Use Case Analysis Document

EOEPCA.TN.005

TVUK System Team

Version Draft for 1.1, In Progress:

Use Case Analysis

EO Exploitation Platform Common Architecture
Use Case Analysis Document
EOEPCA.TN.005

COMMENTS and ISSUES	PDF
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.	This document is available in PDF format here.
EUROPEAN SPACE AGENCY CONTRACT REPORT The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organisation that prepared it.	**TELESPAZIO VEGA UK Ltd** 350 Capability Green, Luton, Bedfordshire, LU1 3LU, United Kingdom. Tel: +44 (0)1582 399000 www.telespazio-vega.com

AMENDMENT HISTORY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

Table 1. Amendment Record Sheet

ISSUE	DATE	REASON
1.1	InProgress	Updates during development of Reference Implementation
1.0	02/08/2019	Issue for domain expert ITT
0.7	18/07/2019	Typos, clarifications and updates to template
0.6	09/07/2019	Added some cross-platform compute requests to use cases
0.5	11/06/2019	Initial functional mapping analysis
0.4	18/04/2019	Complete use case descriptions
0.3	18/04/2019	Addition of more use case descriptions
0.2	15/02/2019	First set of use case descriptions
0.1	09/01/2019	Initial draft

Chapter 1. Introduction

1.1. Purpose and Scope

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

1.2. Structure of the Document

Section 2 - Use Cases

Provides a description of each use case.

Section 3 - Functional Mapping

Derives high-level functionalities from the use-cases and maps these to *Domain Areas*.

Section 4 - Operational Mapping

Provides a mapping from 'operational' use cases (ref. [EP-OP-UC]) into the 'functional' use cases presented in this document.

1.3. Reference Documents

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

Reference	Document Details	Version
[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
[EP-OP-UC]	Operational use cases for EO Platforms Interoperability, EMSS-EOPS-TN-20-08	Issue 1.0, 18/03/2020

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

Term	Meaning
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.
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. *It is assumed that the Hosting Environment for the EP makes available an existing supply of EO Data*
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 *Processing Results* with a well-defined output data type.
On-demand Processing Service	A *Processing Service* whose execution is initiated directly by the user on an ad-hoc basis.
Systematic Processing Service	A *Processing Service* 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 *Processing Service* on large amounts of data specified by AOI and TOI.
Processing Result	The *Products* produced as output of a *Processing Service* 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.

Term	Meaning
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 *Analysis Results* 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 *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.
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).
Analysis Result	The *Products* produced as output of an *Interactive Application* analysis session.
Execution	The act to start a *Processing Service* or an *Interactive Application*.
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 (interpretted) 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 patters within the data. Analytics considered here are performed manually (or in a semi-automatic way) on-line with the aid of *Interactive Applications*.

Term	Meaning
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.
Licenser	A copyright owner or agent authorized to issue licences for access to data or processing services, typically on commercial terms
Resource	A entity, such as a Product, Processing Service or Interactive Application, which is of interest to a user, is indexed in a catalogue and can be returned as a single meaningful search result

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
SSH	Secure Shell
TOI	Time of Interest
VNC	Virtual Network Computing

Chapter 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. The data/products available to the user should not be limited to EO data - it should be possible for the user to augment their processing/analysis with data from other sources such as in-situ sensor data, lidar etc. 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.

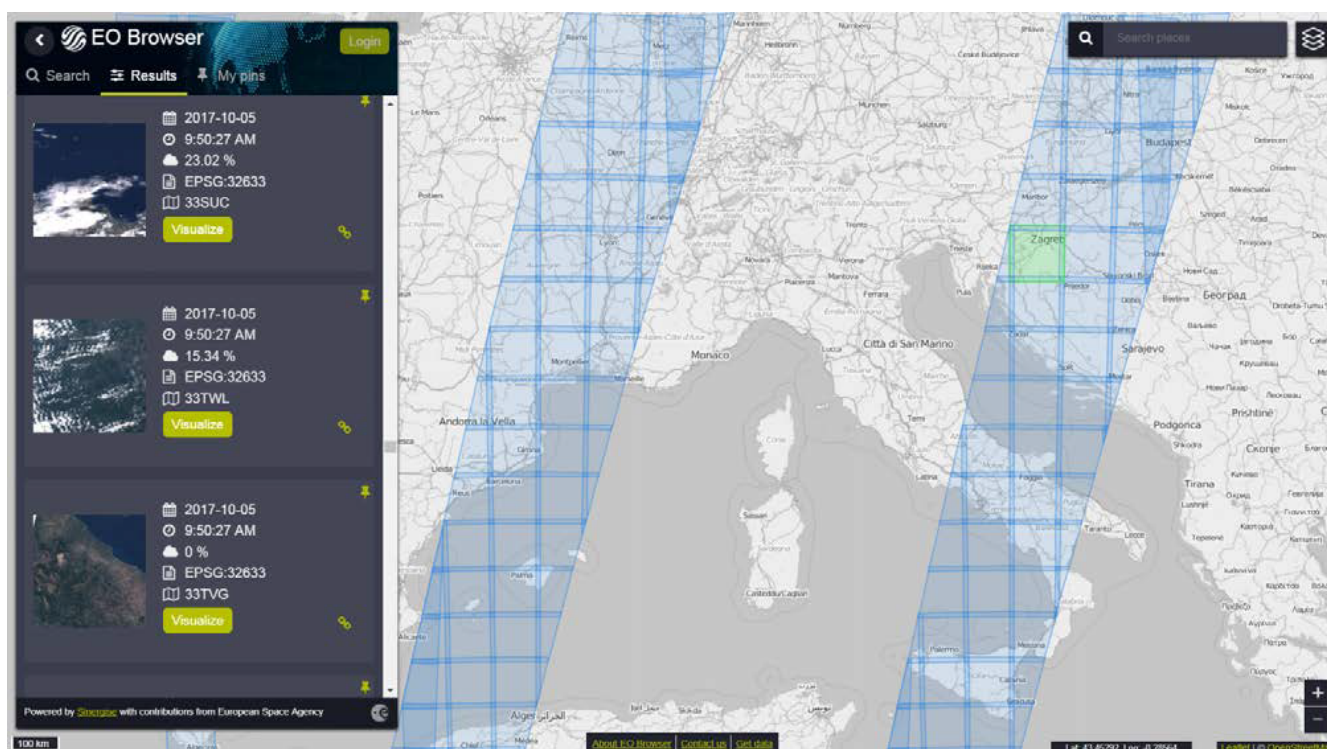Figure 1 illustrates discovery of EO data products using the EO Browser system.

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, value-added products and additional data such as in-situ sensor measurements

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

Alternative Flow: User Discovers Resources using a General-Purpose Search Engine

A User, who may be unaware of the platform, may begin searching for products using a search engine such as Google. The User may then follow a link directly to a product-related page.

Notes

Guest User

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. *This approach might be typical in order to allow a potential user discover EP capabilities before signing up*

Search Engine Indexing

To facilitate Users discovering products and the EP as a whole using a search engine it's necessary to allow search engines to effectively index them. This requires Guest access to as much as possible but also extends further to other SEO requirements.

Data Sub-setting

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. *The extent to which this case should be considered for the EP is TBD*

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.
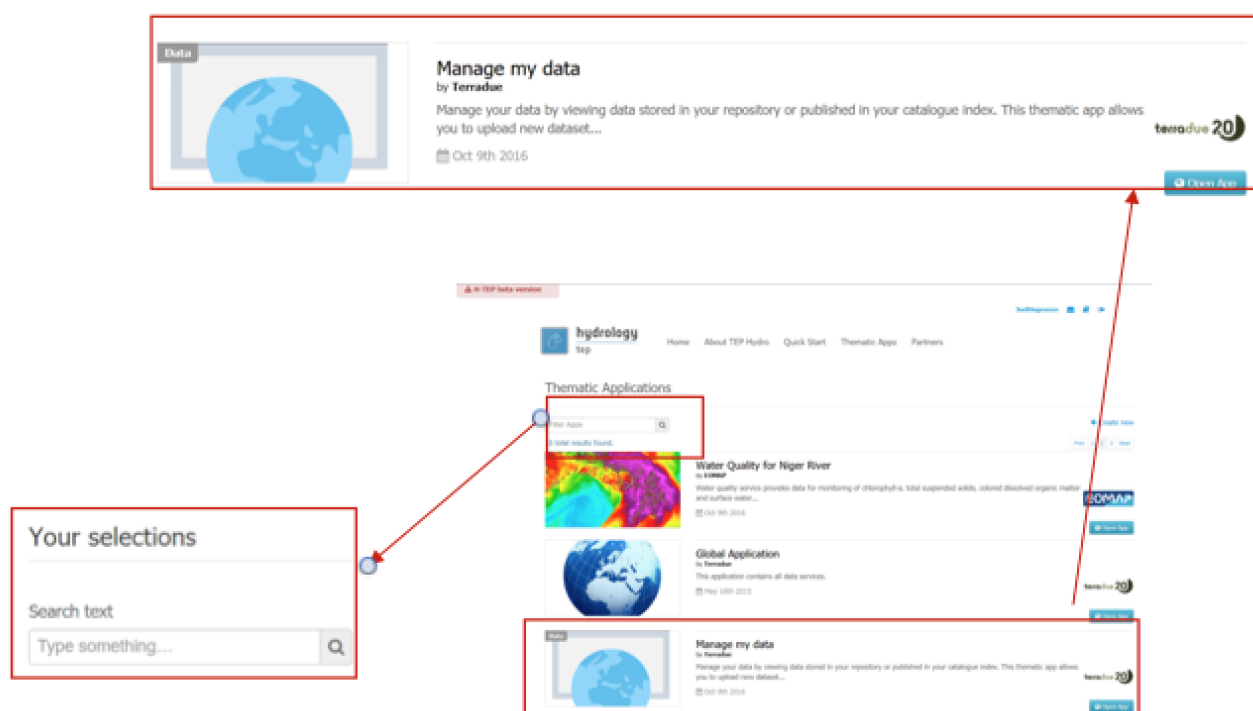
Figure 3. Manage My Data

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.

I want my processing job to be executed efficiently such that, where possible, it is executed close to the data. I want the platform to establish federation relationships with other peer platforms to support this principle, including the platform → platform deployment of processing services on-demand.

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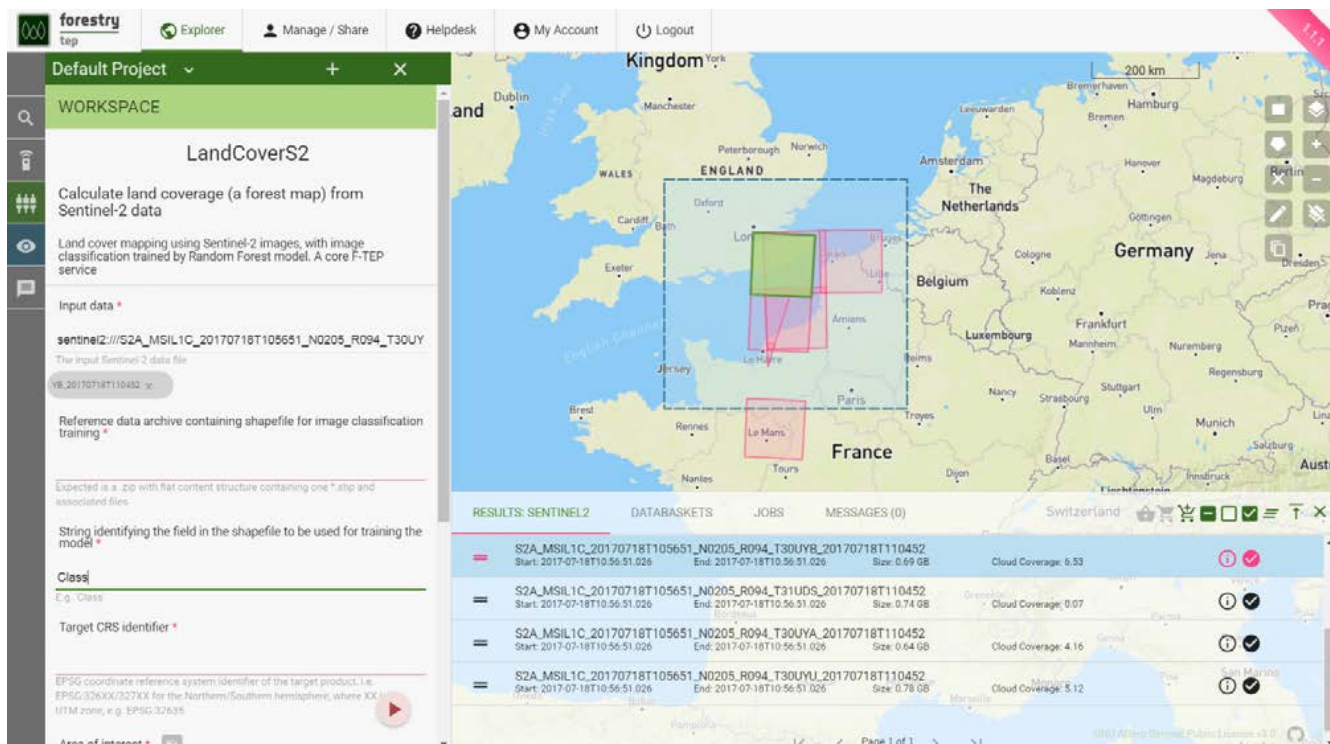
Figure 5. Processing Discovery - Sentinel-2 Land Cover processor (https://f-tep.com/app/)

[Figure 6](#) illustrates the visualisation of processing results, from the Forestry TEP platform.
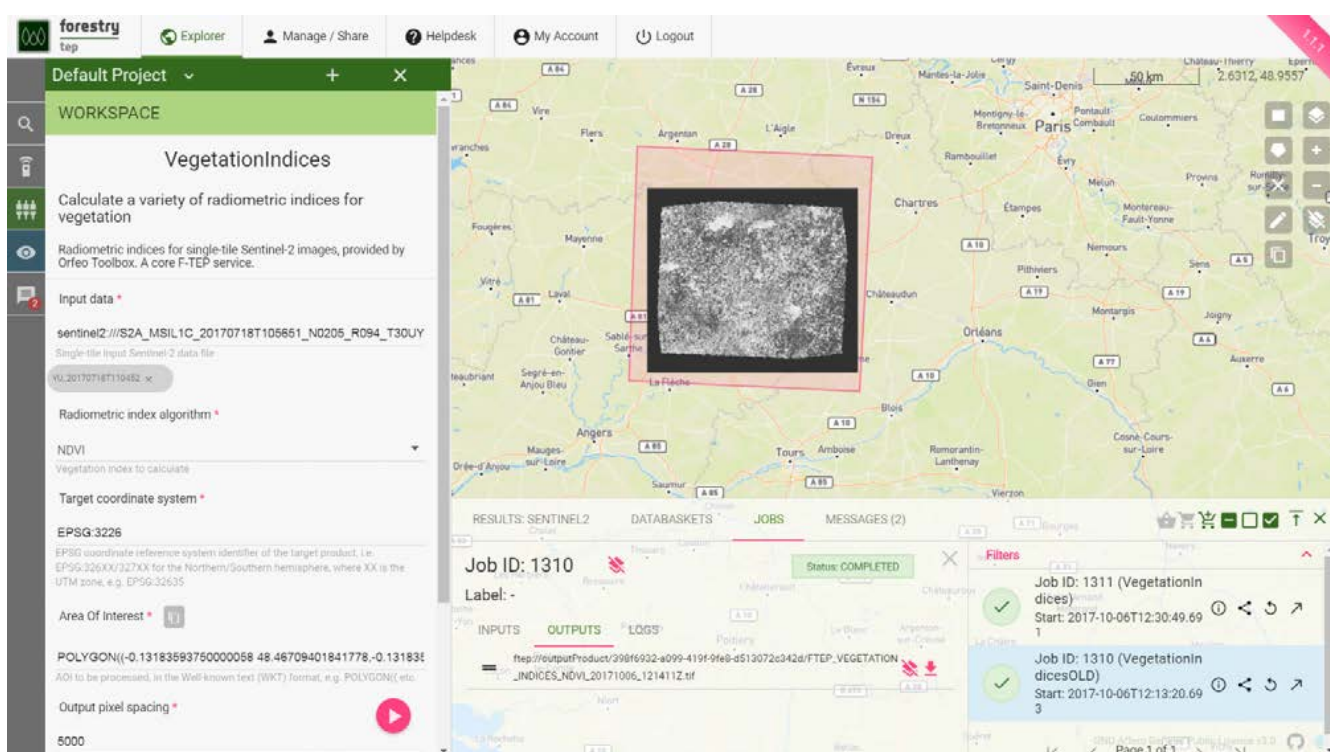
Figure 6. Processing Results (https://f-tep.com/app/)

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. The EP 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 of 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 determines the most appropriate platform, including itself and its federated peers, at which the processing should be conducted in order to execute the procesing 'close-tp-the-data'

23. The EP estimates the cost and duration of the processing and checks the Consumer has enough resources to execute the processing

    a. As required the EP establishes the estimated cost with the remote peer selected for the computation

24. Consumer is presented with the cost/duration estimation and confirms the processing

25. In the case of remote execution at a federated peer...

    a. The EP deploys the processing application to the remote peer selected for the computation

    b. The EP initiates the exection at the remote peer

    c. The EP monitors the status of the processing to provide ongoing user feedback

26. Consumer monitors the status of the processing (%completion, execution logs)

27. When the processing completes successfully the *Processing Results* are made available to the user in their *Workspace*

    a. For remote execution this may include retrieval of the processing results data outputs from the remote peer

28. The Consumer's billing account is updated comensurate with the 'cost' of the processing

    a. As required, the EP reconciles billing with the remote federated peer platform

29. **Exploit Results...**

30. Optionally, the Consumer downloads the results

31. Optionally, the Consumer visualises the processing logs (e.g. for error inspection)

32. 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

33. Optionally, the Consumer publishes their results in the catalogue - specifying all necessary metadata to support discovery

Notes

Data/Processor Selection Order

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

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

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 unecessary 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.

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.

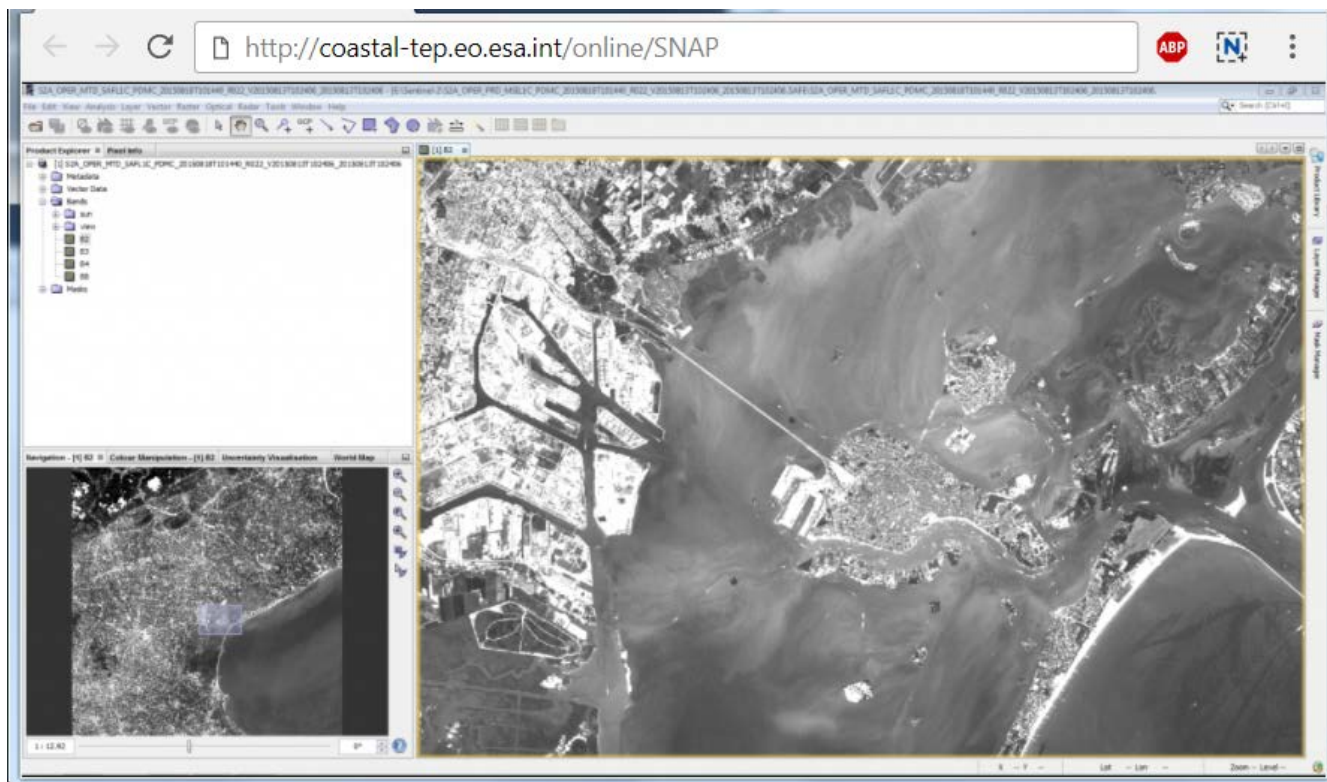*Figure 8. Interactive Application - SNAP running in web browser (https://www.coastal-tep.eu/)*

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 comensurate 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 that 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.
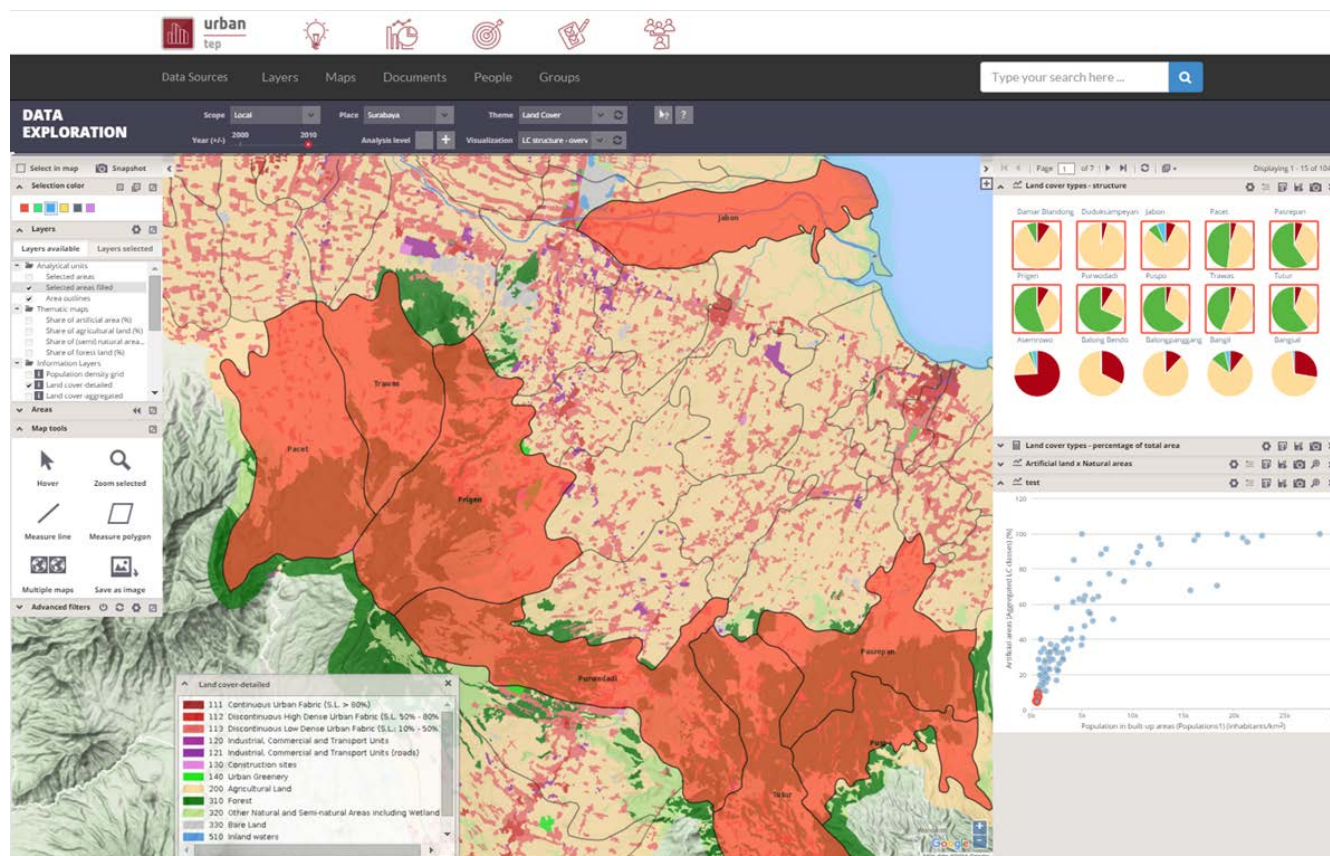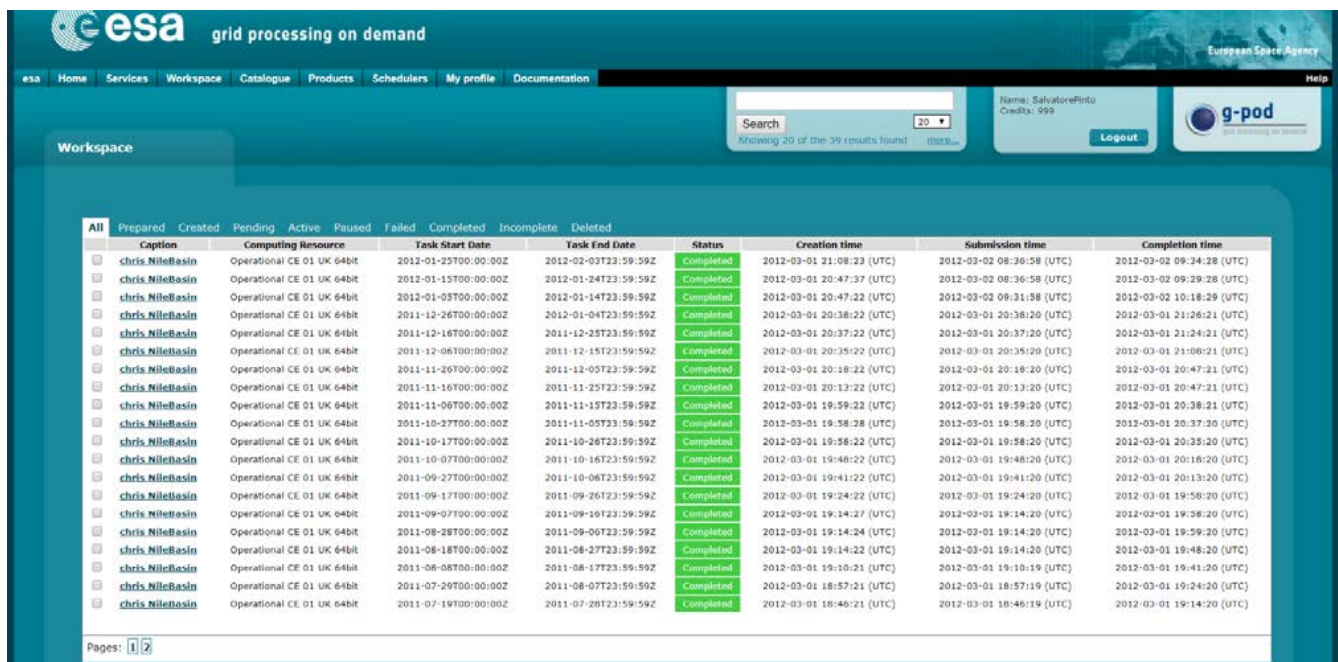

Figure 9. Value Added Products Analyses - PUMA WebGIS (https://urban-tep.eu/)

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.

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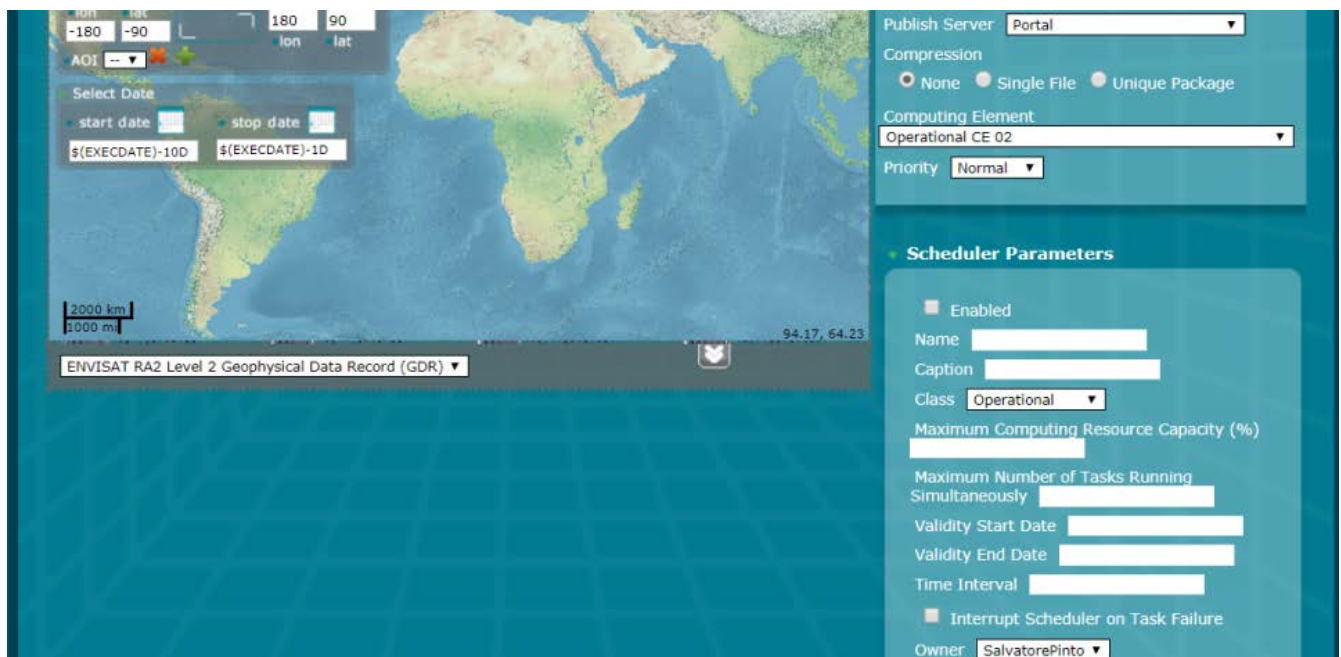
Figure 11. Systematic Processing - Setup of systematic processing from G-POD (http://gpod.eo.esa.int/)

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 occuring 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

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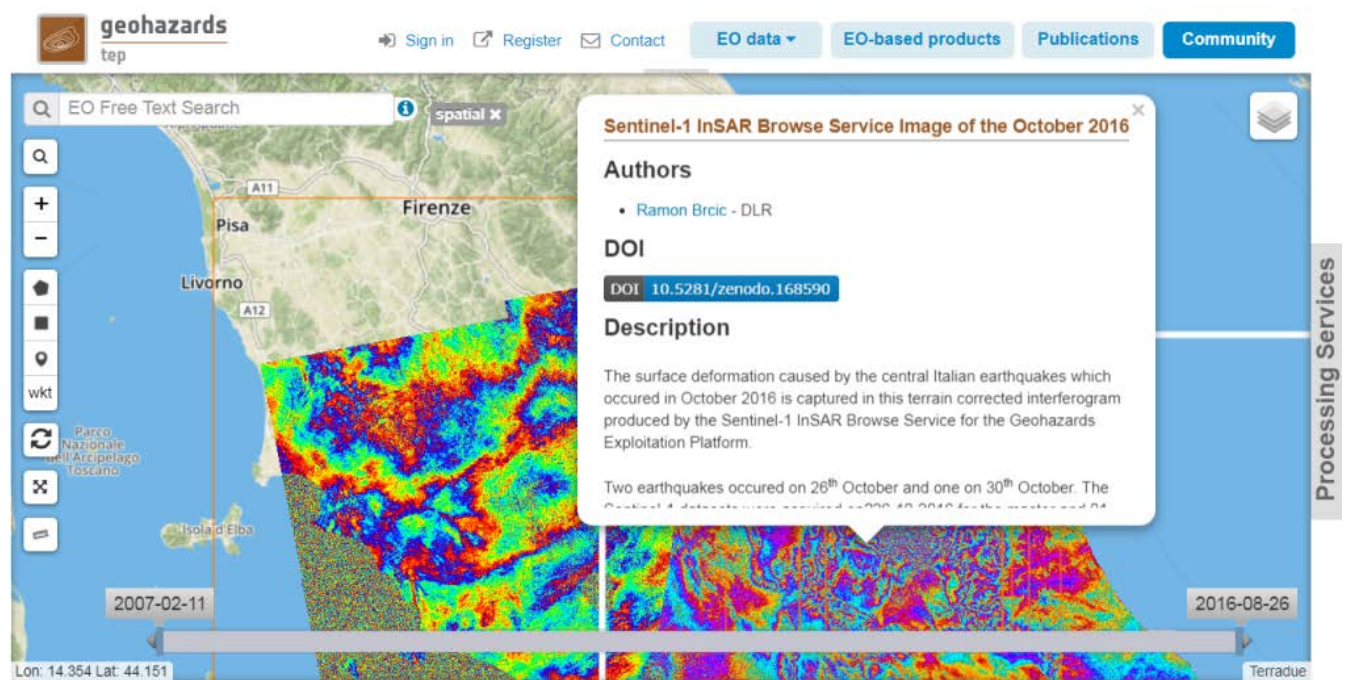
Figure 12. Open Science - Sharing analysis results on the Geohazards TEP (https://geohazards-tep.eu/)

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

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

Collaboration Groups

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 axample, allowing execution of processing services and visualisation of results.

7. The Consumer's billing account is updated comensurate with the 'cost' of the actions performed

Notes

External (non-interative) 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 anciliary information including metadata and processor user manual. The metadata should enable discovery of the service and faciliate 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.
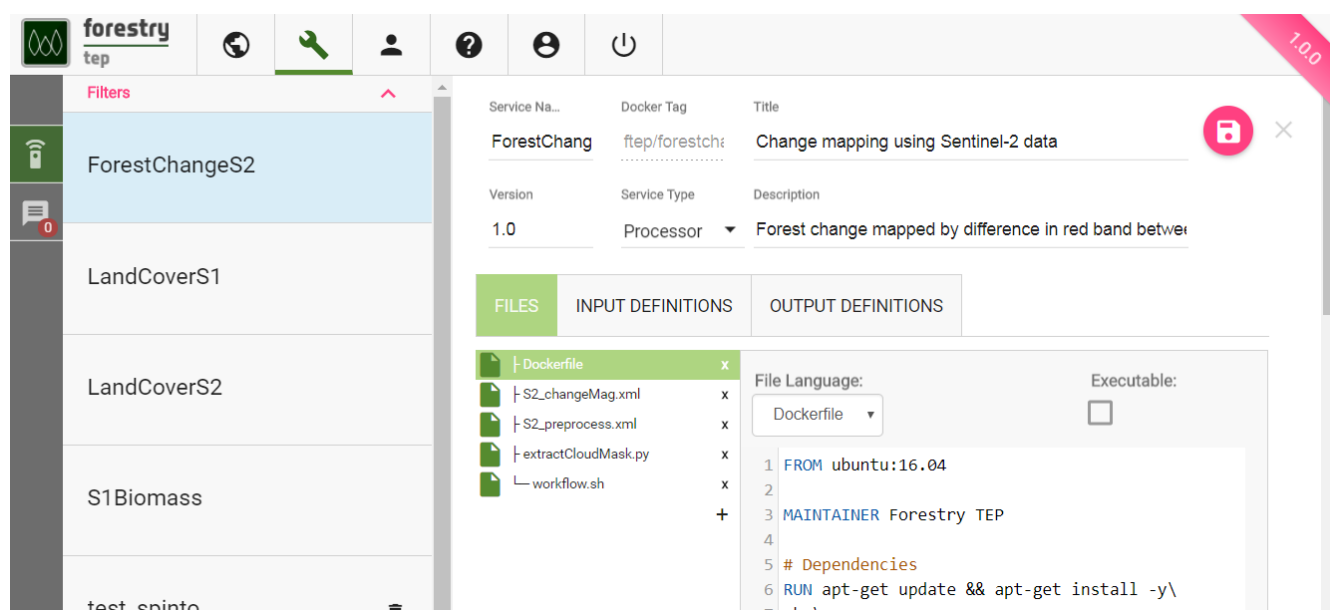
Figure 14. Build Processing Service - Packaging a new processor using Docker image - Forestry TEP (https://f-tep.com/app/)

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 proceing 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 Expert manages the sharing of resources under their administration, by authorizing/revoking access by other platform users/groups

13. The EP checks the integrity of the software and that it adheres to the EP terms and conditions

14. The EP adds the processing service to the Application Catalogue so that is available to Consumers in their discover/browse searches

15. Optionally, EP notifies Consumers about the new service

Notes

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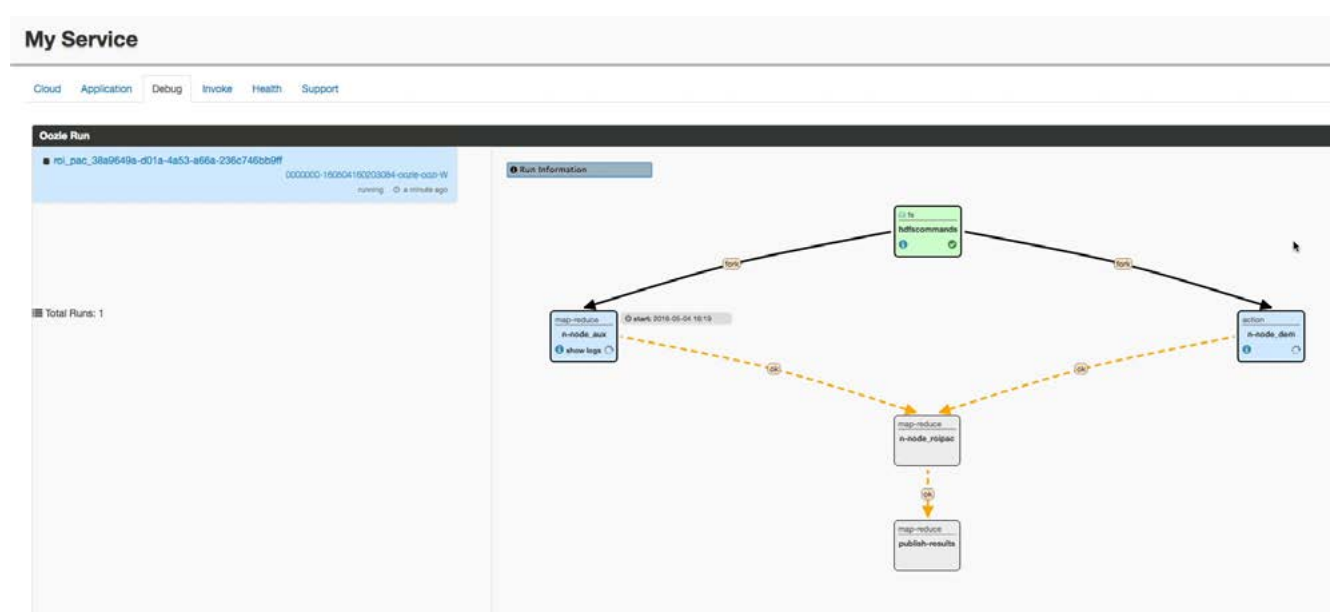
Figure 15. Build Processing Chain - Example taken from the Geohazards TEP (https://geohazards-tep.eu/)

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

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

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 interative 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 interative application, supported by anciliary information included metadata and application user manual. The metadata should enable discovery of the application and faciliate 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 interative 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.
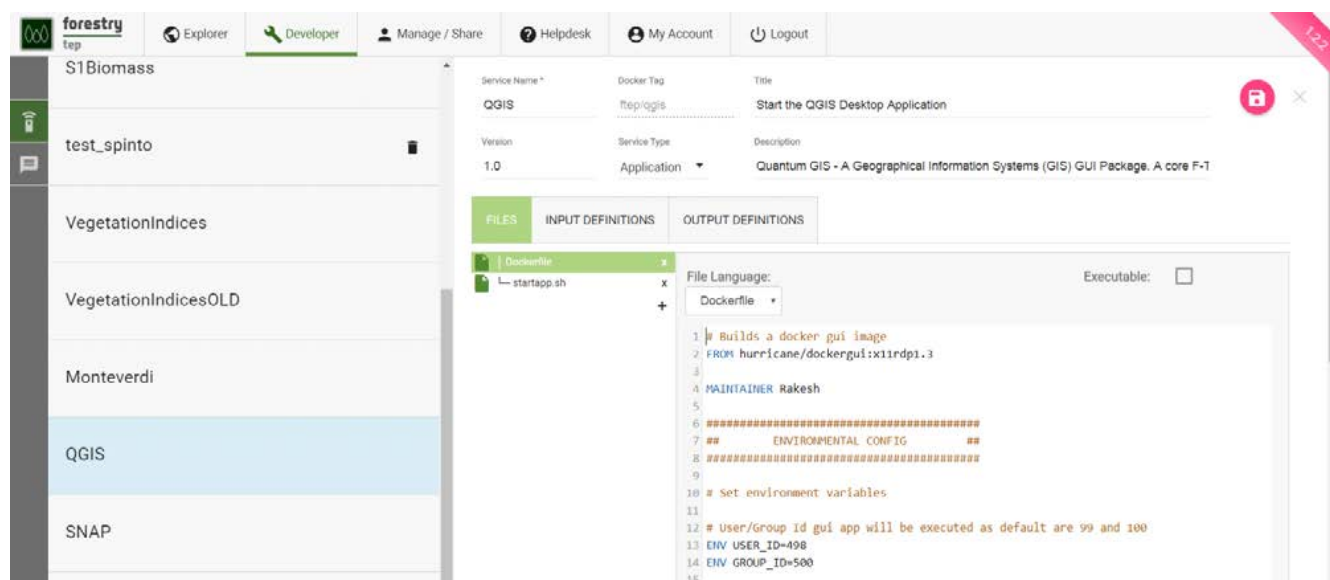
Figure 16. Build Interactive Application - Packaging a new application using Docker image - Forestry TEP (https://f-tep.com/app/)

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.
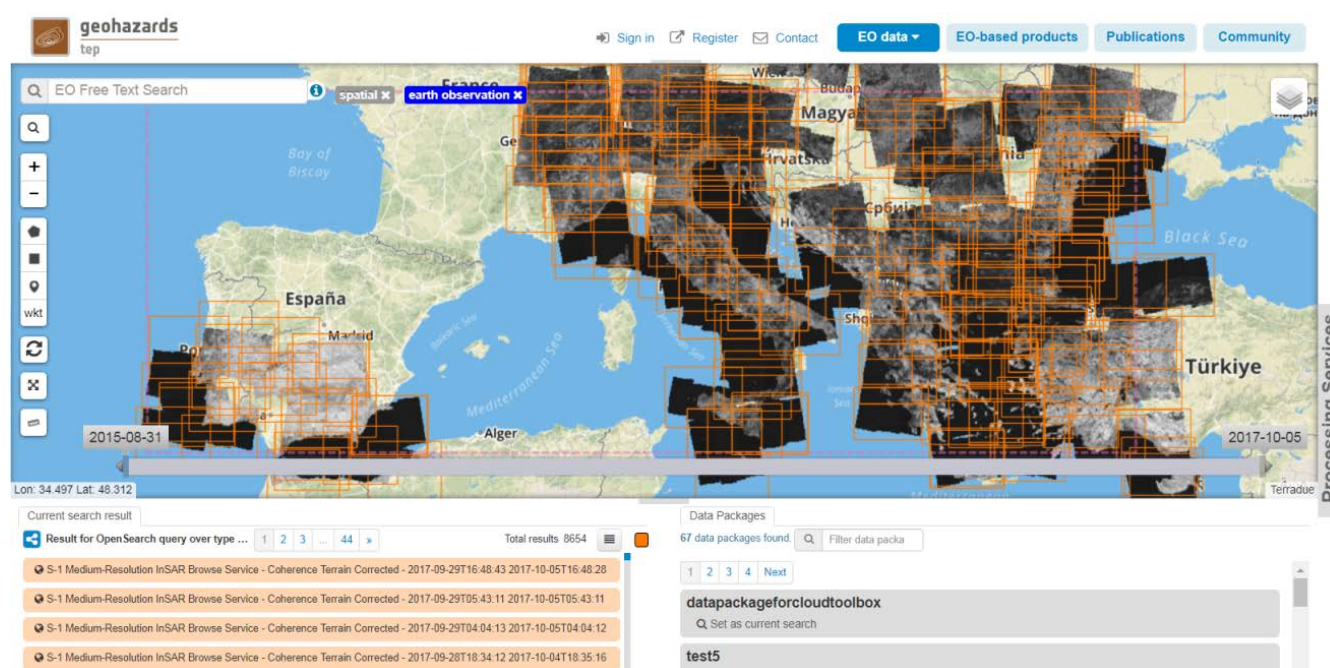
Figure 17. New Value-added Product - Systematic publication of value-added products - Geohazards TEP (https://geohazards-tep.eu/)

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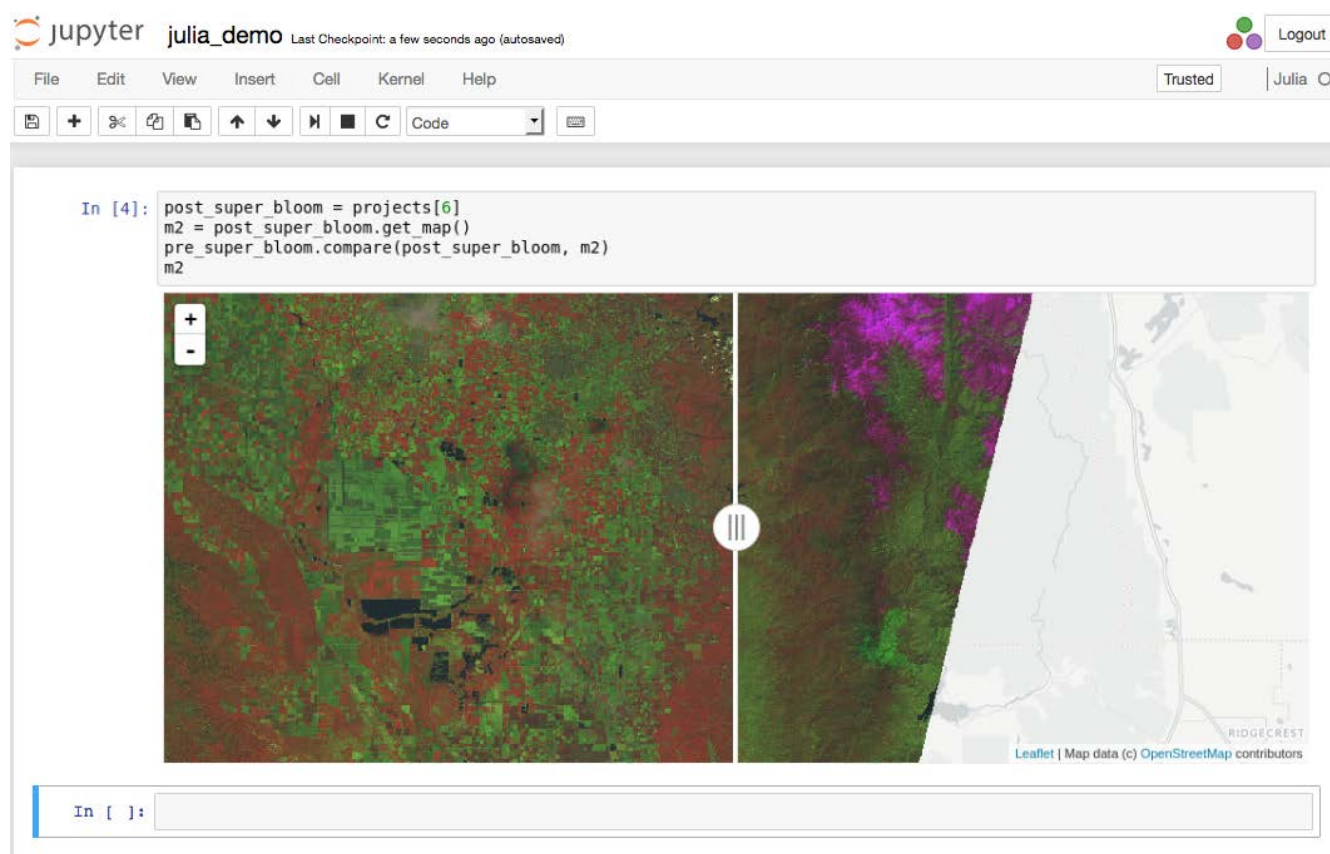
Figure 18. Web Development Environment - Example mocked-up with images from Raster Foundry (https://www.rasterfoundry.com/)

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 environemnt, 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

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

Consumer performs Open Science
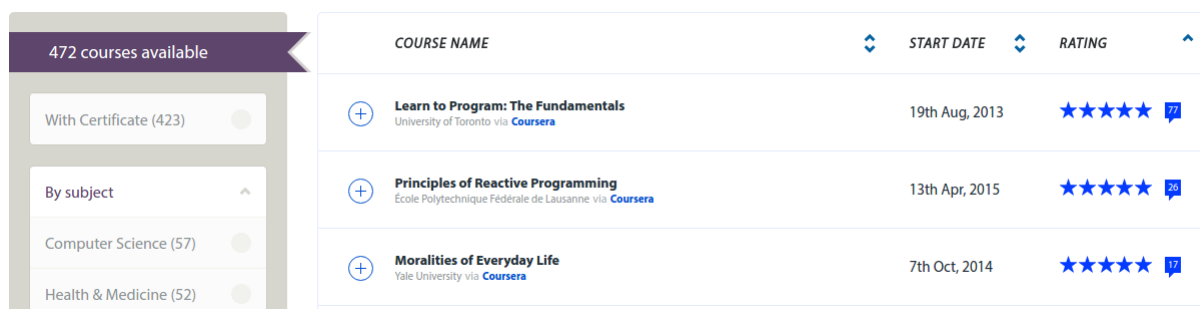
Since the Expert's saved work might for the basis for a *Reusable Research Object*

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.

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

2.17. User uses federated commercial services which he has contracted for directly

As a User I want to buy a licence directly from a provider of data or processing services (not necessarily published on my home platform) and then access this data / service via my home platform. I also wish to directly buy compute capacity from a federated platform and perform part or all of my processing near to the data on this other platform.

In this use case the complexities of sales, billing and licensing are handled outside the federation of exploitation platforms. The user obtains a licence or a remote platform account, tied to his organisation or user ID(s), without interacting with the home platform.

The platform federation must then authorize and report his use of licensed Resources and compute resources. Involved is the User, the Licenser (copyright owner or agent authorized to issue licences), the host platform (in which the resource being accessed is originally published) and the home platform.

In this description the User is using a licensed Resource and compute facilities at a single federated platform. However, many other similar cases are possible - for example, a licensed processing service may be transferred from platform 2 to platform 3, work on data present at platform 3 and then return it to the home platform for visualization. Alternatively, only freely available or user-supplied data and processing services may be used with only compute capacity being paid for, or a remotely catalogued licensed processing service may be transferred to the home platform to run.

---

1. User discovers a commercially licensed Resource using federated search

2. User contracts for the Resource licence directly from the Licenser on the licenser's web site, supplying his EP user ID(s) or an account/organization identifier

3. User contracts with a federated exploitation platform, the host platform, where the data he requires is hosted, and agrees to pay for compute costs

4. User accesses the resource (maybe as part of a wider processing chain) using his home platform, which may involve federated access. The user must specify that the host platform should be used for particular parts of the processing.

5. User authorizes the home platform to access the User's compute resources account at the host platform.

6. Any estimate presented includes the cost of licenses, of host-platform compute servers and of network transfers.

7. Part of the User's processing occurs on the host platform, using data hosted there, and the result is returned to the home platform.

8. The host platform authorizes his use of the Licensed Resource by making a request to an API endpoint operated by the Licenser

9. Processing or data data access may be subject to re-authorization after a time or volume specified by the Licenser

10. The User is billed directly for Resource use by the Licenser and compute capacity use by the

host platform

11. The use of the data or service is reported to the Licenser

Alternative Flow: Using the Licensed Resource on the home platform

The User may buy the licence directly from the Licenser even though the processing is completed on the home platform. The Licensed Resource may need to be transferred to the home platform or may already be present.

Alternative Flow: Using a local Resource on the host platform

The User may have their own processing service or data on the home platform. This may then be transferred to the host platform, where (additional) data required for processing may be present. The User pays for the remote compute costs but no licence costs are involved.

2.18. User uses commercial services from another platform and pays using his platform account (bilateral clearing)

As a User I want to access remote Resources or compute via my home platform and pay via my billing account with this platform. As a Resource Licenser I want to publish my resources via my own platform (the 'host' platform) and receive payment via my billing account with my own platform.

This use case covers several behaviours, which may occur in combination, which require payment to the owner or customers of another EP. Some examples:

- Some commercially licenced data or processing services are catalogued in the host (remote) platform. The data and/or processing service is copied to the home platform to be used in a local processing chain (which may also involve local data).

- A processing service created by the User is transferred to the host platform and runs on data hosted there, using compute resources physically present there. The result is copied back.

The EP must determine where to execute code and when to transfer data. This may depend on estimated data sizes, estimated processing service output size, cloud costs in different locations, User and Licenser settings and commercial arrangements. Different parts of a single chain may happen in different locations if this is most efficient or the only permitted option.

This use-case assumes a network of agreements between exploitation platforms. Each platform has (or may have) an 'account' with each other platform.

Some important notes on the arrangement:

- The home platform always bears the credit risk that its User does not pay. It must still pay the host platform. This is because the home platform is most able to assess the User's creditworthiness and to pursue payment.

- The host platform always bears the credit risk that the home platform does not pay. It must still pay the Licenser.

- The home platform bears the exchange-rate risk if the two platforms use different currencies.

- Costs may simply be reported between platforms. It's possible that no other communication will happen. This permits caching and the transfer of commercial processing services between platforms for local execution, but also requires sufficient trust between platforms. Auditing may be required.

- Either platform can refuse the purchase for risk reasons, but a Licenser cannot.

- Both platforms will wish to charge for this risk and for processing costs.

- It requires $O(n^2)$ agreements between platforms for any User to be able to use any Resource or EP. Licensers may need to publish on multiple platforms and Users may need to open accounts on multiple platforms if an inadequate network is achieved.

1. Licenser publishes a commercially licenced resource via the host platform (his home platform) and configures pricing information in the host platform's currency.

2. Host platform owner configures and publishes pricing information in the host platform currency.

3. User begins accesses involving remote compute or Resources using his home platform's facilities. The EP computes an execution plan (including the location of any data and processing used and any network transfers) and presents estimated costs. Estimated costs include licensing costs, remote and local compute costs and platform fees. The User is required to agree to the terms of the licences involved and the pricing structure.

4. Immediately prior to access occurring the home platform authorizes an inter-platform debt or debts:

   a. The home platform checks that a contract between it and the host platform is in place and in good standing.

   b. A debit to the User's account (a debt to be later owed by the User to the home platform) is authorized by the home platform, after assessing creditworthiness and fraud risk and optionally 'blocking' some available credit. This may be more than the estimated cost so as to cover inaccuracies or it may be increased later.

   c. The home platform makes a request to the host platform to authorize payment (a debt later owed by the home platform to the host platform).

   d. The host platform checks that a contract between it and the home platform is in place and in good standing.

   e. The host platform authorizes a debit to the home platform's account with it.

5. The home platform proceeds with execution (which may involve delegation to other platforms, transfer of data or only locally cached resources).

6. The home platform reports any local licensed Resource use to the host platform and the true cost is confirmed (which should be in line with the pricing but takes in to account repeat access - eg, repeated use of a satellite image for which a licence only needs to be bought once), clearing part of the authorized payment.

7. The host platform reports any remote Resource or compute use to the home platform.

8. The home and host platforms ensure that further authorizations are obtained or access is blocked if the amount authorized is exhausted.

9. The host platform credits the Licenser's account as the payment is cleared.

10. The home platform records the debits in the User's account.

11. The platforms later reconcile and settle payments, transferring a net amount through a manual or external process.

Alternative Flow: Time-based or Unlimited Licences

Instead of depending on use, a licence to use a Resource may be a fixed price for a certain time or be perpetual (like a traditional software purchase). In this case there is no need to delay clearing payment until after resource access has occurred. The User can request a licence and the billing account is updated immediately. Access to the Resource is then possible across the whole federation.

Alternative Flow: Direct Licence Payment via Host Platform Account

Instead of buying a licence using the home platform a User may have bought or buy a licence through a User account at the host platform. This is possible even if no inter-platform payment agreement is operational between the two platforms. However, this only works if the price is fixed (such as a time-based or unlimited licence).

a. EP determines that the User cannot buy access via the home platform but can do so via the host platform

b. EP provides a link to the host platform to do this

c. EP provides access to the licensed resource once the User has bought a suitable licence

Alternative Flow: Access/billing Triangles

A user on platform A (the home platform) could run a processing chain which includes processing on platform B (the compute-host platform) using a licensed Resource with volume-based pricing which is hosted on platform C (the resource-host platform).

a. The user begins processing, as above.

b. The home platform authorizes inter-platform payments to platforms B and C.

c. Execution begins and compute occurs on platform B.

d. Platform B clears payments as compute continues.

e. Before using the licensed Resource hosted on platform C (which may involve cross-platform transfers or may be cached), platform B requests access using the delegated context (user, original platform and overall request).

f. Platform C accepts the request and returns whatever limits apply.

g. Platform B uses the licensed Resource from platform C and reports its use to it.

h. Platform C clears payments from platform A, up to the amount authorized.

i. If the amount authorized is exhausted then platform C refuses additional use when requested from platform B. Platform B communicates this to platform A and, depending on the response, tries again (more payment authorized) or denies access, which may result in processing failing.

TODO: oAuth, or simialr, authorization of remote compute?

2.19. User uses commercial data or processing services and pays using his platform account (via clearing house)

As a User I want to use commercially licensed resources via my home platform and pay via my billing account with this platform. As a Licenser I want to publish my resources via my own platform (the 'host' platform) and receive payment via my billing account with my own platform.

Unlike the previous one, this use-case assumes that each platform forms a contract with a clearing house and payments are cleared via it. Potentially, the clearing house performs credit checks on participants and guarantees accepted payments, depending on the commercial model. Using a clearing house would ensure full interoperability between platforms within the system.

Some important notes on the arrangement:

1. The home platform always bears the credit risk that its User does not pay. It must still pay the clearing house. This is because the home platform is most able to assess the User's creditworthiness and to pursue payment.

2. EITHER the clearing house always bears the credit risk that the home platform does not pay, OR the clearing house provides technical services and payment netting, but passes on default risk to participants in an agreed way. It may require security to be posted or a minimum balance to be kept.

3. The host platform always bears the credit risk that the clearing house does not pay (but this should be made very unlikely). It must still pay the Licenser.

4. Either platform or clearing house can refuse the purchase for risk reasons, but the Licenser cannot.

5. Both platforms and clearing house will wish to charge for this risk and for processing costs.

A clearing house also provides a point where the rules of the system can be established and enforced (eg, a requirement for non-discrimination between platforms) and technical compliance tested. It could also help permit standardization of contracts and terms and reduce the effort required for credit control and audit of participants compared to performing this for all cooperating bilateral pairs.

Chapter 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 stradle 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		
EP customises platform resources based on user identity and data	x		
EP can ban users identified by administrators	x		
User deletes their account and their personal data is removed	x		
User registers	x		
User updates account details	x		

3.2. Consumer discovers and visualizes products

Note: to avoid repetition, this section defines some functionality which is shared between Products and other types of Resource, such as Processing Services.

Platform Function	UM	P&C	RM
The platform catalogues Resources			x
Products have Resource functionality			x
Users can search the catalogue for Resources using free text, AoI/time where applicable and facets			x
The EP provides a browser-based UI for performing searches for Resources and viewing results (but this need not be one single UI for all kinds of Resource)			x
When showing information about a Resource, a User can view detailed metadata and documentation			x
The EP can filter or mark Resource search results and information display based on user account data (eg, licences purchased, academic vs commercial user) and Resource access rules / ownership	x		x
The EP can refuse access to some or all aspects of a Resource based on the User and Resource (eg, show summary or search result but not allow data access, documentation access or visualization)	x		x
The catalogue contains, for each data Product, costing data (licence prices, data to estimate data sizes/processing times)			x
Users can view the T&Cs associated with Resources			x
The EP permits authorized Users to add and update T&C text and metadata (which may be shared across multiple Resources)			x
The EP permits authorized Users to update Resource access rules and settings (licences required, descriptions, costing data, etc)			x
The EP can request, record and display acceptance of identified T&Cs by a User (noting that a User may have to accept T&Cs more than once if he represents multiple organizations)	x		x
The EP can prevent access to Resources when T&C acceptance is required and not completed and can prompt the user to do this (note: this access may happen inside a UI, API or executing processing service)	x	x	x
Users have a 'user workspace' and can 'save' Resources in to it			x
EP can cost proposed access and use of a Product (which may involve estimating data size, processing time and licencing costs)		x	x
A User can view an interactive visualization of Products, considering User parameters (eg, location or band selected) and the dataset (eg, preferred projection, appropriate colouring, key and units, etc)		x	x

Platform Function	UM	P&C	RM
An authorized User can specify a Product's default visualization and parameterization that can be available in search results and used to quickly assess the data's relevance to the user		x	x
A User can download visualization images		x	
The EP presents a site map and per-Resource structured data for external search engines (eg Google, Google Dataset), but only for Resources hosted on the platform and not accessible only via federation			x
A User can download Products/parts of Products, subject to access constraints and configured costs			x
Use of Resources (including search, metadata display, costing, download and execution, but not modification of resources) still functions in as high-performance and uniform a way as possible even when the Resources are located on other platforms within the federation (and access and ownership rules are still enforced)	x	x	x TO DO: Define 'organization'

3.3. Consumer uploads data to their workspace

Platform Function	UM	P&C	RM
Each User has a 'user workspace' which displays information about previously saved or uploaded data			x
The EP permits uploaded data to be used like any other data, including combining it with other data			x
A User can upload data using a browser			x
A User can upload data using a resumable and scriptable method (eg HTTP PUT or ftp)			x
The EP restricts uploads based on user rights, including any applicable quotas or budget constraints			x
The EP provides a means to add metadata to uploaded data (as much metadata as possible should be guessed from the data, eg title, licence, variables, ... from NetCDF)			x
Uploaded data and workspace-saved data can be visualized, similar to search results		x	x
Uploaded data and its catalogue entries can be published, making it available for use by other users, with a requirement that all necessary metadata for discovery is specified			x
The EP can hide from public access or remove (and restore) published datasets which are alleged to breach copyright			x
The EP supports compliance with the Copyright Directive article 17 requirements (eg, via upload and data federation filters to auto-detect copyright violations)			x

3.4. Consumer discovers and executes On-demand Processing Service

Platform Function	UM	P&C	RM
Processing Services have Resource functionality (search, T&Cs, etc)			x
The catalogue contains costing data for Processing Services (eg, licence costs, data to estimate data sizes/processing times, special requirements such as GPUs)			x
A User can parameterize processing by entering required parameters and selecting compatible input data using the catalogue search			x
A User viewing Products can search for / select applicable Processing Services, in which case input data will be pre-specified when arriving at the parameterization			x
The EP can establish federation with remote peer platforms for the purpose of invoking processing platform→platform	x	x	
The EP can cost proposed processing requests (which may involve estimating data size, processing time and licencing costs)		x	x
The EP can obtain cost estimates from remote federated peer platforms for intended processing jobs		x	x
Users can initiate processing once all required parameters and inputs are specified		x	
The EP can delegate processing to another federated peer platform in the interests of processing 'close-to-the-data'		x	
The EP can reject processing if quotas, budgets or credit-control constraints are reached (as appropriate to its pricing model)		x	
The EP can deploy a processing service on-demand to another federated peer platform		x	x
A User can save parameterizations of a Processing Service in the user workspace, not just a reference to the service			x
A User can view information about submitted processing requests (eg, start time, cost incurred, errors encountered, log output, proportion completed if available, result information and link(s))		x	
The EP can monitor remote execution of a processing request it made to a federated peer platform		x	
Processing service outputs can be published (as a Product), subject to compliance with licensing by the owners of the data used and the entry of required catalogue information			x
The EP ensures all processing results are made available to the requesting User (including for steps executed on remote federated peer platforms)		x	
Users' billing accounts are updated as processing continues and completes		x	
Users can download output data		x	

Platform Function	UM	P&C	RM
Users can visualise the output data (and download visualisations) similarly to other data within the platform		x	

3.5. Consumer discovers and executes Interactive Applications

Platform Function	UM	P&C	RM
Interactive Applications have Resource functionality			x
An Interactive Application may require a dedicated VM (or multiple VMs) or other cloud resources which run for the length of an Interactive Application session		x	
If an Interactive Application uses a dedicated VM then it can include a GUI environment (eg X Windows) which the user interacts with remotely (eg using remote desktop or VNC)		x	
An Interactive Application can provide or consist entirely of a web app (which may be a one-page JavaScript app+API, may be server-side generated or may require no server-side resources at all except for static files and access to the platform API)		x	x
If the application has a web app then the platform UI will facilitate the display of this app (eg by providing a link and/or allowing it to be used embedded in the portal UI)		x	x
An interactive application may open ports (eg, an ssh or HTTP port) so that a user can interact with it using a CLI/console-like interface or using a client application		x	
An interactive application can access the platform APIs (eg to run Processing Services) under the delegated credentials of the Consumer	x	x	
The catalogue contains costing data for Interactive Applications (eg, licence costs, size of dedicated VMs required)			x
The EP can cost interactive application use, if possible and appropriate to the application (eg, price per hour)		x	x
The EP can stop applications if quotas, budgets or credit-control constraints are reached		x	
A Consumer can use a data search to find compatible input data for an interactive application, if required by the application			x
A Consumer can use their workspace saved data to specify data for an interactive application			x
An interactive application can request data and/or processing as required by its functionality		x	
A Consumer can save screenshots and other outputs from an application to the workspace as appropriate to the application		x	x
A Consumer can download screenshots and other outputs from an application		x	
A Consumer's billing account is updated as the application is used		x	
A Consumer can publish application output data to the catalogue just as any other data from the workspace can be published as a Product		x	x

Platform Function	UM	P&C	RM
A Consumer can resume an earlier interactive application 'session' by choosing it from a list		x	x
A Consumer can terminate an Interactive Application session		x	

3.6. Consumer analyses value-added product

The functionality in 'Consumer discovers and executes Interactive Applications' is sufficient to cover this use-case as well.

3.7. Consumer executes Bulk processing

Functionality listed for 'Consumer discovers and executes On-demand Processing Service' is also relevant but
is not duplicated here.

Platform Function	UM	P&C	RM
Consumers can create a 'bulk data collection' and add multiple data specifications to it (which should be compatible with a chosen target Processing Service)		x	x
Consumers can add data using data search or using workspace saved data		x	x
Consumers can specify an input data collection when parameterizing a processing request (in place of an individual piece of data)		x	
Consumers can specify an output data collection when parameterizing a processing request		x	
Bulk data processing can be split by the platform in to multiple parallel requests		x	
Functionality for on-demand data processing, such as viewing logs and downloading outputs, can be applied to bulk processing and data collections		x	
Consumers can resubmit processing requests after correcting errors - output data for unmodified completed requests will not be regenerated unnecessarily		x	

3.8. Consumer executes Systematic processing

Functionality listed for 'Consumer discovers and executes On-demand Processing Service' and 'Consumer executes Bulk processing' is also relevant but is not duplicated here.

Furthermore, functionlity listed for 'Expert user builds new value-added products' is also relevant to automatically publishing the output of systematic processing.

Platform Function	UM	P&C	RM
A Consumer, having parameterized and specified inputs for a processing service invocation, can select triggering conditions which can cause it to run repeatedly		x	
The EP can cost the proposed systematic processing (which may involve estimating data size, processing time and licencing costs)		x	x
The platform can ensure that the Consumer has sufficient credit to run systematic processing both at submission time and run time		x	
The platform can recognize triggering conditions a run processing, passing information about the trigger to the processors (eg, which new data is available)		x	
The platform can trigger systematic processing according to a schedule		x	
The platform can trigger systematic processing when new data becomes available		x	
The platform can trigger systematic processing when triggered by an external event		x	
The platform can notify the Consumer when systematic processing completes or fails		x	
The Consumer can see status information on systematic processing, such as execution logs, requests running and percent complete		x	
The Consumer can use other functionality targetted at output data (eg visualisation and publishing) with systematic processing outputs		x	x

3.9. Consumer performs Open Science

Platform Function	UM	P&C	RM
A Consumer can create a 'research collection' of results, links to resources, processing parameterizations, visualisations, documentation, etc.			x
Items included in such a collection can be directly included or can be links, including 'Linked Data' (in the W3C sense) and links to scientific publications			x
A Consumer can share such a collection (eg, via URL), providing he has permission to share everything directly included			x
A Consumer can restrict to specific collaborators and with specific access rights (eg, read vs write)			x
The EP, optionally at the discretion of the sharer, notifies those with whom a collection has been shared			x
Information relevant to creating 'research objects', particularly provenance information for the results of data processing, is maintained automatically by the EP as much as possible		x	x
A Consumer can add information about research collections and elements of them relevant to creating research objects, such as ORCID IDs			x
Resources published in the catalogue contain details of their provenance			x
A Consumer can create a reusable research object from a research collection, in compliance with (a) research object standard(s)			x
A reusable research object will be a snapshot of the research collection at the time it was created			x
A reusable research object may contain result data and/or intermediate data			x
A Consumer can publish a reusable research object in the catalogue			x
A Consumer can register a DOI for a research object or any other item he has registered in the catalogue			x
The EP will ensure that the requirements of the DOI Foundation and DOI Registration Agency are complied with, eg. for long-term availability and immutability of resources published using DOIs			x
A Consumer can search for published research objects in the catalogue			x
A Consumer can view research objects found in the catalogue or via DOI			x
A Consumer can 'unpack' a research object he has access to in to a research collection within his workspace			x
Where possible, processing described within a research collection can be repeated (and may make use of existing intermediate results)		x	
Where randomness is involved in processing and where supported by the processor, the EP shall record any random seeds used and permit these to be reused when processing is repeated		x	

3.10. Consumer accesses EP services with External Application

Platform Function	UM	P&C	RM
The EP provides an API for programmatic access to all suitable platform services except for account management (eg, catalogue search, data download, processing, workspace manipulation, data visualisation, processor and processing chain definition and upload)		x	
The API is programming-language agnostic and accessible to non-professional programmers and scientists	x	x	x
API clients are provided for popular programming languages within the EO domain, eg Python	x	x	x
An API authentication method suitable for use with external applications (such as mobile apps and web apps) is provided, eg oAuth	x		
An API authentication method suitable for use with scientist-written ad-hoc scripts is provided, eg API keys	x		

3.11. Expert user builds new processing service

This use-case is linked to 'Expert develops with interactive development environment' and may rely on some of its functionality.

Platform Function	UM	P&C	RM
The EP can accept, use and publish processing services development in supported container formats (eg Docker and/or Singularity) and following any additional requirements (eg, built-in processing service descriptor files)		x	
The interactive development environment can be used to create containers and submit them as processing services to the catalogue		x	x
The UI and API can be used to upload pre-created containers and submit them to the catalogue		x	x
The EP supports container development workflows by providing a convenient way to deploy and test a processing service via UI and API		x	
An Expert User can publish a processing service so that others can find it in the catalogue, providing all metadata requirements have been met			x
An Expert User can specify processing service documentation and metadata, including information on its inputs and outpus to assist in building processing chains			x
The platform can perform checks on processing chains being published to ensure that they're valid and comply with platform requirements		x	
The platform can understand processing service versions/releases, present information about releases to users and optionally notify users when a processing service is updated			x

3.12. Expert user builds new processing services chains

Platform Function	UM	P&C	RM
An Expert User can build a processing chain by specifying Processing Services (including other chains, and services on other platforms), data flows, parameters, etc		x	
The EP supports chain development workflows by providing a convenient way to deploy and test a processing chain via UI and API		x	
An Expert User can publish a processing chain as a Processing Service so that others can find it in the catalogue, providing all metadata requirements have been met			x
An Expert User can specify processing chain documentation and metadata, including information on its inputs and outpus to assist in building new processing chains			x
The platform can perform checks on processing chains being published to ensure that they're valid and comply with requirements (eg, that all of the Processing Services used are sufficiently accessible to all permitted users of the chain and that T&Cs are complied with)		x	x
The platform can understand processing chain versions/releases, present information about releases to users and optionally notify users when a processing chain is updated			x
The platform permits the separation of input data and parameters which can be specified by a later chain user from the chain definition and from fixed processing service parameters or data which must not be changed		x	
The EP shall have any necessary support for including processing chain and chain execution provenance information in reusable research objects (for example, using the wfdesc and wfprov ontologies)		x	

3.13. Expert user builds new interactive application

Platform Function	UM	P&C	RM
An Expert User can create Interactive Applications which include containers (eg Docker or Singularity images) which must be running through the application session and use the UI or API to add them to the catalogue		x	x
An Expert User can create Interactive Applications which include or consist of static files (particularly JavaScript, CSS and HTML) which will be served directly by the platform without the need for dedicated resources like VMs or running containers		x	x
An Expert User can create Interactive Applications which are implemented using server-side infastructure which is not managed as dedicated resources within an application session - for example, an Interactive Development Environment consisting of JupyterHub and Kubernetes cluster all shared between all users. The application will still be able to use delegated credentials for data access.	x	x	

3.14. Expert user builds new value-added products

Platform Function	UM	P&C	RM
An Expert User can create a Product collection, specifying all necessary metadata for discovery in the catalogue			x
An Expert User can define bulk or systematic processing whose output will automatically be added to the product collection on completion (if succesful), resulting in it being published with the same permissions and visibility as the collection		x	x
Consumers can subscribe to a Resource or collection and will be notified when it changes (eg, new data is available, a new version of a Processing Service is release)			x

3.15. Expert develops with interactive development environment

Platform Function	UM	P&C	RM
An Expert User can access interactive development environments published in the catalogue, subject to access permissions		x	x
Development environments suitable for developing in Python and R will be provided at a minimum		x	
It is not specified whether an interactive development environment takes the form of a dedicated VM, a container, a management system like JupyterHub+Kubernetes, uses a remote UI like Hydrogen, etc - this may vary by environment		x	
An Expert User can import and export code in ways most suitable for the particular language and environment, for example through uploads/downloads, git push/pull or integration with GitHub		x	
Facilities, such as libraries or APIs as best suited to the language and environment, are available within each development environment for accessing platform services, such as data and processing		x	
Access to platform facilities (which may include remote platforms) by the code under development will use the user's credentials and be subject to normal access constraints (eg, T&Cs)	x	x	
The development environments provided contain by default a suitable set of libraries for EO-domain data manipulation and visualisation		x	
Additional libraries can be specified by the Expert User		x	
An Expert User using a development environment can save the code in to the workspace		x	x
An Expert User using a development environment can save data produced in to the workspace		x	x
Development environments and executions will integrate with research object functionality, for example by allowing editing within a 'research collection' context or importing/exporting code from one, and by recording code, data, workflow and execution provenance		x	

3.16. Expert performs training

Platform Function	UM	P&C	RM
An Expert User can define on-line courses, if he has permission to do so			x
An Expert User can link on-line courses to his processing services or interactive applications			x
An Expert User can add content to an on-line course, including documents (eg documentation, tutorials), videos (eg, webinars, lessons or demonstrations), exercises and tests, and define the course structure			x
An Expert User can schedule live video streams			x
A Consumer can find on-line course by searching the catalogue or by following links from related resources			x
A Consumer can join an on-line course, subject to access control and billing constraints and, if configured for the course, its publisher's approval			x
The platform can charge the User's account for course costs as defined by the course publisher, and can credit the publisher's account as appropriate			x
A Consumer can access course content for courses previously joined			x
A Consumer can view live video streams (however, it is not a requirement that they be streamed from the platform and not an external provider)			x
A Consumer is billed for courses via his account according to prices set by its publisher (which may be zero)			x

3.17. User uses federated commercial services which he has contracted for directly

Platform Function	UM	P&C	RM
A Licenser who has published Resources in the catalogue on a commercial basis can specify an API endpoint which will be contacted to authorize access to those Resources or can specify a list of permitted user IDs	x		x
The platform can restrict use of a Resource based on this API endpoint, including time or volume limits after which access must be re-authorized	x	x	x
The platform can report use of resources to their Licensers for billing purposes			x
A Licenser can specify a URL that prospective users will be shown so that they can buy licences			x
The Licenser's sales site can use an authentication protocol (such as OpenID Connect) to authenticate buyers using their EP credentials, and also to avoid a need to log in for users already logged in to the EP	x		x
A User submitting on-demand processing, bulk processing, systematic processing can specify a different federated EP on which the processing (or a component of it, for processing chains) should run		x	
A User submitting processing to run on another EP must authorize the home platform to access his host platform account's compute resources (eg using oAuth)	x		
When calculating estimated costs, the home platform can contact the host platform to request the estimated costs for the remote part of the processing		x	
The EP can execute processing on a remote EP using the requesting user's credentials (note: the container to run may come from the remote platform or a third platform)		x	
The EP can manage processing running on remote EPs - eg, it can display the status and terminate processing		x	
An EP can retrieve processing services or data from any federated EP using the requesting user's credentials			x
When data processing occurs, the inputs and processing services or chains involved can be transferred from other EPs if necessary		x	x
When final outputs are produced by delegated processing run on another EP the User can see them in the local workspace/UI and can choose to transfer them		x	

3.18. User uses commercial services from another platform and pays using his platform account (bilateral clearing)

Platform Function	UM	P&C	RM
A Licenser can configure pricing for Resources in the catalogue in the 'host' platform, subject to access rules			x
A User using the 'home' platform (which may be different to the host platform) can request access to such Resources and will be asked to agree to the pricing, estimated (or fixed) costs and terms and conditions		x	x
Pricing may be time-based (for a licence for the whole Resource) or volume/item-based, with different definitions appropriate for different kinds of Resource (eg, by CPU-time or data volume for processing, by satellite scene or grid square, etc)			x
For volume-based pricing the User will be able to set a limit on the amount he will be billed after which accesses will fail unless the user requests additional access		x	x
A User will not be required to request access and go through payment authorization if the User already has a licence bought using a different platform or directly from the Licenser, providing that no additional payments will be required for use (eg, the licence is time-based)			x
A User will not be required to request access or authorize payment if the User has already completed this and has not reached any payment limits			x
The user will only be able to request federated access to licensed Resources if all necessary platform payment clearing agreements are in place and in good standing to permit purchase, otherwise the user will be told which platform he can use for access			x
If a User cannot request access to a Licensed Resource because payment clearing is not possible or disabled for the resource but a direct purchase via the host platform or Licencer site is possible and would grant access, the home platform provides a link to the host platform or a direct purchase site			x
If a User follows a link to make a purchase from a host platform then an expedited login/registration and purchase process is provided (the user does not need to search for the resource in the catalogue)			x
A User can agree to the T&Cs and prices and, if platform payment is fully authorized, will be granted access	x		
On-demand processing, bulk processing, systematic processing where cross-platform access and payments are possible will be automatically planned for execution using knowledge of Resource locations and estimated sizes and costs (eg, a processing service may be moved to another platform to run close to data). A 'plan' includes the locations of data processing, Resources and the data transfers required. This plan will be used when displaying estimates and for subsequence execution.		x	x

Platform Function	UM	P&C	RM
A User can refuse to allow federated execution of processing, requiring that all remote data and processing services be transferred to the home platform		x	
A Resource owner can specify that the Resource is not permitted to leave its host platform		x	x
When remote execution or the use of licensed Resources hosted remotely occurs the home platform uses platform payments, described below, to arrange for the User to pay the Resource or EP owner		x	x
Platform payments must be authorized by multiple parties, cleared as debts are incurred and subsequently settled when (net) payments are made between platforms		x	
The home platform can assess the User's creditworthiness and prevent platform payment authorization if it feels necessary			x
The home platform can 'hold' some of the credit in the User's account following platform payment authorization, if required by its billing model	x		
The home platform can request Resource licensing platform payment authorization from the Resource's host platform using a defined protocol			x
The home platform can request compute cost platform payment authorization from the compute host platform using the protocol		x	
Each platform maintains a database of platform payments between it and other platforms			x
The host platform can assess the creditworthiness and outstanding net inter-platform balance with the home platform and accept or reject platform payment authorization			x
For volume-based pricing, platforms in either role can limit the amount authorized and expire authorization for credit and financial control purposes. Additional authorization can be sought automatically, providing user-defined limits are kept to.			x
Platforms distinguish the amount authorized from the amount cleared (ie, which is legally owed and for which settlement will be requested). For volume-based Resource licence and compute pricing, settlement of platform payments will only be requested once the resource/compute has been used (and may be 'chunked').			x
For time-based (ie, fixed in advance) licence pricing, the User can request a licence and payment will be immediately authorized and cleared. Access can then proceed.		x	x
A home platform can authorize a payment to a compute platform prior to submitting compute jobs on behalf of its user		x	
A compute-host platform can report actual compute consumption to the home platform, which results in payments being partly cleared		x	

Platform Function	UM	P&C	RM
A compute-host platform can report authorization exhaustion to the home platform, which may respond requesting additional authorization or may request termination.		x	
A home platform beginning execution involving a remotely hosted volume-pricing based licensed Resource can authorize an initial platform payment in advance		x	
A compute platform, on encountering a need to use a commercially licensed volume-based pricing Resource, can request permission from the resource host first, including full context information (the home platform and user and the payment authorization to use)		x	
A resource host can refuse or permit the use of its resource based on extant payment authorizations		x	
A compute platform which has completed Resource use can report this to the resource host, which results in the resource host clearing payments from the home platform (via messaging directly with the home platform). The resource host calculates the true cost of access for this purpose ((eg, repeated access to a single grid square or satellite image may not cost anything extra).		x	
A compute platform whose Resource use is rejected can report this to the home platform (which may be the same platform) which will then decide whether to seek further payment authorizations or to refuse access		x	
When inter-platform messaging is required to perform Resource use or to submit compute jobs the requesting platform will include an identifier for the payment authorization (and other context information such as user and home platform), and the host platform may refuse access if this is insufficient, expired, etc.		x	x
The host platform (according to the Licenser's requirements) can specify whether access must always be made via inter-platform messaging or whether the home platform can transfer and cache data, processing service containers, etc		x	x
The platforms can credit/debit user accounts as platform payment is cleared			x
The platforms can reconcile and report inter-platform debts and outstanding authorizations, flagging any disagreements for manual dispute resolution, so that settlement can occur			x
The platforms can (part-)refund platform payments at the request of the Licenser			x
The platforms can mark payments as refunded following the outcome of manually conducted dispute resolution (eg, following a User complaining that the resource is not as described or does not work)			x
The platforms keep a detailed journal of the payment process for dispute resolution			x
Effective security is applied to inter-platform payment processes to prevent impersonation of platforms and to provide non-repudiation			x

Platform Function	UM	P&C	RM
Account credit given by the host platform to the Licenser will not be reversed if the home platform does not pay; inter-platform payment from the home platform to the host platform will not be reversed if the User does not pay			x
Platforms can be configured to deduct agreed fees from inter-platform and platform-to-Licenser payments, but this may be invisible to the User			x
Pricing settings will specify a currency and all inter-platform requests will use this currency, even if it is not the currency in which the User pays his account			x
The home platform performs currency exchange and calculates and displays the local-currency rates which the User is charged			x TODO: How do we handle organizations and projects which might be licensed as a whole?

3.19. User uses commercial data or processing services and pays using his platform account (via clearing house)

Here it is assumed that a clearing house is commercially possible and provided by an external entity, not by a component specified in this document.

TODO: We must determine if there is any possibility of this happening before including this in a final document, determine if we must develop/specify clearing house functionality and, if not, reconcile it with whatever model an external clearing house might use. I have no idea if it's possible to 'buy' clearing house services from a financial institution but I would guess at 'no chance at all'.

This functionality is only active when a clearing-house is in operation.

Platform Function	UM	P&C	RM
If configured, a platform can direct all inter-platform payment communication and functionality via a clearing-house, treating it as if it were the only other platform except for also specifying the true target platform in the messages.			x

TODO: Do we direct resource use reports and pricing data via the clearing house as well?

Chapter 4. Operational Mapping

The ESA document [EP-OP-UC] provides descriptions of 'real-world' operational scenarios as use cases.

This section provides a mapping from these 'operational' use cases into the 'functional' use cases presented in this document. A sub-section is included for each use presented in [EP-OP-UC] that provides an interpretation and mapping.

The functional use cases in this document are identified as **UCF#2.x** - referencing the use case described in **section 2.x** of this document.

4.1. OP-UC1: Remote Service execution

In this use case a consumer (Emily) interacts with her primary platform (OGEO-REP - Oil and Gas Earth Observation Response Portal) via its dashboard. To support Emily's interactions, the OGEO-REP platform in-turn interacts with data sources for in-situ data, and with another service provider (CLS-Platform) for added-value resources, including a remote processing execution request.

To map these interactions into our functional use cases we consider the following roles:

- **OGEO-REP** ⇒ Exploitation Platform (presented to Emily), External Application (presentation to CLS-Platform)
- **CLS-Platform** ⇒ Exploitation Platform (presentation to OGEO-REP)

The following functional use cases are identified as being of particular relevance:

- **UCF#2.1** (User accesses Platform services)
  - step 1 - user access to system
- **UCF#2.2** (Consumer discovers and visualises products)
  - step 2 - discovery of data according to collection/spatial/temporal criteria
- **UCF#2.8** (Consumer executes Systematic processing)
  - step 2 - systematic processing triggered by new data
- **UCF#2.4** (Consumer discovers and executes On-demand Processing Service)
  - step 5 - remote on-demand processing
  - step 6 - visualisation of processing results
- **UCF#2.10** (Consumer accesses EP services with External Application)
  - step 5 - OGEO-REP acts as External Application exploiting CLS-Platform API

4.2. OP-UC2: Cross platform automatic application deployment and execution

In this use case a consumer (Alice) interacts with her primary platform (FS-TEP). The FS-TEP

publishes a processing algorithm "Biomass Calculation". Alice invokes this processing algorithm on the FS-TEP, providing an associated AOI/TOI to characterise the job. The FS-TEP is federated to the PROBA-V MEP platform and identifies its PROBA-V data to satisfy the job. Thus, the FS-TEP deploys the "Biomass Calculation" process to the PROBA-V MEP for execution close to the data. Alice is presented with the outcome by the FS-TEP.

Note that all of Alice's interactions are with the FS-TEP - the interactions between FS-TEP <→ PROBA-V MEP are unseen by Alice. Both FS-TEP and PROBA-V MEP take the role of Exploitation Platform.

The following functional use cases are identified as being of particular relevance:

- **UCF#2.4** (Consumer discovers and executes On-demand Processing Service)
  This use cases covers all aspects of operational use case OP-UC2.
  Note that since version 1.0 of this document, the use case UCF#2.4 has been updated to meet the needs of OP-UC2 - in particular, the platform→platform federation for deployment and execution of processing 'close-to-the-data'.

4.3. OP-UC3: New Application Deployment

In this use case an expert user (Bob) develops and packages an algorthim "Lava Flow Detection" (LVD), designed for deployment to the Charter Processing Platform that supports the International Charter for "Space and Major Disaster". Bob creates a package in accordance with the platform's specifications and includes everything required to execute the application (self contained), metadata describing the application usage/interfaces/constraints in a machine-readable format, and manual/tutorial material to help future users.

Bob submits the LVD application as a package to the Operator of the Charter Processing Platform. Before the package is made available to other users of the platform, the platform Operator takes responsibility to validate the integrity and function of Bob's submission. The Operator submits the package to the platform via its API which performs automated steps to register the application package, and add it to the application catalogue along with its associated documentation (manual/tutorial) - initially for the Operator's own sole consumption.

Using the platform the Operator performs testing/validation, before sharing the application for wider consumption by authorised platform users (described as Project Managers and Value-adders in this scenario).

The following functional use cases are identified as being of particular relevance:

- **UCF#2.1** (User accesses Platform services)

  ◦ All steps - The interactions of the different user roles with the platform resources are authorized by their configured privileges.

  ◦ step 3 - The definition of the Platform's rules of governance are supported by the authorization policies/rules. In this case the Platform Operator ('Admin') has the rights (perhaps to the exclusion of others) to deploy application packages to the platform and then authorize associated access to other users.

  ◦ step 5 - According to their privileges, users can manage the sharing of resources under their

administration, by authorizing/revoking access by other platform users/groups

- **UCF#2.11** (Expert user builds new processing service)

  ◦ step 1 - Expert development of an application package, ready for deployment to the platform
  Note that, as far as EOEPCA is concerned, it is not strictly necessary for the application developer to expose the Dockerfile used to create the container - the application package is submitted as a ready prepared (binary) image. Nevertheless, it is helpful to review the Dockerfile that creates the container to support the integrity validation of the image.

  ◦ step 3 - Platform Operator deploys the application to the platform for testing
  Note that in the description for UCF#2.11 this is described as being performed directly by the Expert. However, the Platform Operator role can assume these steps to submit the application package to the platform, with associated checks and catalogue regoistration etc. (all automated by the platform).