Master System Interface Control Document

EOEPCA.ICD.001

TVUK System Team

Version Draft for 0.1, In Progress:

Master System Interface Control Document

EO Exploitation Platform Common Architecture

Master System Interface Control Document

EOEPCA.ICD.001

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/master-system-icd/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	28/04/2020	Inistial in-progress draft

Chapter 1. Introduction

1.1. Purpose and Scope

This is the Master System Interface Control Document for the Common Architecture.

1.2. Structure of the Document

Section 2 - Context

Provides the context for Exploitation Platforms within the ecosystem of EO analysis.

Section 3 - Design Overview

Provides an overview of the Common Architecture and the domain areas.

Section 4 - User Management

Describes the User Management domain area.

Section 5 - Processing and Chaining

Describes the Processing & Chaining domain area.

Section 6 - Resource Management

Describes the Resource Management domain area.

Section 7 - Platform API

Describes the Platform API, covering all domain areas.

Section 8 - Web Portal Interfaces

Describes the Web Portal, covering all domain areas.

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-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

Reference	Document Details	Version
[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
[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

Reference	Document Details	Version
[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
[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

Reference	Document Details	Version
[AWS-S3]	Amazon Simple Storage Service REST API, https://docs.aws.amazon.com/AmazonS3/latest/API	API Version 2006-03-01

1.4. Terminology

The following terms are used in the Master System Interface Control Document.

Term	Meaning
Admin	User with administrative capability on the EP
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)

Term	Meaning
Consumer	User accessing existing services/products within the EP. Consumers may be scientific/research or commercial, and may or may not be experts of the domain
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.
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.

Term	Meaning
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.
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.
Network of EO 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.

Term	Meaning
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.
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.

Term	Meaning
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.
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

Term	Definition
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
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. Context

The Master System Interface Control Document provides an EO Exploitation Platform architecture that meets the service needs of current and future systems, as defined by the use cases described in [EOEPCA-UC]. These use cases must be explored under 'real world' conditions by engagement with existing deployments, initiatives, user groups, stakeholders and sponsors within the user community and within overlapping communities, in order to gain a fully representative understanding of the functional requirements.

The ICD takes into consideration existing precursor architectures (such as the Exploitation Platform Functional Model [EP-FM] and Thematic Exploitation Platform Open Architecture [TEP-OA]), including consideration of state-of-the-art technologies and approaches used by current related projects. The master system ICD describes functional blocks linked together by agreed standardised interfaces.

The importance of the OGC in these activities is recognised as a reference for the appropriate standards and in providing mechanisms to develop and evolve standards as required in the development of the architecture. In order to meet the design challenges we must apply the applicable existing OGC standards to the full set of federated use cases in order to expose deficiencies and identify needed evolution of the standards. Standards are equally important in all areas of the Exploitation Platform, including topics such as Authentication & Authorization Infrastructure (AAI), containerisation and provisioning of virtual cloud resources to ensure portability of compute between different providers of resource layer.

Data and metadata are fundamental considerations for the creation of an architecture in order to ensure full semantic interoperability between services. In this regard, data modelling and the consideration of data standards are critical activities.

The ICD must go beyond the provision of a standalone EO Exploitation Platform, by intrinsically supporting federation of similar EO platforms at appropriate levels of the service stack. The Network of EO Resources seeks, ‘to unite the available - but scattered - European resources in a large federated and open environment’. In such a context, federation provides the potential to greatly enhance the utilization of data and services and provide as stimulus for research and commercial exploitation. From the end-user point of view, the federated system should present itself as a single consolidated environment in which all the federated resources are made available as an integrated system. Thus, the system ICD must specify federation-level interfaces that support this data and service-level interoperability in such a way that is seamless to the end users.

The goal is to create an Integrated Data Exploitation Environment. Users will apply their workflows close to the hosted data, supplemented by their own data. Processing outputs may be hosted as new products that can themselves contribute to the global catalogue. This paradigm can then be extended to encompass the federated set of Exploitation Platforms within the Network of EO Resources. The result is a Federated, Integrated Data Analysis Environment.

A Reference Implementation of the full architecture will be developed to prove the concepts and also to provide an off-the-shelf solution that can be instantiated by future projects to implement their EO Exploitation Platform, thus facilitating their ability to join the federated Network of EO Resources. **Thus, the Reference Implementation can be regarded as a set of re-usable platform**

services, in the context of a re-usable platform architecture.

Chapter 3. Design Overview

The overall system ICD has been considered by taking the 'Exploitation Platform – Functional Model' [EP-FM] as a starting point and then evolving these ideas in the context of existing interface standards (with some emphasis on the OGC protocol suite) and the need for federated services.

3.1. Domain Areas

The system architecture is designed to meet the use cases as defined in [EOEPCA-UC] and [EP-FM]. [EOEPCA-UC] makes a high-level analysis of the use-cases to identify the main system functionalities organised into domain areas: 'User Management', 'Processing & Chaining' and 'Resource Management'. The high-level functionalities are often met by more than one domain area, and User Management (specifically Identity & Access Management) cuts across all use cases, and forms the basis of all access control restrictions that are applied to platform services and data.

Figure 1 depicts the domain areas as top level component blocks in a Platform 'A'. The arrows may be read as "uses", each implying one or more service interfaces.

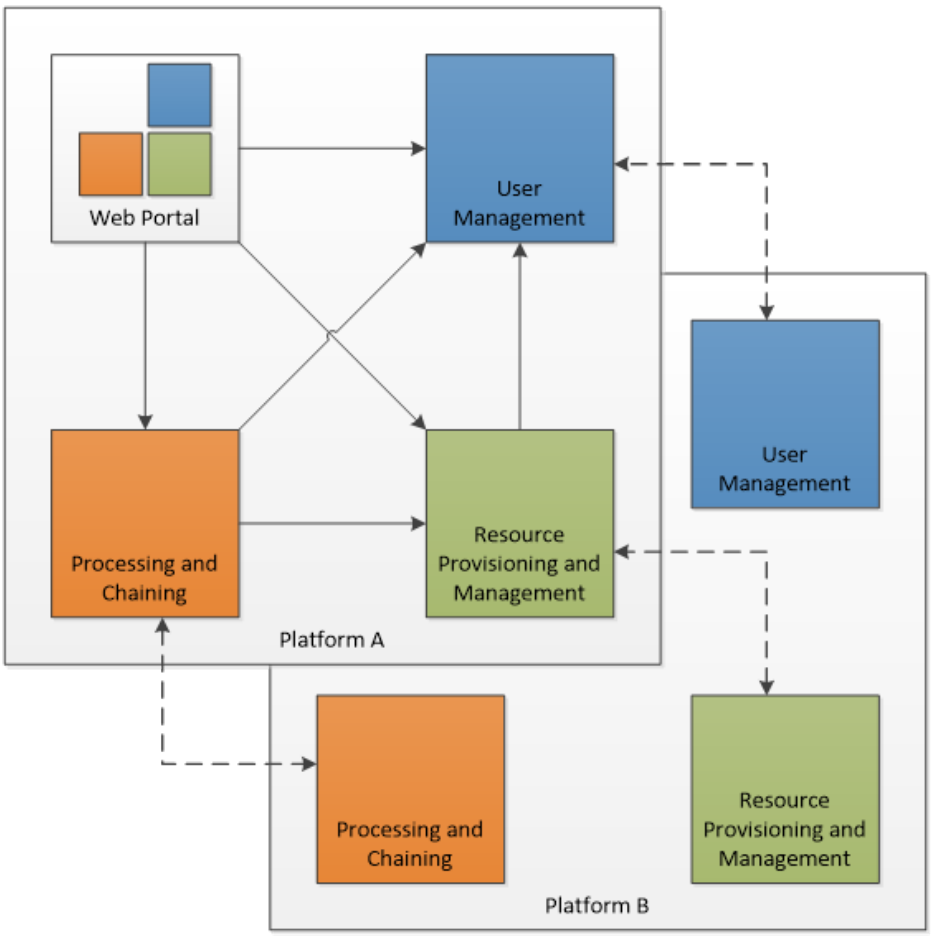


Figure 1. Top-level Architecture

A potential federation concept is represented by interactions between corresponding blocks in a collaborating Platform 'B'. The architecture aims to minimise dependencies and is conducive to the principle of subcontracting the implementation to experts in the respective domains. The web portal integrates various client components to form a rich user-facing interface. **The Web Portal is depicted as it has interfaces with the other domain areas - but it is not a priority concern for**

the Common Architecture. Each exploitation platform would be expected to develop its own web interfaces according to its needs.

3.1.1. User Management

Responsible for all aspects related to user identity, authentication, authorization, accounting and billing in a federated system-of-systems environment.

It provides authentication, authorization and accounting services that are required to access other elements of the architecture, including the processing-oriented services and resource-oriented services. Individual resources may have an associated access and/or charging policy, but these have to be checked against the identity of the user. Resource consumption may also be controlled e.g. by credits and/or quotas associated with the user account. In the Network of EO Resources, a user should not need to create an account on multiple platforms. Therefore some interactions will be required between the User Management functions, whether directly or in directly via trusted third party.

3.1.2. Processing and Chaining

Provides access to a variety of processing functions, tools and applications, as well as execution environments in which to deploy them.

Provides a deployment and execution environment for processing tasks, analysis tools and interactive applications. Supports the chaining of processing tasks in workflows whose execution can include steps executed external to the origin exploitation platform. Handles and abstracts the low-level complexities of the different underlying compute technologies, and ensures the compute layer is scaled in accordance with current demand. Provides an integrated development environment to facilitate development of new processing algorithms and applications. Facilitating the network of EO resources by providing a federated interface to other processing services within the wider EO network.

The development and analysis environment provides a platform for the expert user to develop their own processing chains, experiments and workflows. It integrates with platform catalogue services (for data, processing services and applications) for discovery of available published datasets and processing elements. Subject to appropriate controls and permissions, the user can publish their own processing services and results. Workflows can be executed within the context of the processing facility, with the possibility to execute steps ‘remotely’ in collaborating platforms, with the results being collected for the continuation of the workflow.

3.1.3. Resource Management

Responsible for maintaining an inventory of platform and federated resources, and providing services for data access and visualisation.

Storage and cataloguing of all persistent resources. First and foremost, this will contain multidimensional geo-spatial datasets. In addition it may include a variety of heterogeneous data and other resources, such as documentation, Docker images, processing workflows, etc. Handles and abstracts the low-level complexities of different underlying storage technologies and strategies. Facilitating the network of EO resources by providing a federated interface to other data services

within the wider EO network.

The catalogue holds corresponding metadata for every published resource item in the local platform storage, as well as entries for resources that are located on remote collaborating platforms. Catalogue search and data access is provided through a range of standard interfaces, which are used by the local Web Portal and Processing & Chaining elements and may be exposed externally as web services.

Access to services and resources is controlled according to an associated authorization policy as defined by the IAM approach. This component may interact with corresponding peer components on other platforms - for example to synchronise catalogue entries.

The user has a personal workspace in which to upload files, organise links to resources of interest (services/application/data), and receive/manage results output from processing executions. Shared workspaces for collaboration can be similarly provisioned. The ingestion of new data is controlled to ensure the quality of any published resource, including associated metadata, and to maintain the integrity of the catalogue.

3.1.4. Platform API

Defines standard interfaces at both service and programmatic levels.

The Service API and its associated Client Library together present a standard platform interface against which analysis and exploitation activities may be developed, and through which platform services can be federated. The Platform API encourages interoperation between platforms and provides a consistent and portable programming paradigm for expert users.

3.1.5. Web Portal

Presents the platform user interface for interacting with the local resources and processing facilities, as well as the wider network of EO resources.

The Web Portal provides the user interface (themed and branded according to the owning organisation) through which the user discovers the data/services available within the platform, and the analysis environment through which they can exploit these resources. It provides a rich, interactive web interface for discovering and working with all kinds of resources, including EO data, processing and documentation. It includes web service clients for smart search and data visualisations. It provides a workspace for developing and deploying processing algorithms, workflows, experiments and applications, and publishing results. It includes support and collaboration tools for the community.

Web Portal integrates together various web service clients that uses services provided by the specialist domains (Processing, Resource, User) on the local platform and collaborating platforms.

3.2. Architecture Layers

Figure 2 provides a simplified architectural view that illustrates the broad architecture layes of the Exploitation Platform, presented in the context of the infrastructure in which it is hosted and the end-users performing exploitation activities.

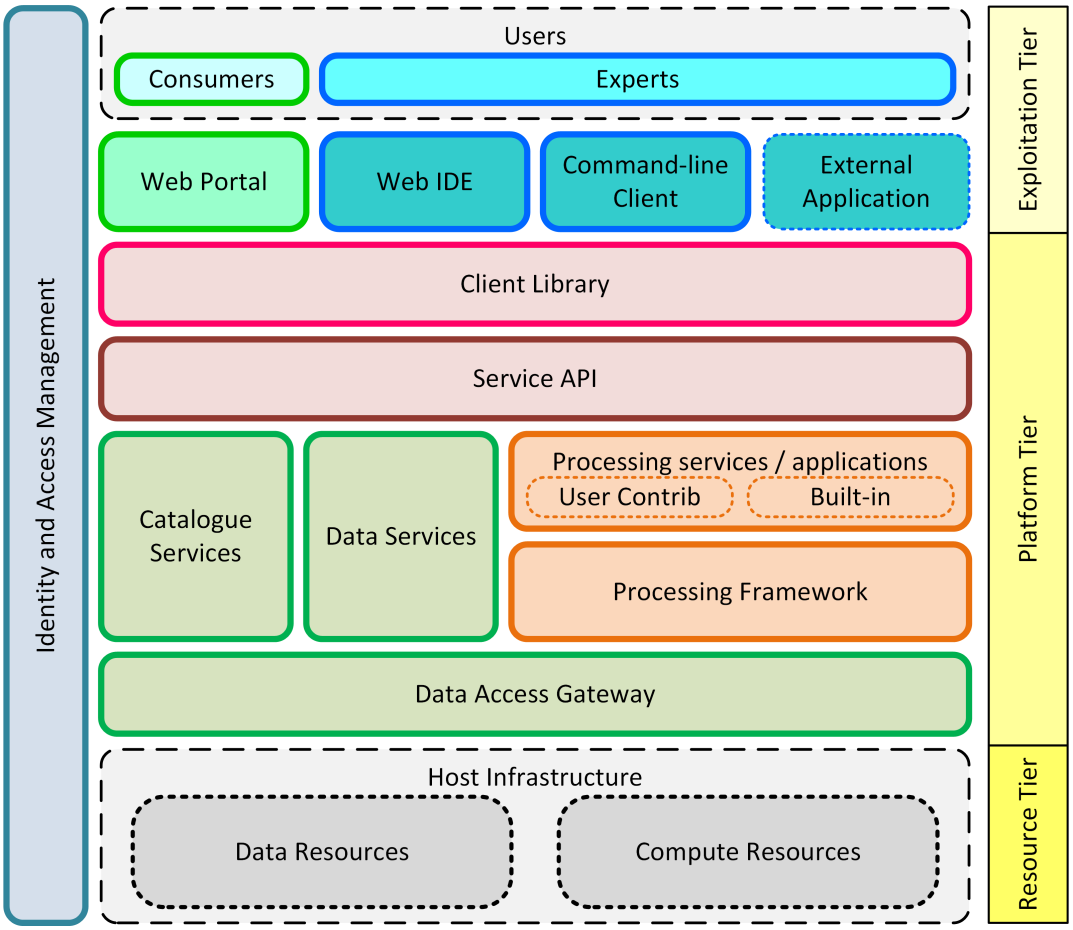


Figure 2. Architecture Layers

Resource Tier

The Resource Tier represents the hosting infrastructure and provides the EO data, storage and compute upon which the exploitation platform is deployed.

Platform Tier

The Platform Tier represents the Exploitation Platform and the services it offers to end-users. The layers comprising the Platform Tier are further described below.

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.

The Exploitation Platform builds upon the services provided by the hosting infrastructure - specifically accessing its data holding and using its compute resources. The components providing the EP services are deployed within the compute offering, with additional compute resources being provisioned on-demand to support end-user analysis activities.

The EP's services access the data resources through a Data Access Gateway that provides an

abstraction of the data access interface provided by the resource tier. This abstraction provides a 'standard' data access semantic that can be relied upon by other EP services - thus isolating specific data access concerns of the resource tier to a single EP component.

The Processing Framework provides the environment through which processing services and applications are executed in support of end-user analysis activities. It might be envisaged that some built-in (common) processing functions are provided, but the main focus of the processing framework is to support deployment and execution of bespoke end-user processing algorithms, and interactive analysis. Access to the underlying data from the executing processes is marshalled through the Data Access Gateway and its supporting Data Access Library.

The EP provides Catalogue services, so that end-users can discover and browse the resources available in the platform and its federated partners. Thus, end-users can discover available processing services and applications, and search for data available for inclusion in their analysis.

Data Services based upon open standards serve the clients of the Exploitation Platform for data access and data visualisation. Access to the underlying data is made via the Data Access Gateway.

The Service API represents the public service interfaces exposed by the Exploitation Platform for consumption by its clients. Covering all aspects of the EP (authentication, data/processing discovery, processing etc.), these interfaces are based upon open standards and are designed to offer a consistent EP service access semantic within the network of EO resources. Use of the network (HTTP) interfaces of the Service API is facilitated by the Client Library that provides bindings for common languages (Python, R, Javascript). The Client Library is a programmatic representation of the Service API which acts as an abstraction of the Exploitation Platform and so facilitates the development of portable client implementations.

The Exploitation Tier hosts the web clients with which the end-user interacts to conduct their analysis/exploitation activities. These clients would typically utilise the Client Library in their implementation. The Web IDE is an interactive web application that Experts use to perform interactive research and to develop algorithms. The Command-line Client builds upon the Client Library to provide a command-line tool that can be used, for example, to automate EP interactions through scripts.

The Web Portal provides the main user interface for the Exploitation Platform. It would be expected that each platform would provide its own bespoke portal implementation, and so is beyond the scope of the Common Architecture. Nevertheless, the architecture and its service interface must meet the needs anticipated by future exploitation platform implementations. Similarly, the External Application represents web applications (external to the hosting environment of the exploitation platform) that use the services of the EP via its Service API and Client Library.

All user interactions with the services of the EP are executed within the context of a given user and their rights to access resources, with associated resource usage billing. Thus, the Identity and Access Management component covers all tiers in this layered model.

The focus of this design document is the Platform Tier, which is elaborated in subsequent sections of the document:

Section User Management

This section adresses the main concerns of User Management which are user identity, access to resources and billing for resource usage.

Section Processing and Chaining

This section covers application packaging and the Processing Framework through which services/applications can be deployed in federated workflows.

Section Resource Management

This section covers resource discovery through catalogues that act as a marketplace for data, services and applications. Resource Management ensures data is accessible through standard interfaces that serve the processing framework, and public data services to visualise and consume platform data.

Section Platform API

This section provides a consolidated decription of the service interface of the EP and its associated client library, which together present a standard platform interface against which analysis and exploitation activities may be developed, and through which platform services can be federated.

3.3. System interfaces

Figure 3 provides a simplified structural view that illustrates the broad interactions between parts of the Exploitation Platform and its external actors, presented in the context of the infrastructure in which it is hosted and the end-users performing exploitation activities.

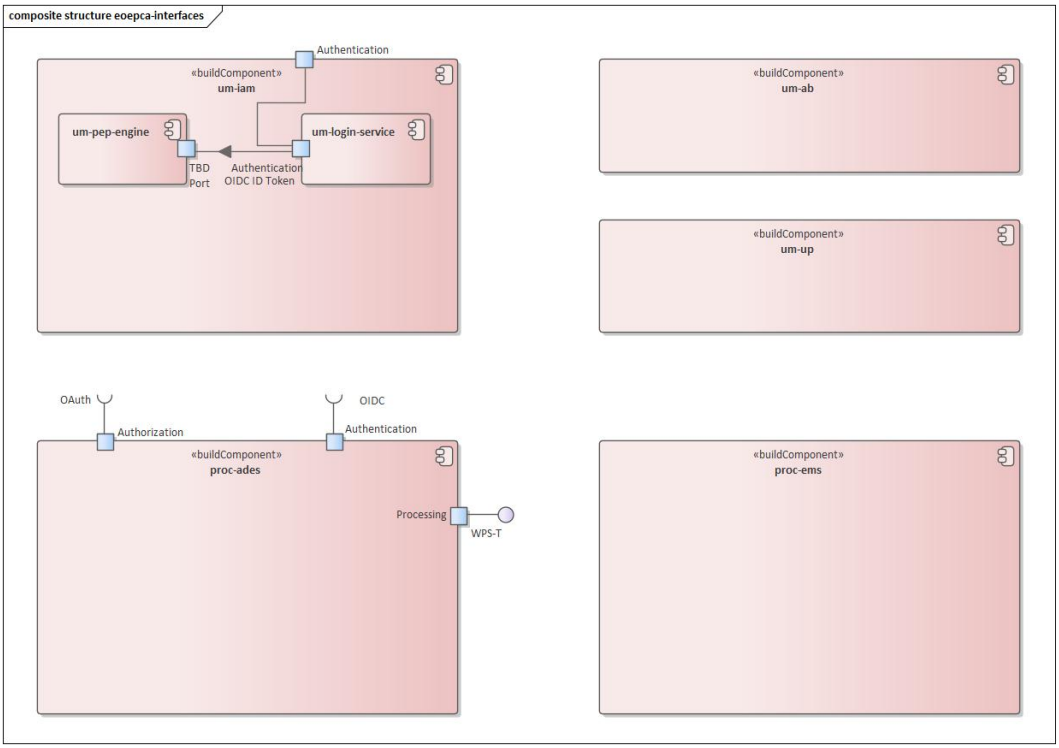


Figure 3. System interfaces

Chapter 4. User Management

4.1. User Management interfaces

Work In Progress

This section focuses on the externally available functionality published over interfaces, and how these interfaces connect with external systems.
Each building block contains:

- Description of functionality
- Component diagram
- List of exposed interfaces by URL
- List of services

Each service contains:

- Description of functionality
- Component diagram
- List of exposed interfaces by URL

Each interface in the list contains:

- Unique URI (*)
- Description - (TBD)
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples (TBD)

(*) URL would be formed as [scheme]://[building-block].[eoepca.org]/[endpoint], e.g. https://iam.eoepca.org/logout

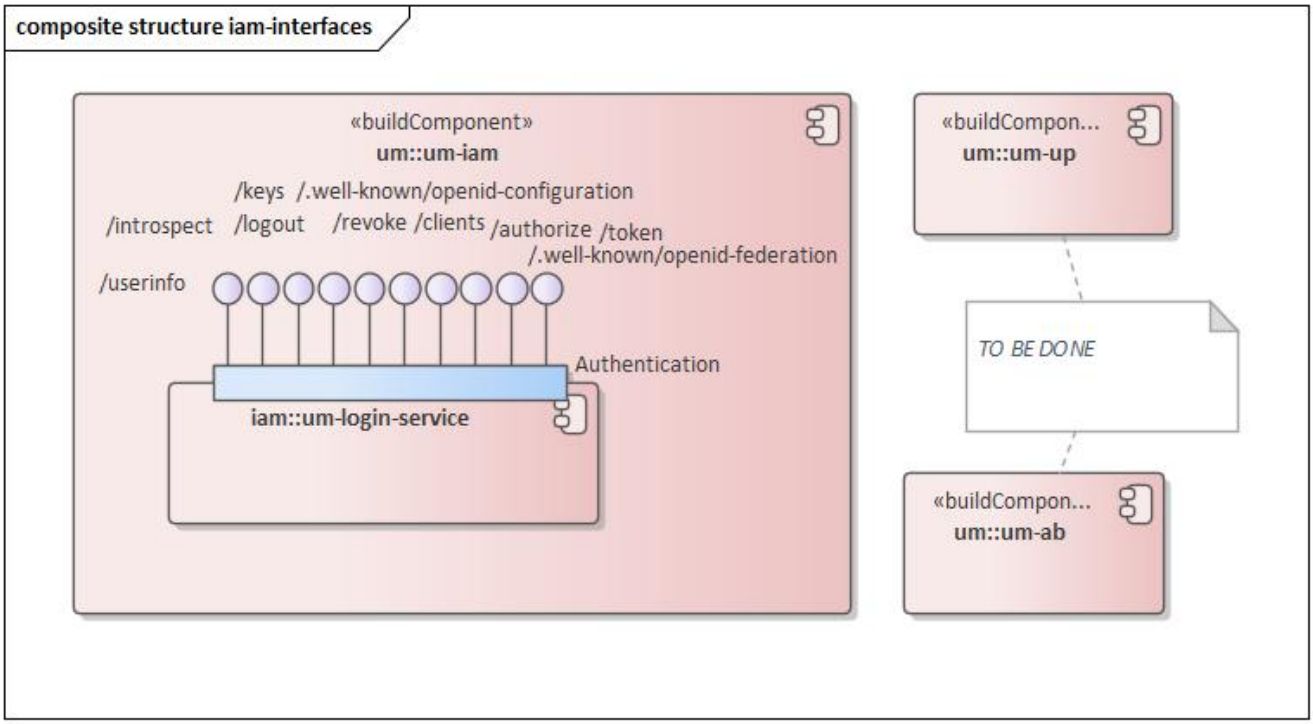
4.1.1. Identity and Authentication management

- Description of functionality

The solution for IAM is driven by the need for Federated Identity and Authorization in the context of a network of collaborating exploitation platforms and connected services. This federated environment should facilitate an end-user experience in which they can use a single identity across collaborating platforms (Single Sign-On), they can bring their own existing identity to the platforms ('Login With' service), and platforms can access the federated services of other platforms on behalf of the end-user (delegated access and authorization).

The goal of IAM is to uniquely identify the user and limit access to protected resources to those having suitable access rights. We assume an Attribute Based Access Control (ABAC) approach in which authorization decisions are made based upon access policies/rules that define attributes required by resources and possessed (as claims) by users. ABAC is is seen as a more flexible approach than Role Base Access Control (RBAC), affording the ability to express more sophisticated authorizations rules beyond the role(s) of the user - and noting the fact that a role-based ruleset could be implemented within an attribute based approach, (i.e. RBAC is a subset/specialisation of ABAC).

- Component diagram



- Link to building block documentation (TBD)

Table 2. IAM interfaces

URL	Name
https://iam.eoepca.org/authorize	[authorization]
https://iam.eoepca.org/token	[token]
https://iam.eoepca.org/userinfo	[userinfo]
https://iam.eoepca.org/keys	[jwks]
https://iam.eoepca.org/logout	[end]
https://iam.eoepca.org/introspect	[introspection]
https://iam.eoepca.org/revoke	[revocation]
https://iam.eoepca.org/.well-known/openid-configuration	[discovery]
https://iam.eoepca.org/clients	[registration]

URL	Name
https://iam.eoepca.org/.well-known/openid-federation	[federation]

Table 3. IAM conveyed information

Information	Interfaces
[oidc_id_token]	[userinfo]

4.1.1.1. IAM interfaces

The primary endpoints required to support the OIDC flows are as follows (these endpoints are taken, by example, from OKTA OIDC discovery metadata, https://micah.okta.com/oauth2/aus2yrcz7aMrmDAKZ1t7/.well-known/openid-configuration):

URI: https://iam.eoepca.org/authorize

- Description - To initiate the authentication, and to return the access tokens / code grant (depending on flow).
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples (TBD)

URI: https://iam.eoepca.org/token

- Description - To exchange the code grant for the access tokens.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

URI: https://iam.eoepca.org/userinfo

- Description - To obtain the user information ID token in accordance with the scopes requested in the authorization request.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - Returns an [oidc_id_token] that asserts a user's authenticated identity with integrity, and non-repudiation.
- Control and data flow descriptions - (TBD)

- Usage examples - (TBD)

URI: https://iam.eoepca.org/keys

- Description - To obtain signing keys for Token validation purposes.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

URI: https://iam.eoepca.org/logout

- Description - To logout the user from the Login Service, i.e. clear session cookies etc. Although, given that the actual IdP is externalised from the Login Service, it would remain the case that any session cookies maintained by the external IdP would still be in place for a future authentication flow.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

URI: https://iam.eoepca.org/introspect

- Description - Used by clients to verify access tokens.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

URI: https://iam.eoepca.org/revoke

- Description - Used for (refresh) token revocation.
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

The following endpoints relate to Discovery:

URI: https://iam.eoepca.org/.well-known/openid-configuration

- Description - Dynamic discovery of OIDC endpoints by clients.

OpenID Connect makes provision for two types of discovery:

- Discovery of the OpenID Provider Issuer based upon the user's identifier
- Discovery of the OpenID Provider Configuration Information

In the case of our usage within the Exploitation Platform, type 1) is not application since the user's ID comes from their 'Home' organisation and is not (necessarily) tied to an OpenID Connect Provider. Instead the Login Service must implement a discovery 'flow' in which the user is able to select the provider of their identity, as one that is supported by the Login Service deployment.

Regarding discovery type 2), the Login Service exposes an OIDC Provider interface, and this should support retrieval of OIDC Provider Configuration Information. Thus, OIDC Clients can utilise the discovery interface of the Login Service to exploit its services.

This is of most interest in the case of access to federated resources in other EPs, where a resource server in one EP may be acting as an OIDC client of the Login Service in another EP – in which case auto-discovery might be more attractive.

- Applicable standards - https://openid.net/specs/openid-connect-discovery-1_0.html
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

The following endpoints relate to Dynamic Client Registration:

URI: https://iam.eoepca.org/clients

- Description - Dynamic registration of clients (Authentication Agents).

The possibility exists for the OIDC Client (Login Service) to perform auto-registration with the Login Service, using OIDC Client Registration. In doing so the OIDC client obtains its Client ID and Secret.

This may be of interest in a couple of cases:

- The case of access to federated resources in other EPs, where a resource server in one EP may be acting as an OIDC client of the Login Service in another EP – in which case auto-client-registration might be of interest.
- The case where a common Login Service is deployed outside of the context of a given Exploitation Platform, acting as an IdP Proxy. In this case, the local Login Service deployed in each EP would register as an OIDC Client of the IdP Proxy.
    - Applicable standards - https://openid.net/specs/openid-connect-registration-1_0.html
    - Operations - (TBD)

- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

The following endpoints relate to the establishment of a federation of collaborating Exploitation Platforms through a dynamic trust model:

URI: https://iam.eoepca.org/.well-known/openid-configuration

- Description - OIDC Federation API endpoint through which Entity Statements are published about itself and other entities (such as other Exploitation Platforms).
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples - (TBD)

4.1.1.2. IAM information

OICD ID token

The ID Token is a JWT that is returned to from the /userinfo endpoint of the Login Service. The returned OIDC ID Token has been signed (JWS) by the Login Service and thus results in a token that asserts a user's authenticated identity with integrity, and non-repudiation.

4.1.2. Accounting and Billing

- Description of functionality
- Component diagram
- Link to building block documentation
- List of exposed interfaces

Table 4. AB interfaces

URL	Name
Cell 1 1	Cell 2 1
Cell 1 2	Cell 2 2

- List of services

4.1.3. User profile

- Description of functionality
- Component diagram
- Link to building block documentation

- List of exposed interfaces

Table 5. UP interfaces

URL	Name
Cell 1 1	Cell 2 1
Cell 1 2	Cell 2 2

- List of services

Chapter 5. Processing and Chaining

5.1. Processing and Chaining interfaces

Work In Progress

This section focuses on the externally available functionality published over interfaces, and how these interfaces connect with external systems.
Each building block contains:

- Description of functionality
- Component diagram
- List of exposed interfaces by URL
- List of services

Each service contains:

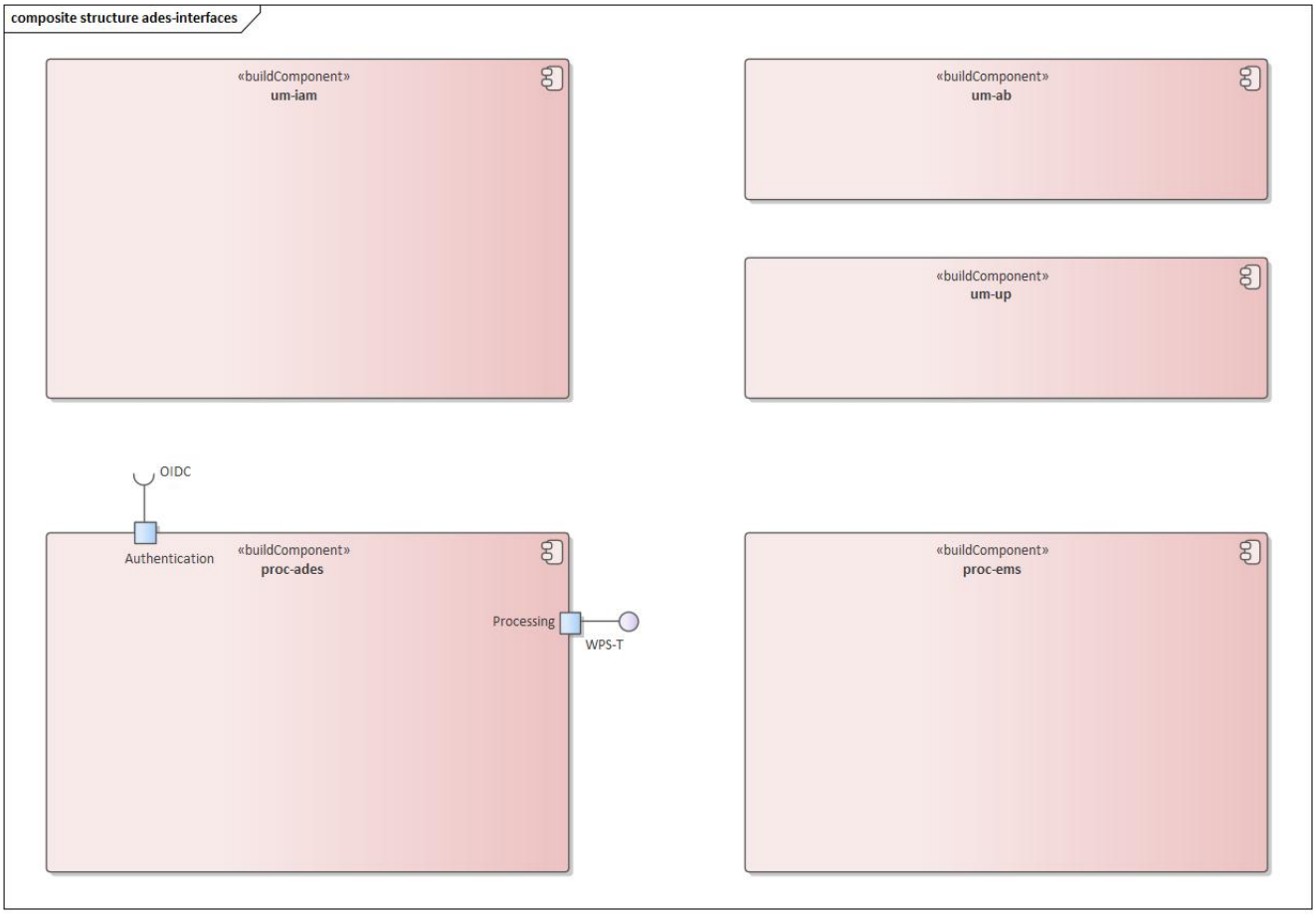
- Description of functionality
- Component diagram
- List of exposed interfaces by URL

Each interface in the list contains:

- Unique URI (*)
- Description - (TBD)
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples (TBD)

(*) URL would be formed as [scheme]://[building-block].[eoepca.org]/[endpoint], e.g. https://iam.eoepca.org/logout

- Description
- Component diagram



- Link to building block documentation

Table 6. ADES interfaces

URL	Operation	Name
https://ades.eoepca.org/	GET	[landing]
https://ades.eoepca.org/processes	GET	[get capabilities]
https://ades.eoepca.org/processes	POST	[deploy process]
https://ades.eoepca.org/processes/{id}	GET	[describe process]
https://ades.eoepca.org/processes/{id}	DELETE	[undeploy process]
https://ades.eoepca.org/processes/{id}/jobs	GET	[get jobs]
https://ades.eoepca.org/processes/{id}/jobs	POST	[execute]
https://ades.eoepca.org/processes/{id}/jobs/{jobID}	GET	[get status]
https://ades.eoepca.org/processes/{id}/jobs/{jobID}	DELETE	[dismiss]

URL	Operation	Name
https://ades.eoepca.org/processes/{id}/jobs/{jobID}/result	GET	[get result]
https://ades.eoepca.org/processes/{id}/quotations	GET	[get quotations list]
https://ades.eoepca.org/processes/{id}/quotations	POST	[request a process quotation]
https://ades.eoepca.org/processes/{id}/quotations/{quotationID}	GET	[retrieve quotation information]
https://ades.eoepca.org/processes/{id}/quotations/{quotationID}	POST	[execute quoted process]
https://ades.eoepca.org/processes/{id}/visibility	GET	[retrieve visibility status]
https://ades.eoepca.org/processes/{id}/visibility	PUT	[change visibility status]
https://ades.eoepca.org/quotations	GET	[retrieve list of all quotations]
https://ades.eoepca.org/quotations/{quotationID}	GET	[retrieve quotation information]
https://ades.eoepca.org/quotations/{quotationID}	POST	[execute a quoted process]
https://ades.eoepca.org/bills	GET	[retrieve list of all bills]
https://ades.eoepca.org/bills/{billID}	GET	[retrieve bill information]
https://ades.eoepca.org/conformance	GET	[retrieve list of all requirements classes]

Table 7. ADES conveyed information

Information	Interfaces
[data]	[tbd]

5.1.1. ADES interfaces

5.1.1.1. Processing

This interface specification is used for both the Client < - > EMS, and the EMS < - > ADES interfaces.

WPS-T extends standard WPS by adding DeployProcess and UndeployProcess operations. Once a process has been deployed to a WPS then the existing wps:Execute operation remains applicable for

execution in the standard way.

URI:

- Description - (TBD)
- Applicable standards - (TBD)
- Operations - (TBD)
- Conveyed information - (TBD)
- Control and data flow descriptions - (TBD)
- Usage examples (TBD)

Chapter 6. Resource Management Interfaces

Chapter 7. Platform API

The Platform API defines standard interfaces at both service and programmatic levels, with the goal of encouraging interoperation between platforms and providing a consistent and portable programming paradigm for expert users. The Service API and its associated Client Library together present a standard platform interface against which analysis and exploitation activities may be developed, and through which platform services can be federated.

The Service API represents the public service interfaces exposed by the Exploitation Platform for consumption by its clients. Covering all aspects of the EP (authentication, data/processing discovery, processing etc.), these interfaces are based upon open standards and are designed to offer a consistent EP service access semantic within the network of EO resources. Use of the network (HTTP) interfaces of the Service API is facilitated by the Client Library that provides bindings for common languages (Python, R, Javascript). The Client Library is a programmatic representation of the Service API which acts as an abstraction of the Exploitation Platform and so facilitates the development of portable client implementations.

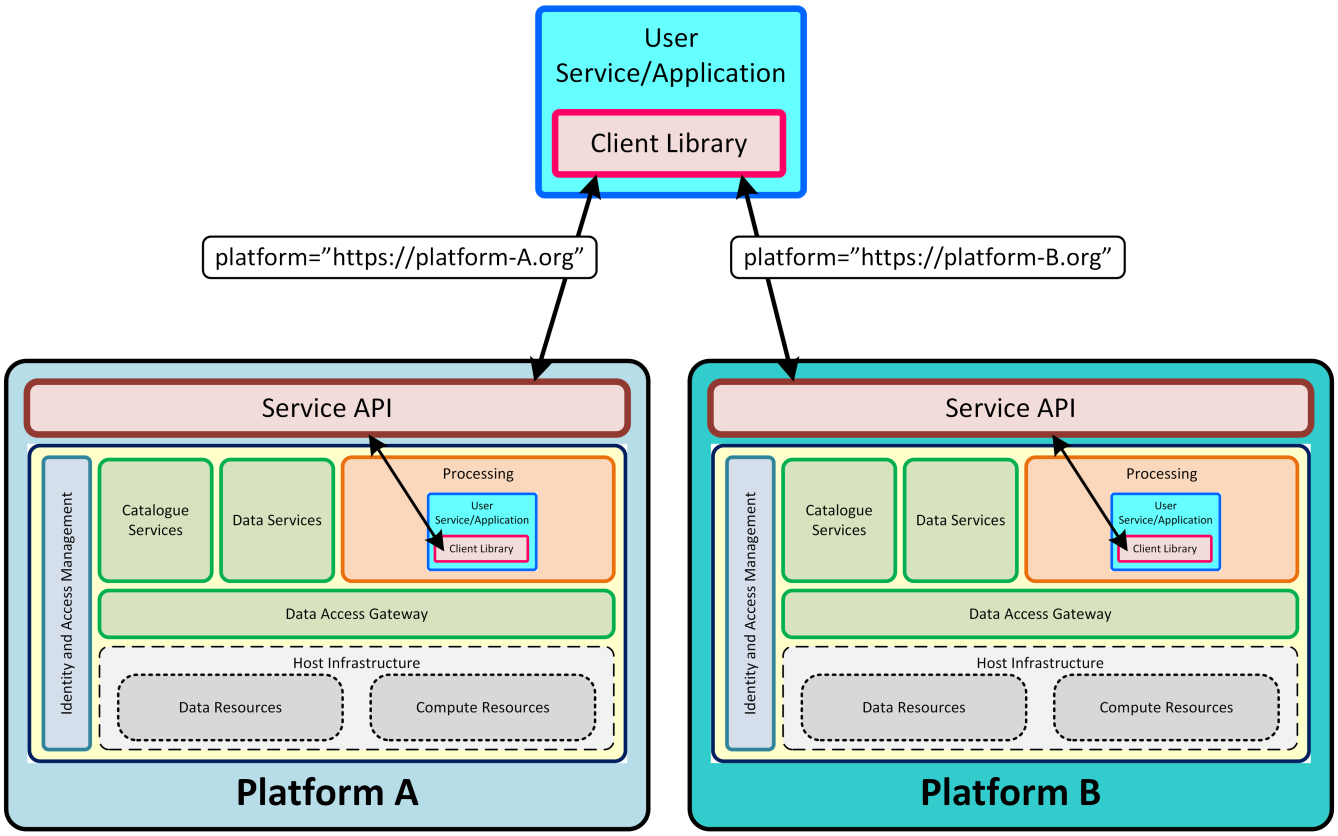


Figure 4. Client Portability

As illustrated in Figure 4, code implemented against the Client Library is not tied to a particular Exploitation Platform, but instead can be initialised and executed against any EP that supports the corresponding Service API. The User Service/Application shown in the figure can either be a process running external to the platform (e.g. on the users local platform), or running as a deployed process within the Processing Framework of the platform. **It should be noted that the use of the Client Library is not mandatory - instead the application can be developed against the Service API directly.**

7.1. Service API

The Service API presented by the Exploitation Platform is largely defined and met by the fundamental data/processing services it offers. There are some additional meta-services that support the clients in the discovery and usage of these core services. The Service API defined in this section seeks to define a standard set of interfaces against which the Client Library can be developed and that can be relied upon for platform-to-platform interoperability.

7.1.1. Platform Capabilities

Provides a single well-known 'bootstrap' URL through which the service capabilities and their endpoints can be discovered:

- Declare platform capabilities
- Discover service endpoints
- Core platform data/processing services
- Additional meta-services to support clients
- Used by Client Library to initialise its platform interfaces
- Used by EMS to match service/application dependencies with platform capabilities
- Used by other platforms to establish interoperability

/.well-known/eoepca-platform

Example Platform Capabilities

```
{
    "services": [
        { "type": "oidc", "role": "authentication", "path": "/connect" },
        { "type": "<tbd>", "role": "billing", "path": "/billing" },
        { "type": "opensearch", "role": "data_search", "path": "/search" },
        { "type": "csw", "role": "data_catalogue", "path": "/catalogue" },
        { "type": "opensearch", "role": "app_search", "path": "/applications" },
        { "type": "wms", "role": "map", "path": "/map" },
        { "type": "wmts", "role": "tile", "path": "/tile" },
        { "type": "wfs", "role": "feature", "path": "/feature" },
        { "type": "wcs", "role": "coverage", "path": "/coverage" },
        { "type": "wcps", "role": "datacube", "path": "/datacube" },
        { "type": "s3", "role": "object_store", "path": "/storage" },
        { "type": "wps-t", "role": "ems", "path": "/ems" },
        { "type": "wps-t", "role": "ades", "path": "/ades" },
        { "type": "<tbd>", "role": "workspace", "path": "/workspace" },
    ],
    "extended_capabilities": [
        { "type": "dask", "role": "cluster", "details": {} },
    ]
}
```

The endpoints referenced in eoepca-platform are described in the following sections.

7.1.2. authentication

The URL of the OpenID Connect Provider that implements the [mainLoginService] for user authentication.

/connect (example)

Implements the OpenID Connect protocol as described in [OIDC].

7.1.3. billing

The URL of the endpoint of the Billing service in the platform.

/billing (example)

The Billing service provides a cenrtralised point of contact within the platform responsible for tracking and billing for usage of resources and services. The approach to billing is currently not defined, but will be documented in section [mainBilling].

7.1.4. data_search

The URL of the OpenSearch interface to the [mainDataCatalogue].

/search (example)

Implements an OpenSearch interface in accordance with section [mainResourceCatalogue].

7.1.5. data_catalogue

The URL of the OGC CSW (Catalogue Services for the Web) interface to the [mainDataCatalogue].

/catalogue (example)

Implements an OGC CSW catalogue in accordance with standard **OGC Catalogue Services 3.0 Specification - HTTP Protocol Binding** as defined in [OGC-CSW].

7.1.6. app_search

The URL of the OpenSearch interface to the [mainAppCatalogue].

/applications (example)

Implements an OpenSearch interface in accordance with section [mainAppCatalogue].

7.1.7. map

The URL of the OGC WMS (Web Map Service) interface that supports the data maintained in the [mainDataCatalogue].

/**map** (example)

Implements an OGC WMS service in accordance with standard **OGC Web Map Server Implementation Specification** as defined in [OGC-WMS].

7.1.8. tile

The URL of the OGC WMTS (Web Map Tile Service) interface that supports the data maintained in the [mainDataCatalogue].

/**tile** (example)

Implements an OGC WMTS service in accordance with standard **OGC Web Map Tile Service Implementation Standard** as defined in [OGC-WMTS].

7.1.9. feature

The URL of the OGC WFS (Web Feature Service) interface that provides a *Feature-oriented* access to the underlying data holding of the platform.

/**feature** (example)

Implements an OGC WFS service in accordance with standard **OGC Web Feature Service 2.0 Interface Standard – With Corrigendum** as defined in [OGC-WFS].

7.1.10. coverage

The URL of the OGC WCS (Web Coverage Service) interface that provides a *Coverage-oriented* access to the underlying data holding of the platform.

/**coverage** (example)

Implements an OGC WCS service in accordance with standard **OGC Web Coverage Service (WCS) 2.1 Interface Standard - Core** as defined in [OGC-WCS].

7.1.11. datacube

The URL of the OGC WCPS (Web Coverage Processing Service) interface that provides a queryable 'data cube' interface to multi-dimensional coverage data.

/**datacube** (example)

Implements an OGC WCPS service in accordance with standard **Web Coverage Processing Service (WCPS) Language Interface Standard** as defined in [OGC-WCPS].

7.1.12. object_store

The URL of the *Amazon S3* interface that provides object storage access to the underlying data holding of the platform.

/**object_store** (example)

Implements an Amazon S3 service in accordance with **Amazon Simple Storage Service REST API** as defined in [AWS-S3].

Amazon S3 Compatibility Subset

For the purposes of this interface, a subset of the full Amazon S3 REST API will be defined as mandatory. The motivation is to define a consistent interface that is supported by third-party object storage implementations that provide an S3-compatible API, such as:

- Ceph (https://ceph.com/ceph-storage/)
  - CEPH OBJECT GATEWAY S3 API (http://docs.ceph.com/docs/mimic/radosgw/s3/)
- OpenStack Swift (https://docs.openstack.org/swift)
  - S3/Swift REST API Comparison Matrix (https://docs.openstack.org/swift/latest/s3_compat.html)

7.1.13. ems

The URL of the [mainProcEMS] service.

/ems (example)

Implements an **EMS service** as described in section [mainProcEMS].

7.1.14. ades

The URL of the [mainProcADES] service.

/ades (example)

Implements an **ADES service** as described in section [mainProcADES].

7.1.15. workspace

The URL of the 'user workspace' service as described in section [mainWorkspace].

/workspace (example)

The interface for the workspace is currently TBD.

7.2. Client Library

The Service API and its associated Client Library together present a standard platform interface against which analysis and exploitation activities may be developed, and through which platform services can be federated.

The Client Library is a programmatic representation of the Service API which acts as an abstraction of the Exploitation Platform and so facilitates the development of portable client implementations.

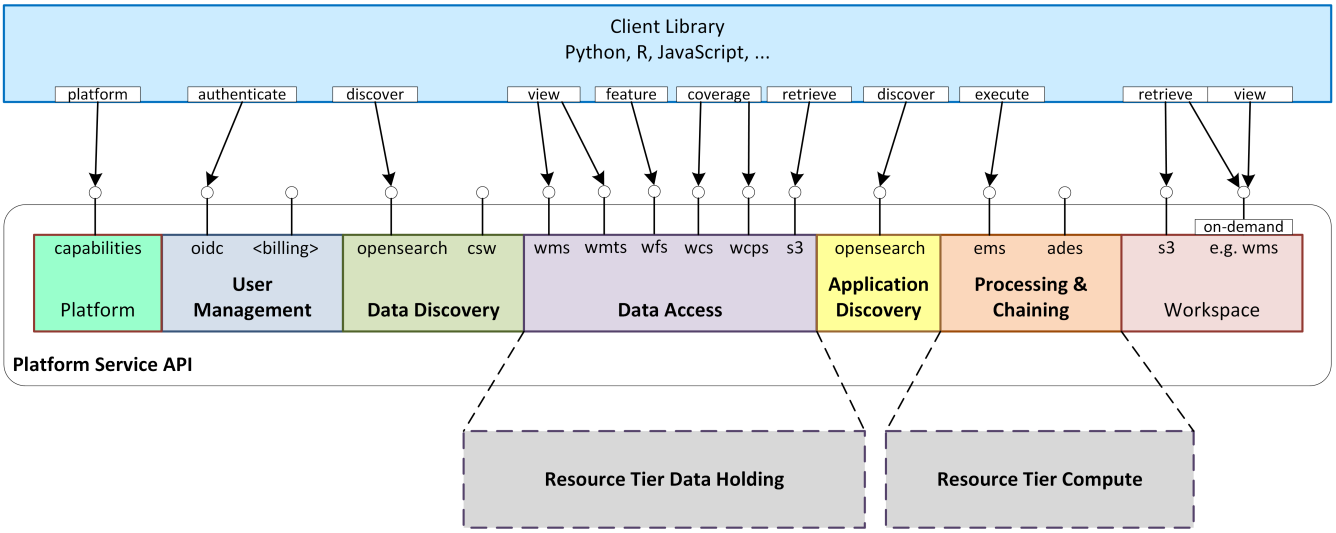


Figure 5. Client Library Service Interfaces

As illustrated in Figure 5, the Client Library provides bindings for common languages (Python, R, Javascript) that utilise the network (HTTP) interfaces of the Service API, covering all aspects of the Exploitation Platform functionality (authentication, data/processing discovery, processing etc.).

7.2.1. Client Library Concept Illustration

The design of the Client Library is not yet established. To illustrate its concept we present an example based upon the workflow scenario that was used to demonstrate the EMS/ADES best practice in OGC Testbed-14 ([TB14-ADES]). The scenario is shown in Figure 6.

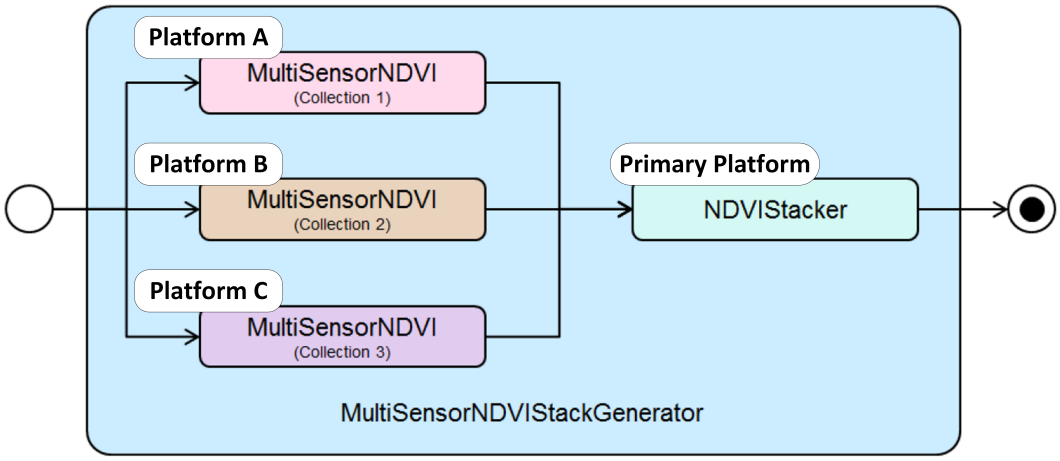


Figure 6. OGC Testbed-14 Workflow Scenario (NDVI Stacker)

Four platforms are involved in the scenario:

Platforms A/B/C

These platforms perform the local execution (ADES) of the MultiSensorNDVI processor on a collection in their local data holding. These executions are managed under the orchestration (EMS) of the Primary Platform.

Primary Platform

The Primary Platform is in charge of the workflow. Thus, the orchestration (EMS) is conducted from this platform, which interfaces to the 'subordinate' platforms (A/B/C) for the execution of

the steps 'close-to-the-data', and then completes the workflow by execution (ADES) of the NDVIStacker processor to produce the final result.

In response to this scenario we might envisage a client application implemented against the Client Library as follows...

Example python program using Client Library

```
import eoepca

# Connect to primary platform - e.g. when running on own desktop
primaryPlatform = eoepca.platform("http://primary.platform.eo") ①
primaryPlatform = primaryPlatform.authenticate("bob@home.org", "<<MY-API-KEY>>") ②
# Or, use the 'local' hosting platform - e.g. when running in 'cloud' platform
primaryPlatform = eoepca.platform().authenticate("bob@home.org", "<<MY-API-KEY>>") ②
③

# Init supporting platforms
platA = eoepca.platform("http://platform-a.eo") ① ④
platB = eoepca.platform("http://platform-b.eo") ① ④
platC = eoepca.platform("http://platform-c.eo") ① ④

# Specify extent
extent = { "bbox": [ -0.489, 51.28, 0.236, 51.686 ], "time": [ "2018-01-01", "2018-12-
31" ] }

# Specify processes
coverage1 = platA.collection("PLAT_A_DATA").coverage(extent) ⑤
proc1 = coverage1.process("MultiSensorNDVI") ⑥
proc2 = platB.collection("PLAT_B_DATA").coverage(extent).process("MultiSensorNDVI") ⑤
⑥
proc3 = platC.collection("PLAT_C_DATA").coverage(extent).process("MultiSensorNDVI") ⑤
⑥

# Specify workflow
workflow = primaryPlatform.parallel([proc1, proc2, proc3]).process("NDVIStacker") ⑦

# Get result - initiates 'lazy' execution
result = workflow.retrieve(format="geotiff", options={}) ⑧
print(result)
```

① When each platform object is initialised, its endpoint `${platform-url}/.well-known/eoepca-platform` is interrogated, to understand its capabilities and learn its service endpoints.

② The user must authenticate to the primary platform (that 'executes' the workflow), using their API key.

③ In the case where the client is **running on the primary platform** then the platform URL is not required in the initialisation (implying 'local' platform). Examples of this case include: code running in a hosted Jupyter notebook, or a deployed processing service that chooses to use the Client Library.

④ Subordinate platforms are initialised (for capabilities) without authentication, on the basis that the primary platform authentication can be carried through the workflow 'call-stack' through delegated/federated IAM solution - ref. section [IAM].

⑤ At each platform the collection is selected through its unique collection identifier, and a data coverage subset is specified through a definition of the required spatial/temporal extent.

⑥ At each platform the processing task to be exectued against the selected coverage is specified.

⑦ The workflow is defined by requesting the parallel execution of the `MultiSensorNDVI` processor on each of the three platforms, with these results providing input to the `NDVIStacker` process executed on the primary platform.

⑧ The call to `retrieve` the outcome of the workflow initiates its **'lazy' execution**. Prior to this point the Client Library has been operating on **'proxy'** objects that record the **specification of the workflow** requested by the code. At this point the Client Library converts the workflow specification into a CWL definition suitable for deployment and execution at the EMS of the primary platform. In response the EMS will interface with the ADES of each subordinate platform to ensure the `MultiSensorNDVI` is deployed and executed against the requested coverage. Subsequently the EMS interfaces with the ADES of the local platform to ensure the `NDVIStacker` is deployed and executed against the outputs of the three `MultiSensorNDVI` process executions.

Chapter 8. Web Portal Interfaces

<< End of Document >>