ADES Design Document

EOEPCA.SDD.xxx

TVUK System Team

Version 0.1, 14/04/2020:

ADES Design Document

EO Exploitation Platform Common Architecture

ADES Design Document

EOEPCA.SDD.xxx

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/proc-ades/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
0.1	14/04/2020	Initial in-progress draft

Chapter 1. Introduction

1.1. Purpose and Scope

This document presents the ADES Design for the Common Architecture.

1.2. Structure of the Document

Section 2 - Overview

Provides an overview of the ADES component, within the context of the wider Common Architecture design.

Section 3 - [mainDesign]

Provides the design of the ADES component.

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
[EOEPCA-MSD]	EOEPCA - Master System Design Document EOEPCA.SDD.001 https://eoepca.github.io/master-system-design/published/v1.0/	Issue 1.0, 02/08/2019
[EOEPCA-UC]	EOEPCA - Use Case Analysis EOEPCA.TN.005 https://eoepca.github.io/use-case-analysis	Issue 1.0, 02/08/2019
[EP-FM]	Exploitation Platform - Functional Model, ESA-EOPSDP-TN-17-050	Issue 1.0, 30/11/2017
[TEP-OA]	Thematic Exploitation Platform Open Architecture, EMSS-EOPS-TN-17-002	Issue 1, 12/12/2017
[WPS-T]	OGC Testbed-14: WPS-T Engineering Report, OGC 18-036r1, http://docs.opengeospatial.org/per/18-036r1.html	18-036r1, 07/02/2019
[WPS-REST-JSON]	OGC WPS 2.0 REST/JSON Binding Extension, Draft, OGC 18-062, https://raw.githubusercontent.com/opengeospatial/wps-rest-binding/develop/docs/18-062.pdf	1.0-draft

Reference	Document Details	Version
[CWL]	Common Workflow Language Specifications, https://www.commonwl.org/v1.0/	v1.0.2
[TB13-AP]	OGC Testbed-13, EP Application Package Engineering Report, OGC 17-023, http://docs.opengeospatial.org/per/17-023.html	17-023, 30/01/2018
[TB13-ADES]	OGC Testbed-13, Application Deployment and Execution Service Engineering Report, OGC 17-024, http://docs.opengeospatial.org/per/17-024.html	17-024, 11/01/2018
[TB14-AP]	OGC Testbed-14, Application Package Engineering Report, OGC 18-049r1, http://docs.opengeospatial.org/per/18-049r1.html	18-049r1, 07/02/2019
[TB14-ADES]	OGC Testbed-14, ADES & EMS Results and Best Practices Engineering Report, OGC 18-050r1, http://docs.opengeospatial.org/per/18-050r1.html	18-050r1, 08/02/2019
[OS-GEO-TIME]	OpenSearch GEO: OpenSearch Geo and Time Extensions, OGC 10-032r8, http://www.opengeospatial.org/standards/opensearchgeo	10-032r8, 14/04/2014
[OS-EO]	OpenSearch EO: OGC OpenSearch Extension for Earth Observation, OGC 13-026r9, http://docs.opengeospatial.org/is/13-026r8/13-026r8.html	13-026r9, 16/12/2016
[GEOJSON-LD]	OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard, OGC 17-003r1/17-084	17-003r1/17-084
[GEOJSON-LD-RESP]	OGC OpenSearch-EO GeoJSON(-LD) Response Encoding Standard, OGC 17-047	17-047
[PCI-DSS]	The Payment Card Industry Data Security Standard, https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss	v3.2.1
[CEOS-OS-BP]	CEOS OpenSearch Best Practise, http://ceos.org/ourwork/workinggroups/wgiss/access/opensearch/	v1.2, 13/06/2017

Reference	Document Details	Version
[OIDC]	OpenID Connect Core 1.0, https://openid.net/specs/openid-connect-core-1_0.html	v1.0, 08/11/2014
[OGC-CSW]	OGC Catalogue Services 3.0 Specification - HTTP Protocol Binding (Catalogue Services for the Web), OGC 12-176r7, http://docs.opengeospatial.org/is/12-176r7/12-176r7.html	v3.0, 10/06/2016
[OGC-WMS]	OGC Web Map Server Implementation Specification, OGC 06-042, http://portal.opengeospatial.org/files/?artifact_id=14416	v1.3.0, 05/03/2006
[OGC-WMTS]	OGC Web Map Tile Service Implementation Standard, OGC 07-057r7, http://portal.opengeospatial.org/files/?artifact_id=35326	v1.0.0, 06/04/2010
[OGC-WFS]	OGC Web Feature Service 2.0 Interface Standard – With Corrigendum, OGC 09-025r2, http://docs.opengeospatial.org/is/09-025r2/09-025r2.html	v2.0.2, 10/07/2014
[OGC-WCS]	OGC Web Coverage Service (WCS) 2.1 Interface Standard - Core, OGC 17-089r1, http://docs.opengeospatial.org/is/17-089r1/17-089r1.html	v2.1, 16/08/2018
[OGC-WCPS]	Web Coverage Processing Service (WCPS) Language Interface Standard, OGC 08-068r2, http://portal.opengeospatial.org/files/?artifact_id=32319	v1.0.0, 25/03/2009
[AWS-S3]	Amazon Simple Storage Service REST API, https://docs.aws.amazon.com/AmazonS3/latest/API	API Version 2006-03-01
[OPENAPI]	OpenAPI Specification, https://swagger.io/specification/	API Version 3.0.3 2020-02-20

1.4. Terminology

The following terms are used in the Master System Design.

Term	Meaning
Admin	User with administrative capability on the EP

Term	Meaning
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.
Analysis Result	The *Products* produced as output of an *Interactive Application* analysis session.
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*.
Application Artefact	The 'software' component that provides the execution unit of the *Application Package*.
Application Deployment and Execution Service (ADES)	WPS-T (REST/JSON) service that incorporates the Docker execution engine, and is responsible for the execution of the processing service (as a WPS request) within the ‘target’ Exploitation Platform.
Application Descriptor	A file that provides the metadata part of the *Application Package*. Provides all the metadata required to accommodate the processor within the WPS service and make it available for execution.
Application Package	A platform independent and self-contained representation of a software item, providing executable, metadata and dependencies such that it can be deployed to and executed within an Exploitation Platform. Comprises the *Application Descriptor* and the *Application Artefact*.
Bulk Processing	Execution of a *Processing Service* on large amounts of data specified by AOI and TOI.
Code	The codification of an algorithm performed with a given programming language - compiled to Software or directly executed (interpretted) within the platform.
Compute Platform	The Platform on which execution occurs (this may differ from the Host or Home platform where federated processing is happening)
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
Data Access Library	An abstraction of the interface to the data layer of the resource tier. The library provides bindings for common languages (including python, Javascript) and presents a common object model to the code.
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.

Term	Meaning
Discovery	User finds products/services of interest to them based upon search criteria.
Execution	The act to start a *Processing Service* or an *Interactive Application*.
Execution Management Service (EMS)	The EMS is responsible for the orchestration of workflows, including the possibility of steps running on other (remote) platforms, and the on-demand deployment of processors to local/remote ADES as required.
Expert	User developing and integrating added-value to the EP (Scientific Researcher or Service Developer)
Exploitation Tier	The Exploitation Tier represents the end-users who exploit the services of the platform to perform analysis, or using high-level applications built-in on top of the platform's services
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).
Guest	An unregistered User or an unauthenticated Consumer with limited access to the EP's services
Home Platform	The Platform on which a User is based or from which an action was initiated by a User
Host Platform	The Platform through which a Resource has been published
Identity Provider (IdP)	The source for validating user identity in a federated identity system, (user authentication as a service).
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 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.
Interactive Web Application	An Interactive Application for analysis provided as a rich user interface through the user's web browser.

Term	Meaning
Key-Value Pair	A key-value pair (KVP) is an abstract data type that includes a group of key identifiers and a set of associated values. Key-value pairs are frequently used in lookup tables, hash tables and configuration files.
Kubernetes (K8s)	Container orchestration system for automating application deployment, scaling and management.
Login Service	An encapsulation of Authenticated Login provision within the Exploitation Platform context. The Login Service is an OpenID Connect Provider that is used purely for authentication. It acts as a Relying Party in flows with external IdPs to obtain access to the user's identity.
EO Network of Resources	The coordinated collection of European EO resources (platforms, data sources, etc.).
Object Store	A computer data storage architecture that manages data as objects. Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier.
On-demand Processing Service	A *Processing Service* whose execution is initiated directly by the user on an ad-hoc basis.
Platform (EP)	An on-line collection of products, services and tools for exploitation of EO data
Platform Tier	The Platform Tier represents the Exploitation Platform and the services it offers to end-users
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.
Processing Result	The *Products* produced as output of a *Processing Service* execution.
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.
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*
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
Resource Tier	The Resource Tier represents the hosting infrastructure and provides the EO data, storage and compute upon which the exploitation platform is deployed
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
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.
Software	The compilation of code into a binary program to be executed within the platform on-line computing environment.
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).
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.
Transactional Web Processing Service (WPS-T)	Transactional extension to WPS that allows adhoc deployment / undeployment of user-provided processors.
User	An individual using the EP, of any type (Admin/Consumer/Expert/Guest)
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
Visualisation	To obtain a visual representation of any data/products held within the platform - presented to the user within their web browser session.
Web Coverage Service (WCS)	OGC standard that provides an open specification for sharing raster datasets on the web.
Web Coverage Processing Service (WCPS)	OGC standard that defines a protocol-independent language for the extraction, processing, and analysis of multi-dimentional coverages representing sensor, image, or statistics data.
Web Feature Service (WFS)	OGC standard that makes geographic feature data (vector geospatial datasets) available on the web.
Web Map Service (WMS)	OGC standard that provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.
Web Map Tile Service (WMTS)	OGC standard that provides a simple HTTP interface for requesting map tiles of spatially referenced data using the images with predefined content, extent, and resolution.
Web Processing Services (WPS)	OGC standard that defines how a client can request the execution of a process, and how the output from the process is handled.

Term	Meaning
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

1.5. Glossary

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

Term	Definition
AAI	Authentication & Authorization Infrastructure
ABAC	Attribute Based Access Control
ADES	Application Deployment and Execution Service
ALFA	Abbreviated Language For Authorization
AOI	Area of Interest
API	Application Programming Interface
CMS	Content Management System
CWL	Common Workflow Language
DAL	Data Access Library
EMS	Execution Management Service
EO	Earth Observation
EP	Exploitation Platform
FUSE	Filesystem in Userspace
GeoXACML	Geo-specific extension to the XACML Policy Language
IAM	Identity and Access Management
IdP	Identity Provider
JSON	JavaScript Object Notation
K8s	Kubernetes
KVP	Key-value Pair
M2M	Machine-to-machine
OGC	Open Geospatial Consortium
PDE	Processor Development Environment
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point

Term	Definition
RBAC	Role Based Access Control
REST	Representational State Transfer
SSH	Secure Shell
TOI	Time of Interest
UMA	User-Managed Access
VNC	Virtual Network Computing
WCS	Web Coverage Service
WCPS	Web Coverage Processing Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
WPS-T	Transactional Web Processing Service
XACML	eXtensible Access Control Markup Language

Chapter 2. Overview

2.1. Building Block Overview

Content Description

This section contains:

- High-Level Description of the Building Block
- Context within EOEPCA

The ADES provides a WPS-T 2.0.0 service that incorporates the Docker execution engine, and is responsible for the execution of the processing service (as a WPS request) within the 'target' Exploitation Platform (i.e. one that is close to the data). The ADES relies upon the EMS to ensure that the processor is deployed as a WPS service before it is invoked.

As illustrated in the EOEPCA Master System Design Document (ref. [EOEPCA-MSD]), the ADES provides an execution engine that handles the complexities of constructing the jobs and interfacing with the processing cluster. See the context wthin the EOEOPA, as depicted below in Figure 1:

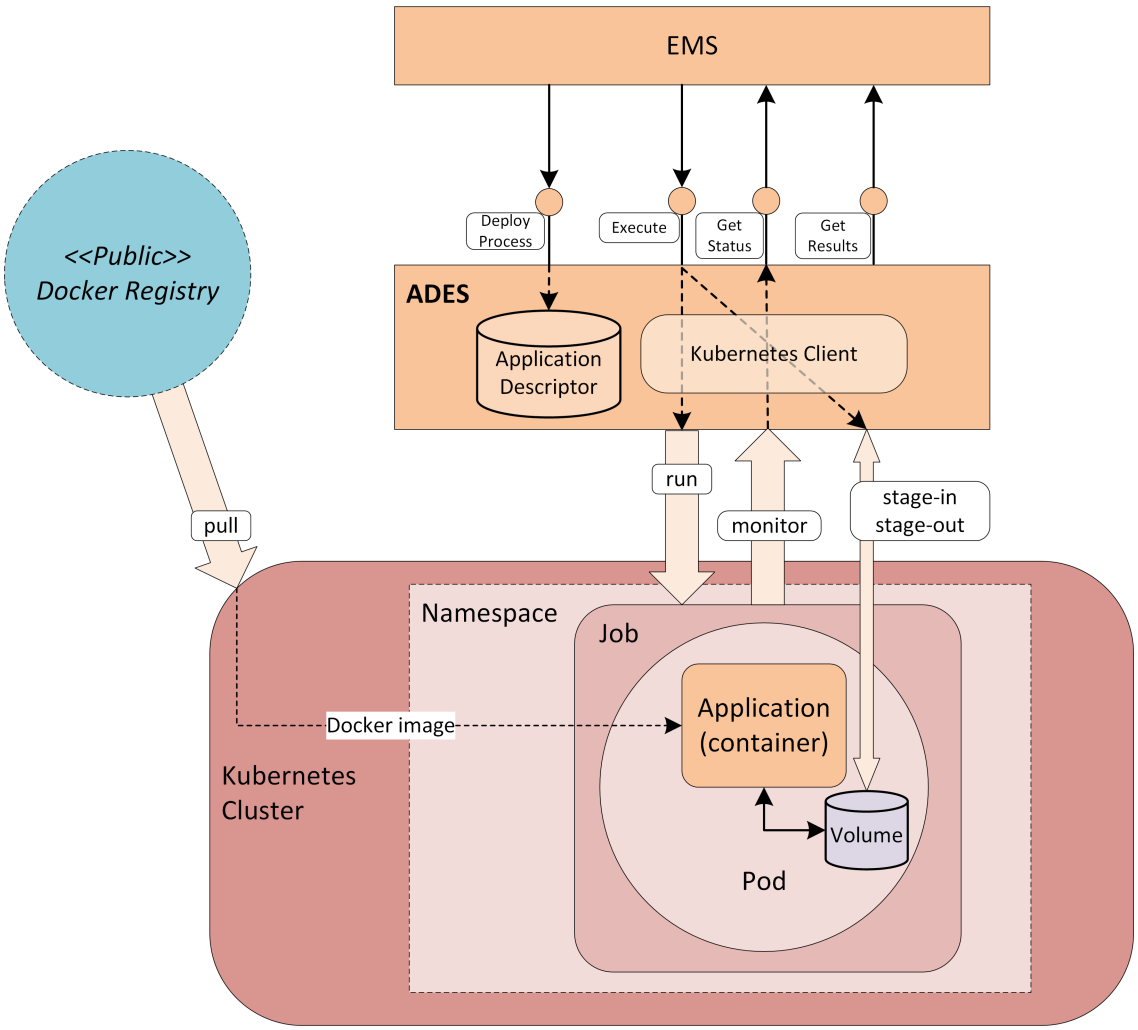

Figure 1. ADES Process Execution

The main responsibilities of the ADES are:

- Validate and accept an execution request from the EMS
- Submit the process execution to the processing cluster
- Monitor the process execution
- Retrieve the processing results

In order to accomplish the execution and monitor steps above, it also need to be responsible for the operations of:

- Data Stage-In for the process inputs
- Data Stage-Out for the process outputs

Those operations are internal sub-steps of the execution step.

2.1.1. Execution

The ADES performs an Execute operation upon a WPS ExecuteRequest, when instantiated by the EMS.

When performing the Execute operation, the ADES inspects the ExecuteRequest document, and instructs the underlying Kubernetes software, via a specific extension of the API named Argo Workflows, to run the following steps:

- Data Stage-In
- Processing
- Data Stage-Out

Argo Workflows is responsible for the internal orchestration of the three steps above.
Additionally, when instructing Kubernetes to launch the jobs (stage-in/out or processing), the ADES provides also the reference of the Docker image to use, which is automatically retrieved from the processing nodes if not already present.

2.1.1.1. Data Stage-In

Data Stage-In is the process to locally retrieve the inputs for the processing. Processing inputs are provided as EO Catalogue references and the ADES is responsible to translate those references into inputs available for the local processing.

ADES leverages an OpenSearch client when interacting with the EO Catalogue and standard libraries and tools for the local product retrieval (libcurl, s3cmd) to support the HTTP(S), FTP(S), FILE, OPeNDAP, and S3 protocols.

2.1.1.2. Processing

Processing is the core step of the Execute operation. During this step input data is transformed into outputs data.

ADES supports it by instructing the Kubernetes software, via its Argo extension, to download and use the configured Docker image, to provide the processing parameters and inputs, to execute the

configured processing command(s).

2.1.1.3. Data Stage-Out

Data Stage-Out is the process to upload remotely the outputs of the processing onto external system(s), and make them available for later usage.

ADES retrieves the processing outputs and automatically stores them onto an external persistent storage. Additionally, ADES publishes the metadata of the outputs onto a Catalogue, exposing the OpenSearch interface, and provides their references as an output.
Monitor

2.1.2. Monitor

ADES monitors a submitted execution at regular intervals and reports back the progress status via a GetStatus operation.

2.1.3. Dismiss

ADES can remove from the execution environment an alredy started processing job.

2.2. Static Architecture

Content Description

This section contains:

- Diagram and description of the major logical components within the Building Block

The ADES architecture is based on two major components:

- The WPS Server, exposing a WPS 2.0.0 (REST/JSON) interface [WPS-T] [WPS-REST-JSON]
- The ADES Core Engine, exposing an internal interface, compliant with the OpenAPI specification [OPENAPI]

Figure 2 shows an overview of the ADES building block. It provides the relationship between the WPS Server and the ADES core engine, and the interaction with the external Kubernetes cluster via the Argo Workflows API.

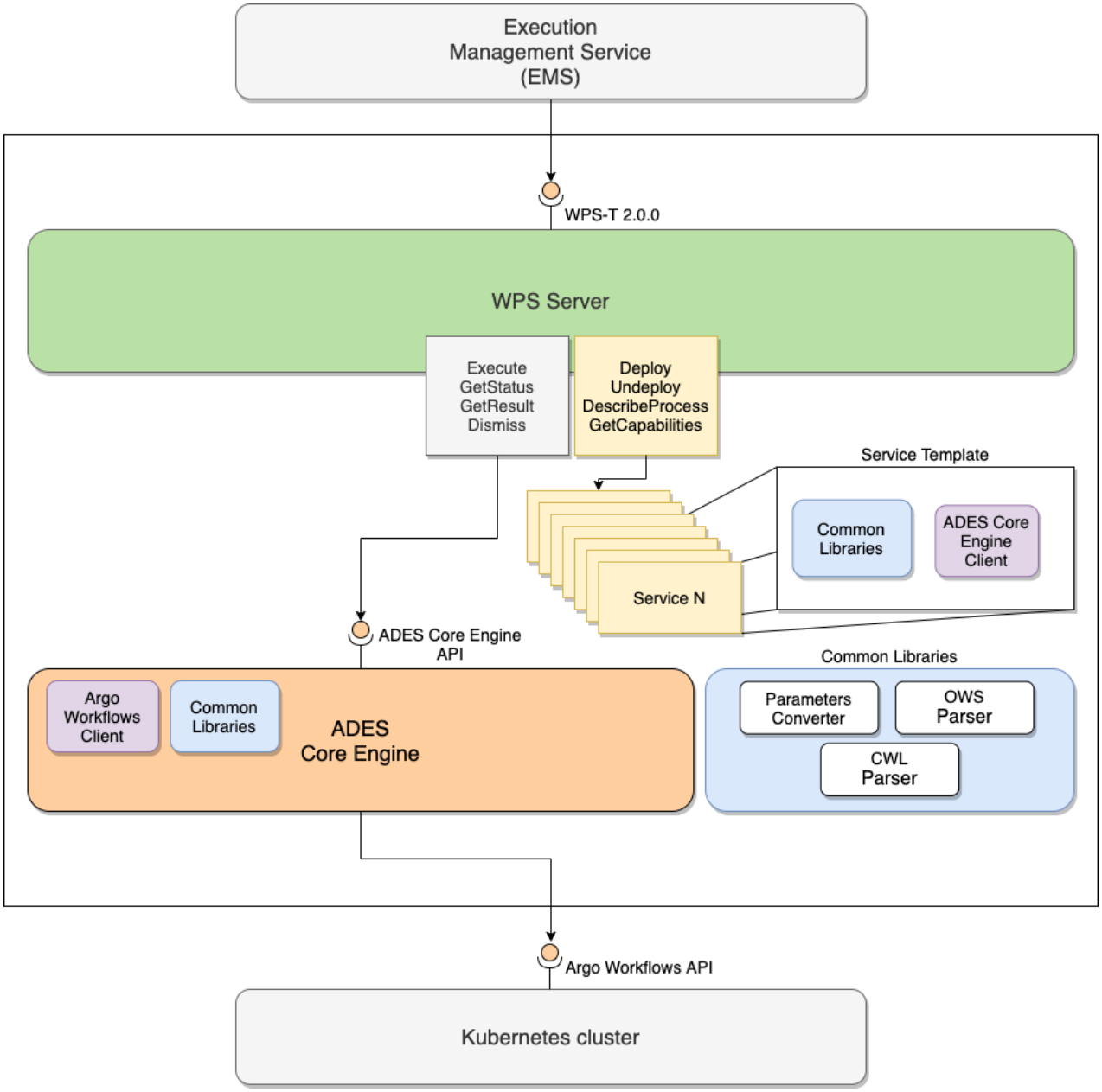

Figure 2. ADES building block overview

The WPS Server provides the external interface of the ADES, available to the EMS only. It relies on internal code and configuration for the GetCapabilites, DescribeProcess, Deploy/Undeploy WPS operations.
On the other hand, the WPS server interacts with the ADES Core Engine API for the Execute, GetStatus, GetResult and Dismiss WPS operations.
In turn, the ADES Core Engine contacts the Argo Workflows API to submit, monitor, retrieve the results and dismiss a processing execution.

Section Building Block Design contains a detailed description of the ADES components.

2.3. Use Cases

Content Description

This section contains:

- Diagrams and definition of the use cases covered by this Building Block

2.4. External Interfaces

Content Description

This section contains:

- Listing of technical external interfaces (with other Building Blocks)

2.4.1. Processing

The ADES exposes an OGC WPS-T 2.0.0 (REST/JSON) interface to deploy, execute, monitor and dismiss processing services.

2.4.2. AuthN / AuthZ

The ADES is a Protected Resource and its front-end is a Policy Enforcement Point (PEP) module. The PEP performs:

- A request for Authentication
  - The Authentication flow is implemented using the OpenID Connect 1.0 protocol
- A request for Authorization
  - The Authorization flow is implemented using the OAuth 2.0 protocol (only after a successful Authentication flow)

The PEP module is provided by **Task 1: User Management**, and it is part of the ADES architecture.

2.5. Required Resources

Content Description

This section contains:

- List of HW and SW required resources for the correct functioning of the building Block
- References to open repositories (when applicable)

2.5.1. Software

The following open-source software are required to support the implementation of the ADES:

- HTTP(S), FTP(S), FILE, OPeNDAP Client
  - libcurl https://curl.haxx.se/libcurl/

- OpenSearch Client
  - DotNetOpenSearchClient https://github.com/Terradue/DotNetOpenSearchClient
- Kubernetes Client
  - libcurl https://curl.haxx.se/libcurl/
- Kubernetes Extension
  - Argoproj https://argoproj.github.io/argo
- S3 Client
  - s3cmd https://github.com/s3tools/s3cmd
- WPS Server
  - ZOO-Kernel http://zoo-project.org/docs/kernel/

2.6. Design Standards, Conventions and Procedures

Content Description

This section contains:

- Explanations on the UML Design notation and necessary naming conventions used throughout the document

2.6.1. UML Design

2.6.2. Naming Conventions

Chapter 3. Building Block Design

Content Description

This section contains:

- A concise breakdown of the Building Block in several independent services (when applicable). For each component, the following subsections are added:
  - Overview and purpose: indicating the functionality covered by the component
  - SW Reuse and Dependencies: indicating reuse of third party open source solutions (if any) and any pre-required Dependencies
  - Interfaces: both internal to the building block and those exposed externally
  - Data: Data usage of the building block, data flow and any GDPR concerns should be addressed here
  - Applicable Resources: links and references to (Reference Docs), and repositories.

When a breakdown is necessary, a genral overview of the building block can be given. On the contrary, no breakdown indicates a single component development with the same expected sections.

3.1. WPS Server

3.1.1. Overview and Purpose

The WPS Server provides the external interface of the ADES, available to the EMS only.

3.1.1.1. WPS-T 2.0.0 compliance

"ZOO-Kernel implements and complies with the WPS 1.0.0 and the WPS 2.0.0 standards edited by the Open Geospatial Consortium".
The Dismiss operation, which is *"only available in WPS 2.0.0, it lets the client ask the server to stop a running service and remove any file it created"*, requires changes in the source code to support the EMS operations. The current ZOO-Kernel implementation relies on local processes which are being removed from the execution environment. This approach prevents the management of remote processes. In this context, the ZOO-Kernel third-party software is modified to allow managing remote execution via APIs, in particular when the remote process exposes an OGC WPS-T 2.0.0 interface. It is foreseen an upstream contribution to the open-source ZOO-Project project.

3.1.2. Software Reuse and Dependencies

The following open-source software is reused:

- WPS Server
  - ZOO-Kernel http://zoo-project.org/docs/kernel/

3.1.3. Interfaces

The WPS Server exposes an OGC WPS-T 2.0.0 (REST/JSON) interface to deploy, execute, monitor and dismiss processing services.

3.1.4. Data

Not Applicable.

3.1.5. Applicable Resources

- [WPS-T]
- [WPS-REST-JSON]

3.2. ADES Core Engine

3.2.1. Overview and Purpose

3.2.2. Software Reuse and Dependencies

3.2.3. Interfaces

3.2.4. Data

3.2.5. Applicable Resources

3.3. ADES Core Engine Client

3.3.1. Overview and Purpose

3.3.2. Software Reuse and Dependencies

3.3.3. Interfaces

3.3.4. Data

3.3.5. Applicable Resources

3.4. Argo Workflows Client

3.4.1. Overview and Purpose

3.4.2. Software Reuse and Dependencies

3.4.3. Interfaces

3.4.4. Data

3.4.5. Applicable Resources

3.5. OWS Context Parser

3.5.1. Overview and Purpose

3.5.2. Software Reuse and Dependencies

3.5.3. Interfaces

3.5.4. Data

3.5.5. Applicable Resources

3.6. CWL Parser

3.6.1. Overview and Purpose

3.6.2. Software Reuse and Dependencies

3.6.3. Interfaces

3.6.4. Data

3.6.5. Applicable Resources

3.7. Parameters Converter

3.7.1. Overview and Purpose

3.7.2. Software Reuse and Dependencies

3.7.3. Interfaces

3.7.4. Data

3.7.5. Applicable Resources

Chapter 4. Use Case Traceability

Content Description

This section contains:

- A traceability matrix against the use case analysis document of the project, indicating which components address each use case

<< End of Document >>