Earth Science Collaborative Open Development Environment (EARTH-CODE)

Technical Proposal

Ref: KPT92154

Date: 04/09/2023

For: ESA ESRIN

Ref: AO/1-11828/23/I-NS

This Page is Intentionally Blank

SIGNATURES AND COPYRIGHT

Title : **EOEPCA+**

Volume : **Use Case Definition Document**

Issued : **04 September 2023**

Client : **ESA ESRIN**

Authored : GSmith

Garin Smith - Senior Software Architect

Approved : Eddie Glover

Alasdair Kyle - Senior Project Manager

Approved :

Simon Farman - Head of Ground Software Systems

Address : Telespazio UK Ltd
350 Capability Green, Luton, Bedfordshire, LU1 3LU, United Kingdom
Tel.: +44 (0)1582 399 000
Fax: +44 (0)1582 728 686

Copyright :

This Page is Intentionally Blank

TABLE OF CONTENTS

This Page is Intentionally Blank

1. INTRODUCTION

Telespazio UK (**TPZ-UK**) is pleased to present this use case specification to the European Space Agency (**ESA**) Invitation to Tender (**ITT**) **ESA-EOP-SD-SOW-0397 – EO PLATFORM INTEROPERABLE BUILDING BLOCK EVOLUTION FRAMEWORK - COORDINATION**. We are pleased to confirm that this use case specification is fully compliant to all technical requirements included in the ESA SoW [**AD-2**].

This task will identify platform community requirements and gaps in current EOEPCA capability. It will capture common requirements that might help increase interoperability amongst platforms.

The following related subtasks are considered in this document:

- R01 - Collect Use Cases
  Collect the use cases from the utilisation domains.
- R02 – Derive Requirements
  Derive requirements from the above Domains.
- R03 - Use Case Validation and Epic Prioritisation
  Validate requirements and prioritise epics with stakeholders
- R04 - Additional Utilisation Domains
  Consider up to 2 new utilisation domains per year and repeat the above process agreeing the priority of epics with the ESA TO.

This document can be thought of as a living document. It can and will be updated throughout the project to capture new requirements and modify existing requirements as needed. More work is expected on R03 and R04 as the project progresses and so they will not be considered in so much detail during early iterations of this document.

1.1 Applicable Documents

The following is a list of applicable documents with a direct bearing on the content of this proposal. Where referenced in the text, these are identified as [AD-n], where 'n' is the number in the list below:

AD-1. Reference Architecture Document

1.2 Reference Documents

The following is a list of reference documents with a direct bearing on the content of this proposal. Where referenced in the text, these are identified as [RD-n], where 'n' is the number in the list below:

RD-1. EO Platform Interoperable Building Block Evolution Framework Statement of Work (ESA-EOP-SD-SOW-0397)

2. R01 - COLLECT USE CASES

This task is responsible for collecting high level use cases from the utilisation domains.

At this stage the following utilisation domains have been identified:

- APEx
- EarthCODE
- Information Factories

However, more utilisation domains can be added later as required.

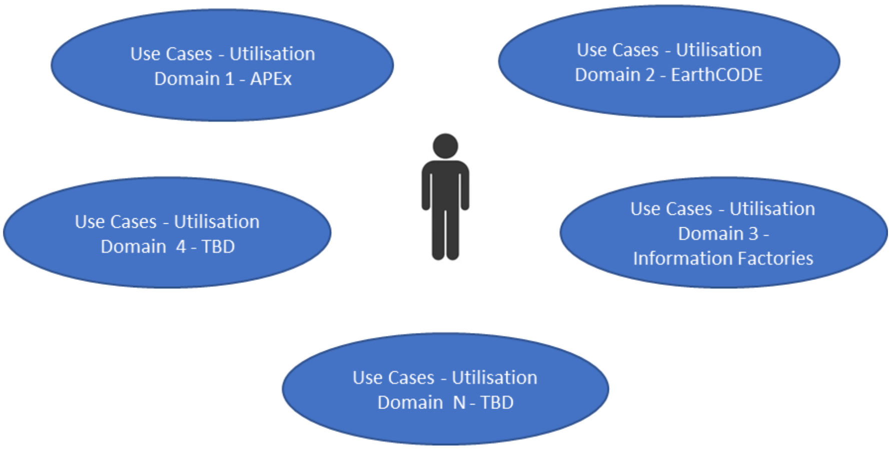
Use Cases - Utilisation Domain 1 - APEx
Use Cases - Utilisation Domain 2 - EarthCODE
Use Cases - Utilisation Domain 4 - TBD
Use Cases - Utilisation Domain 3 - Information Factories
Use Cases - Utilisation Domain N - TBD

Figure 1 : Utilisation Domains

1.1. APEx

The Application Propagation Environment (APEx) aims to increase the uptake and reuse of RD application project results, especially algorithms and workflows, and ensure their long-term availability to the EO community.

It is expected that APEx might achieve this by considering the following architectural components: Algorithm Hosting, Algorithm Enhancement and Algorithm Upscaling.

The Algorithm Hosting Service will provide a registry of hosted services, in which hosted applications are registered and published so that they are discoverable and available for reuse by other projects. This is supported in EOEPCA+ by the Resource Register, Resource Health Check and Resource Discovery use cases.

The Algorithm Enhancement Service is a human-led APEx service that benefits from tooling to automate as much as possible the generalised approach to enhance processing workflow algorithms and bring them to production. This is supported in EOEPCA+ by the Application Testing and Operational Performance use cases.

The Algorithm Upscaling Service supports the execution of processing algorithms across extended large-scale spatial and temporal extents. It is assumed that the upscaled execution is performed within the existing application hosting environment for the processing workflow. This is supported in EOEPCA+ by the Resource Discovery, Federated Orchestrator and Application Processing use cases.

It is expected that APEx will be interested in the majority of EOEPCA+ building blocks.

1.2. EarthCODE

EarthCODE will provide a platform environment to support gold-standard best-practice FAIR open scientific development and research – with access to data, compute resources, services and tooling for development and community. EarthCODE comprises a portal as an entry point to that provides development tooling facilitating FAIR Open Science through tooling for workflow and data management, supported by guides and tutorials to establish and communicate best practice. EarthCODE integrates access to external platform services for workflow execution, data, storage, and other platform resources, whose capabilities are made available through the EarthCODE platform offering.

EarthCODE has defined the following objectives which could benefit from EOEPCA+ building blocks. It is expected that EarthCODE will be potentially interested in the majority of EOEPCA+ building blocks.

EarthCODE Objectves

1) Develop a collaborative, cloud based, community-centric scientific development environment (EarthCODE) for relevant ESA science activities and projects, especially those of the ESA Science for Society programme and its Science Clusters (e.g., Polar, Carbon, Atmosphere, Ocean, etc.);

This is supported primarily by the Application Development use case along with the drive for open standards in EOEPCA+.

2) Offer the EarthCODE open science framework for Earth Observation scientists conducting research in the ESRIN ScienceHub and provide open access to the EarthCODE content for the scientific community;

This is supported by EOEPCA+ reference architecture that demonstrates how the EOEPCA+ building blocks can deliver an EarthCODE solution.

3) Develop the EarthCODE as an open architecture, using open-source building blocks and integrating commercial platform services (for EO data, processing, storage, etc.), offering an effective, efficient and attractive environment to conduct Earth System Science collaboratively, with convenient means to: discover, access, and work with EO and other geospatial data, manage the full lifecycle of data, code, documentation and other research artifacts of Science Cluster activities, manage end-to-end scientific experiments and interact with the community;

This is supported by the EOEPCA+ open-source approach and the resource handling and Application Processing use cases.

4) Design and implement a long-term FAIR Open Science model for scientific research, across the end-to-end workflows developed in the frame of the ESA Science Clusters and the Science Hub activities;

FAIR (Find, Access, Interoperate and Reuse) principles are supported by a number of EOEPCA+ Use Cases. Find is supported by Resource Discovery and Federated Resource Discovery. Access is supported by Data Access and Datacube Access. The Identity Access and Management and Accounting use cases also make Access easier to implement. Interoperability is supported by the use of EOEPCA+ open standards as well as the provision of building blocks that are demonstrated to work collaboratively. Reuse is supported in many ways including a Resource Register use case that might be capable of describing reuse conditions, data lineage, processing requirements, formal data definitions and formal processing/workflow definitions. Reuse is also supported by Application Processing, Scalability and Federated Orchestrator use cases.

5) Contribute to the development of scientific experiments that are end-to-end reproducible on platforms;

This is supported by all the considerations in the previous objective. In addition the Application Quality and Operational Performance use cases will help support the maintainability of applications and platforms, so that experiments do not break because of application or operational issues.

6) Put in place a mechanism to ensure long-term availability and update of geophysical products developed by ESA Science Cluster activities, using industrially provided cloud services;

This is partly an operational platform concern and may be supported by the Operational Performance use case. The EOEPCA+ Resource and Access use cases may also help support this endeavour.

7) Contribute to the development and growth of an Earth Observation Science community conducting reproducible and FAIR Open Science

EOEPCA+ helps to enable this by supporting all the objectives above. The EOEPCA+ open-source approach also encourages the community approach likely to be adopted by EarthCODE.

1.3. Information Factories

Information Factories will enable the setup an information factory operated by a legal entity (the “information factory provider”) for the development and production of indicators derived from EO space data. It is expected that applications processing capability will use EO space data eventually combined with local data.

The information factory shall provide a service to a specific community to host their applications, providing for data/infrastructure/execution environment resources and taking care of operational and billing processes. Applications shall be supported to easily make use of readily available space data in analysis-ready format together with regional non-space data relevant for the topic. The development/execution environment provided shall be able to support applications based on traditional workflows/processing algorithms or based on information extraction via AI techniques.

It is expected that Information Factories could potentially be interested in the majority of EOEPCA+ building blocks.

It is likely that EOEPCA+ could offer most support with regard to data, application hosting and onboarding requirements. An interesting consideration will be the need for sovereignty with respect to providers and the handling of interoperability.

1.4. Use Case Traceability

Utilisation domain requirements will be broken down into one or more use cases and these will in turn be broken down into one or more requirements/user stories. Note that use case definitions can be applicable across multiple utilisation domains . Use Cases can also be thought of as capabilities if helpful. Please see Figure 2 : Examples Use Cases and Requirements

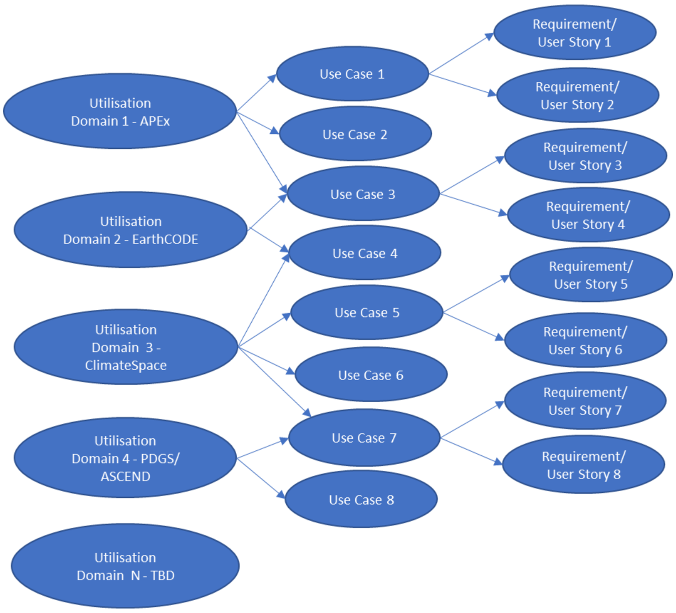
Utilisation Domain 1 - APEx
Utilisation Domain 2 - EarthCODE
Utilisation Domain 3 - ClimateSpace
Utilisation Domain 4 - PDGS/ ASCEND
Utilisation Domain N - TBD
Use Case 1
Use Case 2
Use Case 3
Use Case 4
Use Case 5
Use Case 6
Use Case 7
Use Case 8
Requirement/ User Story 1
Requirement/ User Story 2
Requirement/ User Story 3
Requirement/ User Story 4
Requirement/ User Story 5
Requirement/ User Story 6
Requirement/ User Story 7
Requirement/ User Story 8

Figure 2 : Examples Use Cases and Requirements

Requirements/user stories can later be used to help provide building block outline specifications against which building blocks can be implemented. See Figure 3 : Building Block Outline Specification Defined with User Stories/Requirements

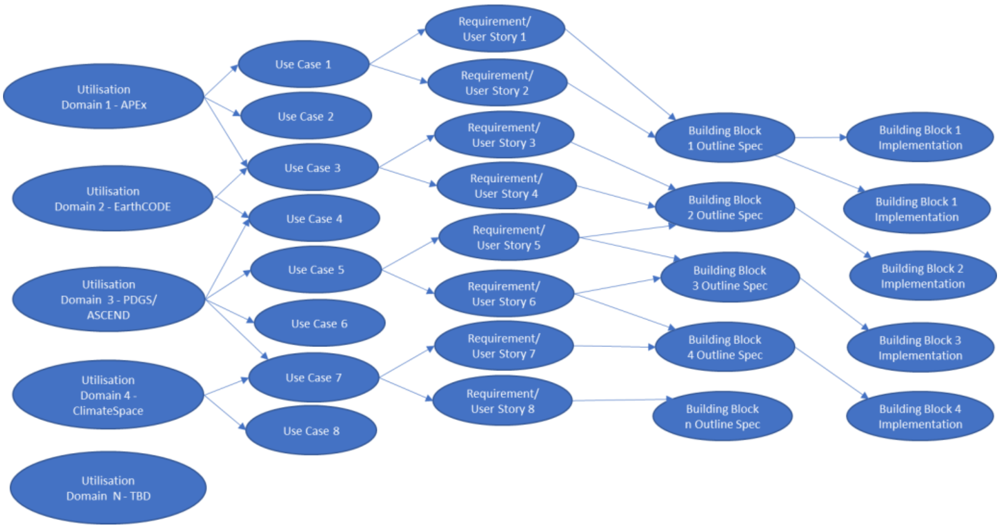
Utilisation Domain 1 - APEx
Utilisation Domain 2 - EarthCODE
Utilisation Domain 3 - PDGS/ ASCEND
Utilisation Domain 4 - ClimateSpace
Utilisation Domain N - TBD
Use Case 1
Use Case 2
Use Case 3
Use Case 4
Use Case 5
Use Case 6
Use Case 7
Use Case 8
Requirement/ User Story 1
Requirement/ User Story 2
Requirement/ User Story 3
Requirement/ User Story 4
Requirement/ User Story 5
Requirement/ User Story 6
Requirement/ User Story 7
Requirement/ User Story 8
Building Block 1 Outline Spec
Building Block 2 Outline Spec
Building Block 3 Outline Spec
Building Block 4 Outline Spec
Building Block n Outline Spec
Building Block 1 Implementation
Building Block 1 Implementation
Building Block 2 Implementation
Building Block 3 Implementation
Building Block 4 Implementation

Figure 3 : Building Block Outline Specification Defined with User Stories/Requirements

1.5. Use Cases

So far, the following use cases have been identified. Further context for these use cases is provided in the next section.

- Resource Discovery
- Federated Resource Discovery
- Resource Register
- Data Access
- Datacube Access
- User Resource Management
- Application Processing
- Application Development
- Train AI Model
- Test AI Model
- Exploit AI Model
- Federated Orchestrator
- Identity Access and Management
- Operational Performance
- Resource Health Check
- Application Development
- Application Testing
- Accounting
- Natural Human Interface

3. R02 – DERIVE REQUIREMENTS

This task is responsible for deriving requirements or user stories from the use cases defined in the previous section. Use cases can consist of a number of user stories (requirements) that can be added to a backlog for consideration in the planning process. See Reference Architecture Document . We expect a Scrum like process to be applied to manage this backlog and this will allow:

- Epics to be defined
- Agile implementation
- Epics and requirements to be maintained while grooming the backlog.

The Use Case Definition Document will contain:

- User Stories (requirements)
- Use Cases
- Utilisation Domains
- Epics (note that this can be added later if required when grouping stories towards the top of the backlog).
- Tech Standards
- Building Block

As explained previously, more detail regarding Epics, Tech Standards and Building Blocks will be provided in later versions of this document.

The backlog will use the following format.

Table 1 : Backlog Format

Backlog					
ID	**User Story (requirement)**	**Use Case**	**Tech Standards**	**Epic**	**Building Block**

It is also perhaps useful to consider the dependency relationship between some cases (key capabilities). This is important because:

- It will help guide the architectural discussion on how key capabilities collaborate and depend upon each other.
- It will help inform the end-to-end V&V and the end-to-end testing suite that is needed later.
- It will help consider key design patterns that may also help inform later stages of the project.
- Considering use case dependencies also informs the planning of epics and user stories. This should help enable the support of end to end demonstrations of building blocks. See Figure 4 : Use Case Dependencies

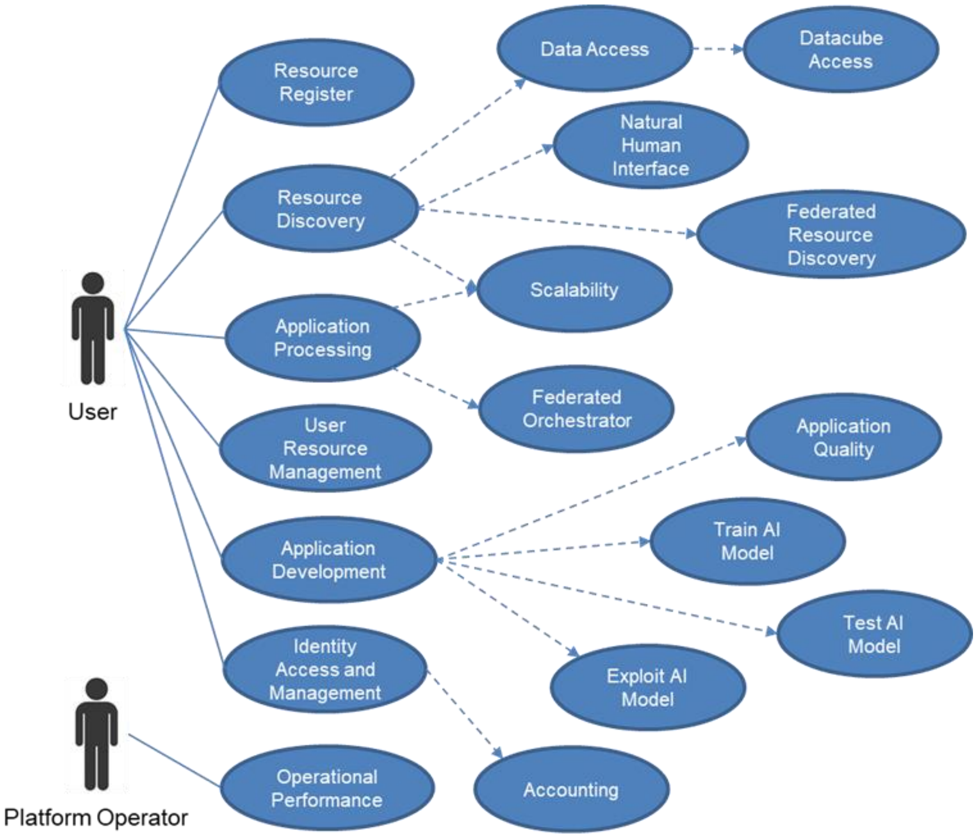

Figure 4 : Use Case Dependencies

It is important to establish the roles supported by use cases and this is also addressed above. This will help inform specific user journeys and it will also help inform Identity Access and Management.

3.1 Use Case Analysis

The following requirements are presented for each of the use cases above.

3.1.1 Resource Discovery

This use case is primarily concerned with supporting Find capability as one of the FAIR principles. However it should be stressed that resource discovery is concerned with more than just data and includes the following types of resources : data (Datasets, Datacube and Virtual Datacube), Replicable Workflows, Reproducible Job Details, Jupyter Notebooks, Executable Services, Platform Services, Web Applications, Machine Learning Training Data, Machine Learning Models, Collections, Source Repositories and Documentation.

This use case is related to the Resource Discovery Building Block.

One is the challenges will be to help support interoperability of resources, so that for instance a Replicable Workflow can also discover data that it depends upon or perhaps a Machine Learning Model can also discover Machine Learning Training Data that it depends upon.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC1	As a user, I can discover a dataset, so that I know where assets are located	Resource Discovery			
UC2	As a user, I can discover a datacube, so that I know where it is located	Resource Discovery			

UC3	As a user, I can discover an Application Package CWL, so that I can use it for processing in an OGC API Processes container	Resource Discovery	OGC API Processes		
UC4	As a user, I can discover a process, so that I can execute it using a defined endpoint (existing user defined process/service)	Resource Discovery			
UC5	As a user, I can discover a software repository, so that I can use software in my own environment	Resource Discovery			
UC6	As a user, I can discover a external catalogue service, so that I can use it to discover external resources	Resource Discovery			
UC7	As a user, I can discover details of a previous job/process execution, so that it helps reproducible open science	Resource Discovery			
UC8	As a user, I can discover details of a Web Application, so that I can use human interfaces to assist me.	Resource Discovery			
UC9	As a user, I can discover Machine Learning Training Data, so I know where is located.	Resource Discovery			
UC10	As a user, I can discover Machine Learning Model, so that I can use it for processing in a model exploitation server.	Resource Discovery			
UC11	As a user, I can discover Collections using an open standards search API, so I can discover the things I need.	Resource Discovery	STAC		

UC12	As a user, I can discover Collections using an open standards browser API, so I can discover the things I need.	Resource Discovery	STAC		

3.1.2 Federated Resource Discovery

This use case is primarily concerned with supporting Find capability as one of the FAIR principles within a federated context. However it should be stressed that resource discovery is concerned with more than just data and includes many types of resource.

One challenge here will be to manage federation, perhaps by implementing common resource definition standards or by implementing adaptors to implement a common resource type.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC13	As a user, I want to make requests to a catalogue broker using the standard Resource Discovery interface, so I can treat it like the Resource Discovery	Federated Resource Discovery	STAC		

3.1.3 Resource Register

This use case is primarily concerned with supporting Reuse capability as one of the FAIR principles. However it should be stressed that resource register is concerned with more than just data and includes many types of resource.

This use case is related to the Resource Registration Building Block.

This is perhaps one of the most important building blocks because it is responsible for defining resources in a way that enables them to be registered, discovered, accessed and used in collaboration with other resources across multiple platforms using multiple types of processing solutions. This is perhaps one of the biggest challenges.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC14	As a user, I want to ingest metadata using STAC builder, so that I can then use it with Resource Discovery	Resource Register	STAC		
UC15	As a user, I want to ingest data into data access services to make it available for retrieval and for visualisation	Resource Register			

3.1.4 Data Access

This use case is primarily concerned with supporting access capability as one of the FAIR principles. The data access use case can refer to both human and machine interfaces.

This use case is related to the Data Access Building Block.

Challenges for this use case may including dealing with dependencies on authentication and authorisation issues or concerns relating to the federation of data. Dealing with archived data may also be a consideration. Another challenge will be to ensure that workflows and processes running in new platforms are provided with the data they need to operate correctly.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC16	As a user, I want data access using OGC WCS Coverages, so I can access my data	Data Access	OGC API Coverages		
UC17	As a user, I want data access using OGC WFS, so I can access my data	Data Access	OGC WFS		
UC18	As a user, I want data access using OGC API Features, so I can access my data	Data Access	OGC API Features		
UC19	As a user, I want data access using OGC WMS, so I can visualise my data	Data Access	OGC WMS		
UC20	As a user, I want data access using OGC API Maps , so I can visualise my maps data	Data Access	OGC API Maps		
UC21	As a user, I want data access using OGC API Coverages, so I can access my data	Data Access	OGC API Coverages		
UC22	As a user, I want data access using OGC WMTS, so I can access my Tiles data	Data Access	OGC WMTS		
UC23	As a user, I want data access using OGC API Tiles, so I can access my Tiles data	Data Access	OGC API Tiles		

UC24	As a user, I want data access using OGC API Features (Part 4) Transactions, so I can access my data and perform CRUD OPS	Data Access	OGC API Features (Part 4)		

3.1.5 Datacube Access

This use case is primarily concerned with supporting Access capability as one of the FAIR principles. The datacube access use case can refer to both human and machine interfaces.

This use case is related to the Datacube Access Building Block.

Challenges for this use case may also including dealing with dependencies of authentication and authorisation issues.

ID	User Story (requirement)	Use Case	Epic	Tech Standards	Building Block
UC25	As a user, I want access to a Datacube backend using OGC Geodatacube API SWG, so I can use a Datacube	Datacube Access			
UC26	As a user, I want access to a Datacube backend using OGC Geodatacube API SWG using Python, so I can develop in Python	Datacube Access			
UC27	As a user, I want access to a Datacube backend using OGC Geodatacube API SWG using R, so I can develop n R	Datacube Access			

UC28	As a user, I want access to a Datacube backend using OGC Geodatacube API SWG using Julia, so I can develop in Julia	Datacube Access			
UC29	As a user, I want access to a Datacube backend using OGC Geodatacube API SWG using Javascript, so I can develop in Javascript	Datacube Access			
UC30	As a user, I want a human interface to access the Datacube Access backend, so I access a Datacube	Datacube Access			

3.1.6 User Resource Management

This use case is concerned with gaining access to services that can help manage resources. These services will include the capability to register, discover and access resources. Access to resources can be at an individual, group, project or global level.

This use case is related to the Workspace Building Block.

Challenges for this use case may also including dealing with dependencies of authentication and authorisation issues. This may be more challenging if resources are federated across platforms.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block

UC31	As a user, I want a Resource Discovery instance, so that I can find and access the things I need.	User Resource Management			
UC32	As a user, I want a Resource Registration instance, so that I can ingest things I need to find later.	User Resource Management			
UC33	As a user, I want a Data Access instance, so that I can access the things I have found.	User Resource Management			
UC34	As a user, I want a Object storage (S3) instance, so that I can I can store and retrieve the data I want.	User Resource Management			
UC35	As a user, I want a Container registry instance, so that I can register and discover new process containers.	User Resource Management			
UC36	As a user, I want a User profile management instance, so that I can manage my personal details.	User Resource Management			
UC37	As a user, I want group access to User Resource Management capabilities so that I can share them with my team.	User Resource Management			

3.1.7 Application Processing

This use case is concerned with the provision of flexible application process capability. This includes capability such as OGC API Process (Part 1 and 2), OpenEO API, High Performance Computing, Serverless and GPU based solutions.

This use case is related to the Processing Building Blocks.

Some of the challenges here will include managing the platform dependencies on authentication and authorisation, as well as managing access to data and also determining how best to provide users access to processing results. A user may be encouraged to visit a platform to get results, or it may be that results are ingested into a catalogue on another platform or indeed a central platform hub.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC38	As a user, I want a Processing Engine that implements OGC API Process (Part 1 and 2), so that I can deploy an Application Package and later use it to execute my requests.	Application Processing	OGC Best Practice for Application Packages		
UC39	As a user, I want a Processing Engine that implements OGC API Process (Part 1 and 2), so that I can execute and get the results of an Application Package	Application Processing	OpenEO process graph		
UC40	As a user, I want a Processing Engine that implements OpenEO API, so that I can deploy a User Defined Process and later use it to execute my requests	Application Processing	OGC API Process (Part 1 and 2)		
UC41	As a user, I want a Processing Engine that implements OpenEO API, so that I can execute and get the results from an Application	Application Processing	OpenEO API		
UC42	As a user, I want a Processing Engine provided in a container over K8S (Docker) with GPU support, so that I can execute and get the results of my process	Application Processing			

UC43	As a user, I want a Processing Engine provided in a container over HPC (Chaliecloud, Singularity), so that I can execute and get the results of my process	Application Processing			
UC44	As a user, I want a Processing Engine provided in a Serverless environment (Knative, AWS Lambda, etc), so that I can execute and get the results of my process	Application Processing			

3.1.8 Application Development

This use case is concerned with supporting developers with tools and environments needed to provide them with development capabilities.

This use case is related to the Application Hub Building Block.

Challenges will perhaps revolve around the complexities of federated solutions and the size of the community being served. As the community increases in size it is perhaps likely that scientists and developers will discover more resources of interest. This means it is possible that users will need access to more development tooling to exploit resources. It is likely that users with different experience and ability will require different levels of support to use the application development toolset available.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC45	As a user, I want an interactive web environment that supports interactive analysis via Jupyter Notebooks	Application Development			

UC46	As a user, I want an interactive web environment that provides tooling that supports development of workflows. Tooling should include IDE, validation tools and tools that facilitate packaging as a reproducible workflow	Application Development			
UC47	As a user, I want an interactive web environment that includes a library of 'applications' that can be launched as interactive applications. I want the library to be extensible so that I can add user-defined additional applications.	Application Development			
UC48	As a user, I want an interactive web environment that allows me to add web applications that run as persistent web services - e.g. Dashboards, so that I can present actionable information - e.g. using streamlit, voila, etc. The web services should be publishable via dedicated URLs for consumption via the public web.	Application Development			
UC49	As a user, I want an interactive web environment that allows me to run hosted native applications (e.g. SNAP, QGIS) delivered 'remotely' through the interactive web application interface (i.e. through web-RDP)	Application Development			
UC50	As a user, I want the applications in my interactive web environment to maintain persistence of my work between sessions	Application Development			

3.1.9 Train AI Model

This use case is related to platform support to allow users to train AI models.

This use case is related to the MLOps Building Block Building Block.

There are many challenges. One of the challenges here will be the provision of a solution that allows developers to use their preferred AI machine learning framework. Provision of access to training datasets and suitable processing capability may also be a challenge. Sharing models and training datasets with the community may also raise a number of obstacles.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC51	As a user, I want to initiate an AI Model training run for the most popular machine learning frameworks, so that I can evaluate the result	Train AI Model	ONNX		

3.1.10 Test AI Model

This use case is related to platform support to allow users to test and review trained AI models.

Challenges here may relate to the complexity of models developed. Some AI solutions require a series of steps to deal with pre and post processing and model synchronisation.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC52	As a user, I want to view the metrics for my AI Model training run, so that I can assess model performance	Test AI Model	ONNX		
UC53	As a user, I want to view the history of my training runs, so that I can view the associated parameterisation and model metrics	Test AI Model	ONNX		

3.1.11 Exploit AI Model

This use case is related to platform support to allow users to exploit AI models. Exploitation may involve sharing an AI model, sharing AI training data or providing an AI service or application.

As mentioned before challenges here may relate to the complexity of models developed. Some form of orchestration may be needed for more complex AI solutions. Exploiting an AI model may require the provision of user friendly human interfaces. AI model services can also be hosted in a number of ways - some users may prefer to use a specialised AI model server, but other may prefer a machine service that hides all of the AI complexity.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block

UC54	As a user, I want to publish an AI model version using Resource Registration, so that it can be discovered locally in my workspace with Resource Discovery	Exploit AI Model			
UC55	As a user, I want to publish an AI model version using Resource Registration, so that it can be discovered globally with Resource Discovery	Exploit AI Model	STAC extensions		

3.1.12 Federated Orchestrator

This use case will allow users to orchestrate the of processing workflows running on different platforms and in different workflow containers.

This use case is related to the Federated Orchestrator Building Block.

Challenges relate to the complexity of handling different platforms, different workflow containers, different applications all of which may use different protocols. Managing the federation of different protocols is important, perhaps by implementing common standards or by implementing adaptors to implement a common protocol type.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block

UC56	As a user, I want to execute processing workflows that use a mixture of workflow engine solutions (such as openEO, API Processes, Pangeo/xarray)	Federated Orchestrator			
UC57	As a user, I want to my workflow (openEO process graph) to be orchestrated for execution on either the local or another (remote) platform according to the platform business logic	Federated Orchestrator	OpenEO process graph		
UC58	As a user, I want to my workflow (OGC Application Package) to be orchestrated for execution on either the local or another (remote) platform according to the platform business logic	Federated Orchestrator	OGC Best Practice for Application Packages		
UC59	As a user, I want to my workflow (Pangeo/xarray) to be orchestrated for execution on either the local or another (remote) platform according to the platform business logic	Federated Orchestrator			

3.1.13 Identity Access and Management

This use case is concerned with user authentication and authorisation across platforms and managing the user attributes associated with a user.

This use case is related to the Identity and Access Management Building Block.

There are many challenges associated with this us case. Agreeing a common IDP (Identity Provider) across all platforms may be helpful. Finding a mechanism to agree how user attributes are shared across platforms will also be important. This may be seen as a key dependency for the implementation of a successful accounting use case.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC60	As a user, I want to authenticate with each platform using the same identity, so that I don't have to maintain multiple identities across many platforms	Identity Access and Management			
UC61	As an administrator, I want to uniquely identify each user and authorise access to the platform resources using policy rules that rely upon the user identity and associated attributes	Identity Access and Management			
UC62	As a user, I want a federated solution to handle user attributes, so that multiple platforms can share these attributes and use them to inform authorisation and service provision decisions	Identity Access and Management			
UC63	As a user, I want my resources to be accessible only by me and those that I selectively permit	Identity Access and Management			
UC64	As a user, I want to selectively share my resources with other users/groups	Identity Access and Management			

UC65	As a user , I want to receive authorised access to resources that have been shared with me	Identity Access and Management			

3.1.14 Operational Performance

This use case relates to supporting platform operators ensure reliable performance by monitoring services and perhaps taking corrective action.

One of the challenges will be to ensure this use case is effective when collaborating with other dependent use cases. These may include Resource Health Check, Application Testing and Test AI Model use cases. It may be that the project can consider this during early development. It might be possible to write high level integrated tests at the beginning of the project that initially fail, but guide integration and platform quality, a type of test first integration. This use case is also likely to make use of the Notification & Automation Building Block.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC66	As a platform operator, I want to automatically check accessibility of collections, so that I know if there are problems	Operational Performance			
UC67	As a platform operator, I want to automatically check reliability of processes, so that I know if there are problems	Operational Performance			

3.1.15 Resource Health Check

This use case is concerned with confirming the correct functioning of workflows and data both informally and formally. Formal checks meaning agreed as part of a published KPI perhaps as part of a service level agreement. Informal meaning as part of development best practise.

This use case is related to the Resource Health Building Block.

Challenges here will include defining KPIs, managing checks across platforms, authentication and authorisation issues and managing risk where a single change affects numerous platforms.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC68	As a user, I want automated checks to verify the continued correct end-to-end functioning of my published workflows, so that I can receive notification when there is a problem	Resource Health Check			
UC69	As a user, I want automated checks to verify the continued availability of my published datasets, so that I can receive notification when there is a problem. Availability of the dataset refers to discovery in the Catalogue and access via Data Retrieval and Visualisation services	Resource Health Check			
UC70	As a user, I want KPIs to be recorded for my published service, in order to demonstrate that I have met the SLA of my NoR offering. In this context, as a published service includes published datasets, workflows, ML models, Dashboards, etc.	Resource Health Check			

3.1.16 Application Testing

The use case is broadly speaking concerned with help support the quality of applications, services and workflows. Quality can be broken down in various ways. This might include Development Best Practice which considers code analysis and the maintainability of code and perhaps unit test coverage. Another consideration might be Quality Tooling which might support Development Best Practice and for instance identify code weaknesses or vulnerabilities. Finally, Application Performance can also be considered and might provide test functionality at both a unit level and system integration level as well as considering performance.

This use case is related to the Application Quality Building Block.

One of the challenges will be to support easy adoption of testing solutions. Perhaps the biggest challenge will be to enable broken tests to be easily fixed, so that good test coverage is maintained. It is important that tests have access to reliable and stable data sets and also that tests can easily be written with longevity in mind.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC71	As a user, I want to access to the logs of my workflow execution, so that I can verify correct function and debug/investigate failures	Application Development			
UC72	As a user, I want an Application Testing Service to performance test the speed of my service, so that I know it Is fit for purpose	Application Testing			

UC73	As a user, I want Application Testing Service to robustness test the reliability of my service, so that I know it Is fit for purpose	Application Testing			
UC74	As a user, I want Application Testing Service to access the quality of my code so that I know it is easy to maintain and share with others	Application Testing			

3.1.17 Accounting

This use case is interested in understanding and collecting the amount of resources consumer by an identified user across one or more platforms and then managing the accounting mechanisms for these resources.

This use case comes with many challenges, firstly to record the resource consumed by a uniquely identified user in an accurate way that is open to scrutiny if challenged by the user. Secondly, it is important to consider a mechanism to support accounting across federated platforms using an integrated solution. This may well be helped by agreeing a common IDP (Identity Provider) across all platforms.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block
UC75	As a user, I want to obtain NoR credits, so that I can exploit 'non-free' platform services	Accounting			

UC76	As a user, I want my NoR credits to be applicable across any platform amongst a federated set, so that I can conduct my work on the most appropriate combination of cross-platform services	Accounting			
UC77	As a platform, I want to account for the resource usage of each user, so that I can limit their usage to that which they are entitled	Accounting			
UC78	As a platform, I want to draw-down the user's credits, so that I receive compensation for services provided	Accounting			

3.1.18 Natural Human Interface

This use case is concerned with the provision of better human interfaces that enable humans to better access and use the underlying capabilities of EOEPCA+.

One of the challenges here will be to better understand where users are more likely to struggle and need support. In this regard some outreach support might be helpful to identify problem areas. It may also be that user feedback can be built into the resource provisioning aspects of EOEPCA+ to help gain better user insight.

ID	User Story (requirement)	Use Case	Tech Standards	Epic	Building Block

UC79	As a user, I want the natural language processor (NLP), to interpret my Resource Discovery request, so that I get a better response than I would without the natural language processor (NLP).	Natural Human Interface			

3.2 Non Functional Requirements

The following nonfunctional requirements are defined.

User stories must consider the following criteria to be considered done:

- Deployment instructions are provided.
- New code does not break existing automated System End-to-end Tests.
- New code does not break existing automated Test suite Execution (building block tests).
- Building Block design is updated if agreed by contractor.
- Automated unit tests are provided if agreed by contractor (building block tests).
- Testing instructions are provided if agreed by contractor.

4. R03 - USE CASE VALIDATION AND EPIC PRIORITISATION

This document will be shared with other teams to help support use case validation and epic prioritisation.

This is a living document, and it is expected that it will evolve over serval iterations.

In addition, in order to engage and communicate with stakeholders, the Outreach Manager will specifically target key stakeholders to improve the understanding of emerging standards and technologies.

Additional communication channels include:

- Web Portal
- Newsletter
- Social media
- GitHub

5. R04 - ADDITIONAL UTILISATION DOMAINS

Additional utilisation domains will be identified by ESA as the project evolves. This document will be updated to support new utilisation domains.