OGC Testbed-13

EP Application Package ER

Table of Contents

Publication Date: 2018-01-30

Approval Date: 2018-01-30

Posted Date: 2018-01-18

Reference number of this document: OGC 17-023

Reference URL for this document: http://www.opengis.net/doc/PER/t13-ES001

Category: Public Engineering Report

Editor: Pedro Gonçalves

Title: OGC Testbed-13: EP Application Package ER

OGC Engineering Report

COPYRIGHT

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

The Application Package OGC Engineering Report (ER) defines a data model and serialization for Thematic Exploitation Platforms (TEP) Application Packages. A TEP refers to a computing platform that follows a given set of scenarios for users, data and ICT provision aggregated around an Earth Science thematic area. This ER is part of the Testbed-13 Earth Observation Clouds (EOC) effort to support the development by the European Space Agency (ESA) of the TEP by exercising envisioned workflows for data integration, processing, and analytics based on algorithms developed by users that are deployed in multiple clouds.

The wide usage of virtualization and the possibility to start virtual environments within Cloud services significantly simplifies the creation of environments and provisioning of resources. However, it still leaves a problem of portability between infrastructures. This ER identifies a strategy for packaging an application in a Cloud environment that will be able to run in a predictable manner in different computing production environments. The application packaging specifies the elements that will ensure:

- Scientific reproducibility,
- Dependencies identification and management,
- Maintainability from an operational perspective and avoid version pilling,
- Portability in different Cloud providers

The ER proposes the use of containers, defining everything required to make a piece of software run packaged into isolated containers. Unlike a Virtual Machine (VM), a container does not bundle a full Operating System (OS) - only libraries and settings required to make the software work are needed. This makes for efficient, lightweight, self-contained systems and guarantees that software will always run the same, regardless of where it’s deployed. A discussion on application deployment and execution is presented in the separate OGC Testbed-13 Application Deployment and Execution Service ER [1].

1.1. Requirements

This ER will define an application package according to the following requirements:

- 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 collections
- Describe additional input parameters

1.2. Key Findings and Prior-After Comparison

The serialization and deployment of applications behind an OGC Web Processing Service (WPS) was the subject of several discussions at OGC Standards Working Groups (SWG) and Domain Working Groups (DWG) in the past. A diverse number of issues were identified but the main barriers were due to the diversity of runtime environments to support and the respective technology maturity hampered its feasibility.

The advance in container technology together with the concept of federation of Clouds presents new opportunities to address this issue. Previously the constraints imposed in describing the runtime environments (e.g. Linux, Java VM, Perl, Python) and the respective application installation would create a considerable amount of information needed for the application package information model outside the OGC aegis. With containers, most of these requirements are addressed at their respective level with configuration and software elements packaged into isolated elements.

This ER will define the application package aggregating containers and application information to defines a data model and serialization for EP Application Packages.

1.3. What does this ER mean for the Working Group and OGC in general

The concepts presented in this ER will define operational procedures to allow the automatic packaging of applications and deployment of WPS in different computing infrastructures. The main focus of the work presented is on Earth Observation (EO) data processing services but it can be directly applied in other geospatial services.

1.4. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Pedro Gonçalves	Terradue
Panagiotis (Peter) A. Vretanos	CubeWerx
Patrick Jacques	Spacebel
Christophe Noel	Spacebel
Paulo Sacramento	Solenix

1.5. Future Work

Future OGC testbeds should look to develop draft OGC Web Services Context (OWS Context) profiles for describing Cloud application packages. The approach adopted for Testbed-13 offers a convenient way of conveying relevant process description information (e.g. inputs/outputs) in an OGC standard way, but future work should consider using this as a process description template, for example using template fields and properties (e.g. $id).

1.6. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following normative documents are referenced in this document.

- RFC4287, The Atom Syndication Format [https://tools.ietf.org/html/rfc4287]
- OGC 06-121r9, OGC® Web Services Common Standard [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2]
- OGC 12-084r2, OGC® OGC Context Atom Encoding Standard [https://portal.opengeospatial.org/files/?artifact_id=55183]
- OGC 14-065, OGC® WPS 2.0 Interface Standard [http://docs.opengeospatial.org/is/14-065/14-065.html]

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard OGC 06-121r9 [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

3.1. Area Of Interest

An Area of Interest is a geographic area that is significant to a user.

3.2. Context Document

A Context Document is a document describing the set of services and their configuration, and ancillary information (area of interest etc) that defines the information representation of a common operating picture.

3.3. Dockerfile

A text document that contains all the commands a user could call on the command line to assemble an image

3.4. Container

A lightweight and configurable virtual machine with a simulated version of an operating system and its hardware, which allow software developers to share their computational environments.

3.5. Infrastructure as a Service (IaaS)

A standardized, highly automated offering, where compute resources, complemented by storage and networking capabilities are owned and hosted by a service provider and offered to customers on-demand. Customers are typically able to self-provision this infrastructure, using a Web-based graphical user interface that serves as an IT operations management console for the overall environment. Application Programming Interface (API) access to the infrastructure may also be offered as an option.

3.6. Resource

A resource is a configured set of information that is uniquely identifiable to a user. Can be realized as in-line content or by one or more configured web services.

3.7. Virtual Machine Image

A virtual machine image is a template for creating new instances. A image can be plain operating systems or can have software installed on them, such as databases, application servers, or other applications.

3.8. Thematic Exploitation Platform (TEP)

A "Thematic Exploitation Platform" refers to an environment providing a user community interested in a common Theme with a set of services such as very fast access to (i) large volume of data (EO/non-space data), (ii) computing resources (iii) processing software (e.g. toolboxes, retrieval baselines, visualization routines), and (iv) general platform capabilities (e.g. user management and access control, accounting, information portal, collaborative tools, social networks etc.). The platforms thus provide a complete work environment for its' users, enabling them to effectively perform data-intensive research and data exploitation by running dedicated processing software close to the data.

Chapter 4. Abbreviated terms

- ADES Application Deployment and Execution Service
- AMC Application Management Client
- API Application Programming Interface
- COTS Commercial Off The Shelf
- CP CheckPoint
- DCE Distributed Computing Environment
- EOC Earth Observation Clouds
- IdP Identity Provider
- STS Security Token Service
- TEP Thematic Exploitation Platform
- WPS Web Processing Service

Chapter 5. Overview

The Application Package targets applications built in the context of the ESA initiative to build an ecosystem of TEP. The TEP defines three canonical user scenarios covering the EO data exploitation (users discover and manipulate data), new EO service development (users deploy a new processor) and new EO product development (users publish new results). The TEPs are implementing these canonical scenarios and moving the processing to the data, rather than the data to the users, thereby enabling ultra-fast data access and processing.

The algorithms are initially developed by a third-party developer and subsequently deployed in a TEP. The Application Package will contain all the information necessary to allow applications to be specified and deployed in federated resources providers. The goal of the exercise is to define a TEP Application Package taking in consideration past efforts on the OGC Web Services Context Document (OWS Context) and OGC WPS as the means to integrate OGC services with well-established formats and protocols that are natively ready for interoperability within Cloud solutions.

This ER will focus on two main sections. This section that lists the requirements for the information model of the application package requirements as identified through thread meetings and discussions. The following section will describe the encoding approach and recommendations for its usage patterns.

5.1. Scenario

There are different valid strategies for packaging and deploying an Application in a Cloud environment. Overall, the goal is always to have an application able to run in a predictable manner in a large computing production environment, potentially formed by hundreds of nodes. The simplest approach for packaging is the "snapshot", also known as the “Golden image”. Its lifecycle is quite straightforward: when the development of the application is finished, the application and the underlying operating system are frozen and then moved in production to be executed (i.e., using multiple resources). In this case the dependencies are included in the snapshot that thwarts the maintainability and versioning of the service. There are a few major disadvantages to this approach due to the large manual intervention required to prepare the snapshots, dependencies not being explicitly specified and, most importantly, the precise reproducibility cannot be guaranteed.

The other, preferable approach is to guide the user (in this case a developer or application integrator) in order to build an application compatible with the platform’s applications. Such applications have the main advantage of being self-contained. In other words, a single package (e.g. in RPM Package Manager (RPM) format, formerly Red Hat Package Manager format) contains the application resources and the dependencies specification. To complement the approach, the contextualization process can be optimized with the definition of specialized baselines with pre-installed packages (e.g., Interactive Data Language, Matlab). Following the proposed approach, the application packaging will include dependencies specification that improve:

- Scientific reproducibility,
- Dependencies identification and management,

- Maintainability from an operations perspective and avoiding disk pilling with several versions of the application,
- Portability in different Cloud providers, avoiding the snapshot building and importing on the specific Cloud Provider

In both cases, the wide usage of virtualization and the possibility to start virtual environments within Cloud services significantly simplify the creation of environments and the provisioning of resources. However, this still leaves a problem of portability of complete environments where applications are developed unresolved. Previous virtualization solutions are not light and flexible enough partially due to the size of the images. Docker fills this gap by providing a platform designed both for developers and system administrators. Docker is an open platform for developers to build, ship and run distributed applications. Composed of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a Cloud service for sharing light images, Docker enables applications to be quickly assembled from components and eliminates the friction between development and production environments. As a result, user applications can ship faster and run the same process into large multi-tenant data centers in the Cloud. The main difference between a Docker container and a Virtual Machine lies in the fact that a Virtual Machine consists of, besides the application itself, also an operating system with all binary files included. Docker images work as an isolated process in the host operating system, which shares a kernel with other containers. Thereby, still having the benefits of virtualization, it is more portable and more effective – an application uses less disk space.

This solution lets the developers package their applications and all of the dependencies in a Docker container to provide a consistent environment for execution and also provides isolation from other applications or software installed on the host. Processes are therefore running in an isolated processing environment which is exactly the same that the developer built for it and can be deployed in several cloud providers (as shown in the figure below).

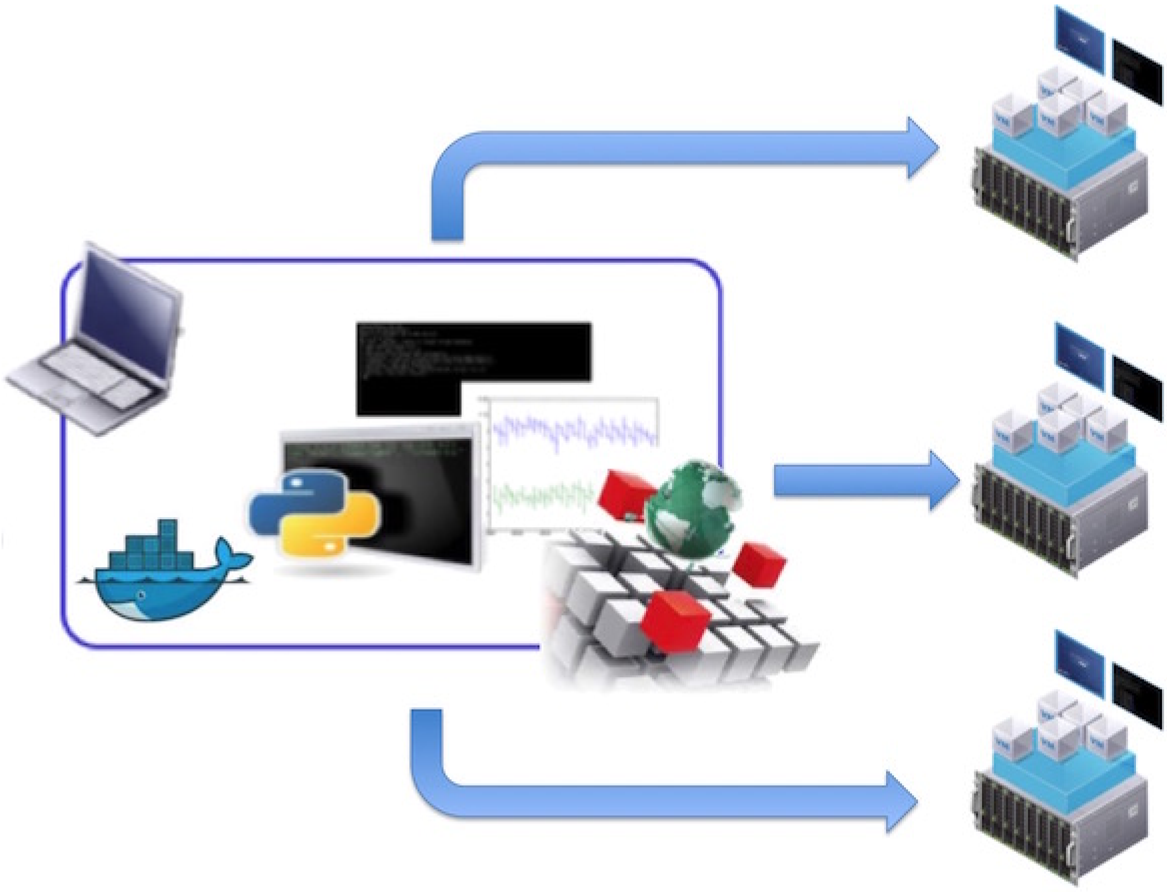

Figure 1. Deploying the same applications in different infrastructures

From the existing developer environments, the process to build a Docker image is fairly straightforward and well documented and outside of the scope of this ER. Overall, all files and dependencies of the application should be self-contained and thus easily installable on the Docker image baseline, the same on which the user has integrated the application and part of the application integration output set. When an application is deployed on a given environment, it will need the docker engine operation to deploy the image on the processing container.

The Application Package will allow the developer to register an application that can be deployed and executed in multiple ICT providers. The application is first built and registered in a third-party Application Hub (e.g. DockerHub). The developer will then directly interact with the Application Management Client (AMC) that will register the application on the Application Deployment and Execution Service (ADES).

The sequence of actions is described in the following image.

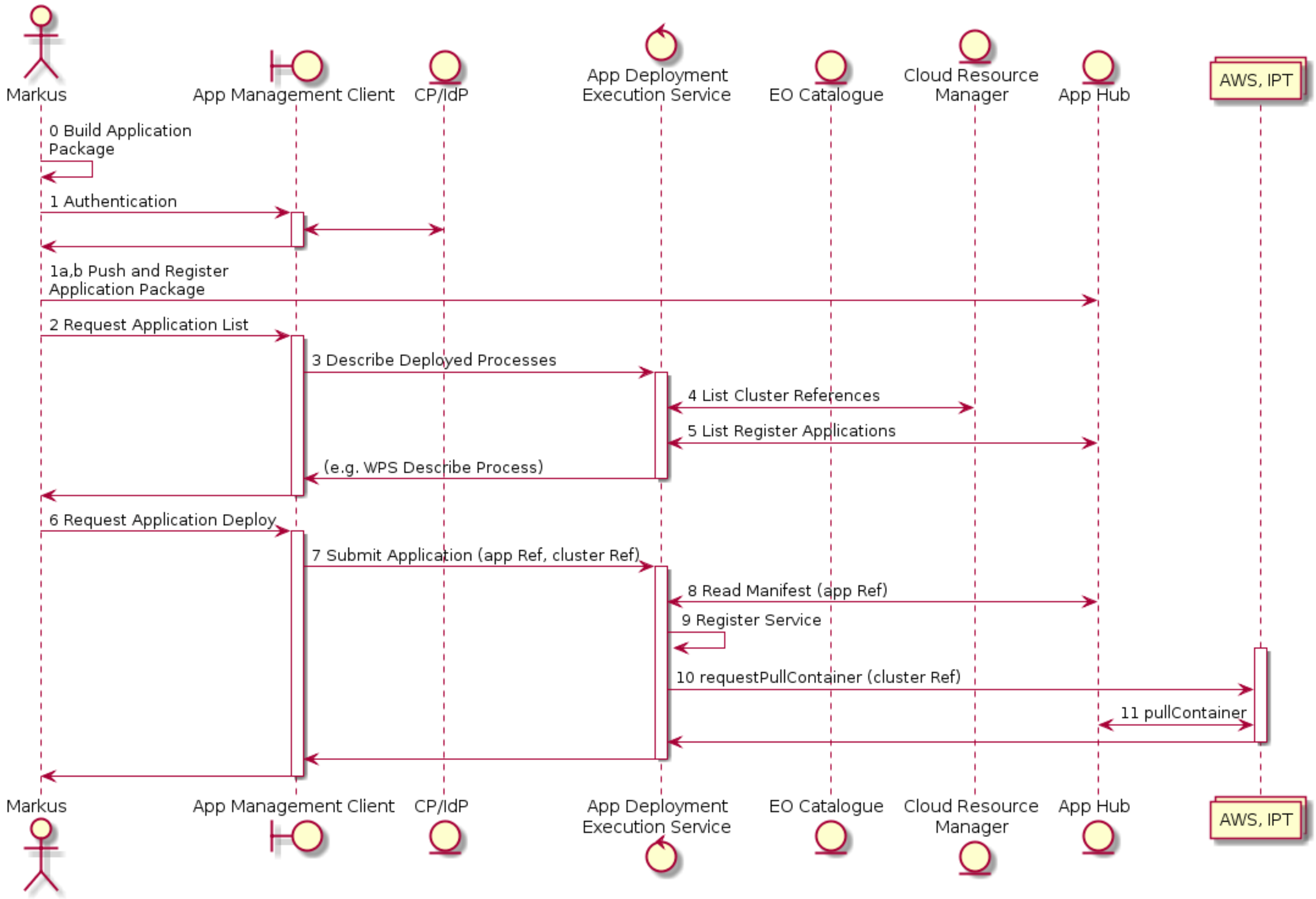
Markus
App Management Client
CP/IdP
App Deployment Execution Service
EO Catalogue
Cloud Resource Manager
App Hub
AWS, IPT
0 Build Application Package
1 Authentication
1a,b Push and Register Application Package
2 Request Application List
3 Describe Deployed Processes
4 List Cluster References
5 List Register Applications
(e.g. WPS Describe Process)
6 Request Application Deploy
7 Submit Application (app Ref, cluster Ref)
8 Read Manifest (app Ref)
9 Register Service
10 requestPullContainer (cluster Ref)
11 pullContainer

Figure 2. Sequence of Deploy and Register Scenario

Initially the developer builds its Application Package after the development & integration phase (Step 0). The output of this activity is generally a file, be it a ZIP or a RPM that, once validated, can be packaged in a container to expose the same software baseline as the original developer environment. The package provides both the Application and the 'delta'-environment required for executing the Application itself on the target environment. The ‘delta’-environment can also be defined as the explicit dependencies of the Application. After the authentication on the CP/IdP (Step 1), the developer will then push and register his/her application in the Application Hub (Step 2).

With the registration of the application the developer can then request its deployment in any of the supported infrastructure. For this - even though not strictly required - the developer can request the available application list (Step 3) to the Application Management Client. This component will retrieve the necessary information by executing a "describe deployed processes" operation (Step 4, consisting of a WPS GetCapabilities and multiple WPS DescribeProcess requests) that will collect the list of supported clusters (Step 5) and list of registered applications (Step 6).

With this information, the user is able to request an application deployment (Step 7) to the Application Management Client that will be responsible for submitting the request to the Application Deployment and Execution Service, taking in consideration the requested application and resources (Step 8).

The ADES will read the manifest (Step 9), register the service (Step 10) and request the application container (Step 11) that will be retrieved for use in the ICT infrastructure (Step 12). The resulting response will be the new address of the deployed WPS service instance.

The overall components and respective flow is also presented below.

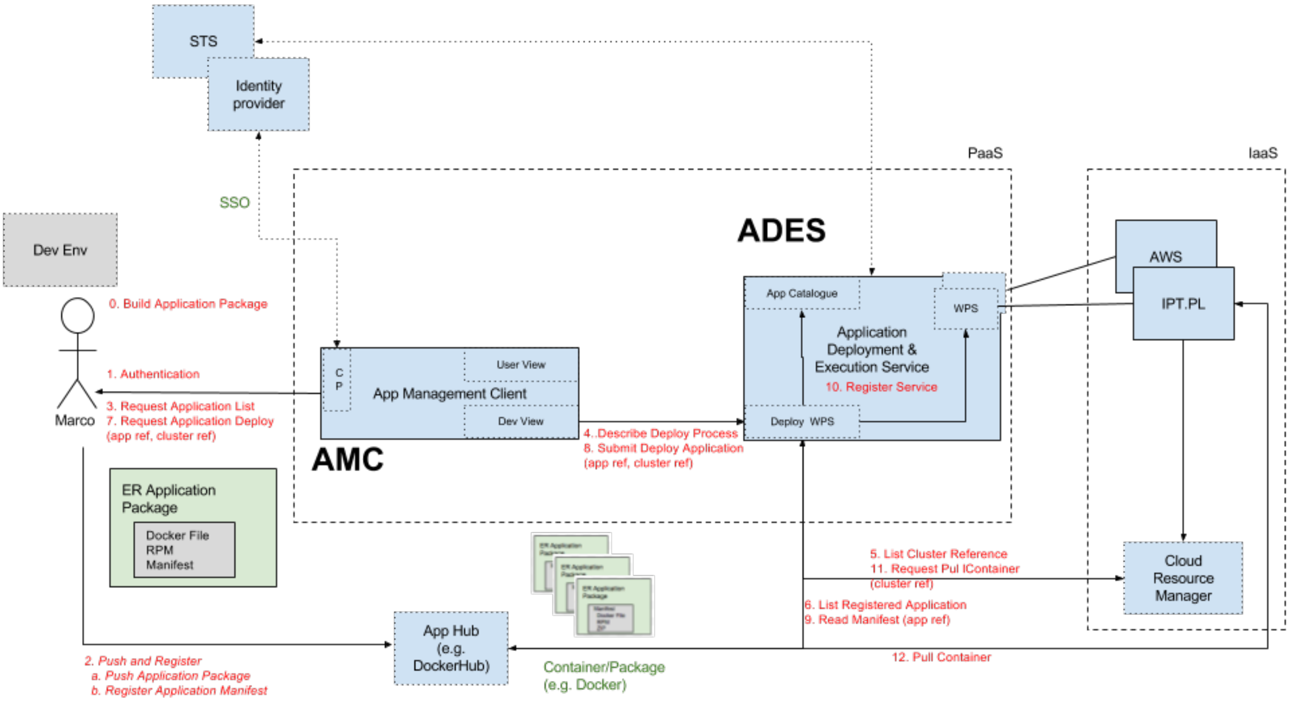

Figure 3. Overview of Deploy and Register Scenario

Chapter 6. Requirements

The Application Package will encapsulate user applications enabling automatic management (upload, deploy, run) and sharing between platforms. It mostly contains information about:

1. Application metadata (e.g. description, author)
2. Application container and resource types
3. Application parameters
4. Input data (data access) and respective catalogue resources
5. Auxiliary map resources to support both the parameters and discovery process
6. Output data

6.1. Application Metadata

The Application Package will contain information about the application title, developer, and short abstract available for display to a human.

Table 2. Generic Application Metadata

Name	Description	Required
id	An unambiguous reference to the identification of the application using an Internationalized Resource Identifier (IRI)	Mandatory
title	A title for the application	Mandatory
abstract	Description of the application purpose or context	Mandatory
updateDate	A date of creation or update of the application	Mandatory
author	Identifier for the author of the application	Mandatory
publisher	Identifier for the publisher of the application	Optional
rights	Information about rights held in and over the application	Optional
areaOfInterest	Restricted geographic area of application applicability	Optional
timeOfInterest	Restricted temporal range of application applicability	Optional
keywords	keywords or tags, it MAY have a related code-list that is identified by a scheme	Optional

6.2. Application Container

The Application should be executed as a Docker Container in a Docker environment.

The developer builds a virtual environment containing all required libraries and resources to execute the application processing (including any input data which will not be provided as an input by the front-end).

A Docker Container Image (the portable snapshot of the container) can be built from a Docker build context stored in a repository. A build context is a Dockerfile and any files at a specific location.

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. A Dockerfile begins with defining an image from which the build process starts, followed by various other methods, commands and arguments (or conditions). The Dockerfile also specifies which command should be called when running the container.

With the Dockerfile, the developer creates a container build context defining the OS, libraries and any environment considerations. It also includes the application files (binary, resources, etc.) and indicates the command that should be called when running the container.

A last but essential step remains, the application should be wrapped by a script in order to read and write the data shared with the ADES.

6.3. Application Parameters

An application execution always implies a set of input parameter(s) and file(s), and generates, as result of the processing, a set of outputs (values or files). The execution of a process behind a WPS relies on the description of the application in an eXtensible Markup Language (XML) document named a WPS Process Description. The WPS specification defines the following terms:

1. Process input: Process inputs are the arguments of a process and refer to data provided to a process. Each process input is an identifiable item.
2. Process output: Process outputs are the results of a process and refer to data returned by a process. Each process output is an identifiable item.

In order to pass simple parameters (WPS Literal Data) to the Application, the ADES can execute the container application passing the parameters by command line parameters (or by environment variables in alternative approaches). The name of the parameters (or environment variable) is specified by the developer in the WPS Process Description as the input Identifier.

6.4. Input Data

When the ADES executes a (WPS) process, the service needs to provide received input files into the (cloud) Docker environment. This folder can be specified by the developer or default to a specific location defined by an environment variable pointing to the working directory

Note that files (inputs and outputs) may be passed by value (file content) or by reference through a Uniform Resource Locator (URL). Passing by reference might impose an additional step of data retrieval by the application. An alternative solution is for the ADES to provide a local reference to the running application (on the platform) if the data is available locally. Constraints or information about this approach should be described in input metadata of the WPS Process Description.

To discover which files are best suited for the required processing, the Application Package must provide the access points of the catalogue(s) where the dataset metadata is stored and where it is possible to perform complex queries. The catalogue should provide a search engine capable to deal with different types of queries (Geographic and non-Geographic) and include the online access points of the files. This catalogue will provide a framework to ease the discovery of the necessary data.

EO products are differentiated by parameters such as the date of acquisition and the image footprint as well as characteristics pertaining to the type of sensor, the type of platform, the applied processing chain, and more. Using the best practices for search services in the EO domain the catalogue should support the OpenSearch specification with Geo, Time (OGC 10-032) and EO extensions. The EO OpenSearch Extension (OGC 13-026) allows also the specializations for specific thematic classes of EO products, such as optical, radar, atmospheric, altimetry, limb-looking and systematic/synthesized EO products and also includes the support for the European Union’s INSPIRE directive.

6.5. Auxiliary Map Resources

The application might also need auxiliary layers to guide the user in selecting the processing parameters or to display the processing result. These layers can be in image format (e.g. map with elevation) or vector (e.g. map with volcanoes positioning) and should be included on the Application Package information.

6.6. Output Data

Once the container is running, the shared input files are consumed by the application running in the virtual container. Finally, the application finishes execution and the ADES collects outputs of the process. The Application shall write (or copy) output data files into the location specified by the metadata.

Chapter 7. Encoding

The Application Package encoding will convey all the necessary information needed for encapsulating user applications and enable their automatic management (upload, deploy, run) in diverse platforms.

The consensus reached in the Testbed-13 EOC is to use the OWS Context document to convey the information model. This document is used as input parameter to a dedicated WPS service that would create and deploy the application as a new WPS process offering.

An alternative approach is also proposed that uses the WPS description document to directly embed the OWS Context Document in an existing WPS Transactional Service. Both technical approaches are presented and analyzed on the other Testbed-13 EOC ER (OGC 17-024) dedicated to reporting of the Application Deployment and Execution Services.

7.1. OWS Context

The OWS Context (OWC) document was initially created to aggregate a set of configured information resources as a collection of services to be shared among different applications or domains. An OWS Context supports OGC services, online resources and in-line content as well. The specification document follows the OGC Approach of a conceptual model (OGC 12-080r2) and implements specific encodings for ATOM (OGC 12-084r2) and GeoJSON (14-055r2) with well-defined mappings.

The OWS Context was designed to support use cases such as the distribution of search results, the exchange of a set of resources such as OGC Web Feature Service (WFS), Web Map Service (WMS), Web Map Tile Service (WMTS), Web Coverage Service (WCS) and others in a ‘common operating picture’. Additionally, OWS Context can deliver a set of configured processing services as Web Processing Service (WPS) parameters to allow the processing to be reproduced. For that, the standard establishes requirements classes for a series of offering extensions, relating to services or content. This ER extends that notion to containers and application parameters to support the deployment of applications in multiple cloud providers.

7.2. Mapping

The Application Package information model uses the ows:Context class (OGC 12-084r2) mapped to the atom:feed element as the overall container class for the document. Its properties are documented below.

Table 3. Application Package Mapping to the OWS Context ATOM document

Name	Definition	Element	Type or Value
Id	An unambiguous reference to the identification of the feed (IRI)	atom:feed/atom:id	URI
Title	A title for the application	atom:feed/atom:title	String

Name	Definition	Element	Type or Value
Abstract	Description of the application purpose or context	atom:feed/atom:subtitle	String
updateDate	A date of creation or update of the application	atom:feed/atom:updated	RFC-3339 date
author	Identifier for the author of the application	atom:feed/atom:author/atom:name	String
publisher	Identifier for the publisher of the application	atom:feed/dc:publisher	String
rights	Information about rights held in and over the application	atom:feed/atom:rights	String
areaOfInterest	Restricted geographic area of application applicability	atom:feed/georss:where	String
timeOfInterest	Restricted temporal range of application applicability	atom:feed/dc:date	String
keywords	keywords or tags, it MAY have a related code-list that is identified by a scheme (TBC)	atom:feed/atom:category/@term	String

The XML representation of a Application Package in feed is shown below

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:georss="http://www.georss.org/georss" xmlns:gml="http://www.opengis.net/gml"
    xmlns:owc="http://www.opengis.net/owc/1.0" xmlns:os="http://a9.com/-
/spec/opensearch/1.1/"
    xml:lang="en">
  <link rel="profile" href="http://www.opengis.net/spec/owc-atom/1.0/req/core"
      title="This file is compliant with version 1.0 of OGC Context"/>
  <link rel="profile" href="http://www.opengis.net/tb13/eoc"
      title="This file is compliant with Testbed-13 EOC Thread for Application
Packing"/>
  <id>http://www.opengis.net/tb13/eoc/examples/app_pkg/dcs-stemp-s2</id>
  <title>OGC Testbed 13 Application Package for GEP Application</title>
  <subtitle type="text">The Application Package OGC Engineering Report (ER) defines a
data model and serialization for EP Application Packages. This document packages the
VOLcanoes Thermal Application for GEP (VOLTAGE) that generates surface temperature
maps over volcanic areas. It generates surface temperature map in file format fitting
the Researcher and Users needs from optical EO missions data</subtitle>
  <updated>2017-22-15T01:28:00Z</updated>
  <author>
    <email>info@terradue.com</email>
  </author>
  <generator uri="https://geohazards-tep.eo.esa.int/" version="2.0">Catalogue Version
XXYXY</generator>
  <dc:publisher>Terradue srl.</dc:publisher>
  <rights>OGC Testbed 13 EOC Thread</rights>
  <!-- Geographic Area of interest for the App -->
  <georss:where><gml:Polygon xmlns:gml="http://www.opengis.net/gml">
<gml:exterior><gml:LinearRing><gml:posList>-90 -180 90 -180 90 180 -90 180 -90 -
180</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></georss:where>
  <!-- A date or range of dates relevant to the App -->
  <dc:date>2005-01-01T09:08:56.0000000Z/2020-01-23T09:14:08.0000000Z</date>
  <entry>
    ..
  </entry>
    ..
  <entry>
    ..
  </entry>
</feed>
```

The OWS Context document includes specific resources to identify specific application information covering:

1. Application container and parameters
2. Catalogue resources for input data (data access)
3. Base Maps Offerings (optional)

Also included on the document is the information about the area of interest of the application package. At feed-level, this optional element expresses a geographic area that is significant to the applications.

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:georss="http://www.georss.org/georss" xmlns:gml="http://www.opengis.net/gml"
    xmlns:owc="http://www.opengis.net/owc/1.0" xmlns:os="http://a9.com/-
/spec/opensearch/1.1/"
    xml:lang="en">
  ...
  <georss:where>
    <gml:Polygon><gml:exterior>
      <gml:LinearRing><gml:posList srsDimension="2">-90 -180 90 -180 90 180 -90 180
-90 -180</gml:posList></gml:LinearRing>
    </gml:exterior></gml:Polygon>
  </georss:where>
  ...
</feed>
```

The Application Package information model uses the ows:Resource class (OGC 12-084r2) mapped to the atom:entry element to define each of the previous elements. Each individual characteristic is documented below.

7.3. Application

The application entry has a profile referenced as *"http://www.opengis.net/tb13/eoc/application"* represented as

```
<feed ...>
  ...
  <entry>
    ...
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/application"
        title="This entry contains an application as specified by Testbed-13 EOC
Thread"/>
    ...
  </entry>
  ...
</feed>
```

The application container information is mapped to a owc:offering that must have a code attribute equal to *"http://www.opengis.net/tb13/eoc/docker"* represented as

```
<feed ...>
  ...
  <entry>
    ...
    <owc:offering code="http://www.opengis.net/tb13/eoc/docker">
      <owc:content type="text/plain">docker-co.terradue.com/geohazards-tep/dcs-stemp-
l8:1.0.4</owc:content>
    </owc:offering>
    ...
  </entry>
  ...
</feed>
```

The application parameters are conveyed by a WPS Description mapped to a owc:offering (OGC 12-084r2) that must have a code attribute equal to *"http://www.opengis.net/tb13/eoc/wpsProcessOffering"*. The offering must have the owc:content (OGC 12-084r2) element with the XML representing the result of a *DescribeProcess* request. It is important to note that even though this WPS Description could match exactly the corresponding *DescribeProcess* response after the process has been deployed, it is not conveyed as part of the Application Package with that intention, given that this poses some problems (e.g. the client/user defines the process identifier before the process is deployed). For the purpose of this testbed, it offers a convenient way of conveying relevant process description information (e.g. inputs/outputs) in an OGC standard way, but future work should consider using this as a process description template, for example using dummy id values or template fields (e.g. $id).

```
<feed ...>
  ...
  <entry>
    ...
    <owc:offering code="http://www.opengis.net/tb13/eoc/wpsProcessOffering">
      <owc:content type="application/xml">
        <wps:ProcessOfferings xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:wps=
"http://www.opengis.net/wps/2.0">
          <wps:ProcessOffering processVersion="1.0.3"
            jobControlOptions="sync-execute async-execute dismiss" outputTransmission
="value reference">
            <wps:Process>
            ...
            </wps:Process>
          </wps:ProcessOffering>
        </wps:ProcessOfferings>
      </owc:content>
    </owc:offering>
    ...
  </entry>
  ...
</feed>
```

Both offerings are part of a single entry as shown below

```
<feed ...>
  ...
  <entry>
    <id>https://github.com/ows13-eoc/dcs-stemp-s2</id>
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/application"
        title="This entry contains an application as specified by Testbed-13 EOC
Thread"/>
    <title>VOLcanoes Thermal Application for GEP</title>
    <content type="text">The STEMP processing chain allows to measure the temperature
for areas identified as «thermally active» in the volcanic areas in quiescent phase
(fumarol fields within active craters along fracture zones of volcanic edifices, or
other areas with anomalous heat fluxes associated to volcanic acitvity). The service
is based on EO optical data from Sentinel-2. Output is a Surface Temperature Map in
GeoTIFF format in a resolution between 10 and 1200m, depending on the Satellite sensor
and bands used. This service is running systematically for about 20 volcanoes in
Europe, Latin America and South-East Asia.</content>
    <owc:offering code="http://www.opengis.net/tb13/eoc/docker">
      <owc:content type="text/plain">docker-co.terradue.com/geohazards-tep/dcs-stemp-
l8:1.0.4</owc:content>
    </owc:offering>
    <owc:offering code="http://www.opengis.net/tb13/eoc/wpsProcessOffering">
      <owc:content type="application/xml">
        <wps:ProcessOfferings xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:wps=
"http://www.opengis.net/wps/2.0">
          <wps:ProcessOffering processVersion="1.0.3"
              jobControlOptions="sync-execute async-execute dismiss"
              outputTransmission="value reference">
            <wps:Process>
            ...
            </wps:Process>
          </wps:ProcessOffering>
        </wps:ProcessOfferings>
      </owc:content>
    </owc:offering>
  </entry>
</feed>
```

It is also possible to extend the entry with additional informative data using the atom category element.

```
<feed ...>
  ...
  <entry>
    ...
    <category scheme="http://www.opengis.net/tb13/eoc/os" term="LINUX"
        label="This app runs in Linux"/>
    ...
  </entry>
  ...
</feed>
```

The information is application-driven and this ER does not provide or mandate any specific guidance for their interpretation.

7.4. Catalogue

An Application Package can contain any given number of entries identified as catalogues. The catalogue entry has a profile referenced as *"http://www.opengis.net/tb13/eoc/catalogue"*

```
<feed ...>
  ...
  <entry>
    ...
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/catalogue"
        title="This entry contains an catalogue as specified by Testbed-13 EOC Thread
"/>
    ...
  </entry>
  ...
</feed>
```

The application catalogue information is mapped to a owc:offering that must have a code attribute equal to *"http://www.opengis.net/spec/owc-atom/1.0/opensearch"* (currently the only catalogue specification supported)

```
<feed ...>
  ...
  <entry>
    ...
    <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/opensearch">
      <owc:operation code="search" method="GET" type=
"application/opensearchdescription+xml"
          href=
"http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3AIPT%3ASentin
el2"/>
    </owc:offering>
    ...
  </entry>
  ...
</feed>
```

The search operation can be refined with the inclusion of a OpenSearch Query element defining a query subset. The following example shows how to define a restriction on the cloud cover query parameter.

```
<feed ...>
...
  <entry>
    ...
    <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/opensearch">
      <ows:operation code="search" method="GET" type=
"application/opensearchdescription+xml"
          href=
"http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3AIPT%3ASentin
el2" >
        <os:Query xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
            xmlns:eo="http://a9.com/-/opensearch/extensions/eo/1.0/"
            title="Restriction for application compatibility" role="subset"
eo:cloudCover="0"/>
      </ows:operation>
    </owc:offering>
    ...
  </entry>
  ...
</feed>
```

A catalogue entry can be represented as

```
<feed ...>
  ...
  <entry>

<id>http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3AIPT%3ASen
tinel2</id>
    <title>IPT Sentinel-2</title>
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/catalogue"
        title="This entry contains a catalogue as specified by Testbed-13 EOC Thread
"/>
    <content type="text">SENTINEL-2 is a European wide-swath, high-resolution, multi-
spectral imaging mission. The full mission specification of the twin satellites flying
in the same orbit but phased at 180°, is designed to give a high revisit frequency of
5 days at the Equator. SENTINEL-2 carries an optical instrument payload that sample 13
spectral bands: four bands at 10 m, six bands at 20 m and three bands at 60 m spatial
resolution. The orbital swath width is 290 km.</content>
    <link type="application/atom+xml" rel="enclosure"
        href=
"http://geo.spacebel.be/opensearch/request?httpAccept=application%2Fatom%2Bxml&amp;par
entIdentifier=EOP:IPT:Sentinel2"/>
    <link type="application/opensearchdescription+xml" rel="search"
        href=
"http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3AIPT%3ASentin
el2"/>
    <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/opensearch">
      <owc:operation code="search" method="GET" type=
"application/opensearchdescription+xml"
          href=
"http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3AIPT%3ASentin
el2">
        <os:Query xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
            xmlns:eo="http://a9.com/-/opensearch/extensions/eo/1.0/"
            title="Restriction for application compatibility" role="subset"
eo:cloudCover="0"/>
      </ows:operation>
    </owc:offering>
  </entry>
  ...
</feed>
```

The catalogue entry can include information about the preferred geographical area of the query using the georss:where element.

```
<feed ...>
  ...
  <entry>
    ...
    <georss:where>
      <gml:Polygon>
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList>38.4921 44.2699 38.6058 43.4414 37.5318 43.2089 37.4215
44.0128 38.4921 44.2699</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </georss:where>
    ...
  </entry>
  ...
</feed>
```

For the preferred temporal extent the date or range of dates relevant to the catalogue resource can be included inside a dc:date element. The content of this element SHALL conform to the "date-time" production of ISO-8601. An uppercase "T" character SHALL be used to separate date and time, and an uppercase "Z" character SHALL be present in the absence of a numeric time zone offset. To specify a range of dates the "/" character SHALL be used.

```
<feed ...>
  ...
  <entry>
    ...
    <dc:date>2009-01-23T09:08:56.000Z/2009-01-23T09:14:08.000Z</dc:date>
    ...
  </entry>
  ...
</feed>
```

If more than one catalogue entry is present the application package should use the OWS Context Active flag value (OGC 12-084r2) indicating to the client if the entry should be used by default.

```
<feed ...>
  ...
  <entry>
    ...
    <category scheme="http://www.opengis.net/spec/owc/active" term="true"/>
    ...
  </entry>
  ...
</feed>
```

If needed, the Application Package can use the OWS Context Folder (OGC 12-084r2) to define an organisation of the entries.

```
<feed ...>
  ...
  <entry>
    ...
    <category scheme="http://www.opengis.net/spec/owc/folder" term="IPT/SENTINEL-2"
label="Field Information"/>
    ...
  </entry>
  ...
</feed>
```

Although it is anticipated that only one application catalogue entry would appear in a context feed, it is possible to have multiple catalogue entries. To link the application catalogue to a given WPS parameter it should be used the ows:Metadata element to include the atom:link with the respective search catalogue.

```
<feed ...>
  ...
  <link rel="profile" href="http://www.opengis.net/spec/owc-atom/1.0/req/core"
      title="This file is compliant with version 1.0 of OGC Context"/>
  <link rel="profile" href="http://www.opengis.net/tb13/eoc"
      title="This file is compliant with Testbed-13 EOC Thread for Application
Packing"/>
  <entry>
    ...
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/catalogue"
        title="This entry contains an catalogue as specified by Testbed-13 EOC Thread
"/>
    ...
  </entry>
  ...
  <entry>
    <title>EOC Land Cover Application</title>
    <id>http://www.opengis.net/tb13/eoc/LandCover</id>
    <link rel="profile" href="http://www.opengis.net/tb13/eoc"
        title="This file is compliant with Testbed-13 EOC Thread for Application
Packing"/>
    ...
    <owc:offering code="http://www.opengis.net/tb13/eoc/docker">
      <owc:content type="text/plain">
registry.hub.docker.com/cnlspacebel/landcover</owc:content>
    </owc:offering>
    <!-- THE WPS PROCESS DESCRIPTION -->
    <owc:offering code="http://www.opengis.net/tb13/eoc/wpsProcessOffering">
      <owc:content type="application/xml">
        <wps:ProcessOffering>
```

```
          <wps:Process>
            <ows:Title>Land Cover Mapping</ows:Title>
            <ows:Abstract>Lang Cover Mapping is based on the Sentinel-2 processing
workflow generated for the F-TEP platform.</ows:Abstract>
            <ows:Identifier>LandCover</ows:Identifier>
            ...
            <wps:Input>
              <ows:Title>Sentinel-2 Image</ows:Title>
              <ows:Abstract>URL of Sentinel-2 Level 1C image product in the format
offered by AWS or IPT, with a size of up to multiple gigabytes.</ows:Abstract>
              <ows:Identifier>Image</ows:Identifier>
              <ows:Metadata>
                <atom:link rel="search" href=
"http://geo.spacebel.be/opensearch/description.xml?parentIdentifier=EOP%3SENTINEL-
HUB%3ASentinel2"/>
              </ows:Metadata>
              <wps:ComplexData>
                <wps:Format mimeType="text/directory" default="true"/>
              </wps:ComplexData>
            </wps:Input>
            ...
            <wps:Input>
              <ows:Title>Reference Data</ows:Title>
              <ows:Abstract>Representative training data set with land cover class
attributes, in OGR vector format supported by GDAL, such as ESRI shapefile, in a flat
zip structure containing .shp and the supporting files.</ows:Abstract>
              <ows:Identifier>ReferenceData</ows:Identifier>
              <ows:Metadata>
                <atom:link rel="search" href="...">...</atom:link>
              </ows:Metadata>
              <wps:ComplexData>
                <wps:Format mimeType="application/zip" encoding="base64" default="
true"/>
              </wps:ComplexData>
            </wps:Input>
            ...
          </wps:Process>
        </wps:ProcessOffering>
      </owc:content>
    </owc:offering>
  </entry>
</feed>
```

In this example, one catalogue is used to search for data for the "Image" input and another catalogue is used to search for data for the "ReferenceData" input.

It was also the case in the thread that different catalogues needed to be used for different platforms. For example, a different catalogue was used to search for data products on IPT POLAND than was used to search for data products on AWS. This was handled in the testbed using two different process offerings and thus two different applications packages. Another, untested, approach for handling this situation could have been to make use of the "about" attribute on the ows:Metadata element to discriminate one platform from the other.

7.5. Base Maps Offerings

An Application Package can contain any given number of entries identified as base maps. The base map is defined as an auxiliary layer of information that supports the Application Package user but is not considered essential to its execution. The base map entry has a profile referenced as *"http://www.opengis.net/tb13/eoc/basemap"*

```
<feed ...>
  ...
  <entry>
    ...
    <link rel="profile" href="http://www.opengis.net/tb13/eoc/basemap"
        title="This entry contains a Base Map resource as specified by Testbed-13 EOC
Thread"/>
    ...
  </entry>
  ...
</feed>
```

The base maps can be of any type of OGC service (for example, not just OGC Web Map Service but also OGC Web Feature Service and OGC Web Coverage Service) or non-OGC compliant service. This section shows an example for a WMS and a MapBox Service as base maps.

7.5.1. Web Map Layers

A base map originating from a OGC WMS Layer has a offering with the code attribute code equal to *"http://www.opengis.net/spec/owc-atom/1.0/req/wms"* and the operation with the attribute code equal to *"GETMAP"*. A complete entry is shown below.

```
<feed ...>
  ...
  <entry>
    ...
    <id>http://osm-demo.wheregroup.com/service?Layers=osm</id>
    <title>OSM-Demo WhereGroup Europe</title>
    <content>Open Street Map Demo from WhereGroup Europe</content>
    <rights>This service is intended for private and evaluation use only. The geodata
is licensed as Open Data Commons Open Database Lizenz (ODbL)
(http://opendatacommons.org/licenses/odbl/)</rights>
    <!-- Geographic Area of interest for the App -->
    <georss:where><gml:Polygon xmlns:gml="http://www.opengis.net/gml">
<gml:exterior><gml:LinearRing><gml:posList>-90 -180 90 -180 90 180 -90 180 -90 -
180</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></georss:where>
    <link type="image/png" rel="enclosure" href="http://osm-
demo.wheregroup.com/service?SERVICE=WMS&Version=1.1.1&REQUEST=getmap&SRS=EPSG:4326&wid
th=600&height=300&bbox=-20,35,40,65&format=image/png&Layers=osm&styles="/>
    <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wms">
      <owc:operation code="GETMAP" method="GET" type="image/png"
          href="http://osm-demo.wheregroup.com/service?SERVICE=WMS&Version=1.1.1
&REQUEST=getmap&SRS=EPSG:4326&width=600&height=300&bbox=-20,35,40,65&format=image/png
&Layers=osm&styles="/>
    </owc:offering>
    <category scheme="http://www.opengis.net/spec/owc/active" term="false"/>
    <category scheme="http://www.opengis.net/spec/owc/folder" term="OSM"/>
  </entry>
  ...
</feed>
```

7.5.2. Using non-OGC services as base maps

A base map can also come from non-OGC compliant services. In this case the offering code value should provide some indication of what API is being accessed and the operation code value should indicate, either specifically or notionally, what operation is being performed.

The following example shows how MapBox may be used as a base map. The offering code is set to *"https://api.mapbox.com/v4"* to indicate the version 4 of the MapBox API is being accessed. The operation code is set to *"retrieve-tiles"* indicating that the operation being performed is the retrieval of tiles. A complete MapBox example is show below.

```
<feed ...>
  ...
  <entry>
    ...
    <id>https://b.tiles.mapbox.com/v4/mapbox.satellite</id>
    <title>Mapbox Satellite</title>
    <content>Mapbox Satellite is our full global base map that is perfect as a blank
canvas or an overlay for your own data.</content>
    <rights>© &lt;a href=&quot;https://www.mapbox.com/feedback/&quot;&gt;Mapbox&lt;
/a&gt; © &lt;a href=&quot;http://www.openstreetmap.org/copyright&quot;
&gt;OpenStreetMap&lt;/a&gt;</rights>
    <!-- Geographic Area of interest for the App -->
    <georss:where><gml:Polygon xmlns:gml="http://www.opengis.net/gml">
<gml:exterior><gml:LinearRing><gml:posList>-90 -180 90 -180 90 180 -90 180 -90 -
180</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></georss:where>
    <link type="image/png" rel="enclosure" href=
"https://b.tiles.mapbox.com/v4/mapbox.satellite/0/0/0.png?access_token=pk.eyJ1IjoibWFw
Ym94IiwiYSI6ImNpejY4NDg1bDA1cjYzM280NHJ5NzlvNDMifQ.d6e-nNyBDtmQCVwVNivz7A"/>
    <category scheme="https://www.mapbox.com/api-documentation/#access-tokens" term=
"pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NDg1bDA1cjYzM280NHJ5NzlvNDMifQ.d6e-
nNyBDtmQCVwVNivz7A" label="Change for your access token"/>
    <owc:offering code="https://api.mapbox.com/v4">
        <owc:operation code="retrieve-tiles" method="GET" type="image/png"
            href=
"https://b.tiles.mapbox.com/v4/mapbox.satellite/{z}/{x}/{y}.png?access_token="/>
    </owc:offering>
    <category scheme="http://www.opengis.net/spec/owc/active" term="false"/>
    <category scheme="http://www.opengis.net/spec/owc/folder" term="mapbox/satellite
"/>
  </entry>
  ...
</feed>
```

7.6. Extensibility

An Application Package can contain any other number of entries. It is out of the scope of this ER to provide guidance for their interpretation.

Appendix A: Revision History

Table 4. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
October 29, 2017	1	P. Gonçalves	all	initial version

Appendix B: Bibliography

[1] OGC: OGC 17-024, OGC Testbed-13 Application Deployment and Execution Service ER (2018).