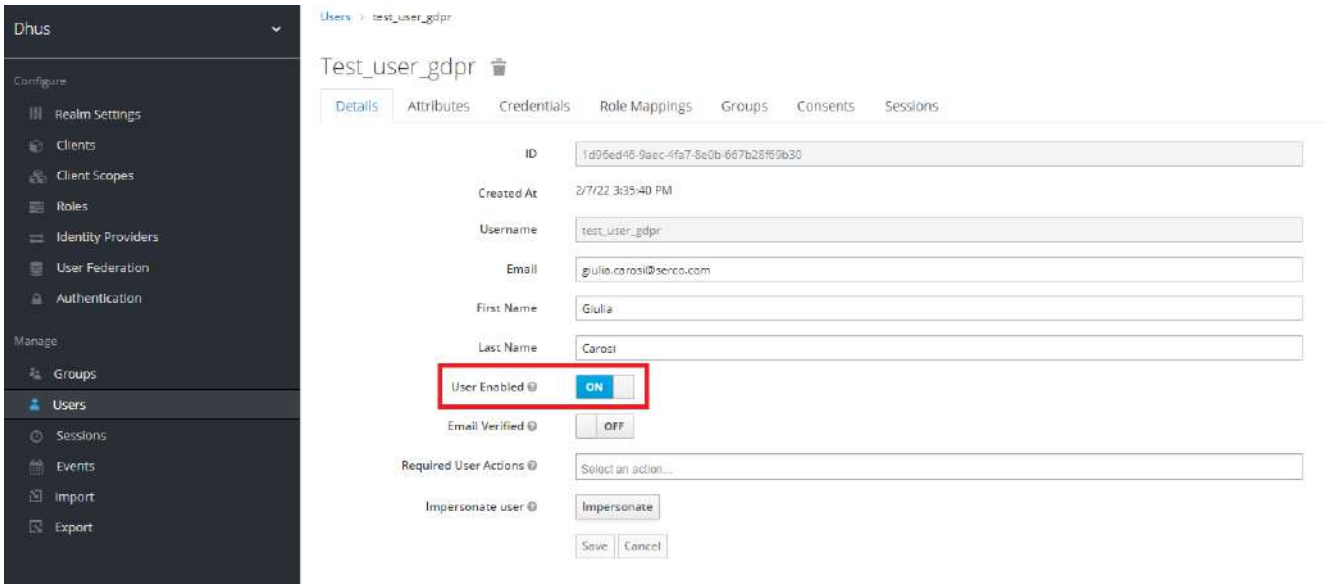

Figure 36: How to unlock a User on Keycloak

7.8.3.4 Change User password

In case of GDPR activate, the following actions shall be executed by an Administrator to change the password of a registered User:

- **Step 1:** Access the Keycloak Administration Console and login as Administrator User.
- **Step 2:** Access the Realm related to the DHuS instance.
- **Step 3:** Access the 'Users' page and search for the User whose password has to be changed.
- **Step 4:** Click on 'Edit' button and access the 'Credentials' tab. Then fill the 'Reset Password' with the new values and click on 'Reset Password'.

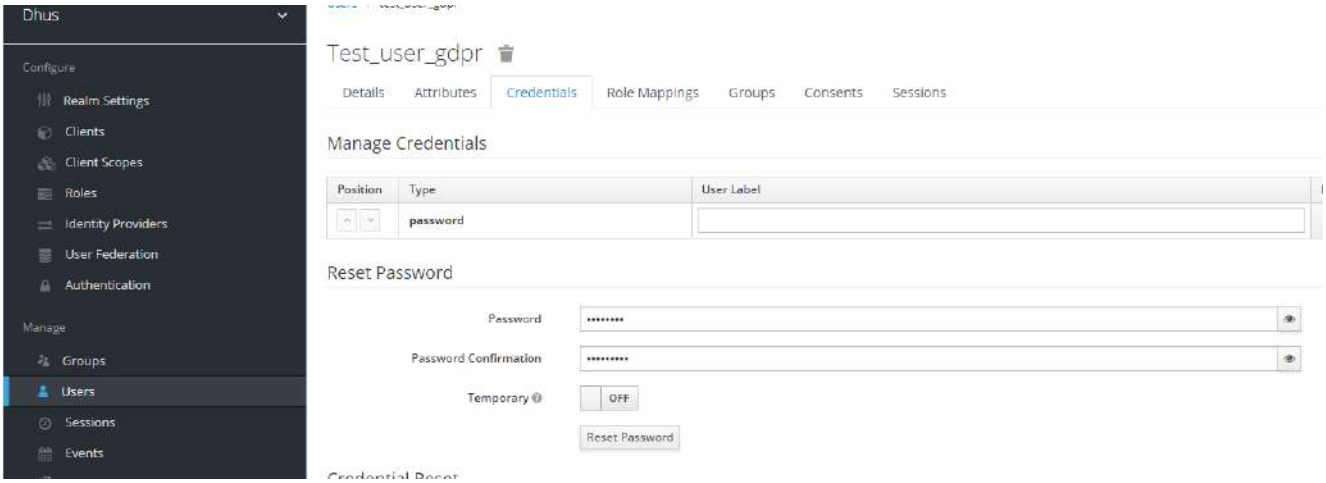

Figure 37: How to change User's password on Keycloak

7.8.3.5 Delete selected User

In case of GDPR activate, the following actions shall be executed by an Administrator to delete a registered User:

- **Step 1:** Access the Keycloak Administration Console and login as Administrator User.
- **Step 2:** Access the Realm related to the DHuS instance.
- **Step 3:** Access the 'Users' page and search for the User to be deleted.
- **Step 4:** Click on 'Edit' button and access the 'Sessions' tab. Then click on 'Log out all sessions'.

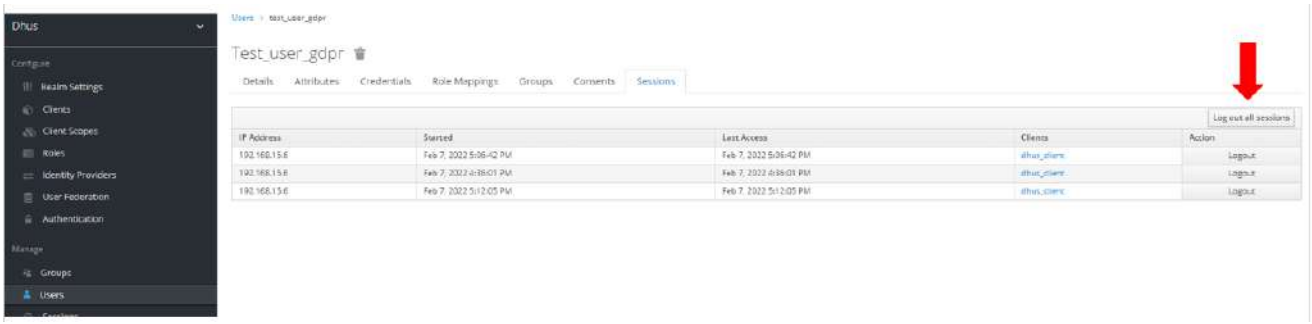

Figure 38: How to logout all User's session from Keycloak

- **Step 5:** Access again the 'Users' page and click on 'Delete' button corresponding to the selected User.

Figure 39: How to delete User on Keycloak

7.9 Collection Management

The Collection management panel lists a set of products to be attached to the collection. The selection of collections is possible by browsing the collection hierarchy on the left. To access the collection management panel, the Administrator has to click on the "collections" link, sited in the upper left side of the management panel. The collection management panel here below will open. It contains the list of collections on the left and the list of archived products on the right.

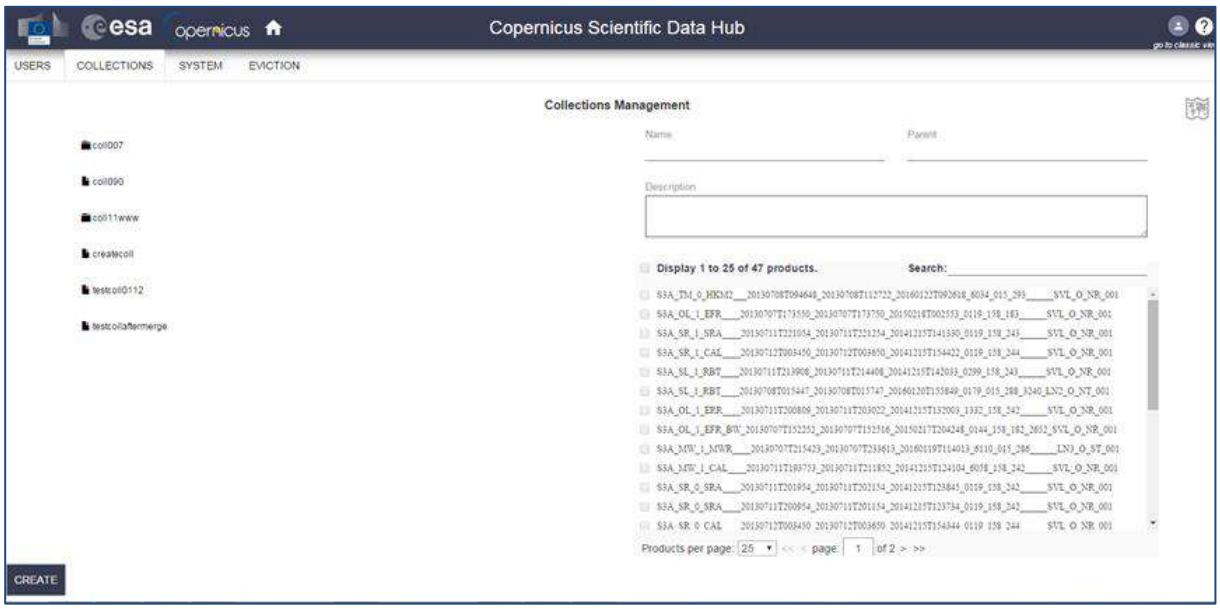

Figure 7.40 Collection Management Panel (AJS)

The collection management can be performed only after the following preliminary actions:

- **Step 1**: Access the DHuS page;
- **Step 2**: Perform the login;
- **Step 3**: Go to the collection management panel.

7.9.1 How to create a new collection

- **Step 1**: Click on the "create" button in the collection management panel;
- **Step 2**: Create collection;
- **Step 3**: Insert the collection information in the upper right side of the panel (the name of collection is mandatory), (optional) select (by clicking on the associated check box) the products to be added to the collection;
- **Step 4**: Click on the "save" button to register the new collection or click on the "cancel" button to abort the 'creation of collection' procedure.

7.9.2 How to delete a collection

- **Step 1**: Click on the collection to delete;
- **Step 2**: Click on the "delete" button.

Note that the collection management page includes a searching box. It is useful to know if a product is collected somewhere.

7.10 System Management

Access the administrator management panel as described in Section 7.7 and go to the System management panel.

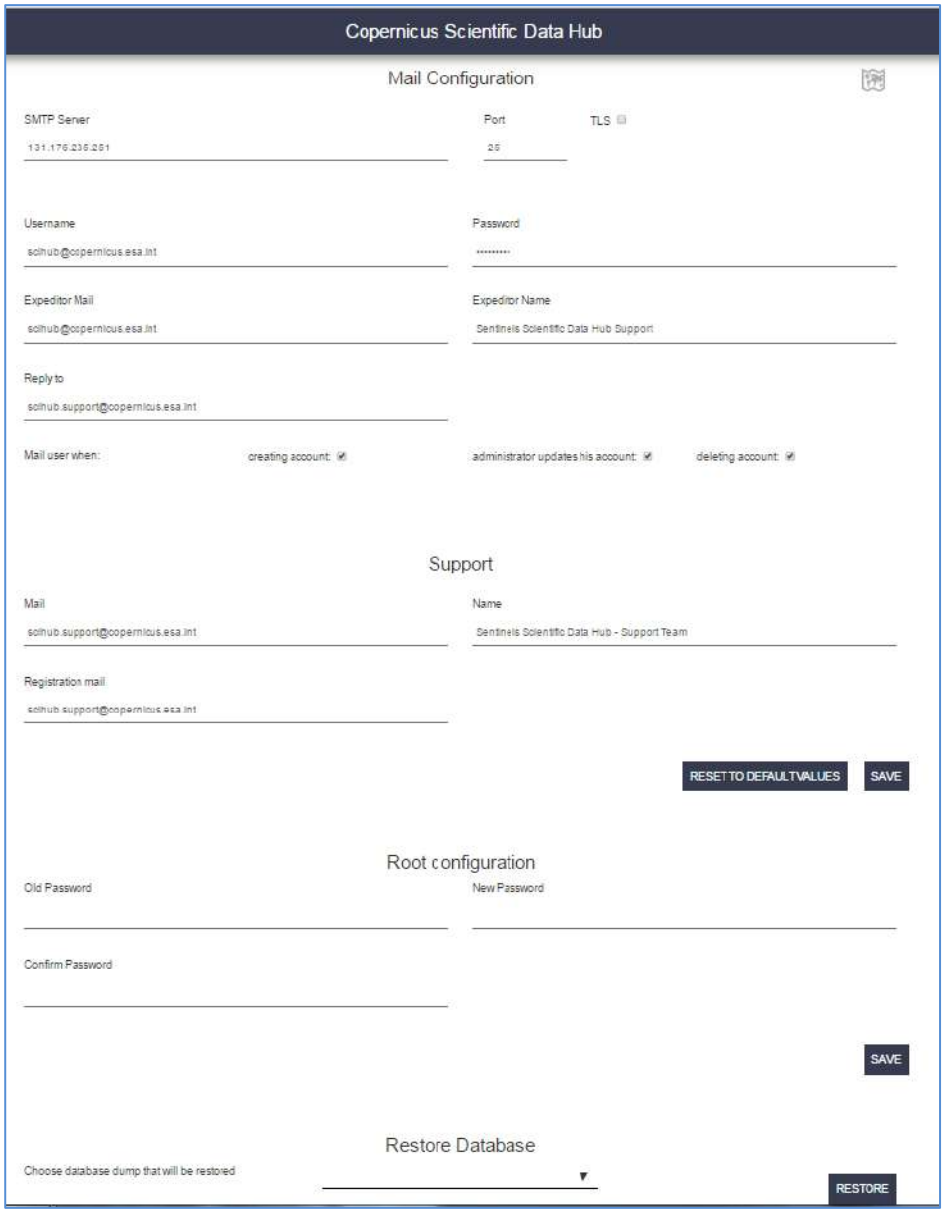

Figure 7.41 System management sections (AJS)

The following Sections detail the possible actions from this panel.

7.10.1 How to set mail configuration

In this form it is possible to configure the SMTP server address, the username, password and e-mail account details to send communications to the users.

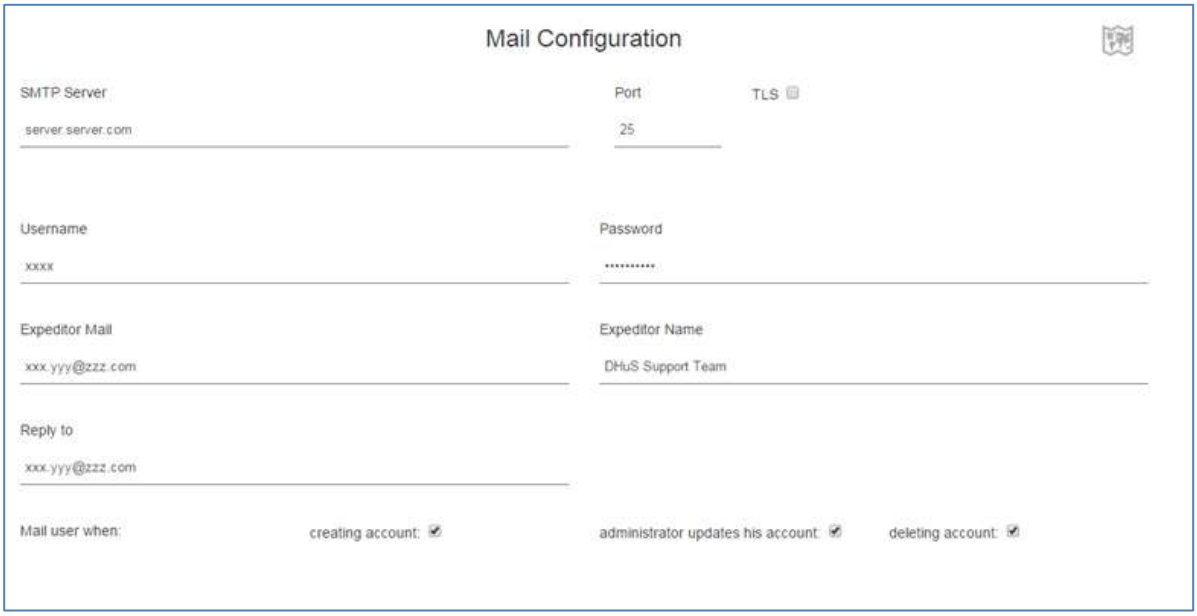

Figure 7.42 Mail configuration management Panel (AJS)

In case of GDPR activate, please refer to [RD 12] for details on how to setup SMTP service via Keycloak.

7.10.2 How to set Support information

For any support information it is possible to configure the DHuS Support Team e-mail to contact.

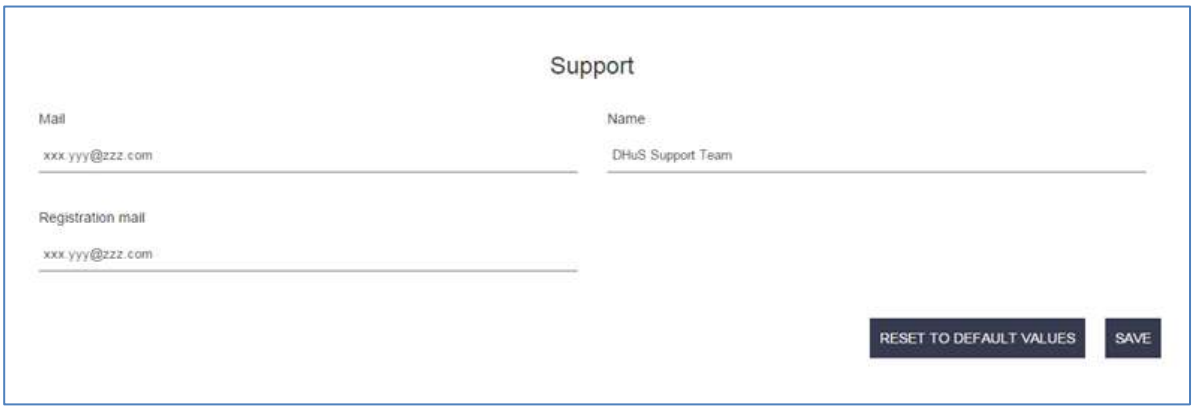

Figure 7.43 Support configuration management panel (AJS)

7.10.3 How to change root password

From this panel it is possible to change the administrator password. To do so, insert the old password, the new one and then confirm the new password.

Figure 7.44 Root configuration management panel (AJS)

In case of GDPR activate, please refer to Section 7.8.3.4 for details on how to change the Administrator User's password.

7.10.4 How to restore database

In the `dhus.xml` file it is possible to configure DHuS so that it performs a periodical dump of the database. From this panel it is possible to restore the database dump.

Figure 7.45 Restore Database panel (AJS)

To do so, perform the following steps:

- **Step 1**: Click on the drop-down menu in the "Restore Database" section: the list of available dumps will be displayed through a list of dates (date during which the dumps have been performed).
- **Step 2**: Select the desired date and then click on "RESTORE". DHuS will automatically stop and restart. Once DHuS will be up again, it will contain just the data inserted before the selected dump date.

7.10.5 How to define download quota configuration

The system allows defining a maximum number of parallel download of Sentinel Products for each user.

This can be don configuring properly the `<conf:network>` section of the dhus.xml file.

It is possible also define a white list of user for which special quota are defined, different from the ones defined for all the other self-registered users.

7.10.5.1 How to configure quota restriction for maximum number of parallel downloads

In the `<conf:network>` section of dhus.xml it is possible to configure the `SelfRegisteredChannel` section specifying the number of concurrent download allowed for each self-registered user:

```
<conf:network>
        <network:outbound>
        <network:channel name="SelfRegisteredChannel" weight="25">
                <network:defaultUserQuotas>
                    <network:maxConcurrent>2</network:maxConcurrent>
                </network:defaultUserQuotas>
        </network:channel>
```

Default value implies no quota limitation.

Please note that for a correct working mechanism, it is necessary to configure the <network:outbound> and <network:inbound> sections with the same values.

7.10.5.2 How to configure users quota restriction with specific rules

It is possible to define specific email address for which special quota rules are defined.

This can be done configuring the `PriorityChannel` sector of the `<network:outbound>` section in the dhus.xml. You can include the `UserEmailPattern` that you want to grant with special download quota rules, different from the ones defined in the `SelfRegisteredChannel`, e.g.:

```
<conf:network>
     <network:outbound>
         <network:channel name="PriorityChannel" weight="75">
             <network:classifier>
                 <network:includes>
                    <network:include>
                       <network:userEmailPattern>.*@pattern1.com</network:userEmailPattern>
                    </network:include>
                </network:includes>
             </network:classifier>
         <network:channel name="SelfRegisteredChannel" weight="25">
             <network:defaultUserQuotas>
                    <network:maxConcurrent>2</network:maxConcurrent>
             </network:defaultUserQuotas>
         </network:channel>
     </network:outbound>
```

It is possible define users with special download quota rules both via the complete e-mail address or via patterns within the address by using regular exprections, as in exemple above.Default quota value for users in the `PriorityChannel` implies no quota limitation. To set specific value of allowed concurrent download for `Prioritychannel` users it is possible including the `<network:defaultUserQuotas>` configuration within the `PriorityChannel` section:

```
<network:channel name="PriorityChannel" weight="75">
   <network:classifier>
      <network:includes>
```

```
          <network:include>
             <network:userEmailPattern>.*@pattern1.com</network:userEmailPattern>
          </network:include>
       </network:includes>
   </network:classifier>
   <network:defaultUserQuotas>
            <network:maxConcurrent>40</network:maxConcurrent>
   </network:defaultUserQuotas>
</network:channel>
```

Please note that for a correct working mechanism, it is necessary to configure the <network:outbound> and <network:inbound> sections with the same values.

In case of GDPR activate, the Users' email is no more available to set the download quota. DHuS SW has been updated in order to accept download quota limitation based on Users' Role.

Following what stated above, the <network:userEmailPattern> shall be substitute by the following entry:

```
<network:rolePattern>USER_ROLE</network:rolePattern>
```

where USER_ROLE is the Role assigned to the User affected by the defined quota.

Please note that USER_ROLE can be a new Role w.r.t those described in Section 2.4. This Role shall be defined on Keycloak and assing to the selected User. Please refer to [RD 12] and Section 7.8.3.2 for more details on how to define and assign a specific User Role.

7.11 Eviction Management

7.11.1 How to manage a Customizable Eviction via AJS GUI

7.11.1.1 Creation

In order to activate a Customizable Eviction via GUI perform the following steps:

- **Step 1**: Access the DHuS page:
- **Step 2**: Perform the login;
- **Step 3:** Access the administrator management panel as described in Section 7.7;
- **Step 4:** Select the Eviction Management and click on "CREATE EVICTION" button;
- **Step 5**: Configure the following Eviction Properties in the pop-up window, as per Figure 7.46 (please refer to Section 5.10.1 for further details):
  - *Name* – insert a unique eviction name. Must be unique, cannot be null, and cannot be updated. White space cannot be included in the Name.
  - *Maximum Evicted Products* – insert the maximum number of product to be evicted at each run
  - *Keep Period* – insert the number representing the time after which a product can be evicted. This will be the trigger of the eviction and it is computed on the `ModificationDate` parameter.
  - *Keep Period Unit* – insert the time unit via drop down menu

- *Filter* – filtering clause to be set by using the same OData syntax used on the Products entity set, restricts which products can be evicted
- *Order By* – ordering clause to be set by using the same OData syntax used on the Products entity set, defines the order in which products will be evicted
- *Target Collection* – selectable via drop down menu if it has been created previously as in Section 7.9.1.
- *Soft Eviction* – to not be ticked to perform an Hard eviction.
- Eviction Scheduling
  - *Schedule* – shall be configured according to the crontab syntax
  - *Is active* – to be ticked to activate the Eviction with the set cron scheduling

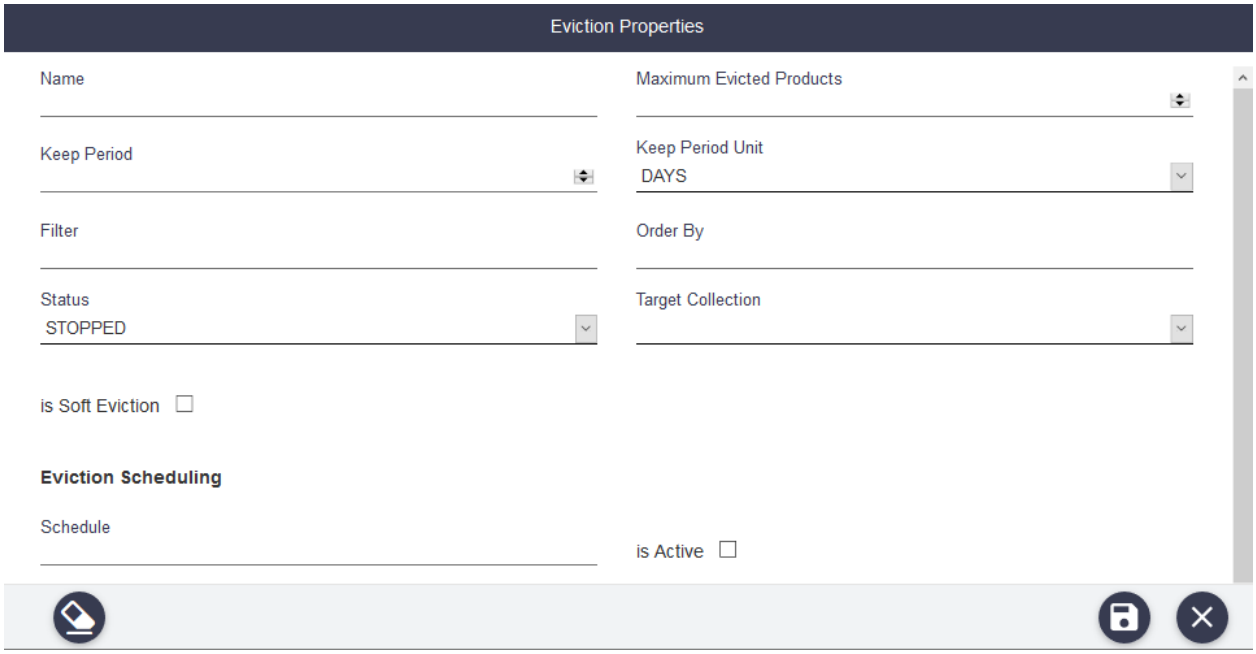

Figure 7.46 Eviction Management panel (GUI)

- **Step 5:** check that the message "EVICTION CREATED" is correctly shown in the bottom right corner of the page.

7.11.1.2 Update

- **Step 1**: In the Eviction Management panel (please refer Step 1 - Step 4 of section 7.11.1.1) move mouse under the eviction to be updated in order to display the following icons highlighted in Figure 7.47

Figure 7.47 Eviction update via GUI

- **Step 2**: click on the "pencil icon";
- **Step 3**: Modify the properties to be updated in the Eviction properties panel (Figure 7.46);
- **Step 4**: Save the updates clicking in the "floppy disk icon" (Figure 7.46);
- **Step 5**: check that the message "EVICTION UPDATED" is correctly shown in the bottom right corner of the page.

7.11.1.3 Deletion

In order to delete a Customizable Eviction via GUI perform the following steps:

- **Step 1**: In the Eviction Management panel (please refer Step 1 - Step 4 of section 7.11.1.1) move mouse under the eviction to be deleted in order to display the icons shown in Figure 7.46.
- **Step2**: click on the "cross icon" (refer to Figure 7.48)

Figure 7.48 Eviction delete via GUI

- **Step 3:** check that the message "EVICTION DELETED" is correctly shown in the bottom right corner of the page.

7.11.2 How to manage a Customizable Eviction via OData API

Evictions can be inspected and configured using the Evictions entity set on the DHuS OData v4 API:

```
http://SERVICE_URL/odata/v2/Evictions
```

Only users with the System Manager role on OData can access Evictions.

7.11.2.1 JSON ODATA representation for a customizable eviction

Here after a JSON ODATA representation as an example for configuring the customizable eviction property, listed in 5.10.1:

```
{
"Name":"Soft_param",
"KeepPeriod":1,
"KeepPeriodUnit":"DAYS",
"MaxEvictedProducts":2,
```

```
"Filter": "startswith(Name,'S2')",
"OrderBy": "ModificationDate asc",
"TargetCollection":null,
"SoftEviction":true,
"Status":"STOPPED",
"Cron":{"Active":false,"Schedule":"0 0/5 * * * ?"},
"SafeMode":false,
"TargetDataStore":"S2_ParamPDGSDataStore",
"BaseDate":"modificationDate"
}
```

7.11.2.2 Creation

The following instruction has to be performed via command line in order to create an eviction:

```
curl -H "Content-Type: application/json" -X POST --data "@/path/to/eviction.json"
-u ${DLOGIN}:${DPASS} "http://${DHOST}/odata/v2/Evictions"
```

where the "`eviction.json`" is the name of the JSON file containing the proper configuration for the creation of the Customizable Eviction.

7.11.2.3 Update

The following instruction has to be performed via command line in order to update an eviction:

```
curl -H "Content-Type: application/json" -X PATCH -data "@/path/to/eviction.json"
-u ${DLOGIN}:${DPASS} "http://${DHOST}/odata/v2/Evictions('Eviction_Name')"
```

where the "`eviction.json`" is the name of the JSON file containing the proper configuration for the update of the Customizable Eviction.

7.11.2.4 Deletion

The following instruction has to be performed via command line in order to delete an eviction:

```
curl -X DELETE --basic -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v2/Evictions('Eviction_Name)"
```

7.11.2.5 Stop

The following instruction has to be performed via command line in order to stop the current eviction run:

```
curl -H "Content-Type: application/json" -X POST --data '{}' -u ${DLOGIN}:${DPASS}
"http://localhost:8081/odata/v2/Evictions/OData.DHuS.StopEviction"
```

Please note that StopEviction action only serves to stop the currently running eviction and if the eviction is set as "Active" at the next run the eviction will work again.

7.11.2.6 Queue

The following instruction has to be launched to perform a manual activation of the eviction (QueueEviction action on OData):

```
curl -H "Content-Type: application/json" -X POST --data '{}' -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v2/Evictions('Eviction_Name')/OData.DHuS.QueueEviction"
```

7.11.2.7 Cancel

This OData action will cancel an eviction that has been queued, setting its status to "CANCELED", meaning this eviction will not be executed once the previously running evictions are finished.

```
curl -X POST --basic -u ${DLOGIN}:${DPASS}
"http://${DHOST}/odata/v2/Evictions('Eviction_Name')/OData.DHuS.CancelEviction"
```

Please note that CancelEviction action serves to cancel the next queued Eviction but not the cron: if the eviction is set as "Active" at the next run the eviction will work again.

7.12 Auxiliary Data Files

7.12.1 How to configure ADF retrieval chain

Pre-requisites

The following instances are needed to set up the Auxiliary Data Files retrieval chain:

- DHuS instance containing Sentinel-3 Products. Users will access to ADF Search feature from OWC GUI installed on this DHuS instance.
- DHuS instance containing Sentinel-3 Auxiliary Data Files. It is recommended to use the same distribution version of DHuS instance containing Sentinel-3 Products. This DHuS instance has to be accessible to DHuS instance containing Sentinel-3 Products via HTTP connection.

Procedure

- **Step 1:** Create an account with search and download rights (e.g. username: adfsearch, password adfsearch) on the DHuS instance containing Sentinel-3 Auxiliary Data Files.
- **Step 2:** Connect to the machine where DHuS instance containing Sentinel-3 Products is installed and check that the application is not running (`ps -eaf | grep dhus`)
- **Step 3:** Edit file `etc/conf/ExternalDHuS.properties` adding the following parameters needed to connect with DHuS instance containing ADF:
  - **external_dhus_host**: hostname or IP address of DHuS instance containing Sentinel-3 auxiliary files, ending with slash ('/') character (e.g. http://[DHuS_address]/)

- **external_dhus_username**: username of the account on DHuS instance containing Sentinel-3 auxiliary files (e.g. adfsearch)
- **external_dhus_password**: password of the account on DHuS instance containing Sentinel-3 Auxiliary files (e.g. adfsearch)

An example of configuration is the following:

```
external_dhus_host=http://192.168.10.79:8091/

external_dhus_username= adfsearch

external_dhus_password= adfsearch
```

- **Step 4:** Start the DHuS as single instance entering the following command in the installation directory:

```
nohup /bin/bash ./start.sh &
```

- **Step 4:** Edit the file /DHUSDIR]/var/tomcat/webapps/ROOT/config/`appconfig.json` and enable the Auxiliary File Search feature:

```
"adf "=true
```

7.13 User details information

It is possible to visualize user details via OData (see [RD 5]) by accessing to the entity "Users" at:

```
https://DHuS_URL/odata/v1/Users
```

Users with the user administration rights are allowed to detail of all the users registered in the Hub, whereas normal users are allowed to visualize only their own.

Please refer to [RD 5] for further details about the "Users" OData entities and its information such as system roles, connections, restrictions and user cart.

Different is the behaviour in case of GDPR activate.

The Users OData Entity is still accessible, but all Users, included the Administrator one, can view only own User info. Moreover, only the following info are reported:

- Username: the info is hashed.
- Created: This information represents the datetime of the User creation.
- Restrictions: The applied User's Restriction are available by navigating the Users Entity.
- SystemRoles: The assigned User's Role are available by navigating the Users Entity.

7.14 Connections information

Administrators can visualize all the connections managed by DHuS via OData if `server.xml` file has been configured properly (8.2). In fact, every user request is available in the "Connections" entity (see [RD 5]). Moreover it is possible to monitor the activity of a specific user visualizing the requests sent by him/her at:

```
https://DHuS_URL/odata/v1/Users('Username')/Connections
```

Please refer to [RD 5] for the list of information exposed for each connection stored in the DHuS cache.

7.14.1 How to identify Active users

Administrators can list active users on DHuS through OData Connections entity. The following query shows all the users in status PENDING, i.e. users whose requests are being processed by DHuS at the time of the verification:

```
https://DHuS_URL/odata/v1/Connections?$filter=Status%20eq%20%27PENDING%27&$expand=
User
```

7.15 Network information

OData exposes to administrators some information about the network used by the Hub. To access such info it is necessary to connect via OData to the "Networks" entity at:

```
https://DHuS_URL/odata/v1/Networks
```

Please refer to [RD 5] for the information exposed for each Network stored in the DHuS cache and its Statistics.

7.16 DataStore Management

7.16.1 JSON ODATA representation for HFSDataStore

All the HFSDataStore properties listed in 6.1.2 can be configured via the following JSON OData representation:

```
{
"@odata.type":"#OData.DHuS.HFSDataStore",
"Name": "HFSDataStore_Name",
"Restriction": "none",
"Priority": 0,
"MaximumSize": 1000,
"CurrentSize": 0,
"AutoEviction": true,
"Path":"/path/to/incoming/folder",
"MaxFileDepth": 10,
"MaxItems": 1024
}
```

7.16.2 JSON ODATA representation for Openstack DataStore

All the Openstack DataStore properties listed in 6.1.2 can be configured via the following JSON OData representation:

```
{
```

```
"@odata.type":"#OData.DHuS.OpenStackDataStore",
"Name": "Openstack_DataStore_Name",
"Restriction": "none",
"Priority": 0,
"MaximumSize": 1073741824,
"CurrentSize": 0,
"AutoEviction": false,
"Provider": "openstack-swift",
"Identity": "Default:project_name:username",
"Credential": "password",
"Url": "https://auth.cloud.ovh.net/v3",
"Region": "SBG3",
"Container": "Container_Name"
}
```

7.16.3 DataStore operations

The DataStores entity set supports CRUD (Create, Read, Update, Delete) operations on DataStores and allows to manage the configuration of DataStores within the DHuS at runtime.

An individual DataStore can be accessed using its reference property, which is the Name:

```
[DHUS_HOST]/odata/v2/DataStores('DataStore_Name')
```

In the following Sections command line samples, using cURL commands, are provided in order to manage the DataStore creation, update and deletion at runtime. When an action related to DataStore entities is performed, it is reported at real-time on the dhus.xml file.

DataStore can be also created directly on dhus.xml file before the first start of the DHuS service. Please refer to Section 8.1 for details about DataStores configuration on dhus.xml file.

Warning: please note that white spaces are not allowed in the DataStore name.

7.16.3.1 How to create a DataStore

The following instruction has to be performed via command line in order to create a DataStore:

```
curl -H "Content-Type: application/json" -X POST --data
"@/path/to/file/DataStore.json" -u ${DLOGIN}:${DPASS}
"http://localhost:8081/odata/v2/DataStores"
```

where the "`DataStore.json`" is the name of the JSON file containing the proper configuration for the creation of the DataStore. Refer to:

- Section 7.16.1 for example related to HFS DataStore;

- Section 7.16.2 for example related to Openstack DataStore.

Warning: Please note that white spaces are not allowed in the DataStore name.

7.16.3.2 How to Update a DataStore

The following instruction has to be performed via command line in order to update a DataStore:

```
curl -H "Content-Type: application/json" -X PATCH --data
"@/path/to/file/DataStore.json " -u ${DLOGIN}:${DPASS}
"http://localhost:8081/odata/v2/DataStores('DataStore_Name')"
```

where the "`DataStore.json`" is the name of the JSON file containing the updates and "`DataStore_Name`" is the name of the DataStore to be updated.

7.16.3.3 How to Delete a DataStore

```
curl -u ${DLOGIN}:${DPASS} -X DELETE --basic
"http://localhost:8081/odata/v2/DataStores('DataStore_Name')"
```

where "`DataStore_Name`" is the name of the DataStore to be deleted.

7.16.3.4 How to list products stored in a DataStore

Administrators can list products stored in a DataStore performing the following OData query:

```
http://localhost:8081/odata/v2/DataStores('DataStore_Name')/Products
```

Products are listed trough their `Id` property.

7.17 OpenStack storage command line samples

In this section, we report some OpenStack storage command line samples, using curl & jq commands:

- Get token and swift url for specific user / project (tenant)

  ```
  curl -s -d '{"auth": {"tenantName": "<tenantName>",
  "passwordCredentials":{"username": "<username>", "password":
  "<password>"}}}' -H "Content-type: application/json"
  https://keystone9915.openstack.ovh.net:35358/v2.0/tokens | jq '{url:
  .access.serviceCatalog[] | select(.name=="swift") | .endpoints[].publicURL,
  token: .access.token.id}'
  ```

- List containers in the account

  ```
  curl -i -H "X-Auth-Token: <token>" <url> -X GET
  ```

- Create container

  ```
  curl -i -H "X-Auth-Token: <token>" <url>/<container_name> -X PUT
  ```

- Info about container content

  ```
  curl -i -H "X-Auth-Token: <token>" <url>/<container_name> -X HEAD
  ```

- Delete container

  ```
  curl -i -H "X-Auth-Token: <token>" <url>/<container_name> -X DELETE
  ```

  Please note that a container can be deleted only if it is empty.

- Info about object in a container

  ```
  curl -H "X-Auth-Token: <token>" <url>/<container_name>/<object_name> --head
  ```

- Delete object

  ```
  curl -i -H "X-Auth-Token: <token>" <url>/<container_name>/<object_name> -X DELETE
  ```

7.18 NetCDF Temporary Cache setting

7.18.1 How to disable the Cache

The following option should be added in the `start.sh` file to disable the cache, if needed:

- `-Dfr.gael.streams.cache.inactive=true`

7.18.2 How to configure the Cache

7.18.2.1 Default cache configuration

The default configuration of the cache mechanism is the following:

- 1000 maximum cached files;
- 10 GB maximum total size.

7.18.2.2 Customized cache configuration

Since the temporary cache files can take a substantial amount of disk space for large products, it is possible:

- to choose the location of the cache;
- to limit the number of files in the cache;
- to limit the size of files in the cache.

Location

The location of the temporary cache folder can be configured by setting the Java system property "fr.gael.streams.tmpdir" to the path of the desired folder.

This system property is set using -Dfr.gael.streams.tmpdir in the `start.sh` startup script of the DHuS instance as follows:

- `-Dfr.gael.streams.tmpdir="/my/temp/file/location" \`

The default temporary cache folder is the Tomcat tmp folder contained in the DHuS var directory.

Size management

The size of the cache can be managed in two ways:

- Limit the number of files that can be created in the cache
- Limit the total size of the cache

Temporary cache files have no expiration date and will remain until they are evicted due to size management limits. These limits can be configured in a `temporary_files_ehcache.xml` file that should be created (if not present) in the `etc` folder of the DHuS instance.

Here follows an example of `temporary_files_ehcache.xml`.

```
<ehcache name="TemporaryFilesCacheManager">
<cache name="TemporaryFilesCache"
maxEntriesLocalHeap="1000"
maxBytesLocalDisk="10000000000" >
<cacheEventListenerFactory
class="fr.gael.streams.TemporaryFilesCacheListener" />
</cache>
</ehcache>
```

The `maxEntriesLocalHeap` limits the number of temporary cache files while the `maxBytesLocalDisk` limits the total size of the cache in bytes.

In order to configure the `maxBytesLocalDisk` parameter with the appropriate size preventing errors due to cache too small, please refer to the following formula:

```
maxBytesLocalDisk = corePoolSize*maximumSize*1.3
```

where `corePoolSize` is the value set in the dhus.xml file of the DHuS instance performing ingestion and `maximumSize` is the maximum known size of the ingested products. The result has to be expressed in bytes.

7.18.2.3 How to configure the shared cache

DHuS deployed in Scalability 2.0 mode support the sharing of netCDf temporary cache using Ehcache and Java RMI (Remote Method Invocation) providing remote communication between DHuS instances (nodes).

The procedure below describes the steps needed to share the cache folder. It assumes that all DHuS instances are stopped.

Step ID	User	Tasks and Commands
1.	Standard user	Create a folder where the temporary cache files will be stored and shared between all DHuS nodes, making sure this folder is achievable by all the DHuS instances (nodes).
2.	Standard user	Configure the location of the netCDF cache folder at Step 1 in the `start.sh` as in 7.18.2.

3.	Standard user	Access the DHuS `etc` configuration folder in the machine where the selected DHuS instance (node) has been installed and perform a backup of the existing `temporary_files_ehcache.xml` file. The new file (named "temporary_files_ehcache.xml") will contain the configuration that will overwrite the default one.
		`> cd <DHuS_INST_DIR>/etc`
		`> cp temporary_files_ehcache.xml temporary_files_ehcache.xml.bk`

4.	Standard user	Open the `temporary_files_ehcache.xml` file and set the properties of `cacheManagerPeerProviderFactory` attribute in order to use manual peer discovery, as follows:
		`<cacheManagerPeerProviderFactory`
		`class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"`
		`properties="peerDiscovery=manual, rmiUrls=<RMI_URLS>"`
		`propertySeparator=","/>`
		Manual peer configuration requires the IP address and port of each listener to be known. Peers cannot be added or removed at runtime.
		The `rmiUrls` property contains the list of all cache peers for all cache listener (detailed via IP address and port); cache peers are separated by means of "|" character. Do not include the server being configured in the list.
		Refer to Step 7. of this procedure for an example containing `cacheManagerPeerProviderFactory` attribute setting.

5.	Standard user	Configure the CacheManagerPeerListener listening for messages from peers to the current CacheManager. In the `cacheManagerPeerListenerFactory` attribute set properties as follows:
		`<cacheManagerPeerListenerFactory`
		`class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"`
		`properties="hostName=<HOST>,port=<PORT>,remoteObjectPort=<RO_PORT>,socketTimeoutMillis=<TIMEOUT>"/>`
		Please note that:
		• `hostname` is the hostname/IP of the host the listener is running on;
		• `port` is the port the listener listens on;
		• `remoteObjectPort` is the port number on which the remote objects bound in the registry receive calls;

<table>
<tr><td></td><td></td><td>

- `socketTimeoutMillis` is the TCP/IP Socket timeout when waiting on response (i.e. number of seconds client sockets will wait when sending messages to this listener until they give up).

Please refer to Step 7. of this procedure for an example of `cacheManagerPeerListenerFactory` attribute setting.
</td></tr>
<tr><td>6.</td><td>Standard user</td><td>Add the following properties in the <cache> </cache> section:

```
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true" />

<bootstrapCacheLoaderFactory
class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory
"                       properties="bootstrapAsynchronously=true"
propertySeparator="," />
```
</td></tr>
<tr><td>7.</td><td>Standard user</td><td>Please find below an example of <code>temporary_files_ehcache.xml</code> file.

```
<ehcache name="TemporaryFilesCacheManager">

<!-- Manual discovery -->
<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFac
tory"
properties="peerDiscovery=manual,rmiUrls=//172.30.246.20:40001/Te
mporaryFilesCache|//172.30.246.46:400001/TemporaryFilesCache"
propertySeparator=","/>

<!-- create cluster listener on port 40001 -->
<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFac
tory"
properties="hostName=172.30.246.11,port=40001,remoteObjectPort=99
99,socketTimeoutMillis=2000" />

<cache name="TemporaryFilesCache" eternal="false"
overflowToDisk="false"
diskPersistent="false"
diskExpiryThreadIntervalSeconds="0"
maxEntriesLocalHeap="1000"
maxBytesLocalDisk="10000000000"
memoryStoreEvictionPolicy="FIFO">

<cacheEventListenerFactory
class="fr.gael.streams.TemporaryFilesCacheListener" />

<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true" />

<bootstrapCacheLoaderFactory
class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory
"
properties="bootstrapAsynchronously=true"
propertySeparator="," />
</cache>

</ehcache>
```
</td></tr>
</table>

8.	Standard user	Repeat Steps 2-8 for all the DHuS nodes.

7.19 OSM Search Engine Interface settings

7.19.1 How to disable Nominatim service

The OSM Search Engine Nominatim service can be disabled removing the following block from dhus.xml file:

```
<search:geocoder url="http://nominatim.openstreetmap.org">
      <search:nominatim boundingBox="false" maxPointNumber="50" />
      <search:geoname username="dhus" />
</search:geocoder>
```

7.20 GDPR Activation

According to the Technical Note [RD 11] the DHuS SW configuration has been updated to cope with GDPR requirements.

Please find below listed steps to follow to properly set up the DHuS and let a proper interface with Keycloak. Please note that these steps are related to those done on Keycloak side and detailed on [RD 12].

1) In the dhus.xml, activate the GDPR by adding the following section:

```
    <conf:system>
        <system:gdpr active="true" SAMLUserId="samlId"/>
        …
    </conf:system>
```

2) In the appconfig.json file, activate the GDPR and configure the related parameters:

```
{
...
"settings":{
...
"gdpr": {
    "enabled": true,
    "showUsername": false,
    "signupUrl":        "https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-
actions/registration?client_id=<CLIENT>",
```

```
    "forgotPasswordUrl":
"https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-actions/reset-
credentials?client_id=<CLIENT>",
    "editProfileUrl":   "https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-
actions/authenticate?client_id=<CLIENT>",
    "adminConsoleUrl":
"https://<KEYCLOAK_ADDRESS>/auth/realms/master/protocol/openid-
connect/auth?client_id=security-admin-console"
}
...
}
}
```

3) Create the dhusKeystore.jks file running the following command:

```
keytool -genkeypair -keysize 2048 -keyalg RSA -alias myKS -keypass pwDpwD -keystore
dhusKeystore.jks
```

Please note that the 'alias', 'keypass' and 'keystore' values can be different from those shown in the command. Only the extension of the 'keystore' key shall be .jks.

Moreover, the following command can be used to check the generated certificate and monitor its duration:

```
keytool -list -v -keystore dhusKeystore.jks
```

4) Add in the start.sh configuration file the following parameters:

```
-Ddhus.saml.idp.name
-Ddhus.saml.idp.url
-Ddhus.saml.sp.id
-Ddhus.saml.keystore.file
-Ddhus.saml.keystore.storePass
-Ddhus.saml.keystore.defaultKey
-Ddhus.saml.keystore.defaultPassword
```

Please refer to [RD 11] and Section 8.3 for the meaning of these parameters.

5) If your DHuS instance is exposed through a proxy, add the following Valve in the server.xml configuration file:

```
<Valve className="org.apache.catalina.valves.RemoteIpValve"
   remoteIpHeader="x-forwarded-for"
   proxiesHeader="x-forwarded-by"
   protocolHeader="x-forwarded-proto" />
```

8. DHuS configuration file parameters

8.1 dhus.xml

This section explains in detail the parameter contained in the `dhus.xml` configuration file.

The whole DHuS configuration is contained inside the `<configuration></configuration>` tag.

In 11 is reported the complete dhus.xml of the DHuS distribution described in this document, as it is before the DHuS start without having configured Eviction, Synchronizer, DataStore and FileScanners. In the dhus.xml file coming from the DHuS distribution, additional sections related to SmartSynchronizers and Sources are present; please note that this kind of feature is not fully supported in the DHuS release covered by this Administration Manual version. As a consequence, it is recommended to avoid the use of this feature.

Please note that since DHuS 0.14.X branch, after the first DHuS start, dhus.xml file is automatically reshaped by the DHuS.

It is recommended to configure a folder where DHuS will put its temporary files. Here follow details about the workingDir attribute, necessary for this purpose:

Parameter	Description	Syntax
`workingDir`	Folder where DHuS puts its temporary files. This folder is automatically deleted when the DHuS is switched off. Currently, DHuS uses this folder during ingestion. Avoid to set it as the DHuS installation folder.	`workingDir="/path/to/working/directory"`

Groups of configurable options are present inside the `<configuration></configuration>` tag:

- `crons`
- `messaging`
- `network`
- `products`
- `search`
- `server`
- `system`
- `DataStores`
- `scanners`
- `synchronizers`
- `evictions`

The following Sections describe the specific parameters for each Group, explaining how to configure them.

8.1.1 crons

[Groups]/ Parameter	Description	Usage
`[crons]/ archiveSynchronization`	Cron used to synchronize local archive (whose path is the one specified in the parameter "archive" in the "system" group). In case of eviction, please schedule it after the eviction.	Activate it and specify the schedule.
`[crons]/cleanDatabase`	Cron used to clean database, like removing old statistics or old not confirmed users.	Activate it and specify the schedule. The parameter `statistics` and `tempUsers` define, respectively, for how long (in days) the system have to keep statistics and how many time (in days) the users have to confirm their registration request
`[crons]/dumpDatabase`	Cron used to dump database	Activate it and specify the schedule.
`[crons]/cleanDatabaseDump`	Cron used to clean database dumps.	Activate it and specify the schedule. The parameter keep defines how many dumps are stored.
`[crons]/searches`	Cron used to execute user saved searches and send results to users	Activate it and specify the schedule.
`[crons]/sendLogs`	Cron used to send system logs.	Activate it and specify the schedule. The parameter addresses logs recipients addresses. They shall be coma-separated.
`[crons]/systemCheck`	Cron used to check all system coherence, including database optimization	Activate it and specify the schedule.

Example:

```
<crons>
      <cron:archiveSynchronization active="false" schedule="0 0 0/5 ? * *"/>
      <cron:cleanDatabase active="false" schedule="0 0 1 ? * *">
        <cron:tempUsers keepPeriod="10" />
        <cron:statistics keepPeriod="90" />
      </cron:cleanDatabase>
      <cron:dumpDatabase active="true" schedule="0 0 3 ? * *" />
      <cron:cleanDatabaseDump active="true" schedule="0 0 4 ? * *" keep="10" />
      <cron:fileScanners active="true" schedule="0 0/2 * * * ?" sourceRemove="true"/>
      <cron:searches active="false" schedule="0 0 5 ? * *" />
      <cron:sendLogs active="false" schedule="0 55 23 ? * *" addresses="ivv@ivv.com" />
      <cron:systemCheck active="false" schedule="0 30 23 ? * *"/>
```

```
   </crons>
```

8.1.2 messaging

[Groups]/ Parameter	Description	Usage
[messaging]/mail	Mail configuration. This field defines the rules about mail notification to users. The set values are used only at first launch of the system. They shall be modified in Management Panel if needed. Every notification type needs to be activated. In case of activation of one or more notification ,the following information shall be configured: • sender mail server specification • information about mail sender • "reply to" address of sent mails	Mail Services: • onUserCreate: true|false. It defines if system send mail when creating user • onUserUpdate: true|false. It defines if system send mail when updating user • onUserDelete: true|false. It defines if system send mail when deleting user Server information: • smtp: server address • port: server port • tls: defines if server is using TLS protocol • username and password: connection information • name: displayed name of "from" part of the notification email • address: displayed address of "from" part of the notification email • reply to: defines the "reply to" address of sent mails

Example:

```
<messaging>
   <msg:mail onUserCreate="true" onUserUpdate="true" onUserDelete="true">
      <msg:server smtp="131.176.235.251" port="25" tls="false"
username="scihub@copernicus.esa.int" password="$ecreT123">
         <msg:from name="IVV DHuS Team" address="ivv@copernicus.esa.int" />
         <msg:replyTo>ivv@copernicus.esa.int</msg:replyTo>
      </msg:server>
   </msg:mail>
</messaging>
```

8.1.3 network

[Groups]/ Parameter	Description	Usage

`[network]/outbound`	Outbound bandwidth configuration. These parameters are recharged every time the dhus is restarted. Parameters to be configured: channel name and its weight, including the user email pattern identifying the oubound channel, default user quota including maxconcurrent, max count, max size, max cumulative size and max bandwidth.	• `maxConcurrent` defines the maximum simultaneous accepted transfers. • `maxCount` defines the maximum number of accepted transfers on a customizable period. "periodUnit" attribute defines the unit of the period. Possible units are "DAYS", "HOURS", "MICROSECONDS", "MILLISECONDS", "MINUTES", "NANOSECONDS", "SECONDS"." period" attribute is a sliding time window used to count number of transfered product from now to the past period delay. • `maxSize` defines the maximum accepted size of transfer file. • `maxCumulativeSize` defines the maximum cumulated accepted transfers size on a customizable period. • `maxBandwidth` defines the maximum bandwidth authorized for the referred channel.
`[network]/inbound`	Inbound bandwidth configuration	see above

Example (GDPR disabled):

```
<network>
   <network:outbound>
      <network:channel name="PriorityChannel" weight="75">
         <network:classifier>
            <network:includes>
               <network:include>
              <network:userEmailPattern>.*@serco.com</network:userEmailPattern>
               </network:include>
            </network:includes>
         </network:classifier>
         <network:defaultUserQuotas>
              <network:maxConcurrent>20</network:maxConcurrent>
         </network:defaultUserQuotas>
      </network:channel>
      <network:channel name="SelfRegisteredChannel" weight="25">
         <network:defaultUserQuotas>
            <network:maxConcurrent>2</network:maxConcurrent>
         </network:defaultUserQuotas>
      </network:channel>
   </network:outbound>
   <network:inbound>
      <network:channel name="PriorityChannel" weight="75">
         <network:classifier>
            <network:includes>
               <network:include>
                  <network:userEmailPattern>.*@serco.com</network:userEmailPattern>
```

```
               </network:include>
            </network:includes>
         </network:classifier>
      </network:channel>
      <network:channel name="SelfRegisteredChannel" weight="25">
         <network:defaultUserQuotas>
         </network:defaultUserQuotas>
      </network:channel>
   </network:inbound>
</network>
```

Example (GDPR activate):

```
<conf:network>
    <network:outbound>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_1</network:rolePattern>
                    </network:include>
                </network:includes>
            </network:classifier>
            <network:defaultUserQuotas>
                <network:maxConcurrent>1</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_3</network:rolePattern>
                    </network:include>
                </network:includes>
            </network:classifier>
            <network:defaultUserQuotas>
                <network:maxConcurrent>3</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_UNLIMITED</network:rolePattern>
                    </network:include>
                </network:includes>
            </network:classifier>
        </network:channel>
        <network:channel name="SelfRegisteredChannel" weight="25">
            <network:defaultUserQuotas>
                <network:maxConcurrent>2</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
    </network:outbound>
    <network:inbound>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_1</network:rolePattern>
                    </network:include>
```

```
                </network:includes>
            </network:classifier>
            <network:defaultUserQuotas>
                <network:maxConcurrent>1</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_3</network:rolePattern>
                    </network:include>
                </network:includes>
            </network:classifier>
            <network:defaultUserQuotas>
                <network:maxConcurrent>3</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
        <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
                <network:includes>
                    <network:include>
                        <network:rolePattern>QUOTA_UNLIMITED</network:rolePattern>
                    </network:include>
                </network:includes>
            </network:classifier>
        </network:channel>
        <network:channel name="SelfRegisteredChannel" weight="25">
            <network:defaultUserQuotas>
                <network:maxConcurrent>2</network:maxConcurrent>
            </network:defaultUserQuotas>
        </network:channel>
    </network:inbound>
</conf:network>
```

8.1.4 products

[Groups]/ Parameter	Description	Usage
`[products]/download`	Download configuration. `CompressionLevel` parameter defines how rigorously the compressor looks the longest string possible. As a general rule of thumb: Compressing at the maximum level 9 requires around twice as much processor time as compressing at the minimum level 1. For typical input, compressing at the maximum as opposed to the minimum level adds around 5% to the compression	• `compressionLevel`: is a value to build the file compression, from 1 to 9. • `checksumAlgorithms`: algorithm to use for checksum computations (e.g. "MD5 ").

	ratio. 0 value means no compression. `checksumAlgorithms` shall be comma separated. **Warning**: Starting from 0.14.2 version, `publicData`=parameter is no more present.	
`[products]/quicklook`	Quicklook calculation parameters	• `height`: height of generated quicklooks • `width`: width of generated quicklooks • `cutting`: true|false. It allows system to cut image when processing quicklooks Specify dimensions in pixel unit.
`[products]/thumbnail`	Thumbnail calculation parameters	• `height`: height of generated thumbnails • `width`: width of generated thumbnails • `cutting`: true|false. It allows system to cut image when processing thumbnails Specify dimensions in pixel unit.

Example:

```
<products>
   <product:download compressionLevel="4" checksumAlgorithms="MD5"/>
   <product:quicklook height="512" width="512" cutting="false" />
   <product:thumbnail height="64" width="64" cutting="false" />
</products>
```

8.1.5 search

[Groups]/ Parameter	Description	Usage
`[search]/geocoder`	Geocoder (nominatim) configuration.	Set up the Nominatim geocoder and the Geoname configurations: • `boundingBox`: defines if the geocoder is querying only the bounding box of the matching place from the Nominatim Web Service i.e. the four corners encompassing the place. Otherwise, it will query the complete polygon boundaries, that may have lower performance according the the number of vertices composing the place's boundaries.

		• `maxPointNumber`: maximum number of points that can be returned for a polygon • `username`: username used to connect to Geoname
`[search]/odata`	OData configuration.	• `defaultTop`: maximum rows returned by OData and OpenSearch services
`[search]/solr`	Solr configuration	• `path`: solr path • `core`: solr core name • `schemaPath`: solr schema path. Shall be empty. • `synonymPath`: path of solr synonyms file

Example:

```
<search>
   <search:geocoder url="http://nominatim.openstreetmap.org">
      <search:nominatim boundingBox="false" maxPointNumber="50" />
      <search:geoname username="dhus" />
   </search:geocoder>
   <search:odata defaultTop="50">
   </search:odata>
   <search:solr path="&varFolder;/solr" core="dhus" schemaPath="" synonymPath="" />
</search>
```

Note: Please note that to disable Nominatim service the entire block has to be removed from dhus.xml file:

```
<search:geocoder url="http://nominatim.openstreetmap.org">
      <search:nominatim boundingBox="false" maxPointNumber="50" />
      <search:geoname username="dhus" />
</search:geocoder>
```

8.1.6 server

[Groups]/ Parameter	Description	Usage
`[server]/external`	External url viewed by users. Used in case of an apache proxy redirection for example. Empty values mean that server values are those which are viewed by users.	`protocol://host:port/path`
`[server]/ftp`	**Warning**: Starting from 0.14.4 version, `ftp` parameter is no more present.	

Example:

```
<server>
   <server:external host="dhs-test.onda-dias.eu" path="/ivv-dhus-fe-03" port="443"
protocol="https"/>
</server>
```

8.1.7 system

[Groups]/ Parameter	Description	Usage
`[system]/gdpr`	GDPR activation	• `active`: Boolean parameter to activate or not the GDPR. • `SAMLUserID`: defines the filed containing the IDP User ID in SAML metadata.
`[system]/administrator`	Definition of principal administrator user. If User exists, DHuS will give him all rights at launch, but will not his password. This shall be done in Management panel of GUI. If User is not existing, DHuS will create it with defined password.	Specify username and password
`[system]/archive`	Definition of local archive path for the error folder where products got in error during ingestion/synchronization will be stored. For products that do not fall into error please refere to 8.1.8.	• `errorPath`: path where to move products not ingested
`[system]/database`	Definition of database path, where dumps are stored and, optionally, the encryption type and encryption key. **Warning:** please note that, from the 0.10.3-4 version, the default behaviour is no encryption.	• `JDBCDriver`: component enabling DHuS to interact with the database • `dumpPath`: defines the location where store database (valid only for HSQL embedded database) • `JDBCUrl`: defines the database to be used and its properties • `Login`: user to access database • `Password`: credential to access database

`[system]/name`	Definition of system long name and short name.	• `long`: system long name • `short`: system short name
`[system]/processing`	Processing configuration. Configure the corePoolSize as an integer number (>0).	• `corePoolSize`: defines maximal number of active threads for the ingestion process. Default is 1.
`[system]/support`	Definition of support name and mail. These values are used only at first launch of the system. They shall be modified in Management Panel if needed via UI.	• `registrationMail`: used to send the administrative registration information. If this field is not set, DHuS is using support mail.
`[system]/tomcat`	Definition of tomcat path	• `path`: defines the location where store tomcat folder
`[system]/executor`	The background service that execute synchronizers. It must be enabled if you want to use the synchronisation feature In the batchmode the executor will run the synchronizers until there is no more to synchronize.	• `enabled`: true|false • `batchModeEnabled`: true|false
`[system]/trashPath`	Definition of trashpath path. The trashpath, if configured, is the folder used by DHuS to move there evicted products. It can be set empty ("") if there is no need to keep product on filesystem.	• `path`: defines the location where store evicted products.

Example:

```
<system>
   <system:gdpr active="true" SAMLUserId="samlId"/>
   <system:administrator name="root" password="password" />
   <system:archive path="&varFolder;/local_archive">
      <system:incoming maxFileNo="10" maxItems="10" path="/PUP2/dhus-test/var_IVV/incoming"
errorPath="/COMMON/IVV/errorpath"/>
   </system:archive>
   <system:database JDBCDriver="org.hsqldb.jdbcDriver"
                    hibernateDialect="org.hibernate.dialect.HSQLDialect"
```

```
                    dumpPath="&varFolder;/dump"
JDBCUrl="jdbc:hsqldb:file:&varFolder;/database/dhus;crypt_type=AES;crypt_key=5f0ebc384f232c
9108469a003724a5e2"
                    login="sa"
                    password="" />
   <system:name long="IVV Data Hub Service" short="IVV DHuS" />
   <system:processing corePoolSize="12" />
   <system:support name="DHuS Support Team" mail="ivv@copernicus.esa.int"
registrationMail="ivv@copernicus.esa.int" />
   <system:tomcat path="&varFolder;/tomcat" />
   <system:executor enabled="true" batchModeEnabled="false" />
   <system:trashPath path="" />
</system>
```

8.1.8 DataStores

It is possible to configure multiple DataStores in DHuS.The following kinds of DataStores are currently available: Hierarchically File System (HFS), Openstack.

[Groups]/ Parameter	Description	Usage
`[dataStores]/dataStore`	This section is referred to the DataStore configuration. All DataStores share some common attributes: • name • restriction • priority It is necessary to configure a DataStore for each of the archive where the products are located.	• `type`: define the DataStore implementation to use ("hfsDataStore", "openStackDataStore"). • `name`: unique DataStore name. • `restriction`: its allowed value are "none", "referencesOnly" and "readOnly". If it is set to "readOnly" the adding, moving and deleting actions are not supported for the DataStore. Default is read-write (i.e. restriction: none). • `priority`: priority with which DataStores are accessed within the system. DataStores with the smallest priority are accessed first. Default is 100. • `maximumSize`: threshold in byte of the DataStore size. It is the trigger for the automatic OnInsert eviction. • `currentSize`: size of the DataStore . Automatically calculated by DHuS. • `autoEviction`: true|false. It means if the On-Insert automatic eviction is to be triggered or not. • `path` (only for HSF DataStore): local storage path. • `maxFileNo` (only for HSF DataStore): maximum number of sub-folder for each stage. Please note that it is labelled `MaxFileDepth` in OData. • `maxItems` (only for HFS DataStore): maximum number of file in a folder. • `provider` (only for Openstack DataStore): provider service to use, only "*openstack-swift*" is currently available.

		• `identity` (only for Openstack DataStore): identifier for the authentication service. • `credential` (only for Openstack DataStore): *project:password* of an openstack account. • `url` (only for Openstack DataStore): URL of Openstack authentication service. • `container` (only for Openstack DataStore): container to use. Please note that the container shall be created manually before configuring this parameter. • `region` (only for Openstack DataStore): Each Region has its own full Openstack deployment, including its own API endpoints and networks . The region is linked to the openstack account. The default value is *regionOne.* o

Example of configuration for HFS DataStore:

```
<ds:dataStore xsi:type="ds:hfsDataStoreConf"
     name="DataStore1_name"
     restriction="none"
     priority="0"
     maximumSize="1000"
     currentSize="0"
     autoEviction="true">
<ds:path>/data/dhus-incoming</ds:path>
<ds:maxFileNo>10</ds:maxFileNo>
<ds:maxItems>1024</ds:maxItems>
</ds:dataStore>
```

Example of configuration for Openstack DataStore:

```
<ds:dataStore xsi:type="ds:openStackDataStoreConf"
     name="DataStore2_name"
     restriction="none">
<ds:provider>openstack-swift</ds:provider>
<ds:identity> tenantName:username</ds:identity>
<ds:credential>password</ds:credential>
<ds:url>https://keystone9915.openstack.ovh.net:35358/v2.0</ds:url>
<ds:container>container_name</ds:container>
<ds:region>RegionOne</ds:region>
</ds:dataStore>
```

8.1.9 scanners

[Groups]/ Parameter	Description	Usage
[scanners]/scanner	This section is referred to product upload via FileScanner. FileScanner configuration is automatically created when file scanners are created via UI or OData. It can be update or modified in the dhus.xml when DHuS is not running.	• id: defines the unique scanner identifier • url: it is the path of the source folder where products are retrieved • pattern: it is the regular expression w.r.t products are scanned in the Inbox folder • collections: it means if products are ingested in collections • sourceRemove: it means if successfully ingested products are removed from the inbox folder • active: it is the trigger for the ingestion via file scanner • schedule: schedule for the FileScanner runs

Example:

```
<conf:scanners>
	<scanner:scanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:type="scanner:fileScannerConf">
		<scanner:id>1</scanner:id>
		<scanner:url>file:///path/to/inbox</scanner:url>
		<scanner:pattern>.*\.zip</scanner:pattern>
		<scanner:collections/>
		<scanner:sourceRemove>true</scanner:sourceRemove>
		<scanner:cron active="false" schedule="0 0/5 0 ? *"/>
	</scanner:scanner>
</conf:scanners>
```

8.1.10 synchronizers

[Groups]/ Parameter	Description	Usage
[synchronizers]/ synchronizer	This section is referred to the synchronizer's configuration. It is automatically created when synchronizers are created. It can be updated or modified in the dhus.xml when DHuS is not running.	• Id: defines the unique synchronizer identifier • label: it is synchronizer name • schedule: schedule of the synchronizer runs • active: trigger of the synchronization process • created: creation date of the synchronizer • modified: last update of the synchronizer

		• `pagesize`: number of products synchronized at each run • `filterParam`: it filters the products and synchronizes only the ones respecting the filter, following the OData filters syntax (see RD 5) • `targetCollection`: name of collection; if specified, the synchronized products will be filled in the referenced target Collection • `geofilterOp`: operator used to match the GeoFilter (reported in geofilterShape parameter) and the product's footprint. It can be set as: - within (a product's footprint is inside the given shape) - contains (a product's footprint is contains the given shape) - disjoint (a product's footprint and the shape don't overlap) - intersects (a product's footprint and the shape overlap) • `geofilterShape`: shape to compare with product's footprint; is a topologically valid shape written in WKT (see: https://en.wikipedia.org/wiki/Well-known_text) • `copyProduct`: means if only product metadata are synchronized or the entire .zip file • `skipOnError`: it means if a product is skipped or not when a synchronization error occurs • `syncOfflineProducts`: allows or not the synchronization of offline products • `retriesForSkippedProducts`: number of synchronizations retries in case of skipped products • `timeoutSkippedProducts`: timeout of each request of synchronization of skipped products • `rankingSchedule`: schedule of the Intelligent Synchronizer source ranking • `retryingSourceDelay`: delay after which a source will be retried even if other sources have better performances • `sources`: sources among which the synchronizer can download data. The parameters for the reported sources are:

		- `referencedId`= it represents the Id of the referenced source - `lastCreationDate`= last successfully synchronized product creation date. It represents the date in the DataSource from which new products will be synchronized. - `lastDateSourceUsed`= the last date the source was used. It is used to understand when reusing a source because it was not used in the last period, as configured in the "RetryingSourceDelay" parameter - `sourceCollection`= with such parameter only the products on the source which are belonging to the configured collection will be synchronized

Example:

```
<conf:synchronizers>
        <sync:synchronizer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="sync:productSynchronizer" retriesForSkippedProducts="1"
timeoutSkippedProducts="60000" retryingSourceDelay="20">
            <sync:id>0</sync:id>
            <sync:label>S2</sync:label>
            <sync:schedule>0 */2 * * * ?</sync:schedule>
            <sync:active>false</sync:active>
            <sync:created>2022-08-19T07:42:43.126Z</sync:created>
            <sync:modified>2022-08-19T07:42:43.181Z</sync:modified>
            <sync:pageSize>2</sync:pageSize>
            <sync:targetCollection>S2A</sync:targetCollection>
            <sync:copyProduct>true</sync:copyProduct>
            <sync:filterParam>startswith(Name,'S2A')</sync:filterParam>
            <sync:geofilterOp>intersects</sync:geofilterOp>
            <sync:geofilterShape>POLYGON((-127.47284999572936 48.86934726209239,-
126.33062006335341 29.624451625469064,-68.99068094388953 27.821547075229674,-
69.2191234445556 48.56791527890829,-127.47284999572936 48.86934726209239,-
127.47284999572936 48.86934726209239))</sync:geofilterShape>
            <sync:skipOnError>true</sync:skipOnError>
            <sync:syncOfflineProducts>false</sync:syncOfflineProducts>
            <sync:rankingSchedule>30 */5 * * * ?</sync:rankingSchedule>
            <sync:sources>
                <sync:source>
                    <sync:referenceId>0</sync:referenceId>
                    <sync:lastCreationDate>2022-06-27T00:00:00.001Z</sync:lastCreationDate>
                </sync:source>
                <sync:source>
                    <sync:referenceId>1</sync:referenceId>
                    <sync:sourceCollection>Collection_Name</sync:sourceCollection>
                    <sync:lastCreationDate>2022-06-27T00:00:00.001Z</sync:lastCreationDate>
                </sync:source>
            </sync:sources>
        </sync:synchronizer>
</conf:synchronizers>
```

8.1.11 product sources

[Groups]/ Parameter	Description	Usage
[productSources]/ productSource	This section is referred to the product sources configuration. It is automatically created when product sources are created via OData. It can be updated or modified in the dhus.xml when DHuS is not running.	• Id= identifier of the Source. Read-only and automatically generated at creation • Url= https://[DHuS_address]/odata/v1. Adress of the DHuS instance used as DataSource. • Login= User name of a user registered in the DataSource. In case of metadata synchronization, the archive manager rights enabled are requested for this user. In case of synchronization with remote copy, a normal user (meaning with search and download rights) can be used. • Password= password used to login the user to the Source • RemoteIncoming= path of the incoming folder configured for the DHuS installed as the source instance. The role of the remote incoming is linked to the kind of synchronizer: metadata synchronizer ("Remote Incoming" field shall be set) or product synchronizer (empty "Remote Incoming" field). • Listable= defines if the Source will be used to retrieve the list of products to be synchronized. If set as false, this source will be only used for downloading and so to synchronize products, but not used to list them. Its default value is false. Please note that if a synchronizer with only one source is configured, and the source is set with listable=false, the synchronizer will not be able to synchronize any products

Example:

```
<conf:productSources>
     <ps:productSource>
         <ps:id>0</ps:id>
         <ps:url>https://colhub.copernicus.eu/dhus/odata/v1/Products</ps:url>
         <ps:login>***</ps:login>
         <ps:password>***</ps:password>
```

```
            <ps:listable>true</ps:listable>
        </ps:productSource>
        <ps:productSource>
            <ps:id>1</ps:id>
            <ps:url>https://colhub2.copernicus.eu/dhus/odata/v1/Products</ps:url>
            <ps:login>***</ps:login>
            <ps:password>***</ps:password>
            <ps:remoteIncoming>/my/directory/for/products</ps:remoteIncoming>
            <ps:listable>true</ps:listable>
        </ps:productSource>
        <ps:productSource>
            <ps:id>2</ps:id>
            <ps:url>https://colhub3.copernicus.eu/dhus/odata/v1/Products</ps:url>
            <ps:login>***</ps:login>
            <ps:password>***</ps:password>
            <ps:listable>false</ps:listable>
        </ps:productSource>
    </conf:productSources>
```

8.1.12 evictions

[Groups]/ Parameter	Description	Usage
`[evictions]/eviction`	This section is referred to the configuration of the Custumizable Eviction. It is automatically created when an eviction is created via UI or via OData. It can be update or modified in the dhus.xml when DHuS is not running.	• `name`: unique eviction name • `keepPeriod`: it is the trigger of the eviction meaning that it is the minimal time after which products can be evicted • `keepPeriodUnit`: time unit corresponding to the KeepPeriod property. Default value is DAYS. • `maxEvictedProducts`: maximum number of products that can be evicted during a single run of the eviction • `filter`: OData filter used to determine which products can be evicted • `orderBy`: OData ordering clause used to determine in which order products will be evicted by the eviction • `softEviction`: true|false. It defines if the eviction is acting only on zip file or also on products metadata • `status`: it's a read only property and shows if the eviction is running or not • safeMode: true|false. It allows to trigger the Eviction run in SAFE mode or no. • `targetDataStore`: DataStore from where products will be evicted • `baseDate`: datetime metadatum with respect to the Eviction run will be performed • `cron`: it defines the schedule of each run of the eviction

Example:

```
<evict: eviction name="Soft_param" keepPeriod="1" keepPeriodUnit="DAYS"
maxEvictedProducts="2" filter="startswith(Name,'S2')" orderBy="ModificationDate asc"
softEviction="true" status="STOPPED" safeMode="false"
targetDataStore="S2_ParamPDGSDataStore" baseDate="modificationDate">
     <evict:cron active="true" schedule="0 0/1 * * * ?"/>
</evict:eviction>
```

8.2 server.xml

The following table summarize the configurable parameters of the `server.xml` file.

[Groups]/ Parameter	Description	Synthax
`[Service]/Connector`	It is possible to configure several connectors in DHuS, two of them are configured by default. Such connectors define ports on which the application is in listening mode. The first default connector is the one for port 8081, which is the port towards which external requests are directed. The port 30333, configured in the second DHuS connector, is used for internal requests (when OpenSearch requests are sent to DHuS, the application redirect such requests internally towards 30333).	• `port="8081"` • `protocol="org.apache.coyote.http11.Http11NioProtocol"` • `maxConnections="1000"` • `maxThreads="400"` • `keepAliveTimeout="2000"` • `URIEncoding="ISO-8859-1"` • `compression="on"` • `compressionMinSize="1024"` • `compressableMimeType="application/json,application/javascript,application/xhtml+xml,application/xml,text/html,text/xml,text/plain,text/javascript,text/css"` • `port="30333"` • `protocol="org.apache.coyote.http11.Http11NioProtocol"` • `maxConnections="1000"` • `maxThreads="400"` • `keepAliveTimeout="2000"` • `URIEncoding="ISO-8859-1"` • `relaxedQueryChars="^{}[]|""` • `compression="on"` • `compressionMinSize="1024”` • `compressableMimeType="application/json,application/javascript,application/xhtml+xml,application/xml,text/html,text/xml,text/plain,text/javascript,text/css"`
`[Service]/Engine/Valve`	DHuS keep track of the requests it manages in a Rolling cache.	Access Filter Settings are:

	From this section of the server.xml is possible to define which kind of request/connections should be recorded in the cache. The default configuration (reported in the usage here on the right) is such that DHuS record in its rolling cache every request it manages (OData, OpenSearch and UI request). Cache is reset at dhus stop.	• **pattern:** the regular expression to filter user request, i.e. `"^.*(/odata/v1/).*$"` only manages odata request and `"^((?!/(home\|new)/).)*$"` consider all request but the UI. • `useLogger="true|false"` show or hide the user access in logger output. This setting does impact keeping internal track of the request. • `enable="true|false"` activate/deactivate the valve.

Example:

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
   <Service name="DHuS-Service">
      <Connector port="8081"
         protocol="org.apache.coyote.http11.Http11NioProtocol"
         maxConnections="1000"
         maxThreads="400"
         keepAliveTimeout="2000"
         URIEncoding="ISO-8859-1"
         relaxedQueryChars="^{}[]|&amp;quot;"
         compression="on"
         compressionMinSize="1024"

compressableMimeType="application/json,application/javascript,application/xhtml+xml,applica
tion/xml,text/html,text/xml,text/plain,text/javascript,text/css" />
      <Connector port="30333"
         protocol="org.apache.coyote.http11.Http11NioProtocol"
         maxConnections="1000"
         maxThreads="400"
         keepAliveTimeout="2000"
         URIEncoding="ISO-8859-1"
         compression="on"
         compressionMinSize="1024"

compressableMimeType="application/json,application/javascript,application/xhtml+xml,applica
tion/xml,text/html,text/xml,text/plain,text/javascript,text/css" />
      <Engine name="DHuS-Engine" defaultHost="localhost">
         <Host name="localhost" appBase="webapps" deployOnStartup="false"
autoDeploy="false">

            <Valve className="org.apache.catalina.valves.AccessLogValve"
               prefix="access_log-"
               suffix=".txt"
               directory="logs"
               pattern="%h %l %u %t %r %s %b %I %D" />

            <Valve className="fr.gael.dhus.server.http.valve.AccessValve"
               pattern=".*"
               useLogger="true"
               enable="true" />
```

```
                        <!-- Valve used to display the potential exception thrown by
processing quota valve in HTML -->
                        <Valve className="org.apache.catalina.valves.ErrorReportValve"/>

            <Valve className="fr.gael.dhus.server.http.valve.processings.ProcessingValve"
               enable="false"
               userSelection="LOGIN,IP"
               userWhiteList=""
               timeWindow="900"
               idleTimeBeforeReset="900"
               maxElapsedTimePerUserPerWindow="300"
               maxRequestNumberPerUserPerWindow="900"
               maxUsedMemoryPerUserPerWindow="53687091200" />
               <Valve className="org.apache.catalina.valves.RemoteIpValve"
                  remoteIpHeader="x-forwarded-for"
                  proxiesHeader="x-forwarded-by"
                  protocolHeader="x-forwarded-proto" />
         </Host>
      </Engine>
   </Service>
</Server>
```

8.3 start.sh

DHuS command line follows Java VM standard command lines. Please find below DHuS internal properties that can be configured.

Parameter	Description & Usage	Optional/ Mandatory

`XX:MaxPermSize`
`Xms`
`Xmx`
`Xss`
	The maximum permanent generation size can be configured with the command-line option `-XX:MaxPermSize`.
	The flag `Xmx` specifies the maximum memory allocation pool for a Java virtual machine (JVM), while `Xms` specifies the initial memory allocation pool. It is recommended to set these parameters equally to avoid overload for the machine. The configuration of those parameters is depending on the users activity.
	The flag `Xss` allows to specify the size of the frame stack used by each thread.
	Example:
	`-XX:MaxPermSize=1g -Xms24g -Xmx24g -Xss10m`
		M

`java.library.path`	Path to the native libraries of the DHuS distribution.	M
`com.sun.media.jai.disableMediaLib`	=true to be removed if media jai native library is provided. DHuS does not requires these libraries for optimization.	M

`user.timezone`	=UTC Mandatory parameter to force the DHuS timezone to a standard, not depending on the operating system settings.	M
`sun.zip.disableMemoryMapping`	=true Currently mandatory to avoid a crash in zip library usage.	M
`java.util.logging.manager`	It sets up the java.util.logging to Log4J bridge. `-Djava.util.logging.manager=org.apache.logging.log4j.jul.LogManager \`	M

`http.proxyHost`
`http.proxyPort`
`http.nonProxyHosts`
	Property `http.proxyHost` defines the host name of the proxy server.
	Property `http.proxyPort` defines the port number, the default value being 80.
	Property `http.nonProxyHosts` defines a list of hosts that should be reached directly, bypassing the proxy. This is a list of patterns separated by '|'. The patterns may start or end with a '*' for wildcards. Any host matching one of these patterns will be reached through a direct connection instead of through a proxy.
	Example:
	`-Dhttp.proxyHost=131.176.235.248`
	`-Dhttp.proxyPort=3128`
	`-Dhttp.nonProxyHosts="172.30.246.*"\`
		O

`java.rmi.server.hostname`	Insert this line in the Java options in order to allow clients to invoke methods on the remote object (needed for the cache replication in Scalability 2.0 mode): `-Djava.rmi.server.hostname=<DHuS_VM_IP>` Where `DHuS_VM_IP` is the IP address of the VM hosting the DHuS service.	O
`java.rmi.activation.port`	Insert this line in the Java options in order to specify the port where peers access each other's cache (needed for the cache replication in Scalability 2.0 mode): `-Djava.rmi.activation.port=40001`	O

sun.rmi.transport.connectionTimeout
sun.rmi.transport.tcp.handshakeTimeout
sun.rmi.transport.tcp.responseTimeout
sun.rmi.transport.tcp.readTimeout
	Insert these lines in the Java options in order to set Java RMI runtime time-out times (needed for the cache replication in Scalability 2.0 mode):
	`-Dsun.rmi.transport.connectionTimeout=15000 \`
	`-Dsun.rmi.transport.tcp.handshakeTimeout=15000 \`
	`-Dsun.rmi.transport.tcp.responseTimeout=15000 \`
	`-Dsun.rmi.transport.tcp.readTimeout=15000 \`
		O

country.synonyms	/path/to/file path to the definition of users countries synonyms	O
webapp.excluded	The name of the webapp to not start at system startup (i.e. "fr.gael.dhus.gwt.GWTWebapp")	O
Archive.check	=true|false (default=false) Force system check at dhus startup	O
Archive.forceReindex	true|false (default=false) Force all the products indexes being re-indexed.	O
Archive.incoming.relocate	true|false (default=false) Force the relocation of all the products of incoming	O
Archive.incoming.relocate.path	/path/to/relocation (default="") Give the new location path to relocate incoming directory. If no pas is provided, incoming will be relocated in its current directory	O
Archive.processings.clean	true|false (default=false) Clean all the interrupted processing instead of recover them.	O
force.public	true|false (default=false) Force all the product contained into DHuS become public. This is an old function and this parameter is deprecated. Please do not use it.	O
Archive.synchronizeLocal	true|false (default=false) Force re-synchronization of local archive path at system startup. This is an old function and this parameter is deprecated. Please do not use it.	O

`users.search.notification.force.inactive`	true|false (default=false) Deactivates all the user search notifications.	O
`checkUserConfiguration`	true|false (default=false) Activates schema aware validation of input xml configuration file	O
`dhus.solr.reindex`	true|false (default=false) Recreate the Solr index from the database	O
`action.record.inactive`	true|false (default=false) Full deactivates read/write statistics	O
`dhus.sync.download_attempts`	(default=10) During product synchronization. if a download is interrupted and the remote DHuS supports HTTP ranges, the parallelised download manager will attempt to resume the download. number of download attempts (-1 for infinite, must be at least 1)	O
`dhus.search.innerTimeout`	Inner timeout for SolR requests	O

`http.timeout.socket`
`http.timeout.connection`
`http.timeout.connection_request`
	Timeout for the OData synchronizers
	http.timeout.socket is the socket timeout, it occurs when no data has been received within the given time span, it is also known as SO_TIMEOUT.
	http.timeout.connection is the connection timeout, it occurs when the remote server does not answers our connection requests.
	http.timeout.connection_request is a timeout on a request to a connection manager, this is a purely programmatic timeout, and you should probably not set this timeout.
	The socket timeout is longer because it can occur in the middle of a transfer, we don't want to throw all the data that has already been downloaded to the rubbish if the remote server is not responsive only a few minutes.
		O

`max.product.page.size`	It limits the number of products that can be displayed at once via OData Products entity set and OpenSearch service. Please note that processing quotas shall be deactivated to let this parameter be considered by DHuS.	O

	Default: • OData have no limitation • OpenSearch has the default limitation to 100 Setting it to "-1", OData and OpenSearch will have no limitation.	
`fr.gael.streams.cache.inactive=true`	This parameter is needed to disable the cache mechanism (see Section 7.18)	O
`fr.gael.streams.tmpdir="/my/temp/file/location"`	location of the temporary cache folder (see Section 7.18)	O
`com.sun.management.jmxremote`	=true|false Set the following system property to allow the JMX client access to the DHuS Java VM. This is needed for SYSMA monitoring feature.	O
`com.sun.management.jmxremote.port`	It defines the port number through which you want to enable JMX RMI connections. This is needed for SYSMA monitoring feature.	O
`com.sun.management.jmxremote.ssl`	=true|false. It allows to enable/disable SSL (Secure Sockets Layer) when monitoring remotely. This is needed for SYSMA monitoring feature.	O
`com.sun.management.jmxremote.authenticate`	=true|false. It allows to enable/disable the password authentication for remote monitoring. Default values is true. This is needed for SYSMA monitoring feature.	O
`com.sun.management.jmxremote.local.only`	=true|false. It allows to define if to accept or not external connections. Default value is true to avoid any incoming connection except from localhost (127.0.0.1). This is needed for SYSMA monitoring feature.	O
`jclouds.mpu.parts.size`	It allows controlling the upload strategy that will split an object up in to individual parts and upload them in parallel. The size (in bytes) of parts being uploaded in parallel by default is 536870912 bytes = 512 MB.	O
`javax.net.ssl.trustStore`	Insert these line in the Java options in order to validate the certificates necessary during LTA and On-Demand processing: `-Djavax.net.ssl.trustStore=/usr/java/jdk1.8.0_191-amd64/jre/lib/security/cacerts`	O

dhus.saml.idp.name	In case of GDPR activate, this parameter represents the name of the Realm created on Keycloak.	O
dhus.saml.idp.url	In case of GDPR activate, this URL is pointing to the XML metadata file of Keycloak: https://<KEYCLOAK_ADDRESS>/auth/realms/dhus/protocol/saml/descriptor	O
dhus.saml.sp.id	In case of GDPR activate, this parameter is the name of the client representing the DhuS instance created on Keycloak.	O
dhus.saml.keystore.file	In case of GDPR activate, this parameters defines the entire path of the generated dhusKeystore.jks file.	O
dhus.saml.keystore.storePass	In case of GDPR activate, this parameter is the password used to generate the .jks file.	O
dhus.saml.keystore.defaultKey	In case of GDPR activate, this represents the key used in the command to generate the .jks file.	O
dhus.saml.keystore.defaultPassword	In case of GDPR activate, this represents the password used in the command to generate the .jks file.	O
dhus.sync.skipped_turns	This parameter allows configuring the number of retrieved pages between two retries of skipped products during synchronization run.	O
dhus.metrics.filter	This parameter allows configuring the metrics to be reported. Using "ingestion" only metrics concerning the ingestion can be seen. Using "prod_sync" only metrics concerning the synchronization can be seen. Please note that if any value is configurated, all the metrics present by default can be seen.	O

8.4 Monitoring.xml

This section explains the parameters contained into monitoring.xml configuration file.

Here can be configured the purge of metrics from `OData/v2/Metrics`, how often metrics are collected and how to collect metrics into log file.

In the table below are listed the main sections to configured to enable monitoring feature using metrics with embedded database, as per *monitoring.xml-dhus* file provided in the DHuS distribution. The complete configuration for *monitoring.xml-dhus* file reported in Appendix B.

The configuration of *monitoring.xml-influx* file, needed to use metrics with externalized database, is reported in Appendix C.

Configuration example	Description & Usage	Optional/ Mandatory
`<metrics:reporter type="dhus-reporter" metric-registry="DHuSMetrics" period="1m" />`	Embbeded metric storage, rate is every 1 minute as reported in parameter "period". Do not change metric-registry value	M

`<bean id="metrics_storage" class="org.springframework.jdbc.datasource.DriverManagerDataSource">`
`<property name="url" value="jdbc:hsqldb:mem:." />`
`<!-- HSQL DB in memory -->`
`</bean>`
	Configure a storage of metrics in a HSQL database loaded in-memory. This database is different from DHuS one.
	Caution: This database is deleted when DHuS is stopped.
		M

`<bean id="metrics_storage" class="org.springframework.jdbc.datasource.DriverManagerDataSource">`
`<property name="url" value="jdbc:hsqldb:file:/my/filepath/databaseName;hsqldb.sqllog=3;hsqldb.applog=3" />`
`<!-- HSQL DB in file -->`
`</bean>`
	Metrics storage is done in the file path configured (/my/filepath) with the given name (databaseName). It will create a directory "databaseName" and some files (properties, script) which contains details about the database and some scripts executed when the database is created.
	Options;hsqldb.sqllog=3;hsqldb.applog=3 added are used to log events in database (file: databaseName.sql.log)and in application (file: databaseName.app.log). We can specify a different level of logs from minimal (1) to maximal (3). In particular:
	• 1 logs only commits and rollbacks
	• 2 logs all the SQL statements executed, together with their parameter values. Long statements and parameter values are truncated
	• 3 is similar to Level 2 but does not truncate long statements and values
	With this configuration it can be possible to access metrics information also when DHuS is stopped.
	Please note that this configuration (metric storage in file) and the one given above (with HSQL database loaded in-memory) are **mutually exclusive**, and so just one of them has to be chosen.
		M

`<bean id="service" class="org.dhus.metrics.embed.MetricsService">`	Configuration of the automatic purge of metrics into OData/v2/Metrics. In this example, metrics are kept 4320 minutes.	M

<constructor-arg ref="jdbi" /> <constructor-arg value="4320" /> <constructor-arg value="MINUTES" /> </bean>		
<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Item"> <property name="itemClassNamePrecision" value="FINE" /> </bean>	Monitor downloads per Item class	O
<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.ItemAndUser"> <property name="itemClassNamePrecision" value="FINE" /> </bean>	Monitor downloads per Item class and user (only on DHuS instances having a limited number of users).	O
<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.User" />	Monitor downloads per user (only on DHuS instances having a limited number of users).	O
<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Glob" />	Monitor downloads per Item class and user (only on DHuS instances having a limited number of users)	O

By commenting or removing from monitoring.xml file the lines of parameters marked as optional in the table above, those metrics will not be reported.

Warning: in general, since for this release only the metrics related to synchronizers have been tested, it is recommended to keep only synchronizers metrics and to comment all the others.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:metrics="http://www.ryantenney.com/schema/metrics"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.ryantenney.com/schema/metrics
http://www.ryantenney.com/schema/metrics/metrics.xsd
           http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

    <!-- Creates a MetricRegistry bean, do not change its ID -->
    <metrics:metric-registry id="DHuSMetrics" />

    <!-- Registers BeanPostProcessors with Spring which proxy beans and capture metrics -->
    <metrics:annotation-driven metric-registry="DHuSMetrics" expose-proxy="true" />

    <metrics:reporter type="dhus-reporter" metric-registry="DHuSMetrics" period="1m" />
    <!-- Configure embedded metrics storage -->
    <bean id="metrics_storage"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <!-- property name="url" value="jdbc:hsqldb:mem:." /-->
        <property name="url" value="jdbc:hsqldb:file:/data/dhus-
3.1.4/metricsdb/metricsdb;hsqldb.sqllog=3" />
    </bean>

    <bean id="jdbi" class="org.jdbi.v3.spring4.JdbiFactoryBean">
        <property name="dataSource" ref="metrics_storage" />
        <property name="autoInstallPlugins" value="true" />
    </bean>

    <!-- Define the service and the period for storing metrics information -->
    <bean id="service" class="org.dhus.metrics.embed.MetricsService">
        <constructor-arg ref="jdbi" />
        <constructor-arg value="4320" />
        <constructor-arg value="MINUTES" />
    </bean>
</beans>
```

8.5 log4j2.xml

Log4j2 file defines the log level of processes run by the Data Hub, in fact in each logger it is possible to raise/lower the level. The RollingFile section of the `log4j2.xml` defines the name of the application log, its location and its rolling policy.

Log4j2 is a Java-based logging utility now project of the Apache Software Foundation. Here in below follows an overview of the configuration parameters used in the default log4j2 file. Please refer to https://logging.apache.org/log4j/2.0/manual/ for configuration details.

[Groups]/ Parameter	Description	Usage
`[Properties]/Property`	Configuration and definition of the pattern used as template of the DHuS log. The pattern available in the default distribution of the log4j2.xml is such that the log has the following shape: `[DHuS_version][timestamp in UTC][level][log message]`	• `name=`name of the template to be used
`[Appenders]/Console`	Configuration of the messages logged for standard output (output of SW run) and standard error (SW errors), including `<Filters>` configuration that allows Log Events to be evaluated to determine if or how they should be published.	• `Name=` name of the console (stdout/stderr) • `Target=` location where ConsoleAppender writes its output. Default value is System.out. • `Patternlayout` o `pattern=` template used in the standard/stderr output log • `ThresholdFilter` o `level=` level of messages to be filtered. Anything at or below this level will be filtered out if `maxBurst` has been exceeded. The default is WARN meaning any messages that are higher than warn will be logged regardless of the size of a burst that is "10" by default o `onMatch=` Action to take when the filter matches. May be ACCEPT, DENY or NEUTRAL. The default value is NEUTRAL. o `onMismatch=` Action to take when the filter does not match. May be ACCEPT, DENY or NEUTRAL. The default value is DENY.
`[Appenders]/RollingFile`	Configuration of the rolling policy for DHuS log file, including policies to define logging criteria, i.e rules defining that all files matching a given file name pattern	• `Name=` name of the log file • `Filename=` location and name of the real time DHuS log

	(`filepattern`) and older than a certain number of days are rolled.	• `Filepattern=` location and name of the rolled log files • `Patternlayout` o `pattern=` template used in the standard/stderr output log • `TimeBasedTriggeringPolicy` o `interval=` It is an integer that defines how often a rollover should occur based on the most specific time unit in the date pattern. For example, with a date pattern with hours as the most specific item and and increment of 4 rollovers would occur every 4 hours. The default value is 1. o `modulate=` true|false. It indicates whether the interval should be adjusted to cause the next rollover to occur on the interval boundary.
`[Loggers]`	Loggers allow using specific log levels for classes. Beyond the loggers provided by the default in the `log4j2.xml`, the following loggers are available for Synchronizers, EvictionServices and DataStores, like in example reported below.	• `Name=` logger name • `Level=` the level may be configured with one of TRACE, FATAL, DEBUG, INFO, WARN, ERROR, ALL or OFF. If no level is specified it will default to ERROR. • `Root=` root logger. The root logger does not have the name attribute. The appenders attached are by default the stdout stderr and the RollingFile.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
        <Properties>
                <Property name="pattern"
>[$${sys:fr.gael.dhus.version}][%d{DEFAULT}{UTC}][%-5p] %m (%file:%line - %t)%n%throwable
</Property>
        </Properties>
        <Appenders>
                <Console name="stdout" target="SYSTEM_OUT">
                        <PatternLayout pattern="${pattern}" />
                        <Filters>
                            <ThresholdFilter level="DEBUG"/>
                            <ThresholdFilter level="WARN" onMatch="DENY"
onMismatch="NEUTRAL"/>
                        </Filters>
```

```
                </Console>
                <Console name="stderr" target="SYSTEM_ERR">
                        <PatternLayout pattern="${pattern}" />
                        <Filters>
                            <ThresholdFilter level="WARN"/>
                        </Filters>
                </Console>
                <RollingFile name="RollingFile" fileName="dhus.log"
                        filePattern="dhus-%d{yyyy-MM-dd-HH}.log">
                        <PatternLayout>
                                <Pattern>${pattern}</Pattern>
                        </PatternLayout>
                        <Policies>
                                <TimeBasedTriggeringPolicy interval="6" modulate="true" />
                        </Policies>
                        <Filters>
                            <ThresholdFilter level="DEBUG"/>
                        </Filters>
                </RollingFile>
        </Appenders>
        <Loggers>
                <logger name="fr.gael.dhus" level="info"/>
                <logger name="fr.gael.drb.query.FunctionCallExpression" level="debug"/>
                <logger name="org.apache.cxf.jaxrs.utils.JAXRSUtils" level="error"/>
                <logger name="org.apache.solr" level="error"/>
                <logger name="fr.gael.dhus.sync" level="debug"/>
                <logger name="fr.gael.dhus.service.EvictionService" level="debug"/>
                <logger name="org.dhus.store.DataStore.hfs" level="debug"/>
                <logger name="org.dhus.metrics.embed.MetricsService" level="debug"/>
                <logger name="org.dhus.store.ParallelProductSetter" level="debug"/>
                <logger name="org.hibernate.orm.deprecation" level="error"/><!-- Hibernate
deprecation warnings -->
                <Root level="info">
                        <AppenderRef ref="stderr" />
                        <AppenderRef ref="stdout" />
                        <AppenderRef ref="RollingFile" />
                </Root>
        </Loggers>
</Configuration>
```

In the table below, some useful loggers are provided. They can be added to the log4j2.xml configuration file in order to have more details on running processes.

Scope	Level	Logger
Visualize actions involving HFS DataStore	DEBUG	• <logger name="org.dhus.store.datastore.hfs.HfsDataStore" level="debug"/>
Visualize information about Cart content at every User access	DEBUG	• <logger name="fr.gael.dhus.server.http.webapp.stub.controller" level="debug"/> Please note that if this logger is not configured, the information will be logged at INFO level.
Visualize information about configured caches	DEBUG	• <logger name="fr.gael.dhus.util.LoggingCacheListenerFactory" level="debug"/>

		• <logger name="fr.gael.dhus.util.LoggingCacheListenerForQuotas" level="debug"/>
Visualize information about synchronization process	DEBUG	• <logger name="fr.gael.dhus.sync.impl.ODataProductSynchronizer" level="debug"/>
Visualize information during synchronization with copy	DEBUG	• <logger name="fr.gael.dhus.util.http.DownloadableProduct" level="debug"/>
Visualize information regarding synchronization with multiple sources	DEBUG	• <logger name="org.dhus.store.ParallelProductSetter" level="debug"/>
Visualize information regarding DHuS metrics	DEBUG	• <logger name="org.dhus.metrics.embed.MetricsService" level="debug"/>
Hide the following WARN message sent by HSQLDB when a Delete, Update, Merge or Delete Truncate operation is performed with no rows affected.: `[WARN ] SQL Warning Code: -1100, SQLState: 02000`	ERROR	• <logger name="org.hibernate.engine.jdbc.spi.SqlExceptionHelper" level="error" />
Hide the following WARN message placed by the Hibernate Dev Team to inform that the method will be removed in a future release: `[WARN ] HHH90000022: Hibernate's legacy org.hibernate.Criteria API is deprecated; use the JPA javax.persistence.criteria.CriteriaQuery instead`	ERROR	• <logger name="org.hibernate.orm.deprecation" level="error"/>

Hide the following LAPACK WARN message telling that the library fails to load a more optimised system library:
 	
`[WARN ] Failed to load implementation from: com.github.fommil.netlib.NativeSystemLAPACK`
 	
`[WARN ] Failed to load implementation from: com.github.fommil.netlib.NativeRefLAPACK`
	ERROR	• <logger name="com.github.fommil.netlib.LAPACK" level="error"/>

8.6 suggester.dic

The file `suggest.dic` can be customized with a list of strings to suggest via /search. Once restarted, the DHuS suggester can be request via http://dhusurl/search/suggest/{query} as shown her below:

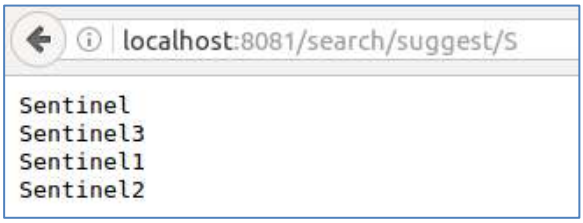

Figure 8.1 Suggester configuration

8.7 opensearch-description-file.xml

This file describes DHuS OpenSearch interface of Data Hub service. The OpenSearch description document describes the web interface of the DHuS search engine. Please refer to [RD 3] for further details about the OpenSearch description document.

8.8 temporary_files_ehcahe.xml

[Groups]/ Parameter	Description	Usage
`[ehcache]/ehcache`	It is the name of the cache manager.	• `Name`: name of the cache manager
`[ehcache]/cache`	This field defines the properties of the netCDF cache. The following attributes shall be configured only in Scalability 2.0 mode: • `Eternal;` • `diskPersistent;` • `diskExpiryThreadIntervalSeconds;` • `bootstrapCacheLoaderFactor`	• `Name`: name of the temporary cache • `maxEntriesLocalHeap`: limits the number of temporary cache files (e.g. 1000) • `maxBytesLocalDisk`: limits the total size of the cache in bytes (e.g. 10000000000) • `Eternal`: true|false • `overflowToDisk`: true|false • `diskPersistent`: true|false • `diskExpiryThreadIntervalSeconds` • `cacheEventListenerFactory` • `bootstrapCacheLoaderFactory`

`[ehcache]/chache ManagerPeerProviderF actory`	This parameter shall be set only for scalability 2.0 mode. It configures the Peer Provider: echache provides two mechanisms for peer discovery, manual and automatic. To use one of the built-in peer discovery mechanisms, specify the class attribute as `net.sf.ehcache.distribution.RM ICacheManagerPeerProviderFacto ry` Set the other attributes in order to configure manual discovery.	• `Class="net.sf.ehcache.distribu tion.RMICacheManagerPeerProvid erFactory"` • `Properties:` o `peerDiscovery= manual` o `rmiUrls= <RMI_URLS>:` list of all cache peers for all cache listener (detailed via IP address and port) In the `rmiUrls` attribute use the "|" character in order to separate each peer. Do not include the server in the list. Use the character set in the `propertySeparator` field to separate all these propertie (e.g. ",")
`[ehcache]/chache ManagerPeerListenerF actory`	This parameter shall be set only for scalability 2.0 mode. This field enable the listening from peers to the current cache Manager. The attributes of cacheManagerPeerListenerFactory are: • `class` - a fully qualified factory class name • `properties` - comma separated properties having meaning only to the factory.	• `Class= "net.sf.ehcache.distribution.R MICacheManagerPeerListenerFact ory"` • `properties` o `hostName=<HOST>:` hostname/IP of the host the listener is running. If unspecified, the hostname will use `InetAddress.getLocalHost(). getHostAddress()`, which corresponds to the default host network interface. **Warning:** Explicitly setting this to localhost refers to the local loopback of 127.0.0.1, which is not network visible and will cause no replications to be received from remote hosts o `port:` port of the host the listener is running o `remoteObjectPort:` port number on which the remote objects bound in the registry receive calls o `socketTimeoutMillis:` TCP/IP Socket timeout when waiting on

		response (i.e. number of seconds client sockets will wait when sending messages to this listener until they give up). By default this is 2000ms.

Example for DHuS deployed in embedded mode:

```
<ehcache name="TemporaryFilesCacheManager">
        <cache name="TemporaryFilesCache"
               maxEntriesLocalHeap="1000"
               maxBytesLocalDisk="10000000000">
               <cacheEventListenerFactory
class="fr.gael.streams.TemporaryFilesCacheListener" />
        </cache>
</ehcache>
```

Example for DhuS deployed in Scalability 2.0 mode:

```
<ehcache name="TemporaryFilesCacheManager">

<!-- Auto discovery -->
      <cacheManagerPeerProviderFactory
      class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
      properties="peerDiscovery=manual,rmiUrls=//172.30.246.11:40001/TemporaryFile
sCache|//172.30.246.46:40001/TemporaryFilesCache"
       propertySeparator=","/>
<!-- create cluster listener on port 40001 -->
      <cacheManagerPeerListenerFactory
      class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
properties="hostName=172.30.246.20,port=40001,remoteObjectPort=9999,socketTimeoutMillis=200
0" />
       <cache name="TemporaryFilesCache" eternal="false" overflowToDisk="false"
            diskPersistent="false" diskExpiryThreadIntervalSeconds="0"
            maxEntriesLocalHeap="1000"
            maxBytesLocalDisk="10000000000"
            memoryStoreEvictionPolicy="FIFO">
            <cacheEventListenerFactory
                class="fr.gael.streams.TemporaryFilesCacheListener" />
            <cacheEventListenerFactory
                class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
                properties="replicateAsynchronously=true" />
            <bootstrapCacheLoaderFactory
                class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
                properties="bootstrapAsynchronously=true"
                propertySeparator="," />
        </cache>
</ehcache>
```

8.9 dhus_ehcache_distributed.xml

The dhus_ehcache_distributed.xml configuration file allows to define the Cache Manager and, if DHuS instance is installed in Scalability 2.0 mode, it allows to share the configured caches among all DHuS instances belonging to the same cluster.

For DHuS deployed in Scalabilty 2.0 mode, this file shall be renamed to `dhus_ehcache.xml` to override the embedded configuration and enable peer-to-peer caching.

The following table summarizes the configurable parameters of the `dhus_ehcache_distributed.xml` file:

[Groups]/ Parameter	Description	Usage
`[ehcache]/ehcache`	It is the name of the cache manager.	• `name`: it sets the name of the cache manager. It must be unique.
`[ehcache]/cache`	This field defines the properties of the netCDF cache.	• `name`: it sets the name of the temporary cache. It is used to identify the cache and it must be unique. • `maxElementsInMemory`: It is a builder that sets the maximum objects to be held in memory, before they are evicted. This property can be modified dynamically while the cache is operating. If set to 0, no memory limits are foreseen. • `Eternal`: true|false. It sets whether elements are eternal. If eternal, timeouts are ignored and the element is never expired. • `timeToIdleSeconds`: It sets the time to idle for an element before it expires, i.e. the maximum amount of time between accesses before an element expires. It is only used if the element is not eternal. A value of 0 means that an element can idle for infinity. The default value is 0. • `timeToLiveSeconds`: It sets the time to live for an element before it expires, i.e. the maximum time between creation time and when an element expires. It is only used if the element is not eternal. A value of 0 means that and Element can live for infinity. The default value is 0. • `memoryStoreEvictionPolicy`: Policy enforced upon reaching the maxEntriesLocalHeap limit. Default policy is Least Recently Used (specified as LRU). Other policies available are First In First Out (specified as FIFO) and Less Frequently Used (specified as LFU)

		• `maxEntriesLocalHeap`: it sets the maximum number of objects that will be created in memory. In case of distribuited cache and no limit configured, the cache is limited by resources. If set to 0, no limits are foreseen. • `maxEntriesLocalDisk`: it sets the maximum number of objects that will be maintained in the DiskStore. The default value is zero, meaning unlimited. • `diskSpoolBufferSizeMB`: this is the size to allocate the DiskStore for a spool buffer. Writes are made to this area and then asynchronously written to disk. The default size is 30MB. • `overflowToDisk`: true|false. If set to true, the cache overflows to disk. • `transactionalMode`: it allows to enable a cache as transactions. If set, all cache operations will need to be done through transactions. To prevent users keeping references on stored elements and modifying them outside of any transaction's control, transactions also require the cache to be configured copyOnRead and copyOnWrite. Its allowed values are: o `transactionalMode="xa"` - high performance JTA/XA implementation o `transactionalMode="xa_strict"` - canonically correct JTA/XA implementation o `transactionMode="local"` - high performance local transactions involving caches only o `transactionalMode="off"` - the default, no transactions

		• `cacheEventListenerFactory`: it enables registration of listeners for cache events, such as put, remove, update, and expire. Its allowed attributes are: o `class` - a fully qualified factory class name. If no class is specified, no listener is created. There is no default. o `properties` - comma separated properties having meaning only to the factory • `persistence`: it configures the cache persistence. It is defined by the attribute 'strategy', which set the type of persistence provided by the configured cache. Its value must be one of the following value: o `localRestartable` - Enables the RestartStore and copies all cache entries (on-heap and/or off-heap) to disk. This option provides fast restartability with fault tolerant cache persistence on disk. It is available for Enterprise Ehcache users only. o `localTempSwap` - Swaps cache entries (on-heap and/or off-heap) to disk when the cache is full. It is not persistent. o `none` - Does not persist cache entries. • `bootstrapCacheLoaderFactory`: it specifies a BootstrapCacheLoader, which is called by a cache on initialisation to prepopulate itself. Its allowed attributes are: o `class` - a fully qualified factory class name. If no class is specified, no listener is created. There is no default.

		○ `properties` - comma separated properties having meaning only to the factory. ○ propertySeparator – it defines the character to separate the properties set.
`[ehcache]/chache ManagerPeerProviderF actory`	It configures the Peer Provider: echache provides two mechanisms for peer discovery, manual and automatic. To use one of the built-in peer discovery mechanisms, specify the class attribute as `net.sf.ehcache.distribution.RM ICacheManagerPeerProviderFacto ry` Set the other attributes in order to configure manual discovery.	• `Class:` a fully qualified factory class name. If no class is specified, no listener is created. There is no default. • `Properties:` comma separated properties having meaning only to the factory. ○ `peerDiscovery` – it shall be set to automatic. ○ `multicastGroupAddress` - multicast address | multicast host name ○ `multicastGroupPort` – port ○ `timeToLive` - 0-255
`[ehcache]/chache ManagerPeerListenerF actory`	This field enable the listening from peers to the current cache Manager.	• `Class:` a fully qualified factory class name. If no class is specified, no listener is created. There is no default. • `Properties:` comma separated properties having meaning only to the factory. ○ `hostName=<HOST>` – hostname/IP of the host the listener is running. If unspecified, the hostname will use `InetAddress.getLocalHost(). getHostAddress()`, which corresponds to the default host network interface. **Warning:** Explicitly setting this to localhost refers to the local loopback of 127.0.0.1, which is not network visible and will cause no replications to be received from remote hosts

		○ `port` – port of the host the listener is running ○ `socketTimeoutMillis` – TCP/IP Socket timeout when waiting on response (i.e. number of seconds client sockets will wait when sending messages to this listener until they give up). By default this is 2000ms.

The following tables describes all used cache and available within the DHuS configuration:

Cache name	Description	Objects Stored	Max number of elements	Keep in cache time
user_connections	It stores recent user connections information to make them accessible via OData (Entities Connections, Network and NetworkStatistic) when Access valve is enabled in etc/server.xml.	https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/server/http/valve/AccessInformation.java	Default 100 000 elements in memory / distributed 10 000 elements in memory	60 minutes
product	It stores recently accessed or modified products.	https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/database/object/Product.java	Default 1 000 in memory / distributed 200 000 in memory and 1 000 in disk in case of overflow	Default 60mins / distributed 10mins, evicted after 5mins if not accessed
product_count	It stores recent product counts, possibly filtered.	Integers	Default 1 000 in memory / distributed 10 000 in memory and 10 000 in disk in case of overflow	Default 60mins / distributed 10mins, evicted after 5mins if not accessed
products	It stores recently accessed product lists, possibly filtered.	Lists of https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/database/object/Product.java	Default 1 000 in memory / distributed 200 000 in memory and 1 000 in	default 60mins / distributed 10mins, evicted after 5mins if not accessed

			disk in case of overflow	
products	It stores recently accessed product list of metadata indexes.	Lists of https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/database/object/MetadataIndex.java	default 1 000 in memory / distributed 100 000 in memory and 100 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed
user	It stores recently accessed or modified users by their UUID.	https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/database/object/User.java	default 1 000 in memory / distributed 10 000 in memory and 1 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed
userByName	It stores recently accessed or modified users by their name.	https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/database/object/User.java	default 1 000 in memory / distributed 1 000 in memory and 1 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed
network_download_count	It stores number of downloads by user.	Integer	default 1 000 in memory / distributed 10 000 in memory and 1 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed
network_download_size	It stores cumulative downloads size by user.	Long	default 1 000 in memory / distributed 10 000 in memory and 1 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed

current_quotas	It stores number of running downloads by user.	Integer	10 000 in memory and 1 000 in disk in case of overflow	default 60mins / distributed 10mins, evicted after 5mins if not accessed
security_context	It is used internally to share spring security context between all web applications deployed in the DHuS tomcat (core, search, solr, odata ...).	org.springframework.security.core.context.SecurityContext	default 30 000 in memory / distributed 30 000 in memory and 3 000 in disk in case of overflow	60 minutes
user_requests	It is used internally as moving window in ProcessingQuota process.	https://github.com/SercoSPA/dhus-core/blob/master/core/src/main/java/fr/gael/dhus/server/http/valve/processings/ProcessingInformation.java	default 1 000 in memory / not in distributed	default 60mins / not in distributed
saml_saved_requests	It is used to handle the request forwarding after being authenticated by Keycloak. This cache stores URLs that require an authentication and accessed by a not logged in User.	•	•	-

Example for DHuS deployed in embedded mode:

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="ehcache.xsd"
         name="dhus_cache">

    <!-- Manual discovery -->
    <!--<cacheManagerPeerProviderFactory-->
    <!--class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"-->
    <!--properties="peerDiscovery=manual,-->
    <!--
rmiUrls=//server2:40001/product|//server2:40001/products|//server2:400001/product_count"/>-
->

    <!-- Auto discovery -->
    <cacheManagerPeerProviderFactory
        class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
        properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1,
multicastGroupPort=4446, timeToLive=32"/>

    <!-- create cluster listener on port 40001 -->
    <cacheManagerPeerListenerFactory
        class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
        properties="hostName=localhost,port=40001,socketTimeoutMillis=2000"/>
```

```
    <!-- Access Information Cache -->
    <cache name="user_connections"
           maxElementsInMemory="10000"
           eternal="false"
           timeToIdleSeconds="0"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LRU">
        <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none"/>
    </cache>

    <!-- User download cache used to manage download completion -->
    <cache name="user_download_ranges"
           maxElementsInMemory="5000"
           eternal="false"
           timeToIdleSeconds="60"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LRU">
        <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none"/>
    </cache>

    <cache name="userByName"
           maxEntriesLocalHeap="1000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="product_eviction_date"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="network_download_count"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
```

```
            eternal="false"
            diskSpoolBufferSizeMB="20"
            timeToIdleSeconds="300" timeToLiveSeconds="600"
            memoryStoreEvictionPolicy="LFU"
            transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="network_download_size"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="current_quotas"
          eternal="false"
          maxElementsInMemory="1000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600">
      <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerForQuotas"/>
      <cacheEventListenerFactory
            properties="replicateUpdatesViaCopy=true,replicateAsynchronously=false"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
      <persistence strategy="none" />
   </cache>

    <cache name="user"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="json_user"
           maxEntriesLocalHeap="1000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
```

```
            timeToIdleSeconds="300" timeToLiveSeconds="600"
            memoryStoreEvictionPolicy="LFU"
            transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="product_count"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="10000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="indexes"
           maxEntriesLocalHeap="100000"
           maxEntriesLocalDisk="100000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="product"
           maxEntriesLocalHeap="200000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="products"
           maxEntriesLocalHeap="200000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"
```

```
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="boundariesWKT"
           maxEntriesLocalHeap="1000000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="security_context"
           eternal="false"
           maxEntriesLocalHeap="30000"
           maxEntriesLocalDisk="3000"
           timeToIdleSeconds="0"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
        <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
                                   properties="replicateAsynchronously=false"/>
        <persistence strategy="none" />
    </cache>

   <cache name="saml_saved_requests"
          eternal="false"
          maxEntriesLocalHeap="30000"
          maxEntriesLocalDisk="3000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600"
          memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
        <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
                                   properties="replicateAsynchronously=false"/>
      <persistence strategy="none" />
   </cache>

</ehcache>
```

Example for DhuS deployed in Scalability 2.0 mode:

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="ehcache.xsd"
         name="dhus_cache">
```

```
    <!-- Manual discovery -->
    <!--<cacheManagerPeerProviderFactory-->
    <!--class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"-->
    <!--properties="peerDiscovery=manual,-->
    <!--
rmiUrls=//server2:40001/product|//server2:40001/products|//server2:400001/product_count"/>-
->

    <!-- Auto discovery -->
    <cacheManagerPeerProviderFactory
        class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
        properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1,
multicastGroupPort=4446, timeToLive=32"/>

    <!-- create cluster listener on port 40001 -->
    <cacheManagerPeerListenerFactory
        class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
        properties="hostName=localhost,port=40001,socketTimeoutMillis=2000"/>

    <!-- Access Information Cache -->
    <cache name="user_connections"
           maxElementsInMemory="10000"
           eternal="false"
           timeToIdleSeconds="0"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LRU">
        <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none"/>
    </cache>

    <!-- User download cache used to manage download completion -->
    <cache name="user_download_ranges"
           maxElementsInMemory="5000"
           eternal="false"
           timeToIdleSeconds="60"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LRU">
        <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none"/>
    </cache>

    <cache name="userByName"
           maxEntriesLocalHeap="1000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
```

```
    </cache>

    <cache name="product_eviction_date"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="network_download_count"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="network_download_size"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="current_quotas"
          eternal="false"
          maxElementsInMemory="1000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600">
      <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerForQuotas"/>
      <cacheEventListenerFactory
            properties="replicateUpdatesViaCopy=true,replicateAsynchronously=false"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
      <persistence strategy="none" />
   </cache>
```

```
    <cache name="user"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="json_user"
           maxEntriesLocalHeap="1000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="product_count"
           maxEntriesLocalHeap="10000"
           maxEntriesLocalDisk="10000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="indexes"
           maxEntriesLocalHeap="100000"
           maxEntriesLocalDisk="100000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="product"
           maxEntriesLocalHeap="200000"
           maxEntriesLocalDisk="1000"
           eternal="false"
```

```
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="products"
           maxEntriesLocalHeap="200000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="boundariesWKT"
           maxEntriesLocalHeap="1000000"
           maxEntriesLocalDisk="1000"
           eternal="false"
           diskSpoolBufferSizeMB="20"
           timeToIdleSeconds="300" timeToLiveSeconds="600"
           memoryStoreEvictionPolicy="LFU"
           transactionalMode="off">
        <cacheEventListenerFactory properties="replicateAsynchronously=true"

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
        <persistence strategy="none" />
    </cache>

    <cache name="security_context"
           eternal="false"
           maxEntriesLocalHeap="30000"
           maxEntriesLocalDisk="3000"
           timeToIdleSeconds="0"
           timeToLiveSeconds="3600"
           memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
        <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
                                   properties="replicateAsynchronously=false"/>
        <persistence strategy="none" />
    </cache>


</ehcache>
[dhus@de-gmesffmpup11 etc]$ cat dhus_ehcache.xml
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="ehcache.xsd"
```

```
         name="dhus_cache">

    <!-- Manual discovery -->
<cacheManagerPeerProviderFactory
        class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
       properties="peerDiscovery=manual,
       rmiUrls=//172.30.246.46:40001/user_connections|//172.30.246.46:40001/userByName|//17
2.30.246.46:40001/product_eviction_date|//172.30.246.46:40001/network_download_count|//172.
30.246.46:40001/network_download_size|//172.30.246.46:40001/user|//172.30.246.46:40001/json
_user|//172.30.246.46:40001/product_count|//172.30.246.46:40001/indexes|//172.30.246.46:400
01/product|//172.30.246.46:40001/products|//172.30.246.46:40001/boundariesWKT|//172.30.246.
46:40001/current_quotas|//172.30.246.46:40001/user_requests|//172.30.246.46:40001/security_
context|//172.30.246.11:40001/user_connections|//172.30.246.11:40001/userByName|//172.30.24
6.11:40001/product_eviction_date|//172.30.246.11:40001/network_download_count|//172.30.246.
11:40001/network_download_size|//172.30.246.11:40001/user|//172.30.246.11:40001/json_user|/
/172.30.246.11:40001/product_count|//172.30.246.11:40001/indexes|//172.30.246.11:40001/prod
uct|//172.30.246.11:40001/products|//172.30.246.11:40001/boundariesWKT|//172.30.246.11:4000
1/current_quotas|//172.30.246.11:40001/user_requests|//172.30.246.11:40001/security_context
"
       propertySeparator=","/>

   <!-- create cluster listener on port 400001 -->
   <cacheManagerPeerListenerFactory
           class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"

properties="hostName=172.30.246.20,port=40001,remoteObjectPort=9999,socketTimeoutMillis=200
0"/>

   <!-- Access Information Cache -->


   <cache name="user_connections"
          maxElementsInMemory="10000"
          eternal="false"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600"
          memoryStoreEvictionPolicy="LRU">
      <cacheEventListenerFactory
            class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none"/>
   </cache>

   <cache name="userByName"
          maxEntriesLocalHeap="1000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
```

```
   <cache name="product_eviction_date"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>

   <cache name="network_download_count"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>

   <cache name="network_download_size"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
   <cache name="user"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
```

```
   <cache name="json_user"
          maxEntriesLocalHeap="1000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
   <cache name="product_count"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="10000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>

   <cache name="indexes"
          maxEntriesLocalHeap="100000"
          maxEntriesLocalDisk="100000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>

   <cache name="product"
          maxEntriesLocalHeap="200000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>

   <cache name="products"
```

```
          maxEntriesLocalHeap="200000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
   <cache name="boundariesWKT"
          maxEntriesLocalHeap="1000000"
          maxEntriesLocalDisk="1000"
          eternal="false"
          diskSpoolBufferSizeMB="20"
          timeToIdleSeconds="300" timeToLiveSeconds="600"
          memoryStoreEvictionPolicy="LFU"
          transactionalMode="off">
      <cacheEventListenerFactory
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"
            class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
      <persistence strategy="none" />
   </cache>
   <cache name="current_quotas"
          eternal="false"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600"
          memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
        propertySeparator="," />
      <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true" />
      <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerForQuotas" />
      <persistence strategy="none" />
   </cache>

   <cache name="user_requests"
          maxElementsInMemory="1000"
          eternal="false"
          timeToIdleSeconds="900"
          timeToLiveSeconds="1800"
          memoryStoreEvictionPolicy="LRU">
      <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"/>
      <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerForQuotas" />
      <persistence strategy="none"/>
   </cache>
```

```
<cache name="security_context"
          eternal="false"
          maxEntriesLocalHeap="10000"
          maxEntriesLocalDisk="1000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600"
          memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
        propertySeparator="," />
       <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=false,replicatePuts=true,replicateUpdates=true,replicat
eUpdatesViaCopy=true,replicateRemovals=true"/>
<cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerForQuotas" />
      <persistence strategy="none" />
   </cache>

   <cache name="saml_saved_requests"
          eternal="false"
          maxEntriesLocalHeap="30000"
          maxEntriesLocalDisk="3000"
          timeToIdleSeconds="0"
          timeToLiveSeconds="3600"
          memoryStoreEvictionPolicy="LFU">
        <bootstrapCacheLoaderFactory
            class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"
            properties="bootstrapAsynchronously=true"
            propertySeparator="," />
        <cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
                                  properties="replicateAsynchronously=false"/>
      <persistence strategy="none" />
   </cache>

</ehcache>
```

To activate logs for a specific cache the following entry shall be added to the cache configuration:

```
<cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
```

As in the example below:

```
<cache name="user_connections" maxElementsInMemory="100000"  eternal="false"
timeToIdleSeconds="0"  timeToLiveSeconds="3600" memoryStoreEvictionPolicy="LRU">
   <cacheEventListenerFactory class="fr.gael.dhus.util.LoggingCacheListenerFactory" />
   <persistence strategy="none"/>
</cache>
```

9. AJS GUI configuration parameters

The DHuS is equipped with an AJS GUI. The files in charge of the GUI configuration management are located in:

- **appconfig.json**, located in [DHUSDIR]/etc/conf
- **styles.json**, located in [DHUSDIR]/var/tomcat/webapps/ROOT/config

9.1 appconfig.json

The configurability of the AJS GUI allows a wide set of configuration actions, which do not need a restart of DHuS to be applied. Due to the growth of the different centres and related installations, a configuration management module has been added into the AJS web app. It allows configuring various aspects of the GUI; mainly it is related to style, texts and layout:

- **Title** (shown in the header bar)
- **Sections visibility** (e.g. Cart, Profile, Sign In, Shapefile etc.)
- **URL and texts of the link logos** (shown in the header panel)
- **Version text** (shown in the info panel)
- **Data Hub Logo** (shown in the info panel)
- **Mission Tags** (shown in the Product List panel)
- **Mission footprint style and color** (shown in the Map panel)
- **Advanced Search Mission specific fields** (shown in Advanced Search Panel)
- **Map Layer** (shown in the Map View)

Please note that all the settings are included in the client side, thus it is possible to change a parameter without restarting the DHuS, but just doing a refresh via browser.

In the next Sections, instructions for specific features configuration are described.

9.1.1 Login badge

Two appconfig.json keys to manage the login badge configurability have been introduced:

- **`hide_login_title`**: `true|false`. Parameter that shows/hides login badge title.
- **`login_title`**: string visualized in the login badge title (if null, the default the string is: "Please login to access our services…")

```
"settings": {
   "hide_login_title":false,
   "login_title": ""
…
}
```

9.1.2 Missions

The configuration file `appconfig.json` manages mission specific filters. A **`missions`** section is present in the file, containing an array with the following structure:

```
"name": , "indexname": , "indexvalue":, "filters":[filter_array]
```

where `[filter_array]` is an array of mission-specific filters with the following structure in the case of all the mission except for Sentinel-5P one:

`"indexname": , "indexlabel": , "indexvalues" , "regex": [OPTIONAL]`

If a set of value is configured, it appears a combo box containing the list of all specified values.

Here follows an example:

```
"missions": [{
        "name": "Mission: Sentinel-1",
        "indexname": "platformname",
        "indexvalue": "Sentinel-1",
        "filters": [{
            "indexname": "filename",
            "indexlabel": "Satellite Platform",
            "indexvalues": "S1A_*|S1B_*",
            "regex": ".*"
        }, {
            "indexname": "producttype",
            "indexlabel": "Product Type",
            "indexvalues": "SLC|GRD|OCN",
            "regex": ".*"
        }, {
            "indexname": "polarisationmode",
            "indexlabel": "Polarisation",
            "indexvalues": "HH|VV|HV|VH|HH+HV|VV+VH",
            "regex": ".*"
        }]
```

9.1.2.1 Sentinel-5P Auxiliary Files filters

In the case of Sentinel-5P Mission the structure of `[filter_array]` is the following:

`"indexname": , "indexlabel": , "complexindexes":`

Here follows an example:

```
"filters": [
        {
            "indexname": "producttype",
            "indexlabel": "Input Auxiliary Files for Product Type",
            "complexindexes": [
            {
                "label": "FRESCO",
                "value": "(AUX_MET_TP OR AUX_MET_2D OR AUX_NISE__)"
            },{
```

```
                "label": "AER_LH",
                "value": "(AUX_MET_TP OR AUX_MET_2D OR AUX_NISE__ OR L2__FRESCO OR L2__NP_BD6
OR L2__AER_AI)"
            },{
                "label": "CO____",
                "value":  "(AUX_MET_TP  OR  AUX_MET_QP  OR  AUX_MET_2D  OR  AUX_CTM_CO  OR
AUX_CTMCH4)"
            },{
                "label": "CH4___",
                "value": "(AUX_MET_TP OR AUX_MET_QP OR AUX_MET_2D OR AUX_CTM_CO OR AUX_CTMCH4
OR L2__FRESCO OR L2__NP_BD6 OR L2__NP_BD7 OR L2__CO____)"
            },{
                "label": "O3__PR",
                "value":  "(AUX_MET_TP  OR  AUX_MET_2D  OR  AUX_NISE__  OR  L2__FRESCO  OR
L2__AER_AI)"
            },{
                "label": "O3_TPR",
                "value": "(AUX_MET_TP OR AUX_MET_2D OR AUX_NISE__ OR L2__FRESCO OR L2__AER_AI
OR L2__O3__PR)"
            },{
                "label": "SO2___",
                "value": "(AUX_MET_TP OR AUX_MET_QP OR AUX_MET_2D OR AUX_NISE__ OR AUX_CTMFCT
OR AUX_CTMANA OR AUX_BGSO2_ OR L2__AER_AI OR L2__O3____ OR L2__CLOUD)"
            },{
                "label": "CLOUD_",
                "value": "(AUX_MET_TP OR AUX_MET_QP OR AUX_MET_2D OR AUX_NISE__ )"
            },{
                "label": "O3____",
                "value":  "(AUX_MET_TP  OR  AUX_MET_QP  OR  AUX_MET_2D  OR  AUX_NISE__  OR
L2__CLOUD)"
            },{
                "label": "HCHO__",
                "value": "(AUX_MET_TP OR AUX_MET_QP OR AUX_MET_2D OR AUX_NISE__ OR AUX_CTMFCT
OR AUX_CTMANA OR AUX_BGHCHO OR L2__AER_AI OR L2__CLOUD)"
            },{
                "label": "AER_AI",
               "value": "AUX_MET_2D"
            },{
                "label": "NO2___",
                "value":  "(AUX_MET_2D  OR  AUX_NISE__  OR  AUX_CTMFCT  OR  L2__FRESCO  OR
L2__AER_AI)"
            },{
```

```
                "label": "O3_TCL",
                "value": "L2__O3____"
            }],
            "regex": ".*"
        }]
}]
```

9.1.3 First node inspection

The first node inspection in the product details dialog can be managed by means of `regex_closed_by_default` parameter to be added in the `settings` section of the `etc/conf/appconfig.json` file.

```
"settings": {
   "regex_closed_by_default": "(.*)",
…
}
```

The value of the parameter should be a javascript regular expression with the identifier of products whose ispection is to be avoided. If all products must be inspectable when showing product details, there is no need to add the configuration parameter.

Here follow some examples:

1. no products inspectable at panel opening:

   `"regex_closed_by_default": "(.*)",`

2. all products inspectable at panel opening:

   `"regex_closed_by_default": "",`

3. no products inspectable for a speciphic mission at panel opening, e.g. Sentinel-5P:

   `"regex_closed_by_default": "(^S5P.*)",`

4. no products inspectable for several mission at panel opening, e.g. Sentinel-1, Sentinel-2A:

   `"regex_closed_by_default": "(^S1.*|^S2A.*)",`

9.1.4 Shapefile

The configuration parameter **`enable_shapefile`** of the file "appconfig.json" enable/disable the shapefile usage for area selection.

The following properties can be configured for the shapefile management:

- `shapefile_description`: description of shapefiles that can be uploaded;
- `shapefile_max_size`: max file size supported;
- `shapefile_max_size_message`: error messages visualized in case of too large shape file upload;

- `max_points_shape_file`: max number of points in the polygon used by DHuS to re-construct the AOI.

Here after a sample of `appconfig.json` with the configuration for the parameters listed above:

```
"enable_shapefile": true,

"shapefile_description": "You can also drag & drop the shapefile on map. Only files with extension .shp are supported. Shapefiles are limited to one record of type POLYGON and must be no larger than 5.00 MB.",

"shapefile_max_size": 5242880,

"shapefile_max_size_message": "Your shapefile cannot be uploaded because it is too large. Shapefiles must be no larger than 5.00 MB.",

"max_points_shape_file": 50
```

The configured shapefile description is displayed clicking on the "Hamburger Button" in the left corner of the Advanced Search Panel:

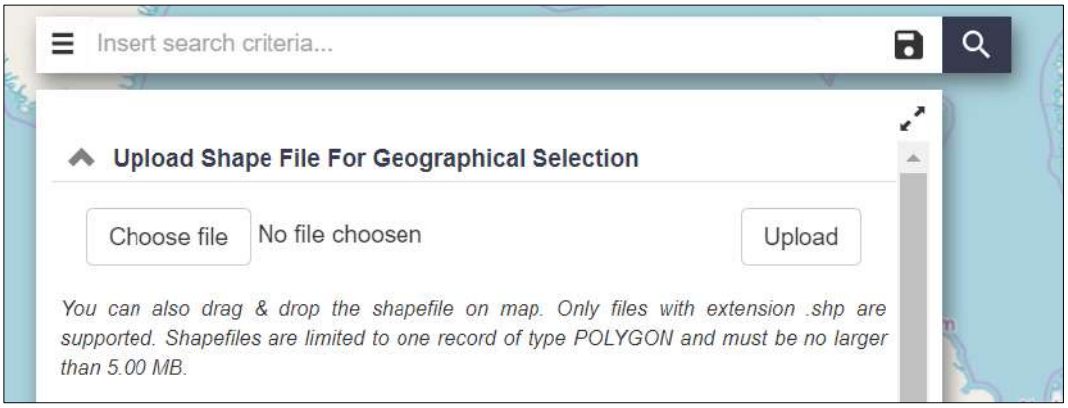

Figure 9.1 Shapefile description display

9.1.5 Terms and conditions link

The T&C link is configurable through the "**`t_and_c`**" parameter in the "settings" section.

9.1.6 List pagination

It is possible to configure the pagination limit via the parameter "`pagination_limit`". The recommended configuration is [25, 50, 75, 100, 125, 150].

Please note that configuring this parameter is mandatory and, since AJS uses the SolR engine, the pagination limit cannot exceed the row limitations configured for SolR (please note that the parameter defining the maximum number of rows retrievable via OpenSearch is `max.product.page.size` of the start.sh).

9.1.7 Deletion cause

AJS GUI configuration file allows the configuration of deletion causes with the following parameter:

```
"settings": {
...,
"deleted_product": { "show_deletion_cause": true, "causes": ["Invalid: bad
geometry","Invalid: reprocessed"], "default_cause": "Invalid" },
...
}
```

Here follows the explanation about the mentioned line:

- `deleted_product.show_deletion_cause (true|false)`: if true the prompt message to insert deletion cause is shown, if false the prompt message is not shown
- `deleted_product.causes`: is the array containing available deletion causes.
- `deleted_product.default_cause`: contains the default cause used when end user inserts a deletion cause not matching with one of the available deletion causes

9.1.8 Map polygon selection/pan management

The management of the map pan function and polygon selection has been upgraded:

- The UI presents one button on the up-right of the map. This button allows the activation of the pan functionality and the draw selection functionality when the user clicks on it.
- If the button is set on "Navigation Mode", the user is able to move the map using both left and right button of the mouse.
- If the button is set on "Area Mode", the user is able to:
  - draw a polygon clicking on the left button of the mouse (each click draws a vertex) and closing the polygon on double click;
  - draw a squared polygon when clicking and shifting the mouse on the map (holding the click);
  - move the map with the right button of the mouse.

The foreseen default mode for the Map is the "Navigation Mode" (`"draw_by_default": false`), but it is possible to switch into "draw" mode setting the following parameter in the appconfig.json:

```
{
   ...
   "settings":{
   ...
   "draw_by_default": true,
   ...
   }
}
```

On "Area Mode" mode, the polygon drawing starts only if no selection area is present on the map, otherwise the click will clear the map.

> Please note that the old configuration parameter "show_map_toolbar" has been removed, so it is no possible to restore the map toolbar via configuration file.

9.1.9 Map layers attributions configurability

It is possible to configure map layers attributions in order to display them in the lower right corner of the map, they can be added in the `appconfig.json` file.

The attribution of each layers are the following:

- Terrain:

```
Terrain: { Data © <a
href=\"http://www.openstreetmap.org/copyright\">OpenStreetMap</a> contributors and
<a href=\"#data\">others</a>, Rendering © <a
href=\"https://github.com/mapserver/basemaps\">MapServer</a> and <a
href=\"http://eox.at\">EOX</a>}
```

- Overlay:

```
Overlay: { Data © <a
href=\"http://www.openstreetmap.org/copyright\">OpenStreetMap</a> contributors,
Processing & Rendering © <a
href=\"https://github.com/mapserver/basemaps\">MapServer</a> and <a
href=\"http://eox.at\">EOX</a> }
```

- OpenStreet:

```
Open Street: { Data © <a
href=\"http://www.openstreetmap.org/copyright\">OpenStreetMap</a> contributors,
Rendering © <a  href=\"https://github.com/mapserver/basemaps\">MapServer</a> and <a
href=\"http://eox.at\">EOX</a>}
```

- Sentinel-2 Cloudless

```
Sentinel-2 Cloudless: { Data © Contains modified Copernicus Sentinel data 2016 &amp;
2017 }
```

- Blue Marble

```
Blue Marble { Data © <a href=\"http://nasa.gov\">NASA</a> }
```

Map layers attributions can be added in the `appconfig.json` file in the following way (one for each layer's source):

```
"Road": {
         "sources": [{
                       "class": "TileWMS",
                       "attributions": "Open Street: { Data © <a
href=\"http://www.openstreetmap.org/copyright\">OpenStreetMap</a> contributors, Rendering ©
<a  href=\"https://github.com/mapserver/basemaps\">MapServer</a> and <a
href=\"http://eox.at\">EOX</a>}",
                       "params": {
                                  "url": "https://tiles.esa.maps.eox.at/wms",
                                  "wrapX": false,
                                  "params":{ "LAYERS": "osm_3857", "TILED": true }
                                 }
                    }],
                       "title": "Open Street",
                       "type": "base",
                       "visible": true
        }
```

The configured attributions are displayed in the lower right corner of the map as Figure 9.2 (corresponding to the described example):

Figure 9.2 Map layer attribution display

9.1.10 Configuration of more than three map layers

It is possible to configure more than three map layers via `appconfig.json` file (as per the schema reported below).

Examples of map layers are Satellite, Hybrid, Road, Humanitarian, etc.

```
"map": {
              "Satellite": {
                  ...
                          },
              "Road": {
                  ...
                     },
              "Hybrid": {
                  ...
                       },
              "Humanitarian": {
                  ...
                                 }
        }
```

9.1.11 Map layer selector restyling

The map layer panel presents a preview of the configured layers, centered on central Italy.

Figure 9.3 Map layer panel

9.1.12 Sorting criteria for datatake sensing start and tile id

AJS GUI configuration file allows the configuration a sorting criteria for datatake sensing start and tile id. This sorting criteria is meaningful only for Sentinel-2 L1C products.

Here follow an example on how configure this criteria:

```
"sortOptions": [
{ ...},
{
       "name": "TileId",
       "value": "datatakesensingstart,hv_order_tileid",
       "message": "This sorting is applicable only for Sentinel-2 L1C products. Clicking on
this checkbox, only Sentinel-2 L1C products will be shown in the list.",
       "filter": "platformname:Sentinel-2 AND producttype:S2MSI1C",
       "show_checkbox": true,
       "checkbox_checked": false,
       "disable_filters": true
}]
```

The following parameters are managed within the "sortOptions" section of the `appconfig.json` file:

- filter: (String) a configurable filter associated to the selected sorting. This filter allows to show in the list only products for which the sorting is meaningful (e.g. tileid or cloudcoverpercentage)
- message: (String) a configurable message shown when selecting the sorting creteria explaining the effects of the selected sorting
- show_checkbox: (Boolean) a configurable parameter used to show a checkbox to the left of the message. Selecting or deselecting the checkbox, the filter associated to the sorting creteria is applied/removed
- checkbox_checked: (Boolean) a configurable parameter used to set as "checked" by default the checkbox shown at the left of the message. Obviously this parameter has sense only if show_checkbox parameters is defined and equals to true
- disable_filters: (Boolean) a configurable parameter used to disable the mission specific filters.

9.1.13 Enlarging thumbnail

The parameter `"hide_popover_list"` in the setting section of the `appconfig.json` file allows enabling the popover feature to enlarge thumbnail on mouseover.

Here follows the explanation about the mentioned line:

- `"hide_popover_list": false` makes the popover as active. This is the default behaviour, i.e. if the parameter is not present, thumnails will enlarge at the mause crossing.
- `"hide_popover_list": true` makes the popover feature disabled.

```
{
   ...
   "settings":{

   "hide_popover_list": false,
```

```
...
    }
}
```

9.1.14 Graticule

The 'Graticule' feature allows to show the network of lines of latitude and longitude upon which the map is drawn. Different configuration are available, e.g. for step, resolution or style.

- The "hide_graticule" configuration parameter for show/hide Graticule checkbox can be added in the settings section of the appconf.json.

```
{
...
"settings":{
"hide_graticule": false,
...
}
}
```

When the property "hide_graticule" is set to "true", the Graticule checkbox is not shown on the Map. To show it, set to 'false' this property.

9.1.15 GDPR Activation

DHuS 3.0.X distribution coming with an updated appconfig.json that allows to activate the GDPR and set some parameter useful to a proper DHuS-Keycloak interface.

```
{
...
"settings":{
...
"gdpr": {
    "enabled": true,
    "showUsername": false,
    "signupUrl": "https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-
actions/registration?client_id=<CLIENT>",
    "forgotPasswordUrl": "https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-
actions/reset-credentials?client_id=<CLIENT>",
    "editProfileUrl": "https://<KEYCLOAK_ADDRESS>/auth/realms/<REALM_NAME>/login-
actions/authenticate?client_id=<CLIENT>",
    "adminConsoleUrl":
"https://<KEYCLOAK_ADDRESS>/auth/realms/master/protocol/openid-
connect/auth?client_id=security-admin-console"
}
...
}
}
```

where:

- <KEYCLOAK_ADDRESS> is the address of the Keycloak instance.
- <CLIENT> is the DHuS client defined on Keycloak and related to the DHuS instance which the AJS configuration file refers to.
- <REALM_NAME> is the Realm defined on Keycloak and related to the DHuS instance which the AJS configuration file refers to.

By default, the GDPR is not enabled and the Username is shown. To activate the GDPR and to be compliant with it, the "enabled" parameter shall be set to true and the "showUsername" one to false.
The other parameters shall be configured according to Operational configuration.

9.2 styles.json

This configuration file allows configuration of product footprints layout, according to the product instrument.

Parameter	Description	Synthax
matches	Product Instrument	• name: it defines the group title • value: instrument name. It defines the flag that appears in the UI near each product name in the product list. Refer to Figure 9.4.
styles	RGB colour for footprint	• fill o color= it defines the RGB colors for the footprint fill • stroke o color= it defines the RGB colors for the footprint borders o width= it defines the thickness of the footprint borders
selected_styles	RGB colour for footprint selction	See the configuration parameters for the styles field above.
highlighted_styles	RGB colour for highlighted footprint i.e. when the mouse passes over them	See the configuration parameters for the styles field above.
label_style	Background color of the flag defining the mission that appears in the UI near each product name in the product list.	background-color: mission background color specified in HEX format.
instrlabel_style	Background color of the flag defining the instrument that appears in the UI near each product name in the product list. Color should be set in HEX format.	background-color: instrument background color specified in HEX format.

Example:

```
[
    {
       "matches": [
```

```
            {
                "name": "Instrument",
                "value": "SAR-C"
            }
        ],
        "styles": {
            "fill": {
                "color": [
                    179,
                    27,
                    27,
                    0.1
                ]
            },
            "stroke": {
                "color": [
                    179,
                    27,
                    27,
                    1
                ],
                "width": 1.5
            }
        },
        "selected_styles": {
            "fill": {
                "color": [
                    255,
                    0,
                    0,
                    0.3
                ]
            },
            "stroke": {
                "color": [
                    255,
                    0,
                    0,
                    1
                ],
                "width": 3.5
            }
        },
        "highlighted_styles": {
            "fill": {
                "color": [
                    179,
                    27,
                    27,
                    0.3
                ]
            },
            "stroke": {
                "color": [
                    179,
                    27,
                    27,
                    1
                ],
                "width": 3
```

```
            }
        },
        "label_style": "background-color: #dc143c !important;",
        "instrlabel_style": "background-color: #dc143c !important;"
    }
]
```

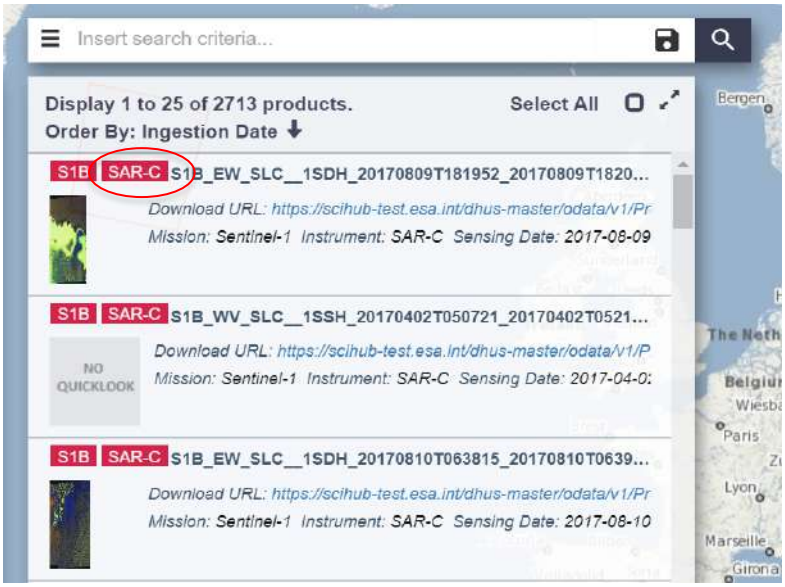

Figure 9.4 Instrument label as defined in the `value` field of the `styles.json` file.

10. Configuration converter script

A script named "`updateConfiguration.sh`" for converting the dhus.xml file of previous DHuS version is included in the distribution.

In order to run the script it is needed to launch the following command in the machine where DHuS is installed:

```
./updateConfiguration.sh /path_to_DHUS_inst_folder/etc/dhus.xml
```

Please be informed of the following outputs after running the updateConfiguration.sh script on a dhus.xml file coming from the previous official DHuS release (i.e. DHuS 2.7.7):

- The following parameter is automatically added to existing Synchronizers:
    - `<sync:syncOfflineProducts>false</sync:syncOfflineProducts>`

  Its value is put equal to false, i.e. only synchronization of online products in the Data Source is allowed.
- The HFSLocation defined in existing PDGSDataStores and paramPDGSDataStores is automatically replaced by the DataStoreCache element, like the following one:

```
<ds:dataStore xsi:type="ds:hfsDataStoreConf" name="PDGSDataStore-HfsCache"

restriction="none" maximumSize="2147483648" currentSize="0"

autoEviction="false">
```

```
        <ds:evictionName>Linked_S2</ds:evictionName>
        <ds:path>/COMMON/IVV/PDGS_Broker/S3_hfslocation</ds:path>
        <ds:maxFileNo>10</ds:maxFileNo>
        <ds:maxItems>1024</ds:maxItems>
</ds:dataStore>
```

Starting from the existing HFSLocation definition, the new element is built as follows:

- `name`: it is set equal to the DataStore Name plus 'HfsCache'.
- `restriction`: It is set equal to the default value, i.e. 'none'.
- `maximumSize`: It is set equal to the value of the corresponding parameter at PDGSDataStore or paramPDGSDataStore level. The parameter at PDGSDataStore or paramPDGSDataStore level is also removed.
- `currentSize`: It is set equal to the value of the corresponding parameter at PDGSDataStore or paramPDGSDataStore level. The parameter at PDGSDataStore or paramPDGSDataStore level is also removed.
- `autoEviction`: It is set equal to the value of the corresponding parameter at PDGSDataStore or paramPDGSDataStore level. The parameter at PDGSDataStore or paramPDGSDataStore level is also removed.
- `evictionName`: It is set equal to the Eviction linked to the PDGSDataStore or paramPDGSDataStore. The parameter at PDGSDataStore or paramPDGSDataStore level is also removed.
- `path`: It is set equal to the path defined in the old HFSLocation parameter.
- `maxFileNo`: It is set equal to the default value, i.e. 10.
- `maxItems`: It is set equal to the default value, i.e. 1024.

- The new Eviction's Attributes `safeMode`, `targetDatastore` and `baseDate` are not automatically inserted in existing Evictions' configuration by the script. Their default values are used:
  - `safeMode` = false
  - `targetDataStore` = none
  - `baseDate` = modificationDate
- The GDPR parameters are not automatically added in the <system> section. Their default values are used:
  - `gdpr` = false
  - `SAMLUserId` = samlId

11. Migration Database tool

The migration of non-embedded databases is not performed automatically by the DHuS at start. At the start, DHuS only checks that the Database's schema matches the current schema for this specific version of the DHuS.

A new script "`updateDatabase.sh`" is present within DHuS distribution package. It is responsible only of the Database migration from previous DHuS versions.

The new script is in the same location as the updateConfiguration.sh script and it can be launched with the following command:

- `nohup ./updateDatabase.sh <JDBCDriver> <JDBCUrl> <login> <password> > log_script.txt &`

where the parameters <JDBCDriver>, <JDBCUrl>, <login> and <password> are the ones reported in the <system:database> element in the dhus.xml configuration file.

The script shall be launched only during installation in externalized mode and before DHuS start. A log file is generated where migration steps are reported together with any error could occur. The statement "Success!" indicates that the migration successfully ends.

Please note that for a successful Database migration, a free space of at least 25% of the size of the Database to migrate shall be available.

12. Deletion User script

As highlighted in the previous Chapters, DHuS 3.0.X branch has been adapted to be compliant with GDPR requirements.

The main consequence is that the User management is fully redirected to Keycloak and no more User information are stored at DHuS level.

To cope with this, a script "`updateGDPRUsers.sh`" to delete all User information from DHuS instance is provided within DHuS 3.0.X distributions.

The new script is in the same location of the updateConfiguration.sh and updateDatabase.sh script and it can be launched with the following command:

- `nohup ./updateGDPRUsers.sh <JDBCDriver> <JDBCUrl> <login> <password> > log_script.txt &`

where the parameters <JDBCDriver>, <JDBCUrl>, <login> and <password> are the ones reported in the <system:database> element in the dhus.xml configuration file.

The script can be launched on DHuS installed both in embedded and externalized mode and shall be launched before DHuS start. A log file is generated where Users deletion steps are reported together with any error could occur. The statement "Success!" indicates that the Users deletion successfully ends.

By processing 10000 Users at a time, the script will delete all the following Users info:

- Users stored information (password, firstname, lastname, email, phone, address, country, domain, usage)
- User roles
- Saved searches
- Product carts
- User transformation
- Order owners
- User restrictions
- Store quotas
- Network usage

Please note that the script shall be run after the User migration, in case the already registered Users on DhuS instance shall be migrated on Keycloak. Please refer to [RD 12 for more details on Users migration.

13. GDPR - Query via command line

By activating the GDPR, the current DHuS SW behaviour is ensured. In particular, it is possible to interrogate the instance both via Graphical Interface, OData, OpenSearch and command line.

Indeed, the basic authentication is still supported as authentication mode and it can be used to access the DHuS instance via command line.

From an Operational point of view, it is important to note that each request executed via command line generates a new User session in Keycloak.

In order to mitigate this behaviour, a script `keycloak_auth_and_get_dhus_cookie_v2` has been developed in order to retrieve cookies and to use them instead of credentials to interrogate the DHuS instance.

Please find below the command to run the script:

```
./keycloak_auth_and_get_dhus_cookie_v2 "<DHUS_ADDRESS>" <USERNAME> <PASSWORD>
```

where:

- <DHUS_ADDRESS> is the address of the DHuS instance to interrogate
- <USERNAME> and <PASSWORD> are the credentials of Users that need to access DHuS instance

As consequence, a file "cookies" is generated and it can be used to execute the query via command line with the following command:

```
curl -b cookies "<ODATA_QUERY>"
```

where:

- <ODATA_QUERY> is the query to be executed towards DHuS instance

Appendix A. Example of dhus.xml file coming from distribution

```
<!DOCTYPE configuration [
  <!ENTITY varFolder "local_dhus">
]>
<!--
  DHuS configuration
  workingDir working directory path. default is "" which leads the DHuS to use a folder
  "dhus" created in the default temporary-file directory (specified by the system property
"java.io.tmpdir")
-->
<configuration workingDir="" xmlns="fr.gael.dhus.database.object.config"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:ds="org.dhus.store.datastore.config"
               xmlns:cron="fr.gael.dhus.database.object.config.cron"
               xmlns:msg="fr.gael.dhus.database.object.config.messaging"
               xmlns:network="fr.gael.dhus.database.object.config.network"
               xmlns:product="fr.gael.dhus.database.object.config.product"
               xmlns:search="fr.gael.dhus.database.object.config.search"
               xmlns:server="fr.gael.dhus.database.object.config.server"
               xmlns:system="fr.gael.dhus.database.object.config.system"
               xmlns:scanner="fr.gael.dhus.database.object.config.scanner"
               xmlns:evict="fr.gael.dhus.database.object.config.eviction"
               xmlns:sync="fr.gael.dhus.database.object.config.synchronizer"
               xmlns:source="fr.gael.dhus.database.object.config.source"
               xmlns:ps="fr.gael.dhus.database.object.config.productsource">
   <!-- Crons configuration. They all have two parameters:
      - active: defines if the cron is currently active or not
      - schedule: defines the schedule of the cron.
         A schedule is configured according to this pattern:
            Seconds Minutes   Hours Day-of-month Month Day-of-week [Year]
         You can find more information on http://www.quartz-
scheduler.org/documentation/quartz-1.x/tutorials/TutorialLesson06
         Here are some simple examples:
            - "0 0 */1 ** ?": every hour.
            - "0 0 9-17 ? * MON-FRI": on the hour nine to five week days.
            - "0 0 0 25 DEC ?": every Christmas Day at midnight
            - "0 0 3 ? * *": every day at 3 AM -->
   <crons>
      <!-- Cron used to synchronize local archive, defined in system/archive/@path -->
      <cron:archiveSynchronization active="false" schedule="0 0 0/5 ? * *"/>
      <!-- Cron used to clean database, like removing old statistics or old not confirmed
users. -->
      <cron:cleanDatabase active="true" schedule="0 0 1 ? * *">
         <!-- Definition of the time (in days) for user to confirm its registration -->
         <cron:tempUsers keepPeriod="10" />
         <!-- Definition of the time (in days) system is keeping statistics -->
         <cron:statistics keepPeriod="90" />
      </cron:cleanDatabase>
      <!-- Cron used to dump database -->
      <cron:dumpDatabase active="true" schedule="0 0 3 ? * *" />
      <!-- Cron used to clean database dumps.
         - keep: defines how dumps are stored. -->
      <cron:cleanDatabaseDump active="true" schedule="0 0 4 ? * *" keep="10" />
      <!-- Cron used to execute user saved searches and send results to users -->
      <cron:searches active="true" schedule="0 0 5 ? * *" />
      <!-- Cron used to send system logs.
         - addresses: logs recipients addresses. They shall be coma-separated -->
      <cron:sendLogs active="true" schedule="0 55 23 ? * *" addresses="dhus@xxx.xx" />
      <!-- Cron used to check all system coherence, including database optimization -->
```

```
      <cron:systemCheck active="true" schedule="0 30 23 ? * *"/>
   </crons>
   <messaging>
      <!-- Mail configuration. These values are used only at first launch of
         the system. They shall be modified in Management Panel if needed.
         - onUserCreate: defines if system send mail when creating user
         - onUserUpdate: defines if system send mail when updating user
         - onUserDelete: defines if system send mail when deleting user -->
      <msg:mail onUserCreate="true" onUserUpdate="true" onUserDelete="true">
         <!-- Mail server definition.
            - smtp: server address
            - port: server port
            - tls: defines if server is using TLS protocol
            - username and password : connection information -->
         <msg:server smtp="smtp.xxx.xx" port="587" tls="false" username="dhus@xxx.xx"
password="password">
            <!-- Information used in "from" part of sent mails.
               - name: displayed name of "from" part
               - address: displayed address of "from" part -->
            <msg:from name="DHuS Support Team" address="dhus@xxx.xx" />
            <!-- Defines the "reply to" address of sent mails -->
            <msg:replyTo>dhus@xxx.xx</msg:replyTo>
         </msg:server>
      </msg:mail>
   </messaging>
   <!-- Network configuration -->
   <network>
      <network:outbound>
         <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
               <network:includes>
                  <network:include>
                     <network:userEmailPattern>.*@xxx.xx</network:userEmailPattern>
                  </network:include>
               </network:includes>
            </network:classifier>
         </network:channel>
         <network:channel name="SelfRegisteredChannel" weight="25">
            <network:defaultUserQuotas>
               <!-- maxConcurrent defines the maximum simultaneous accepted
                   transfers. -->
               <network:maxConcurrent>5</network:maxConcurrent>
               <!-- maxCount defines the maximum number of accepted transfers on
                   a customizable period.
                   "periodUnit" attribute defines the unit of the period. Possible
                   units are "DAYS", "HOURS", "MICROSECONDS", "MILLISECONDS",
                   "MINUTES", "NANOSECONDS", "SECONDS".
                   "period" attribute is a sliding time window used to count
                   number of transfered product from now to the past period delay. -->
               <network:maxCount period="1" periodUnit="DAYS">50</network:maxCount>
               <!-- Maximum accepted size of transfer file. -->
               <network:maxSize>10737418240</network:maxSize> <!-- 10Gb -->
               <!-- maxCumulativeSize defines the maximum cumulated accepted
                   transfers size on a customizable period.
                   period/periodUnit attributes are defined in maxCount description. -->
               <!-- <maxCumulativeSize period="7"
periodUnit="DAYS">1099511627776</maxCumulativeSize> -->
               <!-- maxBandwidth maximum bandwidth authorized for this channel. -->
               <network:maxBandwidth>13107200</network:maxBandwidth> <!-- 100Mb/s -->
            </network:defaultUserQuotas>
         </network:channel>
      </network:outbound>
      <network:inbound>
         <network:channel name="PriorityChannel" weight="75">
            <network:classifier>
```

```
                <network:includes>
                   <network:include>
                      <network:userEmailPattern>.*@xxx.xx</network:userEmailPattern>
                   </network:include>
                </network:includes>
             </network:classifier>
          </network:channel>
          <network:channel name="SelfRegisteredChannel" weight="25">
             <network:defaultUserQuotas>
                <!-- maxConcurrent defines the maximum simultaneous accepted
                    transfers. -->
                <network:maxConcurrent>5</network:maxConcurrent>
                <!-- maxCount defines the maximum number of accepted transfers on
                    a customizable period.
                    "periodUnit" attribute defines the unit of the period. Possible
                    units are "DAYS", "HOURS", "MICROSECONDS", "MILLISECONDS",
                    "MINUTES", "NANOSECONDS", "SECONDS".
                    "period" attribute is a sliding time window used to count
                    number of transfered product from now to the past period delay. -->
                <network:maxCount period="1" periodUnit="DAYS">50</network:maxCount>
                <!-- Maximum accepted size of transfer file. -->
                <network:maxSize>10737418240</network:maxSize> <!-- 10Gb -->
                <!-- maxCumulativeSize defines the maximum cumulated accepted
                    transfers size on a customizable period.
                    period/periodUnit attributes are defined in maxCount description. -->
                <!-- <maxCumulativeSize period="7"
periodUnit="DAYS">1099511627776</maxCumulativeSize> -->
                <!-- maxBandwidth maximum bandwidth authorized for this channel. -->
                <network:maxBandwidth>13107200</network:maxBandwidth> <!-- 100Mb/s -->
             </network:defaultUserQuotas>
          </network:channel>
       </network:inbound>
    </network>
    <!-- Products configuration -->
    <products>
       <!-- Download configuration
          - compressionLevel: is a value to build the Deflater, from 1 to 9.
          The compression level defines how rigorously the compressor looks
          to find the longest string possible. As a general rule of thumb:
          - Compressing at the maximum level (9) requires around twice as
          much processor time as compressing at the minimum level (1)
          - For typical input, compressing at the maximum as opposed to
          the minimum level adds around 5% to the compression ratio.
          0 value means no compression.
          - checksumAlgorithms: shall be coma-separated -->
       <product:download compressionLevel="4" checksumAlgorithms="MD5"/>
       <!-- Quicklook calculation parameters
          - height: height of generated quicklooks
          - width: width of generated quicklooks
          - cutting: allow system to cut image when processing quicklooks -->
       <product:quicklook height="512" width="512" cutting="false" />
       <!-- Thumbnail calculation parameters
          - height: height of generated thumbnails
          - width: width of generated thumbnails
          - cutting: allow system to cut image when processing thumbnails -->
       <product:thumbnail height="64" width="64" cutting="false" />
    </products>
    <!-- Search configuration -->
    <search>
       <!-- Geocoder configuration
          - url: geocoder url -->
       <search:geocoder url="http://nominatim.openstreetmap.org">
          <!-- Nominatim geocoder configuration
             - boundingBox: defines if the geocoder is querying only
             the bounding box of the matching place from the Nominatim Web
```

```
            Service i.e. the four corners encompassing the place.
            Otherwise, it will query the complete polygon boundaries, that
            may have lower performance according the the number of vertices
            composing the place's boundaries.
            - maxPointNumber: maximum number of points that can be returned
            for a polygon -->
         <search:nominatim boundingBox="false" maxPointNumber="50" />
         <!-- Geoname geocoder configuration
            - username: username used to connect to Geoname -->
         <search:geoname username="username" />
      </search:geocoder>
      <!-- Odata configuration.
         - defaultTop: default top used by Odata when none is given -->
      <search:odata defaultTop="50" />
      <!-- Solr configuration.
         path: solr path
         core: solr core name
         schemaPath: solr schema path. Shall be empty.
         synonymPath: path of solr synonyms file -->
      <search:solr path="&varFolder;/solr" core="dhus" schemaPath="" synonymPath="" />
   </search>
   <!-- Server configuration.-->
   <server>
      <!-- External url (protocol://host:port/path) is the url viewed by users
         Used in case of an apache proxy redirection for example
         Empty values mean that server values are those which are viewed by users -->
      <server:external protocol="" host="" port="-1" path="" />
   </server>
   <!-- System configuration -->
   <system>
      <!-- Definition of principal administrator user.
         If User exists, DHuS will give him all rights at launch, but will not
         his password. This shall be done in Management panel of GUI.
         If User is not existing, DHuS will create it with defined password.
         -->
      <system:administrator name="root" password="password" />
      <!-- Definition of local archive path. -->
      <system:archive errorPath="" />
      <!-- Definition of database path and where dumps are stored -->
      <system:database JDBCDriver="org.hsqldb.jdbcDriver"
hibernateDialect="org.hibernate.dialect.HSQLDialect" dumpPath="&varFolder;/database_dump"

JDBCUrl="jdbc:hsqldb:file:&varFolder;/database/hsqldb;shutdown=true;hsqldb.write_delay=fals
e;hsqldb.cache_size=1000000;hsqldb.cache_rows=200000;hsqldb.log_size=50;hsqldb.tx=mvcc;hsql
db.default_table_type=cached;hsqldb.nio_max_size=20480;hsqldb.large_data=true;" login="sa"
password="" />
      <!-- Definition of system long name and short name. -->
      <system:name long="Data Hub Service" short="DHuS" />
      <!-- Processing configuration.
         corePoolSize: defines maximal number of active threads. Default is 1 -->
      <system:processing corePoolSize="4" />
      <!-- Definition of support name and mail.
         - registrationMail: used to send the administrative registration information. If
this
         field is not set, DHuS is using support mail.
         These values are used only at first launch of the system.
         They shall be modified in Management Panel if needed. -->
      <system:support name="DHuS Support Team" mail="dhus@xxx.xx"
registrationMail="dhus@xxx.xx" />i
      <!-- Definition of tomcat path -->
      <system:tomcat path="&varFolder;/tomcat" />
      <!-- Executor (background service that executes synchronizers)
          It must be enabled if you want to use the synchronisation feature.
          batchMode: the executor will run the synchronizers until there is no more to
synchronize. -->
```

```
      <system:executor enabled="false" batchModeEnabled="false" />
      <!-- Evicted products will be copied in the trashPath repository -->
      <system:trashPath path="" />

      <!-- Transformation configuration
        - isMaster: must be set to true if this instance should run transformations and
ingest resulting products (default: false)
        - interval: how often the instance should run the transformation management
routine, in milliseconds (default: 60000)
        - targetDataStore: data store in which to ingest transformed products (default: all
data stores)
        - ingestCorePoolSize: how many transformed products can be ingested in parallel
(default: 4)
       -->
      <system:transformation
        isMaster="true"
        interval="60000"
        targetDataStore="transformationDataStore"
        ingestCorePoolSize="4">

        <!-- Transformer quotas configuration (optional)
          - transformerName: name of the transformer on which to apply quotas (required)
          - maxPendingRequests: how many transformations can be pending at once for this
transformer (default: 4)
          - maxRunningRequests: how many transformations can be running ot once for this
transformer (default: 4)
        -->
        <system:transformerQuotas transformerName="MyTransformer" maxPendingRequests="32"
maxRunningRequests="4"/>

        <!-- User quotas configuration (optional)
          - maxParallelTransformationsPerUser: how many transformations can users request
in parallel (required)
        -->
        <system:userQuotas maxParallelTransformationsPerUser="2"/>
      </system:transformation>
   </system>

   <!-- DataStores Configuration -->
   <dataStores>
      <!-- All DataStores share some attributes:
         name: Name of this dataStore (must be unique)
         restriction: (default: none)
           - none             read-write
           - readOnly         read-only
           - referencesOnly   add/remove references to data only, does not allow
addition/removal of the data itself
         priority: Priority used to sort DataStores. A datastore with the lowest value will
answer first (default: 100) -->

      <!-- HFS DataStore configuration -->
      <ds:dataStore xsi:type="ds:hfsDataStoreConf"
                    name="synchronized-hfs-without-copy"
                    restriction="referencesOnly">
         <!-- Path on disk to this HFS DataStore -->
         <ds:path>/remote/incoming/path</ds:path>
         <!-- Maximum number of sub-folder in one folder (default: 10, at least 5)-->
         <ds:maxFileNo>10</ds:maxFileNo>
         <!-- Maximum number of items in one folder (default: 1024, at least 10)-->
         <ds:maxItems>1024</ds:maxItems>
      </ds:dataStore>

      <!-- OpenStack DataStore configuration -->
      <ds:dataStore xsi:type="ds:openStackDataStoreConf"
                    name="secondaryDataStore"
```

```
                    restriction="none">
         <!-- Name of the provider -->
         <ds:provider>openstack-swift</ds:provider>
         <!-- Identity is tenantName:Username -->
         <ds:identity>my-identity</ds:identity>
         <!-- Password to access the Openstack storage -->
         <ds:credential>my-credential</ds:credential>
         <!-- Url of the Openstack storage -->
         <ds:url>authentication-service-url</ds:url>
         <!-- Target container on the Openstack storage -->
         <ds:container>container-name</ds:container>
         <!-- Region of the target container on the Openstack storage -->
         <ds:region>RegionOne</ds:region>
      </ds:dataStore>

      <!-- DHuS as DataStore -->
      <ds:dataStore xsi:type="ds:remoteDhusDataStoreConf" name="RemoteDHuS">
         <!-- URL to the OData V1 service endpoint -->
         <ds:serviceUrl>http://REMOTE_DHUS/odata/v1</ds:serviceUrl>
         <!-- Valid creadentials on the remote DHuS -->
         <ds:login>root</ds:login>
         <ds:password>password</ds:password>
      </ds:dataStore>
   </dataStores>
   <scanners></scanners>
   <synchronizers>
     <sync:synchronizer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="sync:smartProductSynchronizer">
            <sync:id>0</sync:id>
            <sync:label>example</sync:label>
            <sync:serviceUrl>_deprecated_</sync:serviceUrl>
            <sync:serviceLogin>_deprecated_</sync:serviceLogin>
            <sync:servicePassword>_deprecated_</sync:servicePassword>
            <sync:schedule>0 0/3 * * * ? *</sync:schedule>
            <sync:active>false</sync:active>
            <sync:created>2018-04-27T09:24:29.467Z</sync:created>
            <sync:modified>2018-05-03T13:24:03.249Z</sync:modified>
            <sync:pageSize>5</sync:pageSize>
            <!-- Timeout -->
            <sync:timeout>300000</sync:timeout>
            <!-- Download attemps (per source) -->
            <sync:attempts>10</sync:attempts>
            <!-- Minimal download bandwidth (per download) before trying to switch source -
->
            <sync:threshold>204800</sync:threshold>
            <!-- Synchronizer source list -->
            <sync:sources>
                <sync:source>
                    <!-- Source id reference -->
                    <sync:sourceId>0</sync:sourceId>
                    <!-- Last creation date synchronizer index -->
                    <sync:lastCreated>2018-04-23T15:59:06.732Z</sync:lastCreated>
                </sync:source>
                <sync:source>
                    <sync:sourceId>1</sync:sourceId>
                    <sync:lastCreated>1970-01-01T00:00:00.000Z</sync:lastCreated>
                </sync:source>
            </sync:sources>
        </sync:synchronizer>

        <!-- Product synchronizer -->
        <!-- retriesForSkippedProducts: number of retries for each skipped product -->
        <!-- timeoutSkippedProducts:  timeout in milliseconds of each request of
synchronization for skipped products-->
```

```
        <sync:synchronizer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="sync:productSynchronizer"
        retriesForSkippedProducts="5" timeoutSkippedProducts="30000"
retryingSourceDelay="30">
            <sync:id>0</sync:id>
            <sync:label>sync</sync:label>
            <sync:schedule>0 */3 * ? * *</sync:schedule>
            <sync:active>false</sync:active>
            <sync:created>2022-05-25T17:11:51.173Z</sync:created>
            <sync:modified>2022-05-25T17:30:53.781Z</sync:modified>
            <sync:pageSize>26</sync:pageSize>
            <sync:copyProduct>true</sync:copyProduct>
            <sync:skipOnError>true</sync:skipOnError>
            <sync:syncOfflineProducts>true</sync:syncOfflineProducts>
            <!-- Used for multisources sync. Schedule to process ranking between sources --
>
            <sync:rankingSchedule>0 */2 * ? * *</sync:rankingSchedule>
            <sync:targetCollection>existingCollectionName</sync:targetCollection>
            <sync:filterParam>startswith(Name,'S2')</sync:filterParam>
            <sync:sources>
                <sync:source>
                    <sync:referenceId>1</sync:referenceId>
                    <sync:lastCreationDate>2022-05-20T06:56:55.315Z</sync:lastCreationDate>
                    <sync:lastDateSourceUsed>2022-06-
27T13:56:55.315Z</sync:lastDateSourceUsed>
                </sync:source>
                <sync:source>
                    <sync:referenceId>3</sync:referenceId>
                    <sync:lastCreationDate>2022-05-20T06:56:45.315Z</sync:lastCreationDate>
                </sync:source>
                <sync:source>
                    <sync:referenceId>2</sync:referenceId>
                    <sync:lastCreationDate>2022-05-20T07:00:35.514Z</sync:lastCreationDate>
                </sync:source>
            </sync:sources>
        </sync:synchronizer>
    </synchronizers>

    <!-- Product sources -->
    <productSources>
        <ps:productSource>
            <ps:id>1</ps:id>
            <ps:url>https://server1.url/odata/v1/Products</ps:url>
            <ps:login>login_user1</ps:login>
            <ps:password>p4ssword1</ps:password>
            <ps:listable>true</ps:listable>
        </ps:productSource>
        <ps:productSource>
            <ps:id>2</ps:id>
            <ps:url>https://server2.url/odata/v1/Products</ps:url>
            <ps:login>login_user2</ps:login>
            <ps:password>p4ssword2</ps:password>
            <ps:listable>true</ps:listable>
        </ps:productSource>
        <ps:productSource>
            <ps:id>0</ps:id>
            <ps:url>https://server3/odata/v1</ps:url>
            <ps:login>login_user3</ps:login>
            <ps:password>p4ssword3</ps:password>
            <ps:remoteIncoming>/my/directory/for/products</ps:remoteIncoming>
            <ps:listable>true</ps:listable>
        </ps:productSource>
    </productSources>
    <evictions>
        <!-- Eviction configurations.
```

```
            name: name of the eviction
            maxEvictedProducts: the maximum number of evicted products when running an
eviction
            keepPeriod: the minimal time between product creation and eviction
            keepPeriodUnit: the time unit for the above value (default is DAYS)
            filter: OData filter used to query products to evict
            orderBy: OData ordering clause used to query products to evict
            targetCollection: name of the collection owning products to evict
            softEviction: boolean value (false : physical product and metadatas are deleted -
true : metadatas are kept)
            status: STOPPED or STARTED, to activate or stop and eviction
         -->
      <evict:eviction name="DefaultEviction" maxEvictedProducts="1000" keepPeriod="10" />
   </evictions>
   <sources>
      <source:source>
         <!-- Source id -->
         <source:id>0</source:id>
         <!-- Source URL -->
         <source:url>http://dhusSource0/odata/v1</source:url>
         <!-- Source authentication username -->
         <source:username>usr_source_0</source:username>
         <!-- Source authentication password -->
         <source:password>pwd_source_0</source:password>
         <!-- Max concurrent downloads -->
         <source:maxDownload>5</source:maxDownload>
      </source:source>
      <source:source>
         <source:id>1</source:id>
         <source:url>http://dhusSource1/odata/v1</source:url>
         <source:username>usr_source_1</source:username>
         <source:password>pwd_source_1</source:password>
         <source:maxDownload>2</source:maxDownload>
      </source:source>
   </sources>
</configuration>
```

Warning: Please note that old synchronizers configuration and sources sections are deprecated, to be replaced by the new Product synchronizer and productsources sections

Appendix B. Example of monitoring.xml-dhus file coming from distribution

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   Data Hub Service (DHuS) - For Space data distribution.
   Copyright (C) 2019 GAEL Systems

   This file is part of DHuS software sources.

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU Affero General Public License as
   published by the Free Software Foundation, either version 3 of the
   License, or (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
   GNU Affero General Public License for more details.

   You should have received a copy of the GNU Affero General Public License
   along with this program. If not, see <http://www.gnu.org/licenses/>.
-->
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:metrics="http://www.ryantenney.com/schema/metrics"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.ryantenney.com/schema/metrics
http://www.ryantenney.com/schema/metrics/metrics.xsd
           http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

    <!-- Creates a MetricRegistry bean, do not change its ID -->
    <metrics:metric-registry id="DHuSMetrics" />

    <!-- Registers BeanPostProcessors with Spring which proxy beans and capture metrics -->
    <metrics:annotation-driven metric-registry="DHuSMetrics" expose-proxy="true" />

    <!-- Reporter definiton. -->

    <!-- Report metrics to an embedded storage, attributes are:
          - period (required) reporting rate
          - filter (optional) filter pattern on metric names
          - filter-ref (optional) ref to a MetricFilter bean
          - duration-unit (optional) unit for durations (defaults to milliseconds)
          - rate-unit (optional) unit for rates (defaults to seconds)
    -->
    <metrics:reporter type="dhus-reporter" metric-registry="DHuSMetrics" period="1m" />
    <!-- Configure embedded metrics storage -->
    <bean id="metrics_storage"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="url" value="jdbc:hsqldb:mem:." />
    </bean>
    <!-- Configure database into a file -->
    <!-- <bean id="metrics_storage"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
             <property name="url" value="jdbc:hsqldb:file:/path/to/store/databaseName"
/><!-- HSQL DB in file -->
       </bean> -->
```

```
    <bean id="jdbi" class="org.jdbi.v3.spring4.JdbiFactoryBean">
        <property name="dataSource" ref="metrics_storage" />
        <property name="autoInstallPlugins" value="true" />
    </bean>

    <!-- Define the service and the period for storing metrics information -->
    <bean id="service" class="org.dhus.metrics.embed.MetricsService">
        <constructor-arg ref="jdbi" />
        <constructor-arg value="15" />
        <constructor-arg value="MINUTES" />
    </bean>
    <!-- Configure embedded metrics storage OData API -->

    <!-- Download event collector, a meter named `download.<itemclass>.meter` ItemClass tag
name precision can be COARSE (default) or FINE -->
    <!--<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Item">
        <property name="itemClassNamePrecision" value="FINE" />
    </bean>-->
    <!-- To record both the item class and the username, a meter named
`download.<username>.<itemclass>.meter` ItemClass tag name precision can be COARSE
(default) or FINE -->
    <!--<bean name="download_collector"
class="org.dhus.metrics.DownloadMetrics.ItemAndUser">
        <property name="itemClassNamePrecision" value="FINE" />
    </bean>-->
    <!-- To record both the item class and the username, a meter named
`download.<username>.meter` -->
    <!--<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.User" />-->
    <!-- To record both the item class and the username, a meter named `download.meter` -->
    <bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Glob" />

    <!-- Register metric beans (Optional) -->
    <!-- The metrics in this example require artifact metrics-jvm -->
    <!--
    <metrics:register metric-registry="DHuSMetrics">
        <bean metrics:name="jvm.gc"
class="com.codahale.metrics.jvm.GarbageCollectorMetricSet" />
        <bean metrics:name="jvm.memory"
class="com.codahale.metrics.jvm.MemoryUsageGaugeSet" />
        <bean metrics:name="jvm.thread-states"
class="com.codahale.metrics.jvm.ThreadStatesGaugeSet" />
        <bean metrics:name="jvm.fd.usage"
class="com.codahale.metrics.jvm.FileDescriptorRatioGauge" />
    </metrics:register>
    -->
</beans>
```

Warning: Please note that at line 52, there is an error that causes the dhus to start with an error. That part should be changed by removing `<!-- HSQL DB in file -->`

Appendix C. Example of monitoring.xml-influx file coming from distribution

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   Data Hub Service (DHuS) - For Space data distribution.
   Copyright (C) 2019 GAEL Systems

   This file is part of DHuS software sources.

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU Affero General Public License as
   published by the Free Software Foundation, either version 3 of the
   License, or (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
   GNU Affero General Public License for more details.

   You should have received a copy of the GNU Affero General Public License
   along with this program. If not, see <http://www.gnu.org/licenses/>.
-->
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:metrics="http://www.ryantenney.com/schema/metrics"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.ryantenney.com/schema/metrics
http://www.ryantenney.com/schema/metrics/metrics.xsd
           http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

    <!-- Creates a MetricRegistry bean, do not change its ID -->
    <metrics:metric-registry id="DHuSMetrics" />

    <!-- Registers BeanPostProcessors with Spring which proxy beans and capture metrics -->
    <metrics:annotation-driven metric-registry="DHuSMetrics" expose-proxy="true" />

    <!-- Reporter definiton. -->

    <!-- Poll metrics through JMX (not for operational scenario), attributes are
          - domain (optional)
          - mbean-server-ref (optional)
          - duration-unit (optional)
          - rate-unit (optional)
    -->
    <!--<metrics:reporter type="jmx" metric-registry="DHuSMetrics" />-->

    <!-- Dump metrics in CSV files, attributes are
          - period (required) reporting rate
          - directory (required) path to output directory
          - filter (optional) filter pattern on metric names
          - filter-ref (optional) ref to a MetricFilter bean
          - duration-unit (optional) unit for durations (defaults to milliseconds)
          - rate-unit (optional) unit for rates (defaults to seconds)
          - locale (optional) used to format numbers (defaults to en_US)
    -->
```

```
    <!--<metrics:reporter type="csv" metric-registry="DHuSMetrics" period="5m"
directory="/var/metrics/" />-->

    <!-- Report metrics to an InfluxDB database, attributes are:
          - period (required) reporting rate
          - database (required) name of the database, it must be created
          - layer (required) connection layer to InfluxDB, either "http" or "tcp"
          - host (optional) of an InfluxDB instance
          - port (optional) to connect to an InfluxDB instance
          - filter (optional) filter pattern on metric names
          - filter-ref (optional) ref to a MetricFilter bean
          - duration-unit (optional) unit for durations (defaults to milliseconds)
          - rate-unit (optional) unit for rates (defaults to seconds)
          - group-counters (optional) "true" or "false" (default)
          - group-gauges (optional) "true" or "false" (default)
          - tag-provider (optional) ref to bean that provides base tags
          - template-provider (optional) ref to bean that provides templates
    -->
    <metrics:reporter type="influxdb" metric-registry="DHuSMetrics" period="1m"
                      database="DHuSMetrics" layer="http"
                      group-counters="true" group-gauges="true"
                      tag-provider="tag_provider" template-provider="template_provider" />
    <!-- Tag Provider bean used by the influxdb reporter
         Base tags to attach to every metrics (measurements)
    -->
    <bean name="tag_provider" class="eu.metrics.spring.TagProviderBean">
        <constructor-arg>
            <map>
                <!-- Each entry's key is the column name and value is the row value for
column `key` -->
                <entry key="dhus_inst" value="frenchub_fe_01" />
            </map>
        </constructor-arg>
    </bean>

    <!-- Configure the service for OData access -->
    <bean id="service" class="org.dhus.metrics.external.influx.InfluxMetricService">
        <constructor-arg value="http://<influx_server>:<port>" /> <!-- url of the influxDB-
->
        <constructor-arg value="<user>"/> <!-- username on InfluxDB -->
        <constructor-arg value="<password>"/> <!-- password on influx DB -->
        <constructor-arg value="<influx_database_name>"/> <!-- database name on InfluxDB --
>
    </bean>

    <!-- A default template provider bean is already defined in the dhus-core, there is no
need to
         override that bean unless to work around a bug.
         See file dhus-core-monitoring.xml
    <bean name="template_provider"
class="eu.metrics.spring.TemplateProviderBean">...</bean>-->

    <!-- Download event collector, a meter named `download.<itemclass>.meter` ItemClass tag
name precision can be COARSE (default) or FINE -->
    <!--<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Item">
        <property name="itemClassNamePrecision" value="FINE" />
    </bean>-->
    <!-- To record both the item class and the username, a meter named
`download.<username>.<itemclass>.meter` ItemClass tag name precision can be COARSE
(default) or FINE -->
    <bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.ItemAndUser">
        <property name="itemClassNamePrecision" value="COARSE" />
    </bean>
    <!-- To record both the item class and the username, a meter named
`download.<username>.meter` -->
```

```
    <!--<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.User" />-->
    <!-- To record both the item class and the username, a meter named `download.meter` -->
    <!--<bean name="download_collector" class="org.dhus.metrics.DownloadMetrics.Glob" />-->

    <!-- Monitor the performance of the various caches used in the DHuS, for debugging
purposes only -->
    <!--
    <bean id="cacheManager" class="org.dhus.metrics.InstrumentedEhCacheCacheManager">
        <constructor-arg ref="ehcache" />
    </bean>-->

    <!-- Register metric beans (Optional) -->
    <!-- The metrics in this example require artifact metrics-jvm -->
    <!--
    <metrics:register metric-registry="DHuSMetrics">
        <bean metrics:name="jvm.gc"
class="com.codahale.metrics.jvm.GarbageCollectorMetricSet" />
        <bean metrics:name="jvm.memory"
class="com.codahale.metrics.jvm.MemoryUsageGaugeSet" />
        <bean metrics:name="jvm.thread-states"
class="com.codahale.metrics.jvm.ThreadStatesGaugeSet" />
        <bean metrics:name="jvm.fd.usage"
class="com.codahale.metrics.jvm.FileDescriptorRatioGauge" />
    </metrics:register>
    -->
</beans>
```

<END OF DOCUMENT>