OGC Testbed 13 – ESA Sponsored Threads – Exploitation Platform

Technical Architecture

December 09, 2016
Reference number: PDGS-EVOL-CGI-TN-16/1570
Issue: 1.0

Table of Contents

Appendix B: Technical Architecture

As stated on the ESA website, “Earth Observation” (EO) satellites developed or operated by ESA have provided a wealth of data for more than 20 years. In the coming years, the Sentinel missions, along with the Copernicus Contributing Missions as well as Earth Explorers and other, Third Party missions will provide routine monitoring of our environment at the global scale, thereby delivering an unprecedented amount of data. This expanding operational capability of global monitoring from space, combined with data from long-term EO archive, in-situ networks and models will provide users with unprecedented insight into how our oceans, atmosphere, land and ice operate and interact as part of an interconnected Earth System.

While the availability of the growing volume of environmental data from space represents a unique opportunity for science and applications, it also poses a major challenge to achieve its full potential in terms of data exploitation. In this context ESA has started in 2014 the EO Exploitation Platforms (EPs) initiative, a set of R&D activities that in the first phase (up to 2017) aims to create an ecosystem of interconnected Thematic Exploitation Platforms (TEPs) on European footing, addressing:

- Coastal,
- Forestry,
- Hydrology,
- Geohazards,
- Polar,
- Urban themes; and
- Food Security (under definition),

In short, an EO exploitation platform is a collaborative, virtual work environment providing access to EO data and the tools, processors, and Information and Communication Technology resources required to work with them, through one coherent interface. As such the EP may be seen as a new ground segments operations approach, complementary to the traditional operations concept.

Testbed-13 supports the development of ESA’s Thematic Exploitation Platforms (TEP) by exercising envisioned workflows for data integration, processing, and analytics based on algorithms developed by users. These algorithms are initially developed by TEP users in their local environments and afterwards tested on the Thematic Exploitation Platform. As illustrated in the figure below, the goal is to put an already developed application into a *Exploitation Platform (EP) Application Package*, upload this package to the Thematic Exploitation Platform (*TEP*), and deploy it on infrastructure that is provided as a service (*IaaS*) for testing and execution. The entire workflow should support federated user management (*Identity provider and security token service*) and makes use of already available catalog services (*Central Geospatial Resource Catalog*) and catalog interfaces as part of the cloud platforms.

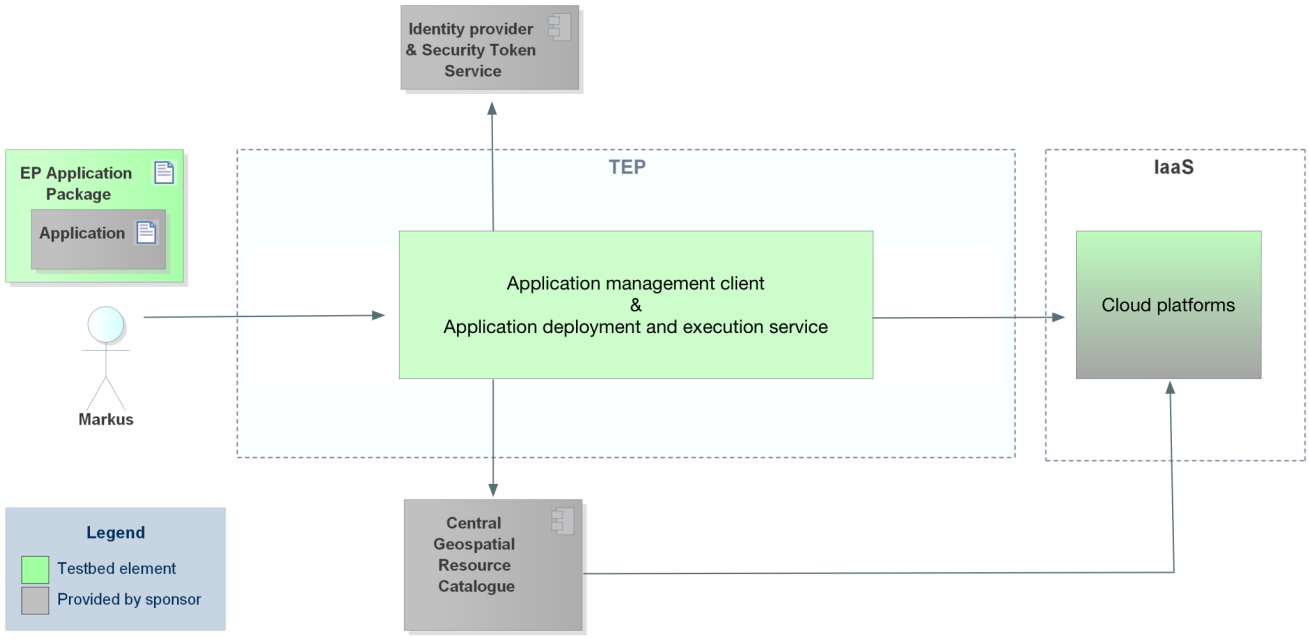

Figure 1. Overview of the Thematic Exploitation Platform base components

Testbed-13 focuses on three elements:

- the *Exploitation Platform Application Package* (in short: *EP Application Package*), which includes the applications and settings for its execution,
- an *Application Management Client* that provides a visual front end to the user and provides the necessary business logic for workflow management and execution
- a *_Application Deployment and Execution Service*, which acts as a front end to cloud platforms.

Together with additional components provided by the sponsor, these elements are implemented and specified in Engineering Reports (ER), both illustrated in the following figure. All deliverables are further defined in the overview table.

In order to ensure high levels of interoperability, two implementations will be funded on the EP Application Package, the Application Management Client and the Application Deployment and Execution Service.

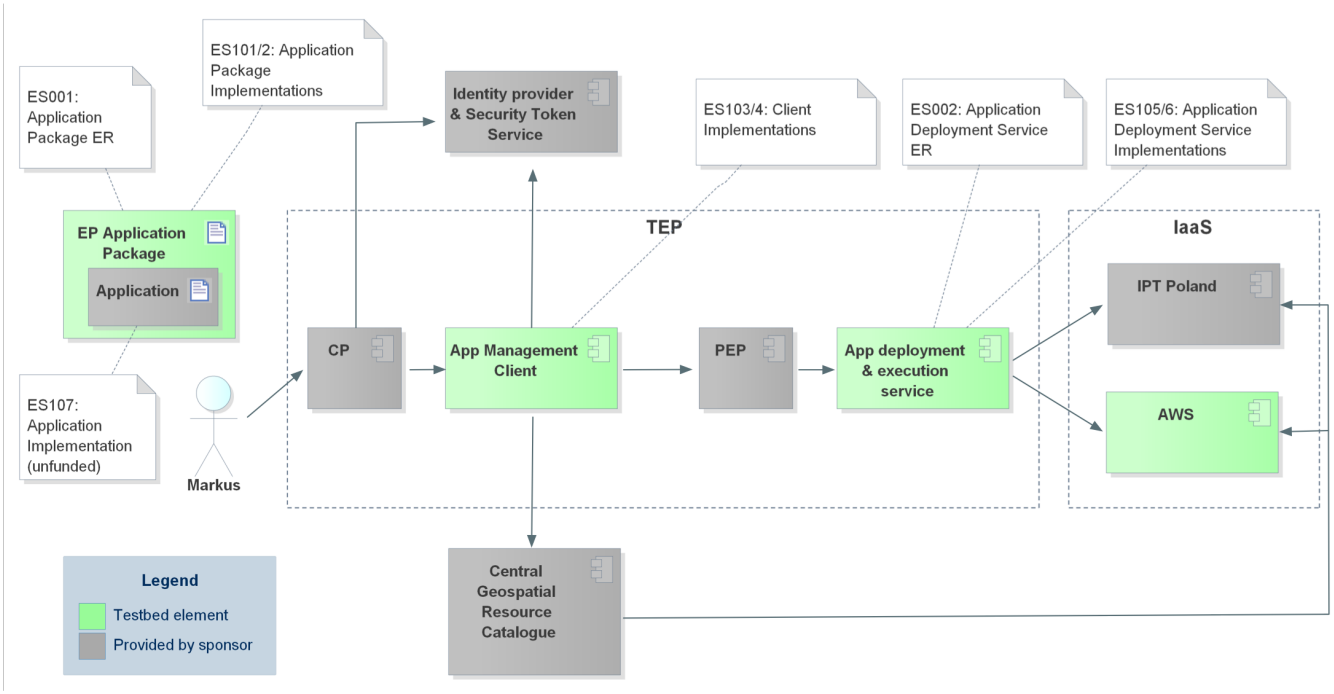

Figure 2. Overview of the Thematic Exploitation Platform base components

B.1. Forestry TEP Scenario

The following describes a forestry use case that shall be implemented as part of Testbed 13. The scenario description shall help understanding the general workflow, separation of responsibilities between the various elements, and the interaction messages exchanged between components.

B.1.1. Actors

Identifier	Description
Markus	Markus is a researcher. He is studying the health of forests in Romania and wants to analyze their changes over the years. To do so he needs a recurring forest map over Romania ranging from as far back as possible and continuously generated for the future. He has developed an algorithm to create forest map from optical images with band adjustment in order to highlight changes on vegetation.
Identity Provider and Security Token Service	All services are part of a federated security infrastructure based on Shibboleth. The identity provider & security token service implements key functionality of this Shibboleth security infrastructure. The infrastructure is further complemented by check points (CP) to handle user authentication and policy enforcement points (PEP) for token encoding.
Central Geospatial Resource Catalogue	The Central Geospatial Resource Catalogue contains information on all Geospatial data collections and services that are part of the scenario. The catalogue identifies also the services which are associated to each collection. In particular how data can be accessed: * Order * Download * Process on a hosted processing platform * Visualization The Geospatial Resource Catalogue provides an Open Search interface with GeoJSON encoding.

Identifier	Description
Forestry TEP	The Forestry TEP is a platform collecting tools for data analysis and information production on forest areas on the globe. It allows the collaboration among several institutions for the development of new algorithm and services. It does not store EO data from satellites but relies on External Platforms. The Forestry TEP contains the *Application Management Client*, the *Application Deployment & Execution Client*, and the security and authentication check and policy enforcement points.
IPT Poland	IPT Poland is a platform providing hosted processing services on all Sentinel-2 data and Landsat-8 data over Europe, which are hosted on the platform. It offers its service only to scientists. IPT Poland provides an OpenSearch catalog interface to allow discovery of all products served.
AWS	Amazon Web Service hosts the Application sandbox of the Forestry Exploitation Platform. Authorized users can test their application with a small set of test data using the sandbox. AWS provides both Landsat-8 and Sentinel-2 data to the sandbox. AWS provides an OpenSearch catalog interface to allow discovery of all products served.

B.1.2. Assumptions

Identifier	Description
A1	IPT Poland publishes all information on the collections managed and the services offered on them. This information is harvested by the Central Geospatial Resource Catalogue
A2	A federation of identity providers exists. Markus has sufficient credentials to deploy and execute services on all components being part of the Testbed.
A3	The billing of resource usage is performed cumulatively by IPT Poland or AWS to the Forestry Exploitation Platform for all resources requested on behalf of its users. IPT Poland and AWS also report an accounting of resource usage by each user in the billing period. The Forestry Exploitation Platform then bills its users taking into account the resource usage in the partner platforms and other internal costs.

B.1.3. Scenario Narrative Description

The following diagram illustrates all steps of Testbed-13 scenario.

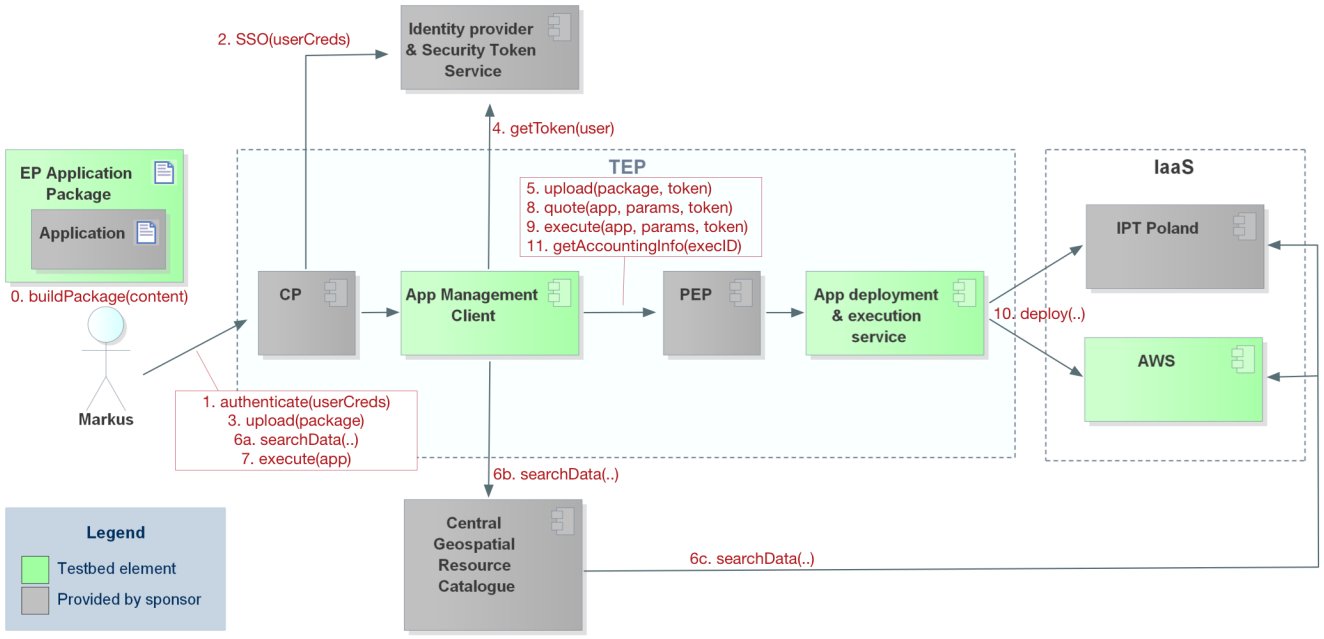

Figure 3. TEP components

Part 1: Application Deployment and Testing

This first part describes the TEP application package generation, the package deployment, and the interaction with the test environment.

Step 0: Application Package generation

Reference: Figure: *TEP components*, step *0*

Markus has developed an algorithm to create a forest map from satellite optical images using Sentinel-2 and Landsat-8. He needs a recurring forest map over Romania ranging from as far back as possible and continuously generated for the future. He is a user of the Forestry Exploitation Platform. He wants to use it to obtain the map, so he packages its algorithm together with additional information into an EP Application Package. The EP Application Package allows to:

- identify the Application to which it refers
- describe the required execution environment for the application
- identify how to deploy and launch the application
- describe the input data (both Sentinel-2 MSI L1C and Landsat-8 OLI L1T)
- describe additional input parameters (e.g. band adjustment variables for its algorithm)
- describe the output (e.g. a JPEG 2000 image)

Testbed-13 shall define the Exploitation Platform Application Package (ES001) and generate instances for the forest map scenario (ES101 & ES102). One example application (the one described in this scenario) will be provided. Participants are invited to develop their own applications and test them as part of this testbed (ES107)

Step 1-3: Application Package upload

Reference: Figure: *TEP components*, step *1, 2 & 3*

Markus authenticates at the security check point (*CP*), which determines if the user has an active session and, discovering that he does not, directs them to the *identity provider and security token service* in order to start the single-sign-on (SSO) process. He then uploads the *Application Package* to the *Application Management Client* (ES103 & ES104).

Step 4-5: Application Package upload, part 2

Reference: Figure: *TEP components*, step *4 & 5*

The *Application Management Client* retrieves the user token from the *Security Service Service* and uses it to pass the policy enforcement point that facades the *Application Deployment & Execution Service*. This service might be developed as a specific WPS profile or as a new OGC Web service interface type. The *Application Deployment & Execution Service* further checks that the execution environment required is available.

Step 6: Data discovery

Reference: Figure: *TEP components*, step *6*

Once the application package has been registered with the TEP, Markus runs a discovery process to find appropriate data for testing his algorithm. He interacts with the *Application Management Client*, which forwards all search requests to the *Central Geospatial Resource Catalog*. The catalog provides an Open Search interface with GeoJSON encoding. The GeoJSON response model will be provided by the sponsor. It will be presented at the OGC TC Meeting in March 2017.

The catalog redirects search requests to other catalogs. IPT Poland supports already an OpenSearch interface for that reason. An additional catalog on AWS will be provided by the sponsor.

The catalog provides hits for Markus search on *IPT Poland* and *AWS*. As Markus is still testing, he selects the cost-free *AWS* platform for testing.

Step 7-10: Application deployment for testing

Reference: Figure: *TEP components*, step *7, 8, 9 & 10*

Markus selects his Forest Map application within the *Application Management Client* of the Forestry TEP and needs to run it in the Application sandbox in order to test it with a significant set of test data.

The *Application Management Client* traverses the *PEP* again and requests a quote for the specific execution request. The *Application Deployment & Execution Service* provides the quote and Markus confirms the execution on *AWS*. The *Application Management Client* then provides all parameters to the *Application Deployment & Execution Service*, which deploys the application on the AWS. Though in Testbed-13 the AWS may be tightly connected to the *Application Deployment & Execution Service*, the Engineering Report ES002 defines all requirements and initial ideas on interface design for cloud APIs. It is planned to develop the cloud API in a future testbed.

Once the *Application Deployment & Execution Service* has transferred the Application package to the sandbox hosted on AWS, it becomes available for Markus to use. The *Application Deployment & Execution Service* shows the sandbox execution panel to Markus. Here, Markus selects as input data a week of coverage of Sentinel-2 and Landsat-8 data over Romania and launches his Application.

He repeats the operation with several other weekly coverages and analyses the results. If the results are not satisfactory he may decide to modify his algorithm, before redeploying it on the Forestry Exploitation Platform and running it in the sandbox environment.

He can keep testing his application until he is satisfied with the results.

Part 2: Historical Data Generation

This second part uses mostly components already introduced in part one. In addition, it uses externally provided platforms such as IPT Poland, which stores satellite data. IPT Poland will be facaded by an implementation of the *Application Deployment & Execution Service* (ES105 & ES106). The front-end interface of this implementation will be identical to the service facading AWS. The *Application Deployment & Execution Service* needs to support OpenStack and here in particular the Nova API in order to communicate with IPT Poland.

In order to obtain weekly forest maps of Romania for the past years, Markus needs to perform a bulk windowed execution of his algorithm. It is a bulk data processing in which data are passed to the Application in sets determined by a temporal window. Each set of data produces an output, the weekly forest map, and the set of all the outputs of each single execution is considered the output of the bulk windowed execution.

Bulk Windowed Execution Programming

In order to obtain weekly forest maps of Romania for the past years, Markus selects his Application within the Forestry Exploitation Platform and requires a Bulk Windowed Execution. He specifies the following parameters:

- 7 days shifting window
- 7 days shift

Markus also selects the option to be notified by email when the execution is completed.

Input Data Collection Search

Markus is required to select the collections on which the application has to run. The Platform searches the *Central Geospatial Resource Catalog* to find Sentinel-2 and Landsat-8 collections which can be processed online without being downloaded (i.e. the products are stored in a hosted processing platform) on an the execution environment supported by the application.

The catalogue returns a set of collections for each input type and the Platform present them to Markus who selects those on IPT Poland.

Data selection option

Then Markus is asked by the Forest Exploitation Platform to provide the selection criteria for the specific products to use. He selects the following options

- AOI: Romania
- Cloud coverage < 10%
- Cumulative time coverage: all available data

Markus requests a quote from the TEP. After agreeing with the quote, Markus launches the application execution.

Application migration to execution environment

The Forestry Exploitation Platform, not hosting the input data, transfers the Mosaic Application Package to the IPT Poland Platform restricting its usage to Markus. The *Application Deployment & Execution Service* submits a Virtual Machine image to IPT Poland (details to be developed). IPT Poland ingests the application checking that:

- Markus is authorized to use IPT Poland resources (i.e. it is a scientist), using the federated identity management system
- it supports the execution environment specified in the application package

Bulk Windowed Execution

The Forestry Exploitation Platform then requires the bulk windowed execution of the Forest Map Application deployed on IPT Poland on behalf of Markus using the parameters specified by Markus. IPT Poland instantiates several instances of the Application passing each of them a different set of inputs. When all the set of inputs have been processed IPT Poland place all the outputs in a web location accessible by Markus and notify the Forestry Exploitation Platform of the execution completion. Then the Forestry Exploitation Platform notifies Markus via email.

Part 3: Continuous Weekly Data Generation

In order to obtain a weekly forest map over Romania until his job is completed, Markus needs to request a *Periodic Execution* of his Application.

Periodic Execution Programming

Markus selects his Forest Map application within the Forestry Exploitation Platform and requires a period execution with:

- frequency “each week on Tuesday”
- shifting time coverage from “execution day -8” to “execution day -2”.
- repeat until manually disabled

As it is an asynchronous operation Markus also selects the option to be notified by email when a new output is available.

Input Data Collection Search

Markus is required to select the collections on which the application shall run. The Platform searches the Central Geospatial Catalogue to find Sentinel-2 and Landsat-8 collections which can be processed online without being downloaded (i.e. the products are stored in a hosted processing platform) on an the execution environment supported by the application.

The catalogue returns a set of collections for each input type and the Platform present them to Markus who selects those on IPT Poland.

Data selection option

Then Markus is asked by the Forest Exploitation Platform to provide the selection criteria for the specific products to use. He selects the following options

- AOI : Romania
- cloud coverage < 10%

Markus requests a quote from the TEP. After agreeing with the quote, Markus launches the application execution.

Periodic Execution

Each Tuesday the Forestry Exploitation Platform requires the execution of the Forest Map Application to IPT Poland on behalf of Markus passing the parameters specified by Markus adapted to the actual execution date.

IPT Poland will check that Markus is an authorized user and then will launch the Forest Map Application. The platform will identify the needed input, using the selection criteria specified by Markus, and will make them available to the application as specified in the application package. After the Mosaic application has completed its execution IPT Poland will collect its output and place it on a web accessible location restricting the access to Markus. It then will notify the Forestry Exploitation Platform of the output availability. The Forestry Exploitation Platform will then email Markus notifying the availability of the Forest Map output in the aforementioned location.

Part 4: Accounting & Billing (Optional)

This part is optional. Idea is to have some accounting and billing mechanisms that allow inter-component accounting and billing.

Figure 4. TEP scenario accounting (optional)

AWS and IPT provide a M2M interface to allow the Forestry TEP to automatically gather accounting information on Markus activities resource usage (e.g. CPU, RAM, disk space, data):

- overall
- per single execution request.

In such a way the Forestry TEP can show to Markus in a dedicated view his resource consumption.

Periodically IPT Poland bills the Forestry Exploitation Platform for all the resources used by the requests sent on behalf of Markus. AWS does the same.

The Forestry Exploitation Platform then bills for Markus activities considering the accounting information received by its partner (IPT Poland and AWS) and other internal additional costs.

B.2. Summary of Deliverables

The following table provides the overview of all deliverables that are part of the Thematic Exploitation Platform thread. Each deliverable is funded exactly once.

Table 1. Deliverables of the Thematic Exploitation Platform Thread

Identifier	Description
ES001 - EP Application Package ER	The EP Application Package ER defines a data model and serialization for EP Application Packages. The packages shall allow to: - identify the Application to which it refers - describe the required execution environment for the application - identify how to deploy and launch the application - describe the input data (both Sentinel-2 MSI L1C and Landsat-8 OLI L1T) - describe additional input parameters (e.g. band adjustment variables for its algorithm) - describe the output (e.g. a JPEG 2000 image)
ES002 - Application Deployment and Execution Service ER	The Application Deployment and Execution Service ER defines an API that supports all functionality provided to realize the Forestry TEP scenario. The ER might define a profile for a WPS v2.0 or a new API to front-end arbitrary platforms and infrastructures that are provided as a service. In addition, the ER shall capture all experiences made during the implementation process and shall document the developed implementations. It shall further contain an exhaustive list of requirements to be fulfilled by cloud APIs in order to allow an automation of the application package deployment and execution workflow. Finally, the ER shall capture all experiences made during the implementation process of both the Application Management Client and the Application Deployment & Execution Service implementations, in particular the handling of the GeoJSON encoding of the catalogue search responses.
ES101 - EP Application Package Implementation 1	An implementation of the EP Application Package as described in the Forestry TEP scenario and defined in **ES001**. The format of the package is to be defined.

Identifier	Description
ES102 - EP Application Package Implementation 2	A second implementation of the EP Application Package as described in the Forestry TEP scenario and defined in **ES001**. The format of the package is to be defined.
ES103 - Application Management Client Implementation 1	An implementation of the *Application Management Client* application as described in the Forestry TEP scenario. The client shall be developed as a Web application. Focus of the client shall be on the business logic, not of the graphical user interface. The client needs to allow uploading of application packages, interacting with the Security Token Service and the Application Deployment and Execution Service, and the Open Search interface of the Central Geospatial Resource Catalog.
ES104 - Application Management Client Implementation 2	A second implementation of the *Application Management Client* application as described in the Forestry TEP scenario. The client shall be developed as a Web application. Focus of the client shall be on the business logic, not of the graphical user interface. The client needs to allow uploading of application packages, interacting with the Security Token Service and the Application Deployment and Execution Service, and the Open Search interface of the Central Geospatial Resource Catalog.
ES105 - Application Deployment & Execution Service Implementation 1	An implementation of the *Application Deployment & Execution Service* as described in the Forestry TEP scenario and defined in **ES002**. The service needs to connect to the AWS to deploy applications on the cloud. AWS resources will be provided. The service needs to support virtual machine image deployments to IPT Poland. Details of this interaction need to be defined during the testbed.
ES106 - Application Deployment & Execution Service Implementation 2	A second implementation of the *Application Deployment & Execution Service* as described in the Forestry TEP scenario and defined in **ES002**. The service needs to connect to the AWS to deploy applications on the cloud. AWS resources will be provided. The service needs to support virtual machine image deployments to IPT Poland. Details of this interaction need to be defined during the testbed.
ES107 - Application implementation (unfunded)	The application provided as part of the scenario described above will be provided. Participants are invited to submit their own applications that will be loaded into EP Application Packages and executed by the clients and services developed as part of this testbed. This deliverable is unfunded!