

Use Case Analysis

TVUK System Team

Version v0.3, 18/03/2019

Table of Contents

1. Introduction	2
1.1. Purpose and Scope	2
1.2. Structure of the Document	2
1.3. Reference Documents	2
1.4. Terminology	2
1.5. Glossary	4
2. Use Cases	6
2.1. User accesses Platform services	6
2.2. Consumer discovers and visualises products	7
2.3. Consumer uploads data to their workspace	10
2.4. Consumer discovers and executes On-demand Processing Service	12
2.5. Consumer discovers and executes Interactive Applications	15
2.6. Consumer analyses value-added product	18
2.7. Consumer executes Bulk processing	19
2.8. Consumer executes Systematic processing	21
2.9. Consumer performs Open Science	24
2.10. Consumer accesses EP services with External Application	26
2.11. Expert user builds new processing service	28
2.12. Expert user builds new processing services chains	30
2.13. Expert user builds new interactive application	32
2.14. Expert user builds new value-added products	34
2.15. Expert develops with interactive development environment	36
2.16. Expert performs training	38
3. Functional Mapping	40
3.1. User accesses Platform services	40

Technical Note**Use Case Analysis for the Common Architecture****COMMENTS and ISSUES**

If you would like to raise comments or issues on this document, please do so by raising an Issue at the following URL <https://github.com/EOEPCA/use-case-analysis/issues>.

PDF

This document is available in PDF format [here](#).

COPYRIGHT

Copyright © 2019 Telespazio VEGA UK Ltd.

LICENSE AGREEMENT

TBD.

1. Introduction

1.1. Purpose and Scope

This document presents the use cases relevant to the Common Architecture.

1.2. Structure of the Document

TBD

1.3. Reference Documents

The following is a list of Reference Documents with a direct bearing on the content of this document.

[EP-UC]

EO Exploitation Platforms - Use Cases, ESA-EOPSDP-TN-17-049, Issue 1.0, 15/12/2017

[EP-FM]

Exploitation Platform - Functional Model, ESA-EOPSDP-TN-17-050, Issue 1.0, 30/11/2017

1.4. Terminology

The following terms are used in the use case descriptions.

Term	Meaning
Platform (EP)	An on-line collection of products, services and tools for exploitation of EO data
User	An individual using the EP, of any type (Admin/Consumer/Expert/Guest)
Admin	User with administrative capability on the EP
Consumer	User accessing existing services/products within the EP. Consumers may be scientific/research or commercial, and may or may not be experts of the domain
Expert	User developing and integrating added-value to the EP (Scientific Researcher or Service Developer)
Scientific Researcher	Expert user with the objective to perform scientific research. Having minimal IT knowledge with no desire to acquire it, they want the effort for the translation of their algorithm into a service/product to be minimised by the platform.
Service Developer	Expert user with the objective to provide a performing, stable and reliable service/product. Having deeper IT knowledge or a willingness to acquire it, they require deeper access to the platform IT functionalities for optimisation of their algorithm.

Term	Meaning
Guest	An unregistered User or an unauthenticated Consumer with limited access to the EP's services
Discovery	User finds products/services of interest to them based upon search criteria.
Products	EO data (commercial and non-commercial) and Value-added products and made available through the EP. <i>It is assumed that the Hosting Environment for the EP makes available an existing supply of EO Data</i>
Value-added products	Products generated from processing services of the EP (or external processing) and made available through the EP. This includes products uploaded to the EP by users and published for collaborative consumption
Processing Service	A non-interactive data processing that has a well-defined set of input data types, input parameterisation, producing <i>Processing Results</i> with a well-defined output data type.
On-demand Processing Service	A <i>Processing Service</i> whose execution is initiated directly by the user on an ad-hoc basis.
Systematic Processing Service	A <i>Processing Service</i> whose execution is initiated automatically (on behalf of a user), either according to a schedule (routine) or triggered by an event (e.g. arrival of new data).
Bulk Processing	Execution of a <i>Processing Service</i> on large amounts of data specified by AOI and TOI.
Processing Result	The <i>Products</i> produced as output of a <i>Processing Service</i> execution.
Reusable Research Object	An encapsulation of some research/analysis that describes all aspects required to reproduce the analysis, including data used, processing performed etc.
Interactive Application	A stand-alone application provided within the exploitation platform for on-line hosted processing. Provides an interactive interface through which the user is able to conduct their analysis of the data, producing <i>Analysis Results</i> as output. Interactive Applications include at least the following types: console application, web application (rich browser interface), remote desktop to a hosted VM.
Interactive Console Application	A simple <i>Interactive Application</i> for analysis in which a console interface to a platform-hosted terminal is provided to the user. The console interface can be provided through the user's browser session or through a remote SSH connection.
Interactive Web Application	An Interactive Application for analysis provided as a rich user interface through the user's web browser.
Interactive Remote Desktop	An Interactive Application for analysis provided as a remote desktop session to an OS-session (or directly to a 'native' application) on the exploitation platform. The user will have access to a number of applications within the hosted OS. The remote desktop session is provided through the user's web browser.
External Application	An application or script that is developed and executed outside of the Exploitation Platform, but is able to use the data/services of the EP via a programmatic interface (API).

Term	Meaning
Analysis Result	The <i>Products</i> produced as output of an <i>Interactive Application</i> analysis session.
Execution	The act to start a <i>Processing Service</i> or an <i>Interactive Application</i> .
Visualisation	To obtain a visual representation of any data/products held within the platform - presented to the user within their web browser session.
Processing	A set of pre-defined activities that interact to achieve a result. For the exploitation platform, comprises on-line processing to derive data products from input data, conducted by a hosted processing service execution.
Algorithm	A self-contained set of operations to be performed, typically to achieve a desired data manipulation. The algorithm must be implemented (codified) for deployment and execution on the platform.
Code	The codification of an algorithm performed with a given programming language - compiled to Software or directly executed (interpreted) within the platform.
Software	The compilation of code into a binary program to be executed within the platform on-line computing environment.
Analytics	A set of activities aimed to discover, interpret and communicate meaningful patterns within the data. Analytics considered here are performed manually (or in a semi-automatic way) on-line with the aid of <i>Interactive Applications</i> .
Development	The act of building new products/services/applications to be exposed within the platform and made available for users to conduct exploitation activities. Development may be performed inside or outside of the platform. If performed outside, an integration activity will be required to accommodate the developed service so that it is exposed within the platform.
Workspace	A user-scoped 'container' in the EP, in which each user maintains their own links to resources (products and services) that have been collected by a user during their usage of the EP. The workspace acts as the hub for a user's exploitation activities within the EP
Terms & Conditions (T&Cs)	The obligations that the user agrees to abide by in regard of usage of products/services of the platform. T&Cs are set by the provider of each product/service.

1.5. Glossary

The following acronyms and abbreviations have been used in this report.

Term	Definition
AOI	Area of Interest
API	Application Programming Interface
EO	Earth Observation
EP	Exploitation Platform

Term	Definition
SSH	Secure Shell
TOI	Time of Interest
VNC	Virtual Network Computing

2. Use Cases

Each use case is presented in the following sub-sections.

See [\[EP-UC\]](#) and [\[EP-FM\]](#) for more information.

2.1. User accesses Platform services

As a User, I want to be able to access the services of the system within the context of my user profile. I want to identify myself with the same unique identifier that I use on other platforms (e.g. email address or GitHub ID etc.). I should be directed to register or login if that is required to access the requested resource.

1. Admin logs in on the EP
 2. Admin configures the access policy of the EP services
 3. Guest successfully accesses unprotected services
 4. Guest is denied access to protected services
 5. Guest registers as a platform user with an existing (external) identity
 6. Unauthenticated Consumer/Expert is denied access to protected services
 7. Consumer/Expert logs in on the EP
 8. Consumer/Expert successfully accesses protected services
-

2.2. Consumer discovers and visualises products

As a Consumer I want to search for Products of all types by specification of textual and faceted search criteria. I want to be able to incrementally narrow my search with more criteria clauses. For each Product in the search result I want to get more detailed information that describes the product, its usage terms and costs, and visualise the data graphically. Having discovered a product of interest I may 'save' it to my Workspace, download it, or use it as input to further exploitation activities.

Figure 1 illustrates discovery of EO data products using the EO Browser system.

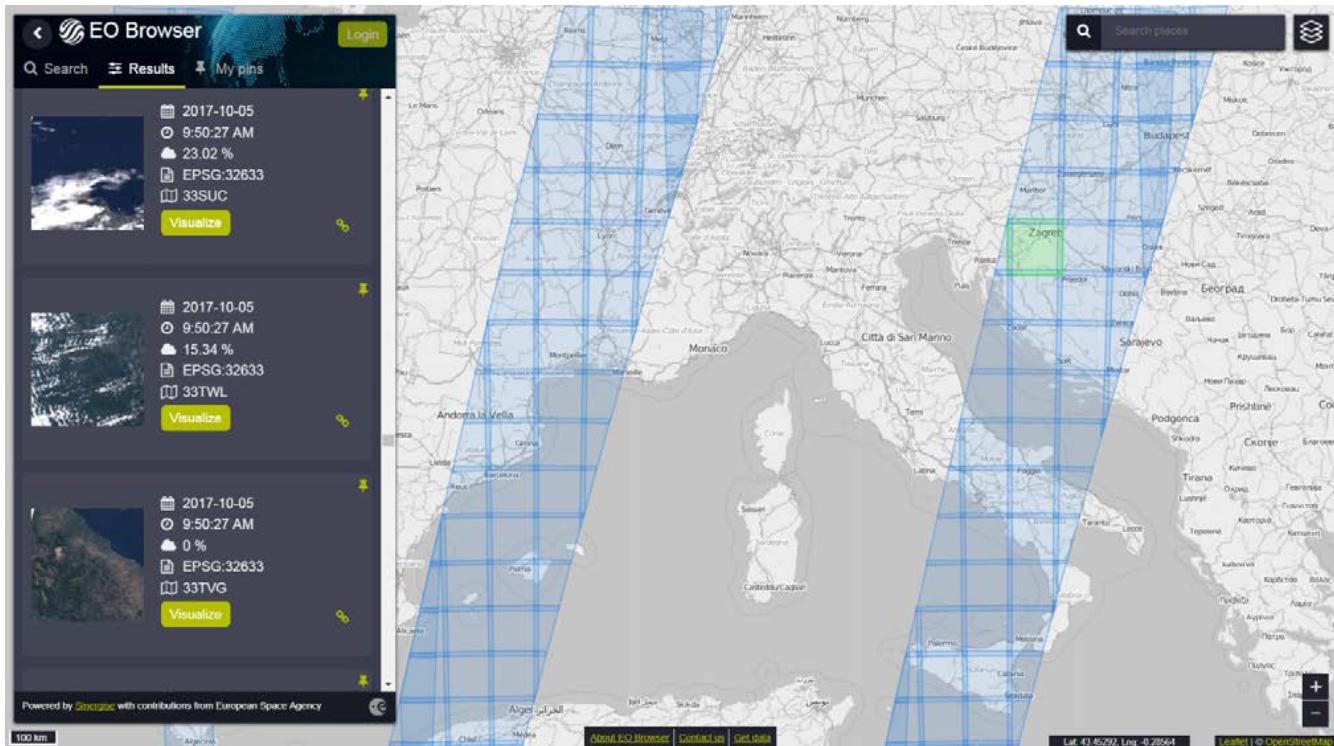


Figure 1. Product Discovery - EO Browser (<http://apps.eocloud.sentinel-hub.com/eo-browser/>)

Figure 2 shows custom visualisation of a Sentinel-2 EO data product.

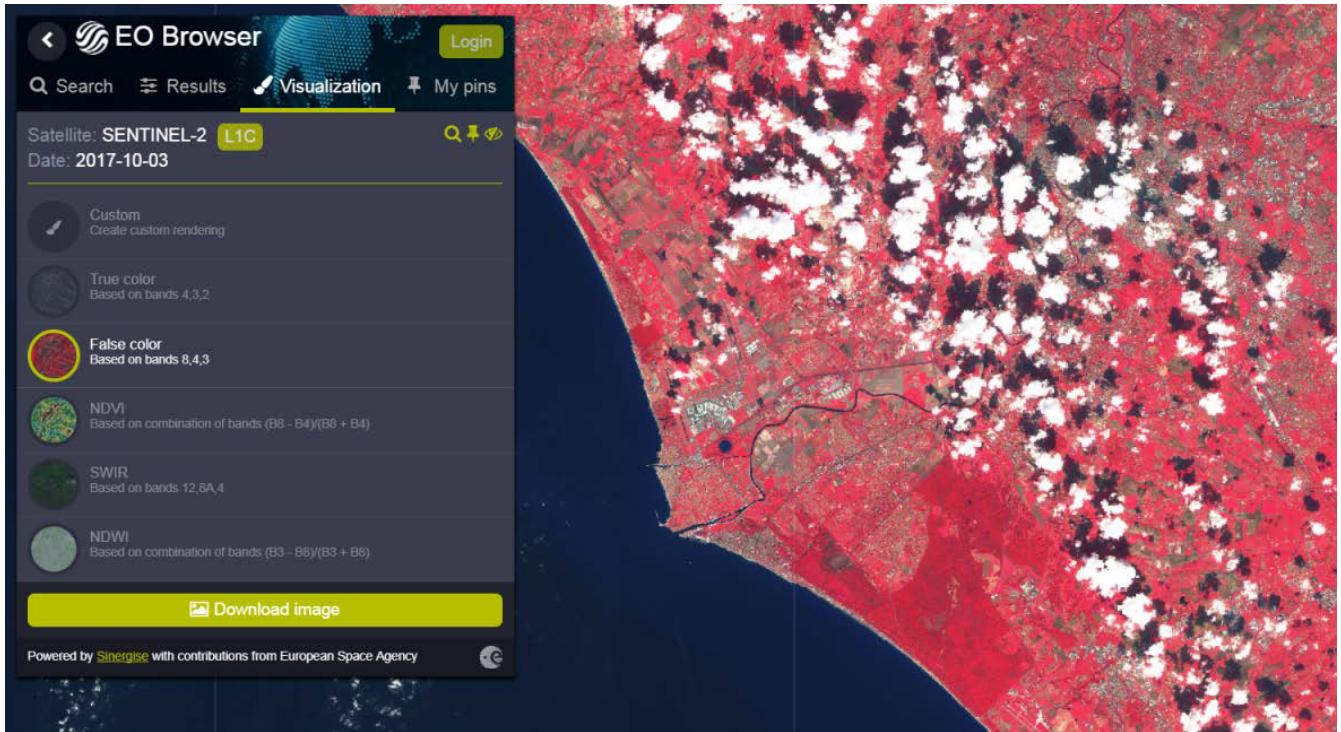


Figure 2. Product Visualisation - EO Browser (<http://apps.eocloud.sentinel-hub.com/eo-browser/>)

1. Consumer logs in on the EP
2. Consumer begins searching for products on the EP. The starting point is the full unfiltered product set that contains commercial/non-commercial EO data and value-added products
3. Optionally, the resultset is automatically filtered to include only those collections that the Consumer has right to visualise. *It may be the case that the EP chooses to make these 'unavailable collections' visible to the Consumer to publicise their existence*
4. Consumer filters the resultset by any combination of textual search terms, area of interest, time of interest, selection of collection(s) and selection of product facets
5. Consumer incrementally adjusts their search criteria to refine the filtered resultset
6. Consumer selects a product of interest; the EP checks they are authorised to access the product
7. Consumer views detailed metadata for the selected product
8. Consumer views T&Cs for the service and accepts terms if not already done so
9. Optionally, the Consumer 'saves' (a reference to) the product to their workspace
10. Consumer requests the cost for access/usage of the data
11. Optionally, the Consumer visualises the product with a graphical representation. *The Consumer is able to parameterise the visualisation interactively, e.g. specification of layers to view etc.*
12. Optionally, the Consumer downloads the visualisation results

Notes

	<p><i>Guest User</i></p> <p>The use case does not mention the Guest user. It should be taken into account that an EP may choose to allow limited access to a Guest user, such as searching/browsing catalogues, before insisting on user login for exploitation activities. <i>This approach might be typical in order to allow a potential user discover EP capabilities before signing up</i></p>
NOTE	<p><i>Data Sub-setting</i></p> <p>The use case does not consider the case where the Consumer is able to select sub-setting (spatial/temporal) of the data product itself, which might be typical of data accessible through a service such as OPeNDAP. <i>The extent to which this case should be considered for the EP is TBD</i></p>

2.3. Consumer uploads data to their workspace

As a Consumer I want to upload data into my workspace and specify the name and destination in the target EP. I would like to treat my uploaded data the same as other (existing) data available in the EP. This includes:

- Visualise the data in the platform
- Use the data as input to analysis activities, e.g. interactively or through processing services

Additionally, I would like to (optionally) publish the data so that it is available to other users, discoverable through the EP catalogue.

The following figures provide implementation examples from the Hydrology TEP, <https://hydrology-tep.eu/>.

Figure 3 demonstrates the invocation of the 'Manage My Data' application.

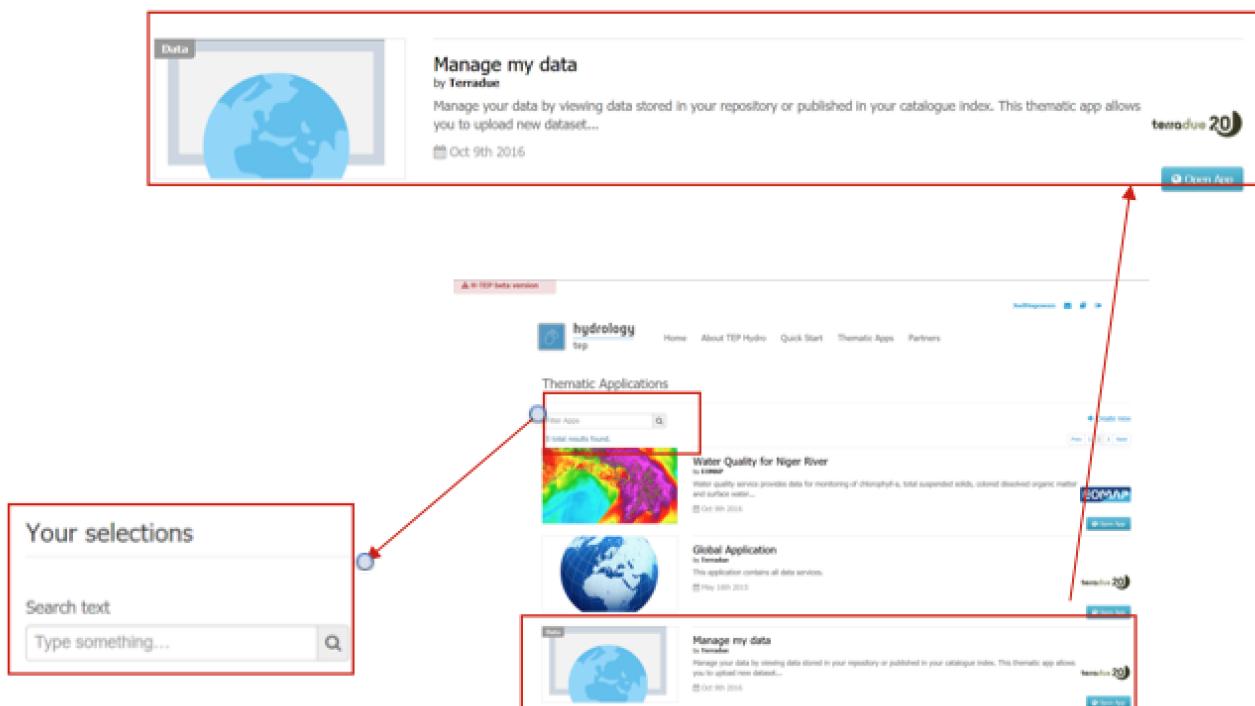


Figure 3. Manage My Data

Figure 4 illustrates the invocation of the data upload capability.



Figure 4. Workspace Upload

1. Consumer logs in on the EP
 2. Consumer navigates to their Workspace to be presented with data they have previously uploaded or 'saved'
 3. Consumer requests to upload new data - specifying appropriate metadata including the location to which the data should be uploaded
 4. Platform confirms the user's rights before accepting the request and ingesting the data into the user's workspace
 5. Optionally, consumer can view the details of any data in their workspace, including that which they have uploaded
 6. Optionally, consumer visualises their uploaded data and is able to manipulate and parameterise the view - with the possibility to download the result of their visualisation
 7. Optionally, consumer publishes the uploaded data in the catalogue - specifying all necessary metadata to support discovery
-

2.4. Consumer discovers and executes On-demand Processing Service

As a Consumer I want to list and search for commercial and non-commercial *Processing Services*, by specification of textual and faceted search criteria. I want to be able to incrementally narrow my search with more criteria clauses. For each *Processing Service* in the search result I want to get more detailed information, including a description and access to its execution manual. If I have not already done so, then any *Terms & Conditions* associated to usage of the processing service should be made clear to me, with the opportunity for me to accept the T&Cs before I am authorised to execute the processing service.

I want to prepare the *Processing Service* for execution by specifying input data from within and outside the platform, and define any other parameters and configuration required to fully specify the job. Based upon my inputs I would like to get an estimation of the execution costs and time, before (optionally) initiating execution of the processing service. Having discovered a processing service of interest I may 'save' it to my Workspace, (which acts as a bookmark) - either with or without my parameterisation.

Having initiated execution of a *Processing Service*, I would like to monitor its progress (including any errors occurring), and have access to the results output at the successful conclusion. I would like to visualise my Processing Results graphically with custom visualisation parameters, and possibly download the results. Additionally, I would like to (optionally) publish the Processing Results so that it is available to other users, discoverable through the EP catalogue.

Figure 5 illustrates discovery and parameterisation of a Sentinel-2 Land Cover processor, from the Forestry TEP platform.

The screenshot shows the Forestry TEP platform interface. At the top, there is a navigation bar with links for 'forestry', 'Explorer', 'Manage / Share', 'Helpdesk', 'My Account', and 'Logout'. Below the navigation bar, the main area is divided into sections: 'Default Project' (with a '+' button), 'WORKSPACE' (containing 'LandCoverS2'), and a central map of Europe. The map displays a study area over the United Kingdom and parts of France, Germany, and Belgium, with various colored regions indicating land cover. On the left, under 'LandCoverS2', there are several input fields and options:

- 'Calculate land coverage (a forest map) from Sentinel-2 data'
- 'Input data *': A dropdown menu showing 'sentinel2:///52A_MSIL1C_20170718T105651_N0205_R094_T30UYB' and a 'Remove' button.
- 'Reference data archive containing shapefile for image classification training *': A dropdown menu showing 'YB_20170718T10452' and a 'Remove' button.
- 'Expected file type': A note stating 'Expected is a .zip with flat content structure containing one *.shp and associated files.'
- 'String identifying the field in the shapefile to be used for training the model *': A dropdown menu showing 'Class' and a 'Remove' button.
- 'Target CRS identifier *': A dropdown menu showing 'EPSG:326XX/327XX' and a 'Remove' button.
- 'EPSG coordinate reference system identifier of the target product, i.e.: EPSG:326XX/327XX for the Northern/Southern hemisphere, where XX is the zone, e.g. EPSG:32635'
- 'Area of interest *': A dropdown menu showing 'London' and a 'Remove' button.

At the bottom of the workspace section, there is a 'RESULTS: SENTINEL2' table with four entries:

	S2A_MSIL1C_20170718T105651_N0205_R094_T30UYB_20170718T110452	Cloud Coverage: 6.53
=	Start: 2017-07-18T10:56:51.026 End: 2017-07-18T10:56:51.026 Size: 0.69 GB	
=	S2A_MSIL1C_20170718T105651_N0205_R094_T31UDS_20170718T110452	
=	Start: 2017-07-18T10:56:51.026 End: 2017-07-18T10:56:51.026 Size: 0.74 GB	
=	S2A_MSIL1C_20170718T105651_N0205_R094_T30UYA_20170718T110452	
=	Start: 2017-07-18T10:56:51.026 End: 2017-07-18T10:56:51.026 Size: 0.64 GB	
=	S2A_MSIL1C_20170718T105651_N0205_R094_T30UYU_20170718T110452	
=	Start: 2017-07-18T10:56:51.026 End: 2017-07-18T10:56:51.026 Size: 0.79 GB	

Figure 5. Processing Discovery - Sentinel-2 Land Cover processor (<https://f-tep.com/app/>)

Figure 6 illustrates the visualisation of processing results, from the Forestry TEP platform.

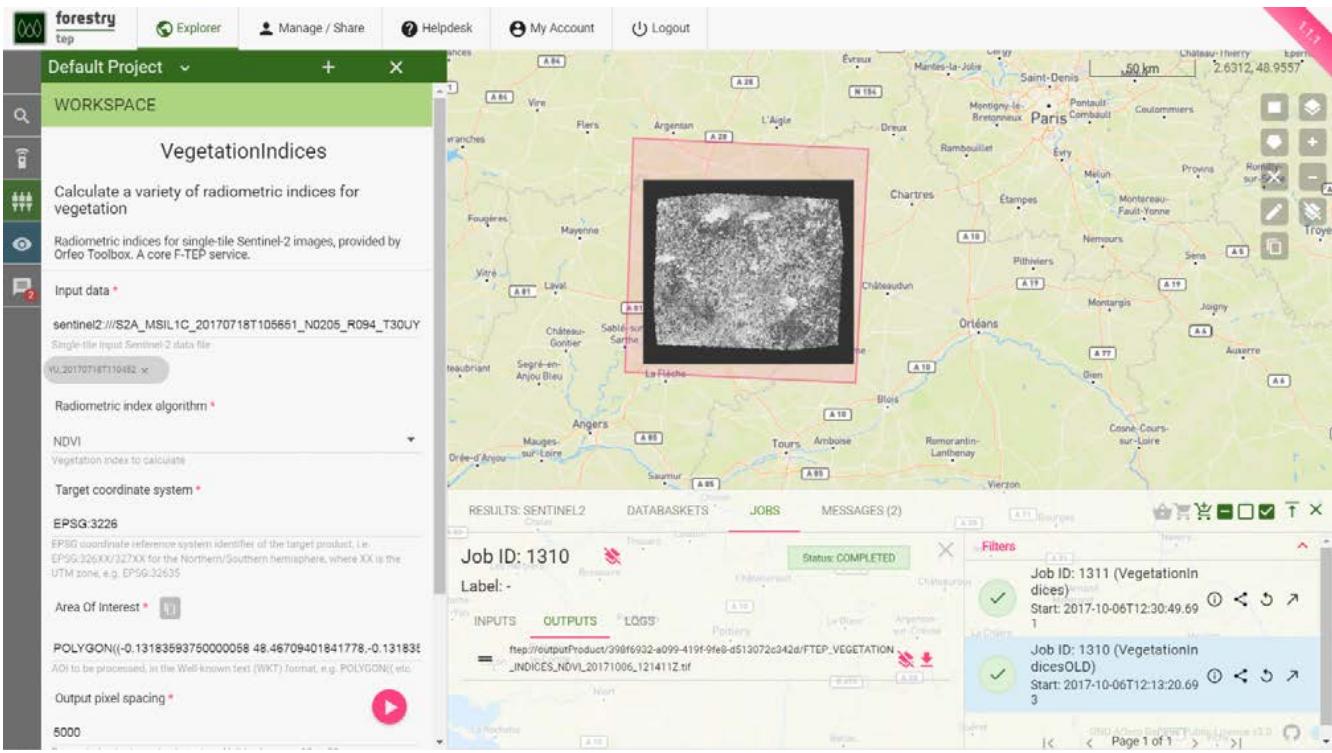


Figure 6. Processing Results (<https://f-tep.com/app/>)

1. Consumer logs in on the EP
- 2. Discover and Select Processing Service...**
3. Consumer begins searching for *Processing Services* on the EP. The starting point is the full unfiltered set of *Processing Services* that contains commercial/non-commercial services
4. Optionally, the resultset is automatically filtered to include only those services that the Consumer has right to visualise. *It may be the case that the EP chooses to make these 'unavailable services' visible to the Consumer to publicise their existence*
5. Consumer filters the resultset by any combination of textual search terms and selection of service facets
6. Consumer incrementally adjusts their search criteria to refine the filtered resultset
7. Consumer selects a *Processing Service* of interest; the EP checks they are authorised to access the product
8. Consumer views the manual for the selected service in order to understand its required input data/parameters and the nature of its algorithm
9. Consumer views T&Cs for the service and accepts terms if not already done so
10. Optionally, the Consumer 'saves' (a reference to) the product to their workspace
- 11. Discover and Select Input Data...**
12. Consumer searches the EP catalogue for input data of interest, by specification of spatial/temporal (and other) characteristics
13. Platforms aids the Consumer in selecting input data that is compatible with the chosen processing service

14. Consumer selects the input data from their search results and/or from their workspace data
15. The EP checks they are authorised to access the product
16. Consumer views detailed metadata for the selected product
17. Consumer views T&Cs for the service and accepts terms if not already done so
18. **Initiate Processing...**
19. Consumer specifies the input parameters of the *Processing Service*
20. Consumer requests processing execution
21. The EP checks that the Consumer has the authorisation to launch the *Processing Service* and access the specified data
22. The EP estimates the cost and duration of the processing and checks the Consumer has enough resources to execute the processing
23. Consumer is presented with the cost/duration estimation and confirms the processing
24. Consumer monitors the status of the processing (%completion, execution logs)
25. When the processing completes successfully the *Processing Results* are made available to the user in their *Workspace*
26. The Consumer's billing account is updated commensurate with the 'cost' of the processing
27. **Exploit Results...**
28. Optionally, the Consumer downloads the results
29. Optionally, the Consumer visualises the processing logs (e.g. for error inspection)
30. Optionally, the Consumer visualises the results and is able to manipulate and parameterise the view - with the possibility to download the result of their visualisation
31. Optionally, the Consumer publishes their results in the catalogue - specifying all necessary metadata to support discovery

Notes

Data/Processor Selection Order

NOTE

The use case considers the user interaction in which the processor is selected first, followed by selection of compatible data. We might also consider the alternative in which the input data is selected first, and the Platform facilitates the selection of compatible processors. Ideally the platform should support both approaches.

Resource Quotas

NOTE

The use case does not explore how the user obtains/maintains a resource quota in the platform in order to 'pay' for their usage

Processor License Key

NOTE

The use case does not consider the possibility of processing services for which the user requires a license key. This would have to be considered as an extension of this case.

2.5. Consumer discovers and executes Interactive Applications

As a Consumer I want to list and search for commercial and non-commercial *Interactive Applications*, by specification of textual and faceted search criteria. I want to be able to incrementally narrow my search with more criteria clauses. For each *Interactive Application* in the search result I want to get more detailed information, including a description and access to its execution manual. If I have not already done so, then any *Terms & Conditions* associated to usage of the *Interactive Application* should be made clear to me, with the opportunity for me to accept the T&Cs before I am authorised to execute the application.

Having started the *Interactive Application* I want to access the application user interface via my web browser, perform analytics accessing EO data, value-added products and other resources available on or outside the platform. I would like to export my Analysis Results (text, products, screenshots etc.) from the application into my EP workspace, allowing me to visualise and download the results. At the conclusion of my analysis session I would like to stop the application to ensure that I am not using unnecessary resources and/or incurring unwanted costs.

As a Consumer, I should be able to request a list of Interactive Application sessions that I currently have 'live' in the system - through which I can manage the sessions by rejoining or stopping as required.

Figure 7 illustrates listing and selection of an Interactive Application, from the Costal TEP platform.

The screenshot shows a web-based interface titled 'Service'. It lists four interactive applications:

Application	Description	Actions
SNAP	Sentinel Toolbox	<button>Start new</button> Running
Ubuntu	Virtual Desktop	<button>Start new</button>
QGIS	Free and Open Source GIS	<button>Start new</button>

The 'SNAP' row is highlighted, indicating it is selected. The 'Running' status is shown next to the 'Start new' button for SNAP.

Figure 7. Interactive Application - Discovery and selection (<https://www.coastal-tep.eu/>)

Figure 8 illustrates the SNAP application running interactively in the user's web browser, from the Costal TEP platform.

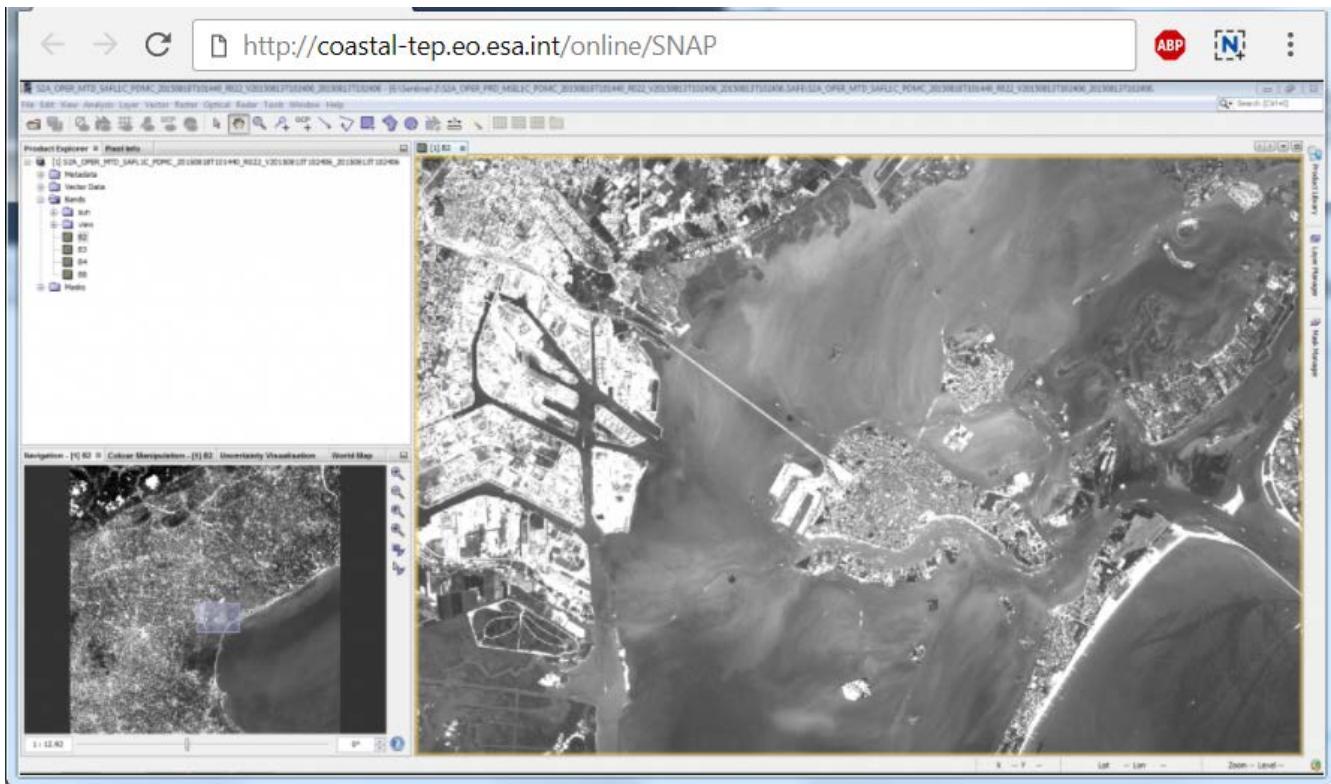


Figure 8. Interactive Application - SNAP running in web browser (<https://www.coastal-tep.eu/>)

1. Consumer logs in on the EP
2. Consumer begins searching for *Interactive Applications* on the EP. The starting point is the full unfiltered set of *Interactive Applications* that contains commercial/non-commercial applications
3. Optionally, the resultset is automatically filtered to include only those services that the Consumer has right to visualise. *It may be the case that the EP chooses to make these 'unavailable applications' visible to the Consumer to publicise their existence*
4. Consumer filters the resultset by any combination of textual search terms and selection of application facets
5. Consumer incrementally adjusts their search criteria to refine the filtered resultset
6. ***Alternative Flow: Existing Session Selection***
7. Consumer views the description and execution manual for the selected application
8. Consumer selects an *Interactive Application* of interest; the EP checks they are authorised to access the application
9. Consumer views T&Cs for the application and accepts terms if not already done so
10. Optionally, the Consumer 'saves' (a reference to) the application to their workspace
11. Consumer searches the EP catalogue for input data of interest, by specification of spatial/temporal (and other) characteristics
12. Consumer selects the input data from their search results and/or from their workspace data
13. ***Alternative Flow: Input Data Selection***
14. EP checks that the Consumer is authorised to access the data and has accepted the associated T&Cs - prompting for confirmation of acceptance as required

15. If applicable, the EP estimates the cost for the application/data, checks the Consumer has enough resources to cover this cost, and the Consumer is presented with the cost indication
16. Consumer accepts and starts the Interactive Application, which is presented to them in their web browser
17. The previously selected input data is made available within the execution context of the *Interactive Application*
18. Consumer uses the application to perform analytics using the input data
19. Optionally, the Consumer accesses additional data to be introduced into the analysis within the *Interactive Application*, (the EP checks rights/costs as required)
20. Consumer saves(exports their Analysis Results from the application to be saved on the EP, within their *Workspace*
21. The Consumer's billing account is updated commensurate with the 'cost' of the processing
22. Optionally, the Consumer downloads the results
23. Optionally, the Consumer visualises the results and is able to manipulate and parameterise the view - with the possibility to download the result of their visualisation
24. Optionally, the Consumer publishes their results in the catalogue - specifying all necessary metadata to support discovery

Alternative Flow: Existing Session Selection

Rather than discovering and starting a new application instance, instead they list existing 'live' sessions, that they can rejoin

- a. Consumer obtains a list of their current 'live' interactive sessions
- b. Consumer selects to rejoin an existing session

Alternative Flow: Input Data Selection

Rather than pre-selecting the input data before invoking the *Interactive Application*, it may be preferable (depending on the application) to make the input data selection from within the *Interactive Application*. The use case is not elaborated here in regard of this approach, but it should be ensured that the data access is made within the context of the user's access rights and associated billing considerations.

2.6. Consumer analyses value-added product

This use-case is described in document [EP-UC]. However, it is seen as the same as use-case [Consumer discovers and executes Interactive Applications](#), just with particular emphasis on the inclusion of Value-added Products in the analysis, and perhaps with an associated set of specific tooling in mind (WebGIS, DataCube, etc.).

By way of illustration, [Figure 9](#) provides an example that uses PUMA WebGIS to combine value-added products with in-situ data, taken from the Urban TEP platform.

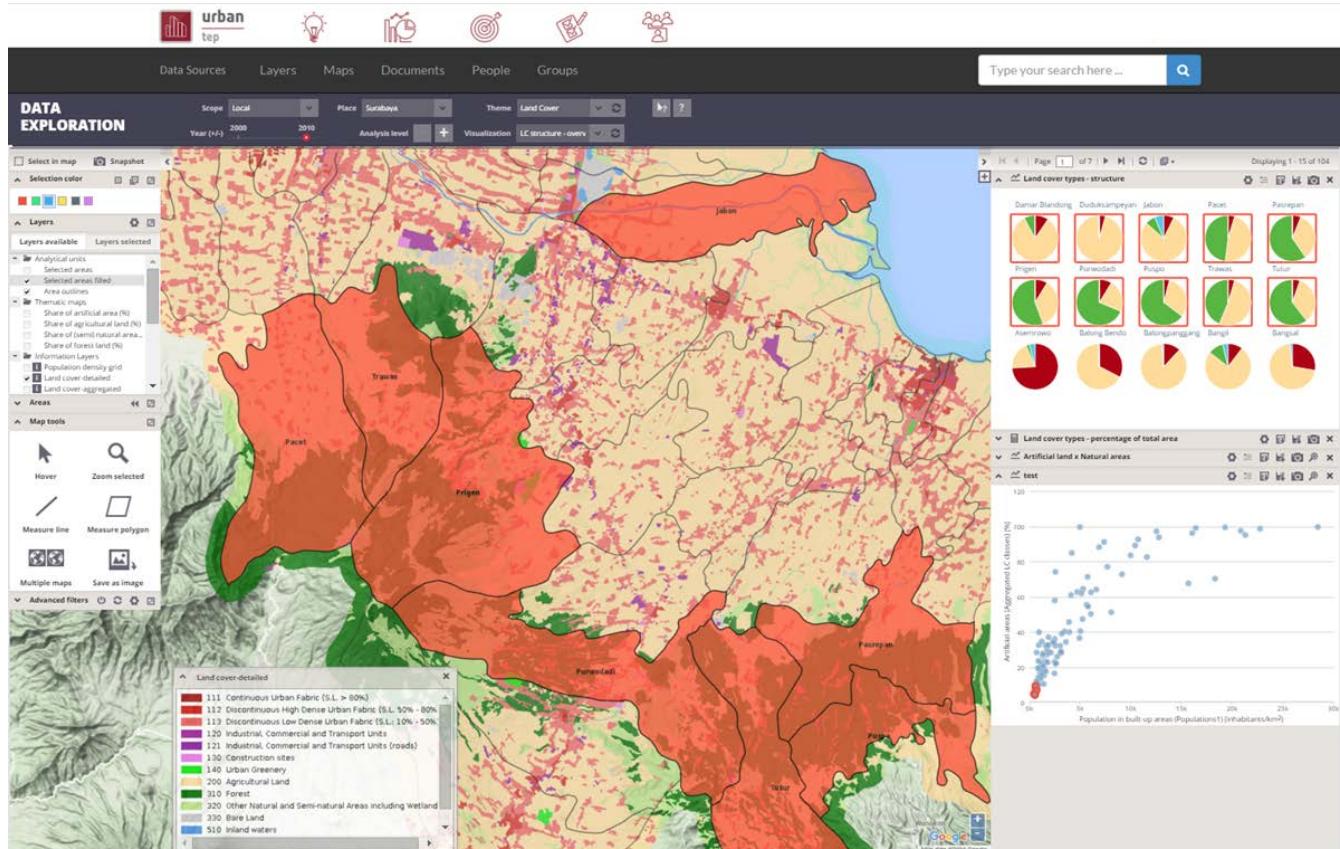


Figure 9. Value Added Products Analyses - PUMA WebGIS (<https://urban-tep.eu/>)

2.7. Consumer executes Bulk processing

This use case builds upon case [Consumer discovers and executes On-demand Processing Service](#).

As a Consumer, I want to execute a processing service over a large amount of data. I want to utilise the EP's discovery service to select a processing service. I want to utilise the EP's discovery service to select multiple datasets, typically by specification or AOI and TOI to define the data inputs. The data discovery should facilitate selection of typical bulk processing scenarios, such as daily acquisitions, or selection by AOI. Having submitted my bulk processing, I would like to monitor the status of the processing, receive the completed *Processing Results*, with the ability to visualise/download etc. the results using the facilities of the EP. In the case of an error, then I should be alerted, with the capability to investigate the cause through access to the processing logs.

Figure 10 shows bulk processing at ESA's Grid Processing On Demand (G-POD) system.

The screenshot shows the G-POD web interface. At the top, there is a navigation bar with links for esa, Home, Services, Workspace, Catalogue, Products, Schedulers, My profile, Documentation, and Help. Below the navigation bar is a search bar with the placeholder "Search" and a dropdown menu set to "20". To the right of the search bar are buttons for "Logout" and "g-pod grid processing on demand". The main area is titled "Workspace" and contains a table of processing tasks. The table has columns: All, Prepared, Created, Pending, Active, Paused, Failed, Completed, Incomplete, Deleted. The "Completed" column is highlighted in green. The table lists numerous entries for "chris.NileBasin" using "Operational CE 01 UK 64bit" as the computing resource. Each entry includes Task Start Date, Task End Date, Status (all shown as "Completed"), Creation time, Submission time, and Completion time. The table is paginated at the bottom with "Pages: [1] [2]".

All	Prepared	Created	Pending	Active	Paused	Failed	Completed	Incomplete	Deleted
chris.NileBasin	Operational CE 01 UK 64bit	2012-01-25T00:00:00Z	2012-02-03T23:59:59Z	Completed	2012-03-01 21:08:23 (UTC)	2012-03-02 08:36:38 (UTC)	2012-03-02 09:34:28 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2012-01-15T00:00:00Z	2012-01-24T23:59:59Z	Completed	2012-03-01 20:47:37 (UTC)	2012-03-02 08:36:58 (UTC)	2012-03-02 09:29:28 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2012-01-05T00:00:00Z	2012-01-14T23:59:59Z	Completed	2012-03-01 20:47:22 (UTC)	2012-03-02 08:31:58 (UTC)	2012-03-02 10:18:29 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-12-26T00:00:00Z	2012-01-04T23:59:59Z	Completed	2012-03-01 20:38:22 (UTC)	2012-03-01 20:38:20 (UTC)	2012-03-01 21:26:24 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-12-16T00:00:00Z	2011-12-25T23:59:59Z	Completed	2012-03-01 20:37:22 (UTC)	2012-03-01 20:37:20 (UTC)	2012-03-01 21:24:21 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-12-06T00:00:00Z	2011-12-15T23:59:59Z	Completed	2012-03-01 20:35:22 (UTC)	2012-03-01 20:35:20 (UTC)	2012-03-01 21:08:21 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-11-26T00:00:00Z	2011-12-05T23:59:59Z	Completed	2012-03-01 20:18:22 (UTC)	2012-03-01 20:18:20 (UTC)	2012-03-01 20:47:21 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-11-16T00:00:00Z	2011-11-25T23:59:59Z	Completed	2012-03-01 20:13:22 (UTC)	2012-03-01 20:13:20 (UTC)	2012-03-01 20:47:21 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-11-06T00:00:00Z	2011-11-15T23:59:59Z	Completed	2012-03-01 19:59:22 (UTC)	2012-03-01 19:59:20 (UTC)	2012-03-01 20:38:21 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-10-27T00:00:00Z	2011-11-05T23:59:59Z	Completed	2012-03-01 19:58:28 (UTC)	2012-03-01 19:58:20 (UTC)	2012-03-01 20:37:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-10-17T00:00:00Z	2011-10-26T23:59:59Z	Completed	2012-03-01 19:58:22 (UTC)	2012-03-01 19:58:20 (UTC)	2012-03-01 20:35:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-10-07T00:00:00Z	2011-10-16T23:59:59Z	Completed	2012-03-01 19:48:22 (UTC)	2012-03-01 19:48:20 (UTC)	2012-03-01 20:18:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-09-27T00:00:00Z	2011-10-06T23:59:59Z	Completed	2012-03-01 19:41:22 (UTC)	2012-03-01 19:41:20 (UTC)	2012-03-01 20:13:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-09-17T00:00:00Z	2011-09-26T23:59:59Z	Completed	2012-03-01 19:24:22 (UTC)	2012-03-01 19:24:20 (UTC)	2012-03-01 19:58:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-09-07T00:00:00Z	2011-09-16T23:59:59Z	Completed	2012-03-01 19:14:27 (UTC)	2012-03-01 19:14:20 (UTC)	2012-03-01 19:58:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-08-28T00:00:00Z	2011-09-06T23:59:59Z	Completed	2012-03-01 19:14:24 (UTC)	2012-03-01 19:14:20 (UTC)	2012-03-01 19:58:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-08-18T00:00:00Z	2011-08-27T23:59:59Z	Completed	2012-03-01 19:14:22 (UTC)	2012-03-01 19:14:20 (UTC)	2012-03-01 19:48:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-08-08T00:00:00Z	2011-08-17T23:59:59Z	Completed	2012-03-01 19:10:21 (UTC)	2012-03-01 19:10:19 (UTC)	2012-03-01 19:41:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-07-29T00:00:00Z	2011-08-07T23:59:59Z	Completed	2012-03-01 18:57:21 (UTC)	2012-03-01 18:57:19 (UTC)	2012-03-01 19:24:20 (UTC)		
chris.NileBasin	Operational CE 01 UK 64bit	2011-07-19T00:00:00Z	2011-07-28T23:59:59Z	Completed	2012-03-01 18:46:21 (UTC)	2012-03-01 18:46:19 (UTC)	2012-03-01 19:14:20 (UTC)		

Figure 10. Bulk Processing - Results of a bulk processing from G-POD (<http://gpod.eo.esa.int/>)

1. Consumer logs in on the EP
2. **Discover and Select Processing Service...**
3. Consumer discovers and selects a Processing Service as described in use case [Consumer discovers and executes On-demand Processing Service](#). Steps not repeated here.
4. Alternatively, the Consumer selects a processing service from their Workspace
5. **Discover and Select Bulk Data...**
6. Consumer searches the EP catalogue for input data of interest, by specification of spatial/temporal (and other) characteristics
7. Consumer is able to select multiple data items by adding them to a preparatory 'bulk-data collection'
8. Consumer can perform multiple discrete searches to add more data to the 'bulk-data collection'

9. Consumer can select data from their workspace for addition to the 'bulk-data collection'
 10. The EP checks they are authorised to access the product
 11. Consumer views detailed metadata for the selected product
 12. Consumer views T&Cs for the service and accepts terms if not already done so
 13. **Initiate Bulk Processing...**
 14. Consumer specifies the input parameters of the *Processing Service*
 15. Optionally, the Consumer defines a collection of data to which the results have to be included
 16. Consumer requests bulk processing execution
 17. The EP checks that the Consumer has the authorisation to launch the *Processing Service* and access the specified data
 18. The EP estimates the cost and duration of the processing and checks the Consumer has enough resources to execute the processing
 19. Consumer is presented with the cost/duration estimation and confirms the processing
 20. The EP creates multiple processing requests, split according to the bulk data that has been selected
 21. Consumer monitors the status of the bulk processing (%completion, execution logs)
 22. When the processing completes successfully the *Processing Results* are made available to the user in their *Workspace* and/or the target *Collection* selected by the user
 23. The Consumer's billing account is updated commensurate with the 'cost' of the bulk processing
- 24. Alternative Flow: Bulk Processing Error**
- 25. Exploit Results...**
26. Optionally, the Consumer downloads the results
 27. Optionally, the Consumer visualises the processing logs (e.g. for error inspection)
 28. Optionally, the Consumer visualises the results and is able to manipulate and parameterise the view - with the possibility to download the result of their visualisation
 29. Optionally, the Consumer publishes their results in the catalogue - specifying all necessary metadata to support discovery

Alternative Flow: Bulk Processing Error

In the case of errors during bulk processing

- a. EP checks for errors during the processing
- b. Consumer is alerted to errors occurring during the bulk processing
- c. Consumer accesses bulk processing logs to investigate the error cause
- d. (Optionally) Consumer diagnoses problem and resubmits corrected bulk processing request.
This assumes that the error cause was under the control of the Consumer, i.e. they made an input error.

2.8. Consumer executes Systematic processing

This use case builds upon case [Consumer discovers and executes On-demand Processing Service](#).

As a Consumer, I want to setup an automated processing task that will be triggered for systematic execution by the platform (on behalf of the user). The triggering event(s) shall include:

- According to a time schedule (daily, weekly, etc.)
- Arrival of specific new input data
- External event trigger (e.g. earthquake alert trigger)

I want to utilise the EP's discovery service to select a processing service. I want to utilise the EP's discovery service to select input dataset(s). I want to define my systematic processing by specification of the triggering event for automated processing. I should be alerted that the systematic processing has been triggered. I would like to monitor the status of the processing, receive the completed *Processing Results*, with the ability to visualise/download etc. the results using the facilities of the EP. In the case of an error, then I should be alerted, with the capability to investigate the cause through access to the processing logs.

[Figure 11](#) shows setup of systematic EO data processing at ESA's Grid Processing On Demand (G-POD) system.

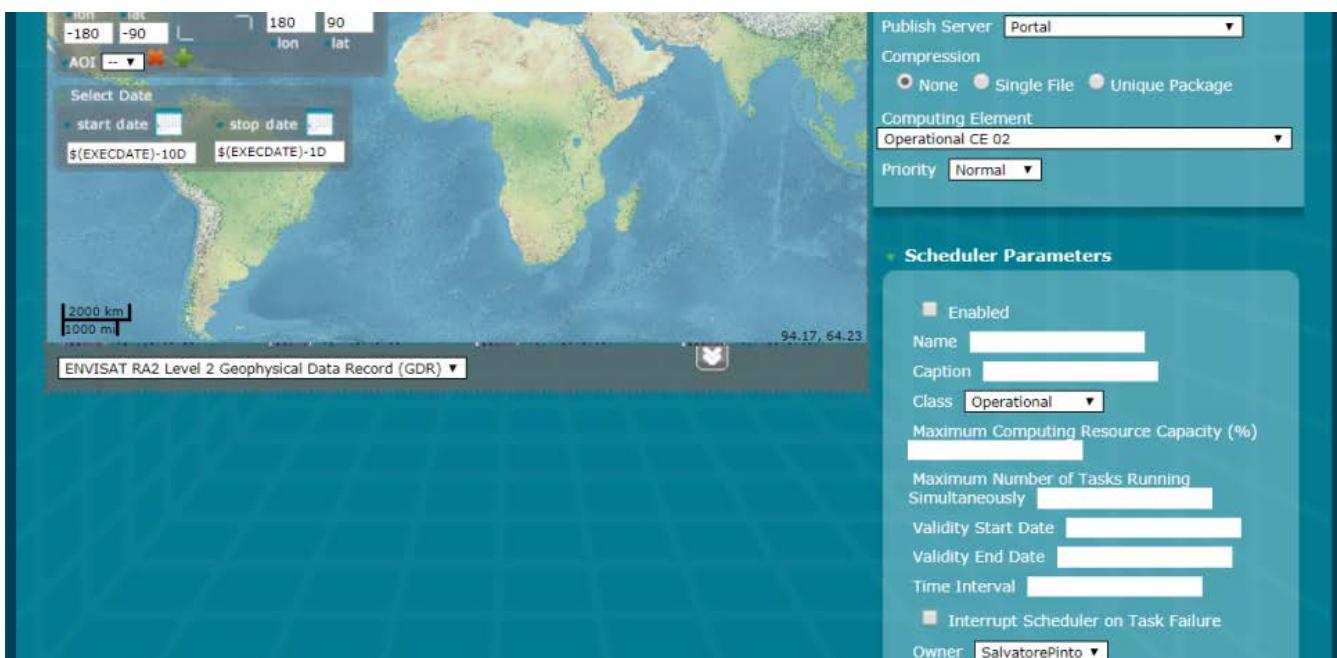


Figure 11. Systematic Processing - Setup of systematic processing from G-POD (<http://gpod.eo.esa.int/>)

1. Consumer logs in on the EP
2. **Discover and Select Processing Service...**
3. Consumer discovers and selects a Processing Service as described in use case [Consumer discovers and executes On-demand Processing Service](#). *Steps not repeated here.*
4. Alternatively, the Consumer selects a processing service from their Workspace

- 5. Discover and Select Input Data...**
 6. Consumer discovers and selects Input Data as described in use case [Consumer discovers and executes On-demand Processing Service](#). *Steps not repeated here.*
 7. Alternatively, the Consumer selects input data from their Workspace
 - 8. Define Systematic Processing...**
 9. Consumer specifies the input parameters of the *Processing Service*
 10. Consumer specifies the systematic processing triggering conditions
 11. Optionally, the Consumer defines a collection of data to which the results have to be included
 12. The EP checks that the Consumer has the authorisation to launch the *Processing Service* and access the specified data
 13. The EP estimates the future cost/duration of the processing and checks the Consumer has enough resources to execute the processing. See [Systematic Processing Costs](#) below.
 14. Consumer is presented with the cost/duration estimation and confirms the systematic processing
 - 15. Execute Systematic Processing...**
 16. EP initiates systematic processing according to the defined trigger condition
 17. Consumer monitors the status of the systematic processing (%completion, execution logs)
 18. When the processing completes successfully the *Processing Results* are made available to the user in their *Workspace* and/or the target *Collection* selected by the user
 19. The Consumer's billing account is updated comensurate with the 'cost' of the systematic processing
 20. Optionally, the EP notifies the Consumer of the occurrence and completion of the systematic processing
 - 21. Alternative Flow: Systematic Processing Error**
 - 22. Exploit Results...**
 23. Optionally, the Consumer downloads the results
 24. Optionally, the Consumer visualises the processing logs (e.g. for error inspection)
 25. Optionally, the Consumer visualises the results and is able to manipulate and parameterise the view - with the possibility to download the result of their visualisation
 26. Optionally, the Consumer publishes their results in the catalogue - specifying all necessary metadata to support discovery
- Alternative Flow: Systematic Processing Error*
- In the case of errors during systematic processing
- a. EP checks for errors during the processing
 - b. Consumer is alerted to errors occurring during the systematic processing
 - c. Consumer accesses systematic processing logs to investigate the error cause
 - d. (Optionally) Consumer diagnoses problem and resubmits corrected systematic processing

definition. This assumes that the error cause was under the control of the Consumer, i.e. they made an input error.

Notes

Systematic Processing Costs

NOTE Given that the systematic processing occurs asynchronous to the Consumer submitting the definition, the possibility exists that, at time of trigger/execution, the Consumer no longer has sufficient resources to cover the task. This condition must be trapped and handled by the EP - perhaps raising an error to the Consumer. See alternative flow [Systematic Processing Error](#).

2.9. Consumer performs Open Science

As a Consumer I want to share my analysis and processing results in such a way that facilitates scientific collaboration. I want to link my results to a scientific publication, and assign a DOI to my results to reference it in my scientific paper. I want to encapsulate my research/analysis as a *Reusable Research Object*, to capture all aspects of my analysis including data used and processing performed etc., so that it can be reproduced, I can re-use my analysis in the future, or share it with others for collaboration. A simple example could be a Jupyter notebook that captures an annotated analysis, identifying input data, executing code to transform the data, and presenting/visualising the output results. This case should then be generalised for execution of more complex analyses using platform data / processing services (local and external).

Figure 12 illustrates a shared analysis result linked to a reference scientific publication.

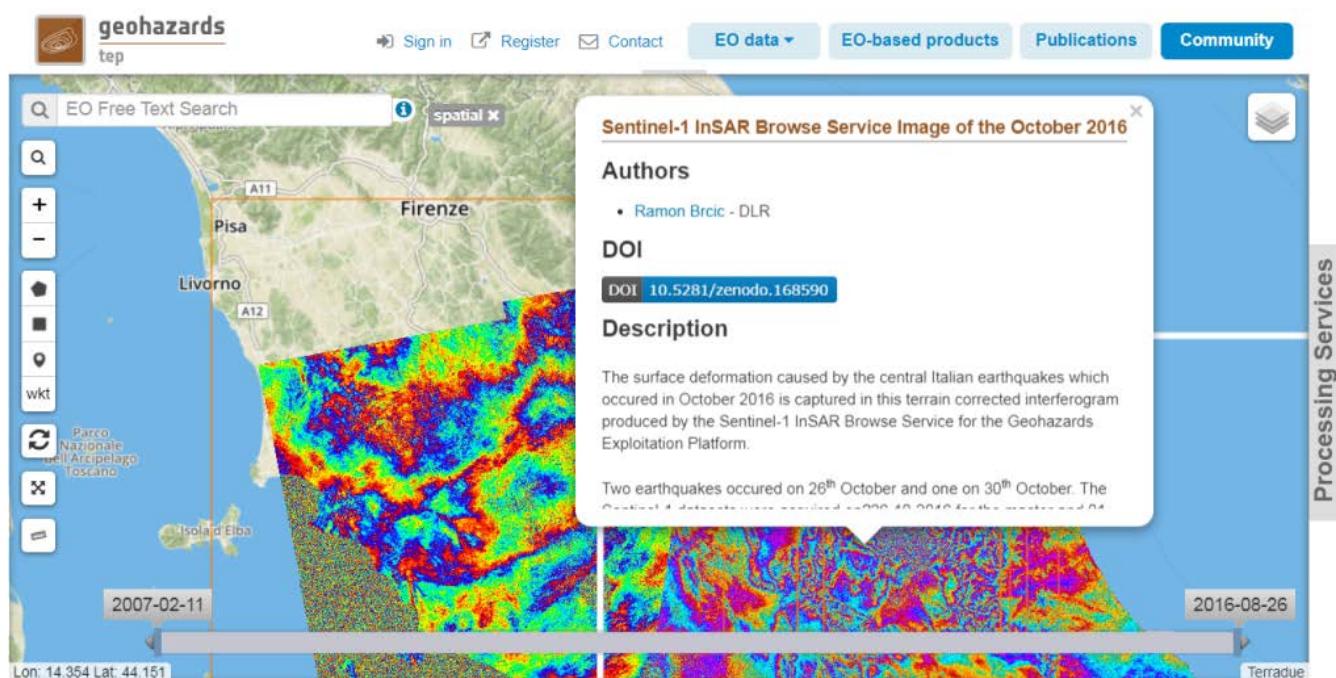


Figure 12. Open Science - Sharing analysis results on the Geohazards TEP (<https://geohazards-tep.eu/>)

1. Consumer logs in on the EP
2. Consumer selects items (e.g. data, processing services, processing results)
3. EP verifies that Consumer has right to share the item
4. Consumer selects one of the collaborative options, including:
 - a. share processing results
 - b. link to scientific publication
 - c. create a Reusable Research Object for sharing
 - d. Others TBD - see note [Collaboration Options](#).
5. Consumer specifies collaborators and associated access rights (read/write - user/group/everyone). See note [Collaboration Groups](#).
6. The EP shares the item according to the Consumers specification

7. Optionally, the EP registers a DOI for the item
8. Optionally, collaborators are automatically notified of the sharing of the item
9. Collaborators access the shared item; the EP checks that the Consumer is authorised to access the item

Notes

Collaboration Options

NOTE Each of the 'collaborative options' needs to be defined in more detail, and perhaps explored through additional use cases.

Collaboration Groups

NOTE This use case assumes the concept of 'collaborators'. This implies a grouping of users that needs to be explored in further details through additional use cases.

2.10. Consumer accesses EP services with External Application

As a Consumer I want to be able to use and/or develop an external application that uses the services of the EP via programmatic interfaces, (i.e. public API). I want to delegate to the application, authorising it to act on my behalf when interfacing to the EP. Hence, my usage of the external application is made within the context of my EP user profile, workspace, account etc. as if I was interacting with the EP through its 'native' user interface. Examples of external applications include mobile applications, or simply scripts that remotely exploit the data / processing services of the EP.

Figure 13 illustrates a mobile application interfacing with the Forestry TEP discovery and visualisation services.

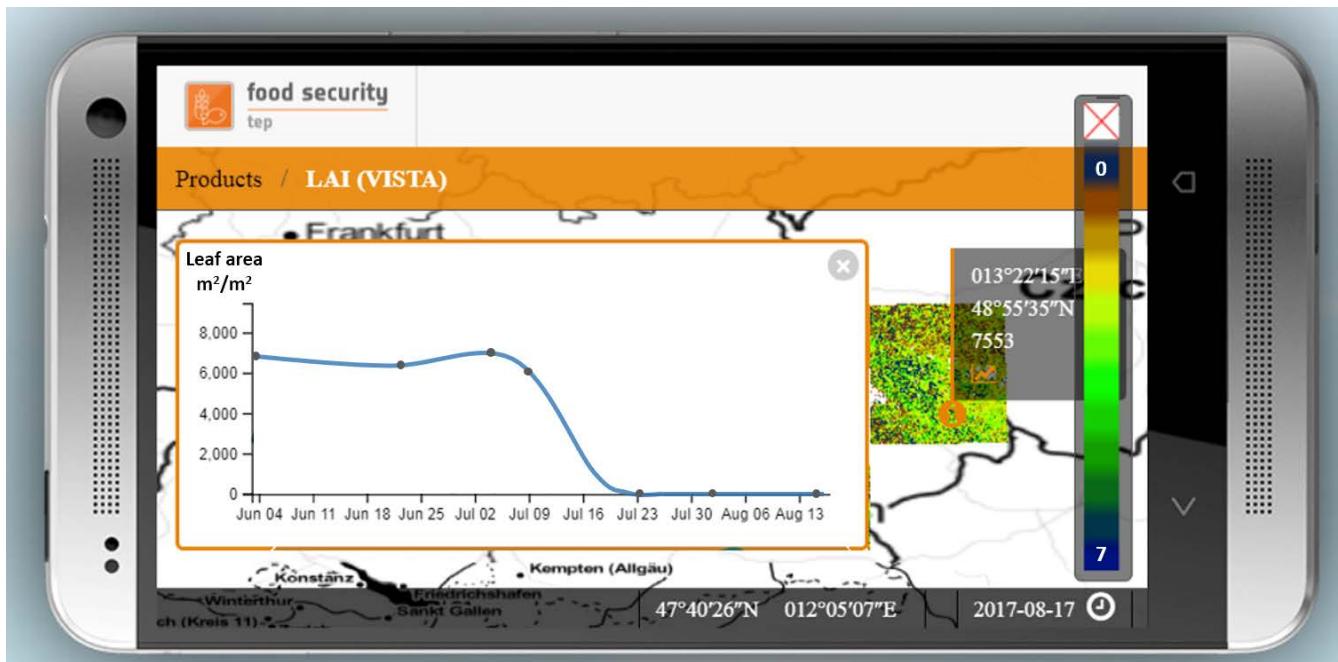


Figure 13. External Application - Mobile application exploiting Forestry TEP services (<https://f-tep.com/>)

1. Consumer installs on their local host an external application, e.g. installs an app on their mobile phone
2. Consumer starts application and are directed to authorise the application to access the EP on their behalf
3. Consumer authenticates to the EP and grants the requested privileges to the application
4. External Application uses the Consumer's delegated credentials to access the services of the EP, via a programmable interface (e.g. API)
5. EP uses the delegated credentials to check that the application is authorised to access the requested data / service
6. External Application utilises the data and services of the EP to provide a full-featured user experience to the Consumer. For example, allowing execution of processing services and visualisation of results.
7. The Consumer's billing account is updated commensurate with the 'cost' of the actions performed

Notes

NOTE

External (non-interactive) Script

The case of an external script might need a slightly different approach (flow) for the authorisation of the delegated access. For example, a non-interactive (batch) script may need to obtain the delegated credentials in advance of the script execution.

2.11. Expert user builds new processing service

As an Expert user, I want to integrate my own software into the platform to be exposed as a new processing service. I want to be able to prepare the software in a self-contained package containing all execution dependencies, for loading into the EP. I want to test the package by deploying and executing it in a hosted test environment, with access to platform data for testing. I want to debug the software and inspect the processing logs. Once satisfied, I want to publish the application as a new processing service, supported by ancillary information including metadata and processor user manual. The metadata should enable discovery of the service and facilitate its usage with compatible data and in chaining. Once published, the new processing service should be discoverable in the EP Catalogue, available for Consumers according to access rights I define.

Figure 14 illustrates packaging of a new processing service, from the Forestry TEP platform.

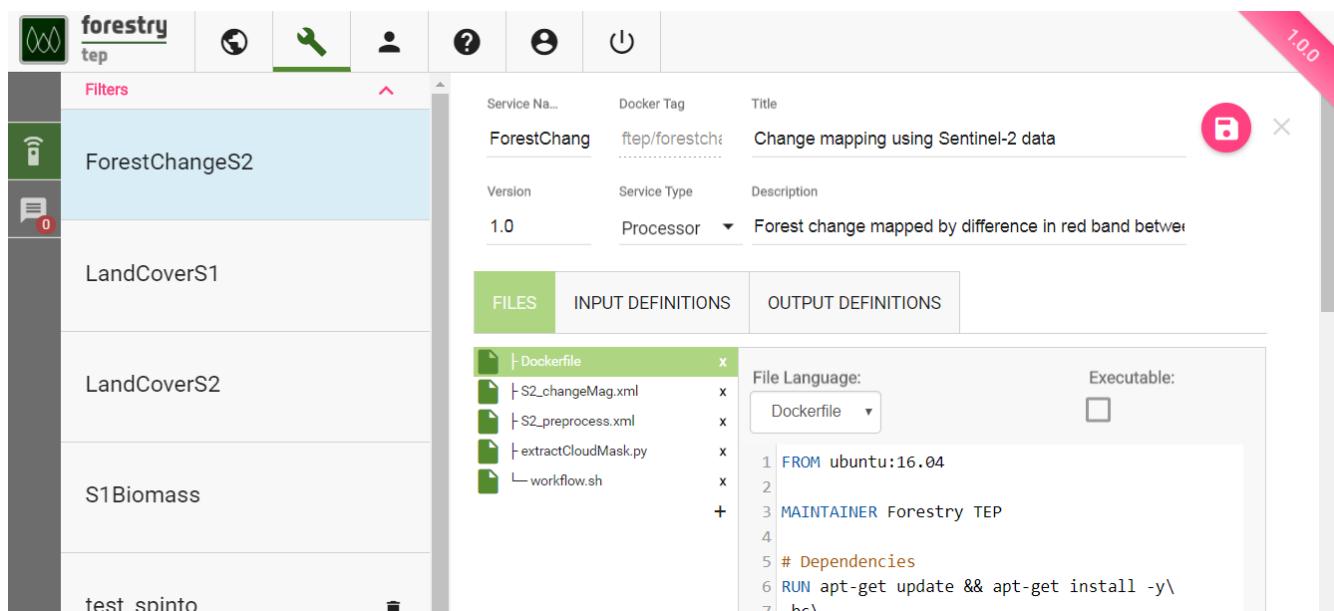


Figure 14. Build Processing Service - Packaging a new processor using Docker image - Forestry TEP (<https://f-tep.com/app/>)

1. Expert user logs in on the EP
2. Expert access EP processor development environment
3. Expert packages the software into a non-interactive application package, conformant with the EPs application package standard
4. Alternatively, Expert prepares the application package offline (conformant with the EPs application package standard) and uploads to the EP processor development environment
5. Expert deploys the package as a new 'test' (unpublished) processing service; the EP checks the Expert is authorised to do so
6. Expert tests the procesing service execution, by performing an execution in accordance with use case [Consumer discovers and executes On-demand Processing Service](#)
7. EP checks for errors and notifies Expert as required
8. Expert checks processor logs for correct operation

9. Expert checks *Processing Results* for expected outputs
10. If necessary, Expert modifies software, re-packages/deploys, and repeats the testing cycle
11. When the process is stable, Expert publishes the packages as a new processing service; the EP checks the Expert is authorised to do so
 - a. Expert specifies metadata to describe the processor to make it discoverable, and to facilitate the EP to ensure compatible use of the processing service (e.g. with compatible input data, and output data for compatible chaining)
 - b. Expert provides user manual to aid users of the processing service
12. The EP checks the integrity of the software and that it adheres to the EP terms and conditions
13. The EP adds the processing service to the Application Catalogue so that is available to Consumers in their discover/browse searches
14. Optionally, EP notifies Consumers about the new service

Notes

NOTE

Processor Preparation Billing

This use-case does not consider whether billing is applied to the usage of the processor test environment, and the associated processor execution. It is assumed that billing may not be applied, in order to encourage such contributions. However, it should be recognised that this could be abused by Experts exploiting the development environment as a convenient processing environment.

2.12. Expert user builds new processing services chains

As an Expert user, I want to chain multiple processing services, potentially offered by different platforms, in parallel or sequentially. I want to prepare the chain by defining sequencing/relations of the processing services, and the input/output parameterisation of each step in the chain. I would like to publish the processing services chain as a new processing service for me or the Consumer to use, supported by ancillary information including metadata and processor-chain user manual, (see note [Reusable Chain Specification](#)). The metadata should enable discovery of the service and facilitate its usage with compatible data and in chaining, (see note [Nested Chaining](#)). Once published, the new processing service should be discoverable in the EP Catalogue, available for Consumers according to access rights I define.

Figure 15 illustrates building a processing chain at the Geohazards TEP platform.

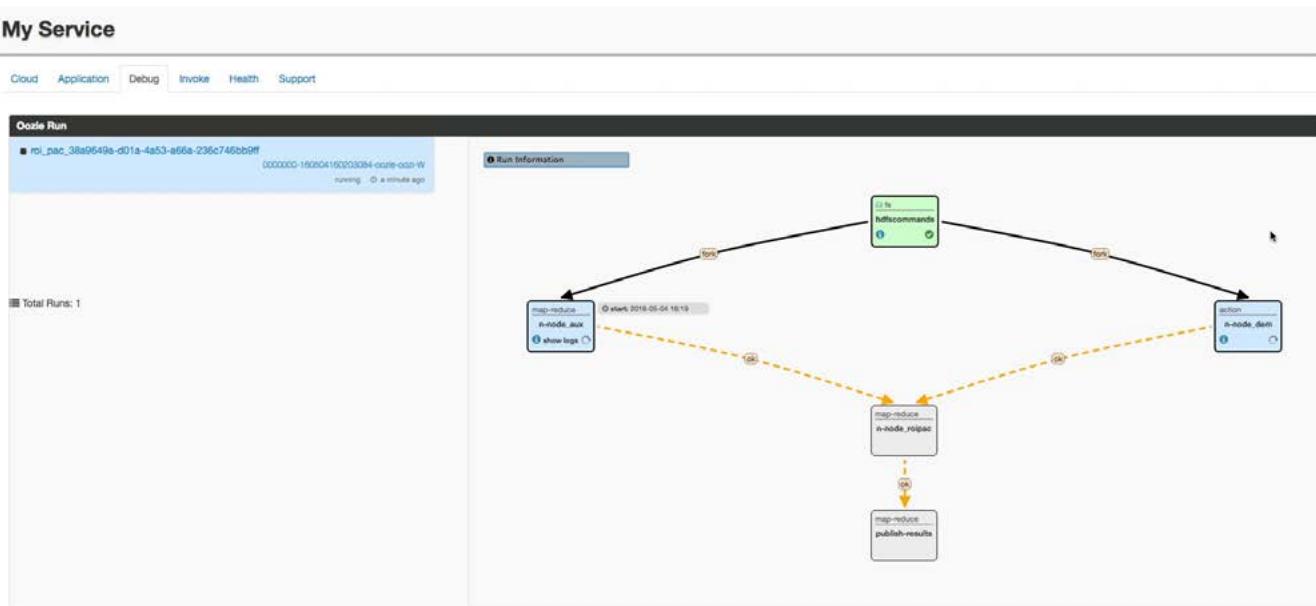


Figure 15. Build Processing Chain - Example taken from the Geohazards TEP (<https://geohazards-tep.eu/>)

1. Expert user logs in on the EP
2. Expert discovers processing services from the platform, in accordance with [Consumer discovers and executes On-demand Processing Service](#). This step should include the facility to discover/include processing services from other platforms.
3. Expert chooses processing services to be chained; the EP checks the Expert is authorised to use those processing services
4. Expert defines all aspects of the chain by specifying the sequencing and relationships of the steps, and the input/output parameters of each step
5. Expert tests the processing-chain, as if it was a new processing service, as described in use case [Expert user builds new processing service](#); EP ensures the Expert is authorised to do so
6. If necessary the Expert refines the processing-chain defintion and repeats the testing cycle
7. When the processing-chain is stable, Expert publishes the chain as a new processing service; the

- EP checks the Expert is authorised to publish the processing chain
- a. Expert specifies metadata to describe the processor-chain to make it discoverable, and to facilitate the EP to ensure compatible use of the processing service
 - b. Expert provides user manual to aid users of the processing service
8. The EP adds the processing service to the Application Catalogue so that is available to Consumers in their discover/browse searches
 9. Optionally, EP notifies Consumers about the new service

Notes

Reusable Chain Specification

NOTE In order to publish the processing-chain as a generally reusable processing service, then the specific input data should not be specified in the chain definition. Instead this should be specified at time of an individual execution, in order to ensure the chain is usable as a general resource.

Nested Chaining

NOTE Once published as a processing service, it should in principle then be possible to regard this processing-chain as a single step in another processing-chain.

2.13. Expert user builds new interactive application

As an Expert user, I want to integrate my standalone application software into the platform to be exposed as a new interactive application. I want to be able to prepare the software in a self-contained package containing all execution dependencies, for loading into the EP. I want to test the package by deploying and executing it in a hosted test environment, with access to platform data for testing. I want to debug the software and inspect the execution logs. Once satisfied, I want to publish the application as a new interactive application, supported by ancillary information included metadata and application user manual. The metadata should enable discovery of the application and facilitate its usage with compatible data. Once published, the new interactive application should be discoverable in the EP Catalogue, available for Consumers according to access rights I define.

Figure 16 illustrates packaging of a new interactive application, from the Forestry TEP platform. The key point here is that the interactive application must be engineered in such a way as to support the remoting of the application's user interface. Thus, the EP's application package standard must support this approach, and the EP must ensure the network connectivity required to deliver the remote view.

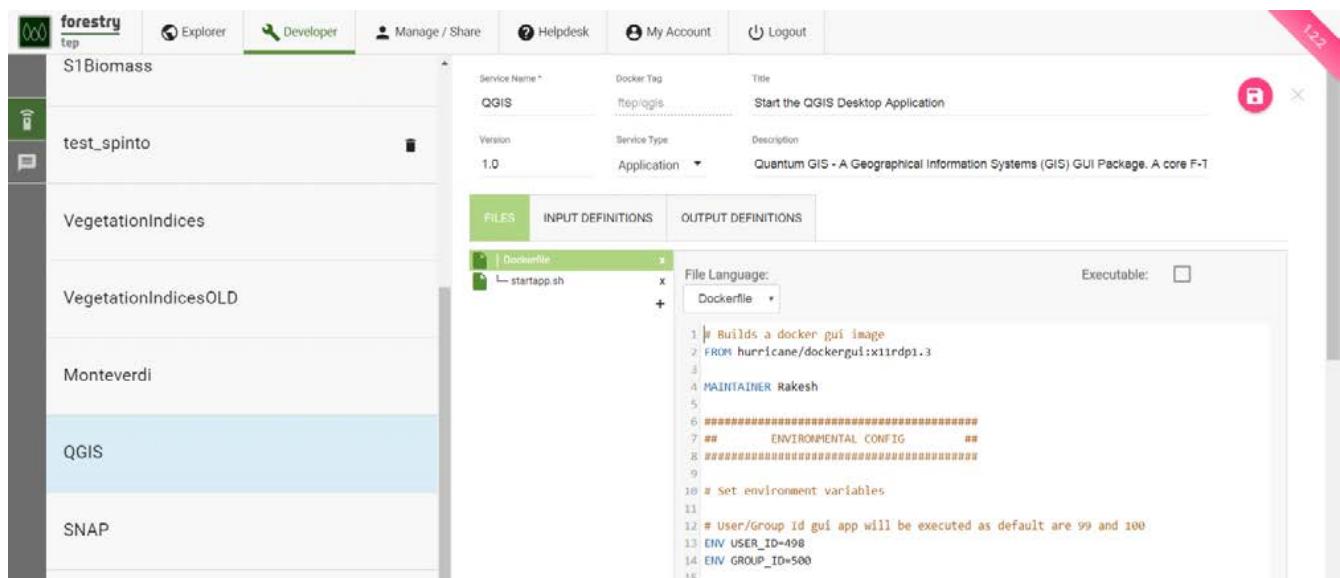


Figure 16. Build Interactive Application - Packaging a new application using Docker image - Forestry TEP (<https://f-tep.com/app/>)

This use-case is largely the same as case [Expert user builds new processing service](#), the main difference being the nature of the application being deployed, i.e. an application that presents a user interface to the Consumer through their web browser interface. The interactive application may be of the following type:

Interactive Console Application

A simple Interactive Application for analysis in which a console interface to a platform-hosted terminal is provided to the user. The console interface can be provided through the user's browser session or through a remote SSH connection.

Interactive Web Application

An Interactive Application for analysis provided as a rich user interface through the user's web

browser.

Interactive Remote Desktop

An Interactive Application for analysis provided as a remote desktop session to an OS-session (or directly to a 'native' application) on the exploitation platform. The user will have access to a number of applications within the hosted OS. The remote desktop session is provided through the user's web browser.

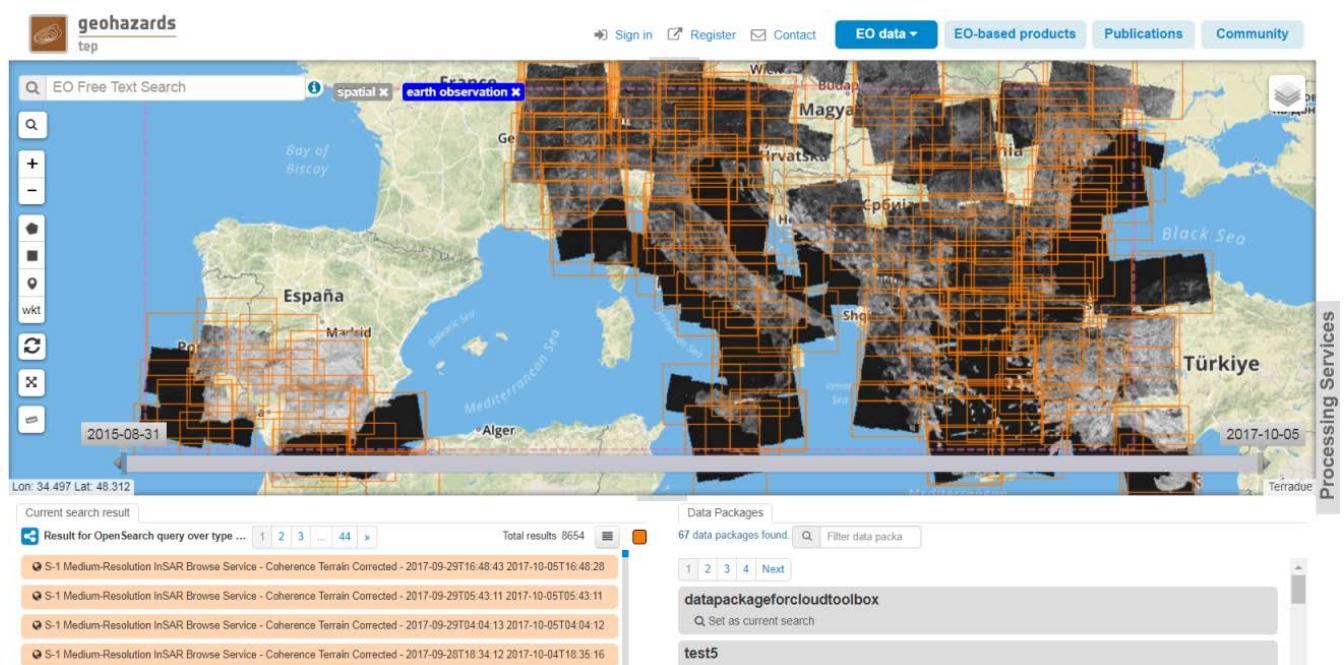
For brevity, the use-case steps are not reproduced here - instead case [Expert user builds new processing service](#) should be referred to, taking into account the deployment of an *Interactive Application* rather than a *Processing Service*.

2.14. Expert user builds new value-added products

This use-case builds upon cases [Consumer executes Bulk processing](#) and [Consumer executes Systematic processing](#).

As an Expert user, I want to be able to configure and execute a bulk processing (use-case [Consumer executes Bulk processing](#)) or systematic processing (use-case [Consumer executes Systematic processing](#)), and publish its results automatically as a new value-added product collection into the platform, (or add products to an existing collection). Once published, the new value-added products should be available in the EP Catalogue, for Consumers to discover/visualise (use-case [Consumer discovers and visualises products](#)), according to access rights I define.

[Figure 17](#) shows Sentinel-1 InSAR products systematically produced after each Sentinel-1 acquisition, and published into a platform collection, in the Geohazards TEP.



[Figure 17. New Value-added Product - Systematic publication of value-added products - Geohazards TEP \(<https://geohazards-tep.eu/>\)](#)

1. Expert user logs in on the EP
2. Expert defines a product collection, including description of the product, basic metadata and other information. *This data should be sufficient to support discovery of the collection*
3. Expert selects a bulk or systematic processing (defined in use-cases [Consumer executes Bulk processing](#) and [Consumer executes Systematic processing](#)); the EP checks the Expert is authorised to access the selected processing
4. Expert configures the bulk or systematic processing to publish data into the defined product collection, (or an already existing collection); the EP checks the Expert is authorised to add products to the selected collection
5. EP publishes results automatically once processing is over, and includes the new value-added products into the selected collection

6. Optionally, EP notifies Consumers about the new products

2.15. Expert develops with interactive development environment

As an Expert user, I want to access an on-line interactive development environment where I can author and/or upload my code (written according to my preferred programming language) and execute it in total or step-by-step, to analyse input data, value-added products and other ancillary data. In my code implementation, I want to use a general set of libraries to process and visualise EO data (e.g. SNAP python library) together with specific libraries required by my application (e.g. Tensorflow machine learning library) and libraries to access the other services provided by the platform.

This use-case might be seen as a special case of case [Consumer discovers and executes Interactive Applications](#), since the interactive development environment may be considered to be an interactive application.

[Figure 18](#) shows an example implementation using a Jupyter Notebook - mocked-up with images from Raster Foundry.

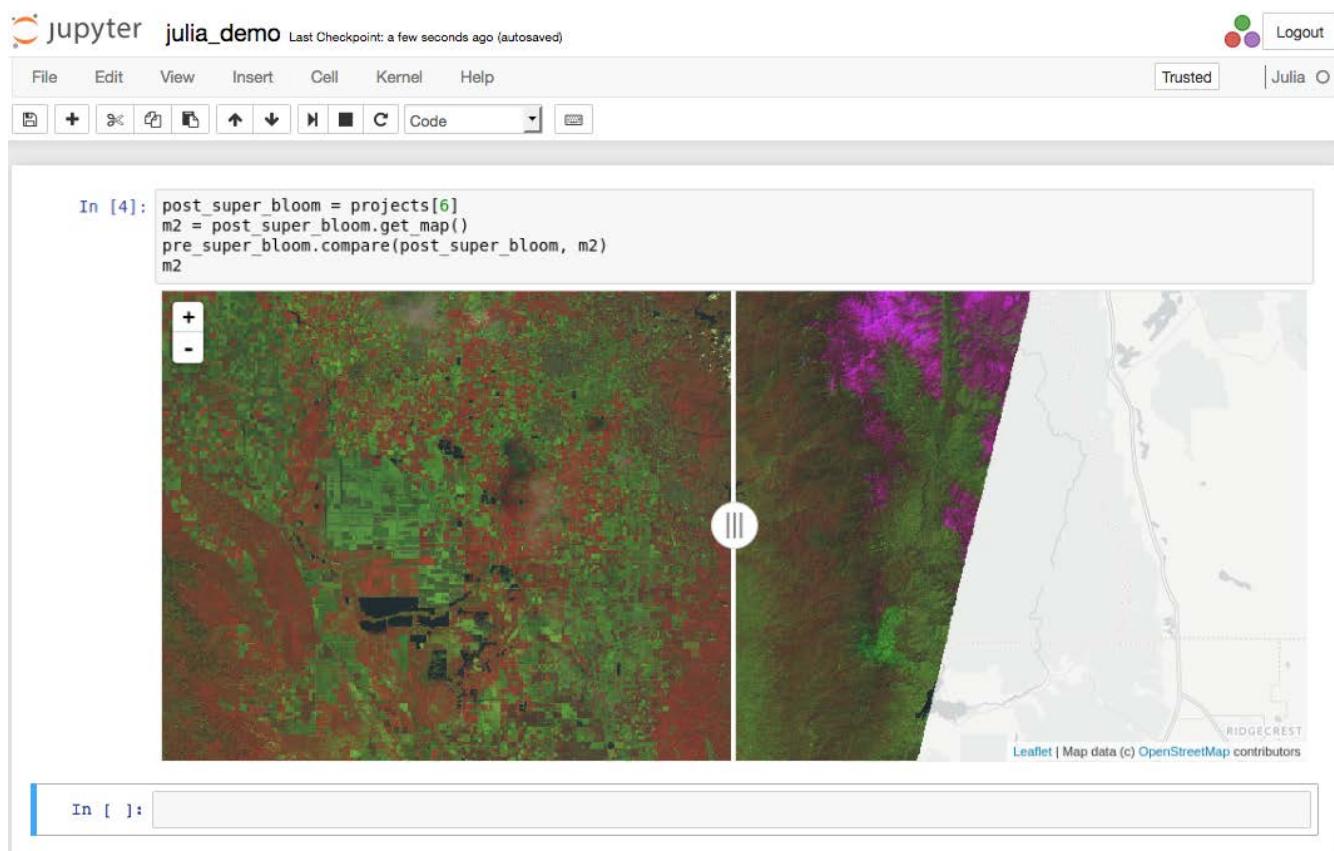


Figure 18. Web Development Environment - Example mocked-up with images from Raster Foundry (<https://www.rasterfoundry.com/>)

1. Expert user logs in on the EP
2. Expert accesses on-line interactive development environment; the EP checks that the Expert is authorised to do so
3. Expert develops and/or uploads code. Supported programming languages include: python, R,

others TBD

4. Within the interactive development environment, Expert is provided with a 'platform' library that provides them access to the platforms resources (data/processing) from their code; according to their user rights in the EP
5. Within the interactive development environment, Expert is provided with a standard set of libraries against which to develop their code, for EO data manipulation and visualisation
6. Expert is able to configure additional specific libraries to be imported into their working environment, against which to develop their code, to satisfy specific code needs
7. Expert executes their code in total or step-by-step to perform their analysis; the EP checks they are authorised to access the product(s) used
8. Optionally, Expert saves their development (code) in their platform *Workspace*, for later re-use.
*Note that this step may overlap with use-case **Consumer performs Open Science**, if the development can be saved within a Reusable Research Object*
9. Optionally, Expert saves their Processing Results in their platform *Workspace*
10. Optionally, Expert downloads their developments (code) and/or saved processing results

Notes

Overlap with Other Use-cases

It is noted that there is some overlap between this use-case and:

Consumer discovers and executes Interactive Applications

NOTE

Since the interactive development environment may be considered to be an interactive application

Consumer performs Open Science

Since the Expert's saved work might form the basis for a *Reusable Research Object*

2.16. Expert performs training

As an Expert user, I want to provide support to Consumers in the use of platform processing services and interactive applications, by provision of learning aids, including: documentation, tutorials, webinars/lessons/demonstrations (live and on-demand), exercies and tests. I would like to organise these into on-line courses for Consumers to attend. *Note that it might be typical for an Expert to create such training content for their own services/application, but we should not preclude the case to create learning content for other's services/applications. This is in the spirit of creating an EP as a collaborative environment.*

As a Consumer, I want to be able to discover/browse the learning content, and attend on-line courses.

Figure 19 provides an illustration (mock-up) of a possible course list implementation.

472 courses available			
COURSE NAME		START DATE	RATING
	Learn to Program: The Fundamentals University of Toronto via Coursera	19th Aug, 2013	
	Principles of Reactive Programming École Polytechnique Fédérale de Lausanne via Coursera	13th Apr, 2015	
	Morality of Everyday Life Yale University via Coursera	7th Oct, 2014	

Figure 19. Training - Course List example mock-up with images from <http://coursera.org/>

1. Expert Course Preparation...

2. Expert user logs in on the EP
3. Expert accesses training environment
4. Expert defines on-line courses; the EP checks the Expert has authorisation to do so
5. Expert identifies processing services and interactive applications as the subject of the on-line course
6. Expert prepares on-demand training content, including: documentation, tutorials, webinars, lessons, demonstrations, exercies and tests
In the simple case, the Expert prepares the content offline (e.g. screencast videos) and uploads to the platform. In the more sophisticated case, the EP provides functions to facilitate the creation of the content
7. Expert schedules live webinars/lessons/demonstrations
8. Expert broadcasts live (at previously arranged date/time) webinars/lessons/demonstrations, for the live 'attendance' by Consumers

9. Consumer Course 'Attendance'...

10. Consumer logs in on the EP
11. Consumer discovers/browses for on-line courses
12. Alternatively, Consumer has selected a processing service or interactive application of interest,

and is provided with links to related on-line courses

13. Consumer requests to attend on-line course
 14. EP checks the Consumer has enough resources to cover the course costs
 15. EP and/or Expert grants authorises Consumer access to the course
 16. The Consumer's billing account is updated comensurate with the 'cost' of the on-line course
 17. Consumer accesses course content via the training environment defined by the Expert; the EP checks the Consumer is authorised to do so
 18. Consumer accesses on-demand course content
 19. Consumer 'attends' live course content as delivered by the Expert
-

3. Functional Mapping

Identification of functional areas and their mapping to domain areas.

The Exploitation Platform architecture is defined by services that together provide the complete functionality of the platform. Domain Areas are defined as groupings of related services, which provide the building blocks for the system. Dedicated specialist expertise can then be applied to each domain area to ensure the best design and development across the system. Inevitably, there will be some services that straddle the notional divide between domain area - in which case close collaboration is essential between domain area experts to establish a holistic solution.

The identified domain areas are as follows:

User Management (UM)

All aspects of user identification, authentication and authorisation in a federated system-of-systems environment.

Processing & Chaining (P&C)

Hosting and maintaining an inventory of all processing tasks, analysis tools and interactive applications. Handles and abstracts the low-level complexities of the different underlying compute technologies, and ensures the compute layer is scaled in accordance with current demand. Provides an integrated development environment to facilitate development of new processing algorithms and applications. Facilitating the network of EO resources by providing a federated interface to other processing services within the wider EO network.

Resource Management (RM)

Storage and cataloguing of all persistent resources, including data and other supporting material such as documents. Handles and abstracts the low-level complexities of different underlying storage technologies and strategies. Facilitating the network of EO resources by providing a federated interface to other data services within the wider EO network.

In the following subsections we identify functionality that is derived from the use-cases, and then map this onto domain areas.

3.1. User accesses Platform services

Platform Function	UM	P&C	RM
Each user uniquely identified	x		
User identification/authentication using external Identity Provider	x		
User assigned access privileges to platform resources (data/services)	x		
Unidentified / unauthorised user is regarded as a Guest with limited access	x		
EP enforces authorisation when user accesses platform resources (data/services)	x		
EP accesses resources of other platform on behalf of a user (delegated)	x		

Platform Function	UM	P&C	RM
EP enforces authorisation when an external platform attempts delegated access on behalf of a user	X		